Loan Quality Prediction Based on Borrowers' Information From the Irish Dummy Banks in Lending Club

Zhihui Chen 118010036
Peiwen Xiao 118020446
Yuxin Xiao 118020446
Leyang Sun 119010272

Abstract

Since net interest margin and asset quality are the key drivers for the income of banks, loan quality is of extreme importance. Being supported by big data, banks nowadays can use statistical models to evaluate the quality of the loans, which means whether the borrower will pay back the loan in time or not. This project explored statistical loan-quality-prediction models and tried to find the model generates the best results from multidimensional personal information input of the borrowers. The results were judged in two ways: the general prediction accuracy and a self-defined 'score', which is identified based on our research. The data source is borrowers' information from the Irish Dummy Bank in Lending Club, which is one of the most reputable destinations for online personal loans. There were over eight hundred thousand observations in raw data set and we selected around a quarter of them for model building, model selection and validation. We used three parametric models: logistic regression, linear discrimination analysis and quadratic discrimination analysis and five non-parametric models: KNN, SVM, decision tree, random forest and bagging. By evaluating all these fin-tuned models on the validation set, the final model we chose is random forest.

1. Introduction

Releasing liquidity and absorbing deposits are the major functions of commercial banks. The difference of their interests, the net interest margin, is the main source of banks' income. An important risk in this kind of profit model is non-performing loans, which means the borrowers fail to pay banks back due to variable reasons.

In this data set, non-performing loans are represented by loan-outcome=0. Standing in banks' shoes, non-performing loans are immensely damaging:they not only brings direct loss in income but also increases the risks faced by banks. So prediction models should be used to help decide whether a bank lends to a borrower. The model learns from historical data and given several information about new borrowers, it can predict things like the probability that the borrower will pay back the money.Since this is a real-world question, the final choice of the model will not only based on the direct results of the models. More detail of the decision rules are in the 'Methods' part.

This project tried to find a model, based on a huge data set, which benefits banks the most. The data set is based upon Lending Club (LC) Information. The Irish Dummy Banks is a peer to peer lending bank based in the Ireland, in which bank provide funds for potential borrowers and bank earn a profit depending on the risk they take (the borrowers credit score). There are 887,379 observations in the original data set and 74 columns in the original data set (as shown below). Obviously, not all of them are useful in this classification task, and we will choose several relative predictors in the data preprocessing part. 'Loan_status' was chosen as the response of this prediction task.

```
##  [1] "id"                          "member_id"
##  [3] "loan_amnt"                   "funded_amnt"
##  [5] "funded_amnt_inv"             "term"
##  [7] "int_rate"                    "installment"
##  [9] "grade"                       "sub_grade"
## [11] "emp_title"                   "emp_length"
## [13] "home_ownership"              "annual_inc"
## [15] "verification_status"         "issue_d"
## [17] "loan_status"                 "pymnt_plan"
## [19] "url"                         "desc"
## [21] "purpose"                     "title"
## [23] "zip_code"                    "addr_state"
## [25] "dti"                         "delinq_2yrs"
## [27] "earliest_cr_line"            "inq_last_6mths"
## [29] "mths_since_last_delinq"      "mths_since_last_record"
## [31] "open_acc"                    "pub_rec"
## [33] "revol_bal"                   "revol_util"
## [35] "total_acc"                   "initial_list_status"
## [37] "out_prncp"                   "out_prncp_inv"
## [39] "total_pymnt"                 "total_pymnt_inv"
## [41] "total_rec_prncp"             "total_rec_int"
## [43] "total_rec_late_fee"          "recoveries"
## [45] "collection_recovery_fee"     "last_pymnt_d"
## [47] "last_pymnt_amnt"             "next_pymnt_d"
## [49] "last_credit_pull_d"          "collections_12_mths_ex_med"
## [51] "mths_since_last_major_derog" "policy_code"
## [53] "application_type"            "annual_inc_joint"
## [55] "dti_joint"                   "verification_status_joint"
## [57] "acc_now_delinq"              "tot_coll_amt"
## [59] "tot_cur_bal"                 "open_acc_6m"
## [61] "open_il_6m"                  "open_il_12m"
## [63] "open_il_24m"                 "mths_since_rcnt_il"
## [65] "total_bal_il"                "il_util"
## [67] "open_rv_12m"                 "open_rv_24m"
## [69] "max_bal_bc"                  "all_util"
## [71] "total_rev_hi_lim"            "inq_fi"
## [73] "total_cu_tl"                 "inq_last_12m"
```

Figure 1: Columns in the orginal dataset

2. Materials and methods

2.1 Data preprocessing, exploratory analysis and standardization

2.1.1 Data preprocessing

First we found that the original data set from the bank contains 74 columns, which are not necessarily useful and relevant in regression model.

