UNIVERSITY OF CALIFORNIA

Los Angeles

Fraud Detection in Vehicle Insurance Claims using Machine Learning

A dissertation submitted in partial satisfaction
of the requirements for the degree
Master of Applied Statistics and Data Science

by

Ziyang Zhang

2024

ABSTRACT OF THE DISSERTATION

Fraud Detection in Vehicle Insurance Claims using Machine Learning

by

Ziyang Zhang
Master of Applied Statistics and Data Science
University of California, Los Angeles, 2024
Professor Yingnian Wu, Chair

Insurance fraud poses a significant financial burden on the industry, with fraudulent vehicle insurance claims being a major contributor. This study explores the application of machine learning techniques to accurately detect fraudulent vehicle insurance claims. Six different models - Logistic Regression, Random Forest, Gaussian Naive Bayes, Decision Tree, XGBoost, and Gradient Boosting classifiers - are evaluated on an imbalanced dataset. To address class imbalance, oversampling techniques like SMOTE, Borderline SMOTE, and ADASYN are employed. Performance is assessed using metrics such as F1 score, recall, and AUC. Results indicate that XGBoost and Gradient Boosting models demonstrate superior overall performance, effectively balancing precision and recall. The Gaussian Naive Bayes model exhibits exceptional recall, making it suitable for minimizing missed fraud cases.

The dissertation of Ziyang Zhang is approved.

Nicolas Christou

Oscar H. Madrid Padilla

Yingnian Wu, Committee Chair

University of California, Los Angeles

2024

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

Insurance fraud is a pervasive issue that poses significant financial and operational challenges to insurers worldwide. Within the realm of vehicle insurance, fraudulent claims not only contribute to substantial monetary losses but also undermine public trust and the overall integrity of the insurance industry. A study by the Federal Bureau of Investigation estimates that non-health insurance fraud costs more than $40 billion per year, with a significant portion stemming from fraudulent vehicle insurance claims [FBI22].

Traditionally, the detection of fraudulent vehicle insurance claims has relied heavily on manual investigations and the expertise of claims adjusters. However, this process is often time-consuming, labor-intensive, and susceptible to human error or bias. As the volume and complexity of claims data continue to grow, insurers are increasingly turning to advanced data analytics and machine learning techniques to enhance their fraud detection capabilities.

Machine learning offers a powerful toolset for identifying patterns, anomalies, and intricate relationships within large datasets that may be indicative of fraudulent behavior. By leveraging historical claims data and a wide range of features, such as policyholder information, vehicle details, and incident circumstances, machine learning models can be trained to accurately classify claims as legitimate or fraudulent.

While numerous studies have explored the application of machine learning in fraud detection across various domains, the unique characteristics of vehicle insurance claims present distinct challenges. These challenges include the inherent class imbalance, where fraudulent cases constitute a small minority compared to legitimate claims, as well as the presence of di-

verse and complex feature interactions that may not be easily discernible through traditional statistical methods.

This research aims to address these challenges by conducting a comprehensive evaluation of multiple machine learning models and data preprocessing techniques specifically tailored for vehicle insurance fraud detection. By employing state-of-the-art algorithms, such as logistic regression, random forests, XGBoost, and gradient boosting classifiers, this study seeks to identify the most effective approaches for accurately classifying fraudulent vehicle insurance claims.

Furthermore, to tackle the class imbalance issue, which can significantly impact the performance of machine learning models, this research explores various oversampling strategies, including the Synthetic Minority Over-sampling Technique (SMOTE), Borderline SMOTE, and Adaptive Synthetic Sampling (ADASYN). These techniques aim to balance the class distributions and enhance the models' ability to learn from the minority class, ultimately improving their overall predictive power.

The findings of this study have practical implications for the insurance industry by providing insights into effective machine learning methods for vehicle insurance fraud detection. Insurers can leverage these methods to mitigate financial losses, streamline claims processing, and allocate investigative resources more efficiently.

CHAPTER 2

Exploratory Data Analysis

2.1 Data Set

For this study, we use the dataset titled "Vehicle Insurance Fraud Claim Detection," which is hosted on Kaggle. This dataset is specifically designed to facilitate the detection of fraudulent activities in vehicle insurance claims. It encompasses a variety of features that describe the insurance policyholder, the insured vehicle, and details of the insurance claims. The dataset comprises 15,420 records, each representing an individual insurance claim. Each record includes multiple attributes such as:

1.Policy Attributes: Policy number, policy deductible.

2.Insured Information: Age, sex, marital status,.

3.Incident Information: Incident type, collision type, authorities contacted, incident location, incident hour of the day.

4.Vehicle Attributes: Vehicle make, vehicle type, year.

5.Claim Attributes: whether the claim was fraudulent

This data provides a comprehensive overview of the factors that may influence fraudulent activities in vehicle insurance claims. The target variable, Fraud Reported, indicates whether a claim was identified as fraudulent, offering a clear endpoint for our analysis. Notably, the dataset exhibits a significant class imbalance, with fraudulent claims constituting merely 6% of the total entries.

2.2 Preparing data for modeling

2.2.1 Data Cleaning

Upon conducting an initial data cleaning process, it was observed that the dataset does not contain missing values in the conventional sense; however, certain anomalies were detected that required attention. Notably, fields such as DayOfWeekClaimed, MonthClaimed, and Age presented entries with a value of zero, which are contextually invalid.

Instances with zero values in DayOfWeekClaimed and MonthClaimed categories were relatively few. Considering the impracticality of these values representing any day or month, these records were deemed erroneous and subsequently removed from the dataset. All records with an Age value of zero corresponded to policyholders aged between 16 and 17 years. Given the unlikely scenario of zero age in any practical context, these age entries were adjusted to 17 years, a more plausible representation within this age group.

Further analysis revealed that the PolicyType variable is a concatenation of VehicleCategory and BasePolicy. Given that this variable redundantly combines information already present in other fields, it was decided to remove PolicyType from the dataset to streamline the data and focus on independent variables.

2.2.2 Feature Engineering

To enhance model performance, it's crucial to appropriately transform the raw data. For instance, without adjustments, the wide range in vehicle prices (often tens of thousands) and the smaller scale of age (typically under 100) could disproportionately influence the model. If used directly, minor changes in vehicle claim amounts might unduly affect outcomes, potentially overshadowing other significant factors like age, which could also impact fraud detection. Hence, processing continuous numerical variables is necessary to ensure reliable model results and to maintain analytical balance.

To manage the wide disparities in scales and potential impacts of the features, the 'VehiclePrice' and 'Age' variables were categorized to ensure a more balanced model input. 'VehiclePrice' was segmented into bins ranging from 'less than 20000' (0) to 'more than 69000' (5), allowing for better normalization across different price levels. Similarly, 'Age' was divided into four groups: 18-25 (1), 26-40 (2), 41-65 (3), and over 65 (4).

Evidence indicates a notably higher propensity for insurance fraud among younger individuals, particularly those under the age of 25. Studies have shown that this demographic is less likely to perceive insurance fraud as a serious crime. Only about 65% of individuals aged 18 to 25 view insurance fraud as a crime, compared to over 95% of those aged 55 and above who acknowledge its severity [Ver23]. This significant discrepancy underscores the importance of incorporating age as a factor in our model, as it substantially influences perceptions and behaviors related to insurance fraud. This trend is also depicted in Figure 2.1:

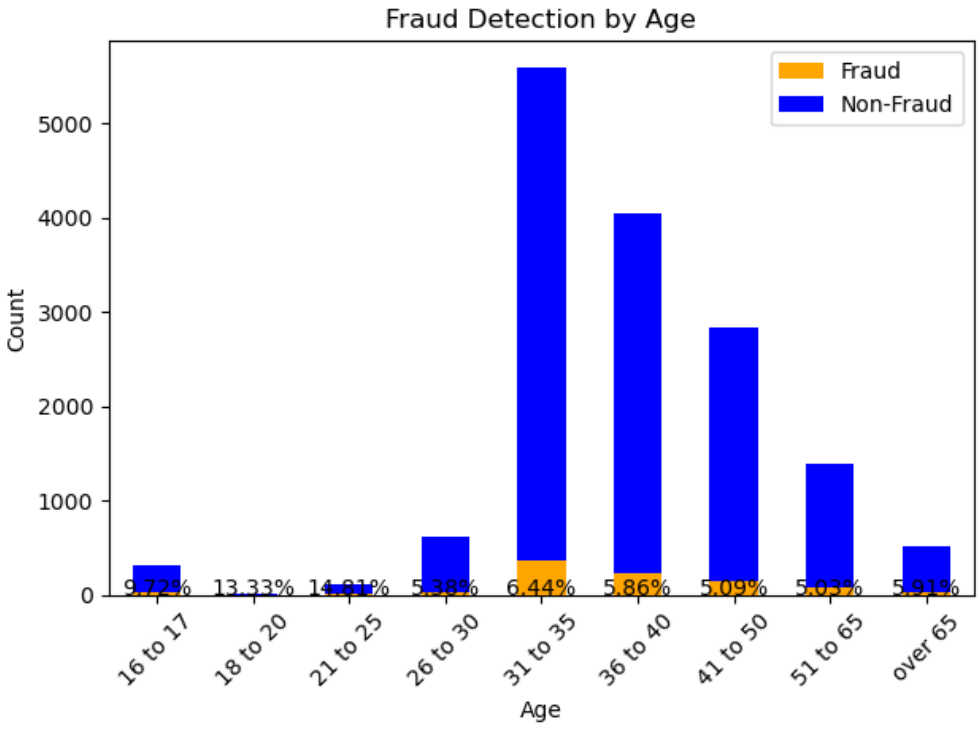

Figure 2.1: Fraud Detection by Age

In the data preparation phase, numerical encoding was applied to temporal features, namely the 'Month' and 'DayOfWeekClaimed' variables, to preserve their natural sequential order. This encoding allows the model to recognize and leverage temporal patterns, which are important for detecting fraudulent activities. For instance, the 'Month' variable was transformed by assigning a unique integer to each month, beginning with January as 1 and concluding with December as 12. Similarly, the days of the week were encoded from Monday as 1 through Sunday as 7. This approach ensures accurate interpretation by the model of the progression of days and months, potentially identifying trends such as increased incidents of fraud at certain times of the year or week. This systematic method not only boosts the model's capability to discern temporal patterns but also simplifies the dataset, facilitating more efficient algorithmic processing.

As shown in Figure 2, the dataset shows a marginally higher probability of fraudulent claims among males than females. To address this pattern, label encoding was employed for binary categorical variables. The 'Sex' feature was encoded with males as 1 and females as 0, allowing the model to directly assess the correlation between gender and fraud likelihood. Similarly, the 'AccidentArea' was categorized into Urban (1) and Rural (0) to further enhance the model's analytical precision.

Label encoding is a simple yet effective way to numerically represent categorical variables. It works by mapping each category to a numerical label, enabling machine learning models to process categorical data. This encoding not only reduces model complexity but also enhances feature interpretability. By preserving any inherent ordinal relationships within the categorical variables (if present), label encoding assists the algorithm in more accurately capturing data patterns, thereby improving fraud detection capabilities. Compared to more complex encoding techniques, such as one-hot encoding, label encoding offers the advantage of being concise and efficient while still retaining the original data's information effectively.

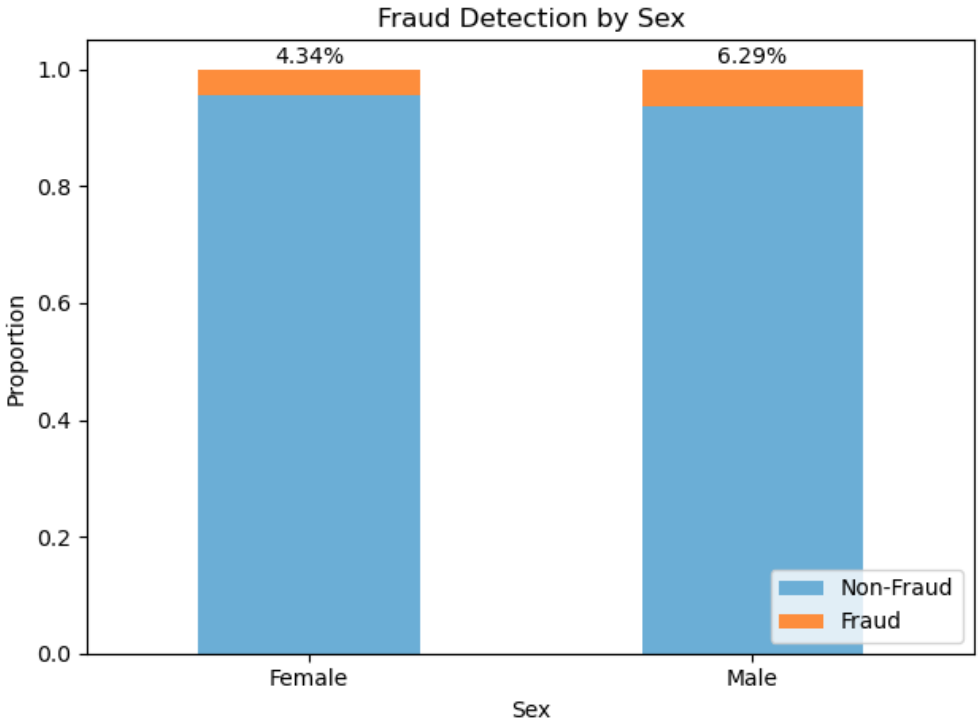

Figure 2.2: Fraud Detection by Sex

As depicted in Figure 2.3, the analysis of fraud probability across different marital statuses revealed significant variations.

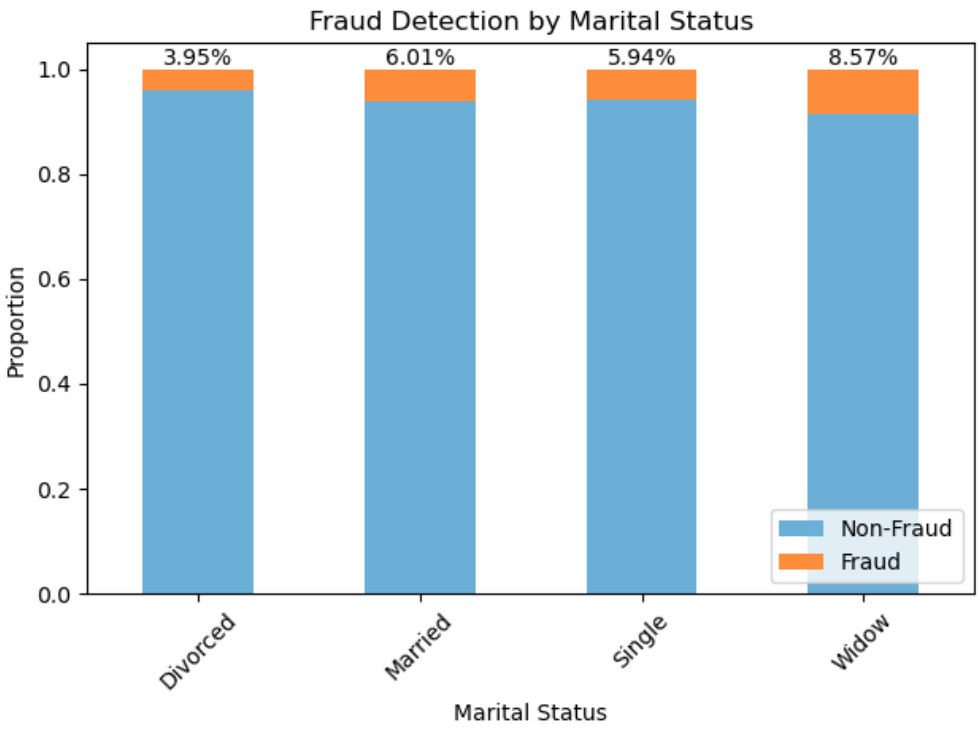

Figure 2.3: Fraud Detection by Marital Status

The data also revealed substantial discrepancies between different types of policyholders. Specifically, fraud rates among primary policyholders were recorded at 7.89%, while those involving third parties were significantly lower at 0.88%. This notable difference highlights potential areas for enhancing fraud prevention measures, particularly among primary policyholders, as illustrated in Figure 2.4.

