Enhancing Fraud Detection in the Automobile Insurance Sector: A Comparative Analysis Utilizing PyCaret

By

Ambarish. N

Final Thesis Report

December 2023

TABLE OF CONTENT

DEDICATION

To my parents, whose unconditional love and unwavering encouragement have been my driving force. Your steadfast belief in my abilities has been my greatest motivation. This work is dedicated to you.

To my cohort members, whose support sustained me during challenging times throughout this thesis. I dedicate this work to you.

To my friends and well-wishers, who have been by my side since day one of this thesis journey. I dedicate this work to you.

This accomplishment is a testament to the support and encouragement I have received from each of you, and for that, I am profoundly grateful.

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to Professor Dr Manoj Jayabalan of LJMU for their invaluable insights and meticulous guidance in navigating the intricacies of creating this thesis. Their expertise and willingness to address my doubts played a pivotal role in shaping this work.

A special acknowledgement is extended to my thesis supervisor, Associate Professor Merlin Arokiamary. Without their unwavering motivation and support, this endeavour would not have been possible. Their guidance has been a beacon throughout this academic journey.

I am thankful to my cohort members, whose collaborative spirit and shared ideas enriched the development of this work. My sincere appreciation goes to my office mates for their assistance in covering my responsibilities during my absence.

In heartfelt gratitude, I extend my thanks to my parents for their enduring emotional and motivational support. Their encouragement has been my constant driving force. I would also like to acknowledge my cousin, whose emotional support and motivational words provided the spirit needed to navigate and complete this challenging process.

This accomplishment is a reflection of the collective support and encouragement I have received from these remarkable individuals, and for that, I am truly grateful.

ABSTRACT

The focus of this research is to improve the effectiveness of detecting vehicle insurance fraud by using machine learning (ML) models. The main objective is to identify the best ML model for fraud detection by comparing multiple models using PyCaret. The research methodology involves encoding numerical data types, implementing data balancing methods, and using feature selection techniques to enhance model efficiency. The original encoded data achieved an impressive accuracy of 98% with XGBoost after applying data balancing. Additionally, the feature-selected data demonstrated a notable accuracy of 94% with Gradient Boosting and 93% with Extra Trees Classifier after data balancing. The findings highlight the effectiveness of using XGBoost with data balancing, which has a substantial impact on fraud detection compared to other ML methods. This research suggests that using XGBoost with data balancing can be a preferred approach for enhanced accuracy and reliability in fraud detection systems within the insurance industry.

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AARP
American Association of Retired Person 14

AdaBoost
Adaptive boosting 54

ADASYN
ADAptive SYNthetic sampling technique 34

AI
Artificial intelligence 16

AML
Automated Machine Learning 38

ANOVA
Analysis of Variance 32

AUC
Area Under the Curve 57

BP
Back Propagation 28

CAIF
Coalition Against Insurance Fraud 14

CART
Classification and Regression Tree..... 51

CatBoost
Categorical Boosting........................... 55

CNN
Condensed Nearest Neighbour Rule... 46
Convolutional Neural Network........... 17

CNN-LSTM
Convolutional Neural Network - Long-Short Term Memory 30

CPU
Central Processing Unit 86

CV
Computer Vision 16

DA
Data Analytics 16

DC
Dummy Classifier 53

DL
Deep learning 21

DM
Data mining 43

DMSVM
Dual Membership Support Vector Machine 27

DNN
Deep Neural Network..........................35

DT
Decision Tree......................................51

EC
Expected Cost......................................55

EDA
Exploratory Data Analysis 65

EFB
Exclusive Feature Bundling 54

ELM
Extreme Learning Machine 33

ENN
Edited Nearest Neighbours Rule 47

ET
Extremely randomized tree classifier..52

FDS
Fraud Detection System 25

FN
False Negative 56

FP
False Positive......................................56

FPR
False Positive Rate 57

FPS
Fraud Prevention System 25

GB
Gradient boosting classifier.................53

genAI
Generative AI 109

GOSS
Gradient-based one-sided sampling.... 54

GPU
Graphical Processing Unit 86

IQR
Inter-Quartile Range 28

KAPPA
Cohen's Kappa coefficient 58

KNN
K-Nearest Neighbours 50

kRNN
kReverse Nearest Neighbour 28

LASSO
Least Absolute Shrinkage and Selection Operator.. 45

LDA
Latent Dirichlet Allocation 35

Linear Discriminant Analysis 52

LE
Label Encoder 43

LightGBM
Light Gradient Boosting Machine 54

LR
Logistic Regression............................ 50

LSTM
Long-Short Term Memory.................. 17

MAE
Mean Absolute Error........................... 51

MCC
Matthew’s correlation coefficient 59

ML
Machine Learning 15

MLP
Multilayer Perceptron 17

MSE
Mean Squared Error............................ 51

MWMOTE
Majority Weighted Minority Oversampling Technique................. 34

NB
Naive Bayes..51

NCR
Neighbourhood Cleaning Rule............47

NDA
normal discriminant analysis...............52

NIFF
National Insurance Fraud Forum.........14

NLP
Natural Language Processing.............16

OCSVM
One-Class Support Vector Machine 28

OSS
One-Sided Selection............................47

PCA
Principal Component Analysis 52

PCARF-PNN
Principal Component Analysis Random Forest – Potential Nearest Neighbour 32

PCC
Pearson’s Correlation Coefficient 66

Percent Correctly Classified................26

PRIDIT
Principal Component Analysis RIDIT 27

QDA
Quadratic Discriminant Analysis 53

RC
Ridge Classifier 53

RF
Random Forest 51

RFE
Recursive Feature Elimination 29

RNN
Recurrent Neural Network 17

Recursive Neural Network..................17

ROC
Receiver Operating Characteristic Curve 57

ROS

Random Oversampling 47

RPA
Robotic Process Automation............... 34

RUS
Random Under-sampling.................... 46

SBS
Sequential Backward Selection........... 45

Seq2Seq
Sequence-to-sequence......................... 17

SFS
Sequential Forward Selection 45

SHAP
SHapley Additive exPlanations 105

SMOTE
Synthetic Minority Oversampling
Technique 33

SNN
Shallow Neural Network 17

SVM
Support Vector Machine......................52

SWQS
Symmetric Weighted Quantile Sketch.55

TN
True Negative 56

TP
True Positive.......................................56

TPR
True Positive Rate 57

USI
Unique Session ID...............................86

WEKA
Waikato Environment for Knowledge
Analysis 28

XGBoost
Extreme Gradient Boosting.................54

CHAPTER 1:
INTRODUCTION

Insurance or insurance policy is a contract between an insurance company and an individual to be protected from financial loss. The individual buys the policy by paying a fee called a premium. The premium can be paid in different ways yearly once, monthly or even quarterly. The paid premium acts as coverage for any sudden financial loss and covers you from that incident. This insurance protects losses against accidents, theft, medical and so on.

There are two distinct types of insurance, Life insurance and General insurance. Life insurance is provided to an individual. It is financial coverage for the most uncertain part of human life. It offers financial protection to the Life Assured's family in case of unfortunate events like the death or disability of the policyholder. This life insurance can also provide a savings component. The insurance not bound by life insurance is called general insurance like health, automobile, travel, commercial and so on.

Multiple companies provide both types of insurance all these companies will be formed and known as the insurance sector. The insurance sector plays a crucial role in the development of both the financial and economic levels of a country (Feyen et al., 2011). The insurance sector provides crucial support in this modern world (Viaene and Dedene, 2004).

The insurance sector plays a crucial role in the economy by reducing the impact of large losses on firms and households through risk pooling. This results in lower capital requirements for covering losses individually, which encourages more output, investment, innovation, and competition. Additionally, by using risk-based pricing for insurance protection, the sector can influence the behaviour of economic agents and contribute to preventing accidents, improving health outcomes, and increasing efficiency. As financial intermediaries with long-term investment horizons, life insurance companies can provide long-term finance and effective risk management. Finally, the insurance sector can improve the efficiency of other financial segments, such as banking and bond markets.

In the event, an incident happens to the insured – the one who pays the policy to be protected, the insurer – the insurance company that offers the policy, will give the payout or the sum mentioned in the policy to the insured (Artís et al., 2002). The mishap or when the insured is not honest in this process and tries to claim the insured amount erroneously is called insurance fraud. Insurance fraud has adverse effects on all forms of insurance. According to (Viaene and Dedene, 2004) until the late 1980s (Viaene et al., 2002) the insurance fraud phenomenon was unaware and first published his paper about insurance and insurance fraud. The insurance fraud cost US insurance payers $15 billion loss in 1987. In his paper, the effect of insurance fraud has been covered in detail. Then many government councils and bureaus have been formed worldwide to monitor the insurance frauds.

Nowadays, most organizations, private companies, and government agencies have adopted electronic commerce to enhance their productivity and efficiency in trading products or services. This is particularly true in areas such as credit cards, telecommunications, healthcare insurance, automobile insurance, online auctions, and many more (Omar et al., 2018). However, electronic commerce systems are not immune to fraud, as both legitimate users and fraudsters use them, making them more vulnerable to large-scale and systematic fraud.

In recent years, according to a non-profit organisation called Coalition Against Insurance Fraud (CAIF), the U.S has faced a loss of $308.6B as of 2022 in total and as per the American Association of Retired person (AARP) also a non-profit group, in their survey on 2018 found that Medicare fraud alone cost $60B each year. In India, around 60% of insurance fraud has been raised in 2023 in a survey conducted by a third-party company. As per the Association of British Insurers in 2021, there are around 89,000+ cases that have been filed as insurance fraud which cost them 1.1 billion euros and it is getting reduced by 5% per year as they fight against the insurance fraud. Insurance fraud is considered to be an act of criminal obtaining money from an insurer under false pretence (Derrig, 2002). The types of fraud are classified (Derrig, 2002) in his paper, he has written and conducted much research against insurance fraud.

In this study, we are focusing on vehicle or automobile insurance fraud. Fraudulent groups stage fake traffic incidents to collect exaggerated insurance claims. Recent research done by the private sector shows that in a year more than $7.7 billion is being lost for auto fraud insurance. Vehicle fraud insurance is on the rise as the development of vehicles is increased in modern society. There are numerous reasons insurance fraud is being conducted (Viaene and Dedene, 2004). This may be due to a fundamental lack of understanding of insurance fraud. As stated by the National Insurance Fraud Forum (NIFF) ‘‘Insurance fraud means many different things to different people, and therein lies one of the biggest challenges in measuring fraud. There is no universally understood definition of insurance fraud’’. According to a study conducted by India Forensic in 2019, insurance fraud has caused India to lose $6.25 billion. Automobile insurance accounts for almost 90% of this staggering amount. India ranks 10th in terms of gross premiums earned for life insurance and 15th for non-life insurance products(Vyas and Serasiya, 2022).

The insurance sector has been digitalised in these years which helps in monitoring the insurance fraud. Digitalisation has helped to reduce the human error factor and increase productivity and efficiency. Fraud detection and prevention are increased due to machine work. (Abdallah et al., 2016) has analysed the fraud patterns using machine works and devised a model for fraud prevention and fraud detection systems using normal mathematical models which is time-consuming.

To take effective measures against insurance fraud Machine Learning is implemented. Machine Learning (ML) was implemented in insurance sectors in the early 2000s. Machine Learning a subset of artificial intelligence, uses given data to find the desired output using statistical algorithms. Machine Learning is well known for its predictive analysis application. Insurance companies began leveraging machine learning algorithms to analyse large volumes of data and identify patterns indicative of fraudulent activities. Many insurance companies are implementing machine learning for fraud detection to prevent fraudulent claims. There are many types of automobile insurance fraud like staged collusion and exaggerated claims. Around 85% of the insurance sector has implemented a dedicated fraud investigation team along with machine learning tools for insurance fraud Prevention. The implementation of machine learning has produced tremendous results in fraud detection in the insurance sector. The machine learns from the data collected previously and learns from that using the algorithms we provide. Insurance fraud detections are used to be carried out mostly by surveys and expert inspection.
The use of statistical methods to identify fraudulent claims in areas like automobile insurance and healthcare insurance has been extensively studied by academics. While linear approaches have been the traditional method, there is an increasing trend of using machine learning techniques to detect insurance claim fraud. The basic principle of machine learning is the study and construction of a system that can learn from data. Machine learning makes computers behave more intelligently (Roy and George, 2017). Machine learning can be used in many ways like to predict customer churn, sales, fraud, and so on. The ML we are using in this study is to predict fraud. Fraud detection is the ability to detect the fraud in a given data. Machine learning is used to predict fraud in a given data in a faster and more efficient way.

Machine Learning is capable of accurately detecting all claims that are suspected of fraud. It processes data in very short periods and can reveal connections between various factors that may be invisible to human eyes. By continuously reviewing and analysing data, we can anticipate the discovery of new fraud schemes and develop strategies to prevent them from happening in the future. To prevent the losses incurred due to insurance fraud it needs to be found out as soon as possible. The accuracy of the results needs to be good. For the sake of accuracy and the speed of prediction, many different ML models have been devised and implemented so far. Many researches has been conducted to procure various forms of machine learning techniques in recent years for detecting fraud in automobile insurance.

In the field of Data, Machine Learning (ML) is of utmost importance, as it can be used to solve many real-world problems as it happens (S.Patil et al., 2021). Data Analytics (DA) has emerged be an indispensable tool in the data world, where ML is now positioned as an added value along with Natural Language Processing (NLP), Computer Vision (CV), and the never-ending Artificial intelligence (AI) (Imaam et al., 2021). The DA is significantly crucial in the analysis of data. The ML system has significantly replaced the labour and time that humans once invested in a vast scope, thereby lessening expenses and enhancing productivity (Artís et al., 2002).

There are three types of machine learning techniques namely Supervised, Unsupervised and Reinforcement learning (Roy and George, 2017). Supervised learning is the most basic type of machine learning. The machine learning algorithm is trained on labelled data. A predefined target variable is called labelled data. Supervised learning is a simple yet powerful technique when used in the right circumstances. Unsupervised learning works with unlabelled data. This technique is quite often used to figure out the hidden pattern and relation between data that is not visible to human eyes. Reinforcement learning is more widely used because of its self-learning ability. It features an algorithm that improves upon itself and learns from new situations using a trial-and-error method.

Depending on the data types any kind of ML techniques can be used for Fraud detection. Studies have shown that the majority of the machine learning models for fraud detection is supervised learning model (Debener et al., 2023). Our research on fraud detection is based on supervised learning.

Many researchers are exploring the use of deep learning in fraud detection to develop better detection models that are more accurate and computationally efficient. One popular deep learning method is neural networks, which have become increasingly prevalent across various sectors. Neural network methods can effectively replace traditional machine learning methods, offering greater accuracy and speed. (RB and KR, 2021) Deep learning incorporates various neural network methods, including artificial neural networks, convolutional neural networks, autoencoders, recurrent neural networks, and restricted Boltzmann machines. These networks work like the human brain, processing data and making decisions. The main difference between machine learning and neural networks is that neural networks can learn autonomously, much like the human brain. Once a pattern is identified, a neural network can take action more quickly than traditional machine learning methods.

There are several types of neural network methods Multilayer Perceptron (MLP), Convolutional Neural Network (CNN), Recursive Neural Network (RNN), Recurrent Neural Network (RNN), Long-Short Term Memory (LSTM), Sequence-to-sequence (Seq2Seq) and Shallow Neural Network (SNN). The most used neural network models for fraud detection are MLP, CNN, RNN and LSTM. (Claver\'\ia Navarrete and Carrasco Gallego, 2021) has provided detailed information about neural networks and the latest models used for fraud detection.

1.1 Background of the study

In machine learning, the predominant research types are Quantitative and Qualitative methods. The quantitative deals with numbers and statistics and the qualitative deals with words and meanings when collecting and analysing data. This research is based on quantitative research methods that deal with numbers. Based on the data, the type of research can be changed. Many different kinds of comparative studies have been conducted on fraud detection problems. The commonly used dataset is automobile insurance fraud detection (Gupta et al., 2012; Roy and George, 2017; Badriyah et al., 2018; Harjai et al., 2019; Itri et al., 2019; S.Patil et al., 2021; Xia et al., 2022; Singhal et al., 2023) has been used for this research.

After thorough research, multiple different ML algorithms have been devised for prediction analysis. In the era of modern digitalisation, many State-of-the-art ML algorithm models have been created. However, not all the models that have been developed will assist in fraud detection. Fraud detection is a classification problem. The classification problems are a type of supervised learning.

In all these comparative studies (Gepp et al., 2012; Panigrahi and Palkar, 2018; Itri et al., 2019; Agrawal and Panigrahi, 2023; Singhal et al., 2023), many different kinds of machine learning models have been implemented and hybrid models (Kotb and Ming, 2021; Alamri and Ykhlef, 2022) have been created to acquire accuracy on fraud prediction. The accuracy resulting in this research is comparatively good based on the research method they conducted.

Machine learning models (Viaene et al., 2002; West and Bhattacharya, 2016; Patel and Subudhi, 2019; Wei et al., 2020; Aslam et al., 2022) like Logistic Regression (LR), Support Vector Machine (SVM), Naïve Bayes (NB), Decision Tree (DT), Random Forest (RF) and other ML models are used for comparative study and the evaluation parameter is set for accuracy in all these research. The accuracy gained increased in all this research according to the pre-processing techniques and ML model used. In recent research, boosting techniques have gained popularity because of their efficiency which can be used by both balanced and imbalanced data and computational power. The accuracy in all this research is at the average of 90%. In one particular research (Dhieb et al., 2019) using a boosting model gained an accuracy of 99%.

In this research, we have implemented PyCaret an emerging Python module that contains various ML models and evaluation metrics. This PyCaret will be used for the comparative research because of its versatile and low coding nature. The ML in fraud detection is evolving along with multiple new models that have been created. These facts make auto insurance fraud detection studies more meaningful and essential. The actual extent and cost of insurance fraud remain hard to quantify with precision.

In conclusion, insurance fraud creates a significant economic issue for insurance companies and national economies. So, there is a need to look into this problem (Viaene and Dedene, 2004).

1.2 Problem Statement

The main purpose of this study is to figure out the best ML model for automobile insurance fraud detection based on the different evaluation metrics. (Dhieb et al., 2019) his research provided a model with 99% accuracy with different data without any data balancing technique whereas (Aslam et al., 2022) his research gained an accuracy of 95%. In this research the ML models used are different.

Through our research, we have taken all the necessary steps to compare different evaluation parameters and determine the best model for fraud detection. Implementing the best model can help in reducing loss and promoting economic stability. Often, the data provided for analysis is imbalanced. Thus, our study aims to identify the most effective data-balancing technique for fraud detection.

In recent years, the global insurance market has been significantly influenced by the changing modern world. This has resulted in a broader range of insurable items, encompassing various valuable products that individuals deem worthy of protection. The fundamental principle of insurance lies in its ability to offer a safety net against unforeseen circumstances, such as accidents or damages. In these cases, the insured amount functions as a means to relieve immediate financial burdens. However, the insurance claims process poses a formidable challenge due to its intricate nature, which is further aggravated by the widespread occurrence of fraudulent activities, where individuals other than legitimate victims attempt to claim insurance payouts, leading to what is commonly referred to as insurance fraud. The machine can identify the fraud immediately once the input has been given to the trained model. Thus, it prevents fraudulent payouts and insurance claims which in turn improves the safety level of the insurance sector.

This widespread issue extends across boundaries, and effectively monitoring and assessing the authenticity of each claim manually poses a significant challenge. To address this concern, the integration of automated systems has emerged as a potential solution. These systems have the ability along with the capability to learn from historical data patterns and input information, thereby enhancing the efficiency of fraud identification and prevention. This paradigm shift has propelled various machine learning (ML) algorithms into the spotlight, offering promise in predicting and pre-empting instances of insurance fraud. However, it is crucial to note that the effectiveness and precision of these algorithms can significantly vary.

This research paper undertakes a methodological and comprehensive comparative study of diverse ML algorithms, meticulously examining their merits and limitations. The ultimate objective of this exploration is to ascertain the most adept algorithm, guided by predefined criteria for evaluation. The actual extent and cost of insurance fraud remain hard to quantify with precision.

Through such a rigorous inquiry, this study aims to contribute to the advancement of fraud detection techniques within the domain of insurance.

1.3 Aim and Objective

This thesis aims to identify an optimal machine learning model that can efficiently and affordably address insurance fraud in the automobile sector. The aim of implementing fraud detection in an insurance claim system is to enhance the efficiency, accuracy, and reliability of the claims processing process by developing and integrating robust mechanisms that can identify and prevent fraudulent activities. This involves leveraging advanced technologies and analytical tools to create a secure and trustworthy insurance ecosystem, ultimately minimizing financial losses and maintaining the integrity of the insurance industry.

Machine learning has multiple different algorithms that are used for appropriate circumstances. In previous research conducted some steps for data preprocessing, balance, and evaluation are not included. This research uses PyCaret, a low-code Python model which has all these functions that help in identifying the best model for fraud prediction.

The objectives of this study are

- To compare and analyse various machine learning methods and approaches for automobile insurance fraud detection.
    - To achieve this objective, we can use the PyCaret model. PyCaret integrates many machine learning models and provides output based on the evaluation criteria. It also provides a detailed comparison and analysis of the models. Moreover, the provided model can be further evaluated and tuned to obtain better results.
- To determine the best Machine Learning (ML) model for fraud detection in automobile insurance by emphasizing the advantages and limitations.
    - PyCaret helps us achieve the best machine learning model by comparing it with other models. The dataset provided is transformed, features are selected and resampled to ensure optimal performance of the model. It is important to evaluate the model to determine its limitations. The proposed model can be studied further to identify its advantages and disadvantages.
- To evaluate the model's performance by conducting assessments and evaluations.
    - The evaluation was performed using a pre-determined set of algorithms. The best model suggested by PyCaret has already been evaluated using the metrics set. To gain a more in-depth understanding of the results obtained from PyCaret, further analysis can be carried out to better the model's performance.

By achieving these objectives, the insurance industry can aim to create a secure and reliable claims processing system that benefits both insurers and policyholders while deterring fraudulent activities.

1.4 Scope of the Study

Fraud prediction in any sector is very important. Our study aims the fraud prediction on automobile insurance fraud. There is no specific set of rules to obtain information for insurance. Every other insurance company has their way of gathering data on insurance and has indicators for fraud prediction. The historical data used for this research has its limitations on features as this data is gathered in the years 1994-96.

The dataset is composed of 15,000+ entries with 32 distinct variables and 1 target variable which holds mixed values of both qualitative and quantitative framework. To make the dataset compatible with the chosen ML algorithm, we transform the dataset into a quantitative framework which is binary numbers. Different feature selection algorithms and class balancing techniques are used for a better model. Various research has been conducted using diverse ML models. Multiple models are being compared in a single study to identify the optimal one. It is important to note that this particular algorithm may not be suitable for qualitative methods or results that are not binary.

The scope of the study is to identify the best model for fraud prediction in the automobile insurance sector using existing machine learning models. These models that are combined will take more time to code, validate and compare the outputs. To reform this huge process in an easy way the PyCaret is used. PyCaret reduces programming time which results in faster computation and less human error for the accuracy of the model.

The growth in the insurance sector might have changed the features of insurance information that differ from the one in this study, the huge amount of data these days needs to be carried using cloud services. Deep learning (DL) and Artificial intelligence (AI) have seen rapid growth and many new types of algorithms namely neural networks are implemented in financial sectors for prediction analysis. These methods will be costly to utilise and to handle huge insurance data the hardware and software are to be created specifically for that.

1.5 Significance of the study

The insurance sector plays a vital role in the financial and economic stability and growth of a country. The fraud claims happening in this insurance sector directly affects the growth of the country’s economy and individual person is affected by the aftereffect of inflation. When a financial institution gets affected by these kinds of acts, to stabilize their loss and not affect the legit insurers the institution raises the price of their products/policies which affects the legit customers. To break this cycle, fraud insurance claims need to be identified at the earliest.

The domain of insurance has assumed an indispensable role, encompassing various sectors in this world. Among these diverse categories, vehicle insurance emerges as one of the most frequently used insurance, inevitably entailing claims that encompass instances of fraudulent activities. The escalating incidence of fraudulent claims has cast a shadow over the insurance landscape, a phenomenon that shows no sign of abating. To comprehend the mechanics underlying these fraudulent occurrences and discern the patterns they exhibit, machine learning (ML) techniques have been strategically employed.

By gaining insights into the distinct categories of insurance scams and deciphering their operational intricacies, a prospect arises for the mitigation of fraud claims, particularly those concerning orchestrated collisions, exaggerated claims, and vehicular theft incidents, among others. The pursuit of this objective has fostered a body of research that delves into these dimensions, with the intent of unravelling the concealed patterns underlying such fraud claims, facilitated through the application of diverse models. This study seeks to contribute by undertaking a comparative evaluation of multiple ML algorithms, thereby furnishing an avenue to identify an apt ML model.

The essence of this research endeavour lies in its potential to furnish the insurance industry with a potent mechanism for mitigating prospective losses. While individual algorithms have been the focus of previous research endeavours, a comprehensive evaluation encompassing the entire spectrum of algorithms remains relatively unexplored. Through such a comprehensive approach, this study aims to elucidate the operational mechanisms, facilitating model selection based on parameters such as accuracy, precision, and F-1 score.

- The optimal Machine Learning model should be adopted in the insurance sector for fraud detection.
- Adaptation of the ML model will help in reducing the loss not only in the vehicle insurance industry but even in other industries.

1.6 Structure of the Study

In this section, we will understand the detail and understanding of what the other chapters in the research entail about

- Chapter 2 of the report provides a literature review of related work and techniques. The author has reviewed various papers to understand how to conduct comparative studies for fraud detection. These papers include topics like machine learning, boosting algorithms, feature selection and PyCaret. The section on machine learning explains how it was used to detect fraud and expedite the process, as well as how it was implemented for research. The boosting algorithm section describes the different types of boosting algorithms in machine learning and how they differ from traditional models, along with results from research papers. The reviewed papers in the feature sampling section help readers understand the importance of feature selection and explain the sampling methods used for machine learning. Finally, the section on PyCaret explains the usefulness of this module compared to other ML models, and how Deep Learning (DL) and Artificial Intelligence (AI) have advanced in the insurance sector for fraud detection.
- In Chapter 3, the report details the workflow of the research. The author has explained the study's dataset in detail and presented a flowchart framework for each section of the research. This section provides valuable information on the ML models used in the study, as well as a detailed explanation of the data balancing technique used. Additionally, the author has provided clear and concise explanations of the evaluation metrics employed in this research.
- In Chapter 4, the information gathered from the dataset is visually presented. The EDA, an important process in data analytics has been conducted to visually understand the data. The process of data gathering has been provided. The features within the dataset are then extracted, profiled and explored. Preprocessing steps have been explained along with data balancing and feature selection from the dataset has been provided.
- The chapter 5 provides detailed information on the results gathered after the model implementation step. The evaluation parameters have been carried out. The models created based on the types of data balancing techniques are explained in each section. A detailed explanation of model comparison has been provided based on the results gathered.

- Chapter 6 presents a conclusive summary of the thesis, including its contributions to knowledge, an evaluation of the framework, and a discussion on imperative future work.

1.7 Summary

In this chapter, details about insurance have been provided along with the introduction of machine learning in the insurance sector, the impact of machine learning that’s been implemented to reduce insurance fraud and details about types of machine learning has been provided. The background study on insurance fraud provides valuable insights into the importance of detecting fraud and the motivation behind researching to find the best possible model to tackle it.

The clear and precise aim and objective of this research help to define the scope of the study, while also shedding light on the significance of this study in changing the economic stability and the use of fraud detection in the finance sector. Additionally, the thesis framework outline has been presented to provide a better understanding of the study's structure and organization.

CHAPTER 2:
LITERATURE REVIEW

2.1 Introduction

In this section, we have reviewed many research papers and journals and their works2 that helped out in deriving our research on detailed analysis of insurance fraud, a classification model. Many research studies have extensively investigated the problem of car insurance fraud, including a thorough comparative study and a deep dive into concepts such as artificial intelligence (AI) and Deep Learning (DL). Traditional machine learning models like Logistic regression [LR], Support Vector Method [SVM], Decision Tree [DT], Random Forest (RF), Naïve Bayes (NB) and many other models have been researched. Many different new types of Machine Learning models have been created and tested extensively in these papers. We have adopted a new ML model that is widely popular these days for comparison study called PyCaret. In this study, instead of using the traditional /manual way of machine learning programming, we are using PyCaret which is an AUTO-ML low-code open-access Python library and have reviewed papers for solving data imbalance and techniques to deal with data imbalance.

2.2 Machine Learning in Fraud Detection

Fraud detection is a problem of classification (Gupta et al., 2019). To effectively handle this problem, we need to develop a model that can accurately classify cases as either legitimate or fraudulent by analysing the characteristics present in the dataset. (Derrig, 2002) in his paper about insurance fraud mentioned that the major problem in the US in the 21st century was insurance fraud which has evolved from the insurance fraud in the 19th century. He classified insurance fraud into 8 scenarios which will assist in finding the fraud type. They also explained the claiming process using pre-data mining by sorting them into bins, which helps in filtering the fraud and non-fraud. (Abdallah et al., 2016) To prevent these frauds a Fraud Prevention System (FPS) and a Fraud Detection System (FDS) have been created and explained how technology helps. The prevention system acts within the system like a firewall whereas the detection system uses Machine Learning (ML) models like supervised, unsupervised and semi-supervised, the classification and regression algorithms are under supervised learning models, whereas clustering and reduction algorithms are under unsupervised learning. (Viaene et al., 2007) In their paper, similar to (Derrig, 2002), mentioned that Insurance companies meticulously investigate the input data. They have opted for a threefold approach, dividing the data into policy issuance, claim handling, and damage evaluation. After gathering the data, then proceed to investigate and negotiate the insurance claims. The collected data has led to two distinct outcomes. All these have information has proved that Machine learning can increase the accuracy of fraud detection.

2.3 Machine Learning Implementations.

There are 3 types of machine learning methods known widely Supervised, unsupervised and reinforcement learning. Classification and Regression are types of supervised learning, clustering method deals with unsupervised learning.

Machine Learning has successfully tackled the tedious task of detecting data fraud. As emphasized by (Roy and George, 2017), machine learning techniques can effectively conquer the obstacles of data analysis. By utilizing machine learning, the investigation of intricate domains of knowledge can be simplified and made more efficient. The approach is simple: establish a feedback loop that samples data to generate output, and then provides feedback to the system along with new data for testing. This considerably diminishes the need for manual labour and streamlines the process. The system comprises different machine learning algorithms, including Logistic Regression, Naïve Bayes, Decision Tree, and many more, which can be programmed to meet specific requirements.

(Viaene et al., 2002) in this paper conducted a comparative study with several modern binary classifications for insurance fraud detection. He proposed to examine the models using the Percent Correctly Classified (PCC) and the mean area under the receiver curve (AUC-ROC) based on several experiments. The modern ML models are used and experimented with the values are verified using evaluation metrics like confusion matrix and ROC curves. The comparison of AUC-ROC curves for the different ML models resulted in the C4.5 DT classifier algorithm being a superior performance model.

(Rukhsar et al., 2022) in his paper also did a comparative study using different modern classification algorithms for insurance fraud prediction. The research has its result focused on the accuracy of the model. The data he used had both categorical and numerical data types. He converted the non-categorical data into categorical data by utilizing the one-hot encoding method. The one-hot encoding method also has its drawbacks such as when it is used the number of features in a dataset will be increased exponentially, which results in decreased performance and increased computational time. As a result, based on the evaluation metrics, DT has performed well followed by AdaBoost.

(Abdallah et al., 2016) paper discusses the utilization of machine learning for fraud detection, utilizing three distinct learning methods. Supervised learning has been the most commonly used method for detecting fraud, while unsupervised learning is not frequently used.

Research on both supervised and unsupervised methods for fraud detection has been conducted (Yaram, 2016) using different datasets. In this study, the vehicle insurance dataset was used for supervised and Twitter data for unsupervised as the insurance dataset can only be used for supervised. The result of this study for supervised learning for fraud detection ends with Random Forest having 90% accuracy. Creating hybrid models for fraud detection has been studied in many types of research similarly (Tao et al., 2012) have devised a hybrid model called Dual Membership Support Vector Machine (DMSVM) using fuzzy logic. This model is created mainly to stop recognising true insurance claims as insurance fraud. The research has used balance data that are based on insurance data from Beijing. That has 400 fraud and 400 no-fraud claims. Utilising the proposed model, the overall performance is collectively higher compared to other ML models used by achieving over 90%. The model proposed has better results compared to neural networks

(Debener et al., 2023) conducted a comprehensive study on fraud detection using both supervised and unsupervised learning. Since most of the fraud detection problems are based on supervised classification problems the research on unsupervised fraud detection has not been conducted for the most part because rather than anomaly detection, they are used for normal observations. He utilised the Isolation Forest method for unsupervised learning and multiple ML models for supervised learning. The test results revealed that the Isolation Forest method and XGBoost from other ML models were effective in detecting fraud individually.

2.4 Related Paper with Corresponding Dataset

The dataset used for this research has been utilised in several other research for fraud detection. The data is gathered from insurance fraud claims from the year 1994-96.

The dataset was first used (Brockett et al., 2002) for his research on insurance fraud claim detection. In his research, he has provided detailed information on how to conduct fraud detection using a statistical approach. This process gives rise to the unclassified observations for the two groups (fraud versus nonfraud) are generally of unknown parametric form. For his research utilized the Principal Component Analysis RIDIT (PRIDIT) method to find fraud. RIDIT is a statistical method to compare two or more qualitative data and score them. This method was used before ML was implemented in fraud detection.

This dataset is the most used dataset for fraud detection that is available with high-class imbalance. (Rai et al., 2018) used the same dataset for fraud detection. This paper used an oversampling method called Majority Weighted Minority Over Sampling Technique (MWMOTE) to handle class imbalance and used 3 different ML algorithms Support Vector Machine (SVM), Decision Tree (DT) and Random Forest (RF). These ML models are highly used for fraud detection. MWMOTE used in this research is to identify the borderline samples that are usually missed by other classifiers like SMOTE, Borderline-SMOTE and ADASYN. For data transformation, one-hot encoding and binary encoding are used. In their approach based on accuracy, Random Forest has performed well. The feature selection method is not used in this research.

As machine learning evolves, new algorithms are being developed to detect fraud. In a study (Badriyah et al., 2018) two different methods were used: Nearest Neighbour (distance and density based) and statistic method (IQR - Inter-Quartile Range). They also compared the results of the One-Class Support Vector Machine (OCSVM) (DeBarr and Wechsler, 2013) method with the other two methods, which showed that the distance-based method had a 99% accuracy rate. OCSVM differs from SVM in that it is trained on only one class, building a boundary that separates it from the rest of the feature space. The feature selection process was used in this study, using the parameters in the Waikato Environment for Knowledge Analysis (WEKA) tools that use visualization and algorithms for data analysis and predictive analysis. Out of 33 attributes in the dataset, only 7 attributes were selected based on the feature selection process.

Using hybrid methods such as OCSVM and kReverse Nearest Neighbour (kRNN) a model derived from KNN in his research (Sundarkumar and Ravi, 2015) for his insurance fraud detection. Instead of using the data transformation technique to transform categorical values into numerical values, he created a new column for date-time values and dropped the numerical values. The research has been carried out using categorical values. Based on that a total of 25 features have been used which has resulted in an accuracy score of 91%.

Frauds can be detected based on the cost of a product. Researchers have conducted studies with the main interest of cost saving. (Phua et al., 2004) utilized this method and dataset for fraud detection. In their study, they used Back Propagation (BP), Naïve Bayes (NB), and C4.5 algorithms for detection. Several classifier methods such as stacking-bagging, a combined method of stacking and bagging, sampling, and partitioning were used with these algorithms. Since the data had more features, researchers created three new attributes along with the previous features. However, no feature selection method was used. The research focused on cost-saving methods, and as a result, the stacking-bagging method had a success rate of over 60% with higher cost savings.

(DeBarr and Wechsler, 2013) investigated fraud detection based on cost saving. They addressed the class imbalance problem by using a cost-sensitive learning method based on Random Forest. Instead of using a feature selection method, they utilized OCSVM along with reputation features to partition the large dataset into two smaller partitions. Their research resulted in a 13.6% cost-saving compared to other methods.

In research by (Harjai et al., 2019), WEKA software has been used to build the machine learning model. The researcher has used the SMOTE technique as a data balancing technique and built a model with Random Forest. Based on the formal research after data balancing the proposed model has been compared with previous models and resulted in 99% accuracy. Using a similar method by utilising WEKA, the same dataset has been used (Itri et al., 2019). The number of features has been reduced to 19 using the feature selection methods. The research has been conducted in two phases before and after feature selection. Multiple ML models are used for comparison.

The faster you find the insurance fraud the better the reputation of the company, focusing on computational speed for fraud detection (Tongesai et al., 2022) conducted a comparative study using the same dataset. For the feature selection method Recursive Feature Elimination (RFE), Pearson Correlation, and Chi-Square methods are utilised and one-hot encoding method is used for data transformation. No Data balancing technique has been conducted. The processed data has been used in different ML models for comparison. Which XGBoost has performed overall and the computational speed for XGBoost achieved 8.6 secs where the highest time is 263.3 sec for KNN. This proves that boost algorithms have higher computational speeds compared to other ML models.

(Alrais, 2022) in his research conducted for fraud insurance using the dataset. Normal research where no feature selection method or class imbalance techniques are used. They used the entire data for classification and implemented ML models. As a result, a 98.6% accuracy rate is achieved after fine-tuning the Random Forest model.

Feature selection in a dataset is more important as it reduces the error values and increases accuracy and machine learning time. This has been proven by (Aslam et al., 2022) in their comparative study for insurance fraud detection in the automobile sector. For his research, the Boruta algorithm was used for feature selection which is one of the wrapper methods. Boruta algorithm is used to identify the most significant features from the given data. Out of 33 original features, using Boruta only 9 features have been selected. This processed data is used for the

ML model for further comparison. The resulting comparison is that the SVM model has 94% accuracy. Using a boosting algorithm here could have had better results. In a recent paper by (Maina et al., 2023) for fraud detection in insurance data, utilised the same data along with a dataset obtained from Kenya country, they did a comparative analysis with different ML models. For class imbalance, they used the SMOTE technique and for feature selection, they used a filter method called the chi-squared test. The models used were Logistic Regression, Random Forest, K-Nearest Neighbour, Light Gradient Boosting and Extreme Gradient Boosting. The results of the analysis are that XGBoost had an accuracy rate of 95%. We used the SMOTE technique as many research suggested that for fraud prediction problems SMOTE will be best for dealing with class imbalance.

The neural network is significantly more advanced than traditional machine learning, programming a neural network is complex and costly to use. Neural networks come from the Deep Learning banner. So most of the research is conducted using machine learning techniques. (Xu et al., 2011) in his paper used a neural network-based method for fraud detection. For this new method, they used the entire dataset and partitioned it according to the need. His proposed method has good results compared to a single neural network classifier. (Xia et al., 2022) used the same insurance fraud dataset for a different neural network-based method for fraud detection. He proposed a new hybrid model combined for both feature extraction and fraud prediction called Convolutional Neural Network - Long-Short Term Memory (CNN-LSTM) when compared with other ML models neural network model outperforms the overall.

