# Default Classification Analysis

FICO800: Sunjoo Park (sp5267), Eva Shuyu Wang (sw2860),
Ruoyu Zhu (rz1403), Zijun Cheng (zc492)

December 8, 2018

## Contents

# 1 Introduction

## 1.a Business Understanding

Every year, credit scoring methodologies provide millions of scores that evaluate the risk in billions of dollars in loans. In fact, these FICO Scores are used in more than 90% of lending decisions in the US. The scores are designed and used to predict the likelihood of repayment of a loan, and the accuracy of these predictions determine the profit or loss a financial institution incurs from making lending decisions.

Because of that, advanced machine learning methods are quickly finding applications throughout the financial services industry and achieved great predictive successes. However, there is a huge gap between our ability to construct effective predictive models and our ability to understand and control these models. The black box nature of machine learning algorithms means that they are currently neither interpretable nor explainable. Furthermore, regulators require financial institutions to provide reasons to customers when taking “adverse action”, i.e. turning down a loan, and these customers likely would demand to have explanations for their result. Some possibilities include “The proportion of your revolving balances to total balances is too high” or “you recently inquired a new loan.”

In settings where regulators or consumers demand explanations, more sophisticated machine learning techniques are needed. The techniques should offer both the promise of increased accuracy and explainability at the same time.

## 1.b Data Understanding

To incentivize research in this area, FICO has launched a challenge with a home equity line of credit dataset where the objective is to create models that are both accurate and interpretable. And our dataset comes from the FICO Explainable Machine Learning Challenge, which focuses on an anonymized dataset of Home Equity Line of Credit applications made by real homeowners. (1) We define the issue as a classification problem: the fundamental task is to use applicants’ information as predictors to forecast whether they will repay their loans within two years. Our goal is to build an explainable model with accurate prediction. Loan borrowers in this data set have requested a credit line in the range of $5,000 - $150,000.

The dataset contains 10,459 rows in total, where each row represents an account of a loan applicant with one target variable and 23 features. The target variable is called “Risk Performance: Paid as negotiated flag” which is a string of Good and Bad. Specifically, a “good” indicates the borrowers have made their payments without ever being more than 90 days overdue. And a “bad” means that a consumer was 90 days past due or worse at least once over a period of 24 months from when the credit account was opened. We convert the target into binary variable where Bad = 1 and Good = 0. The predictors are quantitative or categorical data extracted from applicants’ credit history. Following are some of features:

- Consolidated version of risk markers (variable name: ExternalRiskEstimate)

- Average Months in File (variable name: AverageMInFile)

- Number Trades 60+ Ever (variable name: NumTrade60Ever2DerogPubRec)

- Months Since Most Recent Delinquency (variable name: MSinceMostRecentDelq)

- Number of Inquires Last 6 Months excluding last 7 days (variable name: NumInqLast6Mexcl7days)

For the full descriptions of variables, Please refer to the data dictionary in the Appendix B.

Of all these features, one controversial issue is whether to include the 'Consolidated version of risk markers' feature or not. It collects credit score from external sources which we do not have any information about. It affects our model's explainability so we need to be cautious about using this predictor. However, our goal includes carrying out accurate data, so we retain it, but keep this issue in mind for the future understanding of the model.

## 1.c Data Preparation

The raw data set includes a variety of complicating factors such as the sporadic distribution of special values and a few categorical data in a predominantly continuous data set. To limit the effect of the special value—non-monotonic numerical data—on our model, we did some transformations on the raw data.

Here is a summary of the main transformations that we performed:

### 1.c.1 Dealing with Special Values

- Special Value -9: No Bureau Record or No Investigation

  The special value of -9 was assigned to those fields for which no credit history or score information was available. There were 588 instances for which all the feature has -9 value except for the target variable. The target values of those instances included both 0s and 1s. We deleted those instances as it gives us no information for prediction.

  There were also 10 instances having -9 in the External Risk Estimate feature. For those instances, we used simple linear regression on reset 22 features to replace the -9 values.

