RESEARCH REPORT

PREDICTING RESIDENTIAL DEMAND: APPLYING

RANDOM FOREST TO PREDICT HOUSING DEMAND IN CAPE TOWN

BY

ROSS DYER

A research report presented to the Department of Construction Economics and Management in partial fulfilment of the requirements for a Degree of Master of Science in Property Studies from the University of Cape Town.

October 2018

CON5010Z | Final Dissertation

ABSTRACT

The literature shows that Random Forest is a suitable technique to predict a target variable for a household with completely unseen characteristics. The models produced in this paper show that the characteristics of a household can be used to predict the Type of Dwelling, the Tenure and the Number of Bedrooms to varying degrees of accuracy. While none of the sets of models produced indicate a high degree of predictive accuracy relative to hurdle rates, the paper does demonstrate the value that the Random Forest technique offers in moving closer to an understanding of the complex nature of housing demand. A key finding is that the Census variables available for the models are not discriminatory enough to enable the high degree of accuracy expected from a predictive model.

ACKNOWLEDGEMENTS

I would like to thank Mr. Robert McGaffin for his guidance and expertise around the dynamics of Property Markets in South Africa as well as his input into the structure and development of this Masters Mini-Dissertation.

Similarly I would like to express my gratitude to Dr. Juwa Nyirenda who co-supervised this mini-dissertation and has provided invaluable direction and assistance to the statistical portion of this paper, specifically with regards to the Random Forest technique as well as the R-Programming software.

My gratefulness goes out to my wife, Chloe Dyer, for her consistent love, support and understanding throughout the development of this paper.

DECLARATION

I know the meaning of plagiarism and declare that all the work in the document, save for that which is properly acknowledged, is my own. This thesis/dissertation has been submitted to the Turnitin module (or equivalent similarity and originality checking software) and I confirm that my supervisor has seen my report and any concerns revealed by such have been resolved with my supervisor.

Signed by candidate

21 October 2018

Ross Dyer

Date

CONTENTS

LIST OF FIGURES AND TABLES

Figures

Tables

ABBREVIATIONS

AI	Artificial Intelligence
CDSAM	CityMark Dashboard for South African Metros
FLISP	Finance Linked Individual Subsidy Programme
MDA	Mean Decrease in Accuracy
MDG	Mean Decrease in Gini Coefficient
ML	The field of Machine Learning
OOB	Out of Bag Error
R	R Statistical Programming Software
RF	Random Forest
RL	The Machine Learning field of Reinforcement Learning
RLA	Reinforcement Learning Algorithm
RStudio	User Interface add-on for R
SDT	Supervised Decision Tree
SL	The Machine Learning field of Supervised Learning
SLA	Supervised Learning Algorithm
STATS SA	Statistics South Africa
SVM	Support Vector Model
SVR	Support Vector Regression
UL	The Machine Learning field of Unsupervised Learning
ULA	Unsupervised Learning Algorithm

1. INTRODUCTION

This study aims to segment the Cape Town housing market and produce predictive models by applying the Random Forest statistical technique to the 2011 Census data. The background section of this paper illustrates the need for a more in-depth understanding of housing demand in order for the public sector to develop improved housing policies and the private sector to more successfully tailor housing products to meet housing demand. The background is followed by the problem statement after which the research questions and research propositions are stated. The research objectives are then listed. The research methodology is introduced, the research method is detailed, after which, the scope and limitations of the research are discussed. The chapter concludes with the structure of the report.

1.1. Background

Introduction to the South African Housing Problem

Pre-existing colonial and post-colonial influences coupled with the apartheid regime saw social and economic benefits flowing to only a few South Africans. Goodlad (1996) explains that spatial segregation was the key to limiting these benefits by excluding groups of people from land within the boundaries of towns and cities in the pre-democratic era. As such, supply of housing in the cities only developed to accommodate those living in the cities and demand for city housing was not met. In 1994 when South Africa became a democracy previously disadvantaged groups of people were able to access these benefits and there were large influxes of people moving into metropolitan areas seeking jobs and improved lives. The population between 1996 and 2011 for major cities such as Johannesburg and Cape Town increased by 68% and 46% respectively, while the total population of South Africa rose by only 28% (Statistics South Africa, 1996, Statistics South Africa, 2011). This accelerated demand could not be met and has resulted in a severe housing bottleneck despite government intervention in the form of the SA housing programme (Ndziba et al., 2015).

McGaffin and Kirova (2016) acknowledge the problematic nature of the current South African housing delivery model. Both private and public sector attempts at housing delivery are failing to adequately address the existing housing backlog and cater for growth of new household's (Financial and Fiscal Commission, 2012). The failure of the private sector to provide appropriate financial instruments to the lower end of the market (Financial and Fiscal Commission, 2012, Ndziba et al., 2015) has resulted in a failure to deliver affordable housing stock to large segments of the population (McGaffin and Kirova, 2016). The public sector failure stems from an increasing fiscal constraint relative to increasing demand and an inadequate delivery capacity (McGaffin and Kirova, 2016). The housing delivery model has accentuated the pre-existing ineffective, unbalanced and unfeasible nature of South African cities due to poor location and lack of density which is commonplace in many housing developments (McGaffin and Kirova, 2016). In addition due to excessive regulation and standards, historical inertia as well as the need for economies of scale and equity, the model tends to produce relatively standardised housing products that do not meet the highly diverse needs of household's (McGaffin and Kirova, 2016). The current state of affairs shows a lack of understanding around housing markets and there is a clear need to improve this understanding to meet current and future housing demand. While it is noted that the housing problem is countrywide the study is limited to Cape Town as much of the work that this study builds on is focused on Cape Town.

Current Definition of the Housing Market

Due to the heterogeneous nature of housing, a number of submarkets occur. Submarkets are based on the economic concept of substitution as defined in consumer choice theory (Watkins, 2001). Consumer choice theory suggests that a dwelling type is chosen on the basis of price, a household's income and the characteristics of that household and a dwelling (Milgrom et al., 2011). Hogarth (2015) builds on this relationship between consumer choice theory and substitution, by identifying that the urban housing market can be seen as a set of distinct but interconnected submarkets and household's view houses within a certain sub-market as more or less perfect substitutes. A poor understanding of household's needs resulted in an incorrect understanding of the housing market as only being limited to three submarkets; The Traditional Mortgage

Market, The Gap Market and The Government Subsidy Housing Market (McGaffin and Kirova, 2016).

The Gap market accounts for all household's that neither qualify for subsidy housing nor can afford any housing instruments made available by financial institutions. The lower end of the Gap Market is defined as household's with a maximum monthly household income of R3,501. The upper end of the scale is defined as household's capable of accessing finance instruments to purchase houses in the bonded market, this is currently estimated at a monthly household income of R20,000. The upper end of this market segment has around 5 times the purchasing power of the lower end of the segment. It therefore stands to reason that the nature of demand within this broad band of household income varies greatly, yet demand within this band is generally seen as consistent (McGaffin and Kirova, 2016). As such, it is unsurprising that both state and private intervention in the Gap Market has largely been a failure (McGaffin and Kirova, 2016).

Towards improved segmentation of housing markets

To solve the housing problem requires the supply of a larger range of housing products, both from a construction and finance point of view, which better match the needs, preferences and purchasing power of household's (Ndziba et al., 2015). The supply market needs to appropriately respond to demand by providing the correct product at the correct time and price-point (Ndziba et al., 2015). Both Hogarth (2015) and McGaffin (2014) highlight the importance of understanding both supply and demand. Mtanto and Churr (2014) confirm that insufficient attention has been given to understanding housing demand, and McGaffin and Kirova (2016) point out that; the lack of nuanced understanding around the nature of housing demand is a critical factor causing a failed housing delivery model. Hogarth (2015) is in agreement, indicating that there is significantly more information to determine housing supply and value than there is to determine both effective and social demand, the two components of total demand, which are represented by the private sector and the public sector respectively (UN-Habitat, 2011). It is therefore critical that the characteristics that drive different forms of demand are known and understood and that household's are segmented into suitable sub-markets (Ottawa, 2007).

Hogarth (2015:2) defines segmentation as;

"*The process of dividing the total population of household's into submarkets, within which household's have certain characteristics which generate similar preferences and levels of demand for certain products*".

Segmenting housing markets not only requires an understanding of affordability for each demand sub-market but also an understanding of the multi-dimensional nature of housing demand, including housing type, location and tenure (Mtanto and Churr, 2014). So far, it appears as if a homogenous group of consumers has been assumed in the housing delivery model and a superficial understanding of demand has led to the use of income as a cursory means of segregating the market (McGaffin and Kirova, 2016). It is important to note that the diverse characteristics of household's themselves have an effect on shaping demand for a particular housing product – these have for the most part been ignored in the past (McGaffin and Kirova, 2016). In summary there is a need to understand how household characteristics shape demand for different types of housing.

Segmentation and Disaggregation Approaches

Hogarth (2015) indicates that the most commonly adopted approaches used to define housing submarkets are spatial, affordability and structural or any combination of these.

The **spatial** approach uses location as the basis for disaggregation of the housing market, this can be completed based on statistical clustering of areas with similar characteristics as in "Whereabouts London" by Future Cities Catapult or more commonly on a suburb, or neighbourhood basis, as in the CityMark Dashboard for South African Metros (CAHF 2015) (Bourassa et al., 1999, Hogarth, 2015, McGaffin and Kirova, 2016).

The **Affordability** approach segments housing demand based on a household's affordability ratio, and disaggregates housing markets based on valuation rolls, to determine the surplus/ deficit in each submarket.

The **Structural** approach uses the characteristics of housing types or tenure to define the submarkets, examples of housing types are single-detached, flats or townhouses, and examples of housing tenure would be ownership or rental. UN-Habitat (2011) indicates that census data or household surveys can be used to determine the propensity of a

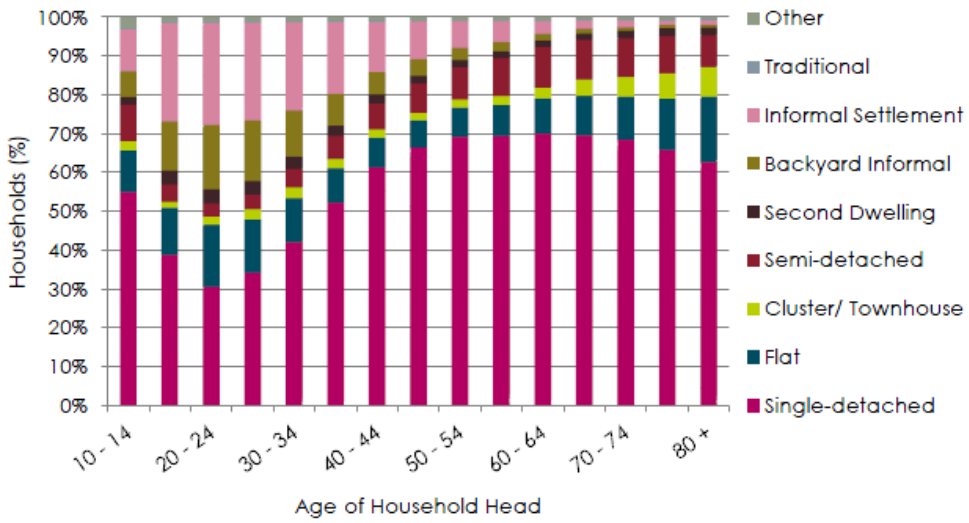
household to demand a certain type of housing based on household characteristics such as age, income, size of household etc.

While these three approaches can be used in both the segmentation of demand and the disaggregation of supply the focus of this paper is on the segmentation of demand and will therefore only explore methodologies used for the segmentation of demand.

Single-Variable Segmentation Techniques – Proportional & Statistical

The structural and affordability approach was taken by both Hogarth (2015) and Ndziba et al. (2015). Hogarth (2015) looked at the different household attributes in isolation to determine what types of houses were demanded when the attribute had different values, i.e. what proportion of household's in a category demanded a particular type of housing. For example Figure 1 below shows the type of house that is demanded per age bracket of the household head. While it is clear that the housing type could be predicted from this single variable, this method would ignore all other household characteristics that may be relevant and would likely produce poor results.

Figure 1 - Housing Type Propensities by Age of Household Head (Hogarth, 2015)



Hogarth (2015) presents the case for statistical testing to determine the most significant factors influencing housing preferences/demand and this was followed through by Ndziba et al. (2015) based on single household attributes. Ndziba et al. (2015) used a Cramer's V statistic to show that the single household attribute that had the most

significant relationship to the type of house chosen was household size, followed by the race of the household head. The least significant attribute was the gender of the household head (Ndziba et al., 2015). These results are shown in Table 1 below in blue and orange respectively.

Household Characteristic	df	Cramer's V Statistic	Effect Size
Gender of Household Head	1	0.08	Small Effect
Age of Household Head	2	0.24	Medium Effect
Household Size	2	0.39	Large Effect
Household Income	2	0.24	Medium Effect
Race of Household Head	3	0.29	Medium Effect

Table 1 – Cramer's V Statistic single attributes (Ndziba et al., 2015)

While this approach is useful to determine the most relevant household attribute that will cause a household to choose a particular type of housing, in reality a combination of household attributes make up this demand. As such a single attribute cannot be viewed as an accurate predictor of the type of housing a household will demand.

Multi-Variable Segmentation Techniques - Clustering

The clustering approach groups household's into different categories based on similar characteristics (McGaffin and Kirova, 2016). There are different statistical models for clustering, however, a simple and commonly used unsupervised learning algorithm is the k-means (McGaffin and Kirova, 2016). While this approach may be useful to determine the spatial demographics of areas and can assist us to learn factors that influence a household's choice of a particular type of housing, this approach cannot predict a household's choice based on its characteristics. The problem with the clustering approach is that the sample is grouped around means to determine logical clusters, and one needs to manually interrogate each cluster to determine what characteristics are most prominent (McGaffin and Kirova, 2016). While associations are developed and categories may be contain commonalities, these may be irrelevant to how household's choose housing (McGaffin and Kirova, 2016).

Determining a useful Multi-Variable Segmentation and Predictive Model Technique

The single household attribute segmentation methods described above oversimplify the complex nature of housing demand, and do not provide the in-depth understanding that is required to address the current housing problem. Similarly the multivariable clustering methodology fails, in that it requires groupings to be determined by the researcher in advance and clusters may have no bearing on how household's choose housing. McGaffin and Kirova (2016) note that while multi-class logistic regression may also be a possibility, tree-based methods are more suitable in cases where the data contains non-linear decision boundaries, as is expected to be case in the Census data. Another predictive modelling technique that could be used is Support Vector Machine (SVM), this technique can be used to achieve a high level of accuracy through optimisation, however, SVM's do not naturally extend to multiple class problems (Torralba et al., 2007, Krammer and Singer, 2001). The supervised Random Forest model offers advantages when working with both numerical and categorical, working with varied response distributions and non-linear decision making boundaries in the data (Breiman et al., 1984, Clark and Pregibon, 1992, Ripley, 1996). These characteristics are all present in the Census data and therefore it is argued that a supervised Random Forest approach is the most suitable.

This paper aims to show that a supervised Random Forest model is a useful tool to segment housing demand and to create a model capable of predicting housing type, tenure and number of bedrooms given household information.

1.2. Problem Statement

The problem to be examined in this study may be stated as:

(a) *An inadequate understanding of housing demand in South Africa has curtailed the ability of both the public and private sector to provide appropriate housing products to satisfy current demand,*

(b) *Market Segmentation methods used to address this problem to date are either too simplistic or conceptually flawed.*

1.3. Research Questions

The research questions may be stated as:

- Is a random forest method of segmentation suitable to overcome the shortcomings of the previously applied methods,
- If so, could the Random Forest models be used to predict a target variable (Housing Type, Tenure and Number of Bedrooms) for a new household with completely unseen characteristics?

1.4. Research Objectives

The research objectives to be achieved are to:

(a) Ascertain current methods of segmenting housing demand,

(b) Identify the appropriate statistical method to construct predictive models for the selected variables,

(c) Gain access to a dataset with suitable information regarding household attributes and housing types in a suitable format, or convert data into a suitable format,

(d) Construct and run Random Forest Models,

(e) Analyse the results of the Random Forest Models,

(f) Draw conclusions.

1.5. Research Methodology

The above objectives will be achieved by adopting the following research method;

(a) An in-depth literature review will be undertaken including an overview of the context of the housing problem in Cape Town, as well as a comprehensive review of consumer choice theory, housing segmentation and suitable machine learning algorithms,

(b) A quantitative methodology will be adopted where a secondary data analysis research design will be applied. The Random Forest Models will be built to analyse the 2011 Census population data for Cape Town,

(c) Conclusions and recommendations will be drawn.

1.6. Scope and Limitations

This study is subject to the following limitations;

(a) The accuracy of the study is dependent on the accuracy and integrity of the Census 2011 data,

(b) Due to resource constraints only secondary data analysis could be conducted on the Census 2011 data, no direct field surveys have been completed,

(c) The study will only be conducted on data for Cape Town,

(d) Current demand drivers may not be suitable predictors for drivers of future demand.

1.7. Structure of the Report

The research report will be structured in five chapters.

Chapter 1 | outlines the area of study and provides the background to the report. Random Forest is introduced as a technique to segment the Cape Town housing market and construct a predictive model. Then the research problem, research question(s) and research proposition are stated. The aim and objectives of the research are then laid out, followed by a description of the methodology. The chapter ends with the scope and limitations of the research.

Chapter 2 | provides and overview of the context of the housing problem in Cape Town, then lays out the theoretical and conceptual framework background for the research. Focus is given to both the theory of housing demand and market segmentation and the chapter highlights the degree to which research has already been completed in this area of study. The chapter explores why Random Forest has been selected as an appropriate tool to segment demand and construct the predictive models.

Chapter 3 | establishes the methodology used during the course of this research and offers justification for why secondary data analysis has been chosen as the research design. The chapter outlines the procedures to be followed when

using secondary data analysis and records its implementation.

Chapter 4 | includes the presentation and analysis of the findings of the research in relation to the literature review.

Chapter 5 | Draws conclusions from both the literature review and the findings. The research question is then answered, followed by the presentation of recommendations and topics for further research.

The research report closes with a full list of references as well as appendices.

2. LITERATURE REVIEW

2.1. Introduction

This chapter begins with an overview of the Cape Town Housing Context. The chapter then goes on to introduce Consumer Choice Theory and explores this as the theoretical basis for housing demand. The research then defines the concept of sub-markets and introduces market segmentation approaches before discussing the lenses through which housing demand market segmentation and supply market disaggregation can be viewed. The theory of propensity and probability is introduced. Demand Market Segmentation Techniques used to date are explored, after which the field of Machine Learning is presented. Types of Machine Learning and suitable Machine Learning Algorithms (MLA's) are discussed. Decision Tree and Random Forest algorithms are then covered and their application to the Segmentation of Housing Demand and construction of predictive models is discussed. This is followed by the conclusion of the chapter.

A chapter roadmap has been included below in Table 2 to guide the reader through the major topics dealt with in the literature review chapter.

General Area	Section	Sub-Section
Overview of the Cape Town Housing Context	Overview of the Cape Town Housing Context	
Introduction of Key Concepts in Housing Demand Theory & Marketing Theory	Consumer Choice Theory	
	Propensity and Probability	
	Defining Sub-Markets	
	Market Segmentation Approaches	a priori & post-hoc
	Housing Market Lenses	Spatial, Affordability & Structural
Techniques Used to Date	Single Variable Segmentation Techniques	Proportional & Statistical
	Multi-Variable Segmentation Techniques	Clustering, Multiple Regression & Discriminant Analysis
Artificial Intelligence & Machine Learning	Introduction to Artificial Intelligence and Machine Learning	
	Types of Machine Learning	Reinforcement, Unsupervised & Supervised Learning
	Suitable Learning Algorithms	Support Vector Machine, Decision Tree & Random Forest
Selecting and Exploring the Selected Machine Learning Algorithm	Selecting an appropriate Machine Learning Algorithm	
	Exploring Decision Trees	Classification & Regression Trees
	Ensemble Methods and Random Forest	Random Forest

Table 2 – Literature Review Roadmap

2.2. Overview of the Cape Town Housing Context

Key factors in the Cape Town Housing Problem

The housing problem in South Africa and more specifically Cape Town is a complex one, the context of which needs to be introduced in order to understand the greater need for research such as this. South Africa today is a highly unequal society which in many ways is rooted in its historical context of colonisation, racial and spatial segregation during the apartheid era and ultimately political liberation (Butler, 2017). Cape Town is no exception and it is commonly recognised that the segregation left as legacy of apartheid has remained entrenched in the city's spatial context (Goodlad, 1996). This is accentuated by the city's unique geographic situation; being constrained on most sides by the either the Atlantic Ocean or the Table Mountain National Park. Cape Town's move towards becoming a "Global City" could also be slowing the down measures to combat inequality (Lemanski, 2007). All this has not assisted advancements in combatting the housing problem in Cape Town and as McGaffin and Kirova (2016) point housing policy which has arguably failed to address the issue.

An overview of the Housing Supply & Demand Context in Cape Town

Using the 2011 Census data it was found that it took 3.1 times the average salary to afford the average house, in addition it was found the only 44% of properties in the metropolitan area could be considered affordable and affordability is was noted to be on the decline (WCG, 2015). Cape Town's residential property market comprises of 703,801 properties with a total estimated value of R807.5 billion in 2015 (CHAF, 2017). This can be divided into four value categories; houses under R300k, between R300k and R600k, between R600k and R1.2 mil, and finally over R1.2 mil. These categories account for 30%, 17%, 22% and 31% respectively in terms of numbers of properties in 2015 (CHAF, 2017). From a value perspective, 77% of the values is held by properties in the over R1.2 mil category (CHAF, 2017). The total value of the government sponsored housing in 2015 was about R31 billion, and related to an estimated 191,887 units (CHAF, 2017). The property market in Cape Town is clearly segregated by value with the higher value properties found in the far north, north west and south of the city bowl and the lower value properties on the

Cape Flats, south east of the city centre and in Kraaifontein east. It is therefore key for the city to create robust transport linkages between areas with lower property values and economic nodes, while investing in public infrastructure and diversity of housing stock in these areas (CHAF, 2017). From a supply perspective there was a backlog of houses for just over 329,000 household's earning less than R3,500 per month (WCG, 2015). The formal housing market (household's earning greater than R15,000 per month) seems to be functioning well even with the unfavourable affordability ratio. There a still a noteworthy share of Western Cape Households residing inadequate housing (WCG, 2015). It is interesting that 21% of these inadequately housed household's earn more than R3,500 per month and 6% earn more than R7,500 per month (WCG, 2015). The total backlog across all income groups in the Western Cape is 414,212 household's.

From a Tenure perspective, a greater share of higher income household's own (whether full paid or being paid off) when compared to lower income brackets as would be expected. Finance appears to be accessed at the R6,400 – R12,800 per month level as there was a rapid increase in household's in the owned but not yet paid off category (WCG, 2015). Rental accounts for a significant portion (20%) of household's across all income groups which seems to indicate that rental is often influenced by factors other than income.

2.3. Understanding the Housing Market

To understand the housing market both supply and demand needs to be understood (McGaffin, 2014). While understanding supply is important to understanding the market as a whole, the focus of this paper is on better understanding and predicting demand. UN-Habitat (2011) states that the factors influencing demand are known as the drivers of demand and include; income, demographics, price of housing, cost and availability of credit, consumer and investor preferences, and the price of substitutes and complements. Total demand can be made up of effective demand (household's willing and able to pay for available housing) and social demand (household's that require government assistance to access housing) (UN-Habitat, 2011). Consumer choice theory is investigated to further explain housing demand.

2.4. Consumer Choice Theory and Housing Demand

Housing Demand has its roots in Consumer Choice Theory. Consumer Choice Theory is the study of how consumers make purchasing choices given their preferences and budget constraints. The theory seeks to explain why consumers make choices given their income and the relative prices of goods and services, in an effort to understand current demand and predict how future demand will be shaped.

The application of Consumer Choice Theory to Housing Demand would imply that housing types are seen as goods with varying characteristics. What type of housing product is chosen by a particular household would therefore depend on that household's preferences and income level. This paper uses household demographic data to predict a household's preference for the Type of Dwelling, Tenure and the Size of Dwelling. Lancaster (1966) presents a refined view of the nineteenth-century utility theory to provide what is referred to as the then "New Approach to Consumer Theory", which as Ndziba et al. (2015) point out, is now widely accepted and used for the statistical modelling of consumer choice.

The three core ideas of Lancaster (1966)'s refined theory are explained below;

1. The characteristics of the good gives rise to utility, not the good itself. This concept provides a basis on which to measure variation in utility of similar goods with diverse characteristics. Thus it is possible to distinguish the utility a certain household may get from a different type of housing. For example, for a particular price range a household may get more utility out of a flat than a single detached dwelling,
2. The characteristics of a good that possesses multiple characteristics may be shared by multiple similar goods. Types of housing, although distinguishable from each other as unique types of goods, will have overlapping and shared characteristics. It is the proportion of these characteristics that will have a bearing on how household's select housing based on their preferences for each characteristic,
3. The combination of a collection of goods will give rise to a set of combined characteristics that may not be present in any of the goods when consumed in their individual capacity. For example, a household may not be willing to take a

loan to purchase a car, however, the same household may be willing to take a loan to purchase a house. In other words the characteristics of a loan are not attractive to that household on their own however, these characteristics, coupled with the characteristics provided by the house, create a unique set of combined characteristics that are attractive for the household.

For sake of simplicity a good is discussed in points above, however, the word good or service can be used interchangeably in these principals.

It is important to note, as Ndziba et al. (2015) point out that a consumer's choice is ultimately constrained by both affordability and availability. The choices that consumers make will therefore not necessarily be directly reflective of their preferences but will also be shaped by the constraints that they encounter. This is critically important in the housing market as housing supply generally has a slow turnaround time and household's generally spend a large proportion of their income on housing. Affordability in this regard is generally seen as between 25 – 30% of a household's gross income (McClure, 2005, Hogarth, 2015).

Consumer Theory indicates that consumers will attempt to achieve the highest possible level of satisfaction or utility (Lancaster, 1990). Consumer utility is gained from the characteristics of a good / service and can be traded off against the characteristics of an alternative good / service, this trade-off decision can be represented by an indifference curve as shown in Figure 2 below.

Figure 3 - Typical Indifference Curves (Viruly, 2015)

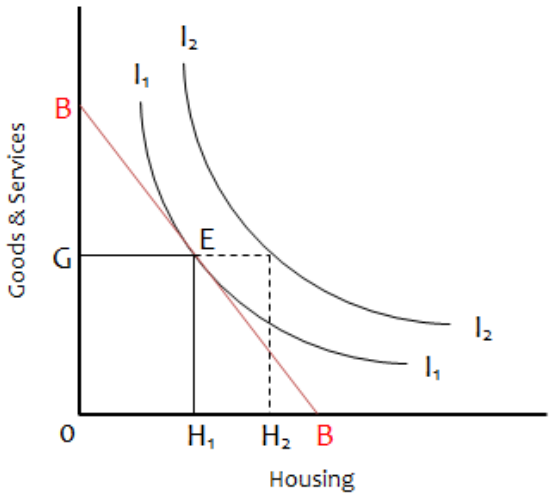
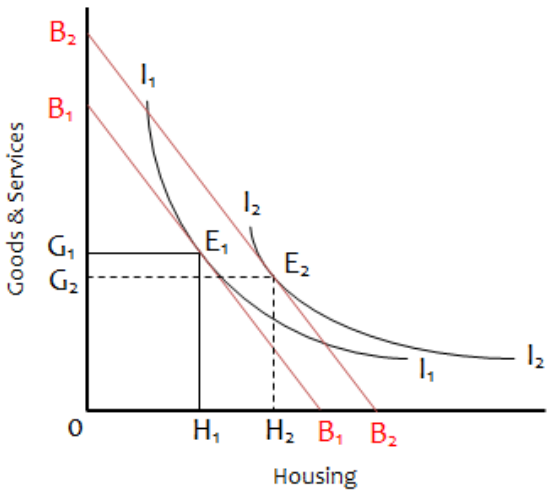


Figure 2 - Income effect

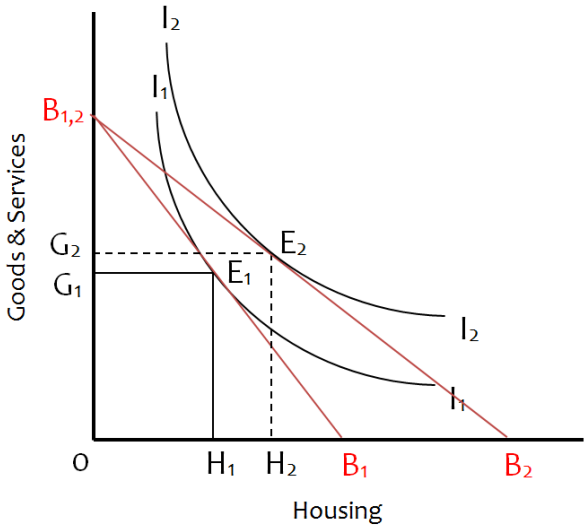


A consumer, or household, will aim to achieve its highest possible indifference curve – the indifference curve that is attainable will depend on budget constraints (Viruly, 2015). In the figure below, a household would want to reach Indifference Curve I_2, however, its budget constraints (B-B) would only allow it to reach Indifference Curve I_1. The household will therefore seek to maximise utility from the combination of goods that is represented by Indifference Curve I_1-I_1 (Viruly, 2015).

The **Income Effect** shifts a household's budget upwards, as shown in Figure 3 above. The household's Budget Curve now intersects Indifference Curve I_2-I_2 and equilibrium is attained at point E_2. This shift shows that the household has an appetite for additional housing relative to other goods and services, all else equal. This is due to the fact that with an increase in income the household demands more housing (shift from H_1 to H_2) while the household demands less of other goods and services (shift from G_1 to G_2). As Ndziba et al. (2015) point out, the income of a household directly impacts the amount of capital that can be invested in housing as well as a household's ability to access finance – this is key as most household's required long-term finance in order to enjoy the benefits of housing.

The **Substitution Effect** is caused by a relative change in prices. The most classic example of this is demand side policy intervention in the housing market, in the form of a housing subsidy. As Figure 4 below shows, a subsidy would effectively drop the price of housing for a household, which would cause the budget to shift from B_1-B_1 to B_2-B_2, this would enable the household to move to Indifference Curve I_2-I_2 and would enable the household to purchase more housing (shift from H_1 to H_2) as well as more goods & services (shift from G_1 to G_2).

Figure 4 - Substitution Effect, adapted from (Viruly, 2015)



Consumer Theory has been used extensively to model consumer behaviour and can be used to understand the relationship between the characteristics of a household and how a household chooses types of housing based on the characteristics of that housing. Price is generally seen to be the primary defining constraint relative to the income of a household and price of housing is determined by the location, choice of tenure and the type of dwelling. A household's preference will inform the selection of housing and will be constrained by availability and affordability of housing (Ndziba et al., 2015). The attributes of housing demand, determined with statistical significance, by Quigley (1976) were location, type and tenure. Quigley (1976) showed household's with different characteristics viewed these housing attributes differently. This indicates that household characteristics influence dwelling type choice (Ndziba et al., 2015).

2.5. Propensity and Probability

The propensity theory of probability is an interpretation that views probability as a physical propensity, tendency or disposition of a given physical situation to yield a particular kind of outcome, or to yield a long-run relative frequency of a particular kind of outcome (Hájek, 2003). The theory of propensity was introduced by Popper (1959) and Albert (2007) recognises it as being the accepted statistical and quantitative measure for decision making. This research aims to better understand and predict three dependent variables (Housing Type, Tenure & Number of Bedrooms) each considered separately when given certain household characteristics (independent variables). The selection of these variables will be explained in chapter 4. The literature has determined the degree of significance between certain variables in order to establish the probability of recurrence. As Ndziba et al. (2015) indicate, if there is a strong degree of significance between variables this would have a bearing on the natural tendency of household's and as such their propensity to demand certain types of housing.

2.6. Defining Sub-Markets

The theory of sub-markets and the concepts of market segmentation and disaggregation first appear in marketing literature and are translatable to housing markets. As such,

marketing literature will be reviewed in order to introduce the concepts. The importance of the concepts stems from the heterogeneous nature of markets as identified in the marketing literature (Alderson, 2006, Assael and Roscoe Jr, 1976, Claycamp and Massy, 1968, Smith, 1956) which makes markets difficult to understand and ultimately to supply suitable products.

In terms of understanding submarkets Hogarth (2015) defines **Disaggregation** as; "*the process of dividing the total housing stock into submarkets*" (i.e. supply side). As the focus of this research is on the segmentation and prediction of market demand, disaggregation will not be dealt with in further detail. **Segmentation** is defined by Hogarth (2015) as; "*the process of dividing the total population of household's into submarkets*" (i.e. demand-side).

Market segmentation helps to better understand the market and meet the needs of the homogeneous submarket consumer populations (Dolničar, 2004). Wedel and Kamakura (2012) point out that market segmentation is important as the heterogeneity of consumers' needs must be accounted at a certain point if one is to continue to produce and sell goods. Smith (1956) defines market segmentation as the viewing of a heterogeneous market as a number of smaller homogeneous submarkets, accounting for differing consumer preferences, in an effort to increase consumers' satisfaction in relation to their varying inclinations. Segmentation results in additional demand profiles (each relating to a market segment) as opposed to a single demand profile for the market as a whole. Market segmentation can therefore be used to better capture the demand of each submarket (Smith, 1956). This is shown clearly in Figure 5 and 6 below.

Figure 5 - Market Penetration without Segmentation (Dobney, 2012)

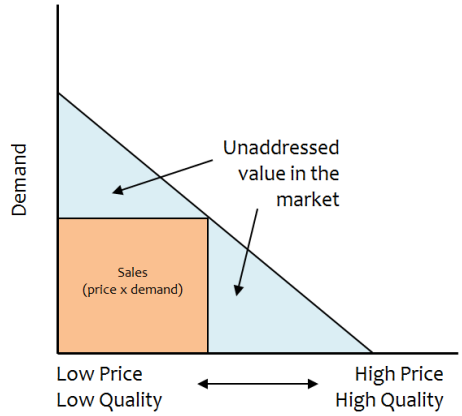
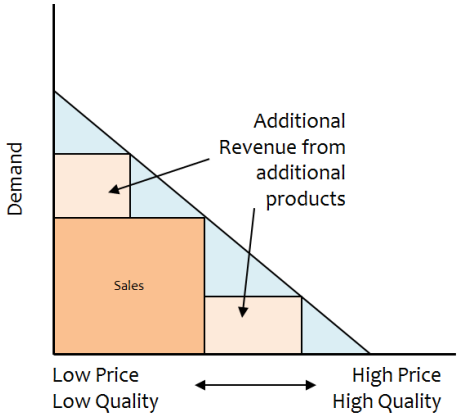


Figure 6 - Market Penetration with Segmentation (Dobney, 2012)



The above is shown for different prices and different levels of quality but essentially the independent variable could be any housing characteristic. This is a simplified view when looking at demand for housing based on only the "price-point/ quality". This does however become more complex when looking at one-to-many relationships[1] as is the case when predicting a household's demand for a particular type of housing, tenure or number of bedrooms. Both Galster (1996) and Hogarth (2015) acknowledge that a housing market can be made up of a set of distinct but interrelated submarkets. It is important to note that each of these interrelated submarkets are substitutes for one another and household's compete for housing in each of these submarkets.

It is possible to ascertain how the characteristics of a household affect the choice of housing type by segmenting household demand and disaggregating housing types (Ndziba et al., 2015). To better contextualise the applicability of market theory to the housing market, both household's and housing have been defined.

A **Household** is defined in line with the (StatsSA Census 2011 questionnaire) definition as;

"A group of persons who live together, and provide themselves jointly with food or other essentials for living, or a single person who lives alone."

Housing on the other hand is defined by Galster (1996) as a;

"spatially immobile, highly durable, highly expensive, multi-dimensionally heterogeneous and physically modifiable commodity."

As can be seen by the definitions above, both household's and housing will be made up of many unique aspects and it is due to these varying preferences of household's and the physical heterogeneity of housing products that submarkets will exist within the overall market (Ndziba et al., 2015). These submarkets are classified by using one of the market segmentation approaches described in the following section.

2.7. Market Segmentation Approaches

Marketing literature defines two primary approaches to segmentation; **a priori** and **post-hoc** or **data driven** (Dolničar, 2004, Kara and Kaynak, 1997, Wind, 1978). The a priori approach requires the researcher to initially select variables of interest and then classify

[1] A one-to-many relationship in this context of this paper is used to describe the relationship between a single dependent variable and multiple independent variables

the market in accordance with each variable (Wind, 1978). In this approach, the market is split up based on pre-existing demographic criteria (Dobney, 2012). The post-hoc segmentation approach, requires the researcher to select a range of interrelated variables and then group buyers into clusters with a high degree of within-cluster similarity and a low degree of between cluster similarity (Wind, 1978).

As Ndziba et al. (2015) point out, regardless of the chosen approach, the starting point in the process of segmentation is the selection of the basis of segmentation, i.e. the dependent variable. The selection of the descriptors (or independent variables) is then required. The independent variables can either be general consumer characteristics (socioeconomic, demographic etc.) or situation-specific (purchase patterns, product perception, product usage etc.) consumer characteristics (Wind, 1978).

2.7.1. a priori segmentation approach

The a -priori approach will have a high degree of member similarity in terms of the independent variables within each segment, for example segmentation may be completed based on income bracket and therefore the members in each segment will all be within a similar income range (Hoek et al., 1996). As the marketing literature shows, however, a consumers response to marketing stimuli may vary between members within a single segment (Hoek et al., 1996). The selection of the variables in an a priori study, to some degree, reflect the preconceived assumptions the researcher has about the underlying market structure and are likely to influence the findings of the study (Fuller et al., 2005). It is noted, however, that the a priori segmentation may be effective when the underlying decision boundaries are clearly related to a particular characteristic, for example in the technology sector there is such a strong relationship between age and use that this simplistic segmentation approach is sufficient (Dobney, 2012). This approach is more unstable in predicting future trends when compared to a post-hoc approach as underlying assumptions may change over time and no empirical research is conducted to validate these assumptions (Ndziba et al., 2015).

2.7.2. Post-hoc Segmentation approach

Post-hoc segmentation allows for the creation of segments along the lines of demand profile, usage habit and attitude (Fuller et al., 2005). The members in the post-hoc approach segments will therefore frequently have dissimilar characteristics (Fuller et al., 2005), if this process is done correctly, these segments will contain a homogeneous demand profile. This approach, unlike the a priori approach, requires that market segments are defined by the data as a result of empirical research on the data (Polaris, 2008). When starting a post-hoc approach the nature of the segments are not necessarily known.

While the approach taken in this paper does have aspects of an a priori approach, as the market segments in the Census data are already formed along demographic lines. It is argued however, that this paper adopts a post-hoc approach as the selection of the segments and the weighting the predictive models give each segment or portion of a segment is data driven.

2.8. Housing Market Lenses

Hogarth (2015) completed a study for the City of Cape Town which aimed to assist with the creation and implementation of new housing delivery models by better understanding the market through segmentation and disaggregation. While Hogarth (2015) uses an a priori approach to segmentation based on clustering or grouping household's with similar characteristics this paper argues for a post-hoc based predictive model that determines demand market segments based on underlying decision boundaries in the data. This will be dealt with further in the methodology section of this paper. Hogarth (2015) describes three broad lenses through which supply, demand and housing submarkets can be viewed; **spatial, affordability** and **structural**. Often a combination of these approaches is used.

2.8.1. Spatial Lens

As Bourassa et al. (1999) points out a common lens for analysing the housing market is spatial, or by location, which can be undertaken based on statistical clustering or at

various levels, for example by suburb or by neighbourhood. The spatial lens can be used to both disaggregate supply and segment demand on a geographical basis.

2.8.2. Affordability Lens

Gan and Hill (2009) identify the key affordability indicators for the rental and ownership market as follows;

- **Rental Market** (rental to income ratio),
- **Ownership Market** (Price to Income and Credit Payment to Income).

Affordable housing is defined as housing for which a household spends no more than 30% of its income, on either rent, in the rental market, or loan payments, in the ownership market (Financial and Commission, 2013, Ottawa, 2007, McClure, 2005). Hogarth (2015) quotes a recent study by the Centre for Affordable Housing Finance in Africa that found that in Cape Town it takes 3.1 times the average annual income to afford the average house. The affordability lens can be used to compare housing surpluses or shortages per submarket, with household's segmented based on income and housing stock disaggregated based on affordability for each income range. The quantity of housing units can then be subtracted from the number of household's to determine a shortage or surplus in each affordability submarket (McClure, 2005).

2.8.3. Structural Lens

The structural lens is an alternative approach to defining submarkets (Hogarth, 2015). This is done by using the structural characteristics of the market such as dwelling unit types (single-detached, flats, townhouses etc.) or tenure (ownership or rental) as the dependent variable. From this, using Census data or a similar type of household survey, household characteristics (Age of Household Head, Income, Size of Household etc.) can be matched to the type of dwelling chosen, which will show the dwelling type propensities based on certain household characteristics (UN-Habitat, 2011). This process can also be completed to determine tenure propensities (Hogarth, 2015). It is clear from the literature that limited structural research has been undertaken and Hogarth (2015) identified as part of future research recommendations that additional research is required in terms of the analysis of structural submarkets. It is noted by Watkins (2001) as

well as Islam and Asami (2009) that it is possible for two or all of these lenses to be combined to create a more precise method than any single lens would produce, for example, both spatial and structural analysis could be combined to form an integrated approach.

These broad lenses assist to establish how housing demand segmentation can be approached and the following sections review the segmentation techniques used to date in the housing literature.

2.9. Single Variable Segmentation Techniques used to date

The single variable segmentation techniques used to date are evaluated below.

2.9.1. Proportional (Descriptive Statistics)

The strategies applied in the studies below are defined in this paper as proportional. These strategies are seen as proportional because in each case the applicable proportion of the dependent variable classes is allocated to the classes of an independent variable. These proportions relative to the nature of the independent variable classes is then what has been analysed.

City of Ottawa (City of Ottawa, 2007)

The City of Ottawa (2007) used the structural lens to analyse housing demand and project this based on the age of the household head and dwelling type propensities for different age groups (Hogarth, 2015). This is a strictly a priori approach and assumes that a strong link between age and type of dwelling exists. The dwelling types used for this research were single detached, semi-detached, row housing and apartments which are

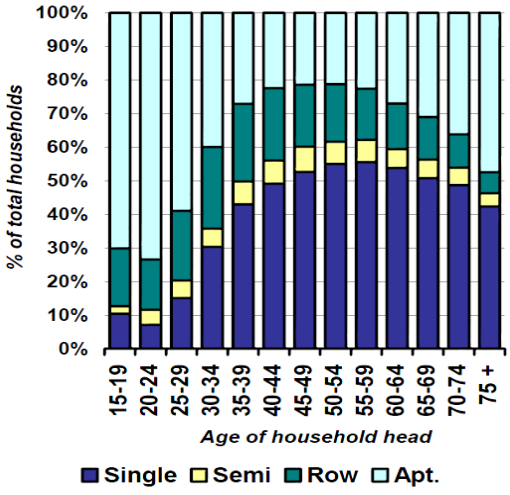


Figure 7 - Dwelling Type Propensities (City of Ottawa, 2007)

argued to account for all dwelling types in Ottawa. This analysis was done on the 2001 census data, which showed that 43% of household's headed by a person aged between 35 and 39 lived in single detached homes and 27.1% of household's in the same age group lived in apartments. This data also showed that household's tended to live in apartments in their younger years as seen in Figure 7.

City of Tshwane (Financial and Fiscal Commission, 2014)

In the South African context the Financial and Fiscal Commission (2014) started to explore the spatial and structural approaches to demand analysis and projection (affordability in each approach was considered). Based on limited questionnaires in the City of Tshwane, a housing demand model was developed showing type, tenure and location of housing and projected these under two different scenarios; Business as Usual (BAU) and Future Aspirations (FA). The results of the spatial approach found that by 2030 in both scenarios, demand for housing in the CBD and immediately adjacent suburbs will be the greatest. Structurally the market was projected to shift in terms of dwelling type, from free-standing to flats and town-houses. In terms of tenure, rental housing would be the most desired by 2030.

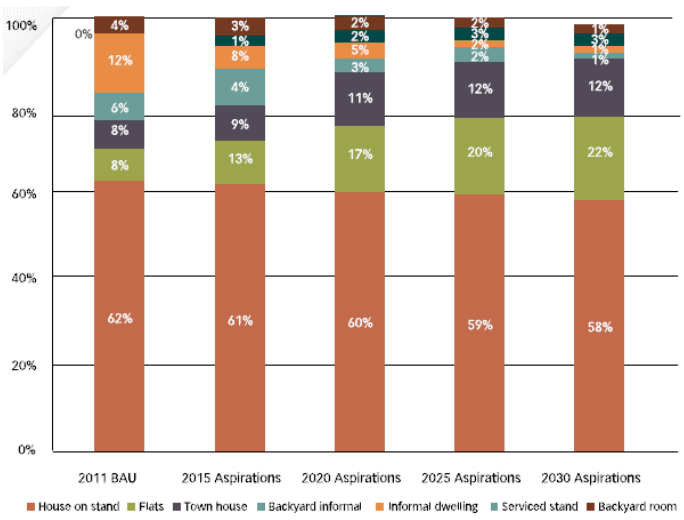


Figure 8 - Changes in preferred housing typology (Financial and Fiscal Commission, 2014)

Figure 8 above shows how the household type proportions based on the aspiration of household's interviewed relate to the BAU proportions of the Census 2011 for the market as a whole. No full market segmentation methodology has been provided in the Financial and Fiscal Commission (2014) but based on the data presentation and the conclusions drawn, this was deemed to be a proportional approach.

City of Cape Town (Hogarth, 2015)

Hogarth (2015) offers the most descriptive methodology of the proportional strategy through the structural lens, this methodology is applied to the STATS SA Census 2011 data to gain a high level understanding of the structure of the market in Cape Town. This is shown visually below in Figure 9.

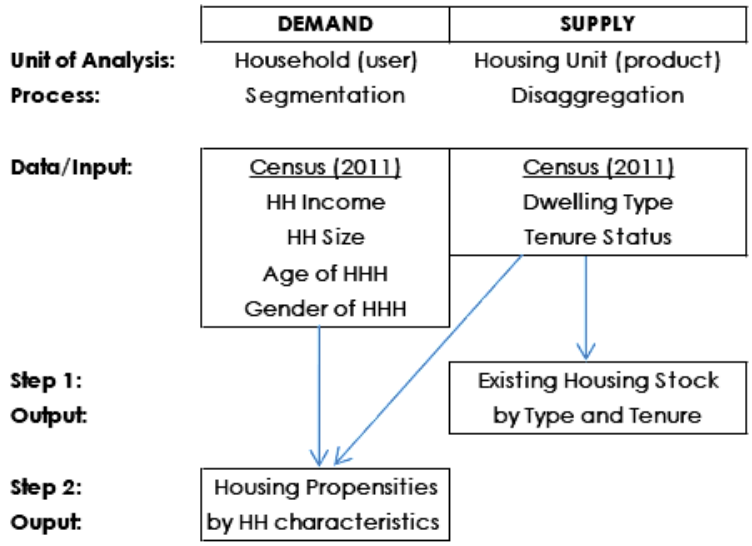
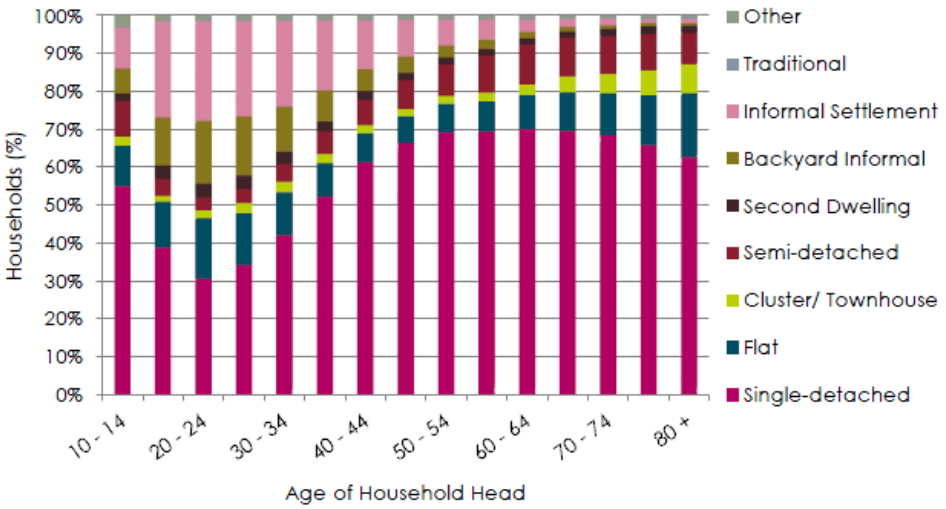


Figure 9 - Structural Analysis Methodology (Hogarth, 2015)

First, the existing housing stock type and tenure is outlined based on the Census data. Therefore two sets of analysis are being completed here with two different dependent variables (Dwelling Type and Tenure). The characteristics of the household's (household income, household size etc.) were then individually used as the single independent variable to determine the household propensities per household characteristic. Hogarth (2015) showed that cross-tabulating household characteristics and dwelling types in the first set of analysis described to some degree the propensities of household's to choose certain types of housing. Figure 10 shows the propensities of household's to demand a particular type of housing based on the Age of the Household Head, shown proportionally in different age categories. Hogarth (2015) notes that given the dominance of single detached dwellings in the housing stock this is the housing type of choice among most age groups, apart from young adult age groups. Hogarth (2015) compares this to the same results found in the City of Ottawa (2007) report which indicated that in Ottawa this lower propensity for single detached dwellings translated into a higher demand for flats, however, given the low stock available for flats in Cape

Town, Hogarth (2015) postulates that this could translate into a propensity for informal dwellings in the Cape Town context.

Figure 10 - Housing Type Propensities by Age of Household Head (Hogarth, 2015)



This same analysis shown in Figure 10 above was completed for household income, household size and gender of household head. As would be expected, lower household incomes where associated with a higher proportion of informal settlement dwelling type while higher household incomes were associated with a higher proportion of single-detached dwelling type. The most notable result from the tenure vs. household income analysis completed by Hogarth (2015) is that the "owner but not fully paid off" form of tenure increased substantially towards the higher income categories, which indicated the lack of access lower income household's have to housing finance products. Smaller household sizes tended to show a lower demand for single detached dwellings and a higher demand for flats. Smaller household's were also more likely to rent and less likely to have a loan. Hogarth (2015) identified that household's with a female household head have a higher propensity towards flats, cluster/ townhouses and semi-detached dwellings and proposed that this may be due to childcare and security related issues.

