Selecting the best model for predicting a term deposit product take-up in banking

Rivalani Willie Hlongwane
HLNRIV002

Supervisors:

Professor K. Rajaratnam and Dr C-K. Huang

A thesis submitted in fulfilment of the requirements for the degree of
Master of Science
of the
University of Cape Town.

Department of Statistical Sciences

04 2018

Declaration

No portion of the work contained in this document has been submitted in support of an application for a degree or qualification of this or any other university or other institution of learning. All verbatim extracts have been distinguished by quotation marks, and all sources of information have been specifically acknowledged.

Signed: Signed by candidate

Date: April 2018

Abstract

In this study, we use data mining techniques to build predictive models on data collected by a Portuguese bank through a term savings product campaign conducted between May 2008 and November 2010. This data is imbalanced, given an observed take-up rate of 11.27%. Ling et al. (1998) indicated that predictive models built on imbalanced data tend to yield low sensitivity and high specificity, an indication of low true positive and high true negative rates. Our study confirms this finding. We, therefore, use three sampling techniques, namely, under-sampling, over-sampling and Synthetic Minority Over-sampling Technique, to balance the data, this results in three additional datasets to use for modelling. We build the following predictive models: random forest, multivariate adaptive regression splines, neural network and support vector machine on the datasets and we compare the models against each other for their ability to identify customers that are likely to take-up a term savings product. As part of the model building process, we investigate parameter permutations related to each modelling technique to tune the models, we find that this assists in building robust models. We assess our models for predictive performance through the use of the receiver operating characteristic curve, confusion matrix, GINI, kappa, sensitivity, specificity, and lift and gains charts. A multivariate adaptive regression splines model built on over-sampled data is found to be the best model for predicting term savings product take-up.

Acknowledgements

This journey required commitment, sacrifice and resilience, the success thereof is attributable to all the support I received throughout the journey, I thank you. I will like to thank my supervisors, Prof Kanshukan Rajaratnam and Dr Chun-Kai Huang for their unwavering support, continued assistance and encouragement. I will also like to thank my family for their patience whilst I burnt the candle at both ends, parents, particularly my mother for guiding and teaching me resilience.

Contents

List of Tables

List of Figures

Abbreviations and Terminology

AUC - Area Under Curve
CART - Classification And Regression Tree
CV - Cross Validation
CRM - Customer Relationship Management
GCV - Generalised Cross Validation
LDA - Linear Discriminant Analysis
MARS - Multivariate Adaptive Regression Splines
NN - Neural Network
OOB - Out Of Bag
PCA - Principal Component Analysis
RBF - Radial Basis Function
RF - Random Forest
RFM - Recency, Frequency, Monetary
ROC - Receiver Operating Characteristics
RSS - Residual Sum of Squares
SMOTE - Synthetic Minority Over-sampling Technique
SVM - Support Vector Machine

Chapter 1

Introduction

1.1 Introduction

Traditional banks face fierce competition from new players who are using technology to simplify banking. These new players are referred to as disruptive innovators. Disruptive innovation was first mentioned by Bower and Christensen (1995), it involves using technology to introduce new propositions in the market, forcing existing corporations to rethink how they do business. JUMO is one example of disruptive innovators; they analyse users' smart-phone data to calculate a "JUMO score" which is used to decide whether to offer loans to customers or not.

Marketing of products and services is normally done through mass or direct marketing, where mass marketing involves marketing to a wider audience and direct marketing is more personalised to a specific customer (Ling and Li, 1998). Whilst mass marketing has the potential to reach a larger audience, some of its shortcomings include the following:

- It is costly.
- It is difficult to precisely measure the success of a campaign. Suppose an advertisement to sell a product is aired on TV, it is difficult to identify customers that respond positively to a campaign due to the aired advert and customers that buy a product irrespective of whether it is advertised or not.
- It is difficult to target specific segments of the population.

The success of direct marketing, on the other hand, can be measured because it normally involves contacting a specific group within a population. On completion of a campaign, the results of the campaign can be analysed to assess performance. Costs can also be managed by contacting customers that are more likely to respond to a campaign. Identifying customers that are more likely to respond to a campaign is done through predictive modelling. An organisation must have the capability to build predictive models that score customers based on a variety attributes to determine the propensity to respond positively to a campaign. An organisation can then decide which customers to contact to achieve the desired response rate.

Lewis and Ling (2016) indicated that due to the scrutiny on the tobacco industry, there are limitations imposed on the industry from using traditional marketing to advertise tobacco products. They indicated that the industry had taken advantage of direct marketing and is benefiting from it.

Companies within the industry cite flexibility, efficiency and the ability to personalise messages directed at customers as some of the benefits of using direct marketing. These benefits are probably some of the main advantages of using direct marketing. This study is a good example of how direct marketing can be used in place of mass marketing to engage customers.

It is critical for a bank to have data mining capability to be in a position to understand what is in their data and to generate quality leads to sell products through marketing campaigns. Data mining is used to extract insights from data through the use of mathematical and statistical models (Witten and Frank, 2005). Data mining holistically covers the process of extracting data from databases, analysing and transforming data, and building models to predict a phenomenon of interest.

1.2 Problem Statement

In the past, banks sold most of their products to customers that walked into their branches. With the emergence of technology, there is a shift to selling products not only to customers that go to branches but to also use telemarketing, mobile applications and the Internet to sell products. This means that banks have to approach customers with a proposition as opposed to waiting for customers to visit their branches. The main focus of our study is on a savings product, and this is a typical product that most banks sell.

Information about existing customers' present new acquisitions, cross-selling and up-selling opportunities to banks and a bank can proactively sell an upgrade to a product that a customer has or sell a product that a customer is not in possession of but might otherwise need. To effectively use this information, predictive models are built to gain a deeper understanding of factors that drive take-up, predict the propensity to take-up a product and use gains chart to decide the number of customers to contact to achieve an expected response.

1.3 Aim and Objective

We use the outcome of a campaign aimed at selling a savings product of a Portuguese bank conducted between 2008 and 2010 to gain an understanding of features that affect take-up. The take-up rate of this specific campaign is 11.27%, the proportion of non-taken-up is, therefore, 88.73%, this response is considered to be imbalanced. The response will be considered to be balanced if both the proportions of take-up and non-taken-up is close to 50%.

The aims of the study:

- To gain a deeper understanding of direct marketing.
- To identify sampling techniques to apply to the campaign outcome data to create copies of balanced data in preparation for predictive modelling.
- To gain an understanding of features that influence savings product take-up.
- To explore various techniques used in marketing analytics to determine cut-off points for the purpose of selecting leads for a marketing campaign.

The objectives of the study:

- We study the literature on the use of data mining techniques to gain an understanding of direct marketing and recommender systems.
- We use the random forest permutation technique to rank the variable importance of our given data.
- We develop multivariate adaptive regression splines, neural networks, random forests and support vector machines, and compare these techniques against each other to identify one suitable for our problem.
- We employ the gain and lift charts, and Youden's index method to assist in selecting the best sensitivity and specificity cut-off point of a predictive model and the optimal number of customers to contact to obtain a desired response rate for a campaign.

1.4 Contribution to Literature

Affes and Hentati-Kaffel (2016) and Lee et al. (2006) indicated that MARS is a robust technique for solving classification problems. However, it is noticeably missing in term deposit take-up prediction literature. Our study shows that MARS is a competitive technique for predicting term deposit take-up.

Chapter 2

Literature Review

This chapter covers work done by other researchers. The insights gained from previous work will assist in strengthening our research and offer us an opportunity to appraise work done by others.

2.1 General Overview of Direct Marketing Models

Moro et al. (2014) used bank data collected between 2008 and 2013 to gain an understanding of features that affect term savings product take-up. Four classification models were developed and compared against each other for take-up prediction accuracy. The data used for the study contained 150 features, 52944 instances and a 12.38% take-up rate. This is exactly the same problem we are studying, the major difference is that whilst we use similar data for our research, our data only covers May 2008 to November 2010, contains 41188 instances, 20 independent variables and a 11.27% take-up rate. Due to privacy concerns, only a subset of instances and features is made available for our study.

Moro et al. (2014) investigated the following classification models:

- Logistic Regression
- Decision Trees
- Neural Network
- Support Vector Machine

Their models were built on data drawn from 2008 to June 2012, where the last 12 months (July 2012 to June 2013) data was used as a validation sample. The other approach Moro et al. (2014) used was to develop models on a rolling window data to ensure changes in micro and macro-economic factors are captured by the models.

Three feature selection approaches were used, the main approach was twofold, a domain expert was involved in a process of selecting features, this was followed by a data mining technique which involved selecting variables that resulted in the increased area under the ROC curve, AUC in short. This resulted in the reduction of features from 150 to 22. The other approach involved developing predictive models on all 150 features and the third approach involved using a forward selection method that reduced features to seven through a method that retained variables that increased AUC. *Euro Interbank Offered Rate*, *direction of call - inbound or outbound* and *agent*

experience were found to be the top three features. A LIFT curve was used to measure the performance of the models and to determine the number of contacts that must be made to reach 79% of customers that will take-up the savings product.

Neural Networks (NN) performed better than the other three classification models in terms of take-up prediction accuracy as measured by the AUC and cumulative LIFT curve. Moro et al. (2014) raised a caution that although NN yields good predictions, they are not easy to interpret by humans. They suggested that plotting Variable Effect Characteristic (VEC) curve assists in interpreting feature importance of NN models. The study by Moro et al. (2014) showed that data mining techniques can be used to select the best model for predicting term deposit product take-up in banking. The study showed that the economic environment can affect product take-up. We suggest that credit bureau indicators should be considered for predicting take-up. An example of credit bureau indicator is a credit bureau score, it decreases for customers whose monthly credit and insurance payments are not adhered to. An economic environment-specific credit bureau indicator is debt to income ratio of consumers, an increasing ratio can be used as an indicator of consumers in financial distress. Customers in financial distress are unlikely to have funds available to save.

The study to understand features that influence term savings product has been getting attention, recent work include that of Keles and Keles (2015) and Vaidehi (2016). They used data collected between May 2008 and November 2010 containing 17 features. Keles and Keles (2015) and Vaidehi (2016) found that *call duration* is a good predictor of term savings product take-up. *Call duration* is the duration of scheduled call to obtain a final answer from clients for the current campaign (Moro et al., 2014). However, Moro et al. (2014) advised that *call duration* as used by Keles and Keles (2015) and Vaidehi (2016) is not suitable for constructing predictive models, this feature is known once a call has been made to a prospective customer (Moro et al., 2014). This feature can be used to improve operational efficiency, and an example will be to identify leads that can be sent to customers' mobile applications instead of calling them. Ansari et al. (2008) found that customers can be successfully migrated from regular to web channels.

A Recommender System (RS) is a system used to make recommendations for items that users might need (Zhang et al., 2016). An example of an RS is the one used by the online retailer Amazon. They make recommendations that might be of interest to a shopper given that a shopper has already shown interest in one or more items on the Amazon online website.

Ricci et al. (2011) indicated that they have successfully deployed a generalized linear mixed models (GLMix) model to a job recommendation system. A GLMix is made up of a mix of generalized linear models (GLMs) with the objective of handling complex relationships in the data. They showed that the use of mix GLMs resulted in the increase of the response rate by between 20% and 40%.

Suppose a customer is subscribed to product A with a bank, cross-selling involves selling product

B as an addition to, upgrade or downgrade of product A. Zakirov and Momtselidze (2015) found that cross-selling is one of the areas that can be improved by applying data mining techniques. Chitra and Subashini (2013) pointed out that automatic credit limit increase for bank customers is an interesting area where data mining can be applied. According to Chitra and Subashini (2013), data mining layers can be summarised as follows:

- Problem understanding
- Customer data understanding
- Data transformation
- Feature selection and modelling
- Model testing
- Performance analysis

We agree with Chitra and Subashini (2013) that a bank that employs data mining techniques gains a competitive advantage over its peers.

Stone and Woodcock (2014) presented a paper which puts a lot of emphasis on business intelligence (BI) and marketing individuals working together to support interactive marketing. They indicated that interactive marketing involves customers using they mobile devices, for example, to search and acquire products they are interested in. They emphasised that it is critical for organisations that offer services to customers to have intrinsic knowledge of their customers through the use of data.

Guo et al. (2018) indicated that the lack of mining of e-commerce users' data limits the creation of predictive recommender systems. They proposed a neural network that assigns weights to items that might be of interest to customers to make recommendations to mobile users on an e-commerce system. E-commerce involves buying goods through an online platform. We agree with Guo et al. (2018) that the limitation in mining data available to companies limits the potential for companies' growth and an opportunity to improve customer experience. They used a Taobao online dress shop to validate their proposed solution and they found that their proposed system increased accuracy when compared with a traditional recommender system.

Bahari and Elayidom (2015) proposed a Customer Relation Management (CRM) framework based on NN and Naive Bayes classifiers to predict take-up of a term savings product. They indicated that banks should use direct marketing as one of the ways to improve customer development through cross-selling measures. Bahari and Elayidom (2015) proposed a CRM-data mining framework that involves understanding business requirements, customer identification, data preparation and transformation, model construction and model evaluation. They indicated that CRM efforts assist in identifying, attracting and retaining effective customers. They used AUC to measure the performance of NN and Naive Bayes classifiers, they found that NN performed better in predicting term savings product take-up. Their paper does not however get into specific details of feature

selection. In their study to provide a comprehensive framework to guide research efforts focusing on direct marketing strategy, Singoei and Wang (2013) indicated that feature selection is one of the steps that should be undertaken prior to model development. We believe it is important to study features for their strengths and weakness so we understand how they affect product take-up. This information will also be useful to campaign managers to gain an understanding of which features are important in product take-up.

Alhakbani and al Rifaie (2016) proposed a Hybrid of Data-level and Algorithmic-level solutions (HybridDA) to deal with imbalanced data. Ling et al. (1998) indicated that most data mining models are more predictive of majority class and minority class are miss-classified resulting in low predictive accuracy. Synthetic Minority Oversampling Technique (SMOTE) was chosen to oversample the minority class, i.e. customers that subscribed to the savings product. After balancing the instances, their data contained 2516 (49.66%) instances that subscribed and 2550 (50.33%) that did not subscribe to the savings product. A HybridDA was used for prediction and its accuracy was compared against results obtained by Moro et al. (2014), Vajiramedhin and Suebsing (2014), Feng et al. (2014), Elsalamony (2014) and Bahnsen et al. (2015). Alhakbani and al Rifaie (2016) showed that applying HybridDA to imbalanced data improves prediction accuracy.

Yap et al. (2014), Kim et al. (2015) and Diapouli et al. (2017) have applied over-sampling and under-sampling techniques to a variety of problems including bankruptcy prediction, prediction of death due to a medical condition and online behavioural targeting, respectively. They have demonstrated that over-sampling and under-sampling can be used to improve sensitivity of predictive models in the presence of data imbalance.

2.2 Classification Models in a Business Setting

An interesting problem in business is identifying customers that are likely to churn. This is important, Hadden et al. (2007) argued that it is much cheaper to retain existing customers than acquiring new ones. Ngai et al. (2009) proposed a framework that used classification techniques to assist in customers churn prediction. Miguéis et al. (2013) employed two classification methods, namely, MARS and LR to predict churn. They defined CRM as a customer-oriented culture created for customer acquisition, retention and profitability. Miguéis et al. (2013) found that MARS performed better than LR when variable selection procedures are not used and it loses its superiority when stepwise feature selection is used to construct LR.

Affes and Hentati-Kaffel (2016) used 2008 to 2013 BankScope data that contained 1247 instances of 411 banks that failed and 836 banks still active to gain an understanding of features that can be used as an early warning to identify banks that will experience bankruptcy. They chose ten features and three classification techniques for the study. The classification techniques chosen are, MARS, CART and hybrid MARS. They indicated that hybrid MARS is a method that combines K-means clustering and MARS, where K-means is a clustering method used to partition data into clusters. They found that some of the important features for predicting bankruptcy is *capital adequacy* and *liquidity*. The three classification methods were compared against each other using AUC. Affes

and Hentati-Kaffel (2016) found that hybrid MARS performed better than MARS and CART, and a comparison between MARS showed that MARS was superior except for the year 2008 where CART performed better.

Multicollinearity is a phenomenon in predictive modelling that affects parameter estimation. It results in the inflated variance of estimated parameters which can lead to incorrect predictions (Dormann et al., 2013). Multicollinearity occurs when two or more predictor variables are highly correlated with each other. Suppose we have two predictor variables x_1 and x_2, we say that these two variables are highly correlated if there is a strong linear relationship between them. Veaux and Ungar (1994) raised a point that MARS is faster to train, easier to interpret and accurate than NN when applied to a variety of problems. However, due to their nature of over parametrisation, NN tend to be insensitive to multicollinearity and they guard against it at the expense of interpretability (Veaux and Ungar, 1994). They suggested that it was the forward pass that MARS uses to construct a model that makes it vulnerable to multicollinearity. Veaux and Ungar (1994) proposed using Principal Components Analysis (PCA) to reduce dimensionality of features to improve the ability of MARS to deal with multicollinearity. A recent study by Dormann et al. (2013) on ecological data found that MARS is not adversely affected by multicollinearity but they cautioned that there is no guarantee that it will select correct predictors. The findings by Veaux and Ungar (1994) and Dormann et al. (2013) indicated that we should investigate the existence of multicollinearity in our bank data.

Grzonka et al. (2016) constructed four classification models on the same data as Moro et al. (2012) to gain an understanding of features that affect long-term savings product take-up of a Portuguese bank. They found that the top four features for predicting take-up was *call duration*, *day of the week*, customers' *job* and *age*. Grzonka et al. (2016) argued that *call duration* related to the current campaign cannot be used to construct a predictive model since it is only available after a telephone conversation with a customer. They subsequently omitted *call duration* as a candidate feature for constructing their model. We agree with Grzonka et al. (2016) that *call duration* cannot be used for predictive modelling in this setup. After *call duration* was omitted, their four best features were *previous campaign outcome*, *month customer called*, customers' *job* and *age*. Four classification models were compared against each other. They found that random forest (RF) had the lowest miss-classification rate when compared to a decision tree, bagging and boosted classifiers.

One of the interesting applications of data mining techniques is in deploying classification models to assist in identifying fraudulent activities. Alibaba is one of the biggest e-commerce company in the world, it enables consumers and businesses to purchase and sell goods online. Chen et al. (2015) indicated that data mining techniques are used by Alibaba to identify fraudulent transactions. On 11 November 2013, Alibaba reached a peak in daily transactions, 188 million transactions were processed that day (Chen et al., 2015). Transactions data processed by Alibaba is made up of hundreds of features including a combination of internal and external features (credit bureau, geospatial, etc.) (Chen et al., 2015). Although the amount of data involved is large, fraud identification and mitigation is done in real-time. Alibaba employs five layers to identify and prevent

fraud, a suspicious activity will trigger these five checks in the following sequence:

(i) Account Check

(ii) Device Check

(iii) Activity Check

(iv) Strategy Review

(v) Manual Review.

Chen et al. (2015) found that employing decision trees and RF result in successful fraud management at Alibaba.

Another interesting application of RF is by Kartasheva and Traskin (2013) who used a RF to predict property-casualty insurers' bankruptcy on data that is inherently unbalanced and non-linear. Kartasheva and Traskin (2013) found that RF performed better in predicting property-casualty insurance companies' bankruptcy than traditional classification methods. Due to their ability to handle unbalanced and non-linear data, RF compared favourably against logistic regression especially when compared against each other using *type I error* and *type II error*. In this instance, *type I error* indicates a prediction that a company will fail when it does not and *type II error* is a prediction that a company will not fail when it does (Kartasheva and Traskin, 2013). Kartasheva and Traskin (2013) highlighted that RF were used in the ranking importance of features, making it easier to understand underlying factors that lead to property-casualty insurance companies failing.

Support Vector Machine (SVM) is one of the most popular classification models that are applied to problems in different sectors such as geology, banking, retail, telecommunication, etc. as studied by Sindhu and Vijaya (2015), Moro et al. (2014), Xia and Jin (2008), Auria and Moro (2008) and Rodriguez-Galiano et al. (2015).

Kim et al. (2013) found that SVM struggled to make accurate predictions in the presence of imbalanced data. They studied three types of data that they classified to have low, moderate and high severity of imbalance. The description of the data is as follows:

1. Non-profit organisation data containing 100000 instances and 27.21% subscribers (donors). Severity of imbalance classified as low.

2. Upscale business mailing catalogues to its customers. The data contains 100000 instances and 9.42% subscribers, the imbalance severity is considered moderate.

3. Business that sells its products through catalogues. The data has 96551 instances and 2.47% subscribers, the imbalance severity of this data is considered high.

Kim et al. (2013) used a combination of Recency, Frequency and Monetary (RFM) features plus data under-sampling to construct SVM, logistic regression, decision tree and NN. They applied under-sampling to the data, this involved sampling the data such that 33% subscribers and 50%

subscribers were part of the sample to result in 2:1 and 1:1 samples, respectively. Model accuracy, sensitivity and specificity were used to measure and benchmark the models. Gain score which evaluates the response rate of all customers across deciles (usually 10) was used to measure the rate of response for the original, 2:1 and 1:1 samples. In direct marketing, we normally use the highest response within the first deciles, Kim et al. (2013) suggested that the five deciles are favoured to balance response rate and contacting fewer customers to optimise operations.

Kim et al. (2013) found that all models are affected negatively by imbalanced data, all the models produced low sensitivity. They also found that sensitivity increases when under-sampling is applied, this, however, comes at a price because there will be a reduction in accuracy and specificity. The models were found to produce higher sensitivity when 2:1 under-sampling method was applied as compared to 1:1 sampling. The reduction in accuracy and specificity in favour of sensitivity is not necessarily negative, especially when our goal is to predict subscribers to a campaign with higher accuracy.

Wisaeng (2013) used the same data as Moro et al. (2012) to predict term savings product take-up for a Portuguese bank through the use of decision trees, radial basis function network and SVM. Prediction accuracy, sensitivity and specificity were used to measure model performance, Wisaeng (2013) found that SVM performed better compared to the other methods.

Most researchers used NN to predict term savings product take-up on the data supplied by Moro et al. (2012) and Moro et al. (2014). In most cases, NN produced higher prediction accuracy and were found to perform better than the other methods. NN have been applied successfully to predict term savings take-up as seen in Moro et al. (2014), Keles and Keles (2015), Bahari and Elayidom (2015), Sharma1 and Chopra (2013) and Olson and Chae (2012). In an environment where the outcomes of predictive models need to be presented to business managers who in most cases are not data mining experts, most researchers have highlighted that results of NN are not easy to interpret, especially in cases where business managers have to understand features that are important in predicting an event (Moro et al., 2014). Moro et al. (2014) suggested using VEC to assist in interpreting results of NN.

