Capstone Project – Credit Card Fraud Detection

Problem description

The problem statement is to build an intelligent model which can detect fraudulent transaction on credit card. The dataset is downloaded from Kaggle and its already transformed into PCA components to maintain confidentiality.

Data understanding

- The dataset has transactions made by credit cards in September 2013 by European cardholders. This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. It is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions.
- To protect the confidentiality the dataset has been modified with PCA. Except two feature 'time' and 'amount' all other features are transformed into V1, V2 etc as principal components. The **feature 'class' represents class labelling**, and it takes the value 1 in cases of fraud and 0 in others.

ML model building pipeline:

The dataset has lots of records. In a low memory system, it might cause problem for Pandas dataframe to load the entire dataset in-memory. In this case we might load data into pandas **dataframe as chunks using "*chunk_size*" at the time of EDA**.

EDA

Explained Variance

We can use "***Scree plot***" to find principal components which explains the maximum variances. If 70% of them could explain the majority of feature variance like say about 98%, then we may entertain only 70% of principal components into our model. Because rest 30% if we consider or not does not make much difference in our model predictability. Advantage of doing so is to **reduce the "*dimensionality*"** of the dataset.

Verify Skewness

Each of the variable in this dataset following Gaussian distribution so we need not perform any z-scaling. But we need to check the skewness of data. For example, the principal component V1 is mostly skewed towards the right side. Here we use "**Shapiro-Wilks**" test from Scipy library to check the skewness of each variables. For the highly skewed variables like V1 we may need to apply some transformation like log or square root.

Next, we will split the dataset in 70-30 ratio of **train and test set**.

Class Imbalance treatment:

This is a Minority Class problem. Here we need to use over-sampling, re-sampling techniques like **SMOTE/ADASYNC** from "**imblearn**" libraries and treat the class imbalance problem.

Model Building

Logistic Regression or SVM:

- First, we can try to see how simple ***logistic regression*** works here. It is simple and yet very trusted model which is more interpretable. If the data is linearly separable then we can use it.
- If there are more data points with overlap between classes, we can try to ***SVM classifier***. SVM classifier also has "class_weight" parameter to treat class imbalance.

Random Forest/XGBoost:

- Next, we will try to use various weak learners and build an ensemble model using Random forest. This will overcome the time complexity of earlier SVM model (which is quadratic) compared to linear in ensemble models. Also, it would help us to control over-fitting problem.

Hyper-parameter Tuning:

We can set a range like 0 to 50 incremented by 10 and try to use tune the hyperparameter using cross-validation techniques. Here, we can use **GridSearchCV** or **Stratified K-Fold Cross Validation** techniques from scikit learn libraries.

Model Prediction & Evaluation:

The dataset is PCA transformed so its better to use "**predict_proba**" library from scikit learn model evaluation. This gives predictions between 0-1.

In model evaluation we cannot rely on accuracy completely. Being a minority class problem, the accuracy here always be on higher side. Here we need to see True positive rate (TPR) v/s False positive rate (FPR) in ROC curve. If the AUC (area under curve) is more skewed towards TPR means the model performance is better. Because the ROC curve is measured at all thresholds, the best threshold would be one at which the **TPR is high and FPR is low, i.e., misclassifications are low.**

The above approach might vary while the training done on the actual data set in practical. This is only approach document based on the problem statement and not final.