2.5 Boosting Algorithms in Machine Learning

Boosting algorithms in machine learning has played a significantly important role in getting more accurate results. boosting models are mostly derived from the Random Forest method. In many research for fraud detection, boosting algorithms have been implemented to have a better model.

A study on auto insurance big data (Hanafy and Ming, 2021), mentioned that incorporating machine learning can enhance the understanding and analysis of data, leading to improved efficiency. The study conducted a comparison of machine learning models using various testing methods, including Confusion Matrix and Kappa. The results indicated that Random Forest (RF) and Decision Tree (DT) were the most suitable models. With an 86% accuracy rate, RF was the best fit whereas the XGBoost has the lowest performance with 65%.

In a paper published by (Tongesai et al., 2022), the advantages of machine learning were discussed, including cost and energy savings, fraud prevention, and increased profitability. To achieve these objectives, traditional algorithms such as K-Nearest Neighbour (KNN), Support Vector Machine (SVM), Logistic Regression (LR), and Naïve Bayes (NB) were tested alongside the boosting model Extreme-Gradient Boosting (XGBoost). The XGBoost model was found to be the most effective algorithm after rigorous testing with an accuracy rate of 71%. Similarly, in a paper by (Singhal et al., 2023), a comparison study for the efficiency and verifiability of machine learning algorithms has been carried out with similar ML models. Just as (Nur Prasasti et al., 2020) did for the imbalanced dataset in their research SMOTE technique has been used to get balanced data. A label encoder was used for data transformation. Based on the comparison analysis, XGBoost has proven to be the most suitable option with a 90% accuracy rate. In a paper (Dhieb et al., 2019), an extensive study was conducted using XGBoost along with other traditional ML models to prove that among Gradient Boosting types, XGBoost provides high performance in Accuracy (99%), Precision (99%), and F1 Score(99%), while Naïve Bayes consumes less time (155ms) in training compared to other models and boosting algorithms.

Boosting models are utilized in various fields for different tasks. Depending on the requirements, various types of Boosting models can be employed. Extreme Gradient Boost has been used more frequently in fraud detection because of its better performance. The research (Hancock and Khoshgoftaar, 2021), employed different types of gradient-boosting methods along with Machine Learning models to find fraud detection in Medicare insurance. For his research, he used a one-hot encoder to pre-process the data. After undergoing numerous tests and trials, the CatBoost algorithm has shown exceptional performance in comparison to other classification algorithms with an AUC value of 88%. It has even demonstrated exceptional results during the random under-sampling test.

The comparison study by (Tongesai et al., 2022), using Extreme Gradient boosting along with other machine learning models, resulted in XGBoost with better overall performance based on other evaluation criteria. Like (Hancock and Khoshgoftaar, 2021), (Tongesai et al., 2022) in their paper dealing with categorical values, the same one-hot encoders have been used to encode categorical values for better results. Through this research, he proposed that XGBoost is reliable for training and implementation for fraud detection.

Based on the research conducted, it has been determined that Extreme Gradient Boosting is the most suitable algorithm for detecting fraud when compared to other state-of-the-art machine learning and boosting algorithms. However, in cases where the Boosting algorithm is not included for comparison, different ML models may prove to be more fitting under certain circumstances like the research done by (Panigrahi and Palkar, 2018). To determine the most effective algorithm for detecting fraud, we performed a comparison between the classification algorithm and the Feature Selection algorithm. We utilized tree-based, L-1-based, and univariate feature algorithms to compare with classification models such as K-Nearest Neighbour (KNN), Random Forest (RF), Naïve Bayes (NB), and Decision Tree (DT). After analysing these models, it was found that Decision Tree (95%) had the best score based on recall parameters, while Random Forest (97%) performed better in other classification parameters. Additionally, the L-1-based feature selection yielded better results compared to the other two feature selections.

A research study (Li et al., 2018) on Random forest proposed a new method of random forest algorithm for automobile fraud detection named PCARF-PNN. The PCARF-PNN is abbreviated as Principal Component Analysis Random Forest – Potential Nearest Neighbour. This method is an example of a Multiple-classifier system where more than two classification algorithms are combined to derive a new algorithm. In the study after several experiments, the results showed that the PCARF-PNN produces better accuracy for classification and the variance is in the lower parts.

A comparative research study was done (Itri et al., 2019) on Machine Learning Algorithms. In this research, the traditional ML models are studied to determine the best model that provides good performance and better efficiency and verifiability for fraud prevention. The most commonly used Classification algorithms and boosting algorithms are compared to create a performance-based result. The results are based on feature selection, with AdaBoost (96%) demonstrating strong performance pre-feature selection, and Random Forest (99%) excelling post-feature selection.

2.6 Feature Selection and Sampling Technique in Machine Learning

Feature selection is the process of selecting relevant features from an original set according to certain criteria. Feature selection is an important pre-processing step in Machine Learning. The feature selection process helps in optimizing the running time and learning time of the algorithm by selecting the required variables. There are many different methods used for feature selection. (Cai et al., 2018; Benkessirat and Benblidia, 2019) have provided detailed information about feature selection methods and uses for machine learning. The feature importance score in this process explains the importance of that particular column to our target variable. The most commonly used feature selection method is the Analysis of Variance (ANOVA) test.

(Elssied et al., 2014) in his paper explained the types of feature selection methods and tests using a spam email classification problem. A detailed comparison has been done for before and after feature selection using the SVM model. The ANOVA method is used to skip unimportant attributes from the dataset. The study is designed in a way to understand the importance of feature selection in ML especially when using large datasets. The results of his study show that the time taken for algorithm learning is optimised after the feature selection which in turn had an increased computational speed and performance.

Similarly, (Perangin-Angin and Bachtiar, 2021) utilized a one-way ANOVA F-test for stress detection classification problems in the office work dataset. The feature selection method is utilized in this study for feature reduction to select the related variable out of 35 variables. After the feature selection process out of 35 features only 2 have been selected. (ELM) An Extreme Learning Machine a type of Artificial Neural network model is used for the study. The results are evaluated using classification metrics. As a result, the accuracy and computation time have seen a better improvement compared to before feature selection.

In machine learning the performance and accuracy of a model is highly dependent on a balanced dataset. An imbalance of data in Machine learning might lead to reduced accuracy and model performance thus by increase in computational time(Mary and Claret, 2021). To tackle these many different data balancing techniques have been created for both classification and regression algorithms.

Data balance is crucial in machine learning, if data is imbalanced the outcome of the model is predicated and accuracy will be less. Since a model is selected based on the accuracy delivered by the proposed model. To deal with data balance there are different techniques available for data balance these methods are known as sampling. In sampling, the feature column mentioned with be distributed with values to equalise both the major and minor sides.

Research (Nur Prasasti et al., 2020), has utilised the data balancing technique for a classification problem. A highly imbalanced dataset has been used to understand how the balancing techniques perform. The evaluation parameter has been configured to assess the accuracy. Based on the parameter, Random Forest (RF) provides better results. In his research, the Synthetic Minority Oversampling Technique (SMOTE) is used to correct imbalanced data. The balanced data is then applied to commonly used Machine Learning algorithms such as Multilayer Perceptron (MLP), Decision Tree (DT), and Random Forest (RF). The Random Forest (98% for accuracy) algorithm produces better results than other classification algorithms in Machine Learning, based on specific parameters.

(Subudhi and Panigrahi, 2018) their research to deal with class imbalance utilised a sampling technique called ADAptive SYNthetic sampling technique (ADASYN), an improved version of SMOTE that makes the distributed data for class balance look realistic. Using the same dataset for fraud detection, they have compared the SMOTE and ADASYN for higher output. As a result, SMOTE accuracy for DT is 57% and ADASYN had 60% for DT. The overall performance of ADASYN is greater than SMOTE. (Gupta et al., 2019) used different types of data balancing techniques for fraud detection to deal with data imbalance, he used SMOTE, Borderline-SMOTE, Majority Weighted Minority Oversampling Technique (MWMOTE)and ADASYN. In his research, he concluded that SMOTE techniques are well suited for fraud detection when large datasets are used.

(Kotb and Ming, 2021) in his research for data balancing techniques has created different types of hybrid data balancing models. For the comparative analysis, we used the Synthetic Minority Oversampling Technique (SMOTE) to predict insurance premiums. SMOTE is used when there is unbalanced data. (Kotb and Ming, 2021) compared the SMOTE family method with existing ML methods to predict the best sampling method. For this research, a huge data set with an imbalanced data set has been selected to get accurate results. By comparing the ML models and SMOTE family types he tested the data with 117 models. According to the evaluation parameters, the SMOTE-TOMEK model outperforms other SMOTE methods. However, in ML models, the Support Vector Machine (AUC – 80%) performs better with the oversampling method.

Machine learning can be used in various fields along with different processes for classification and regression problems. (S.Patil et al., 2021) in his research combined a new method with ML for fraud prediction. A technique called Robotic Process Automation (RPA) was applied in conjunction with Machine Learning models to automate the repetitive manual task of fraud detection. The RPA has been researched in multiple different fields for different works. Various conventional Machine-learning algorithms were used in this study. The results showed that Linear Discriminant Analysis (LDA) with an accuracy of 90% outperformed other ML models with RPA. The improved accuracy and reduced time required to complete these tedious tasks.

SMOTE is a simple algorithm that performs basic tasks. For each sample in a class, it identifies the n-nearest neighbours in the minority class. Afterwards, it generates random points on the lines between those neighbours. ADASYN is an improved version of SMOTE that achieves the same results but with a slightly better performance. It then adds a small random value to the generated samples to make them more realistic. In other words, the variance of the generated sample is slightly higher than its linear correlation to the parent.

For this research, we have utilised the popular sampling method for fraud detection SMOTE and Random over-sampling from the Over-sampling method and Random under-sampling from the Under-sampling method. (Wongvorachan et al., 2023) has researched the same idea of comparing the same sampling methods and has provided detailed information. We are utilizing the same method for sampling data in our study for comparison.

2.7 Machine Learning and Deep Learning

The field of advanced Machine Learning which is Deep Learning (DL)/ Artificial Intelligence (AI) is also well developed in Fraud detection where introducing AI models like Neural-Networks are proving to be better compared to existing Machine Learning models. The method proposed (S.Patil et al., 2021) is also a type of AI method. Artificial intelligence (AI) as mentioned (Sagar and Syrovatskyi, 2022) allows the machine to learn and solve problems on its own. AI has been developed in many fields like education, programming, video creation and even banking that can reduce the repeated work and learn from those, thus creating fewer errors. The insurance industry has adopted Deep Learning for detecting and preventing fraud. Neural networks, a type of Machine Learning algorithm used in Deep Learning and Artificial Intelligence, are modelled after the workings of the human brain.

(Wang and Xu, 2018) researched comparison of deep learning and machine learning models to automobile insurance fraud detection. Here, both the text-based data type and numerical data type have been compared accordingly for text a deep learning model known as the Latent Dirichlet Allocation (LDA) method has been used and for numerical traditional ML model has been used. The research for deep learning (DL) and machine learning (ML) was conducted separately. A data set with mixed data types was implemented into a deep neural network (DNN) for ML. The most commonly used ML algorithms were utilised, including Support Vector Machine (SVM) and Random Forest (RF). Based on the Evaluation parameter, the LDA model performance is higher compared to other ML models. Within the ML models, the Random Forest (RF) performs better than the Support Vector Machine (SVM).

Similarly, comparing the traditional ML model with the Deep Learning model (Kini et al., 2022) in his paper, mentioned that the DL model can easily figure out changes in the behavioural pattern compared to ML models. Extensive research has been conducted on detecting automobile insurance fraud using commonly utilised machine learning (ML) algorithms and a Recurrent Neural Network (RNN). The RNN was chosen due to its highly

accurate algorithms for fraud detection. After conducting multiple tests, the results demonstrate that the performance provided by the RNN is significantly superior to that of the ML algorithms.
A research by (Xia et al., 2022), implemented the idea of combining Convolutional (CNN) and Recreational Neural Networks (RNN) with Long-Short Term Memory (LSTM) to improve the accuracy of fraud detection. A mixed Deep Learning model of Neural Networks and LSTM has been tested based on evaluation parameters. As a result, the CNN-LSTM (90% accuracy) model performs better overall than other mixed models.

Based on Artificial Neural Network (ANN) (Jayasingh and Swain, 2011) has conducted research for fraud detection using credit card data. Using data mining techniques and neural network algorithms for a high fraud coverage has been combined which resulted in a low false alarm rate.

2.8 Machine Learning and PyCaret

After reviewing numerous papers on comparison studies, it has been observed that a limited number of ML models have been used separately. The evaluation criteria are similar, using a confusion matrix, but conducting research using all these models can be time-consuming as programming each model is a tedious task. Several methods have been devised to reduce the programming time required to compare ML models.

Comparing multiple ML models is a challenging and time-consuming task. However, it is essential to come to a conclusion on which model is optimal for a specific task. PyCaret is a tool that can help in this regard. Researchers in various sectors have used PyCaret for comparison and decision-making. PyCaret has been utilized for the first time in the automobile sector.

PyCaret is an auto-ML Low-Code Python Library that automates Machine Learning workflow (Moez, 2020). By using PyCaret, the Machine Learning process gets speeded up. The PyCaret library contains multiple traditional, state-of-the-art Machine Learning and Boosting algorithms within a package.

PyCaret has been widely used for fraud prediction problems (Huang et al., 2023) this paper adopted the PyCaret for credit card fraud detection. The dataset used is a large one which shows that PyCaret can handle a large volume of data. In his research, the random forest attained a 99% accuracy. This research emphasizes the user-friendly programming environment and rich visualization capabilities of PyCaret, which can be achieved with just a few lines of code. This has significant implications for simplifying data analysis for non-technical practitioners and offering preliminary data exploration tools for professional data analysts.

Predictive modelling is also a type of classification model which uses Machine Learning Algorithms to derive the results. A paper by (Pol and Sawant, 2021) has researched the implementation and uses of PyCaret using classification problems. In their research, they used a Cardiovascular prediction medical dataset, after the implementation the performance of Logistic Regression (LR) was better compared with other algorithms including boosting algorithms.

Similarly, (Gain and Hotti, 2021) in their paper, they reviewed the PyCaret along with multiple other low-code Machine Learning algorithms with experiments. Based on their research PyCaret can be used for regression and time-series problems too.

A research paper (Urunkar et al., 2022), used a Python library package called PySpark using Machine Learning Algorithms for the validation of fraudulent transactions in the insurance sector. PySpark is used to validate the data in real time when the data has been fed to the system. Large-scale data can be monitored and validated by Machine learning using PySpark. Traditional ML algorithms have been used for the comparison, in which based on the evaluation parameter Logistic Regression (73% accuracy) has a better performance compared to Extreme-Boosting, Decision Tree, Random Forest and K-Nearest Neighbour. (Moharekar and Pol, 2022) In their paper, they used a medical dataset to predict thyroid diseases utilizing Machine Learning algorithms. An improvised dataset has been created for better accuracy prediction. The data was then used for comparing Machine Learning algorithms with PyCaret. The improvised/balanced dataset has been gathered by oversampling the unbalanced thyroid diseases dataset. Based on the evaluation parameters, the accuracy rate of 99% of the Extra Tree Classifier is higher followed by the Support Vector Machine and Random Forest both with 98%.

Similarly, (Whig et al., 2023) used PyCaret to predict diabetes using a diabetes dataset. The comparison of multiple Machine learning algorithms is tedious work which consumes a lot of time in programming all the algorithms separately along with evaluation metrics. Using PyCaret reduces the time and work needed in a large part. Based on the evaluation from PyCaret, the Extreme Gradient Boosting method with an accuracy of 90% has better performance, when fine-tuned even further.

The development of Machine Learning has been noted in recent years in many industries. The programming is being minimized and easy-to-code algorithms are being created. The process of data analytics in all this research has been done manually. (Sarangpure et al., 2023) in his research Automated Machine Learning (AML), used Streamlit an AutoML application along with PyCaret. Streamlit can automatically preprocess the data and provide insights without complex programming. The error of computing can be minimised greatly by using AutoML programs. The parameters are set to our needs and for ML, PyCaret has been used. Multiple different datasets were tested and the results were proven.

From this section, we concluded, that for a comparison study one needs to have all the available models and required time to work on them. PyCaret is the one providing both of them for our research.

2.9 Summary

In this chapter, we have reviewed many research papers that provide insights for the research to be conducted. The review has been done segment by segment on which the other researchers have proceeded with classification problems. Insurance fraud classification analysis has been researched and studied in many research papers extensively. The process of fraud detection has come a long way from manual investigation and investigation speed to fraud prediction in a matter of hours. This is possible because of Machine learning. Machine learning has also developed in many ways. These papers have explained the research done using machine learning for fraud prediction. The research on feature selection which is an important process in data pre-processing to reduce the number of features to get a high evaluation score. The data balancing technique to deal with data imbalance which was a problem in classification studies and the different kind of balancing technique that helps increase the performance and accuracy of the machine learning model. The machine learning models which is used for fraud prediction. The most commonly and widely used models such as Decision Tree (DT), Random Forest (RF), and Logistic Regression (LR) and the best boosting models like Gradient Boosting (GB) and Extreme Gradient Boosting (XGB) are explained briefly these papers and introduction and usage of PyCaret into Machine Learning are explained.

From these reviewed papers, the idea of a comparative analysis in the automobile insurance sector has derived and helped us with the methods for feature selection and data balance also the implementation of the new PyCaret model in the insurance sector. In the next session, the entire process of the research to be conducted has been explained.

CHAPTER 3:
METHODOLOGIES

3.1 Introduction

This chapter explains the research methodology that has been implemented for this research and each step of the process. In this paper, we are performing a comparative analysis using the Python library in which feature selection and data balancing techniques are used for optimal output. Figure 3.1, portrays the method we implemented in this paper.

In this paper, we use the low-code auto-ML Python library known as PyCaret. PyCaret is mostly used for classification problems and helps in understanding the suitable ML model for the dataset used. Many studies on the comparison of ML algorithms have been extensively researched for better fraud detection. These studies have used traditional ML models like LR, DT, RF, SVM, NB and gradient boosting models such as XGBoost, AdaBoost and CatBoost. The PyCaret package used in this study contains a total of 16 different algorithms namely LR, KNN, LDA, QDA, NB, DT, GBC, ET, RF, LightGB, Ada Boost, CatBoost, RC, SVM, XGB and DC. The common evaluation metric used for evaluation criteria is the Confusion Matrix. The models in these papers are selected based on the evaluation parameter resulting in Accuracy, F1-score, Precision, and Recall. The evaluation metrics for PyCaret contain Accuracy, Precision, Recall, F1-Score, Kappa and MCC. The proposed methodology for this study includes data cleaning, preprocessing, feature selection, implementation, and model output.

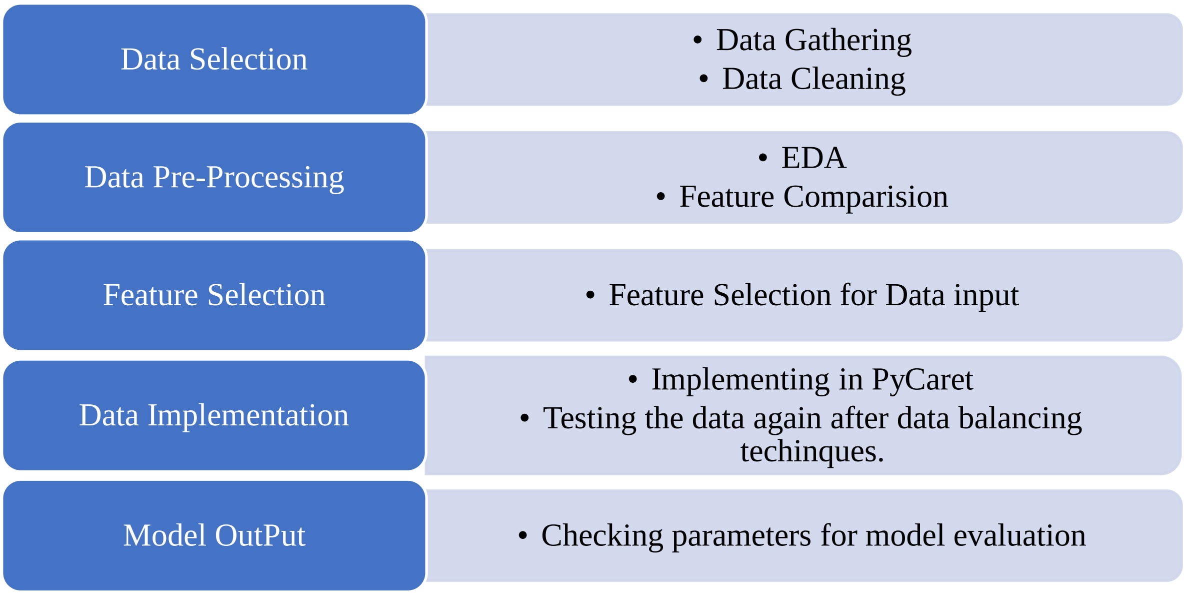

Data Selection
• Data Gathering
• Data Cleaning
Data Pre-Processing
• EDA
• Feature Comparision
Feature Selection
• Feature Selection for Data input
Data Implementation
• Implementing in PyCaret
• Testing the data again after data balancing techinques.
Model OutPut
• Checking parameters for model evaluation

Figure 3.1 Method Process

3.2 Data Collection

Insurance fraud came into existence when people started to get insurance. Until the late 1980s, the importance of insurance fraud and the need to prevent fraud had not been known. To understand how fraud is happening insurance, data about insurance is needed. The only available data set that has a good number of data is gathered by a third-party software called Angoss software seeker (Rai et al., 2018).

This dataset is first utilised (Brockett et al., 2002) in his study about fraud detection. The data for this study is available in Kaggle (Vehicle Insurance Fraud Detection, 2023), an open-source data platform. This particular data is widely used common automobile insurance data (Xia et al., 2022) that has major data imbalance. The data selected for this study is automobile insurance fraud detection which consists of a total of 33 columns and more than 15,420 entries. The dataset consists of 8 numerical/ordinal features and 25 categorical features having 14,497 non-fraud samples (94%) and 923 fraud samples (6%) (Phua et al., 2004). This study is based on a Quantitative research model where we analyse the given data to figure out the desired outcome. The numerical features in this dataset are of discrete type data and the categorical feature is of nominal type data. The data attributes have been briefed in Table 1.

The data has been collected based on the fraud insurance claims that happened in the years 1994-96, based on the years through manual investigation found the fraudulent claims happened. Machine learning has been introduced into the insurance sectors to reduce manual error and increase the investigation speed. The proposed model for this study is based on an evolutionary algorithm that selects only the most prominent attributes from the original data set. This ensures that only the most relevant features are used in the analysis. Data cleaning is then performed to ensure that the data is free of null and missing values. In addition, outliers are checked and replaced during pre-processing. These steps ensure that the data is accurate and reliable and that the results obtained are of the highest quality. For classification purposes, a test set is processed using the best-performing model and the chosen attribute set. This approach ensures that the system can accurately classify new data points and make informed decisions. The effectiveness of the proposed system is demonstrated by conducting several experiments on the data set, which show that it is highly accurate and reliable. In conclusion, the proposed model is a powerful tool for data analysis and classification. By using an evolutionary algorithm and rigorous data-cleaning techniques, it ensures that only the most relevant features are used in the analysis. This approach leads to highly accurate and reliable results that can be used to make informed decisions.

3.3 Experimental Setup

This segment explains the detailed process of how the research study will be carried out. In previous comparative studies that have been carried out the output has been derived based on one single technique of feature selection or data class balancing technique. In this research, we have compared 3 different processes of output that have been derived from 2 different feature selections and 3 different class balancing that have been implemented in 16 different ML models and evaluated against 7 different evaluation parameters to identify the best model for fraud detection. The following flowchart explains the steps involved in the research process.

The purpose of the experiment is to compare various types of output. These include the output from the original data, the encoded original data, encoded data with SMOTE, feature-selected data, and feature-selected resampling technique. Figure 3.2 explains the Framework of the study.

To achieve this, the original experimental setup involves obtaining output from PyCaret after balancing the imbalanced data using SMOTE. The features in the given dataset are helpful.

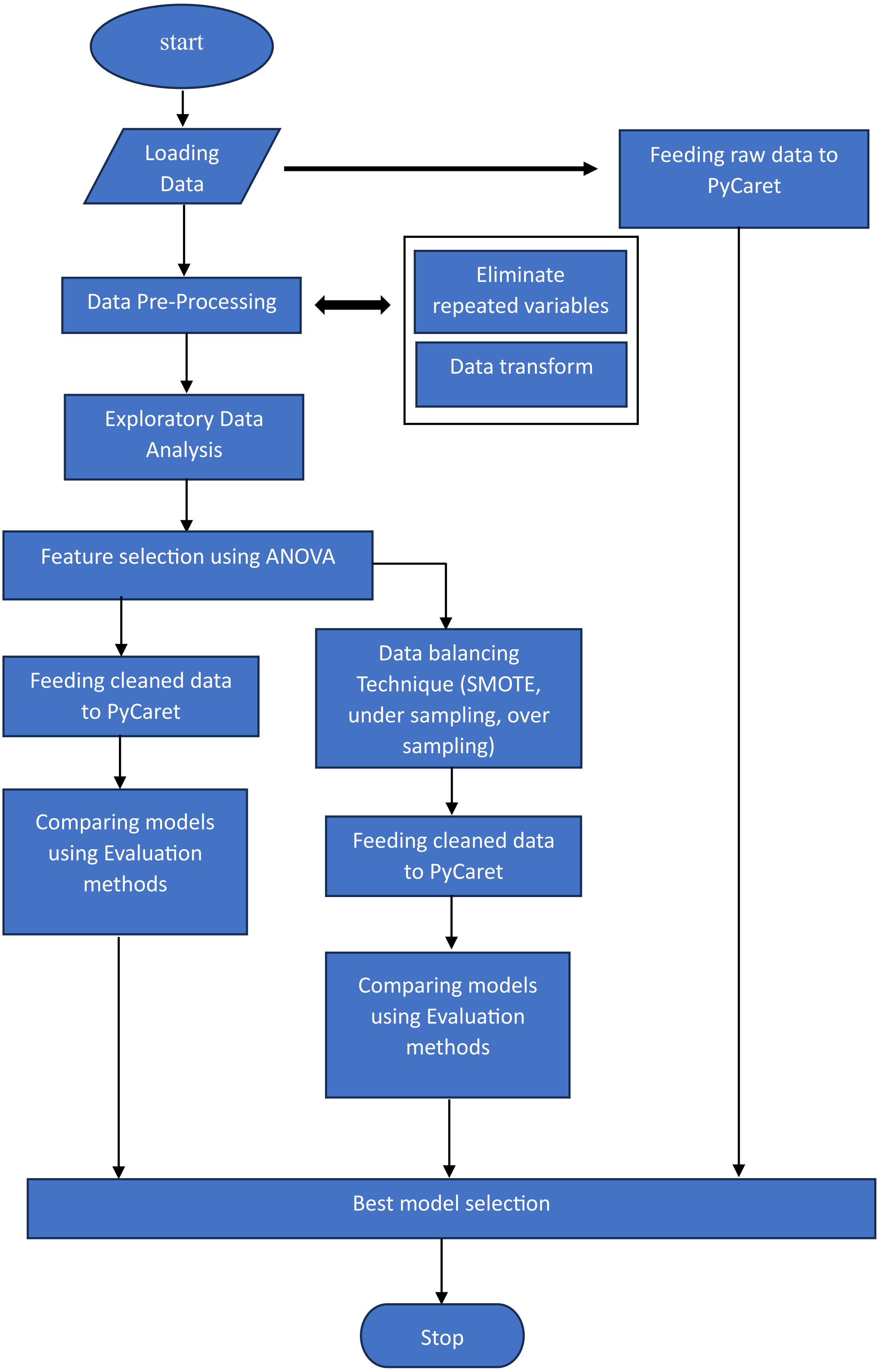

start
Loading Data
Feeding raw data to PyCaret
Data Pre-Processing
Eliminate repeated variables
Data transform
Exploratory Data Analysis
Feature selection using ANOVA
Feeding cleaned data to PyCaret
Data balancing Technique (SMOTE, under sampling, over sampling)
Comparing models using Evaluation methods
Feeding cleaned data to PyCaret
Comparing models using Evaluation methods
Best model selection
Stop

Figure 3.2 Process Flowchart

3.4 Data Pre-Processing

3.4.1 Data Transformation

The data transformation is carried out before cleaning in this research to understand the data. The dataset selected for this research consists of both numerical and categorical datatypes.

The goal of fraud detection is to find fraud in any given data. The machine can understand the data in a specific language which is 0 and 1 (Wei et al., 2020). The classification problem also follows a similar method. Where the given data need to be in a numerical format and the target values are to be in a 0's and 1's format. For that purpose, we need to transform any given data type into binary format. Many such algorithms in machine learning convert the data type. This process is known as data transformation.

To perform EDA and get an understanding of the relation of each variable column the data needs to be transformed into appropriate datatypes. The EDA can be performed by using Numerical values. Data mining (DM), a type of data preprocessing technique has been used to handle the categorical values in the dataset. The categorical columns in the dataset are transformed into numerical values using a Data Mining Technique called Label Encoder (LE). The Label Encoder is an essential tool for converting categorical values to numerical values in classification problems. Among the several data mining techniques such as one-hot encoding, binary encoding, ordinal encoding, and nominal encoding, Label encoding is the most effective choice for this study. Converting categorical columns into numerical ones is a crucial step in machine learning projects. Label Encoding is a technique used to carry out this process. It helps to prepare the data for machine learning models that can only work with numerical data. Therefore, Label Encoding serves as a vital pre-processing step in machine-learning projects. It consumes less time and memory during programming and unlike one-hot encoding, Label Encoding does not create any unusable columns (Jiang et al., 2020), making it the most efficient and effective choice for our needs. Label Encoder assigns integers to represent text under the same categorical parameter. However, it causes meaningless numerical comparisons between the different types and can lead to misinterpretation, negatively affecting model performance.

3.4.2 Data Cleaning

A data cleaning process is carried out to rectify the data. The data is rectified by converting inaccurate information in the dataset like cleaning null values, dropping or removing unwanted data and filling in the missing values in the dataset. To clean data or to understand the data, the data used needs to be in a form that is understandable which we have attained through the data transformation step.

The data cleaning process ensures that the quality of the dataset is maintained and helps in integrity. The Data cleaning helps in getting accurate results during Data Analysis which helps in understanding the data even better. The dataset used in this study has been examined for null/ missing values. No missing or null values were found in the dataset.

3.4.3 Feature Selection

Feature selection is an important process in the data pre-processing step. Feature selection is the process of selecting a subset of features from original data features to optimize the process based on a specific criterion. By feature selection process, the learning algorithm speed is accelerated and the predictive accuracy of the classification algorithm is further increased compared to when not selected. The feature selection process can be done by many methods namely filter method, wrapper method, and embedded method (Benkessirat and Benblidia, 2019). When selecting a feature selection method for fraud detection, there are several factors to consider, such as the nature of the data, the size of the dataset, and the specific goals of the fraud detection system. Here are some commonly used feature selection methods:

- Univariate Feature Selection is a filter method.
- Variance Thresholding: This method removes features with low variance, assuming that features with little variance are less informative.
- SelectKbest: This method selects the top k features based on univariate statistical tests (e.g., chi-squared, ANOVA).
- Recursive Feature Elimination (RFE) is a wrapper method.
- RFE: This method iteratively removes the least important features based on model performance until the desired number of features is reached.
- Tree-based Methods:
- Random Forest Feature Importance: This method measures the importance of each feature by observing how much each feature contributes to reducing impurity in decision trees.

- Gradient Boosting Feature Importance: Similar to Random Forest, this method measures the contribution of each feature to the improvement of the model's performance.
- L1 Regularization (LASSO) is an embedded method:
- LASSO (Least Absolute Shrinkage and Selection Operator): This method adds a penalty term to the linear regression cost function, encouraging sparsity and automatically selecting a subset of important features.
- Correlation-Based Methods:
- Correlation Matrix Analysis: This method identifies and removes highly correlated features, as they may carry redundant information.
- Mutual Information:
- Mutual Information: This method measures the dependency between two variables, helping to identify features that are informative for the target variable.
- Sequential Feature Selection:
- Sequential Forward Selection (SFS) and Sequential Backward Selection (SBS): These methods iteratively add or remove features based on model performance.

In this paper, we are using the filter method for our feature selection. Scikit-learn library in Python is implemented for the feature selection in the pre-processing step. The features are selected based on the F-score (Elssied et al., 2014). These F-scores are calculated using the F-statistic, a part of the ANOVA (Analysis of Variance) method, commonly used for feature selection in classification problems. The F-score is a statistical measure that evaluates the difference between multiple groups and within each group to determine if there is a significant variance among the groups.

$$F = \frac{Within\ Group\ Variance}{Between\ Group\ Variance}$$

Using ‘f-classif’ - f-score for classification problems for the feature section and ‘SelectKbest’ to select the top features that emerge from the f-score. The entire data has been fit after the data transformation to compute the f-score and transform only the top features based on the f-score and P-Values.

3.5 Data-Balancing Technique

A similar comparative study, conducted (Badriyah et al., 2018) with the same data mentioned that the rate of fraudulent claims is only 6%, which means that approximately 1 in every 17 claims are fraudulent which is a prime example of data imbalance.

To rectify the data imbalance issue, data balancing techniques are used. Imbalanced data in a dataset will lead to incorrect outputs. Accuracy for the model can be only correct when both opposing values are of equal terms. The imbalance in data is a common occurrence in classification problems where the target variable is a binary value. Using a decision tree classifier helps achieve a balanced distribution of classes, resulting in improved outcomes (Mary and Claret, 2021). The imbalance in data can be rectified by collecting more data or by changing the performance metrics or trying out a different algorithm or by resampling the data. All these options are useful in individual situations. But the most commonly used option is Resampling.

Data resampling is a method which is used to gather more information about the samples. This means that by adding more values on the minor side to equalize the major side or reducing the number of values from the major side to equalize the minor value. The resampling is a method where the algorithm equally distributes the values. There are 2 different types of resampling

- Over-sampling
- Under-sampling

3.5.1 Under -Sampling

Under-sampling (Down-sampling) removes the values randomly from the majority class where the values are higher compared to the other values. Under-sampling is a technique used when there is a greater number of observations in a particular class or group as compared to others. It involves reducing the number of observations in the over-represented group to create balance in the data set. In this method, the data loss is inevitable which can lead to change in output and may also lead to underfitting. There are many different types of under-sampling methods exist namely

- Random Under-sampling (RUS)
- Condensed Nearest Neighbour Rule (CNN)
- Near Miss Under-sampling
- Tomek Links Under-sampling
- Edited Nearest Neighbours Rule (ENN)
- One-Sided Selection (OSS)

- Neighbourhood Cleaning Rule (NCR)

In these, the most commonly used model is Random under-sampling. It is one of the simplest under-sampling methods it involves randomly deleting examples from the majority class in the training dataset. This technique is highly effective for large-scale datasets(Alamri and Ykhlef, 2022). We have also implemented the Random Under-Sampling method for our study.

3.5.2 Over-Sampling

Over-sampling (up-sampling) adds the value randomly for the minority class to meet the values of the majority class. Oversampling methods duplicate examples in the minority class or synthesize new examples from the examples in the minority class. Oversampling approaches are a popular choice to deal with imbalanced datasets. There are several over-sampling methods including the famous state-of-the-art SMOTE technique. The methods are

- Random Oversampling (ROS)
- Synthetic Minority Oversampling Technique (SMOTE)
- Borderline-SMOTE
- Borderline Oversampling with SVM
- Adaptive Synthetic Sampling (ADASYN)

The Synthetic Minority Oversampling Technique (SMOTE) is famous and used very often in classification imbalance datasets. SMOTE is a versatile technique that can be used with various machine learning algorithms. It is not tied to a specific model, making it applicable in a wide range of fraud detection scenarios. SMOTE generates new instances to increase the number of minority samples in the dataset. SMOTE algorithm effectively generates new samples by selecting similar examples in feature space and drawing a line between them to create additional data points (Kotb and Ming, 2021). We have used SMOTE and Random Oversampling to compare and get the best sampling technique that provides better performance evaluation. Random Oversampling (ROS) is the same as RUS where the working is the exact opposite. The ROS adds values to the minority that equals the majority. The negative side of oversampling is that when adding random values to a minority results in a greater chance of class mixture. SMOTE is also built in PyCaret to deal with class imbalance.

SMOTE is highly sought out for fraud detection problems. SMOTE generates synthetic examples by interpolating between existing minority class instances. This approach helps preserve the information present in the minority class, potentially leading to a more accurate representation of the underlying patterns associated with fraud. SMOTE is a readily available library for Python. SMOTE has been widely adopted and studied in the literature on imbalanced classification. Its effectiveness has been demonstrated in numerous studies and applications, making it a popular choice for addressing class imbalance.

3.6 Machine Learning Model

In this section, we will learn about the ML model used for the implementation. The reduce the overall programming time for each ML algorithm and evaluation metrics we have used a new and promising Python Module called PyCaret.

3.6.1 PyCaret 3.0

PyCaret is a Python library used for machine learning that simplifies the end-to-end model development process with its user-friendly approach and automation capabilities. Unlike traditional classification methods, PyCaret streamlines tasks such as data preprocessing, feature engineering, model selection, and hyperparameter tuning, reducing the need for manual intervention. Its strength lies in its ease of use as it enables both beginners and experienced data scientists to compare multiple models effortlessly, automatically optimize hyperparameters, and visualize results. Moreover, PyCaret supports model deployment and offers consistent APIs across various machine learning tasks, contributing to a more efficient and standardized workflow. While PyCaret excels in automation and accessibility, the choice between PyCaret and traditional methods ultimately depends on project requirements and user preferences for either a streamlined, automated process or a more hands-on, customizable approach.

The model used in this study is a comparative analysis model using a low-code Python library known as PyCaret(Moez, 2020). PyCaret is an open-source, low-code machine learning library in Python that aims to reduce the hypothesis to insight cycle time in an ML experiment. PyCaret's API is arranged in modules. Each module supports a type of supervised learning (classification and regression) or unsupervised learning (clustering, anomaly detection, Natural Language Processing (NLP), association rules mining) (Moez, 2020). Table 3.1 explains the categorisation of learnings.

Table 3.1 PyCaret fields

Supervised	Unsupervised
Classification	Clustering
Regression	Anomaly Detection
Time Series	Association Rule Mining
	Natural Language Processing

The PyCaret used in this study is the classification model. PyCaret enables us to perform end-to-end experiments quickly and efficiently. In comparison with the other open-source machine learning libraries, PyCaret is an alternate low-code library that can be used to perform complex machine learning tasks with only a few lines of code.

PyCaret relies on the principles of automated ML, to assist in algorithm selection (de Holanda et al., 2024). This specialized library can assess algorithm performance using a predefined set of metrics.

PyCaret is a powerful tool that offers various features to simplify machine learning tasks. One of its notable features is automated data splitting, where the model itself splits and trains the data, providing accurate results. PyCaret also offers automatic feature selection, feature importance and other crucial steps which eliminates the need to write separate codes for it. The resulting model can be even more fine-tuned to see greater results.

