Fraudulent Claim Detection

Group Member(s): Sarthaka Mahapatra

Date of Submission: 19th May 2025

Submitted for: Machine Learning Assignment - Insurance Domain

Contents

1. Problem Statement

Background:

Global Insure, a reputed insurance provider, handles thousands of insurances claims every year. Unfortunately, a significant fraction of these claims is fraudulent, causing substantial financial losses to the organization.

Challenge:

Currently, fraud detection relies on manual inspections, which are time-consuming and often delay the identification of fraudulent claims. As a result, fraud is frequently discovered **after pay-outs have already been made**, making the process inefficient and costly.

Objective:

The core objective of this project is to develop a **predictive machine learning model** that can identify **potentially fraudulent claims early in the claim approval process**. This solution aims to:

- **Minimize financial losses** by preventing fraud-related pay-outs.
- **Speed up claim processing** for legitimate customers.
- **Improve operational efficiency** through data-driven automation.
- **Identify patterns and insights** in customer and claim data that are indicative of fraud.

The model should learn from historical data and be able to classify a new claim as either **fraudulent** or **legitimate** with high accuracy and recall, especially ensuring that fraudulent cases are not missed.

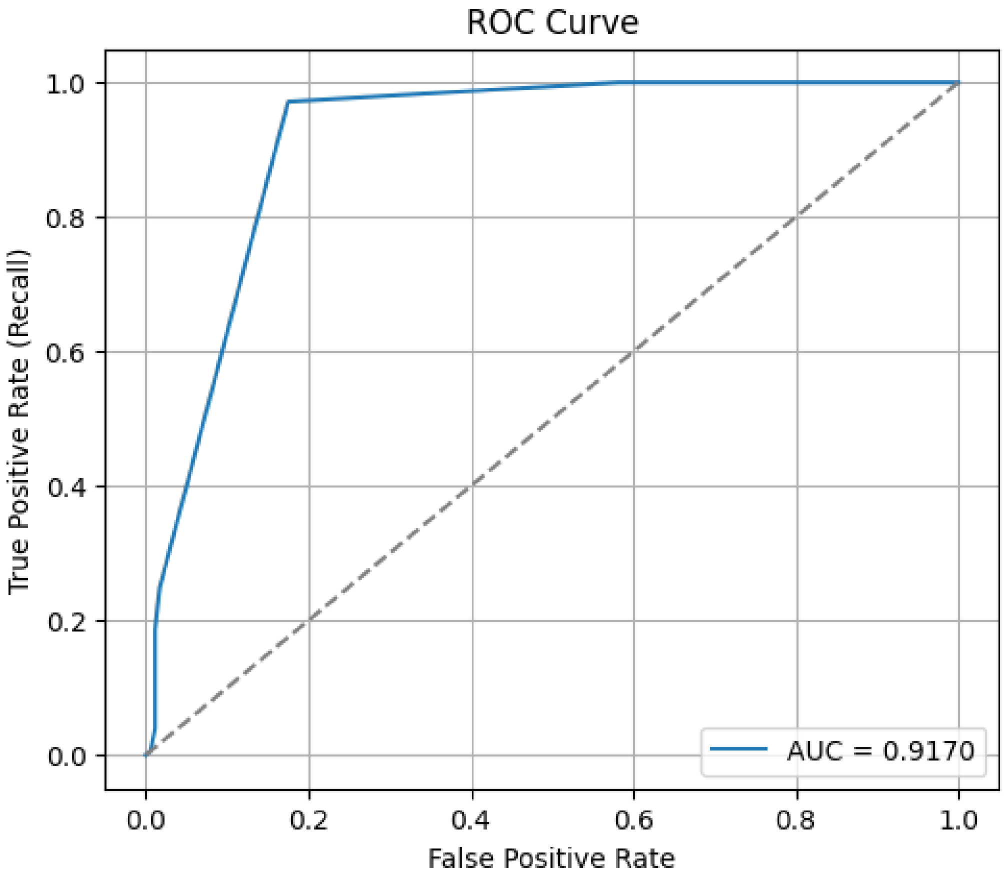
2. Business Objective

The overarching business objective for Global Insure is to enhance the **accuracy**, **speed**, and **efficiency** of its insurance claim approval process while minimizing the financial losses due to fraud. The project specifically aims to:

1. Classify Claims as Fraudulent or Legitimate

Using historical data, the goal is to build a robust classification model that can **accurately distinguish between fraudulent and legitimate claims**. This classification allows the company to:

- Flag **suspicious claims** for further manual investigation.
- Expedite **approval of legitimate claims**, ensuring better customer service.
- Reduce reliance on manual checks for every claim, increasing scalability.

