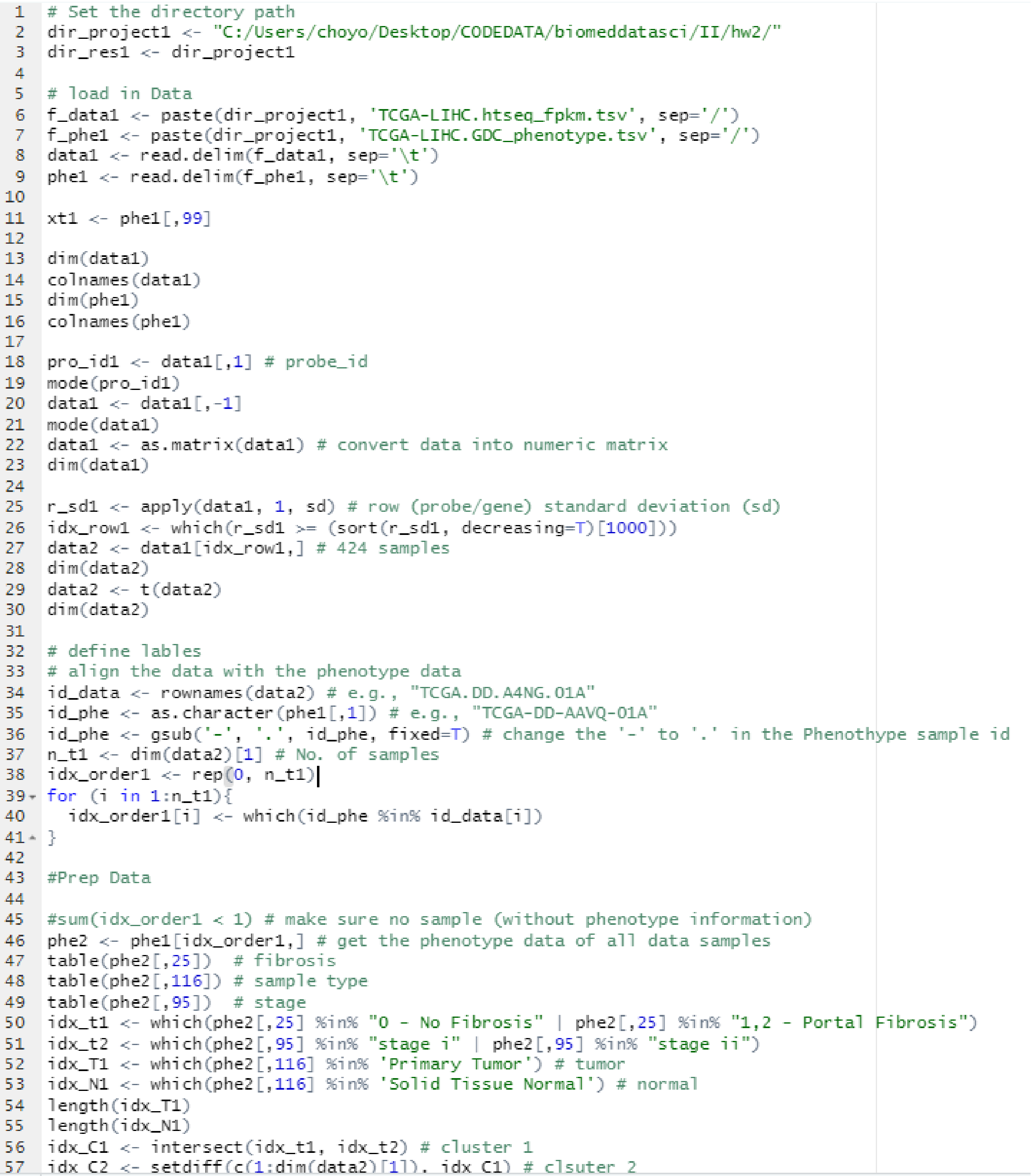
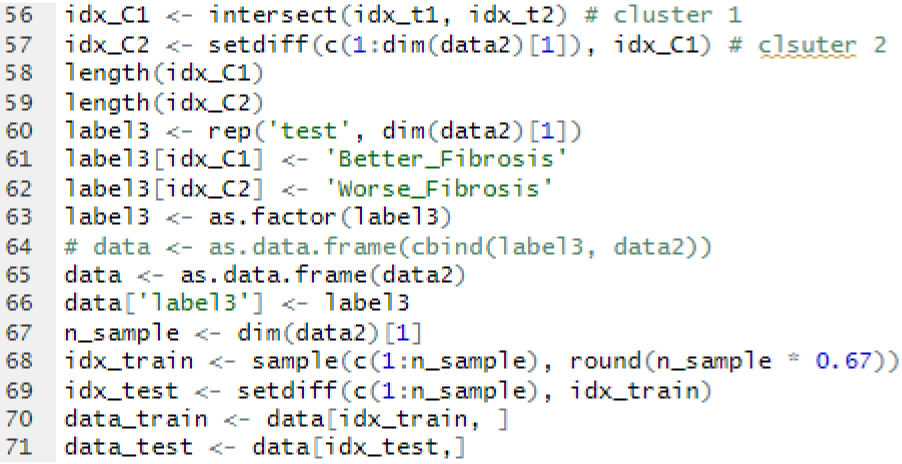
Preparation

Data:

1) TCGA-LIHC.htseq_fpkm.tsv
2) TCGA-LIHC.GDC_phenotype.tsv

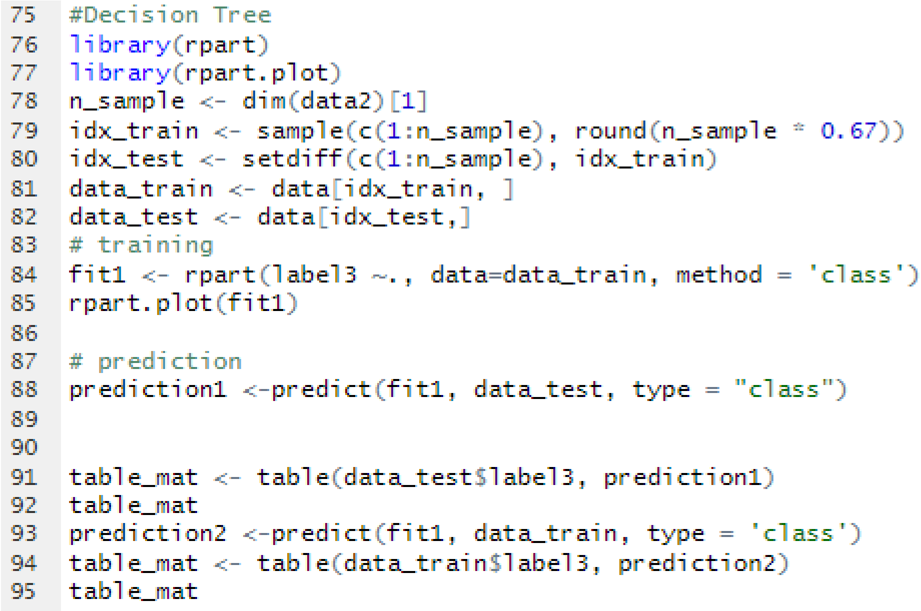


```r
# Set the directory path
dir_project1 <- "C:/Users/choyo/Desktop/CODEDATA/biomeddatasci/II/hw2/"
dir_res1 <- dir_project1

# load in Data
f_data1 <- paste(dir_project1, 'TCGA-LIHC.htseq_fpkm.tsv', sep='/')
f_phe1 <- paste(dir_project1, 'TCGA-LIHC.GDC_phenotype.tsv', sep='/')
data1 <- read.delim(f_data1, sep='\t')
phe1 <- read.delim(f_phe1, sep='\t')

xt1 <- phe1[,99]

dim(data1)
colnames(data1)
dim(phe1)
colnames(phe1)

pro_id1 <- data1[,1] # probe_id
mode(pro_id1)
data1 <- data1[,-1]
mode(data1)
data1 <- as.matrix(data1) # convert data into numeric matrix
dim(data1)

r_sd1 <- apply(data1, 1, sd) # row (probe/gene) standard deviation (sd)
idx_row1 <- which(r_sd1 >= (sort(r_sd1, decreasing=T)[1000]))
