**American Express Artificial Intelligence Problem Statement 2**

**Understanding the dataset and goal:**

The dataset consists of 10 numerical and 44 categorical variables. Target is a binary variable (0 and 1). A value of 1 represents the transaction will be fraud and 0 means it will not be fraud.

The goal is to train a supervised model which will predict whether a given data will lead to fraud or not. For that, we will predict probability scores based on which we can take a decision. A higher probability for a data would mean higher is the chance of that becoming fraud. We are using a supervised model here as we know the target labels: 0 and 1. We need to classify a given data into either of the classes. This is a binary class problem.

**Exploratory Data Analysis (EDA):**

Before deciding on various models, we need to understand our data. I have done basic EDA to check if there are any missing values or any NULL values. Both the train and test data do not have any missing or NULL values. Next, I checked for outliers by using Boxplots. This was done only on the numerical columns: V1-V10. The boxplots showed fliers which means outliers. These outlier values (which are 3 standard deviations away from mean) can be replaced with mean. But in this case, I did not replace the outliers because I do not know the significance of those columns. The column descriptions are not given. Hence, I cannot blindly remove the outliers with no functional knowledge as it could lead to lose of data. So, no outlier treatment was done, and data was used as it is for train and test.

**Pre-processing:**

As part of preprocessing, I have done standard scaling of the data. This is because I do not know if the scales used for the numerical columns are the same or not. So, to be on the safer side, I have done standard scaling. The StandardScaler will normalize the feature so that the mean of the scaled data is 0 and standard deviation is 1. The StandardScaler object which is fitted on train dataset is later used on test dataset and the final test dataset used for scoring.

**Possible models:**

Since the problem belongs to binary classification, I considered a few ensemble and a few non-ensemble algorithms:

**Non-ensemble algorithms:**

* Logistic Regression
* Linear Discriminant Analysis
* Naïve Baiyes
* Decision Tree

**Ensemble algorithms:**

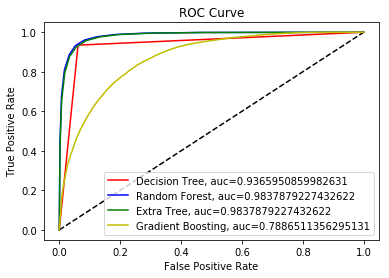
* Ada Boost Classifier
* Gradient Boosting Classifier
* Random Forest Classifier
* Extra Trees Classifier

*Note: I did not use KNN, LinearSVC, XGBoost as these 3 algorithms are not running in my laptop.*

All the models were run on a 5-fold cross validation. Cross validation is done to get a generalized performance of the model which is more thorough than running only in a train & test dataset. In a cross validation of 5-fold, it will split the train dataset into 5 folds and keep one-fold as a test dataset to evaluate the accuracy. Hence the model will be trained 5 different times, each time keeping a different fold as test dataset for evaluating model performance. Mean of the accuracies found in all the folds will give us a general idea of how the model will be once we use it the full train dataset and in real-time data. Accuracy used for evaluation for ‘roc\_auc’. The models which gave the highest scores were used for further evaluation. They are: Decision Tree, Random Forest, Gradient Boost and Extra Trees Classifier.

**High level model comparison:**

The selected models where run on the whole train dataset with no hyperparameter tuning. ROC curves were plotted which is shown below.

****

ROC Curve is a tool to analyze the behavior of classifiers at different thresholds. The X-axis of a ROC Curve represents False Positive Rates (FPR) and Y-axis is True Positive Rates (TPR).

* FPR = False Postives / (False Positives + True Negatives)
* TPR = True Postives / (True Positives + False Negatives)

As we can see from the above diagram, Decision Tree, Random Forest and Extra Trees Classifiers have given the best AUC score. Hence, they were selected for hyper parameter tuning.

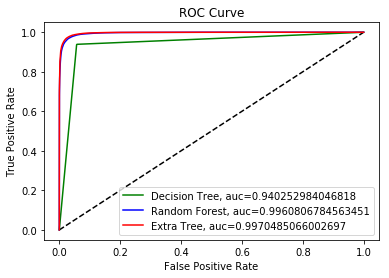
**Model Evaluation and Selection:**

Though we can fine tune many parameters for Decision Tree, I have used only criterion, max\_depth and min\_samples\_split for hyperparameter tuning. This is due to long runtime of the algorithm if I use more parameters for fine tuning. Ideally, we can fine tune many more parameters to get the best parameters for the best model. Using GridSearchCV, a 5-fold cross validation was done on the scaled training dataset and accuracy and AUC scores were obtained.

In the case of Random Forest (RF) and Extra Trees Classifiers (ET), I did not perform GridSearchCV due to runtime constraints. Instead, I used a larger value for **n\_estimators**. This is because a larger value of **n\_estimators** means, that many decision trees will be created, where each tree is different from the others. Also, I have used ‘entropy’ so that quality of each split will be measured by information gain. This is because the GridSearchCV done on Decision Tree has selected ‘entropy’ as a better parameter and hence I applied it for RF and ET also. Once again, it is possible to fine tune many more hyper parameters using GridSearchCV which can actually give the best parameters for the model.

**Accuracy and AUC comparison:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Decision Tree** | **Random Forest** | **Extra Trees** |
| Accuracy score | 0.9403 | 0.9729 | 0.9729 |
| AUC score | 0.9402 | 0.9961 | 0.9970 |



From the ROC curve, accuracy score and AUC score we can see that Random Forest and Extra Trees Classifier have performed the best. Both the algorithms have done almost the same. Also, I checked the precision and recall scores of Random Forest and Extra Trees Classifier using the confusion matrix. Since, Extra Trees Classifier has done marginally better, it was used to score the final test dataset.