Anusha Muppalla
16286311

Lab Assignment7

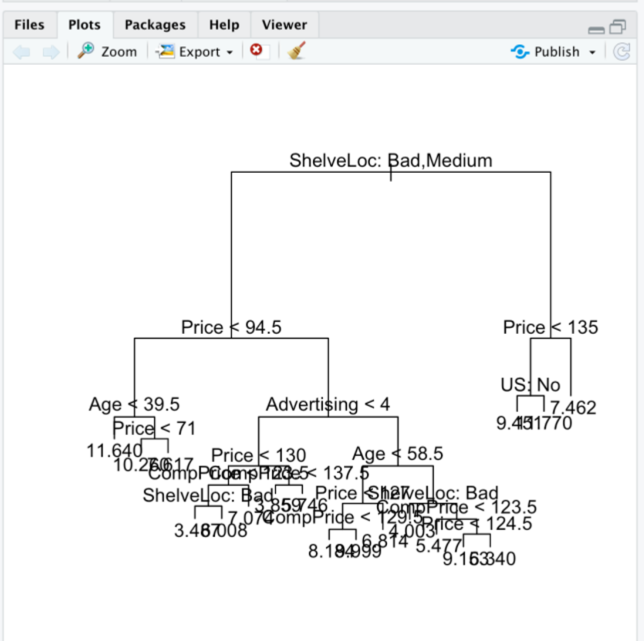
1.I n the lab, a classification tree was applied to the Carseats data set after converting Sales into a qualitative response variable. Now we will seek to predict Sales using regression trees and related approaches, treating the response as a quantitative variable.

(a) (5 points) Split the data set into a training set and a test set.

```
> library(ISLR)
> set.seed(1)
> train = sample(1:nrow(Carseats), nrow(Carseats) / 2)
> Car.train = Carseats[train, ]
> Car.test = Carseats[-train,]
>
```

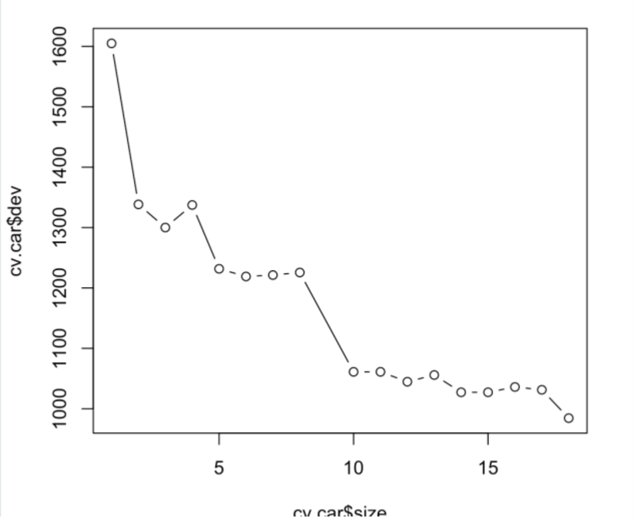
(b) (5 points) Fit a regression tree to the training set. Plot the tree, and interpret the results. What test MSE do you obtain?

```
> library(ISLR)
> set.seed(1)
> train = sample(1:nrow(Carseats), nrow(Carseats) / 2)
> Car.train = Carseats[train, ]
> Car.test = Carseats[-train,]
> library(tree)
> reg.tree = tree(Sales~.,data = Carseats, subset=train)
> reg.tree = tree(Sales~.,data = Car.train)
> #Both above formulas outcome the same result.
> summary(reg.tree)

Regression tree:
tree(formula = Sales ~ ., data = Car.train)
Variables actually used in tree construction:
[1] "ShelveLoc"   "Price"       "Age"
[4] "Advertising" "CompPrice"   "US"
Number of terminal nodes:  18
Residual mean deviance:  2.167 = 394.3 / 182
Distribution of residuals:
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-3.88200 -0.88200 -0.08712  0.00000  0.89590  4.09900
> plot(reg.tree)
> text(reg.tree ,pretty =0)
> yhat = predict(reg.tree,newdata = Car.test)
> mean((yhat - Car.test$Sales)^2)
[1] 4.922039
>
```

