SDM_Assignment5_2

Sri Balaji Muruganandam

10/12/2021

Setting Working Directory

```
rm(list = ls())
setwd("G:\\SDM_Sem01\\Assignment5")
```

Importing necessary libraries

```
options(warn=-1)
library(rpart)
library(gbm)
```

```
## Loaded gbm 2.1.8
```

```
library(MASS)
library(randomForest)
```

```
## randomForest 4.6-14
```

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

```
options(warn=0)
```

The Cleveland heart-disease study was conducted by the Cleveland Clinic Foundation. The response variable is “diag1” (diagnosis of heart disease: buff = healthy, sick = heart disease). There is a second “diag2” that contains stage information about the sick, this can be disregarded. There were 303 patients in the study, and 13 predictive variables, including age, gender, and a range of biological measurements.

Loading the Cleveland dataset

```
load("cleveland.RData")
dim(cleveland)
```

```
## [1] 296  15
```

Exploring the high level overview of the data

```
names(cleveland)
```

```
##  [1] "age"      "gender"   "cp"       "trestbps" "chol"     "fbs"
##  [7] "restecg"  "thalach"  "exang"    "oldpeak"  "slope"    "ca"
## [13] "thal"     "diag1"    "diag2"
```

```
head(cleveland,5)
```

```
##   age gender     cp trestbps chol  fbs restecg thalach exang oldpeak slope ca
## 1  63   male angina      145  233 true     hyp     150   fal     2.3  down  0
## 2  67   male asympt      160  286  fal     hyp     108  true     1.5  flat  3
## 3  67   male asympt      120  229  fal     hyp     129  true     2.6  flat  2
## 4  37   male notang      130  250  fal    norm     187   fal     3.5  down  0
## 5  41    fem abnang      130  204  fal     hyp     172   fal     1.4    up  0
##   thal diag1 diag2
## 1  fix  buff     H
## 2 norm  sick    S2
## 3  rev  sick    S1
## 4 norm  buff     H
## 5 norm  buff     H
```

Splitting the data into training and the test data

```
set.seed(23)
random_index = sample(c(1:nrow(cleveland)), size = round(8/10 * nrow(cleveland)), replace = F
ALSE)
train_data = cleveland[random_index,]
test_data = cleveland[-random_index,]

y_train_data = as.numeric(cleveland$diag1)-1
y_test_data = as.numeric(cleveland$diag1)-1
dim(train_data)
```

```
## [1] 237  15
```

```
dim(test_data)
```

