Homework 8 - Random Forests

Shiloh Bradley

6/24/2020

Charles Book Club

```
df <- read.csv("Charles_BookClub.csv", header = TRUE)
dim(df)  ## 2000 18
```

```
## [1] 2000   18
```

```
head(df)
```

```
##   Seq. ID. Gender   M  R F FirstPurch ChildBks YouthBks CookBks DoltYBks
## 1    1   2      0 138 28 3         40        0        1       0        1
## 2    2  30      1 240 14 1         14        1        0       0        0
## 3    3  59      1  97  6 2         10        0        0       0        0
## 4    4  89      1 348  2 7         38        1        1       1        0
## 5    5  96      0 239 20 2         28        0        0       1        0
## 6    6 120      1 253 10 4         20        1        0       0        0
##   RefBks ArtBks GeogBks ItalCook ItalHAtlas ItalArt Florence
## 1      0      0       1        0          0       0        0
## 2      0      0       0        0          0       0        0
## 3      0      0       0        0          0       0        0
## 4      1      0       1        0          0       0        0
## 5      0      0       1        0          0       0        0
## 6      0      1       0        0          0       0        1
```

```
tail(df)
```

```
##      Seq.   ID. Gender   M  R F FirstPurch ChildBks YouthBks CookBks
## 1995 1995 49781      1 192  8 1          8        0        0       0
## 1996 1996 49801      1 164 12 5         32        0        0       1
## 1997 1997 49866      0 294 10 1         10        0        0       0
## 1998 1998 49872      0 261  4 2         10        0        0       0
## 1999 1999 49914      1  41 32 1         32        0        0       1
