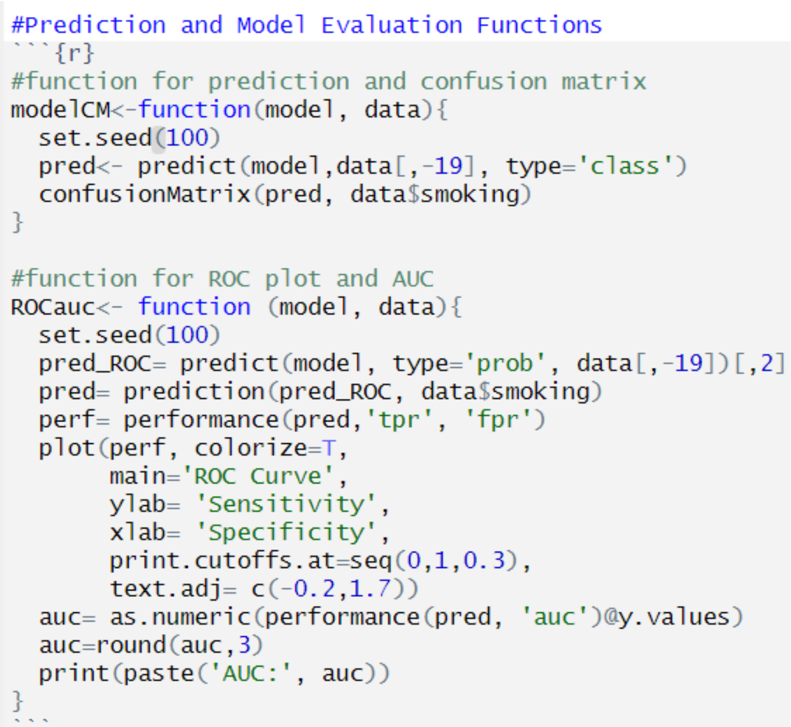
Model Implementation and Validation

Three tree-based machine learning methods Decision Tree, Random Forest, Extreme Gradient Boosting, will be applied on this dataset. Each model will be applied on training and test sets, relevant results will be compared on accuracy, area under the ROC curve, specificity and sensitivity.

Note that for both Decision Tree and Random Forest model, since both methods will be using the same dataset, two functions were created ‘modelCM’ and ‘ROCauc’, shown as follows. modelCM function is for prediction and generate confusion matrix. ROCaucto function is for plotting ROC and computing AUC for model evaluation purposes. These two functions will be called after for prediction and model evaluation.

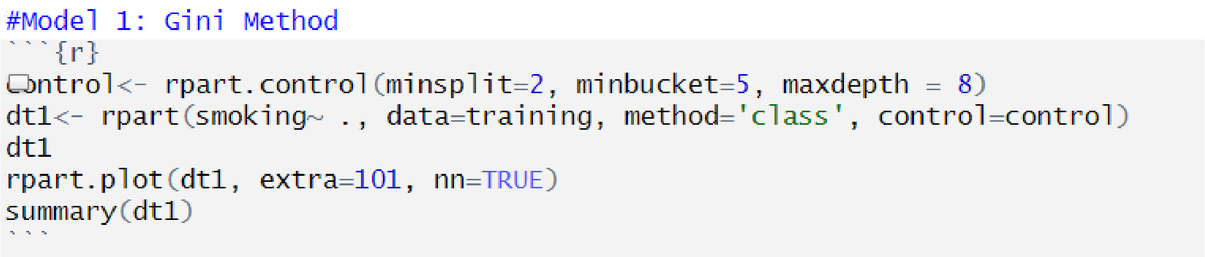


~~~
#Prediction and Model Evaluation Functions
```{r}
#function for prediction and confusion matrix
modelCM<-function(model, data){
  set.seed(100)
  pred<- predict(model,data[,-19], type='class')
  confusionMatrix(pred, data$smoking)
}

#function for ROC plot and AUC
ROCauc<- function (model, data){
  set.seed(100)
  pred_ROC= predict(model, type='prob', data[,-19])[,2]
  pred= prediction(pred_ROC, data$smoking)
  perf= performance(pred,'tpr', 'fpr')
  plot(perf, colorize=T,
       main='ROC Curve',
       ylab= 'Sensitivity',
       xlab= 'Specificity',
       print.cutoffs.at=seq(0,1,0.3),
       text.adj= c(-0.2,1.7))
  auc= as.numeric(performance(pred, 'auc')@y.values)
  auc=round(auc,3)
  print(paste('AUC:', auc))
}
```
~~~

6.1 Decision Tree

6.1.1 Model 1

Build Base Model



~~~
#Model 1: Gini Method
```{r}
control<- rpart.control(minsplit=2, minbucket=5, maxdepth = 8)
dt1<- rpart(smoking~ ., data=training, method='class', control=control)
dt1
rpart.plot(dt1, extra=101, nn=TRUE)
summary(dt1)
```
~~~

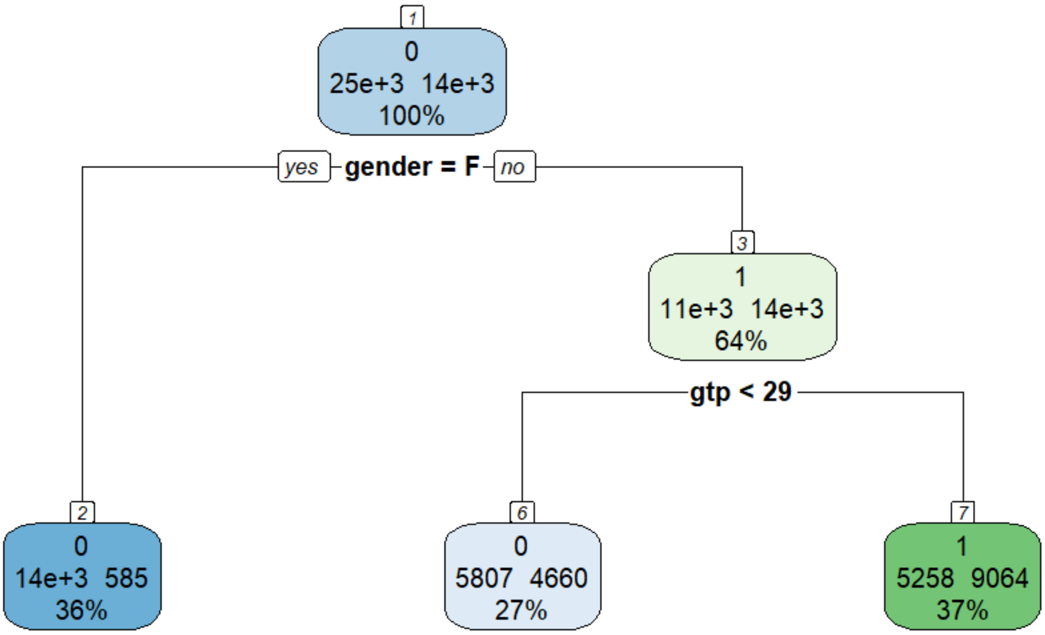
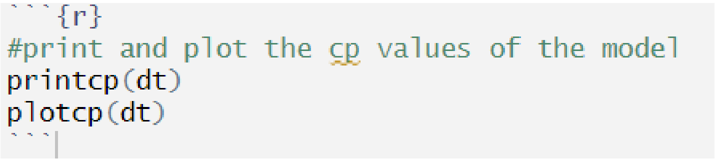
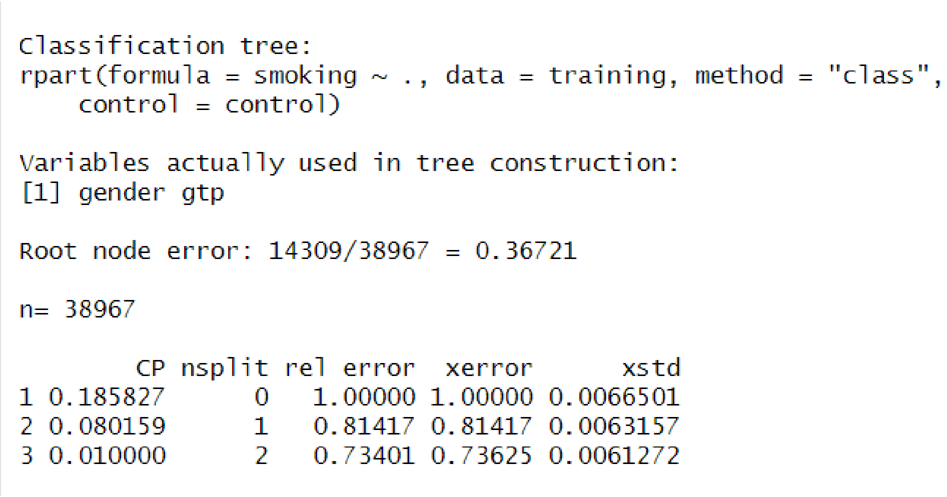


Figure 1 Decision Tree using Gini Split Method

The basic decision tree model uses setting minsplit=2, minbucket=5, maxdepth=8, Gini index splitting method, and all variables were included in the model. Gender and GTP were used to construct the tree. The first node was split on gender feature followed by GTP with condition less than 29.



~~~
```{r}
#print and plot the cp values of the model
printcp(dt)
plotcp(dt)
```
~~~



```
Classification tree:
rpart(formula = smoking ~ ., data = training, method = "class",
    control = control)

Variables actually used in tree construction:
[1] gender gtp

Root node error: 14309/38967 = 0.36721

n= 38967

        CP nsplit rel error  xerror      xstd
1 0.185827      0   1.00000 1.00000 0.0066501
2 0.080159      1   0.81417 0.81417 0.0063157
3 0.010000      2   0.73401 0.73625 0.0061272
```

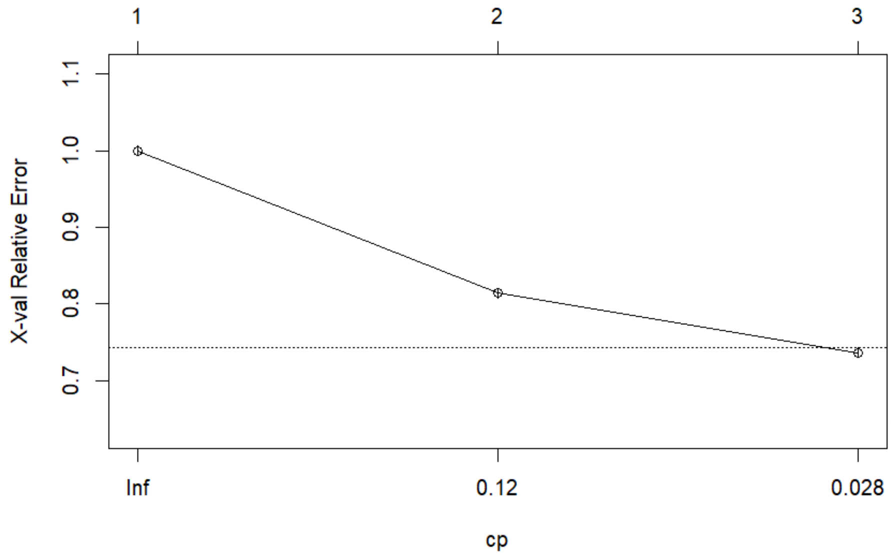
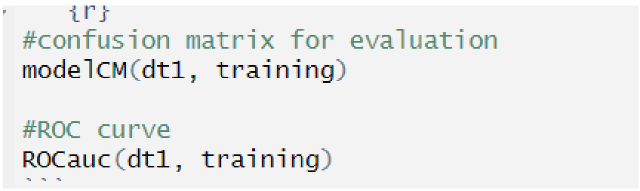


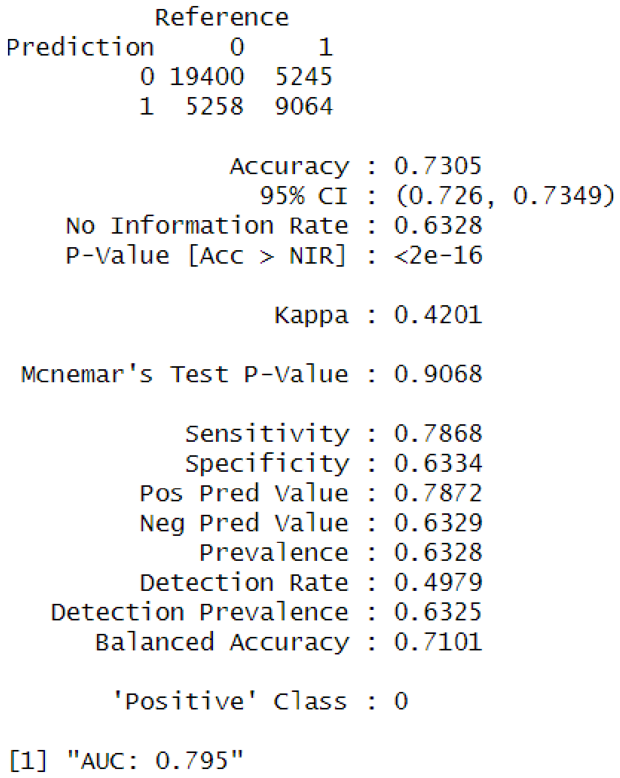
Figure 2 Complexity Parameter vs X-val Relative Error

Complexity Parameter (CP) is used to control the decision tree size, it helps to select an optimal tree size. The general idea is that tree building process stops if adding another variable to the current node does not decrease the error rate further. The goal is to choose for cp value with the smallest cross validation error (xerror). From the X-val Relative Error vs. cp plot, cp 0.028 has the lowest error rate. However, it is not known whether cp value beyond this can achieve better results.

<u>Predict and Evaluation on Training Set</u>



~~~
```{r}
#confusion matrix for evaluation
modelCM(dt1, training)

#ROC curve
ROCauc(dt1, training)
```
~~~



```
          Reference
Prediction     0     1
         0 19400  5245
         1  5258  9064

               Accuracy : 0.7305
                 95% CI : (0.726, 0.7349)
    No Information Rate : 0.6328
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.4201

 Mcnemar's Test P-Value : 0.9068

            Sensitivity : 0.7868
            Specificity : 0.6334
         Pos Pred Value : 0.7872
         Neg Pred Value : 0.6329
             Prevalence : 0.6328
         Detection Rate : 0.4979
   Detection Prevalence : 0.6325
      Balanced Accuracy : 0.7101

       'Positive' Class : 0

[1] "AUC: 0.795"
```

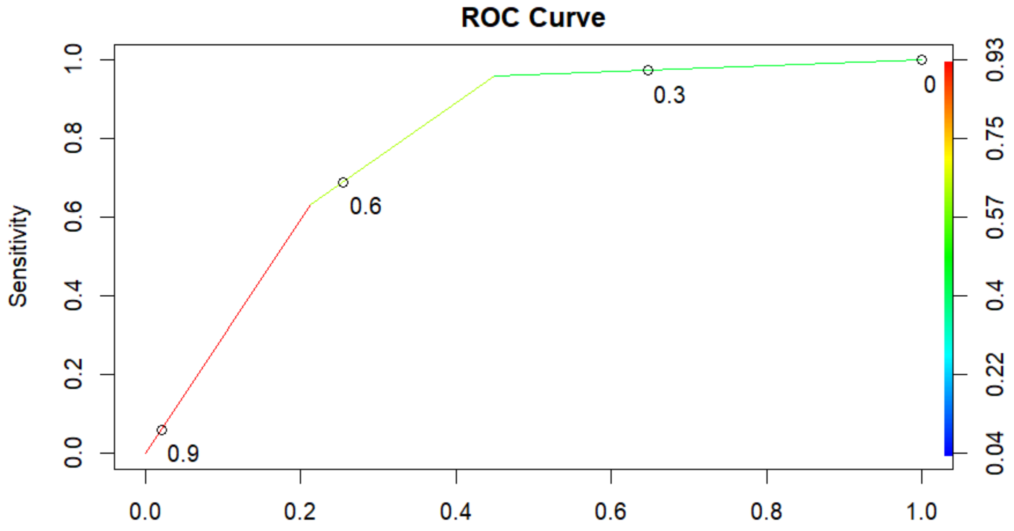
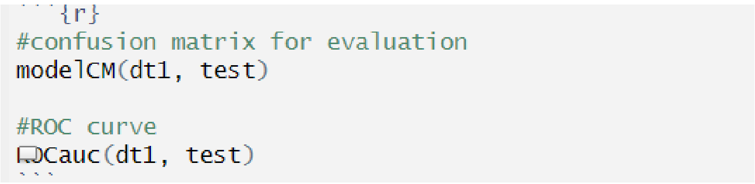


Figure 3 ROC of Decision Tree Model 1 on Training Data

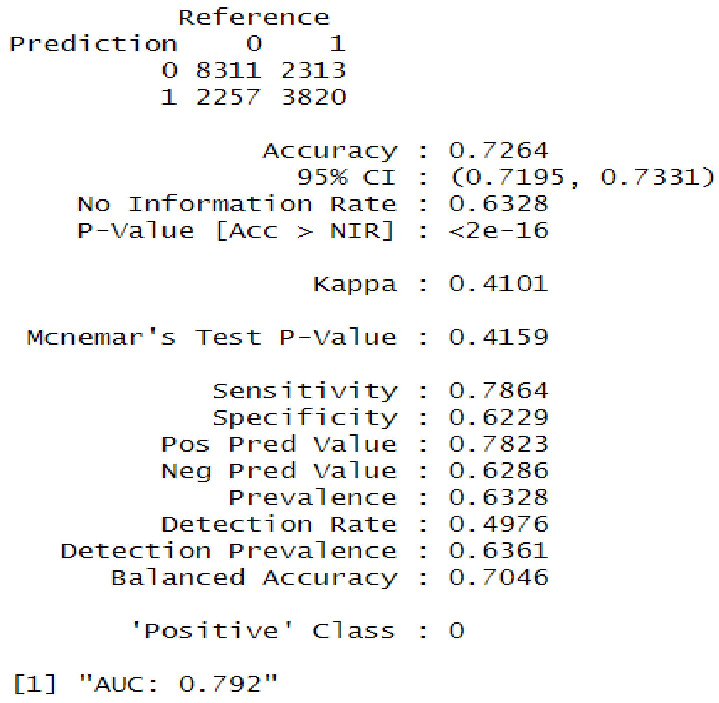
Prediction using Decision Tree model 1 on training data yield 73.05% accuracy, 78.68% sensitivity, 63.34% specificity. AUC is 0.795, which means that there is 79.5% chance that the model can distinguish non-smoking and smoking cases. From figure 15, the ROC curve is away from the 45 degree diagonal line. When the threshold is set to 0.6, sensitivity is around 0.65 while specificity is close to 0.25, which means that the True Positive is higher than the False Positive Rate when the threshold is set at 0.6.

From the confusion matrix output, it was observed that out of the total number of observations, the model correctly classifies 28464 cases which gives an accuracy of 73.05%. Specifically, out of all non-smoking cases, 19400 were correctly classified, achieving 78.68% sensitivity. On the other hand, out of all smoking cases, 9064 cases were correctly classified, achieving 63.34% specificity. Finally, 5258 non-smoking cases and 5245 smoking cases were misclassified.

Predict and Evaluation on Test Set



~~~
```{r}
#confusion matrix for evaluation
modelCM(dt1, test)

#ROC curve
ROCauc(dt1, test)
```
~~~



```
          Reference
Prediction    0    1
         0 8311 2313
         1 2257 3820
                                          
               Accuracy : 0.7264          
                 95% CI : (0.7195, 0.7331)
    No Information Rate : 0.6328          
    P-Value [Acc > NIR] : <2e-16          
                                          
                  Kappa : 0.4101          
                                          
 Mcnemar's Test P-Value : 0.4159          
                                          
            Sensitivity : 0.7864          
            Specificity : 0.6229          
         Pos Pred Value : 0.7823          
         Neg Pred Value : 0.6286          
             Prevalence : 0.6328          
         Detection Rate : 0.4976          
   Detection Prevalence : 0.6361          
      Balanced Accuracy : 0.7046          
                                          
       'Positive' Class : 0               
                                          
[1] "AUC: 0.792"
```

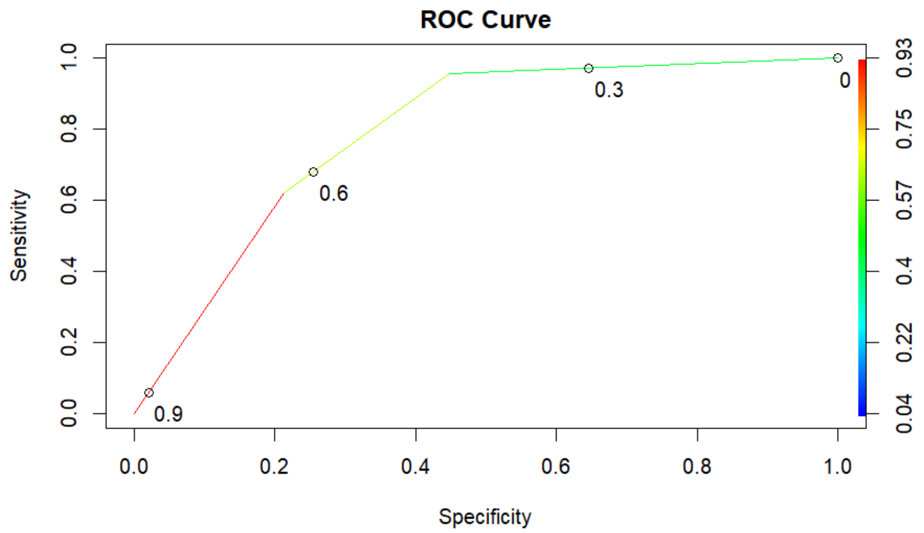
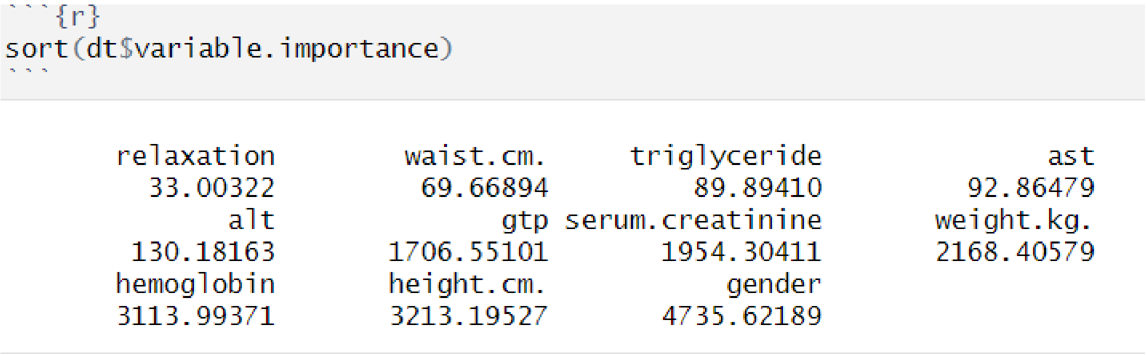


Figure 4 ROC of Decision Tree Model 1 on Test Data

Prediction using Decision Tree model 1 on test data yield 72.64% accuracy, 78.64% sensitivity, 62.29% specificity. The AUC is 0.792, which means that there is 79.2% chance that the model can distinguish non-smoking and smoking cases. It is also observed that the ROC curve is away from the 45 degree diagonal line. From the confusion matrix output, it was observed that out of the total number of observations, the model correctly classifies 12131 cases which gives an accuracy of 72.64%. Specifically, out of all non-smoking cases, 8311 were correctly classified, achieving 78.64% sensitivity. On the other hand, out of all smoking cases, 2313 cases were correctly classified, achieving 62.29% specificity. Finally, 2257 non-smoking cases and 2313 smoking cases were misclassified.

In comparison to the model performance on both training and test data, applying the model on test data has a slightly lower performance. The slight difference between the two suggests that there is no overfitting.

