Report: Fraudulent Claim Detection

Submitted by: Roma Agrawal & Muthukumaran Narayanasamy

Problem Statement

Global Insure struggles with identifying fraudulent claims in a timely manner, relying on manual inspections that are inefficient and prone to delays. This results in significant financial losses as fraudulent claims are often detected too late in the process.

Objectives

Global Insure aims to develop a model that classifies insurance claims as fraudulent or legitimate based on historical data and customer profiles. By analyzing features like claim amounts, customer profiles, and claim types, the company seeks to predict fraud early, improving operational efficiency and minimizing financial losses.

Below are the Steps taken:

1) Data Preparation:

a) Importing Libraries
b) Loading the data and exploring

2) Data Cleaning:

a) Null Handling
b) Redundant columns and values after examining
c) Fixing Data Types of the columns

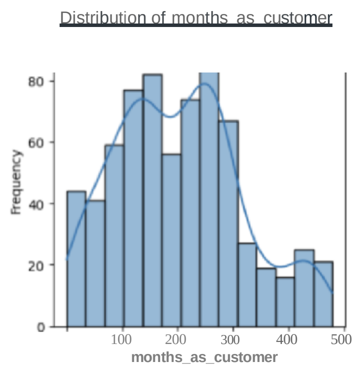
3) Train- Test Split:

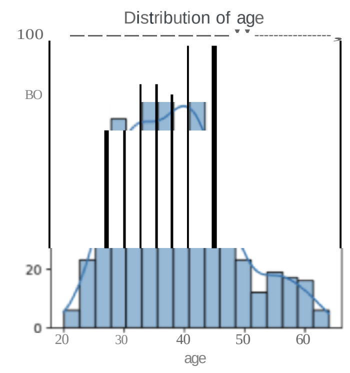
a) The dataset split into 70% train and 30% validation and using stratification on the target variable

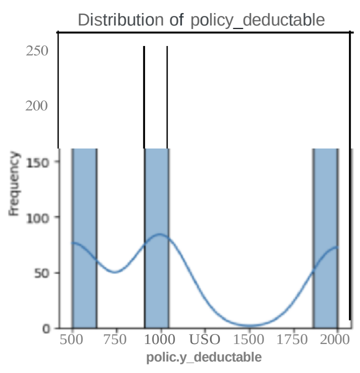
4) Exploratory Data Analysis:

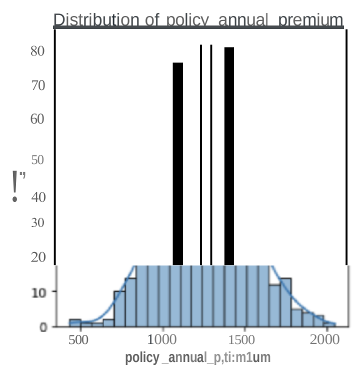
a) Univariate Analysis

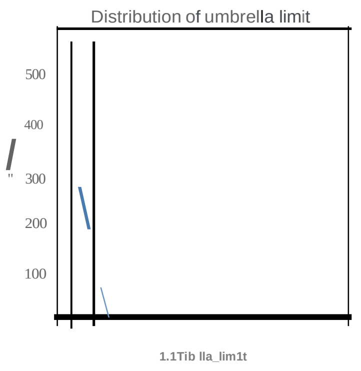
i) Defined numerical columns
ii) Plotted Histogram of numerical columns to understand their Distribution

