HW5_Ronquillo

Melchor Ronquillo

2022-11-07

1) Download the titanic data set:

```
install.packages('ISLR2', repos = "http://cran.us.r-project.org")
```

```
##
## The downloaded binary packages are in
##  /var/folders/b4/vbzhgztj3tj299xgxnklp28c0000gn/T//Rtmpd1Madc/downloaded_packages
```

```
library(ISLR2)
test <- "/Users/melchorronquillo/Desktop/Files/Data/titanic/test.csv"
test<- data.frame(read.csv(test))
train <- "/Users/melchorronquillo/Desktop/Files/Data/titanic/train.csv"
train<- data.frame(read.csv(train))
gender <- "/Users/melchorronquillo/Desktop/Files/Data/titanic/gender_submission.csv"
gender <- data.frame(read.csv(gender))

### combine gender submission w/ test to determine accuracy/error later
testt <- merge(test, gender, by.x = "PassengerId", by.y = "PassengerId", all.x = TRUE, all.y = TRUE)
#testt

### change sex to binary values, male = 1, female = 0
train$Sex <- ifelse(train$Sex == "male",1,0)
### change Survived to factor for classification
train$Survived <- as.factor(train$Survived)
#train

### check for NaN values
train.nan <- colSums(is.na(train))
#train.nan
### Age is the only column with NaN values, impute usung mean age for NaN values
train.nonan <- train
train.nonan$Age[is.na(train.nonan$Age)] <- mean(train.nonan$Age,na.rm = TRUE)
#train.nonan



### repeat process for test data
testt$Sex <- ifelse(testt$Sex == "male",1,0)
testt$Survived <- as.factor(testt$Survived)
#testt

testt.nan <- colSums(is.na(testt))
#testt.nan
testt.nonan <- testt
```

```
testt.nonan$Age[is.na(testt.nonan$Age)] <- mean(testt.nonan$Age,na.rm = TRUE)
testt.nonan$Fare[is.na(testt.nonan$Fare)] <- mean(testt.nonan$Fare,na.rm = TRUE)
#testt.nonan
```

2) Build a classification tree to predict the variable "Survived". Report the cross validation error using k-fold cross validation with a reasonable value of k

```
### classification tree
install.packages('tree', repos = "http://cran.us.r-project.org")
```

```
##
## The downloaded binary packages are in
##  /var/folders/b4/vbzhgztj3tj299xgxnklp28c0000gn/T//Rtmpd1Madc/downloaded_packages
```

```
library(tree)
k <- 5
set.seed(104)
fold <- sample(1:5,nrow(train), replace = TRUE)
yhat = rep(NA,nrow(train))
for (i in 1:k){
tree.titanic <- tree(Survived ~ .  - PassengerId - Name - Ticket - Cabin - Embarked,
                     data = train[fold !=i,])
yhat[fold == i] <- predict(tree.titanic,train[fold == i,])[,2]
}

### find accuracy / misclassification
(table(yhat > 0.5, train$Survived))
```

```
##
##           0   1
##   FALSE 482 118
##   TRUE   67 224
```

```
class_tree.CVerr <- 1 - sum(diag(table(yhat > 0.5, train$Survived))) / nrow(train)
class_tree.CVerr
```

```
## [1] 0.2076319
```

```
### misclassification error from cross validation = %20.76
```

```
    Cross validation error using k-fold = %20.76
```

3) Use a random forest to predict the variable "Survived". Report the out-of-bag cross validation error.

```
### random forest
install.packages('randomForest', repos = "http://cran.us.r-project.org")
```

```
##
## The downloaded binary packages are in
##  /var/folders/b4/vbzhgztj3tj299xgxnklp28c0000gn/T//Rtmpd1Madc/downloaded_packages
```

```
library(randomForest)
```

```
## randomForest 4.7-1.1
```

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

```
rf.titanic <- randomForest(Survived ~ . - PassengerId - Name - Ticket - Cabin - Embarked , data = train
rf.titanic
```

```
##
## Call:
##  randomForest(formula = Survived ~ . - PassengerId - Name - Ticket -      Cabin - Embarked, data = t
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 16.95%
## Confusion matrix:
##     0   1 class.error
## 0 501  48  0.08743169
## 1 103 239  0.30116959
```

```
### Out of Bag estimate of error rate = %16.95
```

```
     Out of Bag estimate of error = %16.95
```

```
#yhelp("randomForest")
#str(rf.titanic2)
#varImpPlot(rf.titanic2)

yhat.rff <- predict(rf.titanic,  newdata = testt.nonan)
#yhat.rff
### random forest test accuracy / misclassification
(table(yhat.rff, testt.nonan$Survived))
```

```
##
## yhat.rff   0   1
##        0 253  31
##        1  13 121
```

```
rff.err <- 1 - sum(diag(table(yhat.rff, testt.nonan$Survived))) / nrow(testt.nonan)
rff.err
```

```
## [1] 0.1052632
```

```
### test misclassifcation error = %10.52
```