Data622_HW3

Mengqin Cai

4/5/2021

Problem 1: KNN

Please use K-nearest neighbor (KNN) algorithm to predict the species variable. Please be sure to walk through the steps you took. (40 points)

```r
head(penguins)
```

```
## Registered S3 method overwritten by 'cli':
##   method     from
##   print.tree tree

## # A tibble: 6 x 8
##   species island bill_length_mm bill_depth_mm flipper_length_~ body_mass_g
##   <fct>   <fct>           <dbl>         <dbl>            <int>       <int>
## 1 Adelie  Torge~           39.1          18.7              181        3750
## 2 Adelie  Torge~           39.5          17.4              186        3800
## 3 Adelie  Torge~           40.3          18                195        3250
## 4 Adelie  Torge~           NA            NA                 NA          NA
## 5 Adelie  Torge~           36.7          19.3              193        3450
## 6 Adelie  Torge~           39.3          20.6              190        3650
## # ... with 2 more variables: sex <fct>, year <int>
```

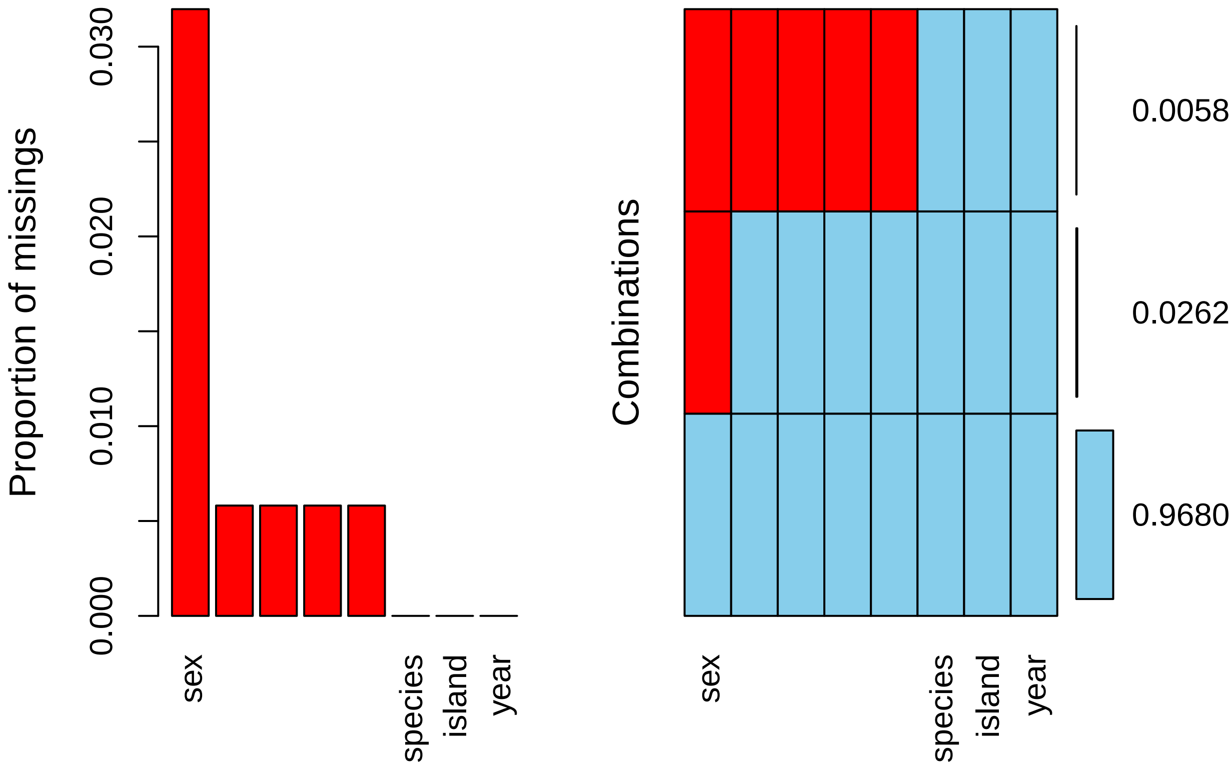
```r
summary(penguins)
```

```
##       species          island    bill_length_mm  bill_depth_mm
##  Adelie   :152   Biscoe   :168   Min.   :32.10   Min.   :13.10
##  Chinstrap: 68   Dream    :124   1st Qu.:39.23   1st Qu.:15.60
##  Gentoo   :124   Torgersen: 52   Median :44.45   Median :17.30
##                                  Mean   :43.92   Mean   :17.15
##                                  3rd Qu.:48.50   3rd Qu.:18.70
##                                  Max.   :59.60   Max.   :21.50
##                                  NA's   :2       NA's   :2
##  flipper_length_mm  body_mass_g       sex           year
##  Min.   :172.0     Min.   :2700   female:165   Min.   :2007
##  1st Qu.:190.0     1st Qu.:3550   male  :168   1st Qu.:2007
##  Median :197.0     Median :4050   NA's  : 11   Median :2008
##  Mean   :200.9     Mean   :4202                Mean   :2008
##  3rd Qu.:213.0     3rd Qu.:4750                3rd Qu.:2009
##  Max.   :231.0     Max.   :6300                Max.   :2009
##  NA's   :2         NA's   :2
```

