Predicting Household Poverty with Machine Learning Methods: The Case of Malawi

A minor dissertation

presented to

The Department of Computer Science

University of Cape Town

In partial fulfilment of the requirements for the degree of

Master's in information technology

by

Francis L Chinyama

Supervisor: Professor Sonia Berman

November 2021

University of Cape Town

The copyright of this thesis vests in the author. No quotation from it or information derived from it is to be published without full acknowledgement of the source. The thesis is to be used for private study or non-commercial research purposes only.

Published by the University of Cape Town (UCT) in terms of the non-exclusive license granted to UCT by the author.

University of Cape Town

PLAGIARISM DECLARATION

1. I know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is one's own.

2. I have used the recognised APA convention for citation and referencing. Each contribution to and quotation in this report from the work(s) of other people has been attributed, cited and referenced.

3. This work has not been previously submitted in whole, or part, for the award of any degree in this or any other university. It is my work. Each significant contribution to and citation in this dissertation from the works of other people has been attributed, cited and referenced.

4. I have not allowed and will not allow anyone to copy my work to pass it off as his or her own work.

5. I authorise the University of Cape Town to reproduce for research either the whole or any portion of the contents of this work.

_____

FRANCIS CHINYAMA

17 November 2021

ABSTRACT

Poverty alleviation continues to be paramount for developing countries. This necessitates the need for poverty tracking tools to monitor progress towards this goal and effect timely interventions. One major way poverty has been tracked in Malawi is by carrying out integrated household surveys every five years to quantify poverty at local and national levels. However, such surveys have been documented as very expensive, tedious, and sparsely administered by many low- and middle-income countries. Therefore, this study looked at whether machine-learning models can be used on existing survey data to predict poor and non-poor households, and whether these models can predict poverty using a smaller number of features than those collected in integrated household surveys.

This was achieved by comparing the performance of three off-the-shelf, open-source machine-learning classification algorithms namely Logistic Regression, Extra Gradient Boosting Machine and Light Gradient Boosting Machine, in correctly predicting poor and non-poor households from Malawi survey data. These supervised learning algorithms were trained using 10-fold cross-validation. The experiments were carried out on the full panel of features which represent all the questions asked in a household survey, as well as on smaller feature subsets. The Filter method and SHapley Additive exPlanations method were used to rank the importance of the features, and smaller data subsets were selected based on these rankings.

The highest prediction accuracy achieved for the full panel data set of 486 features was 87%. When the Filter method rankings were used, the models' prediction accuracy dropped to 63% for the top 50 features subset. However, using the SHAP method rankings, the maximum prediction accuracy level was maintained and only dropped slightly to 86% with the top 50 feature subset; to 84% with the top 20 features; and 73% for the top 10 features. Area under the Curve, Receiver Operating Characteristic curve, recall, precision, F1 score, Matthews Correlation Coefficient and Cohen's Kappa scores confirmed the classification models' reliability.

The study, therefore, established that poverty can be predicted by open-source machine learning algorithms using a substantially reduced number of features with accuracy comparable to using the full feature set. A policy recommendation is to employ only the top explanatory features in surveys. This will enable shorter, lower-cost surveys that can be administered more frequently. The aim is to assist policymakers and aid organisations to make more timely interventions with better targeting of the poorest.

ACKNOWLEDGEMENTS

I thank my wife Chisie, my son Xavier, my daughter Tamika, and the rest of the family, for their patience during my studies and their never-ending support in the furthering of my education. I'm also especially thankful to my Supervisor, Prof Sonia Berman for her patience and guidance through the development of this paper. I thank all my friends and fellow MIT students, Kinsely and Ntsako for their support.

LIST OF ACRONYMS AND ABBREVIATIONS

AB	Ada boost
AUC	Area under the curve
CV	Cross-validation decision tree
ExGBM	Extra gradient boosting machine
FN	False negative
FP	False positive
FPR	False positive rate
LGBM	Light gradient boosting model
LR	Logistic regression
ML	Machine learning
MSE	Mean squared error
NB	Naïve bayes
NN	(Artificial) neural network
OOB	Out of the bag
RF	Random Forest
RNN	Recurrent neural network
ROC	Receiver operating characteristic curve
SGB	Stochastic gradient boost
SHAP	Shapley Additive Explanation
TN	True negative
TP	True positive
TPR	True positive rate
VIF	Variance inflation factor

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 Background

Poverty is a very significant problem for mankind. The United Nations (UN) reported that as of 2015, 736 million people existed below the poverty line of US$1.90 a day (UNDP, 2018). The UN Sustainable Development Goals capture the UN vision to eliminate poverty in all forms and dimensions by 2030. Governments and international aid agencies continue to partner in targeting the most vulnerable members of populations with sustainable development interventions that are typically designed to lift people from poverty by improving their well-being in health, education, and income generation. Despite such interventions, South Asia and sub-Saharan Africa remain the hotbeds of poverty with 80% of people in these regions estimated to be living in extreme poverty (United Nations, 2019). For Malawi, 50.7% of the population were estimated to be living below the poverty line and 25% in extreme poverty (International Monetary Fund, 2017).

Tracking poverty indicators is critical to gauge progress in eradicating the problem. For instance, assessing poverty levels in communities helps to determine the effectiveness of a poverty intervention and whether it is achieving the desired outcome. Using data to track progress and evaluate the effectiveness of interventions is an integral part of any aid programme. National household surveys are an important source of data used in such analyses. However, conducting surveys involves administering questionnaires to selected individuals and households to evaluate household consumption and expenditure on food, clothing, energy, housing, transport, health and assets such as cars. This makes survey administration very time-consuming and expensive (Blumenstock, Cadamuro, & On, 2015).

Various methods have been used to analyse survey data to predict poverty. Before 2011, studies typically applied regression statistical models. Recent studies applied more advanced statistical techniques such as proxy means (a survey-to-survey imputation), the World Bank SWIFT method and poverty scorecards (Dupriez, 2018). While this is so, regression models have a challenge with multicollinearity, where two features are highly correlated with each other causing misleading predictions and poor accuracy (Vatcheva, Lee, McCormick, & Rahbar, 2016).

Machine learning (ML) methods create a better platform for analysing the ever-increasing consumption data to estimate the dynamics of national economic well-being, poverty trends and programme effectiveness. In more recent years, ML methods have begun to be applied to analyse economic well-being status using various data types such as satellite images and mobile phone call data (Jean, et al., 2016). However, the data used here are proprietary and sourcing it is costly. Moreover, the ML methods used are complex and require significant computational power and expertise, which are also costly.

This study explores the use of off-the-shelf and non-proprietary ML methods to predict poverty, using publicly available survey data. It goes further, by attempting to use fewer variables rather than the full range of features tracked in typical surveys. If successful in generating results with similar accuracy to those generated by traditional methods, such an approach would have succeeded in generating desired information by using freely available data and open-source ML methods and low computational power, hence easily applicable in a low-resource setting like Malawi. Moreover, if fewer data points can be used, smaller surveys that take less time to administer could then be carried out at a lower cost and with greater frequency.

The study builds on the work of the World Bank that also explored the use of ML methods to predict poverty. The World Bank conducted this exploration through a company called DrivenData Incorporated who hosted a data science competition on poverty prediction using ML techniques. The three best performing models were published on the World Bank website, were ensembles using ExGBM, LGBM and Logistics Regression (Fitzpatrick, Bull, & Dupriez, 2018).

1.2 Problem Statement

The millennium UN goal is to eliminate extreme poverty in low resource countries (United Nations, 2019). To know if this goal has been reached requires effective ways to monitor poverty trends in poor countries. This information allows policymakers and international development partners to effect the appropriate interventions to ensure that developmental goals are on track.

Traditional approaches have used data from surveys to measure poverty levels. However, research has shown that door-to-door surveys are expensive and time-consuming (Jean et al., 2016). In Malawi, surveys are typically conducted on an average of every five years, but this

depends largely on funding availability from international development organisations (Malawi Government, 2017). This means that poverty estimates of the population are not available in between survey years. This creates a problem of policy design and decision-making because old data are used to design development programmes, leading to inaccurate results (Jerven, 2013).

International aid agencies and governments have employed various methods to generate data for poverty analysis in the period between the five-year surveys. Survey-to-survey imputation is one of the popular ways to predict changes in poverty levels using data obtained from other surveys that are conducted more frequently. Therefore, survey-to-survey imputation has the potential to generate more frequent estimates by using similar data variables captured in other surveys at a negligible additional cost (Newhouse, Shivakumaran, Takamatsu, & Yoshida, 2014).

Newhouse et al. (2014) report three significant drawbacks to successful imputation. These drawbacks include the different wording of questions in any two types of surveys, which will have a bearing on responses provided by households. Imputation is restricted to data variables that are available in both surveys, which may not provide the full set of data points needed to accurately detect household expenditure changes per capita. Besides, if two surveys apply different sampling strategies, validation tests will portray the imputations as less accurate.

More recently, ML methods have been applied to predict poverty levels. However, the data that has been used such as satellite imagery and mobile phone usage data are proprietary and expensive. These studies also require high computing power, making them costly and challenging to conduct in a low-resource setting like Malawi.

The current study attempts to assess poverty levels using ML classification methods instead of survey-to-survey imputation. However, rather than using the more expensive data processing techniques that have primarily been used in ML, this investigation uses survey data from Malawi's five-year household surveys that are freely and publicly available on the World Bank's data portal. The ML frameworks tools that include PyCaret and Sci-kit, are off-the-shelf and open source and can be run on low computing power. This study goes further and explores the assessment of poverty levels by using a shortlisted set of variables to see if similar levels of prediction accuracy can be achieved with fewer variables than when the full range of variables available from an integrated household survey data are used.

This study can therefore be summarized as pilot research to understand the feasibility of using off-the-shelf ML tools to measure poverty using survey data with far fewer features. It mimics recent work of the World Bank that explored the use of ML methods to predict poverty and examines the impact of using reduced feature sets to identify the most vulnerable households and individuals.

1.3 Research Objectives

The main objective of the study is to gauge the feasibility of using fewer features that can be cheaper and quicker to estimate poverty. Specifically, the study aims to:

i) Predict poverty using off-the-shelf open-source ML tools executed on survey data; and

ii) Explore whether poverty can be predicted by using a smaller number of features.

1.4 Research Questions

To achieve the objectives, the study sought to answer the following research questions:

i) Can poverty be predicted using off-the-shelf ML tools employed on survey data?

ii) Can poverty prediction based on a smaller number of features give acceptable results when compared to results obtained using the full-panel feature set?

1.5 Importance of the Study

The goal of this exploration is to minimize the time and cost of data collection while still achieving similar poverty prediction accuracy levels. If successful in achieving similar accuracy levels, the set of shortlisted features can inform the design of much shorter questionnaires. Shorter questionnaires make surveys far less expensive and time-consuming to conduct. Streamlined surveys can be administered through other channels such as smartphones instead of on paper. It could also allow for larger samples to be taken to provide a more precise picture of the population's current welfare. Moreover, improved questionnaires can lead to variance reduction that is useful for monitoring the variables and improved prediction of future trends.

Poverty classification using ML has the potential to monitor poverty level trends more cost-effectively and more closely so that poverty reduction interventions can be effected by policymakers quickly and with improved identification and targeting of those most vulnerable.

1.6 Organisation of the Thesis

The remainder of this thesis is organised as follows:

- Chapter 2 reviews previous and current literature relevant to the study. Concerns and development of the poverty prediction field are reviewed, and the supervised ML classifiers are chosen.
- Chapter 3 addresses the methodology applied in this study. It outlines the experimental process of developing the ML models, selecting feature subsets, and evaluating prediction success.
- Chapter 4 analyses the results and answers the research questions.
- Chapter 5 concludes the study. It makes poverty prediction policy recommendations and suggests areas for further research.

CHAPTER 2
LITERATURE REVIEW

2.1 Introduction

This chapter reviews approaches to poverty prediction using ML methods. It does so by analysing the data types used, the ML tools and approaches used, and model performance achieved in the studies. Data types utilised are survey data (obtained from completed questionnaires), mobile phone records and satellite images.

2.2 Overview of Machine Learning

In recent years, ML methods have gained popularity for use in data analysis due to the increased availability of memory and high computing power. ML seeks to learn from data to identify patterns to use them in another context to produce similar outputs with minimal human intervention (Géron, 2019). ML methods can be grouped into three categories: supervised learning, unsupervised learning, and reinforcement learning.

2.2.1 Supervised Learning

Supervised learning is primarily used to make predictions about unseen or future data, based on past occurrences. Classification is a subgroup of supervised learning which aims to correctly predict categorical class labels of new occurrences based on historical observations. The other subcategory of supervised learning is regression analysis, which is used to predict continuous outcomes (Raschka, 2015).

Supervised learning aims to produce a function that can predict a target variable based on pre-labelled input data. In so doing, the algorithm constructs a model that can process new data to predict that target variable. Korivi (2016) provides one example to demonstrate how supervised learning is used in classification, to say an algorithm can be trained to recognize an animal by running it through a dataset of prelabelled images of the animal's species (Korivi, 2016).

Two main steps are followed to classify data by means of supervised learning:

i) Training: where labelled datasets are used to construct a classification model or classier:

ii) Testing: where the trained model is used to classify new unlabelled data.

The need to label input data for training, validation and testing steps is the major challenge to this approach as this requires domain knowledge and adds to the cost of labelling the input data (Nguyen & Armitage, 2008).

2.2.2 Unsupervised Learning

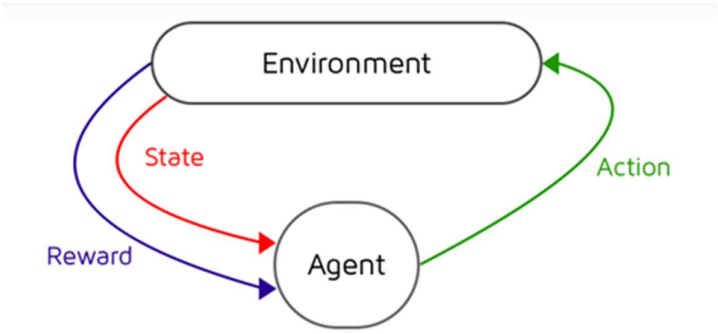
Unsupervised learning techniques such as clustering, are mainly used in discovering hidden patterns or structures in datasets (Raschka, 2015). Input data is not annotated and has no labels. Unsupervised ML algorithms search for patterns and group data into clusters based on similarity in features. The algorithm aims to define the number of clusters and their distributional shape (Nguyen & Armitage, 2008). Complexity and computational resource requirements are some of the challenges that unsupervised training algorithms encounter. Furthermore, unsupervised training algorithm requirements for feature selection can be challenging to achieve (Williams, Zander, & Armitage, 2006).

2.2.3 Reinforcement Learning

Reinforcement learning is an ML technique that uses an agent to find the right policies by learning from interacting with the environment and receiving feedback. It is thus used to solve interactive problems (Géron, 2019; and Raschka, 2015).

Reinforcement learning aims to solve a Markov Decision Process defined by a four-element tuple M = (s, a, Psa, R). The letter "s" represents the state space in which the agent makes a decision and "a" represents a set of actions provided for the agent to choose. Psa is a probability distribution representing the state's probabilities of transferring to different states after action "a" and "R" represents the reward (Géron, 2019). As an illustration, P (s′ | s, a) stands for the probability of the current state (s) transferring to the next state (s′) after action (a). The letter R with parameter (s, a) is a reward function R (s, a) that represents the reward received by the agent after the execution of action under the state (s). A reinforcement learning process can be described as depicted in Figure 1.