Gauteng (CSIR, 2016)

CSIR (2016) completed a study on the Gauteng housing market in mid-2016 which was very similar to the Hogarth (2015) in that it analysed different variables proportionally, the supply side was also considered in relation to demand to determine overcrowding numbers. The study also looked at projections for 2030 in an attempt to understand the future demand. The CSIR (2016) then built a model using what they refer to as a housing

adjustment pseudo decision algorithm which indicated how household's would make decisions based on changing circumstances over time. This then projected the resultant housing situation by housing type in each year leading up to 2030. The model also projected the costs, land required and opportunities for the state housing programme between 2016 and 2030 (CSIR, 2016). The interesting findings of this research showed that under different economic and population forecasts there were major challenges stemming from a growing poor population unsuitable housing conditions were projected to grow from 20 – 25%, which indicated a need to review the business as usual approach (CSIR, 2016). The CSIR (2016) then modelled six scenario's that moved away from the business as usual approach in an attempt to meet housing demand, the scenarios and the results were as follows;

1. Increasing state budget – major expense,
2. Shifting to supply of rental stock – more expensive per housing unit but with positive impacts in relation to municipal viability and urban efficiency,
3. Upgrading informal settlements – further reaching but increased budget requirements,
4. Releasing serviced land – cost effective and addresses urbanisation, however unless coupled with density-based development could lead to urban sprawl,
5. Increasing funding & availability of Finance Linked Individual Subsidy Programme (FLISP) – provided state continues to deliver in other areas this would be very effective in allowing supply to catch up to demand,
6. Shifting support to an enabled private sector – a relatively small increase in private sector capacity resulted in significant gains by 2030.

CSIR (2016) indicates that the strength of this model would be that the right balance of these different strategies that could realistically be implemented could be modelled in order to meet supply, however the paper does not indicate what mix this would be and it is understood that further investigation into this would be required.

City of Cape Town (Ndziba et al., 2015)

Ndziba et al. (2015) completed research on the Cape Town housing market in line with the methodology used by Hogarth (2015) with similar results, however, in addition to

looking at the 2011 Census data, the 2001 data was looked at to see how the market had changed. While these proportional strategies used are useful to gain some general and isolated insights into market demand, they are based on an a priori approach and therefore make presumptions about the nature of demand - it is not clear whether these underlying assumptions are necessarily correct. For example, while it is useful to see what type housing is demanded by different age groups, most notably for housing projections, choice of housing is based on a complex basket of household characteristics, each interconnected and therefore this single factor does not explain the full nature of housing demand.

2.9.2. Statistical (Inferential Statistics)

City of Cape Town (Ndziba et al., 2015)

Ndziba et al. (2015) took their research further from the descriptive statistics shown above. They applied inferential statistics to the data in order to establish dwelling type propensities based on individual household characteristics. This was done on both the Census 2001 and the Census 2011 data however for the purposes of this paper only the Census 2011 data will be covered. The results of each of the four Dwelling Type propensities considered by Ndziba et al. (2015) are shown in Table 3 below.

Dwelling Type	Propensities ---------- Decreasing effect (Cramer's V Statistic) ---------->				
	0.39	0.29	0.24	0.24	0.08
	Household Size	Race of Household Head	Age of Household Head	Household Annual Income	Gender of Household Head
Single Detached	2	Coloured	40 – 49	R38,401 – R307,200	Male
Flat/ Apartment	1	White	25 – 29	R76,801 – R307,200	Male
Townhouse/ Cluster/ Semi-Detached Dwelling	2	Coloured	35 – 54	R38,401 – R153,600	Male
Informal Dwelling	1	Black African	25 – 34	R19,201 – R38,400	Male

Table 3 – Dwelling Type Propensities for STATS SA Census 2011 data based on information provided by Ndziba et al. (2015)

This shows for example, that a household living in single detached dwellings in 2011 was most likely to have a household size of **two,** with a household head belonging to the **Coloured** racial group and an age of between **40 and 49.** The household annual income would most likely be between **R38,401 and R307,200** and the gender of the household head would most likely be **Male.** While the Cramer's V statistic shows the strength of correlation between these household variables and the Type of Housing on a single variable basis, this approach cannot determine the interrelationship between these characteristics and does not provide the basis for model capable of predicting demand based on any combination of characteristics.

2.9.3. Suitability of Single Variable Segmentation Techniques used to date

In summary problems with proportional and single variable statistical techniques used to date mean that the researcher can only assess the relevance of a single household characteristic at a time and this does not provide sufficient basis to understand the complexity of housing demand or to create a suitable predictive model. As a result, a multi-variable segmentation technique is required to overcome this problem.

2.10. Multi-variable Segmentation Techniques used to date

The multi-variable segmentation techniques used to date are evaluated below.

2.10.1. Clustering

Sydney and Melbourne (Bourassa et al., 1999)

Bourassa et al. (1999) uses the K-means statistical clustering[2] method for defining housing submarkets in Sydney and Melbourne, Australia. Bourassa et al. (1999) used principal component analysis[3] to extract the core factors (variables) from local

[2] K-means statistical clustering aims to partition all observations in the data into "k" number of clusters in which each observation is allocated to the cluster with the nearest mean. The number of clusters is selected initially by the user

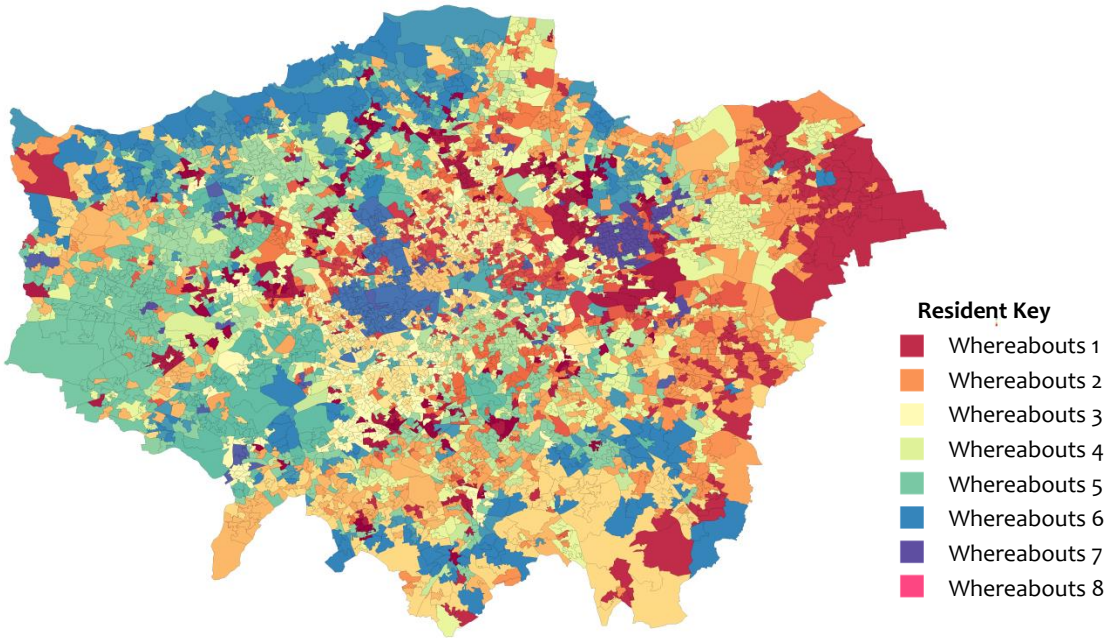
[3] Principal Component Analysis is a statistical method that reduces the dimensionality of a dataset by searching for a small number of principal components that can be linearly related to the original variables in the dataset

government area data and individual dwelling data, to be used within the context of residential location theory. From this factor scores were calculated and cluster analysis was used to determine the composition of housing submarkets. Once this had been completed, Bourassa et al. (1999) estimated hedonic price equations for each city as a whole, for a priori classified submarkets and for submarkets defined through the cluster analysis. Bourassa et al. (1999) then compared the results of the different submarket classification methods and determined that in the case of Melbourne the clustering technique was significantly superior to the a priori technique. In other cases clustering only provided a marginal improvement. One possible reason cited for this marginal improvement was the quality of the data used by Bourassa et al. (1999), where the owners estimate of value was used in the hedonic regressions, as opposed to the actual market price. Imprecise locational factors were also cited as possible a reason.

London (Whereabouts London, by Future Cities Catapult)

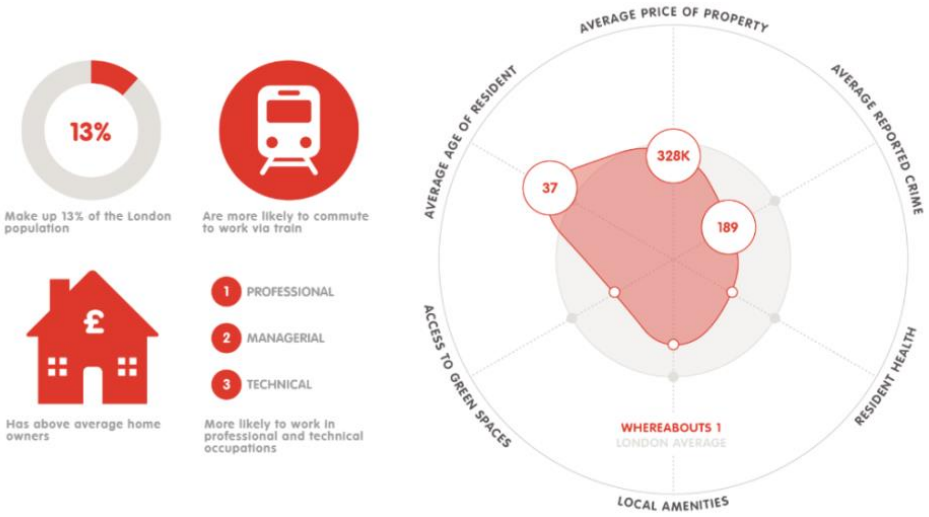
In the Whereabouts London Study by Future Cities Catapult (2014) populations where also grouped into clusters using the K-means algorithm, shown in Figure 11 below.

Figure 11 - Whereabouts London Clustering Map (Bitbucket, 2014)



Unlike Bourassa et al. (1999), populations where not grouped geographically but where grouped based on an array of household characteristics, extracted from data sources including the Food Standards Agency, the Office for National Statistics, Land Registry, OpenStreetMap, Flicker and Transportation for London. This enabled Future Cities Catapult (2014) to build up a picture of what made their local areas both similar to and distinct from each other as seen in Figure 11 on the previous page.

Figure 12 - Notable Characteristics of Whereabouts 1 residents (Future Cities Catapult, 2014)



For example Figure 12 above shows the characteristics of Whereabouts 1 residents who were a professional and well educate population mainly residing in outer London, they make up 13% of Londons population were more likely to commute to work via train and had an above average proportion of home owners.

2.10.2. Multiple Regression and Discriminant Analysis

The difference between Multiple Regression and Multiple Discriminant Analysis is in the nature of the dependent variable, the former deals with variables that are continuous while the latter deals with variables that are categorical (Kort, 1973). A review of the literature did not reveal any examples of multiple regression being applied to segment the housing market, however Kim et al. (2000) segmented the Korean housing marked based on Multiple Discriminant Analysis.

Korea (Kim et al., 2000)

Kim et al. (2000) segmented the Korean housing market using Multiple Discriminant Analysis to classify purchase consumers into three groups; single family housing purchase group, an apartment housing purchase group and a non-purchase group. These groups were classified and predicted using the discriminant function; a linear combination of demographic, socio-economic and residential characteristics. This analysis showed that a household's age, followed by the term of occupancy contributed most to inter-group differences. This research showed, to a satisfactory degree that the discriminant function was accurate in predicting group membership (Kim et al., 2000).

2.10.3. Suitability of Multi-Variable Segmentation Techniques used to date

While the methodologies detailed above may be useful to provide some insight into housing demand, in many cases they do not account for the complex nature of housing demand and apart from Kim et al. (2000), they do not provide a model for predicting variables such as Housing Type, Tenure & Number of Bedrooms. Even though the Kim et al. (2000) research was successful, the approach simplified the entire property market into only three groups.

2.10.4. Towards an appropriate Multi-Variable Segmentation and Predictive Model Technique

The Census information available consists of both continuous and categorical data and it would therefore be useful to have a technique that could seamlessly integrate both types of variables for analysis as well as handle a substantial number of variables and datasets. Even the successful segmentation Multiple Discriminant Analysis technique used by Kim et al. (2000) only dealt with categorical data and did not provide a predictive capabilities.

The field of machine learning offers an alternative methodology that allows shifting the computational onus from the researcher to the computer without necessitating step-by-step inputs from the researcher. This enables high volumes of data to be processed with

the researcher only having to provide a small number of guidelines or parameters. The computer is then able to do the "heavy lifting" in terms of finding the underlying decision boundaries inherent in the data. This data-driven weighting and structuring of the segments is characteristic of a post-hoc segmentation technique introduced in the marketing literature.

2.11. Introduction to Artificial Intelligence and Machine Learning

Michalski et al. (2013, p.3) define learning as a multi-faceted phenomenon that;

"…includes the acquisition of new declarative knowledge, the development of motor and cognitive skills through instruction or practice, the organisation of new knowledge into general, effective representations, and the discovery of new facts through observation and experimentation."

Since the dawn of the computer era, researchers have been trying to equip computers with these capabilities, however, this problem – achieving Artificial Intelligence (AI) - still remains one of the most challenging long-term goals (Michalski et al., 2013). Machine Learning (ML) is the most promising "true-form" AI and the objectives of Machine Learning are focused around three core areas as identified by Michalski et al. (2013);

1. **Task-Oriented Studies** the development and analysis of learning systems to improve the performance of a set of predetermined tasks (this is often referred to as the Engineering Approach),
2. **Cognitive Simulation**, whereby human learning processes are investigated and simulated,
3. **Theoretical Analysis** the theoretical exploration of possible learning methods and algorithms independent of the field of application.

The entire field of AI can be defined using the above set of mutually challenging and supportive objectives which create a cross-pollination of problems and ideas (Michalski et al., 2013). This paper focuses on a subset of "Task-Oriented Studies" which involves development and application of learning systems to real-world problems (and data sets) to determine associations. Initial attempts at AI required the programming of a complete and correct algorithm by specialist personnel to provide the exact "instructions" for the

computer to complete a specific task (Michalski et al., 2013). Machine Learning strives to open the possibilities of instructing computers in new ways to ease the onerous programming burden by shifting the problem solving computational requirements from the user to the computer (Michalski et al., 2013). In other words, Machine Learning seeks to assign the "heavy lifting" to the computer and not the programmer.

Michalski et al. (2013) points out that when approaching a task-oriented study it is critical that the resulting computer system must interact with humans. The requisite "ease of interaction with humans" is the key reason behind the selection of the open source R Statistical Programming Software (R). This will be discussed further in the Methodology section of this paper.

The **Theoretical Analysis** provides a technique for exploring areas of possible learning methods and the Task-Oriented approach provides a means to test and improve the performance of functional learning systems (Michalski et al., 2013). Therefore by constructing and testing available learning systems, the cost-effectiveness, trade-offs and limitations of particular approaches to learning can be determined (Michalski et al., 2013). As Michalski et al. (2013) point out, in this way individual data points within the learning system space are explored and therefore the learning space itself becomes better understood.

As Raschka (2016) points out, the logical way to approach a problem is to apply the simplest modelling technique first and then apply more complex techniques if the model does not produce appropriate predictive capabilities. As shown, simple statistical techniques used to date have failed to yield appropriate predictive results and as such this paper will explore appropriate methods in the field of Machine Learning that may improve on these predictive capabilities.

2.12. Types of Machine Learning

There are broadly three types of Machine Learning Algorithms; Reinforcement Learning, Unsupervised Learning and Supervised Learning.

2.12.1. Reinforcement Learning

In Reinforcement Learning (RL) the machine is trained to make specific decisions (Ray, 2015a). The examples lack target responses, however, positive or negative feedback is given based on the solution the algorithm proposes (Mueller and Massaron, 2017). This is done through the exposure of the machine to an environment where it continually trains itself through trial and error (Ray, 2015a). This is machine learning from past experience and the algorithm attempts to capture the best possible knowledge to make accurate decisions (Ray, 2015a). This technique is often used where the output of the system is a sequence of actions. This sequence of actions is referred to as a policy (Alpaydin, 2014). In these cases a single action is not important but a policy with the correct sequence of actions, that reach a specific goal, is important (Alpaydin, 2014). In these cases, the machine learning algorithm should be capable of assessing the appropriateness of multiple sequences which are then used to generate a useful policy (Alpaydin, 2014). An example of this is a machine learning algorithm being used to play a game of chess. There are a relatively small number of rules in chess, however, there are a relatively large number of possible moves and a move is only good if it is a part of a good sequence of actions – i.e. a winning policy (Alpaydin, 2014).

2.12.2. Unsupervised Learning

In Unsupervised Learning (UL) as in Reinforcement Learning only input data is available and therefore there is no target or outcome to predict (Alpaydin, 2014, Mueller and Massaron, 2017). The aim of unsupervised learning is Density Estimation, i.e. to find the underlying patterns/ structure in the data to ascertain what generally happens and what generally does not (Alpaydin, 2014). One common method of Density Estimation is Clustering where the aim is to find groups or clusters in the input data by grouping, for example, customers based on similar demographic information and historical transactions. This could enable a company to see the demographic profile of customers that consume certain products. This is widely used to segment customers for specific intervention (Ray, 2015a). The same clustering principal can be used in Image Compression (grouping of similar pixels) and Document Clustering (grouping of similar news articles). In the example of customers, these algorithms do not provide insight into

why people choose certain goods or offer a predictive capability, although they can profile what type of people choose a specific good and can be useful to provide new input for supervised machine learning algorithms (Mueller and Massaron, 2017).

2.12.3. Supervised Learning

In Supervised Learning an algorithm learns associated target responses from Training Data in order to later predict the correct response when given new examples (Mueller and Massaron, 2017). In the case of the new example, the target or outcome variable is to be predicted from a set of given independent variables (Ray, 2015a). Using these sets of variables, a function can be generated that maps inputs to desired outputs. The training process continues until the model achieves the desired level of accuracy on the Training Data (Ray, 2015a). The type of target variables determines the type of problem, i.e. a qualitative target variable will mean a classification problem, while a quantitative target variable will be a regression problem (Mueller and Massaron, 2017). Examples of Supervised Learning Algorithms (SLA's) are Decision Tree (Classification & Regression) and Random Forest (Classification & Regression).

2.13. Suitable Machine Learning Algorithms

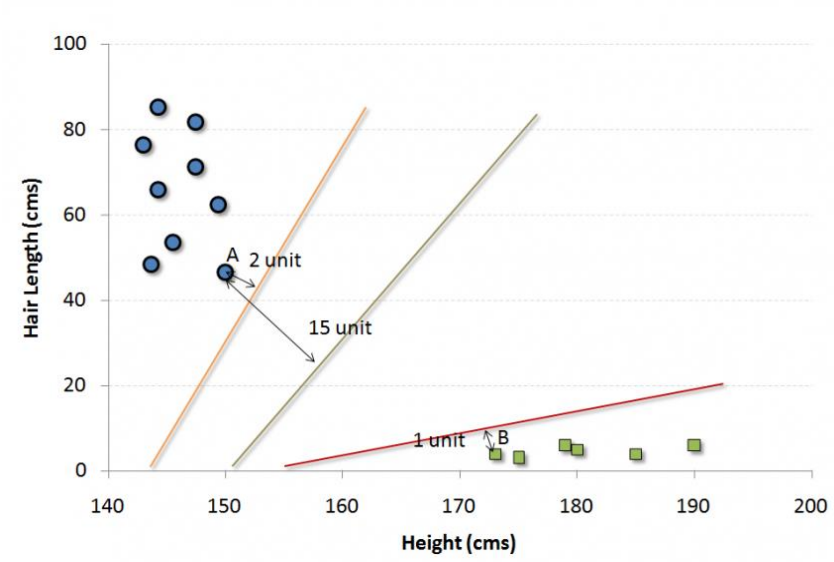
The core nature of Supervised Learning Algorithms means that this type of algorithm would be most suitable to create the predictive models. The suitable Supervised Machine Learning algorithms within the context of this paper are introduced below. The selection and suitability of the algorithm used in this paper is clarified in the sections that follow.

2.13.1. Support Vector Machine (SVM)

SVM is a classification method in which each data item is plotted in n-dimensional space, where n is the number of features that are present, with the value of each feature being the value of a particular co-ordinate (Ray, 2015a). The algorithm then seeks to find the line such that the distances from the closest point in each of the two groups will be farthest away (Ray, 2015a). Figure 13 below shows a two feature plot in two dimensional space, the two sets of data are best split by the black line as this is furthest from the

closest point in each set of data. This line becomes the classifier and new data is classified based on which side of the line the test data was on (Ray, 2015a).

Figure 13 - Two Feature SVM plot (Ray, 2015a).



While SVM's are useful, they require substantial computing power when using more than three variables. Limited computational power was available in the form of a single personal computer with an Intel® Core™ i7 CPU 950 @ 3.07GHz and 8GB RAM.

2.13.2. Decision Tree

Decision Tree is a type of supervised learning algorithm that works for both categorical and continuous dependent variables. This algorithm splits the population into two or more homogeneous sets based on the most significant attributes first to make as distinct groups as possible (Ray, 2015a).

2.13.3. Random Forest

Random Forest is a trademarked term for an ensemble of Decision Trees. The Random Forest Algorithm uses a collection of Decision Trees to classify a new object based on the attributes (or features). Each tree provides a vote for a particular classification and as a collective, the forest chooses the classification of the new object based on the majority vote (Breiman, 2001, Ray, 2015a).

2.14. Selecting an appropriate Machine Learning Algorithm to address the research questions

Decision trees are non-parametric supervised learning models and as such can incorporate both numerical and categorical data as well as data with diverse response distributions (Breiman et al., 1984, Clark and Pregibon, 1992, Ripley, 1996). There are many other advantages of tree-based methods including; a low level of data preparation, computational efficiency, multi-class problem solving, a high degree of stability and accuracy, limited over-fitting, concurrent selection and classification of predictors, managing noise in the data and the production of results that are simple to understand (Breiman et al., 1984, Clark and Pregibon, 1992, Ripley, 1996). The goal would be to produce a model through decision-tree-based machine learning that would be able to use the underlying features within the data to learn decision rules and use this to predict the outcome of a target variable. Single household attribute segmentation methods and the multivariate clustering techniques have failed to provide and in-depth understanding and the predictive capabilities sought in this paper. Tree-based methods perform better than alternatives where the data contains non-linear decision boundaries, as is expected to be case in the Census data. The focus of this paper will be to show that Random Forest is a useful tool for segmenting housing demand to produce suitable predictive models for Housing Type, Tenure and Number of Bedrooms. This tool will be applied with an affordability and structural approach.

2.15. Exploring Decision Trees

There are two types of decision trees, Classification Trees and Regression Trees and together these are generally referred to as "CART" i.e. "Classification and Regression Trees" (Breiman et al., 1984, Ripley, 1996, Liaw and Wiener, 2002, Eremenko and de Ponteves, 2017). **Classification Trees** are used with categorical data (for example, gender of household head which could be male or female) and are used to classify this data into classes. **Regression Trees** are designed to assist in the prediction of outcomes which are continuous and can be related directly to an actual number, for example, the salary of a person (Eremenko and de Ponteves, 2017).

2.15.1. Classification Decision Tree

A Classification Decision Tree works in the following way; the tree will have a node that represents each split, the algorithm used will search for the optimal split that maximises the difference between the types of data and minimises the information entropy[4] (Breiman et al., 1984, Clark and Pregibon, 1992, Ripley, 1996, De'ath and Fabricius, 2000, Breiman, 2001). The second split in the tree follows the same principal as the first, as will the n^{th} at which point all the data will have been processed (Eremenko and de Ponteves, 2017). In other words, tests are run for each class sequentially, and the model systematically works through each different level (or class) of the tree until a leaf node, or decision is reached (Breiman et al., 1984, Breiman, 2001). In a Decision Tree the four most commonly used criteria for splitting the nodes are the Information Gain, Chi-Square, Gini Index and Variance (Raileanu and Stoffel, 2004). The first three are used for categorical data and the fourth is used for continuous data. An example of how this would work with the census data would be as follows; an independent variable such as income would be initially selected by the model and each income category would represent a different branch[5] (McGaffin and Kirova, 2016). Then the next independent variable, for example, age would be integrated to create additional branches and so on until all the variables have been applied by the model (McGaffin and Kirova, 2016). The process is designed so that for each decision path, or branch string, the dominant Dependent Variable Class associated with that decision path would be identified (McGaffin and Kirova, 2016). A simplified example of this process is shown in Figure 14 and Figure 15 below.

[4] Informational entropy in a Decision Tree refers to the homogeneity of the split at each node, i.e. minimising informational entropy at each split will maximise the homogeneity of the split and the information gain due to the split

[5] The analogy of each category representing a different branch is simplified for ease of understanding. Strictly speaking as most tree-based methods are binary, multiple levels of branches would be required to explain categories with more than two classes. For example a category with four classes A,B,C&D, may be split A and not A first, then not A would be split into B and not B and so on until all classes were captured

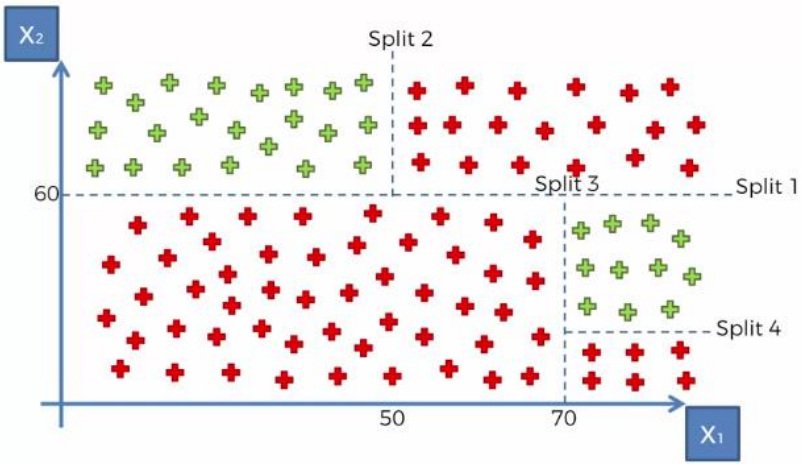
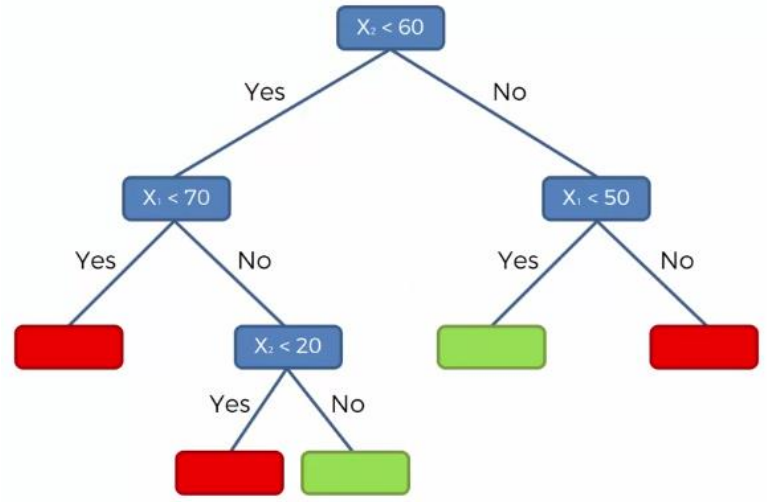


Figure 14 - Classification Decision Tree - Cartesian Plane (Eremenko and de Ponteves, 2017)

In the Figure 14 above, each split will represent a node and each "ring-fenced" part of the data will represent a terminal leaf. Each split is shown in Figure 15 below, this shows that if X2 is less than 60 and X1 is less than 50 the data is green.

Figure 15 - Classification Decision Tree – Tree structure (Eremenko and de Ponteves, 2017)

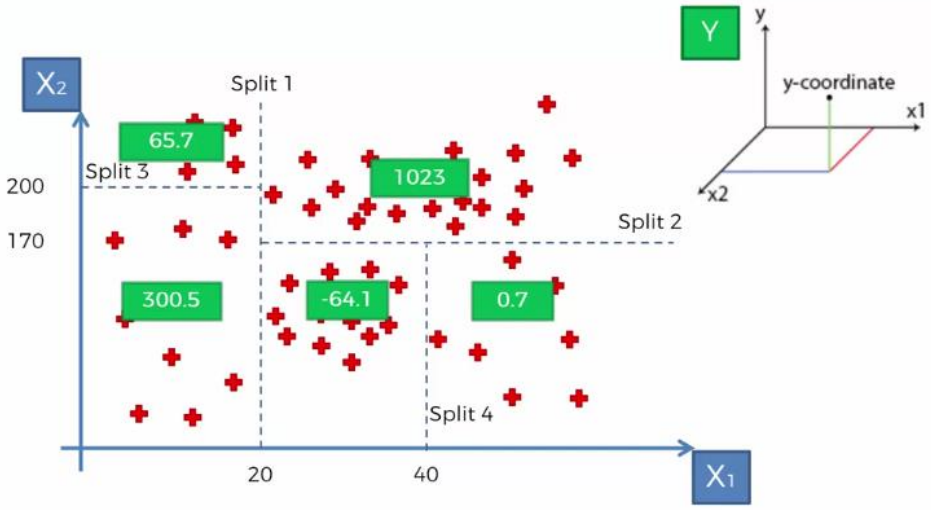


2.15.2. Regression Decision Tree

Much like Classification Decision Trees, the Regression Decision Tree algorithm will split up the two dimensional data points into different sectors by minimising informational entropy (Eremenko and de Ponteves, 2017). Looking at Figure 16 below, two independent variables X1 and X2 are shown in two dimensional space. These sectors are similar to the Classification Decision Trees except instead of having a Class Label, these have a numerical value. The algorithm will cease to create new sectors if there is no information to be gained by completing an additional split (leaf node).

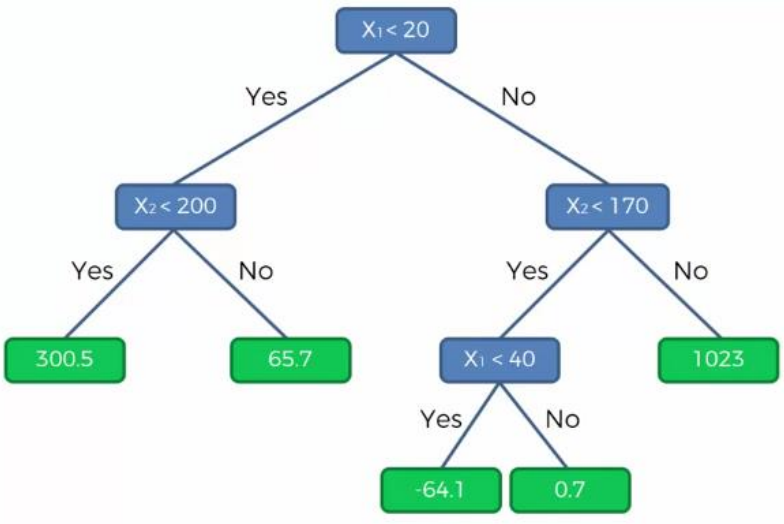
Without machine learning analysis a researcher would have to take the average of all the data points, which would provide a poor prediction of the dependent variable (Eremenko and de Ponteves, 2017). Using this machine learning technique to minimise entropy and maximise information gain the researcher can obtain data segments that will provide a high degree of accuracy when predicting the dependent variable given the values of X1 and X2 (Eremenko and de Ponteves, 2017).

Figure 16 - Regression Decision Tree - Cartesian Plane (Eremenko and de Ponteves, 2017)



The decision tree for the data shown in Figure 16 above is shown in Figure 17 below. The numeric output is held within the tree itself which is why this is referred to a Regression Decision Tree (Eremenko and de Ponteves, 2017).

Figure 17 - Regression Decision Tree – Tree structure (Eremenko and de Ponteves, 2017)



2.16. Ensemble Methods and Random Forest

Ensemble methods train multiple learner algorithms to solve the same problem (Breiman, 2001, Zhou, 2012). This can be contrasted against ordinary machine learning approaches where a single learner is constructed from the Training Data (Breiman, 2001, Zhou, 2012). Ensemble learning constructs a set of learners and combines them to learn a set of hypotheses in order to better predict the dependent variable given new data (Breiman, 2001, Zhou, 2012). Ensemble Learning is also referred to as committee-based learning or learning multiple classifier systems (Breiman, 2001, Zhou, 2012). Tree based methods work well for multi-class problems as "Random Forests" can be created by building multiple decision trees - these are then used together to predict a target variable (Breiman, 2001, Liaw and Wiener, 2002). Random Forest is a type of Ensemble Learning. Random Forest as opposed to a single decision tree helps to reduce over-fitting and to make the prediction more precise. Random Forest was used by Microsoft in the Kinect together with an infrared grid to understand body movements and interpret these to control a gaming console (Shotton et al., 2013). This method enabled a reduction in processing power required to achieve a given level of accuracy and consequently enabled a reduction in the costs of hardware for the Microsoft Kinect consoles (Shotton et al., 2013).

2.16.1. Random Forest

The methodology explained in Eremenko and de Ponteves (2017) is shown in Table 4 below. First a bootstrap sample of the training set is selected at random from the training set, i.e. k data points. Then a decision tree is built for this bootstrap sample of the data, i.e. for the k data points. The required number of trees is chosen and the steps indicated below are repeated to create each new decision tree. For a new data point, each one of the N trees predicts the value of the dependent variable for the data point in question and assigns the new data point the average across all the predicted dependent variable values in the case of regression problems or a label based on majority vote in the case of classification problems (Breiman, 2001, Liaw and Wiener, 2002). While a change in one part of the data could greatly affect the results of a single tree, in Random Forest,

the tree will now have to 'compete' with all the other trees in the forest and therefore the single tree will not have as much impact on the predictive value (Eremenko and de Ponteves, 2017).

Step	Random Forest (Classification Trees)	Random Forest (Regression Trees)
1	Select k data points by sampling from the Training Data Set uniformly with replacement	
2	Build the Decision Tree associated with these k data points. At each node select a random set of $r = p^{0.5}$ predictors and then split using the best predictor among the r available instead of the best among the full p	
3	Repeat steps 1 & 2 until the required number of trees, say, N, have been constructed	
4	For each new data point, obtain the **category prediction** from each of the N trees in the forest, the new data point will be assigned the **category which receives the majority vote**.	For each new data point, obtain the **dependent variable value prediction** from each of the N trees in the forest, the new data point will be assigned the **average of all the predicted dependent variable values.**

Table 4 – Random Forest Classification & Regression Methodology (Eremenko and de Ponteves, 2017)

In comparison to other machine learning algorithms, Random Forests are one of the simplest approaches as there are only two "hyper-parameters" to tune. As a general rule the more trees in the Random Forest the better (Raschka, 2016). SVM on the other hand has a large number of hyper-parameters that need tuning, for example; choosing the right kernel, regularisation penalties, slack variable etc. Random Forests are non-parametric models, i.e. the complexity increases with an increase in the number of training samples. This means they can become computationally expensive with a large number of trees, when compared with a generalised linear model for example (Raschka, 2016). Random Forests are simple to train and can handle multi-class problems without any additional work as in the case of other algorithms such as SVM (Raschka, 2016).

2.17. Conclusion

An overview of the Cape Town housing problem and current supply and demand metrics gives context to the study. Consumer choice theory frames the theory of housing demand. It was established that little work has been completed on the segmentation of

market demand from a structural point of view. Market Segmentation techniques used to date have been inadequate to sufficiently explain how household's make decisions to choose certain types of housing. More specifically studies have over simplified housing demand and have not allowed for the nuanced understanding of demand that is required to address the current Cape Town housing problem, nor have any of the methods applied suitably provided a predictive model for housing Type, Tenure and Number of Bedrooms. The literature clearly shows that the Supervised Learning Algorithm Random Forest is an appropriate methodology to segment housing demand to create a model capable of predicting selected dependent variables given unseen household characteristics. This is due to the robustness of the tool when dealing with different types of data, multi-class problems and non-linear decision boundaries - all characteristics that are present in housing demand. The focus of this paper is therefore to apply a Random Forest model to segment housing demand, with an affordability and structural approach, to produce models capable of predicting housing Type, Tenure and Number of Bedrooms.

3. METHODOLOGY

3.1. Introduction

The chapter starts by defining knowledge which is followed by the requirements of research. The research paradigm is then outlined and the theoretical framework of the research is explored. A baseline conceptual framework is introduced before being further developed in this paper. The research strategy is then outlined, followed by the research design.

3.2. Definition of Knowledge

The historical development of the word science comes from the Latin word *scientia* which means knowledge and Bhattacherjee (2012) defines science as a systematic and organised body of knowledge that is acquired through "the scientific method". The scientific method is a standardised set of techniques that are used to build scientific knowledge, these include principals used to make valid observations, correctly interpret results as well as generalise those results (Bhattacherjee, 2012).

Science can be organised into two broad categories; Natural Science and Social Science (Bhattacherjee, 2012). **Natural Science** is the science of natural phenomena or objects, while **Social Science** is the science dealing with people or groups (Bhattacherjee, 2012). The goal of scientific research is to identify laws and propose theories that explain natural and social phenomena (Bhattacherjee, 2012).

Laws and theories are arrived at through logical processes (theory) and evidence (observations) which together contribute to the development of scientific knowledge (Bhattacherjee, 2012). Scientific inquiry may be conducted by using either Inductive or Deductive research. The goal in **Inductive Research** is to use observed patterns in empirical data to develop theoretical concepts while the goal in **Deductive Research** is to test these theoretical concepts and patterns through the use of empirical data (Bhattacherjee, 2012). For research to be accepted as scientific knowledge it must satisfy certain research requirements.

3.3. Research Requirements

Research is required to follow the scientific method and must satisfy four characteristics; Replicability, Precision, Falsifiability and Parsimony (Bhattacherjee, 2012). Replicability requires that independent replication would obtain similar, if not identical, results. Precision requires that theoretical concepts that are often hard to measure are well defined for independent use and testing. Falsifiability requires that a theory be stated in a way that makes it possible to be disproven. Parsimony (or Occam's Razor) requires that when there are multiple justifications, scientists should accept the simplest or most logically economic justifications (Bhattacherjee, 2012).

To produce successful research suitable approach, design, data collection and analysis methods need to be identified and used in conjunction with the correct protocols to ensure reliability and validity (Bhattacherjee, 2012). Reliability refers to the repeatability of a study, while Validity can be further broken down into Internal Validity, Construct Validity and External Validity. Internal Validity relates to whether the correct concept or variables have been identified, Construct Validity relates to whether the variable represents the concept and External Validity relates to whether the findings can be generalised. In order to achieve this, the paradigm/theory/approach needs to identify the correct variable and the variable needs to suitably represent the concept. The design needs to ensure that the research is repeatable and generalisable. Before all these concepts can be explored, the paradigm of thinking within which the research will be conducted needs to be established.

3.4. Research Paradigm

The origin of the term paradigm stems from Kunh (1962) in his book called *the structure of scientific revolutions*. At the time of the second edition of this book in 1970, a paradigm had become a fully-fledged idea, most notably in the field of natural sciences. The term entered the social sciences and had a major impact on the development of philosophy and methodology in the field, being adopted by many researchers and authors (Vosloo, 2014).

A paradigm can be defined as a system of thinking or belief system which we use to organise our reasoning and observations (Vosloo, 2014, Bhattacherjee, 2012). This concept is centred on the fact that different people can view the same social reality in various ways. This also applies to researchers and the way they view and study social phenomena and Burrell and Morgan (1979) indicate that the two key assumptions that researchers make about the world can be categorised as; Ontological and Epistemological. **Ontological** refers to our interpretation of the world, for example does it consist predominantly of social order or perpetual change (Bhattacherjee, 2012). **Epistemological** relates to how best to study the world, e.g. should we study social reality with a subjective or objective approach (Bhattacherjee, 2012). These two sets of assumptions can be used to classify social science research into four paradigm categories.

	Objectivism	**Subjectivism**
Radical Change	Radical Structuralism	Radical Humanism
Status Quo (Social Order)	Functionalism	Interpretivism

Table 5 –Research paradigm categories (Bhattacherjee, 2012).

Positivism and post-positivism are two paradigms that are often used by social science researchers. **Positivism** holds that science or the creation of knowledge should rely only on theories that can be tested directly and phenomena that can be observed and measured, while **Post-Positivism** argues that through the use of both logical reasoning and empirical observations a researcher can draw rational conclusions about phenomena (Bhattacherjee, 2012). Post-positivists view science as probabilistic and not certain as the positivists do (Bhattacherjee, 2012). Post-positivists can be further divided into subjectivists, who argue that people view the world differently depending on their own subjective interpretation or critical realists who argue that despite the subjective nature of our minds an independent external reality exists (Bhattacherjee, 2012). Hypothetical-deductive generalisations are tested using quantitative methods in the post-positivist and positivist paradigms (Amaratunga et al., 2002). This paper is focused on the social sciences and takes the form of inductive research as the goal of the research is to infer theoretical concepts from patterns in the Census Data. A post-positivist research paradigm is adopted as the research identifies empirical and measureable patterns used

to predict household behaviour. This paradigm was then used to select the body of theoretical knowledge which is discussed below.

3.5. Theoretical Framework

Theories are generated in an effort to explain, understand and predict phenomena, and in many cases, to challenge and expand existing knowledge within the limits of critical boundary assumptions (Abend, 2008). The theory that explains why the research problem put forward by a study exists is introduced and described by using a theoretical framework (Swanson and Chermack, 2013).

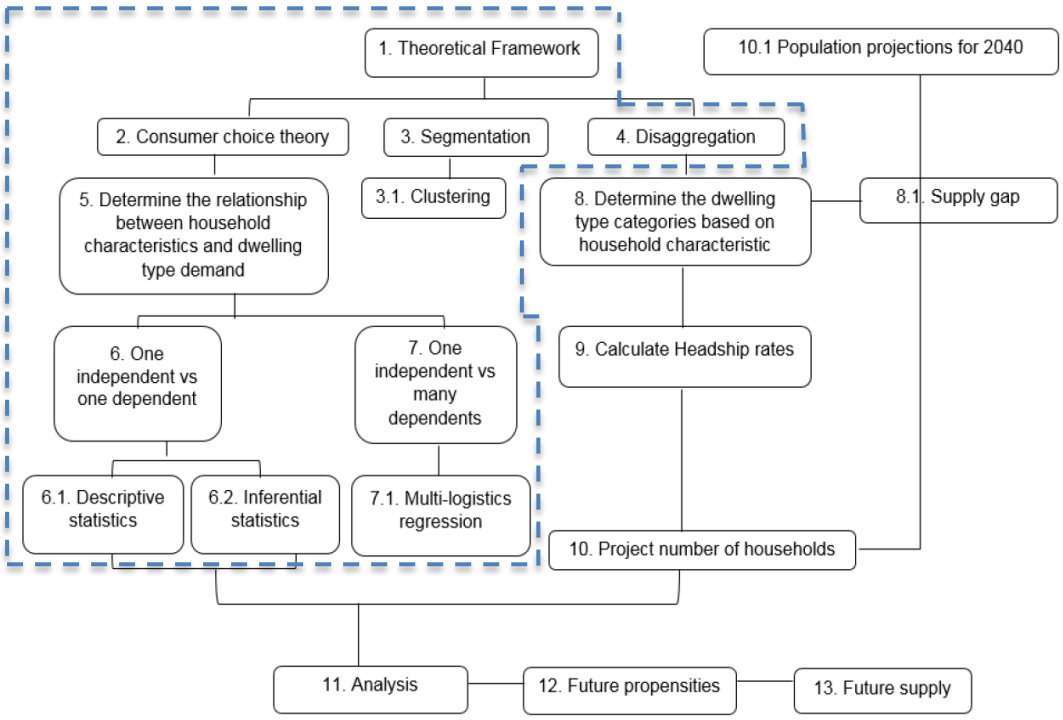
Housing demand theory is one of the key theories that are used in this research, and housing demand theory is closely related to and has its roots in consumer choice theory. The theory of demand informs how household's make decisions and select different dwelling types within the imposed theoretical constraints (Ndziba et al., 2015). The theory of probability is discussed and marketing theory on segmentation is looked at due to its bearing on how different groups of consumers with the same set of preferences have a similar willingness to consume similar products. Machine learning theory is explored, focusing on Random Forest as a segmentation and prediction technique. The relationship between the variables identified in the theory is described in the conceptual framework below.

3.6. Conceptual Framework

The conceptual framework put forward by Ndziba et al. (2015) shown in Figure 18 on the following page is used as a basis for the development of a more holistic and generalised conceptual framework. Ndziba et al. (2015) use a conceptual process framework based on the Hogarth (2015) method for determining housing affordability. While this framework is useful for a specific and integrated segmentation, disaggregation and projection process, it does not provide a comprehensive understanding of the relationships between each theoretical concept.

This paper aims to further define the conceptual framework within the highlighted region of the Ndziba et al. (2015) framework. The projection of housing stock and household's is beyond the scope of this paper.

Figure 18 - Base Conceptual framework (Ndziba et al., 2015)



The conceptual framework shown in Figure 19 is the framework developed in this paper. Each major concept area is listed along the left hand side of the figure and each relevant concept within that concept area is shown in the roadmap. The relationship between each concept is shown with arrows. While this paper introduces most of the concepts below, the focus of the paper is on the concepts highlighted in green.

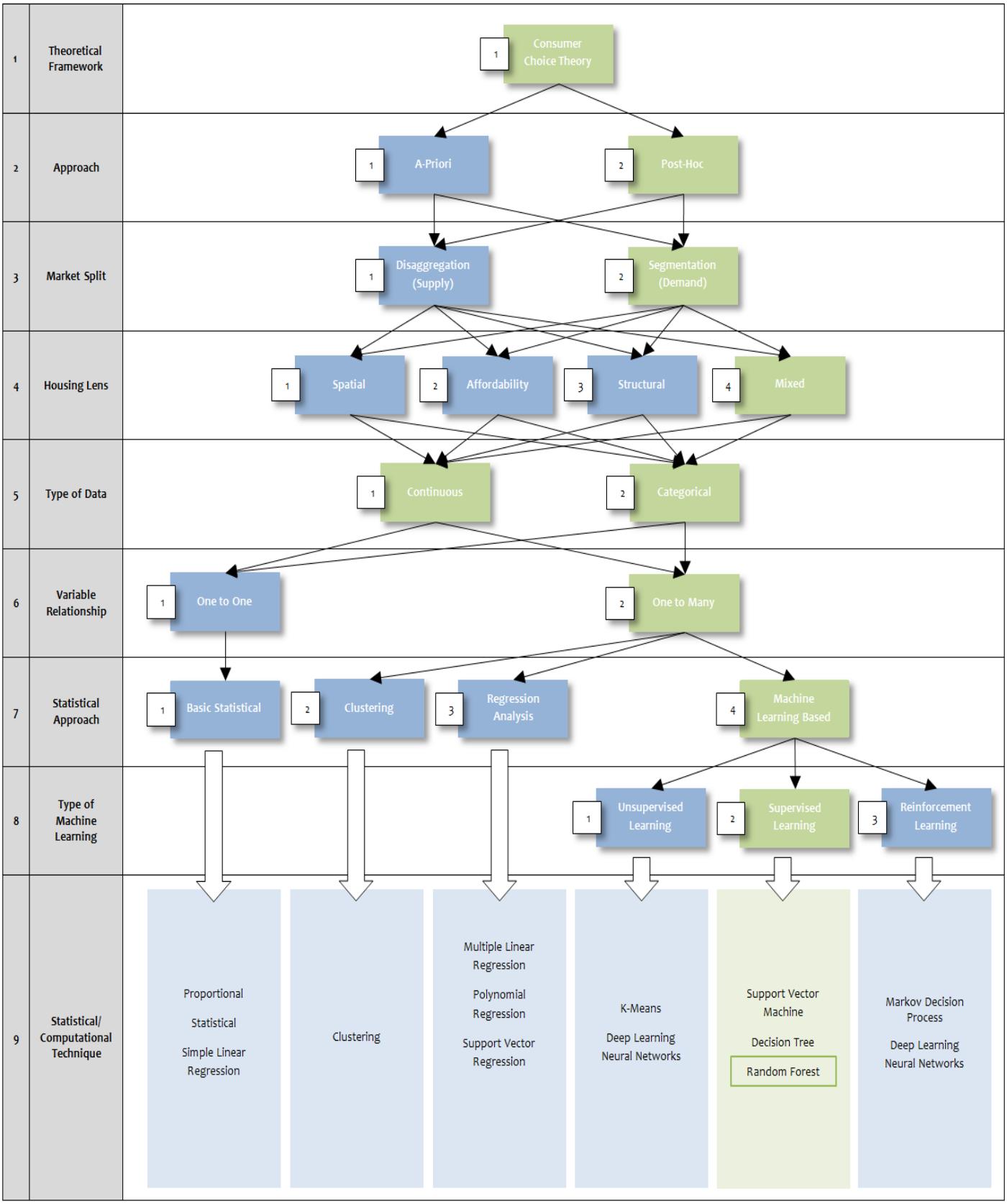
The conceptual framework shown in Figure 19 starts with the theoretical framework of consumer choice theory. The segmentation approach should then be selected and these were identified in the conceptual framework as either an a priori or a post-hoc approach. This paper uses a post-hoc approach as the segmentation is data driven. The Market then needs to be split either through Disaggregation (Supply Side) or Segmentation (Demand Side) and this paper focuses on Segmentation only. For each market split operation, a housing lens needs to be determined; spatial, affordability, structural or any combination

of these. This paper focuses on a mixed approach using the affordability and structural lens and will use the household characteristics found in the census data to segment the demand. The type of data then needs to be determined and this will determine the statistical approach. While it is not shown in this conceptual framework due to the complexity it would bring, certain statistical techniques cannot handle specific types of data. As this paper deals with both data types, a statistical technique capable of dealing with both data types needed to be selected.

Next, the variable relationship is to be selected which also sheds light on which statistical techniques should be used; one-to-one relationships call for basis statistical techniques while one-to-many relationships call for clustering, regression or machine learning based techniques. The variable relationship in this paper is a one-to-many (described in Chapter 2) and the statistical approach is machine learning based. There are three general types of machine learning, unsupervised learning, supervised learning and reinforcement learning. The focus of this paper is on the Random Forest Statistical/ Computational Technique in with the area of Supervised Learning. The adapted conceptual framework discussed above is shown in Figure 19.

Thus far the context and theoretical requirements of the research has been explored in terms of the research paradigm as well as the theoretical and conceptual framework within which the research is conducted. The next sections clarify how the practical requirements of the research are met, starting with the requirement of a suitable approach.