- Special Value -8: No Usable/Valid Trades or Inquiries

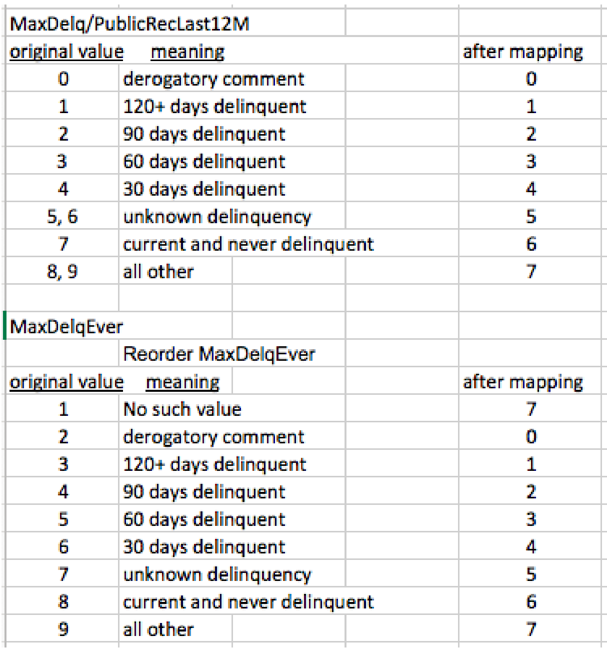
  The -8 values appeared in variables 'MSinceOldestTradeOpen', 'MSinceMostRecentDelq', 'MSinceMostRecentInqexcl7days', 'NetFractionRevolvingBurden', 'NetFractionInstallBurden', 'NumRevolvingTradesWBalance', 'NumInstallTradesWBalance', 'NumBank2NatlTradesWHighUtilization', 'PercentTradesWBalance'. Since all these variables are continuous, we used k-Nearest-Neighbour (k = 5) to make a prediction of what value would have been had it not expired.

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

The -7 values appeared in variables 'MSinceMostRecentInqexcl7days', 'MSinceMostRecentDelq'. If the customer had never been delinquent a -7 value was assigned. Since both of the variables are monotonically decreasing (higher the number, lower the possibility that the customer would default), then we assigned a relatively large value, in our case, 200 to replace -7.

### 1.c.2 Dealing with Categorical Features

There are two categorical predictors labeled 'MaxDelq2PublicRecLast12M' and 'MaxDelqEver'. Both predictors had the same range 0-9, but some of the values were monotonically decreasing while some other values have specific meanings. In order to keep the meaning of values consistent and monotonic, we remapped the values in both columns. See details in Figure 1.



MaxDelq/PublicRecLast12M

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

MaxDelqEver

Reorder MaxDelqEver

| original value | meaning | after mapping |
|---|---|---|
| 1 | No such value | 7 |
| 2 | derogatory comment | 0 |
| 3 | 120+ days delinquent | 1 |
| 4 | 90 days delinquent | 2 |
| 5 | 60 days delinquent | 3 |
| 6 | 30 days delinquent | 4 |
| 7 | unknown delinquency | 5 |
| 8 | current and never delinquent | 6 |
| 9 | all other | 7 |

Figure 1: Result of mapping data

## 1.d Target and Feature Distribution

The target variable is bell-shaped and roughly resembles a normal distribution with mean around 0.52. The data set is not particularly skewed and the population seems to be clean in the sense that they are following one probability distribution (Figure 2).

A summary for the most important features according to logistic regression and random forest is also illustrated after the data is cleaned (Figure. 3)

# 2 Modeling and Validation

There are several good data mining algorithms for binary classification, leading us to a variety of options for prediction. Since we wanted to explore both linear models and black box models given our purpose is

