Loan Delinquency Prediction

Prepared for: India ML Hiring Hackathon 2019
Prepared by: Gonella Surya Uma Maheswara Rao
25 August 2019

EXECUTIVE SUMMARY

Objective

Our Objective is Loan Delinquency Prediction.Predicting the Loan Delinquency i.e if the loan gets repaid or not based on the features given to us like unpaid principal balance, borrower credit score, payment date etc.

Approach

We can know that this is a classification problem from the objective as it has two classes 0 and 1. The first step always would be the better understanding of the problem statement and going through the data thoroughly as visualising the problem according to the data given will give us the better understanding of the problem statement. The next step is to analyse the data in the way how many categorical features are present, what preprocessing steps may be required, is the data is imbalanced, which way the inconsistent data could be removed etc. After these two steps we will start implementing it.In the implementation Exploratory Data Analysis Is must to get familiarised to data Although my EDA is not included in the code file.

Data Preprocessing

From the data analysis we came to know that there are 5 categorical features which require one hot encoding and there are not many categories to handle them with special care normal one-hot encoding works.As we have decided applying Random Forest so no need of feature scaling as the tree based model does not require it if it will be done also that does not have the affect as I tried applying feature scaling my score got reduced from the normal random forest score.So Random Forest generally does not require more preprocessing other than one-hot encoding. One more point that I found that is the data I got is imbalanced as there are majority of 0's and very less number of 1's so there is a need to handle this imbalance. So I have used SMOTE Algorithm to do upsampling and TOMEK links to do downsampling simultaneously so this way I have handled the imbalanced data.There can be so many feature engineering techniques can be applied but once again as we are using tree based model removing correlated features does not work good.There could more techniques tried like indicator variable, feature slicing etc which does not give more effect when we are trying random forest.So I haven't tried more feature engineering techniques.As we are given with the metric as F1 score there is no point of choosing the better performance metric and wherever I discuss the score that is a F1 score that I am discussing.

Model Selection

In Challenges generally we need to follow Kitchen-Sink Approach I.e build all basic models like KNN,Naive Bayes, SVM , Logistic Regression etc and try all the ensembles models on it like bagging , stacking , cascading etc. In this first I tried boosting models like random forest, xgboost , adaboost and from observation I found that normal random forest is giving better scores. Later I tried bagging over them by taking random forest , xgboost , adaboost as base models and logistic regression as meta classifier and bagging is giving me less score than the normal random forest model. So I tried stacking again it gave me less score so from this I

came to conclusion that the hyper parameter tuned random forest model would give the better results so I applied grid search cross validation on the random forest model by taking some parameters and it took me around 5-6 hours to complete after getting the tuned hyper parameters I got the better score and I ended getting 32 rank in the leader board. Then I got an idea of changing the test size as at first I have sliced my train set given into 0.75 and 0.25 for training and testing as test set is always important to know the performance of our model so I thought of increasing the training data would make the model better so I have changed the percent to 0.9 and 0.1 for which I got a best score of 0.3404255 and later I have tried so many attempts of hyper parameter tuned xgboost which gave decent score but not more than the best score and also I have taken three hyper parameter tuned models as my base models and applies stacking which also failed to give the best score.Also I have tried majority vote ensemble classifier which also gave decent score but not the best. So finally my hyper parameter tuned random forest is the best model. More Feature Selection is not a very good task to follow in competitions as they are good in real life to decrease the inference time of a model but in a competition score matters so eliminating more features is not a good task so I have tried eliminating some features so from one hot encoding I got 55 columns so I kept 52 columns which also gave decent score so after keeping 52 features our hyper parameters could change so there is a need of grid search cv again but because of lack of the processing power I am not able to do as the grid search is a computationally heavy process may be that could have given better score.

Key Take Aways From This Challenge

1) **Competitive Model Building** : Before this challenge I was building ML models keeping the real world cases in mind but in competitions like this we can know many things to improve the performance of our model for cost of anything. So this could teach you many lessons.
2) **Kitchen-Sink Approach** : Normally for making a Machine Learning model we analyse the data and the problem to decide which model gives you the best possible result but here we need to build so many models possible and ensembling them to get the better score this is called Kitchen-Sink approach.
3) **All Possible Techniques for Optimisation** : As we are trying for the better score possible we will check all the possibilities to optimise your model because of which we will learn the best possible techniques and experiment with them.

5 Things a Participant Must keep in Mind

1) Don't dive into the implementation without clearly understanding the problem, proper ground plan to attempt the challenge.
2) Don't think the solution as a real world problem attempting a challenge is different from building a real world ML model.
3) Don't remove any features for the cost of performance only remove those features which have zero and negative feature importance unless removing a least important feature also would make your performance little lesser.
4) Try out all the techniques which you feel could improve the model because as a Data Scientist our task is to try experimenting all the possibilities if you are not doing so you may not be a good Data Scientist.

5) Don’t just copy things from any of the references you referred like the blogs , code snippets etc try understanding them and apply it in the context of our problem. Whenever you refer from somewhere give the credits to that reference.