Homework 5

Chance Kang / A13605546 / csk025@ucsd.edu
Jayden Kim / A16271107 / s0k003@ucsd.edu
Sia Sheth / A16357789 / snsheth@ucsd.edu
Math 189
Spring 2023

Conceptual Question

1. A random forest is a learning algorithm that combines the predictions of multiple decision trees. During training, decision trees are built using random subsets of the data and features. When making a prediction, each tree independently classifies the input, and the final prediction is determined by majority voting (for classification) or averaging (for regression) the individual tree predictions. This aggregation of predictions improves accuracy and handles complex patterns.

Application Question

```
library(ISLR2)
library(MASS)
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:ISLR2':
##
##     Boston
```

```
library(randomForest)
```

```
## randomForest 4.7-1.1

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

```
library(tree)
library(caret)
```

```
## Loading required package: ggplot2

##
## Attaching package: 'ggplot2'

## The following object is masked from 'package:randomForest':
##
##     margin

## Loading required package: lattice
```

```
data(Bikeshare)
```

2.

```r
bike <- subset(Bikeshare, select = -c(casual, registered))

bike$season <- as.factor(bike$season)
bike$holiday <- as.factor(bike$holiday)
bike$weekday <- as.factor(bike$weekday)
bike$workingday <- as.factor(bike$workingday)

set.seed(123)
train<-sample(1:nrow(bike), nrow(bike)*0.8)
test<-(-train)
train_s <- bike[train,]
test_s <- bike[test,]

bike_lm <- lm(bikers ~ ., data = train_s)

bike_aic <- stepAIC(bike_lm, direction="both")
```

```
## Start:  AIC=59906.55
## bikers ~ season + mnth + day + hr + holiday + weekday + workingday +
##     weathersit + temp + atemp + hum + windspeed
##
##
## Step:  AIC=59906.55
## bikers ~ season + mnth + day + hr + holiday + weekday + weathersit +
##     temp + atemp + hum + windspeed
##
##              Df Sum of Sq      RSS   AIC
## - weekday     6     44089 39405218 59902
## <none>                    39361129 59907
## - atemp       1     12617 39373746 59907
## - day         1     16104 39377233 59907
## - holiday     1     30849 39391979 59910
## - temp        1     34504 39395633 59911
## - windspeed   1     53037 39414167 59914
## - season      3    405572 39766701 59971
## - mnth       11    770735 40131864 60019
## - hum         1    782568 40143697 60041
## - weathersit  3   1015040 40376170 60077
## - hr         23  43821191 83182320 65035
##
## Step:  AIC=59902.3
## bikers ~ season + mnth + day + hr + holiday + weathersit + temp +
##     atemp + hum + windspeed
##
##              Df Sum of Sq      RSS   AIC
## <none>                    39405218 59902
## - atemp       1     14004 39419223 59903
## - day         1     15755 39420974 59903
## + workingday  1       912 39404306 59904
## - holiday     1     26662 39431880 59905
## - temp        1     30767 39435986 59906
## + weekday     6     44089 39361129 59907
## - windspeed   1     53906 39459124 59910
## - season      3    405853 39811072 59967
```

```
## - mnth       11    780243 40185461 60016
## - hum         1    810524 40215743 60041
## - weathersit  3   1033430 40438648 60075
## - hr         23  43893745 83298963 65033
```

```r
bike_pred <- predict(bike_aic, data = test_s)
bike_rmse <- sqrt(mean((test_s$bikers - bike_pred)^2))

