A Classification Model for Consumer Loans

Author: Sanjiv Shah, PE

OU ID: 113180542

Course: DSA 5103: Intelligent Data Analytics

Instructor: Dr. Charles Nicholson

Semester: Fall 2016

Submission Date: December 31, 2016

Executive Summary:

Lending Club is one the largest online marketplace connecting borrowers and lenders/investors. To date, the Lending Club has facilitated loan over $22 billion since its inception. In 2015, the Lending Club originated over $ 7 billion in over 400,000 loans. This study attempts to analyze lending club's loan portfolio from 2007 to 2015 to develop a predictive model for non-performance (i.e. borrower failing to pay the loan in full) of a loan. In this study, several comprehensive binary classification models were developed and tested using a subset of the data to predict a whether a loan will be paid in full or not. The model developed is intended to be intermediate performance assessment tool rather than an underwriting tool since it relies heavily on the borrower's financial behavior in the months following issuance of the loan. Most consumer loan portfolios that rely on set underwriting criteria (i.e., borrower's ability to pay) rather than solely relying on collateral (e.g. title loans or payday loans) have low (< 10%) default rates. The biggest challenge of this study was to be able to carve out a subset of loans that were representative, had reasonable payment history and a sufficient number defaults to develop the model. Taking into consideration significant class imbalance for the predictive class, down-sampling and cost-sensitive training methods were also implanted. Due to limitation of computing resources some of the complex models like Support Vector Machine and Neural Networks failed to produce model outputs. However, the 2 methods that were capable of computing; Logistic Regression (LR) and Classification and Regression Trees (CART) performed extremely well. LR showed marginally better results and faster performance with down-sampling whereas CART did not show any measurable improvement in performance with down-sampling or cost sensitive analysis.

The Problem:

The data consists of loan information for loans generated by the Lending Club including, geographical information, employment information, and personal financial information such as salary, debt-to-income ratio, credit rating, etc., and loan information such as loan amount, interest rate, repayment history, loan performance, etc. All loan portfolios incur losses due to borrower's inability to pay off a loan. Lenders take this fact into account and it is reflected in either approval/denial of a loan and interest rate they offer to each borrower based on creditworthiness of a prospective borrower. Periodic/continuous monitoring of loan performance can provide early indication of loans that may be headed for default and allow the lender to take mitigating measures to prevent or minimize losses. By minimizing loan losses, the lender is able to offer better rates to its borrowers as wells as better rate of return to its investors in terms of higher return on their investments.

In order to address this problem, a predictive model is developed for loan defaults that can be used to periodically assess predictive performance of current loan portfolio. This will benefit the bank in terms of adjusting its lending and loan pricing criteria.

Data Understanding: (Raw Data)

```
dim(loan)
[1] 887379     74
```

Complete loan data for Lending Club from 2007 to 2015 is available totaling 887,379 records with 74 attributes. These are the loans that originated during this time period.

Observations:

1. Majority of the loans in the dataset are active and current loans. 'Current' is understood to be a loan that is active and is being paid back as per the loan terms without any late payments, missed payments, etc.
2. Final outcome of active loans is yet unknown. It would be not be prudent to assume that all these loans paid in full.
3. The data contains loan originations from 2007 and 2008. These two years are significant due to the fact they include the period prior to financial crisis and very lenient lending criteria in every segment of consumer finance. Also, these loans, if still active during the financial crisis, may have higher default rate than the loans in the following years due to macroeconomic conditions.
4. One of the most important criterion for loan approval and pricing is a person's credit score. This attribute was included in previous versions of the dataset. However, dataset being used for this study does not include the borrower's credit score. However, it does include a number of borrower's financial attributes that are used by different companies that provide credits scores. However, some of those attributes are sparsely populated. Since the model envisioned is a classification model (Default: Yes/No) and Lending Club's own credit rating and sub-rating are provided, it can serve as a proxy for credit score.
5. The interest rate environment changed significantly at the end of 2008. As shown in figure 1, the prime rate dropped from 8.0 % in 2007 to 3.25% at the beginning of 2009. This is very important fact because prime rate is used as a benchmark for pricing (i.e. setting interest rate) a loan. For example, a loan carrying an interest rate of 10% (prime +2%) in 2007 may represent a borrower with high credit rating. A loan carrying same interest rate of 10% (prime + 5.75%) most likely represents a borrower with poor credit rating.

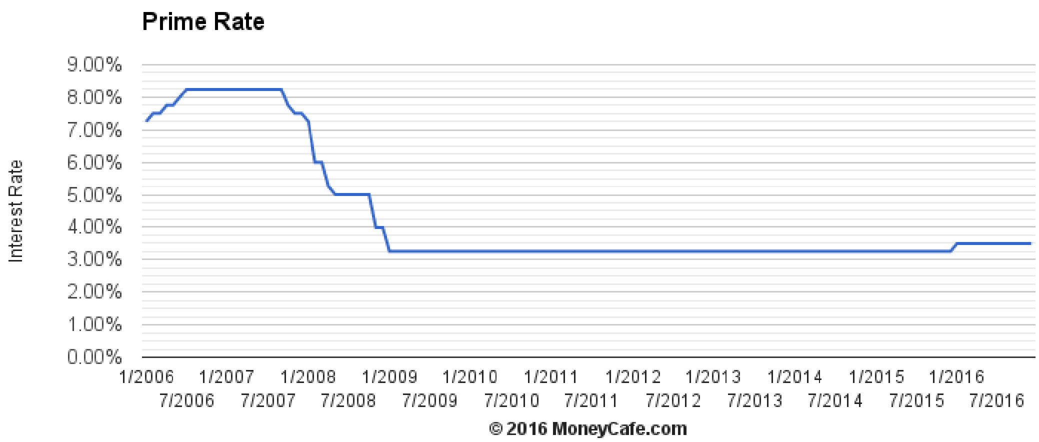


Figure 1: Prime Rate 2006-2016

Despite the challenges the data presents, there are sufficient data to develop a classification model to predict loan defaults based on its payment history.

Data Visualizations:

1. Loan Amounts

```
> Desc(loan$loan_amnt, main = "Loan Amount Distribution", plotit = TRUE)
----------------------------------------------------------------------------
Loan Amount Distribution

     length         n       NAs    unique        0s       mean     meanSE
      9e+05     9e+05         0     1e+03         0   1.48e+04   8.95e+00

        .05       .10       .25    median       .75        .90        .95
   3.60e+03  5.00e+03  8.00e+03  1.30e+04  2.00e+04   2.80e+04   3.20e+04

      range        sd     vcoef       mad       IQR       skew       kurt
   3.45e+04  8.44e+03  5.72e-01  8.60e+03  1.20e+04   6.82e-01  -2.57e-01

lowest : 5.00e+02 (1e+01), 5.50e+02, 6.00e+02 (6e+00), 7.00e+02 (3e+00), 7.25e+02
highest: 3.49e+04 (1e+01), 3.49e+04 (9e+00), 3.50e+04 (2e+01), 3.50e+04 (3e+01), 3.50e+04 (4e+04)
```

