FRAUDULENT INSURANCE CLAIM DETECTION

PREPARED BY:

- **Booragadda Dheeraj**
- **Aishwarya Dwivedi**
- **Harshal shriwas**

TABLE OF CONTENTS

1. Executive Summary
2. Introduction
3. Data Understanding and Preparation
4. Exploratory Data Analysis (EDA) - Training Data
5. Exploratory Data Analysis (EDA) - Validation Data
6. Feature Engineering
7. Model Development and Evaluation (Training Phase)
8. Final Model Evaluation on Validation Data
9. Discussion and Interpretation of Results
10. Conclusions and Recommendations

1. EXECUTIVE SUMMARY

Global Insure suffers from substantial financial losses due to fraudulent insurance claims. Its manual fraud detection process is inefficient, often identifying fraud too late.

This project developed a machine learning pipeline to detect potentially fraudulent claims early.

By analyzing 1000 historical claims with 40 features, two models—**Logistic Regression** and **Random Forest**—were trained and evaluated.

The Logistic Regression model, with a tuned cutoff of 0.49, achieved the best performance on unseen data (accuracy: **82.00%**, recall: **71.62%**, precision: **61.63%**).

Key fraud indicators included incident severity ("Major Damage"), insured hobbies (e.g., "chess," "cross-fit"), witness count, and policy deductible.

This model provides actionable intelligence to prioritize high-risk claims for further review.

2. INTRODUCTION

Problem Statement

Global Insure, a leading insurance company, processes thousands of claims annually. However, a significant percentage of these claims turn out to be fraudulent, resulting in considerable financial losses. The company's current process for identifying fraudulent claims involves manual inspections, which is time-consuming and inefficient. Fraudulent claims are often detected too late in the process, after the company has already paid out significant amounts.

Business Objective

Global Insure wants to build a model to classify insurance claims as either fraudulent or legitimate based on historical claim details and customer profiles. By using features like claim amounts, customer profiles, and claim types, the company aims to predict which claims are likely to be fraudulent before they are approved. This would minimise financial losses and optimise the overall claims handling process.

Key Business Questions

- What patterns in historical claims indicate fraud?
- Which features are most predictive?
- Can fraud be predicted reliably for new claims?
- What model insights can improve the fraud detection process?

3. DATA UNDERSTANDING AND PREPARATION

Data Source

The primary data source for this project is the insurance_claims.csv file, containing 1000 records of insurance claims, each with 40 initial attributes describing customer profiles, policy details, incident specifics, and claim amounts.

Data Dictionary Overview

The dataset encompasses a rich set of variables including customer tenure, age, policy details (state, CSL, deductible, premium, umbrella limit), insured demographics (zip, sex, education, occupation, hobbies, relationship), financial information (capital gains/losses), incident details (date, type, collision, severity, authorities contacted, location, time), vehicle information, and the target variable fraud_reported. A detailed data dictionary was provided and referenced throughout the analysis. (A summary table or full dictionary could be appended).

Initial Exploration

- Dataset shape: 1000 rows × 40 columns
- Data types: int64, float64, object
- Data types were a mix of integer (int64), float (float64), and string/object (object).
- Several object columns represented dates or categorical information.
- Null values were present in authorities_contacted (91 nulls) and _c39 (1000 nulls, i.e., entirely empty).
- Special placeholder characters like '?' were observed in some categorical columns.

Data Cleaning

A systematic data cleaning process was undertaken to prepare the data for analysis and modeling:

Handling Null Values:

- The _c39 column, being entirely empty, was dropped as it provided no information.
- For authorities_contacted, missing values (91 instances) were imputed with "Police", a common and plausible category within the existing values for this feature. This ensured no data loss from row deletion while handling missingness.

Handling Redundant/Special Values:

- The placeholder '?' character was identified in collision_type, property_damage, and police_report_available. These were replaced with meaningful categories:

- collision_type: '?' replaced with 'Not Applicable' (assuming '?' meant the field wasn't relevant, e.g., for vehicle theft).
- property_damage: '?' replaced with 'Unknown'.
- police_report_available: '?' replaced with 'Unknown'.
- One row containing an illogical negative value for umbrella_limit (-1,000,000) was identified and removed, reducing the dataset to 999 rows. This ensures data integrity for numerical features that logically cannot be negative.

Handling High-Cardinality/Identifier Columns:

- policy_number and incident_location were identified as unique identifiers for each record and thus dropped as they offer no generalizable predictive power.
- Columns like policy_bind_date (for policy tenure calculation), policy_annual_premium (numerical feature), and insured_zip (potential for geographical feature engineering or as a categorical feature) were initially retained for further processing, despite high cardinality.

Fixing Data Types:

- Policy_bind_date and incident_date columns, initially objects, were converted to datetime objects to enable date-based feature engineering.
- All remaining columns with an 'object' data type were converted to 'category' type for efficient storage and appropriate handling by analytical and modeling libraries.

Train-Validation Split

To evaluate model performance on unseen data, the cleaned dataset (999 rows) was split into training (70%) and validation (30%) sets.

Methodology: train_test_split from sklearn.model_selection was used.

Stratification: Stratification was applied on the target variable fraud_reported to ensure that the proportion of fraudulent ('Y') and non-fraudulent ('N') claims was similar in both the training and validation sets. This is crucial for reliable model evaluation, especially with an imbalanced target.

- random_state=42: Used to ensure reproducibility of the split.

Final Shapes:

- X_train: (699 rows, 36 feature columns)
- y_train: (699 rows, 1 target column)
- X_val: (300 rows, 36 feature columns)
- y_val: (300 rows, 1 target column)

(Note: The number of feature columns (36) reflects the state after initial cleaning and before extensive feature engineering like dummification).

4. EXPLORATORY DATA ANALYSIS (EDA) – TRAINING DATA

Objectives of EDA

The primary objectives of conducting EDA on the training data were to:

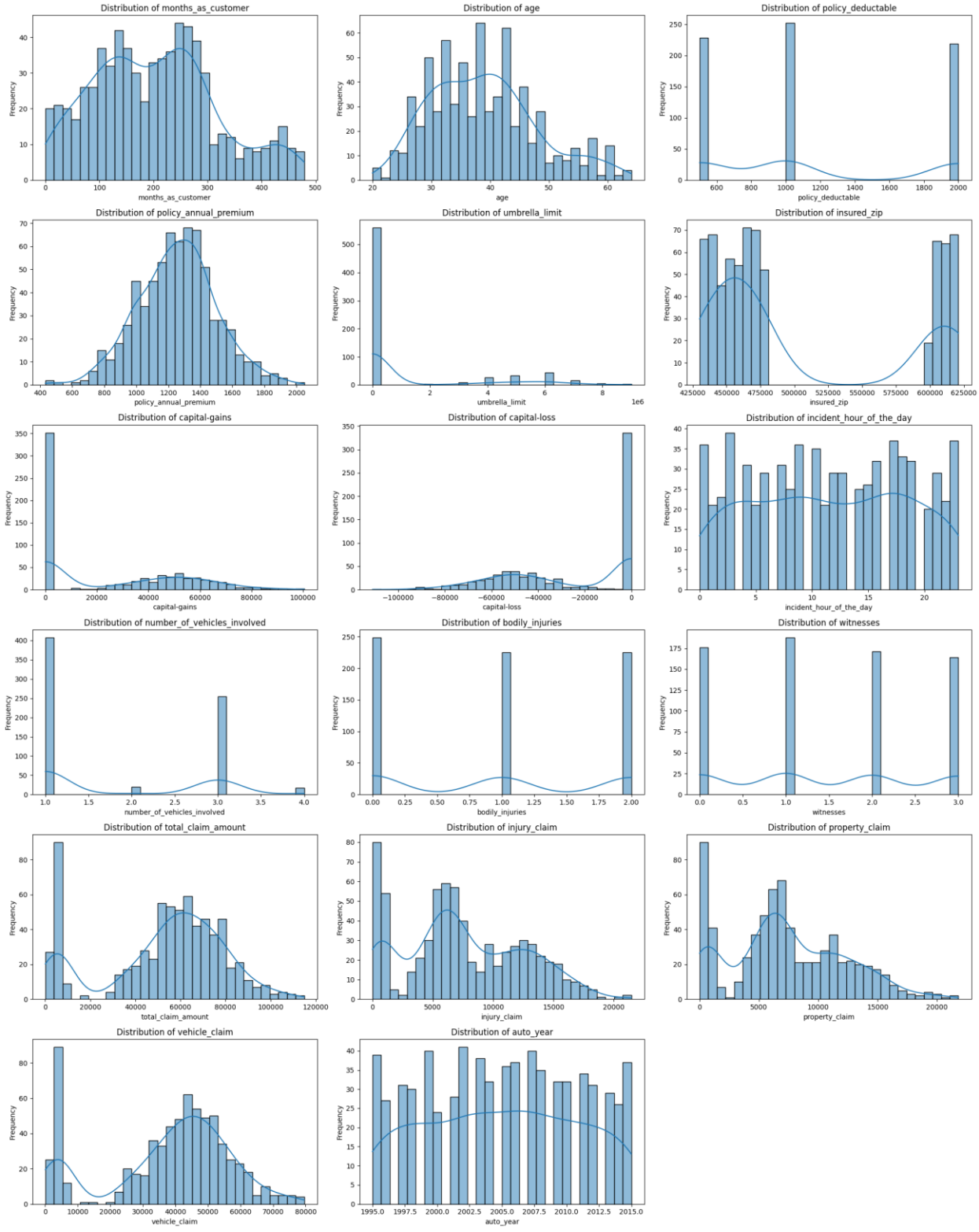
- Understand the underlying distributions and characteristics of individual features (univariate analysis).
- Identify relationships and correlations between different features (bivariate/multivariate analysis).
- Uncover initial patterns or anomalies that might be indicative of fraudulent claims.
- Assess the balance of the target variable (fraud_reported).
- Inform subsequent feature engineering and model selection decisions.

All EDA in this section was performed on the X_train and y_train datasets. A combined DataFrame train_df was used for convenience.

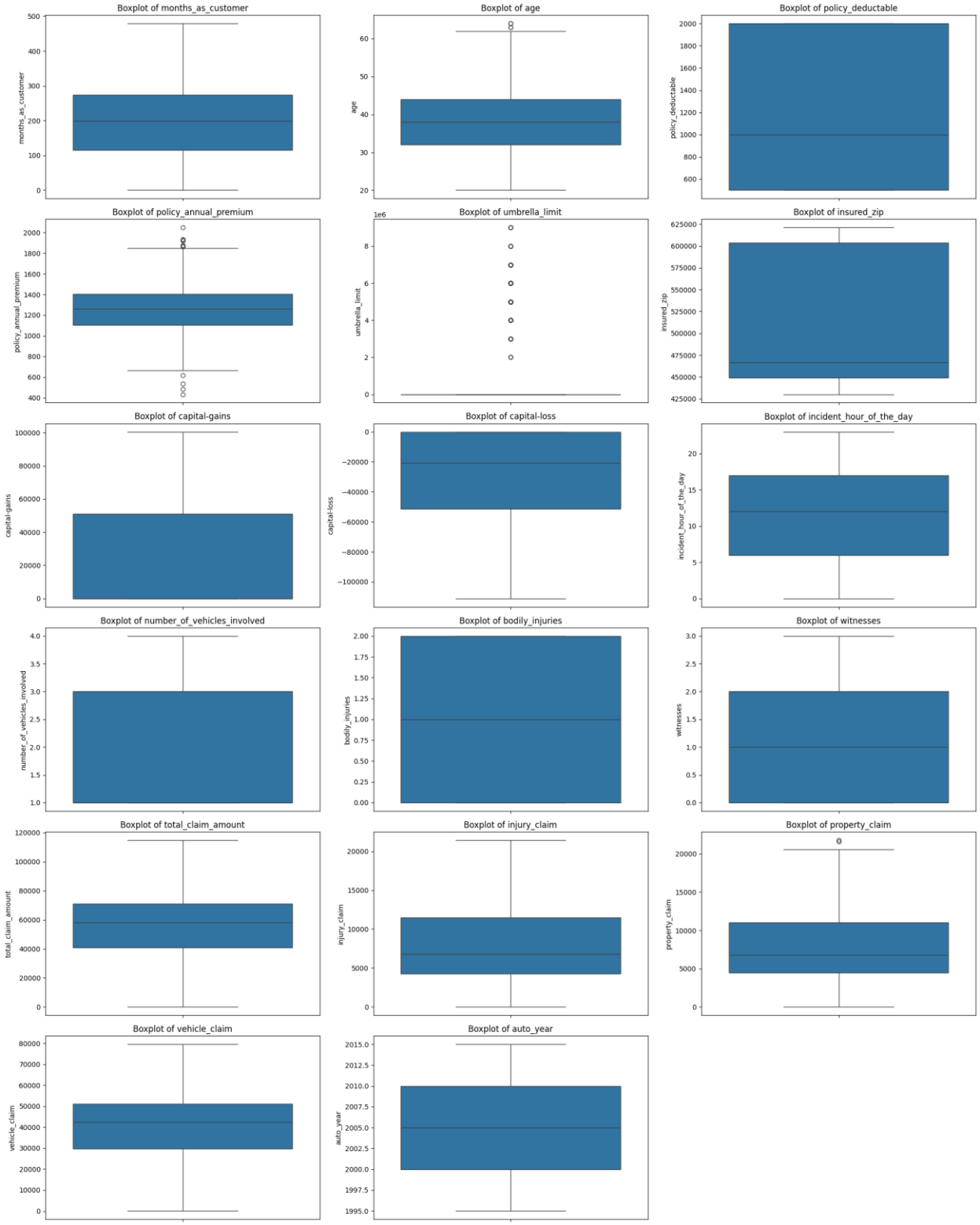
Univariate Analysis of Numerical Features

The 17 numerical features present in the training set (months_as_customer, age, policy_deductable, policy_annual_premium, umbrella_limit, insured_zip, 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) were analyzed.

