

**INSURANCE FRAUD DETECTION USING EXTREME GRADIENT
BOOSTING AND RANDOM FOREST ALGORITHMS**

by

Judith Goodness Khanyisa Mabunda

DISSERTATION

submitted in fulfilment of the requirements for the degree of

MASTER OF SCIENCE

in

e-SCIENCE

in the

**DEPARTMENT OF STATISTICS AND OPERATIONS RESEARCH,
FACULTY OF SCIENCE AND AGRICULTURE**

(School of Mathematical and Computer Sciences)

at the

UNIVERSITY OF LIMPOPO


SOUTH AFRICA.

Supervisor: DR. TI MODIPA

20 February 2023

Declaration

I, Judith Goodness Khanyisa Mabunda, declare that this research work is my own, unaided work. It is being submitted for the degree of MASTER OF SCIENCE in e-SCIENCE at the UNIVERSITY OF LIMPOPO, SOUTH AFRICA. It has not been submitted for any degree or examination at any other university. All materials used from other sources have been referenced to show that this material has been adopted to support my work.



Judith Goodness Khanyisa Mabunda

20 February 2023

Abstract

The rising amount of fraud in claims has been of great concern to the insurance companies. In this research work, we developed two machine learning models namely, Extreme Gradient Boosting (XGBoost) and Random Forest for the purpose of insurance fraud detection based on auto insurance claims data. The models detect fraudulent claims and classify them into fraudulent or non-fraudulent. Different data pre-processing techniques are used to clean, explore, and extract relevant features. The effectiveness of the algorithms are observed using performance evaluation metrics: precision, recall and f1 score and confusion matrix. We also introduced the Synthetic Minority Oversampling (SMOTE) and Random Oversampling (ROS) data augmentation techniques to handle the imbalanced data and compare the results of the models before and after the data is balanced. The comparative results of classification algorithms conclude that the XGBoost model is effective in fraud detection than the Random Forest model on imbalanced data. In addition to this, the Random Forest model was effective in predicting fraudulent claims when the data augmentation techniques were applied.

Acknowledgements

I would like to express my acknowledgement and give my very great appreciation to Dr. TI Modipa, my research work supervisor who made the completion of this research work possible. His guidance and advice carried me throughout this work. I wish to thank the funding towards my Master of Science. I am also thankful to my family for their support and encouragement while working on the research. Finally, I would like to thank God for being by my side throughout this research work.

Contents

Declaration	i
Abstract	ii
Acknowledgements	iii
List of Figures	vi
List of Tables	vii
1 Introduction	1
1.1 Problem Statement	1
1.2 Aim	2
1.3 Objectives	2
1.4 Research Questions	2
1.5 Significance of the Study	3
1.6 Chapter Outline	3
2 Literature Review	4
2.1 Introduction	4
2.2 Data and Models	4
2.2.1 XGBoost	8
2.2.2 Random Forest	9
2.2.3 Comparison of XGBoost and Random Forest	11
2.3 Data Augmentation	11
2.3.1 SMOTE	13
2.3.2 ROS	15
2.4 Conclusion	15

3	Research Methodology	16
3.1	Data	16
3.2	Exploratory Data Analysis	17
3.3	Data Pre-processing	23
3.3.1	Missing Values	23
3.3.2	Categorical Values	24
3.3.3	Feature Selection	24
3.3.4	Training/Validation/Testing Split	27
3.3.5	Data Augmentation	27
3.4	Models	29
3.4.1	Prediction	29
3.4.2	XGBoost	30
3.4.3	Random Forest	31
3.5	Performance Evaluation of Models	32
3.6	Conclusion	34
4	Results and Discussion	35
4.1	Model Results	35
4.1.1	XGBoost: Before data augmentation	35
4.1.2	XGBoost: SMOTE	37
4.1.3	XGBoost: ROS	39
4.2	Random Forest	42
4.2.1	Random Forest: Before data augmentation	42
4.2.2	Random Forest: SMOTE	44
4.2.3	Random Forest: ROS	45
4.3	Discussion	49
4.4	Conclusion	49
5	Conclusion	51
5.1	Introduction	51
5.2	Summary of the Study	51
5.3	Summary of Findings	52
5.4	Limitations of the Study	52
5.5	Recommendations for Future Studies	53
5.6	Conclusion	53

List of Figures

2.1	A general structure of a Random Forest Model [29]	11
2.2	Illustration of the oversampling techniques [34]	13
2.3	An illustration of the SMOTE technique used to generate synthetic samples [36]	14
3.1	Distribution of frauds based on the target variable	19
3.2	Averages for certain claims by Fraud Reported	20
3.3	Confirmed reported fraud by different types of incidents	22
3.4	Correlation heatmap representing the correlation between different variables. The correlation values can be any value from -1 to 1	26
3.5	Splitting Dataset into Training, Validation and Testing Sets	27
3.6	Creation of a model and using it for prediction [57]	30
4.1	PR curves and Confusion Matrices for the XGBoost model before data augmentation	37
4.2	PR curves and Confusion Matrices for the XGBoost model using SMOTE	39
4.3	PR curves and Confusion Matrices for the XGBoost model using ROS	41
4.4	PR curves and Confusion Matrices for the Random Forest model before data augmentation	43
4.5	PR curves and Confusion Matrices for the Random Forest model using SMOTE	46
4.6	PR curves and Confusion Matrices for the Random Forest model using ROS	48

List of Tables

2.1	Differences between XGBoost and Random Forest comparison table [30]	12
3.1	The insurance dataset elements	17
3.2	Number of fraudulent and non-fraudulent claims for each set	28
3.3	Number of fraudulent and non-fraudulent claims for each set when SMOTE and ROS techniques are used	29
3.4	A confusion matrix to show the performance of a classification model where actual values for all the sets are known.	33
4.1	Evaluation metrics of XGBoost on validation and testing sets without data augmentation	36
4.2	Evaluation metrics of XGBoost on validation and testing sets using SMOTE	38
4.3	Evaluation metrics of XGBoost on validation and testing sets using ROS	40
4.4	Evaluation metrics of Random Forest on validation and testing sets before data augmentation	44
4.5	Evaluation metrics of Random Forest on validation and testing sets using SMOTE	45
4.6	Evaluation metrics of Random Forest on validation and testing sets using ROS	47

Chapter 1

Introduction

1.1 Problem Statement

Over the past few years, insurance fraud has been steadily increasing. Although legislation and insurance companies have launched anti-fraud activities, insurance fraud is still a serious problem and shows no sign of slowing down [1]. Frauds in insurance claims have resulted in huge losses to insurance companies in terms of reduced revenue, insufficient reserves, increased insurance costs such as loss costs, and difficulty in pricing [1].

Traditionally, businesses relied on rules alone to block fraudulent activities which required a case by case study to detect fraud. This approach slows down the system and it imposes a heavy maintenance burden on fraud analysts. As fraud develops, more and more manual reviews are required [2].

Machine learning models have been used in previously conducted fraud detection studies in efforts to enhance traditional systems to fight against fraud. The models include, but are not limited to, Random Forest [3], Logistic regression [4], K-Nearest Neighbor [5], Naïve Bayes, Decision tree [3], [6] and Support Vector Machines [7]. These studies show that some models perform well than others due to the kind of data used. Research done from other sources identifies shortfalls that these models are prone to [8]; [5]; [9]. These models are prone to overfitting. Some are not able to handle missing data. They do not perform well in case of imbalanced class distribution. Some works well on large datasets and are found to be biased while working with categorical variables.

The purpose of this research work is to evaluate two supervised machine learning algorithms namely, Random Forest and Extreme Gradient Boosting (XGBoost), for the detection of fraud in auto insurance claims.

1.2 Aim

The aim of this research work is to develop two machine learning-based models using insurance claims data to detect fraudulent activities.

1.3 Objectives

The objectives of this research work are to:

- i. Pre-process the auto insurance claims data.
- ii. Perform exploratory data analysis to develop an understanding of the existing auto insurance claims data.
- iii. Train the XGBoost and Random Forest models.
- iv. Evaluate the performance of the two machine learning models.

1.4 Research Questions

The research questions set out for this research work to answer are:

- i. How to perform data pre-processing to transform the insurance claims data into a format that can be effectively analysed by machine learning models?
- ii. How to perform exploratory data analysis on insurance claims data to gain a better understanding of the data?
- iii. How to develop the XGBoost and Random Forest models for insurance fraud detection?
- iv. What are the performance evaluation metrics we employ to evaluate the performance of the two models in detecting insurance fraud?

1.5 Significance of the Study

The research work has evaluated the performances of two machine learning models in order to determine the effectiveness of detecting fraudulent activities. It is clear that no specific technique is applicable across the board. There are factors that affect the performances of these approaches. Such factors include the type of data as well as the amount used. Furthermore, the challenges of data imbalances may need to be taken into account. Classification tasks are affected by the data imbalances. However, this study has shown the effect of different techniques to address data imbalances and how they affect the performances of the classification algorithms under consideration.

1.6 Chapter Outline

The dissertation is structured as follows:

- In Chapter 2 we provide the related work that highlights work done by other authors.
- Chapter 3 discusses the proposed methodology which comprises of data, exploratory data analysis, data pre-processing, model building, and performance evaluation of models.
- Chapter 4 presents the results and discussion of the findings.
- Chapter 5 concludes the research work by presenting a summary of the overall research, stating the limitations of the research and provides future work and recommendations.

Chapter 2

Literature Review

2.1 Introduction

This chapter presents a discussion on insurance fraud detection. The first section discusses the fraud detection using different machine learning algorithms, which is followed by a description of how the models work and comparisons between the models. Following that is the discussion of studies on data augmentation techniques and the conclusion for the chapter.

2.2 Data and Models

Insurance fraud has attracted much research interest, as it is an entrenched problem in most countries and requires both proactive responses and preventive measures [10]. Insurance fraud is defined in an Insurance Information Institute article on Insurance Fraud Background as an act of seeking insurance payments by misleading the insurer as to the occurrence, cause, or details of an insured event [11]. Different studies apply fraud detection as it is a breakthrough that may solve fraud problems across insurance companies.

Itri and Mohamed [12] compared ten machine learning algorithms, amongst them is the Random Forest model. The models' performance was compared using two evaluation metrics namely, F-Score and K-Score to determine the most suitable model for prediction in real world. K-Score is the overall of F-Score and break even point. They introduced the Root Mean Squared Log error (RMSE) indicator to compute the

square of the difference between each point's prediction and its target, then add the root of those average values. When this value is higher, the model's performance is ineffective. The relevance of the results of the two scores mentioned are judged with the RMSE indicator. According to K-Score, Random Forest has the highest ranking. Unlike F-Score, Random Forest is ranked last. When compared with the RSME indicator, there is a significant rank correlation between K-Score and RMSE, with Random Forest having the lowest value.

Li and Yan [13], developed a Random Forest detection model using real world data of an automobile insurance company. They performed feature selection and obtained the importance of each input variable to the output variable. The misclassification of the model was analyzed. The results show that the Random Forest is effective for large data sets and imbalanced data compared with the traditional model which involves extensive use of auditing, where reports or transactions are manually observed to identify fraudulent behaviour patterns. This method is time consuming and not suitable when it comes to big data [14]. Only the Random Forest model was used for the detection of fraud on automobile insurance claims. In this research work we develop and compare the Random Forest and XGBoost models.

Roy and George [3] used machine learning models like Random Forest, Decision tree (DT) and Naïve Bayes (NB) to detect fraud in auto insurance claims. The machine learning techniques were trained on auto insurance claim data from the United States. The models were evaluated and compared using the confusion matrix. According to the findings of the experiment, the Random Forest algorithm performed better in detecting insurance claims fraud than the DT and NB algorithms. The study did not compare the models based on accuracy, precision, and recall but the confusion matrix only. In this research work we encompass not only confusion matrix, but also the precision and recall to evaluate and compare the models' performance.

Sumalatha and Prabha [4], proposed a predictive analytics platform able to check for suspicious fraud in Medical insurance claims. Mediclaim Insurance data was used. The data includes hospital information, patient's information as well as insurance claims and past insurance claims data. The study used Logistic regression for modelling. The data has been collected from medical sectors and insurance companies. Fraud

was predicted using the multi criteria decision support system. Logistic regression and multi criteria decision analysis improved the results of the proposed system. The study focused on medical insurance claims data. In this research work we use auto insurance claims data.

Faseela and Thangam [15] reviewed the Hidden Markov models and Non Negative Matrix Factorisation approaches to detect health insurance claim fraud. Hidden Markov Models deal with variables that are hidden and can only be observed from other observations. Hidden Markov models deal with hidden variables that rely on inference from other observations rather than direct observation. The Non Negative Matrix Factorisation decomposes data as a matrix into the product of two non-negative matrices. The Hidden Markov model was implemented by decomposing the dataset into groups of claimants with same age since the age is important in the medical condition of a patient. The Non Negative Matrix Factorisation grouped medical treatment items such as medicines or medical measurements according to the usage of different patients. The amount of calculations that the system needs when Hidden Markov model is applied is high when the dataset is large. The findings recommend that a distributed Non Negative Matrix Factorisation be used for large datasets.

Malini and Pushpa [5] used the K-Nearest Neighbour model and outlier detection to identify credit card fraud. They carried out a literature review to compare methods applicable in identifying credit card fraud. K-NN was used because of its interpretation and low calculation time. The outlier detection worked effectively on online large datasets. Compared with other detection methods, the K-NN model is more accurate and consistent in detecting credit card fraud. The methods are efficient in increasing the fraud detection rate and minimize false alert rates.

Rawte and Anuradha [7], conducted a research in Fraud Detection in Health Insurance. Two techniques were used for different purposes. Data was grouped using the Evolving Clustering Method (ECM). The data was categorised using the Support Vector Machine (SVM). The ECM clustered the mediclaim insurance by looking at the type of diagnosis. Then the mediclaim data was later categorised to find the claims which are repeating. The ECM groups new insurance claims by changing the quantity and position of the group. After training, SVM detects a class of a new insurance claim.

The system only handles outliers and duplicate claims.

Dhieb and Ghazzai [16] developed a Smart Insurance System. The system is based on Blockchain and Artificial intelligence. Blockchain and XGBoost and Very Fast Decision Tree (VFDT) were used. The experimental work revealed that the proposed models yield effective results in predicting fraudulent claims and can identify different types insurance fraud. In addition, the XGBoost model proved that it is effective in predicting how customers will behave in the future and how much amount they will claim.

Majhi and Bhattacharya [17] proposed two systems, a hybrid fuzzy clustering technique and a novel hybrid Automobile Insurance Fraud Detection. The fuzzy clustering technique as an undersampling method and optimised cluster centroids. For the novel hybrid automobile fraud detection, undersampling was performed using the fuzzy clustering and removed outliers from the class with a higher distribution (majority class) than the other class (minority class). Three machine learning models namely, Random Forest, Logistic Regression and XGBoost were built based on the balanced dataset for the automobile fraud detection. Sensitivity, accuracy and specificity were used to evaluate how the models perform when predicting fraud. The fuzzy clustering technique along with the XGBoost model performed better than the other methods.

The literature review shows that some of the models used in literature are not able to detect fraud in real-time [5]. We want models that can detect fraud as it occurs to eliminate waste of resources in efforts to recover from fraud that could have been prevented from taking place.