<u>Variable importance</u>



~~~
```{r}
sort(dt$variable.importance)
```
~~~

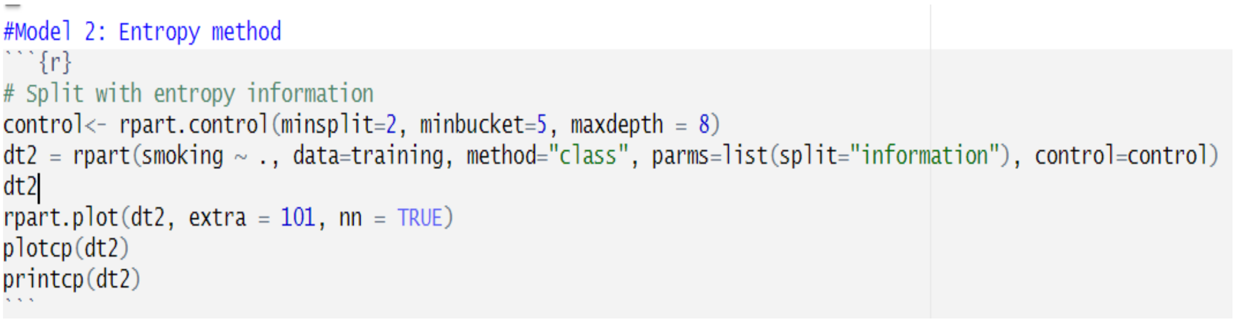
```
       relaxation          waist.cm.      triglyceride               ast
         33.00322           69.66894          89.89410          92.86479
              alt                gtp  serum.creatinine         weight.kg.
        130.18163         1706.55101        1954.30411        2168.40579
       hemoglobin          height.cm.            gender
       3113.99371         3213.19527        4735.62189
```

In this model which uses Gini index as splitting method, gender is the most important variable, followed by height and hemoglobin.

6.1.2 Model 2

Build Model

Using the same setting as model 1, the only changes is the split method, entropy information. Similarly, all variables were included in the model 2.



~~~
#Model 2: Entropy method
```{r}
# Split with entropy information
control<- rpart.control(minsplit=2, minbucket=5, maxdepth = 8)
dt2 = rpart(smoking ~ ., data=training, method="class", parms=list(split="information"), control=control)
dt2
rpart.plot(dt2, extra = 101, nn = TRUE)
plotcp(dt2)
printcp(dt2)
```
~~~

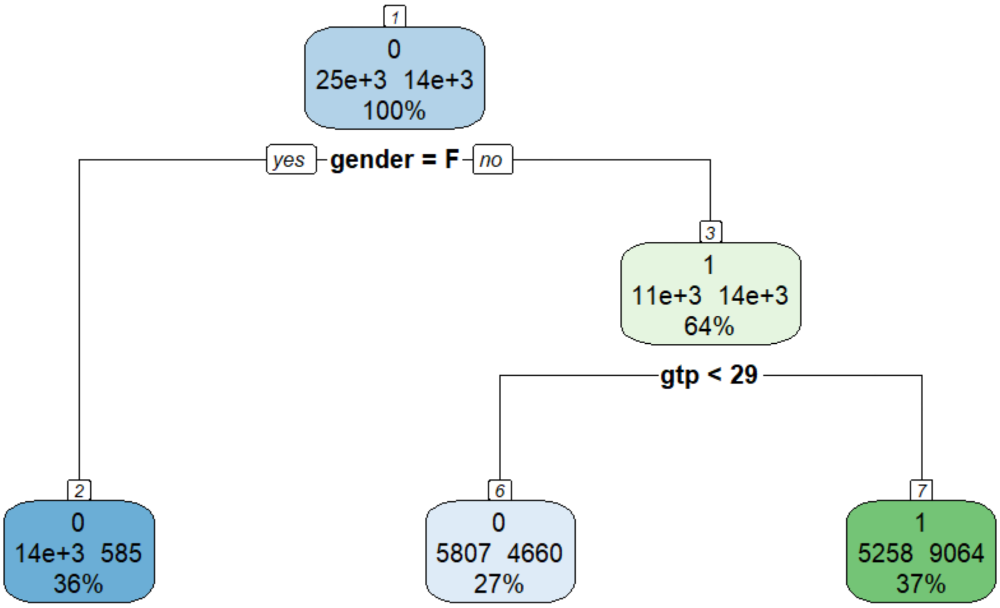


Figure 5 Decision Tree Structure of Model 2 using Entropy Information Split Method

Similar to Model 1, the decision tree diagram shows that gender and GTP were used to construct the tree. It started on the gender feature then GTP with condition less than 29 or otherwise.

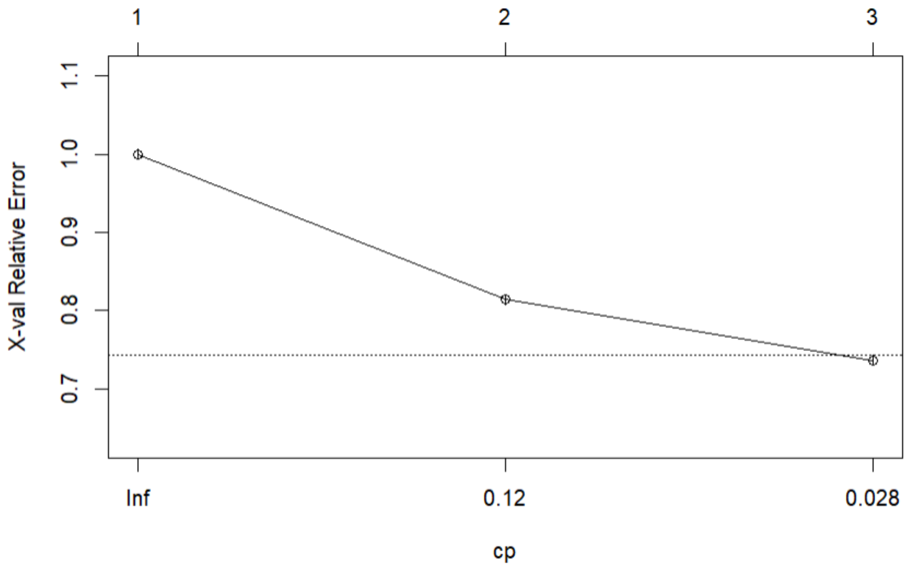
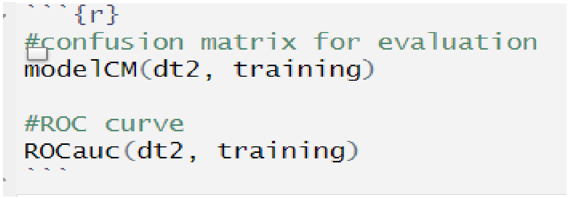


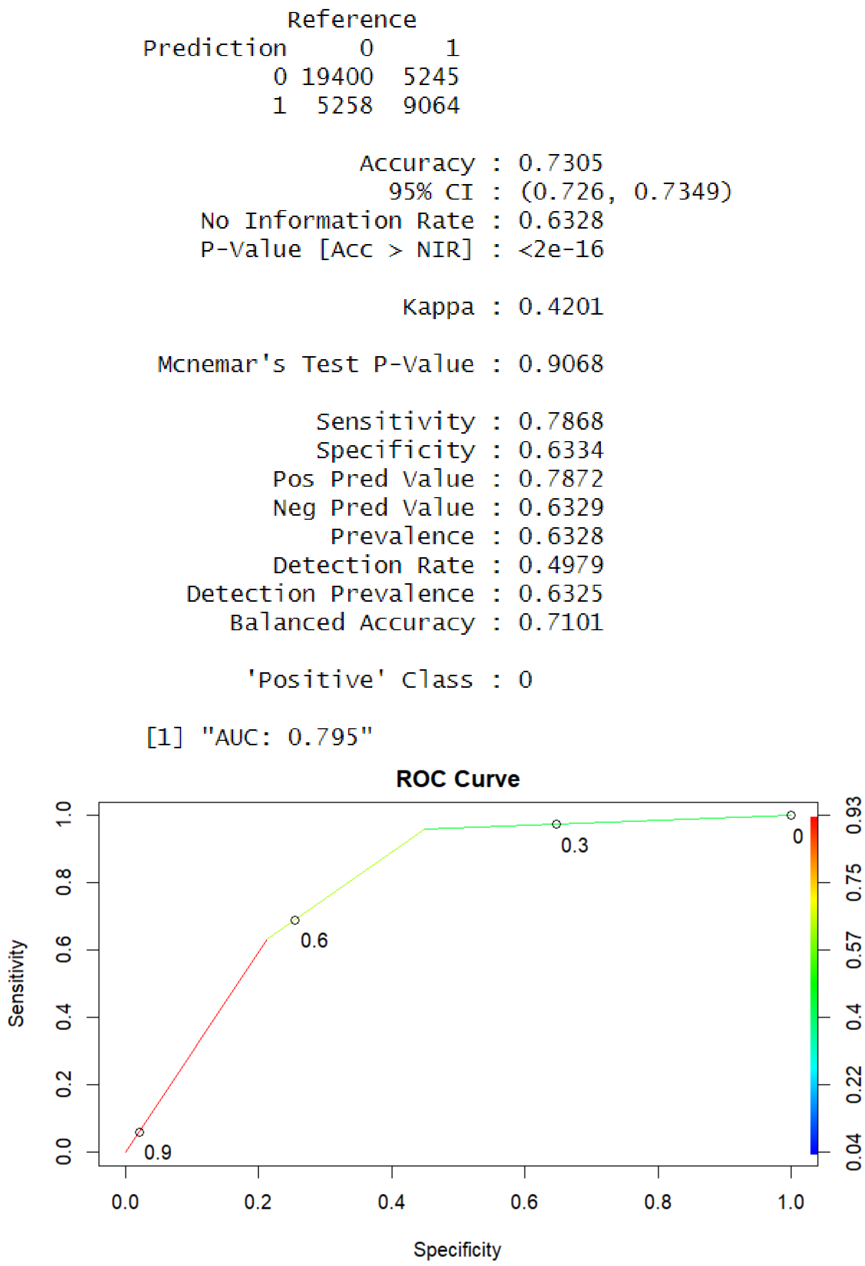
Figure 6 Complexity Parameter of Decision Tree Model 2

Similar to model 1, from the X-val Relative Error vs. cp plot, cp 0.028 has the lowest error rate. However, it is not known whether cp value beyond this can achieve better results.

<u>Prediction and Evaluation on Training Data</u>



~~~
```{r}
#confusion matrix for evaluation
modelCM(dt2, training)

#ROC curve
ROCauc(dt2, training)
```
~~~



```
          Reference
Prediction     0     1
         0 19400  5245
         1  5258  9064

               Accuracy : 0.7305
                 95% CI : (0.726, 0.7349)
    No Information Rate : 0.6328
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.4201

 Mcnemar's Test P-Value : 0.9068

            Sensitivity : 0.7868
            Specificity : 0.6334
         Pos Pred Value : 0.7872
         Neg Pred Value : 0.6329
             Prevalence : 0.6328
         Detection Rate : 0.4979
   Detection Prevalence : 0.6325
      Balanced Accuracy : 0.7101

       'Positive' Class : 0

[1] "AUC: 0.795"
```

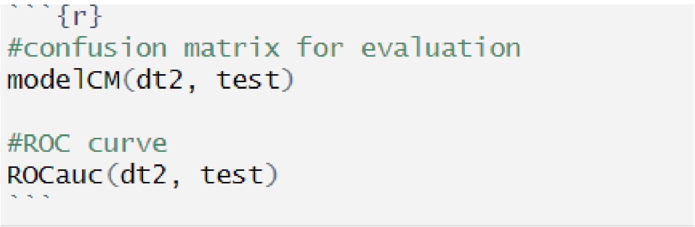
Figure 7 ROC of Decision Tree Model 2 on Training Data

Prediction using Decision Tree model 2 on training data yield 73.05% accuracy, 78.68% sensitivity, 63.34% specificity. AUC is 0.795, which means that there is 79.5% chance that the model can distinguish non-smoking and smoking cases.

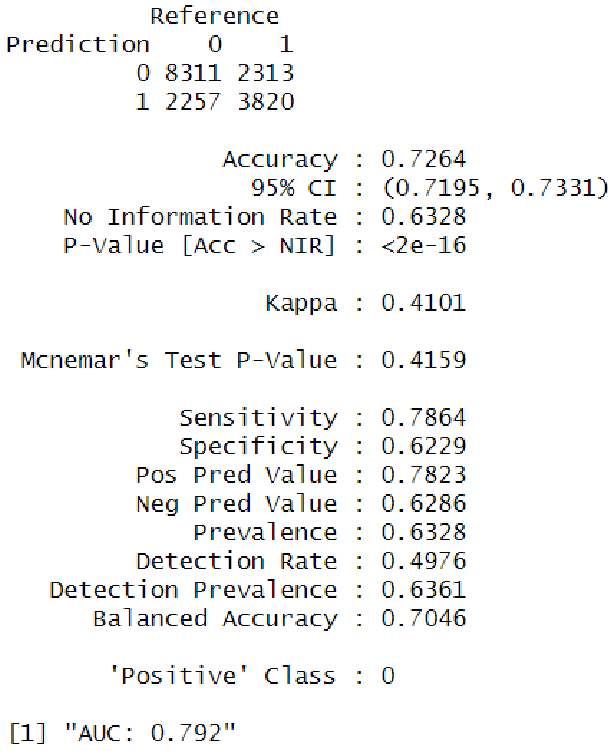
From the confusion matrix output, it was observed that out of the total number of observations, the model correctly classifies 28464 cases which gives an accuracy of 73.05%. Specifically, out of all non-smoking cases, 19400 were correctly classified, achieving 78.68% sensitivity. On the other hand, out of all smoking cases, 9064 cases were correctly classified, achieving

63.34% specificity. Finally, 5258 non-smoking cases and 5245 smoking cases were misclassified.

Prediction and Evaluation on Test Data



~~~
```{r}
#confusion matrix for evaluation
modelCM(dt2, test)

#ROC curve
ROCauc(dt2, test)
```
~~~



```
          Reference
Prediction    0    1
         0 8311 2313
         1 2257 3820

               Accuracy : 0.7264
                 95% CI : (0.7195, 0.7331)
    No Information Rate : 0.6328
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.4101

 Mcnemar's Test P-Value : 0.4159

            Sensitivity : 0.7864
            Specificity : 0.6229
         Pos Pred Value : 0.7823
         Neg Pred Value : 0.6286
             Prevalence : 0.6328
         Detection Rate : 0.4976
   Detection Prevalence : 0.6361
      Balanced Accuracy : 0.7046

       'Positive' Class : 0

[1] "AUC: 0.792"
```

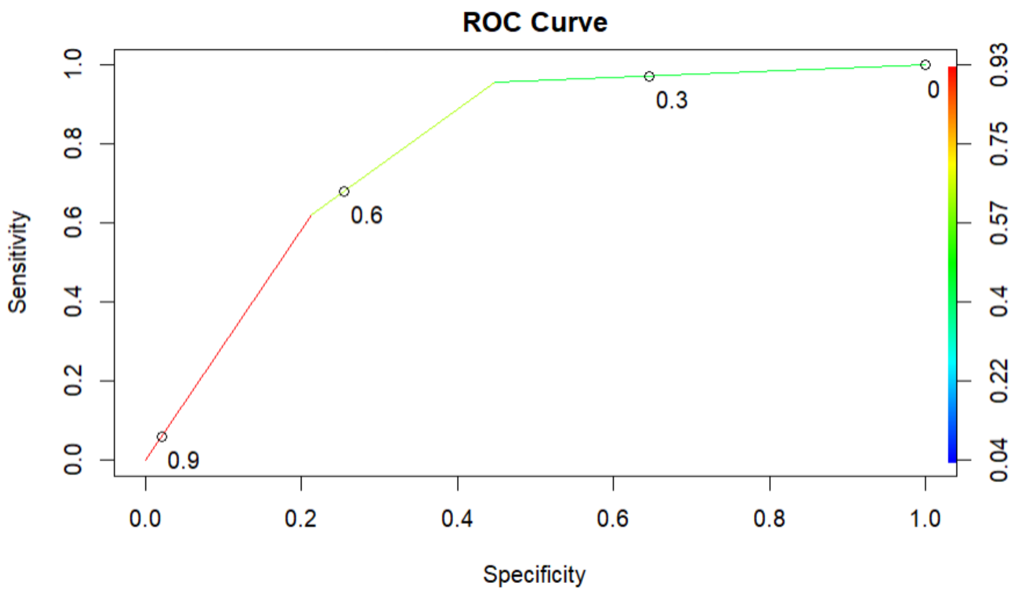
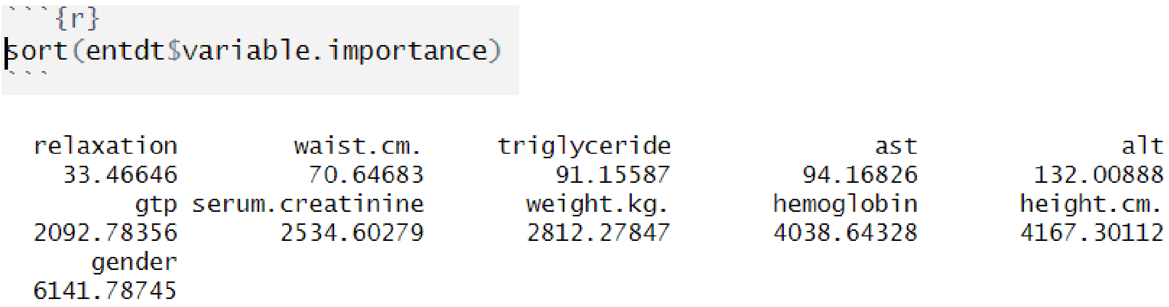


Figure 8 ROC Curve of Decision Tree Model 2 on Test Data

Prediction using Decision Tree model 2 on test data yield 72.64% accuracy, 78.64% sensitivity, 62.29% specificity. The AUC is 0.792, which means that there is 79.2% chance that the model can distinguish non-smoking and smoking cases. From the confusion matrix output, it was observed that out of the total number of observations, the model correctly classifies 12131 cases which gives an accuracy of 72.64%. Specifically, out of all non-smoking cases, 8311 were correctly classified, achieving 78.64% sensitivity. On the other hand, out of all smoking cases, 2313 cases were correctly classified, achieving 62.29% specificity. Finally, 2257 non-smoking cases and 2313 smoking cases were misclassified.

In comparison to the model performance on both training and test data, applying the model on test data has a slightly lower performance. The slight difference between the two suggests that there is no overfitting.

Variable Importance



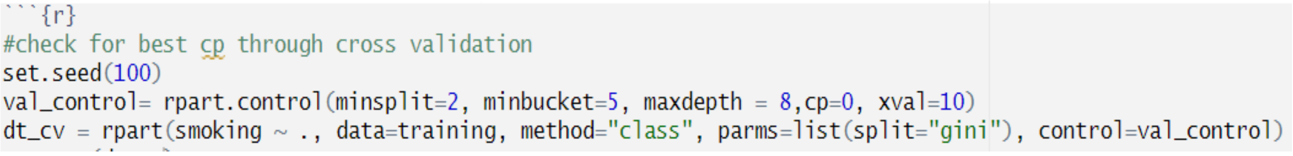
~~~
```{r}
sort(entdt$variable.importance)
```
~~~

```
   relaxation          waist.cm.      triglyceride               ast               alt 
     33.46646           70.64683          91.15587          94.16826         132.00888 
          gtp  serum.creatinine        weight.kg.        hemoglobin        height.cm. 
   2092.78356         2534.60279        2812.27847        4038.64328        4167.30112 
       gender 
   6141.78745 
```

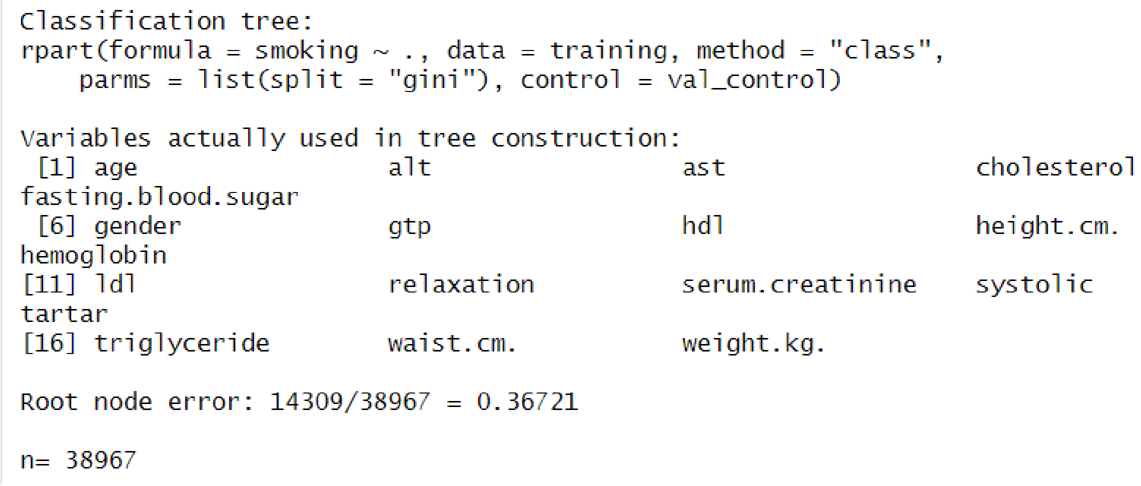
Similar to model 1, the most important variable in this model is gender, followed by height then hemoglobin.

6.1.3 Hyperparameter Tuning: Pruning with Cross Validation

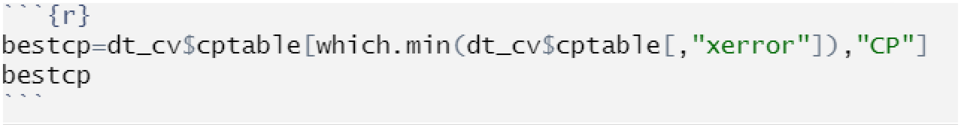
Pruning is a process which was done to reduce the overall complexity of the tree and to reduce the chances of overfitting the model to the training data.



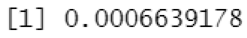
~~~
```{r}
#check for best cp through cross validation
set.seed(100)
val_control= rpart.control(minsplit=2, minbucket=5, maxdepth = 8,cp=0, xval=10)
dt_cv = rpart(smoking ~ ., data=training, method="class", parms=list(split="gini"), control=val_control)
~~~



```
Classification tree:
rpart(formula = smoking ~ ., data = training, method = "class",
    parms = list(split = "gini"), control = val_control)

Variables actually used in tree construction:
 [1] age                alt                ast                cholesterol
fasting.blood.sugar
 [6] gender             gtp                hdl                height.cm.
hemoglobin
[11] ldl                relaxation         serum.creatinine   systolic
tartar
[16] triglyceride       waist.cm.          weight.kg.

