Model for Credit Approval

MOHAMMAD TOUTIAEE, NIKHIL RANJAN, KUSHAJVEER SINGH

HOSSEIN,NIKHIL.RANJAN,KUSHAJVEER.SINGH@UGA.EDU

CONTENTS

Abstract

Executive Summary

The following report was created over the course of developing a fair functional model to predict the risk of default in customers seeking a credit application using data provided by Wells Fargo in the Data Science Competition at the University of Georgia. The triplet sets of training, validation and test sets contained 28000 observations and 20 predictors. Of these predictors, the binary response for the bank XYZ was given to indicate applicants that already have some accounts with the bank. The remaining variables were cleansed and then transformed to be prepared for modeling process. The odds ratio of *Defaulted* account and the log function of the ratio were created to select the best predictors via a logistic regression. After an exhaustive search among 524287 models, a BIC of 3262.076 was obtained to the best 5 variables. The selected predictors were trained by a logistic regression model and random forest and validated to predict the target response. The logistic model and random forest obtained respectively the balanced-accuracy of 0.71 and 0.78. 5 fairness metrics were investigated for both models to prevent discrimination against *young* applicants via controlling "card_age" that appeared as a proxy variable for age. After debiasing the logistic regression and random forest by reweighing method, they obtained balanced-accuracy of 0.73 and 0.79, respectively, without any symptoms of "age" discrimination. Random forest was selected for modeling, and the question of whether or not customers of the bank XYZ would receive favourable treatment by the model was thoroughly investigated. Both effects of "ind_acc_xyz" and "uti_card" were controlled to prevent *aggregation bias*. Two more additional analyses were performed to further explain why an application was rejected. LIME toolset was utilized for a local explanation on a rejected applicant, and "Contrastive Explanation Method" was applied to identify features that are (minimally) absent in the input instance, but whose presence would have altered the classification.

I. Introduction

With the great influence of machine learning techniques, many financial institutes rely on such models to minimize risks and maximize benefits for creating a Fair and Robust Pipeline to evaluate customers' credits. The three main goals of the Fair and Robust Pipeline are to build accurate models, meet the governments' regulations and reduce costs. Through this, the act aims to improve the quality of decision-making system in the US.

This project aims to explore such pipeline through designing a process and interpreting the results to enable Wells Fargo achieve the aforementioned goals easily. One specific way Wells Fargo can improve modeling quality and decrease risks is by developing models that review credit card applications to determine which applicant should be approved. This project proposes a credit card application framework that predicts applicant's request using historical data. The denial/approval of a credit request depends on the various factors governed by a certain number of criteria. Applicants with high/low probability of approval are subject to further investigation as to ensure the required regulations are met.

II. Pre-processing

A. Exploratory Data Analysis

We explored the data by checking the type of data and discovered that there are 20 predictors with one target variable (Indicator of Default). The target variable is a binary response where 1 denoting an account defaulted after an account was approved and opened with bank XYZ in the past 18 months, and 0 representing not defaulted. Here "Default" means no payments for 3 consecutive months. Of those 20 predictors, the feature "State" is the only categorical variable, containing data for 7 different states. Our goal is to predict this target variable using other predictors in the set.

We observed that in the dataset respectively 92% and 8% of the applicants are 0 (non-defaulted) and 1 (defaulted). The disproportion between 2 classes can impact the model performance, and we address it in the training step to prevent extra bias injecting by the dominant class.

B. Missing Values and Outliers

We investigated the dataset for missing values, and we found 2 features "uti_card_50plus_pct" and "rep_income" respectively contain 10% and 8%, which we imputed by median. eda html. We also identified outliers in "avg_card_debt" at which 6% and 0.6% were for non-defaulted and defaulted classes, respectively. The effect of outliers were eliminated by standardization explained in section D.

C. *Feature Selection*

Typically, we are interested in selecting features that are distributed significantly different between the two classes. We performed an exhaustive search to identify the best features through all different combinations among 20 predictors or 524287 different models, and we obtained the lowest BIC of 3262.076 for the following features:
"avg_card_debt", "card_age", "non_mtg_acc_past_due_12_months_num", "inq_12_month_num" and "uti _card". We preferred BIC to AIC in most cases, since SC (BIC) yields a simpler model as Stepwise Logistic Regression, while AIC leads to an over-parameterized model. table AIC/BIC feature.