## 2000 2000 49962      1 308 12 1         12        0        0       0
##      DoltYBks RefBks ArtBks GeogBks ItalCook ItalHAtlas ItalArt Florence
## 1995        0      0      0       0        0          0       0        0
## 1996        0      0      1       2        1          0       1        1
## 1997        0      0      0       0        0          0       0        0
## 1998        0      0      0       0        0          0       0        0
## 1999        0      0      0       0        0          0       0        0
## 2000        0      0      0       0        0          0       0        0
```

```
p <- 0.25
M <- p * nrow(df)
#set initial seed for repeatability
set.seed(113117)

holdout <- sample(1:nrow(df), M, replace = F)
df.train <- df[-holdout, ]
df.test  <- df[holdout, ]
```

```r
dim(df.train) ## 1500 18
```

```
## [1] 1500   18
```

```r
dim(df.test)  ## 500 18
```

```
## [1] 500  18
```

```r
features <- setdiff(names(df.train), "Florence")
rf1 <- randomForest(factor(Florence) ~ ., data = df.train)
rf1
```

```
##
## Call:
##  randomForest(formula = factor(Florence) ~ ., data = df.train)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 10.47%
## Confusion matrix:
##      0 1 class.error
## 0 1335 3 0.002242152
## 1  154 8 0.950617284
```

```r
CM.rf_train <- rf1$confusion
CM.rf_train
```

```
##      0 1 class.error
## 0 1335 3 0.002242152
## 1  154 8 0.950617284
```

```r
OA.rf_train <- sum(diag(CM.rf_train))/sum(CM.rf_train)
```

Tuning Random Forests - only a few parameters