The PyCaret can be considered as a Deep Learning/Artificial Intelligence model too because of the Machine Learning algorithm model it deploys. Instead of implementing each Machine Learning model individually, PyCaret deploys a total of 16 different Machine Learning on its own with the given input and produces output based on the evaluation metrics. The vital information in PyCaret is that more and more new models can be used as we install them in PyCaret. The PyCaret consist of both the traditional and modern machine learning models. The following ML algorithms are integrated into PyCaret,

- LR – Logistic Regression
- DT - Decision Tree
- RC – Ridge Classifier
- ET – Extra Trees Classifier
- LDA – Linear Discriminant Analysis
- RF – Random Forest Classifier
- Light GB – Light Gradient Boosting
- XG-boost – Extreme Gradient Boosting
- GBC – Gradient Boosting Classifier
- ADA – Ada Boost Classifier
- NB – naïve Bayes
- SVM – Support Vector Machine
- KNN – K- Nearest Neighbour
- DC – Dummy Classifier

- QDA – Quadratic Discriminant Analysis
- CatBoost – CatBoost Classifier

Performing a comparative research study by manually integrating separate algorithms can be extremely time-consuming and resource-draining. However, the PYCARET feature obviates the need for such manual work, making the process much more efficient and streamlined. This specific algorithmic choice proves to be advantageous in comparison to other approaches, making it the most optimal and effective way to perform comparative research.

3.6.1.1 Logistic Regression

Logistic Regression (LR) is a statistical model more often used for classification and prediction problems. Logistic regression is mostly commonly used for binary prediction based on the given dataset (What is Logistic regression? | IBM, 2023). The logistic regression is classified into 3 different types. **Binary, multinomial** and **ordinal logistic regressions**. The binary regression is included when the target variable is of only two distinct values like YES or NO, 1 or 0 (Pant, 2023).

The multinomial regression is included when there are 2 or more target variables used such as advertising campaign. Ordinal regression is mostly used when the target variable is based on the ratings such as grading scales from 1 to 5. The LR in ML is used to distinguish the given input into target-based results. The Scikit-Learn package is used when implementing the LR for ML. The PyCaret uses the LR for the classification and prediction problems.

3.6.1.2 K-Nearest Neighbour

K-Nearest neighbour also widely known as KNN is used to classify numbers. KNN is a distance and density-based detection method that is based on regression techniques. K-Nearest Neighbours (KNN) is a straightforward and effective machine learning algorithm based on the principle that similar objects tend to be close to each other (Frenzel, 2023). To predict a new data point, KNN identifies the 'k' with the most similar data points (neighbours) from the dataset. For a classification problem, the KNN uses the majority numbers in the neighbour and for regression, it takes on the average.

This approach is widely used and can be a useful tool for data analysis and prediction tasks. The KNN is measured using Accuracy, Prediction, Recall and F1 Score for classification problems and metrics like Mean Absolute Error (MAE) and Mean Squared Error (MSE) for regression.

3.6.1.3 Naïve Bayes

A supervised learning model that's derived based on Bayes theorem for classification problems. The Naive Bayes (NB) can provide fast ML models that can help with quick predictions. Naïve Bayes operated based on probability like the probability that the mail we received may be spam or not spam (Pedregosa et al., 2011). NB is also used widely for text classifications. Based on the input, either the numbers or text the probability functions determine and results with binary model output like spam or not spam.

3.6.1.4 Decision Tree Classifier

Decision Tree (DT), is a supervised learning method used for classification and regression problems. The Decision tree is devised in a way that follows the way of human decision-making (Bento, 2021). The decision tree follows a set rule to make a decision. By making the tree follow the yes/no questions based on the given dataset. Just like a tree branch that splits into many sub-branches, the DT determines the output using the target variable. The Decision tree is also mentioned as the CART (Classification and Regression Tree) algorithm.

3.6.1.5 Random Forest Classifier

Random Forest (RF) is a widely used supervised machine learning algorithm that combines the output of multiple decision trees to reach a single result. Random Forest is based on a bagging method, which means the more similar values are collected in one bagging. RF is used for its speed and efficiency. RF can handle both classification and regression problems. Random forests have an inherent ability to handle problems with multiple classes (Biau and Scornet, 2016). Multiple decision trees are created using the input data, and based on the opinion created by the DT the prediction can be calculated. Random forest selects observations in a randomized manner, constructs a decision tree, and then computes the average outcome (Harjai et al., 2019). This is achieved without the use of any pre-defined set of formulas.

3.6.1.6 Extra Trees Classifier

Extremely randomized tree classifier (Extra Tree Classifier) can significantly improve the accuracy of classification results. An extra tree classifier is an ensemble model that can be used for both classification and regression problems(Pedregosa et al., 2011). By collecting results from multiple decision trees, this technique can minimize the impact of individual decision tree errors and create a more robust and reliable model. The extra tree classifier is similar to RF by using the DT results, the difference is in their model construction. The extra tree is used to improve the accuracy and control of over-fitting by utilizing the averaging of DT results. The results for the classification are concluded by majority vote and by calculating the mean for regression (Own et al., 2021). The extra tree reduces the computational cost considerably compared to using RF.

3.6.1.7 SVM -Linear Kernal

SVM or Support Vector Machine is a supervised learning method used for classification, regression, and outlier detection based on statistical approaches. It is used in PyCaret for classification problems. SVM classifies data by finding the most optimal decision boundary that maximally separates different classes. Training a Linear Kernel is faster compared to other methods. SVM is mostly used for text and image classification SVM- Linear Kernal is used when the target variable is linearly separable, which means the data points can be classified into two classes. It reduces the computational time used by SVM.

3.6.1.8 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) also known as normal discriminant analysis (NDA) is derived from Fisher's linear discriminant. LDAs are commonly used for dimensionality reduction. LDA is a supervised learning algorithm that finds a linear combination of features that separates two or more classes of data. Just like LR, LDA is also widely used for binary classification. LDA is closely related to Principal Component Analysis (PCA) (Whitfield, 2023). LDA is a dimensionality reduction technique for classification problems. LDA maximizes the distance between means of two classes and minimizes the variance within the individual class. LDA reduces computing costs significantly. The algorithm mentioned above is used for between- class scatter in LDA (Subasi, 2020).

3.6.1.9 Quadratic Discriminant Analysis

Just like LDA, Quadratic Discriminant Analysis (QDA) is also a supervised learning algorithm. In both analyses, there is no extra tuning needed for the hyperparameter. The QDA is used for classification and dimensionality reduction. The objective of QDA is to categorize observations into predefined classes based on their features. It provides more flexibility than LDA in modelling complex data relationships but requires more data to avoid overfitting.

3.6.1.10 Ridge Classifier

Ridge classification is used to analyse the linear discriminant model. The ridge classifier regulates the model to prevent overfitting by utilising Ridge regression. This classifier first converts the target values into {-1, 1} and then treats the problem as a regression task. During prediction, if the predicted value is less than 0, it predicted class label is -1 otherwise the predicted class label is +1. The ridge classifier is used when there are more features in the given dataset. The ridge classifier is derived from ridge regression. It takes the classification target variable converts it to continuous values and determines the output. (Yan et al., 2020) this paper uses a ridge regression model for automobile fraud detection.

3.6.1.11 Dummy Classifier

A dummy classifier usually serves as a baseline to compare with other classification algorithms (Santos et al., 2021). The dummy classifier does not generate any insights on the given data and classifies based on simple rules. The output of any other classification model to know if it's a good score the dummy classifier is used to see if the score we got is of a similar level. The dummy classifier is mostly used when there is a class imbalance in the dataset. The dataset can then be further worked with class balancing techniques for a more optimal model prediction.

3.6.1.12 Gradient Boosting Classifier

Gradient Boosting classifier is a boosting algorithm that gets even better results compared with another model. Gradient boosting classifier (GB) is also an ensemble learning model used for both classification and regression problems. Gradient boosting works by combining multiple weak models into one powerful model. Gradient Boosting is majorly used for prediction speed and accuracy even when large-complex data has been used. The Gradient boosting classifier is used when the target values are binary and the gradient boosting regressor when the target values are continuous. Just as a dummy classifier, the gradient boosting classifier is built after another model is built. Unlike for comparison's sake, gradient boosting provides better results. Just like the gradient boosting algorithm, there are several other boosting algorithms like Ada boost, Cat boost, light Gradient boost and Extreme Gradient Boosting. Compared to gradient boosting other boosting algorithms are widely used namely extreme gradient boost which is a more popular boosting algorithm for fraud detection with a high accuracy rate. The boosting algorithms are even more fine-tuned for better scores.

3.6.1.13 Light Gradient Boosting Classifier

Light Gradient Boosting Machine (LightGBM) is a gradient-boosting method. LightGBM is derived based on decision trees to increase the efficiency of the model and reduce memory usage compared to other gradient-boosting methods. LightGBM is a versatile method that can be used for different datatypes like categorical, numerical and text. (Ke et al., 2017) in his paper thoroughly studied the LightGBM. Gradient-based one-sided sampling (GOSS) and Exclusive Feature Bundling (EFB) are the different types of methods in LightGBM. The LightGBM works in a leaf-wise algorithm which is a fixed model that reduces loss when compared to other boosting algorithms that use level-wise algorithms.

3.6.1.14 AdaBoost Classifier

AdaBoost in short called Adaptive boosting is an ensemble model that is used for both classification and regression models. AdaBoost is a type of statistical classification meta-algorithm. This means that it starts by fitting a classifier on the original dataset. It then fits additional copies of the classifier on the same dataset but adjusts the weights of incorrectly classified instances. This adjustment makes subsequent classifiers focus more on difficult cases, leading to better overall performance. AdaBoost is also a boosting algorithm that is most commonly used when a decision tree is used as a base classifier. Other algorithms can also be used as a base classifier. AdaBoost is easy to implement and is not prone to over-fitting whereas it is slower compared to XGBoost and is affected when there is noise in data. The AdaBoost chose to provide the results by voting just as a random forest model.

3.6.1.15 Extreme Gradient Boosting

XGBoost stands for Extreme Gradient Boosting and it’s an open-source implementation of the gradient boosted trees algorithm. It’s been one of the most popular boosting ML models that’s been widely used for classification problems, particularly for fraud detection. (Chen and Guestrin, 2016) in their research on XGBoost they have explained in detail about the process, types and explained the programming in detail. XGBoost is an ensemble model that is used for both classification and regression problems. XGBoost is used for supervised learning problems, where we use the training data (with multiple features) to predict a target variable. The XGBoost is easy to use and is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. the model has been specifically designed to improve speed and performance. XGBoost works based on the decision tree algorithm where weights are important. XGBoost splits the weight along the variables which are then given as input for decision tree for prediction (Singhal et al., 2023).

3.6.1.16 CatBoost Classifier

CatBoost, also known as Categorical Boosting, is a powerful boosting method designed for regression and classification problems with a large number of independent features. It was released after XGBoost and LightGBM in 2021 (Hancock and Khoshgoftaar, 2021). One of Cat Boost's unique features is its ability to handle both numerical and categorical features without requiring any feature encoding techniques such as One-Hot Encoder or Label Encoder. This is possible because CatBoost uses the Symmetric Weighted Quantile Sketch (SWQS) algorithm, which automatically handles missing values in the dataset to reduce overfitting and improve performance. Due to the way CatBoost handles encoding categorical features and calculates its output values, it is less prone to overfitting compared to XGBoost or LightGBM.

3.7 Evaluation Metrics

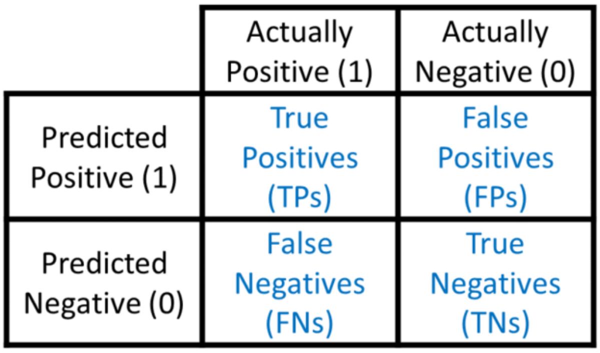
Evaluation metrics are used to predict the outcome of the model we built in Machine Learning. The quality of the model built can be measured through evaluation metrics. Evaluation metrics are utilized for both Supervised and Unsupervised Machine Learning models (Goyal, 2021). The metrics for both models are fundamentally different. Supervised models can be broadly classified into two types - classification and regression models. Clustering belongs to unsupervised learning. In a model, a testing model is always subjected to evaluation, based on the evaluation score the model will be further used. A testing model is created when the given dataset is split into test and train sets instead of using the entire dataset. (Vujovic, 2021) in his research studied the different types of evaluation and has done experiments using a medical dataset. (Ferrer, 2023) this recent study has analysed and compared multiple evaluation metrics with a metric called Expected Cost (EC) which is from a statistical course that was also used in Machine Learning. There are many different types of evaluation metrics for both ML models. In PyCaret, the classification model is evaluated using the following metrics

- Accuracy
- Precision
- AUC curve
- Recall
- F1 Score
- Kappa

- MCC – Matthew's Correlation Coefficient

All the evaluation metrics for classification are based on the Confusion matrix. Confusion matrix is a binary classifier that uses values like Positive (1) and Negative (0) (Vujovic, 2021). The confusion matrix in classification determines the performance of the machine learning model. Based on these metrics the output can be two or more classes. The classes are based on the combinations of predicted and actual values.

- True Positive (TP) - Instances where the positive outcome is accurately predicted.
- False Negative (FN) - Instances where the positive outcome is wrongly predicted.
- True Negative (TN) - Instances where the negative outcome is accurately predicted.
- False Positive (FP) - Instances where the negative outcome is wrongly predicted.



	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

Figure 3.3 Confusion Matrix

The confusion matrix is extremely useful for measuring the Recall, Precision, Accuracy, and AUC-ROC curves. Figure 3.3 explains the Confusion matrix segregation.

3.7.1 Accuracy

Accuracy is used to measure the performance of the model. Accuracy is utilised to measure how often the classifier correctly predicts. Accuracy is used most often when the dataset is balanced. The accuracy is predicted based on the number of correct predictions out of all the predictions.

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

The accuracy of a model is considered good when it is 1 and not when it is 0.

3.7.2 Precision

Precision is a measure of how accurate a model's positive predictions are. It is defined as the ratio of true positive predictions to the total number of positive predictions made by the model. Precision is the proportion of true positives among all predicted positives. Precision is useful when avoiding false positives is more important than avoiding false negatives. Precision is mostly used for recommendation systems. The precision is calculated based on the total number of true positives divided by the total number of predicted positives.

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

3.7.3 Recall

Recall measures the effectiveness of a classification model. Recall also called sensitivity (Roy and George, 2017) is used to determine the model's ability to detect positive samples. The recall score increase is proportional to the positive samples. It is the ratio of the number of true positive (TP) instances to the sum of true positive and false negative (FN) instances. The recall value is also known to be True Positive Rate (TPR).

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

3.7.4 AUC Curve

The AUC-ROC curve is a graphical representation of the effectiveness of a binary classification model. The Area Under the Curve (AUC) can also be termed as AUC-ROC curve. The AUC curve helps us understand the different models based on the data summarised by the Receiver Operating Characteristic (ROC) curve. ROC is a graph model that summarizes the model's performance. The ROC is a probability curve that plots the True Positive Rate (TPR) against the False Positive Rate (FPR). This curve visualises the trade-off between TPR and FPR using different decision thresholds.

- True Positive Rate (TPR)

  The true Positive Rate represents the proportion of actual positive cases that are accurately identified as positive.

- False Positive Rate (FPR)

  FPR refers to the portion of negative data points that are incorrectly classified as positive, concerning all data points that are negative.

$$FPR = \frac{FP}{FP + TN}$$

Both TPR and FPR have values in the range of [0,1] which are computed at varying threshold values. A model is considered a perfect classifier when the TPR value is higher than the FPR value.

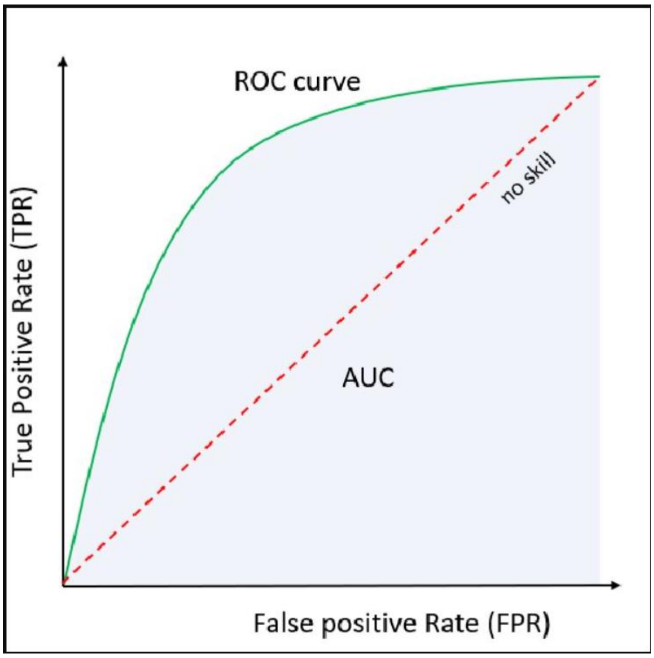


Figure 3.4 AUC-ROC Curve

Figure 3.4 is the AUC curve which is calculated by using the True Positive Rate (TPR) and False Positive Rate (FPR).

3.7.5 F1 Score

F1-score is used to evaluate the overall performance of a classification model. The F1 Score is calculated based on the values of Precision and Recall. The f1 score determines how precise and robust the model is. The F1 score value is proportional to the values of both Precision and Recall.

$$F1\ Score = \frac{2 * precision * recall}{precision + recall}$$

3.7.6 Kappa

Cohen's Kappa coefficient, also known as Kappa, is a measure of how well a machine learning model classifies instances by comparing them to data that is marked as the ground truth. Kappa is used in classification to compare the performance of the classifier models. Kappa considers the accuracy of a random classifier and provides an expected accuracy score. Kappa is used to measure categorical items.

$$k = \frac{(po - pe)}{(1 - pe)}$$

- po - total accuracy of the model
- pe – random accuracy of the model

3.7.7 MCC

Matthew's correlation coefficient (MCC) is a statistical metric proposed for binary classification. MCC is the most informative single score to establish the quality of a binary classifier that outputs hard decisions (Ferrer, 2023). MCC is calculated by utilising the values from the confusion matrix. The MCC is preferred when the dataset is unbalanced and Accuracy is preferred when the dataset is balanced. MCC is more reliable than accuracy and F1 score for imbalanced datasets.

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

The range of values for MCC is [-1, +1]. The highest value is +1 and the lowest is -1 (Chicco and Jurman, 2020)(Vujovic, 2021). The MCC value is high when the classifier works well on both positive and negative elements.

3.8 Summary

In this section, we have explained the total process of research methodology steps in detail. The data selection and description explain the insurance data selection with the factors involved in insurance data were stated briefly. In the Data Pre-processing, steps for data cleaning and clearing null values are used, as the data we used is void of null values and no missing values were found the data cleaning step is not necessary. In Data transformation, the different data types in the dataset are converted into binary values from categorical and numerical values. The label encoder is used for this step as it works well for this study compared to one-hot encoding. In feature selection, the total number of 30 features is subjected to an ANOVA test to find the K model, and from that, we select the best features for our model training.

The problems for class imbalance in the dataset and to rectify the class imbalance and types of balancing techniques are explained. The model training is conducted using the PyCaret module. The models and works are explained in the above section briefly. The types of evaluation metrics that are used for our study are also discussed and explained. More detailed insights of the dataset are given in the next section.

CHAPTER 4:
DESIGN AND ANALYTICS

4.1 Introduction

In the field of fraud detection, the design and analysis phase is crucial for developing reliable methods, advanced algorithms, and solid statistical frameworks. This ensures that the proposed detection mechanisms are effective and accurate in the constantly evolving landscape of financial security.

In this chapter, we thoroughly examine the dataset and carry out essential steps such as data pre-processing, cleaning, and data analysis to gain a better understanding. We provide a detailed explanation of the proposed methodology in each section. The data description section explains the dataset used in this research in detail, followed by data analysis and exploratory data analysis, which provides us with detailed information on each variable in the dataset. After the analysis, data preparation is explained, and data cleaning and transformation are discussed. Finally, we explain feature selection and data balancing that are used for this study.

4.2 Data description

Understanding the data used in research is crucial. Without the necessary knowledge of the data, the accuracy of predictions and overall performance can be hampered. In this section, we will discuss the type of data used in this study.

The most commonly used and only available dataset for fraud detection or insurance fraud, which contains multiple samples and high-class imbalance, was originally derived by a corporate website known as Angoss Knowledge Seeker software. This dataset is now available on open-source data platforms such as Kaggle (Vehicle Insurance Fraud Detection, 2023). The data consists of a total of 15420 claim instances, in which a total of 14497 (94%) claim instances are non-fraud and only 923 (6%) claim instances are fraud.

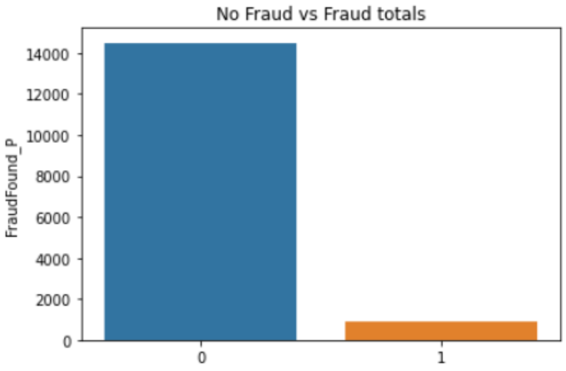
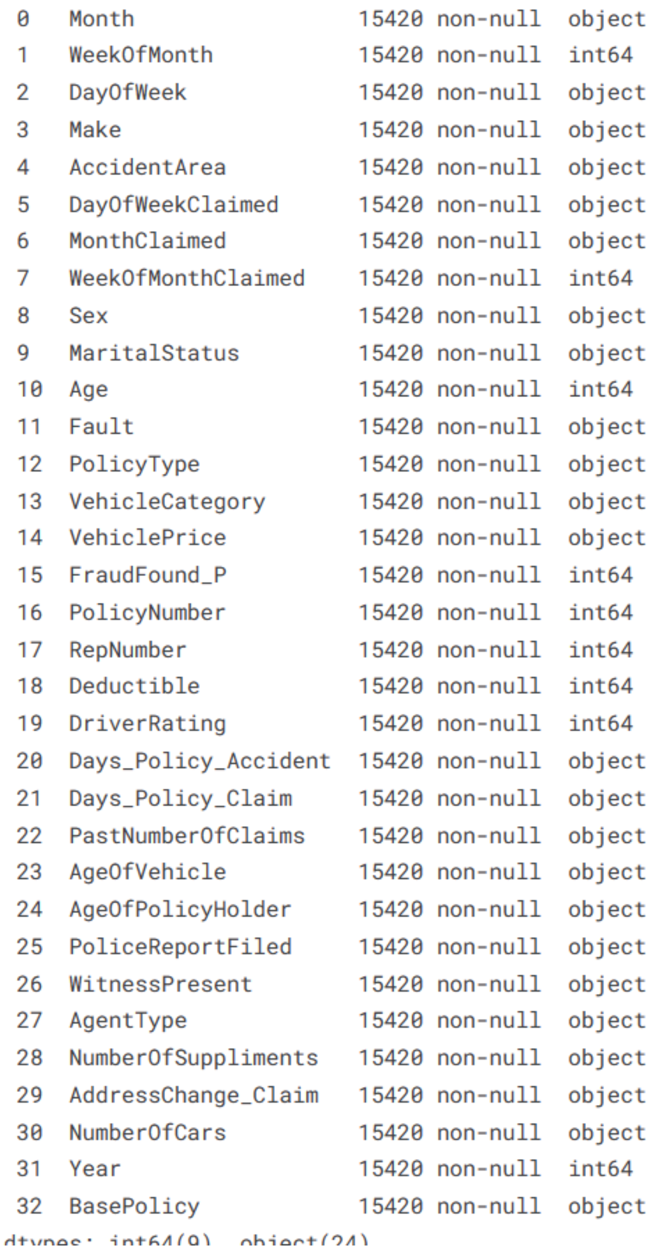


Figure 4.1 Fraud vs non- Fraud

Figure 4.1 explains the visual representation of Fraud and non-fraud in the dataset. The selected dataset has 2 different datatypes Numerical/Ordinal datatype and Nominal/Categorical datatype. The dataset consists of 8 numerical and 25 categorical datatype variables out of 33 variables. The List of features in the dataset is shown in Figure 4.2.



```
0   Month                 15420 non-null   object
1   WeekOfMonth           15420 non-null   int64
2   DayOfWeek             15420 non-null   object
3   Make                  15420 non-null   object
4   AccidentArea          15420 non-null   object
5   DayOfWeekClaimed      15420 non-null   object
6   MonthClaimed          15420 non-null   object
7   WeekOfMonthClaimed    15420 non-null   int64
8   Sex                   15420 non-null   object
9   MaritalStatus         15420 non-null   object
10  Age                   15420 non-null   int64
11  Fault                 15420 non-null   object
12  PolicyType            15420 non-null   object
13  VehicleCategory       15420 non-null   object
14  VehiclePrice          15420 non-null   object
15  FraudFound_P          15420 non-null   int64
16  PolicyNumber          15420 non-null   int64
17  RepNumber             15420 non-null   int64
18  Deductible            15420 non-null   int64
19  DriverRating          15420 non-null   int64
20  Days_Policy_Accident  15420 non-null   object
21  Days_Policy_Claim     15420 non-null   object
22  PastNumberOfClaims    15420 non-null   object
23  AgeOfVehicle          15420 non-null   object
24  AgeOfPolicyHolder     15420 non-null   object
25  PoliceReportFiled     15420 non-null   object
26  WitnessPresent        15420 non-null   object
27  AgentType             15420 non-null   object
28  NumberOfSuppliments   15420 non-null   object
29  AddressChange_Claim   15420 non-null   object
30  NumberOfCars          15420 non-null   object
31  Year                  15420 non-null   int64
32  BasePolicy            15420 non-null   object
dtypes: int64(9), object(24)
```

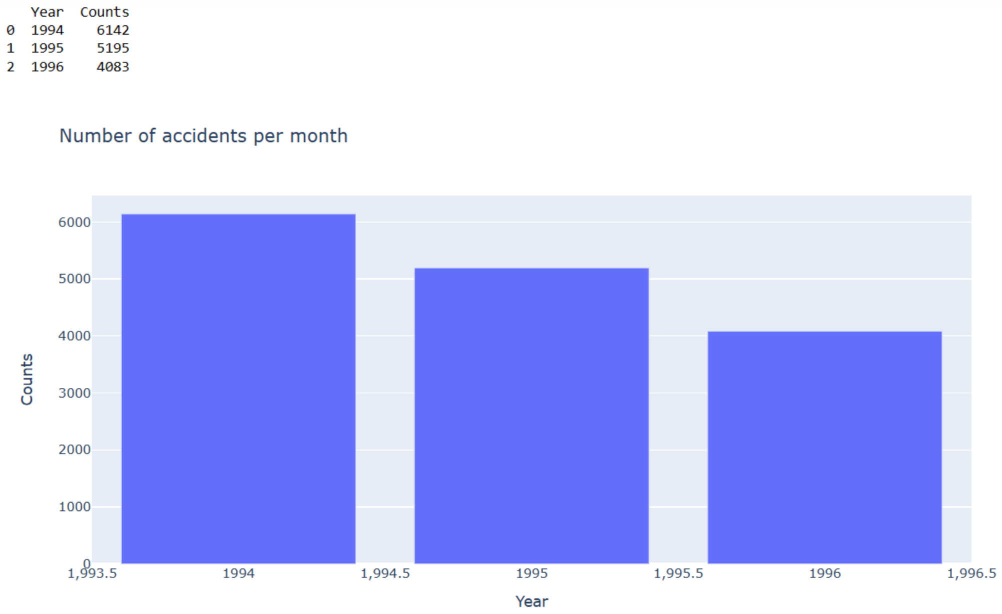
Figure 4.2 Original Data Types

The data is collected from the year January 1994 to December 1996, a 3-year worth of insurance claim data from the United States. The fraud insurance claimed in the year 1994 is about 409 and not-fraud is 5733 and the fraud rate happened in that year is 6.7% flowed in the year 1995 about 301 fraud claim cases have been registered and 4894 cases as not-fraud with 5.8% fraud rate and in the year 1996 the fraud claim is 213 and non-fraud claim is 3870 with 5.2% fraud rate (DeBarr and Wechsler, 2013).

Table 4.1 Fraud rate per year

Year	Fraud	Not-fraud	Total claims	Fraud rate
1994	409	5733	6142	6.7%
1995	301	4894	5195	5.8%
1996	213	3870	4083	5.2%

A graphical representation of years and fraud found is provided in Figure 4.3 and Table 4.1.



```
   Year  Counts
0  1994    6142
1  1995    5195
2  1996    4083
```

Figure 4.3 fraud found vs year

The numerical features in this dataset are of discrete type data and the categorical feature is of nominal type data. A detailed understanding of the data variables is provided in below Table 4.2.

Table 4.2 Data Understanding

Column name	Description
Month	Insurance Filed Month
WeekOfMonth	Insurance Filed Week
DayOfWeek	Insurance Filed Day
Make	Name of the Car
AccidentArea	place the accident took place
DayOfWeekClaimed	Insurance Claimed Day
MonthClaimed	Insurance Claimed Month
WeekOfMonthClaimed	Insurance Claimed Week
Sex	Male or Female
MaritalStatus	Marital status of the customer
Age	Age of the Customer
Fault	Accident Caused by
PolicyType	Insurance Type
VehicleCategory	Category of the Vehicle
VehiclePrice	Price of the Vehicle
PolicyNumber	Policy Serial Number
RepNumber	Representative Number
Deductible	Deductible amount
DriverRating	Rating of the driver
Days: Policy-Accident	Days the accident happened from the policy signed
Days: Policy-Claim	Days taken to claim the insurance
PastNumberOfClaims	Number of insurance claims previously
AgeOfVehicle	The age of the vehicle during the incident
AgeOfPolicyHolder	Policy holder age in the group
PoliceReportFiled	Police Report for Accident
WitnessPresent	incident witness
AgentType	type of the Agent
NumberOfSuppliments	Additional policy numbers
AddressChange-Claim	Days between address change and insurance claim
NumberOfCars	number of cars involved in the incident

Year	the year of the incident
BasePolicy	policy type
FraudFound	Fraud Yes or No

🟨 - Categorical features

🟧 - Numerical features

Out of these 33 variables, 1 dependent variable (target variable) with data in ‘Yes’ or ‘No” type is ‘FraudFound’ the other 32 independent variables are used for prediction.

For a better understanding of the data the class are classified (Itri et al., 2019) as

- Personal data of the insured (gender, age, sex, marital status, etc.)
- Insurance details (Policy type, base policy, deductible, supplements, agent type, number of insurance claims, driver rating, etc.)
- Accident circumstance (accident area, police report filed, date of accident, fault liability, witness present, etc.)
- Target carriable (FraudFound)

Now that we have a basic understanding of the data, we can do an in-depth analysis of each variable in the EDA segment. The EDA can be carried out only when the data are in the correct format.

4.3 Exploratory Data Analysis

The idea of Exploratory Data Analysis (EDA) is to clarify the general structure of the data, obtain simple descriptive summaries and perhaps get ideas for a more sophisticated analysis (Chatfield, 1986). Exploratory data analysis is a powerful technique for analysing and summarising complex datasets. By using statistical and visualisation methods, we can gain a comprehensive understanding of our data and extract the most relevant features for our machine learning models. Through analysing the frequency and correlation between features, we can identify highly uncorrelated features and eliminate them to improve the accuracy of our algorithms.

In this study, we performed EDA (Exploratory Data Analysis) using the Univariate analysis method. This technique helps researchers and data analysts to understand the characteristics of individual variables before exploring their relationships with other variables using bivariate and multivariate analyses. Univariate analysis is a fundamental step in the EDA process, and it provides a foundation for making informed decisions about data preprocessing, transformation, and potential outliers before more complex analyses are performed.

Visualizing the correlation of our dataset in Figure 8 helps us simplify our model, reduce learning time, and ultimately increase the precision of our predictions. Exploratory data analysis enables us to make more informed decisions and achieve better results in our machine learning projects.

In our study, understanding the relationship between all the columns is crucial. Figure 4.4 Pearson's correlation image is a valuable tool that provides us with insights into which columns are related to each other and consist of the same values. When such columns are present, they can have a significant impact on the output. Therefore, we removed the columns that shared the same values.

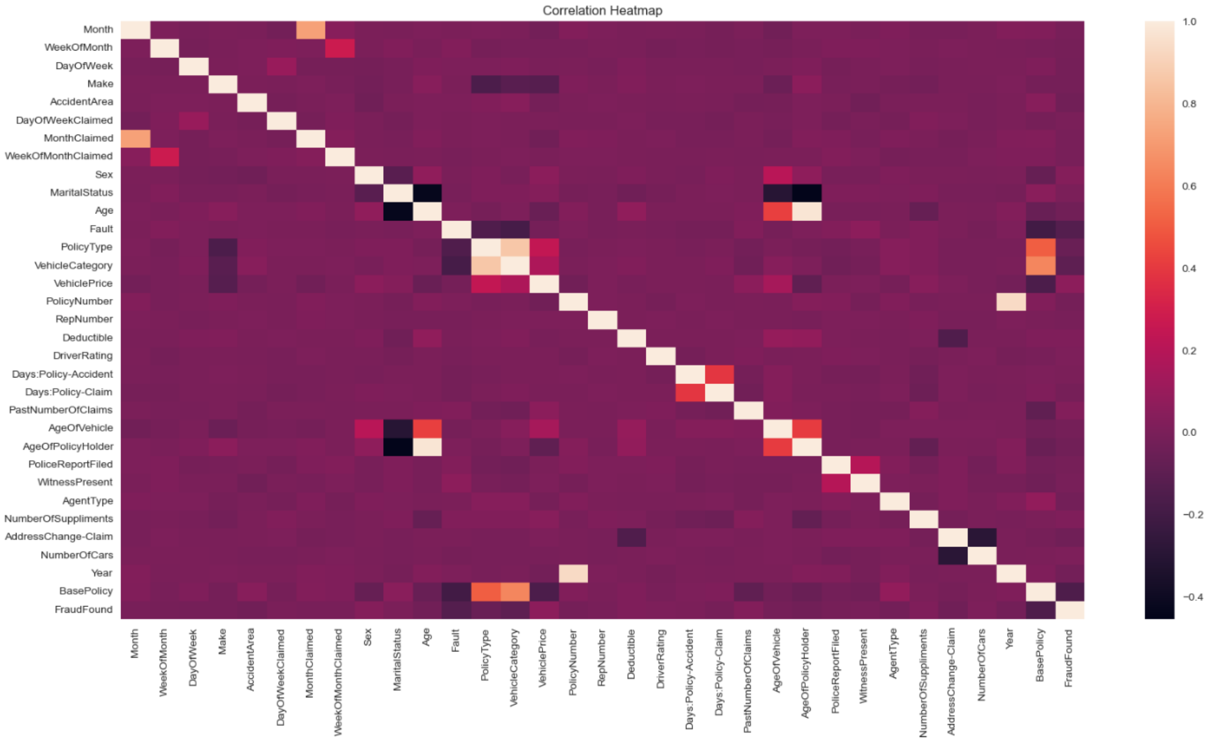
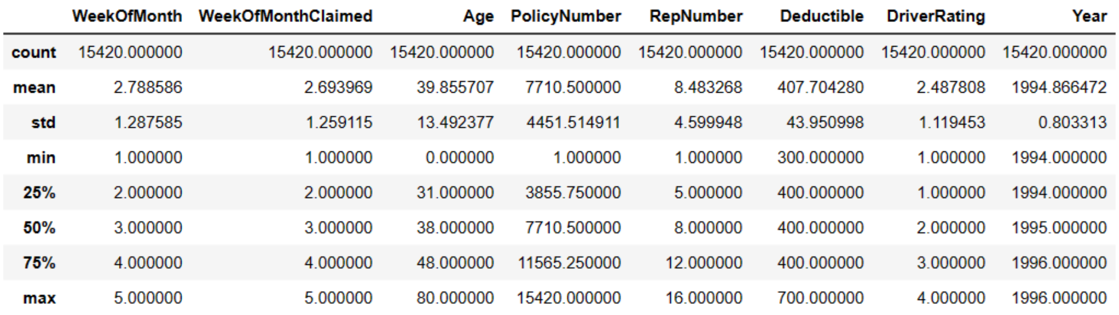


Figure 4.4 Data Correlation

Our objective in this study is to analyse the "FraudFound" variable. We conducted exploratory data analysis on all the relevant columns of our dataset. Figure 4.1 presents a visual representation of the "FraudFound" column. Based on the image, we can conclude that less than 6% of the insurance claims (totalling 15,000) were identified as fraudulent.

The summary statistic of the given dataset is provided in Figure 4.5, as seen out of 33 variables only 8 variables are resulted as they are of ordinal values.

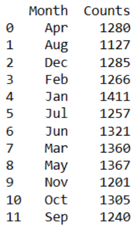


	WeekOfMonth	WeekOfMonthClaimed	Age	PolicyNumber	RepNumber	Deductible	DriverRating	Year
count	15420.000000	15420.000000	15420.000000	15420.000000	15420.000000	15420.000000	15420.000000	15420.000000
mean	2.788586	2.693969	39.855707	7710.500000	8.483268	407.704280	2.487808	1994.866472
std	1.287585	1.259115	13.492377	4451.514911	4.599948	43.950998	1.119453	0.803313
min	1.000000	1.000000	0.000000	1.000000	1.000000	300.000000	1.000000	1994.000000
25%	2.000000	2.000000	31.000000	3855.750000	5.000000	400.000000	1.000000	1994.000000
50%	3.000000	3.000000	38.000000	7710.500000	8.000000	400.000000	2.000000	1995.000000
75%	4.000000	4.000000	48.000000	11565.250000	12.000000	400.000000	3.000000	1996.000000
max	5.000000	5.000000	80.000000	15420.000000	16.000000	700.000000	4.000000	1996.000000

Figure 4.5 Statical Summary

After only analysing multiple variables, we can have a better understanding of the columns through exploratory data analysis. We can conduct an EDA on all the variables from the transformed data.

