**Cap Stone – Credit Card Fraudulent Detection - By Juhi Srivastava and Lakshmi Kalyan Sunku**

**Project Understanding**

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

**Data Understanding**

* The data set includes credit card transactions made by European cardholders over a period of two days in September 2013. Out of a total of**2,84,807 transactions, 492 were fraudulent.**This data set is highly unbalanced, **with the positive class (frauds) accounting for just 0.172% of the total transactions.** The data set has also been modified with Principal Component Analysis (PCA) to maintain confidentiality. Apart from ‘time’ and ‘amount’, all the other features**(V1, V2, V3, up to V28)** are the principal components obtained using PCA. The feature 'time' contains the seconds elapsed between the first transaction in the data set and the subsequent transactions. The feature 'amount' is the transaction amount. The **feature 'class' represents class labelling**, and it takes the value 1 in cases of fraud and 0 in others.

**EDA**

* Since the data is transformed using PCA and is Gaussian Distributed, we can verify the distributions and spread of the data.
* The Data could be skewed and therefore we would do Anomaly Detection and Outlier treatment.

**Handling Class Imbalance**

* The data shows a very high “class imbalance”. Over 2,00,000 cases are mapped to 0, but hardly 500 cases are mapped to 1. This is a minority class problem.
* The underlying algorithm will learn more about the non-fraudulent cases rather than the fraudulent ones.
* This can be solved by over sampling or under sampling or Synthetic Minority Over-Sampling Techniqueor ADAptive SYNthetic (ADASYN).

**Train & Test Split of the data**

* We plan to split the Training and Testing as 75:25 split.
* For validation, we can use the k-fold cross-validation method.
* We would ensure that the split is stratified given the high imbalance in the data-set.

**Model Building**

Our plan is to build following modules

* Model with Logistic Regression as base model
* Model using Decision Trees
* Modelling with Random Forest
* Modelling using XGBoost
* Model using SVM
* Model using Neural Network/Deep Learning

**Hyperparameter Tuning**

* For hyperparameter tuning, random and grid search are the two methods available in scikit-learn in the form of RandomiszedSearchCV and GridSearchCV, respectively.
* We would like to evaluate both though, Grid Search would be giving us the exhaustive results.
* Each model has different types of hyperparameters, and we need to finetune all of these models using RandomiszedSearchCV/GridSearchCV.

**Model Evaluation:**

Our model evaluation criteria would be very careful as the dataset is highly imbalanced.

* We understand that measures like Accuracy can go for a toss with such imbalanced data. We would therefore not rely on the accuracy alone.
* We would also focus on achieving an optimal trade-off between the Precision and Recall. Both metrics are important depending upon the kind of bank and its operations. For banks with smaller average transaction value, high precision is more important while for banks having a larger transaction value Recall is more.
* Also, the predict method takes a default threshold of 0.5, which could not be an optimal one. Therefore, we would like to find the right threshold and therefore use Model AUC as an evaluation metric.

**Final Model Selection:**

Our model selection criteria would be as follows:

* Model should be easily interpretable, intuitive and results should be explanatory.
* Model should be the best trade-off between an overfit and underfitting model such that it generalizes well and at the same time has good performance on the unseen data.
* We would also try to factor in the computational power of the learning algorithm in terms of the space and time complexity. Some algorithms could be highly interpretable, but they may be computationally intensive, for example a K-NN model. We would therefore try to come up with an approach that has the optimal trade-off.