National College of Ireland

Configuration Manual

MSc Research Project
MSc in FinTech

Anisa Nizar Ahmed
Student ID: x18107656

School of Computing
National College of Ireland

Supervisor: Victor Del Rosal

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Anisa Nizar Ahmed

Student ID: x18107656

Programme: MSc in FinTech **Year:** 2018-19

Module: MSc FinTech Research Project

Lecturer: Victor Del Rosal

Submission Due Date: 16/09/2019

Project Title: Credit-Risk Assessment of Small Business Loans using Naïve Bayes, Decision Tree and Random Forest

Word Count: 2100 **Page Count:** 10

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Anisa Nizar Ahmed

Date: 16/09/2019

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Anisa Nizar Ahmed
Student ID: x18107656

1 Introduction

The following manual explains the design and setup of tools and software used in the research paper. Screenshots of the codes used are listed along with a brief description of the steps.

2 Data Set

The data set is downloaded from Kaggle[1] in Excel sheet. It is then imported to R Studio where feature engineering and data mining is performed.

3 System Setup

The project is performed on a system with the following specifications:

Operating System: Windows 10 Home 64-bit (10.0, Build 17134)
System Model: 80YD
Processor: Intel® Core™ i5-7200U CPU @ 2.50 GHz (4CPUs), ~ 2.7GHz
Memory: 6 GB RAM, 1TB HDD

MS Word is used for writing the report and the analysis is performed on R Studio version 3.5.1.

4 R Studio

The following libraries were used in R Studio.

- randomforest
  caret
- lattice
- naivebayes
- dplyr
- ggplot2
- psych
- e1071
- party
- grid
- mvtnorm
  modeltools
- zoo
- party
- rpart
- rpart.plot

[1] https://www.kaggle.com/wendykan/lending-club-loan-data#loan.csv

5 Random Forest

After loading the data, some of the variables were deleted that contained personal information such as member ID and address. Feature Engineering was performed on the data that consisted of converting some numeric variables to factor variables and replacing missing values with 0. Missing values were replaced with 0 since they were negligible in number.

```
> #Feature Engineering
> data4$companyage <- factor(data4$`Company Age`)
> data4$homeownership <- factor(data4$home_ownership)
> data4$verficationstatus <- factor(data4$verification_status)
> data4$loanstatus <- factor(data4$loan_status)
> data4$disbursementmethod <- factor(data4$disbursement_method)
> data4$Grade <- factor(data4$grade)
> data4$Term <- factor(data4$term)
>
> data4$member_id <- NULL
> data4$funded_amnt <- NULL
> data4$funded_amnt_inv <- NULL
> data4$issue_d <- NULL
> data4$zip_code <- NULL
> data4$addr_state<- NULL
> data4$policy_code <- NULL
> data4$application_type <- NULL
> data4$annual_inc_joint <- NULL
> data4$dti_joint <- NULL
> data4$verification_status_joint <-NULL
> data4$tot_cur_bal<- NULL
> data4$total_bal_il <- NULL
> data4$next_pymnt_d <- NULL
> data4$last_credit_pull_d<- NULL
> data4$last_pymnt_d<- NULL
> data4$sub_grade <- NULL
> data4$grade <- NULL
> data4$disbursement_method <- NULL
> data4$loan_status <-NULL
> data4$verification_status <- NULL
> data4$home_ownership <- NULL
> data4$`Company Age`<- NULL
> data4$term <- NULL

> data4[is.na(data4)] <- 0
```

Figure 1: Feature Engineering

The data was split into 70% training and 30% test data after which randomforest model is fitted in to the data and its accuracy was measured.

```
> set.seed(1234)
> ind4 <- sample(2, nrow(data4), replace = TRUE, prob = c(0.7, 0.3))
> training4 <- data4[ind4==1,]
> test4 <- data4[ind4==2,]
> library(randomForest)
>
> set.seed(222)
> rf <- randomForest(Grade~., data = training4)
> print(rf)

Call:
 randomForest(formula = Grade ~ ., data = training4)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 4

        OOB estimate of  error rate: 0.17%
```

```
Confusion matrix:
      A     B     C    D    E  F  G  class.error
A 20163     0     0    0    0  0  0 0.000000e+00
B     1 19707     0    0    0  0  0 5.074082e-05
C     3     0 16997    0    0  0  0 1.764706e-04
D     0     0     0 9323    4  0  0 4.288624e-04
E     1     0     1   10 3359  0  0 3.559775e-03
F     0     0     0    0   69 52  2 5.772358e-01
G     0     0     0    0   19  9 10 7.368421e-01
```