To better evaluate the model, I split the dataset into training and test set.

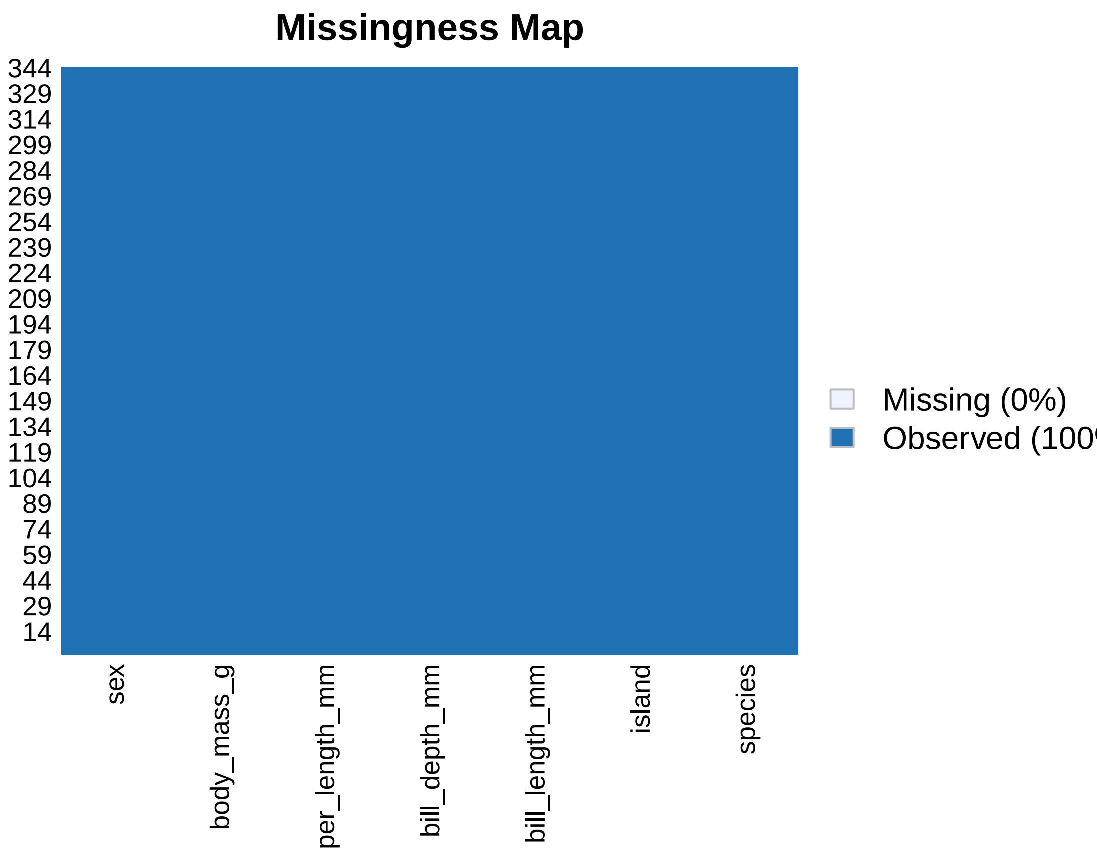
First, check the missing value of the whole dataset and use KNN imputation to impute the dataset

```
aggr(penguins,bars=T, numbers=T, sortVars=T)
```




```
##
##  Variables sorted by number of missings:
##           Variable       Count
##                sex 0.031976744
##     bill_length_mm 0.005813953
##      bill_depth_mm 0.005813953
##  flipper_length_mm 0.005813953
##        body_mass_g 0.005813953
##            species 0.000000000
##             island 0.000000000
##               year 0.000000000
```

```
penguins<-kNN(penguins)
penguins<-subset(penguins,select=species:sex)
missmap(penguins)
```




```r
levels(penguins$species) <- c("Adelie", "Chinstrap", "Gentoo")
penguins$species<-as.numeric(penguins$species)

levels(penguins$island) <- c("Biscoe", "Dream", "Torgersen")
penguins$island<-as.numeric(penguins$island)

levels(penguins$sex) <- c("female", "male")
penguins$sex<-as.numeric(penguins$sex)
```

```r
normalize <- function(x) {
return ((x - min(x)) / (max(x) - min(x))) }
head(penguins)
```

```
##   species island bill_length_mm bill_depth_mm flipper_length_mm
## 1       1      3           39.1          18.7               181
## 2       1      3           39.5          17.4               186
## 3       1      3           40.3          18.0               195
## 4       1      3           37.8          18.1               190
## 5       1      3           36.7          19.3               193
## 6       1      3           39.3          20.6               190
##   body_mass_g sex
## 1        3750   2
## 2        3800   1
## 3        3250   1
## 4        3700   1
```

```
## 5        3450   1
## 6        3650   2
```

```
penguins_Trans<- as.data.frame(lapply(penguins[,3:6], normalize))
penguins_Trans<-cbind(penguins[,1],penguins[,2],penguins_Trans,penguins[,7] )
colnames(penguins_Trans)<-c("species","island","bill_length_mm","bill_depth_mm","flipper_length_mm","bo
head(penguins_Trans)
```

```
##   species island bill_length_mm bill_depth_mm flipper_length_mm
## 1       1      3      0.2545455     0.6666667         0.1525424
## 2       1      3      0.2690909     0.5119048         0.2372881
## 3       1      3      0.2981818     0.5833333         0.3898305
## 4       1      3      0.2072727     0.5952381         0.3050847
## 5       1      3      0.1672727     0.7380952         0.3559322
## 6       1      3      0.2618182     0.8928571         0.3050847
##   body_mass_g sex
## 1   0.2916667   2
## 2   0.3055556   1
## 3   0.1527778   1
## 4   0.2777778   1
## 5   0.2083333   1
## 6   0.2638889   2
```

```
sample_size<-floor(0.8*nrow(penguins))
set.seed(123)
train_ind<-sample(seq_len(nrow(penguins)),size=sample_size)
train_penguins<-penguins_Trans[train_ind,]
test_penguins<-penguins_Trans[-train_ind,]
```