Figure 19 - Conceptual framework adapted from the Ndziba et al. (2015) framework



3.7. Research Approach

There are broadly three different research approaches used in the built environment, **Quantitative**, **Qualitative** and **Mixed Methods** (Amaratunga et al., 2002). These research strategies can be viewed as the procedural framework within which research is completed. The strategy chosen depends on the context of the research as well as its nature and purpose (Thomas, 2010).

3.7.1. Qualitative Research

Qualitative research is conducted through a prolonged and/or intense interaction with the field or a real-life situation (Amaratunga et al., 2002). This type of research is concerned with assessing the opinions, attitudes and behaviours of people in real-life situations. Qualitative research is non-numerical, descriptive and explanatory in nature and aims to describe a situation, find meaning or feeling – it investigates the why and how of decision making (Rajasekar et al., 2006). The strengths and weaknesses of Qualitative Research as identified by Amaratunga et al. (2002) are listed below;

Strengths

- Data-gathering methods seen more as natural than artificial,
- Ability to look at change processes over time,
- Ability to understand people's meaning,
- Ability to adjust to new issues and ideas as they emerge,
- Contribute to theory generation.

Weaknesses

- Data collection can be tedious and require more resources,
- Analysis and interpretation of data may be more difficult,
- Harder to control the pace, progress and end-points of research process,
- Policy makers may give low credibility[6] to results from the qualitative approach.

[6] Credibility by policy makers is generally determined by high levels of validity (external) and reliability, allowing for the results of the research to be generalisable and repeatable respectively

Due to the nature of the problem being investigated a qualitative research strategy will not be used, this research adopts a quantitative research strategy.

3.7.2. Quantitative Research

The Quantitative research strategy is characterised by the assumption that human behaviour can be explained through the deductive logic of the natural sciences (Horna, 1994). Quantitative philosophy can be generally defined as an extreme form of empiricism through which theories are applied to acquired facts and are justified only to the extent to which they can be verified (Chalmers, 2013). Quantitative investigations look for distinguishing characteristics, properties and empirical boundaries and tend to measure "how much" or "how often" (Nau, 1995). Quantitative research aims to measure decision making in terms of the "what", "where" and "when" (Rajasekar et al., 2006). In Quantitative Research, statistics is the most commonly used branch of mathematics and is used in an array of disciplines including the physical and social sciences as well as biology and economics (Rajasekar et al., 2006). The strengths and weaknesses of Quantitative Research as identified by Amaratunga et al. (2002) are listed below;

Strengths

- They can provide a wide coverage of the range of situations,
- They can be fast and economical,
- Where statistics are aggregated from large samples, they may be of considerable relevance to policy decisions.

Weaknesses

- The methods used tend to be inflexible and artificial,
- They are not very effective in understanding processes or the significance people attach to actions,
- They are not very helpful in generating theories,
- Because they focus on what is, or what has been recently, they make it hard for policy makers to infer what changes and actions should take place in the future.

The collection of data for quantitative research is usually conducted through surveys, questionnaires or scientific monitoring and recorded in either numerical or categorical format for analysis through statistical methods. Simple statistical methods to express one-to-one relationships between variables include correlations and relative frequencies. More complex one-to-many relationships require more complex statistical analysis to draw conclusions, such as multiple regression or machine learning algorithms.

3.7.3. Mixed Methods

Mixed method research is conducted when the researcher mixes both qualitative and quantitative methods. It is key to note that this mixed method, much like either of the qualitative and quantitative methods, would require that the researcher select the method prior to research with due consideration for its advantages in relation to the type of research being conducted. Within the research community, qualitative and quantitative methods can be considered complimentary Amaratunga et al. (2002). The prominence of mixed methods has grown special attention to the key strength of mixed methods, the concept of triangulation (Yin, 1994). The key premise of triangulation is that the weaknesses of each method individually will be counter-balanced by the strengths of the other (Amaratunga et al., 2002). While mixed methods can be advantageous when studying phenomena,

The strengths and weaknesses of mixed methods are identified by Amaratunga et al. (2002) as follows:

Strengths

- Triangulation, i.e. the use of multiple methods, data sources etc. to examine the same phenomenon,
- The researcher can structure the research in a way that draws on the advantages of both Qualitative and Quantitative methods, however if this is not done correctly it can be a weakness,
- Can provide better context, assist to explain findings or causal processes.

Weaknesses

- Collecting both quantitative and qualitative data is more time-consuming and may be more resource intensive,
- The research design can be very complex,
- People's cognitive abilities (and paradigms) predispose them to better interpret either quantitative or qualitative research.

While mixed method research can be very powerful, it was not seen to be suitable for this research, primarily due to the scale of the research conducted in this paper. The time and resources available did not allow for a mixed methods research design to be adopted. After the research method is decided a compatible research design must be selected.

3.8. Research Design

The nature of the research determined the research design. Grounded Theory was rejected as a research design as it would not generate the quantitative data required for analysis. While extensive Field Surveys could be useful, unfortunately these were not practical in the context of this research and this research design was therefore rejected. Focus Groups, Case research and Ethnography were similarly rejected as their focus was too narrow and would require too much time to develop the data required to build a robust Machine Learning model. Experimental Studies and Action Research were rejected as research designs as these would involve policy interventions over which the researcher has no control.

Secondary data analysis was determined to be the most suitable. This research design involves the analysis of data collected by external parties such as other researchers, government agencies or private entities. This is an effective research design where the collection of primary data is impractical and/ or too costly and where the secondary data is already available at an appropriate level to address the researchers questions (Bhattacherjee, 2012).

Secondary data analysis was chosen for two main reasons; firstly none of the research designs mentioned above are suitable for practically collecting the necessary data for the

study and secondly as outlined in the literature, this research design has been used by other researchers in Ottawa, Tshwane and Cape Town.

3.8.1. Secondary Data Analysis

Secondary data analysis introduced in the above section can be described as pre-existing data that is analysed by the researcher. Most other research designs require the researcher to collect the primary data, however, in secondary data analysis the data has already been collected and organised by another party (Bhattacherjee, 2012).

Advantages

Secondary data analysis offers immediate access to data that could have been time consuming and expensive to procure. Similar data is often collected over multiple periods of time which both refines the collection of the data and allows for cross-historical comparison of the data (Koziol and Arthur, 2011). The data is generally procured by government agencies which given the often rigorous data collection requirements of these agencies, may lead to data of a high quality being collected. These government funded studies often contain samples that are larger and more representative of the population as a whole and thus have better external validity (Koziol and Arthur, 2011). Datasets often contain considerable breadth (many variables) and the oversampling of low prevalence groups allows for increased statistical precision (Koziol and Arthur, 2011).

Disadvantages

The disadvantage of Secondary Data Analysis stems from the fact that the data collection has already been completed and therefore the researcher using the data cannot give any input regarding the collection of the data. Given the breadth of the data, the data may potentially lack depth in any one particular element. Validity and relatability problems could emerge due to elements only being defined by a single survey item or subset of items (Koziol and Arthur, 2011). Post-hoc model construction may be unsuccessful due to disassociation of survey items (Koziol and Arthur, 2011). Less value could be attributed to

studies using secondary data analysis by certain departments or disciplines (Koziol and Arthur, 2011). Interpretation of the data may require knowledge of survey statistics or methods (Koziol and Arthur, 2011). The collection of the data will need to have been conducted in a systematic and scientific way for the data and thus the outputs of the analysis to be valid (Bhattacherjee, 2012).

3.8.2. Census Data Collection

Census 2011 was the third census completed by Statistics South Africa since South Africa became a democracy in 1994. The census was commissioned by the South African government to assist with development programmes and to facilitate informed decision-making as well as policy formation, implementation and evaluation (Statistics South Africa, 2012).

The collection of the data for Census 2011 was conducted between the 9th and the 31st October 2011. Previously the census was completed every five years by Statistics SA, however, resource constraints meant that this was changed to every ten years (Statistics South Africa, 2012). The data collectors visited all household's in South Africa to complete a questionnaire dealing with an array of information relating to age, income, housing, employment, religion, race, gender and language – the full Census 2011 questionnaire can be found in Annexure A at the end of this paper. The research is conducted from a structural and affordability point of view and therefore these household characteristics are important - the exact household characteristics (independent variables) used in this study are detailed in Chapter 4. Similarly to understand the demand for housing types, tenure and number of bedrooms data on these dependent variables is required. In the Census data, the dwelling types can be simplified as follows; Single-detached, Traditional, Flat, Cluster/ Townhouse, Semi-detached, Second Dwelling, Backyard Informal, Informal Settlement and Other (Hogarth, 2015). The Census questionnaire provides the following options for Tenure; rented, owned but not yet paid off, occupied rent-free, owned and fully paid off and other. The Census Questionnaire allows the following options for number of Bedrooms; 0, 1, 2, 3, 4 or 5+.

In order to use the data collected by the census in research, the researcher needs to trust that the data is accurate and representative of the population. The primary limitation in using the census data stems from the accuracy of the data itself. Every effort is made by Stats SA to ensure the accuracy of the data, for example the design of the questionnaire is concise and the interview with the respondent is kept to less than eighteen minutes. The honesty of the respondent and the due diligence of the interviewer are two factors that can greatly affect the usefulness of the data but are difficult to control and improve (Statistics South Africa, 2012). With that being said, the South African Census conducted by Statistics SA is a well-recognised data-set and has been used in major studies in a broad array of fields.

3.8.3. Data Processing (Introduction to R and R Studio)

R Statistical Programming Software[7] and R Studio[8] were used for the data processing and construction of the Random Forest Predictive Models. Both these programs can be downloaded and used for free. R is described by its developers as;

"A language and environment for statistical computing and graphics. It provides a wide variety of statistical (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering, etc.) and graphical techniques, and is highly extensible".

RStudio is an interface and set of integrated tools to improve ease of use and productivity with R. The package includes a console, syntax-highlighting editor that supports direct execution of code as well as a selection of tools for plotting, debugging, viewing history and managing the workspace.

3.8.4. Data Analysis

Analysis of the results for each model was completed by assessing the overall accuracy of the model relative to the no information rate[9]. The Kappa statistic was also analysed

[7] R can be downloaded from https://www.r-project.org/
[8] R Studio can be downloaded from https://www.rstudio.com/products/rstudio/download/
[9] The no information rate is the percentage proportion of the most common class

against interpretations found in the literature. These two measures allowed for the selection of top-performing sets of models for each of the three dependent variables Housing Type, Tenure and Number of Bedrooms. A confusion matrix was generated for each of the sets of selected models and the class accuracies were analysed, this allowed analysis of which classes the models excelled at predicting and which classes the models struggled to predict correctly.

3.8.5. Conclusion

The research sought to identify empirical and measureable patterns in the data as well as better understand household behaviour and therefore a post-positivist research paradigm was adopted. The theoretical framework of this research was housing demand theory with a focus on Random Forest as a segmentation and prediction technique to create predictive models for the three selected dependent variables; Housing Type, Tenure and Number of Bedrooms. This paper builds on the Ndziba et al. (2015) conceptual framework and presents a more holistic view of the framework. A quantitative research approach was adopted and secondary data analysis was determined to be the most suitable research design. The collection of the Census data was shown to be conducted in a robust and reliable manner. A broad overview to data processing and data analysis techniques was given and these are further explained in the Chapter below.

4. DATA PROCESSING AND ANALYSIS

4.1. Introduction

The chapter begins with an introduction to data processing before detailing how the Census 2011 data was accessed and converted into the CSV format which was used to store and process the data. Data Pre-Processing Steps are explained before the chapter outlines the roadmap used for data processing and the construction of the Random Forest Models. The chapter then presents the results of the models created for each dependent variable, followed by an in-depth analysis of the best performing set of models selected for each dependent variable. The chapter is then concluded.

4.2. Data Pre-Processing

Data pre-processing was completed using Microsoft WordPad, Microsoft Excel, R Statistical Programming Software and R Studio.

4.2.1. Accessing the Census Data and Conversion to CSV

STATS SA only provides the Census 2011 Data in SuperCROSS format (with file extension SXV4) however, UCT's DataFirst[10] website enabled the Census 2011 10% sample to be downloaded in an SPSS file format. While an SPSS file format can be used in R there were various checks that were easier to complete in MS WordPad and MS Excel therefore the data was converted to a Comma Separated file (with file extension CSV). This file type is compatible with both MS WordPad and MS Excel. The steps listed below where used to complete this conversion. The full R coding to complete each step is contained in Annexure C under the relevant step.

STEP A – Setting the Working Directory in R

Setting up the working directory made importing, exporting and using datasets more efficient (see Annexure C for R Script).

[10] To access the data used from UCT's DataFirst Website, create an account and use the link below; https://www.datafirst.uct.ac.za/dataportal/index.php/catalog/485/get_microdata

STEP B – Converting Data from SPSS format to CSV format

The data was converted from SPSS format to CSV format for ease of working, this was completed using R and R Studio (see Annexure C for R Script).

STEP C – Re-Naming the Columns in the CSV File

Importing the data into R before this step resulted in an error, as such, the columns needed to be renamed. This was completed by opening the file in Microsoft Excel, re-naming all columns and ensuring the alignment of the columns with the data. The new column heading line was then exported to CSV by pasting the column heading line into the CSV file using MS Notepad.

STEP D – Correcting Misalignment of Variables

There were errors in the CSV file due to the syntax in the answers to question H-02 and H-02a. One of the possible answers to these questions was as follows; "Informal dwelling (shack not in backyard, e.g. in an informal/squatter settlement or on a farm". This caused a misalignment of the data as the comma in a CSV file is used to separate different data entries. The search and replace function on Microsoft Notepad was used to convert the comma to a period – this solved this misalignment error in the data.

STEP E – Importing the CSV Dataset and converting the Dataset to an R Data Frame

The CSV dataset was then imported into the R global environment, opened and saved as an R Data Frame. This was used to complete all data pre-processing and analysis (see Annexure C for R Script).

4.2.2. SA Census 2011 Data Pre-Processing

General Data Pre-Processing

Data pre-processing is a critical part of creating any machine learning model and there are multiple issues that need to be addressed during this stage. These include dealing with missing data, categorical data and differing ranges of features.

STEP 1 - Reducing Dataset to Cape Town only

This paper focuses on Cape Town in isolation from the rest of South Africa and therefore a new dataset with only data for Cape Town had to be created. This was done on the basis of District and all entries that had a District Value not equal to "City of Cape Town" were removed. A new data frame was created and a new CSV file was written to create a Cape Town specific Dataset (see Annexure C for R Script).

STEP 2 - Repairing Missing Data

Dealing with missing data is critical, when a data entry is missing data in a particular feature this data entry will need to be repaired. In the case of continuous data the most common approach in data science is to take the mean of the applicable feature over all the data entries and use this in place of the missing feature value (Eremenko and de Ponteves, 2017). In the case of categorical data, the most common approach is to replace the missing feature value with the most frequently observed value. Missing data was checked for in the Cape Town dataset using the filter function in MS Excel and the only variable which contained missing data was the Main Dwelling Variable. There were 824 instances of missing data so relative to the 86,658 data entries this accounted for 0.951% of the total dataset. Unfortunately the data could not be repaired but this low percentage was considered insignificant.

STEP 3 - Encoding Categorical Data

As Random Forest is based on mathematical formulas it was key to encode all text-based data with numerical categories (Eremenko and de Ponteves, 2017). The encoding of categorical data was been completed in line with Annexure B (see Annexure C for R Script). Categories were not amended apart from the Main Dwelling Variable, where categories put forward in (Hogarth, 2015) have been used thus requiring the consolidation of categories from a total of 12 to a total of 9. The consolidated encoding is shown in Table 6 below. In both Main Dwelling and Other Dwelling there was an inverted comma in the value of the variable "*Room/flatlet on a property or a larger dwelling/servants'qua*", this was replaced with a space using the MS Excel replace function as R uses the inverted comma in its coding language.

Census Answer	Census Code	Category*	Code assigned
House or brick/concrete block structure on a separate stand	1	Single-detached	1
Traditional dwelling/hut/structure made of traditional mater	2	Traditional	2
Flat or apartment in a block of flats	3	Flat	3
Cluster house in complex	4	Cluster/Townhouse	4
Town house (semi-detached house in complex)	5		
Semi-detached house	6	Semi-Detached	5
House/flat/room in back yard	7	Second Dwelling	6
Room/flatlet on a property or a larger dwelling/servants qua	10		
Informal dwelling/shack in back yard	8	Backyard Informal	7
Informal dwelling/shack NOT in back yard. e.g. in an informa	9	Informal Settlement	8
Caravan or tent	11	Other	9
Other	12		

*Table 6 – Main Dwelling categories encoded *as defined by (Hogarth, 2015)*

4.3. Data Processing

Multiple sets of models were created for each of the three dependent variables used; Main Dwelling, Tenure and Number of Bedrooms. For each set of models an Un-Tuned[11] and a Tuned[12] model was created. The roadmap showing how each set of models was created is shown below. The R Script for all the data processing can be found in Annexure C.

4.3.1. Roadmap for Random Forest Model Construction

The process for producing the Random Forest model is as follows;

Part 1. Import the CSV data using the *read.csv* function,

Part 2. Select the appropriate variables, using Main Dwelling, Tenure or Number of Bedrooms as the dependent variables using the *subset* function,

Part 3. At this point, a frequency table was generated for the dependent variable using the *table* function,

[11] An Un-Tuned model as discussed in this paper is a model in which the parameters of the Random Forest algorithm have not been tuned or in other words the default parameters have been used, i.e. *ntree* = 500 and *mtry* = Classification sqrt(p) and regression (p/3)

[12] A Tuned model refers to model where the parameters of the Random Forest algorithm have been tuned. In the case of *ntree*, the tuning is done in order to improve the computing performance of the model and in the case of *mtry* to improve the accuracy of the model

Part 4. The dataset was split into a training set and a test set, this was completed using the *caTools*, this is explained further in part 4 below,

Part 5. Change dependent variable to a factor using the *as.factor* function,

Part 6. The HHAGE variable was scaled using the *scale* function,

Part 7. The *randomForest* function was then used to create the model for each of the three dependent variables,

Part 8. The following results were published for each model;

a. A histogram showing the number of nodes per tree in the Random Forest was plotted using the *hist* function,

b. The importance of each variable in the model was then plotted using the *varImPlot* and *importance*,

c. A training set prediction and confusion matrix was then created using the *caret* package. This was completed using the *predict* and *confusionMatrix* functions respectively,

d. Step 8.c was repeated for the test set, at this point the first model in the group had been created for the respective dependent variable,

Part 9. This first model was then tuned by using two methods;

a. Using the *plot* function to show the improvement in the Random Forest's accuracy given the number of trees, this method primarily focused on increasing the computing efficiency of the model,

b. The *tuneRF* function was then used to find the optimum *mtry* value,

Part 10. Step 7 was then completed using the *ntree* value obtained from Step 9.a and the *mtry* value obtained from Step 9.b, importance was set as TRUE,

Part 11. Step 8 was then completed to show the results of the Tuned model.

4.3.2. Part 1 – Importing the Dataset

Part 1 involved the re-importation of the already pre-processed and saved CSV dataset, this was named "Census_2011_Household_CT_Hgth_Enc.csv". This enabled for rapid construction of each of the models as the pre-processing did not have to be re-completed for each model.

4.3.3. Part 2 – Selecting the appropriate Variables

The full descriptions of all the variables mentioned below can be found in Annexure B.

If the variable was a serial/ reference number or if it was a sample weighting the variable was removed. The three variables that have been removed for these reasons were DATREF, SN and X10PERCENT. From this point the variables contained in the Census Data could essentially be divided into two categories; variables that relate to the household and variables that relate to the housing. The housing variables (dependent variables) selected for the predictive models were informed by the focus of the literature, these included MAIN DWELLING, TENURE and BEDROOMS where the number of bedrooms was a proxy for the size of the house (McGaffin, 2014, Hogarth, 2015, Ndziba et al., 2015, McGaffin and Kirova, 2016). In the first set of models MAINDWELLING was the dependent variable with TENURE and BEDROOMS excluded, in the second set of models TENURE was the dependent variable with MAINDWELLING and BEDROOMS excluded and the logic followed for the third set of models where BEDROOMS was the dependent variable. The housing variables that were excluded from the Census data were; OTHERDWELLING, ROOF, WALL, DININGROOMS, LIVINGROOMS, DINING, STUDYROOMS, MULTIUSE, OTHERROOMS, TOTROOMS, WATERPIPED, WSOURCE, WSUPPLY, WSUPPLY2, ALTWSOURCE and TOILET. It could be argued that ECOOKING, EHEATING, ELIGHTING, REFUSE and INTERNET are household variables, however, in the context of this paper it is argued that these variables are housing characteristics and are therefore also excluded.

In assessing the remaining variables it was essential to remove variables from the model that had a low variance. If a variable in a dataset has only one or two classes the variable will not contribute to the accuracy of the model and a large number of variables would significantly increase the size and computing requirements to generate the model. The variables that have been excluded from the model due to low variance are as follows; QNTYPE, QUARTERS, REFRIDGERATOR, STOVE, VACUUM, WASHINGM, COMPUTER, SATELLITE, DVD_PLAYER, MOTORCAR, TV, RADIO, LANDLINE, CELLPHONE, POSTBOX, MAIL, DEATHS, GEOTYPE, PROVINCE, DISTRICT and MUNIC. DEATHSNO has also been removed as this has not been found to be relevant as informed by the literature.

The variables that remained after the above elimination procedure were as follows;

- HHAGE – Age of the Household Head,
- HHPOP – Population Group of the Household Head,
- HHSEX – Gender of the Household Head,
- HINCOME – Household Head Income,
- HHEMPLOY – Employment Status of the Household Head,
- HSIZE – Size of Household,
- XPOP – Majority Population Group of the Household,
- INCOME – Income of the Household.

While the above variables have not been included in a predictive model in the literature, both Hogarth (2015) and Ndziba et al. (2015) use each of these independent variables one at a time to compare and analyse the relationship between the single independent variable and the selected dependent variable. It is important to note that there is an "overlap" between HINCOME and INCOME as well as between HHPOP and XPOP. Due to the "overlap" or multicollinearity[13] between HINCOME (Household Head Income) and INCOME (Household Income) only one could be selected for the model. INCOME was chosen as the literature shows this to be a better measure for household decision making than the merely the income of the household head (Hogarth, 2015, Ndziba et al., 2015, McGaffin and Kirova, 2016). Similarly multicollinearity was present between HHPOP (Population Group of the Household Head) and XPOP (Majority Population Group of the Household). In this case decision making is generally made by the household head and therefore HHPOP was selected for the models (Hogarth, 2015, Ndziba et al., 2015, McGaffin and Kirova, 2016). The Models were therefore constructed using the variables in Table 7 below. The selection of a small, core set of variables was key in order to keep the models as simple as possible given that many of these variables were categorical and each had many possible values. The more variables that are added to a model, the more computational capacity would be required to run the models.

[13] Multicollinearity generally occurs when there are high correlations between two or more predictor (or independent) variables, this "overlap" makes it difficult to assess the individual effect of the predictor variables on the prediction (or dependent variable).

Independent Variables	Dependent Variable(s)
HHAGE HHPOP HHSEX HHEMPLOY HSIZE INCOME	MAIN DWELLING or TENURE or BEDROOMS

Table 7 – Variables selected for the model

4.3.4. Part 3 – Generating a Frequency Table for the Dependent Variable

A frequency table was generated using the *table* function in R for the dependent variable in each of the three groups of models. This assisted in the data analysis by demonstrating how the no-information rate[14] was calculated. The comparison between the model's predicative accuracy and the no-information rate was then used as a proxy for contextual model performance. The Frequency Table for the data was used to determine if there was an Imbalance Problem[15] and if Stratified Sampling[16] was required. While the Stratified Sampling technique did not directly address the imbalance in the data it was used to ensure that the Test Set had sufficient data to test each class in the model, this is discussed further in the next section. The techniques explored to directly address the Imbalance Problem in the data are explored later in this Chapter. The Frequency Tables and Graphs for each of the three sets of models can be found in the Data Analysis section of this paper.

4.3.5. Part 4 – Splitting the Dataset into Training Set and Test Set

The data had to be split into a **Training Set** and a **Test Set**. This was done as the algorithm needed to learn certain characteristics about the data, the training set assisted with this learning while the test set was used to determine the performance (or predictive accuracy) of the machine learning model (Eremenko and de Ponteves, 2017). Initially only

[14] The no information rate is the percentage proportion of the most common class
[15] An imbalance problem exists where one of the classes has a frequency which differs substantially from the frequency of the other classes. This can create a bias in the machine learning algorithm to predict one class as this is the class that dominates other classes
[16] Stratified Sampling is a technique that can be used to eliminate the problem of having an unsuitable number of observations in test set classes. This is done by dividing the entire population into subgroups and then randomly selecting the final subjects proportionally from these different subgroups

'standard splitting' was completed but due to the imbalance problem identified, most notably in the Main Dwelling data, the standard splitting resulted in some test classes not containing any data. This meant that the predictive accuracy for these classes was zero, which reduced the overall model accuracy in cases where the data was more imbalanced and where there was a high training to test ratio.

Standard Splitting

The subset function in R was used to split the data in a predetermined ratio. The split chosen is usually Training Set (80%) and Test Set (20%) (Eremenko and de Ponteves, 2017) however in an attempt to obtain the optimum predictive model three different split ratios were used, these were (70:30) Training:Test, (80:20) and (90:10). These are shown in further detail in Table 8 below.

Stratified Sampling

Stratified sampling is a technique that can be used to counter not having data samples in certain test set classes by ensuring that each class is represented in both the training and test data in a predetermined ratio. Selection of samples within each class is completed randomly, however, it ensures that minority classes are represented at the specified ratio. This procedure was completed by extracting data by class, splitting each of the class-based datasets by the different split ratios described above and then re-combining each of these class-based subsets into a global training set and test set. The split type and ratios used for the models are shown in Table 8 below.

Dependent Variable	Models	Split % (Training:Test)	Split Type
MAINDWELLING	1 & 2	70 : 30	Standard
	3 & 4	80 : 20	
	5 & 6	90 : 10	
	7 & 8	70 : 30	Stratified Sampling
	9 & 10	80 : 20	
	11 & 12	90 : 10	
TENURE	13 & 14	70 : 30	Standard
	15 & 16	80 : 20	
	17 & 18	90 : 10	

TENURE	19 & 20	70 : 30	Stratified Sampling
	21 & 22	80 : 20	
	23 & 24	90 : 10	
BEDROOMS	25 & 26	70 : 30	Standard
	27 & 28	80 : 20	
	29 & 30	90 : 10	
	31 & 32	70 : 30	Stratified Sampling
	33 & 34	80 : 20	
	35 & 36	90 : 10	

Table 8 – Types and ratios of splitting for each of the sets of models

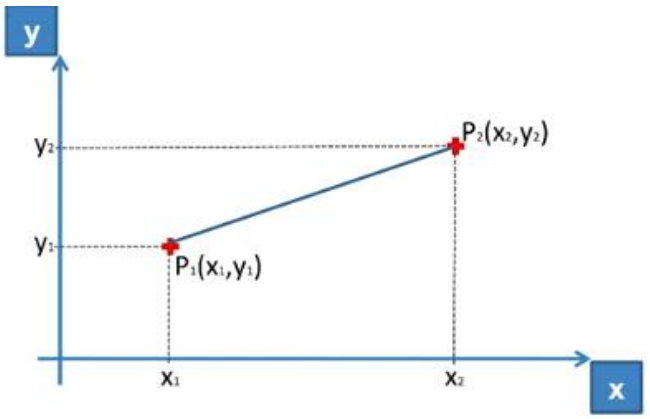
4.3.6. Part 5 – Making the Dependent Variable a Factor

In each of the three groups of models the dependent variable was made a factor, this is a requirement in order to use the *randomForest* function in R. The function *as.factor* was used in R.

4.3.7. Part 6 – Feature Scaling

Most machine learning models are based on the Euclidean distance shown in Figure 20 below, therefore the difference in ranges of features can cause a skewed result in the machine learning model. For example, age and annual salary features would have vastly different ranges. It can clearly be seen by the formula in Figure 20 below that the annual salary would dominate the Euclidean distance rendering the age feature close to irrelevant (Eremenko and de Ponteves, 2017).

Figure 20 - Euclidean distance visualisation & formula (Eremenko and de Ponteves, 2017)



$$\text{Euclidean Distance between } P_1 \text{ and } P_2 = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

To avoid this bias, researchers must scale features. Therefore ranges of -1 to +1 should be allocated to each feature and data entries should be scaled to fit into this range (Eremenko and de Ponteves, 2017). Two primary ways of scaling this data is through **Standardisation** and **Normalisation**, the formulas for these two methods are shown in Table 9 below. Feature scaling was used on HHAGE as this was the only continuous variable left in the dataset. Scaling this feature marginally improved the performance of the models.

Standardisation	Normalisation
$x_{stand} = \frac{x - mean(x)}{standard\ deviation\ (x)}$	$x_{norm} = \frac{x - \min(x)}{\max(x) - \min(x)}$

Table 9 –Standardisation and Normalisation Formulas (Eremenko and de Ponteves, 2017).

4.3.8. Part 7 – Creating the Random Forest Model

The *randomForest* Package was installed in R and the *randomForest* function within this package was used to create the Random Forest Models. This function implements Breiman's Random Forest algorithm (based on Breiman and Cutler's original Fortran code) for classification and regression problems. The full Description, Usage, Arguments and Notes for this function can be found in Annexure D. The procedure's main arguments are number of trees grown in the forest (*ntree*); and number of predictors sampled for splitting at each node (*mtry*). The default value for *ntree* is 500 while that of *mtry* is sqrt(p) for classification and (p/3) for regression. Initially, the *randomForest* with default values was applied to each set of models. These are the models referred to herein as Un-Tuned. Later, Tuned models were generated were the two parameters were varied to improve computational efficiency.

4.3.9. Part 8 – Publishing the Results

The Caret Package was installed in R to assist with the analysis of the results. First a histogram showing the number of nodes per tree in the random Forest was plotted using the *hist* function, this showed the scale of each tree in the forest and the distribution of the scale. Second the importance of each variable in the model was plotted using the

varImPlot to show Mean Decrease in the Gini Coefficient (MDG)[17] in the Un-Tuned Models and both the MDG & Mean Decrease in Accuracy (MDA)[18] in the Tuned Models. The actual values for both MDG & MDA were obtained using *importance*. A prediction and confusion matrix were then created using the *predict* and *confusionMatrix* functions in the caret package. Analysis of these results is discussed further in the data analysis section of this chapter.

4.3.10. Part 9 – Tuning the Random Forest Model

As the initial model in each of the sets had a large number of trees these initial models needed to be tuned to ensure computational efficiency. There are primarily two main parameters which can be tuned in a Random Forest procedure to improve the predictive power of the model. These are: *mtry* and *ntree*. In R it is simple to find the best value of *mtry* through the use of the function *tuneRF*. Starting with the default value of *mtry*, *tuneRF* searches for the optimal value (with respect to Out-of-Bag error estimate) of *mtry* for *randomForest*. The default output includes a plot of OOB error against *mtry* from which the optimal value of *mtry* can be read off. The impact of varying *ntree* on improved performance is also achieved with *tuneRF* which produces an additional plot showing improved accuracy of the model against the number of trees used.

4.3.11. Part 10 – Creating the updated Random Forest Model

Based on Part 9 the Un-Tuned models were then updated to the Tuned models for each of the three dependent variables. This was done by explicitly stating the *ntree* and *mtry* values in the randomForest function. *Importance* was set to TRUE.

4.3.12. Part 11 – Publishing the Results of the New Model

Part 11 involved repeating the steps followed in Part 8 but for the Tuned models.

---

[17] The Mean Decrease in the Gini Coefficient (MDG) is a measure of how each variable contributes to the homogeneity of the nodes and leaves (terminal nodes) in a Random Forest model. This is calculated by comparing the Gini Coefficient for the child nodes with that of the original node. MDG shows the "purity of split" of a variable relative to other variables in the model

[18] The Mean Decrease in Accuracy (MDA) refers mean decrease in accuracy that would occur on average should one of the variables be removed from the model, this measure is most relevant to rank the variables in order of importance and in relation to the global accuracy of the model

4.4. Data Analysis

4.4.1. Predicting Main Dwelling Type

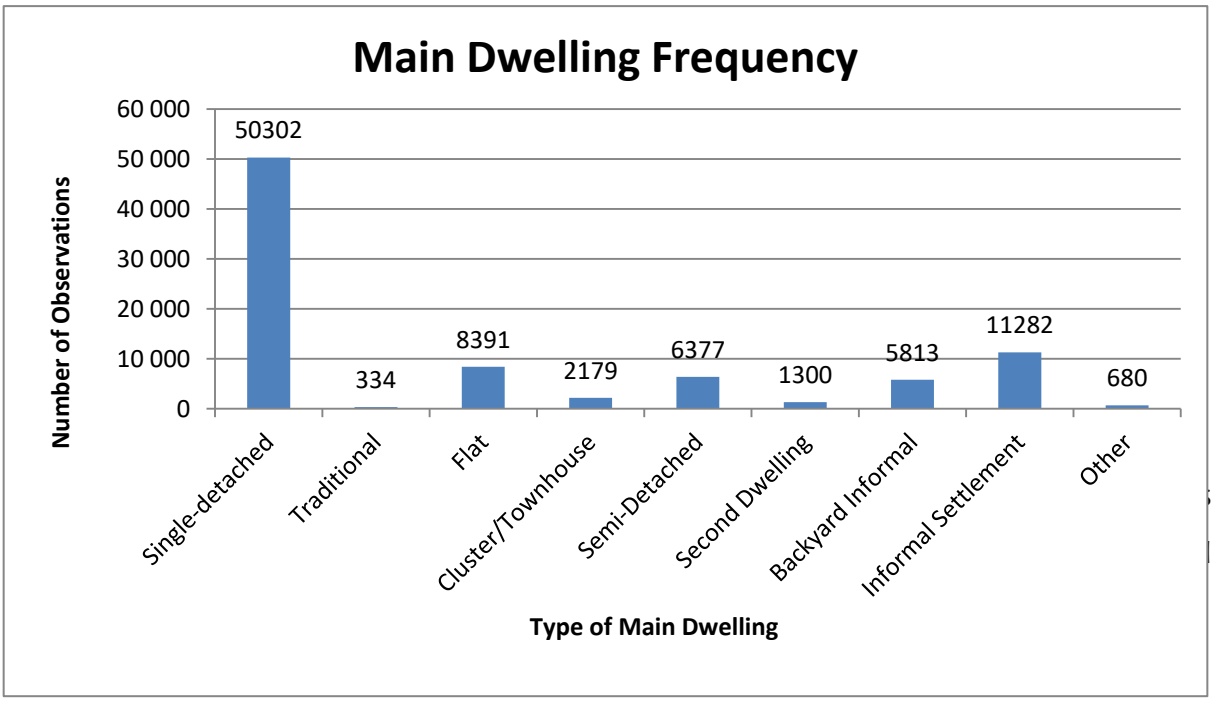
The process set out under 4.3 Data Processing has been followed to produce the results for all the Main Dwelling Models. The frequency distribution for Main Dwelling is shown in Table 10 and Figure 21.

Encoding	Description	Frequency	Relative Frequency
1	Single-detached	50 302	58.0%
2	Traditional	334	0.4%
3	Flat	8 391	9.7%
4	Cluster/Townhouse	2 179	2.5%
5	Semi-Detached	6 377	7.4%
6	Second Dwelling	1 300	1.5%
7	Backyard Informal	5 813	6.7%
8	Informal Settlement	11 282	13.0%
9	Other	680	0.8%
Total		**86 658**	**100.0%**

Table 10 – Frequency and Relative Frequency of Main Dwelling observations

It can clearly be seen that the Single-Detached dwelling was the most popular dwelling type and represents 58% of all household's in the Census 2011 data for Cape Town.

Figure 21 - Main Dwelling Frequency



in this table. For each of the Main Dwelling Models, a rank has been given based on the predictive accuracy of the model and the Kappa statistic.

The predictive accuracy of the model is simply the percentage of correctly predicted values of the dependent variable when the model is presented with independent variable test data. The Kappa statistic (or value) is also an important metric to compare observed accuracy with expected accuracy (random chance). It is used not only to evaluate a single classifier but to evaluate classifiers amongst themselves. Although there is not a standardised interpretation of the Kappa statistic two main interpretations from Landis and Koch (1977) as well as Fleiss et al. (1981) are shown below in Table 11 and Table 12.

Slight	0 – 20%
Fair	>20 – 40%
Moderate	>40 – 60%
Substantial	>60 – 80%
Almost Perfect	>80 - 100%

Table 11 – Landis and Koch (1977) Kappa Interpretation

Poor	<40%
Fair to Good	40 – 75%
Excellent	>75%

Table 12 – Fleiss et al. (1981) Kappa Interpretation

It is significant to note that both scales are to some degree arbitrary and it is important to consider a Confusion Matrix[19] in conjunction with a Kappa statistic as well as interpret the Kappa value in relation to the context of the research. In the case of easily observable phenomena a Kappa of 40% may be considered low but in terms of phenomena that are not easily observable and not easy to interpret a Kappa of 40% may be considered exceptional. Housing Demand is difficult to observe, interpret and predict. Classifiers built and evaluated on data sets with different class distributions can be more reliably compared using Kappa (in contrast to only using accuracy). Kappa is a better indicator of how the classifier performed across all classes (and instances) as simple accuracy can be skewed if the class distribution is similarly skewed. For example, although the simple accuracy of a classifier may be high the distribution may be heavily weighted for a single class (in a 5 class model) and therefore the model may actually have

[19] A Confusion Matrix shows the actual values of the dependent variable in relation to the predicted values. This is useful to visualise and analyse the predictive accuracy of individual classes within the model

poor predictive capabilities in relation to the other four classes. Interpreting the Kappa statistic would assist to determine if this was the case or not.

The highest performing Main Dwelling Model in Table 13 in terms of Accuracy and Kappa was Model 12, this model is presented and analysed in full in the sections that follow.

Model	Type	Tuning	No Info Rate	Test Set			Accuracy Rank	Kappa Rank
				Model Predictive Accuracy	Predictive Improvement	Kappa Statistic		
1	70% Training Data	None	58.05%	60.93%	2.88%	22.25%	9	6.5
2		Yes ntree = 150 mtry = 2		60.95%	2.90%	22.45%	8	4
3	80% Training Data	None		60.75%	2.70%	21.77%	11.5	11
4		Yes ntree = 150 mtry = 2		60.75%	2.70%	21.67%	11.5	12
5	90% Training Data	None		61.00%	2.95%	22.19%	6	8
6		Yes ntree = 150 mtry = 2		61.04%	2.99%	22.13%	4.5	10
7	70% Training Data, Stratified Sampling	None		61.04%	2.99%	22.14%	4.5	9
8		Yes ntree = 150 mtry = 2		61.07%	3.02%	22.25%	3	6.5
9	80% Training Data, Stratified Sampling	None		60.96%	2.91%	22.36%	7	5
10		Yes ntree = 150 mtry = 2		60.91%	2.86%	22.47%	10	3
11	90% Training Data, Stratified Sampling	None		61.16%	3.11%	22.56%	2	2
12		Yes ntree = 150 mtry = 2		61.19%	3.14%	22.78%	1	1

Table 13 – Performance Statistics and Rank for Main Dwelling Models

4.4.2. Analysis of the Highest Performing Main Dwelling Model

All models were produced in sets, initially producing an Un-Tuned version of the model and then a Tuned version, both of these versions are analysed for the best performing model. In terms of the Main Dwelling Models the Tuned Models generally outperformed the Un-Tuned Models and the higher the ratio of training data to test data, the better the models performed. The stratified sampling approach worked better than the standard split approaches used. The best performing model was a Tuned model that had a 90:10 split ratio and was split using stratified sampling.

Main Dwelling - Model 11 – 90% Training, Stratified Sampling (Un-Tuned)

It is possible to visually plot one of the decision trees in the Random Forest, however, given the complexity of each tree this visualisation is not useful for a person to interpret. Instead a histogram showing the number of nodes for each tree in the forest has been used in order to visualise the scale, complexity and shape of the forest. This is shown in Figure 22 below.

Figure 22 - Main Dwelling Model 1
Number of Nodes for the Trees

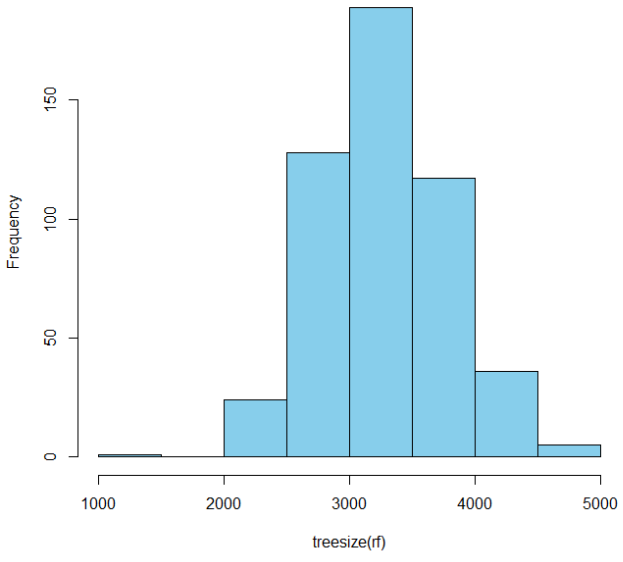
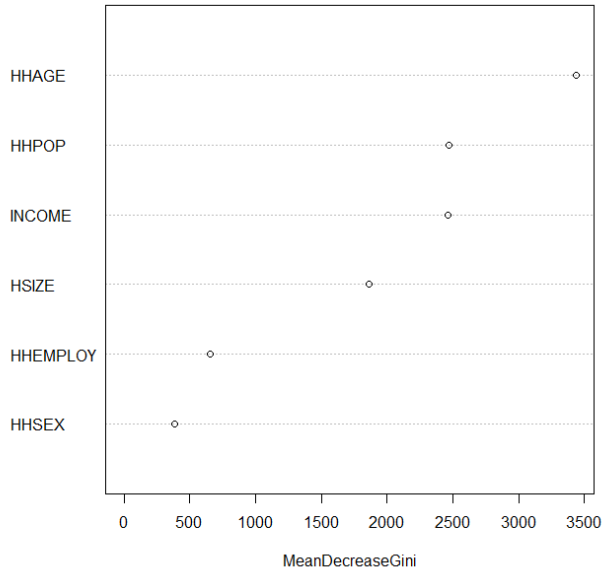


Figure 23 - Main Dwelling Model 1
Variable Importance



The Variable importance for the Initial Main Dwelling Model is shown in Figure 23 above, the importance is based on the Mean Decrease in the Gini Coefficient (MDG) should the variable be removed from the model. This shows that the variable that contributed the most to the purity of split of the model was the Age of the Household Head with a MDG of 3357, followed by the Population Group of the Household head with a MDG of 2522, very closely followed by the Income of the Household with a MDG of 2378. The Size of the Household was the fourth most important factor with a MDG of 1859. The Employment status and Gender of the household head were the least important factors in terms of homogeneity of split.

The performance statistics for the Un-Tuned Main Dwelling Model measured against the Test Data are shown below in Table 14. These statistics are available for the training data as well, however, these were not analysed as the test of the predictive model is on data the model has not seen before, i.e. the test data. The accuracy of model 11 was 61.16%. The no information rate for the Main Dwelling data was high at 58.05%, this meant that the model with a predictive accuracy of 61.16% only made an improvement on the no information rate by 3.11%. The no information rate is the frequency of the most common class, i.e. as opposed to constructing a predictive model, if the most common class was always selected as the outcome, one would predict the correct outcome 58.05%. Each dependent variable would have a different no information rate. The 95% confidence interval ranged from 60.13% to 62.19%.

Indicator	Value/Range
Accuracy	61.16%
95% CI	60.13% - 62.19%
No Information Rate	58.05%
Kappa	22.56%

Table 14 – Performance Statistics for the Un-Tuned Main Dwelling Model

According to Landis and Koch (1977) and Fleiss et al. (1981) the Kappa for this Main Dwelling Model is fair and poor respectively.

The confusion matrix for the selected Un-Tuned Main Dwelling Model (Model 11) is shown in Table 15 below.

		Reference									Total Predictions
		1	2	3	4	5	6	7	8	9	
Predict	1	4 596	28	681	191	604	102	330	518	59	**7 109**
	2	0	0	0	0	0	0	0	0	0	**0**
	3	58	0	97	22	7	4	2	1	0	**191**
	4	0	0	0	0	0	0	0	0	0	**0**
	5	2	0	0	0	0	0	0	0	0	**2**
	6	0	0	0	0	0	0	0	0	0	**0**
	7	0	0	0	0	0	0	0	0	1	**1**
	8	375	6	62	5	27	24	250	610	8	**1 367**
	9	0	0	0	0	0	0	0	0	0	**0**
Total Ref		**5 031**	**34**	**840**	**218**	**638**	**130**	**582**	**1 129**	**68**	**8 670**
Class Accuracy		**91.35%**	**0.00%**	**11.55%**	**0.00%**	**0.00%**	**0.00%**	**0.00%**	**54.03%**	**0.00%**	**n/a**
Weighted Class Accuracy		**53.01%**	**0.00%**	**1.12%**	**0.00%**	**0.00%**	**0.00%**	**0.00%**	**7.04%**	**0.00%**	**61.16%**

Table 15 – Confusion Matrix for the selected Un-Tuned Main Dwelling Model

The Confusion Matrix has been colour coded, the green cells indicate the where the model has correctly predicted a class relative to the reference data, the red cells are where the model has incorrectly predicted a class and the white cells indicate where the model has made no predictions. For example, the model predicted Class 1 (Single Detached) 5,031 times. 4,596 of these predictions were correct and 435 predictions were incorrect, therefore the accuracy of predicting a single-detached dwelling was 91.35% above both the predictive accuracy of the model as a whole and the no information rate. The overall predictive accuracy was 61.16% which as previously discussed does improve on the no information rate but only marginally so. In general the model predicted other classes poorly, with the second ranked class only being predicted 54.03% of the time.

Main Dwelling – Tuning Model 11 to Produce Model 12

Model 11 was then plotted as described in Part 9 above to provide the graph shown in Figure 24 below. Each of the coloured lines represents one of the Main Dwelling Classes and the error present when the forest is made up of a certain number of trees. For

example the black line in Figure 24 below indicates that when the forest is made up of a single tree, the error is greater than 0.4, when the number of trees is increased to 150, this error reduces to below 0.4. Some classes caused the error to increase when a greater number of trees were added. The plot shows that if more than 150 trees were added, there would be no noticable reduction in the error globally and additional trees would simply slow down the compuational performance of the model, therefore model 12 has been Tuned in terms of the *ntree* variable to be *ntree* = 150.

Figure 24 - Main Dwelling Model 1
Random Forest Error Plot

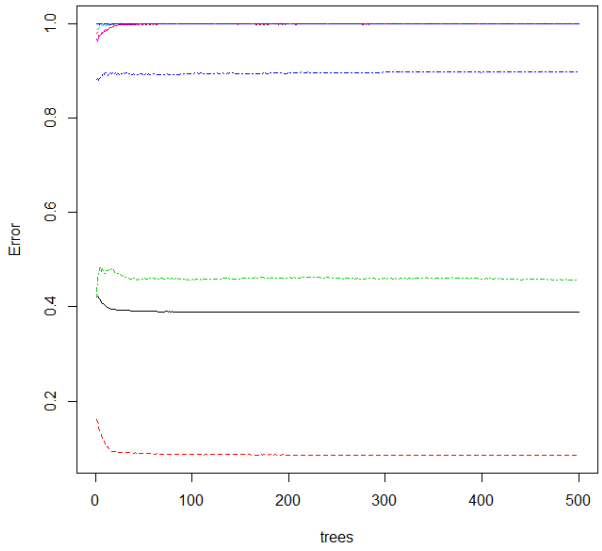


Figure 25 - Main Dwelling Model 1
Tuning mtry

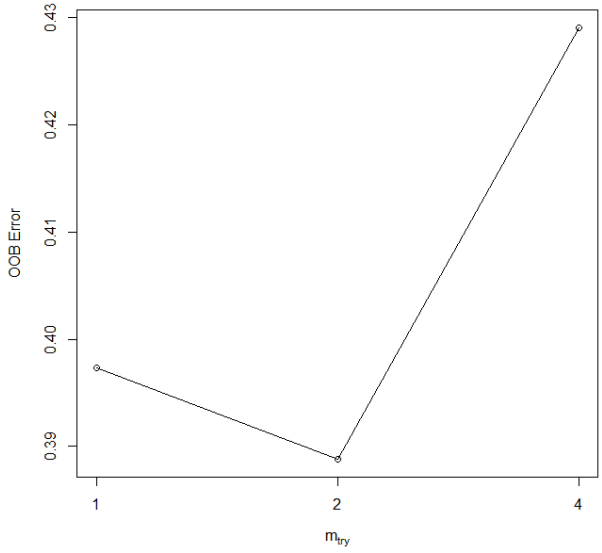


Figure 25 above shows the graph plotted when using the *mtry* tuning function, this showed that the optimum *mtry* value was 2. This corresponds to an OOB Error (Out of Bag) of less than 0.39. This tuning was done using a *stepFactor* of 0.5 and an *improve* value of 0.05. *Trace* was set to TRUE. Based on the outcome shown in Figure 25 above, the *mtry* value in the Tuned model was set to 2.

Main Dwelling - Model 12 – 90% Training, Stratified Sampling (Tuned)

The histogram of the Model 12 Random Forest has been plotted in Figure 26. This now shows a larger number of trees having a higher number of nodes as the distribution is skewed to the right, unlike the Un-Tuned Main Dwelling Model which showed a

distribution closer to a normal distribution and centred around 3500 nodes. Figure 27 shows the variable importance from two different perspectives, the Mean Decrease in Accuracy (MDA) and as discussed previously the MDG. Population Group of the Household Head was ranked 1 (98.9) and 2 (2523) respectively and the Age of the Household head is Ranked 2 (81.0) and 1 (3357) respectively. The Income of the Household was ranked 3 (65.4 MDA and 2378 MDG) and the Household Size was ranked 4 (54.3 MDA and 1859 MDG) for both measures. As was seen in the Un-Tuned model the two least important factors in terms of both MDA & MDG were Employment Status and the Gender of the Household Head.

