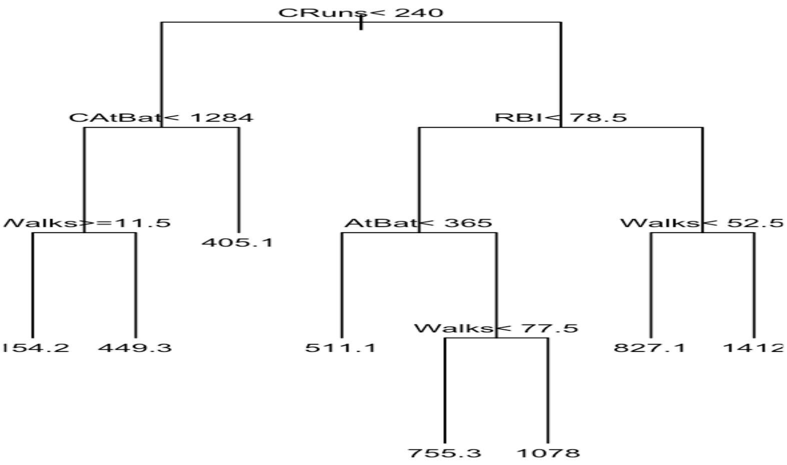
Assignment Seven

```
install.packages("rpart")
install.packages("MASS")
library(rpart)
library(MASS)

cols<- c("AtBat",
        "Hits",
       "HmRun",
        "Runs",
        "RBI",
        "Walks",
        "Years",
        "CAtBat",
        "CHits",
        "CHmRun",
        "CRuns",
        "CRBI",
        "CWalks",
        "PutOuts" ,
        "Assists",
        "Errors",
        "Salary")

hitters_data[cols] <-lapply(hitters_data[cols], as.factor)
df1_data<-hitters_data
df_data<-na.omit(df1_data)
#3 Build a single regression tree to predict Salary using all of the features except the player
#name of course

#10-fold CV...
cv_values <- rep(0, 10)

for(i in 1:length(cv_values)){
  print(i)
  inds <- sample(1:nrow(df_data), 0.80*nrow(df_data))
  tr_df <- df_data[inds,]
  te_df <- df_data[-inds,]

  tree1 <- rpart(Salary~., data = tr_df)

  preds <- predict(tree1, newdata = te_df)
  mse <- sqrt(mean( te_df$Salary - preds)^2)

  cv_values[i] <- mse
}

cv_values

#CV-error:
mean(cv_values)

#the pruning results are stored in
tree1$cptable




pfit <- prune(tree1, cp = 0.01, "CP")
plot(pfit, uniform=TRUE)
text(pfit, cex = 0.7)
```



```
#4 Random Forest
```

```
install.packages("RTools")
install.packages("randomForest")
library(randomForest)


sqrt(ncol(tr_df))
#take m = 4

cv_values1 <- rep(0, 10)
set.seed(1)
for(i in 1:length(cv_values1)){
  inds <- sample(1:nrow(df_data), 0.80*nrow(df_data))
  tr_df1 <- df_data[inds,]
  te_df1 <- df_data[-inds,]
  rf1 <- randomForest(Salary~., data = tr_df1, mtry = 4, importance = TRUE)
#importance = TRUE: tells the function to keep track of variable importance
#as the random forest is being built
  preds <- predict(rf1, newdata = te_df1)
  mse <- sqrt(mean( (preds - te_df1$Salary)^2))
  cv_values1[i] <-mse
}
cv_values1

#CV-error:
mean(cv_values1)
#we'll also generate a variable importance plot...
varImpPlot(rf1)
```

#5. Gradient boosted tree

```
install.packages("gbm")
library(gbm)
numTrees = seq(100,5000,100)
numDepth = 1:10
numShrinkage = seq(0.01,0.2,0.005)
hp<-expand.grid(numTrees, numDepth, numShrinkage)
hp
cv_values2 <- rep(0, 10)
mseo =100
for(i in 1:length(cv_values2)){
  inds <- sample(1:nrow(df_data), 0.80*nrow(df_data))
  tr_df2 <- df_data[inds,]
  te_df2 <- df_data[-inds,]
```

```
 set.seed(1)
 for(j in 1:nrow(hp)) {      # for-loop over rows

   boost1 <- gbm(Salary~., data = tr_df2, distribution = "gaussian", n.trees = hp[j,1],
       interaction.depth = hp[j,2], verbose = TRUE, shrinkage = hp[j,3])
   preds <- predict(boost1, newdata = te_df2, n.trees = hp[j,1])
   mse <- sqrt(mean( (preds - te_df2$Salary)^2))
   if (mse < mseo)
     mseo = mse
     ontree=hp[j,1]
     oshrinkage=hp[j,3]
     ondepth=hp[j,2]
 }
 cv_values2[i] <-mseo

}

cv_values2

#CV-error:
mean(cv_values2)
```