Considering the goal of this report is to predict loan default with data available before the loan has been approved by Lending Club, many features such as "current balance", "last_pymnt_d(last payment day)", "next_pymnt_d(last payment day)", "recovery", could not be used by the model. We exclude these variables in the preliminary feature selection step. The following are variables that are selected, which are expected to have significance in predicting default rate.

```
loan = loan %>%
        select(loan_status, loan_amnt , term , int_rate , installment , grade , emp_length , home_ownership,annua
l_inc, dti, purpose,installment )
loan
```

Figure 2: Relavant predictors

Then we cleaned up term and emp_length variables by removing string characters in term and emp_length, and encode emp_length to rank 1-10 (where "10+" is encoded into 10). Then we encode loan_status to binary outcomes: loan_outcome, with the following criterion:

- The observations under "current" can not be used in learning and should be left out;

- Most of the observation are labeled as fully paid or charged off;
- "Fully paid" is 1; "Charged off", "Late (31-120 days)", "Late (16-30days)", "Default", and "in grace period" are 0.

  Finally, we replace the missing values in the emp_length with the mean.

2.1.2 Data Exploratory analysis

We did a data exploratory analysis after the above basic operations on the data.

First we check if this is a balanced data set.

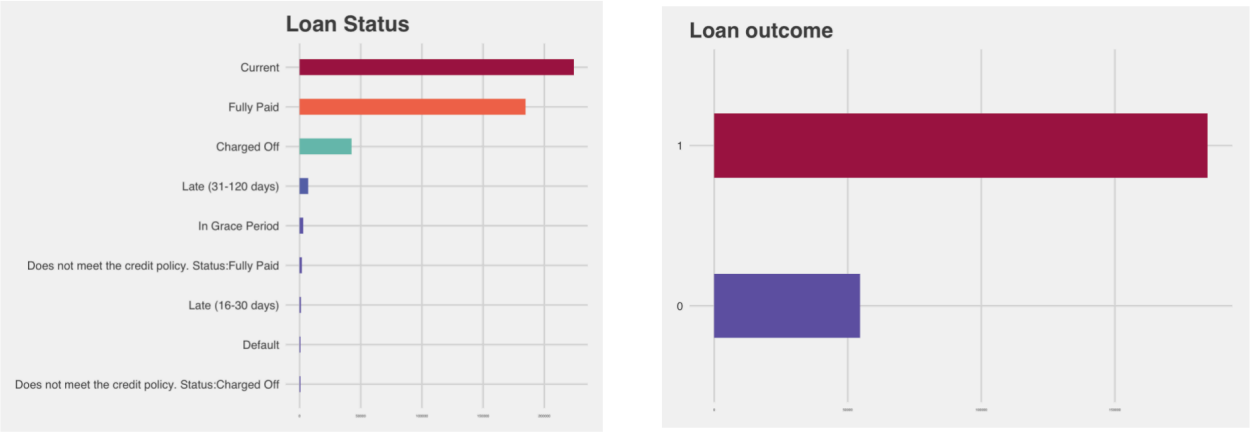
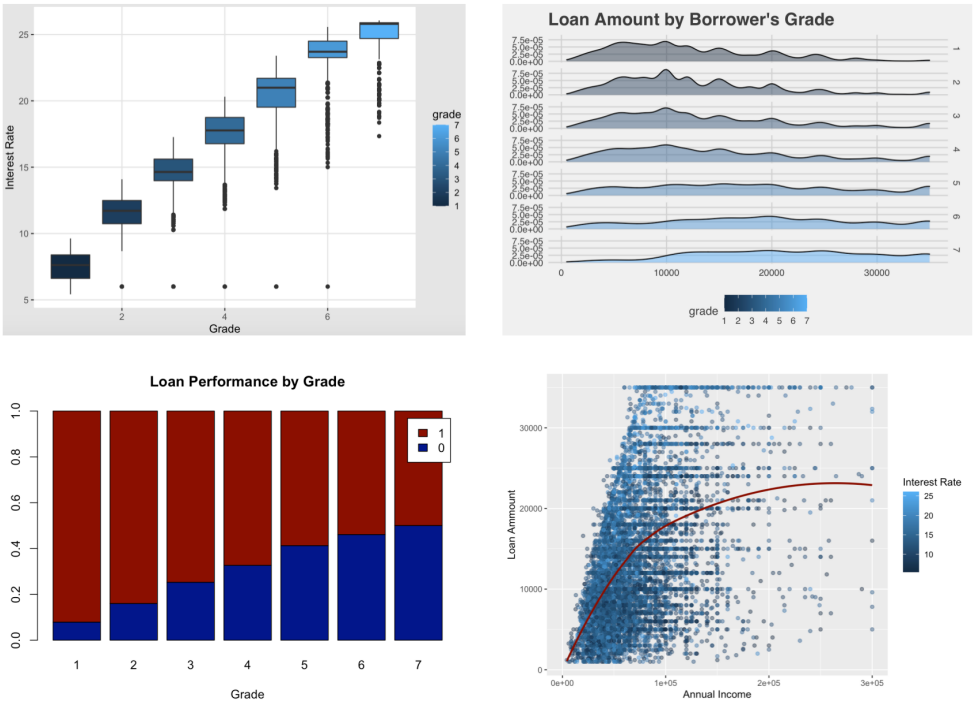


Figure 3: Class imbalance

We found that observations under loan_outcome = 1 is much more than loan_outcome = 0, which means this dataset is highly imbalanced. To take care of this problem, we use the smote package later to balance our training data set.

Then we check correlations between several variables:



From the plots we get the following conclusions:

1) Feature **"grade"** is a variable that highly correlated with almost all other variables, especially the interest rate and loan amount.

