Application Scorecard Modeling

by Huayu Li

Content

- Introduction of Application Scorecard Modeling
- Introduction of dataset & Exploratory Data Analysis
- Methods and tools
- Final result
- Addition upon variance importance
- Conclusion

Introduction of Application Scorecard Modeling

- Definition: Application Scorecards are tools that allow organisations to predict the probability that an applicant will behave in a particular way, helping businesses to make effective automated decisions. The most commonly used application scorecard for credit, predicts the risk of a customer paying or not. This supports you as a business to make automated, accurate and consistent decisions on whether to approve, review or decline applicants.

- Application scorecards are statistical models typically developed using an institution's historical data for the relevant product, if sufficient such data is available.
- After the data has been extracted and verified it is critical to design a modelling data sample that is representative of the target portfolio and allows the resultant scorecard to meet the business objectives. This is achieved through detailed analysis of the available criteria, portfolio stability and behaviour. The model can then be developed using several methodologies, with linear and logistic regression proving to be the most common.
- In addition to the data, captured at the point of application, the most predictive application scorecard developments include credit bureau data which provides a detailed view of credit history.

- Dataset chosen: the Give Me Some Credit dataset from kaggle: `https://www.kaggle.com/c/GiveMeSomeCredit/overview`
- Details: (1) 150000 records in the training dataset, and 101503 records in the testing dataset; (2) the variable *SeriousDlqin2yrs* shows the result whether somebody will experience financial distress in the next two years or not. (3) The amount in the training set that have experienced financial distress in the following 2 years is 10026, about $\frac{1}{15}$ proportion of all users.
- Target: Using the datasets to predict the probability that somebody will experience financial distress in the next two years; using the result, we can construct the proper model that borrowers can use to help make the best financial decisions.

- Variable Description

Variable	Description
SeriousDlqin2yrs	Person experienced 90 days past due delinquency or worse
RevolvingUtilizationOfUnsecuredLines	Total balance on credit cards and personal lines of credit except real estate and no installment debt like car loans divided by the sum of credit limits
age	Age of borrower in years
NumberOfTime30-59DaysPastDueNotWorse	Number of times borrower has been 30-59 days past due but no worse in the last 2 years.
DebtRatio	Monthly debt payments, alimony, living costs divided by monthly gross income
MonthlyIncome	Monthly income
NumberOfOpenCreditLinesAndLoans	Number of Open loans (installment like car loan or mortgage) and Lines of credit (e.g. credit cards)
NumberOfTimes90DaysLate	Number of times borrower has been 90 days or more past due.
NumberRealEstateLoansOrLines	Number of mortgage and real estate loans including home equity lines of credit
NumberOfTime60-89DaysPastDueNotWorse	Number of times borrower has been 60-89 days past due but no worse in the last 2 years.
NumberOfDependents	Number of dependents in family excluding themselves (spouse, children etc.)

Table: Variable description

- Basic Statistics:

	RevolvingUtilizationOfUnsecuredLines	age
count	150000	150000
mean	6.048438	52.295207
std	249.755371	14.771866
min	0.0	0.0
25%	0.029867	41.0
50%	0.154181	52.0
75%	0.559046	63.0
max	50708.0	109.0

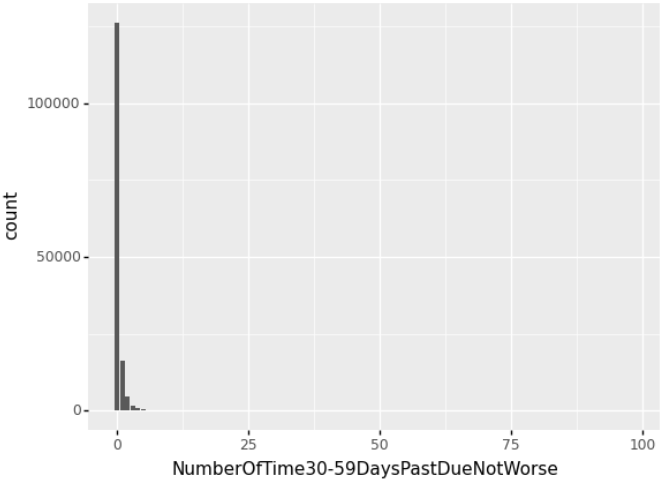
Table: Basic Statistics of variables

- Basic Statistics:

