DS-GA 1001 Final Project

Default Prediction from Payment Behavior

Yi Xu (yx2090), Man Jin (mj1637), Yuwei Wang (yw1854)

Dec.8th 2019

I. Business Understanding

With machine learning techniques, especially the improvement in solutions to classification problems, the prediction for loan repayment and default nowadays has been highly relying on data science. A common default prediction project observes data from individual borrower's information in combination with loan conditions, and final outcome of whether a default occurs as label, to train model as a classification prediction for new applicants. It has been widely used and proved effective, and become one of the routine tasks for financial sectors to perform on loan analysis.

But such traditional default predictions also have inevitable restrictions. For example, the mechanism is applied to borrowers only before their loan approval. Theoretically the data won't support in predicting borrower's default outcome because all approved loans are from the same class from traditional model. Yet the financial institution still has much to do if they can detect highly potential defaults among the approved loans, in order to maximize the profits. Another limit for traditional default analysis is the inability to detect default time. Because data is provided as individual borrowers as instances, it may work well in predicting default or not eventually, but they lack information to see which payment period a borrower likely to default. The time sense is also important for financial institutions to predict its cash flow, as well as to catch the right time to contact borrowers for other solutions of the loan.

With these business needs in mind, we obtained a dataset for mortgage payment records, a time series data set specifically containing payment behavior in every month, along with supporting individual information, and the outcome of default or payoff at each specific time periods. The dataset may help us meet our needs to make prediction for a certain time period. For example, with late payments in 3 consecutive months, intuitively the borrower may declare default in near future, as opposed to someone consistently pay loans back. We will make use of the time series features to mine data and expect the behavior history would produce a forecast for a certain window of future.

II. Data Understanding

1. Data Source and General Description

The data was published as supporting material with the book *Credit Risk Analytics – Measurement Techniques and Applications* (Baesens B., Roesch D., Scheule H. 2016), and was used as an example for case studies of loan analysis in this book. We obtained the data through the public website of the book, after contacting the author, and received the raw data.

This data, in panel form, reports origination and performance observations for 50,000 residential U.S. mortgage borrowers over 60 periods, with 622,489 lines in total. The periods have been deidentified. As in the real world, loans may originate before the start of the observation period (this is an issue where loans are transferred between banks and investors as in securitization). The loan observations may thus be censored as the loans mature or borrowers refinance. The data set provided along with the book is a randomized selection of mortgage-loan-level data collected from the original source - the portfolios underlying U.S. residential mortgage-backed securities (RMBS) securitization portfolios and provided by International Financial Research.

The dataset contains 23 variables and can be roughly grouped into following categories:

- Index: Borrower ID, time stamp for observation
- Loan information: origination time, maturity time, initial balance, outstanding balance, loan-to-value ratio, interest rate
- Borrower's information: investor type, original FICO score, real estate type
- Macroeconomics environment: GDP growth, house price index, unemployment rate, at each observation time
- Outcome: Default / Payoff / Neither default nor payoff, at each observation time

2. Selection Bias

The selection bias issue is caused by RMBS securitization portfolios as the original source. RMBS is similar to a bond that pays out based on payments from many individual mortgages. Government agency or non-agency investment firms who control a large number of residential loans, would package a large number of them together into a single pool of loans (or "portfolio") and sell bonds backed by the pool of loans. For this reason, our dataset may not be a good representative of general mortgage or home-equity loans, as we expect certain thresholds of mortgages to be selected into the structure of RMBS. The bias would depend on the strategy of agency entities who manage the pool of mortgages and operate RMBS.

3. Data Leakage

Since data for each individual is presented as a time series, with multiple lines associated with one borrower, we need to be careful that no same borrowers should be included in both training and testing set. This issue is trivial if we take individual borrower as instances, however, in our project we also created a part where instances is taken as (borrower, time stamp). If same borrower is included in both training and test set, even with different time stamps, there's still leakage issue. In our project, because of our sampling strategy, we didn't need to process further steps to avoid data leakage. But in reality, with longer time series, ideally each individual borrower's ID need to be hashed into lists before testing and validation.

III. Data Preparation

1. Data Mining Goal and Integration

Our goal for this project consists of two parts utilizing the same data set:

Part 1: Observing the final outcome of default for each individual – We'll aggregate data for each borrower by ID, and only interested in the final outcome irrespective of specific time period. An individual borrower will be taken as an instance; we write functions to extract time series features such as total payment history length, number of late payments, number of significantly lower payments, etc. This part helps us to generally observe the performance of predicting default with our dataset, to better understand certain features, as well as a comparison with traditional default prediction (non-time series dataset) methods.

Part 2: To predict a certain window length of outcome based on all available information up to a certain period – We will "cast" a training sample from certain selected periods and construct each instance as a tuple (borrower, time). For example, if take time T as a casting point and w as window length, then for all existing borrowers at time T, we extract all features from payment history up to T, and predict whether they default within the next w periods. In our actual work we selected several different time stamps and combines together as our training test sets.

2. Resampling strategy

Here we'll fully explain how to construct the training sample for part 2. There are in total 60 observation periods, we choose casting points uniformly to represent the full observation history.

Another issue is we want to keep the observations as independent instances as possible, in consistent with common assumptions for training sample. Borrowers make payments in consecutive periods. Thus, the closer periods we extract, the more dependent instances should be. We calculated the average (observed) payment duration in our sample and get a mean of 11.47 and a median 7.0. Most borrowers have a

relatively short existing payment history in observation.

We also want to compare the predicting results across different window lengths; and would hope to avoid an overlapping in time with outcomes and features. For example, if we want to predict default in 5 months (window length = 5), we may pick period 10 as casting point combined with outcome up to period 15, but we won't select any period before 15 as another casting point.

