uc3m | Universidad Carlos III de Madrid

Master Degree in Statistics for Data Science
2019-2020

Master Thesis

"A Study on Imbalanced classes in Machine Learning"

Kambhampati Jyothi

Dr. Iñaki Úcar Marques

Madrid, September 2020

SUMMARY

This master thesis is a study about handling an imbalanced classification problem where atleast one of the classes constitutes only a very small minority of the data. In this thesis, we have evaluated the performance of various linear and non-linear classifiers such as logistic regression, LDA, random forest, support vector machines etc and reviewed methods such as joint sampling techniques, alternative cutoffs for classifiers thresholds to deal with imbalanced classification. We identified the few performance metrics such as precision, recall, F-score in order to evaluate classifier performance when combined with methods dealing with imbalance. We applied our learning’s to a case study based on bank marketing data after data pre-processing including feature selection methods based on RFE, Boruta and XGBoost. Processed data is used to create new balanced datasets using joint sampling techniques such as SMOTE, NCL, Tomek, K-means clustering which are then feeded as input to the classifier. With this approach, we realized there is information loss as we are modifying the data in order to remove imbalance and it is challenging to find how much amount of sampling has to be done. We then reviewed another alternative approach of finding optimal threshold used by classifier using ROC and PR curves. We established maximizing F-score using PR curve as a better alternative to find the optimal value for cut off. With this approach, we were able to overcome the problem of information loss seen with joint sampling methods.

Keywords: Imbalanced data, Machine Learning, Over/Under Samplings, Classification, Optimal Threshold, ROC / PR curve, Feature selection.

DEDICATION

Foremost, I would like to express my sincere gratitude to my advisor Dr. Iñaki Úcar Marques who read some lines of a very incipient thesis proposal and believed enough in what he read to accept becoming my supervisor. Thank you for your patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of study and writing of this thesis.

I am also grateful to all the teaching staff for giving some great advices during failures, motivated and guiding me all throughout this course and made me a better person professionally.

I would like to pay my special regards to all my Colleagues who helped me learn Spanish culture. Especially some of my good friends David, Manu, Miguel, Josué, Alejandro, Andres, Ignacio and Javier. They thought me handle difficult situations, encouraged me for being competitive with others.

A very special word of thanks to my family for their continuous and unparalleled love, help and support. I am forever indebted to my parents for supporting me throughout the years, financially, practically and with moral support and also giving me the opportunities and experiences that have made me who I am.

Finally, I have to thank my husband, Abhinay, for always being there for me as a friend. It is whole-heartedly appreciated that your great advice for my study proved monumental towards the success of this study. Thank you!

Thank you all….!!!

Table of Contents

List of figures

List of Tables

1. Introduction

1.1. Motivation of Work

Now days, classification is most general problem which identifies the set of categories.

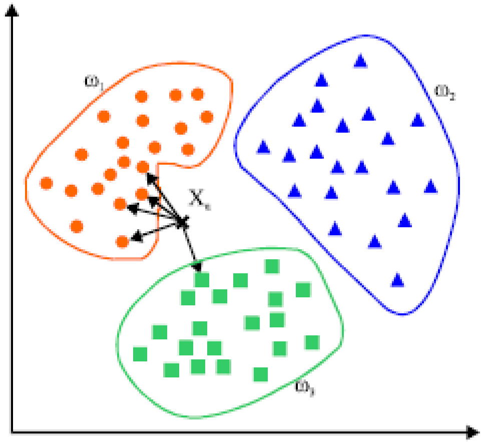


Figure 1: Example of Classification.

An Imbalanced classification is an example for classification. This is a case where the number of observations which belongs to one class (minority class or positive class) is significantly lower than those belonging to other classes (majority class or negative class). This situation is mostly seen in cases where anomaly detection is the key such as fraudulent transactions, identification of rare diseases, identification of electricity thefts, identifying bombs and pattern recognition etc. The number of majority classes over minority classes is known as Imbalanced ratio (IR ratio) [1].

The problem is when the class of interest is the majority class and the minority class is often ignored. Imbalanced classification is very challenging with respect to determining the performance of the classifiers. For example a dataset has 99% of the data as majority class and 1% of data as minority class (usually 99:1 IR ratio). In these cases, most classifiers always prioritize the majority class and give an accuracy of 99% which is false. The consequences of this drawback can affect the performance of the classifiers when applied to an Imbalanced problem. This is because of the fact that the classifiers optimize the accuracy and build a model that is similar to the naive model. It is obvious that such classifiers with high accuracies are of no use without considering the minority class as the class of interest.

As a result, there are several techniques to address this Imbalanced class problem. One such is the re-sampling techniques. Sampling methods (Over/Under sampling) are the most popular Standard approaches to solve Imbalanced problems. Although, it is easy to use, re-sampling has its own issues because you are modifying the balance artificially, and the class balance also contains information. In order to avoid Information loss, we explored an alternative approach by optimizing the Probability cut-off. In general, most classifiers have a default cut-off of 50-50 and we have implemented to find the Optimal.

Probability threshold using ROC and PR curves using F-score as the evaluation metric. Finally, we compare the performance of the classifiers in combination with Re-sampling methods and optimal threshold method and discuss the results in further chapters.

1.2. Goals

To achieve this, we will try to build a step-by-step understanding of each and every method. Given below steps to solve this Imbalanced class problem.

- First, understand the importance of the classifiers and explore a number of typical classification algorithms.
- Review methods to deal with Imbalanced data i.e., sampling methods and Optimal Threshold method.
- Applying these methods to a Case Study: Bank Marketing data
- Compare them with a proper metric: F-Score
- Discuss the results.

The master thesis has been organized in the following way: Section 1 is about the Introduction of the topic, explaining the motivation behind choosing the topic and the goals of this project. Section 2 consists of various Classification algorithms, Re-sampling techniques and Evaluation metrics. Section 3 is the methodology and explaining the data along with its pre-processing steps. Section 4 is about the results of a Case study and Section 5 is the Conclusion.

2. Related Work

2.1. Classification Algorithms

A classifier mainly classifies the data points into different classes known as Labels/Targets or categories. Consider, for example a fraudulent dataset consists of a “Fraud” and a “Non-Fraud” transactions and the classifier has to identify which of those are fraud/no fraud transactions. This is an example for binary classification. On the other hand, if there are more than two classes, then the problem is named as multiclass or multinomial classification. There are two different approaches in machine learning: Supervised models and unsupervised models. In supervised models, a classifier uses some of the input data as *Training data* to understand how the input variables relate to the class. That means, it understands which of those transactions are frauds/no. Once the classifier is accurately trained, some *Testing data* is used to compare with that to determine if there is any fraudulent transaction.

Today, we have so many classifiers available but we cannot conclude which ones are the best as it completely depends on the application. They are mainly used in Artificial Intelligence, Machine Learning and Deep Learning fields. Some of the classifiers are Decision Trees, Naive Bayes, Random Forest, k-Nearest Neighbor (kNN), and Logistic Regression, Support Vector machine, Artificial Neural networks etc. Some of the Classification algorithms that are used in this project are explained below.

2.1.1 Linear Classifiers

Logistic Regression

Logistic regression is a supervised linear classification algorithm for which the output variable ‘y’ or target variable has only *discrete categorical values* for the given set of input variables [2]. It is a predictive algorithm based on the concept of probability. For binary classification, we can draw a hypothesis function to fit, known as logistic function or sigmoid function. A Sigmoid function has *S* curved shape taking values between 0 and 1 using sigmoid function.

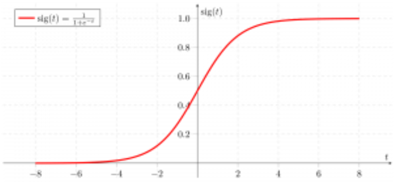


Figure 2: Sigmoid Function

$$\text{Sigmoid Function } \sigma(z) = \frac{1}{1+e^{-z}} \text{ where z } = \sum w_i x_i + bias.$$

Note:
$\sigma(z)$ = Output between 0 and 1, z = input value and e = base of natural log.

The logistic regression for classification is based on the decision threshold value. The decision of the threshold is dependent on precision and recall values (we will see in later sections).

Linear Discriminant Analysis (LDA):

Linear Discriminant analysis is also a supervised predictive modelling algorithm, which is mainly used in machine learning, Statistics and Pattern recognition. LDA is used to find the linear combination of attributes which separates into two or more classes [3]. The resulting combinations of attributes are used as linear classifiers. This algorithm is also used for multi-class classification performs dimensionality reduction with good class discrimination to avoid Over-fitting problems.

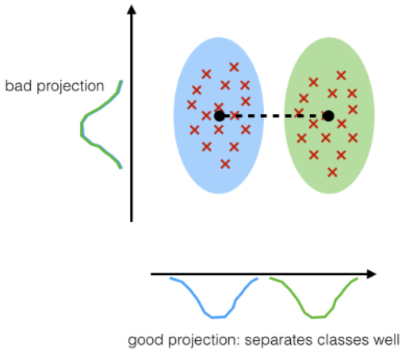


Figure 3: Linear Discriminant Analysis

2.1.2. Non-Linear Classifiers

k- Nearest Neighbors (kNN):

k-Nearest Neighbors is a simple Classification algorithm which is mainly used for highly non-linear data. The performance is not good for higher dimensions. This algorithm identifies similar data points that are close to each other. It is easy to use and low computational time.

Algorithm below [4] explains how a k-NN works.
Step 1: Load the data and initialize k number of neighbors.
Step 2: Calculate the distance between the new observation to classify, z, and all the current observations.
Step 3: Select the k nearest observation to z and sort them from smallest to largest.
Step 4: Classify z with largest proportion and return the mode.

Naive Bayes:

Naive Bayes is a probabilistic machine learning algorithm [5] widely used for classification problems. The algorithm is based on Bayes theorem with an assumption of conditional independence amongst each predictor. It is widely used for Text classification and sentiment analysis. This classifier is easy to build and good for very large datasets.

Bayes Theorem:

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$

- P(x|c) is the probability of predictor given class.
- P(c) is the class prior probability.
- P(c|x) is the posterior probability of the class.
- P(x)is the predictor prior probability.

Neural Networks:

Neural networks are one of the most powerful tools in Artificial Intelligence and Deep Learning, to solve higher dimensional data problems [6]. They are a very efficient and complex model which works similar to our human brain. Basically, a neural network consists of several layers which are interconnected through a network. Each layer consists of several neurons and each neuron takes the input from its previous layer, process and outputs to the next layer and so on. Neural Networks are undoubtedly incredible when it comes to accuracy, but it takes a very high computational time. They are widely used for Image recognition applications and natural language processing, autonomous cars etc.