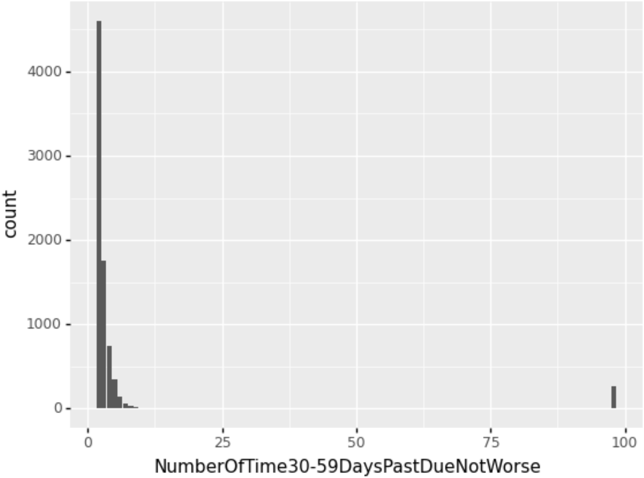
	DebtRatio	MonthlyIncome
count	150000	120269
mean	353.005076	6670.221
std	2037.818523	14384.67
min	0.0	0.0
25%	0.175074	3400.0
50%	0.366508	5400.0
75%	0.868254	8249.0
max	329664	3008750.0

Table: Basic Statistics of variables

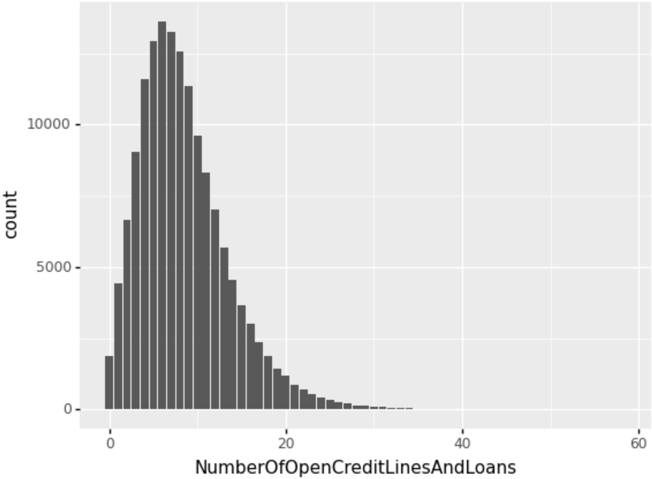
- NumberOfTime30-59DaysPastNotWorse



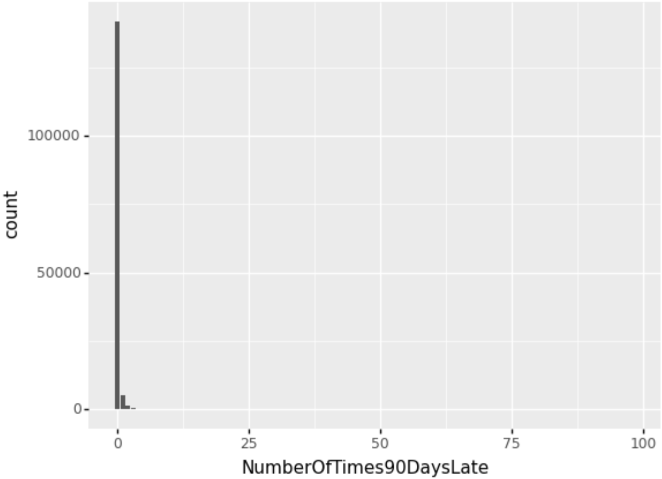
- NumberOfTime30-59DaysPastNotWorse($>$1 part)



