CIS432: Final Project Report

A Prototype for a decision support system using predictive modeling

Simon Business School, University of Rochester
Team members: Liqing Huang, Dan Luo, Shaozhuo Xu, Ying Huang

I. Introduction

1. Business objectives

The objective of this project is to develop a predictive model and a decision support system (DSS) that evaluates the risk of Home Equity Line of Credit (HELOC) applications, and design an interactive interface that sales representatives in a bank/credit card company can use to decide on accepting or rejecting applications and explain the decision to customers.

The output would be used by the sales department to consider when deciding whether or not to approve a loan, and provide reasons to customers when turning down a loan or in customer service settings when responding to customer inquiries.

2. Model Performance Measurement

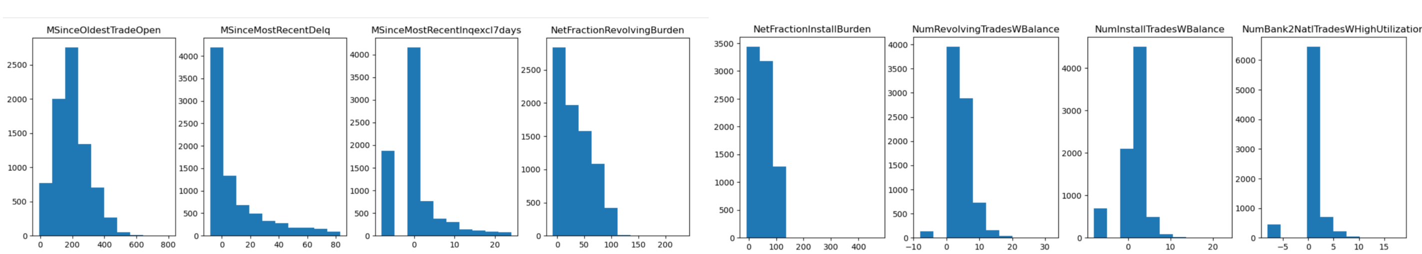
For our project, our key performance metric is test accuracy.

II. Data Preprocessing

1. Handling missing values

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

The data set (HELOC) we used to train models contains missing values encoded by -7, -8, and -9. -9 represents no record in file. We assume we are unable to make predictions without knowing clients' estimated external risk. Based on that, we remove any observation where ExternalRiskEstimate is -9. Since we expect -7, "condition not met", and -8, "no usable trades" are informative and that whatever generated them in the data that we have would generate them also in the future. Therefore, we impute missing data for observations with -7 and -8. This can help us improve model accuracy and reduce bias. As the histograms below suggest, most columns with missing values are heavily skewed or have a few dominant values.



Therefore, replacing missing values with mode is a better choice compared to mean or median, as it represents the most common value in the columns. In our pipeline, we used “most-frequent”(mode) to replace missing values. Meanwhile, we created columns to label observations that miss values. for simplicity that these values are informative and that whatever generated them in the data that we have would generate them also in the future.

2. Data transformation

a. Feature scaling

We apply standard scaling to the data in predictive modeling to ensure that all features are on the same scale and have equal importance in the model. This helps prevent features with larger magnitudes from dominating the model and producing biased results.

III. Interface Design

1. Assumption

a. Technical Proficiency of Users

The users of our interface are sales representatives in a bank/credit card company, who are not expected to have high knowledge about the principle of the models and how the predictions are made and do not have the prerequisites on machine learning. However, they will be trained to input basic information of customers into the interface we designed to generate recommendations.

b. Model Requirements

To successfully get recommendations from our interface, data on 23 features of customers will be needed as those in the training dataset. We expected sales representatives to extract the information of a credit application and be able to type in the information into the interface.

c. Prediction Explanation

After the user types in the information of the customer, our interface will provide both the prediction and the explanation for it. Specifically, the model will provide the features with their interpretations that cause the risk performance to be “Bad” to help explain why a customer is not recommended to be issued a loan.

2. Interface Design & Thoughts

a. Applicant ID Input

After considering the practicability of this interface, we think it would be more convenient for the user to extract applicants’ data by typing in only applicant ID. We assume that before the user typing, the dataset of applicants has already been loaded into our model.

b. Applicant Data Display

After inputting the customer ID, the system will extract the data of that specific customer and display his or her personal information. Although our users are not required to understand each feature, this display works as a check to make sure information of the applicant is complete.

c. Recommendation Display

To make the interface look neat and allow our user to quickly get the outcome of the model, a simple sentence of " Accept the Application" will be displayed if the model considers that the applicant is eligible to be issued a loan.

d. Recommendation Explanation

Explanation of the recommendation is generated by calculating the weight of each features that contribute to this result, the following is the process in details:

i. Obtain the weights (coefficients) associated with each feature from our SVM model
ii. Identify the top features that influenced the decision. Calculate the absolute values of the weights and sort them in descending order. Pick the top 3 features that have the largest absolute weights.
iii. For each of the top 3 features, determine whether the feature value in the user's application is above or below the threshold that separates accepted and declined applications. We use the average of the feature in the accepted applications as a reference.
iv. For each of the top 3 features, we prepare an explanation highlighting how the user's application deviates from the reference. Focus on the direction of the deviation (above or below) and its possible impact on the decision.
v. Combine the explanations for each of the top 3 features into a coherent message, and present it to the user.