2) Other variables also have high correlations and we might consider drop and use different combinations of them when selecting certain features to build the model.
   The charts shows how grade affects several important variables and the final outcome and how **"annual_income"** correlates with **"loan_amount"**.

2.1.3 Data standardization

Since each numerical variable is measured on different scale and some distance-based model like knn will be used later, we standardized the raw data set to set features on a similar scale.

2.1.4 Data set splitting and smote on the training set

We splitted the whole data set into three parts, train set, validation set and test set. The train set was used for training different types of models and tuning parameters through k-fold cross validation, and we selected one representative final model of each type and compare how they performed on the validation set. After compare all the best models under each type, we decided which type and the corresponding best parameters to use in the end, and built a final model on the whole training + validation set.

From the EDA, we have detected high imbalance in our data set. This is a serious issue since if we simply use the imbalance training set to training the data and use the general metrics (i.e. accuracy rate), models will only predicting 1, which means it's completely ignoring the minority class in favor of the majority class.

We considered taking the following strategy —— up-sampling minority class and down-sampling majority class by SMOTE. On the training set, we used SMOTE package to handle class imbalance by up-sampling minority class and down-sampling majority class. After re-sampling, we get a new training data set containing same amount of class 1 instance and class 0 instance. We keep the validation set and test set unchanged since these two sets would be used to estimate the prediction error of new data, and they should reserve the original ratio of class 0 and 1.

```
balanced.train_data <-
  DMwR::SMOTE(
    form = loan_outcome ~ .,
    data = train_data,
    perc.over = 100,
    perc.under = 200,
  )
```

2.1.5 Setting self-defined score to evaluate models

One important issue when evaluating different models is to choose the most suitable evaluation metric. Generally, a classification model can be evaluated through the total prediction accuracy or the accuracy under each class (calculated from the confusion matrix).

Here we propose a new metric based on the goal of this classification model, which is trying to judge whether banks will lend loans to individuals from the standpoint of banks'own credit risk management strategies. We use a weighted results from the confusion matrix to evaluate the effects of the models instead of simply using the accuracy rate of each class

The main income of the commercial banks comes from the difference between interests on loan and interests on deposits and an important risk towards this kind of income is non-performing loans. Basel Accord restricted the non-performing loans ration to be under 15% and in practice, banks usually restrict themselves more strictly. False negative, the FN, leads to non-performing loans and brings the highest credit risk to banks, so we treated it most carefully and weight FP negatively. In the same way, we treated true negative, the TN, carefully because it corresponds to avoiding the risk of signing up for non-performing loans.

Compared with the former two, FN and TP have less effects on the credit risk of banks so they should be weighted lower. The most common weighting method in this kind of projects is Analytic Hierarchy Process, ***AHP***. According to regular uses in practice, we use the simplified version of the ***AHP*** and got the score to evaluate the models:

$$score = -4FP - 2FN + 3TN + TP.$$

We choose the model with the biggest score to be our final result.

2.2 Parametric models

2.2.1 LDA and QDA models

Since we do not know the true boundary is linear or not, we decide to build LDA and QDA respectively and observed the result.

Based on EDA in the first part, we make 7 different sets of combination of variables and build 7 LDA models and 7 QDA models with 14 indexes respectively.

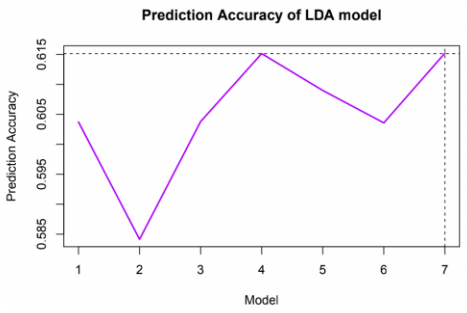
```
##    Model.index                                          Predictors Method
## 1       l.1(1)           term,installment,emp_length, annual_inc,dti    LDA
## 2       l.2(2)      loan_amnt,installment,emp_length, annual_inc,dti    LDA
## 3       l.3(3)             loan_amnt,term,emp_length, annual_inc,dti    LDA
## 4       l.4(4)             loan_amnt,term,installment,annual_inc,dti    LDA
## 5       l.5(5)            loan_amnt,term,installment,emp_length, dti    LDA
## 6       l.6(6)     loan_amnt,term,installment,emp_length, annual_inc    LDA
## 7    l.Full(7) loan_amnt,term,installment,emp_length, annual_inc,dti    LDA
## 8       q.1(8)           term,installment,emp_length, annual_inc,dti    QDA
## 9       q.2(9)       loan_amntinstallment,emp_length, annual_inc,dti    QDA
## 10     q.3(10)             loan_amnt,term,emp_length, annual_inc,dti    QDA
## 11     q.4(11)            loan_amnt,term,installment, annual_inc,dti    QDA
## 12     q.5(12)             loan_amnt,term,installment,emp_length,dti    QDA
## 13     q.6(13)     loan_amnt,term,installment,emp_length, annual_inc    QDA
## 14  q.Full(14) loan_amnt,term,installment,emp_length, annual_inc,dti    QDA
```

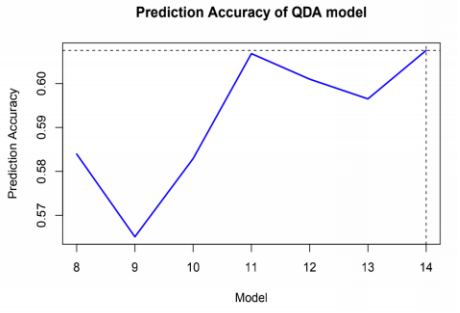
By comparing the weighted average scores we choose the final best LDA model with with 6 predictors (**loan_amnt,term,installment,emp_length, annual_inc,dti**), and the best QDA model with 6 predictors (**loan_amnt,term,installment,emp_length, annual_inc,dti**).