Figure 26 - Main Dwelling Model 2 Number of Nodes for the Trees

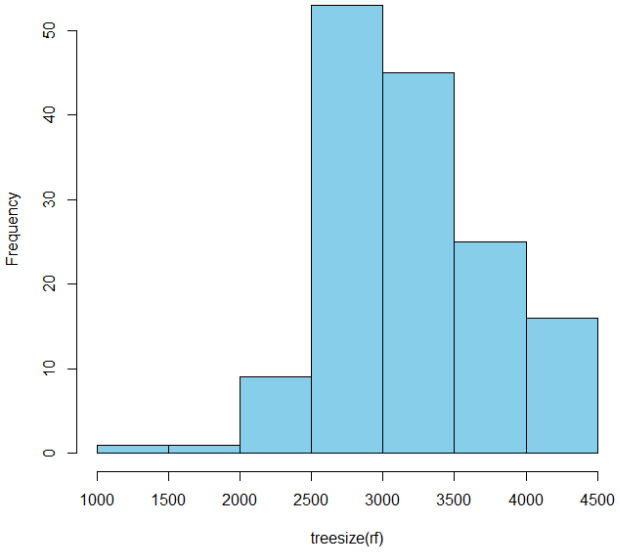
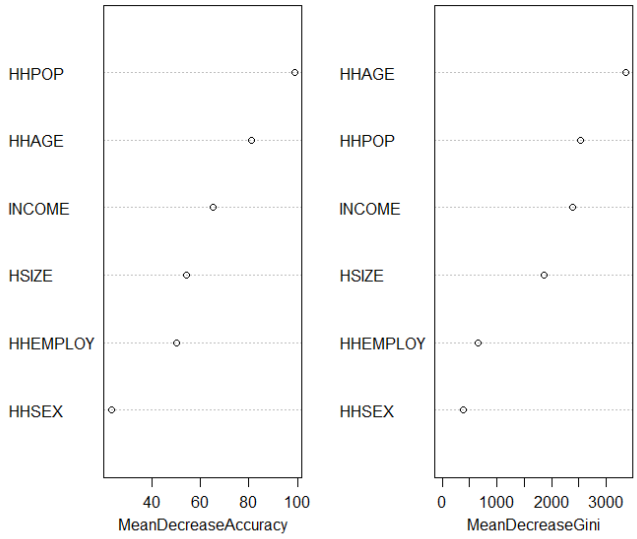


Figure 27 - Main Dwelling Model 2 Variable Importance



The performance statistics for the Tuned Main Dwelling Model measured against the Test Data are shown below in Table 16.

Indicator	Value/Range
Accuracy	61.19%
95% CI	60.15% - 62.22%
No Information Rate	58.05%
Kappa	22.78%

Table 16 – Performance Statistics for the Tuned Main Dwelling Model

The accuracy of the model was 61.19%. The no information rate remained at 58.05%. This meant that the model only improved on the no information rate by 3.14%. The 95% confidence interval ranged from 60.15% to 62.22%. According to the Landis and Koch (1977) approach the Kappa value of 22.78% was fair and according to Fleiss et al. (1981) it was poor.

		Reference									Total Predictions
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	
Predict	**1**	4 589	27	676	190	603	100	329	516	59	**7 089**
	2	0	0	0	0	0	0	0	0	0	**0**
	3	61	0	103	23	8	5	2	1	0	**203**
	4	0	0	0	0	0	0	0	0	0	**0**
	5	2	0	0	0	0	0	0	0	0	**2**
	6	0	0	0	0	0	0	0	0	0	**0**
	7	1	0	0	0	0	0	1	0	1	**3**
	8	378	7	61	5	27	25	250	612	8	**1 373**
	9	0	0	0	0	0	0	0	0	0	**0**
Total Ref		**5 031**	**34**	**840**	**218**	**638**	**130**	**582**	**1 129**	**68**	**8 670**
Class Accuracy		**91.21%**	**0.00%**	**12.26%**	**0.00%**	**0.00%**	**0.00%**	**0.17%**	**54.21%**	**0.00%**	**n/a**
Weighted Class Accuracy		**52.93%**	**0.00%**	**1.19%**	**0.00%**	**0.00%**	**0.00%**	**0.01%**	**7.06%**	**0.00%**	**61.19%**

Table 17 – Confusion Matrix for the selected Tuned Main Dwelling Model

The above Confusion Matrix shows that similar to the Un-Tuned Main Dwelling model the highest class accuracy was for Class 1 (Single-Detached) with 91.21%, the second highest class accuracy was 54.21% for Class 8 (Informal Settlement). Class 8 was predicted better and Class 1 was predicted worse in the Tuned Model when compared to the Un-Tuned Model. Similar to the Un-Tuned model, the remaining classes were predicted poorly.

4.4.3. Predicting Tenure Type

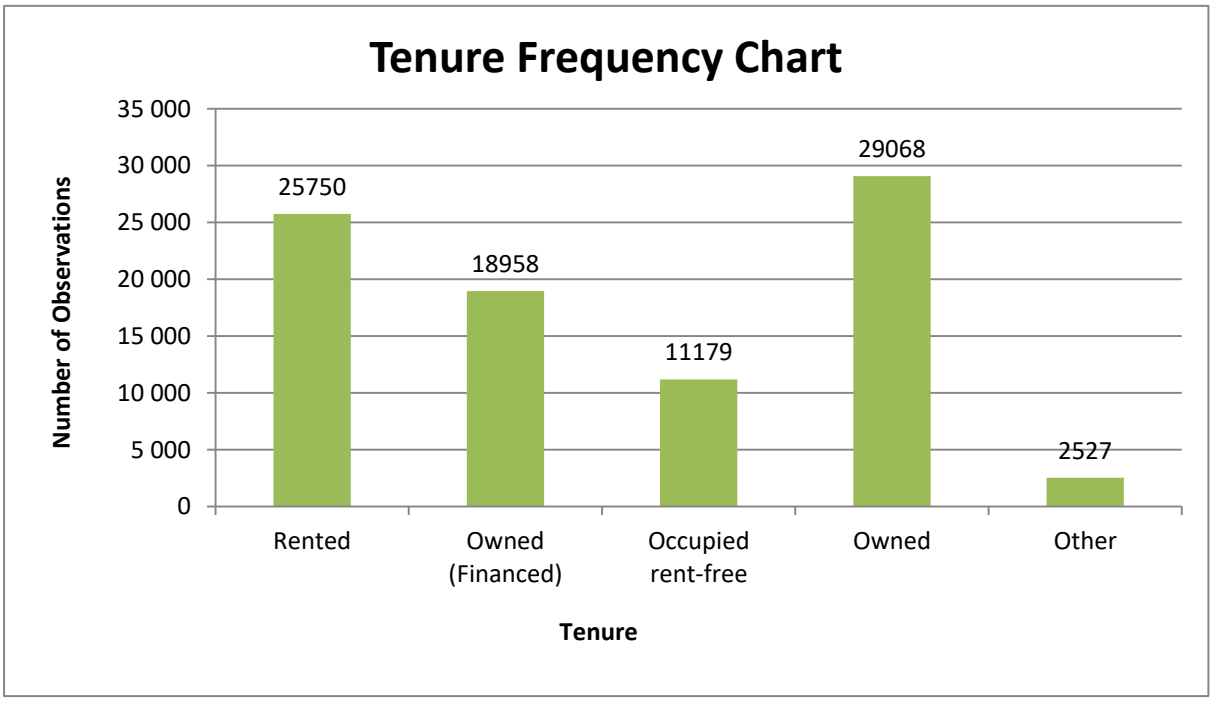
The Tenure Models as with the Main Dwelling Models followed the process set out under 4.3 Data Processing. The Frequency distribution for Tenure is shown in Table 18 and Figure 28 below.

Encoding	Description	Frequency	Relative Frequency
1	Rented	25 750	29.4%
2	Owned (Financed)	18 958	21.7%
3	Occupied rent-free	11 179	12.8%
4	Owned	29 068	33.2%
5	Other	2 527	2.9%
Total		**87 482**	**100.0%**

Table 18 – Frequency and Relative Frequency of Tenure Observations

Based on the Census 2011 data 33.2% of household's in the Cape Town region owned their housing due finance on these homes already being paid off.

Figure 28 - Tenure Frequency



The results for all the Tenure Models are shown in Table 19 below. Each Tenure Model has been ranked based on the predictive accuracy of the model and the Kappa. Table 19 shows that the highest performing model in terms of Accuracy and Kappa was Model 14. This model is presented and analysed in full in the sections that follow.

Model	Type	Tuning	No Info Rate	Test Set			Accuracy Rank	Kappa Rank
				Model Predictive Accuracy	Predictive Improvement	Kappa Statistic		
13	70% Training Data	None	33.23%	47.42%	14.19%	25.74%	2	3
14		Yes ntree = 150 mtry = 2		47.54%	14.31%	25.93%	1	1
15	80% Training Data	None		47.30%	14.07%	25.70%	4	5
16		Yes ntree = 150 mtry = 2		47.29%	14.06%	25.73%	5	4
17	90% Training Data	None		46.54%	13.31%	24.43%	12	12
18		Yes ntree = 150 mtry = 2		46.74%	13.51%	24.73%	10	10
19	70% Training Data, Stratified Sampling	None		47.28%	14.05%	25.63%	6	6
20		Yes ntree = 150 mtry = 2		47.40%	14.17%	25.83%	3	2
21	80% Training Data, No strat Sampling	None		47.08%	13.85%	25.37%	8	8
22		Yes ntree = 150 mtry = 2		47.19%	13.96%	25.57%	7	7
23	90% Training Data, No strat Sampling	None		46.58%	13.35%	24.56%	11	11
24		Yes ntree = 150 mtry = 2		46.77%	13.54%	24.84%	9	9

Table 19 – Performance Statistics and Rank for Tenure Models

4.4.4. Analysis of the Highest Performing Tenure Model

Between the Un-Tuned and Tuned Tenure Models, there was no clearly superior type if this factor was analysed between sets of models, in some cases the Tuned models performed better and in other cases the Un-Tuned models performed better. In general, the lower the ratio of training data to test data, the better the models performed. The stratified sampling approach did not perform as well as the standard approach in the Tenure models. It is argued that this is most likely due to the fact that the Tenure data is not as unbalanced as the Main Dwelling data. The random selection from all the data in the standard approach would therefore be less likely to result in test classes with no observations. The best performing model was a Tuned Model that had a 70:30 split ratio and was split using the standard technique. Both the Un-Tuned and Tuned versions of the models are analysed below for the best performing set of models, Model 13 & 14.

Tenure - Model 13 – 70% Training, Standard Sampling (Un-Tuned)

The Tenure Random Forest Model 13 can be described by Figures 29 and 30 below.

Figure 29 - Tenure Model 3

Number of Nodes for the Trees

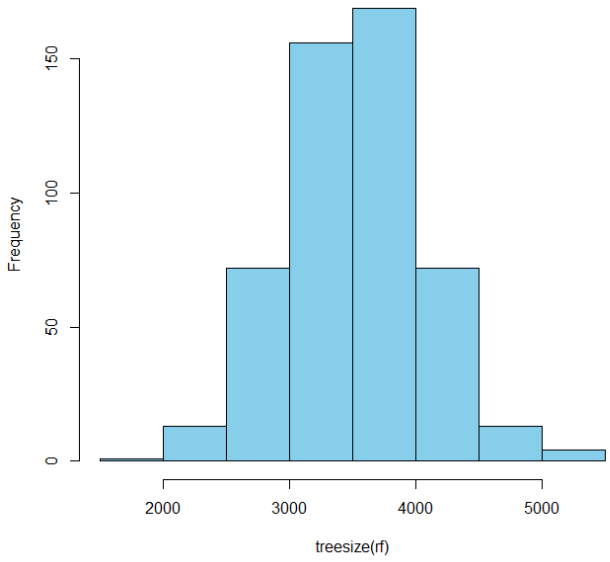
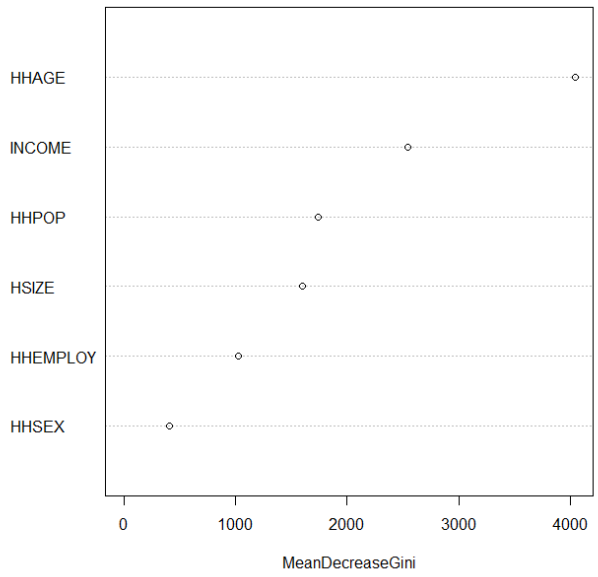


Figure 30 - Tenure Model 3

Variable Importance



Model 13 is composed of a total of 500 trees with a distribution close to a normal distribution, the majority of the trees have between 3000 and 4500 nodes, the largest tree in the forest has 5500 nodes and the smallest has 2000 nodes.

The Variable importance for the Un-Tuned Tenure Model is shown in Figure 30 above. This shows that the variable that contributes the most to the purity of split of the model is the Age of the Household Head with a MDG of 4044, followed by the Income of the Household with a MDG of 2545. The Population Group of the Household head and the Size of the Household followed, with a MDG of 1741 and 1599 respectively. As in the case of the both Main Dwelling Models, the Employment Status and Gender of the household head both resulted in a lower homogeneity of split.

The performance statistics for the Un-Tuned Tenure Model measured against the Test Data are shown below in Table 20 below. The accuracy of the model was 47.42%, however the no information rate was 33.23% which was substantially lower than that for Main Dwelling. The model accuracy therefore improved on the no information rate by 14.19%. The 95% confidence interval ranged from 46.82% to 48.03%.

Indicator	Value/Range
Accuracy	47.42
95% CI	46.82% - 48.03%
No Information Rate	33.23%
Kappa	25.74%

Table 20 – Performance Statistics for the Un-Tuned Tenure Model

The Kappa statistic was 25.74%, this according to Landis and Koch (1977) is considered fair and according to Fleiss et al. (1981) it is considered poor. It is however an improvement on the best Main Dwelling Models Kappa Statistic.

The below Confusion Matrix shows that the highest class accuracy for the Un-Tuned Tenure Model was for Class 4 (Owned) with 62.59% followed by Class 1 (Rented) with 52.05%. Class 2 (Owned, Financed) has the third highest class accuracy with 49.11%. All of these three classes had a class accuracy higher than the no information rate. Class 3 was predicted poorly by the model and Class 5 was not predicted correctly by the model.

		Reference					Total Predictions
		1	2	3	4	5	
Predict	1	4 021	1 307	1 106	1 897	284	**8 615**
	2	1 203	2 793	85	1 185	44	**5 310**
	3	149	23	174	180	37	**563**
	4	2 351	1 564	1 989	5 458	393	**11 755**
	5	1	0	0	0	0	**1**
Total Ref		**7 725**	**5 687**	**3 354**	**8 720**	**758**	**26 244**
Class Accuracy		**52.05%**	**49.11%**	**5.19%**	**62.59%**	**0.00%**	**n/a**
Weighted Class Accuracy		**15.32%**	**10.64%**	**0.66%**	**20.80%**	**0.00%**	**47.42%**

Table 21 – Confusion Matrix for the selected Un-Tuned Tenure Model

Tenure - Tuning Model 13 to Produce Model 14

Model 13 was then plotted as described in part 9 of the 4.3 Data Processing section, this is shown in Figure 31 below. This shows that after 150 trees there was no decrease in error, therefore *ntree* was set to 150.

Figure 31 - Tenure Model 13

Random Forest Error Plot

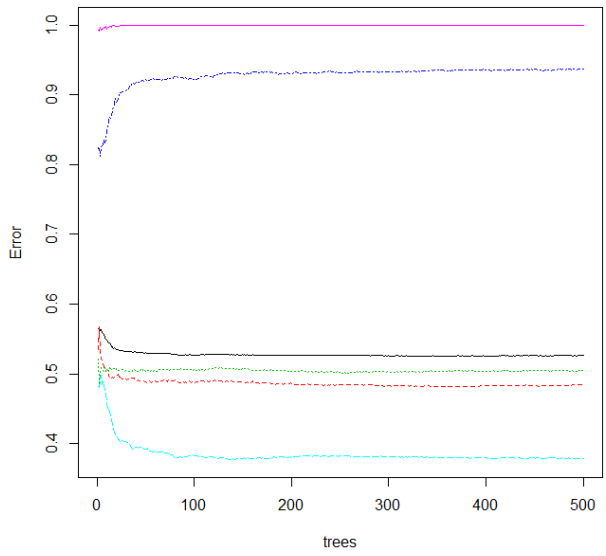


Figure 32 - Tenure Model 13

Tuning mtry

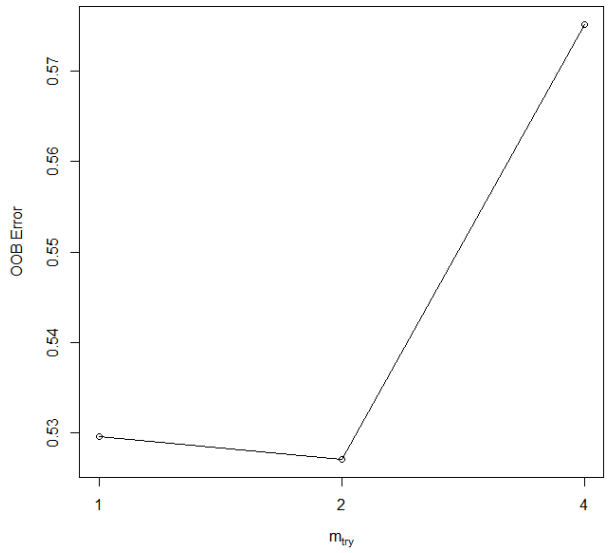


Figure 32 shows that the OOB Error was minimised when *mtry* = 2. As in the Main Dwelling set of models, the *stepFactor* was 0.5 and the *improve* value was 0.05. *Trace* was again set to TRUE. This analysis again showed that *mtry* should be set to 2.

Tenure - Model 14 – 70% Training, Standard Sampling (Tuned)

The Tenure Random Forest Model 14 can be described by Figures 33 and 34 below.

Figure 33 - Tenure Model 14

Number of Nodes for the Trees

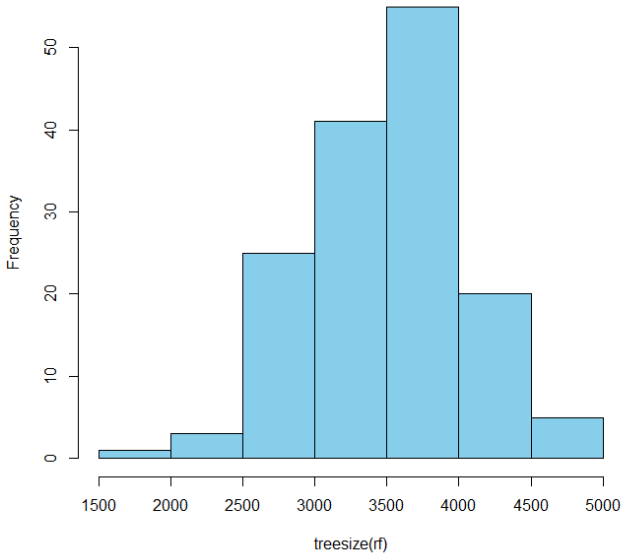
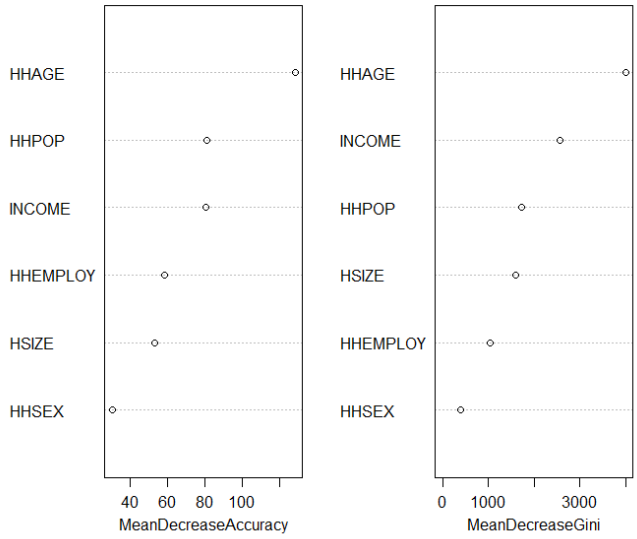


Figure 34 - Tenure Model 14

Variable Importance



Model 14 shown in Figure 33 was composed of a total of 150 trees with a distribution skewed to the left, unlike the Un-Tuned model which approximates a normal distribution. The majority of the trees have between 3000 and 4500 nodes, the largest tree in the forest has 5000 nodes and the smallest has 2000 nodes.

As in the Main Dwelling Models Figure 34 shows the variable importance using both MDA and MDG. Household Age was ranked the most important factor as it was the highest contributor to both MDA (128.5) and MDG (4004). Population Group of the Household Head was ranked 2 (80.9) and 3 (1721) respectively and Income of the Household was ranked 3 (80.7) and 2 (2567) respectively. The Employment Status was ranked 4 (58.6) and 5 (1040) respectively and the Size of the Household was ranked 5 (53.1) and 4 (1593) respectively. The Gender of the household head was the least important factor when it came to predicting the Tenure of a household.

The performance statistics for the Tuned Tenure Model measured against the Test Data are shown below in Table 22. The accuracy of the model was 47.54% and improved on the no information rate by 14.31%, this was the highest improvement on the no information rate of any of the three dependent variables. The 95% confidence interval ranged from 46.94% to 48.15%.

Indicator	Value/Range
Accuracy	47.54%
95% CI	46.94% - 48.15%
No Information Rate	33.23%
Kappa	25.93%

Table 22 – Performance Statistics for the selected Tuned Tenure Model

The Kappa statistic was 25.93% which according to Landis and Koch (1977) is considered fair and according to Fleiss et al. (1981) it is considered poor. This was marginally better than the Kappa for the Un-Tuned model and notably better than the Kappa for the Main Dwelling Models.

		Reference					**Total Predictions**
		1	**2**	**3**	**4**	**5**	
Predict	**1**	4 063	1 341	1 117	1 917	283	**8 721**
	2	1 208	2 813	86	1 201	49	**5 357**
	3	135	23	174	175	39	**546**
	4	2 318	1 510	1 977	5 427	387	**11 619**
	5	1	0	0	0	0	**1**
Total Ref		**7 725**	**5 687**	**3 354**	**8 720**	**758**	**26 244**
Class Accuracy		**52.60%**	**49.46%**	**5.19%**	**62.24%**	**0.00%**	**n/a**
Weighted Class Accuracy		**15.48%**	**10.72%**	**0.66%**	**20.68%**	**0.00%**	**47.54%**

Table 23 – Confusion Matrix for the selected Tuned Tenure Model

The above Confusion Matrix shows that the highest class accuracy was for Class 4 (Owned) with 62.24%, the second highest class accuracy was for Class 1 (Rented) with 52.60% and the third highest was for Class 2 (Owned, Financed) with 49.46%. Class 3 was poorly predicted by the model and Class 5 was not predicted correctly by the model. The model accuracy of the Tuned Tenure Model was marginally better than that of the Un-

Tuned Model. Even though the accuracy of this model was lower than that of the best Main Dwelling model, the model improved on the no information rate substantially better than in the Main Dwelling Model. The model also had a better Kappa statistic than the Main Dwelling Model and is therefore viewed as a superior predictive model.

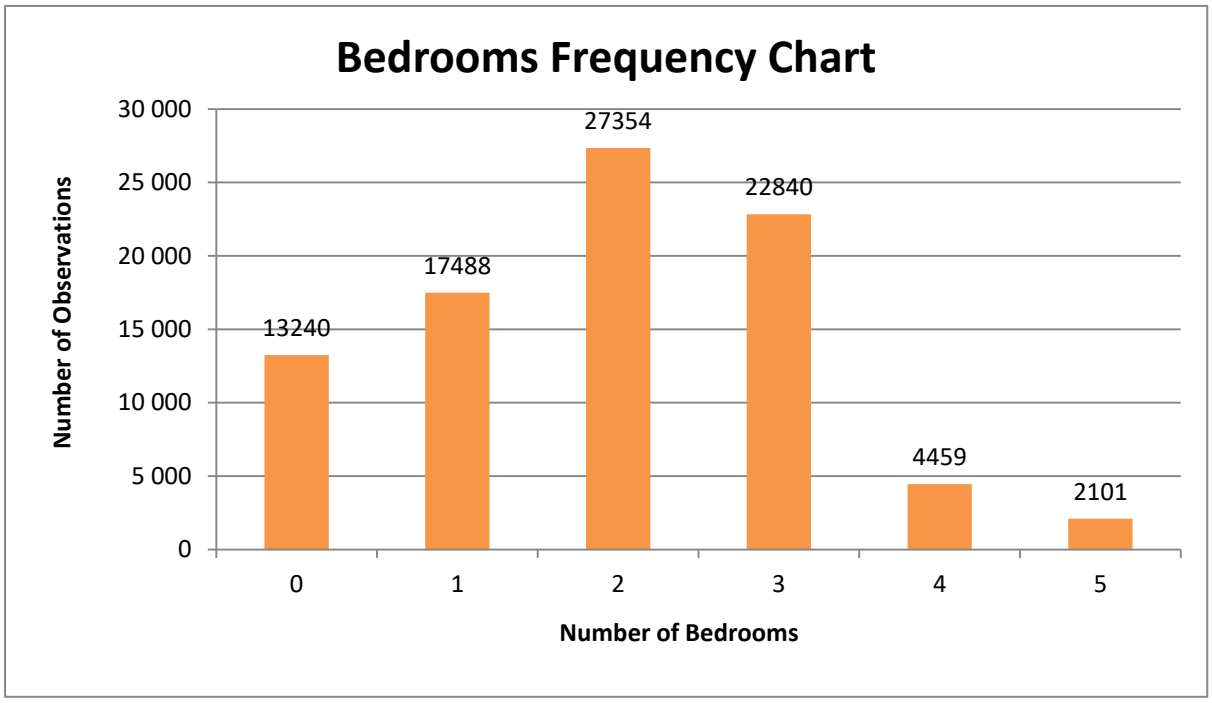
4.4.5. Predicting Number of Bedrooms

The Number of Bedrooms models have followed the process set out under 4.3 Data Processing. The Frequency distribution for Bedrooms is shown in Table 24 and Figure 35 below. In Cape Town in 2011 31.3% of Household's resided in Housing with 2 bedrooms.

Encoding	Description	Frequency	Relative Frequency
0	No Private Bedroom	13 240	15.1%
1	1 Bedroom	17 488	20.0%
2	2 Bedroom	27 354	31.3%
3	3 Bedroom	22 840	26.1%
4	4 Bedroom	4 459	5.1%
5	5 Bedroom	2 101	2.4%
Total		**87 482**	**100.0%**

Table 24 – Frequency and Relative Frequency of Number of Bedrooms Observations

Figure 35 - Number of Bedrooms Frequency



The results for all the Number of Bedrooms Models are shown in Table 25 below. As in the case with all the other Dependent variables, only test sets have been presented due to the focus of this paper on a predictive model. For each of the Number of Bedrooms Models a rank has been given based on the predictive accuracy of the model and the Kappa. Table 25 shows that the highest performing model in terms of Accuracy and Kappa was Model 29. This model is presented and analysed in full below.

Model	Type	Tuning	No Info Rate	Test Set			Accuracy Rank	Kappa Rank
				Model Predictive Accuracy	Predictive Improvement	Kappa Statistic		
25	70% Training Data	None	31.27%	44.68%	13.41%	25.50%	4	3
26		Yes ntree = 150 mtry = 2		44.48%	13.21%	25.31%	8	7
27	80% Training Data	None		44.44%	13.17%	25.25%	10	9
28		Yes ntree = 150 mtry = 2		44.52%	13.25%	25.34%	7	6
29	90% Training Data	None		45.08%	13.81%	26.23%	1	1
30		Yes ntree = 150 mtry = 2		44.88%	13.61%	25.95%	2	2
31	70% Training Data, Stratified Sampling	None		44.61%	13.34%	25.44%	5	4.5
32		Yes ntree = 150 mtry = 2		44.47%	13.20%	25.29%	9	8
33	80% Training Data, No strat Sampling	None		44.70%	13.43%	25.44%	3	4.5
34		Yes ntree = 150 mtry = 2		44.54%	13.27%	25.24%	6	10
35	90% Training Data, No strat Sampling	None		44.42%	13.15%	25.16%	11	11
36		Yes ntree = 150 mtry = 2		44.27%	13.00%	24.96%	12	12

Table 25 – Performance Statistics and Rank for Number of Bedrooms Models

4.4.6. Analysis of the Highest Performing Number of Bedrooms Model

In terms of the Number of Bedrooms Models generally the Un-Tuned Models performed better than the Tuned Models. There was no clear progressive improvement of accuracy given the different split ratios and there was also no clear benefit of using the stratified sampling technique with these Models. The best performing model was an Un-Tuned model that had a 90:10 split ratio and was split using the standard technique previously described. While the Un-Tuned model was the better performer, the associated Tuned model will also be presented in an attempt to explore why the tuned model underperformed.

Number of Bedrooms - Model 29 – 90% Training, Standard Sampling (Un-Tuned)

Random Forest Model 29 can be described by Figures 36 and 37 below.

Figure 36 - Bedrooms Model 5

Number of Nodes for the Trees

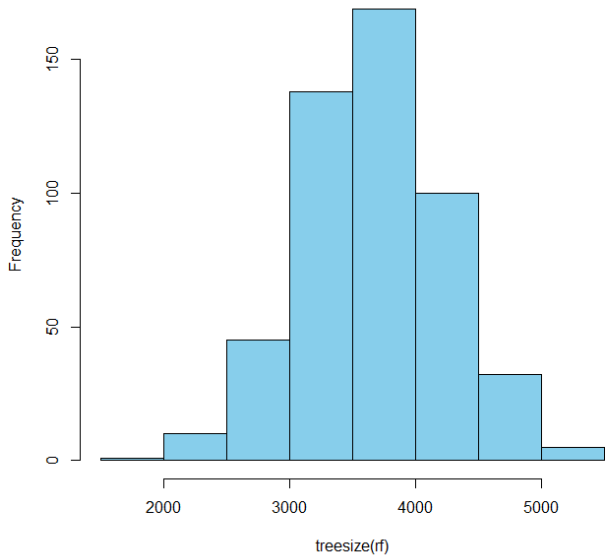
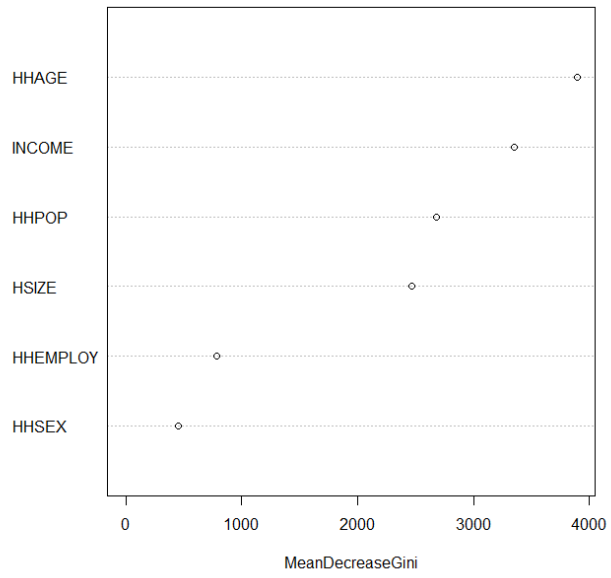


Figure 37 - Tenure Model 5

Variable Importance



Model 29 was composed of a total of 500 trees the majority of which had between 3000 and 4500 nodes, the largest tree in the forest had 5500 nodes and the smallest had 1500 nodes. The distribution could be approximated by a normal distribution. The Variable importance for the initial Bedrooms model is shown in Figure 37 above. This shows that

the variable that contributed the most to the purity of split of the model was the Age of the Household Head with a MDG of 3896 followed by the Income of the Household with a MDG of 3350. The Population Group of the Household Head and the Size of the Household followed, with a MDG of 2678 and 2468 respectively. As in the case of the Main Dwelling and Tenure Models, the Employment status and Gender of the household head both resulted in the second lowest and lowest homogeneity of split respectively.

The performance statistics for the Un-Tuned Bedrooms Model measured against the Test Data are shown below in Table 26. The accuracy of the model was 45.08% and the no information rate was 31.27%. The model accuracy of improved on the no information rate by 13.81%. The 95% confidence interval ranged from 44.04% to 46.13%.

Indicator	Value/Range
Accuracy	45.08%
95% CI	44.04% - 46.13%
No Information Rate	31.27%
Kappa	26.23%

Table 26 – Performance Statistics for the selected Un-Tuned Bedrooms Model

The Kappa statistic was 26.23% which according to Landis and Koch (1977) is considered fair and according to Fleiss et al. (1981) it is considered poor. This Kappa statistic was better than the Kappa statistic for both the selected Main Dwelling Models and the Tenure Models.

		Reference						**Total Predictions**
		0	**1**	**2**	**3**	**4**	**5**	
Predict	**0**	714	507	242	35	4	8	1 510
	1	193	315	254	93	8	10	873
	2	356	772	1 545	789	83	48	3 593
	3	59	154	692	1 346	324	131	2 706
	4	2	1	1	18	22	11	55
	5	0	0	1	3	5	2	11
Total Ref		**1 324**	**1 749**	**2 735**	**2 284**	**446**	**210**	**8 748**
Class Accuracy		**53.93%**	**18.01%**	**56.49%**	**58.93%**	**4.93%**	**0.95%**	**n/a**
Weighted Class Accuracy		**8.16%**	**3.60%**	**17.66%**	**15.39%**	**0.25%**	**0.02%**	**45.08%**

Table 27 – Confusion Matrix for the selected Un-Tuned Bedrooms Model

The above Confusion Matrix shows that the highest class accuracy was for Class 3 (3 Bedrooms) with 58.93%, the second highest was Class 2 (2 Bedrooms) with 56.49% and the third highest was Class 0 (No Distinct Bedrooms) with 53.93%. Class 1 (1 Bedroom) was only predicted 18.01% of the time and the model poorly predicted the remaining classes.

Number of Bedrooms - Tuning Model 29 to Produce Model 30

Model 29 was then plotted as described in part 9 of the 4.3 Data Processing section, this is shown in Figure 39 below. *ntree* was set to 150 when running the tuned models, however, after analysis of Figure 29 below it is clear that additional trees would have been required to reduce the error. To improve the Tuned model ntree should have been adjusted to around 200.

Figure 38 - Bedrooms Model 29

Random Forest Error Plot

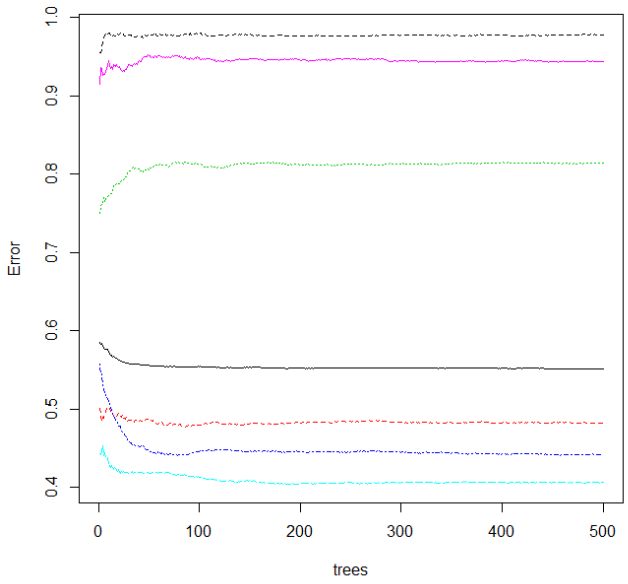


Figure 39 - Bedrooms Model 29

Tuning mtry

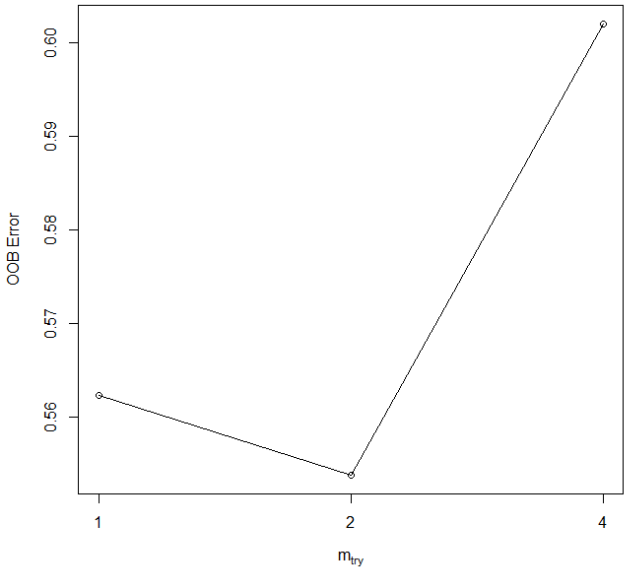


Figure 39 shows that the OOB Error is minimised when *mtry* = 2. As in the Main Dwelling and Tenure Models, the *stepFactor* was 0.5 and the *improve* value was 0.05. *Trace* was again set to TRUE.

Number of Bedrooms - Model 30 – 90% Training, Standard Sampling (Tuned)

Random Forest Model 30 can be described by Figures 40 and 41 below.

Figure 40 - Bedrooms Model 30

Number of Nodes for the Trees

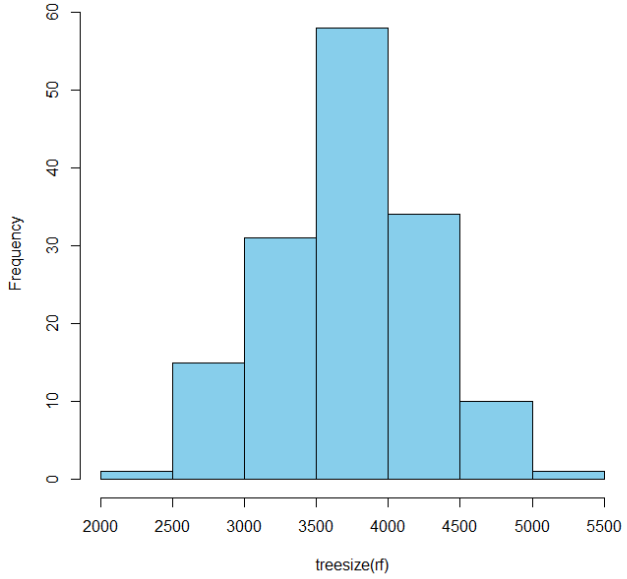
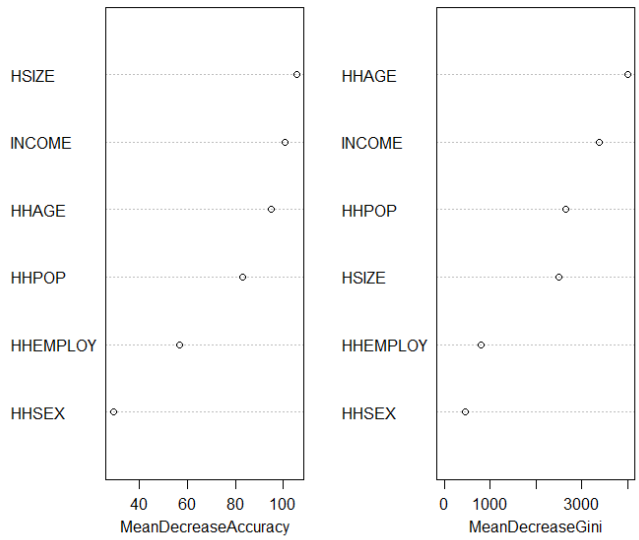


Figure 41 - Bedrooms Model 30

Variable Importance



Model 30 shown in Figure 40 is composed of a total of 150 trees, the majority of which have between 3500 and 4500 nodes, the largest tree in the forest has 5500 nodes and the smallest has 2500 nodes. The distribution could be approximated with a normal distribution.

As in the Main Dwelling and Tenure Models, Figure 34 shows variable importance using both MDA and MDG as measures. Size of the Household was ranked 1 (105.5) and 4 (2492) respectively and Income was ranked 2 for both MDA (100.5) and MDG (3378). Household Head Age was ranked 3 (95.1) and 1 (3996) for MDA & MDG respectively. Population Group of the Household Head was ranked 4 (82.9) and 3 (2638) for MDA & MDG respectively. As found with all the other models, the Employment status and Gender of the household head both resulted in the second lowest and lowest respectively for both MDA and MDG.

The performance statistics for the Tuned Bedrooms Model measured against the Test Data are shown below in Table 28. The accuracy of the model was 44.88% and therefore the model improved on the no information rate by 13.61%. The 95% confidence interval ranged from 43.83% to 45.93%. This is the only instance where the Tuned model does not outperform the Un-Tuned Model.

Indicator	Value/Range
Accuracy	44.88%
95% CI	43.83% - 45.93%
No Information Rate	31.27%
Kappa	25.95%

Table 28 – Performance Statistics for the selected Tuned Bedrooms Model

The Kappa statistic was 25.95% which according to Landis and Koch (1977) is considered fair and according to Fleiss et al. (1981) it is considered poor. This is still better than the Kappa statistic for the Main Dwelling and Tenure Models however this is marginally lower than the Kappa Statistic for the Un-Tuned Bedrooms Model.

		Reference						**Total Predictions**
		0	**1**	**2**	**3**	**4**	**5**	
Predict	**0**	705	504	236	35	4	8	1 492
	1	201	317	269	100	9	10	906
	2	358	779	1 538	788	84	49	3 596
	3	59	148	690	1 340	320	131	2 688
	4	1	1	1	18	25	11	57
	5	0	0	1	3	4	1	9
Total Ref		**1 324**	**1 749**	**2 735**	**2 284**	**446**	**210**	**8 748**
Class Accuracy		**53.25%**	**18.12%**	**56.23%**	**58.67%**	**5.61%**	**0.48%**	**n/a**
Weighted Class Accuracy		**8.06%**	**3.62%**	**17.58%**	**15.32%**	**0.29%**	**0.01%**	**44.88%**

Table 29 – Confusion Matrix for the selected Tuned Bedrooms Model

The above Confusion Matrix shows that the highest class accuracy was for Class 3 (3 Bedrooms) with 58.67%, the second highest was for Class 2 (2 Bedrooms) with 56.23% and the third highest was for Class 0 (No Distinct Bedrooms) with 53.25%. These class accuracy rankings were exactly the same as for the Un-Tuned Model however, each of the class accuracies were slightly lower. Class 1 (1 Bedroom) was the only class the Tuned

Model predicted slightly better than the Un-Tuned model. Much like the Un-Tuned model the other classes were poorly predicted. As identified above, the Tuned model was not set to the optimum *ntree* number and this is likely why the Tuned model underperformed relative to the Un-Tuned model.

4.4.7. Comparison of the selected Models

Table 30 below shows a comparison of the top performing sets of models produced for each of the Dependent variables that the models were attempting to predict.

Dependent Variable	Selected Model	Type	Tuning	No Info Rate	Test Set		
					Model Predictive Accuracy	Predictive Improvement	Kappa Statistic
MAIN DWELLING	11	90% Training Data, Stratified Sampling	None	58.05%	61.16%	3.11%	22.56%
	12		Yes ntree = 150 mtry = 2		61.19%	3.14%	22.78%
TENURE	13	70% Training Data	None	33.23%	47.42%	14.19%	25.74%
	14		Yes ntree = 150 mtry = 2		47.54%	14.31%	25.93%
BEDROOMS	29	90% Training Data	None	31.27%	45.08%	13.81%	26.23%
	30		Yes ntree = 150 mtry = 2		44.88%	13.61%	25.95%

Table 30 – Comparative Summary of the selected Models

The top performing models in each of the three statistical measures used are highlighted in blue in the above Table. Model 12 (Main Dwelling) had the highest overall predictive accuracy with 61.19%. Model 14 (Tenure) had the highest predictive improvement when compared with the no-information rate and Model 29 (Bedrooms) had the highest Kappa statistic. It is argued in this paper that Model 14, the Tuned Tenure Model, is the most useful model as this has the highest improvement on the no information rate. The

Bedrooms Model 29 is viewed as the second most successful model, also based on its predictive improvement on the no information rate.

This metric is critical as the time and effort it takes to construct these models is significant and if there is no noteworthy improvement in predictive accuracy than simply selecting the most common class then the model cannot be considered a successful predictive model. While Main Dwelling Model 12 has a very high overall predictive accuracy its class accuracy was generally poor and it did not notably improve on the no information rate.

The imbalanced nature of the classes in the Main Dwelling variable is one of the reasons identified for the poor performance of the models, specifically the Main Dwelling models. This imbalance created two problems; firstly it meant that where the training to test split ratio was high, some test classes did not contain any data to test the models performance. Secondly it meant that the models became good at predicting the most frequent classes but were poor at predicting minority classes. The multi-class imbalance problem is a complex one and whether or not it can be solved, depends on the outcome that is required by the research.

4.4.8. Understanding the multi-class imbalance classification problem and improving the predictive accuracy

Imbalanced data occurs when the number of observations in one class is significantly lower than those belonging to the other classes (Upasana, 2017). The reason for this is that Machine Learning Algorithms generally improve accuracy by reducing the error and therefore they do not take class distribution or balance into account (Upasana, 2017). In the case of electricity fraud for example, fraudulent transactions account for around 1-2% of the total number of observations. In this case the requirement is to improve the identification of the rare, minority class as opposed to achieving higher overall accuracy (Upasana, 2017). Standard classifier algorithms such as Decision Tree and Logistic Regression have a bias towards classes with a higher number of observations. The features of the minority class are therefore ignored (Upasana, 2017).

The two major strategies for dealing with imbalanced data are improving classification algorithms or balancing classes in the training data, the latter strategy is preferred due to its wide application. Both up and down sampling techniques were considered to further improve the predictive accuracy of the models. Up sampling is the technique of growing the data pool for analysis by inserting synthetically generated samples into the recorded dataset, examples include SMOTE (synthetic minority oversampling technique) and ROSE (random over sampling). Down sampling is the technique of removing samples from the dataset to create more alignment in the class frequency distribution.

A quick experiment to align class frequency was completed using the Synthpop algorithm. This was completed on the Main Dwelling data, the most unbalanced data in this study. The data was split into subsets by class and then the population of each class to was set to 10,000 observations. This 'alignment' function in the Synthpop algorithm allows for both up sampling and down sampling to be conducted at the same time. The subsets were then recombined to create a dataset with class frequency that was perfectly aligned. This experiment worked extremely well for the minority classes - in some cases increasing class accuracy from 0% to 54%, however, the global accuracy went down as the class accuracy for the majority class dropped significantly. This finding aligns with Upasana (2017)'s observation that balancing will not increase overall accuracy. This paper is concerned with predicting demand as a whole and therefore these techniques have not been included in this paper.

4.5. Conclusion

The Random Forest models generated showed that this method of segmentation was capable of defining and ranking the "importance" of multiple variables to the predictive accuracy of the model.

In the best performing Main Dwelling Model the Population Group of the Household Head was the most important factor, followed by the Age of the Household Head. The Income of the Household, the Household Size and the Employment Status of the Household Head were the next three most important factors. The Gender of the

Household Head played an almost insignificant role in determining the accuracy of the model.

As one may expect in the chosen Tenure Model the Age of the Household Head played the important role in the accuracy of the model and this variable far outweighed the role any other individual variable played. The Population Group of the Household Head was the next most important variable, very closely followed by the Income of the Household. The Employment Status of the Household Head was the next most important contributor followed closely by the Size of the Household. Once again the Gender of the Household Head played was the least important factor.

The selected Number of Bedrooms Model showed that the Size of the Household was the most important factor. This aligned with expectations as the high cost of housing would mean it would be unlikely for a Household to stay in a larger house if they did not need to do so due to the Size of the Household. The Income of the Household and the Age of the Household Head were the next two key contributors to the accuracy of the model. These were followed by the Population Group of the Household Head and the Employment Status of the Household Head. Like both the Main Dwelling Model and the Tenure Model the Gender of the Household Head played a negligible role.

The use of Random Forest to segment housing demand worked well as it was able to assess the contribution of both categorical and continuous data in the same model as well as being able to assess the contribution of all the independent variables in relation to each other. The models were robust and handled the non-linear decision boundaries present in the Census data. This method therefore overcame the shortcomings of previously applied methods.

The Models developed were successfully used to predict the outcomes for each of the three target variables (Main Dwelling, Tenure and Number of Bedrooms) with data the model had not seen before. The predictive accuracy of these models with this test data showed that the models could be used to predict a target variable for new household with completely unseen characteristics.

The predictive accuracy for the best performing Main Dwelling Model (Model 12) was 61.19% and represented the model with the highest overall predictive accuracy when

compared to the Tenure and Main Dwelling Models. This however, was only 3.14% over the no information rate so there was not a large predictive improvement. The Kappa Statistic for this model was 22.78% and was lower than both the Kappa Statistics for the best performing Tenure and Number of Bedrooms Models.

The best performing Tenure Model (Model 14) had a predictive accuracy of 47.54%, however, this model had the highest improvement on the no information rate at 14.31% predictive improvement. It is therefore argued that this is the most useful predictive model produced. The Kappa Statistic was 25.93%.

The Main Dwelling Model that performed the best was Model 29 and had a predictive accuracy of 45.08% and improved on the no-information rate by 13.81%. This model had the best Kappa statistic of 26.23%

In the case of the best performing Main Dwelling model the predictive accuracy did not meaningfully surpass the no information rate as it did in the case of the Tenure and Number of bedrooms models. Understanding and predicting housing demand is a complex problem and while the Random Forest technique does offer some noteworthy predictive improvement, it would be desirable to predict the dependent variables with a higher degree of accuracy.

Stratified sampling was implemented to counter not having observations in certain test set classes and improve the predictive accuracy which worked well on the Main Dwelling data.

Up and down sampling techniques were also considered to improve the predictive accuracy of the models however these techniques were not successful and are therefore not included in this paper. A key conclusion of this paper is that the variables available in the Census data are not discriminatory enough and that perhaps further questions should be added to the Census in order to better understand and predict the housing demand problem in the future.

5. CONCLUSIONS AND RECCOMMENDATIONS

5.1. Introduction

The chapter begins with a discussion around the research problem found in Chapter 1. Then an overview of the Research Aims and Objectives is then outlined, before an evaluation of the research questions is conducted. Conclusions are then drawn after which recommendations are made.

5.2. Research Problem

The research problems that have been addressed in this research are as follows;

Research problem A

An inadequate understanding of housing demand in South Africa has curtailed the ability of both the public and private sector to provide appropriate housing products to satisfy current demand

While housing demand is complex in nature this study provides a model that can successfully predict the demand for Main Dwelling Type, Tenure and Number of bedrooms based on very few structural characteristics of a household provided by the Census data. This methodology can be applied to any Census data nationally and globally and can be used by both private and public stakeholders to make more informed decisions about households demand for different types of housing stock and tenure options. It is noted that the lower-than-expected predictive capabilities of the model do dilute its application for decision makers and key role players but this points to the need for more relevant and applicable input data.