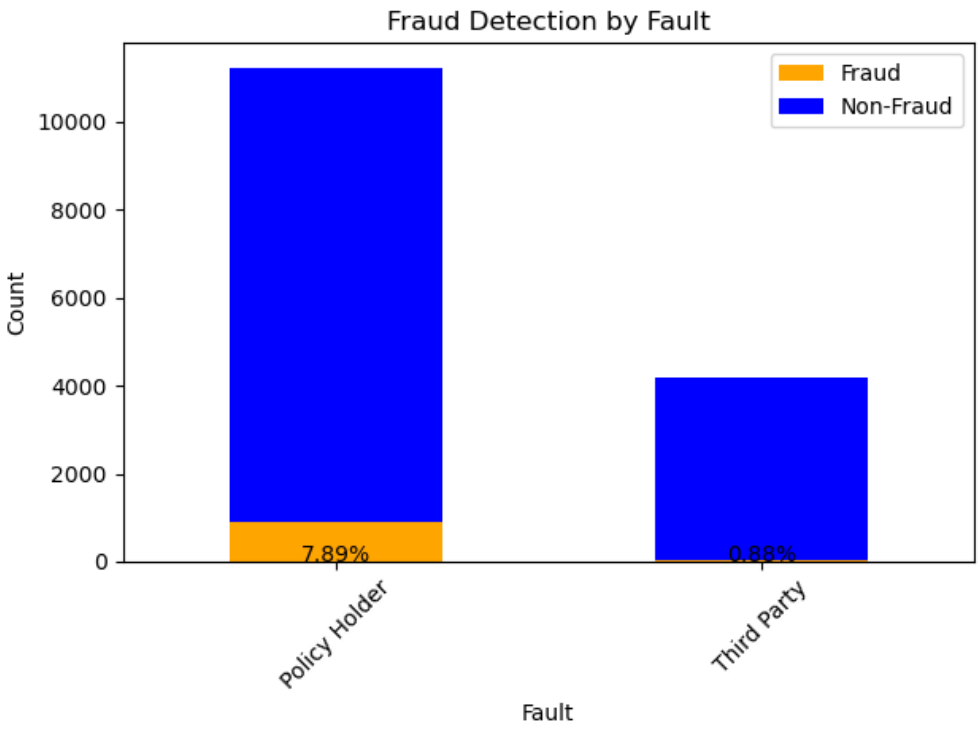

Figure 2.4: Fraud Detection by Fault

Primary policyholders, as direct contract holders with insurance companies, often have greater familiarity with the nuances of their policies and might be more adept at exploiting loopholes to commit fraud. In contrast, third parties typically lack direct access to the policy details and are less incentivized to commit fraud due to the absence of a direct financial relationship with the insurer.

As shown in Figure 2.5, further analysis of vehicle categories indicates that sports vehicles have a fraud rate of 1.57%, which is significantly lower than the 8.22% observed for sedans. The sample sizes for these categories are comparable, highlighting the crucial role that vehicle type plays in the likelihood of fraud occurring.

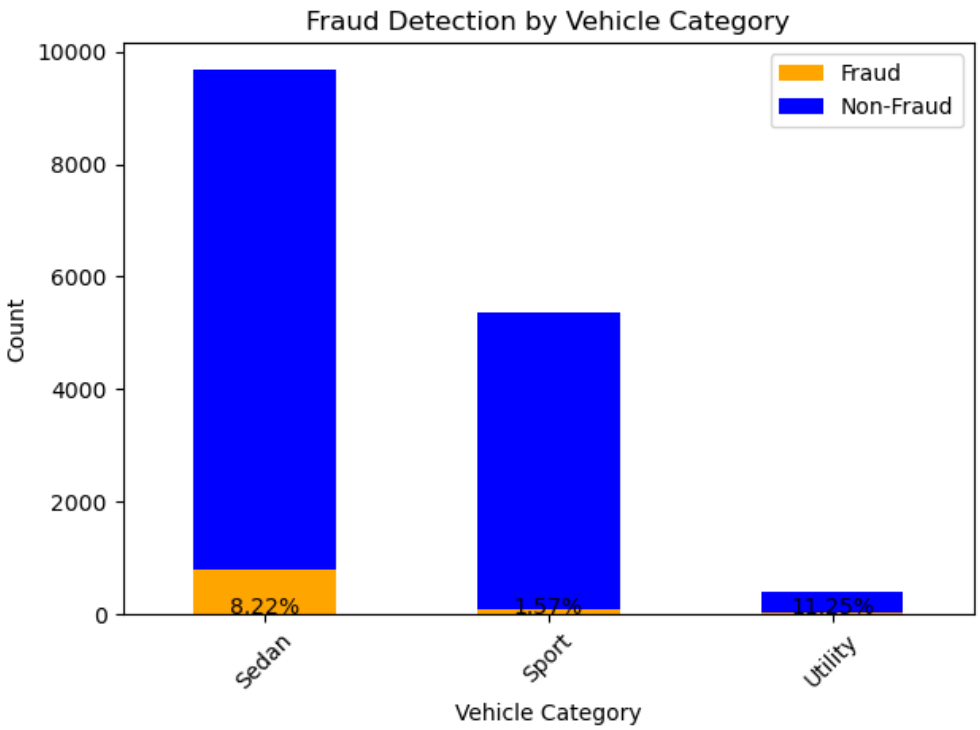

Figure 2.5: Fraud Detection by Vehicle Category

In the analysis of the "AddressChange" feature, as illustrated in the Figure 2.6, it's noted that individuals who changed their address within the past six months exhibit a high fraud rate of 75%. However, this observation is based on a very small sample size of only 4 cases, compared to 14,323 cases involving individuals who did not change their address. This stark difference in sample sizes means that conclusions about the impact of address changes on the likelihood of fraud should be approached with caution, as the data does not robustly support a definitive correlation.

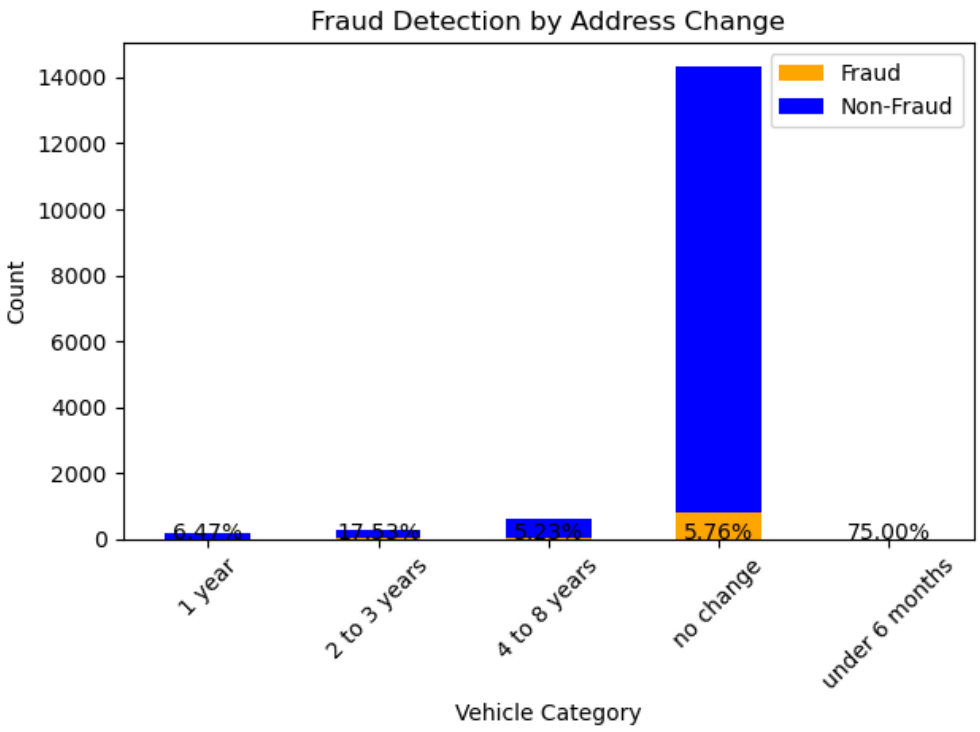

Figure 2.6: Fraud Detection by Address Change

As shown in Figure 2.7, the fraud rates vary significantly among different types of claims: 'Liability' claims exhibit a notably low fraud rate of 0.72%, compared to 'All Perils' at 10.16% and 'Collision' at 7.3%. The similar sample sizes across these categories ensure that the comparison is statistically valid. The much lower fraud rate in 'Liability' claims can be attributed to the nature of 'Liability' insurance, which typically covers damages or injuries that the policyholder causes to others. This type of claim might be harder to falsify compared to claims under 'Collision' and 'All Perils' coverage, which often deal with the policyholder's own losses.