Root node error: 14309/38967 = 0.36721

n= 38967
```



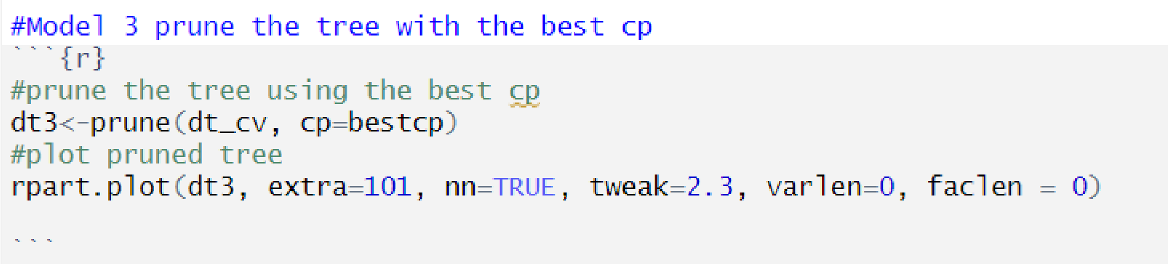
~~~
```{r}
bestcp=dt_cv$cptable[which.min(dt_cv$cptable[,"xerror"]),"CP"]
bestcp
```
~~~



```
[1] 0.0006639178
```

Using the same hyperparameter settings while setting cp=0, the model was trained on the training data through 10-fold cross validation. In this model, only variable tartar was not used for tree construction. From the output, cp 0.00066 will be chosen as the best cp since it has the lowest cross validation error, 0.69082. Hence, this cp value will be used to tune the basic decision tree model.

6.1.4 Model 3



~~~
#Model 3 prune the tree with the best cp
```{r}
#prune the tree using the best cp
dt3<-prune(dt_cv, cp=bestcp)
#plot pruned tree
rpart.plot(dt3, extra=101, nn=TRUE, tweak=2.3, varlen=0, faclen = 0)

```
~~~

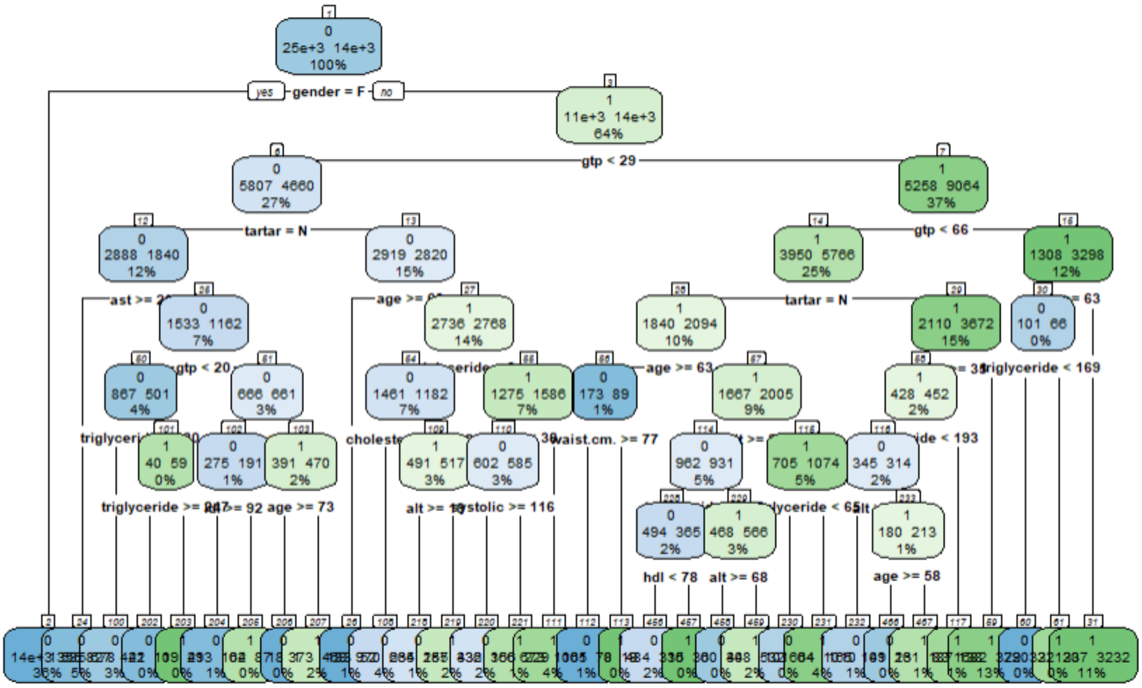
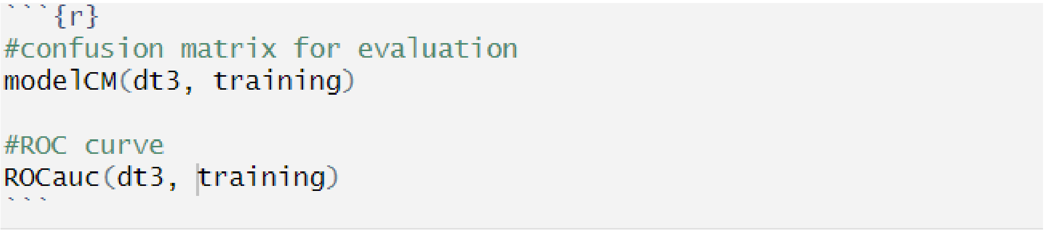


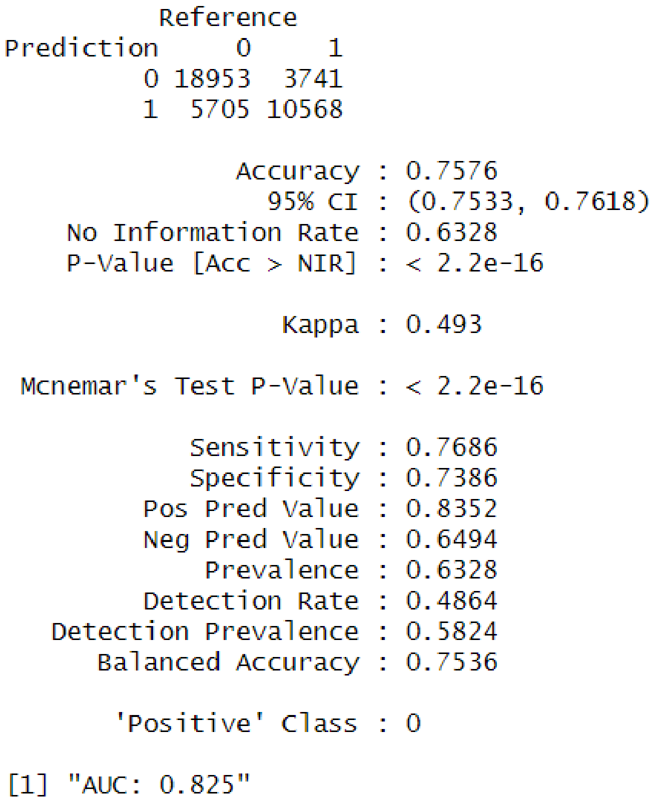
Figure 9 Decision Tree Structure with Best CP

By using the best cp value, the decision tree generated was more complex. In the first layer, gender was used to split the tree node. In the second layer, GTP with condition less than 29 is used to split the node. In the third layer, tartar and GTP with condition less than 66 are used to split the node. Layer by layer, different variables with condition is used to split the tree until homogenous subsets or individual leaf nodes are formed. Next, the tuned model was tested on the training set and evaluated.

Prediction and Evaluation on Training Data



~~~
```{r}
#confusion matrix for evaluation
modelCM(dt3, training)

#ROC curve
ROCauc(dt3, training)
```
~~~



```
          Reference
Prediction     0     1
         0 18953  3741
         1  5705 10568
                                          
               Accuracy : 0.7576          
                 95% CI : (0.7533, 0.7618)
    No Information Rate : 0.6328          
    P-Value [Acc > NIR] : < 2.2e-16       
                                          
                  Kappa : 0.493           
                                          
 Mcnemar's Test P-Value : < 2.2e-16       
                                          
            Sensitivity : 0.7686          
            Specificity : 0.7386          
         Pos Pred Value : 0.8352          
         Neg Pred Value : 0.6494          
             Prevalence : 0.6328          
         Detection Rate : 0.4864          
   Detection Prevalence : 0.5824          
      Balanced Accuracy : 0.7536          
                                          
       'Positive' Class : 0               

[1] "AUC: 0.825"
```

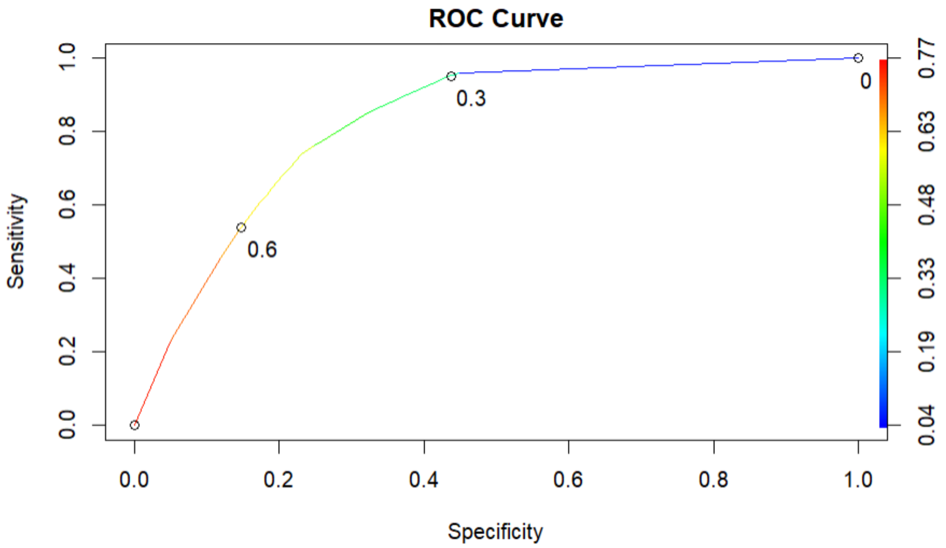
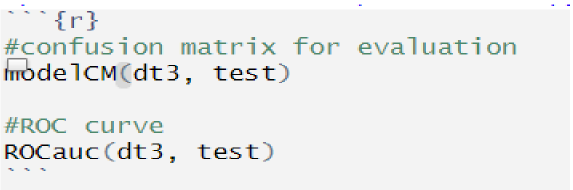


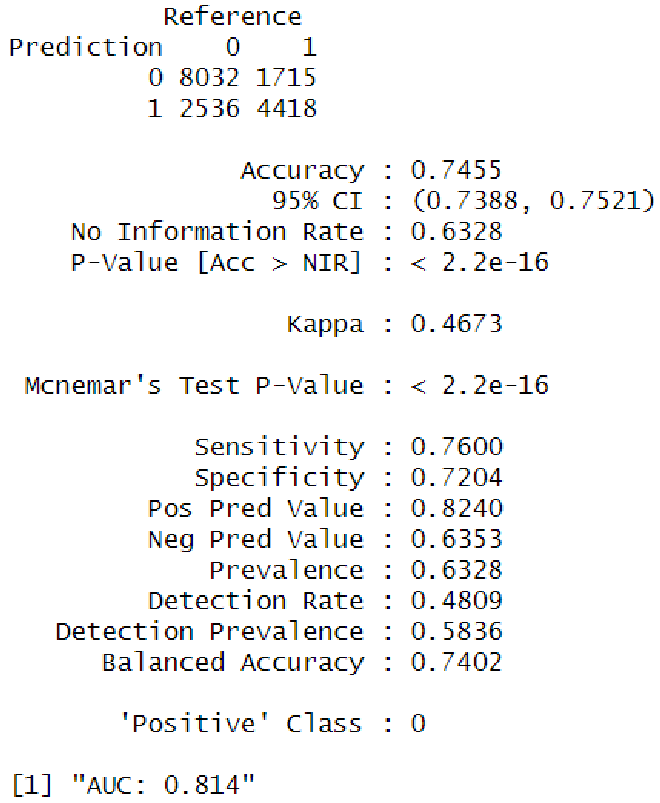
Figure 10 ROC of Decision Tree Model 3 on Training Data

Prediction using Decision Tree tuned model on training data yield 75.76% accuracy, 76.86% sensitivity, 73.86% specificity. The AUC is 0.825, which means that there is 82.5% chance that the model can distinguish non-smoking and smoking cases. From the confusion matrix output, it was observed that out of the total number of observations, the model correctly classifies 29521 cases which gives an accuracy of 75.76%. Specifically, out of all non-smoking cases, 5705 was correctly classified, achieving 76.86% sensitivity. On the other hand, out of all smoking cases, 3741 cases were correctly classified, achieving 73.86% specificity.

Prediction and Evaluation on Test Set



~~~
```{r}
#confusion matrix for evaluation
modelCM(dt3, test)

#ROC curve
ROCauc(dt3, test)
```
~~~



```
           Reference
Prediction    0    1
         0 8032 1715
         1 2536 4418
                                          
               Accuracy : 0.7455          
                 95% CI : (0.7388, 0.7521)
    No Information Rate : 0.6328          
    P-Value [Acc > NIR] : < 2.2e-16       
                                          
                  Kappa : 0.4673          
                                          
 Mcnemar's Test P-Value : < 2.2e-16       
                                          
            Sensitivity : 0.7600          
            Specificity : 0.7204          
         Pos Pred Value : 0.8240          
         Neg Pred Value : 0.6353          
             Prevalence : 0.6328          
         Detection Rate : 0.4809          
   Detection Prevalence : 0.5836          
      Balanced Accuracy : 0.7402          
                                          
       'Positive' Class : 0               

[1] "AUC: 0.814"
```

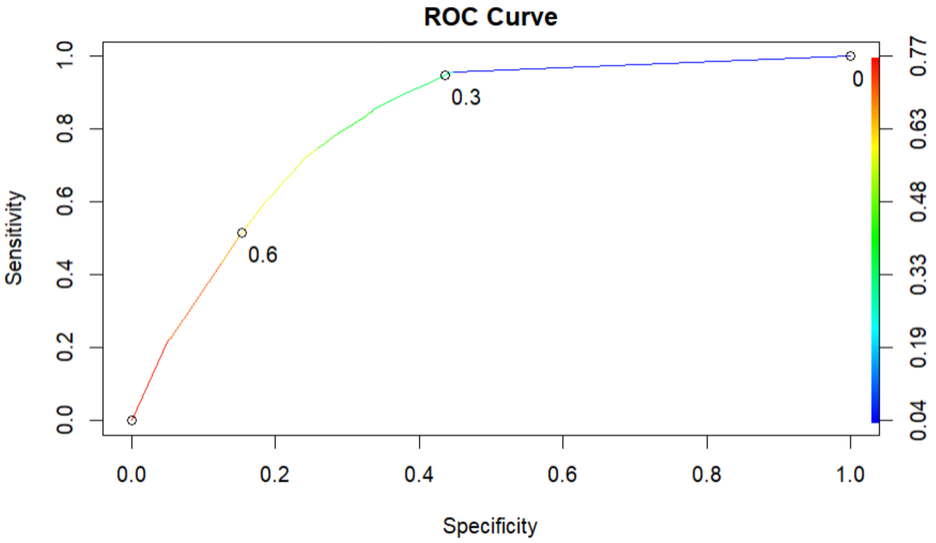
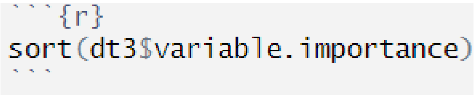


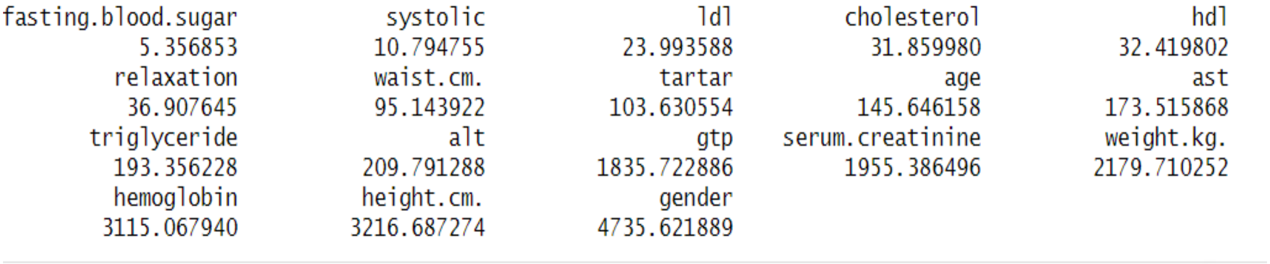
Figure 11 ROC Curve of Decision Tree Model 3 on Test Data

Prediction using Decision Tree tuned model on test data yield 74.55% accuracy, 76% sensitivity, 72.04% specificity. The AUC is 0.814, which means that there is 81.4% chance that the model can distinguish non-smoking and smoking cases. From the confusion matrix output, it was observed that out of the total number of observations, the model correctly classifies 12450 cases which gives an accuracy of 74.55%. Specifically, out of all non-smoking cases, 8032 were correctly classified, achieving 76% sensitivity. On the other hand, out of all smoking cases, 4418 cases were correctly classified, achieving 72.04% specificity. Finally, 2536 non-smoking cases and 1715 smoking cases were misclassified.

Variable Importance



~~~
```{r}
sort(dt3$variable.importance)
```
~~~



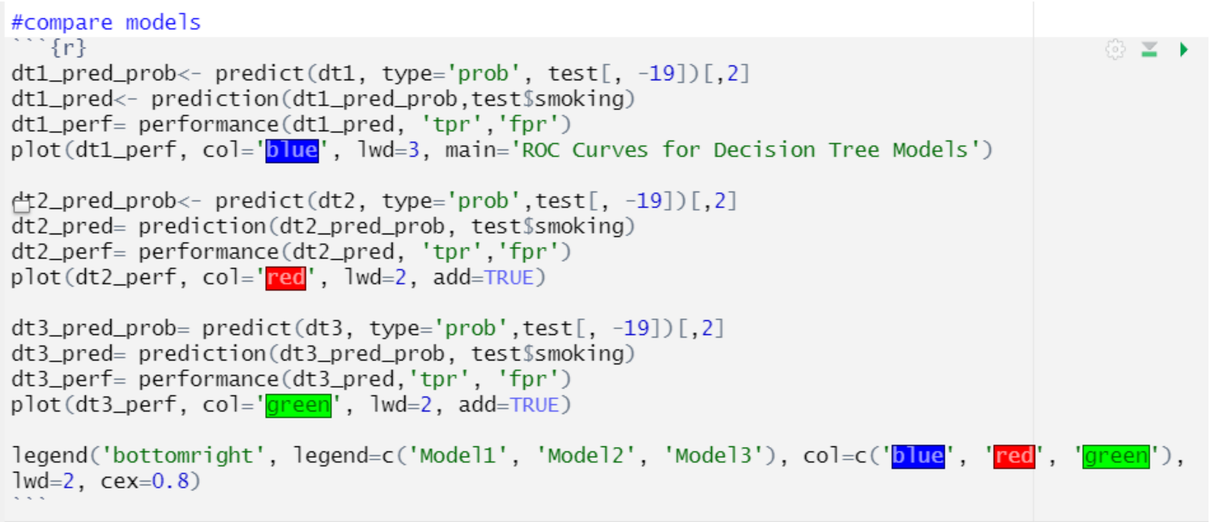
```
fasting.blood.sugar            systolic                 ldl         cholesterol                 hdl 
           5.356853           10.794755           23.993588           31.859980           32.419802 
         relaxation           waist.cm.              tartar                 age                 ast 
          36.907645           95.143922          103.630554          145.646158          173.515868 
       triglyceride                 alt                 gtp    serum.creatinine          weight.kg. 
         193.356228          209.791288         1835.722886         1955.386496         2179.710252 
         hemoglobin          height.cm.              gender 
        3115.067940         3216.687274         4735.621889 
```

In this model, gender is the most important variable, followed by height, hemoglobin and so forth.

6.1.5 Model Comparison

Model	**Data**	**Accuracy**	**AUC**	**Sensitivity**	**Specificity**
Model 1 -split method ‘Gini’	Training	73.05%	0.795	78.68%	63.34%
	Test	72.64%	0.792	78.64%	62.29%
Model 2 -split method ‘entropy information’	Training	73.05%	0.795	78.68%	63.34%
	test	72.64%	0.792	78.64%	62.29%
Model 3 -tuned best cp=0.000664	Training	75.76%	0.825	76.86%	73.86%
	Test	74.55%	0.814	76%	72.04%

Figure 12 Decision Tree Model Comparison



~~~
#compare models
```{r}
dt1_pred_prob<- predict(dt1, type='prob', test[, -19])[,2]
dt1_pred<- prediction(dt1_pred_prob,test$smoking)
dt1_perf= performance(dt1_pred, 'tpr','fpr')
plot(dt1_perf, col='blue', lwd=3, main='ROC Curves for Decision Tree Models')

dt2_pred_prob<- predict(dt2, type='prob',test[, -19])[,2]
dt2_pred= prediction(dt2_pred_prob, test$smoking)
dt2_perf= performance(dt2_pred, 'tpr','fpr')
plot(dt2_perf, col='red', lwd=2, add=TRUE)

dt3_pred_prob= predict(dt3, type='prob',test[, -19])[,2]
dt3_pred= prediction(dt3_pred_prob, test$smoking)
dt3_perf= performance(dt3_pred,'tpr', 'fpr')
plot(dt3_perf, col='green', lwd=2, add=TRUE)

legend('bottomright', legend=c('Model1', 'Model2', 'Model3'), col=c('blue', 'red', 'green'),
lwd=2, cex=0.8)
```
~~~

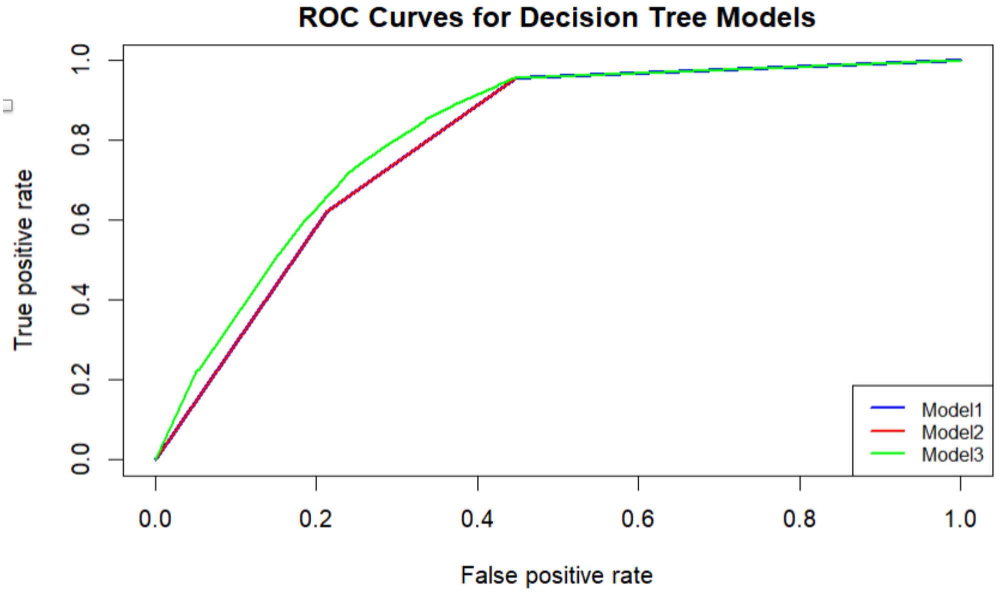


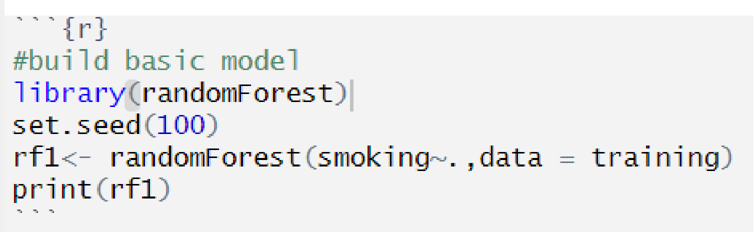
Figure 13 ROC Curve of All Decision Tree Models

It is observed that Model 3 which was tuned with the best cp value has the best performance. As shown on figure 25, Model 3 is closer towards the upper left corner, which false positive rate is low while true positive rate is high, which shows that model 3 can classify the classes better than model 1 and 2. The performance between training and the test data does not differ much, suggesting that the model is not overfit. Moreover, the performance of Model 3 on training data is considered in a good range (accuracy: 75.76%, AUC:0.825). Therefore, model 3 is selected as the best Decision Tree model.

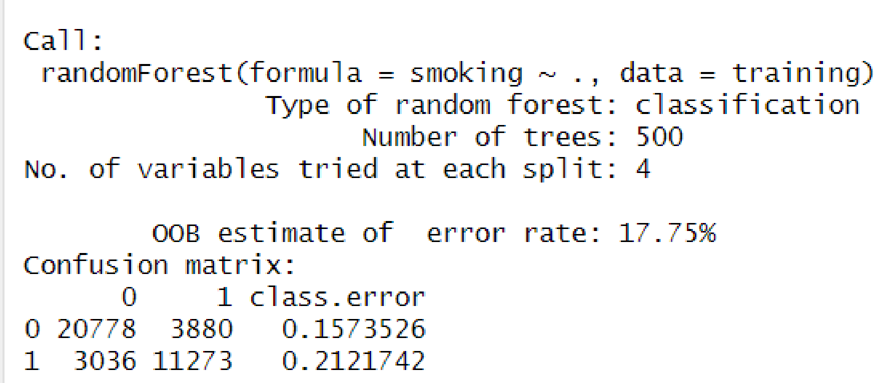
6.2 Random Forest

6.2.1 Model 1

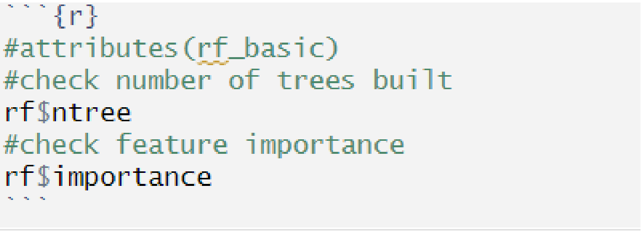
<u>Build Model</u>



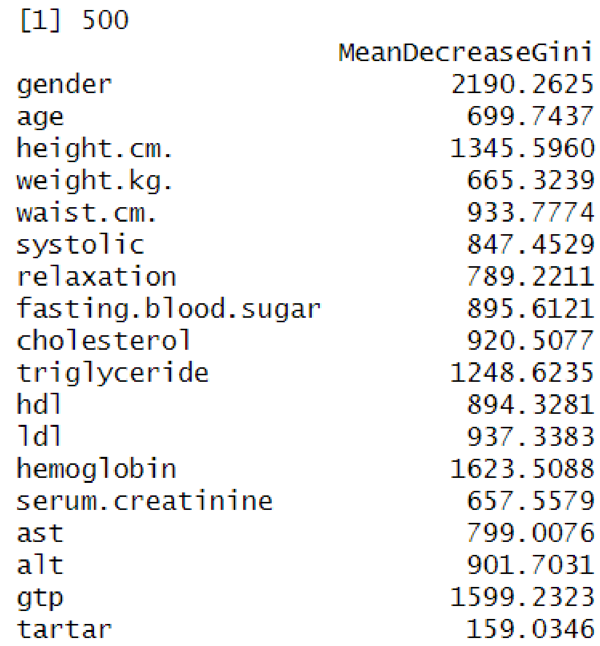
~~~
```{r}
#build basic model
library(randomForest)
set.seed(100)
rf1<- randomForest(smoking~.,data = training)
print(rf1)
```
~~~



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

        OOB estimate of  error rate: 17.75%
Confusion matrix:
      0     1 class.error
0 20778  3880   0.1573526
1  3036 11273   0.2121742
```



~~~
```{r}
#attributes(rf_basic)
#check number of trees built
rf$ntree
#check feature importance
rf$importance
```
~~~

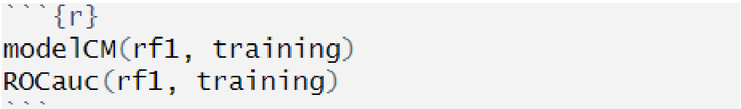


```
[1] 500
                    MeanDecreaseGini
gender                     2190.2625
age                         699.7437
height.cm.                 1345.5960
weight.kg.                  665.3239
waist.cm.                   933.7774
systolic                    847.4529
relaxation                  789.2211
fasting.blood.sugar         895.6121
cholesterol                 920.5077
triglyceride               1248.6235
hdl                         894.3281
ldl                         937.3383
hemoglobin                 1623.5088
serum.creatinine            657.5579
ast                         799.0076
alt                         901.7031
gtp                        1599.2323
tartar                      159.0346
```

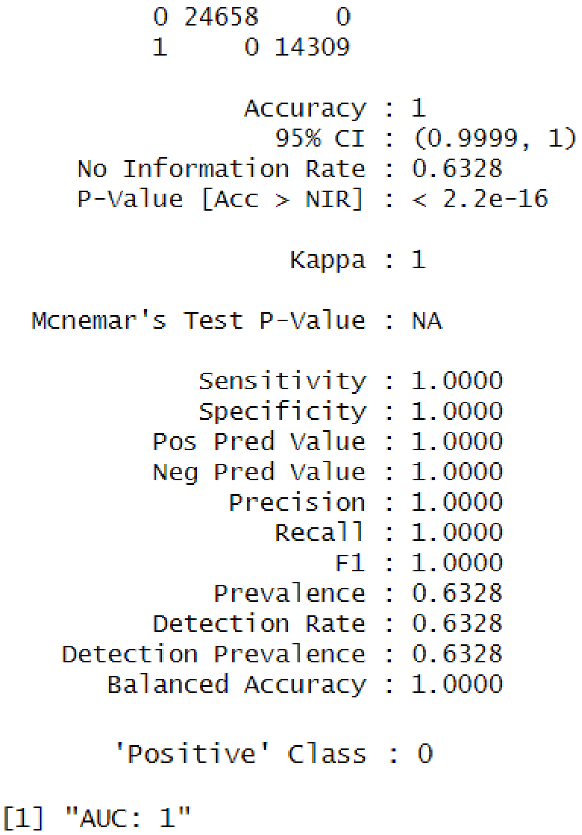
Model 1 uses 500 trees and 4 variables in each split (mtry) to build the classifier. The OOB estimate of error rate for this model is 17.75%, which means that out of the total number of observations, 6916 of the observations in this training set were misclassified.

