ADS 503 Project

Gabriel Duffy

```
suppressMessages({
  library(caret)
  library(pROC)
})
```

```
Warning: package 'caret' was built under R version 4.3.3
```

```
Warning: package 'lattice' was built under R version 4.3.3
```

```
Warning: package 'pROC' was built under R version 4.3.3
```

```
options(repos = c(CRAN = "https://cran.rstudio.com"))
```

```
#read and clean csv file
met_data <- read.csv("C:/Users/gabed/OneDrive/Documents/Metabolic Syndrome.csv")

# Remove rows with any missing values
met_data_clean <- na.omit(met_data)

# Convert target variable to factor
met_data_clean$MetabolicSyndrome <- as.factor(met_data_clean$MetabolicSyndrome)

# Log transform skewed variables
met_data_clean$LogBloodGlucose <- log(met_data_clean$BloodGlucose)
met_data_clean$LogTriglycerides <- log(met_data_clean$Triglycerides)
```

```
# Split data (80/20)
set.seed(123)
split_index <- createDataPartition(met_data_clean$MetabolicSyndrome, p = 0.8, list = FALSE)
train_data <- met_data_clean[split_index, ]
```

```r
test_data  <- met_data_clean[-split_index, ]
# Random Forest Model using 6 selected predictors
library(randomForest)
```

```
Warning: package 'randomForest' was built under R version 4.3.3
```

```
randomForest 4.7-1.2
```

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

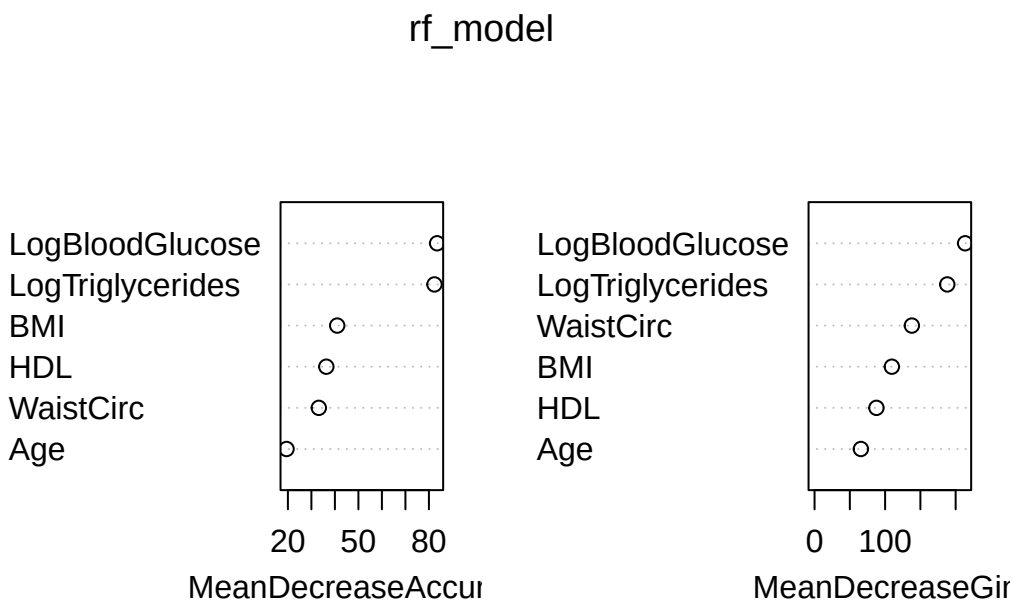
```
Attaching package: 'randomForest'
```

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

    margin
```

```r
rf_model <- randomForest(
  MetabolicSyndrome ~ BMI + Age + LogBloodGlucose + HDL + WaistCirc + LogTriglycerides,
  data = train_data,
  ntree = 500,
  mtry = 2,
  importance = TRUE
)

# Print the model summary
print(rf_model)
```

```
Call:
 randomForest(formula = MetabolicSyndrome ~ BMI + Age + LogBloodGlucose +      HDL + WaistCi
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 2

        OOB estimate of  error rate: 12.6%
Confusion matrix:
     0   1 class.error
0 1039 105  0.09178322
1  117 501  0.18932039
```

