Statistical learning for class imbalanced data: a case study of malaria indicator survey data

Maangi Daniel Ongera

Submitted in total fulfilment of the requirements for the degree of Master of Science in Statistical Sciences of Strathmore University

Institute of Mathematical Sciences

Strathmore University

Nairobi, Kenya

May 2022

This thesis is available for Library use through open access on the understanding that it is copyright material and that no quotation from the thesis may be published without proper acknowledgement.

Declaration

I declare that this work has not been previously submitted and approved for award of a degree by this or any other University. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made in the thesis itself.

© No part of this thesis may be reproduced without the permission of the author and Strathmore University.

Maangi Daniel Ongera

.....

August 22, 2022

Approval

The thesis of Maangi Daniel Ongera was reviewed and approved by the following:

Professor Bernard Omolo

Supervisor,

Institute of Mathematical Sciences, Strathmore University.

Dr. Godfrey Madigu

Dean,

Institute of Mathematical Sciences, Strathmore University.

Dr. Bernard Shibwabo

Director,

Office of Graduate Studies, Strathmore University.

Abstract

Background

Class imbalanced problems are predominant in real-life applications. In most cases, the minority class is the most important. Standard statistical learning algorithms tend to produce poor results for the minority class and very good results for the majority class. One of the widely used mechanism to address this problem is by re-sampling the training data. The objective of this study is to examine the performance of statistical learning algorithms by using different re-sampling approaches for handling class imbalance.

Methods

Two classical and ensemble statistical learning techniques were trained on an imbalanced Malaria Indicator Survey data set while handling the majority-minority problem through re-sampling. These included: Logistic regression, support vector machines, random forest, and extreme gradient boosting. The algorithms were trained without handling class imbalance first. Secondly, the algorithms were trained using six re-sampling procedures to handle class imbalance: random under-sampling, random over-sampling, Synthetic Minority Over-sampling technique (SMOTE), Random Over Sampling Examples (ROSE) techniques and Adaptive Synthetic Sampling Approach (ADASYN). We further investigated whether combining randomly under-sampled and over-sampled data can result in improved performance. Eighty percent of the data was used for model training using 5 fold cross validation.

Results

All methods that were considered for handling class imbalance had strengths and weaknesses depending on the performance metric. For instance, random under-sampling (RU) resulted in models with higher sensitivity than random over-sampling (RO). To get a trade-off between sensitivity and specificity, these two methods can be combined (RURO). This approach resulted in 99.5% sensitivity, 88.1 % specificity, 89.6 % precision, 94.3 % F1 score and a 93.9 % accuracy on the test set using the Extreme Gradient Boosting machine.

Table of contents

List of figures

List of tables

List of abbreviations

WHO	World Health Organisation
KMS	Kenya Malaria Strategy
DHS	Demographic Health Surveys
MIS	Malaria Indicator Survey
ML	Machine Learning
LR	Logistic Regression
SVM	Support Vector Machines
RF	Random Forest
XGBoost	Extreme Gradient Boosting
SMOTE	Synthetic minority over-sampling technique
BSMOTE	Borderline synthetic minority over-sampling technique
ROSE	Random oversampling Examples
ADASYN	Adaptive snythetic sampling approach
RURO	Random Undersampling and Random Oversampling

Acknowledgement

First and foremost, I am grateful to the Almighty God for good health, wellbeing and the resources that were necessary in completing this research study.

My very special gratitude to my supervisor, Prof. Bernard Omolo, for his responsiveness, suggestions, feedback and motivation that was a key driver in this research. Without his assistance, this research work would not have been successful. Writing this paper was difficult than I would ever imagined. Therefore, I am equally grateful to the defense committee for their generous feedback and motivation.

A very special thanks to my in loving memory parents, Evans Maangi and Hellen Maangi. I also express my profound gratitude to my siblings, for always believing in me, and their unwavering support and encouragement throughout my studies.

I would also like to record my sincere gratitude to one and all, who in one way or another, lent their hand in this work.

Chapter 1

Introduction

1.1 Background

Statistical learning is a vast set of tools for understanding data James et al. (2013). It is a new field that has emerged in the field of statistics. Broadly, statistical learning can be classified into supervised or unsupervised learning. Supervised learning refers to the development of statistical models to predict or estimate an output using one or more inputs James et al. (2013). Supervised learning is based on ground truth. That is, we already have prior information on what the output from the analysis should be. Unsupervised learning on the other hand does not have labeled or supervising outputs.

In supervised learning, there's an associated response for each observation of the predictor measurements. The goal is to fit a model that can be used for prediction or inference by relating the predictor measurements to the response. For prediction, the aim is to develop a model that accurately predicts the response on unseen data. For inference, the aim is to understand the relationship between the predictors and the response. Response variables can either be quantitative or qualitative. Quantitative variables take on numerical values, for example age, salary, height, weight et cetera. Qualitative variables are also known as categorical variables. They take on one class out of the different k classes. For instance, whether an individual is negative or positive for a particular disease.

Based on the type or response variables, supervised learning problems are broadly categorized into regression and classification. Regression problems involve a quantitative response whereas classification problems involve a qualitative response. In general, supervised learning methods are selected on the basis of the response variable. For example, linear regression when the response is quantitative and logistic regression when the response is qualitative.

The type of the predictor variables is not important in supervised learning James et al. (2013). Classification methods can further be classified into linear and nonlinear. In a linear problem, a simple linear classifier can be used to approximate the class boundaries. In a nonlinear problem, the class boundaries cannot be be approximated well with linear hyperplanes.

Classification is a key task of pattern recognition. One problem hindering performance of classification methods is class imbalance. This is because of their accuracy tailored design Fernández et al. (2018). A class imbalanced problem is one in which the distribution across classes is skewed or biased. That is to say, the class imbalance problem arises when the samples representing one class are much lower than the rest of the classes Fernández et al. (2018). In a binary classification setting, a class imbalanced problem arises when one of the two classes has more samples (the majority class) than the other class (the minority class) Longadge and Dongre (2013). This can vary from a slight difference between the classes to a severe imbalance in which there's only one instance of the minority class in hundreds, thousands or millions of observations. Usually, the most important class is the one with fewer samples Guo et al. (2008). This is a common problem in most real world datasets. For instance, in fraud detection, majority of the transactions are authentic. In spam detection, majority of emails or text messages are not spam. In disease screening, majority of people may not have the disease. One of the common approaches to mitigate the issues due to class imbalances is by re-sampling the training data. This project is an empirical evaluation of the predictive performance of various statistical learning algorithms using re-sampling approaches as a way of mitigating the class imbalance problem. Kenya's 2020 Malaria Indicator Survey data set is used as a case study Division of National Malaria Programme, Ministry of Health, Kenya et al. (2021).

1.2 Problem statement

Malaria remains to be one of the major public health threats of our time. It is among the leading causes of mortality in low-income countries alongside Tuberculosis and HIV/AIDS WHO (2020). In 2020, there were 241 million cases of malaria and approximately 627,000

people died from Malaria - an increase of about 69,000 from 2019. African countries continue to carry a disproportionately high share of the global Malaria burden. In 2020, 95% of these world Malaria cases were from Africa WHO (2021). In Kenya, Malaria is still a major public health and socioeconomic problem, with the highest prevalence being among children less than 14 years.

According to the Division of National Malaria Programme, Ministry of Health, Kenya et al. (2021), 6% of children aged between 6 months and 14 years were positive for Malaria by microscopy. In this data set, the majority class (94 %) are negative whereas the minority class (6 %) are positive for malaria infection. In this majority-minority setting where the positive examples are rare, it can be very difficult to estimate the probability of a child having malaria infection given a set of malaria risk factors (prediction). Furthermore, this problem can pose a challenge in trying to understand the relationship between the predictors and the response (inference). Majority of statistical learning algorithms applied on this unbalanced data will tend to result in biased predictions against the positive class. Many methods have been identified to handle class imbalance during modelling. One of the ways is through re-sampling the training data Fernández et al. (2018). Broadly, this method entails removing samples from the majority class (negative for malaria) or increasing the samples in the minority class (positive for malaria). This study applies no sampling, under-sampling, over-sampling, and hybrid methods when training the statistical learning algorithms. In addition, we investigate whether combining both randomly under-sampled and randomly over-sampled data can result in improved performance. We consider two classical statistical learning techniques and two algorithms from the ensemble framework.

1.3 Objectives

1.3.1 Main objective

The objective of this study is to examine the predictive performance of statistical learning algorithms by using different data level approaches to handle class imbalance.

1.3.2 Specific objectives

- To compare the performance of data level approaches in handling class imbalance.
- To investigate whether combining randomly undersampled and randomly oversampled data can result in improved perfomance.

Chapter 2

Literature review

2.1 Introduction

Technically, a data set can be considered to be imbalanced if it has unequal distribution between the majority and minority classes Leevy et al. (2018). This is common in real-world data sets, and it can vary from slight to severe imbalance. The bulk of the data set is composed of the majority class, whereas the minority class, which is often the interest, has limited representation. This section provides a review of the approaches in learning from class imbalanced data.

2.2 Handling class imbalance

Class imbalance in a data set can skew the predictive performance of classifiers in a dramatic way Leevy et al. (2018). There are many approaches that have have been developed in order to correctly distinguish the minority class from the majority class. According to Fernández et al. (2018), these approaches can be categorized into four groups. The grouping is based on how they handle the class imbalance problem. That is, data level, algorithm level, cost-sensitive, and ensemble based approaches.

2.2.1 Algorithm level approaches

Algorithm level approaches are those that bias learning towards the minority class by adapting existing classification algorithms. The aim of the approach is to modify the actual classifier learning procedure to handle class imbalance issues. However, a deep understanding of the

learning algorithm is needed to identify the mechanisms driving the bias to the majority class. These algorithms are however difficult to design and implement than other methods. Despite some outstanding performers, the results in literature are conflicting and inconsistent Leevy et al. (2018).

2.2.2 Cost-sensitive learning approaches

Majority of statistical learning algorithms assume equality of misclassification errors. However, this is not usually the case. A false negative can be considered more costly or worse than a false positive. For instance, in medical cancer diagnosis, falsely classifying a patient as negative, when the patient is actually is positive, is considered much more costly Ling and Sheng (2008). Cost-sensitive learning takes into account the costs of prediction errors during model training. Cost-sensitive classifiers can be categorized into two groups: direct approaches that involve direct introduction of the misclassification cost into the training procedure, and meta-learning approaches that either modify the training data (pre-processing) or the outputs (post-processing) Fernández et al. (2018). The performance of cost-sensitive classifiers are heavily dependent on the provided cost matrix. Incorrect initial costs can affect the learning process. These costs can originate from a domain expert or during model training Fernández et al. (2018).

2.2.3 Ensemble-based approaches

Most of the prediction errors of standard statistical learning algorithms are as a result of bias or variance. The idea behind ensemble learning is to minimize these errors by combining several base learners (often weak learners) Hastie et al. (2009). The final result is usually a more accurate meta model. Ensemble learning methods include: bagging, boosting and stacking. However, these techniques cannot solve the majority-minority problem by themselves Fernández et al. (2018). To handle class imbalance, ensemble learning algorithms are adapted by combining them with either data level, algorithm-level or cost-sensitive approaches. For instance, Huda et al. (2018) developed three base learners using different

oversampling techniques. A random forest based ensemble of these learners resulted in improved predictive performance.

2.2.4 Data level approaches

These are also known as sub-sampling or re-sampling approaches. Re-sampling training data was the first method for handling class imbalance Fernández et al. (2018). It is still one of the commonly used mechanism. The main advantage of sampling methods is that they are simple to implement and can be used with any classifier. They involve the use of different sampling procedures to provide an adequately balanced data set to the statistical learning algorithms. The goal of data level approaches is to obtain a balanced class distribution by resampling the training data. These methods are independent of the learning algorithm and can be easily implemented. According to Fernández et al. (2018), there are three different ways in which resampling techniques can be used to produce a balanced class distribution: undersampling, oversampling and hybrid methods.

Undersampling methods

These approaches involve randomly subsetting all the classes in the training data set such that the frequency of each class matches the least common class. A major drawback in this approach is the potential loss of useful information Leevy et al. (2018). This is because most of the examples from the majority class are ignored. There are several strategies proposed for undersampling, the simplest being random downsampling. This method randomly eliminates instances of the majority class to obtain a balanced distribution. Although this method is simple and effective, majority class examples are deleted without knowing how important they might be in estimating the class boundary. Several heuristic solutions that try to identify redundant majority class instances that can be removed have been suggested. These solutions informatively undersample the prevalent class, i.e. the choice of examples to be removed is targetted Sun et al. (2009). Classical undersampling methods include: Tomek Links (TL), Condensed Nearest Neighbour Rule (US-CNN), One-Sided Selection (OSS), US-CNN + TL,

Neighbourhoos Cleaning Rule (NCL), Class Purity Maximization (CPM), Undersampling Based on Clustering (SBC) and Near Miss approaches. Advanced undersampling techniques include: evolutionary undersampling, undersampling by cleaning data, ensemble based undersampling, and clustering based undersampling Fernández et al. (2018).

Burez and Van den Poel (2009) investigated the impact of random and advanced undersampling in the prediction of customer churn through gradient boosting and weighted random forests. AUC and lift were used as the model evaluation metrics. Overall, the results showed that under-sampling resulted in improved prediction accuracy. They further confirmed the finding by Weiss and Provost (2003) that there is no specific answer on which class distribution will result in better predictive performance. This entirely depends on the method and area of application. The researchers noted that it would be interesting to apply the resampling approaches they adopted to other learning novel algorithms such support vector machines, neural networks et cetera.

Oversampling methods

These approaches involve randomly sampling with replacement the minority class to obtain the same size as the majority class. The approach however may cause over-fitting because it produces exact copies of instances of the minority class Leevy et al. (2018). In addition, it may require extra computational burden. There are several strategies proposed for oversampling, the simplest being random over-sampling. This method randomly replicates instances of the majority class to obtain a balanced distribution. There are also solutions that informatively over-sample the minority class, i.e. the choice of examples to replicate is targetted Sun et al. (2009). According to Fernández et al. (2018), advanced oversampling methods include Synthetic Minority Oversampling Technique (SMOTE), and SMOTE extensions i.e. Borderline SMOTE, Adaptive Synthetic Sampling approach (ADASYN), Random Oversampling Examples (ROSE), Safe-level SMOTE, Majority Weighted Minority Oversampling Technique (MWMOTE), and Mahalanobis Distance Based Oversampling Technique (MDO). Batista et al. (2004), noted that over-sampling methods generally outperformed under-sampling methods using the area under the ROC curve metric. Besides, the authors

demonstrated that random over-sampling, the simplest approach, was very competitive to advanced over-sampling approaches.

Hybrid methods

These methods combine both under-sampling and over-sampling strategies. They arise due to the drawbacks for each of the resampling approaches. They aim to obtain an optimal balance of eliminating majority class instances and generating new instances of the minority class to obtain the best possible performance Fernández et al. (2018). Most of these approaches have involved enhancing the performance of classical SMOTE. For example, SMOTE + Tomek Link and SMOTE + ENN. More complex mechanisms include: Agglomerative Hierarchical Clustering (AHC), SPIDER, SMOTE-RSB, and SMOTE-IPF. Despite all these methods, there's no clarity on which of the strategies is the best.

2.3 Evaluation metrics

Standard statistical learning algorithms tend to have high accuracy for the majority class but poor results for the minority class. In other words, the samples from the minority class are misclassified more often than those from the majority class Fernández et al. (2018). This is because these methods tend to assume that the training data set is balanced. Accuracy is therefore less robust and unable to catch the nuances as a result of the different types of errors. For this reason, more informative metrics obtained from a confusion matrix are commonly used.

2.3.1 Confusion matrix

In two class problems, a confusion matrix is a summary of all possible outcomes in four categories: true positives, true negatives, false postives and false negatives.

Table 2.1: Two class confusion matrix

		Predicted: Positive	Predicted: Negative	Total
Actual	Positive	TP	FN	$(TP+FN)$
	Negative	FP	TN	$(FP+TN)$
	Total	$(TP+FP)$	$(FN+TN)$	N

These categories are defined as:

- True Positives (TP) : the number of positive examples correctly classified.
- True Negatives (TN) : the number of negative examples correctly classified.
- False Positives (FP) : the number of positive examples incorrectly classified.
- False Negatives (FN) : the number of negative examples incorrectly classified.

Many standard performance evaluation metrics to measure the quality of modelling can be derived using the the above $2x2$ confusion matrix.

2.3.2 Accuracy

Accuracy is one of the common methods for evaluating the performance of a classifier. It refers to the proportion of predictions that a classifier got right. That is, how well the predictions align with reality. Accuracy is formally defined as:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} = \frac{\text{TP + TN}}{\text{TP + TN + FP + FN}}$$

An accuracy of 99 % doesn't necessarily mean that the model is good if it does not illustrate the true positive and true negative attributes. Furthermore, in imbalanced data sets, accuracy is biased towards the majority class.

2.3.3 Other metrics

True Negative Rate

It refers to the proportion of negatives that were correctly classified as negative. The True Negative Rate (TNR) is formally defined as:

$$\text{TNR} = \frac{\text{True Negatives}}{\text{Total actual negatives}} = \frac{\text{True Negatives}}{\text{True Negatives} + \text{False Positives}} = \frac{\text{TN}}{\text{TN + FP}}.$$

True Positive Rate/ Recall/ Sensitivity

It refers to the proportion of positives that were correctly classified as positive. The True Positive Rate (TPR) is formally defined as:

$$\text{TPR} = \frac{\text{True positives}}{\text{Total actual positives}} = \frac{\text{True positives}}{\text{True Negatives} + \text{False Negatives}} = \frac{\text{TP}}{\text{TP + FN}}.$$

True positive rate is also referred to as recall or sensitivity. In disease diagnosis, it refers to the people who tested positive and are actually positive. It can be viewed as the probability that the classifier shows a positive result given that the individual is indeed positive for the disease. It is a good measure, when the cost of a false negative is high.

False Positive Rate

The False Positive Rate (FPR) is also referred to as the false alarm rate. It refers to the proportion of negatives that are incorrectly classified out of the total negatives Fawcett (2005).

$$\text{FPR} \approx \frac{\text{Negatives incorrectly classified}}{\text{Total negatives}}.$$

Specificity

Also known as the true negative rate. It refers to the proportion of negatives that were correctly classified. In disease diagnosis, it refers to the people who tested negative and are actually negative. It can be viewed as the probability that the classifier shows a negative result given that the individual is indeed negative for the disease.

$$\text{Specificity} = \text{TNR} = \frac{\text{True Negatives}}{\text{True Negatives} + \text{False Positives}} = \frac{\text{TN}}{\text{TN + FP}} = 1 - \text{FPR}.$$

Positive Predicted Value

It refers to the proportion of predicted positives that is truly positive. Positive Predicted Value (PPV) is formally defined as:

$$\text{PPV} = \frac{\text{True positive}}{\text{Total predicted positive}} = \frac{\text{True Positive}}{\text{True Positive + False Positive}} = \frac{\text{TP}}{\text{TP + FP}}.$$

Precision

Precision is the same as the Positive Predicted Value. It is a good measure, when the cost of a false positive is high.

$$\text{Precision} = \text{PPV} = \frac{\text{True Positive}}{\text{True Positive + False Positive}} = \frac{\text{TP}}{\text{TP + FP}}.$$

Negative predicted value

It refers to the proportion of predicted negatives that is truly negative. It is a good measure, when the cost of a false negative is high. Negative Predicted Value (NPV) is formally defined as:

$$\text{NPV} = \frac{\text{True negative}}{\text{Total predicted negative}} = \frac{\text{True Negatives}}{\text{True Negatives + False Negatives}} = \frac{\text{TN}}{\text{TN + FN}}.$$

F1 Score

The F1 score (F measure) has been reported as a good evaluation metric than accuracy in binary classification problems Sun et al. (2009). This is the harmonic mean of precision and recall. That is:

$$\text{F1 Score} = (1 + \beta^2) * \frac{\text{precision + recall}}{(\beta^2 * \text{precision}) + \text{recall}}.$$

Area under an ROC curve (AUC)

The Receiver Operating Characteristic (ROC) curve is a two-dimensional representation of the performance of a classification method. The y-axis represents the true postive rate (Sensitivity) while the x-axis has the false positive rate (1 - specificity). In order to compare different classifiers, a single scaler value is needed Sun et al. (2009). The area under the ROC curve (AUC) is the common metric used.

Youden's J index

Youden's J index is defined as:

$$\mathrm{J} = \mathrm{Sensitivity} + \mathrm{Specificity} - 1.$$

2.4 Malaria risk factors

Malaria is a disease caused by a parasite transmitted by an infected female anopheles mosquito. The risk of malaria infection, especially among children, can be increased by certain demographic, socioeconomic and environmental factors. Understanding these factors is important for targetted control interventions.

