Describe your fraud detection model in elaboration.

The fraud detection model is designed to accurately identify fraudulent transactions while minimizing false positives and maintaining efficiency in processing large volumes of data. Here's an elaboration of the model:

Data Preprocessing:

1. Impoting Dependencies:

The model begins by importing essential Python packages, modules, and libraries required for subsequent operations.

2. Data Analysis :

A thorough analysis of the dataset is conducted, including an overview of its size, shape, data types, missing values, and duplicate entries. Insights into feature distributions and transaction types are gleaned through exploratory data analysis (EDA).

3. Missing Value Handling :

No missing values are observed in the dataset, ensuring data completeness.

4. Duplicate values :

Duplicate Values: Duplicate values are absent in the dataset, ensuring data integrity.

5. Exploratory Data Analysis:

Identifying Unique Categories in Each column. Counting Total number of unique values in each column

Percentages of the types of transaction in type column features which is the important factor in transactions.

Exploratory Data Analysis (EDA):

6. Identifying Unique Categories:

The model identifies unique categories within each column and calculates the total number of unique values in each feature.

7. Transaction Type Distribution:

A detailed examination of transaction types and their percentages provides insights into the dataset's composition.

8. Data Visualization:

Visualization techniques, such as value counts and histograms, are employed to gain further insights into feature categories and distributions.

Data Cleaning:

9. Removing Unnecessary Columns :

Irrelevant columns are removed to streamline the dataset.

10. Correlation Matrix :

Checking Feature Importance and relation using correlation Matrix Taking Relevent Datapoints Raws from observations

11. Handling Multicollinearity:

Address multicollinearity by performing Correlation Matrix techniques identify redundant features.

12. Feature Engineering:

Extract relevant features from the dataset, such as transaction amounts, timestamps, account balances, Originator account type, destination account type. and transaction types. Create additional features, such as error_balance_orig and error_balance_dest, to capture inconsistencies in account balances before and after transactions.

Model Selection:

Algorithm Selection: Robust algorithms, such as decision trees, random forests, and XGBoost classifiers, are chosen for their resilience to outliers and class imbalance.

Logistic Regression :

The logistic Regression model not perform well it is biased along with positive class can predict non_fraud class accurately because there is some issue with out data i.e. imbalanced dataset

Tree-based algorithms (e.g., Decision Tree, Random Forest, Gradient Boosting): These algorithms handle non-linear relationships and are robust to outliers.

Decision tree perform well on the data with highest accuracy with precision and f1 score 0.99 for both

Xgb classifier performs best on the model like decision tree with high accuracy precision recall

So we select the XGB classifier for our model training testing and evaluation and our work of detection of fraud.

For we that we perform hyperparameter tuning with fewer parameter because of the computational capacity is no sufficient for the size of the data

Neural Networks: I can't use this techniques because of the computional resources limitations for the large size of data almost 650K rows datapoints

Model Training:

Training Process: The chosen algorithms are trained on the training dataset while adjusting hyperparameters to achieve optimal performance.

Evaluate model performance using appropriate metrics such as precision, recall, F1-score, and area under the receiver operating characteristic (ROC) curve.

Model Evaluation and Validation:

Hyperparameter Tuning: The selected algorithm XGBoost Classifier is fine-tuned using GridSearchCV and cross-validation techniques to optimize performance and reduce Overfitting.

Performance Evaluation: The model's effectiveness is evaluated using metrics such as precision, recall, F1-score, and area under the ROC curve.

Validation: The model's performance is validated on a separate Testing data to ensure its ability to generalize to unseen data.

Another Approach:

"We opted to address the class imbalance issue by employing random under sampling, a technique where the majority class instances are randomly subsampled to match the number of instances in the minority class. Following this preprocessing step, we applied the Random Forest Classifier algorithm to the balanced dataset. This approach yielded promising results, demonstrating high accuracy in our classification task."

Conclusion:

Overall Assessment: The fraud detection model represents a comprehensive system equipped with advanced techniques in data preprocessing, model selection, training, evaluation, and validation.

Effective Fraud Detection: The model demonstrates high accuracy in identifying fraudulent transactions while minimizing false positives and maintaining operational efficiency.

By following this structured approach, the fraud detection model is poised to effectively combat fraudulent activities while optimizing performance and resource utilization.