VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)
CHENNAI

RISK AND FRAUD ANALYTICS (MGT3013)

REVIEW 3 - PROJECT REPORT

Automated Detection of Automobile Insurance Fraud Using Machine Learning with Real-Time Alert Notification System

By

SUSILASHA M - (21MIA1006)

THAYUMANAVAN T - (21MIA1007)

REVATHY P - 21MIA1016

HARISH A S - 21MIA1021

HARISH S - 21MIA1046

Fall Semester 2024-2025

18/11/2024

SUBMITTED TO

Dr. Manu Jose

VIT Business School

VIT CHENNAI

DECLARATION

I hereby declare that the thesis entitled "**Automated Detection of Automobile Insurance Fraud Using Machine Learning with Real-Time Alert Notification System** " submitted by **Susilasha M (21MIA1006), Thayumanavan T (21MIA1007), Revathy P (21MIA1016), Harish A S (21MIA1021), Harish S (21MIA1046),** for the award of the degree of Bachelor of Technology in Computer Science and Engineering, Vellore Institute of Technology, Chennai is a record of Bonafide work carried out by me under the supervision of Dr. Manu Jose.

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place: Chennai

Date: 18/11/2024 **Signature of the Candidate**

CERTIFICATE

This is to certify that the report entitled "**Automated Detection of Automobile Insurance Fraud Using Machine Learning with Real-Time Alert Notification System"** is prepared and submitted by **Susilasha M (21MIA1006), Thayumanavan T (21MIA1007), Revathy P (21MIA1016), Harish A S (21MIA1021), Harish S (21MIA1046)** to Vellore Institute of Technology, Chennai, in partial fulfillment of the requirement for the award of the degree of **Integrated MTech Computer Science and Engineering with Specialization in Business Analytics** is a Bonafide record carried out under my guidance. The project fulfills the requirements as per the regulations of this University and in my opinion meets the necessary standards for submission. The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma and the same is certified.

Name: Dr Manu Jose

Date: 18/11/2024 **Signature of the Guide:**

Abstract

Auto insurance fraud means the presentation of a claim to an automobile insurer for payment of an amount to which the presenter is not entitled under the policy, and usually entails an unlawful act such as the staging of an accident for compensation. People can embark on staging accidents with little or no injury, inflating or just lying about the circumstances of events so as to be compensated heavily. The insurance industry is at greatest risk by this form of fraud, given the large sums of money that the fraudsters, inflated costs of compensating the defaults, and high premiums for genuine policy holders. The nature of identifying fake claims often requires human effort and the current high processing of claims per day makes this a big problem to insurance companies.

This paper is concerned with establishing a supervised machine learning model that is capable of accurately predicting auto insurance claim fraud. In this project, an insurance claims dataset will be joined with an automobile accident dataset in a way to obtain a single dataset for analysis. Feature engineering is employed in order to identify correlations associated with fraudulent behavior. A RF classifier is then trained to differentiate between real and fake claims while an alert notification system will be implemented such that whenever the RF model predicts fraud, relevant stakeholders will be informed. This automatically build model offers an edge over the manual approach of fraud detection which increases the accuracy, decreases the time spent and does not put a massive hole in the insurance industry's pocket.

CONTENTS

1. Introduction

1.1 Background

The automobile insurance industry is a cornerstone of financial protection for individuals and businesses, offering coverage for accidents, theft, and other unforeseen events. However, the integrity of this industry is constantly threatened by the prevalence of fraudulent activities. Insurance fraud, including staged accidents, exaggerated damage claims, and falsified reports, imposes severe financial and operational burdens on insurers. Globally, fraudulent claims cost the insurance industry billions of dollars annually, with these costs often passed on to genuine customers in the form of higher premiums.

In India, the rapid growth of the automobile sector has led to an exponential increase in the number of insurance claims filed each year. Alongside this growth, a significant proportion of claims are suspected to be fraudulent, placing an immense strain on insurers. Fraudulent activities not only erode the financial health of insurers but also diminish trust in the system among legitimate policyholders. Customers who file genuine claims face delays and increased premiums, resulting in dissatisfaction and a loss of faith in the industry.

The traditional approach to fraud detection relies heavily on manual review processes and rule-based systems. While effective to some extent, these methods are labour-intensive, time-consuming, and increasingly inadequate in handling the growing volume and complexity of claims. As fraudsters continually evolve their techniques, static rules fail to adapt, leaving insurers vulnerable to new and sophisticated schemes.

1.2 Motivation

The need for an automated fraud detection system in the automobile insurance industry arises from the pressing challenges and inefficiencies associated with manual processes. Fraud detection is critical for several reasons:

1. ***Economic Necessity:***
   - Fraudulent claims directly impact insurers' profitability, reducing their ability to offer competitive premiums.
   - The financial burden of fraud is ultimately borne by honest customers through increased premiums, creating an uneven and unfair system.

2. ***Operational Efficiency****:*
   - The increasing volume of claims demands scalable solutions that can process data quickly and accurately.
   - Automated fraud detection reduces the reliance on human intervention, enabling insurers to allocate resources more effectively and prioritize genuine claims.
3. ***Customer Trust and Satisfaction****:*
   - By focusing on detecting fraudulent claims, insurers can expedite the processing of legitimate claims, fostering trust and loyalty among policyholders.
   - Enhanced fraud detection minimizes false accusations against honest customers, improving their overall experience.
4. ***Regulatory Compliance****:*
   - Governments and regulatory bodies emphasize transparency, fairness, and efficiency in insurance claim processing. Advanced fraud detection aligns with these goals and helps insurers maintain compliance.

1.3 Challenges

Detecting fraudulent claims is a complex and multifaceted challenge. Key issues include:

1. ***Volume and Complexity of Data****:*
   - Insurance companies process thousands of claims daily, each involving multiple data points such as policy details, accident reports, and repair estimates. Analysing such vast and complex datasets manually is neither feasible nor efficient.
2. ***Evolving Fraud Tactics****:*
   - Fraudsters continually refine their strategies to exploit vulnerabilities in existing systems. Staged accidents, coordinated networks of fraudsters, and fabricated documentation are becoming increasingly sophisticated, making it harder to detect fraud using traditional methods.
3. ***False Positives and Negatives****:*
   - Rule-based systems often flag legitimate claims as fraudulent, leading to customer dissatisfaction and resource wastage. Conversely, some fraudulent claims go undetected, resulting in financial losses for insurers.

4. ***Data Quality Issues****:*
   - Inconsistent, incomplete, or erroneous data is a significant obstacle to effective fraud detection. Machine learning models require clean and structured data for accurate predictions.
5. ***Resource Constraints****:*
   - Many insurers lack the technological infrastructure and expertise required to implement advanced fraud detection systems.

1.4 Proposed Solution

This project develops a machine learning-powered system for automated fraud detection by analyzing historical claim data to identify fraudulent patterns. A real-time alert system will promptly notify stakeholders, streamlining claim processing, reducing financial losses, and enhancing customer satisfaction.

2. Objectives

1. ***Automated Fraud Detection****:* Develop a machine learning-based system to automatically identify fraudulent automobile insurance claims with high accuracy and minimal false positives/negatives.
2. ***Real-Time Fraud Identification****:* Enable the system to process claims data in real-time and promptly flag suspicious activities for further investigation.
3. ***Cost Reduction****:* Minimize financial losses associated with fraudulent claims by proactively detecting and preventing payouts for dubious cases.
4. ***Operational Efficiency****:* Streamline the claims processing workflow by reducing the reliance on manual reviews and static rule-based systems.
5. ***Customer Satisfaction****:* Expedite the resolution of genuine claims by focusing resources on legitimate policyholders, thereby fostering trust and loyalty among customers.
6. ***Adaptability to Evolving Fraud Techniques****:* Leverage machine learning models capable of learning and adapting to new fraud patterns and techniques over time.
7. ***Real-Time Alert Notification System****:* Implement a robust notification system that informs stakeholders (e.g., claim reviewers, investigators) of potentially fraudulent claims via email, SMS, or messaging platforms.
8. ***Scalable and Reliable Solution****:* Ensure the system is scalable to handle large volumes of data and robust enough to operate efficiently across various claim types and scenarios.

3. Scope of the Project

The project focuses on creating an automated fraud detection system specifically for automobile insurance claims. It aims to provide a robust, scalable, and efficient solution for detecting fraudulent claims and assisting insurance companies in expediting their investigation processes.

1. ***Targeted Datasets:***
   - **Insurance Claims Dataset**: This dataset includes cyclical information such as past claims history, customer demographics, policy characteristics, and claim outcomes. It serves as the primary source for identifying patterns and trends associated with fraudulent claims.
   - **Automobile Accident Dataset**: This dataset contains accident-related statistics, including the type of vehicles involved, accident severity, and contextual details such as the circumstances surrounding the accident. These data points add context to claims, improving the detection of potential fraud.
2. ***Fraud Detection Goals:***
   - Develop a system to assess claims and generate a probability score indicating the likelihood of fraud.
   - Focus on claims related to staged accidents, exaggerated losses, or falsified documentation, which are common forms of automobile insurance fraud.
3. ***Real-Time Analysis and Alerts:***
   - Incorporate mechanisms to process claims in real-time, allowing the system to flag suspicious claims immediately.
   - Provide warning signals or alerts for high-probability fraudulent claims, helping insurance firms prioritize and expedite investigations.
4. ***Integration with Stakeholder Processes:***
   - Enable insurance firms to integrate the system into their workflows for efficient handling of flagged claims.
   - Support compatibility with existing communication systems (e.g., email, SMS, or APIs) for delivering fraud notifications to relevant stakeholders.

5. ***Scalability and Adaptability:***
    - Design the system to handle large volumes of claims data, ensuring scalability for expanding datasets and operational requirements.
    - Ensure adaptability to evolving fraud patterns and techniques through updates and refinements over time.
6. ***Exclusions:***
    - The project will focus exclusively on detecting fraudulent automobile insurance claims. It will not include the approval or rejection of claims beyond fraud detection.
    - Fraud types outside the automobile insurance domain or unrelated to the provided datasets will not be addressed.
7. ***Deliverables:***
    - A comprehensive fraud detection system capable of providing actionable insights and aiding in fraud prevention.
    - A real-time alert notification mechanism to inform stakeholders of suspicious claims promptly.

4. Literature Review

Adedotun et al. focused on identifying the most effective machine learning algorithms for detecting fraudulent automobile insurance claims. They ***analyzed Random Forest, KNN, and XGBoost***, with the primary objective of determining which algorithm exhibited the highest classification accuracy for identifying fraudulent claims. Their results demonstrated the superior performance of ensemble learning methods such as Random Forest and XGBoost, which achieved classification accuracies of up to 98.5%. However, the study revealed significant limitations, particularly with logistic regression, which struggled with imbalanced datasets and performed poorly on complex fraud scenarios. Additionally, the study's reliance on outdated datasets restricted the ability to generalize findings to modern-day claims data. The authors recommended adopting newer datasets and employing cross-validation to enhance model robustness and applicability across diverse datasets and environments. **[1]**

Dhieb et al. developed a fraud detection framework for automobile insurance companies using XGBoost, aiming to increase the speed and accuracy of fraud detection while reducing dependency on human review.