Distribution Plots (Histograms)



Outlier Detection (Boxplots)



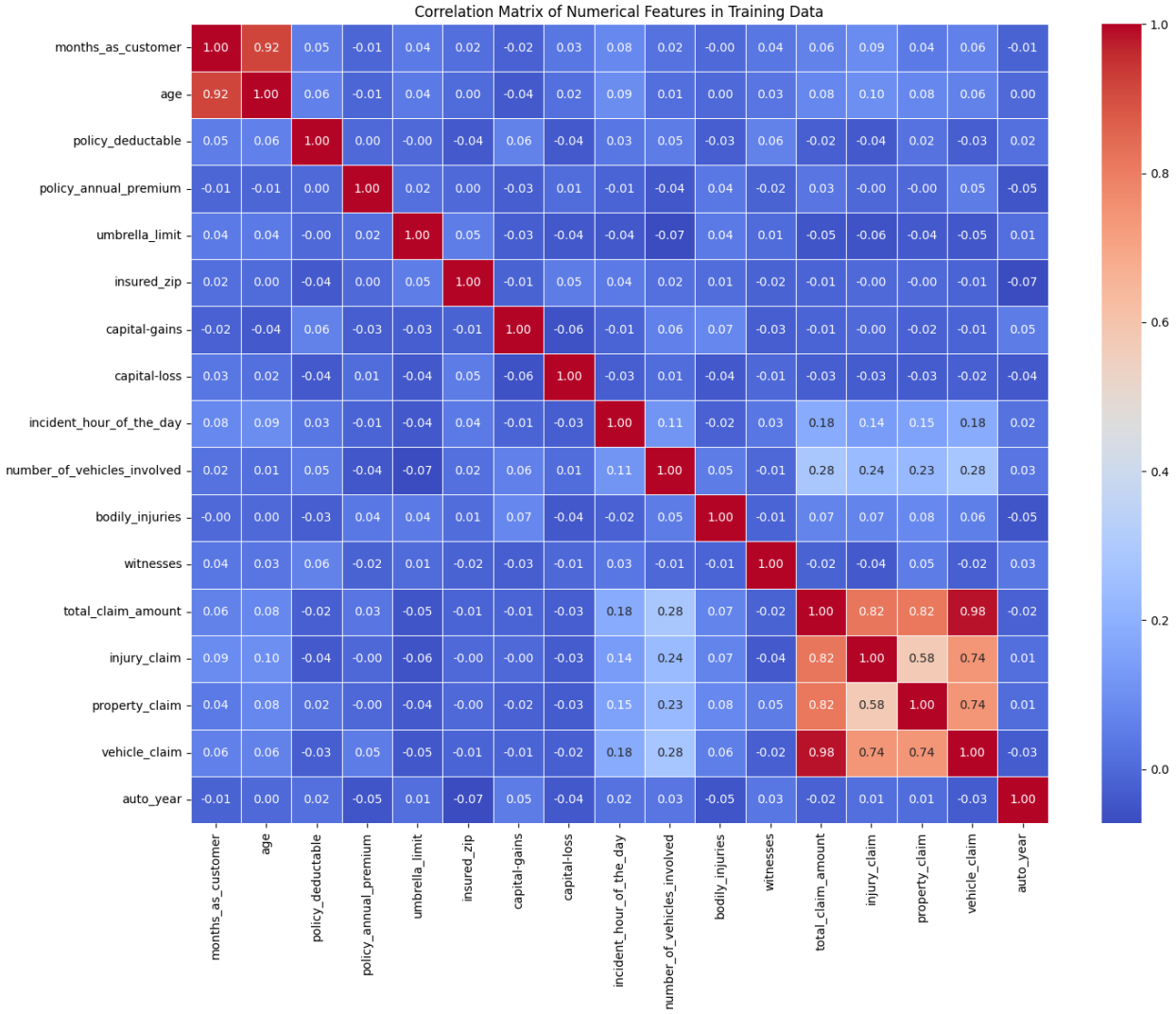
Summary of Observations for Key Numerical Features:

- months_as_customer: Showed a relatively spread-out, somewhat multimodal distribution, indicating varied customer tenures. No significant outliers were apparent.
- age: Exhibited a roughly bell-shaped distribution, slightly right-skewed, with the majority of customers aged between 30 and 50. A few potential outliers were noted on the higher age side.
- policy_deductable: Displayed three distinct peaks corresponding to the categorical values of 500, 1000, and 2000, as expected.
- policy_annual_premium: Appeared somewhat normally distributed, with a slight right skew. Most premiums ranged between approximately $750 and $1750. Some outliers were present at both low (around $400) and high (above $2000) ends.
- umbrella_limit: The distribution was heavily dominated by the value 0, indicating most policies lack an umbrella limit. Non-zero values were sparse and appeared as outliers in the boxplot due to this zero-dominance.
- insured_zip: Displayed a multimodal distribution, suggesting geographical clustering. Its numerical representation was less meaningful for direct interpretation as a continuous variable.
- capital-gains & capital-loss: Both were highly skewed, with a vast majority of values at 0. Non-zero capital-gains were positive, and non-zero capital-loss were negative. These non-zero values appeared as outliers.
- incident_hour_of_the_day: Showed a relatively uniform distribution across the 24 hours, with minor variations.
- number_of_vehicles_involved: Primarily peaked at 1 and 3 vehicles.
- bodily_injuries & witnesses: These discrete numerical features showed reasonable distributions across their limited range of values (0-2 for injuries, 0-3 for witnesses).
- total_claim_amount, injury_claim, property_claim, vehicle_claim: All claim-related amounts exhibited right-skewed distributions, with most claims being of lower to moderate value but with a tail of higher-value claims. Potential outliers were noted on the higher side for these features. vehicle_claim often constituted a large portion of the total_claim_amount.
- auto_year: Showed a varied distribution of manufacturing years, with a higher frequency of newer cars.

Correlation Analysis of Numerical Features

Correlation Matrix and Heatmap

To understand linear relationships between numerical features and identify potential multicollinearity, a Pearson correlation matrix was computed and visualized as a heatmap.



Heatmap of the Pearson correlation coefficients between numerical features in the training data. Red indicates positive correlation, blue indicates negative correlation, and intensity indicates strength.

Identification of Multicollinearity and Key Correlations:

Very Strong Positive Correlation:

- age and months_as_customer (0.92): Highly correlated, as expected. Longer-term customers are generally older. This points to significant multicollinearity.
- total_claim_amount and vehicle_claim (0.98): Extremely high, indicating vehicle_claim is a primary driver of total_claim_amount.

Strong Positive Correlations:

- total_claim_amount with injury_claim (0.82) and property_claim (0.82).
- vehicle_claim with injury_claim (0.74) and property_claim (0.74).

Moderate Positive Correlation:

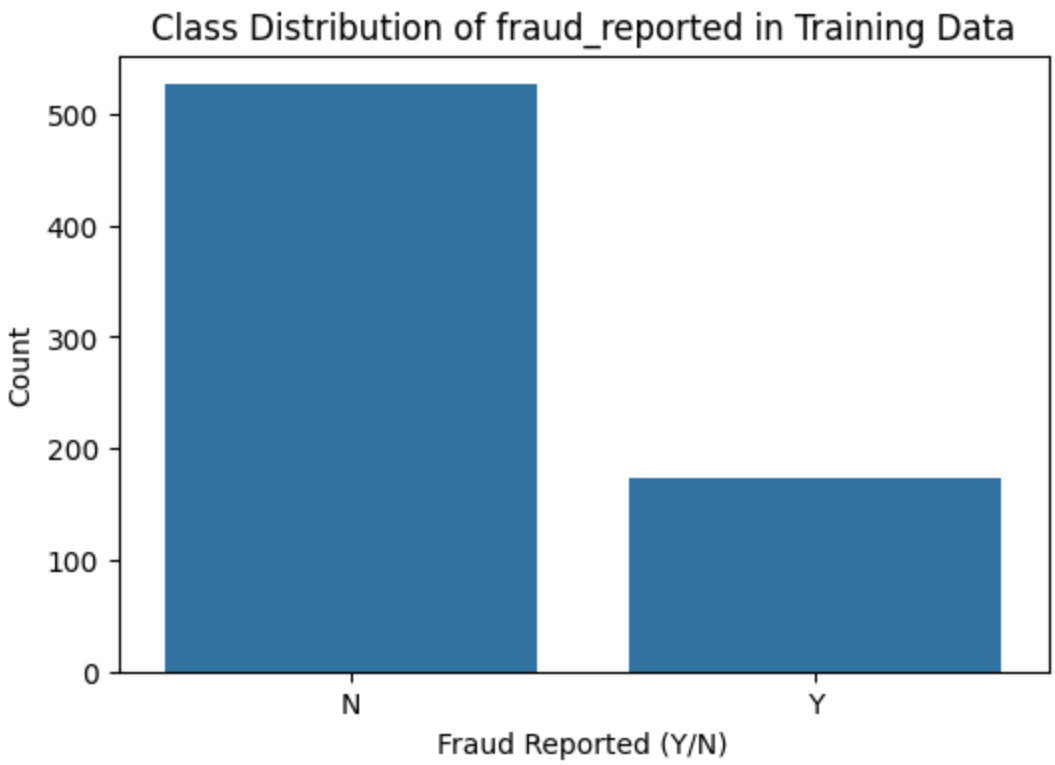
- injury_claim and property_claim (0.58).
- Weak Correlations: Most other pairs showed weak linear relationships.

Implication: The high correlations between total_claim_amount and its components, and between age and months_as_customer, suggested that some of these features might be redundant for certain modeling techniques and would need to be addressed in feature engineering.

Class Balance Assessment of Target Variable (fraud_reported)

Distribution Plot

The distribution of the target variable fraud_reported was examined to check for class imbalance.



Distribution of the target variable 'fraud_reported' in the training set, showing counts for 'N' (Not Fraudulent) and 'Y' (Fraudulent) claims.

Implication of Imbalance:

- Counts in Training Data (y_train): 'N' (Not Fraudulent) = 526, 'Y' (Fraudulent) = 173.
- Percentage Distribution: Approximately 75.25% 'N' and 24.75% 'Y'.

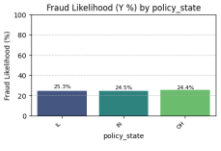
Conclusion: The training dataset is imbalanced, with a significantly larger proportion of non-fraudulent claims. This imbalance can bias models towards the majority class and requires mitigation strategies like resampling during feature engineering.

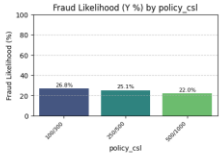
Bivariate Analysis

To understand the relationship between individual features and the target variable (fraud_reported).

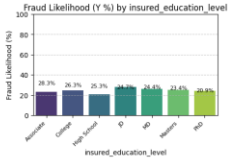
Categorical Features vs. Target Variable (Fraud Likelihood Analysis)

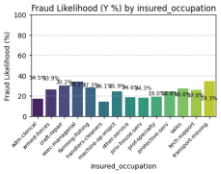
The likelihood of a claim being fraudulent ('Y') was calculated for each category within the 17 categorical features.

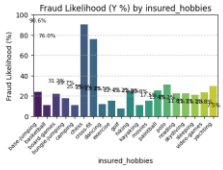

Fraud Likelihood (Y %) by policy_state
Fraud Likelihood (%)
25.3%
24.5%
24.4%
IL
IN
OH
policy_state

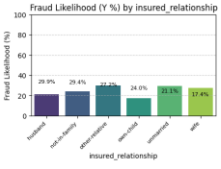

Fraud Likelihood (Y %) by policy_csl
Fraud Likelihood (%)
26.8%
25.1%
22.0%
100/300
250/500
500/1000
policy_csl

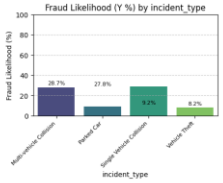

Fraud Likelihood (Y %) by insured_sex
Fraud Likelihood (%)
24.9%
24.5%
FEMALE
MALE
insured_sex

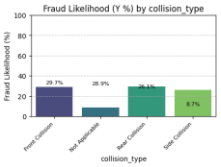

Fraud Likelihood (Y %) by insured_education_level
Fraud Likelihood (%)
28.3%
26.3%
25.3%
24.4%
24.4%
23.4%
20.9%
Associate
College
High School
JD
MD
Masters
PhD
insured_education_level

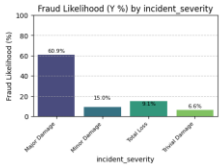

Fraud Likelihood (Y %) by insured_occupation
Fraud Likelihood (%)
machine-op-inspct
armed-forces
sales
tech-support
transport-moving
insured_occupation

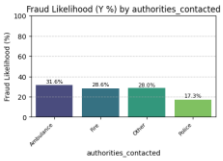

Fraud Likelihood (Y %) by insured_hobbies
Fraud Likelihood (%)
96.6%
76.0%
chess
cross-fit
base-jumping
skydiving
insured_hobbies

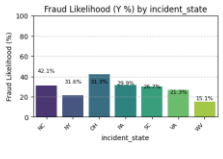

Fraud Likelihood (Y %) by insured_relationship
Fraud Likelihood (%)
29.9%
29.4%
27.2%
24.0%
21.3%
17.4%
husband
not-in-family
other-relative
own-child
unmarried
wife
insured_relationship


Fraud Likelihood (Y %) by incident_type
Fraud Likelihood (%)
28.7%
27.8%
9.2%
8.2%
Multi-vehicle Collision
Parked Car
Single Vehicle Collision
Vehicle Theft
incident_type

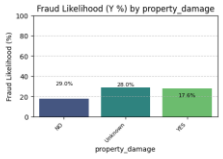

Fraud Likelihood (Y %) by collision_type
Fraud Likelihood (%)
29.7%
28.9%
8.7%
Front Collision
Not Applicable
Rear Collision
Side Collision
collision_type

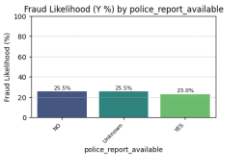

Fraud Likelihood (Y %) by incident_severity
Fraud Likelihood (%)
60.9%
15.0%
6.6%
Major Damage
Minor Damage
Total Loss
Trivial Damage
incident_severity


Fraud Likelihood (Y %) by authorities_contacted
Fraud Likelihood (%)
31.6%
28.6%
26.0%
17.3%
Ambulance
Fire
Other
Police
authorities_contacted

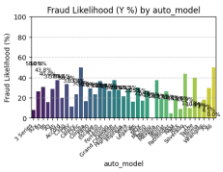

Fraud Likelihood (Y %) by incident_state
Fraud Likelihood (%)
42.1%
31.6%
15.1%
NC
NY
OH
PA
SC
VA
WV
incident_state


Fraud Likelihood (Y %) by incident_city
Fraud Likelihood (%)
29.6%
27.6%
27.3%
25.0%
21.3%
21.2%
20.0%
Arlington
Columbus
Hillsdale
Northland
Northbrook
Riverwood
Springfield
incident_city


Fraud Likelihood (Y %) by property_damage
Fraud Likelihood (%)
29.0%
28.0%
17.6%
NO
YES
property_damage


Fraud Likelihood (Y %) by police_report_available
Fraud Likelihood (%)
25.5%
25.5%
23.0%
NO
YES
police_report_available


Fraud Likelihood (Y %) by auto_make
Fraud Likelihood (%)
Acura
Audi
BMW
Chevrolet
Dodge
Ford
Honda
Jeep
Nissan
Saab
Suburu
Toyota
Volkswagen
auto_make


Fraud Likelihood (Y %) by auto_model
Fraud Likelihood (%)
auto_model

Selected Bar Charts of Fraud Likelihood by Categorical Feature - Training Data (e.g., for incident_severity, insured_hobbies, insured_occupation)]

Key Findings:

Strong Differentiators:

- incident_severity: "Major Damage" had a ~60.9% fraud likelihood, whereas "Trivial Damage" and "Minor Damage" had very low likelihoods (~6.6% and ~9.1% respectively).
- insured_hobbies: Showed extreme variations. "Chess" (90.6% fraud likelihood) and "Cross-fit" (76.0%) were strongly associated with fraud, while hobbies like "Golf" (7.5%) and "Camping" (10.8%) had very low fraud likelihood.
- insured_occupation: Occupations like "transport-moving" (34.5%) and "exec-managerial" (33.9%) had higher fraud rates than "handlers-cleaners" (14.3%) or "adm-clerical" (17.0%).

Moderate Differentiators:

- Features like insured_relationship, incident_type, collision_type, insured_education_level, property_damage, authorities_contacted, incident_state, incident_city, and auto_make showed some variation in fraud likelihood across their categories.

