Ex9 - Random Forest Classifier

Register Number:19BLC1186

Name:Tarun Sidhu

Lab Exercise No:9

Date:26/4/2021

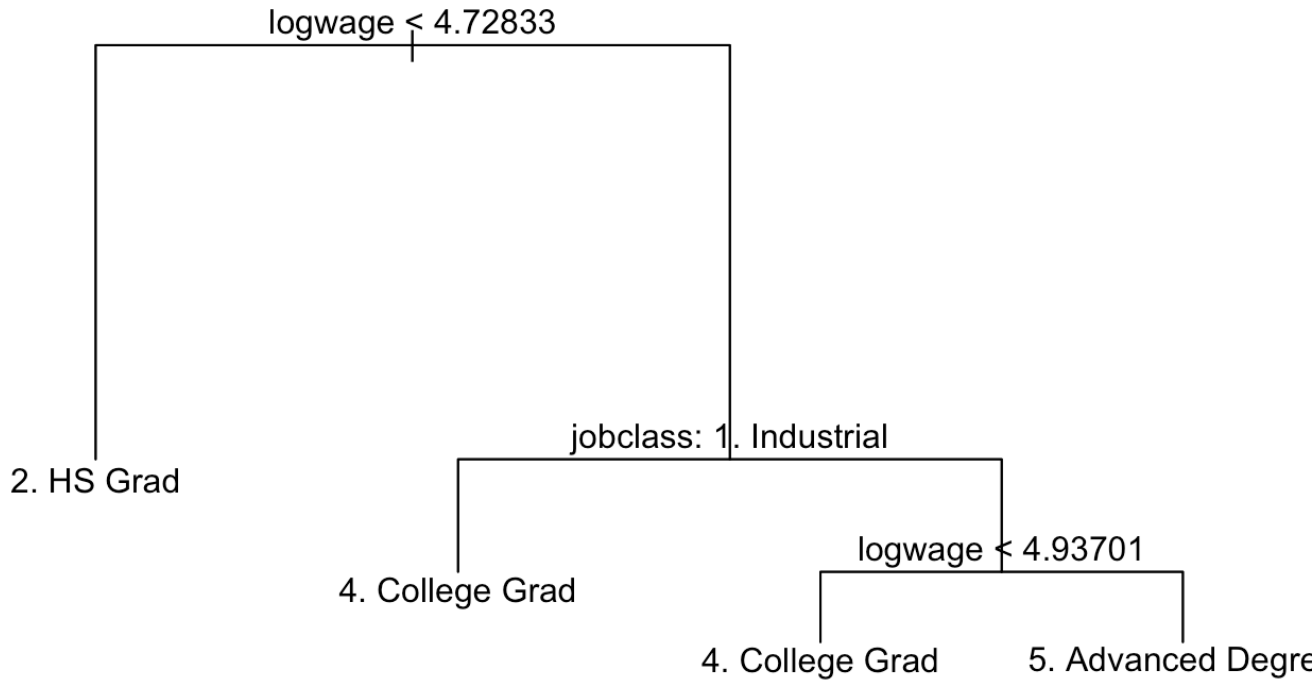
Dataset: Wage

Task:Perform a Random Forest Classification and put a for loop as shown in the given video