summary(bike_aic)
```

```
##
## Call:
## lm(formula = bikers ~ season + mnth + day + hr + holiday + weathersit +
##     temp + atemp + hum + windspeed, data = train_s)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -285.52  -45.58   -6.59   41.05  408.81
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 -20.7165     7.4519  -2.780 0.005450 **
## season2                      18.2797     5.7795   3.163 0.001569 **
## season3                      27.4474     6.7434   4.070 4.75e-05 ***
## season4                      47.9867     5.7522   8.342  < 2e-16 ***
## mnthFeb                      13.0663     5.6582   2.309 0.020958 *
## mnthMarch                    19.4728     7.9805   2.440 0.014710 *
## mnthApril                    45.1995    12.1054   3.734 0.000190 ***
## mnthMay                      82.2476    14.8509   5.538 3.17e-08 ***
## mnthJune                     74.3382    17.7428   4.190 2.83e-05 ***
## mnthJuly                     49.7287    21.0026   2.368 0.017924 *
## mnthAug                      67.4224    23.7455   2.839 0.004533 **
## mnthSept                     89.1862    26.4720   3.369 0.000758 ***
## mnthOct                      80.3054    29.5996   2.713 0.006683 **
## mnthNov                      71.8834    32.7640   2.194 0.028271 *
## mnthDec                      79.0718    35.6130   2.220 0.026430 *
## day                          -0.1746     0.1053  -1.657 0.097518 .
## hr1                         -14.2819     6.3202  -2.260 0.023869 *
## hr2                         -18.5274     6.3268  -2.928 0.003418 **
## hr3                         -27.1657     6.3247  -4.295 1.77e-05 ***
## hr4                         -33.2814     6.4348  -5.172 2.38e-07 ***
## hr5                         -19.7886     6.3331  -3.125 0.001788 **
## hr6                          25.0996     6.2978   3.985 6.80e-05 ***
## hr7                         126.3960     6.2324  20.280  < 2e-16 ***
## hr8                         225.5370     6.2296  36.204  < 2e-16 ***
## hr9                         117.3277     6.3467  18.486  < 2e-16 ***
## hr10                         77.7651     6.3048  12.334  < 2e-16 ***
## hr11                         96.0577     6.3629  15.096  < 2e-16 ***
## hr12                        126.4169     6.4097  19.723  < 2e-16 ***
## hr13                        124.4616     6.4234  19.376  < 2e-16 ***
## hr14                        115.5118     6.5465  17.645  < 2e-16 ***
## hr15                        122.0046     6.4537  18.905  < 2e-16 ***
## hr16                        168.0769     6.5509  25.657  < 2e-16 ***
## hr17                        276.1538     6.4336  42.924  < 2e-16 ***
## hr18                        266.4758     6.3887  41.710  < 2e-16 ***
```

```
## hr19                        180.8234     6.3369  28.535  < 2e-16 ***
## hr20                        123.2173     6.2921  19.583  < 2e-16 ***
## hr21                         84.7781     6.2882  13.482  < 2e-16 ***
## hr22                         56.9888     6.3019   9.043  < 2e-16 ***
## hr23                         26.1961     6.3326   4.137 3.57e-05 ***
## holiday1                    -12.2368     5.6761  -2.156 0.031131 *
## weathersitcloudy/misty       -2.9104     2.2860  -1.273 0.203009
## weathersitlight rain/snow -47.6953     3.6427 -13.093  < 2e-16 ***
## weathersitheavy rain/snow -62.7723    76.0453  -0.825 0.409139
## temp                        100.5022    43.3970   2.316 0.020594 *
## atemp                        71.6077    45.8307   1.562 0.118230
## hum                         -77.0758     6.4843 -11.886  < 2e-16 ***
## windspeed                   -26.1460     8.5294  -3.065 0.002182 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 75.74 on 6869 degrees of freedom
## Multiple R-squared:  0.6832, Adjusted R-squared:  0.681
## F-statistic:   322 on 46 and 6869 DF,  p-value: < 2.2e-16
```

