Improvement of Classification in Credit Default

1 Introduction

1.1 Problem statement

Credit default becomes a big concern for most of the credit card companies, therefore, they try to predict potential default in advance. The earlier the potential accounts are detected and the lower losses [1]. So, an effective approach for predicting a potential default account in advance is crucial for the creditors.

1.2 Motivation

In this project, we are interested in addressing this issue by predicting if a new customer should be granted credit card or not based on the information of the past customer whose credit card has or has not become default.

1.3 Review of other researches

In [2], the authors present an ensemble approach of four approaches, 'majority voting', 'bagging', 'boosting' and 'stacking' for customer default prediction. And their conclusion is that a single classifier is not good enough for a classification problem of this type. In [3], the authors proposed an application of online learning for a credit card default detection system that achieves real-time model tuning with minimal efforts for computations, and one of their conclusions is **random forest** exhibits great performance in terms of efficiency and accuracy (81.96%). In [4], the authors used two approaches, Machine Learning and Heuristic to mine default accounts, and one of their conclusions is Machine Learning Approach gives higher accuracy. In [5], the authors compared the accuracy of different data mining techniques for predicting the credit card defaulters.

1.4 Open questions in the domain

Can we achieve a higher prediction accuracy when compared with the best machine model random forest, and at the same time, can we improve the computing efficiency?

1.5 Short summary of your proposed approach

As we have learnt from related work [3], [4], Random Forest is the best machine learning model when giving prediction with the customer default problem. Our approach aims to enhance the accuracy on top of random forest as well as improve the computing efficiency, we set our baseline as Random Forest.

Recently, in paper[6][10], the authors develop the first umbrella algorithm that implements the NP paradigm for all scoring-type classification methods, such as logistic regression, support vector machines, and random forests, which can effectively minimize type II error 9the conditional probability of misclassifying a class 1 observation as class 0) to enhance the accuracy of classification, in our project, we will apply this method to our model.

Feature selection becomes increasingly important in classification problems, in this project, we compare some feature selection methods aim to find the best feature engineering method that can improve the computation efficiency as well as raise the prediction accuracy.

1.5 Novelty of this project

Below we listed the **novelty** of this project:

Our approach is based on two research works on NP-Umbrella [6], [10], and we apply this new method NP-Umbrella into our credit default problem with the purpose of enhancing the prediction accuracy, which is never done before in this field.

In order to further improve the computing efficiency and enhance the prediction accuracy, we use two feature selection methods for feature engineering, which are based on the model and the dataset.

In summary, our novelty is we built a credit default prediction system that based on NP-Umbrella and Feature selection, which can give a raise the prediction accuracy enhances the efficiency of computing time.

1.6 Challenge

The main challenge for us is that as feature selection is data oriented as well as model oriented, therefore, we give a comprehensive comparison among different feature selection methods.

2 Backgrounds

2.1 Research on Credit Default Prediction

In work [2], the authors worked on bankruptcy prediction and credit scoring from four different summarized credit datasets. Their conclusion is that a single classifier is not good enough for a classification problem of this type. What they have proposed is an ensemble approach of four approaches, 'majority voting', 'bagging', 'boosting' and 'stacking', where multiple classifiers are used on the same problem and then the result from all classifiers are combined to get the final result.

In work [3], they proposed an application of online learning for a credit card default detection system that achieves real-time model tuning with minimal efforts for computations. They use Online Sequential Extreme Learning Machine (OS-ELM) and Online Adaptive Boosting (Online AdaBoost) methods in their experiment as well as other classic algorithms such as KNN, SVM, RF, and NB. They found RF exhibits great performance in terms of efficiency and accuracy (**81.96%**). They conclude that the online AdaBoost has the best computational efficiency.

In work [4], the authors used two approaches, Machine Learning and Heuristic to mine default accounts. The main idea of the Heuristic Approach is to calculate the risk factor from the recent transactional data (online) and combine the results with pre-computed risk factors from historical (offline) data in an efficient way. They showed the heuristic approach can predict a default account significantly in advance, which is very cost efficient.

In the work of [5], the authors compared the accuracy of different data mining techniques for predicting the credit card defaulters. From the experiment, based on the area ratio in the lift chart on the validation data, they ranked the algorithms as follows: artificial neural network,

classification trees, naive Bayesian classifiers, K-nearest neighbor classifiers, logistic regression, and discriminant analysis.

From above research work, we get the conclusion that Random Forest gives the best accuracy, what we would like to research is how to raise its prediction accuracy as well as improve the computing efficiency with the credit default problem.

In work [6][10], the authors develop the first umbrella algorithm that implements the NP paradigm for all scoring-type classification methods, such as logistic regression, support vector machines, and random forests, which can effectively minimize type II error 9the conditional probability of misclassifying a class 1 observation as class 0) to enhance the accuracy of classification.

2.2 Dataset

2.2.1 Background

This research aimed at the case of customer default payments in Taiwan and compares the predictive accuracy of probability of default among six data mining methods. From the perspective of risk management, the result of predictive accuracy of the estimated probability of default will be more valuable than the binary result of classification - credible or not credible clients. Because the real probability of default is unknown, this study presented the novel Sorting Smoothing Method to estimate the real probability of default.

2.2.2 data Describe

This research employed a binary variable, default payment (Yes = 1, No = 0), as the response variable. Y: default payment (Yes = 1, No = 0)

This study reviewed the literature and used the following 23 variables as explanatory variables:

- X1: Amount of the given credit (NT dollar): it includes both the individual consumer credit and his/her family (supplementary) credit. continuous variable
- X2: Gender (1 = male; 2 = female).
- X3: Education (1 = university; 2 = high school; 3 = others (lower than high school); 4 = graduate school).

- X4: Marital status (1 = married; 2 = single; 3 = others).
- X5: Age (year).

X6 - X11: History of past payment. We tracked the past monthly payment records (from April to September, 2005) as follows:

- X6: the repayment status in September, 2005;
- X7: the repayment status in August, 2005;
- X8: the repayment status in July, 2005;
- X9: the repayment status in June, 2005;
- X10: the repayment status in May, 2005;
- X11: the repayment status in April, 2005.

The measurement scale for the repayment status is: -2 = prepayment ; -1 = pay duly; 1 = payment delay for one month; 2 = payment delay for two months; 3 = payment delay for three months; 4 = payment delay for four months; 5 = payment delay for five months; 6 = payment delay for six months; 7 = payment delay for seven months; 8 = payment delay for eight months; 9 = payment delay for nine months and above.

X12-X17: Amount of bill statement (NT dollar), continuous variable.

- X12: amount of bill statement in September, 2005;
- X13: amount of bill statement in August, 2005;
- X14: amount of bill statement in July, 2005;
- X15: amount of bill statement in June, 2005;
- X16: amount of bill statement in May, 2005;
- X17: amount of bill statement in April, 2005.

X18-X23: Amount of previous payment (NT dollar), continuous variable.

- X18: amount paid in September, 2005;
- X19: amount paid in August, 2005;
- X20: amount paid in July, 2005;
- X21: amount paid in June, 2005;
- X22: amount paid in May, 2005;
- X23: amount paid in April, 2005.