Fit the model

```
set.seed(123)
sqrt(nrow(train_penguins))
```

```
## [1] 16.58312
```

```
k16<-knn(train_penguins,test_penguins,cl=train_penguins$species,k=16)
k17<-knn(train_penguins,test_penguins,cl=train_penguins$species,k=17)
```

```
misClassError <- mean(k16 != test_penguins$species)
misClassError
```

```
## [1] 0
```

```
table(k16,test_penguins$species)
```

```
##
## k16  1  2  3
##   1 28  0  0
##   2  0 16  0
##   3  0  0 25
```

```r
misClassError <- mean(k17 != test_penguins$species)
misClassError
```

```
## [1] 0
```

```r
table(k17,test_penguins$species)
```

```
##
## k17  1  2  3
##   1 28  0  0
##   2  0 16  0
##   3  0  0 25
```

There is no different with K16 or K17, we can choose either of the model.

Problem 2: Decision Trees

Please use the attached dataset on loan approval status to predict loan approval using Decision Trees. Please be sure to conduct a thorough exploratory analysis to start thetask and walk us through your reasoning behind all the steps you are taking.

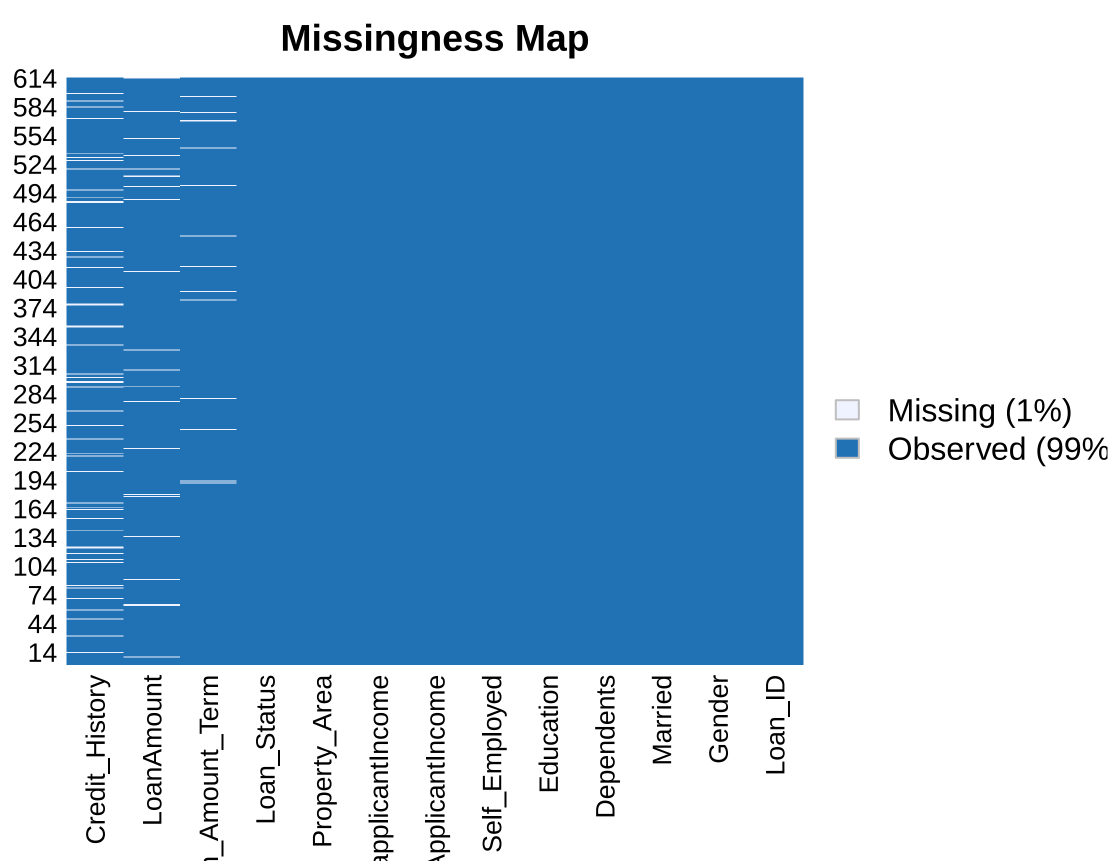
```r
loan<-read.csv("https://raw.githubusercontent.com/DaisyCai2019/NewData/master/Loan_approval.csv")
head(loan)
```

```
##    Loan_ID Gender Married Dependents    Education Self_Employed
## 1 LP001002   Male      No          0     Graduate            No
## 2 LP001003   Male     Yes          1     Graduate            No
## 3 LP001005   Male     Yes          0     Graduate           Yes
## 4 LP001006   Male     Yes          0 Not Graduate            No
## 5 LP001008   Male      No          0     Graduate            No
## 6 LP001011   Male     Yes          2     Graduate           Yes
##   ApplicantIncome CoapplicantIncome LoanAmount Loan_Amount_Term
## 1            5849                 0         NA              360
## 2            4583              1508        128              360
## 3            3000                 0         66              