The final issue is to keep a reasonable base rate for target. As the data for Part 1, default rate in aggregated data is approximately 30%, which is a satisfying sample for training. For Part 2, with different prediction windows, the base rate varies. Window length = 1 has highly imbalance class, as default in an arbitrary period is rare; as compared to window length = 12 is closer to Part 1 aggregate since most loans would have ended within 12 months of observation. Thus, we choose different number of casting points for each window length, and process a down sampling (with positive targets reserved) to keep base rate for each sample approximately the same (0.3).

The detailed information of resampling for (Borrower, Time) is in Appendix A

3. Target Variable

We will choose 'default_time' as target variable. It's already labeled as 1 (default) or 0 (payoff, or neither default/payoff), so we don't need further transformation. We do have another variable 'status_time' which further differentiate label 0 into payoff and neither default or payoff, thus 3 classes in stored. But in this project we decided not to move for multi-class classification.

4. Missing Values

There are two variables which contain missing values: Original interest rate and LTV ratio (Loan to Value) at different times. Because of our resampling strategy of casting observation times, we would hope to keep all available records (without dropping any) to avoid complication. We fill the missing values of two variables in different ways:

- Original Interest Rate: it's supposed to be constant across different time periods, so we just look up the original interest at other available observation period; and fill in the same value in missing periods. Fortunately, we are able to fill in all missing ones from existing records.
- LTV ratio: the loan to value ratio is highly dependent on outstanding balance, HPI, so we conducted a simple polynomial regression with existing records, to predict the missing values and fill in.

The processing of missing values helps us to keep a complete dataset, with most possibly accurate information preserved.

5. Feature Engineering

Apart from what's already included in the dataset, we need to extract time-series related features from payment history for each borrower. These domain knowledge based features which we constructed include:

- Payment history length: total length of payment history up to the casting time point.
- Average / Standard Deviation/ Maximum / Minimum Payment Amount: average amount of payment every month (up to casting point)
- Count of Zero Payment: number of zero payments (late payments) in payment history (up to casting point)
- Count of Low Payment: number of low payment (defined as lower than mean minus 1 times std), in payment history (up to casting point)

IV. Model and Evaluation

(Section 2 – 4 are all based on Part 1 data)

1. Evaluation Metric

There is an imbalance in our dataset between two classes, where the negative class takes up 70% of the data, and the positive class only takes up 30%. If a classifier predicts every sample to be negative, then its accuracy ratel can reach 70% but it is in practice useless. So, accuracy may not be a good measure here. Additionally, precision and recall are equally important in our use case, so we use F1 score as our primary matrix for feature selection and grid search since it incorporates the information of both precision and recall.

Moreover, we compared different algorithms and chose the best among them using Area Under Precision Recall Curve rather than the Area Under ROC. It is because "A large number change in the number of false positives can lead to a small change in the false positive rate used in ROC analysis. Precision, on the other hand, by comparing false positives to true positives rather than true negatives, captures the effect of the large number of negative examples on the algorithm's performance." (Davis, Goadrich, 2006) In our case, we are less interested in how the model performs on the negative class, so FPR from the ROC curve are not really helpful. But we are more interested in how meaningful a positive result is from our classifier, we want to make sure that positive prediction is correct (precision), and that we get as many of the positives predicted as positives as possible (recall). Therefore, AUPRC is a better choice.

2. Baseline Model

We have three baseline models. The first one is a model that generates random predictions based on the class proportions of labels, so that we can later compare how well our model performs when compared with random guess. The second one is a logistic regression model with default

parameters since it's the most basic and commonly used algorithm in classification problems. And the third one is a decision tree model, so that we can later compare the tree-based algorithm such as gradient boosting and random forest with this baseline. Their performances are shown in Table 1.

Table 1 - Part 1: Baseline Models

Model	F1 score	Precision	Recall	AUPRC	Accuracy
Random	0.2958	0.2980	0.2937	0.4015	0.5815
LR	0.6091	0.7150	0.5306	0.6889	0.7962
DT	0.6722	0.6603	0.6846	0.7197	0.8002

3. Feature Selection

In general, there are three common strategies to select features: filter methods, wrapper methods and embedded methods. Considering that wrapper methods is computationally intensive, we employed the embedded method as our primary method, and used the filter method as a reference.

The most straightforward way in terms of filter methods is correlation coefficient. We visualized the pairwise correlation between all features and the label by heatmap (Appendix B), and we found that there are 13 features whose correlations with the label are greater than 0.05. But since correlation only considers linear relationship between two features, and it doesn't take model and scoring method into account, so we only used this as a reference and used embedded method to further select features.

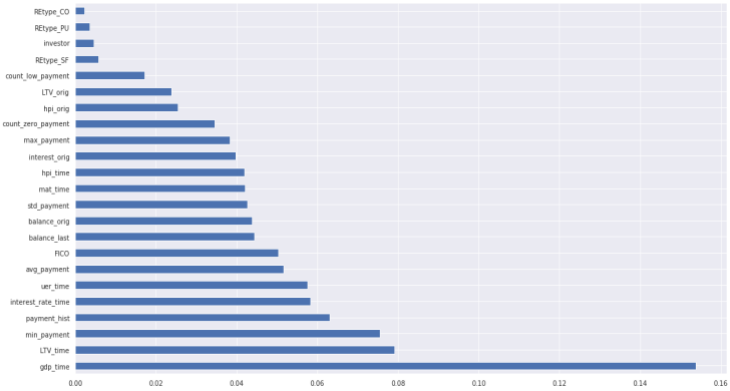


Figure 1 Feature Importance

Embedded methods combine the advantages of filter and wrapper methods. One of the most typical embedded techniques is feature importance from random forest algorithm. Figure 1 demonstrates the feature importance from random forest. Feature 'gdp_time' is the most significant feature under both selection methods, and we also see that features like 'min_payment' which has a low correlation with the label turned out to have a high feature importance from random forest, indicating a strong non-linear relationship between these features and the label. In order to find out the optimal number of features to use, we ranked these features by feature importance, added one more feature at each step to the model and fit it on training set and calculated the F1 score on the validation set. As shown in Figure 2, the performance of model increases until we have the top 7 features in the model, and it achieves the highest F1 score with 16 features. We decided to use the top 16 important features without loss of much information.