ntree: number of trees. We want enough trees to stabalize the error but using too many trees is unncessarily inefficient, especially when using large data sets. mtry: # of variables to randomly sample at each split. mtry = start with 5 values evenly spaced across the range from 2 to p, # of predictors

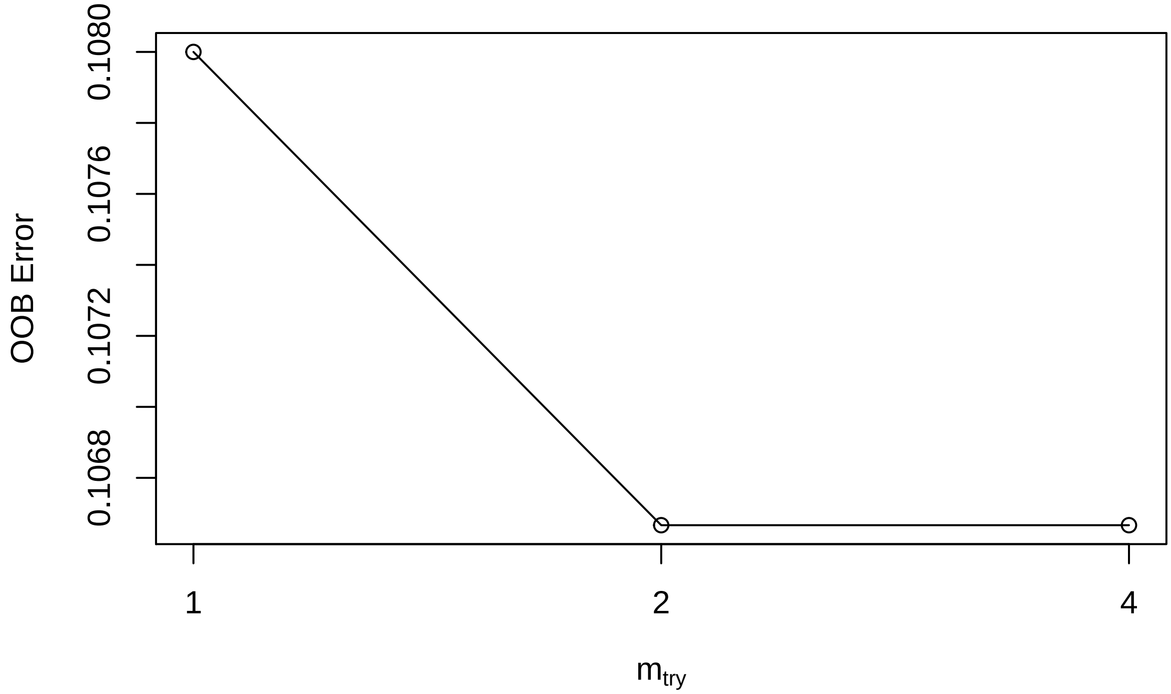
sampsize: the number of samples to train on. default = 63.25% average # of unique observations in a bootstrap sample. Lower sampsize reduces the training time but may increase bias High sampsize can increase accuracy but may end up overfitting sampsize between 60-80% range seems to work best

nodesize: minimum number of samples within the terminal nodes nodesize small -> deeper and more complex trees nodesize large -> shallow trees, less accuracy

maxnodes: maximum number of terminal nodes. high maxnodes -> deep, more complex trees

```r
set.seed(7231)
rf2 <- tuneRF(
  x          = df.train[features],
  y          = factor(df.train$Florence),
  ntreeTry   = 500,
  mtryStart  = 2,
  stepFactor = 2,
  improve    = 0.01,
  trace      = FALSE      # to not show real-time progress
)
```

```
## -0.0125 0.01
## 0 0.01
```



```
set.seed(11713)
rf2 <- randomForest(factor(Florence) ~ ., mtry = 4, ntree = 500, importance = TRUE, data = df.train)
rf2
```

```
##
## Call:
##  randomForest(formula = factor(Florence) ~ ., data = df.train,      mtry = 4, ntree = 500, importanc
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 10.47%
## Confusion matrix:
##      0 1  class.error
## 0 1337 1 0.0007473842
## 1  156 6 0.9629629630
```

```
CM.rf_train <- rf2$confusion
CM.rf_train
```

```
##      0 1  class.error
## 0 1337 1 0.0007473842
## 1  156 6 0.9629629630
```

```
OA.rf_train <- sum(diag(CM.rf_train))/sum(CM.rf_train)
