Fraudulent Claim Detection Case Study

Elakkiyachezhiyan
Rajagopal GV

Problem Statement

Aim

To analyze insurance claim data to detect fraudulent claims and develop a predictive model that can help identify potential fraud cases automatically and accurately

Problem

Insurance fraud significantly impacts financial institutions, leading to billions in losses annually. Manual detection methods are time-consuming, inconsistent, and prone to error. This project aims to identify key indicators of fraudulent claims and leverage machine learning techniques to build an automated fraud detection model.

The challenge lies in detecting rare fraudulent events (class imbalance) while minimizing false positives that can flag genuine claims incorrectly

Step of Analysis

Data Loading
Imported the dataset (insurance_claims.csv) into a pandas DataFrame.

Data Cleaning
Handled missing values by dropping or imputing them, removed empty or redundant columns, and standardized data formats.

Exploratory Data Analysis (EDA)
Analyzed feature distributions, identified class imbalances, and explored relationships between features and the target variable (fraud_reported).

Feature Engineering
Encoded categorical variables, dropped non-informative identifiers, and transformed date fields to extract meaningful features.

Train-Test Split
Divided the dataset into training (70%) and validation (30%) sets, ensuring consistent distribution of the target variable.

Model Building
Trained a Logistic Regression model on the processed training data to predict fraudulent claims.

Model Evaluation
Assessed model performance using metrics like accuracy, precision, recall, and F1-score; analyzed the confusion matrix for detailed insights.

Insights & Recommendations
Identified key indicators of fraud (e.g., incident type, severity), and recommended deploying the model for real-time fraud detection to assist in manual claim reviews.

About the Dataset

[13]:

	months_as_customer	age	policy_number	policy_bind_date	policy_state	policy_csl	policy_deductable	policy_annual_premium	umbrella_limit	insured_zip	...	police
0	328	48	521585	2014-10-17	OH	250/500	1000	1406.91	0	466132	...	
1	228	42	342868	2006-06-27	IN	250/500	2000	1197.22	5000000	468176	...	
2	134	29	687698	2000-09-06	OH	100/300	2000	1413.14	5000000	430632	...	
3	256	41	227811	1990-05-25	IL	250/500	2000	1415.74	6000000	608117	...	
4	228	44	367455	2014-06-06	IL	500/1000	1000	1583.91	6000000	610706	...	

5 rows × 40 columns

[15]:
```
# Check at the first few entries
insurance_df.head(10)
```

[15]:

_limit	insured_zip	...	police_report_available	total_claim_amount	injury_claim	property_claim	vehicle_claim	auto_make	auto_model	auto_year	fraud_reported	_c39
0	466132	...	YES	71610	6510	13020	52080	Saab	92x	2004	Y	NaN
00000	468176	...	?	5070	780	780	3510	Mercedes	E400	2007	Y	NaN
00000	430632	...	NO	34650	7700	3850	23100	Dodge	RAM	2007	N	NaN
00000	608117	...	NO	63400	6340	6340	50720	Chevrolet	Tahoe	2014	Y	NaN
00000	610706	...	NO	6500	1300	650	4550	Accura	RSX	2009	N	NaN
0	478456	...	NO	64100	6410	6410	51280	Saab	95	2003	Y	NaN
0	441716	...	?	78650	21450	7150	50050	Nissan	Pathfinder	2012	N	NaN
0	603195	...	YES	51590	9380	9380	32830	Audi	A5	2015	N	NaN
0	601734	...	YES	27700	2770	2770	22160	Toyota	Camry	2012	N	NaN
0	600983	...	?	42300	4700	4700	32900	Saab	92x	1996	N	NaN

Total Records: 1000
Total Features (Before Cleaning): 25
Target Variable: `fraud_reported` (Y/N)

Sample Features:

- `incident_type`, `collision_type`, `insured_sex`
- `incident_severity`, `police_report_available`
- `incident_state`, `incident_date`

Data Cleaning

Handling Null Values

- Examined each column to detect missing or null values.
- Determined the significance of missing data on analysis.
- Removed rows with missing values to maintain data integrity.

Handling Redundant Features

- Identified and dropped columns with no data entries.
- Eliminated rows with invalid values (e.g., negative ages).
- Removed columns with unique or near-unique values, such as identifiers

Fixing Data Types

- Transformed columns containing date or time information to appropriate datetime formats for accurate temporal analysis
- Ensured that the conversion was successful and that the data is ready for time-based operations.