D. *Data Prep*

All the selected features were numerical, and we standardized them using "`MinMaxScaler`" from "`Scikit-Learn`" to address outliers and skewed distributions.

III. Modeling

A. *Why We Chose Random Forest?*

There are three machine learning models suggested for this competition to compete with Logistic Regression: 1) Random Forest, 2) Gradient Boosting and 3) Feed-Forward Neural Networks. Although all the aforementioned models are powerful and widely used in many problems, we chose Random Forest because it is easier to tune and harder to overfit compared to other two models.

We trained a Logistic Regression model over the training set using the best features, and we chose Random Forest as the competitor model for comparison. The Both models' hyper-parameters were optimized over the validation set and the results were provided in table for LR and RF performance. We used "undersample" technique to tackle with the class-imbalanced issue by only taking a random sample of observations of the majority class (i.e. non-defaulted) equal to that of the minority class (defaulted), so that we have a balanced data set. The training set size under this approach was 3172 observations in which two target classes were equally distributed.

B. *Hyper-parameters Tuning*

`Scikit-learn` provides tools to tune the hyperparameters for our models. We used `RandomizedSearchCV` first initialize the hyperparameters to establish a narrower range before implementing `GridSearchCV`. Additionally, `StratifiedKFold` was used to ensure that the class ratio was the same for the folds used in cross-validation.

C. Evaluate Metrics; Type I & II Errors

We evaluated our model by AUC scoring and balanced-accuracy ((specificity + Sensitivity)/2) metrics. Additionally, we used a confusion matrix to summarize the actual vs. predicted labels, and the ROC curve where the X axis is the false positive and the Y axis is the true positive rates. (auc table and confusion matrix).

Ideally, a perfect predictive model would be a diagonal matrix where values off the main diagonal, indicating incorrect predictions, would be zero. There is hardly any ambiguity surrounding this matrix as far as statistical hypothesis testing is concerned. However, there is a debate over Type I and Type II error concepts in the context of credit assessment. By definition, credit assessment is a process performed by financial institutes and other lenders to evaluate a potential borrower's creditworthiness. Accordingly, there are two types of errors: False acceptance of a bad applicant and False rejection of a good applicant and therefore, Type I and Type II errors are proposed by two opposite views:

- View 1: states that Type I error is defined as good credit misclassified as bad credit and Type II error is defined as bad credit misclassified as good credit, and Type II error is costlier than a Type I error. This view is supported by [1], [2], [3].
- View 2: supported by [4], [5], [6] states that the misclassification cost of a Type I error is more than that of a Type II error. According to this view, Type I error happens when a borrower is incorrectly deemed creditworthy, when in fact, the institution should not give the borrower a loan. Type II error happens when a financial institution denies a loan to a creditworthy borrower.

Thus, the definition of Type I and Type II errors in this context should be defined by the lenders. We optimized the cut-off point of both classifiers under the balanced-accuracy function, since this metric reinforces equal policy on both types of errors.

IV. Results

We trained the proposed models and evaluated them on the test set to obtain the performance metrics, and the results were provided in table for LR and RF.

A. Logistic Regression

The Logistic Regression generated results with the balanced-accuracy of 0.73 and 0.71 for training and test sets, respectively with the decision boundary at point 0.37 for classification. The cut-off point was optimized over the validation set, testing at which point the balanced-accuracy was maximized. The full details can be found in table for LR and RF. The final model with estimated coefficients obtained from Logistic Regression is as follow:

$$
\begin{aligned}
intercept &= -3.42 \\
non\_mtg\_acc\_past\_due\_12\_months\_num &= 7.52 \\
uti\_card &= 5.52 \\
avg\_card\_debt &= 2.75 \\
inq\_12\_month\_num &= 1.32 \\
card\_age &= -0.75
\end{aligned}
$$

The estimated coefficients indicate that the model assigned the most and least weights on respectively *non_mtg_acc_past_due_12_months_num* and *card_age*, where the latter one appeared with a negative direction. So, 4 positive coefficients impact on the prediction of *Defaulted* positively, where the odds of being defaulted account are multiplied by 0.45 by the variable *card_age* ($e^{-0.75} = 0.45$), if *card_age* increases by 1.

