Assignment 3

Kyle Lee 27118158

2019

Q1

```
#***************Data Setup****************#
set.seed(1234)
job <- read.csv("jobs.csv", header = TRUE)
job <- data.frame(job[-c(1,16)])
job <- job[!(job$treat == 1 & job$comply ==0),]
job$treat <- as.factor(job$treat)
job$comply <- as.factor(job$comply)
#Setting Train and Test data set
observ_num = nrow(job)
trainindex <- sample(1:nrow(job), 450)
job_train <- job[trainindex, ]
job_test <- job[-trainindex, ]
```

```
# #***************Mixture Model Clustering***************#
# library(teigen)
# df_teigen <- subset(job, select=c(depress2, depress1))
# car_teigen <- teigen(df_teigen, 2,  models = "all", init="kmeans", scale = TRUE, gauss =FALSE)
# plot(car_teigen)
#
#
# df_teigen <- subset(job, select=c(income, job_seek))
# car_teigen <- teigen(df_teigen, 2,  models = "all", init="kmeans", scale = TRUE, gauss =FALSE)
# plot(car_teigen)
```

```
#***************Random Forest****************#
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 3.5.3
```

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

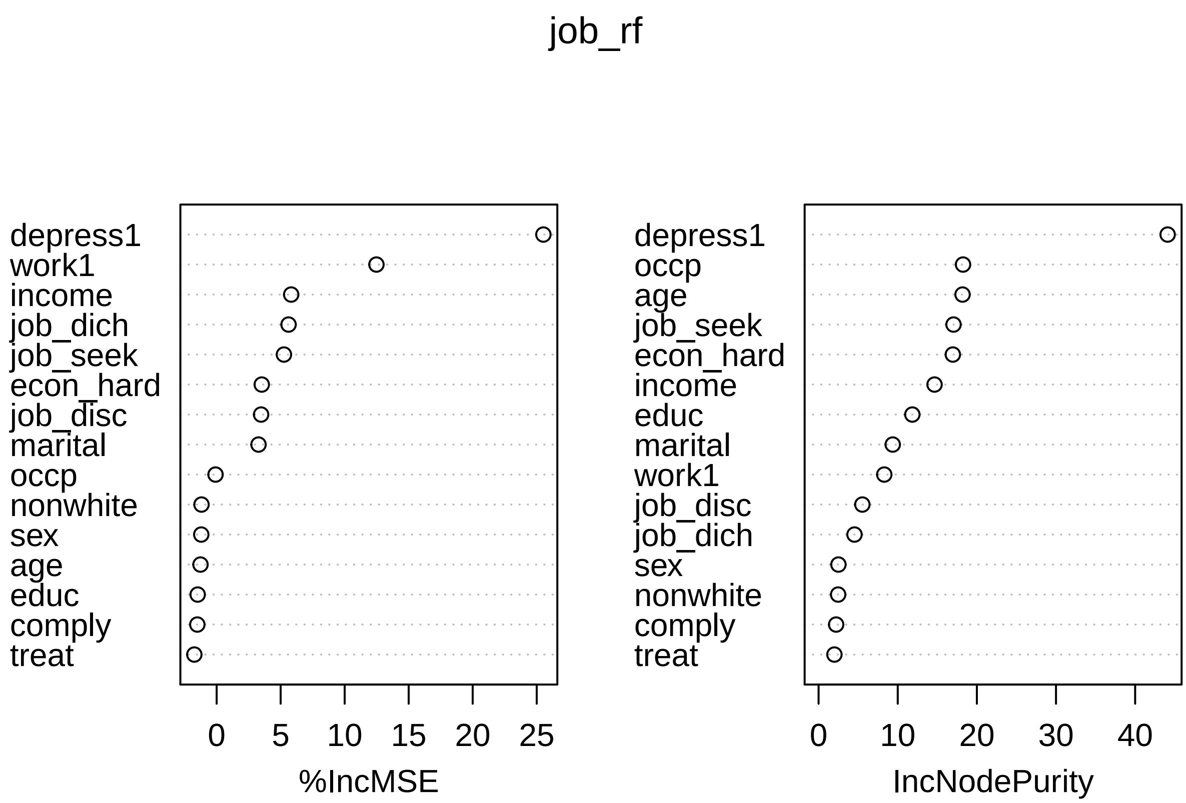
```
## Type rfNews() to see new features/changes/bug fixes.
```

```
set.seed(1234)

job_rf <- randomForest(depress2~., data=job_train, importance=TRUE, mtry=5)

job_predicted <- predict(job_rf, job_test)
MSE <- mean((job_predicted-job_test$depress2)^2)
MSE
```

```
## [1] 0.3514462
```

```
# plot(job_predicted, col='red' )
# plot(job_test$depress1)
#
#
```

