Model for Credit Risk Performance

Qianyu Dong: qianyu.dong@simon.rochester.edu

Xueqing Hou: xueqing.hou@simon.rochester.edu

Yaqing Sun: yaqing.sun@simon.rochester.edu

Yue Teng: yue.teng@simon.rochester.edu

Yilan Zhang: yilan.zhang@simon.rochester.edu

December 18th, 2019

1. Introduction

Our goal of this project is to develop a predictive model to evaluate the credit risk. According to our understanding, we first did some data cleaning to improve training data purity. After running eight different models and tuning them, we chose Boosting as our final model (with a test score of 73.25%). The final prediction showed a decision of good or bad for the entry row. Also, we designed an interactive interface that can be used visually by companies to decide on accepting or rejecting applications.

2. Data cleaning

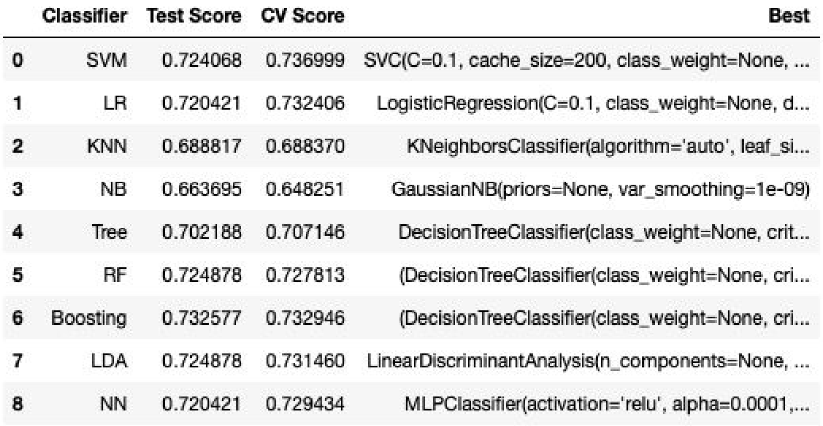
We first removed those rows with all -9 in every column since the values of all the features are missing and can contribute nothing to our prediction. Then, we remained two categorical features 'MaxDelq2PublicRecLast12M' and 'MaxDelqEver'. We used One-Hot Encoding to convert these categorical variables into a form that could be used for ML algorithms to perform better in prediction. For other 21 numerical features, we standardized them within each column.

Pipeline is used here to combine categorical and numerical features into one integration. This helps with new data handling since it'll also go through this combination process as well.

3. Model training

Our choice of model depends on the accuracy we get. We tried SVM, LR, KNN, NB, Tree, RF, Boosting, LDA and NN, and tuned these models accordingly. We looked at test score for model performance. This is because a better test score has the advantage of generalization when facing new, previously unseen data.

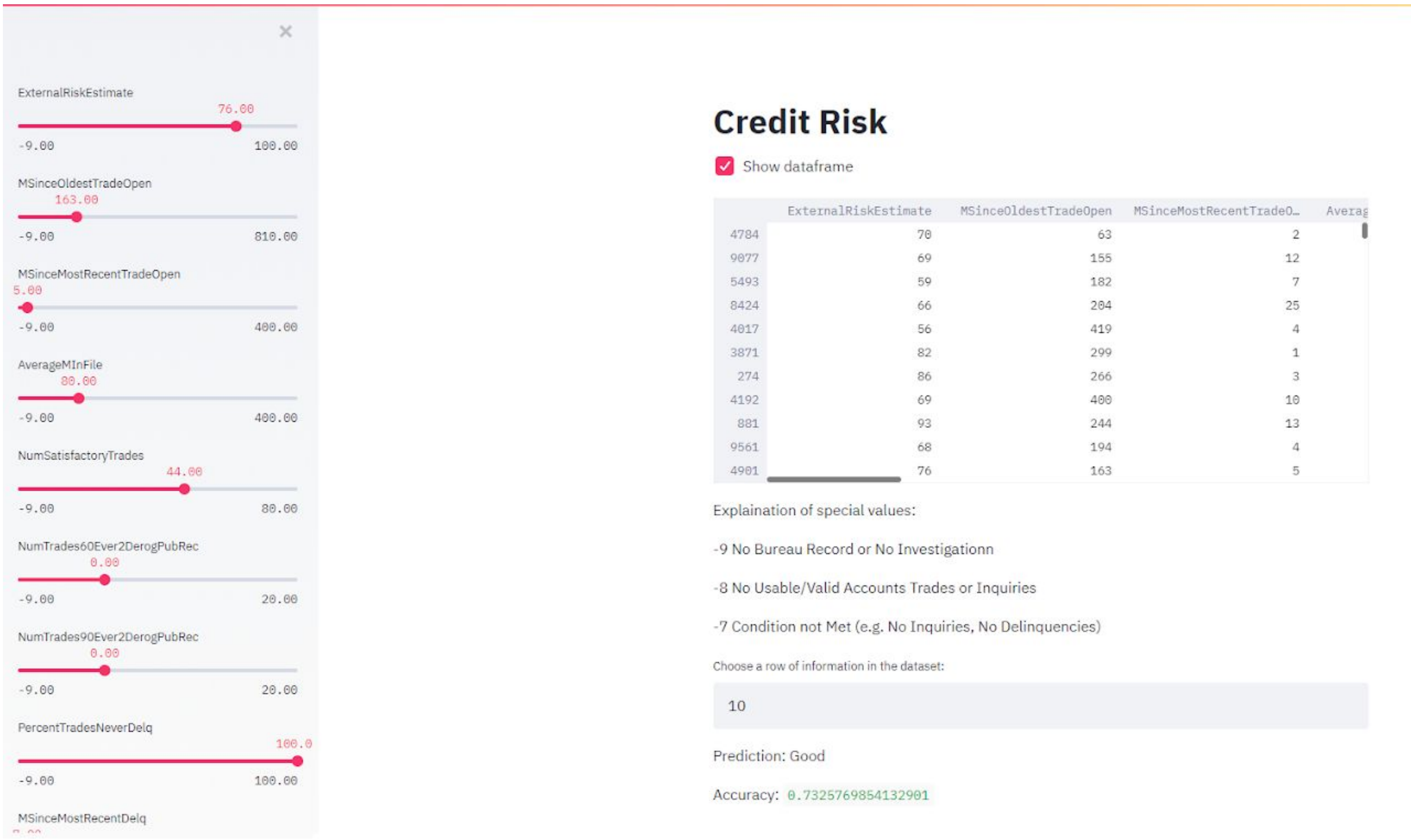
Boosting performed best here since it works well for solving the two-class classification problem and can optimize on different loss functions and provides several hyperparameter tuning options that make the function fit very flexible. The main idea of boosting is to add new models to the ensemble sequentially. At each particular iteration, a new weak, base-learner model is trained with respect to the error of the whole ensemble learnt so far.