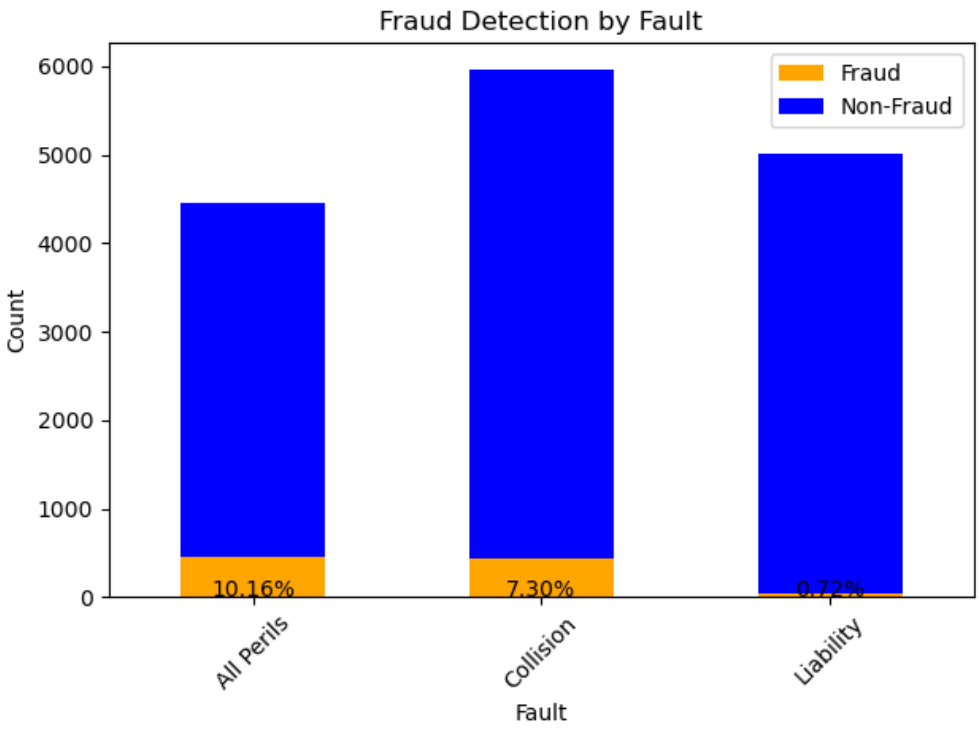

Figure 2.7: Fraud Detection by Fault

As illustrated in Figure 2.8, the analysis of the 'Past Number of Claims' feature reveals a trend where the fraud rate decreases as the number of previous claims increases: no previous claims show a fraud rate of 7.79%, one claim has a rate of 6.21%, two to four claims are at 5.36%, and more than four claims drop to 3.38%. This pattern suggests that individuals with a higher number of past claims are less likely to commit fraud, possibly due to increased scrutiny from insurance companies for repeat claimants, or perhaps these individuals become more risk-averse after undergoing multiple claims processes.

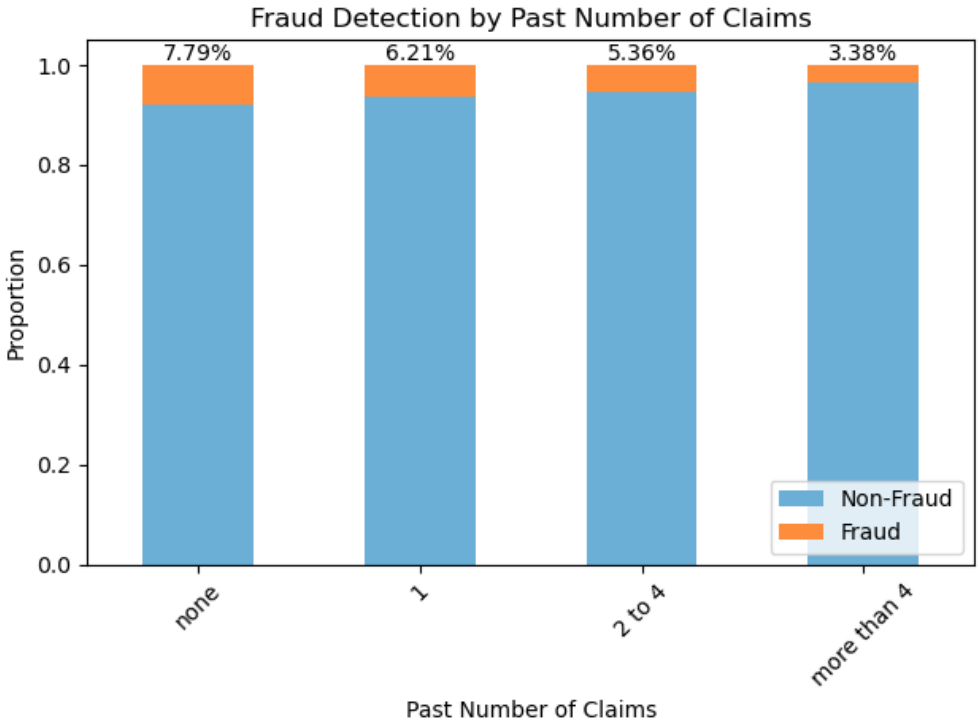

Figure 2.8: Fraud Detection by Past Number of Claims

Analysis of the variables Days Policy Accident and Days Policy Claim revealed that over 98% of the claims occurred more than 30 days after the policy was issued. A proportional analysis across categories—'1 to 7', '8 to 15', '15 to 30', and 'more than 30' days—indicated no significant difference in the likelihood of fraud, leading to their removal for simplicity.

The variable Make, representing the brand of the vehicle, was also assessed. Figure 2.9 and Figure 2.10 from the data showed that some brands had a higher number of fraudulent claims. However, these figures were proportional to the higher incidence of these brands in the dataset, suggesting no inherent link between brand and fraud likelihood. Furthermore, the high diversity of values within this variable complicated data processing and was not conducive to model accuracy, resulting in its exclusion from further analysis.