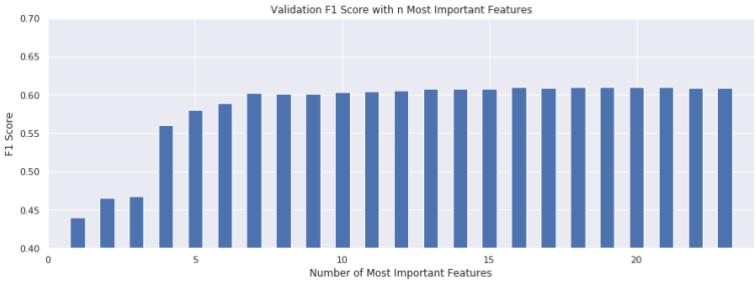


Figure 2 Validation F1 score with n Most Important Features

4. *Models and Results*

A. Algorithms Candidates

In order to find out the optimal algorithm, we used algorithms including Logistic Regression, SVM, Random Forest, Gradient Boosting and Neural Network. Logistic Regression is the simplest algorithm with fast speed, but it considers the linear relationship between the label and features. SVM takes the nonlinearity into consideration with the use of kernel, but takes much longer time when fit on a large dataset. Both Random forest and Gradient Boosting are ensembled tree-based methods and have a much better performance than decision tree. Random forest in general is less likely to overfit, while Gradient Boosting is more likely to overfit if the data is noisy, and the training generally takes longer because trees are built sequentially. Neural networks are a common way for large-scale classification problems, but it takes more effort to interpret the resulting model.

For each algorithm above, we chose the most important hyperparameters given by the following list for grid search:

- Logistics Regression:
  - ‘penalty’: ‘l1’, ‘l2’, ‘elasticnet’
  - ‘C’: 0.8, 1, 2
- SVM:
  - ‘Kernel’: ‘rbf’, ‘sigmoid’
  - C’: 0.8, 1, 2
- Random Forest:
  - ‘n_estimators’: 50,100, 150
  - ‘max_depth’: None, 5, 10
- Gradient Boosting:
  - ‘n_estimators’: 50,100, 150
  - ‘max_depth’: None, 5, 10
- Neural Networking:
  - ‘activation’: ‘logistic’, ‘tanh’, ‘relu’
  - ‘alpha’: 0.001, 0.0001, 0.00001

Take random forest as an example, two most important hyperparameters for this model are 'n_estimators' and 'max_depth', which represents the number of trees in the forest and the maximum depth of the tree. Then, we fit each grid search model on training set with and without feature selection, and found the optimal hyperparameters for a model which provides the best F1 score on validation data.

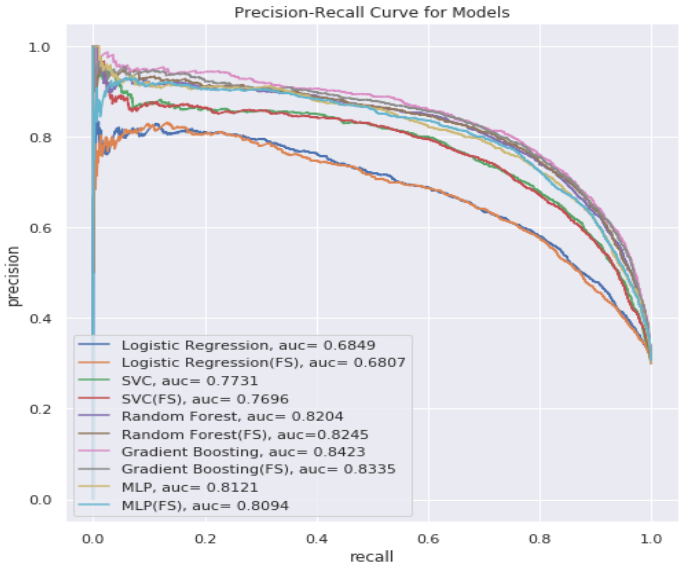


Figure 3 PRC Curve

Table 2 - Part 1: Classification Model Comparison					
Without Feature Selection With Feature Selection	F1 score	Precision	Recall	AUPRC	Accuracy
Logistics Regression	0.6755	0.5981	0.7758	0.6849	0.7769
	0.6723	0.5907	0.7802	0.6807	0.7724
SVM	0.7323	0.6527	0.8339	0.7731	0.8175
	0.7271	0.6493	0.8259	0.7696	0.8144
Random Forest	0.7707	0.7177	0.8323	0.8204	0.8518
	0.7686	0.7147	0.8313	0.8245	0.8502
Gradient Boosting	0.7759	0.7970	0.7558	0.8423	0.8693
	0.7683	0.7925	0.7454	0.8335	0.8654
Neural Network	0.7543	0.7610	0.7477	0.8121	0.8542
	0.7554	0.7833	0.7294	0.8094	0.8586

The results are shown in Table 2. We can see that there's only a small difference of F1 scores for every model before and after feature selection. This shows that features we selected already include most information from the dataset. From Figure 3, with AUPRC as our evaluation metric, gradient boosting model with the 'max_depth' = 10, 'n_estimators' = 100 without feature selection is the best model. The F1 score, precision, and recall all increases by around 10% when compared with the baseline decision tree model, the AUPRC increases by 12% and the accuracy rate increases by 7%.