Figure 1. A depiction of reinforcement learning

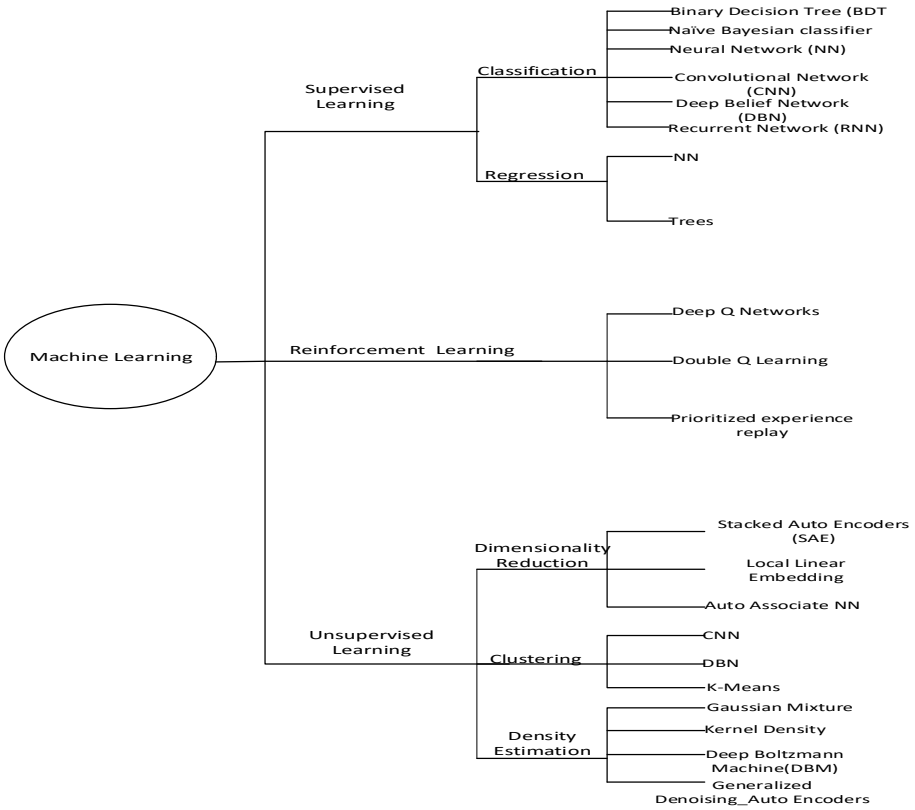


Source: Analytics Vidhya (2020, p. 32)

2.2.4 A Depiction of Machine Learning Categories and Algorithms

Figure 2 below presents an overview of ML techniques and the algorithms associated with each.

Figure 2. A summary of machine-learning techniques and algorithms



Source: Fadlullah, et al., 2017 (p. 10)

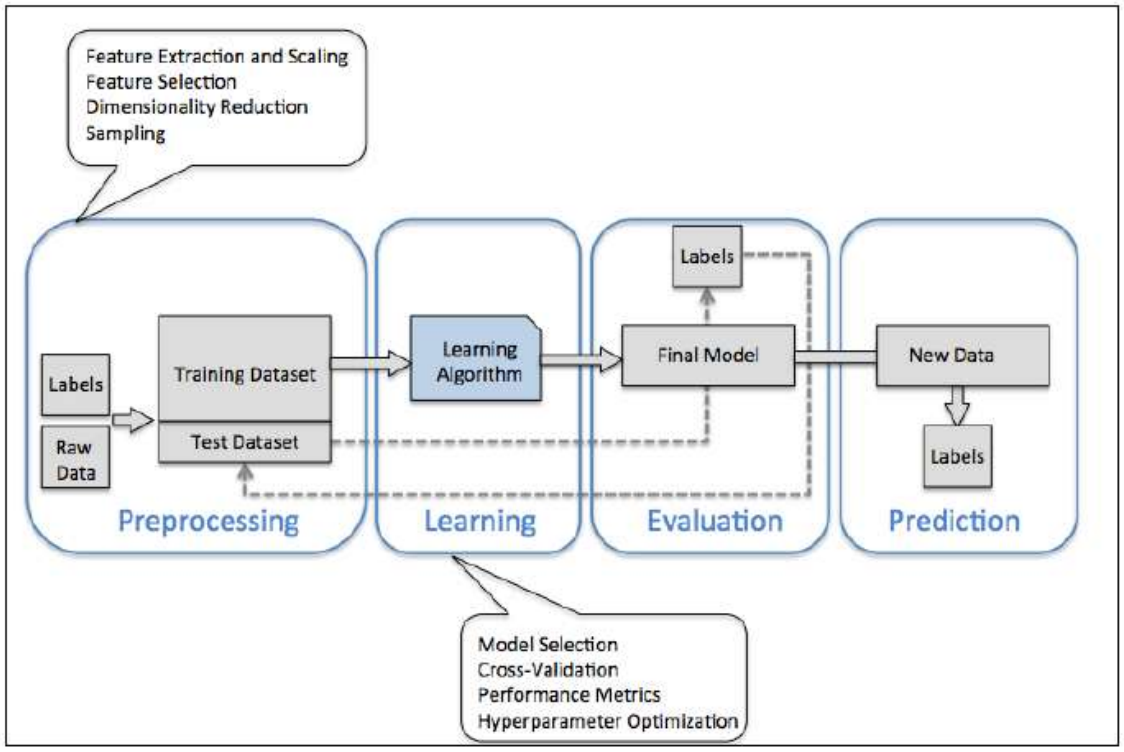
2.2.5 A Roadmap of Constructing Machine Learning Systems

Raschka (2015) captures the typical pipeline for building a machine learning system. This includes four main steps starting with data pre-processing, to prepare it for use by any ML algorithm to get better outcomes.

As captured in Figure 2 in the above section, many different ML algorithms have been created to solve different problem sets. The second step in the ML workflow is learning, where an ML algorithm is trained using input data from the problem task at hand to develop the predictive model. This is followed by model evaluation to gauge the performance of the predictive model, based on predefined performance metrics. If satisfied with its performance, the model can then be used for prediction on new or unseen data.

Raschka (2015)'s workflow diagram is presented in Figure 3 below.

Figure 3. A typical workflow diagram for using ML in predictive modelling



Source: Raschka (2015), p.11

2.3 A Review of Studies Applying Machine Learning to Predict Poverty

ML techniques have been applied in different fields with varying success. Lately, ML techniques have also gained attention in predicting economic well-being and improving poverty alleviation targeting (Mullainathan & Spiess, 2017). These studies have used different data types and the most common are survey data, mobile phone data, satellite imagery, or a combination of these. The section below analyses investigations published in the last 10 years applying ML to measure poverty level in populations.

2.3.1 Machine Learning Classification Using Satellite Imagery and Geospatial Data

Several studies have used a combination of satellite images and ML to predict poverty. Jean et al. (2016) pre-trained a convolutional neural network (CNN) to identify image features. The CNN was then further trained to identify night light intensities corresponding to the daytime satellite imagery. Using a combination of expenditure data and the corresponding daytime imagery features for those clusters, ridge regression models were trained to predict economic activity. The study was conducted on five countries namely Rwanda, Uganda, Nigeria, Tanzania, and Malawi. The results were compared with those obtained using survey data alone. The ML method by Jean et al. (2016) was found to explain between 55% (Malawi) and 75% (Rwanda) of the variation in economic outcomes in those countries.

Head, Tran, Manguin and Blumenstock (2017) replicated Jean et al.'s (2016) prediction method and conducted a study in Rwanda, Nigeria, Haiti and Nepal. Head et al. (2017) also explored if the same method could be applied to predict other sustainable development indicators namely access to water, education, and health. They had similar results to Jean et al. (2016) for poverty prediction for the two common countries (74% for both Rwanda and Nigeria), though the results for the other sustainable parameters were different.

Notably, the ML on satellite imagery while successful to varying degrees in predicting poverty results for economic clusters, does not provide results at the individual household level. In addition, acquiring spatio-temporal satellite images as input for the ML algorithm may prove to be challenging for low-resource countries such as Malawi. This is due to the ability to source computational power needed to process high-resolution satellite images.

Bilton, Jones, Ganesh, and Haslett (2017) used geospatial data coupled with survey data to predict poverty in Nepal. The poverty prediction once calculated were imposed on a spatial map of Nepal to show the incidence of poverty. Decision tree-based methods and Random

Forest were employed using the household survey data and geospatial data to predict the target variable poverty at the cluster level. Their method was used because of its inherent ability to automatically select important features while avoiding over-fitting. Results gave an accuracy of 67% for predicting poor clusters. The poverty estimates attached to geo-locations are important in providing information to policymakers and aid agencies to optimize aid allocation. However, this method also measures poverty at a cluster level, not at the individual household level.

Tingzon et al. (2019) used the methods developed by Xie, Jean, Burke, Lobell and Ermon, (2016) and Jean et al. (2016), incorporating open-source geospatial data from the OpenStreetMap project to predict poverty. The results of the models, which combined regional indicators as features, gave an accuracy in predicting poverty of 63%. The conclusion of the investigation showed that models trained on publicly available, and volunteer labelled geographic data attained the same accuracy as that of models trained using proprietary satellite images. The challenge and drawback of this method is the requirement for domain knowledge to curate and annotate the geospatial images for poverty features.

2.3.2 Machine Learning Classification Approaches Using Mobile Phone Data

Blumenstock et al. (2015) used a combination of ML methods, survey data and mobile data to predict poverty. Their method combined feature engineering and feature selection in a two-step approach. First, they used combinatorial deterministic finite automatic to transform individuals' call logs into a large set of quantitative features. After creating the features, elastic net regularization was used, to eliminate all irrelevant metrics identified to have low predictive influence towards the target variable wealth. These features were additionally used by Random Forest base ensemble classifier to predict poverty. Cross-validation was carried out to limit model over-fit on the small training sample. A composite wealth index was constructed using the first principal components of responses related to wealth from several demographic and health surveys. The results generated by the lean model were then compared against this index to predict if an individual is poor or non-poor

Steele et al. (2017) used surveys and mobile phone call records to classify individuals' economic well-being in Bangladesh. They used the ensemble method on large-scale mobile phone data to predict household poverty levels. The study aggregated datasets from census data, mobile phone operator data and geospatial data to train models that could predict poverty. Models using the combined datasets' features achieved an r-squared coefficient of

determination of 78%. However, models using only mobile phone operator data produced comparable results. This investigation demonstrated that mobile data only could be used to predict poverty. The findings showed the potential to estimate and continually monitor poverty at a local level.

Using mobile phone data to predict poverty generates information at an individual level and can easily be applied when survey data is not available. However, not all households own a mobile phone so that sections of the population, especially the extremely poor, would be left out. In addition, Steele et al. (2017) reported the reluctance of mobile phone operators to share data due to business and privacy concerns, pointing to a key challenge on relying on the availability of this proprietary data to monitor poverty.

2.3.3 Machine Learning Classification Using Survey Data

ML methods have also been applied to survey data on their own in addition to being used in combination with or to validate or supplement other data types in classifying poverty. Several studies that used ML classification methods on survey data to predict poverty are outlined below.

Fitzpatrick et al. (2018) compared the results of optimized linear regression with 10 modern approaches to classification including advanced decision tree methods, genetic algorithms and deep learning methods. In so doing, they were able to assess the trade-offs between the computational complexity of the methods and model performance. The classification algorithms used were all open source and applied to survey data for Malawi and Indonesia. Survey responses were converted into features, which were used as predictors to identify "poor" or "non-poor" households. The results found no major difference in the predictive ability of linear regression and the more advanced ML methods. In fact, in all the cases, linear regression performed better than the more advanced methods, emerging among the three top predictors. The results in the accuracy of prediction ranged between 73 – 87% for Malawi and 88% to 91% for Indonesia before model fine-tuning, 86% - 87% for Malawi and 81% - 85% for Indonesia thereafter. For their simple data set of 10 features, model accuracy ranged from 73 – 77%. Notably though, their top 10 features were derived using a stepwise approach analogous to that used in regression techniques in identifying top predictors.

Thoplan (2014) explored using random forests with census survey data to predict poverty. This approach was chosen because of the model's low computing requirements, accuracy, and ability not to overfit to training data. Furthermore, random forests are very efficient in selecting

features that are important in influencing the target variable. The random forests prediction results showed that out of the bag (OOB) error mean in classification of poor and non-poor households to be 0.175.

McBride and Nichols (2018) employed survey data to predict poverty using cross-validation and stochastic ensemble methods to improve the predicting ability of proxy means test (PMT) tools used by the United States Agency for International Development (USAID). The study concluded that using a stochastic ensemble approach could reduce the time it takes to perform feature selection and improve the process of comparing multiple ML models' performance. Overall prediction results of the ensemble approach showed that there was a conservative gain in performance against other methods. The results in the accuracy of prediction ranged between and 87 % for Malawi and 55% for Bolivia.

The challenges of sourcing proprietary data and the computer processing requirements for high-resolution images reported for the other data types are not the case when survey data is used.

Table 1 summarizes these studies, including the nature of the datasets, the ML approaches to classification methods used, and levels of accuracy achieved

Table 1. Poverty classification studies using machine-learning methods

Study	Dataset	Sampling Technique	Feature selection	Classification	Performance
Studies classifying poverty using satellite and geospatial data					
Jean et al. (2016) Rwanda, Uganda, Nigeria, Tanzania, Malawi	High-resolution satellite imagery covering 300,000 locations in the 5 African countries + corresponding survey data	Convolutional Neural Network (CNN) pre-trained to identity image features in the daytime and fine-tuned to identify the same features at night	Trained CNN with over 55 million parameters used as a feature extractor	Mean cluster-level values from survey data + corresponding daytime imagery features extracted by the CNN used to train ridge regression models that can predict economic activity at cluster level	Rwanda 2010 r^2= 0.75 Uganda 2011 r^2= 0.69 Nigeria 2013 r^2= 0.68 Tanzania 2010 r^2= 0.57 Malawi 2010 r^2= 0.55
Head et al. (2017) Rwanda, Nigeria, Haiti, Nepal	Satellite imagery (millions of google map images) of 4 countries + corresponding survey data	Same as above	Same as above	Same as above	Rwanda 2010 r^2= 0.74 Nigeria 2013 r^2= 0.74 Nepal r^2= 0.64 Haiti 2010 r^2= 0.51
Bilton et al. (2017)	Geospatial data and Survey data of 3912 households	Monte Carlo simulations + Bootstrap resampling	Bootstrap method	Small area estimation of poverty incidence Tree classification	ELL gives P0 = 0.351, SE = 0.014

Study	Dataset	Sampling Technique	Feature selection	Classification	Performance
Tingzon et al. (2019)	Combined Survey data 27,000 households, Satellite imagery 297,000 and Geospatial data 150	CNN pre-trained to identity image features in the daytime and fine-tuned to identify the same features at night	Trained CNN 4,096 vector and Ridge regression	CNN with transfer learning Random forest	r^2= 0.63
Xie et al. (2016)	Survey data and 14 million images with 1000 classes	CNN	CNN	CNN	Accuracy 71.71%
Studies classifying poverty using mobile phone data					
Steele et al. (2017)	Combined Survey data for 90,000, mobile phone data for 76,000, Geospatial data	Non-spatial generalized linear modelling (glms).	Bivariate Pearson correlations	Bayesian geostatistical models (BGMs)	r^2= 0.78
Blumenstock et al. (2015) Rwanda	Billions of anonymized interactions on Rwanda's largest mobile phone network + a phone survey of randomly selected 856 subscribers to collect data on economic welfare indicators, Demographic and health survey data to create wealth index	5 metrics selected	Two-step approach to predict composite wealth: i) Combinatorial deterministic finite automatic (DFA) to generate quantitative metrics from phone logs and ii) Elastic net regularization to eliminate irrelevant features and create a generalizing model for prediction + cross-validation to limit overfitting on the small training sample	Composite wealth index constructed using first principal component of survey data Lean model generates results marked against wealth index Tree-based ensemble regressors and classifiers (random forest) to aggregate prediction results	Local cluster level r = 0.79 District level r = 0.92 Average wealth of a district, as predicted by the mobile phone data, r = 0.917

Study	Dataset	Sampling Technique	Feature selection	Classification	Performance
Studies classifying poverty using survey data					
Fitzpatrick et al. (2018) Malawi, Indonesia	Household survey data from a nationally representative sample	Synthetic Minority Oversampling Technique (SMOTE), for the Indonesia dataset	Categorical variables (response to survey questions) are as poverty predictors. Classification is performed using the full features dataset and a set of shortlisted features Variance inflation factor (VIF) testing to eliminate redundant features to generate shortlisted set	10 open-source ML classification algorithms are applied to predict poverty and results compared. The target prediction variable of all methods is a binary label "poor" or "non-poor", to classify households	MALAWI: Accuracy = 78 - 87% for full feature set; and = 73 - 77% for simple feature set INDONESIA: Accuracy = 88% - 91% for full features set; and = 90 - 91% for fewer features set
Thoplan (2014)	Survey data, 296 294	Random Forest Bagging	Gini Score	Bootstrap on tree applied on a random sample of observations Out-of-bag (OOB) prediction obtained using a majority vote across trees	Out of the bag (OOB) Error Mean = 0.175
McBride and Nichols (2018)	Survey data, 1800- 11280	Regression forest Quantile regression forest bootstrap aggregation	Random forest for feature selection	Ensemble methods (Bagging and then applying random forest algorithm for classification) Model selection based on cross-validation	Total Accuracy Malawi Mean = 80% East Timor Mean = 75% Bolivia Mean = 64%
Kshirsagar, Wieczorek,	2015 Zambia LCMS survey data	Elastic net logistic regression	Bootstrap variable selection	Elastic net logistics regression	Probability = 0.85

Study	Dataset	Sampling Technique	Feature selection	Classification	Performance
Ramanathan and Wells (2017)		Cross validation			
Knippenberg, Jensen and Constas (2019)	Survey data, 576 households	N/A	Least Absolute Shrinkage and Selection Operator (LASSO) and Random Forest to identify the best predictors of Need.	LASSO and Random Forest to identify the best predictors of Need.	Out of sample (April, May) LASSO r^2= 56.4% Random Forest r^2= 55.6%
Sohnesen and Stender (2017)	Survey data, 1800 – 18000 In 6 countries	Random Forest	Entropy loss function Gini loss function	Random Forest	National Mean square error (MSE) Gini = 1.71 Entropy = 1.94 Urban/Rural MSE Gini =2.58 Entropy =2.58
Gravemeyer, Gries and Xue (2010)	Survey data, 1056 households and 3256 individuals	Logit regression Tobit regression Probit regressions	Empirical Truncated Censored	Regression statistical method of measuring poverty Regression is applied, and variables are truncated; others are censored. This allows us to have coefficients	Probit r^2= 74% Tobit r^2= 75% OLS r^2= 53.6%

2.4 An Analysis of the Machine Learning Approaches to Predicting Poverty

For classification, the type of learning used has been supervised learning where training and testing data has labels. Classification algorithms used include regression algorithms, tree-based algorithms, neural networks, and ensemble methods which combine tree based and regression algorithms.