Based on the ‘Mean Decreases Gini’, it shoes that gender has the highest value, followed by hemoglobin and GTP.

<u>Prediction and Evaluation on Training Set</u>



~~~
```{r}
modelCM(rf1, training)
ROCauc(rf1, training)
```
~~~



```
          0 24658     0
          1     0 14309
                                      
               Accuracy : 1           
                 95% CI : (0.9999, 1) 
    No Information Rate : 0.6328      
    P-Value [Acc > NIR] : < 2.2e-16   
                                      
                  Kappa : 1           
                                      
 Mcnemar's Test P-Value : NA          
                                      
            Sensitivity : 1.0000      
            Specificity : 1.0000      
         Pos Pred Value : 1.0000      
         Neg Pred Value : 1.0000      
              Precision : 1.0000      
                 Recall : 1.0000      
                     F1 : 1.0000      
             Prevalence : 0.6328      
         Detection Rate : 0.6328      
   Detection Prevalence : 0.6328      
      Balanced Accuracy : 1.0000      
                                      
       'Positive' Class : 0           
                                      
[1] "AUC: 1"
```

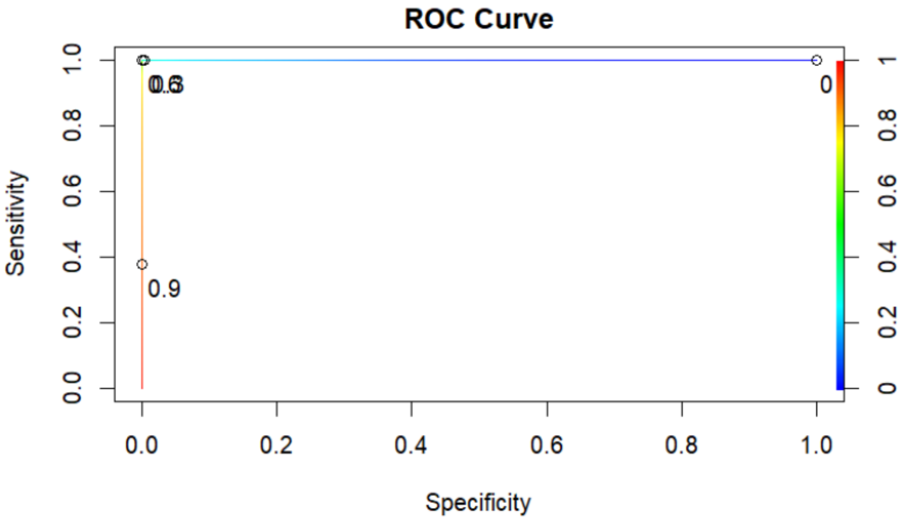
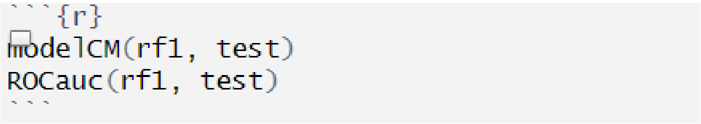


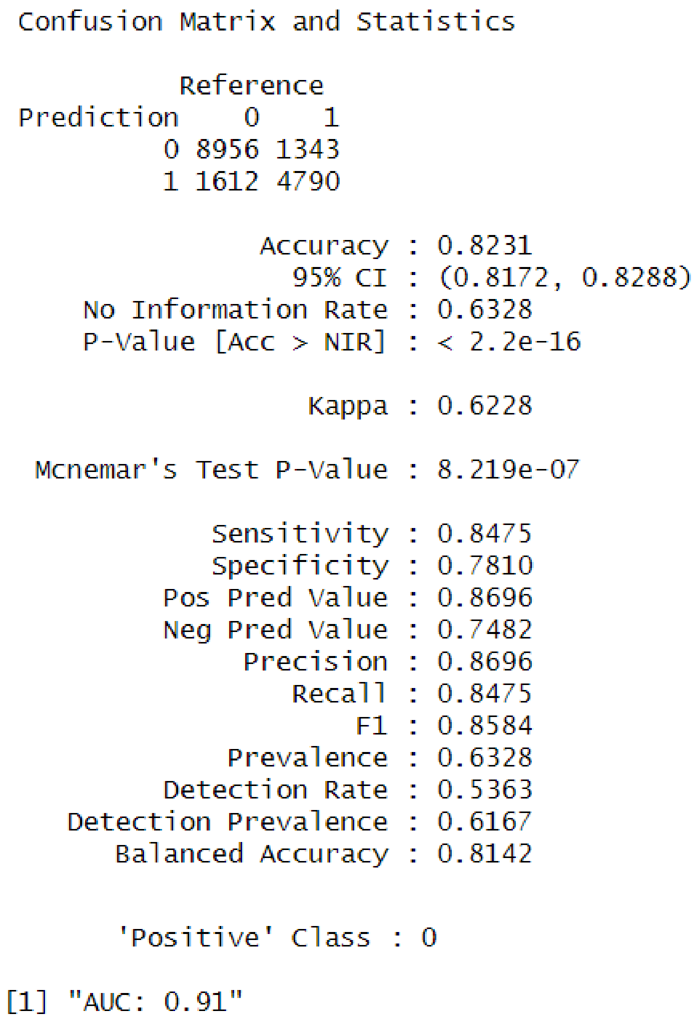
Figure 14 ROC curve of Random Forest Model 1 on Training Data

Prediction using model 1 on training data yield the perfect performance 100% in accuracy, sensitivity and specificity. The AUC is 1, which means that there is 100% chance that the model can distinguish non-smoking and smoking cases. From the confusion matrix output, all observations are correctly classified.

Prediction and Evaluation on Test Set



~~~
```{r}
modelCM(rf1, test)
ROCauc(rf1, test)
```
~~~



```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 8956 1343
         1 1612 4790

               Accuracy : 0.8231
                 95% CI : (0.8172, 0.8288)
    No Information Rate : 0.6328
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.6228

 Mcnemar's Test P-Value : 8.219e-07

            Sensitivity : 0.8475
            Specificity : 0.7810
         Pos Pred Value : 0.8696
         Neg Pred Value : 0.7482
              Precision : 0.8696
                 Recall : 0.8475
                     F1 : 0.8584
             Prevalence : 0.6328
         Detection Rate : 0.5363
   Detection Prevalence : 0.6167
      Balanced Accuracy : 0.8142

       'Positive' Class : 0

[1] "AUC: 0.91"
```

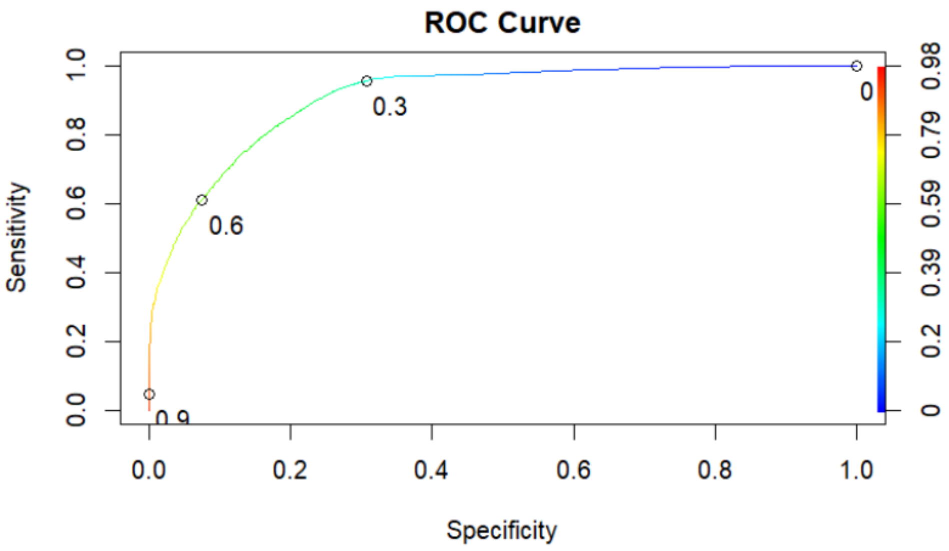


Figure 15 ROC curve of Random Forest Model 1 on Test Data

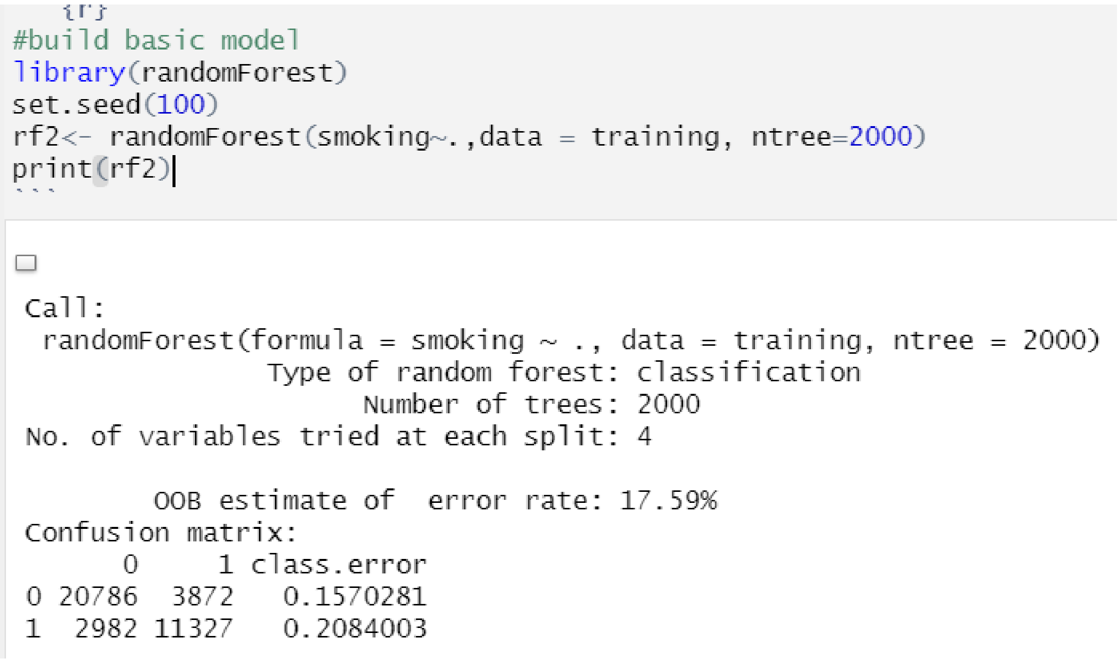
Prediction using model 1 on test data yield 82.31% accuracy, 84.75% sensitivity, 78.10% specificity. The AUC is 0.91, which means that there is 91% chance that the model can distinguish non-smoking and smoking cases. From the confusion matrix output, it was observed that out of the total number of observations, the model correctly classifies 13746 cases which gives an accuracy of 82.31%. Specifically, out of all non-smoking cases, 8956 were correctly classified, achieving 84.75% sensitivity. On the other hand, out of all smoking cases, 4790 cases were correctly classified, achieving 78.10% specificity. Finally, 1612 non-smoking cases and 1343 smoking cases were misclassified.

In RF model, multiple hyperparameters can be tuned to increase performance of the model. These include ntree (the number of trees in the forest), mtry (the number of variables randomly selected at each split, mtry is the sqrt of number of features when performing classification), nodesize (maximum number size of terminal nodes. Large number causes smaller trees to be grown) and maxnodes (the maximum number of terminal nodes in the forest). Different approaches of tuning the hyperparameter is available.

In the next model, ntree will be increased to 2000.

6.2.2 Model 2

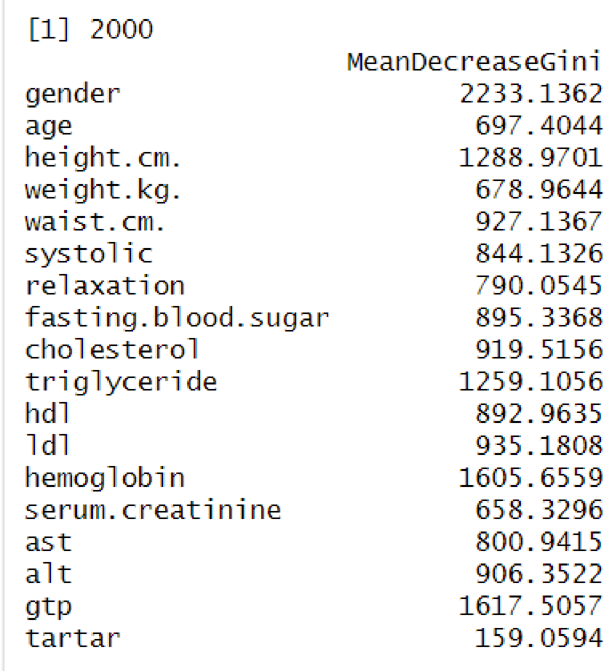
Build Model



```
   {r}
#build basic model
library(randomForest)
set.seed(100)
rf2<- randomForest(smoking~.,data = training, ntree=2000)
print(rf2)
```

```
□

Call:
 randomForest(formula = smoking ~ ., data = training, ntree = 2000) 
               Type of random forest: classification
                     Number of trees: 2000
No. of variables tried at each split: 4

        OOB estimate of  error rate: 17.59%
Confusion matrix:
      0     1 class.error
0 20786  3872   0.1570281
1  2982 11327   0.2084003
```

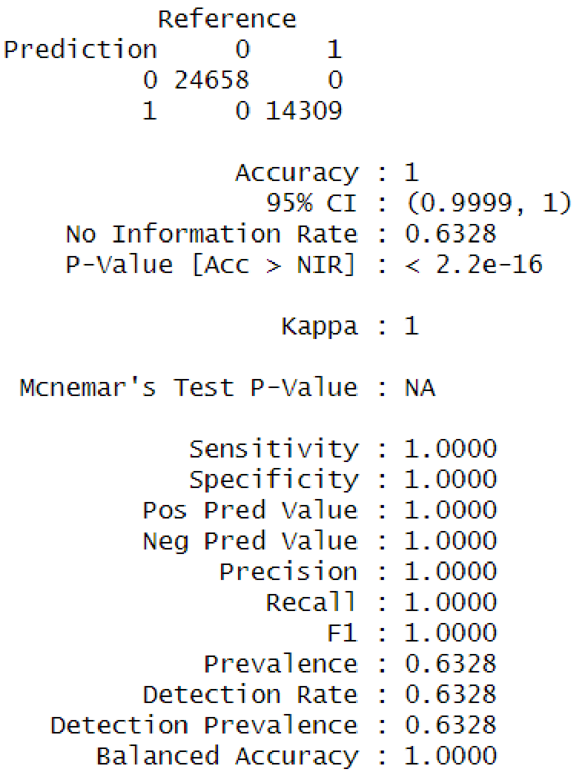
Model 2 uses 2000 trees and 4 variables in each split (mtry) to build the classifier. The OOB estimate of error rate for this model is 17.59%, which means that out of the total number of observations, 6854 of the observations in this training set were misclassified.



```
[1] 2000
                     MeanDecreaseGini
gender                      2233.1362
age                          697.4044
height.cm.                  1288.9701
weight.kg.                   678.9644
waist.cm.                    927.1367
systolic                     844.1326
relaxation                   790.0545
fasting.blood.sugar          895.3368
cholesterol                  919.5156
triglyceride                1259.1056
hdl                          892.9635
ldl                          935.1808
hemoglobin                  1605.6559
serum.creatinine             658.3296
ast                          800.9415
alt                          906.3522
gtp                         1617.5057
tartar                       159.0594
```

Based on the 'Mean Decreases Gini', it shoes that gender has the highest value, followed by hemoglobin and GTP.

Prediction and Evaluation on Training Set



```
          Reference
Prediction     0     1
         0 24658     0
         1     0 14309
                                     
               Accuracy : 1          
                 95% CI : (0.9999, 1)
    No Information Rate : 0.6328     
    P-Value [Acc > NIR] : < 2.2e-16  
                                     
                  Kappa : 1          
                                     
 Mcnemar's Test P-Value : NA         
                                     
            Sensitivity : 1.0000     
            Specificity : 1.0000     
         Pos Pred Value : 1.0000     
         Neg Pred Value : 1.0000     
              Precision : 1.0000     
                 Recall : 1.0000     
                     F1 : 1.0000     
             Prevalence : 0.6328     
         Detection Rate : 0.6328     
   Detection Prevalence : 0.6328     
      Balanced Accuracy : 1.0000     
```

```
       'Positive' Class : 0

[1] "AUC: 1"
```

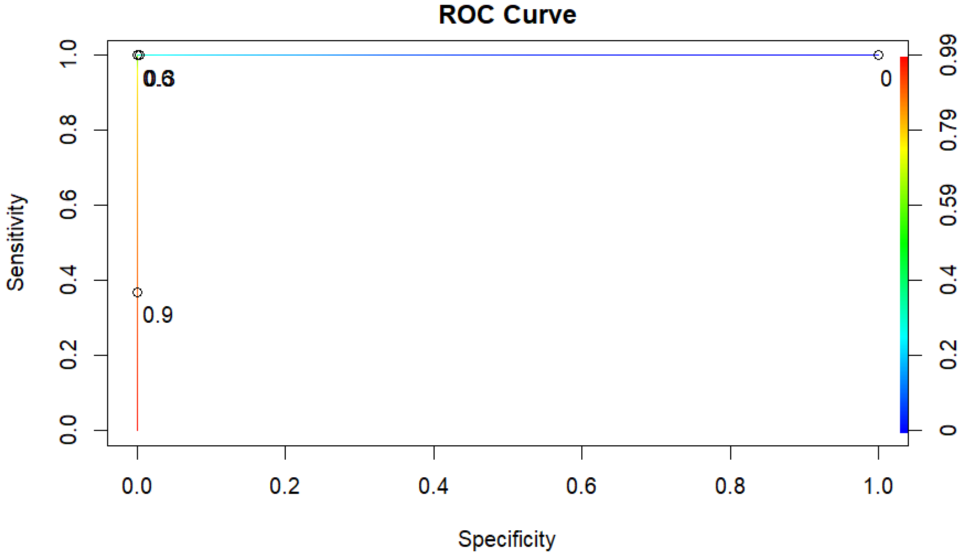
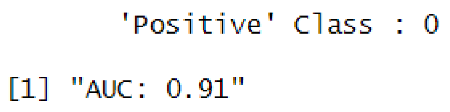


Figure 16 ROC Curve Random Forest Model 2 on Training Data

Similarly, applying model 2 on the training data gives the perfect performance.

Prediction and Evaluation on Test Set

```
 Confusion Matrix and Statistics

           Reference
 Prediction    0    1
          0 8960 1323
          1 1608 4810

                Accuracy : 0.8245
                  95% CI : (0.8186, 0.8302)
     No Information Rate : 0.6328
     P-Value [Acc > NIR] : < 2.2e-16

                   Kappa : 0.626

  Mcnemar's Test P-Value : 1.556e-07

             Sensitivity : 0.8478
             Specificity : 0.7843
          Pos Pred Value : 0.8713
          Neg Pred Value : 0.7495
               Precision : 0.8713
                  Recall : 0.8478
                      F1 : 0.8594
              Prevalence : 0.6328
          Detection Rate : 0.5365
    Detection Prevalence : 0.6157
       Balanced Accuracy : 0.8161
```



```
       'Positive' Class : 0

[1] "AUC: 0.91"
```

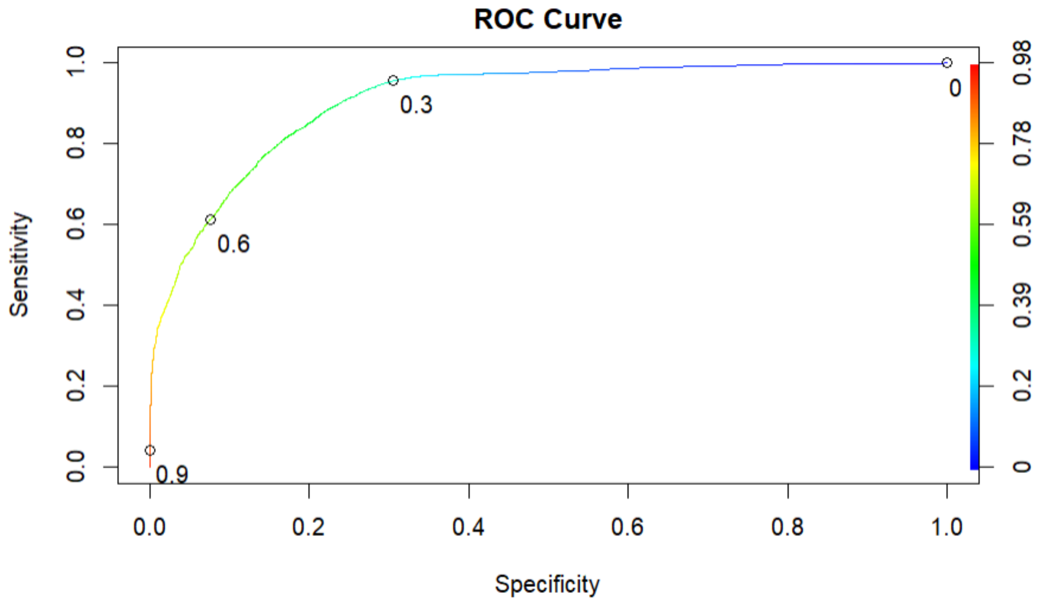
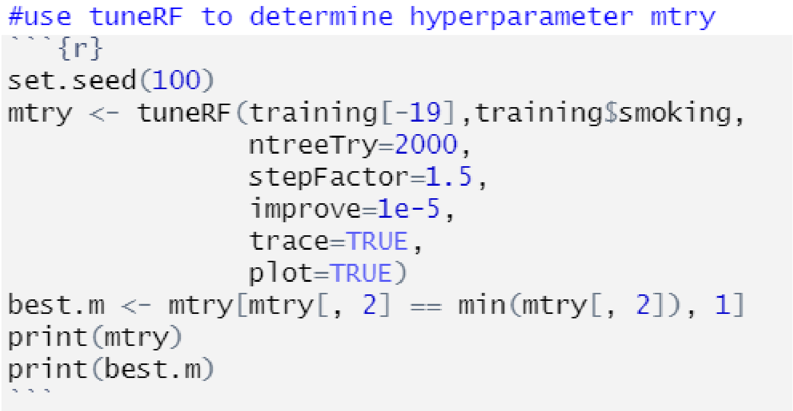


Figure 17 ROC curve of Random Forest Model 2 on Test Data

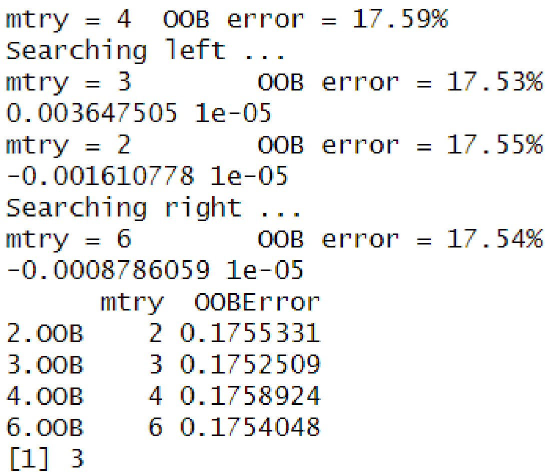
Prediction using model 2 on test data yield 82.45% accuracy, 84.78% sensitivity, 78.43% specificity. The AUC is 0.91, which means that there is 91% chance that the model can distinguish non-smoking and smoking cases. From the confusion matrix output, it was observed that out of the total number of observations, the model correctly classifies 13770 cases which gives an accuracy of 82.45%. Specifically, out of all non-smoking cases, 8960 were correctly classified, achieving 84.78% sensitivity. On the other hand, out of all smoking cases, 4810 cases were correctly classified, achieving 78.43% specificity. Finally, 1608 non-smoking cases and 1323 smoking cases were misclassified.