Research problem B

Market Segmentation methods used to address this problem to date are either too simplistic or conceptually flawed.

Studies have over simplified housing demand and have not allowed for the nuanced understanding of demand that is required to address the current Cape Town housing problem, nor have any of the methods applied suitably provided a predictive model for housing Type, Tenure and Number of Bedrooms. The literature and results of this study clearly show that the Supervised Learning Algorithm Random Forest is an appropriate methodology to segment housing demand to create a model capable of predicting selected dependent variables given unseen household characteristics. The robustness of this tool when dealing with different types of data, multi-class problems and non-linear decision boundaries contributes to its success when predicting housing demand.

5.3. Overview of Research Objectives

The research objectives that have been achieved through this research are as follows;

Research objective A

Ascertain current methods of segmenting housing demand

A review of the literature showed that current single variable methods to segment housing demand all over-simplify the complex nature of housing demand. Both Single variable techniques and multivariable techniques explored in the literature do not provide the in-depth understanding that is required to address the current housing problem.

Research objective B

Identify the appropriate statistical method to determine the propensity of household's to demand particular types of housing

Market Segmentation techniques used to-date have inadequately explained how household's make decisions to choose certain types of housing. The literature and the results of the paper show that Random Forest is an appropriate methodology to provide insight into housing demand through the construction of predictive models. The Random Forest models are robust when dealing with different types of data, multi-class problems

and non-linear decision boundaries, all characteristics that are present in housing demand.

Research objective C

Gain access to a dataset with suitable information regarding household attributes and housing types in a suitable format, or convert data into a suitable format

The Census 2011 dataset was used this was accessed through UCT's Data First Website and was converted from SPSS format to CSV with the use of R, MS WordPad and MS Excel. Converting the dataset into CSV format allowed for successful error checking and the setup for pre-processing of the data.

Research objective D

Construct and run Random Forest Models

The six Random Forest Models were successfully constructed and run. Graphical, tabular and statistical information was generated for each model using standard R packages as well as bolt-on packages, these included; caTools, randomForest and caret.

Research objective E

Analyse the results of the Random Forest Models

The data was successfully presented using graphs and tables. The structures of the Random Forest models were also discussed and key statistics were analysed to establish the overall performance of the models. The most successful model was Model 14 which predicted the Tenure of a household given certain household characteristics. The Age of the Household Head was the most important factor followed by the Population Group of the Household Head, which was narrowly more important than the Income of the Household. Although the predictive accuracy of the model was only 47.47%, the improvement of Predictive Accuracy and the Kappa statistic were good given the difficult to observe phenomenon of household demand for a particular type of housing.

Research objective F

Draw conclusions

It is possible to apply Random Forest to the 2011 Census data to produce a predictive model. The predictive accuracy of the model does vary largely, primarily based on the dependent variable being predicted, there is also a variance across the classes predicted. These models provide insight into the complex nature of a household's demand for a particular type of Dwelling, Tenure and Number of Bedrooms.

5.4. Evaluation of Research Questions

The research questions in this paper are as follows;

Is a Random Forest method of segmentation suitable to overcome the shortcomings of the previously applied methods?

The Random Forest models generated show that this segmentation method can define the most important variables that contribute to the predictive accuracy of the model thereby providing a clear "importance" ranking to each of the independent variables. In the Main Dwelling the Population Group of the Household head was the most important, in the Tenure Model the Age of the Household head played the key role and in the Number of Bedrooms the Income of the Household was the most important independent variable. The Random Forest model overcame the shortcomings of previously applied methods by not only being able to assess the contribution of all the independent variables in relation to each other but also by being able to deal with categorical and continuous data as well as handling the non-linear underlying decision boundaries in the data well.

If so, could the Random Forest models be used to predict a target variable for a new household with completely unseen characteristics?

The Random Forest models selected show that the Main Dwelling Type, Tenure and Number of Bedrooms could be predicted 61.19%, 47.54% and 45.08% of the time respectively when given entirely new information never seen by the model. While in the case of Main Dwelling Type this did not significantly surpass the no information rate, in

the case of Tenure and Number of Bedrooms the increase in predictive accuracy was notable. It would be preferable to predict the dependent variable with a higher degree of accuracy however increasing the accuracy of similar models would be limited by the relevance of the available household characteristic data. It was found that the available variables in the Census data are not discriminatory enough to provide a large predictive improvement on the no information rate.

5.5. Limitations

There were a number of difficulties encountered throughout the research process. Notably the data was difficult to get hold of in a useful (non-aggregated) format. Once the data had been obtained it needed to be converted and once converted, error checked. The magnitude of the data made the error checking challenging, however, using a combination of MS Excel and MS WordPad this was completed.

Given the sheer complexity of the data and the resulting Decision Trees and Random Forests it was difficult to convey the structure of the Random Forests. A single Decision Tree in one of the Random Forest Models was generated for sake of interest however, the complexity of this tree meant that a human would struggle to interpret even a single tree in this form. It was therefore more useful to visualise the Random Forests based on the number of trees in the forest and the average number of nodes per tree to get an idea of the scale, structure and complexity of the Random Forest Models.

There is a very high percentage of Single Detached Dwellings in Cape Town and this meant that there was a very high no-information rate. This could have contributed to the low improvement of predictive accuracy in terms of the Type of Main Dwelling. It is possible that as the majority of the housing stock available in Cape Town is of the Single Detached Type, household's with a certain array of characteristics that would usually cause them to select another Type of Dwelling were forced to select the Single Detached Type. This conclusion is drawn as the major error made by the Main Dwelling Model was when the model predicted a singled-detached house to be one of the other housing types. The results of the study perhaps comment not only on the success of the methodology applied but could in fact point to another poignant issue, the diversity of housing stock available in cape town.

The lack of software packages available to tackle the multi-class imbalance problem prevalent in the Census data was a major constraint to model performance.

5.6. Areas for Further Research

Housing Stock Limitation on the Interpretation of Demand

It is noted in many areas of the literature and it is reiterated in this paper that the model can only predict what household's actually do and not what household's want. This is the case for two reasons;

1. A household can only choose a dwelling type that is already present in the housing stock, housing choices are therefore limited at the very least proportionally by the housing stock available. It could be argued that the housing market cannot be truly understood unless these forced underlying structural characteristics can be quantified and removed from the analysis to determine a "pure housing demand", and,
2. The model is built on "static data" at a certain point in time and the prediction is based on the status quo as it was then.

It may be possible to overcome these shortfalls to a degree by adding two additional questions to the Census, for example; "All else equal, if you had the choice would you change the type of dwelling that you are currently living in?" and "If so, what type of dwelling would you choose?"

Nature of Submarkets over time

It would be useful to test if the definitions of submarkets are unchanging or if significant changes do occur over time.

Application of Random Forest to include the spatial lens

It would be useful to produce a composite Random Forest Model taking into account not only the affordability and structural lens but also the spatial lens.

REFERENCES

ABEND, G. 2008. The meaning of 'theory'. *Sociological Theory,* 26**,** 173-199.

ALBERT, M. 2007. The propensity theory: a decision-theoretic restatement. *Synthese,* 156**,** 587-603.

ALDERSON, W. 2006. The heterogeneous market and the organized behavior system. *A Twenty-First Century Guide to Aldersonian Marketing Thought*.

ALPAYDIN, E. 2014. *Introduction to machine learning*, MIT press.

AMARATUNGA, D., BALDRY, D., SARSHAR, M. & NEWTON, R. 2002. Quantitative and qualitative research in the built environment: application of "mixed" research approach. *Work study,* 51**,** 17-31.

ASSAEL, H. & ROSCOE JR, A. M. 1976. Approaches to market segmentation analysis. *The Journal of Marketing***,** 67-76.

BHATTACHERJEE, A. 2012. Social science research: principles, methods, and practices.

BITBUCKET. 2014. *Whereabouts London - Command Line K-Means Data Clusterer* [Online]. Bitbucket.org. Available: https://bitbucket.org/fcclab/command-line-k-means-data-clusterer [Accessed 2017-07-01].

BOURASSA, S. C., HAMELINK, F., HOESLI, M. & MACGREGOR, B. D. 1999. Defining housing submarkets. *Journal of Housing Economics,* 8**,** 160-183.

BREIMAN, L. 2001. Random forests. *Machine learning,* 45**,** 5-32.

BREIMAN, L., FRIEDMAN, J., STONE, C. J. & OLSHEN, R. A. 1984. *Classification and regression trees*, CRC press.

BURRELL, G. & MORGAN, G. 1979. *Sociological paradigms and organisational analysis*, london: Heinemann.

BUTLER, A. 2017. *Contemporary South Africa*, Macmillan International Higher Education.

CHAF 2017. Cape Town Residential Property Market Report.

CHALMERS, A. F. 2013. *What is this thing called science?*, Hackett Publishing.

CITY OF OTTAWA. 2007. *Growth Projections for Ottawa 2006 – 2031: Prospects for Population, Housing and Jobs* [Online]. Available: www.ottawa.ca [Accessed 2017-04-02].

CLARK, L. & PREGIBON, D. 1992. Statistical models in S. *chapter Tree-Based Models***,** 377-419.

CLAYCAMP, H. J. & MASSY, W. F. 1968. A theory of market segmentation. *Journal of Marketing Research***,** 388-394.

CSIR 2016. Gauteng Housing Demand and Backlog Findings Report.

DE'ATH, G. & FABRICIUS, K. E. 2000. Classification and regression trees: a powerful yet simple technique for ecological data analysis. *Ecology,* 81**,** 3178-3192.

DOBNEY. 2012. *Market Segmentation Research and Processes* [Online]. Available: http://www.dobney.com/Research/segmentation.htm [Accessed 2017-04-01].

DOLNIČAR, S. 2004. Beyond "commonsense segmentation": A systematics of segmentation approaches in tourism. *Journal of Travel Research,* 42**,** 244-250.

EREMENKO, K. & DE PONTEVES, H. 2017. Machine Learning A-Z™: Hands-On Python & R In Data Science. Udemy.

FINANCIAL & COMMISSION, F. 2013. Exploring alternative finance and policy options for effective and sustainable delivery of housing in South Africa. *Midrand, South Africa*.

FINANCIAL AND FISCAL COMMISSION 2012. Building an Inclusionary Housing Market: Shifting the Paradigm for Housing Delivery in South Africa.
FINANCIAL AND FISCAL COMMISSION. 2014. *Understanding Housing Demand in South Africa* [Online]. Available: http://www.ffc.co.za/ [Accessed 2017-04-02].
FLEISS, J. L., LEVIN, B. & PAIK, M. C. 1981. *Statistical methods for rates and proportions*.
FULLER, D., HANLAN, J. & WILDE, S. J. 2005. Market segmentation approaches: Do they benefit destination marketers? *School of Commerce and Management Papers*, 149.
FUTURE CITIES CATAPULT. 2014. *Whereabouts London* [Online]. Available: http://whereaboutslondon.org/#/map [Accessed 2017-07-01].
GALSTER, G. 1996. William Grigsby and the analysis of housing sub-markets and filtering. *Urban Studies,* 33, 1797-1805.
GAN, Q. & HILL, R. J. 2009. Measuring housing affordability: Looking beyond the median. *Journal of Housing economics,* 18, 115-125.
GOODLAD, R. 1996. The housing challenge in South Africa. *Urban Studies,* 33, 1629-1646.
HÁJEK, A. Interpretations of probability. In The Stanford Encyclopedia of Philosophy (Zalta, 2003. Citeseer.
HOEK, J., GENDALL, P. & ESSLEMONT, D. 1996. Market segmentation: A search for the Holy Grail? *Journal of Marketing Practice: Applied Marketing Science,* 2, 25-34.
HOGARTH, K. 2015. Analysis of the Cape Town Housing Market: Supply, Demand and Housing Submarkets. *In:* DEPARTMENT, S. P. A. U. D. (ed.). City of Cape Town.
HORNA, J. 1994. *The study of leisure: an introduction*, Oxford University Press.
ISLAM, K. S. & ASAMI, Y. 2009. Housing market segmentation: A review. *Review of Urban & Regional Development Studies,* 21, 93-109.
KARA, A. & KAYNAK, E. 1997. Markets of a single customer: exploiting conceptual developments in market segmentation. *European journal of marketing,* 31, 873-895.
KIM, J.-C., KIM, D.-H., KIM, J.-J., YE, J.-S. & LEE, H.-S. 2000. Segmenting the Korean housing market using multiple discriminant analysis. *Construction Management & Economics,* 18, 45-54.
KORT, F. 1973. Regression analysis and discriminant analysis: An application of RA Fisher's theorem to data in political science. *American Political Science Review,* 67, 555-559.
KOZIOL, N. & ARTHUR, A. 2011. An introduction to secondary data analysis. *Research Methodology Series*.
KRAMMER, K. & SINGER, Y. 2001. On the algorithmic implementation of multi-class SVMs. *Proc. of JMLR*, 265-292.
KUNH, T. 1962. The structure of scientific revolutions. Chicago: The University of Chicago Press.
LANCASTER, K. J. 1966. A new approach to consumer theory. *Journal of political economy,* 74, 132-157.
LANCASTER, K. J. 1990. Modern consumer theory. *Books*.
LANDIS, J. R. & KOCH, G. G. 1977. The measurement of observer agreement for categorical data. *biometrics*, 159-174.
LEMANSKI, C. 2007. Global cities in the South: deepening social and spatial polarisation in Cape Town. *Cities,* 24, 448-461.
LIAW, A. & WIENER, M. 2002. Classification and regression by randomForest. *R news,* 2, 18-22.

MCCLURE, K. 2005. Deciphering the Need in Housing Markets A Technique to Identify Appropriate Housing Policies at the Local Level. *Journal of Planning Education and Research,* 24**,** 361-378.
MCGAFFIN, R. Understanding the Fundamentals of Real Estate Market Demand. Presentation to the South Africa National Treasury Cities' Support Program (CSP) Training Session and National Forum: Leveraging Real Estate Value to Catalyze Urban (re)Development. Johannesburg, 3- 5 November, 2014.
MCGAFFIN, R. & KIROVA, M. 2016. A way to understanding housing markets beyond "Subsidy, Gap and Market". *IHS Conference 2016.* UCT-Nedbank Urban Real Estate Research Unit.
MICHALSKI, R. S., CARBONELL, J. G. & MITCHELL, T. M. 2013. *Machine learning: An artificial intelligence approach*, Springer Science & Business Media.
MILGROM, P. R., LEVIN, J. & EILAT, A. 2011. The case for unlicensed spectrum. *Available at SSRN 1948257*.
MTANTO, S. & CHURR, N. 2014. Submission for the Division of Revenue. *Understanding Housing Demand in South Africa.* Financial and Fiscal Commission.
MUELLER, J. P. & MASSARON, L. 2017. *3 Types of Machine Learning* [Online]. Available: http://www.dummies.com/programming/big-data/data-science/3-types-machine-learning/ [Accessed 2017-07-02].
NAU, D. S. 1995. Mixing methodologies: can bimodal research be a viable post-positivist tool? *The Qualitative Report,* 2**,** 1-6.
NDZIBA, N., LENDOR, B. & OERTEL, M. 2015. *The Propensity of Different Households to Demand Particular Dwelling Types in Cape Town between 2011 and 2040.* Bachelor of Science (Honours) in Property Studies, University of Cape Town.
OTTAWA, C. O. 2007. Growth Projections for Ottawa - Prospects for Population, Housing and Jobs 2006-2031. *In:* DEPARTMENT OF PLANNING, T. A. T. E. (ed.). City of Ottawa.
POLARIS. 2008. *Take a Look at Your Customer Clusters* [Online]. Polaris Marketing Research. Available: http://cdn2.hubspot.net/hub/58820/file-15751967-htm/newsletters/mr_best_practices/mrp_0408_segmentation_analysis.htm [Accessed 2017-04-01].
POPPER, K. R. 1959. The propensity interpretation of probability. *The British journal for the philosophy of science,* 10**,** 25-42.
QUIGLEY, J. M. 1976. Housing demand in the short run: An analysis of polytomous choice. *Explorations in Economic Research, Volume 3, number 1.* NBER.
RAILEANU, L. E. & STOFFEL, K. 2004. Theoretical comparison between the gini index and information gain criteria. *Annals of Mathematics and Artificial Intelligence,* 41**,** 77-93.
RAJASEKAR, S., PHILOMINATHAN, P. & CHINNATHAMBI, V. 2006. Research methodology. *arXiv preprint physics/0601009*.
RASCHKA, S. 2016. When Does Deep Learning Work Better Than SVMs or Random Forests? Available from: http://www.kdnuggets.com/2016/04/deep-learning-vs-svm-random-forest.html [Accessed 19 March 2017].
RAY, S. 2015a. *Essentials of Machine Learning Algorithms with Python and R Codes* [Online]. Analytics Vidhya. Available: https://www.analyticsvidhya.com/blog/2015/08/common-machine-learning-algorithms/ [Accessed 2017-07-01].

RIPLEY, B. D. 1996. Pattern recognition via neural networks. *a volume of Oxford Graduate Lectures on Neural Networks, title to be decided. Oxford University Press.[See http://www. stats. ox. ac. uk/ripley/papers. html.]*.
SHOTTON, J., SHARP, T., KIPMAN, A., FITZGIBBON, A., FINOCCHIO, M., BLAKE, A., COOK, M. & MOORE, R. 2013. Real-time human pose recognition in parts from single depth images. *Communications of the ACM,* 56**,** 116-124.
SMITH, W. R. 1956. Product differentiation and market segmentation as alternative marketing strategies. *The Journal of Marketing***,** 3-8.
STATISTICS SOUTH AFRICA 1996. Census 1996. *In:* STATISTICS SOUTH AFRICA (ed.). statssa.gov.za.
STATISTICS SOUTH AFRICA. 2011. *Census 2011 Statistics by Place | Statistics South Africa* [Online]. statssa.gov.za. Available: http://www.statssa.gov.za/?page_id=964 [Accessed 06 Nov 2016].
STATISTICS SOUTH AFRICA 2012. Census 2011 Statistical Release (Revised). *In:* AFRICA, S. S. (ed.).
SWANSON, R. A. & CHERMACK, T. J. 2013. *Theory building in applied disciplines*, Berrett-Koehler Publishers.
THOMAS, P. Y. 2010. *Towards developing a web-based blended learning environment at the University of Botswana.*
TORRALBA, A., MURPHY, K. P. & FREEMAN, W. T. 2007. Sharing visual features for multiclass and multiview object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 29**,** 854-869.
UN-HABITAT 2011. A practical guide for conducting: housing profiles: Supporting evidence-based housing policy and reform. *United Nations Human Settlements Programme (UN-HABITAT)*.
UPASANA. 2017. *How to handle Imbalanced Classification Problems in machine learning?* [Online]. Analytics Vidhya. Available: https://www.analyticsvidhya.com/blog/2017/03/imbalanced-classification-problem/ [Accessed 2018-04-07].
VIRULY, F. 2015. Urban Economics and Real Estate Markets: An Emperging Country Perspective. Unpublished.
VOSLOO, J. J. 2014. *A sport management programme for educator training in accordance with the diverse needs of South african schools.*
WATKINS, C. A. 2001. The definition and identification of housing submarkets. *Environment and Planning A,* 33**,** 2235-2253.
WCG 2015. A Human Settlement Demand Study in the Western Cape.
WEDEL, M. & KAMAKURA, W. A. 2012. *Market segmentation: Conceptual and methodological foundations*, Springer Science & Business Media.
WIND, Y. 1978. Issues and advances in segmentation research. *Journal of marketing research***,** 317-337.
YIN, R. K. 1994. Case Study Research: Design and Methods (Applied Social Research Methods, Vol. 5). *Sage Publications, Beverly Hills, CA. Rick Rantz Leading urban institutions of higher education in the new millennium Leadership & Organization Development Journal,* 23**,** 2002.
ZHOU, Z.-H. 2012. *Ensemble methods: foundations and algorithms*, CRC press.

Annexure A

Census 2011 Questionnaire

Statistics South Africa

Preferred supplier of quality statistics

HOUSEHOLD QUESTIONNAIRE A

FOR STATISTICAL USE ONLY

CENSUS 2011 You Count!

STATISTICS ACT NO. 6 OF 1999 (CONFIDENTIALITY)

17(1) Despite any other law, no return or other information collected by Statistics South Africa for the purposes of official or other statistics that relates to an individual or a household may be disclosed to any person.

17(3b) Any person who is involved in the collection of, or who may use, that information or data, must first take an oath of confidentiality.

18(1e) Any officer of Statistics South Africa who willfully discloses any data or

18(1g) information obtained in the course of such employment to a person not authorised to receive that information is guilty of an offence and liable on conviction to a fine not exceeding R10 000, or to imprisonment for a period not exceeding 6 months or to both.

FOR OFFICE USE ONLY

ENUMERATION AREA NUMBER

Province ..... Local municipality .....

Main place ..... Sub-place .....

Physical identification of the dwelling unit .....

.....

Postal code Landline/Cell phone of enumerated household .....

PARTICULARS OF THE HOUSEHOLD

Dwelling unit number

Household number

Total number of households at this dwelling

Map reference number

Listing record number

Total number of persons in the household — Males Females Total

Questionnaire of completed for this household

If more than one questionnaire is used in the household, write the barcode of the 1st questionnaire below

METHOD OF QUESTIONNAIRE COMPLETION - *Mark the appropriate circle with an X*

A fieldworker through an interview ○ A household member through self-completion ○

FIELD STAFF

Fieldworker ID No. Supervisor ID No.

Signature ..... Signature .....

RESPONSE DETAILS

		Interview			Next Visit (Planned)	
Visit No.	Date (actual)	Start Time	End Time	Result Code	Date	Time
1						
2						
3						
4						

RESULT CODE	RESPONSE DETAILS
11	Completed
12	Partly completed
21	Non-contact
22	Refusal
31	Unoccupied
32	Vacant
33	Demolished
34	New dwelling under construction

Comments and full details of all non-response / unusual circumstances

.....

.....

.....

FINAL RESULT CODE

SHOULD YOU ENCOUNTER ANY DIFFICULTIES IN THE COMPLETION OF THE QUESTIONNAIRE, PLEASE CONTACT:

..... ON .....

OR PHONE THE CENSUS HOTLINE, TOLL FREE, ON 0800 110 248

X-123456789

A0C

Proof Version A6 - 02DEC2010 9.30AM

PROCEDURES OF ENUMERATION

How to complete the questionnaire

- ***Read*** *every question carefully*
- *Make sure that all the codes are written inside the boxes.*
  For example: [3] *Correct* [3] *Incorrect*
- *For numeric values, such as age, person number, number of children, the enumerator/respondent should write the correct answer in the box and include leading zeros. For example:* [0][0][7]
- *For open-ended questions, the enumerator/respondent should write legibly in CAPITAL LETTERS in the boxes provided with no spaces between the words. For example Cape Town should be written as:* [C][A][P][E][T][O][W][N]
- *Do not write zeros in boxes where questions are not applicable*

What to use when completing this questionnaire?

Use only a pencil. If you make a mistake, use a soft rubber to erase the mistake and write the correct answer.

Who should be the respondent?

- *The head/acting head of the household.*
- *In the absence of head/acting head, any responsible adult member left in charge of the household.*

Note:

A household *is a group of persons who live together, and provide themselves jointly with food or other essentials for living, or a single person who lives alone.*

- *Domestic workers are counted as a separate household even if they live in the same dwelling as the employer.*

Who should be counted in this questionnaire?

- *All persons present in the household on the reference night (midnight 9-10 October 2011)*
- *Include babies born before the reference night as well as visitors.*
- *Members who died after the reference night must be counted as alive.*
- *Members of the household who were absent overnight, for example working, travelling or at an entertainment venue, religious gathering, if they returned to the household the next day.*
- *Individuals in converted hostels, residential hotels and old age homes (depending on arrangement).*

X-123456789

CENSUS 2011 DRESS REHEARSAL X-123456789

CENSUS 2011 DRESS REHEARSAL X-123456789

CENSUS 2011 DRESS REHEARSAL X-123456789

FLAP: PARTICULARS OF ALL INDIVIDUALS

Please write the name and surname of the household head and first names of every person who was present in this household on the census night (midnight 9-10 October 2011)
One name on each row. Start with head or acting head of household.

The head or acting head is the person who is the main decision-maker of the household. If people are equally decision-makers, then take the oldest person as the household head.
For babies with no name, write BABY.

Please include babies, small children, old people and visitors who were present in this household on the census night (9-10 October 2011)

F-00 PERSON NUMBER	**F-01** PERSON NAME	**F-02** AGE IN COMPLETED YEARS	**F-03** SEX 1 = Male 2 = Female
Write 0 or 1 in the first box for all persons listed on the flap Example: Row 1 0 1 Row 10 1 0	**Example** J O H N M A L U L E K E	**Example 1** 0 3 1 **Example 2** Child less than 1year 0 0 0	**Example** X 1 Male 2 Female *Mark the appropriate circle with an X.*
1			1 Male 2 Female
2			1 Male 2 Female
3			1 Male 2 Female
4			1 Male 2 Female
5			1 Male 2 Female
6			1 Male 2 Female
7			1 Male 2 Female
8			1 Male 2 Female
9			1 Male 2 Female
0			1 Male 2 Female

A0F

SECTION A: DEMOGRAPHICS - ASK OF EVERYONE LISTED ON THE FLAP

P-01 DATE OF BIRTH	**P-02** RELATIONSHIP	**P-03** MARITAL STATUS	**P-04** SPOUSE OR PARTNER	**P-05** POPULATION GROUP	**P-06** LANGUAGE
What is *(name's)* **date of birth?** **Example** 1 9 0 4 1 9 7 9	**What is** *(name's)* **relationship to the head or acting head of the household?** *The head or acting head is the person listed in row 1 of the first questionnaire, if more than one questionnaire has been completed for this household.* 01 = Head/Acting Head 02 = Husband/Wife/Partner 03 = Son/Daughter 04 = Adopted Son/Daughter 05 = Stepchild 06 = Brother/Sister 07 = Parent (Mother/Father) 08 = Parent-in-law 09 = Grand/Great Grandchild 10 = Son/Daughter-in-law 11 = Brother/Sister-in-law 12 = Grandmother/Father 13 = Other relative 14 = Non-related person *Write the appropriate code in the boxes.*	**What is** *(name's)* **PRESENT marital status?** 1 =Married 2 =Living together like married partners 3 =Never married 4 =Widower/ widow 5 =Separated 6 =Divorced *Write the appropriate code in the box.* ***If 3-6, Go to P-05***	**Who in this house-hold is** *(name's)* **spouse or partner?** *Write the* ***person number*** *of the spouse or partner in the appropriate boxes. If the spouse or partner does not reside in the house-hold,* ***write 98****.* ***Note:*** *Refer to person on flap e.g. 02*	**How would** *(name)* **describe him/herself in terms of population group?** 1 = Black African 2 = Coloured 3 = Indian or Asian 4 = White 5 = Other *Write the appropriate code in the box.*	**Which two languages does** *(name)* **speak most often in this household?** 01 = Afrikaans 02 = English 03 = IsiNdebele 04 = IsiXhosa 05 = IsiZulu 06 = Sepedi 07 = Sesotho 08 = Setswana 09 = SiSwati 10 = Tshivenda 11 = Xitsonga 12 = Sign language 13 = Other *Write the appropriate code in the boxes.* *If no other language, write 00 in the second box.*
D D M M Y Y Y Y					**First** **Second**
D D M M Y Y Y Y					**First** **Second**
D D M M Y Y Y Y					**First** **Second**
D D M M Y Y Y Y					**First** **Second**
D D M M Y Y Y Y					**First** **Second**
D D M M Y Y Y Y					**First** **Second**
D D M M Y Y Y Y					**First** **Second**
D D M M Y Y Y Y					**First** **Second**
D D M M Y Y Y Y					**First** **Second**
D D M M Y Y Y Y					**First** **Second**

Census 2011 - A© Statistics South Africa, November 2010

A01

SECTION B: MIGRATION - ASK OF EVERYONE LISTED ON THE FLAP

P-07 PROVINCE OF BIRTH	**P-08** COUNTRY OF BIRTH	**P-08a** YEAR MOVED TO SOUTH AFRICA	**P-09** SOUTH AFRICAN CITIZENSHIP	**P-10** USUAL RESIDENCE	**P-10a** PROVINCE OF USUAL RESIDENCE	**P-10b** MUNICIPALITY/ MAGISTERIAL DISTRICT OF USUAL RESIDENCE
In which province was *(name)* **born?** 01 = Western Cape 02 = Eastern Cape 03 = Northern Cape 04 = Free State 05 = Kwa-Zulu Natal 06 = North West 07 = Gauteng 08 = Mpumalanga 09 = Limpopo 10 = Outside South Africa 11 = Do not know *Write the appropriate code in the boxes.* ***If 01-09 or 11, Go to P-09***	**In which country was** *(name)* **born?** *Use CAPITAL LETTERS only* *Examples: NEWZEALAND, BOTSWANA, SIERRALEONE*	**In which year did** *(name)* **move to South Africa?** *If moved more than once into South Africa, please indicate the year of last move.* **Example** 1 9 9 8	**Is** *(name)* **a South African citizen?** 1 = Yes 2 = No *Mark the appropriate circle with an X.*	**Does** *(name)* **usually live in this household for at least four nights a week and has done so for the last six months? OR intends to live in this household for at least four nights a week for the next six months?** 1 = Yes 2 = No *Mark the appropriate circle with an X.* ***If 1, Go to P-11***	**In which province does** *(name)* **usually live?** 01 = Western Cape 02 = Eastern Cape 03 = Northern Cape 04 = Free State 05 = Kwa-Zulu Natal 06 = North West 07 = Gauteng 08 = Mpumalanga 09 = Limpopo 10 = Outside South Africa 11 = Do not know *Write the appropriate code in the boxes.* ***If 10, Go to P-11***	**In which** <u>**municipality**</u> **or magisterial district does** *(name)* **usually live?** *Use CAPITAL LETTERS only* **Example** J O B U R G M E T R O
		Y Y Y Y	1 Yes 2 No	1 Yes 2 No		
		Y Y Y Y	1 Yes 2 No	1 Yes 2 No		
		Y Y Y Y	1 Yes 2 No	1 Yes 2 No		
		Y Y Y Y	1 Yes 2 No	1 Yes 2 No		
		Y Y Y Y	1 Yes 2 No	1 Yes 2 No		
		Y Y Y Y	1 Yes 2 No	1 Yes 2 No		
		Y Y Y Y	1 Yes 2 No	1 Yes 2 No		
		Y Y Y Y	1 Yes 2 No	1 Yes 2 No		
		Y Y Y Y	1 Yes 2 No	1 Yes 2 No		
		Y Y Y Y	1 Yes 2 No	1 Yes 2 No		

SECTION B: MIGRATION (Continued)

P-10c CITY/TOWN OF USUAL RESIDENCE	**P-11** SINCE 2001	**P-11a** MONTH AND YEAR MOVED	**P-11b** PROVINCE OF PREVIOUS RESIDENCE	**P-11c** MUNICIPALITY/ MAGISTERIAL DISTRICT OF PREVIOUS RESIDENCE	**P-11d** CITY/TOWN OF PREVIOUS RESIDENCE
In which city/town does *(name)* **usually live or what is the nearest city/town?** *Use CAPITAL LETTERS only*	**Has** *(name)* **been living in this place since October 2001?** 1 = Yes 2 = No 3 = Born after October 2001 but never moved 4 = Born after October 2001 and moved *Write the appropriate code in the box.* ***If 1 or 3, Go to P-12***	**When did** *(name)* **move to this place?** *Write the month and year in the appropriate boxes.* **Example** 0 4 2 0 0 2	**In which province did** *(name)* **live before moving to this place?** 01 = Western Cape 02 = Eastern Cape 03 = Northern Cape 04 = Free State 05 = Kwa-Zulu Natal 06 = North West 07 = Gauteng 08 = Mpumalanga 09 = Limpopo 10 = Outside South Africa 11 = Do not know *Write the appropriate code in the boxes.* ***If 10, Go to P-12***	**In which municipality or magisterial district did** *(name)* **live before moving to this place?** *Use CAPITAL LETTERS only* **Example** J O B U R G M E T R O	**In which city/town did** *(name)* **live before or what was the nearest city/town?** *Use CAPITAL LETTERS only* **Example** J O H A N N E S B U R G
		M M Y Y Y Y			
		M M Y Y Y Y			
		M M Y Y Y Y			
		M M Y Y Y Y			
		M M Y Y Y Y			
		M M Y Y Y Y			
		M M Y Y Y Y			
		M M Y Y Y Y			

A03

SECTION C: GENERAL HEALTH AND FUNCTIONING - ASK OF EVERYONE LISTED ON THE FLAP

SECTION D: PARENTAL SURVIVAL AND INCOME - ASK OF EVERYONE LISTED ON THE FLAP

P-12 HEALTH AND FUNCTIONING	**P-13** ASSISTIVE DEVICES AND MEDICATION	**P-14** MOTHER ALIVE	**P-14a** MOTHER PERSON NUMBER	**P-15** FATHER ALIVE
Does *(name)* **have difficulty in the following?** A = Seeing even when using eye glasses? B = Hearing even when using a hearing aid? C = Communicating in his/her language (i.e. understanding others or being understood by others)? D = Walking or climbing stairs? E = Remembering or concentrating? F = With self-care such as washing all over, dressing or feeding? 1 = No difficulty 2 = Some difficulty 3 = A lot of difficulty 4 = Cannot do at all 5 = Do not know 6 = Cannot yet be determined *Write the appropriate code in the box.*	**Does** *(name)* **use any of the following?** A = Eye glasses B = Hearing aid C = Walking stick or frame D = A wheelchair E = Chronic medication 1 = Yes 2 = No 3 = Do not know *Write the appropriate code in the box.*	**Is** *(name's)* **own biological mother still alive?** 1 = Yes 2 = No 3 = Do not know *Mark the appropriate circle with an X.* ***If 2-3, Go to P-15***	**Who in this household is** *(name's)* **biological mother?** *If the person's mother does not reside in the household (not listed on the flap), write 98.* ***Note:*** *Refer to person number on flap e.g. 02*	**Is** *(name's)* **own biological father still alive?** 1 = Yes 2 = No 3 = Do not know *Mark the appropriate circle with an X.* ***If 2-3, Go to P-16***
Seeing (A) / Walking / Climbing (D) Hearing (B) / Remembering / Concentrating (E) Communicating (C) / Self-care (F)	Glasses (A) / Wheelchair (D) Hearing aid (B) / Chronic medication (E) Walking stick / frame (C)	1 Yes 2 No 3 Do not know		1 Yes 2 No 3 Do not know
Seeing (A) / Walking / Climbing (D) Hearing (B) / Remembering / Concentrating (E) Communicating (C) / Self-care (F)	Glasses (A) / Wheelchair (D) Hearing aid (B) / Chronic medication (E) Walking stick / frame (C)	1 Yes 2 No 3 Do not know		1 Yes 2 No 3 Do not know
Seeing (A) / Walking / Climbing (D) Hearing (B) / Remembering / Concentrating (E) Communicating (C) / Self-care (F)	Glasses (A) / Wheelchair (D) Hearing aid (B) / Chronic medication (E) Walking stick / frame (C)	1 Yes 2 No 3 Do not know		1 Yes 2 No 3 Do not know
Seeing (A) / Walking / Climbing (D) Hearing (B) / Remembering / Concentrating (E) Communicating (C) / Self-care (F)	Glasses (A) / Wheelchair (D) Hearing aid (B) / Chronic medication (E) Walking stick / frame (C)	1 Yes 2 No 3 Do not know		1 Yes 2 No 3 Do not know
Seeing (A) / Walking / Climbing (D) Hearing (B) / Remembering / Concentrating (E) Communicating (C) / Self-care (F)	Glasses (A) / Wheelchair (D) Hearing aid (B) / Chronic medication (E) Walking stick / frame (C)	1 Yes 2 No 3 Do not know		1 Yes 2 No 3 Do not know
Seeing (A) / Walking / Climbing (D) Hearing (B) / Remembering / Concentrating (E) Communicating (C) / Self-care (F)	Glasses (A) / Wheelchair (D) Hearing aid (B) / Chronic medication (E) Walking stick / frame (C)	1 Yes 2 No 3 Do not know		1 Yes 2 No 3 Do not know
Seeing (A) / Walking / Climbing (D) Hearing (B) / Remembering / Concentrating (E) Communicating (C) / Self-care (F)	Glasses (A) / Wheelchair (D) Hearing aid (B) / Chronic medication (E) Walking stick / frame (C)	1 Yes 2 No 3 Do not know		1 Yes 2 No 3 Do not know
Seeing (A) / Walking / Climbing (D) Hearing (B) / Remembering / Concentrating (E) Communicating (C) / Self-care (F)	Glasses (A) / Wheelchair (D) Hearing aid (B) / Chronic medication (E) Walking stick / frame (C)	1 Yes 2 No 3 Do not know		1 Yes 2 No 3 Do not know
Seeing (A) / Walking / Climbing (D) Hearing (B) / Remembering / Concentrating (E) Communicating (C) / Self-care (F)	Glasses (A) / Wheelchair (D) Hearing aid (B) / Chronic medication (E) Walking stick / frame (C)	1 Yes 2 No 3 Do not know		1 Yes 2 No 3 Do not know
Seeing (A) / Walking / Climbing (D) Hearing (B) / Remembering / Concentrating (E) Communicating (C) / Self-care (F)	Glasses (A) / Wheelchair (D) Hearing aid (B) / Chronic medication (E) Walking stick / frame (C)	1 Yes 2 No 3 Do not know		1 Yes 2 No 3 Do not know

X-123456789

SECTION D: PARENTAL SURVIVAL AND INCOME (Continued)

SECTION E: EDUCATION - ASK OF ALL PERSONS AGED 5 YEARS AND OLDER LISTED ON THE FLAP

P-15a FATHER PERSON NUMBER	**P-16** INCOME CATEGORY	**P-17** SCHOOL ATTENDANCE	**P-18** EDUCATIONAL INSTITUTION	**P-19** PUBLIC OR PRIVATE
Who in this household is *(name's)* **biological father?** *If the person's father does not reside in the household (not listed on the flap), write 98.* ***Note:*** *Refer to person number on flap e.g. 02*	**What is the income category that best describes the gross monthly or annual income of** (*name*) **before deductions and including all sources of income?** **Monthly** / **Annual** 01 = No income / No income 02 = R1 - R400 / R1 - R4 800 03 = R401 - R800 / R4 801 – R9 600 04 = R801 – R1 600 / R9 601 – R19 200 05 = R1 601 – R3 200 / R19 201 – R38 400 06 = R3 201 – R6 400 / R38 401 – R76 800 07 = R6 401 – R12 800 / R76 801 – R153 600 08 = R12 801 – R25 600 / R153 601 – R307 200 09 = R25 601 – R51 200 / R307 201 – R614 400 10 = R51 201 – R102 400 / R614 401 – R1 228 800 11 = R102 401 – R204 800 / R1 228 801 – R2 457 600 12 = R204 801 or more / R2 457 601 or more *Gross income should include **all sources of income** e.g. Social grants, UIF, remittances, rentals, investments, sales or products, services, etc.*	**Does** *(name)* **presently attend an educational institution?** 1 = Yes 2 = No 3 = Do not know *Mark the appropriate circle with an X.* *Attendance includes all part-time and full-time studies, whether in person or as a distance learner.* ***If 2-3, Go to P-20***	**Which of the following educational institutions does** *(name)* **attend?** 1 = Pre-school (including day care, crèche, Grade R and Pre-Grade R in an ECD centre) 2 = Ordinary school (including Grade R learners who attend a formal school, Grade 1-12 learners & learners in special class) 3 = Special school 4 = Further Education and Training College (FET) 5 = Other College 6 = Higher Educational Institution (University/University of Technology) 7 = Adult Basic Education and Training Centre (ABET Centre) 8 = Literacy classes (e.g. Kha Ri Gude, SANLI) 9 = Home based education/home schooling *Write the appropriate code in the box.*	**Is the institution that** *(name)* **is attending public or private?** 1 = Public (Government) 2 = Private (Independent) 3 = Do not know *Mark the appropriate circle with an X.*
		1 Yes 2 No 3 Do not know		1 Public 2 Private 3 Do not know
		1 Yes 2 No 3 Do not know		1 Public 2 Private 3 Do not know
		1 Yes 2 No 3 Do not know		1 Public 2 Private 3 Do not know
		1 Yes 2 No 3 Do not know		1 Public 2 Private 3 Do not know
		1 Yes 2 No 3 Do not know		1 Public 2 Private 3 Do not know
		1 Yes 2 No 3 Do not know		1 Public 2 Private 3 Do not know
		1 Yes 2 No 3 Do not know		1 Public 2 Private 3 Do not know
		1 Yes 2 No 3 Do not know		1 Public 2 Private 3 Do not know
		1 Yes 2 No 3 Do not know		1 Public 2 Private 3 Do not know
		1 Yes 2 No 3 Do not know		1 Public 2 Private 3 Do not know

SECTION E: EDUCATION (Continued)

P-20 LEVEL OF EDUCATION

What is the highest level of education that *(name)* has completed?

98 = No schooling
00 = Grade 0
01 = Grade 1/Sub A
02 = Grade 2/Sub B
03 = Grade 3/Std 1/ABET 1 (Kha Ri Gude, SANLI)
04 = Grade 4/Std 2
05 = Grade 5/Std 3 / ABET 2
06 = Grade 6/Std 4
07 = Grade 7/Std 5 / ABET 3

If 98 or 00-07, Go to P-22

08 = Grade 8/Std 6 / Form 1
09 = Grade 9/Std 7/Form 2/ ABET 4
10 = Grade 10/Std 8/Form 3
11 = Grade 11/Std 9/Form 4
12 = Grade 12/Std 10 /Form 5

If 08-12, Go to P-23

13 = NTC I/N1/ NIC/(V) Level 2
14 = NTCII/N2/ NIC/(V) Level 3
15 = NTCIII/N3/NIC/(V) Level 4
16 = N4/NTC 4
17 = N5/NTC 5
18 = N6/NTC 6
19 = Certificate with less than Grade 12 /Std 10
20 = Diploma with less than Grade 12/Std 10
21 = Certificate with Grade 12/Std 10
22 = Diploma with Grade 12/Std 10
23 = Higher Diploma
24 = Post Higher Diploma (Masters, Doctoral Diploma)
25 = Bachelors degree
26 = Bachelors degree and Post graduate diploma
27 = Honours degree
28 = Higher degree (Masters/PhD)
29 = Other

If 13-28, Go to P-21

If 29, Go to P-22

READ OUT: Diploma or certificate should have been at least six months study duration full-time (or equivalent).

Write the appropriate code in the boxes.

P-21 FIELD OF EDUCATION

In which field is *(name's)* highest post-school qualification?

UNIVERSITY/TECHNIKON/COLLEGE

01 = Agriculture or Renewable Natural Resources
02 = Architecture or Environmental Design
03 = Arts, Visual or Performing
04 = Business, Commerce or Management Sciences
05 = Communication
06 = Computer Sciences
07 = Education, Training or Development
08 = Engineering or Engineering Technology
09 = Health Care or Health Sciences
10 = Home Economics
11 = Industrial Arts, Traders or Technology
12 = Languages, Linguistics or Literature
13 = Law
14 = Libraries or Museums
15 = Life Sciences or Physical Sciences
16 = Mathematical Sciences
17 = Military Sciences
18 = Philosophy, Religion or Theology
19 = Physical Education or Leisure
20 = Psychology
21 = Public Administration or Social Services
22 = Social Sciences or Social Studies
23 = Other

FURTHER EDUCATION AND TRAINING (FET)

24 = Management
25 = Marketing
26 = Information Technology and Computer Science
27 = Finance, Economics and Accounting
28 = Office Administration
29 = Electrical Infrastructure Construction
30 = Civil Engineering and Building Construction
31 = Engineering
32 = Primary Agriculture
33 = Hospitality
34 = Tourism
35 = Safety in society
36 = Mechatronics
37 = Education and Development
38 = Other

Write the appropriate code in the boxes.

Any response, Go to P-23

X-123456789

A06

SECTION E: EDUCATION (Continued)

SECTION F: EMPLOYMENT - ASK OF ALL PERSONS AGED 15 YEARS AND OLDER LISTED ON THE FLAP

P-22 LITERACY	**P-23** EMPLOYMENT STATUS *(Answer all three questions and then follow the skip instruction below)*		
Does *(name)* **have difficulty in doing any of the following?** A = Writing his/her name B = Reading (e.g. newspapers, magazines, religious books etc) in any language C = Filling in a form (e.g. social grants forms) D = Writing a letter in any language E = Calculating/working out how much change he/she should receive when buying something F = Reading road signs 1 = No difficulty 2 = Some difficulty 3 = A lot of difficulty 4 = Unable to do 5 = Do not know *Write the code in the appropriate box.*	**In the SEVEN DAYS before 10 October …** **P-23a** **Did** *(name)* **work for a wage, salary, commission or any payment in kind (including paid domestic work), even if it was for only one hour?** 1 = Yes 2 = No 3 = Do not know *Mark the appropriate circle with an X.*	**In the SEVEN DAYS before 10 October …** **P-23b** **Did** *(name)* **run or do any kind of business, big or small, for herself/himself or with one or more partners, even if it was for only one hour?** 1 = Yes 2 = No 3 = Do not know *Mark the appropriate circle with an X.*	**In the SEVEN DAYS before 10 October …** **P-23c** **Did** *(name)* **help without being paid in any kind of business run by her/his household, even if it was for only one hour?** 1 = Yes 2 = No 3 = Do not know *Mark the appropriate circle with an X.*
	If 1 (Yes) to any of P-23a, P-23b or P-23c, Go to P-29a		
Writing his/her name (A) Writing a letter (D) Reading (B) Calculating (E) Filling a form (C) Reading road signs (F)	1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know
Writing his/her name (A) Writing a letter (D) Reading (B) Calculating (E) Filling a form (C) Reading road signs (F)	1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know
Writing his/her name (A) Writing a letter (D) Reading (B) Calculating (E) Filling a form (C) Reading road signs (F)	1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know
Writing his/her name (A) Writing a letter (D) Reading (B) Calculating (E) Filling a form (C) Reading road signs (F)	1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know
Writing his/her name (A) Writing a letter (D) Reading (B) Calculating (E) Filling a form (C) Reading road signs (F)	1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know
Writing his/her name (A) Writing a letter (D) Reading (B) Calculating (E) Filling a form (C) Reading road signs (F)	1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know
Writing his/her name (A) Writing a letter (D) Reading (B) Calculating (E) Filling a form (C) Reading road signs (F)	1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know
Writing his/her name (A) Writing a letter (D) Reading (B) Calculating (E) Filling a form (C) Reading road signs (F)	1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know
Writing his/her name (A) Writing a letter (D) Reading (B) Calculating (E) Filling a form (C) Reading road signs (F)	1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know
Writing his/her name (A) Writing a letter (D) Reading (B) Calculating (E) Filling a form (C) Reading road signs (F)	1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know

SECTION F: EMPLOYMENT (Continued)

P-24 TEMPORARY ABSENCE FROM WORK	**P-25** LOOKING FOR WORK	**P-26** LIKED TO WORK	**P-27** REASONS FOR NOT WORKING	**P-28** AVAILABLE TO WORK
Even though *(name)* **did not do any work for pay, profit or did not help without pay in a household business in the SEVEN DAYS before 10 October, did he/she have a paid job or business that he/she would definitely return to?** 1 = Yes 2 = No 3 = Do not know *Mark the appropriate circle with an X.* ***If 1, Go to P-29a***	**In the four weeks before 10 October was** *(name)* **looking for any kind of job or trying to start any kind of business?** 1 = Yes 2 = No 3 = Do not know *Mark the appropriate circle with an X.* ***If 1, Go to P-28***	**Would** *(name)* **have liked to work in the SEVEN DAYS before 10 October?** 1 = Yes 2 = No 3 = Do not know *Mark the appropriate circle with an X.* ***If 2 or 3, Go to P-32***	**What was the main reason for not trying to find work or starting a business in the last four weeks before 10 October?** 01 = Awaiting the season for work 02 = Waiting to be recalled to former job 03 = Health reasons 04 = Pregnancy 05 = Disabled or unable to work (handicapped) 06 = Housewife/homemaker (family considerations/child care) 07 = Undergoing training to help find work 08 = No jobs available in the area 09 = Lack of money to pay for transport to look for work 10 = Unable to find work requiring his/her skills 11 = Lost hope of finding any kind of work 12 = No transport available 13 = Scholar or student 14 = Retired 15 = Too old/young to work 16 = Did not want to work 17 = Other *Write the appropriate code in the boxes.*	**If a suitable job had been offered or circumstances had allowed, would** *(name)* **have been able to start work or a business in the SEVEN DAYS before 10 October?** 1 = Yes 2 = No 3 = Do not know *Mark the appropriate circle with an X.* ***Any response, Go to P-32***
1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know		1 Yes 2 No 3 Do not know
1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know		1 Yes 2 No 3 Do not know
1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know		1 Yes 2 No 3 Do not know
1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know		1 Yes 2 No 3 Do not know
1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know		1 Yes 2 No 3 Do not know
1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know		1 Yes 2 No 3 Do not know
1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know		1 Yes 2 No 3 Do not know
1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know		1 Yes 2 No 3 Do not know
1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know		1 Yes 2 No 3 Do not know
1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know		1 Yes 2 No 3 Do not know

SECTION F: EMPLOYMENT (Continued)

P-29a INDUSTRY	**P-29b** MAIN GOODS OR SERVICES	**P-30a** OCCUPATION	**P-30b** MAIN TASK/DUTY	**P-31** TYPE OF SECTOR
What is the name of *(name's)* **place of work/organisation/ company/business?** *Examples: KOMANIHOSPITAL, RAPELEPRIMARYSCHOOL, HARMONYGOLDMINING* *Write OWNHOUSE or NOFIXEDLOCATION, if relevant* *Use CAPITAL LETTERS only*	**What are the main goods or services produced at** *(name's)* **place of work or its main functions?** *Examples: REALESTATE, CONSTRUCTION, CARREPAIRING, HOSPITALITYSERVICES* *For domestic workers, write PRIVATEHOUSEHOLD* *Use CAPITAL LETTERS only*	**What kind of work does** *(name)* **usually do in his/her main job/business?** *Examples:* *PRIMARYSCHOOLTEACHER, BUSINESSOWNER, OFFICECLEANER* *Use CAPITAL LETTERS only*	**What is** *(name's)* **main task or duty in this work?** *Examples: TEACHINGCHILDREN, SELLINGFRUIT, BOOKKEEPING, FEEDINGCATTLE* *Use CAPITAL LETTERS only*	**Is** *(name's)* **place of work .....?** 1 = In the formal sector 2 = In the informal sector 3 = Private household 4 = Do not know *Write the appropriate code in the box.*

A09

SECTION G: FERTILITY - ASK OF WOMEN AGED 12-50 YEARS LISTED ON THE FLAP

P-32 CHILDREN EVER BORN	**P-33** AGE AT FIRST BIRTH	**P-34** TOTAL CHILDREN EVER BORN	**P-35** TOTAL SURVIVING AND LIVING IN THE HOUSEHOLD	**P-36** TOTAL SURVIVING AND LIVING ELSEWHERE	**P-37** TOTAL CHILDREN NO LONGER ALIVE	**P-38** LAST CHILD BORN	**P-39** SEX OF LAST CHILD BORN	**P-40** LAST CHILD BORN ALIVE	**P-41** DATE OF DEATH OF LAST CHILD BORN
Has *(name)* ever given birth to a live child, even if the child died soon after birth? 1 = Yes 2 = No 3 = Do not know	**At what age did *(name)* have her first child born?**	**How many children has *(name)* ever had that were born alive?**	**How many of *(name's)* children are still alive and living with her in this household, including grown-ups?**	**How many of *(name's)* children are still alive and living elsewhere, including grown-ups?**	**How many of *(name's)* children are no longer alive?**	**When was *(name's)* last child born, even if the child died soon after birth?**	**Is *(name's)* last child born male or female?** 1 = Male 2 = Female 3 = Do not know	**Is *(name's)* last child born still alive?** 1 = Yes 2 = No 3 = Do not know	**When did *(name's)* last child born die?**
Mark the appropriate circle with an X.	**Example** 2 5	**Example** Boys 0 2 Girls 0 2 Total 0 4	**Example** Boys 0 2 Girls 0 1 Total 0 3	**Example** Boys 0 0 Girls 0 0 Total 0 0	**Example** Boys 0 0 Girls 0 1 Total 0 1	**Example** 1 9 0 4 2 0 0 5	*Mark the appropriate circle with an X.*	*Mark the appropriate circle with an X.*	**Example** 1 0 0 3 2 0 0 7
If 2 or 3, Go to H-01		*Write the correct number in the boxes below*	*Write the correct number in the boxes below*	*Write the correct number in the boxes below*	*Write the correct number in the boxes below*			***If 1 or 3, Go to H-01***	
1 Yes 2 No 3 Do not know		Boys Girls Total	Boys Girls Total	Boys Girls Total	Boys Girls Total	D D M M Y Y Y Y	1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know	D D M M Y Y Y Y
1 Yes 2 No 3 Do not know		Boys Girls Total	Boys Girls Total	Boys Girls Total	Boys Girls Total	D D M M Y Y Y Y	1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know	D D M M Y Y Y Y
1 Yes 2 No 3 Do not know		Boys Girls Total	Boys Girls Total	Boys Girls Total	Boys Girls Total	D D M M Y Y Y Y	1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know	D D M M Y Y Y Y
1 Yes 2 No 3 Do not know		Boys Girls Total	Boys Girls Total	Boys Girls Total	Boys Girls Total	D D M M Y Y Y Y	1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know	D D M M Y Y Y Y
1 Yes 2 No 3 Do not know		Boys Girls Total	Boys Girls Total	Boys Girls Total	Boys Girls Total	D D M M Y Y Y Y	1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know	D D M M Y Y Y Y
1 Yes 2 No 3 Do not know		Boys Girls Total	Boys Girls Total	Boys Girls Total	Boys Girls Total	D D M M Y Y Y Y	1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know	D D M M Y Y Y Y
1 Yes 2 No 3 Do not know		Boys Girls Total	Boys Girls Total	Boys Girls Total	Boys Girls Total	D D M M Y Y Y Y	1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know	D D M M Y Y Y Y
1 Yes 2 No 3 Do not know		Boys Girls Total	Boys Girls Total	Boys Girls Total	Boys Girls Total	D D M M Y Y Y Y	1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know	D D M M Y Y Y Y
1 Yes 2 No 3 Do not know		Boys Girls Total	Boys Girls Total	Boys Girls Total	Boys Girls Total	D D M M Y Y Y Y	1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know	D D M M Y Y Y Y
1 Yes 2 No 3 Do not know		Boys Girls Total	Boys Girls Total	Boys Girls Total	Boys Girls Total	D D M M Y Y Y Y	1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know	D D M M Y Y Y Y

SECTION H: HOUSING, HOUSEHOLD GOODS AND SERVICES AND AGRICULTURAL ACTIVITIES - ASK OF EVERY HOUSEHOLD

H-01 TYPE OF LIVING QUARTERS

What is the type of these living quarters?

