Quiz8

Srinivasa Phani Madhav Marupudi

2024-04-08

Q1 Part I. Random Forest with the GreatUnknown Data – Classification Task

1.

```r
library(rpart)
```

```
## Warning: package 'rpart' was built under R version 4.3.3
```

```r
library(caTools)
```

```
## Warning: package 'caTools' was built under R version 4.3.3
```

```r
library(caret)
```

```
## Loading required package: ggplot2

## Loading required package: lattice
```

```r
library(randomForest)
```

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

## randomForest 4.7-1.1

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

##
## Attaching package: 'randomForest'

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

```r
library(party)
```

```
## Warning: package 'party' was built under R version 4.3.3

## Loading required package: grid

## Loading required package: mvtnorm

## Warning: package 'mvtnorm' was built under R version 4.3.3

## Loading required package: modeltools

## Loading required package: stats4

## Loading required package: strucchange

## Warning: package 'strucchange' was built under R version 4.3.3

## Loading required package: zoo
```

```
## Warning: package 'zoo' was built under R version 4.3.3

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Loading required package: sandwich

## Warning: package 'sandwich' was built under R version 4.3.3
```

```
library(rattle)
```

```
## Warning: package 'rattle' was built under R version 4.3.3

## Loading required package: tibble

## Loading required package: bitops

## Rattle: A free graphical interface for data science with R.
## Version 5.5.1 Copyright (c) 2006-2021 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.

##
## Attaching package: 'rattle'

## The following object is masked from 'package:randomForest':
##
##     importance
```

```
data<-read.csv("C:/Users/MSP/Downloads/GreatUnknown.csv", header=T)
head(data)
```

```
##     w1   w2   w3   w4   w5 w6   w7   w8   w9   w10 w11   w12 y
## 1 0.32 0.00 0.00 0.32 0.00  0 0.00 0.00 0.00 0.000   0 0.000 1
## 2 0.14 0.21 0.94 0.14 0.00  0 0.00 0.00 0.00 0.132   0 0.180 1
## 3 1.23 0.19 0.25 0.06 0.32  0 0.06 0.06 0.01 0.143   0 0.184 1
## 4 0.63 0.31 0.63 0.31 0.00  0 0.00 0.00 0.00 0.137   0 0.000 1
## 5 0.63 0.31 0.63 0.31 0.00  0 0.00 0.00 0.00 0.135   0 0.000 1
## 6 1.85 0.00 0.00 0.00 0.00  0 0.00 0.00 0.00 0.223   0 0.000 1
```

```
data$y = as.factor(data$y)
data<-data[complete.cases(data),]
cat("Number of rows left", nrow(data))
```

```
## Number of rows left 4601
```

```
set.seed(123)
split<-createDataPartition(data$y,p=0.75,list=F)
train.data<-data[split, ]
test.data<-data[-split, ]
```

2.

```
model = train(y ~ ., data = train.data, method = "rf", trControl=trainControl("cv", number = 10), impor

model$bestTune
```

```
##   mtry
## 2    7
```

```r
model$finalModel
```

```
##
## Call:
##  randomForest(x = x, y = y, mtry = param$mtry, importance = ..1)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 7
##
##         OOB estimate of  error rate: 8.11%
## Confusion matrix:
##      0    1 class.error
## 0 1972  119  0.05691057
## 1  161 1199  0.11838235
```

```r
conf_matrix_pruned <- model$finalModel$confusion[, -3]

sensitivity_pruned <- conf_matrix_pruned[2, 2] / sum(conf_matrix_pruned[2, ])
specificity_pruned <- conf_matrix_pruned[1, 1] / sum(conf_matrix_pruned[1, ])
accuracy_pruned <- sum(diag(conf_matrix_pruned)) / sum(conf_matrix_pruned)

print("Confusion Matrix (Pruned Tree):")
```

```
## [1] "Confusion Matrix (Pruned Tree):"
```

```r
print(conf_matrix_pruned)
```

```
##      0    1
## 0 1972  119
## 1  161 1199
```

```r
print(paste("Sensitivity (Pruned Tree):", sensitivity_pruned))
```

```
## [1] "Sensitivity (Pruned Tree): 0.881617647058824"
```

```r
print(paste("Specificity (Pruned Tree):", specificity_pruned))
```

```
## [1] "Specificity (Pruned Tree): 0.943089430894309"
```

```r
print(paste("Overall Accuracy (Pruned Tree):", accuracy_pruned))
```

```
## [1] "Overall Accuracy (Pruned Tree): 0.918864097363083"
```