Winskill et al. (2011) noted that there was a significant association between both age and gender and malaria risk. Older male children had increased likelihood for malaria infection. This possibly indicated that older male children were associated with a higher exposure behaviour. Further, the authors demonstrated that use of insecticide-treated mosquito nets was associated with decreased odds for malaria infection among children. This adds to existing extensive literature on the efficacy of insecticide treated nets in malaria prevention. Habyarimana and Ramroop (2020), identified age of the child, location, poverty, number of household members, availability of household sleeping mosquito, residual spraying in the past twelve months, location of the household's source of drinking water, main wall materials of the dwelling, and the age of the head of the household to be significant risk factors for malaria infection among children. AIC, SC and $-2\mathrm{LogL}$ were used to compare the reduced

logistic regression model (model of intercept only) and the full logistic regression model (model of intercept and covariates). These metrics were smaller for the full model (model with intercept and covariates) compared to the reduced model (model with intercept only), suggesting that the full model was a better fit.

These models can be classified as inferential. In similar papers, authors typically focus on drawing qualitative statements on the relative influence the risk factors have on the response. For example, based on the p-value alone, one can state that there is a statistically significant relationship between malaria infection and the risk factors. However, there's is a connection between inferential and predictive models, which if ignored, can be dangerous. This is because such a model based only on statistical significance can perform poorly on other predictive capacity metrics such as sensitivity. Even when the model is not used for prediction, it can be difficult to trust inferences from a model with significant p-values but dismal sensitivity. In general, the predictive performance of a statistical learning algorithm involves determining how close fitted values are to the sample data.

Chapter 3

Methodology

3.1 Study data

Kenya's Malaria Indicator Survey data was used as the study data Division of National Malaria Programme, Ministry of Health, Kenya et al. (2021). The data was acquired in StataCorp (2019) format from the Demographic Health Surveys upon approval of the study. The outcome of interest was Malaria infection, whether negative or positive from a blood smear test. Based on literature, 9 variables were selected as malaria predictors among children aged 6 moths to 14 years. The table below provides a description of the study variables.

Table 3.1: Description of variables used in the study

Code	Description
malaria	Final result of malaria from blood smear test. The outcome variable.
child_age	Child's age in months.
hh_head_age	Age of head of household in years.
hh_members	Number of household members.
endemicity	Malaria endemicity zone.
has_net	Has mosquito bed net for sleeping.
residence	Residence (urban or rural).
wealth_index	Wealth index combined.
television	Whether household has television.
water_source	Source of drinking water.

We summarized the data using the mean for continuous variables and frequency for categorical variables. Statistical tests of independence were done at at p$-$value < 0.05.

3.2 Statistical learning methods

In this section, a description of each of the statistical learning methods adopted for this study is provided.

3.2.1 Logistic regression

Logistic regression is one of the most widely used statistical learning algorithms in classification. The model is used in prediction when the response variable is categorical (2 classes in this case). In this classification problem, there are two classes (C) : 1 - denoting malaria positive and 0 - malaria negative . Mathematically, the probability of a child having malaria given the predictors or risk factors (X) can be expressed as:

$$p(X) = P(C = \text{ malaria positive } | \text{ risk factors }) = P(C = 1|X). \tag{3.1}$$

Suppose that we use multiple linear regression to model the probability of a child having malaria as :

$$p(X) = \boldsymbol{\beta X} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_p. \tag{3.2}$$

We notice that we will will potentially have probabilities below zero (negative) and above one. This does not make sense. Logistic regression model arises from the need to ensure that the predicted probabilities of the two classes fall and remain in [0,1] Hastie et al. (2009). There are many functions that can be used. In logistic regression, the sigmoid (logistic) function is used. Hence, equation (3.2) can be written in terms of the logit (log odds) as:

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \boldsymbol{\beta X} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_p. \tag{3.3}$$

Equation (3.3) can be expressed as follows:

$$p(X) = \frac{e^{\beta X}}{1+e^{\beta X}} = \frac{e^{\beta_0+\beta_1 X_1+\beta_2 X_2+\cdots+\beta_n X_p}}{1+e^{\beta_0+\beta_1 X_1+\beta_2 X_2+\cdots+\beta_n X_p}}. \tag{3.4}$$

Where $X \in \mathbb{R}^p$ is vector of explanatory variables (malaria risk factors) and $\beta \in \mathbb{R}^p$ are unknown constants estimated by the method of maximum likelihood. This function takes an S-Shaped curve, and thus the predictions can never be below 0 or above 1 James et al. (2013).

3.2.2 Support Vector Machine

Since development in the 1990s, support vector machines (SVMs) have demonstrated good performance in different domains, and it is one of the most popular algorithm James et al. (2013). It can solve both classification and regression problems. If the positive and the negative classes can be separated perfectly using a hyperplane, then there exist an infinite number of these hyperplanes. The maximal margin hyperplane, or optimal separating hyperplane is the natural choice among these. It is the the hyperplane with the largest margin between the classes. This algorithm of classifying observations based on a maximal margin hyperplane is called the maximal margin classifier. However, if there exists no separating hyperplane, then we cannot obtain a maximal margin classifier that can separate the positive class from the negative class. Furthermore, even if a separating hyperplane exists, it might be less desirable to use it in some cases. For instance, when p is large, it can result in overfitting. Suppose that a separating hyperplane exists, and that we can obtain the maximal margin hyperplane. If $\beta_0, \beta_1, \ldots, \beta_p$ are the coefficients of this hyperplane, the maximal margin classifier will classify an observation, say x^* based on $f(x^*) = \beta_0 + \beta_1 x_1^* + \beta_1 x_2^* + \cdots + \beta_1 x_p^*$. That is, class 1 if $f(x^*)$ is positive and class -1 if $f(x^*)$ is negative.

However, in majority of the cases, a separating hyperplane does not exist. The support vector classifier is a generalization of the maximal margin classifier using a soft margin. It performs well in classifying most of the observations at the expense of misclassifying a few observations. The margin is called soft because of this. When the boundary between the positive and negative class is linear, the support vector classifier is the natural choice. In practice however, non-linear class boundaries are common. The support vector machine is an extension of the support vector classifier to solve the non-linear classification problems through a kernel trick which involves enlarging the feature space James et al. (2013). The

purpose of enlarging the feature space is to facilitate separating the classes with a non-linear boundary.

The linear support vector classifier is given by:

$$f(x) = \beta_0 + \sum_{i \in S}^{n} \alpha_i \langle x, x_i \rangle. \tag{3.5}$$

Whereby, $\langle x, x_i \rangle$ are inner products and S is a collection of the support points' indices. Support vectors define the position and margin of the separating hyperplane. They are called support vectors because if even one of them is moved, the position, margin or both will change. There will be no effect on the hyperplane if other points are moved instead.

In Support Vector Machines, the inner product is replaced with a generalized form known as the the kernel, $K(x, x_i)$. A kernel is function that estimates the similarity between any two observations. If a linear kernel is used, then the support vector machine will be simply the support vector classifier. There are three popular forms of SVM kernels which include:

- Polynomial kernel: $K(x_i, x_j) = (1 + \alpha_i \langle x_i, x_j \rangle)^p$
- Radial basis kernel: $K(x_i, x_j) = \exp(-\gamma \parallel x_i - x_j \parallel^2)$
- Neural network: $K(x_i, x_j) = \tanh(k_1 \langle x_i, x_j \rangle + k_2)$

In this paper, we have used radial basis kernel which is the most widely used. Suppose that $x^* = (x_1^*, \ldots, x_p^*)$ is a test observation, which has a large Euclidean distance ($\sum_{j=1}^{p} (x_j^* - x_{ij})^2$) from a training sample x_i. Then, the radial kernel, $K(x^*, x_i) = \exp(-\gamma \sum_{j=1}^{p} (x_j^* - x_{ij})^2)$ will very small. This implies that the training sample x_i plays virtually no role in $f(x^*)$ James et al. (2013). That is to say, training samples that are further away from the unseen test samples, will have no role in predicting the class of x^*. For this reason, the radial basis kernel a very localised behaviour. That is, only nearby training samples have an effect in predicting the class of x^*.

3.2.3 Bagging - Random forest

The random forest is one of the most popular algorithm from the ensemble learning framework. It is an improvement over bagging James et al. (2013). The idea behind bagging or

bootstrap aggregating is to reduce the variance of a statistical learning algorithm through a combination of many noisy homogeneous base learners.The expected value of B trees is the same as that of an individual tree. This is because, the trees are identically distributed. This implies that the bias of a single tree is the same as the bias of the average of any B such trees. In contrast to boosting which reduces both bias and variance, random forests improve performance by reducing variance. Suppose that the variance of each individual tree is σ^2. Averaging B of these trees (identically distributed but not independent with a positive pairwise correlation p), then the variance is:

$$p\sigma^2 + \frac{1-p}{B}\sigma^2$$

We notice that, as B increases, the second term will disappear. This suggests that the pairwise correlations limit the benefits resulting from averaging. This is the reason behind the random forests. As opposed to bagging, the random forest involves averaging a large collection of de-correlated trees Hastie et al. (2009). Instead of using the entire feature space to grow a tree, the random forest creates a collection of uncorrelated trees (homogeneous base learners) by using a random subset of predictors whenever a split is considered. Approximately, the number of the new set of predictors (m) to be considered at each split is equal to the square root of the total number of the entire feature space (p) James et al. (2013). That is:

$$m \approx \sqrt{p}$$

Trees are however notoriously known to be noisy Hastie et al. (2009). The out-of-bag (OOB) samples are often used to estimate performance of a random forest. These are samples that have left out in each bootstrap sample. That is, we predict the response for each observation in the out of bag sample using trees where that observation was an OOB James et al. (2013). To obtain a single prediction, the predicted responses are averaged (regression) or through majority voting (classification).

However, bagging or random forest leads to increased complexity in model interpretation as compared to a single tree. That is, it is not clear which variables are the most important in the bagged or random forest model James et al. (2013). We can however estimate the

importance of each predictor in the final model by using RSS (bagged regression) or the Gini index (bagged classification). The predictor with the largest decrease is the most important predictor. The algorithm below provides a step by step implementation of the random forest Hastie et al. (2009).

Algorithm 1 Random forest for Classification

- For $b = 1$ to B:
  - Draw a bootstrap sample Z^* with size n from the training set D_{train}.
  - Recursively repeat the steps below to grow a random forest tree T_b using the bootstrap sample Z^* until you reach you reach the minimum node size n_{min}.
    - Randomly select m features from the entire feature space p.
    - Among the m variables, pick the best split-point/ variable.
    - Split the node into two daughter nodes.
- Output the ensemble of trees T_{b1}^{B}.
- Let $\hat{C}_b(X)$ be the predicted class by the bth random forest tree.
- Classification: $\hat{C}_{rf}^{B}(x) =$ Majority Vote$\{\hat{C}_b(x)\}_1^B$

3.2.4 Boosting - XGBoost

Boosting is another ensemble technique for improving the performance of a statistical learning algorithm. In contrast to bagging, boosting does not involve drawing random samples. It involves growing a tree sequentially using the original data by fitting each new tree to the residuals from the previous trees and updating the residuals James et al. (2013). Boosting has shown to provide better predictive performance than bagging. The Gradient Boosting Machine (GBM), originally coined by Friedman (2001), is based on the boosting concept and can handle both regression and classification problems. Whereas the random forests involves averaging deep individual trees, the GBM sequentially builds a committee of weak and shallow trees which are an improvement from the previous. It is a redefinition of boosting whereby weak learners are added using gradient descent. Gradient descent algorithm is an iterative first-order optimization for finding the local minimum of a function that is differentiable. Usually, to reach this local minimum involves many steps James et al. (2013). The number of the required steps is determined by the learning rate, γ.

Algorithm 2 Gradient Boosting algorithm

- Initialize $f_0(x) = \arg\min_\gamma \sum_{i=1}^{N} L(y_i, \gamma)$
- For $m = 1$ to M:
  - For $i = 1, 2, \ldots, M$ calculate

$$r_{im} = -\left[\frac{\partial L(y_i, f_1(x_i), \ldots, f_K(x_i))}{\partial f_k(x_i)}\right]_{f(x_i)=f_{m-1}(x_i)}$$

.

$$= I(y_i = G_k) - p_k(x_i)$$

  - Fit a classification tree to the targets r_{im} giving terminal regions $R_{jm}, j = 1, 2, \ldots, J_m$.
  - For $j = 1, 2, \ldots J_m$ calculate

$$\gamma_{jm} = \arg\min_\gamma \sum_{x \in R_{jm}} L(y_i, f_{m-1}(x_i) + \gamma)$$

.

  - Update $f_m(x) = f_{m-1}(x) + \sum_{j=1}^{j_m} \gamma_{jm} I(x \in R_{jm})$.
- Output $\hat{f}(x) = f_M(x)$.

However, the gradient boosting machine can be prone to over-fitting. Extreme Gradient Boosting (XGBoost) Chen et al. (2015) is a scalable and efficient implementation of the gradient boosting machine. To control for over-fitting and achieve attain better predictive performance, XGBoost uses a regularized model formulation:

$$\text{Obj}^{(t)} = \sum_{i=1}^{n} L(y_i, \hat{y}_i^{(t)}) + \sum_{t=1}^{t} \Omega(f_i). \tag{3.6}$$

Where, the first term represents loss function that is used to measure the difference between the predictions $\hat{y}_i$ and the targets y_i. The second term is used to penalize the complexity of the model i.e, the regression tree functions. It is given by,

$$\sum_{t=1}^{t} \Omega(f_i) = \gamma_T + \frac{1}{2}\lambda \parallel \omega \parallel^2$$

XGBoost is a widely used algorithm that has shown to achieve high predictive performance on many supervised learning challenges. For instance, more than half of the winning solutions in 2015 Kaggle machine learning challenges used XGBoost Chen and Guestrin (2016).

3.3 Class imbalance handling

In this section, an overview of the methods for handling class imbalance is provided. There are many methods for handling class imbalance. However, in this paper we have used common data-level approaches.

3.3.1 Random undersampling

Random undersampling (Figure 3.1) is one of the common and simplest approaches for handling class imbalance. It is a non-heuristic approach where examples of the majority class are randomly eliminated from the data set. In this study, the majority class is the number of children who have negatively tested for malaria (94%). Therefore in this method, we randomly deleted malaria negative cases to obtain a balanced data set. The main advantage of this approach is that; deletion of some instances of the majority class leads to a significant reduction in the size of the training data and therefore reducing the time required in fitting a statistical learning model. A major problem associated with this technique is the potential loss of useful information.

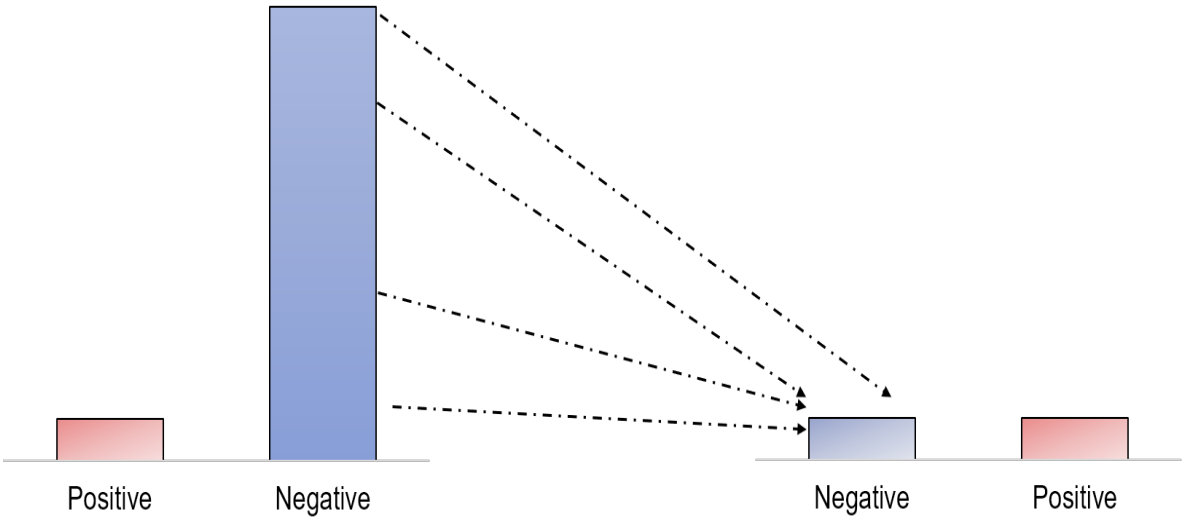

Figure 3.1: Random under-sampling

3.3.2 Random oversampling

Random oversampling (Figure 3.2) is also another simple approach for handling class imbalance. It is a non-heuristic approach where examples of the minority class are randomly replicated from the data set. In this study, the minority class is the number of children who have positively tested for malaria. Therefore in this method, we randomly replicated malaria positive cases to obtain a balanced class distribution. The main disadvantage of this approach is that; replication of some instances of the minority class leads to a significant increase in the size of the training data, and therefore this can increase time required in fitting a statistical learning model. Furthermore, this approach may result in an over-fitted model.

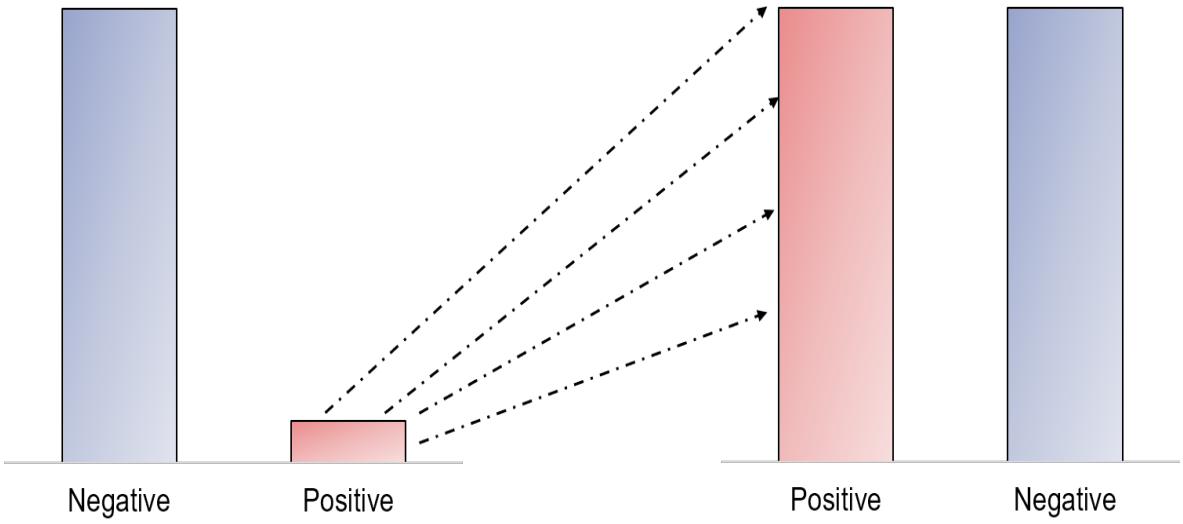

Figure 3.2: Random over-sampling

3.3.3 Synthetic Minority Over-sampling Technique

Suggested by Chawla et al. (2002), Synthetic Minority Oversampling Technique (SMOTE), rebalances the training data through oversampling. In contrast to random oversampling, the SMOTE algorithm involves generating synthetic examples for the minority class (malaria positive in this case). Positive class examples that are close are selected, and then a line drawn between them. A new sample is then drawn along this line. For this reason, this approach is

said to operate on the "feature space" rather than the "data space" and thus it is a less of an application specific method Chawla et al. (2002). Formally, a decision on the needed amount of over-sampling is made, usually that leads to a 50:50 class distribution. After which, a positive class example is selected randomly from the training set and its k-nearest neighbours (say 5) are determined. Lastly, n of these k examples are randomly selected to generate new instances via interpolation. This is done by taking the difference between the considered feature vector or sample and its nearest neighbour Chawla et al. (2002). This difference is then multiplied by any number randomly drawn from (0,1) and then adding to the considered feature vector. This process leads to a random point being selected between features that lie together Chawla et al. (2002).

3.3.4 Adaptive snythetic sampling Approach

Adaptive snythetic sampling Approach (ADASYN) He et al. (2008) is an adaptive technique for learning from imbalanced data motivated by the success of approaches that involve generation of synthetic data such as Synthetic Minority Over-sampling Technique (SMOTE). The objective of ADASYN is to reduce bias and learn adaptively. More synthetic examples that are more difficult to learn from the minority class are generated. This forces the learning algorithm to focus on these examples. This is the major difference between SMOTE and ADASYN. The procedure for ADASYN algorithm He et al. (2008) is as follows:

1. Consider the training data set D_{train} with n examples $\{x_i, y_i\}, i = 1, \ldots, n$. $x_i \in X$, is an instance of the feature space and $y_i \in Y = \{\mathrm{P}, \mathrm{N}\}$ is the response label, positive or negative, associated with x_i.

2. Define n_p and n_n be the number of minority (positive) class instances and the number of majority (negative) class instances, respectively in the training data, D_{train}. Then, $n_p \leq n_n$ and $n_p + n_n = n$.

