Credit Risk Prediction Model With LightGBM

Khanh Tran
Khanh.Tran@simon.rochester.edu

Chenxi Tao
Chenxi.Tao@simon.rochester.edu

Pin Li
Pin.Li@simon.rochester.edu

Ruiling Shen
Ruiling.Shen@simon.rochester

Jiawen Liang
Jiawen.Liang@simon.rochester.edu

December 13rd, 2019

1 Introduction

Our goal for this project is not only to find a relatively accurate and robust model for risk prediction, but more importantly, to improve its interpretability and give understandable explanations for sales representatives in a bank/credit card company can use to decide on accepting or rejecting applications.

The variable names are somewhat cryptic and the user might want more intuitive information for both of the mechanism and the outcome. For that purpose, we manually classified 23 variables into groups and labeled them using a name that makes sense in the business context. Combined with the most important contributing factors, users can easily get insights about the predictive relationship. To improve the user experience, the full model and explanations can be conditionally displayed in an interactive interface. We deployed our best models on GitHub and Heroku. Users can access our web app by:

- opening http://credit-risk.herokuapp.com/ via web browser, or
- running these command lines on the terminal:

```
pip install --upgrade streamlit
streamlit run
https://raw.githubusercontent.com/chriskhanhtran/credit-risk-prediction/master/app.py
```

A 5-minute explanation and live demonstration can be found at youtu.be/IEk8rmLnJDk.

2 Description of Models

Our model is basically a risk model based on Light Gradient Boosting Machine (LightGBM), which is the optimal one compared to the other three models (Random Forest, Logistic Regression & Support Vector Machine) that we've evaluated.

The prototype of the interactive interface we created is shown as Figure 1 to 3. Moving the slider to set values for 23 predictors, the according result is generated and displayed automatically on the right. The user could also alternatively choose other models and tick the box to compare their performance. We follow the principles and steps below to build the model:

- **Data pre-processing**: Firstly, we specified 'RiskPerformance' to be the dependent variable and transformed it to a factor. Secondly, Special values of -7, -8, -9 assigned to variables that represent unavailable data for different reasons are converted to binary features; meanwhile, we replaced the missing values with the mean of each variable and utilized StandardScalar to normalize all of them. Lastly, the full dataset is split into two, one for training and the other for test.

- **Modeling**: We successively trained four models, which are Random Forest, Logistic Regression, SVM and LightGBM to compare their performances for risk prediction, and chose the optimal one according to an AUC (Area Under The Curve) - ROC (Receiver Operating Characteristics) Curve. Plus, Another evaluation metric we used to measure performance of the model is Accuracy; since the original dataset contains a stratified random sample of records, we care little about Precision and Recall.

- **Model Selection**: As a result, both Random Forest and LightGBM are tuned to be great models to use, with the highest accuracy of 74% and the highest AUC respectively. We chose LightGBM as the optimal model and all the pth

3 Explaining Individual Predictions

In this case, the users are sales representatives from a bank/credit card company and we assume that they have limited technical proficiency. Thus, we have created an interactive web-interface to help them decide whether to accept or decline the applications having available customer information. In general, we offer the four main intuitive explanations in our web-interface:

1. **Variable input sliders with labels:**
   Figure 5 shows how it looks like. We provide labels beside each variable input slider to show the categories that the variable belongs to, for example, trade frequency or delinquency. Intuitively, this gives users a basic understanding of what the inputs are. For example, in Figure 1 we label 'MSinceOldestTradeOpen', 'AverageMinFile', and 'MSinceMostRecentTradeOpen' with 'Trade Open Time', meaning that these three sliders are associated with the trade open time. This would make it easier for users to input and interpret the data.

2. **Result with risk and confidence:**
   With inputs in the variable sliders, the result will show as the risk of being more than 90 days overdue. Users can optionally choose to check the accuracy to refer it as the

confidence. For example, an accuracy of 76% means that the user could be 76% confident about the risk generated by the model.

3. **Most important factors:**
   Together with the result of risk and confidence, we identify the most important factors and will be listed in decreasing order. These case-oriented factors will also appear with their contribution labels. Since we are choosing the LightGBM model as our default model, the way we select the five important factors are from `lgb_trained_model.feature_importance()` and then sort by importance values. To illustrate, Figure 2 shows the five most important features that influence the result. This design gives an intuitive understanding of what factors contribute most to the result and pave the way for the next decision-making process.