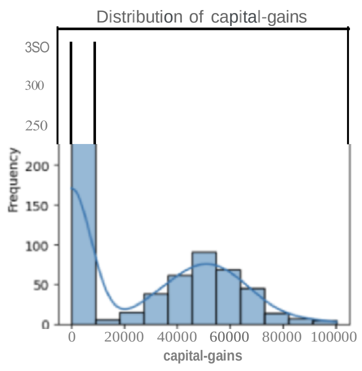

Distribution of months_as_customer
Frequency
months_as_customer

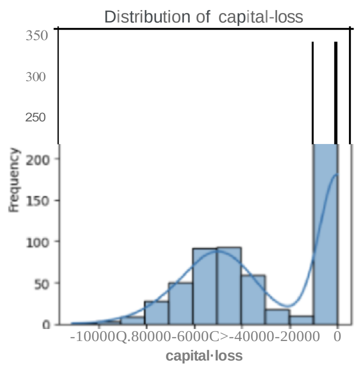

Distribution of age
age

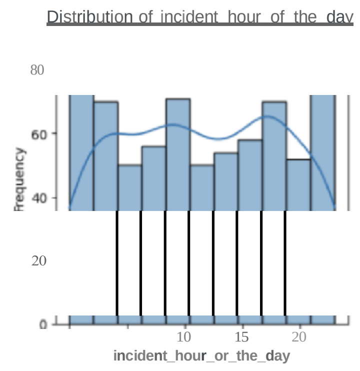

Distribution of policy_deductable
Frequency
polic.y_deductable

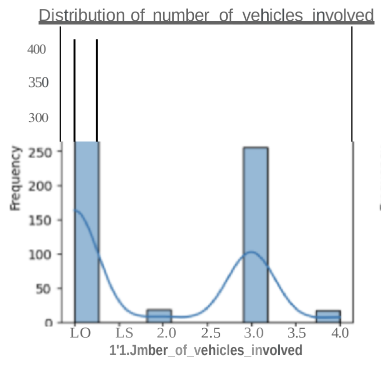

Distribution of policy_annual_premium
policy_annual_p,ti:m1um

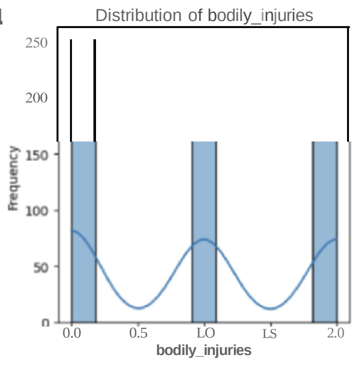

Distribution of umbrella limit
1.1Tib lla_lim1t

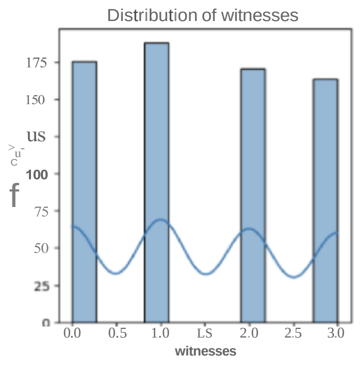

Distribution of capital-gains
Frequency
capital-gains

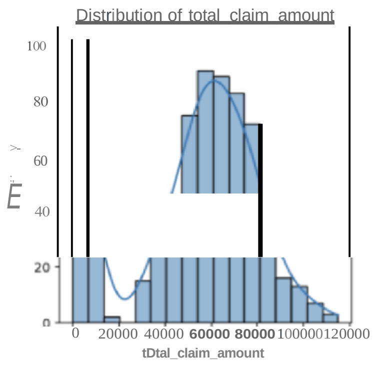

Distribution of capital-loss
Frequency
capital·loss

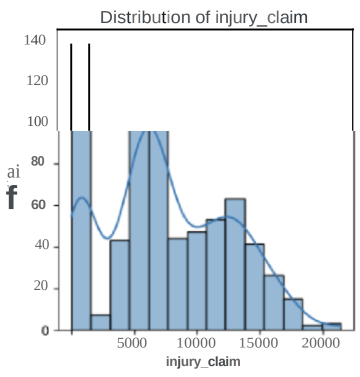

Distribution of incident_hour_of_the_day
Frequency
incident_hour_or_the_day

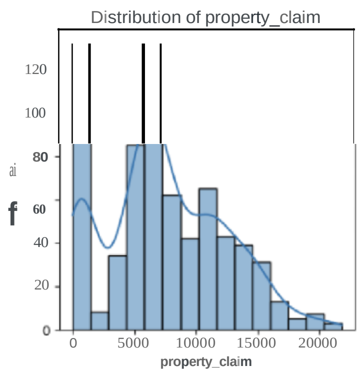

Distribution of number_of_vehicles_involved
Frequency
1'1.Jmber_of_vehicles_involved