B. Best Model Evaluation

- Bias-Variance Tradeoff

In order to determine the best value for parameter 'max_depth' and see how the model performance is influenced by bias-variance tradeoff, we draw the validation curve (Figure 4) of cross-validation F1 score with respect to different values of 'max_depth'. We know that the model complexity increases as the 'max_depth' increases. And we can see from the graph that when the model complexity is low, cross-validation F1 score is low because the estimator is underfitting, and the model is suffering from high bias and low variance. As 'max_depth' increases to 8, both the training and the validation F1 score improves as the decrease in bias outweighs the increase in variance. And the best cross-validation F1 score is achieved at 8 when the bias and

variance together are minimized. Then, as 'max_depth' continuous to increase, the model starts to overfit and doesn't generalizes as good as before, where the training F1 score converges to 1 and cross-validation F1 score starts to decrease. Here, the model suffers from high variance and low bias.

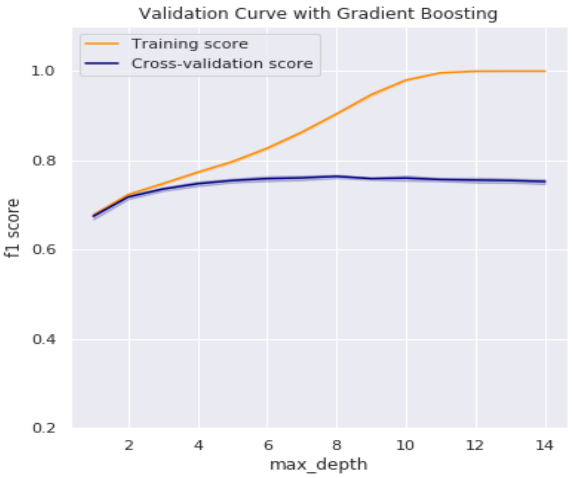


Figure 4 Validation Curve

- Feature Importance

Figure 5 shows the top 10 features of the best gradient boosting model based on feature importance, which can be given more attention when the financial institution monitors the default likelihood of a borrower.

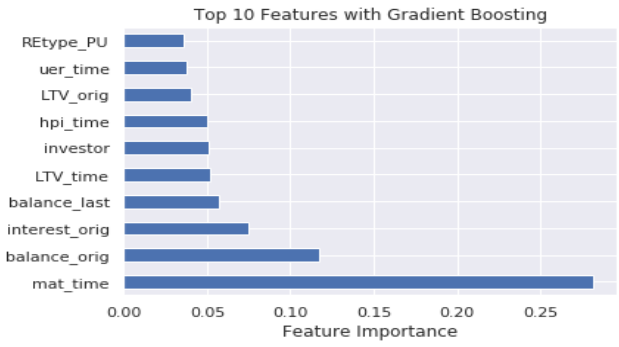


Figure 5 Feature Importance with Gradient Boosting

- Learning curve

Figure 6 shows the learning curve of validation and training score of an estimator for varying numbers of training samples. We can see from the graph that the gap between the training F1 score and cross-validation F1 score decreases as the training sample increases, but there's still a gap between them with all our training data and they haven't level out. This indicates a high variance and we can benefit from getting more training data.

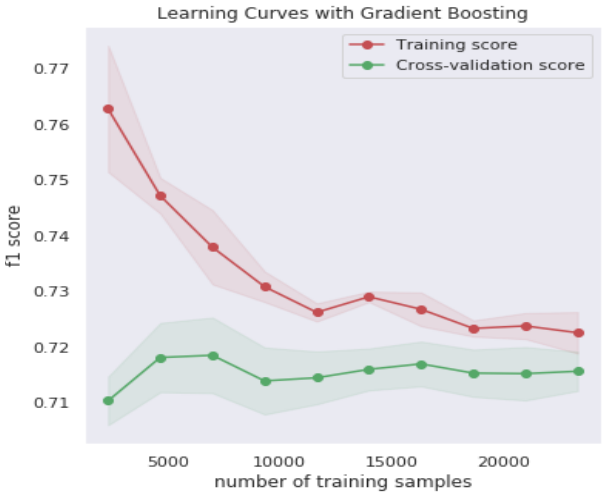


Figure 6 Learning Curve

- Test data

We apply our best model into the test dataset and get the performance in Table 3.

Table 3 - Part 1: Test Data

Best Model	Gradient Boosting
F1 score	0.7447
Precision	0.7850
Recall	0.7076
AUPRC	0.8223
Accuracy	0.8520

5. Model Performance on Different Prediction Window Length (Part 2)

Applying the same feature selection and model construction process in part 1, we got the performance graph (Figure 7) of the best model for each prediction window lengths in part 2. Except the recall, the other four metric show the same trend. Window length of 1 month has high evaluation score since the data information are more related with the near future than the far future. Downtrend at window length of 3 months indicates that the default status may vary a lot so it's hard to predict within the future 3 months. Similar results are shown for window length of 6 months and 12 months, thus predicting the default status within the window length of future 6 months is enough for predicting the long-term payment behavior. Thus, the short-term future prediction would be the ideal focus for financial institutions to adjust their business strategies.