3 Methods

Our method contains below parts: Loading data, Data preprocessing, Data visualization, Baseline model, NP-Umbrella algorithm(Version1), compare the accuracy between Baseline model and NP-Umbrella algorithm (Version1), Feature selection based on the better accuracy rate's model, using the new variables in the better model (we get before), as shown in Figure 1. In the whole project, what we would like to study is how to raise the prediction accuracy as well as improve the computing efficiency with the credit default problem.

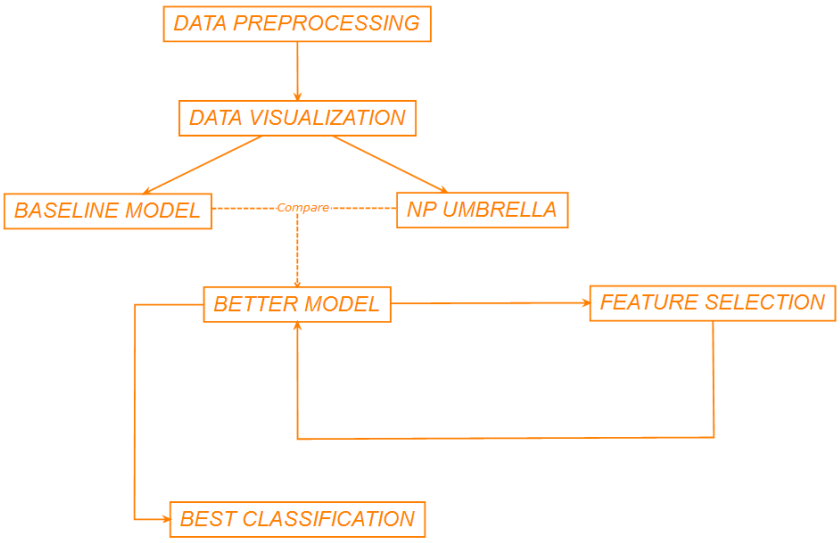

Figure 1 Flow chart of the approach

3.1 Baseline Model

Based on the research survey, with credit default classification, the best prediction accuracy is given by random forest, therefore, we set our baseline model as random forest.

3.2 NP-Umbrella algorithm (Version1)

NP-Umbrella-Version1 is based on Neyman-Pearson classification algorithms. NP-umbrella is a new method that enhances the binary classification accuracy. In classical method the oracle risk is defined as the:

*[Probability of type I error] * [Proportion of class zero in population] + [Probability of type II error] * [proportion of class one in population]*

And all classical methods try to minimize the Type II error based on a fixed probability of type I error = α. If it was possible to reach α in our classical methods, we could have classified all of the true class 1 incident even when we observed. Since it does not seems to be practical (in sample level) to always reach the prespecified α for the type I error probability, it is more realistic to set an upper bound for Type I error probability (called threshold) δ and the minimize the Type II error probability under this restriction.

The following algorithm is how it can be done:

First determine α, δ, M as number of splits

Define the following functions:

- Rank(n,α, δ)
  For each sub-sample size k between 1 to n, calculate the upper bound of type I error violation by the lemma provided in theorem. It is a function of α, sample size n and and sub-sample size k.
  Find the smallest subsample size k such that violation error is less than δ.
  Output is this smallest subsample size.
- NP (Sample, M,α, δ)
  Split the training data to 0,1 class.
  Apply Rank on (size of class zero/2, α, δ) call it k*.
  For each number i between 1 to M as the split size randomly split the class zero to i and **rest** of the elements.
  And do the following.
    Unite the first part of this split with your class 1. Apply your desired classifier (Logistic, Random forest and etc.) on this union, to train a score function.
    Apply scoring function on the **rest** of your random split (bolded), to get scorings.
    Order this scoring.
    Find the score threshold with respect to k* of ordered scores. This is only one threshold among all of the scored. Call it t*.
    Now for a given new incident whenever the score of new data calculated based on the classifier is greater than t*, report as 1, otherwise, report 0.
  Since we have i different answers where $1 < i < M$, we can average all of the answers, and see if average is greater than 0.5 (which is decide by majority.)

The detail of algorithm is in the table:

Algorithm: An NP umbrella algorithm

1. **Input**:
   Training data: A mixed i.i.d. sample $S = S^0 \cup S^1$, where S^0 and S^1 are class 0 and 1 samples, respectively
   α: Type I error upper bound, $0 \leq \alpha \leq 1$; (default α = 0.05)
   δ: A small tolerance level, $0 \leq \delta \leq 1$; (default δ = 0.05)
   M: Number of random splits on S^0 ; (default M = 1)
2. **Function** Rank (n, α, δ)

3.	**For** k in $\{1, \ldots, n\}$ **do**	◂ For each rank threshold candidate k
4.	$V(k) \leftarrow \sum_{j=k}^{n} C_j^n (1-\alpha)^j \alpha^{n-j}$	◂ Calculate the violation rate upper bound
5.	$k^* \leftarrow \min\{k \in \{1, \ldots, n\} : V(k) \leq \delta\}$	◂ Pick the rank threshold
6.	**Return** k^*	
7.	**Procedure** NP $(Sample, \alpha, \delta, M)$	
8.	$n = \lvert S^0 \rvert / 2$	◂ Denote half of the size of |S0|as n
9.	$k^* \leftarrow$ Rank (n, α, δ)	◂ Find the rank threshold
10.	**For** i in $\{1, \ldots, M\}$ **do**	◂ Randomly split S0 for M times
11.	$S_{i,1}^0, S_{i,2}^0 \leftarrow$ random split on S^0	◂ Each time randomly split S0 into two halves with equal sizes
12.	$S_i \leftarrow S_{i,1}^0 \cup S^1$	◂ Combine $S_{i,1}^0$ and S^1
13.	$S_{i,2}^0 = \{x_1, \ldots, x_n\}$	◂ Write $S_{i,2}^0$ as a set of n data points
14.	$f_i \leftarrow$ ClassificationAlgorithm(S_i)	◂ Train a scoring function f_i on S_i
15.	$\tau_i = \{t_{0,1}, \ldots, t_{i,n}\} \leftarrow \{f_i(x_i), \ldots, f_i(x_n)\}$	◂ Apply the scoring function f_i to $S_{i,2}^0$ to obtain a set of score threshold candidates
16.	$\{t_{i,(1)}, \ldots, t_{i,(n)}\} \leftarrow$ sort(τ)	◂ Sort elements of τ_i in an increasing order
17.	$t_i^* \leftarrow t_{i,(k^*)}$	◂ Find the score threshold corresponding to the rank threshold k^*
18.	$\Phi_i(X) = I(f_i(X) > t_i^*$	◂ Construct an NP classifier based on the

scoring function f_i and the threshold t_i^*

19. **Output**: an ensemble NP classifier $\widehat{\Phi}_\alpha(X) = I(\frac{1}{M}\sum_{i=1}^{M}\Phi_i(X) \geq \frac{1}{2})$

◂ By majority vote

We can find that NP-Umbrella algorithm is similar to ROC curve but its more accuracy. Since it splits the train data smaller and smaller to get more accuracy rate. In this method, NP-Umbrella algorithm gets the curve illustrate the overall performance at all possible values of the threshold c on the output classification scores. And its defined as a two dimensional [0, 1] × [0, 1] space whose horizontal and vertical axes correspond to "type I error" (or "false-positive rate") and "1 − type II error" (or "true-positive rate"), respectively. And the area under the curve is to evaluate a classification method and compare different methods.