The framework demonstrated superior performance, achieving higher precision and accuracy rates compared to Decision Tree, Naive Bayes, and KNN classifiers. By leveraging XGBoost's ability to handle imbalanced datasets and prioritize important features, the framework successfully identified fraudulent claims more efficiently than traditional methods. However, the study had two notable limitations: the lack of validation across multiple datasets and the absence of comparisons with advanced deep learning methods. The authors suggested future work to explore diverse datasets and expand the scope to additional insurance prediction problems, emphasizing the need for real-world testing and model scalability. **[2]**

Thanuj Kumar S. et al. (2021) This study explored the efficiency of machine learning methods, including Support Vector Machine (SVM) and XGBoost, in detecting fraudulent insurance claims. Using metrics such as accuracy, precision, recall, and F1-score, the authors compared the models' performance. Their findings highlighted XGBoost as the most accurate model, capable of effectively detecting fraudulent claims with minimal false positives, while logistic regression was the least effective. Despite these results, the study had limited scope due to its reliance on a single dataset, raising concerns about the model's ability to generalize across diverse datasets. The authors recommended incorporating datasets from various domains and refining models to include high-level features such as interactions between claims and claimants for improved fraud detection. **[3]**

Aslam F. et al. and colleagues proposed a framework for fraud detection in automobile insurance using Logistic Regression, Support Vector Machine, and Naive Bayes. Their analysis focused on six metrics derived from the confusion matrix, such as precision, recall, and F1-score, to evaluate the models' real-world applicability. The results revealed the utility of these models in addressing fraud detection challenges, with SVM performing better than Logistic Regression and Naive Bayes. However, the study did not explore ensemble methods, which are known for their superior performance in imbalanced datasets, nor did it include the practical deployment of models in a real-time environment. The lack of scalability and adaptability to dynamic fraud patterns limited the study's overall impact. **[4]**

Schrijver et al. conducted a systematic literature review (SLR) on data mining techniques for automobile insurance fraud detection, examining 50 studies published between 2019 and 2023. Their review revealed that most research relied heavily on supervised machine learning methods and cost-sensitive classifiers, while less attention was given to advanced approaches such as graph-based methods or deep learning. The

review highlighted the scarcity of publicly available datasets, which limited the ability to test models across diverse scenarios. Additionally, many studies focused solely on structured data, neglecting unstructured data sources like textual claim descriptions or image data from accident reports. The authors recommended future work to incorporate deep learning algorithms and explore more comprehensive datasets to improve fraud detection effectiveness and scalability. **[5]**

4.1 Gap Analysis and Contribution:

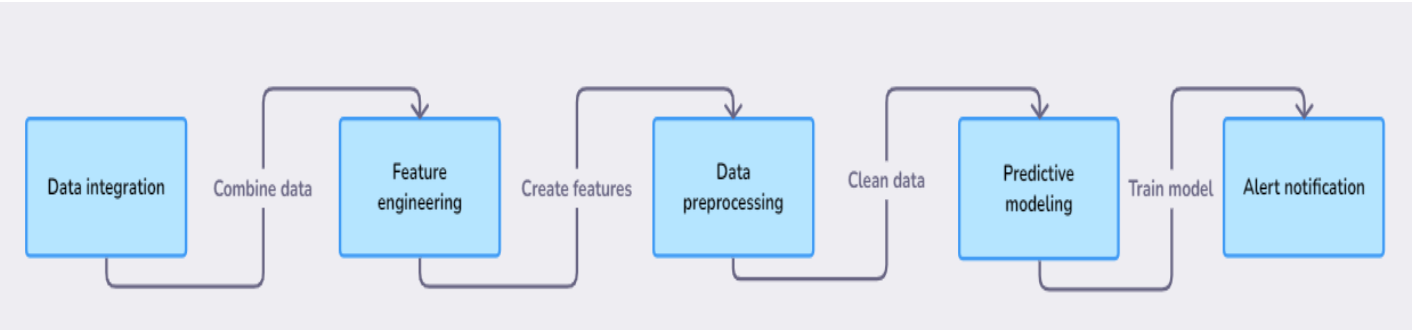
4.1.1 Identified Gaps

1. ***Limited Dataset Diversity****:*
    - Most studies relied on single or outdated datasets, which limited their generalizability to modern fraud detection challenges.
    - The scarcity of public datasets hindered the ability to test models across varied and realistic scenarios.
2. ***Insufficient Validation Techniques:***
    - Many studies lacked robust cross-validation methods or failed to validate their frameworks across multiple datasets, resulting in limited reliability for real-world applications.
3. ***Underutilization of Advanced Techniques****:*
    - While ensemble methods like XGBoost demonstrated strong performance, few studies explored integrating them with deep learning models for more robust fraud detection.
    - Graph-based methods and unstructured data analysis, such as using textual or image data, were largely overlooked.
4. ***Absence of Real-Time Systems:***
    - Most studies focused on offline analysis and did not address the need for real-time fraud detection or alert systems, which are critical for timely intervention.
5. ***Overemphasis on Accuracy****:*
    - Most studies focused on accuracy, overlooking key metrics like precision, recall, and F1-score, especially in imbalanced datasets where false negatives are crucial.

4.1.2 How This Project Addresses the Gaps

1. ***Diverse Dataset Usage:***
    - This project integrates a robust dataset encompassing a mix of historical claims data with varied attributes such as policy details, claim history, and incident specifics. Cross-validation ensures that the findings are generalizable across multiple scenarios.
2. ***Comprehensive Validation:***
    - The models will be rigorously validated using multiple data splits, simulating diverse real-world conditions. This ensures reliability and adaptability to changing fraud patterns.
3. ***Advanced Machine Learning and Deep Learning:***
    - The project leverages state-of-the-art machine learning algorithms, including XGBoost and Gradient Boosting, and explores deep learning approaches to enhance predictive accuracy and adapt to complex fraud behaviours.
4. ***Integration of Graph-Based and Unstructured Data:***
    - Beyond structured data, this project incorporates graph-based relationships (e.g., links between claims and claimants) and unstructured data (e.g., textual descriptions and images) to identify fraud more comprehensively.
5. ***Real-Time Fraud Detection:***
    - A real-time alert notification system will be developed to flag suspicious claims as they are processed, enabling stakeholders to act swiftly and prevent fraudulent payouts.
6. ***Focus on Relevant Metrics:***
    - The project emphasizes a balanced evaluation of models using precision, recall, and F1-score to address imbalanced datasets effectively, reducing false negatives and ensuring robust fraud detection.
7. ***Practical Deployment:***
    - The system enables scalable API integration with insurance workflows.

Workflow Of the Automated Detection of Automobile Insurance Fraud



The methodology for this project is a structured approach to identifying fraudulent automobile insurance claims using advanced machine learning techniques. The process integrates data preparation, model development, hyperparameter tuning, evaluation, and deployment for a reliable and interpretable fraud detection system. Each step is meticulously designed to ensure efficiency, accuracy, and robustness in handling real-world insurance claim data.

5. Methodology

5.1 Data Collection and Understanding

Two primary datasets are utilized to construct the fraud detection system:

1. **Auto Insurance Dataset**:
    - ***Purpose****:* Provides detailed information on insurance policies, customer demographics, and vehicle specifications.
    - ***Key Features****:*
        - ***policy_id:*** Unique identifier for each insurance policy, used as the primary key for merging datasets.
        - ***policy_annual_premium:*** Annual premium amount paid by the policyholder.
        - ***vehicle_age, auto_make:*** Describes the insured vehicle, essential for detecting patterns related to fraud.
        - ***months_as_customer:*** Duration of customer association with the insurer, which may correlate with fraudulent behavior.

2. **Insurance Claims Dataset**:
    - ***Purpose****:* Offers data on claims, accident details, and fraud indicators.
    - ***Key Features:***
        - ***incident_type, collision_type:*** Descriptions of reported incidents, useful for distinguishing genuine from fabricated claims**.**
        - ***fraud_reported:*** Binary label (Yes/No) indicating whether the claim is fraudulent**.**
        - ***vehicle_claim, total_claim_amount:*** Financial data that helps assess anomalies in claim amounts.

5.2 Merge Strategy

- The datasets are merged using an **inner join** on the ***policy_id*** column.
- **Reason for Inner Join**:
    - Ensures that only records with complete information from both datasets are retained, minimizing inconsistencies.
    - Provides a unified view of policyholder information and claim details, critical for model training.

5.3 Data Preprocessing

To prepare the merged dataset for machine learning, several preprocessing steps are applied:

5.3.1 Handling Missing Data

- **Numerical Variables**:
    - Missing values in features like ***total_claim_amount*** and ***vehicle_claim*** are imputed with their respective mean or median values.
- **Categorical Variables**:
    - Features like ***authorities_contacted*** and ***collision_type*** are imputed with the most frequent value or ***labeled*** as '*Unknown*'.

5.3.2 Feature Engineering

- **Outlier Detection and Handling**:
    - Statistical techniques *(e.g., IQR method)* are used to identify and cap extreme values in numerical features such as ***total_claim_amount.***

- **Combining Rare Categories**:
    - Low-frequency categories in columns like *auto_make* are grouped under a new category *"Other"* to reduce dimensionality and improve model interpretability.

5.3.3 Feature Encoding

- **Label Encoding**:
    - Binary variables (***fraud_reported, authorities_contacted***) are encoded as 0 and 1.
- **One-Hot Encoding**:
    - Multi-class categorical variables (***auto_make, collision_type, incident_type***) are converted into dummy variables to make them suitable for machine learning.

5.3.4 Feature Scaling

- **Why Scaling?**
    - Scaling ensures that numerical features like policy_annual_premium and vehicle_claim have consistent ranges, avoiding dominance by features with larger scales.
- **Technique Used**:
    - **StandardScaler** is applied to transform features to have a mean of 0 and a standard deviation of 1.

5.3.5 Handling Class Imbalance

- Fraud detection datasets are often imbalanced, with fewer fraudulent claims compared to legitimate ones.
- **Technique Used**:
    - **SMOTE (Synthetic Minority Oversampling Technique)** is employed to generate synthetic samples for the minority class (***fraud_reported=1***), balancing the dataset.

5.4 Model Development

To classify fraudulent claims effectively, multiple machine learning models are developed and evaluated:

5.4.1 Logistic Regression

1. **Why Used**:
    - Logistic Regression is chosen as a baseline model for binary classification problems like fraud detection.
    - It interprets linear relationships between predictors and fraud outcomes.

2. **Strengths**:
    - **Simplicity**: Easy to implement and computationally efficient.
    - **Interpretability**: Provides coefficients that indicate the strength and direction of relationships between features and the target variable.
    - **Baseline Performance**: Helps in setting a benchmark for other complex models.
3. **Limitations**:
    - Assumes a linear relationship between features and the log-odds of the target variable, which may not hold in real-world fraud detection scenarios.
    - Struggles with non-linear patterns and imbalanced datasets.
4. **Relevance to the Dataset**:
    - Logistic Regression can help identify whether features like ***policy_annual_premium*** or ***vehicle_age*** are directly associated with fraud likelihood.

5.4.2 Decision Tree Classifier

1. **Why Used**:
    - Decision Trees are ideal for discovering feature interactions and non-linear relationships in the data.
    - It can easily handle both categorical and numerical features without requiring much preprocessing.
2. **Strengths**:
    - **Interpretability**: The tree structure makes it easy to visualize decision paths and understand feature importance.
    - **Handles Feature Interactions**: Effective at capturing relationships between features such as ***incident_severity*** and ***policy_state***.
    - **Flexibility**: Works well with missing values and outliers.
3. **Limitations**:
    - Prone to overfitting, especially with high-dimensional data.
    - Less robust compared to ensemble methods like Random Forest.
4. **Relevance to the Dataset**:
    - Decision Trees help determine how combinations of variables like ***auto_make***, ***total_claim_amount***, and ***incident_severity*** contribute to predicting fraud.

5.4.3 Random Forest Classifier

1. **Why Used**:
    - Random Forest builds an ensemble of Decision Trees to improve predictive accuracy and control overfitting.
    - It is robust to noise and can handle high-dimensional data effectively.
2. **Strengths**:
    - **Reduces Overfitting**: Averages predictions from multiple trees, reducing variance.
    - **Feature Importance**: Provides a ranking of feature importance, which can be insightful for understanding fraud drivers.
    - **Handles Imbalanced Data**: With class weighting, it can focus on minority classes like fraudulent claims.
3. **Limitations**:
    - Computationally intensive for large datasets.
    - Less interpretable compared to single Decision Trees.
4. **Relevance to the Dataset**:
    - Random Forest is suited to this problem because it can handle the mix of categorical and numerical features and is robust to outliers in columns like ***total_claim_amount.***

5.4.4 Gradient Boosting Machines (GBM)

1. **Why Used**:
    - Gradient Boosting focuses on correcting the errors of previous models, making it powerful for capturing complex patterns in fraud detection.
    - It is particularly effective for datasets with imbalanced classes, as it places more weight on misclassified examples.
2. **Strengths**:
    - **Captures Complex Patterns**: Sequential learning allows GBM to model intricate relationships between features like ***incident_type***, witnesses, and ***fraud_reported.***
    - **Effective with Imbalanced Data**: Handles class imbalance by giving higher weight to minority class errors.
    - **Feature Importance**: Highlights key features contributing to fraud detection.

