Credit Card Fraud Detection

Name: INDRAJIT BURMAN

Problem Statement:

Credit cards are used widely for online payments and purchases things, it is providing convenience in managing personal finances. Nevertheless, it comes with risks, especially the threat of credit card fraud, it involves unauthorized use of a card, or card information for cash withdrawals and transactions. Hence, it is important for credit card companies to detect fraudulent transactions to protect their consumers from being unfairly charged for purchases which they did not buy.

In this context, I am analyzing a dataset containing credit card transactions made by European cardholders over two days in September 2013. This dataset includes 284,807 transactions, among which 492 are fraudulent. This makes the data highly imbalanced, fraudulent transactions constitute only 0.172% of the total.

My goal with this project is to develop a classification machine learning model to predict whether a transaction is fraud or a legit one.

APPROACH:

Exploratory Data Analysis (EDA) & Data Preprocessing and Cleaning:

In this highly imbalance dataset, I can see there is a target variable which is 'class' column and that indicates the 0 or 1 mean normal transaction or fraud transactions. Fraud transactions are 0.172% of the total.

1. I check if there are any missing values, I found there are no missing values.
2. Check the duplicated values, I found there are 1081 duplicate values, so I drop them.
3. Checking the datatype info of the columns.
4. Perform statistical analysis.
5. After dropping the duplicates values, the value count of class are 283253 and 473.

CLASS	0	1
COUNT	283253	473

6. The AMOUNT column is scaled by standard scaler.
7. I found there are wrong datatype in the Time column, so I change time column data type with datetime datatype.
8. Then I check the correlation, show it with a heatmap for outliers.
9. Next, I visualize the count plot of the class column using bar plot and also with the pie plot.
10. I drop the time column, because it was unnecessary column.

Dealing With Imbalanced Data:

I used SMOTE method to balance the data.

SMOTE (Synthetic Minority Over- Sampling Technique) is a over- sampling technique which create synthetic examples to oversample the minority class. In this dataset fraud transaction are very less hence it create synthetic examples to balance with normal transactions.

Model Selection and Model Building:

I start to build the model with train_test_split. I used 80% of the data for training purpose and 20% of the data for testing purpose. (80 : 20 ratio). I had to find out which machine learning model is best for the given data. Hence, I apply several machine learning models.

1. Logistic Regression: It is used for binary classification where the aim is to predict the discrete outcome 0 or 1. In my logistic regression model the accuracy, precision, recall, F1 scores are as follows:

   Accuracy: 0.944396391943655
   Precision: 0.9459755567325869
   Recall: 0.944396391943655
   F1 score:0.9443529341915856

2. Decision Tree: It is intuitive and versatile machine learning model which breaks down data by making decisions based on feature values and create a tree like structure. In my decision tree model the accuracy, precision, recall, F1 scores are as follows:

   Accuracy: 0.9980582867027943
   Precision: 0.9980594785165191
   Recall: 0.9980582867027943
   F1 score: 0.9980582805266436

3. Random Forest: It is an ensemble learning method, which builds multiple decision trees and merges their result to improve the predictive accuracy and control overfitting. In my random forest model, the accuracy, precision, recall, F1 score as follows:

   Accuracy: 0.9997616988226157
   Precision: 0.9997617547734947
   Recall: 0.9997616988226157
   F1 score: 0.9997616986832977

4. XG Boost: Extreme Gradient Boosting is an advanced and powerful implementation of gradient boosting algorithms designed for speed and performance. XG Boost builds an ensemble of decision trees sequentially, where each new tree attempts to correct the errors of the previous trees. In my XG Boost model the accuracy, precision, Recall, F1 scores are as follows:

   Accuracy: 0.9997352209140175
   Precision: 0.9997353605924464
   Recall: 0.9997352209140175
   F1 score: 0.9997352206627931

Check the Confusion Matrix and ROC AUC: Next, I check the confusion matrix and ROC AUC of the following models, hence I can see that the *XG Boost model is performing best among all models*.

Then I create an empty data frame to store the scores of various algorithms.

XG Boost model performing best.

Hyperparameter Tuning: Hyperparameter tuning is the process of optimizing the parameters that control the learning process of a machine learning model, as opposed to model parameters that are learned from the data during training. Hyperparameters are crucial for the performance and effectiveness of models like XG Boost. The goal is to find the best set of hyperparameters to maximize the model's predictive performance on unseen data.

In my XG Boost model I use Grid Search hyperparameter tuning method. Then I show the classification report.

In my logistic Regression model, I use Random search hyperparameter tuning method. Then I show the classification report.

Cross- Validation: In my analysis XG Boost model gives the best results. Hence, I am choosing XG Boost model for cross validation. I use *stratified K Fold cross validator* to check the model's mean ROC AUC score.

Model Deployment Plan:

- I select XG Boost model with Grid Search for deploying, because it gives the best predictive results.
- I save the model using joblib library and name it Credit_card_model

For checking the model, I load the credit_card_model and check the model's prediction on the actual data which was given. And write a code for 0 means Normal Transaction and 1 means Fraud Transaction.

Hence, I found that model is performing very well.

Cost- Benefit Analysis:

In this case we have to account for what we need: high precision or high recall.

For bank with smaller average transaction value, we want a high precision because we want to label the relevant transaction as fraudulent. If the transaction is fraud the bank can add human interaction to verify by calling the consumer, if precision is low there will be burden hence the human interaction has to be increased.

For the banks which have larger transaction value, if the recall is low then it will unable to detect the transactions which are labeled as normal transaction. Hence consider the losses if the missed transaction is a high value fraud!

Hence, to save the banks from high value fraud transactions I have to focus on a high recall in order to detect the actual fraud transactions. Our XG Boost model is best suitable model for this.