```r
# Plot variable importance
varImpPlot(rf_model)
```




```r
# Predict on test set
rf_preds <- predict(rf_model, newdata = test_data)

# Confusion Matrix
conf_matrix_rf <- confusionMatrix(
  rf_preds,
  test_data$MetabolicSyndrome,
  positive = "1"
)
print(conf_matrix_rf)
```

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 249  27
         1  36 127

               Accuracy : 0.8565
```

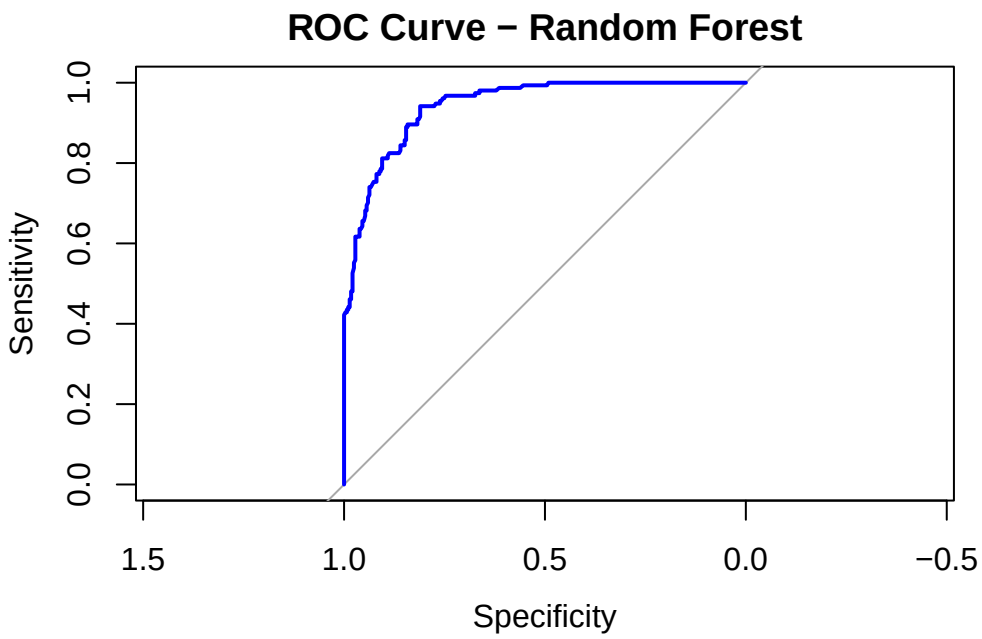
```
                 95% CI : (0.8202, 0.8879)
    No Information Rate : 0.6492
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.6891

 Mcnemar's Test P-Value : 0.3135

            Sensitivity : 0.8247
            Specificity : 0.8737
         Pos Pred Value : 0.7791
         Neg Pred Value : 0.9022
             Prevalence : 0.3508
         Detection Rate : 0.2893
   Detection Prevalence : 0.3713
      Balanced Accuracy : 0.8492

       'Positive' Class : 1
```

```r
# ROC Curve and AUC
rf_probs <- predict(rf_model, newdata = test_data, type = "prob")[, 2]
roc_rf <- roc(test_data$MetabolicSyndrome, rf_probs)
```

```
Setting levels: control = 0, case = 1
```

```
Setting direction: controls < cases
```

```r
# Plot ROC
plot(roc_rf, main = "ROC Curve - Random Forest", col = "blue", lwd = 2)
```




```
# Print AUC
auc(roc_rf)
```

```
Area under the curve: 0.9434
```

```
# Convert MetabolicSyndrome to labeled factor in both train and test sets
train_data$MetabolicSyndrome <- factor(
  train_data$MetabolicSyndrome,
  levels = c(0, 1),
  labels = c("No", "Yes")
)

test_data$MetabolicSyndrome <- factor(
  test_data$MetabolicSyndrome,
  levels = c(0, 1),
  labels = c("No", "Yes")
)
```