Weak Differentiators:

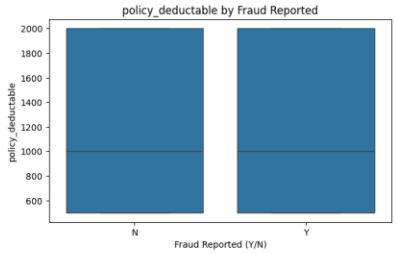
- policy_state, insured_sex, and police_report_available showed very little variation in fraud likelihood across their categories, with rates close to the overall average of ~24.75%. These appeared less predictive on their own.

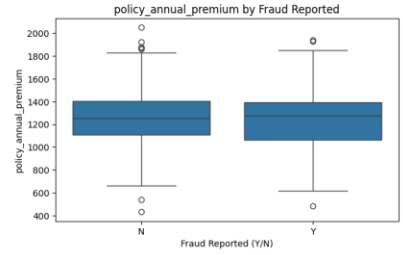
Numerical Features vs. Target Variable (Distribution Comparison)

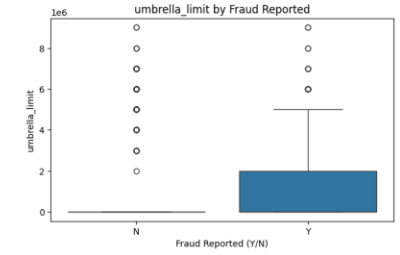
Box plots and violin plots were used to compare the distributions of numerical features for fraudulent ('Y') vs. non-fraudulent ('N') claims.

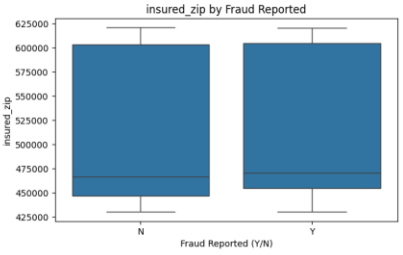

months_as_customer by Fraud Reported
months_as_customer
Fraud Reported (Y/N)
N
Y

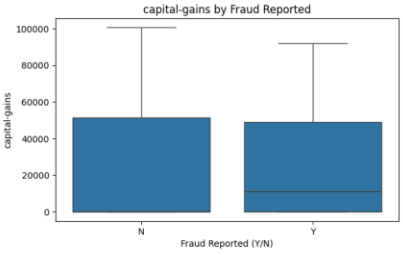

age by Fraud Reported
age
Fraud Reported (Y/N)
N
Y

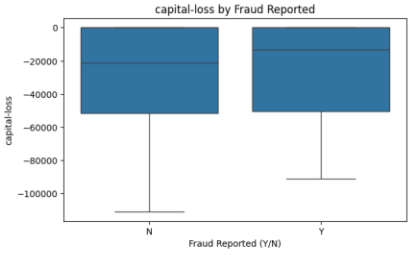

policy_deductable by Fraud Reported
policy_deductable
Fraud Reported (Y/N)
N
Y

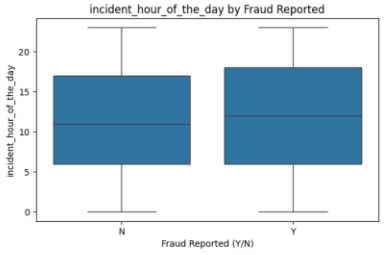

policy_annual_premium by Fraud Reported
policy_annual_premium
Fraud Reported (Y/N)
N
Y

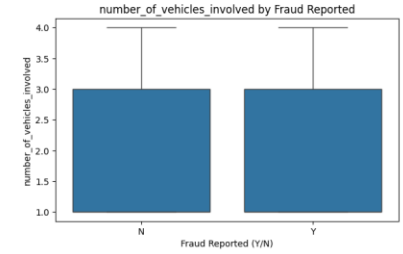

umbrella_limit by Fraud Reported
1e6
umbrella_limit
Fraud Reported (Y/N)
N
Y

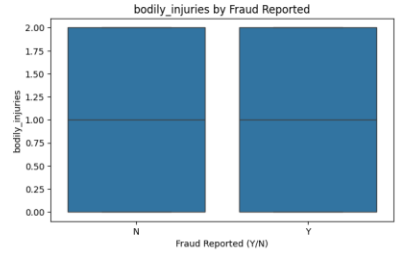

insured_zip by Fraud Reported
insured_zip
Fraud Reported (Y/N)
N
Y

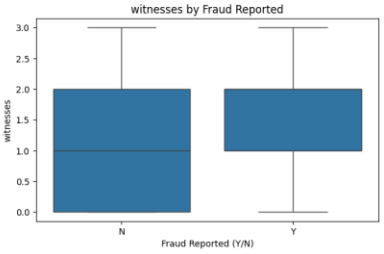

capital-gains by Fraud Reported
capital-gains
Fraud Reported (Y/N)
N
Y

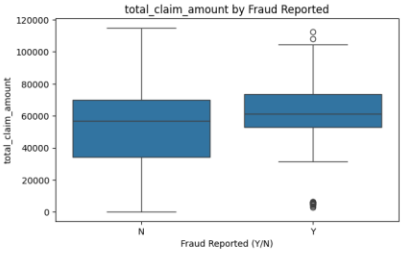

capital-loss by Fraud Reported
capital-loss
Fraud Reported (Y/N)
N
Y

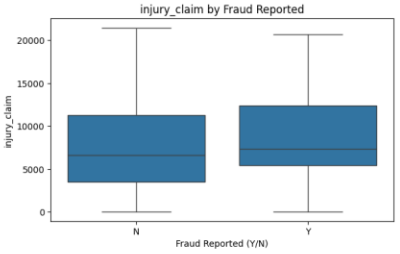

incident_hour_of_the_day by Fraud Reported
incident_hour_of_the_day
Fraud Reported (Y/N)
N
Y

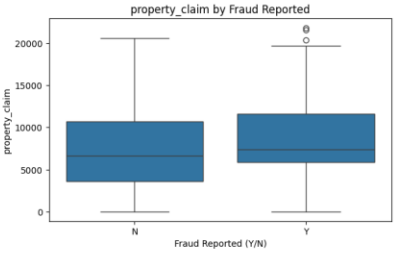

number_of_vehicles_involved by Fraud Reported
number_of_vehicles_involved
Fraud Reported (Y/N)
N
Y

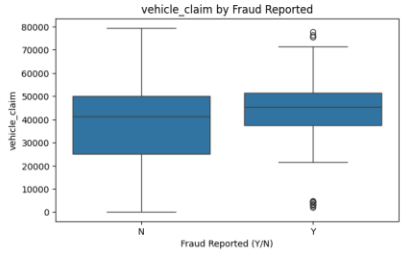

bodily_injuries by Fraud Reported
bodily_injuries
Fraud Reported (Y/N)
N
Y

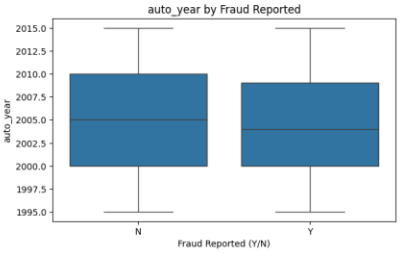

witnesses by Fraud Reported
witnesses
Fraud Reported (Y/N)
N
Y

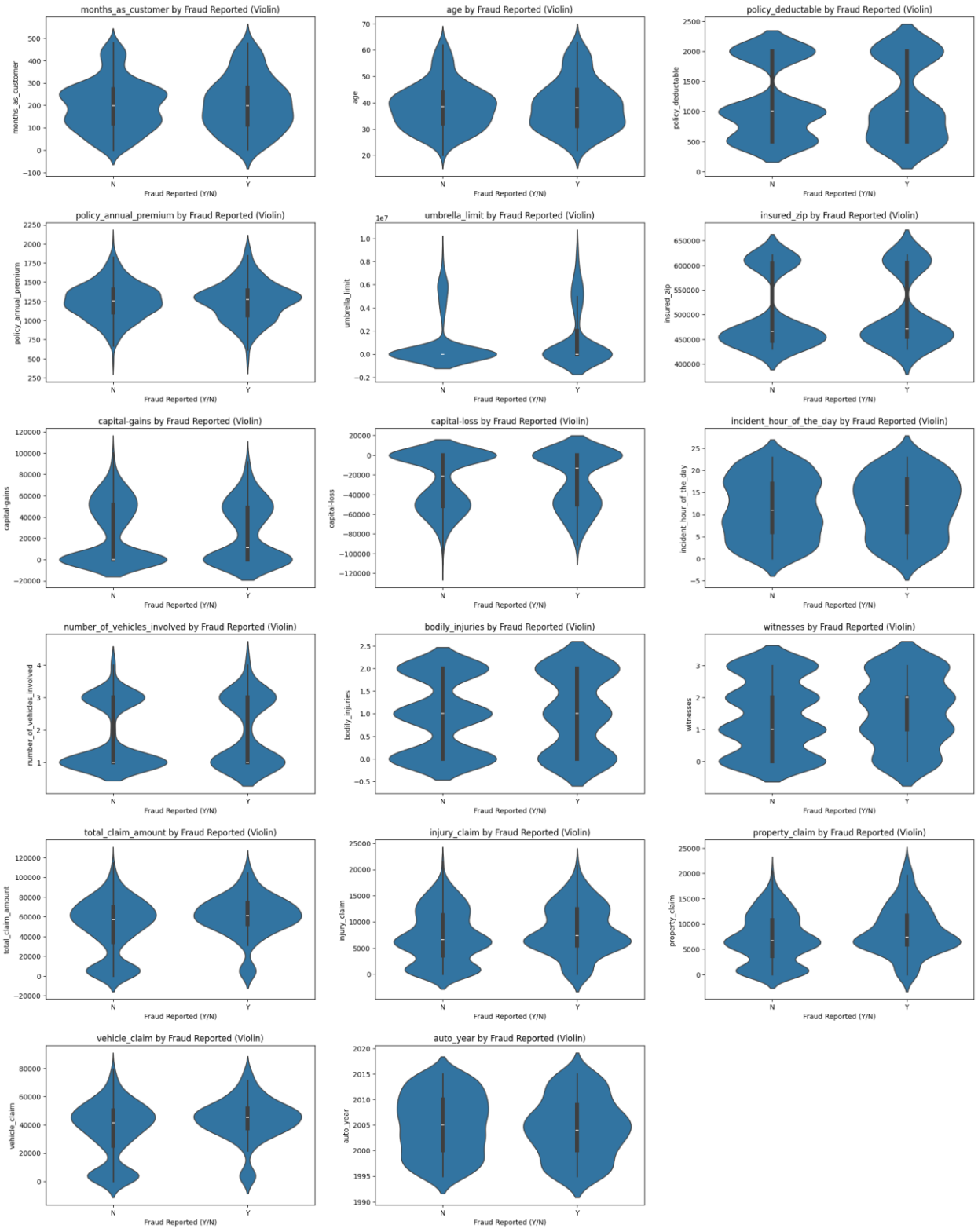

total_claim_amount by Fraud Reported
total_claim_amount
Fraud Reported (Y/N)
N
Y


injury_claim by Fraud Reported
injury_claim
Fraud Reported (Y/N)
N
Y


property_claim by Fraud Reported
property_claim
Fraud Reported (Y/N)
N
Y


vehicle_claim by Fraud Reported
vehicle_claim
Fraud Reported (Y/N)
N
Y


auto_year by Fraud Reported
auto_year
Fraud Reported (Y/N)
N
Y



Selected Boxplots/Violin Plots of Numerical Features by 'fraud_reported' - Training Data (e.g., for total_claim_amount, months_as_customer, witnesses)

Key Findings:

- total_claim_amount (and its components injury_claim, property_claim, vehicle_claim): Fraudulent claims ('Y') surprisingly exhibited slightly lower median values and a more compressed distribution compared to non-fraudulent claims ('N').
- months_as_customer & age: Median values were slightly lower for fraudulent claims, suggesting newer or younger customers might have a higher fraud tendency.
- witnesses: Fraudulent claims tended to have a slightly lower median number of witnesses.
- policy_deductable: Fraudulent claims appeared more associated with lower deductible amounts (500, 1000).

Features like policy_annual_premium, umbrella_limit (mostly 0), incident_hour_of_the_day, and auto_year showed very similar distributions for both fraudulent and non-fraudulent claims, indicating less individual predictive power.

Summary of Key EDA Insights for Training Data

- The target variable fraud_reported is imbalanced.
- Strong indicators of potential fraud include incident_severity ('Major Damage'), specific insured_hobbies ('chess', 'cross-fit'), and potentially certain insured_occupations.
- Counter-intuitively, fraudulent claims in this dataset tended to have slightly lower total claim amounts.
- Fewer witnesses and lower policy deductibles were also associated with a higher likelihood of fraud.
- High multicollinearity was identified between age and months_as_customer, and between total_claim_amount and its constituent claim parts, necessitating careful handling in feature engineering.
- Several features (e.g., policy_state, insured_sex) showed weak individual relationships with the target variable.

5. EXPLORATORY DATA ANALYSIS – VALIDATION DATA

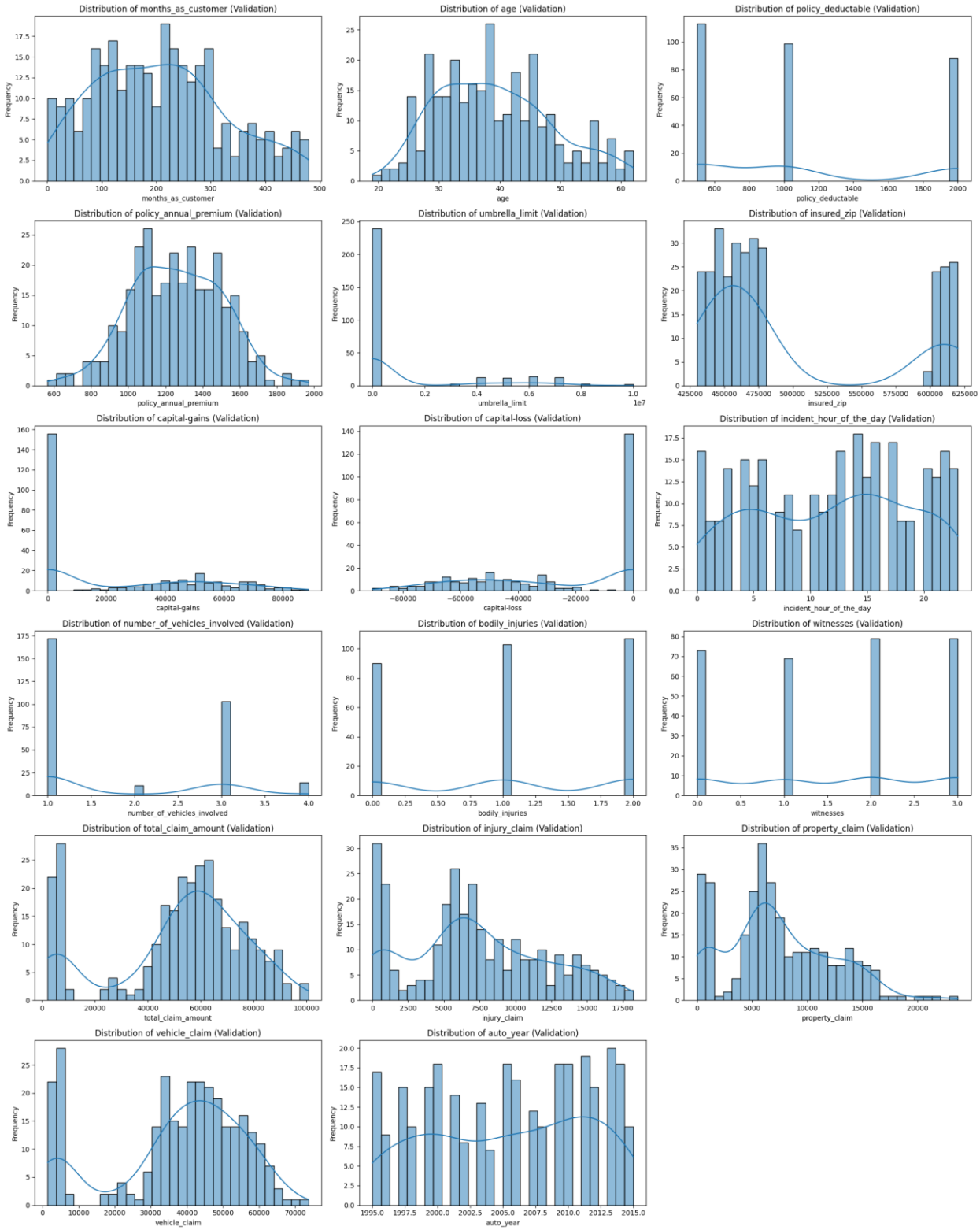
Rationale for Validation Set EDA

- Performing EDA on the validation set is a crucial step to ensure that the patterns, distributions, and relationships observed in the training data are consistent and hold true for unseen data. This helps to:
- Detect potential data drift or discrepancies between the training and validation sets.
- Confirm the generalizability of insights derived from the training data.
- Validate assumptions made during feature engineering and model selection.
- If significant differences are found, it might indicate issues with the data split or that the training data insights are not robust.