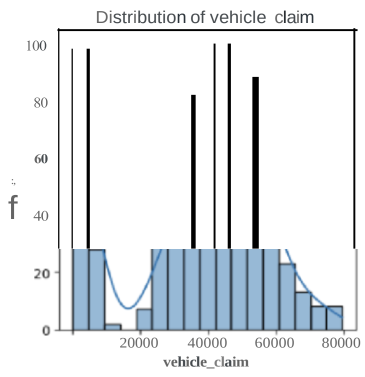

Distribution of bodily_injuries
Frequency
bodily_injuries

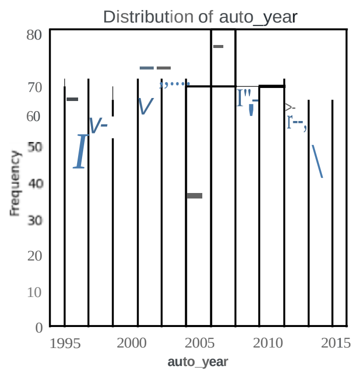

Distribution of witnesses
witnesses

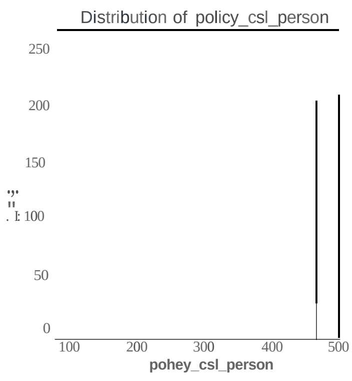

Distribution of total_claim_amount
tDtal_claim_amount

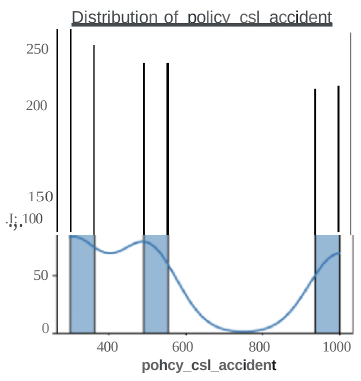

Distribution of injury_claim
injury_claim


Distribution of property_claim
property_claim


Distribution of vehicle claim
vehicle_claim

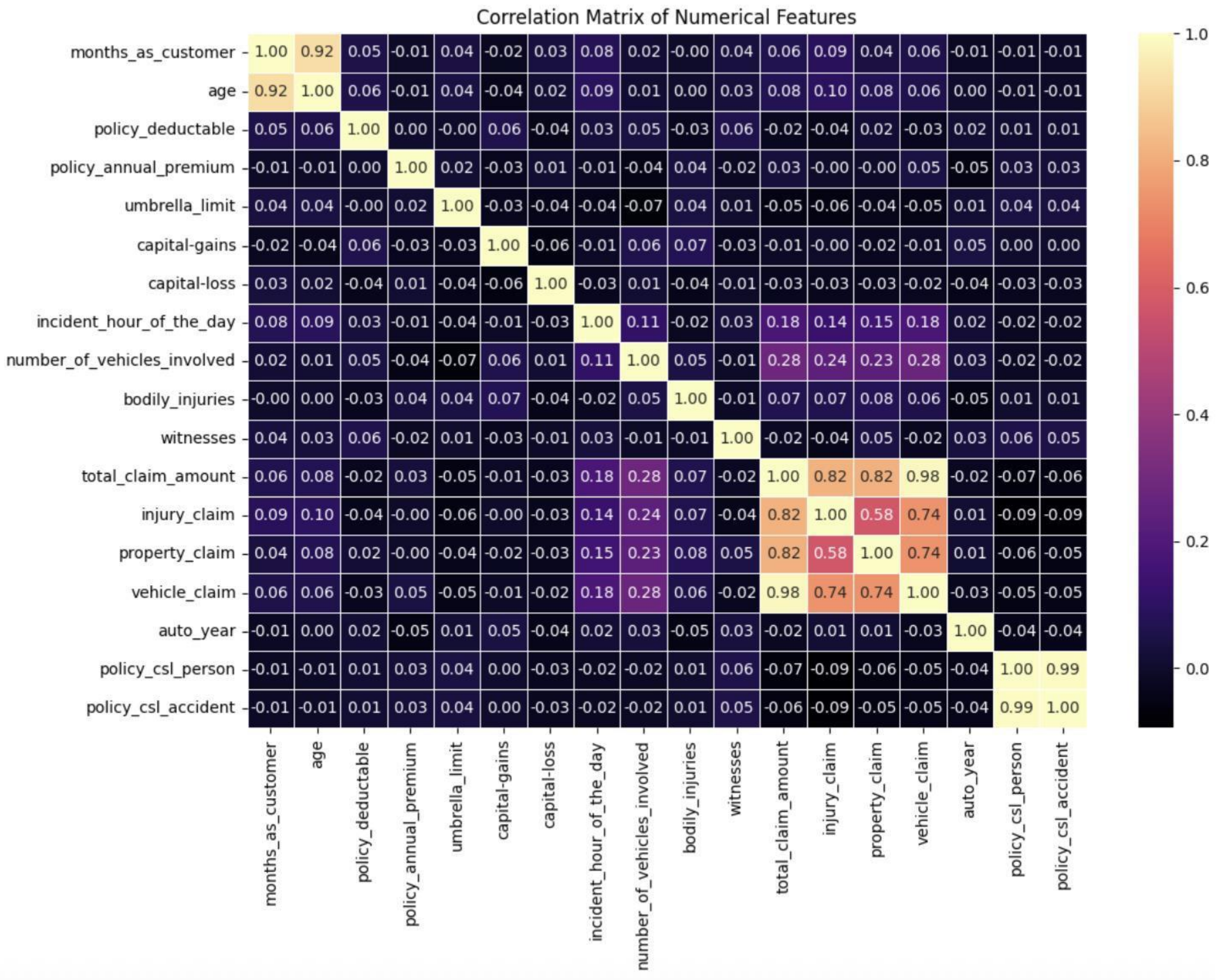

Distribution of auto_year
Frequency
auto_year


Distribution of policy_csl_person
pohey_csl_person


Distribution of policy_csl_accident
pohcy_csl_accident

5) Correlation Analysis for Numeric Features:

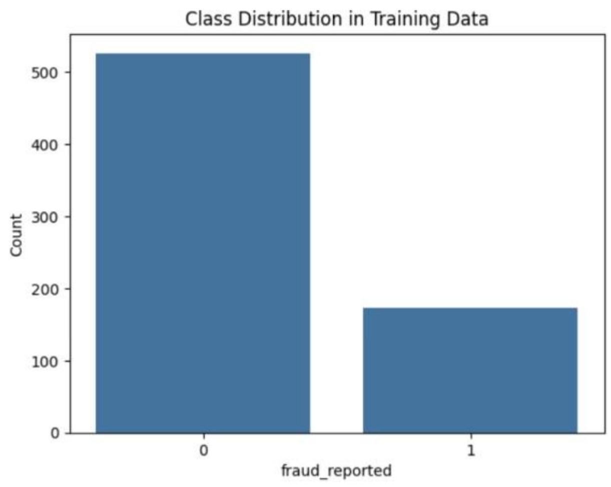
a) Below is the correlation matrix vis, and based on this performed feature engineering in upcoming steps.