B. Random Forest

Random Forest appeared to be a better model with the balanced-accuracy of 0.78 on both training and test sets, due to the fact that this model benefits from a complex training. The cut-off decision boundary obtained by Random Forest is larger than Logistic Regression ($0.57 > 0.37$), implying that this model leniently classifies accounts as defaulted.

1) Gain-based Feature Importance: The feature importance of Random Forest provides us with an idea of the factors that are potentially impacting credit applications to be approved. The feature importance in Random Forest is calculated through *Information Gain* which is commonly used in the Tree-based algorithms. The Gain function implies the relative contribution of the corresponding feature to the model calculated by taking each feature's contribution for each tree in the model. A higher value of this metric when compared to another feature implies it is more important for generating a prediction.

As the feature importance RF orig shows, the most important feature for the Random Forest model is *avg_card_debt*. The higher the average debt, the higher the probability of the applicant being defaulted. Unlike Logistic Regression model, the most important feature claimed by Random Forest is *non_mtg_acc_past_due_12_months_num*, while they both agree on the variables *card_age* and *inq_12_month_num* to be least and second to least important in prediction, respectively. The red color in feature importance

RF orig demonstrates a negative impact on the prediction of defaulted, which is also evidenced by Logistic Regression.

2) Partial Dependence Plots: We also investigated the Partial Dependence Plots (PDPs) to explain individual and interaction predictions using the approach explained in [7]. A PDP plots the change in the average predicted Y as X varies over its marginal distribution. This is generated by producing a prediction for Y for each observation in the training set for some (perturbed) value of X, keeping other predictors as they are, then averaging the results. This mechanism is applied for the range of values of interest for X. The plot is then generated through the averaged predictions over the range of values of X. The PDPs for 5 individual predictors and all pairwise interactions are provided in 1-d PDP orig and 2-d PDP orig, respectively. pick example from 1-D and 2-D PDP orig.

V. Model Debiasing

Federal laws[1] prohibit discrimination in credit application on nine protected bases, and *age* is one of these prohibited attributes. Although the variable “age” does not exist in the dataset alone, the variable “card_age” is so highly correlated with “age” that it serves as a *proxy* for the age, and it is among the best selected features for model training.

Based on our model, “card_age” plays a significant role in the prediction of defaulted/not-defaulted account, because the training dataset happened to have better repayment for one age group than for another. This raises two problems: 1) the training dataset may not be representative of the true population of people of all age groups, and 2) even if it is representative, it is illegal to base any decision on a applicant’s age, regardless of whether this is a good prediction based on historical data. This incident may raise an unintentional discrimination, imposing burdens against the sub-populations in the prediction based on age group.

The bias was measured in the model plots for showing biases (5 bars) based on 5 most usable fairness functions, including disparate impact, average odds difference, statistical parity difference, equal opportunity difference and Theil index. The formulas of these metrics are as follow:

Disparate impact uses the formula

$$\frac{Pr((Y = 1|D = unprivileged))}{Pr((Y = 1|D = privileged))}$$

and a fair model yields 1.

Average odds difference equals to:

$$\frac{(FPR_{unpriv} - FPR_{priv}) + (TPR_{unpriv} - TPR_{priv})}{2}$$

[1] https://www.fdic.gov/regulations/examinations/supervisory/insights/sisum05/sisummer05-article3.pdf

and it must be close to 0 for a classifier to be fair.

Statistical parity difference measures the bias between the probability that a random instance drawn from unprivileged is labeled 1 (or defaulted) (so here an applicant is young) and the probability that a random instance from privileged is labeled 1, and a fair model must produce zero (or near to zero).

Equal opportunity difference is just a difference between the true positive rate of unprivileged group and the true positive rate of privileged group, and the formula is as follows:

$$TPR_{unpriv} - TPR_{priv}$$

and this expression is equal to 0 for fair models.