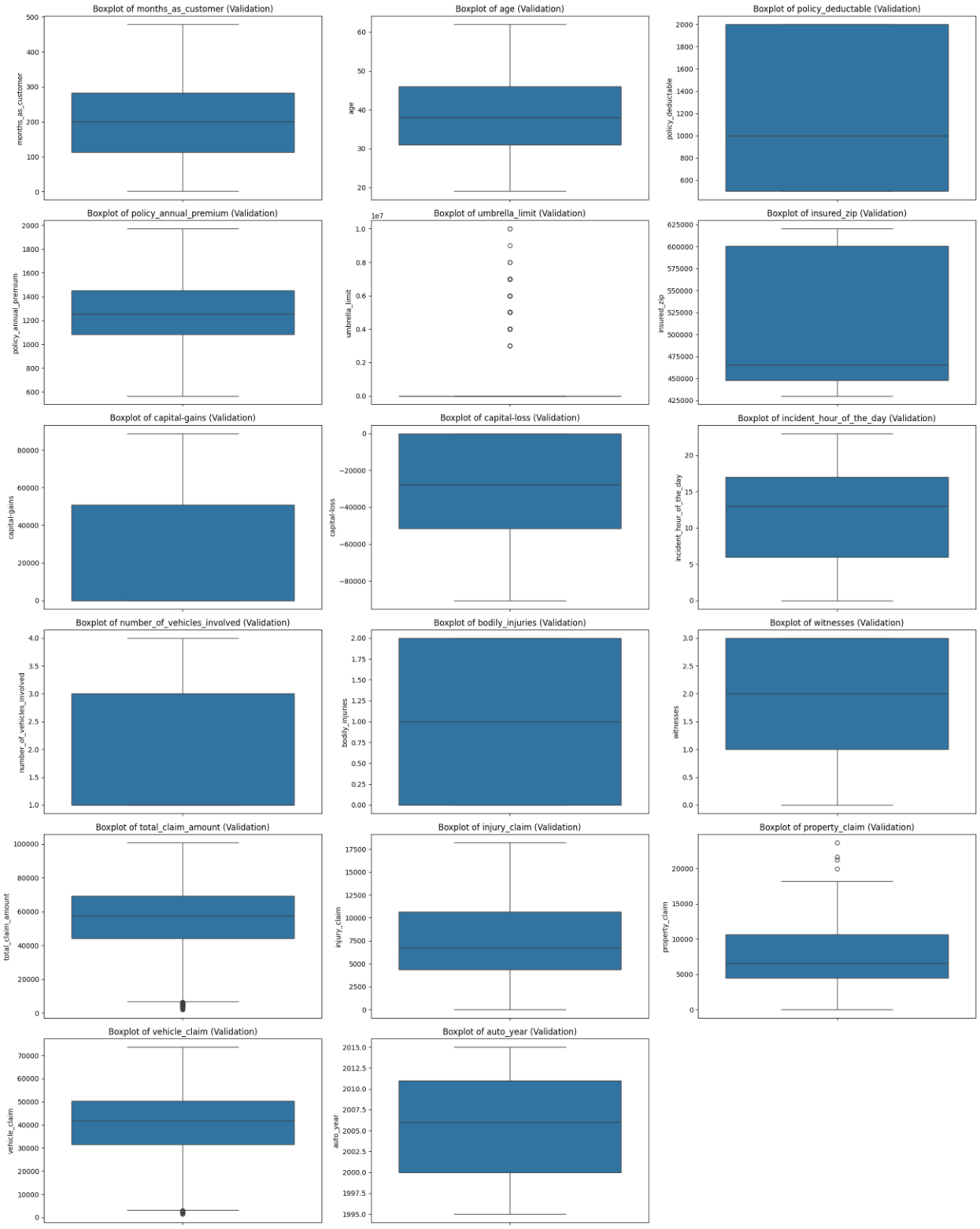
Comparative Univariate, Correlation, and Bivariate Analyses

- The same EDA steps performed on the training data were replicated for the validation set (X_val, y_val).

Univariate Analysis (Numerical Features - Validation)



Histograms of Numerical Features - Validation Data



Boxplots of Numerical Features - Validation Data

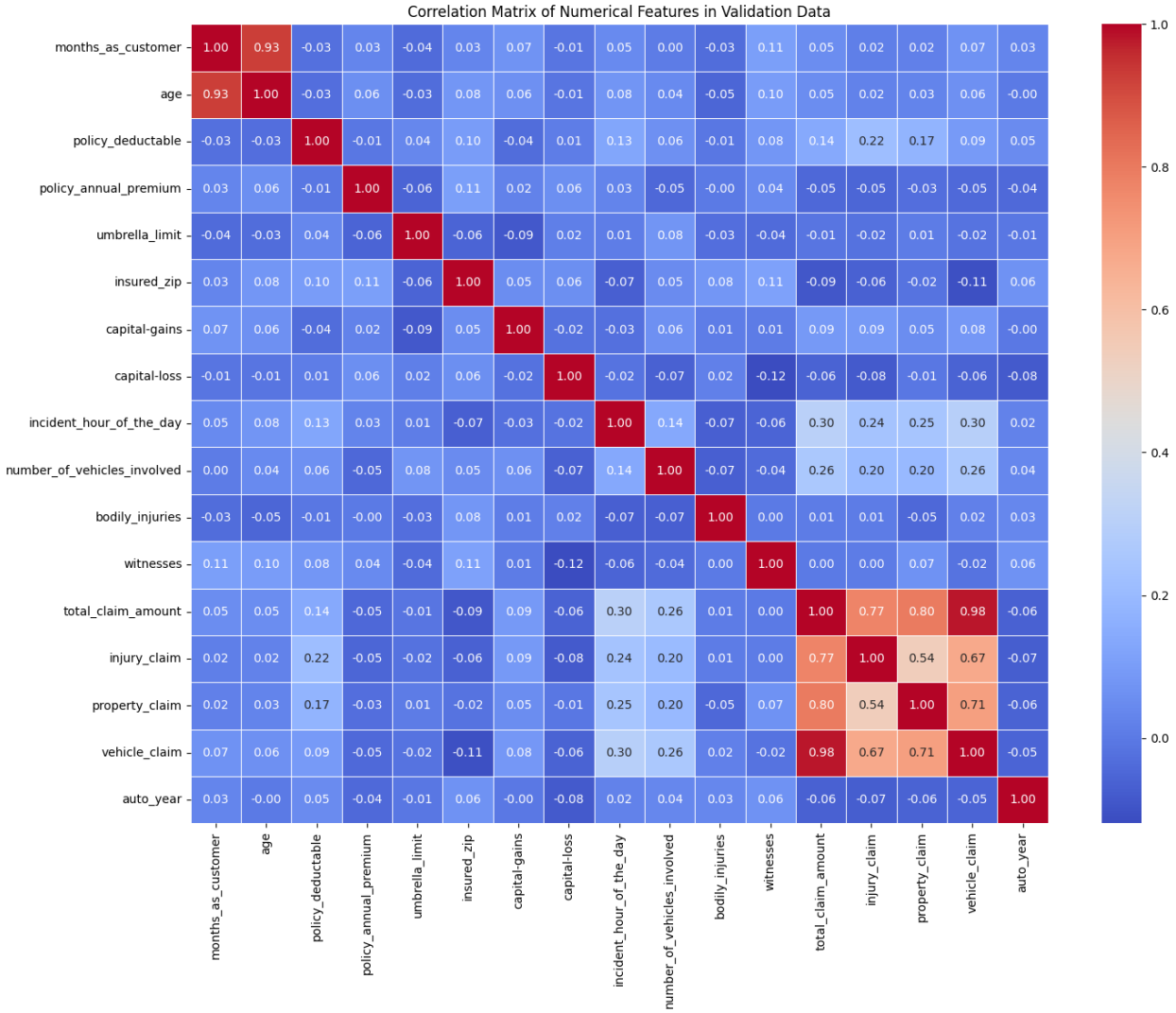
Observations: The distributions (shape, central tendency, spread, outliers) of key numerical features like age, policy_annual_premium, total_claim_amount, months_as_customer, capital-gains/loss in the validation set

were found to be generally consistent with those observed in the training set. No drastic shifts in these underlying distributions were noted.

Data Distribution Comparison (Validation vs. Training)

- **Age**: Similar range and shape (bellshaped, slightly rightskewed).
- **Policy Annual Premium**: Comparable distribution and outliers; slightly rightskewed.
- **Total Claim Amount**: Similar rightskewed distribution and range.
- **Months as Customer**: Consistent multimodal pattern.
- **Capital Gains / Loss**: High concentration at zero; similar skew for nonzero values.
- **Insured Zip**: Multimodal pattern matches training set.

Correlation Analysis (Numerical Features - Validation)



Observations from the Correlation Matrix of Numerical Features (Validation Data)

The heatmap below presents the **Pearson correlation coefficients** between all numerical features in the validation dataset. Key relationships and trends are as follows:

Strong Positive Correlations (r > 0.70)

Feature Pair	Correlation	Interpretation
total_claim_amount & vehicle_claim	**0.98**	Vehicle claims make up the largest portion of total claims.
total_claim_amount & injury_claim	**0.77**	Injury claims significantly contribute to total claim value.
vehicle_claim & injury_claim	**0.67**	Higher injury claims often occur alongside high vehicle claims.
total_claim_amount & property_claim	**0.80**	Property claims are also strongly linked with total claims.
injury_claim & property_claim	**0.54**	Suggests moderate co-occurrence between injury and property claims.
property_claim & vehicle_claim	**0.71**	Property and vehicle damages often appear together.

⚠️ **Multicollinearity Alert**

Feature Pair	Correlation	Note
months_as_customer & age	**0.93**	Highly collinear; older customers tend to have longer policy tenures. One should be dropped or transformed to avoid multicollinearity in linear models.

🟠 **Moderate Correlations (0.30 ≤ r < 0.70)**

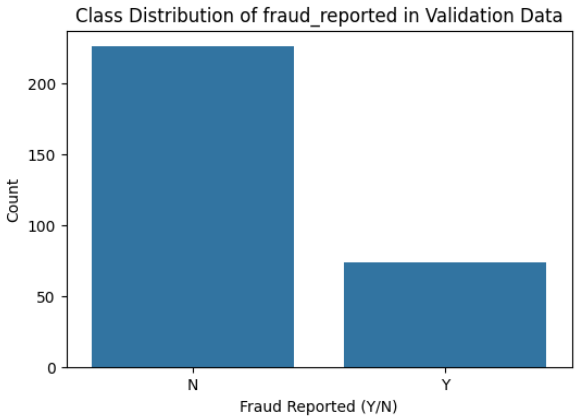
Feature Pair	Correlation	Interpretation
witnesses & injury_claim	**0.54**	More witnesses may be associated with injury-related claims.
witnesses & total_claim_amount	**0.30**	A slight tendency for higher witness count with higher claims.
incident_hour_of_the_day & injury_claim	**0.24**	Some weak patterns by time of day for injury claims.

🔻 **Weak or Negligible Correlations (|r| < 0.30)**

Most other pairs exhibit low to negligible correlation, suggesting **independent or weakly related variables**, including:

- umbrella_limit with most features (r < ±0.10)
- insured_zip with claim amounts (low predictive power as a continuous variable)
- capital-gains and capital-loss with all other features (mostly sparse or zero)

Class Balance Assessment (Target Variable - Validation)



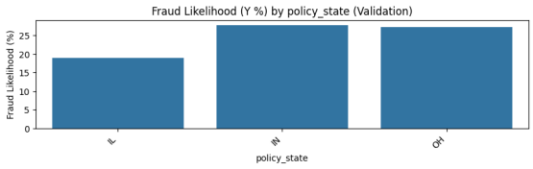
Observations:

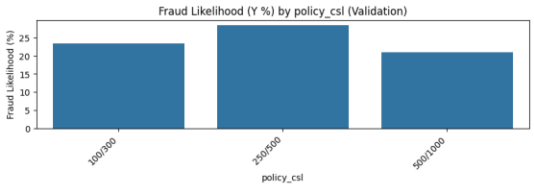
- Counts in Validation Data (y_val): 'N' (Not Fraudulent) = 226, 'Y' (Fraudulent) = 74.
- Percentage Distribution: Approximately 75.33% 'N' and 24.67% 'Y'.

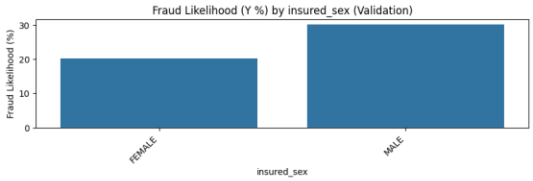
This is highly consistent with the training set's imbalance (Training: ~75.25% 'N', ~24.75% 'Y'), as expected due to stratified splitting.