From this dataset, we will follow the top-down approach. From the first variable to the last variable, we will understand each feature and how it relates to fraud found feature. As shown in Figure 4.6, we can see that in January the highest number of insurances have been claimed. These months are a combined data of all three years and the lowest claims are from August.



```
   Month  Counts
0    Apr    1280
1    Aug    1127
2    Dec    1285
3    Feb    1266
4    Jan    1411
5    Jul    1257
6    Jun    1321
7    Mar    1360
8    May    1367
9    Nov    1201
10   Oct    1305
11   Sep    1240
```

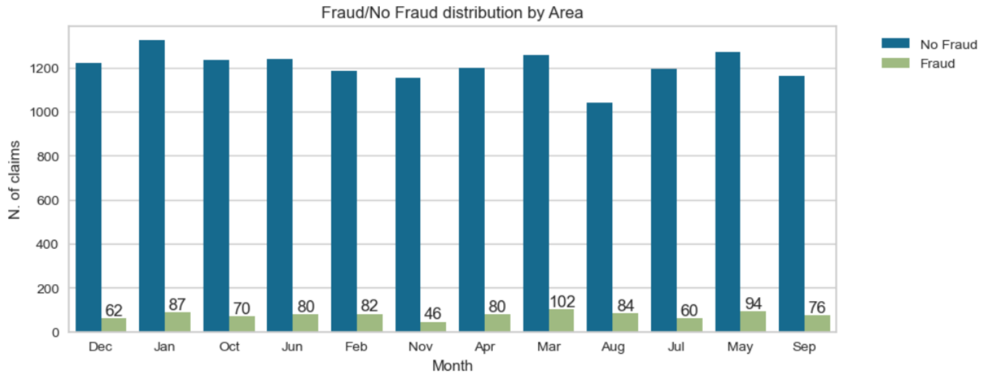
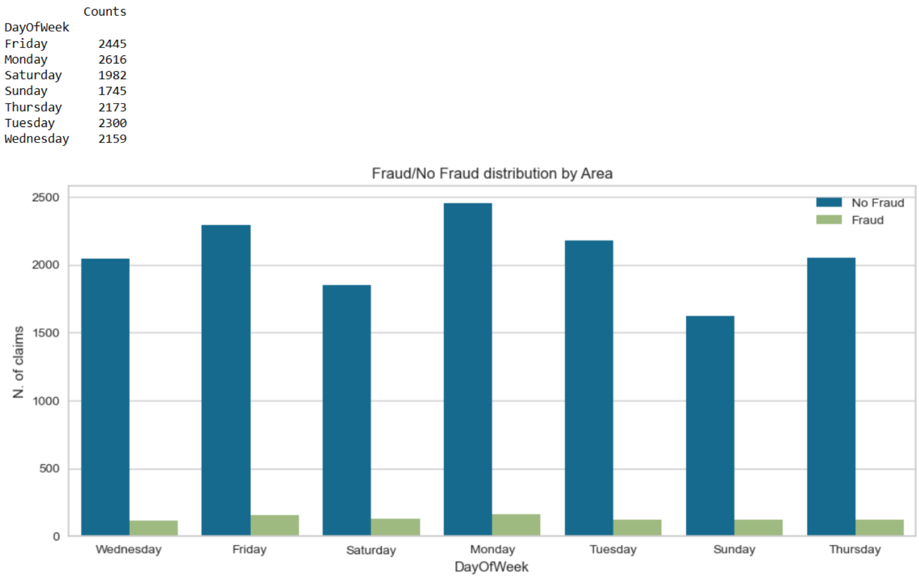


Figure 4.6 Fraud Found Vs Months

From Figure 4.7, we can also understand that in March the number of claims combined for all three years is 102 and, in the month, November is the lowest with 46 claims. From all three years combined, the total number of insurances reported is higher on Monday with a total count of 2616 and is lower on Sunday with 1745 total claims reported but the fraud report is lower on Wednesday.



```
              Counts
DayOfWeek
Friday          2445
Monday          2616
Saturday        1982
Sunday          1745
Thursday        2173
Tuesday         2300
Wednesday       2159
```

Figure 4.7 DayofWeek vs Fraud found

Automobile insurance is based on the insurance a car claimed, based on Figure 4.8 we know that the Pontiac car model has the highest number of claims totalling about 3624 with the highest fraud claim followed by Toyota claiming 2935.

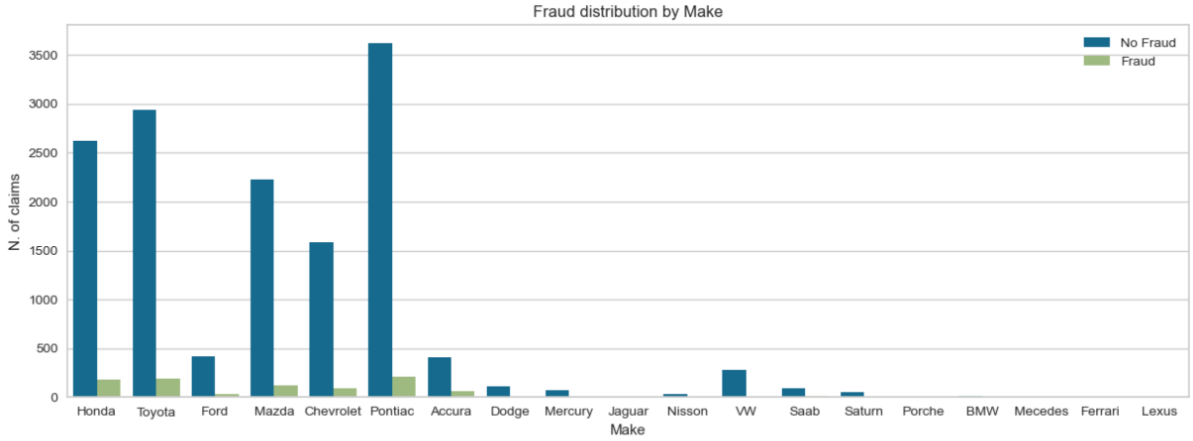
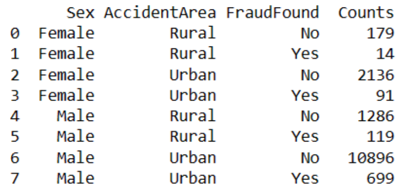


Figure 4.8 Make vs Fraud found

The area where an accident takes place can have a lot of information. As shown in Figure 4.9, the accidents where more insurance is claimed are urban with 13822 and rural with 1598. In this, the fraud claims by males in urban are 699 and females are 91. Detailed info can be shown in Figure 4.9. From this, we can conclude that male has claimed the most fraud claim urban areas.



```
      Sex AccidentArea FraudFound  Counts
0  Female        Rural         No     179
1  Female        Rural        Yes      14
2  Female        Urban         No    2136
3  Female        Urban        Yes      91
4    Male        Rural         No    1286
5    Male        Rural        Yes     119
6    Male        Urban         No   10896
7    Male        Urban        Yes     699
```

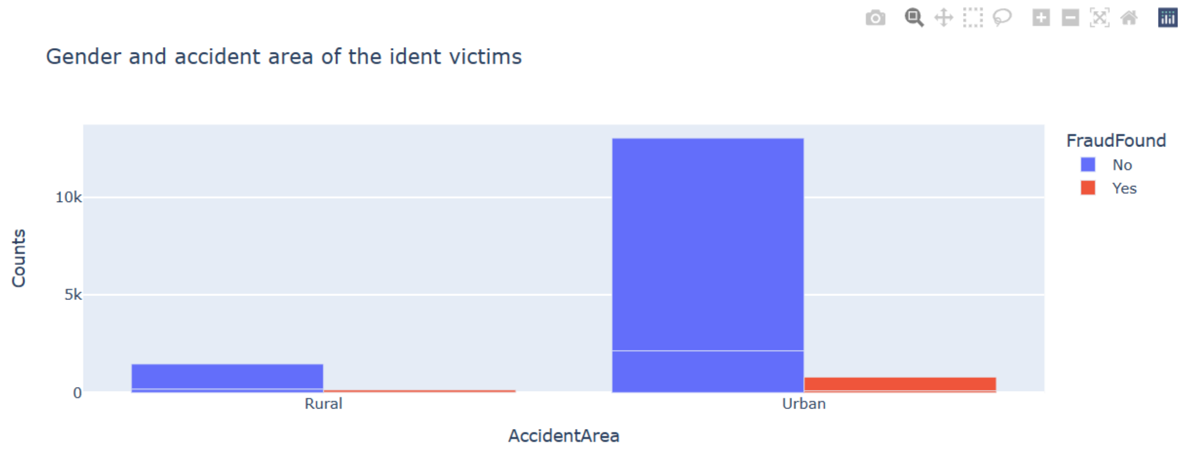


Figure 4.9 Accident area vs fraud found

The ‘DayOfWeekClaimed’, ‘MonthClaimed’, and ‘WeekOfMonthClaimed’ is also a combination of 3 years of data based on the data we can know that the insurance claimed is higher on the 3rd week and Mondays. Figures 4.10 and 4.11 represent the fraud claimed and reported days, months and weeks.

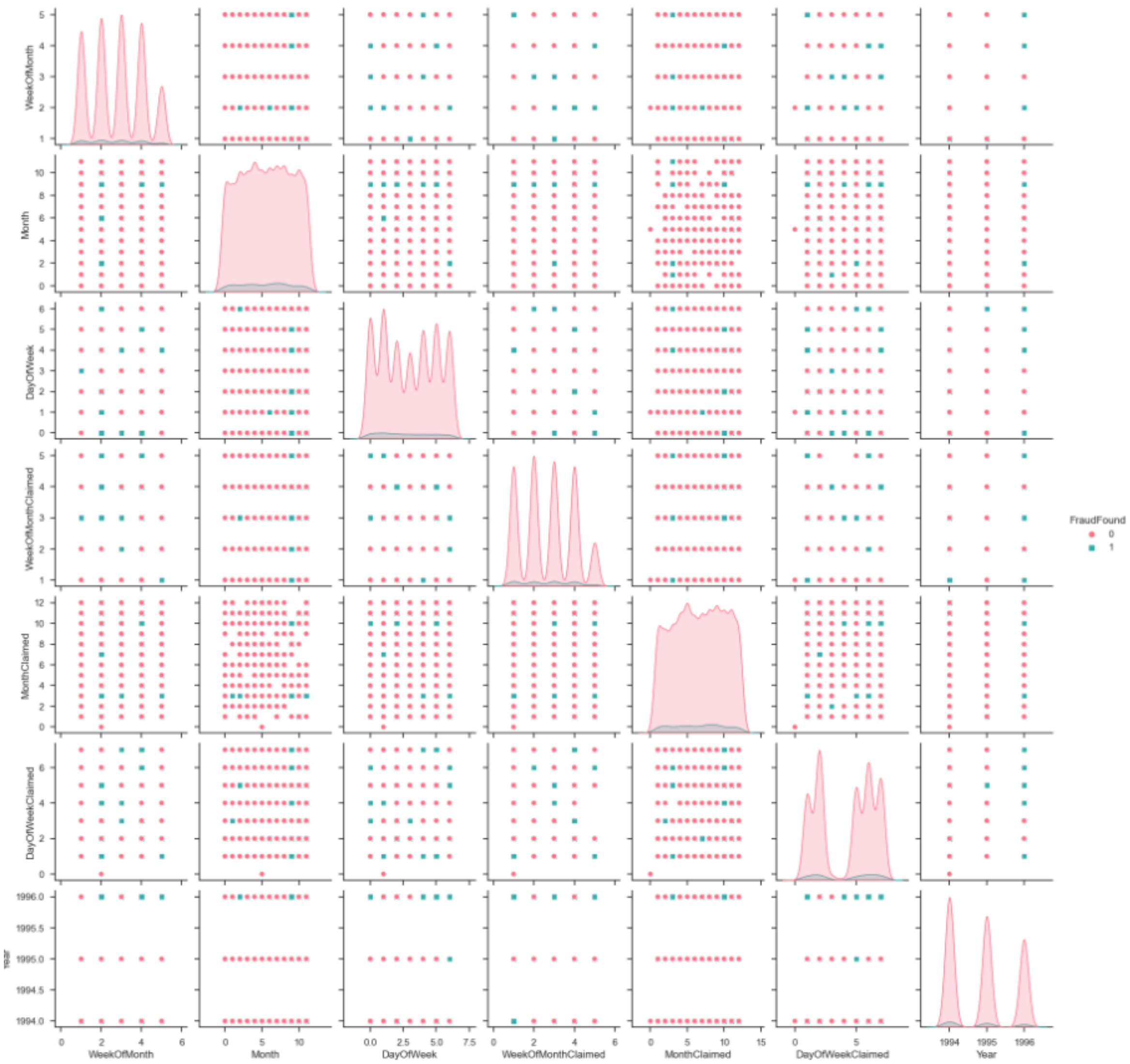
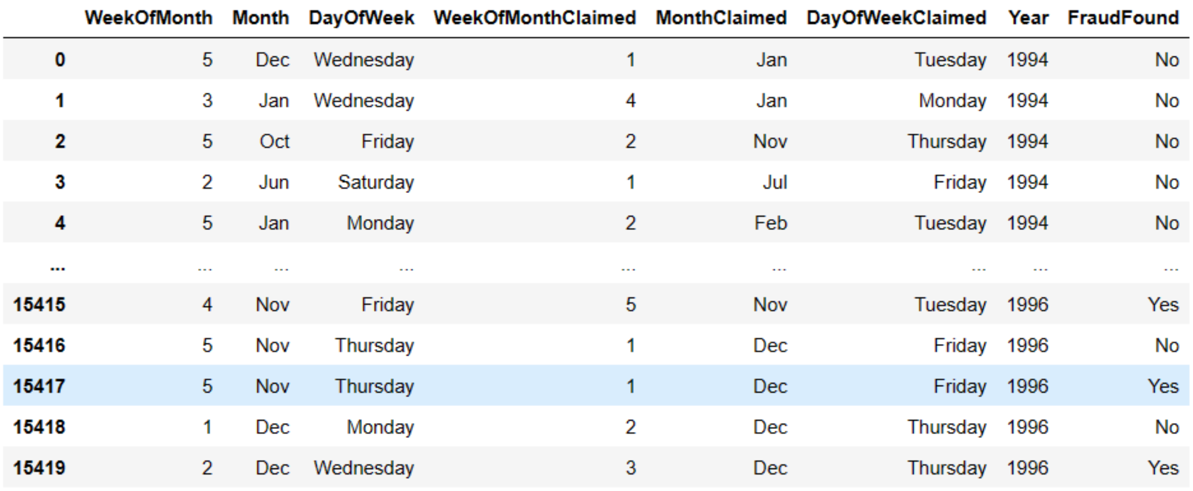


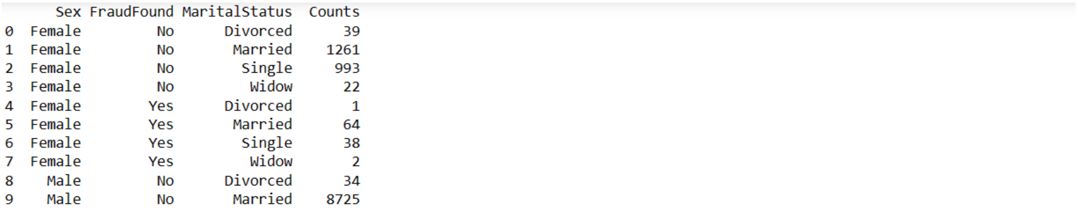
Figure 4.10 Reported vs claimed 1



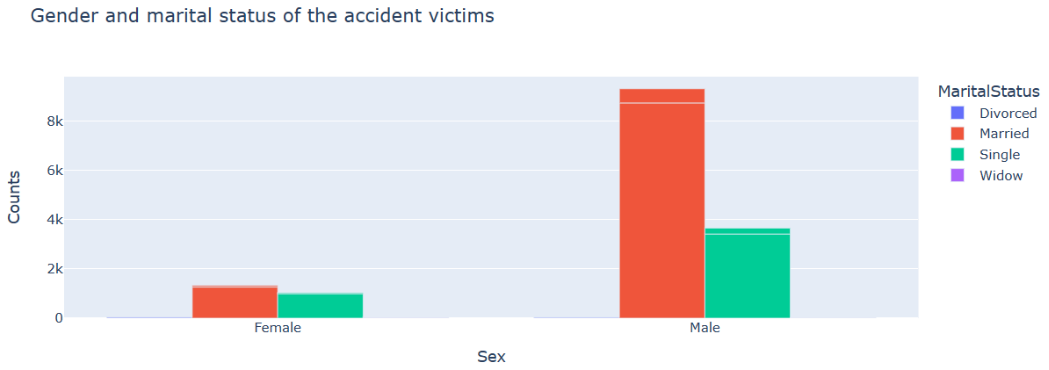
	WeekOfMonth	Month	DayOfWeek	WeekOfMonthClaimed	MonthClaimed	DayOfWeekClaimed	Year	FraudFound
0	5	Dec	Wednesday	1	Jan	Tuesday	1994	No
1	3	Jan	Wednesday	4	Jan	Monday	1994	No
2	5	Oct	Friday	2	Nov	Thursday	1994	No
3	2	Jun	Saturday	1	Jul	Friday	1994	No
4	5	Jan	Monday	2	Feb	Tuesday	1994	No
...	...	...	...	...	...	...	...	...
15415	4	Nov	Friday	5	Nov	Tuesday	1996	Yes
15416	5	Nov	Thursday	1	Dec	Friday	1996	No
15417	5	Nov	Thursday	1	Dec	Friday	1996	Yes
15418	1	Dec	Monday	2	Dec	Thursday	1996	No
15419	2	Dec	Wednesday	3	Dec	Thursday	1996	Yes

Figure 4.11 Reported vs Claimed 2

Many insurance claims have been filed by males totalling 13,000 of which 818 of them are fraud claims and out of 2420 female reports 105 of them were fraud claims. Figure 4.12 shows that males have claimed more fraud insurance



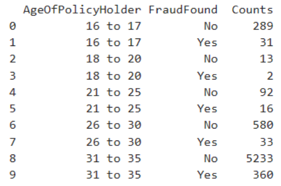
```
      Sex FraudFound MaritalStatus  Counts
0  Female         No      Divorced      39
1  Female         No       Married    1261
2  Female         No        Single     993
3  Female         No         Widow      22
4  Female        Yes      Divorced       1
5  Female        Yes       Married      64
6  Female        Yes        Single      38
7  Female        Yes         Widow       2
8    Male         No      Divorced      34
9    Male         No       Married    8725
```



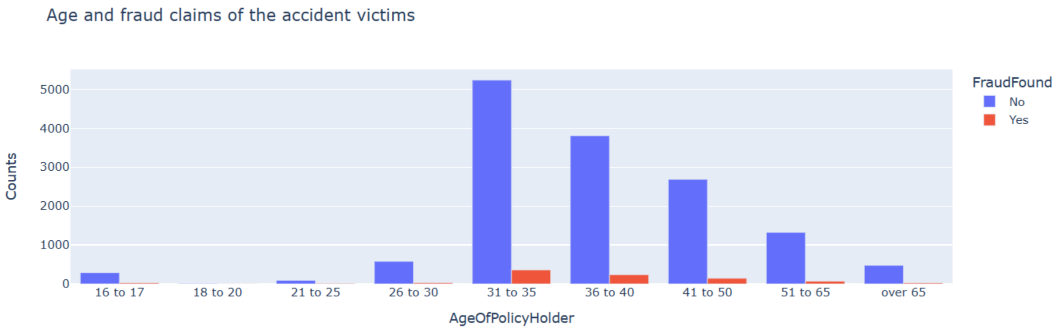
Gender and marital status of the accident victims

Figure 4.12 Fraud found vs Sex

Further analysis revealed that men who are married, from Figure 4.12, have reported claiming more fraud insurance that are in the age group of 31 to 35, figure 4.13. This proves that men claim more fraud when they are married and middle-aged men claim more fraud insurance.

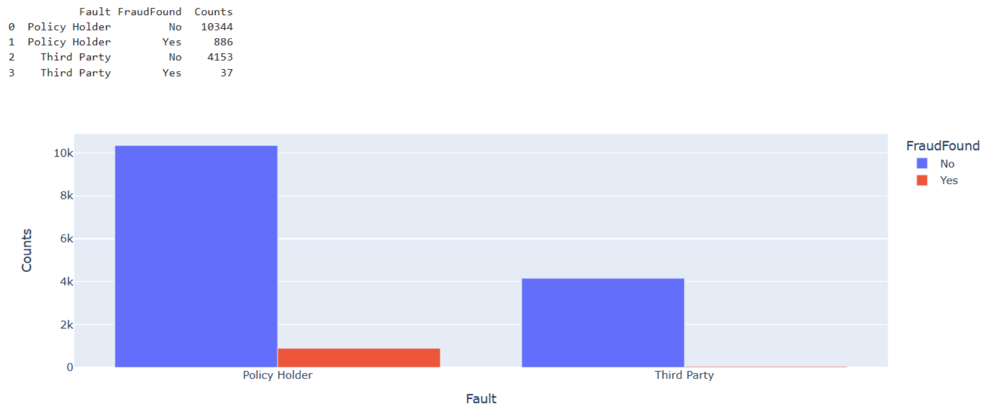


```
  AgeOfPolicyHolder FraudFound  Counts
0          16 to 17         No     289
1          16 to 17        Yes      31
2          18 to 20         No      13
3          18 to 20        Yes       2
4          21 to 25         No      92
5          21 to 25        Yes      16
6          26 to 30         No     580
7          26 to 30        Yes      33
8          31 to 35         No    5233
9          31 to 35        Yes     360
```



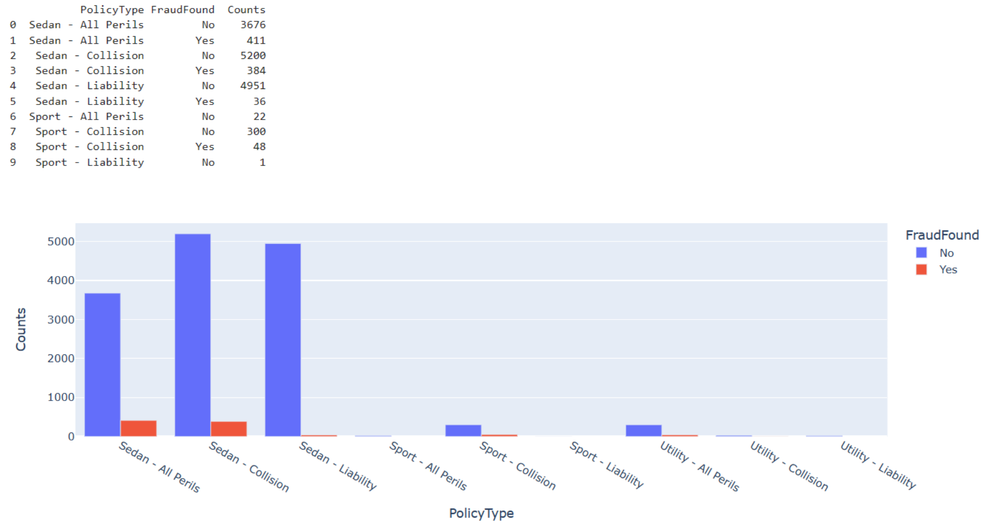
Age and fraud claims of the accident victims

Figure 4.13 Fraud Found vs Age



```
        Fault FraudFound  Counts
0  Policy Holder         No   10344
1  Policy Holder        Yes     886
2    Third Party         No    4153
3    Third Party        Yes      37
```

Figure 4.14 Fault vs Fraud Found



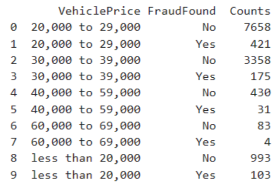
```
         PolicyType FraudFound  Counts
0  Sedan - All Perils         No    3676
1  Sedan - All Perils        Yes     411
2   Sedan - Collision         No    5200
3   Sedan - Collision        Yes     384
4   Sedan - Liability         No    4951
5   Sedan - Liability        Yes      36
6  Sport - All Perils         No      22
7   Sport - Collision         No     300
8   Sport - Collision        Yes      48
9   Sport - Liability         No       1
```

Figure 4.15 Policy Type vs Fraud Found

From figures 4.14 & 4.15, we can conclude that most fraud claims are conducted by policyholders rather than a third party and the most filed policy type is those who used sedan cards with policy type ‘ALL Perils’ followed by ‘Collusion’. The reason for this trend is the safety factor of the vehicle when staging the accidents.

Most fraud claims are filed for vehicles that are in the price range of 20,000 to 29,000 with a count of 421 claims out of 8079 and the driver rating of these fraud claims is higher with rating number 3 followed by rating 1. Figure 4.16 & 4.17 show, that people having a driver rating of 3 that have a vehicle in the price range of 20,000 to 29,000 tends to file more fraud claim than others.

In Figure 4.18, we can see that a vehicle with an age of 7 years have filed more insurance claim with a count of 5807 of which the total fraud claim is 325.



```
      VehiclePrice FraudFound  Counts
0  20,000 to 29,000         No    7658
1  20,000 to 29,000        Yes     421
2  30,000 to 39,000         No    3358
3  30,000 to 39,000        Yes     175
4  40,000 to 59,000         No     430
5  40,000 to 59,000        Yes      31
6  60,000 to 69,000         No      83
7  60,000 to 69,000        Yes       4
8  less than 20,000         No     993
9  less than 20,000        Yes     103
```

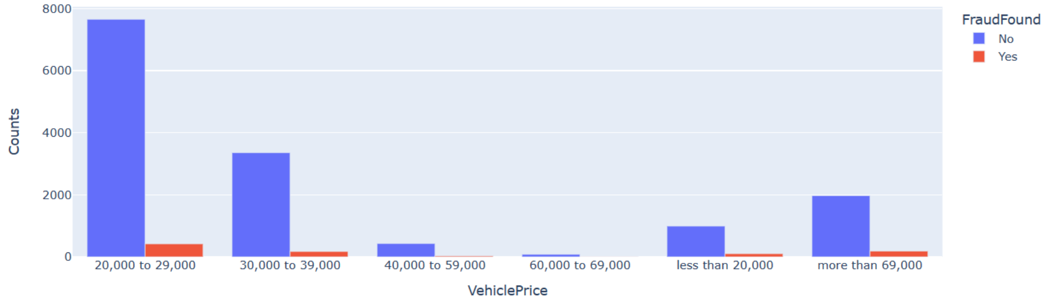
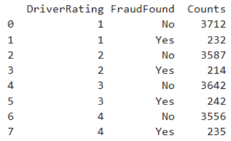


Figure 4.16 Fraud found vs Vehicle price



```
   DriverRating FraudFound  Counts
0             1         No    3712
1             1        Yes     232
2             2         No    3587
3             2        Yes     214
4             3         No    3642
5             3        Yes     242
6             4         No    3556
7             4        Yes     235
```

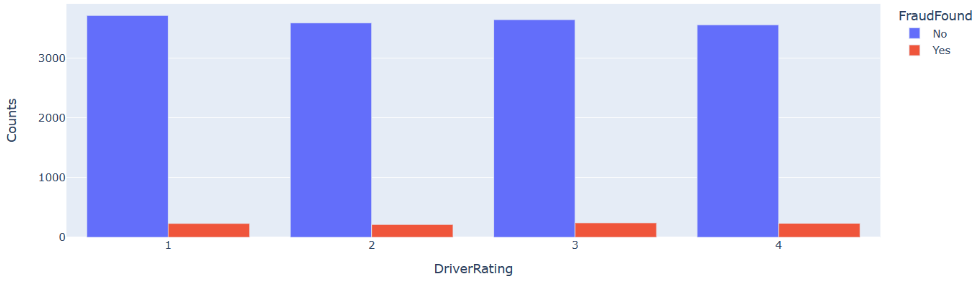
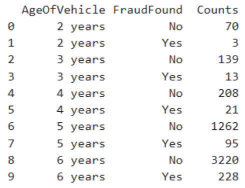


Figure 4.17 Fraud Found vs Driver Rating



```
  AgeOfVehicle FraudFound  Counts
0      2 years         No      70
1      2 years        Yes       3
2      3 years         No     139
3      3 years        Yes      13
4      4 years         No     208
5      4 years        Yes      21
6      5 years         No    1262
7      5 years        Yes      95
8      6 years         No    3220
9      6 years        Yes     228
```

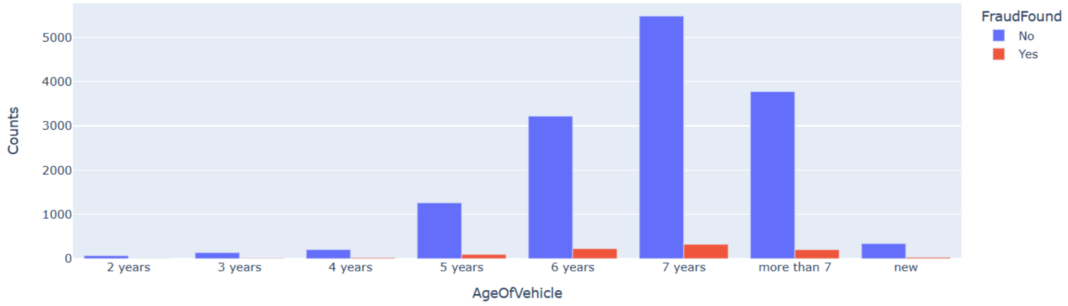
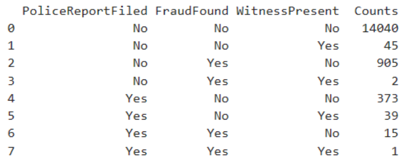


Figure 4.18 Fraud found vs age of vehicle



```
   PoliceReportFiled FraudFound WitnessPresent  Counts
0                 No         No             No   14040
1                 No         No            Yes      45
2                 No        Yes             No     905
3                 No        Yes            Yes       2
4                Yes         No             No     373
5                Yes         No            Yes      39
6                Yes        Yes             No      15
7                Yes        Yes            Yes       1
```

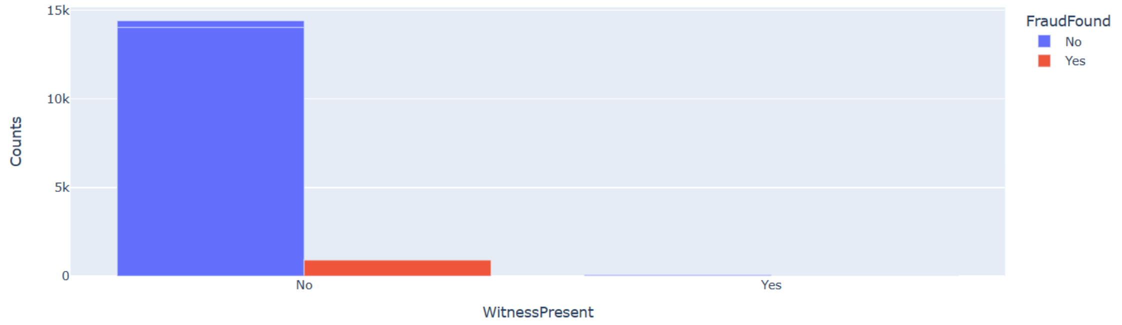
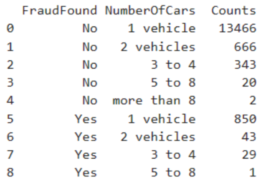


Figure 4.19 Fraud Found vs Witness & police

The insurance claim can only be done when the police have filed a complaint and need a witness for a report. In Figure 4.19, we can see that a total of 905 fraud insurance has been claimed when there is no police report filed and no witness present during the report filed.



```
  FraudFound NumberOfCars  Counts
0         No    1 vehicle   13466
1         No   2 vehicles     666
2         No       3 to 4     343
3         No       5 to 8      20
4         No  more than 8       2
5        Yes    1 vehicle     850
6        Yes   2 vehicles      43
7        Yes       3 to 4      29
8        Yes       5 to 8       1
```

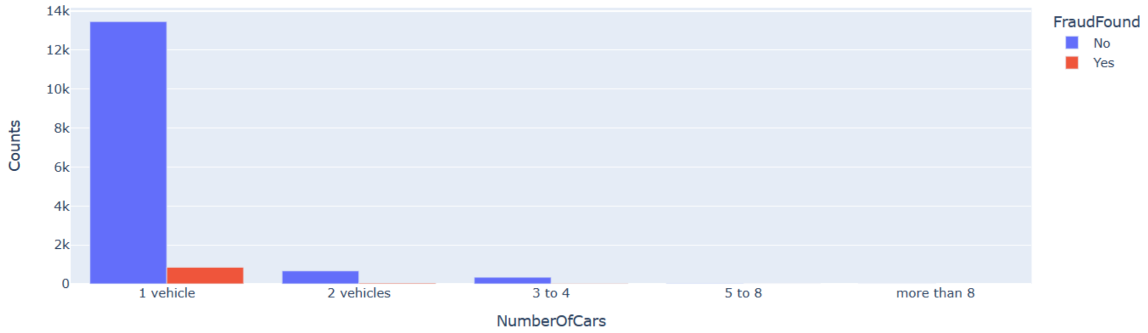
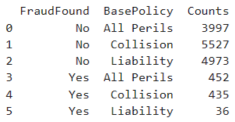


Figure 4.20 Fraud Found vs No.of Cars

In the collective data for 3 years of insurance, a total of 850 insurance claims filed were fraud where only one vehicle has been reported for insurance claim. Figure 4.20 shows that the vehicle that is bought within a year has more insurance claims. Also, the fraud insurance that has been claimed by using an external agent is higher which amounts to 919 claims.



```
  FraudFound  BasePolicy  Counts
0         No  All Perils    3997
1         No   Collision    5527
2         No   Liability    4973
3        Yes  All Perils     452
4        Yes   Collision     435
5        Yes   Liability      36
```

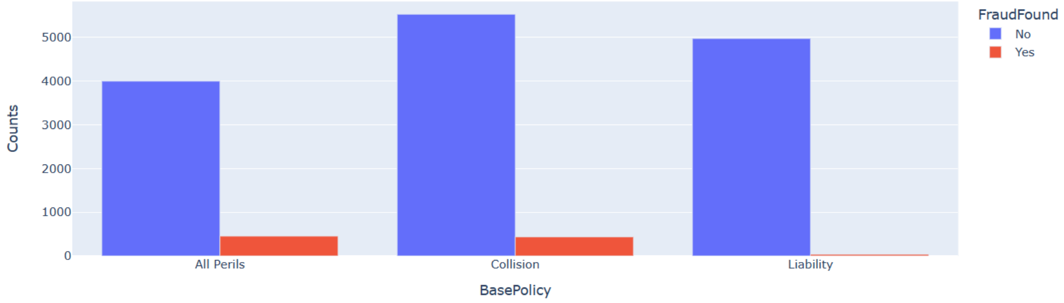


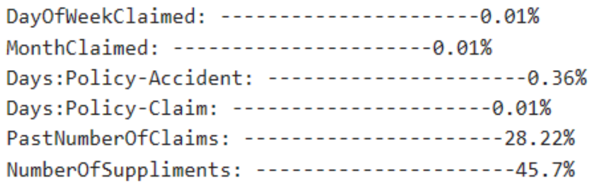
Figure 4.21 Fraud Found vs Base Policy

From Figure 4.21 based on the policy provided, we can see that ‘All Perils’ and ‘Collusion’ have the highest fraud insurance claiming rate.

After conducting an exploratory data analysis, we found out that male drivers aged between 31 and 35, who own sedan cars from Pontiac and Toyota, priced between 20,000 to 29,000 and are based in urban areas, have taken policies covering 'All Perils' and 'Collision' for their cars. Furthermore, we discovered that these drivers tend to file fraudulent claims in the absence of a police report or witness for the incident.

4.4 Data Preparation

In this section, we will perform data preprocessing steps such as data cleaning and transformation. Data cleaning is a crucial task in machine learning as it helps to remove unwanted data from the file or fill up any missing values.



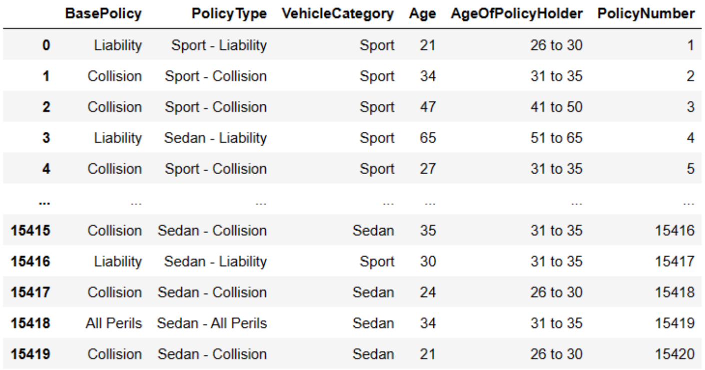
```
DayOfWeekClaimed: ---------------------0.01%
MonthClaimed: ---------------------0.01%
Days:Policy-Accident: ---------------------0.36%
Days:Policy-Claim: ---------------------0.01%
PastNumberOfClaims: ---------------------28.22%
NumberOfSuppliments: ---------------------45.7%
```

Figure 4.22 Missing values

The data in Figure 4.22 shows that certain values in the dataset have no-data values such as 'none' and '0', but these values still have meaning. For example, in the 'PastNumberOfClaims' feature, a value of 'none' indicates that the customer has not previously made any claims. Therefore, we can conclude that there are no 'NULL' values in the provided dataset. Figure 4.2 also shows that there are no missing values and that no data cleaning is necessary for our dataset.

However, datasets with repeated values can make the computational time of the system longer. Therefore, we need to drop such irrelevant variables. In this dataset, we identified the following columns as not relevant to our "Target Variable": 'Age', 'AgeOfPolicyHolder', 'BasePolicy', 'PolicyType', 'VehicleCategory', and 'PolicyNumber'. Figure 4.23 shows the common features that have repeated variables. We will use a heat map (Pearson's coefficient) to drop some columns that are not relevant to our "Target Variable."

A heatmap is a graphical representation of data where values in a matrix are represented using colors as shown in Figure 4.24. It is a way of visualizing data in a 2D space, where each point in the matrix is assigned, a color based on its value. Typically, the colors range from cool to warm hues, with cool colors representing lower values and warm colors representing higher values.



	BasePolicy	PolicyType	VehicleCategory	Age	AgeOfPolicyHolder	PolicyNumber
0	Liability	Sport - Liability	Sport	21	26 to 30	1
1	Collision	Sport - Collision	Sport	34	31 to 35	2
2	Collision	Sport - Collision	Sport	47	41 to 50	3
3	Liability	Sedan - Liability	Sport	65	51 to 65	4
4	Collision	Sport - Collision	Sport	27	31 to 35	5
...	...	...	...	...	...	...
15415	Collision	Sedan - Collision	Sedan	35	31 to 35	15416
15416	Liability	Sedan - Liability	Sport	30	31 to 35	15417
15417	Collision	Sedan - Collision	Sedan	24	26 to 30	15418
15418	All Perils	Sedan - All Perils	Sedan	34	31 to 35	15419
15419	Collision	Sedan - Collision	Sedan	21	26 to 30	15420

Figure 4.23 Irrelevant Data

To perform data implementation, it is necessary to transform the features into numerical values. Since the dataset we are dealing with contains both numerical and categorical values, data transformation is required. For example, the 'FraudFound' target variable is a categorical variable with values 'Yes' and 'No'. This feature needs to be transformed into '1' and '0' for classification in the ML model.

While data analysis can be carried out using the value count method, data transformation is essential for classification purposes. In this particular case, Label Encoding is used for data transformation. Label Encoding is a popular encoding technique used for handling categorical variables. It represents each label with a unique integer or alphabetical ordering. We are using label encoding instead of one-hot encoding to reduce the number of variables. One-hot encoding would create more dummy variables that would need to be rearranged after data transformation. The difference in data transformation can be seen in Figures 4.2 & 4.25, where the categorical variable 'Object' has been changed to numerical values of 'int32'.