3. **Limitations**:
    - Computationally expensive, especially with large datasets and extensive hyperparameter tuning.
    - Sensitive to overfitting if not tuned carefully.
4. **Relevance to the Dataset**:
    - GBM is effective at identifying subtle patterns in the data that separate fraudulent claims from legitimate ones.

5.4.5 XGBoost (Extreme Gradient Boosting)

1. **Why Used**:
    - XGBoost is an optimized version of Gradient Boosting that improves training speed and handles large datasets efficiently.
    - It includes built-in regularization techniques ***(L1 and L2)*** to reduce overfitting.
2. **Strengths**:
    - **Efficiency**: Faster training and prediction compared to standard Gradient Boosting.
    - **Regularization**: Prevents overfitting by penalizing complex models.
    - **Handles Missing Data**: Automatically learns which features are most predictive, even with missing values.
    - **Scalability**: Handles large datasets with high cardinality features like ***auto_model*** and ***policy_state.***
3. **Limitations**:
    - Computationally demanding for hyperparameter tuning.
    - Requires careful tuning to balance bias and variance.
4. **Relevance to the Dataset**:
    - XGBoost is well-suited for the problem because it can effectively model interactions between features like ***claim_status, incident_location***, and ***bodily_injuries***, even in the presence of noisy or imbalanced data.

5.4.6. Voting Classifier

1. **Why Used**:
    - Combines the strengths of multiple models (e.g., *Gradient Boosting and Decision Tree*) to improve overall predictive performance.
    - Employs soft voting, which uses the predicted probabilities from each model to make a final decision.

2. **Strengths**:
    - **Improved Accuracy**: Leverages the strengths of individual models to achieve better performance.
    - **Robustness**: Reduces the risk of poor predictions from any single model.
3. **Limitations**:
    - Requires careful selection and tuning of base models.
    - Computationally intensive due to the use of multiple models.
4. **Relevance to the Dataset**:
    - The Voting Classifier is effective for this task as it balances the high precision of Gradient Boosting with the interpretability of Decision Trees.

6. Evaluation Metrics

Evaluation metrics are critical for assessing the performance of machine learning models, particularly in fraud detection, where the dataset is often imbalanced. Fraudulent cases typically form a minority of the data, making it essential to use metrics that emphasize both precision and recall. Below is a detailed explanation of the metrics used, including their formulae, purposes, and how they help in evaluating the performance of fraud detection models

6.1 Accuracy

$$\text{Accuracy} = \frac{\text{True Positives (TP)} + \text{True Negatives (TN)}}{\text{Total Number of Samples}}$$

1. **Purpose**:
    - Measures the overall correctness of the model's predictions.
    - Provides a simple evaluation by calculating the ratio of correctly classified instances to the total number of instances.
2. **Strengths**:
    - Easy to understand and compute.
    - Suitable for balanced datasets.
3. **Limitations**:
    - For imbalanced datasets, accuracy can be misleading because it may favor the majority class (e.g., legitimate claims).

4. **Example**:
    - If the model predicts all claims as legitimate in an imbalanced dataset, it could achieve high accuracy but fail to detect fraudulent claims.

6.2 Precision

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}}$$

1. **Purpose**:
    - Indicates the proportion of correctly identified fraudulent claims out of all claims predicted as fraudulent.
    - High precision means fewer false positives.
2. **Strengths**:
    - Useful when false positives are costly (e.g., wrongly flagging legitimate claims as fraudulent).
3. **Limitations**:
    - Does not account for false negatives, which may result in undetected fraud.
4. **Example**:
    - In fraud detection, high precision ensures that flagged claims are likely to be genuinely fraudulent, reducing the burden of manual review.

6.3 Recall (Sensitivity or True Positive Rate)

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}}$$

1. **Purpose**:
    - Measures the ability of the model to identify all actual fraudulent claims.
    - High recall ensures fewer missed fraud cases.
2. **Strengths**:
    - Essential in fraud detection, where missing fraudulent claims (false negatives) can result in significant financial losses.

3. **Limitations**:
    - May lead to more false positives if optimized alone, reducing precision.
4. **Example**:
    - A model with high recall captures most fraudulent claims, ensuring minimal undetected fraud cases.

6.4 F1 Score

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

1. **Purpose**:
    - Provides a harmonic mean of precision and recall, balancing both metrics.
    - Useful when the dataset is imbalanced, as it accounts for both false positives and false negatives.
2. **Strengths**:
    - Combines the strengths of precision and recall into a single metric.
    - Suitable for scenarios where both false positives and false negatives have significant consequences.
3. **Limitations**:
    - Does not differentiate between high recall and high precision.
4. **Example**:
    - A model with an F1 score of 0.85 balances the detection of fraudulent claims while minimizing false alarms.

6.5 Area Under the Receiver Operating Characteristic Curve (ROC-AUC)

1. **Purpose**:
    - Evaluates the model's ability to distinguish between fraudulent and legitimate claims across various thresholds.
    - The ROC curve plots the true positive rate (recall) against the false positive rate (1-specificity).
2. **Strengths**:
    - Summarizes the model's performance across all classification thresholds.
    - Robust to imbalanced datasets.
3. **Limitations**:
    - Does not provide specific information on a single threshold.

4. **Interpretation**:
    - A ROC-AUC value closer to 1 indicates excellent discrimination between classes, while a value of 0.5 suggests random guessing.

6.6 Confusion Matrix

1. **Purpose**:
    - True Positives (TP): Fraudulent claims correctly identified.
    - True Negatives (TN): Legitimate claims correctly identified.
    - False Positives (FP): Legitimate claims wrongly flagged as fraudulent.
    - False Negatives (FN): Fraudulent claims wrongly classified as legitimate.
2. **Strengths**:
    - Offers a granular view of prediction performance.
    - Helps identify specific types of errors, such as false positives or false negatives.
3. **Example**:
    - A confusion matrix with high TP and TN values but low FP and FN values indicates a well-performing model.

6.7 Specificity (True Negative Rate)

$$\text{Specificity} = \frac{\text{True Negatives (TN)}}{\text{True Negatives (TN)} + \text{False Positives (FP)}}$$

1. **Purpose**:
    - Measures the ability of the model to correctly identify legitimate claims.
    - Complements recall to assess the model's overall performance.
2. **Strengths**:
    - Useful when false positives have significant consequences, such as unnecessary investigations.
3. **Limitations**:
    - Does not address the detection of fraudulent claims (true positives).
4. **Example**:
    - A model with high specificity minimizes false alarms, ensuring legitimate claims are processed smoothly.

6.8 Why Use Multiple Models for Fraud Detection in This Dataset?

1. **Model Comparison:** Describing different models let us determine the optimum one for this particular insurance claims dataset, considering accuracy, readability, and time demanding aspects.
2. **Capturing Complex Patterns:** There are curvilinear relationships between customers' characteristics, policies, and claims in the features dataset. Random Forest, GBM, and XGBoost are particularly good for finding such patterns, whereas Logistic Regression is used as a baseline.
3. **Handling Class Imbalance:** Since there are way less fraudulent claims in this dataset compared to legitimate ones, different algorithms, such as Gradient Boosting or XGBoost, with class imbalance in mind focus on the misclassified instances.
4. **Overfitting vs. Generalization:** For instance, use of features like Random Forest and XG Boost minimizes overfitting and makes it possible to test for the result on other unseen claims most important in the detection of fraud.
5. **Efficiency and Performance:** XGBoost has been developed to handle big data and it poses high performance. While models such as Logistic Regression offer a more easily interpreted view of the information and can be used to set a benchmark more quickly in this relation to insurance claims.

6.9 How These Metrics Address Fraud Detection Challenges

- **Imbalanced Dataset**: Metrics like F1 Score, Precision, Recall, and ROC-AUC focus on detecting rare fraudulent cases while minimizing false positives and negatives.
- **Real-World Applicability**: Metrics like the confusion matrix and specificity help ensure the model's practicality in reducing unnecessary investigations while maintaining fraud detection accuracy.
- **Balanced Evaluation**: Combining multiple metrics ensures a comprehensive understanding of the model's strengths and weaknesses, leading to better decision-making during deployment.

7. Results and Discussion

7.1 Exploratory Data Analysis

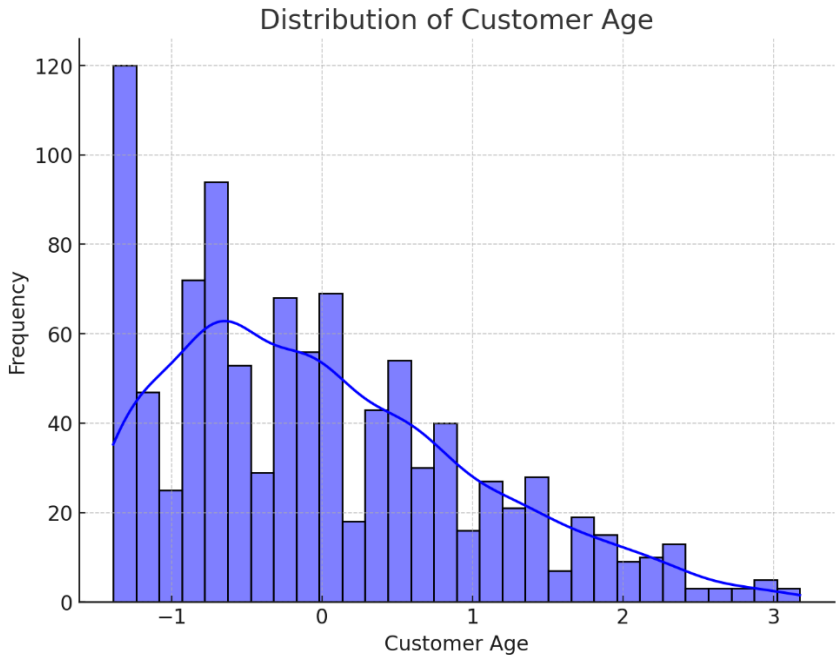


Fig.1 Distribution of Customer Age

***Observation**:*

- The age distribution shows a concentration in the 30-40 age range, with fewer younger (under 25) and older (above 60) customers.
- There is a slight tail extending toward older ages, indicating a smaller proportion of senior customers.

***Inference**:*

- Fraudulent activity may not be evenly distributed across all age groups. If younger or older age groups deviate from the central peak, they could represent higher-risk segments.
- Insurers may need to focus on targeted risk management strategies for customers in these outlier age brackets. For instance, younger customers may commit fraud due to financial instability, whereas older customers may face financial pressures or misinformation.

***Actionable Insight**:*

- Segmentation by age groups could be combined with other variables (e.g., claim type or premium size) for tailored fraud detection policies.