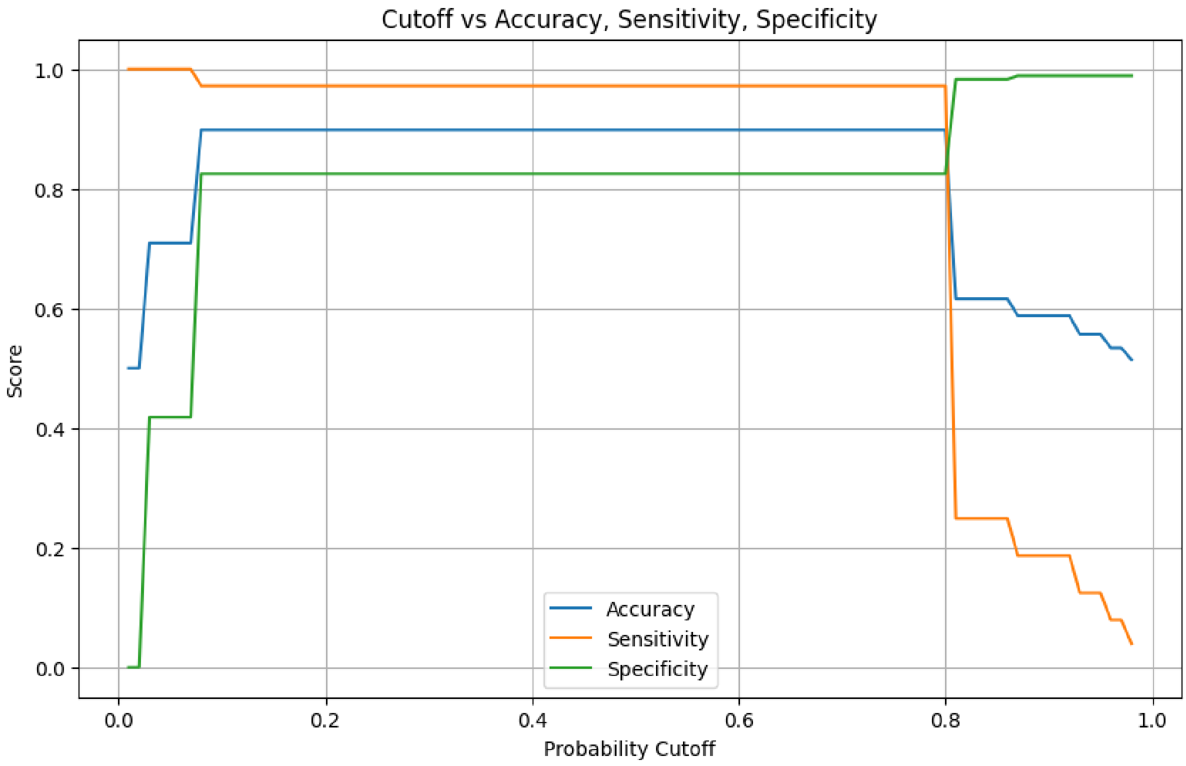


❖ *This image visualizes the performance of the classification model, particularly how well it distinguishes between the two classes.*

2. Improve Operational Efficiency and Reduce False Pay-outs

Fraudulent claims cost insurers millions annually. By predicting fraud early:

- The company avoids **unnecessary pay-outs** on ineligible claims.
- Internal teams can **focus resources** on investigating truly suspicious cases.
- **Operational costs** are reduced by limiting manual efforts on clearly legitimate claims.