data2 <- data1[idx_row1,] # 424 samples
dim(data2)
data2 <- t(data2)
dim(data2)

# define lables
# align the data with the phenotype data
id_data <- rownames(data2) # e.g., "TCGA.DD.A4NG.01A"
id_phe <- as.character(phe1[,1]) # e.g., "TCGA-DD-AAVQ-01A"
id_phe <- gsub('-', '.', id_phe, fixed=T) # change the '-' to '.' in the Phenothype sample id
n_t1 <- dim(data2)[1] # No. of samples
idx_order1 <- rep(0, n_t1)
for (i in 1:n_t1){
  idx_order1[i] <- which(id_phe %in% id_data[i])
}

#Prep Data

#sum(idx_order1 < 1) # make sure no sample (without phenotype information)
phe2 <- phe1[idx_order1,] # get the phenotype data of all data samples
table(phe2[,25])  # fibrosis
table(phe2[,116]) # sample type
table(phe2[,95])  # stage
idx_t1 <- which(phe2[,25] %in% "0 - No Fibrosis" | phe2[,25] %in% "1,2 - Portal Fibrosis")
idx_t2 <- which(phe2[,95] %in% "stage i" | phe2[,95] %in% "stage ii")
idx_T1 <- which(phe2[,116] %in% 'Primary Tumor') # tumor
