### **Fraudulent Transaction Detection Using Machine Learning Documentation**

#### **1. Importing the Data**

* The first step involved importing the dataset into the environment for analysis. The data likely included various features that describe financial transactions, with one column indicating whether the transaction is fraudulent (target variable).

#### **2. Rearranging the Columns**

* Columns were rearranged to clearly distinguish between the feature variables (X) and the target variable (Y). This organization helps streamline the modeling process.

#### **3. Creating a Copy of the DataFrame**

* A copy of the original DataFrame was created for preprocessing purposes. This ensures that any modifications during preprocessing do not affect the original dataset, preserving data integrity.

#### **4. Dropping Columns (Feature Selection)**

* Several columns were dropped based on feature selection criteria:
  + **"Step"**: This column represented a simple numerical series that would not contribute to the model's performance or predictive power.
  + **"nameOrig" and "nameDest"**: These columns contained unique identifiers (e.g., IDs) which would not be relevant for model training as they do not provide useful patterns or trends for prediction.

#### **5. Exploratory Data Analysis (EDA)**

* EDA was conducted to explore the dataset, identify patterns, and understand the distribution of data. This likely included visualizations like histograms, bar charts, and correlation matrices to identify relationships between variables.

#### **6. Data Preprocessing**

* Data preprocessing steps included:
  + Handling missing values.
  + Encoding categorical variables.
  + Scaling numerical features to normalize the data for better model performance.

#### **7. Balancing the Data with SMOTE**

* The dataset was imbalanced, with a significantly lower number of fraudulent transactions compared to non-fraudulent ones. To address this, the **Synthetic Minority Over-sampling Technique (SMOTE)** was applied to balance the data by generating synthetic samples for the minority class. This step is crucial as it improves the performance of the model, particularly in detecting fraudulent transactions.

#### **8. Model Selection and Training**

* Several machine learning models were trained and evaluated on the dataset:
  + **Logistic Regression**: A baseline model used for binary classification.
  + **Random Forest Classifier**: A robust ensemble method that uses multiple decision trees to make predictions.
  + **Gradient Boosting Classifier**: Another ensemble method that builds models sequentially, with each model correcting the errors of the previous one.
  + **Decision Tree Classifier**: A model that splits the data based on feature values to make predictions.
* Among these models, the **Decision Tree Classifier** with balanced data (using SMOTE) performed the best. The success of this model was attributed to its ability to handle complex, non-linear relationships in the data.

#### **9. Model Evaluation**

* The models were evaluated using key metrics: **Precision**, **Recall**, and **F1-score**. The **Decision Tree Classifier** with SMOTE-balanced data produced the best results:
  + **Class 0 (Non-Fraudulent Transactions)**:
    - Precision: **1.00**
    - Recall: **1.00**
    - F1-Score: **1.00**
    - Support: **828,730**
  + **Class 1 (Fraudulent Transactions)**:
    - Precision: **0.61**
    - Recall: **0.97**
    - F1-Score: **0.75**
    - Support: **2,375**
* The high recall of **0.97** for fraudulent transactions indicates that the model was very effective in identifying most fraudulent activities, which is crucial in fraud detection.

#### **10. Conclusion**

* The project effectively implemented a fraud detection model using machine learning techniques, with the Decision Tree Classifier trained on SMOTE-balanced data emerging as the most successful model. The evaluation metrics, particularly the recall and F1-score, demonstrate the model's strong performance in identifying fraudulent transactions.