3. Calculate class imbalance degree, $d = \frac{n_p}{n_n}$ where $d \in (0, 1]$.

4. Let d_{th} be the preset threshold ratio for the maximum tolerated degree of class imbalance. If $d < d_{th}$:

(a) Determine the number of synthetic minority (positive) class examples that need to be generated as $G = (n_n - n_p) \times \beta$, where $\beta \in [0,1]$ is the required balance level after synthetic data is generated. If $\beta = 1$, then a fully balanced data is created after the generalized process.

(b) For each $x_i \in$ minority class (P), use the Euclidean distance measure to find the K nearest neighbours of x_i. Then, calculate the ratio $r_i = \Delta_i / K$, where $i = 1, \ldots, n_p$ and Δ_i is the number of examples in the K nearest neighbours of x_i which belong to the majority (negative) class. Hence, $r_i \in [0,1]$.

(c) Normalize r_i by using $\hat{r}_i = r_i / \sum_{i=1}^{n_p} r_i$, such that $\hat{r}_i$ is a density distribution ($\sum_i \hat{r}_i = 1$).

(d) For each minority (positive) instance x_i, determine the number of synthetic instances that will be generated, $g_i = \hat{r}_i \times G$, where G is from 4 (a).

(e) For each positive (minority) class, generate the g_i examples: From 1 to g_i,

i. From the K nearest neighbours of x_i, choose randomly one positive (minority) example,

ii. Generate the synthetic example, $s_i = x_i + (x_{zi} - x_i)$, where $\lambda \in [0,1]$ is a random number and $(x_{zi} - x_i)$ is a difference vector.

3.3.5 Random Over-sampling Examples

Random over-sampling examples (ROSE) Menardi and Torelli (2014) is based on a smoothed bootstrap method to generate new artificial examples from the classes. Let D_{train} be the training data set with N samples $\{x_i, y_i\}$ where $i = 1, \ldots, N$, $x_i \in x$ is an example in the dimension m feature space, and $y_i \in C = \{Y_0, Y_1\} = \{\text{Positive}, \text{Negative}\}$ the associated class label. Let the probability density function of $x \in R^d$ be $f(x)$ and N_j the number of examples that belong to class Y_j. Then, the ROSE procedure is as follows Leevy et al. (2018):

1. Select $y^* = Y_j$ with probability π_j.

2. Select $\{x_i, y_j\} \in D_{train}$ in such a way that $y_i = y^*$, with probability $1/N_j$.

3. Sample x^* from $K_{H_j(.,x_i)}$, where K_{H_j} is a probability distribution centered at x_i and covariance matrix H_j.

Basically, ROSE involves drawing an example from one of the two classes and generating a new instance in the neighborhood $\{x^*, y'^*\}$. The shape of the countour sets of K and the covariance matrix H_j determine the shape and width of this neighbourhood, respectively. Repeating steps 1 to 3 leads to as many examples as one may want. Adjusting the probability π_j is used to balance the new data as desired.

3.3.6 Random under-sampling and Random over-sampling (RURO)

As noted before, both random under-sampling and random over-sampling have drawbacks. The main problem with random under-sampling is the loss of potentially useful data. On the other hand, random over-sampling may lead to an over-fitted model and increased time in the model fitting process. To address these issues, we propose a method that combines these two approaches. Below is the description of the RURO algorithm.

1. Consider an imbalanced training data set D_{train} with n examples $\{x_i, y_i\}, i = 1, \ldots, n$ where $x_i \in X$, is an example of the feature space and $y_i \in Y = \{\mathrm{P}, \mathrm{N}\}$ is the associated binary response.

2. Let n_p and n_n be the number of minority class instances and the number of majority class instances, respectively in D_{train} such that $n_p < n_n$ and $n_p + n_n = n$.

3. Obtain D^u_{train} by randomly under-sampling the majority class N in D_{train}.

4. Obtain D^o_{train} by randomly over-sampling the majority class P in D_{train}.

5. Obtain D^a_{train} by combining D^u_{train} and D^o_{train}.

6. Obtain D^{ou}_{train} by randomly sampling examples of size n from D^a_{train}. Note that n is the size of the original training data.

7. D^{uo}_{train} is the new balanced training data set.

3.4 Implementation

3.4.1 Software

The modelling is performed using R version 4.1.3 and R studio as the integrated development environment. R "meta packages" **tidyverse** Wickham et al. (2019) for data import, manipulation and visualization; and **tidymodels** Kuhn and Wickham (2020) for model development using tidy principles. The **stargazer** package Hlavac (2015) was used to translate R statistical tables into a Latex code.

3.4.2 Training and testing

When modelling, it is important to make a decision on how data will be spent at the initial stage. This is because it is related to the empirical validation of the models developed thereafter. The commonly used approach to empirically evaluate model performance is by splitting the existing data into two sets, that is: the training set and test set. Usually, the training set forms the bulk of the split. In this paper, 80% of the original Division of National Malaria Programme, Ministry of Health, Kenya et al. (2021) malaria indicator survey data set is used as the training set. Further, 20% of the data was held as a reserve to evaluate the performance of the best performing model. This was the final arbitration to determine the model's efficacy.

3.4.3 Validation

In modern statistics, resampling methods are indispensable James et al. (2013). They are used to obtain additional information about a model of interest by repeatedly drawing samples from the training set. For instance, resampling methods can help us to determine the variability of a model. There are two commonly used approaches for resampling, that is, cross-validation and the bootstrap. In this paper, we have used k-fold cross validation. It involves dividing the training data into k folds (groups) of approximately equal size James et al. (2013). The first fold was used as the validation set and the rest $k-1$ folds to fit the model. Overall, to compare the predictive performance of the statistical learning methods using the various

approaches for handling class imbalance, 5 fold cross validated metrics such as F1 score, sensitivity, specificity, precision and accuracy were used. Finally, the performance of the best performing model was evaluated on the unseen set.

Chapter 4

Results

In this section, we present the results obtained when we fitted different statistical learning models on the imbalanced malaria data set using the different methods for handling class imbalance discussed in chapter three. Furthermore, we also present the effect of combining randomly undersampled and randomly oversampled data on the performance of the learning algorithms. The section is divided into two subsections i.e. summary of study data and the statistical learning results.

4.1 Summary of the study data

Table 4.1 provides a descriptive summary of the potential predictors for malaria infection among children aged between 6 months and 14 years. Overall, out of the 11,549 observations, 969 (8.3%) were positive for malaria by microscopy. In general, there was a significant association (p-value < 0.001) between the 9 predictors and the outcome variable. As further illustrated by figure 4.1, the children with malaria (red box) were on average older compared with the children without malaria (blue box). Similarly, malaria incidence among children was on average higher in households where the head was older. Further, the malaria positive cases were on average higher in households with many members. Figure 4.2 indicates that majority of the malaria positive cases were observed in highland epidemic, lake endemic and coastal endemic regions compared to the seasonal and low risk areas. The highest malaria cases among children was highest in the rural areas, households without television, or which did not have their water sources piped into their yard or dwelling or those that were poor.

Table 4.1: Summary of the study data

Predictor	**Negative, N = 10,580**	**Positive, N = 969**	**p-value**
Age of child (months)	86 (45, 131)	111 (72, 147)	<0.001
Age of household head (years)	42 (35, 53)	45 (36, 57)	<0.001
Number of household members	6.00 (4.00, 7.00)	6.00 (5.00, 8.00)	<0.001
Malaria endemicity zone			<0.001
Highland epidemic	2,089 (20%)	33 (3.4%)	
Lake endemic	3,855 (36%)	842 (87%)	
Coastal endemic	1,153 (11%)	60 (6.2%)	
Seasonal	2,177 (21%)	33 (3.4%)	
Low risk	1,306 (12%)	1 (0.1%)	
Availability of mosquito bed net	7,285 (69%)	751 (78%)	<0.001
Residence			<0.001
Urban	3,759 (36%)	148 (15%)	
Rural	6,821 (64%)	821 (85%)	
Wealth index			<0.001
Poorest	2,924 (28%)	304 (31%)	
Poorer	2,350 (22%)	322 (33%)	
Middle	2,168 (20%)	210 (22%)	
Richer	1,993 (19%)	108 (11%)	
Richest	1,145 (11%)	25 (2.6%)	
Household has television	4,439 (42%)	311 (32%)	<0.001
Source of drinking water			<0.001
Piped into dwelling, yard/ plot	1,827 (17%)	36 (3.7%)	
Other	8,753 (83%)	933 (96%)	

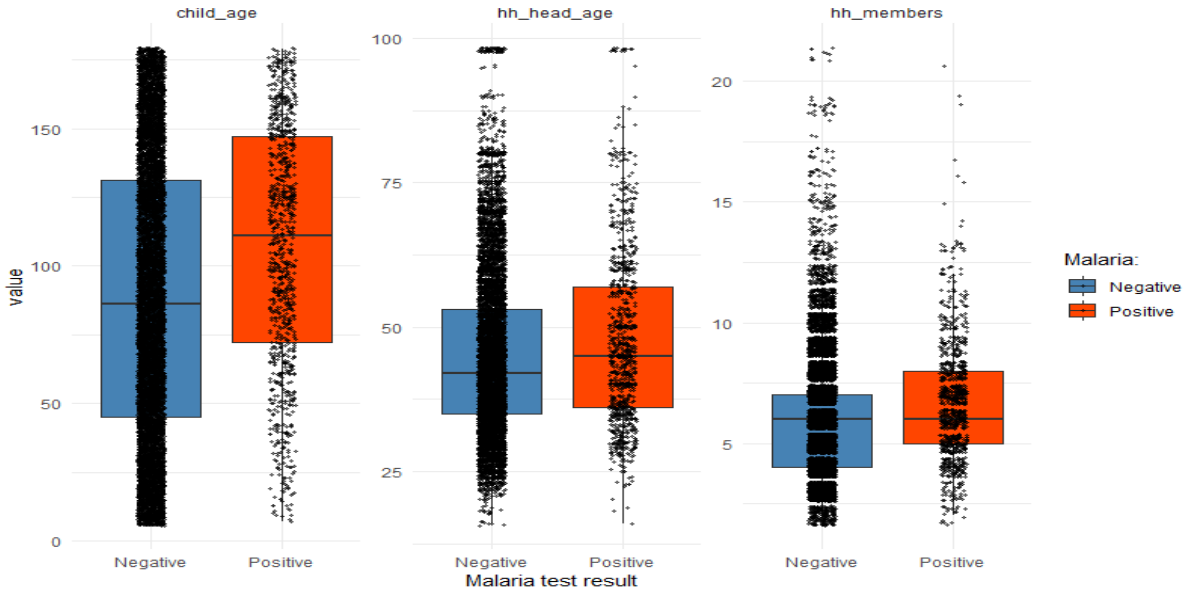

Figure 4.1: Bivariate relations between continous predictors and malaria infection

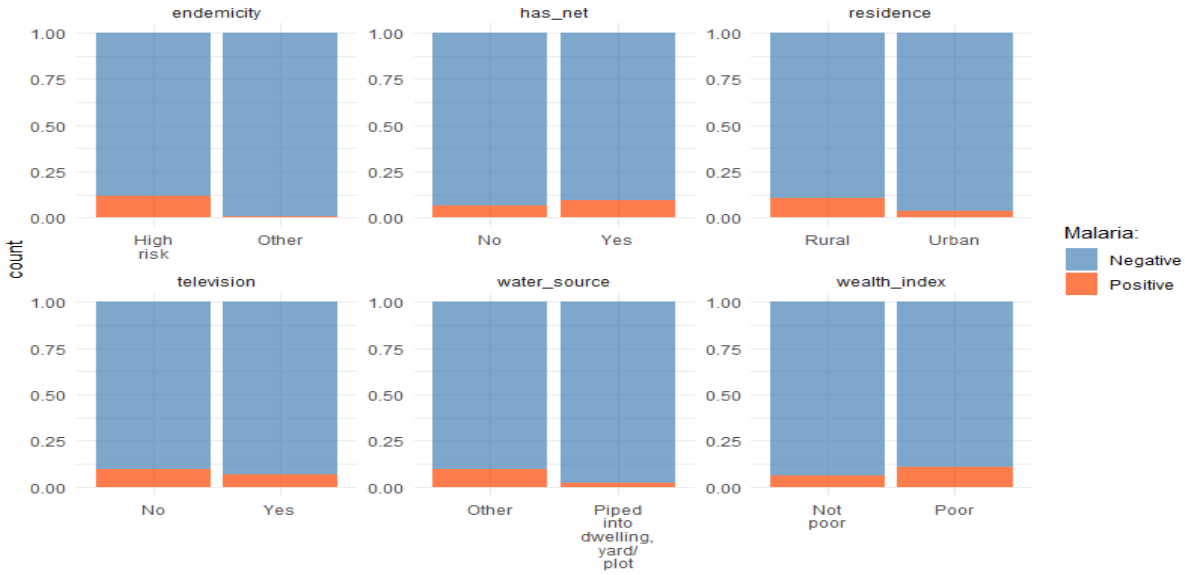

Figure 4.2: Bivariate relations between categorical predictors and malaria infection

4.2 Model Performance

4.2.1 Classical learning

Table 4.2 provides a summary of the results obtained from fitting classical statistical learning algorithms using 5-fold cross validation. The two classical models considered were logistic

regression and the support vector machines. The methods for handling class imbalance included, none, random undersampling, random oversampling, SMOTE, ADASYN, ROSE. We further explored whether combining random undersampling and random oversampling (RURO) had an effect on performance of the classical models.

For both logistic regression and the support vector machines, the accuracy was high (more than 91.3 %) when the models were fitted without handling class imbalance. The specificity for both models was also high (100 %) while the sensitivity was very low (0 %). The area under the receiver operating characteristic curve (ROC_AUC) was 82.3% and 78.5% for logistic regression and the support vector machines, respectively. It was not possible to determine the F1 score and precision values.

Table 4.2: Classical statistical learning

	Model	.metric	None	down	up	smote	adasyn	rose	ruro
1	LR	accuracy	0.913	0.696	0.698	0.703	0.697	0.701	0.769
2	LR	f_meas	–	0.324	0.326	0.328	0.323	0.326	0.789
3	LR	precision	–	0.201	0.202	0.204	0.201	0.203	0.733
4	LR	roc_auc	0.823	0.819	0.823	0.822	0.821	0.821	0.828
5	LR	sensitivity	0	0.839	0.840	0.834	0.832	0.832	0.855
6	LR	specificity	1	0.682	0.684	0.691	0.685	0.688	0.681
7	SVM	accuracy	0.913	0.682	0.707	0.834	0.833	0.622	0.797
8	SVM	f_meas	–	0.322	0.338	0.376	0.372	0.296	0.821
9	SVM	precision	–	0.198	0.211	0.279	0.277	0.177	0.742
10	SVM	roc_auc	0.785	0.818	0.827	0.824	0.820	0.828	0.853
11	SVM	sensitivity	0	0.869	0.864	0.575	0.570	0.914	0.919
12	SVM	specificity	1	0.664	0.692	0.859	0.858	0.594	0.673

On applying various methods to handle imbalanced class distribution, there was an overall reduction in the accuracy of the classical models. Among the widely used methods, Synthetic Minority Oversampling Technique (SMOTE) resulted in the highest accuracy, i.e. 70.3 % for LR and 83.4 % for the SVM. Further, there was a reduction in specificity across all widely used sub-sampling methods. The SMOTE algorithm generally performed better than the other methods available in literature, including the proposed RURO approach. However, in terms of sensitivity, SMOTE was not the best among the common methods. The RURO approach performed well in this metric than any other method considered. The ROC_AUC

for logistic regression was generally lower after re-sampling. On the other hand, the metric improved for the SVM model. In general, the RURO approach outperformed all the methods in terms of ROC_AUC. In contrast to accuracy, there was an improvement in the F1 score from undetermined to at least 29 %. The SMOTE algorithm performed best compared to the common re-sampling approaches. However, RURO had the highest F1 score, 78.9 % and 82.1 % for logistic regression and support vector machines, respectively. A similar trend was noted for precision.

4.2.2 Ensemble learning

Table 4.3 provides a summary of the results obtained from fitting algorithms from the ensemble framework using 5-fold cross validation. The two ensemble models considered were the random forest (RF) and the Extreme Gradient Boosting (XGBoost). The methods for handling class imbalance included, none, random undersampling, random oversampling, SMOTE, ADASYN, ROSE. We also explored whether stacking random undersampling and random oversampling (RURO) had an effect on performance of the ensemble models.

For both the random forest and extreme gradient boosting, the accuracy was high (more than 90 %) when the models were fitted without handling class imbalance. The specificity for both models was also high i.e. 100 % for RF and 98 % for XGBoost. Just like the classical models, the sensitivity was very low. However, the ensemble methods were at least able to make some true positive predictions. The XGBoost model returned a 12.4 % sensitivity which was high compared to the other models that had almost 0 % without handling class imbalance. The area under the receiver operating characteristic curve (ROC_AUC) was 84.8% and 82.4% for the random forest and the XGBoost, respectively. In constrast to the classical models, where it was not possible to determine the F1 score and precision values, it was possible to estimate these metrics. The RF model had 0 % for both metrics while the XGoost had 18.3 % and 36.7 % scores for the F1 score and precision, respectively.

When different methods to handle imbalanced class distribution were used, there was an overall reduction in the accuracy of the classical models. Among the widely used methods, both Synthetic Minority Oversampling Technique (SMOTE) and Adaptive Synthetic Sam-

Table 4.3: Ensemble learning

	Model	.metric	None	down	up	smote	adasyn	rose	ruro
1	RF	accuracy	0.913	0.721	0.773	0.887	0.888	0.222	0.851
2	RF	f_meas	0	0.338	0.367	0.268	0.258	0.183	0.867
3	RF	precision	0	0.213	0.242	0.315	0.307	0.101	0.790
4	RF	roc_auc	0.848	0.833	0.850	0.840	0.840	0.814	0.932
5	RF	sensitivity	0	0.821	0.760	0.235	0.223	1	0.960
6	RF	specificity	1.000	0.712	0.774	0.949	0.952	0.148	0.739
7	XGBoost	accuracy	0.906	0.718	0.847	0.904	0.903	0.199	0.906
8	XGBoost	f_meas	0.183	0.323	0.347	0.208	0.202	0.178	0.914
9	XGBoost	precision	0.367	0.205	0.276	0.372	0.357	0.098	0.856
10	XGBoost	roc_auc	0.824	0.802	0.819	0.826	0.823	0.818	0.961
11	XGBoost	sensitivity	0.124	0.775	0.469	0.146	0.142	1	0.979
12	XGBoost	specificity	0.980	0.713	0.883	0.976	0.976	0.123	0.832

pling (ADASYN) resulted in the highest accuracy. In addition, there was a reduction in the specificity of each ensemble model across all methods for handling class imbalance that were considered. The SMOTE and ADASYN algorithm generally performed better than the other methods available in literature, including the proposed RURO approach. However, in terms of sensitivity, both SMOTE and ADASYN performed poorly compared to other methods. The ROSE algorithm performed best, yielding 100 % sensitivity but very poor specificity of about 12-14 %. . The ROC_AUC for both models was generally lower after re-sampling. However, the RURO approach improved the outcome of this metric for both models. In contrast to accuracy, there was an improvement in the F1 score from 0 % for RF and 18.3 % for XGBoost to 36.7 % and 34.3 % among the widely used algorithms. The RURO approach resulted in the highest performance, 86.7 % for RF and 91.4 % for the XGBoost. Similar results were noted for precision. Based on these results, we can conclude that if the outcome of interest was the minority (positive) cases, then the XGBoost model paired with the RURO class imbalance handling approach performed best. Figure 4.3 shows the ROC curve of the XGBoost model as fitted using RURO for each fold.

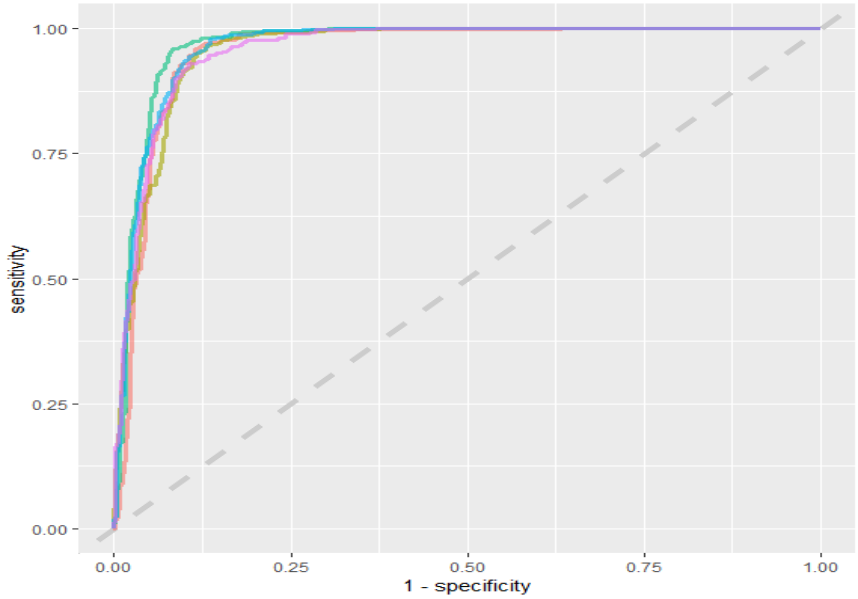

Figure 4.3: XGBoost model fitted on each RURO fold

4.2.3 Performance on test set

From the previous section, we noticed that the best performing model was the XGboost fitted on a balanced data set by the RURO approach. However, this result is an early estimate or peek on the the performance of the final model. Russell and Norvig (2002), suggested the importance of completely locking the test set away from model development. This is now a common approach in statistical learning, and for this reason, 20 % of the overall data set was set aside for the final evaluation of the best performing model. As shown in table 4.4 below, the model generally resulted in improved prediction on unseen test set at the expense of a slight decline in specificity.