idx_N1 <- which(phe2[,116] %in% 'Solid Tissue Normal') # normal
length(idx_T1)
length(idx_N1)
idx_C1 <- intersect(idx_t1, idx_t2) # cluster 1
idx_C2 <- setdiff(c(1:dim(data2)[1]), idx_C1) # clsuter 2
```

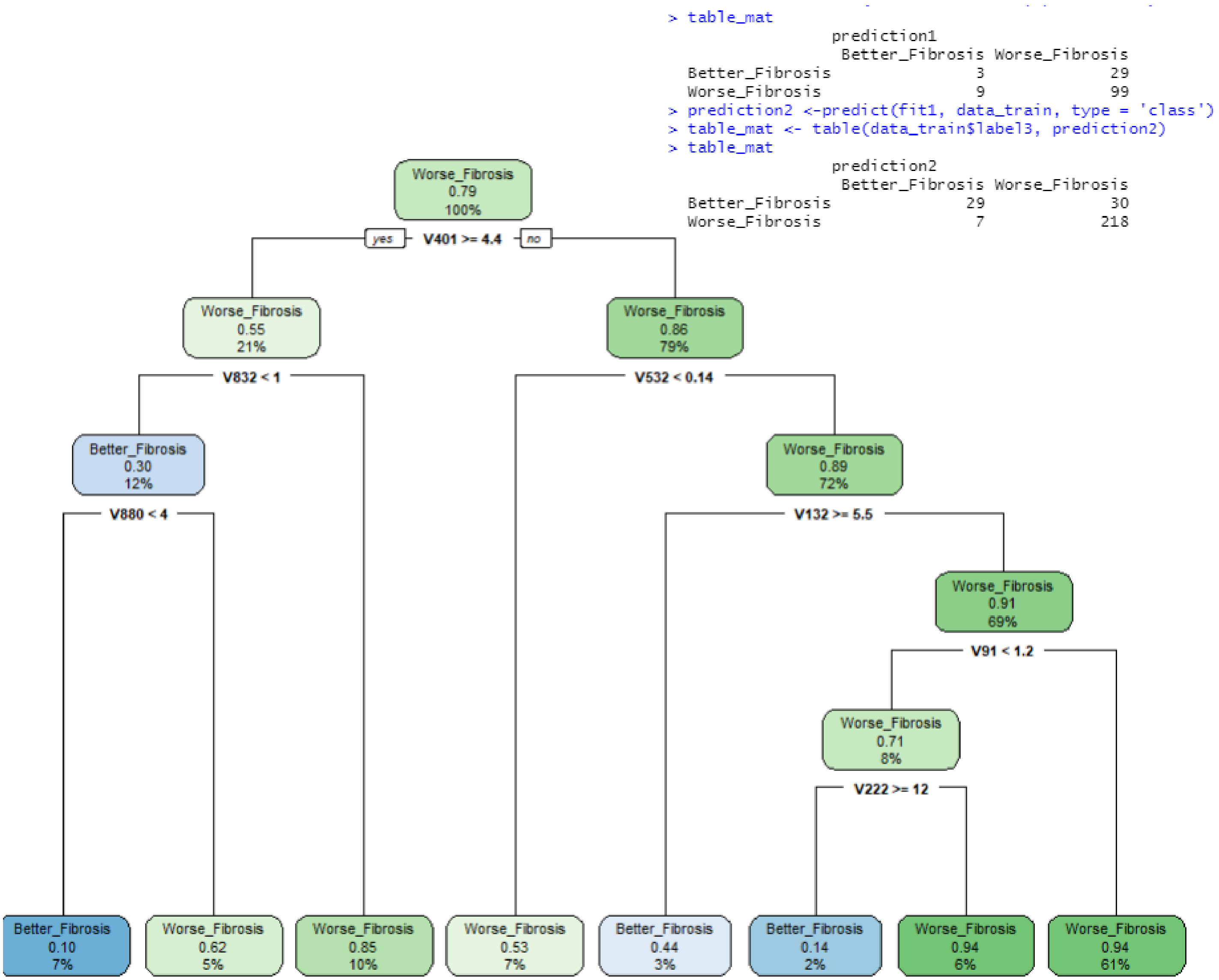


```r
idx_C1 <- intersect(idx_t1, idx_t2) # cluster 1
idx_C2 <- setdiff(c(1:dim(data2)[1]), idx_C1) # clsuter 2
length(idx_C1)
length(idx_C2)
label3 <- rep('test', dim(data2)[1])
label3[idx_C1] <- 'Better_Fibrosis'
label3[idx_C2] <- 'Worse_Fibrosis'
label3 <- as.factor(label3)
# data <- as.data.frame(cbind(label3, data2))
data <- as.data.frame(data2)
data['label3'] <- label3
n_sample <- dim(data2)[1]
idx_train <- sample(c(1:n_sample), round(n_sample * 0.67))
idx_test <- setdiff(c(1:n_sample), idx_train)
data_train <- data[idx_train, ]
data_test <- data[idx_test,]
```

Decision Tree

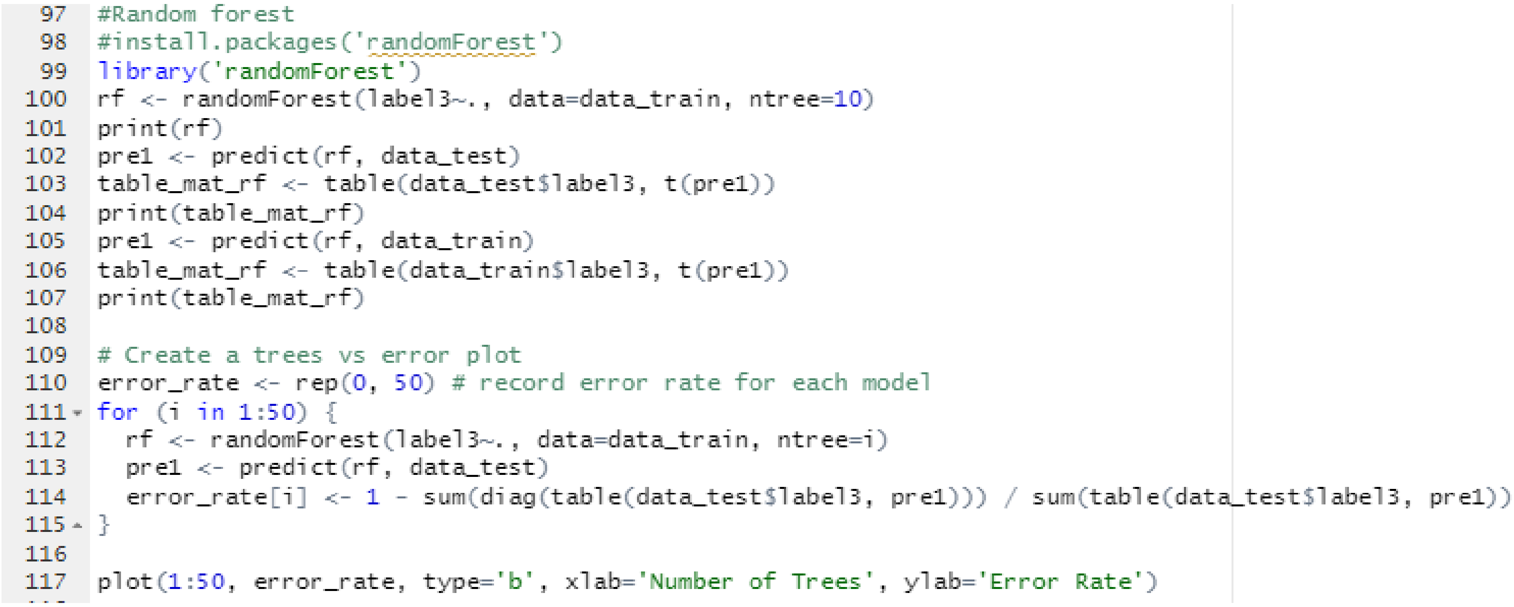


```r
#Decision Tree
library(rpart)
library(rpart.plot)
n_sample <- dim(data2)[1]
