Loan Delinquency Prediction Prepared for

India ML Hiring Hackathon 2019

Gopal Kumar

26 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

Loan Delinquency is a classification problem from the objective as it has two classes 0 and 1 (0 = non-delinquent, 1 = delinquent) The first step would always be the better understanding of the problem statement and try to explore data analysis (EDA)
The general approach to tackle the classification problem is the following.

a) Cleaning the dataset:-
   i) Take care of NA Value
   ii) Remove the outlier

b) Scaling of data
   i) using min_max
   ii)using StandardScaler

c) Balancing the dataset

   a) Modifying loss function
      i)Use focal loss
   b) sampling
      i) undersampling - TOMEK
      ii) oversampling -SMOTE
      iii) upsampling and downsampling simultaneously

d) Feature engineering and feature extraction
   i) Correlation matrix
   ii) K-best
   iii) RFE
   iv)PCA

e) Model
   i) Random Forest with Grid search
   ii) Xgboost
   iii) Neural Network with focal loss
   iv) Auto-encoder

f) Evaluation matrix
   i) F1 score

Data Preprocessing

From the EDA the first thing we came to know that there are 5 categorical features ie (financial_institution,first_payment_date,loan_purpose,origination_date, source)
which require one-hot encoding and there are not many categories to handle them with special care normal one-hot encoding works. The second thing is the data is highly unbalanced there is 115422 non-delinquent, 636 delinquent. After visualization of EDA
We find that there are many outliers present in different Feature and when we try to remove this outlier we lose 10% of minority class so we don't remove the outlier.
Data is highly unbalanced so we try to balance the data set there are two way 1) updating the loss function (example Focal loss). 2) upsampling (SMOTE), downsampling (TOMEK) as well as upsampling and downsampling simultaneously. we work on both but I have used the SMOTE Algorithm to do upsampling and TOMEK links to do downsampling simultaneously for final submission.
Feature scaling plays an important role in ML but my final model is Random Forest so we did not worry about feature scaling on the other hand when we try to make auto-encoder for Loan Delinquency Prediction, we scale all feature in the range of (-2, 2).
We try to explore the feature engineering in this data like K-best, RFE, etc. actual all this decrease my model performance so we didn't take stress about feature engineering. For evaluation here we chose the F1 score.

Model Selection

In most Hackathon, people use bagging and stacking of basic models like KNN, Naive Bayes, RF, SVM, Logistic Regression, etc.
My selection of the models are,

a) Basic Model Logistic Regression, RF.
b) Neural Network with focal loss.
c) Auto-encoder.
d) Gradient boosting(Xgboost).
e) RF with Grid search.

Taking the test size 30% we trained all basic model-independent Logistic Regression gives f1 score 0.22, RF gives 0.28 f1 score and Xgboost gives 0.287 f1 scores.
And again combine all three models and the majority don't get any change in my model we get worse results than the previous.

One point that I noticed with decreasing the test size gives slightly better results than previous ones. Again we repeated the steps for test size 20%, at last, we fixed the test size on 10 %. At the beginning I don't apply Grid search so we take The basic ensemble model RF and apply grid search with cv =5 .it takes approx 4-5 hour train . in meanwhile we look this question different view actual all problems arise due to an unbalanced data set so when not we make an auto-encoder, trained on negative data set as non-delinquent (epoch = 100,batch_size = 64, learning rate 0.001) and define some threshold (i.e 3.4) if the MSE of predicting output and input is less than the threshold that goes to 0 and if greater that goes to 1. We try this approach and evaluate the model and we get a 0.26 f1 score. focal loss is one of the best loss tackle the unbalanced data set so we train Neural Network with two hidden layers and following parameters (batch_size =256, epochs =100, learning rate 0.001) but we did not get better results than the previous.
So now come to our final model RF with grid search we get best parameter is (criterion='entropy', max_depth=7, n_estimators=600, bootstrap=True)
And that gives the 34.042 f1 scores. after this, we try different combinations of stacking of model and tuning the hyperparameter that makes any sense in this context.
We try a different experiment on xgboost like feature engineering we find that if we remove categorical columns and without resampling data, we pass on xgboost i.e gives 30.5 f1 scores. so our final model is RF with a grid search that gives 34.042 on the leaderboard and secures the 47th rank.

Key Takeaways From This Challenge

Kitchen-Sink Approach - This new fashion for data scientists to ensemble the different find the best possible result.

Optimization - Find the best parameter using different optimization techniques like Grid search and bayesian optimization

Sensitive data - At last but not least take care of sensitive data.

5 Things a Participant Must keep in Mind

1) A clear understanding of the problem and try to make a road map for a given problem that helps more when your model predicts the unexpected result.

2) Instead of focusing model prediction try to focus on solving a real-world business problem.

3) Feature engineering is one of the best tools for data scientists that make the model efficient focus on those areas.

4) Perform some statistical tests and Try all possible ways to solve the problem including deep learning.

5) You refer from somewhere give the credits to that reference.