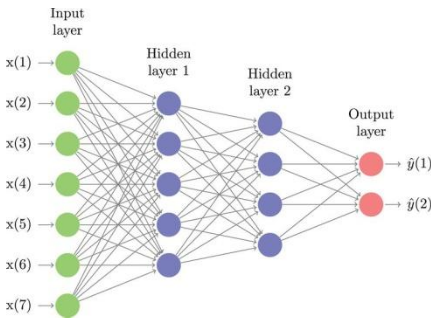
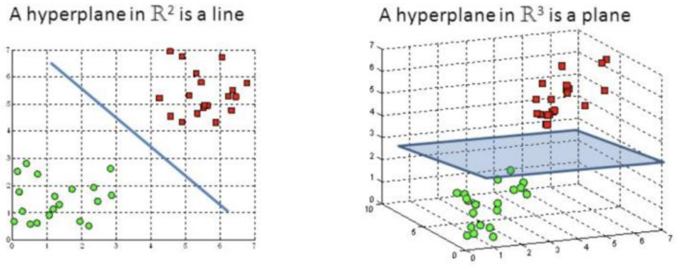


Figure 4: Neural networks

Support Vector Machine (SVM):

Support vector machine is a machine learning algorithm which is the mostly preferred technique in higher dimensions with good accuracy and less computation power. SVM’s are very expensive to train large complex problems in general, but can handle non-linear data. SVM basically identifies the optimal Hyperplane (Decision boundary) in an n-dimensional space and classifies the data points into two classes [7]. A hyperplane is a line in two dimensional space where as it is a plane in three dimensional space. The data points represents Support vectors, when changed will impact the position of the hyperplane.


A hyperplane in $\mathbb{R}^2$ is a line

A hyperplane in $\mathbb{R}^3$ is a plane

Figure 5: A Hyper plane separating two classes.

2.1.3. Tree based classifiers

Decision Tree:

A Decision Tree is also a supervised learning model [8] used for both Classification and regression problems. It uses the tree representation to solve large complex problems where an internal node represents feature (or attribute), the branch represents a decision rule, and each leaf node represents the outcome. The top most node is known as root node. The two types of nodes: Decision nodes, used to make any decision and the leaf nodes, are the outcomes of those decision nodes. Basically, the decision tree learns to partition based on the feature value and continues to partition recursively.

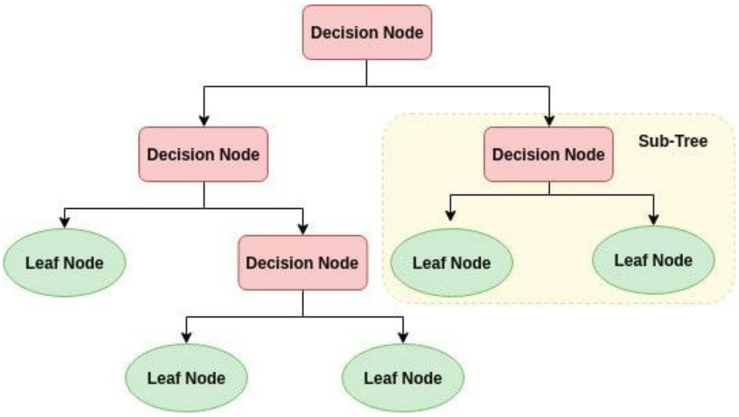


Figure 6: Decision Tree Algorithm

Step 1: Start the tree with a root node.
Step 2: Identify the best attribute using Attribute Selection Measure (ASM).
Step 3: Partition the root node into sub nodes that contains the best attributes.
Step 4: Build the decision tree node, which contains the best attribute and continue to make new Decision trees recursively until you reach a point where you cannot partition the nodes anymore and have leaf nodes.

Iterative Dichotomiser 3(ID3) Decision Tree:

The ID3 algorithm [9] starts at the Set A as root node and iterates through every used attribute and then calculates its entropy. Now it selects the attribute which has the smallest entropy and partitions into subsets of data. The algorithm recursively continues the procedure for all its subsets considering only the unused attributes.

$$\text{Entropy (A)} = \sum_{x \varepsilon X} -p(x) \log_2 p(x)$$

- A is the original Dataset.
- p(x) is the proportion to the number of elements in class x to number of elements in A.
- X is the set of classes in A.

C5.0 Decision Tree:

C5.0 decision tree algorithm [10-11] is one of the most used standard algorithms in industry today. It performs really well when compared to Neural networks and Support vector Machine and it is very easy to understand and install. This algorithm can solve any type of Classification problem.

Random Forests:

Random forest is one of the most popular machine learning algorithms. It uses supervised learning methods for both Classification and Regression modelling. But however, it is widely used for Classification problems. The Random forest algorithm is

Based out of an ensemble technique which combines multiple classifiers to solve a large complex problem [12]. The random forest algorithm creates various Decision trees on training data samples, then gets the prediction values from each Decision tree, and finally picks the best prediction by maximum voting. The higher number of decision trees, the higher accuracy and also prevents over fitting problems.

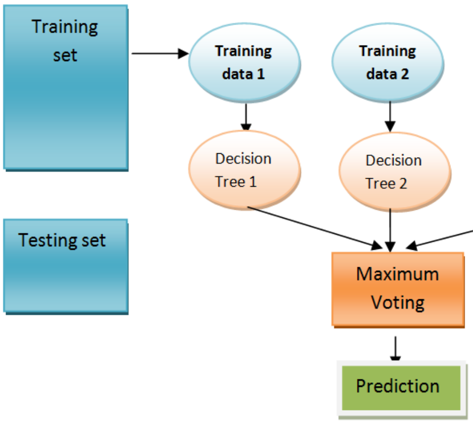


Figure 7: Random Forest Algorithm

Algorithm explains how a random forest works?
Step 1: First, select *n* random data samples from the training data set.
Step 2: Build the decision trees associated to each data sample.
Step 3: Get the prediction result from each Decision Tree.
Step 4: Voting will be done on every prediction.
Step 5: Best prediction value is picked based on Maximum voting's.

XGBOOST:

XGBOOST stands for Extreme Gradient Boosting is an advanced implementation of Gradient Boosting algorithm [13]. This algorithm is the most powerful machine learning algorithms in which accuracy and speed are most concerned. It is highly sophisticated tool which can deal all types of data irregularities. This algorithm is easy to use but difficult while improving the model since XGBOOST uses multiple parameters for parameter tuning.

2.2. Data Re-Sampling Methods

Sampling methods are the most common solution for Imbalanced Class problem. The objective of the sampling methods is to build a balanced dataset such that the classifiers can perform well with better accuracy. The two types of sampling methods: Under-sampling and Over-sampling methods are discussed below.

2.2.1. Under Sampling Methods

In Random under-sampling methods, the majority class observations are discarded in order to make more balanced dataset. For example, if there are 55 majority class instances and 5 minority class instances with an IR ratio of 11:1. Now, 20 majority class instances are discarded and we are left with only 35 majority and 5 minority class instances with an IR ratio of 7:1. In this example, we are trying to reduce the Imbalanced ratio of the dataset and make it to a balanced dataset. Let us discuss some of the Under Sampling methods used in this work.

Neighborhood Cleanup Method (NCL)

The NCL algorithm is an Imbalanced algorithm consists of two phases. In the first phase, Edited Nearest Neighbor (ENN) method is employed to Undersample the negative examples whose class labels are misclassified. In the second phase, the neighbors of the positive examples are identified and those belonging to the majority class are discarded [14].

Tomek Link Removal Method

A pair of examples belonging to different classes is neighbors to each other's is known as Tomek link. Under sampling is performed on all the Tomek links associated to the Imbalanced dataset and removes the majority class instances. The objective of this algorithm is to make boundary decision efficiently such that minority class is more distinct than majority class [15].

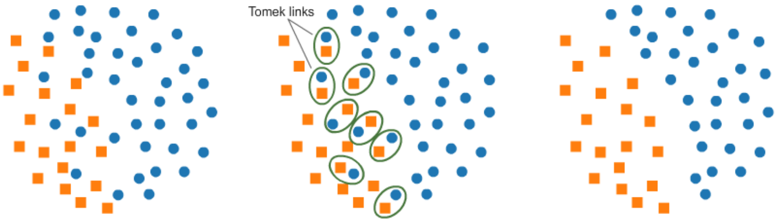


Figure 8: Tomek links removal

K-means Clustering under-sampling

Removing of data using Under-sampling method might lead to loosing of some important information. In order to avoid that, we can use a Clustered based under sampling technique. In this approach, we cluster all the training data into k clusters and then choose an appropriate cluster which has suitable amount of majority class examples. That means, there will be some clusters which has high majority class examples and low minority examples and vice-versa. In this approach, we will choose a cluster which has balanced amount of both the classes [16].

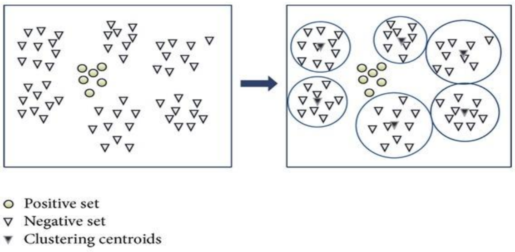


Figure 9: Under sampling of k clusters.

2.2.2. Over Sampling Methods

Over-sampling methods also tries to mimic the data to make a more balanced dataset.

Random Over-sampling

Random Oversampling provides a traditional re-arrangement of data that can always result to over fitting of the models. This is generally not a recommended approach for Imbalanced data problems especially Pattern Recognition or any other Imbalanced classification problems. It can be used for simple applications but surely not for real time applications [17].

Synthetic Minority Oversampling Technique (SMOTE)

In this approach, SMOTE randomly identifies the minority class instance and finds its k nearest neighbor and tries to find the decision boundary by calculating the distance between two randomly chosen nearest points[18]. Considering the previous example with 55 majority class instances and 5 minority class instances and an IR ratio of 11:1. Now, some 20 samples are added to the minority class resulting to 55 majority and 25 minority class instances and having an Imbalanced ratio of 11:5 which means, we are trying to achieve a balanced dataset.