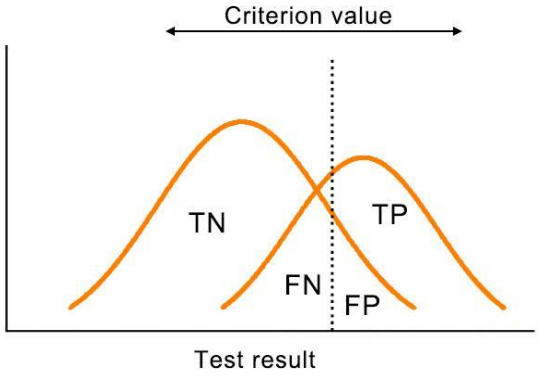

Figure 2 ROC explanation

We applied NP-Umbrella algorithm into Logistic regression, Random forest, and SVM, the one that gives the best performance is selected as our NP-Umbrella algorithm (Version 1).

3.3 compare the accuracy between Baseline model and NP-Umbrella algorithm (Version1):

Comparing these two models' accuracy rate and choose the better one to use in the following experiment.

3.4 Feature selection:

In this part, we compare some popular methods on feature selection, the purpose is to improve the computation efficiency as well as further enhance the prediction accuracy.

3.4.1 Stepwise method:

which belongs to wrapper methods, that is to marry the feature selection process to the type of model being built, evaluating feature subsets in order to detect the model performance between features, and subsequently select the best performing subset. In other words, instead of existing as an independent process taking place prior to model building, step method attempts to optimize feature selection process for a given machine learning algorithm in tandem with this algorithm [7]

In this project, our stepwise feature selection is based on RFE [8]. After data preprocessing, we have 78 features in total, we rank the feature importance when we pick different sets of features and compare the prediction accuracy from picking one feature to picking 78 features. The feature selection is based on the model the better accuracy rate's model, which we get before.

3.4.2 Feature selection based on logistic regression

For the logistic model, we will use the logistic regression model to select variables. First, we put all variables in the model and to test every coefficient whether they are useful, base this test to decide which variable is significant in our model. Next, we drop the variables, which are not significant. Then, we use the new variables to run the logistic model again. And repeating the first step again, until we get the model which conclude that all of variables are significant.

3.5 Train the model with new variables

We applied the feature selection results into the better model (we get before), and build a final version of our model

4 Experiments

4.1 Baseline Model

Our baseline model is Random forest, and the prediction result is 0.707467, the result is show in Table 1.

Table 1 Result of baseline model

	precision	recall	f1-score	support
0	0.88	0.72	0.79	11712
1	0.4	0.66	0.5	3288
avg / total	0.78	0.71	0.73	15000

4.2 Result of NP-Umbrella algorithm (Version1)

In this part, we run NP-Umbrella algorithm with SVM, Random Forest and Logistic regression, Table 2 shows the result, we can see that Np-logistic gives the best overall accuracy.

Table 2 Overall accuracy and Type I error

	Overall Accuracy	Type I error
NP-SVM-Version1	0.7830667	0.01014233
NP-Random Forest-Version1	0.8159333	0.04338191
NP-Logistic-Version1	**0.8216**	0.04602403

Following the figure 2, we can find that shows the relation between type I error upper bound alpha=0.05 and 1-conditional type II error of Np-logistic.

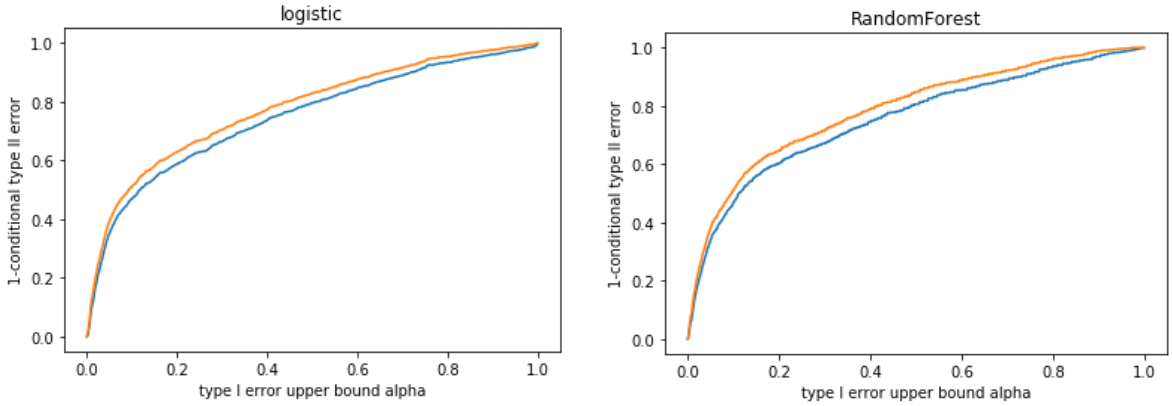

Figure 3 relation between type I error upper bound alpha and 1-conditional type II error.

From figure 3, we compare these two pictures, there is a slight difference between logistic regression and random forest under NP-Umbrella. But we also can see when type I error increase, logistic model' s type II error is greater than random forest model. It also shows that NP-logistic model is better than NP-randomforest.

4.3 Comparing two models' result

For here, we are comparing the 4.1 Baseline model - random forest model and NP-Umbrella model. The NP-Umbrella model's accuracy rate is much better than random forest model, since all of models' accuracy rate, which are under NP- umbrella are higher than random forest model's accuracy rate, which is 0.707. So, we will choose the Np-umbrella method in our following experiments. In the next step, we will do variables selection under Np-umbrella method.

4.4 Result of Feature Selection

We conduct feature selection based on model NP-Umbrella algorithm(Version1), we use two methods to select the features, as feature selection is model oriented, therefore, with stepwise method, we select features based on model SVM and Random forest, features that selected are applied into NP-SVM algorithm (Version 2), and NP-Random forest algorithm (Version 2). With logistic method, the features that we selected is applied into NP-Logistic algorithm (Version2).

