FindDefault: Predicting Credit Card Fraud Documentation

Problem Statement

Credit card fraud poses a significant threat to the financial security of cardholders and can result in substantial losses for credit card companies. FindDefault addresses this problem by leveraging machine learning techniques to identify fraudulent transactions in real-time.

*Exploratory Data Analysis (EDA***)**

- Libraries Used: Pandas, Matplotlib, Seaborn
- We use descriptive statistics, visualizations like heatmap, and data quality checks to identify patterns, relationships, and trends within the data.

Data Cleaning

- Libraries Used: Pandas
- Data cleaning involves standardizing the format, handling missing values, and addressing outliers to ensure the dataset is accurate and consistent.
- By cleaning the data, we prepare it for further analysis and model training.

Dealing with Imbalanced Data

- Libraries Used: Scikit-learn
- The credit card transaction dataset is highly imbalanced, with fraudulent transactions being a minority class.
- We use techniques such as undersampling

Feature Engineering

- Libraries Used: Scikit-learn (StandardScaler)
- Feature engineering involves creating new features or transforming existing ones to improve the predictive performance of machine learning models.
- By selecting relevant features and engineering them appropriately, we aim to capture meaningful patterns in the data.

Model Selection

- Libraries Used: Scikit-learn
- We evaluate various machine learning algorithms, such as Decision Tree, Logistic Regression, and Random Forest.

Model Training

- Libraries Used: Scikit-learn

- The dataset is divided into training and testing sets using train_test_split() function.
- We train the selected model using the training data to learn patterns and relationships within the data.

Model Evaluation

- Libraries Used: Scikit-learn
- We evaluate the trained model's performance using metrics such as accuracy, precision, recall and F1-score,
- These metrics help us assess the model's ability to correctly classify fraudulent and non-fraudulent transactions.

Success Metrics:

- The accuracy of the model on the test data set should be &gt; 75% (Subjective in nature)

  Our accuracy score is 94%

- Add methods for Hyperparameter tuning

Hyperparameter Tuning:

- Methods Used: GridSearchCV
- Hyperparameters of machine learning models are tuned using techniques like GridSearchCV.
- GridSearchCV exhaustively searches through a specified hyperparameter space to find the optimal combination of hyperparameters that maximizes model performance.

Pipeline:

- This Python script demonstrates the utilization of a machine learning pipeline for credit card fraud detection.
- The pipeline also includes a trained Random Forest Classifier model.
- The purpose of the pipeline is to predict fraudulent transactions.

Purpose:

- The purpose of this script is to automate the process of loading data, preprocessing features, and making predictions using a pre-trained machine learning model.
- By encapsulating these steps within a pipeline, it ensures consistency and reproducibility.
- Additionally, it simplifies the deployment of the fraud detection model.

Conclusion:

- This script demonstrates a streamlined approach to deploy a pre-trained machine learning model for credit card fraud detection. By encapsulating preprocessing and prediction steps within a pipeline, it ensures efficient and consistent model deployment, facilitating real-time fraud detection in financial transactions

- ➢ Perform model validation

Model Validation:

- Model validation is performed to ensure the model's ability to generalize to new, unseen data.
- Technique we use is Train-Test Split
- By evaluating the model on different subsets of the data, we can identify any potential issues such as overfitting or underfitting.
- Validation metrics such as accuracy, precision, recall and F1-score are used to measure the performance of the model on unseen data.