Their performance on the training set is shown below:

```
## lda.pred
## true_outcome 1 0
## 1 9561 3996
## 0 6459 7152
## [1] 0.6151723

## qda.pred
## true_outcome 1 0
## 1 6282 7275
## 0 3387 10224
## [1] 0.607553
```





We refit the LDA and QDA model using the chosen feature combinations on the whole training set again and evaluate these two models on the validation set later.

2.2.2 Logistic regression model

At first, we fit the model using all features and get an accuracy rate of about 63%. Considering that variables might be highly correlated / there may not be linear relationship between xi and y / interaction effect might exist. We also tried the following:

1) Dropping features that were reported as non-important from the full model: "installment", "loan_amnt", "home_ownership", "purpose".

2) Taking log transformation on several variables:

   log(int_rate) + log(emp_length) + log(annual_inc)

3) Adding interaction terms (as we suspect there will be interaction effect between the following pairs:

   int_rate * grade; amount * term

These three trials all gives a cross validation accuracy rate on the training set at about $0.63 - 0.64$. Since several variable transformation fails to improve accuracy further, it was suspected that doing much transformation on logistic model can not lead to further better result.

We verified by using caret package to automatically generate the final model with best feature combinations and do cross validation with k = 7 to estimate the accuracy on the validation set.

```
glm_mod = train(
  form = loan_outcome ~ .,
  data = balanced_data_trn,
  trControl = trainControl(method = "cv", number = 7),
  method = "glm",
  family = "binomial"
)
glm_mod$finalModel
```

```
Call:  NULL

Coefficients:
              (Intercept)                 loan_amnt                      term                  int_rate                installment
                3.071e+00                 7.396e-07                -2.188e-02                -1.571e-01                -4.474e-04
                    grade                emp_length         home_ownershipNONE        home_ownershipOTHER          home_ownershipOWN
                1.370e-01                -8.383e-03                 3.187e-01                -3.410e-01                -3.287e-01
       home_ownershipRENT                annual_inc                       dti         purposecredit_card  purposedebt_consolidation
                -1.437e-01                 6.012e-06                -2.227e-02                -1.079e-01                 1.749e-01
      purposeeducational    purposehome_improvement              purposehouse      purposemajor_purchase             purposemedical
                -1.930e-01                -3.330e-01                -1.959e-01                -5.578e-02                -2.113e-01
           purposemoving               purposeother    purposerenewable_energy      purposesmall_business           purposevacation
                -8.185e-02                -1.552e-01                -9.204e-02                -7.501e-01                -8.899e-02
          purposewedding
                 2.124e-01

Degrees of Freedom: 80375 Total (i.e. Null);  80350 Residual
Null Deviance:      111400
Residual Deviance: 100700       AIC: 100800
```

The model generated by caret has test accuracy at about 0.65. Therefore, we infer that logistic model on this data set has a prediction prediction accuracy at about 0.65. We interpreted this no further improvement as it's possible that there is a highly non-linear and complex relationship between the features and the response. Therefore, we will give up the linear part and try to fit some nonlinear models that are more suitable to data with complex relationship.

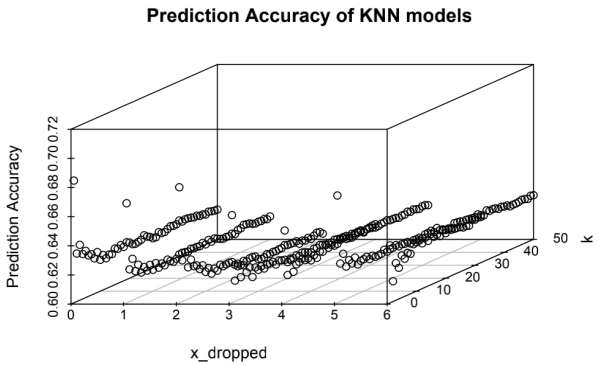
2.3 Nonparametric models

2.3.1 KNN classifier

First, we tried different feature combinations and different k to build our KNN models.