Through the use of RFM features, Olson and Chae (2012) constructed four response models; namely, NN, decision tree, LR and RFM. Olson and Chae (2012) used two data-sets for their study, the first study used catalogue sales data from 1982 to 1992 and the second study used individual donors that contributed to a non-profit organisation from 1991 to 2006. The target was defined on the period August to December 1992 for the first study and August to December 2006 for the second study. The average response rate was 9.60% and 6.20% for the first and second studies, respectively. In both studies, they found that traditional data mining techniques, LR, NN and decision tree outperformed RFM model in both accuracy and cumulative gain. We agree with Olson and Chae (2012) that the error type is an important measure to gauge relative costs of selecting cut-off points of the percentage of customers to target. It is important in this case to measure

type I and *type II* errors because the data used by Olson and Chae (2012) is imbalanced. Classification models applied to imbalanced data tend to yield low sensitivity ratio, Kim et al. (2013) addressed this by under-sampling the data.

Sharma1 and Chopra (2013) investigated applications of NN to business problems, they found that NN can be applied to the following environments:

- Marketing and sales
- Finance and accounting
- Manufacturing and production
- Strategic management and business policy

Some of the advantages of NN include the following:

- High accuracy when compared to regression models
- Ability to handle variable interactions
- Parameters are easy to calibrate as new information is received
- Ability of handle missing and inaccurate data

The disadvantages of NN have been highlighted by Sharma1 and Chopra (2013) to be the following:

- Importance of features is not easy to interpret
- Results of neural networks are not easy to interpret
- Training neural networks takes time

2.3 Summary

NN have been widely used to solve a variety of classification problems including prediction of long-term savings product and in most instances have been found to offer robust predictions. MARS is notably missing from previous research aimed at predicting subscription of a long-term savings product even though it has been found to be robust in predicting classification problems than LR and other data mining models as mentioned for example in Miguéis et al. (2013). The application of MARS to classification problems on non-linear and imbalanced data as seen in the prediction of bank insolvency and churn as studied by Affes and Hentati-Kaffel (2016) and Miguéis et al. (2013), respectively, indicates that MARS has a potential to solve the problem of predicting long-term savings product take-up.

The variable, *call duration* is used extensively in constructing predictive models on the data supplied by Moro et al. (2012) and Moro et al. (2014), we disagree with this approach and support the warning provided by Moro et al. (2014) that this variable should not be used in this context

because its value is only available once a customer has been contacted.

Most researchers used AUC to assess the strength of their predictive models. However most researchers do not assess *type I* and *type II* errors, these errors should be assessed to ensure the chosen model predicts both minority and majority classes accurately. We have seen in the study by Kim et al. (2013) that data mining models applied to imbalanced data can yield low sensitivity ratios. This is a problem if the objective of constructing a data mining model is to predict minority classes. To counter the challenge of low sensitivity ratio, under-sampling of majority classes is suggested.

One of the challenges of constructing predictive models is multicollinearity, which results in models that have unstable coefficients (Dormann et al., 2013). Multicollinearity occurs when two or more independent variables are highly correlated with each other. Dormann et al. (2013) highlighted that MARS is not highly affected by multicollinearity and Veaux and Ungar (1994) suggested that NN is not affected by multicollinearity either. Most researchers do not cover multicollinearity analysis in their study of savings product take-up of a Portuguese bank even though they use LR for example which is known to be affected by multicollinearity. Midi et al. (2013) highlighted that LR is affected negatively by multicollinearity. To counter the effect of multicollinearity, Veaux and Ungar (1994) proposed using Principal Component Analysis (PCA).

Chapter 3

Theory of Models

This chapter introduces the theory behind the four modelling techniques we use to solve our classification problem.

3.1 Multivariate Adaptive Regression Splines (MARS)

MARS was introduced by Friedman (1991) for flexible regression modelling of high dimensional data. Lee et al. (2006) demonstrated that MARS can be used to model customers that have a propensity to default on a credit banking product. Miguéis et al. (2013) used MARS in a retail setting to identify customers that are likely to churn. In these applications, it has been shown that MARS can be used to solve classification problems. One of our objectives is to demonstrate that MARS can be used to predict customers that are likely to subscribe to a term savings banking product.

MARS is a non-parametric regression modelling technique that can be used to model non-linear relationships. It has the ability to identify non-linearity in the data, and it does this by fitting basis functions to help explain relationships.

Friedman (1991) proposed a typical MARS function to take the form,

$$f(X) = \beta_0 + \sum_{i=1}^{M} \beta_i B_i(X) \tag{3.1}$$

where $X = (x_1, x_2, ..., x_k)$ is a vector of k independent variables, x_k is an independent variable for some arbitrary k; and $B_i(X)$ is a basis function, $\forall i \in \{1, 2, ..., M\}$. A MARS function is made up of $M+1$ terms, where the first term is an intercept, β_0. M is the number of all other terms of a MARS function excluding the intercept, β_0. The coefficients β_i are jointly adjusted to give the best fit to the data (Friedman, 1991).

Let m be the number of interactions of hinge functions of a basis function; a basis function, $B_i(X)$, is given by:

$$B_i(X) = (h_1(x_1)).(h_2(x_2))...(h_m(x_m)) = \prod_{j=1}^{m} h_i(x_j) \tag{3.2}$$

where $h_i(x_j)$ is a hinge function given by either:

$$h_i(x_j) = \max(0, x_j - t) \tag{3.3}$$

or

$$h_i(x_j) = \max(0, t - x_j) \tag{3.4}$$

for some arbitrary i and x_j is an independent variable for some arbitrary j, where t is a constant indicating a knot location (Friedman, 1991). A MARS model with a higher m value will usually be more predictive than a model with a lower m value, however an increase in the value of m increases the complexity of a model and does not guarantee greater predictive power.

Suppose we have a non-linear predictor variable whose input values can be partitioned into linear regions. Knots define the beginning and end of each linear region. Within each region, linear functions are fitted, given by either equation 3.3 or 3.4, and these functions are tied together by knots.

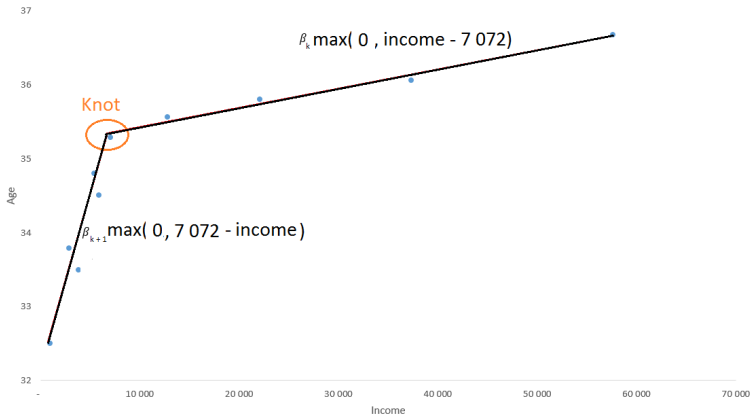

Figure 3.1: An illustration of how MARS partition non-linear data

Suppose we are interested in predicting a person's age (in years) from their monthly personal gross income. Figure 3.1 illustrates how MARS can be used to solve the given problem, it shows the relationship between *age* and *income*. The given example indicates that when *age* is about 35.5 and *income* is 7072, the data changes slope and become non-linear. MARS deals with this by fitting two basis functions with a knot defined when *income* equals 7072. The basis functions for this problem are given by the following equations:

$$BF_1 = \max(0, income - 7072) \tag{3.5}$$

and

$$BF_2 = \max(0, 7072 - income). \tag{3.6}$$

A MARS model is built in two stages, forward and backward passes. In the forward pass, data

is partitioned into regions through the use of a recursive regression model (Friedman, 1991), this involves placing candidate knots within the data range of each predictor variable (Zhang and Goh, 2016). A pair of basis functions is added to the equation with the objective of reducing the Residual Sum of Squares (RSS). The basis function will be of the form:

$$BF_n = \max(0, x_k - t) \tag{3.7}$$

and

$$BF_{n+1} = \max(0, t - x_k) \tag{3.8}$$

given that there are already $n-1$ basis functions in our model; x_k is an independent variable for some arbitrary k.

Suppose we are interested in predicting a person's age given their weight (in kilograms). Figure 3.2 illustrates how a pair of basis functions for the given problem are created. This example shows two basis functions that are created simultaneously during the model building process, tied by a knot when the value of the *weight* variable is given by 43. The simultaneously created basis functions are given by equations 3.9 and 3.10.

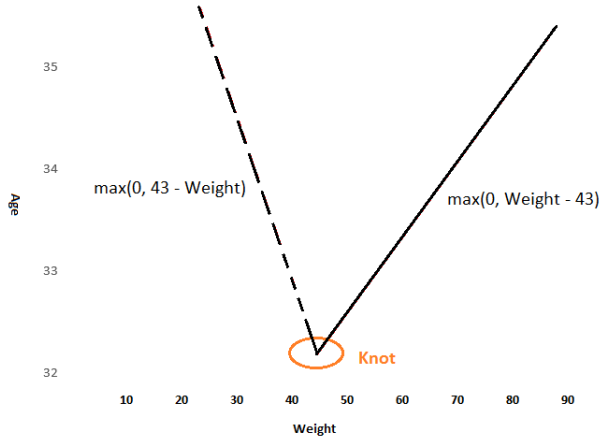

Figure 3.2: An illustration of a pair of basis functions

$$BF_1 = \max(0, weight - 43) \tag{3.9}$$

and

$$BF_2 = \max(0, 43 - weight) \tag{3.10}$$

.

In a case where MARS has been tuned to allow interaction between hinge functions, i.e. $m > 1$, then the algorithm searches for an existing compatible hinge function to combine with the new pair of hinge functions and when found, a new basis function is added to the equation. The additional basis functions added to the model are:

$$BF_n = BF_1 \max(0, x_k - t) \tag{3.11}$$

and

$$BF_{n+1} = BF_1 \max(0, t - x_k) \tag{3.12}$$

where BF_1 is a basis function that was already in the model and the new pair of basis functions is chosen to reduce the RSS. Basis functions are continuously added to the model until a predetermined number of basis functions is reached resulting in a over-fitted model (Zhang and Goh, 2016).

The backward pass starts with an over-fitted model built in the forward pass stage. Friedman (1991) extended the Generalised Cross Validation (GCV) method introduced by Craven and Wahba (1979) to prune over-fitted MARS models. The numerator of a GCV is the RSS and the denominator is a penalising function.

A GCV function is expressed as follows:

$$GCV = \frac{RSS}{[1 - \frac{M + d\frac{(M-1)}{2}}{N}]^2}. \tag{3.13}$$

The penalising function can be broken down as follows, M is the number of basis functions, d is a penalty term with a value normally 2 or 3, N is the number of data instances, and $\frac{(M-1)}{2}$ is the number of hinge function knots. A GCV function penalises a model that has higher number of basis functions and knots, a smaller GCV value is preferred (Zhang and Goh, 2016).

At each step, the backward pass removes basis functions that are less significant in predicting the outcome whilst penalising the model for having larger number of basis functions. Once complete, the backward pass produces a MARS model given in equation 3.1.

3.2 Neural Networks (NN)

NN are a set of algorithms used to solve classification problems. Moro et al. (2014) showed that NN perform better than three other classification algorithms in a study to identify customers that are likely to take-up a term savings product of a Portuguese retail bank.

NN have been getting a lot of attention in solving classification problems. This is attributable to their relaxed assumptions in dealing with non-linear problems. Multi-layer neural networks have the ability to generalise beyond immediate neighbours, as a result they are able to solve non-linear and complex tasks (Bengio and LeCun, 2007).

An NN takes a set of inputs, $a_i, \forall i \in \{1, 2, ..., n\}$, where n is the number of predictor variables used in the model. Each input variable is multiplied by the weight, $w_{i,j}$, where j is the number of hidden layers.

The sum of the product of each input variable and its associated weight is referred to as a pre-activation, z_j. It is common for a pre-activation to include a bias, a bias has a weight b and an input value 1. It is used to shift the activation function to the left or right to ensure best model fit.

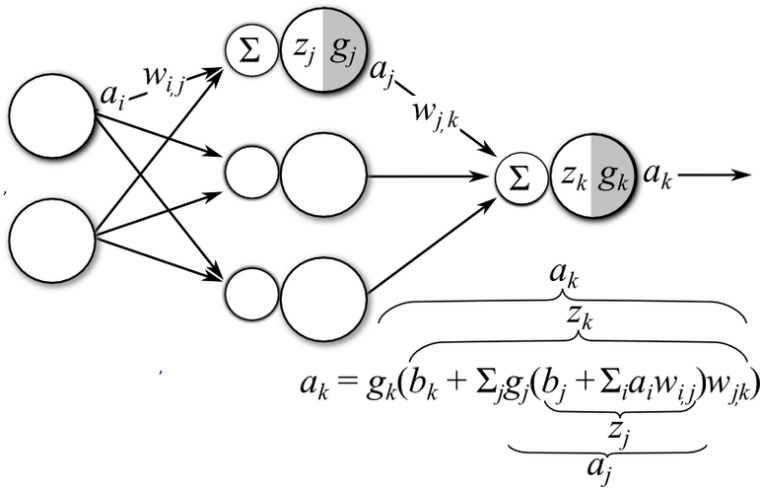

Figure 3.3: Diagram of a multi-layer NN.
Source: *Stansbury (2017)*

The most common activation functions take one of the following forms:

$$g = n, \tag{3.14}$$

$$g = \frac{1}{1+\exp(-n)} \tag{3.15}$$

or

$$g = \frac{\exp(n)-\exp(-n)}{\exp(n)+\exp(-n)}. \tag{3.16}$$

Equations 3.14, 3.15 and 3.16 represent linear, logistic and hyperbolic tangent activation functions, respectively. Selection of an activation function depends on the problem we are trying to solve (see Olgac and Karlik (2011)).

Hecht-Nielsen (1992) indicated that for each hidden layer, a pre-activation z_j is an input of an activation function g_j that has output a_j represented as follows:

$$a_j = g_j(b_j + \sum_j a_j w_{j,k}) = g_j(z_j). \tag{3.17}$$

The final output of the model is given by

$$a_k = g_k(b_k + \sum_j g_j(z_j) w_{j,k}). \tag{3.18}$$

NN use the Mean Squared Error (MSE) to determine the error of a model. Given a target, t_k, the error of our model is:

$$E = \frac{1}{2}(a_k - t_k)^2. \tag{3.19}$$

Once the error has been obtained, an error signal is calculated as a product of the derivative of the error and derivative of the activation function for the output and hidden layers. The error signal of the output is given by:

$$\delta_k = g_k^{'}(z_k)E^{'}(a_k - t_k). \tag{3.20}$$

The error signal of the hidden layer is:

$$\delta_j = g_j^{'}(z_j)\sum_k \delta_k w_{j,k}. \tag{3.21}$$

The forward signal a and backward signal δ are used to calculate the gradient of the error with respect to the weights w of each layer.

For each layer, l, this is given by:

$$\frac{\partial E}{\partial w_{l-1,l}} = a_{l-1}\delta_l \tag{3.22}$$

and for the bias, this is given by;

$$\frac{\partial E}{\partial b_l} = (1)\delta_l. \tag{3.23}$$

The weights of the model are updated as

$$w_{l-1,l} = w_{l-1,l} - \eta \frac{\partial E}{\partial w_{l-1,l}} \tag{3.24}$$

and the bias,

$$b_l = b_l - \eta \frac{\partial E}{\partial b_l}. \tag{3.25}$$

The learning rate, η is initialised upfront to determine the proportion of the weights gradient to use each time the weights are updated. The model continues to update the weights until it reaches a pre-determined number of iterations or when the squared error is not changing much with each iteration.

3.3 Support Vector Machine (SVM)

SVM is a classification method that uses a hyperplane to separate data into classes (Cortes and Vapnik, 1995). Rodriguez-Galiano et al. (2015) found that SVM performed better than NN and regression trees in predicting areas that have minerals.

We define two groups of customers; customers that take-up and those that do not take-up a term savings product. Given these two groups of customers, SVM seek to find a hyperplane defined by a vector w that can be used to separate these two groups. Suppose customers that take-up are of class $+1$ and customers that do not take-up are of class -1. Each class is made up of data points defined by (x_i, y_i) where x_i is a location of a point in the space and $y_i \in \{-1, +1\}$ for some arbitrary i. The space between these two groups is called a margin and it is defined as $\frac{1}{||w||}$.

SVM was introduced by Cortes and Vapnik (1995), they define a function:

$$f(x,w,b) = sgn(< w,x > +b) \tag{3.26}$$

where $< w,x >$ is the dot product of a given point and hyper plane vector; *sgn* is a signum function, it determines whether the output of the function $f(x,w,b) \in \{-1,+1\}$ is positive or negative.

To ensure that all the points in our training data sets are classified correctly, i.e. as take-up or no take-up, we require

$$\max \frac{1}{||w||}, \text{or equivalently} \tag{3.27}$$

$$\min \frac{1}{2}||w||^2 + \Sigma_i(C)(\xi_i) \tag{3.28}$$

such that $y_i(< w,x > +b) \geq 1 - \xi_i$ where C is trade-off parameter between risk and complexity, and ξ_i is the error margin of points that are not classified correctly (Cortes and Vapnik, 1995).

Equation 3.28 presents a constrained optimisation problem which can be expressed as a Lagrangian equation by introducing Lagrangian multiplies, α_i such that:

$$L(w,b,\alpha) = \frac{1}{2}||w||^2 + C\Sigma_i\alpha_i(y_i(< w,x_i > +b) - 1 + \xi_i). \tag{3.29}$$

The following Wolfe dual equation is built from equation 3.29:

$$\max \Sigma_i\alpha_i - \frac{1}{2}\Sigma_i\Sigma_j\alpha_i\alpha_j y_i y_j < x_i,x_j > \tag{3.30}$$

such that $\forall_i : 0 \leq \alpha_i \leq C$ (Cortes and Vapnik, 1995).

In equation 3.30, the dot product $< x_i,x_j >$ can be replaced by the kernel function $k(x_i,x_j) =< \phi(x_i),\phi(x_j) >$ to enable us to solve non-linear problems.

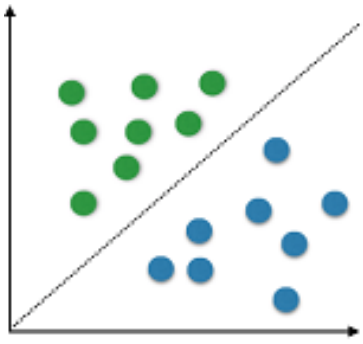

Figure 3.4: Diagram of linearly separable classes.
Source: *Schultebraucks (2017)*

Figure 3.4 is a representation of two classes represented by dots and squares. In our case, dots represent customers that take-up a savings product and squares represent customers that do

not take-up a savings product. The figure indicates clearly that these two classes can be separated by a hyperplane. The kernel used to solve linearly separable classes is given by equation 3.31.

$$k(x_i, x_j) = x_i^T x_j + c. \tag{3.31}$$

Most real-world problems are non-linear in nature and therefore require a method that can solve non-linearly spaced classes. SVM models use a kernel trick to solve problems of this nature, they do this by applying to the data, kernel functions that project classes into a higher dimensional space to result in classes that can be linearly separated.

Figure 3.5 is a representation of how data is transformed by applying a kernel trick, picture (a) represents classes that are not linearly separable, and picture (b) shows that a non-linear function is required to separate the two classes. Using a kernel trick involves applying a kernel function to the data resulting in picture (c) in Figure 3.5, indicating linearly separable classes.

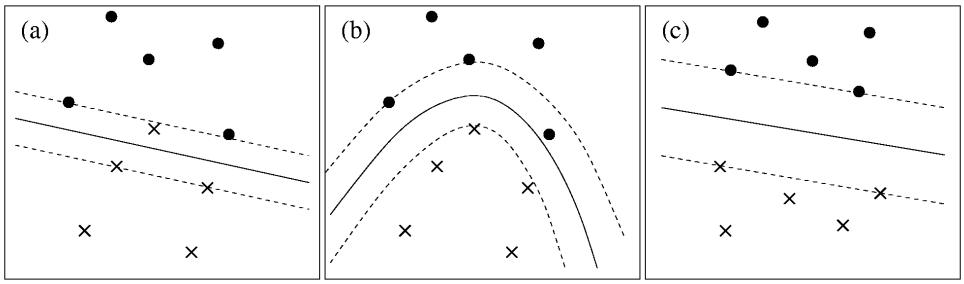

Figure 3.5: Diagram of non-linearly separable classes.
Source: *Zien et al. (2017)*

Examples of kernel functions that can be applied to data are polynomial, radial basis function (RBF) or sigmoid represented by equations 3.32, 3.33 and 3.34, respectively.

$$k(x_i, x_j) = (\alpha x_i^T x_j + c)^d, \tag{3.32}$$

$$k(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right) \tag{3.33}$$

or

$$k(x_i, x_j) = tanh(\alpha x_i^T x_j + c) \tag{3.34}$$

where σ, α and d are kernel parameters.

The parameter d in equation 3.32 is the degree of a polynomial, when $d = 1$ and $\alpha = 1$ this equation is the same as that of a linear kernel given in equation 3.31.

Moro et al. (2014) found that SVM yield less predictive power than NN. We believe that by balancing the data, using different kernel functions and a combinations of kernel parameters and

the trade-off parameter C, we can improve the performance of SVM.

3.4 Random Forest (RF)

An RF is made up of an ensemble of decision trees; the method was introduced by Breiman (2001).

It is important for us to gain an understanding of what decision trees are before we provide the methodology used to build an RF.

3.4.1 Decision Trees

In decision trees, a top-down tree is built by splitting predictor variables starting with the most predictive variables and ending with the decision nodes (leafs). The most predictive variable is the starting point of the tree, and it is referred to as the root node.

Suppose we have a set of attributes, $\{x_1,...,x_n\} \in X$, and a binary target variable, $T \in \{0,1\}$. The information gain of each attribute on the target variable is calculated and the attribute with the highest information gain is used as the root node.

Information gain determines which attribute is more useful in discriminating between the responses of a target variable. Rutkowski et al. (2014) indicated that information gain is calculated as follows:

$$Gain(T,x_i) = Entropy(T) - Entropy(T,x_i), \forall i \in \{1,2,...,n\}. \tag{3.35}$$

Entropy measures the homogeneity of classes of some attribute, $x_i, \forall i \in \{1,2,...,n\}$ in response to the target variable, where the entropy of 0 shows a completely homogeneous set with all the responses the same and a value of 1 indicates a response that is equally divided amongst classes (Rutkowski et al., 2014).