4.4.1 Stepwise method

There are 78 features after data preprocessing: *['X1', 'X5', 'X12', 'X13', 'X14', 'X15', 'X16', 'X17', 'X18', 'X19', 'X20', 'X21', 'X22', 'X23', 'Y', 'X2_1', 'X3_1', 'X3_2', 'X3_3', 'X4_1', 'X4_2', 'X6_-2', 'X6_-1', 'X6_1', 'X6_2', 'X6_3', 'X6_4', 'X6_5','X6_6', 'X6_7', 'X6_8', 'X7_-2', 'X7_-1', 'X7_1', 'X7_2', 'X7_3', 'X7_4', 'X7_5', 'X7_6', 'X7_7', 'X7_8', 'X8_-2', 'X8_-1', 'X8_1', 'X8_2', 'X8_3', 'X8_4', 'X8_5', 'X8_6', 'X8_7', 'X8_8', 'X9_-2', 'X9_-1', 'X9_1', 'X9_2', 'X9_3', 'X9_4', 'X9_5', 'X9_6', 'X9_7', 'X9_8', 'X10_-2', 'X10_-1', 'X10_2', 'X10_3', 'X10_4', 'X10_5', 'X10_6','X10_7', 'X10_8', 'X11_-2', 'X11_-1', 'X11_2', 'X11_3', 'X11_4', 'X11_5', 'X11_6', 'X11_7', 'X11_8']*

In Figure 4, it shows the ‘Correlation with quality’

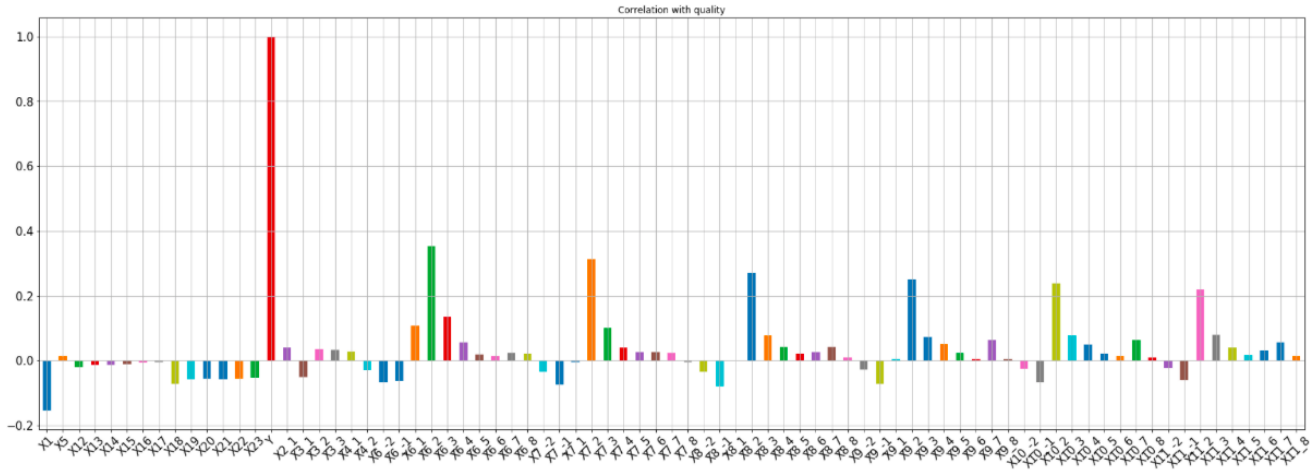

Figure 4 Correlation with quality

4.4.1.1 Feature selection based on SVM

We input the different sets of features in to the model, and figure 5 shows the accuracy change when picking different sets of features, from picking one feature to pick all of the features.

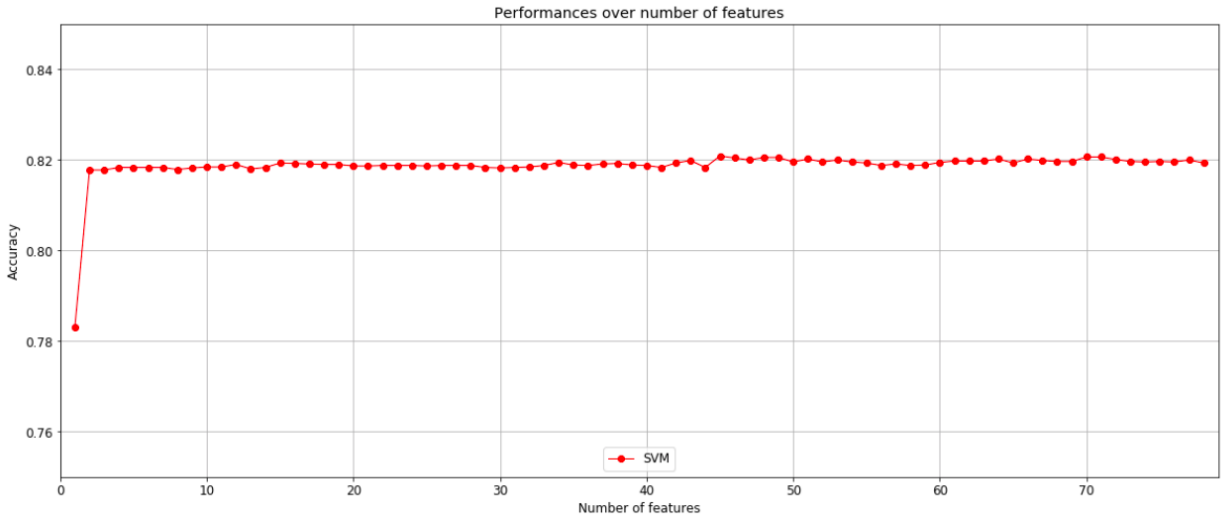

Figure 5 Performance over number of features based on SVM

From figure 5, we can see that when we pick 45 or 70 features, it gives the best prediction accuracy:

When we pick 45 features:
[X5', 'X12', 'X13', 'X16', 'X18', 'X19', 'X20', 'X21', 'X22', 'X2_1', 'X3_1', 'X3_2', 'X3_3', 'X4_1', 'X4_2', 'X6_-2', 'X6_-1', 'X6_1', 'X6_2', 'X6_3', 'X6_4', 'X7_-2', 'X7_-1', 'X7_2', 'X7_3', 'X8_-2', 'X8_-1', 'X8_2', 'X8_3', 'X9_-2', 'X9_-1', 'X9_2', 'X9_3', 'X9_4', 'X9_7', 'X10_-2', 'X10_-1', 'X10_2', 'X10_3', 'X10_7', 'X11_-2', 'X11_-1', 'X11_2', 'X11_3', 'X11_7']

When we pick 70 features:

['X1', 'X5', 'X12', 'X13', 'X14', 'X15', 'X16', 'X17', 'X18', 'X19', 'X20', 'X21', 'X22', 'X23', 'X2_1', 'X3_1', 'X3_2', 'X3_3', 'X4_1', 'X4_2', 'X6_-2', 'X6_-1', 'X6_1', 'X6_2', 'X6_3', 'X6_4', 'X6_5', 'X6_6', 'X6_7', 'X6_8', 'X7_-2', 'X7_-1', 'X7_2', 'X7_3', 'X7_4', 'X7_5', 'X7_6', 'X7_7', 'X8_-2', 'X8_-1', 'X8_2', 'X8_3', 'X8_4', 'X8_5', 'X8_6', 'X8_7', 'X9_-2', 'X9_-1', 'X9_2', 'X9_3', 'X9_4', 'X9_5', 'X9_7', 'X10_-2', 'X10_-1', 'X10_2', 'X10_3', 'X10_4', 'X10_5', 'X10_6', 'X10_7', 'X11_-2', 'X11_-1', 'X11_2', 'X11_3', 'X11_4', 'X11_5', 'X11_6', 'X11_7', 'X11_8']