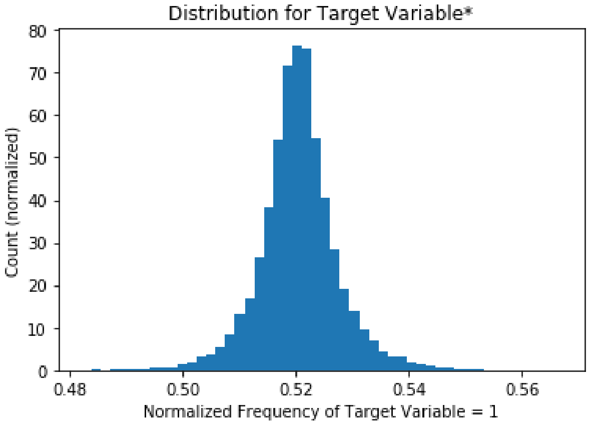
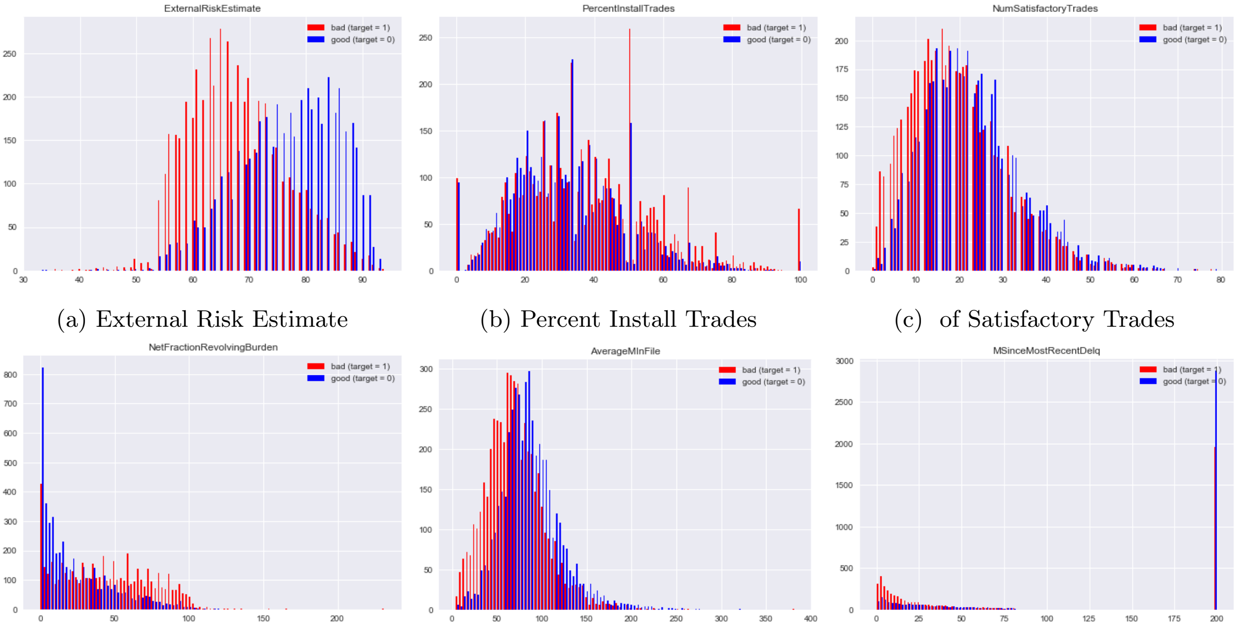


Figure 2: Distribution of Target Positives

*The distribution was obtained by randomly sampling a certain portion (a random number between 2000 and 8000, this threshold is arbitrary) of the population for 10,000 times and count each time the number of positive cases. The frequencies are normalized.



(a) External Risk Estimate

(b) Percent Install Trades

(c) of Satisfactory Trades

(d) Revolving Burden

(e) Avg. Min File

(f) M since Most Recent Delq.

Figure 3: Top Feature Importance

to predict binary classification with high explainability, we choose to use logistic regression, decision tree, support vector machine, random forest, adaptive boosting, and neural network.

## 2.a Baseline Model

The baseline model is essentially a decision tree with only one split using one feature — ExternalRiskPerformance (ERP). The model can also be interpreted as a linear split in the instance space using only this feature as the predictor. To make estimates on one instance's target value, the mean of the feature from the training set is calculated. We then labeled all the instances with an ERP smaller than the mean with a uniform score of "1" (positive cases), while the instances with a higher ERP than the training mean will be labeled as "0" . Due to its simplicity and the linearity, this model could be the starting point for many other models such as decision trees, svm and logistic regression.

## 2.b Linear Models

### 2.b.1 Improved Baseline Model

To improve the baseline model, one simple technique — a distance function $ERP_i - mean(ERP_{train})$ — was applied to substitute the binary scores, for each instance i, in the baseline: the more negative the distance function is, the more likely it is for an instance to be of a positive case. This variation is equivalent to the baseline model with a customized score function dependent only on the most dominant factor ERP (Figure 8). In particular, data was both trained and tested using the distance function, and the AUC has been improved to 77.5% although accuracy remains the same as the one from the baseline. It is worthwhile to mention that modifying the scoring function of one feature alone has elevated the AUC to the level of the tuned decision tree (AUC = 77.7%).

### 2.b.2 Decision Trees

Data was thrown into a default decision tree classifier first, and the result was poor with an accuracy rate of only 64%, which is even lower than that of the baseline model. The low accuracy was mainly a result of over-fitting (Figure.4).