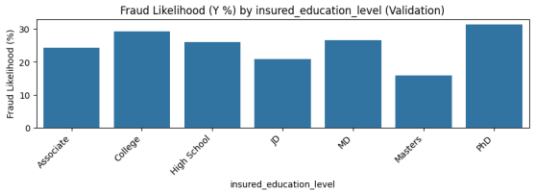
Bivariate Analysis (Validation)

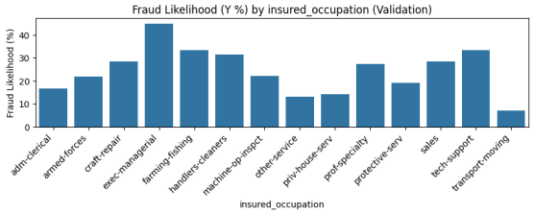
Categorical Features vs. Target:

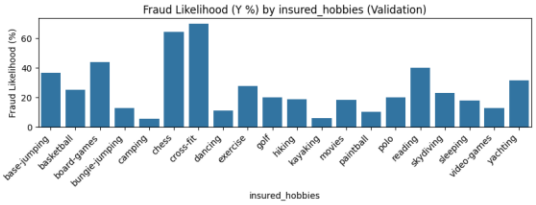


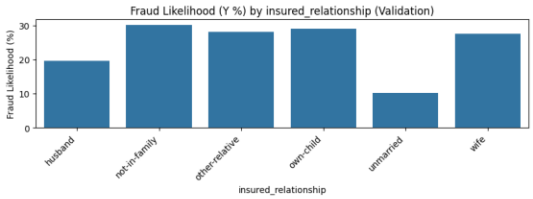








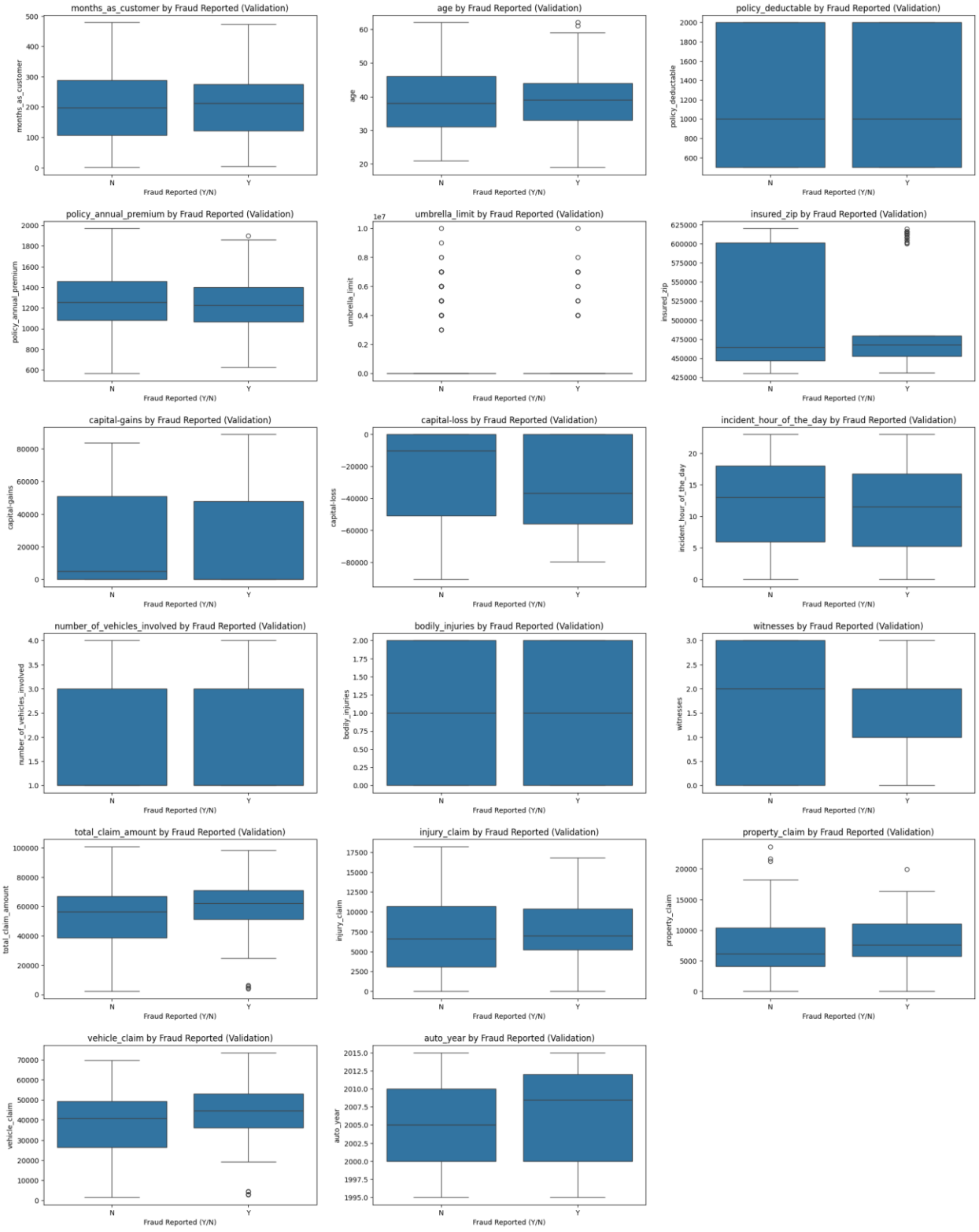


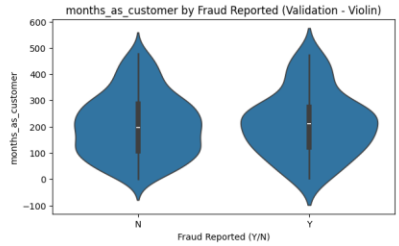


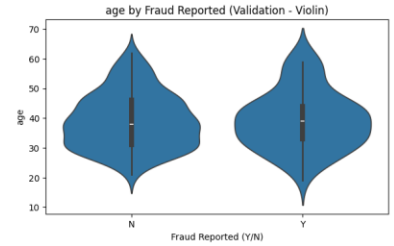
Observations

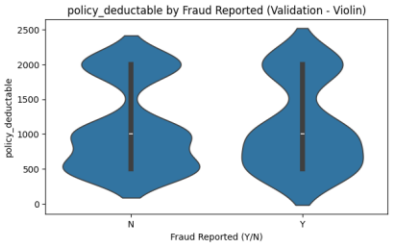
- The strong differentiating patterns for features like incident_severity (Major Damage still high fraud likelihood at ~60.2%) and insured_hobbies (e.g., 'cross-fit' at 70% fraud, 'chess' at ~64% fraud; 'camping' and 'kayaking' very low) generally held in the validation set, though with expected minor fluctuations in exact percentages due to smaller sample sizes per category.
- Some features like insured_occupation showed more variability for specific categories (e.g., transport-moving fraud likelihood dropped significantly compared to training).
- Weak differentiators like police_report_available remained weak. insured_sex showed a more pronounced difference in validation (Males higher fraud likelihood) than in training, which could be due to sampling variance.

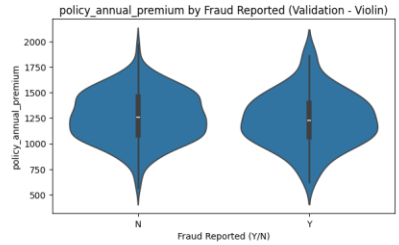
Numerical Features vs. Target:

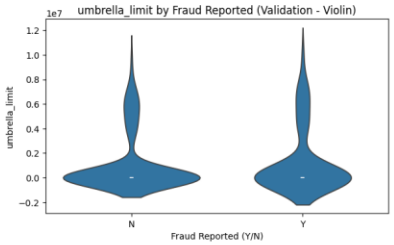


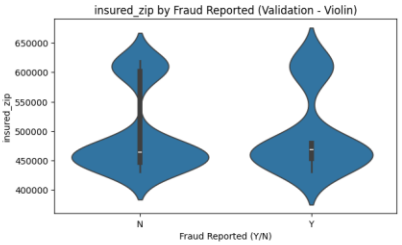

months_as_customer by Fraud Reported (Validation - Violin)
months_as_customer
Fraud Reported (Y/N)
N
Y

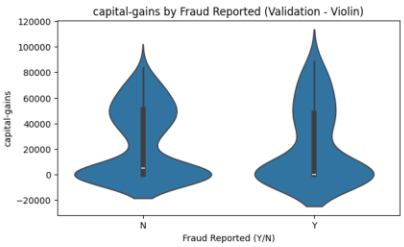

age by Fraud Reported (Validation - Violin)
age
Fraud Reported (Y/N)
N
Y

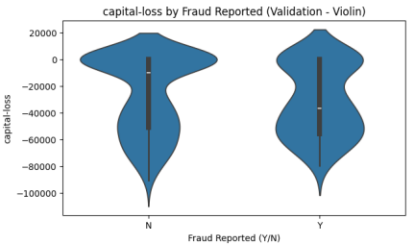

policy_deductable by Fraud Reported (Validation - Violin)
policy_deductable
Fraud Reported (Y/N)
N
Y

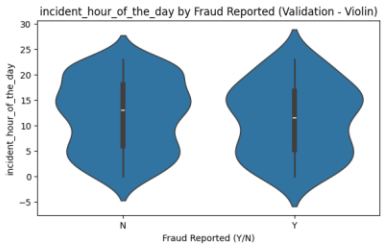

policy_annual_premium by Fraud Reported (Validation - Violin)
policy_annual_premium
Fraud Reported (Y/N)
N
Y

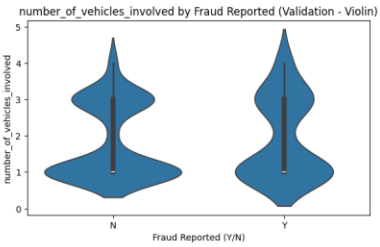

umbrella_limit by Fraud Reported (Validation - Violin)
1e7
umbrella_limit
Fraud Reported (Y/N)
N
Y

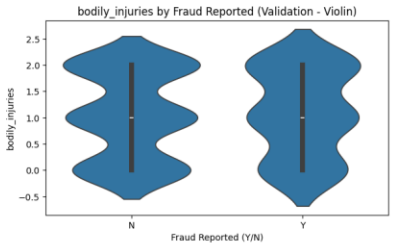

insured_zip by Fraud Reported (Validation - Violin)
insured_zip
Fraud Reported (Y/N)
N
Y

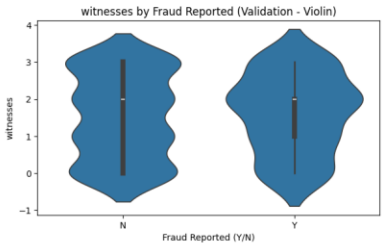

capital-gains by Fraud Reported (Validation - Violin)
capital-gains
Fraud Reported (Y/N)
N
Y

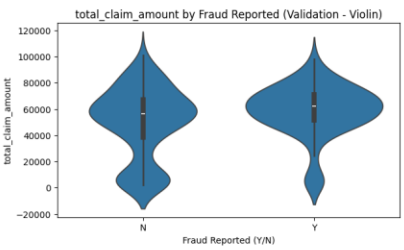

capital-loss by Fraud Reported (Validation - Violin)
capital-loss
Fraud Reported (Y/N)
N
Y

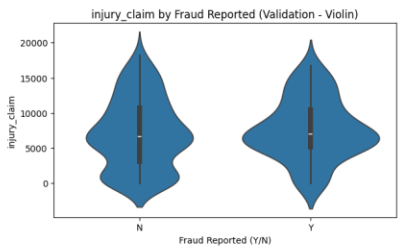

incident_hour_of_the_day by Fraud Reported (Validation - Violin)
incident_hour_of_the_day
Fraud Reported (Y/N)
N
Y

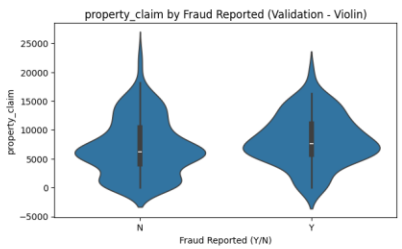

number_of_vehicles_involved by Fraud Reported (Validation - Violin)
number_of_vehicles_involved
Fraud Reported (Y/N)
N
Y

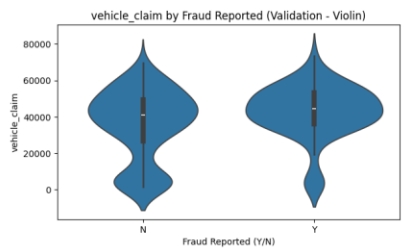

bodily_injuries by Fraud Reported (Validation - Violin)
bodily_injuries
Fraud Reported (Y/N)
N
Y

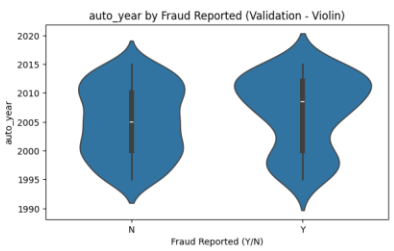

witnesses by Fraud Reported (Validation - Violin)
witnesses
Fraud Reported (Y/N)
N
Y


total_claim_amount by Fraud Reported (Validation - Violin)
total_claim_amount
Fraud Reported (Y/N)
N
Y


injury_claim by Fraud Reported (Validation - Violin)
injury_claim
Fraud Reported (Y/N)
N
Y


property_claim by Fraud Reported (Validation - Violin)
property_claim
Fraud Reported (Y/N)
N
Y


vehicle_claim by Fraud Reported (Validation - Violin)
vehicle_claim
Fraud Reported (Y/N)
N
Y


auto_year by Fraud Reported (Validation - Violin)
auto_year
Fraud Reported (Y/N)
N
Y

Observations:

Trends observed in training generally persisted. For example, median total_claim_amount, months_as_customer, and age appeared slightly lower for fraudulent ('Y') claims in the validation set, similar to training. The pattern for witnesses (fewer for fraud) also seemed consistent.

Bivariate Numerical Analysis (Validation vs. Training)

- **`total_claim_amount`**:
  - **Training**: Median for Fraud (Y) slightly lower; more compressed distribution.
  - **Validation**: Same pattern; N (Not Fraud) shows higher upper range.
  - **Observation**: Consistent — fraudulent claims tend to be slightly lower in amount.
- **`months_as_customer`**:
  - **Training & Validation**: Median for Y lower than N; subtle but consistent.
  - **Observation**: Fraudulent claims linked to slightly newer customers.
- **`witnesses`**:
  - **Training & Validation**: Fraudulent cases (Y) show lower median witnesses, mostly 0–1.
  - **Observation**: Pattern holds — fraud often has fewer witnesses.

Assessment of Data Consistency between Training and Validation Sets

- Overall, the EDA on the validation data confirmed that its characteristics are largely consistent with the training data.
- The distributions of individual features, correlations between features, class balance of the target variable, and the general relationships between features and the target variable did not show major, unexpected deviations.
- This consistency provides confidence that models trained on the training data should generalize reasonably well to this validation set, and that insights derived from the training EDA are robust.

6. FEATURE ENGINEERING

Rationale for Feature Engineering

- Feature engineering aims to transform raw data into a format that better represents the underlying problem to the machine learning models, thereby improving their predictive performance.
- This involves creating new relevant features, handling problematic existing features, and ensuring data is in a suitable numerical format for algorithms.

Resampling of Training Data (Handling Class Imbalance)

Technique: RandomOverSampler

- As identified in the EDA (Section 4.4), the training data exhibited a significant class imbalance, with non-fraudulent claims (~75%) outnumbering fraudulent claims (~25%).
- To mitigate potential model bias towards the majority class and improve the model's ability to learn patterns from the minority (fraudulent) class, RandomOverSampler from the imblearn library was applied.
- This technique works by randomly duplicating instances from the minority class until the classes are balanced.
- This was applied only to the training data (X_train, y_train) after all other feature transformations were complete to prepare the final X_train_final_resampled and y_train_final_resampled.

