ams580\_quiz5

Mustafa Yigit Isik

2023-03-08

**PART 1** *Question 1*

#Install Packs  
if (!requireNamespace("tidyverse")) install.packages('tidyverse')

## Loading required namespace: tidyverse

if (!requireNamespace("caret")) install.packages('caret')

## Loading required namespace: caret

if (!requireNamespace("randomForest")) install.packages('randomForest')

## Loading required namespace: randomForest

if (!requireNamespace("party")) install.packages('party')

## Loading required namespace: party

library(tidyverse)

## ── Attaching packages  
## ───────────────────────────────────────  
## tidyverse 1.3.2 ──

## ✔ ggplot2 3.4.0 ✔ purrr 1.0.1   
## ✔ tibble 3.1.8 ✔ dplyr 1.0.10  
## ✔ tidyr 1.3.0 ✔ stringr 1.5.0   
## ✔ readr 2.1.4 ✔ forcats 1.0.0   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

library(caret)

## Loading required package: lattice  
##   
## Attaching package: 'caret'  
##   
## The following object is masked from 'package:purrr':  
##   
## lift

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

library(party)

## Loading required package: grid  
## Loading required package: mvtnorm  
## Loading required package: modeltools  
## Loading required package: stats4  
## Loading required package: strucchange  
## Loading required package: zoo  
##   
## Attaching package: 'zoo'  
##   
## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric  
##   
## Loading required package: sandwich  
##   
## Attaching package: 'strucchange'  
##   
## The following object is masked from 'package:stringr':  
##   
## boundary

#Read/Clean Data  
data <- read.csv('/Users/mustafayigitisik/Desktop/stuff/semesters/spring 2023/ams 580/quizzes/quiz5/GreatUnknown.csv')  
data <- na.omit(data)  
cat('There are', nrow(data), 'observations left.')

## There are 4601 observations left.

data$y <- as.factor(data$y)  
str(data)

## 'data.frame': 4601 obs. of 13 variables:  
## $ w1 : num 0.32 0.14 1.23 0.63 0.63 1.85 1.92 1.88 0.61 0.19 ...  
## $ w2 : num 0 0.21 0.19 0.31 0.31 0 0 0 0.3 0.38 ...  
## $ w3 : num 0 0.94 0.25 0.63 0.63 0 0.64 0 0.76 0 ...  
## $ w4 : num 0.32 0.14 0.06 0.31 0.31 0 0.96 0 0 0 ...  
## $ w5 : num 0 0 0.32 0 0 0 0 0 3.53 0.06 ...  
## $ w6 : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ w7 : num 0 0 0.06 0 0 0 0 0 0 0 ...  
## $ w8 : num 0 0 0.06 0 0 0 0 0 0 0 ...  
## $ w9 : num 0 0 0.01 0 0 0 0 0 0 0.04 ...  
## $ w10: num 0 0.132 0.143 0.137 0.135 0.223 0.054 0.206 0.271 0.03 ...  
## $ w11: num 0 0 0 0 0 0 0 0 0 0 ...  
## $ w12: num 0 0.18 0.184 0 0 0 0.054 0 0.203 0.081 ...  
## $ y : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...

#Split Data  
set.seed(123)  
training.samples <- data$y %>%   
 createDataPartition(p = 0.75, list = FALSE)  
train.data <- data[training.samples, ]  
test.data <- data[-training.samples, ]

*Question 2*

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

## Sensitivity  
1190/(1190+170)

## [1] 0.875

## Specificity  
1974/(1974+117)

## [1] 0.9440459

## Accuracy  
(1190+1974)/(1190+1974+117+170)

## [1] 0.9168357

*Question 3*

pred <- model %>% predict(test.data)  
confusionMatrix(factor(pred), factor(test.data$y))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 648 64  
## 1 49 389  
##   
## Accuracy : 0.9017   
## 95% CI : (0.8831, 0.9183)  
## No Information Rate : 0.6061   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.793   
##   
## Mcnemar's Test P-Value : 0.1878   
##   
## Sensitivity : 0.9297   
## Specificity : 0.8587   
## Pos Pred Value : 0.9101   
## Neg Pred Value : 0.8881   
## Prevalence : 0.6061   
## Detection Rate : 0.5635   
## Detection Prevalence : 0.6191   
## Balanced Accuracy : 0.8942   
##   
## 'Positive' Class : 0   
##