6.2.3 Hyperparameter Tuning Using tuneRF

While keeping ntree=2000, tuneRF function will be used to determine the best mtry hyperparmeter.



~~~
#use tuneRF to determine hyperparameter mtry
```{r}
set.seed(100)
mtry <- tuneRF(training[-19],training$smoking,
               ntreeTry=2000,
               stepFactor=1.5,
               improve=1e-5,
               trace=TRUE,
               plot=TRUE)
best.m <- mtry[mtry[, 2] == min(mtry[, 2]), 1]
print(mtry)
print(best.m)
```
~~~



```
mtry = 4  OOB error = 17.59%
Searching left ...
mtry = 3        OOB error = 17.53%
0.003647505 1e-05
mtry = 2        OOB error = 17.55%
-0.001610778 1e-05
Searching right ...
mtry = 6        OOB error = 17.54%
-0.0008786059 1e-05
      mtry  OOBError
2.OOB    2 0.1755331
3.OOB    3 0.1752509
4.OOB    4 0.1758924
6.OOB    6 0.1754048
[1] 3
```

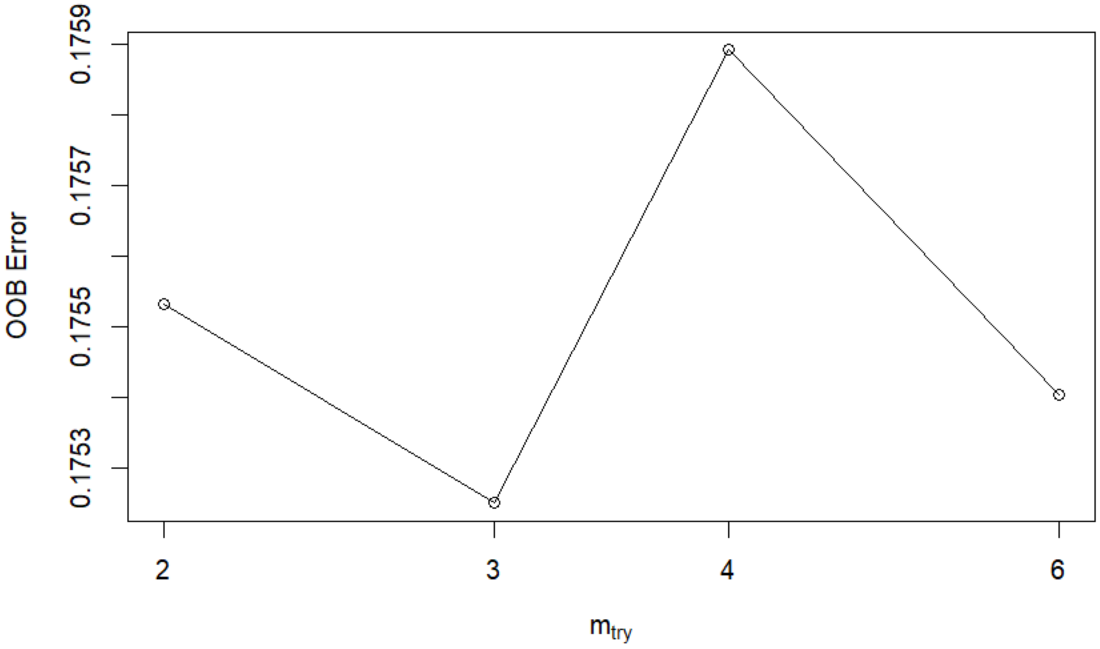
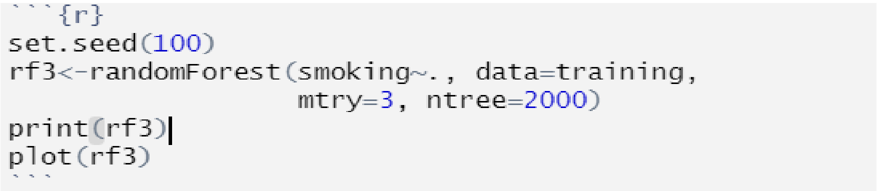


Figure 18 mTry vs. OOB Error

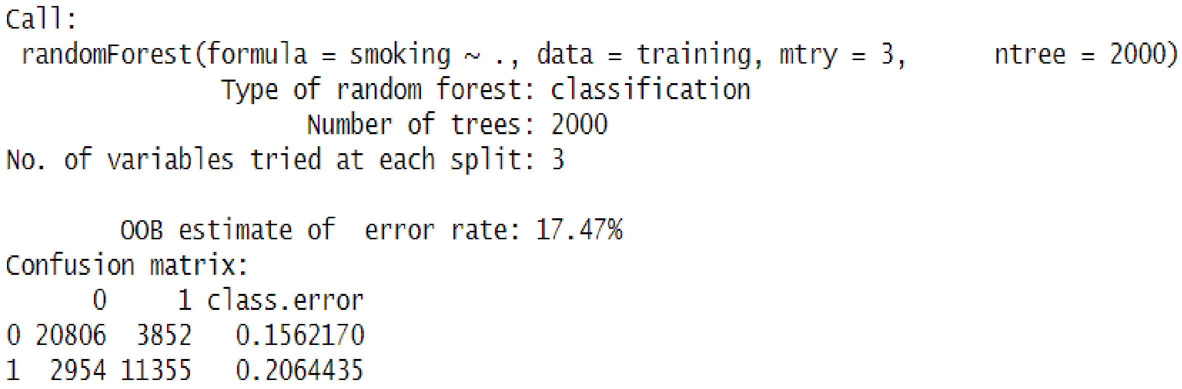
As shown in the plot, when OOB Error is in the lowest when mtry is 3. So, the third model will be built using mtry=3 and ntree=2000.

6.2.4 Model 3

Build Model with best mtry



~~~
```{r}
set.seed(100)
rf3<-randomForest(smoking~., data=training,
                  mtry=3, ntree=2000)
print(rf3)
plot(rf3)
```
~~~



```
Call:
 randomForest(formula = smoking ~ ., data = training, mtry = 3,      ntree = 2000) 
               Type of random forest: classification
                     Number of trees: 2000
No. of variables tried at each split: 3

        OOB estimate of  error rate: 17.47%
Confusion matrix:
      0     1 class.error
0 20806  3852   0.1562170
1  2954 11355   0.2064435
```

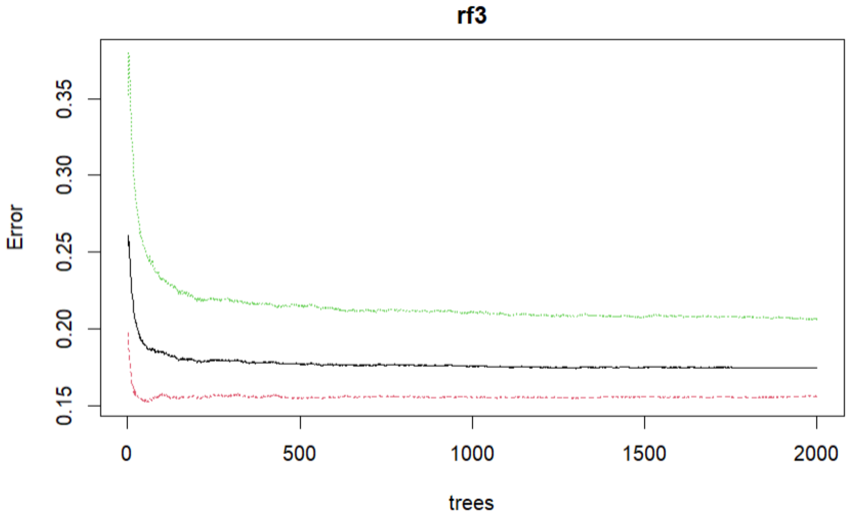
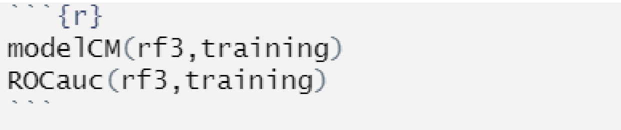


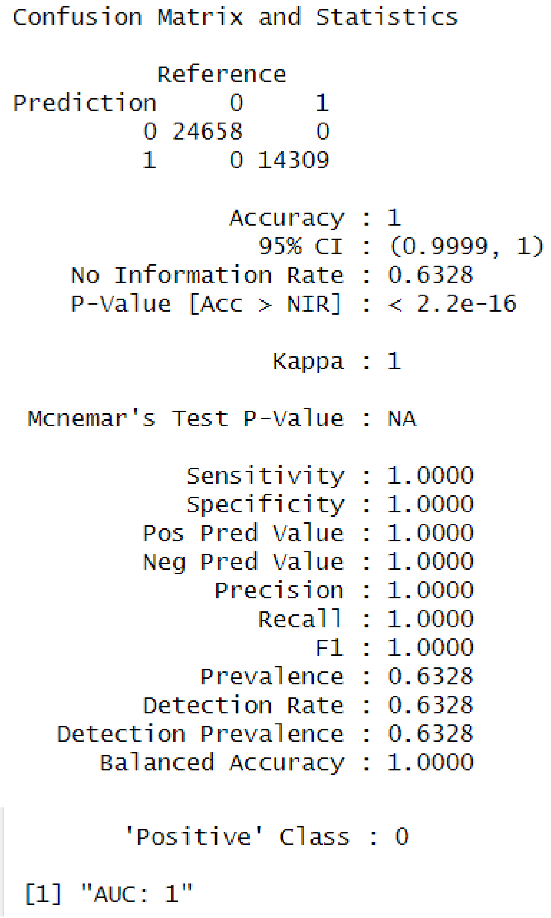
Figure 19Number of Trees vs. Error Rate

Based on Figure 31, Model 3 uses 2000 trees and 3 variables in each split (mtry) to build the classifier. The OOB estimate of error rate for this model is 17.47%, which means that out of the total number of observations, 6806 of the observations in this training set were misclassified.

Prediction and Evaluation on Training Set



~~~
```{r}
modelCM(rf3,training)
ROCauc(rf3,training)
```
~~~



```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 24658     0
         1     0 14309
                                          
               Accuracy : 1               
                 95% CI : (0.9999, 1)     
    No Information Rate : 0.6328          
    P-Value [Acc > NIR] : < 2.2e-16       
                                          
                  Kappa : 1               
                                          
 Mcnemar's Test P-Value : NA              
                                          
            Sensitivity : 1.0000          
            Specificity : 1.0000          
         Pos Pred Value : 1.0000          
         Neg Pred Value : 1.0000          
              Precision : 1.0000          
                 Recall : 1.0000          
                     F1 : 1.0000          
             Prevalence : 0.6328          
         Detection Rate : 0.6328          
   Detection Prevalence : 0.6328          
      Balanced Accuracy : 1.0000          
                                          
       'Positive' Class : 0               
                                          
[1] "AUC: 1"
```

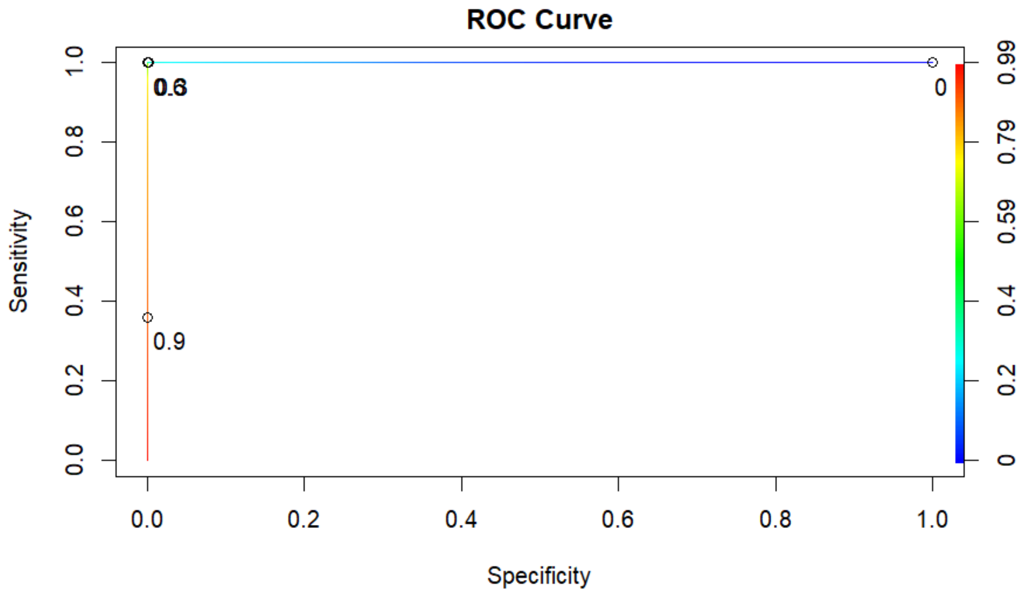
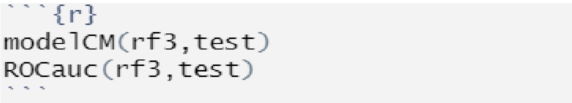


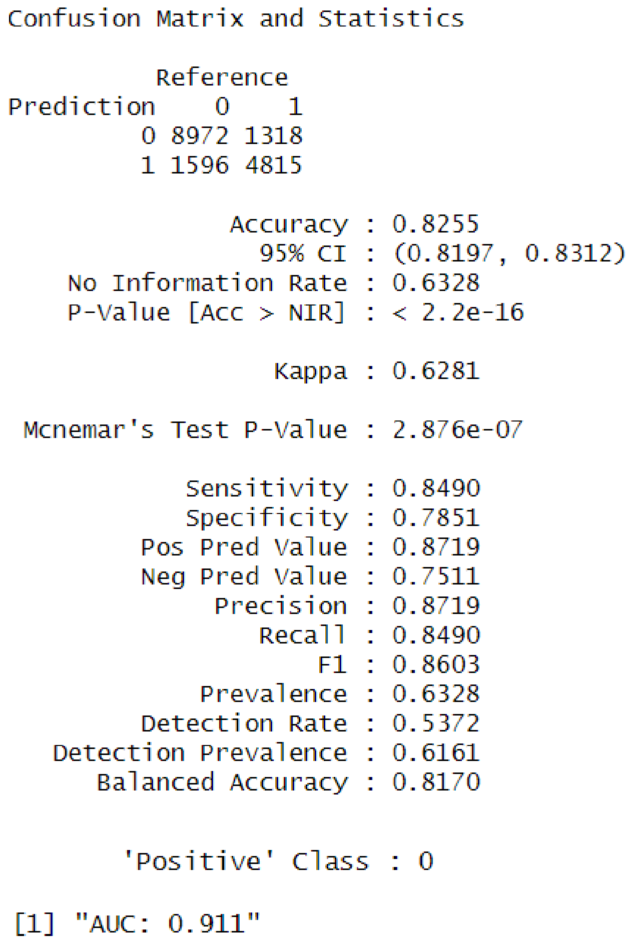
Figure 20 ROC Curve of Random Forest Model 3 on Training Data

Similar to model 1 and 2, applying model 3 on the training data gives the perfect performance.

Prediction and Evaluation on Test Set



~~~
```{r}
modelCM(rf3,test)
ROCauc(rf3,test)
```
~~~



```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 8972 1318
         1 1596 4815

               Accuracy : 0.8255
                 95% CI : (0.8197, 0.8312)
    No Information Rate : 0.6328
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.6281

 Mcnemar's Test P-Value : 2.876e-07

            Sensitivity : 0.8490
            Specificity : 0.7851
         Pos Pred Value : 0.8719
         Neg Pred Value : 0.7511
              Precision : 0.8719
                 Recall : 0.8490
                     F1 : 0.8603
             Prevalence : 0.6328
         Detection Rate : 0.5372
   Detection Prevalence : 0.6161
      Balanced Accuracy : 0.8170

       'Positive' Class : 0

[1] "AUC: 0.911"
```

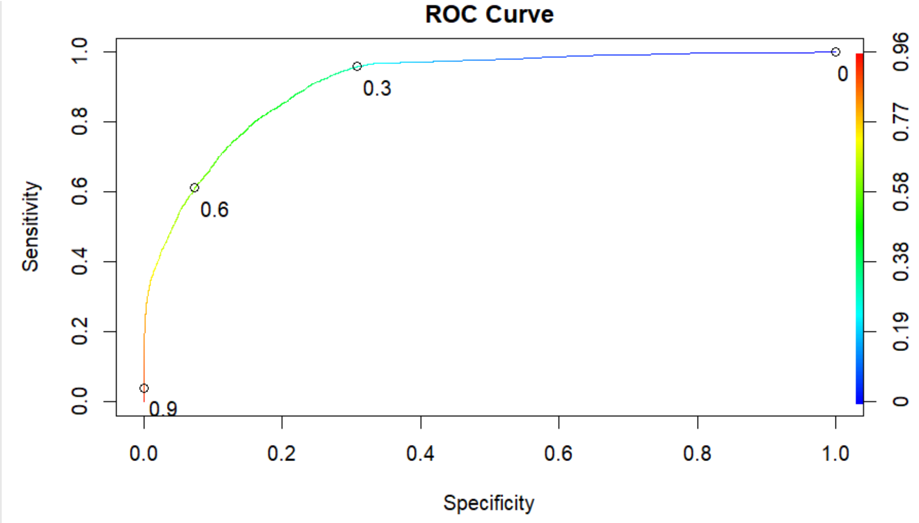
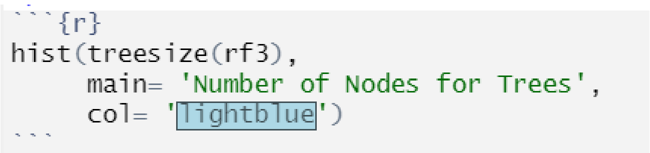


Figure 21 ROC curve of Random Forest Model 3 on Test Data

Prediction using model 3 on test data yield 82.55% accuracy, 84.90% sensitivity, 78.51% specificity. The AUC is 0.911, which means that there is 91.1% chance that the model can distinguish non-smoking and smoking cases. From the confusion matrix output, it was observed that out of the total number of observations, the model correctly classifies 13787 cases which gives an accuracy of 82.55%. Specifically, out of all non-smoking cases, 8972 were correctly classified, achieving 84.90% sensitivity. On the other hand, out of all smoking cases, 4815 cases were correctly classified, achieving 78.51% specificity. Finally, 1596 non-smoking cases and 1318 smoking cases were misclassified.

Check Number of Nodes



~~~
```{r}
hist(treesize(rf3),
     main= 'Number of Nodes for Trees',
     col= 'lightblue')
```
~~~

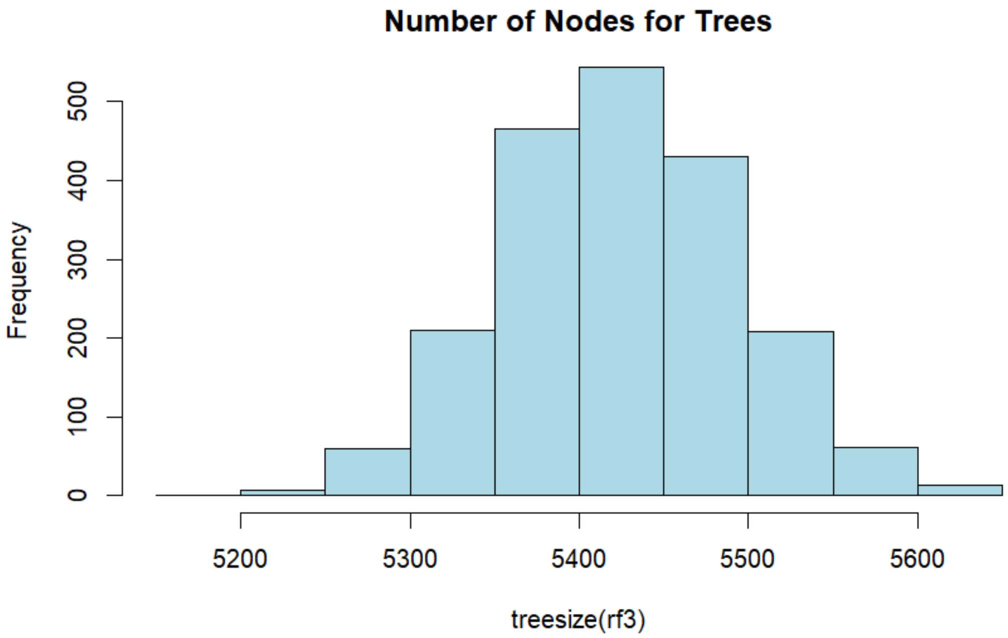
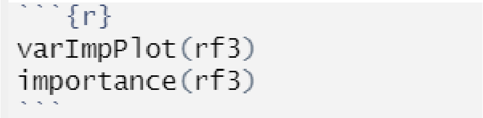


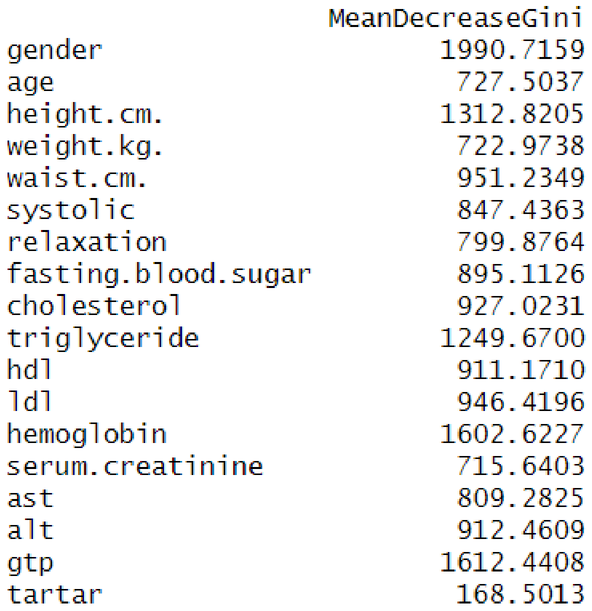
Figure 22 Histogram of Tree Sizes in Random Forest Model 3

The number of nodes of trees built is between 5200-5700 with majority of the trees having 5400 tree nodes.