360
## 4            2583              2358        120              360
## 5            6000                 0        141              360
## 6            5417              4196        267              360
##   Credit_History Property_Area Loan_Status
## 1              1         Urban           Y
## 2              1         Rural           N
## 3              1         Urban           Y
## 4              1         Urban           Y
## 5              1         Urban           Y
## 6              1         Urban           Y
```

```r
summary(loan)
```

```
##      Loan_ID       Gender     Married    Dependents        Education
##  LP001002:  1         : 13       :  3      : 15     Graduate    :480
##  LP001003:  1   Female:112   No :213    0 :345     Not Graduate:134
##  LP001005:  1   Male  :489   Yes:398    1 :102
##  LP001006:  1                           2 :101
##  LP001008:  1                           3+: 51
##  LP001011:  1
##  (Other) :608
##  Self_Employed ApplicantIncome CoapplicantIncome   LoanAmount
##     : 32       Min.   :  150   Min.   :     0     Min.   :  9.0
##  No :500       1st Qu.: 2878   1st Qu.:     0     1st Qu.:100.0
##  Yes: 82       Median : 3812   Median : 1188      Median :128.0
##                Mean   : 5403   Mean   : 1621      Mean   :146.4
##                3rd Qu.: 5795   3rd Qu.: 2297      3rd Qu.:168.0
##                Max.   :81000   Max.   :41667      Max.   :700.0
##                                                   NA's   :22
##  Loan_Amount_Term Credit_History     Property_Area Loan_Status
##  Min.   : 12      Min.   :0.0000   Rural    :179   N:192
##  1st Qu.:360      1st Qu.:1.0000   Semiurban:233   Y:422
##  Median :360      Median :1.0000   Urban    :202
##  Mean   :342      Mean   :0.8422
##  3rd Qu.:360      3rd Qu.:1.0000
##  Max.   :480      Max.   :1.0000
##  NA's   :14       NA's   :50
```

```r
missmap(loan)
```