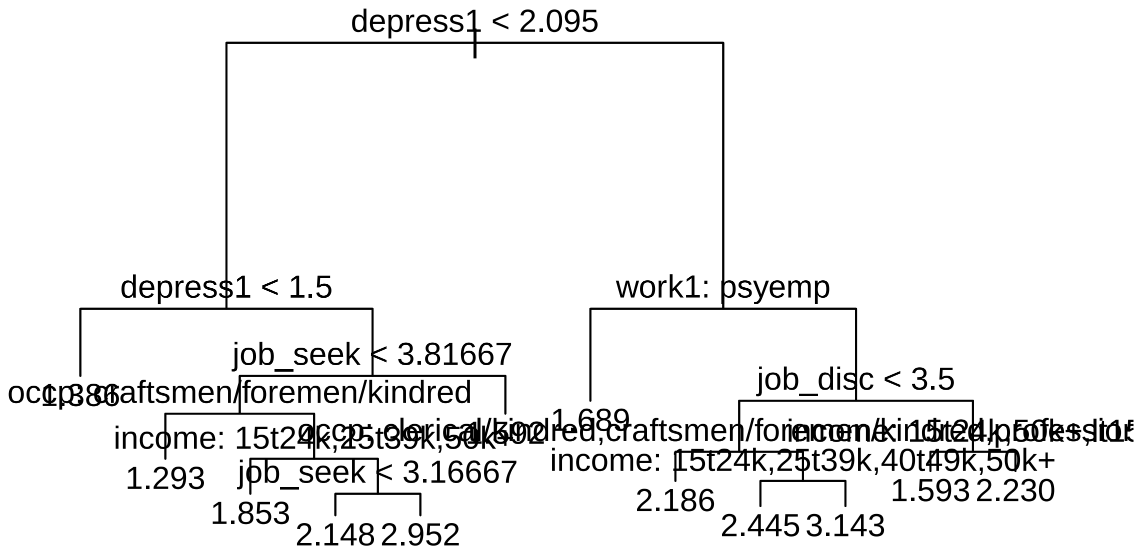
```r
# plot((job_predicted-job_test$depress1)^2)
#
# importance(job_rf)
varImpPlot(job_rf)
```



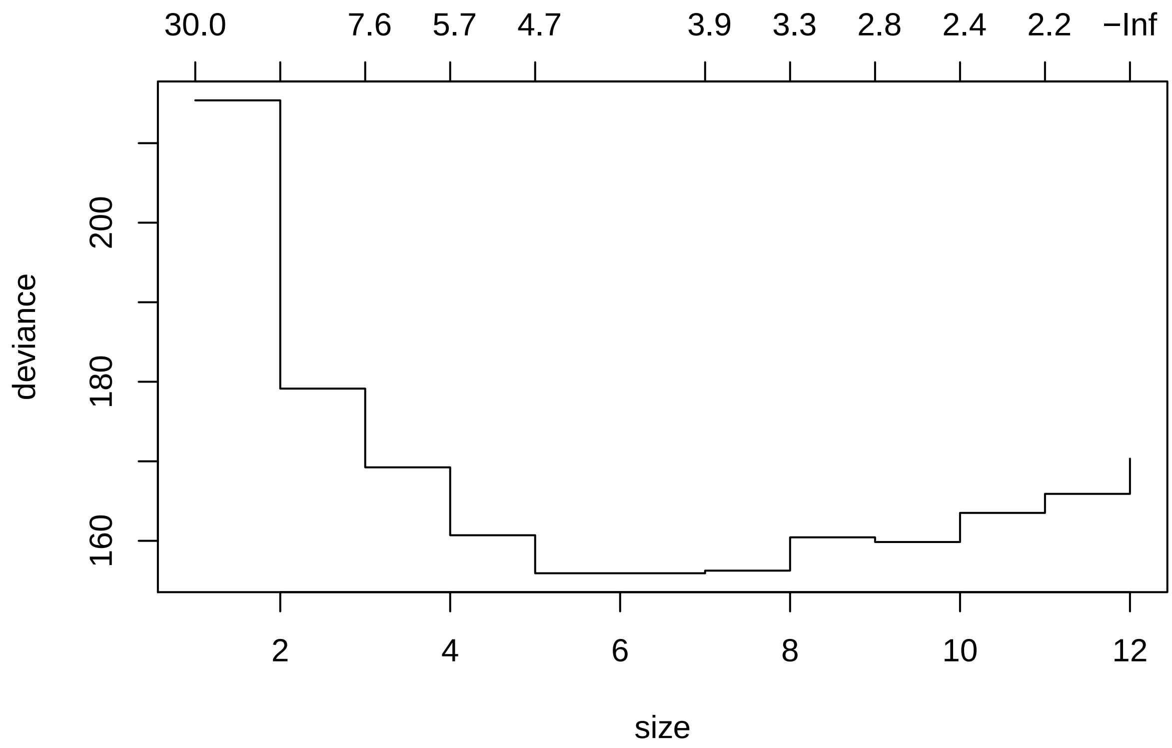

```r
#**************** Tree ****************#
library(tree)
```

```
## Warning: package 'tree' was built under R version 3.5.3
```

```r
set.seed(431)
depress2_tree <- tree(depress2~., data=job_train)
plot(depress2_tree)
text(depress2_tree, pretty=0)
```




```r
cv_depress2 <- cv.tree(depress2_tree, K=120)
plot(cv_depress2)
```