Figure 2: Random Forest Model

The importance of each variable is identified using ‘randomforest’ library function.

```
> importance(rf)
                   MeanDecreaseGini
loan_amnt               1019.211015
int_rate               33341.617287
installment             1509.959535
annualrevenue            225.508832
dti                      260.332885
delinq_2yrs               61.771973
inq_last_6mths            74.689031
revol_bal                248.298411
revol_util               553.098754
total_acc                168.203071
out_prncp               1284.258958
out_prncp_inv           1277.172722
total_pymnt             1265.980660
total_pymnt_inv         1240.501420
total_rec_prncp         3241.616056
total_rec_int           3878.750381
total_rec_late_fee         4.232247
last_pymnt_amnt         1323.784716
companyage               277.849037
homeownership             56.018953
verficationstatus         95.300842
loanstatus                38.207582
disbursementmethod       756.952597
Term                     565.531563
> varImpPlot(rf)
```

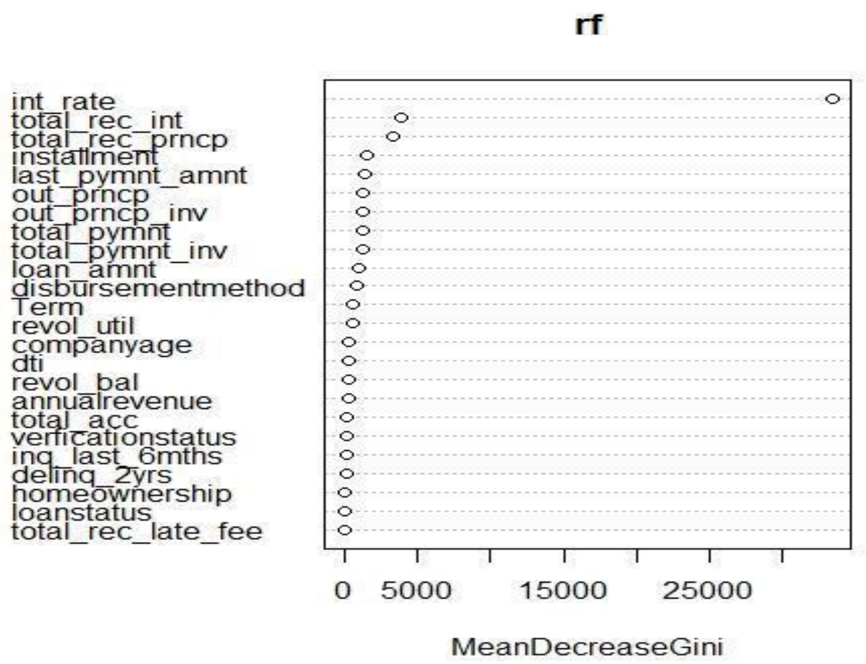


Figure 3: Variable Importance

The model was fitted to the test data and accuracy is calculated using Confusion Matrix from Caret library.

```
> confusionMatrix(table(p2, test4$Grade))
Confusion Matrix and Statistics


p2      A    B    C    D    E    F    G
  A 8687    0    1    0    1    0    0
  B    0 8312    0    0    0    0    0
  C    0    0 7453    1    0    0    0
  D    0    0    0 3959    4    1    0
  E    0    0    0    2 1483   27   11
  F    0    0    0    0    0   21    4
  G    0    0    0    0    0    0    2

Overall Statistics

               Accuracy : 0.9983
                 95% CI : (0.9977, 0.9987)
    No Information Rate : 0.2899
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9977
 Mcnemar's Test P-Value : NA

Statistics by Class:

           Class: A Class: B Class: C Class: D Class: E  Class: F  Class: G
Sensitivity  1.0000   1.0000   0.9999   0.9992  0.99664 0.4285714 1.176e-01
Specificity  0.9999   1.0000   1.0000   0.9998  0.99860 0.9998663 1.000e+00
Pos Pred Val 0.9998   1.0000   0.9999   0.9987  0.97374 0.8400000 1.000e+00
Neg Pred Val 1.0000   1.0000   1.0000   0.9999  0.99982 0.9990649 9.995e-01
Prevalence   0.2899   0.2774   0.2487   0.1322  0.04965 0.0016350 5.673e-04
DetectionRate0.2899   0.2774   0.2487   0.1321  0.04948 0.0007007 6.674e-05
Det Prevalenc0.2899   0.2774   0.2487   0.1323  0.05082 0.0008342 6.674e-05
Balanced Acc 1.0000   1.0000   0.9999   0.9995  0.99762 0.7142189 5.588e-01
```

Figure 5: Confusion Matrix – Test Data

6 Naïve Bayes

For implementing Naïve Bayes, following packages are loaded: naivebayes, gplyr, ggplot2 and psych. Response Variable ‘Grade’ was converted to numeric variable to calculate the correlation coefficient after which it was converted back to factor form.

```
> #Converting 'Grade' to numeric variable for correlation
> data2$Grade= as.numeric(data2$Grade)
>
> #Correlation
> A <- cor(data2$Grade, data2$int_rate)
> A
[1] 0.9731721
>
> #Convering Grade back to factor variable
> data2$GRADE <- factor(data2$Grade)
> data2$Grade <- NULL
>
```

Figure 6: Correlation Coefficient

The data was partitioned into 80% training and 20% test data for implementing Naïve Bayes. Confusion Matrix was used to calculate the accuracy of predicted model.

```
> confusionMatrix(table(q, test1$GRADE))
Confusion Matrix and Statistics


q      1    2    3    4    5    6    7
  1 5618    4    1    0    0    0    0
  2   58 5329   79    0    1    0    0
  3    7   87 4686   23    1    0    0
  4    4    4   33 2428   16    0    0
  5    1    3    8   35  836   22    1
  6    0    0    0    1    3    8    0
  7   78  131  138  123  118    3   11

Overall Statistics

               Accuracy : 0.9506
                 95% CI : (0.9475, 0.9536)
    No Information Rate : 0.2898
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9352
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 1 Class: 2 Class: 3 Class: 4 Class: 5 Class: 6  Class: 7
Sensitivity            0.9743   0.9588   0.9476   0.9303  0.85744 0.242424 0.9166667
Specificity            0.9996   0.9904   0.9921   0.9967  0.99630 0.999799 0.9702821
Pos Pred Value         0.9991   0.9748   0.9754   0.9771  0.92274 0.666667 0.0182724
Neg Pred Value         0.9896   0.9841   0.9828   0.9895  0.99268 0.998743 0.9999482
Prevalence             0.2898   0.2793   0.2485   0.1312  0.04900 0.001658 0.0006030
Detection Rate         0.2823   0.2678   0.2355   0.1220  0.04201 0.000402 0.0005528
Detection Prevalence   0.2826   0.2747   0.2414   0.1249  0.04553 0.000603 0.0302528
Balanced Accuracy      0.9870   0.9746   0.9699   0.9635  0.92687 0.621111 0.9434744
```