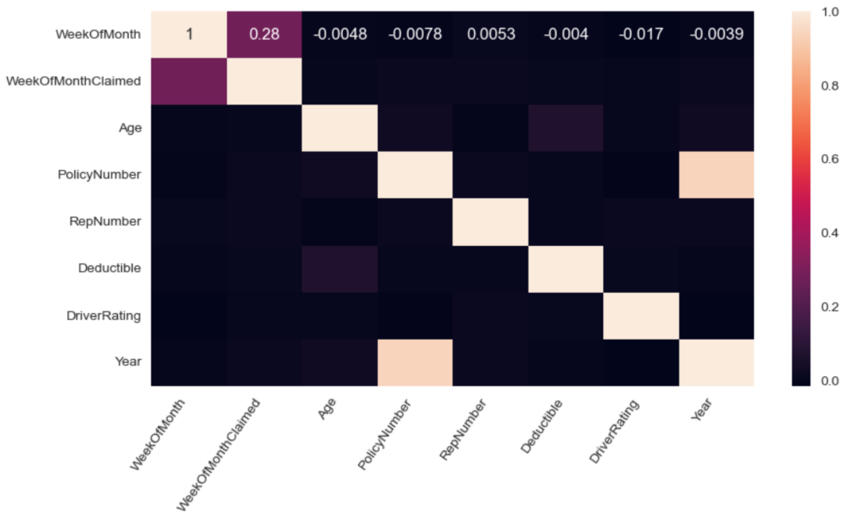
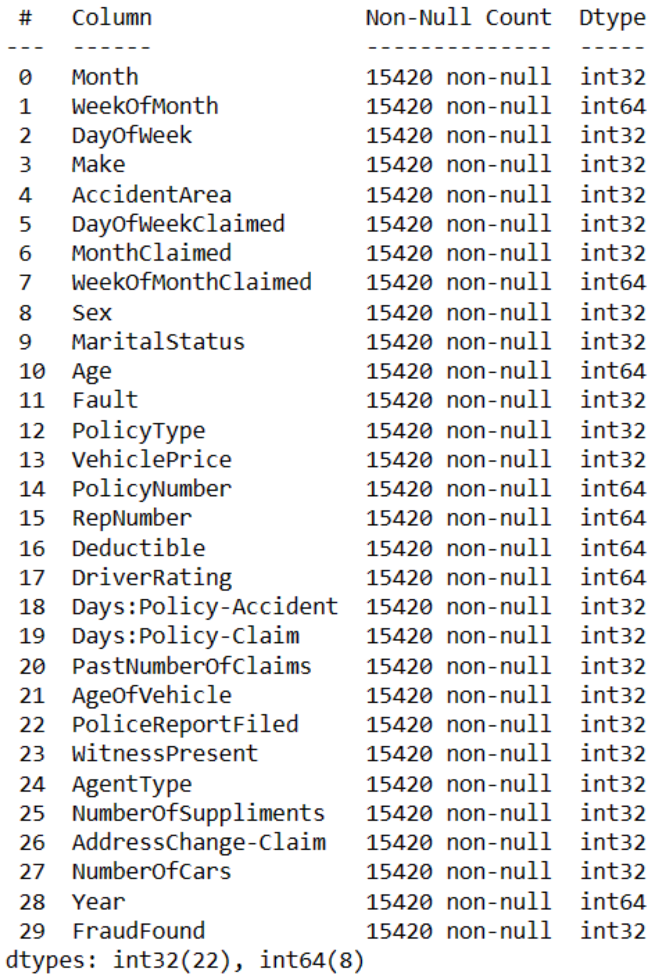


Figure 4.24 Correlation map



```
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Month                  15420 non-null  int32
 1   WeekOfMonth            15420 non-null  int64
 2   DayOfWeek              15420 non-null  int32
 3   Make                   15420 non-null  int32
 4   AccidentArea           15420 non-null  int32
 5   DayOfWeekClaimed       15420 non-null  int32
 6   MonthClaimed           15420 non-null  int32
 7   WeekOfMonthClaimed     15420 non-null  int64
 8   Sex                    15420 non-null  int32
 9   MaritalStatus          15420 non-null  int32
 10  Age                    15420 non-null  int64
 11  Fault                  15420 non-null  int32
 12  PolicyType             15420 non-null  int32
 13  VehiclePrice           15420 non-null  int32
 14  PolicyNumber           15420 non-null  int64
 15  RepNumber              15420 non-null  int64
 16  Deductible             15420 non-null  int64
 17  DriverRating           15420 non-null  int64
 18  Days:Policy-Accident   15420 non-null  int32
 19  Days:Policy-Claim      15420 non-null  int32
 20  PastNumberOfClaims     15420 non-null  int32
 21  AgeOfVehicle           15420 non-null  int32
 22  PoliceReportFiled      15420 non-null  int32
 23  WitnessPresent         15420 non-null  int32
 24  AgentType              15420 non-null  int32
 25  NumberOfSuppliments    15420 non-null  int32
 26  AddressChange-Claim    15420 non-null  int32
 27  NumberOfCars           15420 non-null  int32
 28  Year                   15420 non-null  int64
 29  FraudFound             15420 non-null  int32
dtypes: int32(22), int64(8)
```

Figure 4.25 Transformed Data

This transformed data can be used for further pre-processing steps like feature selection and data balancing process.

Figure 4.24, shows the visual representation of integer values in the original dataset which only has 8 features. After the data transformation in Figure 4.26 can see, that the categorical values are changed to numerical values. To find the correlation between the transformed values, a feature correlation check algorithm has been performed with a threshold limit set to 0.7. Based on the heat map -1 represents the lowest correlation and 1 represents the high correlation. When the limit is set to 0.7 the features higher than that can be considered highly correlated and will not be useful for the ML model. Since it represents, that the values in those features are of same. Based on this check, 4 features ‘Age’, 'MonthClaimed', 'VehicleCategory' and ‘PolicyNumber’ have been removed which can be viewed in Figure 4.26.

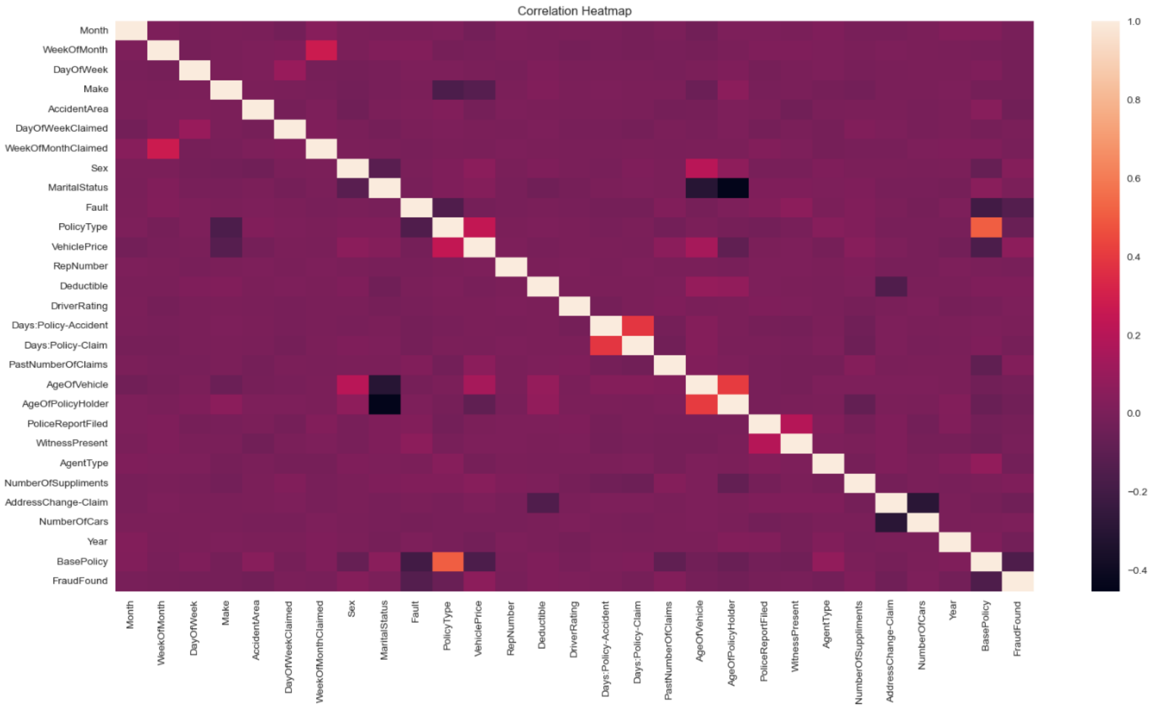


Figure 4.26 correlation after features dropped

The remaining 29 features will be subjected to the feature selection method and the best features from that will be used for ML model implementation.

PyCaret provides an encoding parameter in its environment. By default, it uses one-hot encoding for basic encoding. However, if you wish to use different encoding methods, you must implement them before feeding the data into PyCaret.

4.5 Feature Selection and Class Balancing

In this section, we will understand the feature selection and class balancing techniques used for this study.

4.5.1 Feature Selection

Feature selection plays a critical role in machine learning for several reasons. Firstly, it enhances model performance by reducing the time required for training. Secondly, selecting fewer features helps us gain a better understanding of the model, thereby improving its interpretability. Thirdly, it helps avoid multicollinearity. By choosing important features, we can improve the quality of the data, which is essential for scalability when dealing with high-dimensional datasets. By reducing the size of the data, feature selection also helps in optimizing storage space.

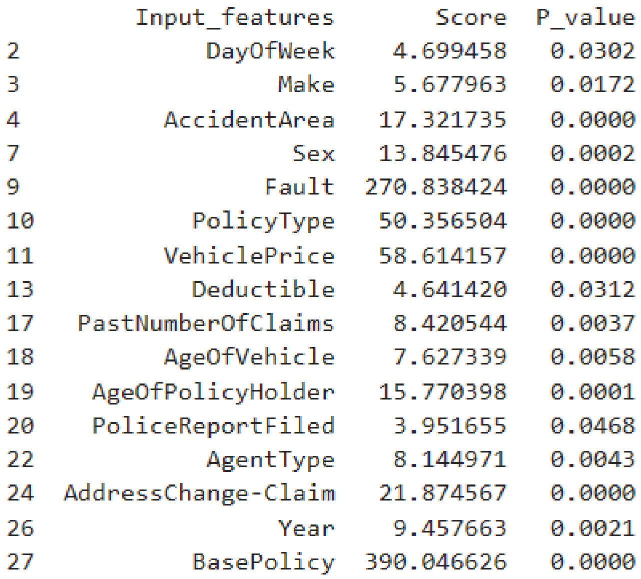
The feature selection method used in this study is the ANOVA test, A filter method in feature selection. The filler method is used because it provides good efficiency and model independence.

The 'SelectKbest' selects the 'k' number of features that are mentioned from the given dataset. In this, we are selecting the best features based on p-values. The processed data is split into dependent and independent variables. The 'FraudFound' feature is taken as the dependent variable and the remaining columns 'Month', 'WeekOfMonth', 'DayOfWeek', 'Make', 'AccidentArea', 'DayOfWeekClaimed', 'WeekOfMonthClaimed', 'Sex', 'MaritalStatus', 'Fault', 'PolicyType', 'VehiclePrice', 'RepNumber', 'Deductible', 'DriverRating', 'Days:Policy-Accident', 'Days:Policy-Claim', 'PastNumberOfClaims', 'AgeOfVehicle', 'AgeOfPolicyHolder', 'PoliceReportFiled', 'WitnessPresent', 'AgentType', 'NumberOfSuppliments', 'AddressChange-Claim', 'NumberOfCars', 'Year', 'BasePolicy' are considered as independent variables the score for the feature is based on how the independent variable is related to dependent variable.

When selecting the most important features for our predictive model, we use F-scores and p-values calculated through the fs object. The F-score, or F-statistic, is used to measure the difference in means between two or more groups. A higher F-score indicates significant differences in means between groups, which can suggest that the corresponding feature is relevant for predicting the target variable. Based on the F-scores, we have identified 17 features out of 29: 'DayOfWeek', 'Make', 'AccidentArea', 'Sex', 'Fault', 'PolicyType', 'VehiclePrice', 'Deductible', 'PastNumberOfClaims', 'AgeOfVehicle', 'AgeOfPolicyHolder', 'PoliceReportFiled', 'AgentType', 'AddressChange-Claim', 'Year', 'BasePolicy' and 'FraudFound'

P-values are calculated based on hypothesis testing. If the null hypothesis is true, there is no relationship between the feature and the target variable. Our alpha value is set to 0.05, so if the P-value is less than 0.05, we can reject the null hypothesis. A low p-value suggests that the corresponding feature is likely to be relevant for predicting the target variable. Figure 4.27 represents features selected based on the p-value and score.

F-scores and p-values are crucial in feature selection since they help identify statistically significant features that can contribute valuable information to the predictive model. Features with higher F-scores and lower p-values are more likely to be important in predicting the target variable.



```
          Input_features       Score  P_value
2              DayOfWeek    4.699458   0.0302
3                   Make    5.677963   0.0172
4           AccidentArea   17.321735   0.0000
7                    Sex   13.845476   0.0002
9                  Fault  270.838424   0.0000
10            PolicyType   50.356504   0.0000
11          VehiclePrice   58.614157   0.0000
13            Deductible    4.641420   0.0312
17    PastNumberOfClaims    8.420544   0.0037
18          AgeOfVehicle    7.627339   0.0058
19      AgeOfPolicyHolder   15.770398   0.0001
20      PoliceReportFiled    3.951655   0.0468
22              AgentType    8.144971   0.0043
24   AddressChange-Claim   21.874567   0.0000
26                   Year    9.457663   0.0021
27             BasePolicy  390.046626   0.0000
```

Figure 4.27 Selected Feature

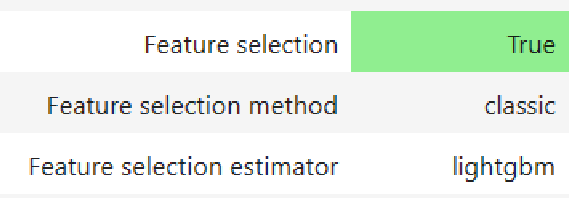
Based on the p-values, 'DayOfWeek', 'Make', 'AccidentArea', 'Sex', 'Fault', 'PolicyType', 'VehiclePrice', 'Deductible', 'PastNumberOfClaims', 'AgeOfVehicle', 'AgeOfPolicyHolder', 'PoliceReportFiled', 'AgentType', 'AddressChange-Claim', 'Year', 'BasePolicy' are the features selected for the ML implementation. In Table 4.1, we can see the final features we selected.

Table 4.3 Selected Features

Feature Selected	Unique Data’s
DayOfWeek	[6, 0, 2, 1, 5, 3, 4]
Make	[6, 17, 5, 9, 2, 13, 0, 3, 11, 7, 12, 18, 15, 16, 14, 1, 10, 4, 8]
AccidentArea	[1, 0]
Sex	[0, 1]
Fault	[0, 1]
PolicyType	[5, 4, 2, 6, 0, 1, 7, 8, 3]
VehiclePrice	[5, 0, 1, 4, 2, 3]
Deductible	[300, 400, 500, 700]

PastNumberOfClaims	[3, 0, 1, 2]
AgeOfVehicle	[1, 4, 5, 6, 3, 7, 2, 0]
AgeOfPolicyHolder	[3, 4, 6, 7, 2, 5, 0, 8, 1]
PoliceReportFiled	[0, 1]
AgentType	[0, 1]
AddressChange-Claim	[0, 3, 2, 1, 4]
Year	[0, 1, 2]
BasePolicy	[2, 1, 0]
FraudFound	[0, 1]

PyCaret setup has its own feature selection method when given true for feature selection, PyCaret utilises the classic method to select the feature selection. The feature estimator is set to LightGBM as default, if needed we can alter it manually. PyCaret feature selection can be seen in Figure 4.28.



Feature selection	True
Feature selection method	classic
Feature selection estimator	lightgbm

Figure 4.28 PyCaret Feature Selection

4.5.2 Class Imbalance Technique

To ensure accurate ML comparison, the data is balanced using a technique called Synthetic Minority Over-sampling Technique (SMOTE). Figure 4.29 provides a visual representation of the balancing process applied to the data. The original data set contains a total of 15420 claim instances, with 94% being non-fraud and only 6% being fraud, resulting in an imbalanced data set. This imbalance can impact the performance and accuracy of the ML model.

SMOTE is a resampling technique used to address overfitting issues. Overfitting is problematic because machine learning algorithms are evaluated on training data which may not necessarily reflect their performance on unseen data. By balancing the data set using SMOTE, we can ensure that the ML model is trained on a representative sample of data that accurately reflects the entire data set. In Table 4.4, we can see the difference in the count of data before and after SMOTE.

Table 4.4 SMOTE and ROS data distribution

Fraud Found	0	1
Before SMOTE	14497	923
After SMOTE	14497	14497

Just as the data split is done for the feature selection, the data is split into dependent and independent variables. Each variable is then fit back after transformation using the resampling techniques. The results are shown in Figure 33. The same data splitting has been carried out for Random Under Sampling (RUS) and Random Over Sampling (ROS). This data is then again implemented into the ML model for verification.

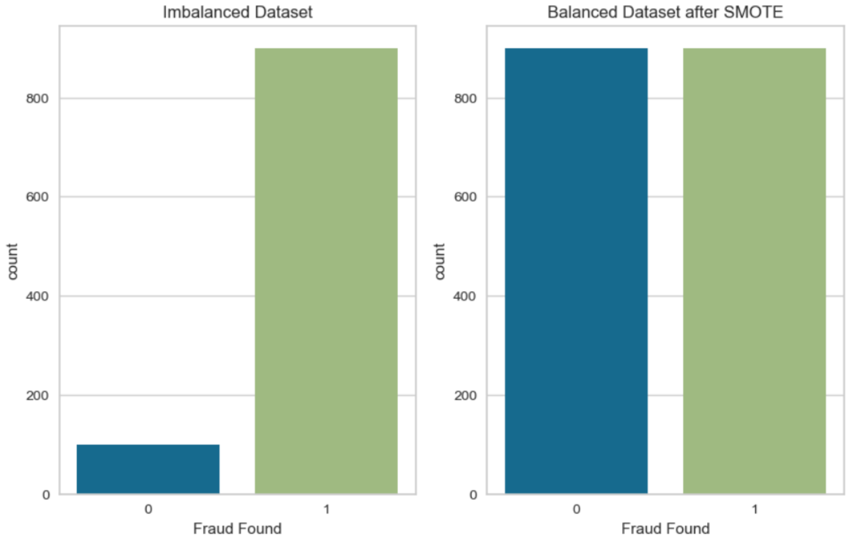


Figure 4.29 Data Balance

We also used Random Over and Under Sampling methods to understand the impact of balancing techniques on imbalanced data sets, leading to the best ML model. Table 4.5 shows the data distribution of RUS and table 4.4 has the data for ROS. These balancing techniques equalise the data by creating duplicate values to equalise both variables.

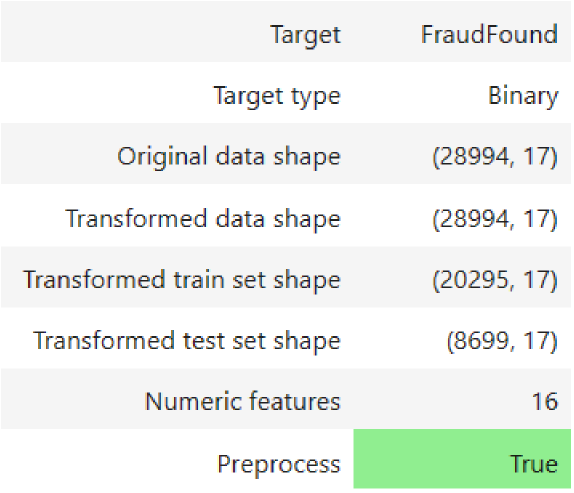
Table 4.5 Random Under sample data distribution

Fraud Found	0	1
Before RUS	14497	923
After RUS	923	923

The data has been carefully selected and balanced and is now prepared to be used for machine learning implementation. The RUS and ROS are used for comparison of different resampling techniques.

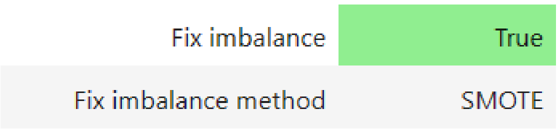
In Figure 4.30, we can see the data that was selected after going through the resampling technique. This technique helps to equalize majority and minority data by increasing the sample size. As a result of the resampling technique, the number of data increased from 15420 to 28994.

PyCaret offers an option to automatically balance the data using the "Fix_imbalance = True" parameter. By default, PyCaret uses SMOTE as the balancing technique, which is shown in Figure 4.31. However, we can change the imbalance method to any other method we need, as shown in Figure 4.32.



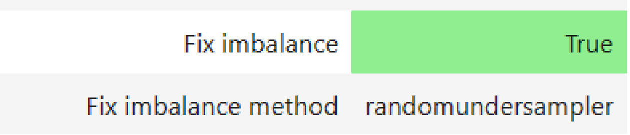
Target	FraudFound
Target type	Binary
Original data shape	(28994, 17)
Transformed data shape	(28994, 17)
Transformed train set shape	(20295, 17)
Transformed test set shape	(8699, 17)
Numeric features	16
Preprocess	True

Figure 4.30 Feature selected SMOTE model



Fix imbalance	True
Fix imbalance method	SMOTE

Figure 4.31 PyCaret Feature selection SMOTE



Fix imbalance	True
Fix imbalance method	randomundersampler

Figure 4.32 PyCaret Feature selection RUS

In the context of imbalanced classification challenges, Random Over-Sampling (ROS), Random Under-Sampling (RUS), and Synthetic Minority Over-sampling Technique (SMOTE) have become pivotal strategies for addressing skewed class distributions. ROS duplicates instances of the minority class to tackle the issue, which is simple to implement but may lead to overfitting. On the other hand, RUS fights imbalance by randomly removing instances from the majority class, optimizing computational efficiency but possibly discarding important information. SMOTE, on the other hand, employs a more complex approach by generating synthetic minority class instances through interpolation. This method addresses overfitting concerns and facilitates the capture of complex decision boundaries. While ROS simplifies the task with straightforward replication, SMOTE's sophistication lies in its ability to diversify the minority class.

We chose ROS and RUS for our study of class imbalance because unlike SMOTE, which equalizes the data by increasing and decreasing it randomly, these methods resample the data without any specific modifications. This allows us to use the original data as is when applying resampling techniques.

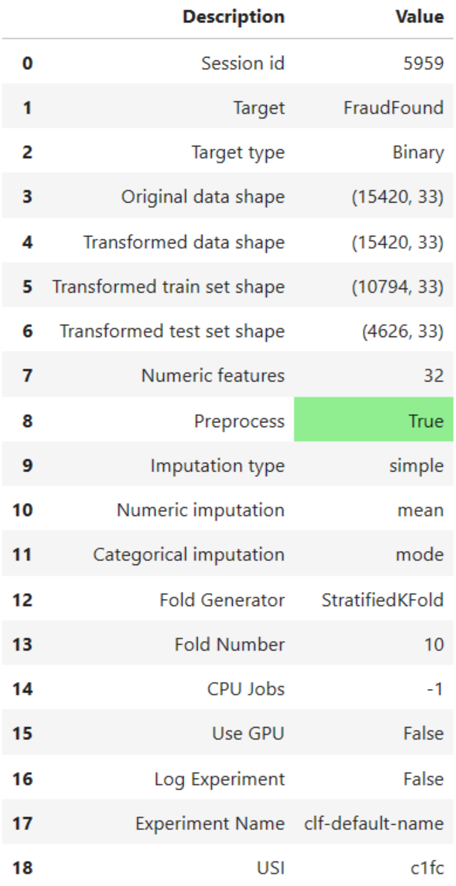
4.6 Machine Learning Implementation

To develop the best model for fraud prediction, we will now implement the previously processed data using the PyCaret module. The reason we utilize PyCaret for the comparison is it provides a higher-level interface that simplifies and accelerates the machine learning workflow

4.6.1 Train-Test Split

When working with machine learning, the first and most important step is data splitting. Unlike splitting the data for feature and resampling techniques, where the dataset is separated into dependent and independent variables, here we split the data into test and train models. Generally, the data is split into either 80% and 20% or 70% and 30%. The 80% or 70% data split is used as the training set, which is used to train the ML model. The remaining 30% or 20% of data is used to test the model's performance and accuracy.

However, in our study, we are not splitting the data. Instead, we are feeding the entire dataset to PyCaret. PyCaret is an ML model that provides all the necessary steps we need to implement the data. To use PyCaret, we first need to set up the environment. This environment setup includes all the necessary details such as data- dataset we use, target – target variable of the data, categorical variable- categorical variables in the data, and 72 other parameters that define the environment we need.



	Description	Value
0	Session id	5959
1	Target	FraudFound
2	Target type	Binary
3	Original data shape	(15420, 33)
4	Transformed data shape	(15420, 33)
5	Transformed train set shape	(10794, 33)
6	Transformed test set shape	(4626, 33)
7	Numeric features	32
8	Preprocess	True
9	Imputation type	simple
10	Numeric imputation	mean
11	Categorical imputation	mode
12	Fold Generator	StratifiedKFold
13	Fold Number	10
14	CPU Jobs	-1
15	Use GPU	False
16	Log Experiment	False
17	Experiment Name	clf-default-name
18	USI	c1fc

Figure 4.33 PyCaret setup

Figure 4.33 represents the PyCaret setup environment, the session ID is auto-generated or can be created manually. The setup is explained briefly.

- Session ID: It is an identifier assigned to our PyCaret session.
- Target: The target variable for our model to predict is 'FraudFound'.
- Target type: It specifies the type of the target variable, and in this case, it's 'Binary', which means it's a binary classification problem with two classes: 0 or 1.
- Original data shape: It shows the shape of the original dataset before any transformations, which has 15,420 rows and 33 columns.
- Transformed data shape: It shows the shape of the dataset after any preprocessing or transformations. In this case, it remains the same, with 15,420 rows and 33 columns.
- Transformed train set shape: It shows the shape of the training set after the data has been split into training and testing sets, which have 10,794 rows and 33 columns.
- Transformed test set shape: It shows the shape of the testing set, which has 4,626 rows and 33 columns.
- Numeric features: It indicates the number of numeric features in the dataset, which is 32.

- Preprocess: It indicates whether preprocessing steps have been applied, and in this case, it's set to True, meaning that PyCaret has performed preprocessing.
- Imputation type: It specifies the type of imputation used for missing values, which is 'simple', indicating a basic imputation strategy. Since our dataset has no missing values.
- Numeric imputation: It shows the method used for imputing missing values in numeric features, which is 'mean', meaning that missing numeric values are filled with the mean of the respective column.
- Categorical imputation: It shows the method used for imputing missing values in categorical features, which is 'mode', meaning that missing categorical values are filled with the mode of the respective column.
- Fold Generator: It shows the method used for generating folds in cross-validation, which is set to 'StratifiedKFold', a type of cross-validation that maintains the distribution of the target variable in each fold.
- Fold Number: It shows the number of folds used in cross-validation, which is set to 10.
- CPU Jobs: It shows the number of Central Processing Unit (CPU) jobs to use during parallel processing, which is set to -1, meaning that it will use all available CPU cores.
- Use GPU: It indicates whether Graphical Processing Unit (GPU) acceleration is used, and in this case, it's set to False.
- Log Experiment: It specifies whether to log the experiment and, in this case, it's set to False.
- Experiment Name: It's the name assigned to the experiment, and it's set to 'clf-default-name'.
- USI (Unique Session ID): It's a unique identifier for the current PyCaret session.

These parameters provide an overview of the setup and configuration of our PyCaret experiment. Each parameter contributes to defining how the machine learning experiment is conducted, including data preprocessing, feature engineering, model training, and evaluation.

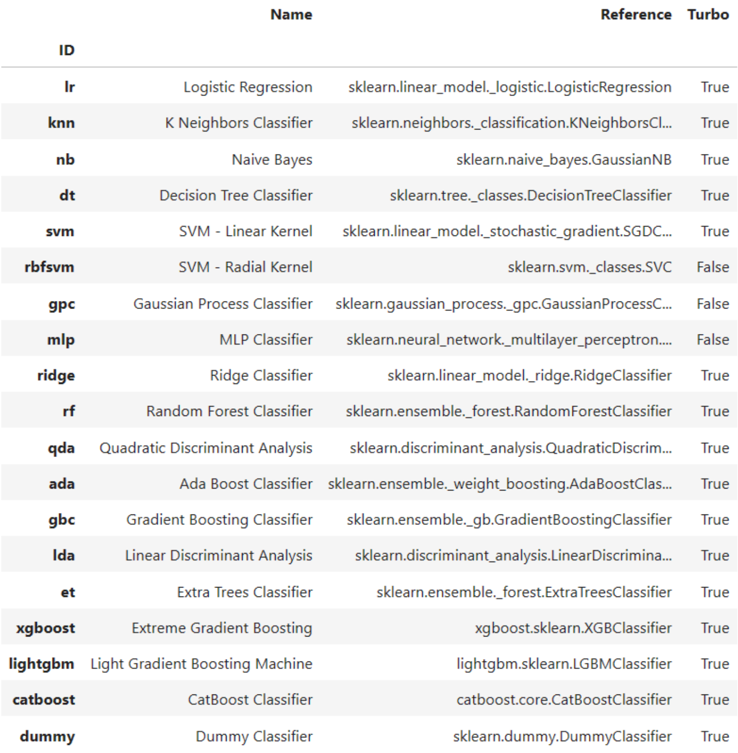
The PyCaret itself has a lot of parameters that don’t need any programming. Such as the data split. Instead of splitting the data set manually, PyCaret split the dataset into test and train sets automatically in a 70:30 ratio.

4.6.2 Implementation

The PyCaret has features that read the type of dataset used such as ordinal, numerical or categorical. It identifies the type of data set used and separates them or we can manually mention the type of data the features we are using. The train and test set are subjected to the available ML models. The data is split into 10,794 rows of data for the train and 4,626 rows of data for the test. The test and train data will be implemented in ML and the result will be cross-validated automatically. The test data is cross-validated using 10-cross validation, we can manually change the cross-validation to the number of folds we need.

PyCaret can handle the missing values itself or we can provide any parameter to follow. PyCaret follows a predetermined set of parameters to function, which can be altered when setting the environment. The ML model within PyCaret can be increased. Our Processed dataset is subjected to 16 ML models. These models are known as classifier models as they are used for classification purposes. From Figure 4.34, we can see the number of available models in PyCaret for classification functions. The turbo shown in the figure represents the availability of the said model.

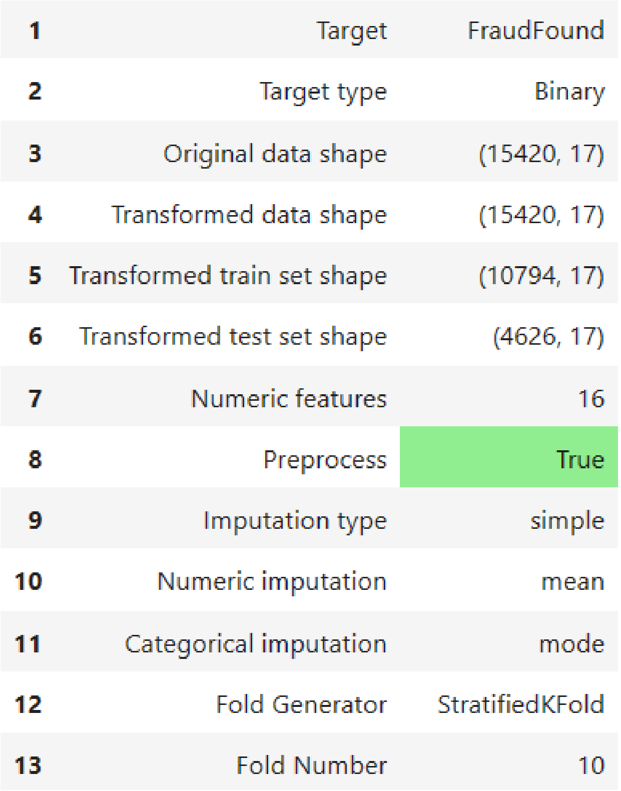
For our study we utilize 16 ML models for comparison in PyCaret those are LR, KNN, LDA, QDA, NB, DT, GBC, ET, RF, LightGB, Ada Boost, CatBoost, RC, SVM, XGB and DC. As we can see several models used are ensemble models as previously mentioned.



	Name	Reference	Turbo
ID			
lr	Logistic Regression	sklearn.linear_model._logistic.LogisticRegression	True
knn	K Neighbors Classifier	sklearn.neighbors._classification.KNeighborsCl...	True
nb	Naive Bayes	sklearn.naive_bayes.GaussianNB	True
dt	Decision Tree Classifier	sklearn.tree._classes.DecisionTreeClassifier	True
svm	SVM - Linear Kernel	sklearn.linear_model._stochastic_gradient.SGDC...	True
rbfsvm	SVM - Radial Kernel	sklearn.svm._classes.SVC	False
gpc	Gaussian Process Classifier	sklearn.gaussian_process._gpc.GaussianProcessC...	False
mlp	MLP Classifier	sklearn.neural_network._multilayer_perceptron....	False
ridge	Ridge Classifier	sklearn.linear_model._ridge.RidgeClassifier	True
rf	Random Forest Classifier	sklearn.ensemble._forest.RandomForestClassifier	True
qda	Quadratic Discriminant Analysis	sklearn.discriminant_analysis.QuadraticDiscrim...	True
ada	Ada Boost Classifier	sklearn.ensemble._weight_boosting.AdaBoostClas...	True
gbc	Gradient Boosting Classifier	sklearn.ensemble._gb.GradientBoostingClassifier	True
lda	Linear Discriminant Analysis	sklearn.discriminant_analysis.LinearDiscrimina...	True
et	Extra Trees Classifier	sklearn.ensemble._forest.ExtraTreesClassifier	True
xgboost	Extreme Gradient Boosting	xgboost.sklearn.XGBClassifier	True
lightgbm	Light Gradient Boosting Machine	lightgbm.sklearn.LGBMClassifier	True
catboost	CatBoost Classifier	catboost.core.CatBoostClassifier	True
dummy	Dummy Classifier	sklearn.dummy.DummyClassifier	True

Figure 4.34 Machine Learning models

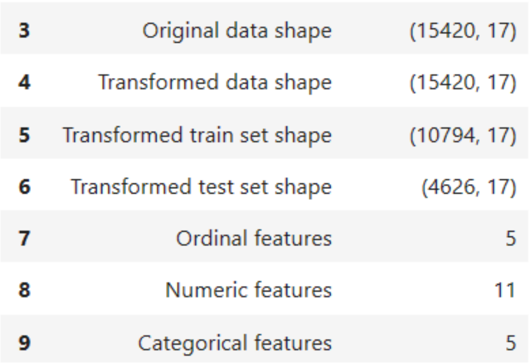
For our comparative study, we separated the process into three categories: feeding raw encoded data without feature selection, with feature selection, and balanced data. After comparing the results, we selected the best model as our final model. The first implementation of the process is the raw data with the encoded value. Figure 4.33 explains the raw data implementation of the entire data without any feature selection or resampling techniques being used. The train-test split has been carried out and since it's been encoded all the values are in numerical form. The next implementation will be the feature-selected data.



1	Target	FraudFound
2	Target type	Binary
3	Original data shape	(15420, 17)
4	Transformed data shape	(15420, 17)
5	Transformed train set shape	(10794, 17)
6	Transformed test set shape	(4626, 17)
7	Numeric features	16
8	Preprocess	True
9	Imputation type	simple
10	Numeric imputation	mean
11	Categorical imputation	mode
12	Fold Generator	StratifiedKFold
13	Fold Number	10

Figure 4.35 Feature selected implementation

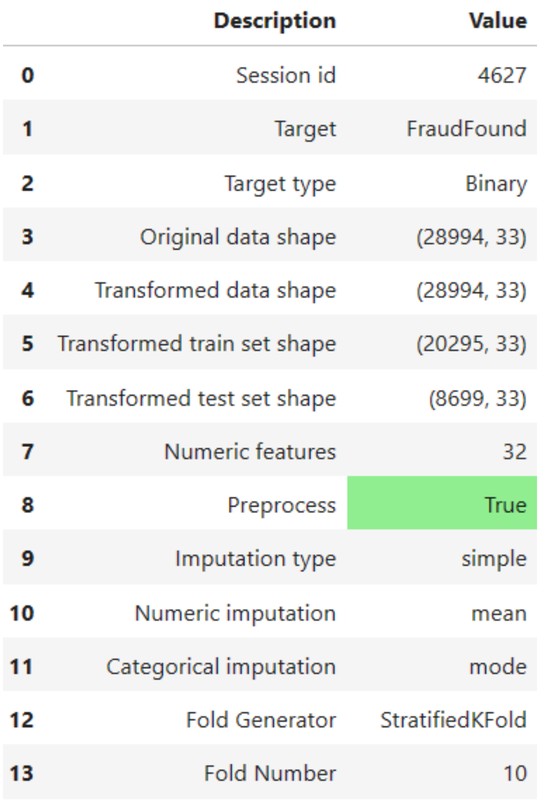
In Figure 4.35, we see how PyCaret has implemented feature data. The data was split into test and train sets. The test data is being validated using 10-cross validation. All the feature data is in numerical form. The original data consists of 15420 data points, with 17 features including the target variable. PyCaret has automatically recognized all the features as numerical. However, if needed, we can manually specify some features as categorical. There are different types of categorical features, such as Binary Categorical Variables (e.g., "AccidentArea", "Sex", "Fault", "PoliceReportFiled", “AgentType” and "FraudFound"), Ordinal Categorical Variables (e.g., "PastNumberOfClaims", "AgeOfVehicle" and "AgeOfPolicyHolder" as they represent the data in an orderly manner, Reference in table 4) and Nominal Categorical Variables (e.g., "Make", "PolicyType", "VehiclePrice", "Deductible", "AddressChange-Claim", "Year" and "BasePolicy" as they have values representing numerical format).



3	Original data shape	(15420, 17)
4	Transformed data shape	(15420, 17)
5	Transformed train set shape	(10794, 17)
6	Transformed test set shape	(4626, 17)
7	Ordinal features	5
8	Numeric features	11
9	Categorical features	5

Figure 4.36 Feature classification

The selected data features are classified as categorical variables and implemented in PyCaret. Figure 4.36 shows PyCaret correctly classifying the variables.



	Description	Value
0	Session id	4627
1	Target	FraudFound
2	Target type	Binary
3	Original data shape	(28994, 33)
4	Transformed data shape	(28994, 33)
5	Transformed train set shape	(20295, 33)
6	Transformed test set shape	(8699, 33)
7	Numeric features	32
8	Preprocess	True
9	Imputation type	simple
10	Numeric imputation	mean
11	Categorical imputation	mode
12	Fold Generator	StratifiedKFold
13	Fold Number	10

Figure 4.37 SMOTE Raw Data

Figure 4.37 provides a preview of the encoded data subject to resampling methods. The data has been split with a 70:30 ratio, due to the data resampling method the number of data has risen from 15420 to 28994 with all 33 features.

PyCaret setup is a comprehensive tool that guides us in implementing data into ML algorithms. We can perform various operations, such as handling categorical variables, fixing imbalance, feature selection, cross-validation, and more, as per our needs. All the necessary steps for machine learning are included in the PyCaret environment. To obtain better results, it is essential to mention these steps in detail.

In our research, we did not use the predefined algorithm for preprocessing in PyCaret. Instead, we handled feature selection and resampling separately. This is because the output gained through PyCaret's predefined algorithm is comparatively less helpful for our research.