Theil index which is also known as the generalized entropy can be measured via:

$$\frac{1}{n}\sum_{i=0}^{n}\frac{b_i}{\mu}ln\frac{b_i}{\mu},$$

where $b_i = \hat{y}_i - y_i + 1$. Similar to other fairness functions, this expression must be close to 0 for fair models.

We alleviated the injected bias to enforce the predictions from trained models to be generated discrimination free via *reweighing* technique [8] (pre-processing algorithm), and the results were provided in debias table.

A. Debiasing Implementation

For each dataset, the protected attribute “card_age” constructed as follows: ’Old’ (privileged class) defined by the features card_age $\geq$ median (Old) and card_age $<$ median (young). The median cut-off point was selected for identifying the privileged class based on the maximum separation between two distributions of defaulted and not-defaulted accounts.

Disparate impact (DI) is defined as the probability of favorable outcome for unprivileged instances over the probability of favorable outcome for privileged instances, and a dataset with the DI of 1 implies that there is no bias between privileged and unprivileged classes. The DI of 0.482 was calculated in the training set for the variable “card_age”, implying that prediction in defaulted account would adversely affect the young applicants.

The plot LR and RF orig plot disparate shows $1 - min(\text{disparate impact}, 1/\text{disparate impact})$ $(1 - min(DI, 1/DI))$ against the classification threshold and balanced-accuracy for Logistic Regression and Random Forest models. We plotted $1-min(DI, 1/DI)$ since it was possible to overcorrect and produce values greater than 1, indicating unfairness for the original privileged group. We aimed to adjust this metric to be less than 0.2, according to fairness literature, to make the trained models discrimination free. Similarly, the average odds difference was plotted against the classification threshold and balanced-accuracy for both trained models LR and RF orig plot avg odds, and this metric must be close to zero for a classifier to be fair. The analysis for the rest of fairness metrics were provided in all metric table. All the provided metrics indicate unfair predictions against unprivileged group in both Logistic Regression and Random Forest, encouraging us to correct the situation by reweighing method. We aimed to mitigate the bias by transforming the data, and both classifiers were evaluated by those 5 fairness functions over the test set before and after transformation via reweighing algorithm, and the results were provided in all metric table. After applying reweighing method on the training set, we obtained a disparate impact of 1, showing no bias in the data. Of both models, the Random Forest model exhibits the best balance in terms of balanced accuracy and fairness ($0.79 > 0.73$). The logistic model is slightly unfair compared to the random forest model (larger Teil Index, statistical parity difference, disparate impact and average odds difference values). Hence, we chose the random forest model as the winner in this case.

VI. XYZ Bank; Simpson Paradox

Of interest to us was to investigate whether customers who already have an account with the bank XYZ receive any favourable treatment in our trained model. We answered this question via analyzing 2×2 contingency tables via Logistic Regression, which the model is written as:

$$ln(\frac{p}{q}) = 0.7 + 0.14 \times I_{XYZ},$$

where the *bank XYZ* coefficient is significantly positive (relative to the non-account holder baseline level) in our random forest model. Thus, this model indicates that if a customer has an account with the bank XYZ, the odds of being a defaulted account multiplied by $e^{0.14}$ (or 1.15) compared to a non-account holder. One advantage of using Logistic Regression over other 2×2 table tests is we can investigate whether the 2×2 table has been collapsed, so that there are really two or more explanatory variables in play. In this case, our goal is to know if the the aggregation of the variables into one explanatory variable is hiding some other explanatory effect, referring to as “aggregation bias” or “Simpson’s Paradox”. This might be shocking to the bank XYZ when they figured that their customers tend to be more careless about their payments, which might damage its reputation. The way we answered the above dilemma was to perform a Logistic Regression, controlling for one of the five input predictors and XYZ effects by binning the lurking variables into 5 bins. We performed this standard statistical approach via the logistic-scale grand mean parameterization (i.e. forcing the two XYZ effects to sum to zero and also forcing the 5 bin effects to sum to zero). After testing all the models, the final logistic regression model looks as shown in the Table I.

