Untitled

#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 the task and walk us through your reasoning behind all the steps you are taking. (40 points)

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(dplyr)
library(caret)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
library(nnet)
library(forcats)
library(knitr)
library(rpart)
```

Import Data

```
loan <- read.csv("https://raw.githubusercontent.com/Zchen116/data-622/main/Loan_approval.csv")
```

```
head(loan)
```

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

```r
summary(loan)
```

```
##    Loan_ID             Gender             Married            Dependents       
##  Length:614         Length:614         Length:614         Length:614        
##  Class :character   Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character  
##                                                                             
##                                                                             
##                                                                             
##                                                                             
##   Education         Self_Employed      ApplicantIncome CoapplicantIncome
##  Length:614         Length:614         Min.   :  150   Min.   :    0    
##  Class :character   Class :character   1st Qu.: 2878   1st Qu.:    0    
##  Mode  :character   Mode  :character   Median : 3812   Median : 1188    
##                                        Mean   : 5403   Mean   : 1621    
##                                        3rd Qu.: 5795   3rd Qu.: 2297    
##                                        Max.   :81000   Max.   :41667    
##                                                                         
##    LoanAmount    Loan_Amount_Term Credit_History   Property_Area     
##  Min.   :  9.0   Min.   : 12      Min.   :0.0000   Length:614        
##  1st Qu.:100.0   1st Qu.:360      1st Qu.:1.0000   Class :character  
##  Median :128.0   Median :360      Median :1.0000   Mode  :character  
##  Mean   :146.4   Mean   :342      Mean   :0.8422                     
##  3rd Qu.:168.0   3rd Qu.:360      3rd Qu.:1.0000                     
##  Max.   :700.0   Max.   :480      Max.   :1.0000                     
##  NA's   :22      NA's   :14       NA's   :50                         
##  Loan_Status       
##  Length:614        
##  Class :character  
##  Mode  :character  
##                    
##                    
##                    
```

```
##
```

Clean Data

1, Remove N/A from the dataset 2, Combine ApplicantIncome and CoapplicantIncome 3, Remove the variable “Loan_ID”, “ApplicantIncome” and “CoapplicantIncome”

```r
data <- na.omit(loan) %>%
  mutate(TotalIncome = ApplicantIncome + CoapplicantIncome) %>%
  dplyr::select(-c(Loan_ID, ApplicantIncome, CoapplicantIncome))
```

```r
data <- transform(
  data,
  Gender = as.factor(Gender),
  Married = as.factor(Married),
  Dependents = as.factor(Dependents),
  Education = as.factor(Education),
  Self_Employed = as.factor(Self_Employed),
  LoanAmount = as.integer(LoanAmount),
  Loan_Amount_Term = as.integer(Loan_Amount_Term),
  Credit_History = as.factor(Credit_History),
  Property_Area = as.factor(Property_Area),
  Loan_Status = as.factor(Loan_Status))

sapply(data, class)
```

```
##           Gender          Married       Dependents        Education
##         "factor"         "factor"         "factor"         "factor"
##    Self_Employed       LoanAmount Loan_Amount_Term   Credit_History
##         "factor"        "integer"        "integer"         "factor"
##    Property_Area      Loan_Status      TotalIncome
##         "factor"         "factor"        "numeric"
```

```r
summary(data)
```

```
##     Gender    Married   Dependents         Education   Self_Employed
##        : 12      :  2     : 12     Graduate     :421      : 25
##  Female: 95   No :188   0 :295     Not Graduate:108   No :434
##  Male  :422   Yes:339   1 : 85                        Yes: 70
##                         2 : 92
##                         3+: 45
##
##    LoanAmount     Loan_Amount_Term Credit_History   Property_Area Loan_Status
##  Min.   :  9.0   Min.   : 36.0    0: 79          Rural    :155   N:163
##  1st Qu.:100.0   1st Qu.:360.0    1:450          Semiurban:209   Y:366
##  Median :128.0   Median :360.0                   Urban    :165
##  Mean   :145.9   Mean   :342.4
##  3rd Qu.:167.0   3rd Qu.:360.0
##  Max.   :700.0   Max.   :480.0
##   TotalIncome
##  Min.   : 1442
##  1st Qu.: 4166
##  Median : 5332
##  Mean   : 7050
##  3rd Qu.: 7542
```