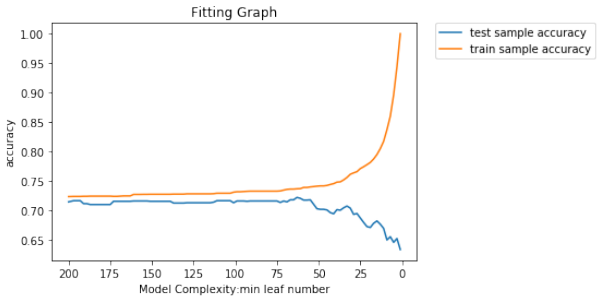


Figure 4: Fitting Graph for Decision Tree

Based on Figure 4, it looks like the sweet spot is around a minimum leaf number of 100. To be precise, model complexity was controlled systematically through searching for the combination of minimum number of leaf and minimum number of splits between $[2, 100]$ and $[2, 1000]$ to obtain the highest level of accuracy. The model lands in the set of parameter with minimum number of leaf = 112 and minimum number of splits = 2 for an accuracy score of 72% and an AUC of 77.7% (by holding 10% of data for validation). This result is slightly better than the improved baseline model in terms of AUC. However, please note that the tree has used 23 more features than the improved baseline while AUC only improved by less than 1%.

### 2.b.3 Logistic Regression

Logistic regression is empirically proven to be robust on small data classification problems with relatively large number of predictors. It is also suited for feature rankings, and more explainable than black-box methods. Another possible mining technique is Naive Bayes. But it assumes that all of the predictors are independent with each other, while some of our predictors are likely to be correlated. (For example, (NumInqLast6M and NumInqLast6Mexcl7days), we chose Logistic Regression over Naive Bayes as one of our initial methods.

We started initial training with default settings. (C = 1.0, penalty = L1) The initial accuracy is 72.9%, and the AUC is 80.7%. Then we tried to improve its performance by applying polynomial features and grid search with cross-validation(with $n_{splits} = 10$). Key parameters for tuning include Cs, regularization terms, and Polynomial Features and interactions terms. We also include normalization in the pipeline to ensure the results are scaled and coefficients are comparable. The best model from grid search has a C = 0.1, penalty = L1, polynomial degree = 1. It is pretty close to the hyper-parameter settings of the initial LR model. Therefore, the accuracy of the best model is slightly improved to 73.3%, and the AUC remains at 80.7%.

## 2.c Black Box Models

### 2.c.1 SVM

SVM is another linear model which can also efficiently perform non-linear classifications using kernel tricks. It has intuitive geometric interpretation and strong learning guarantees. Our initial model with default settings (C = 1, gamma = 0.04, kernel = 'rbf') was able to achieve an accuracy of 72.8%, and the AUC is 80.6%.

Following the same procedure as we improve logistic regression, we implemented the algorithm using grid search with 10 folds cross-validation. There is a slight improvement on the best model compared to the previous one, with C = 1, gamma = 0.04, kernel = 'rbf', The best performance is achieved with accuracy = 73.3% , and AUC = 81.0%

### 2.c.2 Random Forest

Since random forest assures less variance and overfitting by selecting trees randomly and averaging them out, we chose to use this model. Here, we used Scikit-learn randomforestclassifier. To find the best key parameters — $N\_estimators$ and $Max\_features$ — we first used random search to narrow down the range of each parameter, setting specific numbers of parameters for using grid search. Even with the best parameters as $N\_estimators = 700$ and $Max\_features = 3$, however, our model did not achieve the best result which random forest model normally does. In our case, the reason of low accuracy 72.5% came from that the one feature 'ExternelRiskEstimate' dominated other features.