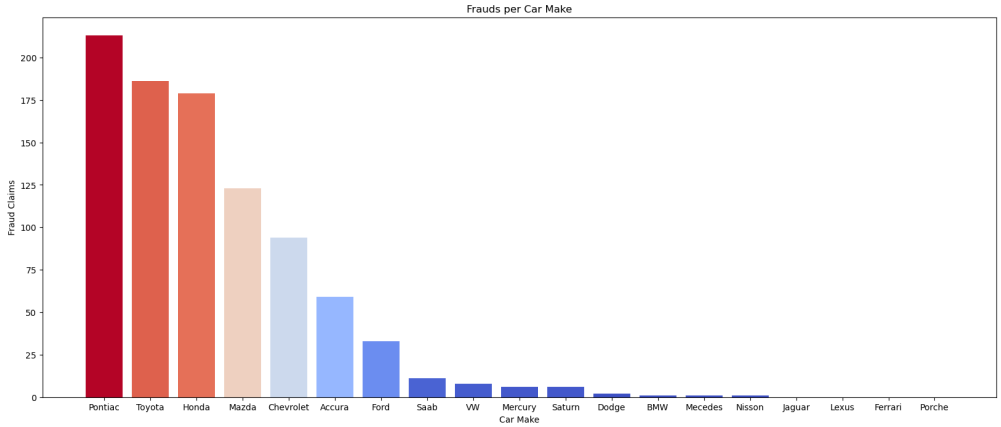

Figure 2.9: Frauds per Car Make

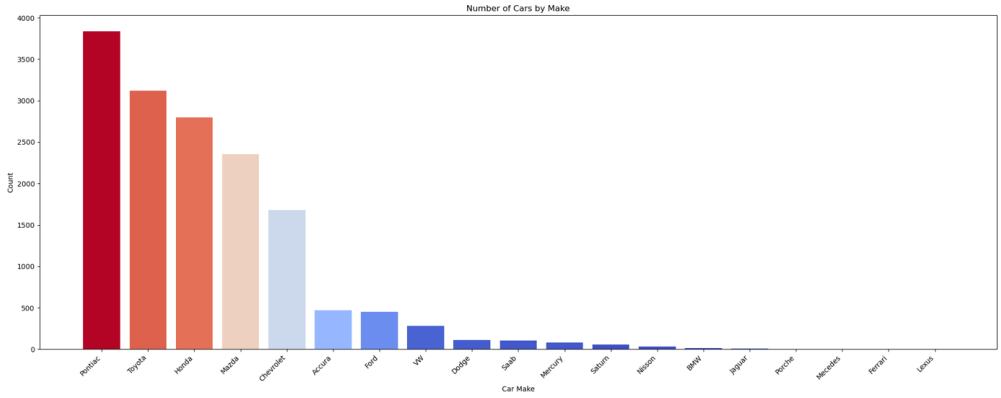

Figure 2.10: Number of Cars by Make

While some feature processing methods are quite similar and thus not extensively showcased here, all features have been transformed into numerical formats after the processing steps. This conversion facilitates easier computation and model training. The complete transformations and the final formats of the features are detailed in Table 2.1.

Table 2.1: Unique Values by Column

Column	Unique_Values
Month	12, 1, 10, 6, 2, 11, 4, 3, 8, 7, 5, 9
WeekOfMonth	5, 3, 2, 4, 1
DayOfWeek	3, 5, 6, 1, 2, 7, 4
AccidentArea	1, 0
DayOfWeekClaimed	2, 1, 4, 5, 3, 6, 7
MonthClaimed	1, 11, 7, 2, 3, 12, 4, 8, 5, 6, 9, 10
WeekOfMonthClaimed	1, 4, 2, 3, 5
Sex	0, 1
MaritalStatus	2, 3, 4, 1
Age	1, 2, 3, 5, 4
Fault	0, 1
VehicleCategory	1, 3, 2
VehiclePrice	5, 1, 2, 0, 3, 4
FraudFound	0, 1
Deductible	0, 1, 2, 3
DriverRating	1, 4, 3, 2
PastNumberOfClaims	0, 1, 3, 5
AgeOfVehicle	2, 5, 6, 7, 4, 0, 3, 1
AgeOfPolicyHolder	2, 3, 1, 4
PoliceReportFiled	0, 1
WitnessPresent	0, 1
AgentType	0, 1
NumberOfSuppliments	0, 6, 4, 1
AddressChange_Claim	1.0, 0.0, 4.0, 2.0, 0.5
NumberOfCars	3, 1, 2, 7, 9
Year	0, 1, 2
BasePolicy	1, 2, 3

2.2.3 Feature Selection

Several variables were identified as irrelevant to the prediction of insurance fraud and subsequently removed from the dataset to streamline the analysis. Specifically, PolicyNumber and RepNumber were excluded due to their lack of correlation with fraudulent activity. Additionally, the PolicyType variable, which appeared to be a concatenation of VehicleCategory and BasePolicy, was deemed redundant and therefore eliminated.

Based on the correlation analysis performed with respect to the 'FraudFound' variable, detailed in Table 2.2, I selected features that have a correlation coefficient greater than 0.02 for inclusion in the model. This threshold was chosen to ensure that the features included are likely to have a significant impact on the model's ability to predict fraud, while excluding features with negligible relationships to reduce model complexity and potential overfitting.

Table 2.2: Feature Importance for Fraud Detection

Feature	FraudFound
BasePolicy	0.1571103145945941
VehicleCategory	0.135620477209498
Fault	0.1314107989324906
PastNumberOfClaims	0.0573612666072198
AccidentArea	0.03357274880743924
AgeOfVehicle	0.033312627812110474
NumberOfSuppliments	0.03260915172104851
VehiclePrice	0.031716474447517844
Sex	0.02996062638299658
AgeOfPolicyHolder	0.029165574517125606
MonthClaimed	0.02898223561418115
Month	0.02727555107811587
Age	0.025765528676146312
Deductible	0.025277719797434296
Year	0.02477830121562461
AgentType	0.0229803252664285
DayOfWeek	0.017429835207084115
AddressChange_Claim	0.016472963394018037
PoliceReportFiled	0.01601017752105214
WeekOfMonth	0.01187166934971578
WitnessPresent	0.00808545420374933
DayOfWeekClaimed	0.00797394215784522
DriverRating	0.007259032798237225
NumberOfCars	0.007242908295227828
WeekOfMonthClaimed	0.00578338907885248
MaritalStatus	0.00362094088212295

From the correlation heatmap, a high correlation of 0.82 can be observed between ‘BasePolicy‘ and ‘VehicleCategory‘, indicating a strong linear relationship between these two features. This likely suggests that the type of base policy chosen is closely tied to the category of the insured vehicle. For instance, specific policies may be more common or tailored for certain types of vehicles, such as luxury cars, sports cars, or utility vehicles.

Due to the high redundancy between ‘VehicleCategory‘ and ‘BasePolicy‘, the ‘VehicleCategory‘ feature was removed to simplify the model and reduce the impact of multicollinearity. By eliminating highly correlated redundant features, not only can model complexity be reduced, but it also helps to improve the model’s generalization ability and robustness.

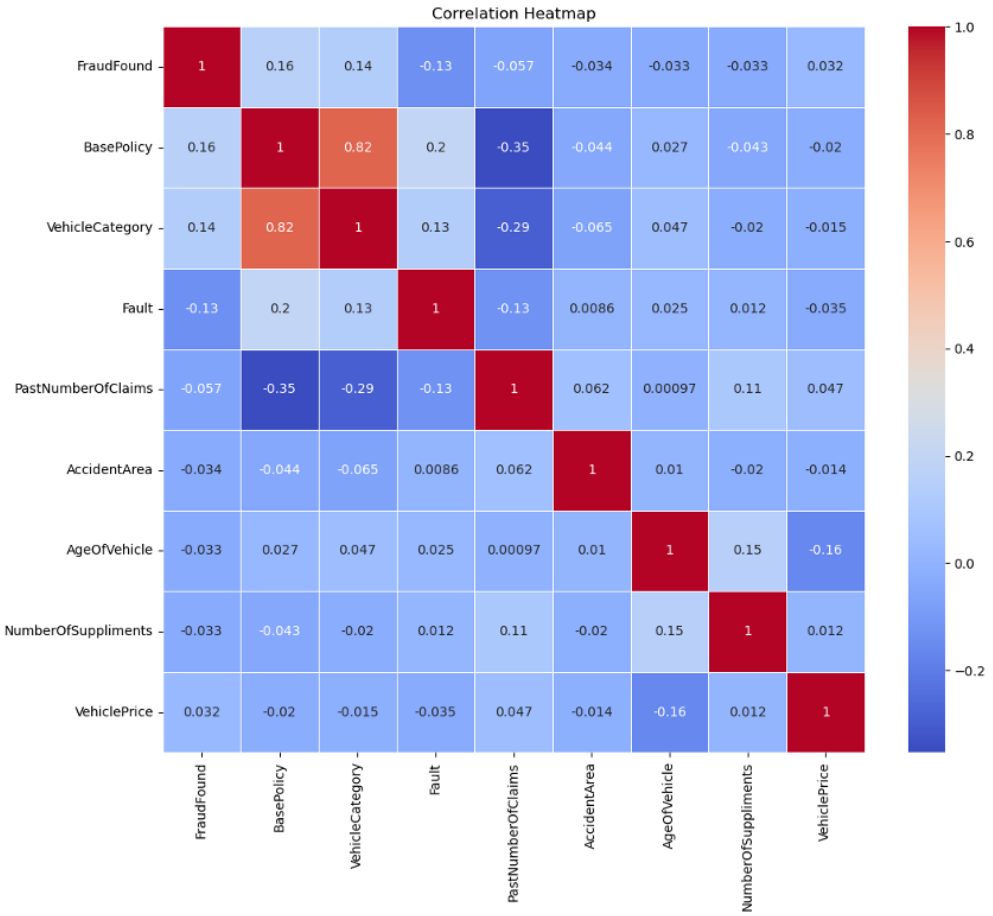

Figure 2.11: Correlation Heatmap

CHAPTER 3

Methodology

3.1 Model Introduction

3.1.1 Logistic Regression

Principle: Logistic Regression is a statistical model used in machine learning for binary classification tasks. It predicts the probability that a given input belongs to a particular category (usually labeled as 0 or 1) by applying the logistic function. The model estimates the probabilities using a logistic function, which is an S-shaped curve that can take any real-valued number and map it into a value between 0 and 1, but never exactly at those limits.