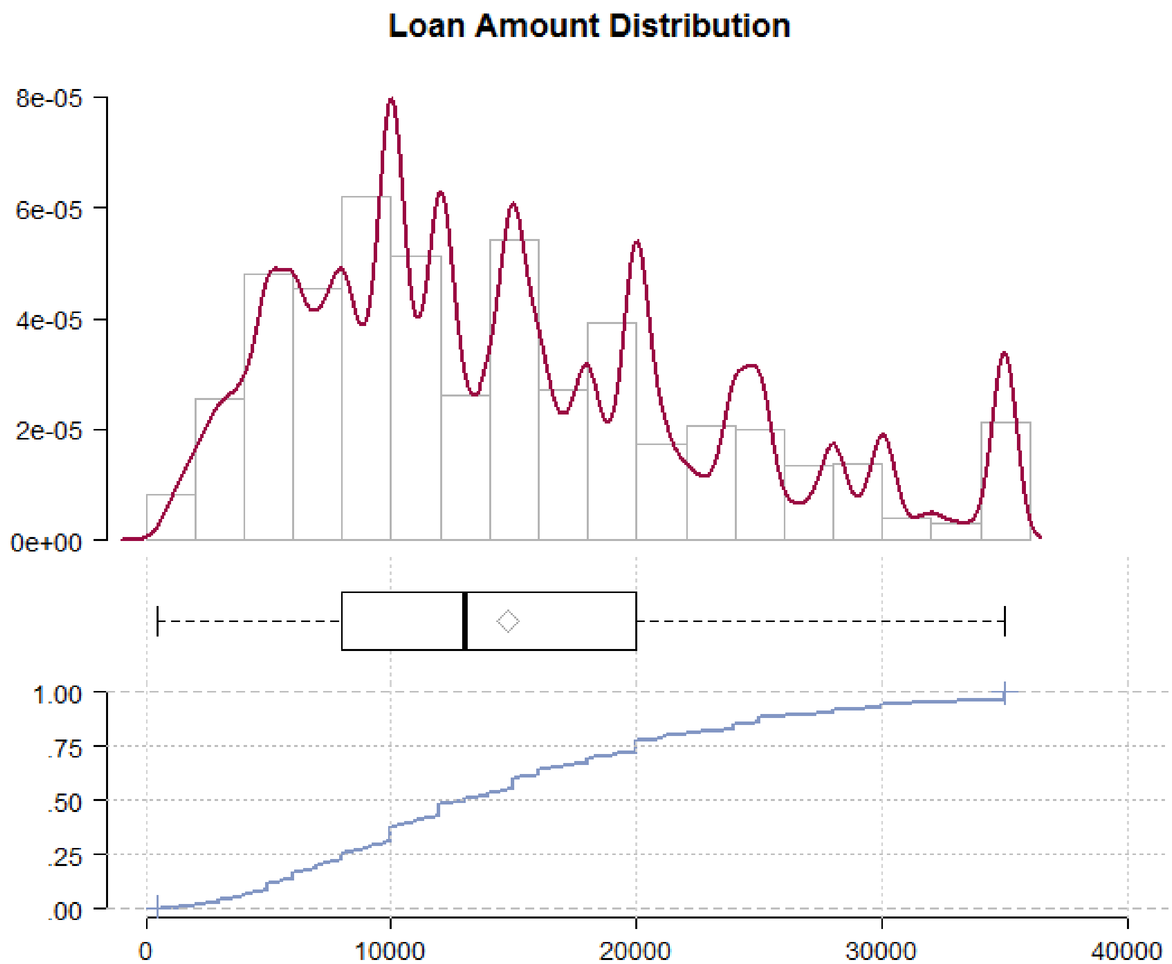


Figure 2: Loan Amount Distribution

2. Interest Rate Distribution

```
> Desc(loan$int_rate, main = "Interest Rate Distribution", plotit = TRUE)
-------------------------------------------------------------------------------------------------
Interest Rate Distribution
     length          n        NAs     unique         0s       mean     meanSE
      9e+05      9e+05          0      5e+02          0   1.32e+01   4.65e-03

        .05        .10        .25     median        .75        .90        .95
   6.62e+00   7.69e+00   9.99e+00   1.30e+01   1.62e+01   1.90e+01   2.10e+01

      range         sd      vcoef        mad        IQR       skew       kurt
   2.37e+01   4.38e+00   3.31e-01   4.45e+00   6.21e+00   4.29e-01  -1.55e-01

lowest : 5.32e+00 (1e+04), 5.42e+00 (6e+02), 5.79e+00 (4e+02), 5.93e+00 (2e+03), 5.99e+00 (3e+02)
highest: 2.75e+01 (7e+00), 2.79e+01 (2e+02), 2.80e+01 (5e+00), 2.85e+01 (1e+02), 2.90e+01 (1e+02)
```

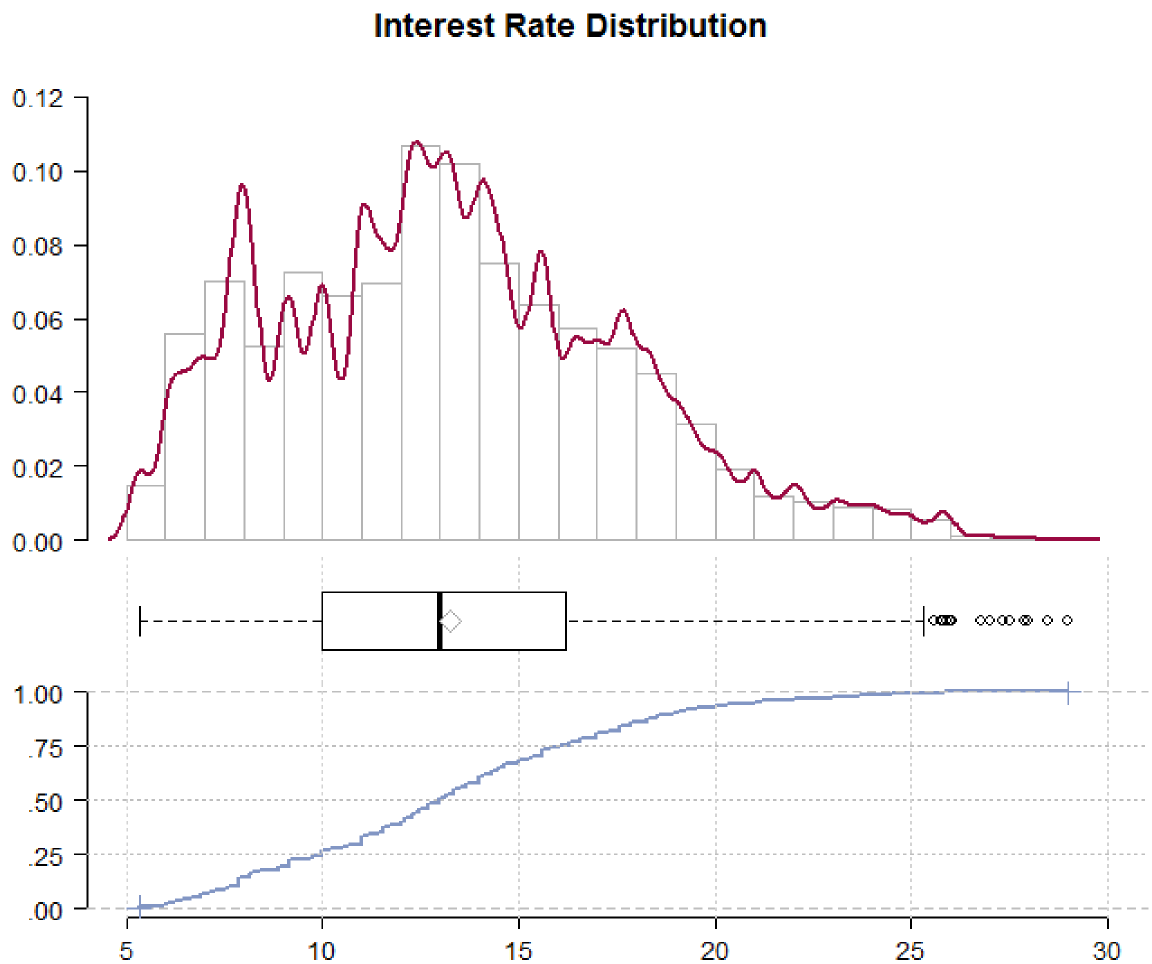


Figure 3: Interest Rate Distribution of all loans (2007-2015)

