Home Equity Line of Credit

Risk Prediction | December 2019

Agenda

- ❑ **Project Objectives and Scope**
- ❑ Project Methodology and Approach
- ❑ Appendix

Project Objectives and Scope

Background

Credit scores are an important factor that financial institutions consider when deciding whether or not to approve a loan. The scores are designed to predict the likelihood of repayment of a loan, and customers can get explanations for their scores, e.g. “The proportion of your revolving balances to total balances is too high” or “you recently opened a new account.” Regulators require that financial institutions provide reasons to customers when taking “adverse action” (i.e. turning down a loan) or in customer service settings when responding to customer inquiries.

Project purpose

The goal of the project is to develop a model to assess the risk of credit line applications. A data set consisting of 23 features and an outcome variable classifying applications as either ‘Bad’ or ‘Good’ was provided.

Scope

The project scope includes the following activities based on requirements from FICO community competition*:

- The main part is to develop a predictive model and a decision support system (DSS) that evaluates the risk of Home Equity Line of Credit (HELOC) applications.
- In addition, an interactive interface designed for bank/other credit lender employees to use is built using Python. This interface allows an employee to enter the relevant data from the application, and a result of either ‘Good’ or ‘Bad’ will be returned based on the model prediction for the entered parameters.

*https://community.fico.com/s/explainable-machine-learning-challenge

Agenda

- ❑ Project Objectives and Scope
- ❑ **Project Methodology and Approach**
- ❑ Appendix

Project Methodology and Approach

Step 1

Data cleaning / preparation*

Step 2

Model testing and selection

Step 3

Developing an interface

Step 4

Conclusions / Future recommendations

Model selection

Interface design

Two issues

- **Negative numbers obviously unreasonable. For example, AverageMInFile(Average Months in File), NumSatisfactoryTrades(Number Satisfactory Trades) are non-negative numbers but in some lines all negative numbers generate a "Good". So drop all of -9 because 598 rows are entirely negative 9.**
  Syntax:df = df[df!=-9]
  df = df.dropna()
  df.reset_index(inplace=True, drop=True)

- **"MaxDelq2PublicRecLast12M" and "MaxDelqEver", these two columns should be categorical values according to their definitions.**
  Syntax:
  df['MaxDelqEver'].astype(''category)

*See dataset description in Appendix

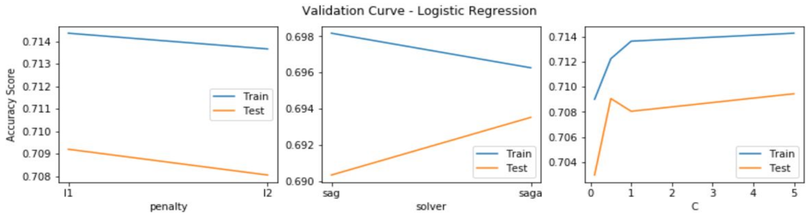
Project Methodology and Approach
Model Testing and Selection

A Plot and manually investigate accuracy of various models against variable hyperparameters

B Use grid search to tune parameters and find the best performing model for each of the three algorithms

A

An example for a logistic regression model with changing parameters



B

The optimal parameters for each algorithm are as follows

```
Classification Algorithm: Logistic Regression
Optimal Hyperparameters: {'C': 1, 'penalty': 'l1', 'solver': 'saga'}
Classification Algorithm: SVC
Optimal Hyperparameters: {'C': 0.5, 'gamma': 'scale', 'kernel': 'rbf'}
Classification Algorithm: Random Forest
Optimal Hyperparameters: {'max_depth': 8, 'min_samples_leaf': 4, 'min_samples_split': 5, 'n_estimators': 300}
```