01 = Housing unit
02 = Converted Hostel (e.g. family unit)
03 = Residential Hotel
04 = Home for the aged
05 = Other

Write the appropriate code in the boxes.

If 03-05, Go to H-07

H-02 TYPE OF MAIN DWELLING

Which of the following best describes the MAIN dwelling and OTHER dwelling(s) that this household occupies?

01 = House or brick/concrete block structure on a separate stand or yard or on a farm
02 = Traditional dwelling/hut/structure made of traditional materials
03 = Flat or apartment in a block of flats
04 = Cluster house in complex
05 = Townhouse (semi-detached house in a complex)
06 = Semi-detached house
07 = House/flat/room in backyard
08 = Informal dwelling (shack in backyard)
09 = Informal dwelling (shack not in backyard, e.g. in an informal/squatter settlement or on a farm)
10 = Room/flatlet on a property or a larger dwelling/servants' quarters/ granny flat
11 = Caravan/tent
12 = Other

Main dwelling **Other dwelling**

Write the appropriate code in the boxes.

H-02a CONSTRUCTION MATERIAL

What is the main material used for the construction of the roof and wall of the MAIN dwelling?

01 = Brick
02 = Cement block/Concrete
03 = Corrugated iron/zinc
04 = Wood
05 = Plastic
06 = Cardboard
07 = Mud and cement mix
08 = Wattle and daub
09 = Tile
10 = Mud
11 = Thatch/Grass
12 = Asbestos
13 = Other

ROOF **WALL**

Write the appropriate code in the boxes.

H-03 ROOMS

How many rooms are there in the MAIN dwelling of this household?

Dining rooms
Living rooms
Dining/Living room
Bedrooms
Study Rooms
One room with multiple uses
Other rooms
Total Rooms

Write the correct number of rooms in the boxes.

Exclude bathrooms and kitchen
Include garages if some members of the household are living in them

H-04 TENURE STATUS

What is the tenure status of this dwelling?

1 = Rented
2 = Owned but not yet paid off
3 = Occupied rent-free
4 = Owned and fully paid off
5 = Other

Write the appropriate code in the box.

*Refers to the MAIN dwelling structure only and **NOT** to the land that it is situated on.*

H-05 ESTIMATED VALUE OF PROPERTY

What would you estimate the market value or municipal valuation of this property to be?

1 = Less than R50 000
2 = R50 001 – R100 000
3 = R100 001 – R200 000
4 = R200 001 – R400 000
5 = R400 001 – R800 000
6 = R800 001 – R1 600 000
7 = R1 600 001 – R3 200 000
8 = More than R3 200 001
9 = Do not know

Write the appropriate code in the box.

H-06 AGE OF THE PROPERTY

What is the age of this dwelling?

01 = Less than one year
02 = 1 - 5 years
03 = 6 - 10 years
04 = 11 - 20 years
05 = 21 - 30 years
06 = 31 - 40 years
07 = 41 - 50 years
08 = 51 - 60 years
09 = 61 years or older
10 = Do not know

Write the appropriate code in the boxes.

The age of the dwelling refers to when the building was completed, not the time of any later remodelling, additions or conversions. If the actual age is not known, give the best estimate.

H-07 ACCESS TO PIPED WATER

In which way does this household mainly get piped water for household use?

1 = Piped (tap) water inside the dwelling
2 = Piped (tap) water inside the yard
3 = Piped (tap) water on community stand: distance less than 200m from dwelling
4 = Piped (tap) water on community stand: distance between 200m and 500m from dwelling
5 = Piped (tap) water on community stand: distance between 500m and 1000m (1 km) from dwelling
6 = Piped (tap) water on community stand: distance greater than 1000m (1 km) from dwelling
7 = No access to piped water

Write the appropriate code in the box.

H-08 SOURCE OF WATER

What is this household's MAIN source of WATER for household use?

1 = Regional/local water scheme (operated by municipality or other water services provider)
2 = Borehole
3 = Spring
4 = Rain water tank
5 = Dam/pool/stagnant water
6 = River/stream
7 = Water vendor
8 = Water tanker
9 = Other

Write the appropriate code in the box.

If 2-9, Go to H-10

SECTION H: HOUSING, HOUSEHOLD GOODS AND SERVICES AND AGRICULTURE ACTIVITIES (Continued)

H-09 RELIABILITY OF WATER SUPPLY

In the last 12 months, has this household had any interruptions in piped water supply?

1 = Yes
2 = No

If 2, Go to H-10

Mark the appropriate circle with an X.

H-09a RELIABILITY OF WATER SUPPLY

Did any specific interruption(s) in piped water supply last longer than two days ?

1 = Yes
2 = No

If 2, Go to H-10

Mark the appropriate circle with an X.

H-09b ALTERNATIVE WATER SOURCE

What alternative water source did the household use during water supply interruption?

1 = Borehole
2 = Spring
3 = Rain water tank
4 = Dam/pool/stagnant water
5 = River/stream
6 = Water vendor
7 = Water tanker
8 = Other
0 = None

Write the appropriate code in the box.

H-10 TOILET FACILITIES

What is the MAIN type of TOILET facility used by this household?

1 = Flush toilet (connected to sewerage system)
2 = Flush toilet (with septic tank)
3 = Chemical toilet
4 = Pit toilet with ventilation (VIP)
5 = Pit toilet without ventilation
6 = Bucket toilet
7 = Other
0 = None

Write the appropriate code in the box.

H-11 ENERGY/FUEL

What type of energy/fuel does this household MAINLY use for cooking, heating and lighting?

COOKING
HEATING
LIGHTING

1 = Electricity
2 = Gas
3 = Paraffin
4 = Wood
5 = Coal
6 = Candles
7 = Animal Dung
8 = Solar
9 = Other
0 = None

Write the appropriate code in the box.

Note

- *Wood (4), coal (5) and animal dung (7) cannot be used for lighting*
- *Candles (6) cannot be used for heating or cooking*

H-12 REFUSE DISPOSAL

How is the refuse or rubbish from this household MAINLY disposed of?

1 = Removed by local authority/private company at least once a week
2 = Removed by local authority/private company less often
3 = Communal refuse dump
4 = Own refuse dump
5 = No rubbish disposal
6 = Other

Write the appropriate code in the box.

H-13 HOUSEHOLD GOODS AND SERVICES

Does this household own any of the following in working order?

1 = Yes
2 = No

Write the appropriate code in the box.

Refrigerator		Motorcar	
Electric/gas stove		Television	
Vacuum cleaner		Radio	
Washing machine		Landline/Telephone	
Computer		Cell phone	
Satellite television		Mail Post box/bag	
DVD Player		Mail delivery at home	

H-13a ACCESS TO INTERNET

How does this household MAINLY access internet?

1 = From home
2 = From Cell phone
3 = From work
4 = From elsewhere
5 = No access to internet

Write the appropriate code in the box.

H-14 AGRICULTURAL ACTIVITIES

What kind of agricultural activity is the household involved in? (More than 1 activity can be chosen)

1 = Livestock production (cattle, goats, sheep, pigs, etc)
2 = Poultry production (chicken, ducks, geese, guinea fowl, ostrich, etc)
3 = Vegetable production
4 = Production of other crops (grains, fruits, etc)
5 = Fodder grazing/pasture/grass for animals
6 = Other
0 = None

Mark the appropriate circle with an X.

If only 2-6, Go to H-14b. If 0, Go to M-00

H-14a LIVESTOCK

How many of the following does the household own?

	0	1 - 10	11 - 100	+ 100
1 = Cattle				
2 = Sheep				
3 = Goats				
4 = Pigs				
5 = Other				

Mark the appropriate circle with an X.

H-14b PLACE OF AGRICULTURAL ACTIVITIES

Where does this household operate its agricultural activities?

1 = Farm land
2 = Backyard or school
3 = Communal or tribal land
4 = Other

Mark the appropriate circle with an X.

X-123456789

A12

SECTION I: MORTALITY IN THE LAST 12 MONTHS

M-00 DEATH OCCURRED

Has any member of this household passed away in the last 12 months (between 10 October 2010 and 9 October 2011)?

1 Yes
2 No
3 Do not know

Mark the appropriate circle with an X.

If 2 or 3, Questionnaire completed

M-00a NUMBER OF DEATHS

How many members of the household passed away in the last 12 months (between 10 October 2010 and 9 October 2011)?

ASK ONLY ABOUT DECEASED WOMEN THAT WERE AGED 12 - 50 AT THE TIME OF DEATH

M-01 NAME OF DECEASED	**M-02** MONTH AND YEAR OF DEATH	**M-03** SEX OF THE DECEASED	**M-04** AGE OF THE DECEASED	**M-05** NATURAL OR UNNATURAL DEATH	**M-06** PREGNANT AT TIME OF DEATH	**M-07** DEATH DURING BIRTH	**M-08** POSTNATAL DEATH
What was the first name of *(the deceased)***?** *Use CAPITAL LETTERS only*	**What was the MONTH and the YEAR of** *(the deceased's)* **death?** *Write the month and year in the appropriate boxes.*	**Was** *(the deceased)* **male or female?** 1 = Male 2 = Female *Mark the appropriate circle with an X.*	**What was** *(the deceased's)* **age in completed years at the time of death?** *Write the age in the boxes. If age is less than 1 year, write 000.*	**Was the death due to a natural or an unnatural cause?** 1 = Natural (e.g. illness) 2 = Unnatural (e.g. accident, assault) 3 = Do not know *Mark the appropriate circle with an X.*	**Did** *(the deceased)* **die while pregnant?** 1 = Yes 2 = No 3 = Do not know *Mark the appropriate circle with an X.*	**Did** *(the deceased)* **die while giving birth?** 1 = Yes 2 = No 3 = Do not know *Mark the appropriate circle with an X.*	**Did** *(the deceased)* **die within 6 weeks after delivery?** 1 = Yes 2 = No 3 = Do not know *Mark the appropriate circle with an X.*
					If 1 to M-06 or M-07, Questionnaire completed		
	M M Y Y Y Y	1 Male 2 Female		1 Natural 2 Unnatural 3 Do not know	1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know
	M M Y Y Y Y	1 Male 2 Female		1 Natural 2 Unnatural 3 Do not know	1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know
	M M Y Y Y Y	1 Male 2 Female		1 Natural 2 Unnatural 3 Do not know	1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know
	M M Y Y Y Y	1 Male 2 Female		1 Natural 2 Unnatural 3 Do not know	1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know
	M M Y Y Y Y	1 Male 2 Female		1 Natural 2 Unnatural 3 Do not know	1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know
	M M Y Y Y Y	1 Male 2 Female		1 Natural 2 Unnatural 3 Do not know	1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know
	M M Y Y Y Y	1 Male 2 Female		1 Natural 2 Unnatural 3 Do not know	1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know
	M M Y Y Y Y	1 Male 2 Female		1 Natural 2 Unnatural 3 Do not know	1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know	1 Yes 2 No 3 Do not know

If more than 8 deaths in the household, use a second questionnaire. Write the barcode of the 1st questionnaire below:

THANK YOU FOR YOUR CO-OPERATION

Annexure B

Census 2011 Variable Encoding

CATEGORICAL DATA ENCODING (CT DATASET)

COL REF	CENSUS CODE	STUDY CODE	VARIABLES & POSSIBLE VARIABLE VALUES
1			**DATREF** **(UNIQUE IDENTIFIER)**
2	n/a	n/a	**QNTYPE** **(SINGLE ANSWER)**
	1	1	Household questionnaire
3			**SN** **(UNIQUE IDENTIFIER)**
4	n/a	n/a	**QUARTERS** **CATEGORICAL**
	1	1	Housing Unit
	2	2	Converted Hostel
5			**MAINDWELLING** **CATEGORICAL**
	1	1	House or brick/concrete block structure on a separate stand
	2	2	Traditional dwelling/hut/structure made of traditional mater
	3	3	Flat or apartment in a block of flats
	4	4	Cluster/Townhouse
	5	4	Cluster/Townhouse
	6	5	Semi-detached house
	7	6	Second Dwelling
	8	7	Informal dwelling/shack in back yard
	9	8	Informal dwelling/shack NOT in back yard. e.g. in an informa
	10	6	Second Dwelling
	11	9	Other
	12	9	Other
6			**OTHERDWELLING** **CATEGORICAL**
	1	1	House or brick/concrete block structure on a separate stand
	2	2	Traditional dwelling/hut/structure made of traditional mater
	3	3	Flat or apartment in a block of flats
	4	4	Cluster house in complex
	5	5	Town house (semi-detached house in complex)
	6	6	Semi-detached house
	7	7	House/flat/room in back yard
	8	8	Informal dwelling/shack in back yard
	9	9	Informal dwelling/shack NOT in back yard. e.g. in an informa
	10	10	Room/flatlet on a property or a larger dwelling/servants qua
	11	11	Caravan or tent
	12	12	Other
	99	13	Unspecified

CATEGORICAL DATA ENCODING (CT DATASET)

COL REF	CENSUS CODE	STUDY CODE	VARIABLES & POSSIBLE VARIABLE VALUES	
7			**ROOF**	**CATEGORICAL**
	2	1	Cement block/concrete	
	3	2	Corrugated iron/zink	
	4	3	Wood	
	5	4	Plastic	
	6	5	Cardboard	
	8	6	Wattle and daub	
	9	7	Tile	
	11	8	Thatch/grass	
	12	9	Asbestos	
	13	10	Other	
8			**WALL**	**CATEGORICAL**
	1	1	Brick	
	2	2	Cement block/concrete	
	3	3	Corrugated iron/zink	
	4	4	Wood	
	5	5	Plastic	
	6	6	Cardboard	
	7	7	Mud and cement mix	
	8	8	Wattle and daub	
	10	9	Mud	
	11	10	Thatch/grass	
	12	11	Asbestos	
	13	12	Other	
9			**DININGROOMS**	**COUNT**
			0 1 2	
10			**LIVINGROOMS**	**COUNT**
			0 1 2 3	
11			**DINING**	**COUNT**
			0 1 2	

CATEGORICAL DATA ENCODING (CT DATASET)

COL REF	CENSUS CODE	STUDY CODE	VARIABLES & POSSIBLE VARIABLE VALUES
12			**BEDROOMS** **COUNT**
13			0 1 2 3 4 5 **STUDYROOMS** **COUNT**
14			0 1 2 **MULTIUSE** **COUNT**
15			1 **OTHERROOMS** **COUNT**
16			0 1 2 3 4 5 **TOTROOMS** **COUNT**
			1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

CATEGORICAL DATA ENCODING (CT DATASET)

COL REF	CENSUS CODE	STUDY CODE	VARIABLES & POSSIBLE VARIABLE VALUES	
17			**TENURE**	**CATEGORICAL**
	1	1	Rented	
	2	2	Owned but not yet paid off	
	3	3	Occupied rent-free	
	4	4	Owned and fully paid off	
	5	5	Other	
18			**WATERPIPED**	**CATEGORICAL**
	1	1	Piped (tap) water inside the dwelling	
	2	2	Piped (tap) water inside the yard	
	3	3	Piped (tap) water on community stand: distance less than 200	
	4	4	Piped (tap) water to community stand: distance less than 200	
	5	5	Piped (tap) water to community stand: distance less than 500	
	6	6	Piped (tap) water on community stand: distance greater than	
	7	7	No access to piped (tap) water	
19			**WSOURCE**	**CATEGORICAL**
	1	1	Regional/local water scheme (operated by a Water Service Aut	
	2	2	Borehole	
	3	3	Spring	
	4	4	Rain-water tank	
	5	5	Dam / pool / stagnant water	
	6	6	River/stream	
	7	7	Water vendor	
	8	8	Water tanker	
	9	9	Other	
20			**WSUPPLY**	**CATEGORICAL**
	1	1	Yes	
	2	2	No	
	Sysmiss	3	NA	
21			**WSUPPLY2**	**CATEGORICAL**
	1	1	Yes	
	2	2	No	
	Sysmiss	3	NA	

CATEGORICAL DATA ENCODING (CT DATASET)

COL REF	CENSUS CODE	STUDY CODE	VARIABLES & POSSIBLE VARIABLE VALUES	
22			**ALTWSOURCE**	**CATEGORICAL**
	0	1	None	
	1	2	Borehole	
	2	3	Spring	
	3	4	Rain water tank	
	4	5	Dam/pool/stagnant water	
	5	6	River/stream	
	6	7	Water vendor	
	7	8	Water tanker	
	8	9	Other	
	Sysmiss	10	NA	
23			**TOILET**	**CATEGORICAL**
	0	1	None	
	1	2	Flush toilet (connected to sewerage system)	
	2	3	Flush toilet (with septic tank)	
	3	4	Chemical toilet	
	4	5	Pit latrine with ventilation (VIP)	
	5	6	Pit latrine without ventilation	
	6	7	Bucket latrine	
	7	8	Other	
24			**ECOOKING**	**CATEGORICAL**
	0	1	None	
	1	2	Electricity	
	2	3	Gas	
	3	4	Paraffin	
	4	5	Wood	
	5	6	Coal	
	7	7	Animal dung	
	8	8	Solar	
	9	9	Other	

CATEGORICAL DATA ENCODING (CT DATASET)

COL REF	CENSUS CODE	STUDY CODE	VARIABLES & POSSIBLE VARIABLE VALUES	
25			**EHEATING**	**CATEGORICAL**
	0	1	None	
	1	2	Electricity	
	2	3	Gas	
	3	4	Paraffin	
	4	5	Wood	
	5	6	Coal	
	7	7	Animal dung	
	8	8	Solar	
	9	9	Other	
26			**ELIGHTING**	**CATEGORICAL**
	0	1	None	
	1	2	Electricity	
	2	3	Gas	
	3	4	Paraffin	
	6	5	Candles	
	8	6	Solar	
27			**REFUSE**	**CATEGORICAL**
	1	1	Removed by local authority at least once a week	
	2	2	Removed by local authority less often	
	3	3	Communal refuse dump	
	4	4	Own refuse dump	
	5	5	No rubbish disposal	
	6	6	Other	
28			**REFRIDGERATOR**	**CATEGORICAL**
	1	1	Yes	
	2	2	No	
29			**STOVE**	**CATEGORICAL**
	1	1	Yes	
	2	2	No	
30			**VACUUM**	**CATEGORICAL**
	1	1	Yes	
	2	2	No	

CATEGORICAL DATA ENCODING (CT DATASET)

COL REF	CENSUS CODE	STUDY CODE	VARIABLES & POSSIBLE VARIABLE VALUES	
31			**WASHINGM**	**CATEGORICAL**
	1	1	Yes	
	2	2	No	
32			**COMPUTER**	**CATEGORICAL**
	1	1	Yes	
	2	2	No	
33			**SATELLITE**	**CATEGORICAL**
	1	1	Yes	
	2	2	No	
34			**DVD_PLAYER**	**CATEGORICAL**
	1	1	Yes	
	2	2	No	
35			**MOTORCAR**	**CATEGORICAL**
	1	1	Yes	
	2	2	No	
36			**TV**	**CATEGORICAL**
	1	1	Yes	
	2	2	No	
37			**RADIO**	**CATEGORICAL**
	1	1	Yes	
	2	2	No	
38			**LANDLINE**	**CATEGORICAL**
	1	1	Yes	
	2	2	No	
39			**CELLPHONE**	**CATEGORICAL**
	1	1	Yes	
	2	2	No	
40			**POSTBOX**	**CATEGORICAL**
	1	1	Yes	
	2	2	No	

CATEGORICAL DATA ENCODING (CT DATASET)

COL REF	CENSUS CODE	STUDY CODE	VARIABLES & POSSIBLE VARIABLE VALUES	
41			**MAIL**	**CATEGORICAL**
	1	1	Yes	
	2	2	No	
42			**INTERNET**	**CATEGORICAL**
	1	1	From home	
	2	2	From cell phone	
	3	3	From work	
	4	4	From elsewhere	
	5	5	No access to internet	
43			**DEATHS**	**CATEGORICAL**
	1	1	Yes	
	2	2	No	
	3	3	Dont know	
44			**DEATHSNO**	**COUNT**
			1	
			2	
			3	
			4	
			NA	
45			**HHAGE**	**CONTINUOUS**
46			**HHPOP**	**CATEGORICAL**
	1	1	Black African	
	2	2	Coloured	
	3	3	Indian or Asian	
	4	4	White	
	5	5	Other	
	9	6	Unspecified	
47			**HHSEX**	**CATEGORICAL**
	1	1	Male	
	2	2	Female	
	9	3	Unspecified	
48			**HINCOME**	**CONTINUOUS**

CATEGORICAL DATA ENCODING (CT DATASET)

COL REF	CENSUS CODE	STUDY CODE	VARIABLES & POSSIBLE VARIABLE VALUES	
49			**HHEMPLOY**	**CATEGORICAL**
	1	1	Employed	
	2	2	Unemployed	
	3	3	Discouraged work-seeker	
	4	4	Not economically active	
	5	5	Household head out of working age scope i.e. 15-64	
50			**HSIZE**	**COUNT**
51			**XPOP**	**CATEGORICAL**
	1	1	Black African	
	2	2	Coloured	
	3	3	Indian or Asian	
	4	4	White	
	5	5	Other	
	Sysmiss	6	NA	
52			**INCOME**	**CATEGORICAL**
	1	1	No income	
	2	2	R 1 - R 4800	
	3	3	R 4801 - R 9600	
	4	4	R 9601 - R 19200	
	5	5	R 19201 - R 38400	
	6	6	R 38401 - R 76800	
	7	7	R 76801 - R 153600	
	8	8	R 153601 - R 307200	
	9	9	R 307201 - R 614400	
	10	10	R 614401- R 1228800	
	11	11	R 1228801 - R 2457600	
	12	12	R2457601 or more	
	99	13	Unspecified	
53			**GEOTYPE**	**CATEGORICAL**
	1	1	Urban	
	2	2	Traditional	
	3	3	Farms	
54			**PROVINCE**	**CATEGORICAL**
	1	1	Western cape	
	Var	n/a	Many Others	

CATEGORICAL DATA ENCODING (CT DATASET)

COL REF	CENSUS CODE	STUDY CODE	VARIABLES & POSSIBLE VARIABLE VALUES	
55			**DISTRICT**	**CATEGORICAL**
	199 Var	1 n/a	City of Cape Town Many Others	
56			**MUNIC**	**CATEGORICAL**
	199 Var	1 n/a	City of Cape Town Many Others	
57			**CONT**	**CONTINUOUS**

Annexure C

R Scripts Used

ACCESSING THE CENSUS DATA AND CONVERSION TO CSV

STEP A – Setting the Working Directory in R

```
#Setting the working directory

setwd(dir="C:/Users/Ross/Documents/R/Working Directory")

#Requesting a list of all files located in the working directory

dir()
```

STEP B – Converting Data from SPSS format to CSV format

```
#Covert Data in working directory from SPSS format to CSV format

library(foreign)

write.table(read.spss("sa-census-2011-household-v1.1-20140618.sav"), file="Census_2011_Household.csv", quote
= FALSE, sep = ",")
```

STEP E – Importing CSV Dataset and converting the Dataset to an R Data Frame

```
#Importing the Dataset

dataset = read.csv("Census_2011_Household.csv")
```

SA CENSUS 2011 DATA PRE-PROCESSING

STEP 1 - Reducing Dataset to Cape Town only

```
##Creating a Cape Town only Dataset

#Importing the Dataset

dataset = read.csv("Census_2011_Household.csv")

#Creating a new Cape Town dataframe with irrelevant data entries removed

dataset_CT <- dataset[!(dataset$DISTRICT %in% c('West Coast','Cape Winelands','Overberg','Eden','Central
Karoo','Buffalo City','Cacadu','Amathole','Chris Hani','Ukhahlamba','O.R.Tambo','Alfred Nzo','Nelson Mandela
Bay','John Taolo Gaetsewe','Namakwa','Pixley ka Seme','Siyanda','Frances
Baard','Xhariep','Lejweleputswa','Thabo Mofutsanyane','Fezile
Dabi','Mangaung','Ugu','Uthukela','Amajuba','Zululand','Uthungulu','iLembe','UMgungundlovu','Umzinyathi','Umk
hanyakude','Sisonke','eThekwini Metropolitan','Bojanala','Ngaka Modiri Molema','Dr Ruth Segomotsi
Mompati','Dr Kenneth Kaunda','Sedibeng','West Rand','Ekurhuleni','City of Johannesburg','City of
Tshwane','Gert Sibande','Nkangala','Ehlanzeni','Mopani','Vhembe','Capricorn','Waterberg','Greater
Sekhukhune')),]

#Creating a new CSV file for the new Cape Town dataframe

write.csv(dataset_CT,'Census_2011_Household_CT.csv')
```

STEP 3 - Encoding Categorical Data

```
##Encoding the Categorical Variables

#Importing the Dataset

dataset = read.csv('Census_2011_Household_CT_Hgth.csv')

#Encoding QNTYPE

dataset$QNTYPE  = factor(dataset$QNTYPE ,

                         levels = c('Household questionnaire'),

                         labels = c(1))

#Encoding QUARTERS

dataset$QUARTERS = factor(dataset$QUARTERS,

                          levels = c('Housing Unit', 'Converted Hostel'),

                          labels = c(1, 2))

#Encoding MAINDWELLING

dataset$MAINDWELLING = factor(dataset$MAINDWELLING,

                              levels = c('House or brick/concrete block structure on a separate stand ',
'Traditional dwelling/hut/structure made of traditional mater', 'Flat or apartment in a block of flats',
'Cluster/Townhouse', 'Semi-detached house', 'Second Dwelling', 'Informal dwelling/shack in back yard',
'Informal dwelling/shack NOT in back yard. e.g. in an informa', 'Other'),

                              labels = c(1, 2, 3, 4, 5, 6, 7, 8, 9))

#Encoding OTHERDWELLING

dataset$OTHERDWELLING = factor(dataset$OTHERDWELLING,

                               levels = c('House or brick/concrete block structure on a separate stand ',
'Traditional dwelling/hut/structure made of traditional mater', 'Flat or apartment in a block of flats',
'Cluster house in complex', 'Town house (semi-detached house in complex)', 'Semi-detached house',
'House/flat/room in back yard', 'Informal dwelling/shack in back yard', 'Informal dwelling/shack NOT in back
yard. e.g. in an informa', 'Room/flatlet on a property or a larger dwelling/servants qua', 'Caravan or tent',
'Other', 'Unspecified'),

                               labels = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13))

#Encoding ROOF

dataset$ROOF = factor(dataset$ROOF,

                      levels = c('Cement block/concrete', 'Corrugated iron/zink', 'Wood', 'Plastic',
'Cardboard', 'Wattle and daub', 'Tile', 'Thatch/grass', 'Asbestos', 'Other'),

                      labels = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10))

#Encoding WALL

dataset$WALL = factor(dataset$WALL,

                      levels = c('Brick', 'Cement block/concrete', 'Corrugated iron/zink', 'Wood', 'Plastic',
'Cardboard', 'Mud and cement mix', 'Wattle and daub', 'Mud', 'Thatch/grass', 'Asbestos', 'Other'),

                      labels = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12))

#Encoding TENURE

dataset$TENURE = factor(dataset$TENURE,

                        levels = c('Rented', 'Owned but not yet paid off', 'Occupied rent-free', 'Owned and
fully paid off', 'Other'),

                        labels = c(1, 2, 3, 4, 5))

#Encoding WATERPIPED

dataset$WATERPIPED = factor(dataset$WATERPIPED,

                            levels = c('Piped (tap) water inside the dwelling', 'Piped (tap) water inside the
yard', 'Piped (tap) water on community stand: distance less than 200', 'Piped (tap) water to community stand:
distance less than 200', 'Piped (tap) water to community stand: distance less than 500', 'Piped (tap) water
on community stand: distance greater than ', 'No access to piped (tap) water'),

                            labels = c(1, 2, 3, 4, 5, 6, 7))

#Encoding WSOURCE

dataset$WSOURCE = factor(dataset$WSOURCE,

                         levels = c('Regional/local water scheme (operated by a Water Service Aut',
'Borehole', 'Spring', 'Rain-water tank', 'Dam / pool / stagnant water', 'River/stream', 'Water vendor',
```

```
'Water tanker', 'Other'),
                          labels = c(1, 2, 3, 4, 5, 6, 7, 8, 9))
#Encoding WSUPPLY
dataset$WSUPPLY = factor(dataset$WSUPPLY,
                          levels = c('Yes', 'No', 'NA'),
                          labels = c(1, 2, 3))
#Encoding WSUPPLY2
dataset$WSUPPLY2 = factor(dataset$WSUPPLY2,
                           levels = c('Yes', 'No', 'NA'),
                           labels = c(1, 2, 3))
#Encoding ALTWSOURCE
dataset$ALTWSOURCE = factor(dataset$ALTWSOURCE,
                            levels = c('None', 'Borehole', 'Spring', 'Rain water tank', 'Dam/pool/stagnant
water', 'River/stream', 'Water vendor', 'Water tanker', 'Other', 'NA'),
                            labels = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10))
#Encoding TOILET
dataset$TOILET = factor(dataset$TOILET,
                        levels = c('None', 'Flush toilet (connected to sewerage system)', 'Flush toilet (with
septic tank)', 'Chemical toilet', 'Pit latrine with ventilation (VIP)', 'Pit latrine without ventilation',
'Bucket latrine', 'Other'),
                        labels = c(1, 2, 3, 4, 5, 6, 7, 8))
#Encoding ECOOKING
dataset$ECOOKING = factor(dataset$ECOOKING,
                          levels = c('None', 'Electricity', 'Gas', 'Paraffin', 'Wood', 'Coal', 'Animal dung',
'Solar', 'Other'),
                          labels = c(1, 2, 3, 4, 5, 6, 7, 8, 9))
#Encoding EHEATING
dataset$EHEATING = factor(dataset$EHEATING,
                          levels = c('None', 'Electricity', 'Gas', 'Paraffin', 'Wood', 'Coal', 'Animal dung',
'Solar', 'Other'),
                          labels = c(1, 2, 3, 4, 5, 6, 7, 8, 9))
#Encoding ELIGHTING
dataset$ELIGHTING = factor(dataset$ELIGHTING,
                           levels = c('None', 'Electricity', 'Gas', 'Paraffin', 'Candles', 'Solar'),
                           labels = c(1, 2, 3, 4, 5, 6))
#Encoding REFUSE
dataset$REFUSE = factor(dataset$REFUSE,
                        levels = c('Removed by local authority at least once a week', 'Removed by local
authority less often', 'Communal refuse dump', 'Own refuse dump', 'No rubbish disposal', 'Other'),
                        labels = c(1, 2, 3, 4, 5, 6))
#Encoding REFRIDGERATOR
dataset$REFRIDGERATOR = factor(dataset$REFRIDGERATOR,
                               levels = c('Yes', 'No'),
                               labels = c(1, 2))
#Encoding STOVE
dataset$STOVE = factor(dataset$STOVE,
                       levels = c('Yes', 'No'),
                       labels = c(1, 2))
#Encoding VACUUM
dataset$VACUUM = factor(dataset$VACUUM,
                        levels = c('Yes', 'No'),
```

```
                        labels = c(1, 2))
#Encoding WASHINGM
dataset$WASHINGM = factor(dataset$WASHINGM,
                          levels = c('Yes', 'No'),
                          labels = c(1, 2))
#Encoding COMPUTER
dataset$COMPUTER = factor(dataset$COMPUTER,
                          levels = c('Yes', 'No'),
                          labels = c(1, 2))
#Encoding SATELLITE
dataset$SATELLITE = factor(dataset$SATELLITE,
                           levels = c('Yes', 'No'),
                           labels = c(1, 2))
#Encoding DVD_PLAYER
dataset$DVD_PLAYER = factor(dataset$DVD_PLAYER,
                            levels = c('Yes', 'No'),
                            labels = c(1, 2))
#Encoding MOTORCAR
dataset$MOTORCAR = factor(dataset$MOTORCAR,
                          levels = c('Yes', 'No'),
                          labels = c(1, 2))
#Encoding TV
dataset$TV = factor(dataset$TV,
                    levels = c('Yes', 'No'),
                    labels = c(1, 2))
#Encoding RADIO
dataset$RADIO = factor(dataset$RADIO,
                       levels = c('Yes', 'No'),
                       labels = c(1, 2))
#Encoding LANDLINE
dataset$LANDLINE = factor(dataset$LANDLINE,
                          levels = c('Yes', 'No'),
                          labels = c(1, 2))
#Encoding CELLPHONE
dataset$CELLPHONE = factor(dataset$CELLPHONE,
                           levels = c('Yes', 'No'),
                           labels = c(1, 2))
#Encoding POSTBOX
dataset$POSTBOX = factor(dataset$POSTBOX,
                         levels = c('Yes', 'No'),
                         labels = c(1, 2))
#Encoding MAIL
dataset$MAIL = factor(dataset$MAIL,
                      levels = c('Yes', 'No'),
                      labels = c(1, 2))
#Encoding INTERNET
dataset$INTERNET = factor(dataset$INTERNET,
                          levels = c('From home', 'From cell phone', 'From work', 'From elsewhere', 'No
access to internet'),
```

```
                          labels = c(1, 2, 3, 4, 5))
#Encoding DEATHS
dataset$DEATHS = factor(dataset$DEATHS,
                        levels = c('Yes', 'No', 'Dont know'),
                        labels = c(1, 2, 3))
#Encoding HHPOP
dataset$HHPOP = factor(dataset$HHPOP,
                       levels = c('Black African', 'Coloured', 'Indian or Asian', 'White', 'Other',
'Unspecified'),
                       labels = c(1, 2, 3, 4, 5, 6))


#Encoding HHSEX
dataset$HHSEX = factor(dataset$HHSEX,
                       levels = c('Male', 'Female', 'Unspecified'),
                       labels = c(1, 2, 3))
#Encoding HHEMPLOY
dataset$HHEMPLOY = factor(dataset$HHEMPLOY,
                          levels = c('Employed', 'Unemployed', 'Discouraged work-seeker', 'Not economically
active', 'Household head out of working age scope i.e. 15-64'),
                          labels = c(1, 2, 3, 4, 5))
#Encoding XPOP
dataset$XPOP = factor(dataset$XPOP,
                      levels = c('Black African', 'Coloured', 'Indian or Asian', 'White', 'Other', 'NA'),
                      labels = c(1, 2, 3, 4, 5, 6))
#Encoding INCOME
dataset$INCOME = factor(dataset$INCOME,
                        levels = c('No income', 'R 1 - R 4800', 'R 4801 - R 9600', 'R 9601 - R 19200', 'R
19201 - R 38400', 'R 38401 - R 76800', 'R 76801 - R 153600', 'R 153601 - R 307200', 'R 307201 - R 614400', 'R
614401- R 1228800', 'R 1228801 - R 2457600', 'R2457601 or more', 'Unspecified'),
                        labels = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13))
#Encoding GEOTYPE
dataset$GEOTYPE = factor(dataset$GEOTYPE,
                         levels = c('Urban', 'Traditional', 'Farms'),
                         labels = c(1, 2, 3))
#Encoding PROVINCE
dataset$PROVINCE = factor(dataset$PROVINCE,
                          levels = c('Western cape'),
                          labels = c(1))
#Encoding DISTRICT
dataset$DISTRICT = factor(dataset$DISTRICT,
                          levels = c('City of Cape Town'),
                          labels = c(1))
#Encoding MUNIC
dataset$MUNIC = factor(dataset$MUNIC,
                       levels = c('City of Cape Town'),
                       labels = c(1))
#Creating a new CSV file for the Categorical Variable Encoded CT dataframe
write.csv(dataset,'Census_2011_Household_CT_Hgth_Enc.csv')
```

DATA PROCESSING

MAINDWELLING – Model 1 & 2 (70:30 Training:Test Split)

```
## Part 1 - Importing the dataset
dataset = read.csv('Census_2011_Household_CT_Hgth_Enc.csv')

## Part 2 - Selecting the appropriate variables
dataset_maindwelling <- subset(dataset,
select=c("MAINDWELLING","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))

## Part 3 – Generating a Frequency Table for the Dependent Variable
table(dataset_maindwelling$MAINDWELLING)

## Part 4 - Splitting the dataset into the Training Set and Test Set
#install.packages('caTools')
library(caTools)
set.seed(123)
split = sample.split(dataset_maindwelling$MAINDWELLING, SplitRatio = 0.70)
maindwelling_training_set <- subset(dataset_maindwelling, split == TRUE)
maindwelling_test_set <- subset(dataset_maindwelling, split == FALSE)

## Part 5 – Making the Dependent Variable a Factor
maindwelling_training_set$MAINDWELLING <- as.factor(maindwelling_training_set$MAINDWELLING)
maindwelling_test_set$MAINDWELLING <- as.factor(maindwelling_test_set$MAINDWELLING)

## Part 6 - Feature Scaling
maindwelling_training_set[,2] = scale(maindwelling_training_set[,2])
maindwelling_test_set[,2] = scale(maindwelling_test_set[,2])

## Part 7 – Creating the Random Forest Model
library(randomForest)
set.seed(222)
rf <- randomForest(MAINDWELLING~.,data=maindwelling_training_set)
print(rf)

## Part 8 – Publishing the Results (Model 1)
# Installation of caret
install.packages('caret', dependencies = TRUE)
# Lava error, install lava
#install.packages('lava')
# Histogram showing No. of nodes for trees (Model 1)
hist(treesize(rf),
     main = "No. of Nodes for the Trees",
     col = "SkyBlue")
# Variable importance (Model 1)
varImpPlot(rf)
importance(rf)
```

```
# Prediction & Confusion Matrix (Model 1)
library(caret)
p1 <- predict(rf, maindwelling_training_set)
confusionMatrix(p1, maindwelling_training_set$MAINDWELLING)
p2 <- predict(rf, maindwelling_test_set)
confusionMatrix(p2, maindwelling_test_set$MAINDWELLING)

## Part 9 – Tuning the Random Forest Model
# Visualise Error Rate of Random Forest
plot(rf)
# Tuning mtry
t <- tuneRF(maindwelling_training_set[,-1],maindwelling_training_set[,1],
       stepFactor = 0.5,
       plot = TRUE,
       ntreeTry = 150,
       trace = TRUE,
       improve = 0.05)

## Part 10 – Creating the updated Random Forest Model (Model 2)
library(randomForest)
set.seed(222)
rf<- randomForest(MAINDWELLING~.,data=maindwelling_training_set,
                  ntree = 150,
                  mtry = 2,
                  importance = TRUE)
print(rf)

## Part 11 – Publishing the Results of the New Model (Model 2)
# Histogram showing No. of nodes for trees (Model 2)
hist(treesize(rf),
     main = "No. of Nodes for the Trees",
     col = "SkyBlue")
# Variable importance (Model 2)
varImpPlot(rf)
importance(rf)
# Prediction & Confusion Matrix (Model 2)
library(caret)
p1 <- predict(rf, maindwelling_training_set)
confusionMatrix(p1, maindwelling_training_set$MAINDWELLING)
p2 <- predict(rf, maindwelling_test_set)
confusionMatrix(p2, maindwelling_test_set$MAINDWELLING)
```

MAINDWELLING – Model 3 & 4 (80:20 Training:Test Split)

```
## Part 1 - Importing the dataset
dataset = read.csv('Census_2011_Household_CT_Hgth_Enc.csv')
```

```
## Part 2 - Selecting the appropriate variables
dataset_maindwelling <- subset(dataset,
select=c("MAINDWELLING","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))

## Part 3 – Generating a Frequency Table for the Dependent Variable
table(dataset_maindwelling$MAINDWELLING)

## Part 4 - Splitting the dataset into the Training Set and Test Set
#install.packages('caTools')
library(caTools)
set.seed(123)
split = sample.split(dataset_maindwelling$MAINDWELLING, SplitRatio = 0.80)
maindwelling_training_set <- subset(dataset_maindwelling, split == TRUE)
maindwelling_test_set <- subset(dataset_maindwelling, split == FALSE)

## Part 5 – Making the Dependent Variable a Factor
maindwelling_training_set$MAINDWELLING <- as.factor(maindwelling_training_set$MAINDWELLING)
maindwelling_test_set$MAINDWELLING <- as.factor(maindwelling_test_set$MAINDWELLING)

## Part 6 - Feature Scaling
maindwelling_training_set[,2] = scale(maindwelling_training_set[,2])
maindwelling_test_set[,2] = scale(maindwelling_test_set[,2])

## Part 7 – Creating the Random Forest Model
library(randomForest)
set.seed(222)
rf <- randomForest(MAINDWELLING~.,data=maindwelling_training_set)
print(rf)

## Part 8 – Publishing the Results (Model 3)
# Installation of caret
install.packages('caret', dependencies = TRUE)
# Lava error, install lava
#install.packages('lava')
# Histogram showing No. of nodes for trees (Model 3)
hist(treesize(rf),
     main = "No. of Nodes for the Trees",
     col = "SkyBlue")
# Variable importance (Model 3)
varImpPlot(rf)
importance(rf)

# Prediction & Confusion Matrix (Model 3)
library(caret)
p1 <- predict(rf, maindwelling_training_set)
confusionMatrix(p1, maindwelling_training_set$MAINDWELLING)
p2 <- predict(rf, maindwelling_test_set)
confusionMatrix(p2, maindwelling_test_set$MAINDWELLING)
```

```
## Part 9 – Tuning the Random Forest Model
# Visualise Error Rate of Random Forest
plot(rf)
# Tuning mtry
t <- tuneRF(maindwelling_training_set[,-1],maindwelling_training_set[,1],
       stepFactor = 0.5,
       plot = TRUE,
       ntreeTry = 150,
       trace = TRUE,
       improve = 0.05)

## Part 10 – Creating the updated Random Forest Model (Model 4)
library(randomForest)
set.seed(222)
rf<- randomForest(MAINDWELLING~.,data=maindwelling_training_set,
                  ntree = 150,
                  mtry = 2,
                  importance = TRUE)
print(rf)

## Part 11 – Publishing the Results of the New Model (Model 4)
# Histogram showing No. of nodes for trees (Model 4)
hist(treesize(rf),
     main = "No. of Nodes for the Trees",
     col = "SkyBlue")
# Variable importance (Model 4)
varImpPlot(rf)
importance(rf)
# Prediction & Confusion Matrix (Model 4)
library(caret)
p1 <- predict(rf, maindwelling_training_set)
confusionMatrix(p1, maindwelling_training_set$MAINDWELLING)
p2 <- predict(rf, maindwelling_test_set)
confusionMatrix(p2, maindwelling_test_set$MAINDWELLING)
```

MAINDWELLING – Model 5 & 6 (90:10 Training:Test Split)

```
## Part 1 - Importing the dataset
dataset = read.csv('Census_2011_Household_CT_Hgth_Enc.csv')


## Part 2 - Selecting the appropriate variables
dataset_maindwelling <- subset(dataset,
select=c("MAINDWELLING","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))


## Part 3 – Generating a Frequency Table for the Dependent Variable
table(dataset_maindwelling$MAINDWELLING)
```

```
## Part 4 - Splitting the dataset into the Training Set and Test Set
#install.packages('caTools')
library(caTools)
set.seed(123)
split = sample.split(dataset_maindwelling$MAINDWELLING, SplitRatio = 0.90)
maindwelling_training_set <- subset(dataset_maindwelling, split == TRUE)
maindwelling_test_set <- subset(dataset_maindwelling, split == FALSE)

## Part 5 – Making the Dependent Variable a Factor
maindwelling_training_set$MAINDWELLING <- as.factor(maindwelling_training_set$MAINDWELLING)
maindwelling_test_set$MAINDWELLING <- as.factor(maindwelling_test_set$MAINDWELLING)

## Part 6 - Feature Scaling
maindwelling_training_set[,2] = scale(maindwelling_training_set[,2])
maindwelling_test_set[,2] = scale(maindwelling_test_set[,2])

## Part 7 – Creating the Random Forest Model
library(randomForest)
set.seed(222)
rf <- randomForest(MAINDWELLING~.,data=maindwelling_training_set)
print(rf)

## Part 8 – Publishing the Results (Model 5)
# Installation of caret
install.packages('caret', dependencies = TRUE)
# Lava error, install lava
#install.packages('lava')
# Histogram showing No. of nodes for trees (Model 5)
hist(treesize(rf),
     main = "No. of Nodes for the Trees",
     col = "SkyBlue")
# Variable importance (Model 5)
varImpPlot(rf)
importance(rf)

# Prediction & Confusion Matrix (Model 5)
library(caret)
p1 <- predict(rf, maindwelling_training_set)
confusionMatrix(p1, maindwelling_training_set$MAINDWELLING)
p2 <- predict(rf, maindwelling_test_set)
confusionMatrix(p2, maindwelling_test_set$MAINDWELLING)

## Part 9 – Tuning the Random Forest Model
# Visualise Error Rate of Random Forest
plot(rf)
# Tuning mtry
t <- tuneRF(maindwelling_training_set[,-1],maindwelling_training_set[,1],
       stepFactor = 0.5,
       plot = TRUE,
```

```
       ntreeTry = 150,
       trace = TRUE,
       improve = 0.05)

## Part 10 – Creating the updated Random Forest Model (Model 6)
library(randomForest)
set.seed(222)
rf<- randomForest(MAINDWELLING~.,data=maindwelling_training_set,
                  ntree = 150,
                  mtry = 2,
                  importance = TRUE)
print(rf)

## Part 11 – Publishing the Results of the New Model (Model 6)
# Histogram showing No. of nodes for trees (Model 6)
hist(treesize(rf),
     main = "No. of Nodes for the Trees",
     col = "SkyBlue")
# Variable importance (Model 6)
varImpPlot(rf)
importance(rf)
# Prediction & Confusion Matrix (Model 6)
library(caret)
p1 <- predict(rf, maindwelling_training_set)
confusionMatrix(p1, maindwelling_training_set$MAINDWELLING)
p2 <- predict(rf, maindwelling_test_set)
confusionMatrix(p2, maindwelling_test_set$MAINDWELLING)
```