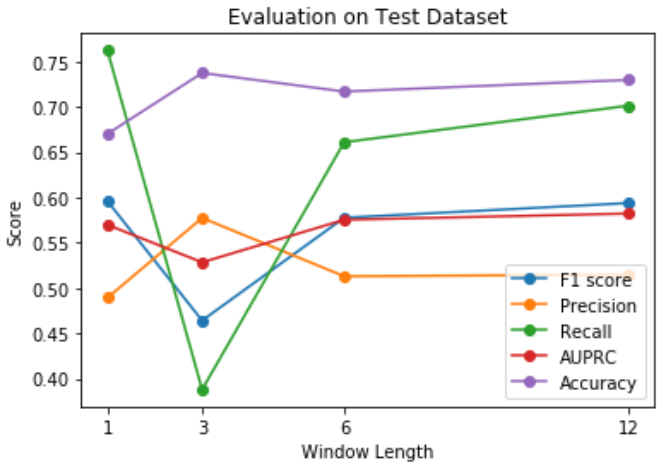


Figure 7 Window Length Comparison

Comparing the feature importance of different window lengths (Appendix G), we noticed that 'LTV_time', 'mat_time' and 'interest_rate_time' all play significant roles in both short-term and long-run prediction. However, the feature importance of 'gdp_time' increases remarkably as the window length increases, indicating that 'gdp_time' is a key feature in terms of long-term prediction, and this result is consistent with the feature selection result in part 1. Thus, it is necessary for companies to shift the feature focus depending on the window length of prediction.

And for the learning curves of all four window lengths (Appendix H), there exists a gap between training score and cross validation score, indicating high variance. Therefore, we can gather more data to improve the performance of our model.

V. Model Deployment

The model above can be deployed into a monitor system for financial institutions, where the features in the model become an indicator of potential defaults. Every time a borrower pays a loan, the database would be updated, and the model can detect unusual payment data and flag potential default in the near future, so that the financial institution is able to manage these loans separately, and have a better understanding of its future cash flow. Additionally, financial institutions can work with insurance companies

and offer insurance products to these potential default borrowers. Moreover, the default probability predicted by the model can be used as a reference for pricing if the financial institution wants to sell these loans in the secondary market.

In an actual production system, we monitor and evaluate our model by checking the consistency of the actual payment behavior and our prediction. The higher consistency, the better our model. Additionally, it is also necessary to examine if there exists the contradiction among results for different window lengths. For example, it doesn't make sense to predict default within 1 month but not default in the next 12 months. Finally, we also need to check if the proportion of default prediction is stable throughout time. If not, we need to check if there's a mistake in deployment or it's because of economic factors that more people start to default.

One issue with this is that as time goes by and new data comes in, the model should be retrained once in a while on the latest data for better prediction performance. Another issue worth noticing is that before every retrain, personal information should be excluded from the dataset, and down sampling may be required to balance the class distribution.

There are also important ethical considerations. One ethical risk with this model is that financial institutions might make decisions involving some extents of discrimination. They should not disapprove loan of people whose gender and race are more likely to default, based on the prediction of the model. The financial institutes should be careful about the individual demographic information. It's necessary to take out personal information before training data and making predictions. Also, if financial institutions want to sell potential bad loans based on the model prediction result to the secondary market, they should not conceal the potential high risk it may have. Transparency is one of the rules that loan organizations should follow tightly.

References

- Baesens B., Roesch D., Scheule H. (2016) Credit Risk Analytics: Measurement Techniques, Applications, and Examples in SAS. *Wiley & SAS Institution Inc.*
- Davis J., Goadrich M. (2006) The Relationship Between Precision-Recall and ROC Curves. *Proceeding ICML '06 Proceedings of the 23rd international conference on Machine learning* (pp. 233-240)
- Li M., Mickel A., Taylor S. (2018) "Should This Loan be Approved or Denied?": A Large Dataset with Class Assignment Guidelines. *Journal of Statistics Education* (pp. 55-66)
- Kaushik S. (2016) Introduction to Feature Selection methods with an example. *Analytics Vidhya Bootcamp (Website)*

Team Contribution:

Yuwei Wang (yw1854):

Exploratory Data Analysis, Business Understanding, Data Understanding, Data Preparation, Feature Engineering, Model Deployment

Yi Xu (yx2090):

Data Cleaning, Missing Value, Feature Selection, Baseline Model, Grid Search, Model Selection and Evaluation, Model Deployment

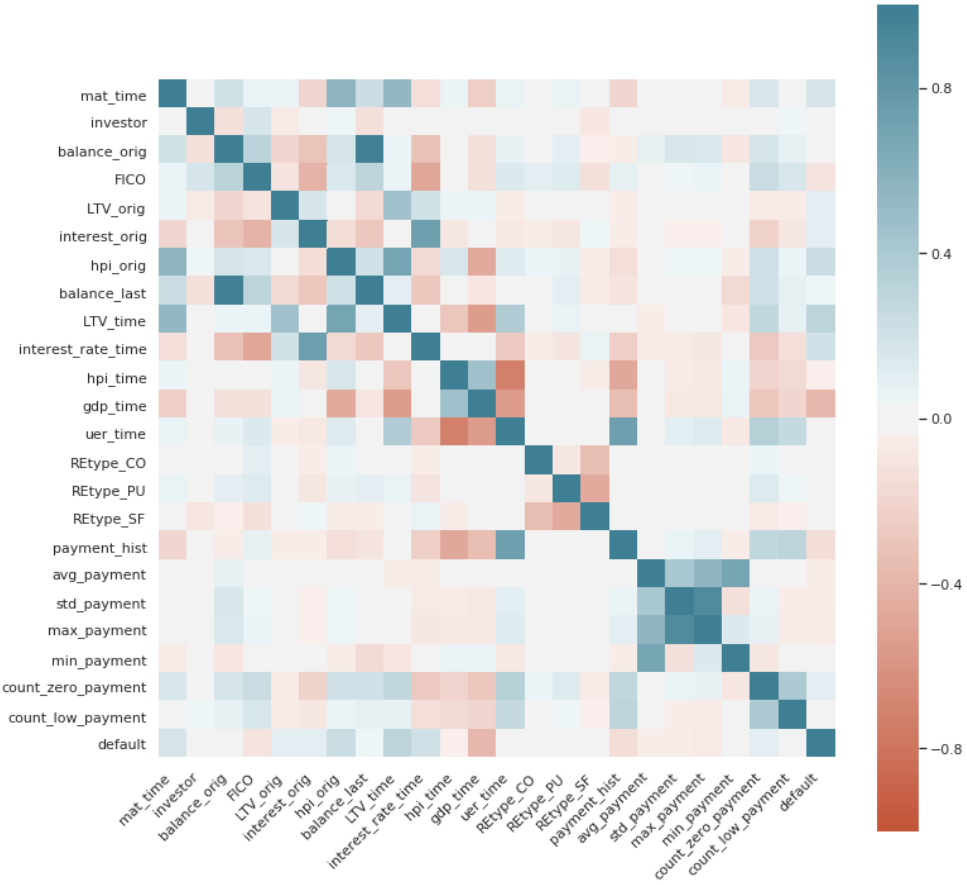
Man Jin (mj1637):