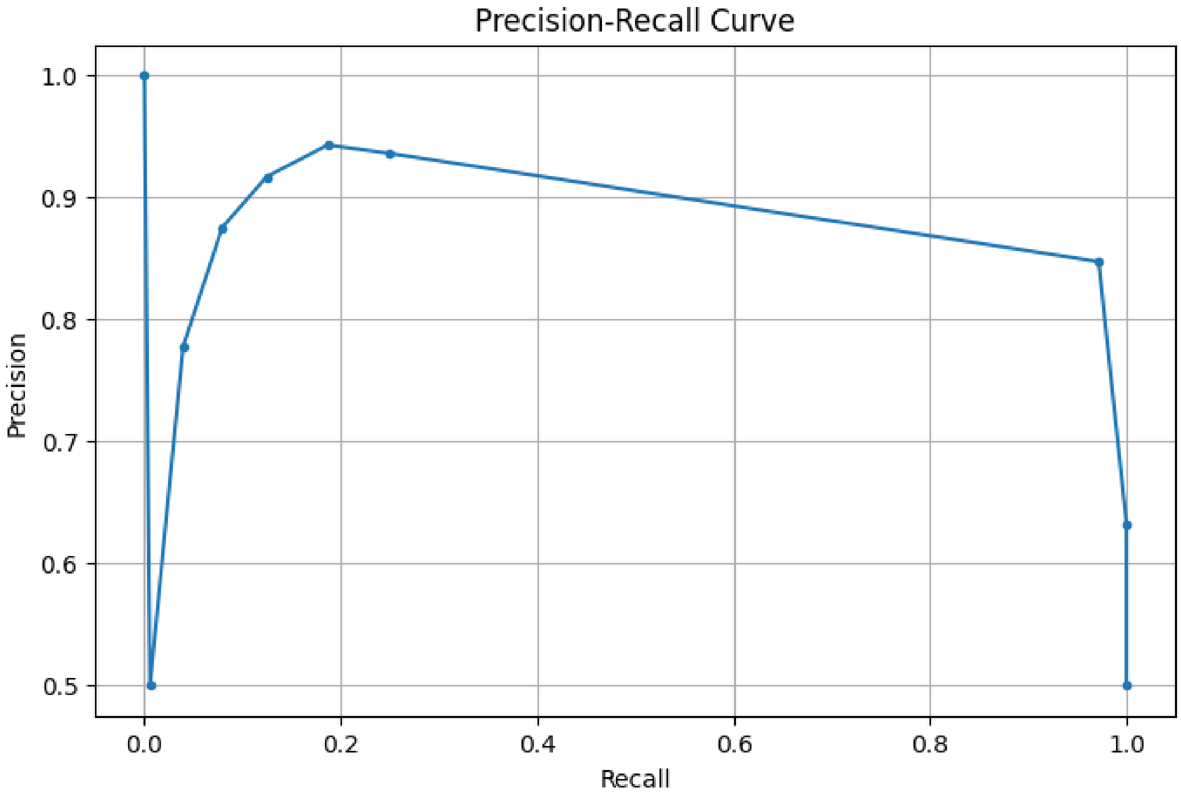


❖ *Shows how different probability thresholds impact the model's sensitivity (catching fraud) and specificity (avoiding false alarms). It supports the goal of optimizing efficiency.*

3. Accelerate Genuine Claim Approvals

Not all claims are suspicious, and holding up genuine customers negatively affects customer satisfaction and brand reputation. By automating the detection of **clearly legitimate claims**, the system can:

- Approve them **instantly** or with minimal verification.
- Improve **turnaround time** for policyholders.
- Ensure a **better customer experience**, increasing trust in the insurer.



❖ *This illustrates the trade-off between precision (approving only genuine claims) and recall (not missing frauds). It supports the business objective of faster approvals for legitimate claims while still catching fraud.*

3. Assumptions

To ensure the reliability and interpretability of the model, several data-related assumptions were made during pre-processing and feature engineering. These assumptions are listed below with justifications:

---

1. Rows with '?' Values Were Treated as Missing and Dropped

Many categorical columns contained the placeholder `'?'`, representing missing information (e.g., in fields like `authorities_contacted`, `property_damage`, etc.).

- These entries were replaced with `NaN` and dropped to avoid introducing bias or misleading patterns.
- This ensured that the model trained only on **complete and accurate records**.