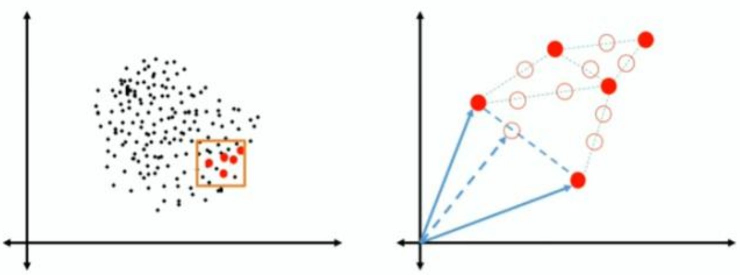
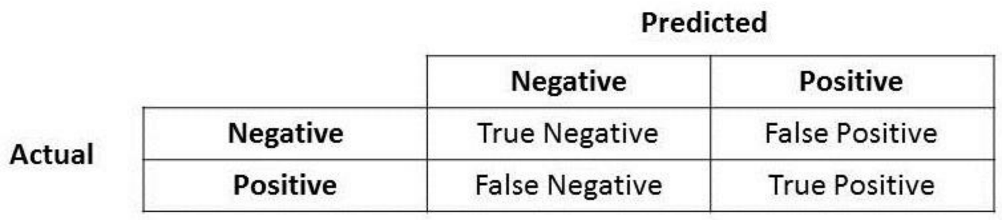


Figure 10: SMOTE

2.3. Evaluation Metrics

The most common method for evaluating the performance of a classifier is using of Confusion matrix.



		Predicted	
		Negative	Positive
Actual	Negative	True Negative	False Positive
	Positive	False Negative	True Positive

Figure 11: Confusion Matrix

True positives (TP): These are cases in which we predicted yes and the actual is yes.
True negatives (TN): We predicted no, and the actual is no.
False positives (FP): We predicted yes, but the actual is no (Type I error).
False negatives (FN): We predicted no, but the actual is yes (Type II error).

There are many evaluation metrics from confusion matrix [19]. One of the basic standards metric is the accuracy.

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

Although, we can use accuracy metric for performance evaluation, but in some cases like the dataset consists of 99% data of majority and 1% data from minority class. The accuracy seems to be high if one considers the majority class. Moreover, accuracy can sometimes leads to wrong conclusions as there are Type 1 and Type 2 errors. Also there are many alternative metrics for evaluating the performance of the classifiers for Imbalanced Data.

Precision: Precision measures how often an instance that was predicted as positive is actually positive [19].

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

Recall: Recall measures how often a positive class instance in the dataset was predicted as a positive class instance by the classifier [19].

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

F_β Score: Final common metric F_β measure attempts to measure the tradeoff between precision and recall by outputting a single value that reflects the goodness of the classifier in the presence of rare cases. Here β is the relative importance of precision and recall [20].

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

***Sensitivity*:** Sensitivity measures how often a classifier can recognize positive examples.

$$\text{Sensitivity = Recall} = \frac{\text{TP}}{\text{TP+FN}}$$

***Specificity*:** Specificity measures how often a classifier can recognize negative examples.

$$\text{Specificity} = \frac{TN}{TN+FP}$$

3. Methods

3.1. Data description

The data set studied in this master thesis was collected and summarized by a Portuguese retail bank. This is a publicly available data set that is accessible via URL: http://archive.ics.uci.edu/ml/datasets/Bank+Marketing.

This data is related to direct marketing campaign done by banking institution and it contains a total of 17 attributes of 45,211 customers. The marketing campaigns were based on direct phone calls where the manager called the customers to explain the product (bank term deposit) and afterwards the bank records the result whether the customer finally purchased the product(bank term deposit). The dataset is highly imbalanced the positive class accounts for 13% of all records.

The first 16 attributes of the data set are features, which are divided into three categories (see Table 1, Table 2 and Table 3 respectively). The last attribute of the data set is the class label, which is also the target variable of the dataset for classification problems: Whether the customer has made a term deposit (Yes: 1; No: 0).

The first type of features is the basic information of the customer (Table 1). The second type of features includes the last telephone communication in the current marketing activity (Table 2). The third type of features contains relevant information about previous marketing activities and current marketing activities (Table 3).

Table 1: Customer Information

Name	Type	Description	Levels
Age	Numeric	-	-
Job	Categorical	type of job	admin., unknown, unemployed, management, housemaid, entrepreneur, student, blue-collar, self-employed, retired, technician, services
marital	Categorical	marital status	married, divorced, single
education	Categorical	-	unknown, primary, secondary, tertiary
default	Categorical	Has credit in default?	yes, no
balance	Numeric	Average annual deposit amount in EUR	-
housing	Categorical	Has housing loan?	yes, no
Loan	Categorical	Has personal loan?	yes, no

Table 2: Last Contact Information.

Name	Type	Description	Levels
contact	Categorical	Type of communication	telephone, cellular
day	Numeric	Last contact day of the month	-
month	Categorical	Last contact month of the year	-
duration	Numeric	Last contact time (seconds)	-

Table 3: Previous and Current Marketing Information

Name	Type	Description	Levels
campaign	Numeric	Number of times contacted	-
pdays	Numeric	Number of days since the last contact	999 means client was not previously contacted
previous	Numeric	Number of times contacted before current campaign	-
poutcome	Categorical	Outcome of the previous marketing campaign	unknown, other, failure, success

The age distribution of the bank's customers is concentrated between 30 and 60 years with the most people between 30 and 40 years. The average annual deposit of the bank's customers is concentrated in the range of 0 to 1000 EUR and about 8% of people have zero deposits.

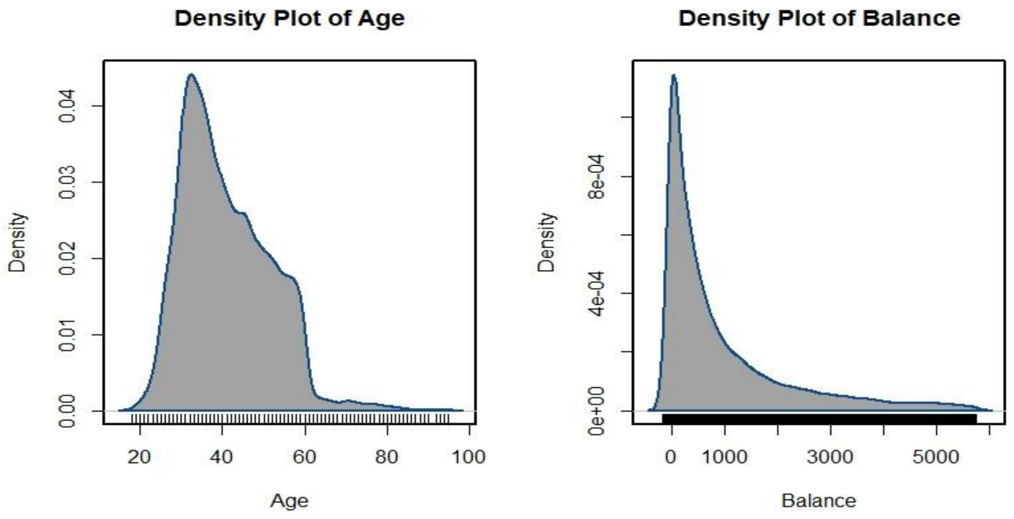


Figure 12: Age distribution of customers and distribution of average annual deposit amount.

The monthly last contact date and duration between the bank and the customer in the current marketing activities can be seen Figure 13. We observed that the most of the bank customers are contacted in the middle of each month (median 16) while last phone contact duration mainly distributed in 100 to 250 seconds.

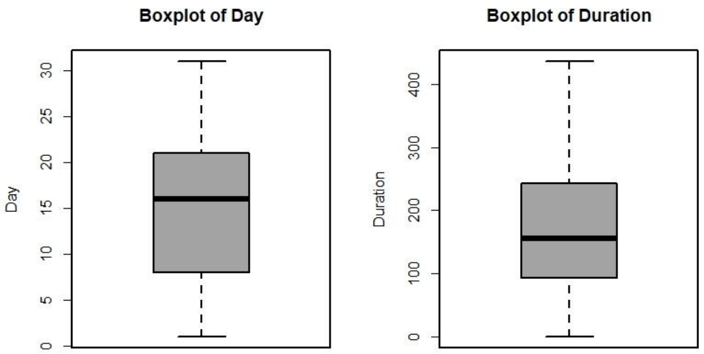


Figure 13: The monthly last contact date and duration (in seconds)

Understanding the distribution of customer occupational groups we seen there are more customers belonging to blue collars, managers and technicians than others.

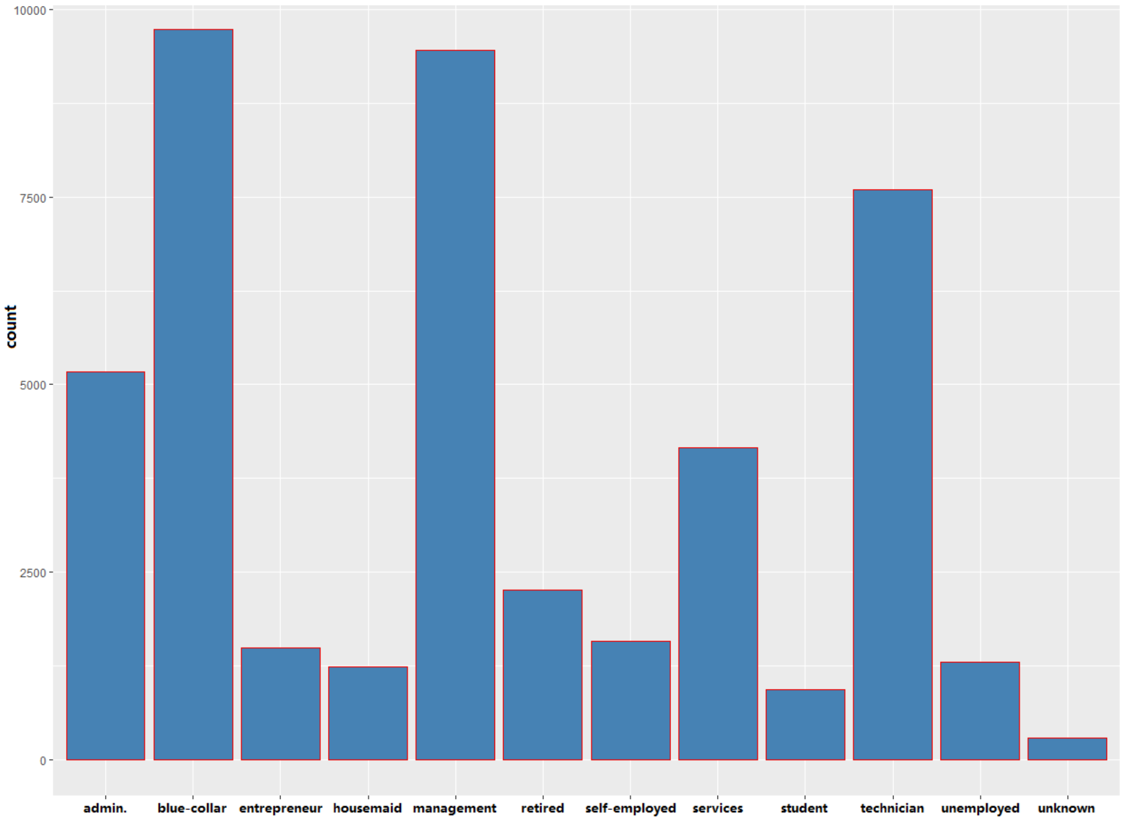


Figure 14: Occupational groups from which customers originate – bar graph

In order to check the relationship between the average annual deposit amount of customers and the credit default status we used the violin charts. It can be seen that for customers who have a credit default their average annual deposits are concentrated on the right side of 0, which is significantly lower than that of customers who do not default on credit.