Rutkowski et al. (2014) indicated that the entropy of the target variable is given by,

$$Entropy(T) = \Sigma_i - p_i \log_2 p_i; \tag{3.36}$$

and the entropy of the interaction between target and attributes:

$$Entropy(T,x_i) = \Sigma_{\text{all classes of } x_i}(P(c))(Entropy(c)), \tag{3.37}$$

where $P(c)$ and $Entropy(c)$ are probability and Entropy of classes of some attribute $x_i \in \{1,2,...,n\}$, respectively.

The decision tree is made up of only predictive variables with the most predictive variable used as a root node and the other variables and their classes used as branches, referred to as decision nodes. The bottom of the tree is made up of the target/decision, referred to as leaf nodes.

A fully constructed decision tree is traversed by the ID3 greedy algorithm which searches the tree

for a decision. Once a branch is chosen, the algorithm does not backtrack, it continues searching to the bottom of the tree until a decision is reached. An ensemble of decision trees is constructed to form an RF, a decision tree will typically look like the tree illustrated in Figure 3.6, in this illustration, a decision tree is used to predict if a person drinks alcohol or not. The tree illustrates that the most predictive variable is *age*, split by individuals older than 30 years and, 30 years and younger. The other predictive variables in the decision tree are *gender* and *currently studying*, a variable indicating the level of education.

An example of how the decision tree can be interpreted is as follows, individuals older than 30 years that are currently not studying or enrolled for a postgraduate degree are predicted to consume alcohol.

3.4.2 Methodology Random Forest

Breiman (2001) proposed the following methodology to build an RF model; given a training data set, T, with N rows and m features, an RF is built as follows:

(i) Two-thirds of the data is randomly selected (with replacement) to form a subset data, S, through a process called *bagging*.

(ii) Approximately $\sqrt{m}$ features from the data set, S are randomly selected to form a new data set U in a process called *attribute bagging*.

(iii) The data set U is used to build a decision tree.

(iv) The remaining data that is not selected in step (i) is called Out Of Bag (OOB) sample and it is used to validate the decision tree.

Steps (i) - (iv) are repeated to add decision trees into the RF until a point when adding more trees does not improve the accuracy of the prediction. Oshiro et al. (2012) suggest that an RF made up of 64 to 128 decision trees is sufficient for prediction.

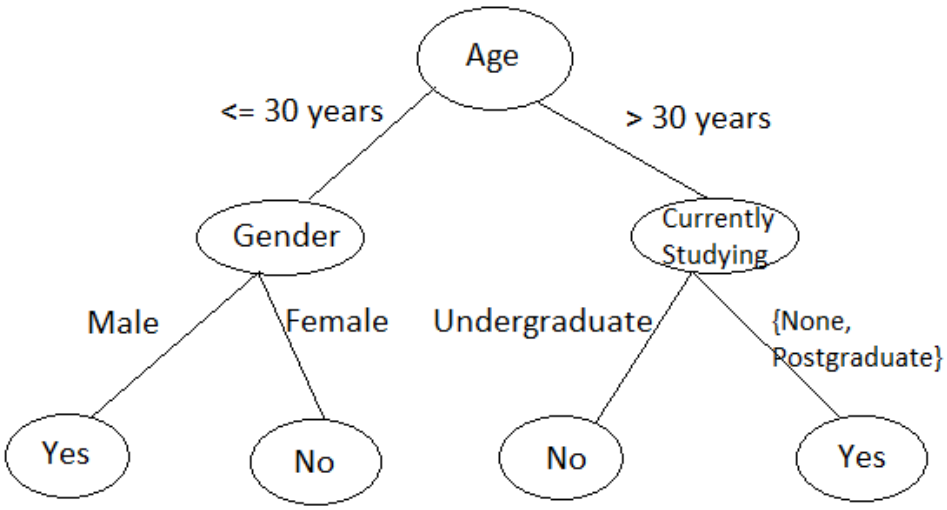

Figure 3.6: An example decision tree: Predict whether a person drinks alcohol or not

Chapter 4

Data and Methods

This chapter introduces the data used for the research and covers feature transformation and model evaluation.

4.1 Research Data

Our study involves gaining an understanding of features that affect take-up of a long-term savings product of a Portuguese bank. We use the data made available by Moro et al. (2014) through the University of California, Irvine (UCI) machine learning repository; we refer to this data as the original data. The data contains customers that were contacted between May 2008 and November 2010 with a long-term savings product offer. Our data comprises of 41188 instances and 21 independent variables, due to privacy concerns, only a subset of instances and features were made available for our study.

The reader should note that Moro et al. (2014) selected twenty-two variables for modelling from a total of hundred and fifty variables. Only seven of the twenty variables were made available for our study, the other, fifteen variables that Moro et al. (2014) used are not made available due to privacy concerns. Figure 4.1 illustrates the intersection of the variables used by Moro et al. (2014) as well as variables we used in our study.

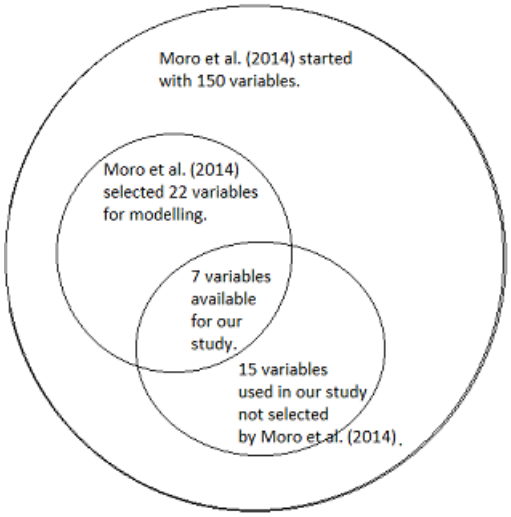

Figure 4.1: Variables available for modelling from Moro et al. (2014) data

Although twenty-two independent variables were made available for our study, we omit two variables, namely, *monthyear* and *call duration* in the modelling phase. The reason for not using the variable *call duration* is discussed at the end of this section. The variable *monthyear* is used to select the period to be used for modelling. Some of the information for the variable *monthyear* are embedded in the variables *Month*, the month a customer was contacted and *day_of_week*, the day of the week; Monday to Sunday, as a result this variable is not included in the modelling phase.

The reader should note that the overall response rate observed between May 2008 and November 2010 is 11.27%. We propose using a methodology described below to select a subset of the data for analysis. The period we considered has a higher response rate and therefore the response rate adopted in our analysis is 23.61%.

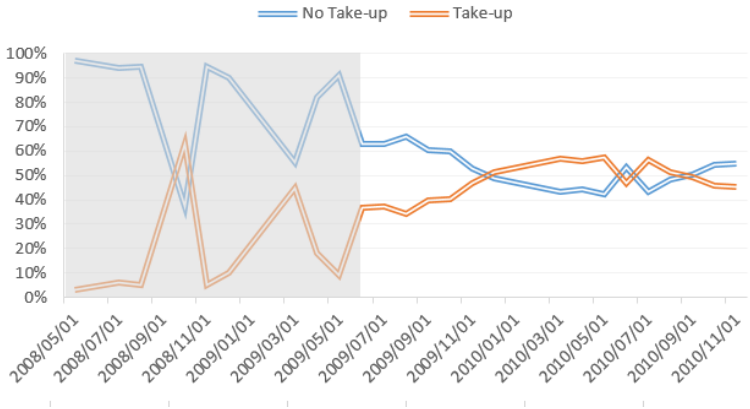

Figure 4.2: Monthly savings product take-up

Moro et al. (2014) recommend that predictive models should be updated using the most recent data as it becomes available to ensure changes in the economy and the propensity to take-up is integrated into the models. We use Figure 4.2 to gain an understanding of the developing trends in the propensity to take-up a savings product. We find that between May 2008 and May 2009 the trend (highlighted in the grey background) is erratic, however, from June 2009 the trend is more stable. The take-up rate between April - May 2009 and June 2009 - November 2010 is 11.71% and 44.50%, respectively. The possible cause of the low take-up rate in the former period is the economic downturn experienced by most countries including Portugal between 2008 and 2009. Canals-Cerda and Kerr (2015) indicated that building models that include cyclical changes in the economy result in robust models. Moro et al. (2014) used the most recent data to develop propensity models, we, therefore, follow this approach and select the data between April 2009 and November 2010 for further analysis and modelling to ensure that the training sample includes different cycles in the economy.

Table 4.1 describes variables used for our analysis. We use the words variables and features interchangeably in our study. The response variable, which we refer to as a *class* variable is not listed in Table 4.1, it can be described as a binary variable having either a value "yes" or "no" where yes indicates a successful long-term savings product take-up and no indicates the opposite.

	Variable	Type	Description
1	Contact	Categorical	Contact number type e.g. cellular, telephone etc.
2	Job	Categorical	Customer job type
3	Marital	Categorical	Marital status
4	Education	Categorical	Education level
5	Default	Categorical	Credit accounts default status
6	Housing	Categorical	Mortgage account indicator
7	Loan	Categorical	Personal loan account indicator
8	Poutcome	Categorical	Outcome of the previous marketing campaign
9	Month	Categorical	Last contact month of year
10	day_of_week	Categorical	Last contact day of the week
11	Age	Numeric	Age of customer
12	Campaign	Numeric	Number of contacts performed during this campaign and for this client
13	Pdays	Numeric	Number of days that passed by after the client was last contacted from a previous campaign
14	Previous	Numeric	Number of contacts performed before this campaign and for this client
15	call duration	Numeric	Last contact call duration, in seconds
16	Emp_var_rate	Numeric	Employment variation rate - quarterly indicator
17	Cons_price_idx	Numeric	Consumer price index - monthly indicator
18	Cons_conf_idx	Numeric	Consumer confidence index - monthly indicator
19	Euribor3m	Numeric	Euribor 3 month rate - daily indicator
20	Nr_employed	Numeric	Number of employees - quarterly indicator

Table 4.1: Variables description

Our study involves developing predictive models, and as indicated in the literature review, the variable *call duration* (variable number 15 in Table 4.1) is not used in our study because it is only known once calls have been made to customers. Moro et al. (2014) advised against using this variable to develop predictive models.

4.2 Data Preparation

Our data is made up of categorical and numerical variables. One of the challenges of working with numerical variables is the possible presence of outliers. Outliers occur when there are values of a variable that deviate drastically from most of the points (Aguinis et al., 2013). This poses a challenge where an estimated model over-compensates its parameters to accommodate outliers resulting in a less accurate model. Martin and Roberts (2010) and Aguinis et al. (2013) proposed using visual tools such as a box plot to identify outliers, they suggested restricting the bottom and top values to maximum values of the 2.5th and 97.5th percentiles, respectively.

Figure 4.3 represents the box plots of nine numerical variables in the data. The distribution of the numerical variables for both groups, i.e. subscribers (taken-up) and non-subscribers (non-taken-up) of a long-term savings product is represented in each box plot. The dots at either end of a box plot indicate the presence of outlier(s). Our approach to treating outliers involves setting

values below the 2.5th percentile to the maximum value of the 2.5th percentile and values above 97.5th percentile are set to the maximum value of the 97.5th percentile, as suggested by Martin and Roberts (2010) and Aguinis et al. (2013). The following numerical variables: *age*, *previous*, *pdays* and *campaign* were treated to remove outliers. For example, the *age* variable, the bottom value is set to 24 and the top value to 72, where the minimum and maximum values were previously 17 and 98, respectively.

Figure 4.4 is a visual representation of the ten categorical variables in the data. Each plot indicates the proportion of subscribers and non-subscribers in each input level of the categorical variable. For example, the variable *contact type* indicates the channels used to contact customers in the previous campaign, customers were contacted either via telephone or cellular phone. Most customers were contacted via a cell phone, and of these customers, 24.63% subscribed to a savings product, whilst customers that were contacted via a telephone had a lower subscription rate, 18.47%.

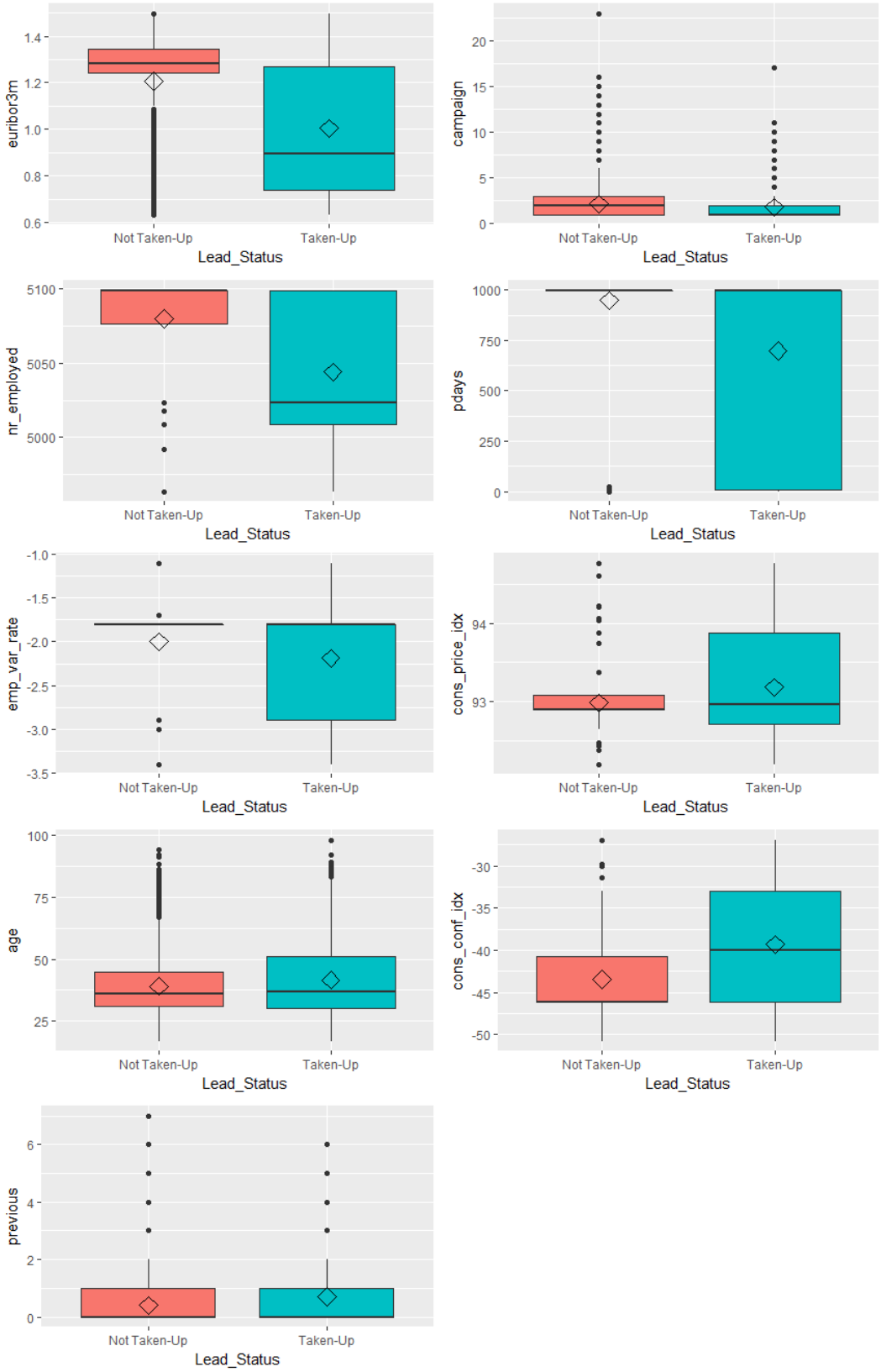

Figure 4.3: Box Plot - Original numerical variables

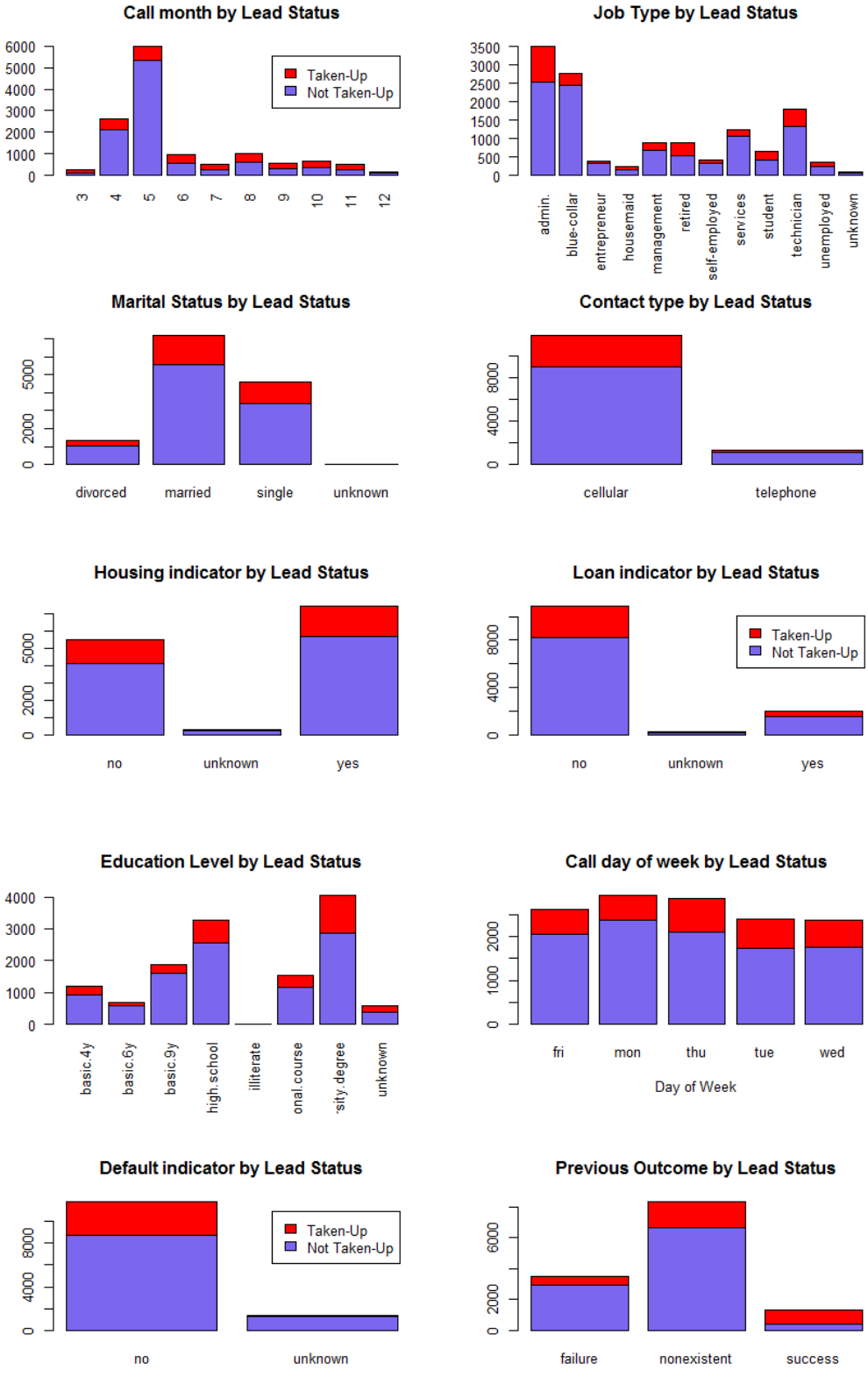

Figure 4.4: Original categorical variables

The Wald test was proposed by Abraham Wald in 1943 (Dobek et al., 2015), it is used to test if variables are statistically significant (Kyngäs and Rissanen, 2001). We test the significance of factor levels for all our categorical variables using the Wald test; we find that all factor levels of

our categorical variables are statistically significant except for the variable, *education*. The Wald test results given in Figure 4.5 indicate that the factor level associated with the value *illiterate* is not statistically significant.

```
Coefficients:
                                Estimate Std. Error z value Pr(>|z|)
education basic.4y             -1.09091    0.06629 -16.456  < 2e-16 ***
education basic.6y             -1.67868    0.10580 -15.866  < 2e-16 ***
education basic.9y             -1.83274    0.06692 -27.386  < 2e-16 ***
education high.school          -1.29772    0.04268 -30.407  < 2e-16 ***
education illiterate            0.00000    0.81650   0.000        1
education professional.course -1.03593    0.05777 -17.933  < 2e-16 ***
education university.degree    -0.86968    0.03440 -25.279  < 2e-16 ***
education unknown              -0.65212    0.08742  -7.459  8.7e-14 ***
---
```

Figure 4.5: Factor levels of the variable *education*

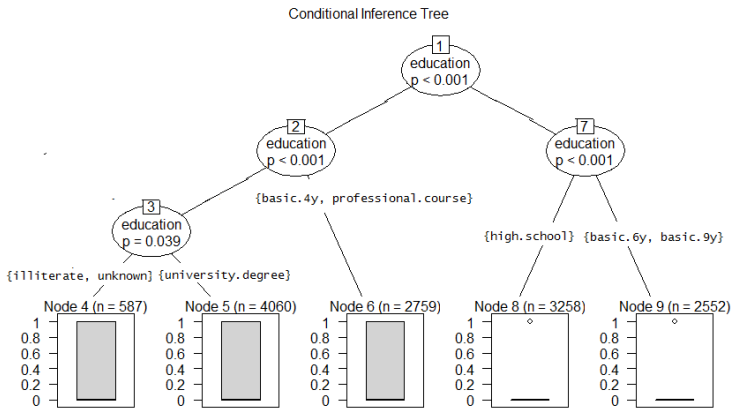

Figure 4.6: Decision Tree - Optimal variable split

Decision trees can be used to reduce the number of categorical variable input levels by combining the input levels that have similar traits (Song and Lu, 2015). We use a decision tree to obtain the optimal input level splits for the variable, *education*, the tree shows that *unknown* and *illiterate* can be combined. Performing the Wald test on the variable, *education*, indicates that by combining the inputs, *unknown* and *illiterate* into one category results in factor levels that are statistically significant as indicated in Figure 4.7.

```
Coefficients:
                                   Estimate Std. Error z value Pr(>|z|)
education_new basic.4y             -1.09091    0.06629 -16.456  < 2e-16 ***
education_new basic.6y             -1.67868    0.10580 -15.866  < 2e-16 ***
education_new basic.9y             -1.83274    0.06692 -27.386  < 2e-16 ***
education_new high.school          -1.29772    0.04268 -30.407  < 2e-16 ***
education_new professional.course -1.03593    0.05777 -17.933  < 2e-16 ***
education_new university.degree    -0.86968    0.03440 -25.279  < 2e-16 ***
education_new unknw_illitrate      -0.64498    0.08688  -7.424 1.14e-13 ***
```

Figure 4.7: Collapsed factor levels of the variable *education*

4.3 Sampling

Sampling is an important component of our study. It will assist us to divide our data into training, test and validation samples, balancing response variable classes, and to select more recent data for additional model testing.