Each of these models has significant drawbacks, including decreasing accuracy and inefficiency, sometimes predicting non-fraudulent claims as fraudulent claims and vice versa. Therefore, continuous studying of this field is important in order to fight insurance fraud in order to benefit insurance companies and customers. Based on these disadvantages, this study has also reviewed techniques used to generate enough fraudulent claims to fit to the machine learning models. It is clear from the review that the issue of data imbalance is serious, especially for the class that we are mostly interested in and are trying to detect.

This research work trained two models in order to determine the efficiency in fighting fraud to ensure that insurance companies do not suffer great loss and loyal customers are provided with satisfactory services. Insurance companies need strategies that are in line with the current fraud trends to fight against fraud, hence, this research work is conducted to further assist in this matter.

Based on the literature reviewed, Random Forest is chosen to be used in this research work together with XGBoost. Random Forest is chosen among the other models in literature because it performed better in most cases when detecting fraudulent claims. The Random Forest chooses features randomly. Random Forest works better because of the randomisation feature [18]. XGBoost is considered because it has been built and developed for the purpose of improving both classification and prediction model performance [19]; [20]; [21]. The two models are briefly explained in the sections that follow.

2.2.1 XGBoost

XGBoost is a machine learning algorithm that uses the Gradient Boosting framework to implement machine learning algorithms. XGBoost continuously trains models, training each new model to correct the mistakes made by the previous model. Models are added in a sequential manner until no further improvements can be made. It combines many models based on decision trees to generate the final best model. It is designed to be highly efficient, flexible and portable [22].

We need to understand decision tree and Gradient Boosting algorithms before the XGBoost. A Decision tree model is a tree-like structure, where each internal node represents a test on a variable, each branch represents the test result, and each leaf node (terminal node) contains a class label. A tree can be learned by dividing the original set into subsets based on a test of variable values. This process repeats itself for each subset generated. When the value of a subset at a node is the same as the value of the target variable, or when the split adds no more value to the predictions,

the process is complete [23].

Gradient Boosting is a well-known algorithm for boosting. Every predictor in gradient boosting minimizes the loss function of its predecessor. In contrast to Adaboost (also known as Adaptive Boosting, this is an Ensemble modeling technique used in machine learning to find the best model), the weights of the training instances are not changed; rather, Every predictor is trained using the residual errors of the predecessor as labels. In Gradient Boosting, there is a Gradient Boosted Tree technique where its base learner is CART (Classification and Regression Trees) [24].

The XGBoost algorithm generates decision trees sequentially. Weights are very important in XGBoost. All input variables are assigned with weights. The weights are then fit into the decision tree that predicts the outcomes. When variables are incorrectly predicted by the tree, their weight increases, and those variables are then fit into the second decision tree. These individual models are then combined to form a model that is accurate and has a strong ability to perform prediction [25].

2.2.2 Random Forest

Random Forest is a supervised classification model that trains many tree-composed classifiers on a number of sub-datasets of the original dataset [26]. Random Forest is trained through the bootstrap aggregation (bagging) method which entails selecting subsets of the training data at random, training a model on the subsets and aggregating the results. Each new data point in a Random Forest goes through the same procedure as before, but now it visits all of the trees in the ensemble. The trees are generated using samples that have been randomly selected for both training and features. The majority votes is used to aggregate the predictions.

We now explain how each tree is built and how randomness kicks in. Each node of an individual tree in original forests is connected to a cell that is hyperrectangular [27]. Each tree in a Random Forest model is made up of three parts: the root node, the decision node, and the terminal node. A root node is the node from which the population begins to divide. The nodes that result from splitting a root node are known as decision nodes, and the node that cannot be split further is known as a terminal

node. The tree's root is X . A node is split into two parts for each step in the construction of a tree. Together, the terminal nodes form an X partition. The algorithm generates M different trees randomly. An observation from the original dataset, with or without replacement, is randomly chosen from each tree before construction. In the tree building, observations with possible repetitions are considered. The split is then completed at each tree's cell level by maximizing the CART-criterion over $mtry$ ($mtry$ is a number of variables sampled at random as candidates at each split) directions selected randomly in a uniform manner from the p original ones [28].

The process of building individual trees comes to an end when each cell has fewer points than the `nodesize` points. In each classification tree, the average of the Y_i (that were among the points) is predicted for any query point $x \in X_i$ for which the corresponding X_i falls into the cell of x . The tree's growth and final prediction are solely dependent on a set of data points. A new data point's class is determined by a majority vote. The following parameters are critical to this algorithm:

- $a_n \in \{1, \dots, n\}$: the number of sampled data points each tree accounts for;
- `max_features`: Maximum number of variables in each individual tree that the Random Forest can try;
- `max_depth`: Each tree's maximum depth;
- `min_samples_leaf`: the minimum number of samples that determines the split of an internal node;
- $mtry \in \{1, \dots, p\}$: the number of possible splitting directions at each tree node;
- `n_estimator`: The number of trees that will be built;
- $nodesize \in \{1, \dots, a_n\}$: the number of observations that are in each cell below the cell which is not split.

Figure 2.1 shows a Random Forest model in which decision trees are built. The decision for the final output is made by committee based on the results of many individual trees. The Random Forest principle states that each tree is constructed with a randomly selected subset of variables. The results from each tree are then combined, usually through voting.

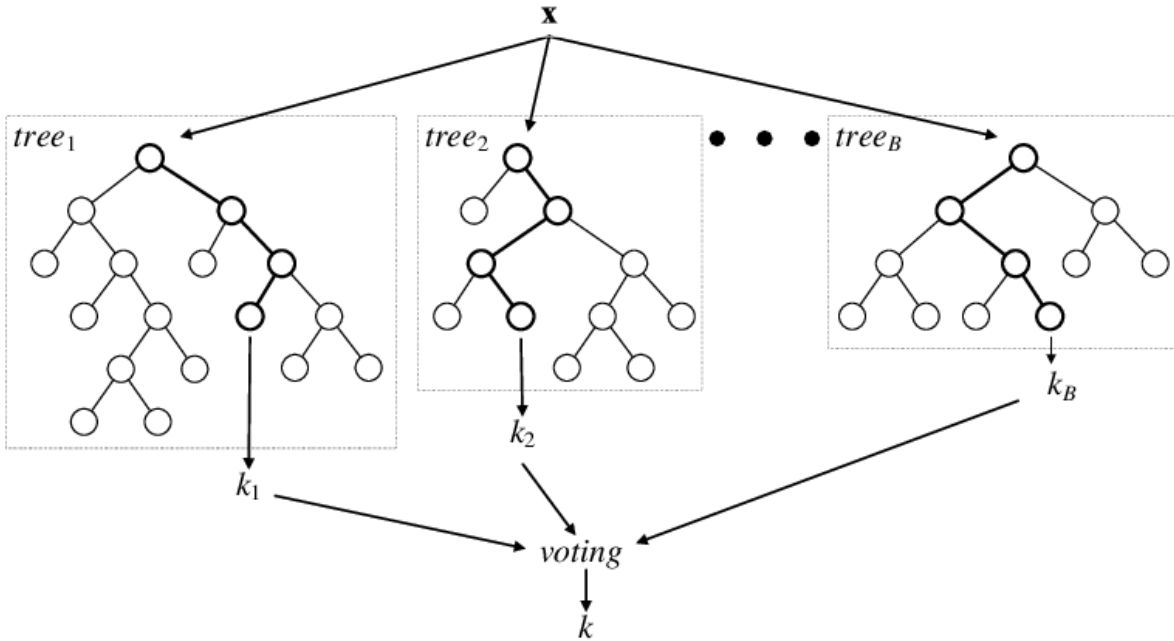


FIGURE 2.1: A general structure of a Random Forest Model [29]

2.2.3 Comparison of XGBoost and Random Forest

This section presents the differences between the XGBoost and Random Forest models.

The way in which decision trees are created and aggregated for both the Random Forest and XGBoost differ for each model. The decision trees in XGBoost are built one after another. In Random Forest, the decision trees are built at the same time. XGBoost is faster than Random Forest and it works effectively on imbalanced data [30]. Table 2.1 shows a comparison table between XGBoost and Random Forest models.

2.3 Data Augmentation

The previous studies conducted reviewed the effectiveness of the algorithms on the data that is currently available, but they do not address the issue of data scarcity. Insurance claim fraud is usually a small percentage of the total volume of claims. This makes it difficult for insurance companies to detect fraud cases in the midst of a high volume of claims to manage, resulting in a highly sparse dataset of fraudulent claims. A prediction model's optimality is hampered by a lack of data, which causes it to

TABLE 2.1: Differences between XGBoost and Random Forest comparison table [30]

XGBoost	Random Forest
XGBoost builds one tree at a time, making sure to include all data relevant to the decision tree and fill in any missing data. This allows developers to improve results by combining gradient algorithms with the decision tree algorithm.	The Random Forest model builds decision trees independently, so if an algorithm contains five trees, all of them are built at the same time but with different features and data. This forces developers to examine the trees and model them concurrently.
Because the gradient of the data is taken into account for each tree, the calculation is faster and the precision is higher than with Random Forest. This forces developers to rely on XGBoost rather than Random Forest.	When compared to XGBoost, the calculation takes time and is inaccurate. As a result, developers should not rely solely on Random Forest if other algorithms are available.
If the data is real-time so the data is imbalanced, we can use XGBoost where it performs exceptionally well.	Random Forest does not perform well on imbalanced data
The number of leaves in the algorithm is not taken into account by XGBoost. When the predictability of the model is poor, the algorithm performs better with more leaves in the decision tree. This reduces bias, and the results are entirely dependent on the data in the algorithm.	Random Forest has many trees with equal weight leaves, allowing for high accuracy and precision with the available data.

overfit based on the small volume of data available, causing models to underperform [31]. Under and over-sampling techniques are the two types of data augmentation techniques. The data augmentation techniques should be chosen with care because they significantly impact the quality of data and performance of models. However, the selection depends on the kind and size of data available.

The undersampling technique discards observations from the majority class (class with more samples than the other) to get an equal distribution of the majority and minority classes. This method is straightforward, but important information can be lost for prediction. The oversampling technique, on the other hand, generates more observations

for the minority class (class with less observations than the other). Representative oversampling techniques include random oversampling (ROS), and the synthetic minority oversampling technique (SMOTE) [32].

Oversampling techniques proved to be more more effective than undersampling techniques [33] because they don't delete any information from the dataset. However, the learning time is lengthy, and overfitting may occur. To solve class imbalance problems, this study aims to use two different data augmentation techniques, the SMOTE and ROS to generate synthetic data from actual data and then train machine learning models on it. Figure 2.2 illustrates the oversampling techniques. We discuss the two techniques in the following subsections.

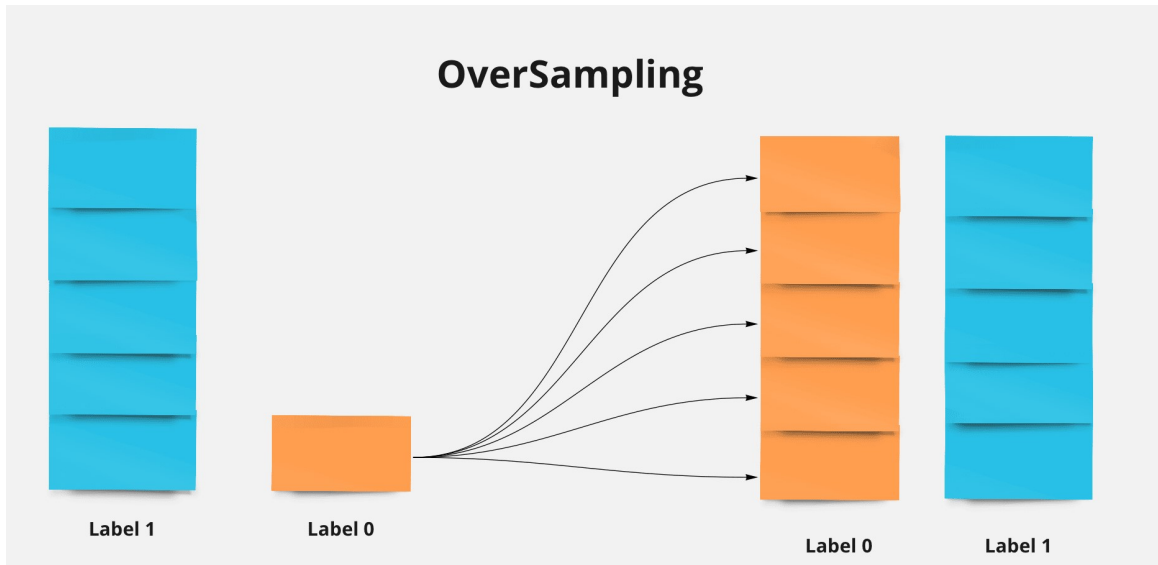


FIGURE 2.2: Illustration of the oversampling techniques [34]

2.3.1 SMOTE

SMOTE increases the size of a class that has fewer samples (minority samples) than the other class by combining existing samples to create synthetic samples. As a result, it lessens the problem of overfitting by generating new samples rather than making copies of existing ones. Also, there is no loss of importance.

The SMOTE procedure is made up of three tasks that are repeated. First, a sample of a minority class is chosen at random and denoted as x_{start} . Second, the k-nearest neighbors algorithm is used to further select more k minority class samples that are near x_{start} . In most cases, k of the nearest neighbors for the minority samples are found at $k=5$. Third, by randomly interpolating x_{start} and the k minority class samples, k synthetic samples are generated. This process is repeated until an equal distribution of the minority and majority classes is achieved [35]. Figure 2.3 shows how the SMOTE technique is used to create synthetic samples.

Sahayalakila and Aishwaryasikhakolli [37] conducted a study in Credit card fraud detection and proposed a detection system to detect fraudulent transactions based on the number of cardholder transactions. Credit card transactions are derived using kaggle datasets. Because of the small sample of fraudulent transactions in the dataset, the authors proposed the use of SMOTE technique to generate data for the minority class. The use of the SMOTE technique improved the quality of the generated data significantly. The models' performance improved.