IV. Model Training, Evaluation and Interpretation

1. Model Selection & Evaluation

a. Models:

i. Linear Regression models: linear regression, Ridge regression, Lasso regression
ii. Linear Classification models: SVM, SVC, LDA, logistic regression
iii. Tree-based models: decision tree, random forests, Boosting, Bagging

a. Metrics: Accuracy score & Cross-validation score

Without Standardization

Models:	Logistic Regression	LDA	SVC	SVM	Decision-tree	Random Forest	Boosting	Bagging
Without standardization: Accuracy (out-of-sample)	0.7444	0.7430	0.6580	0.7454	0.7126203750633553	0.7227572225038014	0.728839330968069	0.7399
Model Setup:			kernel= 'linear'		{'criterion': 'gini', 'max_depth': 5, 'min_samples_leaf': 30, 'min_samples_split': 2}	RandomForestClassifier(max_depth=6, max_leaf_nodes=16, min_samples_leaf=10, n_estimators=50, random_state=0)	n_estimators=80, random_state=0, learning_rate=1	

With Standardization

Models:	Logistic Regression	LDA	SVC	SVM	Decision-tree	Random Forest	Boosting	Bagging
With standardization: Accuracy (out-of-sample)	0.7108	0.7362	0.6869	0.7141	0.7121	0.7227	0.7288	0.7364
Model Setup:					{'criterion': 'gini', 'max_depth': 5, 'min_samples_leaf': 30, 'min_samples_split': 2}	RandomForestClassifier(max_depth=6, max_leaf_nodes=16, min_samples_leaf=10, n_estimators=50, random_state=0)	AdaBoostClassifier(learning_rate=1, n_estimators=80, random_state=0)	

2. Model Tuning

a. Parameters tuning

i. Grid search
ii. Bayesian optimization

3. Model Interpretation

a. Feature importance

Linear models offer the advantage of interpreting the model's weights or parameters as feature importance. Similarly, for SVMs, we can use the kernel weights of a linear SVM model to identify the most significant features. This information can be used to provide tailored feedback to the user based on the factors that affected the decision the most.

To achieve this, we first extracted the weights or coefficients associated with each feature from our SVM model. We then calculated the absolute values of these weights and sorted them in descending order. The top 3 features with the largest absolute weights were identified.

For each of these top 3 features, we determined whether the feature value in the user's application was above or below the threshold that separates accepted and declined applications. We used the average values of the feature in the accepted applications as a reference to prepare an explanation highlighting how the user's application deviates from the reference. We focused on the direction of the deviation (above or below) and its possible impact on the decision. Finally, we combined the explanations for each of the top 3 features into a coherent message and output it to the user.

Some of the factors that contributed to this decision include credit score, debt-to-income ratio, and recent late payments on other credit accounts. These factors had high weights for some observations, indicating their importance in the decision-making process. We recommend improving these factors and reapplying in the future when the user's financial situation has improved.

V. Summary

1. Outcomes

a. Model selection

After training and tuning three main types of models with the dataset standardized or not, the model with the best accuracy is SVM with non-standardized dataset, with an cross-validation accuracy of 0.7454.

b. Model Interpretation

For each of those applications we recommended to deny, we offered Top 3 reasons why the application is not considered, to help our sales team to understand our prediction, and also provided feedback for that customer.

2. Lessons learned, limitations and future improvements

a. Lessons learned

i. Underfitting and overfitting

During our initial model training, we observed an unusual pattern where the cross-validation score was consistently higher than the accuracy score for most of the models. Upon further investigation, we found that this was due to a limited number of rows in the validation set. To address this issue, we increased the proportion of the validation set when splitting the training and testing sets. As a result, we were able to successfully resolve this issue and the validation score is now consistently lower than the accuracy score.

ii. Feature Scaling

After attempting feature scaling on the dataset, we observed that the models perform better on average when trained on the unscaled data, rather than the scaled data. This is because most of the models we trained do not assume normalized data. By scaling the data, we made incorrect assumptions about these models, leading to decreased accuracy. In the future, we should normalize the data before training models like KNN, for which normalization is crucial, to avoid wasting time.

3. Implications for practical applications

Our objective is to predict credit risk and provide recommendations to the sales department to make better decisions. This involves avoiding the risk of fraudulent applications and identifying unreliable applicants. To achieve this, we provided the sales team with concrete reasons for the denial of certain applications. By doing so, we can improve customer service and help those who are able to improve their application conditions to continue the loan application process.