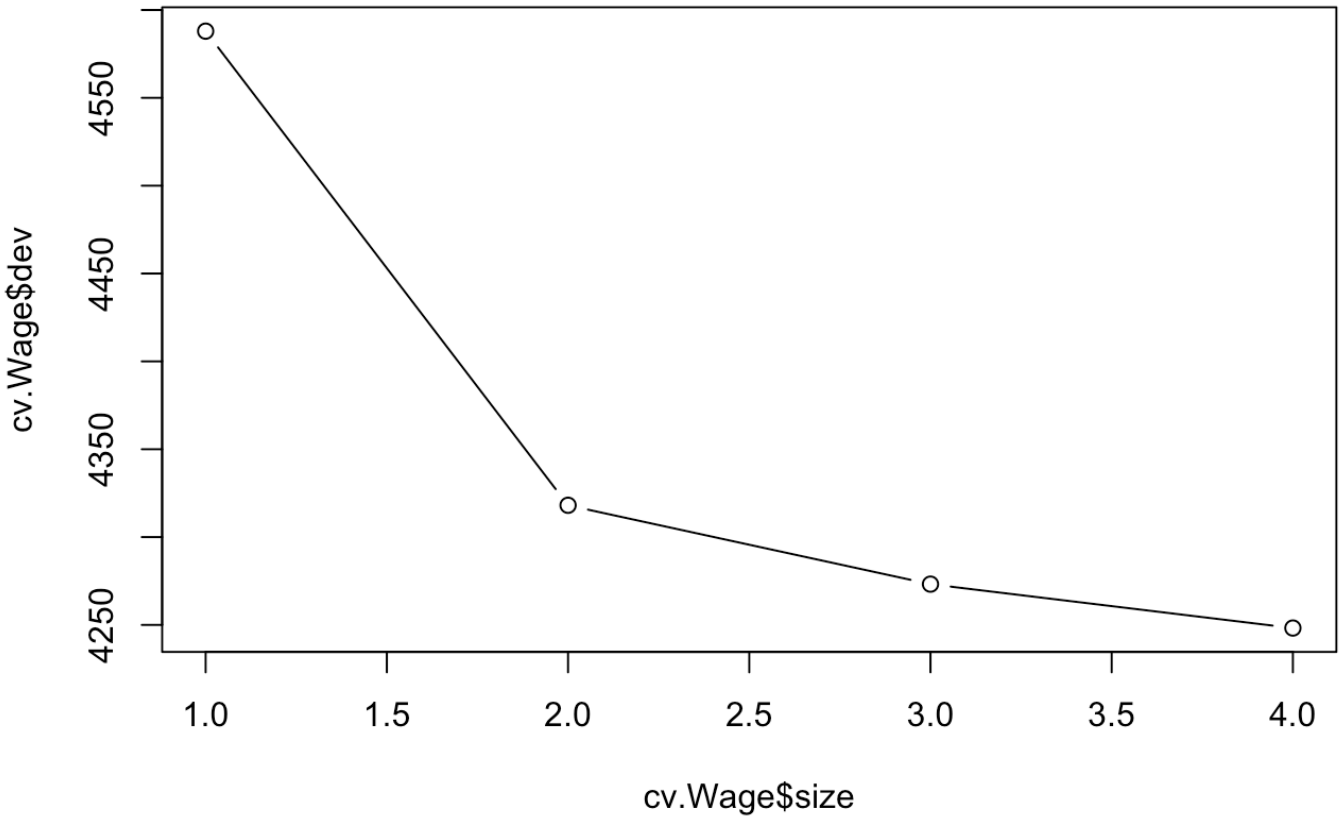
```
# Fitting Regression Trees
library(tree)
library(ISLR)
library(MASS)
set.seed(1)
train = sample(1:nrow(Wage), nrow(Wage)/2)
tree.Wage=tree(education~.,Wage,subset=train)
summary(tree.Wage)
```

```
##
## Classification tree:
## tree(formula = education ~ ., data = Wage, subset = train)
## Variables actually used in tree construction:
## [1] "logwage"  "jobclass"
## Number of terminal nodes:  4
## Residual mean deviance:  2.789 = 4172 / 1496
## Misclassification error rate: 0.6073 = 911 / 1500
```

```
plot(tree.Wage)
text(tree.Wage,pretty=0)
```