4.7 Summary

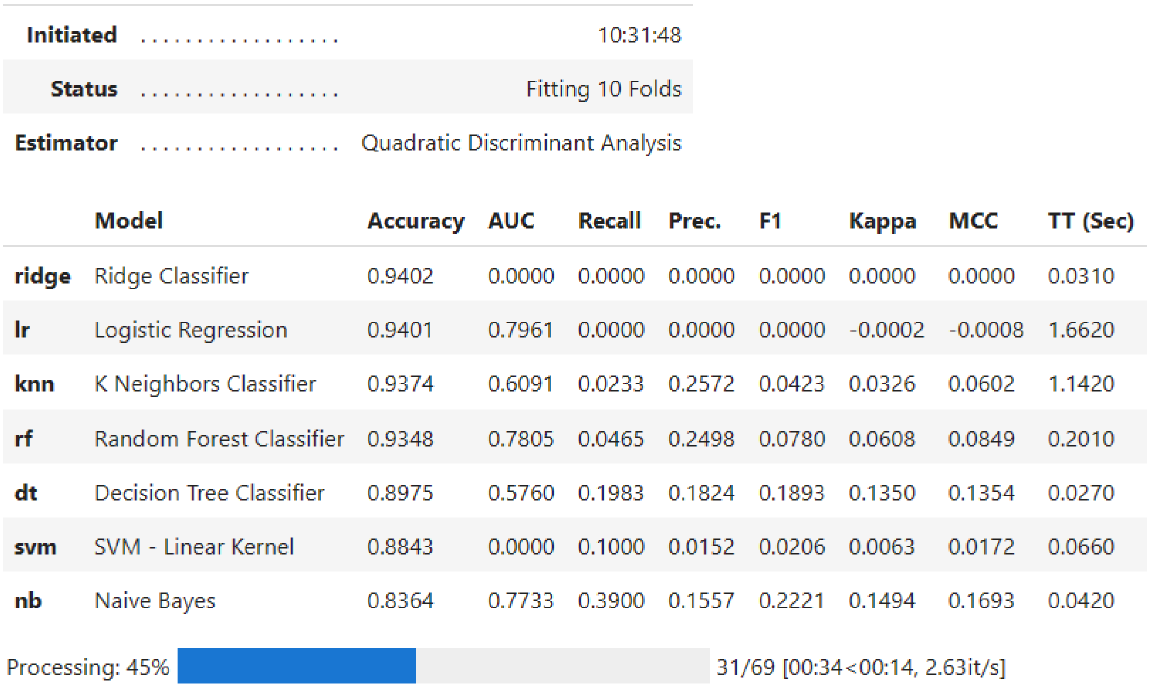
In this chapter, we provide a detailed explanation of the insurance data used for this study in the data description section. We extensively studied the dataset in the EDA section and found that male drivers aged between 31 and 35, who own sedan cars from Pontiac and Toyota, with a price range between 20,000 to 29,000 and based in urban areas tend to commit insurance fraud by taking policies covering 'All Perils' and 'Collision'. We have transformed the data from categorical format to numerical format using label encoding. Based on the p-value from the ANOVA test, we have selected the necessary features for further study. To balance the data, we have used resampling techniques such as SMOTE, RUS, and ROS. Finally, we have implemented the processed data in PyCaret, which carried out the data splitting itself and subjected the data into 16 ML models namely LR, KNN, LDA, QDA, NB, DT, GBC, ET, RF, LightGB, Ada Boost, CatBoost, RC, SVM, XGB and DC. We have explained the Pre-Processing techniques within PyCaret and the PyCaret environment setup. The next chapter will cover the results and evaluation of our study.

CHAPTER 5:
RESULTS AND EVALUATION

In this chapter, we will explain the process of applying the processed data to PyCaret and the results obtained from it. After setting up and comparing models, we will explain the subsequent steps, followed by the explanation of the results obtained using all the methods for our study in section 5.1. We will discuss and explain the evaluation results of the model in section 5.2, and the limitations will be discussed in section 5.3.

5.1 Introduction

After completing the setup for PyCaret, we can proceed to model comparison. In this stage, the pre-processed data will be utilized and compared against the machine learning models. Once the comparison process is initiated in PyCaret, the data is trained and evaluated using a 10-fold cross-validation technique. This means that the model is trained and evaluated ten times, each time using a different subset as the test set and the remaining data for training. This approach helps in assessing the model's performance across different data subsets. The number of folds can be manually changed to any desired number. Figure 5.1 depicts the process of comparing ML models. The estimator displays the name of the model being compared. Similar to k-fold, the ML models can also be selected as per the user's requirements. For our study, we have selected all the available models.



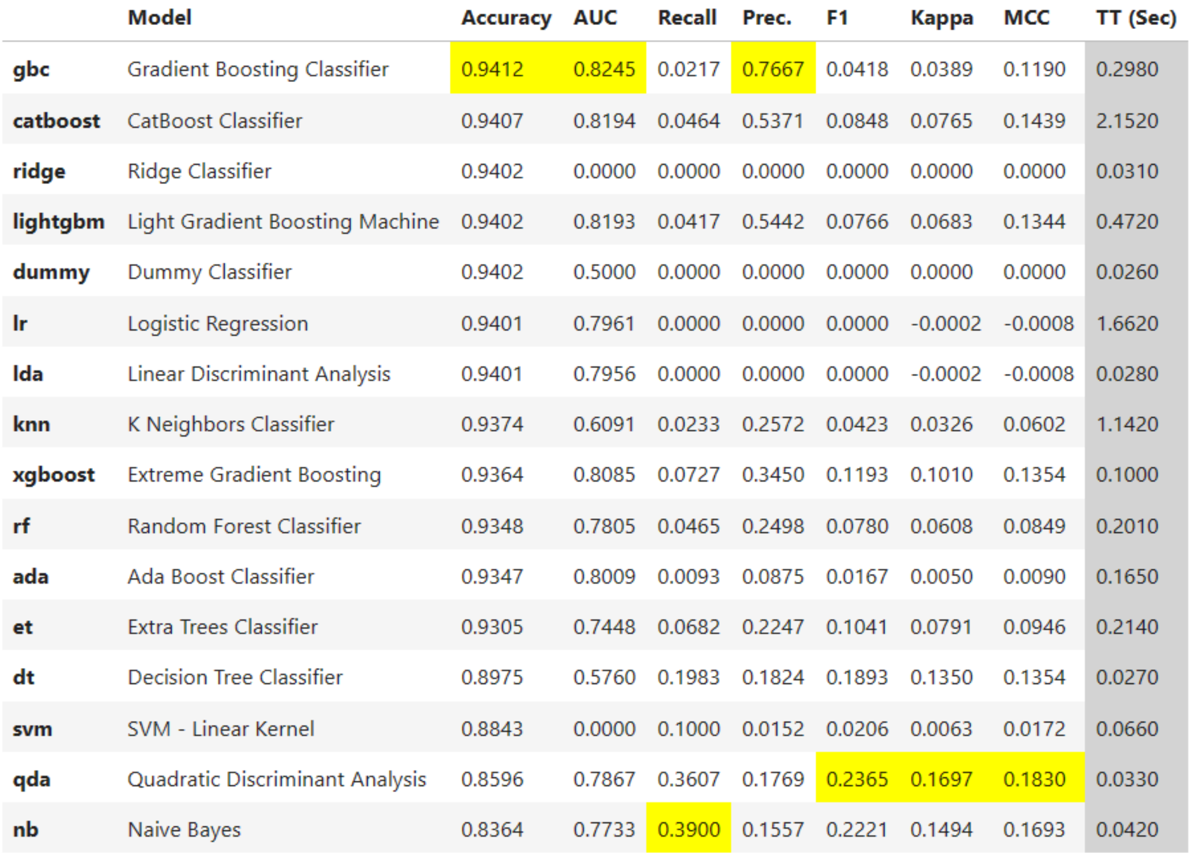
Initiated	10:31:48
Status	Fitting 10 Folds
Estimator	Quadratic Discriminant Analysis

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
ridge	Ridge Classifier	0.9402	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0310
lr	Logistic Regression	0.9401	0.7961	0.0000	0.0000	0.0000	-0.0002	-0.0008	1.6620
knn	K Neighbors Classifier	0.9374	0.6091	0.0233	0.2572	0.0423	0.0326	0.0602	1.1420
rf	Random Forest Classifier	0.9348	0.7805	0.0465	0.2498	0.0780	0.0608	0.0849	0.2010
dt	Decision Tree Classifier	0.8975	0.5760	0.1983	0.1824	0.1893	0.1350	0.1354	0.0270
svm	SVM - Linear Kernel	0.8843	0.0000	0.1000	0.0152	0.0206	0.0063	0.0172	0.0660
nb	Naive Bayes	0.8364	0.7733	0.3900	0.1557	0.2221	0.1494	0.1693	0.0420

Processing: 45% 31/69 [00:34<00:14, 2.63it/s]

Figure 5.1 Comparing ML models

The results after comparing ML models can be viewed in Figure 43. The PyCaret itself select the best model based on the evaluation criteria as a whole and also selects the individual model that performs well on each evaluation. In Figure 5.2, the Gradient Boosting classifier is resulted as the best model based on accuracy, AUC and Precision and is highlighted and based on recall the Naïve Bayes performs well and Quadratic Discriminant analysis performs well in F1-score, Kappa and MCC. The TT (sec) in the figure represents the time taken for the particular model. It shows that Gradient Boosting (GB) took 0.2980 secs to complete 10-fold validation. The best model is arranged based on the accuracy score.



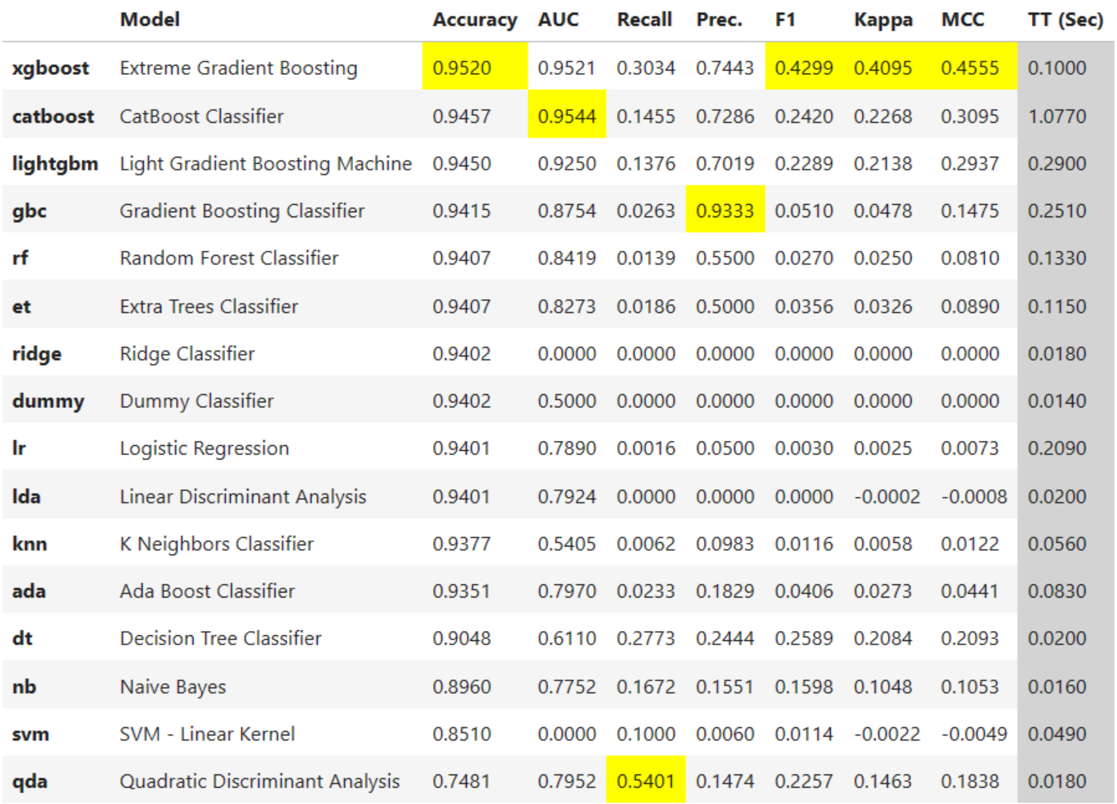
	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
gbc	Gradient Boosting Classifier	0.9412	0.8245	0.0217	0.7667	0.0418	0.0389	0.1190	0.2980
catboost	CatBoost Classifier	0.9407	0.8194	0.0464	0.5371	0.0848	0.0765	0.1439	2.1520
ridge	Ridge Classifier	0.9402	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0310
lightgbm	Light Gradient Boosting Machine	0.9402	0.8193	0.0417	0.5442	0.0766	0.0683	0.1344	0.4720
dummy	Dummy Classifier	0.9402	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0260
lr	Logistic Regression	0.9401	0.7961	0.0000	0.0000	0.0000	-0.0002	-0.0008	1.6620
lda	Linear Discriminant Analysis	0.9401	0.7956	0.0000	0.0000	0.0000	-0.0002	-0.0008	0.0280
knn	K Neighbors Classifier	0.9374	0.6091	0.0233	0.2572	0.0423	0.0326	0.0602	1.1420
xgboost	Extreme Gradient Boosting	0.9364	0.8085	0.0727	0.3450	0.1193	0.1010	0.1354	0.1000
rf	Random Forest Classifier	0.9348	0.7805	0.0465	0.2498	0.0780	0.0608	0.0849	0.2010
ada	Ada Boost Classifier	0.9347	0.8009	0.0093	0.0875	0.0167	0.0050	0.0090	0.1650
et	Extra Trees Classifier	0.9305	0.7448	0.0682	0.2247	0.1041	0.0791	0.0946	0.2140
dt	Decision Tree Classifier	0.8975	0.5760	0.1983	0.1824	0.1893	0.1350	0.1354	0.0270
svm	SVM - Linear Kernel	0.8843	0.0000	0.1000	0.0152	0.0206	0.0063	0.0172	0.0660
qda	Quadratic Discriminant Analysis	0.8596	0.7867	0.3607	0.1769	0.2365	0.1697	0.1830	0.0330
nb	Naive Bayes	0.8364	0.7733	0.3900	0.1557	0.2221	0.1494	0.1693	0.0420

Figure 5.2 Comparison result

5.2 Model Output

The output is organized clearly and logically, starting with the use of original data and then moving on to the utilization of data resampling methods.

Our first method uses the original data encoded with the Label encoding technique. The encoded data is then implemented into PyCaret without any feature selection or resampling. In Figure 5.3, the output for the first model can be seen. Based on this output, we can understand that Extreme Gradient Boosting (XGBoost) is the best model for the data with a 95% accuracy rate. The entire process took only 0.1 seconds to complete the 10-fold validation. The XGBoost model also performed best for F1-score, Kappa, and MCC metrics. CatBoost followed XGBoost as the second-best model with the best AUC curve compared to XGBoost.



	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
xgboost	Extreme Gradient Boosting	0.9520	0.9521	0.3034	0.7443	0.4299	0.4095	0.4555	0.1000
catboost	CatBoost Classifier	0.9457	0.9544	0.1455	0.7286	0.2420	0.2268	0.3095	1.0770
lightgbm	Light Gradient Boosting Machine	0.9450	0.9250	0.1376	0.7019	0.2289	0.2138	0.2937	0.2900
gbc	Gradient Boosting Classifier	0.9415	0.8754	0.0263	0.9333	0.0510	0.0478	0.1475	0.2510
rf	Random Forest Classifier	0.9407	0.8419	0.0139	0.5500	0.0270	0.0250	0.0810	0.1330
et	Extra Trees Classifier	0.9407	0.8273	0.0186	0.5000	0.0356	0.0326	0.0890	0.1150
ridge	Ridge Classifier	0.9402	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0180
dummy	Dummy Classifier	0.9402	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0140
lr	Logistic Regression	0.9401	0.7890	0.0016	0.0500	0.0030	0.0025	0.0073	0.2090
lda	Linear Discriminant Analysis	0.9401	0.7924	0.0000	0.0000	0.0000	-0.0002	-0.0008	0.0200
knn	K Neighbors Classifier	0.9377	0.5405	0.0062	0.0983	0.0116	0.0058	0.0122	0.0560
ada	Ada Boost Classifier	0.9351	0.7970	0.0233	0.1829	0.0406	0.0273	0.0441	0.0830
dt	Decision Tree Classifier	0.9048	0.6110	0.2773	0.2444	0.2589	0.2084	0.2093	0.0200
nb	Naive Bayes	0.8960	0.7752	0.1672	0.1551	0.1598	0.1048	0.1053	0.0160
svm	SVM - Linear Kernel	0.8510	0.0000	0.1000	0.0060	0.0114	-0.0022	-0.0049	0.0490
qda	Quadratic Discriminant Analysis	0.7481	0.7952	0.5401	0.1474	0.2257	0.1463	0.1838	0.0180

Figure 5.3 Output - Encoded Raw Data

The model’s precision is higher when using a gradient boosting classifier having a 93% precision rate. The confusion matrix is shown in Figure 5.4. The matrix is created based on the test data set separated by PyCaret. The test consists of 4626 data of which the XGBoost found 4304 data as fraud. The AUC-ROC curve is presented in Figure 5.5. Based on the curve we can derive that the model predicted by PyCaret is good, but the AUC curve for CatBoost is higher compared to XGBoost.

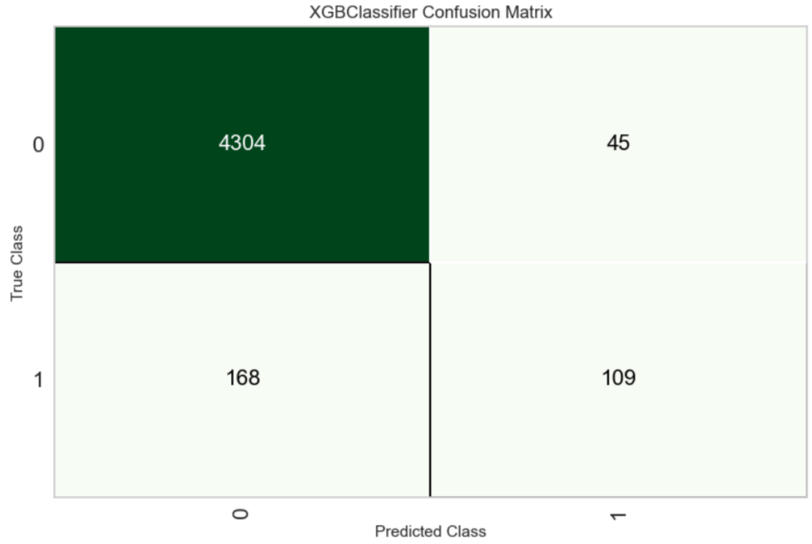


Figure 5.4 XGBoost confusion matrix

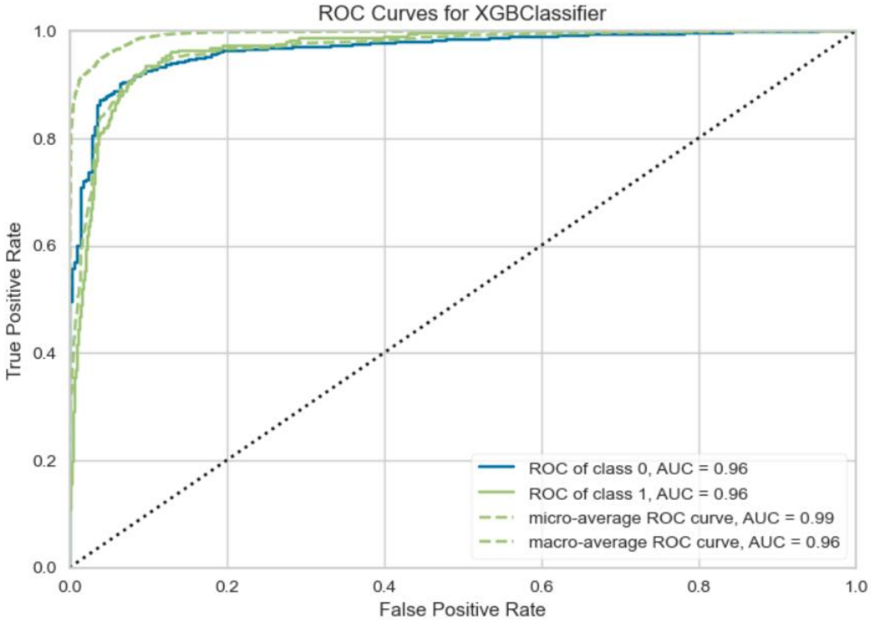
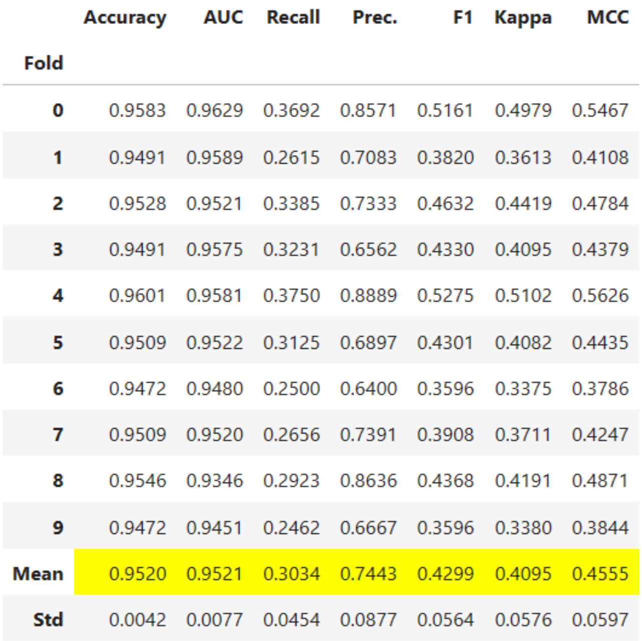


Figure 5.5 XGBoost AUC curve

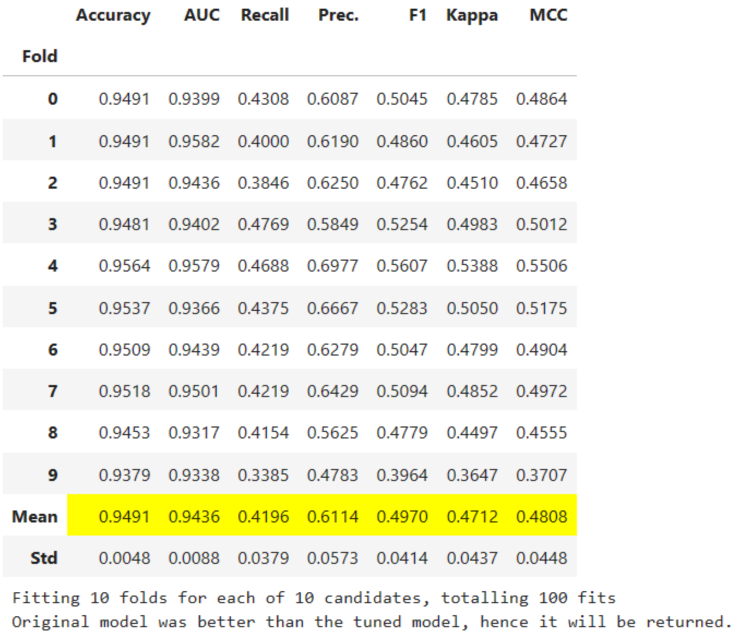
The evaluation matrix results are based on a 10-fold cross-validation. The final output is the mean of the validation, which can be seen in Figure 5.6. The accuracy of the XGBoost model is 0.9520, which is the mean of 10 validations.



	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
Fold							
0	0.9583	0.9629	0.3692	0.8571	0.5161	0.4979	0.5467
1	0.9491	0.9589	0.2615	0.7083	0.3820	0.3613	0.4108
2	0.9528	0.9521	0.3385	0.7333	0.4632	0.4419	0.4784
3	0.9491	0.9575	0.3231	0.6562	0.4330	0.4095	0.4379
4	0.9601	0.9581	0.3750	0.8889	0.5275	0.5102	0.5626
5	0.9509	0.9522	0.3125	0.6897	0.4301	0.4082	0.4435
6	0.9472	0.9480	0.2500	0.6400	0.3596	0.3375	0.3786
7	0.9509	0.9520	0.2656	0.7391	0.3908	0.3711	0.4247
8	0.9546	0.9346	0.2923	0.8636	0.4368	0.4191	0.4871
9	0.9472	0.9451	0.2462	0.6667	0.3596	0.3380	0.3844
Mean	0.9520	0.9521	0.3034	0.7443	0.4299	0.4095	0.4555
Std	0.0042	0.0077	0.0454	0.0877	0.0564	0.0576	0.0597

Figure 5.6 Cross-validation

When tuning the proposed model, PyCaret validates the cross-validation 10x10 for XGBoost and provides the mean of the results. Figure 5.7 displays the hyper-tuning results for XGBoost.

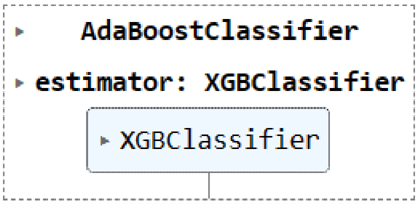


	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
Fold							
0	0.9491	0.9399	0.4308	0.6087	0.5045	0.4785	0.4864
1	0.9491	0.9582	0.4000	0.6190	0.4860	0.4605	0.4727
2	0.9491	0.9436	0.3846	0.6250	0.4762	0.4510	0.4658
3	0.9481	0.9402	0.4769	0.5849	0.5254	0.4983	0.5012
4	0.9564	0.9579	0.4688	0.6977	0.5607	0.5388	0.5506
5	0.9537	0.9366	0.4375	0.6667	0.5283	0.5050	0.5175
6	0.9509	0.9439	0.4219	0.6279	0.5047	0.4799	0.4904
7	0.9518	0.9501	0.4219	0.6429	0.5094	0.4852	0.4972
8	0.9453	0.9317	0.4154	0.5625	0.4779	0.4497	0.4555
9	0.9379	0.9338	0.3385	0.4783	0.3964	0.3647	0.3707
Mean	0.9491	0.9436	0.4196	0.6114	0.4970	0.4712	0.4808
Std	0.0048	0.0088	0.0379	0.0573	0.0414	0.0437	0.0448

```
Fitting 10 folds for each of 10 candidates, totalling 100 fits
Original model was better than the tuned model, hence it will be returned.
```

Figure 5.7 Hyper-Tuning XGBoost

The tuned model or the proposed model can be further boosted for better results. The default boosting model for this is carried out by the AdaBoost classifier. Explained in Figure 5.8.



```
▸ AdaBoostClassifier
▸ estimator: XGBClassifier
▸ XGBClassifier
```

Figure 5.8 AdaBoost classifier for XGBoost

The feature importance plot is shown in Figure 5.9 using results provided by XGBoost. The important features include ‘Fault’, ‘BasePolicy’, ‘AddressChange-Claim’, ‘Month’, and others.

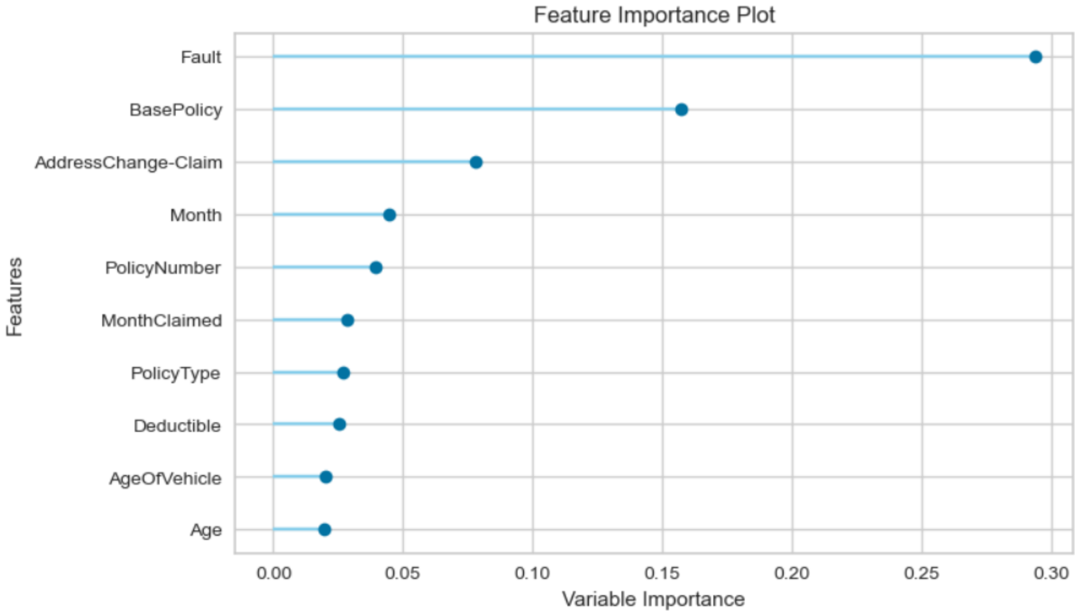
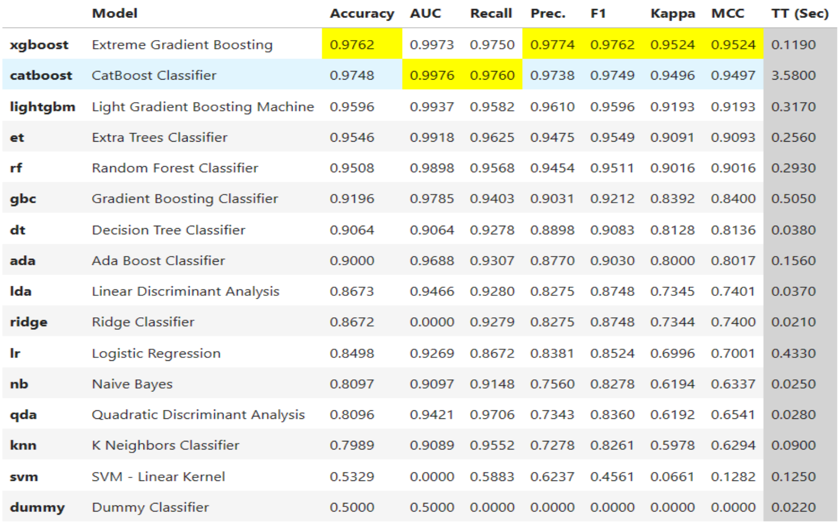


Figure 5.9 Feature importance -XGBoost

Our study on the previous method had an accuracy of 96% for XGBoost without the class imbalance technique. We then used the SMOTE method in Figure 5.10 to address the class imbalance problem, which led to a 98% overall performance improvement for XGBoost. This is followed by CatBoost, which further improved its overall performance. The confusion matrix in Figure 5.11 shows the model's performance, and the AUC-ROC curve in Figure 5.12 represents a perfect model after boosting. We then conducted hyper-tuning, which resulted in a 97% accuracy, confirming that the previous model with 98% accuracy was better than the tuned model. However, boosting the model further using AdaBoost led to only 50% accuracy, indicating that boosting further would make it useless.



	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
xgboost	Extreme Gradient Boosting	0.9762	0.9973	0.9750	0.9774	0.9762	0.9524	0.9524	0.1190
catboost	CatBoost Classifier	0.9748	0.9976	0.9760	0.9738	0.9749	0.9496	0.9497	3.5800
lightgbm	Light Gradient Boosting Machine	0.9596	0.9937	0.9582	0.9610	0.9596	0.9193	0.9193	0.3170
et	Extra Trees Classifier	0.9546	0.9918	0.9625	0.9475	0.9549	0.9091	0.9093	0.2560
rf	Random Forest Classifier	0.9508	0.9898	0.9568	0.9454	0.9511	0.9016	0.9016	0.2930
gbc	Gradient Boosting Classifier	0.9196	0.9785	0.9403	0.9031	0.9212	0.8392	0.8400	0.5050
dt	Decision Tree Classifier	0.9064	0.9064	0.9278	0.8898	0.9083	0.8128	0.8136	0.0380
ada	Ada Boost Classifier	0.9000	0.9688	0.9307	0.8770	0.9030	0.8000	0.8017	0.1560
lda	Linear Discriminant Analysis	0.8673	0.9466	0.9280	0.8275	0.8748	0.7345	0.7401	0.0370
ridge	Ridge Classifier	0.8672	0.0000	0.9279	0.8275	0.8748	0.7344	0.7400	0.0210
lr	Logistic Regression	0.8498	0.9269	0.8672	0.8381	0.8524	0.6996	0.7001	0.4330
nb	Naive Bayes	0.8097	0.9097	0.9148	0.7560	0.8278	0.6194	0.6337	0.0250
qda	Quadratic Discriminant Analysis	0.8096	0.9421	0.9706	0.7343	0.8360	0.6192	0.6541	0.0280
knn	K Neighbors Classifier	0.7989	0.9089	0.9552	0.7278	0.8261	0.5978	0.6294	0.0900
svm	SVM - Linear Kernel	0.5329	0.0000	0.5883	0.6237	0.4561	0.0661	0.1282	0.1250
dummy	Dummy Classifier	0.5000	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0220

Figure 5.10 Boosted - XGBoost output

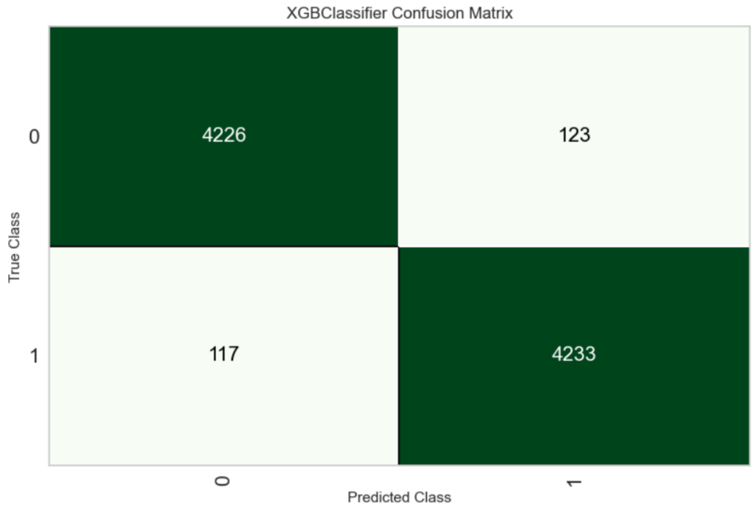


Figure 5.11 Boosted - Confusion matrix

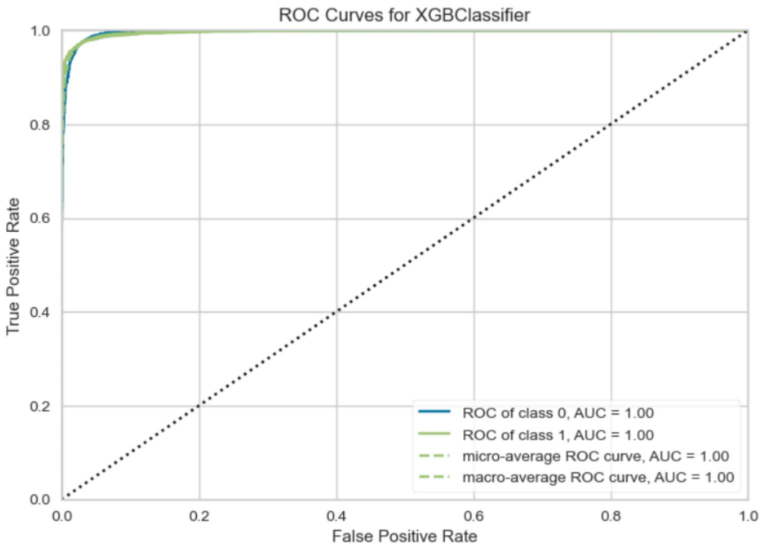
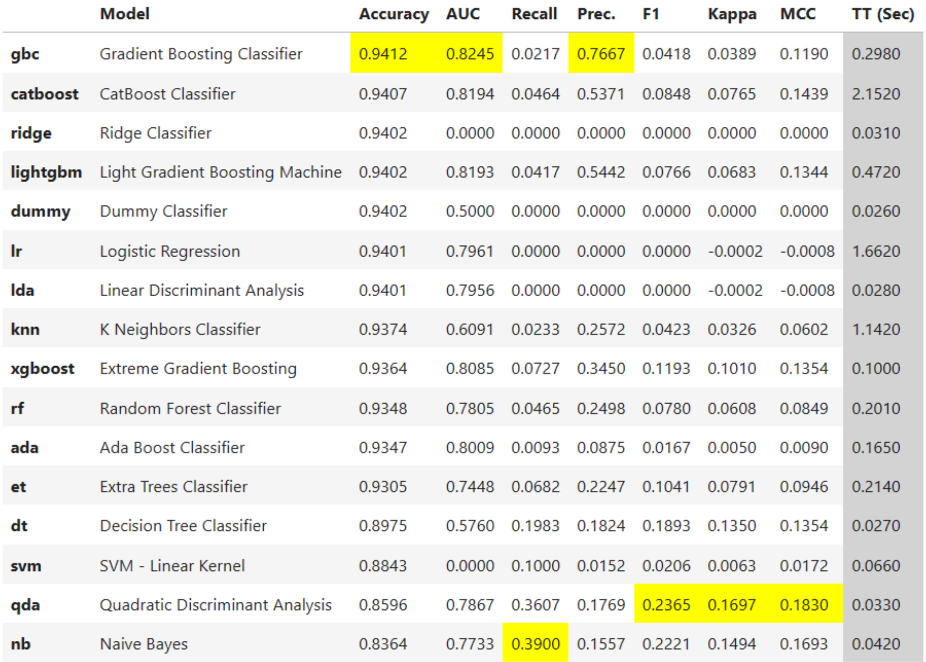


Figure 5.12 Boosted - AUC curve

After feature selection, from the 33 total variables including the target variable based on the p-value 17 features were selected. These features are implemented in PyCaret. Based on the implementation, the Gradient Boosting classifier had an accuracy rate of 94% for the test data. Figures 5.13 & 5.14, provide information on the feature selected model output and confusion matrix visual. Further tuning and boosting the model provides a lower score than the original output. The AUC curve for the model is at 83% making it a good model compared to other models



	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
gbc	Gradient Boosting Classifier	0.9412	0.8245	0.0217	0.7667	0.0418	0.0389	0.1190	0.2980
catboost	CatBoost Classifier	0.9407	0.8194	0.0464	0.5371	0.0848	0.0765	0.1439	2.1520
ridge	Ridge Classifier	0.9402	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0310
lightgbm	Light Gradient Boosting Machine	0.9402	0.8193	0.0417	0.5442	0.0766	0.0683	0.1344	0.4720
dummy	Dummy Classifier	0.9402	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0260
lr	Logistic Regression	0.9401	0.7961	0.0000	0.0000	0.0000	-0.0002	-0.0008	1.6620
lda	Linear Discriminant Analysis	0.9401	0.7956	0.0000	0.0000	0.0000	-0.0002	-0.0008	0.0280
knn	K Neighbors Classifier	0.9374	0.6091	0.0233	0.2572	0.0423	0.0326	0.0602	1.1420
xgboost	Extreme Gradient Boosting	0.9364	0.8085	0.0727	0.3450	0.1193	0.1010	0.1354	0.1000
rf	Random Forest Classifier	0.9348	0.7805	0.0465	0.2498	0.0780	0.0608	0.0849	0.2010
ada	Ada Boost Classifier	0.9347	0.8009	0.0093	0.0875	0.0167	0.0050	0.0090	0.1650
et	Extra Trees Classifier	0.9305	0.7448	0.0682	0.2247	0.1041	0.0791	0.0946	0.2140
dt	Decision Tree Classifier	0.8975	0.5760	0.1983	0.1824	0.1893	0.1350	0.1354	0.0270
svm	SVM - Linear Kernel	0.8843	0.0000	0.1000	0.0152	0.0206	0.0063	0.0172	0.0660
qda	Quadratic Discriminant Analysis	0.8596	0.7867	0.3607	0.1769	0.2365	0.1697	0.1830	0.0330
nb	Naive Bayes	0.8364	0.7733	0.3900	0.1557	0.2221	0.1494	0.1693	0.0420

Figure 5.13 Feature selection – output

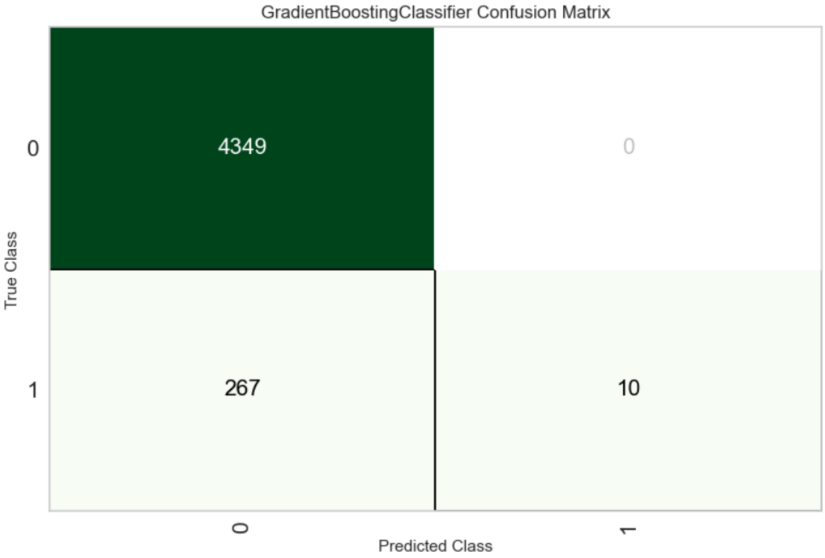
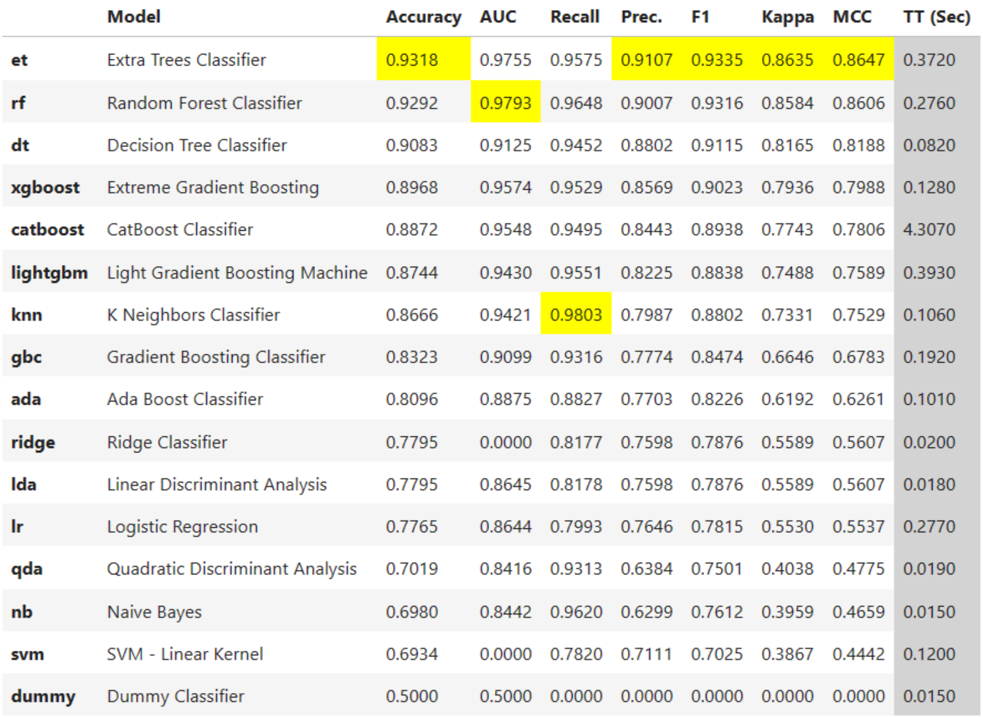


Figure 5.14 Feature selected confusion matrix

After applying a resampling method to the feature-selected data, the Extra-Tree (ET) classifier achieved 93% accuracy, which was better than the performance of other models. On the other hand, the accuracy of the Gradient Boosting Classifier (GBC) decreased to 83%. Figure 5.15 provides a detailed result of the resampling method used on the feature-selected data, which indicates that the tree and forest classifiers are compatible with the data. Additionally, the AUC curve in Figure 5.16 shows that the ET classifier performed well.



