Quiz 8

Chaeeun Shin

04/08/2024

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.2.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(caret)
```

```
## Loading required package: ggplot2
```

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

```
library(randomForest)
```

```
## 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
```

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

#1

```
GreatUnknown <- read.csv("GreatUnknown.csv")
GreatUnknown <- na.omit(GreatUnknown)
cat("There are ",nrow(GreatUnknown),"cases left.")
```

```
## There are  4601 cases left.
```

```
GreatUnknown$y <- as.factor(GreatUnknown$y)

set.seed(123)
training.samples <- GreatUnknown$y %>%
```

```
  createDataPartition(p=0.75,list=FALSE)
train.data <- GreatUnknown[training.samples,]
test.data <- GreatUnknown[-training.samples,]
nrow(train.data)
```

```
## [1] 3451
```

```
nrow(test.data)
```

```
## [1] 1150
```

#2

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

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

```
model$finalModel
```

```
##
## Call:
##  randomForest(x = x, y = y, mtry = param$mtry, importance = TRUE)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 7
##
##         OOB estimate of  error rate: 8.32%
## Confusion matrix:
##      0    1 class.error
## 0 1974  117  0.05595409
## 1  170 1190  0.12500000
```

```
cat("Sensitivity: ", 1190/(1190+170),"\n")
```

```
## Sensitivity:  0.875
```

```
cat("Specificity: ", 1974/(1974+117),"\n")
```

```
## Specificity:  0.9440459
```

```
cat("Accuracy: ", (1974+1190)/(1974+117+170+1190),"\n")
```

```
## Accuracy:  0.9168357
```

#3

```
pred <- model %>% predict(test.data)
table(pred,test.data$y)
```

```
##
## pred   0   1
##    0 648  64
##    1  49 389
```

```
cat("Sensitivity: ",389/(49+389),"\n")
```

```
## Sensitivity:  0.8881279
```

```
cat("Specificity: ",648/(64+648),"\n")
```

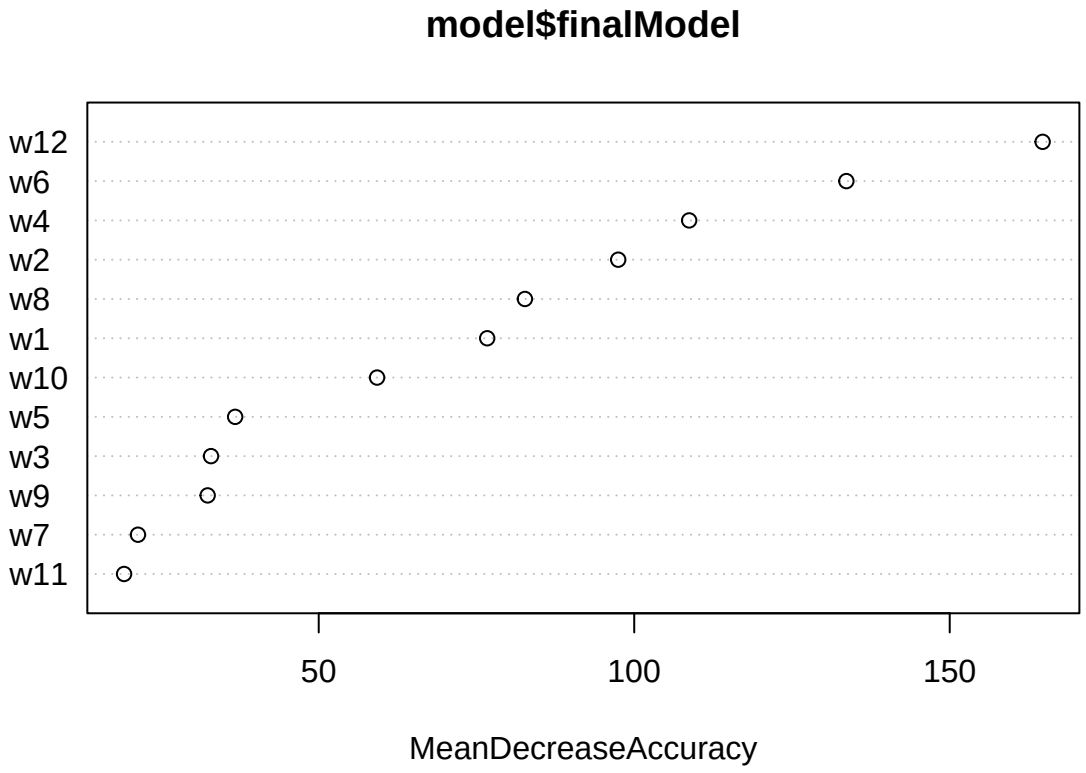
```
## Specificity:  0.9101124
```