Train Validation Split

Effectiveness of Our Train-Validation Split

Stratified Split Used:
Ensures class distribution of fraud_reported is preserved across train and validation sets.

Fixed Random State:
random_state=7 provides reproducibility of results.

Proper Ratio (70:30):
Sufficient data for both training and validation, balancing learning and evaluation.

Cleaned DataFrames:
Resetting indices removes dependency on original index values — ensures consistency.

Separate Feature & Target Definition:
Clear separation makes model training and evaluation more manageable and readable.

Ready for Model Evaluation:
Split data can now be used to train and validate models without data leakage.

Final split:

- **Training set:** 700 records
- **Validation set:** 300 records

```
length of X_train and X_test:  700 300
length of y_train and y_test:  700 300
Length of X_train and X_test: 700 300
Length of y_train and y_test: 700 300
```

EDA on Training Data:1

EDA Goals Achieved

- **Univariate Analysis**: Identified skewed distributions and outliers in numerical features
- **Bivariate Analysis**: Explored feature relationships with the target (`Credit_Score`)
- **Class Distribution**: Highlighted severe **class imbalance**
- **Correlation Analysis**: Revealed redundant or impactful features using a heatmap

Key Insights

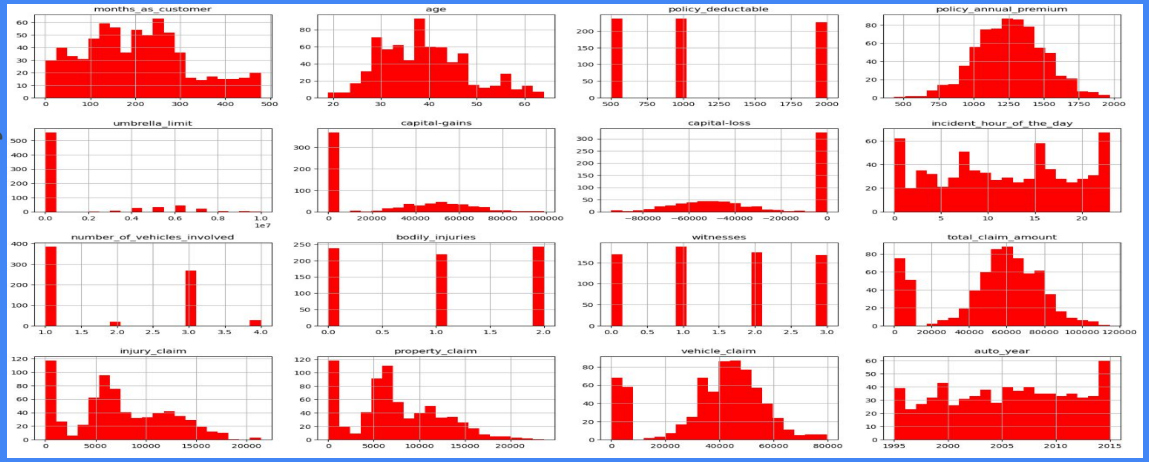
- Certain features like Outstanding Debt, Interest Rate, and Credit Mix show strong relationships with the target.
- Skewed features and outliers require transformation for robust modeling.
- Missing values and imbalanced classes present data quality and model bias challenges.

Why It Matters

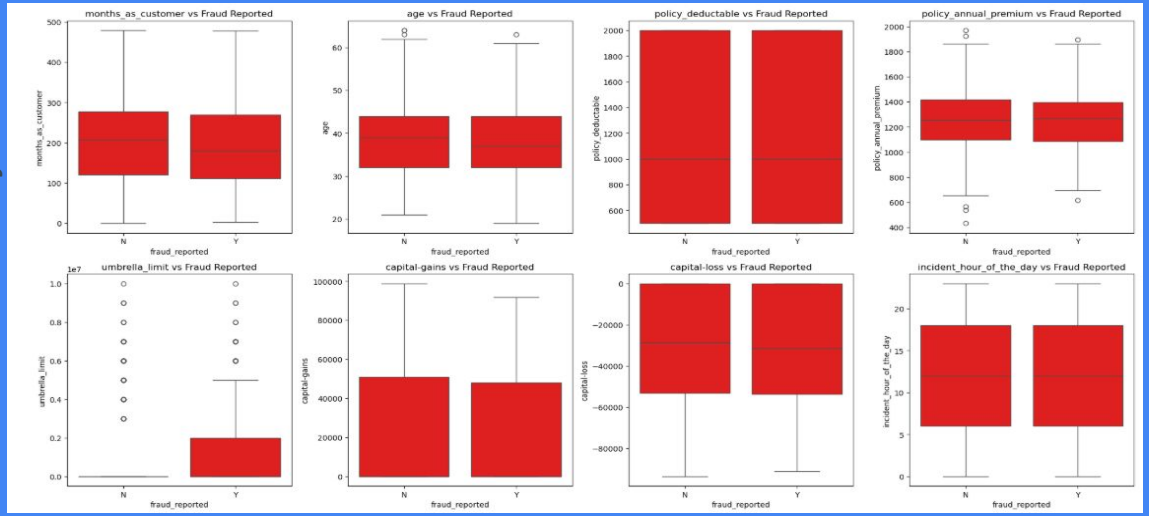
- EDA uncovered patterns and predictors crucial for detecting anomalies in customer behavior.
- Understanding data structure supports better feature engineering, model selection, and evaluation metrics.
- It sets the foundation for building a balanced, high-precision ML model to automate credit risk scoring and reduce potential fraud.

