Application of Machine Learning Techniques in Insurance Fraud Detection

by

Khutso Mphelo

HONOURS PROJECT
submitted in the fulfilment of the requirements for the degree

Bachelor of Science Honours

in

Applied Mathematics

at the

UNIVERSITY OF JOHANNESBURG

November 2024

SUPERVISOR: Prof C. Harley

Contents

Page

Abstract

The objective of this project is to support insurance companies in their endeavors to counter fraud by improving fraud detection systems. Initially, we will review current research on insurance fraud, analysing successful detection methods and their limitations, examine mathematical techniques utilised prior to machine learning for identifying insurance fraud, alongside exploring data analysis tools for detecting insurance fraud. Building upon this foundation, our focus will shift towards developing a machine learning(ML) model dedicated to flagging suspicious insurance claims. We will develop classification models and assess their performance by computing confusion matrices. Subsequently, we will evaluate these models using diverse performance metrics such as accuracy, precision, recall, F1 score, and AUC (Area Under the Curve) curve analysis. This model will leverage historical data to identify patterns and irregularities associated with fraudulent activities. Our dataset for this project will be "Vehicle claim Fraud Detection" (Bansal, 2021).

Keywords: Insurance Fraud, Machine Learning, Data Analysis, Fraud Detection, Underwriting Fraud, Claim Fraud, Fraudulent Behavor.

1 Introduction

Insurance fraud poses a significant challenge to insurance companies at a global scale, not just because it costs them money but also because it undermines trust between the insurer and the insured. Insurance fraud involves intentionally deceiving insurance providers for financial gain, typically by individuals filing a claim seeking to acquire advantages or benefits to which they have no rightful entitlement. There are three commonly encountered functional classifications of insurance fraud which are internal vs external, underwriting vs claim, and soft vs hard (Viaene and Dedene 2004). Internal fraud schemes can involve asset misappropriation, financial statement fraud, or corruption whereas external fraud involves individuals or entities outside the insurance company trying to gain an unfair advantage. Fraudulent behavior can also occur during both the underwriting phase and when submitting a claim (Viaene and Dedene 2004). According to the PricewaterhouseCoopers International Limited global economic crime survey, government entities around the globe that suffered from economic crime reported that 57% perpetrators were internal while only 37% were external (PricewaterhouseCoopers International Limited 2011). This internal threat is significantly higher in the public sector than in most other sectors. Underwriting fraud entails engaging in deceptive practices during the application process for insurance coverage. These practices encompass misrepresentation on the application form to secure more favorable premiums, withholding information regarding existing insurance policies, and attempting to obtain coverage for risks that may not be pertinent at the time of application. On the other hand, claim fraud pertains to fraudulent behaviors undertaken when submitting an insurance claim. Such actions may involve falsifying information in order to maximise the compensation received from the insurance provider (Viaene and Dedene 2004). Soft fraud, also referred to as opportunistic fraud, happens when policyholders exaggerate what would otherwise be a legitimate claim, this signifies that they are making false statements or inflating the value of their losses while hard fraud happens when someone portrays a non-insured event to appear as if it is covered by their insurance policy (Derrig and Richard 2002). To tackle the problems head-on, insurers are always on the lookout for better fraud detection systems. This project dives into that challenge, aiming to improve how we spot fraud by blending traditional mathematical techniques with modern data analysis tools, all leading to the creation of a machine learning (ML) model.

We begin by delving into the present landscape of fraud detection methods. Then, we will revisit the mathematical approaches predating the popularity of machine learning. Finally, we'll explore modern data tools to evaluate their potential in detecting insurance fraud. However, our endeavors extend beyond that point since the primary objective of this project is to develop a machine learning model tailored to uncover suspicious insurance claims. For the development of our model, we employed a dataset made available by Oracle on the Kaggle platform, curated by Shivam Bansal, specifically the "Vehicle claim Fraud Detection" (Bansal, 2021). The data provides information about an American insurance company. It provides detailed information about occurrences such as accidents, the timing of claim submissions, as well as specific information about individuals and vehicles affected. This enables us to understand the frequency of accidents, the process of handling claims, and the demographics of the insured individuals. Moreover, we can gather insights into vehicle characteristics and the types of insurance policies held by customers. Certain aspects of the data assists the company in detecting potential fraudulent behavior or dishonesty related to claims. So we will use this dataset to design a classifier that is able to identify fraudulent behaviour, and additionally we will extract insights regarding behaviors of people who are involved in those fraudulent activities. Analysing fraudster behaviour can help us to understand the patterns, techniques used by fraudsters. We can then develop more effective prevention and detection strategies. Additionally, such analysis can enhance on identifying vulnerabilities in current systems and processes, allowing for targeted improvements to mitigate the risk of fraud.

2 Literature review

A review of existing literature was conducted to explore the methodologies utilised in detecting fraud in the insurance industry. Historically, insurance fraud detection has relied heavily on human expert audits and inspection techniques (Artís et al., 2002; Dionne et al., 2009; Nian et al., 2016). These techniques involve a thorough manual inspection review of claims to identify discrepancies, thereafter field visits are often conducted to confirm the information submitted with the claims. These visits are essential for obtaining tangible evidence and for ensuring that the details provided reflect the actual circumstances, and interviews are conducted to determine the legitimacy of claims. These discussions are aimed at gathering firsthand accounts and insights, which can be instrumental in painting a complete picture of the situation.

However, the landscape of insurance fraud detection is rapidly evolving, driven by technological advancements and the expanding scope of business operations. These traditional methods are now facing challenges in keeping pace with the sophistication of modern fraudulent activities (Kemp 2010). Based on the findings of Krambia-Kapardis (2002) which focused on the underlying issues faced by traditional methods says that audits are typically based on sampling and testing a portion of transactions or accounts. However, fraudsters can exploit this approach by hiding their activities within the larger population of legitimate transactions, which makes it difficult for auditors to detect anomalies. Kapardis further extended the discussion on auditors' lack of necessary skills to detect fraud, in additionally to that the time pressure they face and the built in conflict investigating with upper management who indirectly hired them further complicate the situation and make it more difficult for them to do their job to full capacity. The emergence of digital technologies and big data analytics has transformed the landscape of fraud detection. Insurers are increasingly turning to advanced analytical techniques, such as predictive modeling, machine learning, and artificial intelligence, to detect and prevent fraudulent activities (Baesens et al., 2015). These technologies offer the promise of greater accuracy and efficiency in identifying suspicious patterns and anomalies in vast amounts of insurance data. Despite the promise of technological solutions, fraudsters are continuously adapting and innovating their tactics to evade detection. The dynamic nature of fraud requires a proactive and adaptive approach from insurers. Fixed algorithmic criteria, while effective to some extent, may fall short in capturing the nuanced and evolving strategies employed by fraudsters since these algorithms are typically utilised in fraud detection systems to flag potential fraudulent activities based on predefined rules and patterns (Nguyen and Perez 2020) and the challenge arises when fraudsters adapt and evolve their strategies, leading to discrepancies that may not align with the fixed criteria, therefore potentially evading techniques.

In response to swift macro environmental shifts, the insurance sector needed to incorporate rapid statistical models and machine learning in to their fraud detection systems (Gomes et al., 2021). The integration of statistical and machine learning models into fraud detection systems offers several advantages. These models can analyse large volumes of data in real-time, enabling insurers to detect fraudulent activity more quickly and accurately. Additionally, by continuously learning from new data, these models can adapt to evolving fraud schemes and improve their effectiveness over time. A study by the statistical analysis system found that in the United States, 75% of all insurers had integrated automated systems for fraud detection by 2016 (Gomes et al., 2021). In Debenaar, et al., (2012) the application of both supervised and unsupervised learning algorithms for claim fraud detection were explored. Isolation Forests, an unsupervised learning algorithm, were used for identifying anomalies (outliers) in a dataset. These forests create decision trees by randomly selecting features to isolate anomalies. They are particularly useful when labeled data is scarce. On the other hand, XGBoost, a powerful supervised machine learning algorithm, was compared with other methods, including neural-network-based and clustering-based fraud detection algorithms. The dataset used in this exercise comprises of 7,750 automobile insurance claims filed between January 2020 and April 2021. These claims pertain to damages incurred by the policyholders' vehicles due to collisions. Debenaar then found that unsupervised learning (isolation forests) is valuable for insurance companies, especially when labeled data is unavailable. However, even with limited labeled data, the supervised learning approach (XGBoost) performed strongly well hand in hand with unsupervised methods. Both approaches detected fraud cases that the existing mechanisms had missed, and the detected cases sometimes differed. In contrast Lakshmi and Kavila (2018) focused on three machine learning algorithms to detect fraud in credit card systems namely logistic regression, decision tree, and random forest classifier. The random forest classifier emerged as the superior method compared to logistic regression and decision tree, based on the performance metrics of sensitivity, specificity, accuracy, and error rate. Abbasi et al., (2022) argued in their paper that the ensemble learning approach developed by Hanson and Salmon in 1990 tends to yield better accuracy in most cases. This is because it integrates various algorithms to achieve higher accuracy and improve overall performance compared to using individual algorithms alone. Additionally, they highlighted that the ensemble approach reduces latency, enhances robustness in the presence of noise, and is more effective in handling datasets with class imbalance issues.

The table below illustrates some of the model classes and their corresponding modelling techniques used in fraud detection post 2016. The machine learning models that are commonly used to detect fraud are predominantly supervised classification models, which include techniques like Logistic regression, Random Forest, Naive Bayes, and Support Vector Machines (Debener et al., 2023).

Table 1: Overview of Model Classes and Methodologies

Model class	**Modelling methodologies**	**Description**
Clustering	K-means Nearest neighbors	Unsupervised learning methods that group data points into clusters based on similarity.
Classification	Logistic regression Random forest Naive Bayes Support Vector Machine (SVM)	Supervised learning techniques for categorising data into predefined classes or labels.
Regression	Linear regression Polynomial regression Neural networks	Supervised learning methods for predicting continuous outcomes or estimating probabilities.
Outlier detection	Isolation forest Gaussian mixture models	Techniques for identifying anomalous data points that deviate significantly from the norm.

3 Project Analysis

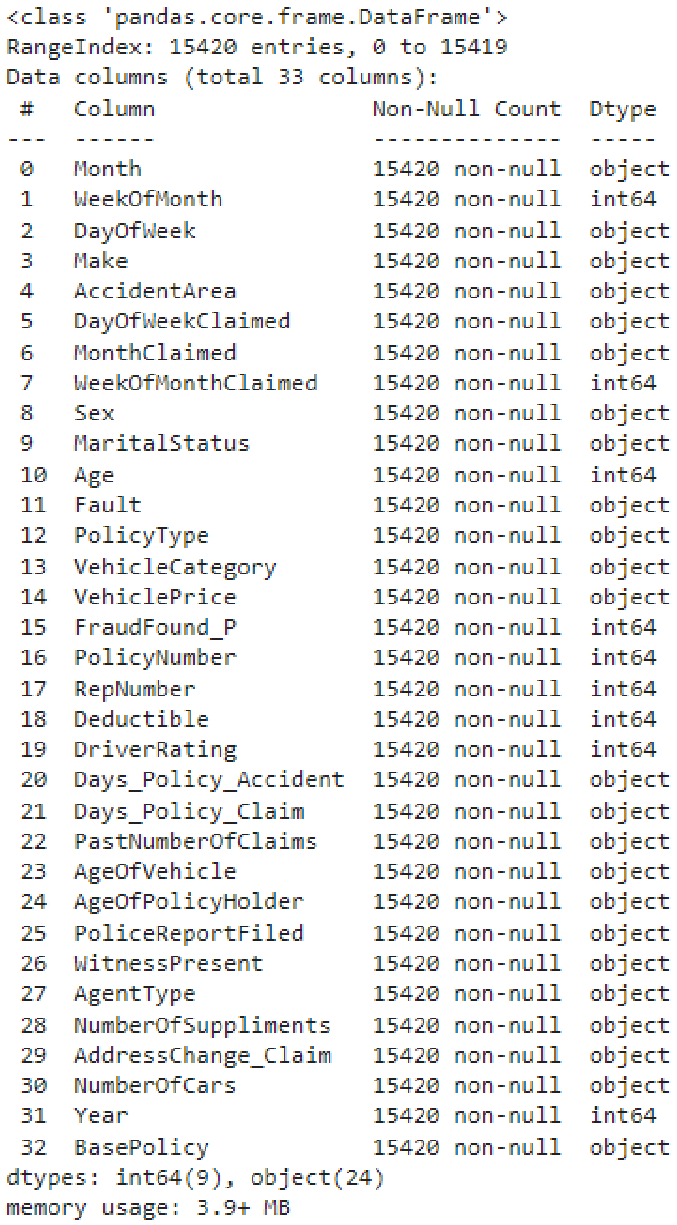
3.1 Data Overview

In the data analysis part of our project, we carefully looked at the insurance claims data (Bansal, 2021) to find patterns, unusual cases, and connections that might show fraud. This helped us get ready to build a machine learning model that can detect fraud. As discussed the dataset used in this exercise comprises 15420 automobile insurance claims filed between January 2020 and April 2021, and is titled 'Vehicle Claim Fraud Detection', provided by Bansal in 2021. The features included in the dataset are:

Feature	**Description**
`Month`	The month when the incident occurred
`WeekOfMonth`	The week of the month when the incident occurred
`DayOfWeek`	The day of the week when the incident occurred
`Make`	The make of the vehicle involved
`AccidentArea`	The area where the accident occurred
`DayOfWeekClaimed`	The day of the week when the claim was made
`MonthClaimed`	The month when the claim was made
`WeekOfMonthClaimed`	The week of the month when the claim was made
`Sex`	The gender of the policyholder
`MaritalStatus`	The marital status of the policyholder
`Age`	The age of the policyholder
`Fault`	Indicates whether the fault was of the policyholder
`PolicyType`	The type of insurance policy
`VehicleCategory`	The category of the vehicle involved
`VehiclePrice`	The price range of the vehicle
`PolicyNumber`	The unique number of the insurance policy
`RepNumber`	The number of the insurance representative
`Deductible`	The deductible amount on the policy
`DriverRating`	The rating of the driver by the insurer
`Days_Policy_Accident`	Days between the start of the policy and the accident
`Days_Policy_Claim`	Days between the start of the policy and the claim
`PastNumberOfClaims`	The number of claims made in the past
`AgeOfVehicle`	The age of the vehicle involved
`AgeOfPolicyHolder`	The age of the policyholder
`PoliceReportFiled`	Indicates if a police report was filed
`WitnessPresent`	Indicates if a witness was present
`AddressChange_Claim`	Indicates if there was an address change at the time of the claim
`NumberOfSuppliments`	The number of supplements added to the claim
`AgentType`	The type of agent handling the claim
`NumberOfCars`	The number of cars covered under the policy
`Year`	The year of the incident
`BasePolicy`	The base policy type
`FraudFound_P`	Target variable indicating if fraud was found in the claim or not

Table 2: Features in the dataset and their descriptions

Among these, `FraudFound_P` is our target variable, indicating whether a claim was found to be fraudulent or not. A full detailed view of the data is shown below:



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15420 entries, 0 to 15419
Data columns (total 33 columns):
 #   Column                Non-Null Count  Dtype 
---  ------                --------------  ----- 
 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)
memory usage: 3.9+ MB
```

Figure 1: Information about the data.

Out of the 33 features, only 9 are numerical features, and the remaining 24 are categorical features thus a thorough investigation into both during Exploratory Data Analysis (EDA) is mandated. The aim of EDA was to look at how these aspects distributed themselves and related with one another so that any patterns which might point towards fraudulent claims could be unveiled. This entailed checking how often and how much each categorical feature occurred via bar graphs and cross-tabulations together with looking at the statistical characteristics for instance histograms, box plots, and correlation matrices for numerical variables. The resulting analysis resulted in some important insights which guided selection of features and modeling methodologies. Given the importance of accurately detecting fraudulent claims, it is essential to understand the distribution of the target variable. In the next subsection we will consider looking at the distribution of the target variable.

3.2 Imbalanced Target Variable

In this subsection, we explore the distribution of our target variable, FraudFound_P. One of the key challenges in building a robust fraud detection model is dealing with the imbalance in the target variable. In most real-world datasets, fraudulent claims constitute a small fraction of the total claims, leading to a class imbalance. This imbalance can significantly impact the model's performance, as the model may become biased towards predicting the majority class (non-fraudulent claims).

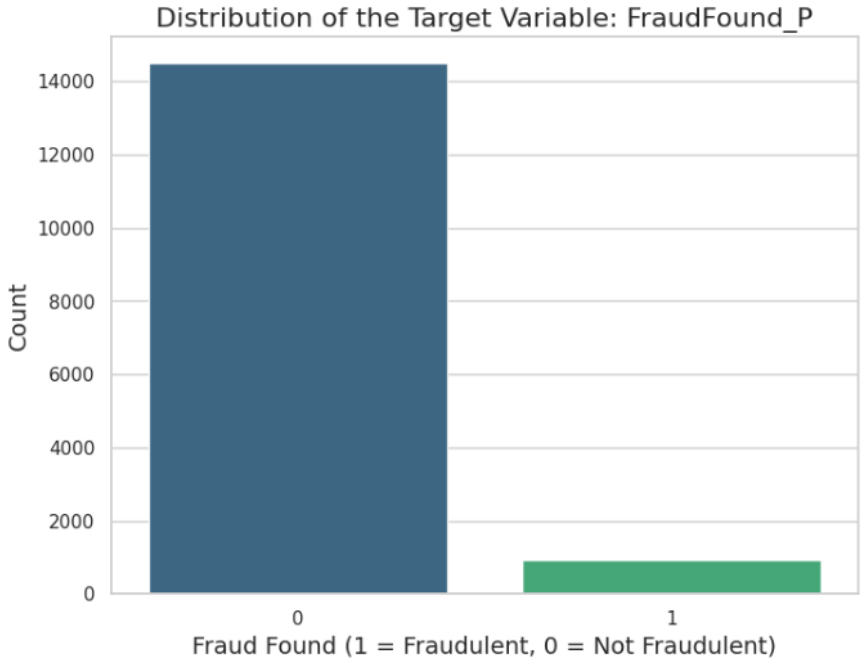


Figure 2: Distribution of the target variable.

As observed from the target variable, FraudFound_P, reveals an uneven distribution within the dataset, showing a notably higher number of non-fraudulent claims compared to fraudulent ones. In fact, only 6 percent of the records are fraudulent claims, indicating that fraud represents a small portion of the overall dataset. This significant imbalance creates considerable challenges for fraud detection, as instances of fraud are generally rare compared to non-fraudulent cases. Therefore, creating models that can accurately predict and effectively identify fraudulent activities demands careful consideration and the use of specialised techniques.

It is essential to address these discrepancies to improve the performance and reliability of fraud detection models. To counteract the adverse effects of this class imbalance on our analysis, various strategies can be utilised, including the Synthetic Minority Over-sampling Technique (SMOTE) and under-sampling. SMOTE is a technique that generates synthetic examples of the minority class, in this case, fraudulent claims, which helps create a more balanced and representative dataset. This method aids in preventing the model from favoring the majority class and enhances its capability to detect fraud.

Conversely, under-sampling involves intentionally reducing the number of majority-class instances to prevent the dataset from being biased towards non-fraudulent claims. By thoughtfully selecting and decreasing the number of non-fraudulent instances, the dataset can achieve a more balanced distribution, which is crucial for training effective fraud detection models. Both SMOTE and under-sampling are useful techniques that, when applied correctly, can greatly enhance the model's ability to recognise fraudulent activities, ultimately resulting in more robust and dependable fraud detection systems. We will implement an appropriate strategy in a later section.

3.3 Exploratory Data Analysis

This section provides an exploratory data analysis (EDA) of the dataset, with the goal of revealing important patterns, trends, and relationships within the data. By performing EDA, we can obtain preliminary insights that will shape further analysis and model development. The analysis includes both visual and statistical assessments of the data, concentrating on key variables that could impact fraudulent insurance claims. We will start by analysing the timing of fraudulent cases.

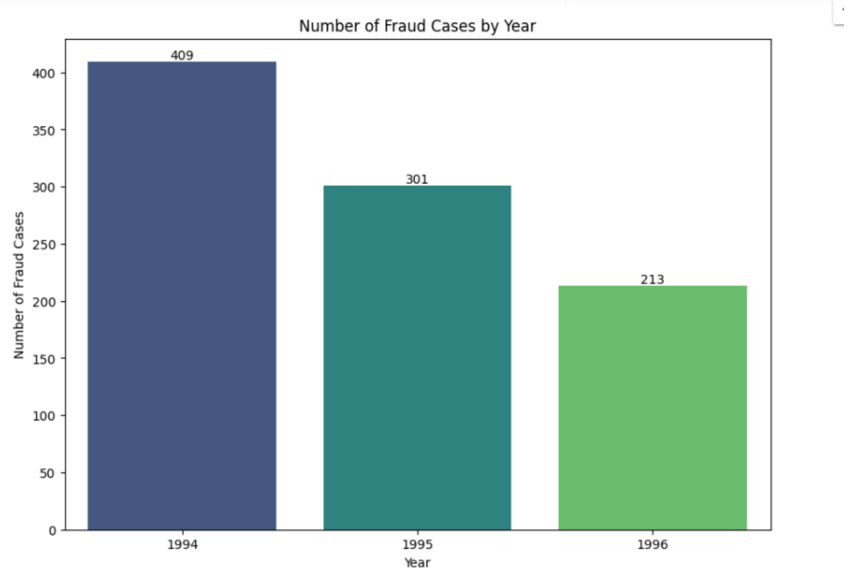
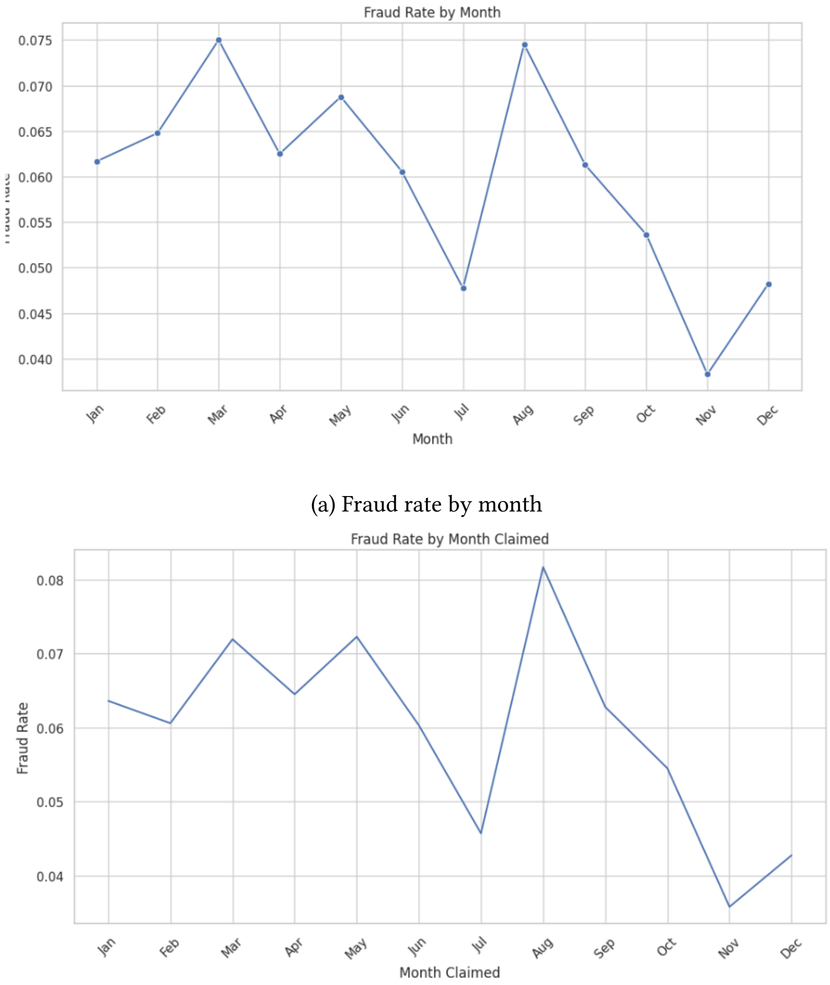


Figure 3: Fraud cases by year

Analysing the timing of fraudulent cases uncovers significant patterns. In 1994, there was a peak in fraud cases with a total of 409 reported incidents. This peak could indicate a notable rise in fraudulent activities in that year or advancements in fraud detection techniques resulting in increased detection rates. The decrease in fraud cases to 301 in 1995 could be due to the introduction of new fraud prevention methods or a change in fraudulent activities leading to fewer detected cases. In 1996, the downward trend of fraud cases continued, with 213 cases reported. This continued decrease could indicate additional progress in identifying and preventing fraud, or it may suggest a shift in fraudulent techniques that have made detection more challenging. Examining these trends over a period of time will offer beneficial understandings on the success of fraud prevention measures and the changing patterns of deceptive behaviors.



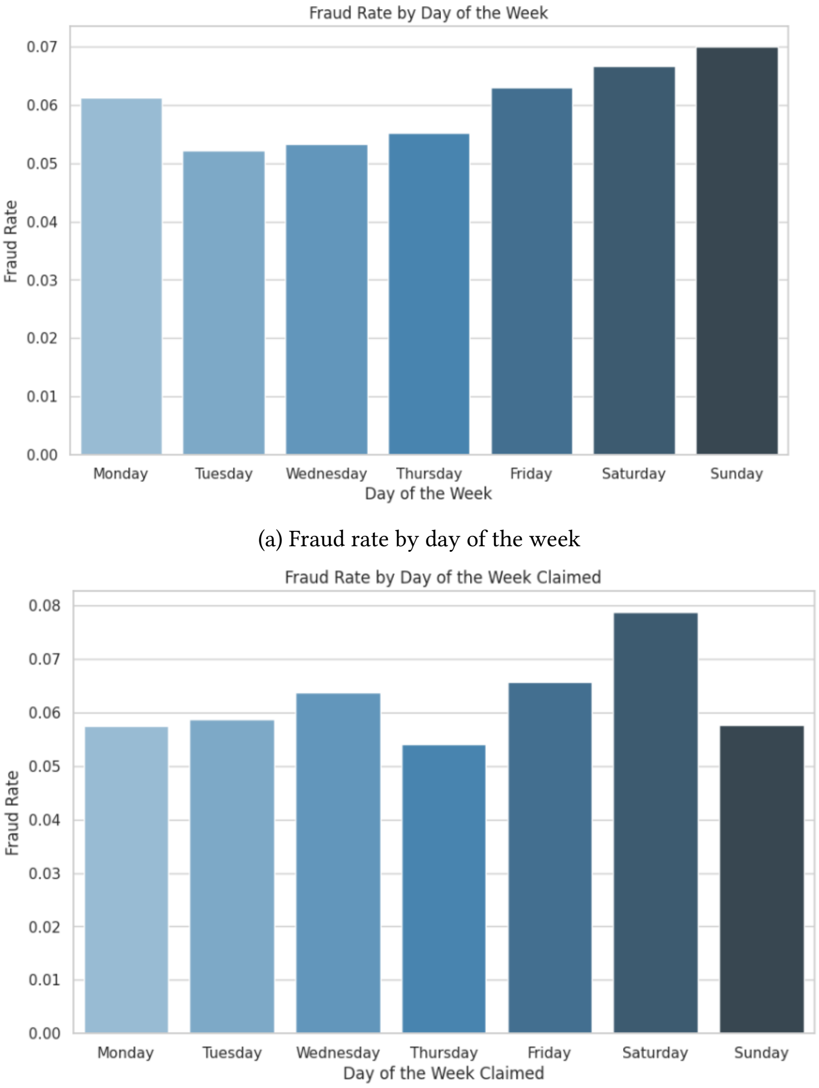
(a) Fraud rate by month

(b) Fraud rate by month claimed

Figure 4: Fraud by month and month claimed

The incidence of fraud reflects strong seasonal variation according to the month of the year in which the incident occurred and the month of claim submission, as is seen from figure 4(a) and 4(b). For the analysis of cases of fraud between months, a strong peak in March and August may be seen where the rate is approximately 0.07, indicating a period with higher fraudulent tendency. May also shows high fraud rates, standing just below 0.07. These months, particularly March and August, might be the crucial months when claims can easily be fraudulent, maybe even with the influence of economic conditions or seasonal financial burdens. On the other hand the lowest rate of fraud occurs in November, when it goes below 0.04, this points out that towards the end of the year, fraudulent activities become low.

The persistent decrease observed prior to November could indicate the efficacy of specific fraud prevention strategies or a typical reduction in claim submissions as the calendar year comes to a close. Nonetheless, subsequent to this lowest point, there is an increase in the rate during December, suggesting that either the implemented preventive strategies are diminishing in their effectiveness or that particular conditions associated with the holiday period may be fostering fraudulent activity. Notably, the fraud rate difference is observable not just through the calendar year, but also in monitoring fraud rates by the month in which claims are submitted. Figure 4(b) shows fraud rates sorted by the claim submission month behaves similarly, peaking in August, and further solidifies this time as the most crucial month for fraudulent claims. This would insinuate that claims in August could be more fraudulent, possibly due to the opportunistic nature of mid-year, related to financial burdens, or any other socio-economic factor. The next two months, March and May, both have fraud rates above 0.07, showing that they, too, are months that tend to see greater claims. On the other hand, November is identified as the month when fraudulent claims are at their lowest, which therefore confirms the trend indicated by the annual data. Building on the results obtained from monthly seasonality, we would now proceed to a more granular level of fraud rate analysis using observations that distinguish between the week of the year and the week in which the claim was submitted.

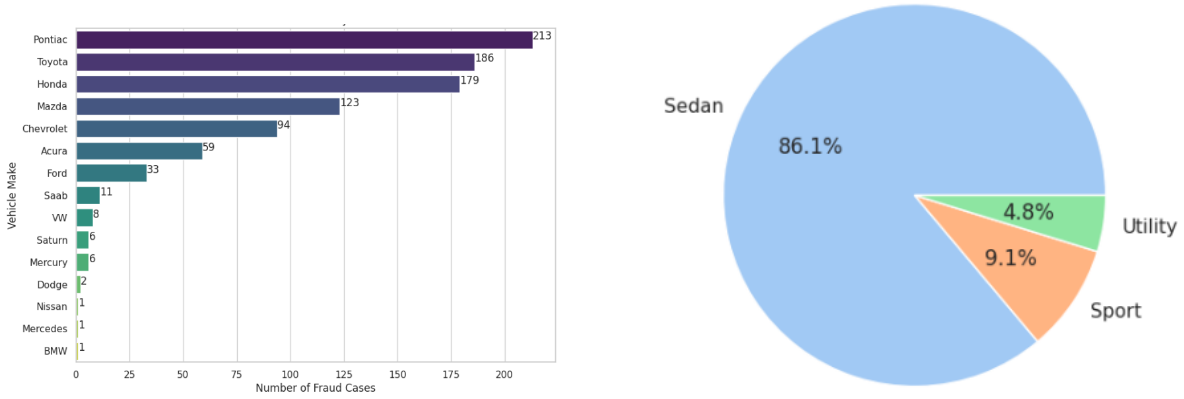


(a) Fraud rate by day of the week

(b) Fraud rate by day of the week claimed

Figure 5: Fraud rate by day of the week and day of the week claimed claimed

The analysis of fraudulent insurance claims reveals distinct and insightful patterns in the frequency of fraud occurrences throughout the week. Figure 5(a) indicates that Saturday and Sunday experience the highest rate of fraudulent activities, while Tuesday has the least, suggesting an end of week peak in dishonest behavior. This trend could be attributed to various factors, such as specific operational practices within insurance companies that might be exploited by fraudsters. Conversely, when examining the days on which claims are filed shown on figure 5(b), Saturdays emerge as the most common day for reporting fraudulent claims, with Thursday being the least common. This discrepancy between the day of occurrence and the day of reporting could indicate strategic timing by fraudsters to avoid detection, possibly leveraging weekends when oversight might be reduced. We will now focus on exploring the distribution of fraud cases by vehicle make and category. Figure 6(a) and 6(b) below provide insights into the number of fraud cases by make and the percentage of fraud cases across different vehicle categories.

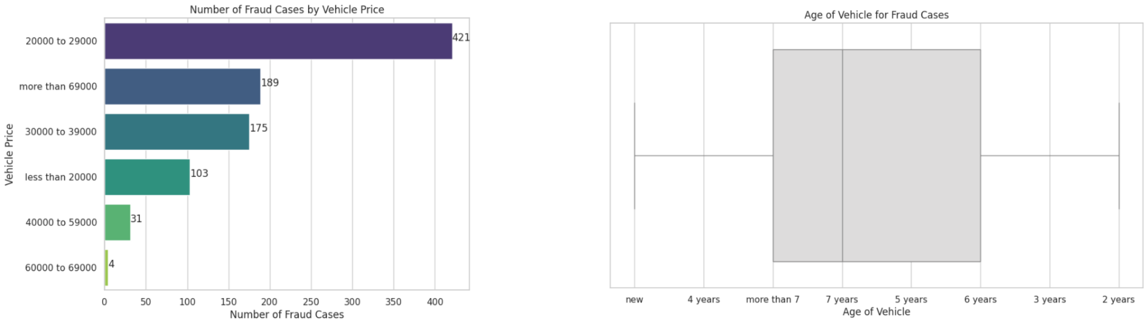


(a) Number of fraud cases by make

(b) Percentage of fraud cases by vehicle category

Figure 6: Fraud by vehicle make and vehicle category

The bar graph on figure 6(a) shows that Pontiac is in first place with 213 instances of fraud, while Toyota and Honda come in second and third with 186 and 179 cases, respectively. Mazda and Chevrolet also have notable figures, with 123 and 94 instances, respectively. Acura, Ford, Saab, VW, Mercury, Saturn, Dodge, Nissan, and BMW have a lower number of cases, with BMW having only 1 case and the rest ranging from 59 to just 1. This breakdown shows that some car brands are more commonly linked to fraud, possibly because of factors like popularity, market share, or specific weaknesses. Furthermore, the pie chart on figure 6(b) shows that most fraudulent incidents are linked to Sedans, making up an astonishing 86.1% of all cases. Utility vehicles account for 9.1% of the total, with Sport vehicles making up the smallest share at 4.8%. This notable difference implies that Sedans, which are a popular type of vehicle, are at a higher risk for fraud. The prevalent use and accessibility of Sedans could be linked to the high number of fraud instances associated with them, making them a primary focus for fraudulent behavior. By bringing together these perspectives, it is clear that the type and brand of a car are both important factors in the occurrence of fraud. The occurrence of fraudulent activities across various vehicle brands and types provides important information for the analysis of automotive fraud and awareness of consumers. We now pose the following questions, how are fraud cases distributed based on vehicle price? Moreover what percentage of fraud cases is associated with the age of the vehicle?



(a) Number of fraud cases by vehicle price

(b) Percentage of fraud cases by age of vehicle

Figure 7: Fraud by vehicle price and vehicle age

Fraud cases mostly occurred in cars that cost between 20 000 and 30 000 dollars, with 421 cases as a result. Next on the list are cars worth more than 69 000 dollars with 189 cases. There is a marked decrease in fraudulent activity for other price ranges, with the lowest number of cases occurring in vehicles priced between 60 000 dollars and 69 000 dollars. On the other hand the distribution of fraud cases by vehicle age reveals an interesting trend. Most fraud cases are found in vehicles that are about 5 to 7 years old. This age group appears particularly susceptible to fraudulent activities, likely due to higher repair costs and the fact that these vehicles are approaching the end of their useful life. The financial strain of maintaining these vehicles may lead policyholders to exaggerate claims or engage in fraudulent behavior.

Interestingly, there are outliers in the data, with fraud cases involving both new vehicles and those older than 7 years. Although these instances are less frequent, they show that fraudulent activities can occur across various vehicle ages, indicating that no specific age group is completely safe from the risk of fraud. In fact, fraud has been noted in vehicles as young as 2 years and as old as over 7 years, highlighting that fraudulent claims can happen at any point in a vehicle's lifespan. Several factors may account for these trends. First, vehicles aged 5 to 7 years often require more extensive repairs, which could lead to inflated claims or other fraudulent actions. Second, as vehicles age and their market value decreases, policyholders might resort to fraud to recover perceived financial losses. Together, these factors contribute to an increased risk of fraud within this age range.

Now, moving away from vehicle age, we will examine fraud patterns based on the age of the policy holder.

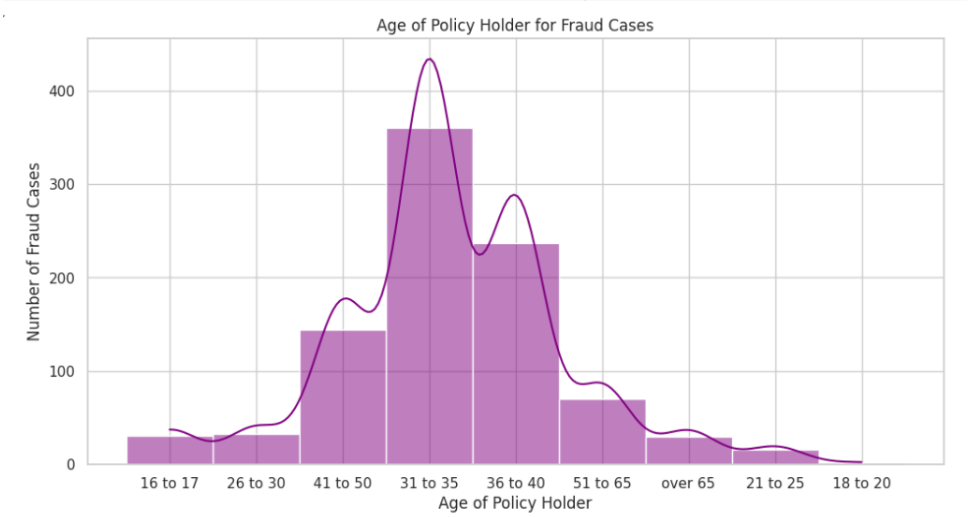
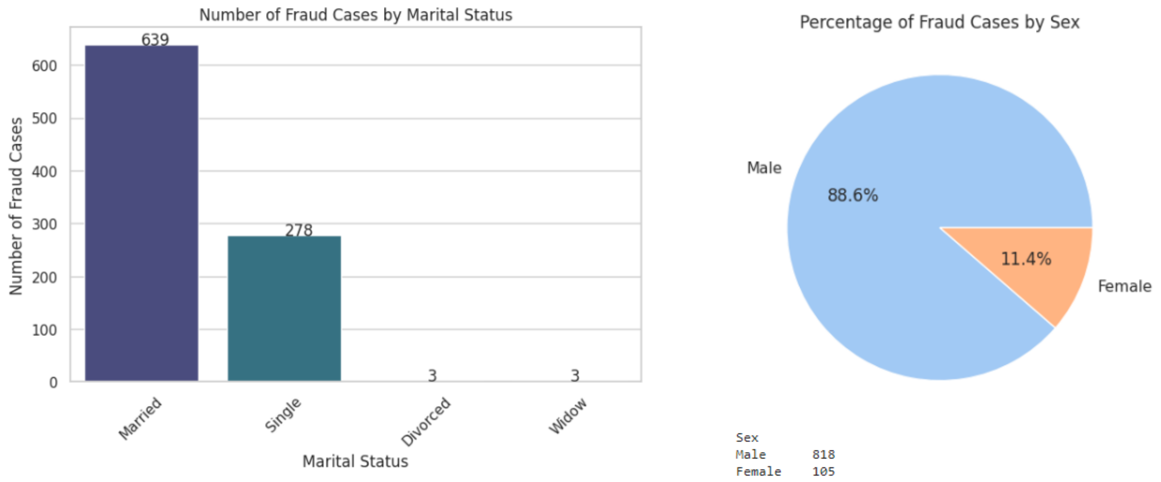


Figure 8: Fraud cases age of policy holder

Figure 8 gives information on the age distribution of fraudulent insurance claims, it is easy to see the age ranges that have most of the fraudulent claims. The most common fraudulent policy holders are middle aged, specifically in the 31-35 and 41-50 age group. These are the spikes in the distribution, meaning there is an especially high chance for fraudulent behavior in these age ranges. This could be attributed to the fact that many people in these age groups are likely to have families to support and face more financial pressures, potentially leading to a greater number of bills and financial responsibilities.

On the other hand, younger insured's, particularly the 18-20 and 21-25 groups, show a very low number of fraud cases. That might mean that younger people are either more honest or just don't have as many insurance policies so they don't have as much opportunity to commit fraud. Also older insured, especially over 65, seem to have less fraud, that could be due to conservative behavior or just less contact with the insurance system. It seems when people get older they are less likely to participate in fraudulent activities. Another thing to note is that there are moderate fraud cases in the age groups of 26-30 and 36-40. They don't have as high a frequency as the

31-35 or 41-50 age groups, but they still constitute a large percentage of fraud cases, which indicates that the risk of fraud is still quite prevalent during these stages of life. The kernel density estimation(KDE) line evens it out even more and the spikes at 31-35 and 41-50 are further supported. The KDE also shows a steady downward trend of fraud cases after the age of 50, which adds more evidences that elderly policy holders are less likely to commit fraud.