Formula: The logistic regression model calculates the probability p that an observation belongs to the class 1 using the logistic function:

$$p = \frac{1}{1 + e^{-z}} \tag{3.1}$$

. where z is a linear combination of the input features x, weighted by the coefficients β, plus an intercept β_0

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n \tag{3.2}$$

. Advantages and Disadvantages: Logistic Regression is highly valued for its straightforward implementation and the clarity of its output, particularly in scenarios that require a

probabilistic perspective for binary decision-making. Its computational efficiency allows for quick model training and prediction, making it ideal for applications with less complex data structures. However, the model's effectiveness is constrained by its assumption of a linear relationship between predictors and the logit of the outcome. This assumption can lead to underperformance in cases involving nonlinear relationships or high-dimensional feature spaces. Furthermore, logistic regression may exhibit biases when faced with imbalanced datasets, necessitating additional techniques to handle such scenarios effectively.

3.1.2 Random Forest Classifier

Principle: The Random Forest Classifier is an ensemble learning method used for classification (and regression) tasks that operates by constructing multiple decision trees during training and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. It integrates the simplicity of decision trees with flexibility, resulting in higher accuracy without substantial increase in complexity. Random forests correct for decision trees' habit of overfitting to their training set by averaging multiple trees, which are trained on different parts of the same training set.

Formula: The Random Forest algorithm doesn't involve a straightforward mathematical formula like linear models. Instead, it builds upon the idea of decision trees and bagging. The final prediction is made based on the majority vote (in classification tasks) from all the trees. The conceptual formula for the classification performed by a Random Forest can be described as follows:

$$RandomForestPrediction = mode(\{T_1(x), T_2(x), \ldots, T_n(x)\}) \tag{3.3}$$

. Where $T_i(x)$ represents the prediction of the i-th decision tree for input x.

Advantages and Disadvantages: Random Forest Classifier excels in handling high-dimensional data and can model complex relationships without requiring feature scaling. Its ensemble

nature makes it robust against overfitting, which is a common pitfall in decision tree models. Moreover, it is capable of handling both numerical and categorical data and provides feature importance scores, which are helpful in understanding the input features' influence on the model prediction.

Despite its strengths, the Random Forest model can be computationally intensive, which may lead to longer training times, especially with large datasets. The model's complexity can also make it harder to interpret compared to a simple decision tree. Additionally, if the data includes categorical variables with a large number of levels, Random Forests might become biased in their favor, potentially leading to overfitting. Random Forests also require careful tuning of parameters like the number of trees and tree depth to avoid underfitting or overfitting, depending on the complexity of the data and the signal-to-noise ratio.

3.1.3 Gaussian Naive Bayes

Principle: Gaussian Naive Bayes is a variant of the Naive Bayes algorithm that is particularly suited for continuous data that is assumed to be normally distributed. This classifier applies Bayes' theorem, assuming strong (naive) independence between the features. For each class, it calculates the likelihood of the data assuming that the input features are normally distributed, which simplifies computation and makes it efficient for high-dimensional datasets.

Formula: In Gaussian Naive Bayes, the conditional probability of a feature xi given a class c is modeled using the Gaussian (normal) distribution. The probability density function for the Gaussian distribution is given by:

$$p(x_i \mid c) = \frac{1}{\sqrt{2\pi\sigma_c^2}} \exp\left(-\frac{(x_i - \mu_c)^2}{2\sigma_c^2}\right) \tag{3.4}$$

. Where μ_c and the variance σ_c^2 are the mean and the variance of the feature x_i for class c, respectively. The overall probability of a data point $x = (x_1, x_2, ..., x_n)$ belonging to class c

is then computed as:

$$p(c \mid x) \propto p(c) \prod_{i=1}^{n} p(x_i \mid c) \tag{3.5}$$

. Here, p(c) is the prior probability of class c.

Advantages and Disadvantages: Gaussian Naive Bayes is incredibly fast and easy to implement, making it highly suitable for applications that require real-time predictions. Its simplicity also allows for very good performance in cases where the assumption of independence holds reasonably well. Furthermore, it requires a small amount of training data to estimate the necessary parameters (mean and variance).

However, the naive assumption of feature independence can lead to significant performance issues if the true underlying relationships are ignored, especially in cases where features are correlated. This model also tends to perform poorly if the Gaussian distribution assumption does not hold for the numeric features. Additionally, it can be particularly sensitive to data with zero variance (features with the same value across all samples), which can skew the probability estimates.

3.1.4 Decision Tree Classifier

Principle: A Decision Tree Classifier is a non-parametric supervised learning method used for classification and regression tasks. It models decisions and their possible consequences as a tree structure, consisting of nodes that represent tests on attributes, branches that represent the outcome of those tests, and leaf nodes that represent class labels or target values. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.

Formula: While decision trees do not rely on mathematical formulas in the traditional sense, they use algorithms to split the data at each node based on specific criteria. The most common criteria for splitting include Gini impurity and entropy (information gain), which

can be represented as follows:

- Gini Impurity:

$$G(p) = 1 - \sum_{i=1}^{k} p_i^2 \tag{3.6}$$

. • Entropy (Information Gain):

$$H(S) = -\sum_{i=1}^{k} p_i \log_2(p_i) \tag{3.7}$$

. Where S is the total set of samples, p_i is the proportion of the samples that belong to class i, and k is the number of classes. The decision at each node is made to maximize the information gain — the reduction in entropy or impurity after the split.

Advantages and Disadvantages: Decision Tree Classifier is renowned for its interpretability, as it provides a clear visualization of decision-making processes, making it highly accessible for analysis. The model handles both numerical and categorical data effectively and manages non-linear relationships, which enhances its versatility in practical applications. Furthermore, decision trees do not require data normalization or scaling, simplifying the preprocessing steps.

However, the model has several drawbacks, notably its propensity for overfitting, especially when the trees grow deep with many branches. This overfitting can often be mitigated by pruning the tree, setting minimum sample sizes for node splitting, or limiting the tree's depth. Another significant issue is the high variance of decision trees; small changes in the dataset can lead to drastically different tree structures, which can undermine the model's reliability. Additionally, decision trees can develop a bias towards attributes with more levels, necessitating careful balancing of the dataset to ensure fair and effective decision-making.

3.1.5 XGBoost Classifier

Principle: XGBoost (Extreme Gradient Boosting) is an advanced implementation of gradient boosting algorithms. It is designed to be highly efficient, flexible, and portable. XGBoost systematically constructs a series of decision trees in a sequential manner, where each subsequent tree aims to correct the errors made by the previous ones. The method combines the output of multiple weak learners (typically decision trees) to produce a strong predictive model. XGBoost optimizes both the computational speed and model performance by implementing hardware optimizations and efficient handling of sparse data.

Formula: XGBoost uses the gradient boosting framework, where the model is trained to minimize a loss function. Each tree is added to minimize the following objective function, which is a combination of a loss term and a regularization term:

$$L(\phi) = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t)}) + \sum_{k=1}^{t} \Omega(f_k) \tag{3.8}$$

.

Where:

- $l(y_i, \hat{y}_i^{(t)})$ is a differentiable loss function that measures the difference between the predicted $\hat{y}_i^{(t)}$ and actual $y_i^{(t)}$ values.
- $\Omega(f_k)$ is the regularization term (typically used to penalize the complexity of the model), and f_k represents the k-th tree.
- t denotes the number of trees.

The gradient boosting process updates the model using an additive strategy:

$$\hat{y}_i^{(t+1)} = \hat{y}_i^{(t)} + \eta \cdot f_t(x_i) \tag{3.9}$$

Where η is the learning rate that scales the contribution of each tree.