	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
et	Extra Trees Classifier	0.9318	0.9755	0.9575	0.9107	0.9335	0.8635	0.8647	0.3720
rf	Random Forest Classifier	0.9292	0.9793	0.9648	0.9007	0.9316	0.8584	0.8606	0.2760
dt	Decision Tree Classifier	0.9083	0.9125	0.9452	0.8802	0.9115	0.8165	0.8188	0.0820
xgboost	Extreme Gradient Boosting	0.8968	0.9574	0.9529	0.8569	0.9023	0.7936	0.7988	0.1280
catboost	CatBoost Classifier	0.8872	0.9548	0.9495	0.8443	0.8938	0.7743	0.7806	4.3070
lightgbm	Light Gradient Boosting Machine	0.8744	0.9430	0.9551	0.8225	0.8838	0.7488	0.7589	0.3930
knn	K Neighbors Classifier	0.8666	0.9421	0.9803	0.7987	0.8802	0.7331	0.7529	0.1060
gbc	Gradient Boosting Classifier	0.8323	0.9099	0.9316	0.7774	0.8474	0.6646	0.6783	0.1920
ada	Ada Boost Classifier	0.8096	0.8875	0.8827	0.7703	0.8226	0.6192	0.6261	0.1010
ridge	Ridge Classifier	0.7795	0.0000	0.8177	0.7598	0.7876	0.5589	0.5607	0.0200
lda	Linear Discriminant Analysis	0.7795	0.8645	0.8178	0.7598	0.7876	0.5589	0.5607	0.0180
lr	Logistic Regression	0.7765	0.8644	0.7993	0.7646	0.7815	0.5530	0.5537	0.2770
qda	Quadratic Discriminant Analysis	0.7019	0.8416	0.9313	0.6384	0.7501	0.4038	0.4775	0.0190
nb	Naive Bayes	0.6980	0.8442	0.9620	0.6299	0.7612	0.3959	0.4659	0.0150
svm	SVM - Linear Kernel	0.6934	0.0000	0.7820	0.7111	0.7025	0.3867	0.4442	0.1200
dummy	Dummy Classifier	0.5000	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0150

Figure 5.15 SMOTE output- Feature selection

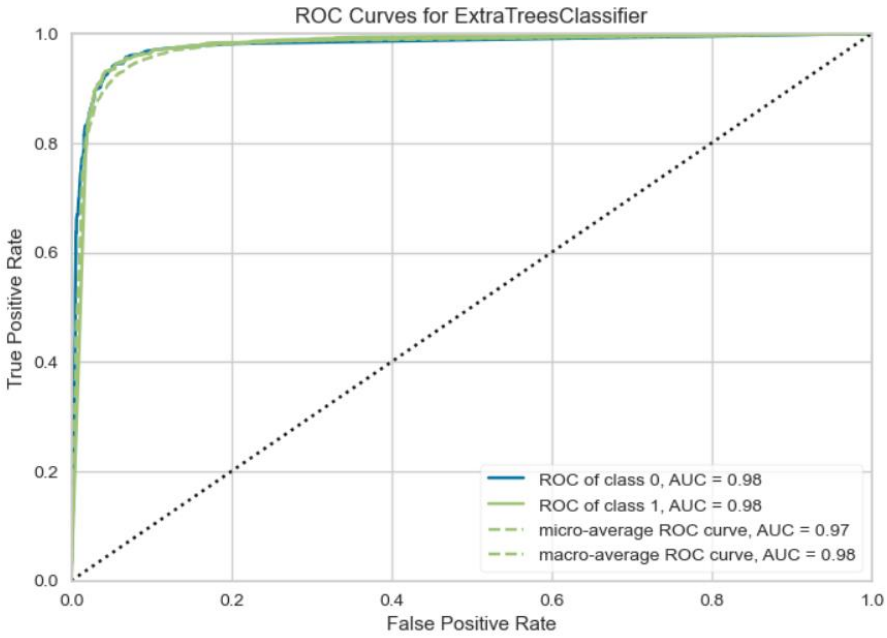
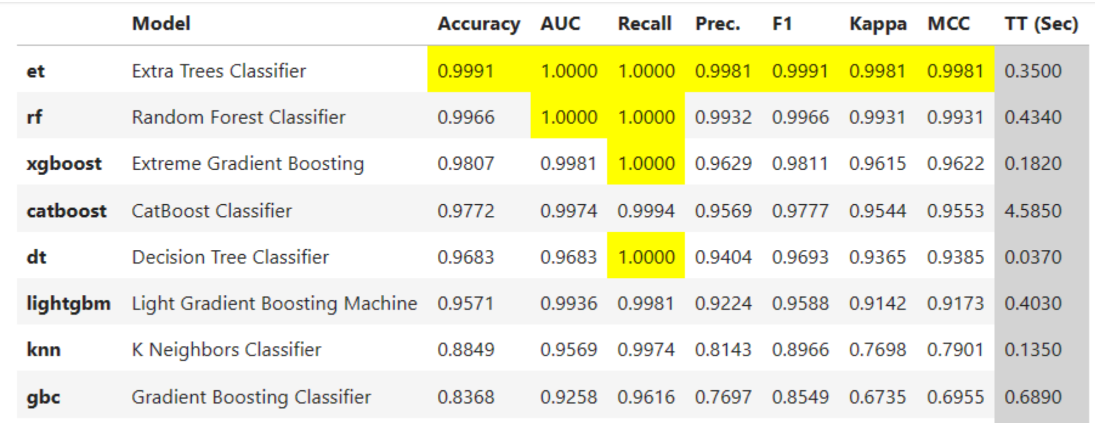


Figure 5.16 SMOTE - AUC curve

Further tuning the model provided an accuracy of 83% and boosting provided the same result 93%, proving that the original model is better than the tuned and boosted model. Based on the results of the resampling method and the AUC curve, we can conclude that the Extra-Tree classifier is a suitable model for the given dataset.

To understand the effects of data balancing in a given dataset, we applied two different sampling techniques to both the original data and the feature-selected data. We used SMOTE, an over-sampling technique, as well as the random over-sampling (ROS) method from the over-sampling technique and the random under-sampling (RUS) method from the under-sampling technique.

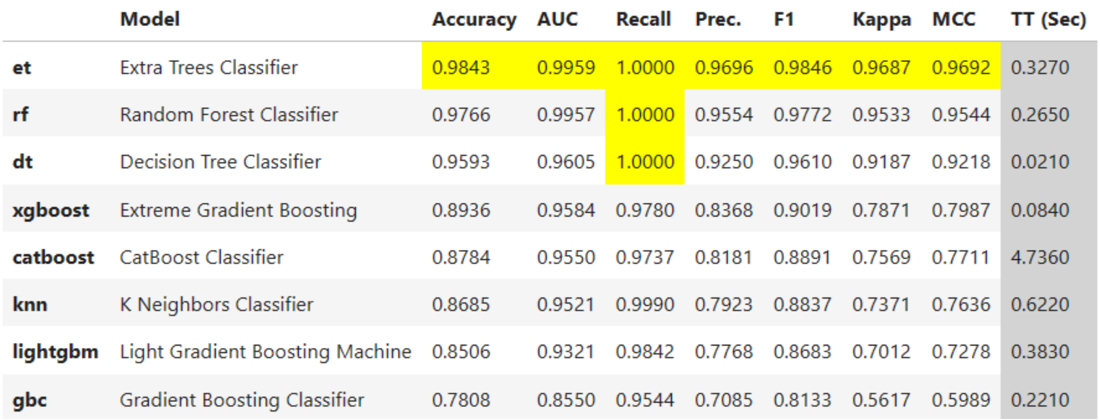
In terms of addressing over-fitting problems in data, the SMOTE method is considered better than ROS. When the ROS method is applied to the original data, the Extra Trees classifier achieves 99% accuracy, which outperforms other models. Figure 5.17 provides a clear visualization of the results.



	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
et	Extra Trees Classifier	0.9991	1.0000	1.0000	0.9981	0.9991	0.9981	0.9981	0.3500
rf	Random Forest Classifier	0.9966	1.0000	1.0000	0.9932	0.9966	0.9931	0.9931	0.4340
xgboost	Extreme Gradient Boosting	0.9807	0.9981	1.0000	0.9629	0.9811	0.9615	0.9622	0.1820
catboost	CatBoost Classifier	0.9772	0.9974	0.9994	0.9569	0.9777	0.9544	0.9553	4.5850
dt	Decision Tree Classifier	0.9683	0.9683	1.0000	0.9404	0.9693	0.9365	0.9385	0.0370
lightgbm	Light Gradient Boosting Machine	0.9571	0.9936	0.9981	0.9224	0.9588	0.9142	0.9173	0.4030
knn	K Neighbors Classifier	0.8849	0.9569	0.9974	0.8143	0.8966	0.7698	0.7901	0.1350
gbc	Gradient Boosting Classifier	0.8368	0.9258	0.9616	0.7697	0.8549	0.6735	0.6955	0.6890

Figure 5.17 Original data -ROS method

The model was further tuned, resulting in an accuracy of 79%. Boosting the model did not increase the accuracy, which remained at 99%. Resampling the feature selection method using the ROS method produced a result similar to the original data. An accuracy rate of 98% was achieved, with an overall performance better than other models. However, further tuning and boosting resulted in decreased accuracy compared to the original output. Figures 5.18 & 5.19 provide a clear understanding of the results, and the AUC curve indicates that the ET classifier is better.



	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
et	Extra Trees Classifier	0.9843	0.9959	1.0000	0.9696	0.9846	0.9687	0.9692	0.3270
rf	Random Forest Classifier	0.9766	0.9957	1.0000	0.9554	0.9772	0.9533	0.9544	0.2650
dt	Decision Tree Classifier	0.9593	0.9605	1.0000	0.9250	0.9610	0.9187	0.9218	0.0210
xgboost	Extreme Gradient Boosting	0.8936	0.9584	0.9780	0.8368	0.9019	0.7871	0.7987	0.0840
catboost	CatBoost Classifier	0.8784	0.9550	0.9737	0.8181	0.8891	0.7569	0.7711	4.7360
knn	K Neighbors Classifier	0.8685	0.9521	0.9990	0.7923	0.8837	0.7371	0.7636	0.6220
lightgbm	Light Gradient Boosting Machine	0.8506	0.9321	0.9842	0.7768	0.8683	0.7012	0.7278	0.3830
gbc	Gradient Boosting Classifier	0.7808	0.8550	0.9544	0.7085	0.8133	0.5617	0.5989	0.2210

Figure 5.18 Feature selected - ROS method

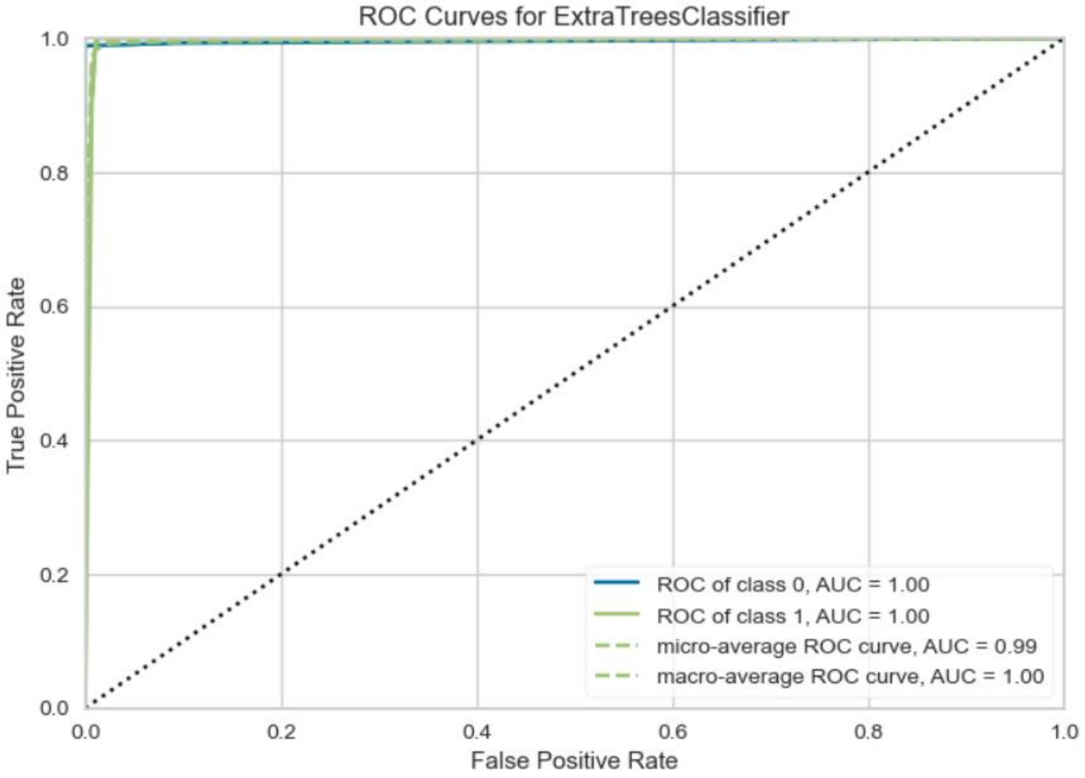
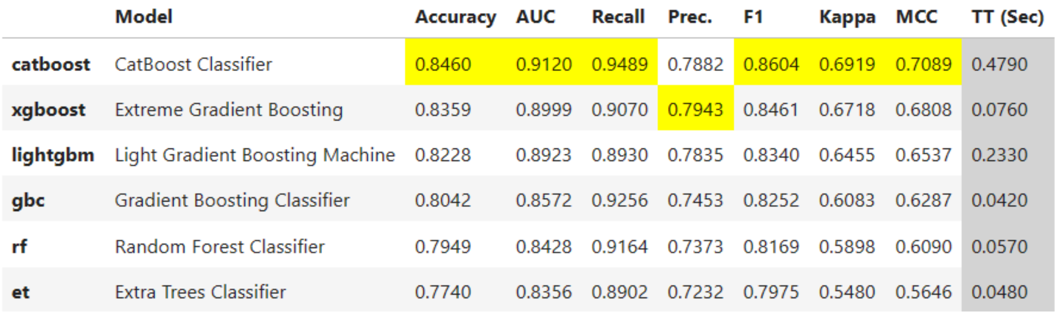


Figure 5.19 ROS- AUC curve

As a result, when using the ROS method on both data the Extra Trees classifier had a good result with 98% accuracy on average. The Extra Trees Classifier is well-suited for handling imbalanced classification scenarios when used with the Random Over-Sampling (ROS) method. This is due to its unique characteristics as an ensemble learning algorithm. Extra Trees employs a randomization process during decision tree construction, which makes it robust to noisy data and helps prevent overfitting. By selecting random subsets of features and data points at each split, the classifier can handle imbalanced datasets effectively. The ensemble nature of Extra Trees, where multiple decision trees are built and their predictions are combined, enhances generalization and allows the model to capture complex patterns in the data. Random feature sampling also serves as an implicit feature selection mechanism, which can be advantageous when working with high-dimensional feature spaces. In summary, the Extra Trees Classifier's combination of randomization, ensemble learning, and robustness to noise makes it an excellent choice for handling imbalanced datasets and implementing techniques like Random Over-Sampling.

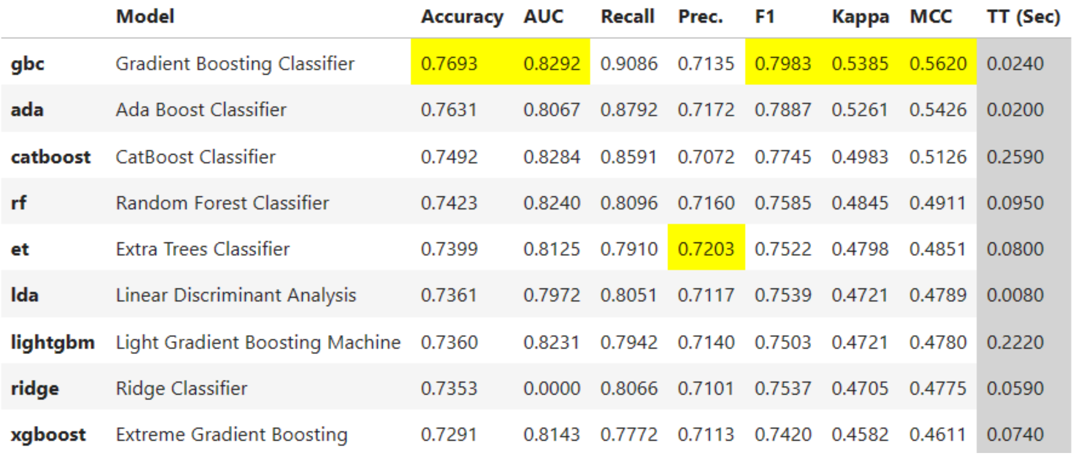
Similarly, when the RUS method is used on these data the results are not promising enough like ROS. Figure 5.20 shows the results of the original data using the RUS method. The CatBoost has an overall performance greater compared to other methods. The CatBoost had an accuracy of 85%. Further tuning and boosting the model provides the same accuracy.



	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
catboost	CatBoost Classifier	0.8460	0.9120	0.9489	0.7882	0.8604	0.6919	0.7089	0.4790
xgboost	Extreme Gradient Boosting	0.8359	0.8999	0.9070	0.7943	0.8461	0.6718	0.6808	0.0760
lightgbm	Light Gradient Boosting Machine	0.8228	0.8923	0.8930	0.7835	0.8340	0.6455	0.6537	0.2330
gbc	Gradient Boosting Classifier	0.8042	0.8572	0.9256	0.7453	0.8252	0.6083	0.6287	0.0420
rf	Random Forest Classifier	0.7949	0.8428	0.9164	0.7373	0.8169	0.5898	0.6090	0.0570
et	Extra Trees Classifier	0.7740	0.8356	0.8902	0.7232	0.7975	0.5480	0.5646	0.0480

Figure 5.20 RUS - Original method

After implementing the feature-selected data using the RUS method, the accuracy of the Gradient Boosting method improved, achieving a score of 77%. However, further tuning and boosting of the model did not yield better results. The accuracy decreased even further in comparison. Figure 5.21 displays the results of the feature-selected RUS method, whereas Figures 5.22 and 5.23 show the AUC curve for both datasets using the RUS method.



	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
gbc	Gradient Boosting Classifier	0.7693	0.8292	0.9086	0.7135	0.7983	0.5385	0.5620	0.0240
ada	Ada Boost Classifier	0.7631	0.8067	0.8792	0.7172	0.7887	0.5261	0.5426	0.0200
catboost	CatBoost Classifier	0.7492	0.8284	0.8591	0.7072	0.7745	0.4983	0.5126	0.2590
rf	Random Forest Classifier	0.7423	0.8240	0.8096	0.7160	0.7585	0.4845	0.4911	0.0950
et	Extra Trees Classifier	0.7399	0.8125	0.7910	0.7203	0.7522	0.4798	0.4851	0.0800
lda	Linear Discriminant Analysis	0.7361	0.7972	0.8051	0.7117	0.7539	0.4721	0.4789	0.0080
lightgbm	Light Gradient Boosting Machine	0.7360	0.8231	0.7942	0.7140	0.7503	0.4721	0.4780	0.2220
ridge	Ridge Classifier	0.7353	0.0000	0.8066	0.7101	0.7537	0.4705	0.4775	0.0590
xgboost	Extreme Gradient Boosting	0.7291	0.8143	0.7772	0.7113	0.7420	0.4582	0.4611	0.0740

Figure 5.21 RUS - Feature data

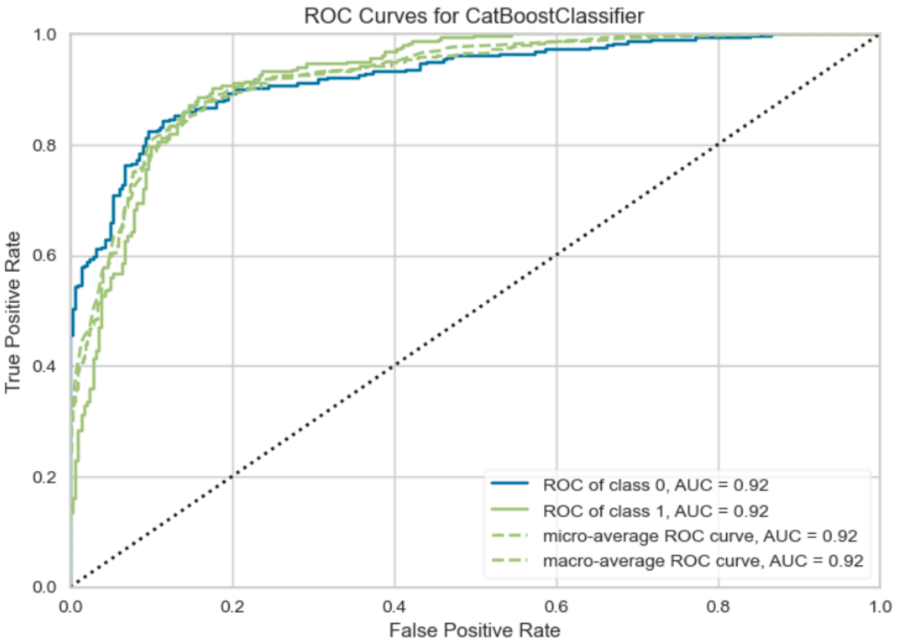


Figure 5.22 AUC - Original data

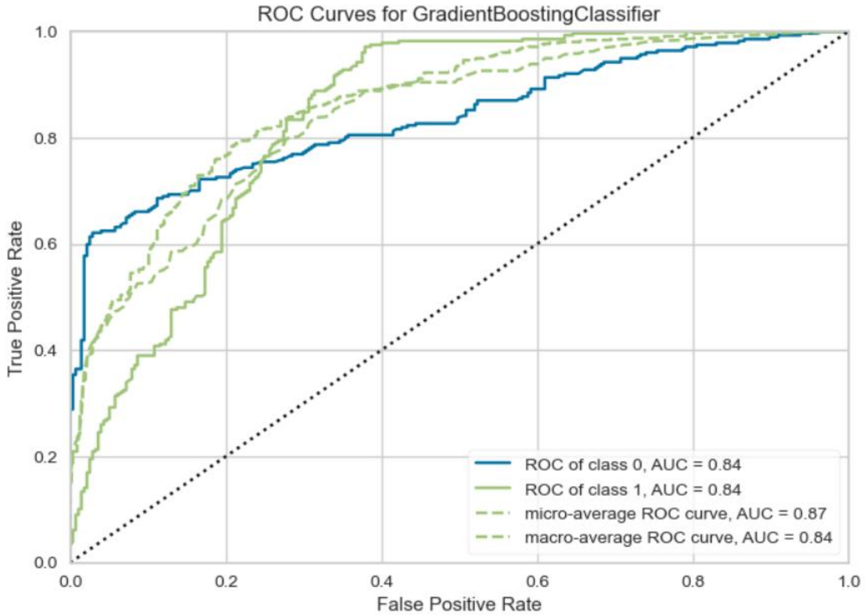


Figure 5.23 AUC- RUS feature data

The top-performing ML models using the RUS method are boosting methods, as shown in Figures 5.20 and 5.21. However, the AUC curve indicates that the proposed model can only be considered average and not good when using the RUS method.

Random Under-Sampling (RUS) is highly compatible with Boosting Family models due to its inherent characteristics that align well with the principles of boosting. The goal of boosting is to enhance the performance of weak learners by combining them sequentially into strong learners, with a focus on misclassified instances. RUS reduces the size of the majority class by randomly removing instances, which naturally amplifies the importance of minority class instances during the training process. This aligns with the boosting strategy of assigning higher weights to misclassified observations, allowing subsequent weak learners to concentrate on correctly classifying the minority class. The synergy between RUS and boosting effectively addresses class imbalance, improving the ability of boosting algorithms to capture subtle patterns within the minority class while preventing them from being overwhelmed by the majority class. This cooperative relationship contributes to improved predictive performance, making RUS a suitable companion for boosting models when dealing with imbalanced datasets.

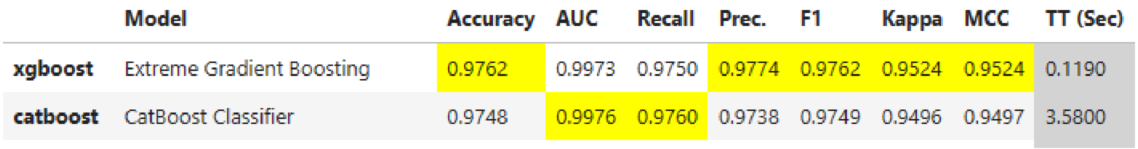
5.3 Results

The accuracy of the original data and the feature-selected data are given in Table 5.1.

Table 5.1 Accuracy Result

Accuracy	Original Data		Feature Selected Data	
	Before SMOTE	After SMOTE	Before SMOTE	After SMOTE
Proposed model	XGB – 95%	XGB - 98%	GBC - 94%	ET - 93%
Tuned model	95%	97%	94%	84%
Boosted model	94%	50%	94%	93%

Based on the analysis of the two datasets implemented in PyCaret, Extreme Gradient Boosting (XGBoost) provided the best output with 98% accuracy and the highest overall performance among all evaluation metrics as illustrated in Figure 5.24. Additionally, when the feature-selected model was implemented in PyCaret, the Gradient Boosting Classifier (GB) produced the best results with an accuracy rate of 94%, despite several random feature selections. Figure 5.25 shows the Shapley Additive explanations (SHAP) value, which demonstrates how the features impact the model's output.



	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
xgboost	Extreme Gradient Boosting	0.9762	0.9973	0.9750	0.9774	0.9762	0.9524	0.9524	0.1190
catboost	CatBoost Classifier	0.9748	0.9976	0.9760	0.9738	0.9749	0.9496	0.9497	3.5800

Figure 5.24 XGBoost Overall performance

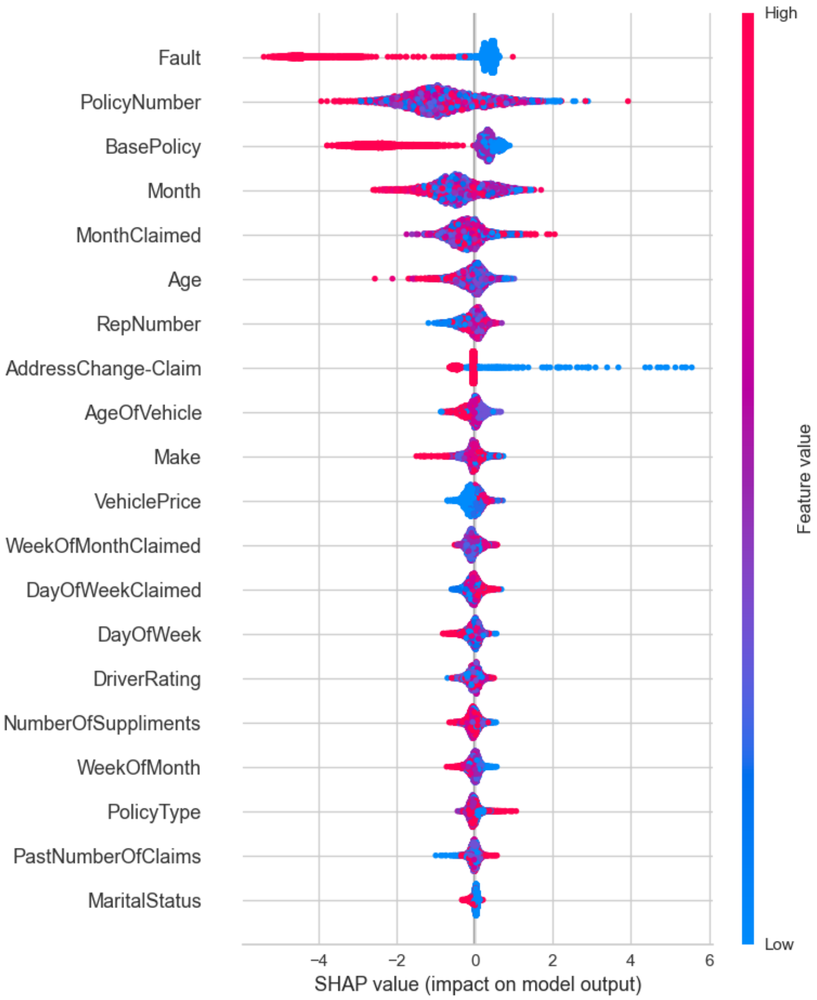


Figure 5.25 SHAP model output

Not all ML models produce absolute results. Data and models vary depending on the usage environment, and there are limitations to all models. PyCaret model limitations are explained briefly.

5.4 Limitations

PyCaret is a machine-learning library that is powerful and user-friendly. However, it has certain limitations when it comes to the specialized domain of fraud detection. One significant challenge is the need for domain-specific knowledge, where understanding fraud patterns, evolving tactics and the intricate behaviours associated with fraudulent activities is crucial. Although PyCaret automates aspects of the machine learning pipeline, it may not capture the nuanced feature engineering required for effective fraud detection. Fraud datasets are often imbalanced, with only a small proportion of instances being fraudulent, and while PyCaret provides options to address imbalanced datasets, the complexities of fraud detection might require more sophisticated techniques like anomaly detection or specialized resampling methods.

Model interpretability is a critical factor in fraud detection, and some of the complex ensemble models generated by PyCaret may lack transparency, making it challenging to explain model decisions. Moreover, the adaptability of models to evolving fraud patterns is essential, and continuous monitoring and updates to the model may be required. Additionally, users should consider data privacy concerns, especially given the sensitive nature of fraud-related data.

Finally, while PyCaret offers automation, the trade-off is a potential reduction in customization and fine-tuning options, which might be necessary for addressing specific requirements in fraud detection scenarios. Therefore, while PyCaret can expedite the model development process, users should complement its use with domain expertise and careful consideration of the unique challenges posed by fraud detection datasets.

5.5 Summary

In this chapter, we implemented the datasets described in Chapter 4 and evaluated both models separately. We provide the corresponding results in each section. After comparing 16 ML models on the two datasets, we found that two different ML models are the best fit for fraud detection. The XGBoost model is the most preferred for fraud detection in the insurance sector or any other sector that deals with fraud detection. In the next chapter, we will discuss the thesis conclusion and future works.

CHAPTER 6:
CONCLUSION AND RECOMMENDATIONS

In this chapter, we will discuss the recommendations and future works based on our thesis.

6.1 Introduction

Our research study aimed to select the best ML model for fraud detection in the insurance sector. In this section, we summarize our key findings by revisiting our research objectives and methodologies employed. Despite certain limitations, the results of our study are significant for fraud detection. We explore unexpected findings, acknowledge study constraints, and present recommendations to address fraud detection. These recommendations are designed to provide actionable insights and contribute to the ongoing discourse in fraud detection.

6.2 Conclusion

In an economy-based society, industries and sectors strive for economic and financial stability. However, fraud claims can significantly impact this stability. Dealing with fraud claims is often a long, tedious, and error-prone task. With the advent of digitalization, fraud claims can take many forms. Fortunately, Machine Learning techniques have made fraud detection more efficient and effective. In this study, we propose a model for detecting fraud in automobile insurance. We tested various Machine Learning models to identify the best fit for fraud detection. Our findings demonstrate that fraud detection depends on several factors, including feature selection, data balancing, data transformation, and data cleaning. To reduce programming time and conduct a comparative study, we used PyCaret. Our study shows that XGBoost with SMOTE on original data provides the best accuracy of 98% for fraud detection in the insurance sector. Even with feature selection, our model still performs better.

It is important to note that the data used in this study was collected between 1994-1996 and is the only available data with a high-class imbalance issue for fraud detection. As the field of fraud detection has evolved since then, with the introduction of many new ML models and even AI, it is recommended to use updated data and models for fraud detection in the real world. This study suggests the strategic integration of XGBoost, along with thoughtful sampling methodologies and the use of appropriate performance metrics, to enhance the overall security and reliability of financial systems in the face of evolving fraudulent threats.

6.3 Recommendation and Future Work

Fraud detection has come a long way with the development of many new machine-learning models. However, there are still cases where the machine misses new fraudulent patterns. This is where AI can come in, as it can learn and adapt on its own, allowing it to detect new patterns as they emerge. The faster fraud is detected, the better it is for the growth of any industry. Many neural network models based on AI have been tested and implemented for fraud detection. In the future, it is recommended to use the best and most cost-effective neural networks for fraud detection. Additionally, the emerging field of Generative AI (genAI) (Stevens, 2023) may be a good area for research in real-time fraud detection, as it has been utilized for detecting deep fakes, voice spoofing, and other fraudulent activities.

REFERENCES

Abdallah, A., Maarof, M.A. and Zainal, A., (2016) Fraud detection system: A survey. *Journal of Network and Computer Applications*, [online] 68, pp.90–113. Available at: https://www.sciencedirect.com/science/article/pii/S1084804516300571.

Agrawal, N. and Panigrahi, S., (2023) A Comparative Analysis of Fraud Detection in Healthcare using Data Balancing & Machine Learning Techniques. In: *2023 International Conference on Communication, Circuits, and Systems (IC3S)*. pp.1–4.

Alamri, M. and Ykhlef, M., (2022) Survey of Credit Card Anomaly and Fraud Detection Using Sampling Techniques. *Electronics*, [online] 1123. Available at: https://www.mdpi.com/2079-9292/11/23/4003.

Alrais, A.I., (2022) Fraudulent Insurance Claims Detection Using Machine Learning.

Anon (2023) *Vehicle Insurance Fraud Detection*. [online] Available at: https://www.kaggle.com/datasets/khusheekapoor/vehicle-insurance-fraud-detection/data [Accessed 1 Dec. 2023].

Anon (2023) *What is Logistic regression? | IBM*. [online] Available at: https://www.ibm.com/topics/logistic-regression [Accessed 27 Oct. 2023].

Artís, M., Ayuso, M. and Guillén, M., (2002) Detection of Automobile Insurance Fraud With Discrete Choice Models and Misclassified Claims. *Journal of Risk and Insurance*, [online] 693, pp.325–340. Available at: https://onlinelibrary.wiley.com/doi/abs/10.1111/1539-6975.00022.

Aslam, F., Hunjra, A.I., Ftiti, Z., Louhichi, W. and Shams, T., (2022) Insurance fraud detection: Evidence from artificial intelligence and machine learning. *Research in International Business and Finance*, [online] 62, p.101744. Available at: https://www.sciencedirect.com/science/article/pii/S0275531922001325.

Badriyah, T., Rahmaniah, L. and Syarif, I., (2018) Nearest Neighbour and Statistics Method based for Detecting Fraud in Auto Insurance. In: *2018 International Conference on Applied Engineering (ICAE)*. pp.1–5.