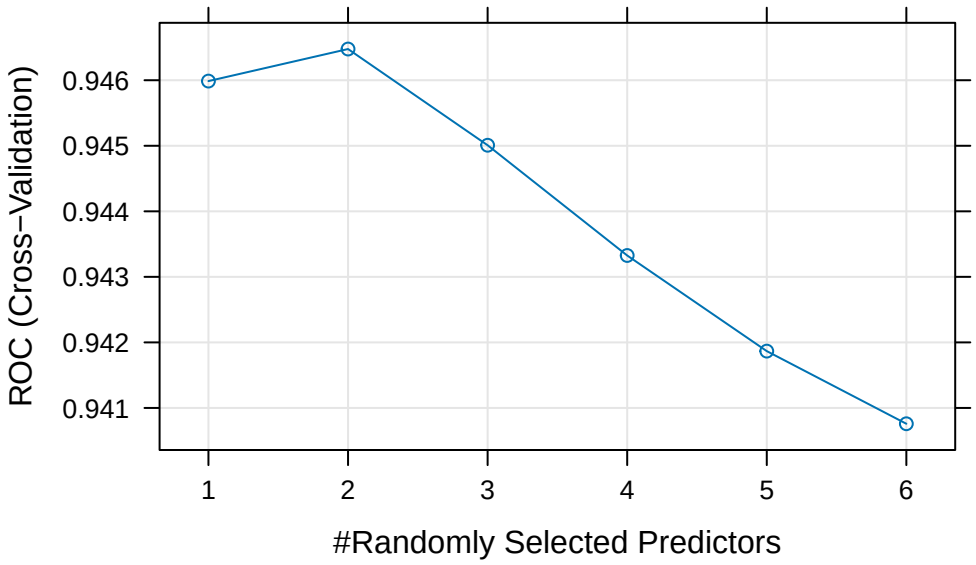
```
# Set tuning grid for mtry values
tune_grid <- expand.grid(mtry = c(1, 2, 3, 4, 5, 6))

# Define 5-fold cross-validation strategy with ROC as the metric
```

```r
ctrl <- trainControl(
  method = "cv",
  number = 5,
  classProbs = TRUE,
  summaryFunction = twoClassSummary,
  savePredictions = TRUE
)

# Train Random Forest model with tuning
set.seed(123)
rf_tuned <- train(
  MetabolicSyndrome ~ BMI + Age + LogBloodGlucose + HDL + WaistCirc + LogTriglycerides,
  data = train_data,
  method = "rf",
  trControl = ctrl,
  tuneGrid = tune_grid,
  metric = "ROC"
)

# Display best model summary and tuning plot
print(rf_tuned)
```

```
Random Forest

1762 samples
   6 predictor
   2 classes: 'No', 'Yes'

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 1410, 1409, 1409, 1411, 1409
Resampling results across tuning parameters:

  mtry  ROC        Sens       Spec
  1     0.9459872  0.9108098  0.7750590
  2     0.9464760  0.9029342  0.7912667
  3     0.9450083  0.9003218  0.8009966
  4     0.9433262  0.9003218  0.8042355
  5     0.9418668  0.9003179  0.7993706
  6     0.9407586  0.8959473  0.7961317

ROC was used to select the optimal model using the largest value.
```

```
The final value used for the model was mtry = 2.
```

```r
plot(rf_tuned)
```




```r
# Final Evaluation on test data
rf_preds <- predict(rf_tuned, newdata = test_data)
rf_probs <- predict(rf_tuned, newdata = test_data, type = "prob")

# Confusion matrix and metrics
conf_rf <- confusionMatrix(
  rf_preds,
  test_data$MetabolicSyndrome,
  positive = "Yes"
)
print(conf_rf)
```

```
Confusion Matrix and Statistics

          Reference
Prediction  No Yes
       No  249  27
       Yes  36 127
```