TABLE I: Aggregation Bias Detection by Controlling for Important Factor

	Estimate	Std. Error	z value	Pr(> $\|z\|$)
(Intercept)	0.8964	0.0402	22.30	0.0000
ind_acc_XYZ1	-0.0554	0.0389	-1.42	0.1548
bin_uti_card1	1.0944	0.0824	13.28	0.0000
bin_uti_card2	0.8023	0.0758	10.59	0.0000
bin_uti_card3	0.1276	0.0654	1.95	0.0511
bin_uti_card4	-0.3746	0.0615	-6.09	0.0000
bin_uti_card5	1.64972			
BIC = 76.05				

This model obtained the lowest BIC value of 76.05 compared to other 4 models, and the Likelihood Ratio Test of Nested Models indicated that the model with the lurking variable "uti_card" fits significantly better than the null model (without "uti_card"). The BIC values of all models were provided in the table for BIC models XYZ.

Table I and plot xyz display the idea, in logit scale, of the effects of the different variables. The estimated XYZ1-XYZ0 difference, in logit scale, is about 0.11, which is not statistically significant. According to Table I and table for xyz percentage, **Card Utilization** is a very significant factor, with bins **1**, **2** and **3** being very easy to be predicted as a "good" customer, but bins **5** being very difficult to be classified as good (lower not defaulted rate). The final conclusion to be drawn from all of this is that if **uti_card** had not been controlled for, one would incorrectly conclude that there had been massive discrimination against the bank XYZ's customers, when, in fact, there was no evidence at all to support this claim.

VII. Rejection Scenario

A. *Why was one's application rejected?*

Of interest to applicants is they want to know *how* and *why* the decision was made to reject their credit application. The explanation given will help them understand if they have been treated fairly, and also provide insight into what – if their application was rejected – they can improve in order to increase the likelihood it will be accepted in the future. To answer why an application was rejected, we use an LIME [9], an interpretable toolset, to explain the decision at local level.

B. LIME

LIME is short for Local Interpretable Model-agnostic Explanations. Local refers to local fidelity, where we want the explanation to be reflective of the behavior of the underlying model in the vicinity of the instance being predicted. LIME relies on the assumption that every complex model is linear on a local scale. So, we usually expect that similar observations behave predictably even in a complex model, and it enables LIME to fit a linear regression model around a single observation that will mimic how the global (underlying) model predicts at the locality.

Theory: LIME provides explanation by assuming that the relationships between variables in a complex model can be estimated linearly. This encourages LIME use weighted OLS, and the following equation should be calibrated to generate an explanation:

$$\begin{aligned}\xi(y) &= arg\min_{\beta}\left\{\sum_{i=1}^{N} exp(-\frac{D(y,z)^2}{\sigma^2})(f(z_i)-\beta z_i')^2+\Omega(g)\right\},\\ \Omega(g) &= \infty\mathbf{1}_{\{||\beta||_0>K\}} \equiv \begin{cases}\infty, & \text{if } ||\beta||_0 > K\\ 0, & \text{if } ||\beta||_0 \leq K\end{cases}\end{aligned}$$

where $\Omega(g)$ is a regularization term that measures the complexity of the explanation $g \in G$, and $||\beta||_0$ denotes the number of non-zero coefficients of β. The parameter K determines the explanation task, specifying what number of variables should be reported to the user. This equation should be approximated by selecting a subset of K features using a feature selection technique and then estimating the coefficients via weighted least squares.

We identified one instance in the test set that was predicted as the defaulted (or bad) account, and we ran LIME on it to provide an explanation. The Table II demonstrates how LIME predicted that customer on a local scale when their application was classified as "bad" (or defaulted) by our fair model. the plots shows that variables "non_mtg_acc_past_due_12_months_num", "inq_12_month_num" and "uti_card" positively contributed to the class of being rejected, and variables "avg_card_debt" and "card_age" negatively affected the prediction. We also observe that the effect of "card_age" which serves as a proxy for the applicant's age is the weakest compared to other effects. The local explanation generated by LIME for our case is: The probability of class for this particular application was 0.974. Since this probability was greater than the decision threshold (0.5758), so the model classified it as "rejection". In