Studies to predict household poverty such as the one used by Steele et al. (2017), aggregates several base classifiers to improve performance over a single classifier (Sundsøy et al., 2016). Steele et al. (2017) used the ensemble method to classify individuals' economic well-being using mobile phone call details records which are big data. Steele et al. (2017) investigated the performance of two different ensemble methods known as Random Forest and Gradient Boosting Machines.

Convolutional neural networks (CNNs) are another approach used by Jean et al. (2016), Head et al. (2017), Blumenstock et al. (2015), and Xie et al. (2016). The studies employed convolutional neural networks (CNN) on daytime and night-time satellite images to identify image features that could classify poor households. All these studies leveraged the transfer learning technique where the images were pre-trained using another dataset, and the results obtained were used as input for deep learning algorithms (Jean et al., 2016). However, the first step in the process was establishing a baseline using demographic health survey datasets. This was followed by computing satellite-based "features" for each image cluster using a CNN to extract features from satellite imagery covering the area of interest. The CNNs were pre-trained on ImageNet and then optimized to predict categories of night-time light intensity from daytime satellite images. The final step employed a ridge regression model to learn the functional parameters from satellite features to produce cluster-level indicators. The optimized CNN and linear models are then used to predict poverty levels given random daytime satellite images ((Blumenstock et al., 2015; and Jean, et al., 2016).

The highest accuracies achieved using this approach were around R^2 of 0.74 for Rwanda and Nigeria. However, the studies showed that night-lights' dependence to provide features for the algorithm could not generalize well in different settings.

Sundsøy et al.'s (2016) investigation compared deep learning with traditional methods using mobile phone data. Deep learning has sub-domains and the most popular are the Artificial Neural Network (ANN), Deep Neural Networks (DNN), Convolutional Neural Networks

(CNN), Deep Belief Networks (DBN) and the Stacked Auto-Encoders (SAE). An ANN was applied by Sundsøy et al. (2016) because of its ability to deduce complicated functions representing the weights and bias of the dataset and the capability to replace handcrafted feature engineering. The investigation showed promising results for Deep Learning techniques. The classification receiver operating characteristic (ROC) area under the curve (AUC) metric of Deep Learning was 0.77, while Random Forest was 0.64 and Gradient Boosting 0.61 (Sundsøy et al., 2016). Deep Learning has better performance than traditional data mining approaches, but it is not interpretable and is also computationally expensive (Doshi-Velez & Kim, 2017).

In a study titled "Predicting poverty and wealth from mobile phone metadata", phone call detail records (CDR) records of 876 volunteers were converted into more than 5000 measures that could determine total volume, intensity, timing, and directionality of communication (Blumenstock et al., 2015). Examples of features derived from the metrics were numbers of calls per day, unique cell towers visited, weekly airtime expenditure and the number of international contacts, to name a few. The poverty estimates were compared against the poverty estimates from national consumer survey data. The second step used an elastic net regularization algorithm to eliminate irrelevant phone metrics and select a low resource model. Blumenstock et al. (2015) compared the phone-based estimates of average cluster wealth against the National demographic and health (DHS) survey of 492 clusters of 2010 achieving $R^2 = 0.79$ at the cluster level, which is lower than at the district level $R^2 = 0.92$. This method showed improved results in the classification of households with different economic status by reducing the number of features.

Another popular ML classification algorithm is the C4.5 decision tree-based classifier that was applied to poverty mapping studies by Sani, Rahman, Bakar, Sahran, & Sarim (2018). The binary decision trees algorithm was used to estimate poverty in Nepal and the results from the simulation showed low bias and variance against the training set. Decision trees also make few feature assumptions on the training data. Alternative techniques such as multiple regression apply the stepwise method that requires feature engineering at a preliminary stage. This requirement causes overfitting problems and reduces reproducibility due to the high dependence on domain experts in carrying out the feature engineering process (Kshirsagar et al., 2017). On the other hand, the decision tree method automatically selects features and is better suited for non-linear relationships in the features.

Kshirsagar et al. (2017) used elastic net logistic regression with cross-validation and parameter regularization on survey data. The method involved variable selection, fitting the selected model, and translating the derived model into a usable scorecard. The results showed that logistic regression could predict the probability of poverty of rural non-poor at 0.50, rural poor at 0.85, urban non-poor at 0.20 and urban poor at 0.75. The main limitation of employing the above method to quantify poverty is the limited choice of features as input that could be used to accurately predict poverty across all sectors of households in a country.

McBride and Nichols (2018) applied cross-validation and stochastic ensemble methods to create a proxy means tool. The results showed that the prioritization of the proxy means tool performance using out-of-sample data could improve by selecting a model based on utilizing cross-validation regression algorithms or using a method such as stochastic ensemble methods. The study showed a gain in poverty accuracy, a reduction in false-positive classification rates as compared to traditional methods classification.

2.5 Choosing the Machine Learning Approach to this Study

Classifiers with the highest complexity such as deep learning classification algorithms have been applied on more the complex data types like satellite imagery and big data like mobile phone records. Such techniques which explore the entire feature space will have higher classification accuracy but require high computational power. Use of such data as mobile phone records and satellite imagery can be further challenging due to proprietary reasons.

In a place like Malawi with limited and costly internet, the methods currently applied to classify poverty in between surveys such as Proxy Means Test (PMT), require domain knowledge (McBride & Nichols, 2018). Some of the studies reviewed have explored alternative ML approaches that are lower on computational complexity, yet still achieve acceptable accuracy. These have used data from national surveys that are publicly and readily available on government websites or those of international development institutions such as the World Bank. This was a key motivation for this study to focus on ML classification methods that utilize survey datasets that are readily and publicly available in predicting poverty.

In a pioneer study by Fitzpatrick et al. (2018) to use ML approaches to poverty classification using household survey data, the performance of the various classification algorithms was ranked. Three of the top-performing classification algorithms in the Fitzpatrick et al. (2018) study, namely Logistic Regression (LR), Extra Gradient Boosting Machine (ExGBM) and

Light Gradient Boosting Machine (LGBM) were selected for use in this study. LR was of particular interest due to its simplicity. ExGBM and LGBM were among Fitzpatrick et al. (2018)'s top performing classification algorithms and were selected together as they are closely related. They were also among the top-performing models in the World Bank ML poverty classification competition. These methods are elaborated below.

2.5.1 Logistic regression

Logistic Regression was chosen to assess household poverty probability following the method devised by (Rahman, 2013). This model or classifier was selected as a baseline model to compare with the other two classifiers. The LR classifier computes a weighted sum of the input features in addition to a bias term and it outputs the logistic of the result following the below equation.

$$\hat{p} = h_\theta(x) = \sigma(\theta^T \cdot x)$$

Equation 1

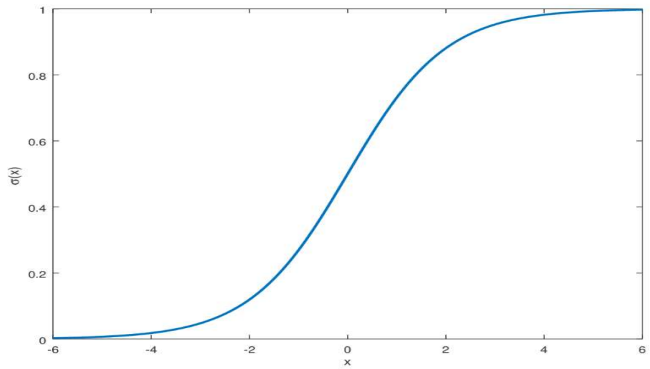
The logistic is also called the logit and is represented by the symbol $\sigma(\cdot)$ and this is denoted as the sigmoid function that outputs a number between 0 and 1 using the below logistic equation.

$$\sigma(t) = \frac{1}{1 + exp(-t)}$$

Equation 2

In Figure 3 below, the Logistic Regression model estimates the probability $\hat{p} = h_\theta(x)$ the probability that a household instance x belongs to the class poor if it is more than 0.5 and non-poor if it is less than 0.5.

Figure 4. Logistic function plot as a function of x



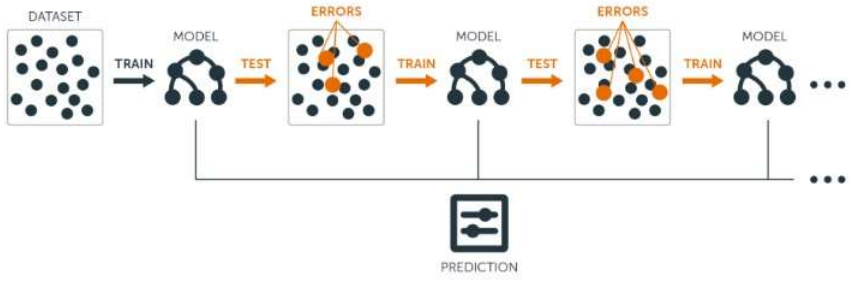
Source: Adapted from Géron (2019)

2.5.2 Gradient Boosting

Gradient boosting is a technique that combines classic weak ML algorithms and turns them into much more powerful classifiers called ensemble classifiers or ensembles. This process is iterative, and the contribution of each weak base classifier is optimised at each reiteration until suitable classification performance is achieved (Friedman, 2002).

Gradient boosting methods is a simple sequential and gradual process where weak ML algorithms are optimised by adding more classifiers to create a much more powerful classifier. The aim is to build an ensemble classifier that allows the weak classifiers to learn from the mistakes of the predecessor classifier thus optimizing the loss function. Technically this is attained by tweaking the loss function into a least-squares optimization problem (Géron, 2019). This process has been captured in Figure 4 below:

Figure 5. Gradient boosting pipeline processing



Source: Analytics Vidhya (2020, p. 17)

2.5.3 Extreme Gradient Boosting

Extreme gradient boosting (ExGBM) is an improved sub-class of gradient boosting, designed to be efficient, and flexible in handling both large and small datasets. This was developed by Chen and Guestrin (2016) to address some limitations of gradient boosting machines. Although ExGBM provides the same boosting and tree-based hyperparameter optimisation of GBM it does a remarkable job in handling overfitting through the process of regularization. This ability makes the model faster and flexible during model training. The regularization technique is achieved by adding a parameter to the model loss function as presented below.

$$L(\phi) = \sum l(\hat{p}_i, p_i) + \sum_k \Omega(f_k)$$

$$where \Omega(f) = \gamma T + \frac{1}{2} \lambda \| \omega \|^2$$

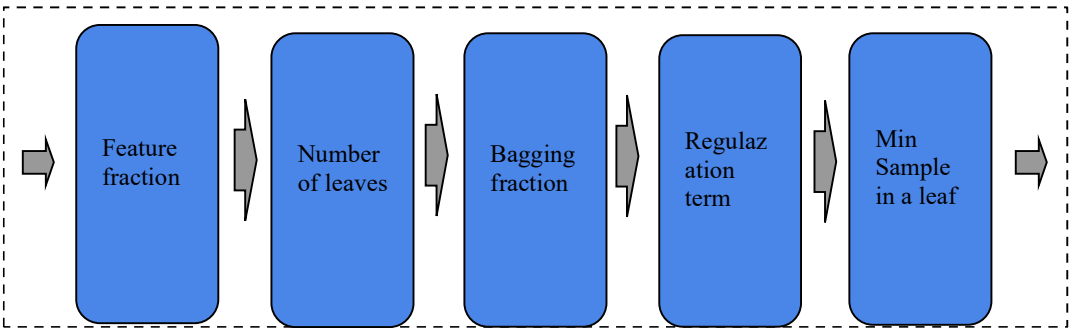
Equation 3

The l is a loss function that calculates the difference between the prediction $\hat{p}_i$ and the target p_i. The second function Ω punishes the model with a penalty when the model tries to overfit the training data. The additive parameter contributes towards the smoothing of the final weights to avoid overfitting, thus improving the accuracy of the model.

2.5.4 Light Gradient Boosting Machine (LGBM)

Microsoft introduced the light gradient boosting machine (LGBM) in April of 2017 (Abou Omar, 2018) to improve decision trees' speed. This method was developed to improve the process and implementation time by growing the tree leaf-wise instead of checking through all the previous leaves for each new leaf, as shown in Figure 5 below.

Figure 6. The processing pipeline of LGBM



Source: Adapted from Ferlitsch (2020)

All the features are sorted and grouped as bins and implemented through a technique called histogram implementation. LGBM converts category features, unlike methods such as EXGBM, which do not automatically support category features and convert the features using the OneHot encoding process. Several teams have used LGBM ML competition on Kaggle due to high accuracy, better training speed and the capability to handle large datasets and trained on GPUs (Abou Omar, 2018).

2.6 Summary

This chapter reviewed the literature on past studies that applied ML to predict poverty. It reviewed the studies by types of data used, which included satellite imagery and geospatial data, mobile phone data and survey data. It reviewed the techniques to explore those that are simpler and computationally cheaper classification methods that could be applied in low resource settings like Malawi to classify households into poor and non-poor. This information is needed to aid policy formulation, poverty alleviation and development programme intervention. The ML approaches to poverty classification that use household survey data were ranked by Fitzpatrick et al. (2018); three that were among the top performers in that study and were not resource-heavy were selected for application in this study: Logistic Regression, and Extreme Gradient Boosting and Light Gradient Boosting Machines. These were also among the most successful models in the World Bank poverty prediction competition.

CHAPTER 3
METHODOLOGY

3.1 Introduction

This chapter first introduces the off-the-shelf open-source ML tools and libraries employed in this study, as well as the dataset. It then describes the research methodology followed to achieve the research objective of the study. The study applies the ML pipeline for predictive modelling by Raschka (2015) as the research methodology.