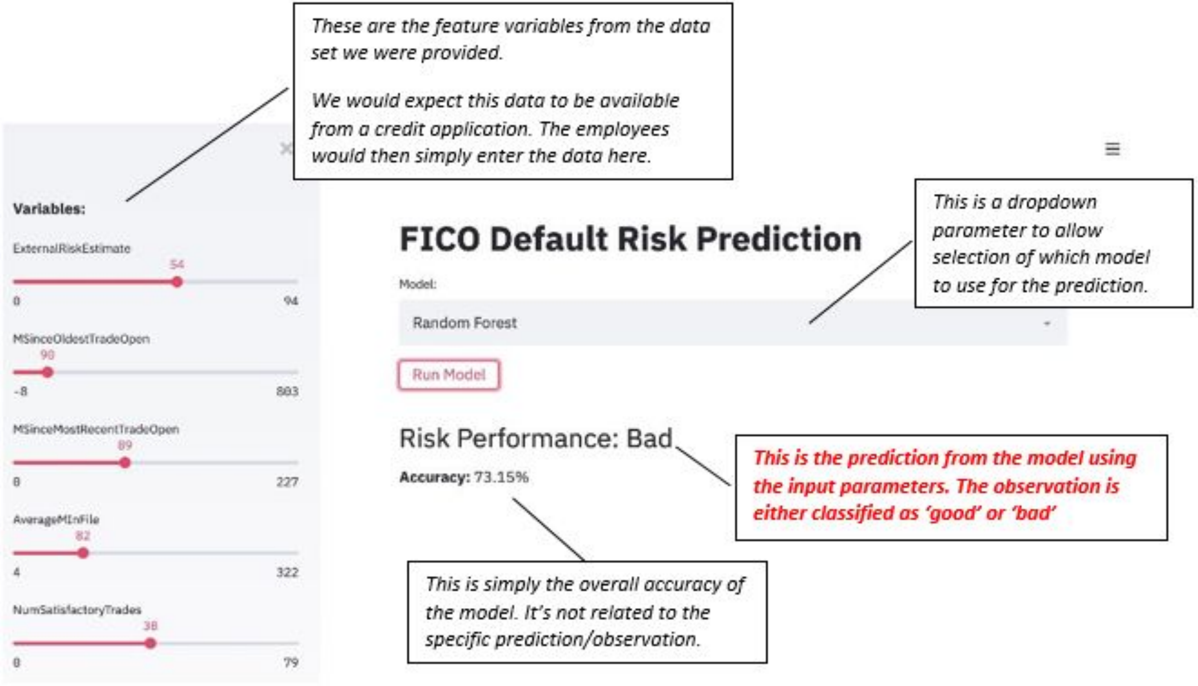
Three measures to choose which model to use

- **Cross validation score**: This is somewhat obvious. The most accurate model is the best.
- **Recall**: From context of the project this is important. Falsely classifying a bad as good is more costly than falsely classifying a good as bad, as defaulted loans are more costly than the missed opportunity cost of one an additional loan.
- **Interpretability**: Ideally, not only will predictive model be accurate, it will also be informative. We would like to be able to explain to applicant what features are preventing them from getting a mortgage.

	Model	Accuracy	Precision	Recall	F1-Score
0	Logistic Regression	0.711273	0.711273	0.711273	0.711273
1	SVC	0.507299	0.257353	0.507299	0.341475
2	Random Forest	0.732766	0.733209	0.732766	0.732477

Project Methodology and Approach
Developing an interface

The working interface is successfully developed using Streamlit in Python, that an employee of a credit lender could use to help inform them to either accept or reject an application for credit.



Project Methodology and Approach
Conclusions / Future recommendations

Model Selection

Looking back more than 3 kinds of algorithms as well as different cross validation / optimization techniques for parameter selection have been tried in this project. There are many models to try and although the random forest model we ended up on is preferred, it doesn't hurt to try more.

Interface Design

The user is expected to have good business knowledge and understand what the parameters are but any real statistics knowledge is not required. Thus, we would also like to create a more informative interface.

Remove the model selection drop-down parameter.

In case that a bank employee does not know what SVC or a random forest is and they shouldn't have to, we should simply pick the best model beforehand and only use that model. (For our first pass at that it would have been the random forest model)

Remove the accuracy score.

Currently it is showing overall performance of model from our tests/validation. It is unrelated to the current prediction of the model. In case that the bank employee and applicant don't understand the figure and misinterpret it as somehow related to the current prediction, the accuracy score is suggested to be removed.

Add some assistance for interpretability.

One technique could be tried is LIME, by changing feature(s) from observation while holding other features constant and seeing how it impacts prediction. Thus, a ranked list of feature importance for that specific observation can be created.

Ideally supportive statistics that align with the model can also be included. For example, if LIME analysis showed that FactorA contributed most to their denial, a historical statistic could be presented as ‘87% of people with a FactorA <= theirs had a poor risk performance’.

Agenda

- ❑ Project Objectives and Scope
- ❑ Project Methodology and Approach
- ❑ **Appendix**

Appendix
Dataset Description

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 "MaxDelq" for each c	Values 0-7 are monotonically decreasing	predictor
MaxDelqEver	Max Delinquency Ever. See tab "MaxDelq" for each category	Values 2-8 are monotonically decreasing	predictor
NumTotalTrades	Number of Total Trades (total number of credit accounts)	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 the last 7 days remo	Monotonically Increasing	predictor
NetFractionRevolvingBurden	Net Fraction Revolving Burden. This is revolving balance divided by cre	Monotonically Increasing	predictor
NetFractionInstallBurden	Net Fraction Installment Burden. This is installment balance divided l	Monotonically Increasing	predictor
NumRevolvingTradesWBalance	Number Revolving Trades with Balance	No constraint	predictor
NumInstallTradesWBalance	Number Installment Trades with Balance	No constraint	predictor
NumBank2NatlTradesWHighUtilizatior	Number Bank/Natl Trades w high utilization ratio	Monotonically Increasing	predictor
PercentTradesWBalance	Percent Trades with Balance	No constraint	predictor