```
Sex
Male      818
Female    105
```

(a) Number of fraud cases by marital status

(b) Percentage of fraud cases by gender

Figure 9: Fraud by marital status and gender

Figure 9(a) under consideration is a simple bar chart named ‘Number of Fraud Cases by Marital Status’ depicting fraud cases based on marital status. The data indicates that most of the fraud cases were committed by married individuals with married individuals being able to commit approximately 639 cases. This figure leads one to believe that married people are often lured into fraud probably due to some financial challenges or socio-economic reasons. Single individuals account for around 278 cases, which is a significant case size but still a smaller fraction of the total number of fraud cases in comparison to married individuals. However, divorced and widowed individuals are engaged in a much lesser number of cases indicating lesser chances of committing fraudulent activities or representing lower populations within the population. This bar chart explains the nexus between the marital status of the individual and the individual’s engagement in insurance fraud and will assist insurance companies in coming up with preventive measures targeting specific demographic groups against the vices.

Gender is the basis for figure 9(b) which is a pie chart named "Percentage of Fraud Cases by Sex” whereby fraud cases are grouped based on this category. The research reveals that men are the predominant offenders of all occurrences of 88.6 percent of incidences. This significant majority implies that as compared to women, men are more likely to commit insurance fraud compared to women. The results are not surprising because research has repeatedly indicated that women typically maintain higher ethical standards and are less prone to criminal behaviour compared to men (Wang et al., 2022). It can also be hypothesised that men, traditionally seen as the primary earners in families, may feel increased financial pressure to provide for their households, which could contribute to a higher likelihood of engaging in fraudulent behavior.

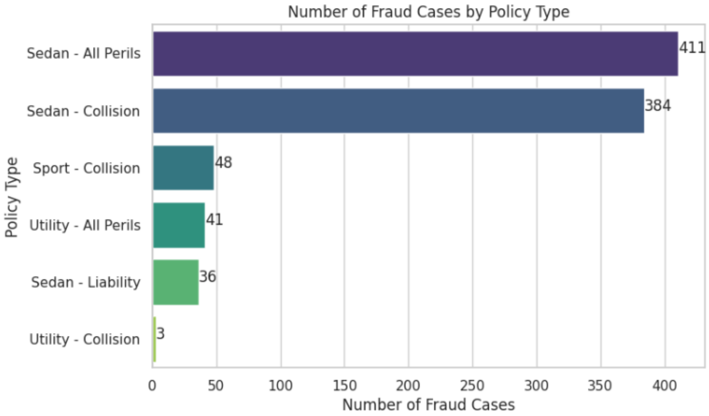


Figure 10: Number of fraud cases by policy type

Figure 10 shows the number of fraud cases based on policy types. The bar chart depicts that some of the categories of policies are more vulnerable than the other policies. In particular, it can be observed that the policy type of “Sedan – All Perils” tally the most cases of fraudulent activity with 411 cases being recorded. This is closely followed by policy cases of “Sedan - Collision” with cases of 384. It is rather evident that these two classes together account for the most proportion of fraudulent cases, which points to a problem of risk in policies that cover sedan vehicles.

The difference between the first two usages of the policies within the Table and the rest is rather telling. The next most common fraud type was noted as “Sport – Collision” which had 48 cases, "Utility – All Perils", which had 41 cases, and “Sedan – Liability.” which recorded 36 cases. The policy type which attracted the least number of fraud incidents was the “Utility – Collision” where only 3 cases were reported.

These results suggest that sedan vehicles present the highest risk towards fraudulent claims under "All Perils" and "Collision" policies. This observed trend could be for a number of factors, including the high usage rate of sedan vehicles, higher frequency of claims under comprehensive coverage options such as "All Perils", or the fact that fraud opportunities could be greater given the more inclusive nature of these policies. From an insurer’s policy and risk management perspective, there will be a need to reconsider underwriting guidelines and processes that review claims submitted for sedans. The frequency of fraudulent cases is on the higher side, and there is likely to be additional or more stringent mechanisms. Specific fraud prevention programs should also be educative and include effective procedures for investigation with a view to maintaining the fraud rate at low levels for such policies.

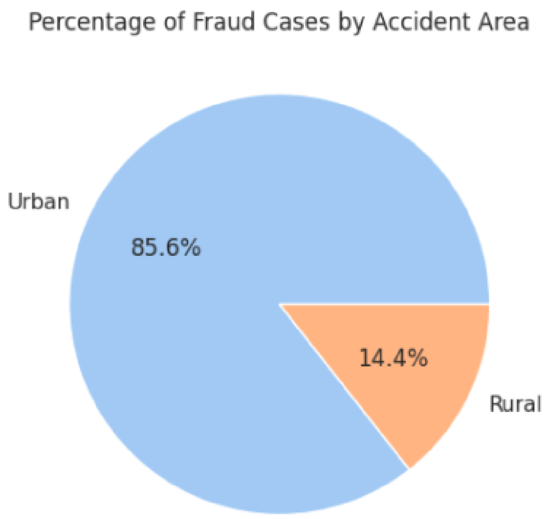


Figure 11: Percentage of fraud cases by accident area

The analysis on figure 11 indicates that fraudulent claims are considerably more prevalent in urban areas than rural regions. Approximately 85.6% of fraud cases occur in urban settings: this underscores pronounced dominance of fraud incidents within these environments. However, this pattern implies (that) fraud is more common in urban locales, potentially because various demographic and economic factors inherent to these areas. Higher population densities, coupled with more frequent vehicular activity in cities, enhance probability of accidents, thus creating increased opportunities for individuals to submit fraudulent claims. Although the data is compelling, one must take into account the context; urban settings present unique conditions that might contribute to these trends.

In contrast, only 14.4% of fraud cases are reported in rural areas (which indicates a significantly lower representation of fraudulent activities). This disparity can be attributed to various factors that are unique to rural settings: lower population density, fewer vehicles on the road and distinct reporting or claim-filing practices. However, because there are fewer accidents in these regions, opportunities (or incentives) for engaging in fraudulent behavior may be limited. Although rural communities often have closer social networks, this makes it more challenging for individuals to file fraudulent claims without attracting suspicion or facing potential repercussions.

The higher incidence of fraud in urban areas could be attributed to multiple factors. One possibility is that the sheer volume of traffic and higher population density in cities naturally result in more accidents, thereby increasing the number of claims filed. Among these claims, there may be a greater proportion of fraudulent cases, as claimants attempt to exploit the higher frequency of incidents. Furthermore, the infrastructure and resources available for detecting and reporting fraud may be more robust in urban areas, which could lead to higher detection and reporting rates compared to rural regions. Urban areas typically have better access to technology and investigative resources, enabling more effective identification of fraudulent activities.

Economic factors might also play a role in this pattern. Urban areas often have a higher concentration of insured vehicles and greater economic activity, potentially influencing the volume of claims filed and the likelihood of fraud. Given the higher economic stakes involved, individuals in urban environments may be more motivated to commit fraud, especially if they perceive an opportunity to gain financial benefits or avoid financial losses. Overall, the discrepancy between urban and rural fraud rates suggests that geographical and socio-economic factors significantly influence the prevalence of fraud, and that distinct strategies may be needed to address these differences effectively.

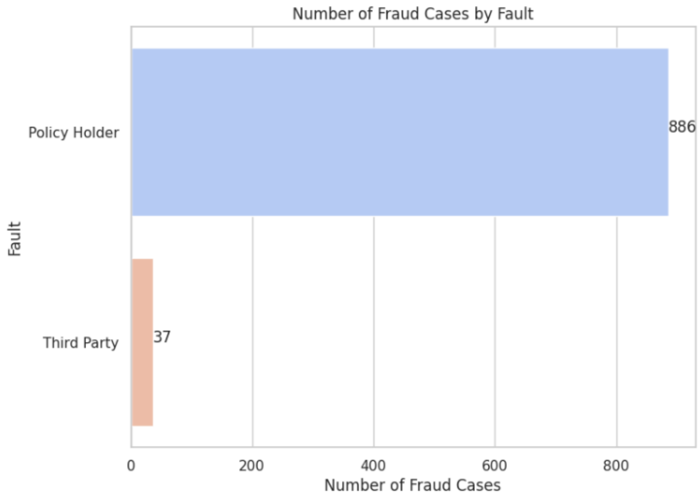


Figure 12: Percentage of fraud cases by fault

The analysis on figure 12 shows a clear difference between postulations, partially the blame on the policyholder and those on a third party. Quite a disturbing 886 out of the 923 unreasonable claim cases assessed suggest that it is the policyholder who is to blame, which implies that when the policyholder has caused the accident, he/she may be the filing obvious claims. This trend can be explained by the need to reduce the burden of personal responsibility or avoid paying the penalties that arise when one is said to be at fault and these include increased insurance costs or self-pay costs for repairs and /or damages.

Though, there were only 37 dishonest claims reported in which the negligence was on the third party. Not all out of the above scenarios were found to commit fraudulent acts, that is dishonest policyholders seeking benefits. In such a situation, er since the burden of charge rests with a third party there is low tendency for the claimants to twist or incite some fabrication of details since their assets are not at stake. This means that fraudulent claims tend to occur more often in proportion for policyholders seeking to reduce or do away with potential liability.

The noted contrast between these two categories stresses the perception of liability and its influence on the willingness to commit fraud. When the policyholders are the ones at fault, they might feel greater threat to their financial or legal security and could thus resort to dishonest practices as a way of protecting themselves. Conversely, when a third party is to blame, the perception of being less liable makes fraud more unappealing thus lowering chances of such claims being raised.

This fact illustrates that false insurance claim management, which also involves determining who is at fault for fraud as one of the steps, must be more sophisticated than it is at present. It may be more useful for insurers to use a more aggressive strategy targeting claims whereby policyholders are the ones at risk of being defrauded which in our case seems to induce potential fraud. More attention should be directed to these claims to possible fraud detection and management thereby safeguarding the system of insurance and limiting unnecessary expenditure.

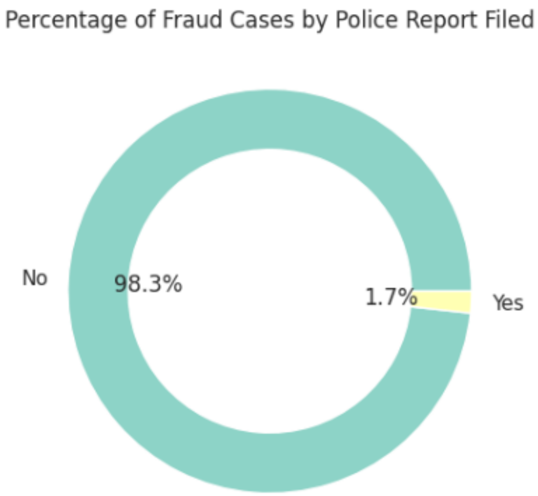


Figure 13: Percentage of fraud cases by police reports filed

The donut chart in Figure 13 represents the proportion of fraud cases depending on a filed police report which is further categorised into two sub-headings. There is a clear distinction between the two categories, most cases filed about reports (98.3%) do not have a police report attached to them. In sharp contrast, the percentage of instances of fraud reporting was 1.7 percent among the total number of cases.

From this analysis, it is evident that there has been an improvement in the response to such cases. Most of the recorded incidents are not convenient for presentation to the authorities as they should be or there is no need of presenting reports concerning such incidents to the law. There could be various reasons which could lead to this low level of reporting. This could explain the absence of willingness to report such incidences for fears that law enforcement may not be effective in dealing with such problems or may not prefer engaging law enforcement at all.

The absence of formal documentation may prevent the identification of fraudulent patterns, which may also be a consequence of fraud evolution. Such findings make it essential for organisations and law enforcement agencies to work together more effectively to record fraud. Organisations would be able to better manage the problem of investigation and prevention of fraud and create an environment where several- if not all- individuals would have the will to report acts of fraud. It would also lessen the ignorance of the general public and businesses in putting forth such outmost importance about reporting fraud.

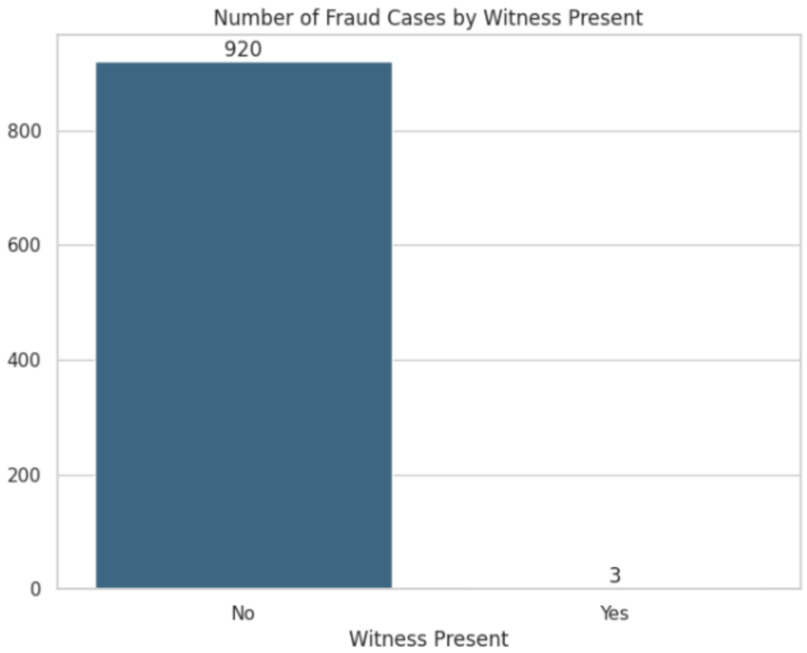


Figure 14: Number of fraud cases by witness present

The analysis on figure 14 demonstrates a distinct pattern in the presence of witnesses in connection with a fraudulent claim. Almost all of the cases, as many as 920, were carried out without any witnesses. It means that the vast majority of fraudulent claims are lodged without the support of any outside observer, an observation that seems to be a strategy targeting at reinforcing the fraudulent claims by the claimants. One reason may be that fraudsters seem to understand the risks that external witnesses actually pose to the fraud and so try to eliminate any external participants who would provide an opposing statement contradicting the claimed scenario.

Contrary, only three of the fraudulent reporting incidents were found to have included witnesses making document that even where a fraud is possible, third parties in these scenarios rarely get involved. Generally, the recall of witnesses when dealing with fraud cases is a rare occurrence, and several reasons could have caused this. One such reason could be that, the claimants are deliberately avoiding bringing in witnesses in order to reduce the chances of their fraudulent activities being discovered. Witnesses are deadly sources of evidence. They are sources who have no motive in support or rejection of the case made by the claimant. However, in given understanding of the fraud process, it appears more convenient for perpetrators to avoid the witnesses to reduce variation from anti daisy narratives that would discredit their fantasies.

Witnesses' absence not only makes it easy to undertake the fraudulent acts, but also increases the complexity of the claim assessment by the insurers, or other parties who have key interest in assessing such claims. In such scenarios, the availability of third parties as "witnesses" erodes the credibility of the case because such usefulness was not available. The absence of witnesses should not simply be seen as a coincidence, but a tactic aimed at committing fraud. This makes it easy for them to have greater leeway in how the version of events is construed in order to no risk their claims being disproved or denied. This reveals the importance of the opinion of witnesses to satisfying the intention of fraud and makes worthy the effort to develop appropriate ways of examining such instances.

4 Methodology

This chapter presents the description of the research methodology that shall be followed using machine learning in insurance claim fraud detection. The steps taken in conducting the research will be described in such a manner that the process will be both understandable and reproducible. The methodology involves a number of major stages including the collection and preprocessing of the data, feature engineering, the building of the models, and the evaluation. Much emphasis was therefore done during data collection: selecting the relevant data and also taking care of missing or wrong values with much emphasis on accuracy in the data set. Preprocessing, followed through different steps, involved transforming all categorical variables and normalising numerical data to set the data in the right format for analysis.

The section then discusses the different machine learning models that were built and tested, including simpler models like Logistic Regression and Decision Trees, as well as more complex ones like Gradient Boosting Machines. Each model was evaluated based on its ability to correctly identify fraudulent claims, focusing on metrics such as precision and recall to handle the imbalance in the dataset. The models were also evaluated using measures like the Receiver Operating Characteristic (ROC) curve and F1 score, which helped provide a better understanding of their performance. By following this structured approach, the methodology ensures that the project findings are reliable and can contribute to further studies and practical solutions for detecting insurance fraud.

4.1 Tools and Software used

Development Environment: Jupyter Notebook was the chosen environment for the development tasks such as writing, testing, and documenting the code. Its interactive nature facilitated step-by-step code execution and visualization, making it ideal for data science projects.

Programming language: Python was chosen as the main programming language in this analysis because of its strong ecosystem of libraries and frameworks that sustain different steps of machine learning and data science pipelines. Its flexibility and simplicity made manipulation of data, construction, and evaluation of the models easy to perform.

Python Libraries: The following libraries were used to handle data preprocessing, analysis, Visualization and for model development and evaluation.

- **Pandas:** utilised for reading, cleaning, exploration and manipulation of data. Pandas provides a flexible structure for loading and transforming the data.
- **Numpy:** Employed for numerical operations and array manipulations. Numpy supports efficient data transformations
- **Matplotlib and Searborn:** Used to create informative statistical graphics and visualizations, to help illustrate trends and patterns in the data
- **LabelEncoder:** Applied to convert categorical variables in to numerical format, making them ready for machine learning model.
- **Scikit-learn:** It provides a wide range of basic classification algorithms and tools for model evaluation.

4.2 Data Preprocessing

Since the dataset did not have any missing values, the preprocessing process was simple and straight forward. Several steps were taken to ensure that the data was in a suitable format for model training.

4.2.1 Data Cleaning