3.2 Tools and Libraries

To develop a framework for all the models, Python was used as the programming language and development environment. Python is a high-level, open-source language governed by the Python Software Foundation and is run entirely by volunteers. Python is one of the top programming languages as of 2014 (Guo, 2014). This language is considered the first coding language in most computer science courses in colleges within the USA (Guo, 2014). Python also has an overwhelming abundance of libraries that are used for data analysis and scientific computing. The open-source libraries used in this project are:

- NumPy by Harris et. al (2020)
- SciPy, by Harris et. al (2020)
- Pandas, by McKinney et. al. (2010)
- Matplotlib, by Hunter (2007)
- Scikit-learn, by Pedregosa, Varoquaux, Gramfort, and Michel (2011)
- PyCaret by Moez (2020)
- SHAP (SHapley Additive exPlanation), by Lundberg and Lee (2017)
- Keras, by Chollet et. al (2015)
- Statsmodels, by Seabold and Perktold (2010)
- Seaborn, by Waskom (2021).

These libraries provide the bulk of pre-processing tools, ML algorithms and data visualization. All the packages are available under the free open-source license on the URL https://pypi.org/ or via pip from the Python Package Index (Pypi, 2020).

3.3 The Dataset

The study used data from Malawi's Third Integrated Household Survey (IHS3) carried out in 2010-2011, sourced through the World Bank data portal. This portal is open to the public with a creative commons license to the data and the data is anonymized to protect the respondents' identity (World Bank Group, 2020).

The dataset contained two tables, one on households and the other on individuals. The first dataset contained data on 12,244 households annotated as either "poor" if their per capita consumption was less than or equal to the poverty line of USD 1.90 per day, or "non-poor" if consumption was above the poverty line (Malawi Government, 2012). These households comprise 56,211 individuals whose data are in the second table.

The table with household data had 346 features while the table with data on individuals had 42 features.

3.4 Predictive Modelling

Raschka (2015)'s pipeline for using ML in predictive modelling was followed in the classification of households target variable as either "poor" or "non-poor". Accordingly, the ML prediction process started with (i) data pre-processing, followed by (ii) learning, that is, training and selection of a predictive model, and finally, (iii) model evaluation and data prediction.

The classifiers used in the ML pipeline were also applied by Fitzpatrick et al. (2018) to predict poverty using survey data. However, modifications were made to some of Fitzpatrick's steps to enhance the predictive models' performances, and to answer the research questions of this study. Such modifications are described in the relevant steps.

3.4.1 Dataset Pre-processing

Raschka (2015) notes data pre-processing as one of the most important steps in any ML application. Since the survey dataset came from the National Statistical Office of Malawi, they carried out the initial pre-processing and data cleaning before submitting it to the World Bank portal. This assures that the dataset is of good quality.

3.4.1.1 Data Extraction

The data was collected initially in tabular format and extracted using Stata enterprise guide software. However, for the dataset to be processed through ML models, it was converted into

a comma-separated value (CSV) format. This was achieved by the Pandas function "read_stata()".

3.4.1.2 Handling Missing Data

The IHS3 dataset was well-curated and annotated and had very few missing data values that were encoded as blanks or NaNs. They were subsequently dropped using the drop the method from the Pandas library.

3.4.1.3 Derived Features

The way poverty is calculated in Malawi is not per individual but per household. However, the individuals' dataset contains additional information useful for this study.

This stage involved using the individuals' survey data to derive features for their households. The derived features were then merged with the initial dataset on households. This was done by running python functions to produce features derived from the individuals' dataset, and another to add the derived features to the household dataset.

Examples of derived features that were obtained from the individuals' dataset and added to the initial households' dataset include: "the number of children (age 10 and under) in household"; and "the number of males in household (over 10 years old)".

3.4.1.4 One-hot Encoding of Categorical Features

While the great majority of the modern classifiers can handle categorical features, others require the inputs to be converted into numerical form. Pandas "get_dummies()" function was used for one-hot encoding, where features are created for each category and categorical values are assigned a binary value of 0 or 1 for these accordingly.

Conversion to dummy variables however can create the problem of multicollinearity, where two or more independent variables are highly linearly correlated which cause challenges on how the predictive model coefficients are calculated during training thus distorting the results. The python Pandas library function "get_dummies()" creates dummy variables from the categorical features while dropping the first dummy variable for each categorical variable to avoid multicollinearity in the dataset (Géron, 2019).

3.4.1.5 Dropping Features

Often, ML deals with data with a large number of features. This is a challenge as it not only increases the risk of high correlation among features, but also increases computational power

requirements. The creation of numerical features can further contribute to these problems (Raschka, 2015).

The next step was to remove features that were deemed not to be useful for classification so that only features with predictive value are preserved for training the ML models. The Pandas drop() function was run to remove columns with only one unique value and to remove duplicate and empty columns.

3.4.1.6 Data Exploration

After the pre-processing was complete, the dataset was explored to get a better understanding of how some of the features related to one another. Exploratory data analysis plays an important role in deciding which ML model to apply and which features to focus on. Paper (2018) explains that data analysis is not only useful for testing a predefined hypothesis but also for using the data to discover new hypotheses.

In this study, any distinct relationships between features and the target variable on poverty were explored. The data inspection was achieved by plotting the distributions of features against the poverty target, or by measuring their correlation.

First, the study mapped the numerical features which were few in Malawi's dataset, against the target variable. Next in the exploration was to look at the 212 consumables in the dataset to identify: the consumables most consumed by the poor, followed by the top items consumed by both poor and non-poor and lastly, the ones most consumed by the non-poor.

The various data exploration results are presented in section 4.3.

3.4.1.7 Feature Selection

The full panel dataset contained 486 features. The goal of the study was to compare results in prediction accuracy of the target variables when the full panel data was used and when smaller subsets of features were used. To achieve this, two different types of feature selection techniques were employed. These are the filter method documented by Zheng and Casari (2018) and the SHAP method by Shap (2020). The Filter method is the one applied by Fitzpatrick et al. (2018) while the SHAP method used in this study is a modification motivated by its common use by the top performers in the ML poverty classification competition by the World Bank as well as some of the studies reviewed in the literature review section.

a) Feature Selection Using Filter Method

Filter methods rank features by running statistical tests to calculate the correlation coefficients between input features. The Filter method was used in the classification approach by Fitzpatrick et al. (2018). In this study, a Python script "statsmodels" was run to generate statistical metrics for the data. The features are then ranked according to product of the correlation coefficients and the standard deviation of corresponding parameters in the data. In this way, the top 50 features were identified, and a data subset formed accordingly.

b) Feature Selection Using SHAP Method

The SHAP method applies mathematical functions using the Shapley values method from the game theory to determine the weight of each feature. The SHAP feature ranking method was developed by Shapley (1953) in contributing to the Game Theory and was recommended by Lundberg and Lee (2017) following their exploration in search of a feature ranking method that led to both high model accuracy and interpretability of complex models. The steps in the SHAP method are automated in the Python package SHAP. The full data panel was run through the Python Scikit ensemble package and SHAP library to generate the importance of each feature and its ranks. From the results, feature subsets were created for the top 20, 50, 100, 150, 200, 250, 300, 350, 400, and 450 features.

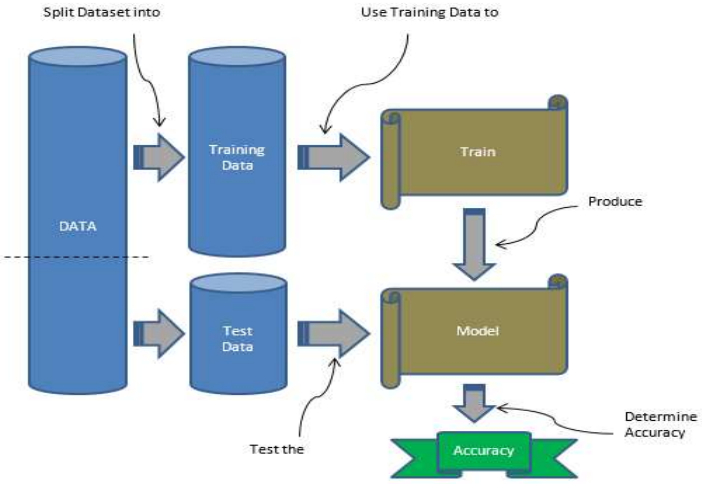
The minimum number of features of 20 was inspired by the World Bank Proxy means test which uses 20 to 30 questions to do a proxy study to determine if the household is poor or non-poor (Dupriez, 2018; Kshirsagar et al., 2017).

3.4.2 Learning: Training and Selecting Predicting Models

Two main steps are followed in supervised learning: training of prediction models and testing of their prediction ability to correctly label te target variables, in this case, poor and non-poor. The pre-labelled dataset is split into two sets at a ratio of 80:20, the larger portion being the training data, used to train pre-existing classification algorithms to produce classification models to apply to our case. The smaller portion, the test data, is then used to test the performance of the model produced in prior step, in correctly classifying the target variables. Based on the various performance metrics, the models' performances can be observed and compared, and the best ones selected.

An overview of model training in supervised learning is captured by Figure 6 below.

Figure 7. Overview of the model training and testing process in supervised learning



Adapted from Ferlitsch (2020, p. 238)

PyCaret, an open-source ML framework used was used in this study to train and evaluate the classification models and in prediction.

3.4.2.1 Model Selection

Since the aim of the study was to predict the target variable "poor" or "nonpoor", this task was a binary classification problem.

Once the data are inputted in PyCaret, the algorithms therein analyse and learn from the training data to identify the best prediction parameters, which they use to produce a classification model or classifier. The classifier is then tested on the testing dataset to determine the model's performance in correctly predicting the target variable, in this case, "poor" or "non-poor".

The PyCaret library has at least 15 classification algorithms that can be used for binary classification problems. The study pre-selected three classification models identified as the most suitable for the study, based on the review of literature in Chapter 2 on ML methods used in poverty classifications using survey data. These are the: i) Logistic Regression (LR); ii) Extra Gradient Boosting Model (ExGBM); and iii) Light Gradient Boosting Model (LGBM) algorithms. PyCaret Version 1.1. was used to train, test, and evaluate and predict labels in this binary classification.

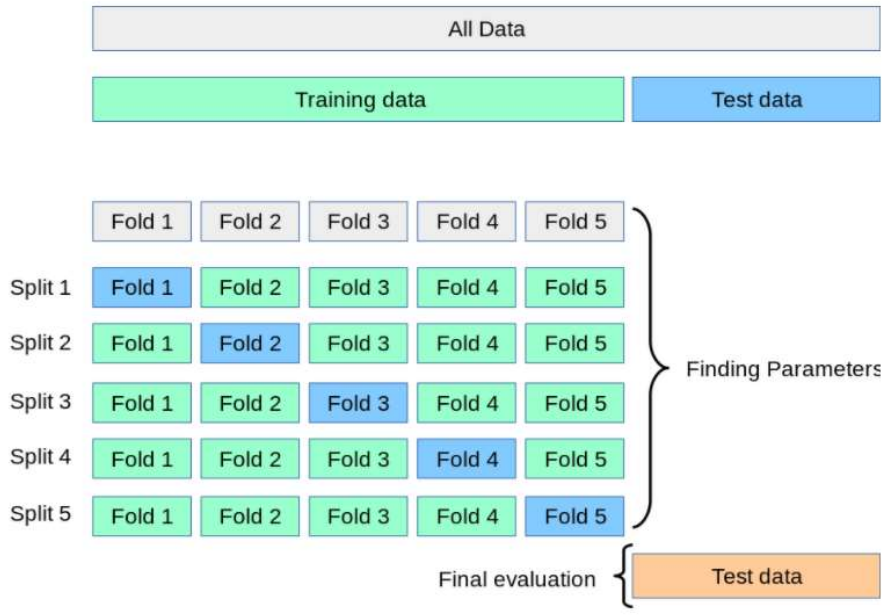
3.4.2.2 Cross-Validation

Training and testing alone are insufficient as this puts the model at risk of overfitting to the training dataset. The model only learns the labels, not the parameters or the weights of the

classification model due to training and testing using the same dataset sample. A typical example of overfitting is when a model scores highly during training but performs poorly during validation. Therefore, to avoid this, there needs to be a separate subset drawn from the dataset for model validation. This requires the dataset to be split into three parts: one for training, one for testing and the other for validating the model. However, by splitting the data into three parts, the number of samples available for the algorithm to train from is considerably reduced in impacting a model's performance.

Cross-validation provides a means to safeguard from model overfit. This is achieved by expanding the number of samples for training and testing the model while at the same time reserving the needed data for validation. A testing set is still reserved for the final evaluation; however, the training dataset is split into *k* smaller sets or folds. The model is then trained on *k* - 1 folds of the data and tested on the remaining portion of the testing dataset. This process is repeated *k* number of times, and the model performance values reported are the average of the performance values observed in each loop. The process is thus termed k-fold cross-validation, illustrated in Figure 7 below.

Figure 8. Model training and testing using cross-validation



Source: SciKit Learn (2021, p. 33)

In this study, 10-fold cross-validation is used, given the relatively small data set of 12,244. The performance of classifiers generated by LR, LGBM and ExGBM were recorded. PyCaret is also used to carry out cross-validation.

3.4.2.3 Performance Metrics

In classification, there are several metrics that can used to evaluate the performance of a model. These include (i) accuracy; (ii) area under the curve (AUC); (iii) Recall; (iv) Precision; v) F1-measure; vi) Kappa; and vii) Matthew's correlation coefficient (MCC). PyCaret reports the performance of the trained models using all these metrics. The metrics are described below.

a) Confusion Matrix

A confusion matrix is a tool that summarises the performance of a classification model in correctly predicting the target variable classes, in this case, the positive class "poor" and negative class "non-poor". For any classification model, the bins in the matrix capture the aggregated true predictions (true positives and true negatives) and the wrong predictions or misclassifications (false positives and false negatives). Table 2 provides the illustrative form of the confusion matrix for a binary classifier.

Table 2. The confusion matrix

Predicted→ Actual↓	**Poor**	**Non-Poor**
Poor	TP	FN
Non-Poor	FP	TN

Where: TP = True positive; FP = False positive; FN = False negative; TN = True negative.

All the performance metrics used to evaluate the performance in this study which are described below can be deduced from the confusion matrix (Tiziana Rancati, 2019).

b) Accuracy

Accuracy is the most frequently used metric in classification (Raschka, 2015; Géron, 2019). In simple terms, accuracy reports the portion or percentage of correct predictions, mathematically expressed as below.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}}$$

Equation 4

Where TP = true positive (i.e., poor household); TN = true negative (i.e., non-poor house); FP = False positive and FN = false negative.

Accuracy can be relied on as a singular performance metric where a dataset is well balanced. However, this metric is very sensitive to imbalanced datasets, where one class is much more dominant than the others. In such cases, relying on accuracy alone as a model performance measure can be misleading (Fitzpatrick et al., 2018; Géron, 2019). This metric is therefore often used in combination with the other performance metrics and model performance judged based on the overall results.

As the dataset used in this study is well balanced with 45.1% poor and 54.9% non-poor households, following the reports of Fitzpatrick et al. (2018) and (Géron, 2019), accuracy is considered a key performance metric in this study. Nonetheless, for purposes of robustness, the models' performance in terms of other metrics is also considered.

c) Recall

Recall focuses on the portion of the dataset that is positive. It is therefore the rate at which the classifier was able to correctly predict the true positive. It is the ratio of true positives (TP) to all the positives, which is both the TPs and FNs. The equation representation is provided below.