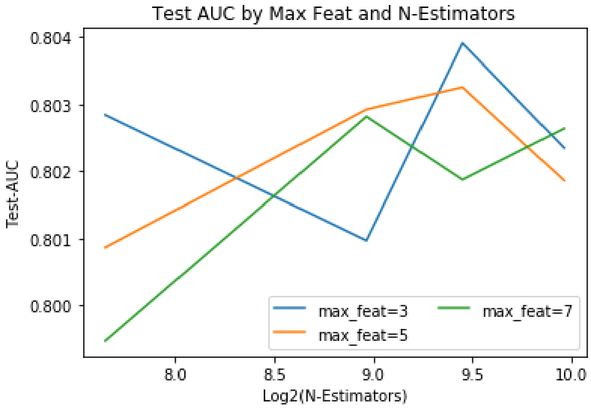


Figure 5: Feature Selection for Random Forest

### 2.c.3 Ada Boosting

Adaboost, short for Adaptive boosting, is a popular ensemble algorithm to convert a set of weak classifiers into a strong one. Adaboost generally performs well on balanced datasets, which is appropriate to use on our data set. We started with a model in default settings, then improved it by creating a grid search to tune the number of weak learners and the Learning rate. The best parameters in our model is when learning rate is 0.3 and the number of estimators is 100. Compared to the initial adaboost model, the results is improved from 71% to 72.7%, and the AUC is improved from 79.5% to 80.1%.

### 2.c.4 Neural Network

MLP, short for Multi-Layer perception network, is a feed forward artificial neural network. It can distinguish data that are not linearly separable. MLP is extremely tolerant to noisy data, and well-suited for continuous-valued inputs and outputs. On the other hand, it takes a long training time, and extremely difficult to interpret, due to the hidden units in the network.

We started with a baseline model without tuning any parameters, the resulting accuracy and AUC is 71.1% and 79.9%. Then, we created grid search to find the optimal combination of hyper-parameters by tuning:

- hidden layer sizes : The $i^{th}$ element represents the number of neurons in the $i^{th}$ hidden layer.
- activation: Active function for the hidden layer
- solver: The solver for weight optimization
- alpha: L2 penalty (regularization term) parameter.
- earning rate: Learning rate schedule for weight updates.

The optimal model is able to achieve an accuracy of 73.6%, and an AUC of 80.9%, with the best parameter combinations: activation = 'tanh', alpha = 0.0001, hidden layer sizes = (50, 50, 50), learning rate: 'adaptive', and solver = 'sgd'.

# 3 Evaluation

There are several ways to measure the performance of models, including accuracy, AUC, RMSE, and F-score ,etc. Since our project is a classification problem, we will mainly focus on the results of accuracy and AUC criteria. At first, we used 10-fold cross-validation to find the mean accuracy and standard deviation. However, using the accuracy to evaluate models is not credible since it makes no distinction between false positive and false negative errors. Hence, we also used AUC/ROC and expected value for evaluating our models (Table 1). Based on accuracy and in particular the AUC, the optimized SVM regression has the best score among all. Nevertheless, when considering real-world profit-and-loss expectations, Neural Network seems to beat all the others.

| Model | Accuracy | AUC | Expected Max Profit ($\$M$) |
|---|---|---|---|
| Baseline | 70.4% | 70.6% | 0.53 |
| Default Logistic Regression | 72.9% | 80.7% | 1.11 |
| Best Logistic Regression | 73.1% | 80.7% | 1.11 |
| Default SVM | 72.9% | 80.1% | 1.15 |
| Best SVM | 73.3% | 81% | 1.16 |
| Improved Baseline | 70.4% | 77.5% | 0.92 |
| Decision Trees | 70.8% | 77.7% | 0.94 |
| Random Forest | 72.2% | 80.4% | 1.13 |
| Ada Booster | 72.7% | 80.1% | 1.12 |
| Neural Network | 73.3% | 80.6% | 1.15 |

Table 1: Accuracy,AUC and Max Profit (after KFold X-Val)

## 3.a Explainability-Complexity Trade-off

In addition of these factors, it is also crucial to include an analysis of the trade-off effect between model complexity and explainability: after all, the model would mean nothing to a firm's board of directors if none of the employees can explain it well within 10 minutes. To illustrate, we have the following plot of mean accuracy(Figure.7):