Evaluation Metrics, Feature Selection, Grid Search, Model Selection and Performance Comparison, Model Deployment

Appendix A - Description of casted samples for Part 2 (Borrower * Time)

	Casting time	Outcome time	Original Sample Size	Down Sample Size
Sample 1 (window=1)	3, 18, 33, 48, 8, 13, 23, 28, 38, 43, 53, 58	4, 19, 34, 49, 9, 14, 24, 29, 39, 44, 54, 59	115375	9154
Sample 2 (window=3)	3, 18, 33, 48, 13, 28, 43	6, 21, 36, 51, 16, 31, 46	77074	20923
Sample 3 (window=6)	3, 18, 33, 48, 13, 43	9, 24, 39, 54, 19, 49	48445	24599
Sample 4 (window=12)	3, 18, 33, 48	15, 30, 45, 60	35729	28070

Appendix B – Heatmap for Feature Selection (Part 1)



Appendix C – 1-Month Prediction Window (Part 2)

Part 2: 1-Month Prediction Window Baseline Models Comparison					
Model	F1 score	Precision	Recall	AUPRC	Accuracy
Random Prediction	0.3026	0.2980	0.3074	0.4055	0.5794
Logistics regression	0.3844	0.5603	0.2926	0.5025	0.7218
Decision Tree	0.4372	0.4301	0.4444	0.5197	0.6603

Part 2: 1-Month Prediction Window Model Comparison					
Without Feature Selection / With Feature Selection	F1 Score	Precision	Recall	AUPRC	Accuracy
Logistics Regression	0.5740	0.4778	0.7185	0.5014	0.6833
	0.5721	0.4760	0.7167	0.5009	0.6817
SVM	0.5997	0.4810	0.7963	0.5480	0.6844
	0.5958	0.4767	0.7944	0.5462	0.6800
Random Forest	0.5895	0.4781	0.7685	0.5546	0.6822
	0.5829	0.4744	0.7556	0.5573	0.6789
Gradient Boosting	0.4744	0.5752	0.4037	0.5287	0.7345
	0.4635	0.5620	0.3944	0.5336	0.7290
Neural Network	0.5034	0.5320	0.4778	0.5103	0.7201
	0.4720	0.5130	0.4370	0.4937	0.7097

Best Model	F1 Score	Precision	Recall	AUPRC	Accuracy
Random Forest with ‘max_depth’=5 and ‘n_estimators’=200	0.5961	0.4893	0.7625	0.5699	0.6702

Appendix D – 3-Month Prediction Window (Part 2)

Part 2: 3-Month Prediction Window Baseline Models Comparison					
Model	F1 score	Precision	Recall	AUPRC	Accuracy
Random Prediction	0.2943	0.2965	0.2921	0.3998	0.5825
Logistics regression	0.3416	0.5613	0.2455	0.4922	0.7180
Decision Tree	0.4385	0.4305	0.4468	0.5211	0.6590

Part 2: 3-Month Prediction Window Model Comparison					
Without Feature Selection / With Feature Selection	F1 Score	Precision	Recall	AUPRC	Accuracy
Logistics Regression	0.5659	0.4683	0.7148	0.4909	0.6732
	0.5678	0.4673	0.7234	0.4906	0.6719
SVM	0.5745	0.4645	0.7528	0.5176	0.6678
	0.5778	0.4688	0.7528	0.5204	0.6722
Random Forest	0.5737	0.4707	0.7347	0.5342	0.6747
	0.5742	0.4731	0.7303	0.5300	0.6773
Gradient Boosting	0.4679	0.5633	0.4001	0.5177	0.7288
	0.4666	0.5577	0.4010	0.5165	0.7268
Neural Network	0.4642	0.5324	0.4114	0.5087	0.7170
	0.4662	0.5322	0.4149	0.5008	0.7170

Best Model	F1 Score	Precision	Recall	AUPRC	Accuracy
Random Forest with ‘max_depth’=5 and ‘n_estimators’=50	0.5367	0.4399	0.6882	0.5016	0.6481

Appendix E – 6-Month Prediction Window (Part 2)

Part 2: 6-Month Prediction Window Baseline Models Comparison					
Model	F1 score	Precision	Recall	AUPRC	Accuracy
Random Prediction	0.2882	0.2912	0.2852	0.3945	0.5809
Logistics regression	0.3749	0.5495	0.2845	0.5014	0.7178
Decision Tree	0.4513	0.4446	0.4583	0.5320	0.6686