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

Equation 5

A good classifier is one that has few FNs, and therefore a high true positive rate (TPR). Recall is particularly helpful to gauge model performance where a dataset is imbalanced (Fitzpatrick et al., 2018). In this study, low recall would mean too many poor households do not receive aid.

d) Precision

Precision is defined as the positive predictive value (PPV) (Fitzpatrick et al., 2018; Zheng, 2015). It looks at the totality of what the model identified as positive, and how many of those were truly positive. The equation for precision is therefore the one below, where FP = False positives.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

Equation 6

The higher the number of false positives, the lower the model's precision. Precision is therefore a counterbalance to recall, and it is helpful to look at the two metrics together. In this study, poor precision would mean that aid is targeted at too many households that do not need it.

e) F1 Measure

The F1 score combines precision and recall providing a "harmonic mean". It provides a convenient way to look at precision and recall together (Fitzpatrick et al., 2018; Géron, 2019; Zheng, 2015). Fitzpatrick et al. (2018) represent F1 as follows.

$$\text{F1} = \frac{2\text{TP}}{2\text{TP} + \text{FN} + \text{FP}}$$

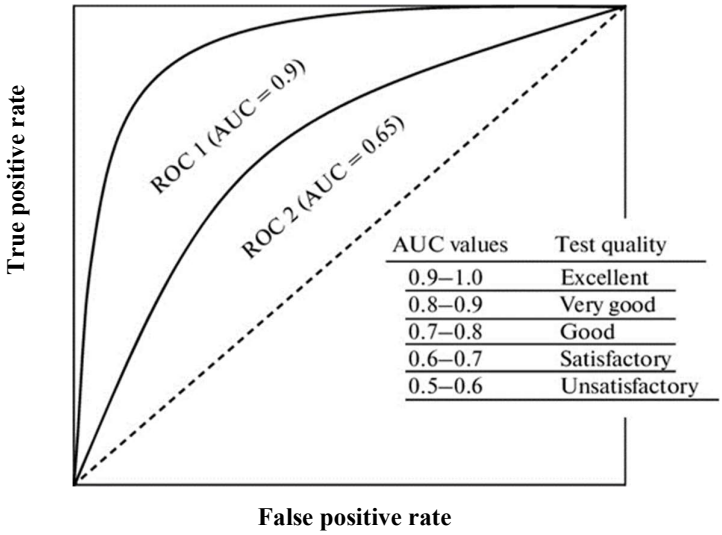
Equation 7

A classifier gets a high F1 score if both recall, and precision are high.

f) Area Under the Curve

Receiver Operating Characteristic (ROC) Curve is a performance metric representable in graphic form, displaying the relationship between the True Positive Rate (TPR) and the False Positive Rate (FPR) classes of prediction. It is among the most used tools to gauge the performance of ML classifiers, especially visually (Davis & Goadrich, 2006) and (Fitzpatrick et al., 2018). The ROC is constructed by plotting the true positive rate (or Recall) against the false positive rate. A single point on the ROC curve, the TPR and the associated FPR, is obtained from the confusion matrix of a classifier. A high AUC value represents a high TPR and low FPR. Figure 8 provides a guideline to interpreting the ROC curve and AUC values.

Figure 9. Guideline to interpret receiver operating characteristic curve based on area under the curve values



Source: Zheng (2015, p. 15)

A perfect classifier will align in the top left-hand corner, signalling False Positive Rate (FPR) = 0 and a True Positive Rate (TPR) = 1. A worst-case classifier will score in the bottom right-hand corner where FPR = 1, TPR = 0. A random classifier would be expected to score somewhere along the positive diagonal (the dotted line, where TPR=FPR) signalling that the model will generate false positive and false negative predictions at the same rate.

g) Kappa

Cohen's Kappa κ tries to measure the influence of chance on a model's accuracy. It does this by comparing the accuracy achieved and expected accuracy (Cohen , 1968). Fitzpatrick et al. (2018) define expected accuracy as "the accuracy rate that a random classifier would be expected to achieve based on confusion matrix of predictions". Accuracy (A) is as measured in equation 7 above. Expected accuracy (EA) is represented by the following equation.

$$\mathrm{EA} = \frac{(\mathrm{TP} + \mathrm{FN})(\mathrm{TP} + \mathrm{FP}) + (\mathrm{TN} + \mathrm{FP})(\mathrm{FN} + \mathrm{TN})}{\mathrm{TP} + \mathrm{FN} + \mathrm{FP} + \mathrm{TN}}$$

Equation 8

The Kappa statistic is calculated as follows:

$$\kappa = \frac{A - EA}{1 - EA}$$

Equation 9

A Kappa score of 1 is the perfect score, representing complete agreement between the model's performance score and the ground truth, whereas a negative score indicates chance agreement. A kappa of zero means all the accuracy achieved was by chance.

h) Mathew's Correlation Coefficient

This measure of classification performance was developed by a biochemist, Brian Mathews, in 1975 for comparing chemical structures (Chicco & Jurman, 2020). This metric was repurposed for ML by Baldi (2000) in the year 2000 to measure classification performance for datasets that have unbalanced target labels or classes. This method treats the true class and predicted class of the confusion matrix as two binary variables and it calculates their correlation coefficient. The lower the correlation between the true and predicted values, the lower the classification performance. Computing the Mathews Correlation Coefficient (MCC) involves solving the phi coefficient (φ): The MCC or phi coefficient can be derived from the formula below.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

Equation 10

The MCC score represents the correlation between the predicted positives and the true positives. When the classifier MCC = 1, the classifier performance is considered perfect and if it is -1 is considered flawed. A negative number represents a negative correlation between predicted and true negatives.

3.4.2.4 Hyperparameter Optimization

Raschka (2015) mentions the two types of parameters in ML: those learned from training data, such as weights in LR; and the parameters of a learning algorithm, also called hyperparameters of a model, e.g., the depth parameter in decision trees and regularization in LR. Hyperparameters can be and are used to improve model performance.

The default parameters of pre-existing learning algorithms available from ML libraries are not ideal for every ML problem task. Hyperparameter optimization techniques are thus employed to help to fine-tune predictive models. Grid search is one such technique that is powerful and helps to improve performance model by finding the optimal combination of hyperparameter values (Raschka, 2015).

Pycaret by default extracts the optimal configuration parameters for tree-based classifiers and LR classifiers by obtaining the maximum number of features to split into nodes, the Gini-index and the number of trees in a forest, the solvers to be used and regularization penalty for LR classifiers. The performance and impact of the hyperparameter is measured against the standard performance metrics such as accuracy, recall, precision, f1-score, misclassification error, Out-of-bag (OOB) error and confusion matrix. Improvements made to model performance are included in the performance results reported for each predictive model (Moez, 2020).

Pycaret was used in this study to automatically optimize and tune the hyperparameters of all the three algorithms using the random grid search method. This was achieved by using the Pycaret library function "the *tune_model()*".

3.4.3 Evaluation and Prediction

Based on the metrics explained above, model performance in correctly predicting poor and non-poor households can be evaluated. In this study, the perfomance results measured in terms of the eight metrics are recorded for the full 486 feature-panel, and on the smaller feature subsets of the top 450, 400, 350, 300, 250, 200, 150, 100, 50 and 20 features selected using both the Filter and SHAP methods.

In Raschka (2015)'s ML workflow diagram, the fourth and final step is model prediction. If satisfied with the models' performance in the third step, trained models are applied on the unseen test data, prediction is carried out on new, future data.

In this study, the test data from the same IHS3 survey data acts as a proxy of new unseen data. Hence prediction is carried out on the portion of the IHS3 data which was reserved for model testing.

3.5 Summary

This chapter has described the steps and methods followed to answer the study's research questions. These include the data pre-processing procedures; the two approaches to feature

selection applied; the approach to model training and testing; and a description of the metrics used to evaluate the performance of the classifiers which were built from three classification algorithms. Steps taken to explore key features to better understand the dataset were explained. The results drawn from the execution of the process are provided and discussed in the next chapter.

CHAPTER 4
RESULTS AND DISCUSSION

4.1 Introduction

This chapter presents the results obtained following the steps outlined in chapter 3 to achieve the aim of the study by answering the study's research questions of whether poverty can be predicted using off-the-shelf ML tools employed using survey data; and if poverty can be predicted based on a smaller number of features with acceptable results. Provided here are the outcomes of data pre-processing, data exploration, feature selection, and model training and evaluation. The results obtained at each stage are discussed. The chapter concludes by returning to the two research questions to determine their answers based on the results obtained.

4.2 Dataset Pre-processing Outcomes

The survey data once converted from Stata to CSV format was explored. The data included some numerical values, categorical string values and some missing data points. Two target classes were well balanced at 47.1% poor and 52.9% non-poor. The household data had 346 features while the data on individuals had 42 features.

The drop method from Pandas library dropped the 8 rows with missing data. This number was low compared to the remaining 12,244 rows, therefore causing no concern about the loss of valuable information and compromise to data integrity.

The manually created python functions produced derived features from the individuals' dataset and added the derived features to the dataset. The features' derivation was performed to capture additional information from the data on individuals useful for household classification.

The Python Pandas library "get_dummies()" function when ran created dummy variables from the categorical features in the dataset. With the dummy variables, the aggregated dataset contained 816 features.

Duplicate columns, empty columns, and columns with only one unique value were dropped by running Pandas drop functions. After dropping, the number of features was reduced from 816 to 486. Of the 486 features, eight were numerical value features and the rest were categorical.

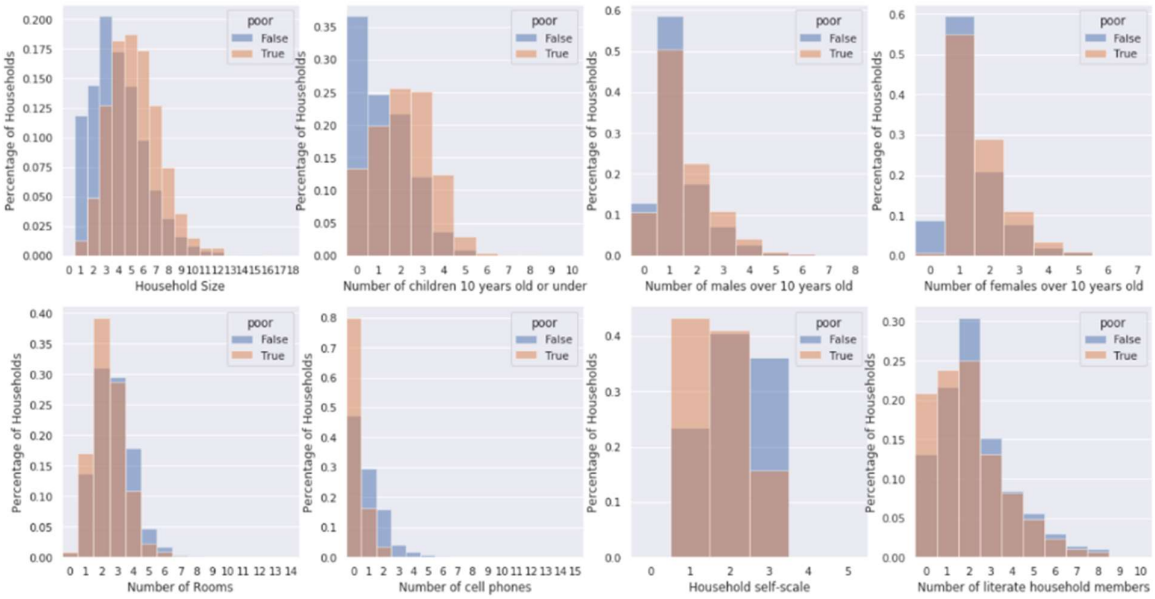
4.3 Data Exploration

The observations from the data exploration are presented below.

4.3.1 Exploratory Results for Numerical Features

The numerical features in Malawi's dataset were few. Exploratory results for all eight numerical features are presented in the histograms in Figure 9 below.

Figure 10. Distribution of numerical features across poor and non-poor classes



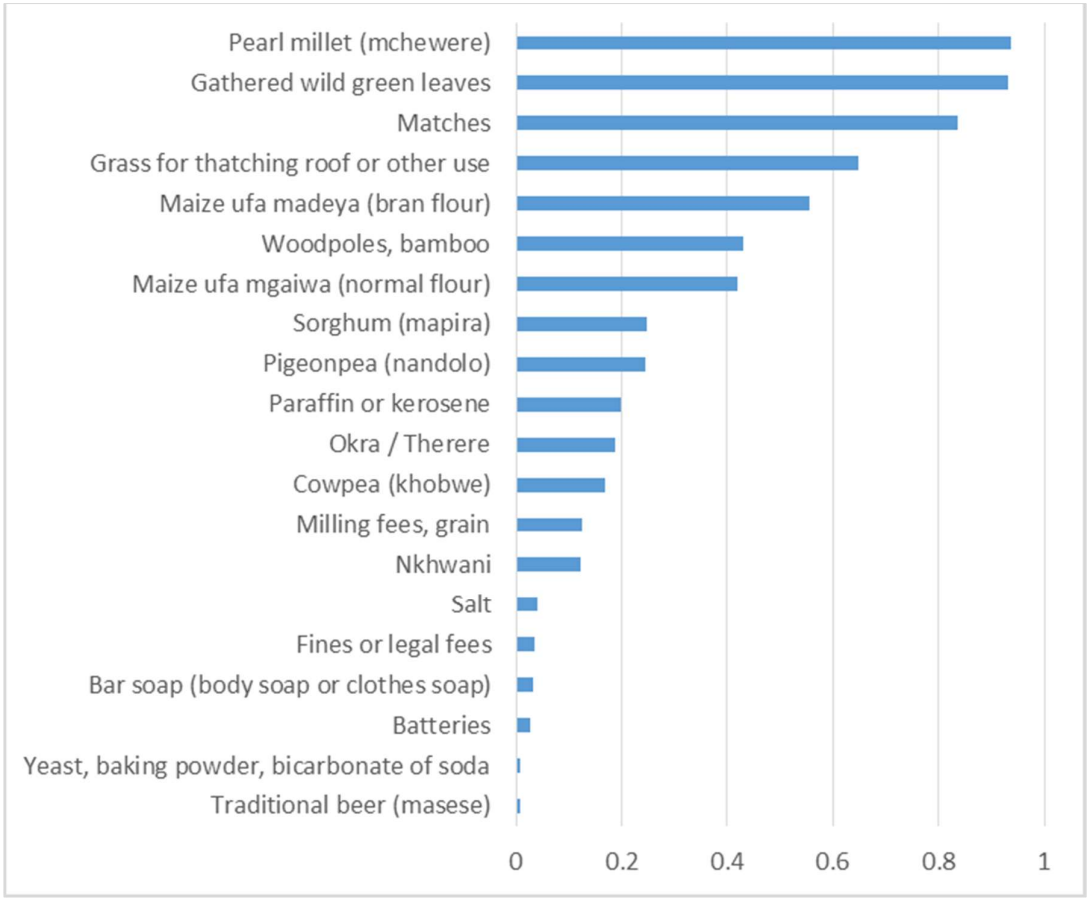
The histograms above reveal that non-poor households tend to have smaller household sizes, with fewer children under the age of 10 years. Also, the non-poor appear to have more rooms in their houses, a greater number of cell phones and a higher number of literate household members.

Overall, the histograms show the numeric data does not provide much distinction between poor and non-poor households.

4.3.2 Exploratory Results for Top Consumables for the Poor

It was important to identify the consumables most likely to be associated only with the target variable "poor". Items classified as "other" were also excluded due to their limited explanatory value. Figure 10 below shows the top 20 items identified as the most consumed by poor households.

