STAT 3612 Final Report

FICO HELOC Interpretable Machine Learning

Statistical Contemplation of Financial Data Modelling

GROUP 3612202004

Choi Suhyun (3035554658)

Chong Inbum (3035553355)

Kathleen Low Zi Yi (3035549017)

Kim Sunghyun (3035603526)

Beatrice Wong Chi Kan (3035374436)

Abstract

Due to the nature of the financial market, any decision process within the machine learning model has to be explainable. This report offers the overall contemplation of widely-used machine learning techniques in terms of model performance interpretability. It explores the two sets of models: with and without monotonicity constraints. We apply the models to the FICO HELOC dataset, which divides the group of people based on the credibility criteria.

Table of Contents

Introduction

Home Equity Line of Credit

A Home Equity Line of Credit , or HELOC is a loan in which the lender agrees to lend a maximum amount within an agreed period of time. The HELOC data contains anonymized credit applications of HELOC credit lines. It introduces a dataset containing 23 features influencing the responsible variable, 'RiskFlag'. Among 10,459 observations, 5,000 of them are classified as 'Good' which means that clients repaid their HELOC account within 2 years. Rest of them are classified as 'Bad'.

Credit scoring is a statistical analysis for banks and other lenders to perform before deciding on extending or denying a person's credit. Due to the increasing number of applications for loans received on a daily basis, it becomes imperative to come up with a model to decide if a person is risky or not.

The Importance of Interpretability in Financial Data

In financial data like HELOC, the interpretability of a predictive model is crucial. The interpretable models give causal relationships between input data and conclusions. Hence, the model can explain why a certain person can not get a loan. Compared to a blax box, interpretable models reduce the human's suspicions on the system's conclusion. Companies can easily make decisions with trustworthy systems and customers can easily accept the results.

The Needs of Monotonicity Constraints

The monotonic function is a function that is either not entirely increasing or not entirely decreasing. With monotonicity constraints in models, the oscillatory behaviour would be removed. Companies often implement monotonicity constraints in their predictive models to make logical and ethical decisions. Without monotonicity constraints, they may have wrong decisions such that a person with lower credit still gets approved.

Data Preprocessing

Missing Value Imputation

As the missing values in the dataset can have effects on the conclusion, we imputed missing values by the property of each feature. The missing values in the dataset are recorded as '*-7 (Record or No Investigation)*', '*-8 (Usable/Valid trades or inquiries)*' and '*-9 (Condition not met)*'. Initially, the missing values are replaced with NaN values.

By plotting the histograms , distributions and box plots of each features, we considered '*x2 (Months Since Oldest Trade Open)*', '*x5 (Number of Satisfactory Trades)*', '*x8 (Percent Trades Never Delinquent)*', '*x13 (Number of Trades Open in Last 12 Months)*', '*x18 (Net Fraction Revolving Burden)*' and '*x20 (Number of Revolving Trades with Balance)*' as skewed distributions. Therefore, the missing values of these 6 features are imputed with the median of each feature.

For the distributions of feature '*x10 (Max Delq/Public Records Last 12 Months)*' and '*x11 (Max Delinquency Ever)*', they were mostly based on the feature's mode. The missing values of these 2 features are imputed with the mode of the feature. For other features, the missing values are imputed with their medians.

Outlier Analysis

Including the outliers into the model may reduce the predictability of models. As the box plots show too many outliers in the dataset, we choose to conduct interquartile range (IQR) analysis. We considered the observation that is outside of 3 standard deviations from the mean as an outlier. In the training set, 16 observations are considered as outliers.