4.4.1.2 Feature selection based on Random Forest

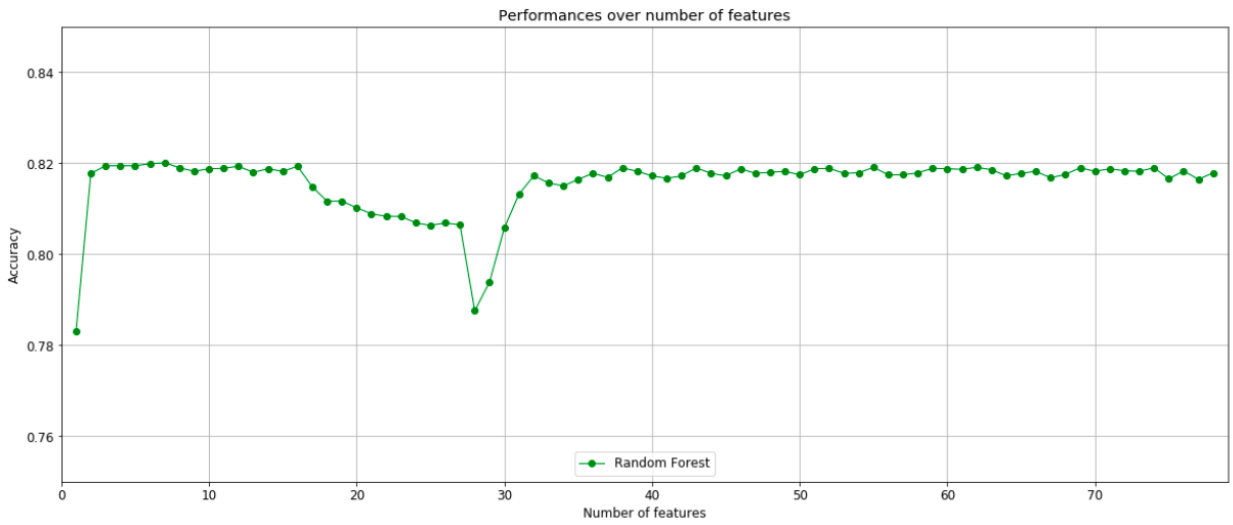

Figure 6 Performance over number of features based on Random Forest

From figure 6, we can see that when we pick 7 features, it gives the best prediction accuracy:
['X3_1', 'X3_2', 'X3_3', 'X6_1', 'X6_2', 'X6_3', 'X7_3']

4.4.2 Logistic method

Following appendix, it shows a step by step result of selecting features based on logistic method. By the result, we drop x_trainX4m, x_trainX4s, x_trainX5 ,x_trainX6pre, x_trainX6pd6, x_trainX6pd7, x_trainX6pd8, x_trainX7pre, x_trainX7pd1,x_trainX7pd3, *x_trainX7pd5, x_trainX7pd6, x_trainX7pd7,* x_trainX7pd8, x_trainX8pre, x_trainX8pd, x_trainX8pd1, x_trainX8pd3, x_trainX8pd4, x_trainX8pd5, x_trainX8pd6, x_trainX8pd7, x_trainX8pd8, x_trainX9pre, x_trainX9pd, x_trainX9pd3, x_trainX9pd1, x_trainX9pd4, x_trainX9pd5, x_trainX9pd6, x_trainX9pd7, x_trainX9pd8, x_trainX9pre, x_trainX9pd, x_trainX9pd1, x_trainX9pd3, x_trainX9pd4, x_trainX9pd5, x_trainX9pd6, x_trainX9pd7, x_trainX12 , x_trainX9pd8, x_trainX10pre, x_trainX10pd, x_trainX10pd1, x_trainX10pd2, x_trainX10pd3, x_trainX10pd4, x_trainX10pd5, x_trainX10pd6, x_trainX10pd7, x_trainX10pd8, x_trainX11pd4, x_trainX11pd5, x_trainX11pd6, x_trainX11pd7, x_trainX11pd8, x_trainX13, x_trainX14, x_trainX15, x_trainX16, x_trainX17, x_trainX20, x_trainX21, x_trainX22, x_trainX23
Under these variables' coefficients tests', P-value is greater than 0.05, that's means they are not significant in this model, so we drop these variables and do the regression again.

In appendix, we omit mid-process. For mid-process, and we repeat the first step until all variables are significant in the model. And the final result is also given in the appendix.

The feature that we select are: Variable X1,X2, X3u,X3hs, X3ot, X5, X6pd, X6pd1, X6pd2, X6pd3, X9pd2, X11pre,X11pd2, X11pd3, X18,X19 and they are significant in our model.
Following the final result and our data describe, X1: Amount of the given credit; X18: amount of previous payment paid in September, 2005; X19: amount of previous payment paid in August, 2005 have negative effect on the default payment(Y). X2: Gender; X3: Education; X5: Age ; X6pred : the repayment status is prepayment in September, 2005; X6pd1: the repayment status is payment delay for one month in September, 2005; X6pd2: the repayment status is payment delay for two months in September, 2005; X6pd3: the repayment status is payment delay for three months in September, 2005; X9pd2: the repayment status is payment delay for two months in June, 2005; X11pre: the repayment status is prepayment in April, 2005 ; X11pd2: the repayment status is payment delay for two months in April, 2005; X11pd3: the repayment status is payment delay for three months in April, 2005 have positive effect on the default payment(Y).

4.5 Result of NP-Umbrella algorithm (Version2)

As we have mentioned in 4.3(Feature selection part), features selected based on different models are applied into the corresponding NP-Umbrella algorithm (Version2) models.

4.5.1 NP-Logistic (Version2)

We put the features selected from Logistic method into our NP-logistic algorithm (Version1) to build NP-Logistic (Version2). Figure 6 shows the relation of FP and TP with NP-Logistic (Version2). We can see from the result, with feature selection, the overall accuracy decreased from 0.8216 (NP-Logistic-Version1) to 0.8183333(NP-Logistic-Version2).

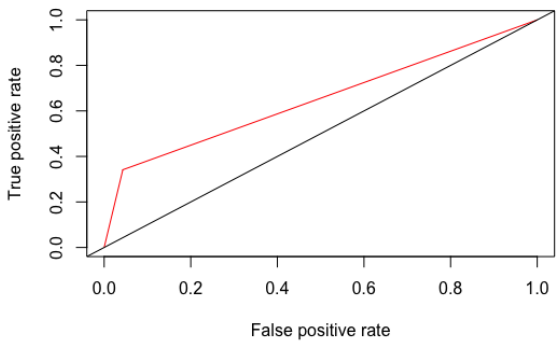

Figure 7 Relation between FP and TP

Overall Accuracy for logistic: 0.8183333
Type I error for logistic: 0.04310419

So, comparing NP-Logistic (Version2) with NP-logistic algorithm (Version1), there is no improvement.

4.5.2 NP-SVM(Version2)

We put the features selected from SVM based step wise method, from the result of feature selection, when picking 45 or 70 features, the performance is better, therefore, here, we build two NP-SVM(Version2), one is based on 45 feature, the other is based on 70 features.