	Classifier	Test Score	CV Score	Best
0	SVM	0.724068	0.736999	SVC(C=0.1, cache_size=200, class_weight=None, ...
1	LR	0.720421	0.732406	LogisticRegression(C=0.1, class_weight=None, d...
2	KNN	0.688817	0.688370	KNeighborsClassifier(algorithm='auto', leaf_si...
3	NB	0.663695	0.648251	GaussianNB(priors=None, var_smoothing=1e-09)
4	Tree	0.702188	0.707146	DecisionTreeClassifier(class_weight=None, crit...
5	RF	0.724878	0.727813	(DecisionTreeClassifier(class_weight=None, cri...
6	Boosting	0.732577	0.732946	(DecisionTreeClassifier(class_weight=None, cri...
7	LDA	0.724878	0.731460	LinearDiscriminantAnalysis(n_components=None, ...
8	NN	0.720421	0.729434	MLPClassifier(activation='relu', alpha=0.0001,...

4. Interface design

Within our interface, we add explanations to the categorical number so that users can know which number to choose in that feature. The range of the number for each feature is based on the round up number of the training data. They can either pull the bar to fit their own situation or enter a number in the input-box to get final result through prediction model. Here, for example, we entered ‘10’ to get the tenth row and the prediction shows good, which means the company should accept the application. However, row 100th is predicted to be bad so that the company should reject the application. The example results are presented in the figures below.



ExternalRiskEstimate 76.00 (-9.00 – 100.00)

MSinceOldestTradeOpen 163.00 (-9.00 – 810.00)

MSinceMostRecentTradeOpen 5.00 (-9.00 – 400.00)

AverageMInFile 80.00 (-9.00 – 400.00)

NumSatisfactoryTrades 44.00 (-9.00 – 80.00)

NumTrades60Ever2DerogPubRec 0.00 (-9.00 – 20.00)

NumTrades90Ever2DerogPubRec 0.00 (-9.00 – 20.00)

PercentTradesNeverDelq 100.0 (-9.00 – 100.00)

MSinceMostRecentDelq

Credit Risk

☑ Show dataframe

	ExternalRiskEstimate	MSinceOldestTradeOpen	MSinceMostRecentTradeO…	Averag
4784	70	63	2	
9077	69	155	12	
5493	59	182	7	
8424	66	204	25	
4017	56	419	4	
3871	82	299	1	
274	86	266	3	
4192	69	400	10	
881	93	244	13	
9561	68	194	4	
4901	76	163	5	

Explaination of special values:

-9 No Bureau Record or No Investigationn

-8 No Usable/Valid Accounts Trades or Inquiries

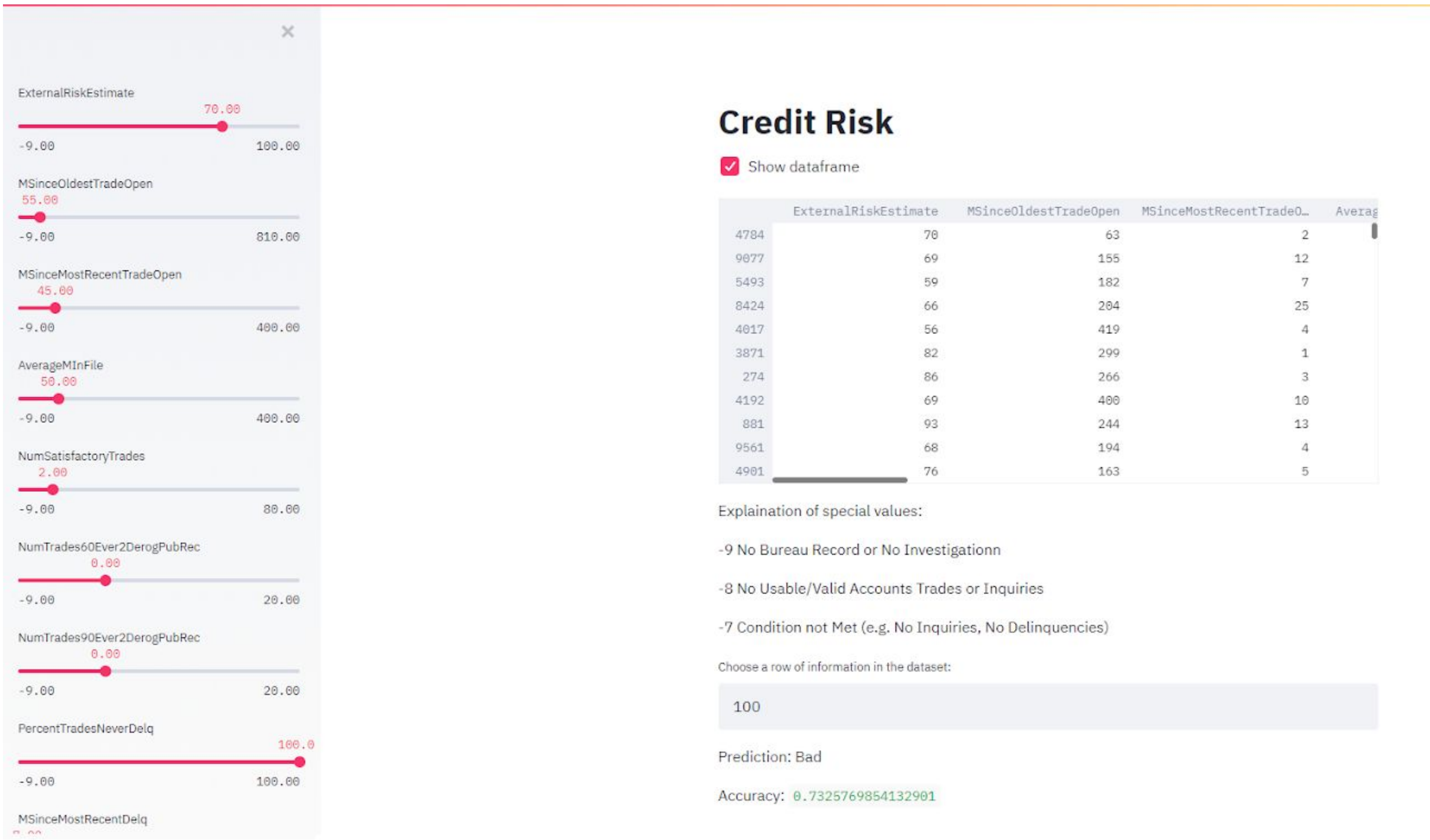
-7 Condition not Met (e.g. No Inquiries, No Delinquencies)

Choose a row of information in the dataset:

10

Prediction: Good

Accuracy: 0.7325769854132901



ExternalRiskEstimate
70.00
-9.00 100.00

MSinceOldestTradeOpen
55.00
-9.00 810.00

MSinceMostRecentTradeOpen
45.00
-9.00 400.00

AverageMInFile
50.00
-9.00 400.00

NumSatisfactoryTrades
2.00
-9.00 80.00

NumTrades60Ever2DerogPubRec
0.00
-9.00 20.00

NumTrades90Ever2DerogPubRec
0.00
-9.00 20.00

PercentTradesNeverDelq
100.0
-9.00 100.00

MSinceMostRecentDelq

Credit Risk

☑ Show dataframe

	ExternalRiskEstimate	MSinceOldestTradeOpen	MSinceMostRecentTradeO…	Averag
4784	70	63	2	
9077	69	155	12	
5493	59	182	7	
8424	66	204	25	
4017	56	419	4	
3871	82	299	1	
274	86	266	3	
4192	69	400	10	
881	93	244	13	
9561	68	194	4	
4901	76	163	5	

Explaination of special values:

-9 No Bureau Record or No Investigationn

-8 No Usable/Valid Accounts Trades or Inquiries

-7 Condition not Met (e.g. No Inquiries, No Delinquencies)

Choose a row of information in the dataset:

100

Prediction: Bad

Accuracy: `0.7325769854132901`