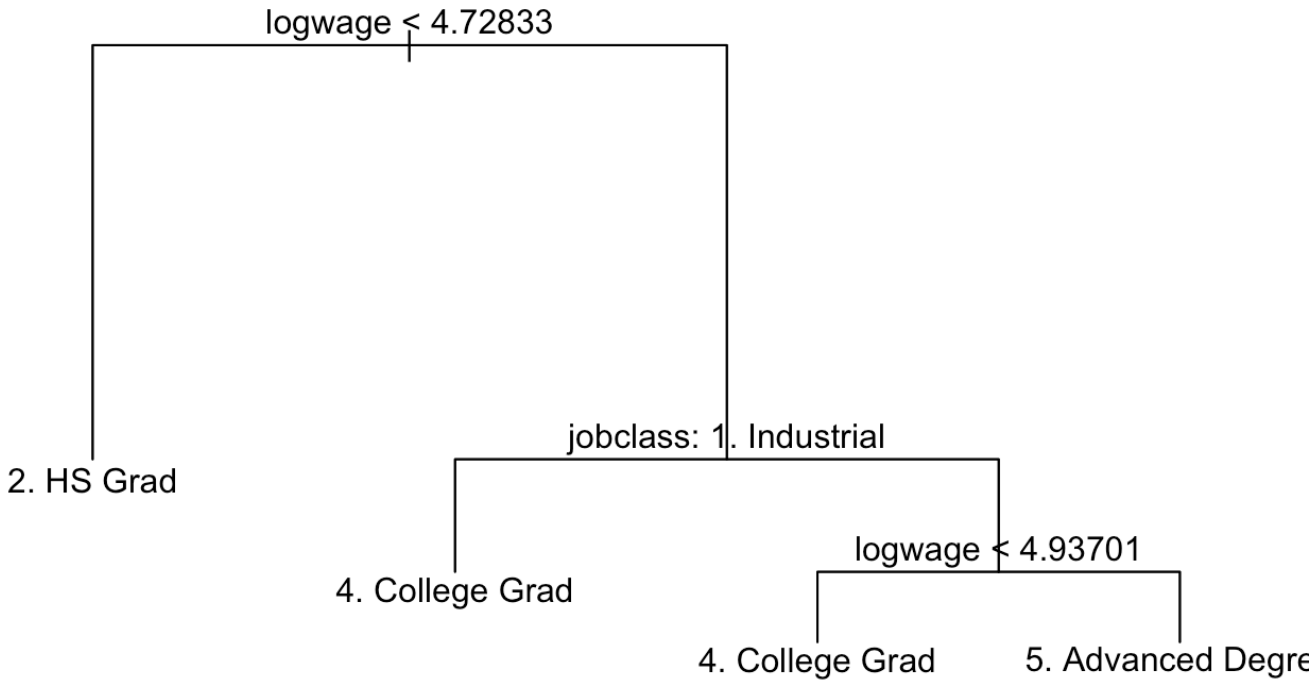

```
cv.Wage=cv.tree(tree.Wage)
plot(cv.Wage$size,cv.Wage$dev,type='b')
```