Data collected between May 2008 and March 2009 is omitted as discussed in Section 4.1, our final data contains customers that were contacted between April 2009 and November 2010. Newer data collected between October and November 2010 is set aside, we use it to validate our models. The newer data sample is referred to as the out-of-time validation sample. This approach to selecting a sample from the data is referred to as judgemental sampling (Ishak and Bakar, 2014).

On the other hand, stratified sampling is a probability-based sampling that involves selecting a subset of data from a complete sample whilst ensuring the stratum (dependent variable, in our case) input levels have similar proportion in both training and testing data (Ishak and Bakar, 2014). We apply stratified sampling to the data observed from April 2009 to September 2010, where 70% and 30% of the data are allocated to model training and testing, respectively.

Our data is comprised of two classes, subscribers (23.61%) and non-subscribers (76.39%) of a long-term savings product. This data is not made up of 50/50 split between positive and negative responses. Developing predictive models on data of this nature can result in models that are more predictive of majority classes than minority classes. Alhakbani and al Rifaie (2016) suggested using Synthetic Minority Oversampling Technique (SMOTE) which involves over-sampling by creating synthetic copies of minority class to balance the data.

Yap et al. (2014) found that over-sampling and under-sampling can improve the accuracy of minority class prediction. Under-sampling involves reducing instances of the majority class by discarding information relating to the majority class (Yap et al., 2014). The obvious shortfall of under-sampling is information loss since it involves discarding instances of the majority class to balance the minority class.

Over-sampling involves creating copies of the minority class adding them in the data to increase the minority class instances (Yap et al., 2014). The drawback of the over-sampling method is over-fitting due to the fact that we add copies of the minority class to increase the response rate.

In addition to developing predictive models on 70% of the original data, over-sampling, under-sampling and SMOTE sampling is applied to the original data to create additional copies of modelling data. Applying SMOTE, under- and over-sampling to the original data changes the distribution of the classes and the training data sample size. Table 4.2 illustrates the distribution of the classes in our data.

It is important to note that applying sampling to balance data is only applicable to the training data, the test data is not balanced. Balancing data to optimise model performance results in a shift

	Treatment	Training sample size	Proportion : Take-up	Proportion : No Take-up
1	None	9019	23.61%	76.39%
2	Under-sampling	4258	50.00%	50.00%
3	Over-sampling	13780	50.00%	50.00%
4	SMOTE sampling	9019	50.80%	49.20%

Table 4.2: Training data: Distribution of response (classes)

of the response distribution, the predicted probability of take-up will be based on the new distribution. When applying the predictive model to the test data and ultimately new cases, the model will produce predicted probabilities based on the balanced data. To ensure that we obtain probabilities of the true population, the probability score obtained on the test data and new cases needs to be calibrated, this is done by using function A.1 found in the Appendix.

To ensure that each instance of the training data is used for model development and testing, cross-validation (CV) is used. Yang and Huang (2014) advised that CV works by splitting training data into k mutually exclusive samples of equal sizes, selected randomly. A model is trained on $k-1$ samples, the kth sample is used to test the model. This is repeated until all the samples have been used for testing and participated $k-1$ times in the model development. To assist in deriving a more accurate model, the sample errors of each model are averaged out. Depending on the size of the data, Hastie et al. (2009) advised that the value of k should be 5 or 10.

4.4 Feature Selection

Feature selection is also known as variable selection, it is used to eliminate variables that are statistically insignificant and less predictive of the outcome, reduce dimensionality of data and to avoid model over-fitting. Over-fitting is problematic in predictive modelling, its presence results in a situation where a model predicts the outcome of the training data, however, the accuracy of the model diminishes on newer data.

Song and Lu (2015) indicated that decision trees can be used for variable selection, however, we are concerned that decision trees tend to suffer from over-fitting as noted by Farid et al. (2014) and Bramer (2013). Random forest reduces the chance of overfitting by using an ensemble of decision trees to form a forest. Breiman (2001) indicated that a larger forest is less likely to overfit than a smaller forest. The study by Chen and Ishwaran (2012) indicated that the permutation importance measure, a feature of random forest can be used to rank variable importance.

We propose using a random forest model to select possible features for modelling. Suppose we are investigating the importance of an independent variable, x_k, for some arbitrary k, to predict an outcome. The method involves the following:

(i) Build a random forest that includes the variable x_k.

(ii) Calculate accuracy of the random forest using out-of-bag error, e_k.

(iii) Permute values of the variable, x_k, call the rearranged variable, x_p for some arbitrary p.

(iv) Build another random forest, use x_p in place of x_k.

(v) Calculate the accuracy of the random forest using out-of-bag error calculation, e_p.

(vi) Compare errors obtained in (i) and (v), a value of e_p larger than e_k indicates x_k is possibly an important predictor variable.

Gregorutti et al. (2013) found that this method is effective in identifying predictive variables.

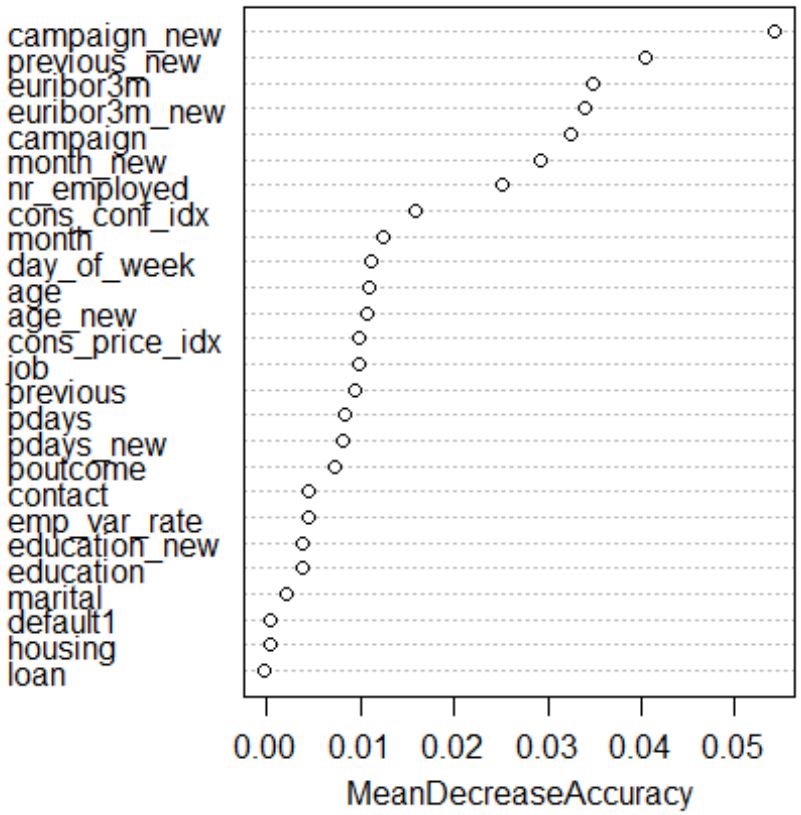

Figure 4.8: Variable Importance

Figure 4.8 illustrates the importance of the candidate variables (including the variables transformed in Section 4.2). The reader should note that the suffix *_new* indicates a transformed variable whilst a variable without this suffix indicates the original variable without any treatment. The variables are ordered from top to bottom starting with the most to the least important variable, in this case, the most important variable is *campaign_new* and the least important variable is *loan*. In instances where we have an original and transformed variable, the least predictive variable of the two will be dropped. The following variables are dropped: *campaign, euribor3m_new, month, previous* and *pdays*.

One of the challenges of constructing predictive models is multicollinearity, which results in models that have unstable coefficients (Dormann et al., 2013). We use a correlation matrix to gain an

understanding of the correlation between variables. Figure 4.9 gives us an indication of the correlation between the variables, we consider variables that have a correlation of 0.80 or higher to be highly correlated (Franke., 2010), we therefore discard the least predictive variable. The variable, *euribor3m* is highly correlated to the variables *cons_conf_idx* and *nr_employed* with correlation values higher than 0.80. The variable *euribor3m* is the most predictive variable, we therefore discard the variables *cons_conf_idx* and *nr_employed*.

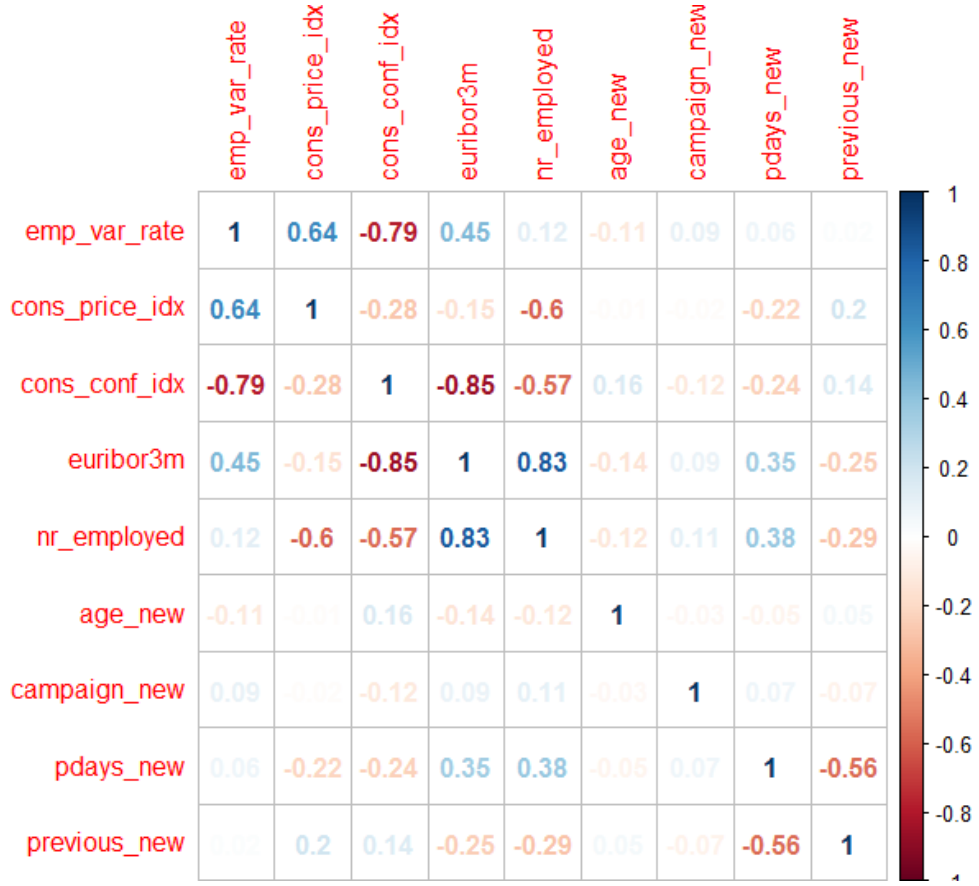

Figure 4.9: Correlation matrix

We used data transformation to remove outliers from the numerical variables, and re-classified the input levels of the categorical variables that had minimal representation of the overall distribution of the outcome. This increased the number of independent variables from twenty to twenty-six, however due to the nature of the transformation applied in the attributes, we cannot use both the original and reconstructed variables together for prediction, and therefore one of the variables is dropped. We then used the random forest model to determine which of the two variables is the most predictive and we dropped the least predictive variable. We used a correlation matrix to determine the variables that are highly correlated with each other, and in the presence of multicollinearity, the most predictive variable is retained and the least predictive variable is discarded. Table 4.3 outlines the transformed and non-transformed variables.

	Variable	Type	Use transformed variable?	Keep Variable for modelling?
1	Campaign	Numeric	Yes	Yes
2	Euribor3m	Numeric	No	Yes
3	Month	Categorical	Yes	Yes
4	Age	Numeric	Yes	Yes
5	Previous	Numeric	Yes	Yes
6	Pdays	Numeric	Yes	Yes
9	Education	Categorical	Yes	Yes
7	Marital	Categorical	Not Applicable	Yes
8	Default	Categorical	Not Applicable	Yes
10	Cons_price_idx	Numeric	Not Applicable	Yes
11	Emp_var_rate	Numeric	Not Applicable	Yes
12	Job	Categorical	Not Applicable	Yes
13	Contact	Categorical	Not Applicable	Yes
14	day_of_week	Categorical	Not Applicable	Yes
15	Poutcome	Categorical	Not Applicable	Yes
16	Housing	Categorical	Not Applicable	Yes
17	Loan	Categorical	Not Applicable	Yes
18	Nr_employed	Numeric	Not Applicable	No
19	Cons_conf_idx	Numeric	Not Applicable	No
20	Call Duration	Numeric	Not Applicable	No

Table 4.3: Variables details

4.5 Model Evaluation

Model evaluation is an important aspect of predictive modelling. It gives us an indication of whether a model meets its main objectives, which is to predict new cases with acceptable precision.

One of the most common metrics used for assessing model performance is a Receiver Operating Characteristics (ROC) curve. Examples of ROC curve use can be found in Breiman et al. (2004) and Moro et al. (2014). Figure 4.10 is a representation of a ROC curve, the blue line is called a line of equality, it is essentially a 45° line and the red line is called a Lorenz curve. The area represented by the sum of **A** and **B** is called the Area Under Curve (AUC), it is the measure of this area that tells us how precise a model is in predicting the outcome. If the area under the red line is given by the sum of **A**, **B** and **C**, then the model predicts the outcome perfectly. In such cases, the red line will be stretched to cover area **C**. However, if the red line moves along the blue line then the model is equivalent to selecting instances of interest randomly, that is, we do not require a model to select campaign leads. The $y-axis$ represents the true positive rate, in this case, subscribers of a savings product. On the other hand, $x-axis$ represents cases whose actual outcome is non-subscribed and the model predicts subscribed.

The AUC of a model is given by:

$$AUC = area(\boldsymbol{A}) + area(\boldsymbol{B}). \tag{4.1}$$

We use the GINI statistic to measure the performance of our models, Bekkar et al. (2013) indicated that model performance can be expressed as a GINI measure given by the following function:

$$GINI = 2 * AUC - 1. \tag{4.2}$$

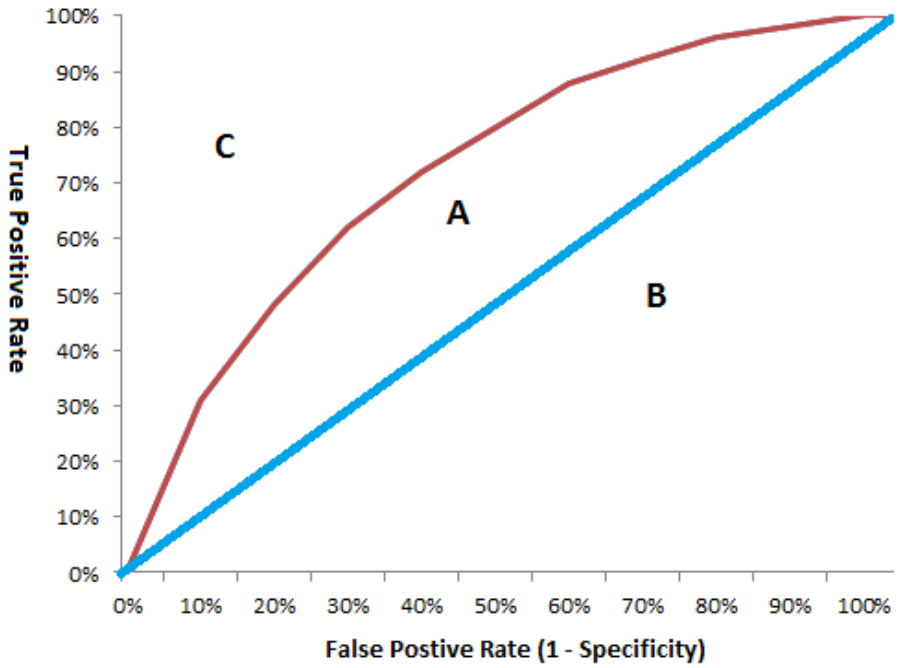

Figure 4.10: Receiver Operating Characteristics

Bekkar et al. (2013) suggested that the scale for model performance interpretation is given by Table 4.4.

Model Performance	GINI range
excellent	0.80 - 1
good	0.60 - 0.80
fair	0.40 - 0.60
poor	0.20 - 0.40
fail	0 - 0.20

Table 4.4: Guideline of model performance using GINI

The GINI statistic is an ideal measure of the overall model performance, however, to gain an understanding of how accurately a model predicts majority and minority classes, a confusion matrix is used. Applications of a confusion matrix in literature can be found in the studies by Lee et al. (2006), Muzir (2013) and Keles and Keles (2015).

A confusion matrix, such as the one in Table 4.5 is used to compare actual and predicted out-

	Actual : Yes	**Actual : No**
Predicted : Yes	True Positive (TP)	False Positive (FP)
Predicted : No	False Negative (FN)	True Negative (TN)

Table 4.5: Confusion matrix

comes. Test data created in Section 4.3 has actual campaign outcome, it is scored by the model to predict the outcome. Customers that subscribe to the savings product are regarded as positives and non-subscribers are regarded as negatives. True negative (TN) and true positive (TP) refers to instances where actual and model outcomes are in agreement. However, false negative (FN) and false positive (FP) refers to instances where actual and model outcomes are not in agreement. Suppose the test data indicates a customer subscribed to a savings product and the model predicts that this customer did not subscribe, this is referred to as a false negative. On the other hand, a false positive is when a model predicts a customer subscribed to a savings product whilst in actual fact they did not.

A cumulative gains chart is used to gain an understanding of the proportion of customers to contact for a campaign to achieve a desired response rate. Related work on cumulative gain chart use is covered in Olson and Chae (2012) and Kim and Street (2004). We demonstrate how cumulative gains charts are used in Figure 4.11. Suppose we only have capacity to contact 40% of a given population for a campaign. A cumulative gains chart assists us to determine an approximate proportion of positive responses that the campaign will yield given the constraint. The $x-axis$ represents the overall population size binned into ten deciles by decreasing propensity to subscribe to an offer, and $y-axis$ represents the proportion of customers that will respond to a campaign. The shaded area below the curve in Figure 4.11 indicates that if we only contact 40% of customers, we expect to convert 80% of the customers that have a propensity to subscribe.
Zacharis (2016) and Shiny et al. (2015) indicated that in addition to using a gains chart to visualise and gain an understanding of model performance, a lift chart is used. A lift chart is used widely in targeted marketing campaigns to establish whether using a predictive model is any better than selecting leads randomly. The model that we choose has to perform better than selecting a pool of leads randomly.

The probability of take-up produced by the chosen model is sorted in descending order, ranking customers from best to least probability to respond positively to a campaign. The ranked sample is divided into smaller samples of equal proportions, usually, ten, referred to as deciles. A calculation is performed at each decile as a proportion of responses divided by the proportion of the total population of the current decile, this gives us the lift of a model. Suppose Figure 4.12 is derived from a predictive model that predicts take-up of a savings product, the curve indicates that for

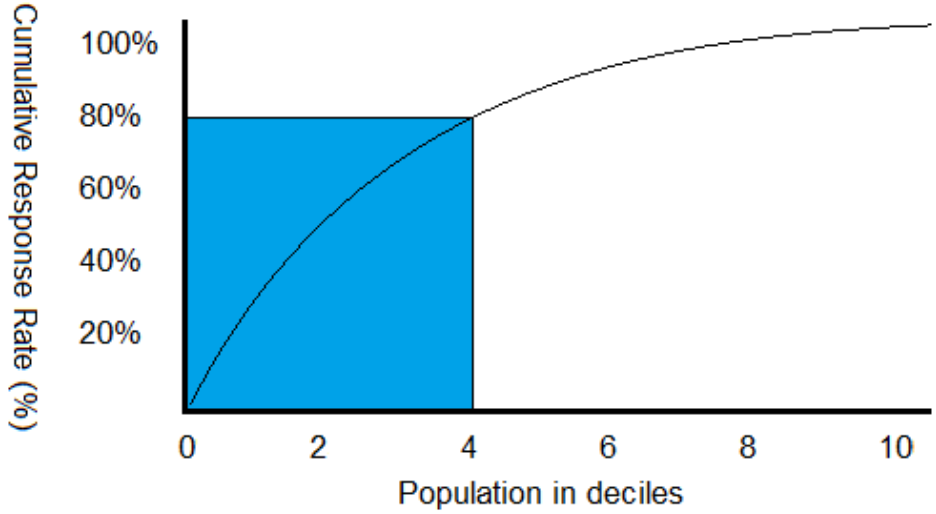

Figure 4.11: Cumulative Gain Chart

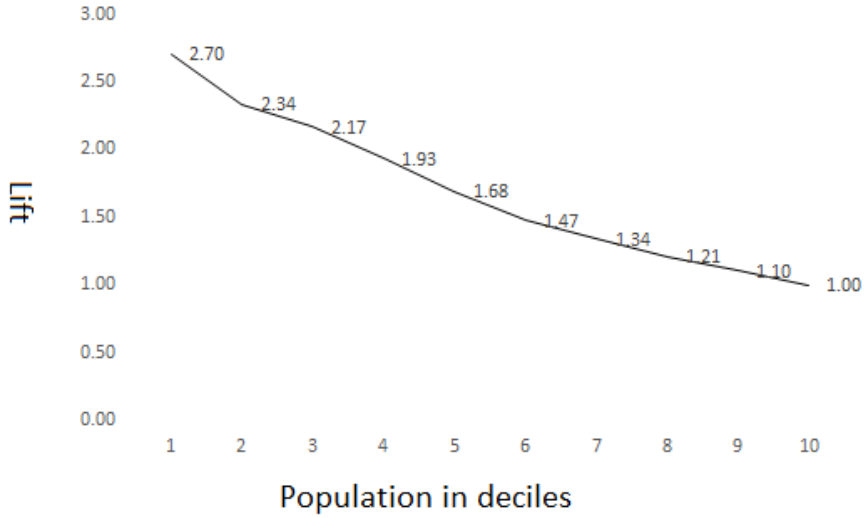

Figure 4.12: Lift curve

the first decile, 10% of the total population, the lift is given by 2.70, this indicates that the model yields 2.70 higher positive responses than selecting leads randomly. Furthermore, considering the first five deciles, 50% of the total populations, the model yields 1.68 higher positive responses than selecting leads randomly. However, considering the 10^{th} decile, i.e. the total population, the model will not yield any benefit because all the leads are contacted.

In addition to assessing our models using the methods described above, we also use another measure, a kappa statistic. A kappa statistic was introduced by Cohen (1968) to compare the outcome of a classifier and the actual distribution of the outcome in order to gain an understanding of how much agreement there is between a classifier and the actual outcome distribution. Table 4.6 gives us a guideline to assist in interpreting the kappa statistic as proposed by Landis and Koch (1977),

whilst Fleiss (1981) proposed classifier performance given in Table 4.7.