idx_train <- sample(c(1:n_sample), round(n_sample * 0.67))
idx_test <- setdiff(c(1:n_sample), idx_train)
data_train <- data[idx_train, ]
data_test <- data[idx_test,]
# training
fit1 <- rpart(label3 ~., data=data_train, method = 'class')
rpart.plot(fit1)

# prediction
prediction1 <-predict(fit1, data_test, type = "class")


table_mat <- table(data_test$label3, prediction1)
table_mat
prediction2 <-predict(fit1, data_train, type = 'class')
table_mat <- table(data_train$label3, prediction2)
table_mat
```

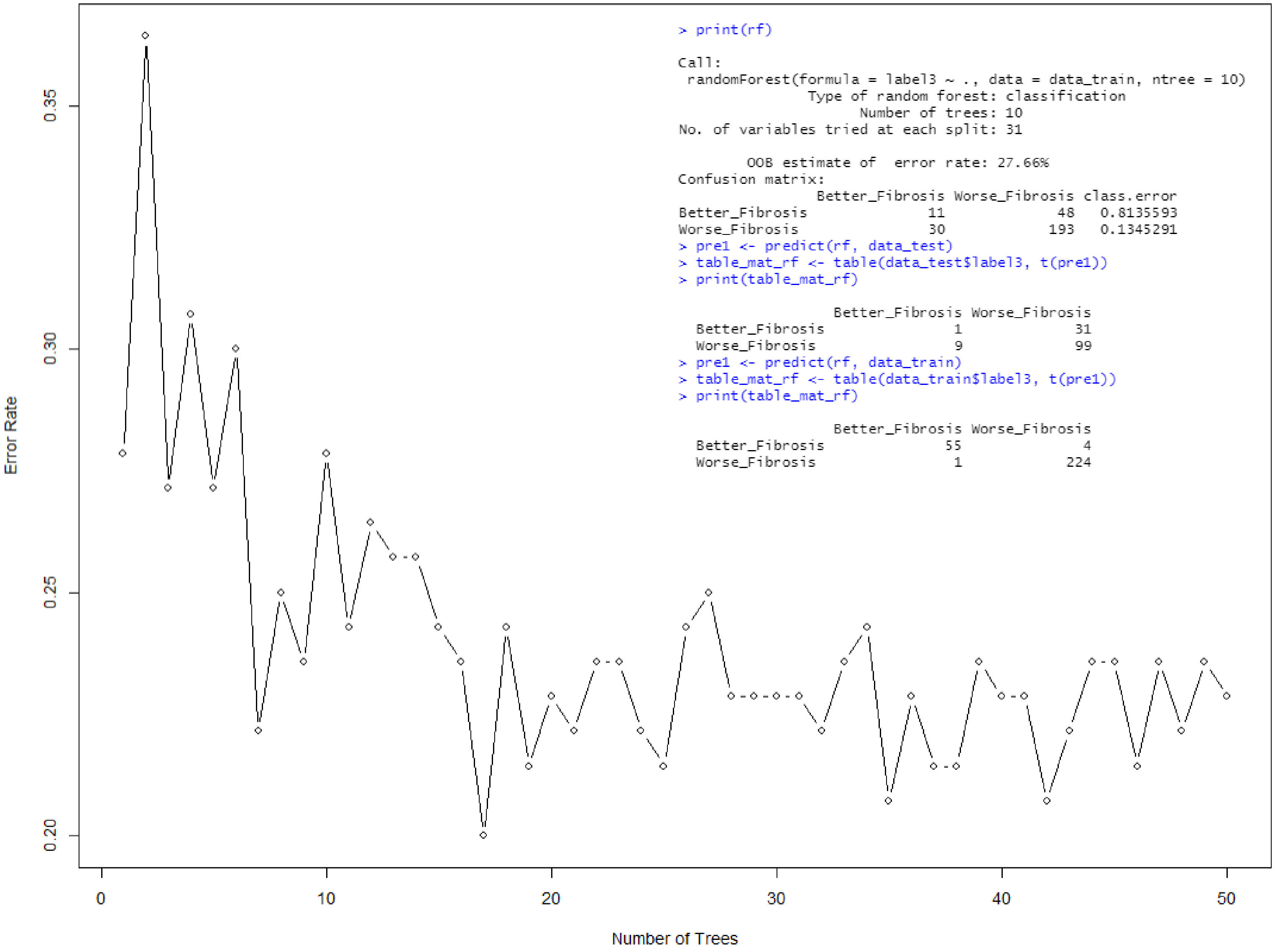


```
> table_mat
                 prediction1
                  Better_Fibrosis Worse_Fibrosis
  Better_Fibrosis               3             29
  Worse_Fibrosis                9             99
> prediction2 <-predict(fit1, data_train, type = 'class')
> table_mat <- table(data_train$label3, prediction2)
> table_mat
                 prediction2
                  Better_Fibrosis Worse_Fibrosis