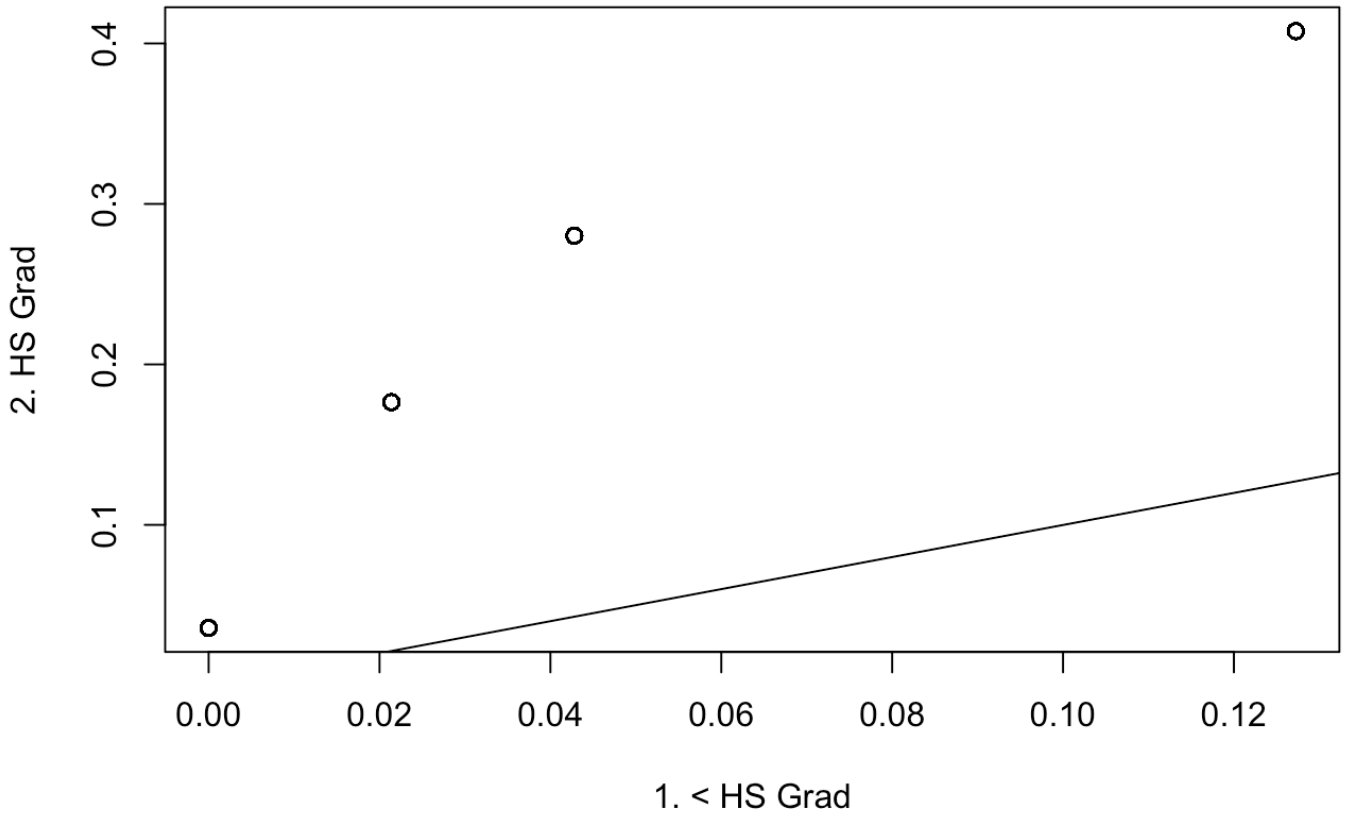

```
prune.Wage=prune.tree(tree.Wage,best=5)
```

```
## Warning in prune.tree(tree.Wage, best = 5): best is bigger than tree size
```

```
plot(prune.Wage)
text(prune.Wage,pretty=0)
```




```
yhat=predict(tree.Wage,newdata=Wage[-train,])
Wage.test=Wage[-train,"Education"]
plot(yhat,Wage.test)
abline(0,1)
```




```
mean((yhat-Wage.test)^2)
```

```
## [1] NaN
```

```
# Bagging and Random Forests

library(randomForest)
```