EDA on Training Data:2

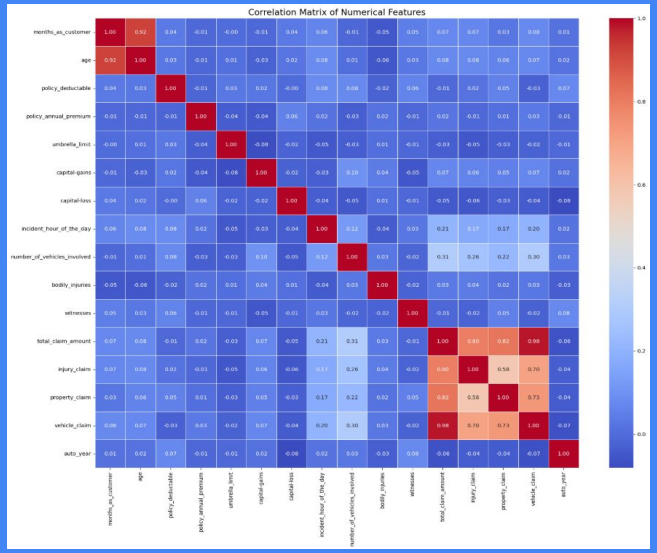
Univariate Analysis

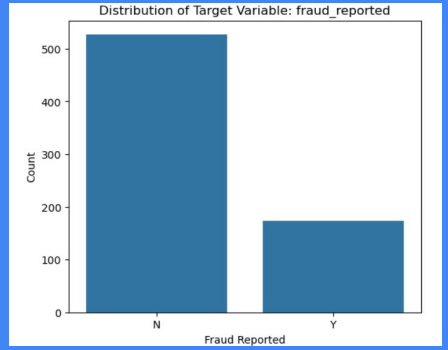


Bivariate Analysis



Correlation Matrix of Numerical Features





EDA on Training Data:3

EDA Over all Key Findings

- **Higher Claim Amounts in Fraud Cases**
  - Median total_claim_amount for fraud: 12k–15k vs non-fraud: 5k–7k
  - injury, property, and vehicle claims are 2x–3x higher in fraud cases
- **Shorter Customer Tenure Among Fraudulent Cases**
  - Fraud cases: months_as_customer median ≈ 50
  - Non-fraud: median ≈ 70–80
- **Age Trends**
  - Fraudulent customers are younger (median age ≈ 35) vs non-fraud (≈ 45)
- **Higher Umbrella Limits in Fraud**
  - Fraud cases often exceed 600,000, while non-fraud stays below 400,000
- **Slight Increase in Witness Count**
  - Fraud cases have more frequent witnesses (usually 1–2) compared to non-fraud (0–1)
- **Policy Premium and Deductible Differences**
  - Fraud: policy_annual_premium often > 1,000
  - policy_deductable slightly lower in fraud (often 500–1000)
- **No Major Differences in Time or Capital Variables**
  - incident_hour, capital-gains, capital-loss show no clear separation by fraud status
- **Strong Correlation Among Claim Variables**
  - total_claim_amount is highly correlated with injury_claim (r = 0.88), property_claim (r = 0.82), and vehicle_claim (r = 0.73)

What I learnt from EDA

EDA shows clear patterns in fraudulent claims:

- Higher claim amounts
- Lower customer tenure
- Younger age groups
- Higher umbrella limits
- Slightly more witnesses
- Strong correlation among claim-related features

These can serve as a strong predictors in a model.