Figure 11. Goods most consumed by poor households



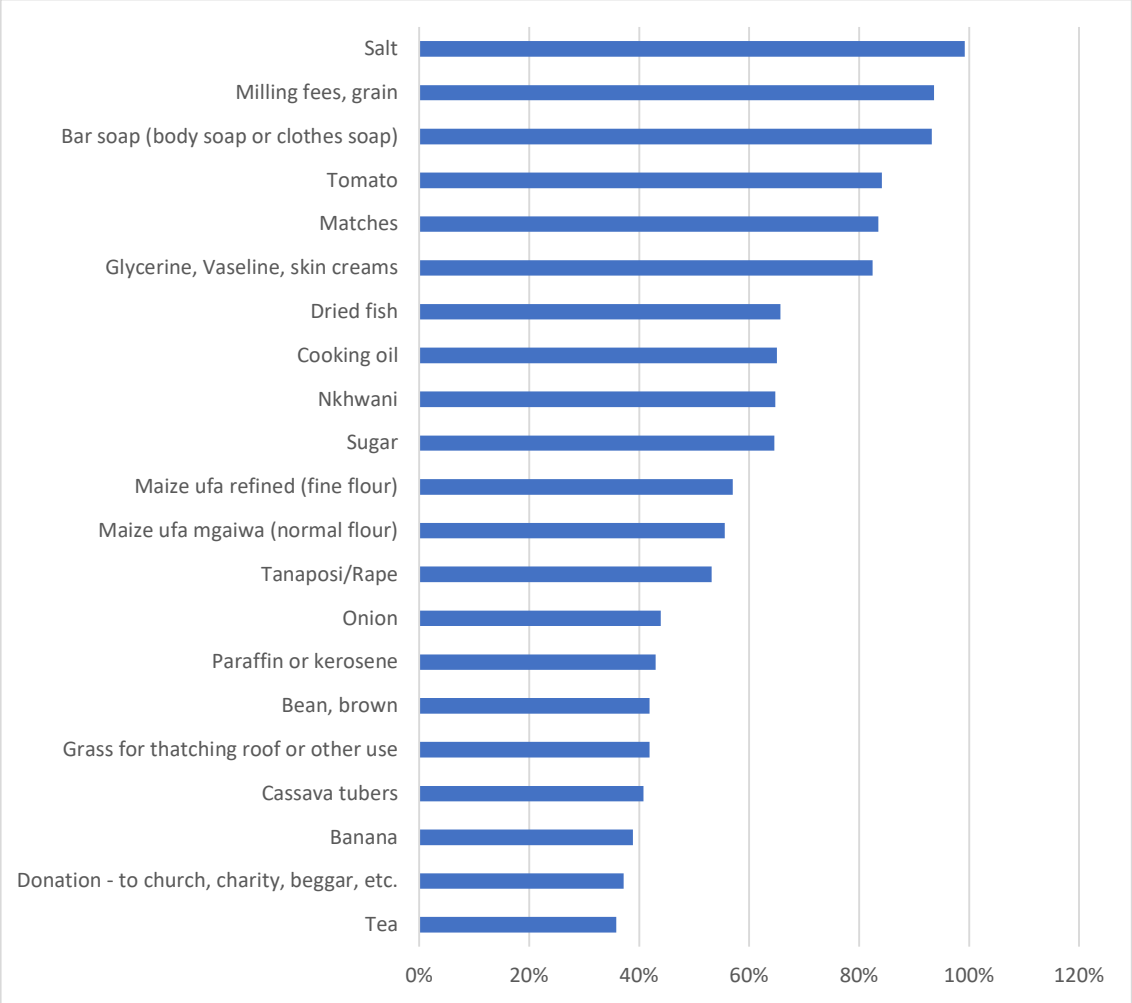
Fraction of poor households consuming items most used by the poor

Top consumables by the poor include pearl millet, gathered wild green and matches. While this is the case, to identify goods most likely to be associated with poor, it would be useful to drop features common to both poor and non-poor.

4.3.3 Exploratory Results for Top Consumables Common to both the Poor and Non-poor

It was also essential to understand the consumables consumed by both the poor and non-poor households as they would not have much value in explaining the target variable. Figure 11 below shows the distribution of the top 20 most popular consumables common to both the poor and non-poor.

Figure 12. Top 20 consumables



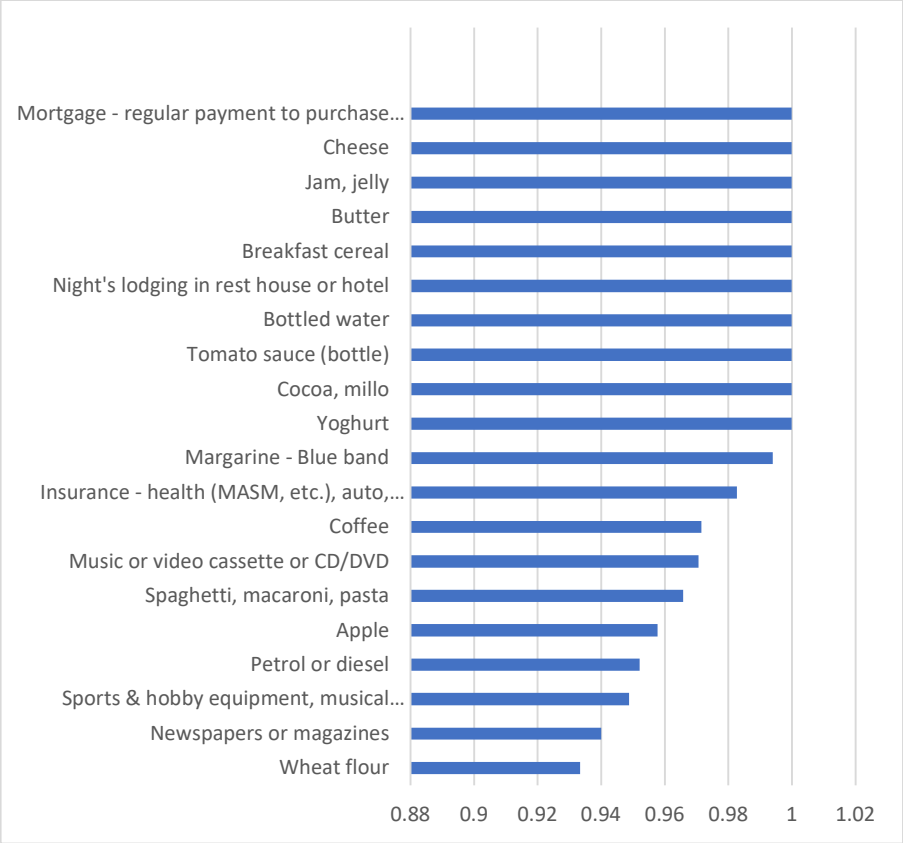
Percentage of households consuming the most common items

Of the total sample of 12,244, 99.2% of the households reported consuming salt, 93.6% pay milling fees for grain and 93.2% use soap. As these sets of features are very commonly consumed regardless of poverty or wealth, they are likely not useful in the differentiation between poor and non-poor.

4.3.4 Exploratory Results for Top Consumables by the Non-poor

The study also explored the consumables consumed by non-poor households more frequently than poor households. The aim was to deduce the features most likely to be associated with the non-poor alone that could be useful to identify this class of target variable. Figure 12 below shows the top 20 items consumed by the non-poor.

Figure 13. Top 20 items consumed by the non-poor



Fraction of consumers who are non-poor households

There were 10 features consumed exclusively (100%) by the non-poor. The other 10 of the top 20 features were predominantly consumed by the non-poor. The consumption of these goods could be translated as a good indicator of "non-poor" status. Of these items, the top 3 which were among those exclusively consumed by the non-poor, were mortgage payments, consumption of cheese, and jam or jelly.

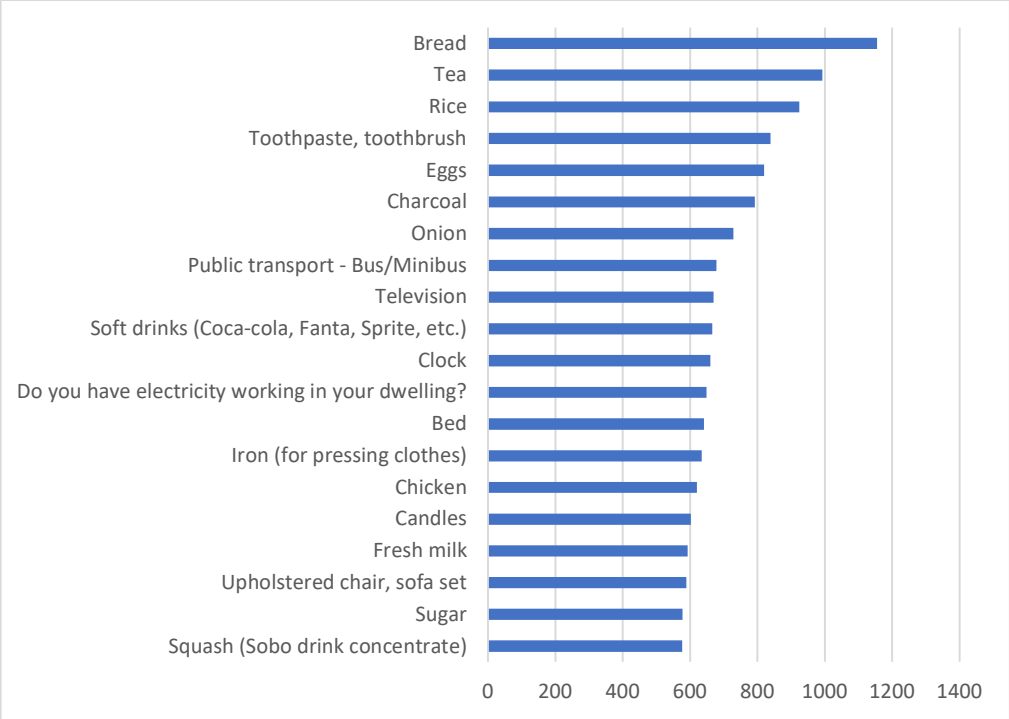
4.4 Feature Selection Results

Running the two feature selection techniques produced two ranking lists of the 486 features. The top 20 features selected by the Filter and the SHAP methods are presented below.

4.4.1 Top Features Selected by the Filter Method

Running the dataset through the Filter methods ranked the 486 in order of their explanatory values in predicting the variables poor and non-poor basing on their chi-square values. Figure 13 below shows the top 20 features which according to the Filter method, have the highest explanatory value for the target variables.

Figure 14. Top 20 features selected by the Filter method

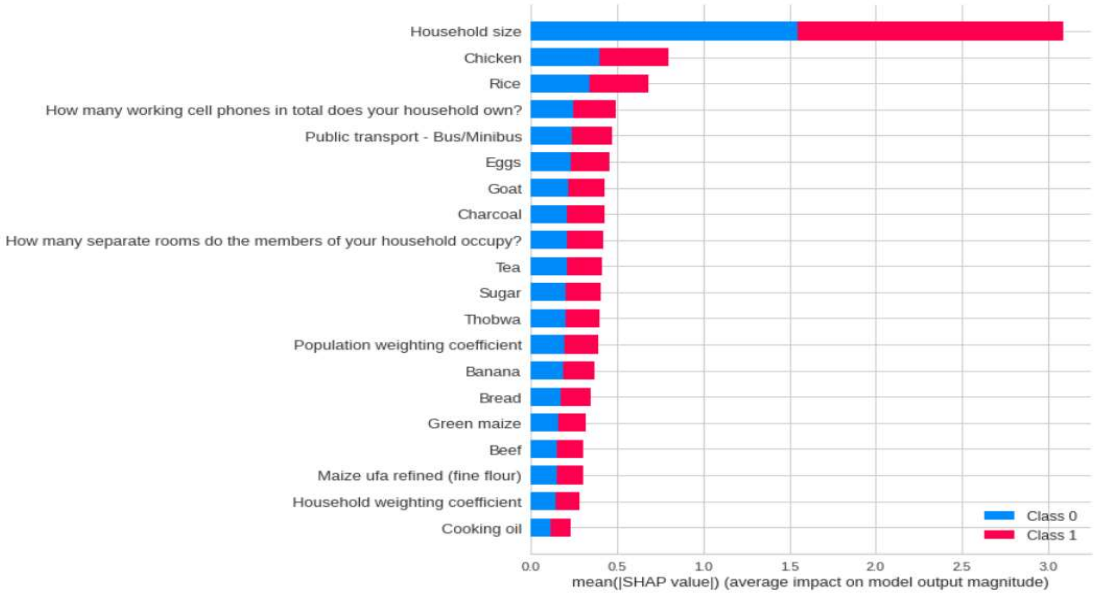


The Filter method identifies the consumption of bread, tea and rice to be among the features having the highest contribution to identifying whether a household is poor or non-poor.

4.4.2 Top Features Selected by the SHAP Method

The SHAP method also generated its ranking of the 486 features according to their level of contribution to determining whether a household is poor or non-poor. The features' level of contribution is calculated and presented in form of SHAP values. Figure 14 below shows the 20 features with the highest mean SHAP values, and therefore the top 20 explanatory features according to the SHAP method.

Figure 15. Top 20 explanatory features selected by SHAP method



Class 0 = non-poor; Class 1 = poor

According to the SHAP ranking, the household size had the most influence in explaining the status of the wealth of a household. This was followed by the consumption of chicken and rice.

The full top 50 features as ranked by SHAP method is as follows: 1) Household size; 2) Chicken; 3) Rice; 4) Number of working cell phones in the household ; 5) Public transport – bus/minibus; 6) Eggs; 7) Goat; 8) Charcoal; 9) Number of separate rooms in household; 10) Tea; 11) Sugar; 12) Thobwa; 13) Population weight coefficient; 14) Banana; 15) Bread; 16) Green maize; 17) Beef; 18) Maize ufa refined (fine flour); 19) Household weight coefficient; 20) Cooking oil; 21) Pumpkin; 22) Past 7 days, number of people not listed as household members having eaten meal in the household ; 23) Fresh milk ; 24) Toothpaste, toothbrush 25) Groundnut; 26) Irish potato; 27) Television; 28) Working electricity in dwelling place; 29) Other personal products (shampoo, razor blades, cosmetics, hair products, etc); 30) Dried fish; 31) Radio ("wireless") ; 32) Buns, scones; 33) Donations – to church, charity, beggar, etc; 34)Cabbage; 35) Fresh fish; 36) Soft drinks (Coca cola, Fanta, Sprite, etc); 37) Iron (for pressing cloths); 38) Cash transfers / gifts; 39) Grass for thatching roof or other use; 40) Bean, brown ; 41) Bed; 42) Pork; 43) Mortar/pestle (mtondo); 44) Groundnut flour 45) Orange sweet potato; 46) Watering can; 47) Presence of larger weekly market in the community ; 48) Bean, white; 49) Squash (Sobo drink concentrate) ; and 50) Candles.

Notably, the ranking and top features selected by the Filter and SHAP methods are different. The classification model performance results generated on smaller subsets of data using the Filter Method and SHAP method rankings are presented in the next section.

4.5 Model Training and Evaluation Results

4.5.1 Model Evaluation Results on the Full Panel of Features

The primary experiment was run on the full panel of 486 features using 10-fold cross-validation to train the models, and the reserved portion of the dataset to test the models, and the performance of the LR, ExGBM and LGBM classifiers was observed. The performance results of the 3 classifiers on the training and test data sets are presented in Tables 3 and 4 respectively.

Table 3. Model performance on the full panel of features in the training phase

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
0	Light Gradient Boosting Machine	0.8721	0.9500	0.8584	0.8592	0.8585	0.7418	0.7423	2.6025
1	Extreme Gradient Boosting	0.8666	0.9461	0.8509	0.8536	0.8521	0.7307	0.7309	26.5143
2	Logistic Regression	0.8608	0.9413	0.8563	0.8390	0.8474	0.7195	0.7198	2.8290

Table 4. Model performance on the full panel of features in the testing phase

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
lightgbm	Light Gradient Boosting Machine	0.8674	0.9414	0.8527	0.8553	0.8540	0.7326	0.7326	0.5900
lr	Logistic Regression	0.8669	0.9425	0.8545	0.8530	0.8537	0.7316	0.7316	6.8500
xgboost	Extreme Gradient Boosting	0.8634	0.9423	0.8521	0.8480	0.8501	0.7246	0.7246	5.1100

The LGBM classifier was evaluated as the best performer on the full feature set, with the highest accuracy score of 87.21% in the model training phase, and again in the testing phase with an accuracy score of 86.74 %. The ExGM and LR classifiers had relatively close accuracy scores of 86.66% and 86.08% respectively on training data and accuracy scores of 86.34% and 86.69% respectively on test data.

While the LGBM classifier was also the top scorer in terms of the other six performance metrics in the training phase, LR performed slightly better in terms of AUC and Recall. While that is

the case, the very high AUC scores ranging from 94.23% - 95% for all three models in both the training and testing phases representing very high true positive rates and low false positive rates.

It also had the highest Recall score of 85.54% representing the rate at which the classifier correctly predicted poor households. The high precision score of 85% shows that the rate at which the classifier wrongly predicts non-poor households as poor households is low. The near-identical accuracy and precision performances are reflected in the proportionately high harmonic mean of 85.85% denoted as the F1 score. The LGBM classifier also had the highest Kappa score indicating it was the least influenced by chance. Also having the highest MCC coefficient infers the highest correlation between the true and predicted values.

Overall while the LGBM classifier had the best performance, the performances of ExGM and LR classifiers were relatively close.

4.5.2 Model Performance on Feature Subsets Extracted Based on the Filter Method Ranking

The same experiment was run on the dataset of top 50 features selected by the Filter method using both cross validation training data and holdout test data, and the performance results of the LR, ExGBM and LGBM observed. The results are presented in Tables 5 and 6 below.