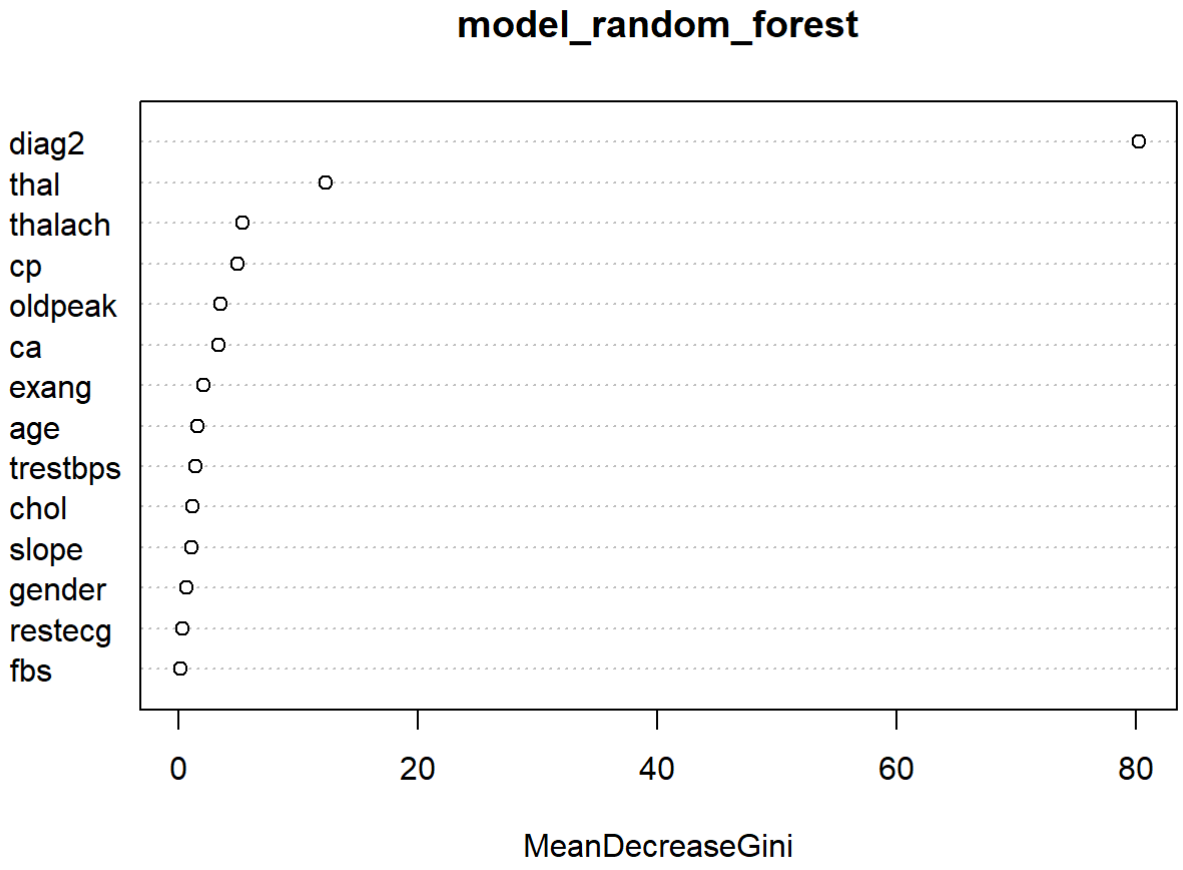
```
## [1] 59 15
```

Modelling Random Forest

Modelling with m = 5

```
model_random_forest <- randomForest(diag1~., data = train_data, n.tree = 1000, mtry = 5)

varImpPlot(model_random_forest)
```




```
importance(model_random_forest)
```

```
##          MeanDecreaseGini
## age             1.5850848
## gender          0.6035291
## cp              4.9343593
## trestbps        1.3845789
## chol            1.1254968
## fbs             0.1034671
## restecg         0.3126166
## thalach         5.2820791
## exang           2.0888872
## oldpeak         3.4765741
## slope           1.0685151
## ca              3.3217610
## thal           12.3022402
## diag2          80.2133423
```

Predicting with bagging model

```
y_predict = predict(model_random_forest, newdata = test_data, type = "response")
y_predict = as.numeric(y_predict)-1
#print(y_predict)
#print(y_test_data)
```

Prediction Error for Bagging model

```
prediction_error_rf = sum(abs(y_predict - y_test_data))/length(y_test_data)
```

```
## Warning in y_predict - y_test_data: longer object length is not a multiple of
## shorter object length
```

```
print(prediction_error_rf)
```