Kappa	Agreement
< 0.01	Less than chance agreement
0.01 - 0.20	Slight agreement
0.21 - 0.40	Fair agreement
0.41 - 0.60	Moderate agreement
0.61 - 0.80	Substantial agreement
0.81 - 0.99	Almost perfect agreement

Table 4.6: Landis and Koch (1977) guideline of model performance using kappa statistic

Kappa	Agreement
< 0.40	Poor
0.40 to 0.75	Intermediate to Good
> 0.75	Excellent

Table 4.7: Fleiss (1981) guideline of model performance using kappa statistic

4.6 Summary

In this chapter, we introduced the research data. Data transformation was applied to some of the numerical features to remove outliers. Decision trees were used to re-allocate insignificant input levels of the categorical variables. Most of the variables we transformed were shown to be more predictive than non-transformed variables, this emphasises the importance of data transformation in preparation of building predictive models.

We used the random forest model to identify predictive variables and their predictive strength. To avoid drawbacks that result from multicollinearity, we used a correlation matrix to identify highly correlated variables. The most predictive variable is retained, and the least predictive variables were discarded in case there is presence of correlation higher than 0.8. This resulted in the following variables being removed: *emp_var_rate* and *nr_employed*.

In preparation of building predictive models, the data is split between training and test data where the training data is used for model development and the test data is used to test model performance. To further validate our models on independent and newer cases, we have set-aside the data made up of 332 instances, observed between October and November 2010. The newer data sample is referred to as the out-of-time validation sample (Berg, 2007).

In addition, copies of SMOTE, over-sampling and under-sampling data are created from the training sample, this is done so we can build our models on the original data which is imbalanced and the other sets of models on the balanced data. This will assist us to assess if there is any benefit to building our models on the balanced data.

The accuracy of models is assessed through the use of the GINI statistic which is determined through the calculation of the AUC obtained from an ROC curve. A confusion matrix is used to assess *type I* and *type II* errors. To gain an understanding of the proportion of customers to contact for a campaign to achieve the desired response rate, a cumulative gains chart curve is proposed.

Chapter 5

Results and Analysis

We build NN, RF, MARS and SVM models to determine which of these is suitable for predicting take-up of a savings product in a bank. This chapter covers the steps undertaken to develop the models, in addition, we cover model performance.

The models are developed on the data described in Table 4.2, our reason for not focusing only on the original data is supported by Alhakbani and al Rifaie (2016) and Yap et al. (2014) who found that the error associated with classifying positive responses is reduced when applying SMOTE, under-sampling and over-sampling to the data. Cross-validation of 5-folds is chosen to train our models, this is in line with the recommendation made by Hastie et al. (2009), we choose this value as opposed to a higher value with the goal of ensuring that the execution time of training our models is lower. The models are built using R statistical software version 3.4.1.

5.1 Multivariate Adaptive Regression Splines

Table 5.1 outlines the parameters we use to build the MARS models on the original, under-sampled, over-sampled and SMOTE datasets.

	Dataset type	**Degree**	**Nprune**	**Model validation**	**Validation splits**
1	Original	1	19	Cross Validation	5
2	Under-sampled	1	19	Cross Validation	5
3	Over-sampled	1	20	Cross Validation	5
4	SMOTE	1	21	Cross Validation	5

Table 5.1: MARS model tuning

The tuning parameter $degree = 1$ indicates that, the basis function (BF),

$$B_i(X) = \prod_{j=1}^{m} h_i(x_j) \tag{5.1}$$

is given by $m = 1$, indicating that our models will not have a product of hinge functions.

We build four MARS models on each dataset where $m \in \{1,2,3,4\}$, and as illustrated in Table 5.2, the highest degree of the models is four, a degree higher than four does not increase model performance. The GCV decreases as the degree of the models' increases as seen in Table 5.2. A GCV function penalises a model that has a higher number of basis functions and knots, a smaller

GCV value is preferred (Zhang and Goh, 2016). Although we observe an increase in accuracy, kappa and AUC statistics as the degree of the models is increased, the improvement is generally on the third decimal place of the indicated statistics, we believe the improvement is marginal and not enough for us to choose higher degree models which in their nature are more complex than models of $degree = 1$.

Dataset	degree	AUC	GCV	Accuracy	Sensitivity	Kappa
Original	1	0.839	0.129	0.822	0.476	0.446
Original	2	0.841	0.128	0.821	0.490	0.454
Original	3	0.843	0.127	0.825	0.470	0.455
Original	4	0.843	0.127	0.825	0.470	0.455
Under-sampling	1	0.835	0.172	0.750	0.753	0.499
Under-sampling	2	0.842	0.169	0.756	0.776	0.511
Under-sampling	3	0.842	0.168	0.762	0.788	0.524
Under-sampling	4	0.842	0.168	0.762	0.788	0.524
Over-sampling	1	0.841	0.170	0.755	0.764	0.510
Over-sampling	2	0.840	0.167	0.754	0.773	0.508
Over-sampling	3	0.836	0.166	0.762	0.789	0.523
Over-sampling	4	0.833	0.166	0.760	0.788	0.521
SMOTE	1	0.815	0.179	0.732	0.759	0.464
SMOTE	2	0.826	0.175	0.743	0.758	0.485
SMOTE	3	0.826	0.173	0.746	0.741	0.491
SMOTE	4	0.826	0.173	0.746	0.741	0.491

Table 5.2: MARS models different degree scenarios

A MARS model is built in two stages, forward and backward stages. The forward pass involves adding basis functions and dummy variables to the model to reduce the residual sum of squares, resulting in an over-fitted model. The backward pass involves pruning the model by removing hinge functions and dummy variables that are less significant in predicting the outcome. The parameter $Nprune$ is part of the R programming language, it controls when the backward pass should stop pruning the model. A value $Nprune = 1$ indicates a model that is only made up of the intercept.

Categorical variables are automatically converted to dummy variables by the R model building function, all ten categorical variables in our data are converted to dummy coded variables, we end up with forty-nine candidate variables (including numerical variables).

	Dataset type	AUC	GINI	Accuracy	Sensitivity	Specificity	Kappa
1	Original	0.839	0.679	0.820	0.476	0.926	0.446
2	Under-sampled	0.834	0.668	0.750	0.753	0.746	0.499
3	Over-sampled	0.839	0.678	0.755	0.764	0.745	0.510
4	SMOTE	0.815	0.630	0.732	0.759	0.705	0.464

Table 5.3: MARS model performance on training data

The results in Table 5.3 represent the performance of our MARS models on the training data, whilst Table 5.4 illustrates the performance of our models on the test and out-of-time validation

Sample type	Dataset type	Accuracy	Sensitivity	Specificity	Kappa
Test	Original	0.815	0.457	0.922	0.421
Out-of-time validation	Original	0.732	0.623	0.823	0.452
Test	Under-sampled	0.750	0.752	0.742	0.425
Out-of-time validation	Under-sampled	0.714	0.695	0.729	0.424
Test	Over-sampled	0.753	0.761	0.742	0.433
Out-of-time validation	Over-sampled	0.714	0.656	0.762	0.420
Test	SMOTE	0.738	0.625	0.773	0.352
Out-of-time validation	SMOTE	0.605	0.709	0.519	0.223

Table 5.4: MARS model performance on test data

samples. The GINI of all our MARS models fall within the range 0.600 - 0.800, as indicated in Table 4.4, all our MARS models are rated as good. Model sensitivity indicates how well a model is able to predict positive responses whilst specificity indicates how well a model can predict negative responses. A kappa value between 0.410 and 0.600 indicates a moderate agreement between the model and actual distributions, all the MARS models fall within this range, with the model built on the over-sampled data having the highest kappa statistic.

The model built on the original data has the highest accuracy compared to the other MARS models whilst sensitivity and kappa statistics are the lowest. Although the accuracy and specificity of this model are the highest, the low sensitivity compared to the other MARS models indicates that this model struggles to predict customers that are likely to take-up a savings product. A similar performance is observed between the training and test samples. A higher sensitivity on the out-of-time validation sample is observed, this is attributable to a change in take-up rate as illustrated in Figure 4.2. The higher sensitivity on the out-of-time validation sample indicates that the model will identify a higher number of cases that are likely to take-up a savings product.

The model built on the under-sampled data yields a similar performance between the training and test samples, however the performance of the model on the out-of-time validation sample yields lower and fairly close results. The higher sensitivity of this model compared to the model built on the original data indicates that higher number of positive cases will be identified.

The model built on the over-sampled data yields the highest sensitivity and kappa statistics, this indicates that this model will identify a higher number of customers' that are likely to take-up a savings product. Although the kappa statistic on the test and out-of-time validation samples are lower than that of the training sample, the accuracy of the training and test samples is similar. We observe that the sensitivity of the model on the out-of-time validation sample is lower than the sensitivity on both the training and test samples.

SMOTE involves over-sampling by creating synthetic copies of the minority class to balance the data. We observe that the kappa statistics of the test and out-of-time validation samples are lower

and not comparable to that of the training sample. We also observe a low specificity on the out-of-time validation sample, and this indicates that the model will have a higher number of false positives.

Our problem requires that we predict customers that are likely to take-up a savings product, we, therefore, require a model that will yield high sensitivity and specificity. Where a model with high specificity will assist us to identify customers that will respond negatively to a campaign, a high sensitivity model will lead us to more customers that are likely to respond positively to a campaign. The model built on the original data has the lowest sensitivity by a large margin compared to the other models, we believe it is not suitable for this problem. The model built on the SMOTE data has the highest sensitivity and lowest specificity on the out-of-time validation sample, it will, therefore, yield a higher number of false positives, this will lead to higher number of customers' contacted and a low positive response. Although the sensitivity of this model is the highest on the out-of-time validation sample compared to the other models, it is important to get a balance between sensitivity and specificity, we, therefore, conclude that this model is not suitable for our problem because of its performance on the out-of-time validation sample.

The model built on the over-sampled has a lower sensitivity compared to the model built on the under-sampled data, however, it compares favourably on the other model performance statistics. The model built using over-sampled data is our chosen MARS model because it gives a balance between sensitivity and specificity on the training, test and validation samples, this will lead to lower false positive rate and ability to identify customers that have a high propensity to take-up. The confusion matrix of the test data for each MARS model for the four datasets is given in Table 5.5.

Dataset type		**Actual - Yes**	**Actual - No**
Original	**Predicted - Yes**	420	242
Original	**Predicted - No**	475	2728
Under-sampled	**Predicted - Yes**	697	776
Under-sampled	**Predicted - No**	198	2194
Over-sampled	**Predicted - Yes**	706	766
Over-sampled	**Predicted - No**	189	2204
SMOTE	**Predicted - Yes**	559	673
SMOTE	**Predicted - No**	336	2297

Table 5.5: MARS confusion matrix - test data

Table 5.6 outlines the terms used to build our chosen MARS model, ordered by variable importance from the most important (*Euribor3m*) to the least important (*default1unknown*). Figure 5.1 is a graphical representation of variable importance.

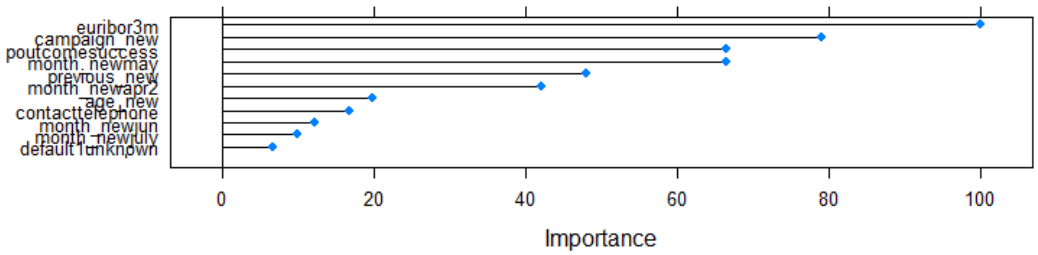

Figure 5.1: Variable Importance - MARS (over-sampled data)

	Model term	Variable type	Model term impact
1	(Intercept)		Positive
2	h(euribor3m-0.56307)	Numeric	Positive
3	h(0.742412-euribor3m)	Numeric	Negative
4	h(euribor3m-0.855872)	Numeric	Negative
5	h(euribor3m-0.892472)	Numeric	Positive
6	h(euribor3m-1.09743)	Numeric	Negative
7	h(campaign_new-0.127391)	Numeric	Negative
8	h(campaign_new-1.04503)	Numeric	Positive
9	h(campaign_new-1.96267)	Numeric	Negative
10	poutcomesuccess	Binary	Negative
11	month_newmay	Binary	Positive
12	h(0.825928-previous_new)	Numeric	Negative
13	h(previous_new-0.825928)	Numeric	Negative
14	month_newapr2	Binary	Negative
15	h(-0.216617-age_new)	Numeric	Negative
16	h(age_new- -0.216617)	Numeric	Negative
17	contacttelephone	Binary	Positive
18	month_newjun	Binary	Negative
19	month_newjuly	Binary	Negative
20	default1unknown	Binary	Positive

Table 5.6: Terms used to build MARS models (over-sampled data)

5.2 Neural Networks

We investigate building NN models using linear, logistic and hyperbolic tangent activation functions. We find that the hyperbolic tangent activation function results in more accurate models as outlined in Table A.1 found in the Appendix. All our NN models are therefore built using the hyperbolic tangent activation function.

Table 5.7 outlines the parameters used to tune our NN models. The decay is used to penalise the weights of an NN model resulting in a less complex model. We use a grid search to choose the best combination of decay parameter and hidden nodes that result in the best model for each data sample. A higher decay value indicates that the penalty imposed on higher NN weight is greater. A less complex NN model is one that has less number of hidden nodes, in our case, the models built on the original and under-sampled data are less complex than the other models.

	Dataset type	Decay	Number of hidden nodes	Model validation	Validation splits
1	Original	0.4	1	Cross Validation	5
2	Under-sampled	0.5	1	Cross Validation	5
3	Over-sampled	0.1	5	Cross Validation	5
4	SMOTE	0.5	5	Cross Validation	5

Table 5.7: Neural Network model tuning

The GINI statistics of all our NN models fall within the range 0.600 - 0.800, as indicated in Table 4.4, all our NN models are rated as good. Table 5.8 and Table 5.9 illustrate the performance of the NN models on the training, and test and out-of-time validation samples, respectively.

	Dataset type	AUC	GINI	Accuracy	Sensitivity	Specificity
1	Original	0.808	0.616	0.822	0.384	0.957
2	Under-sampled	0.811	0.622	0.740	0.723	0.758
3	Over-sampled	0.848	0.696	0.776	0.778	0.775
4	SMOTE	0.829	0.658	0.751	0.772	0.730

Table 5.8: Neural Network model performance on training data

Sample type	Dataset type	Accuracy	Sensitivity	Specificity	Kappa
Test	Original	0.805	0.339	0.946	0.342
Out-of-time validation	Original	0.744	0.503	0.945	0.465
Test	Under-sampled	0.740	0.727	0.741	0.388
Out-of-time validation	Under-sampled	0.753	0.735	0.768	0.502
Test	Over-sampled	0.721	0.693	0.750	0.374
Out-of-time validation	Over-sampled	0.684	0.735	0.641	0.371
Test	SMOTE	0.713	0.616	0.751	0.317
Out-of-time validation	SMOTE	0.590	0.669	0.525	0.190

Table 5.9: Neural Network model performance on test data

The model built on the SMOTE data has the lowest kappa statistics on the test and out-of-time validation samples when compared to the other NN models. Specificity is also the lowest on the out-of-time validation sample, an indication that this model will yield a higher rate of false positives. We do not believe this is a suitable model for this problem because of the low kappa statistics and lower specificity which will lead to increased false positives.

We observe that the model built on the original data has the highest accuracy compared to the other NN models. Although the accuracy and specificity statistics of this model are the highest, the low sensitivity compared to the other NN models indicate that this model struggles to predict customers that are likely to take-up a savings product. A higher sensitivity is observed on the out-of-time validation sample than the test sample, this is attributable to a change in take-up rate as illustrated in Figure 4.2. The higher sensitivity on the out-of-time sample indicates that this model will identify a higher number of cases that are likely to take-up a savings product.

The model built using the under-sampled data yield similar performance on the training, test and validation samples. The higher sensitivity of this model compared to the model built on the original data indicates that higher number of positive cases will be identified. The model built on the over-sampled data yields the highest GINI and sensitivity statistics, however, the kappa statistics on the test and out-of-time validation samples are lower.

Where a model with high specificity will assist us to identify customers that will respond negatively to a campaign, a high sensitivity model will lead us to more customers that are likely to respond positively to a campaign. The model built on the original data has the lowest sensitivity by a large margin compared to the other models, we believe it is not suitable for this problem. The model built on the SMOTE data has a lower specificity which will lead to increased false positives, it is therefore not suitable for our problem.

The model built on the under-sampled data has consistent performance statistics between the training, test and out-of-time validation samples, it also has the highest sensitivity, specificity and kappa statistics compared to the model built on the over-sampled data, we, therefore, choose this model over the model built on the over-sampled data because of high and consistent performance statistics on the test and out-of-time validation samples. The confusion matrix of the NN models on the training sample is given in Table 5.10.

Dataset type		Actual - Yes	Actual - No
Original	**Predicted - Yes**	303	161
Original	**Predicted - No**	592	2809
Under-sampled	**Predicted - Yes**	651	770
Under-sampled	**Predicted - No**	244	2200
Over-sampled	**Predicted - Yes**	620	744
Over-sampled	**Predicted - No**	275	2226
SMOTE	**Predicted - Yes**	551	741
SMOTE	**Predicted - No**	344	2229

Table 5.10: Neural Network confusion matrix - test data

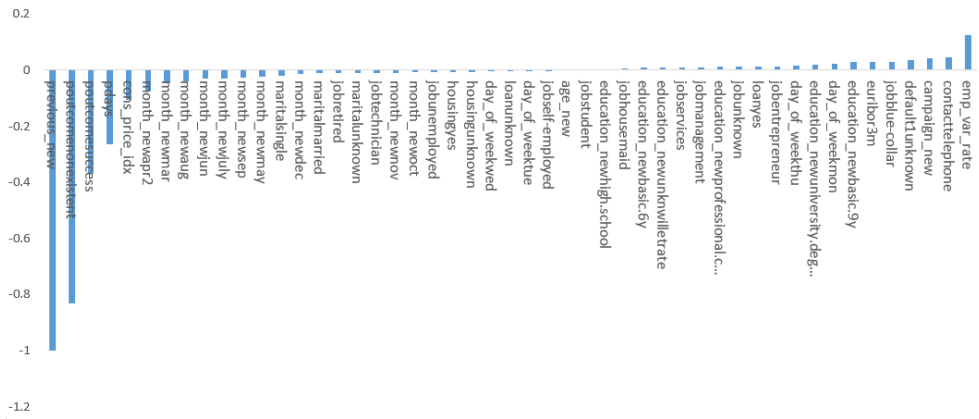

Figure 5.2: Variable Importance - Neural Network (under-sampled data)

The weight associated with each variable ranges from -1 to 1. Variables that have a weight less than zero have an inverse relationship with product take-up whilst the opposite is true for variables that have a positive weight. The farther the weight of a variable is to zero, the stronger the variable will be in predicting the outcome.

Figure 5.2 shows all variables that are used in the NN model built on the under-sampled data, the top three strongest variables that have a negative effect on product take-up are *previous_new*, *poutcomenonexistent* and *poutcomesuccess* indicating that customers that have been contacted frequently in the past or where the previous contact related to a campaign was either a success or no information exists are unlikely to take-up a savings product. The top three variables that have the strongest positive relationship in our NN model are *emp_var_rate*, *contacttelephone* and *campaign_new* indicating that when employment rate is high, a telephone is used as a contact channel or number of campaign contact for a savings product are high then a customer is most likely to take-up. The variable that has the least influence in the model is *age_new* indicating that the age of the customer is not important in predicting take-up.

The variable importance plot provides a way to visually represent features that contribute positively and negatively to how customers will respond to a campaign that seeks to sell a savings product.

5.3 Random Forest

An RF model is made up of an ensemble of decision trees which form a forest. Each decision tree in an RF model makes a prediction of whether a customer will take-up a savings product or not. The predictions from the decision trees are tallied up, and the final prediction outcome is based on a majority vote.

Forty-nine variables derived from seventeen predictor variables are used to build RF models from the original, under-and over-sampled, and SMOTE samples. Each decision tree in an RF model is built on randomly selected (with replacement) two-thirds of a given data, the remainder of the data is used for out-of-bag testing. Furthermore, a subset of predictor variables is selected at random and used to build each decision tree.

Table 5.11 outlines parameters used to build decision trees in an RF model, hundred and fifty decision trees were built per RF in each of the four datasets. Furthermore, thirteen variables were used to build decision trees on the original and under-sampled data, whilst twenty-five and seventeen randomly selected variables were used to build decision trees on the over-sampled and SMOTE samples, respectively. A five-fold cross-validation approach was used on all our models. We observe in Table 5.12 that the model built on the SMOTE data has a fair GINI rating whilst all the other models are rated as good as indicated in Table 4.4.

Table 5.13 illustrates the performance statistics of the RF models on the test and out-of-time validations samples. The accuracy, sensitivity and kappa performance statistics of the model built

	Dataset type	Num variables	Num trees	Model validation	Validation splits
1	Original	13	150	Cross Validation	5
2	Under-sampled	13	150	Cross Validation	5
3	Over-sampled	25	150	Cross Validation	5
4	SMOTE	17	150	Cross Validation	5

Table 5.11: Random Forest model tuning

	Dataset type	AUC	GINI	Sensitivity	Specificity
1	Original	0.826	0.652	0.669	0.810
2	Under-sampled	0.835	0.670	0.774	0.749
3	Over-sampled	0.817	0.634	0.649	0.803
4	SMOTE	0.774	0.548	0.620	0.799

Table 5.12: Random Forest model performance on training data

on the SMOTE data are the lowest when compared to the other RF models. We do not believe this model is suitable for our problem as it will lead to lost opportunities due to low sensitivity when compared to the other models.

The RF model built on the original data has the highest sensitivity compared to all the other models (NN, MARS and SVM) built on the original data. The performance statistics between the training and out-of-time validation samples is consistent, sensitivity on the out-of-time validation sample increased from 0.669 to 0.702, the kappa statistic is also the highest.

The model built on the under-sampled data has the highest sensitivity on the training sample compared to the other models, however, a specificity of 0.646 on the out-of-time validation sample is low and as a result, we expect high false rate compared to the models built on the original and over-sampled samples. The models built on the over-sampled and original data are competitive when compared against each other on sensitivity, specificity and performance on the test and out-of-time validation samples. We believe both these models are suitable for our problem, however, we choose the model built on the original data because it has the lowest miss-classified cases as seen in Table 5.14 and simplicity, we can build a model without any need to change the structure of the data through balancing the target variable.