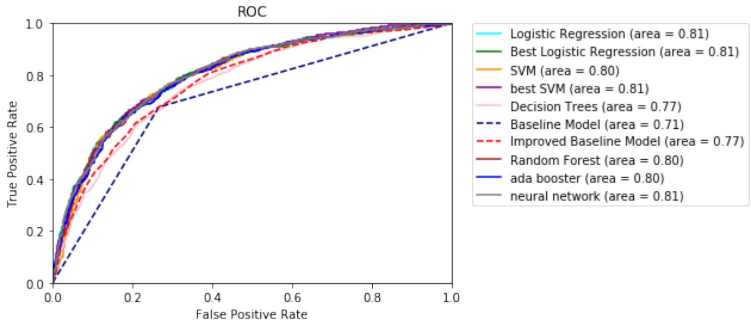


Figure 6: ROC Curve

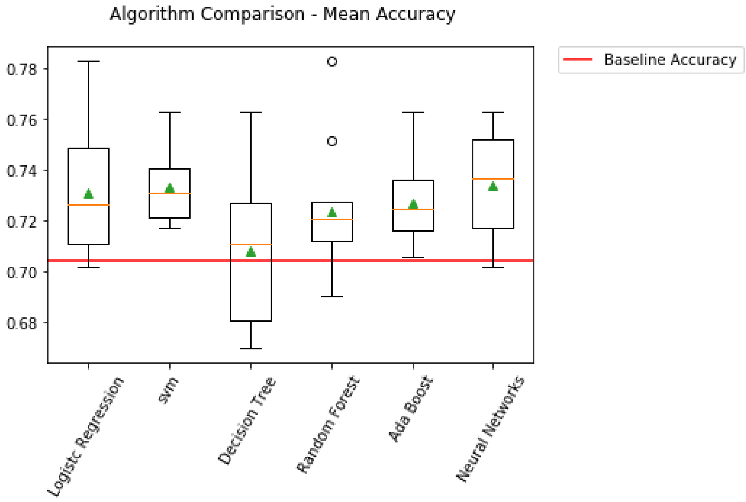


Figure 7: Mean Accuracy for all models

For linear models, we could see they spreads out more than black box models. Here, we guess the cost of explainablity would be high variance but there is some uncertainty since we did not validate the variance-bias trade-off between models due to a lack of time. However, one thing for certain is, same as the bias-variance trade-off, we should find the best model based on explainability-complexcity trade-off. Based on the results among explainable models (which are logistic regression and decision tree), our best model is logistic regression.

## 3.b Best explainable model

A major advantage of logistic regression is the high explainability from interpreting coefficients. As shown in the coefficient plot above, the logistic regression classifier with lasso regularization automatically does feature selection on the data set. Here we see 4 of the coefficient estimates are exactly 0, which are number of trades that are 60 days past due, number of trades that are 90 days past due, total number of trades, and number of inquiries over last month(excluding 7 days).

On the other side, predictors with high coefficients are worthy of attention. From the current data set and model, key features that financial institution should keep an eye on include (top reasons for turning down an application):

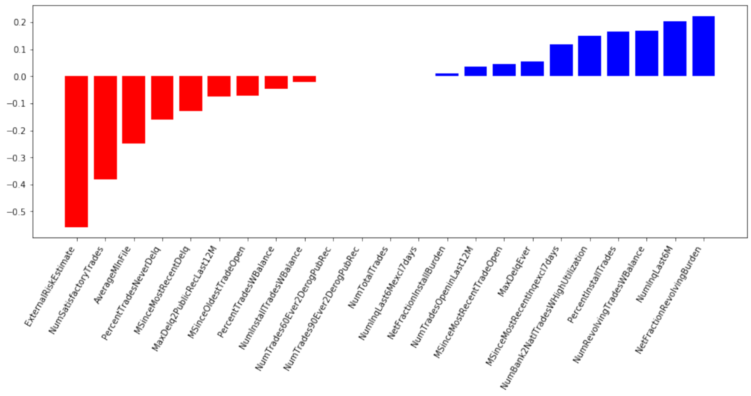


Figure 8: Coefficient plot for Logistic Regression

- External Risk Estimate (negative coefficient): Credit score generated by other system/institutions, the lower the external risk estimate value, the more likely the applicants will default.
- Number of satisfactory trades (negative coefficient): number of on-time payments. Increasing number of on-time payments would decrease the likelihood of default.
- Average months in file (negative coefficient): people with longer credit history are less likely to default.
- Revolving burden (positive coefficient): Revolving balance divided by credit limit, which measures the relative amount unpaid at the end of a billing cycle. People with higher revolving burden tend to fail in repaying within two years.
- Number of Inquiries last 6 months (positive coefficient). Applicants with more pull records indicates more frequent loan inquiries. Those people are more likely to default.

The magnitude of the coefficient tells how much influence the feature has on the outcome. Specifically, for example, 1 standardized unit decrease in the number of satisfactory trades increases the odds that the target will default by 38%.