Impact on Class Distribution:

- Before Resampling (y_train): 'N' (Not Fraud) = 526, 'Y' (Fraud) = 173.
- After Resampling (y_train_final_resampled): 'N' (Not Fraud) = 526, 'Y' (Fraud) = 526.

The training data used for model fitting was perfectly balanced with 1052 total instances.

```
Class distribution in y_train before resampling:
fraud_reported
N    526
Y    173
Name: count, dtype: int64

Shape of X_train after resampling: (1052, 144)
Shape of y_train after resampling: (1052,)

Class distribution in y_train after resampling:
fraud_reported
N    526
Y    526
Name: count, dtype: int64
```

Creation of New Features

To potentially capture more nuanced information, the following features were derived from existing date columns:

Date-Derived Features:

- policy_age_at_incident_days: Calculated as (incident_date - policy_bind_date).dt.days. This represents the tenure of the policy in days at the time the incident occurred, which might be more relevant than total customer tenure for specific claim events.
- incident_month: Extracted from incident_date (e.g., 1 for January, 2 for February). This captures potential seasonality in incidents.
- incident_day_of_week: Extracted from incident_date (Monday=0, Sunday=6). This captures potential patterns related to the day of the week.
- incident_year: Was also extracted but later dropped as all incidents occurred in 2015, providing no variance.

```
...   X_train with new features (head):
      policy_bind_date incident_date  policy_age_at_incident_days  incident_month  \
    0       1990-02-23    2015-02-21                         9129               2
    1       2012-11-02    2015-01-30                          819               1
    2       2008-09-03    2015-02-16                         2357               2
    3       1998-12-12    2015-01-28                         5891               1
    4       1998-02-12    2015-02-01                         6198               2

       incident_day_of_week  incident_year
    0                     5           2015
    1                     4           2015
    2                     0           2015
    3                     2           2015
    4                     6           2015

    X_val with new features (head):
      policy_bind_date incident_date  policy_age_at_incident_days  incident_month  \
    0       2009-03-05    2015-02-17                         2175               2
    1       1990-09-22    2015-02-18                         8915               2
    2       2009-04-10    2015-01-08                         2099               1
    3       2002-09-10    2015-02-14                         4540               2
    4       1995-02-21    2015-02-05                         7289               2

       incident_day_of_week  incident_year
    0                     1           2015
    ...
    4                     3           2015

    New shape of X_train: (699, 40)
    New shape of X_val: (300, 40)
```

Rationale and Expected Impact:

These temporal features can help models identify patterns related to policy maturity at the time of incident or time-based trends in claim occurrences, potentially improving predictive accuracy.

Handling Redundant or Low-Value Features (Post-EDA & Feature Creation)

- Based on EDA findings and the creation of new features, several columns were deemed redundant or of low predictive value and subsequently dropped from both training and validation sets:
- Original Date Columns (policy_bind_date, incident_date): Dropped after extracting relevant information into new features like policy_age_at_incident_days, incident_month, and incident_day_of_week.
- Zero-Variance Feature (incident_year): Dropped as all incidents occurred in 2015, offering no predictive information.
- Highly Correlated Feature (age): Dropped due to its very high correlation (0.92) with months_as_customer. months_as_customer and the new policy_age_at_incident_days were retained as they represent different aspects of tenure/time.

Composite Feature (total_claim_amount): EDA confirmed that total_claim_amount was an exact sum of injury_claim, property_claim, and vehicle_claim. To avoid perfect multicollinearity, total_claim_amount was dropped, and its more granular components were retained as predictors.

After these steps, the number of features was reduced from 40 (after new date features) to 35.

```
Models in X_train to be grouped into 'Other_Model' (count < 15):
['Impreza', '3 Series', '93', 'C300', 'Corolla', 'Civic', 'RSX', 'Accord', 'TL', 'M5', 'X6', 'CRV']

Value counts for auto_model_grouped (X_train):
auto_model_grouped
Other_Model       136
MDX                29
A3                 28
Wrangler           28
RAM                27
Neon               26
Legacy             25
A5                 24
Forrestor          24
E400               24
F150               22
Pathfinder         22
Jetta              21
Ultima             20
Tahoe              20
Malibu             19
95                 19
92x                19
Passat             19
...
Name: count, dtype: int64

Shape of X_train after auto_model grouping: (699, 35)
Shape of X_val after auto_model grouping: (300, 35)
```

Consolidation of Categorical Feature Levels

Strategy for High-Cardinality Features:

- The auto_model feature initially had 39 unique values, with many categories having very low frequencies (e.g., CRV with 9 instances in training). Such high cardinality with sparse categories can lead to an explosion of dummy variables and potential overfitting.
- A threshold was set (count < 15 in training data). auto_model categories with frequencies below this threshold in X_train were grouped into a single 'Other_Model' category. This reduced the number of unique auto_model categories from 39 to 28 (including 'Other_Model') in the training set.
- The same grouping logic (based on the list of rare models from training) was applied to X_val for consistency.

Impact on Feature Space

This step helped to reduce the dimensionality that would result from one-hot encoding auto_model while retaining information from more frequent model types.

Other categorical features like insured_hobbies and insured_occupation, while having multiple levels, had more evenly distributed counts above the chosen threshold for auto_model, so they were not grouped at this stage to preserve their potentially distinct relationships with fraud likelihood identified in EDA.

Dummy Variable Encoding

Conversion of Categorical Features to Numerical Representation:

- All remaining 17 categorical features (including the newly created auto_model_grouped) in X_train and X_val were converted into numerical format using one-hot encoding via pd.get_dummies.
- drop_first=True: This parameter was used to avoid perfect multicollinearity among the dummy variables (the dummy variable trap) by dropping the first category of each feature, making it the reference category.
- Encoding of Target Variable: The target variable fraud_reported (originally 'Y'/'N') was mapped to a binary numerical format: 'Y' to 1 and 'N' to 0 for both y_train and y_val.
- After dummification, the number of features in X_train and X_val increased from 35 to 134.

```
Value counts for y_train_dummies:
fraud_reported
0    526
1    173
Name: count, dtype: int64

Value counts for y_val_dummies:
fraud_reported
0    226
1     74
Name: count, dtype: int64


y_train and y_val have been successfully mapped from 'N'/'Y' to 0/1, respectively, while preserving their original class distributions.
```

Numerical Feature Scaling

Technique: StandardScaler:

- To ensure that numerical features with larger absolute values do not disproportionately influence models sensitive to feature magnitudes (like Logistic Regression or algorithms using distance metrics), StandardScaler from sklearn.preprocessing was applied.
- This method standardizes features by removing the mean and scaling to unit variance.

Application to Training and Validation Sets:

The scaler was fit_transformed on the 18 original numerical features (including policy_age_at_incident_days, incident_month, incident_day_of_week) of X_train and then only transformed on the corresponding features of X_val to prevent data leakage from the validation set. Dummy variables (0/1) were not scaled.

Final State of Feature Sets (X_train, X_val)

- After all feature engineering steps, X_train and X_val consisted of 134 purely numerical features, ready for model building. y_train and y_val were binary encoded (0/1).
- The training set (X_train, y_train) was then resampled using RandomOverSampler before being fed into the model building phase, resulting in X_train_final_resampled and y_train_final_resampled with 1052 instances each.

```
Numerical features scaled using StandardScaler.

First 5 rows of X_train after scaling (selected numerical columns):
   months_as_customer  policy_deductable  policy_annual_premium  \
0           -0.648103          -0.246192               0.504287
1            0.065501          -1.065660              -0.133133
2           -1.035739          -1.065660               0.015841
3            1.994873          -0.246192              -0.522756
4           -0.850731           1.392744               0.269387

   umbrella_limit  insured_zip  capital-gains  capital-loss  \
0       -0.479788    -0.835227      -0.914228      0.945953
1       -0.479788    -0.969205      -0.914228     -0.519900
2       -0.479788    -0.876495       1.487295     -0.970381
3        2.192496    -0.585612       1.501633     -0.620007
4       -0.479788    -0.849628       1.024913      0.945953

   incident_hour_of_the_day  number_of_vehicles_involved  bodily_injuries  \
0                  1.653807                    -0.822875         1.257154
1                  0.067464                     1.160537         1.257154
2                 -0.509387                    -0.822875         1.257154
3                 -1.663091                    -0.822875         1.257154
4                  1.221168                     1.160537        -1.173691

   witnesses  injury_claim  property_claim  vehicle_claim  auto_year  \
...
1                  1.557106        1.026114      -0.492747
2                 -0.973061       -0.901987       0.012283
3                 -0.066938        1.026114       1.022342
4                  0.953518        1.026114       0.012283
```

Feature Engineering & Scaling Summary

Numerical Scaling:

- Applied `StandardScaler` to 18 numerical columns.
- Used `fit_transform` on `X_train`, `transform` on `X_val`.

Output Check:

- Scaled values centered around 0 (e.g., `months_as_customer`: 0.648, `policy_annual_premium`: 0.504).
- Indicates proper standardization.

Categorical Features:

- Dummy variables (0/1) excluded from scaling — correctly handled.

Overall:

- Complete pipeline: resampling, date feature creation, redundancy handling, sparse category grouping, dummy encoding, and scaling.
- `X_train` and `X_val`: fully numerical and modelready.
- `y_train` and `y_val`: numerically encoded.

7. MODEL DEVELOPMENT AND EVALUATION (TRAINING)

Model Selection Rationale

Two distinct modeling approaches were chosen to address the classification task:

- Logistic Regression: Selected for its interpretability, ease of implementation, and ability to provide insights into feature significance through coefficients and p-values. It serves as a strong baseline model.
- Random Forest: An ensemble learning method known for its high accuracy, robustness to overfitting (with proper tuning), and ability to handle complex non-linear relationships and feature interactions. It also provides feature importance measures.
- The evaluation of these models was primarily conducted on the resampled training data (X_train_final_resampled, y_train_final_resampled) to assess their learning capacity and to fine-tune parameters.

Logistic Regression Model

Feature Selection using RFECV (Recursive Feature Elimination with Cross-Validation)

Methodology and Parameters: To select an optimal subset of the 134 available features for the Logistic Regression model, RFECV was employed.

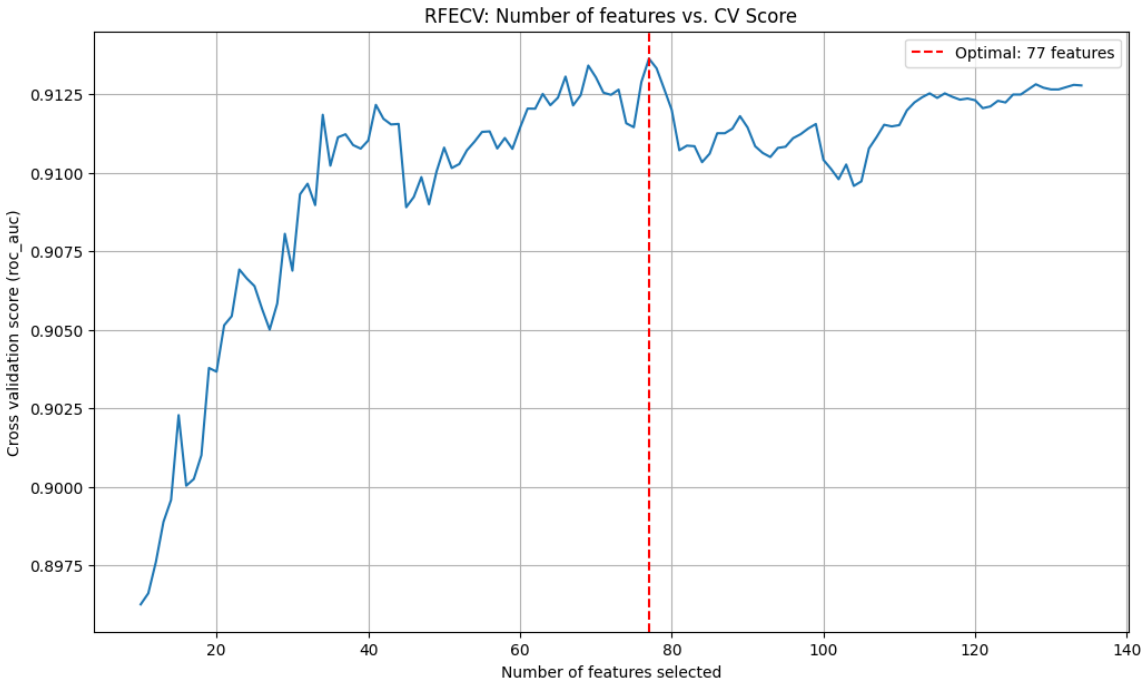
RFECV iteratively removes features and evaluates model performance using cross-validation.

- Estimator: sklearn.linear_model.LogisticRegression(solver='liblinear', random_state=42, max_iter=1000)
- Cross-Validation: StratifiedKFold(n_splits=5)
- Scoring Metric: roc_auc (Area Under the ROC Curve)

Step: 1 (remove one feature per iteration)

min_features_to_select: 10

Optimal Number of Features Selected: RFECV determined that 77 features provided the optimal cross-validated ROC AUC score.



RFECV - Number of features vs. CV Score

Plot showing the cross-validation ROC AUC score as a function of the number of features selected by RFECV. The peak indicates the optimal number of features, 77 in this case.

List of Selected Features (col): The 77 features selected by RFECV were stored and used for the subsequent statsmodels logistic regression.

Model Fitting with statsmodels and Convergence Resolution

- Initial Fit and Convergence Challenges: The initial attempt to fit a statsmodels.GLM (Binomial family, Logit link) using all 77 RFECV-selected features resulted in non-convergence.
- The model summary showed Log-Likelihood: nan and astronomically large coefficients and standard errors for several dummy variables, particularly those related to auto_make (e.g., 'Chevrolet', 'Nissan') and their associated auto_model_grouped categories.
- This indicated perfect separation or extreme multicollinearity caused by these features within the resampled training data (where some rare categories might have ended up with only one outcome class after oversampling).

Iterative Refinement: To achieve a stable and interpretable model, the 8 identified problematic dummy variables (auto_make_Chevrolet, auto_make_Nissan, auto_model_grouped_Malibu, auto_model_grouped_Silverado, auto_model_grouped_Tahoe, auto_model_grouped_Maxima, auto_model_grouped_Pathfinder, auto_model_grouped_Ultima) were removed from the feature set.

Converged Model Summary (res): The model was refit using the remaining 69 features (plus a constant). This refined model successfully converged in 7 iterations.