3.

```r
predictions <- model$finalModel %>% predict(test.data) %>% as.vector()

conf_matrix <- table(predictions, test.data$y)

sensitivity <- conf_matrix[2, 2] / sum(conf_matrix[2, ])
specificity <- conf_matrix[1, 1] / sum(conf_matrix[1, ])
accuracy <- sum(diag(conf_matrix)) / sum(conf_matrix)

print("Confusion Matrix :")
```

```
## [1] "Confusion Matrix :"
```

```r
print(conf_matrix)
```

```
##
## predictions   0   1
##           0 648  63
##           1  49 390
```

```r
print(paste("Sensitivity :", sensitivity))
```

```
## [1] "Sensitivity : 0.888382687927107"
```

```r
print(paste("Specificity :", specificity))
```

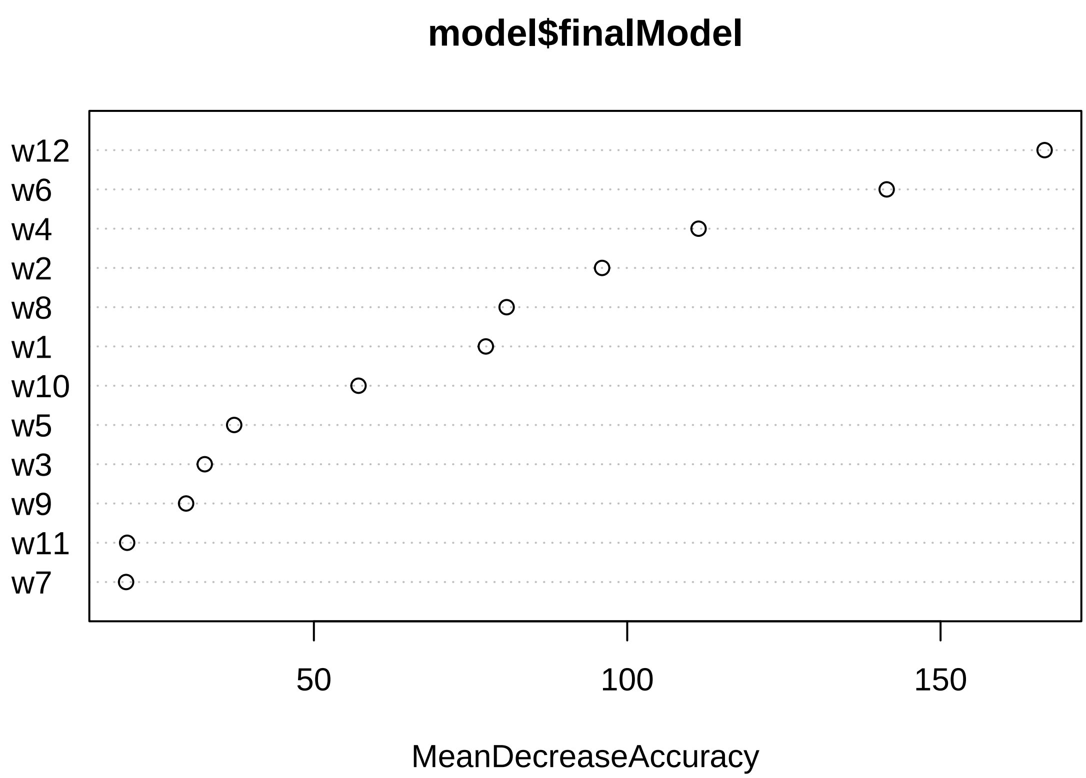
```
## [1] "Specificity : 0.911392405063291"
```

```r
print(paste("Overall Accuracy :", accuracy))
```

```
## [1] "Overall Accuracy : 0.902608695652174"
```