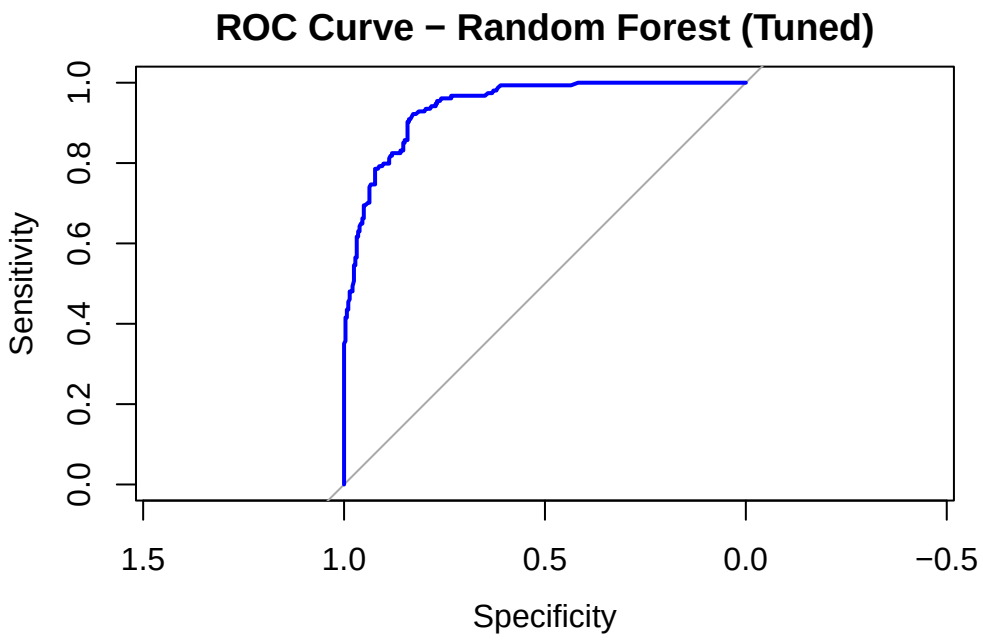
```
               Accuracy : 0.8565
                 95% CI : (0.8202, 0.8879)
    No Information Rate : 0.6492
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.6891

 Mcnemar's Test P-Value : 0.3135

            Sensitivity : 0.8247
            Specificity : 0.8737
         Pos Pred Value : 0.7791
         Neg Pred Value : 0.9022
             Prevalence : 0.3508
         Detection Rate : 0.2893
   Detection Prevalence : 0.3713
      Balanced Accuracy : 0.8492

       'Positive' Class : Yes
```

```r
# ROC Curve and AUC
library(pROC)

rf_roc <- roc(
  response = test_data$MetabolicSyndrome,
  predictor = rf_probs$Yes
)
```

```
Setting levels: control = No, case = Yes
```

```
Setting direction: controls < cases
```

```r
plot(rf_roc, main = "ROC Curve - Random Forest (Tuned)", col = "blue", lwd = 2)
```




```
auc(rf_roc)
```

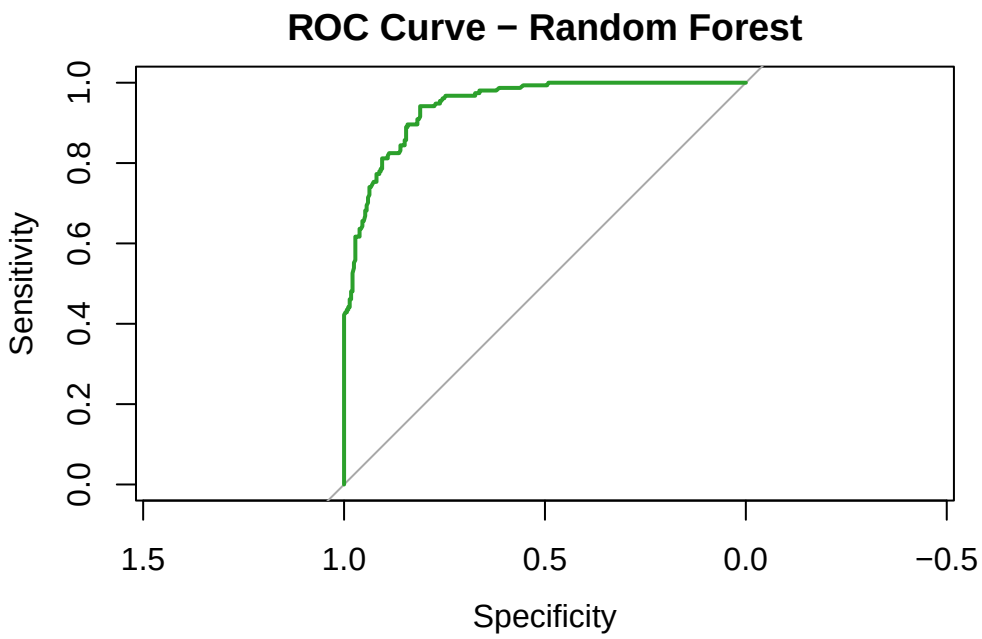
```
Area under the curve: 0.9428
```

```
#ROC Curve + AUC comparison

# Predicted probabilities for class 1 ("Yes")
rf_probs <- predict(rf_model, newdata = test_data, type = "prob")[, 2]

# Plot ROC curve
rf_roc <- roc(test_data$MetabolicSyndrome, rf_probs)
```

```
Setting levels: control = No, case = Yes
```

```
Setting direction: controls < cases
```

```
plot(
  rf_roc,
  col = "#2ca02c",  # green color
  lwd = 2,
  main = "ROC Curve - Random Forest"
)
```




```
# Print AUC
auc(rf_roc)
```