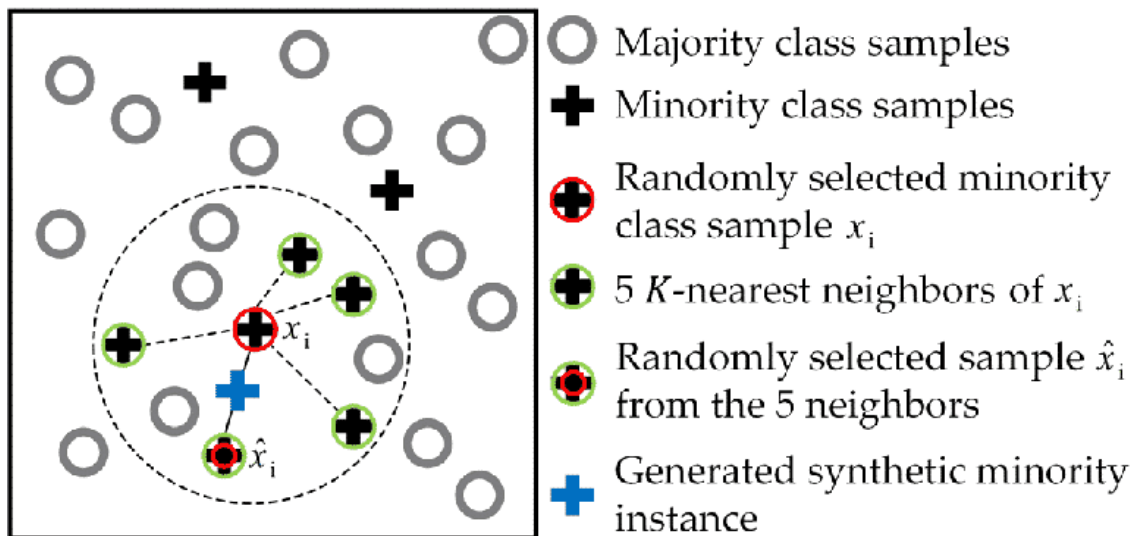


FIGURE 2.3: An illustration of the SMOTE technique used to generate synthetic samples [36]

2.3.2 ROS

Random oversampling duplicates samples from the class that makes up a smaller proportion of the dataset, which can lead in overfitting in some models. Random selection and replacement of observations from the training dataset is used. In other words, observations from the class with few samples can be chosen again because the original training dataset can be used as a starting point for selecting observations, adding them to the new training dataset, and then returning or replacing them. The class is defined with a `sampling_strategy` argument which is set to 'minority' to have equal distribution with the majority class [38].

Najadat and Altit [39] presented a system to detect fraud by applying machine learning and deep learning models on the IEEE-CIS Fraud Detection dataset to address the problem of credit card fraud. The results of machine learning models after applying random oversampling show that the random oversampling technique had no effect on the dataset.

2.4 Conclusion

In this chapter, we presented and discussed previous studies on fraud detection. Different machine learning models have been used to detect fraudulent claims based on insurance claims data. We then proposed two models, the XGBoost and Random Forest for our research work and discussed how the models work and compared them. These two models differ as XGBoost builds one tree at a time. On the other hand, the Random Forest model builds decision trees at the same time but with different features and data. We further discussed the SMOTE and ROS data augmentation techniques which are used to generate more data when there is data imbalance. The SMOTE technique is different from ROS because it does not repeat the minority examples as it generates more data in the minority class. However, the ROS technique generates more data for the minority class by repeating the minority examples.

Chapter 3

Research Methodology

This chapter provides the proposed methodology. In more details, it provides information about the data used in this research work, the techniques used to perform exploratory data analysis in order to understand the data first and try to gather as many insights from it. It also provides information about techniques used to pre-process the data to prepare it before the modelling phase. Finally, this chapter provides details about the machine learning classification models used, and the types of data performance evaluation metrics which are used to determine the performance of the models in detecting auto insurance claim fraud.

In this research work, we follow a quantitative research methodology to detect fraudulent insurance claims. Quantitative research is a type of methodology where numerical data and data that can be converted to numbers is processed and analysed [40]. This type of research approach can help us compute averages, find patterns, make predictions and test causal relationships [41].

3.1 Data

This research work uses an existing auto insurance claims data from the Kaggle website. Kaggle is a data science platform that enables users to discover and share datasets, and develop data-driven models. The dataset includes 1000 individual claims for an auto insurance company in the United States from 1990 to 2015 [42]. The dataset is in excel format. The reported fraud is a target variable whose values are predicted by other variables. This target variable is labelled 'N' if the claim is reported as non-fraudulent and 'Y' if the claim is reported as fraudulent. There are 40 different

attributes that are associated with every claim in the data.

Information about the incident includes the insured individual, the policy of the insured individual, the incident's details, and the features of the vehicle that was in the incident. The insurance claims dataset includes attributes that have both numerical and categorical values. Some variables include the insured's age, the insurable amount, and the premiums paid, the insured's job, how many vehicles were in the incident and the model of the vehicle that the claim was filed for.

TABLE 3.1: The insurance dataset elements

Dataset Elements	Quantity
Number of Rows	1000
Number of Columns	40
Number of Observations	40000
Missing Values	1888

3.2 Exploratory Data Analysis

Exploratory data analysis is used to discover meaningful information from data and to gain a better understanding of dataset variables and their relationships. Data analysis entails interpretation and an attempt to comprehend real-world data. This is accomplished by organizing the data in a way that improves understanding and also allows for a better presentation of the findings.

To begin with the exploratory data analysis, we created a dataframe to load the insurance claims data in Python Jupyter Notebook. The dataset was cleaned and analysed. We first examined the overall percentage of all claims reported as fraudulent, from the examination we found that the fraud rate is 24.7%. This indicates that the insurer has a significant problem with claims that are fraudulent as from this dataset nearly a quarter of the claims are confirmed fraudulent. The percentages of fraudulent and non-fraudulent claims, as well as the frequency with which they occur, are depicted in Figure 3.1.

In the next step we created some simple data visualizations to gain more insight. In Figure 3.2, we created graphs for some of the variables based on averages for insurance claims confirmed as fraudulent or non-fraudulent. From the figure we can see that fraudulent claims are not significantly different from non-fraudulent claims. However, there is a noticeable difference on the average claim amount for fraudulent claims for both vehicle and injury claims, it is higher. It is much expected for the number of fraudulent claims to be higher because people want to profit from them.

Furthermore, the umbrella limit is significantly higher for fraudulent claims on average. An umbrella limit provides an additional excess liability coverage beyond existing limits. The umbrella policy is of great worth when the claim exceeds a normal limit. It is worth the investment for people who have significant assets worth a lot of money. Only 20% of insured customers have umbrella limit in the insurance claims dataset. For everyone else, the umbrella limit is 0. The number of months a customer has been covered by insurance before filing a claim is not significantly different. Therefore, it cannot be concluded that customers commit more fraud during the early months of their insurance policy.

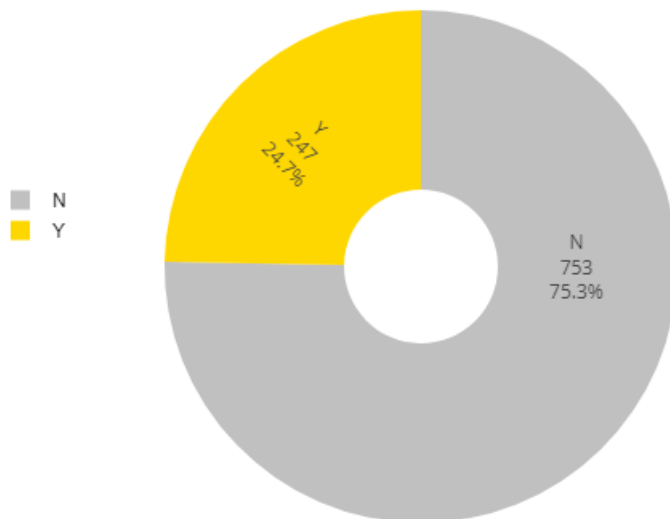
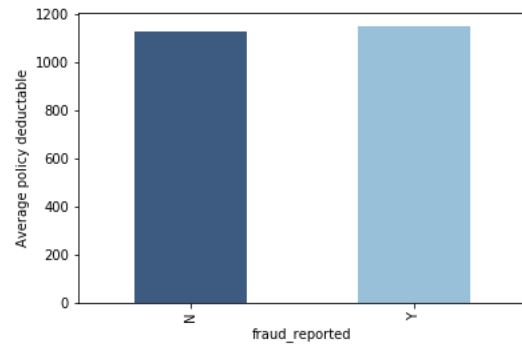
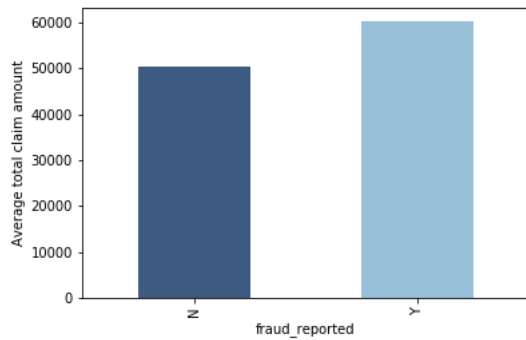
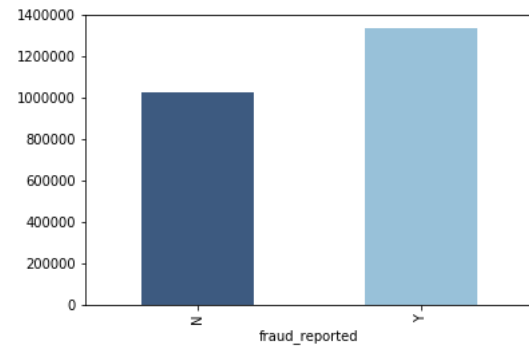
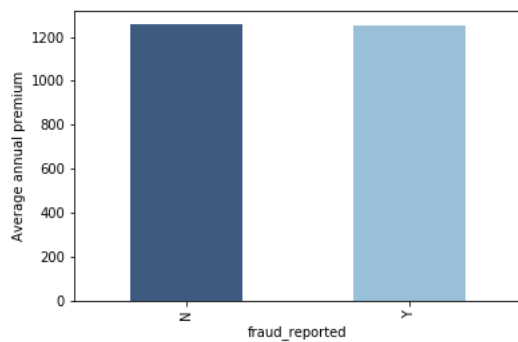


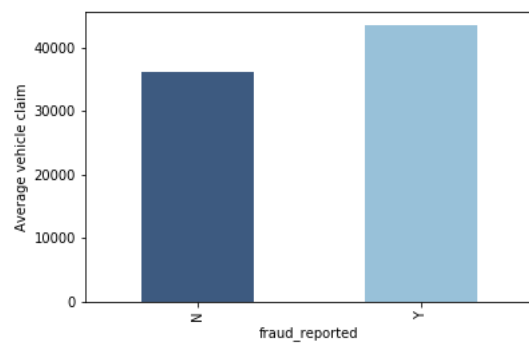
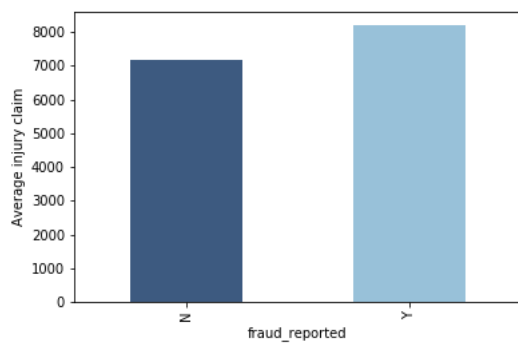
FIGURE 3.1: Distribution of frauds based on the target variable



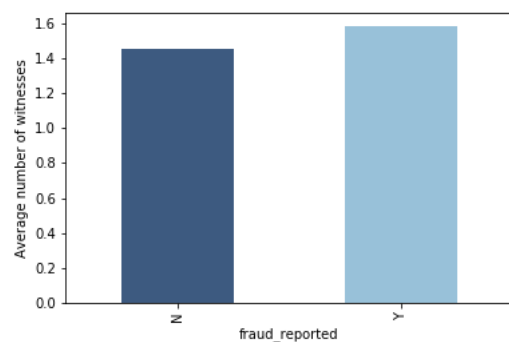
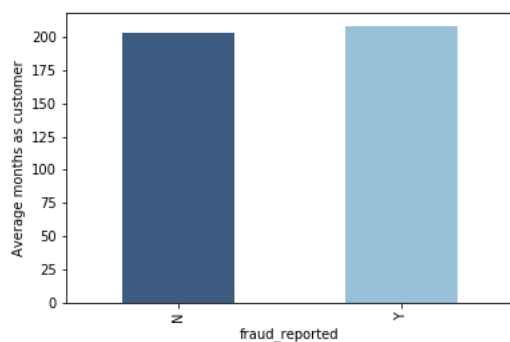
(A) Average total claim amount by Fraud Reported. (B) Average policy deductible by Fraud Reported



(C) Average annual premium by Fraud Reported (D) Average umbrella limit by Fraud Reported



(E) Average injury claim by Fraud Reported (F) Average vehicle by Fraud Reported

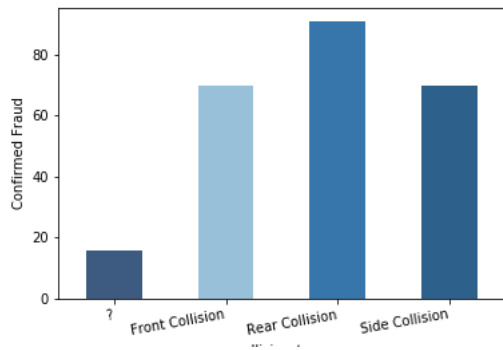


(G) Average months as customer by Fraud Reported (H) Average number of witnesses by Fraud Reported

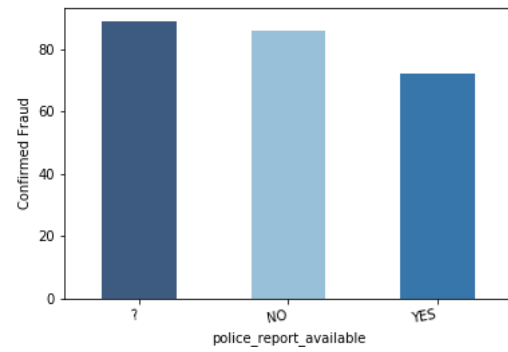
FIGURE 3.2: Averages for certain claims by Fraud Reported

Figure 3.3 is not focused on the insured's characteristics but more on the incidents. The figure displays the total number of fraudulent claims over the total number of all claims for each type of incident. It has been observed that single-vehicle and multi-vehicle collisions have a fraud rate that is triple the fraud rate of parked car and vehicle theft incidents. It is possible to conclude that these incidents must be carefully considered in for modeling. Even more impressive is the graph that depicts the fraud rate for various incident severity levels. Major damage claims are greater than five times as common as all other severities. Because this set of claims has a large proportion of all claims, it is critical to remember.

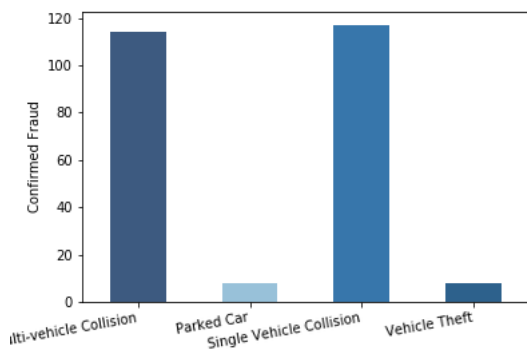
It has been observed from the analysis that, on fraud rate, females tend to make fraudulent claims than males. This could have resulted from the size of data we have and the fact that the data only covers a small portion of fraud from one specific area. Studies concluded, though, that when it comes to insurance fraud, gender and dishonesty are linked, and males are more likely than females to be dishonest [43]–[45]. The role of gender in insurance fraud tolerance was first investigated by [46], [47], to determine the customers' attitudes toward filing exaggerated auto insurance claims. Females, according to the author, are less tolerant of claim fraud. The author's findings have also been supported by experimental economic research, in which [48] and [49] discovered that it is not often that females commit insurance fraud.