3. Status of Loans

```
> Desc(loan$loan_status, plotit = TRUE)
------------------------------------------------------------------------------------------------
loan$loan_status (character)

  length      n    NAs unique levels  dupes
   9e+05  9e+05      0  1e+01  1e+01      y

                                                  level  freq   perc  cumfreq  cumperc
1                                               Current  6e+05  67.8%    6e+05    67.8%
2                                            Fully Paid  2e+05  23.4%    8e+05    91.2%
3                                           Charged Off  5e+04   5.1%    9e+05    96.3%
4                                     Late (31-120 days)  1e+04   1.3%    9e+05    97.6%
5                                                Issued  8e+03   1.0%    9e+05    98.6%
6                                       In Grace Period  6e+03   0.7%    9e+05    99.3%
7                                     Late (16-30 days)  2e+03   0.3%    9e+05    99.6%
8    Does not meet the credit policy. Status:Fully Paid  2e+03   0.2%    9e+05    99.8%
9                                               Default  1e+03   0.1%    9e+05    99.9%
10  Does not meet the credit policy. Status:Charged Off  8e+02   0.1%    9e+05   100.0%
```

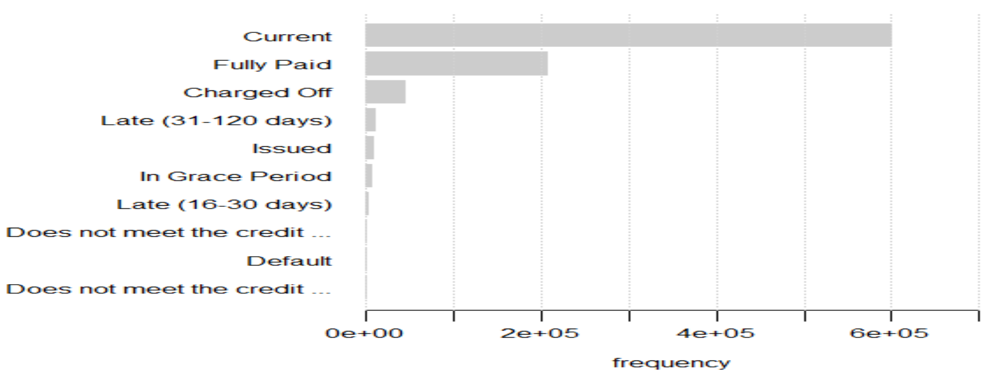


Figure 4: Loan Status (Classification Criterion)

4. Loans by Credit Grading

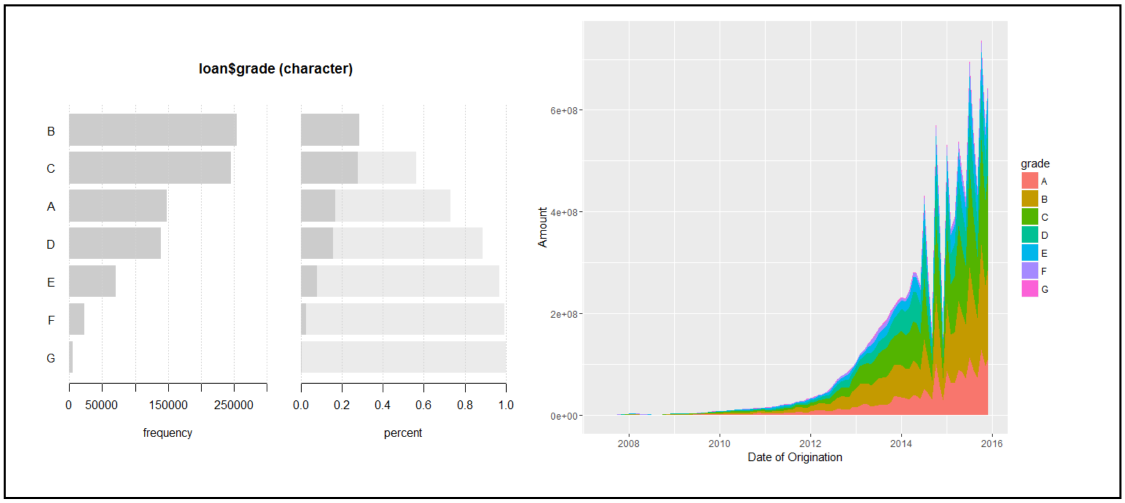


Figure 5: Loans by Credit Rating

5. Loans by State

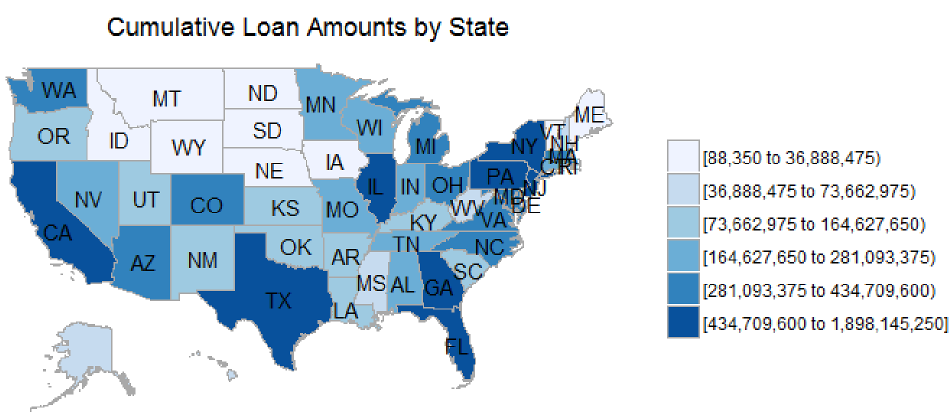


Figure 6: Loans by State (cumulative 2007 – 2015, total Loans: 13,093,511,950)