	months_as_customer	age	policy_deductable	policy_annual_premium	umbrella_limit	capital-gains	capital-loss	incident_hour_of_the_day	number_of_vehicles_involved	bodily_injuries	witnesses	total_claim_amount	injury_claim	property_claim	vehicle_claim	auto_year	policy_csl_person	policy_csl_accident
months_as_customer	1.00	0.92	0.05	-0.01	0.04	-0.02	0.03	0.08	0.02	-0.00	0.04	0.06	0.09	0.04	0.06	-0.01	-0.01	-0.01
age	0.92	1.00	0.06	-0.01	0.04	-0.04	0.02	0.09	0.01	0.00	0.03	0.08	0.10	0.08	0.06	0.00	-0.01	-0.01
policy_deductable	0.05	0.06	1.00	0.00	-0.00	0.06	-0.04	0.03	0.05	-0.03	0.06	-0.02	-0.04	0.02	-0.03	0.02	0.01	0.01
policy_annual_premium	-0.01	-0.01	0.00	1.00	0.02	-0.03	0.01	-0.01	-0.04	0.04	-0.02	0.03	-0.00	-0.00	0.05	-0.05	0.03	0.03
umbrella_limit	0.04	0.04	-0.00	0.02	1.00	-0.03	-0.04	-0.04	-0.07	0.04	0.01	-0.05	-0.06	-0.04	-0.05	0.01	0.04	0.04
capital-gains	-0.02	-0.04	0.06	-0.03	-0.03	1.00	-0.06	-0.01	0.06	0.07	-0.03	-0.01	-0.00	-0.02	-0.01	0.05	0.00	0.00
capital-loss	0.03	0.02	-0.04	0.01	-0.04	-0.06	1.00	-0.03	0.01	-0.04	-0.01	-0.03	-0.03	-0.03	-0.02	-0.04	-0.03	-0.03
incident_hour_of_the_day	0.08	0.09	0.03	-0.01	-0.04	-0.01	-0.03	1.00	0.11	-0.02	0.03	0.18	0.14	0.15	0.18	0.02	-0.02	-0.02
number_of_vehicles_involved	0.02	0.01	0.05	-0.04	-0.07	0.06	0.01	0.11	1.00	0.05	-0.01	0.28	0.24	0.23	0.28	0.03	-0.02	-0.02
bodily_injuries	-0.00	0.00	-0.03	0.04	0.04	0.07	-0.04	-0.02	0.05	1.00	-0.01	0.07	0.07	0.08	0.06	-0.05	0.01	0.01
witnesses	0.04	0.03	0.06	-0.02	0.01	-0.03	-0.01	0.03	-0.01	-0.01	1.00	-0.02	-0.04	0.05	-0.02	0.03	0.06	0.05
total_claim_amount	0.06	0.08	-0.02	0.03	-0.05	-0.01	-0.03	0.18	0.28	0.07	-0.02	1.00	0.82	0.82	0.98	-0.02	-0.07	-0.06
injury_claim	0.09	0.10	-0.04	-0.00	-0.06	-0.00	-0.03	0.14	0.24	0.07	-0.04	0.82	1.00	0.58	0.74	0.01	-0.09	-0.09
property_claim	0.04	0.08	0.02	-0.00	-0.04	-0.02	-0.03	0.15	0.23	0.08	0.05	0.82	0.58	1.00	0.74	0.01	-0.06	-0.05
vehicle_claim	0.06	0.06	-0.03	0.05	-0.05	-0.01	-0.02	0.18	0.28	0.06	-0.02	0.98	0.74	0.74	1.00	-0.03	-0.05	-0.05
auto_year	-0.01	0.00	0.02	-0.05	0.01	0.05	-0.04	0.02	0.03	-0.05	0.03	-0.02	0.01	0.01	-0.03	1.00	-0.04	-0.04
policy_csl_person	-0.01	-0.01	0.01	0.03	0.04	0.00	-0.03	-0.02	-0.02	0.01	0.06	-0.07	-0.09	-0.06	-0.05	-0.04	1.00	0.99
policy_csl_accident	-0.01	-0.01	0.01	0.03	0.04	0.00	-0.03	-0.02	-0.02	0.01	0.05	-0.06	-0.09	-0.05	-0.05	-0.04	0.99	1.00

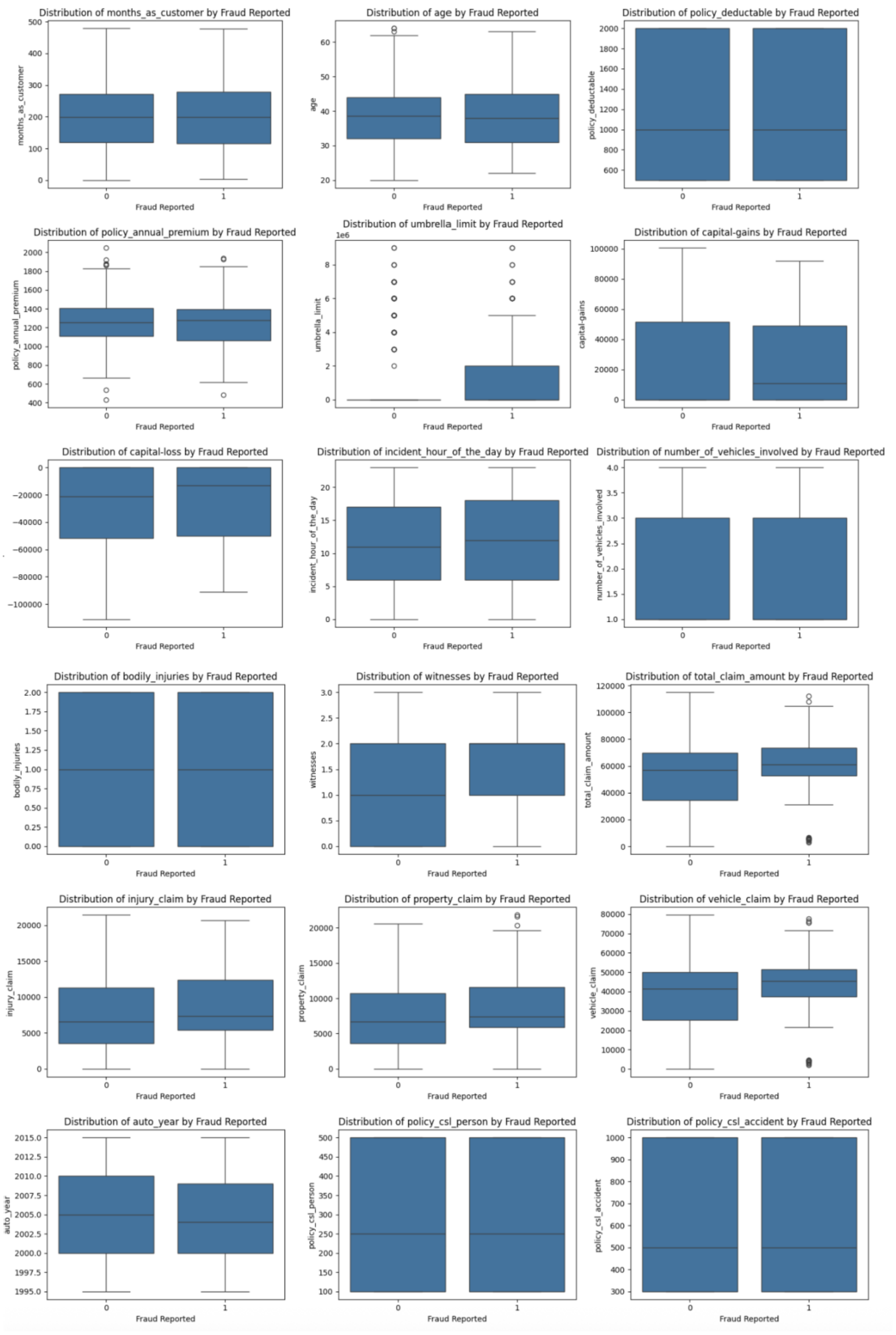
6) Class Balance Distribution:

a) So there is clear class imbalance in the data, shown below:



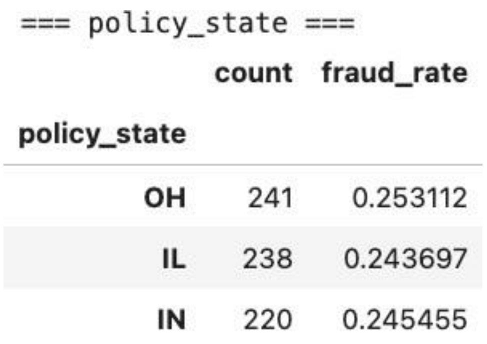
7) Bivariate Analysis:

a) Below is the visualisation of the relationship between numerical features and the target variable to understand their impact on the target outcome



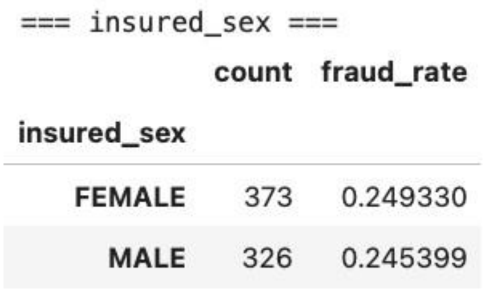
8) **Target Likelihood Analysis for all categorical features:**

a) Categorical features shows the fraud rates for different categories, such as policy state and insured sex, providing insights into how these features relate to fraudulent behaviour.



=== policy_state ===

policy_state	count	fraud_rate
OH	241	0.253112
IL	238	0.243697
IN	220	0.245455



=== insured_sex ===

insured_sex	count	fraud_rate
FEMALE	373	0.249330
MALE	326	0.245399

9) **Feature Engineering:**

a) **Resampling**: To address class imbalance, **RandomOverSampler** was used to increase the number of fraudulent claims in the training set, improving the model's ability to predict the minority class.

b) Feature Creation: New features were created based on time-based, claim-related, and customer-related data to enhance the model’s ability to capture patterns indicative of fraud.

c) Handel Redundant Columns: To improve model efficiency and reduce noise, redundant features were identified and removed, such as those with high correlation, minimal predictive value, or duplication of previously extracted features, including date-related columns and various claim-related attributes.

d) Combine Values in Categorical Column: Categorical features with infrequent values or minimal predictive power were combined, using a threshold of 95 occurrences per category or a 3% difference in fraud rate from the global average, to improve model generalization and reduce sparsity.

e) Dummy Variable Creation: Categorical columns such as 'policy_state', 'insured_sex', and others were identified for dummy variable creation using one-hot encoding, converting them into numerical representations and ensuring consistent encoding between the training and validation datasets.

f) Feature Scaling: Numerical features were scaled using StandardScaler for Logistic Regression to ensure consistent value ranges, while unscaled features were kept as copies for use in Random Forest to maintain the model's compatibility with unscaled data.

10) **Model Building:**

a) **Feature selection (Logit):**

i) RFECV was applied to identify the most relevant features for the Logistic Regression model.

ii) The model was trained using StratifiedKFold cross-validation with the **ROC-AUC score** as the performance metric.

iii) **11** features were retained from the initial **109** features, which were found to be the most influential for predicting fraudulent claims.

iv) RFECV helps ensure that the model is not impacted by irrelevant features, improving model performance and reducing overfitting.

11) **Build Logistic Regression Model:**

a) The relevant features were selected using **RFECV** and a constant (intercept) was added to the training data to account for the baseline level of fraud.

b) The **Logistic Regression model** was built using the selected features, and the coefficients for each feature, along with p-values, were analysed to assess their significance. The model successfully converged, with the **log-likelihood** of -379.85 and a **pseudo R-squared value of 0.4791.**

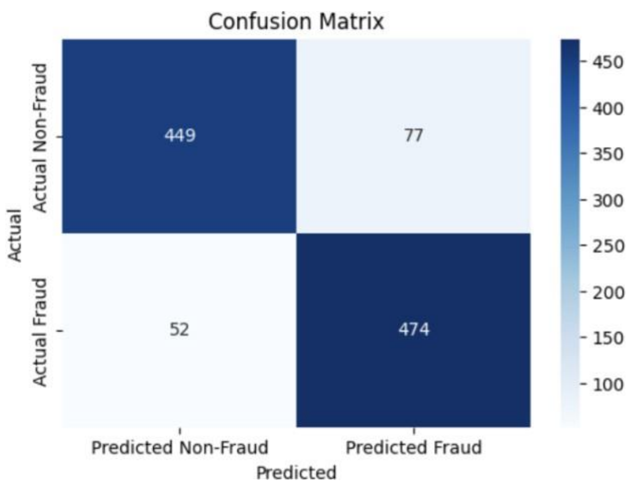
c) VIFs were calculated for each feature to detect **multicollinearity**, indicating no significant multicollinearity.

d) Predictions were made on the training data using the fitted Logistic Regression model to predict the probability of fraudulent claims. The probabilities were stored in the variable **y_train_pred.**

e) A Data Frame was created with the **actual fraud reported flags** and **predicted probabilities**. A new column was added to indicate **predicted classifications** based on a cutoff value of **0.5**, where values above 0.5 were classified as fraudulent claims.

f) The **accuracy of the Logistic Regression model** was calculated on the training data, yielding an **accuracy score of 87.74%**, indicating that the model correctly classified approximately **88% of claims** as either fraudulent or legitimate.

g) **Confusion Matrix:**



h) **Performance Metrics:**

i) **Sensitivity (Recall)**: The model correctly identifies **90.11% of fraudulent claims**.
ii) **Specificity**: The model correctly identifies **85.36% of legitimate claims**.
iii) **Precision**: **86.03% of the flagged fraudulent claims** were actually fraudulent.
iv) **F1 Score**: with **88.02%**, the model balances both **precision** and **recall**, achieving a solid F1 score.