4. **Dictionary:**
   Dictionary is shown in another tab, helping users form a deeper understanding of the models. When selecting the dictionary, the original data information appears as well as the special values and special categorical value meanings. As we can see in Figure 3, when the users would like to know what 'AverageMInFile' is, he/she could refer to the dictionary and find the initial definition under the 'description' column. In this case, users would interpret their inputs and the most important features better.

5. **Document:**
   The document tab displays all information related, including the report, Notebook and Github. Since the codes are uploaded, anyone who has questions could refer to codes and basically find their answers themselves. Otherwise, our contact information is also attached, as is shown in Figure 4. If the users have any problem using or interpreting the results, they could contact us for help.

4 Summary

We succeed in building the model from training the existing dataset and giving intuitive explanations to the users through the web-interface. Our process includes pre-processing, modeling, and web-interface building. The approach that we use has the following key advantages:

- **LightGBM model advantages:** After comparing the mean accuracy of each model, we choose the highest accuracy model, lightGBM, as our default model. It offers the following distinguished features: "faster training speed, higher efficiency, lower memory usage, better accuracy, support of parallel and GPU learning, and Capable of handling large-scale data(3149)".

- **Interactive Web-interface:** We provide interactive web-interface to ensure users' individual prediction of models.

- **Case-oriented explanations:** The most important factors are shown with results, as well as the label that demonstrates the category. This made the prediction work as intuitive interpretation to non-technical users.

Acknowledgments: Thank you Yaron Shaposhnik for your assistance to our team in the whole process.

