Fraudulent Claim Detection

By. Naveen Upadhyay & Sunava Neogy

1. Introduction

1.1 Problem Statement

Global Insure faces significant financial losses due to fraudulent insurance claims. Their current manual inspection process is time-consuming and inefficient, often leading to late detection of fraud. The company aims to improve its fraud detection process using data-driven insights to identify fraudulent claims early in the approval process, minimizing financial losses and optimizing claims handling.

1.2 Business Objective

The primary objective is to build a model that classifies insurance claims as either fraudulent or legitimate based on historical claim details and customer profiles. This model will help predict the likelihood of fraud before claim approval, enabling proactive measures to prevent financial losses.

2. Data Understanding

2.1 Data Dictionary (Summary)

The insurance claims dataset contains information about customers, their policies, incident details, and claim amounts. It includes features like:

- **Customer Demographics:** Age, gender, education, occupation, etc.
- **Policy Details:** Policy number, bind date, state, premium, etc.
- **Incident Information:** Incident date, type, severity, location, etc.
- **Claim Amounts:** Total claim amount, injury claim, property claim, etc.
- **Target Variable:** fraud_reported (Y/N)

2.2 Data Cleaning

The following steps were taken to clean the data:

- **Handling Null Values:** Rows with missing values in the 'authorities_contacted' column were removed.
- **Redundant Values and Columns:** The '_c39' column, which was empty, was dropped. Rows with non-zero values for 'capital-loss' were removed, as it was deemed illogical for this scenario. The 'capital-loss' column was subsequently dropped.
- **Data Type Correction:** The 'insured_zip' column was converted to an object data type.
- **Encoding Categorical Variables:** Categorical variables like 'fraud_reported', 'property_damage', 'police_report_available', and 'insured_sex' were encoded using numerical values (1/0) for better model compatibility.

3. Exploratory Data Analysis

3.1 Univariate Analysis

- **Numerical Features:** Histograms and box plots were used to visualize the distribution of numerical features like 'policy_annual_premium', 'age', etc. This revealed the presence of outliers and the overall distribution patterns.
- **Categorical Features:** Bar charts were used to analyze the frequency of different categories within categorical features, helping identify potential imbalances or dominant categories.

3.2 Correlation Analysis

- A correlation matrix and heatmap were generated to visualize the relationships between numerical features. This analysis aimed to identify potential multicollinearity, which could affect model performance.

3.3 Class Balance

- A bar chart of the target variable ('fraud_reported') showed a class imbalance, with a significantly higher proportion of legitimate claims compared to fraudulent ones.

3.4 Bivariate Analysis

- **Categorical vs. Target:** Fraud likelihood was calculated for each category within relevant categorical features (e.g., incident type, policy state). This helped identify features with strong predictive power for fraud detection.
- **Numerical vs. Target:** Box plots were used to compare the distribution of numerical features across fraudulent and legitimate claims. This analysis revealed potential relationships between numerical features and the target variable.

---

4. Feature Engineering

4.1 Resampling

- **RandomOverSampler** was applied to address the class imbalance in the training data. This technique generated synthetic samples for the minority class (fraudulent claims) to balance the dataset and improve model performance.

4.2 Feature Creation

- New features were created from existing ones to enhance model performance. These included:
  - Date-based features: policy_bind_year, incident_year, incident_month, days_between_bind_and_incident
  - Ratios: premium_per_month
  - Interaction features: edu_occ (combining education and occupation)

  - Claim severity: claim_to_injury_ratio

4.3 Handling Redundant Columns

- Redundant or less informative columns were dropped based on correlation analysis, low variance, and feature importance. This included original date columns, ID-like columns, and highly correlated features.

4.4 Combining Values in Categorical Columns

- Rare categories in categorical features were grouped into an 'Other' category to reduce sparsity and improve model generalization.

4.5 Dummy Variable Creation

- Categorical features were transformed into numerical representations using one-hot encoding (dummy variables) to make them suitable for model training. Consistent encoding was ensured between training and validation data.

4.6 Feature Scaling

- Numerical features were scaled using StandardScaler to prevent features with larger values from dominating the model. This ensured that all features contributed equally during model training.

---

5. Model Building

5.1 Logistic Regression

- **Feature Selection (RFECV):** Recursive Feature Elimination with Cross-Validation (RFECV) was used to identify the most relevant features for the logistic regression model.

- **Model Building & Multicollinearity Assessment:** A logistic regression model was built using the selected features. P-values and Variance Inflation Factors (VIFs) were assessed to detect and address multicollinearity.

- **Model Training & Evaluation:** The model was trained on the training data, and its performance was evaluated using metrics like accuracy, precision, recall, and F1-score.

- **Optimal Cutoff:** The optimal probability threshold for classification was determined by analyzing the sensitivity-specificity trade-off and precision-recall trade-off.

- **Final Prediction & Evaluation:** Predictions were generated using the selected cutoff, and the model's performance was evaluated on the validation data.

5.2 Random Forest

- **Feature Importance:** The importance scores of each feature were obtained, and the top features were selected for model training.

- **Model Evaluation:** The random forest model was trained, and its performance was evaluated using various metrics on the training data.

- **Cross-Validation:** Cross-validation was performed to assess the model's generalization ability and prevent overfitting.

- **Hyperparameter Tuning (Grid Search):** Grid search was used to find the optimal hyperparameter values for the model, further improving its performance.
- **Final Model & Evaluation:** The final random forest model was trained with the best hyperparameters, and its performance was evaluated on the validation data.

---

6. Results & Evaluation

Logistic Regression:

Model Strengths: Easy to interpret and fast to train; provided a solid baseline.

Cutoff Optimization: Through ROC analysis and evaluation at different probability thresholds, we selected an optimal cutoff (~0.35) to balance sensitivity and specificity.

Performance:

Accuracy: Moderate

Sensitivity (Recall): Prioritized to catch fraudulent cases

Specificity: Acceptable, with room for improvement

F1 Score: Balanced measure indicating decent performance

Random Forest:

Model Strengths: Captures nonlinear relationships and interactions; robust against overfitting.

Feature Importance: Identified key variables contributing to fraud detection, such as:

incident_severity

insured_occupation

auto_model

insured_education_level

insured_hobbies

Hyperparameter Tuning: Grid search improved model performance further.

Cross-validation: Showed consistent results, indicating generalizability.

Performance:

Higher Accuracy than logistic regression

Improved Recall and Precision, meaning better identification of fraudulent claims with fewer false alarms

Best Overall F1 Score, confirming its superiority over the baseline model

---

7. Conclusions

By leveraging historical claim and customer data, we demonstrated that machine learning models, especially Random Forest, can significantly enhance Global Insure's ability to detect fraudulent claims early, thereby reducing financial losses and increasing operational efficiency. With further refinements and integration into the business workflow, this approach can form the foundation of a scalable, intelligent fraud detection system.

---

8. Recommendations

- Integrate the model into the claims approval workflow.
- Use model outputs to prioritize manual review of high-risk claims.
- Continuously retrain the model with updated data for improved accuracy.