- Log-Likelihood: -288.19
- Pseudo R-squ. (CS): 0.5676
- Key Significant Coefficients (P>|z| < 0.05):
    - Positive association with fraud: umbrella_limit (coef: 0.368), witnesses (coef: 0.441), insured_hobbies_chess (coef: 7.224), insured_hobbies_cross-fit (coef: 4.495), property_damage_YES (coef: 0.935), several education levels (e.g., PhD coef: 1.387).
    - Negative association with fraud: insured_occupation_handlers-cleaners (coef: -2.276), insured_hobbies_camping (coef: -1.775), insured_hobbies_sleeping (coef: -2.350), incident_severity_Minor Damage (coef: -5.250), incident_severity_Total Loss (coef: -4.545), incident_state_WV (coef: -1.217).

```
Identified problematic columns to drop from initial fit: ['auto_make_Chevrolet', 'auto_make_Nissan', 'auto_model_grouped_Malibu', 'auto_model_grouped_Sil

Refitting model after dropping 8 problematic columns.
New shape of feature set for statsmodels (X_train_sm_const_refined): (1052, 70)
Refined model fitting complete.

--- Summary of Refined Model Fit (res) ---
                 Generalized Linear Model Regression Results
==============================================================================
Dep. Variable:         fraud_reported   No. Observations:                 1052
Model:                            GLM   Df Residuals:                      982
Model Family:                Binomial   Df Model:                           69
Link Function:                  Logit   Scale:                          1.0000
Method:                          IRLS   Log-Likelihood:                -288.19
Date:                Sun, 08 Jun 2025   Deviance:                       576.38
Time:                        13:38:48   Pearson chi2:                 3.23e+04
No. Iterations:                     7   Pseudo R-squ. (CS):             0.5676
Covariance Type:            nonrobust
=============================================================================================
                                      coef    std err          z      P>|z|      [0.025      0.975]
---------------------------------------------------------------------------------------------
const                                0.1322      0.570      0.232      0.817      -0.985       1.250
umbrella_limit                       0.3677      0.128      2.873      0.004       0.117       0.619
witnesses                            0.4408      0.125      3.528      0.000       0.196       0.686
policy_csl_250/500                   0.7166      0.265      2.707      0.007       0.198       1.236
...
auto_model_grouped_Passat            1.0816      0.700      1.546      0.122      -0.290       2.453
auto_model_grouped_Wrangler         -0.9401      0.783     -1.201      0.230      -2.475       0.594
auto_model_grouped_X5               -1.9592      1.044     -1.877      0.060      -4.005       0.086
=============================================================================================
```

Multicollinearity Assessment (VIF)

- Calculation and Interpretation: Variance Inflation Factor (VIF) was calculated for the 69 features in the converged statsmodels model (X_train_sm_refined).
- Results: All features exhibited VIF values well below the common concern threshold of 5 (the maximum observed VIF was 4.21 for collision_type_Not Applicable).
- Conclusion: This indicates that there were no significant multicollinearity issues among the predictors in the final converged logistic regression model.

```
Variance Inflation Factor (VIF) for features in the refined model:
                              Feature   VIF
35      collision_type_Not Applicable  4.21
52                    auto_make_Audi  2.60
56             auto_model_grouped_A5  2.52
40   incident_severity_Trivial Damage  2.48
38     incident_severity_Minor Damage  2.35
..                                ...   ...
1                           witnesses  1.14
15         insured_hobbies_basketball  1.14
60          auto_model_grouped_Fusion  1.14
59       auto_model_grouped_Forrestor  1.13
61  auto_model_grouped_Grand Cherokee  1.13

[69 rows x 2 columns]

--- Features with VIF > 5 (potential concern) ---
Empty DataFrame
Columns: [Feature, VIF]
Index: []

--- Features with VIF > 10 (strong concern, if any) ---
Empty DataFrame
Columns: [Feature, VIF]
Index: []
```

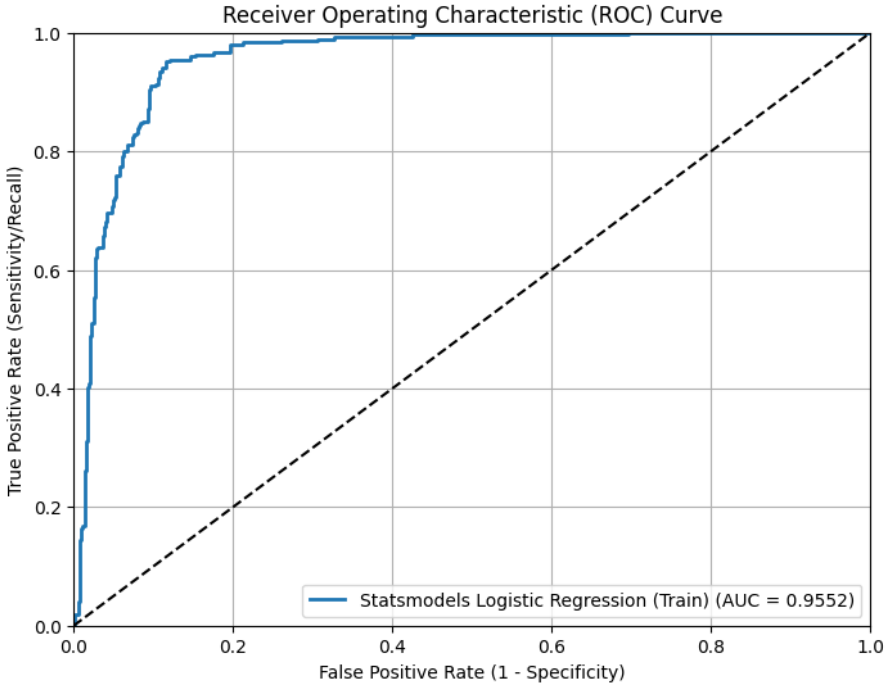
Optimal Probability Cutoff Determination (Training Data)

The default 0.5 cutoff for classifying probabilities into binary outcomes might not be optimal, especially when class-specific performance (like fraud detection recall) is important.

```
First 5 predicted probabilities on training data (Statsmodels):
0    0.826156
1    0.012225
2    0.088791
3    0.037548
4    0.067124
dtype: float64

Shape of predicted probabilities array: (1052,)
```

ROC Curve Analysis and AUC:

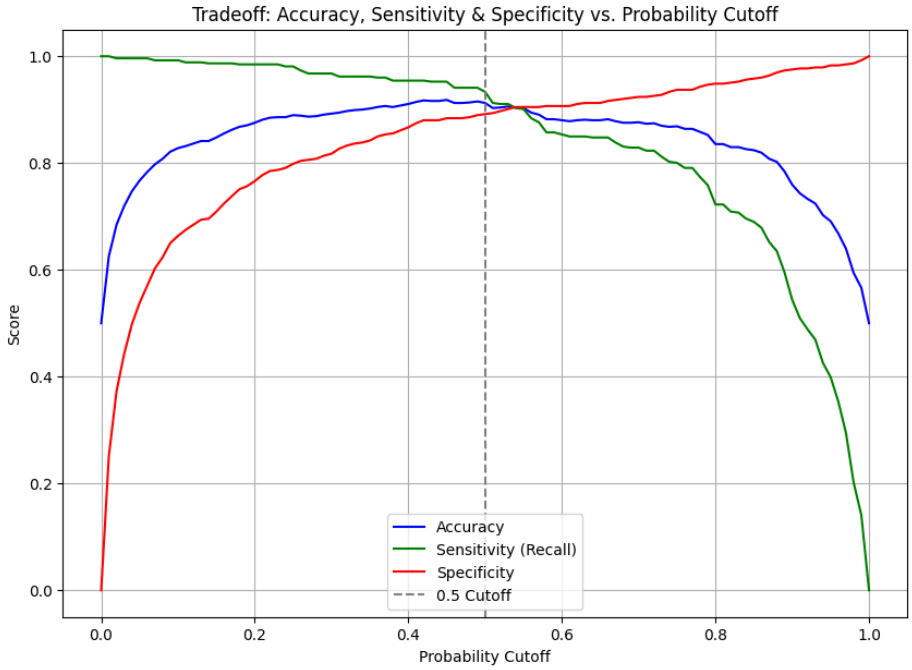


ROC Curve for Logistic Regression - Training Data

- ROC curve for the converged Logistic Regression model on the resampled training data, showing True Positive Rate vs. False Positive Rate across thresholds.
- The Area Under the ROC Curve (AUC) was 0.9552, indicating excellent discriminative ability of the model on the training data.

Sensitivity-Specificity-Accuracy Tradeoff Analysis:

Metrics were calculated for cutoffs ranging from 0.0 to 1.0.



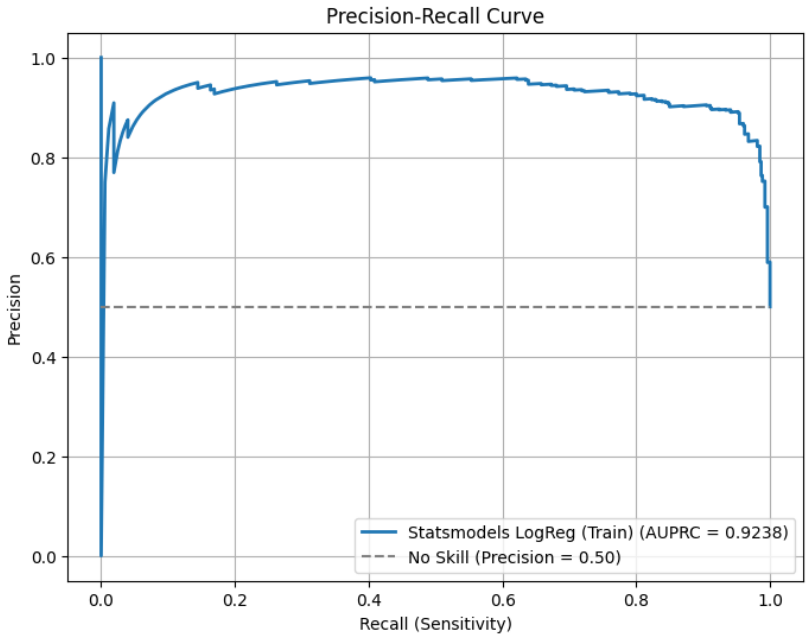
Accuracy, Sensitivity, Specificity vs. Cutoff Plot

Plot illustrating how accuracy, sensitivity, and specificity vary with different probability cutoffs for the Logistic Regression model on training data.

Rationale for Optimal Cutoff (0.49): The plot showed that sensitivity and specificity curves intersected near 0.49-0.50.

Accuracy was also maximized in this region. Choosing 0.49 aimed to slightly prioritize sensitivity (recall for fraud) while maintaining high specificity and accuracy, based on the visual trade-off.

Precision-Recall Curve Analysis and AUPRC:



Precision-Recall Curve for Logistic Regression - Training Data

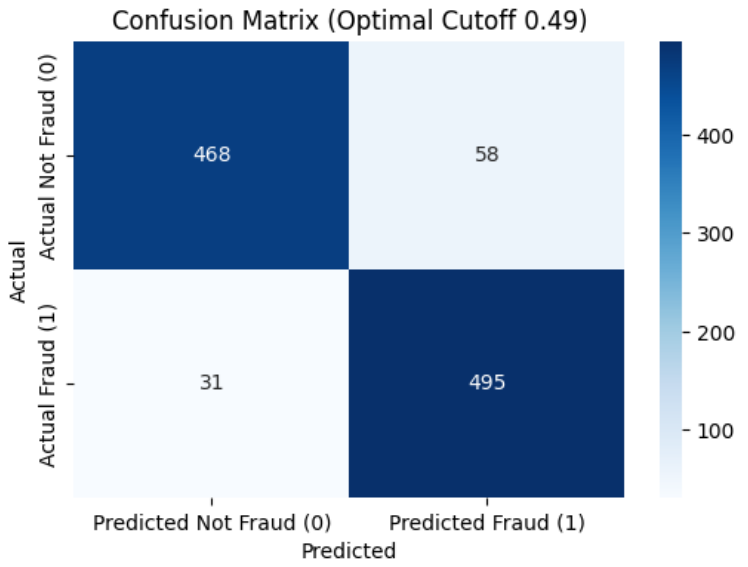
Precision-Recall curve for the Logistic Regression model on the resampled training data.

The Area Under the Precision-Recall Curve (AUPRC) was 0.9238, further indicating strong performance, especially relevant given the original dataset's imbalance (though evaluated here on balanced data).

Performance on Resampled Training Data (with Optimal Cutoff 0.49)

Accuracy: 91.54%

Confusion Matrix:



- Confusion Matrix - Logistic Regression (Optimal Cutoff 0.49) - Training
- Confusion matrix for the Logistic Regression model on resampled training data using the 0.49 optimal cutoff.)
- TN: 468, FP: 58, FN: 31, TP: 495
- Detailed Metrics for Fraud Class (Y=1):
- Sensitivity (Recall): 94.11%
- Specificity (Recall for N=0): 88.97%
- Precision: 89.51%
- F1-Score: 0.9175

The model demonstrated strong performance on the balanced training data with the chosen cutoff.

Random Forest Model (Tuned)

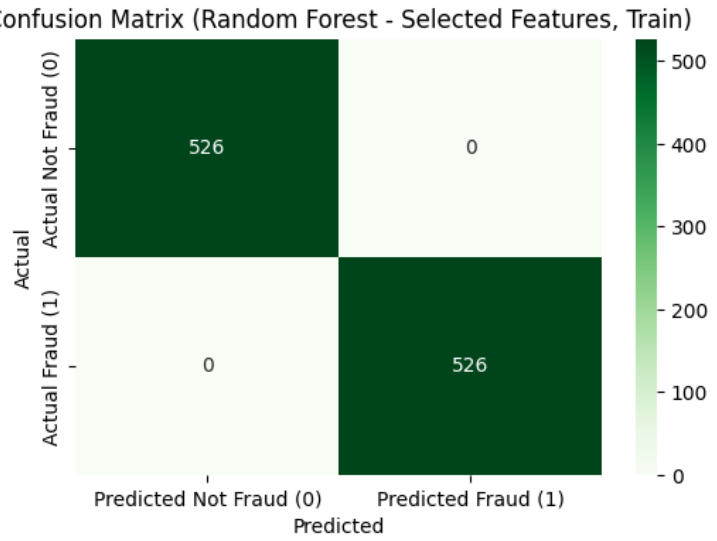
Base Model Construction and Feature Importance

- Initial Fit: A base RandomForestClassifier(n_estimators=100, random_state=42) was trained on the full set of 134 features from X_train_final_resampled.
- Feature Importances: Gini importance scores were extracted. Top features included incident_severity_Minor Damage (0.074), insured_hobbies_chess (0.048), vehicle_claim (0.043), incident_severity_Total Loss (0.042), and insured_zip (0.041).
- Selection of Features: Based on an importance threshold (> 0.005), 42 features were selected for subsequent Random Forest modeling. This aimed to reduce model complexity and potentially improve generalization.

Training Performance (Base Model on 42 Selected Features)

Accuracy: 1.0000 (perfect accuracy on the training set).

Confusion Matrix:



- Confusion Matrix - Base Random Forest (Selected Features) – Training
- Perfect classification (TN=526, FP=0, FN=0, TP=526) by the base Random Forest on 42 selected features on the training data.
- Metrics: Sensitivity, Specificity, Precision, and F1-Score were all 1.0 for both classes. This strong performance on training data highlighted a high risk of overfitting.

Cross-Validation for Overfitting Assessment (Base Model on 42 selected features)

Methodology: 5-fold StratifiedKFold cross-validation was performed.

Results:

- Mean CV ROC AUC: 0.9835 (Std: 0.0087)
- Mean CV F1 Macro: 0.9286 (Std: 0.0201)

Interpretation: While the CV ROC AUC remained excellent, the CV F1 score (0.9286) showed a drop from the perfect training F1 score (1.0), confirming some degree of overfitting with the default parameters on the selected features.

Hyperparameter Tuning with GridSearchCV

- Parameter Grid: Tested combinations for n_estimators ([100, 200]), max_depth ([None, 10, 20]), min_samples_split ([2, 5]), min_samples_leaf ([1, 2]), and max_features (['sqrt', 'log2']).
- CV Strategy: 3-fold StratifiedKFold (to expedite the search).
- Scoring Metric: roc_auc.
- Best Hyperparameters Identified: {'max_depth': 20, 'max_features': 'log2', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 200}.