Figure 7: Confusion Matrix – Naïve Bayes

7 Decision Tree

The data was split in to 70% training and 30% test data and the model was implemented into the training data.

```
> set.seed(1234)
>
> partidata <- sample(2,nrow(dtdata),replace=TRUE, prob=c(0.7,0.3))
> trainingdata <- dtdata[partidata==1,]
> validatedata <- dtdata[partidata==2,]
>
> library(party)
> library(grid)
> library(mvtnorm)
> library(modeltools)
> library(stats4)
> library(strucchange)
> library(zoo)
> library(party)
> library(rpart)
> library(rpart.plot)
>
> #Decision Tree with all variables
> decisiontrees <- ctree(Grade~ ., data=trainingdata, controls =
ctree_control(mincriterion=0.95, minsplit=5000))
> decisiontrees

          Conditional inference tree with 7 terminal nodes
```

```
Response:  Grade
Inputs:  loan_amnt, int_rate, installment, annualrevenue, dti, delinq_2yrs,
inq_last_6mths, revol_bal, revol_util, total_acc, out_prncp, out_prncp_inv,
total_pymnt, total_pymnt_inv, total_rec_prncp, total_rec_int, total_rec_late_fee,
last_pymnt_amnt, companyage, homeownership, verficationstatus, loanstatus,
disbursementmethod, Term
Number of observations:  69730

1) int_rate <= 8.81; criterion = 1, statistic = 66471.559
  2) total_rec_int <= 839.86; criterion = 1, statistic = 52.276
    3) delinq_2yrs <= 1; criterion = 0.996, statistic = 35.988
      4)*  weights = 19538
    3) delinq_2yrs > 1
      5)*  weights = 422
  2) total_rec_int > 839.86
    6)*  weights = 208
1) int_rate > 8.81
  7) int_rate <= 12.98; criterion = 1, statistic = 45937.168
    8)*  weights = 19707
  7) int_rate > 12.98
    9) int_rate <= 16.91; criterion = 1, statistic = 26238.769
      10)*  weights = 16997
    9) int_rate > 16.91
      11) int_rate <= 22.35; criterion = 1, statistic = 9949.402
        12)*  weights = 9327
      11) int_rate > 22.35
        13)*  weights = 3531
> plot(decisiontrees)
```