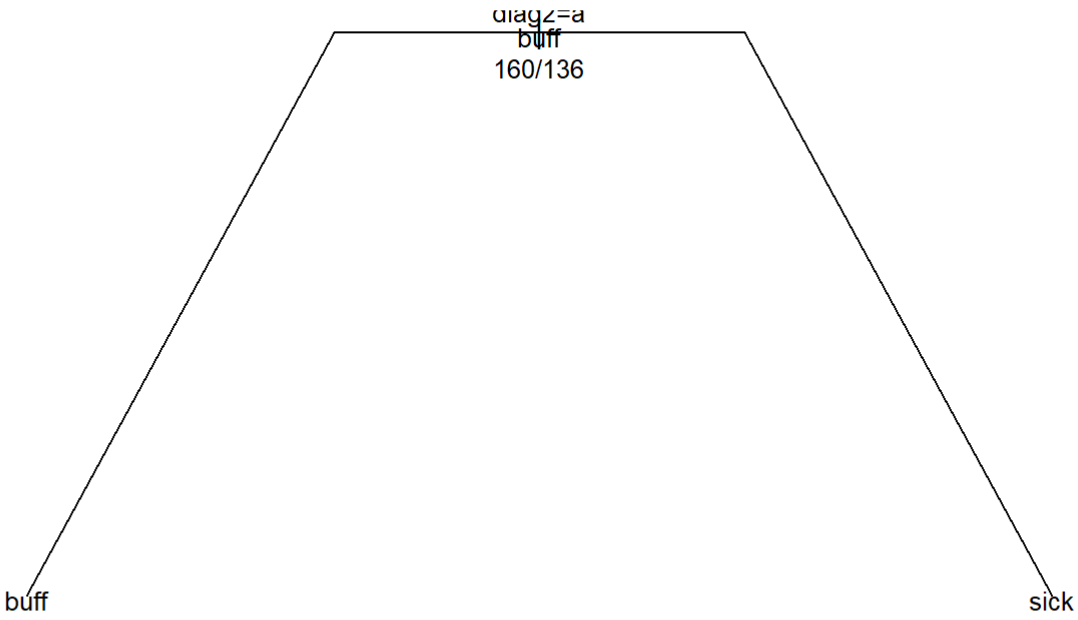
```
## [1] 0.4797297
```

Modelling CART

```
model_control <- rpart.control(minsplit = 7, xval = 10, cp = 0)
model_fit <- rpart(diag1~., data = cleveland, method = "class", control = model_control)
```

Plotting the tree

```
plot(model_fit, branch = .4, uniform = T, compress = T)
text(model_fit, use.n = T, all = T, cex = 0.8)
```



Pruning the tree

```
#model_control <- rpart.control(minsplit = 7, xval = 10, cp = 0)
#model_fit_prune <- rpart(diag1~., data = cleveland, method = "class", control = model_contro
l)

#plot(model_fit_prune$cptable[,4], main = "CP for Model Selection", ylab = "CP")

#minimum_cp = which.min(model_fit_prune$cptable[,4])
#model_prune <- prune(model_fit_prune, cp = model_fit_prune$cptable[minimum_cp,1])
```

```
model_controls = rpart.control(minbucket = 2, minsplit = 4, xval = 10, cp = 0)
model_diag = rpart(diag1~., data = cleveland, control = model_controls)

minimum_cp = which.min(model_diag$cptable[,4])
model_prune = prune(model_diag, cp = model_diag$cptable[minimum_cp, 1])
```