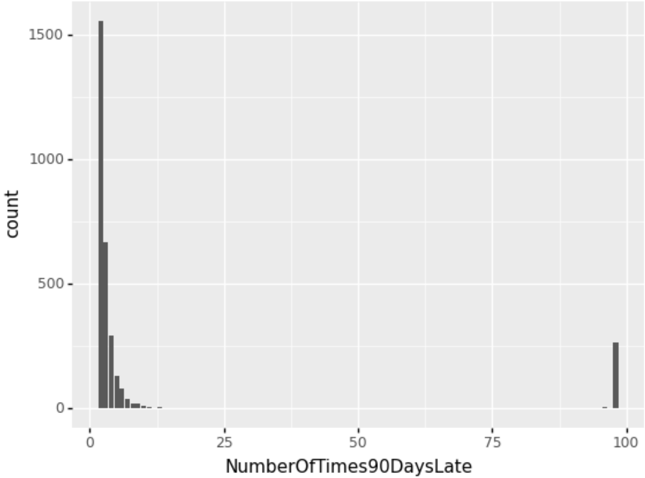
- NumberOfOpenCreditLinesAndLoans



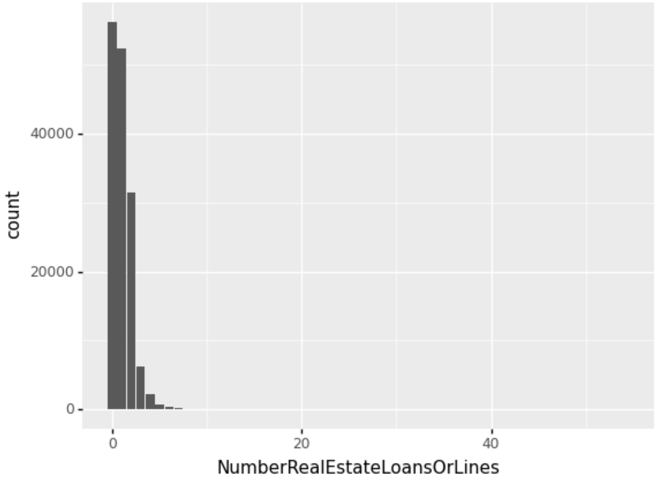
- NumberOfTimes90DaysLate



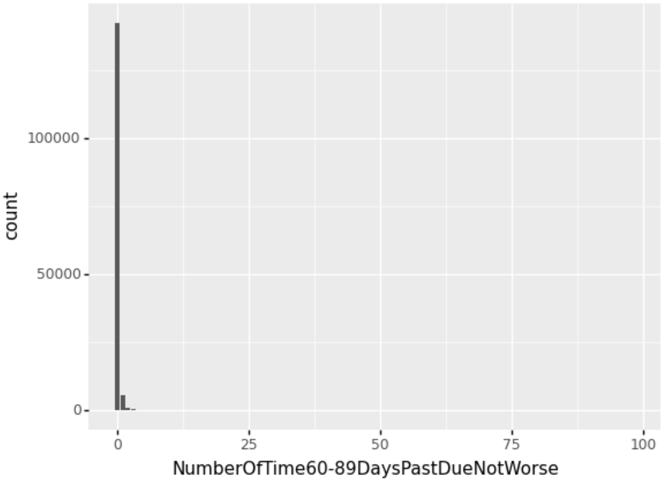
- NumberOfTimes90DaysLate (>1 part)