```

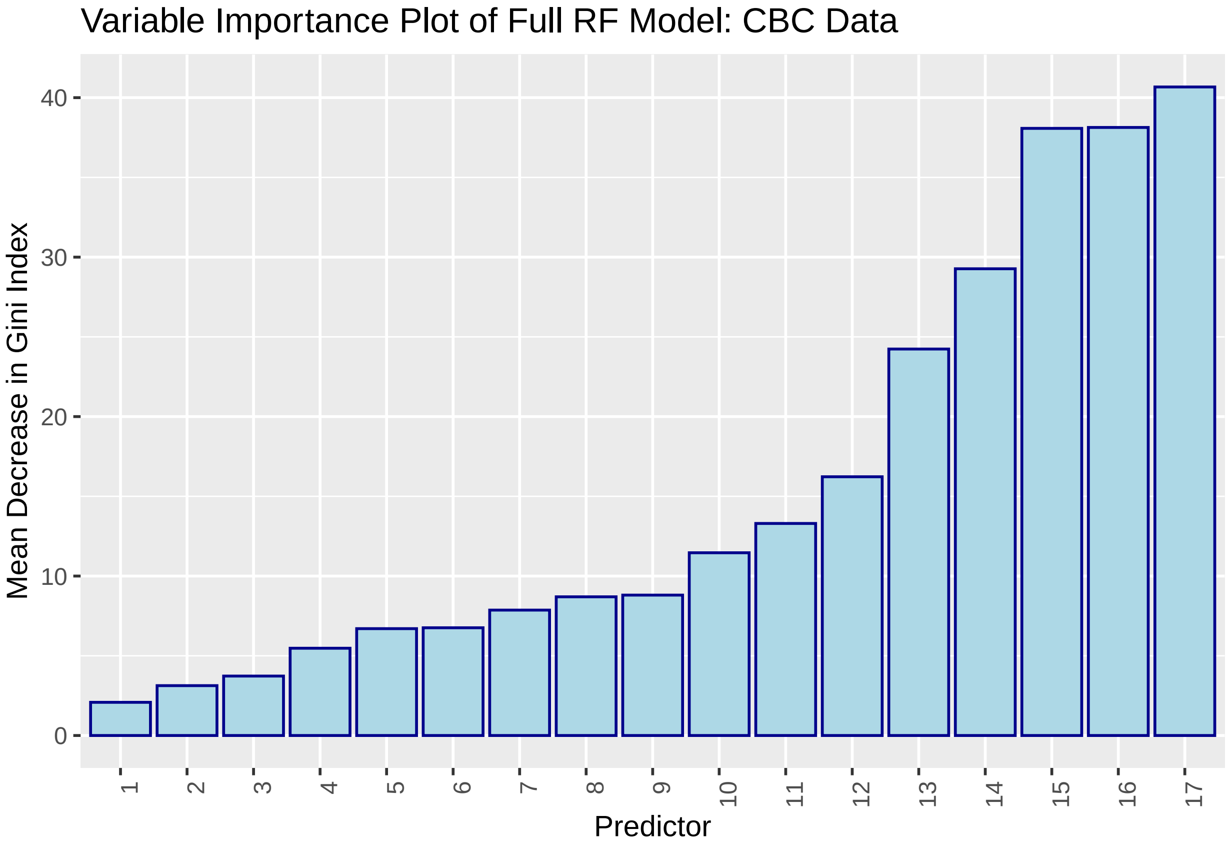
```
VI.FL <- as.data.frame(rf2$importance)
names(VI.FL)
```

```
## [1] "0"                    "1"                    "MeanDecreaseAccuracy"
## [4] "MeanDecreaseGini"
```

```
VIFL.sort <- VI.FL %>% arrange(MeanDecreaseGini)
# write.csv(VIFL.sort, "VIFL 120118.csv")
VIFL.sort$X <- rownames(VIFL.sort)
VIFL.sort$X <- factor(VIFL.sort$X, levels = VIFL.sort$X)

p.FL <- ggplot(VIFL.sort, aes(x = X, y = MeanDecreaseGini)) +
  geom_bar(stat = "identity", position = "dodge", fill = "lightblue", color = "darkblue") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  ylab("Mean Decrease in Gini Index") +
  xlab("Predictor") +
  ggtitle("Variable Importance Plot of Full RF Model: CBC Data")

p.FL
```



```
set.seed(11713)
rf3 <- randomForest(factor(Florence) ~ . , mtry = 4, ntree = 500, data = df.train)
rf3
```

```
##
## Call:
##  randomForest(formula = factor(Florence) ~ ., data = df.train,      mtry = 4, ntree = 500)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 10.6%
## Confusion matrix:
```

```
##      0 1 class.error
## 0 1334 4 0.002989537
## 1  155 7 0.956790123
```

```
CM.rf_train <- rf3$confusion
CM.rf_train
```

```
##      0 1 class.error
## 0 1334 4 0.002989537
## 1  155 7 0.956790123
```

```
OA.rf_train <- sum(diag(CM.rf_train))/sum(CM.rf_train)
OA.rf_train
```

```
## [1] 0.8934283
```

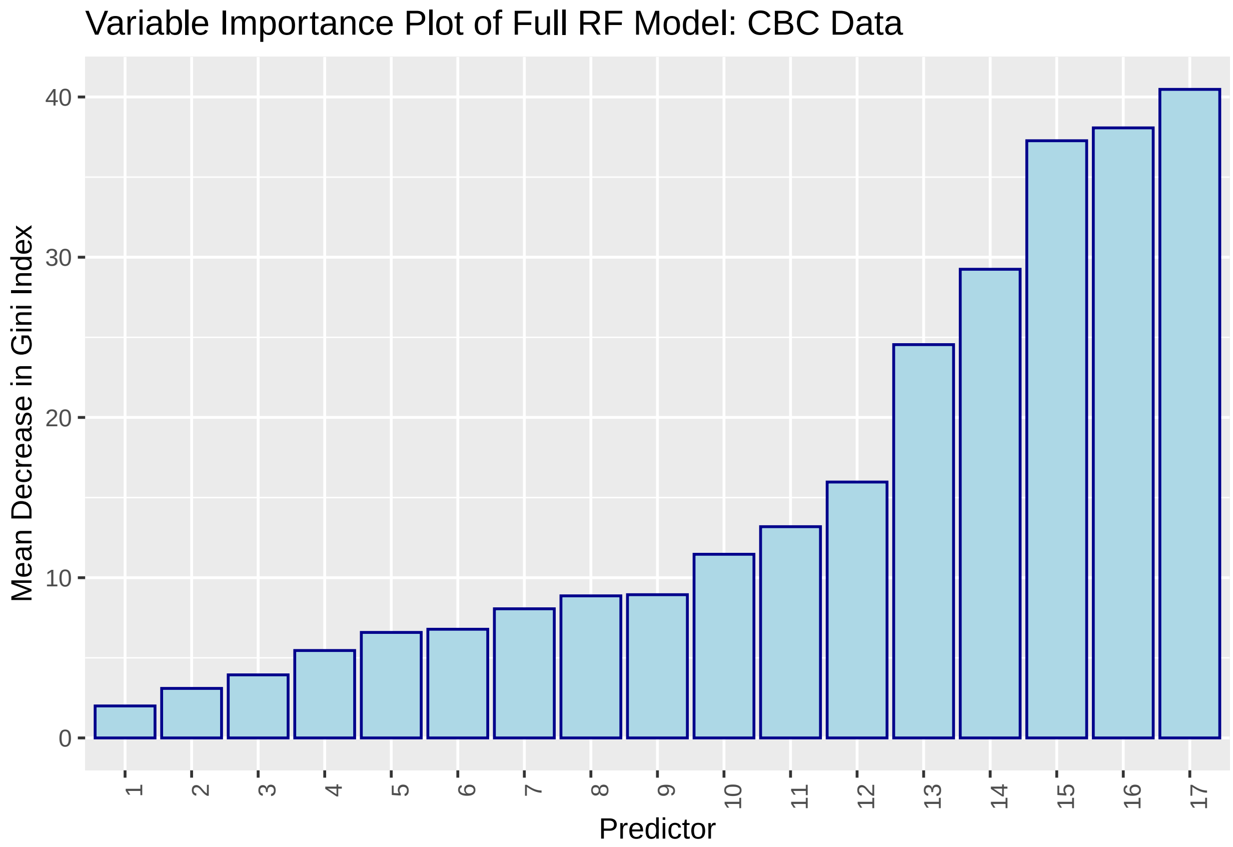
```
VI.FL <- as.data.frame(rf3$importance)
names(VI.FL)
```

```
## [1] "MeanDecreaseGini"
```

```
VIFL.sort <- VI.FL %>% arrange(MeanDecreaseGini)
# write.csv(VIFL.sort,"VIFL 120118.csv")
VIFL.sort$X <- rownames(VIFL.sort)