Appendix A - User Interface

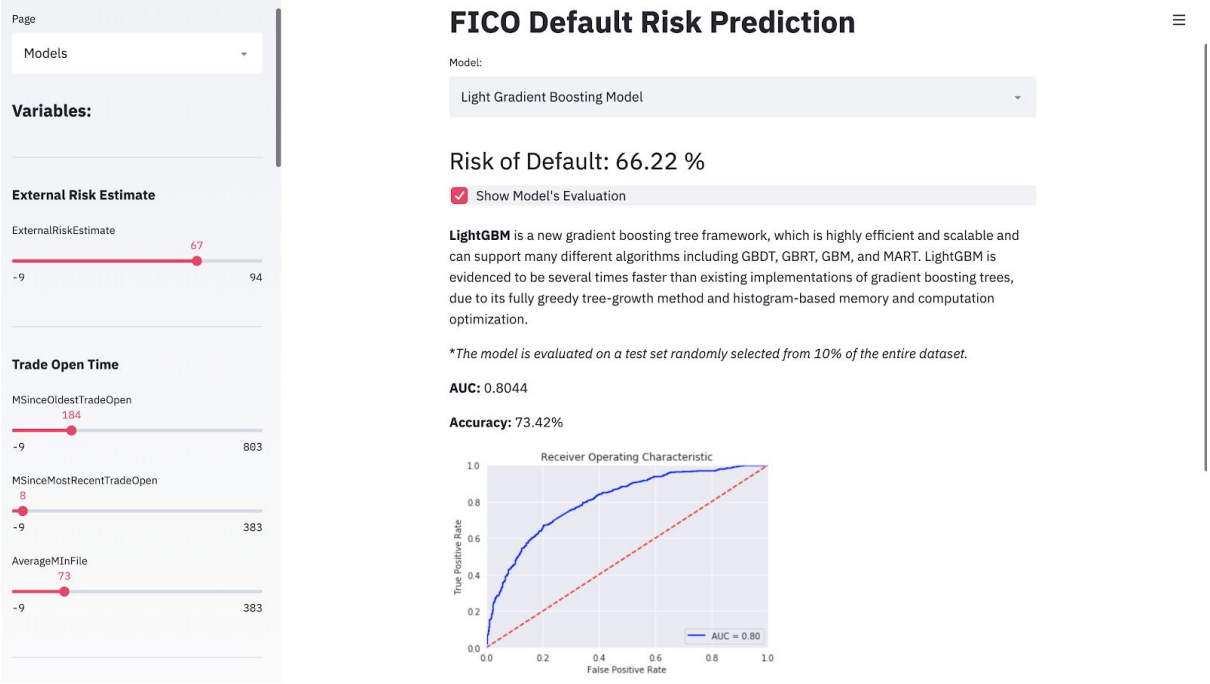

Figure 1: Interface - The Model

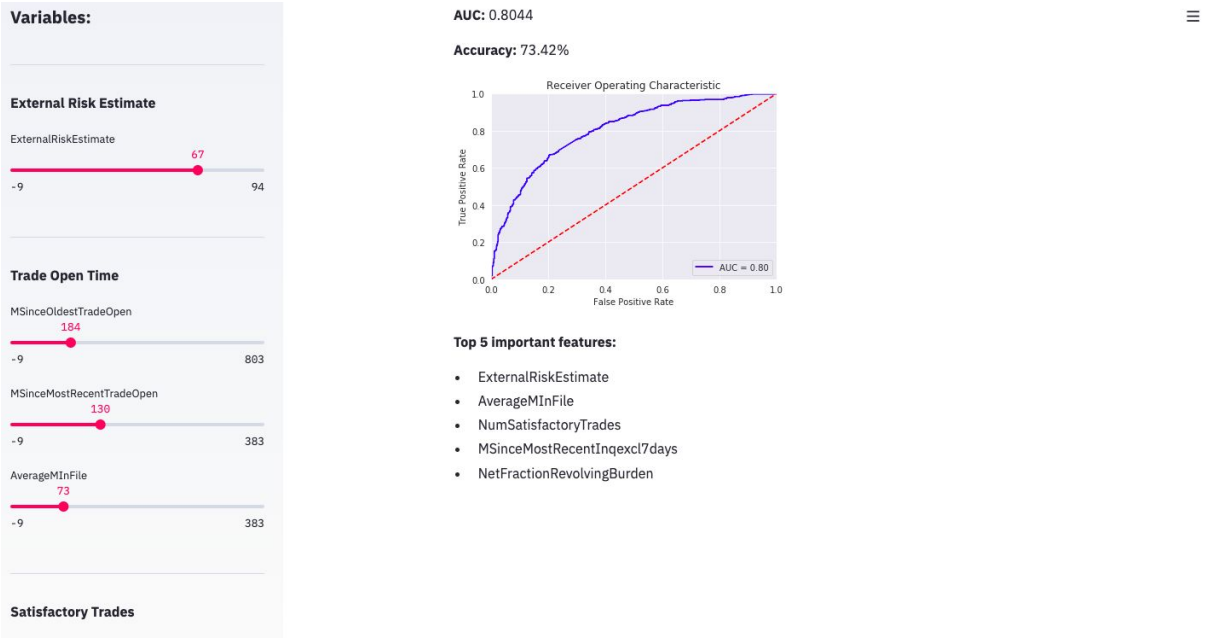

Figure 2: Interface - The model - Important Features

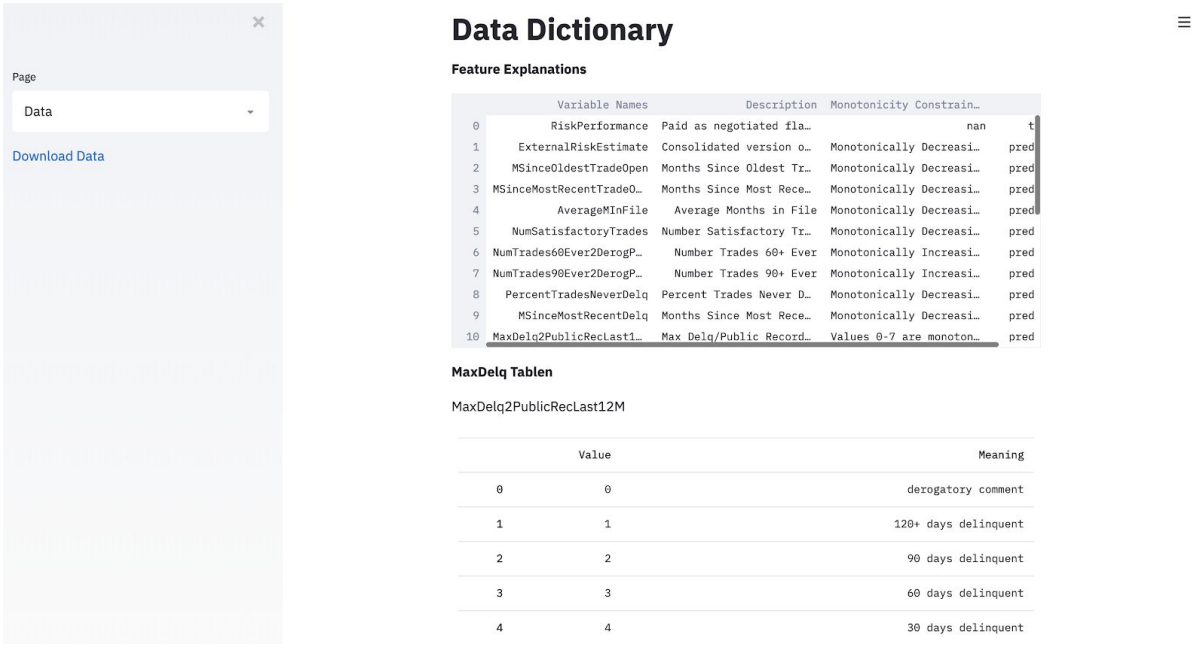

Figure 3: Interface - The Data Dictionary

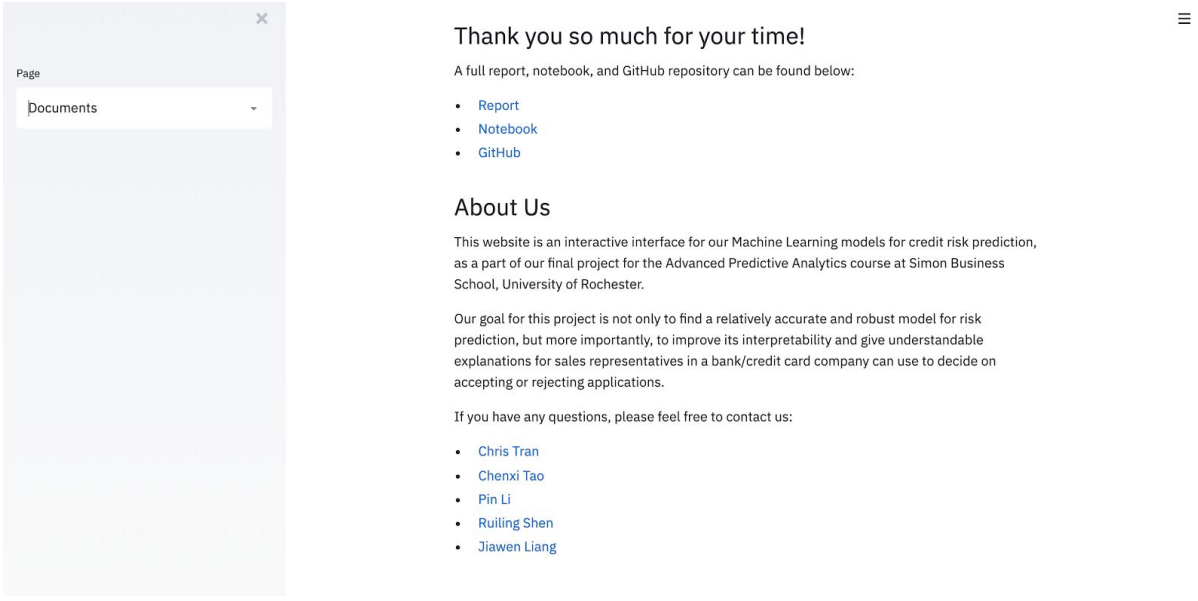

Figure 4: Interface - About Us

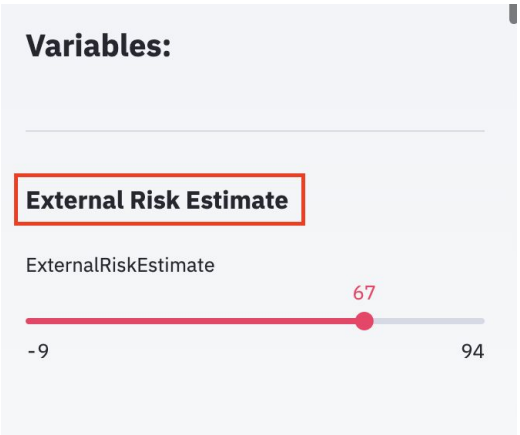

Figure 5: Variable With Label

Appendix B - References

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, Tie-Yan Liu. "LightGBM: A Highly Efficient Gradient Boosting Decision Tree." Advances in Neural Information Processing Systems 30 (NIPS 2017), pp. 3149-3157.