Some categorical variables contained inconsistent values (e.g., misspelled car makes such as “Porche” instead of “Porsche” and ’Mecedes’ instead of ’Mercedes’). These inconsistencies were corrected through manual replacements to standardize the data. The column ’Dayofweekclaimed’ also consisted of ’0’ instead of a particular day of the week and since there was only few rows, I then dropped the rows.

4.2.2 Feature Encoding

Categorical variables such as Month, DayOfWeek, and Make were encoded using LabelEncoder to convert them into numerical values suitable for model training. LabelEncoder is used to convert categorical variables into numerical values by assigning a unique integer to each distinct category.

4.2.3 Standardization

Numerical variables, including Age, PolicyNumber, and Deductible, were standardized using StandardScaler to ensure they had a mean of 0 and a standard deviation of 1. This step was essential for models sensitive to feature scaling, such as logistic regression.

4.3 Data splitting

For each experiment, we divided the dataset into 80% for training and 20% for testing. The training set was used for resampling, hyperparameter tuning, and model training, while the test set was utilised to evaluate the model’s performance. A random seed was set during the data split to ensure that the same split was consistently applied each time the program was run.

4.4 Handling Class Imbalance

The target variable FraudFound_P was highly imbalanced, with a significant majority of claims being non-fraudulent. An imbalanced class data distribution refers to a scenario where the number of instances belonging to one class is substantially lower than those in other classes, leading to a disproportionate representation of certain categories within the dataset (Pradipta et al., 2021). This imbalance can negatively impact the performance of classification models, as they may become biased towards the majority class. The class imbalance was addressed using the following techniques:

4.4.1 Oversampling

Synthetic Minority Over-sampling Technique (SMOTE) was applied to generate synthetic samples for the minority class (fraudulent claims), resulting in a balanced training set. The necessary procedure of SMOTE is by creating the new synthetic data referring to the neighboring minority class instance, then the new synthetic data is made to determine the closest minority instance and to make the interpolation line between the sample data and selected minority instance so that the synthetic data made along the line can increase the number of minority class instances by introducing new minority class examples in the neighborhood, thereby assisting the classifiers to improve their generalisation capacity (Pradipta et al., 2021).

4.4.2 Undersampling

To complement the oversampling approach, random undersampling was also performed to reduce the number of non-fraudulent samples, ensuring a balanced class distribution and mitigating potential model bias. Undersampling is the process of reducing the number of instances in the majority class by removing existing samples until the class distribution achieves a specified ratio relative to the minority class (Mulyana Saripuddin et al., 2021). This technique helps to balance the dataset and mitigate the effects of class imbalance, enabling the model to focus more on learning the patterns of the minority class.

4.5 Hyperparameter tuning

To further optimise the performance of the machine learning models, hyperparameter tuning was performed. Hyperparameters are model-specific parameters that cannot be learned from the data and therefore need to be set before the training process. These parameters can significantly affect the performance of the models. We employed GridSearchCV, a method that systematically works through multiple combinations of hyperparameter values, cross-validating as it goes to determine the best parameter set. This technique was applied to fine-tune key parameters for each model. We only applied hyperparameter tuning on the XGBoost model since it was the best performing. The following parameter were the best parameters for the model:

- **Fitting 5 folds:** This indicates that 5-fold cross-validation is used to validate the model. The dataset is split into 5 parts, where each part is used once as a validation set while the remaining 4 parts are used for training. This helps in ensuring the model generalizes well to unseen data.

- **Learning rate = 0.1:** The learning rate determines the step size at each iteration while moving towards a minimum of the loss function. A lower learning rate can lead to a better model but requires more trees (estimators) to converge.

- **Max depth = 5:** This parameter sets the maximum depth of the trees in the model. A deeper tree can model more complex relationships but may also lead to overfitting. A depth of 5 strikes a balance between model complexity and generalization.

- **n_estimators = 100:** This parameter represents the number of trees (estimators) to be built in the ensemble. More trees generally improve performance, but too many can lead to overfitting. A value of 100 is often a good starting point for many problems.

4.6 Model Development

Several machine learning models were developed and evaluated, including:

4.6.1 Random Forest Classifier

Random Forest involves using several tree classifiers to make predictions and is considered an ensemble learning technique that builds multiple classification trees, and for each new feature vector being classified, it is assessed by every tree in the forest. Every tree provides a classification, essentially giving a "vote" for that specific class and the outcome of the Random Forest is based on choosing the classification with the most votes across all trees (Arun et al., 2016). Among the many advantages of Random Forest, some of the most notable include its exceptional accuracy compared to other contemporary algorithms, its ability to efficiently process large datasets, and its straightforward structure that allows for easy storage and future utilisation of pre-generated. Random forest was trained on both the original and balanced datasets to evaluate the impact of data balancing on model performance.

4.6.2 Logistic Regression

Logistic regression is a commonly used statistical technique for binary classification, aiming to depict the link between a dependent variable and independent variables. A logistic function is applied to a weighted total of input features to calculate the probability of a certain input being part of a certain class. The outcome ranges from 0 to 1, indicating the likelihood of the input being classified as the positive category. One of the main benefits of logistic regression is how easily understandable it is, allowing for a clear understanding of how each predictor influences the outcome. Additionally, it has the capability to manage continuous and categorical independent variables, making it a flexible choice for various situations (James et al., 2013). Logistic regression was employed as a baseline model due to its simplicity and interpretability.

4.6.3 XGBoost Classifier

XGBoost (Extreme Gradient Boosting) is a powerful machine learning algorithm that enhances the traditional gradient boosting framework through several optimisations. It constructs an ensemble of decision trees sequentially, with each new tree designed to correct the errors made by its predecessor. XGBoost stands out for its use of regularization methods, which prevent over-fitting and enhance generalisation, resulting in its high effectiveness with big datasets and complex tasks. The algorithm utilises a sophisticated objective function that aims to enhance both accuracy and computational efficiency while training. Moreover, XGBoost enables parallel processing, which leads to a considerable increase in speed compared to traditional boosting techniques. It's flexibility and strong performance have helped make it widely used in different machine learning competitions and practical scenarios (Chen & Guestrin, 2016; Friedman, 2001; Li et al., 2019). XGBoost was implemented to leverage its robustness in handling class imbalance and capturing complex patterns in the data.

4.6.4 K-Nearest Neighbor Classifier

K-Nearest Neighbors (KNN) is a non-parametric algorithm that relies on instances for classification and regression purposes. It identifies a given input by locating the 'K' nearest training instances in the feature space and labeling it with the most frequent class among these neighbors. Distance metrics like Euclidean or Manhattan distance are commonly used to measure the closeness of points. Implementing KNN is straightforward and easy to understand, as it only considers the distance between data points when making decisions. Nevertheless, it may require high computational costs and is also influenced by the selection of 'K' and the feature's scale. Although KNN has limitations, it is still widely used because of its flexibility and capacity to understand intricate connections in small to medium datasets (Cover & Hart, 1967; Altman, 1992; Peterson, 2009).

4.7 Model Evaluation

This section assesses the effectiveness of the models trained for identifying insurance claim fraud by analysing various metrics such as Accuracy, Precision, Recall, F1-Score, Confusion Matrix, and Area Under the Precision-Recall Curve (AUC-PR). These measurements offer a complete comprehension of the models' effectiveness in detecting fraudulent claims and reveal the pros and cons of each model.

4.7.1 Accuracy

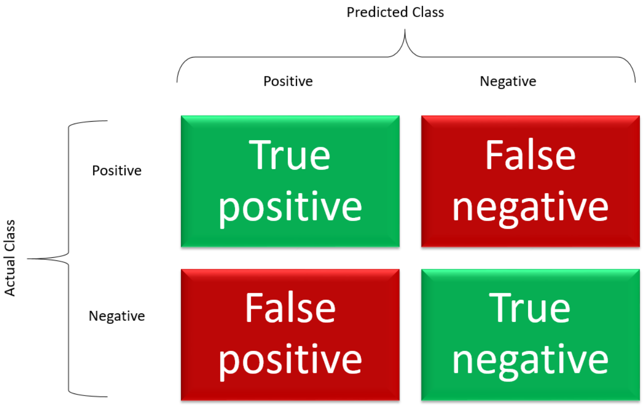
assesses how many correctly predicted observations there are compared to the total observations, indicating the overall performance of the model. A greater accuracy value shows that more instances were correctly classified by the model.

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \tag{1}$$

While accuracy is a useful measure in balanced datasets, it can be misleading when the classes are imbalanced. For example, if the dataset has 95% non-fraudulent claims and 5% fraudulent claims, a model that always predicts "non-fraudulent" will achieve an accuracy of 95%. However, such a model is ineffective in detecting the minority class (Sokolova et al., 2009).

4.7.2 Confusion Matrix

The Confusion Matrix is a tabular representation that summarises the performance of a classification model. It helps visualize the number of correct and incorrect predictions and provides deeper insights into model performance for each class. The Confusion Matrix is represented as:



		Predicted Class	
		Positive	Negative
Actual Class	Positive	True positive	False negative
	Negative	False positive	True negative

True Positive (TP): The number of correctly classified positive samples (fraudulent claims correctly identified).

False Negative (FN): The number of positive samples incorrectly classified as negative (fraudulent claims missed by the model).

False Positive (FP): The number of negative samples incorrectly classified as positive (non-fraudulent claims wrongly identified as fraudulent).

True Negative (TN): The number of correctly classified negative samples (non-fraudulent claims correctly identified).
These values can be used to compute other evaluation metrics like Precision, Recall, and F1-Score. A good model aims to maximise TP and TN while minimising FP and FN (Powers, 2011).

4.7.3 Precision

Measures the ratio of correctly predicted positive observations (fraudulent claims) to the total predicted positive observations. It is particularly important in scenarios where the cost of false positives is high, such as in fraud detection, where wrongly classifying a legitimate claim as fraudulent can have serious consequences.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{2}$$

A high precision value means that the model generates few false positives, ensuring that the identified fraudulent claims are likely to be truly fraudulent. However, focusing solely on precision can lead to a low recall, which means that many actual fraudulent claims are missed (Powers, 2011).

4.7.4 Recall

Measures the ratio of correctly predicted positive observations to all observations in the actual class. It is used to capture as many positive cases as possible and is crucial when missing a positive case (e.g., a fraudulent claim) has a high cost.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{3}$$

A high recall value signifies that most of the actual fraudulent claims are identified by the model, so there will be fewer false negatives. However, maximising recall may result in an increase in those false positives that reduce precision.

4.7.5 F1-Score

The F1-Score is the harmonic mean of Precision and Recall and proves to be particularly useful in cases of imbalanced class distribution. It provides a balance between precision and recall, taking both values into consideration, hence making it more informative than accuracy when using model evaluation on highly skewed datasets.

$$\text{F1-Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2 \times \text{TP}}{2 \times \text{TP + (FP + FN)}} \tag{4}$$

The F1-Score ranges from 0 to 1, with a higher value indicating a better balance between precision and recall. It is an effective metric when one class is more important than the other, as it emphasis the need to minimise both false positives and false negatives.

4.7.6 Area Under the Precision-Recall Curve(AUC-PR)

AUC-PR provides an aggregate measure of model performance across the different threshold values by considering the trade-offs between precision and recall. The AUC-PR curve is better than the ROC curve for imbalanced datasets, because the ROC curve is more indicated for balanced datasets whereas AUC-PR focuses on the positive class performance in cases of fraud.

$$\text{AUC-PR} = \int_0^1 \text{Precision}(r)\, dr \tag{5}$$

A higher AUC-PR score indicates that the model maintains a good trade-off between precision and recall, even when the threshold for classification is varied. This metric is particularly valuable in scenarios where the primary goal is to accurately detect a rare class, such as fraud detection (Davis & Goadrich, 2016).

5 Results and Discussions

In this section we discuss the results of each model. For a comprehensive view of the code and additional results, please visit my GitHub repository: GitHub Repository.

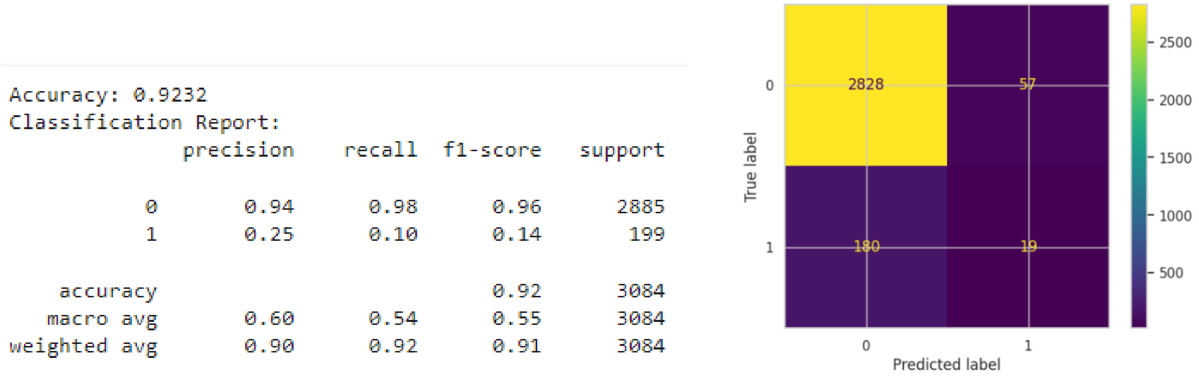
5.1 Random Forest

5.1.1 No resampling

The Random Forest model was evaluated on the original dataset without any resampling, and the results show a strong performance in classifying non-fraudulent claims (class 0), while struggling significantly with identifying fraudulent claims (class 1). The overall accuracy of the model was 92.32%, indicating that the model made correct predictions for the vast majority of samples. However, as seen in previous models, this high accuracy largely reflects the model's performance on the majority class. For class 0 (non-fraudulent claims), the model achieved a precision of 0.94 and a recall of 0.98, which means that nearly all of the claims predicted as non-fraudulent were correct and the model was able to correctly identify most non-fraudulent claims. The F1-score for class 0 was 0.96, indicating a strong balance between precision and recall for the majority class.

However, the model's performance for class 1 (fraudulent claims) was less satisfactory. The precision for fraudulent claims was 0.25, meaning that only 25% of the claims predicted as fraudulent were actually fraudulent. The recall was even lower, at 0.10, indicating that the model was only able to identify 10% of the actual fraudulent claims. The F1-score for fraudulent claims was 0.14, reflecting the imbalance between precision and recall for the minority class. The AUC-PR value of 0.2019 further emphasis the model's difficulty in distinguishing between fraudulent and non-fraudulent claims, particularly when dealing with the minority class (fraudulent claims)

Figure 15(b) shows the confusion matrix which further highlights these performance issues. The model correctly classified 2828 non-fraudulent claims out of 2885 (True Negatives) but misclassified 57 non-fraudulent claims as fraudulent (False Positives). For the fraudulent claims, the model correctly identified 19 out of 199 fraudulent claims (True Positives) but misclassified 180 fraudulent claims as non-fraudulent (False Negatives). This large number of false negatives is a significant concern, as it means the model missed a majority of the actual fraudulent claims. In conclusion, while the Random Forest model performs exceptionally well in classifying non-fraudulent claims, it struggles to detect fraudulent claims, as indicated by the low precision, recall, and F1-score for class 1. The high number of false negatives suggests that the model is heavily biased towards the majority class, making it unreliable for detecting fraud in this dataset.