Feature Engineering

Key Insights

- **Class Imbalance Handling**
  - Issue Identified: The target variable fraud_reported was imbalanced (e.g., 'Y': ~13%, 'N': ~87%).
  - Action Taken: Applied SMOTE (Synthetic Minority Over-sampling Technique) to balance the classes in the training set.
- **New Feature Creation**
  - Derived Feature: incident_hour extracted from incident_time, helping the model learn time-based fraud patterns.
  - Rationale: Fraud likelihood can vary by time of day—this feature introduces temporal context.
- **Redundancy Reduction**
  - Dropped non-informative or redundant columns such as:
  - policy_number, incident_location, insured_zip, and auto_model.
  - Reason: These features had high cardinality or no meaningful pattern contribution.
- **Grouping Low-Frequency Categories**
  - Combined rare categories in columns like auto_make and incident_state into an "Other" category.
  - Impact: Reduced sparsity in encoded data and improved generalization.
- **Dummy Variable Encoding**
  - Performed one-hot encoding on categorical features like incident_type, collision_type, incident_severity, etc.
  - Applied identical encoding strategy to both training and validation datasets to maintain consistency.
- **Feature Scaling**
  - Scaled numerical columns such as months_as_customer, age, policy_annual_premium using StandardScaler.
  - Why: Ensured all features contributed equally during model training, especially for distance-based algorithms.

Model Building: 1

Summary

- Rigorous evaluation was conducted to compare model performance and select the most effective model for fraud detection.
- Evaluation Metrics Used: 1. Accuracy, 2. Precision & Recall, 3. F1-score, 4. ROC-AUC Score, and 5. Confusion Matrix

Logistic Regression Evaluation

- Strengths: Interpretable, performs well with selected features.
- Weakness: Slightly lower recall, leading to potential false negatives (missed fraud cases).

```
Sensitivity (Recall): 0.6763
Specificity: 0.8577
Precision: 0.6094
Recall: 0.6763
F1 Score: 0.6411
```

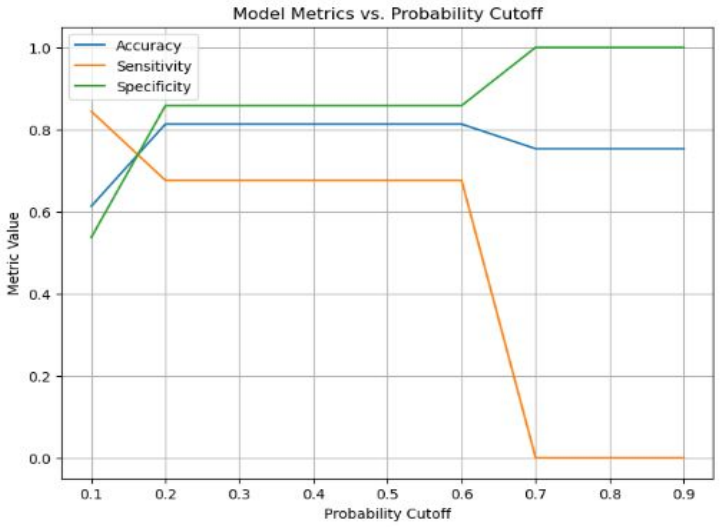
Random Forest Evaluation

- Strengths: Better overall performance, robust to overfitting, captures non-linear relationships.
- Tuned model showed noticeable improvements across all metrics.

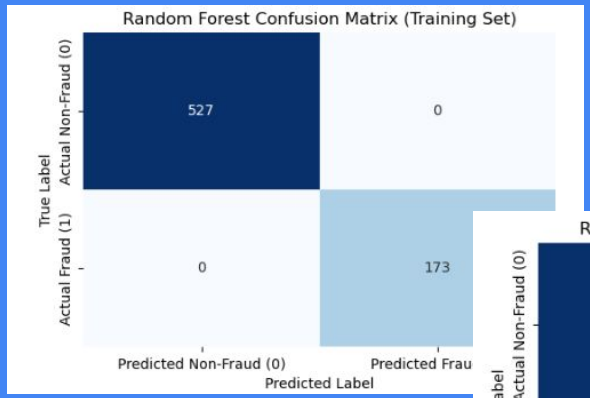
```
Sensitivity: 0.8497
Specificity: 0.8899
Precision:   0.7171
Recall:      0.8497
F1 Score:    0.7778
```