From the EDA conclusion in the first part, we make 7 different sets of combination of variables and make a grid of k from 1 50. Then we build a total number of $350(7 * 50)$ KNN models as follows.

```
##      K                                                        Predictors
## 1 1~50 loan_amnt,term,installment,emp_length, annual_inc,dti (Full set)
## 2 1~50                      term,installment,emp_length, annual_inc,dti
## 3 1~50               loan_amnt,installment,emp_length, annual_inc,dti
## 4 1~50                      loan_amnt,term,emp_length, annual_inc,dti
## 5 1~50                      loan_amnt,term,installment,annual_inc,dti
## 6 1~50                     loan_amnt,term,installment,emp_length, dti
## 7 1~50               loan_amnt,term,installment,emp_length, annual_inc
```



The best 3 models are:

(a) KNN_1: with K=1 and predictors:
"loan_amnt" + "term" + "installment"+ "emp_length" + "annual_inc" + "dti"
(b) KNN_2: with K=1 and predictors:
"term" + "installment"+ "emp_length" + "annual_inc" + "dti"
(c) KNN_3: with k=1 and predictors:
"loan_amnt" + "term" + "installment"+ "emp_length" + "dti"

Among the three models, the KNN_1 with $K = 1$ and predictors **"loan_amnt" + "term" + "installment"+ "emp_length" + "dti"** gives the best accuracy on train set, and the confusion matrix is set as below:

```
## true_outcome 0 1
## 0 11761 5120
## 1 5976 6674
```

We use this best setting (each k matches one of the combination of features) to fit the whole training set and collect their performance based on validation set.

2.3.2 SVM

For SVM, we use Radial Basis as a kernel function and tuned the parameters C and sigma on the training set. We use the caret package to hepl us tune the parameter and choose the best one. We first set up for cross validation with 10 fold cross validation, 5 times of repetition, and decided to use AUC to pick the best model.

After setting up for cross validation, we train and tune the SVM by setting our kernel be radial kernel and set 5 values of the cost function.

```
# Setup for cross validation
ctrl <- trainControl(method="repeatedcv",
                     repeats=5,
                     summaryFunction=twoClassSummary,
                     classProbs=TRUE)
svm.tune <- train(x=train_data,
                  y=train_data$loan_outcome
                  method = "svmRadial",
                  tuneLength = 5,
                  preProc = c("center","scale"),
                  metric="ROC",
                  trControl=ctrl)
```

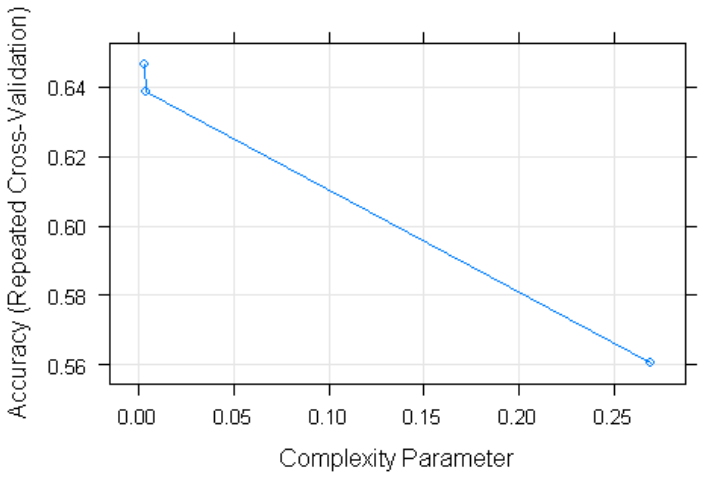
The best parameter for the model is sigma = 0.003909534, and C = 0.1. We use this values to fit the whole training set and collect its performance based on validation set later.

2.3.3 Decision Tree

We use decision tree to train a model with high interpretability. After certain trials, we found that whenever the predictor "grade" is included, it will definitely lead the whole tree no matter how we tune the parameters. Considering that in EDA, we have detected the high correlation between "grade" and other predictors, we decide to drop "grade" in building a single classification tree. We use rpart to do cross validation and tune the tree size automatically.

```
library(caret)
data <- balanced.data
set.seed(430)
ctrl <- trainControl(method = "repeatedcv",number = 10,repeats = 3)
model <- train(
              form = loan_outcome~.,
              data=balanced.train_data,
              method = "rpart",
              trControl = ctrl)
```

Accuracy on training set and the tree size chosen show the following relationship:



Output accuracy:

```
##    cp           Accuracy   Kappa
##  0.003234796  0.6476218  0.2952428
##  0.003695133  0.6403527  0.2807048
##  0.268960884  0.5611013  0.1222244
```

Tuning result shows that when maximum tree size = 4, the model has maximum accuracy in cross validation. But the accuracy rate on training set is not good enough. We want to see whether other tree models will yield a better result. And now we try bagging model as follows.

2.3.4 Bagging

After several trials we found that even if we have tune the parameters like tree size, the leading variable is still grade and this yielded similar bagged trees. Once we drop the "grade", the leading variables becomes "int_rate" and if we keep dropping this variable, the accuracy keep reducing and there is no predictive power in this model, as the result shows below.

```
bagging <- bagging(loan_outcome~ .,data = traindata, mfinal=20,control=rpart.control(maxdepth=4))
bagging$importance
bagging.pred <- predict.bagging(bagging,newdata=testdata)
bagging.pred$confusion
bagging.pred$error
```

Output accuracy:

```
## Predicted Class    0    1
##              0 7797 5125
##              1 5599 8271
## [1] 0.4002687
```

Based on bagging, random forests utilize an important key component of bootstrapping and bagging - only a subset of samples from the original data set are used to create each tree. On average, about two thirds of of each data set is sampled each time a bootstrap sample is taken. With one third of observations remaining, we utilize this subset for testing each newly created tree, creating out-of-bag (OOB) error.