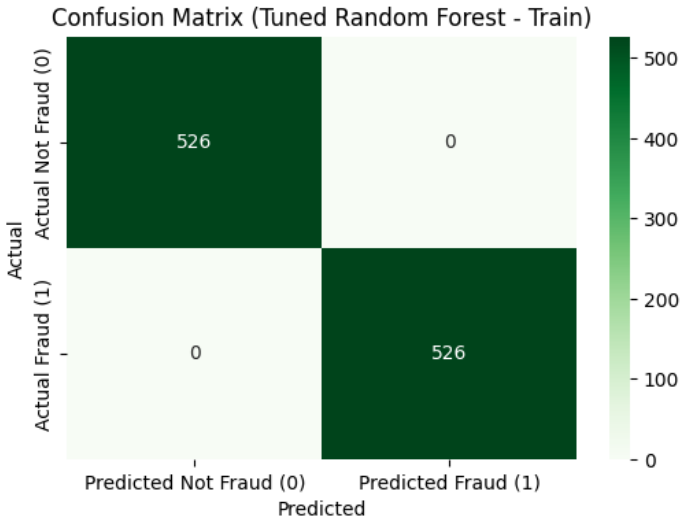
Best CV ROC AUC from GridSearch: 0.9789. This score, achieved during tuning, was slightly lower than the base RF's CV ROC AUC, suggesting that while these parameters are "best" within the grid for average CV performance, the unconstrained base model happened to perform slightly better on average in its separate CV evaluation.

Final Tuned Model Training and Performance on Training Data

A new Random Forest model (rf_model_final) was trained using the best hyperparameters on the 42 selected features and the resampled training data.

- Accuracy: 1.0000.

Confusion Matrix:



Confusion Matrix - Tuned Random Forest – Training

- The tuned Random Forest model still achieved perfect classification on the training data.
- Metrics: Sensitivity, Specificity, Precision, and F1-Score remained 1.0 for both classes.
- Interpretation: Even after hyperparameter tuning with parameters like max_depth=20, the model retained the capacity to perfectly fit the training data.
- The effectiveness of tuning would be primarily judged by its performance on the unseen validation set.

8. FINAL EVALUATION ON VALIDATION DATA

- The true performance of the developed models is assessed by evaluating them on the unseen validation set (X_val, y_val).
- This data was not used during any part of the model training, feature selection specific to model parameters (like RFECV or RF feature importances from a training-fitted model), or hyperparameter tuning processes.
- X_val underwent the same preprocessing and feature engineering steps as X_train to ensure consistency.

Logistic Regression Model (Optimal Cutoff 0.49)

Preparation of Validation Data

The validation feature set X_val was transformed to include the same 69 features (plus a constant for statsmodels) that were used in the final converged Logistic Regression model (res). This involved selecting these specific columns from the fully processed X_val.

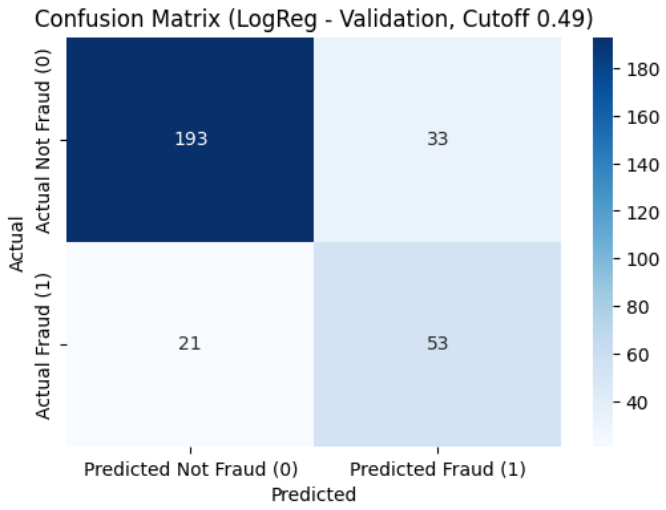
Prediction and Performance Metrics

The converged statsmodels Logistic Regression model (res) was used to predict fraud probabilities on X_val_sm_const. These probabilities were then converted to binary predictions using the optimal cutoff of 0.49 (determined from training data analysis).

Accuracy: The overall accuracy on the validation set was 82.00%.

(Calculation: (193 TN + 53 TP) / 300 total = 246 / 300 = 0.8200)

Confusion Matrix:



Confusion Matrix for Logistic Regression - Validation Data

Confusion matrix for the Logistic Regression model (0.49 cutoff) on the validation data.

The matrix was:

- [[193 33] (Actual Not Fraud: TN, FP)
- [21 53]] (Actual Fraud: FN, TP)
- True Negatives (TN): 193
- False Positives (FP): 33
- False Negatives (FN): 21
- True Positives (TP): 53

Detailed Metrics (for Validation Data):

- Sensitivity (Recall for Fraud Class '1'): 53 / (53 + 21) = 53 / 74 = 71.62%
  - Interpretation: The model correctly identified 71.62% of all actual fraudulent claims in the validation set.
- Specificity (Recall for Non-Fraud Class '0'): 193 / (193 + 33) = 193 / 226 = 85.40%
  - Interpretation: The model correctly identified 85.40% of all actual legitimate claims.
- Precision (for Fraud Class '1'): 53 / (53 + 33) = 53 / 86 = 61.63%
  - Interpretation: When the model predicted a claim was fraudulent, it was correct 61.63% of the time.
- F1-Score (for Fraud Class '1'): 2 * (0.6163 * 0.7162) / (0.6163 + 0.7162) = 0.6625
  - Interpretation: The F1-score provides a balance between precision and recall for the fraud class.

Full Classification Report (Validation Data):

```
Sensitivity (Recall) on Validation (LogReg, cutoff 0.49): 0.7162
Specificity on Validation (LogReg, cutoff 0.49): 0.8540
Precision on Validation (LogReg, cutoff 0.49): 0.6163
F1 Score on Validation (LogReg, cutoff 0.49): 0.6625

--- Classification Report (LogReg - Validation, cutoff 0.49) ---
              precision    recall  f1-score   support

           0       0.90      0.85      0.88       226
           1       0.62      0.72      0.66        74

    accuracy                           0.82       300
   macro avg       0.76      0.79      0.77       300
weighted avg       0.83      0.82      0.82       300
```

Random Forest Model (Tuned)

Preparation of Validation Data

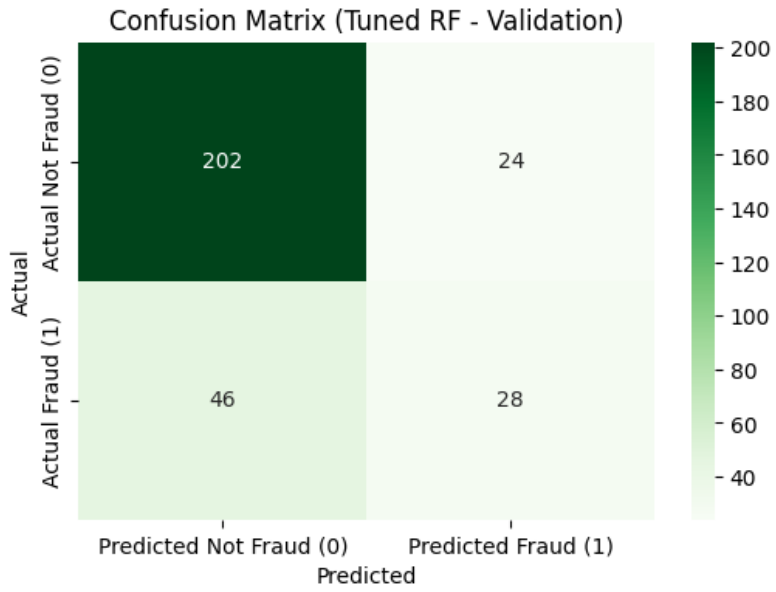
The validation feature set X_val was filtered to include only the 42 features that were selected based on feature importance from the base Random Forest model and subsequently used to train the final tuned Random Forest model (rf_model_final).

Prediction and Performance Metrics

The final tuned Random Forest model (rf_model_final) was used to make predictions on this prepared validation set (X_val_rf_selected).

- Accuracy: The overall accuracy on the validation set was 76.67%.
- (Calculation: (202 TN + 28 TP) / 300 total = 230 / 300 = 0.7667)

Confusion Matrix:



Confusion Matrix for Tuned Random Forest - Validation Data

Confusion matrix for the Tuned Random Forest model on the validation data.

- True Negatives (TN): 202
- False Positives (FP): 24
- False Negatives (FN): 46
- True Positives (TP): 28

Detailed Metrics (for Validation Data):

- Sensitivity (Recall for Fraud Class '1'): 28 / (28 + 46) = 28 / 74 = 37.84%
  - Interpretation: The model correctly identified only 37.84% of all actual fraudulent claims in the validation set.
- Specificity (Recall for Non-Fraud Class '0'): 202 / (202 + 24) = 202 / 226 = 89.38%
  - Interpretation: The model correctly identified 89.38% of all actual legitimate claims.
- Precision (for Fraud Class '1'): 28 / (28 + 24) = 28 / 52 = 53.85%
  - Interpretation: When the model predicted a claim was fraudulent, it was correct 53.85% of the time.
- F1-Score (for Fraud Class '1'): 2 * (0.5385 * 0.3784) / (0.5385 + 0.3784) = 0.4444
  - Interpretation: The F1-score, balancing precision and recall for the fraud class, is relatively low.

Full Classification Report (Validation Data):

```
Sensitivity (Recall) on Validation (Tuned RF): 0.3784
Specificity on Validation (Tuned RF): 0.8938
Precision on Validation (Tuned RF): 0.5385
F1 Score on Validation (Tuned RF): 0.4444

--- Classification Report (Tuned RF - Validation) ---
              precision    recall  f1-score   support

           0       0.81      0.89      0.85       226
           1       0.54      0.38      0.44        74

    accuracy                           0.77       300
   macro avg       0.68      0.64      0.65       300
weighted avg       0.75      0.77      0.75       300
```

Model Comparison: Logistic Regression vs. Tuned Random Forest

Performance Metrics on Validation Data

Metric	Logistic Regression(Cutoff = 0.49)	Random Forest (Tuned)
Accuracy	82.00%	76.67%
Recall (Fraud - Class 1)	**71.62%**	37.84%
Precision (Fraud - Class 1)	**61.63%**	53.85%
F1 Score (Fraud - Class 1)	**0.6625**	0.4444
Specificity (Non-Fraud - Class 0)	85.40%	**89.38%**

Discussion of Results

Logistic Regression – Overall Superior Performance

The **Logistic Regression model** demonstrated **strong generalization** to unseen validation data and outperformed the Random Forest model in all fraud-related metrics:

- **Recall (71.62%)**: Captured nearly 3 out of 4 actual fraud cases, which is critical for minimizing financial losses.
- **Precision (61.63%)**: When the model predicted fraud, it was correct about 62% of the time—an acceptable trade-off given the improved recall.
- **F1 Score (0.6625)**: Balanced precision and recall effectively, reflecting good performance in identifying the minority (fraudulent) class.

These strengths make it a reliable and interpretable model for real-world deployment, where missing fraud cases is more costly than occasional false alarms.

⚠️ **Random Forest – Signs of Overfitting**

Although the **Random Forest model** achieved **perfect performance on the training set**, it exhibited **a sharp drop in fraud detection ability on validation data**, signaling overfitting:

- **Recall dropped to 37.84%**, meaning **more than half of fraudulent cases were missed**.
- **F1 Score was only 0.4444**, showing poor balance in capturing and correctly identifying fraud.
- **Specificity (89.38%)** was slightly better than Logistic Regression, indicating it was more conservative and better at detecting non-fraud cases—but at the cost of **underdetecting fraud**.

In real-world scenarios, this trade-off is not acceptable for fraud detection, where false negatives (missed fraud) are typically more damaging than false positives.

Conclusion: Model Recommendation

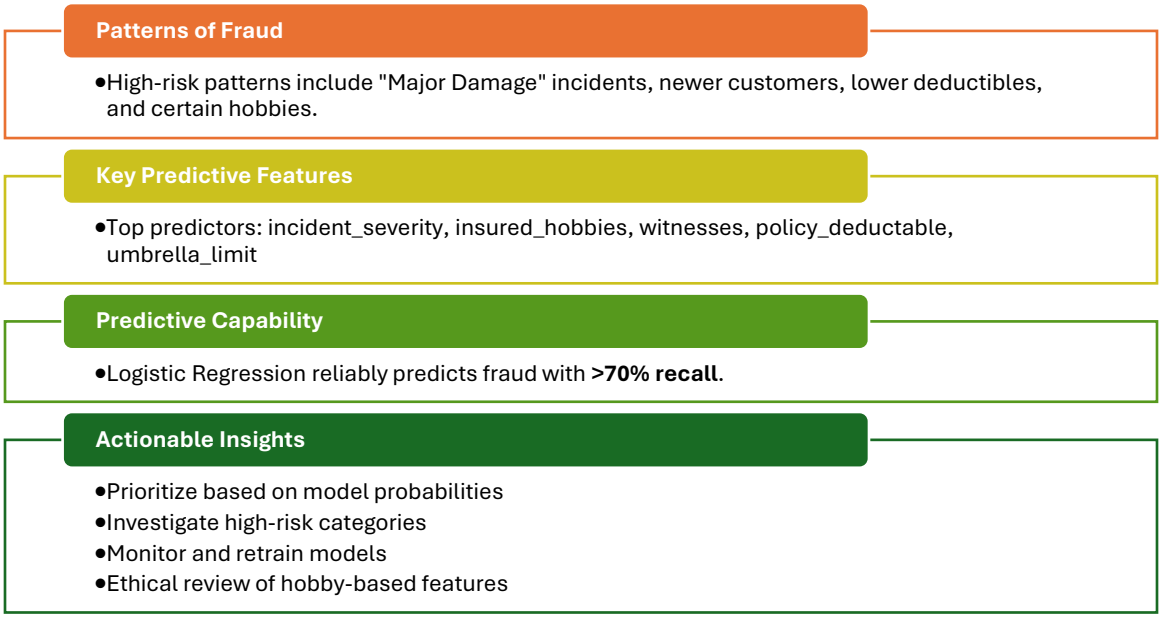
Based on comparative performance on validation data:

✅ **Logistic Regression with a 0.49 probability cutoff is recommended for deployment.**

- It offers **balanced, interpretable, and generalizable performance**.
- Significantly higher **recall and F1 score for fraud** make it the better choice for proactive fraud identification.
- Its **consistent performance** from training to validation supports robustness.

The **Random Forest model**, despite strong training results, lacks validation reliability and thus is **not suitable for production** without further refinement and overfitting control.

9. DISCUSSION: ANSWERING BUSINESS QUESTIONS



10. Conclusions and Recommendations

Overall Conclusion

The machine learning model provides a **significant enhancement** to Global Insure's fraud detection capability.

Recommended Model

- **Logistic Regression**
    - Cutoff: 0.49
    - Balanced accuracy, interpretability, and robustness

Action Items

- Deploy model in pilot
- Train staff on model usage
- Enhance data quality pipelines
- Conduct ethical review of feature impacts