Check Variable Importance



~~~
```{r}
varImpPlot(rf3)
importance(rf3)
```
~~~



```
                        MeanDecreaseGini
gender                         1990.7159
age                             727.5037
height.cm.                     1312.8205
weight.kg.                      722.9738
waist.cm.                       951.2349
systolic                        847.4363
relaxation                      799.8764
fasting.blood.sugar             895.1126
cholesterol                     927.0231
triglyceride                   1249.6700
hdl                             911.1710
ldl                             946.4196
hemoglobin                     1602.6227
serum.creatinine                715.6403
ast                             809.2825
alt                             912.4609
gtp                            1612.4408
tartar                          168.5013
```

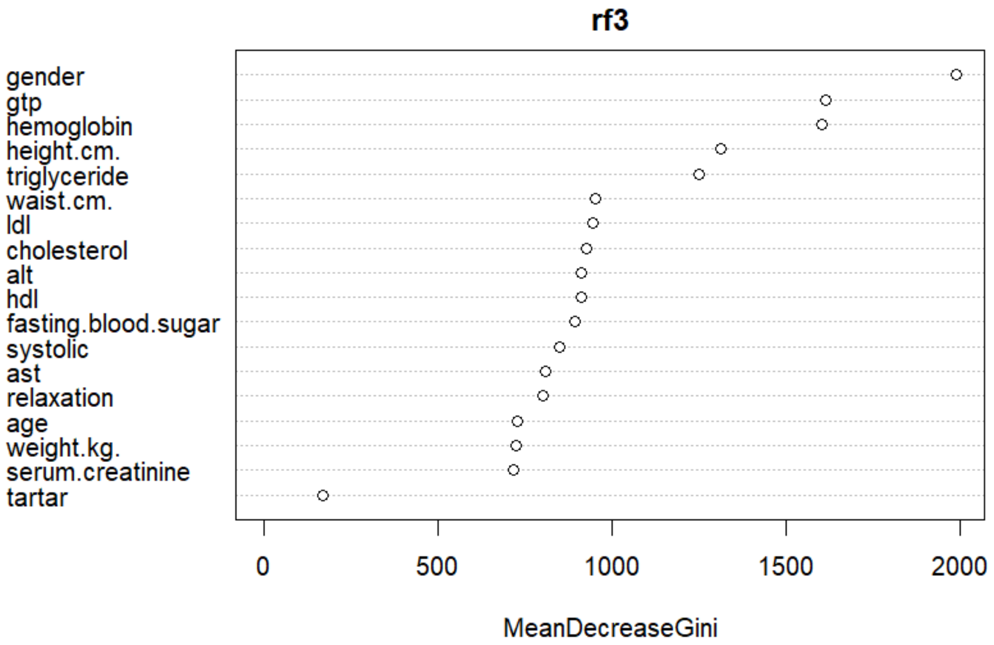


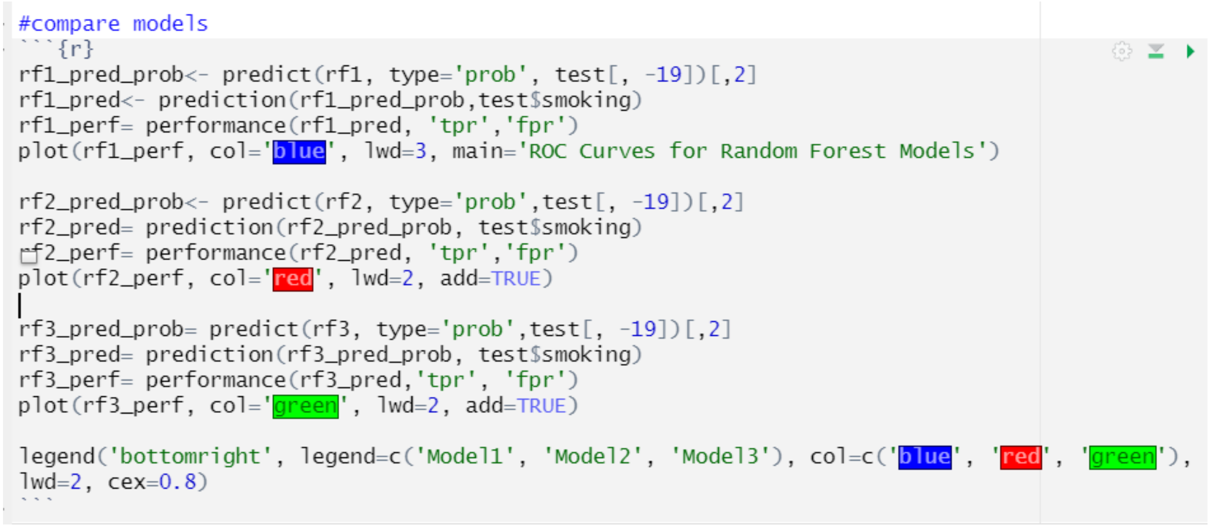
Figure 23 Feature Importance in Random Forest Model 3

The mean decrease in Gini index measures how much each variable contributes to the homogeneity nodes when constructing the random forest. The higher the value of mean decrease Gini score, the higher the importance of the variable in the model. Based on the graph, gender is the most important variable in this model followed by GTP and haemoglobin. The least important variable is tartar.

6.2.5 Summary

Model	OOB error	Data	Accuracy	AUC	Sensitivity	Specificity
Model 1 ntree=500, mtry=4	17.75%	Training	100%	1	100%	100%
		Test	82.31%	0.91	84.75%	78.1%
Model 2 ntree=2000, mtry=4	17.59%	Training	100%	1	100%	100%
		test	82.45%	0.91	84.78%	78.43%
Model 3 -ntree=2000, mtry=3	17.53%	Training	100%	1	100%	100%
		Test	82.55%	0.911814	84.90%	78.51%

Table 1 Comparison of Random Forest Model Performance



~~~
#compare models
```{r}
rf1_pred_prob<- predict(rf1, type='prob', test[, -19])[,2]
rf1_pred<- prediction(rf1_pred_prob,test$smoking)
rf1_perf= performance(rf1_pred, 'tpr','fpr')
plot(rf1_perf, col='blue', lwd=3, main='ROC Curves for Random Forest Models')

rf2_pred_prob<- predict(rf2, type='prob',test[, -19])[,2]
rf2_pred= prediction(rf2_pred_prob, test$smoking)
rf2_perf= performance(rf2_pred, 'tpr','fpr')
plot(rf2_perf, col='red', lwd=2, add=TRUE)

rf3_pred_prob= predict(rf3, type='prob',test[, -19])[,2]
rf3_pred= prediction(rf3_pred_prob, test$smoking)
rf3_perf= performance(rf3_pred,'tpr', 'fpr')
plot(rf3_perf, col='green', lwd=2, add=TRUE)

legend('bottomright', legend=c('Model1', 'Model2', 'Model3'), col=c('blue', 'red', 'green'),
lwd=2, cex=0.8)
```
~~~

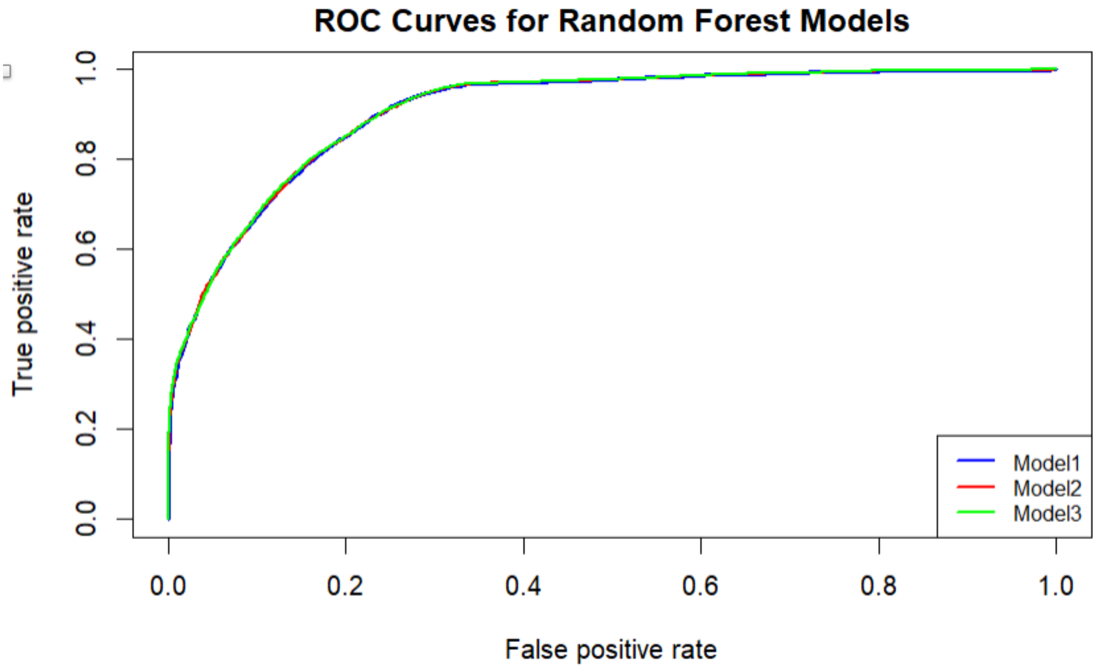
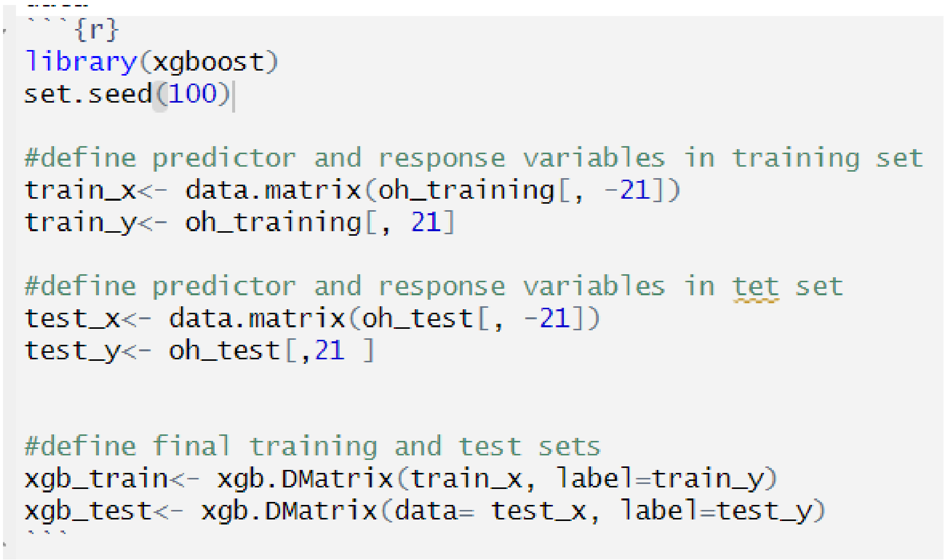


Figure 24 ROC curve of All Random Forest Models

Among the three random forest models, Model 3, tuned with ntree=2000, mtry=3, has the best performance. However, the performance between all three models does not differ much. Overall, for each model, similar trend is observed that the performance between training and the test data does not differ much, suggesting that there is no overfitting issue. Moreover, the performance of Model 3 on training data is in a good range (accuracy: 82.55%, AUC:0.911). Therefore, model 3 is selected as the best Decision Tree model.

6.3 XGBoost

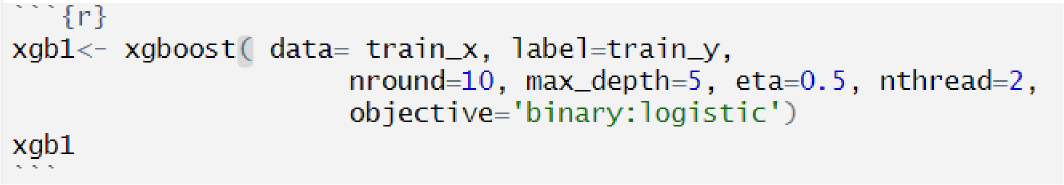
Since XGBoost only works with numeric vectors, the categorical variables were converted into numeric vector through one hot encoding. Besides that, XGBoost uses a specific data structure, known as DMatrix to store data. Both training and test dataset were stored in DMatrix structure.



~~~
```{r}
library(xgboost)
set.seed(100)

#define predictor and response variables in training set
train_x<- data.matrix(oh_training[, -21])
train_y<- oh_training[, 21]

#define predictor and response variables in tet set
test_x<- data.matrix(oh_test[, -21])
test_y<- oh_test[,21 ]


#define final training and test sets
xgb_train<- xgb.DMatrix(train_x, label=train_y)
xgb_test<- xgb.DMatrix(data= test_x, label=test_y)
```
~~~

6.3.1 Model 1

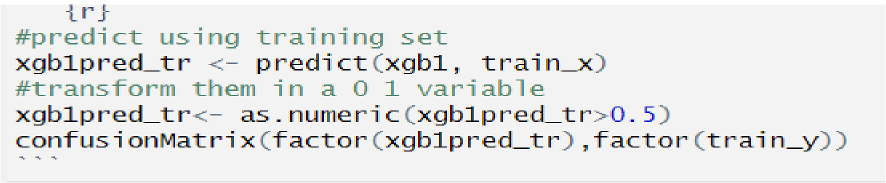
Build basic model.



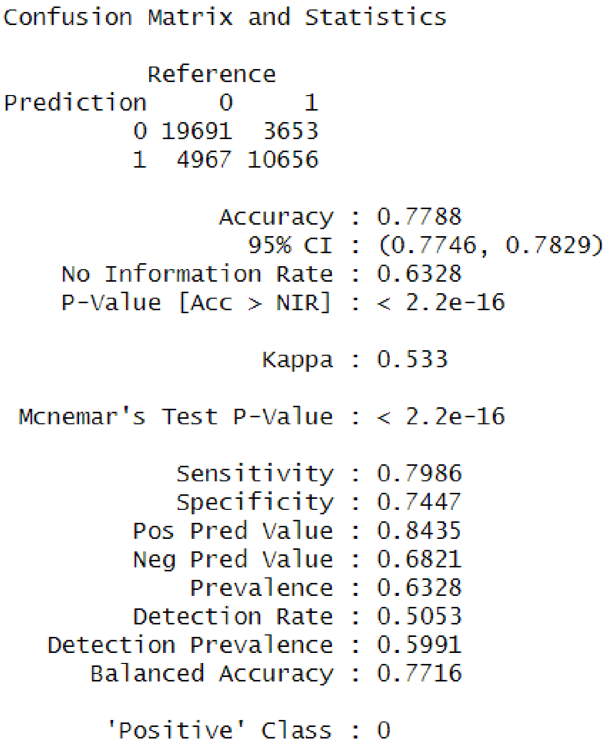
~~~
```{r}
xgb1<- xgboost( data= train_x, label=train_y,
                nround=10, max_depth=5, eta=0.5, nthread=2,
                objective='binary:logistic')
xgb1
```
~~~

Multiple hyperparameters were set in the first XGBoost model. These include the number of rounds for boosting (nround=10), maximum depth of the tree (max_depth=5), step size shrinkage which shrinks the feature weights after each boosting step (eta=0.5), and number of parallel threads used to run the model (nthread=2). Note that high value of max_depth tend to make the more complex and more likely to overfit.

Prediction and Evaluation on Training Set

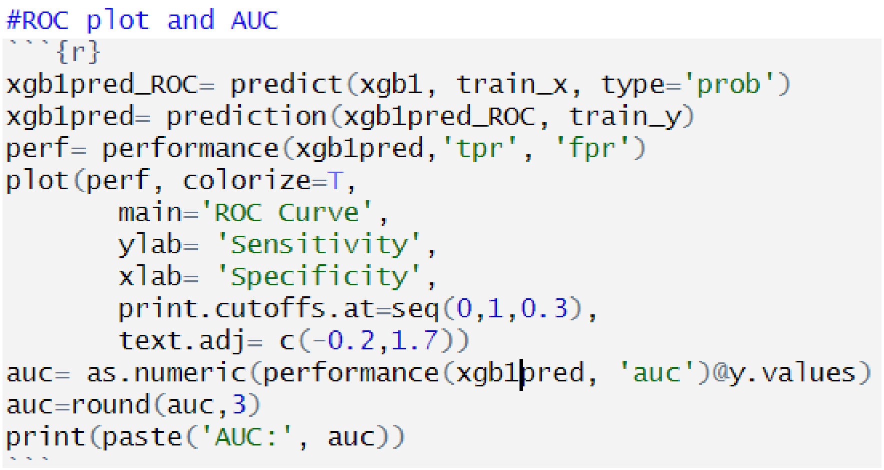


```
    {r}
#predict using training set
xgb1pred_tr <- predict(xgb1, train_x)
#transform them in a 0 1 variable
xgb1pred_tr<- as.numeric(xgb1pred_tr>0.5)
confusionMatrix(factor(xgb1pred_tr),factor(train_y))
```



```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 19691  3653
         1  4967 10656
                                          
               Accuracy : 0.7788          
                 95% CI : (0.7746, 0.7829)
    No Information Rate : 0.6328          
    P-Value [Acc > NIR] : < 2.2e-16       
                                          
                  Kappa : 0.533           
                                          
 Mcnemar's Test P-Value : < 2.2e-16       
                                          
            Sensitivity : 0.7986          
            Specificity : 0.7447          
         Pos Pred Value : 0.8435          
         Neg Pred Value : 0.6821          
             Prevalence : 0.6328          
         Detection Rate : 0.5053          
   Detection Prevalence : 0.5991          
      Balanced Accuracy : 0.7716          
                                          
       'Positive' Class : 0               
```

Prediction using XGBoost model 1 on training data yield 77.88% accuracy, 79.86% sensitivity, 74.47% specificity. From the confusion matrix output, it was observed that out of the total number of observations, the model correctly classifies 30347 cases which gives an accuracy of 77.88%. Specifically, out of all non-smoking cases, 19691 cases were correctly classified, achieving 79.86% sensitivity. On the other hand, out of all smoking cases, 10656 cases were correctly classified, achieving 74.47% specificity. Finally, 4967 non-smoking cases and 3653 smoking cases were misclassified.



~~~
#ROC plot and AUC
```{r}
xgb1pred_ROC= predict(xgb1, train_x, type='prob')
xgb1pred= prediction(xgb1pred_ROC, train_y)
perf= performance(xgb1pred,'tpr', 'fpr')
plot(perf, colorize=T,
       main='ROC Curve',
       ylab= 'Sensitivity',
       xlab= 'Specificity',
       print.cutoffs.at=seq(0,1,0.3),
       text.adj= c(-0.2,1.7))
auc= as.numeric(performance(xgb1pred, 'auc')@y.values)
auc=round(auc,3)
print(paste('AUC:', auc))
```
~~~

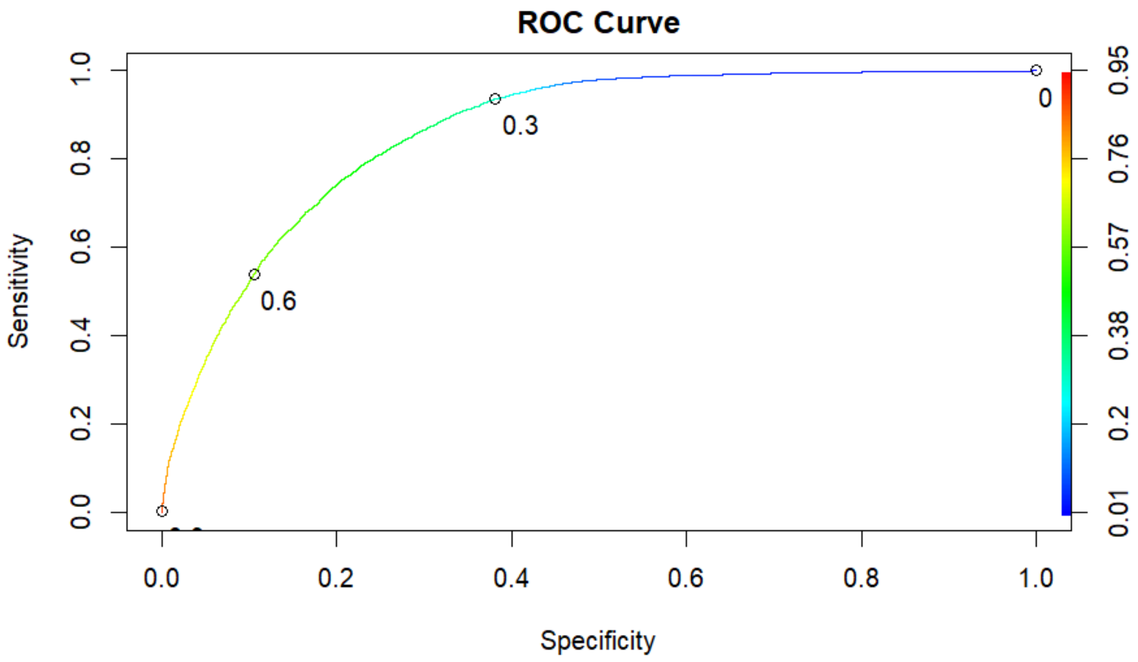
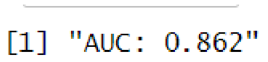


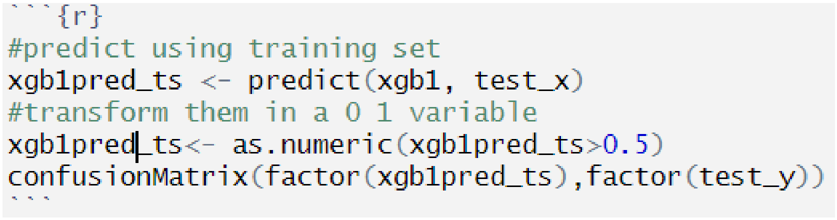
Figure 25 ROC curve of XGBoost Model 1 on Training Data



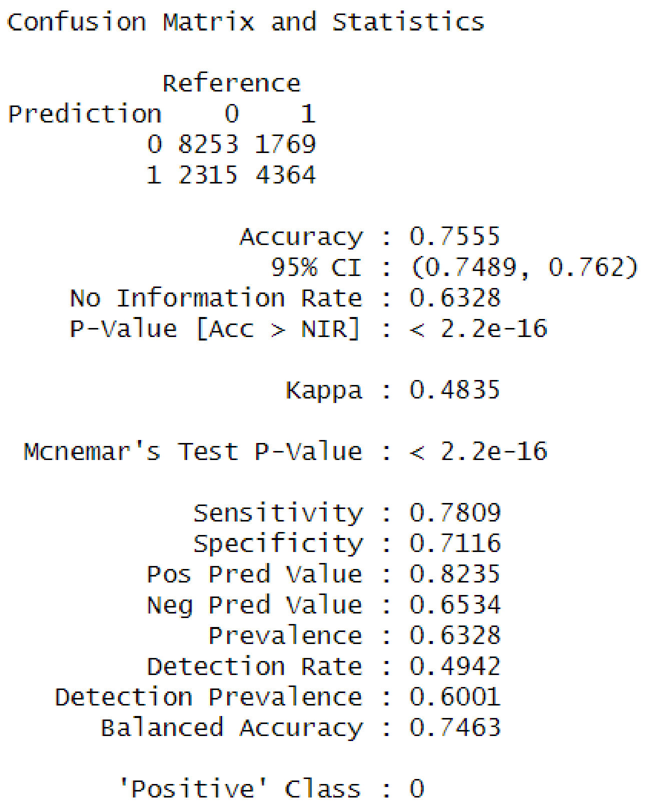
```
[1] "AUC: 0.862"
```

AUC is 0.862, which means that there is 86.2% chance that the model can distinguish non-smoking and smoking cases.