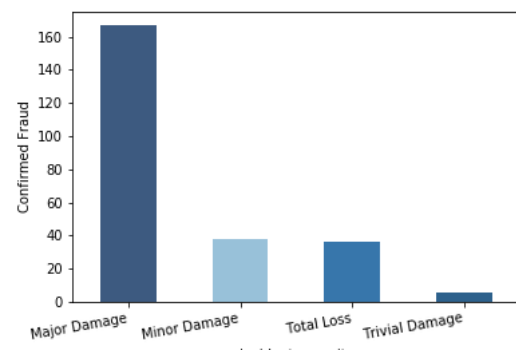
(A) Confirmed fraud by collision type



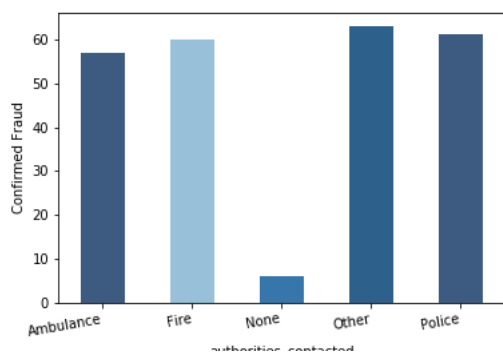
(B) Confirmed fraud by policy report available



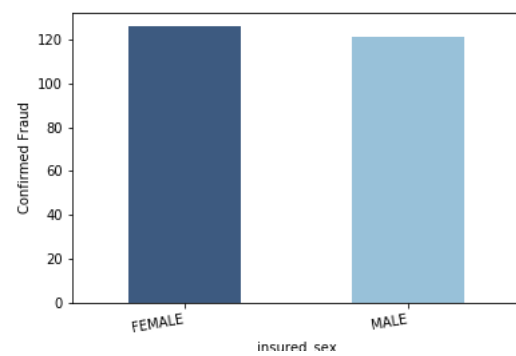
(C) Confirmed fraud by incident type



(D) Confirmed fraud by incident severity



(E) Confirmed fraud by authorities contacted



(F) Confirmed fraud by insured sex

FIGURE 3.3: Confirmed reported fraud by different types of incidents

3.3 Data Pre-processing

Data pre-processing is a fundamental stage of machine learning methods, which has a significant impact onto how the machine learning methods perform [50]. Data pre-processing is a technique used for cleaning and organising raw data to make it fit for training and evaluating machine learning models. Most real-world datasets contain missing and inconsistent data. Data pre-processing involves various steps that help to handle missing values, categorical values, biasness, and select meaningful features in the data and split data [51].

The data used in this research work is not processed for use by statistical methods. Proper pre-processing of the data is crucial for obtaining a reliable data matrix in which the actual data analysis can be performed. The pre-processing of the data greatly simplified data analysis. We discuss the pre-processing steps in the following sections.

3.3.1 Missing Values

Handling missing values is important as some machine learning algorithms do not recognise missing values. Missing values can reduce the performance of models and can produce biased estimates. We replace missing values as the dataset contains only 1000 observations, which is a very small number, and also to avoid deleting variables or rows with important information.

Variables with 100% missing values are removed as they do not serve any purpose in modelling. Other missing values are from categorical variables and the most frequent category method is used for some variables to replace the missing values. For variables having 'Yes' and 'No' responses we replaced the missing values with 'No' response by assuming that there were no responses. At the beginning of the study, we proposed to replace the missing values by predicting them using the XGBoost and Random Forest models, however this would be time consuming and not necessary as our dataset is small and the missing values are not many from the variables. The methods used works best on a small dataset and are easy to implement.

From the 40 variables in our dataset, only four have values that are missing. We used different methods to handle the values that are missing. The missing values were represented with a '?' that Python doesn't recognize, we replaced these with 'NaN'. The total size of our observations in the dataset is 40000. Out of this total, there are 1881 missing values. We removed one feature with 100% missing values. Collision type variable had 17.8% missing values. We looked for the category that occurred most in this variable and replaced the missing values with that category. For property damage and police report available features with missing values of 36% and 34.3% respectively, we assumed that it may be the case that there are no responses and took it as No property damage and No police report available.

3.3.2 Categorical Values

Label encoding is used to convert categorical data into numeric data as most models do not recognise categorical data. Label encoding creates categories with a value ranging from 0 to the number of classes minus one ($n_classes - 1$) where n is the number of distinct categories. If a category is repeated, it is assigned the same value as before [52]. For the target variable, the labels for non-fraudulent and fraudulent claims are converted to '0' and '1' respectively.

3.3.3 Feature Selection

Often when we get a dataset, we find too many highly correlated and non-relevant features in the dataset. All of the features we find in the dataset might not be useful and may negatively impact how the machine learning models perform. Thus it is good practice only to use a selection of the most valuable features.

To check and select the best relevant features for detecting fraudulent claims, we use the high correlation method for numerical features and chi-squared method for categorical variables. The high correlation method looks at the correlation between numerical variables and if they are highly correlated one of them is discarded as they have the same impact on the target variable. Correlation analysis depicts relationships among at least more than one variable. The aim of correlation analysis is to find the variables that have a significant relationship with the target variable [53]. In this research work, three variables namely, the vehicle claim, property claim and injury claim

were highly correlated with the total claim amount variable on a threshold of 0.8. The total claim amount was then dropped and we were left with 16 best numerical features.

The correlation heatmap in Figure 3.4 depicts the relationship between two variables, one on each axis. We can see if there are any patterns in value for one or both variables by observing how cell colors change across each axis. A good independent variable subset includes variables that are highly correlated (predictive of) to the target variable but uncorrelated (not predictive of) to each other [54].

The chi-squared method is used for categorical variables. This checks the importance of each variable in predicting the target variable. Chi-squared values range from 0 to 1. The closer the chi-squared score is to zero, the less significant the relationship between the two variables. The greater the chi-squared value, the stronger the relationship between the two variables [53]. Variables with chi-squared greater than 0.1 were chosen for modeling. The chi-squared method selected five best features from a total of 17 categorical features. We were then left with 21 variables for fraud prediction.

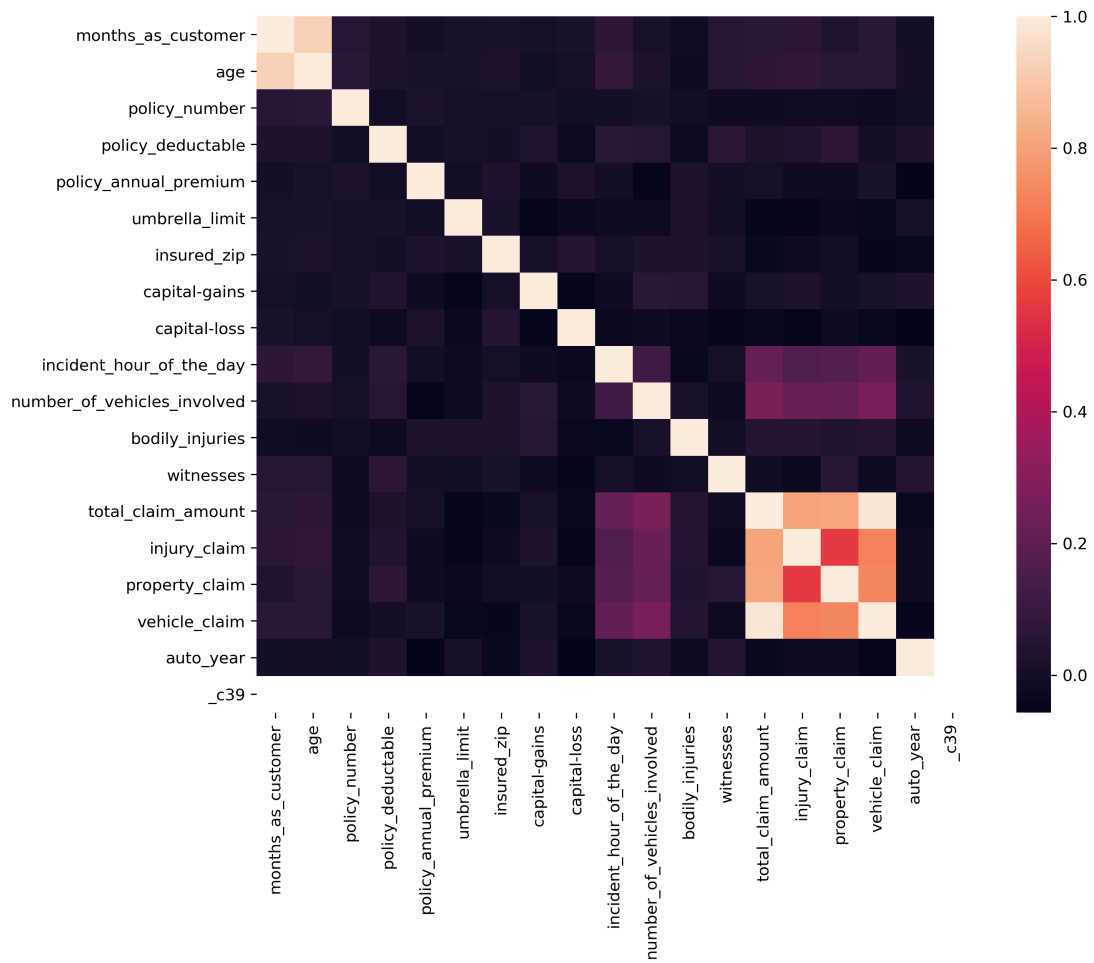


FIGURE 3.4: Correlation heatmap representing the correlation between different variables. The correlation values can be any value from -1 to 1

3.3.4 Training/Validation/Testing Split

The data is split into three sets to be able to train, validate and test the models: 70% training, 15% validation and 15% testing. The main reason for creating these sets is because we do not want to train the models and then validate and test them on the same data used for training because the models will definitely perform well because they would have been fed with the data before. This helps in terms of developing models that are not prone to overfitting, that is, models that are able to perform well on new data.

Training data is used to train the models so that they learn as much as they can from the data. Validation data introduces into the models new data that has not been seen by the models before. Validation data is the first test against unseen data, allowing us to evaluate how well the models predict new data. Testing data validates that the models can make accurate predictions after they are built. The test data is used to do a final, real-world check of data that has not been used during training and validation to check if a model performs accurately. Figure 3.5 depicts the process of splitting the original dataset into training, validation and testing sets. Table 3.2 shows the number of fraudulent and non-fraudulent claims for each set.

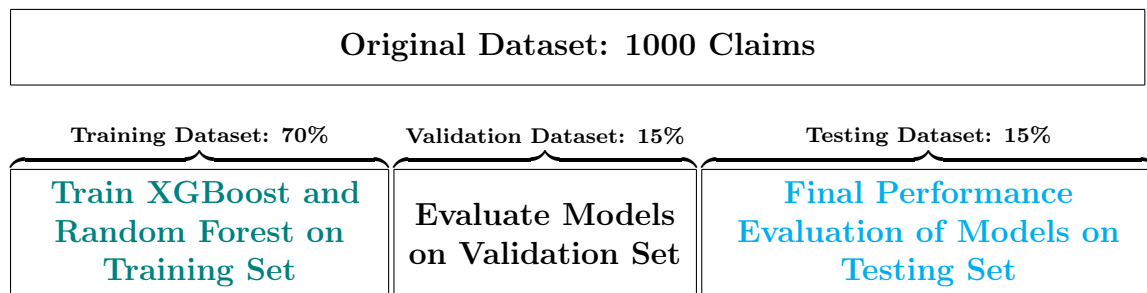


FIGURE 3.5: Splitting Dataset into Training, Validation and Testing Sets

3.3.5 Data Augmentation

The performance of most machine learning models depends on the quantity and diversity of the data. The data we are working on has more non-fraudulent claims than fraudulent claims (247 fraudulent claims and 753 non-fraudulent claims) as shown in

TABLE 3.2: Number of fraudulent and non-fraudulent claims for each set

Original Sets			
Claims	Sets		
	Training	Validation	Testing
Fraudulent	167	39	41
Non-fraudulent	533	111	109

Figure 3.1. This means that observations in the data have more non-fraudulent claims than fraudulent claims. For the models that is developed, this means that there is a chance for bias towards the fraudulent class. This could cause the models to overpredict a high number of non-fraudulent claims.

We need to ensure that our models not only have good accuracy and good quality, but are fair. To address this issue, we want to ensure that the data contains a large number of examples of both fraudulent and non-fraudulent claims. We achieve this by performing data augmentation which artificially increases the data used to train a model. Data augmentation is done by applying domain-specific techniques: Synthetic Minority Oversampling Technique (SMOTE) and Random Oversampling to observations from the training data that generates new and representative training observations. These techniques are only applied to the training dataset so that we train our models properly on the data. The validation and testing data remain unchanged so that they correctly represent the original data [55].

The SMOTE increases the number of claims in our dataset. It generates new instances from the minority class (fraudulent claims) in the training dataset. The implementation of SMOTE does not change the majority class (non-fraudulent claims). The Random Oversampling technique increases the number of claims in our dataset through duplication of the original fraudulent claims. SMOTE is different from Random Oversampling in that SMOTE does not only increase the size of the training data, but also increases the variety because it doesn't duplicate claims. The techniques result in 50:50 distribution for the fraudulent and non-fraudulent claims. Table 3.3 shows the distribution of claims when SMOTE and ROS are used.

TABLE 3.3: Number of fraudulent and non-fraudulent claims for each set when SMOTE and ROS techniques are used

SMOTE			
Claims	Sets		
	Training	Validation	Testing
Fraudulent	533	39	41
Non-fraudulent	533	111	109
ROS			
Claims	Sets		
	Training	Validation	Testing
Fraudulent	533	39	41
Non-fraudulent	533	111	109

3.4 Models

This section explains how the XGBoost and Random Forest models are used to predict whether a claim is fraudulent or non-fraudulent. This is a classification problem as our input data is assigned into fraudulent and non-fraudulent classes. Classification refers to a method of identifying the class of a new input data point from the set of classes based on a labeled training set. Classification problems are classified as supervised machine learning. Supervised learning is the most commonly used learning technique, in which a model is trained using labelled data (i.e., data for which the outcome is known). The models are provided with the input along with the output [56]. In auto insurance fraud detection, the class labels we are interested in are whether a claim is fraudulent or not fraudulent. The training set of data is used to develop the two models. The category of any new insurance claim can be found by using the trained model. If the claim follows a pattern similar to non-fraudulent behaviour, the claim is classified as a non-fraudulent claim, otherwise it is classified as a fraudulent claim.

3.4.1 Prediction

Supervised machine learning uses data that has a target variable and input variables (predictors of the target variable) to train algorithms that learn the data and discover patterns in the data in order to predict a class of new data points. The process of how the XGBoost and Random Forest models are trained is presented in details in chapter 2 of Literature Review. Figure 3.6 shows the process of creating a model and how the

created model is used for prediction. In the following subsections we briefly explain how the models are used for prediction.