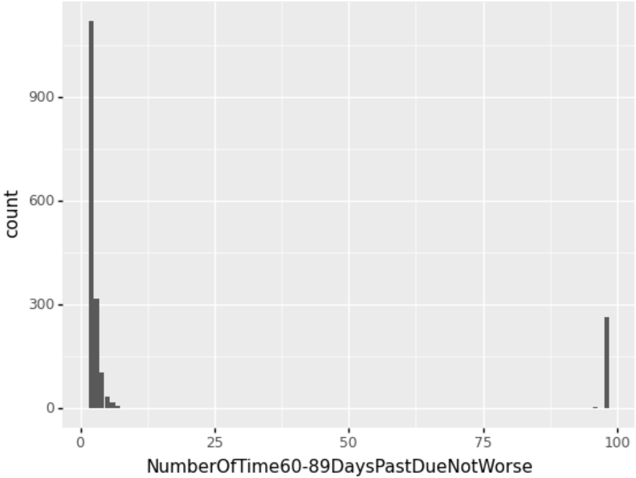
- NumberRealEstateLoansOrLines



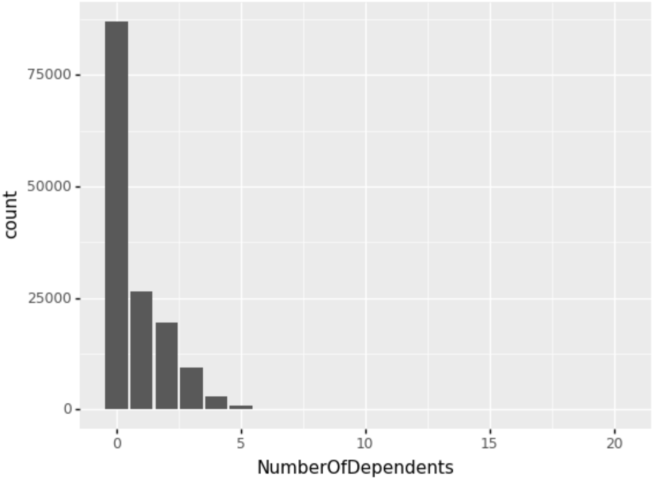
- NumberOfTime60-89DaysPastDueNotWorse



- NumberOfTime60-89DaysPastDueNotWorse (¿1)



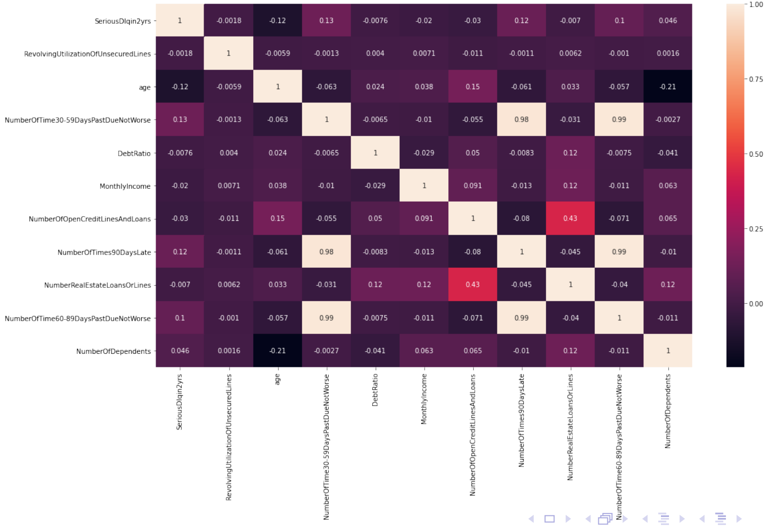
- NumberOfDependents



Introduction of dataset & Exploratory Data Analysis

Exploratory Data Analysis (Only on Training dataset)

- Correlation Heatmap



- Method: Random Forest Regression
- Procedures:
  (1) Extract the columns with missing data out (here the excluded variables are 'MonthlyIncome', 'NumberOfDependents', and the responsor 'SeriousDlqin2yrs').
  (2) For one given variable in 'MonthlyIncome' and 'NumberOfDependents', use the missing condition of this variable to split the dataset; then use the full part (the part with no missing data) to fit the Random Forest Regression model.
  (3) Use the model we gained to fit the part with missing data, and fill the missing values.