Prediction and Evaluation on Test Set

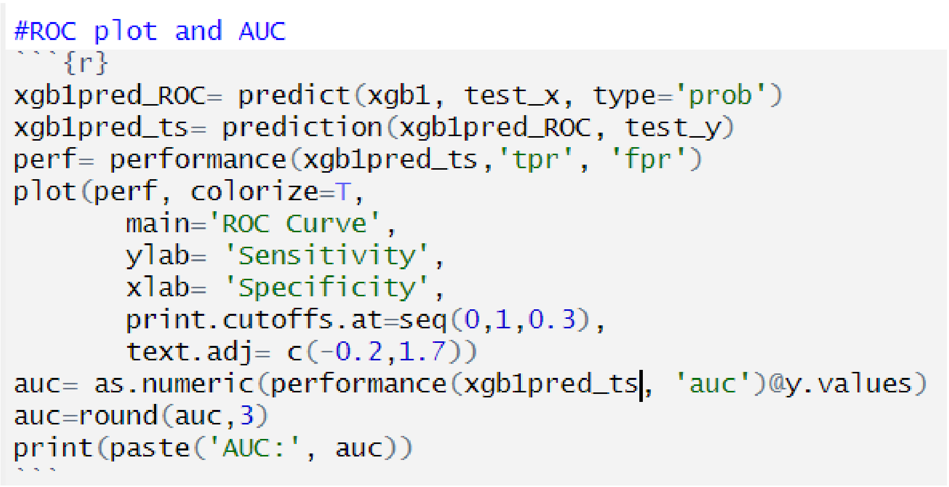


~~~
```{r}
#predict using training set
xgb1pred_ts <- predict(xgb1, test_x)
#transform them in a 0 1 variable
xgb1pred_ts<- as.numeric(xgb1pred_ts>0.5)
confusionMatrix(factor(xgb1pred_ts),factor(test_y))
```
~~~



```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 8253 1769
         1 2315 4364
                                          
               Accuracy : 0.7555          
                 95% CI : (0.7489, 0.762) 
    No Information Rate : 0.6328          
    P-Value [Acc > NIR] : < 2.2e-16       
                                          
                  Kappa : 0.4835          
                                          
 Mcnemar's Test P-Value : < 2.2e-16       
                                          
            Sensitivity : 0.7809          
            Specificity : 0.7116          
         Pos Pred Value : 0.8235          
         Neg Pred Value : 0.6534          
             Prevalence : 0.6328          
         Detection Rate : 0.4942          
   Detection Prevalence : 0.6001          
      Balanced Accuracy : 0.7463          
                                          
       'Positive' Class : 0               
```

Prediction using XGBoost model 1 on test data yield 75.55% accuracy, 78.09% sensitivity, 71.16% specificity. From the confusion matrix output, it was observed that out of the total number of observations, the model correctly classifies 12617 cases which gives an accuracy of 75.55%. Specifically, out of all non-smoking cases, 8253 cases were correctly classified, achieving 78.09% sensitivity. On the other hand, out of all smoking cases, 4364 cases were correctly classified, achieving 71.16% specificity. Finally, 2315 non-smoking cases and 1769 smoking cases were misclassified.



~~~
#ROC plot and AUC
```{r}
xgb1pred_ROC= predict(xgb1, test_x, type='prob')
xgb1pred_ts= prediction(xgb1pred_ROC, test_y)
perf= performance(xgb1pred_ts,'tpr', 'fpr')
plot(perf, colorize=T,
       main='ROC Curve',
       ylab= 'Sensitivity',
       xlab= 'Specificity',
       print.cutoffs.at=seq(0,1,0.3),
       text.adj= c(-0.2,1.7))
auc= as.numeric(performance(xgb1pred_ts, 'auc')@y.values)
auc=round(auc,3)
print(paste('AUC:', auc))
```
~~~

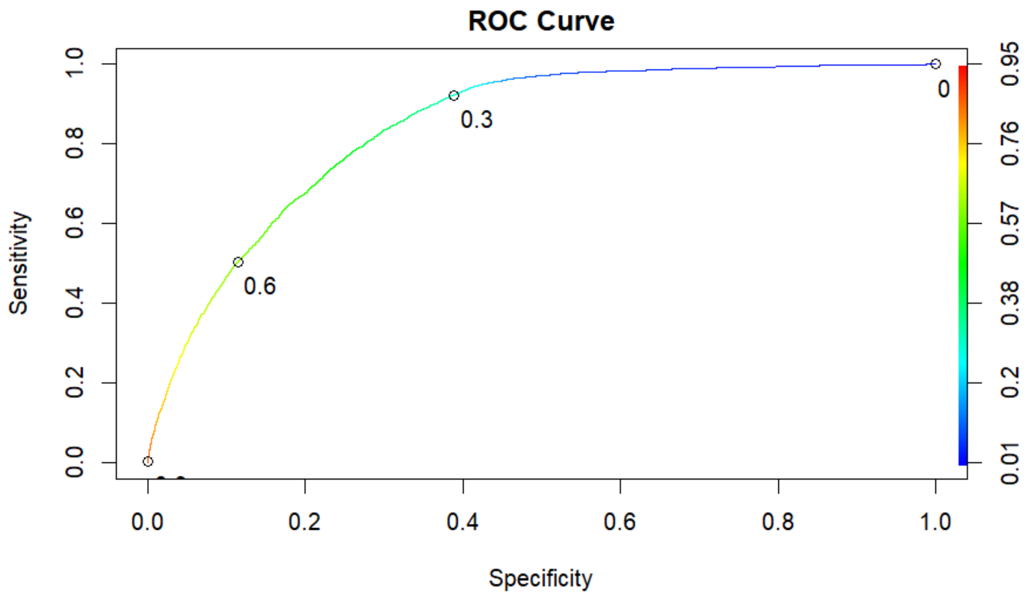
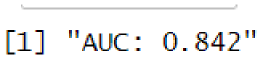


Figure 26 ROC curve of XGBoost Model 1 on Test Data

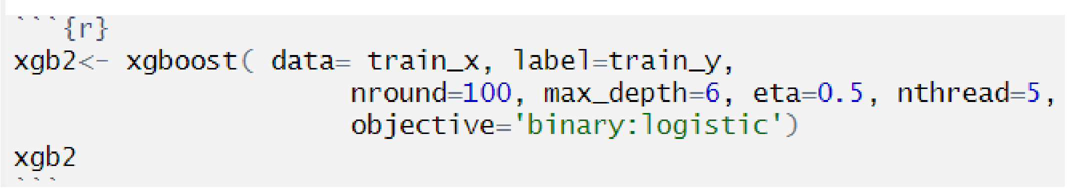


```
[1] "AUC: 0.842"
```

The AUC from model 1 is 0.842. there is a 84.2% chance that the model can distinguish between smoking and non-smoking cases.

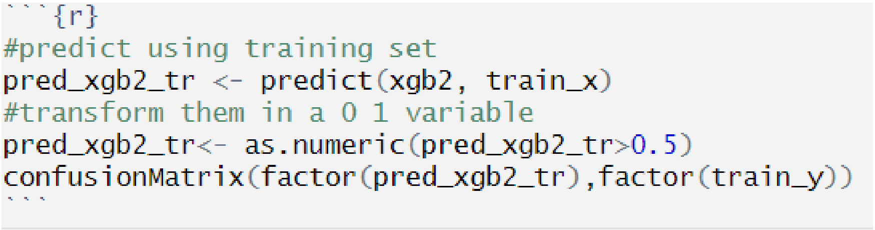
6.3.2 Model 2

In model 2, nround and max_depth values were increased to test on the model performance.

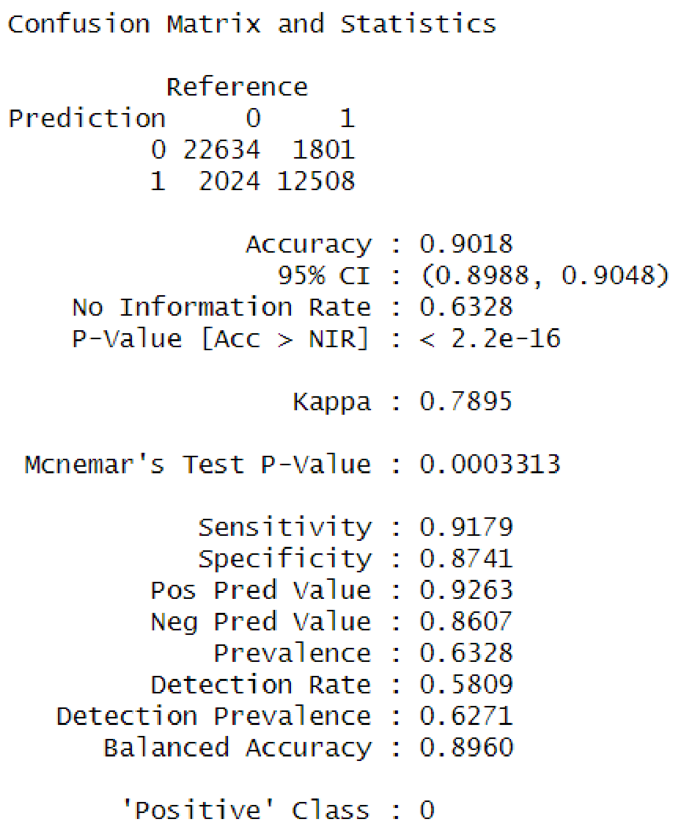


~~~
```{r}
xgb2<- xgboost( data= train_x, label=train_y,
                            nround=100, max_depth=6, eta=0.5, nthread=5,
                            objective='binary:logistic')
xgb2
```
~~~

<u>Prediction and Evaluation on Training Set</u>

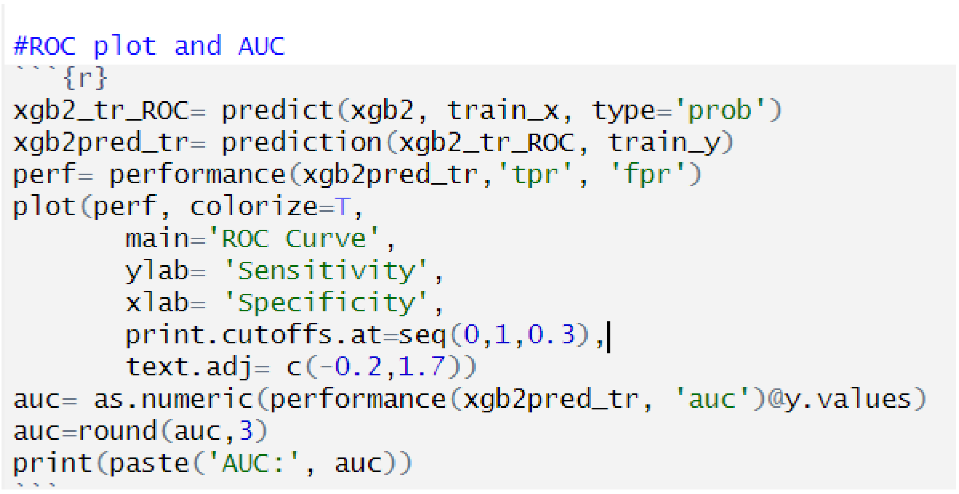


~~~
```{r}
#predict using training set
pred_xgb2_tr <- predict(xgb2, train_x)
#transform them in a 0 1 variable
pred_xgb2_tr<- as.numeric(pred_xgb2_tr>0.5)
confusionMatrix(factor(pred_xgb2_tr),factor(train_y))
```
~~~



```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 22634  1801
         1  2024 12508

               Accuracy : 0.9018
                 95% CI : (0.8988, 0.9048)
    No Information Rate : 0.6328
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.7895

 Mcnemar's Test P-Value : 0.0003313

            Sensitivity : 0.9179
            Specificity : 0.8741
         Pos Pred Value : 0.9263
         Neg Pred Value : 0.8607
             Prevalence : 0.6328
         Detection Rate : 0.5809
   Detection Prevalence : 0.6271
      Balanced Accuracy : 0.8960

       'Positive' Class : 0
```

Prediction using XGBoost model 2 on training data yield 90.18% accuracy, 91.79% sensitivity, 87.41% specificity. From the confusion matrix output, it was observed that out of the total number of observations, the model correctly classifies 35142 cases which gives an accuracy of 90.18%. Specifically, out of all non-smoking cases, 22634 cases were correctly classified, achieving 91.79% sensitivity. On the other hand, out of all smoking cases, 12508 cases were correctly classified, achieving 87.41% specificity. Finally, 2024 non-smoking cases and 1801 smoking cases were misclassified.



~~~
#ROC plot and AUC
```{r}
xgb2_tr_ROC= predict(xgb2, train_x, type='prob')
xgb2pred_tr= prediction(xgb2_tr_ROC, train_y)
perf= performance(xgb2pred_tr,'tpr', 'fpr')
plot(perf, colorize=T,
       main='ROC Curve',
       ylab= 'Sensitivity',
       xlab= 'Specificity',
       print.cutoffs.at=seq(0,1,0.3),
       text.adj= c(-0.2,1.7))
auc= as.numeric(performance(xgb2pred_tr, 'auc')@y.values)
auc=round(auc,3)
print(paste('AUC:', auc))
```
~~~

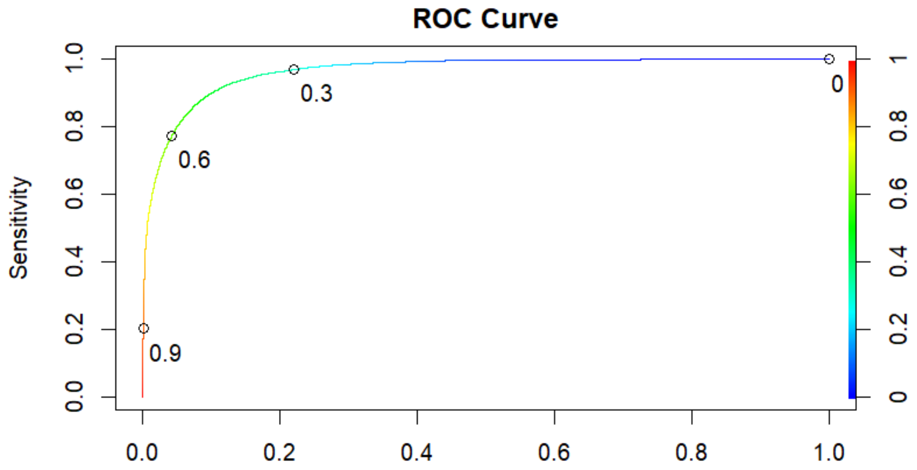
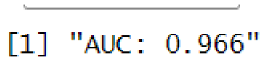


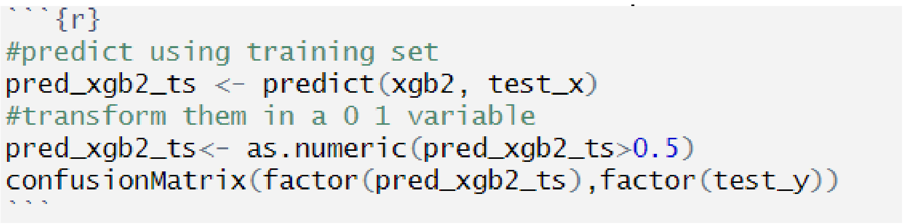
Figure 27 ROC Curve of XGBoost Model 2 on Training Data



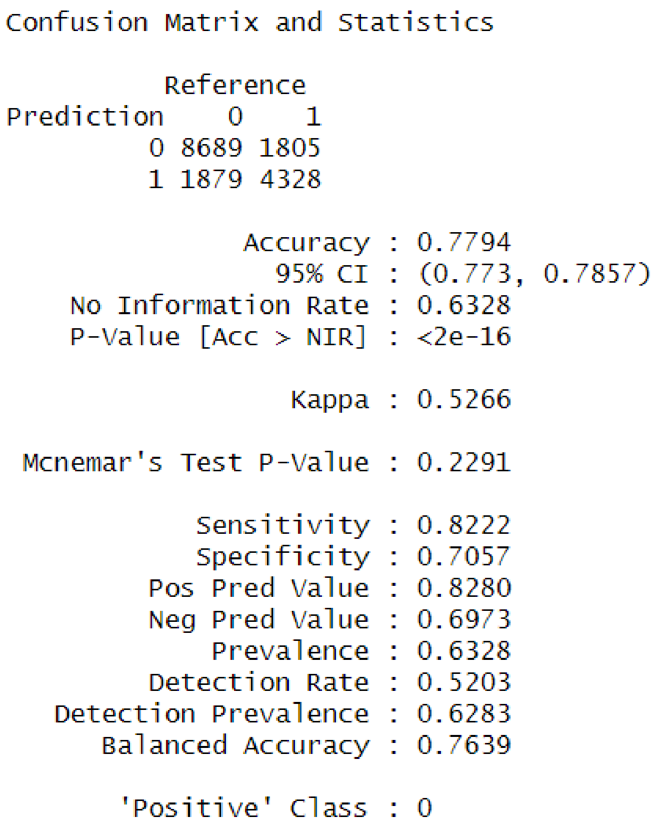
```
[1] "AUC: 0.966"
```

The AUC from model 2 on training data is 0.966, there is a 96.6% chance that the model can distinguish between smoking and non-smoking cases.

Prediction and Evaluation on Test Set

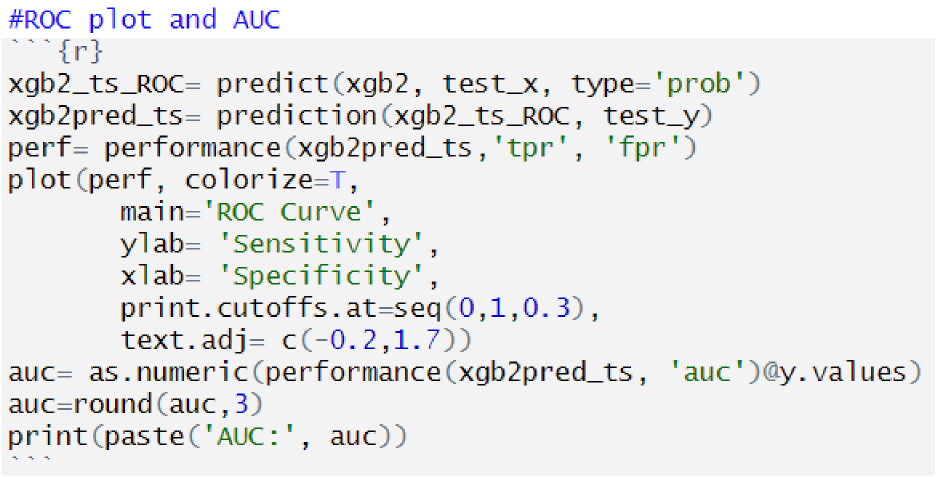


~~~
```{r}
#predict using training set
pred_xgb2_ts <- predict(xgb2, test_x)
#transform them in a 0 1 variable
pred_xgb2_ts<- as.numeric(pred_xgb2_ts>0.5)
confusionMatrix(factor(pred_xgb2_ts),factor(test_y))
```
~~~



```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 8689 1805
         1 1879 4328

               Accuracy : 0.7794
                 95% CI : (0.773, 0.7857)
    No Information Rate : 0.6328
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.5266

 Mcnemar's Test P-Value : 0.2291

            Sensitivity : 0.8222
            Specificity : 0.7057
         Pos Pred Value : 0.8280
         Neg Pred Value : 0.6973
             Prevalence : 0.6328
         Detection Rate : 0.5203
   Detection Prevalence : 0.6283
      Balanced Accuracy : 0.7639

       'Positive' Class : 0
```

Prediction using XGBoost model 2 on test data yield 77.94% accuracy, 82.22% sensitivity, 70.57% specificity. From the confusion matrix output, it was observed that out of the total number of observations, the model correctly classifies 13017 cases which gives an accuracy of 77.94%. Specifically, out of all non-smoking cases, 8689 cases were correctly classified, achieving 82.22% sensitivity. On the other hand, out of all smoking cases, 4328 cases were correctly classified, achieving 70.57% specificity. Finally, 1879 non-smoking cases and 1805 smoking cases were misclassified.



~~~
#ROC plot and AUC
```{r}
xgb2_ts_ROC= predict(xgb2, test_x, type='prob')
xgb2pred_ts= prediction(xgb2_ts_ROC, test_y)
perf= performance(xgb2pred_ts,'tpr', 'fpr')
plot(perf, colorize=T,
       main='ROC Curve',
       ylab= 'Sensitivity',
       xlab= 'Specificity',
       print.cutoffs.at=seq(0,1,0.3),
       text.adj= c(-0.2,1.7))
auc= as.numeric(performance(xgb2pred_ts, 'auc')@y.values)
auc=round(auc,3)
print(paste('AUC:', auc))
```
~~~

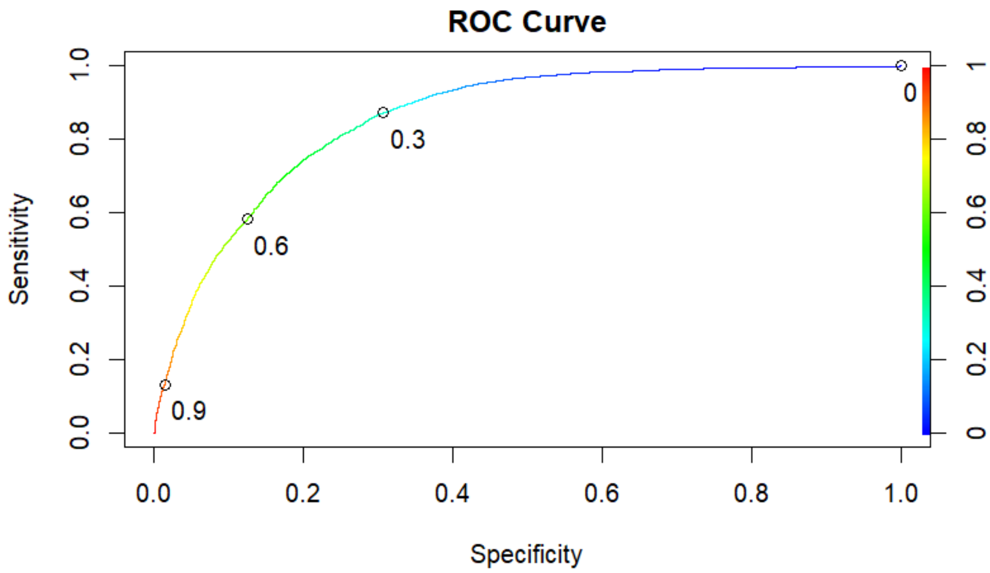
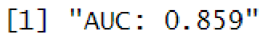


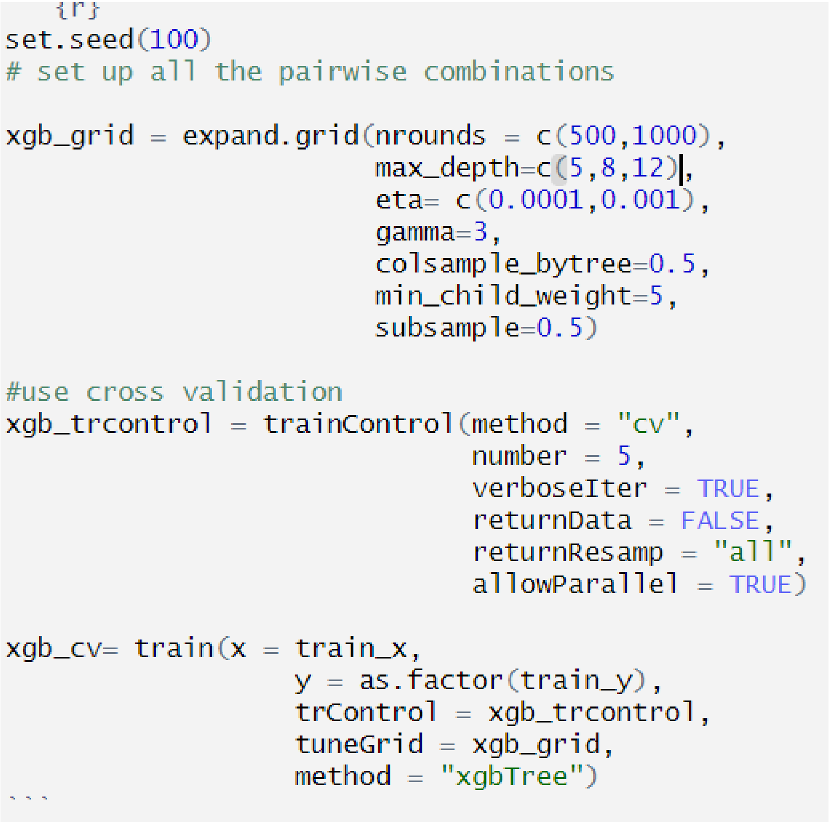
Figure 28 ROC curve of XGBoost Model 2 on Test Data



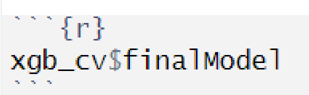
```
[1] "AUC: 0.859"
```

The AUC from model 2 on test data is 0.859, there is a 85.9% chance that the model can distinguish between smoking and non-smoking cases.