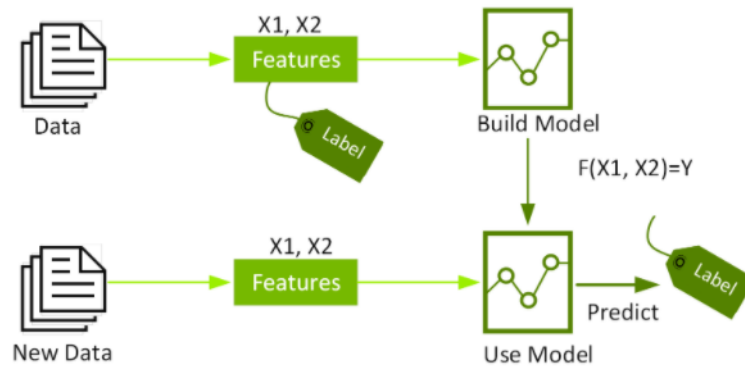


FIGURE 3.6: Creation of a model and using it for prediction [57]

3.4.2 XGBoost

To develop the XGBoost model in Python we applied the following procedure:

- i. Import Pandas library and load the insurance claims data using Pandas to create a dataframe
- ii. Select the features to be used for prediction and the target
- iii. Use sklearn to split the dataset into training, validation and testing
- iv. Import the XGBoost classifier function (XGBClassifier) from XGBoost ensemble module.
- v. Define the XGBoost model
- vi. Fit a XGBoost model using the train set
- vii. Predict the labels or target variable using the XGBoost model and the scikit-learn function `model.predict()`
- viii. Evaluate the model (confusion matrix, precision, recall, F1 score).

The XGBoost model makes predictions by following the process shown below [25]:

- i. XGBoost algorithm generates decision trees in a sequential manner
- ii. All the input variables are assigned with weights. The weights are then fit into the first decision tree that predicts the outcomes
- iii. When data points are incorrectly predicted by the first tree, their weight increases, and those incorrectly predicted data points are then fit into the second decision tree
- iv. The preceding step is repeated until the final decision tree predicts all data points correctly or a maximum number of decision trees are added
- v. The final model makes predictions on new claims by taking the testing input variables from the insurance claims data and predicts the outcome based on how many votes each predicted target received
- vi. Take the final prediction of the model to be the target with the highest number of votes.

3.4.3 Random Forest

The following procedure was applied to develop the Random Forest model:

- i. Import Pandas library and load the insurance data using Pandas to create a dataframe
- ii. Find out important features and visualize them using Seaborn and Matplotlib
- iii. Define the predictors (features) and the target
- iv. Split the dataset into training, validation and testing using sklearn
- v. Import the Random Forest classifier function (RandomForestClassifier) from sklearn ensemble module
- vi. Define the Random Forest model and state the random state to 1 to produce same sets every time we run
- vii. Fit a Random Forest classifier model using the train set

- viii. Predict the labels or target variable using the Random Forest classifier model and the scikit-learn function `model.predict()`
- ix. Evaluate the model (confusion matrix, precision, recall, F1 score).

The Random Forest model makes predictions by following the process shown below:

- i. Takes the test features from the insurance claims data and predicts the outcome based on the rules of each decision tree generated randomly, then stores the predicted outcome of the target
- ii. Determine how many votes each predicted target received
- iii. Take the final prediction of the Random Forest algorithm to be the target with the highest number of votes.

3.5 Performance Evaluation of Models

Evaluation metrics such as confusion matrix, precision, recall and f1 score are used to evaluate the models. The metrics help us to see if the models are able to correctly detect a claim as fraudulent or non-fraudulent. From the metrics, we are able to conclude if the models were able to learn as much during training. The results are compared for each model based on training, validation and testing to observe how the models perform towards the unseen data.

There are four cases the models could end up with. We map these cases to the performance evaluation metrics based on their relation. These cases are the summary of prediction results which help us uncover important details about the performance of the models. They are easily found in the confusion matrix as displayed in Table 3.4. In this research work, positive means fraudulent and negative means non-fraudulent.

- i. True Positive (TP), when a claim is fraudulent and the models correctly predict it as fraudulent.
- ii. True Negative (TN), when a claim is non-fraudulent and the models correctly predict it as non-fraudulent.

TABLE 3.4: A confusion matrix to show the performance of a classification model where actual values for all the sets are known.

		Predicted Values	
		Negative	Positive
Actual Values	Negative	TN	FP
	Positive	FN	TP

- iii. False Positive (FP), when a claim is non-fraudulent, but the models predict it as fraudulent.
- iv. False Negative (FN), when a claim is fraudulent, but the models predict it as non-fraudulent.

What we desire is True Positive and True Negative but due to misclassifications that may be present in machine learning models, we may also end up with more False Positive and False Negative. So it means there is confusion in detecting whether a claim is fraudulent or non-fraudulent. The evaluation metrics are defined based on their relation with the cases discussed above.

Accuracy - the number of claims that are predicted correctly out of all the claims.

Precision - the number of positive claims that are correctly predicted out of the total predicted positive claims. The question that this metric answer is of all claims that are predicted as Y (fraudulent), how many actually are fraudulent?

Recall - the number of positive claims correctly predicted to the total claims in the true positive class. The question that recall answers is: Of all the claims that truly are Y (fraudulent), how many did we label?

F1 Score - mean from precision and recall, it rates how the machine learning models perform.

Support - the number of actual occurrences that lies in each class of the target variable.

The metrics are computed as:

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

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

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

$$F1Score = \frac{2 * (Precision + Recall)}{Precision + Recall}$$

3.6 Conclusion

In this chapter, we discussed the insurance claims data. The data was explored and pre-processed to prepare it for modeling. The data preparation and exploration removed irrelevant variables and missing values; conducted correlation analysis to understand the relationship between variables and the target variable; selected important variables for modeling; and transformed categorical variables to numerical form for the two models. We also discussed how the two models are used for prediction of fraudulent claims. Finally, we discussed the model performance evaluation metrics.

Chapter 4

Results and Discussion

This chapter contains the presentation and discussion of the results of this research work. The findings contain results before and after data augmentation was performed.

4.1 Model Results

This section presents the experimental results of the XGBoost and Random Forest models. The accuracy for the models is good but it is not considered to make a conclusion on how the models perform because it is misleading. The accuracy might only be predicting the non-fraudulent claims correctly as the classes are not balanced, whereas we are interested in fraudulent claims. This is then not a correct metric for our models given the seriousness of the issue. We should be measuring how many fraudulent claims we can predict correctly to fight the rise of insurance fraud or maybe, out of the correctly predicted claims, we should check fraudulent claims to check the reliability of our model. To solve this problem of false idea about the models' performance based on accuracy, we used the precision and recall, confusion matrix and Precision-Recall curves.

4.1.1 XGBoost: Before data augmentation

The XGBoost model on the validation set based on fraudulent claims has a precision of 0.57, that means when it predicts fraudulent claims, it is correct 57% of the time. The model has a recall of 0.51, meaning it correctly predicts 51% of all fraudulent claims as shown in Table 4.1. The model on the testing set based on fraudulent claims has a precision of 0.55, that means it is correct 55% of the time when it predicts fraudulent claims. The model has a recall of 0.59, meaning it correctly predicts 59%

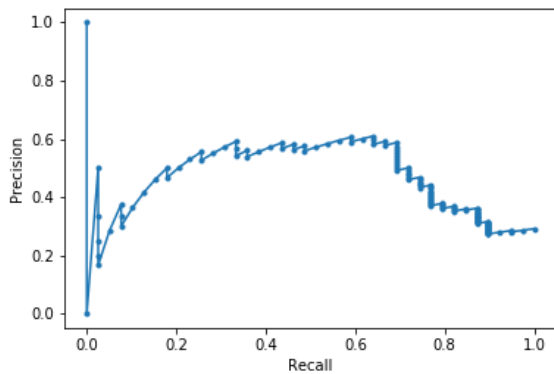
of all fraudulent claims. This is shown in Table 4.1.

Figure 4.1 shows precision vs recall curves and confusion matrices as subfigures on both the validation and testing sets. The precision vs recall curves is a graph with the precision on the y axis and the recall on the x axis. From the curves, we can observe the shapes we would expect; when precision is high, the recall is correspondingly low, and at very low precision, the recall is high. High precision means that incorrectly predicting a claim as fraudulent is costly whereas incorrectly predicting the claim as non-fraudulent is not as costly. On the other hand, high recall means that claims that are incorrectly predicted as non-fraudulent are more costly than claims that are incorrectly predicted as fraudulent, that is, predicting a fraudulent claim when it is not fraudulent is much better than saying a claim is not fraudulent when it actually is fraudulent.

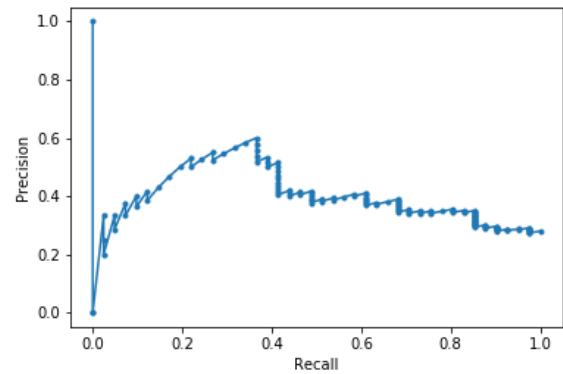
The confusion matrices of the XGBoost on both the validation and testing sets model contain a summary of the actual and predicted values including correct and incorrect predictions. Based on the validation set from the confusion matrix on the bottom left, the XGBoost model predicted 116 claims correctly. Of this 116, only 20 are fraudulent claims. The model has 34 incorrectly predicted claims. Of this 34 claims, 19 are incorrectly predicted as non-fraudulent on the validation set. On the bottom right confusion matrix we can see that the XGBoost on the testing set correctly predicted 112 claims and from this 112 claims, only 13 are fraudulent. The model has predicted 38 claims incorrectly. Of this 38, 28 are incorrectly predicted as non-fraudulent.

TABLE 4.1: Evaluation metrics of XGBoost on validation and testing sets without data augmentation

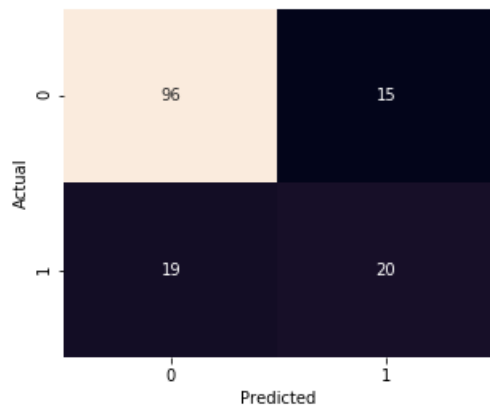
Validation				
Class	Precision	Recall	F1 Score	Support
0	0.83	0.86	0.85	111
1	0.57	0.51	0.54	39
Avg/Total	0.70	0.69	0.70	150
Testing				
Class	Precision	Recall	F1 Score	Support
0	0.85	0.83	0.84	111
1	0.55	0.59	0.57	39
Avg/Total	0.70	0.71	0.70	150



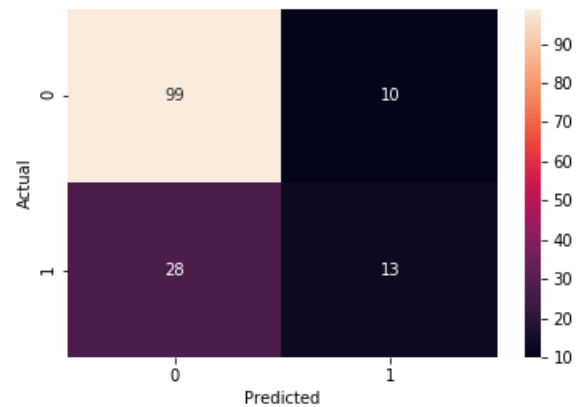
(A) Precision vs Recall for XGBoost on validation set



(B) Precision vs Recall for XGBoost on testing set



(C) Confusion Matrix of the XGBoost on validation set before data augmentation



(D) Confusion Matrix of the XGBoost on testing set before data augmentation

FIGURE 4.1: PR curves and Confusion Matrices for the XGBoost model before data augmentation

4.1.2 XGBoost: SMOTE

Here we present the model results when SMOTE is used. The XGBoost model on the validation set based on fraudulent claims has a precision of 0.55, that means when it predicts fraudulent claims, it is correct 55% of the time. The model has a recall of 0.59, meaning it correctly predicts 59% of all fraudulent claims as shown in Table 4.2. The model on the testing set based on fraudulent claims has a precision of 0.56, that means it is correct 56% of the time when it predicts fraudulent claims. The model has a recall of 0.44, meaning it correctly predicts 44% of all fraudulent claims. This is

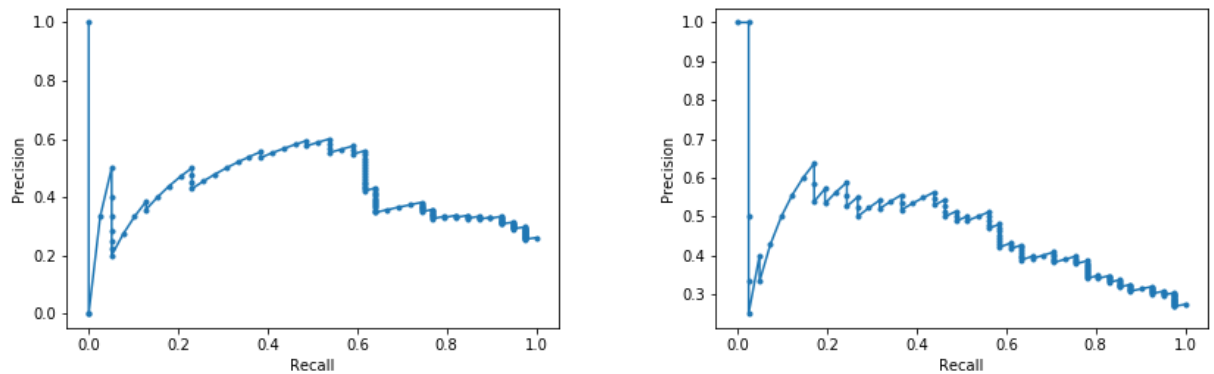
shown in Table 4.2. When compared with prediction without data augmentation on fraudulent claims, the prediction on validation set only increased the recall and the prediction on testing set increased both the the precision and recall.

Figure 4.2 shows precision vs recall curves and confusion matrices as subfigures on both the validation and testing sets. From the curves we can observe that when precision is high, the recall is correspondingly low, and at very low precision, the recall is high. High precision means that incorrectly predicting a claim as fraudulent is costly whereas incorrectly predicting the claim as non-fraudulent is not as costly. On the other hand, high recall means that claims that are incorrectly predicted as non-fraudulent are more costly than claims that are incorrectly predicted as fraudulent, that is, predicting fraud when it is not fraud is much better than saying no fraud when there actually is fraud.