6. Loans by Purpose

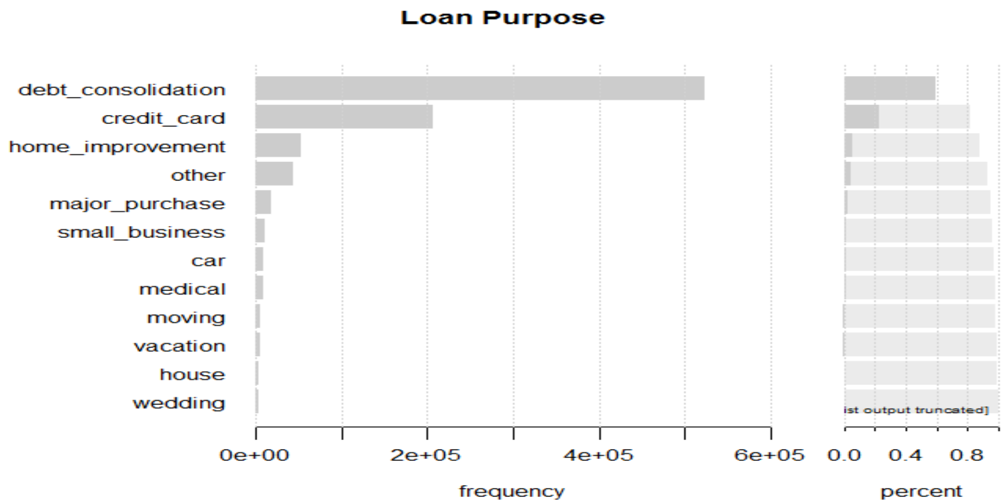


Figure 7: Loans by Purpose for borrowing

7. Loans by Year

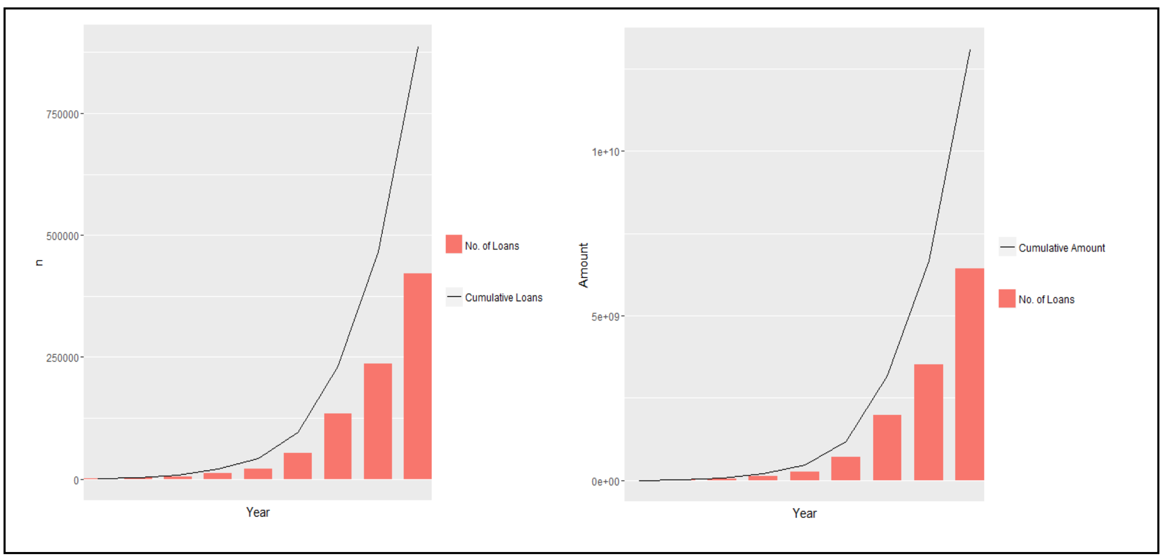


Figure 8: No. of Loans and Loan Amounts by Year

Data Preparation:

1. Remove all loans issued in 2007 and 2008 as they represent significantly different macroeconomic environment than loans that were originated since then.

2. Remove loans with "status" indicating "current" or "issued" (they can't be classified yet)

Remaining records:

```
> dim(loan_non_active)
```

```
[1] 274397     74
> unique(loan_non_active$loan_status)
[1] "Fully Paid"
[2] "Charged Off"
[3] "Default"
[4] "Late (31-120 days)"
[5] "In Grace Period"
[6] "Late (16-30 days)"
[7] "Does not meet the credit policy. Status:Fully Paid"
[8] "Does not meet the credit policy. Status:Charged Off"
```

Note: Remaining records do include some recent loans that have experienced late payment or missing payment or non-payment, etc. The final dataset that will be used will have all post-2008 loans that either have been paid off or charged-off (default) or have experienced an adverse event to have its status changed from “current” to something else. For classification purposes these loans will be considered to be in ‘default’

Predictor Set:

1. The attribute *loan_status* is the outcome.
2. Some of the attributes have > 99% missing values. They are not included in the predictor set.
3. Of the remaining attributes, only 4 had any missing data. None of the 4 had more than 10 % missing values. Considering the size of the dataset, these data were dropped from the modeling dataset.
4. Some of the attributes only contained identify information and they were excluded from the predictor set.
5. Since the dates in the future will be different, all date fields were dropped from the modeling dataset to make the model usable for future loans (Initial modeling runs included the origination date field. However, the modeling results were not impacted due to its exclusion. This is a clear indication that macro-economical conditions very stable during the entire period)

Feature Engineering

Initially, models were created using the entire predictor set (19 predictors). Since these models were able to predict excellent results, no further feature engineering efforts were undertaken.

Data Imbalance:

Once final dataset was created it was checked for prediction class (loan_status) imbalance. The ratio of good vs bad loans was approximately 3:1. However, in the preliminary runs, there seem to no significant effect on the predicted outcomes (discussed late in the results section). Despite that, it was determined to address the imbalance during modeling phase.

The dataset was divided in training (80%) and testing (20%). Even though it was random split, training dataset had the same proportion (3:1) as the undivided dataset.

Modeling:

Initial modeling plan included developing several models using various techniques such as logistic regression (LR), classification and regression trees (CART), support vector machines (SVM) and neural networks (NN) with entire dataset. However, due to limitations of the computing hardware, NN (insufficient memory) and SVM (3+ hour running time) were dropped from the analysis. Additionally, LR and CART were able to generate excellent predictive models (discussed in the results section).

1. Logistic Regression (with entire training set)
2. Logistic Regression (with down-sampling)
3. CART (with entire training set)
4. CART (with down-sampling)
5. CART (cost-sensitive using cost matrix)

Results for each model are shown in Appendix A.

Results:

Table 1 shows results of various models. Some salient observations:

Logistic Regression:

1. Simplest model (LR) produced best results.
2. Down-sampling ***marginally*** improved the results. However, since down-sampling reduced the size of training dataset, it took much less time to generate and test the model.

Metric	**Logistic Regression (data as is)**	**CART (data as is)**	**Logistic Regression (data: downsample)**	**CART (data: downsample**	**CART (cost sensitive/class weighted)**
ROC	0.9975	0.9932	0.9976	0.9931	0.9941
CM-Accuracy	0.9885	0.9921	0.9922	0.9917	0.9811
Kappa Value	0.9702	0.9796	0.9798	0.9786	0.9521
Sensitivity	0.9640	0.9799	0.9808	0.9836	0.9882
Specificity	0.9973	0.9965	0.9963	0.9946	0.9786

Table 1: Summary of Predictive Modeling Metrics

Classification and Regression Trees:

1. CART model with entire training dataset generated excellent results.
2. Down-sampling training dataset did not measurably improve the results.
3. Cost-sensitive CART model (using entire training dataset) did not measurably improve the results. However, it took the longest time to generate this model.