  Better_Fibrosis              29             30
  Worse_Fibrosis                7            218
```

Random Forest

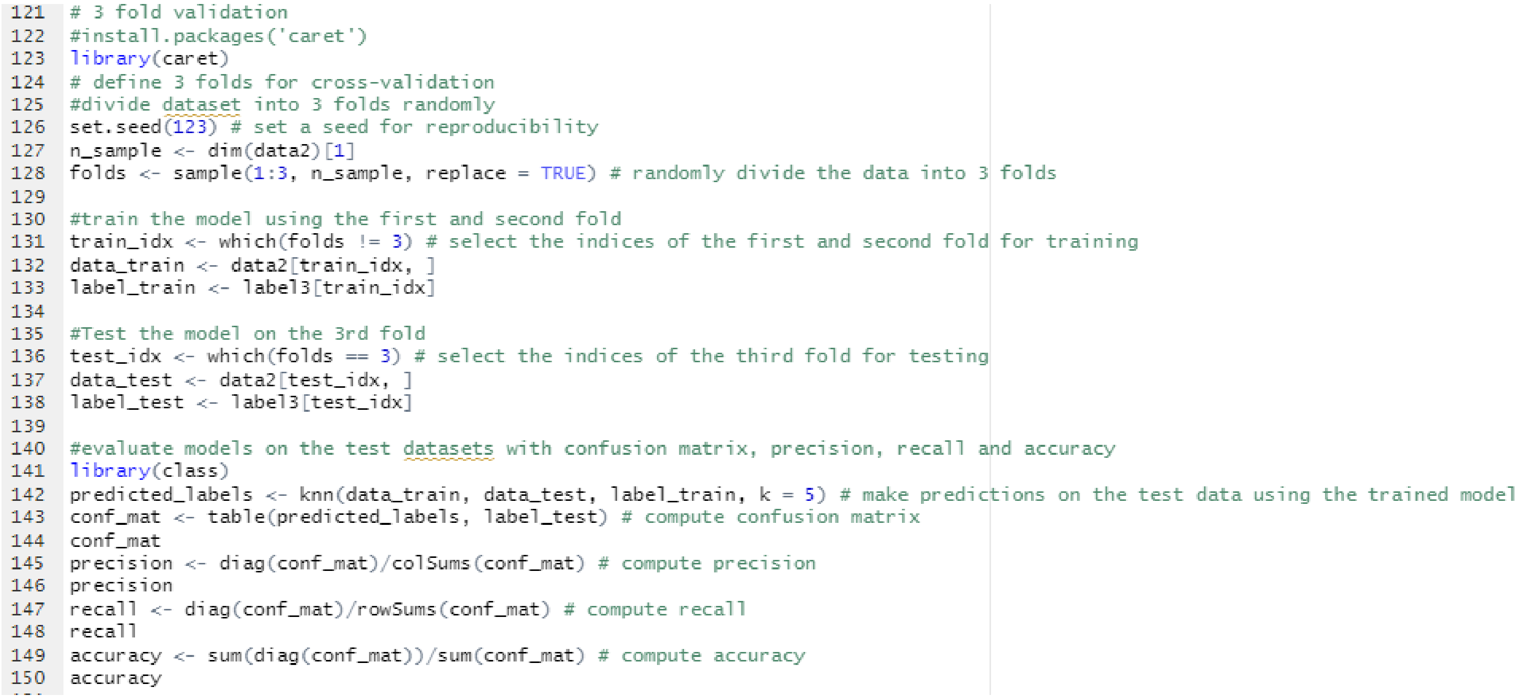


```r
#Random forest
#install.packages('randomForest')
library('randomForest')
rf <- randomForest(label3~., data=data_train, ntree=10)
print(rf)
pre1 <- predict(rf, data_test)
table_mat_rf <- table(data_test$label3, t(pre1))
print(table_mat_rf)
pre1 <- predict(rf, data_train)
table_mat_rf <- table(data_train$label3, t(pre1))
print(table_mat_rf)

# Create a trees vs error plot
error_rate <- rep(0, 50) # record error rate for each model
for (i in 1:50) {
  rf <- randomForest(label3~., data=data_train, ntree=i)
  pre1 <- predict(rf, data_test)
  error_rate[i] <- 1 - sum(diag(table(data_test$label3, pre1))) / sum(table(data_test$label3, pre1))
}

plot(1:50, error_rate, type='b', xlab='Number of Trees', ylab='Error Rate')
```

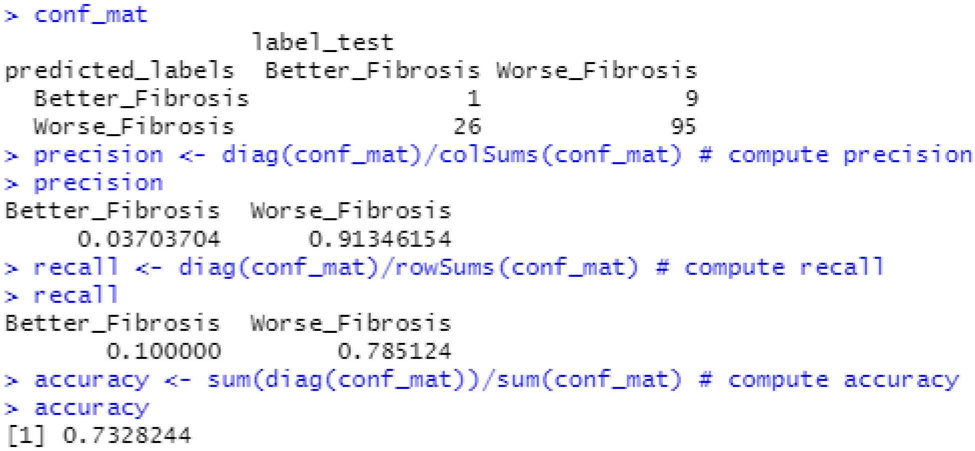