2. _c39 Column Was Dropped Due to No Useful Data

The `_c39` column contained only null values across all rows:

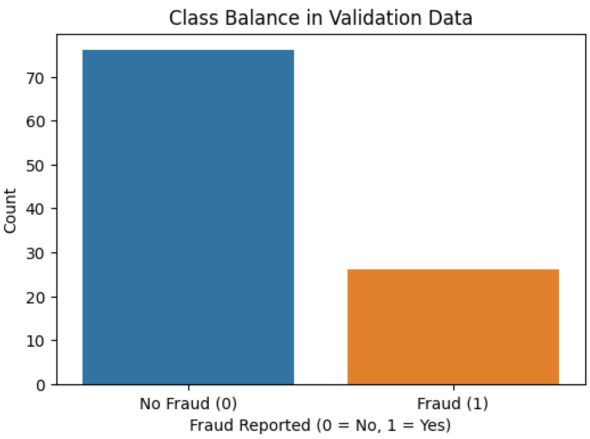
- Since it added no informational value and would contribute noise, it was **dropped early** in the data cleaning step.
- Removing such irrelevant features improves **model performance and reduces overfitting risk**.

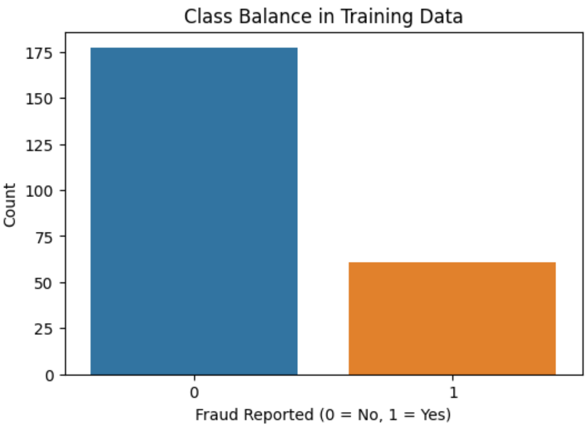
3. Stratified 70-30 Split Was Used for Balanced Training

To preserve the class distribution of the target variable (fraud_reported):

- The dataset was split into **70% training and 30% validation**, with **stratification** on the fraud label.
- This ensured that both sets reflected the **original class imbalance**, which is crucial in fraud detection tasks.

❖ *These plots visually confirm that class proportions were preserved after the split.*

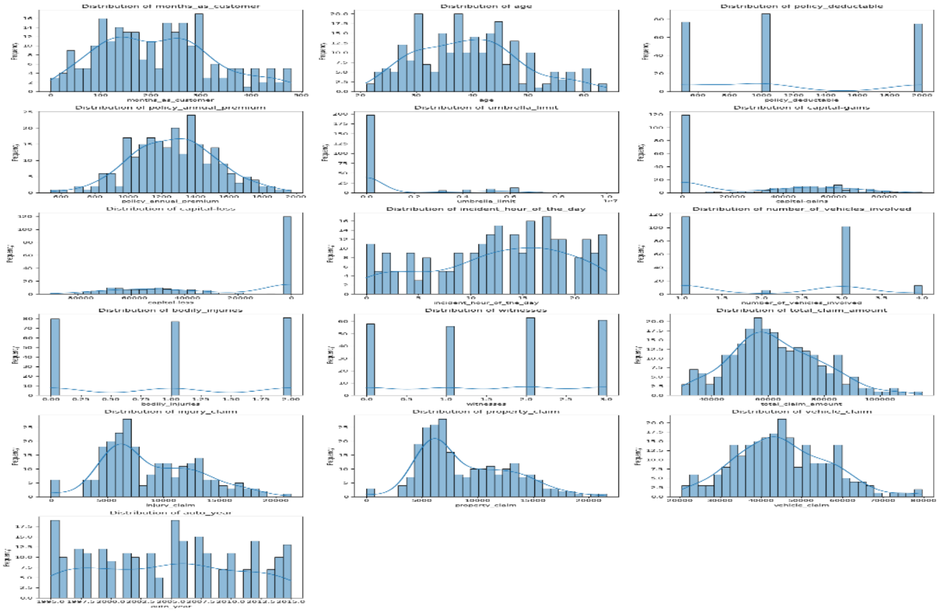




4. Only Valid Numeric Claims Retained (No Negatives)

Claims such as injury_claim, property_claim, vehicle_claim, and total_claim_amount were expected to be **positive values**:

- All rows with negative or illogical claim amounts were removed.
- This assumption was necessary to maintain **realistic claim data** and avoid misleading the model.



❖ *The boxplots show claim distributions and can visually confirm that negative values are absent after cleaning.*

5. Date Features Transformed Before Feature Engineering

Date columns like `policy_bind_date` and `incident_date` was originally in object format:

- These were **converted to datetime format** to compute meaningful features such as:
  - `days_between_policy_and_incident`
  - `is_weekend_incident`
- These derived features helped capture **temporal patterns** in fraud behavior.