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

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

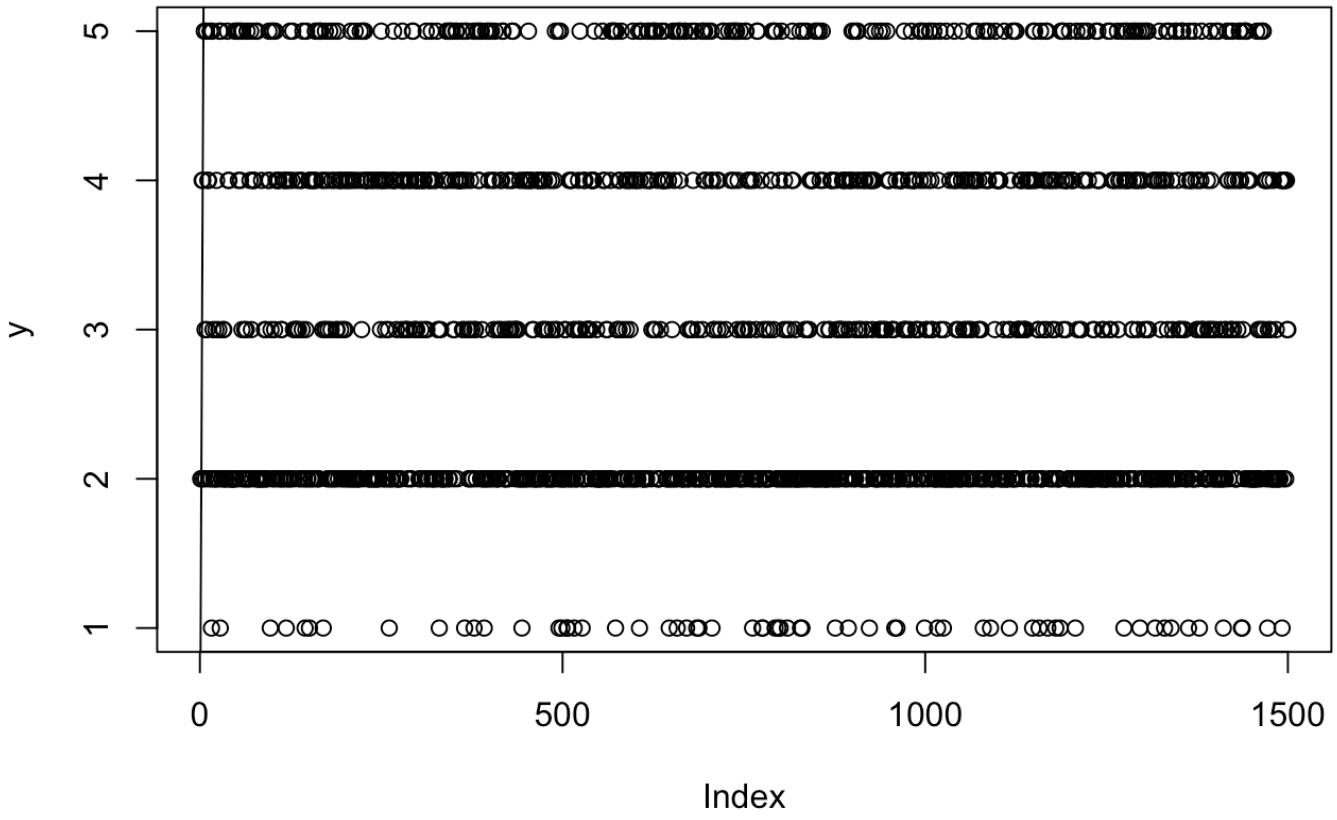
```
set.seed(1)
bag.Wage=randomForest(education~.,data=Wage,subset=train,mtry=13,importance=TRUE)
```

```
## Warning in randomForest.default(m, y, ...): invalid mtry: reset to within valid
## range
```

```
bag.Wage
```

```
##
## Call:
##  randomForest(formula = education ~ ., data = Wage, mtry = 13,      importance
= TRUE, subset = train)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 10
##
##         OOB estimate of  error rate: 67.8%
## Confusion matrix:
##                     1. < HS Grad 2. HS Grad 3. Some College 4. College Grad
## 1. < HS Grad                  10         84              21               12
## 2. HS Grad                    27        241              93               91
## 3. Some College               11        139              64               75
## 4. College Grad                9        129              66               91
## 5. Advanced Degree             0         32              32               74
##                     5. Advanced Degree class.error
## 1. < HS Grad                         1   0.9218750
## 2. HS Grad                          21   0.4904863
## 3. Some College                     33   0.8012422
## 4. College Grad                     67   0.7486188
## 5. Advanced Degree                  77   0.6418605
```

```
yhat.bag = predict(bag.Wage,newdata=Wage[-train,])
plot(yhat.bag, Wage.test)
abline(0,1)
```




```
mean((yhat.bag-Wage.test)^2)
```

```
## Warning in Ops.factor(yhat.bag, Wage.test): '-' not meaningful for factors
```

```
## [1] NA
```

```
bag.Wage=randomForest(education~.,data=Wage,subset=train,mtry=13,ntree=25)
```

```
## Warning in randomForest.default(m, y, ...): invalid mtry: reset to within valid
## range
```

```
yhat.bag = predict(bag.Wage,newdata=Wage[-train,])
mean((yhat.bag-Wage.test)^2)
```

```
## Warning in Ops.factor(yhat.bag, Wage.test): '-' not meaningful for factors
```

```
## [1] NA
```

```
set.seed(1)
rf.Wage=randomForest(education~.,data=Wage,subset=train,mtry=6,importance=TRUE)
yhat.rf = predict(rf.Wage,newdata=Wage[-train,])
mean((yhat.rf-Wage.test)^2)
```

```
## Warning in Ops.factor(yhat.rf, Wage.test): '-' not meaningful for factors
```

```