## 4 Profit and Loss Analysis

Another way to evaluate the model is expectation analysis: which model would enable a bank to be more profitable, provided with a reasonable budget? To proceed with the analysis, several assumptions are made:

- Bankers work with a flat rate of $30 dollars per hour, regardless of his/her seniority
- Current lending rate APR is 5.5% (2)
- Current yield for 2 year US Treasury Bond is 2.18% (3)
- Monthly re-payment

- Prepayment risk is not taken into account
- The data contains loan applications between $\$5,000$ to $\$150,000$.For simplicity, we assume each applicant is looking for a loan of $\$77.5K$

Next, each fraction of the confusion matrix are assigned the following profit and cost scenario:

- True Positive Rate (TPR): High-risk people are correctly identified and are denied of their loan applications. In this case, there would be no major loss for the bank. There would be, however, a small labor cost associated with paying the bankers to work on reviewing the loans. Assume that one loan application would cost 2 hours to review. The cost would sum up to $\$30 * 2 = \$60$ for one application.
- False Positive Rate (FPR): Low-risk people are incorrectly identified as high-risk people and are denied of their loans. There would be no major cost to the bank except the labor fee of \$60 per application.
- True Negative Rate (TNR): Low-risk applicants are approved with their loans. Assume everyone is applying for a $\$77.5K$ loan with a 2 year fixed rate. Under the current market, the expected revenue is $\$82K$ at the end of the 2 years payment. In the meantime, the cost would be the borrowing cost for the bank on this loan amount. Assuming the bank is borrowing from the Fed (or another financial institution) with a current rate of 2.18% in accordance to the yield on 2 year treasury bond. The expected profit would be $\$2.7K$ after having each monthly profit discounted bank to the present. The profit per applicant would be $\$2.7K - \$30 * 2 * 2$ to account for the labor cost at the same time. We assume that it takes 4 hours to approve a loan.
- False Negative Rate (FNR): People with high financial risk are approved with their loan application. In the simple case, we also assume that such applicant will default before the first payment, meaning we will lose the entire notional ($\$77.5K$). In addition, we will need to pay back the interest to be compounded for the $\$77.5K$ notional borrowed from the fed (or the other institutions). However, since the loan is for homeowners, one can at least assume that the bank can get the house back as a collateral, liquidate the house and secure the most part of the notional (i.e., through foreclosure). It would be reasonable to assume that that it would incur a cost of 5% on the notional plus the labor cost for each applicant.

Combining the above scenarios to each threshold of the ranking, we have the following formula for expectation:

$$f(x, y) = -60y - 60x + (2694 - 30 * 2 * 2)(1 - y) - (60 + 77500 * 0.05)(1 - x) \tag{1}$$

where $x = TPR$ and $y = FPR$

After re-scaling the thresholds in proportion to the population size, we can fit every model to the expectation analysis (Figure 9 and 10).

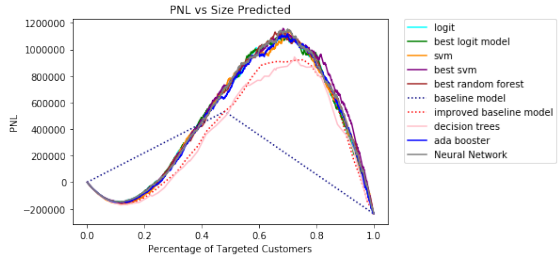


Figure 9: Expectation

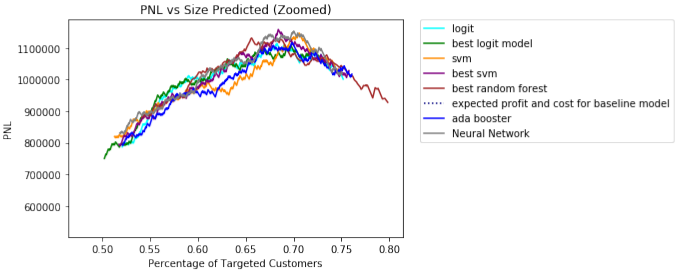


Figure 10: Expectation Zoomed

The graphs show an intriguing message: to reach the optimal benefit the firm need to target roughly the worst 70% of customers in terms of financial viability and only give the loans to the rest 30% of the applicants. Given unlimited budget (i.e., unlimited amount of bankers or extremely efficient systems), a firm can reach the optimal expectation by using tuned SVM (\$1.16$M$).