## Sensitivity  
389/(389+49)

## [1] 0.8881279

## Specificity  
648/(648+64)

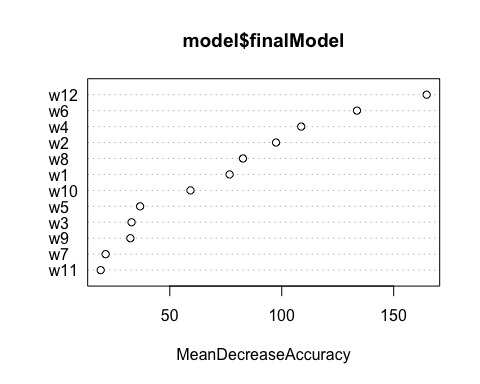
## [1] 0.9101124

## Accuracy  
(389+648)/(389+648+64+49)

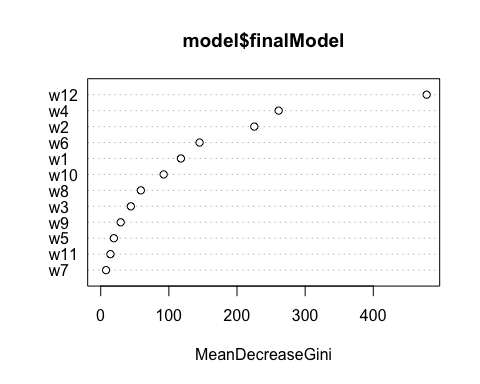
## [1] 0.9017391

*Question 4*

# Plot MeanDecreaseAccuracy  
varImpPlot(model$finalModel, type = 1)



# Plot MeanDecreaseGini  
varImpPlot(model$finalModel, type = 2)

 *Question 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 2** *Question 1*

#Read/Clean Data  
data <- read.csv('/Users/mustafayigitisik/Desktop/stuff/semesters/spring 2023/ams 580/quizzes/quiz5/QuestionMark.csv')  
data <- na.omit(data)  
cat('There are', nrow(data), 'observations left.')

## There are 1460 observations left.

data$y <- as.factor(data$y)  
data$w4 <- ifelse(data$w4=="Y",1,0)  
str(data)

## 'data.frame': 1460 obs. of 15 variables:  
## $ w1 : int 7 6 7 7 8 5 8 7 7 5 ...  
## $ w2 : int 5 8 5 5 5 5 5 6 5 6 ...  
## $ w3 : num 178 201 234 200 294 ...  
## $ w4 : num 1 1 1 1 1 1 1 1 1 1 ...  
## $ w5 : int 856 1262 920 961 1145 796 1694 1107 1022 1077 ...  
## $ w6 : int 854 0 866 756 1053 566 0 983 752 0 ...  
## $ w7 : int 2 2 2 1 2 1 2 2 2 1 ...  
## $ w8 : int 1 0 1 0 1 1 0 1 0 0 ...  
## $ w9 : int 3 3 3 3 4 1 3 3 2 2 ...  
## $ w10: int 1 1 1 1 1 1 1 1 2 2 ...  
## $ w11: int 0 1 1 1 1 0 1 2 2 2 ...  
## $ w12: int 2 2 2 3 3 2 2 2 2 1 ...  
## $ w13: int 548 460 608 642 836 480 636 484 468 205 ...  
## $ w14: int 5 31 6 36 8 14 2 36 58 58 ...  
## $ y : Factor w/ 663 levels "714","722.22",..: 413 340 443 195 495 204 574 391 152 114 ...

#Split Data  
set.seed(123)  
training.samples <- data$y %>%  
 createDataPartition(p = 0.95, list = FALSE)

