Simon Business School
Predictive Model for Credit Risk Performance
Group Number: #34

Xu Liu
xu.liu@simon.rochester.edu
Chen Cao
chen.cao@simon.rochester.edu
Shuyi Chen
shuyi.chen@simon.rochester.edu
Shuyu Huang
shuyu.huang@simon.rochester.edu
Yi Huang
yi.huang@simon.rochester.edu

1. Overview

In this assignment, we try to develop a predictive model and a decision support system (DSS) that evaluates the risk of a Home Equity Line of Credit (HELOC) applications. In this project, we want to train several models and select the best one to build up an interface for risk evaluation.

2. Data and Data Cleaning

First of all, we get the data about user info and their risk performance for training models.

In the data, there are 24 variables in all. The target (Dependent variable) is 'RiskPerformance', and other 23 variables are predictors. Here are the details of these variables. All of them are stored in numerical data type.

Variable Names	Description	Monotonicity Constraint (with respect to probability of bad = 1	Role
RiskPerformance	Paid as negotiated flag (12-36 Months). String of Good and Bad		target
ExternalRiskEstimate	Consolidated version of risk markers	Monotonically Decreasing	predictor
MSinceOldestTradeOpen	Months Since Oldest Trade Open	Monotonically Decreasing	predictor
MSinceMostRecentTradeOpen	Months Since Most Recent Trade Open	Monotonically Decreasing	predictor
AverageMInFile	Average Months in File	Monotonically Decreasing	predictor
NumSatisfactoryTrades	Number Satisfactory Trades	Monotonically Decreasing	predictor
NumTrades60Ever2DerogPubRec	Number Trades 60+ Ever	Monotonically Increasing	predictor
NumTrades90Ever2DerogPubRec	Number Trades 90+ Ever	Monotonically Increasing	predictor
PercentTradesNeverDelq	Percent Trades Never Delinquent	Monotonically Decreasing	predictor
MSinceMostRecentDelq	Months Since Most Recent Delinquency	Monotonically Decreasing	predictor
MaxDelq2PublicRecLast12M	Max Delq/Public Records Last 12 Months. See tab "M	Values 0-7 are monotonically decreasing	predictor
MaxDelqEver	Max Delinquency Ever. See tab "MaxDelq" for each c	Values 2-8 are monotonically decreasing	predictor
NumTotalTrades	Number of Total Trades (total number of credit acco	No constraint	predictor
NumTradesOpeninLast12M	Number of Trades Open in Last 12 Months	Monotonically Increasing	predictor
PercentInstallTrades	Percent Installment Trades	No constraint	predictor
MSinceMostRecentInqexcl7days	Months Since Most Recent Inq excl 7days	Monotonically Decreasing	predictor
NumInqLast6M	Number of Inq Last 6 Months	Monotonically Increasing	predictor
NumInqLast6Mexcl7days	Number of Inq Last 6 Months excl 7days. Excluding t	Monotonically Increasing	predictor
NetFractionRevolvingBurden	Net Fraction Revolving Burden. This is revolving bala	Monotonically Increasing	predictor
NetFractionInstallBurden	Net Fraction Installment Burden. This is installment b	Monotonically Increasing	predictor
NumRevolvingTradesWBalance	Number Revolving Trades with Balance	No constraint	predictor
NumInstallTradesWBalance	Number Installment Trades with Balance	No constraint	predictor
NumBank2NatlTradesWHighUtilization	Number Bank/Natl Trades w high utilization ratio	Monotonically Increasing	predictor
PercentTradesWBalance	Percent Trades with Balance	No constraint	predictor

MaxDelq2PublicRecLast12M

value	meaning
0	derogatory comment
1	120+ days delinquent
2	90 days delinquent
3	60 days delinquent
4	30 days delinquent
5, 6	unknown delinquency
7	current and never delinquent
8, 9	all other

-9 No Bureau Record or No Investigation
-8 No Usable/Valid Trades or Inquiries
-7 Condition not Met (e.g. No Inquiries, No Delinquencies)

We made the following changes:

a. Change the data types of 'MaxDelq2PublicRecLast12M' and 'MaxDelqEver' from numeric variables to categorical variables, since the numbers of these two variables are not just numbers, having special meanings.

b. In 'MaxDelq2PublicRecLast12M', value 5,6 / 8,9 have the same meaning.We combined 5,6 into 5 and 8,9 into 8.

c. Dealing with special value -7, -8, -9.

   Value of -7 is caused by a sample that does not meet specific requirements or conditions. When training a model, the special value -7 is the most troublesome, because its negative value directly contradicts the monotonicity of the features appearing in the model. We use a trial and error-based approach to test the model's response to a wide range of positive values. When -7 was replaced by 150, the training accuracy of the model peaked, which makes sense in the context of the problem.

   A special value of -8 indicates that no available or valid cases were found. This means that the accounts, transactions and queries involved are either inactive or very stale. However, we are not sure to replace it by mean or median, so we decided to replace -8 with both mean and median values and compared respective results afterwards.