Table 4.4

	.metric	.estimate
1	accuracy	0.939
2	sens	0.995
3	spec	0.881
4	precision	0.896
5	f_meas	0.943

Chapter 5

Discussion

5.1 Key findings

This paper explored the performance of both classical and ensemble learning methods using different methods to handle class imbalance. The learning algorithms considered were: Logistic regression (LR), support vector machines (svm), Random Forest (RF), and eXtreme Gradient Boosting (XGBoost). Kenya's 2020 Malaria Indicator Survey data was used as the study data set. First, the algorithms were first trained on an imbalanced data set. Secondly, the models were fitted after balancing the data using common methods such as: random under-sampling, random over-sampling, Synthetic Minority Over-sampling technique (SMOTE), Adaptive Synthetic Sampling Approach (ADASYN), and Random Over Sampling Examples (ROSE) techniques. In addition, we evaluated the performance of these models when trained on a data set that was derived by combining random under-sampling and random over-sampling. Eighty percent of the data was used for model training, and the rest reserved for evaluating performance of the best performing model.

As expected, for both the classical and ensemble methods, the accuracy was high (more than 90%) when the models were fitted without handling class imbalance. This is because the models are biased towards the majority (negative) class. For this reason, the models correctly predicted the negative cases (high specificity) and performed poorly in identifying the positive cases (almost 0 % sensitivity). Generally, in all aspects, the ensemble methods outperformed their classical counterparts. The Extreme Gradient Boosting (XGBoost) performed better than the random forest in all approaches except for the random undersampling method. Overall, there were strengths and weaknesses in each method that was considered. For example, the random under-sampling method performed poorly in the F1 score and precision metrics and relatively well on the others. The random over-sampling method performed poorly on the F1 score, precision and sensitivity. This was a similar trend for SMOTE

and ADASYN. The ROSE approach performed better on sensitivity compared to the other metrics. Finally, the proposed approach that involved combining random undersampling and random oversampling approaches (RURO) performed well on generally all metrics compared to any of the existing methods. This was further confirmed by the results from the unseen test set.

5.2 Limitations

Although the results from this study are promising, it is important to note that a single data set was used. Hence, we do not have information on how the RURO approach can perform from data set to data set. Therefore, it will be important to study this proposed approach using numerous data sets. Furthermore, there exists a number of approaches to handle class imbalance such as Tomek's links, B-SMOTE, Near miss et cetera that were not considered. Furthermore, this paper did not focus on class imbalance within the independent variables which also has the potential to affect the results obtained. The other limitation was that, this paper used 5-fold cross validation. However, evidence in statistical learning literature suggests that we can get better performance by using repeated k fold validation where we can have more than 5 folds and at least 2 repeats.

5.3 Conclusion and recommendations

Class imbalanced problems are common in the real world. There's no doubt about their negative effect on the performance of classification models. The main focus of this paper was to evaluate the predictive performance of various statistical learning algorithms by using different data level approaches for handling class imbalance. In addition, we explored whether combining random under-sampling and random over-sampling can result in improved performance of the learning algorithms. In general, random under-sampling, random-oversampling, SMOTE, ADASYN and ROSE all had strengths and weaknesses depending on the performance metric. Evidence from this paper indicates that using our proposed approach, RURO, that involves combining randomly under-sampled and randomly

over-sampled data is a simple yet potentially powerful method to improve performance of a classification algorithm when the interest is to get a trade-off between sensitivity and specificity. We recommend further research to explore how the RURO approach will compare between numerous datasets and a software implementation of the approach.

References

Batista, G. E., Prati, R. C., and Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD explorations newsletter*, 6(1):20–29.

Burez, J. and Van den Poel, D. (2009). Handling class imbalance in customer churn prediction. *Expert Systems with Applications*, 36(3):4626–4636.

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357.

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

Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., Chen, K., et al. (2015). Xgboost: extreme gradient boosting. *R package version 0.4-2*, 1(4):1–4.

Division of National Malaria Programme, Ministry of Health, Kenya, Kenya National Bureau of Statistics, and The DHS Program (2021). Kenya Malaria Indicator Survey 2020.

Fawcett, T. (2005). Roc analysis in pattern recognition. *Pattern Recognition Letters*, 8:861–874.

Fernández, A., García, S., Galar, M., Prati, R. C., Krawczyk, B., and Herrera, F. (2018). *Learning from imbalanced data sets*, volume 10. Springer.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232.

Guo, X., Yin, Y., Dong, C., Yang, G., and Zhou, G. (2008). On the class imbalance problem. In *2008 Fourth international conference on natural computation*, volume 4, pages 192–201. IEEE.

Habyarimana, F. and Ramroop, S. (2020). Prevalence and risk factors associated with malaria among children aged six months to 14 years old in rwanda: evidence from 2017 rwanda malaria indicator survey. *International Journal of Environmental Research and Public Health*, 17(21):7975.

Hastie, T., Tibshirani, R., Friedman, J. H., and Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer.

He, H., Bai, Y., Garcia, E. A., and Li, S. (2008). Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*, pages 1322–1328. IEEE.

Hlavac, M. (2015). stargazer: beautiful latex, html and ascii tables from r statistical output.

Huda, S., Liu, K., Abdelrazek, M., Ibrahim, A., Alyahya, S., Al-Dossari, H., and Ahmad, S. (2018). An ensemble oversampling model for class imbalance problem in software defect prediction. *IEEE access*, 6:24184–24195.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An introduction to statistical learning*, volume 112. Springer.

Kuhn, M. and Wickham, H. (2020). Tidymodels: a collection of packages for modeling and machine learning using tidyverse principles. *Boston, MA, USA.[(accessed on 10 December 2020)]*.

Leevy, J. L., Khoshgoftaar, T. M., Bauder, R. A., and Seliya, N. (2018). A survey on addressing high-class imbalance in big data. *Journal of Big Data*, 5(1):1–30.

Ling, C. X. and Sheng, V. S. (2008). Cost-sensitive learning and the class imbalance problem. *Encyclopedia of machine learning*, 2011:231–235.

Longadge, R. and Dongre, S. (2013). Class imbalance problem in data mining review. *arXiv preprint arXiv:1305.1707*.

Menardi, G. and Torelli, N. (2014). Training and assessing classification rules with imbalanced data. *Data mining and knowledge discovery*, 28(1):92–122.

Russell, S. and Norvig, P. (2002). Artificial intelligence: a modern approach.

StataCorp, L. (2019). Stata statistical software: Release vol. 16.

Sun, Y., Wong, A. K., and Kamel, M. S. (2009). Classification of imbalanced data: A review. *International journal of pattern recognition and artificial intelligence*, 23(04):687–719.

Weiss, G. M. and Provost, F. (2003). Learning when training data are costly: The effect of class distribution on tree induction. *Journal of artificial intelligence research*, 19:315–354.

WHO (2020). The top 10 causes of death. https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death. Accessed: 2021-03-02.

WHO (2021). *Word Malaria Report 2021*.

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., Grolemund, G., Hayes, A., Henry, L., Hester, J., et al. (2019). Welcome to the tidyverse. *Journal of open source software*, 4(43):1686.

Winskill, P., Rowland, M., Mtove, G., Malima, R. C., and Kirby, M. J. (2011). Malaria risk factors in north-east tanzania. *Malaria Journal*, 10(1):1–7.

Appendix A

Appendix

appendix1-imbalanced-learning

Required libraries

```
library(tidyverse)
library(haven)
library(tidymodels)
library(themis)
library(discrim)
library(MASS)
library(stargazer)
library(gtsummary)
library(kableExtra)
library(conflicted)

tidymodels_prefer()
conflict_prefer("roc_auc", "yardstick")
conflict_prefer("sensitivity", "yardstick")
conflict_prefer("specificity", "yardstick")
conflict_prefer("precision", "yardstick")
conflict_prefer("f_meas", "yardstick")
conflict_prefer("accuracy", "yardstick")
```

Malaria data

```
PersonRecode <- read_dta("KE_2020_MIS/Household Member Recode/KEPR81FL.DTA")
dim(PersonRecode)
```

Feature selection

```
mal_data <- PersonRecode %>%
  dplyr::select(hml32, # result of microscopy test
                sml16a, # child's age in months
                hv220, # age of household head
                hv009, # number of household members
                shzone, # malaria endemicity zone/ county
                hv227, # has mosquito bed net for sleeping
                hv025, # residence
                hv270, # wealth index
                hv208, # household has television
                hv201  # source of drinking water
                ) %>%
  filter(!is.na(hml32))

glimpse(mal_data)
```

Variable transformation

```
malaria <- mal_data %>%
  rename(malaria = hml32,
         child_age = sml16a,
```

```
          hh_head_age = hv220,
          hh_members = hv009,
          endemicity = shzone,
          water_source = hv201,
          has_net = hv227,
          television = hv208,
          residence = hv025,
          wealth_index = hv270
          ) %>%
  mutate(water_source = case_when(water_source == 11 ~ 0,
                                  water_source == 12 ~ 0,
                                  TRUE ~ 1
                                  )
          )
# convert to numeric and factors where necessary
malaria$malaria <- as.factor(malaria$malaria)
levels(malaria$malaria)[1] <- "Negative"
levels(malaria$malaria)[2] <- "Positive"
malaria$child_age <- as.numeric(malaria$child_age)
malaria$hh_head_age <- as.numeric(malaria$hh_head_age)
malaria$hh_members <- as.numeric(malaria$hh_members)
malaria$has_net <- as.factor(malaria$has_net)
malaria$endemicity <- as.factor(malaria$endemicity)
malaria$residence <- as.factor(malaria$residence)
malaria$wealth_index <- as.factor(malaria$wealth_index)
malaria$television <- as.factor(malaria$television)
malaria$water_source <- as.factor(malaria$water_source)
# view results
glimpse(malaria)
```

Table 1

```
# summarize the data
table1data <- malaria

table1data$endemicity <- as.factor(as.integer(table1data$endemicity))
levels(table1data$endemicity) <- c("Highland epidemic",
                                   "Lake endemic",
                                   "Coastal endemic",
                                   "Seasonal",
                                   "Low risk"
                                   )

table1data$has_net <- as.factor(as.integer(table1data$has_net))
levels(table1data$has_net) <- c("No","Yes")

table1data$residence <- as.factor(as.integer(table1data$residence))
levels(table1data$residence) <- c("Urban","Rural")

table1data$television <- as.factor(as.integer(table1data$television))
levels(table1data$television) <- c("No","Yes")
```

```
table1data$water_source <- as.factor(as.integer(table1data$water_source))
levels(table1data$water_source) <- c("Piped into dwelling, yard/ plot","Other")

table1data$wealth_index <- as.factor(as.integer(table1data$wealth_index))
levels(table1data$wealth_index) <- c("Poorest",
                                     "Poorer",
                                     "Middle",
                                     "Richer",
                                     "Richest"
                                     )

table1 <- tbl_summary(table1data,
                      by = malaria, # split table by group
                      label = list(
                        child_age ~ "Age of child (months)",
                        hh_head_age ~ "Age of household head (years)",
                        hh_members ~ "Number of household members",
                        endemicity ~ "Malaria endemicity zone",
                        water_source ~ "Source of drinking water",
                        has_net ~ "Availability of mosquito bed net",
                        television ~ "Household has television",
                        residence ~ "Residence",
                        wealth_index ~ "Wealth index"
                      ),
                       missing = "no"
                       ) %>%
  add_p() %>% # test for a difference between groups
  modify_header(label = "**Variable**")  # update the column header

table1
# latex
# as_kable(table1, format = "latex")
```

Bivariate relations between continous predictors and the outcome

```
continuous_predictors <- malaria %>% dplyr::select(malaria, child_age, hh_members, hh_head_age) %>%
  pivot_longer(!malaria, values_to = "value") %>%
  ggplot(aes(x=factor(malaria), y=value, fill=factor(malaria))) +
  geom_boxplot(outlier.shape = NA) + geom_jitter(size=.7, width=.1, alpha=.5) +
  scale_fill_manual(values=c("steelblue", "orangered1")) +
  labs(fill="Malaria:") +
  xlab("Malaria test result") +
  theme_minimal() +
  facet_wrap(~name, scales="free")

# continuous_predictors
```

Bivariate relations between categorical predictors and the outcome

```
categoricalViz <- table1data %>%
  mutate(
    endemicity = case_when(endemicity == "Highland epidemic" ~ "High risk",
                           endemicity == "Lake endemic" ~ "High risk",
                           endemicity == "Coastal endemic" ~ "High risk",
                           endemicity == "Seasonal" ~ "Other",
```

```
                            endemicity == "Low risk" ~ "Other",
                            TRUE ~ NA_character_
                            ),

    wealth_index = case_when(wealth_index == "Poorest" ~ "Poor",
                            wealth_index == "Poorer" ~ "Poor",
                            wealth_index == "Middle" ~ "Not poor",
                            wealth_index == "Richer" ~ "Not poor",
                            wealth_index == "Richest" ~ "Not poor",
                            TRUE ~ NA_character_
                            )

  )

categorical_predictors <- categoricalViz %>% dplyr::select(-c(child_age, hh_members, hh_head_age)) %>%
  pivot_longer(!malaria, values_to = "value") %>%
  ggplot(aes(x=factor(value), fill=factor(malaria))) +
  scale_fill_manual(values=c("steelblue", "orangered1")) +
  geom_bar(position="fill", alpha=.7)+
  theme_minimal() +
  labs(fill="Malaria:") +
  scale_x_discrete(labels = wrap_format(5)) +
  facet_wrap(~name, scales="free") +
  xlab("")

# categorical_predictors
```

Training, validation and testing specification.

```
set.seed(123)
malaria_split <- initial_split(malaria, strata = malaria)
malaria_train <- training(malaria_split)
malaria_test <- testing(malaria_split) # to estimate performance of selected model in new data


set.seed(345)
# create a set of resamples
malaria_folds <- vfold_cv(malaria_train, v = 5, repeats = 1) # change to 10 later
# malaria_folds
```

Model specification

```
# Classical statistical learning
# logistic regression
glm_spec <- logistic_reg() %>%
  set_engine("glm")

# support vector machines
svm_spec <- svm_rbf() %>%
  set_engine("kernlab") %>%
  set_mode("classification")

# Ensemble statistical learning
# random forest
rf_spec <- rand_forest(trees = 1000) %>% # enough trees
```

```
  set_engine("ranger") %>%
  set_mode("classification") # can be used used for classification or regression

# xgboost
xgboost_spec <- boost_tree(
  mode = "classification",
  trees = 100) # can change later to 1000
```

Modelling without handling class imbalance

```
recipe_none <- recipe(malaria ~. , data = malaria_train) %>%
  step_relevel(malaria, ref_level = "Positive") %>%
    step_string2factor(all_nominal(), -all_outcomes()) %>%
  step_dummy(all_nominal(), - malaria)

# recipe results
recipe_none %>%
  prep() %>%
  bake(new_data = NULL) %>%
  count(malaria)
```

Initialize no sampling workflow

```
workflow_none <- workflow() %>%
  add_recipe(recipe_none)
```

Train models using imbalanced data

```
set.seed(456)
lr_none <- workflow_none %>%
  add_model(glm_spec) %>%
  fit_resamples(
    resamples = malaria_folds,
    metrics = metric_set(roc_auc, sensitivity, specificity, precision, f_meas, accuracy),
    control = control_resamples(save_pred = TRUE)
  )


set.seed(456)
svm_none <- workflow_none %>%
  add_model(svm_spec) %>%
  fit_resamples(
    resamples = malaria_folds,
    metrics = metric_set(roc_auc, sensitivity, specificity, precision, f_meas, accuracy),
    control = control_resamples(save_pred = TRUE)
  )

set.seed(456)
rf_none <- workflow_none %>%
  add_model(rf_spec) %>%
  fit_resamples(
    resamples = malaria_folds,
    metrics = metric_set(roc_auc, sensitivity, specificity, precision, f_meas, accuracy),
    control = control_resamples(save_pred = TRUE)
  )
```

```
set.seed(456)
xgboost_none <- workflow_none %>%
  add_model(xgboost_spec) %>%
  fit_resamples(
    resamples = malaria_folds,
    metrics = metric_set(roc_auc, sensitivity, specificity, precision, f_meas, accuracy),
    control = control_resamples(save_pred = TRUE)
  )
```

```
none_results <- rbind(
  collect_metrics(lr_none) %>%
  mutate(Model = "LR",
         Resampling = "None"),
  collect_metrics(svm_none) %>%
  mutate(Model = "SVM",
         Resampling = "None"),
  collect_metrics(rf_none) %>%
  mutate(Model = "RF",
         Resampling = "None"),
  collect_metrics(xgboost_none) %>%
  mutate(Model = "XGBoost",
         Resampling = "None")
)

head(none_results)
```

```
nonestargazer_classical <- none_results %>%
  select(Resampling, Model, .metric, mean) %>%
  filter(Model %in% c("LR", "SVM")) %>%
  as.data.frame()

nonestargazer_ensemble <- none_results %>%
  select(Resampling, Model, .metric, mean)%>%
  filter(Model %in% c("RF", "XGBoost"))%>%
  as.data.frame()

# stargazer(nonestargazer, summary = FALSE)
```

Downsampling

```
recipe_down <- recipe(malaria ~. , data = malaria_train) %>%
  step_relevel(malaria, ref_level = "Positive") %>%
  step_string2factor(all_nominal(), -all_outcomes()) %>%
  step_dummy(all_nominal(), - malaria) %>%
  themis::step_downsample(malaria, seed = 101)

# recipe results
recipe_down %>%
  prep() %>%
  bake(new_data = NULL) %>%
  count(malaria)
```

Initialize downsampling workflow

```
workflow_down <- workflow() %>%
  add_recipe(recipe_down)
```

Train models using downsampled data

```
set.seed(456)

lr_down<- workflow_down %>%
  add_model(glm_spec) %>%
  fit_resamples(
    resamples = malaria_folds,
    metrics = metric_set(roc_auc, sensitivity, specificity, precision, f_meas, accuracy),
    control = control_resamples(save_pred = TRUE)
  )

set.seed(456)
svm_down <- workflow_down %>%
  add_model(svm_spec) %>%
  fit_resamples(
    resamples = malaria_folds,
    metrics = metric_set(roc_auc, sensitivity, specificity, precision, f_meas, accuracy),
    control = control_resamples(save_pred = TRUE)
  )

set.seed(456)
rf_down <- workflow_down %>%
  add_model(rf_spec) %>%
  fit_resamples(
    resamples = malaria_folds,
    metrics = metric_set(roc_auc, sensitivity, specificity, precision, f_meas, accuracy),
    control = control_resamples(save_pred = TRUE)
  )

set.seed(456)
xgboost_down <- workflow_down %>%
  add_model(xgboost_spec) %>%
  fit_resamples(
    resamples = malaria_folds,
    metrics = metric_set(roc_auc, sensitivity, specificity, precision, f_meas, accuracy),
    control = control_resamples(save_pred = TRUE)
  )
```

```
down_results <- rbind(
  collect_metrics(lr_down) %>%
  mutate(Model = "LR",
         Resampling = "down"),
  collect_metrics(svm_down) %>%
  mutate(Model = "SVM",
         Resampling = "down"),
  collect_metrics(rf_down) %>%
  mutate(Model = "RF",
         Resampling = "down"),
  collect_metrics(xgboost_down) %>%
  mutate(Model = "XGBoost",
```

```
         Resampling = "down")
)

head(down_results)
```

Upsampling

```
recipe_up <- recipe(malaria ~. , data = malaria_train) %>%
  step_relevel(malaria, ref_level = "Positive") %>%
  step_string2factor(all_nominal(), -all_outcomes()) %>%
  step_dummy(all_nominal(), - malaria) %>%
  themis::step_upsample(malaria, seed = 101)

# recipe results
recipe_up %>%
  prep() %>%
  bake(new_data = NULL) %>%
  count(malaria)
```

Initialize upsampling workflow

```
workflow_up <- workflow() %>%
  add_recipe(recipe_up)
```