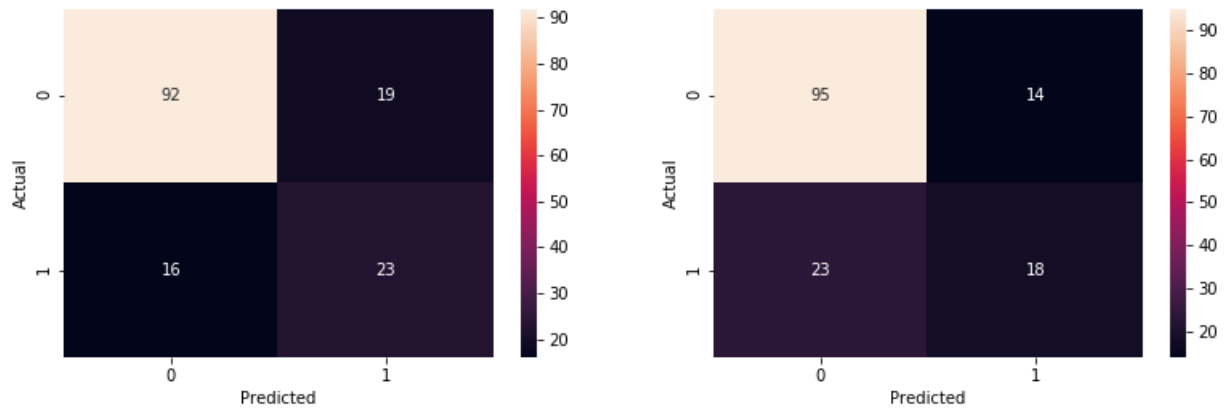
The confusion matrices of the XGBoost on both the validation and testing sets model contain a summary of the actual and predicted values including correct and incorrect predictions. Based on the validation set from the confusion matrix on the bottom left, the XGBoost model predicted 115 claims correctly. Of this 115, only 23 are fraudulent claims. The model has 34 incorrectly predicted claims. Of this 35 claims, 16 are incorrectly predicted as non-fraudulent on the validation set. On the bottom right confusion matrix we observe that the XGBoost on the testing set correctly predicted 113 claims and from this 113 claims, only 18 are fraudulent. The model has predicted 37 claims incorrectly. Of this 37, 23 are incorrectly predicted as non-fraudulent.

TABLE 4.2: Evaluation metrics of XGBoost on validation and testing sets using SMOTE

Validation				
Class	Precision	Recall	F1 Score	Support
0	0.85	0.83	0.84	111
1	0.55	0.59	0.57	39
Avg/Total	0.70	0.71	0.70	150
Testing				
Class	Precision	Recall	F1 Score	Support
0	0.81	0.87	0.84	109
1	0.56	0.44	0.49	41
Avg/Total	0.68	0.66	0.67	150



(A) Precision vs Recall for XGBoost on validation set (B) Precision vs Recall for XGBoost on testing set using SMOTE



(C) Confusion Matrix of the XGBoost on validation set (D) Confusion Matrix of the XGBoost on testing set using SMOTE

FIGURE 4.2: PR curves and Confusion Matrices for the XGBoost model using SMOTE

4.1.3 XGBoost: ROS

Here we present the model results when ROS is used. The XGBoost model on the validation set based on fraudulent claims has a precision of 0.55, that means when it predicts fraudulent claims, it is correct 55% of the time. The model has a recall of 0.54, meaning it correctly predicts 54% of all fraudulent claims as shown in Table 4.3. The model on the testing set based on fraudulent claims has a precision of 0.50, that means it is correct 50% of the time when it predicts fraudulent claims. The model has a recall of 0.39, meaning it correctly predicts 39% of all fraudulent claims. This is

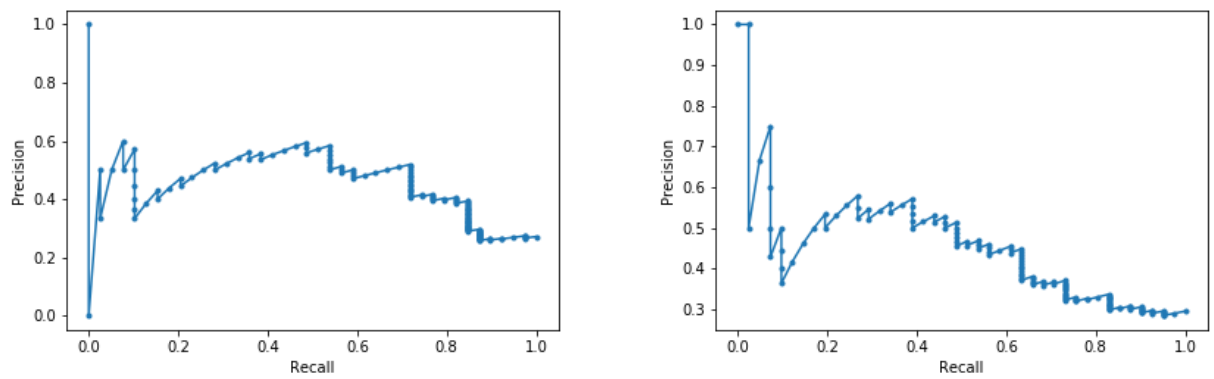
shown in Table 4.3. When compared with prediction without data augmentation on fraudulent claims, the prediction on the validation set only increased the recall and the prediction on testing set decreased both the the precision and recall.

Figure 4.3 shows precision vs recall curves and confusion matrices as subfigures on both validation and testing sets. From the curves we can observe that when precision is high, the recall is correspondingly low, and at very low precision, the recall is high. High precision means that incorrectly predicting a claim as fraudulent is costly whereas incorrectly predicting the claim as non-fraudulent is not as costly. On the other hand, high recall means that claims that are incorrectly predicted as non-fraudulent are more costly than claims that are incorrectly predicted as fraudulent, that is, predicting fraud when it is not fraud is much better than saying no fraud when there actually is fraud.

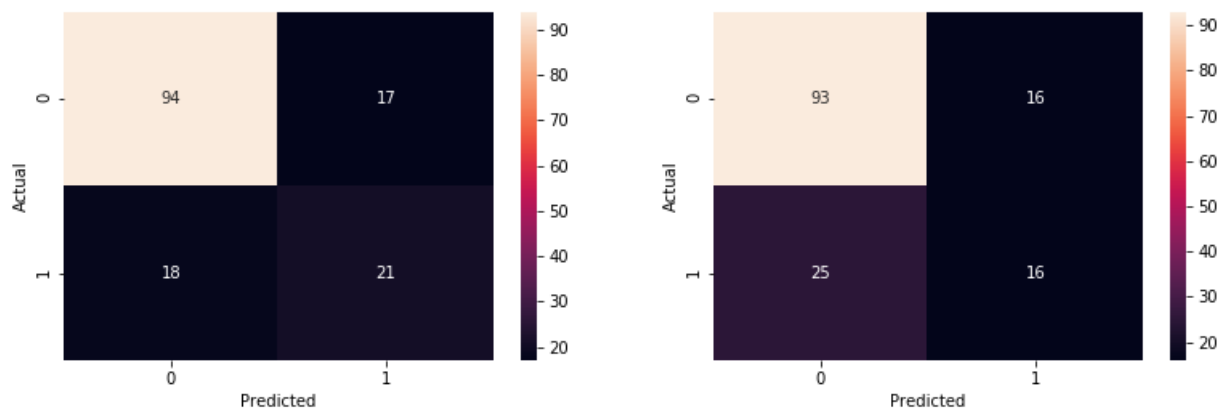
The confusion matrices of the XGBoost on both the validation and testing sets model contain a summary of the actual and predicted values including correct and incorrect predictions. Based on validation set from the confusion matrix on the bottom left, the XGBoost model predicted 115 claims correctly. Of this 115, only 21 are fraudulent claims. The model has 35 incorrectly predicted claims. Of this 35 claims, 18 are incorrectly predicted as non-fraudulent on the validation set. On the bottom right confusion matrix we observe that the XGBoost on the testing set correctly predicted 109 claims and from this 109 claims, only 16 are fraudulent. The model has predicted 41 claims incorrectly. Of this 41, 25 are incorrectly predicted as non-fraudulent.

TABLE 4.3: Evaluation metrics of XGBoost on validation and testing sets using ROS

Validation				
Class	Precision	Recall	F1 Score	Support
0	0.84	0.85	0.84	111
1	0.55	0.54	0.55	39
Avg/Total	0.70	0.69	0.69	150
Testing				
Class	Precision	Recall	F1 Score	Support
0	0.79	0.85	0.82	109
1	0.50	0.39	0.44	41
Avg/Total	0.64	0.62	0.63	150



(A) Precision vs Recall for XGBoost on validation set (B) Precision vs Recall for XGBoost on testing set using ROS



(C) Confusion Matrix of the XGBoost on validation set (D) Confusion Matrix of the XGBoost on testing set using ROS

FIGURE 4.3: PR curves and Confusion Matrices for the XGBoost model using ROS

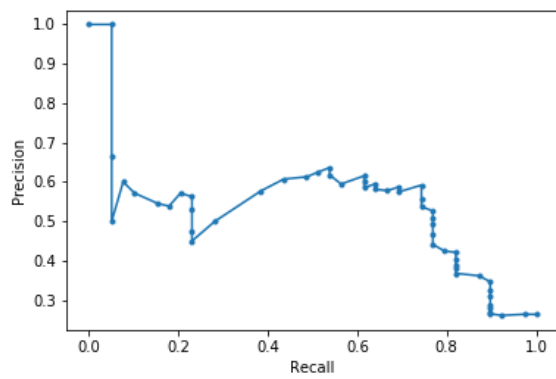
4.2 Random Forest

4.2.1 Random Forest: Before data augmentation

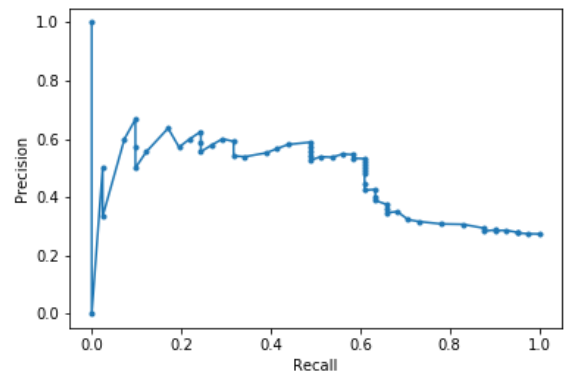
The Random Forest model on the validation set based on fraudulent claims has a precision of 0.60, that means when it predicts fraudulent claims, it is correct 60% of the time. The model has a recall of 0.46, meaning it correctly predicts 46% of all fraudulent claims as shown in Table 4.4. The model on the testing set based on fraudulent claims has a precision of 0.52, that means it is correct 52% of the time when it predicts fraudulent claims. The model has a recall of 0.34, meaning it correctly predicts 34% of all fraudulent claims. This is shown in Table 4.4.

Figure 4.4 shows precision vs recall curves and confusion matrices as subfigures on both the validation and testing sets. From the curves we can observe that when precision is high, the recall is correspondingly low, and at very low precision, the recall is high. High precision means that incorrectly predicting a claim as fraudulent is costly whereas incorrectly predicting the claim as non-fraudulent is not as costly. On the other hand, high recall means that claims that are incorrectly predicted as non-fraudulent are more costly than claims that are incorrectly predicted as fraudulent, that is, predicting fraud when it is not fraud is much better than saying no fraud when there actually is fraud.

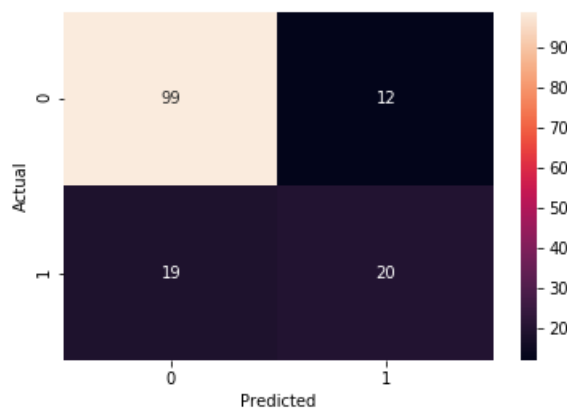
The confusion matrices of the Random Forest on both the validation and testing sets model contain a summary of the actual and predicted values including correct and incorrect predictions. Based on the validation set from the confusion matrix on the bottom left, the Random Forest model predicted 119 claims correctly. Of this 119, only 20 are fraudulent claims. The model has 31 incorrectly predicted claims. Of this 31 claims, 19 are incorrectly predicted as non-fraudulent on the validation set. On the bottom right confusion matrix we can see that the Random Forest on the testing set correctly predicted 113 claims and from this 113 claims, only 13 are fraudulent. The model has predicted 37 claims incorrectly. Of this 37, 28 are incorrectly predicted as non-fraudulent.



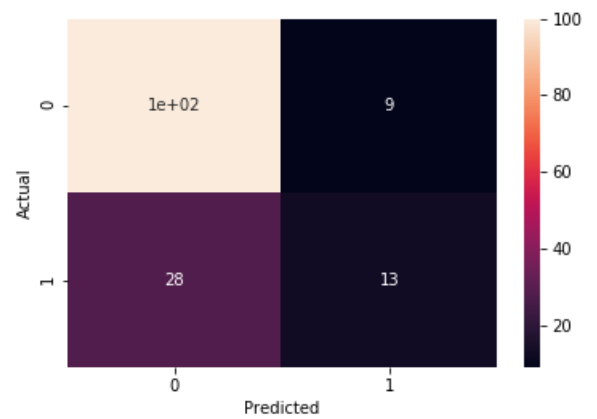
(A) Precision vs Recall for Random Forest on validation set



(B) Precision vs Recall for Random Forest on testing set



(C) Confusion Matrix of the Random Forest on validation set



(D) Confusion Matrix of the Random Forest on testing set

FIGURE 4.4: PR curves and Confusion Matrices for the Random Forest model before data augmentation

TABLE 4.4: Evaluation metrics of Random Forest on validation and testing sets before data augmentation

Validation				
Class	Precision	Recall	F1 Score	Support
0	0.82	0.89	0.86	111
1	0.60	0.46	0.52	39
Avg/Total	0.71	0.68	0.69	150
Testing				
Class	Precision	Recall	F1 Score	Support
0	0.78	0.88	0.83	109
1	0.52	0.34	0.41	41
Avg/Total	0.65	0.61	0.62	150

4.2.2 Random Forest: SMOTE

When the SMOTE technique is used, the Random Forest model on the validation set based on fraudulent claims has a precision of 0.59, that means when it predicts fraudulent claims, it is correct 59% of the time. The model has a recall of 0.62, meaning it correctly predicts 62% of all fraudulent claims as shown in Table 4.5. The model on the testing set based on fraudulent claims has a precision of 0.65, that means it is correct 65% of the time when it predicts fraudulent claims. The model has a recall of 0.59, meaning it correctly predicts 59% of all fraudulent claims. This is shown in Table 4.5. When compared with prediction without data augmentation on fraudulent claims, the prediction on validation set only increased the recall score and the prediction on testing set increased both the the precision and recall.

Figure 4.5 shows precision vs recall curves and confusion matrices as subfigures on both the validation and testing sets. From the curves we can observe that when precision is high, the recall is correspondingly low, and at very low precision, the recall is high. High precision means that incorrectly predicting a claim as fraudulent is costly whereas incorrectly predicting the claim as non-fraudulent is not as costly. On the other hand, high recall means that claims that are incorrectly predicted as non-fraudulent are more costly than claims that are incorrectly predicted as fraudulent, that is, predicting fraud when it is not fraud is much better than saying no fraud when there actually is fraud.