Figure 5.3 gives us the contribution of each variable in the model, listed in order of importance, the most important predictor variable is *euribor3m* and the least predictive variable is *educationilletrate*. We indicated in Chapter 4 that seventeen variables will be used for modelling, however Figure 5.3 indicates that forty-nine variables were assessed for importance, this is because the categorical variables in our data were converted to dummy variables.

Decision trees are known to suffer from over-fitting (Farid et al., 2014), to avoid over-fitting, a subset of features are selected at random (without replacement) and used to split each node in a decision tree. The optimal number of variables used at each decision tree node split is selected to

Sample type	Dataset type	Accuracy	Sensitivity	Specificity	Kappa
Test	Original	0.771	0.657	0.806	0.419
Out-of-time validation	Original	0.744	0.702	0.779	0.482
Test	Under-sampled	0.752	0.758	0.751	0.421
Out-of-time validation	Under-sampled	0.717	0.801	0.646	0.440
Test	Over-sampled	0.763	0.658	0.795	0.405
Out-of-time validation	Over-sampled	0.723	0.715	0.729	0.443
Test	SMOTE	0.745	0.616	0.785	0.359
Out-of-time validation	SMOTE	0.581	0.729	0.459	0.181

Table 5.13: Random Forest model performance on test data

Dataset type		Actual - Yes	Actual - No
Original	**Predicted - Yes**	588	577
Original	**Predicted - No**	307	2393
Under-sampled	**Predicted - Yes**	678	741
Under-sampled	**Predicted - No**	217	2229
Over-sampled	**Predicted - Yes**	589	610
Over-sampled	**Predicted - No**	306	2360
SMOTE	**Predicted - Yes**	551	640
SMOTE	**Predicted - No**	344	2330

Table 5.14: Random forest confusion matrix - test data

be thirteen as indicated in Table 5.11. Figure 5.4 indicates that if the number of variables selected for each node split is less or greater than thirteen, the accuracy of the model reduces.

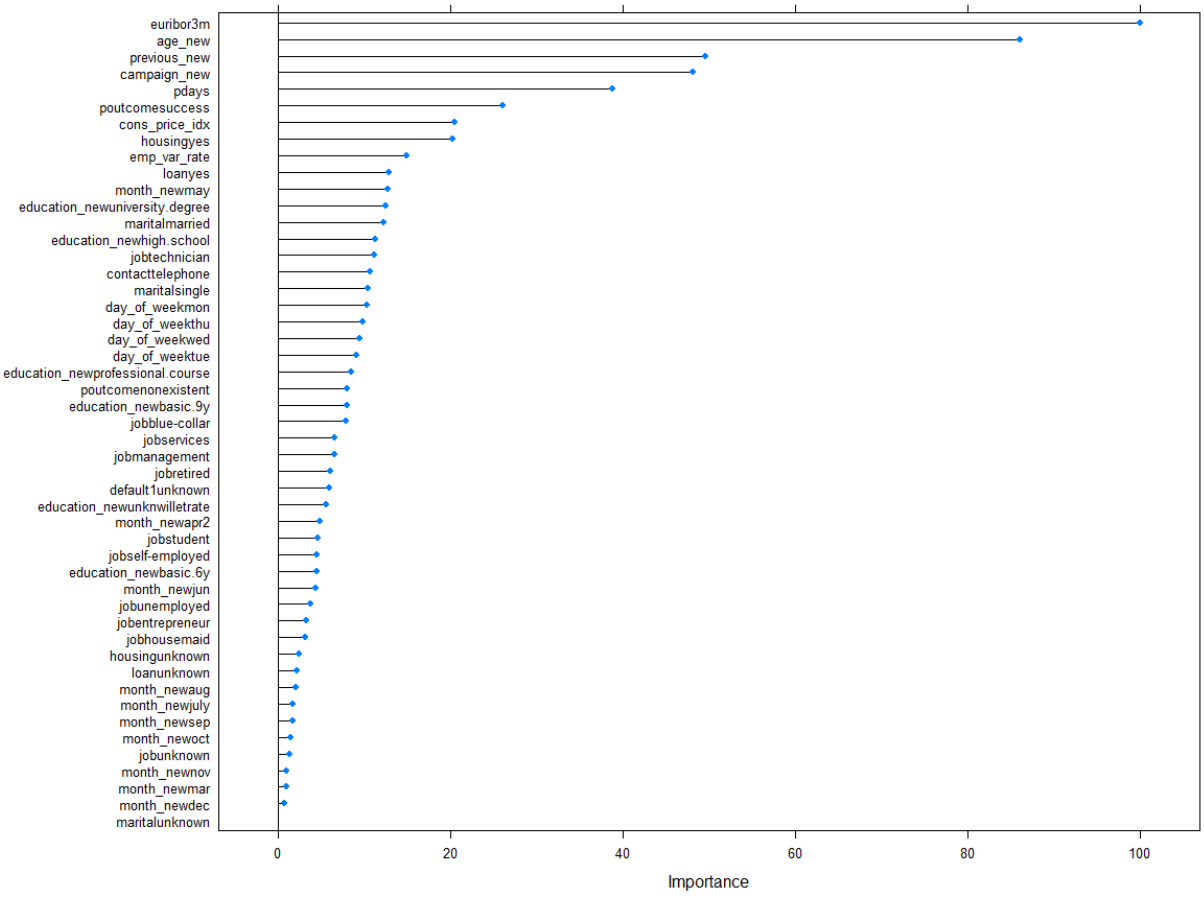

Figure 5.3: Variable Importance - Random Forest (original data)

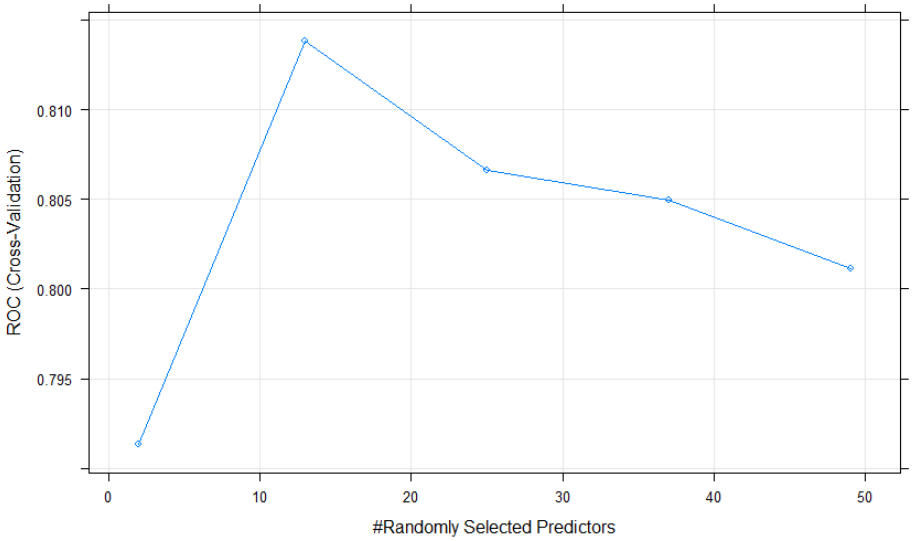

Figure 5.4: Number of variables used for node split - Random Forest (original data)

5.4 Support Vector Machine

SVM seeks to find a hyperplane that separates two or more classes. Our problem involves finding a hyperplane that separates customers that will take-up a savings product and customers that will not. Assuming that the data is linear and the classes are linearly separable, we can use a linear kernel function, equation 3.31. However, most real-world problems present data that is non-linear and we, therefore, have to consider other functions to assist in solving the problem. Linear, polynomial, RBF and sigmoid kernel functions are some of the widely used kernel functions to solve most classification problems, we, therefore, build our models on all these kernels.

We find that models built using the RBF kernel function yield the best results as outlined in Table A.2 found in the Appendix, we therefore build all our models using the RBF kernel function. To obtain the best performance out of our SVM model, we perform a grid search to find the best combination of sigma and cost parameters that yield the lowest error.

We build four models using the original, under-and over-sampled and SMOTE samples, Table 5.15 gives us an indication of the optimal tuning parameters values used to build the models.

	Dataset type	**sigma**	**Cost (C)**	**Model validation**	**Validation splits**
1	Original	0.014	0.5	Cross Validation	5
2	Under-sampled	0.013	1	Cross Validation	5
3	Over-sampled	0.014	0.5	Cross Validation	5
4	SMOTE	0.013	1	Cross Validation	5

Table 5.15: Support Vector Machine model tuning

	Dataset type	**AUC**	**GINI**	**Sensitivity**	**Specificity**
1	Original	0.766	0.532	0.388	0.970
2	Under-sampled	0.791	0.582	0.799	0.688
3	Over-sampled	0.829	0.658	0.779	0.721
4	SMOTE	0.814	0.628	0.728	0.731

Table 5.16: Support Vector Machine model performance on training data

The performance of the SVM models on the training sample is given in Table 5.16. In their study, Kim et al. (2013) found that SVM struggle to predict positive responses in instances where the data is imbalanced in favour of negative responses. The results of our SVM models confirm this finding. The accuracy and specificity of this model are similar to that of the NN model on the original data. We expect the model that will be used to identify customers that are likely to take-up a savings product to have a higher sensitivity, as a result, the model built on the original data is therefore not suitable for our problem.

All our SVM models have low kappa statistics, Table 4.6 indicates that the models with kappa values between 0.21 and 0.40 yield predictions that have a fair agreement with the development

sample distribution. We observe in Table 5.16 and Table 5.17 that all our models perform consistently on the training, test and out-of-time validation samples with an exception of specificity on the out-of-time validation sample.

Sample type	Dataset type	Accuracy	Sensitivity	Specificity	Kappa
Test	Original	0.805	0.306	0.956	0.323
Out-of-time validation	Original	0.705	0.583	0.807	0.396
Test	Under-sampled	0.710	0.789	0.686	0.369
Out-of-time validation	Under-sampled	0.672	0.735	0.619	0.348
Test	Over-sampled	0.723	0.777	0.707	0.384
Out-of-time validation	Over-sampled	0.672	0.742	0.613	0.349
Test	SMOTE	0.724	0.722	0.725	0.365
Out-of-time validation	SMOTE	0.6181	0.755	0.503	0.250

Table 5.17: Support Vector Machine model performance on test data

The model built on the over-sampled data is the most suitable SVM model for our problem, it yields a true positive rate similar to that of the model with the highest sensitivity, in addition, the confusion matrix, Table 5.18 shows that this model has the lowest error rate compared to the model built using the under-sampled data.

Dataset type		Actual - Yes	Actual - No
Original	**Predicted - Yes**	274	132
Original	**Predicted - No**	621	2838
Under-sampled	**Predicted - Yes**	706	933
Under-sampled	**Predicted - No**	189	2037
Over-sampled	**Predicted - Yes**	695	869
Over-sampled	**Predicted - No**	200	2101
SMOTE	**Predicted - Yes**	646	818
SMOTE	**Predicted - No**	249	2152

Table 5.18: SVM confusion matrix - test data

Figure 5.5 is a representation of variables used to build the SVM model on the over-sampled data, ordered by the most important to the least important variable. The variable, *campaign_new* is the most important variable and *loan* is less predictive compared to other variables used to build the model. An indication that euro interbank offered Rate plays a major role and whether a customer has a loan or not plays the least role in determining whether a customer will take-up a savings product. The model was built using a radial basis function with tuning parameters, $sigma = 0.014$ and $C = 0.5$, Figure 5.6 indicates that we obtain the best performance when these values are used together.

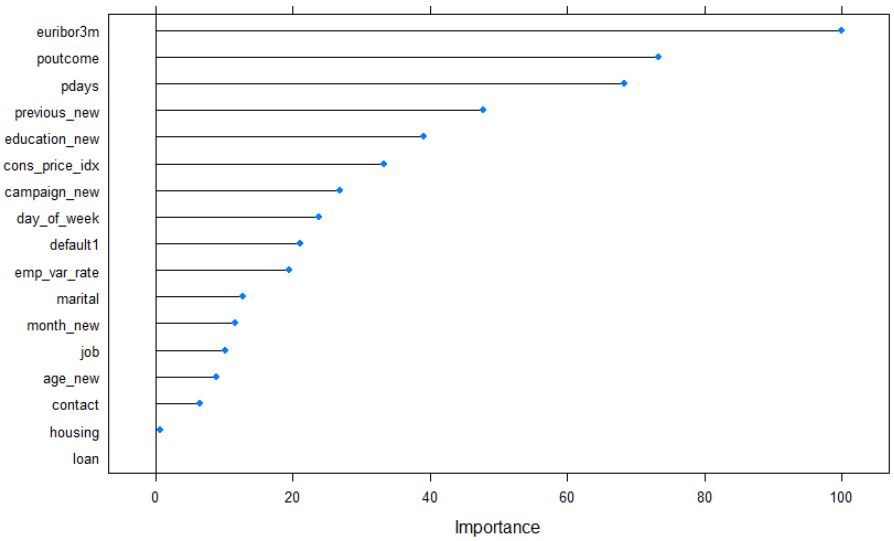

Figure 5.5: Variable importance - (over-sampled data)

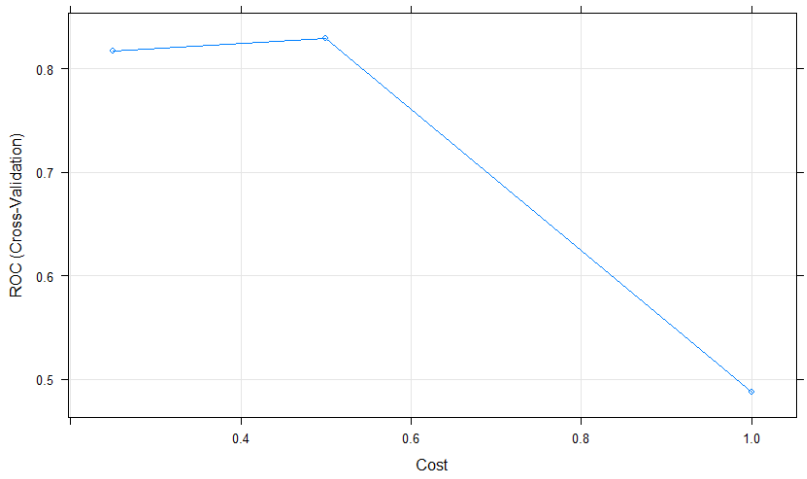

Figure 5.6: Cost (C) factor - (over-sampled data)

5.5 Summary

We started with imbalanced data drawn from the results of a savings product campaign of a Portuguese bank. Due to the imbalance in the data, three sampling techniques, namely, under-sampling, over-sampling and SMOTE were used to balance the data, resulting in three additional data samples. SVM, RF, NN and MARS models were built on each data sample resulting in a total of sixteen predictive models.

One of each SVM, RF, NN and MARS model built on the four types of data samples is selected if found to yield a balance between model accuracy, sensitivity, specificity, GINI and kappa statistics to ensure a higher number of customers that are likely to take-up a savings product are identified. We found that models built on the imbalanced data have a low sensitivity and therefore struggled

to identify customers that are likely to take-up a savings product, this can result in lost opportunity for a business. Balancing the data through the use of sampling techniques mentioned above resulted in models with higher sensitivity.

	Model	**Treatment on data**
1	Multivariate Adaptive Regression Splines	Over-sampling
2	Random Forest	Original
3	Neural Network	Under-sampling
4	Support Vector Machine	Over-sampling

Table 5.19: Selected models

As indicated in Table 5.19 applying over-sampling to the original data results in two of the four selected models, none of the models built on the SMOTE data were chosen. In addition to using the suitable data to build a model, we found that model tuning assisted in improving model accuracy.

Tuning a MARS model involves choosing the number of terms in a model, when the pruning parameter equals one, the model is made up of only an intercept whilst a higher value results in a model made up of multiple terms. A pruning parameter, *Nprune = 20* was chosen as the optimal value for our model, resulting in a model made up of an intercept, twelve hinge functions and seven features. The degree tuning parameter in a MARS model determines whether a product of hinge functions is allowable or not, in our case, a product of hinge functions did not substantially improve model performance and we, therefore, opted for *degree = 1* for all our MARS models.

As part of building NN models, an activation function suitable for the problem is chosen, we chose a hyperbolic tangent activation function for our NN models. Tuning an NN model involves selecting the decay and number of hidden nodes in a model; a decay parameter penalises the weights of an NN model resulting in a less complex model whilst a higher number of hidden nodes indicate a complex model and vice versa. Our chosen NN model was built using *decay = 0.5* and *number of hidden nodes = 1*.

Tuning an RF involves determining the number of trees that will be used in the forest and the number of randomly selected variables used to split each node of a decision tree. The number of randomly selected variables used for each node split is selected to be thirteen, and hundred and fifty decision trees were built for each RF model.

Our chosen SVM model was built using the radial basis function kernel and tuning parameters, *sigma = 0.014* and *Cost = 0.5*, these yielded the best results for our model.

All the chosen models agree that *euribor3m* is the most predictive attribute of customers that are likely to take-up a savings product except in the case of the NN model which rates *previous_new* as the most predictive variable. We have also found that *previous campaign outcome*, *education*

level, *customer age*, *consumer price index*, *contact channel* e.g. telephone, cellphone, etc. and the *month a customer is contacted* and *day of the week customer is contacted* are important features.

Chapter 6

Discussion

The objective of this chapter is to choose a model from the four models we selected in the previous chapter. We use gains and lift charts to assist us in selecting a model that will lead to the lowest number of customers contacted to yield the desired response rate. In addition, kappa statistics of each model are considered to assist in selecting the best model for our problem. Although the chosen model should have a high sensitivity, we also expect a high specificity, this will result in a model that yields lower misclassified cases and therefore deliver efficiency in the leads management processes.

6.1 Model Comparison

We showed in the previous chapter that three out of the four selected models were built on under and over-sampled data, the RF model is the only model built on the original data. One of the challenges that campaign managers face is deciding how many customers to contact to yield the desired response rate. Our models are compared against each other for their ability to identify the lowest possible number of customers to contact to yield the desired response rate.

Olson and Chae (2012) and Kim and Street (2004) used a cumulative gains chart to obtain a balance between customer gain and prediction accuracy. A common example of using gains chart is covered in the research by Banslaben and Nash (1992) who used a gains chart to select the optimal number of individuals to send a direct marketing mail to achieve the desired response rate.

Creating a gains chart involves using a predictive model to score a list of customers previously contacted for a campaign, the scored probability is sorted in descending order and segmented into deciles (ten deciles in our case) ranked in order of the most to fewest customers having a propensity to respond positively to a campaign. The cumulative ratio of actual campaign take-up obtained from previous campaigns is computed, the computed ratio will reach 100% at some point, usually on the last decile. The business decides the number of leads they would like to contact and the desired response rate, a gains chart is then used to decide how many customers must be contacted to achieve the desired results. We use gains charts to compare our models and to gain an understanding of which model will yield the best balance between customer gain and prediction accuracy.

The gains chart of the MARS model is presented in Figure 6.1, we find that if we contact 10% of the scored individuals ranked by probability to take-up a savings product offer, the model will

assist us to identify 31.06% of the customers that will respond positively to a savings product campaign. If we contact 50% of the scored individual, the model will assist us to gain 88.26% of the customers that will respond positively to a campaign. This suggests that if a campaign manager is given a total of 1,000 individuals, for example, and they will like to know the number of customers to contact to obtain the desired response rate. What we observe from the gain chart is that by contacting 500 of the available leads, 88.26% of the customers that will respond positively to a campaign will be among the first 500 customers contacted, a decision can then be made on how much more the business is willing to spend on costs related to contacting individuals to obtain the remaining 11.74% of respondents. The chart also suggests that only 90% of customers should be contacted to yield 100% of customers that are likely to take-up.

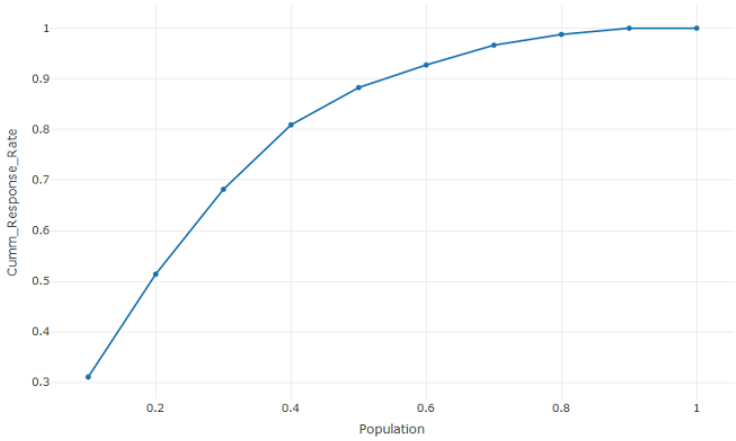

Figure 6.1: Gains chart - MARS model

Figure 6.2 is the gains chart of the NN model, it suggests that all customers must be contacted to yield 100% of customers that are likely to respond to a savings product campaign. Suppose 50% of the scored individual are contacted, the model will assist us to gain 83.13% of the customers that will respond positively to a campaign. The MARS model requires the lowest number of customers to contact to return a higher number of individuals that will respond positively to a campaign, this is ideal because it will result in lowering operational costs where instead of contacting all leads, only a subset of individuals are contacted to yield the desired response rate.

Figure 6.3 is a gains chart of the SVM model, whilst it yields higher response rate at lower deciles than the NN model, given that contacting 50% of individuals will return 84.24% of customers that respond positively to a campaign, it lags behind the MARS model. The SVM probability rank also requires that all customers are contacted to obtain 100% of the customers that are likely to respond positively.

The gains chart of the RF model is given by Figure 6.4, contacting 10% of the customers' yield 30.95% of the individuals that will respond positively to an offer and by contacting 50% of the individuals' 85.59% of the customers that will respond positively to an offer are identified. The RF model suggests that all leads must be contacted to yield 100% of the positive respondents.