```
## Max.   :81000
```

let's give a look at the categorical variables in the dataset:

```r
par(mfrow=c(2,3))

counts <- table(data$Loan_Status, data$Gender)
barplot(counts, main="Loan Status by Gender",
        xlab="Gender", col=c("darkgrey","maroon"),
        legend = rownames(counts))

counts2 <- table(data$Loan_Status, data$Education)
barplot(counts2, main="Loan Status by Education",
        xlab="Education", col=c("darkgrey","maroon"),
        legend = rownames(counts2))

counts3 <- table(data$Loan_Status, data$Married)
barplot(counts3, main="Loan Status by Married",
        xlab="Married", col=c("darkgrey","maroon"),
        legend = rownames(counts3))

counts4 <- table(data$Loan_Status, data$Self_Employed)
barplot(counts4, main="Loan Status by Self Employed",
        xlab="Self_Employed", col=c("darkgrey","maroon"),
        legend = rownames(counts4))

counts5 <- table(data$Loan_Status, data$Property_Area)
barplot(counts5, main="Loan Status by Property_Area",
        xlab="Property_Area", col=c("darkgrey","maroon"),
        legend = rownames(counts5))

counts6 <- table(data$Loan_Status, data$Credit_History)
barplot(counts6, main="Loan Status by Credit_History",
        xlab="Credit_History", col=c("darkgrey","maroon"),
        legend = rownames(counts5))
```

When we look at the Gender graph, we can note that males have more records and more than half of the applicants' applications have been approved. And there are less female applicants but still more than half of their applications have been approved. When We look at the other charts, we can notice the similar situation as the Gender graph.

Decision Trees Part:

A decision tree is a supervised machine learning algorithm that can not only be used for both classification and regression problems, but also can be used to visualize the decision-making process by mapping out different potential outcomes. It create a set of binary splits on the predictor variables in order to create a tree that can be used to classify new observations into one of two groups.

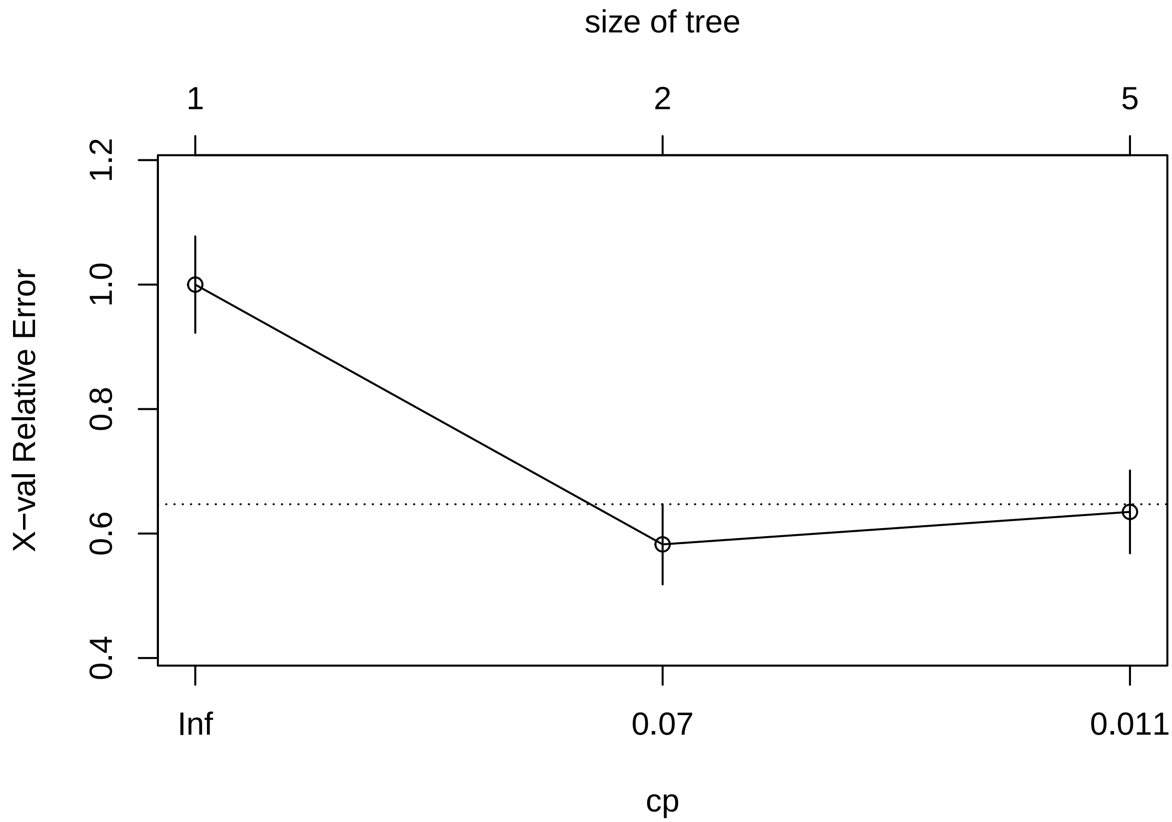
The data is split into training and testing sets 70%/30%.