Predicting for Test data

```
y_predict = predict(model_prune, newdata = train_data)
print(y_predict)
```

```
##     buff sick
## 284    1    0
## 248    0    1
## 121    0    1
## 171    1    0
## 103    1    0
## 290    1    0
## 149    0    1
## 145    1    0
## 164    1    0
## 198    0    1
## 31     1    0
## 86     0    1
## 236    1    0
## 10     0    1
## 38     1    0
## 167    1    0
## 251    0    1
## 47     1    0
## 66     1    0
## 69     0    1
## 105    1    0
## 280    1    0
## 91     0    1
## 231    1    0
## 206    1    0
## 59     1    0
## 178    1    0
## 159    0    1
## 241    1    0
## 278    1    0
## 190    1    0
## 3      0    1
## 50     0    1
## 225    1    0
## 60     0    1
## 14     1    0
## 9      0    1
## 147    0    1
## 51     0    1
## 244    0    1
## 270    1    0
## 279    1    0
## 262    1    0
## 146    0    1
## 219    1    0
## 243    1    0
## 188    1    0
## 294    0    1
## 125    0    1
## 275    0    1
## 182    1    0
## 132    1    0
## 172    1    0
## 174    1    0
```

```
## 28     0    1
## 56     0    1
## 76     1    0
## 107    0    1
## 126    0    1
## 266    1    0
## 161    1    0
## 58     0    1
## 156    1    0
## 195    1    0
## 138    0    1
## 295    1    0
## 133    0    1
## 263    0    1
## 74     1    0
## 252    0    1
## 101    0    1
## 37     1    0
## 127    1    0
## 289    0    1
## 199    0    1
## 232    0    1
## 41     0    1
## 89     1    0
## 120    1    0
## 189    1    0
## 140    0    1
## 287    1    0
## 98     0    1
## 162    0    1
## 128    1    0
## 292    1    0
## 222    1    0
## 30     1    0
## 90     0    1
## 166    1    0
## 19     0    1
## 223    1    0
## 276    0    1
## 228    0    1
## 112    1    0
## 210    0    1
## 249    0    1
## 40     0    1
## 208    1    0
## 35     1    0
## 193    0    1
## 291    1    0
## 155    1    0
## 286    0    1
## 25     0    1
## 204    0    1
## 234    0    1
## 212    0    1
## 179    0    1
## 78     1    0
```

```
## 152    0    1
## 183    0    1
## 64     0    1
## 273    0    1
## 170    1    0
## 230    0    1
## 197    1    0
## 237    1    0
## 32     0    1
## 80     1    0
## 57     1    0
## 43     0    1
## 181    0    1
## 106    0    1
## 88     1    0
## 216    1    0
## 175    0    1
## 184    0    1
## 119    1    0
## 253    0    1
## 137    0    1
## 87     1    0
## 95     1    0
## 268    1    0
## 36     0    1
## 220    1    0
## 130    1    0
## 12     1    0
## 256    0    1
## 16     1    0
## 293    1    0
## 226    0    1
## 53     0    1
## 274    1    0
## 247    0    1
## 250    1    0
## 116    0    1
## 62     0    1
## 18     0    1
## 79     0    1
## 123    0    1
## 254    0    1
## 82     1    0
## 46     1    0
## 221    1    0
## 55     1    0
## 224    1    0
## 255    0    1
## 196    1    0
## 265    1    0
## 115    1    0
## 1      1    0
## 165    0    1
## 169    0    1
## 83     1    0
## 269    0    1
```

```
## 283    1    0
## 71     1    0
## 217    1    0
## 81     1    0
## 6      1    0
## 134    0    1
## 209    1    0
## 180    1    0
## 94     1    0
## 144    0    1
## 8      1    0
## 261    1    0
## 233    1    0
## 49     1    0
## 104    1    0
## 77     1    0
## 20     0    1
## 214    0    1
## 102    0    1
## 34     0    1
## 142    0    1
## 93     1    0
## 202    0    1
## 205    1    0
## 186    1    0
## 272    0    1
## 13     0    1
## 176    0    1
## 151    1    0
## 281    1    0
## 192    1    0
## 122    1    0
## 239    1    0
## 54     1    0
## 52     0    1
## 185    1    0
## 154    1    0
## 200    1    0
## 157    1    0
## 96     0    1
## 21     1    0
## 257    0    1
## 61     0    1
## 131    1    0
## 124    1    0
## 296    1    0
## 203    0    1
## 85     1    0
## 117    0    1
## 136    0    1
## 260    0    1
## 177    0    1
## 84     1    0
## 158    0    1
## 135    0    1
## 17     1    0
```

```
## 282    0    1
## 245    0    1
## 118    0    1
## 191    1    0
## 160    0    1
## 23     1    0
## 150    1    0
## 227    0    1
## 75     0    1
## 22     1    0
## 7      0    1
## 111    0    1
## 97     0    1
## 201    0    1
## 72     0    1
```