```
Accuracy: 0.9232
Classification Report:
              precision    recall  f1-score   support

           0       0.94      0.98      0.96      2885
           1       0.25      0.10      0.14       199

    accuracy                           0.92      3084
   macro avg       0.60      0.54      0.55      3084
weighted avg       0.90      0.92      0.91      3084
```

(a) Classification Report

(b) Confusion Matrix

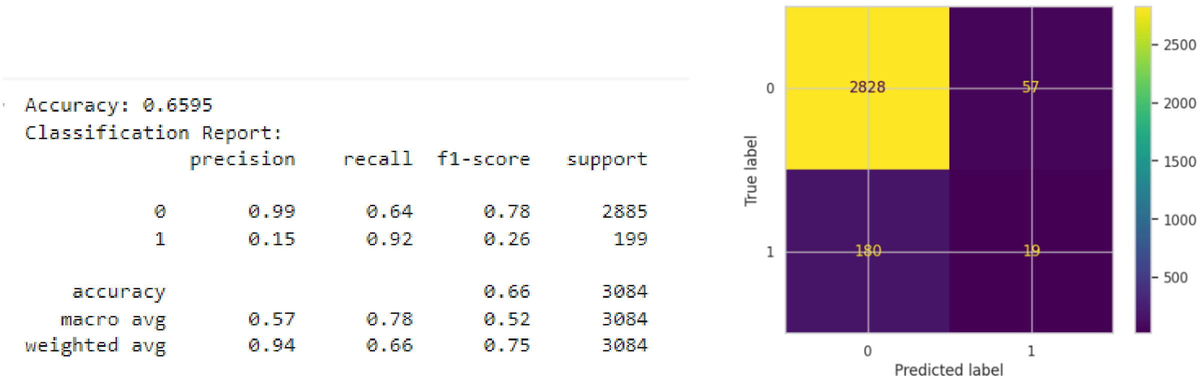
Figure 15: Classification report and confusion matrix Random Forest without resampling

5.1.2 Sampling

The Random Forest model was re-evaluated after applying SMOTE to address the class imbalance, and the results demonstrate improvements in detecting fraudulent claims, though there are still limitations. The overall accuracy of the model decreased to 65.95%, reflecting the effects of resampling on the performance of the model, especially with respect to the majority class. For class 0 (non-fraudulent claims), the precision remained high at 0.99, meaning that almost all non-fraudulent claims predicted by the model were correct. However, the recall for non-fraudulent claims dropped significantly to 0.64, meaning that the model correctly identified only 64% of the actual non-fraudulent claims. This is a considerable decrease compared to the model without resampling, where the recall for class 0 was much higher. The F1-score for class 0 was 0.78, indicating a balance between precision and recall, but the lower recall impacted overall performance.

For class 1 (fraudulent claims), the sampling technique had a positive impact, particularly on recall. The precision for fraudulent claims was 0.15, which is relatively low, indicating that 15% of the claims flagged as fraudulent were actually fraudulent. However, the recall for fraudulent claims increased significantly to 0.92, meaning that the model was able to correctly identify 92% of the actual fraudulent claims. The F1-score for class 1 was 0.26, reflecting an improvement in the model's ability to detect fraudulent claims, although the precision is still low. The AUC-PR value of 0.2019 highlights the difficulty the model faces in balancing precision and recall for the minority class, especially after resampling.

The confusion matrix (as shown in 16(b)) provides further insights into the model's performance after sampling. The model correctly classified 2828 non-fraudulent claims out of 2885 but misclassified 57 non-fraudulent claims as fraudulent (false positives). For fraudulent claims, the model correctly identified 19 out of 199 (true positives), while misclassifying 180 fraudulent claims as non-fraudulent (false negatives). The large number of false negatives indicates that the model still struggles with detecting fraudulent claims even after resampling. While resampling improved the recall for fraudulent claims, it came at the expense of precision and the ability to correctly classify non-fraudulent claims. The low precision for fraudulent claims and the high number of false positives suggest that further improvements are necessary to optimise the model's performance.