```
set.seed(622)
sample <- createDataPartition(data$Loan_Status, p = 0.70, list = FALSE, times = 1)
trainnew <- data[sample, ]
testnew  <- data[-sample, ]
```

```
dtree <- rpart(Loan_Status ~  Credit_History+Education+Self_Employed+Property_Area+LoanAmount+TotalInco

dtree$cptable
```

```
##           CP nsplit rel error    xerror       xstd
## 1 0.4173913      0 1.0000000 1.0000000 0.07750794
## 2 0.0115942      1 0.5826087 0.5826087 0.06444927
## 3 0.0100000      4 0.5478261 0.6347826 0.06660820
```

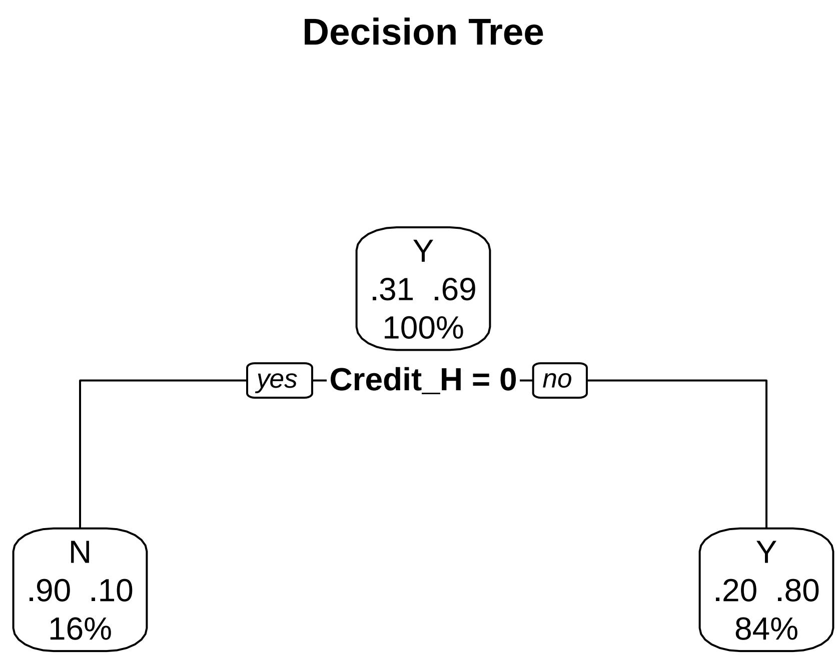
```r
plotcp(dtree)
```




```r
dtree.pruned <- prune(dtree, cp=.02290076)
library(rpart.plot)
```

```
## Warning: package 'rpart.plot' was built under R version 4.0.4
```

```r
prp(dtree.pruned, type = 2, extra = 104,
    fallen.leaves = TRUE, main="Decision Tree")
```




```
dtree.pred_train <- predict(dtree.pruned, trainnew, type="class")
dtree.perf_train <- table(trainnew$Loan_Status, dtree.pred_train,
                    dnn=c("Actual", "Predicted"))
dtree.perf_train
```

