Fraud Transaction Detection Project Overview 🚀

In this project, I aimed to build an efficient model to detect fraudulent transactions using a dataset of credit card transactions. Below are the detailed steps I undertook during the project:

1. Data Loading 📥

I began by loading the dataset using **Pandas**, a powerful data manipulation library in Python. The dataset comprised 284,807 transactions with 31 features, including transaction amount and various anonymised features (V1 to V28). I used data.info() and data.describe() to gain insights into the structure and summary statistics of the data.

2. Data Preprocessing 🖌️

In this step, I ensured the 'Class' column, indicating fraud (1) or not fraud (0), was in integer format. I checked for any missing values using data.isnull().sum() and found none. I then separated the features from the target variable, 'Class', and split the data into training (80%) and testing (20%) sets using **train_test_split** from Scikit-learn.

3. Feature Scaling 🔄

To ensure that all features contributed equally to the model, I employed **StandardScaler** for feature scaling. This transformed the features to have a mean of 0 and a standard deviation of 1, which is crucial for the performance of many machine learning algorithms.

4. Handling Class Imbalance ⚖️

Given the highly imbalanced nature of the dataset, with significantly fewer fraudulent transactions, I applied **SMOTE (Synthetic Minority Over-sampling Technique)** to balance the classes. This technique generated synthetic samples to enrich the minority class, ensuring a more robust model.

5. Model Training with Random Forest and GridSearchCV 🏋️

I chose to use the **Random Forest Classifier** for its ability to handle large datasets and its effectiveness in classification tasks. I created a pipeline to streamline the training process. After training the model, I confirmed that the process was complete.

6. Model Evaluation 📈

To assess the model's performance, I generated predictions on the test set and evaluated the model using a confusion matrix and a classification report. The results were impressive:

- **Precision for Class 0 (Not Fraud): 100%**
- **Precision for Class 1 (Fraud): 91.67%**
- Overall accuracy was around 99.83%, indicating a strong model performance with minimal false positives.

7. Visualization 📊

To further interpret the model's performance, I visualised key metrics:

- **Feature Importance**: I plotted the importance of each feature, revealing which attributes contributed most to the model's predictions.
- **ROC Curve**: The Receiver Operating Characteristic curve indicated a high area under the curve (AUC = 0.99), demonstrating excellent model discrimination capability.
- **Precision-Recall Curve**: This plot helped visualise the trade-off between precision and recall, reinforcing the model's ability to detect fraudulent transactions accurately.

Skills Developed 🛠️

Throughout this project, I enhanced my skills in:

- Data Preprocessing,
- Feature Engineering,
- Handling Class Imbalance,
- Model Evaluation,
- Data Visualisation.

Conclusion 🎯

This project not only deepened my understanding of fraud detection mechanisms but also equipped me with practical skills in applying machine learning techniques effectively.

Hashtags

#DataScience #MachineLearning #FraudDetection #DataAnalysis #RandomForest #Python #Pandas #SMOTE #FeatureScaling #ModelEvaluation #DataVisualisation #AI #DeepLearning #BigData #DataMining #Classification #Statistics #Analytics #BusinessIntelligence #DataScientist #DataEngineering