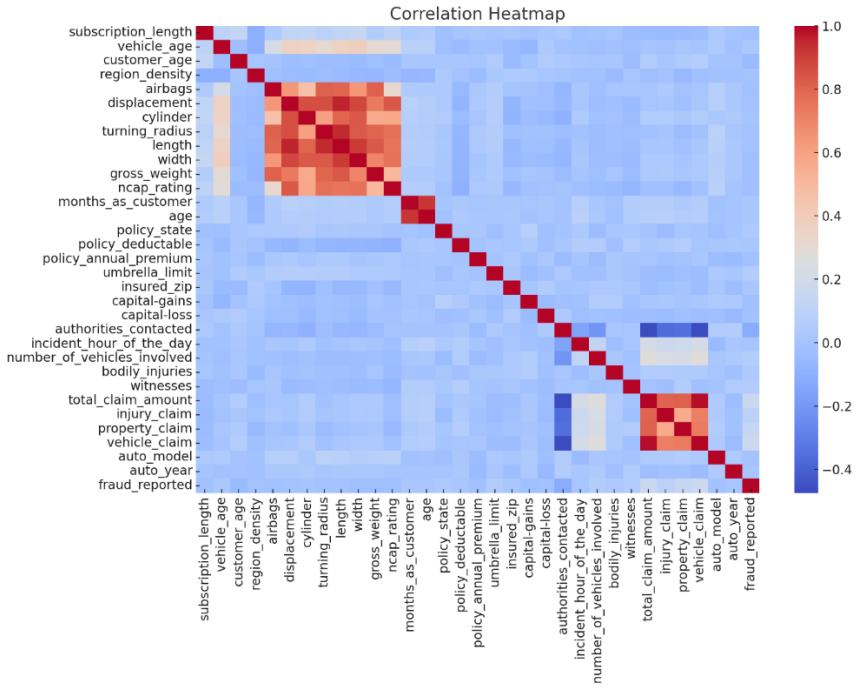


Fig.2 Correlation heatmap between the features

Observation:

- Strong correlations are observed among vehicle attributes such as displacement, cylinder, and gross_weight.
- Premium-related variables like policy_annual_premium also correlate with customer or vehicle features.

Inference:

- Fraudulent claims could involve combinations of correlated features (e.g., specific vehicle configurations and higher premiums).
- Unusual correlations between unrelated features (e.g., high displacement vehicles with very low premiums) might serve as red flags.

Actionable Insight:

- Leverage these correlations for feature engineering in predictive models. For example, derive risk scores based on unexpected combinations of variables.

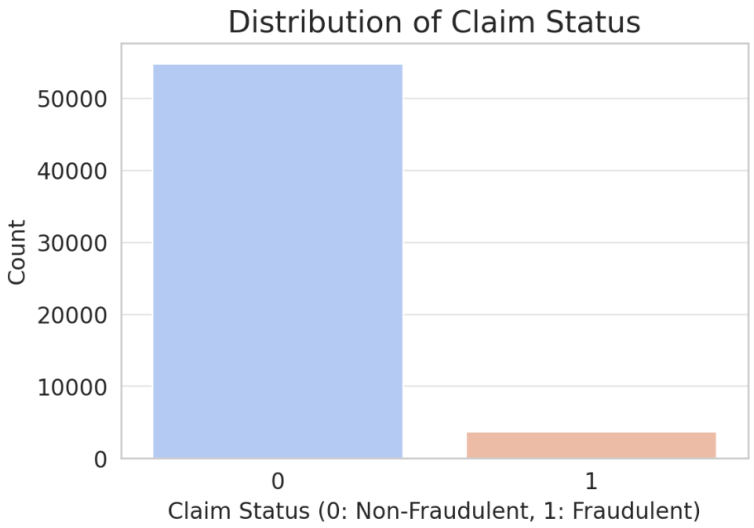


Fig.3 Distribution of Claim Status

Observation:

- Non-fraudulent claims dominate the dataset, with fraudulent claims making up a minority.

Inference:

- Fraud detection needs high precision. Even though fraudulent claims are fewer, they can result in significant financial losses due to high payouts.
- False positives in identifying fraud could alienate genuine customers, emphasizing the need for robust detection models.

Actionable Insight:

- Focus on improving precision (to minimize false positives) and recall (to detect as many fraudulent claims as possible).

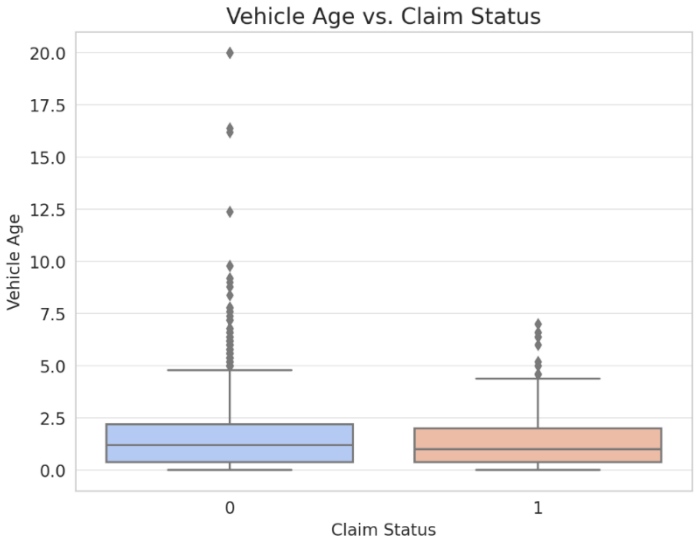


Fig.4 Vehicle Age vs. Claim Status

Observation:

- Fraudulent claims occur across a broader range of vehicle ages, including very old vehicles, while non-fraudulent claims cluster around newer vehicles.

Inference:

- Older vehicles may attract fraud due to higher repair costs, easier wear-and-tear justifications, or staged damage reports.
- Fraudsters may also avoid very new vehicles due to stricter manufacturer warranties or traceability.

Actionable Insight:

- Flag claims involving older vehicles for additional scrutiny, especially if repair costs appear inflated.

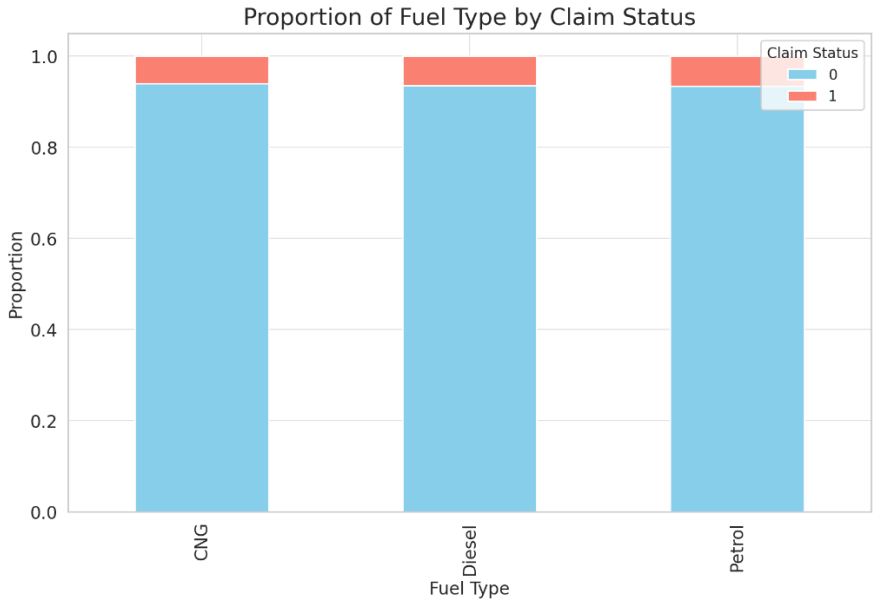


Fig.5 Proportion of Fuel Type by Claim Status

Observation*:*

- Fraudulent claims are disproportionately higher for diesel vehicles compared to petrol or CNG vehicles.

Inference*:*

- Diesel vehicles are often used for commercial purposes, leading to higher wear-and-tear and potentially exaggerated claims.
- Commercial vehicle fraud could involve intentional damage, staged accidents, or inflated repair costs.

***Actionable Insight**:*

- Pay closer attention to claims involving diesel vehicles, particularly for high-mileage or fleet vehicles.

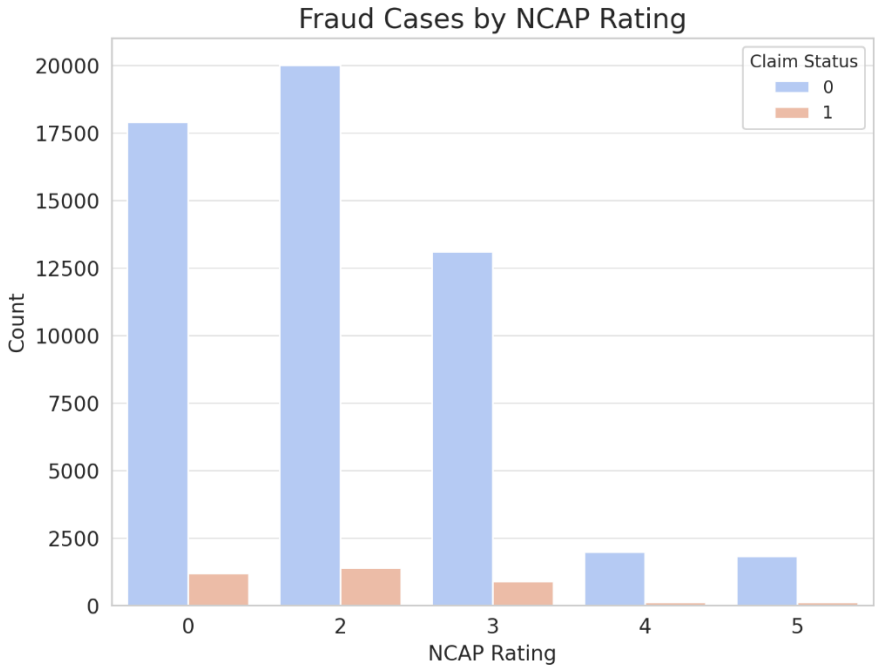


Fig.6 Fraud Cases by NCAP Rating

***Observation**:*

- Fraudulent claims are more common among vehicles with lower NCAP safety ratings.

***Inference**:*

- Vehicles with poor safety ratings are more likely to sustain genuine damage in accidents, which could be exploited by fraudsters to exaggerate claims.
- Fraudsters might prefer vehicles with fewer safety features due to lower traceability or tamper-resistant technologies.

***Actionable Insight**:*

- Introduce additional checks for claims involving low NCAP-rated vehicles.
- Cross-reference accident severity with safety ratings to identify anomalies.

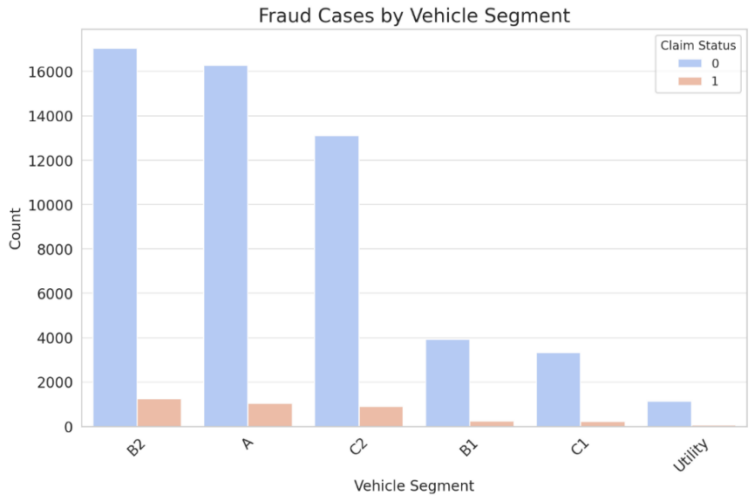


Fig.7 Fraud Cases by Vehicle Segment

Observation:

- Certain vehicle segments (e.g., C1, C2) have a higher number of fraudulent claims compared to premium or luxury segments.

Inference:

- Mid-range segments may represent a balance between plausible claim values and sufficient financial incentives for fraud.
- Luxury segments may have lower fraud rates due to stricter claim validation and fewer owners in the dataset.

Actionable Insight:

- Focus fraud detection efforts on mid-range segments.
- Validate high-value claims in premium segments with additional scrutiny.