Train models upsampled data

```
set.seed(456)
lr_up <- workflow_up %>%
  add_model(glm_spec) %>%
  fit_resamples(
    resamples = malaria_folds,
    metrics = metric_set(roc_auc, sensitivity, specificity, precision, f_meas, accuracy),
    control = control_resamples(save_pred = TRUE)
  )

set.seed(456)
svm_up <- workflow_up %>%
  add_model(svm_spec) %>%
  fit_resamples(
    resamples = malaria_folds,
    metrics = metric_set(roc_auc, sensitivity, specificity, precision, f_meas, accuracy),
    control = control_resamples(save_pred = TRUE)
  )


set.seed(456)
rf_up <- workflow_up %>%
  add_model(rf_spec) %>%
  fit_resamples(
    resamples = malaria_folds,
    metrics = metric_set(roc_auc, sensitivity, specificity, precision, f_meas, accuracy),
    control = control_resamples(save_pred = TRUE)
  )

set.seed(456)
xgboost_up <- workflow_up %>%
```

```
  add_model(xgboost_spec) %>%
  fit_resamples(
    resamples = malaria_folds,
    metrics = metric_set(roc_auc, sensitivity, specificity, precision, f_meas, accuracy),
    control = control_resamples(save_pred = TRUE)
  )
```

```
up_results <- rbind(
  collect_metrics(lr_up) %>%
  mutate(Model = "LR",
         Resampling = "up"),
  collect_metrics(svm_up) %>%
  mutate(Model = "SVM",
         Resampling = "up"),
  collect_metrics(rf_up) %>%
  mutate(Model = "RF",
         Resampling = "up"),
  collect_metrics(xgboost_up) %>%
  mutate(Model = "XGBoost",
         Resampling = "up")
)

head(up_results)
```

SMOTE

```
recipe_smote <- recipe(malaria ~. , data = malaria_train) %>%
  step_relevel(malaria, ref_level = "Positive") %>%
  step_string2factor(all_nominal(), -all_outcomes()) %>%
  step_dummy(all_nominal(), - malaria) %>%
  themis::step_smote(malaria, seed = 101)

# recipe results
recipe_smote %>%
  prep() %>%
  bake(new_data = NULL) %>%
  count(malaria)
```

Initialize SMOTE workflow

```
workflow_smote <- workflow() %>%
  add_recipe(recipe_smote)
```

Train models oversampling by smote data

```
set.seed(456)
lr_smote <- workflow_smote %>%
  add_model(glm_spec) %>%
  fit_resamples(
    resamples = malaria_folds,
    metrics = metric_set(roc_auc, sensitivity, specificity, precision, f_meas, accuracy),
    control = control_resamples(save_pred = TRUE)
  )


set.seed(456)
```

```
svm_smote <- workflow_smote %>%
  add_model(svm_spec) %>%
  fit_resamples(
    resamples = malaria_folds,
    metrics = metric_set(roc_auc, sensitivity, specificity, precision, f_meas, accuracy),
    control = control_resamples(save_pred = TRUE)
  )


set.seed(456)
rf_smote <- workflow_smote %>%
  add_model(rf_spec) %>%
  fit_resamples(
    resamples = malaria_folds,
    metrics = metric_set(roc_auc, sensitivity, specificity, precision, f_meas, accuracy),
    control = control_resamples(save_pred = TRUE)
  )

set.seed(456)
xgboost_smote <- workflow_smote %>%
  add_model(xgboost_spec) %>%
  fit_resamples(
    resamples = malaria_folds,
    metrics = metric_set(roc_auc, sensitivity, specificity, precision, f_meas, accuracy),
    control = control_resamples(save_pred = TRUE)
  )
```

```
smote_results <- rbind(
  collect_metrics(lr_smote) %>%
  mutate(Model = "LR",
         Resampling = "smote"),
  collect_metrics(svm_smote) %>%
  mutate(Model = "SVM",
         Resampling = "smote"),
  collect_metrics(rf_smote) %>%
  mutate(Model = "RF",
         Resampling = "up"),
  collect_metrics(xgboost_smote) %>%
  mutate(Model = "XGBoost",
         Resampling = "smote")
)

head(smote_results)
```

ADASYN

```
recipe_adasyn <- recipe(malaria ~. , data = malaria_train) %>%
  step_relevel(malaria, ref_level = "Positive") %>%
  step_string2factor(all_nominal(), -all_outcomes()) %>%
  step_dummy(all_nominal(), - malaria) %>%
  step_adasyn(malaria, seed = 101)

# recipe results
recipe_adasyn %>%
```

```
  prep() %>%
  bake(new_data = NULL) %>%
  count(malaria)
```

Initialize ADASYN workflow

```
workflow_adasyn <- workflow() %>%
  add_recipe(recipe_adasyn)
```

Train models oversampled by ADASYN

```
set.seed(456)
lr_adasyn <- workflow_adasyn %>%
  add_model(glm_spec) %>%
  fit_resamples(
    resamples = malaria_folds,
    metrics = metric_set(roc_auc, sensitivity, specificity, precision, f_meas, accuracy),
    control = control_resamples(save_pred = TRUE)
  )


set.seed(456)
svm_adasyn <- workflow_adasyn %>%
  add_model(svm_spec) %>%
  fit_resamples(
    resamples = malaria_folds,
    metrics = metric_set(roc_auc, sensitivity, specificity, precision, f_meas, accuracy),
    control = control_resamples(save_pred = TRUE)
  )

set.seed(456)
rf_adasyn <- workflow_adasyn %>%
  add_model(rf_spec) %>%
  fit_resamples(
    resamples = malaria_folds,
    metrics = metric_set(roc_auc, sensitivity, specificity, precision, f_meas, accuracy),
    control = control_resamples(save_pred = TRUE)
  )

set.seed(456)
xgboost_adasyn <- workflow_adasyn %>%
  add_model(xgboost_spec) %>%
  fit_resamples(
    resamples = malaria_folds,
    metrics = metric_set(roc_auc, sensitivity, specificity, precision, f_meas, accuracy),
    control = control_resamples(save_pred = TRUE)
  )
```

```
adasyn_results <- rbind(
  collect_metrics(lr_adasyn) %>%
  mutate(Model = "LR",
         Resampling = "adasyn"),
  collect_metrics(svm_adasyn) %>%
  mutate(Model = "SVM",
         Resampling = "adasyn"),
```

```
  collect_metrics(rf_adasyn) %>%
  mutate(Model = "RF",
         Resampling = "adasyn"),
  collect_metrics(xgboost_adasyn) %>%
  mutate(Model = "XGBoost",
         Resampling = "adasyn")
)

head(adasyn_results)
```

ROSE

```
recipe_rose <- recipe(malaria ~. , data = malaria_train) %>%
  step_relevel(malaria, ref_level = "Positive") %>%
  step_dummy(all_nominal(), - malaria) %>%
  step_rose(malaria, seed = 101)

# recipe results
recipe_rose %>%
  prep() %>%
  bake(new_data = NULL) %>%
  count(malaria)
```

Initialize ROSE workflow

```
workflow_rose <- workflow() %>%
  add_recipe(recipe_rose)
```

Train models using data balanced by ROSE

```
set.seed(456)
lr_rose <- workflow_rose %>%
  add_model(glm_spec) %>%
  fit_resamples(
    resamples = malaria_folds,
    metrics = metric_set(roc_auc, sensitivity, specificity, precision, f_meas, accuracy),
    control = control_resamples(save_pred = TRUE)
  )


set.seed(456)
svm_rose <- workflow_rose %>%
  add_model(svm_spec) %>%
  fit_resamples(
    resamples = malaria_folds,
    metrics = metric_set(roc_auc, sensitivity, specificity, precision, f_meas, accuracy),
    control = control_resamples(save_pred = TRUE)
  )

set.seed(456)
rf_rose <- workflow_rose %>%
  add_model(rf_spec) %>%
  fit_resamples(
    resamples = malaria_folds,
    metrics = metric_set(roc_auc, sensitivity, specificity, precision, f_meas, accuracy),
    control = control_resamples(save_pred = TRUE)
```

```
  )

set.seed(456)
xgboost_rose <- workflow_rose %>%
  add_model(xgboost_spec) %>%
  fit_resamples(
    resamples = malaria_folds,
    metrics = metric_set(roc_auc, sensitivity, specificity, precision, f_meas, accuracy),
    control = control_resamples(save_pred = TRUE)
  )
```

```
rose_results <- rbind(
  collect_metrics(lr_rose) %>%
  mutate(Model = "LR",
         Resampling = "rose"),
  collect_metrics(svm_rose) %>%
  mutate(Model = "SVM",
         Resampling = "rose"),
  collect_metrics(rf_rose) %>%
  mutate(Model = "RF",
         Resampling = "rose"),
  collect_metrics(xgboost_rose) %>%
  mutate(Model = "XGBoost",
         Resampling = "rose")
)

head(rose_results)
```

Stacking downsampling and oversampling

```
downsampled <- recipe(malaria ~. , data = malaria_train) %>%
  themis::step_downsample(malaria, seed = 101) %>% prep() %>%
  bake(new_data = NULL) %>%
  select(malaria, child_age:water_source)

upsampled <- recipe(malaria ~. , data = malaria_train) %>%
  themis::step_upsample(malaria, seed = 101) %>% prep() %>%
  bake(new_data = NULL)%>%
  select(malaria, child_age:water_source)

downsampled %>% count(malaria)
upsampled %>% count(malaria)
```

Combine the data

```
ruroData <- bind_rows(downsampled, upsampled) %>%
  select(malaria, child_age:water_source)

glimpse(ruroData)
```

Extract new training set

```
set.seed(123)
RUROTrainSet <- sample_n(ruroData, nrow(malaria_train))

glimpse(RUROTrainSet)
```

Custom split

```
imbalSplit <- make_splits(RUROTrainSet, assessment = malaria_test)
```

```
set.seed(345)
# create a set of resamples
RUROFolds <- vfold_cv(RUROTrainSet, v = 5, repeats = 1) # change to 10 later
```

```
RURORecipe <- recipe(malaria ~., data = RUROTrainSet) %>%
  step_relevel(malaria, ref_level = "Positive") %>%
  step_string2factor(all_nominal(), -all_outcomes()) %>%
  step_dummy(all_nominal(), - malaria)
```

```
RUROWorkflow <- workflow() %>%
  add_recipe(RURORecipe)
```

RURO - Classical and ensemble

```
set.seed(456)
lrRURO <- RUROWorkflow %>%
  add_model(glm_spec) %>%
  fit_resamples(
    resamples = RUROFolds,
    metrics = metric_set(roc_auc, sensitivity, specificity, precision, f_meas, accuracy),
    control = control_resamples(save_pred = TRUE)
  )

set.seed(456)
svmRURO <- RUROWorkflow %>%
  add_model(svm_spec) %>%
  fit_resamples(
    resamples =RUROFolds,
    metrics = metric_set(roc_auc, sensitivity, specificity, precision, f_meas, accuracy),
    control = control_resamples(save_pred = TRUE)
  )

set.seed(456)
rfRURO <- RUROWorkflow %>%
  add_model(rf_spec) %>%
  fit_resamples(
    resamples = RUROFolds,
    metrics = metric_set(roc_auc, sensitivity, specificity, precision, f_meas, accuracy),
    control = control_resamples(save_pred = TRUE)
  )


set.seed(456)
xgboostRURO <- RUROWorkflow %>%
  add_model(xgboost_spec) %>%
  fit_resamples(
    resamples = RUROFolds,
    metrics =  metric_set(roc_auc, sensitivity, specificity, precision, f_meas, accuracy),
    control = control_resamples(save_pred = TRUE)
  )
```

```
RUROresults <- rbind(
  collect_metrics(lrRURO) %>%
  mutate(Model = "LR",
         Resampling = "RURO"),
  collect_metrics(svmRURO) %>%
  mutate(Model = "SVM",
         Resampling = "RURO"),
  collect_metrics(rfRURO) %>%
  mutate(Model = "RF",
         Resampling = "RURO"),
  collect_metrics(xgboostRURO) %>%
  mutate(Model = "XGBoost",
         Resampling = "RURO")
)

head(RUROresults)
```

```
modellingResults <- bind_rows(none_results, down_results, up_results,smote_results, adasyn_results, ros

write_csv(modellingResults,
          "Results.csv")
```

XGBOOST Model ROC curve

```
xgboost_roccurve <- xgboostRURO %>%
  collect_predictions() %>%
  group_by(id) %>%
  roc_curve(malaria, .pred_Positive) %>%
  ggplot(aes(1 - specificity, sensitivity, color = id)) +
  geom_abline(lty = 2, color = "gray80", size = 1.5) +
  geom_path(show.legend = FALSE, alpha = 0.6, size = 1.2) +
  coord_equal()

# xgboost_roccurve
```

Final Model and evaluation on the test set

Recipe as a function

```
recipe_xgboost_ruro <- function(dataset) {
  recipe(malaria ~., data = RUROTrainSet) %>%
  step_relevel(malaria, ref_level = "Positive") %>%
  step_string2factor(all_nominal(), -all_outcomes()) %>%
  step_dummy(all_nominal(), - malaria) %>%
    prep(data = dataset)
}
```

```
xgboost_fun <- function(split, id, try, tree) {

  analysis_set <- split %>% analysis()
  analysis_prepped <- analysis_set %>% recipe_xgboost_ruro()
  analysis_baked <- analysis_prepped %>% bake(new_data = analysis_set)
  model_xgboost <- boost_tree(
  mode = "classification",
  mtry = try,
  trees = tree) %>%
```

```
    fit(malaria ~ ., data = analysis_baked)
  assessment_set <- split %>% assessment()
  assessment_prepped <- assessment_set %>% recipe_xgboost_ruro()
  assessment_baked <- assessment_prepped %>% bake(new_data = assessment_set)
  tibble(
    "id" = id,
    "truth" = assessment_baked$malaria,
    "prediction" = model_xgboost %>%
      predict(new_data = assessment_baked) %>%
      unlist()
  )

}
```

```
set.seed(4567)
pred_xgboost_ruro <- map2_df(
  .x = RUROFolds$splits,
  .y = RUROFolds$id,
  ~ xgboost_fun(split = .x, id = .y, try = 3, tree = 1000)
)
head(pred_xgboost_ruro )
```

```
Final_Results <- pred_xgboost_ruro %>%
  conf_mat(truth, prediction) %>%
  summary() %>%
  select(-.estimator) %>%
  filter(.metric %in%
    c("accuracy", "precision", "f_meas", "sens", "spec"))

# Final_Results
```

Strathmore UNIVERSITY

24th May 2022

Mr Maangi, Daniel
daniel.maangi@strathmore.edu

Dear Mr Maangi,

RE: Statistical learning for class imbalanced data. A case study of Malaria Indicator Survey data.

This is to inform you that SU-IERC has reviewed and **approved** your above **SU Masters'** research proposal. Your application reference number is **SU-IERC1341/22.** The approval period is **24th May 2022 to 23rd May 2023.**

This approval is subject to compliance with the following requirements:

i. Only approved documents including (informed consents, study instruments, MTA) will be used
ii. All changes including (amendments, deviations, and violations) are submitted for review and approval by SU-IERC.
iii. Death and life-threatening problems and serious adverse events or unexpected adverse events whether related or unrelated to the study must be reported to SU-IERC within 48 hours of notification
iv. Any changes, anticipated or otherwise that may increase the risks or affected safety or welfare of study participants and others or affect the integrity of the research must be reported to SU-IERC within 48 hours
v. Clearance for export of biological specimens must be obtained from relevant institutions.
vi. Submission of a request for renewal of approval at least 60 days prior to expiry of the approval period. Attach a comprehensive progress report to support the renewal.
vii. Submission of an executive summary report within 90 days upon completion of the study to SU-IERC.

Prior to commencing your study, you will be expected to obtain a research license from National Commission for Science, Technology, and Innovation (NACOSTI) https://research-portal.nacosti.go.ke/ and obtain other clearances needed.

Yours sincerely,

for: **Dr Ben Ngoye,**
Secretary; SU-IERC

Cc: Prof Fred Were,
Chairperson; SU-IERC

Curiginal

Document Information

Analyzed document	Imbalanced_Learning_Manuscript.pdf (D138945710)
Submitted	2022-06-02T07:25:00.0000000
Submitted by	
Submitter email	Daniel.Maangi@strathmore.edu
Similarity	5%
Analysis address	library.strath@analysis.urkund.com

Sources included in the report

SA	**report.docx** Document report.docx (D26218015)	2
SA	**K.Ulaga Priya - Thesis for Plagarisim.doc** Document K.Ulaga Priya - Thesis for Plagarisim.doc (D133478724)	4
W	URL: https://azpdf.tips/learning-from-imbalanced-data-sets-pdf-free.html Fetched: 2021-03-08T10:14:20.3770000	10
SA	**paper 2.docx** Document paper 2.docx (D37901395)	1
SA	**final paper review.docx** Document final paper review.docx (D91005008)	1
W	URL: https://www.diva-portal.org/smash/get/diva2:1519153/FULLTEXT01.pdf Fetched: 2021-11-12T22:37:02.8870000	2
SA	**Shidha. M V.pdf** Document Shidha. M V.pdf (D95772883)	1
SA	**New Microsoft Word Document (3).docx** Document New Microsoft Word Document (3).docx (D28496563)	1
W	URL: https://lirias.kuleuven.be/retrieve/584423 Fetched: 2021-12-13T14:14:01.2370000	4
W	URL: https://lirias.kuleuven.be/retrieve/657601 Fetched: 2022-06-02T07:25:30.2730000	1

Curiginal

Entire Document

Statistical learning for class imbalanced data: a case study of malaria indicator survey data Maangi Daniel Ongera

70% MATCHING BLOCK 1/27 SA report.docx (D26218015)

Submitted in total fulfilment of the requirements for the degree of Master of Science in

Statistical Sciences of Strathmore University Institute of Mathematical Sciences Strathmore University Nairobi, Kenya May 2022 This thesis is available for Library use through open access on the understanding that it is copyright material and that no quotation from the thesis may be published without proper acknowledgement.

Declaration I declare that this work has not been previously submitted and approved for award of a degree by this or any other University. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made in the thesis itself. © No part of this thesis may be reproduced without the permission of the author and Strathmore University. Name: ..... Maangi Daniel Ongera ..... Signature: ..... Date: ..... May 22, 2022 ..... Approval The thesis of Maangi Daniel Ongera was reviewed and approved by the following: Professor Bernard Omolo Supervisor, Institute of Mathematical Sciences, Strathmore University. Signature: ..... Dr. Godfrey Madigu Dean, Institute of Mathematical Sciences, Strathmore University. Dr. Bernard Shibwabo Director, Office of Graduate Studies, Strathmore University. ii

Abstract Background Class imbalanced problems are predominant in real-life applications. In most cases, the minority class is the most important. Standard statistical learning algorithms tend to produce poor results for the minority class and very good results for the majority class. One of the widely used mechanism to address this problem is by re-sampling the training data. The objective of this study is to examine the performance of statistical learning algorithms by using different re-sampling approaches for handling class imbalance. Methods Two classical and ensemble statistical learning techniques were trained on an imbalanced Malaria Indicator Survey data set while handling the majority-minority problem through re-sampling. These included: Logistic regression, support vector machines, random forest, and extreme gradient boosting. The algorithms were trained without handling class imbal- ance first. Secondly, the algorithms were trained using six re-sampling procedures to handle class imbalance:

100% MATCHING BLOCK 2/27 SA K.Ulaga Priya - Thesis for Plagarisim.doc (D133478724)

random under-sampling, random over-sampling, Synthetic Minority Over- sampling technique (SMOTE),

Random Over Sampling Examples (ROSE) techniques and Adaptive Synthetic Sampling Approach (ADASYN). We further investigated whether com- bining randomly under-sampled and over-sampled data can result in improved performance. Eighty percent of the data was used for model training using 5 fold cross validation. Results All methods that were considered for handling class imbalance had strengths and weaknesses depending on the performance metric. For instance, random under-sampling resulted in models with higher sensitivity than random over-sampling. To get a trade-off between sensitivity and specificity, these two methods can be combined. This approach resulted in 99.5% sensitivity, 88.1 % specificity, 89.6 % precision, 94.3 % F1 score and a 93.9 % accuracy on the test set. iii

Curiginal

List of abbreviations WHO World Health Organisation KMS Kenya Malaria Strategy DHS Demographic Health Surveys MIS Malaria Indicator Survey ML Machine Learning LR Logistic Regression SVM Support Vector Machines RF Random Forest XGBoost Extreme Gradient Boosting

76% MATCHING BLOCK 3/27 SA K.Ulaga Priya - Thesis for Plagarisim.doc (D133478724)

SMOTE Synthetic minority over-sampling technique BSMOTE Borderline synthetic minority over-sampling technique

ROSE Random oversampling Examples ADASYN Adaptive snythetic sampling approach RURO Random Undersampling and Random Oversampling ix

Acknowledgement First and foremost, I am grateful to the Almighty God for good health, wellbeing and the resources that were necessary in completing this research study. My very special gratitude to my supervisor, Prof. Bernard Omolo, for his responsiveness, suggestions, feedback and motivation that was a key driver in this research. Without his assistance, this research work would not have been successful. Writing this paper was difficult than I would ever imagined. Therefore, I am equally grateful to the defense committee for their generous feedback and motivation. Many thanks Dr. Collins Odhiambo and Dr. Linda Chaba. A very special thanks to my in loving memory parents, Evans Maangi and Hellen Maangi. I also express my profound gratitude to my siblings, for always believing in me, and their unwavering support and encouragement throughout my studies. I would also like to record my sincere gratitude to one and all, who in one way or another, lent their hand in this work. x