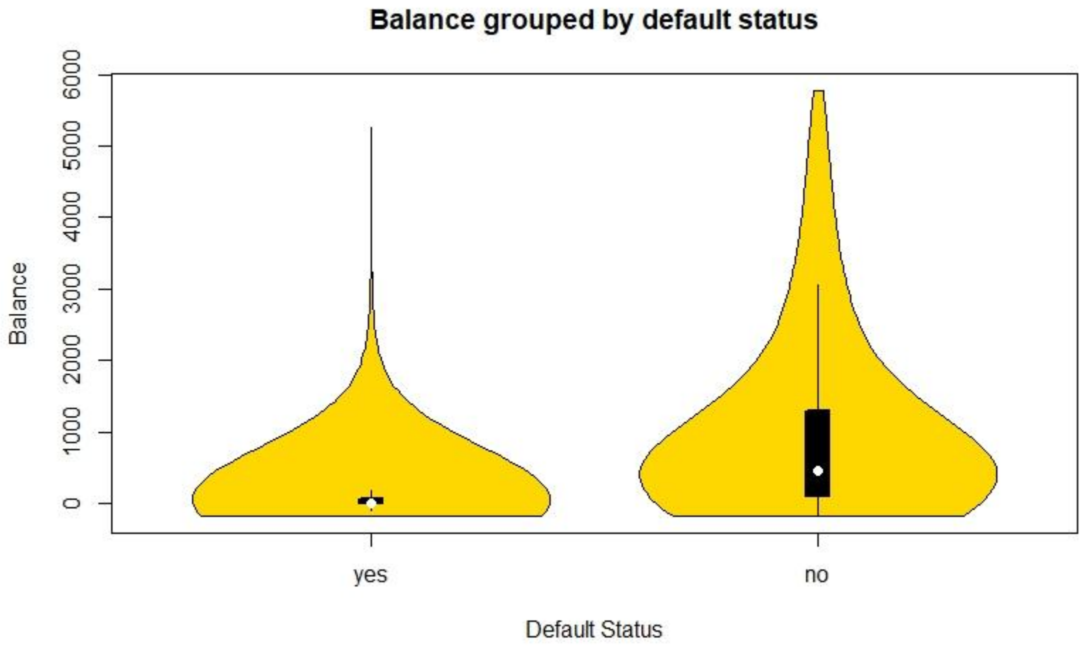


Figure 15: Distribution of average annual customer deposits by credit default status

3.2. Data Pre-processing

Multiple categorical variables which are out of order converted into a one-hot coded multi-dimensional mutual exclusion features using dummies package while if variables are in order they are directly given to each class in hierarchical order.

For complex variables (including both qualitative and quantitative records) first separate qualitative and quantitative records and then use the combination method of continuous variable discretization and stratification + one-hot encoding to process it. Consider pdays variable in the original data as an example, although it is a numeric type it contains a large number of records with a value of -1 (here -1 is not a missing value and it represents a specific type of situation and is a qualitative description). After data pre-processing the new data contains 45211 samples and 41 variables. Given the increased number of variables lead to need for feature selection algorithms.

Feature selection algorithms are used to extract the most important features that can improve the time and space complexity of the classifier algorithm resulting in improved prediction performance of the classifier.
Recursive Feature Elimination (RFE), Boruta and XGBoost algorithm are adopted to select important features.

- Recursive feature elimination (RFE) is a greedy algorithm for finding the optimal feature subset. We use the caret package to implement this method and select 10 important features as shown below.

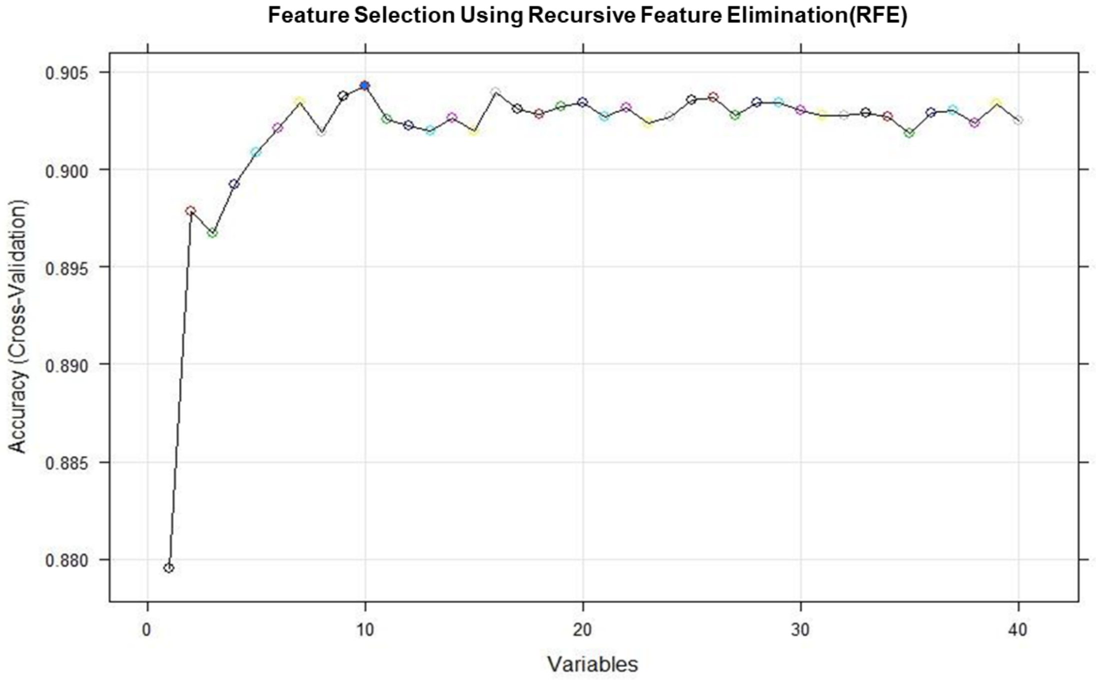


Figure 16: Feature selection using Recursive feature elimination (RFE)

- Boruta algorithm is based on random forest. This method can process each iteration recursively. The features that perform poorly in the process minimize the error of the model, and finally form minimized optimal features. Using Boruta package we have selected 31 important features as shown below.

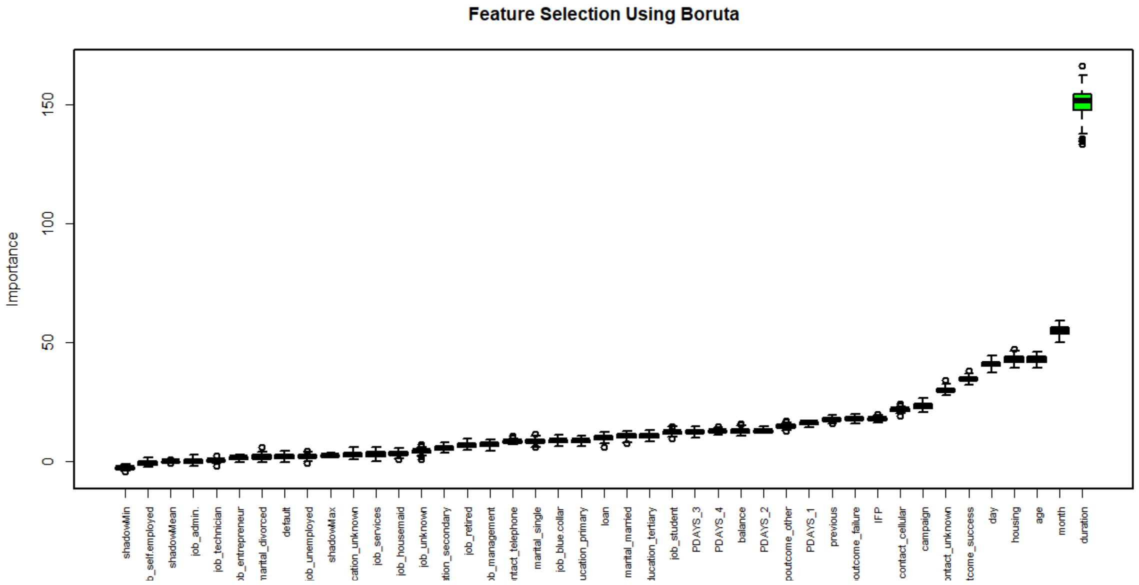


Figure 17: Variable importance under the Boruta feature selection method

- XGBoost is an implementation of Gradient Boosting that can complete distributed computing. Using XGBoost for feature selection is much faster than the RFE and Boruta methods. Using *xgboost* package after proper parameter adjustments we have selected 23 important features as shown below.

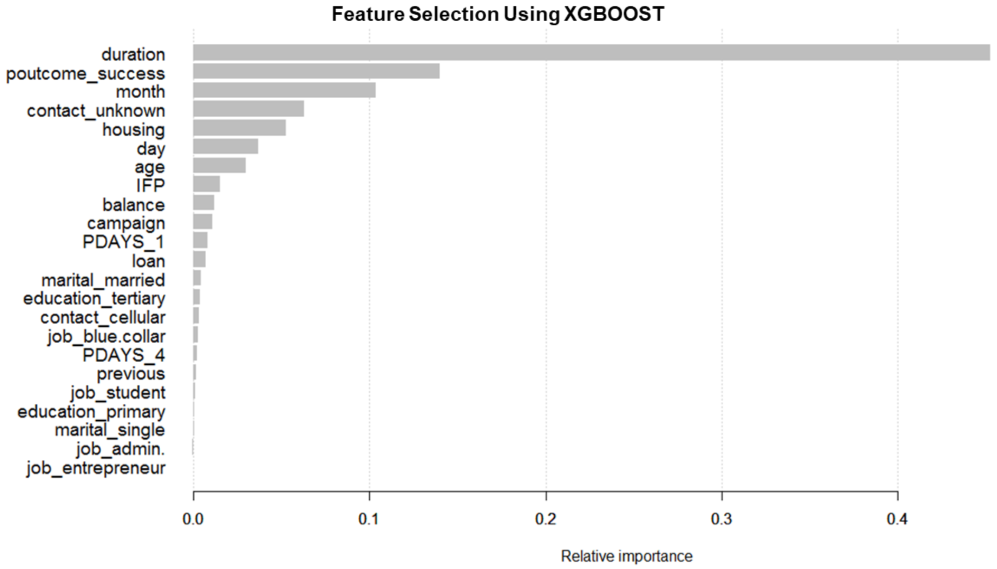


Figure 18: Relative importance of variables under the XGBoost feature selection method

Finally, based on the above results of three feature selection methods we were able to identify 21 important attributes out of 41 attributes which are age, day, month, duration, balance, housing, loan, campaign, education_primary, education_tertiary, marital_married, marital_single, job_blue.collar, job_student, contact_cellular, contact_unknown, IFP, previous, PDAYS_1, PDAYS_4, poutcome_success.