Correlation Analysis

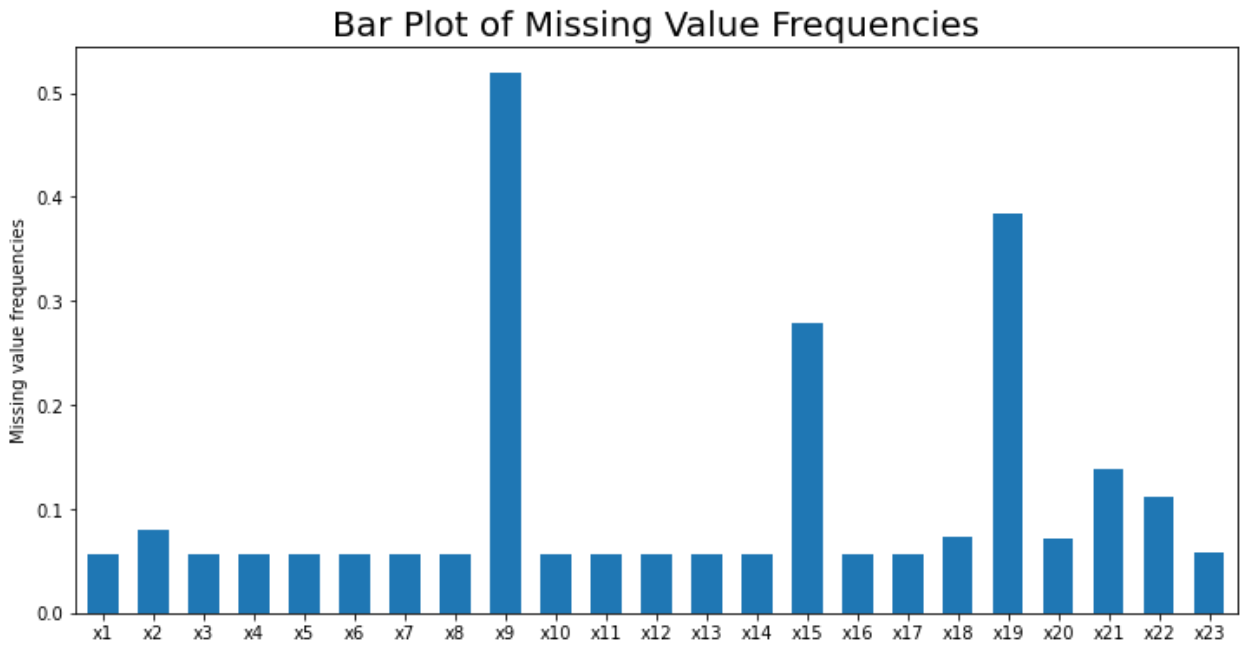
Existence of strong multicollinearity may impact the performance of models. From the heat map, the correlation between features are analyzed. The feature '*x6 (Number of Trades 60+ Ever)*' and '*x7 (Number of Trades 90+ Ever)*' are highly correlated (0.89). Also, '*x16 (Number of Inq Last 6 Months)*' and '*x17 (Number of Inq Last 6 Months excl 7 days)*' are extremely highly

correlated (0.99). In general, *'x1 (Consolidated version of risk markers)'* is highly correlated with other features among all features.

Analysis of Individual Features

We plotted distributions of each feature grouped by RiskFlag to intuitively explain each feature. For *'x6 Number Trades 60+ Ever'* and *'x7 Number Trades 90+ Ever',* people with bad risk had lower number of trades than people with good risk. For *'x10 Max Delq/Public Records Last 12 Months'* and *'x11 Max Delinquency Ever'*, people with good risk had less delinquent trades, which were mostly 'unknown','current and never delinquent', etc. For *'x18 Net Fraction Revolving Burden', 'x22 Number Bank/Natl Trades w high utilization ratio'* and *'x23 Percent Trades with Balance',* the people with good risk had lower fraction of revolving burden, lower number of trades with high utilization ratio and lower percent trades with balance.

Feature Selection and Its Influence on Model Implementations



Based on the missing value frequencies of *'x9 (Months Since Most Recent Delinquency), 'x15 (Months Since Most Recent Inq excl 7days)'* and '*x19 (Net Fraction Installment Burden)'* are significantly higher than other features, they are excluded in our model.

From the outlier analysis, 16 observations that are considered as outliers in the training set are removed.