```
##        Predicted
## Actual   N   Y
##      N  54  61
##      Y   6 251
```

```
dtree.cm_train <- confusionMatrix(dtree.pred_train, trainnew$Loan_Status)
dtree.cm_train
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   N   Y
##          N  54   6
##          Y  61 251
##
##                Accuracy : 0.8199
##                  95% CI : (0.777, 0.8576)
##     No Information Rate : 0.6909
##     P-Value [Acc > NIR] : 1.131e-08
##
##                   Kappa : 0.5142
##
```

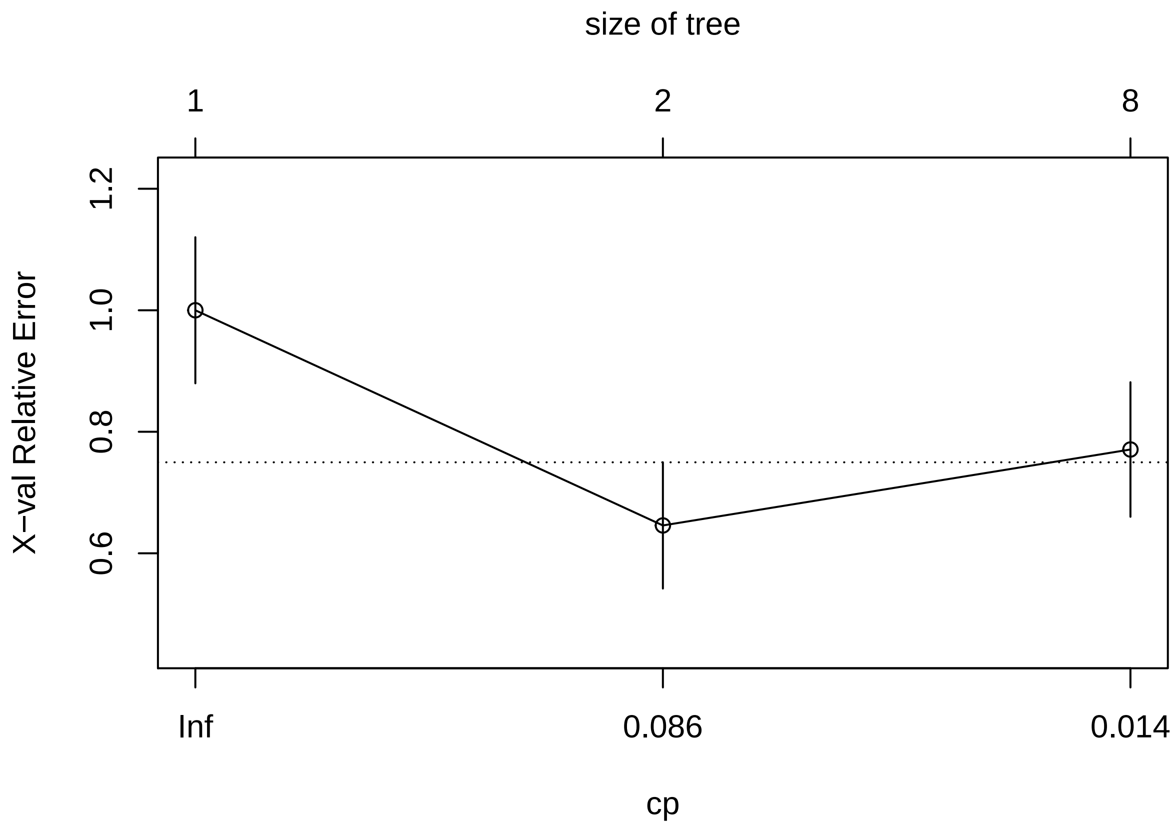
```
