FindDefault - Credit Card Fraud Prediction

Name: Anirban Chakraborti

Problem Statement

Credit cards are widely used for online transactions, offering convenience but also exposing users to the risk of fraud. Credit card fraud refers to the unauthorized use of someone else's credit card details to make purchases or withdrawals. Identifying fraudulent transactions effectively is crucial for credit card companies to prevent customers from being charged for unauthorized purchases. This project focuses on building a classification model to predict whether a transaction is fraudulent based on a dataset containing credit card transactions from European cardholders in September 2013. The dataset consists of 284,807 transactions, of which only 492 are fraudulent, making it highly imbalanced. The fraudulent transactions account for just 0.17% of the total transactions.

Data Overview

The dataset contains 284,807 rows and 31 columns. To protect user privacy, the original numerical features were transformed using Principal Component Analysis (PCA) into 28 principal components, in addition to the columns **Time**, **Amount**, and **Class**. The **Class** column is the target variable, with values indicating fraudulent (1) or non-fraudulent (0) transactions.

Data Cleaning

Upon inspecting the data for null and duplicate values, we found no null values. However, there were some duplicate entries, which were removed prior to proceeding with the analysis.

Exploratory Data Analysis (EDA)

We performed an exploration of the dataset to understand the distribution of fraudulent and non-fraudulent transactions. A graph was plotted to visualize this distribution, revealing that fraudulent transactions make up only 0.17% of the total dataset. We also analyzed the **Time** and **Amount** columns by plotting graphs to determine transaction timings and identify any outliers in the **Amount** column.

Feature Engineering

Since the **Time** and **Amount** columns were deemed unnecessary for modeling, they were removed. A new scaled column was introduced to represent the scaled **Amount** feature. The data was then split into two variables:

- **X**: The feature matrix containing the PCA components and the scaled **Amount**.
- **Y**: The target variable, representing the **Class**.

Model Training

We split the dataset into training and testing sets using a 70-30 ratio with the **train_test_split()** function. The parameters were as follows:

- **X**: Feature matrix
- **Y**: Target variable
- **test_size**: 0.3 (30% of the data allocated to the test set)

Model Selection

We selected two machine learning algorithms for model building:

- **Decision Tree**: A supervised learning algorithm that splits the data into subsets based on feature values, forming a tree-like structure of decisions.
- **Random Forest**: An ensemble learning method that combines multiple decision trees to improve accuracy and reduce overfitting.

Model Validation

Both models were validated and tested using various evaluation metrics:

- **Accuracy Score**
- **Precision Score**
- **Recall Score**
- **F1 Score**
- **Confusion Matrix**

Heatmaps were generated for the confusion matrices of both models to visually assess their performance.

Dealing with Imbalanced Data

Due to the highly imbalanced nature of the dataset, where fraudulent transactions make up only 0.17% of the total, we applied the **SMOTE (Synthetic Minority Oversampling Technique)**. SMOTE is an oversampling method that generates synthetic examples of the minority class to balance the class distribution.

After balancing the data, we trained the **Random Forest** model on the resampled dataset, as it showed better performance than the **Decision Tree**. We then evaluated the model using the same performance metrics (accuracy, precision, recall, F1 score, confusion matrix) and generated heatmaps for the confusion matrices.

Model Deployment

For model deployment, we plan to use the **pickle** library to save both the trained model and the dataframe for future use. This will allow the model to be easily integrated into production environments where predictions can be made on new, unseen data.

Conclusion

The model demonstrated improved performance after addressing the class imbalance using SMOTE and training with the **Random Forest** algorithm. With an accuracy exceeding 99%, the model shows strong potential for identifying fraudulent transactions in real-world applications. Moving forward, we aim to deploy the model and continuously monitor its performance to ensure its effectiveness.