Even though down-sampling did not show measurable improvement in performance metrics for the models, it must be noted that both LR and CART were able generate excellent models with complete dataset and did not present significant opportunity or room for improvement.

Conclusions:

In consumer finance, loan-loss prevention is a key issue. Since 2008 financial crisis, a lot of investors experienced significant losses in real estate and real-estate related debt. For its capital needs, the Lending Club solely relies on investors looking for above-market return. In low interest rate environment since the financial crisis, it is imperative for the Lending Club to keep loan-losses at a minimum so that it can provide decent return to its investors. Under this context the use of a simple but effective predictive model to monitor its loan portfolio is a valuable tool to capture early signs of loan quality degradation. If loans headed for default are identified early on, the Lending Club can deploy some proven strategies to prevent it from a default.

In this study, a five different classification models were developed and tested. These models were compared using five metrics – Receiver Operating Characteristic (ROC), Confusion Matrix (CM) Accuracy, Kappa value, Sensitivity and specificity. In the final evaluation, LR model with down-sampling produced the best results for all metrics and was fastest to compute.

References:

1. Kuhn M, Johnson K, Applied Predictive Modeling (Springer, 2013)
2. Moro S, Cortez P, Rita Paulo, A data-driven approach to predict the success of Bank Telemarketing, Decision Support Systems, 62(2014).

Appendix A

Classification Models

```
> str(loan_closed)
'data.frame':	201187 obs. of  24 variables:
 $ id                 : int  10139658 10179520 10149577 10127816 10149566 10119590 10148818 10149
488 10129506 10079457 ...
 $ member_id          : int  11991209 12031088 12001118 11979581 12001108 11971211 12000415 12001
033 11981122 11931082 ...
 $ loan_amnt          : int  12000 3000 28000 24000 8000 11500 15000 4800 20800 10000 ...
 $ int_rate           : num  13.53 12.85 7.62 13.53 10.99 ...
 $ installment        : num  407 101 873 815 262 ...
 $ grade              : Factor w/ 7 levels "A","B","C","D",..: 2 2 1 2 2 5 3 2 2 2 ...
 $ sub_grade          : Factor w/ 35 levels "A1","A2","A3",..: 10 9 3 10 7 24 12 7 10 8 ...
 $ emp_length         : int  10 10 5 10 2 4 10 2 10 10 ...
 $ home_ownership     : Factor w/ 5 levels "ANY","NONE","OTHER",..: 5 5 4 4 4 5 5 4 5 4 ...
 $ annual_inc         : num  40000 25000 325000 100000 33000 ...
 $ verification_status: Factor w/ 2 levels "Not Verified",..: 2 2 2 2 1 2 1 2 2 1 ...
 $ issue_d            : Factor w/ 41 levels "Apr-2013","Apr-2014",..: 9 9 9 9 9 9 9 9 9 9 ...
 $ loan_status        : Factor w/ 2 levels "Default","Fully_Paid": 2 2 2 2 1 1 2 2 2 2 ...
 $ purpose            : Factor w/ 13 levels "car","credit_card",..: 3 3 3 2 3 3 3 4 3 3 ...
 $ dti                : num  16.9 24.7 18.6 22.2 15.8 ...
 $ inq_last_6mths     : int  0 0 1 0 1 0 2 2 2 1 ...
 $ open_acc           : int  7 5 15 14 9 12 16 3 29 10 ...
 $ revol_bal          : int  5572 2875 29581 21617 7203 9996 5749 4136 23473 12409 ...
 $ revol_util         : num  68.8 54.2 54.6 76.7 34.6 70.9 22.3 16.1 54.5 65 ...
 $ out_prncp          : num  0 0 0 0 4145 ...
 $ total_pymnt        : num  13360 3182 29151 28652 4990 ...
 $ application_type   : Factor w/ 2 levels "INDIVIDUAL","JOINT": 1 1 1 1 1 1 1 1 1 1 ...
 $ tot_cur_bal        : num  13605 19530 799592 199834 15949 ...
 $ total_rev_hi_lim   : num  8100 5300 54200 28200 20800 14100 25800 25700 43100 19100 ...
> table(loan_closed$loan_status)

   Default Fully_Paid 
     53122     148065 
> predictors
 [1] "loan_amnt"           "int_rate"            "installment"         "sub_grade"          
 [5] "emp_length"          "home_ownership"      "annual_inc"          "verification_status"
 [9] "purpose"             "dti"                 "inq_last_6mths"      "open_acc"           
[13] "revol_bal"           "revol_util"          "out_prncp"           "total_pymnt"        
[17] "application_type"    "tot_cur_bal"         "total_rev_hi_lim"   

> dim(loan_train)
[1] 160950     24

> dim(loan_eval)
[1] 40237    24
```

Closed_loans: Training set: Logistic Regression (training)

```
> lrfit
Generalized Linear Model
```

```
160950 samples
    18 predictor
     2 classes: 'Default', 'Fully_Paid'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 144855, 144855, 144855, 144855, 144855, 144855, ...
Resampling results:

  ROC        Sens       Spec       Accuracy   Kappa
  0.9971058  0.9658808  0.9974082  0.9890836  0.9716661

> lrROC

Call:
roc.default(response = eval_results$loan_status, predictor = eval_results$LogReg,     levels = re
v(levels(eval_results$loan_status)))

Data: eval_results$LogReg in 29613 controls (eval_results$loan_status Fully_Paid) < 10624 cases (
eval_results$loan_status Default).
Area under the curve: 0.9975

> lrEvalCM
Confusion Matrix and Statistics

            Reference
Prediction   Default Fully_Paid
  Default      10242         79
  Fully_Paid     382      29534

               Accuracy : 0.9885
                 95% CI : (0.9875, 0.9896)
    No Information Rate : 0.736
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9702
 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.9640
            Specificity : 0.9973
         Pos Pred Value : 0.9923
         Neg Pred Value : 0.9872
             Prevalence : 0.2640
         Detection Rate : 0.2545
   Detection Prevalence : 0.2565
      Balanced Accuracy : 0.9807

       'Positive' Class : Default
```