Part 2: 6-Month Prediction Window Model Comparison					
Without Feature Selection / With Feature Selection	F1 Score	Precision	Recall	AUPRC	Accuracy
Logistics Regression	0.5699	0.4597	0.7497	0.4987	0.6635
	0.5658	0.4582	0.7394	0.4974	0.6625
SVM	0.5786	0.4756	0.7387	0.5340	0.6800
	0.5729	0.4704	0.7326	0.5296	0.6751
Random Forest	0.5747	0.5003	0.6751	0.5654	0.7027
	0.5787	0.5072	0.6737	0.5682	0.7082
Gradient Boosting	0.4984	0.5975	0.4275	0.5659	0.7440
	0.5	0.5994	0.4289	0.5620	0.7449
Neural Network	0.4621	0.5392	0.4042	0.5048	0.7200
	0.5004	0.5324	0.4720	0.5064	0.7196

Best Model	F1 Score	Precision	Recall	AUPRC	Accuracy
Random Forest with ‘max_depth’=10 and ‘n_estimators’=150	0.5777	0.5130	0.6611	0.5754	0.7170

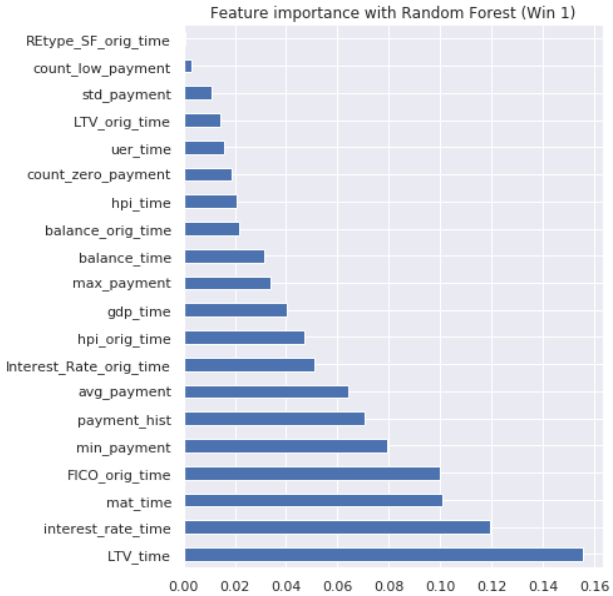
Appendix F – 12-Month Prediction Window (Part 2)

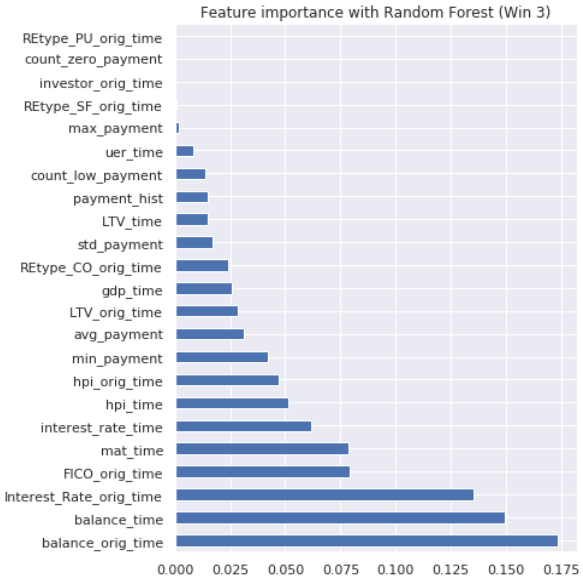
Part 2: 12-Month Prediction Window Baseline Models Comparison					
Model	F1 score	Precision	Recall	AUPRC	Accuracy
Random Prediction	0.2830	0.2814	0.2846	0.3856	0.5864
Logistics regression	0.4260	0.5503	0.3474	0.5084	0.7314
Decision Tree	0.4287	0.4133	0.4452	0.5088	0.6596

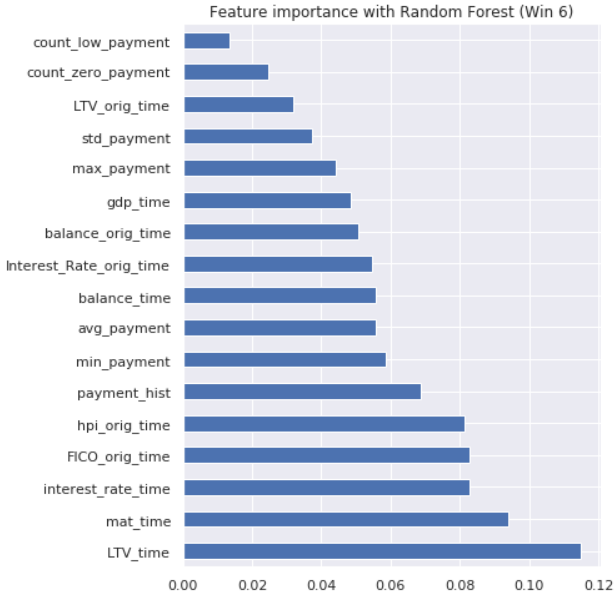
Part 2: 12-Month Prediction Window Model Comparison					
Without Feature Selection / With Feature Selection	F1 Score	Precision	Recall	AUPRC	Accuracy
Logistics Regression	0.5816	0.4634	0.7808	0.5049	0.6778
	0.5823	0.4638	0.7821	0.5032	0.6782
SVM	0.5919	0.4829	0.7646	0.5504	0.6976
	0.5922	0.4832	0.7646	0.5399	0.6980
Random Forest	0.5985	0.5139	0.7167	0.5694	0.7242
	0.5955	0.5143	0.7073	0.5701	0.7244
Gradient Boosting	0.5222	0.5892	0.4689	0.5536	0.7539
	0.5342	0.5963	0.4838	0.5579	0.7580
Neural Network	0.5285	0.5470	0.5112	0.5283	0.7383
	0.5611	0.5743	0.5486	0.5614	0.7539