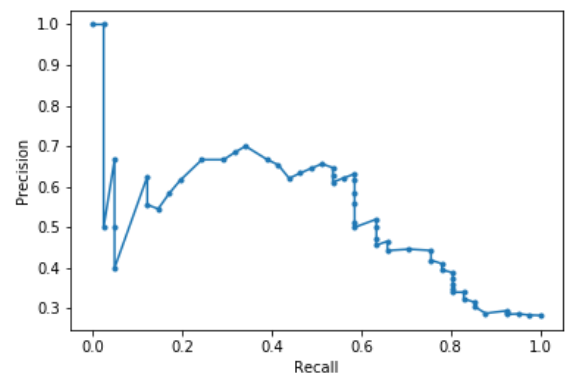
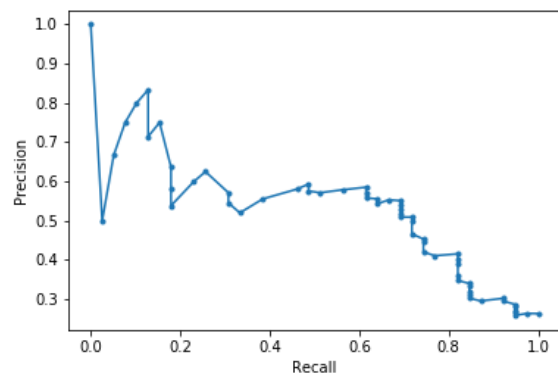
The confusion matrices of the Random Forest on both the validation and testing sets model contain a summary of the actual and predicted values including correct and incorrect predictions. Based on the validation set from the confusion matrix on the bottom left, the Random Forest model predicted 117 claims correctly. Of this 117, only 22 are fraudulent claims. The model has 35 incorrectly predicted claims. Of this 35 claims, 16 are incorrectly predicted as non-fraudulent on the validation set. On the bottom right confusion matrix we can see that the Random Forest on the testing set correctly predicted 113 claims and from this 113 claims, only 18 are fraudulent. The model has predicted 37 claims incorrectly. Of this 37, 23 are incorrectly predicted as non-fraudulent.

TABLE 4.5: Evaluation metrics of Random Forest on validation and testing sets using SMOTE

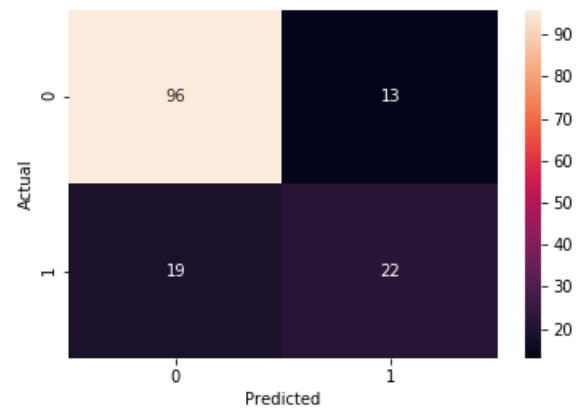
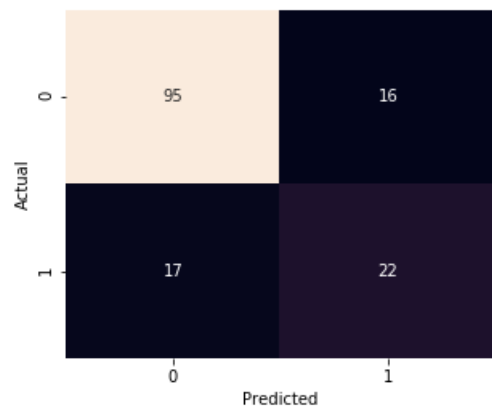
Validation				
Class	Precision	Recall	F1 Score	Support
0	0.86	0.85	0.85	111
1	0.59	0.62	0.60	39
Avg/Total	0.72	0.73	0.73	150
Testing				
Class	Precision	Recall	F1 Score	Support
0	0.85	0.88	0.86	109
1	0.65	0.59	0.62	41
Avg/Total	0.75	0.73	0.74	150

4.2.3 Random Forest: ROS

When the ROS technique is used, the Random Forest model on the validation set based on fraudulent claims has a precision of 0.56, that means when it predicts fraudulent claims, it is correct 56% of the time. The model has a recall of 0.56, meaning it correctly predicts 56% of all fraudulent claims as shown in Table 4.6. The model on the testing set based on fraudulent claims has a precision of 0.65, that means it is correct 65% of the time when it predicts fraudulent claims. The model has a recall of 0.49, meaning it correctly predicts 49% of all fraudulent claims. This is shown in Table 4.6. When compared with prediction without data augmentation on fraudulent claims, the prediction on validation set only increased the recall score and the prediction on



(A) Precision vs Recall for Random Forest on validation set using SMOTE (B) Precision vs Recall for Random Forest on testing set using SMOTE



(C) Confusion Matrix of the Random Forest on validation set using SMOTE (D) Confusion Matrix of the Random Forest on testing set using SMOTE

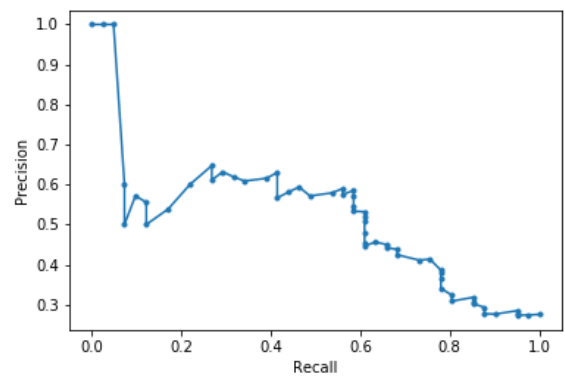
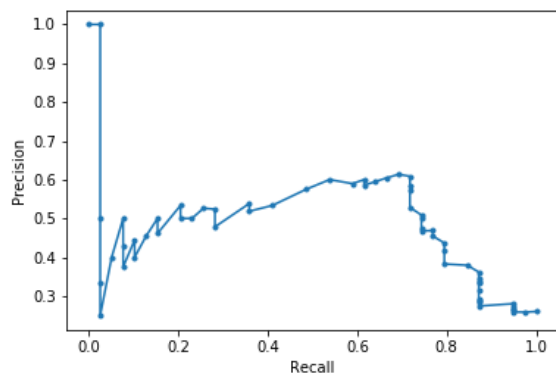
FIGURE 4.5: PR curves and Confusion Matrices for the Random Forest model using SMOTE

testing set increased both the the precision and recall.

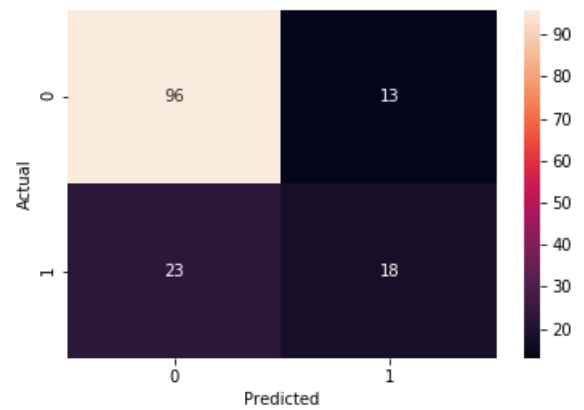
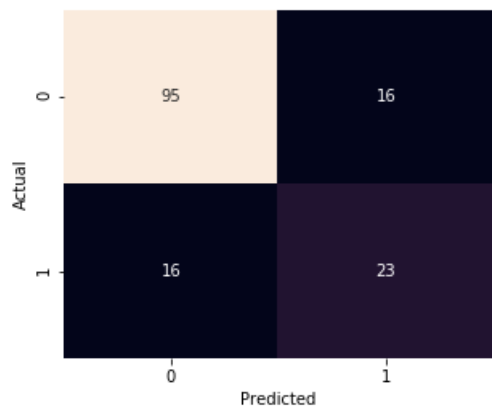
Figure 4.6 shows precision vs recall curves and confusion matrices as subfigures on both the validation and testing sets. From the curves we can observe that when precision is high, the recall is correspondingly low, and at very low precision, the recall is high. High precision means that incorrectly predicting a claim as fraudulent is costly whereas incorrectly predicting the claim as non-fraudulent is not as costly. On the other hand, high recall means that claims that are incorrectly predicted as non-fraudulent are more costly than claims that are incorrectly predicted as fraudulent, that is, predicting fraud when it is not fraud is much better than saying no fraud when there actually is fraud. The confusion matrices of the Random Forest on both the validation and testing sets model contain a summary of the actual and predicted values including correct and incorrect predictions. Based on the validation set from the confusion matrix on the bottom left, the Random Forest model predicted 118 claims correctly. Of this 118, only 23 are fraudulent claims. The model has 32 incorrectly predicted claims. Of this 32 claims, 16 are incorrectly predicted as non-fraudulent on the validation set. On the bottom right confusion matrix we can see that the Random Forest on the testing set correctly predicted 114 claims and from this 114 claims, only 18 are fraudulent. The model has predicted 36 claims incorrectly. Of this 36, 23 are incorrectly predicted as non-fraudulent.

TABLE 4.6: Evaluation metrics of Random Forest on validation and testing sets using ROS

Validation				
Class	Precision	Recall	F1 Score	Support
0	0.85	0.85	0.85	111
1	0.56	0.56	0.56	39
Avg/Total	0.71	0.71	0.71	150
Testing				
Class	Precision	Recall	F1 Score	Support
0	0.82	0.90	0.86	109
1	0.65	0.49	0.56	41
Avg/Total	0.73	0.69	0.71	150



(A) Precision vs Recall for Random Forest on validation set using ROS (B) Precision vs Recall for Random Forest on testing set using ROS



(C) Confusion Matrix of the Random Forest on validation set using ROS (D) Confusion Matrix of the Random Forest on testing set using ROS

FIGURE 4.6: PR curves and Confusion Matrices for the Random Forest model using ROS

4.3 Discussion

The above results clearly show that XGBoost and Random Forest without data augmentation performed better considering their overall average f1 score. This demonstrates the effectiveness of ensemble techniques in achieving higher performance even when there is a problem with class imbalance. The recall score of Random Forest is very low without data augmentation. The Random Forest model, when used with SMOTE, gave good recall, precision and f1 score for both validation and testing sets. Comparing with the models without data augmentation, the precision and recall score of Random Forest model improved in the case where SMOTE and ROS were used. The scores for XGBoost did not improve when SMOTE and ROS were used, however, when compared to results without data augmentation, the XGBoost performed better. From the experimental results we can see that XGBoost is effective on imbalanced data irregardless of the risk of overfitting.

From the literature reviewed, the Random Forest algorithm was more effective than the other models. The recall for Random Forest is low on the original set before data augmentation, this means that most of our fraudulent values are never predicted. The Random Forest does not offer good results when applied on imbalanced data. Its ability is weakened on imbalanced data [58]–[60], because each decision tree in Random Forest has the same weight when the classifiers are combined [59], [61]. Besides the low recall for Random Forest on imbalanced data we have observed that both the Random Forest and XGBoost models as demonstrated by the results offer a better performance even on imbalanced data. We are interested in the fact that the models are better without data augmentation which is impressive as it would be effective on real-world data which is often imbalanced. Random Forest outperforms the XGBoost when the SMOTE and ROS techniques are used.

4.4 Conclusion

In this chapter, we presented and explained the experimental results for both models on the original dataset and when the SMOTE and ROS techniques were used. The results are based on validation and testing sets. We evaluated our models using precision,

recall, f1 score and the confusion matrices. Finally, we discussed what our results really mean and why we achieved the results we have.

Chapter 5

Conclusion

5.1 Introduction

This chapter concludes the study in four sections. The first section contributes to an overall summary of the study, which is followed by a summary of the findings and conclusions derived from them. Following that are the limitations of the study which are followed by recommendations for future research and the conclusion of the overall research work.

5.2 Summary of the Study

This section summarizes the overall study. Recently, it has become possible to turn any kind of data into useful, actionable information including insurance claims data, and a methodology for analyzing such data is being actively researched. In this research work, we aimed to detect fraudulent insurance claims by developing two machine learning models based on labelled insurance claims data downloaded from Kaggle.

With the development of machine learning techniques, insurance companies are attempting to detect fraudulent claims by analyzing customer claims data. The proposed models have relevance for this current trend pattern of fraudulent claims in insurance companies. The proposed models are developed by fulfilling the following four processes: Pre-process the insurance claims data, perform exploratory data analysis, train the XGBoost and Random Forest algorithms and lastly evaluate the trained models using new data which is the validation and testing datasets. Among these processes, some technique has been introduced by this study to solve data imbalance problem

and this technique is called data augmentation techniques.

Based on the experimental results of fraud detection on claims data, it can be concluded that machine learning models are important to consider when trying to fight fraud in insurance companies. The results indicate that sizeable and diverse fraudulent claims are required so that machine learning models can learn as much as they can about what to look out for in fraudulent claims.

The purpose of the developed models is to predict whether an insurance claim is fraudulent or not. The findings of this study will contribute to insurance companies and future research considering that fraud is of great concern. The increase of fraud activities justifies the need for effective machine learning models to fight against it.

5.3 Summary of Findings

This section summarizes the main findings of the research work. Findings show that machine learning models are effective in predicting fraudulent claims. However, we need to gather diverse and large datasets with more examples of fraudulent claims so the models can learn much about how fraudulent claims look like and the differences and similarities between fraudulent and non-fraudulent claims. The XGBoost and Random Forest model performed better, however, each model performed well on a specific dataset. The XGBoost was effective on the original imbalanced dataset in terms of the scores used for evaluation. The Random Forest model performed better than the XGBoost model when the SMOTE technique was used. From the reviewed literature, many studies have concluded that the Random Forest model is the best model, however, in our findings we observed that it is effective but it does not offer better results than the XGBoost with imbalanced data as explained in section 4.3 of chapter 4.

5.4 Limitations of the Study

In this research work, we are interested in fraudulent claims against non-fraudulent claims. The size of non-fraudulent claims is much bigger than fraudulent claims. This type of problem is known as imbalanced class classification. This is a challenge behind

insurance fraud detection because fraudulent claims are far less common as compared to non-fraudulent claims. This made it difficult for the models to learn more about fraudulent claims during training, hence, they are bias towards the fraudulent claims to the extent that it negatively impacts the models' performance when applied to new data. The models normally require a larger sample size and balanced distribution of classes in order to produce significant analytics findings and results.

5.5 Recommendations for Future Studies

This research work presents machine learning models for insurance fraud and a data augmentation approach for data imbalance problem. Future researchers that refer to these findings to conduct further and effective research will be able to build models that will be able to keep up with fraud trends. Insurance Companies that will utilize these models will be able to mitigate fraudulent activities. In this research work we were able to uncover critical information about the kind of data to use in order to build effective models. A few areas emerged as potential future research areas.

As mentioned in section 5.4 the models are predicting more non-fraudulent claims than they should because of lack of enough fraudulent claims in the dataset. To solve this problem, we used SMOTE and ROS techniques to increase the size of fraudulent claims to be the same as the size of non-fraudulent claims, however, the two techniques could not be optimized as extensively as it would be desirable because SMOTE does not consider that nearby examples might belong to different classes when creating synthetic examples. As a result, there may be more class overlap, which will add to the noise. The ROS increased the probability of the models overfitting, since it duplicated the fraudulent claims. Therefore, especially concerning the data, it would be desirable to gather more diverse claims with many examples of fraudulent claims for insurance fraud detection. Balanced data poses a very good performance of models.