After identifying important features, the observations have been split into 3 folds (for 3-fold CV) with the number and distribution of instances in each fold presented below.
Table 4 Distribution of instances in the folds

	Fold1	*Fold2*	*Fold3*
total	15070	15071	15070
negative	13307	13308	13307
positive	1763	1763	1763

3.3. Joint Sampling Techniques

Given below the experimental process for handling the class imbalance using joint sampling techniques. We established index called F-Score for binary classification. After data pre-processing along with feature selection methods: RFE, Boruta and XGBoost. Considering the pros and cons of existing under sampling and oversampling methods, a joint combination of SMOTE, NCL, Tomek Link Removal, K-means sampling methods to process the original data set. Their results are input for training classifiers such as Random Forest, Logistic regression, LDA, SVM etc after which the model is applied on test data creating a confusion matrix resulting in performance metrics.

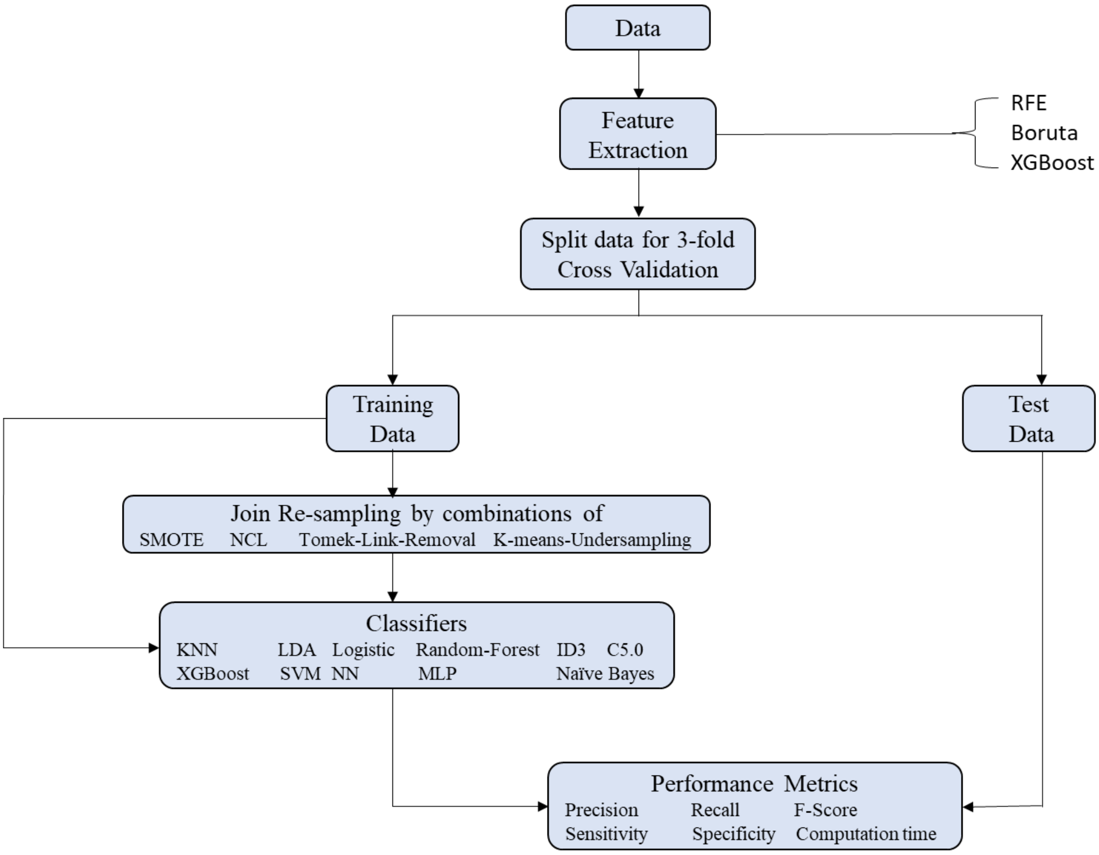

Data
Feature Extraction
RFE
Boruta
XGBoost
Split data for 3-fold Cross Validation
Training Data
Test Data
Join Re-sampling by combinations of
SMOTE NCL Tomek-Link-Removal K-means-Undersampling
Classifiers
KNN LDA Logistic Random-Forest ID3 C5.0
XGBoost SVM NN MLP Naïve Bayes
Performance Metrics
Precision Recall F-Score
Sensitivity Specificity Computation time

Figure 19: Flow chart of Joint sampling techniques

Among the data resampling methods, the most widely used algorithm is the SMOTE algorithm, but the samples generated by SMOTE randomly may be close to the majority samples, and the classification edge of the classifier. This would cause interference.

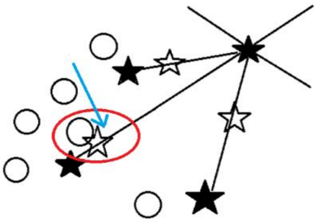


Figure 20: Defects of the SMOTE method

The above randomly generated minority samples have interference samples. Because they are close to the majority, they will be recognized by NCL Tomek Link Removal and can be removed. At the same time K-means clustering under-sampling can also remove the above-mentioned interference samples by sorting the distance within the class. Therefore, in the follow-up experiments, this paper adopts a joint sampling method combining SMOTE and NCL, Tomek Link Removal, and K-means clustering under-sampling. In short, the joint sampling method can take advantage of both under-sampling and over-sampling. While keeping the effective information of most classes as much as possible.

3.4. Alternative Optimal Probability Threshold

One major drawback of sampling techniques is that one needs to determine how much sampling to apply. An over-sampling level must be chosen so as to promote the minority class, while avoiding over fitting to the given data. Similarly, an under-sampling level must be chosen so as to retain as much information about the majority class as possible, while promoting a balanced class distribution. This could lead to information loss.

Another alternative is to increase the performance of minority class is to determine optimal probability thresholds when evaluating the model. By default, the classifiers use 50-50 class frequencies but we can change this by either maximizing the F-score on a PR curve or maximizing the sensitivity and specificity on a ROC curve. We choose one these approaches based on which performance metric to be maximized. With this approach, we can improve classifier performance without using data re-sampling techniques, thereby eliminating the risk of information loss. Given below flow chart represents the iterative approach used for finding the optimal probability threshold.

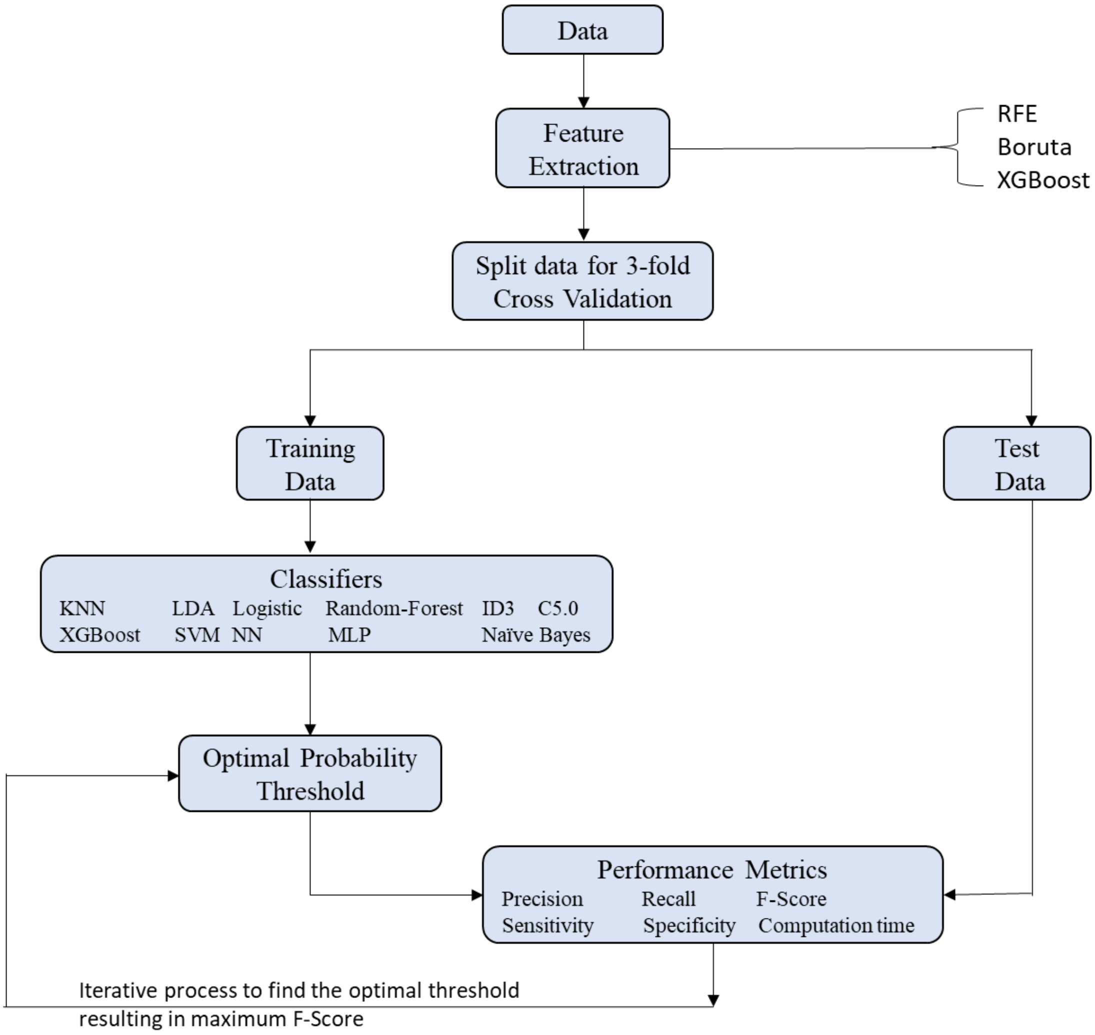
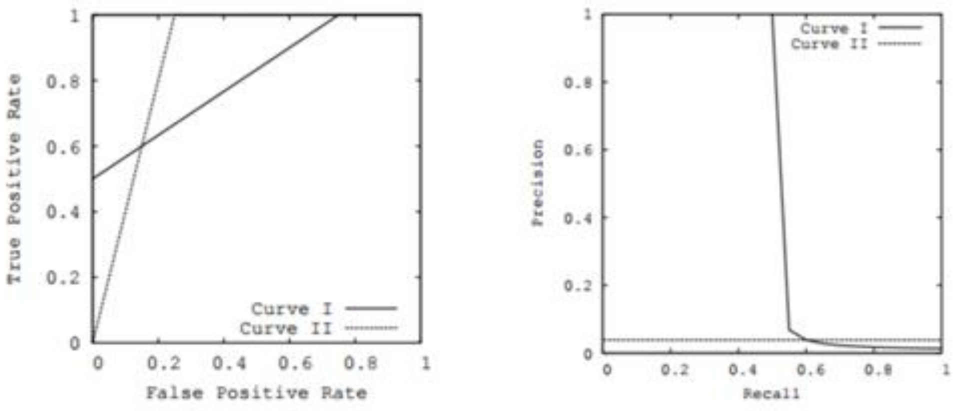


Figure 21: Flow chart for optimal calculation threshold.