   The special value of -9 is assigned to fields that do not have credit history or score information. In the data set, most of these values appear together, and each feature has such a value. Since no information can be inferred from these -9 samples, as they are only used as noise, we decided to ignore these samples.

d. Changing the target variable into dummy. The values for target variable ’Risk Performance’ are ‘bad’ and ‘good’, we change bad into ‘1’ and good into ‘0’ for future prediction.

3. Model Selection

   After data cleaning, we ran several models to train the best prediction models.

   At first, we chose some common models which were taught from class like LDA, Decision Tree Classifier, AdaBoost, Logistic Regression, QDA and GaussianNB. After tuning parameters, we got the best score for each model.

Since we have different imputer (mean/median), we compare the test results using mean and using median. Here are the results:

		SVC	LDA-svd	LDA-lsqr	AdaBoost	Logistic Regression	KNN	Decision Tree	QDA
Imputer= mean	CV_Score	70.34%	73.17%	72.42%	**73.53%**	72.34%	69.80%	70.32%	70.23%
	Test_Score	71.52%	71.87%	71.70%	**72.07%**	71.75%	69.74%	71.02%	69.31
Imputer= median	CV_Score	**72.51%**	71.48%	71.50%	72.24%	70.17%	71.41%	70.59%	70.29%
	Test_Score	**72.42%**	70.94%	70.94%	72.32%	70.83%	70.41%	69.69%	69.98%

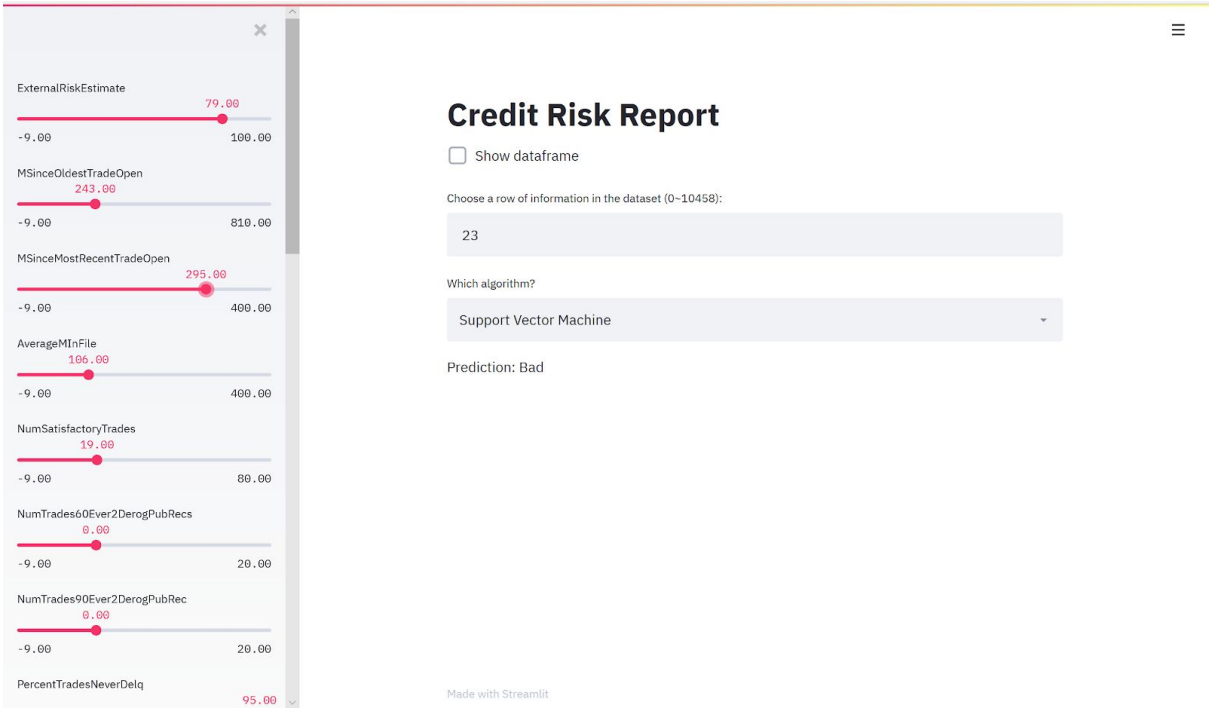
(Using median/mean as the imputer)

The best models are quite different using these two imputer. But the top test score for each imputer are around 72%. As a result we decided to use SVC(test score 72.42%) using median as imputer as our rating model.

4. Best Model and Interface
   The performance of model SVC is the best.

   SVC=svm.SVC(C=1, cache_size=200, class_weight=None, coef0=0.0, decision_function_shape='ovr', degree=3, gamma='auto_deprecated', kernel='rbf', max_iter=-1, probability=False, random_state=1, shrinking=True, tol=0.001, verbose=False)

   Following is a picture of our interface. On the left, the side bar can control the input of 23 variables. We insert SVM as the algorithm, after dragging the variables, a result of good/bad will pop out.



5. Summary

According to the test scores, SVC has the best performance among all the models that we rated. However, based on our attempt, Simple Neural Network could have better accuracy that may be applied to improve the risk prediction. Our Team will keep optimizing our model and interface in the future.