The idea of selecting randomly a set of possible variables at each node can help us dealing with the dilemma of whether we should drop the **grade** and **int_rate**. Now we fit a random forest model.

2.3.5 Random forest

```
library(randomForest)
library(caret)
library(e1071)
set.seed(1234)
trControl <- trainControl(method = "oob",
    number = 10,
    search = "grid")
# Run the model
rf_default <- train(loan_outcome~.,
    data = traindata,
    method = "rf",
    metric = "Accuracy",
    trControl = trControl)
```

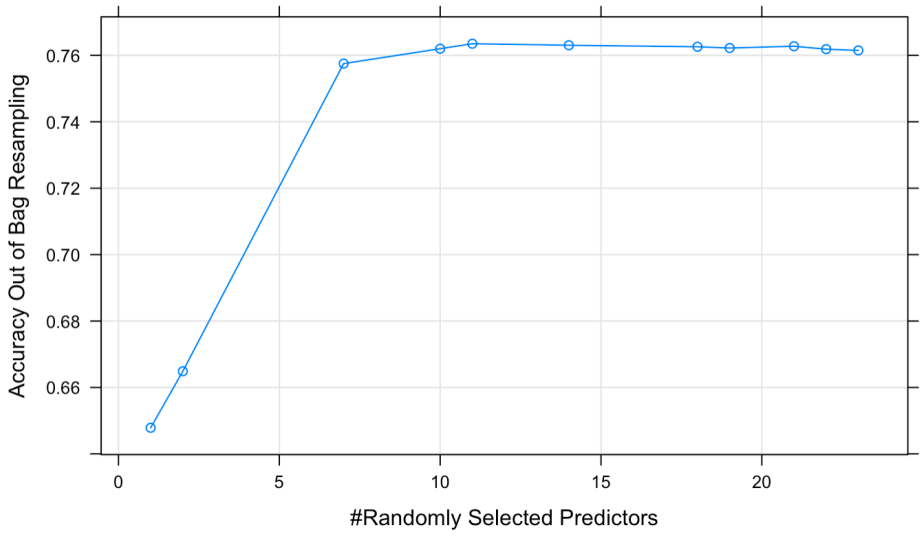
Output:

```
## Predicted Class    0    1
##               0 7797 5125
##               1 5599 8271
## [1] 0.4002687
```

```
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 72338, 72339, 72338, 72338, 72338, 72339, ...
## Resampling results across tuning parameters:
##  mtry  Accuracy   Kappa
##   2    0.6640412  0.3280824
##   13    0.7586094  0.5172189
##   25    0.7553124  0.5106248

##         OOB estimate of  error rate: 23.59%
## Confusion matrix:
##       0     1 class.error
## 0 30210  9978   0.2482831
## 1  8984 31204   0.2235493
```

After tuning, the final value used for the model was mtry = 13. Also, by randomly select predictors at every split, the random forest outperform single tree and bagging tree and give class error = 0.2482831 on the 0 class and class error = 0.2235493 on the 1 class.



3. Results and Discussion

3.1 Result

Since we have decided that linear model is not a good choice, we will just show the evaluation result of all the nonlinear models in the second part on the validation set using the best model selected under each type.

The best model under each type and confusion matrix of prediction on the validation data is shown below.

3.1.1 KNN classifier

We use the best KNN model with K=1 and predictors (loan_amnt,term,installment,emp_length, annual_inc, dti)to predict the labels on validation set.

Output:

Confusion matrix:

	True 0	True 1
Pred 0	4832	15072
Pred 1	4410	19343

Figure 4: confusion matrix of KNN

Accuracy: 0.5537

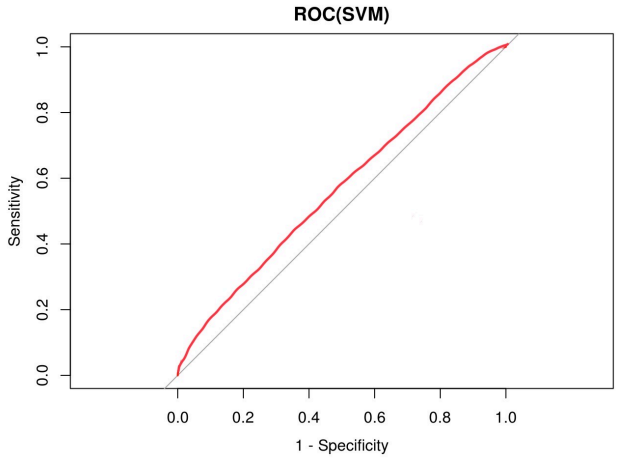
Self-defined score: -0.65403

3.1.2 SVM

The best parameter for the svm model is sigma = 0.003909534, and C = 0.1. The result of roc curve, confusion matrix and score are shown below.