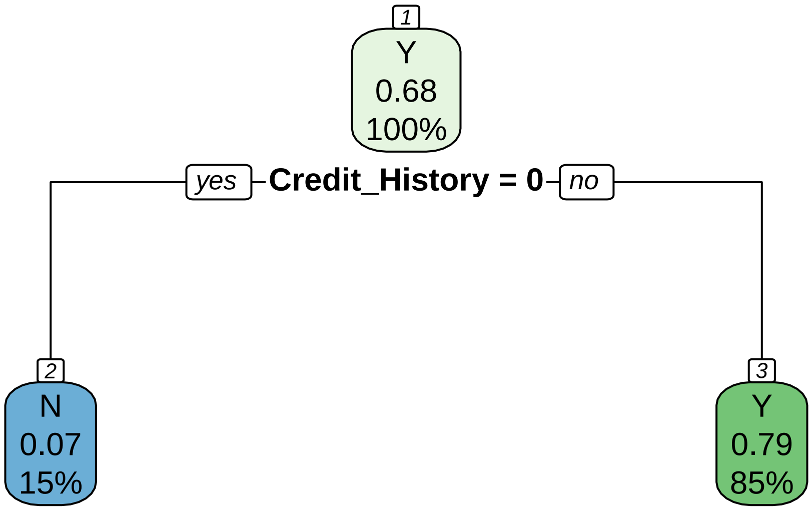
```r
loanTrans<-kNN(loan)%>%
          subset(select = Loan_ID:Loan_Status)
```

```r
loanTrans$Loan_Status<-factor(loanTrans$Loan_Status)


loanTrans<-loanTrans %>%
           mutate(Gender = factor(Gender),
                  Married = factor(Married),
                  Dependents=factor(Dependents),
                  Education=factor(Education),
                  Self_Employed=factor(Self_Employed),
                  Property_Area=factor(Property_Area),
                  Loan_Status=factor(Loan_Status))

summary(loanTrans)
```

```
##      Loan_ID       Gender    Married   Dependents        Education
##   LP001002:  1         : 13      :  3     : 15     Graduate    :480
##   LP001003:  1   Female:112   No :213   0 :345     Not Graduate:134
##   LP001005:  1   Male  :489   Yes:398   1 :102
##   LP001006:  1                          2 :101
##   LP001008:  1                          3+: 51
##   LP001011:  1
##   (Other) :608
##   Self_Employed ApplicantIncome CoapplicantIncome   LoanAmount
##      : 32       Min.   :  150   Min.   :    0     Min.   :  9.0
##   No :500       1st Qu.: 2878   1st Qu.:    0     1st Qu.:100.0
##   Yes: 82       Median : 3812   Median : 1188     Median :128.0
##                 Mean   : 5403   Mean   : 1621     Mean   :145.6
##                 3rd Qu.: 5795   3rd Qu.: 2297     3rd Qu.:165.8
##                 Max.   :81000   Max.   :41667     Max.   :700.0
##
##   Loan_Amount_Term Credit_History     Property_Area Loan_Status
##   Min.   : 12.0    Min.   :0.0000   Rural    :179   N:192
##   1st Qu.:360.0    1st Qu.:1.0000   Semiurban:233   Y:422
##   Median :360.0    Median :1.0000   Urban    :202
##   Mean   :342.4    Mean   :0.8485
##   3rd Qu.:360.0    3rd Qu.:1.0000
##   Max.   :480.0    Max.   :1.0000
##
```

```r
set.seed(123)
train_sample<-sample(1:nrow(loanTrans),size = floor(0.80*nrow(loanTrans)))
train_loan<-loanTrans[train_sample,]
test_loan<-loanTrans[-train_sample,]
```

```r
tree<- rpart(Loan_Status~Gender+Married+Dependents+Education+Self_Employed+ApplicantIncome+CoapplicantI
rpart.plot(tree,nn=TRUE)
```




```r
summary(tree)
```

```
## Call:
## rpart(formula = Loan_Status ~ Gender + Married + Dependents +
##     Education + Self_Employed + ApplicantIncome + CoapplicantIncome +
##     LoanAmount + Loan_Amount_Term + Credit_History + Property_Area,
##     data = train_loan)
##   n= 491
##
##          CP nsplit rel error    xerror       xstd
## 1 0.4102564      0 1.0000000 1.0000000 0.06613317
## 2 0.0100000      1 0.5897436 0.5897436 0.05542619
##
## Variable importance
##  Credit_History ApplicantIncome
##              99               1
##
## Node number 1: 491 observations,    complexity param=0.4102564
##   predicted class=Y  expected loss=0.3177189  P(node) =1
##     class counts:   156   335
##    probabilities: 0.318 0.682
##   left son=2 (74 obs) right son=3 (417 obs)
##   Primary splits:
##       Credit_History    < 0.5    to the left,  improve=65.849520, (0 missing)
##       ApplicantIncome   < 1858   to the left,  improve= 2.531041, (0 missing)
```

```
##       CoapplicantIncome < 8656.5 to the right, improve= 2.233556, (0 missing)
##       LoanAmount        < 163    to the right, improve= 2.231237, (0 missing)
##       Property_Area     splits as  LRL,        improve= 2.078251, (0 missing)
##   Surrogate splits:
##       ApplicantIncome < 39573  to the right, agree=0.851, adj=0.014, (0 split)
##
## Node number 2: 74 observations
##   predicted class=N  expected loss=0.06756757  P(node) =0.1507128
##     class counts:    69     5
##    probabilities: 0.932 0.068
##
## Node number 3: 417 observations
##   predicted class=Y  expected loss=0.2086331  P(node) =0.8492872
##     class counts:    87   330
##    probabilities: 0.209 0.791
```

```r
loanPre<-predict(tree,test_loan,type="class")
table(loanPre,test_loan$Loan_Status)
```

```
##
## loanPre  N  Y
##       N 17  2
##       Y 19 85
```

```r
accuracy<-mean(loanPre==test_loan$Loan_Status)
accuracy
```

```
## [1] 0.8292683
```