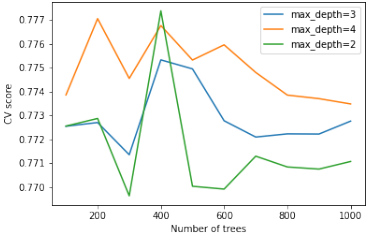
- Methods chosen: Logistic Regression, Random Forest, XGBoost, Gradient Boosting
- Procedures:
  (1) Split the training dataset into two parts with the proportion 9:1, and use the former part as the training set, the other as valid set.
  (2) Centerize the training set, on the variables about age and income; then make the transformations upon the valid and test dataset.
  (3) For the logistic regression, basicly use the method to fit the training set, to get the training dataset; for the other two methods, use the cross-validation score to judge the model that performs the best.
  (4) Use the valid set to check the performance of the models, using the recall score and the AUC score.
  (5) Use the methods on testing dataset to gain the predicted probabilities, and check the AUC scores on kaggle.

- Evaluation tools: AUC score, Recall score
- Recall score: The recall is the ratio $\frac{TP}{TP+FN}$ where TP is the number of true positives and FN the number of false negatives. The recall is intuitively the ability of the classifier to find all the positive samples.
- AUC score: AUC represents the probability that a random positive example is positioned to the right of a random negative example.

Final result

Model Selected

- Random Forest:



The parameter selected: max_depth=2, n_estimators=400

- XGBoost & Gradient Boosting: consider the function RandomizedSearchCV to get the best parameters.
  The parameter selected for XGBoost: scale_pos_weight=14, n_estimators=140, max_depth=3, learning_rate=0.1.
  The parameter selected for XGBoost: n_estimators=200, loss=deviance, learning_rate=0.1.

Final result

Result about classification

- Logistic Regression

	Pred_0	Pred_1
True_0	10692	3275
True_1	356	677

Table: Classification Result for Logistic Regression

- Random Forest

	Pred_0	Pred_1
True_0	10766	3201
True_1	240	793

Table: Classification Result for Random Forest

Final result

Result about classification

- XGBoost

	Pred_0	Pred_1
True_0	11077	2890
True_1	238	795

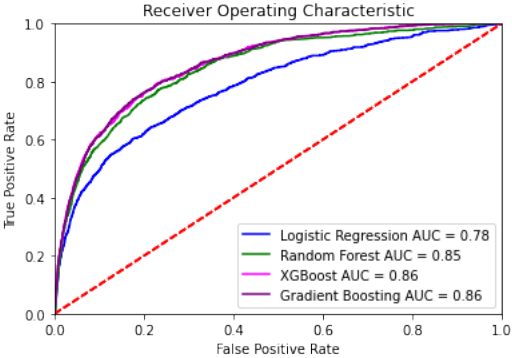
Table: Classification Result for XGBoost

- Gradient Boosting

	Pred_0	Pred_1
True_0	11089	2878
True_1	241	792

Table: Classification Result for Gradient Boosting

- AUC score:


Receiver Operating Characteristic
1.0
0.8
0.6
0.4
0.2
0.0
True Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate
Logistic Regression AUC = 0.78
Random Forest AUC = 0.85
XGBoost AUC = 0.86
Gradient Boosting AUC = 0.86

- Metric scores:

	LR	RF	XG	GB
Recall Score	0.65537	0.76766	0.76960	0.76670
F1 Score	0.27161	0.31549	0.33701	0.33681
AUC Score	0.78414	0.84821	0.86286	0.86343

Table: Metric results for the methods

- AUC scores for testing dataset on Kaggle:

	LR	RF	XG	GB
AUC score	0.79066	0.84341	0.86091	0.86052

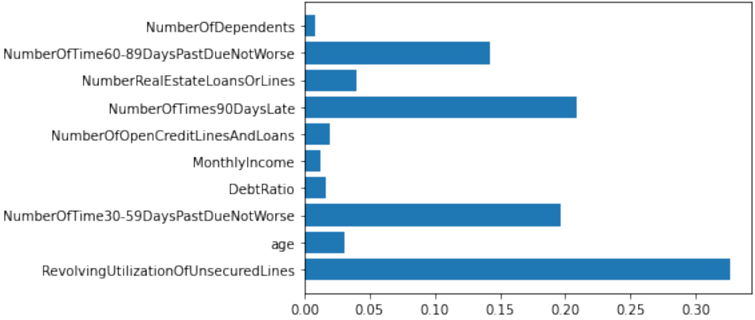
Table: AUC scores for testing dataset

- Bagging result (here the bagging classifier is the average of the previous three classifiers):

