Credit Card Fraud Detection

Background:

In recent times, the number of fraud transactions has increased drastically due to which credit card companies are facing a lot of challenges. For many banks, retaining high profitable customers is the most important business goal. Banking fraud, however, poses a significant threat to this goal. In terms of substantial financial loss, trust, and credibility, banking fraud is a concerning issue for both banks and customers alike. With the rise in digital payment channels, the number of fraudulent transactions is also increasing as fraudsters are finding new and different ways to commit such crimes.

The aim of this project is to identify and predict fraudulent credit card transactions using machine learning models.

Approach:

1. Data Understanding, Data Preparation and EDA:

From the given dataset suggests that data is highly imbalanced in nature. The positive class (frauds) account for only 0.5% of all transactions:

Class	0	1
Count	1842743	9651

Class is the target variable which we have to predict where 0 is normal transaction and 1 is fraudulent transaction. The other variables such as Time, V1-V28, Amount, and Class are features that can help us predict the target variable. Time is the time the transaction was done, V1-V28 are the principal components of the transaction, and Amount is the amount of money involved in the transaction.

Next Steps:

- No need of normalization since the PCA transformed variable are Gausian.
- Will start with basic EDA like correlation, boxplots etc for outliers.
- Transformation to mitigate and check the skewness in the data.
- Class Imbalances:
    - The normal Oversampling method won't be used here as it does not add any new information to the dataset and Under sampling will also not be used as it leads to the loss of information.
    - Therefore, we will try the below two class imbalance handling techniques

- **SMOTE** is a process where you can generate new data points, which lie vectorially between two data points that belong to the minority class.
- **ADASYN is similar to SMOTE**, with a minor change i.e. the number of synthetic samples that it will add will have a density distribution. The aim here is to create synthetic data for minority examples that are harder to learn, rather than the easier ones.

2. Model Selection and Building:

First, we need to split the data into train and test sets. We can use stratified sampling to ensure that at least 100 class 1 rows are in the test set. We can set the ratio to 80-20, with 80% of the data in the train set and 20% in the test set. Once we have the train and test sets, we can use various ML models to find which one works best with the imbalance data and produces the best results on the test data. We can try a variety of different models, such as logistic regression, random forests, support vector machines, and neural networks. We can use cross-validation to evaluate the performance of each model and compare the results.

- **Logistic regression** produces a formula that can be used to predict the probability of a given outcome. It is also useful for binary classification problems, where the output is either a 0 or a 1. Logistic regression is a **fast and simple algorithm** that can be used to quickly obtain useful results.
- **KNN** is a highly **interpretable algorithm**, but its use is not preferred when working with large datasets due to its computation intensive nature.
- **Decision tree models** are a good choice for data sets with a small number of features and low complexity, where the output needs to be easily interpreted and understood. However, decision tree models can be prone to overfitting if not properly monitored and tuned.
- KNN is a simple, supervised machine learning algorithm used for both classification and regression tasks. The k value in KNN should be an odd number because you have to take the majority vote from the nearest neighbours by breaking the ties.
- In **Gradient Boosted machines/trees**. newly added trees are trained to reduce the errors (loss function) of earlier models.
- **XGBoost** is an extended version of gradient boosting, with additional features like regularization and parallel tree learning algorithm for finding the best split. We will start with the Logistic regression model with the different value of regularisation and hyper param tuning, then go to the decision tree and so on etc.

We will also try ensemble models and use votingclassifier, Bagging, Boosting etc on the best performing individual models to find the best model.

Hyperparameter Tuning:

- we divide our data into k-parts, train the model k times each time using a different part as our testing set and the remaining parts as our training set. We then average

the results to get a final model. K-Fold Cross Validation helps us maximize our ability to use all our data for both training and testing while still getting an accurate estimate of our model's performance on unseen data. This process also helps us to reduce the variance of a single trial of a train/test split.

- Stratified K-Fold Cross Validation is an extension of K-Fold cross-validation, in which we rearrange the data to ensure that each fold is a good representative of all the strata of the data.
- When you have a small data set, the computation time will be manageable to test out different hyperparameter combinations. In this scenario, it is advised to use a grid search.
- But, with large data sets, it is advised to use a randomized search because the sampling will be random and not uniform.

In our case, we will use the Stratified K-Fold Cross Validation as the dataset is not really huge

3. Model Evaluation:

ROC AUC Score is the metric which helps us identify how well the model can distinguish between two things (e.g. a user buying a product or not). It is the area under the ROC curve which is calculated by plotting the True Positive Rate (TPR) against the False Positive Rate (FPR). The higher the AUC, the better the model can distinguish between the two classes.

ROC have better false negative than the false positives.

ROC-Curve = Plot between TPR and FPR

The threshold with highest value for TPR-FPR on the train set is usually the best cut-off.

We should not use the confusion matrix as the performance metrics as well as they have internally defined hard threshold of 0.5.

We also can't completely rely on the precision, recall and F1-score for now as they also have their strings attached of some threshold value.

ROC curve takes into cognizance of all the possible threshold values.

The ROC curve is used to understand the strength of the model by evaluating the performance of the model at all the classification thresholds.

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.

4. Cost benefit Evaluation:

High precision means that the model is correctly identifying the items it is trying to identify with minimal false positives. High recall means that the model is correctly identifying most, if not all, of the items it is trying to identify.

For example, if we are trying to diagnose medical conditions, then high precision is more important than high recall, since false positives can lead to incorrect treatments. On the other hand, if we are trying to detect fraudulent activity, then high recall is more important, since false negatives can lead to criminals getting away with their activities.

For banks with smaller average transaction value, we would want high precision because we only want to label relevant transactions as fraudulent. For every transaction that is flagged as fraudulent, you can add the human element to verify whether the transaction was done by calling the customer. However, when precision is low, such tasks are a burden because the human element has to be increased. For banks having a larger transaction value, if the recall is low, i.e., it is unable to detect transactions that are labelled as non-fraudulent. So consider the losses if the missed transaction was a high-value fraudulent one, for e.g., a transaction of $10,000

So here, to save banks from high-value fraudulent transactions, we have to focus on a high recall in order to detect actual fraudulent transactions.

We need to determine how much profit or dollar/rupee value we are saving with our best selected model