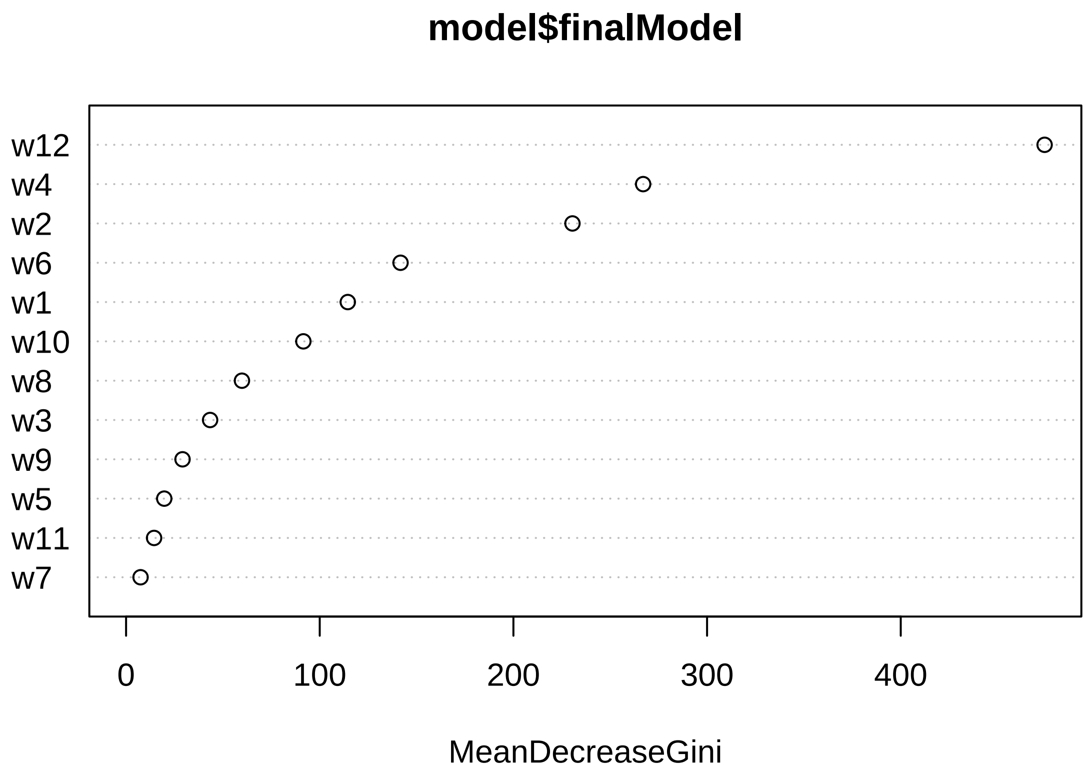
4. (a)

```r
varImpPlot(model$finalModel, type = 1)
```



4 (b)

```r
varImpPlot(model$finalModel, type = 2)
```



5.

```
varImp(model, type = 1)
```

```
## rf variable importance
##
##      Overall
## w12 100.0000
## w6   82.8084
## w4   62.3254
## w2   51.8261
## w8   41.4308
## w1   39.1673
## w10  25.3052
## w5   11.7706
## w3    8.5651
## w9    6.5357
## w11   0.1183
## w7    0.0000
```

Q2 Part II. Random Forest with the QuestionMark Data – Regression Task

1.

```
library(rpart)
library(caTools)
library(caret)
library(randomForest)
```

```r
library(party)
library(rattle)

data<-read.csv("C:/Users/MSP/Downloads/QuestionMark.csv", header=T)
head(data)
```

```
##   w1 w2    w3 w4   w5   w6 w7 w8 w9 w10 w11 w12 w13 w14    y
## 1  7  5 178.0  Y  856  854  2  1  3   1   0   2 548   5 4186
## 2  6  8 201.0  Y 1262    0  2  0  3   1   1   2 460  31 3646
## 3  7  5 234.0  Y  920  866  2  1  3   1   1   2 608   6 4486
## 4  7  5 200.0  Y  961  756  1  0  3   1   1   3 642  36 2816
## 5  8  5 294.2  Y 1145 1053  2  1  4   1   1   3 836   8 5016
## 6  5  5 291.3  Y  796  566  1  1  1   1   0   2 480  14 2876
```

```r
data$w4 = as.factor(data$w4)
cat("Number of rows with missing values", sum(is.na(data)))
```

```
## Number of rows with missing values 0
```

```r
set.seed(123)
split<-createDataPartition(data$y,p=0.95,list=F)
train.data<-data[split, ]
test.data<-data[-split, ]
```

2

```r
set.seed(123)
model <- train(
y ~., data = train.data, method = "rf",
trControl = trainControl("cv", number = 10)
)

model$bestTune
```

```
##   mtry
## 2    8
```

3

```r
predictions <- model %>% predict(test.data)
RMSE(predictions, test.data$y)
```

```
## [1] 563.0106
```

4.

```r
rf = randomForest(y ~ ., data = train.data, ntree=500,
mtry=8,keep.forest=FALSE,importance=TRUE)

rf
```

```
##
## Call:
##  randomForest(formula = y ~ ., data = train.data, ntree = 500,      mtry = 8, keep.forest = FALSE, i
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 8
##
##           Mean of squared residuals: 368262.4
```

```
##                     % Var explained: 85.43
```

```r
# a
sqrt(rf$mse[500])
```

```
## [1] 606.8463
```

```r
# b
randomForest::importance(rf)
```

```
##       %IncMSE IncNodePurity
## w1  54.208114    1486112304
## w2  12.526202      33786868
## w3  17.076624     189714639
## w4  13.690677      13023676
## w5  30.498005     324499476
## w6  32.067614     245338451
## w7  15.571431     182057642
## w8  16.007134      24128736
## w9  12.170567      38111299
## w10  9.551974       6986144
## w11 20.746812      75181567
## w12 20.551711     479212026
## w13 21.243398     252465055
## w14 19.701785     108462679
```

```r
# c
varImpPlot(rf)
```