Chapter 1 Introduction 1.1 Background Statistical learning is a vast set of tools for understanding data James et al. (2013). It is a new field that has emerged in the field of statistics. Broadly, statistical learning can be classified into supervised or unsupervised learning. Supervised learning refers to the development of statistical models to predict or estimate an output using one or more inputs James et al. (2013). Supervised learning is based on ground truth. That is, we already have prior information on what the output from the analysis should be. Unsupervised learning on the other hand does not have labeled or supervising outputs. In supervised learning, there's an associated response for each observation of the predictor measurements. The goal is to fit a model that can be used for prediction or inference by relating the predictor measurements to the response. For prediction, the aim is to develop a model that accurately predicts the response on unseen data. For inference, the aim is to understand the relationship between the predictors and the response. Response variables can either be quantitative or qualitative. Quantitative variables take on numerical values, for example age, salary, height, weight et cetera. Qualitative variables are also known as categorical variables. They take on one class out of the

different k classes. For instance, whether an individual is negative or positive for a particular disease. Based on the type or response variables, supervised learning problems are broadly categorized into regression and classification. Regression problems involve a quantitative response whereas classification problems involve a qualitative response. In general, supervised learning methods are selected on the basis of the response variable. For example, linear regression when the response is quantitative and logistic regression when the response is qualitative. 1

The type of the predictor variables is not important in supervised learning James et al. (2013). Classification methods can further be classified into linear and nonlinear. In a linear problem, a simple linear classifier can be used to approximate the class boundaries. In a nonlinear problem, the class boundaries cannot be be approximated well with linear hyperplanes. Classification is a key task of pattern recognition. One problem hindering performance of classification methods is class imbalance. This is because of their accuracy tailored design Fernández et al. (2018). A class imbalanced problem is one in which the distribution across classes is skewed or biased. That is to say, the class imbalance problem arises when the samples representing one class are much lower than the rest of the classes Fernández et al. (2018). In a binary classification setting, a class imbalanced problem arises when one of the two classes has more samples (the majority class) than the other class (the minority class) Longadge and Dongre (2013). This can vary from a slight difference between the classes to a severe imbalance in which there's only one instance of the minority class in hundreds, thousands or millions of observations. Usually, the most important class is the one with fewer samples Guo et al. (2008). This is a common problem in most real world datasets. For instance, in fraud detection, majority of the transactions are authentic. In spam detection, majority of emails or text messages are not spam. In disease screening, majority of people may not have the disease. One of the common approaches to mitigate the issues due to class imbalances is by re-sampling the training data. This project is an empirical evaluation of the predictive performance of various statistical learning algorithms using re-sampling approaches as a way of mitigating the class imbalance problem. Kenya's 2020 Malaria Indicator Survey data set is used as a case study Ministry of Health Division of National Malaria Programme Kenya et al. (2021). 1.2 Problem statement Malaria remains to be one of the major public health threats of our time and is among the leading causes of mortality in low-income countries alongside Tuberculosis and HIV/AIDS WHO (2020). In 2020, there were 241 million cases of malaria and approximately 627,000 2

people died from Malaria - an increase of about 69,000 from 2019. African countries continue to carry a disproportionately high share of the global Malaria burden. In 2020, 95% of these world Malaria cases were from Africa WHO (2021). In Kenya, Malaria is still a major public health and socioeconomic problem, with the highest prevalence being among children less than 14 years. According Ministry of Health Division of National Malaria Programme Kenya et al. (2021), 6% of children aged between 6 months and 14 years were positive for Malaria by microscopy. In this data set, the majority class (94 %) are negative whereas the minority class (6 %) are positive for malaria infection. In this majority-minority setting where the positive examples are rare, it can be very difficult to estimate the probability of a child having malaria infection given a set of malaria risk factors (prediction). Furthermore, this problem can pose a challenge in trying to understand the relationship between the predictors and the response (inference). Majority of statistical learning algorithms applied on this unbalanced data will tend to result in biased predictions against the positive class. Many methods have been identified to handle class imbalance during modelling. One of the ways is through re-sampling the training data Fernández et al. (2018). Broadly, this method entails removing samples from the majority class (negative for malaria) or increasing the samples in the minority class (positive for malaria). This study applies no sampling, under-sampling, over-sampling, and hybrid methods when training the statistical learning algorithms. In addition, we investigate whether combining both randomly under-sampled and randomly over-sampled data can result in improved performance. We consider two classical statistical learning techniques and two algorithms from the ensemble framework. 1.3 Objectives 1.3.1 Main objective The objective of this study is to examine the predictive performance of statistical learning algorithms by using different data level approaches to handle class imbalance. 3

1.3.2 Specific objectives • To compare the performance of data level approaches in handling class imbalance. • To investigate whether combining randomly undersampled and randomly oversampled data can result in improved perfomance. 4

Chapter 2 Literature review 2.1 Introduction Technically, a data set can be considered to be imbalanced if it has unequal distribution between the majority and minority classes Leevy et al. (2018). This is common in real-world data sets, and it can vary from slight to severe imbalance. The bulk of the data set is composed of the majority class, whereas the minority class, which is often the interest, has limited representation. This section provides a review of the approaches in learning from class imbalanced data. 2.2 Handling class imbalance Class imbalance in a data set can skew the predictive performance of classifiers in a dramatic way Leevy et al. (2018). There are many approaches that have have been developed in order to correctly distinguish the minority class from the majority class. According to Fernández et al. (2018),

these approaches can be categorized into four groups. The grouping is based on how they handle the class imbalance problem. That is, data level, algorithm level, cost-sensitive, and ensemble based approaches. 2.2.1 Algorithm level approaches Algorithm level approaches are those that bias learning towards the minority class by adapting existing classification algorithms. The aim of the approach is to modify the actual classifier learning procedure to handle class imbalance issues. However, a deep understanding of the 5

learning algorithm is needed to identify the mechanisms driving the bias to the majority class. These algorithms are however difficult to design and implement than other methods. Despite some outstanding performers, the results in literature are conflicting and inconsistent Leevy et al. (2018). 2.2.2 Cost-sensitive learning approaches Majority of statistical learning algorithms assume equality of misclassification errors. How- ever, this is not usually the case. A false negative can be considered more costly or worse than a false positive. For instance, in medical cancer diagnosis, falsely classifying a patient as negative, when the patient is actually is positive, is considered much more costly Ling and Sheng (2008). Cost-sensitive learning takes into account the costs of prediction errors during model training. Cost-sensitive classifiers can be categorized into two groups: direct approaches that involve direct introduction of the misclassification cost into the training procedure, and meta-learning approaches that either modify the training data (pre-processing) or the outputs (post-processing) Fernández et al. (2018). The performance of cost-sensitive classifiers are heavily dependent on the provided cost matrix. Incorrect initial costs can affect the learning process. These costs can originate from a domain expert or during model training Fernández et al. (2018). 2.2.3 Ensemble-based approaches Most of the prediction errors of standard statistical learning algorithms are as a result of bias or variance. The idea behind ensemble learning is to minimize these errors by combining several base learners (often weak learners) Hastie et al. (2009). The final result is usually a more accurate meta model. Ensemble learning methods include: bagging, boosting and stacking. However, these techniques cannot solve the majority-minority problem by themselves Fernández et al. (2018). To handle class imbalance, ensemble learning algorithms are adapted by combining them with either data level, algorithm-level or cost-sensitive approaches. For instance, Huda et al. (2018) developed three base learners using different 6

oversampling techniques. A random forest based ensemble of these learners resulted in improved predictive performance. 2.2.4 Data level approaches These are also known as sub-sampling or re-sampling approaches. Re-sampling training data was the first method for handling class imbalance Fernández et al. (2018). It is still one of the commonly used mechanism. The main advantage of sampling methods is that they are simple to implement and can be used with any classifier. They involve the use of different sampling procedures to provide an adequately balanced data set to the statistical learning algorithms. The goal of data level approaches is to obtain a balanced class distribution by resampling the training data. These methods are independent of the learning algorithm and can be easily implemented. According to Fernández et al. (2018), there are three different ways in which resampling techniques can be used to produce a balanced class distribution: undersampling, oversampling and hybrid methods. Undersampling methods These approaches involve randomly subsetting all the classes in the training data set such that the frequency of each class matches the least common class. A major drawback in this approach is the potential loss of useful information Leevy et al. (2018). This is because most of the examples from the majority class are ignored. There are several strategies proposed for undersampling, the simplest being random downsampling. This method randomly eliminates instances of the majority class to obtain a balanced distribution. Although this method is simple and effective, majority class examples are deleted without knowing how important they might be in estimating the class boundary. Several heuristic solutions that try to identify redundant majority class instances that can be removed have been suggested. These solutions informatively undersample the prevalent class, i.e. the choice of examples to be removed is targetted Sun et al. (2009). Classical undersampling methods include: Tomek Links (TL), Condensed Nearest Neighbour Rule (US-CNN), One-Sided Selection (OSS), US-CNN + TL, 7

Neighbourhoos Cleaning Rule (NCL), Class Purity Maximization (CPM), Undersampling Based on Clustering (SBC) and Near Miss approaches.

88% MATCHING BLOCK 4/27 W https://azpdf.tips/learning-from-imbalanced-da ...

Advanced undersampling techniques include: evolutionary undersampling, undersampling by cleaning data, ensemble based undersampling,

and clustering based undersampling Fernández et al. (2018). Burez and Van den Poel (2009) investigated the impact of random and advanced undersam- pling in the prediction of customer churn through gradient boosting and weighted random forests. AUC and lift were used as the model evaluation metrics. Overall, the results showed that under-sampling

resulted in improved prediction accuracy. They further confirmed the finding by Weiss and Provost (2003) that there is no specific answer on which class distribu- tion will result in better predictive performance. This entirely depends on the method and area of application. The researchers noted that it would be interesting to apply the resampling approaches they adopted to other learning novel algorithms such support vector machines, neural networks et cetera. Oversampling methods These approaches involve randomly sampling with replacement the minority class to obtain the same size as the majority class. The approach however my cause over-fitting because it produces exact copies of instances of the minority class Leevy et al. (2018). In addi- tion, it may require extra computational burden. There are several strategies proposed for oversampling, the simplest being random over-sampling. This method randomly replicates instances of the majority class to obtain a balanced distribution. There are also solutions that informatively over-sample the minority class, i.e. the choice of examples to replicate is targetted Sun et al. (2009). According to Fernández et al. (2018), advanced oversampling methods include Synthetic Minority Oversampling Technique (SMOTE), and SMOTE exten- sions i.e. Borderline SMOTE, Adaptive Synthetic Sampling approach (ADASYN), Random Oversampling Examples (ROSE), Safe-level SMOTE, Majority Weighted Minority Oversam- pling Technique (MWMOTE), and Mahalanobis Distance Based Oversampling Technique (MDO). Batista et al. (2004), noted that over-sampling methods generally outperformed under-sampling methods using the area under the ROC curve metric. Besides, the authors 8

demonstrated that random over-sampling, the simplest approach, was very competitive to advanced over-sampling approaches. Hybrid methods These methods combine both under-sampling and over-sampling strategies. They arise due to the drawbacks for each of the resampling approaches. They aim to obtain an optimal balance of eliminating majority class instances and generating new instances of the minority class to obtain the best possible performance Fernández et al. (2018). Most of these approaches have involved enhancing the performance of classical SMOTE. For example, SMOTE + Tomek Link and SMOTE + ENN. More complex mechanisms include: Agglomerative Hierarchical Clustering (AHC), SPIDER, SMOTE-RSB, and SMOTE-IPF. Despite all these methods, there's no clarity on which of the strategies is the best. 2.3 Evaluation metrics Standard statistical learning algorithms

68% MATCHING BLOCK 5/27 W https://azpdf.tips/learning-from-imbalanced-da ...

tend to have high accuracy for the majority class but poor results for the minority class.

In other words, the samples from the minority class are misclassified more often than those from the majority class Fernández et al. (2018). This is because these methods tend to assume that the training data set is balanced. Accuracy is therefore less robust and unable to catch the nuances as a result of the different types of errors. For this reason, more informative metrics obtained from a confusion matrix are commonly used. 9

2.3.1 Confusion matrix In two class problems, a confusion matrix is a summary of all possible outcomes in four categories: true positives, true negatives, false postives and false negatives. Table 2.1: Two class confusion matrix Predicted Positive Negative Total Actual Positive TP FN (TP+FN) Negative FP TN (FP+TN) Total (TP+FP) (FN +TN) N These categories

32% MATCHING BLOCK 6/27 SA paper 2.docx (D37901395)

are defined as: • True Positives (TP) : the number of positive examples correctly classified. • True Negatives (TN) : the number of negative examples correctly classified. • False Positives (FP) : the number of positive examples incorrectly classified. • False Negatives (FN) : the number of

negative examples incorrectly classified. Many standard performance evaluation metrics to measure the quality of modelling can be derived using the the above 2x2 confusion matrix. 2.3.2 Accuracy Accuracy is one of the common methods for evaluating the performance of a classifier. It refers to the proportion of predictions that a classifier got right. That is, how well the predictions align with reality. Accuracy is formally defined as: Accuracy =

82% MATCHING BLOCK 7/27 SA final paper review.docx (D91005008)

Number of correct predictions Total number of predictions = TP + TN TP + TN + FP + FN

An accuracy of 99 % doesn't necessarily mean that the model is good if it does not illustrate the true positive and true negative attributes. Furthermore, in imbalanced data sets, accuracy is biased towards the majority class. 10

2.3.3 Other metrics True Negative Rate It refers to the proportion of negatives that were correctly classified as negative. The True Negative Rate (TNR) is formally defined as: TNR = True Negatives Total actual negatives = True Negatives True Negatives+False Positives = TN TN + FP . True Positive Rate It refers to the proportion of positives that were correctly classified as positive. The True Positive Rate (TPR) is formally defined as: TPR = True positives Total actual positives = True positives True Negatives+False Negatives = TP TP + FN . Recall Recall is the same as true positive rate. Recall = TPR = True positives True Negatives+False Negatives = TP TP + FN . Sensitivity Sensitivity is the same as the true positive rate or recall. In disease diagnosis, it refers to the people who tested positive and are actually positive. It can be viewed as the probability that the classifier shows a positive result given that the individual is indeed positive for the disease. It is a good measure, when the cost of a false negative is high. Sensitivity (Recall) = TPR = True positives True Negatives+False Negatives = TP TP + FN . False Positive Rate The False Positive Rate (FPR) is also referred to as the false alarm rate. It refers to the proportion of negatives that are incorrectly classified out of the total negatives Fawcett (2005). 11

FPR ≈ Negatives incorrectly classified Total negatives . Specificity Also known as the true negative rate. It refers to the proportion of negatives that were correctly classified. In disease diagnosis, it refers to the people who tested negative and are actually negative. It can be viewed as the probability that the classifier shows a negative result given that the individual is indeed negative for the disease. Specificity = TNR = True Negatives True Negatives+False Positives = TN TN + FP = 1–FPR. Positive Predicted Value It refers to the proportion of predicted positives that is truly positive. Positive Predicted Value (PPV) is formally defined as: PPV = True positive Total predicted positive = True Positive True Positive + False Positive = TP TP + FP . Precision Precision is the same as the Positive Predicted Value. It is a good measure, when the cost of a false positive is high. Precision = PPV = True Positive True Positive + False Positive = TP TP + FP . Negative predicted value It refers to the proportion of predicted negatives that is truly negative. It is a good measure, when the cost of a false negative is high. Negative Predicted Value (NPV) is formally defined as: 12

NPV = True negative Total predicted negative = True Negatives True Negatives + False Negatives = TN TN + FN . F1 Score The F1 score (F measure) has been reported as a good evaluation metric than accuracy in binary classification problems Sun et al. (2009). This is the harmonic mean of precision and recall. That is: F1 Score = (1+β 2)∗ precision + recall (β 2 ∗precision)+recall . Area under an ROC curve (AUC) An Receiver Operating Characteristic (ROC) curve is a two-dimensional representation of the performance of a classification method. The y-axis represents the true postive rate (Sensitivity) while the x-axis has the false positive rate (1 - specificity). In order to compare different classifiers, a single scaler value is needed Sun et al. (2009). The area under the ROC curve (AUC) is the common metric used. Youden's J index Youden's J index is defined as: J = Sensitivity+Specificity–1. 2.4 Malaria risk factors Malaria is a disease caused by a parasite transmitted by an infected female anopheles mosquito. The risk of malaria infection, especially among children, can be increased by certain demographic, socioeconomic and environmental factors. Understanding these factors is important for targetted control interventions. 13

Winskill et al. (2011) noted that there was a significant association between both age and gender and malaria risk. Older male children had increased likelihood for malaria infection. This possibly indicated that older male children were associated with a higher exposure behaviour. Further, the authors demonstrated that use of insecticide-treated mosquito nets was associated with decreased odds for malaria infection among children. This adds to existing extensive literature on the efficacy of insecticide treated nets in malaria prevention. Habyarimana and Ramroop (2020), identified age of the child, location, poverty, number of household members, availability of household sleeping mosquito, residual spraying in the past twelve months, location of the household's source of drinking water, main wall materials of the dwelling, and the age of the head of the household to be significant risk factors for malaria infection among children. AIC, SC and –2LogL were used to compare the reduced logistic regression model (model of intercept only) and the full logistic regression model (model of intercept and covariates). These metrics were smaller for the full model (model with intercept and covariates) compared to the reduced model (model with intercept only), suggesting that the full model was a better fit. These models can be classified as inferential. In similar papers, authors typically focus on drawing qualitative statements on the relative influence the risk factors have on the response. For example, based on the p-value alone, one can state that there is a statistically significant relationship between malaria infection and the risk factors. However, there's is a connection between inferential and predictive models, which if ignored, can be dangerous. This is because such a model based only on statistical significance can perform poorly on other predictive capacity metrics such as sensitivity. Even when the model is not used for prediction, it can be difficult to trust inferences from a model with significant p-values but dismal sensitivity. In general, the predictive performance of a statistical learning algorithm involves determining how close fitted values are to the sample data. 14

Chapter 3 Methodology 3.1 Study data Kenya's Malaria Indicator Survey data was used as the study data Ministry of Health Division of National Malaria Programme Kenya et al. (2021). The data was acquired in StataCorp (2019) format from the

Demographic Health Surveys upon approval of the study. The outcome of interest was Malaria infection, whether negative or positive from a blood smear test. Based on literature, 9 variables were selected as malaria predictors among children aged 6 moths to 14 years. The table below provides a description of the study variables. Table 3.1: Description of variables used in the study Code Description malaria Final result of malaria from blood smear test. The outcome variable. child_age Child's age in months. hh_head_age Age of head of household in years. hh_members Number of household members. endemicity Malaria endemicity zone. has_net Has mosquito bed net for sleeping. residence Residence (urban or rural). wealth_index Wealth index combined. television Whether household has television. water_source Source of drinking water. We summarized the data using the mean for continuous variables and frequency for categori- cal variables. Statistical tests of independence were done at at p–value > 0.05. 15

3.2 Statistical learning methods In this section, a description of each of the statistical learning methods adopted for this study is provided. 3.2.1 Logistic regression Logistic regression is one of the most widely used statistical learning algorithms in classifica- tion. The model is used in prediction when the response variable is categorical (k classes). In this classification problem, there are two classes (C) : 1 - denoting malaria positive and 0 - malaria negative . Mathematically, the probability of a child having malaria given the predictors or risk factors (X) can be expressed as: p(X) = P(C = malaria positive | risk factors) = P(C = 1|X). (3.1) Suppose that we use multiple linear regression to model the probability of a child having malaria as : p(X) = βX = β 0 + β 1 X 1 + β 2 X 2 + ···+β n X p . (3.2) We notice that we will will potentially have probabilities below zero (negative) and above one. This does not make sense. Logistic regression model arises from the need to ensure that the predicted probabilities of the two classes fall and remain in [0,1] Hastie et al. (2009). There are many functions that can be used. In logistic regression, the sigmoid (logistic) function is used. Hence, equation (3.2) can be written in terms of the logit (log odds) as: log p(X) 1– p(X) = βX = β 0 + β 1 X 1 + β 2 X 2 + ···+β n X p . (3.3) Equation (3.3) can be expressed as follows: p(X) = e βX 1+e βX = e β 0 +β 1 X 1 +β 2 X 2 +···+β n X p 1 +e β 0 +β 1 X 1 +β 2 X 2 +···+β n X p . (3.4) 16

Where X ∈ R p is vector of explanatory variables (malaria risk factors) and β ∈ R p are unknown constants estimated by the method of maximum likelihood. This function takes an S-Shaped curve, and thus the predictions can never be below 0 or above 1 James et al. (2013). 3.2.2 Support Vector Machine Since development in the 1990s, support vector machines (SVMs) have demonstrated good performance in different domains, and it is one of the most popular algorithm James et al. (2013). It can solve both classification and regression problems. If the positive and the negative classes can be separated perfectly using a hyperplane, then there exist an infinite number of these hyperplanes. The maximal margin hyperplane, or optimal seperating hyperplane is the natural choice among these. It is the the hyperplane with the largest margin between the classes. This algorithm of classifying observations based on a maximal margin hyperplane is called the maximal margin classifier. However, if there exists no separating hyperplane, then we cannot obtain a maximal margin classifier that can separate the positive class from the negative class. Furthermore, even if a separating hyperplane exists, it might be less desirable to use it in some cases. The support vector classifier is a generalization of the maximal margin classifier using a soft margin. It performs well in classifying most of the observations at the expense of misclassifying a few observations. The margin is called soft because of this. When the boundary between the positive and negative class is linear, the support vector classifier is the natural choice. In practice however, non-linear class boundaries are common.