4.5.2.1 Variable(45)-NP-SVM(Version2)

We use the 45 features that we have chosen, Figure 7 shows the relation of FP and TP of variable(45)-NP-SVM (Version2)**.** We can see from the result, with feature selection, the overall accuracy raised from 0.7830667 (NP-SVM-Version1) to 0.8120667 (Variable (45) - NP - SVM - Version2).

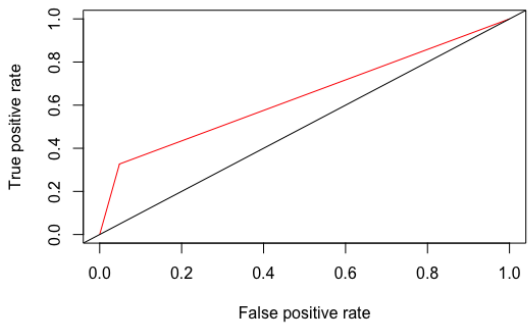

Figure 8 Relation between FP and TP

Overall Accuracy for svm: 0.8120667
Type I error for svm: 0.03871909

4.5.2.2 Variable (70)-NP-SVM-Version2

We use the 70 features that we have chosen. Figure 8 shows the relation of FP and TP of variable (70) NP-SVM(Version2)**.** We can see from the result, with feature selection, the overall accuracy raised from 0.7830667 (NP-SVM-Version1) to 0.8132667 (Variable (70) -NP -SVM - Version2).

Overall Accuracy for svm: 0.8132667
Type I error for svm: 0.03868869

So, by these two results, the variables selection is useful to improve the accuracy rate. Also, comparing 4.5.2.1 and **4.5.2.2** 's result, we can find the **4.5.2.2** has the better result.

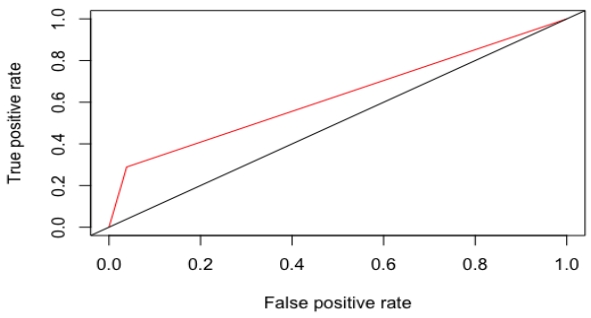

Figure 9 Relation between FP and TP

4.5.3 NP-Random Forest-Version2

We use the features that we have chosen from Random Forest based step wise method. Figure 9 shows the relation of FP and TP of NP-Random Forest-Version2.
We can see from the result, with feature selection, the overall accuracy raised from 0.8159333 (NP-Random Forest-Version1) to 0.8233 (NP-Random Forest-Version2)

Overall Accuracy: 0.8233
Type I error: 0.03871909

As random forest is computing intensive, with feature selection, we only select 7 features, which effectively reduce the total running time; At the same time, we have effectively raised its overall accuracy, which is higher than the accuracy in research work [3].

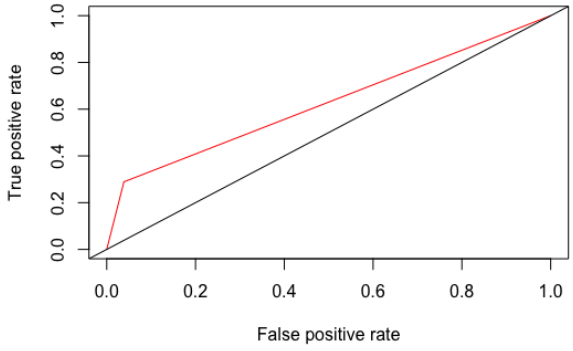

Figure 10 Relation between FP and TP

Table 3 shows the comparison between the best accuracy of baseline model, NP-Umbrella-Version 1 and UP-Umbrella-Version2.

Table 3 Overall accuracy and Type I error

	Model gives the best overall accuracy	Overall accuracy
Baseline model	Random Forest	0.707467
NP-Umbrella-Version1	NP-Logistic-Version1	0.8216
NP-Umbrella-Version2	NP-Random Forest Version2	**0.8233**

5. Conclusions

We firstly conduct a research survey with the credit default prediction problem, and based on the survey result, we set Random Forest as our baseline model. Based on the research conclusion that 'NP Umbrella' can effectively raise the accuracy classification in work[6], we apply 'NP Umbrella' approach into credit default problem, which is never done before, we combined 'NP-Umbrella' with Random Forest, SVM, and Logistic Regression models, we made comparison among the baseline model with NP-Logistic-Version1, NP-Random forest-Version1, and NP-SVM-Version1, after comparison, NP-Logistic(Version1) gives the best performance, therefore it is chosen as our 'NP-Umbrella(Version1)'.

As feature selection is model and data oriented, we use stepwise method to select features based on model SVM and Random forest, and features selected are applied into NP-SVM(Version1), and NP-Random Forest (Version1) respectively to build the new model versions; With logistic method, the features that we selected is applied into NP-Logistic(Version1).

We build new model by applying feature selection into our NP-Umbrella(Version1), our new model is called 'NP-Umbrella(Version2)', after comparing NP-Logistic(Version2), NP-Random forest(Version2) and NP-SVM(Version2), the result shows that NP-Random forest (Version2) gives the best prediction accuracy, and by feature selection, we decrease the number of features from 78 to 7, which then effectively improve the computing efficiency.

From this project, we have practiced what we have learnt in machine learning class, and we get a deeper and better understanding of the concept with data preprocessing, model selection and feature engineering, these skills are helpful for our future study.

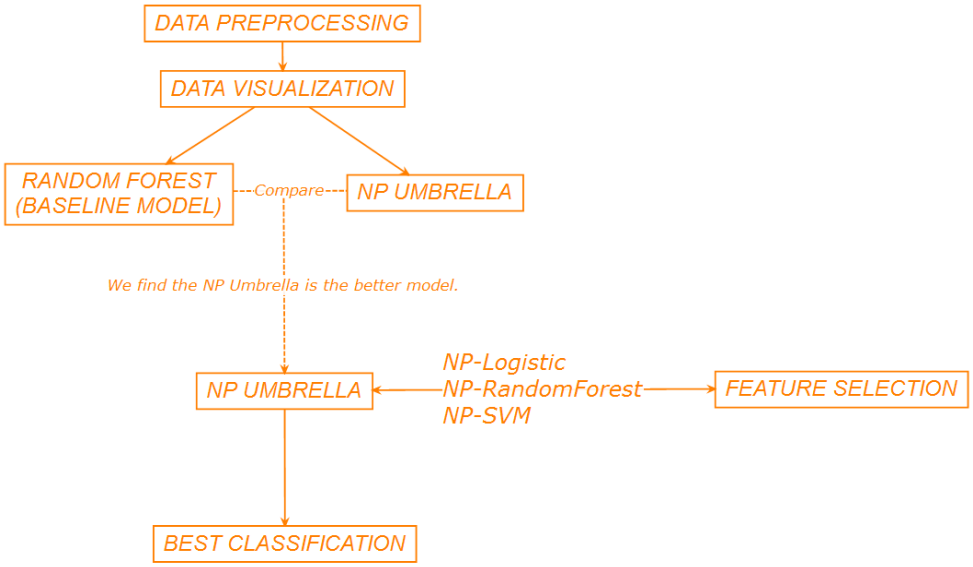

Figure 11 The final flow chart of the approach

6. Response to the feedback

We have addressed most of Dr. Lee's feedback in previous description of this final report.

