Insurance Fraud Detection Project Report

Objective

The project aims to identify fraudulent claims in insurance data using advanced machine learning techniques. The goal is to:

1. Accurately detect fraudulent claims.
2. Develop a robust and scalable machine learning pipeline.
3. Prioritize minimizing false negatives to ensure that more fraudulent claims are captured.

Dataset Overview

The dataset used for the project consists of multiple features, which can be categorized as:

- **Target Variable**: FraudFound_P, indicating whether a claim was fraudulent (1) or not (0).
- **Feature Types**:
    - **Numerical Columns**: Includes variables like Age, Deductible, and Year.
    - **Categorical Columns**: Includes variables such as VehiclePrice, PolicyType, and Make.

Initial Exploration:

- Checked for missing values and cleaned the data.
- Irrelevant columns such as Month, PolicyNumber, and WeekOfMonth were dropped to reduce noise.

Data Preprocessing

Encoding & Transformation

1. **Label Encoding**: Boolean columns ('this or that' columns) were label-encoded.
2. **Feature Groups**:
    - **Ordinal Features**: BasePolicy.
    - **Nominal Features**: Variables like VehicleCategory and Days_Policy_Claim.
    - **Numerical Features**: Variables like Age, Deductible.

Balancing the Dataset

- **SMOTE (Synthetic Minority Oversampling Technique)** was applied to address class imbalance. This ensures the model can better identify fraudulent claims despite their minority representation in the dataset.

Feature Selection

- **Recursive Feature Elimination with Cross-Validation (RFECV)** was performed using a LightGBM classifier to select the most important features.
- Selected features were used to create refined training and test datasets.

Modeling Techniques

Algorithms Explored

1. **RandomForestClassifier**
2. **GradientBoostingClassifier**
3. **BalancedRandomForestClassifier**
4. **SVC (Support Vector Classifier)**
5. **LGBMClassifier (LightGBM)**

Hyperparameter Tuning

- Optuna was employed for hyperparameter tuning.
- The process involved running 50 trials to optimize parameters such as n_estimators, learning_rate, and num_leaves.

Pipeline Structure

The pipeline consisted of:

1. Preprocessing steps (encoding, scaling).
2. Oversampling using SMOTE.
3. Feature selection via RFECV.
4. Model training with hyperparameter tuning.

Results

Performance Metrics

The best-performing model achieved the following:

- **Accuracy**: 74.60%
- **Precision**: 13.91%
- **Recall**: 60.34%

- **F1 Score**: 22.61%
- **ROC AUC Score**: 76.07%
-

Key Insights

1. The high recall (60.34%) indicates the model's ability to capture a significant portion of fraudulent claims.
2. Precision (13.91%) suggests that many flagged claims are false positives. However, in fraud detection, false positives are less critical than false negatives.
3. The **ROC AUC score (76.07%)** demonstrates the model's robustness in distinguishing between fraudulent and non-fraudulent claims.

Prioritizing Fraud Detection

Focus on False Negatives

- **Why?**
    - A false negative (failing to identify a fraudulent claim) can result in significant financial losses.
    - The goal is to minimize the number of fraudulent claims that go undetected.

Strategies to Prioritize Fraud Detection

1. **Lowering the Decision Threshold**:
    - Adjusting the classification threshold (e.g., from 0.5 to 0.098 as done in this project) increases recall by flagging more claims as potentially fraudulent.
2. **Cost-Sensitive Learning**:
    - Assign higher penalties to false negatives during model training.
3. **Ensemble Methods**:
    - Combining multiple models to improve prediction stability and reduce the chance of missing fraud cases.
4. **Explainable AI (XAI)**:
    - Use tools like SHAP or LIME to understand feature contributions, enabling targeted interventions.

Conclusions and Recommendations

1. **Model Effectiveness**:
    - The pipeline effectively detected fraudulent claims with high recall and a robust ROC AUC score.
2. **Trade-Offs**:
    - Precision was sacrificed to achieve higher recall, aligning with the business need to prioritize fraud detection over false alarms.

Future Directions:

1. **Enhance Feature Engineering**:
    - Introduce domain-specific features, such as geographic patterns or external data (e.g., weather, demographics).
2. **Explore Advanced Models**:
    - Implement deep learning models like neural networks or advanced tree-based ensembles like CatBoost.
3. **Real-Time Fraud Detection**:
    - Deploy the model in production for real-time claim assessments.
4. **Periodic Retraining**:
    - Update the model periodically with new data to adapt to evolving fraud patterns.

By focusing on minimizing false negatives, the organization can mitigate financial losses while continuously refining the model for better performance.