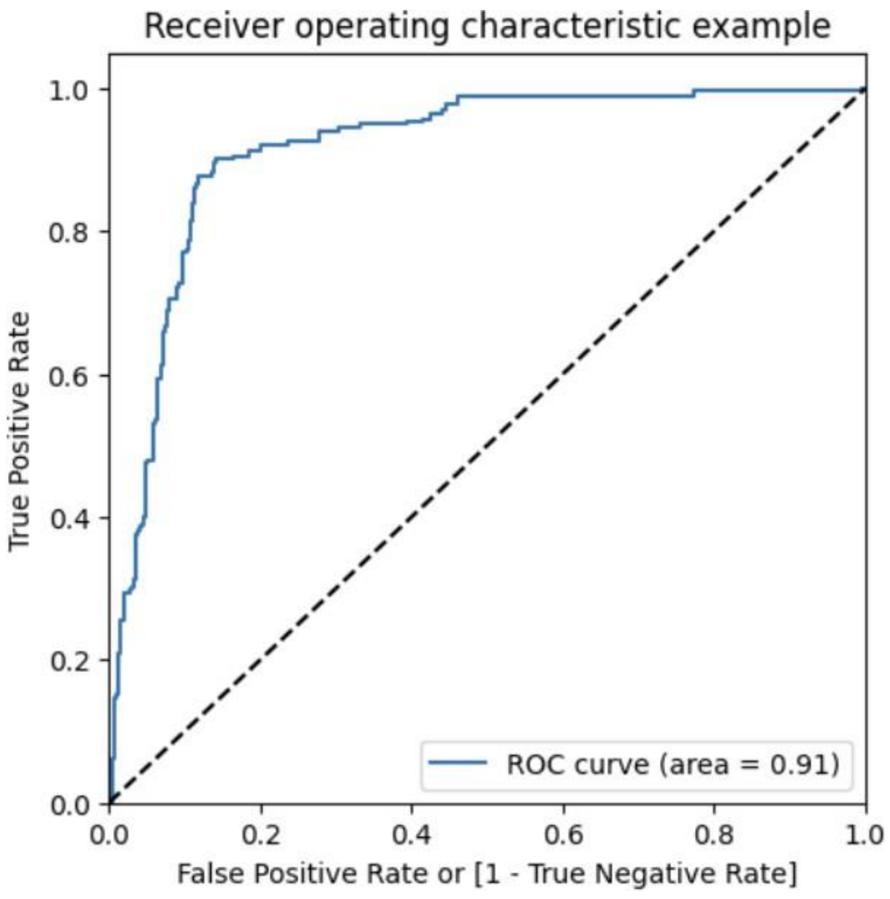
12) Find the Optimal Cutoff:

a) **ROC Curve Plot**: The **ROC curve** was plotted to visualize the trade-off between the **true positive rate** (sensitivity) and **false positive rate** (1-specificity) across different thresholds, helping to understand the model's overall performance and identify the area under the curve (AUC).

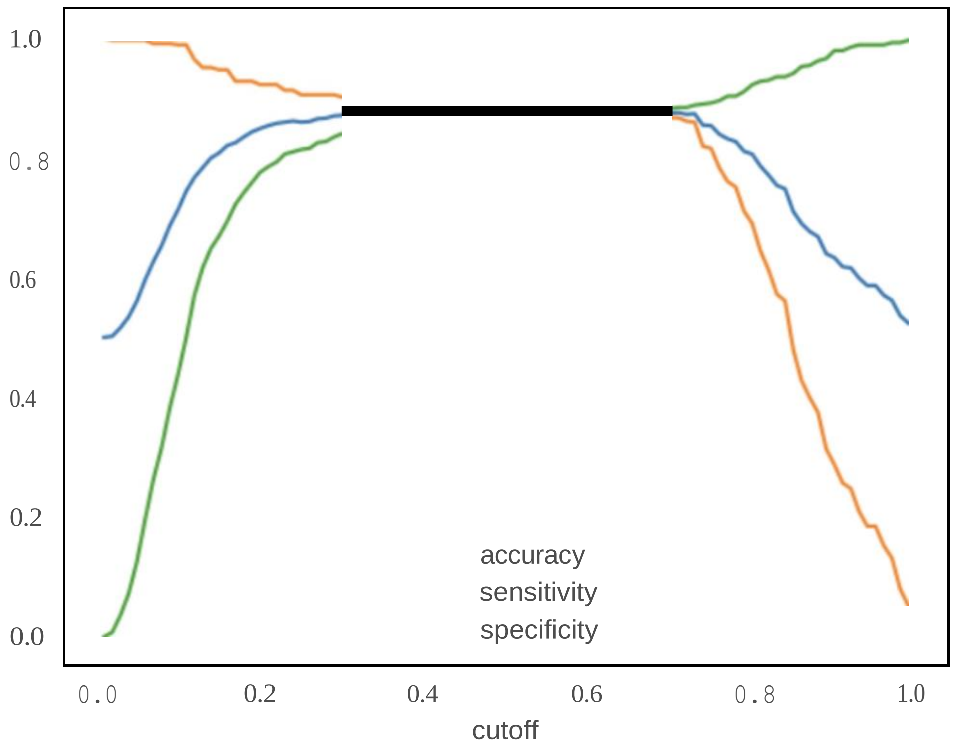
b) **Sensitivity-Specificity Tradeoff**: The **optimal cutoff** was determined by analyzing the sensitivity-specificity tradeoff across various probability cutoffs. The **best cutoff** (0.54) was found using the **Youden index**, maximizing both sensitivity and specificity.