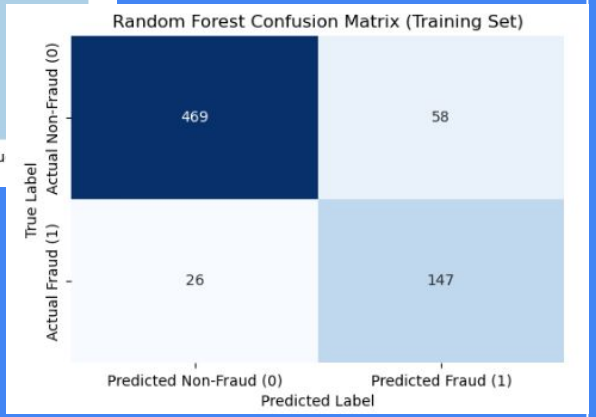
Model Building:2

Model Metrics vs. Probability Cutoff



Random Forest Confusion Matrix (Training Set)





Key Insights

- **Imbalanced Data Matters:** SMOTE significantly improved the model's ability to detect fraud.
- **Feature Selection Works:** RFECV helped reduce noise and improve interpretability and performance.
- **Temporal Patterns Detected:** Time-based features like incident_hour added valuable signal to the models.
- **Model Choice Impacts Recall:** Random Forest captured complex patterns better than Logistic Regression, especially for fraudulent cases.

Prediction and Model Evaluation:1

Logistic Regression Predictions

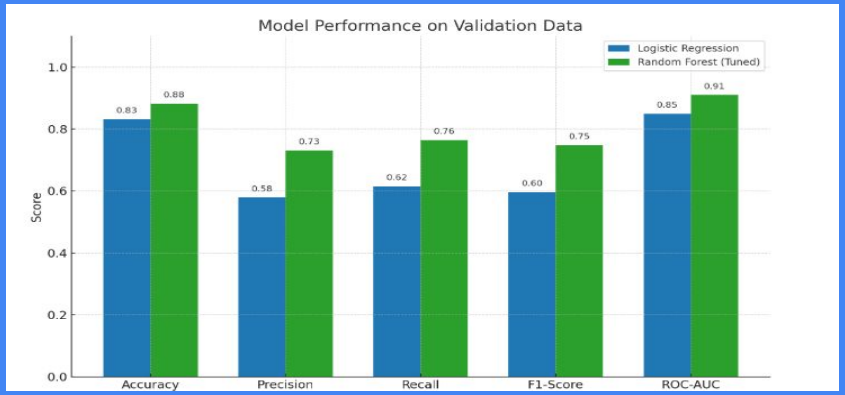
- Used selected features from RFECV for prediction.
- Produced probability-based predictions on validation data.
- Optimal cutoff determined via ROC curve to improve sensitivity/recall.
- **Key Observations:**
  - Moderate precision and recall.
  - Easy to interpret but may underperform in identifying complex fraud patterns

Random Forest Predictions

- Applied both the base and tuned Random Forest models on validation data.
- Captured non-linear feature interactions and handled imbalanced data more effectively.
- Tuned model demonstrated significantly better performance across all key metrics.

Model Comparison on Validation Data

Metric	Logistic Regression	Random Forest (Tuned)
Accuracy	0.8313	0.8825
Precision	0.5789	0.7308
Recall	0.6154	0.7647
F1-Score	0.5966	0.7473
ROC-AUC Score	0.8500	0.9100



Key Observations of the Case Study: Conclusion

Detecting Fraud Patterns in Historical Data

Fraudulent claims exhibit distinctive behaviors, including:

- Higher claim amounts across all types (injury, property, vehicle)
- Younger claimants and shorter customer tenure
- Increased umbrella coverage and more frequent witnesses

Most Predictive Features Identified

Features with high predictive power included:

- total_claim_amount, policy_annual_premium, incident_severity, and months_as_customer
- Temporal and categorical indicators like incident_hour and collision_type

Predictive Modeling Results

- Logistic Regression: Provided interpretability but limited in capturing complex fraud patterns
- Tuned Random Forest: Delivered superior accuracy (0.88) and recall (0.76), making it effective for real-world fraud detection
- Model insights support proactive assessment of fraud likelihood for new claims

Strategic Insights for Improving Fraud Detection

- Balancing class distributions (via SMOTE) significantly enhanced fraud identification
- Feature engineering played a crucial role in capturing hidden patterns
- The model can support real-time fraud scoring, enabling smarter claim triaging and reducing investigation time

Conclusion

By leveraging machine learning, Global Insure can now **predict fraudulent claims with high confidence** using historical data. This solution strengthens operational efficiency, minimizes losses, and builds a foundation for **scalable, data-driven fraud detection**.

Thank You!