Figure 8 (a): Decision Tree Model – Training Data

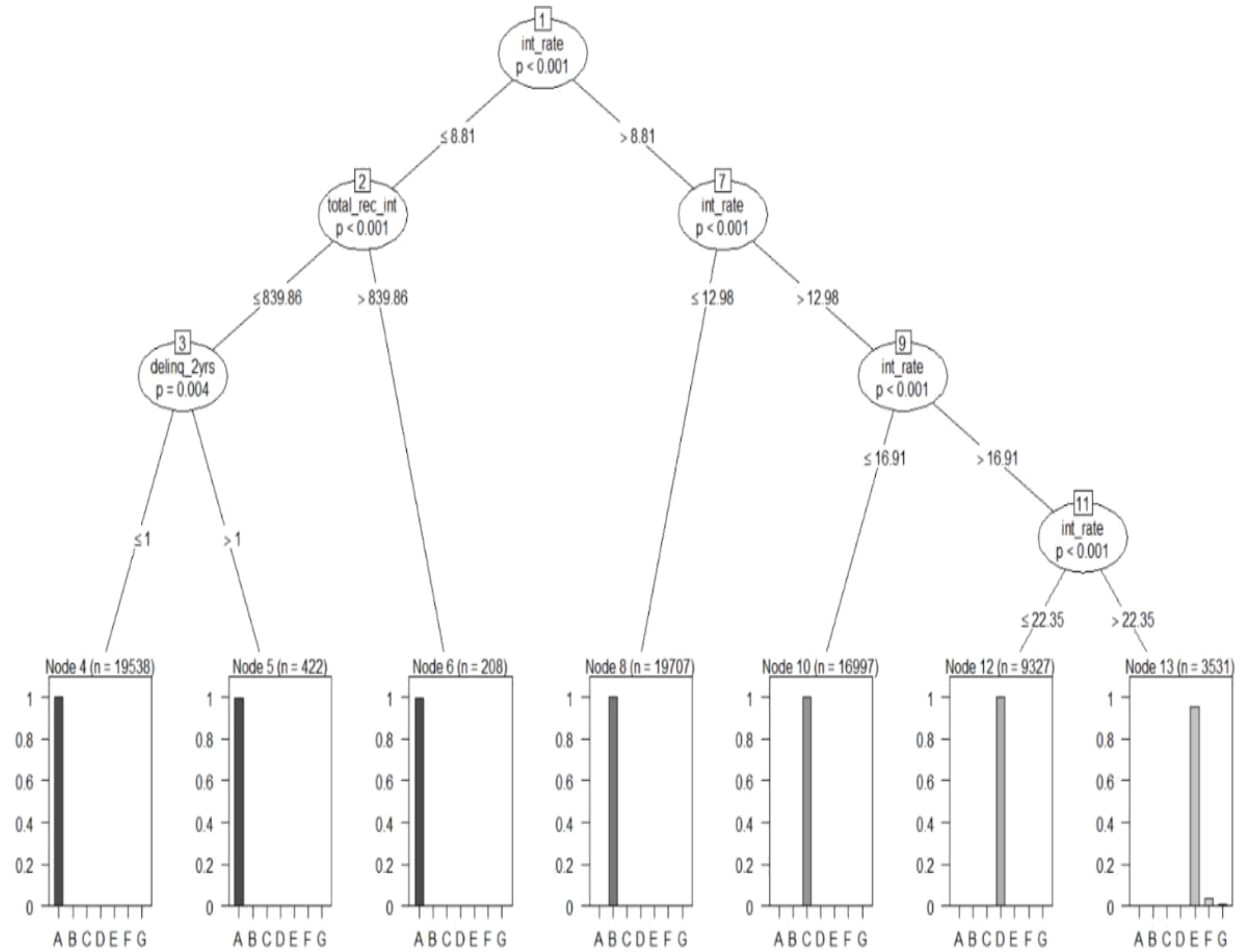


Figure 8 (b): Decision Tree

The model was trained again with only selected number of independent variables of the same dataset.

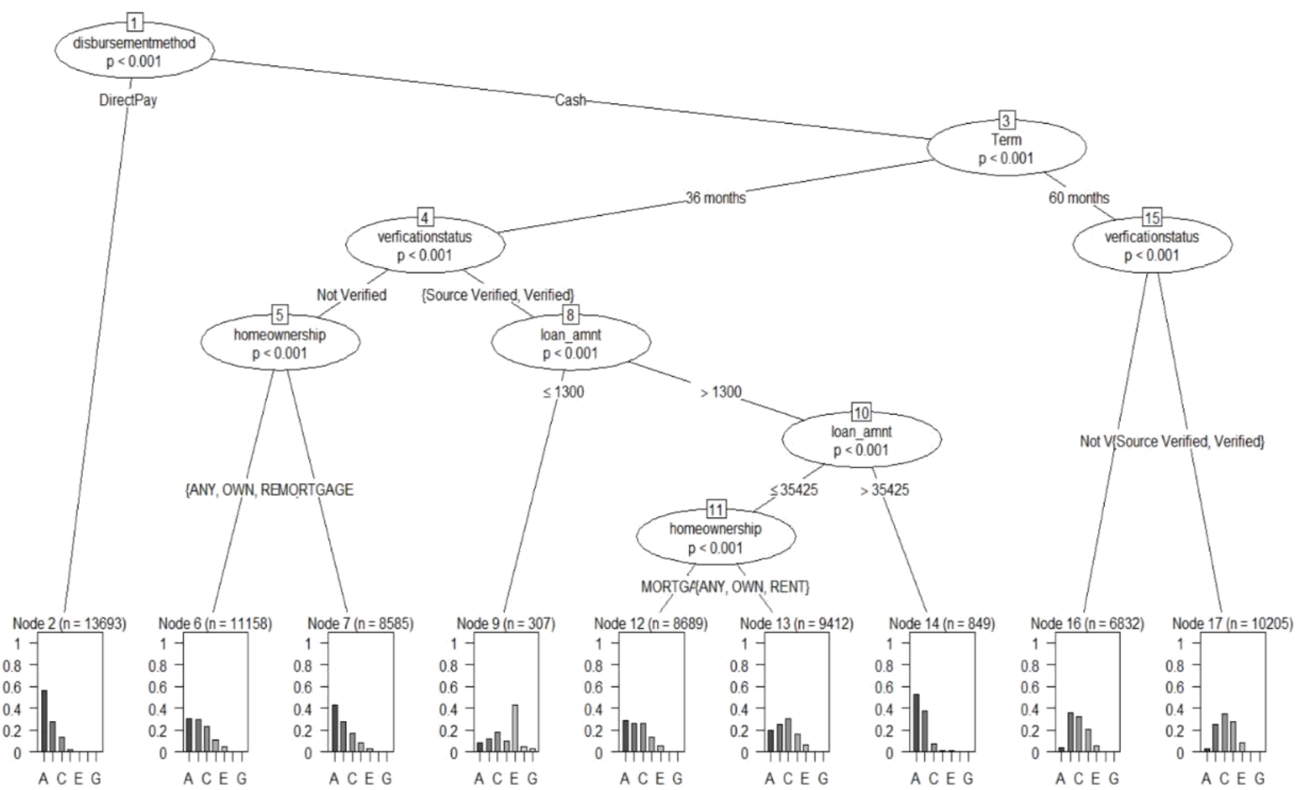
```
> #Decision Tree with selected variables
> decisiontrees3 <- ctree(Grade ~ loan_amnt + annualrevenue + dti + revol_bal +
companyage + homeownership + verficationstatus + disbursementmethod + Term, data =
trainingdata, controls = ctree_control(mincriterion = 0.95, minsplit = 15000))
> decisiontrees3

         Conditional inference tree with 9 terminal nodes

Response:  Grade
Inputs:  loan_amnt, annualrevenue, dti, revol_bal, companyage, homeownership,
verficationstatus, disbursementmethod, Term
Number of observations:  69730

1) disbursementmethod == {DirectPay}; criterion = 1, statistic = 7531.821
  2)*  weights = 13693
1) disbursementmethod == {Cash}
  3) Term == {36 months}; criterion = 1, statistic = 5780.714
    4) verficationstatus == {Not Verified}; criterion = 1, statistic = 1027.551
      5) homeownership == {ANY, OWN, RENT}; criterion = 1, statistic = 421.027
        6)*  weights = 11158
      5) homeownership == {MORTGAGE}
        7)*  weights = 8585
    4) verficationstatus == {Source Verified, Verified}
      8) loan_amnt <= 1300; criterion = 1, statistic = 612.569
        9)*  weights = 307
      8) loan_amnt > 1300
        10) loan_amnt <= 35425; criterion = 1, statistic = 506.382
          11) homeownership == {MORTGAGE}; criterion = 1, statistic = 247.45
            12)*  weights = 8689
          11) homeownership == {ANY, OWN, RENT}
            13)*  weights = 9412
        10) loan_amnt > 35425
          14)*  weights = 849
  3) Term == {60 months}
    15) verficationstatus == {Not Verified}; criterion = 1, statistic = 419.069
      16)*  weights = 6832
    15) verficationstatus == {Source Verified, Verified}
      17)*  weights = 10205
> plot(decisiontrees3)
```