Advantages and Disadvantages: XGBoost Classifier is celebrated for its superior performance and scalability, making it a preferred choice for many machine learning competitions. It effectively handles a variety of data types and integrates smoothly into diverse data science workflows. One of its notable strengths is its regularization feature, which significantly reduces the risk of overfitting—a common drawback in many other boosting algorithms. Additionally, XGBoost offers extensive flexibility, allowing users to define custom objectives and fine-tune the tree splitting criteria to best suit their specific needs.

However, the complexity of the model makes it less interpretable compared to simpler alternatives, which can be a drawback when explanation or transparency of the decision-making process is required. Furthermore, despite its optimization for performance, XGBoost can be computationally demanding, particularly with large datasets or an extensive number of trees. Achieving optimal performance also demands careful parameter tuning, which can be intricate and time-consuming, especially for those not well-versed in the nuances of gradient boosting techniques.

3.1.6 Gradient Boosting Classifier

Principle: The Gradient Boosting Classifier is a popular machine learning technique that builds on the concept of boosting, where multiple weak learners (typically decision trees) are trained sequentially to correct the errors of the previous learners. Each new learner focuses more on the instances that were misclassified or had larger errors by the previous ones. This process continues until a predefined number of learners are created or improvements become negligible, creating a strong overall model from many weak ones. The key concept behind gradient boosting is to use the gradient of the loss function, which guides the algorithm on how to effectively reduce errors in subsequent models.

Formula: The Gradient Boosting Classifier minimizes a loss function over each iteration by adding weak learners that predict the gradients of the loss. Mathematically, this is expressed as:

$$F_t(x) = F_{t-1}(x) + \rho_t h_t(x)$$

Where:

- $F_t(x)$ is the boosted model at iteration t.
- $F_{t-1}(x)$ is the boosted model from the previous iteration.
- ρ_t is the learning rate which scales the contribution of each weak learner.
- $h_t(x)$ is the weak learner added at iteration t.

Each weak learner h_t is fitted to the negative gradient of the loss function evaluated at F_{t-1}, and ρ_t is typically determined by line search.

Advantages and Disadvantages: The Gradient Boosting Classifier offers remarkable flexibility and is highly effective for both classification and regression tasks, capable of optimizing various differentiable loss functions. Its strength lies in its predictive power, often outperforming other algorithms, particularly in structured data scenarios. Additionally, it adeptly handles heterogeneous features, making it suitable for a wide range of applications

However, the training process for a Gradient Boosting model is computationally intensive and can be quite time-consuming, as it builds numerous models sequentially. There is also a significant risk of overfitting, especially if the model is not properly tuned or if the dataset is noisy. Furthermore, the effectiveness of Gradient Boosting is heavily dependent on the correct setting of its parameters, such as the number of trees, tree depth, and learning rate. This sensitivity necessitates careful parameter tuning, often requiring extensive cross-validation or grid search, which can complicate the training process and increase the time and resources needed to develop an effective model.

3.2 Handling Imbalanced Data

3.2.1 Under-sampling

Under-sampling involves reducing the size of the more prevalent class in an imbalanced dataset. This technique achieves a more balanced class distribution by randomly removing samples from the dominant class. Consider the dataset where 94% of the samples are non-fraudulent transactions and only 6% are fraudulent. Under-sampling would reduce the number of non-fraudulent transactions to match the fraudulent ones. However, this method may not be ideal in scenarios where the minority class constitutes a small percentage and the total number of samples is limited. Removing significant portions of the majority class could result in the loss of critical information, potentially leading to underfitting and a decrease in model performance.

3.2.2 Over-sampling

Over-sampling adjusts the class distribution by increasing the size of the underrepresented class through replication of existing minority class samples. For example, in a dataset with 100 fraudulent transactions and 1,900 non-fraudulent transactions, over-sampling might replicate the fraudulent transactions to balance the dataset. This method helps to preserve all original information from the minority class but might lead to overfitting since it increases the likelihood of the model learning noise from the replicated samples.

SMOTE, or Synthetic Minority Over-sampling Technique, addresses the over-sampling problem by generating synthetic samples instead of duplicating existing ones. By choosing two or more similar instances of the minority class and interpolating new instances between them, SMOTE creates a more diverse and extensive feature space [CBH02]. For instance, if there are minority class samples such as two fraudulent transactions with slightly different transaction amounts or times, SMOTE would generate a new fraudulent transaction that

combines features of these existing transactions. This approach allows for a broader and more general decision boundary, potentially enhancing classification performance.

Borderline SMOTE refines the SMOTE algorithm by focusing synthetic sample generation specifically on the minority class samples near the decision boundary. This method first identifies minority class samples that are close to the majority class—often those most likely to be misclassified—and then generates new samples around these 'borderline' instances [HWM05]. By focusing on these critical areas, Borderline SMOTE helps to improve model sensitivity and specificity around the decision boundary, effectively enhancing classifier performance on hard-to-distinguish samples.

3.2.3 Synthetic Sampling

Synthetic sampling involves creating artificial data points based on the characteristics of existing data. Unlike simple replication in over-sampling, this method uses statistical techniques to generate new data points that are plausible yet not exact copies of existing ones. This technique enhances the diversity of the training dataset, which is crucial for capturing complex patterns in the data and avoiding model overfitting.

Adaptive Synthetic Sampling, or ADASYN, is an advanced form of synthetic sampling that focuses on generating synthetic data for minority class samples according to their learning difficulty. The algorithm adjusts the number of synthetic samples for each minority instance proportionally to the level of classification difficulty [HBG08]. For instance, if certain fraudulent transactions are frequently misclassified, ADASYN will generate more synthetic samples of these types to force the classifier to focus more on these challenging areas. This adaptive approach promotes a more balanced learning process and helps improve the predictive accuracy across varied instances within the minority class.

Table 3.1: Model Evaluation Metrics

Prediction/Actual	Fraudulent	Normal
Fraudulent	TP	FN
Normal	FP	TN

3.3 Criteria to Measure Performance

3.3.1 Accuracy

Accuracy, which measures the proportion of correct predictions (both true positives and true negatives) among all cases, can be deceptive in datasets with imbalanced classes. For example, in the dataset where fraudulent transactions make up only 6% of the total, a model that predicts all transactions as non-fraudulent would still achieve an accuracy of about 94%. Despite this high accuracy, the model completely fails to identify any fraudulent transactions, which are critical to detect. Thus, while accuracy provides a basic measure of a model's overall performance, it is not a reliable metric in scenarios with significant class imbalances, where the detection of the minority class is crucial.

The formula for accuracy is given by:

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

3.3.2 Precision

Precision measures the proportion of positive identifications that were actually correct and is crucial when the cost of a false positive is high.

The formula for precision is:

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

For Example, in email spam detection, precision is critical because misclassifying regular emails as spam (false positives) can be more disruptive than missing a spam email (false negative). High precision ensures that almost all emails marked as spam are truly spam, minimizing inconvenience to users.

3.3.3 Recall

Recall indicates the ability of a model to find all relevant cases within a dataset, which is vital when the cost of missing a positive (false negative) is significant.

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

In fraud detection, a high recall rate is crucial because failing to detect a fraudulent transaction (false negative) can have serious financial implications. Therefore, it is more acceptable to endure some false positives (innocent transactions flagged as possible fraud) than to allow actual fraud to go undetected.

3.3.4 F1 Score

F1 Score is the harmonic mean of precision and recall and is used when seeking a balance between precision and recall.

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

For example, in a scenario where both false positives and false negatives are costly, such as in legal document classification, the F1 score provides a more balanced view of the model's performance than using precision or recall alone.

3.3.5 AUC

AUC measures the entire two-dimensional area underneath the entire Receiver Operating Characteristic (ROC) curve from (0,0) to (1,1). It provides an aggregate measure of performance across all possible classification thresholds, distinguishing between classes effectively:

$$\text{AUC} = \int_0^1 \text{ROC Curve}\, dx$$

These metrics provide a comprehensive framework for assessing the effectiveness of classification models, aiding in the comparative evaluation and optimization of algorithms, especially in scenarios where different types of classification errors carry different costs.