##  Mcnemar's Test P-Value : 4.191e-11
##
##             Sensitivity : 0.4696
##             Specificity : 0.9767
##          Pos Pred Value : 0.9000
##          Neg Pred Value : 0.8045
##              Prevalence : 0.3091
##          Detection Rate : 0.1452
##    Detection Prevalence : 0.1613
##       Balanced Accuracy : 0.7231
##
##        'Positive' Class : N
##
```

Use test dataset to analysis

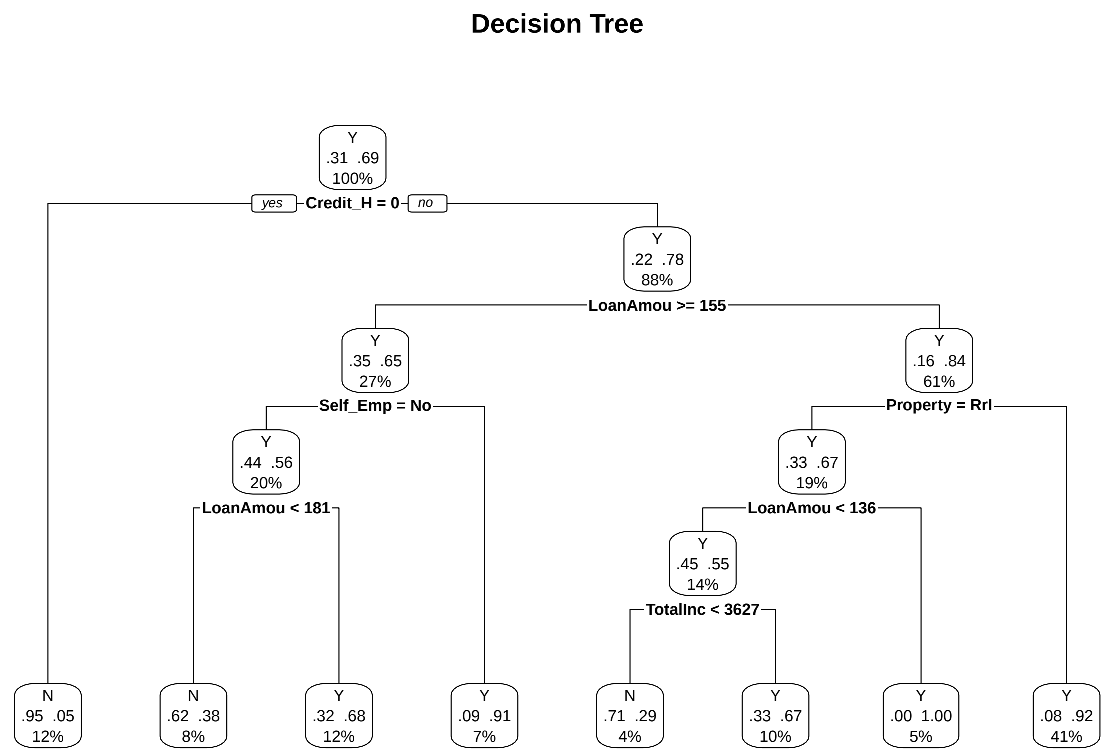
```r
dtree_test <- rpart(Loan_Status ~ Credit_History+Education+Self_Employed+Property_Area+LoanAmount+TotalI