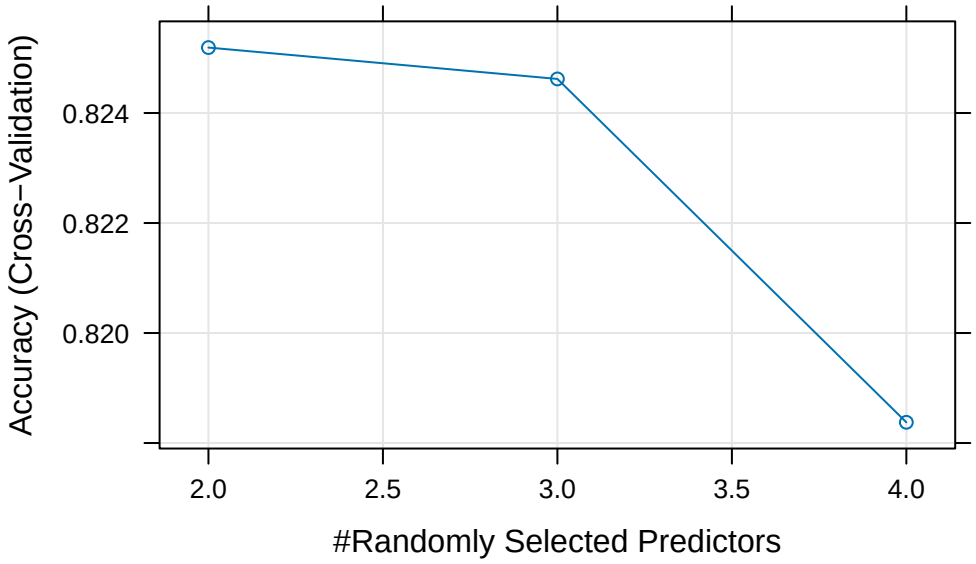
```
Area under the curve: 0.9434
```

```
#Random Forest Hyperparameter Tuning

#Define training control with 5-fold cross-validation
ctrl <- trainControl(
  method = "cv",
  number = 5
)

# Define tuning grid for mtry (number of predictors at each split)
tune_grid <- expand.grid(
  mtry = c(2, 3, 4)  # Adjust as needed
)
# Run grid search with caret::train
 rf_tuned <- train(
 MetabolicSyndrome ~ BMI + Age + BloodGlucose + HDL,
 data = train_data,
 method = "rf",
```

```
metric = "Accuracy",
trControl = ctrl,
tuneGrid = tune_grid,
ntree = 500
)
print(rf_tuned)
```

```
Random Forest

1762 samples
   4 predictor
   2 classes: 'No', 'Yes'

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 1409, 1410, 1410, 1409, 1410
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa
  2     0.8251899  0.6127976
  3     0.8246201  0.6120094
  4     0.8183766  0.5974250

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was mtry = 2.
```

```
plot(rf_tuned)
```




```
#SVM model


# Define cross-validation control
ctrl <- trainControl(
  method = "cv",
  number = 5,
  classProbs = TRUE,
  summaryFunction = twoClassSummary,
  savePredictions = "final"
)

# Define tuning grid for cost (C) and RBF kernel parameter (sigma)
svm_grid <- expand.grid(
  C = 2^(-1:2),
  sigma = 2^(-6:-2)
)
```