95% MATCHING BLOCK 9/27 W https://www.diva-portal.org/smash/get/diva2:15 ...

The support vector machine is an extension of the support vector classifier

to solve the non-linear classification problems through a kernel trick which involves enlarging the feature space James et al. (2013). The linear support vector classifier is given by: f(x) = β 0 + n ∑ i∈S α i ⟨x,x i ⟩. (3.5) 17

Whereby, ⟨x,x i ⟩ are inner products and S is a collection of the support points' indices. Support vectors define the position and margin of the separating hyperplane. They are called support vectors because if even one of them is moved, the position, margin or both will change. There will be no effect on the hyperplane if other points are moved instead. In Support Vector Machines, the inner product is replaced with a generalized form known as the the kernel, K(x,x i). A kernel is function that estimates the similarity between any two observations. If a linear kearnel is used, then the support vector machine will be simply the support vector classifier. There are three popular forms of SVM kernels which include: • Polynomial kernel: K(x i ,x j) = (1+α i ⟨x i ,x j ⟩) p • Radial basis kernel: K(x i ,x j) = exp(–γ ‖ x i – x j ‖ 2) • Neural network: K(x i ,x j) = tanh(k 1 ⟨x i ,x j ⟩+k 2) In this paper, we have used radial basis kernel which is the most widely used. 3.2.3 Bagging - Random forest The random forest is one of the most popular algorithm from the ensemble learning frame- work. It is an improvement over bagging James et al. (2013). The idea behind bagging or bootstrap aggregating is to reduce the

Curiginal

variance of a statistical learning algorithm through a combination of many noisy homogeneous base learners. As opposed to bagging, the random forest involves averaging a large collection of de-correlated trees Hastie et al. (2009). Instead of using the entire feature space to grow a tree, the random forest creates a collection of uncorrelated trees (homogeneous base learners) by using a random subset of predictors whenever a split is considered. Approximately, the number of

58% MATCHING BLOCK 14/27 W https://www.diva-portal.org/smash/get/diva2:15 ...

the new set of predictors (m) to be considered at each split is equal to the square root of the total number of

the entire feature space (p) James et al. (2013). That is: m ≈ √ p The out-of-bag (OOB) samples are often used to estimate performance of a random forest. These are samples that have left out in each bootstrap sample. That is, we predict the response 18

for each observation in the out of bag sample using trees where that observation was an OOB James et al. (2013). To obtain a single prediction, the predicted responses are averaged (regression) or through majority voting (classification). However, bagging or random forest leads to increased complexity in model interpretation as compared to a single tree. That is, it is not clear which variables are the most important in the bagged or random forest model James et al. (2013). We can however estimate the importance of each predictor in the final model by using RSS (bagged regression) or the Gini index (bagged classification). The predictor with the largest decrease is the most important predictor. 3.2.4 Boosting - XGBoost Boosting is another ensemble technique for improving the performance of a statistical learning algorithm. In contrast to bagging, boosting does not involve drawing random samples. It involves growing a tree sequentially using the original data by fitting each new tree to the residuals from the previous trees and updating the residuals James et al. (2013). Boosting has shown to provide better predictive performance than bagging. The Gradient Boosting Machine (GBM), originally coined by Friedman (2001), is based on the boosting concept and can handle both regression and classification problems. Whereas the random forests involves averaging deep individual trees, the GBM sequentially builds a committee of weak and shallow trees which are an improvement from the previous. It is a redefinition of boosting whereby weak learners are added using gradient descent. Gradient descent algorithm is an iterative first-order optimization for finding the local minimum of a function that is differentiable. Usually, to reach this local minimum involves many steps James et al. (2013). The number of the required steps is determined by the learning rate, γ. However, the gradient boosting machine can be prone to over-fitting. Extreme Gradient Boosting (XGBoost) Chen et al. (2015) is a scalable and efficient implementation of the gradient boosting machine. To control for over-fitting and achieve attain better predictive performance, XGBoost uses a regularized model formulation: Obj (t) = n ∑ i=1 L(y i , ˆ y (t) i)+ t ∑ t=1 Ω(f i). (3.6) 19

Where, the first term represents loss function and second term is the regularization parameter. XGBoost is a widely used algorithm that has shown to achieve high predictive performance on many supervised learning challenges. For instance, more than half of the winning solutions in 2015 Kaggle machine learning challenges used XGBoost Chen and Guestrin (2016). 3.3 Class imbalance handling In this section, an overview of the methods for handling class imbalance is provided. There are many methods for handling class imbalance. However, in this paper we have used common data-level approaches. 3.3.1 Random undersampling Random undersampling (Figure 3.1) is one of the common and simplest approaches for handling class imbalance. It is a non-heuristic approach where examples of the majority class are randomly eliminated from the data set. In this study, the majority class is the number of children who have negatively tested for malaria. Therefore in this method, we randomly deleted malaria negative cases to obtain a balanced data set. The main advantage of this approach is that; deletion of some instances of the majority class leads to a significant reduction in the size of the training data and therefore reducing the time required in fitting a statistical learning model. A major problem associated with this technique is the potential loss of useful information. 3.3.2 Random oversampling Random oversampling (Figure 3.2) is also another simple approach for handling class imbalance. It is a non-heuristic approach where examples of the minority class are randomly replicated from the data set. In this study, the minority class is the number of children who have positively tested for malaria. Therefore in this method, we randomly replicated malaria positive cases to obtain a balanced class distribution. The main disadvantage of this approach is that; replication of some instances of the minority class leads to a significant increase in the 20

Figure 3.1: Random under-sampling size of the training data, and therefore this can increase time required in fitting a statistical learning model. Furthermore, this approach may result in an over-fitted model. Figure 3.2: Random over-sampling 21

Curiginal

3.3.3 Synthetic Minority Over-sampling Technique Suggested by Chawla et al. (2002), Synthetic Minority Oversampling Technique (SMOTE), rebalances the training data through oversampling. In contrast to random oversampling, the SMOTE algorithm involves generating synthetic examples for the minority class (malaria positive in this case). Positive class examples that are close are selected, and then a line drawn between them. A new sample is then drawn along this line.

69% MATCHING BLOCK 8/27 W https://azpdf.tips/learning-from-imbalanced-da ...

For this reason, this approach is said to operate on the "feature space" rather than the "data space"

and thus it is a less of an application specific method Chawla et al. (2002). Formally, a decision on the needed amount of over-sampling is made, usually that leads to a 50:50 class distribution. After which,

39% MATCHING BLOCK 10/27 W https://azpdf.tips/learning-from-imbalanced-da ...

a positive class example is selected randomly from the training set and its k-nearest neighbours (say 5) are determined. Lastly, n of these k examples are randomly selected to generate new instances via interpolation.

This is done by taking the difference between the considered feature vector or sample and its nearest neighbour Chawla et al. (2002). This difference is then multiplied by any number randomly drawn from (0,1) and then adding to the considered feature vector. This process leads to a random point being selected between features that lie together Chawla et al. (2002). 3.3.4 Adaptive snythetic sampling Approach Adaptive snythetic sampling Approach (ADASYN) He et al. (2008) is an adaptive technique for learning from imbalanced data motivated by the success of approaches that involve generation of synthetic data such as Synthetic Minority Over-sampling Technique (SMOTE). The objective of ADASYN is to reduce bias and learn adaptively. More synthetic examples that are more difficult to learn from the minority class are generated. This forces the learning algorithm to focus on these examples. This is the major difference between SMOTE and ADASYN. The procedure for ADASYN algorithm He et al. (2008) is as follows: 1. Consider the training data set D train with n examples {x i ,y i },i = 1,...,n. x i ∈ X, is an instance of the

28% MATCHING BLOCK 11/27 W https://azpdf.tips/learning-from-imbalanced-da ...

feature space and y i ∈ Y = {P,N} is the response label, positive or negative, associated with x i . 22 2. Define n p and n n be the number of minority (positive) class instances and the number of majority (negative) class instances, respectively

in the training data, D train . Then, n p ≤ n n and n p + n n = n. 3. Calculate class imbalance degree, d = n p n n where d ∈ (0,1]. 4. Let d th be the

59% MATCHING BLOCK 12/27 W https://azpdf.tips/learning-from-imbalanced-da ...

preset threshold ratio for the maximum tolerated degree of class imbal- ance. If d > d th : (a) Determine the number of synthetic

minority (positive) class examples that need to be generated as G = (n n – n p)×β, where β ∈ [0,1] is the required balance level after synthetic data is generated. If

70% MATCHING BLOCK 13/27 W https://azpdf.tips/learning-from-imbalanced-da ...

β = 1, then a fully balanced data is created after the generalized process. (b) For each x i ∈ minority

class (P), use the Euclidean distance measure to find the K nearest neighbours of x i . Then, calculate the ratio r i = Δ i /K, where i = 1,...,n p and Δ i

Curiginal

82% MATCHING BLOCK 17/27 SA Shidha. M V.pdf (D95772883)

is the number of examples in the K nearest neighbours of x i which belong to the majority (

negative) class. Hence, r i ∈ [0,1]. (c) Normalize r i by using ˆr i = r i / ∑ n p i=1 r i , such that ˆr i is a density distribution (∑ i ˆr i = 1). (d) For each minority (positive) instance x i , determine the number of synthetic instances that will be generated, g i = ˆr i × G, where G is from 4 (a). (e) For each positive (minority) class, generate the g i examples: From 1 to g i , i. From the K nearest neighbours of x i , choose randomly one positive (minority) example, ii. Generate the synthetic example, s i = x i +(x zi – x i), where λ ∈ [0,1] is a random number and (x zi –x i) is a difference vector. 3.3.5 Random Over-sampling Examples Random over-sampling examples (ROSE) Menardi and Torelli (2014) is based on a smoothed bootstrap method to generate new artificial examples from the classes. Let D train be the 23

training data set with N samples {x i ,y i } where i = 1,...,N, x i ∈ x is an example in the dimension m feature space, and y i ∈ C = {Y 0 ,Y 1 } = {Positive,Negative} the associated class label. Let the probability density function of x ∈ R d be f(x) and N j the number of examples that belong to class Y j . Then, the ROSE procedure is as follows Leevy et al. (2018): 1. Select y ∗ = Y j with probability π j . 2. Select {x i ,y j } ∈ D train in such a way that y i = y ∗ , with probability 1/N j . 3. Sample x ∗ from K H j (.,x i) , where K H j is a probability distribution centered at x i and covariance matrix H j . Basically, ROSE involves drawing an example from

50% MATCHING BLOCK 15/27 W https://azpdf.tips/learning-from-imbalanced-da ...

one of the two classes and generating a new instance in the neighborhood {x ∗ ,y ′∗ }. The shape of the countour sets of K and

the covariance matrix H j determine the shape and width of this neighbourhood, respectively. Repeating steps 1 to 3 leads to as many examples as one may want. Adjusting the probability π j is used to balance the new data as desired. 3.3.6 Random under-sampling and Random over-sampling (RURO) As noted before, both random under-sampling and random over-sampling have drawbacks. The main problem with random under-sampling is the loss of potentially useful data. On the other hand, random over-sampling may lead to an over-fitted model and increased time in the model fitting process. To address these issues, we propose a method that combines these two approaches. Below is the description of the RURO algorithm. 1. Consider an imbalanced training data set D train with n examples {x i ,y i },i = 1,...,n where x i ∈ X, is an example of the feature space and y i ∈ Y = {P,N} is the associated binary response. 2. Let n p

66% MATCHING BLOCK 16/27 W https://azpdf.tips/learning-from-imbalanced-da ...

and n n be the number of minority class instances and the number of majority class instances, respectively

in D train such that n p > n n and n p + n n = n. 3. Obtain D u train by randomly under-sampling the majority class N in D train . 4. Obtain D o train by randomly over-sampling the majority class P in D train . 24

5. Obtain D a train by combining D u train and D o train . 6. Obtain D ou train by randomly sampling examples of size n from D a train . Note that n is the size of the original training data. 7. D uo train is the new balanced training data set. 3.4 Implementation 3.4.1 Software The modelling is performed using R version 4.1.3 and R studio as the integrated development environment. R "meta packages" tidyverse Wickham et al. (2019) for data import, manipula- tion and visualization; and tidymodels Kuhn and Wickham (2020) for model development using tidy principles. The stargazer package Hlavac (2015) was used to translate R statistical tables into a Latex code. 3.4.2 Training and testing When modelling, it is important to make a decision on how data will be spent at the initial stage. This is because it is related to the empirical validation of the models developed thereafter. The commonly used approach to empirically evaluate model performance is by splitting the existing data into two sets, that is: the training set and test set. Usually, the training set forms the bulk of the split. In this paper, 80% of the original Ministry of Health Division of National Malaria Programme Kenya et al. (2021) malaria indicator survey data set is used as the training set. Further, 20% of the data is held as a reserve to evaluate the performance of the best performing model(s). This is the final arbitration to determine the model efficacy. 3.4.3 Validation In modern statistics, resampling methods are indispensable James et al. (2013). They are used to obtain additional information about a model of interest by repeatedly drawing samples from 25

the training set. For instance, resampling methods can help us to determine the variability of a model. There are two commonly used approaches for resampling, that is, cross-validation and the bootstrap. In this paper, we have used k-fold cross validation. It involves dividing the training data into k folds (groups) of approximately equal size James et al. (2013). The first fold is is used as the validation set and the rest k – 1 folds are used to fit the model. The procedure is repeated k times and the k– fold cross validation estimate calculated by averaging the classification errors. Overall, to compare the predictive performance of the statistical learning methods using the various approaches for handling class imbalance, 5 fold cross validated metrics such as F1 score, sensitivity, specificity, precision and accuracy are used. Finally, the performance of the best performing model was evaluated on the unseen set. 26

Chapter 4 Results In this section, we present the results obtained when we fitted different statistical learning models on the imbalanced malaria data set using the different methods for handling class imbalance discussed in chapter three. Furthermore, we also present the effect of combining randomly undersampled and randomly oversampled data on the performance of the learning algorithms. The section is divided into two subsections i.e. summary of study data and the statistical learning results. 4.1 Summary of the study data Table 4.1 provides a descriptive summary of the potential predictors for malaria infection among children aged between 6 months and 14 years. Overall, out of the 11,549 observations, 969 (8.3%) were positive for malaria by microscopy. In general, there was a significant association (p-value > 0.001) between the 9 predictors and the outcome variable. As further illustrated by figure 4.1, the children with malaria (red box) are on average older compared with the children without malaria (blue box). Similarly, malaria incidence among children was on average higher in households where the head was older. Further, the malaria positive cases were on average higher in households with many members. Figure 4.2 indicates that majority of the malaria positive cases were observed in highland epidemic, lake endemic and coastal endemic regions compared to the seasonal and low risk areas. The highest malaria cases among children was highest in the rural areas, households without television, or which did not have their water sources piped into their yard or dwelling or those that were poor. 27

Table 4.1: Summary of the study data Predictor Negative, N = 10,580 Positive, N = 969 p-value Age of child (months) 86 (45, 131) 111 (72, 147) >0.001 Age of household head (years) 42 (35, 53) 45 (36, 57) >0.001 Number of household members 6.00 (4.00, 7.00) 6.00 (5.00, 8.00) >0.001 Malaria endemicity zone >0.001 Highland epidemic 2,089 (20%) 33 (3.4%) Lake endemic 3,855 (36%) 842 (87%) Coastal endemic 1,153 (11%) 60 (6.2%) Seasonal 2,177 (21%) 33 (3.4%) Low risk 1,306 (12%) 1 (0.1%) Availability of mosquito bed net 7,285 (69%) 751 (78%) >0.001 Residence >0.001 Urban 3,759 (36%) 148 (15%) Rural 6,821 (64%) 821 (85%) Wealth index >0.001 Poorest 2,924 (28%) 304 (31%) Poorer 2,350 (22%) 322 (33%) Middle 2,168 (20%) 210 (22%) Richer 1,993 (19%) 108 (11%) Richest 1,145 (11%) 25 (2.6%) Household has television 4,439 (42%) 311 (32%) >0.001 Source of drinking water >0.001 Piped into dwelling, yard/ plot 1,827 (17%) 36 (3.7%) Other 8,753 (83%) 933 (96%) 28

Figure 4.1: Bivariate relations between continous predictors and malaria infection Figure 4.2: Bivariate relations between categorical predictors and malaria infection 4.2 Model Performance 4.2.1 Classical learning Table 4.2 provides a summary of the results obtained from fitting classical statistical learning algorithms using 5-fold cross validation. The two classical models considered were logistic 29

regression and the support vector machines. The methods for handling class imbalance included, none, random undersampling, random oversampling, SMOTE, ADASYN, ROSE. We further explored whether combining random undersampling and random oversampling (RURO) had an effect on performance of the classical models. For both logistic regression and the support vector machines, the accuracy was high (more than 91.3 %) when the models were fitted without handling class imbalance. The specificity for both models was also high (100 %) while the sensitivity was very low (0 %). The area under the receiver operating characteristic curve (ROC_AUC) was 82.3% and 78.5% for logistic regression and the support vector machines, respectively. It was not possible to determine the F1 score and precision values. Table 4.2: Classical statistical learning Model .metric None down up smote adasyn rose ruro 1 LR accuracy 0.913 0.696 0.698 0.703 0.697 0.701 0.769 2 LR f_meas – 0.324 0.326 0.328 0.323 0.326 0.789 3 LR precision – 0.201 0.202 0.204 0.201 0.203 0.733 4 LR roc_auc 0.823 0.819 0.823 0.822 0.821 0.821 0.828 5 LR sensitivity 0 0.839 0.840 0.834 0.832 0.832 0.855 6 LR specificity 1 0.682 0.684 0.691 0.685 0.688 0.681 7 SVM accuracy 0.913 0.682 0.707 0.834 0.833 0.622 0.797 8 SVM f_meas – 0.322 0.338 0.376 0.372 0.296 0.821 9 SVM precision – 0.198 0.211 0.279 0.277 0.177 0.742 10 SVM roc_auc 0.785 0.818 0.827 0.824 0.820 0.828 0.853 11 SVM sensitivity 0 0.869 0.864 0.575 0.570 0.914 0.919 12 SVM specificity 1 0.664 0.692 0.859 0.858 0.594 0.673 On applying various methods to handle imbalanced class distribution, there was an overall reduction in the accuracy of the classical models. Among the widely used methods, Synthetic Minority Oversampling Technique (SMOTE) resulted in the highest accuracy, i.e. 70.3 % for LR and 83.4 % for the SVM. Further, there was a reduction in specificity across all widely used sub-sampling methods. The SMOTE algorithm generally performed better than the other methods available in literature, including the proposed RURO approach. However, in

terms of sensitivity, SMOTE was not the best among the common methods. The RURO approach performed well in this metric than any other method considered. The ROC_AUC 30

for logistic regression was generally lower after re-sampling. On the other hand, the metric improved for the SVM model. In general, the RURO approach outperformed all the methods in terms of ROC_AUC. In contrast to accuracy, there was an improvement in the F1 score from undetermined to at least 29 %. The SMOTE algorithm performed best compared to the common re-sampling approaches. However, RURO had the highest F1 score, 78.9 % and 82.1 % for logistic regression and support vector machines, respectively. A similar trend was noted for precision. 4.2.2 Ensemble learning Table 4.3 provides a summary of the results obtained from fitting algorithms from the ensemble framework using 5-fold cross validation. The two ensemble models considered were the random forest (RF) and the Extreme Gradient Boosting (XGBoost). The methods for handling class imbalance included, none, random undersampling, random oversampling, SMOTE, ADASYN, ROSE. We also explored whether stacking random undersampling and random oversampling (RURO) had an effect on performance of the ensemble models. For both the random forest and extreme gradient boosting, the accuracy was high (more than 90 %) when the models were fitted without handling class imbalance. The specificity for both models was also high i.e. 100 % for RF and 98 % for XGBoost. Just like the classical models, the sensitivity was very low. However, the ensemble methods were at least able to make some true positive predictions. The XGBoost model returned a 12.4 % sensitivity which was high compared to the other models that had almost 0 % without handling class imbalance. The area under the receiver operating characteristic curve (ROC_AUC) was 84.8% and 82.4% for the random forest and the XGBoost, respectively. In constrast to the classical models, where it was not possible to determine the F1 score and precision values, it was possible to estimate these metrics. The RF model had 0 % for both metrics while the XGoost had 18.3 % and 36.7 % scores for the F1 score and precision, respectively. When different methods to handle imbalanced class distribution were used, there was an overall reduction in the accuracy of the classical models. Among the widely used methods, both Synthetic Minority Oversampling Technique (SMOTE) and Adaptive Synthetic Sam- 31