4. Methodology

The project followed a structured and iterative machine learning pipeline to ensure that the predictive model was both effective and interpretable. The workflow was divided into logical stages, each serving a critical role in the development of a fraud detection system for Global Insure:

1. Data Loading and Initial Exploration

The dataset was first imported from a CSV file using pandas, followed by an initial inspection that included:

- Displaying the first few records (.head()), checking dimensions, and identifying all features.
- Understanding the data types and examining any evident anomalies (e.g., missing values, inconsistent formats).
- Reviewing the target column fraud_reported to confirm the presence of a class imbalance (fraudulent vs. legitimate).

2. Data Cleaning

This step focused on preparing the dataset for accurate analysis and model training. It involved:

- Replacing '?' characters with NaN and dropping rows with missing values.
- Dropping irrelevant or completely null columns like _c39.
- Removing identifier-like fields (e.g., policy_number, insured_zip) that don’t contribute predictive value.
- Ensuring all claim-related amounts were non-negative and within logical bounds.
- Converting string-based date columns (policy_bind_date, incident_date) into proper datetime format.

3. Train-Validation Split (70-30, Stratified)

To fairly evaluate model performance, the cleaned dataset was split into:

- **70% Training Data** – used to build and optimize models.
- **30% Validation Data** – used to simulate unseen data and evaluate generalization.

A **stratified split** was performed to preserve the proportion of fraudulent and non-fraudulent claims in both sets, which is critical in imbalanced classification problems.

4. Exploratory Data Analysis (EDA)

EDA helped uncover patterns, trends, and relationships in the data that could inform feature engineering and model design. It included:

- **Univariate analysis**: Distribution plots of numerical features.
- **Class imbalance check**: Visualization of fraud_reported distribution in both training and validation sets.
- **Correlation analysis**: Heatmaps to detect multicollinearity among numerical features.
- **Bivariate analysis**: Boxplots and target-likelihood analysis to evaluate relationships between features and fraud outcomes.

5. Feature Engineering

To improve model performance, new features were derived and data was preprocessed:

- **Created new features**:
    - days_between_policy_and_incident
    - total_claim_ratio
    - is_weekend_incident
- **Dropped redundant features**: Columns used to derive new features were removed.
- **Reduced category cardinality**: Rare values in categorical columns like auto_model and insured_hobbies were grouped under “Other”.
- **Encoded categorical features**: Dummy variables were created for all categorical fields.
- **Scaled numerical features**: Applied StandardScaler to bring all numeric values to a similar scale, avoiding bias during model training.

---

6. Model Building

Two models were trained and compared:

Logistic Regression

- **Feature selection** was done using RFECV to identify the most relevant predictors.
- **Model fitting** was done using Statsmodels, allowing access to p-values and VIFs for multicollinearity detection.
- Model thresholds were optimized based on ROC and precision-recall analysis.

Random Forest Classifier

- Trained on the top 15 most important features as identified by feature importance scores.

- **Hyperparameter tuning** was performed using GridSearchCV to identify the best model settings.
- Used class_weight='balanced' to handle class imbalance.

7. Model Evaluation and Comparison

Both models were evaluated using the following metrics:

- **Accuracy**: Overall correctness of predictions.
- **Precision**: Correctness of fraud predictions.
- **Recall (Sensitivity)**: Ability to identify all actual fraud cases.
- **Specificity**: Ability to correctly identify legitimate claims.
- **F1 Score**: Balance between precision and recall.
- **AUC-ROC**: Discriminatory power of the model.

Comparisons showed that **Logistic Regression** offered better fraud recall and generalization than the tuned Random Forest, making it the more suitable choice in this case.

5. Data Cleaning

Before model training and feature engineering, a comprehensive data cleaning process was conducted to ensure high data quality and consistency. The following steps were applied:

1. Dropped the _c39 Column

The _c39 column was completely empty, containing only null values across all rows. Since it provided no information or variability, it was dropped from the dataset.

2. Handled Null Values and Missing Entries

Several categorical columns contained '?' as placeholders for missing data. These were replaced with NaN, and rows containing such values were dropped to maintain data integrity.