	Bagging
F1-Score	0.33966
Recall Score	0.74443
AUC Score	0.85893
Testing AUC Score	0.86045

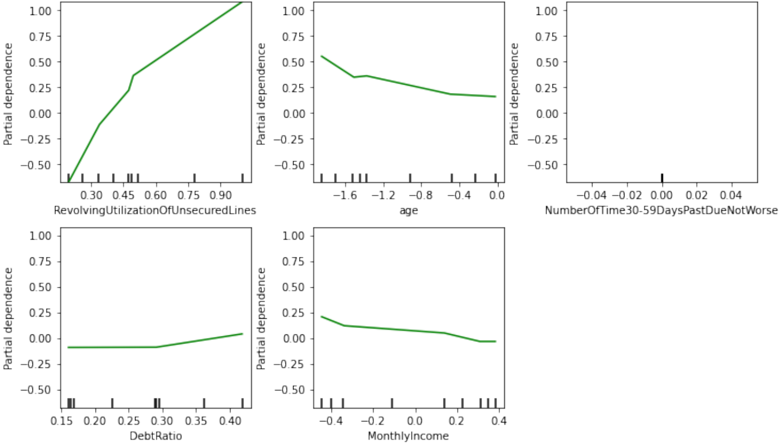
Table: Result of Bagging method

- Variance Importance plot (According to the XGBoost method):



- According to the Variance Importance Plot, the variables NumberOfTime60-89DaysPastDueNotWorse, NumberOfTime90DaysLate, NumberOfTime30-59DaysPastDueNotWorse, RevolvingUtilizationOfUnsecuredLines have really high importance.

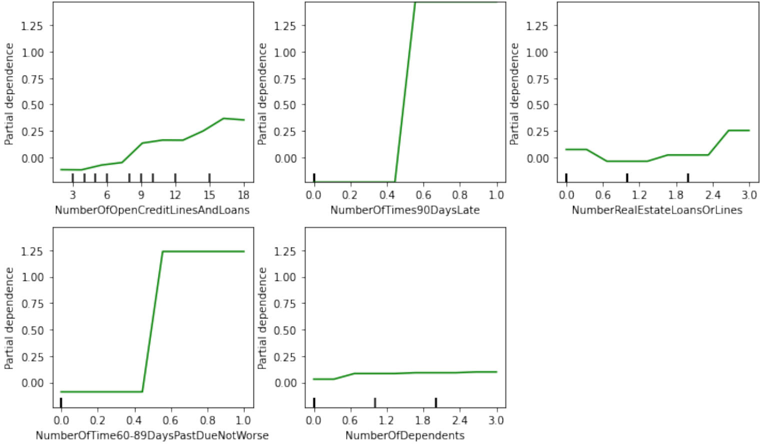
- Variance Importance plot (According to the XGBoost method):



Addition upon variance importance

Partial Dependence Plot

- Variance Importance plot (According to the XGBoost method):



- Here we focused on the Give Me Some Credit dataset, and finished the construction of the scorecard model, by predicting the probability that person experienced 90 days past due delinquency or worse. We have tried several different models, and the XGBoost and Gradient Boosting methods performed the best; we also considered the bagging of the given methods, and the bagging result still have really good performance. We also use the Variance Importance Plot and Partial Dependence Plot methods to find out the importance of the variables, and the trend of risk when one variable changes; these can be regarded as useful features to judge the risk.
- However, there are still some shortcoming about my work. The models are focused on the prediction of the risk probabilities, but the models are weak to judge the actual classification of the risk level, for the F1-scores are really low. The group that suffer from the risk may be really tiny, so anomaly detection may be considerable for this type of question.

References

Dataset source: https://www.kaggle.com/c/GiveMeSomeCredit

Chen, Tianqi, and Carlos Guestrin (2016)
Xgboost: A scalable tree boosting system
Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining pp. 785-794. 2016.

The End