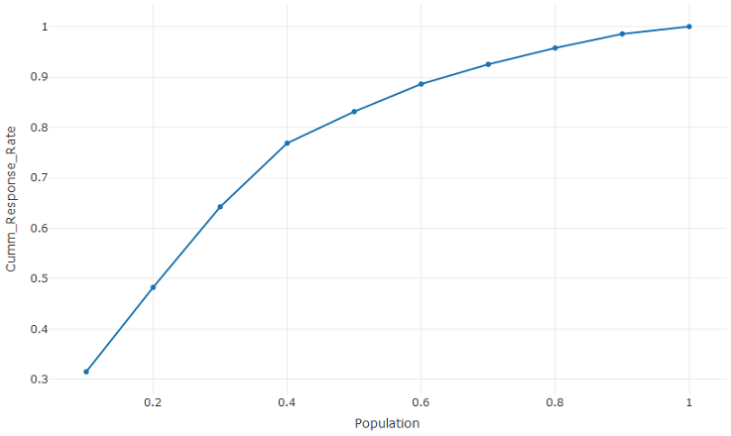

Figure 6.2: Gains chart - Neural network model

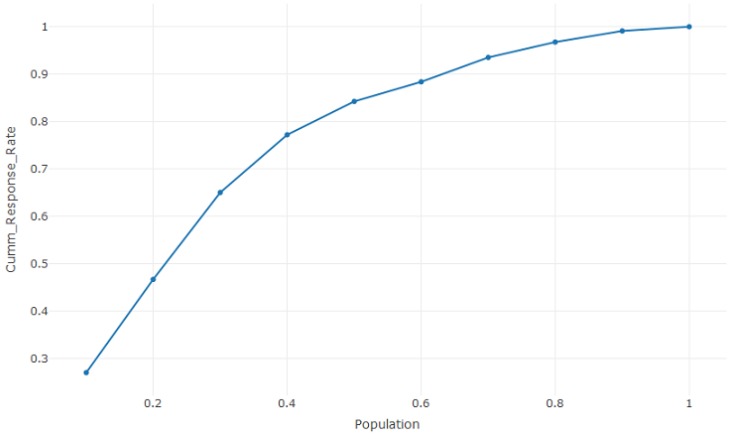

Figure 6.3: Gains chart - SVM model

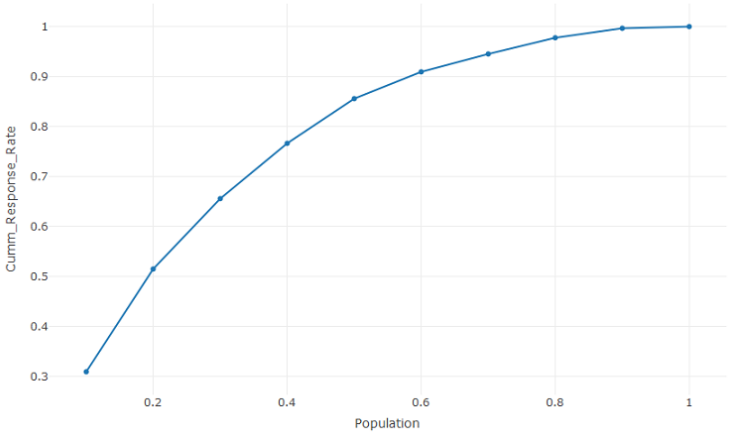

Figure 6.4: Gains chart - RF model

Zacharis (2016) and Shiny et al. (2015) indicated that a lift chart can be used to decide whether our chosen model is any better than randomly selecting leads for a campaign. Table 6.1 indicates that the NN model yields 3.17 higher savings product take-up than selecting leads randomly, the highest compared to the other models. However if our required number of leads is higher than 10% of the population, MARS has a higher lift than all the other models. The scope of the population that is considered for leads selection is usually higher than 10%, we, therefore, conclude that the MARS model yields higher lift than all the other models.

	MARS	NN	RF	SVM
Population	Lift	Lift	Lift	Lift
10%	3.10	3.17	3.08	2.70
20%	2.57	2.43	2.53	2.34
30%	2.27	2.14	2.24	2.17
40%	2.02	1.90	1.95	1.93
50%	1.76	1.63	1.72	1.68
60%	1.55	1.47	1.52	1.47
70%	1.38	1.32	1.36	1.34
80%	1.23	1.20	1.22	1.21
90%	1.11	1.09	1.10	1.10
100%	1.00	1.00	1.00	1.00

Table 6.1: Models lift

A Youden index is a function of sensitivity and specificity. It is used to decide the optimal cut-off point of a model (Bozikov and Lijana, 2010). The Youden index calculates the maximum vertical distance between the 45° line (line of equality) and the Lorenz curve to find an optimal sensitivity and specificity of a model. Of course, this is not always the objective of a modelling exercise, there are cases for example where the objective is to obtain a trade-off between sensitivity and specificity, and there are instances, for example, where we require a higher sensitivity than specificity. However, in this instance, we use the Youden index to find out which of our models perform better when the index is used. The index is given by the function:

$$J = \max((\text{sensitivity} + \text{specificity}) - 1). \tag{6.1}$$

For each ROC curve obtained for our models, we calculate the Youden index to determine cut-off points to apply on a hypothetical example to assist us to establish which model yields the lowest misclassification rate. Suppose we employ our predictive models on 1,000 leads (assuming a 23.61% response rate), using the Youden index to obtain optimal cut-off points, we obtain the results in Table 6.2.

The RF model yields the best sensitivity and lowest specificity, whilst the other three models have similar sensitivity. A model that yields a balance between sensitivity and specificity is desirable to ensure we identify a higher number of individuals that will respond positively to an offer whilst minimising false positives. The misclassification of cases indicate either a lost opportunity due to not contacting customers that will usually respond positively to an offer or wasted resources by contacting customers that will not respond positively to an offer. The RF model yields 8.98%

	Model	Sensitivity	Specificity	True positive	True negative	misclassified
1	MARS	0.713	0.823	168	629	203
2	RF	0.777	0.707	183	540	276
3	NN	0.696	0.781	164	597	239
4	SVM	0.709	0.777	167	594	239

Table 6.2: Results of using Youden index to select model cut-off

higher true positive rate and 36.21% higher misclassified cases than the MARS model. Although the RF model will identify a higher number of customers that will respond positively to a campaign, the higher misclassification of the RF model leads us to conclude that the MARS model performed better than all the models in this instance due to its low misclassification rate and a competitive true positive rate compared to the NN, RF and SVM models. The ROC curve of the MARS model is given by Figure A.5 found in the Appendix, and as indicated in Table 6.2, the sensitivity and specificity statistics are given by 0.713 and 0.823, respectively.

A kappa statistic is used to compare the outcome of a predictive model and the original population distribution to gain an understanding of how much agreement there is between the two distributions. The kappa statistic of our models on the test sample is given in Table 6.3. The MARS model yields the best kappa statistic of the four models, Landis and Koch (1977) proposed Table 4.6 which indicates that the outcome of the MARS model has a moderate agreement with the original population distribution, whilst the benchmark proposed by Fleiss (1981) indicates an intermediate to good model. We are satisfied that the MARS model has a potential to achieve its main objective of identifying customers that are likely to take-up a savings product.

	Model	kappa
1	MARS	0.433
2	RF	0.419
3	NN	0.388
4	SVM	0.384

Table 6.3: Kappa statistics of our models

Of the four machine learning techniques, we considered, MARS is the only model that can be expressed in a form that can be easily understood, the target has a linear relationship with a combination of intercept, independent variables and basis functions formed from independent variables. The other three models are referred to as a black-box, once created, it is not possible to visualise how the final model looks. However, a MARS model is formed as a summation of the terms given in Table 5.6 with each term multiplied by their respective coefficients to result in a model of the form:

$$f(X) = \beta_0 + \sum_{i=1}^{M} \beta_i B_i(X) \tag{6.2}$$

where $X = (x_1, x_2, ..., x_k)$ a vector of k inputs, x_k is an independent variable for some arbitrary k and $B_i(X)$ is a basis function, $\forall i \in \{1, 2, ..., M\}$. The coefficients β_i are jointly adjusted to give the

best fit to the data (Friedman, 1991). MARS is the fastest model to train as shown in Figure A.13 found in the Appendix and the easiest to interpret compared to the other models we considered.

6.2 Model Choice

Our chosen model is one that can be deployed in an operational environment within a bank to yield a balance between sensitivity and specificity, highlight insights on attributes that drive savings product take-up and reduce operational costs by identifying the lowest possible number of customers to contact to yield the desired response rate. We have highlighted attributes that drive savings product take-up in Chapter 5 for each of our models.

Some of the attributes that have been identified to drive take-up of a savings product include:

- *euribor3m*
- *previous_new*
- *poutcome*
- *month_new*
- *age*
- *contact*
- *emp_var_rate*
- *default*

We have found that the movement of the euro interbank interest rate has an impact on take-up rate, previous interaction and previous outcome of a campaign also has an impact, particularly if the interaction yielded a successful campaign outcome. The month that we call a customer with an offer, the channel used to contact a customer and whether a customer has defaulted on their credit commitments play a role in predicting savings product take-up. Customer age and a country's workforce employment rate also play a role.

We have considered a variety of techniques to decide on the model to choose, the MARS model has consistently performed competitively when compared to the NN, RF and SVM models. We have demonstrated through gains and lift charts that the MARS model will yield the highest number of responses and perform better than selecting leads randomly.

Through the use of the Youden's index to select an optimal sensitivity and specificity of our model, we found that the MARS model yields the least misclassified cases. The kappa statistic of the MARS model has also been found to be the highest compared to all the other models, an indication that the predicted outcome of the model is the closest to the original population distribution. We believe that MARS (with over-sampling) is the most suitable model for predicting customers that are likely to take-up a savings product, MARS is, therefore, our chosen model.

The variables of the MARS model with over-sampling can be summarised as follows:

- A lower Euribor rate increases the propensity to take-up.
- A successful outcome of a previous campaign increases the propensity to take-up.
- Campaigns conducted in the month of May are less likely to be successful.
- Contacting customers between Tuesday and Thursday in April increases the likelihood of take-up.
- Older customers are more likely to take-up a savings product.
- Customers contacted via a landline telephone are less likely to take-up.
- Customers contacted in June and July are more likely to take-up.
- Customers that are in arrears on their credit commitments are less likely to take-up.

Chapter 7

Conclusion

This chapter highlights what we have learned throughout our study and covers further work that can be undertaken to enhance marketing analytics through the use of data mining techniques.

7.1 Conclusion

We set out to build a predictive model that can be deployed in a bank to identify customers that are likely to take-up a long-term savings product. As part of the model building process, we identified factors that have an influence in driving propensity to respond positively to a campaign. Communicating insights about these factors to the bank is important to assist in increasing operational efficiency related to new acquisitions, cross-sell and up-sell opportunities.

Through the studies conducted by others, we have learned of techniques that are commonly used to prepare data for model building. We found that building predictive models on imbalanced data results in models that have a low sensitivity and high specificity, the challenge presented by this is that these models struggle to identify positive respondents to a campaign. To counter against low sensitivity in predictive models, our data was balanced through the use of the following sampling techniques; under-and over-sampling and SMOTE. We built sixteen models using the following techniques; random forest, support vector machine, multivariate adaptive regression splines and neural network on imbalanced and balanced data due to the sampling techniques mentioned above.

Our study supports the finding by (Claesen and Moor, 2015) that hyper-parameter tuning is an important aspect of the model building process. Our models were trained on a different combination of hyper-parameters selected through a grid search, we found that model performance varied across different hyper-parameter values.

The most common metrics used for measuring performance of classification models covered in literature include the receiver operating characteristics curve, confusion matrix, GINI, kappa, sensitivity, specificity, and lift and gains charts. We considered all these metrics to assess the performance and robustness of our models, and we found that given a set of models, the best model will not necessarily yield the best performance across all these metrics. It was, therefore, important to be clear about the objective of the model to ensure that the chosen model yields an optimal solution for our problem. Sensitivity and specificity proved to be important performance metrics for our problem. An optimal balance between these two metrics ensures that we have a model that

can identify customers that are likely to take-up a savings product whilst reducing the false positive rate. The gains chart was found to be instrumental in identifying model performance across different segments of the population.

We caution other researchers against using the feature, *Call duration* to predict take-up on the data provided by (Moro et al., 2014). This feature is only available after campaign leads have been selected, as a result, it is not useful for the initial leads selection process. We highlighted in our study that this feature can be used, for example, to improve operational efficiency, a study can be done to identify areas where *call duration* can be reduced by introducing other channels to finalise the sale process.

We conclude that MARS is the best model to deploy in a bank to identify the lowest number of customers to contact to yield optimal response rate for a long-term savings product campaign. Compared to the other models we considered for our study, MARS is the easiest model to interpret and the fastest to train.

7.2 Future Work

We found that model tuning is an important and integral part of the model building process, as a result, there is an opportunity to improving model performance through extending the parameter grid used in model building to consider a higher number of permutations when training predictive models. Of course, this comes at a price because a computer will take longer to build models. However, we believe it is still important in spite of this shortcoming.

The time-series in Figure 4.2 demonstrates that take-up rate changes over time, particularly when the economy is on an upward or downward trajectory. We support the methodology applied by Moro et al. (2014) where they developed models on a rolling window period to compensate for changes in micro and macroeconomic factors. MARS was not one of the models that Moro et al. (2014) applied this method to, we believe there is merit to building MARS models on rolling window period to improve predictive accuracy.

Another predictive modelling technique to consider for future research is eXtreme Gradient Boosting (XGBoost), a gradient boosting tree technique introduced by Friedman et al. (2000). Chen and Guestrin (2016) indicated that seventeen (17) out of twenty-nine (29) solutions that were conducted on Kaggle used XGBoost. Kaggle is a platform that different businesses submit challenging problems and data related to the problems on the platform, the problems are of nature that can be solved using supervised learning techniques. Data scientists compete to solve these problems using a variety of machine learning techniques.

References

Affes, Z. and Hentati-Kaffel, R. (2016). Forecast bankruptcy using a blend of clustering and mars model - case of us banks. *Laboratory of Excellence on Financial Regulation*, pages 1–36.

Aguinis, H., Gottfredson, R., and Joo, H. (2013). Best-practice recommendations for defining, identifying, and handling outliers. *Organizational Research Methods*, 16(2):270–301.

Alhakbani, H. and al Rifaie, M. (2016). Handling class imbalance in direct marketing dataset using a hybrid data and algorithmic level solutions. In *SAI Computing Conference, 2016*, page 446–451.

Ansari, A., Mela, C., and Neslin, S. (2008). Customer channel migration. *Journal of Marketing Research*, 45(1):60–76.

Auria, L. and Moro, R. (2008). Support vector machines (svm) as a technique for solvency analysis. *SSRN Electronic Journal*, 1(1):1–16.

Bahari, F. and Elayidom, S. (2015). An efficient crm-data mining framework for the prediction of customer behaviour. *Procedia Computer Science*, 46(1):725–731.

Bahnsen, A., Aouada, D., and Ottersten, B. (2015). Example-dependent cost-sensitive decision trees. *arXiv preprint*, 42(19):6609–6619.

Banslaben, J. and Nash, E. (1992). *The direct marketing handbook*. McGraw-Hill, New York.

Bekkar, M., Djemaa, H., and Alitouche, T. (2013). Evaluation measures for models assessment over imbalanced data sets. *Journal of Information Engineering and Applications*, 3(10):2224–5782.

Bengio, Y. and LeCun, Y. (2007). Scaling learning algorithms towards ai. *Large-Scale Kernel Machines*, 34(5):1–41.

Berg, D. (2007). Bankruptcy prediction by generalized additive models. *Applied Stochastic Models in Business and Industry*, 23(2):129–143.

Bower, J. and Christensen, C. M. (1995). Disruptive technologies: Catching the wave. *Harvard Business Review*, 73(1):43–53.

Bozikov, J. and Lijana, Z. (2010). *Test Validity Measures and Receiver Operating Characteristic (ROC) Analysis*. Hans Jacobs Publishing Company.

Bramer, M. (2013). *Avoiding Overfitting of Decision Trees*, pages 121–136. Springer London, London.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

Breiman, L., Chen, C., and Liaw, A. (2004). Using random forest to learn imbalanced data. *Department of Statistics, UC Berkeley*, pages 1–12.

Canals-Cerda, J. and Kerr, S. (2015). Forecasting credit card portfolio losses in the great recession: A study in model risk. *Journal of Credit Risk*, 11(1):29–57.

Chen, J., Y.T., H.W., and Chen, T. (2015). Big data based fraud risk management at alibaba. *The Journal of Finance and Data Science*, 1(1):1–10.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. *CoRR*, abs/1603.02754:785–794.

Chen, X. and Ishwaran, H. (2012). Random forests for genomic data analysis. *Brief Bioinform*, 99(6):323–329.

Chitra, K. and Subashini, B. (2013). Data mining techniques and its applications in banking sector. *International Journal of Emerging Technology and Advanced Engineering*, 3(8):219–226.

Claesen, M. and Moor, B. D. (2015). Hyperparameter search in machine learning. *CoRR*, abs/1502.02127:1–5.

Cohen, J. (1968). Weighted kappa: nominal scale agreement provision for scaled disagreement or partial credit. *Psychological Bulletin*, 70(4):213–220.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3):273–297.

Craven, P. and Wahba, G. (1979). Smoothing noisy data with spline functions. *Numerische Mathematik*, 31(4):377–403.

Diapouli, M., Kapetanakis, S., Petridis, M., and Evans, R. (2017). Behavioural analytics using process mining in on-line advertising. In *ICCBR*, pages 147–156.

Dobek, A., Moliński, K., and Skotarczak, E. (2015). Power comparison of raoś score test, the wald test and the likelihood ratio test in (2xc) contingency tables. *Biometrical Letters*, 52(2):95–104.

Dormann, C., Elith, J., Bacher, S., Buchmann, C., Carl, G., Carré, G., Marquéz, García, J., Gruber, B., Lafourcade, B., ao, P. L., Münkemüller, T., McClean, C., Osborne, P., Reineking, B., Schröder, B., Skidmore, A., Zurell, D., and Lautenbach, S. (2013). Collinearity: a review of methods to deal with it and a simulation study evaluating their performance. *Ecography*, 36(1):27–46.

Elsalamony, H. (2014). Bank direct marketing analysis of data mining techniques. *International Journal of Computer Applications*, 85(7):12–27.

Farid, D., Zhang, L., Rahman, C., Hossain, M., and Strachan, R. (2014). Hybrid decision tree and naive bayes classifiers for multi-class classification tasks. *Expert Systems with Applications*, 41(4):1937–1946.

Feng, G., Zhang, J., and Liao, S. (2014). A novel method for combining bayesian networks, theoretical analysis, and its applications. *Pattern Recognition*, 47(54):2057–2069.

Fleiss, J. (1981). *Statistical Methods for Rates and Proportions. Second Edition*. Wiley, John and Sons, Incorporated, New York, N.Y.

Franke., G. (2010). *Multicollinearity*. John Wiley & Sons, Ltd.

Friedman, J. (1991). Multivariate adaptive regression splines. *The Annals of Statistics*, 19(1):1–67.

Friedman, J., Hastie, T., and Tibshirani, R. (2000). Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *Ann. Statist.*, 28(2):337–407.

Gregorutti, B., Michel, B., and Saint-Pierre, P. (2013). Correlation and variable importance in random forests. *ArXiv e-prints*, pages 1–31.

Grzonka, D., Suchacka, G., and Borowik, B. (2016). Application of selected supervised classification methods to bank marketing campaign. *Information Systems in Management*, 5(1):36–48.

Guo, Y., Yin, G., Li, M., Ren, X., and Liu, P. (2018). Mobile e-commerce recommendation system based on multi-source information fusion for sustainable e-business. *Sustainability, Open Access Journal*, 10(147):1–13.

Hadden, J., Tiwari, A., Roy, R., and Rutab, D. (2007). Computer assisted customer churn management: State-of-the-art and future trends. *Computers & Operations Research*, 34(10):2902–2917.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The elements of statistical learning*, page 745. Springer New York, New York.

Hecht-Nielsen, R. (1992). *Neural Networks for Perception (Vol. 2)*. Harcourt Brace & Co.

Ishak, N. and Bakar, A. (2014). Developing sampling frame for case study: Challenges and conditions. *World Journal of Education*, 4(3):29–35.

Kartasheva, A. and Traskin, M. (2013). Insurers' insolvency prediction using random forest classification. In *International Association of Insurance Supervisors*, pages 1–45.

Keles, A. and Keles, A. (2015). Ibmms decision support tool for management of bank telemarketing campaign. *International Journal of Database Management Systems*, 7(5):1–15.

Kim, G., Chae, B., and Olson, D. (2013). A support vector machine (svm) approach to imbalanced datasets of customer responses: comparison with other customer response models. *Service Business*, 7(1):167–182.

Kim, M.-J., Kang, D.-K., and Kim, H. B. (2015). Geometric mean based boosting algorithm with over-sampling to resolve data imbalance problem for bankruptcy prediction. *Expert Systems with Applications*, 42(3):1074–1082.

Kim, Y. and Street, W. (2004). An intelligent system for customer targeting: A data mining approach. *Decis. Support Syst.*, 37(2):215–228.

Kyngäs, H. and Rissanen, M. (2001). Support as a crucial predictor of good compliance of adolescents with a chronic disease. *Journal of Clinical Nursing*, 10(2):767–774.

Landis, J. and Koch, G. (1977). The measurement of observer agreement for categorical data. *International Biometric Society*, 33(1):159–174.

Lee, T., Chiu, C., Chou, Y., and Lu, C. (2006). Mining the customer credit using classification and regression tree and multivariate adaptive regression splines. *Computational Statistics & Data Analysis*, 50(4):1113–1130.

Lewis, J. and Ling, P. (2016). "gone are the days of mass-media marketing plans and short term customer relationships": Tobacco industry direct mail and database marketing strategies. *British Medical Association*, 25(4):430–436.

Ling, C. and Li, C. (1998). Data mining for direct marketing: Problems and solutions. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, KDD'98, pages 73–79.

Ling, C., Ling, C., and Li, C. (1998). Data mining for direct marketing: Problems and solutions. In *In Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98*, pages 73–79. AAAI Press.

Martin, M. and Roberts, S. (2010). Jackknife-after-bootstrap regression influence diagnostics. *Journal of Nonparametric Statistics*, 22(2):257–269.

Midi, H., Sarkar, S., and Rana, S. (2013). Collinearity diagnostics of binary logistic regression model. *Journal of Interdisciplinary Mathematics*, 13(3):253–267.

Miguéis, V., Camanho, A., and Cunha, J. (2013). Customer attrition in retailing: An application of multivariate adaptive regression splines. *Expert Systems with Applications*, 40(16):6225–6232.

Moro, S., Cortez, P., and Rita, P. (2012). Enhancing bank direct marketing through data mining. *In Proceedings of the Fourty-First International Conference of the European Marketing Academy*, 62:22–31.

Moro, S., Cortez, P., and Rita, P. (2014). A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems*, 62(1):22–31.

Muzir, E. (2013). Impact of placement choices and governance issues on credit risk in banking: Nonparametric evidence from an emerging market. *Journal of Knowledge Management,*

Economics and Information Technology, 3(3):1–56.

Ngai, E., Xiu, L., and Chau, D. (2009). Application of data mining techniques in customer relationship management: A literature review and classification. *Expert Systems with Applications*, 36(2):2592–2602.