```
print(bike_rmse)
```

```
## [1] 174.3052
```

```
print(sd(bike$bikers))
```

```
## [1] 133.7979
```

RMSE value surpassed standard deviation of `bikers`, hence not a great performance.

3. The `casual` and `registered` variables were removed from the set. The following variables were already given and kept as factors: `mnth`, `hr`, `weathersit`.

The following variables were converted to factors:

- `holiday` and `workinday` variable uses 1 for yes and 0 for no
- `season` uses 1,2,3,4 for Winter, Spring, Summer, and Fall
- `weekday` uses 0 ~ 6 for Sunday, Monday, Tuesday, etc hence immediate that they are categorical.

Rest of the variables were kept the same (numerical).

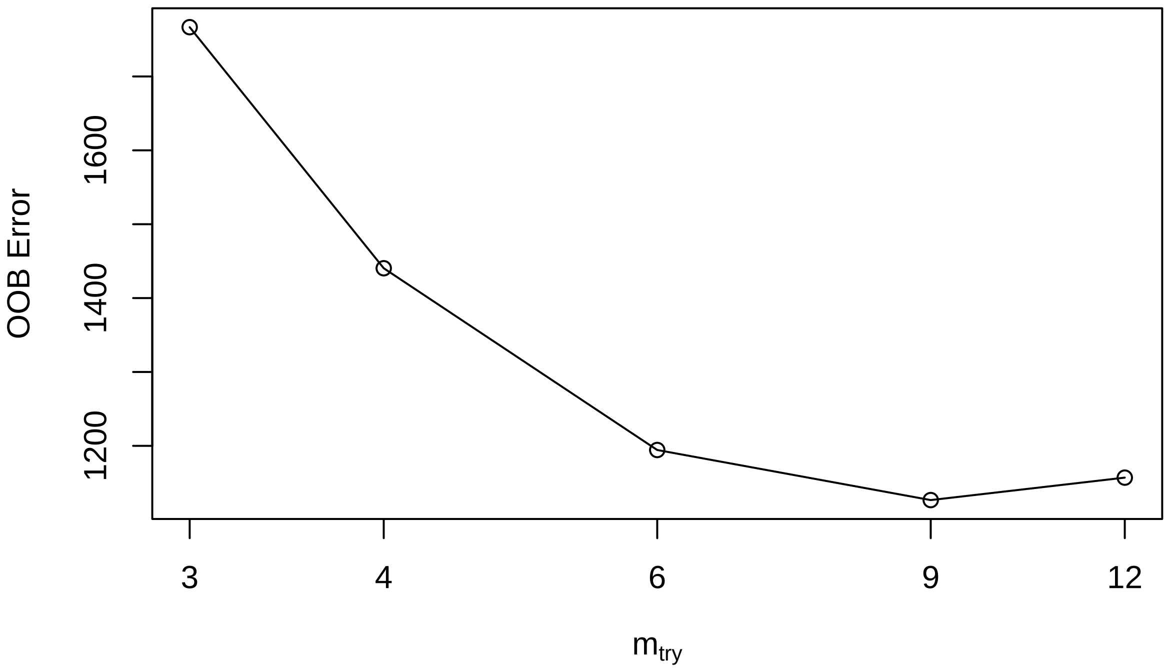
As suggested on course Piazza forum, bi-direction `stepAIC` function from `MASS` package was utilized for variable selection.

We split the train and test set as 80:20 ratio using random sample, for many previous homework applied the same ratio.

4.

```
set.seed(123)
bike_rf <- randomForest(bikers~., data = bike, subset = train, mtry = 1, ntree = 100, importance = TRUE
x.train = train_s[,1:12]
y.train = train_s[,13]
set.seed(123)
tuneRF(x = x.train, y = y.train, ntreeTry = 200, mtrystart = 2, stepFactor = 1.5, improve = 0.01, trace
```

```
## -0.2265298 0.01
## 0.1707941 0.01
## 0.05673303 0.01
## -0.02695025 0.01
```




```
##    mtry OOBError
## 3     3 1766.672
## 4     4 1440.382
## 6     6 1194.373
## 9     9 1126.613
## 12   12 1156.975
```

```
bike_rf_tuned <- randomForest(bikers~., data = train_s, ntree = 200, mtry = 9, importance = TRUE)
bike_rf_pred <- predict(bike_rf_tuned, newdata = test_s)
bike_rf_rmse <- sqrt(mean((test_s$bikers - bike_rf_pred)^2))