However, is it realistic to attain this \$1.16$M$ of profit? Note that the peak occurs at roughly 70% of the test set if one decides to deploy SVM, which means the bank has to review 593 applications in the bottom 30% (that are perceived as low risk) to obtain the full benefit. As mentioned before, each application would take 4 hours of work for it to be approved (reviewing the application, communicating with the clients, etc). Therefore, the bank need to devote $593 * 4 = 2372$ hours of manpower (or technology) to reach the goal. Since each hour would take typically \$30 dollars, the bank need to have a budget of \$71$K$. For medium-sized commercial banks that are usually specializes in sourcing fund, \$71$K$ is not very difficult to obtain. Moreover, if banks could seek ways to bring down the hourly rate (for example, outsourcing), the budget requirement could be even lower.

It is important to clarify that, in real-world deployment, the issue could be more complicated: loans of different sizes are usually involved; prepayment risk and re-financing could always happen, especially in a declining interest rate environment; not every bank could afford spending \$71$K$ in people/machines to reap the full benefit. Nevertheless, this profit and cost analysis could provide a guidance to business decision making, provided that proper financial models are used to refine the expected profit and a more precise budget is obtained.

## 5 Ethics, Risks, and Future Work

Since this data set does not include any demographic or confidential information, it should not cause any ethical problem. In terms of data sources, it is directly pulled from the credit bureau, which ensure the quality and objectivity of the data set.

Input predictors in the model should always be monitored to make sure there is no data leakage and implicit racial bias. In terms of potential risks, we would like to ask for more information about how external risk estimate score is computed, to protect the transparency of the entire machine learning system.

Further work on the project would be to improve the overall performance of the classifier. We do believe black-box models can achieve a performance with more data to train, further feature engineering, and in-depth fine tuning. In terms of explainability, instance-level visual explanations models could be utilized to interpret those black-box classifiers. In such case, lenders can potentially produce highly accurate and customized recommendations to applicants.

# A Appendix A

# References

[1] FICO Community, community.fico.com/s/explainable-machine-learning-challenge.

[2] Bank Rate,
`https://www.bankrate.com/home-equity.aspx`

[3] CNBC news,
`https://www.cnbc.com/quotes/?symbol=US30Y`

[4] Koehrsen, William. “Hyperparameter Tuning the Random Forest in Python – Towards Data Science.” Towards Data Science, Towards Data Science, 10 Jan. 2018, towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-scikit-learn-28d2aa77dd74.

[5] Moreira , João Mendes, and José Luís Borges. “Neural Networks and Ensemble Methods for Classification.” Porto.

[6] Provost, Foster, and Tom Fawcett. Data Science for Business. O’Reilly, 2013.

[7] Swalin, Alvira. “Choosing the Right Metric for Evaluating Machine Learning Models - Part 2.” Medium.com, Medium, 2 May 2018, medium.com/usf-msds/choosing-the-right-metric-for-evaluating-machine-learning-models-part-2-86d5649a5428.

[8] Zoldi, Scott, and Analytics Optimization. “How to Build Credit Risk Models Using AI and Machine Learning.” FICO® — Decisions, 23 May 2017, www.fico.com/blogs/analytics-optimization/how-to-build-credit-risk-models-using-ai-and-machine-learning/.

# B Appendix B: Data Dictionary

| 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 category | 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 Inqires Last 6 Months | Monotonically Increasing | predictor |
| NumInqLast6Mexcl7days | Number of Inqires Last 6 Months excl 7days.  Excluding the last 7 days removes inquiries that are likely due to price comparision shopping. | Monotonically Increasing | predictor |
| NetFractionRevolvingBurden | Net Fraction Revolving Burden. This is revolving balance divided by credit limit | Monotonically Increasing | predictor |
| NetFractionInstallBurden | Net Fraction Installment Burden. This is installment balance divided by original loan amount | 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 |

# C Contribution

## C.a SunJoo

- Write codes of data preparation
- Write codes and report of Random Forest and Evaluation
- Write explainability-complexity trade-off

## C.b Eva

- Write codes and report of Logistic regression, SVM, Ada Boost, and Neural Networks
- Write Best Explainable Model
- Ethics and Risks

## C.c Ruoyu

- Write code of data processing
- Feature Distribution
- Write Business Understanding and Data Understanding

## C.d Zijun

- Write codes and report of Baseline Model, Improved Baseline and Decision Tree
- Target Distribution
- Write Evaluation
- Profit and Loss Analysis