Figure 9: Decision Tree with selected number of variables



The model was fitted to test data twice; once with all the independent variables and then with only selected variables to compare the accuracy of the model in both the cases.

```
> #Misclassification error for first tree
> testpred <- predict(decisiontrees, newdata=validatedata)
> tab <- table(testpred,validatedata$Grade)
> print(tab)

testpred    A    B    C    D    E    F    G
       A 8687    0    1    0    1    0    0
       B    0 8312    0    0    0    0    0
       C    0    0 7453    0    0    0    0
       D    0    0    0 3962    0    0    0
       E    0    0    0    0 1487   49   17
       F    0    0    0    0    0    0    0
       G    0    0    0    0    0    0    0
> 1-sum(diag(tab))/sum(tab)
[1] 0.002269011
>
> #Misclassification error for second tree
> testpred1 <- predict(decisiontrees3, newdata=validatedata)
> tab2 <- table(testpred1, validatedata$Grade)
> print(tab2)

testpred1    A    B    C    D    E    F    G
        A 7660 5159 3636 1449  575   11    8
        B  107 1052  953  626  165   10    1
        C  913 2086 2842 1861  688   23    5
        D    0    0    0    0    0    0    0
        E    7   15   23   26   60    5    3
        F    0    0    0    0    0    0    0
        G    0    0    0    0    0    0    0
> 1-sum(diag(tab2))/sum(tab2)
[1] 0.6124662
```

Figure 10: Comparison of Accuracy