In this part, we mainly give more explanation with our novelties that we have proposed in our mid-progress report:

The paper '*Neyman-Pearson classification algorithms and NP receiver operating characteristics*' published in '*Science Advances'* is a newly proposed paper that aim to raise the prediction accuracy with most of the classification methods, and the Paper that Dr. Lee mentioned in the feedback is '*Neyman-Pearson classification: parametrics and power enhancement*' which is written by Xia and Tong etc.

- Our approach is based on above two papers, and we apply both feature selection process and this new method NP-Umbrella into our credit default problem with the purpose of enhancing the prediction accuracy, which is never done before in this field.

- In order to further improve the computing efficiency and enhance the prediction accuracy, we use two feature selection methods for feature engineering, which are based on the model and the dataset.

- In summary, our novelty is we try to build a credit default prediction system that based on NP-Umbrella and Feature selection, that can give a higher prediction accuracy and decrease the computing time.

The roles for each team member are depicted in Table 4.

Table 4 Team member roles

	Ali	Yunfei	Hongyuan	Yuehua
Topic selection	✓	✓	✓	✓
Proposal	✓	✓	✓	✓
Approaches Discussion	✓	✓	✓	✓
Model Selection	✓	✓	✓	✓
Mid-report	✓	✓	✓	✓
Research on NP	✓			
Run Data on Model		✓		
Data Preprocessing			✓	✓
Explanation of Results		✓	✓	✓
Further Research		✓	✓	✓
Written Report	✓	✓	✓	✓
Poster		✓	✓	✓

References

[1] Xiong, Tengke, et al. "Personal bankruptcy prediction by mining credit card data." Expert systems with applications 40.2 (2013): 665-676.
[2] Liang, Deron, et al. "A novel classifier ensemble approach for financial distress prediction." Knowledge and Information Systems 54.2 (2018): 437-462.

[3] Lu, Hongya, Haifeng Wang, and Sang Won Yoon. "Real Time Credit Card Default Classification Using Adaptive Boosting-Based Online Learning Algorithm." IIE Annual Conference. Proceedings. Institute of Industrial and Systems Engineers (IISE), 2017.

[4] Islam, S. R., Eberle, W., & Ghafoor, S. K. (2018). Credit Default Mining Using Combined Machine Learning and Heuristic Approach. arXiv preprint arXiv:1807.01176.

[5] Yeh, I-Cheng, and Che-hui Lien. "The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients." Expert Systems with Applications 36.2 (2009): 2473-2480. J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[6] Tong, X., Feng, Y., & Li, J. J. (2018). Neyman-Pearson classification algorithms and NP receiver operating characteristics. Science advances, 4(2), eaao1659.

[7] https://www.kdnuggets.com/2018/06/step-forward-feature-selection-python.html

[8] https://www.kaggle.com/tasalta/credit-default-analysis-over-different-sets

[9]https://www.pymnts.com/news/payment-methods/2017/credit-card-debit-card-payments-federal-reserve-remote-payments/

[10] Tong, X, Xia, L Wang, J (2018). Neyman-Pearson classification: parametrics and power enhancement. stat.

Appendix

Coefficients: (5 not defined because of singularities)

	Estimate	Std. Error	z value	Pr(>|z|)	
(Intercept)	-2.695e+00	3.450e-01	-7.811	5.66e-15	***
x_trainX1	-2.137e-06	2.528e-07	-8.454	< 2e-16	***
x_trainX2	7.876e-02	4.617e-02	1.706	0.088009	.
x_trainX3u	7.225e-01	2.530e-01	2.856	0.004295	**
x_trainX3hs	8.036e-01	2.516e-01	3.194	0.001404	**
x_trainX3ot	7.629e-01	2.555e-01	2.985	0.002831	**
x_trainX4m	9.230e-02	2.119e-01	0.436	0.663163	
x_trainX4s	-7.633e-02	2.141e-01	-0.356	0.721479	
x_trainX5	3.334e-03	2.806e-03	1.188	0.234881	
x_trainX6pre	2.665e-01	1.683e-01	1.584	0.113274	
x_trainX6pd	8.018e-01	1.100e-01	7.286	3.19e-13	***
x_trainX6pd1	1.130e+00	1.205e-01	9.376	< 2e-16	***
x_trainX6pd2	2.289e+00	9.529e-02	24.026	< 2e-16	***
x_trainX6pd3	2.519e+00	2.283e-01	11.037	< 2e-16	***
x_trainX6pd4	2.230e+00	4.419e-01	5.046	4.51e-07	***
x_trainX6pd5	2.304e+00	7.569e-01	3.044	0.002338	**
x_trainX6pd6	1.433e+00	1.117e+00	1.283	0.199367	
x_trainX6pd7	2.849e+00	1.954e+00	1.458	0.144892	
x_trainX6pd8	2.203e+00	9.565e+02	0.002	0.998162	
x_trainX7pre	-2.142e-01	1.984e-01	-1.080	0.280305	
x_trainX7pd	-3.143e-01	1.311e-01	-2.397	0.016517	*
x_trainX7pd1	-1.098e+00	1.083e+00	-1.013	0.310844	
x_trainX7pd2	-5.850e-02	1.167e-01	-0.501	0.616079	
x_trainX7pd3	1.100e-01	2.234e-01	0.492	0.622473	
x_trainX7pd4	-1.166e+00	4.841e-01	-2.409	0.015990	*
x_trainX7pd5	4.455e-01	9.818e-01	0.454	0.649981	
x_trainX7pd6	-2.788e+00	2.171e+00	-1.284	0.199136	
x_trainX7pd7	NA	NA	NA	NA	
x_trainX7pd8	-2.789e+01	7.572e+02	-0.037	0.970614	
x_trainX8pre	3.771e-02	1.791e-01	0.211	0.833271	
x_trainX8pd	4.976e-02	1.206e-01	0.413	0.679860	
x_trainX8pd1	-1.122e+01	5.354e+02	-0.021	0.983288	
x_trainX8pd2	3.590e-01	9.129e-02	3.932	8.41e-05	***
x_trainX8pd3	5.451e-02	2.867e-01	0.190	0.849186	
x_trainX8pd4	-2.764e-01	5.791e-01	-0.477	0.633201	
x_trainX8pd5	1.938e+00	1.527e+00	1.269	0.204375	
x_trainX8pd6	1.637e+00	9.565e+02	0.002	0.998634	
x_trainX8pd7	1.699e+00	1.232e+00	1.379	0.167849	
x_trainX8pd8	-2.468e+01	5.845e+02	-0.042	0.966313	
x_trainX9pre	1.223e-01	1.727e-01	0.708	0.478816	
x_trainX9pd	-3.473e-02	1.096e-01	-0.317	0.751322	

```
x_trainX9pd1   2.596e+01  7.572e+02   0.034 0.972646
x_trainX9pd2   3.353e-01  9.741e-02   3.442 0.000578 ***
x_trainX9pd3   3.215e-01  3.044e-01   1.056 0.290967
x_trainX9pd4   3.998e-01  6.373e-01   0.627 0.530485
x_trainX9pd5  -1.895e+00  1.383e+00  -1.370 0.170648
x_trainX9pd6   1.266e+01  5.354e+02   0.024 0.981139
x_trainX9pd7   1.225e+01  2.344e+02   0.052 0.958303
x_trainX9pd8          NA         NA      NA       NA
x_trainX10pre -2.781e-01  1.653e-01  -1.682 0.092525 .
x_trainX10pd  -1.601e-01  1.062e-01  -1.507 0.131842
x_trainX10pd2  1.582e-01  1.034e-01   1.530 0.126116
x_trainX10pd3  1.555e-01  3.137e-01   0.496 0.620214
x_trainX10pd4 -6.128e-01  7.014e-01  -0.874 0.382232
x_trainX10pd5  7.379e-01  1.140e+00   0.647 0.517485
x_trainX10pd6 -8.683e-01  5.845e+02  -0.001 0.998815
x_trainX10pd7         NA         NA      NA       NA
x_trainX10pd8         NA         NA      NA       NA
x_trainX11pre  5.316e-01  1.228e-01   4.329 1.50e-05 ***
x_trainX11pd   2.331e-01  9.566e-02   2.437 0.014809 *
x_trainX11pd2  4.332e-01  8.949e-02   4.840 1.30e-06 ***
x_trainX11pd3  9.878e-01  3.290e-01   3.002 0.002678 **
x_trainX11pd4 -1.254e-01  6.898e-01  -0.182 0.855789
x_trainX11pd5  9.711e-02  1.075e+00   0.090 0.928014
x_trainX11pd6 -2.378e-01  1.432e+00  -0.166 0.868144
x_trainX11pd7 -1.209e+01  2.344e+02  -0.052 0.958878
x_trainX11pd8         NA         NA      NA       NA
x_trainX12    -1.872e-07  1.442e-06  -0.130 0.896673
x_trainX13     1.309e-06  1.912e-06   0.684 0.493858
x_trainX14     3.157e-06  1.710e-06   1.847 0.064819
x_trainX15    -1.452e-06  1.799e-06  -0.808 0.419376
x_trainX16     1.920e-06  2.121e-06   0.905 0.365556
x_trainX17    -1.995e-06  1.621e-06  -1.231 0.218410
x_trainX18    -1.207e-05  3.437e-06  -3.511 0.000446 ***
x_trainX19    -1.592e-05  3.722e-06  -4.278 1.89e-05 ***
x_trainX20     2.094e-06  1.935e-06   1.082 0.279197
x_trainX21    -4.846e-06  2.751e-06  -1.762 0.078104
x_trainX22    -2.461e-06  2.547e-06  -0.966 0.333939
x_trainX23    -9.564e-07  1.635e-06  -0.585 0.558589
```