Closed_loans: Training set: CART (training)

```
> rpartFit
CART

160950 samples
    19 predictor
     2 classes: 'Default', 'Fully_Paid'
```

```
No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 144855, 144855, 144855, 144855, 144855, 144855, ...
Resampling results across tuning parameters:

  cp            ROC        Sens       Spec       Accuracy   Kappa
  0.0001882442  0.9926338  0.9816226  0.9964374  0.9925256  0.9807161
  0.0002117747  0.9921678  0.9802578  0.9963952  0.9921342  0.9796983
  0.0002353052  0.9920572  0.9799519  0.9963952  0.9920534  0.9794876
  0.0002588357  0.9919794  0.9794578  0.9964289  0.9919478  0.9792109
  0.0002823662  0.9918452  0.9787283  0.9964543  0.9917738  0.9787562
  0.0003058967  0.9917677  0.9783519  0.9963867  0.9916247  0.9783698
  0.0003137403  0.9917466  0.9781871  0.9963867  0.9915812  0.9782564
  0.0003294273  0.9916112  0.9776930  0.9963614  0.9914321  0.9778683
  0.0004353146  0.9911189  0.9767754  0.9962770  0.9911277  0.9770771
  0.0004941409  0.9907703  0.9763283  0.9959900  0.9907984  0.9762294
  0.0005529672  0.9903956  0.9757636  0.9957029  0.9904380  0.9752992
  0.0006117935  0.9903951  0.9756695  0.9956776  0.9903945  0.9751864
  0.0007647419  0.9896251  0.9742105  0.9955341  0.9899037  0.9739076
  0.0008235682  0.9893182  0.9734810  0.9954159  0.9896241  0.9731814
  0.0009059250  0.9889720  0.9725398  0.9953905  0.9893569  0.9724831
  0.0010118123  0.9874215  0.9695748  0.9955256  0.9886735  0.9706840
  0.0010588734  0.9867974  0.9679747  0.9955594  0.9882759  0.9696378
  0.0011529954  0.9854265  0.9646570  0.9956354  0.9874557  0.9674806
  0.0015883100  0.9841573  0.9628923  0.9947996  0.9863747  0.9646857
  0.0022412819  0.9827541  0.9588452  0.9943353  0.9849643  0.9609917
  0.0024001129  0.9803768  0.9544687  0.9943015  0.9837838  0.9578711
  0.0028001318  0.9782478  0.9501392  0.9943353  0.9826654  0.9548975
  0.0028824886  0.9779166  0.9473626  0.9944113  0.9819882  0.9530899
  0.0065885453  0.9757702  0.9335261  0.9946054  0.9784778  0.9436865
  0.0110828745  0.9646367  0.9072424  0.9952892  0.9720410  0.9261436
  0.0173184620  0.9604669  0.8928895  0.9955256  0.9684250  0.9161862
  0.0297425761  0.9441135  0.8523696  0.9957029  0.9578565  0.8863802
  0.0819450327  0.8618567  0.7036778  0.9968764  0.9194595  0.7699201
  0.1362181750  0.7597427  0.5130594  0.9986661  0.8704442  0.5986143
  0.3762059391  0.5551500  0.1102999  1.0000000  0.7650761  0.1383425

ROC was used to select the optimal model using  the largest value.
The final value used for the model was cp = 0.0001882442.

> rpartROC

Call:
roc.default(response = eval_results$loan_status, predictor = eval_results$RPART,     levels = rev
(levels(eval_results$loan_status)))

Data: eval_results$RPART in 29613 controls (eval_results$loan_status Fully_Paid) < 10624 cases (e
val_results$loan_status Default).
Area under the curve: 0.9932

> rpartEvalCM
Confusion Matrix and Statistics

            Reference
Prediction   Default Fully_Paid
  Default      10410        104
```

```
  Fully_Paid     214      29509

             Accuracy : 0.9921
               95% CI : (0.9912, 0.9929)
  No Information Rate : 0.736
  P-Value [Acc > NIR] : < 2.2e-16

                Kappa : 0.9796
Mcnemar's Test P-Value : 9.813e-10

          Sensitivity : 0.9799
          Specificity : 0.9965
       Pos Pred Value : 0.9901
       Neg Pred Value : 0.9928
           Prevalence : 0.2640
       Detection Rate : 0.2587
 Detection Prevalence : 0.2613
    Balanced Accuracy : 0.9882

     'Positive' Class : Default
```

Class Imbalance:

In the Closed loan data

```
> table(loan_closed$loan_status)

   Default Fully_Paid
     53122     148065
```

In the Training Set

```
> table(loan_train$loan_status)
   Default Fully_Paid
     42498     118452
```

Downsample Training data:

```
> dim(ds_loan_train)
[1] 84996    20

> table(ds_loan_train$loan_status)

   Default Fully_Paid
     42498      42498
```

Closed_loans: Down-sampled set: Logistic Regression (training)

```
> ds_lrfit
Generalized Linear Model

84996 samples
   18 predictor
    2 classes: 'Default', 'Fully_Paid'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 76497, 76496, 76496, 76497, 76496, 76497, ...
Resampling results:

  ROC        Sens       Spec        Accuracy    Kappa
```

```
  0.9972635  0.980799  0.9959292  0.9883641  0.9767282


> ds_lrROC

Call:
roc.default(response = ds_eval_results$loan_status, predictor = ds_eval_results$LogReg,     level
s = rev(levels(ds_eval_results$loan_status)))

Data: ds_eval_results$LogReg in 29613 controls (ds_eval_results$loan_status Fully_Paid) < 10624 c
ases (ds_eval_results$loan_status Default).
Area under the curve: 0.9976

> ds_lrEvalCM
Confusion Matrix and Statistics

            Reference
Prediction   Default Fully_Paid
  Default      10420        111
  Fully_Paid     204      29502

               Accuracy : 0.9922
                 95% CI : (0.9913, 0.993)
    No Information Rate : 0.736
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9798
 Mcnemar's Test P-Value : 2.176e-07

            Sensitivity : 0.9808
            Specificity : 0.9963
         Pos Pred Value : 0.9895
         Neg Pred Value : 0.9931
             Prevalence : 0.2640
         Detection Rate : 0.2590
   Detection Prevalence : 0.2617
      Balanced Accuracy : 0.9885

       'Positive' Class : Default
```

Closed_loans: Down-sampled set: CART

```
> ds_rpartFit
CART

84996 samples
   19 predictor
    2 classes: 'Default', 'Fully_Paid'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 76497, 76496, 76496, 76497, 76496, 76497, ...
Resampling results across tuning parameters:

  cp            ROC        Sens       Spec       Accuracy   Kappa
  0.0001647136  0.9931658  0.9846345  0.9937879  0.9892112  0.9784224
  0.0002117747  0.9926645  0.9837638  0.9935997  0.9886818  0.9773636
```

```
  0.0002470705  0.9924045  0.9833638  0.9935055  0.9884347  0.9768694
  0.0002823662  0.9919840  0.9825403  0.9933408  0.9879406  0.9758812
  0.0003294273  0.9914440  0.9814343  0.9933408  0.9873876  0.9747752
  0.0003647230  0.9910373  0.9807284  0.9933879  0.9870582  0.9741163
  0.0004235493  0.9907150  0.9799284  0.9932938  0.9866111  0.9732222
  0.0005647325  0.9901493  0.9779989  0.9931055  0.9855522  0.9711044
  0.0005882630  0.9899367  0.9774342  0.9931996  0.9853169  0.9706338
  0.0006706198  0.9897180  0.9768695  0.9931996  0.9850345  0.9700691
  0.0007882724  0.9888518  0.9755282  0.9910584  0.9832933  0.9665866
  0.0008000376  0.9885393  0.9748223  0.9911055  0.9829639  0.9659278
  0.0008118029  0.9884124  0.9743045  0.9911290  0.9827168  0.9654336
  0.0008235682  0.9884124  0.9743045  0.9911290  0.9827168  0.9654336
  0.0009176902  0.9877513  0.9734810  0.9900230  0.9817521  0.9635041
  0.0009765165  0.9874204  0.9728692  0.9898113  0.9813403  0.9626805
  0.0021118641  0.9833442  0.9646574  0.9894584  0.9770577  0.9541155
  0.0029295496  0.9807487  0.9604686  0.9888935  0.9746811  0.9493621
  0.0031766201  0.9787169  0.9563272  0.9896229  0.9729752  0.9459503
  0.0037178220  0.9774676  0.9461384  0.9899759  0.9680573  0.9361145
  0.0045649207  0.9702392  0.9381851  0.9818343  0.9600097  0.9200194
  0.0046119817  0.9694985  0.9366321  0.9818578  0.9592450  0.9184899
  0.0050120006  0.9673575  0.9310084  0.9812225  0.9561155  0.9122310
  0.0061767613  0.9644135  0.9241140  0.9813402  0.9527271  0.9054542
  0.0098828180  0.9558746  0.9080427  0.9820696  0.9450562  0.8901124
  0.0162478234  0.9542056  0.9091487  0.9671755  0.9381618  0.8763236
  0.0483552167  0.9244374  0.8686055  0.9364441  0.9025248  0.8050496
  0.0738387689  0.8486618  0.8351921  0.8327460  0.8339690  0.6679381
  0.0969222081  0.7864063  0.7962963  0.7654480  0.7808721  0.5617442
  0.5145183303  0.6273029  0.6730763  0.5815294  0.6272853  0.2546058

ROC was used to select the optimal model using  the largest value.
The final value used for the model was cp = 0.0001647136.
```

```
> ds_rpartROC

Call:
roc.default(response = ds_eval_results$loan_status, predictor = ds_eval_results$RPART,     levels
= rev(levels(ds_eval_results$loan_status)))

Data: ds_eval_results$RPART in 29613 controls (ds_eval_results$loan_status Fully_Paid) < 10624 ca
ses (ds_eval_results$loan_status Default).
Area under the curve: 0.9931
```

```
> ds_rpartEvalCM
Confusion Matrix and Statistics

            Reference
Prediction   Default Fully_Paid
  Default      10450        160
  Fully_Paid     174      29453

               Accuracy : 0.9917
                 95% CI : (0.9908, 0.9926)
    No Information Rate : 0.736
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9786
```