Table 4.3: Ensemble learning Model .metric None down up smote adasyn rose ruro 1 RF accuracy 0.913 0.721 0.773 0.887 0.888 0.222 0.851 2 RF f_meas 0 0.338 0.367 0.268 0.258 0.183 0.867 3 RF precision 0 0.213 0.242 0.315 0.307 0.101 0.790 4 RF roc_auc 0.848 0.833 0.850 0.840 0.840 0.814 0.932 5 RF sensitivity 0 0.821 0.760 0.235 0.223 1 0.960 6 RF specificity 1.000 0.712 0.774 0.949 0.952 0.148 0.739 7 XGBoost accuracy 0.906 0.718 0.847 0.904 0.903 0.199 0.906 8 XGBoost f_meas 0.183 0.323 0.347 0.208 0.202 0.178 0.914 9 XGBoost precision 0.367 0.205 0.276 0.372 0.357 0.098 0.856 10 XGBoost roc_auc 0.824 0.802 0.819 0.826 0.823 0.818 0.961 11 XGBoost sensitivity 0.124 0.775 0.469 0.146 0.142 1 0.979 12 XGBoost specificity 0.980 0.713 0.883 0.976 0.976 0.123 0.832 pling (ADASYN) resulted in the highest accuracy. In addition, there was a reduction in the specificity of each ensemble model across all methods for handling class imbalance that were considered. The SMOTE and ADASYN algorithm generally performed better than the other methods available in literature, including the proposed RURO approach. However, in terms of sensitivity, both SMOTE and ADASYN performed poorly compared to other methods. The ROSE algorithm performed best, yielding 100 % sensitivity but very poor specificity of about 12-14 %. . The ROC_AUC for both models was generally lower after re-sampling. However, the RURO approach improved the outcome of this metric for both models. In contrast to accuracy, there was an improvement in the F1 score from 0 % for RF and 18.3 % for XGBoost to 36.7 % and 34.3 % among the widely used algorithms. The RURO approach resulted in the highest performance, 86.7 % for RF and 91.4 % for the XGBoost. Similar results were noted for precision. Based on these results, we can conclude that if the outcome of interest was the minority (positive) cases, then the XGBoost model paired with the RURO class imbalance handling approach performed best. Figure 4.3 shows the ROC curve of the XGBoost model as fitted using RURO for each fold. 32

Figure 4.3: XGBoost model fitted on each RURO fold 4.2.3 Performance on test set From the previous section, we noticed that the best performing model was the XGboost fitted on a balanced data set by the RURO approach. However, this result is an early estimate or peek on the the performance of the final model. Russell and Norvig (2002), suggested the importance of completely locking the test set away from model development. This is now a common approach in statistical learning, and for this reason, 20 % of the overall data set was set aside for the final evaluation of the best performing model. As shown in table 4.4 below, the model generally resulted in improved prediction on unseen test set at the expense of a slight decline in specificity. Table 4.4 .metric .estimate 1 accuracy 0.939 2 sens 0.995 3 spec 0.881 4 precision 0.896 5 f_meas 0.943 33

Chapter 5 Discussion 5.1 Key findings This paper explored the performance of both classical and ensemble learning methods using different methods to handle class imbalance. The learning algorithms considered were: Logistic regression (LR), support vector machines (svm), Random Forest (RF), and eXtreme Gradient Boosting (XGBoost). Kenya's 2020 Malaria Indicator Survey data was used as the study data set. First, the algorithms were first trained on an imbalanced data set. Secondly, the models were fitted after balancing the data using common methods such as:

Curiginal

90% MATCHING BLOCK 18/27 SA K.Ulaga Priya - Thesis for Plagarisim.doc (D133478724)

random under- sampling, random over-sampling, Synthetic Minority Over-sampling technique (SMOTE), Adaptive Synthetic Sampling Approach (ADASYN),

and Random Over Sampling Examples (ROSE) techniques. In addition, we evaluated the performance of these models when trained on a data set that was derived by combining random under-sampling and random over- sampling. Eighty percent of the data was used for model training, and the rest reserved for evaluating performance of the best performing model. As expected, for both the classical and ensemble methods, the accuracy was high (more than 90%) when the models were fitted without handling class imbalance. This is because the models are biased towards the majority (negative) class. For this reason, the models correctly predicted the negative cases (high specificity) and performed poorly in identifying the positive cases (almost 0 % sensitivity). Generally, in all aspects, the ensemble methods outperformed their classical counterparts. The Extreme Gradient Boosting (XGBoost) performed better than the random forest in all approaches except for the random undersampling method. Overall, there were strengths and weaknesses in each method that was considered. For example, the random under-sampling method performed poorly in the F1 score and precision metrics and relatively well on the others. The random over-sampling method performed poorly on the F1 score, precision and sensitivity. This was a similar trend for SMOTE 34

and ADASYN. The ROSE approach performed better on sensitivity compared to the other metrics. Finally, the proposed approach that involved combining random undersampling and random oversampling approaches (RURO) performed well on generally all metrics compared to any of the existing methods. This was further confirmed by the results from the unseen test set. 5.2 Limitations Although the results from this study are promising, it is important to note that a single data set was used. Hence, we do not have information on how the RURO approach can perform from data set to data set. Therefore, it will be important to study this proposed approach using numerous data sets. Furthermore, there exists a number of approaches to handle class imbalance such as Tomek's links, B-SMOTE, Near miss et cetera that were not considered. The other limitation was that, this paper used 5-fold cross validation. However, evidence in statistical learning literature suggests that we can get better performance by using repeated k fold validation where we can have more than 5 folds and at least 2 repeats. 5.3 Conclusion and recommendations Class imbalanced problems are common in the real world. There's no doubt about their negative effect on the performance of classification models. The main focus of this pa- per was to evaluate the predictive performance of various statistical learning algorithms by using different data level approaches for handling class imbalance. In addition, we ex- plored whether combining random under-sampling and random over-sampling can result in improved performance of the learning algorithms. In general, random under-sampling, random-oversampling, SMOTE, ADASYN and ROSE all had strengths and weaknesses depending on the performance metric. Evidence from this paper indicates that using our proposed approach, RURO, that involves combining randomly under-sampled and randomly over-sampled data is a simple yet potentially powerful method to improve performance 35

of a classification algorithm when the interest is to get a trade-off between sensitivity and specificity. 36

References Batista,

76% MATCHING BLOCK 19/27 SA New Microsoft Word Document (3).docx (D28496563)

G. E., Prati, R. C., and Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD explorations

newsletter, 6(1):20–29.

Burez, J. and Van den Poel, D. (2009). Handling class imbalance in customer churn prediction. Expert Systems with Applications, 36(3):4626–4636.

99% MATCHING BLOCK 20/27 W https://lirias.kuleuven.be/retrieve/584423

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. Journal of artificial intelligence research, 16:321–357. Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, pages 785–794. Chen,

T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., Chen, K., et al. (2015). Xgboost: extreme gradient boosting. R package version 0.4-2, 1(4):1–4. Fawcett, T. (2005). Roc analysis in pattern recognition.

71% MATCHING BLOCK 21/27 W https://lirias.kuleuven.be/retrieve/657601

Pattern Recognition Letters, 8:861– 874. Fernández, A., García, S., Galar, M., Prati, R. C., Krawczyk, B.,

and Herrera, F. (2018). Learning from imbalanced data sets, volume 10. Springer.

100% MATCHING BLOCK 24/27 W https://lirias.kuleuven.be/retrieve/584423

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. Annals of statistics, pages 1189–1232.

Guo, X., Yin, Y., Dong, C., Yang, G., and Zhou, G. (2008).

100% MATCHING BLOCK 25/27 SA report.docx (D26218015)

On the class imbalance problem. In 2008 Fourth international conference on natural computation,

volume 4, pages 192–201. IEEE. Habyarimana, F. and Ramroop, S. (2020). Prevalence and risk factors associated with malaria among children aged six months to 14 years old in rwanda: evidence from 2017 rwanda malaria indicator survey. International Journal of Environmental Research and Public Health, 17(21):7975. Hastie, T., Tibshirani, R., Friedman, J. H., and Friedman, J. H. (2009).

100% MATCHING BLOCK 22/27 W https://lirias.kuleuven.be/retrieve/584423

The elements of statistical learning: data mining, inference, and prediction, volume 2. Springer.

100% MATCHING BLOCK 23/27 W https://lirias.kuleuven.be/retrieve/584423

He, H., Bai, Y., Garcia, E. A., and Li, S. (2008). Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In 2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence), pages 1322–1328. IEEE.

Hlavac, M. (2015). stargazer: beautiful latex, html and ascii tables from r statistical output. Huda, S., Liu, K., Abdelrazek, M., Ibrahim, A., Alyahya, S., Al-Dossari, H., and Ahmad, S. (2018). An ensemble oversampling model for class imbalance problem in software defect prediction. IEEE access, 6:24184–24195. James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). An introduction to statistical learning, volume 112. Springer. 37

Kuhn, M. and Wickham, H. (2020). Tidymodels: a collection of packages for modeling and machine learning using tidyverse principles. Boston, MA, USA.[(accessed on 10 December 2020)]. Leevy, J. L., Khoshgoftaar, T. M., Bauder, R.

86% MATCHING BLOCK 27/27 SA K.Ulaga Priya - Thesis for Plagarisim.doc (D133478724)

A., and Seliya, N. (2018). A survey on addressing high-class imbalance in big data. Journal of Big Data, 5(1):1–30.

Curiginal

Ling, C. X. and Sheng, V. S. (2008). Cost-sensitive learning and the class imbalance problem. Encyclopedia of machine learning, 2011:231–235. Longadge, R. and Dongre, S. (2013). Class imbalance problem in data mining review. arXiv preprint arXiv:1305.1707. Menardi, G. and Torelli, N. (2014). Training and assessing classification rules with imbal- anced data. Data mining and knowledge discovery, 28(1):92–122. Ministry of Health Division of National Malaria Programme Kenya, Kenya National Bureau of Statistics, and The DHS Program (2021). Kenya Malaria Indicator Survey 2020. Russell, S. and Norvig, P. (2002). Artificial intelligence: a modern approach. StataCorp, L. (2019). Stata statistical software: Release vol. 16. Sun, Y., Wong, A. K., and Kamel, M. S. (2009). Classification of imbalanced data: A review. International journal of pattern recognition and artificial intelligence, 23(04):687–719. Weiss, G. M. and

100% MATCHING BLOCK 26/27 W https://azpdf.tips/learning-from-imbalanced-da ...

Provost, F. (2003). Learning when training data are costly: The effect of class distribution on tree induction.

Journal of artificial intelligence research, 19:315–354. WHO (2020). The top 10 causes of death. https://www.who.int/news-room/fact-sheets/ detail/the-top-10-causes-of-death. Accessed: 2021-03-02. WHO (2021). Word Malaria Report 2021. Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., Grolemund, G., Hayes, A., Henry, L., Hester, J., et al. (2019). Welcome to the tidyverse. Journal of open source software, 4(43):1686. Winskill, P., Rowland, M., Mtove, G., Malima, R. C., and Kirby, M. J. (2011). Malaria risk factors in north-east tanzania. Malaria Journal, 10(1):1–7. 38

Appendix A Appendix 39

Curiginal

Hit and source - focused comparison, Side by Side

Submitted text As student entered the text in the submitted document.

Matching text As the text appears in the source.

1/27 **SUBMITTED TEXT** 15 WORDS **70%** **MATCHING TEXT** 15 WORDS

Submitted in total fulfilment of the requirements for the degree of Master of Science in

SA report.docx (D26218015)

2/27 **SUBMITTED TEXT** 10 WORDS **100%** **MATCHING TEXT** 10 WORDS

random under-sampling, random over-sampling, Synthetic Minority Over- sampling technique (SMOTE),

SA K.Ulaga Priya - Thesis for Plagarisim.doc (D133478724)

3/27 **SUBMITTED TEXT** 10 WORDS **76%** **MATCHING TEXT** 10 WORDS

SMOTE Synthetic minority over-sampling technique BSMOTE Borderline synthetic minority over-sampling technique

SA K.Ulaga Priya - Thesis for Plagarisim.doc (D133478724)

4/27 **SUBMITTED TEXT** 12 WORDS **88%** **MATCHING TEXT** 12 WORDS

Advanced undersampling techniques include: evolutionary undersampling, undersampling by cleaning data, ensemble based undersampling,

Advanced Undersampling Techniques . . . . . 5.3.1 Evolutionary Undersampling . . . . . 5.3.2 Undersampling by Cleaning Data . . . . . 5.3.3 Ensemble Based Undersampling 5.3.4

W https://azpdf.tips/learning-from-imbalanced-data-sets-pdf-free.html

5/27 **SUBMITTED TEXT** 16 WORDS **68%** **MATCHING TEXT** 16 WORDS

tend to have high accuracy for the majority class but poor results for the minority class.

tend to have a great accuracy for the majority class while obtaining poor results (closer to 0%) for the minority class.

W https://azpdf.tips/learning-from-imbalanced-data-sets-pdf-free.html

Curiginal

6/27	SUBMITTED TEXT	44 WORDS	32%	MATCHING TEXT	44 WORDS

are defined as: • True Positives (TP) : the number of positive examples correctly classified. • True Negatives (TN) : the number of negative examples correctly classified. • False Positives (FP) : the number of positive examples incorrectly classified. • False Negatives (FN) : the number of

SA paper 2.docx (D37901395)

7/27	SUBMITTED TEXT	15 WORDS	82%	MATCHING TEXT	15 WORDS

Number of correct predictions Total number of predictions = TP + TN TP + TN + FP + FN

SA final paper review.docx (D91005008)

9/27	SUBMITTED TEXT	12 WORDS	95%	MATCHING TEXT	12 WORDS

The support vector machine is an extension of the support vector classifier

The support vector machine (SVM) is an extension of the support vector classifier.

W https://www.diva-portal.org/smash/get/diva2:1519153/FULLTEXT01.pdf

14/27	SUBMITTED TEXT	23 WORDS	58%	MATCHING TEXT	23 WORDS

the new set of predictors (m) to be considered at each split is equal to the square root of the total number of

the full set of p predictors. Typically, the number of predictors considered at each split is approximately equal to the square root of the total numbers of

W https://www.diva-portal.org/smash/get/diva2:1519153/FULLTEXT01.pdf

8/27	SUBMITTED TEXT	18 WORDS	69%	MATCHING TEXT	18 WORDS

For this reason, this approach is said to operate on the "feature space" rather than the "data space"

For this reason, the procedure is said to be focused on the “feature space” rather than on the “data space”.

W https://azpdf.tips/learning-from-imbalanced-data-sets-pdf-free.html

Curiginal

10/27	SUBMITTED TEXT	32 WORDS	39%	MATCHING TEXT	32 WORDS

a positive class example is selected randomly from the training set and its k-nearest neighbours (say 5) are determined. Lastly, n of these k examples are randomly selected to generate new instances via interpolation.

a positive class instance is selected at random from the training set. Next, its KNN (5 by default) are obtained. Finally, N of these K instances are randomly chosen to compute the new instances by interpolation.

W https://azpdf.tips/learning-from-imbalanced-data-sets-pdf-free.html

11/27	SUBMITTED TEXT	42 WORDS	28%	MATCHING TEXT	42 WORDS

feature space and y i ∈ Y = {P,N} is the response label, positive or negative, associated with x i . 22 2. Define n p and n n be the number of minority (positive) class instances and the number of majority (negative) class instances, respectively

feature space M and yi ∈ C = {pos, neg} is the class identity label associated with xi . Define and N – as the number of minority class examples and the number of majority class examples, respectively.

W https://azpdf.tips/learning-from-imbalanced-data-sets-pdf-free.html

12/27	SUBMITTED TEXT	24 WORDS	59%	MATCHING TEXT	24 WORDS

preset threshold ratio for the maximum tolerated degree of class imbal- ance. If d > d th : (a) Determine the number of synthetic

preset threshold for the maximum tolerated degree of class imbalance ratio. If d > dth : (a) Calculate the number of synthetic

W https://azpdf.tips/learning-from-imbalanced-data-sets-pdf-free.html

13/27	SUBMITTED TEXT	19 WORDS	70%	MATCHING TEXT	19 WORDS

β = 1, then a fully balanced data is created after the generalized process. (b) For each x i ∈ minority

β = 1 means a fully balanced data set is created after the generalization process. (b) For each minority

W https://azpdf.tips/learning-from-imbalanced-data-sets-pdf-free.html

17/27	SUBMITTED TEXT	17 WORDS	82%	MATCHING TEXT	17 WORDS

is the number of examples in the K nearest neighbours of x i which belong to the majority (

SA Shidha. M V.pdf (D95772883)

Curiginal

15/27	SUBMITTED TEXT	29 WORDS	50%	MATCHING TEXT	29 WORDS

one of the two classes and generating a new instance in the neighborhood {x ∗ ,y ′∗ }. The shape of the countour sets of K and

one of the two classes, and generate a new example {x ∗ , y′∗} in its neighborhood, where the shape of the neighborhood is determined by the shape of the contour sets of K and

W https://azpdf.tips/learning-from-imbalanced-data-sets-pdf-free.html

16/27	SUBMITTED TEXT	17 WORDS	66%	MATCHING TEXT	17 WORDS

and n n be the number of minority class instances and the number of majority class instances, respectively

and N – as the number of minority class examples and the number of majority class examples, respectively.

W https://azpdf.tips/learning-from-imbalanced-data-sets-pdf-free.html

18/27	SUBMITTED TEXT	14 WORDS	90%	MATCHING TEXT	14 WORDS

random under- sampling, random over-sampling, Synthetic Minority Over-sampling technique (SMOTE), Adaptive Synthetic Sampling Approach (ADASYN),

SA K.Ulaga Priya - Thesis for Plagarisim.doc (D133478724)

19/27	SUBMITTED TEXT	26 WORDS	76%	MATCHING TEXT	26 WORDS

G. E., Prati, R. C., and Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD explorations

SA New Microsoft Word Document (3).docx (D28496563)

20/27	SUBMITTED TEXT	51 WORDS	99%	MATCHING TEXT	51 WORDS

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. Journal of artificial intelligence research, 16:321–357. Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, pages 785–794. Chen,

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research, 16:321–357. Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, ACM. Chen,

W https://lirias.kuleuven.be/retrieve/584423

Curiginal

21/27	SUBMITTED TEXT	15 WORDS	71%	MATCHING TEXT	15 WORDS

Pattern Recognition Letters, 8:861– 874. Fernández, A., García, S., Galar, M., Prati, R. C., Krawczyk, B.,

Pattern recognition letters 27, 861{874. Fern andez, A., Garc a, S., Galar, M., Prati, R.C., Krawczyk, B.,

W https://lirias.kuleuven.be/retrieve/657601

24/27	SUBMITTED TEXT	15 WORDS	100%	MATCHING TEXT	15 WORDS

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. Annals of statistics, pages 1189–1232.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. Annals of statistics, pages 1189–1232.

W https://lirias.kuleuven.be/retrieve/584423

25/27	SUBMITTED TEXT	13 WORDS	100%	MATCHING TEXT	13 WORDS

On the class imbalance problem. In 2008 Fourth international conference on natural computation,

SA report.docx (D26218015)

22/27	SUBMITTED TEXT	12 WORDS	100%	MATCHING TEXT	12 WORDS

The elements of statistical learning: data mining, inference, and prediction, volume 2. Springer.

The elements of statistical learning: data mining, inference, and prediction, volume 2. Springer

W https://lirias.kuleuven.be/retrieve/584423

23/27	SUBMITTED TEXT	35 WORDS	100%	MATCHING TEXT	35 WORDS

He, H., Bai, Y., Garcia, E. A., and Li, S. (2008). Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In 2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence), pages 1322–1328. IEEE.

He, H., Bai, Y., Garcia, E. A., and Li, S. (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), pages 1322–1328. IEEE. 168

W https://lirias.kuleuven.be/retrieve/584423

Curiginal

27/27	SUBMITTED TEXT	18 WORDS	86%	MATCHING TEXT	18 WORDS

A., and Seliya, N. (2018). A survey on addressing high-class imbalance in big data. Journal of Big Data, 5(1):1–30.

SA K.Ulaga Priya - Thesis for Plagarisim.doc (D133478724)

26/27	SUBMITTED TEXT	17 WORDS	100%	MATCHING TEXT	17 WORDS

Provost, F. (2003). Learning when training data are costly: The effect of class distribution on tree induction.

Provost, F.: Learning when training data are costly: the effect of class distribution on tree induction.

W https://azpdf.tips/learning-from-imbalanced-data-sets-pdf-free.html