Output:

ROC curve：



Confusion matrix:

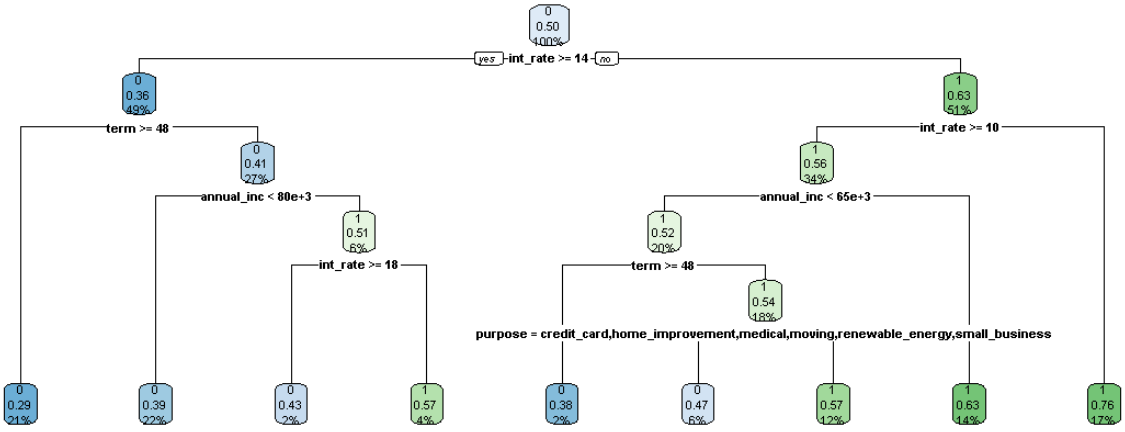
	True 0	True 1
Pred 0	8735	13086
Pred 1	4224	17611

Figure 5: confusion matrix of SVM

Self-defined score: -0.56297

3.1.3 Decision Tree

We plot out the best decision tree with max depth = 4. It shows that interest rate is the most important feature to predict a loan is a good loan or a bad one, followed by the term of loan. Other features like purpose, loan_amount are less important compared with the above two features.



The prediction result of confusion matrix and score of the best tree on validation set are shown below.

	True 0	True 1
Pred 0	5591	11922
Pred 1	3651	22849

Figure 6: confusion matrix of CART

Accuracy: 0.6433

Self-defined score: 0.195433

3.1.4 Bagging

The prediction result of confusion matrix and score of the best bagging trees on validation set are shown below.

	True 0	True 1
Pred 0	4121	9766
Pred 1	5121	21818

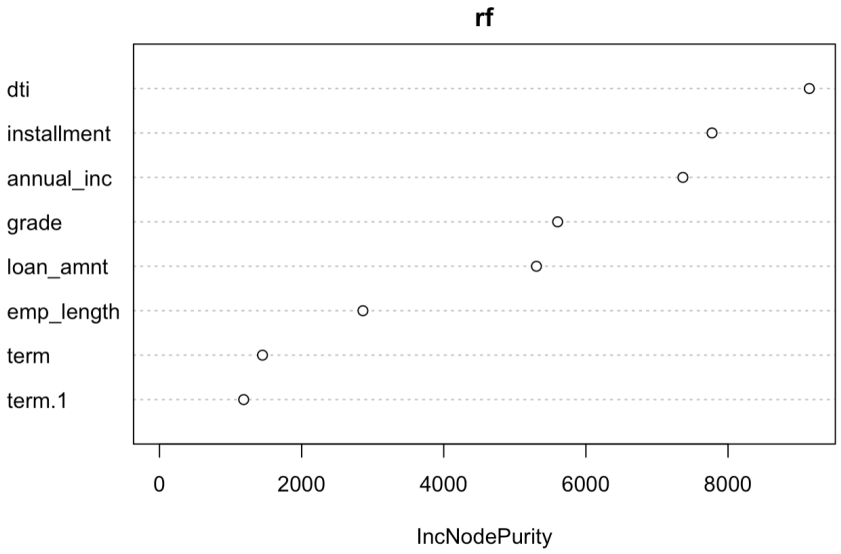
Figure 7: confusion matrix of bagging

Accuracy: 0.6353

Self-defined score: -0.80633

3.1.5 Random Forest

We plot out the importance of variables in the best random forest model.



The prediction result of confusion matrix and score of the best bagging trees on validation set are shown below.

	True 0	True 1
Pred 0	5394	11566
Pred 1	3848	22849

Figure 8: confusion matrix of random forest

Accuracy: 0.6469

Self-defined score: -0.80633

The final summary table of each model performance on the validation set is shown below.

Best Model	Accuracy	Score
Logistic Regression	0.65061	0.265038
LDA	0.687198	-0.11936
QDA	0.520466	0.518905
KNN	0.553749	-0.65403
SVM	0.6087	-0.56297
ClassificatioN Tree	0.643287	0.643287
Bagging	0.635355	0.635355
Random Forest	0.646929	0.646929

Figure 9: summary table of candidate models' performance on validation set

3.1.6 Estimation on the test data

Based on the self-defined metric, the random forest model outperform other models and was chosen as our final model. We evaluate the final model using the reserved test set and got the following confusion matrix.

	True 0	True 1
Pred 0	8735	13086
Pred 1	4224	17611

Figure 10: confusion matrix of the final model on test data

Self-defined score = 0.439455

Accuracy = 0.643491

False positive rate = 0.325851

False negative rate = 0.426296

3.2 Discussion

3.2.1 Overall comparisons between SVM, KNN and Decision tree

This project use several kinds of non parametric classification models, including KNN, Decision tree and SVM.The advantages and disadvantages of these models on this dataset are listed as follows.