The most straightforward approach is to find an optimal threshold using the ROC curve since it calculates the sensitivity and specificity across a continuum of cutoffs. Using this curve, an appropriate balance between sensitivity and specificity can be determined.

Although, using ROC curve is the standard approach, there are few challenges while using the AUC of ROC curve as a performance metric for classifiers on imbalanced data is a popular choice, it can be a misleading one if you are not careful. As mentioned in the following example from Davis and Goadrich (2006). Given below the model performance for the two classifiers on an Imbalanced dataset, with the ROC curve on the left and the precision-recall curve on the right. In the left chart, the AUC for Curve 1 is reported in this work as 0.813 while the AUC for Curve 2 is 0.875. So if were to choose the best AUC value we go with Model 2 as the best. However, the precision-recall curve on the right has area under Curve 1 is 0.513 while Curve 2 it is 0.038. As Curve 1 is having better early retrieval compared to Curve 2, we see this massive discrepancy in the precision and recall performance between the two classifiers.



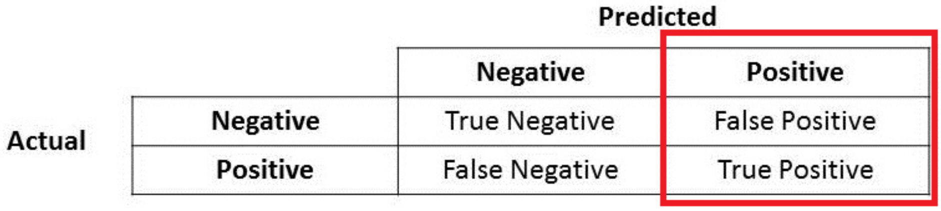
(a) Comparing AUC-ROC for two algorithms

(b) Comparing AUC-PR for two algorithms

Figure 22: Difference in optimizing area under the curve in each space.

Also, an important question while evaluating the performance of a classifier is which of the metrics is the most appropriate for Imbalanced data. Although, we can use accuracy but for special cases like Imbalanced classes we need to be careful while choosing a metric. In this project, we have employed F-score for evaluating the performance of the classifiers.

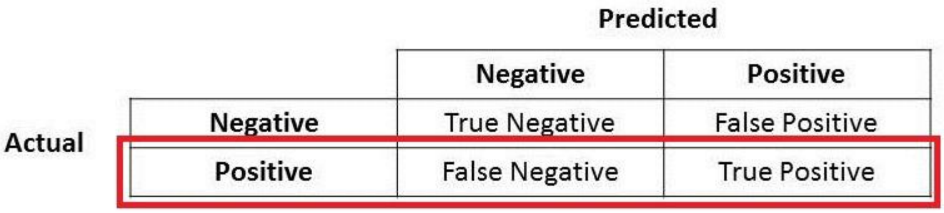
$$\text{Precision} = \frac{True\ positive}{True\ positive\ +False\ positive} == \frac{True\ positive}{Total\ Predicted\ Positives}$$



		Predicted	
		Negative	**Positive**
Actual	**Negative**	True Negative	False Positive
	Positive	False Negative	True Positive

The denominator gives Total predicted rate. So the precision here gives the Total positives out of Total predicted positives.

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad == \quad \frac{True\ Positive}{Total\ Actual\ Positives}$$



		Predicted	
		Negative	**Positive**
Actual	**Negative**	True Negative	False Positive
	Positive	False Negative	True Positive

The denominator gives Total actual rate. So the recall here gives the Total positives out of Total actual positives.

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

Now, F_β is a function of both precision and recall values. Precision can be taken from false positive and True positive where as Recall takes from false negative and True Positive. The goal of F-score is to reduce the values to false positives and false negatives thereby improving True positive value. F-score helps to maximize True positive values and hence improving the accuracy of the classifier. So for Imbalanced data, we are trying to concentrate on the positive class, hence F-score metric is justified.

4. Results and Discussion

4.1 Joint Sampling Techniques

A preliminary study on the classifier without applying re-sampling methods is evaluated using the comprehensive index F-Score and perform a preliminary evaluation of each classifier using three-fold cross-validation.

Table 4: Preliminary performance of the classifiers

Classifier	Precision	Recall	F-Score	Computation time (in seconds)
LDA	61%	38%	43%	2.8
LR	65%	33%	38%	1.9
ID3	63%	36%	41%	10.7
C5.0	60%	47%	50%	8.7
Random Forest	66%	35%	41%	3.7
KNN	65%	22%	28%	4.2
Naïve Bayes	41%	45%	43%	44.9
MLP feedforwdNetworks	61%	40%	45%	47.0
Neural Networks BP	61%	40%	45%	2.1
SVM	32%	76%	53%	32.8
XGBoost	67%	37%	43%	5.7
Average	58%	41%	43%	16.2

As can be seen from the above table 4, the unbalanced characteristics of the data lead to generally poor classification results. The average precision rate Precision is 58%, while the recall rate that is the most concerned in unbalanced scenarios is only 41%. The computation time for SVM, Naive Bayes, MLP Feed forward networks are high.

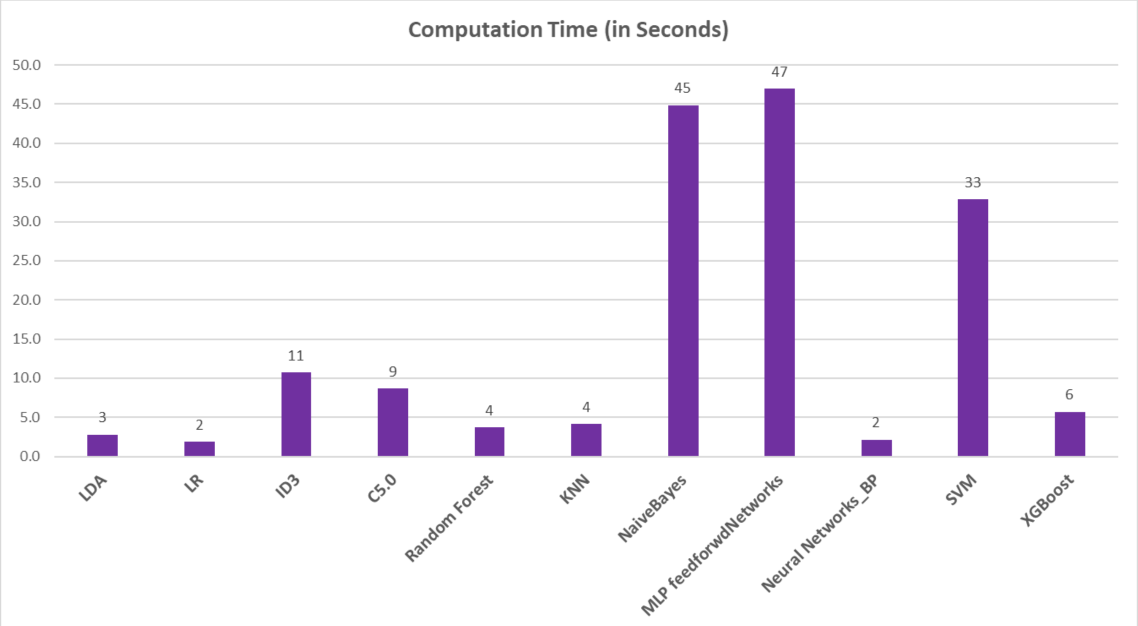


Figure 23: Computation time of Imbalanced algorithms

Then we used SMOTE along with five joint sampling methods (SNT, SNK, STK, SK, SNTK) implemented for multiple classifiers.

Table 5: Performance of the classifiers with joint sampling methods.

Classifier \ Sampling methods	None	SMOTE	SK	STK	SNK	SNT	SNTK
LDA	43%	53%	60%	60%	60%	59%	60%
LR	38%	56%	60%	60%	60%	60%	59%
ID3	41%	56%	58%	55%	55%	54%	53%
C5.0	50%	**62%**	**64%**	**63%**	63%	**64%**	63%
Random Forest	41%	59%	**66%**	**66%**	**65%**	**65%**	**64%**
KNN	28%	54%	56%	56%	56%	59%	56%
Naive Bayes	43%	47%	49%	52%	51%	53%	53%
MLP feed forward Networks	45%	**61%**	51%	60%	49%	58%	50%
Neural Networks BP	45%	59%	60%	59%	59%	59%	58%
SVM	53%	53%	52%	52%	52%	53%	52%
XGBoost	43%	59%	**66%**	**64%**	**64%**	**64%**	**63%**