TABLE II: LIME Explanation for Class "Rejected" of a Defaulted Applicant

	non_mtg_acc_past_due_12_months_num	avg_card_debt	inq_12_month_num	uti_card	card_age
Coefficient	0.40	-0.21	0.07	0.06	-0.03

order to increase the likelihood of acceptance and explore how the decision could have been different through changes to the profile, the odds should be dropped by 40.91% to be in the “good” region. So, based on the information provided by Table II, if the variable non_mtg_acc_past_due_12_months_num decreases by 2 via removing delinquent payments, the odds of rejection (i.e. 0.974) are multiplied by 0.58 (or $e^{0.40\times(-2)}$), yielding 0.36 which would be less than the threshold of 0.5758.

VIII. Model Use Case

In order to demonstrate how our model can be utilized to score credit application, we suggest the following steps:

- A data scientist develops a ’fair’ credit assessment scoring model with respect to defined protected classes. Fairness specialist should ensure the model’s attributes are complied with government regulations and credit decisions be not predicated on prohibited factors.
- A developer takes the model and performance characteristics (e.g. accuracy, fairness tests, etc.) and deploys the model in an enterprise app that evaluates applicants for credit assessment.
- The app is deployed for production and starts scoring applicants and making recommendations.
- Explanations are generated for each recommendation.
- Both recommendations and associated explanations are given to loan officers (a person who makes the final judgement) as a part of the credit assessment process. The loan officers can evaluate the recommendations for quality and correctness and provide feedback.
- Loan officers’ feedback as well as analysis of usage data with respect to specs of the model with respect to accuracy and fairness are communicated to engineering department.
- When a significant deviation in the model specs relative to the model factsheet is observed, the model is sent back for retraining.
- All the steps above can be repeated periodically.

IX. Conclusions

We developed a fair and accurate credit model for the data science competition at the University of Georgia Our model can help loan officers reduce the cost of risk by identifying qualified applicants. EDA was implemented on the data and identified the best features among 20 predictors via comparing many sub models. We trained two model, logistic model and random forest, and performed comparisons with respect to model performance and fairness. The random forest model was chosen over logistic regression, because it achieved the best performance without any signs of discrimination. The bank XYZ account holder was evaluated in the winner model to identify if the model treats them differently. We answered this question through an extensive analysis and discovered a lurking variable to prevent incorrect conclusions when aggregation bias existed. We also described how a case predicted as defaulted by our model can be explained to the applicant by LIME, and we suggested how the applicant could improve their credit application to be accepted by our model. Finally, the model use case was described in the report and the major steps for deployment were provided.

Our modeling process allows financial institutes to provide a better quality of lending products to their customers, reduce the risk of unpaid balance and also meet the government regulations.

References

[1] O. Adasme, G. Majnoni, and M. Uribe, *Access and risk-friends or foes? Lessons from Chile*. The World Bank, 2006.

[2] X.-L. Li and Y. Zhong, “An overview of personal credit scoring: techniques and future work,” 2012.

[3] A. M. Kern, “Credit score analysis,” 2017.

[4] J. B. Caouette, J. B. Caouette, E. I. Altman, P. Narayanan *et al.*, “Managing credit risk: the next great financial challenge,” 1998.

[5] S.-W. Shen, T.-D. Nguyen, and U. Ojiako, “Modelling the predictive performance of credit scoring,” *Acta Commercii*, vol. 13, no. 1, pp. 1–12, 2013.

[6] V. Limsombunchai, C. Gan, and M. Lee, “An analysis of credit scoring for agricultural loans in thailand,” 2005.

[7] J. H. Friedman, “Greedy function approximation: a gradient boosting machine,” *Annals of statistics*, pp. 1189–1232, 2001.

[8] F. Kamiran and T. Calders, “Data preprocessing techniques for classification without discrimination,” *Knowledge and Information Systems*, vol. 33, no. 1, pp. 1–33, 2012.

[9] M. T. Ribeiro, S. Singh, and C. Guestrin, “” why should i trust you?” explaining the predictions of any classifier,” in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135–1144.

Appendix