From the heat map, Since '*x6 (Number Trades 60+ Ever)'* and '*x7 (Number Trades 90+ Ever)'* are highly correlated (0.89), and '*x16 (Number of Inq Last 6 Months)*' and '*x17 (Number of Inq Last 6 Months excl 7days)'* are extremely highly correlated (0.99), one feature from each pair is dropped.

Therefore, the dropped features are '*x7 (Number Trades 90+ Ever)', 'x9 (Months Since Most Recent Delinquency), 'x15 (Months Since Most Recent Inq excl 7days)'* , '*x17 (Number of Inq Last 6 Months excl 7days)'* and '*x19 (Net Fraction Installment Burden)'* .

Subsidiary Processes of Data Preprocessing

The two classes 'Bad' and 'Good' are replaced by binary numbers, 0 and 1 respectively.

As *'x10 Max Delq/Public Records Last 12 Months'* and *'x11 Max Delinquency Ever'* are categorical data, we encoded them. Also, we scaled the data to obtain better performance for some models.

Models Without Monotonicity Constraints

Model Selections

We selected popular models that we felt could adequately predict the test set. Six for the non-monotonicity constrained model and 3 for the monotonicity model. To choose our final model, we tested the models on our test set to obtain an accuracy score, as well as generate a unique way to interpret the data for each model. The final model was chosen based on both the accuracy and the interpretability.

Logistic Regressions

Accuracy of logistic regression on the testing set = 0.7051

Accuracy of logistic regression with IV binning on the testing set = 0.7089

The logit regression result allows us to see the importance of each feature based on their z values. The probability of each feature to be relevant can also be seen with the p-values, and any feature with a high value can then be dropped.

General Additive Models

The Accuracy of B-Spline GAM model on testing set: 0.7094

The Accuracy of Piecewise ReLu GAM model on testing set: 0.7075

The pyGAM package comes with a built-in function to obtain the partial dependence plots for each feature, and from the plots we can see the contribution each feature makes according to the value of the feature. For example, RiskFlag depends more on x1 at higher values, as seen by the increasing partial dependence plots.

Support Vector Machines

The Accuracy of SVM model with Linear SVC on testing set: 0.7103

The Accuracy of Stochastic gradient descent version of SVM on testing set: 0.7108

The Accuracy of SVM model with RBF kernel on testing set: 0.7137

The Accuracy of SVM model with RBF kernel on testing set: 0.7137

The Accuracy of SVM model with RBF kernel with hyperparameter tuning on testing set: 0.7108

We can run post-hoc analysis on the model to obtain the partial dependence plots.

Decision Tree

The Accuracy of Decision Tree on testing set: 0.7089

A decision tree diagram can be printed out to analyse the cut-off for each feature on the branches.

Gradient Boosting: XGBoost

The Accuracy of XGBoost with high learning rate with its optimal estimator=34 on testing set: 0.71702

The Accuracy of XGBoost with low learning rate with its optimal estimator=130 on testing set: 0.71941

Various methods of interpretation can be obtained post-hoc, including partial dependence plots and SHAP values. As this is our chosen model, the interpretation will be elaborated at the end of the report.

Artificial Neural Network

The Accuracy of MLPClassifier with Piecewise ReLU on testing set: 0.7189

Similarly to XGBoost, we can obtain post-hoc analysis with SHAP values to find the feature importance or the influence of each individual value.

Models With Monotonicity Constraints

General Additive Model

To control the overfitting, we added monotonicity constraints to the logistic General Additive Model. By adding monotonicity constraints, we can draw more reliable conclusions.

As the monotonicity constraints in the data dictionary show the monotonicity constraint of each feature when the 'Bad' is encoded as 1, we interpreted them oppositely. According to

each feature's property, we added 'constraints = 'monotonic_inc" or 'constraints = 'monotonic_dec" in the spline terms. With the features that have no monotonicity constraints and categorical data, the factor terms are used.

The accuracy on the training set for GAM with monotonicity constraints is 0.7006.

The accuracy on the test set for GAM with monotonicity constraints is 0.7032.