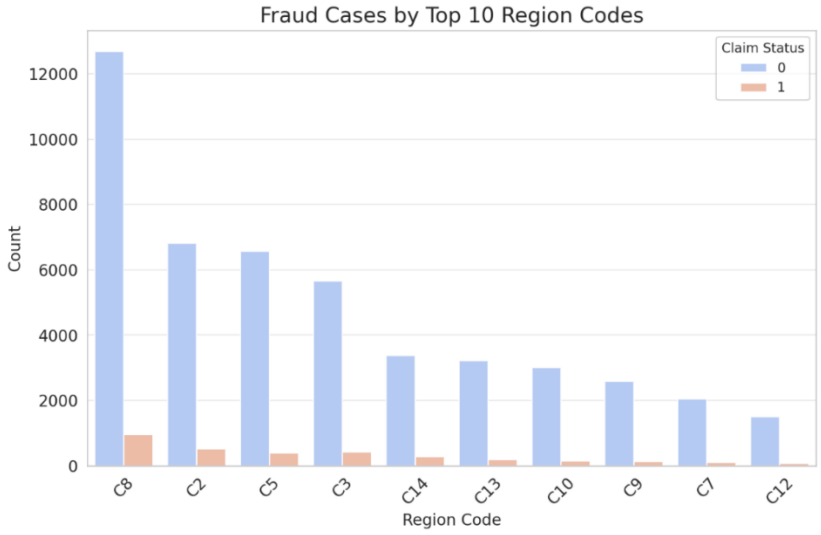


Fig.8 Fraud Cases by Region Code

Observation:

- Specific regions (e.g., C8, C2) exhibit significantly higher fraud cases.

Inference:

- Regional fraud trends could be influenced by local economic conditions or the presence of organized fraud networks.
- Certain regions might also reflect insurer policy gaps, such as lenient approval processes or lack of anti-fraud measures.

Actionable Insight:

- Conduct a regional analysis to identify root causes of fraud hotspots.
- Implement region-specific fraud detection policies and collaborate with local authorities to address organized fraud.

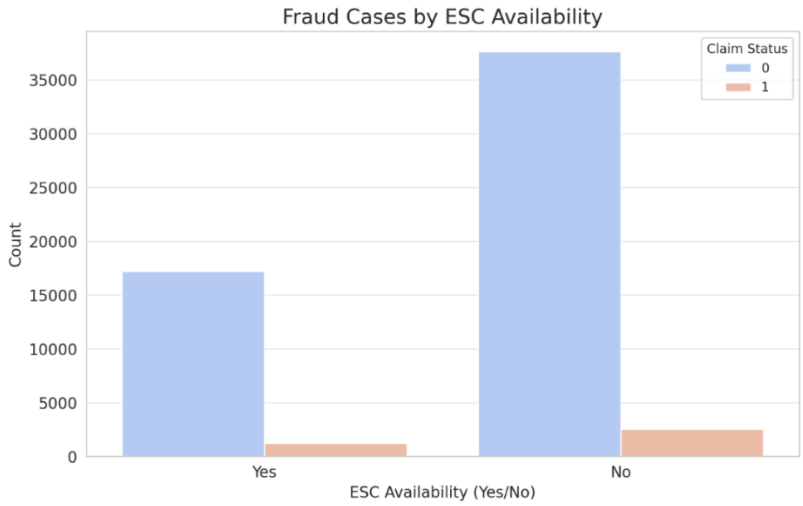


Fig.9 Fraud Cases by ESC Availability

Observation:

- Vehicles without ESC (Electronic Stability Control) show a higher number of fraudulent claims compared to vehicles with ESC, although the proportion of fraudulent claims remains small relative to non-fraudulent claims. Non-ESC vehicles also account for a significantly larger total number of claims.

Inference:

- Vehicles without ESC are likely more prone to accidents due to reduced stability control, increasing the frequency of both genuine and fraudulent claims.

- Fraudsters may target vehicles without ESC because the lack of safety technology makes it easier to justify accidents or exaggerated damages.
- ESC-equipped vehicles are inherently safer, leading to fewer accidents and thus fewer opportunities for fraud.

Actionable Insight:

- Enhanced Monitoring: Pay closer attention to claims involving vehicles without ESC, as these are both more accident-prone and fraud-prone.
- Policy Incentives: Encourage customers to adopt vehicles with ESC by offering reduced premiums for enhanced safety features.
- Fraud Patterns: Investigate recurring fraudulent patterns in claims for non-ESC vehicles, such as repair shop fraud or staged accidents, to improve fraud detection strategies.

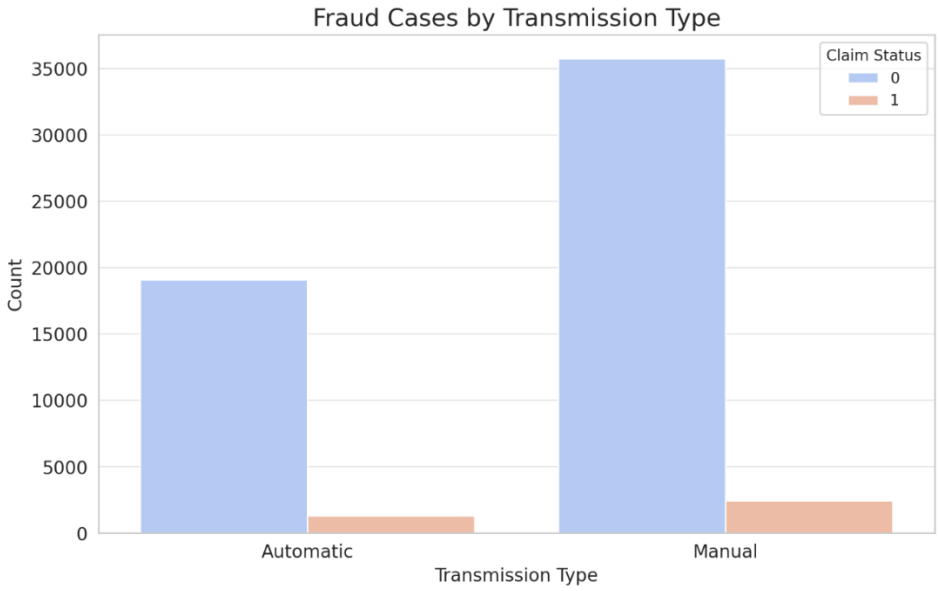


Fig.10 Fraud Cases by Transmission Type

Observation*:*

- Manual transmission vehicles show slightly higher fraudulent claims compared to automatic vehicles.

Inference*:*

- Manual transmission vehicles are often older, potentially used in commercial contexts, which might correlate with higher fraud exposure.
- Fraudsters may target older manual vehicles as their claims are harder to dispute due to lack of modern diagnostic technology.

***Actionable Insight**:*

- Pay closer attention to claims involving manual transmission vehicles, especially those used commercially.
- Investigate patterns of repair costs and accident reports for such vehicles.

7.2 Model Results

The results of the fraud detection models were evaluated using multiple machines learning algorithms, including Logistic Regression, Random Forest, Decision Tree, Gradient Boosting, XGBoost, and a Voting Classifier. The models were trained on a dataset comprising both legitimate and fraudulent insurance claims, and their performance was assessed using key evaluation metrics such as accuracy, precision, recall, F1 score, and ROC-AUC. Below is a detailed summary and discussion of the results:

1. Logistic Regression

- **Performance**:
  - *Accuracy: **73%***
  - *Precision (Fraudulent): **49%***
  - *Recall (Fraudulent): **33%***
  - *F1 Score (Fraudulent): **39%***
- **Discussion**: Logistic Regression serves as a baseline model due to its simplicity and interpretability. While the accuracy is reasonable, the model struggles to identify fraudulent claims effectively, as indicated by its low recall. This limitation highlights its inability to capture complex relationships in the dataset, especially in the presence of imbalanced data. Logistic Regression is better suited for initial exploratory analysis and testing.

2. Random Forest

- **Performance**:
  - *Accuracy: **72.67%***
  - *Precision (Fraudulent): **25%***
  - *Recall (Fraudulent): **1%***
  - *F1 Score (Fraudulent): **2%***
- **Discussion**: Random Forest demonstrated robustness with high-dimensional data but showed low recall for fraudulent claims, indicating difficulty in identifying fraud. This suggests the need for hyperparameter tuning or resampling techniques to address class imbalance and enhance fraud detection performance.

3. XGBoost

- **Performance**:
  - *Accuracy:* ***76.67%***
  - *Precision (Fraudulent):* ***59%***
  - *Recall (Fraudulent):* ***40%***
  - *F1 Score (Fraudulent):* ***48%***
- **Discussion**: XGBoost demonstrated better performance compared to Logistic Regression and Random Forest. Its high precision and moderate recall indicate that it is better at identifying fraudulent claims while minimizing false positives. This model effectively handles class imbalance and complex feature interactions, making it a strong contender for fraud detection tasks.

4. Decision Tree Classifier

- **Performance**:
  - *Accuracy:* ***82.33%***
  - *Precision (Fraudulent):* ***68%***
  - *Recall (Fraudulent):* ***62%***
  - *F1 Score (Fraudulent):* ***65%***
- **Discussion**: Decision Tree provided competitive results with a significant improvement in recall and F1 score for fraudulent claims. Its ability to handle feature interactions and imbalanced data contributed to its performance. However, Decision Trees are prone to overfitting, and their performance may vary significantly depending on the dataset.

5. Gradient Boosting

- **Performance**:
  - Best Parameters: ***{'learning_rate': 0.2, 'max_depth': 7, 'n_estimators': 200}***
  - Best F1 Score (Cross-Validation): ***0.89***
- **Discussion**: Gradient Boosting emerged as one of the top-performing models, achieving a high F1 score during cross-validation. Its iterative approach to reduce misclassification errors allowed it to detect fraudulent claims effectively. Gradient Boosting is well-suited for datasets with complex relationships and imbalanced classes.

6. Voting Classifier

- **Performance**:
  - *Accuracy:* ***82%***
  - *Precision (Fraudulent):* ***68%***
  - *Recall (Fraudulent):* ***61%***
  - *F1 Score (Fraudulent):* ***64%***
- **Discussion**: The Voting Classifier combined the strengths of Gradient Boosting and Decision Tree models. It achieved an overall accuracy of 82% and balanced performance across all metrics. This ensemble approach demonstrates the benefits of combining multiple models to leverage their individual strengths.

8. Key Observations

1. **Model Performance**:
   - Gradient Boosting and XGBoost outperformed other models in terms of F1 score and recall for fraudulent claims, making them ideal candidates for fraud detection tasks.
   - Logistic Regression and Random Forest were less effective in identifying fraudulent claims, likely due to their inability to handle class imbalance and complex feature interactions.
2. **Imbalanced Dataset Challenges**:
   - Class imbalance in the dataset posed a significant challenge, particularly for models like Logistic Regression and Random Forest.
   - Techniques such as SMOTE (Synthetic Minority Over-sampling Technique) were applied to address this issue, resulting in improved recall and F1 scores for fraudulent claims.
3. **Evaluation Metrics**:
   - Accuracy alone was insufficient to evaluate model performance due to the imbalanced nature of the dataset.
   - Metrics such as precision, recall, and F1 score provided a more comprehensive understanding of each model's ability to detect fraudulent claims.
4. **Ensemble Learning**:
   - The Voting Classifier demonstrated the effectiveness of ensemble learning by combining Gradient Boosting and Decision Tree models.

- It achieved balanced performance across metrics, making it a reliable choice for real-world deployment.