```
cat("Accuracy: ",(648+389)/(64+648+49+389),"\n")
```

```
## Accuracy:  0.9017391
```

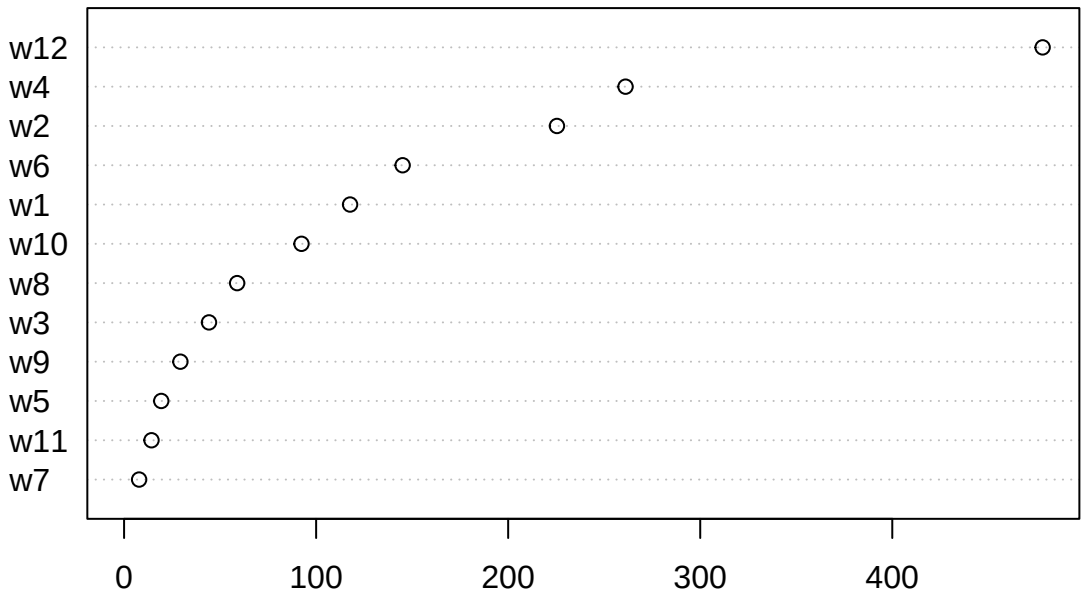
#4: plot MeanDecreaseAccuracy

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



#4: plot MeanDecreaseGini

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



#5

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

```
## rf variable importance
##
##     Overall
## w12 100.000
## w6   78.632
## w4   61.520
## w2   53.795
## w8   43.638
## w1   39.534
## w10  27.544
## w5   12.092
## w3    9.464
## w9    9.099
## w7    1.513
## w11   0.000
```

#Part II #1

```
QuestionMark <- read.csv("QuestionMark.csv")
QuestionMark <- na.omit(QuestionMark)
cat("There are", nrow(QuestionMark), "observations left.")
```

```
## There are 1460 observations left.
```

```
set.seed(123)
training.samples <- QuestionMark$y %>%
  createDataPartition(p=0.95,list=FALSE)
train.data <- QuestionMark[training.samples,]
test.data <- QuestionMark[-training.samples,]
```

```r
nrow(train.data)
```

```
## [1] 1388
```

```r
nrow(test.data)
```

```
## [1] 72
```

#2

```r
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)
cat("RMSE: ", RMSE(predictions, test.data$y))
```

```
## RMSE:  566.1903
```

#4

```r
set.seed(123)
rf <- randomForest(y~., data=QuestionMark, ntree=500, mtry=8, keep.forest=FALSE, importance=TRUE)
sqrt(rf$mse[500])
```

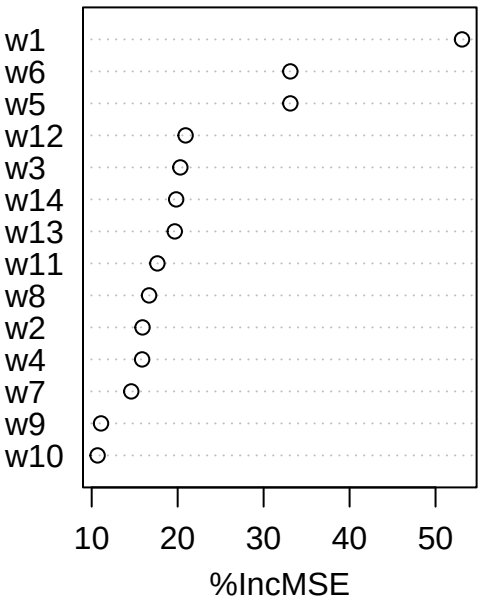
```
## [1] 605.1423
```

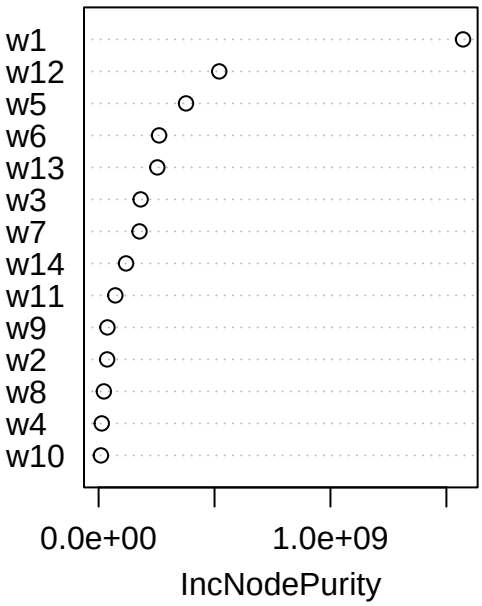
```r
importance(rf)
```

```
##      %IncMSE IncNodePurity
## w1  53.08075    1571844466
## w2  15.91305      36854953
## w3  20.31184     181246038
## w4  15.86470      13576603
## w5  33.10700     377005198
## w6  33.10911     260844692
## w7  14.60269     176359175
## w8  16.67467      22715940
## w9  11.08663      38034755
## w10 10.68435      10025458
## w11 17.63950      72047230
## w12 20.92142     520303858
## w13 19.66340     253331597
## w14 19.82812     118343211
```

```r
varImpPlot(rf)
```

rf


w1
w6
w5
w12
w3
w14
w13
w11
w8
w2
w4
w7
w9
w10
10
20
30
40
50
%IncMSE


w1
w12
w5
w6
w13
w3
w7
w14
w11
w9
w2
w8
w4
w10
0.0e+00
1.0e+09
IncNodePurity

“‘