Classification tree:

Advantages:

1. This data set contains both quality and quantity data, and classification tree can deal with these two types of data easily.
2. Classification tree is easy to understand and explain. Since this is a model to serve a bank, interpretability is a very important factor when evaluating different models. From the result of tree based methods, it's obvious that grade and interest rate of a customer are two most powerful features to describ the loan to be a good or bad loan. But this two variables are more like two summarized features generated from other sub-features. If we do not consider these two most powerful variables and try to find what may affect one's grade and interset rate, then debt-to-income ratio is the most important feature to consider.
3. Classification tree take less time than other methods.

Disadvantages:

1. For data with different data sets, the results of information gain in the decision tree tend to be those with more numerical characteristics, which is line with its performance on this dataset (as dti is numerical);
2. Over-fitting (can be avoided by applying random forest in this dataset).

SVM:

Advantages:

1. SVM can solve the problem in high dimension by applying the kernel functions, and thus deal with the interaction of nonlinear characteristics.
2. SVM can improve generalization ability of the model;
3. SVM usually have higher accuracy than other methods.

Disadvantages:

1. When the observed samples are large, the efficiency is still not high since it depends on the size of the training set, which is the case of this data set.
2. There is no general solution for nonlinear problems, and sometimes it is difficult to find a suitable kernel function.

KNN:

Advantages:

1. Classes don't have to be linearly separable.
2. Effective in deal with huge amount of data in this data set.

Disadvantages:

1. Large computation work load;
2. Sensitiveness to very unbalanced datasets, where most entities belong to one or a few classes, and infrequent classes are therefore often dominated in most neighborhoods. We alleviated this issue through balanced sampling of the more popular classes in the training stage.
3. Sensitiveness to noisy or irrelevant attributes, which can result in less meaningful distance numbers. We mitigate this issue by scaling and do feature selection.

3.2.2 Discussion on different tree based models

From the prediction result of of bagging models on validation set we see that, bagging does not have the advantage on this specific data set. Since the size of training set is large enough, a bootstrap strategy does not reduce variance as effectively as it performs on a smaller data set. The bagging trees are highly correlated and use the same most powerful predictor, interest rate, as a single classification tree.

Though bagging can not solve the problem of most powerful variable effectively, random forest, which is another tree based method, performs well under this dilemma. From the variable importance plot, we see that dti has the higher value of mean decrease accuracy , which means it is the most important variable in this random forest model. Also, **"installment"** and **"annual_income"** becomes important, which is not the same case as a single classification tree. This shows that by randomly choosing a subset each time, random forest model "decorrelates" each bag tree and the predictors is not necessarily the same as cart and bagging. So random forest is an ideal model to deal with a dataset that contains several powerful variables.

3.2.3 Discussion on the final model

Although compared with the accuracy on training data set, the accuracy on test set seemed to decrease, this result is still acceptable since the test set is not included in the pre-processing stage and is still highly imbalance. But the self-defined score shows that final model is good when focusing on the goal of judging whether banks will lend loans to individuals from the standpoint of banks'own credit risk management strategies.

4. Conclusion

In this project, we want to find a best loan-quality-prediction model that predict the binary outcome (good/bad) of a loan from multidimensional personal information input of the borrowers. We first did an **EDA** on the data and found it highly unbalanced. Therefore, we use **SMOTE** to balance the data and also standardize the data for some distance based model later.

We first tried several parametric models. Without knowing the true boundary, we tried models with linear classification boundary like **LDA** and **logistic regrssion**, and also one with quadratic boundary, **QDA**. After several trials on feature recombination, adding interaction terms and doing log transformation, none of these three model reach our expectation, and we suspected that the multidimensional personal information has a more complex structure than the assumption of linear/quadratic. This drove us to explore more on non parametric models.

We tried three most commonly used non-parametric models: **KNN, SVM and Tree**. After tuning each model through cross validation, we found that tree performed the best, and we tried to improve our tree model with other tree based methods. After trying **bagging and random forest**, we found that under the serious issue of powerful predictors, the effectiveness in reducing variance of bagging is reduced, whereas random forest provides a best strategy to deal with this dilemma. Therefore we choose the well-tuned random forest model as our final model and evaluate it on the untouched test set, reaching an accuracy of 0.643491 and a score of 0.439455.

Another highlight of this project is the **self-defined evaluation metric**. Referring to some literature, we proposed our own evaluation metric by calculating the weighted sum of the confusion matrix, which is supposed to meet the bank's requirement better than the general accuracy rate and finally give more powerful predictions.

5. References

1. Loan Data in Dummy Bank:https://www.kaggle.com/mrferozi/loan-data-for-dummy-bank

2. The main driver of banks' income is net interest margin http://www.cbirc.gov.cn

3. Basel III: A global regulatory framework for more resilient banks and banking systems https://www.bis.org/publ/bcbsl89.html

4. Practical decision making method:the principle of analytic hierarchy process https://bbs.pinggu.org/thread- 527563-l-l.html

5. An example of banksˆon-performing loan ratio in practice: http://www.cbirc.gov.cn/cn/view/pages/ItemDetail.html?docId=880477&itemId= 928&generaltype=0

6. Summary of the advantages and disadvantages of the commonly used classification algorithms: DT/ANN/KNN/SVM/GA/Bayes/Adaboosting/Rocchi
https://blog.csdn.net