# Step 1: Import Libraries and Necessary Installation

Load the necessary libraries and do required installation.

Text, letter

Description automatically generated

**Step 2: Read file.**

Load the transaction and identity datasets. Check the wall time. Checked memory usage using the info method in Pandas.

**Step 3: Merging both the datasets**

Merged datasets transaction (train\_tr) and identity (train\_id) into one dataset so that the analysis and related data handling activities can be performed on one dataset. Left outer join was done as every transaction id in the transaction dataset may not have equivalent record in the identity dataset and any transaction records should not be missed.

**Step 4: Exploratory Data Analysis (EDA)**

The shape, feature size and statistics for the merged dataset were analyzed. The percentage of Fraud (only 3.5%) vs Normal transactions in the dataset helped to understand the imbalance in the dataset. The statistics for Normal transactions in the Dataset and the statistics for Fraud transactions in the Dataset shows that a fraud transaction cannot be identified only by amount. Duplicates and outliers were checked. Numerical and categorical variables were separated out so that they can be processed accordingly during Feature Engineering. Correlation was found between the attributes using Correlation Matrix. V1 to V11 has high correlation with D6 to D14 as well as with other V attributes. The C\* (i.e., C1, C2, ...) attributes are more correlated within themselves with respect to D\* (i.e., D1, D2, ...) attributes internal correlation. Also C\* attributes are not much correlated with D\* attributes. V107 is highly correlated with most of the other V\* attributes. dst1 is highly correlated with a large number of V\* columns.

**Step 7: Feature Engineering**

As part of feature engineering, duplicates and Null values were checked. There were no duplicates to handle. NaN values were filled with zeroes as there are many columns having Null value, droping the columns with Null values will delete a lot of data where other important non-NaN attributes will be missing for the same recordset. Most of the ML algorithms perform better and produce better accuracy with numerical variables. Hence, Label Encoding was done to convert categorical variables into numerical variables. Chi square test does not apply to negative values. That’s why any negative values were checked and replaced with zeroes.

**Step 8: Feature Selection**

In Feature selection **the number of input variables were reduced and selected features of high importance.** Chi square Test was used for feature selection. First, the target (isFraud) variable is separated as a separate dataframe (Y\_data). Copied the original dataset excluding the target variable into a separate dataframe(df\_input). This helped to keep the original dataset intact, so that it can be referred whenever required. Then the SelectKBest algorithm was used with Chi square function to score the attributes using df\_input and Y\_data. Based on descending order of the scores, the best 150 features were selected to be utilized for next step of model training. The Y\_data (the target isFraud variable) was merged to these 150 features to prepare the dataset (df\_selected\_features\_io) which will be subjected to Train-test Split.

**Step 9: Split the dataset into train and test.**

The merged dataset (df\_selected\_features\_io) containing 151 columns was split into train and test, i.e 80% of the dataset is used for training the model and 20% of the dataset is used for testing the model. Then the output (Y) variable (‘isFraud’) is separated from input (X) variable for both train and test datasets.

**Step 10: Model Selection**

The Accuracy and F1 scores for some ML models were checked and selected the model with highest Accuracy and F1 score.

1. Simple Random Forest Model

Roc Auc Score: 62%

F1 score: 0.37

1. Decision Tree

Roc AUC Score: 97%

F1 Score: 0.22

1. K-Nearest Neighbors

Roc AUC Score: 96%

F1 Score: 0.04

1. Logistic Regression

Roc AUC Score: 96%

F1 Score: 0.002

1. Random Forest

Roc AUC Score: 97%

F1 Score: 0.27

1. XGBoost

Roc AUC Score: 98%

F1 Score: 0.68

The F1 score was an important differentiator in this model selection, because F1 score being harmonic mean of Precision and Recall, handles the imbalance of the dataset while calculating the metric. This metrics shows that XGBoost is having a high F1-score along with greater accuracy with respect to other models.

**Step 11: Balancing the unbalanced data**

The imbalanced Train dataset was balanced using SMOTE. In the original dataset, it was noticed that it contains only 3.5% fraudulent transactions which represents the minority class. Using SMOTE data augmentation methodology, helps to synthesize examples of the minority class without adding any new information.

**Step 12: Model Training after balancing the data**

After applying the SMOTE, the XGBoost model was re-executed and captured the performance metrics. It was noticed that the Accuracy score of the XGBoost model remained 98% and the F1 score of the XGBoost model is 0.6684412895593019, which is still the highest among the other

Models tested for the purpose. The ROC Curve shows that the model has higher chance of predicting the probability that an online transaction is fraudulent based on the AUC score of 0.92 which is much higher compared to no-skill line of 0.5.