```
> print(rf)

Call:
 randomForest(formula = label3 ~ ., data = data_train, ntree = 10)
               Type of random forest: classification
                     Number of trees: 10
No. of variables tried at each split: 31

        OOB estimate of  error rate: 27.66%
Confusion matrix:
                Better_Fibrosis Worse_Fibrosis class.error
Better_Fibrosis              11             48   0.8135593
Worse_Fibrosis               30            193   0.1345291
> pre1 <- predict(rf, data_test)
> table_mat_rf <- table(data_test$label3, t(pre1))
> print(table_mat_rf)

                  Better_Fibrosis Worse_Fibrosis
  Better_Fibrosis               1             31
  Worse_Fibrosis                9             99
> pre1 <- predict(rf, data_train)
> table_mat_rf <- table(data_train$label3, t(pre1))
> print(table_mat_rf)

                  Better_Fibrosis Worse_Fibrosis
  Better_Fibrosis              55              4
  Worse_Fibrosis                1            224
```

3-cross folds validation



```r
# 3 fold validation
#install.packages('caret')
library(caret)
# define 3 folds for cross-validation
#divide dataset into 3 folds randomly
set.seed(123) # set a seed for reproducibility
n_sample <- dim(data2)[1]
folds <- sample(1:3, n_sample, replace = TRUE) # randomly divide the data into 3 folds

#train the model using the first and second fold
train_idx <- which(folds != 3) # select the indices of the first and second fold for training
data_train <- data2[train_idx, ]
label_train <- label3[train_idx]

#Test the model on the 3rd fold
test_idx <- which(folds == 3) # select the indices of the third fold for testing
data_test <- data2[test_idx, ]
label_test <- label3[test_idx]

#evaluate models on the test datasets with confusion matrix, precision, recall and accuracy
library(class)
predicted_labels <- knn(data_train, data_test, label_train, k = 5) # make predictions on the test data using the trained model
conf_mat <- table(predicted_labels, label_test) # compute confusion matrix
conf_mat
precision <- diag(conf_mat)/colSums(conf_mat) # compute precision
precision
recall <- diag(conf_mat)/rowSums(conf_mat) # compute recall
recall
accuracy <- sum(diag(conf_mat))/sum(conf_mat) # compute accuracy
accuracy
```



```
> conf_mat
                 label_test
predicted_labels  Better_Fibrosis Worse_Fibrosis
  Better_Fibrosis               1              9
  Worse_Fibrosis               26             95
> precision <- diag(conf_mat)/colSums(conf_mat) # compute precision
> precision
Better_Fibrosis  Worse_Fibrosis 
     0.03703704      0.91346154 
> recall <- diag(conf_mat)/rowSums(conf_mat) # compute recall
> recall
Better_Fibrosis  Worse_Fibrosis 
       0.100000        0.785124 
> accuracy <- sum(diag(conf_mat))/sum(conf_mat) # compute accuracy
> accuracy
[1] 0.7328244
```