9. Code And Output Snippets

```
import pandas as pd

# Step 1.1: Load datasets
claims_data = pd.read_csv('/content/Insurance claims data.csv')
auto_data = pd.read_csv('/content/auto_insurance.csv', )

# Perform an inner join on 'policy_id' to retain only matching rows (1000 rows)
merged_data = pd.merge(claims_data, auto_data, on='policy_id', how='inner')

# Display the merged dataset structure to confirm 1000 rows are present
print("Merged Data (Inner Join) - First 5 Rows:")
print(merged_data.head())
print("\nMerged Data - Summary Info:")
print(merged_data.info())
```

```
filtered_data = merged_data.dropna(subset=['months_as_customer'])

# Display the first few rows and summary of the filtered merged data to confirm 1000 rows
print("Filtered Merged Data - First 5 Rows:")
print(filtered_data.head())
print("\nFiltered Merged Data - Summary Info:")
print(filtered_data.info())
```

```
# Fill missing values in `authorities_contacted` with 'Unknown'
filtered_data['authorities_contacted'] = filtered_data['authorities_contacted'].fillna('Unknown')

# Verify there are no more missing values
missing_data_summary = filtered_data.isnull().sum()
print("Remaining Missing Data After Imputation:")
print(missing_data_summary[missing_data_summary > 0])  # Should output nothing if all missing values are handled
```

```
from sklearn.preprocessing import LabelEncoder, StandardScaler
import pandas as pd

# Make a copy to avoid modifying the original dataset
data = filtered_data.copy()

# Step 1: Combine Rare Categories in `auto_model`
# Define a threshold to combine rare categories
threshold = 20  # Example threshold for combining categories
model_counts = data['auto_model'].value_counts()
rare_models = model_counts[model_counts < threshold].index
data['auto_model'] = data['auto_model'].apply(lambda x: 'Other' if x in rare_models else x)

# Step 2: Encode Categorical Variables
# Binary columns: Label encoding
binary_cols = ['fraud_reported', 'authorities_contacted']
label_encoders = {}
```

```
for col in binary_cols:
    le = LabelEncoder()
    data[col] = le.fit_transform(data[col])
    label_encoders[col] = le

# High cardinality columns: Label encoding instead of one-hot encoding
data['auto_model'] = LabelEncoder().fit_transform(data['auto_model'])
data['policy_state'] = LabelEncoder().fit_transform(data['policy_state'])  # example of another high-cardinality column

# One-hot encode other lower-cardinality categorical columns
data = pd.get_dummies(data, columns=['fuel_type', 'incident_type', 'collision_type', 'incident_severity'], drop_first=True)
```

```
# Step 3: Scale Numerical Features
numerical_cols = data.select_dtypes(include=['float64', 'int64']).columns
scaler = StandardScaler()
data[numerical_cols] = scaler.fit_transform(data[numerical_cols])

# Check the processed data
print("Processed Data - First 5 Rows:")
print(data.head())
print("Processed Data Shape:", data.shape)
```

```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, accuracy_score
from imblearn.over_sampling import SMOTE
import pandas as pd

# Assuming `data` is the preprocessed DataFrame with all categorical columns encoded and no missing values

# Step 1: Ensure `fraud_reported` is encoded as integers (0 and 1)
data['fraud_reported'] = data['fraud_reported'].astype(int)

# Drop `policy_id` if it's still present in the dataset
data = data.drop(columns=['policy_id'], errors='ignore')

# Dynamically identify and encode all categorical columns using one-hot encoding
categorical_cols = data.select_dtypes(include=['object']).columns
data = pd.get_dummies(data, columns=categorical_cols, drop_first=True)

# Step 2: Define features (X) and target (y)
X = data.drop(columns=['fraud_reported'])
y = data['fraud_reported']

# Step 3: Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Step 4: Apply SMOTE to the training data
smote = SMOTE(random_state=42)
X_train_res, y_train_res = smote.fit_resample(X_train, y_train)
```

```python
# Step 5: Train a Random Forest model with class weights
model = RandomForestClassifier(random_state=42, class_weight='balanced')
model.fit(X_train_res, y_train_res)

# Step 6: Make predictions
y_pred = model.predict(X_test)

# Step 7: Evaluate the model
print("Random forest Accuracy:", accuracy_score(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))
```

```
Random forest Accuracy: 0.7266666666666667
Classification Report:
              precision    recall  f1-score   support

           0       0.73      0.99      0.84       220
           1       0.25      0.01      0.02        80

    accuracy                           0.73       300
   macro avg       0.49      0.50      0.43       300
weighted avg       0.60      0.73      0.62       300
```

OUTPUT OF RANDOM FOREST

```python
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, accuracy_score

# Train and evaluate Logistic Regression with class weighting
logistic_model = LogisticRegression(class_weight='balanced', random_state=42)
logistic_model.fit(X_train_res, y_train_res)
y_pred_logistic = logistic_model.predict(X_test)

print("Logistic Regression Model Accuracy:", accuracy_score(y_test, y_pred_logistic))
print("Logistic Regression Classification Report:\n", classification_report(y_test, y_pred_logistic))
```

```
Logistic Regression Model Accuracy: 0.73
Logistic Regression Classification Report:
              precision    recall  f1-score   support

           0       0.78      0.88      0.83       220
           1       0.49      0.33      0.39        80

    accuracy                           0.73       300
   macro avg       0.64      0.60      0.61       300
weighted avg       0.70      0.73      0.71       300
```

OUTPUT OF LOGISTIC REGRESSION

```python
from xgboost import XGBClassifier

# Calculate class weight for XGBoost
scale_pos_weight = y_train_res.value_counts()[0] / y_train_res.value_counts()[1]

# Train and evaluate XGBoost with class weighting
xgb_model = XGBClassifier(random_state=42, scale_pos_weight=scale_pos_weight, use_label_encoder=False, eval_metric='logloss')
xgb_model.fit(X_train_res, y_train_res)
y_pred_xgb = xgb_model.predict(X_test)

print("XGBoost Model Accuracy:", accuracy_score(y_test, y_pred_xgb))
print("XGBoost Classification Report:\n", classification_report(y_test, y_pred_xgb))
```

```
XGBoost Model Accuracy: 0.7666666666666667
XGBoost Classification Report:
              precision    recall  f1-score   support

           0       0.80      0.90      0.85       220
           1       0.59      0.40      0.48        80

    accuracy                           0.77       300
   macro avg       0.70      0.65      0.66       300
weighted avg       0.75      0.77      0.75       300
```

OUTPUT OF XGBOOST MODEL

```python
from sklearn.ensemble import GradientBoostingClassifier

# Train and evaluate Gradient Boosting Classifier
gb_model = GradientBoostingClassifier(random_state=42)
gb_model.fit(X_train_res, y_train_res)
y_pred_gb = gb_model.predict(X_test)

print("Gradient Boosting Model Accuracy:", accuracy_score(y_test, y_pred_gb))
print("Gradient Boosting Classification Report:\n", classification_report(y_test, y_pred_gb))
```

```
Gradient Boosting Model Accuracy: 0.7966666666666666
Gradient Boosting Classification Report:
              precision    recall  f1-score   support

           0       0.84      0.89      0.86       220
           1       0.64      0.55      0.59        80

    accuracy                           0.80       300
   macro avg       0.74      0.72      0.73       300
weighted avg       0.79      0.80      0.79       300
```

OUTPUT OF GRANDIENT BOOSTING MODEL

```
from sklearn.tree import DecisionTreeClassifier

# Train and evaluate Decision Tree Classifier with class weighting
dt_model = DecisionTreeClassifier(class_weight='balanced', random_state=42)
dt_model.fit(X_train_res, y_train_res)
y_pred_dt = dt_model.predict(X_test)

print("Decision Tree Model Accuracy:", accuracy_score(y_test, y_pred_dt))
print("Decision Tree Classification Report:\n", classification_report(y_test, y_pred_dt))
```

```
Decision Tree Model Accuracy: 0.8233333333333334
Decision Tree Classification Report:
              precision    recall  f1-score   support

           0       0.87      0.90      0.88       220
           1       0.68      0.62      0.65        80

    accuracy                           0.82       300
   macro avg       0.78      0.76      0.77       300
weighted avg       0.82      0.82      0.82       300
```

OUTPUT OF DECISION TREE MODEL

```
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import GridSearchCV

# Define parameter grids
gb_param_grid = {
    'n_estimators': [50, 100, 200],
    'learning_rate': [0.01, 0.1, 0.2],
    'max_depth': [3, 5, 7]
}

dt_param_grid = {
    'max_depth': [5, 10, 15, None],
    'min_samples_split': [2, 10, 20],
    'class_weight': ['balanced']
}

# Set up GridSearchCV for Gradient Boosting
gb_grid_search = GridSearchCV(GradientBoostingClassifier(random_state=42), gb_param_grid, scoring='f1', cv=5)
gb_grid_search.fit(X_train_res, y_train_res)

# Set up GridSearchCV for Decision Tree
dt_grid_search = GridSearchCV(DecisionTreeClassifier(random_state=42), dt_param_grid, scoring='f1', cv=5)
dt_grid_search.fit(X_train_res, y_train_res)

# Display best parameters and scores
print("Best Gradient Boosting Parameters:", gb_grid_search.best_params_)
print("Best Gradient Boosting F1 Score:", gb_grid_search.best_score_)

print("Best Decision Tree Parameters:", dt_grid_search.best_params_)
print("Best Decision Tree F1 Score:", dt_grid_search.best_score_)
```

```
Best Gradient Boosting Parameters: {'learning_rate': 0.2, 'max_depth': 7, 'n_estimators': 200}
Best Gradient Boosting F1 Score: 0.8956403852997337
Best Decision Tree Parameters: {'class_weight': 'balanced', 'max_depth': 15, 'min_samples_split': 2}
Best Decision Tree F1 Score: 0.8634956247456247
```

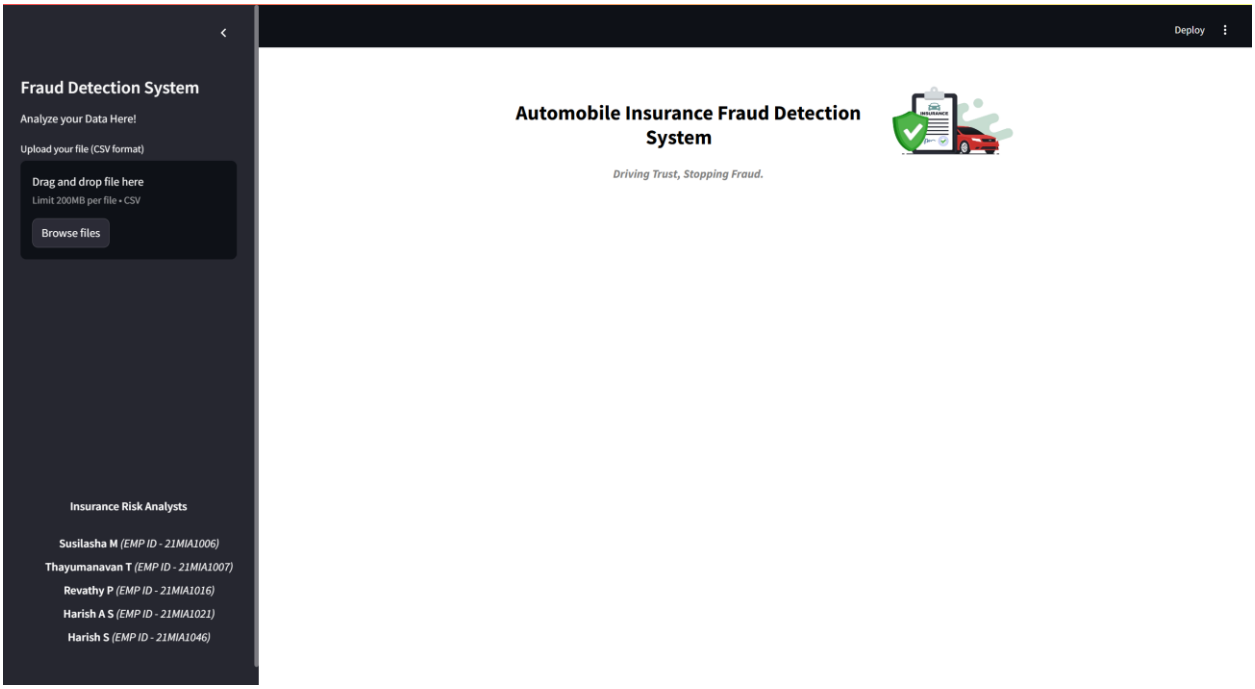
HYPER PARAMETER SELECTION OF GRANDIENT BOOSTING & DECISION TREE MODELS

```python
from sklearn.ensemble import VotingClassifier

# Initialize models with best parameters from Grid Search
best_gb = GradientBoostingClassifier(**gb_grid_search.best_params_, random_state=42)
best_dt = DecisionTreeClassifier(**dt_grid_search.best_params_, random_state=42)

# Create a Voting Classifier with the tuned models
voting_clf = VotingClassifier(estimators=[('gb', best_gb), ('dt', best_dt)], voting='soft')
voting_clf.fit(X_train_res, y_train_res)

# Make predictions with the Voting Classifier
y_pred_voting = voting_clf.predict(X_test)

# Evaluate the Voting Classifier
print("Voting Classifier Accuracy:", accuracy_score(y_test, y_pred_voting))
print("Voting Classifier Classification Report:\n", classification_report(y_test, y_pred_voting))
```