5.6 Conclusion

This research work presented two supervised machine learning models, the XGBoost and Random Forest, to predict insurance claims as non-fraudulent or fraudulent. On our experimental work we covered a number of processes that led to the results we

obtained. The processes include, exploratory data analysis, data pre-processing which covers handling of missing values, feature selection and conversion of categorical variables into numeric variables. We also looked at the prediction results where we trained, validated and tested the XGBoost and Random Forest models. The dataset of insurance claims was downloaded from the Kaggle website. The experimental work was performed using Jupyter Notebook in Python. Because of data imbalance observed from the exploratory data analysis, we increased the data by increasing the fraudulent claims using data augmentation techniques namely, SMOTE and ROS. Random Forest model performed better in predicting whether claims are fraudulent or not under both imbalanced and balanced data but has a poor recall score on imbalanced data. The XGBoost performed better on imbalanced data but the performance did not improve with the use of SMOTE and ROS techniques. The scores for the evaluation metrics increased when the data is increased. These results also shows that models play an important role in predicting fraudulent claims. In concluding this work, future work will focus on collecting a large size and diverse data for the models to learn more about fraudulent patterns and give best accurate results.

Bibliography

- [1] A. A. A. Salem and P. S. E. Wan, “The common types of health insurance fraud among insured and healthcare provider”, *World Journal of Pharmaceutical Research*, Scientific Journal Impact Factor, pp. 558–594, 2019.
- [2] J. West, M. Bhattacharya, and R. Islam, “Intelligent financial fraud detection practices: An investigation”, in *International Conference on Security and Privacy in Communication Networks*, Springer, 2014, pp. 186–203.
- [3] R. Roy and K. T. George, “Detecting insurance claims fraud using machine learning techniques”, in *2017 International Conference on Circuit, Power and Computing Technologies (ICCPCT)*, IEEE, 2017, pp. 1–6.
- [4] M. Sumalatha and M. Prabha, “Medicclaim fraud detection and management using predictive analytics”, in *2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE)*, IEEE, 2019, pp. 517–522.
- [5] N. Malini and M. Pushpa, “Analysis on credit card fraud identification techniques based on knn and outlier detection”, in *2017 Third International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB)*, IEEE, 2017, pp. 255–258.
- [6] R. Bhowmik, “Detecting auto insurance fraud by data mining techniques”, *Journal of Emerging Trends in Computing and Information Sciences*, Citeseer, vol. 2, no. 4, pp. 156–162, 2011.
- [7] V. Rawte and G. Anuradha, “Fraud detection in health insurance using data mining techniques”, in *2015 International Conference on Communication, Information & Computing Technology (ICCICT)*, IEEE, 2015, pp. 1–5.
- [8] K. Dineva and T. Atanasova, “Systematic look at machine learning algorithms—advantages, disadvantages and practical applications”, *International Multidisciplinary Scientific GeoConference: SGEM*, Surveying Geology & Mining Ecology Management (SGEM), vol. 20, no. 2.1, pp. 317–324, 2020.

- [9] A. Singh, N. Thakur, and A. Sharma, “A review of supervised machine learning algorithms”, in *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, Ieee, 2016, pp. 1310–1315.
- [10] S. Viaene and G. Dedene, “Insurance fraud: Issues and challenges”, *The Geneva Papers on Risk and Insurance-Issues and Practice*, Springer, vol. 29, no. 2, pp. 313–333, 2004.
- [11] I. I. Institute. (2022). “Background on: Insurance fraud”, [Online]. Available: <https://www.iii.org/article/background-on-insurance-fraud> (Accessed on 05/05/2021).
- [12] B. Itri, Y. Mohamed, Q. Mohammed, and B. Omar, “Performance comparative study of machine learning algorithms for automobile insurance fraud detection”, in *2019 Third International Conference on Intelligent Computing in Data Sciences (ICDS)*, IEEE, 2019, pp. 1–4.
- [13] Y. Li, C. Yan, W. Liu, and M. Li, “Research and application of random forest model in mining automobile insurance fraud”, in *2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, IEEE, 2016, pp. 1756–1761.
- [14] J. West, M. Bhattacharya, and R. Islam, “Intelligent financial fraud detection practices: An investigation”, in *International Conference on Security and Privacy in Communication Networks: 10th International ICST Conference, SecureComm 2014, Beijing, China, September 24-26, 2014, Revised Selected Papers, Part II 10*, Springer, 2015, pp. 186–203.
- [15] V. Faseela and P. Thangam, “A review on health insurance claim fraud detection”, *International Journal of Engineering Research Science (IJOER)*, Academia, vol. 1, pp. 1–3, 2015.
- [16] N. Dhieb, H. Ghazzai, H. Besbes, and Y. Massoud, “A secure ai-driven architecture for automated insurance systems: Fraud detection and risk measurement”, *IEEE Access*, IEEE, vol. 8, pp. 58 546–58 558, 2020.
- [17] S. K. Majhi, S. Bhattacharya, R. Pradhan, and S. Biswal, “Fuzzy clustering using salp swarm algorithm for automobile insurance fraud detection”, *Journal of Intelligent & Fuzzy Systems*, IOS Press, vol. 36, no. 3, pp. 2333–2344, 2019.

- [18] D. Niklas. (2021). “Random forest classifier: A complete guide to how it works in machine learning”, [Online]. Available: <https://builtin.com/data-science/random-forest-algorithm> (Accessed on 02/04/2022).
- [19] T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system”, in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, ACM Digital Library, 2016, pp. 785–794.
- [20] W. Li, Y. Yin, X. Quan, and H. Zhang, “Gene expression value prediction based on xgboost algorithm”, *Frontiers in genetics*, Frontiers Media SA, vol. 10, p. 1077, 2019.
- [21] M. Wang, J. Yu, and Z. Ji, “Credit fraud risk detection based on xgboost-lr hybrid model”, in *Proc. Int. Conf. Electron. Bus.*, Association for Information Systems, vol. 2, 2018, pp. 336–343.
- [22] I. Hanif, “Implementing extreme gradient boosting (xgboost) classifier to improve customer churn prediction”, *Proceedings of the 1st International Conference on Statistics and Analytics, ICSA*, European Union Digital Library (EUDL), pp. 1–20, 2020.
- [23] S. D. Jadhav and H. Channe, “Comparative study of k-nn, naive bayes and decision tree classification techniques”, *International Journal of Science and Research (IJSR)*, Academia, vol. 5, no. 1, pp. 1842–1845, 2016.
- [24] S. Dey, Y. Kumar, S. Saha, and S. Basak, “Forecasting to classification: Predicting the direction of stock market price using xtreme gradient boosting”, *PESIT South Campus*, Working paper, pp. 1–10, 2016.
- [25] S. Wang, S. Liu, J. Zhang, X. Che, Y. Yuan, Z. Wang, and D. Kong, “A new method of diesel fuel brands identification: Smote oversampling combined with xgboost ensemble learning”, *Fuel*, Elsevier, vol. 282, p. 118 848, 2020.
- [26] R. Primartha and B. A. Tama, “Anomaly detection using random forest: A performance revisited”, in *2017 International conference on data and software engineering (ICoDSE)*, IEEE, 2017, pp. 1–6.
- [27] L. Breiman, “Random forests”, *Machine learning*, Springer, vol. 45, no. 1, pp. 5–32, 2001.
- [28] G. Biau and E. Scornet, “A random forest guided tour”, *Test*, Springer, vol. 25, no. 2, pp. 197–227, 2016.

- [29] P. Ražanskas, A. Verikas, C. Olsson, and P.-A. Viberg, “Predicting blood lactate concentration and oxygen uptake from semg data during fatiguing cycling exercise”, *Sensors*, Multidisciplinary Digital Publishing Institute, vol. 15, no. 8, pp. 20 480–20 500, 2015.
- [30] EDUCBA. (2022). “Random forest vs xgboost”, [Online]. Available: <https://www.educba.com/random-forest-vs-xgboost/> (Accessed on 02/03/2022).
- [31] A. Sethia, R. Patel, and P. Raut, “Data augmentation using generative models for credit card fraud detection”, in *2018 4th International Conference on Computing Communication and Automation (ICCCA)*, IEEE, 2018, pp. 1–6.
- [32] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “Smote: Synthetic minority over-sampling technique”, *Journal of artificial intelligence research*, AI Access Foundation and Morgan Kaufmann Publishers, vol. 16, pp. 321–357, 2002.
- [33] G. E. Batista, R. C. Prati, and M. C. Monard, “A study of the behavior of several methods for balancing machine learning training data”, *ACM SIGKDD explorations newsletter*, ACM New York, NY, USA, vol. 6, no. 1, pp. 20–29, 2004.
- [34] J. Nagidi. (2020). “Best ways to handle imbalanced data in machine learning”, [Online]. Available: <https://dataaspirant.com/handle-imbalanced-data-machine-learning/> (Accessed on 01/06/2022).
- [35] D.-H. Lee, J.-K. Yang, C.-H. Lee, and K.-J. Kim, “A data-driven approach to selection of critical process steps in the semiconductor manufacturing process considering missing and imbalanced data”, *Journal of Manufacturing Systems*, Elsevier, vol. 52, pp. 146–156, 2019.
- [36] H. Ma, W. Huang, Y. Jing, C. Yang, L. Han, Y. Dong, H. Ye, Y. Shi, Q. Zheng, L. Liu, *et al.*, “Integrating growth and environmental parameters to discriminate powdery mildew and aphid of winter wheat using bi-temporal landsat-8 imagery”, *Remote Sensing*, Multidisciplinary Digital Publishing Institute, vol. 11, no. 7, p. 846, 2019.
- [37] V. Sahayasakila, D. Aishwaryasikhakolli, and V. Yaraswi, “Credit card fraud detection system using smote technique and whale optimization algorithm”, *International Journal of Engineering and Advanced Technology (IJEAT)*, Blue

- Eyes Intelligence Engineering and Sciences Publication (BEIESP), vol. 8, no. 5, pp. 190–192, 2019.
- [38] J. Brownlee. (2021). “Random oversampling and undersampling for imbalanced classification”, [Online]. Available: <https://machinelearningmastery.com/random-oversampling-and-undersampling-for-imbalanced-classification/> (Accessed on 07/23/2021).
- [39] H. Najadat, O. Altit, A. A. Aqouleh, and M. Younes, “Credit card fraud detection based on machine and deep learning”, in *2020 11th International Conference on Information and Communication Systems (ICICS)*, IEEE, 2020, pp. 204–208.
- [40] J. Sheard, “Quantitative data analysis”, in *Research Methods: Information, Systems, and Contexts, Second Edition*, Elsevier, 2018, pp. 429–452.
- [41] S. Sukamolson, “Fundamentals of quantitative research”, *Language Institute Chulalongkorn University*, Bangkok, vol. 1, no. 3, pp. 1–20, 2007.
- [42] Kaggle. (2018). “Auto insurance claims data”, [Online]. Available: <https://www.kaggle.com/bunttyshah/auto-insurance-claims-data> (Accessed on 03/22/2021).
- [43] A. Dreber and M. Johannesson, “Gender differences in deception”, *Economics Letters*, Elsevier, vol. 99, no. 1, pp. 197–199, 2008.
- [44] J. Childs, “Gender differences in lying”, *Economics Letters*, Elsevier, vol. 114, no. 2, pp. 147–149, 2012.
- [45] L. Friesen and L. Gangadharan, “Individual level evidence of dishonesty and the gender effect”, *Economics Letters*, Elsevier, vol. 117, no. 3, pp. 624–626, 2012.
- [46] S. Tennyson, “Economic institutions and individual ethics: A study of consumer attitudes toward insurance fraud”, *Journal of Economic Behavior & Organization*, Elsevier, vol. 32, no. 2, pp. 247–265, 1997.
- [47] S. Tennyson, “Insurance experience and consumers’ attitudes toward insurance fraud”, *Journal of Insurance Regulation*, ABI/INFORM Global, vol. 21, no. 2, pp. 35–56, 2002.
- [48] D. H. Dean, “Perceptions of the ethicality of consumer insurance claim fraud”, *Journal of Business Ethics*, Springer, vol. 54, no. 1, pp. 67–79, 2004.

- [49] A. D. Miyazaki, “Perceived ethicality of insurance claim fraud: Do higher deductibles lead to lower ethical standards?”, *Journal of business ethics*, Springer, vol. 87, no. 4, pp. 589–598, 2009.
- [50] S. Zhang, C. Zhang, and Q. Yang, “Data preparation for data mining”, *Applied artificial intelligence*, Taylor & Francis, vol. 17, no. 5-6, pp. 375–381, 2003.
- [51] S. B. Kotsiantis, D. Kanellopoulos, and P. E. Pintelas, “Data preprocessing for supervised learning”, *International journal of computer science*, Citeseer, vol. 1, no. 2, pp. 111–117, 2006.
- [52] J. T. Hancock and T. M. Khoshgoftaar, “Survey on categorical data for neural networks”, *Journal of Big Data*, SpringerOpen, vol. 7, no. 1, pp. 1–41, 2020.
- [53] Y. Peng, G. Kou, A. Sabatka, J. Matza, Z. Chen, D. Khazanchi, and Y. Shi, “Application of classification methods to individual disability income insurance fraud detection”, in *International Conference on Computational Science*, Springer, 2007, pp. 852–858.
- [54] M. A. Hall, “Correlation-based feature selection for machine learning”, Ph.D. dissertation, The University of Waikato, 1999.
- [55] V. Iosifidis and E. Ntoutsi, “Dealing with bias via data augmentation in supervised learning scenarios”, *Jo Bates Paul D. Clough Robert Jäschke*, CEUR Workshop Proceedings, vol. 24, p. 11, 2018.
- [56] V. Nasteski, “An overview of the supervised machine learning methods”, *Computer Science*, Horizon Research Publishing(HRPUB), vol. 4, pp. 51–62, 2017.
- [57] NVIDIA. (2022). “Xgboost”, [Online]. Available: <https://www.nvidia.com/en-us/glossary/data-science/xgboost/> (Accessed on 01/04/2022).
- [58] B. Xia, H. Jiang, H. Liu, and D. Yi, “A novel hepatocellular carcinoma image classification method based on voting ranking random forests”, *Computational and mathematical methods in medicine*, Hindawi, pp. 1–8, 2016.
- [59] M. E. H. Daho, N. Settouti, M. E. A. Lazouni, and M. E. A. Chikh, “Weighted vote for trees aggregation in random forest”, in *2014 International Conference on Multimedia Computing and Systems (ICMCS)*, IEEE, 2014, pp. 438–443.
- [60] T. Perry, M. Bader-El-Den, and S. Cooper, “Imbalanced classification using genetically optimized cost sensitive classifiers”, in *2015 IEEE Congress on Evolutionary Computation (CEC)*, IEEE, 2015, pp. 680–687.

- [61] C. A. Ronao and S.-B. Cho, “Random forests with weighted voting for anomalous query access detection in relational databases”, in *International Conference on Artificial Intelligence and Soft Computing*, Springer, 2015, pp. 36–48.