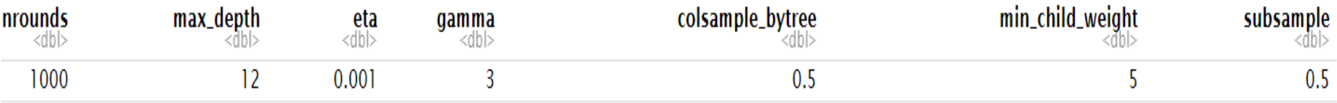
6.3.3 Hyperparameter Tuning through Cross Validation



```
   {r}
set.seed(100)
# set up all the pairwise combinations

xgb_grid = expand.grid(nrounds = c(500,1000),
                       max_depth=c(5,8,12),
                       eta= c(0.0001,0.001),
                       gamma=3,
                       colsample_bytree=0.5,
                       min_child_weight=5,
                       subsample=0.5)

#use cross validation
xgb_trcontrol = trainControl(method = "cv",
                             number = 5,
                             verboseIter = TRUE,
                             returnData = FALSE,
                             returnResamp = "all",
                             allowParallel = TRUE)

xgb_cv= train(x = train_x,
                  y = as.factor(train_y),
                  trControl = xgb_trcontrol,
                  tuneGrid = xgb_grid,
                  method = "xgbTree")
```
```
```



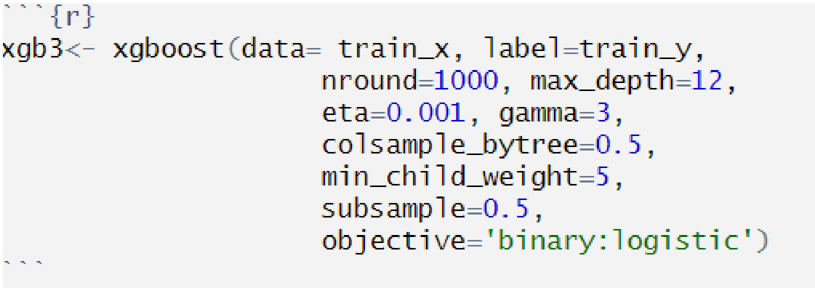
~~~
```{r}
xgb_cv$finalModel
```
~~~



nrounds <dbl>	max_depth <dbl>	eta <dbl>	gamma <dbl>	colsample_bytree <dbl>	min_child_weight <dbl>	subsample <dbl>
1000	12	0.001	3	0.5	5	0.5

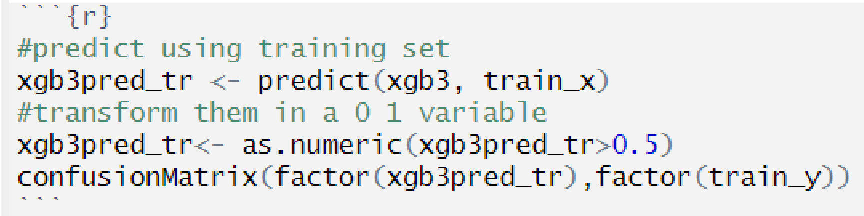
The final model hyperparameters will be used to build model 3.

6.3.4 Model 3

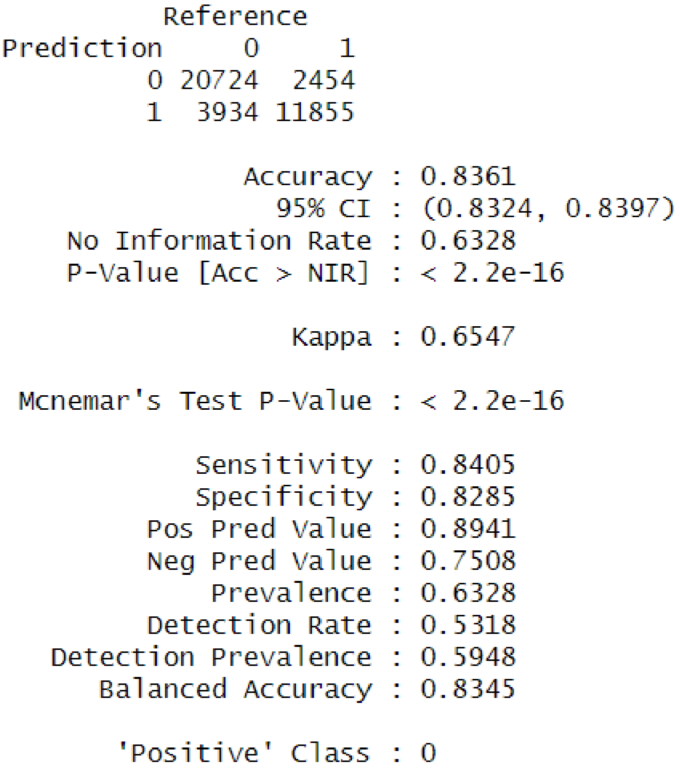


~~~
```{r}
xgb3<- xgboost(data= train_x, label=train_y,
                    nround=1000, max_depth=12,
                    eta=0.001, gamma=3,
                    colsample_bytree=0.5,
                    min_child_weight=5,
                    subsample=0.5,
                    objective='binary:logistic')
```
~~~

Prediction and Evaluation on Training Set

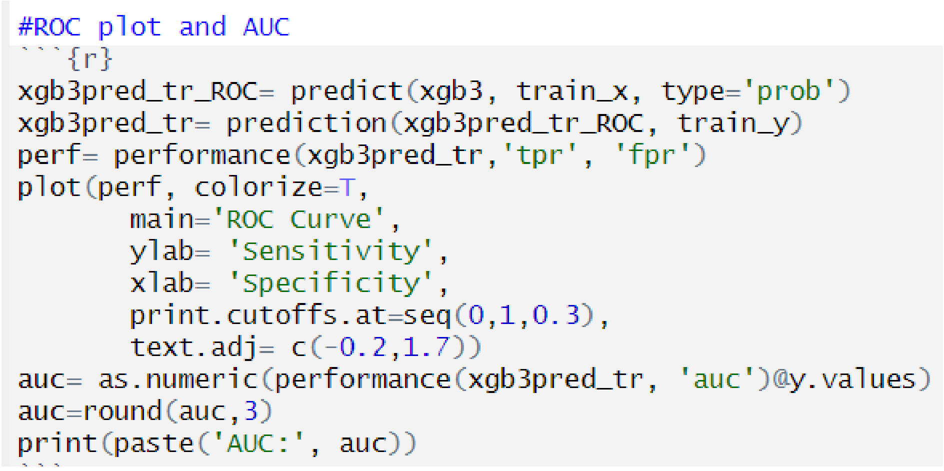


~~~
```{r}
#predict using training set
xgb3pred_tr <- predict(xgb3, train_x)
#transform them in a 0 1 variable
xgb3pred_tr<- as.numeric(xgb3pred_tr>0.5)
confusionMatrix(factor(xgb3pred_tr),factor(train_y))
```
~~~



```
          Reference
Prediction     0     1
         0 20724  2454
         1  3934 11855
                                          
               Accuracy : 0.8361          
                 95% CI : (0.8324, 0.8397)
    No Information Rate : 0.6328          
    P-Value [Acc > NIR] : < 2.2e-16       
                                          
                  Kappa : 0.6547          
                                          
 Mcnemar's Test P-Value : < 2.2e-16       
                                          
            Sensitivity : 0.8405          
            Specificity : 0.8285          
         Pos Pred Value : 0.8941          
         Neg Pred Value : 0.7508          
             Prevalence : 0.6328          
         Detection Rate : 0.5318          
   Detection Prevalence : 0.5948          
      Balanced Accuracy : 0.8345          
                                          
       'Positive' Class : 0               
```

Prediction using XGBoost model 3 on training data yield 83.61% accuracy, 84.05% sensitivity, 82.85% specificity. From the confusion matrix output, it was observed that out of the total number of observations, the model correctly classifies 32579 cases which gives an accuracy of 83.61%. Specifically, out of all non-smoking cases, 20724 cases were correctly classified, achieving 84.05% sensitivity. On the other hand, out of all smoking cases, 11855 cases were correctly classified, achieving 82.85% specificity. Finally, 3934 non-smoking cases and 2454 smoking cases were misclassified.



~~~
#ROC plot and AUC
```{r}
xgb3pred_tr_ROC= predict(xgb3, train_x, type='prob')
xgb3pred_tr= prediction(xgb3pred_tr_ROC, train_y)
perf= performance(xgb3pred_tr,'tpr', 'fpr')
plot(perf, colorize=T,
       main='ROC Curve',
       ylab= 'Sensitivity',
       xlab= 'Specificity',
       print.cutoffs.at=seq(0,1,0.3),
       text.adj= c(-0.2,1.7))
auc= as.numeric(performance(xgb3pred_tr, 'auc')@y.values)
auc=round(auc,3)
print(paste('AUC:', auc))
```
~~~

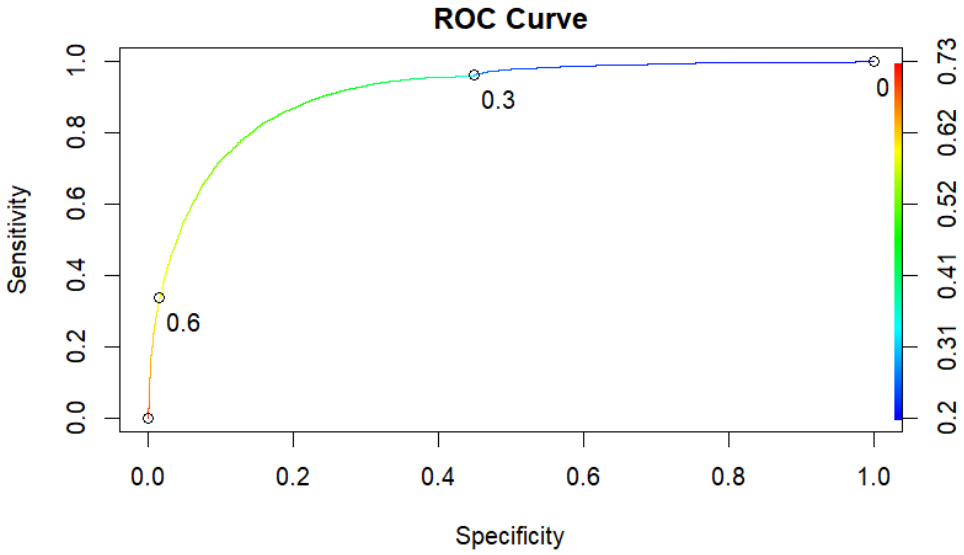
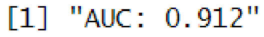


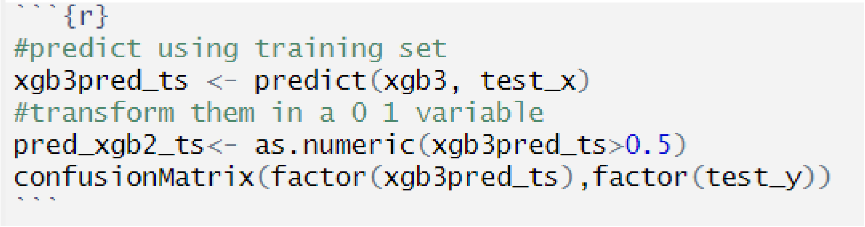
Figure 29 ROC Curve of XGBoost Model 3 on Training Data



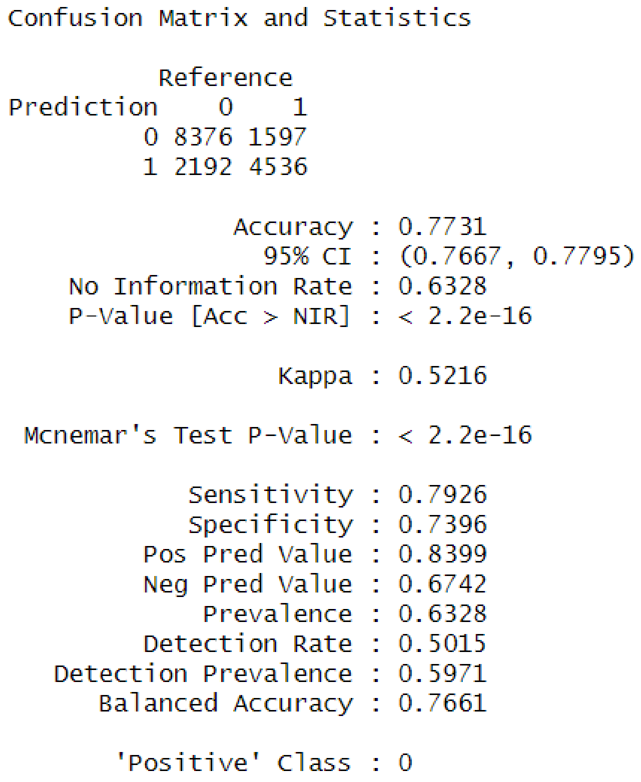
```
[1] "AUC: 0.912"
```

The AUC from model 3 on training data is 0.912, there is a 91.2% chance that the model can distinguish between smoking and non-smoking cases.

Prediction and Evaluation on Test Set

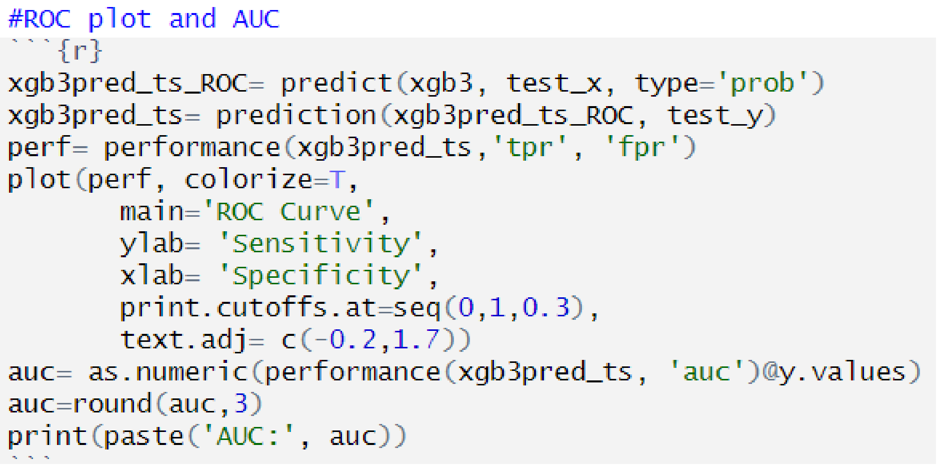


~~~
```{r}
#predict using training set
xgb3pred_ts <- predict(xgb3, test_x)
#transform them in a 0 1 variable
pred_xgb2_ts<- as.numeric(xgb3pred_ts>0.5)
confusionMatrix(factor(xgb3pred_ts),factor(test_y))
```
~~~



```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 8376 1597
         1 2192 4536
                                          
               Accuracy : 0.7731          
                 95% CI : (0.7667, 0.7795)
    No Information Rate : 0.6328          
    P-Value [Acc > NIR] : < 2.2e-16       
                                          
                  Kappa : 0.5216          
                                          
 Mcnemar's Test P-Value : < 2.2e-16       
                                          
            Sensitivity : 0.7926          
            Specificity : 0.7396          
         Pos Pred Value : 0.8399          
         Neg Pred Value : 0.6742          
             Prevalence : 0.6328          
         Detection Rate : 0.5015          
   Detection Prevalence : 0.5971          
      Balanced Accuracy : 0.7661          
                                          
       'Positive' Class : 0               
```

Prediction using XGBoost model 3 on test data yield 77.31% accuracy, 79.26% sensitivity, 73.96% specificity. From the confusion matrix output, it was observed that out of the total number of observations, the model correctly classifies 12912 cases which gives an accuracy of 77.31%. Specifically, out of all non-smoking cases, 8376 cases were correctly classified, achieving 79.26% sensitivity. On the other hand, out of all smoking cases, 4536 cases were correctly classified, achieving 73.96% specificity. Finally, 2192 non-smoking cases and 1597 smoking cases were misclassified.



~~~~
#ROC plot and AUC
```{r}
xgb3pred_ts_ROC= predict(xgb3, test_x, type='prob')
xgb3pred_ts= prediction(xgb3pred_ts_ROC, test_y)
perf= performance(xgb3pred_ts,'tpr', 'fpr')
plot(perf, colorize=T,
       main='ROC Curve',
       ylab= 'Sensitivity',
       xlab= 'Specificity',
       print.cutoffs.at=seq(0,1,0.3),
       text.adj= c(-0.2,1.7))
auc= as.numeric(performance(xgb3pred_ts, 'auc')@y.values)
auc=round(auc,3)
print(paste('AUC:', auc))
```
~~~~

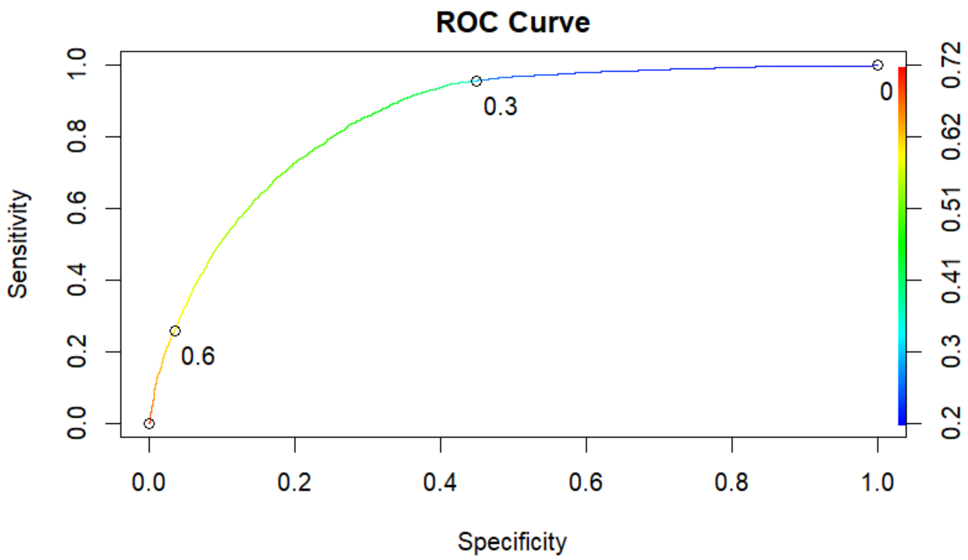
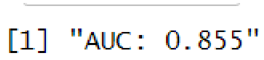


Figure 30 ROC Curve of XGBoost Model 3 on Test Data



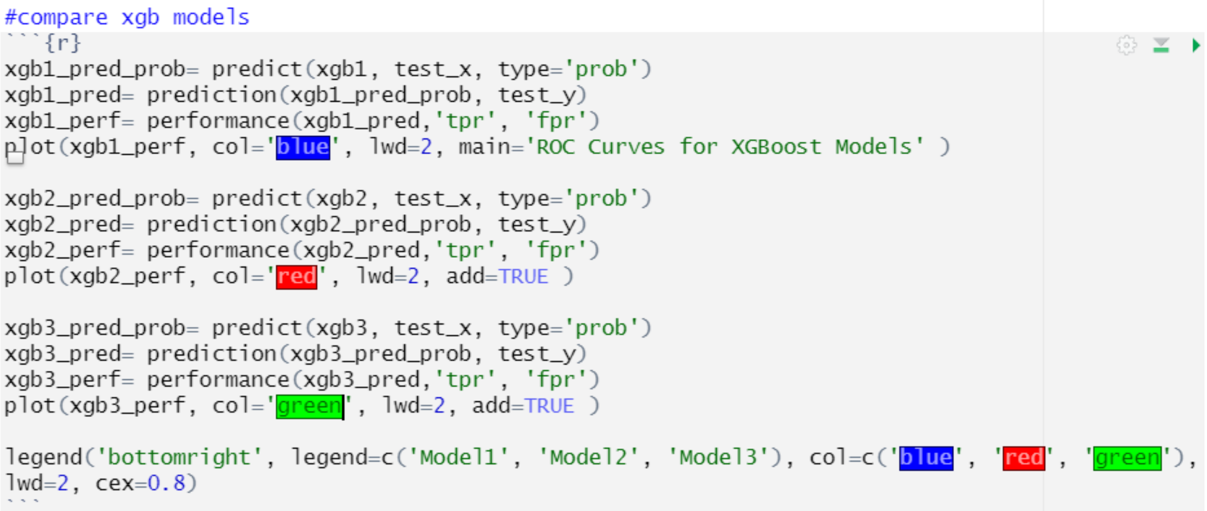
```
[1] "AUC: 0.855"
```

The AUC from model 3 on training data is 0.855, there is a 85.5% chance that the model can distinguish between smoking and non-smoking cases.

6.3.5 Summary

Model	Data	Accuracy	AUC	Sensitivity	Specificity
Model 1 Nround=10, max_depth=5, eat=0.5, nthread=2	Training	77.88%	0.862	79.86%	74.47%
	Test	75.55%	0.842	78.09%	71.16%
Model 2 Nround=100, max_depth=6, eta=0.5, nthread=5,	Training	90.18%	0.966	91.79%	87.41%
	Test	77.94%	0.859	82.22%	70.57%
Model 3 -nround=1000, max_depth=12, eta=0.001, gamma=3, colsample_bytree=0.5, min_child_weight=5, subsample=0.5	Training	83.61%	0.912	84.05%	82.85%
	Test	77.31%	0.855	79.26%	73.96%

Table 2 XGBoost Model Performance Comparison



~~~
#compare xgb models
```{r}
xgb1_pred_prob= predict(xgb1, test_x, type='prob')
xgb1_pred= prediction(xgb1_pred_prob, test_y)
xgb1_perf= performance(xgb1_pred,'tpr', 'fpr')
plot(xgb1_perf, col='blue', lwd=2, main='ROC Curves for XGBoost Models' )

xgb2_pred_prob= predict(xgb2, test_x, type='prob')
xgb2_pred= prediction(xgb2_pred_prob, test_y)
xgb2_perf= performance(xgb2_pred,'tpr', 'fpr')
plot(xgb2_perf, col='red', lwd=2, add=TRUE )

xgb3_pred_prob= predict(xgb3, test_x, type='prob')
xgb3_pred= prediction(xgb3_pred_prob, test_y)
xgb3_perf= performance(xgb3_pred,'tpr', 'fpr')
plot(xgb3_perf, col='green', lwd=2, add=TRUE )

legend('bottomright', legend=c('Model1', 'Model2', 'Model3'), col=c('blue', 'red', 'green'),
lwd=2, cex=0.8)
```
~~~

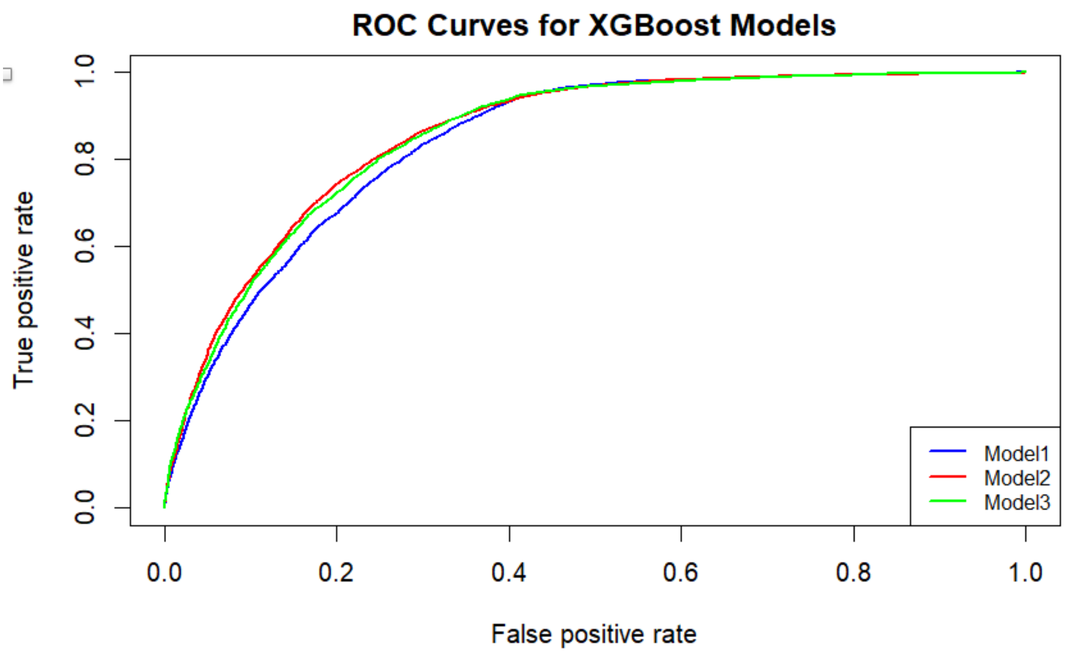


Figure 31 ROC Curve of XGBoost Models

From the ROC curve, it is found that model 2 and 3 has a better performance than model 1. Both curves are closer towards the upper left corner than model 1, where false positive rate is low and true positive rate is high. In Model 2, although the accuracy is slightly higher. However, the gap between the accuracy on the training set and the test set is slightly larger compared to Model 3. Hence, it is suspected that there is overfitting. Therefore, model 3 is selected to be the best XGBoost model with 77.31% accuracy and 0.855 AUC, which is considered in the good range.