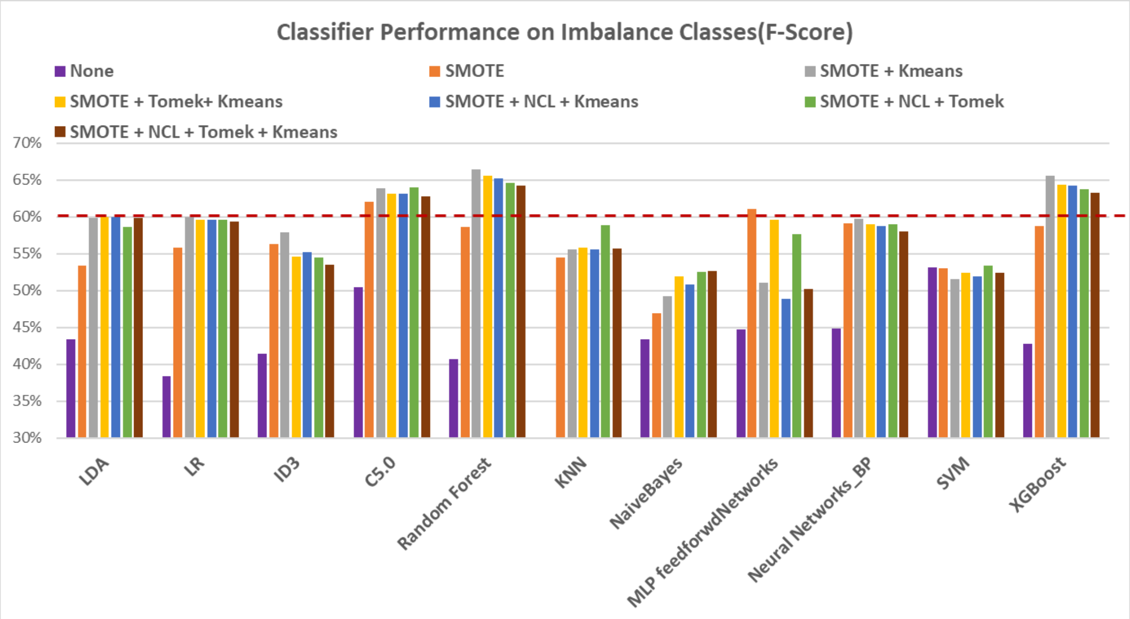


Figure 24: Classifier performance F-Score (based on various sampling methods)

From the results, combined with actual experience, we observed classifiers performance improved significantly using sampling methods such as SMOTE. Further we were able to improve classifiers performance by applying five joint sampling methods discussed earlier surpassing the SMOTE method. Given below computation time for different combinations of re-sampling techniques and classifiers. XGBoost and Random forest classifiers seems to have better F-score with lower computation time and were comparable with other good performing classifiers.

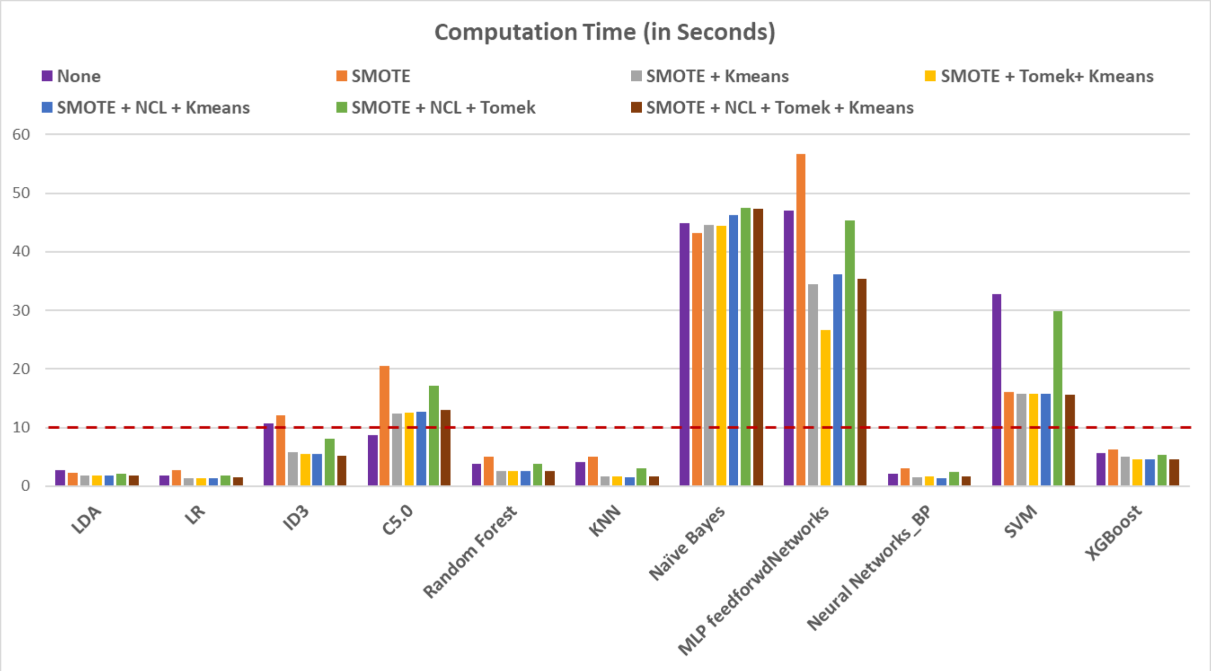


Figure 25: Classifier Computation time (based on various sampling methods)

4.2 Alternative Optimal Probability Threshold

Figure 26, shows, ROC curve for the random forest model based on the test dataset. Several threshold cutoffs are shown on the curve and decreasing the cutoff for the probability of results in increased sensitivity at the expense of the specificity. There may be scenarios where we can find a trade-off between sensitivity and specificity without severely impacting the accuracy of the majority class. The plot shows the default probability cut off value of 50%. The sensitivity and specificity values associated with this point indicate that performance is not optimal.

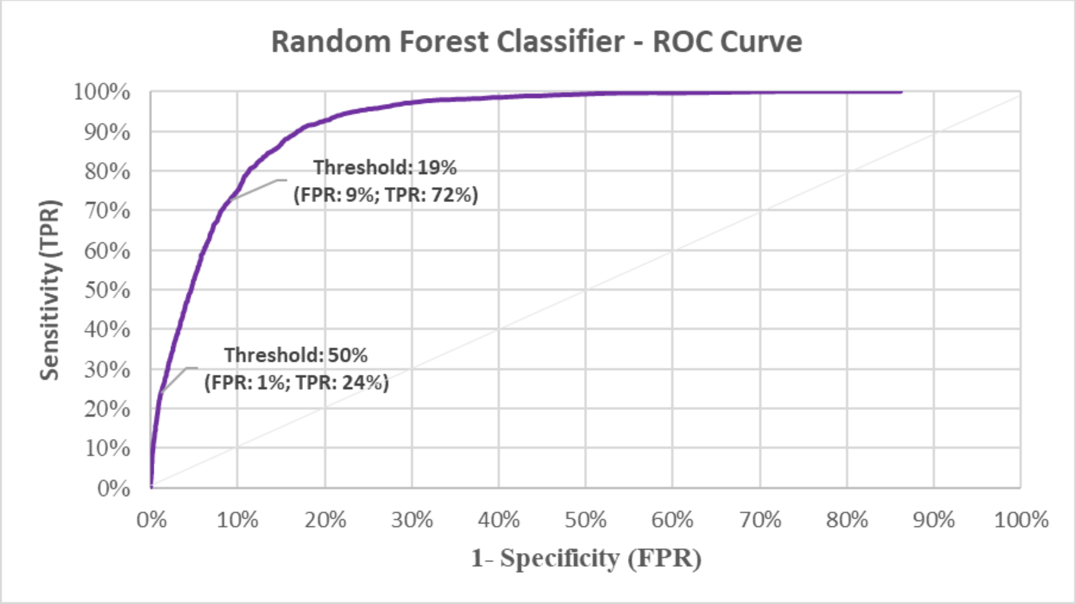


Figure 26: The random forest ROC curve

Several techniques exist for determining a new cutoff. First, if there is a particular target that must be met for the sensitivity or specificity, this point can be found on the ROC Curve and the corresponding cutoff can be determined. Another approach is to find the point on the ROC curve that is closest (i.e., the shortest distance) to the perfect model (with 100% sensitivity and 100% specificity), which is associated with the upper left corner of the plot.

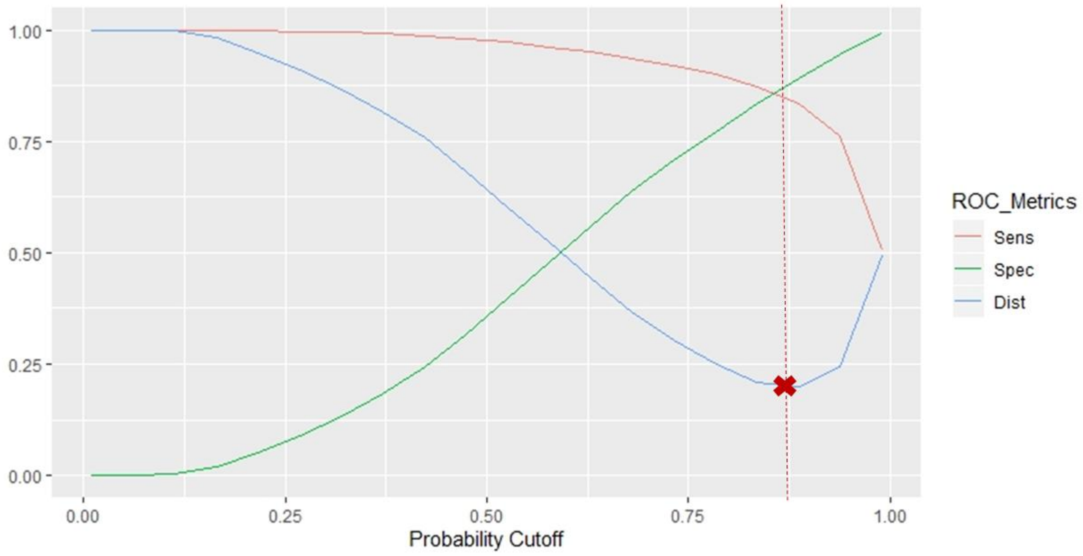


Figure 27: ROC curve that is closest to the perfect model

Optimizing threshold using PR curve, we aim to the find optimal recall and precision value in order to maximize F score of the classifier. The harmonic mean of precision and recall, the F measure is widely used to evaluate the success of a binary classifier when there is a minority class.

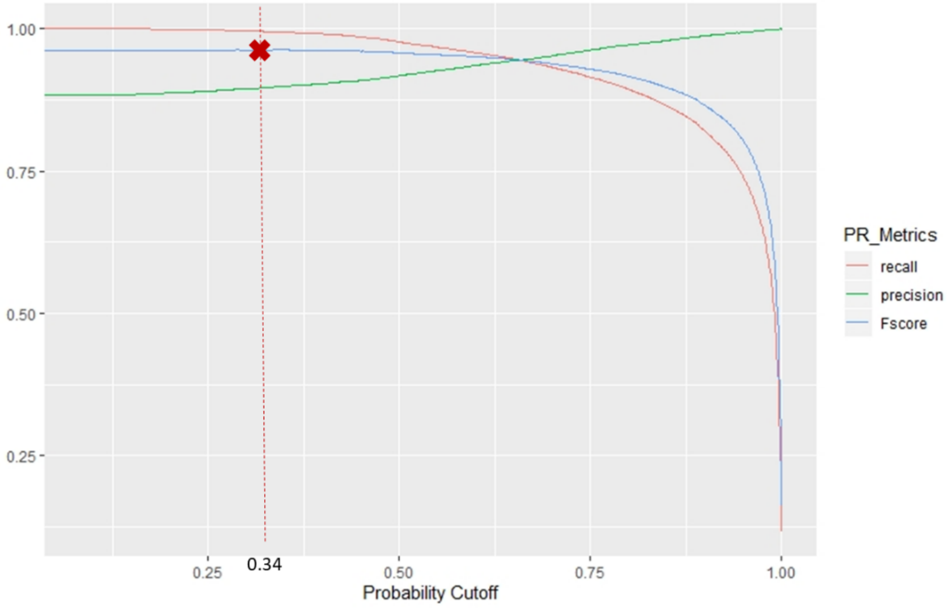


Figure 28: PR Curve maximizing F score.