```
Accuracy: 0.6595
Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.64      0.78      2885
           1       0.15      0.92      0.26       199

    accuracy                           0.66      3084
   macro avg       0.57      0.78      0.52      3084
weighted avg       0.94      0.66      0.75      3084
```

(a) Classification Report with Sampling

(b) Confusion Matrix with Sampling

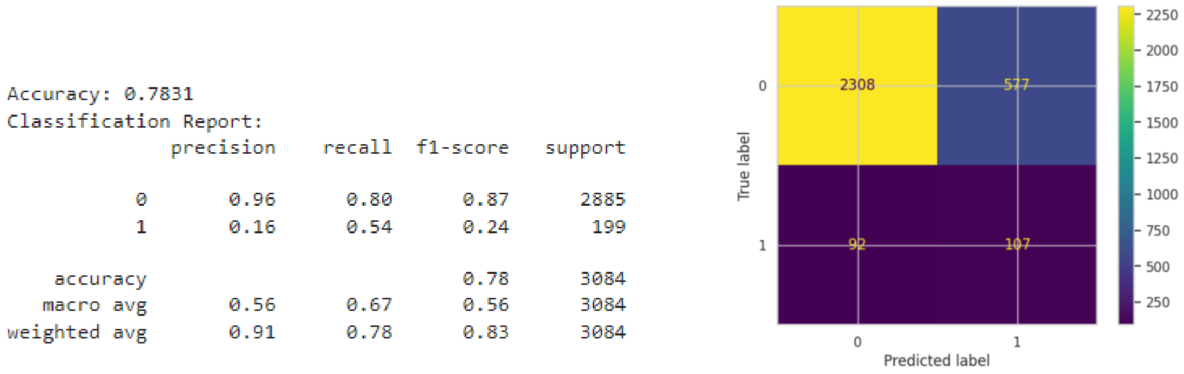
Figure 16: Classification report and confusion matrix for Random Forest with SMOTE

5.2 Logistic Regression

5.2.1 No resampling

The Logistic Regression model performed well in classifying the legitimate (non-fraudulent) claims, as expected, given the imbalanced nature of the dataset. The precision, recall, and F1-score for class 0 (non-fraudulent) were 0.96, 0.80, and 0.87, respectively. These metrics indicate that the model was very capable of correctly predicting non-fraudulent claims, with relatively few errors in this class. However, the model struggled with detecting fraudulent claims. The precision for class 1 (fraudulent claims) was 0.16, which indicates that only 16% of the claims flagged as fraudulent were truly fraudulent. The recall was 0.54, meaning that only 54% of the actual fraudulent claims were correctly identified by the model. This imbalance in performance highlights that the model is not performing well at detecting fraud, which is critical in this application. The F1-score for class 1 was 0.24, which reflects the poor balance between precision and recall for fraudulent claims.

The AUC-PR value of 0.361 further supports this finding, as it shows that the model struggles to maintain a good balance between precision and recall when classifying fraudulent claims. The confusion matrix shown in Figure 17(b) further illustrates this, where the model correctly classified 2308 non-fraudulent claims and 107 fraudulent claims, but it misclassified 577 non-fraudulent claims as fraudulent and 92 fraudulent claims as non-fraudulent. These results show that while the logistic regression model performs well in detecting non-fraudulent claims, it fails to adequately identify fraudulent ones.



```
Accuracy: 0.7831
Classification Report:
              precision    recall  f1-score   support

           0       0.96      0.80      0.87      2885
           1       0.16      0.54      0.24       199

    accuracy                           0.78      3084
   macro avg       0.56      0.67      0.56      3084
weighted avg       0.91      0.78      0.83      3084
```

(a) Classification Report

(b) Confusion Matrix

Figure 17: Classification report and confusion matrix for Logistic Regression without resampling

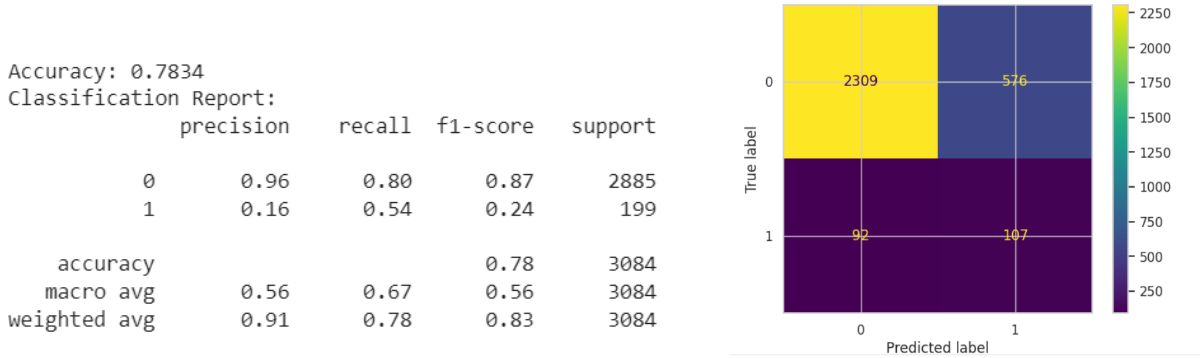
5.2.2 Sampling

The Logistic Regression model was re-evaluated following the application of SMOTE to address the inherent class imbalance within the dataset. The model achieved an overall accuracy of 78.34%, demonstrating its capability to manage imbalanced data more effectively, particularly in terms of detecting fraudulent claims (class 1). Despite this improvement, the model continues to exhibit certain limitations, particularly with respect to precision. In terms of performance on class 0 (non-fraudulent claims), the model maintained a high precision of 0.96, indicating that 96% of the predicted non-fraudulent claims were correctly classified. The recall for class 0, however, declined slightly to 0.80, reflecting the model's ability to correctly identify 80% of the actual non-fraudulent claims. The F1-score for this class was 0.87, signifying a strong balance between precision and recall, although the decrease in recall compared to the non-resampled model suggests that the resampling process introduced a minor trade-off in performance for the majority class.

For class 1 (fraudulent claims), the resampling technique markedly improved the model's recall but had a less pronounced effect on precision. The precision for class 1 remained low at 0.16, suggesting that only 16% of the claims flagged as fraudulent were indeed fraudulent. However, the recall for class 1 increased to 0.54, signifying that the model was able to correctly identify 54% of the actual fraudulent claims, a substantial improvement from the results prior to resampling. The F1-score for fraudulent claims was 0.24, reflecting a modest improvement in the balance between precision and recall. Nevertheless, the low precision indicates a propensity for the model to misclassify non-fraudulent claims as fraudulent, resulting in a high number of false positives. The AUC-PR score of 0.362 further underscores the model's limitations in distinguishing between fraudulent and non-fraudulent claims, even after the application of resampling techniques. While the recall for class 1 improved, the model's ability to precisely classify fraudulent claims remains constrained.

The confusion matrix on figure 18(b) provides a detailed breakdown of the model's performance. The model correctly classified 2309 non-fraudulent claims, while 576 non-fraudulent claims were incorrectly classified as fraudulent, resulting in a substantial number of false positives. With respect to fraudulent claims, the model correctly identified 107 out of 199 fraudulent claims, while 92 fraudulent claims were misclassified as non-fraudulent, indicating that the model continues to exhibit a notable number of false negatives.

In conclusion, while the application of resampling techniques led to an improvement in recall for the minority class, the model still struggles with low precision and a high number of false positives. These results suggest that while resampling can enhance the model's sensitivity to fraudulent claims, it also introduces trade-offs that limit overall precision. Further optimisation of resampling methods or the exploration of more advanced classification algorithms may be necessary to achieve a more balanced performance in fraud detection tasks, particularly in reducing the number of misclassified claims.



```
Accuracy: 0.7834
Classification Report:
              precision    recall  f1-score   support

           0       0.96      0.80      0.87      2885
           1       0.16      0.54      0.24       199

    accuracy                           0.78      3084
   macro avg       0.56      0.67      0.56      3084
weighted avg       0.91      0.78      0.83      3084
```

(a) Classification Report

(b) Confusion Matrix

Figure 18: Classification report and confusion matrix for Logistic Regression with SMOTE

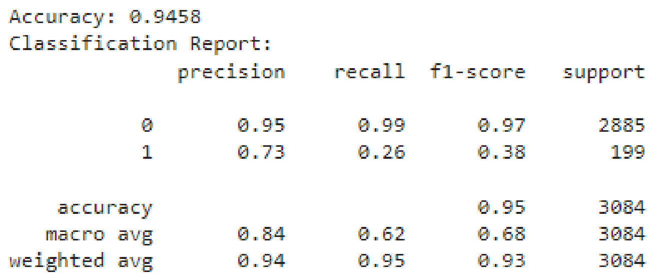
5.3 XGBoost Classifier

5.3.1 No resampling

The XGBoost model performed exceptionally well in classifying legitimate claims, even without resampling, with precision, recall, and F1-scores of 0.95, 0.99, and 0.97, respectively, as shown in figure 16(a). This high performance was expected due to the imbalanced nature of the dataset, where most samples belonged to the legitimate class. The model's ability to predict legitimate claims was strong, minimising false positives and ensuring high precision and recall for non-fraudulent claims. The model correctly identified 2866 non-fraudulent claims out of 2885, demonstrating its effectiveness in handling the majority class.

However, the performance of the model was less satisfactory when dealing with the fraudulent class. The precision and recall for the fraudulent class were 0.73 and 0.26, respectively, indicating that although a significant proportion of the flagged fraudulent claims were correct, the model missed a large number of actual fraudulent cases. Specifically, the model identified only 51 fraudulent claims out of 199, with 148 fraudulent claims misclassified as non-fraudulent. This low recall shows that the model struggled to identify a majority of fraudulent claims, which is critical for this task since missing fraudulent claims can result in significant financial losses for insurance companies. figure 19(b) illustrates the confusion matrix of the XGBoost model without the application of any resampling methods. The matrix reveals that while the model effectively classified a large number of legitimate claims (non-fraudulent), it struggled to detect fraudulent claims, with 148 fraudulent cases being incorrectly classified as legitimate.

Despite the high accuracy of 94.58%, the overall ability of the model to detect fraudulent claims remains limited. While high accuracy can be achieved by focusing on the majority class, the goal of this project is to maximise the detection of fraudulent claims.

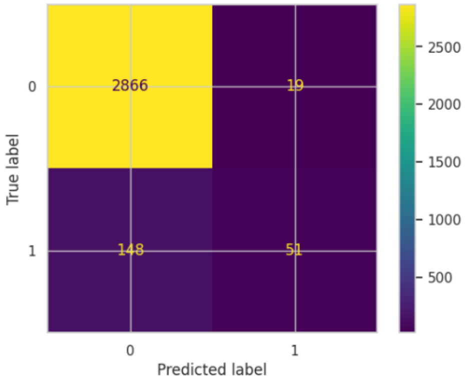