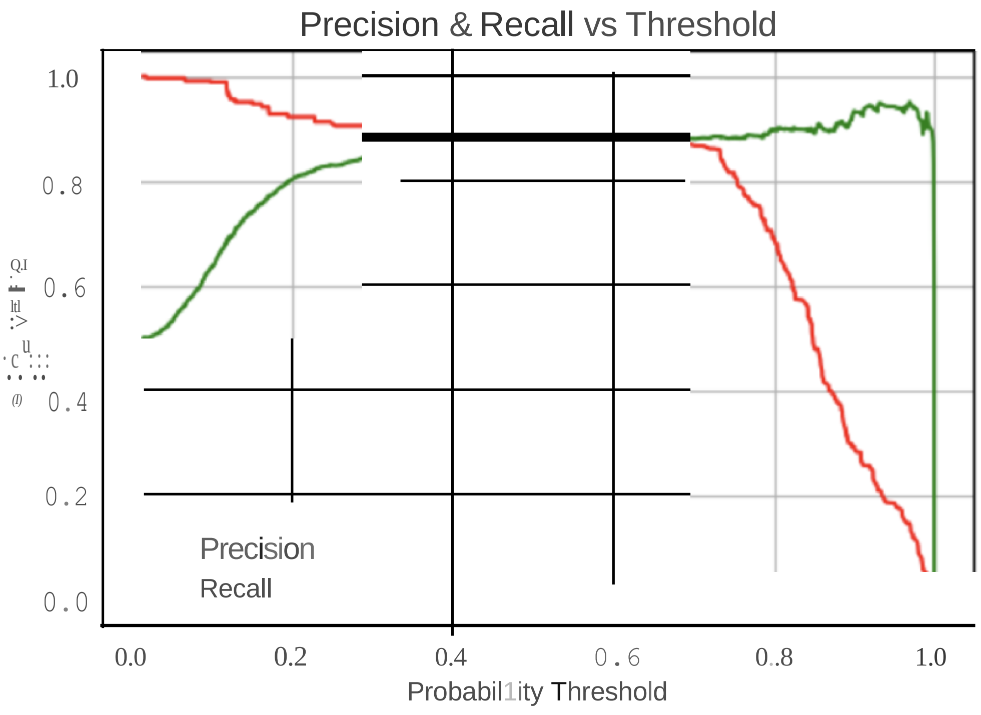
c) **Precision-Recall Curve**: The **precision-recall curve** was plotted to evaluate the tradeoff between precision and recall across different cutoffs. The **optimal cutoff** based on the **F1 score** was calculated, showing that the cutoff of **0.567** provided the best balance between precision and recall.

Below are the charts visualizing the ROC curve**,** accuracy, sensitivity, and specificity at various cutoffs**, and the** precision-recall curve **for optimal threshold selection.**



	cutoff	accuracy	sensitivity	specificity	gap
0	0.01	0.500000	1.000000	0.000000	1.000000
1	0.02	0.501901	0.996198	0.007605	0.988593
2	0.03	0.516160	0.996198	0.036122	0.960076
3	0.04	0.534221	0.996198	0.072243	0.923954
4	0.05	0.560837	0.996198	0.125475	0.870722





13) **Build Random Forest Model:**

a) Initially, a base Random Forest model was built using unscaled features (saved before using for Logistic Regression).

b) The model was trained using the entire set of features, which was followed by feature selection based on importance scores.

c) After feature selection, a final model was trained on the top 15 important features, ensuring that only the most relevant features were used.

d) **Predictions** were made on the training data, resulting in an **accuracy of 88.97%**.

e) A **confusion matrix** was used to evaluate model performance, and key metrics like **sensitivity**, **specificity**, **precision**, **recall**, and **F1-score** were calculated.

f) **Cross-validation** showed a **mean accuracy of 92.30%** ± 0.83%, confirming the model's ability to generalize well without overfitting.

14) **Hyperparameter Tuning:**

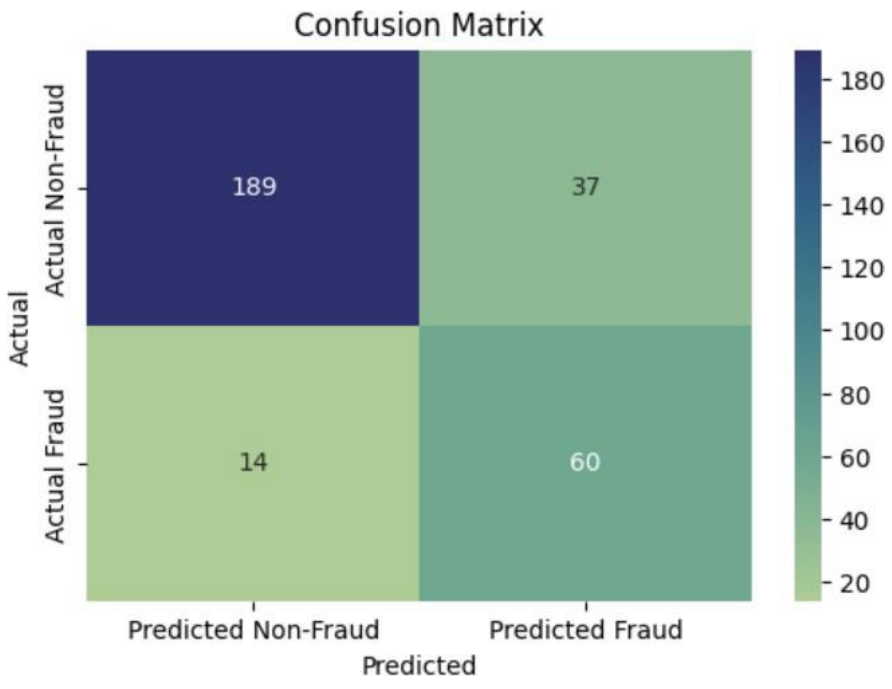
a) **Grid Search to Find the Best Hyperparameter Values**: To optimize the performance of the **Random Forest** model, **Grid Search** was used to systematically explore a range of hyperparameters. The following parameters were tuned:

- **max_depth**, **min_samples_split**, **min_samples_leaf**, **max_features**, **n_estimators**, and **oob_score**.
- **StratifiedKFold cross-validation** with 5 folds was used to ensure reliable evaluation during hyperparameter search
- The **best hyperparameters** found through grid search were:
    - **max_depth = 10**
    - **min_samples_split = 10**
    - **min_samples_leaf = 10**
    - **n_estimators = 200**
    - **n_jobs = -1**
    - **oob_score = True**

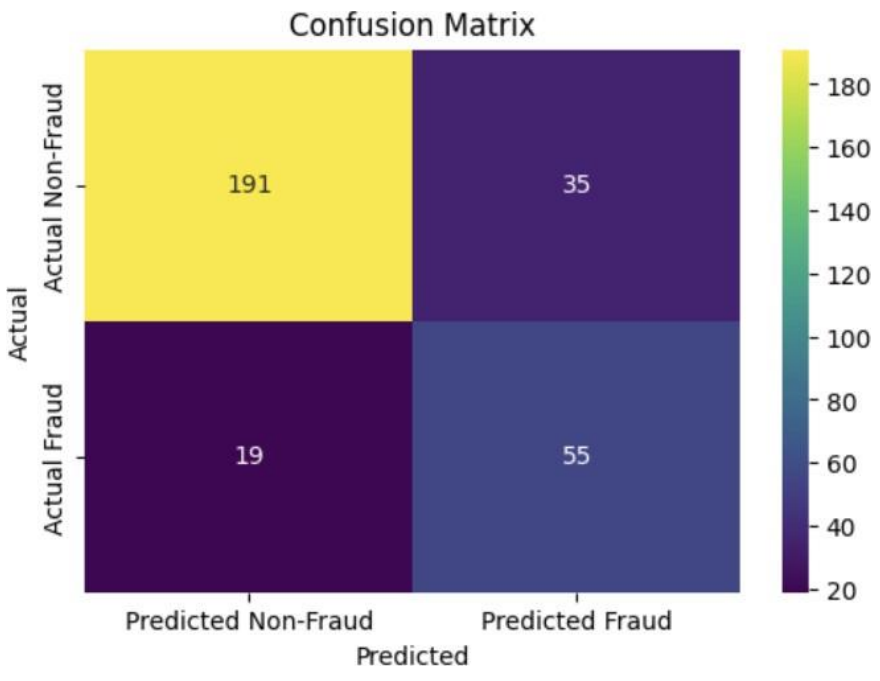
b) **Model Evaluation After Hyperparameter Tuning:** The tuned Random Forest model was trained on the top 15 selected features, achieving an accuracy of 92.21% on the training data, with strong metrics for sensitivity (88.21%), specificity (88.67%), precision (92.21%), and F1-score (90.40%).

Evaluation od Both Models on Validation:

Confusion Matrix for Logit:



Confusion Matrix for RF:



Recall (Sensitivity):

- Logistic Regression (Logit) has a higher recall of 81.08%, meaning it successfully identifies 81% of fraudulent claims. This is crucial for minimizing the chances of fraud going undetected.
- In comparison, Random Forest (RF) has a recall of 74.32%, meaning that some fraudulent claims still go undetected, which is a concern in fraud detection.

Precision:

- With 61.86% precision, Logit flags 62% of the claims it identifies as fraudulent correctly. This is quite reliable for fraud detection.
- RF, with a slightly lower precision of 61.11%, still does well in identifying fraudulent claims but is a bit less accurate than Logit.

F1 Score:

- Logit has an F1 score of 70.18%, showing a decent balance between precision and recall. This means it effectively detects fraudulent claims while minimizing false positives.
- RF's F1 score of 67.07% is slightly lower, indicating it is not as well-balanced in identifying fraud while minimizing mistakes, comparative to Logit.

Specificity:

- Logit achieves 83.63% specificity, correctly identifying legitimate claims with minimal false positives. This ensures that genuine claims are processed without delays.
- RF performs slightly better with 84.51% specificity, but it sacrifices recall for fraud Detection, meaning its better at identifying legitimate claims but misses more fraudulent Ones.

Conclusion:

When it comes to detecting fraudulent insurance claims, the **Logit** model stands out as the better option for achieving the business goal of identifying fraud early. Here's why:

- **Recall (81%)**: Logit does a fantastic job of catching a high proportion of fraudulent claims (class 1). With **81% recall**, it reduces the risk of fraud slipping through the cracks, which is exactly what we need to prevent financial losses.

- **Precision and F1 Score**: While the precision of **Logit** (61.86%) is a bit lower, the higher recall is the real winner here. It's more important to catch **as many fraudulent claims as possible** as to risk missing them. The **F1 score** of 70.18% shows that Logit strikes a solid balance between both precision and recall.

- **Specificity (83.63%)**: Logit also does a good job of **distinguishing legitimate claims** from fraudulent ones. With **83.63% specificity**, it ensures that **real claims** aren't unfairly flagged, making sure that the claims process runs smoothly.

On the other hand, while **RF** performs well at **identifying legitimate claims** (88.88% specificity), it falls short in catching fraud. With only **66.67% recall**, it misses a significant portion of fraudulent claims, making it less reliable for **fraud detection**.

In conclusion, **Logit** is better suited for the job, as it meets the business goal of detecting fraud early, preventing financial losses, and ensuring legitimate claims aren’t unnecessarily delayed.