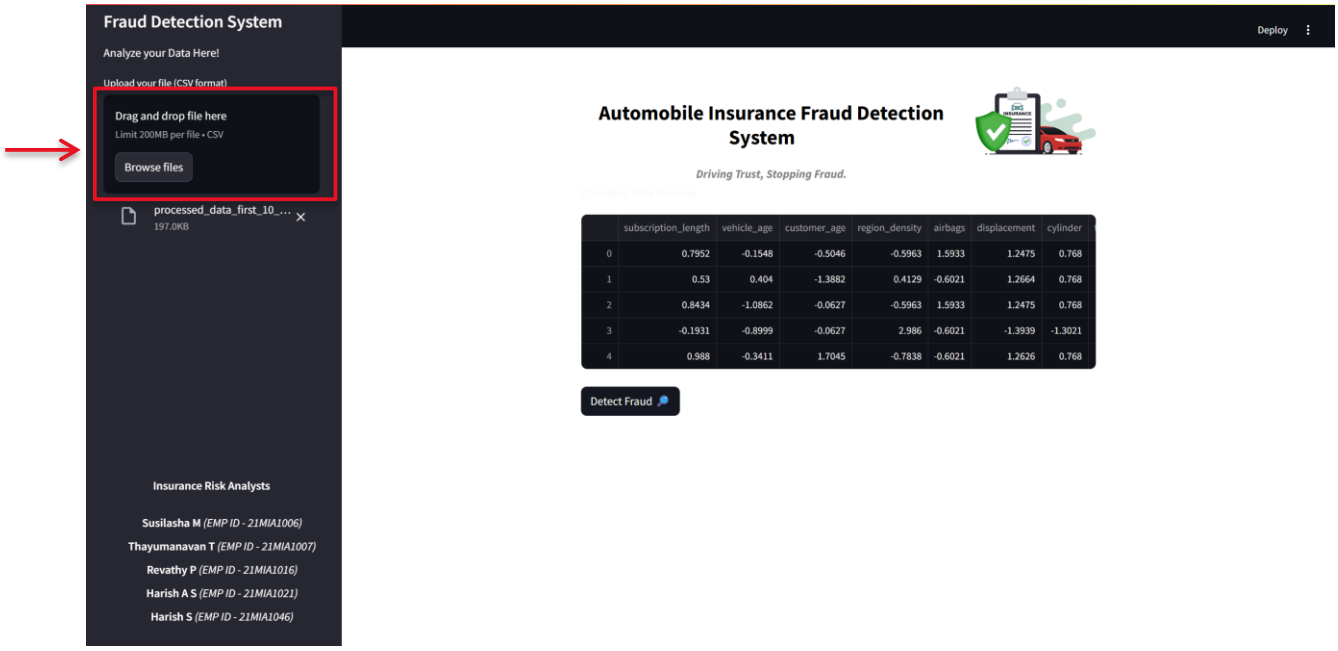
```
Voting Classifier Accuracy: 0.82
Voting Classifier Classification Report:
              precision    recall  f1-score   support

           0       0.86      0.90      0.88       220
           1       0.68      0.61      0.64        80

    accuracy                           0.82       300
   macro avg       0.77      0.75      0.76       300
weighted avg       0.82      0.82      0.82       300
```

OUTPUT OF VOTING CLASSIFIER MODEL

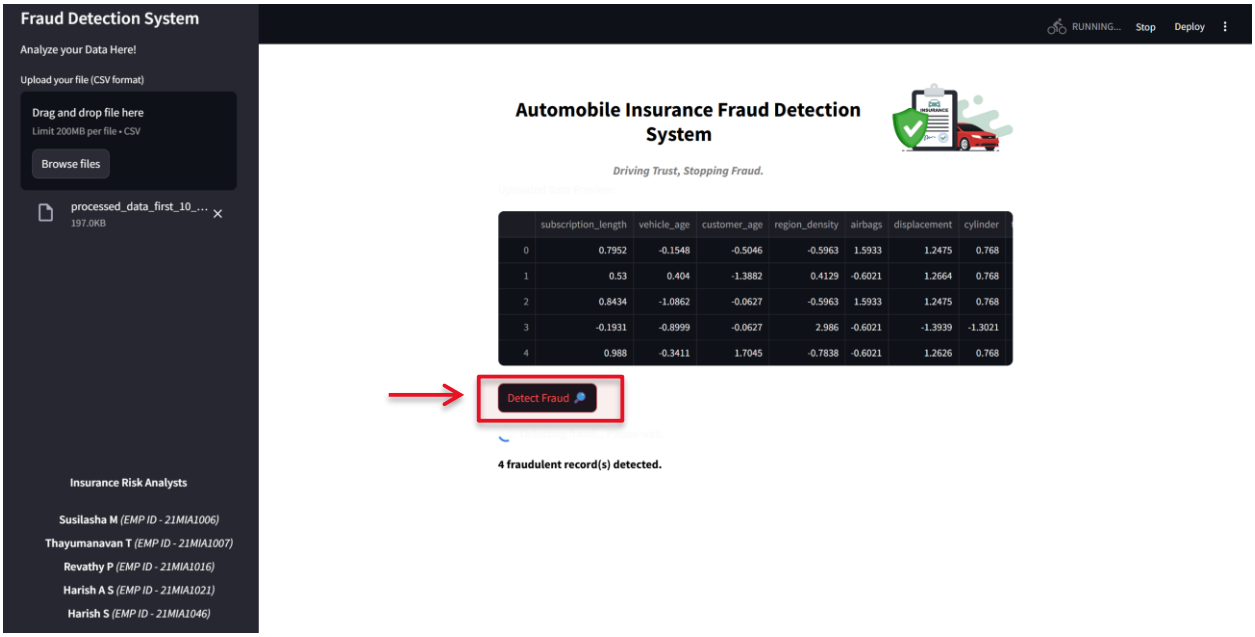
PROTOTYPE OF AUTOMATED FRAUD DETECTION AND ALERT NOTIFICATION SYSTEM



Home Page of the Interface

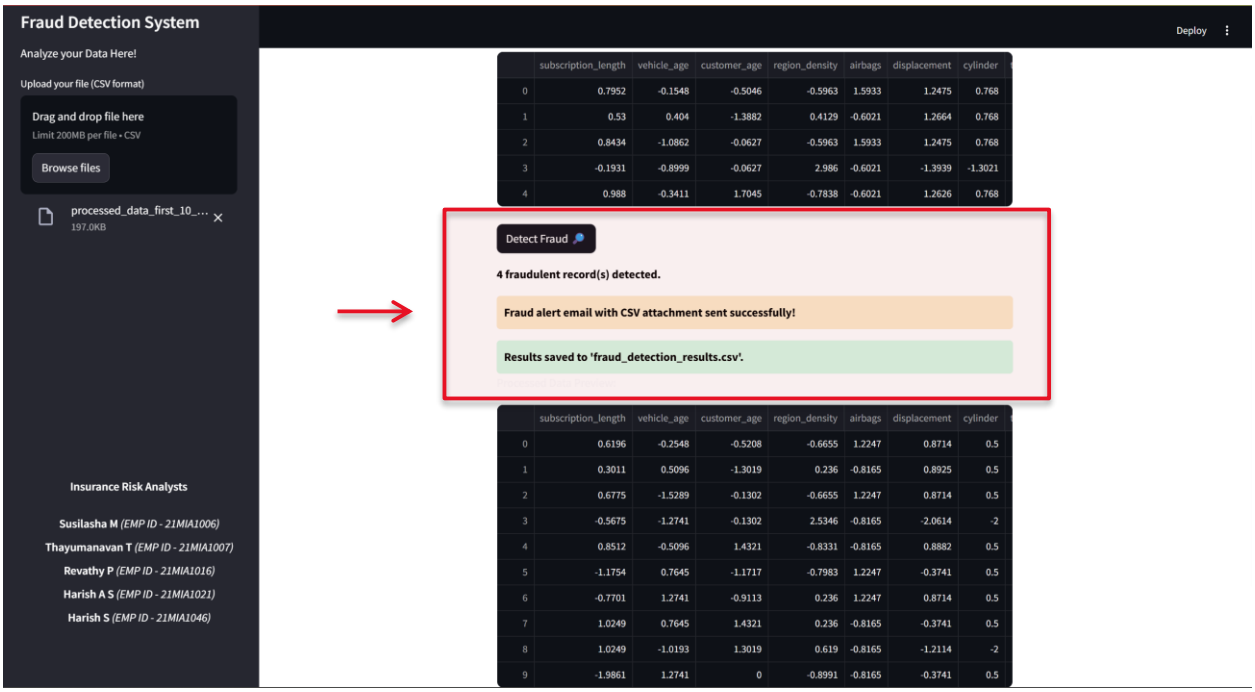


File is being Uploaded

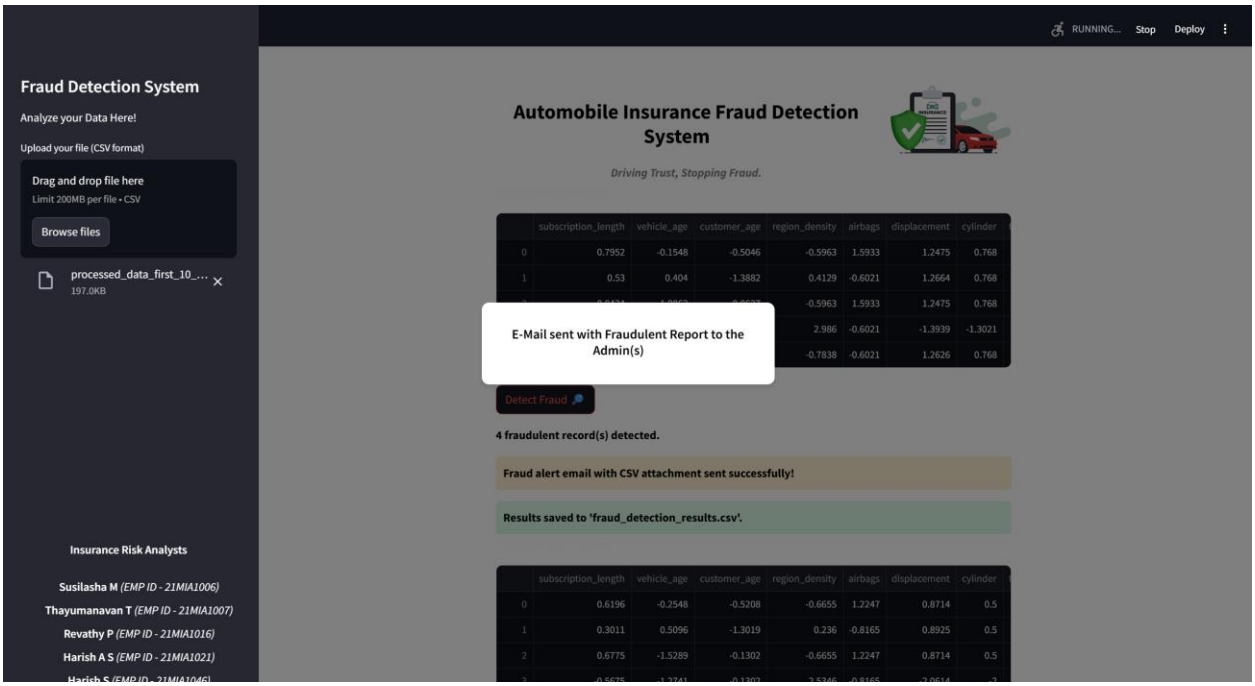


Loading Screen – While Analyzing the Data

<u>After loading the data, we have click detect fraud so that the pretrained model analyses the data and gives output.</u>