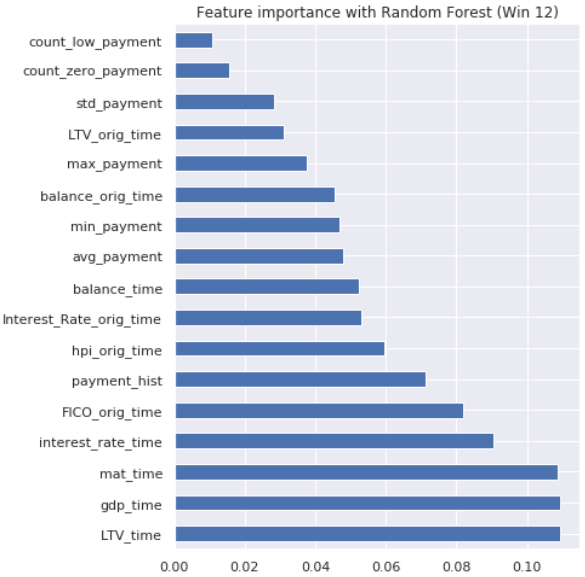
Best Model	F1 Score	Precision	Recall	AUPRC	Accuracy
Random Forest with ‘max_depth’=10 and ‘n_estimators’=150	0.5939	0.5149	0.7014	0.5823	0.7299

Appendix G -- Feature Importance with Different Window Lengths

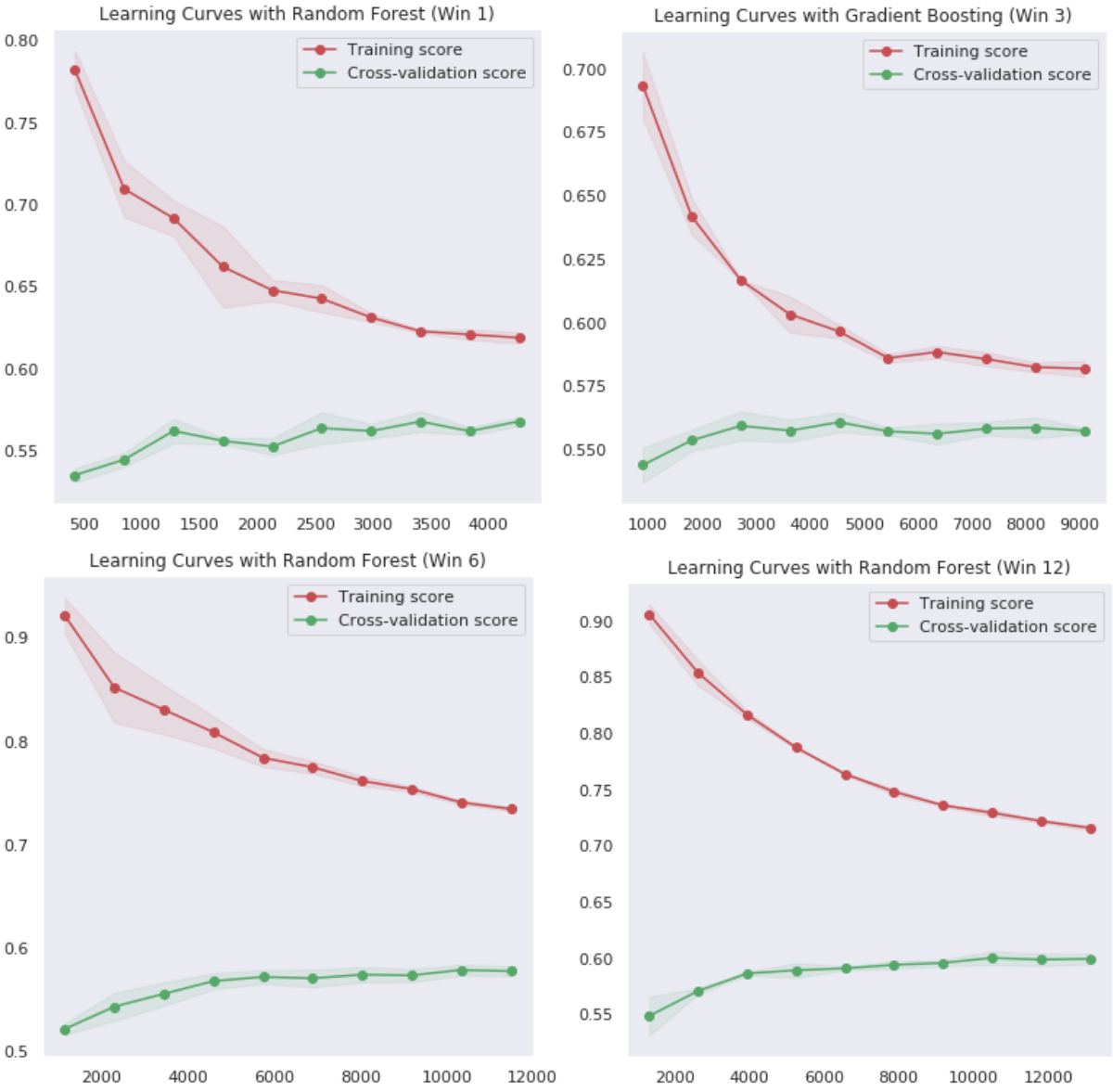








Appendix H -- Learning Curve with Different Window Length



Appendix I -- PRC for Different Models Under Various Window Length