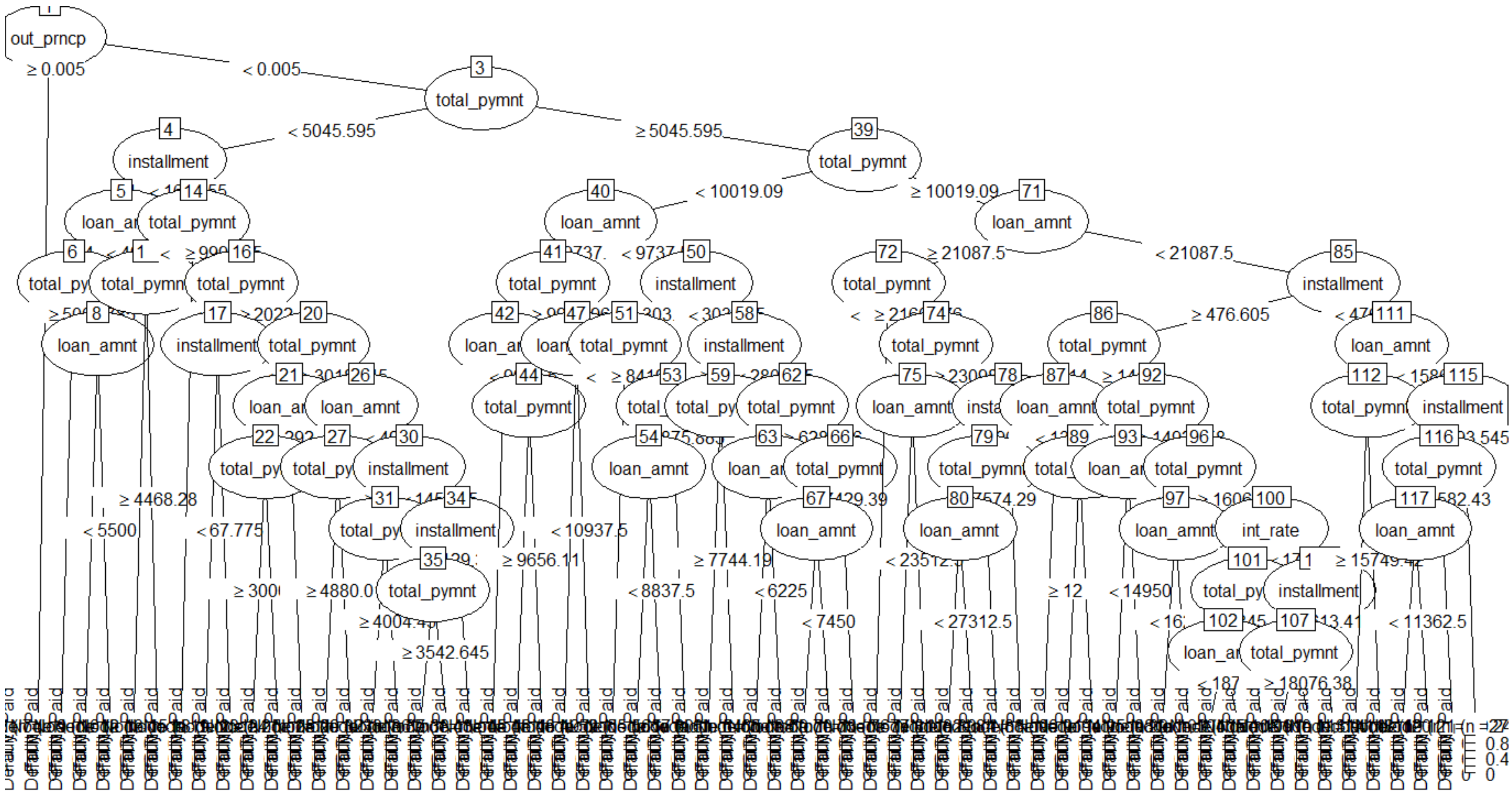
```
 Mcnemar's Test P-Value : 0.4769

            Sensitivity : 0.9836
            Specificity : 0.9946
         Pos Pred Value : 0.9849
         Neg Pred Value : 0.9941
             Prevalence : 0.2640
         Detection Rate : 0.2597
   Detection Prevalence : 0.2637
      Balanced Accuracy : 0.9891

       'Positive' Class : Default
```