Problem 3: Random Forests

Using the same dataset on Loan Approval Status, please use Random Forests to predict on loan approval status. Again, please be sure to walk us through the steps you took to get to your final model. (50 points)

```r
rf <- randomForest(Loan_Status~Gender+Married+Dependents+Education+Self_Employed+ApplicantIncome+Coappl
rf
```

```
##
## Call:
##  randomForest(formula = Loan_Status ~ Gender + Married + Dependents +      Education + Self_Employed
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 20.16%
## Confusion matrix:
##    N   Y class.error
## N 73  83  0.53205128
## Y 16 319  0.04776119
```

```
importance(rf)
```

```
##                   MeanDecreaseGini
## Gender                    5.240292
## Married                   4.672921
## Dependents               10.868038
## Education                 3.721062
## Self_Employed             6.076099
## ApplicantIncome          36.421362
## CoapplicantIncome        22.031365
## LoanAmount               35.010853
## Loan_Amount_Term          9.110774
## Credit_History           58.569036
## Property_Area             8.908209
```

```
rfPre<-predict(rf,test_loan)
table(rfPre,test_loan$Loan_Status)
```

```
##
## rfPre  N  Y
##     N 18  5
##     Y 18 82
```

```
accuracy2<-mean(rfPre==test_loan$Loan_Status)
accuracy2
```

```
## [1] 0.8130081
```

Problem 4: Gradient Boosting

Using the Loan Approval Status data, please use Gradient Boosting to predict on the loan approval status. Please use whatever boosting approach you deem appropriate;but please be sure to walk us through your steps. (50 points)

Problem 5: Model performance

Model performance: please compare the models you settled on for problem # 2 − 4.Comment on their relative performance. Which one would you prefer the most? Why?(20 points)