Gradient Boosting: XGBoost

We have used a 5-fold cross-validation and early-stopping on the training dataset to determine the optimal number of trees. Then, we use the entire training set to train the model and evaluate its performance on the testset.

In `Xgboost`, `'monotone_constraints'` is where the monotonicity constraints are set. The values '-1', '0', '1' imply monotonically increasing, not monotonically constrained and monotonically decreasing respectively. Evaluation on the test set is based on `'error'`, which is one of the evaluation metrics of xgboost. Prediction accuracy is calculated using [1-error].

Relationship between RiskFlag and each feature variable can be shown in the partial dependence plot. To plot the partial dependence, we have created a function where a grid of values of a feature variable is sampled and for each value, every row of that variable is replaced with the sampled values.Then the function calculates the average prediction.

The accuracy on the training set for XGBoost with monotonicity constraints is 0.7285.

The accuracy on the test set for XGBoost with monotonicity constraints is 0.7141.

Monotone MLP

One of the important findings in order to apply monotonicity constraints in ANN is to guarantee that keeping all the weights in the model non-negative for case of increasing constraint (Zhang and Zhang, 1999). The 'monmlp' package was constructed based on the modified feedforward network structure suggested by Zhang and Zhang in 1999.

In their network structure, they transformed the weight into the exponential form, which does not take negative value into account. Furthermore, it avoids the issues of domain restrictions. They derived the gradient calculation of network using transformed weights to prove the network structure in monotone.

In order to apply the idea into our HELOC binary classification problem, we have transformed the predictive values into sigmoid functions, and regarded any values below 0.5 as 0 and any values above 0.5 as 1.

The accuracy of the model on test set was reported as 0.7156.

According to the research done by Lang (2005), the monotonic MLP can enhance the model performance independently from the quality of training data set or complexity in terms of multidimensionality. Hence, the monotonic MLP have a potential to give the best performance among other models constrained with monotonicity.

However there are two problems exist in order for monotone MLP is chosen for the final model. First, such model confronts the problem of restriction in weights which hinders the nature of learning algorithm. Thereby, the interpretation of model using existing model-agnostics is even harder. Secondly, the 'monmlp' package can only use upto two layers, which have a potential in oversimplification. Nonetheless, the second issue can be solved with using lately developed model.

Final Model Selection and Further Model Agnostics

Why XGBoost?

Gradient boosting method is powerfully performing machine learning model among widely used machine learning models. It utilizes both classification and regression in learning via negative gradients. This implies that there is a high possibility of capturing the potential relationship between each features and response variable. As shown in the above experiments, the XGBoost method resulted in high performance in general.

However, comparing XGBoost from different white-box models, it is factual that the interpretability of the model is comparatively low. Although there are some tools for model-agnostic in order to capture the global and local behavior of the model, still the decision processes made by the model are in question. Nonetheless, along with the consideration of information demand from financial industry, we conclude that capturing global and local behavior of the model would suffice. Hence, it is worthwhile to trade-off interpretability with model performance, as we notice the difference in performance between XGBoost and other white-box models is significant.

In order to increase the interpretability of XGBoost, we have used several methods for model-agnostics. This section includes: feature importance, partial dependence plots, bivariate partial dependence plots, and local behavior of the model. All the results can be find in the enclosed jupyter notebook.

References

Molnar, C. (2020, November 30). Interpretable Machine Learning. From https://christophm.github.io/interpretable-ml-book/interpretability-importance.html

Tiwari, A. (2020, May 01). Application of Monotonic Constraints in Machine Learning Models. From https://medium.com/analytics-vidhya/application-of-monotonic-constraints-in-machine-learning-models-334564bea616

Kurzelewski, T., & Radzikowski, T. (2020, October 14). XAI Stories. From https://pbiecek.github.io/xai_stories/story-heloc-credits.html

Zhang, H., & Zhang, Z. (1999). Feedforward Networks with Monotone Constraints. **DOI:** 10.1109/IJCNN.1999.832655

Lang B. (2005) Monotonic Multi-layer Perceptron Networks as Universal Approximators. DOI https://doi.org/10.1007/11550907_6