Cost-Sensitive Training: CART

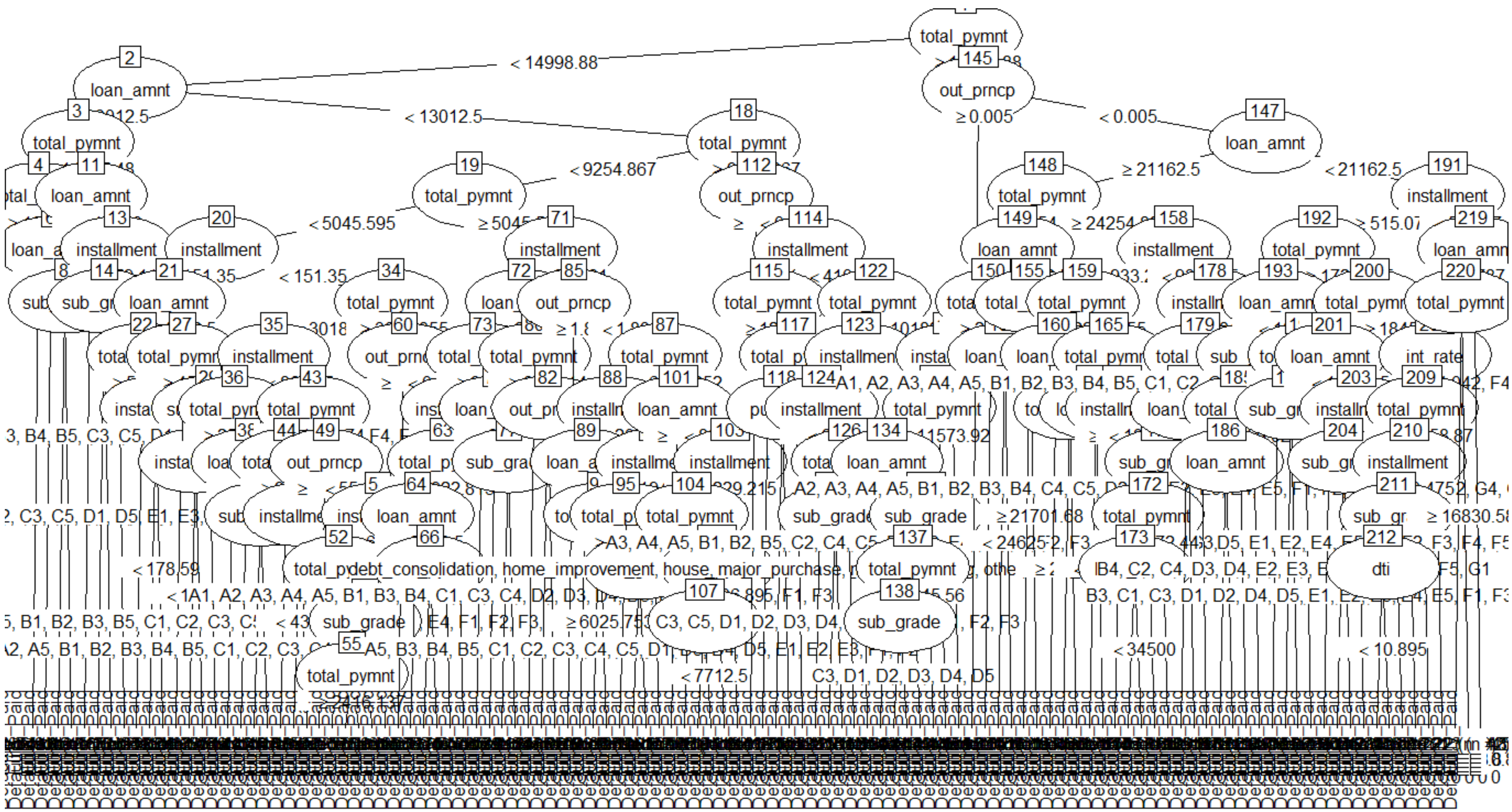
```
> cartWMod <- train(x = loan_train[, predictors],
+                   y = loan_train$loan_status,
+                   method = "rpart",
+                   trControl = ctrlNoProb,
+                   tuneLength = 30,
+                   metric = "Kappa",
+                   parms = list(loss = costMatrix))
> cartWMod
CART

160950 samples
    19 predictor
     2 classes: 'Default', 'Fully_Paid'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 144855, 144855, 144855, 144855, 144855, 144855, ...
Resampling results across tuning parameters:

  cp            Accuracy   Kappa      Sens       Spec
  0.0001882442  0.9844672  0.9594577  0.9485389  0.9973576
  0.0002117747  0.9839826  0.9581654  0.9465858  0.9973998
  0.0002353052  0.9838708  0.9578660  0.9461152  0.9974167
  0.0002588357  0.9835539  0.9570186  0.9448210  0.9974505
  0.0002823662  0.9832805  0.9562877  0.9437150  0.9974758
  0.0003058967  0.9833675  0.9565175  0.9439974  0.9974927
  0.0003137403  0.9832992  0.9563352  0.9437385  0.9974927
  0.0003294273  0.9831687  0.9559890  0.9432679  0.9974842
  0.0004353146  0.9835663  0.9570463  0.9446328  0.9975349
  0.0004941409  0.9835601  0.9570290  0.9446090  0.9975349
  0.0005529672  0.9832122  0.9560965  0.9431736  0.9975771
  0.0006117935  0.9822554  0.9535343  0.9393852  0.9976362
  0.0007647419  0.9815968  0.9517651  0.9366793  0.9977122
  0.0008235682  0.9818453  0.9524326  0.9376441  0.9977037
  0.0009059250  0.9820441  0.9529653  0.9384440  0.9976868
  0.0010118123  0.9817832  0.9522672  0.9375027  0.9976700
  0.0010588734  0.9814974  0.9514970  0.9363971  0.9976784
  0.0011529954  0.9810749  0.9503638  0.9348675  0.9976531
  0.0015883100  0.9815906  0.9517499  0.9368915  0.9976277
  0.0022412819  0.9792979  0.9456001  0.9283967  0.9975602
  0.0024001129  0.9790308  0.9448779  0.9273378  0.9975771
  0.0028001318  0.9790991  0.9450655  0.9276437  0.9975602
```

```
  0.0028824886  0.9789127  0.9445555  0.9269378  0.9975602
  0.0065885453  0.9682448  0.9153330  0.8859481  0.9977713
  0.0110828745  0.9667723  0.9112422  0.8801588  0.9978472
  0.0173184620  0.9667785  0.9112618  0.8802294  0.9978304
  0.0297425761  0.9512333  0.8671011  0.8208148  0.9980245
  0.0819450327  0.8352842  0.4702048  0.3761823  1.0000000
  0.1362181750  0.8352842  0.4702048  0.3761823  1.0000000
  0.3762059391  0.7650761  0.1383425  0.1102999  1.0000000

Kappa was used to select the optimal model using  the largest value.
The final value used for the model was cp = 0.0001882442.
> cartWModROC <-
+   pROC::roc(eval_results$loan_status,
+             eval_results$cartWMod,
+             levels = rev(levels(eval_results$loan_status)))
> cartWModROC

Call:
roc.default(response = eval_results$loan_status, predictor = eval_results$cartWMod,     levels = 
rev(levels(eval_results$loan_status)))

Data: eval_results$cartWMod in 29613 controls (eval_results$loan_status Fully_Paid) < 10624 cases 
(eval_results$loan_status Default).
Area under the curve: 0.9941
> cartWModEvalCM <-
+   confusionMatrix(predict(cartWMod, loan_eval), eval_results$loan_status)
> cartWModEvalCM
Confusion Matrix and Statistics

            Reference
Prediction   Default Fully_Paid
  Default      10072         65
  Fully_Paid     552      29548

               Accuracy : 0.9847
                 95% CI : (0.9834, 0.9858)
    No Information Rate : 0.736
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.96
 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.9480
            Specificity : 0.9978
         Pos Pred Value : 0.9936
         Neg Pred Value : 0.9817
             Prevalence : 0.2640
         Detection Rate : 0.2503
   Detection Prevalence : 0.2519
      Balanced Accuracy : 0.9729

       'Positive' Class : Default
```

CART Trees:
Complete training set



CART Tree: Down-sampled training data



CART Tree: Cost Weighted CART Model