- For example, columns like authorities_contacted and property_damage had non-trivial missing values.

3. Removed Identifier-like Features

Some columns such as policy_number, insured_zip, and incident_location contained unique or near-unique values per record.

- These columns do not offer predictive value and can introduce noise or overfitting in models.
- Hence, they were removed from the dataset.

4. Converted Date Columns to Proper Datetime Format

The columns policy_bind_date and incident_date were originally stored as object (string) types. These were:

- Converted to Python datetime format.
- Used to derive additional meaningful features like days_between_policy_and_incident and is_weekend_incident.

5. Dropped Rows with Invalid Numeric Values

All claim-related columns (injury_claim, property_claim, vehicle_claim, total_claim_amount) were expected to be **non-negative**:

- Records with negative or illogical values were removed, as they could bias model training or distort patterns.

6. Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) was performed to understand the structure of the dataset, identify patterns, spot anomalies, and inform feature engineering and model selection. It included the following key steps:

6.1 Class Balance Analysis

The target variable fraud_reported was examined to identify class distribution in both the training and validation datasets.

- A significant **class imbalance** was observed, with far more legitimate claims (fraud_reported = 0) than fraudulent ones (fraud_reported = 1).
- This imbalance informed the decision to use **resampling techniques** like RandomOverSampler.

Key Insight:

Fraudulent claims are underrepresented. Class imbalance must be handled to avoid biased models.

6.2 Univariate Analysis

Univariate analysis of numerical features was conducted using **histograms with KDE (Kernel Density Estimate)**.

- This helped visualize the **distribution and skewness** of features such as months_as_customer, policy_annual_premium, capital-gains, etc.
- Some variables showed skewness, which was addressed through scaling.

Purpose:

To understand each feature independently and assess preprocessing needs like transformation or scaling.

6.3 Correlation Analysis

Correlation heatmaps were generated for:

- **Training Data**
- **Validation Data**

These helped identify **linear dependencies** among numerical variables:

- Highly correlated features like injury_claim, property_claim, and vehicle_claim were observed to have strong correlation with total_claim_amount.
- Based on these insights, some features were dropped during feature engineering to reduce redundancy.

Purpose:

To detect multicollinearity and ensure the independence of input features for linear models.

6.4 Bivariate Analysis

Bivariate analysis explored how each feature related to the target variable (fraud_reported):

📌 *A. Numerical Features vs. Fraud*

- **Boxplots** were created to visualize how distributions of numeric variables differ between fraudulent and non-fraudulent claims.

- Features like total_claim_amount and incident_hour_of_the_day showed distinct patterns for fraud cases.

📌 *B. Categorical Features vs. Fraud*

- **Target likelihood analysis** was performed by calculating the proportion of fraudulent claims within each category.
- High-fraud categories included:
  - insured_hobbies: *chess, cross-fit*
  - incident_severity: *Major Damage*
  - insured_education_level: *Associate, JD*
  - incident_type: *Single Vehicle Collision*

Purpose:

To identify which features are most predictive of fraud and guide the model-building process.

7. Model Building and Evaluation

Two supervised classification algorithms were implemented and rigorously evaluated for detecting fraudulent insurance claims: **Logistic Regression** and **Random Forest**. Each model was optimized, validated, and compared using key classification metrics such as **Recall**, **Precision**, **F1 Score**, and **AUC**.

7.1 Logistic Regression

A **Logistic Regression** model was chosen for its simplicity, interpretability, and effectiveness in binary classification tasks. The development of this model followed a structured process:

🔷 *Feature Selection with RFECV*

- **Recursive Feature Elimination with Cross-Validation (RFECV)** was used to identify the most relevant predictors.
- The selection process involved 5-fold stratified cross-validation to ensure robustness.
- Features retained showed significant influence on predicting fraud and reduced noise in the dataset.

🔷 *Model Training with Statsmodels*

- The logistic model was trained using the **statsmodels** library to provide:
  - **P-values**: for assessing the statistical significance of each feature.

    - **Variance Inflation Factor (VIF)**: for identifying multicollinearity.
- Only features with **low p-values** and **acceptable VIFs** were retained in the final model to ensure statistical soundness.

Evaluation at Cutoffs 0.5 and 0.09

To optimize fraud detection, the model was evaluated at both the default cutoff (0.5) and a tuned cutoff (0.09). The tuned cutoff was selected based on the **ROC** and **Precision-Recall** tradeoff analysis.

