IEEE- CIS Fraud Detection

In this project we have classified whether an online transaction is fraudulent or not. It is known as binary target, named: isFraud. The first step is to download the dataset, we will be using train dataset, which has two sections, “Train_transaction” and “Train_identity”. Further, we have merged these two DataFrames on a column named “TransactionID”. But not all transactions have corresponding identity information.

FEATURE SELECTION:

❖ STEP 1: Reading Train-Dataset and Displaying rows and columns present in it: -

The pandas read_csv() function is used to read a CSV file into a dataframe. Here we are reading "train_transaction.csv"file and "train_identity.csv" file into a defined function df_train_transaction and df_train_identity.

Furthermore, we have displayed maximum number of rows, display.max_rows represent the maximum number of rows that pandas will display while displaying a data frame. The default value of max_rows is 10. If set to ‘None’ then it means all rows of the data frame.

Similarly, we have displayed maximum number of rows which Ranges all columns of df such that the minimum value in each column is 0 and max is 1.

❖ STEP 2: Merging both Transaction and Identity Training files and Exporting them: -

Merging of both identity data and transaction data will be done through TransactionID column because it is common in both the files. For safety purpose, we are exporting our merged file in our directory.

STEP 3: Deleting ROWS and COLUMNS where “NAN” values are more than 75%

We are selecting only transaction dataset for filling null values as identity dataset contains maximum null but it also contains essential details which are used to predict fraud detection like "Devicetype", "DeviceInfo" etc; so not dropping identity table but, only selecting transaction dataset.

Deleting rows: We are removing the rows because it contains negligible value to the dataset. Start by checking the initial shape of the dataset, further row by row we start calculating null values with the help of “shape” and dropping the rows where null values are more than assigned percentage, at the end checking the final shape of the dataset. We are removing the rows because it contains negligible value to the dataset.

Deleting columns: We are removing the columns because it contains negligible value to the dataset. Similarly, we start it by checking the initial shape of the dataset, then we have selected only train_transaction dataframe with “loc” function, further columns by columns we calculate the null values with the help of “shape” and dropping the columns where null values are more than assigned percentage, further we are merging identity dataset on new transaction dataset where null values are removed, and at the last we are checking the final shape of the dataset. We are removing the rows because it contains negligible value to the dataset.

- EDA (Exploratory Data Analysis): This step is used to modify and understand the given data set by analyzing various columns and rows by applying Pandas and Data visualization packages.

❖ Filling up the “NAN/NULL” values with ‘mode’, ‘-555’, ‘not Found’ and ‘Found’:

Now we start replacing all the null values with some value which is -555.We have put -555 values as it displays. The Pandas “fillna” method fills in missing values in Pandas dataframes. In 'card2','card3','card5','addr1', 'adrr2', 'dist1', further from ‘D1 to D15’, from ‘V1 to id-11’ and from ‘id-13 to id-22” we have filled "-555" for null/nan numeric values, further more for objects like 'card4', 'card6’, ‘M1 to M9’, ‘id-35 to id-38’, ‘DeviceInfo’ and ‘DeviceType’ we have filled it with “mode” values. Hence further, since id-28 has three values NaN, NotFound and Found which we are replacing it with appropriate string value, where “NAN” values are replaced by “NotFound” and, “New” is replaced with “Found”, Whenever a NULL is replaced by found, the next row that contains a NOT NULL value will be found and that value is used to update the row with the NULL value. Similarly, id_15 has 4 distinct values as NaN, Unknown, New, Found and replacing those values with suitable notation, “NAN” and “UNKNOWN” is replaced by “NotFound” and “New” is replaced by “Found”.

❖ Dropping “id-23 to “id-27” and ‘id-30 to id-34’ - these features are dropped as they have more than 85%:

Missing Values in the dataframe

```
In [28]: (np.sum(pd.isnull(df_train_new)).sort_values(ascending=False)/len(df_train_new))*100

Out[28]: id_33          87.589494
         id_30          86.865411
         id_32          86.861855
         id_34          86.824771
         DeviceInfo     79.905510
         id_31          76.245132
         DeviceType     76.155722
         id_36          76.126088
         id_35          76.126088
         id_37          76.126088
         id_38          76.126088
```

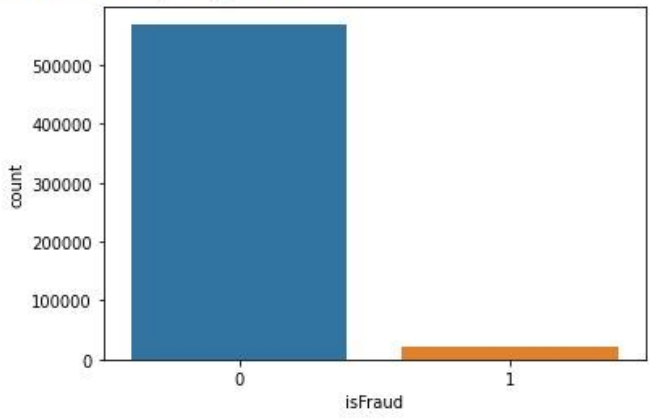
❖ Replacing all Objects: The entire data set where we observed T (TRUE) AND F(FALSE), we are replacing with 1 and 0, which are called "Binary Targets".
The data set where we observe terms Found and Not Found are replaced with 1 and 0 respectively, which are called "Binary Targets".
Replacing objects with binary targets will help us to increase our dataset accuracy.

❖ Selecting all the objects and checking unique values:
Checking unique values from the objects in the data set. There are 5 objects in the column ProductCD, so we are using the **Encoder method** if the data frame has more than 3 distinct object values then we are using an encoder otherwise we are assigning 0,1,2 in the data frame.

TRAIN-TEST-SPLIT: - Importing from skearn train-test-split, assigning all Features to X except isFraud, as isFraud is our target for fraud detection, as y we are selecting isFraud feature, splitting the dataframe with test size of 20% with fixed random_state value on X and y.

```
sns.countplot(x='isFraud', data=df_train)
df_train['isFraud'].value_counts()
```

```
0    569830
1     20663
Name: isFraud, dtype: int64
```



Balancing the dataframe with imblearn: -

Importing from Imblearn software, used to balance the dataset. In the dataframe 'isFraud' is Unbalanced so, using 'SMOTE' algorithm to make the dataset evenly distributed based on binary target 0 and 1, resampling the dataset same as 1 for 0 with same random state 31

Best Model Selection procedure using scikit-learn

We have used three models:

1) Logistic Regression: importing from sklearn model selection classifier with 'SMOTE' algorithm, where we are getting 81.3% accuracy with accuracy_score and 68.08% accuracy with roc_auc_score on the 100000 iterations, 'lbfgs' optimizer to predict with quasi-newton method, same random values are generated with random_state taken as 31.

2) MLPclassifier: importing from sklearn neural network classifier with 'SMOTE' algorithm there we are getting 81.81% accuracy with accuracy_score and 73% accuracy with roc_auc_score on the 100000 iterations and with same random state 31.

3) Random forest: importing from sklearn random forest classifier applying 'SMOTE' algorithm fitting X and y train with selecting tree size 2 and with same random state as 31 here we are getting the accuracy of 87.32% with accuracy_score and 69.28 with roc_auc_score.

CONCLUSION: - The best accuracy we are getting from Random Forest Classifier is **87.32%** on accuracy_score matrix.