We have applied PR curve based approach of maximizing F1 score across various classifiers discussed in this thesis and calculated optimal thresholds which are unique for a given classifier. We have summarized the results comparing F1 scores below.

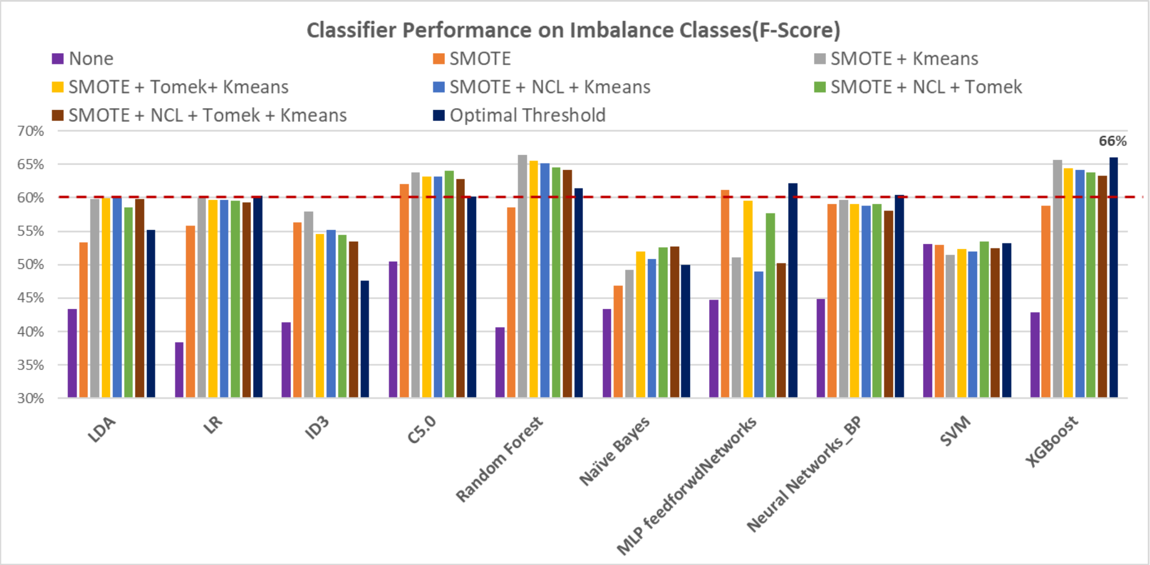


Figure 29: Classifier performance F-score (with optimal threshold)

As shown in Figure 29, classifiers performance (F-score) has significantly improved using optimal threshold than classifier with default threshold of 50%. Using optimal threshold approach, we were able to achieve an F-score of 66% using XGBoost classifier. We also observed some classifiers such as random forest where F-score using optimal threshold approach is slightly lower than joint sampling methods. In such instances, we can trade-off with elimination information loss seen in re-sampling methods. Computation time increased by 1-3% using this approach, as we need to calculate the optimal threshold iteratively (vs base classifier).

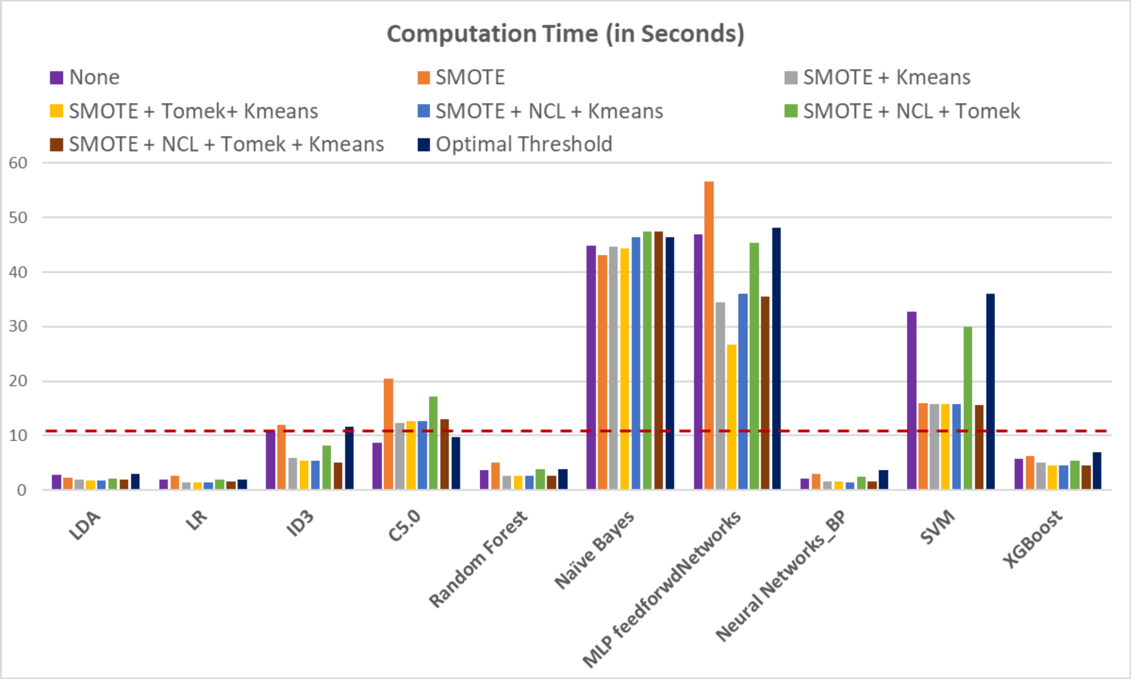


Figure 30: Classifier Computation time (with optimal threshold)

5 .Conclusion

In the thesis, we have explored 11 mainstream classifiers with average precision of 58% while recall has only 41% resulting in F-score of 43%. Using data re-sampling methods in combination with classifiers, we were able to increase recall up to 84% and resulting F-score of 66%. From the comparison of results using joint sampling methods, we observed XGBoost and Random forest classifiers seems to have better F-score with lower computation time and were comparable with other good performing classifiers. Since we realized the risk of information loss in data re-sampling methods, we explored another alternative of changing the probability cut-off. By default, most classifiers have 50-50 cut-off. Using PR curve by maximizing the F-score metric we were able to estimate the optimal threshold cut-off for a given classifier. Using optimal threshold approach, we observed XGBoost has highest F-score of 66% with lower computation time. We also observed some classifiers such as random forest where F-score using optimal threshold approach is slightly lower than joint sampling methods, we can trade-off with elimination information loss seen in re-sampling methods in such instances.

Bibliography

1. V. García, J.S. Sánchez, R.A. Mollineda, On the effectiveness of preprocessing methods when dealing with different levels of class imbalance, Knowledge Based Systems 25 (1) (2012) 13–21.

2. Wright, R. E. (1995). *Logistic regression.* In L. G. Grimm & P. R. Yarnold (Eds.), *Reading and understanding multivariate statistics* (p. 217–244). American Psychological Association

3. (2009) LDA (Linear Discriminant Analysis). In: Li S.Z., Jain A. (eds) Encyclopedia of Biometrics. Springer, Boston, MA.

4. Mucherino A., Papajorgji P.J., Pardalos P.M. (2009) *k*-Nearest Neighbor Classification. In: Data Mining in Agriculture. Springer Optimization and Its Applications, vol 34. Springer, New York, NY.

5. Webb G.I. (2011) Naive Bayes. In: Sammut C., Webb G.I. (eds) Encyclopedia of Machine Learning. Springer, Boston, MA.

6. Schmidhuber, J. (2015). "Deep Learning in Neural Networks: An Overview". Neural Networks. 61: 85–117.
7. Cortes, Corinna; Vapnik, Vladimir N. (1995). "Support-vector networks" (PDF). Machine Learning. 20 (3): 273–297.
8. Quinlan, J. Induction of Decision Trees. *Mach Learn* **1,** 81–106 (1986).
9. Quinlan, J. R. 1986. Induction of Decision Trees. Mach. Learn. 1, 1 (Mar. 1986), 81–106
10. Quinlan R (1993). C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers
11. M. Kuhn and K. Johnson, Applied Predictive Modeling, Springer 2013
12. A. Liaw and M. Wiener (2002). Classification and Regression by randomForest. R News 2(3), 18—22
13. Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16). Association for Computing Machinery, New York, NY, USA, 785–794.
14. K. Agustianto and P. Destarianto, "Imbalance Data Handling using Neighborhood Cleaning Rule (NCL) Sampling Method for Precision Student Modeling," 2019 International Conference on Computer Science, Information Technology, and Electrical Engineering (ICOMITEE), Jember, Indonesia, 2019, pp. 86-89, doi: 10.1109/ICOMITEE.2019.8921159.
15. Elhassan, Tusneem & M, Aljourf & F, Al-Mohanna & Shoukri, Mohamed. (2016). Classification of Imbalance Data using Tomek Link (T-Link) Combined with Random Under-sampling (RUS) as a Data Reduction Method. Global Journal of Technology and Optimization. 01. 10.4172/2229-8711.S1111.

16. Xiong, H., Wu, J.J., Chen, J.: K-means clustering versus validation measures: a data-distribution perspective. IEEE Trans. Syst. Man Cybern. B Cybern. **39**(2), 318–331 (2009)
17. Alejandro Moreo, Andrea Esuli, and Fabrizio Sebastiani. 2016. Distributional Random Oversampling for Imbalanced Text Classification. In Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval (SIGIR '16). Association for Computing Machinery, New York, NY, USA, 805–808.
18. Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. 2002. SMOTE: synthetic minority over-sampling technique. J. Artif. Int. Res. 16, 1 (January 2002), 321–357.
19. Gu, Q., L. Zhu and Zhihua Cai. “Evaluation Measures of the Classification Performance of Imbalanced Data Sets.” (2009).
20. Sasaki, Y. (2007). "The truth of the F-measure"