Coefficients:

	Estimate	Std. Error	z value	Pr(>|z|)	
(Intercept)	-3.108e+00	2.920e-01	-10.644	< 2e-16	***
x_train1X1	-1.482e-06	2.118e-07	-6.995	2.65e-12	***
x_train1X2	1.555e-01	4.505e-02	3.453	0.000555	***
x_train1X3u	1.116e+00	2.785e-01	4.007	6.15e-05	***

*x_train1X3hs 1.164e+00 2.775e-01 4.194 2.74e-05 ****
*x_train1X3ot 1.101e+00 2.808e-01 3.922 8.79e-05 ****
*x_train1X5 6.894e-03 2.458e-03 2.804 0.005040 ***
*x_train1X6pd 4.086e-01 6.763e-02 6.042 1.52e-09 ****
*x_train1X6pd1 9.819e-01 6.225e-02 15.773 < 2e-16 ****
*x_train1X6pd2 2.366e+00 7.172e-02 32.995 < 2e-16 ****
*x_train1X6pd3 2.680e+00 1.913e-01 14.012 < 2e-16 ****
*x_train1X6pd4 2.126e+00 3.637e-01 5.846 5.03e-09 ****
x_train1X6pd5 8.583e-01 6.233e-01 1.377 0.168519
x_train1X6pd7 1.397e+00 1.432e+00 0.976 0.328962
*x_train1X9pd2 5.315e-01 7.088e-02 7.500 6.40e-14 ****
*x_train1X11pre 2.373e-01 6.471e-02 3.666 0.000246 ****
x_train1X11pd -3.993e-02 6.887e-02 -0.580 0.562102
*x_train1X11pd2 5.940e-01 7.655e-02 7.759 8.58e-15 ****
*x_train1X11pd3 9.843e-01 2.477e-01 3.975 7.05e-05 ****
x_train1X11pd4 -9.166e-02 4.229e-01 -0.217 0.828415
*x_train1X18 -5.847e-06 2.361e-06 -2.476 0.013269 **
*x_train1X19 -1.008e-05 2.571e-06 -3.921 8.81e-05 ****

……

The final result is
Coefficients:

	Estimate	Std. Error	z value	Pr(>|z|)	
(Intercept)	*-3.065e+00*	*2.887e-01*	*-10.616*	*< 2e-16*	***
x_train3X1	*-2.013e-06*	*2.126e-07*	*-9.471*	*< 2e-16*	***
x_train3X2	*1.743e-01*	*4.476e-02*	*3.894*	*9.88e-05*	***
x_train3X3u	*1.077e+00*	*2.768e-01*	*3.891*	*9.96e-05*	***
x_train3X3hs	*1.096e+00*	*2.758e-01*	*3.975*	*7.04e-05*	***
x_train3X3ot	*1.045e+00*	*2.792e-01*	*3.743*	*0.000182*	***
x_train3X5	*9.273e-03*	*2.428e-03*	*3.819*	*0.000134*	***
x_train3X6pd	*3.913e-01*	*6.078e-02*	*6.439*	*1.21e-10*	***
x_train3X6pd1	*9.480e-01*	*6.158e-02*	*15.394*	*< 2e-16*	***
x_train3X6pd2	*2.298e+00*	*7.058e-02*	*32.558*	*< 2e-16*	***
x_train3X6pd3	*2.584e+00*	*1.923e-01*	*13.434*	*< 2e-16*	***
x_train3X9pd2	*5.844e-01*	*6.975e-02*	*8.379*	*< 2e-16*	***
x_train3X11pre	*2.217e-01*	*6.160e-02*	*3.599*	*0.000320*	***
x_train3X11pd2	*5.111e-01*	*7.460e-02*	*6.851*	*7.32e-12*	***
x_train3X11pd3	*1.357e+00*	*2.579e-01*	*5.262*	*1.42e-07*	***
x_train3X18	*-5.434e-06*	*2.249e-06*	*-2.417*	*0.015654*	*
x_train3X19	*-1.076e-05*	*2.794e-06*	*-3.850*	*0.000118*	***