CHAPTER 4

Models

4.1 Model Establishment

To tackle this vehicle insurance fraud detection task, we employed six different machine learning models: Logistic Regression, Random Forest Classifier, Gaussian Naive Bayes, Decision Tree Classifier, XGBoost Classifier, and Gradient Boosting Classifier. To address the class imbalance present in the dataset, we utilized three oversampling techniques: SMOTE, Borderline SMOTE, and ADASYN, generating four dataset variants. Consequently, a total of 24 model-dataset combinations were evaluated and compared.

4.2 Results Analysis and Comparison

4.2.1 Evaluation Metrics Selection

When evaluating the performance of models for this highly imbalanced vehicle insurance fraud detection task, we should not place excessive emphasis on the accuracy metric. There are two primary reasons for this:

First, the dataset exhibits a significant class imbalance, with non-fraudulent transactions (negative class) accounting for 94% of the samples, while fraudulent transactions (positive class) constitute only 6%. In such a scenario, a model that consistently predicts the negative class, even without detecting any positive samples, would still achieve an accuracy of 94%. Evidently, relying solely on accuracy is inadequate for assessing a model's ability to identify

the minority positive class.

Second,the data shows that across all models, whether using the original imbalanced dataset or the balanced datasets generated through techniques like SMOTE, the accuracy values are comparable, failing to demonstrate a clear distinction or discriminative power. Therefore, accuracy is not an effective metric for evaluating the relative performance of different models on this task.

Model	Original Data	SMOTE	Borderline SMOTE	ADASYN
LogisticRegression	0.935473	0.656615	0.669585	0.657588
RandomForestClassifier	0.929313	0.756809	0.778534	0.755512
GaussianNB	0.761673	0.382944	0.419585	0.383593
DecisionTreeClassifier	0.924773	0.753891	0.772698	0.749027
XGBClassifier	0.935473	0.732166	0.764591	0.730545
GradientBoostingClassifier	0.935798	0.662451	0.660182	0.659533

Figure 4.1: Results of Model's Accuracy

Instead, we propose focusing on the following three key metrics:

4.2.2 F1 Score Analysis

The F1 score, being the harmonic mean of precision and recall, effectively balances these two important measures. For this imbalanced binary classification problem, the F1 score can provide a robust evaluation of a model's ability to identify the positive class (minority fraudulent transactions).

The data reveals that on the original imbalanced dataset, all models exhibit F1 scores close to zero, indicating an almost complete inability to recognize the minority positive class. However, after applying oversampling techniques like SMOTE to balance the data, the F1 scores of all models show a significant improvement, with the XGBoost and Gradient Boost-

ing models achieving the highest F1 scores of 0.242 and 0.247, respectively, demonstrating relatively superior performance.

Model	Original Data	SMOTE	Borderline SMOTE	ADASYN
LogisticRegression	0.000000	0.229818	0.236704	0.238095
RandomForestClassifier	0.052174	0.225207	0.235162	0.238384
GaussianNB	0.230366	0.167906	0.175115	0.165862
DecisionTreeClassifier	0.100775	0.224719	0.223699	0.230616
XGBClassifier	0.074419	0.242202	0.246888	0.243858
GradientBoostingClassifier	0.029412	0.245105	0.247126	0.242424

Figure 4.2: Results of Model's F1 Score

4.2.3 Recall Analysis

Recall reflects a model's ability to detect all positive class samples, which is a crucial metric for this insurance fraud detection task. A high recall is essential, as failing to identify fraudulent transactions can result in substantial economic losses for insurance companies.

The data shows that the Gaussian Naive Bayes model exhibits recall values exceeding 0.95 across all oversampled datasets, an outstanding performance indicating its capability to detect the vast majority of fraudulent transactions. If the business objective is to minimize the risk of missed fraud cases, the Gaussian Naive Bayes model can be considered a viable option.

Model	Original Data	SMOTE	Borderline SMOTE	ADASYN
LogisticRegression	0.000000	0.793970	0.793970	0.829146
RandomForestClassifier	0.030151	0.547739	0.527638	0.592965
GaussianNB	0.552764	0.964824	0.954774	0.949749
DecisionTreeClassifier	0.065327	0.552764	0.507538	0.582915
XGBClassifier	0.040201	0.663317	0.597990	0.673367
GradientBoostingClassifier	0.015075	0.849246	0.864322	0.844221

Figure 4.3: Results of Model's Recall

4.2.4 AUC Analysis

AUC provides an overall evaluation of a binary classification model's performance across various threshold settings, serving as a widely adopted comprehensive metric. The data shows that on both the original imbalanced dataset and the oversampled datasets, the XGBoost and Gradient Boosting models demonstrate relatively high and stable AUC values, ranging from 0.78 to 0.81.

Model	Original Data	SMOTE	Borderline SMOTE	ADASYN
LogisticRegression	0.795178	0.781621	0.784100	0.784121
RandomForestClassifier	0.760765	0.742019	0.750140	0.745782
GaussianNB	0.776833	0.769364	0.771064	0.763020
DecisionTreeClassifier	0.729206	0.709343	0.700245	0.712850
XGBClassifier	0.812549	0.783971	0.788200	0.781738
GradientBoostingClassifier	0.811478	0.798473	0.809541	0.795721

Figure 4.4: Results of Model's AUC

4.3 Comprehensive Analysis

Considering the F1 score, recall, and AUC collectively, the XGBoost Classifier and Gradient Boosting Classifier exhibit the most outstanding overall performance. They demonstrate a superior ability to balance precision and recall while maintaining a high overall classification level on this imbalanced dataset. Therefore, if the objective is to strike a balance between precision and recall, these two models can be prioritized for this task.

On the other hand, if the business goal is to minimize the risk of missed fraud cases, the Gaussian Naive Bayes model, with its exceptional recall performance, can also be considered a viable option, despite its relatively lower F1 score and AUC values.

CHAPTER 5

Limitations and Conclusion

While the proposed methods and models have demonstrated promising results in detecting fraudulent vehicle insurance claims, it is important to acknowledge some inherent limitations of this study.

Firstly, the analysis and conclusions drawn are primarily based on the specific dataset utilized, which may not fully represent the diverse range of scenarios encountered in real-world insurance fraud cases. Different geographic regions, insurance providers, or claim types could potentially exhibit varying patterns and characteristics, necessitating further validation and fine-tuning of the models.

Secondly, the feature engineering and selection processes employed in this study were guided by exploratory data analysis and domain knowledge. However, it is possible that certain relevant features were inadvertently overlooked or that more sophisticated feature extraction techniques could enhance the models' predictive capabilities further.

Thirdly, while measures were taken to address the class imbalance issue through over-sampling techniques, the synthetic data generation process may have introduced biases or artifacts that could impact the models' generalizability to unseen data. Additional techniques, such as ensemble methods or cost-sensitive learning, could be explored to further mitigate the effects of class imbalance.

This study highlights the potential of machine learning models, particularly XGBoost and Gradient Boosting Classifiers, in accurately identifying fraudulent vehicle insurance claims. By leveraging advanced algorithms and data preprocessing techniques, insurers can

enhance their fraud detection capabilities, reduce financial losses, and maintain the integrity of their operations. However, it is crucial to continually refine and adapt these models as new data becomes available and to integrate them into a broader risk management strategy that incorporates human expertise and domain knowledge.

REFERENCES

[CBH02] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. "SMOTE: synthetic minority over-sampling technique." *Journal of artificial intelligence research*, **16**:321–357, 2002.

[FBI22] FBI. "Insurance Fraud." https://www.fbi.gov, 2022.

[HBG08] H. He, Y. Bai, E. A. Garcia, and S. Li. "ADASYN: Adaptive synthetic sampling approach for imbalanced learning." In *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*, pp. 1322–1328. IEEE, 2008.

[HWM05] H. Han, W. Y. Wang, and B. H. Mao. "Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning." In *International conference on intelligent computing*, pp. 878–887, Berlin, Heidelberg, 2005. Springer.

[Ver23] Verisk and Coalition Against Insurance Fraud. "Survey Finds Younger Generations Have a Higher Tolerance For Insurance Fraud – What Insurers Should Know." *MehaffyWeber News*, Oct 2023.