```
Accuracy: 0.9458
Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.99      0.97      2885
           1       0.73      0.26      0.38       199

    accuracy                           0.95      3084
   macro avg       0.84      0.62      0.68      3084
weighted avg       0.94      0.95      0.93      3084
```

(a) Classification Report



(b) Confusion Matrix

Figure 19: Classification report and confusion matrix for XGBoost without resampling
Fitting 5 folds for each of 36 candidates, totalling 180 fits

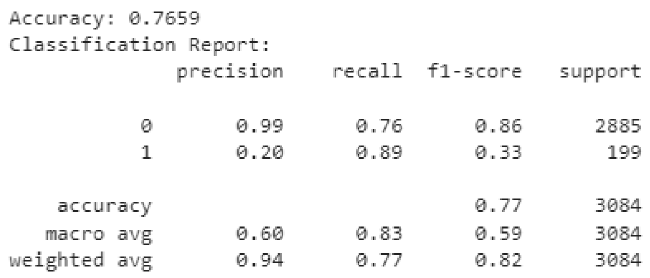
5.3.2 Sampling

Parameters: Fitting 5 folds, Learning rate = 0.1, max depth = 5, n_estimators = 100.

The XGBoost model was re-evaluated after applying SMOTE to address the class imbalance in the dataset. With resampling, the model showed a marked improvement in detecting fraudulent claims but still struggled with balancing precision and recall for the minority class. In terms of classifying non-fraudulent claims (class 0), the precision remained very high at 0.99, indicating that the model correctly classified most non-fraudulent claims with a minimal number of false positives. However, the recall for class 0 dropped to 0.76, meaning that 24% of the non-fraudulent claims were misclassified as fraudulent. The F1-score for class 0 was 0.86, reflecting a strong balance between precision and recall despite the reduced recall for this class.

For fraudulent claims (class 1), the model's precision dropped to 0.20, suggesting that only 20% of the claims predicted as fraudulent were indeed fraudulent. This indicates that a large number of non-fraudulent claims were incorrectly flagged as fraudulent. However, the recall for fraudulent claims improved significantly to 0.89, indicating that the model was able to identify 89% of the actual fraudulent claims. This is a significant improvement compared to the model without resampling, where a majority of fraudulent claims were missed. The F1-score for class 1 improved to 0.33, which, while still low, shows a better balance between precision and recall for detecting fraudulent claims compared to the previous model.

The confusion matrix (as shown in Figure 20(b)) provides further insight into the model's performance. For class 0 (non-fraudulent claims), the model correctly classified 2184 legitimate claims but misclassified 701 of them as fraudulent. This represents a noticeable increase in false positives compared to the model without resampling. For class 1 (fraudulent claims), the model correctly identified 178 out of 199 fraudulent claims, with only 21 being misclassified as non-fraudulent. This is a significant improvement in detecting fraudulent claims, with far fewer false negatives compared to the model without resampling.

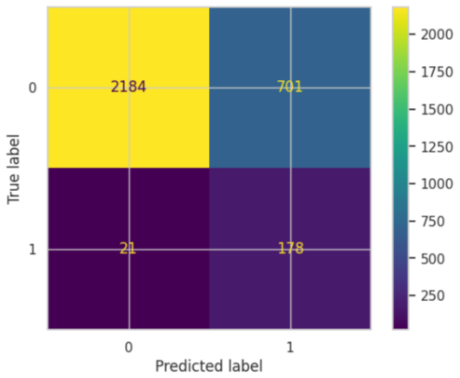


```
Accuracy: 0.7659
Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.76      0.86      2885
           1       0.20      0.89      0.33       199

    accuracy                           0.77      3084
   macro avg       0.60      0.83      0.59      3084
weighted avg       0.94      0.77      0.82      3084
```

(a) Classification Report with sampling



(b) Confusion Matrix with sampling

Figure 20: Classification report and confusion matrix for XGBoost with SMOTE

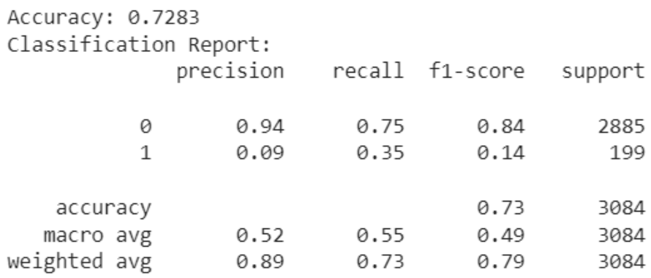
5.4 K-Nearest Neighbor Classifier

5.4.1 Sampling

The K-Nearest Neighbors (KNN) model was evaluated with SMOTE, and the results indicate that while the model performed reasonably well for the majority class (non-fraudulent claims), it faced significant difficulties in identifying fraudulent claims. The overall accuracy of the model was 72.83%, reflecting its ability to correctly classify a majority of the samples, but as expected with imbalanced data, the model struggled with the minority class.

For class 0 (non-fraudulent claims), the KNN model achieved a precision of 0.94, indicating that 94% of the claims predicted as non-fraudulent were indeed non-fraudulent. The recall for non-fraudulent claims was 0.75, meaning that 75% of the actual non-fraudulent claims were correctly identified by the model. The F1-score for class 0 was 0.84, reflecting a decent balance between precision and recall, although there is room for improvement in recall, as the model missed a significant portion of non-fraudulent claims. However, the model's performance for class 1 (fraudulent claims) was much less satisfactory. The precision for fraudulent claims was 0.09, meaning that only 9% of the claims predicted as fraudulent were actually fraudulent, indicating a high number of false positives. The recall for fraudulent claims was 0.35, meaning that the model was able to correctly identify only 35% of the actual fraudulent claims. The F1-score for class 1 was 0.14, reflecting the significant imbalance between precision and recall for detecting fraudulent claims. The AUC-PR value of 0.24 further highlights the model's struggles in distinguishing between fraudulent and non-fraudulent claims, particularly when dealing with the minority class.

The confusion matrix (as shown in Figure 21(b)) provides further insight into the model's performance. The model correctly classified 2176 non-fraudulent claims as non-fraudulent, but misclassified 709 non-fraudulent claims as fraudulent (false positives). For fraudulent claims, the model correctly identified 70 fraudulent claims (true positives) but misclassified 129 fraudulent claims as non-fraudulent (false negatives). This indicates that the model is biased towards the majority class and has a high rate of both false positives and false negatives when dealing with fraudulent claims.

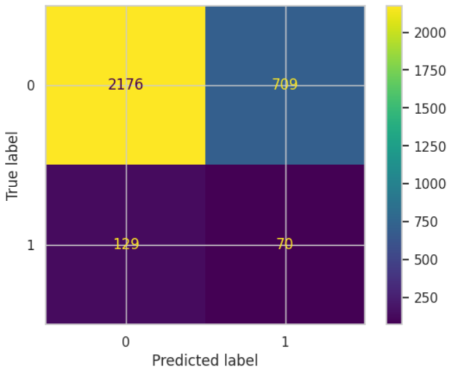


```
Accuracy: 0.7283
Classification Report:
              precision    recall  f1-score   support

           0       0.94      0.75      0.84      2885
           1       0.09      0.35      0.14       199

    accuracy                           0.73      3084
   macro avg       0.52      0.55      0.49      3084
weighted avg       0.89      0.73      0.79      3084
```

(a) Classification Report



(b) Confusion Matrix

Figure 21: Classification report and confusion matrix for KNN with SMOTE

6 Conclusion

The primary goal of this project was to improve the detection of fraudulent insurance claims using machine learning models. Insurance fraud, especially within vehicle claims, presents a significant financial burden to insurance companies and undermines the trust between providers and policyholders. In this project, we explored the application of several machine learning models—including Random Forest, Logistic Regression, XGBoost, and K-Nearest Neighbors—to identify fraudulent claims based on a real-world dataset.

One of the main challenges was the imbalanced nature of the dataset, with fraudulent claims making up only a small percentage of the total. This imbalance often leads to models that, while accurate overall, fail to effectively identify fraud. To address this, SMOTE(Synthetic Minority Over-sampling Technique) were used to re-balance the dataset and enhance the model's ability to detect fraudulent claims. XGBoost showed particularly strong performance when these resampling techniques were applied.

These results confirmed that maintaining precision with recall in fraud detection is critical since missing fraudulent claims (False negatives) and flagging non-fraudulent one as threats result substantial costs. This surprisingly achieved high accuracy, however, the models lacked precision which resulted in an increase of false positives. The problem with this is that when authentic claims are falsely established as fraudulent, it can prompt an investigation and ruin the relationship between those clients who were honest. Despite these challenges, the project has shown that machine learning is a useful weapon in the battle to better anti-fraud. But more still needs to be done, as policies can also lead to false positives and false negatives. More sophisticated techniques, such as deep learning or ensemble methods (combining the output of multiple machine classifiers) may also be more effective in detecting fraud, and the incorporation external data sources like economic indicators or social trends into the models. Further, the inclusion of time-correlated data such as when claims were submitted may provide additional granularity on fraud patterns.

The next steps are to work on improving these models and testing them against larger, more varied datasets. This will help further enhance their accuracy and high performance. We know that machine learning is a field that evolves, and as fraud techniques get more elaborate so must our technology to detect them. Ongoing research and development in this arena are essential to building systems that not only detect fraud, but do so with minimal disruption for actual policyholders. Ongoing work will be required to ensure they remain effective in the face of changing fraud tactics. By combining machine learning with other analytical techniques, insurers can build more robust and adaptable fraud detection systems that protect their business and foster trust with their customers.

References

[1] Viaene, S. and Dedene, G., 2004. Insurance Fraud: Issues and Challenges. *The Geneva Papers on Risk and Insurance*, 29(2), pp.313–333.

[2] Derrig, R.A., 2002. Insurance Fraud. *The Journal of Risk and Insurance*, 69(3), pp.271–287.

[3] Bansal, S., Vehicle Claim Fraud Detection Dataset. Kaggle. Available at: `https://www.kaggle.com/datasets/shivamb/vehicle-claim-fraud-detection/data` [Accessed 20 Oct. 2024].

[4] Gomes, C., Jin, Z. and Yang, H., 2021. Insurance fraud detection with unsupervised deep learning. *Journal of Risk and Insurance*, 88, pp.591–624.

[5] Kemp, G., 2010. Fighting public sector fraud in the 21st century. *Computer Fraud and Security*, 11, pp.16–28.

[6] Debener, J., Heinke, V. and Kriebel, J., 2023. Detecting insurance fraud using supervised and unsupervised machine learning. *Journal of Risk and Insurance*, 90, pp.743–768.

[7] PricewaterhouseCoopers International Limited, 2011. Fighting fraud in the public sector. *Controlling fraud in the public sector.* Available at: `http://www.pwc.com/gx/en/psrc/pdf/fighting_fraud_in_the_public_sector_june2011.pdf` [Accessed 20 Oct. 2024].

[8] Abbasi, E., Moghaddam, M. and Kowsari, E., 2022. A systematic and critical review on development of machine learning based-ensemble models for prediction of adsorption process efficiency. *Journal of Cleaner Production*, 379.

[9] Krambia-Kapardis, M., 2002. A fraud detection model: A must for auditors. *Journal of Financial Regulation and Compliance*, 10(3).

[10] Wang, Y., Yu, M. and Gao, S., 2022. Gender diversity and financial statement fraud. *Journal of Accounting and Public Policy*, 41(2).

[11] Pradipta, G.A., Wardoyo, R., Musdholifah, A., Sanjaya, I.N.H. and Ismail, M., 2021. SMOTE for Handling Imbalanced Data Problem: A Review. *IEEE Xplore.* Available at: `https://doi.org/10.1109/ICIC54025.2021.9632912` [Accessed 20 Oct. 2024].

[12] Saripuddin, M., Suliman, A. and Sameon, S.S., 2021. Random undersampling on imbalance time series data for anomaly detection. *The 4th International Conference on Machine Learning and Machine Intelligence.* Available at: `https://doi.org/10.1145/3490725.3490748` [Accessed 20 Oct. 2024].

[13] Kulkarni, A.D. and Lowe, B., 2016. Random Forest Algorithm for Land Cover Classification. *Computer Science Faculty Publications and Presentations.* Available at: `https://scholarworks.uttyler.edu/compsci_fac/1/` [Accessed 20 Oct. 2024].

[14] James, G., Witten, D., Hastie, T. and Tibshirani, R., 2013. *An Introduction to Statistical Learning with Applications in R.* Springer.

[15] Chen, T. and Guestrin, C., 2016. XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.785–794.

[16] Friedman, J.H., 2001. Greedy Function Approximation: A Gradient Boosting Machine. *The Annals of Statistics*, 29(5), pp.1189–1232.

[17] Li, Y., Chen, Y. and Xu, Z., 2019. An Empirical Study on the Performance of XGBoost for Classification and Regression. *International Journal of Machine Learning and Computing*, 9(1), pp.1–7.

[18] Cover, T. and Hart, P., 1967. Nearest Neighbor Pattern Classification. *IEEE Transactions on Information Theory*, 13(1), pp.21–27.

[19] Altman, N.S., 1992. An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression. *The American Statistician*, 46(3), pp.175–185.

[20] Peterson, L.E., 2009. K-nearest neighbor. *Scholarpedia*, 4(2), p.1883.

[21] Sokolova, M. and Lapalme, G., 2009. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4), pp.427–437.

[22] Powers, D.M., 2011. Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. *Journal of Machine Learning Technologies*, 2(1), pp.37–63.

[23] Davis, J. and Goadrich, M., 2006. The relationship between Precision-Recall and ROC curves. *Proceedings of the 23rd International Conference on Machine Learning*, pp.233–240.