Benkessirat, A. and Benblidia, N., (2019) Fundamentals of Feature Selection: An Overview and Comparison. In: *2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA)*. pp.1–6.

Bento, C., (2021) *Decision Tree Classifier explained in real-life: picking a vacation destination | by Carolina Bento | Towards Data Science*. [online] https://medium.com/. Available at: https://towardsdatascience.com/decision-tree-classifier-explained-in-real-life-picking-a-vacation-destination-6226b2b60575 [Accessed 27 Oct. 2023].

Biau, G. and Scornet, E., (2016) A random forest guided tour. *TEST*, [online] 252, pp.197–227. Available at: https://doi.org/10.1007/s11749-016-0481-7.

Brockett, P.L., Derrig, R.A., Golden, L.L., Levine, A. and Alpert, M., (2002) Fraud Classification Using Principal Component Analysis of RIDITs. *Journal of Risk and Insurance*, [online] 693, pp.341–371. Available at: https://onlinelibrary.wiley.com/doi/abs/10.1111/1539-6975.00027.

Cai, J., Luo, J., Wang, S. and Yang, S., (2018) Feature selection in machine learning: A new perspective. *Neurocomputing*, [online] 300, pp.70–79. Available at: https://www.sciencedirect.com/science/article/pii/S0925231218302911.

Chatfield, C., (1986) Exploratory data analysis. *European Journal of Operational Research*, [online] 231, pp.5–13. Available at: https://www.sciencedirect.com/science/article/pii/0377221786902092.

Chen, T. and Guestrin, C., (2016) XGBoost: A Scalable Tree Boosting System. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16. [online] New York, NY, USA: Association for Computing Machinery, pp.785–794. Available at: https://doi.org/10.1145/2939672.2939785.

Chicco, D. and Jurman, G., (2020) The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics*, [online] 211, p.6. Available at: https://doi.org/10.1186/s12864-019-6413-7.

Claver\'\ia Navarrete, A. and Carrasco Gallego, A., (2021) Neural network algorithms for fraud detection: a comparison of the complementary techniques in the last five years. *Journal of Management Information and Decision Sciences, 24 (special 1), 1-16.*

DeBarr, D. and Wechsler, H., (2013) Fraud detection using reputation features, SVMs, and random forests. In: *Proceedings of the International Conference on Data Science (ICDATA)*. p.1.

Debener, J., Heinke, V. and Kriebel, J., (2023) Detecting insurance fraud using supervised and unsupervised machine learning. *Journal of Risk and Insurance*, [online] 903, pp.743–768. Available at: https://onlinelibrary.wiley.com/doi/abs/10.1111/jori.12427.

Derrig, R.A., (2002) Insurance Fraud. *Journal of Risk and Insurance*, [online] 693, pp.271–287. Available at: https://onlinelibrary.wiley.com/doi/abs/10.1111/1539-6975.00026.

Dhieb, N., Ghazzai, H., Besbes, H. and Massoud, Y., (2019) Extreme Gradient Boosting Machine Learning Algorithm For Safe Auto Insurance Operations. In: *2019 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*. pp.1–5.

Elssied, N.O.F., Ibrahim, O. and Osman, A.H., (2014) A novel feature selection based on one-way anova f-test for e-mail spam classification. *Research Journal of Applied Sciences, Engineering and Technology*, 73, pp.625–638.

Ferrer, L., (2023) *Analysis and Comparison of Classification Metrics*.

Feyen, E., Lester, R.R. and Rocha, R. de R., (2011) What drives the development of the insurance sector? An empirical analysis based on a panel of developed and developing countries. *An Empirical Analysis Based on a Panel of Developed and Developing Countries (February 1, 2011). World Bank Policy Research Working Paper*, 5572.

Frenzel, P., (2023) *#KB K-Nearest Neighbors (KNN) — Intro | by Prof. Frenzel | Oct, 2023 | Medium*. [online] https://medium.com/. Available at: https://prof-frenzel.medium.com/dear-friends-854ba66361d7 [Accessed 27 Oct. 2023].

Gain, U. and Hotti, V., (2021) Low-code AutoML-augmented Data Pipeline – A Review and Experiments. *Journal of Physics: Conference Series*, [online] 18281, p.12015. Available at: https://dx.doi.org/10.1088/1742-6596/1828/1/012015.

Gepp, A., Wilson, J.H., Kumar, K. and Bhattacharya, S., (2012) A comparative analysis of decision trees vis-a-vis other computational data mining techniques in automotive insurance fraud detection. *Journal of Data Science*, 103, pp.537–561.

Goyal, S., (2021) *Evaluation Metrics for Classification Models | by Shweta Goyal | Analytics Vidhya | Medium*. [online] Analytics Vidhya. Available at: https://medium.com/analytics-vidhya/evaluation-metrics-for-classification-models-e2f0d8009d69 [Accessed 4 Nov. 2023].

Gupta, R.Y., Mudigonda, S.S. and Baruah, P.K., (2012) TGANs with machine learning models in automobile insurance fraud detection and comparative study with other data imbalance techniques. *International Journal of Recent Technology and Engineering*, 95, pp.236–244.

Gupta, R.Y., Sai Mudigonda, S., Kandala, P.K. and Baruah, P.K., (2019) Implementation of a Predictive Model for Fraud Detection in Motor Insurance using Gradient Boosting Method and Validation with Actuarial Models. In: *2019 IEEE International Conference on Clean Energy and Energy Efficient Electronics Circuit for Sustainable Development (INCCES)*. pp.1–6.

Hanafy, M. and Ming, R., (2021) Machine Learning Approaches for Auto Insurance Big Data. *Risks*, [online] 92. Available at: https://www.mdpi.com/2227-9091/9/2/42.

Hancock, J.T. and Khoshgoftaar, T.M., (2021) Gradient Boosted Decision Tree Algorithms for Medicare Fraud Detection. *SN Computer Science*, [online] 24, p.268. Available at: https://doi.org/10.1007/s42979-021-00655-z.

Harjai, S., Khatri, S.K. and Singh, G., (2019) Detecting Fraudulent Insurance Claims Using

Random Forests and Synthetic Minority Oversampling Technique. In: *2019 4th International Conference on Information Systems and Computer Networks (ISCON)*. pp.123–128.

de Holanda, W.D., e Silva, L.C. and de Carvalho César Sobrinho, Á.A., (2024) Machine learning models for predicting hospitalization and mortality risks of COVID-19 patients. *Expert Systems with Applications*, [online] 240, p.122670. Available at: https://www.sciencedirect.com/science/article/pii/S095741742303172X.

Huang, C., Tu, P.-M. and Lin, C.-Y., (2023) Making Data Analysis Easier: A Case Study on Credit Card Fraud Detection Based on PyCaret. In: *Proceedings of the 2023 4th International Conference on Management Science and Engineering Management (ICMSEM 2023)*. [online] Atlantis Press, pp.1203–1211. Available at: https://doi.org/10.2991/978-94-6463-256-9_122.

Imaam, F., Subasinghe, A., Kasthuriarachchi, H., Fernando, S., Haddela, P. and Pemadasa, N., (2021) Moderate Automobile Accident Claim Process Automation Using Machine Learning. In: *2021 International Conference on Computer Communication and Informatics (ICCCI)*. pp.1–6.

Itri, B., Mohamed, Y., Mohammed, Q. and Omar, B., (2019) Performance comparative study of machine learning algorithms for automobile insurance fraud detection. In: *2019 Third International Conference on Intelligent Computing in Data Sciences (ICDS)*. pp.1–4.

Jayasingh, S. and Swain, A.K., (2011) Neural Network in Fraud Detection.

Jiang, D., Lin, W. and Raghavan, N., (2020) A Novel Framework for Semiconductor Manufacturing Final Test Yield Classification Using Machine Learning Techniques. *IEEE Access*, 8, pp.197885–197895.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q. and Liu, T.-Y., (2017) LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17. Red Hook, NY, USA: Curran Associates Inc., pp.3149–3157.

Kini, A., Chelluru, R., Naik, K., Naik, D., Aswale, S. and Shetgaonkar, P., (2022) Automobile Insurance Fraud Detection: An Overview. In: *2022 3rd International Conference on Intelligent Engineering and Management (ICIEM)*. pp.7–12.

Kotb, M.H. and Ming, R., (2021) Comparing SMOTE family techniques in predicting insurance premium defaulting using machine learning models. *International Journal of Advanced Computer Science and Applications*, 129.

Li, Y., Yan, C., Liu, W. and Li, M., (2018) A principle component analysis-based random forest with the potential nearest neighbor method for automobile insurance fraud identification. *Applied Soft Computing*, [online] 70, pp.1000–1009. Available at:

https://www.sciencedirect.com/science/article/pii/S1568494617304386.

Maina, D.G., Moso, J.C. and Gikunda, P.K., (2023) Detecting Fraud in Motor Insurance Claims Using XGBoost Algorithm with SMOTE. In: *2023 International Conference on Information and Communication Technology for Development for Africa (ICT4DA)*. pp.61–66.

Mary, A.J. and Claret, S.P.A., (2021) Imbalanced Classification Problems: Systematic Study and Challenges in Healthcare Insurance Fraud Detection. In: *2021 5th International Conference on Trends in Electronics and Informatics (ICOEI)*. pp.1049–1055.

Moez, A., (2020) *PyCaret: An open source, low-code machine learning library in Python. Retrieved July 11, 2022*.

Moharekar, T. and Pol, U., (2022) Thyroid Disease Detection Using Machine Learning and Pycaret. *Specialusis Ugdymas*, 1, pp.10150–10160.

Nur Prasasti, I.M., Dhini, A. and Laoh, E., (2020) Automobile Insurance Fraud Detection using Supervised Classifiers. In: *2020 International Workshop on Big Data and Information Security (IWBIS)*. pp.47–52.

Omar, S.J., Fred, K. and Swaib, K.K., (2018) A State-of-the-Art Review of Machine Learning Techniques for Fraud Detection Research. In: *Proceedings of the 2018 International Conference on Software Engineering in Africa*, SEiA '18. [online] New York, NY, USA: Association for Computing Machinery, pp.11–19. Available at: https://doi.org/10.1145/3195528.3195534.

Own, R.M., Salem, S.A. and Mohamed, A.E., (2021) TCCFD: An Efficient Tree-based Framework for Credit Card Fraud Detection. In: *2021 16th International Conference on Computer Engineering and Systems (ICCES)*. pp.1–6.

Panigrahi, S. and Palkar, B., (2018) Comparative Analysis on Classification Algorithms of Auto-Insurance Fraud Detection based on Feature Selection Algorithms. *International Journal of Computer Sciences and Engineering*, 6, pp.72–77.

Pant, A., (2023) *Introduction to Logistic Regression | by Ayush Pant | Towards Data Science*. [online] https://medium.com/. Available at: https://towardsdatascience.com/introduction-to-logistic-regression-66248243c148 [Accessed 27 Oct. 2023].

Patel, D.K. and Subudhi, S., (2019) Application of Extreme Learning Machine in Detecting Auto Insurance Fraud. In: *2019 International Conference on Applied Machine Learning (ICAML)*. pp.78–81.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, E., (2011) Scikit-learn: Machine Learning in {P}ython. *Journal of Machine Learning Research*, 12, pp.2825–2830.

Perangin-Angin, D.J. and Bachtiar, F.A., (2021) Classification of Stress in Office Work Activities Using Extreme Learning Machine Algorithm and One-Way ANOVA F-Test Feature Selection. In: *2021 4th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*. pp.503–508.

Phua, C., Alahakoon, D. and Lee, V., (2004) Minority Report in Fraud Detection: Classification of Skewed Data. *SIGKDD Explor. Newsl.*, [online] 61, pp.50–59. Available at: https://doi.org/10.1145/1007730.1007738.

Pol, U.R. and Sawant, T.U., (2021) Automl: Building An Classfication Model With Pycaret. *Ymer*, 20, pp.547–552.

Rai, N., Baruah, P.K., Mudigonda, S.S. and Kandala, P.K., (2018) Fraud Detection Supervised Machine Learning Models for an Automobile Insurance. *Int. J. Sci. Eng. Res*, 911, pp.473–479.

RB, A. and KR, S.K., (2021) Credit card fraud detection using artificial neural network. *Global Transitions Proceedings*, [online] 21, pp.35–41. Available at: https://www.sciencedirect.com/science/article/pii/S2666285X21000066.

Roy, R. and George, K.T., (2017) Detecting insurance claims fraud using machine learning techniques. In: *2017 International Conference on Circuit ,Power and Computing Technologies (ICCPCT)*. pp.1–6.

Rukhsar, L., Bangyal, W.H., Nisar, K. and Nisar, S., (2022) Prediction of insurance fraud detection using machine learning algorithms. *Mehran University Research Journal of Engineering \& Technology*, 411, pp.33–40.

S.Patil, N., Kamanavalli, S., Hiregoudar, S., Jadhav, S., Kanakraddi, S. and Hiremath, N.D., (2021) Vehicle Insurance Fraud Detection System Using Robotic Process Automation and Machine Learning. In: *2021 International Conference on Intelligent Technologies (CONIT)*. pp.1–5.

Sagar, G. and Syrovatskyi, V., (2022) Artificial Intelligence: Making Machines Learn. In: *Technical Building Blocks: A Technology Reference for Real-world Product Development*. [online] Berkeley, CA: Apress, pp.213–274. Available at: https://doi.org/10.1007/978-1-4842-8658-6_5.

Santos, R.N., Yamouni, S., Albiero, B., Vicente, R., A. Silva, J., F. B. Souza, T., C. M. Freitas Souza, M. and Lei, Z., (2021) Gradient boosting and Shapley additive explanations for fraud detection in electricity distribution grids. *International Transactions on Electrical*

Energy Systems, [online] 319, p.e13046. Available at: https://onlinelibrary.wiley.com/doi/abs/10.1002/2050-7038.13046.

Sarangpure, N., Dhamde, V., Roge, A., Doye, J., Patle, S. and Tamboli, S., (2023) Automating the Machine Learning Process using PyCaret and Streamlit. In: *2023 2nd International Conference for Innovation in Technology (INOCON)*. pp.1–5.

Singhal, A., Singhal, N., Divya and Sharma, K., (2023) Machine Learning Methods for Detecting Car Insurance Fraud: Comparative Analysis. In: *2023 3rd International Conference on Intelligent Technologies (CONIT)*. pp.1–5.

Stevens, L., (2023) *Generative AI and Fraud – What are the risks that firms face? | Deloitte UK*. [online] Available at: https://www2.deloitte.com/uk/en/blog/auditandassurance/2023/generative-ai-and-fraud-what-are-the-risks-that-firms-face.html [Accessed 4 Dec. 2023].

Subasi, A., (2020) Chapter 3 - Machine learning techniques. In: A. Subasi, ed., *Practical Machine Learning for Data Analysis Using Python*. [online] Academic Press, pp.91–202. Available at: https://www.sciencedirect.com/science/article/pii/B9780128213797000035.

Subudhi, S. and Panigrahi, S., (2018) Effect of Class Imbalanceness in Detecting Automobile Insurance Fraud. In: *2018 2nd International Conference on Data Science and Business Analytics (ICDSBA)*. pp.528–531.

Sundarkumar, G.G. and Ravi, V., (2015) A novel hybrid undersampling method for mining unbalanced datasets in banking and insurance. *Engineering Applications of Artificial Intelligence*, [online] 37, pp.368–377. Available at: https://www.sciencedirect.com/science/article/pii/S0952197614002395.

Tao, H., Zhixin, L. and Xiaodong, S., (2012) Insurance fraud identification research based on fuzzy support vector machine with dual membership. In: *2012 International Conference on Information Management, Innovation Management and Industrial Engineering*. pp.457–460.

Tongesai, M., Mbizo, G. and Zvarevashe, K., (2022) Insurance Fraud Detection using Machine Learning. In: *2022 1st Zimbabwe Conference of Information and Communication Technologies (ZCICT)*. pp.1–6.

Urunkar, A., Khot, A., Bhat, R. and Mudegol, N., (2022) Fraud Detection and Analysis for Insurance Claim using Machine Learning. In: *2022 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES)*. pp.406–411.

Viaene, S., Ayuso, M., Guillen, M., Van Gheel, D. and Dedene, G., (2007) Strategies for detecting fraudulent claims in the automobile insurance industry. *European Journal of Operational Research*, [online] 1761, pp.565–583. Available at:

https://www.sciencedirect.com/science/article/pii/S0377221705006405.

Viaene, S. and Dedene, G., (2004) Insurance Fraud: Issues and Challenges. *The Geneva Papers on Risk and Insurance - Issues and Practice*, [online] 292, pp.313–333. Available at: https://doi.org/10.1111/j.1468-0440.2004.00290.x.

Viaene, S., Derrig, R.A., Baesens, B. and Dedene, G., (2002) A Comparison of State-of-the-Art Classification Techniques for Expert Automobile Insurance Claim Fraud Detection. *Journal of Risk and Insurance*, [online] 693, pp.373–421. Available at: https://onlinelibrary.wiley.com/doi/abs/10.1111/1539-6975.00023.

Vujovic, Z., (2021) Classification Model Evaluation Metrics. *International Journal of Advanced Computer Science and Applications*, Volume 12, pp.599–606.

Vyas, S. and Serasiya, S., (2022) Fraud Detection in Insurance Claim System: A Review. In: *2022 Second International Conference on Artificial Intelligence and Smart Energy (ICAIS)*. pp.922–927.

Wang, Y. and Xu, W., (2018) Leveraging deep learning with LDA-based text analytics to detect automobile insurance fraud. *Decision Support Systems*, [online] 105, pp.87–95. Available at: https://www.sciencedirect.com/science/article/pii/S0167923617302130.

Wei, Y., Qi, Y., Ma, Q., Liu, Z., Shen, C. and Fang, C., (2020) Fraud Detection by Machine Learning. In: *2020 2nd International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI)*. pp.101–115.

West, J. and Bhattacharya, M., (2016) Intelligent financial fraud detection: A comprehensive review. *Computers & Security*, [online] 57, pp.47–66. Available at: https://www.sciencedirect.com/science/article/pii/S0167404815001261.

Whig, P., Gupta, K., Jiwani, N., Jupalle, H., Kouser, S. and Alam, N., (2023) A novel method for diabetes classification and prediction with Pycaret. *Microsystem Technologies*. [online] Available at: https://doi.org/10.1007/s00542-023-05473-2.

Whitfield, B., (2023) *Principal Component Analysis (PCA) Explained | Built In*. [online] https://builtin.com/. Available at: https://builtin.com/data-science/step-step-explanation-principal-component-analysis [Accessed 27 Oct. 2023].

Wongvorachan, T., He, S. and Bulut, O., (2023) A Comparison of Undersampling, Oversampling, and SMOTE Methods for Dealing with Imbalanced Classification in Educational Data Mining. *Information*, [online] 141. Available at: https://www.mdpi.com/2078-2489/14/1/54.

Xia, H., Zhou, Y. and Zhang, Z., (2022) Auto insurance fraud identification based on a CNN-LSTM fusion deep learning model. *International Journal of Ad Hoc and Ubiquitous*

Computing, [online] 391–2, pp.37–45. Available at: https://www.inderscienceonline.com/doi/abs/10.1504/IJAHUC.2022.120943.

Xu, W., Wang, S., Zhang, D. and Yang, B., (2011) Random Rough Subspace Based Neural Network Ensemble for Insurance Fraud Detection. In: *2011 Fourth International Joint Conference on Computational Sciences and Optimization*. pp.1276–1280.

Yan, C., Li, Y., Liu, W., Li, M., Chen, J. and Wang, L., (2020) An artificial bee colony-based kernel ridge regression for automobile insurance fraud identification. *Neurocomputing*, [online] 393, pp.115–125. Available at: https://www.sciencedirect.com/science/article/pii/S0925231219310550.

Yaram, S., (2016) Machine learning algorithms for document clustering and fraud detection. In: *2016 International Conference on Data Science and Engineering (ICDSE)*. pp.1–6.

APPENDIX A: RESEARCH PROPOSAL

1. Background

Insurance – protection for a financial loss. Insurance serves as a safeguard against financial losses, providing coverage or compensation for damages incurred by the insured individual or entity. The Association of Certified Fraud Examiners (ACFE) defines fraud as the deliberate use of an organization's resources for personal gain.(Abdallah et al., 2016)(West and Bhattacharya, 2016). It has assumed a pivotal role across industries, with diverse types such as medical, fire, vehicle, and life insurance, among others. While insurance offers protection, it also attracts instances of fraudulent activities, aimed at illicitly obtaining the insured amount through deceptive means(Kini et al., 2022). This study delves into the realm of vehicle insurance, a common target for fraudsters seeking to exploit false pretences to secure insurance payouts. "Insurance fraud" denotes acts involving criminal deception to gain insurance proceeds. In a report from the Federal Bureau of Investigations (FBI), the US is taking a financial loss of $40 billion annually. This means that the average American family pays between $400 and $700 more in premiums due to insurance fraud.(Aslam et al., 2022). A 2022 survey by CAIF (Coalition Against Insurance Fraud) in the US highlighted nearly $308.6 billion in fraudulent damage claims, contributing to substantial financial losses. Auto insurers encounter annual losses of approximately $29 billion due to fraudulent claims (Insurance Information Institute). Similarly, the ABI (Association of British Insurers) conducted a UK survey in 2021, revealing nearly 89,000 fraudulent insurance claims with an average value of £12,283.

In the field of Data, Machine Learning (ML) is of utmost importance, as it can be used to solve many real-world problems as it happens(S.Patil et al., 2021). Data analytics(DA) has emerged be an indispensable tool in the data world, where ML is now positioned as an added value along with Natural Language Processing (NLP), Computer Vision (CV), and the never-ending Artificial intelligence (AI)(Imaam et al., 2021). The DA is significantly crucial in the analysis of data. The ML system has significantly replaced the labour and time that humans once invested in a vast scope, thereby lessening expenses and enhancing productivity(Artís et al., 2002).

Hence, we are implementing DA in our research to analyse and utilise the powerful ML to create a solution for fraudulent claims in the insurance industry.

2. Problem Statement

In recent years, the global insurance market has been significantly influenced by the changing modern world. This has resulted in a broader range of insurable items, encompassing various valuable products that individuals deem worthy of protection. The fundamental principle of insurance lies in its ability to offer a safety net against unforeseen circumstances, such as accidents or damages. In these cases, the insured amount functions as a means to relieve immediate financial burdens. However, the insurance claims process poses a formidable challenge due to its intricate nature, which is further aggravated by the widespread occurrence of fraudulent activities, where individuals other than legitimate victim attempt to claim insurance payouts, leading to what is commonly referred to as insurance fraud. This widespread issue extends across boundaries, and effectively monitoring and assessing the authenticity of each claim manually poses a significant challenge. To address this concern, the integration of automated systems has emerged as a potential solution. These systems have the ability along with the capability to learn from historical data patterns and input information, thereby enhancing the efficiency of fraud identification and prevention. This paradigm shift has propelled various machine learning (ML) algorithms into the spotlight, offering promise in predicting and pre-empting instances of insurance fraud. However, it is crucial to note that the effectiveness and precision of these algorithms can significantly vary. This research paper undertakes a methodical and comprehensive comparative study of diverse ML algorithms, meticulously examining their merits and limitations. The ultimate objective of this exploration is to ascertain the most adept algorithm, guided by predefined criteria for evaluation. Through such a rigorous inquiry, this study aims to contribute to the advancement of fraud detection techniques within the domain of insurance.

2.1 Related work

Considerable research has been dedicated to vehicle insurance, encompassing a diverse array of concepts and models. For instance, one study employed Robotic Process Automation to enhance fraud detection and diminish human intervention. Additionally, various machine learning (ML) models such as LR, DT, LDA, and KNN have been explored. In a comparative evaluation, LDA emerged as prominent, achieving a 90% accuracy rate (S.Patil et al., 2021). Similarly, different papers have harnessed ML models in conjunction with NB and RF. Notably, DT and RF consistently maintained a 90% accuracy rate, utilizing the calculation algorithm PYSpark, yielding consistent results across R-trained models (Yaram, 2016). The application of boosting techniques further elevated accuracy. The utilization of XGB-Boost alongside LR, DT, RF, and SVM produced a precision score of 90%, surpassing KNN's 75% (Urunkar et al., 2022). A similar boosting method, GB, yielded even more compelling results. Employed with comparable ML models, GB achieved a 99% precision score, underscoring the efficacy of boosted models (Dhieb et al., 2019). Another journal paper undertook a comparative study but within a narrower scope, incorporating DT, KNN, NB, and RF models. Notably, RF achieved a 95% accuracy score (Panigrahi and Palkar, 2018). This paper adopted a comparable dataset, employing traditional ML techniques and quantifying outcomes through the Confusion Matrix (Roy and George, 2017). Amidst the evolution of the insurance sector, the realm of artificial intelligence (AI) has also made inroads. A research endeavour considered both ML and AI models, specifically Long Short-Term Memory Recurrent Neural Networks (LSTM RNN). Impressively, the AI model exhibited superior accuracy compared to the ML counterpart (Kini et al., 2022).

3. Aim and Objectives

This research aims a comparative study on different types of fraud detection models such as naïve Bayes, Linear Regression (LR), Support Vector Machines (SVM), Decision Tree (DT), Random Forest (RF), Ridge Classifier (RC), Extreme Gradient Boosting (XG-Boost) and K-Nearest neighbour (KNN). This study aims to identify an optimal model that can efficiently and affordably address insurance fraud.

3.1 Research Objectives

The objectives of this study are:

- To compare and analyse various fraud detection methods and approaches for fraud insurance detection.
- To determine the best Machine Learning (ML) model for fraud detection in automobile insurance by emphasizing the advantages and limitations.
- To evaluate the model’s performance by conducting assessments and evaluations.

4. Significance of the Study

The domain of insurance has assumed an indispensable role, encompassing various sectors in this world. Among these diverse categories, vehicle insurance emerges as one of the most frequently used insurance, inevitably entailing claims that encompass instances of fraudulent activities. The escalating incidence of fraudulent claims has cast a shadow over the insurance landscape, a phenomenon that shows no sign of abating. To comprehend the mechanics underlying these fraudulent occurrences and discern the patterns they exhibit, machine learning (ML) techniques have been strategically employed. By gaining insights into the distinct categories of insurance scams and deciphering their operational intricacies, a prospect arises for the mitigation of fraud claims, particularly those concerning orchestrated collisions, exaggerated claims, and vehicular theft incidents, among others. The pursuit of this objective has fostered a body of research that delves into these dimensions, with the intent of unravelling the concealed patterns underlying such fraud claims, facilitated through the application of diverse models. This study seeks to contribute by undertaking a comparative evaluation of multiple ML algorithms, thereby furnishing an avenue to identify an apt ML model. The essence of this research endeavour lies in its potential to furnish the insurance industry with a potent mechanism for mitigating prospective losses. While individual algorithms have been the focus of previous research endeavours, a comprehensive evaluation encompassing the entire spectrum of algorithms remains relatively unexplored. Through such a comprehensive approach, this study aims to elucidate the operational mechanisms, facilitating model selection based on parameters such as accuracy, precision, and F-1 score.

- More this ML model uses will help in finding fraud in insurance.
- Adaptation of the ML model will help in reducing the loss not only in the vehicle insurance industry but even in other industries.

5. Scope of the study

The scope of this study is to determine the most suitable ML model through a comprehensive comparative analysis of various models. The dataset used in this study is sourced from an open-access data platform. In this study, we are implementing an advanced ML algorithm which consists of many diverse sub-algorithms within it and enables it to provide an efficient output. The dataset is composed of 15000 entries with 32 distinct variables and 1 target variable which holds mixed values of both qualitative and quantitative framework. To make the dataset compatible with the chosen ML algorithm, we transform the dataset into a quantitative framework. Various research has been conducted using diverse ML models. Multiple models are being compared in a single study to identify the optimal one. It is important to note that this particular algorithm may not be suitable for qualitative methods or results that are not binary in nature.

6. Research Methodology

This research study aims to identify the most effective Machine Learning (ML) model through a comparative analysis study. The study will be conducted using PYCARET within a jupyter notebook environment. PYCARET is an accessible ML framework that holds a variety of algorithms. By utilizing this model, the study aims to answer key questions within its scope, specifically comparing multiple models using this approach.

6.1 Data Description

The dataset used in this research study is sourced from KAGGLE open-access data platform. the dataset is a mixed framework which is then transformed into a quantitative framework. The quantitative model is appropriate for this study because we use regression models that require input in a binary format. The data used is based on the collective information of the insurance claimed from the year 1994 to 1996. The data consists of precisely 33 columns and an excess of 15,000 rows.

6.2 Data Processing Method

We are carefully removing any unnecessary rows and columns from these samples to guarantee that the data is thoroughly cleaned. Using the DA method to carefully examine the data by comparing the rows and selecting the most useful information. Then, utilize random sampling to split up the data into training and test sets at a ratio of 80:20. Ensuring a high-quality sample for the ML process heavily relies on cleaning up the data. By effectively removing any excess, unwanted, or null values, you can substantially enhance the accuracy of the output scores.

To rectify an imbalanced target variable in a data sample, the SMOTE method can be used to balance it. This method is highly suitable for processed data that will be utilized with various algorithms. The utilization of an open-source data model in this study eliminates any potential ethical concerns. The algorithm model used in this study has some constraints which will not pose an issue for our research. The algorithmic framework employed in this study is a user-friendly machine learning (ML) model that incorporates numerous distinct ML algorithms within its structure.

The proposed model that will be used in this study

- LR – Linear Regression
- DT - Decision Tree
- RC – Ridge Classifier
- ETC – Extra Trees Classifier
- LDA – Linear Discriminant Analysis
- RFC – Random Forest Classifier

- Light GB – Light Gradient Boosting
- XG-boost – Extreme Gradient Boosting
- GBC – Gradient Boosting Classifier
- ADA – Ada Boost Classifier
- NB – naïve Bayes
- SVM – Support Vector Machine
- KNN – K- Nearest Neighbour
- DC – Dummy Classifier
- QDA – Quadratic Discriminant Analysis

This PYCARET feature obviates the need for manual integration of separate algorithms. Given the comparative nature of this research, this specific algorithmic choice proves advantageous, streamlining the process and economizing the time and resources required for comprehensive analysis.

Performing a comparative research study through manual work on all known ML algorithms can significantly increase time and resource requirements, which may not be the most efficient approach.

6.3 Data Evaluation Method

After the data has been processed by the PYCARET algorithm framework, it is then evaluated by different models.

The models are

- Accuracy
- Precision
- AUC curve
- Recall
- F-1 Score
- Kappa
- MCC – Matthew's Correlation Coefficient

By selecting the appropriate model after the evaluation, the most effective Machine Learning algorithm for Fraudulent prediction can be proposed.

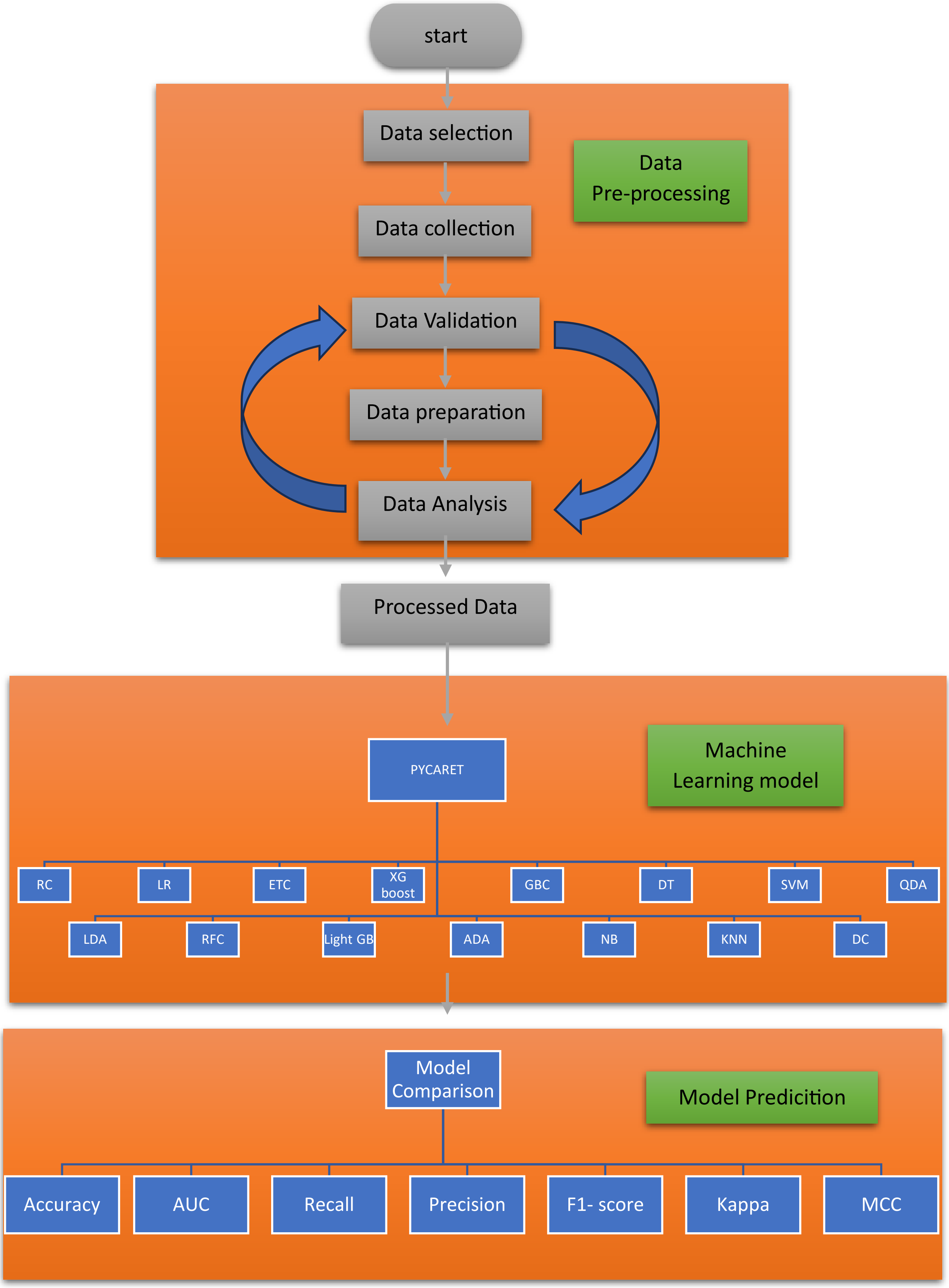

start
Data selection
Data collection
Data Validation
Data preparation
Data Analysis
Data
Pre-processing
Processed Data
PYCARET
Machine
Learning model
RC
LR
ETC
XG
boost
GBC
DT
SVM
QDA
LDA
RFC
Light GB
ADA
NB
KNN
DC
Model
Comparison
Model Predicition
Accuracy
AUC
Recall
Precision
F1- score
Kappa
MCC

Fig. 1

7. Required Resources

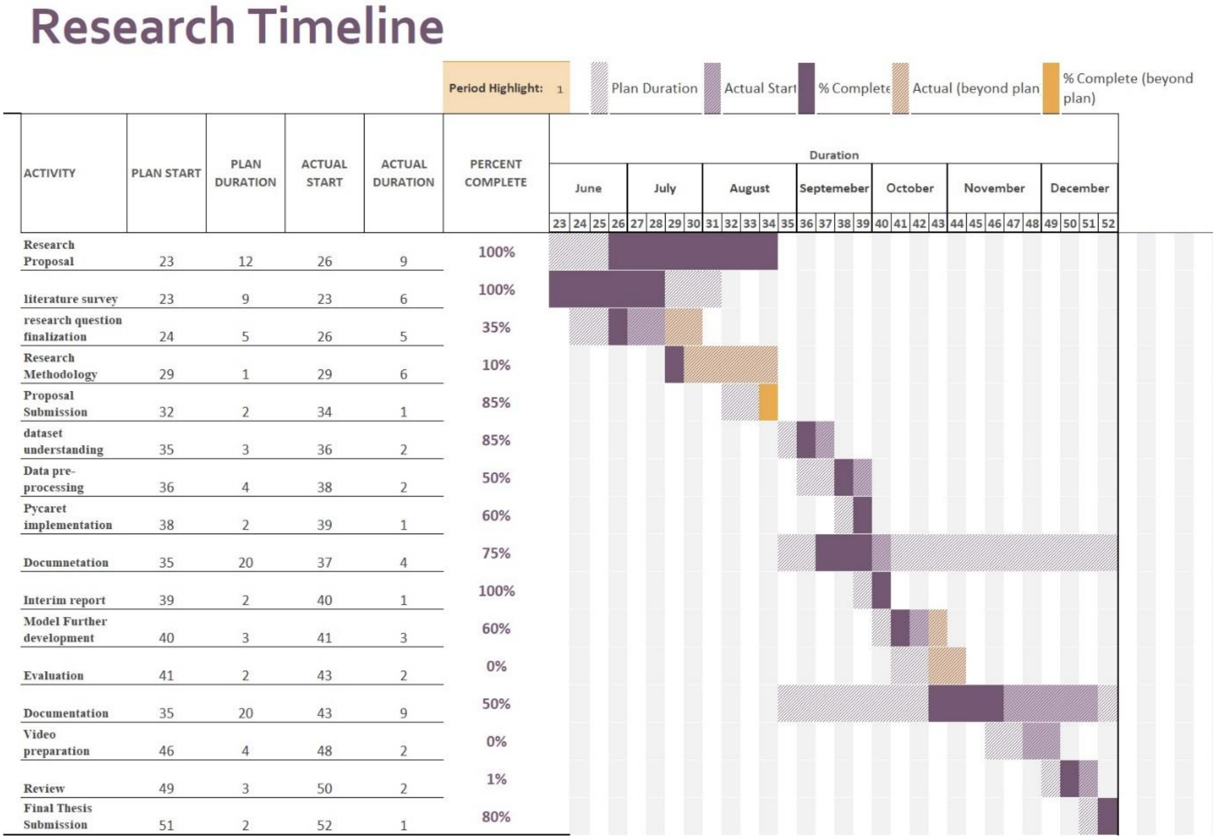
7.1 Hardware Requirement

- Ram – 8 GB
- Processor - 12th Gen Intel(R) Core (TM) i5-1240P 1.70 GHz
- Graphics – intel iRIS xe
- System – 64-bit
- Version – windows 11

7.2 Software Requirement

- Jupyter workbook/Python 3.8
- PYCaret 2.3.10
- Pandas
- Matplotlib
- Seaborn

8. Research Timeline Plan



Research Timeline

ACTIVITY	PLAN START	PLAN DURATION	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
Research Proposal	23	12	26	9	100%
literature survey	23	9	23	6	100%
research question finalization	24	5	26	5	35%
Research Methodology	29	1	29	6	10%
Proposal Submission	32	2	34	1	85%
dataset understanding	35	3	36	2	85%
Data pre-processing	36	4	38	2	50%
Pycaret implementation	38	2	39	1	60%
Documnetation	35	20	37	4	75%
Interim report	39	2	40	1	100%
Model Further development	40	3	41	3	60%
Evaluation	41	2	43	2	0%
Documentation	35	20	43	9	50%
Video preparation	46	4	48	2	0%
Review	49	3	50	2	1%
Final Thesis Submission	51	2	52	1	80%

Fig. 2