bike_rf_rmse
```

```
## [1] 34.05884
```

5. The `casual` and `registered` variables were removed from the set. The following variables were already given and kept as factors: `mnth`, `hr`, `weathersit`.

The following variables were converted to factors:

- `holiday` and `workinday` variable uses 1 for yes and 0 for no
- `season` uses 1,2,3,4 for Winter, Spring, Summer, and Fall
- `weekday` uses 0 ~ 6 for Sunday, Monday, Tuesday, etc hence immediate that they are categorical.

Rest of the variables were kept the same (numerical).

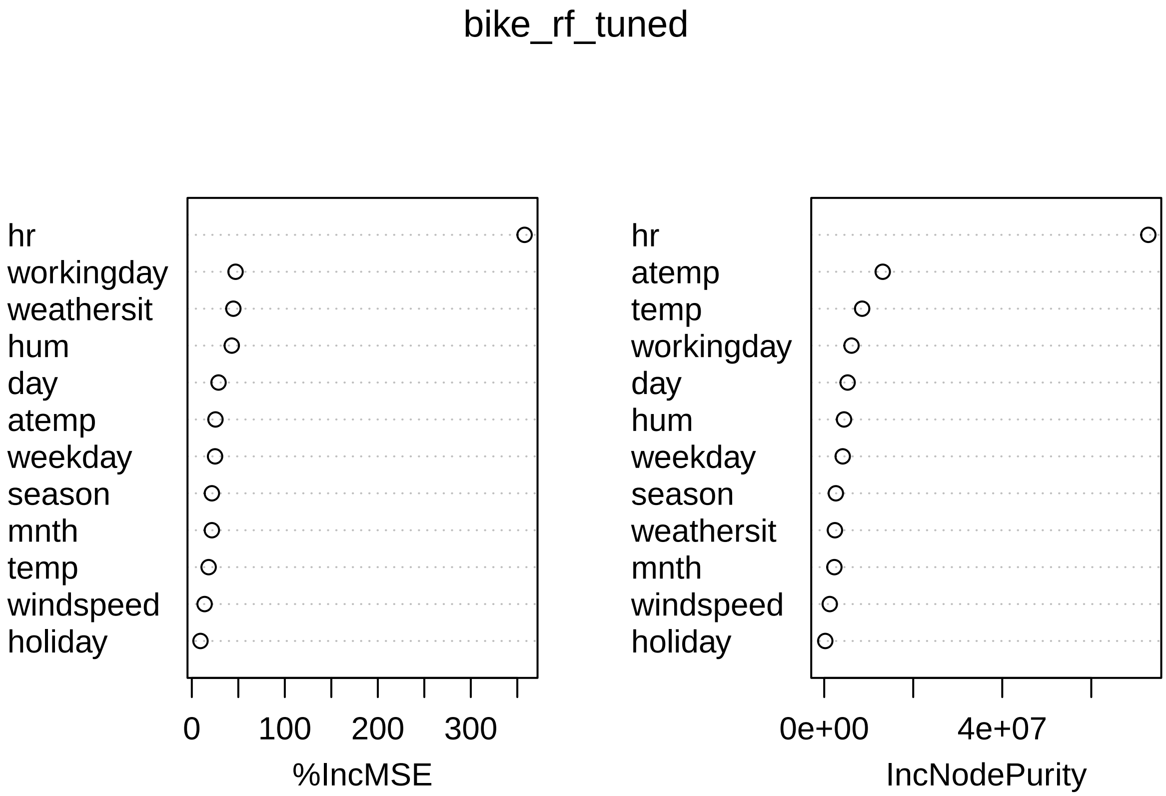
We split the train and test set as 80:20 ratio using random sample, for many previous homework applied the same ratio.

We have attempted a `k = 5` fold cross validation attempting to split the training set in to 5 folds, but due to the time consumption we have resorted to fitting an initial (base) fit with `ntree = 100` and `mtry = 1` then improving the fit through the `tuneRF()` function of the same package to fine tune the parameters.

As the above shows, the OOB error is the lowest when `mtry = 9`, hence model fit and prediction were proceeded using `mtry = 9`. The random forest performed far better when comparing the RMSE.

6.

```
varImpPlot(bike_rf_tuned)
```



Both models considers `hr` as an important predictor. The linear model removed `workingday` and `weekday` variable and found some categories of `mnth` and `season` to be important, whereas the random forest seems to find `hr` the only important variable relative to others.

Contribution

Our group homework process goes as the following:
1. Each member attempts to complete the homework
2. Compare and discuss the answers
3. Complete a finalized version to submit

All members have contributed about the equal amount to complete this homework.