MAINDWELLING – Model 7 & 8 (70:30 Training:Test Split, with Stratified Sampling)

```
## Part 1 - Importing the dataset
dataset = read.csv('Census_2011_Household_CT_Hgth_Enc.csv')

## Part 2 - Selecting the appropriate variables
dataset_maindwelling <- subset(dataset,
                        select=c("MAINDWELLING","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))

## Part 3 - Generating a Frequency Table for the Dependent Variable
table(dataset_maindwelling$MAINDWELLING)

## PART 4 - stratified sampling

# Extract maindwelling by class
dataset_maindwelling_1 <- subset(dataset_maindwelling, MAINDWELLING=="1",
                           select=c("MAINDWELLING","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))
dataset_maindwelling_2 <- subset(dataset_maindwelling, MAINDWELLING=="2",
                           select=c("MAINDWELLING","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))
dataset_maindwelling_3 <- subset(dataset_maindwelling, MAINDWELLING=="3",
```

```
                          select=c("MAINDWELLING","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))
dataset_maindwelling_4 <- subset(dataset_maindwelling, MAINDWELLING=="4",
                          select=c("MAINDWELLING","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))
dataset_maindwelling_5 <- subset(dataset_maindwelling, MAINDWELLING=="5",
                          select=c("MAINDWELLING","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))
dataset_maindwelling_6 <- subset(dataset_maindwelling, MAINDWELLING=="6",
                          select=c("MAINDWELLING","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))
dataset_maindwelling_7 <- subset(dataset_maindwelling, MAINDWELLING=="7",
                          select=c("MAINDWELLING","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))
dataset_maindwelling_8 <- subset(dataset_maindwelling, MAINDWELLING=="8",
                          select=c("MAINDWELLING","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))
dataset_maindwelling_9 <- subset(dataset_maindwelling, MAINDWELLING=="9",
                          select=c("MAINDWELLING","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))

# split maindwelling into training and test sets
# install.packages('caTools')
library(caTools)
# Maindwelling 1
set.seed(123)
split = sample.split(dataset_maindwelling_1$MAINDWELLING, SplitRatio = 0.70)
dataset_maindwelling_1_training_set <- subset(dataset_maindwelling_1, split == TRUE)
dataset_maindwelling_1_test_set <- subset(dataset_maindwelling_1, split == FALSE)
# Maindwelling 2
set.seed(123)
split = sample.split(dataset_maindwelling_2$MAINDWELLING, SplitRatio = 0.70)
dataset_maindwelling_2_training_set <- subset(dataset_maindwelling_2, split == TRUE)
dataset_maindwelling_2_test_set <- subset(dataset_maindwelling_2, split == FALSE)
# Maindwelling 3
set.seed(123)
split = sample.split(dataset_maindwelling_3$MAINDWELLING, SplitRatio = 0.70)
dataset_maindwelling_3_training_set <- subset(dataset_maindwelling_3, split == TRUE)
dataset_maindwelling_3_test_set <- subset(dataset_maindwelling_3, split == FALSE)
# Maindwelling 4
set.seed(123)
split = sample.split(dataset_maindwelling_4$MAINDWELLING, SplitRatio = 0.70)
dataset_maindwelling_4_training_set <- subset(dataset_maindwelling_4, split == TRUE)
dataset_maindwelling_4_test_set <- subset(dataset_maindwelling_4, split == FALSE)
# Maindwelling 5
set.seed(123)
split = sample.split(dataset_maindwelling_5$MAINDWELLING, SplitRatio = 0.70)
dataset_maindwelling_5_training_set <- subset(dataset_maindwelling_5, split == TRUE)
dataset_maindwelling_5_test_set <- subset(dataset_maindwelling_5, split == FALSE)
# Maindwelling 6
set.seed(123)
split = sample.split(dataset_maindwelling_6$MAINDWELLING, SplitRatio = 0.70)
dataset_maindwelling_6_training_set <- subset(dataset_maindwelling_6, split == TRUE)
dataset_maindwelling_6_test_set <- subset(dataset_maindwelling_6, split == FALSE)
# Maindwelling 7
set.seed(123)
```

```
split = sample.split(dataset_maindwelling_7$MAINDWELLING, SplitRatio = 0.70)

dataset_maindwelling_7_training_set <- subset(dataset_maindwelling_7, split == TRUE)

dataset_maindwelling_7_test_set <- subset(dataset_maindwelling_7, split == FALSE)

# Maindwelling 8

set.seed(123)

split = sample.split(dataset_maindwelling_8$MAINDWELLING, SplitRatio = 0.70)

dataset_maindwelling_8_training_set <- subset(dataset_maindwelling_8, split == TRUE)

dataset_maindwelling_8_test_set <- subset(dataset_maindwelling_8, split == FALSE)

# Maindwelling 9

set.seed(123)

split = sample.split(dataset_maindwelling_9$MAINDWELLING, SplitRatio = 0.70)

dataset_maindwelling_9_training_set <- subset(dataset_maindwelling_9, split == TRUE)

dataset_maindwelling_9_test_set <- subset(dataset_maindwelling_9, split == FALSE)


# Recombine into single training set

dataset_maindwelling_global_training_set <- rbind(dataset_maindwelling_1_training_set,
dataset_maindwelling_2_training_set,dataset_maindwelling_3_training_set,dataset_maindwelling_4_training_set,dat
aset_maindwelling_5_training_set,dataset_maindwelling_6_training_set,dataset_maindwelling_7_training_set,datase
t_maindwelling_8_training_set,dataset_maindwelling_9_training_set)


# Recombine into single test set

dataset_maindwelling_global_test_set <- rbind(dataset_maindwelling_1_test_set,
dataset_maindwelling_2_test_set,dataset_maindwelling_3_test_set,dataset_maindwelling_4_test_set,dataset_maindwe
lling_5_test_set,dataset_maindwelling_6_test_set,dataset_maindwelling_7_test_set,dataset_maindwelling_8_test_se
t,dataset_maindwelling_9_test_set)


# Remove unnecessary data frames

# Unnecessary Maindwelling data frames

rm(dataset_maindwelling_1,dataset_maindwelling_2,dataset_maindwelling_3,dataset_maindwelling_4,dataset_maindwel
ling_5,dataset_maindwelling_6,dataset_maindwelling_7,dataset_maindwelling_8,dataset_maindwelling_9)

# Unnecessary training set data frames

rm(dataset_maindwelling_1_training_set,dataset_maindwelling_2_training_set,dataset_maindwelling_3_training_set,
dataset_maindwelling_4_training_set,dataset_maindwelling_5_training_set,dataset_maindwelling_6_training_set,dat
aset_maindwelling_7_training_set,dataset_maindwelling_8_training_set,dataset_maindwelling_9_training_set)

# Unnecessary test set data frames

rm(dataset_maindwelling_1_test_set,
dataset_maindwelling_2_test_set,dataset_maindwelling_3_test_set,dataset_maindwelling_4_test_set,dataset_maindwe
lling_5_test_set,dataset_maindwelling_6_test_set,dataset_maindwelling_7_test_set,dataset_maindwelling_8_test_se
t,dataset_maindwelling_9_test_set)


## Part 5 - Making the Dependent Variable a Factor

dataset_maindwelling_global_training_set$MAINDWELLING <-
as.factor(dataset_maindwelling_global_training_set$MAINDWELLING)

dataset_maindwelling_global_test_set$MAINDWELLING <-
as.factor(dataset_maindwelling_global_test_set$MAINDWELLING)


## Part 6 - Feature Scaling

dataset_maindwelling_global_training_set[,2] = scale(dataset_maindwelling_global_training_set[,2])

dataset_maindwelling_global_test_set[,2] = scale(dataset_maindwelling_global_test_set[,2])


## Part 7 - Creating the Random Forest Model

library(randomForest)

set.seed(222)

rf <- randomForest(MAINDWELLING~.,data=dataset_maindwelling_global_training_set)
```

```
print(rf)
## Part 8 - Publishing the Results (Model 7)

# Histogram showing No. of nodes for trees (Model 7)
hist(treesize(rf),
     main = "No. of Nodes for the Trees",
     col = "SkyBlue")
# Variable importance (Model 7)
varImpPlot(rf)
importance(rf)

# Installation of caret
# install.packages('caret', dependencies = TRUE)
# Installation of caret dependencies not identified (lava)
# install.packages('lava')
# library(lava)

# Prediction & Confusion Matrix (Model 7)
library(caret)
p1 <- predict(rf, dataset_maindwelling_global_training_set)
confusionMatrix(p1, dataset_maindwelling_global_training_set$MAINDWELLING)
p2 <- predict(rf, dataset_maindwelling_global_test_set)
confusionMatrix(p2, dataset_maindwelling_global_test_set$MAINDWELLING)

## Part 9 - Tuning the Random Forest Model
# Visualise Error Rate of Random Forest
plot(rf)
# Tuning mtry
t <- tuneRF(dataset_maindwelling_global_training_set[,-1],dataset_maindwelling_global_training_set[,1],
            stepFactor = 0.5,
            plot = TRUE,
            ntreeTry = 150,
            trace = TRUE,
            improve = 0.05)

## Part 10 - Creating the updated Random Forest Model (Model 8)
library(randomForest)
set.seed(222)
rf<- randomForest(MAINDWELLING~.,dataset_maindwelling_global_training_set,
                  ntree = 150,
                  mtry = 2,
                  importance = TRUE)
print(rf)

## Part 11 - Publishing the Results of the New Model (Model 8)
# Histogram showing No. of nodes for trees (Model 8)
hist(treesize(rf),
     main = "No. of Nodes for the Trees",
     col = "SkyBlue")
```

```
# Variable importance (Model 8)
varImpPlot(rf)
importance(rf)
# Prediction & Confusion Matrix (Model 8)
library(caret)
p1 <- predict(rf, dataset_maindwelling_global_training_set)
confusionMatrix(p1, dataset_maindwelling_global_training_set$MAINDWELLING)
p2 <- predict(rf, dataset_maindwelling_global_test_set)
confusionMatrix(p2, dataset_maindwelling_global_test_set$MAINDWELLING)
```

MAINDWELLING – Model 9 & 10 (80:20 Training:Test Split, with Stratified Sampling)

```
## Part 1 - Importing the dataset
dataset = read.csv('Census_2011_Household_CT_Hgth_Enc.csv')

## Part 2 - Selecting the appropriate variables
dataset_maindwelling <- subset(dataset,
                        select=c("MAINDWELLING","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))

## Part 3 - Generating a Frequency Table for the Dependent Variable
table(dataset_maindwelling$MAINDWELLING)

## PART 4 - stratified sampling

# Extract maindwelling by class
dataset_maindwelling_1 <- subset(dataset_maindwelling, MAINDWELLING=="1",
                           select=c("MAINDWELLING","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))
dataset_maindwelling_2 <- subset(dataset_maindwelling, MAINDWELLING=="2",
                           select=c("MAINDWELLING","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))
dataset_maindwelling_3 <- subset(dataset_maindwelling, MAINDWELLING=="3",
                           select=c("MAINDWELLING","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))
dataset_maindwelling_4 <- subset(dataset_maindwelling, MAINDWELLING=="4",
                           select=c("MAINDWELLING","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))
dataset_maindwelling_5 <- subset(dataset_maindwelling, MAINDWELLING=="5",
                           select=c("MAINDWELLING","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))
dataset_maindwelling_6 <- subset(dataset_maindwelling, MAINDWELLING=="6",
                           select=c("MAINDWELLING","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))
dataset_maindwelling_7 <- subset(dataset_maindwelling, MAINDWELLING=="7",
                           select=c("MAINDWELLING","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))
dataset_maindwelling_8 <- subset(dataset_maindwelling, MAINDWELLING=="8",
                           select=c("MAINDWELLING","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))
dataset_maindwelling_9 <- subset(dataset_maindwelling, MAINDWELLING=="9",
                           select=c("MAINDWELLING","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))

# split maindwelling into training and test sets
# install.packages('caTools')
library(caTools)
```

```
# Maindwelling 1
set.seed(123)
split = sample.split(dataset_maindwelling_1$MAINDWELLING, SplitRatio = 0.80)
dataset_maindwelling_1_training_set <- subset(dataset_maindwelling_1, split == TRUE)
dataset_maindwelling_1_test_set <- subset(dataset_maindwelling_1, split == FALSE)
# Maindwelling 2
set.seed(123)
split = sample.split(dataset_maindwelling_2$MAINDWELLING, SplitRatio = 0.80)
dataset_maindwelling_2_training_set <- subset(dataset_maindwelling_2, split == TRUE)
dataset_maindwelling_2_test_set <- subset(dataset_maindwelling_2, split == FALSE)
# Maindwelling 3
set.seed(123)
split = sample.split(dataset_maindwelling_3$MAINDWELLING, SplitRatio = 0.80)
dataset_maindwelling_3_training_set <- subset(dataset_maindwelling_3, split == TRUE)
dataset_maindwelling_3_test_set <- subset(dataset_maindwelling_3, split == FALSE)
# Maindwelling 4
set.seed(123)
split = sample.split(dataset_maindwelling_4$MAINDWELLING, SplitRatio = 0.80)
dataset_maindwelling_4_training_set <- subset(dataset_maindwelling_4, split == TRUE)
dataset_maindwelling_4_test_set <- subset(dataset_maindwelling_4, split == FALSE)
# Maindwelling 5
set.seed(123)
split = sample.split(dataset_maindwelling_5$MAINDWELLING, SplitRatio = 0.80)
dataset_maindwelling_5_training_set <- subset(dataset_maindwelling_5, split == TRUE)
dataset_maindwelling_5_test_set <- subset(dataset_maindwelling_5, split == FALSE)
# Maindwelling 6
set.seed(123)
split = sample.split(dataset_maindwelling_6$MAINDWELLING, SplitRatio = 0.80)
dataset_maindwelling_6_training_set <- subset(dataset_maindwelling_6, split == TRUE)
dataset_maindwelling_6_test_set <- subset(dataset_maindwelling_6, split == FALSE)
# Maindwelling 7
set.seed(123)
split = sample.split(dataset_maindwelling_7$MAINDWELLING, SplitRatio = 0.80)
dataset_maindwelling_7_training_set <- subset(dataset_maindwelling_7, split == TRUE)
dataset_maindwelling_7_test_set <- subset(dataset_maindwelling_7, split == FALSE)
# Maindwelling 8
set.seed(123)
split = sample.split(dataset_maindwelling_8$MAINDWELLING, SplitRatio = 0.80)
dataset_maindwelling_8_training_set <- subset(dataset_maindwelling_8, split == TRUE)
dataset_maindwelling_8_test_set <- subset(dataset_maindwelling_8, split == FALSE)
# Maindwelling 9
set.seed(123)
split = sample.split(dataset_maindwelling_9$MAINDWELLING, SplitRatio = 0.80)
dataset_maindwelling_9_training_set <- subset(dataset_maindwelling_9, split == TRUE)
dataset_maindwelling_9_test_set <- subset(dataset_maindwelling_9, split == FALSE)

# Recombine into single training set
dataset_maindwelling_global_training_set <- rbind(dataset_maindwelling_1_training_set,
```

```
dataset_maindwelling_2_training_set,dataset_maindwelling_3_training_set,dataset_maindwelling_4_training_set,dat
aset_maindwelling_5_training_set,dataset_maindwelling_6_training_set,dataset_maindwelling_7_training_set,datase
t_maindwelling_8_training_set,dataset_maindwelling_9_training_set)


# Recombine into single test set

dataset_maindwelling_global_test_set <- rbind(dataset_maindwelling_1_test_set,
dataset_maindwelling_2_test_set,dataset_maindwelling_3_test_set,dataset_maindwelling_4_test_set,dataset_maindwe
lling_5_test_set,dataset_maindwelling_6_test_set,dataset_maindwelling_7_test_set,dataset_maindwelling_8_test_se
t,dataset_maindwelling_9_test_set)


# Remove unnecessary data frames

# Unnecessary Maindwelling data frames

rm(dataset_maindwelling_1,dataset_maindwelling_2,dataset_maindwelling_3,dataset_maindwelling_4,dataset_maindwel
ling_5,dataset_maindwelling_6,dataset_maindwelling_7,dataset_maindwelling_8,dataset_maindwelling_9)

# Unnecessary training set data frames

rm(dataset_maindwelling_1_training_set,dataset_maindwelling_2_training_set,dataset_maindwelling_3_training_set,
dataset_maindwelling_4_training_set,dataset_maindwelling_5_training_set,dataset_maindwelling_6_training_set,dat
aset_maindwelling_7_training_set,dataset_maindwelling_8_training_set,dataset_maindwelling_9_training_set)

# Unnecessary test set data frames

rm(dataset_maindwelling_1_test_set,
dataset_maindwelling_2_test_set,dataset_maindwelling_3_test_set,dataset_maindwelling_4_test_set,dataset_maindwe
lling_5_test_set,dataset_maindwelling_6_test_set,dataset_maindwelling_7_test_set,dataset_maindwelling_8_test_se
t,dataset_maindwelling_9_test_set)


## Part 5 - Making the Dependent Variable a Factor

dataset_maindwelling_global_training_set$MAINDWELLING <-
as.factor(dataset_maindwelling_global_training_set$MAINDWELLING)

dataset_maindwelling_global_test_set$MAINDWELLING <-
as.factor(dataset_maindwelling_global_test_set$MAINDWELLING)


## Part 6 - Feature Scaling

dataset_maindwelling_global_training_set[,2] = scale(dataset_maindwelling_global_training_set[,2])

dataset_maindwelling_global_test_set[,2] = scale(dataset_maindwelling_global_test_set[,2])


## Part 7 - Creating the Random Forest Model

library(randomForest)

set.seed(222)

rf <- randomForest(MAINDWELLING~.,data=dataset_maindwelling_global_training_set)

print(rf)


## Part 8 - Publishing the Results (Model 9)


# Histogram showing No. of nodes for trees (Model 9)

hist(treesize(rf),

     main = "No. of Nodes for the Trees",

     col = "SkyBlue")

# Variable importance (Model 9)

varImpPlot(rf)

importance(rf)


# Installation of caret

# install.packages('caret', dependencies = TRUE)

# Installation of caret dependencies not identified (lava)

# install.packages('lava')
```

```
# library(lava)
# Prediction & Confusion Matrix (Model 9)
library(caret)
p1 <- predict(rf, dataset_maindwelling_global_training_set)
confusionMatrix(p1, dataset_maindwelling_global_training_set$MAINDWELLING)
p2 <- predict(rf, dataset_maindwelling_global_test_set)
confusionMatrix(p2, dataset_maindwelling_global_test_set$MAINDWELLING)


## Part 9 - Tuning the Random Forest Model
# Visualise Error Rate of Random Forest
plot(rf)
# Tuning mtry
t <- tuneRF(dataset_maindwelling_global_training_set[,-1],dataset_maindwelling_global_training_set[,1],
            stepFactor = 0.5,
            plot = TRUE,
            ntreeTry = 150,
            trace = TRUE,
            improve = 0.05)

## Part 10 - Creating the updated Random Forest Model (Model 10)
library(randomForest)
set.seed(222)
rf<- randomForest(MAINDWELLING~.,dataset_maindwelling_global_training_set,
                  ntree = 150,
                  mtry = 2,
                  importance = TRUE)
print(rf)

## Part 11 - Publishing the Results of the New Model (Model 10)
# Histogram showing No. of nodes for trees (Model 10)
hist(treesize(rf),
     main = "No. of Nodes for the Trees",
     col = "SkyBlue")
# Variable importance (Model 10)
varImpPlot(rf)
importance(rf)
# Prediction & Confusion Matrix (Model 10)
library(caret)
p1 <- predict(rf, dataset_maindwelling_global_training_set)
confusionMatrix(p1, dataset_maindwelling_global_training_set$MAINDWELLING)
p2 <- predict(rf, dataset_maindwelling_global_test_set)
confusionMatrix(p2, dataset_maindwelling_global_test_set$MAINDWELLING)
```

MAINDWELLING – Model 11 & 12 (90:10 Training:Test Split, with Stratified Sampling)

```
## Part 1 - Importing the dataset
dataset = read.csv('Census_2011_Household_CT_Hgth_Enc.csv')
```

```r
## Part 2 - Selecting the appropriate variables
dataset_maindwelling <- subset(dataset,
                        select=c("MAINDWELLING","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))

## Part 3 - Generating a Frequency Table for the Dependent Variable
table(dataset_maindwelling$MAINDWELLING)

## PART 4 - stratified sampling

# Extract maindwelling by class
dataset_maindwelling_1 <- subset(dataset_maindwelling, MAINDWELLING=="1",
                           select=c("MAINDWELLING","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))
dataset_maindwelling_2 <- subset(dataset_maindwelling, MAINDWELLING=="2",
                           select=c("MAINDWELLING","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))
dataset_maindwelling_3 <- subset(dataset_maindwelling, MAINDWELLING=="3",
                           select=c("MAINDWELLING","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))
dataset_maindwelling_4 <- subset(dataset_maindwelling, MAINDWELLING=="4",
                           select=c("MAINDWELLING","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))
dataset_maindwelling_5 <- subset(dataset_maindwelling, MAINDWELLING=="5",
                           select=c("MAINDWELLING","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))
dataset_maindwelling_6 <- subset(dataset_maindwelling, MAINDWELLING=="6",
                           select=c("MAINDWELLING","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))
dataset_maindwelling_7 <- subset(dataset_maindwelling, MAINDWELLING=="7",
                           select=c("MAINDWELLING","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))
dataset_maindwelling_8 <- subset(dataset_maindwelling, MAINDWELLING=="8",
                           select=c("MAINDWELLING","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))
dataset_maindwelling_9 <- subset(dataset_maindwelling, MAINDWELLING=="9",
                           select=c("MAINDWELLING","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))

# split maindwelling into training and test sets
# install.packages('caTools')
library(caTools)
# Maindwelling 1
set.seed(123)
split = sample.split(dataset_maindwelling_1$MAINDWELLING, SplitRatio = 0.90)
dataset_maindwelling_1_training_set <- subset(dataset_maindwelling_1, split == TRUE)
dataset_maindwelling_1_test_set <- subset(dataset_maindwelling_1, split == FALSE)
# Maindwelling 2
set.seed(123)
split = sample.split(dataset_maindwelling_2$MAINDWELLING, SplitRatio = 0.90)
dataset_maindwelling_2_training_set <- subset(dataset_maindwelling_2, split == TRUE)
dataset_maindwelling_2_test_set <- subset(dataset_maindwelling_2, split == FALSE)
# Maindwelling 3
set.seed(123)
split = sample.split(dataset_maindwelling_3$MAINDWELLING, SplitRatio = 0.90)
dataset_maindwelling_3_training_set <- subset(dataset_maindwelling_3, split == TRUE)
dataset_maindwelling_3_test_set <- subset(dataset_maindwelling_3, split == FALSE)
# Maindwelling 4
```

```
set.seed(123)
split = sample.split(dataset_maindwelling_4$MAINDWELLING, SplitRatio = 0.90)
dataset_maindwelling_4_training_set <- subset(dataset_maindwelling_4, split == TRUE)
dataset_maindwelling_4_test_set <- subset(dataset_maindwelling_4, split == FALSE)
# Maindwelling 5
set.seed(123)
split = sample.split(dataset_maindwelling_5$MAINDWELLING, SplitRatio = 0.90)
dataset_maindwelling_5_training_set <- subset(dataset_maindwelling_5, split == TRUE)
dataset_maindwelling_5_test_set <- subset(dataset_maindwelling_5, split == FALSE)
# Maindwelling 6
set.seed(123)
split = sample.split(dataset_maindwelling_6$MAINDWELLING, SplitRatio = 0.90)
dataset_maindwelling_6_training_set <- subset(dataset_maindwelling_6, split == TRUE)
dataset_maindwelling_6_test_set <- subset(dataset_maindwelling_6, split == FALSE)
# Maindwelling 7
set.seed(123)
split = sample.split(dataset_maindwelling_7$MAINDWELLING, SplitRatio = 0.90)
dataset_maindwelling_7_training_set <- subset(dataset_maindwelling_7, split == TRUE)
dataset_maindwelling_7_test_set <- subset(dataset_maindwelling_7, split == FALSE)
# Maindwelling 8
set.seed(123)
split = sample.split(dataset_maindwelling_8$MAINDWELLING, SplitRatio = 0.90)
dataset_maindwelling_8_training_set <- subset(dataset_maindwelling_8, split == TRUE)
dataset_maindwelling_8_test_set <- subset(dataset_maindwelling_8, split == FALSE)
# Maindwelling 9
set.seed(123)
split = sample.split(dataset_maindwelling_9$MAINDWELLING, SplitRatio = 0.90)
dataset_maindwelling_9_training_set <- subset(dataset_maindwelling_9, split == TRUE)
dataset_maindwelling_9_test_set <- subset(dataset_maindwelling_9, split == FALSE)


# Recombine into single training set
dataset_maindwelling_global_training_set <- rbind(dataset_maindwelling_1_training_set,
dataset_maindwelling_2_training_set,dataset_maindwelling_3_training_set,dataset_maindwelling_4_training_set,dat
aset_maindwelling_5_training_set,dataset_maindwelling_6_training_set,dataset_maindwelling_7_training_set,datase
t_maindwelling_8_training_set,dataset_maindwelling_9_training_set)


# Recombine into single test set
dataset_maindwelling_global_test_set <- rbind(dataset_maindwelling_1_test_set,
dataset_maindwelling_2_test_set,dataset_maindwelling_3_test_set,dataset_maindwelling_4_test_set,dataset_maindwe
lling_5_test_set,dataset_maindwelling_6_test_set,dataset_maindwelling_7_test_set,dataset_maindwelling_8_test_se
t,dataset_maindwelling_9_test_set)


# Remove unnecessary data frames
# Unnecessary Maindwelling data frames
rm(dataset_maindwelling_1,dataset_maindwelling_2,dataset_maindwelling_3,dataset_maindwelling_4,dataset_maindwel
ling_5,dataset_maindwelling_6,dataset_maindwelling_7,dataset_maindwelling_8,dataset_maindwelling_9)
# Unnecessary training set data frames
rm(dataset_maindwelling_1_training_set,dataset_maindwelling_2_training_set,dataset_maindwelling_3_training_set,
dataset_maindwelling_4_training_set,dataset_maindwelling_5_training_set,dataset_maindwelling_6_training_set,dat
aset_maindwelling_7_training_set,dataset_maindwelling_8_training_set,dataset_maindwelling_9_training_set)
# Unnecessary test set data frames
rm(dataset_maindwelling_1_test_set,
dataset_maindwelling_2_test_set,dataset_maindwelling_3_test_set,dataset_maindwelling_4_test_set,dataset_maindwe
```

```
lling_5_test_set,dataset_maindwelling_6_test_set,dataset_maindwelling_7_test_set,dataset_maindwelling_8_test_se
t,dataset_maindwelling_9_test_set)


## Part 5 - Making the Dependent Variable a Factor

dataset_maindwelling_global_training_set$MAINDWELLING <-
as.factor(dataset_maindwelling_global_training_set$MAINDWELLING)

dataset_maindwelling_global_test_set$MAINDWELLING <-
as.factor(dataset_maindwelling_global_test_set$MAINDWELLING)


## Part 6 - Feature Scaling

dataset_maindwelling_global_training_set[,2] = scale(dataset_maindwelling_global_training_set[,2])

dataset_maindwelling_global_test_set[,2] = scale(dataset_maindwelling_global_test_set[,2])


## Part 7 - Creating the Random Forest Model

library(randomForest)

set.seed(222)

rf <- randomForest(MAINDWELLING~.,data=dataset_maindwelling_global_training_set)

print(rf)


## Part 8 - Publishing the Results (Model 11)


# Histogram showing No. of nodes for trees (Model 11)

hist(treesize(rf),

     main = "No. of Nodes for the Trees",

     col = "SkyBlue")

# Variable importance (Model 11)

varImpPlot(rf)

importance(rf)


# Installation of caret

# install.packages('caret', dependencies = TRUE)

# Installation of caret dependencies not identified (lava)

# install.packages('lava')

# library(lava)


# Prediction & Confusion Matrix (Model 11)

library(caret)

p1 <- predict(rf, dataset_maindwelling_global_training_set)

confusionMatrix(p1, dataset_maindwelling_global_training_set$MAINDWELLING)

p2 <- predict(rf, dataset_maindwelling_global_test_set)

confusionMatrix(p2, dataset_maindwelling_global_test_set$MAINDWELLING)



## Part 9 - Tuning the Random Forest Model

# Visualise Error Rate of Random Forest

plot(rf)

# Tuning mtry

t <- tuneRF(dataset_maindwelling_global_training_set[,-1],dataset_maindwelling_global_training_set[,1],

            stepFactor = 0.5,

            plot = TRUE,
```

```
            ntreeTry = 150,
            trace = TRUE,
            improve = 0.05)

## Part 10 - Creating the updated Random Forest Model (Model 12)
library(randomForest)
set.seed(222)
rf<- randomForest(MAINDWELLING~.,dataset_maindwelling_global_training_set,
                  ntree = 150,
                  mtry = 2,
                  importance = TRUE)
print(rf)

## Part 11 - Publishing the Results of the New Model (Model 12)
# Histogram showing No. of nodes for trees (Model 12)
hist(treesize(rf),
     main = "No. of Nodes for the Trees",
     col = "SkyBlue")
# Variable importance (Model 12)
varImpPlot(rf)
importance(rf)
# Prediction & Confusion Matrix (Model 12)
library(caret)
p1 <- predict(rf, dataset_maindwelling_global_training_set)
confusionMatrix(p1, dataset_maindwelling_global_training_set$MAINDWELLING)
p2 <- predict(rf, dataset_maindwelling_global_test_set)
confusionMatrix(p2, dataset_maindwelling_global_test_set$MAINDWELLING)
```

TENURE – Model 13 & 14 (70:30 Training:Test Split)

```
## Part 1 - Importing the dataset
dataset = read.csv('Census_2011_Household_CT_Hgth_Enc.csv')

## Part 2 - Selecting the appropriate variables
dataset_tenure <- subset(dataset, select=c("TENURE","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))

## Part 3 – Generating a Frequency Table for the Dependent Variable
table(dataset_tenure$TENURE)

## Part 4 - Splitting the dataset into the Training Set and Test Set
#install.packages('caTools')
library(caTools)
set.seed(123)
split = sample.split(dataset_tenure$TENURE, SplitRatio = 0.70)
tenure_training_set <- subset(dataset_tenure, split == TRUE)
tenure_test_set <- subset(dataset_tenure, split == FALSE)

## Part 5 – Making the Dependent Variable a Factor
```

```
tenure_training_set$TENURE <- as.factor(tenure_training_set$TENURE)
tenure_test_set$TENURE <- as.factor(tenure_test_set$TENURE)

## Part 6 - Feature Scaling
tenure_training_set[,2] = scale(tenure_training_set[,2])
tenure_test_set[,2] = scale(tenure_test_set[,2])

## Part 7 – Creating the Random Forest Model
library(randomForest)
set.seed(222)
rf <- randomForest(TENURE~.,data=tenure_training_set)
print(rf)

## Part 8 – Publishing the Results (Model 13)
# Installation of caret
install.packages('caret', dependencies = TRUE)
# Lava error, install lava
#install.packages('lava')
# Histogram showing No. of nodes for trees (Model 13)
hist(treesize(rf),
     main = "No. of Nodes for the Trees",
     col = "SkyBlue")
# Variable importance (Model 13)
varImpPlot(rf)
importance(rf)
# Prediction & Confusion Matrix (Model 13)
library(caret)
p1 <- predict(rf, tenure_training_set)
confusionMatrix(p1, tenure_training_set$TENURE)
p2 <- predict(rf, tenure_test_set)
confusionMatrix(p2, tenure_test_set$TENURE)

## Part 9 – Tuning the Random Forest Model
# Visualise Error Rate of Random Forest
plot(rf)
# Tuning mtry
t <- tuneRF(tenure_training_set[,-1],tenure_training_set[,1],
       stepFactor = 0.5,
       plot = TRUE,
       ntreeTry = 150,
       trace = TRUE,
       improve = 0.05)

## Part 10 – Creating the updated Random Forest Model (Model 14)
library(randomForest)
set.seed(222)
rf<- randomForest(TENURE~.,data=tenure_training_set,
                  ntree = 150,
                  mtry = 2,
```

```
                    importance = TRUE)
print(rf)

## Part 11 – Publishing the Results of the New Model (Model 14)
# Histogram showing No. of nodes for trees (Model 14)
hist(treesize(rf),
     main = "No. of Nodes for the Trees",
     col = "SkyBlue")
# Variable importance (Model 14)
varImpPlot(rf)
importance(rf)
# Prediction & Confusion Matrix (Model 14)
library(caret)
p1 <- predict(rf, tenure_training_set)
confusionMatrix(p1, tenure_training_set$TENURE)
p2 <- predict(rf, tenure_test_set)
confusionMatrix(p2, tenure_test_set$TENURE)
```

TENURE – Model 15 & 16 (80:20 Training:Test Split)

```
## Part 1 - Importing the dataset
dataset = read.csv('Census_2011_Household_CT_Hgth_Enc.csv')

## Part 2 - Selecting the appropriate variables
dataset_tenure <- subset(dataset, select=c("TENURE","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))

## Part 3 – Generating a Frequency Table for the Dependent Variable
table(dataset_tenure$TENURE)

## Part 4 - Splitting the dataset into the Training Set and Test Set
#install.packages('caTools')
library(caTools)
set.seed(123)
split = sample.split(dataset_tenure$TENURE, SplitRatio = 0.80)
tenure_training_set <- subset(dataset_tenure, split == TRUE)
tenure_test_set <- subset(dataset_tenure, split == FALSE)

## Part 5 – Making the Dependent Variable a Factor
tenure_training_set$TENURE <- as.factor(tenure_training_set$TENURE)
tenure_test_set$TENURE <- as.factor(tenure_test_set$TENURE)

## Part 6 - Feature Scaling
tenure_training_set[,2] = scale(tenure_training_set[,2])
tenure_test_set[,2] = scale(tenure_test_set[,2])

## Part 7 – Creating the Random Forest Model
library(randomForest)
set.seed(222)
```

```
rf <- randomForest(TENURE~.,data=tenure_training_set)
print(rf)

## Part 8 – Publishing the Results (Model 15)
# Installation of caret
install.packages('caret', dependencies = TRUE)
# Lava error, install lava
#install.packages('lava')
# Histogram showing No. of nodes for trees (Model 15)
hist(treesize(rf),
     main = "No. of Nodes for the Trees",
     col = "SkyBlue")
# Variable importance (Model 15)
varImpPlot(rf)
importance(rf)
# Prediction & Confusion Matrix (Model 15)
library(caret)
p1 <- predict(rf, tenure_training_set)
confusionMatrix(p1, tenure_training_set$TENURE)
p2 <- predict(rf, tenure_test_set)
confusionMatrix(p2, tenure_test_set$TENURE)

## Part 9 – Tuning the Random Forest Model
# Visualise Error Rate of Random Forest
plot(rf)
# Tuning mtry
t <- tuneRF(tenure_training_set[,-1],tenure_training_set[,1],
       stepFactor = 0.5,
       plot = TRUE,
       ntreeTry = 150,
       trace = TRUE,
       improve = 0.05)

## Part 10 – Creating the updated Random Forest Model (Model 16)
library(randomForest)
set.seed(222)
rf<- randomForest(TENURE~.,data=tenure_training_set,
                  ntree = 150,
                  mtry = 2,
                  importance = TRUE)
print(rf)

## Part 11 – Publishing the Results of the New Model (Model 16)
# Histogram showing No. of nodes for trees (Model 16)
hist(treesize(rf),
     main = "No. of Nodes for the Trees",
     col = "SkyBlue")
# Variable importance (Model 16)
varImpPlot(rf)
```

```
importance(rf)



# Prediction & Confusion Matrix (Model 16)
library(caret)
p1 <- predict(rf, tenure_training_set)
confusionMatrix(p1, tenure_training_set$TENURE)
p2 <- predict(rf, tenure_test_set)
confusionMatrix(p2, tenure_test_set$TENURE)
```

TENURE – Model 17 & 18 (90:10 Training:Test Split)

```
## Part 1 - Importing the dataset
dataset = read.csv('Census_2011_Household_CT_Hgth_Enc.csv')

## Part 2 - Selecting the appropriate variables
dataset_tenure <- subset(dataset, select=c("TENURE","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))

## Part 3 – Generating a Frequency Table for the Dependent Variable
table(dataset_tenure$TENURE)

## Part 4 - Splitting the dataset into the Training Set and Test Set
#install.packages('caTools')
library(caTools)
set.seed(123)
split = sample.split(dataset_tenure$TENURE, SplitRatio = 0.90)
tenure_training_set <- subset(dataset_tenure, split == TRUE)
tenure_test_set <- subset(dataset_tenure, split == FALSE)

## Part 5 – Making the Dependent Variable a Factor
tenure_training_set$TENURE <- as.factor(tenure_training_set$TENURE)
tenure_test_set$TENURE <- as.factor(tenure_test_set$TENURE)

## Part 6 - Feature Scaling
tenure_training_set[,2] = scale(tenure_training_set[,2])
tenure_test_set[,2] = scale(tenure_test_set[,2])

## Part 7 – Creating the Random Forest Model
library(randomForest)
set.seed(222)
rf <- randomForest(TENURE~.,data=tenure_training_set)
print(rf)

## Part 8 – Publishing the Results (Model 17)
# Installation of caret
install.packages('caret', dependencies = TRUE)
# Lava error, install lava
#install.packages('lava')
```

```
# Histogram showing No. of nodes for trees (Model 17)
hist(treesize(rf),
     main = "No. of Nodes for the Trees",
     col = "SkyBlue")
# Variable importance (Model 17)
varImpPlot(rf)
importance(rf)
# Prediction & Confusion Matrix (Model 17)
library(caret)
p1 <- predict(rf, tenure_training_set)
confusionMatrix(p1, tenure_training_set$TENURE)
p2 <- predict(rf, tenure_test_set)
confusionMatrix(p2, tenure_test_set$TENURE)

## Part 9 – Tuning the Random Forest Model
# Visualise Error Rate of Random Forest
plot(rf)
# Tuning mtry
t <- tuneRF(tenure_training_set[,-1],tenure_training_set[,1],
       stepFactor = 0.5,
       plot = TRUE,
       ntreeTry = 150,
       trace = TRUE,
       improve = 0.05)

## Part 10 – Creating the updated Random Forest Model (Model 18)
library(randomForest)
set.seed(222)
rf<- randomForest(TENURE~.,data=tenure_training_set,
                  ntree = 150,
                  mtry = 2,
                  importance = TRUE)
print(rf)

## Part 11 – Publishing the Results of the New Model (Model 18)
# Histogram showing No. of nodes for trees (Model 18)
hist(treesize(rf),
     main = "No. of Nodes for the Trees",
     col = "SkyBlue")
# Variable importance (Model 18)
varImpPlot(rf)
importance(rf)
# Prediction & Confusion Matrix (Model 18)
library(caret)
p1 <- predict(rf, tenure_training_set)
confusionMatrix(p1, tenure_training_set$TENURE)
p2 <- predict(rf, tenure_test_set)
confusionMatrix(p2, tenure_test_set$TENURE)
```

TENURE – Model 19 & 20 (70:30 Training:Test Split, with Stratified Sampling)

```
## Part 1 - Importing the dataset
dataset = read.csv('Census_2011_Household_CT_Hgth_Enc.csv')

## Part 2 - Selecting the appropriate variables
dataset_tenure <- subset(dataset,
                         select=c("TENURE","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))

## Part 3 - Generating a Frequency Table for the Dependent Variable
table(dataset_tenure$TENURE)

## PART 4 - stratified sampling

# Extract tenure by class
dataset_tenure_1 <- subset(dataset_tenure, TENURE=="1",
                           select=c("TENURE","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))
dataset_tenure_2 <- subset(dataset_tenure, TENURE=="2",
                           select=c("TENURE","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))
dataset_tenure_3 <- subset(dataset_tenure, TENURE=="3",
                           select=c("TENURE","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))
dataset_tenure_4 <- subset(dataset_tenure, TENURE=="4",
                           select=c("TENURE","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))
dataset_tenure_5 <- subset(dataset_tenure, TENURE=="5",
                           select=c("TENURE","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))

# split tenure into training and test sets
# install.packages('caTools')
library(caTools)
# Tenure 1
set.seed(123)
split = sample.split(dataset_tenure_1$TENURE, SplitRatio = 0.70)
dataset_tenure_1_training_set <- subset(dataset_tenure_1, split == TRUE)
dataset_tenure_1_test_set <- subset(dataset_tenure_1, split == FALSE)
# Tenure 2
set.seed(123)
split = sample.split(dataset_tenure_2$TENURE, SplitRatio = 0.70)
dataset_tenure_2_training_set <- subset(dataset_tenure_2, split == TRUE)
dataset_tenure_2_test_set <- subset(dataset_tenure_2, split == FALSE)
# Tenure 3
set.seed(123)
split = sample.split(dataset_tenure_3$TENURE, SplitRatio = 0.70)
dataset_tenure_3_training_set <- subset(dataset_tenure_3, split == TRUE)
dataset_tenure_3_test_set <- subset(dataset_tenure_3, split == FALSE)
# Tenure 4
set.seed(123)
split = sample.split(dataset_tenure_4$TENURE, SplitRatio = 0.70)
```

```
dataset_tenure_4_training_set <- subset(dataset_tenure_4, split == TRUE)

dataset_tenure_4_test_set <- subset(dataset_tenure_4, split == FALSE)

# Tenure 5

set.seed(123)

split = sample.split(dataset_tenure_5$TENURE, SplitRatio = 0.70)

dataset_tenure_5_training_set <- subset(dataset_tenure_5, split == TRUE)

dataset_tenure_5_test_set <- subset(dataset_tenure_5, split == FALSE)


# Recombine into single training set

dataset_tenure_global_training_set <- rbind(dataset_tenure_1_training_set,
dataset_tenure_2_training_set,dataset_tenure_3_training_set,dataset_tenure_4_training_set,dataset_tenure_5_trai
ning_set)


# Recombine into single test set

dataset_tenure_global_test_set <- rbind(dataset_tenure_1_test_set,
dataset_tenure_2_test_set,dataset_tenure_3_test_set,dataset_tenure_4_test_set,dataset_tenure_5_test_set)


# Remove unnecessary data frames

# Unnecessary Tenure data frames

rm(dataset_tenure_1,dataset_tenure_2,dataset_tenure_3,dataset_tenure_4,dataset_tenure_5)

# Unnecessary training set data frames

rm(dataset_tenure_1_training_set,dataset_tenure_2_training_set,dataset_tenure_3_training_set,dataset_tenure_4_t
raining_set,dataset_tenure_5_training_set)

# Unnecessary test set data frames

rm(dataset_tenure_1_test_set,
dataset_tenure_2_test_set,dataset_tenure_3_test_set,dataset_tenure_4_test_set,dataset_tenure_5_test_set)


## Part 5 - Making the Dependent Variable a Factor

dataset_tenure_global_training_set$TENURE <- as.factor(dataset_tenure_global_training_set$TENURE)

dataset_tenure_global_test_set$TENURE <- as.factor(dataset_tenure_global_test_set$TENURE)


## Part 6 - Feature Scaling

dataset_tenure_global_training_set[,2] = scale(dataset_tenure_global_training_set[,2])

dataset_tenure_global_test_set[,2] = scale(dataset_tenure_global_test_set[,2])


## Part 7 - Creating the Random Forest Model

library(randomForest)

set.seed(222)

rf <- randomForest(TENURE~.,data=dataset_tenure_global_training_set)

print(rf)


## Part 8 - Publishing the Results (Model 19)


# Histogram showing No. of nodes for trees (Model 19)

hist(treesize(rf),

     main = "No. of Nodes for the Trees",

     col = "SkyBlue")

# Variable importance (Model 19)

varImpPlot(rf)

importance(rf)
```

```
# Installation of caret
# install.packages('caret', dependencies = TRUE)
# Installation of caret dependencies not identified (lava)
# install.packages('lava')
# library(lava)

# Prediction & Confusion Matrix (Model 19)
library(caret)
p1 <- predict(rf, dataset_tenure_global_training_set)
confusionMatrix(p1, dataset_tenure_global_training_set$TENURE)
p2 <- predict(rf, dataset_tenure_global_test_set)
confusionMatrix(p2, dataset_tenure_global_test_set$TENURE)

## Part 9 - Tuning the Random Forest Model
# Visualise Error Rate of Random Forest
plot(rf)
# Tuning mtry
t <- tuneRF(dataset_tenure_global_training_set[,-1],dataset_tenure_global_training_set[,1],
            stepFactor = 0.5,
            plot = TRUE,
            ntreeTry = 150,
            trace = TRUE,
            improve = 0.05)

## Part 10 - Creating the updated Random Forest Model (Model 20)
library(randomForest)
set.seed(222)
rf<- randomForest(TENURE~.,dataset_tenure_global_training_set,
                  ntree = 150,
                  mtry = 2,
                  importance = TRUE)
print(rf)


## Part 11 - Publishing the Results of the New Model (Model 20)
# Histogram showing No. of nodes for trees (Model 20)
hist(treesize(rf),
     main = "No. of Nodes for the Trees",
     col = "SkyBlue")
# Variable importance (Model 20)
varImpPlot(rf)
importance(rf)
# Prediction & Confusion Matrix (Model 20)
library(caret)
p1 <- predict(rf, dataset_tenure_global_training_set)
confusionMatrix(p1, dataset_tenure_global_training_set$TENURE)
p2 <- predict(rf, dataset_tenure_global_test_set)
confusionMatrix(p2, dataset_tenure_global_test_set$TENURE)
```

TENURE – Model 21 & 22 (80:20 Training:Test Split, with Stratified Sampling)

```
## Part 1 - Importing the dataset
dataset = read.csv('Census_2011_Household_CT_Hgth_Enc.csv')

## Part 2 - Selecting the appropriate variables
dataset_tenure <- subset(dataset,
                         select=c("TENURE","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))

## Part 3 - Generating a Frequency Table for the Dependent Variable
table(dataset_tenure$TENURE)

## PART 4 - stratified sampling

# Extract tenure by class
dataset_tenure_1 <- subset(dataset_tenure, TENURE=="1",
                           select=c("TENURE","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))
dataset_tenure_2 <- subset(dataset_tenure, TENURE=="2",
                           select=c("TENURE","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))
dataset_tenure_3 <- subset(dataset_tenure, TENURE=="3",
                           select=c("TENURE","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))
dataset_tenure_4 <- subset(dataset_tenure, TENURE=="4",
                           select=c("TENURE","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))
dataset_tenure_5 <- subset(dataset_tenure, TENURE=="5",
                           select=c("TENURE","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))

# split tenure into training and test sets
# install.packages('caTools')
library(caTools)
# Tenure 1
set.seed(123)
split = sample.split(dataset_tenure_1$TENURE, SplitRatio = 0.80)
dataset_tenure_1_training_set <- subset(dataset_tenure_1, split == TRUE)
dataset_tenure_1_test_set <- subset(dataset_tenure_1, split == FALSE)
# Tenure 2
set.seed(123)
split = sample.split(dataset_tenure_2$TENURE, SplitRatio = 0.80)
dataset_tenure_2_training_set <- subset(dataset_tenure_2, split == TRUE)
dataset_tenure_2_test_set <- subset(dataset_tenure_2, split == FALSE)
# Tenure 3
set.seed(123)
split = sample.split(dataset_tenure_3$TENURE, SplitRatio = 0.80)
dataset_tenure_3_training_set <- subset(dataset_tenure_3, split == TRUE)
dataset_tenure_3_test_set <- subset(dataset_tenure_3, split == FALSE)
# Tenure 4
set.seed(123)
split = sample.split(dataset_tenure_4$TENURE, SplitRatio = 0.80)
```

```
dataset_tenure_4_training_set <- subset(dataset_tenure_4, split == TRUE)

dataset_tenure_4_test_set <- subset(dataset_tenure_4, split == FALSE)

# Tenure 5

set.seed(123)

split = sample.split(dataset_tenure_5$TENURE, SplitRatio = 0.80)

dataset_tenure_5_training_set <- subset(dataset_tenure_5, split == TRUE)

dataset_tenure_5_test_set <- subset(dataset_tenure_5, split == FALSE)


# Recombine into single training set

dataset_tenure_global_training_set <- rbind(dataset_tenure_1_training_set,
dataset_tenure_2_training_set,dataset_tenure_3_training_set,dataset_tenure_4_training_set,dataset_tenure_5_trai
ning_set)


# Recombine into single test set

dataset_tenure_global_test_set <- rbind(dataset_tenure_1_test_set,
dataset_tenure_2_test_set,dataset_tenure_3_test_set,dataset_tenure_4_test_set,dataset_tenure_5_test_set)


# Remove unnecessary data frames

# Unnecessary Tenure data frames

rm(dataset_tenure_1,dataset_tenure_2,dataset_tenure_3,dataset_tenure_4,dataset_tenure_5)

# Unnecessary training set data frames

rm(dataset_tenure_1_training_set,dataset_tenure_2_training_set,dataset_tenure_3_training_set,dataset_tenure_4_t
raining_set,dataset_tenure_5_training_set)

# Unnecessary test set data frames

rm(dataset_tenure_1_test_set,
dataset_tenure_2_test_set,dataset_tenure_3_test_set,dataset_tenure_4_test_set,dataset_tenure_5_test_set)


## Part 5 - Making the Dependent Variable a Factor

dataset_tenure_global_training_set$TENURE <- as.factor(dataset_tenure_global_training_set$TENURE)

dataset_tenure_global_test_set$TENURE <- as.factor(dataset_tenure_global_test_set$TENURE)


## Part 6 - Feature Scaling

dataset_tenure_global_training_set[,2] = scale(dataset_tenure_global_training_set[,2])

dataset_tenure_global_test_set[,2] = scale(dataset_tenure_global_test_set[,2])


## Part 7 - Creating the Random Forest Model

library(randomForest)

set.seed(222)

rf <- randomForest(TENURE~.,data=dataset_tenure_global_training_set)

print(rf)


## Part 8 - Publishing the Results (Model 21)


# Histogram showing No. of nodes for trees (Model 21)

hist(treesize(rf),

     main = "No. of Nodes for the Trees",

     col = "SkyBlue")

# Variable importance (Model 21)

varImpPlot(rf)

importance(rf)
```

```
# Installation of caret
# install.packages('caret', dependencies = TRUE)
# Installation of caret dependencies not identified (lava)
# install.packages('lava')
# library(lava)

# Prediction & Confusion Matrix (Model 21)
library(caret)
p1 <- predict(rf, dataset_tenure_global_training_set)
confusionMatrix(p1, dataset_tenure_global_training_set$TENURE)
p2 <- predict(rf, dataset_tenure_global_test_set)
confusionMatrix(p2, dataset_tenure_global_test_set$TENURE)

## Part 9 - Tuning the Random Forest Model
# Visualise Error Rate of Random Forest
plot(rf)
# Tuning mtry
t <- tuneRF(dataset_tenure_global_training_set[,-1],dataset_tenure_global_training_set[,1],
            stepFactor = 0.5,
            plot = TRUE,
            ntreeTry = 150,
            trace = TRUE,
            improve = 0.05)

## Part 10 - Creating the updated Random Forest Model (Model 22)
library(randomForest)
set.seed(222)
rf<- randomForest(TENURE~.,dataset_tenure_global_training_set,
                  ntree = 150,
                  mtry = 2,
                  importance = TRUE)
print(rf)

## Part 11 - Publishing the Results of the New Model (Model 22)
# Histogram showing No. of nodes for trees (Model 22)
hist(treesize(rf),
     main = "No. of Nodes for the Trees",
     col = "SkyBlue")
# Variable importance (Model 22)
varImpPlot(rf)
importance(rf)
# Prediction & Confusion Matrix (Model 22)
library(caret)
p1 <- predict(rf, dataset_tenure_global_training_set)
confusionMatrix(p1, dataset_tenure_global_training_set$TENURE)
p2 <- predict(rf, dataset_tenure_global_test_set)
confusionMatrix(p2, dataset_tenure_global_test_set$TENURE)
```