Olgac, V. and Karlik, B. (2011). Performance analysis of various activation functions in generalized mlp architectures of neural networks. *International Journal of Artificial Intelligence And Expert Systems*, 1(4):111–122.

Olson, D. and Chae, B. (2012). Direct marketing decision support through predictive customer response modelling. *Decision Support Systems*, 54(1):443–451.

Oshiro, T., Perez, P., and Baranauskas, J. (2012). How many trees in a random forest? In Perner, P., editor, *Machine Learning and Data Mining in Pattern Recognition*, pages 154–168.

Ricci, F., Rokach, L., and Shapira, B. (2011). *Introduction to Recommender Systems Handbook*, pages 1–35. Springer US, Boston, MA.

Rodriguez-Galiano, V., Sanchez-Castillo, M., Chica-Olmo, M., and Chica-Rivas, M. (2015). Machine learning predictive models for mineral prospectivity: An evaluation of neural networks, random forest, regression trees and support vector machines. *Ore Geology Reviews*, 71(1):804–818.

Rutkowski, L., Jaworski, M., Pietruczuk, L., and Duda, P. (2014). The cart decision tree for mining data streams. *Information Sciences*, 266:1–15.

Schultebraucks, L. (2017). Introduction to support vector machines. `https://medium.com/@LSchultebraucks/introduction-to-support-vector-machines-9f8161ae2fcb`.

Sharma1, A. and Chopra, A. (2013). Artificial neural networks: Applications in management. *Journal of Business and Management*, 12(5):32–40.

Shiny, K., Swaminathan, M., Kumar, N., and Thiyagarajan, L. (2015). Implementation of data mining algorithm to analysis breast cancer. *International Journal for Innovative Research in Science & Technology*, 1(9):207–212.

Sindhu, M. and Vijaya, M. (2015). Predicting churners in telecommunication using variants of support vector machine. *American Journal of Engineering Research*, 4(3):11–18.

Singoei, L. and Wang, J. (2013). Data mining framework for direct marketing: A case study of bank marketing. *International Journal of Computer Science*, 10(2):198–203.

Song, Y. and Lu, Y. (2015). Decision tree methods: applications for classification and prediction. *Shanghai Arch Psychiatry*, 27(2):130–135.

Stansbury, D. (2017). A gentle introduction to artificial neural networks. `https://theclevermachine.files.wordpress.com/2014/09/neural-net1.png`.

Stone, M. and Woodcock, N. (2014). Interactive, direct and digital marketing: A future that depends on better use of business intelligence. *Journal of Research in Interactive Marketing*, 8(1):4–17.

Vaidehi, R. (2016). Predictive modeling to improve success rate of bank direct marketing campaign. *IJMBS*, 6(1):22–24.

Vajiramedhin, C. and Suebsing, A. (2014). Feature selection with data balancing for prediction of bank telemarketing. *Applied Mathematical Sciences*, 8(114):5667–5672.

Veaux, R. D. and Ungar, L. (1994). Multicollinearity: A tale of two nonparametric regressions.

Wisaeng, K. (2013). A comparison of different classification techniques for bank direct marketing. *International Journal of Soft Computing and Engineering*, 3(4):2231–2307.

Witten, I. and Frank, E. (2005). *Data Mining Practical Machine Learning Tools and Techniques, Second Edition*. Elsevier., 500 Sansome Street, Suite 400, San Francisco, CA 94111.

Xia, G. and Jin, W. (2008). Model of customer churn prediction on support vector machine. *Systems Engineering - Theory & Practice*, 28(1):71–77.

Yang, Y. and Huang, S. (2014). Suitability of five cross validation methods for performance evaluation of nonlinear mixed-effects forest models – a case study. *Forestry (Lond)*, 87(5):654–662.

Yap, B., Rani, K., Rahman, H., Fong, S., Khairudin, Z., and Abdullah, N. (2014). *An Application of Oversampling, Undersampling, Bagging and Boosting in Handling Imbalanced Datasets*, pages 13–22. Springer Singapore, Singapore.

Zacharis, Z. (2016). Predicting student academic performance in blended learning using artificial neural networks. *International Journal of Artificial Intelligence and Applications*, 7(5):17–29.

Zakirov, D. and Momtselidze, N. (2015). Application of data mining in the banking sector. *Journal of Technical Science and Technologies*, 4(1):13–16.

Zhang, W. and Goh, A. (2016). Multivariate adaptive regression splines and neural network models for prediction of pile drivability. *Geoscience Frontiers*, 7(1):45–52.

Zhang, X., Zhou, Y., Ma, Y., Chen, B., Zhang, L., and Agarwal, D. (2016). Glmix: Generalized linear mixed models for large-scale response prediction. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 363–372.

Zien, A., Rätsch, G., Mika, S., Schölkopf, B., Lemmen, C., Smola, A., Lengauer, T., and Müller, K. (2017). Engineering support vector machine kernels that recognize translation initiation sites. `http://www.bioinfo.de/isb/gcb99/talks/zien/fig2.gif`.

Appendix A

Appendix

Building a predictive model on the data balanced through over-sampling, under-sampling or SMOTE results in predicted probabilities that are different from the original sample, and applying the model to the test data or new cases results in a shift in known probability to take-up a savings product, function A.1 is used to re-calibrate the scored probabilities to shift towards the known distribution.

$$CALIBP = 1/(1 + ((1/IMBALRAT) - 1)/((1/BALRAT) - 1) * [(1/prob) - 1]) \tag{A.1}$$

where *IMBALRAT* is the ratio of positive response before balancing the data, in our case, this ratio is 0.236, *BALRAT* is the ratio of the balanced data, in the case of the under-sampled data, for example, this value is 0.500 and *prob* is the probability produced by the model built on the balanced data.

Figures A.1, A.2 and A.3 illustrate the probability distribution on the un-balanced test data as scored by models built on the balanced data. The left box plot represents the scores as scored by the models and the box-plot on the right represents calibrated scores. The random forest model scores distribution is not calibrated since the model is built on the original data-set, figure A.4 is a representation of the probability distribution.

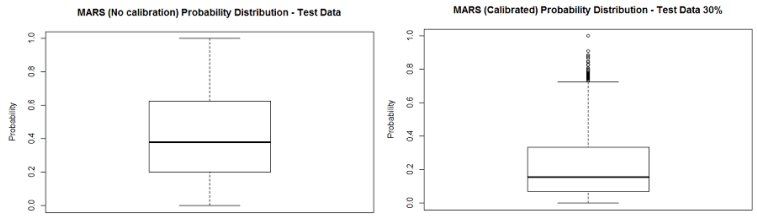

Figure A.1: MARS - Test data - Model probability distribution non-calibrated and calibrated

Figure A.2: SVM - Test data - Model probability distribution non-calibrated and calibrated

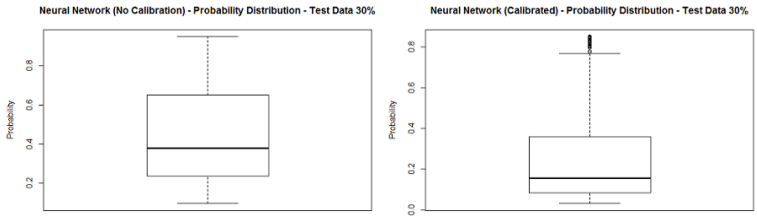

Figure A.3: NN - Test data - Model probability distribution non-calibrated and calibrated

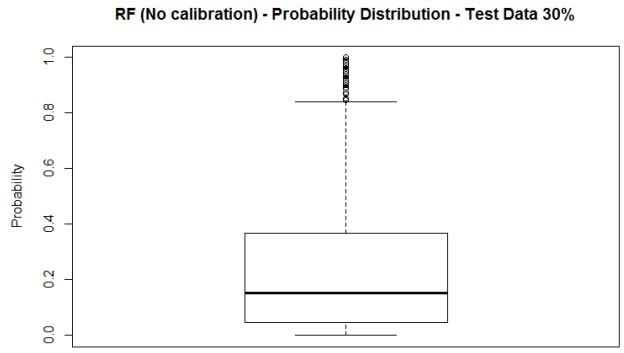

Figure A.4: RF - Test data - Model probability distribution non-calibrated

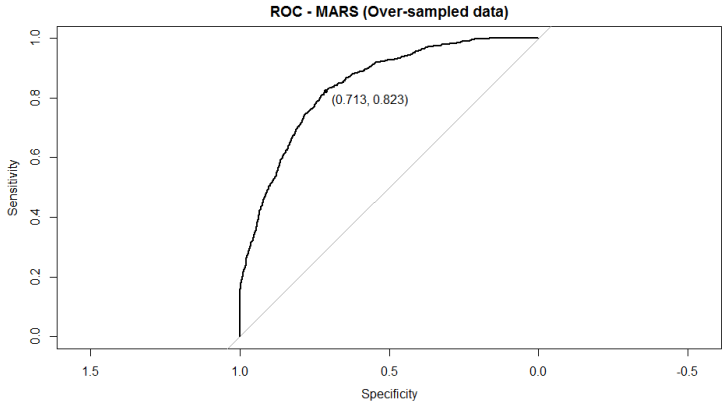

Figure A.5: ROC curve - MARS model

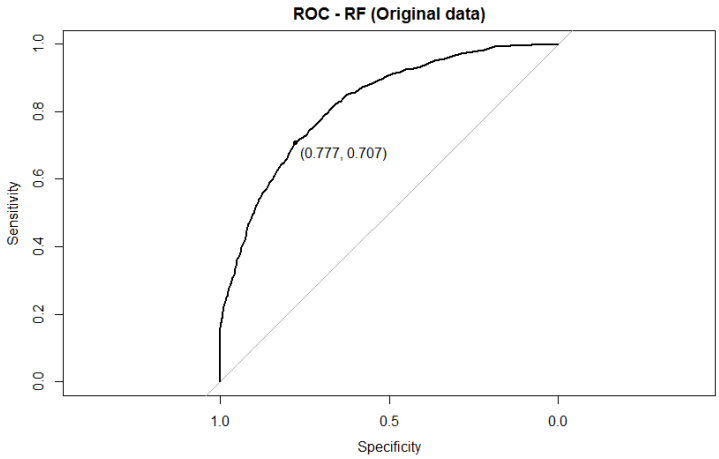

Figure A.6: ROC curve - RF model

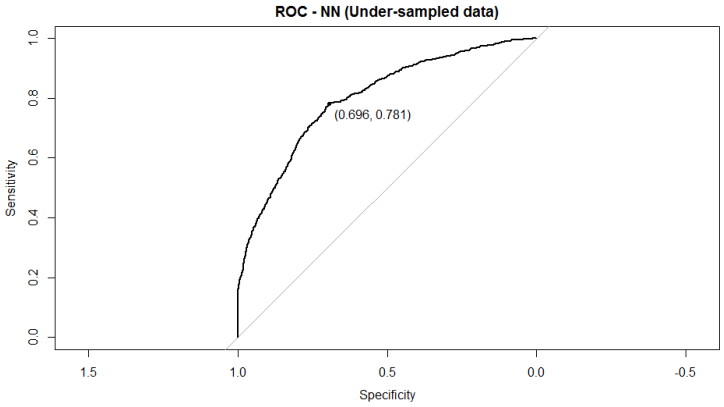

Figure A.7: ROC curve - NN model

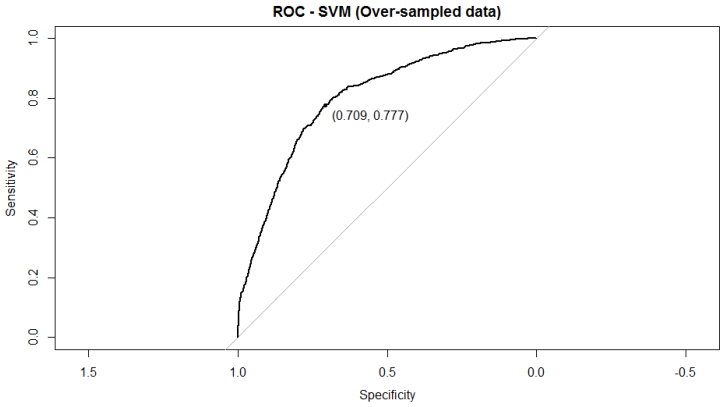

Figure A.8: ROC curve - SVM model

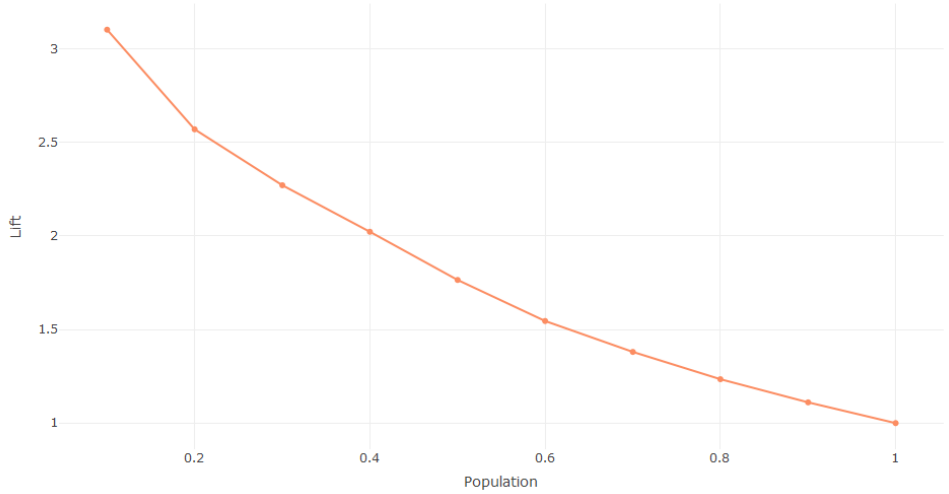

Figure A.9: Lift curve - MARS model

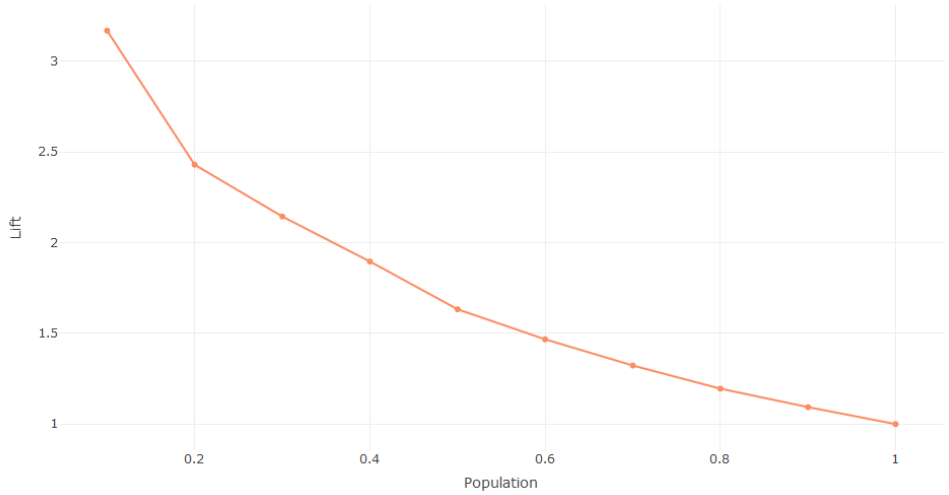

Figure A.10: Lift curve - NN model

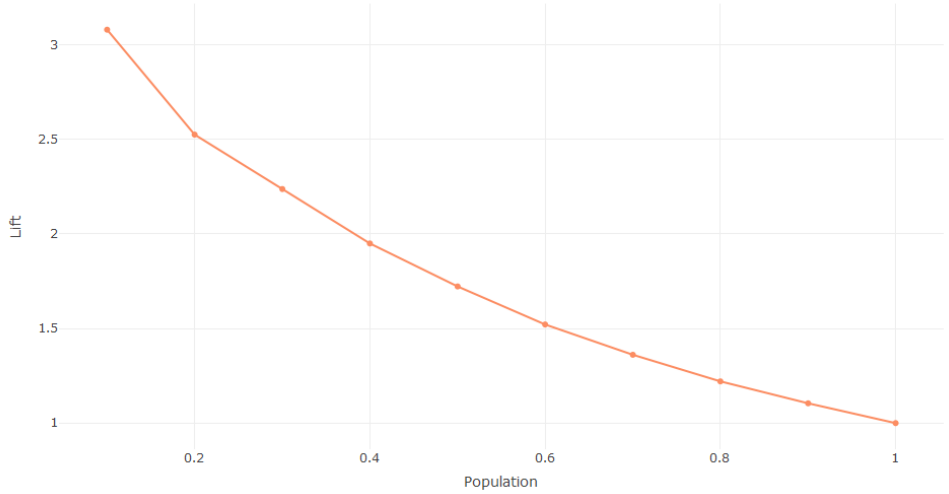

Figure A.11: Lift curve - RF model

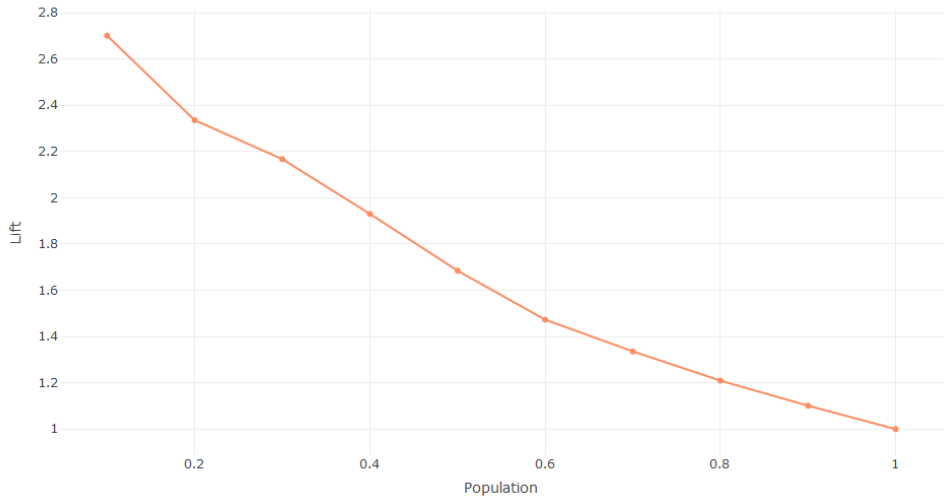

Figure A.12: Lift curve - SVM model

We described three activation functions in Section 3.2, all these functions were considered, and we found that the models built using hyberbolic tangent activation function have the highest sensitivity, specificity and AUC performance statistics as outlined in table A.1.

	Performance statistic	hyperbolic tangent	logistic	linear	Data-set
1	Sensitivity	0.384	0.356	0.344	Original
2	Specificity	0.957	0.956	0.960	Original
3	AUC	0.808	0.803	0.801	Original
4	Sensitivity	0.723	0.713	0.713	Under-sampled
5	Specificity	0.758	0.754	0.749	Under-sampled
6	AUC	0.811	0.810	0.800	Under-sampled
7	Sensitivity	0.778	0.733	0.732	Over-sampled
8	Specificity	0.775	0.755	0.760	Over-sampled
9	AUC	0.848	0.822	0.822	Over-sampled
10	Sensitivity	0.772	0.750	0.742	SMOTE
11	Specificity	0.730	0.700	0.700	SMOTE
12	AUC	0.829	0.788	0.789	SMOTE

Table A.1: Neural network performance: radial basis, polynomial and linear activation functions

In section 3.3, we learned that SVM models use a kernel trick to solve non-linear problems, it does this by applying to the data, kernel functions that project classes into a higher dimensional space to result in linearly separated classes. Table A.2 outlines performance of the kernel functions we considered, and we find that radial basis function yield the best overall performance.

	Performance statistic	Radial basis function	Polynomial	Linear	Data-set
1	Sensitivity	0.388	0.302	0.326	Original
2	Specificity	0.970	0.960	0.960	Original
3	AUC	0.766	0.770	0.696	Original
4	Sensitivity	0.799	0.760	0.759	Under-sampled
5	Specificity	0.688	0.698	0.674	Under-sampled
6	AUC	0.791	0.797	0.777	Under-sampled
7	Sensitivity	0.779	0.761	0.764	Over-sampled
8	Specificity	0.721	0.790	0.667	Over-sampled
9	AUC	0.829	0.839	0.785	Over-sampled
10	Sensitivity	0.728	0.721	0.650	SMOTE
11	Specificity	0.731	0.729	0.717	SMOTE
12	AUC	0.814	0.796	0.650	SMOTE

Table A.2: SVM kernel performance: radial basis function, polynomial & linear kernels

To demonstrate the speed of training the models, we use the under-sampled data, the smallest sample of all the data-sets used to train our models. The results will give us an indication of which machine learning technique is the fastest to train.

The models were trained on a machine with the following specification:
Processor: Intel(R) Core(TM) i7-6820HQ CPU @ 2.70GHz
Installed memory (RAM): 32.0 GB
Operating system: Windows 7 Enterprise 64 bit
R software version: 3.4.1

The outcome of the test is given in Figure A.13, it indicates that MARS is the fastest model to train.

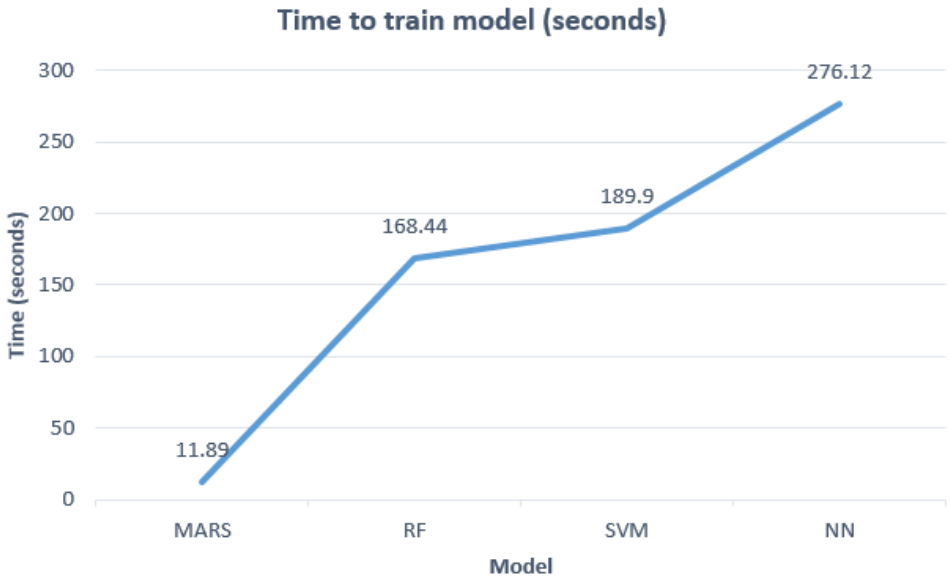

Figure A.13: Time required to train models on the under-sampled data.