Supervised Modeling with Emphasis on LAUC

Problem Description

Supervised models are widely used to predict the probability of an event, such as whether a transaction is a fraud or an account will default or not. Many powerful algorithms exist to build suitable models, such as Boosting, Deep Learning, Support Vector Machines, etc. Measuring the effectiveness of such Machine Learning models is important to understand the value in the business. One measure is AUC (area under the curve), where the x-axis is the percentage of false positives and y-axis percentage of true positives. Due to the operational constraint/cost in many domains, only a small fraction of transactions can only be reviewed to confirm a prediction. For example, if a model scores 1MM transactions and the goal is to catch frauds, only top 5% transactions sorted by scores in descending order (assuming higher the score, more likely it will be a fraud) may be reviewed. In such a case, we also want high accuracy in the left area under the curve (LAUC) of the model besides good AUC. The goal is to build a supervised model on a sample of fraud data where both AUC and LAUC is as strong as possible

Data Description

Train Data Description:
Binary-class data with 406709 rows and 54 variables (10 Numerical and 44 Categorical). The header file is provided whose first row mentions the variable names and second row mentions the type of those respective variables (Numerical, Categorical). The first column of the data is the unique identifier for each row. The last column mentions the class/label for each row.

Test Data Description:
It is a 174303-row data without labels for scoring. Similar formatting as the train set, just without the last column.

This being a binary classification example, applying label encoding to categorical features and scaling to numerical features using LabelEncoder and MinMaxScaler from sklearn.

The distribution of classes is also not imbalanced.

Different models like LogisitcRegression, GaussianNB, KNNClassifier with 2 neighbours, DecisionTreeClassifier, AdaBoostClassifier RandomForest Classifier, Extra Trees Classifier were tried.
This involved fine tuning RF and Extra Trees Classifier(n_estimators=500, max_features=50).

Also tried ensembling RF and Extra Trees with LR to see if it outperforms both.

Also tried boosting models like XGBoost and LightGBM.

Results:

Auc score of LogisticRegression is 0.5
Time taken for training 0.7090487480163574 sec

Auc score of KNeighborsClassifier is 0.9510175850082945
Time taken for training 10.756164073944092 sec

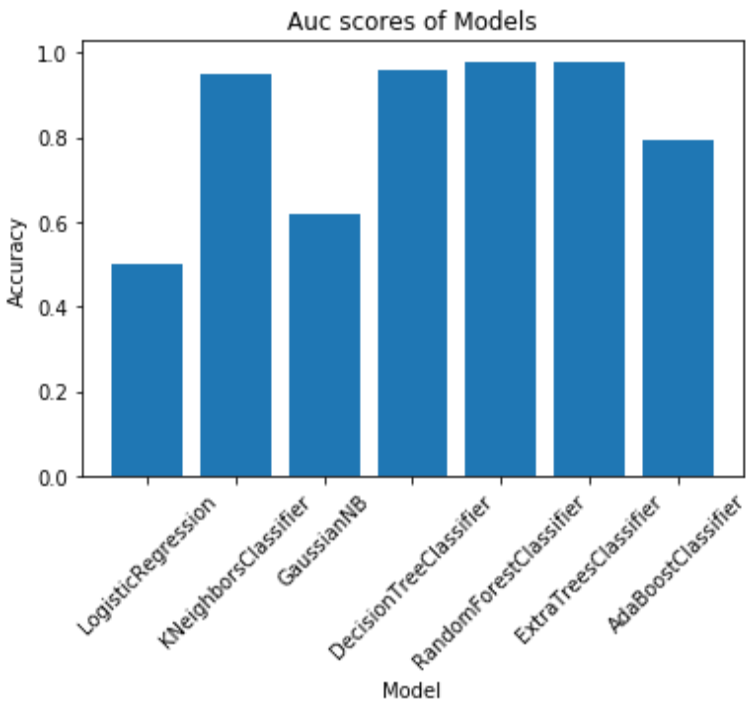
Auc score of GaussianNB is 0.6170495041473263
Time taken for training 0.7364809513092041 sec

Auc score of DecisionTreeClassifier is 0.9574693809621017
Time taken for training 5.860397100448608 sec

Auc score of RandomForestClassifier is 0.9759800694977427
Time taken for training 1720.5248358249664 sec

Auc score of ExtraTreesClassifier is 0.9782157653965512
Time taken for training 799.2141754627228 sec

Auc score of AdaBoostClassifier is 0.7909972645040979
Time taken for training 217.85605430603027 sec



Stacking (Ensembling)

RF Classifier : 0.9761478530111771
Extra Trees : 0.9777140384427824
Bagging Classifier: 0.9763476969144623
Stacked Classifier: 0.9767623085300513

XGBoost:

AUC: 0.8504329599279287

Light GBM:

AUC: 0.9776975984192887