Table 5. Performance results on Filter method's Top 50 features' training data

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa
0	Extreme Gradient Boosting	0.6399	0.6978	0.6415	0.5951	0.6172	0.2782
1	Logistic regression	0.6356	0.6911	0.6236	0.5931	0.6078	0.2679
2	Light Gradient Boosting Machine	0.6277	0.6824	0.6171	0.5846	0.6002	0.2524

Table 6. Performance results on Filter method's Top 50 features' test data

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
lightgbm	Light Gradient Boosting Machine	0.6367	0.6847	0.6252	0.5898	0.6069	0.2697	0.2701	0.2700
lr	Logistic Regression	0.6361	0.6880	0.6258	0.5889	0.6068	0.2687	0.2692	3.0000
xgboost	Extreme Gradient Boosting	0.6350	0.6813	0.5945	0.5930	0.5938	0.2623	0.2623	2.1200

While the results were encouraging, they were not sufficiently accurate. The ExGBM classifier was the best performing model using the training cross validation dataset. While the LGBM

had the accuracy of 0.6367 using the test data. However, the highest accuracy levels achieved dropped substantially from 87.21% to 63.99% for the training data and 63.61% for the test data. There was also a drop in the other performance scores achieved with the highest AUC score dropping from 0.95 to 0.6978; highest Recall from 0.8584 to 0.6415; highest Precision from 0.8592 to 0.5951; highest F1 from 0.8585 to 0.6172; and highest Kappa score from 0.7418 to 0.2782.

The experiments using any smaller data subsets extracted using the filter method was thus discontinued. Next, the study looked at the performance of smaller data subsets extracted using the SHAP method.

4.5.3 Accuracy Results on Ten Feature Subsets Extracted Based on the SHAP Method Ranking

Data subsets were created from the top 450, 400, 350, 300, 250, 200, 150, 100, 50 and 20 features as ranked by the SHAP method. The same experiment of 10-fold cross-validation was run sequentially on each of the subsets. The accuracy results obtained by the three pre-selected classification algorithms namely LR, ExGBM and LGBM on the datasets are presented in Tables 7 and 8 below.

Table 7. Accuracy results on training data of feature subsets extracted using SHAP method

No. of features	LR	LGBM	ExGBM
486	0.8608	0.8721	0.8666
450	0.8775	0.8690	0.8615
400	0.8774	0.8690	0.8614
350	0.8775	0.8689	0.8615
300	0.8779	0.8694	0.8615
250	0.8771	0.8700	0.8615
200	0.8781	0.8713	0.8608
150	0.8742	0.8705	0.8616
100	0.8727	0.8653	0.8593
50	0.8623	0.8546	0.8550
20	0.8356	0.8341	0.8285
10	0.7289	0.7172	0.7156

Table 8. Accuracy results on test data of features subsets extracted using SHAP method

No. of Features	LR	LGBM	ExGBM
486	0.8707	0.8677	0.8620
450	0.8789	0.8715	0.8655
400	0.8789	0.8811	0.8732
350	0.8778	0.8704	0.8685
300	0.8685	0.8702	0.8683
250	0.8704	0.8585	0.8593
200	0.8745	0.8718	0.8658
150	0.8655	0.8683	0.8593
100	0.8694	0.8601	0.8604
50	0.8625	0.8579	0.8465
20	0.8381	0.8326	0.8310
10	0.7334	0.7344	0.7336

For the smaller feature sets, the LR classifier provided the largest number of the highest accuracy scores, followed by LGBM. Nonetheless once again, the scores by all three classifiers were relatively close. The best accuracy scores for the smaller subsets stayed above 87% until the dataset size was reduced to 50 which is where the accuracy dropped to 86.23% and dropped further to 83.41 when the data subset size was reduced to 20, and 73.44% when the dataset was reduced to 10%.

Based on the accuracy scores, this suggests a positive finding to the research question of whether a smaller number of features can be used to predict if a household is poor or not poor. The next section gives details of model performance using the top 50, top 20 and top 10 features.

4.5.4 Model Performance on SHAP Method's Top 50 Features

The performance results of the three classifiers on the subset of SHAP method's top 50 features using training data and testing data are presented in Tables 9 and 10 below.

Table 9. Performance of the three classifiers on SHAP's top 50 features training data

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
0	Logistic Regression	0.8623	0.9421	0.8524	0.8445	0.8482	0.7223	0.7226	0.2396
1	Extreme Gradient Boosting	0.8550	0.9360	0.8431	0.8372	0.8400	0.7074	0.7077	1.5369
2	Light Gradient Boosting Machine	0.8546	0.9375	0.8392	0.8391	0.8391	0.7065	0.7066	0.3253

Table 10. Performance of the three classifiers on SHAP's top 50 features test data

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
lr	Logistic Regression	0.8625	0.9399	0.8546	0.8439	0.8492	0.7229	0.7230	1.4000
lightgbm	Light Gradient Boosting Machine	0.8579	0.9366	0.8456	0.8415	0.8435	0.7134	0.7134	0.2300
xgboost	Extreme Gradient Boosting	0.8465	0.9308	0.8347	0.8278	0.8312	0.6905	0.6905	1.6200

For SHAP method's top 50 features, the LR classifier was the best performing model. Again though, overall, the performance of all three models was very close.

There only a slight drop in accuracy of the best performing model of about 1 percentage point from 87.21% for the full panel set to 86.23% for the training dataset, and from 87.07% to 86.25% for the test data. Compared with results using the full feature set, there were also very slight changes in the AUC score from 95% to 94.21% for the training dataset, and from 94.25% to 93.99% for the test data; Recall from 85.84% to 85.24% for the training dataset, and from 85.45% to 85.46% for the test data; Precision from 85.92% to 84.45% for the training dataset, and from 85.45% to 84.39% for the test data; F1 from 85.85% to 84.82% for the training dataset, and from 85.40% to 84.92% for the test data; Kappa score from 74.18% to 72.23% for the training dataset, and from 73.26.% to 72.29% for the test data; and MCC from 74.23% to 0.7226 for the training dataset, and from 73.26% to 72.30% for the test data.

The classification model's performance on SHAP method's top 50 features was thus very close to that on the full feature set.

4.5.5 Model Performance on SHAP Method's Top 20 Features

The performance results of the three classifiers on the subset of SHAP method's top 20 features are presented in Tables 11 and 12 below.

Table 11. Performance of the three classifiers on SHAP's top 20 features training data

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
0	Logistic Regression	0.8356	0.9163	0.8191	0.8173	0.8181	0.6681	0.6683	0.4770
1	Light Gradient Boosting Machine	0.8341	0.9148	0.8211	0.8134	0.8171	0.6653	0.6656	0.4753
2	Extreme Gradient Boosting	0.8285	0.9090	0.8196	0.8044	0.8117	0.6542	0.6547	2.3923

Table 12. Performance of the three classifiers on SHAP's top 20 features test data

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
lr	Logistic Regression	0.8381	0.9202	0.8242	0.8228	0.8235	0.6739	0.6739	0.4300
lightgbm	Light Gradient Boosting Machine	0.8326	0.9160	0.8177	0.8172	0.8175	0.6629	0.6629	0.1700
xgboost	Extreme Gradient Boosting	0.8310	0.9107	0.8278	0.8081	0.8178	0.6602	0.6604	1.1300

For the top 20 features, the LR classifier was the best performing model. As with the top 50 feature set, the exception was the recall metric where LGBM scored the highest on the training data and ExGBM on the test data. Again however, overall, the performance of all three models was very close.

There was a percentage drop between 3 and 4 in accuracy of the best performing model from 87.21% for the full panel set to 83.56% for the training dataset, and from 87.07% to 83.81% for the test data. Compared with results using the full feature set, there were also small but higher degrees of change in the AUC score from 95% to 91.63% for the training dataset, and from 94.25% to 92.02% for the test data; Recall from 85.84% to 81.91% for the training dataset, and from 85.45% to 82.78% for the test data; Precision from 85.92% to 81.73% for the training dataset, and from 85.45% to 82.28% for the test data; F1 from 85.85% to 81.81% for the training dataset, and from 85.40% to 82.35% for the test data; Kappa score from 74.18% to 66.81% for the training dataset, and from 73.26.% to 67.39% for the test data; and MCC from 74.23% to 66.83 for the training dataset, and from 73.26% to 67.39% for the test data.

This represents an overall 3 to 4-percentage drop in accuracy, AUC, Recall, Precision and F1, and a 6-percentage drop in the Kappa and MCC scores signalling this level of increase in randomness in correctly predicting the target variables.

4.5.6 Model Performance on SHAP Method's Top 10 Features

The performance results of the three classifiers on the subset of SHAP method's top 10 features are presented in Tables 13 and 14 below.

Table 13. Performance of the three classifiers on SHAP's top 10 features training data

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
lightgbm	Light Gradient Boosting Machine	0.7344	0.8010	0.7777	0.6770	0.7236	0.4704	0.4750	0.2460
xgboost	Extreme Gradient Boosting	0.7336	0.7976	0.7822	0.6745	0.7242	0.4693	0.4744	0.8310
lr	Logistic Regression	0.7334	0.8033	0.7965	0.6701	0.7276	0.4703	0.4774	0.7370

Table 14. Performance of the three classifiers on SHAP's top 10 features test data

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
lr	Logistic Regression	0.7289	0.7940	0.7881	0.6770	0.7283	0.4609	0.4662	0.1600
lightgbm	Light Gradient Boosting Machine	0.7172	0.7852	0.7621	0.6700	0.7131	0.4366	0.4401	0.2500
xgboost	Extreme Gradient Boosting	0.7156	0.7799	0.7639	0.6674	0.7124	0.4336	0.4375	1.7900

For SHAP method's top 10 features, LR and LGBM alternated on the highest performance metrics achieved for the training dataset, while the LR classifier was the best performing model for the test dataset. Again, overall, the performance of all three models was very close.

The percentage drop in accuracy of the best performing model was from 87.21% for the full panel set to 73.44% for the training dataset, and from 87.07% to 72.89% for the test data. Compared with results using the full feature set, there were also higher degrees of change in the AUC score from 95% to 80.33% for the training dataset, and from 94.25% to 79.40% for the test data; Recall from 85.84% to 79.65% for the training dataset, and from 85.45% to 78.81% for the test data; Precision from 85.92% to 67.70% for the training dataset, and from 85.45% to 67.70% for the test data; F1 from 85.85% to 72.76% for the training dataset, and from 85.40% to 72.83% for the test data; Kappa score from 74.18% to 47.04% for the training dataset, and from 73.26.% to 46.09% for the test data; and MCC from 74.23% to 47.74 for the training dataset, and from 73.26% to 46.62% for the test data.

This represents a 14-percentage drop in accuracy and similar percentage drops in AUC, Recall, Precision and F1, and a circa 27-percentage drop in the Kappa and MCC scores signalling this level of increase in randomness in correctly predicting the target variables.

The classification performance results for SHAP's top 10 feature set were therefore not as strong.

4.6 Discussion

This study set out to explore the use of low code open-source ML tools to predict poverty, and to do so using a few features from datasets. The results obtained above address the questions this study sought to answer, which are: i) if poverty can be predicted using off-the-shelf ML tools employed on survey data, and ii) whether poverty prediction based on a smaller number of features can give acceptable results compared to those obtained using the full feature set.

The literature review unearthed three past studies that used ML methods on Malawi survey data to predict poverty. These are: Jean et al. (2016); Nichols and McBride (2018); and Fitzpatrick et al. (2018).

While the first one, Jean et al. (2016) used a combination of satellite and survey data, their prediction performance was relatively low, with an r-squared value of 55%. In addition, as noted in the literature review, these results were at cluster level rather than at individual household level.

On the other hand, like this study, Nichols and McBride (2018) sought to use ML rather than a combination of domain knowledge expertise and regression to identify best poverty predictors to use in the development of poverty PMTs. They too used survey data but employed ML ensemble techniques and achieved a predictive model performance of 80% for Malawi, though using IHS2 (2004 – 2005) data. However, there were no deliberate feature selection techniques to select top explanatory features. They left it to the ML algorithms to achieve the good predictive accuracy from the full feature sets.

The 2018 World Bank competition challenged experts to build best ML poverty predictors for three countries including Malawi, using survey data but without a restraint on methods. The top competitors succeeded to increase predictive accuracy performance to nearly 89% though complexity of their methods was high which is contrary to our research objective to use low-code ML tools and resources.

On the other hand, Fitzpatrick et al. (2018) set out to determine the best algorithm for predicting poverty. Consequently, he applied 10 pre-existing ML classification algorithms with no feature engineering. All ten classifiers performed consistently well with accuracy scores ranging from 86.4% – 87.4%. This represented only a marginal drop from the World Bank competition winning methods but using simpler methods.

This study closely followed FitzPatrick et al. (2018)'s approach but used only three classification algorithms and achieved an accuracy score of 87.07% for the full feature panel set. The main difference was that in this study, the Game Theory-based SHAP method was used in selecting the best explanatory features, rather than the filter methods employed by FitzPatrick et al. (2018). While filter methods require domain knowledge, the SHAP method was run automatically using python libraries. Yet, the classifiers' performance in this study was within the range of FitzPatrick et al. (2018)'s best predictive scores. This is a key achievement of this study in managing to achieve good results using simpler methods.

The predictive accuracy level was also maintained well at 86.94% when the subset of SHAP's top 100 features was used. Accuracy only dropped slightly by nearly a percentage point to 86.25% when the subset was reduced to 50 features, and by about 3 percentage points to 83.81% when the subset was reduced to 20 features. Accuracy however dropped substantially by nearly 14 percentage points to 73.34% when only 10 features were used. Even so, the ability of the ML methods to correctly predict poverty using a substantially reduced number of features at a similar or slightly lower number of features without any domain knowledge is quite remarkable.

For the smallest feature sets, FitzPatrick et al. (2018) achieved 76.7% as their highest predictive model accuracy on 10 features whereas this study achieved 73.34% on its top 10 features. While the 10-feature accuracy score of this study is lower, it still compares well with the 10-feature accuracy score that FitzPatrick et al. (2018) achieved, especially considering that this study's score was achieved without domain knowledge. FitzPatrick et al. (2018)'s set of top 10 features was derived using a manipulative stepwise approach analogous to the one used in regression analysis that McBride and Nichols (2018) sought to avoid. On the other hand, this study selected its set of top 10 features automatically using the Game Theory-based SHAP method.

According to McBride and Nichols (2018), PMTs use 20 – 50 questions. If the 10-feature score of 73.34% is considered too low, this study achieved 83.8% accuracy on 20 features and 86.25% accuracy on 50 features. The simpler approach in this study could therefore be used to develop PMT's with acceptable accuracy scores, not far from those acquired using the set of features, close to 500.

Model performance varied, with LGBM being the best classifier for the full panel dataset, while ExGBM performed better on Filter Method top 50 feature subsets, and LR performing better

on all smaller feature subsets selected using SHAP ranking and giving the overall best performance. However, the models performed close to one another in all cases, with a difference of within 1 percentage point across results reported at each stage. This suggests the use of several models in such exercises should be maintained as it is useful to verify the robustness of results reported by one model and considering the ease with which the methods can be applied, made possible by ML frameworks like PyCaret.

The literature review suggested that accuracy can be relied on as a model performance metric where a dataset is well balanced, which was the case in this study. The high accuracy scores of models in this study for both full and smaller feature sets were validated by equally high areas under the curve, precision, recall and F1 scores as well as MCC and Kappa coefficients. Where there was a drop in accuracy, there was a proportionate drop in the other metrics.

The findings meet the objectives of this study by demonstrating the potential to successfully predict poverty using low code, open source, ML algorithms and publicly available survey data at high accuracy levels, similar to the best performing ML approaches thus far. The use of SHAP method rather than the filter method also enabled an automated determination of a shortlist of best poverty predictors without requiring manipulation or domain knowledge, with accuracy only slightly lower than that achieved when a full feature panel set is used. These findings can assist in the development and deployment of PMTs in a much easier manner.

4.7 Summary and Analysis of the Results

In this chapter, the results obtained from the experiments have been presented and discussed. Based on the results, answers to the research questions have been posited. These are that poverty can be predicted from survey data using off-the-shelf ML tools. Also, poverty can be predicted at acceptable levels based on a much smaller number of features with the right feature selection methods, although with slight reductions in accuracy when the number of features is reduced to 50 or less.

The next chapter concludes the study. It considers the results obtained and offers suggestions on areas of further exploration building on the findings of this study.