TENURE – Model 23 & 24 (90:10 Training:Test Split, with Stratified Sampling)

```
## Part 1 - Importing the dataset
dataset = read.csv('Census_2011_Household_CT_Hgth_Enc.csv')

## Part 2 - Selecting the appropriate variables
dataset_tenure <- subset(dataset,
                         select=c("TENURE","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))

## Part 3 - Generating a Frequency Table for the Dependent Variable
table(dataset_tenure$TENURE)

## PART 4 - stratified sampling

# Extract tenure by class
dataset_tenure_1 <- subset(dataset_tenure, TENURE=="1",
                           select=c("TENURE","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))
dataset_tenure_2 <- subset(dataset_tenure, TENURE=="2",
                           select=c("TENURE","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))
dataset_tenure_3 <- subset(dataset_tenure, TENURE=="3",
                           select=c("TENURE","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))
dataset_tenure_4 <- subset(dataset_tenure, TENURE=="4",
                           select=c("TENURE","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))
dataset_tenure_5 <- subset(dataset_tenure, TENURE=="5",
                           select=c("TENURE","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))

# split tenure into training and test sets
# install.packages('caTools')
library(caTools)
# Tenure 1
set.seed(123)
split = sample.split(dataset_tenure_1$TENURE, SplitRatio = 0.90)
dataset_tenure_1_training_set <- subset(dataset_tenure_1, split == TRUE)
dataset_tenure_1_test_set <- subset(dataset_tenure_1, split == FALSE)
# Tenure 2
set.seed(123)
split = sample.split(dataset_tenure_2$TENURE, SplitRatio = 0.90)
dataset_tenure_2_training_set <- subset(dataset_tenure_2, split == TRUE)
dataset_tenure_2_test_set <- subset(dataset_tenure_2, split == FALSE)
# Tenure 3
set.seed(123)
split = sample.split(dataset_tenure_3$TENURE, SplitRatio = 0.90)
dataset_tenure_3_training_set <- subset(dataset_tenure_3, split == TRUE)
dataset_tenure_3_test_set <- subset(dataset_tenure_3, split == FALSE)
# Tenure 4
set.seed(123)
split = sample.split(dataset_tenure_4$TENURE, SplitRatio = 0.90)
dataset_tenure_4_training_set <- subset(dataset_tenure_4, split == TRUE)
```

```
dataset_tenure_4_test_set <- subset(dataset_tenure_4, split == FALSE)

# Tenure 5

set.seed(123)

split = sample.split(dataset_tenure_5$TENURE, SplitRatio = 0.90)

dataset_tenure_5_training_set <- subset(dataset_tenure_5, split == TRUE)

dataset_tenure_5_test_set <- subset(dataset_tenure_5, split == FALSE)


# Recombine into single training set

dataset_tenure_global_training_set <- rbind(dataset_tenure_1_training_set,
dataset_tenure_2_training_set,dataset_tenure_3_training_set,dataset_tenure_4_training_set,dataset_tenure_5_trai
ning_set)


# Recombine into single test set

dataset_tenure_global_test_set <- rbind(dataset_tenure_1_test_set,
dataset_tenure_2_test_set,dataset_tenure_3_test_set,dataset_tenure_4_test_set,dataset_tenure_5_test_set)


# Remove unnecessary data frames

# Unnecessary Tenure data frames

rm(dataset_tenure_1,dataset_tenure_2,dataset_tenure_3,dataset_tenure_4,dataset_tenure_5)

# Unnecessary training set data frames

rm(dataset_tenure_1_training_set,dataset_tenure_2_training_set,dataset_tenure_3_training_set,dataset_tenure_4_t
raining_set,dataset_tenure_5_training_set)

# Unnecessary test set data frames

rm(dataset_tenure_1_test_set,
dataset_tenure_2_test_set,dataset_tenure_3_test_set,dataset_tenure_4_test_set,dataset_tenure_5_test_set)


## Part 5 - Making the Dependent Variable a Factor

dataset_tenure_global_training_set$TENURE <- as.factor(dataset_tenure_global_training_set$TENURE)

dataset_tenure_global_test_set$TENURE <- as.factor(dataset_tenure_global_test_set$TENURE)


## Part 6 - Feature Scaling

dataset_tenure_global_training_set[,2] = scale(dataset_tenure_global_training_set[,2])

dataset_tenure_global_test_set[,2] = scale(dataset_tenure_global_test_set[,2])


## Part 7 - Creating the Random Forest Model

library(randomForest)

set.seed(222)

rf <- randomForest(TENURE~.,data=dataset_tenure_global_training_set)

print(rf)


## Part 8 - Publishing the Results (Model 23)


# Histogram showing No. of nodes for trees (Model 23)

hist(treesize(rf),

     main = "No. of Nodes for the Trees",

     col = "SkyBlue")

# Variable importance (Model 23)

varImpPlot(rf)

importance(rf)
```

```
# Installation of caret
# install.packages('caret', dependencies = TRUE)
# Installation of caret dependencies not identified (lava)
# install.packages('lava')
# library(lava)

# Prediction & Confusion Matrix (Model 23)
library(caret)
p1 <- predict(rf, dataset_tenure_global_training_set)
confusionMatrix(p1, dataset_tenure_global_training_set$TENURE)
p2 <- predict(rf, dataset_tenure_global_test_set)
confusionMatrix(p2, dataset_tenure_global_test_set$TENURE)


## Part 9 - Tuning the Random Forest Model
# Visualise Error Rate of Random Forest
plot(rf)
# Tuning mtry
t <- tuneRF(dataset_tenure_global_training_set[,-1],dataset_tenure_global_training_set[,1],
            stepFactor = 0.5,
            plot = TRUE,
            ntreeTry = 150,
            trace = TRUE,
            improve = 0.05)

## Part 10 - Creating the updated Random Forest Model (Model 24)
library(randomForest)
set.seed(222)
rf<- randomForest(TENURE~.,dataset_tenure_global_training_set,
                  ntree = 150,
                  mtry = 2,
                  importance = TRUE)
print(rf)
## Part 11 - Publishing the Results of the New Model (Model 24)
# Histogram showing No. of nodes for trees (Model 24)
hist(treesize(rf),
     main = "No. of Nodes for the Trees",
     col = "SkyBlue")
# Variable importance (Model 24)
varImpPlot(rf)
importance(rf)
# Prediction & Confusion Matrix (Model 24)
library(caret)
p1 <- predict(rf, dataset_tenure_global_training_set)
confusionMatrix(p1, dataset_tenure_global_training_set$TENURE)
p2 <- predict(rf, dataset_tenure_global_test_set)
confusionMatrix(p2, dataset_tenure_global_test_set$TENURE)
```

BEDROOMS – Model 25 & 26 (70:30 Training:Test Split)

```
## Part 1 - Importing the dataset
dataset = read.csv('Census_2011_Household_CT_Hgth_Enc.csv')

## Part 2 - Selecting the appropriate variables
dataset_bedrooms <- subset(dataset, select=c("BEDROOMS","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))

## Part 3 – Generating a Frequency Table for the Dependent Variable
table(dataset_bedrooms$BEDROOMS)

## Part 4 - Splitting the dataset into the Training Set and Test Set
#install.packages('caTools')
library(caTools)
set.seed(123)
split = sample.split(dataset_bedrooms$BEDROOMS, SplitRatio = 0.70)
bedrooms_training_set <- subset(dataset_bedrooms, split == TRUE)
bedrooms_test_set <- subset(dataset_bedrooms, split == FALSE)

## Part 5 – Making the Dependent Variable a Factor
bedrooms_training_set$BEDROOMS <- as.factor(bedrooms_training_set$BEDROOMS)
bedrooms_test_set$BEDROOMS <- as.factor(bedrooms_test_set$BEDROOMS)

## Part 6 - Feature Scaling
bedrooms_training_set[,2] = scale(bedrooms_training_set[,2])
bedrooms_test_set[,2] = scale(bedrooms_test_set[,2])

## Part 7 – Creating the Random Forest Model
library(randomForest)
set.seed(222)
rf <- randomForest(BEDROOMS~.,data=bedrooms_training_set)
print(rf)

## Part 8 – Publishing the Results (Model 25)
# Installation of caret
install.packages('caret', dependencies = TRUE)
# Lava error, install lava
#install.packages('lava')
# Histogram showing No. of nodes for trees (Model 25)
hist(treesize(rf),
     main = "No. of Nodes for the Trees",
     col = "SkyBlue")
# Variable importance (Model 25)
varImpPlot(rf)
importance(rf)
# Prediction & Confusion Matrix (Model 25)
library(caret)
p1 <- predict(rf, bedrooms_training_set)
```

```
confusionMatrix(p1, bedrooms_training_set$BEDROOMS)
p2 <- predict(rf, bedrooms_test_set)
confusionMatrix(p2, bedrooms_test_set$BEDROOMS)

## Part 9 – Tuning the Random Forest Model
# Visualise Error Rate of Random Forest
plot(rf)
# Tuning mtry
t <- tuneRF(bedrooms_training_set[,-1],bedrooms_training_set[,1],
       stepFactor = 0.5,
       plot = TRUE,
       ntreeTry = 150,
       trace = TRUE,
       improve = 0.05)

## Part 10 – Creating the updated Random Forest Model (Model 26)
library(randomForest)
set.seed(222)
rf<- randomForest(BEDROOMS~.,data=bedrooms_training_set,
                  ntree = 150,
                  mtry = 2,
                  importance = TRUE)
print(rf)

## Part 11 – Publishing the Results of the New Model (Model 26)
# Histogram showing No. of nodes for trees (Model 26)
hist(treesize(rf),
     main = "No. of Nodes for the Trees",
     col = "SkyBlue")
# Variable importance (Model 26)
varImpPlot(rf)
importance(rf)
# Prediction & Confusion Matrix (Model 26)
library(caret)
p1 <- predict(rf, bedrooms_training_set)
confusionMatrix(p1, bedrooms_training_set$BEDROOMS)
p2 <- predict(rf, bedrooms_test_set)
confusionMatrix(p2, bedrooms_test_set$BEDROOMS)
```

BEDROOMS – Model 27 & 28 (80:20 Training:Test Split)

```
## Part 1 - Importing the dataset
dataset = read.csv('Census_2011_Household_CT_Hgth_Enc.csv')

## Part 2 - Selecting the appropriate variables
dataset_bedrooms <- subset(dataset, select=c("BEDROOMS","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))

## Part 3 – Generating a Frequency Table for the Dependent Variable
```

```
table(dataset_bedrooms$BEDROOMS)

## Part 4 - Splitting the dataset into the Training Set and Test Set
#install.packages('caTools')
library(caTools)
set.seed(123)
split = sample.split(dataset_bedrooms$BEDROOMS, SplitRatio = 0.80)
bedrooms_training_set <- subset(dataset_bedrooms, split == TRUE)
bedrooms_test_set <- subset(dataset_bedrooms, split == FALSE)

## Part 5 – Making the Dependent Variable a Factor
bedrooms_training_set$BEDROOMS <- as.factor(bedrooms_training_set$BEDROOMS)
bedrooms_test_set$BEDROOMS <- as.factor(bedrooms_test_set$BEDROOMS)

## Part 6 - Feature Scaling
bedrooms_training_set[,2] = scale(bedrooms_training_set[,2])
bedrooms_test_set[,2] = scale(bedrooms_test_set[,2])

## Part 7 – Creating the Random Forest Model
library(randomForest)
set.seed(222)
rf <- randomForest(BEDROOMS~.,data=bedrooms_training_set)
print(rf)

## Part 8 – Publishing the Results (Model 27)
# Installation of caret
install.packages('caret', dependencies = TRUE)
# Lava error, install lava
#install.packages('lava')
# Histogram showing No. of nodes for trees (Model 27)
hist(treesize(rf),
     main = "No. of Nodes for the Trees",
     col = "SkyBlue")
# Variable importance (Model 27)
varImpPlot(rf)
importance(rf)
# Prediction & Confusion Matrix (Model 27)
library(caret)
p1 <- predict(rf, bedrooms_training_set)
confusionMatrix(p1, bedrooms_training_set$BEDROOMS)
p2 <- predict(rf, bedrooms_test_set)
confusionMatrix(p2, bedrooms_test_set$BEDROOMS)

## Part 9 – Tuning the Random Forest Model
# Visualise Error Rate of Random Forest
plot(rf)
# Tuning mtry
t <- tuneRF(bedrooms_training_set[,-1],bedrooms_training_set[,1],
       stepFactor = 0.5,
```

```
       plot = TRUE,
       ntreeTry = 150,
       trace = TRUE,
       improve = 0.05)

## Part 10 – Creating the updated Random Forest Model (Model 28)
library(randomForest)
set.seed(222)
rf<- randomForest(BEDROOMS~.,data=bedrooms_training_set,
                  ntree = 150,
                  mtry = 2,
                  importance = TRUE)
print(rf)

## Part 11 – Publishing the Results of the New Model (Model 28)
# Histogram showing No. of nodes for trees (Model 28)
hist(treesize(rf),
     main = "No. of Nodes for the Trees",
     col = "SkyBlue")
# Variable importance (Model 28)
varImpPlot(rf)
importance(rf)
# Prediction & Confusion Matrix (Model 28)
library(caret)
p1 <- predict(rf, bedrooms_training_set)
confusionMatrix(p1, bedrooms_training_set$BEDROOMS)
p2 <- predict(rf, bedrooms_test_set)
confusionMatrix(p2, bedrooms_test_set$BEDROOMS)
```

BEDROOMS – Model 29 & 30 (90:10 Training:Test Split)

```
## Part 1 - Importing the dataset
dataset = read.csv('Census_2011_Household_CT_Hgth_Enc.csv')

## Part 2 - Selecting the appropriate variables
dataset_bedrooms <- subset(dataset, select=c("BEDROOMS","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))

## Part 3 – Generating a Frequency Table for the Dependent Variable
table(dataset_bedrooms$BEDROOMS)

## Part 4 - Splitting the dataset into the Training Set and Test Set
#install.packages('caTools')
library(caTools)
set.seed(123)
split = sample.split(dataset_bedrooms$BEDROOMS, SplitRatio = 0.90)
bedrooms_training_set <- subset(dataset_bedrooms, split == TRUE)
bedrooms_test_set <- subset(dataset_bedrooms, split == FALSE)
```

```
## Part 5 – Making the Dependent Variable a Factor
bedrooms_training_set$BEDROOMS <- as.factor(bedrooms_training_set$BEDROOMS)
bedrooms_test_set$BEDROOMS <- as.factor(bedrooms_test_set$BEDROOMS)

## Part 6 - Feature Scaling
bedrooms_training_set[,2] = scale(bedrooms_training_set[,2])
bedrooms_test_set[,2] = scale(bedrooms_test_set[,2])

## Part 7 – Creating the Random Forest Model
library(randomForest)
set.seed(222)
rf <- randomForest(BEDROOMS~.,data=bedrooms_training_set)
print(rf)

## Part 8 – Publishing the Results (Model 29)
# Installation of caret
install.packages('caret', dependencies = TRUE)
# Lava error, install lava
#install.packages('lava')
# Histogram showing No. of nodes for trees (Model 29)
hist(treesize(rf),
     main = "No. of Nodes for the Trees",
     col = "SkyBlue")
# Variable importance (Model 29)
varImpPlot(rf)
importance(rf)
# Prediction & Confusion Matrix (Model 29)
library(caret)
p1 <- predict(rf, bedrooms_training_set)
confusionMatrix(p1, bedrooms_training_set$BEDROOMS)
p2 <- predict(rf, bedrooms_test_set)
confusionMatrix(p2, bedrooms_test_set$BEDROOMS)

## Part 9 – Tuning the Random Forest Model
# Visualise Error Rate of Random Forest
plot(rf)
# Tuning mtry
t <- tuneRF(bedrooms_training_set[,-1],bedrooms_training_set[,1],
       stepFactor = 0.5,
       plot = TRUE,
       ntreeTry = 150,
       trace = TRUE,
       improve = 0.05)

## Part 10 – Creating the updated Random Forest Model (Model 30)
library(randomForest)
set.seed(222)
rf<- randomForest(BEDROOMS~.,data=bedrooms_training_set,
                  ntree = 150,
```

```
                    mtry = 2,
                    importance = TRUE)
print(rf)

## Part 11 – Publishing the Results of the New Model (Model 30)
# Histogram showing No. of nodes for trees (Model 30)
hist(treesize(rf),
     main = "No. of Nodes for the Trees",
     col = "SkyBlue")
# Variable importance (Model 30)
varImpPlot(rf)
importance(rf)
# Prediction & Confusion Matrix (Model 30)
library(caret)
p1 <- predict(rf, bedrooms_training_set)
confusionMatrix(p1, bedrooms_training_set$BEDROOMS)
p2 <- predict(rf, bedrooms_test_set)
confusionMatrix(p2, bedrooms_test_set$BEDROOMS)
```

BEDROOMS – Model 31 & 32 (70:30 Training:Test Split, with Stratified Sampling)

```
## Part 1 - Importing the dataset
dataset = read.csv('Census_2011_Household_CT_Hgth_Enc.csv')

## Part 2 - Selecting the appropriate variables
dataset_bedrooms <- subset(dataset,
                        select=c("BEDROOMS","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))

## Part 3 - Generating a Frequency Table for the Dependent Variable
table(dataset_bedrooms$BEDROOMS)


## PART 4 - stratified sampling

# Extract bedrooms by class
dataset_bedrooms_0 <- subset(dataset_bedrooms, BEDROOMS=="0",
                           select=c("BEDROOMS","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))
dataset_bedrooms_1 <- subset(dataset_bedrooms, BEDROOMS=="1",
                           select=c("BEDROOMS","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))
dataset_bedrooms_2 <- subset(dataset_bedrooms, BEDROOMS=="2",
                           select=c("BEDROOMS","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))
dataset_bedrooms_3 <- subset(dataset_bedrooms, BEDROOMS=="3",
                           select=c("BEDROOMS","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))
dataset_bedrooms_4 <- subset(dataset_bedrooms, BEDROOMS=="4",
                           select=c("BEDROOMS","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))
dataset_bedrooms_5 <- subset(dataset_bedrooms, BEDROOMS=="5",
                           select=c("BEDROOMS","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))
```

```
# split bedrooms into training and test sets
# install.packages('caTools')
library(caTools)


# Bedrooms 0
set.seed(123)
split = sample.split(dataset_bedrooms_0$BEDROOMS, SplitRatio = 0.70)
dataset_bedrooms_0_training_set <- subset(dataset_bedrooms_0, split == TRUE)
dataset_bedrooms_0_test_set <- subset(dataset_bedrooms_0, split == FALSE)
# Bedrooms 1
set.seed(123)
split = sample.split(dataset_bedrooms_1$BEDROOMS, SplitRatio = 0.70)
dataset_bedrooms_1_training_set <- subset(dataset_bedrooms_1, split == TRUE)
dataset_bedrooms_1_test_set <- subset(dataset_bedrooms_1, split == FALSE)
# Bedrooms 2
set.seed(123)
split = sample.split(dataset_bedrooms_2$BEDROOMS, SplitRatio = 0.70)
dataset_bedrooms_2_training_set <- subset(dataset_bedrooms_2, split == TRUE)
dataset_bedrooms_2_test_set <- subset(dataset_bedrooms_2, split == FALSE)
# Bedrooms 3
set.seed(123)
split = sample.split(dataset_bedrooms_3$BEDROOMS, SplitRatio = 0.70)
dataset_bedrooms_3_training_set <- subset(dataset_bedrooms_3, split == TRUE)
dataset_bedrooms_3_test_set <- subset(dataset_bedrooms_3, split == FALSE)
# Bedrooms 4
set.seed(123)
split = sample.split(dataset_bedrooms_4$BEDROOMS, SplitRatio = 0.70)
dataset_bedrooms_4_training_set <- subset(dataset_bedrooms_4, split == TRUE)
dataset_bedrooms_4_test_set <- subset(dataset_bedrooms_4, split == FALSE)
# Bedrooms 5
set.seed(123)
split = sample.split(dataset_bedrooms_5$BEDROOMS, SplitRatio = 0.70)
dataset_bedrooms_5_training_set <- subset(dataset_bedrooms_5, split == TRUE)
dataset_bedrooms_5_test_set <- subset(dataset_bedrooms_5, split == FALSE)
# Recombine into single training set
dataset_bedrooms_global_training_set <- rbind(dataset_bedrooms_0_training_set,dataset_bedrooms_1_training_set,
dataset_bedrooms_2_training_set,dataset_bedrooms_3_training_set,dataset_bedrooms_4_training_set,dataset_bedroom
s_5_training_set)


# Recombine into single test set
dataset_bedrooms_global_test_set <- rbind(dataset_bedrooms_0_test_set,dataset_bedrooms_1_test_set,
dataset_bedrooms_2_test_set,dataset_bedrooms_3_test_set,dataset_bedrooms_4_test_set,dataset_bedrooms_5_test_set
)


# Remove unnecessary data frames
# Unnecessary Bedrooms data frames
rm(dataset_bedrooms_0,dataset_bedrooms_1,dataset_bedrooms_2,dataset_bedrooms_3,dataset_bedrooms_4,dataset_bedro
oms_5)
# Unnecessary training set data frames
rm(dataset_bedrooms_0_training_set,dataset_bedrooms_1_training_set,dataset_bedrooms_2_training_set,dataset_bedr
ooms_3_training_set,dataset_bedrooms_4_training_set,dataset_bedrooms_5_training_set)
```

```
# Unnecessary test set data frames
rm(dataset_bedrooms_0_test_set,dataset_bedrooms_1_test_set,dataset_bedrooms_2_test_set,dataset_bedrooms_3_test_
set,dataset_bedrooms_4_test_set,dataset_bedrooms_5_test_set)


## Part 5 - Making the Dependent Variable a Factor
dataset_bedrooms_global_training_set$BEDROOMS <- as.factor(dataset_bedrooms_global_training_set$BEDROOMS)
dataset_bedrooms_global_test_set$BEDROOMS <- as.factor(dataset_bedrooms_global_test_set$BEDROOMS)


## Part 6 - Feature Scaling
dataset_bedrooms_global_training_set[,2] = scale(dataset_bedrooms_global_training_set[,2])
dataset_bedrooms_global_test_set[,2] = scale(dataset_bedrooms_global_test_set[,2])


## Part 7 - Creating the Random Forest Model
library(randomForest)
set.seed(222)
rf <- randomForest(BEDROOMS~.,data=dataset_bedrooms_global_training_set)
print(rf)


## Part 8 - Publishing the Results (Model 31)


# Histogram showing No. of nodes for trees (Model 31)
hist(treesize(rf),
     main = "No. of Nodes for the Trees",
     col = "SkyBlue")
# Variable importance (Model 31)
varImpPlot(rf)
importance(rf)


# Installation of caret
# install.packages('caret', dependencies = TRUE)
# Installation of caret dependencies not identified (lava)
# install.packages('lava')
# library(lava)


# Prediction & Confusion Matrix (Model 31)
library(caret)
p1 <- predict(rf, dataset_bedrooms_global_training_set)
confusionMatrix(p1, dataset_bedrooms_global_training_set$BEDROOMS)
p2 <- predict(rf, dataset_bedrooms_global_test_set)
confusionMatrix(p2, dataset_bedrooms_global_test_set$BEDROOMS)


## Part 9 - Tuning the Random Forest Model
# Visualise Error Rate of Random Forest
plot(rf)
# Tuning mtry
t <- tuneRF(dataset_bedrooms_global_training_set[,-1],dataset_bedrooms_global_training_set[,1],
            stepFactor = 0.5,
            plot = TRUE,
            ntreeTry = 150,
```

```
            trace = TRUE,
            improve = 0.05)


## Part 10 - Creating the updated Random Forest Model (Model 32)
library(randomForest)
set.seed(222)
rf<- randomForest(BEDROOMS~.,dataset_bedrooms_global_training_set,
                  ntree = 150,
                  mtry = 2,
                  importance = TRUE)
print(rf)

## Part 11 - Publishing the Results of the New Model (Model 32)
# Histogram showing No. of nodes for trees (Model 32)
hist(treesize(rf),
     main = "No. of Nodes for the Trees",
     col = "SkyBlue")
# Variable importance (Model 32)
varImpPlot(rf)
importance(rf)
# Prediction & Confusion Matrix (Model 32)
library(caret)
p1 <- predict(rf, dataset_bedrooms_global_training_set)
confusionMatrix(p1, dataset_bedrooms_global_training_set$BEDROOMS)
p2 <- predict(rf, dataset_bedrooms_global_test_set)
confusionMatrix(p2, dataset_bedrooms_global_test_set$BEDROOMS)
```

BEDROOMS – Model 33 & 34 (80:20 Training:Test Split, with Stratified Sampling)

```
## Part 1 - Importing the dataset
dataset = read.csv('Census_2011_Household_CT_Hgth_Enc.csv')

## Part 2 - Selecting the appropriate variables
dataset_bedrooms <- subset(dataset,
                           select=c("BEDROOMS","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))

## Part 3 - Generating a Frequency Table for the Dependent Variable
table(dataset_bedrooms$BEDROOMS)

## PART 4 - stratified sampling

# Extract bedrooms by class
dataset_bedrooms_0 <- subset(dataset_bedrooms, BEDROOMS=="0",
                             select=c("BEDROOMS","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))
dataset_bedrooms_1 <- subset(dataset_bedrooms, BEDROOMS=="1",
                             select=c("BEDROOMS","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))
dataset_bedrooms_2 <- subset(dataset_bedrooms, BEDROOMS=="2",
```

```
                          select=c("BEDROOMS","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))
dataset_bedrooms_3 <- subset(dataset_bedrooms, BEDROOMS=="3",
                          select=c("BEDROOMS","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))
dataset_bedrooms_4 <- subset(dataset_bedrooms, BEDROOMS=="4",
                          select=c("BEDROOMS","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))
dataset_bedrooms_5 <- subset(dataset_bedrooms, BEDROOMS=="5",
                          select=c("BEDROOMS","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))


# split bedrooms into training and test sets
# install.packages('caTools')
library(caTools)
# Bedrooms 0
set.seed(123)
split = sample.split(dataset_bedrooms_0$BEDROOMS, SplitRatio = 0.80)
dataset_bedrooms_0_training_set <- subset(dataset_bedrooms_0, split == TRUE)
dataset_bedrooms_0_test_set <- subset(dataset_bedrooms_0, split == FALSE)
# Bedrooms 1
set.seed(123)
split = sample.split(dataset_bedrooms_1$BEDROOMS, SplitRatio = 0.80)
dataset_bedrooms_1_training_set <- subset(dataset_bedrooms_1, split == TRUE)
dataset_bedrooms_1_test_set <- subset(dataset_bedrooms_1, split == FALSE)
# Bedrooms 2
set.seed(123)
split = sample.split(dataset_bedrooms_2$BEDROOMS, SplitRatio = 0.80)
dataset_bedrooms_2_training_set <- subset(dataset_bedrooms_2, split == TRUE)
dataset_bedrooms_2_test_set <- subset(dataset_bedrooms_2, split == FALSE)
# Bedrooms 3
set.seed(123)
split = sample.split(dataset_bedrooms_3$BEDROOMS, SplitRatio = 0.80)
dataset_bedrooms_3_training_set <- subset(dataset_bedrooms_3, split == TRUE)
dataset_bedrooms_3_test_set <- subset(dataset_bedrooms_3, split == FALSE)
# Bedrooms 4
set.seed(123)
split = sample.split(dataset_bedrooms_4$BEDROOMS, SplitRatio = 0.80)
dataset_bedrooms_4_training_set <- subset(dataset_bedrooms_4, split == TRUE)
dataset_bedrooms_4_test_set <- subset(dataset_bedrooms_4, split == FALSE)
# Bedrooms 5
set.seed(123)
split = sample.split(dataset_bedrooms_5$BEDROOMS, SplitRatio = 0.80)
dataset_bedrooms_5_training_set <- subset(dataset_bedrooms_5, split == TRUE)
dataset_bedrooms_5_test_set <- subset(dataset_bedrooms_5, split == FALSE)
# Recombine into single training set
dataset_bedrooms_global_training_set <- rbind(dataset_bedrooms_0_training_set,dataset_bedrooms_1_training_set,
dataset_bedrooms_2_training_set,dataset_bedrooms_3_training_set,dataset_bedrooms_4_training_set,dataset_bedroom
s_5_training_set)


# Recombine into single test set
dataset_bedrooms_global_test_set <- rbind(dataset_bedrooms_0_test_set,dataset_bedrooms_1_test_set,
dataset_bedrooms_2_test_set,dataset_bedrooms_3_test_set,dataset_bedrooms_4_test_set,dataset_bedrooms_5_test_set
)
```

```
# Remove unnecessary data frames
# Unnecessary Bedrooms data frames
rm(dataset_bedrooms_0,dataset_bedrooms_1,dataset_bedrooms_2,dataset_bedrooms_3,dataset_bedrooms_4,dataset_bedrooms_5)
# Unnecessary training set data frames
rm(dataset_bedrooms_0_training_set,dataset_bedrooms_1_training_set,dataset_bedrooms_2_training_set,dataset_bedrooms_3_training_set,dataset_bedrooms_4_training_set,dataset_bedrooms_5_training_set)
# Unnecessary test set data frames
rm(dataset_bedrooms_0_test_set,dataset_bedrooms_1_test_set,dataset_bedrooms_2_test_set,dataset_bedrooms_3_test_set,dataset_bedrooms_4_test_set,dataset_bedrooms_5_test_set)

## Part 5 - Making the Dependent Variable a Factor
dataset_bedrooms_global_training_set$BEDROOMS <- as.factor(dataset_bedrooms_global_training_set$BEDROOMS)
dataset_bedrooms_global_test_set$BEDROOMS <- as.factor(dataset_bedrooms_global_test_set$BEDROOMS)

## Part 6 - Feature Scaling
dataset_bedrooms_global_training_set[,2] = scale(dataset_bedrooms_global_training_set[,2])
dataset_bedrooms_global_test_set[,2] = scale(dataset_bedrooms_global_test_set[,2])

## Part 7 - Creating the Random Forest Model
library(randomForest)
set.seed(222)
rf <- randomForest(BEDROOMS~.,data=dataset_bedrooms_global_training_set)
print(rf)

## Part 8 - Publishing the Results (Model 33)

# Histogram showing No. of nodes for trees (Model 33)
hist(treesize(rf),
     main = "No. of Nodes for the Trees",
     col = "SkyBlue")
# Variable importance (Model 33)
varImpPlot(rf)
importance(rf)

# Installation of caret
# install.packages('caret', dependencies = TRUE)
# Installation of caret dependencies not identified (lava)
# install.packages('lava')
# library(lava)

# Prediction & Confusion Matrix (Model 33)
library(caret)
p1 <- predict(rf, dataset_bedrooms_global_training_set)
confusionMatrix(p1, dataset_bedrooms_global_training_set$BEDROOMS)
p2 <- predict(rf, dataset_bedrooms_global_test_set)
confusionMatrix(p2, dataset_bedrooms_global_test_set$BEDROOMS)

## Part 9 - Tuning the Random Forest Model
```

```
# Visualise Error Rate of Random Forest
plot(rf)



# Tuning mtry
t <- tuneRF(dataset_bedrooms_global_training_set[,-1],dataset_bedrooms_global_training_set[,1],
            stepFactor = 0.5,
            plot = TRUE,
            ntreeTry = 150,
            trace = TRUE,
            improve = 0.05)

## Part 10 - Creating the updated Random Forest Model (Model 34)
library(randomForest)
set.seed(222)
rf<- randomForest(BEDROOMS~.,dataset_bedrooms_global_training_set,
                  ntree = 150,
                  mtry = 2,
                  importance = TRUE)
print(rf)

## Part 11 - Publishing the Results of the New Model (Model 34)
# Histogram showing No. of nodes for trees (Model 34)
hist(treesize(rf),
     main = "No. of Nodes for the Trees",
     col = "SkyBlue")
# Variable importance (Model 34)
varImpPlot(rf)
importance(rf)
# Prediction & Confusion Matrix (Model 34)
library(caret)
p1 <- predict(rf, dataset_bedrooms_global_training_set)
confusionMatrix(p1, dataset_bedrooms_global_training_set$BEDROOMS)
p2 <- predict(rf, dataset_bedrooms_global_test_set)
confusionMatrix(p2, dataset_bedrooms_global_test_set$BEDROOMS)
```

BEDROOMS – Model 35 & 36 (90:10 Training:Test Split, with Stratified Sampling)

```
## Part 1 - Importing the dataset
dataset = read.csv('Census_2011_Household_CT_Hgth_Enc.csv')

## Part 2 - Selecting the appropriate variables
dataset_bedrooms <- subset(dataset,
                           select=c("BEDROOMS","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))

## Part 3 - Generating a Frequency Table for the Dependent Variable
table(dataset_bedrooms$BEDROOMS)
```

```
## PART 4 - stratified sampling
# Extract bedrooms by class
dataset_bedrooms_0 <- subset(dataset_bedrooms, BEDROOMS=="0",
                          select=c("BEDROOMS","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))
dataset_bedrooms_1 <- subset(dataset_bedrooms, BEDROOMS=="1",
                          select=c("BEDROOMS","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))
dataset_bedrooms_2 <- subset(dataset_bedrooms, BEDROOMS=="2",
                          select=c("BEDROOMS","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))
dataset_bedrooms_3 <- subset(dataset_bedrooms, BEDROOMS=="3",
                          select=c("BEDROOMS","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))
dataset_bedrooms_4 <- subset(dataset_bedrooms, BEDROOMS=="4",
                          select=c("BEDROOMS","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))
dataset_bedrooms_5 <- subset(dataset_bedrooms, BEDROOMS=="5",
                          select=c("BEDROOMS","HHAGE","HHPOP","INCOME","HHEMPLOY","HSIZE","HHSEX"))

# split bedrooms into training and test sets
# install.packages('caTools')
library(caTools)
# Bedrooms 0
set.seed(123)
split = sample.split(dataset_bedrooms_0$BEDROOMS, SplitRatio = 0.90)
dataset_bedrooms_0_training_set <- subset(dataset_bedrooms_0, split == TRUE)
dataset_bedrooms_0_test_set <- subset(dataset_bedrooms_0, split == FALSE)
# Bedrooms 1
set.seed(123)
split = sample.split(dataset_bedrooms_1$BEDROOMS, SplitRatio = 0.90)
dataset_bedrooms_1_training_set <- subset(dataset_bedrooms_1, split == TRUE)
dataset_bedrooms_1_test_set <- subset(dataset_bedrooms_1, split == FALSE)
# Bedrooms 2
set.seed(123)
split = sample.split(dataset_bedrooms_2$BEDROOMS, SplitRatio = 0.90)
dataset_bedrooms_2_training_set <- subset(dataset_bedrooms_2, split == TRUE)
dataset_bedrooms_2_test_set <- subset(dataset_bedrooms_2, split == FALSE)
# Bedrooms 3
set.seed(123)
split = sample.split(dataset_bedrooms_3$BEDROOMS, SplitRatio = 0.90)
dataset_bedrooms_3_training_set <- subset(dataset_bedrooms_3, split == TRUE)
dataset_bedrooms_3_test_set <- subset(dataset_bedrooms_3, split == FALSE)
# Bedrooms 4
set.seed(123)
split = sample.split(dataset_bedrooms_4$BEDROOMS, SplitRatio = 0.90)
dataset_bedrooms_4_training_set <- subset(dataset_bedrooms_4, split == TRUE)
dataset_bedrooms_4_test_set <- subset(dataset_bedrooms_4, split == FALSE)
# Bedrooms 5
set.seed(123)
split = sample.split(dataset_bedrooms_5$BEDROOMS, SplitRatio = 0.90)
dataset_bedrooms_5_training_set <- subset(dataset_bedrooms_5, split == TRUE)
dataset_bedrooms_5_test_set <- subset(dataset_bedrooms_5, split == FALSE)
# Recombine into single training set
```

```
dataset_bedrooms_global_training_set <- rbind(dataset_bedrooms_0_training_set,dataset_bedrooms_1_training_set,
dataset_bedrooms_2_training_set,dataset_bedrooms_3_training_set,dataset_bedrooms_4_training_set,dataset_bedroom
s_5_training_set)


# Recombine into single test set

dataset_bedrooms_global_test_set <- rbind(dataset_bedrooms_0_test_set,dataset_bedrooms_1_test_set,
dataset_bedrooms_2_test_set,dataset_bedrooms_3_test_set,dataset_bedrooms_4_test_set,dataset_bedrooms_5_test_set
)


# Remove unnecessary data frames

# Unnecessary Bedrooms data frames

rm(dataset_bedrooms_0,dataset_bedrooms_1,dataset_bedrooms_2,dataset_bedrooms_3,dataset_bedrooms_4,dataset_bedro
oms_5)

# Unnecessary training set data frames

rm(dataset_bedrooms_0_training_set,dataset_bedrooms_1_training_set,dataset_bedrooms_2_training_set,dataset_bedr
ooms_3_training_set,dataset_bedrooms_4_training_set,dataset_bedrooms_5_training_set)

# Unnecessary test set data frames

rm(dataset_bedrooms_0_test_set,dataset_bedrooms_1_test_set,dataset_bedrooms_2_test_set,dataset_bedrooms_3_test_
set,dataset_bedrooms_4_test_set,dataset_bedrooms_5_test_set)


## Part 5 - Making the Dependent Variable a Factor

dataset_bedrooms_global_training_set$BEDROOMS <- as.factor(dataset_bedrooms_global_training_set$BEDROOMS)

dataset_bedrooms_global_test_set$BEDROOMS <- as.factor(dataset_bedrooms_global_test_set$BEDROOMS)


## Part 6 - Feature Scaling

dataset_bedrooms_global_training_set[,2] = scale(dataset_bedrooms_global_training_set[,2])

dataset_bedrooms_global_test_set[,2] = scale(dataset_bedrooms_global_test_set[,2])


## Part 7 - Creating the Random Forest Model

library(randomForest)

set.seed(222)

rf <- randomForest(BEDROOMS~.,data=dataset_bedrooms_global_training_set)

print(rf)


## Part 8 - Publishing the Results (Model 35)


# Histogram showing No. of nodes for trees (Model 35)

hist(treesize(rf),

     main = "No. of Nodes for the Trees",

     col = "SkyBlue")

# Variable importance (Model 35)

varImpPlot(rf)

importance(rf)


# Installation of caret

# install.packages('caret', dependencies = TRUE)

# Installation of caret dependencies not identified (lava)

# install.packages('lava')

# library(lava)


# Prediction & Confusion Matrix (Model 35)
```

```
library(caret)
p1 <- predict(rf, dataset_bedrooms_global_training_set)
confusionMatrix(p1, dataset_bedrooms_global_training_set$BEDROOMS)
p2 <- predict(rf, dataset_bedrooms_global_test_set)
confusionMatrix(p2, dataset_bedrooms_global_test_set$BEDROOMS)

## Part 9 - Tuning the Random Forest Model
# Visualise Error Rate of Random Forest
plot(rf)
# Tuning mtry
t <- tuneRF(dataset_bedrooms_global_training_set[,-1],dataset_bedrooms_global_training_set[,1],
            stepFactor = 0.5,
            plot = TRUE,
            ntreeTry = 150,
            trace = TRUE,
            improve = 0.05)

## Part 10 - Creating the updated Random Forest Model (Model 36)
library(randomForest)
set.seed(222)
rf<- randomForest(BEDROOMS~.,dataset_bedrooms_global_training_set,
                  ntree = 150,
                  mtry = 2,
                  importance = TRUE)
print(rf)

## Part 11 - Publishing the Results of the New Model (Model 36)
# Histogram showing No. of nodes for trees (Model 36)
hist(treesize(rf),
     main = "No. of Nodes for the Trees",
     col = "SkyBlue")
# Variable importance (Model 36)
varImpPlot(rf)
importance(rf)
# Prediction & Confusion Matrix (Model 36)
library(caret)
p1 <- predict(rf, dataset_bedrooms_global_training_set)
confusionMatrix(p1, dataset_bedrooms_global_training_set$BEDROOMS)
p2 <- predict(rf, dataset_bedrooms_global_test_set)
confusionMatrix(p2, dataset_bedrooms_global_test_set$BEDROOMS)
```

Annexure D

randomForest Function in R

Classification and Regression with Random Forest

Description

`randomForest` implements Breiman's random forest algorithm (based on Breiman and Cutler's original Fortran code) for classification and regression. It can also be used in unsupervised mode for assessing proximities among data points.

Usage

```
## S3 method for class 'formula'
randomForest(formula, data=NULL, ..., subset, na.action=na.fail)
## Default S3 method:
randomForest(x, y=NULL,  xtest=NULL, ytest=NULL, ntree=500,
             mtry=if (!is.null(y) && !is.factor(y))
             max(floor(ncol(x)/3), 1) else floor(sqrt(ncol(x))),
             replace=TRUE, classwt=NULL, cutoff, strata,
             sampsize = if (replace) nrow(x) else ceiling(.632*nrow(x)),
             nodesize = if (!is.null(y) && !is.factor(y)) 5 else 1,
             maxnodes = NULL,
             importance=FALSE, localImp=FALSE, nPerm=1,
             proximity, oob.prox=proximity,
             norm.votes=TRUE, do.trace=FALSE,
             keep.forest=!is.null(y) && is.null(xtest), corr.bias=FALSE,
             keep.inbag=FALSE, ...)
## S3 method for class 'randomForest'
print(x, ...)
```

Arguments

`data`	an optional data frame containing the variables in the model. By default the variables are taken from the environment which `randomForest` is called from.
`subset`	an index vector indicating which rows should be used. (NOTE: If given, this argument must be named.)
`na.action`	A function to specify the action to be taken if NAs are found. (NOTE: If given, this argument must be named.)
`x, formula`	a data frame or a matrix of predictors, or a formula describing the model to be fitted (for the `print` method, an `randomForest` object).
`y`	A response vector. If a factor, classification is assumed, otherwise regression is assumed. If omitted,`randomForest` will run in unsupervised mode.
`xtest`	a data frame or matrix (like `x`) containing predictors for the test set.
`ytest`	response for the test set.
`ntree`	Number of trees to grow. This should not be set to too small a number, to ensure that every input row gets predicted at least a few times.
`mtry`	Number of variables randomly sampled as candidates at each split. Note that the default values are different for classification (sqrt(p) where p is number of variables in `x`) and regression (p/3)

replace	Should sampling of cases be done with or without replacement?
classwt	Priors of the classes. Need not add up to one. Ignored for regression.
cutoff	(Classification only) A vector of length equal to number of classes. The 'winning' class for an observation is the one with the maximum ratio of proportion of votes to cutoff. Default is 1/k where k is the number of classes (i.e., majority vote wins).
strata	A (factor) variable that is used for stratified sampling.
sampsize	Size(s) of sample to draw. For classification, if sampsize is a vector of the length the number of strata, then sampling is stratified by strata, and the elements of sampsize indicate the numbers to be drawn from the strata.
nodesize	Minimum size of terminal nodes. Setting this number larger causes smaller trees to be grown (and thus take less time). Note that the default values are different for classification (1) and regression (5).
maxnodes	Maximum number of terminal nodes trees in the forest can have. If not given, trees are grown to the maximum possible (subject to limits by `nodesize`). If set larger than maximum possible, a warning is issued.
importance	Should importance of predictors be assessed?
localImp	Should casewise importance measure be computed? (Setting this to `TRUE` will override `importance`.)
nPerm	Number of times the OOB data are permuted per tree for assessing variable importance. Number larger than 1 gives slightly more stable estimate, but not very effective. Currently only implemented for regression.
proximity	Should proximity measure among the rows be calculated?
oob.prox	Should proximity be calculated only on "out-of-bag" data?
norm.votes	If `TRUE` (default), the final result of votes are expressed as fractions. If `FALSE`, raw vote counts are returned (useful for combining results from different runs). Ignored for regression.
do.trace	If set to `TRUE`, give a more verbose output as`randomForest` is run. If set to some integer, then running output is printed for every `do.trace` trees.
keep.forest	If set to `FALSE`, the forest will not be retained in the output object. If `xtest` is given, defaults to `FALSE`.
corr.bias	perform bias correction for regression? Note: Experimental. Use at your own risk.
keep.inbag	Should an `n` by `ntree` matrix be returned that keeps track of which samples are "in-bag" in which trees (but not how many times, if sampling with replacement)
...	optional parameters to be passed to the low level function`randomForest.default`.

Value

An object of class `randomForest`, which is a list with the following components:

call	the original call to `randomForest`
type	one of `regression`, `classification`, or`unsupervised`.
predicted	the predicted values of the input data based on out-of-bag samples.
importance	a matrix with `nclass` + 2 (for classification) or two (for regression) columns. For

	classification, the first `nclass`columns are the class-specific measures computed as mean descrease in accuracy. The `nclass` + 1st column is the mean descrease in accuracy over all classes. The last column is the mean decrease in Gini index. For Regression, the first column is the mean decrease in accuracy and the second the mean decrease in MSE. If`importance=FALSE`, the last measure is still returned as a vector.
`importanceSD`	The “standard errors” of the permutation-based importance measure. For classification, a `p` by `nclass + 1` matrix corresponding to the first `nclass + 1` columns of the importance matrix. For regression, a length `p` vector.
`localImp`	a p by n matrix containing the casewise importance measures, the [i,j] element of which is the importance of i-th variable on the j-th case. `NULL` if `localImp=FALSE`.
`ntree`	number of trees grown.
`mtry`	number of predictors sampled for spliting at each node.
`forest`	(a list that contains the entire forest; `NULL` if`randomForest` is run in unsupervised mode or if`keep.forest=FALSE`.
`err.rate`	(classification only) vector error rates of the prediction on the input data, the i-th element being the (OOB) error rate for all trees up to the i-th.
`confusion`	(classification only) the confusion matrix of the prediction (based on OOB data).
`votes`	(classification only) a matrix with one row for each input data point and one column for each class, giving the fraction or number of (OOB) ‘votes’ from the random forest.
`oob.times`	number of times cases are ‘out-of-bag’ (and thus used in computing OOB error estimate)
`proximity`	if `proximity=TRUE` when `randomForest` is called, a matrix of proximity measures among the input (based on the frequency that pairs of data points are in the same terminal nodes).
`mse`	(regression only) vector of mean square errors: sum of squared residuals divided by `n`.
`rsq`	(regression only) “pseudo R-squared”: 1 - `mse` / Var(y).
`test`	if test set is given (through the `xtest` or additionally`ytest` arguments), this component is a list which contains the corresponding `predicted`, `err.rate`,`confusion`, `votes` (for classification) or `predicted`,`mse` and `rsq` (for regression) for the test set. If`proximity=TRUE`, there is also a component,`proximity`, which contains the proximity among the test set as well as proximity between test and training data.

Note

The `forest` structure is slightly different between classification and regression. For details on how the trees are stored, see the help page for`getTree`.

If `xtest` is given, prediction of the test set is done “in place” as the trees are grown. If `ytest` is also given, and `do.trace` is set to some positive integer, then for every `do.trace` trees, the test set error is printed. Results for the test set is returned in the `test` component of the resulting `randomForest`object. For classification, the `votes` component (for training or test set data) contain the votes the cases received for the classes. If `norm.votes=TRUE`, the fraction is given, which can be taken as predicted probabilities for the classes.

For large data sets, especially those with large number of variables, calling`randomForest` via the formula interface is not advised: There may be too much overhead in handling the formula.

The “local” (or casewise) variable importance is computed as follows: For classification, it is the increase in percent of times a case is OOB and misclassified when the variable is permuted. For regression, it is the average increase in squared OOB residuals when the variable is permuted.

Author(s)

Andy Liaw andy_liaw@merck.com and Matthew Wienermatthew_wiener@merck.com, based on original Fortran code by Leo Breiman and Adele Cutler.

References

Breiman, L. (2001), *Random Forests*, Machine Learning 45(1), 5-32.

Breiman, L (2002), “Manual On Setting Up, Using, And Understanding Random Forests V3.1”,https://www.stat.berkeley.edu/~breiman/Using_random_forests_V3.1.pdf.

See Also

`predict.randomForest`, `varImpPlot`

Examples

```
## Classification:
##data(iris)
set.seed(71)
iris.rf <- randomForest(Species ~ ., data=iris, importance=TRUE,
                        proximity=TRUE)
print(iris.rf)
## Look at variable importance:
round(importance(iris.rf), 2)
## Do MDS on 1 - proximity:
iris.mds <- cmdscale(1 - iris.rf$proximity, eig=TRUE)
op <- par(pty="s")
pairs(cbind(iris[,1:4], iris.mds$points), cex=0.6, gap=0,
      col=c("red", "green", "blue")[as.numeric(iris$Species)],
      main="Iris Data: Predictors and MDS of Proximity Based on
RandomForest")
par(op)
print(iris.mds$GOF)

## The `unsupervised' case:
set.seed(17)
iris.urf <- randomForest(iris[, -5])
MDSplot(iris.urf, iris$Species)

## stratified sampling: draw 20, 30, and 20 of the species to grow each
tree.
(iris.rf2 <- randomForest(iris[1:4], iris$Species,
                          sampsize=c(20, 30, 20)))

## Regression:
## data(airquality)
set.seed(131)
ozone.rf <- randomForest(Ozone ~ ., data=airquality, mtry=3,
```

```
                          importance=TRUE, na.action=na.omit)
print(ozone.rf)
## Show "importance" of variables: higher value mean more important:
round(importance(ozone.rf), 2)

## "x" can be a matrix instead of a data frame:
set.seed(17)
x <- matrix(runif(5e2), 100)
y <- gl(2, 50)
(myrf <- randomForest(x, y))
(predict(myrf, x))

## "complicated" formula:
(swiss.rf <- randomForest(sqrt(Fertility) ~ . - Catholic + I(Catholic <
50),
                          data=swiss))
(predict(swiss.rf, swiss))
## Test use of 32-level factor as a predictor:
set.seed(1)
x <- data.frame(x1=gl(53, 10), x2=runif(530), y=rnorm(530))
(rf1 <- randomForest(x[-3], x[[3]], ntree=10))

## Grow no more than 4 nodes per tree:
(treesize(randomForest(Species ~ ., data=iris, maxnodes=4, ntree=30)))

## test proximity in regression
iris.rrf <- randomForest(iris[-1], iris[[1]], ntree=101, proximity=TRUE,
oob.prox=FALSE)
str(iris.rrf$proximity)
```