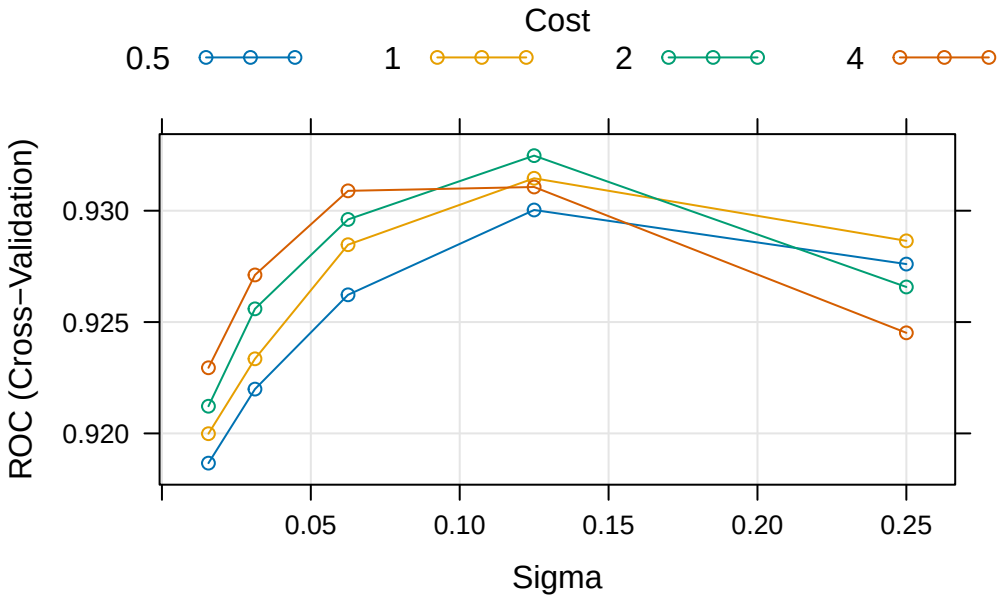
```
#SavingtunedRandomForestmodeltoan.RDatafile
 save(rf_tuned,file= "rf_model.RData")
```

```r
# Train SVM model using RBF kernel
svm_model <- train(
  MetabolicSyndrome ~ BMI + Age + LogBloodGlucose + HDL + WaistCirc + LogTriglycerides,
  data = train_data,
  method = "svmRadial",
  metric = "ROC",
  tuneGrid = svm_grid,
  trControl = ctrl
)

# Output model summary and performance plot
print(svm_model)
```

```
Support Vector Machines with Radial Basis Function Kernel

1762 samples
   6 predictor
   2 classes: 'No', 'Yes'

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 1410, 1409, 1410, 1409, 1410
Resampling results across tuning parameters:

  C    sigma     ROC        Sens       Spec
  0.5  0.015625  0.9186641  0.9038344  0.7475085
  0.5  0.031250  0.9219889  0.9064621  0.7426436
  0.5  0.062500  0.9262236  0.9108366  0.7442958
  0.5  0.125000  0.9300322  0.9047192  0.7523735
  0.5  0.250000  0.9276039  0.9108289  0.7555599
  1.0  0.015625  0.9199866  0.9047077  0.7458694
  1.0  0.031250  0.9233489  0.9090899  0.7458956
  1.0  0.062500  0.9284732  0.9108328  0.7491477
  1.0  0.125000  0.9314564  0.9073316  0.7701547
  1.0  0.250000  0.9286474  0.9047116  0.7717152
  2.0  0.015625  0.9212196  0.9064698  0.7410176
  2.0  0.031250  0.9255932  0.9134605  0.7378180
  2.0  0.062500  0.9296080  0.9117061  0.7523866
  2.0  0.125000  0.9324735  0.9090822  0.7701023
  2.0  0.250000  0.9265757  0.9108366  0.7668502
  4.0  0.015625  0.9229475  0.9125833  0.7378049
  4.0  0.031250  0.9271149  0.9143300  0.7378049
```

```
  4.0  0.062500  0.9308930  0.9143224  0.7588251
  4.0  0.125000  0.9310654  0.9073316  0.7668634
  4.0  0.250000  0.9245173  0.9108404  0.7587857

ROC was used to select the optimal model using the largest value.
The final values used for the model were sigma = 0.125 and C = 2.
```

```r
plot(svm_model)
```




```r
# Predict on test set
svm_preds <- predict(svm_model, newdata = test_data)
svm_probs <- predict(svm_model, newdata = test_data, type = "prob")

# Confusion matrix
confusionMatrix(
  svm_preds,
  test_data$MetabolicSyndrome,
  positive = "Yes"
)
```