VIFL.sort$X <- factor(VIFL.sort$X, levels = VIFL.sort$X)

p.FL <- ggplot(VIFL.sort, aes(x = X, y = MeanDecreaseGini)) +
  geom_bar(stat = "identity", position = "dodge", fill = "lightblue", color = "darkblue") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  ylab("Mean Decrease in Gini Index") +
  xlab("Predictor") +
  ggtitle("Variable Importance Plot of Full RF Model: CBC Data")

p.FL
```



```
prf1_train <- PRF1(CM.rf_train)
prf1_train
```

```
## Precision_1    Recall_1        F1_1 Precision_0    Recall_0        F1_0
##        0.90        1.00        0.94        0.90        1.00        0.94
```

```
pred.test <- predict(rf3, df.test)
pred.test <- as.numeric(levels(pred.test)[pred.test])
CM.test <- table(df.test$Florence,round(pred.test))
CM.test
```

```
##
##       0   1
##   0 439   6
##   1  53   2
```

```
prf1_test <- PRF1(CM.test)
prf1_test
```

```
## Precision_1    Recall_1        F1_1 Precision_0    Recall_0        F1_0
##        0.89        0.99        0.94        0.89        0.99        0.94
```

Titanic

```
df <- read.csv("titanic3.csv", header = TRUE) %>%
  select(survived, pclass, sex, age, sibsp, parch) %>%
  filter(!is.na(pclass) & !is.na(sex) & !is.na(age) & !is.na(sibsp) & !is.na(parch)) %>%
  mutate(survived = as.numeric(survived))
```

```r
dim(df)  ## 1309 14
```

```
## [1] 1046    6
```

```r
head(df)
```

```
##   survived pclass sex     age sibsp parch
## 1        1      1   1 29.0000     0     0
## 2        1      1   0  0.9167     1     2
## 3        0      1   1  2.0000     1     2
## 4        0      1   0 30.0000     1     2
## 5        0      1   1 25.0000     1     2
## 6        1      1   0 48.0000     0     0
```