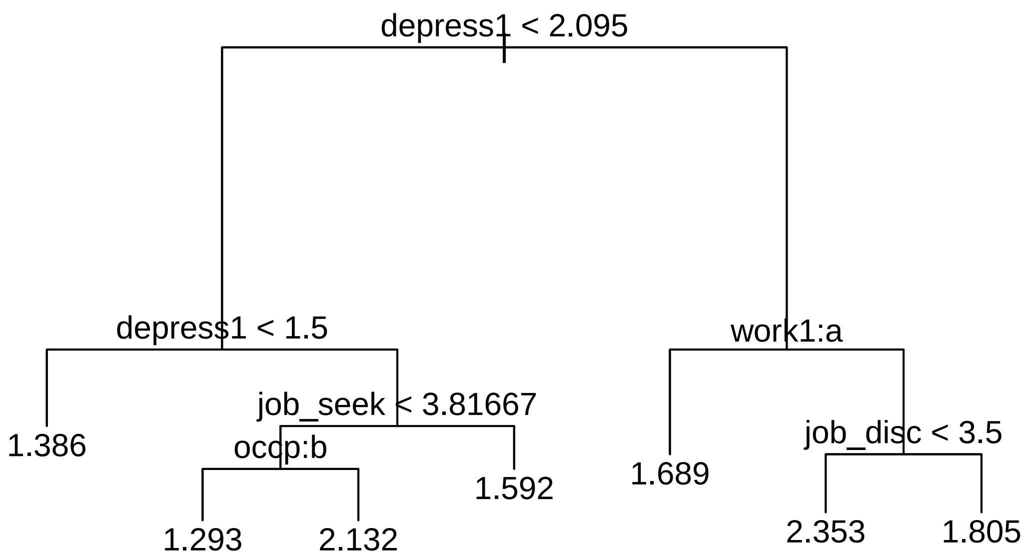



```
prediction <- predict(depress2_tree, job_test)
MSE = mean((prediction-job_test$depress2)^2)
print(MSE)
```

```
## [1] 0.4529052
```

Trying K=10, 20, 50, 100, 120, LOOCV, the result suggests 5,7,7,8,7,9 numbers of nodes provides the best long run MSE for our tree. Therefore, picking the highest occurance of suggestion: 7, the tree is pruned accrodingly.

```
pruned_depress2 <- prune.tree(depress2_tree, best=7)
plot(pruned_depress2)
text(pruned_depress2)
```

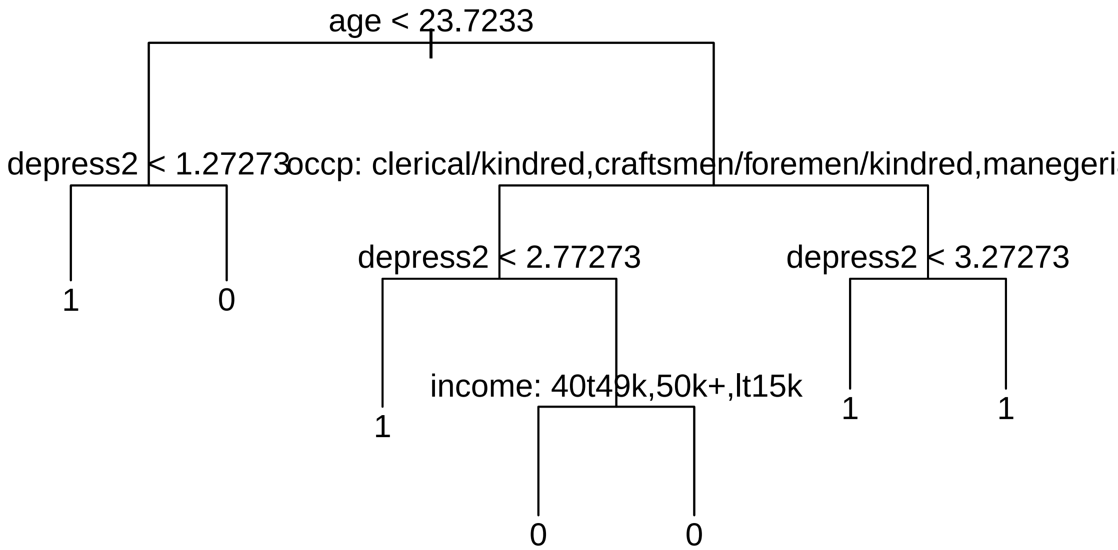



```
prediction <- predict(pruned_depress2, job_test)
MSE = mean((prediction-job_test$depress2)^2)
print(MSE)
```

```
## [1] 0.3744439
```

The MSE calculated from our prediction of the testing set using the pruned tree is indeed lower than than our original tree. $0.4037717 < 0.4054654$

```
set.seed(12348048)
train = job_train[-1]
tr <- tree(comply~., data=train)
plot(tr)
text(tr, pretty=0)
```


age < 23.7233
depress2 < 1.27273
occp: clerical/kindred,craftsmen/foremen/kindred,manegeria
1
0
depress2 < 2.77273
depress2 < 3.27273
income: 40t49k,50k+,lt15k
1
1
1
0
0