dtree_test$cptable
```

```
##           CP nsplit rel error    xerror      xstd
## 1 0.35416667      0 1.0000000 1.0000000 0.1202660
## 2 0.02083333      1 0.6458333 0.6458333 0.1039142
## 3 0.01000000      7 0.5208333 0.7708333 0.1107900
```

```r
plotcp(dtree_test)
```



```r
dtree_test.pruned <- prune(dtree_test, cp=.01639344)
prp(dtree_test.pruned, type = 2, extra = 104,
    fallen.leaves = TRUE, main="Decision Tree")
```




```r
dtree_test.pred <- predict(dtree_test.pruned, newdata = testnew, type="class")
dtree_test.perf <- table(testnew$Loan_Status, dtree_test.pred,
                         dnn=c("Actual", "Predicted"))
dtree_test.perf
```

```
##        Predicted
## Actual   N   Y
##       N  31  17
##       Y   8 101
```

```r
dtree.cm_test <- confusionMatrix(dtree_test.pred, testnew$Loan_Status)
dtree.cm_test
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   N   Y
##          N  31   8
##          Y  17 101
##
##                Accuracy : 0.8408
##                  95% CI : (0.774, 0.8942)
##     No Information Rate : 0.6943
```

```
##     P-Value [Acc > NIR] : 1.892e-05
##
##                   Kappa : 0.6041
##
##  Mcnemar's Test P-Value : 0.1096
##
##             Sensitivity : 0.6458
##             Specificity : 0.9266
##          Pos Pred Value : 0.7949
##          Neg Pred Value : 0.8559
##              Prevalence : 0.3057
##          Detection Rate : 0.1975
##    Detection Prevalence : 0.2484
##       Balanced Accuracy : 0.7862
##
##        'Positive' Class : N
##
```

Accuracy: Train data: 82% and Test data: 84.08%

Random Trees Part:

Random Forest is a tree-based machine learning algorithm that leverages the power of multiple decision trees for making decisions. This approach develops multiple predictive models, and the results are aggregated to improve classification.

```r
library(randomForest)
```

```
## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':
##
##     combine

## The following object is masked from 'package:ggplot2':
##
##     margin
```

```r
fit.forest <- randomForest(Loan_Status ~ Credit_History+Education+Self_Employed+Property_Area+LoanAmoun

fit.forest
```

```
##
## Call:
##  randomForest(formula = Loan_Status ~ Credit_History + Education +      Self_Employed + Property_Are
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 19.35%
## Confusion matrix:
##    N   Y class.error
```

```
## N 56  59  0.51304348
## Y 13 244  0.05058366
```

```
forest.pred <- predict(fit.forest, newdata = trainnew)
forest.cm <- table(trainnew$Loan_Status, forest.pred,
                   dnn=c("Actual", "Predicted"))
forest.cm
```

```
##       Predicted
## Actual   N   Y
##      N  82  33
##      Y   6 251
```

```
forest.cm_train <- confusionMatrix(forest.pred, trainnew$Loan_Status)
forest.cm_train
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   N   Y
##          N  82   6
##          Y  33 251
##
##                Accuracy : 0.8952
##                  95% CI : (0.8595, 0.9244)
##     No Information Rate : 0.6909
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7375
##
##  Mcnemar's Test P-Value : 3.136e-05
##
##             Sensitivity : 0.7130
##             Specificity : 0.9767
##          Pos Pred Value : 0.9318
##          Neg Pred Value : 0.8838
##              Prevalence : 0.3091
##          Detection Rate : 0.2204
##    Detection Prevalence : 0.2366
##       Balanced Accuracy : 0.8448
##
##        'Positive' Class : N
##
```

Use test dataset to analysis

```
fit.forest_test <- randomForest(Loan_Status ~ Credit_History+Education+Self_Employed+Property_Area+Loan

fit.forest_test
```

```
##
## Call:
##  randomForest(formula = Loan_Status ~ Credit_History + Education +      Self_Employed + Property_Are
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
```

```
##         OOB estimate of  error rate: 22.93%
## Confusion matrix:
##    N   Y class.error
## N 19  29  0.60416667
## Y  7 102  0.06422018
```

```r
forest.pred_test <- predict(fit.forest_test, newdata = testnew)
forest.cm_test <- table(testnew$Loan_Status, forest.pred_test,
                     dnn=c("Actual", "Predicted"))
forest.cm_test
```

```
##       Predicted
## Actual   N   Y
##      N  40   8
##      Y   0 109
```

```r
forest.cm_test <- confusionMatrix(forest.pred_test, testnew$Loan_Status)
forest.cm_test
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   N   Y
##          N  40   0
##          Y   8 109
##
##                Accuracy : 0.949
##                  95% CI : (0.9021, 0.9777)
##     No Information Rate : 0.6943
##     P-Value [Acc > NIR] : 1.615e-15
##
##                   Kappa : 0.8741
##
##  Mcnemar's Test P-Value : 0.01333
##
##             Sensitivity : 0.8333
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.9316
##              Prevalence : 0.3057
##          Detection Rate : 0.2548
##    Detection Prevalence : 0.2548
##       Balanced Accuracy : 0.9167
##
##        'Positive' Class : N
##
```

Here, we notice slight improvements on both samples where accuracy for the training sample is 89.52% and the accuracy for the test sample is 94.90%.