## Warning in createDataPartition(., p = 0.95, list = FALSE): Some classes have  
## a single record ( 714, 722.22, 774, 802, 816, 1056, 1066, 1135.86, 1186, 1236,  
## 1263.66, 1306, 1346, 1384, 1386, 1466, 1476, 1516, 1526, 1536, 1546, 1576, 1606,  
## 1626, 1686, 1696, 1714, 1724, 1726, 1766, 1796, 1805.42, 1823, 1842, 1856, 1874,  
## 1896, 1906, 1911, 1966, 1982, 1988, 2006, 2014, 2036, 2052, 2071.52, 2076, 2080,  
## 2088, 2096, 2126, 2134, 2141, 2164, 2174, 2185.6, 2186, 2195.18, 2196.16, 2236,  
## 2241, 2306.08, 2337, 2346, 2354, 2384, 2393.16, 2395.28, 2400, 2411, 2436, 2446,  
## 2448, 2474, 2486, 2488, 2539.5, 2546, 2580, 2594, 2595, 2621, 2644, 2661, 2686,  
## 2690, 2704.64, 2705, 2712, 2731, 2734, 2735.2, 2754, 2754.1, 2765, 2774, 2792,  
## 2793.74, 2806, 2808, 2814, 2815, 2820, 2846, 2858.5, 2868, 2875.06, 2881, 2891,  
## 2894, 2899.04, 2914, 2921, 2926, 2934, 2946, 2952, 2964, 2966, 2992, 3002, 3003,  
## 3010, 3026, 3031, 3034, 3044, 3046, 3082.74, 3087.5, 3102, 3106, 3114, 3132.7,  
## 3134, 3146, 3154.64, 3186, 3194, 3204.68, 3213.9, 3215, 3220, 3251, 3266, 3294,  
## 3295.8, 3310, 3314, 3315.8, 3319, 3324, 3328, 3360.8, 3414, 3415.8, 3446, 3451,  
## 3454, 3456, 3464, 3471.7, 3486, 3490.66, 3494, 3506, 3514, 3544.64, 3545.7,  
## 3584, 3590.8, 3594, 3604, 3606, 3606.8, 3608, 3609.3, 3638.68, 3646, 3654,  
## 3656, 3674, 3676, 3680, 3686, 3694, 3698, 3711, 3714, 3726, 3731, 3733, 3734,  
## 3736, 3750, 3758, 3771, 3786, 3790, 3815, 3858.8, 3893.58, 3900.02, 3910, 3924,  
## 3986, 3994, 4018, 4018.82, 4028.48, 4052, 4069.3, 4074, 4111, 4114, 4135, 4136,  
## 4142, 4154, 4182, 4186, 4206, 4274, 4281, 4285.8, 4306, 4314, 4320, 4336, 4346,  
## 4352.74, 4366, 4376, 4400.2, 4446, 4476, 4506, 4550, 4569.6, 4573.5, 4595, 4596,  
## 4605.12, 4626, 4668, 4676, 4679.4, 4680.6, 4718.56, 4766, 4776, 4806, 4809.72,  
## 4811.98, 4814, 4836, 4876, 4904, 4908, 4916, 4923, 4926, 4947.56, 4982.56,  
## 4994, 5010, 5027.6, 5036, 5056, 5069.56, 5076, 5081.86, 5096, 5114, 5126, 5134,  
## 5142, 5156, 5166, 5176, 5196, 5206, 5224, 5246, 5256, 5261.6, 5276, 5284.7,  
## 5298.64, 5307.22, 5316, 5334, 5335.58, 5346, 5356, 5406, 5411.8, 5454, 5496,  
## 5502, 5510.5, 5514, 5515.4, 5526, 5536, 5566, 5606, 5636, 5640.26, 5674.44,  
## 5685.26, 5736, 5757.8, 5796, 5877.54, 5896, 5916, 5925.86, 5956, 6012, 6036,  
## 6046, 6085.54, 6116, 6134, 6136, 6156, 6196, 6246, 6253.44, 6266, 6276, 6312.26,  
## 6326, 6331, 6348, 6356, 6377.22, 6396, 6414, 6496, 6522, 6528.48, 6536, 6576,  
## 6594, 6679.36, 6736, 6756, 6766, 6811, 6836, 6868.86, 6976, 7096, 7198, 7216,  
## 7254.38, 7361.88, 7414, 7433.56, 7464.04, 7466, 7496, 7516, 7564.52, 7566, 7586,  
## 7616, 7636, 7695.4, 7741, 7856, 7866, 7904.64, 7908.34, 7916, 7919.84, 8056,  
## 8073.22, 8076, 8216, 8266, 8321.96, 8476, 8513.4, 8536, 8616, 8759.08, 8791.6,  
## 8816, 8941.22, 9055, 9316, 9346, 9516, 9716, 10052.74, 10776, 11116, 11147.62,  
## 11674.66, 12249.14, 12516, 14916, 15116 ) and these will be selected for the  
## sample

train.data <- data[training.samples, ]  
test.data <- data[-training.samples, ]  
str(train.data)

## 'data.frame': 1459 obs. of 15 variables:  
## $ w1 : int 7 6 7 7 8 5 8 7 7 5 ...  
## $ w2 : int 5 8 5 5 5 5 5 6 5 6 ...  
## $ w3 : num 178 201 234 200 294 ...  
## $ w4 : num 1 1 1 1 1 1 1 1 1 1 ...  
## $ w5 : int 856 1262 920 961 1145 796 1694 1107 1022 1077 ...  
## $ w6 : int 854 0 866 756 1053 566 0 983 752 0 ...  
## $ w7 : int 2 2 2 1 2 1 2 2 2 1 ...  
## $ w8 : int 1 0 1 0 1 1 0 1 0 0 ...  
## $ w9 : int 3 3 3 3 4 1 3 3 2 2 ...  
## $ w10: int 1 1 1 1 1 1 1 1 2 2 ...  
## $ w11: int 0 1 1 1 1 0 1 2 2 2 ...  
## $ w12: int 2 2 2 3 3 2 2 2 2 1 ...  
## $ w13: int 548 460 608 642 836 480 636 484 468 205 ...  
## $ w14: int 5 31 6 36 8 14 2 36 58 58 ...  
## $ y : Factor w/ 663 levels "714","722.22",..: 413 340 443 195 495 204 574 391 152 114 ...

str(test.data)

## 'data.frame': 1 obs. of 15 variables:  
## $ w1 : int 6  
## $ w2 : int 5  
## $ w3 : num 177  
## $ w4 : num 1  
## $ w5 : int 1192  
## $ w6 : int 403  
## $ w7 : int 1  
## $ w8 : int 0  
## $ w9 : int 2  
## $ w10: int 1  
## $ w11: int 2  
## $ w12: int 1  
## $ w13: int 240  
## $ w14: int 6  
## $ y : Factor w/ 663 levels "714","722.22",..: 195

*Question 2*

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

*Question 3*

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

*Question 4*

##set.seed(123)  
##rf <- randomForest(y ~ ., data=data, ntree=500, mtry=5,keep.forest=FALSE,importance=TRUE)  
##rf  
##sqrt(rf$mse[500])  
##importance(rf)  
##varImpPlot(rf)