```r
tail(df)
```

```
##      survived pclass sex  age sibsp parch
## 1041        1      3   1 15.0     1     0
## 1042        0      3   0 45.5     0     0
## 1043        0      3   1 14.5     1     0
## 1044        0      3   0 26.5     0     0
## 1045        0      3   0 27.0     0     0
## 1046        0      3   0 29.0     0     0
```

```r
p <- 0.25
M <- p * nrow(df)
#set initial seed for repeatability
set.seed(113117)

holdout <- sample(1:nrow(df), M, replace = F)
df.train <- df[-holdout, ]
df.test  <- df[holdout, ]

dim(df.train) ## 982 14
```

```
## [1] 785   6
```

```r
dim(df.test)  ## 327 14
```

```
## [1] 261   6
```

```r
features <- setdiff(names(df.train), "survived")
rf1 <- randomForest(factor(survived) ~ ., data = df.train)
rf1
```

```
##
## Call:
##  randomForest(formula = factor(survived) ~ ., data = df.train)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 20.38%
## Confusion matrix:
##     0   1 class.error
## 0 417  45   0.0974026
## 1 115 208   0.3560372
```

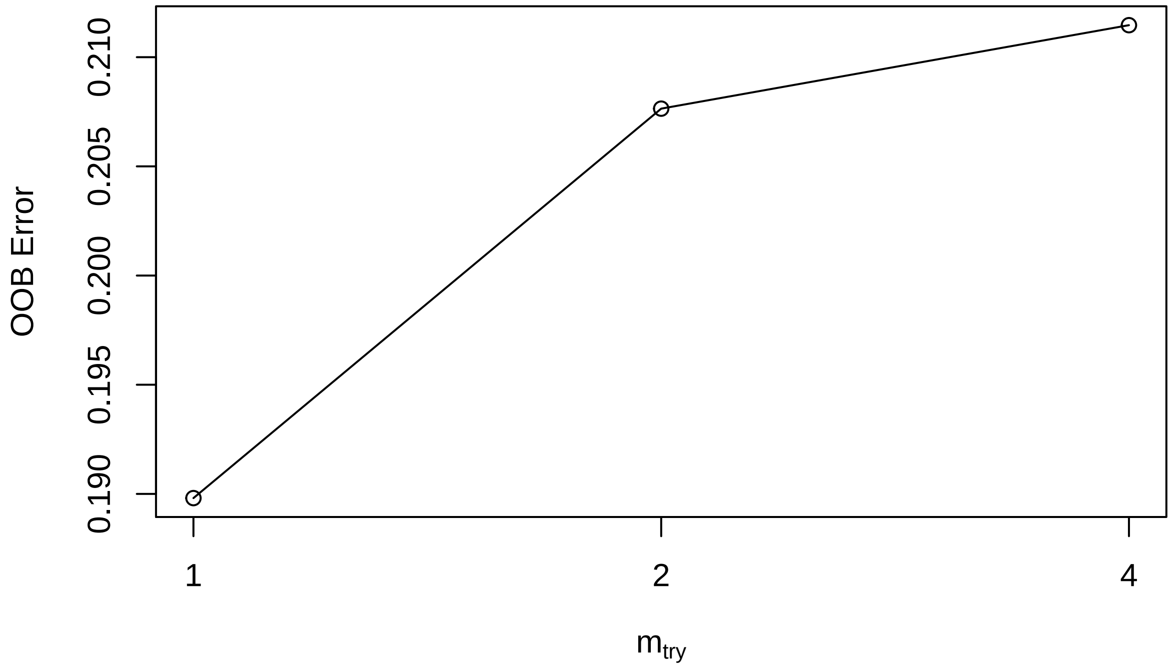
```
CM.rf_train <- rf1$confusion
CM.rf_train
```

```
##     0   1 class.error
## 0 417  45   0.0974026
## 1 115 208   0.3560372
```

```
OA.rf_train <- sum(diag(CM.rf_train))/sum(CM.rf_train)
```

```
set.seed(7231)
rf2 <- tuneRF(
  x           = df.train[features],
  y           = factor(df.train$survived),
  ntreeTry    = 500,
  mtryStart   = 2,
  stepFactor = 2,
  improve     = 0.01,
  trace       = FALSE      # to not show real-time progress
)
```

```
## 0.08588957 0.01
## -0.114094 0.01
```



```
set.seed(11713)
