**About:**

In business today, generation of leads could help save resources like time and money and increase the revenue. Moreover, if the business is able to correctly identify the customers of different segments and cross sell their products, it would obviously improve the business revenue, customer satisfaction, customer life time value through a deeper integration in a customer’s business.

Thus, generating leads for cross selling becomes a great strategy for both the business and the customer, creating a Win-Win.

**Problem Statement:**

The client is a happy bank with different kinds of bank accounts such as investment account, savings account, NRI accounts, fixed deposit account and so on. The idea is to find if the bank can do cross sales of the credit products among the customers of a different account.

Thus, the problem statement is to classify the customers and find if they will be interested to buy a credit card or not and their probability of getting the credit card.

**Approach:**

The train and test data consists of null values and categorical values in the columns. Thus, it is important to convert all the categorical data from object data type to integer data type and also fill the null values in the Credit product column. I have used user defined functions to label encode all the categorical variables.

Moving forward, the average account balance column in the data is positively skewed. But, for more efficient predictions of the model, I have used log transformations on the column to standardize the data. The feature engineering process is required for both the training and test datasets and thus, I have combined both the training and test dataset for data mining.

Now that all the data preprocessing and feature engineering is done, I have built models for prediction. The test data doesn’t have the predicting column “Is Lead “ and thus, to check the accuracy of the model, I have used roc\_auc\_score and Startified k folds to find the accuracy score the training data with the model I have built.

I have used Stratified K fold in the model, since a stratified K fold, preserves the percentage of the sample that is picked randomly, similar to the percentage of the groups in the actual population.

I have used three models, which are:

* Cat boosting
* Light Gradient boosting and
* Extreme gradient boosting.

Different models give different accuracies in roc\_auc score. So the output file consists of the average probabilities of customers getting the credit card from all the above three models.

**Cat Boosting Technique:**

I have used **Cat boosting** as one of the models because it is a machine learning algorithm that gives good results without even extensive data training compared to other machine learning algorithms. Cat boosting is the combination from the two words Category and Boosting. And thus, Cat boosting works well with multiple categories of data including, audio, images, text or a historical data. It also reduces the need for high parameter tuning and reduces the overfitting problem of the data on the model.

**Light gradient boosting technique:**

The next model used is **Light gradient boosting**, which is yet another framework based on the decision tree, which increases the efficiency of the model and reduces the memory that is being used. More over LGB works over two techniques:

1. Gradient based one side sampling
2. Exclusive feature bundling

The features with larger gradients contribute to higher information gain. This Gradient based one side sampling technique, keeps those features with larger gradients and only randomly drops those features with smaller gradients, thus retaining the accuracy of the information gain estimation.

LGB also helps in bundling the exclusive features safely into a single feature with nearly lossless approach to reduce the number of features. Moreover, since LGB is leaf wise split, this algorithm has a lower loss compared to other algorithms with level wise splits.

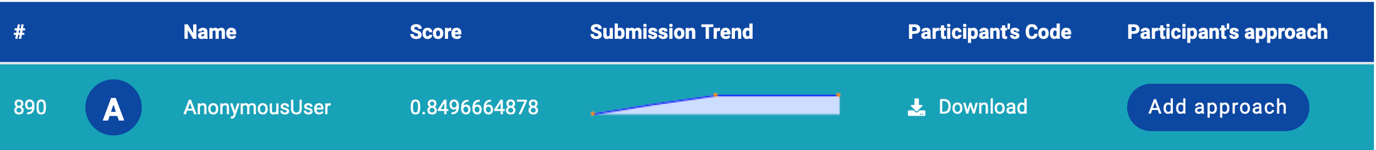
**Extreme Gradient Boosting:**

The third model used here is Extreme gradient boosting. These are constructed on decision tree models. The trees keep adding to the ensemble and it tries to fit, in order to correct the prediction errors that are made by prior models.

This technique uses a differentiable loss function and gradient descent optimization algorithm. Thus is the name gradient boosting, since the loss gradient is minimized as the model is fit, more like a neural network. It also pushes the limit of computations resources for boosted tree algorithms.

The output prediction is the average values of all the probabilities that is obtained from all the three models. Few other models such as Random Forest classifier, Ada boosting and bagging also gives a prediction, however it gives a lower accuracy compared to the LGB, XGB and Cat boosting.

**Leader Board – public leaderboard**



**Leader Board – private leaderboard**