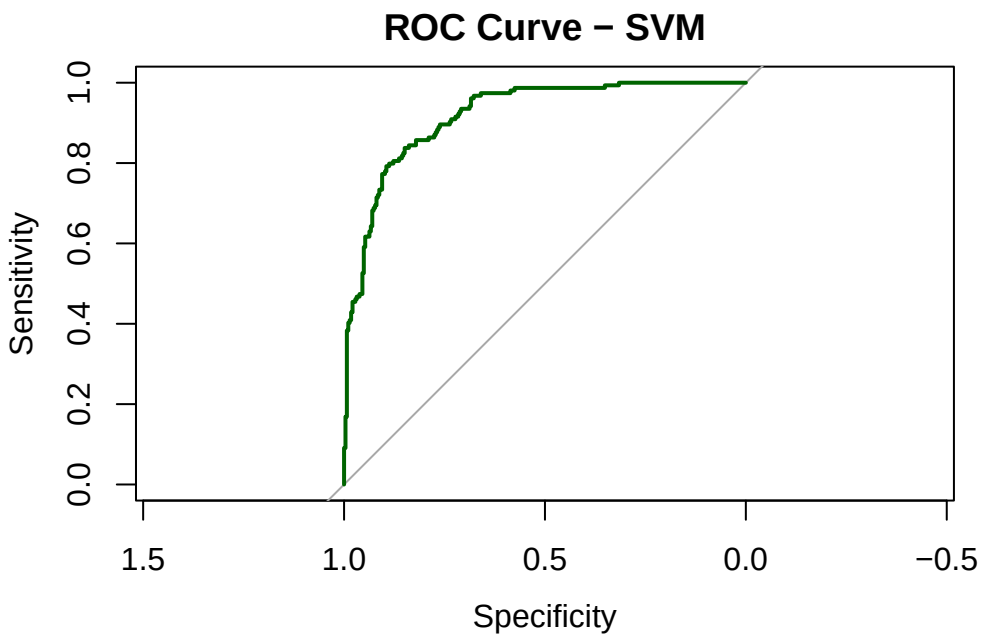
```
Confusion Matrix and Statistics
```

```
          Reference
Prediction  No Yes
       No  252  31
       Yes  33 123

               Accuracy : 0.8542
                 95% CI : (0.8177, 0.8859)
    No Information Rate : 0.6492
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.6809

 Mcnemar's Test P-Value : 0.9005

            Sensitivity : 0.7987
            Specificity : 0.8842
         Pos Pred Value : 0.7885
         Neg Pred Value : 0.8905
             Prevalence : 0.3508
         Detection Rate : 0.2802
   Detection Prevalence : 0.3554
      Balanced Accuracy : 0.8415

       'Positive' Class : Yes
```

```r
# ROC curve
library(pROC)
svm_roc <- roc(
  response = test_data$MetabolicSyndrome,
  predictor = svm_probs$Yes,
  levels = c("No", "Yes")
)
```

```
Setting direction: controls < cases
```

```r
# Plot and display AUC
plot(svm_roc, main = "ROC Curve - SVM", col = "darkgreen", lwd = 2)
```




```
auc(svm_roc)
```

```
Area under the curve: 0.9191
```