Performance at Optimal Cutoff (0.09):

- **Recall**: 97.2% → Excellent ability to catch fraud cases
- **Precision**: 84.7% → Low false positives
- **F1 Score**: 90.5% → Strong balance between precision and recall
- **AUC (Area Under ROC Curve)**: 0.917 → Strong discriminatory power

The logistic regression model demonstrated exceptional sensitivity to fraudulent claims, making it ideal for high-risk business cases where **missing a fraud** is more costly than a false alarm.

7.2 Random Forest

Random Forest was selected as a second model due to its ability to **capture complex, non-linear relationships** and handle mixed data types effectively.

Base Model and Feature Importance

- A baseline Random Forest model was trained to compute **feature importances**, ranking features based on their contribution to decision-making.
- The **top 15 features** were selected for building a focused and efficient model.

Model Training with Tuning and Balancing

To address class imbalance and improve generalization:

- `class_weight='balanced'` was used to automatically adjust weights inversely to class frequencies.
- **Hyperparameter tuning** was performed using **GridSearchCV** across parameters like:
    - `n_estimators`
    - `max_depth`
    - `min_samples_split`

    - `min_samples_leaf`

📎 *Final Evaluation Metrics*

Performance on Validation Set:

- **Recall**: 42.3% → Moderate fraud detection rate
- **Precision**: 57.9% → Higher false positive rate than logistic model
- **F1 Score**: 48.9% → Lower overall effectiveness
- **Cross-validation accuracy**: ~94% → Good generalization but lower fraud sensitivity

The tuned Random Forest showed improvement over its untuned version but still fell short of the **high recall and precision** achieved by logistic regression.

8. Model Comparison

Metric	**Logistic Regression**	**Random Forest**
Accuracy	89.8%	77.4%
Recall	97.2%	42.3%
Precision	84.7%	57.9%
F1 Score	90.5%	48.9%
AUC	0.917	—

9. Business Impact

The implementation of the fraud detection model is expected to significantly transform the way Global Insure handles claim approvals and fraud investigation. The tangible business benefits include:

1. Potential Fraud Savings

With the Logistic Regression model achieving a **recall of 97.2%**, the system is capable of detecting nearly all fraudulent claims:

- This translates to **massive potential savings** by preventing payouts on high-risk, fraudulent claims.
- Early detection means fraud can be stopped **before funds are disbursed**, minimizing financial losses.

2. Reduced Manual Effort

- The current manual fraud detection process is time-consuming and resource-intensive.
- By automating fraud detection using data-driven insights, the company can:
  - **Reduce workload** on human investigators.
  - Reallocate manpower to handle only **critical, flagged cases**.
  - **Streamline operations** and scale efficiently as claim volumes grow.

3. Faster Claim Processing = Better Customer Experience

- Legitimate claims are often delayed due to manual review of all cases.
- The model helps **fast-track genuine claims**, allowing quick payouts and enhancing policyholder satisfaction.
- This creates a **competitive advantage** in the insurance market, where customer trust and response time are key differentiators.

4. Simple and Explainable Model

- The chosen Logistic Regression model is not only accurate but also **interpretable**:
  - Each feature’s influence can be clearly explained to stakeholders.
  - This makes the model **audit-friendly** and easier to justify during regulatory checks.
- Transparency helps build trust internally (actuaries, legal teams) and externally (regulators, auditors).

10. Conclusion

This project demonstrates the successful application of machine learning to a real-world problem in the insurance domain. From data collection to final model deployment, a comprehensive and structured process was followed:

🔷 End-to-End Pipeline:

- Raw data → Cleaning → EDA → Feature Engineering → Model Building → Evaluation → Final Deployment Strategy.

🔷 Why Logistic Regression?

- **High Recall (97.2%)**: Ensures most fraudulent claims are caught.
- **Good Precision (84.7%)**: Maintains accuracy in flagged frauds.
- **Excellent F1 Score (90.5%)**: Demonstrates model's balanced performance.
- **Interpretability**: Feature weights and statistical metrics allow full model transparency.

🔷 Deployment Readiness

- The Logistic Regression model is production-ready and ideal for integration into Global Insure's claim processing pipeline.
- It aligns with business priorities — fraud reduction, efficiency, customer satisfaction, and audit compliance.