rf2 <- randomForest(factor(survived) ~ ., mtry = 4, ntree = 500, importance = TRUE, data = df.train)
rf2
```

```
##
## Call:
##  randomForest(formula = factor(survived) ~ ., data = df.train,      mtry = 4, ntree = 500, importanc
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 4
##
```

```
##         OOB estimate of  error rate: 21.15%
## Confusion matrix:
##     0   1 class.error
## 0 399  63   0.1363636
## 1 103 220   0.3188854
```

```
CM.rf_train <- rf2$confusion
CM.rf_train
```

```
##     0   1 class.error
## 0 399  63   0.1363636
## 1 103 220   0.3188854
```

```
OA.rf_train <- sum(diag(CM.rf_train))/sum(CM.rf_train)
```

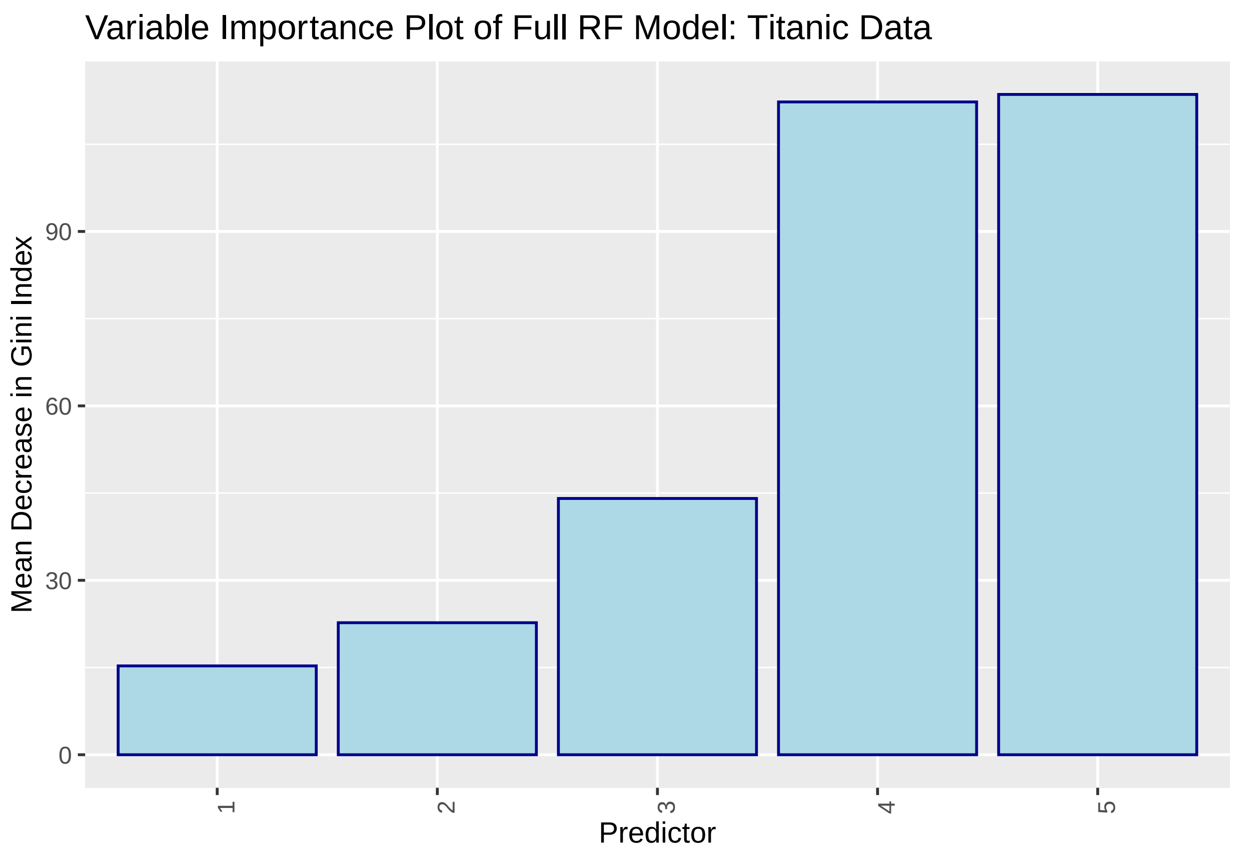
```
VI.FL <- as.data.frame(rf2$importance)
names(VI.FL)
```

```
## [1] "0"                    "1"                    "MeanDecreaseAccuracy"
## [4] "MeanDecreaseGini"
```

```
VIFL.sort <- VI.FL %>% arrange(MeanDecreaseGini)
# write.csv(VIFL.sort, "VIFL 120118.csv")
VIFL.sort$X <- rownames(VIFL.sort)
VIFL.sort$X <- factor(VIFL.sort$X, levels = VIFL.sort$X)

p.FL <- ggplot(VIFL.sort, aes(x = X, y = MeanDecreaseGini))+
  geom_bar(stat = "identity", position = "dodge", fill = "lightblue", color = "darkblue") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  ylab("Mean Decrease in Gini Index") +
  xlab("Predictor") +
  ggtitle("Variable Importance Plot of Full RF Model: Titanic Data")

p.FL
```



```
set.seed(11713)
rf3 <- randomForest(factor(survived) ~ . , mtry = 4, ntree = 500, data = df.train)
rf3
```

```
##
## Call:
##  randomForest(formula = factor(survived) ~ ., data = df.train,      mtry = 4, ntree = 500)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 20.89%
## Confusion matrix:
##     0   1 class.error
## 0 402  60   0.1298701
## 1 104 219   0.3219814
```