## [1] NA
```

```
importance(rf.Wage)
```

```
##             1. < HS Grad 2. HS Grad 3. Some College 4. College Grad
## year       -0.0793252614  0.3845728       0.0778268       2.0341287
## age         0.9594635455  1.7519811       1.0537875      -0.6432954
## maritl      3.4061761769  6.9322919       3.6131096       2.0988452
## race       -0.0816198690  4.6241024       2.9291564      -0.4524506
## region      0.0000000000  0.0000000       0.0000000       0.0000000
## jobclass    1.9668960509  2.8529161      -4.5109726       4.8465867
## health     -0.0001129919  5.1131494      -3.7068436      -0.5347277
## health_ins  6.7698701377  1.2329665      -6.7815167      -7.6505749
## logwage     8.5617389326 12.1742357      -0.9647908       0.9403824
## wage        8.4399177692 11.2533127      -0.6014296       1.1757907
##            5. Advanced Degree MeanDecreaseAccuracy MeanDecreaseGini
## year                 1.785247             1.837703        165.59534
## age                  7.544974             4.303833        303.33060
## maritl              -1.199283             7.990316         60.73061
## race                 3.585829             5.412398         59.54323
## region               0.000000             0.000000          0.00000
## jobclass            27.739871            14.628208         38.32821
## health               1.689627             1.951308         43.01624
## health_ins           1.932245            -2.892992         38.60155
## logwage             24.648261            22.159665        215.46698
## wage                23.357716            22.109760        215.21954
```

```
varImpPlot(rf.Wage)

oob.err=double(13)
test.err=double(13)
for (mtry in 1:13){
  fit=randomForest(education~.,data = Wage,subset=train,mtry=mtry,ntree=400)
  oob.err[mtry]=fit$mse[400]
  pred=predict(fit,Wage[-train,])
  test.err[mtry]=with(Wage[-train,],mean((education-age)^2))
  cat(mtry," ")
}
```

```
## Error in oob.err[mtry] <- fit$mse[400]: replacement has length zero
```

```
matplot(1:mtry,cbind(test.err,oob.err),pch=19,col=c("red","blue"),type = "b",ylab=
"Mean Squared Error")
```

```
## Error in matplot(1:mtry, cbind(test.err, oob.err), pch = 19, col = c("red", : '
x' and 'y' must have same number of rows
```

```
legend("topright",legend=c("00b","Test"),pch=19,col=c("red","blue"))
```