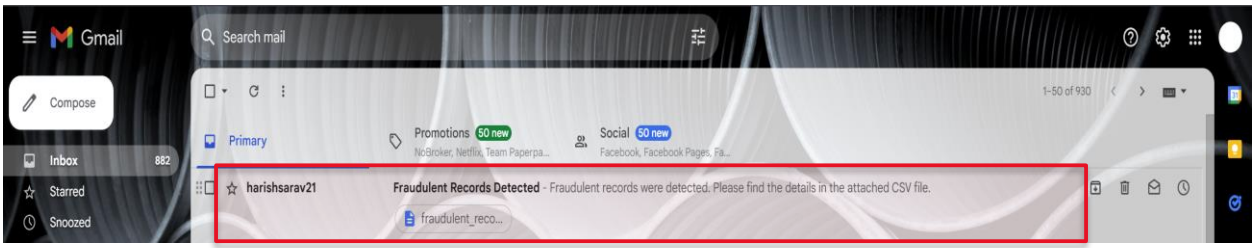


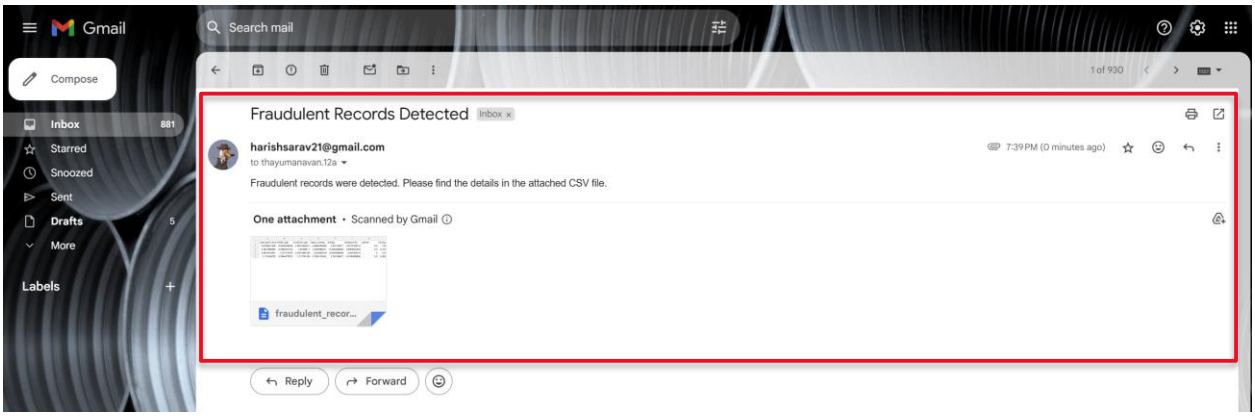
Once the Fraudulent Entries are found, they are displayed here.

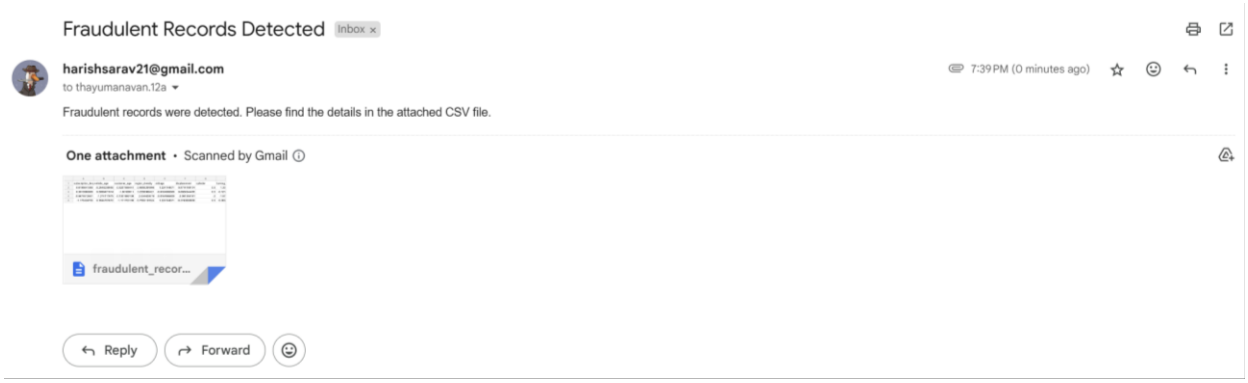


Once After the Fraudulent Entries are found an E – mail is sent to the Admin(s)

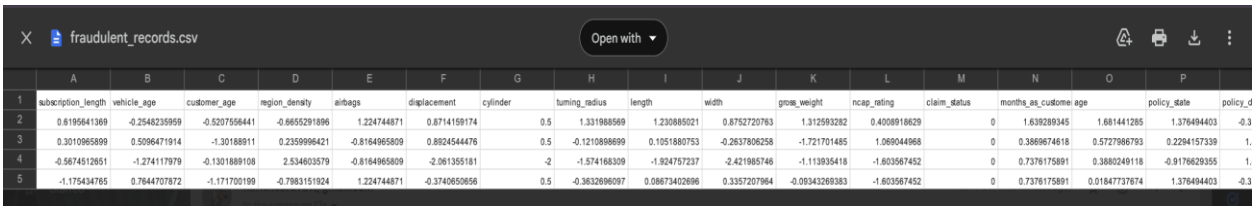
Email sent by our Systems







Fraudulent Report Received via E – Mail



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	
1	subscription_length	vehicle_age	customer_age	region_density	airbags	displacement	cylinder	turning_radius	length	width	gross_weight	ncap_rating	claim_status	months_as_custome	age	policy_state	policy_d
2	0.6195641369	-0.2548235959	-0.5207556441	-0.6655291896	1.224744871	0.8714159174	0.5	1.331988569	1.230885021	0.8752720763	1.312593282	0.4008918629	0	1.639289345	1.681441285	1.376494403	-0.3
3	0.3010965899	0.5096471914	-1.30188911	0.2359996421	-0.8164965809	0.8924544476	0.5	-0.1210898699	0.1051880753	-0.2637806258	-1.721701485	1.069044968	0	0.3869674618	0.5727986793	0.2294157339	1.
4	-0.5674512651	-1.274117979	-0.1301889108	2.534603579	-0.8164965809	-2.061355181	-2	-1.574168309	-1.924757237	-2.421985746	-1.113935418	-1.603567452	0	0.7376175891	0.3880249118	-0.9176629355	1.
5	-1.175434765	0.7644707872	-1.171700199	-0.7983151924	1.224744871	-0.3740650656	0.5	-0.3632696097	0.08673402696	0.3357207964	-0.09343269383	-1.603567452	0	0.7376175891	0.01847737674	1.376494403	-0.3

10. Conclusion

The study aimed to develop and evaluate machine learning models for detecting fraudulent automobile insurance claims, addressing the challenges posed by the inherent imbalance in the dataset. Various models, including Logistic Regression, Random Forest, XGBoost, Decision Tree, Gradient Boosting, and a Voting Classifier, were implemented to assess their suitability for fraud detection. Logistic Regression served as a baseline model, offering simplicity and interpretability but struggled with non-linear relationships and imbalanced data. Random Forest demonstrated moderate performance but required additional tuning to improve recall for fraudulent claims. XGBoost and Gradient Boosting emerged as the most effective models, with Gradient Boosting achieving the highest F1 score and XGBoost excelling in recall and precision, thanks to their ability to handle complex patterns and class imbalance. The ensemble Voting Classifier, combining Gradient Boosting and Decision Tree models, provided balanced and reliable performance, making it an excellent candidate for real-world applications. Preprocessing techniques such as SMOTE to address class imbalance, feature engineering, and encoding proved critical to enhancing model performance. Evaluation metrics like precision, recall, F1 score, and ROC-AUC were emphasized over accuracy to provide a more comprehensive performance assessment, particularly for imbalanced datasets. The study highlighted the practical implications of using machine learning for fraud detection, enabling insurance companies to streamline claim processing, reduce fraud-related financial losses, and improve customer trust. Future

directions include testing on more diverse datasets, incorporating real-time data for immediate fraud detection, exploring advanced techniques like neural networks, and integrating the models with real-world insurance processes through user-friendly tools. Ultimately, the study underscores the potential of advanced machine learning techniques to transform fraud detection in the insurance industry, paving the way for efficient and reliable fraud prevention systems.

11. Future work

Future work in the domain of automobile insurance fraud detection presents several promising avenues for exploration and improvement. One key area is the integration of real-time data processing capabilities to enable immediate detection of fraudulent claims, thereby reducing delays in claim verification and investigation. Advanced algorithms, including neural networks, deep learning, and graph-based models, could be explored to uncover intricate fraud patterns and enhance detection accuracy. Expanding model validation with diverse datasets from different geographic and industrial contexts would improve generalizability and robustness. Additionally, incorporating explainable AI (XAI) techniques can make the models more interpretable, fostering trust and transparency among stakeholders. Practical deployment could involve integrating these models into existing claim management systems, accompanied by interactive dashboards or tools for analysts. Addressing challenges such as class imbalance through advanced sampling techniques and including socioeconomic or behavioral features could refine fraud detection further. Dynamic thresholding mechanisms and assigning risk scores based on prediction probabilities could prioritize investigations and enhance precision. Furthermore, leveraging machine learning insights to develop fraud prevention strategies, such as identifying recurring patterns and implementing stricter policy measures, can contribute to mitigating fraudulent activities. Finally, deploying pilot programs within insurance companies would allow real-world testing and refinement of the models, ensuring their scalability and effectiveness in practical applications. These future directions can collectively strengthen the reliability and adaptability of fraud detection systems in the ever-evolving landscape of the insurance industry.

12. References

[1] *F, A. A., A, O. O., S, A. O., A, A. J., Okagbue, H., I., & O, O. (2023). Prediction of automobile insurance fraud claims using machine learning. THE SCIENTIFIC TEMPER, 14(03), 756–762. https://doi.org/10.58414/scientifictemper.2023.14.3.29*

[2] *Dhieb, N., Ghazzai, H., Besbes, H., & Massoud, Y. (2019). Extreme Gradient Boosting Machine Learning Algorithm For Safe Auto Insurance Operations. In 2019 IEEE International Conference on Vehicular Electronics and Safety (ICVES). https://doi.org/10.1109/icves.2019.8906396*

[3] *S, T. K., Deep, U., Shoiab, S., Atif, S., Bhatnagar, T., & T, R. (2021). Insurance Fraud Detection Using Machine Learning. INTERNATIONAL JOURNAL OF ADVANCED INFORMATION AND COMMUNICATION TECHNOLOGY, 1–4. https://doi.org/10.46532/ijaict-2020210101*

[4] *Aslam, F., Hunjra, A. I., Ftiti, Z., Louhichi, W., & Shams, T. (2022). Insurance fraud detection: Evidence from artificial intelligence and machine learning. Research in International Business and Finance, 62, 101744. https://doi.org/10.1016/j.ribaf.2022.101744*

[5] *Schrijver, G., Sarmah, D. K., & El-Hajj, M. (2024). Automobile Insurance Fraud Detection Using Data Mining: A Systematic Literature Review. Intelligent Systems With Applications, 200340. https://doi.org/10.1016/j.iswa.2024.200340*