```
CM.rf_train <- rf3$confusion
CM.rf_train
```

```
##     0   1 class.error
## 0 402  60   0.1298701
## 1 104 219   0.3219814
```

```
OA.rf_train <- sum(diag(CM.rf_train))/sum(CM.rf_train)
OA.rf_train
```

```
## [1] 0.7906277
```

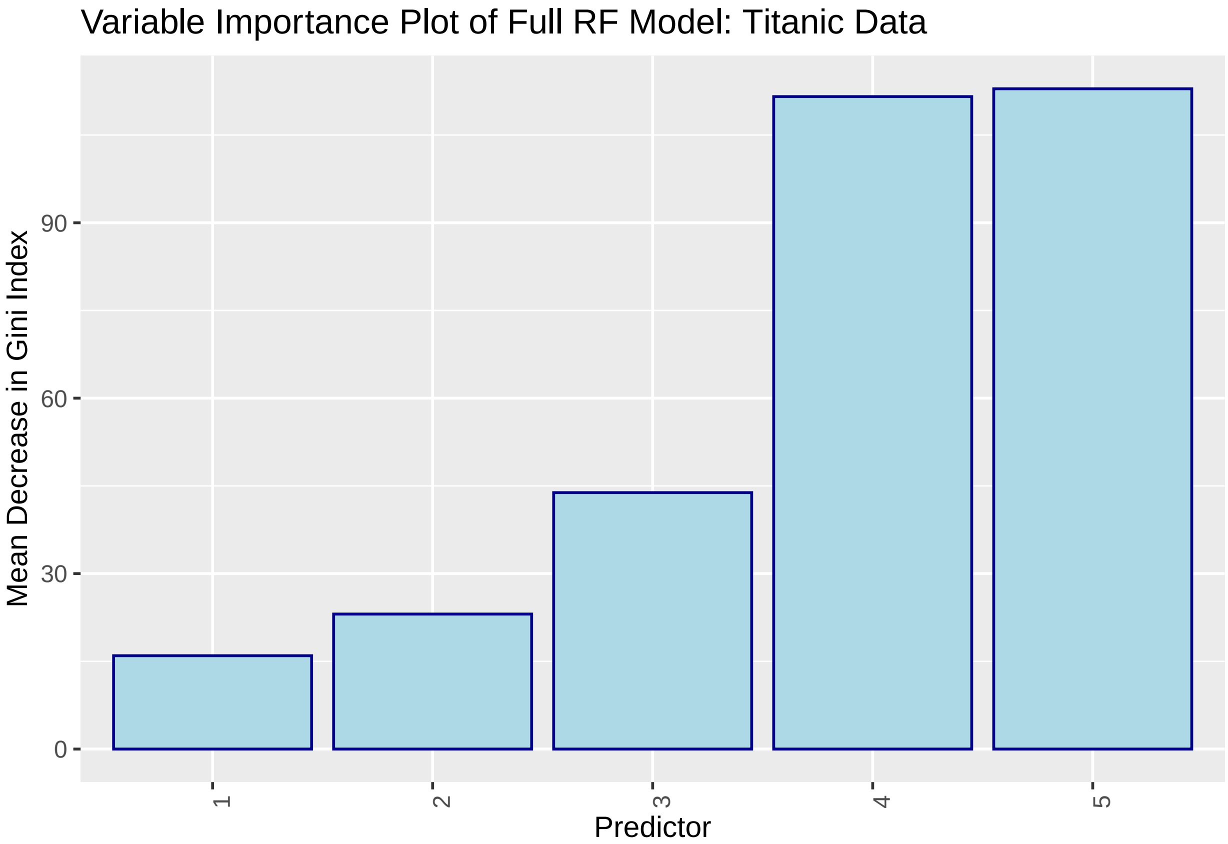
```
VI.FL <- as.data.frame(rf3$importance)
names(VI.FL)
```

```
## [1] "MeanDecreaseGini"
```

```
VIFL.sort <- VI.FL %>% arrange(MeanDecreaseGini)
# write.csv(VIFL.sort,"VIFL 120118.csv")
VIFL.sort$X <- rownames(VIFL.sort)
VIFL.sort$X <- factor(VIFL.sort$X, levels = VIFL.sort$X)

p.FL <- ggplot(VIFL.sort, aes(x = X, y = MeanDecreaseGini)) +
  geom_bar(stat = "identity", position = "dodge", fill = "lightblue", color = "darkblue") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  ylab("Mean Decrease in Gini Index") +
  xlab("Predictor") +
  ggtitle("Variable Importance Plot of Full RF Model: Titanic Data")

p.FL
```



```
prf1_train <- PRF1(CM.rf_train)
prf1_train
```

```
## Precision_1    Recall_1        F1_1 Precision_0    Recall_0        F1_0
##        0.79        0.87        0.83        0.79        0.87        0.83
```

```
pred.test <- predict(rf3, df.test)
pred.test <- as.numeric(levels(pred.test)[pred.test])
CM.test <- table(df.test$survived,round(pred.test))
CM.test
```

```
##
##       0   1
##   0 130  27
##   1  34  70
```

```
prf1_test <- PRF1(CM.test)
prf1_test
```

```
## Precision_1    Recall_1        F1_1 Precision_0    Recall_0        F1_0
##        0.79        0.83        0.81        0.79        0.83        0.81
```