CHAPTER 5
CONCLUSION

5.1 Introduction

This study explored the use of off-the-shelf and non-proprietary ML methods to predict poverty, using publicly available survey data. It went further to attempt to use fewer variables rather than the full range of features tracked in typical surveys. This was done with the aim to determine whether poverty can be predicted using off-the-shelf low code ML tools on survey data; and if poverty can be predicted by using a smaller number of features. The research therefore sought to answer two questions: (i) whether poverty be predicted using off-the-shelf ML tools employed on survey data; and (ii) whether poverty prediction based on a smaller number of features can give acceptable results when compared to results obtained using the full-panel feature set.

If successful in generating results with similar accuracy to those generated by traditional methods, such an approach would have succeeded in generating desired information by using freely available data and open-source ML methods with low computational power, hence easily applicable in a low-resource setting like Malawi. Moreover, if fewer data points can be used, the research will have demonstrated that smaller surveys that take less time and cost less could be administered with greater frequency, enabling closer monitoring of poverty trends.

In this chapter, the study is concluded by discussing the findings obtained in addressing the two goals of the study and reflecting on the implication of the answers to the research questions vis a vis the research objectives. Policy recommendations are made, based on the findings. Limitations of the study are noted, and areas for further exploration and research are also suggested.

5.2 Summary of Main Findings

This study aimed to explore how accurately off-the-shelf ML tools can classify household poverty. It also aimed to explore whether a smaller number of features could be used to successfully predict whether a household is poor or non-poor. It did so by training three off-the-shelf open-source predictive ML algorithms, namely Logistic Regression, ExGBM and LGBM, using 10-fold cross-validation, applying Raschka (2015)'s workflow for predictive modelling in ML and closely following the steps taken by Fitzpatrick et al. (2018) in apply the

ML pipeline for poverty classification. The full panel feature set of 486 was first applied to these classification models, followed by feature subsets of different sizes, and results compared. The features were ranked using firstly Filter (Zheng & Casari, 2018) and secondly SHAP (Lundberg & Lee, 2017) methods, and feature subsets extracted based on these rankings.

The study established that off-the-shelf ML tools can be used to predict which households are poor and which ones are not. It also established that poverty can be predicted using ML with a substantially reduced number of features. The top 100 features predicted poverty just as well as the full set of 486 features with the same accuracy rate of 87%. There was a slight drop in accuracy by close to 1 percentage point to 86.23% when the feature set size dropped to 50, and by about 3 percentage points to 83.81% when the top 20 features were used. Thus, the study confirmed that far fewer features could be used to predict poverty with the same or very similar success as with the full feature set. Therefore, this research could significantly aid in the design of surveys that can be administered more cheaply and more frequently.

The 87.07% accuracy result reached in this study for the full panel feature set also compares well with those achieved by the three winners of the worldwide competition on poverty prediction by DrivenData Incorporated, who achieved accuracy scores between 88.5 and 88.9% for their full panel sets (Fitzpatrick et al., 2018). Notably, the three winners used ensembles which are more complex and computationally expensive methods that combine several classification algorithms to create a blended classifier.

An important finding of this work is that feature selection is crucial when seeking to classify poverty correctly using surveys with fewer questions. The accuracy of the SHAP method using 50 features was 86.25% while the Filter method produced 63.67% accuracy, a difference of 22.58 percentage points. For the 10 feature sets, Fitzpatrick et. al (2018)'s highest score was 76.7% while for this study which used SHAP method it was 73.4%, so a difference of 3.3 percentage points. Therefore, the SHAP method (Lundberg & Lee, 2017) that is low code produced better or comparable results for smaller feature sets when compared with the Filter method used by Fitzpatrick (2018) that requires domain knowledge. Accuracy of 73.4% is still relatively high and if acceptable, then 10 features could quickly be deployed to predict poverty, still with a relatively high accuracy level. Otherwise, the number of features could be increased to 20 or to 50, to gain the increase accuracy of up to around 86% as desired.

5.3 Recommendations for Policy

It is clearly impractical and costly to carry out surveys of close to 500 questions frequently, but it would be easier to carry out surveys of 20 to 50 questions, or even 10 an accuracy level of circa 73% is acceptable, with greater frequency. A policy recommendation for Malawi is therefore to use shortlists of the top features to design shorter or proxy surveys or to fine-tune other surveys, to gather information in between the five-yearly integrated household surveys that would provide adequate data for poverty classification. Up-to-date information and closer monitoring of trends would also enhance timeliness in effecting interventions and better targeting of resources to those most in need.

5.4 Limitations of the Study

According to the ML pipeline, once satisfied with the results of a predictive model, it can be applied to predict "unseen" and "new future" data. This study was limited to IHS3 (2010-2011), and it is the data reserved for test that was used as the "unseen data" and used for prediction. IHS4 (2016 – 2017) and IHS5 (2019-2020) are now also available on the World Bank Portal. However, this study did not apply its predictive models on either IHS4 or IHS5 data as "new future" data.

Based on the work of Fitzpatrick (2018), this project focused on three ML models namely, LR, LGBM and ExGBM. It is possible that different models and techniques may perform better on smaller feature sets.

5.5 Areas for Further Research

This study focused on three classifiers among the top performers in the ML poverty classification study by Fitzpatrick et al. (2018). These three classifiers were also part of the base classifiers in the top-performing classification models in a worldwide competition on poverty prediction hosted by DrivenData Incorporated (Fitzpatrick et al., 2018).

The performance results from the models demonstrate that it is possible to classify poor households using fewer features and the re

Having demonstrated here that far smaller feature sets can give accuracy comparable to the full feature set, the next step would be to investigate whether even greater accuracy on the small feature sets can be achieved using different ML techniques. More complex approaches, for example, ensemble techniques such as bagging and stacking, should be investigated to check

if they can boost prediction levels on few features. Another area to explore is the use of recommendation algorithms that ask the next question based on the previous response. These can therefore quickly filter out irrelevant features and reach confirmation of whether a household is poor or non-poor. Notably, surveys based on recommendation algorithms would need to be administered through electronic devices such as tablets or mobile phones. Furthermore, future studies can focus on much more recent Malawi data survey datasets such as IHS4 and IHS5 and survey data from other developing countries to check if models trained on IHS3 can classify a poor and non-poor household.

Limiting this study to IHS3 data made possible the comparison of the performance of this study's models with those in Jean et al. (2016) and the World Bank top three competitors reported by FitzPatrick (2018), since they also used Malawi IHS3 data for their studies. This comparison was important given the second objective of this study that explored the use of fewer features to predict poverty at acceptable levels. However, the findings of this study can be applied to IHS4 and IHS5 survey data to see if model performance is maintained on both full and reduced feature sets.

Finally, this study focused on data for Malawi; its findings should be tested on data for other countries to verify if much smaller questionnaires can also be used successfully elsewhere.

REFERENCES

Abou Omar, K. B. (2018). *XGBoost and LGBM for Porto Seguro's Kaggle challenge: A comparison.* Zurich : Computer Engineering and Networks Laboratory ETH Zurich .

Analytics Vidhya. (2020, July 20). *Analytics Vidhya*. Retrieved from Analytics Vidhya: https://www.analyticsvidhya.com/

Baldi, P. (2000). Assessing the accuracy of prediction algorithms for classification: An overview. *PubMed*.

Bilton, P., Jones, G., Ganesh, S., & Haslett, S. (2017). Classification trees for poverty mapping. *Computational Statistics and Data Analysis*.

Blumenstock , J., Cadamuro, G., & On, R. (2015). Predicting poverty and wealth from mobile phone metadata. *Science*.

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. (pp. 785-794). *In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785-794). ACM.

Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlatio coefficient (MCC) over F1 score and accuracy in binary classification evaluation . *BMC Genomics* .

Chollet , F., & Others . (2015). *Keras*. Retrieved from GitHub: https://github.com/fchollet/keras

Cohen , J. (1968). Weighed kappa: Nominal scale agreement with provision for scaled disagreement or partial credit. *Psychological Bulletin*, 213-220.

Davis , J., & Goadrich, M. (2006). The Relationship Between Precision-Recall and ROC Curves. *Procedings of the 23rd International Conference on Machine Learning.* Pittsburgh.

Doshi-Velez, F., & Kim, B. (2017). Towards A Rigorous Science of Interpretable Machine Learning. *arXiv e-prints*, arXiv:1702.08608.

Dupriez, O. (2018). An empirical comparison of machine learning classification algorithms. *Development Data Group* (p. 4). Washington DC: Worldbank. Retrieved from http://pubdocs.worldbank.org/en/666731519844418182/PRT-OD-presentation-V2.pdf

Fadlullah, Z., Tang, F., Mao, B., Kato, N., Akashi, O., Takeru, I., & Mizutani, K. (2017). State-of-the-Art Deep Learning: Evolving Machine Intelligence Toward Tomorrow's Intelligent Network Traffic Control Systems. *IEEE Communications Surveys & Tutorials*, 2432-2455.

Ferlitsch. (2020). *Deep Learning Design Patterns.* New York : Manning Publications.

Fitzpatrick, C. A., Bull, P., & Dupriez, O. (2018). *Machine learning for poverty prediction:A comparative assessment of classification algorithms.* Washington DC: WorldBank. Retrieved from https://github.com/worldbank/ML-classificationalgorithms-poverty.

Friedman, J. H. (2002). Stochastic gradient boosting. *Computational Statistics & Data Analysis, 38*(4), 367 - 378.

Géron, A. (2019). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems.* O'Reilly Media.

Gravemeyer, S., Gries, T., & Xue, J. (2010). Poverty in Shenzhen. Rising China in the Changing World Economy. 10.4324/9780203144596. . *Center for international Economics* .

Guo, P. (2014). Python is now the most popular introductory teaching language at top us universities (2014). *Communications in ACM*.

Harris , C. R., & Others. (2020). Array programming NumPy. *Nature*, 357-362.

Head, A., Tran, N., Manguin, M., & Blumenstock, J. (2017). Can Human Development be Measured with Satellite Imagery? *ICTD '17, November 16–19, 2017, Lahore, Pakistan.* Lahore : Association for Computing Machinery.

Hunter , J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering, 9*(3), 90-95.

International Monetory Fund. (2017). *Malawi Economic Development Document.*

Jean, N., Burke, M., Xie, M., Davis, W., Lobell , D., & Ermon, S. (2016). Combining satellite imagery and machine learning to predict poverty. *Science*, 790-794.

Jerven, M. (2013). *Poor Numbers: How We Are Misled by African Development Statistics and What to Do about It.* London : Cornell University Press.

Knippenberg, E., Jensen, N., & Constas, M. (2019). Quantifying household resilience with high frequency data: Temporal dynamics and methodological options. *World Development 121*, 1-15.

Korivi, K. (2016). *Identifying Poverty-driven Need by Augmenting Census and Community Survey Data.* Kansas: Kansas State University.

Kshirsagar, V., Wieczorek, J., Ramanathan, S., & Wells, R. (2017). Household poverty classification in data-scarce environments: a machine learning approach. *arXiv*.

Lars, B., Gilles, L., Mathieu, B., Pedregosa, F., Mueller, A., Grisel, O., & Holt, B. (2013). *Scikit Learn.* Retrieved from Scikit Learn. Retrieved from : https://scikit-learn.org/stable/

Lundberg, S. M., & Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions. *Advances in Neural Information Processing Systems*.

Malawi Government. (2012). *Household Socio-Economic Characteristics Report.* Zomba: National Statistical Office.

Malawi Government. (2017). *Household Socio-Economic Characteristics 2016.* Lilongwe: National Statistical Office.

McBride , L., & Nichols, A. (2018). Retooling Poverty Targeting Using Out-of-Sample Validation and Machine Learning. *The World Bank Economic Review*, 531-550.

McKinney , W., & et. al. (2010). *Data structures for statistical computing in python* (Vol. 445). Austin, Texas.

Moez, A. (2020). *Pycaret: An open source, low-code machine learning library in Python.* Retrieved from https://www.pycaret.org

Mullainathan, S., & Spiess, J. (2017). Machine learning: an applied econometric approach. *Journal of Economic Perspectives*, 87-106.

Newhouse, D., Shivakumaran, S., Takamatsu, S., & Yoshida, N. (2014). *How Survey-to-Survey Imputation Can Fail.* Washington DC: World Bank .

Nguyen, T. T., & Armitage, G. (2008). A survey of techniques for internet traffic classification using machine learning. *IEEE communications surveys & tutorials*, 56-76.

Paper, D. (2018). *Exploring Data: Data Science Fundamentals for Python and MongoDB, 167-209.* Berkeley, CA: Apress.

Pedregosa, F., Varoquaux, G., Gramfort, A., & Michel, V. (2011). Scikit-learn: Machine learning in Python. *Journal of machine learning research, 12*(Oct), 2825-2830.

Pypi. (2020, 10 10). *https://pypi.org/*. Retrieved from pypi: https://pypi.org/

Rahman, M. A. (2013). Household characteristics and poverty: A logistic regression analysis. *The Journal of Developing Areas, 303-317.*

Raschka, S. (2015). *Python Machine Learning* . Birmingham : Packt Publishing.

Sani, N. S., Rahman, M. A., Bakar, A. A., Sahran, S., & Sarim, H. M. (2018). Machine Learning Approach for Bottom 40 Percent Households (B40) Poverty Classification . *International Journal on Advance Science Engineering Information Technology*.

SciKit Learn. (2021, March 6). *Cross-validation: evaluating estimator performance*. Retrieved from Scikit-Learn: Machine Learning in Python: https://scikit-learn.org/stable/modules/cross_validation.html

Seabold, S., & Perktold , J. (2010). statsmodels; Econometrics and statistical modeling with python. *9th Python in Science Conference* .

Shap. (2020, Oct 12). *https://shap.readthedocs.io/en/latest/.* Retrieved from https://shap.readthedocs.io/en/latest/: https://shap.readthedocs.io/en/latest/

Shapley, L. S. (1953). A Value for n-person games. *Contributions to the theory of Games, 2*(28), 3017-317.

Sohnesen, T. P., & Stender, N. (2017). Is Random Forest a Superior Methodology for PredictingPoverty? An Empirical Assessment. *Poverty and Public Policy* .

Steele, J., Sundsoy, P., Pezzulo, C., Alegana, V., Bird, T., Blumenstock , J., . . . Bengtsson, L. (2017). Mapping poverty using mobile phone and satellite data. *J.R. Soc. Interface 14*.

Thoplan, R. (2014). Random Forests for Poverty Classification. *International Journal of Sciences:Basic and Applied Research (IJSBAR).* , 252-259.

Tiziana Rancati, C. F. (2019). *Modelling Radiotherapy Side Effects: Practical Applications for Planning Optimisation.* CRC Press.

UNDP. (2018). *United Nations Development Programme Annual Report 2018.* New York, NY 10017: UNDP.

United Nations. (2019). *The Sustainable Development Goal Report 2019.* New York: United Nations.

Vatcheva, K. P., Lee, M., McCormick, J. B., & Rahbar, M. H. (2016). Multicollinearity in Regression Analyses Conducted in Epidemiologic Studies. *Epidemiology (Sunnyvale, Calif.),, 6*(2), 227.

Waskom, M. L. (2021). Seaborn: statistical data visualization. *Journal of Open Source Software*, 3021.

Williams , N., Zander , S., & Armitage , G. (2006). *Evaluating machine learning algorithms for automated network application identification.* Swinburne: CAIA Technical Report 060410B.

World Bank Group . (2020, July 12). *https://datacatalog.worldbank.org/dataset/*. Retrieved from https://datacatalog.worldbank.org/dataset/: https://datacatalog.worldbank.org/dataset/

World Bank Group. (2020, June 20). *https://microdata.worldbank.org/index.php/catalog/3016/data-dictionary/F3?file_name=MWI_2010_household*. Retrieved from https://microdata.worldbank.org: https://microdata.worldbank.org

Xie, M., Jean, N., Burke, M., Lobell, D., & Ermon, S. (2016). Transfer learning from deep features for remote sensing and poverty mapping. *Thirtieth AAAI conference on artificial intelligence.* AAAI.

Zheng, A. (2015). *Evaluating Machine Learning Models.* O'Reilly Media Inc.

Zheng, A., & Casari, A. (2018). *Feature Engineering for Machine Learning.* O'Reilly Media Inc.