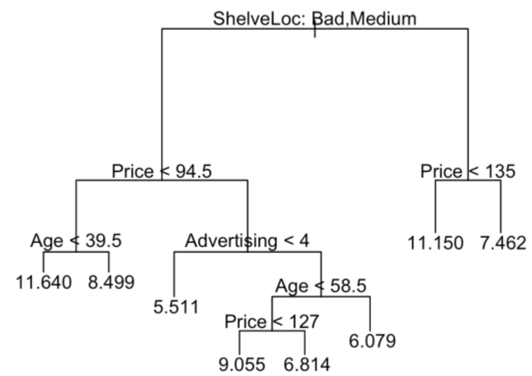


(c) (5 points) Use cross-validation in order to determine the optimal level of tree complexity. Does pruning the tree improve the test MSE?

```
> reg.tree = tree(Sales~.,data = Carseats, subset=train)
> reg.tree = tree(Sales~.,data = Car.train)
> #Both above formulas outcome the same result.
> summary(reg.tree)

Regression tree:
tree(formula = Sales ~ ., data = Car.train)
Variables actually used in tree construction:
[1] "ShelveLoc"   "Price"       "Age"
[4] "Advertising" "CompPrice"   "US"
Number of terminal nodes:  18
Residual mean deviance:  2.167 = 394.3 / 182
Distribution of residuals:
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-3.88200 -0.88200 -0.08712  0.00000  0.89590  4.09900
> plot(reg.tree)
> text(reg.tree ,pretty =0)
> yhat = predict(reg.tree,newdata = Car.test)
> mean((yhat - Car.test$Sales)^2)
[1] 4.922039
> set.seed(1)
> cv.car = cv.tree(reg.tree)
> plot(cv.car$size, cv.car$dev, type = "b")
>
```

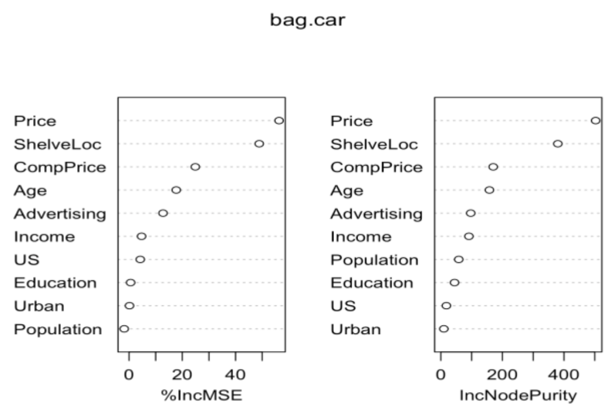



```
> summary(reg.tree)

Regression tree:
tree(formula = Sales ~ ., data = Car.train)
Variables actually used in tree construction:
[1] "ShelveLoc"   "Price"       "Age"
[4] "Advertising" "CompPrice"   "US"
Number of terminal nodes:  18
Residual mean deviance:  2.167 = 394.3 / 182
Distribution of residuals:
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-3.88200 -0.88200 -0.08712  0.00000  0.89590  4.09900
> plot(reg.tree)
> text(reg.tree ,pretty =0)
> yhat = predict(reg.tree,newdata = Car.test)
> mean((yhat - Car.test$Sales)^2)
[1] 4.922039
> set.seed(1)
> cv.car = cv.tree(reg.tree)
> plot(cv.car$size, cv.car$dev, type = "b")
> prune.car = prune.tree(reg.tree, best = 8)
> plot(prune.car)
> text(prune.car,pretty=0)
> yhat=predict(prune.car, newdata= Car.test)
> mean((yhat-Car.test$Sales)^2)
[1] 5.113254
>
```



(d) (5 points) Use the bagging approach in order to analyze this data. What test MSE do you obtain? Use the importance() function to determine which variables are most important.

```
> prune.car = prune.tree(reg.tree, best = 8)
> plot(prune.car)
> text(prune.car,pretty=0)
> yhat=predict(prune.car, newdata= Car.test)
> mean((yhat-Car.test$Sales)^2)
[1] 5.113254
> library(randomForest)
> set.seed(1)
> bag.car = randomForest(Sales~.,data=Car.train,mtry = 10, imp
ortance = TRUE)
> yhat.bag = predict(bag.car,newdata=Car.test)
> mean((yhat.bag-Car.test$Sales)^2)
[1] 2.605253
> importance(bag.car)
              %IncMSE IncNodePurity
CompPrice   24.8888481    170.182937
Income       4.7121131     91.264880
Advertising 12.7692401     97.164338
Population  -1.8074075     58.244596
Price       56.3326252    502.903407
ShelveLoc   48.8886689    380.032715
Age         17.7275460    157.846774
Education    0.5962186     44.598731
Urban        0.1728373      9.822082
US           4.2172102     18.073863
> varImpPlot(bag.car)
>
```



(e) (5 points) Use random forests to analyze this data. What test MSE do you obtain? Use the importance() function to determine which variables are most important. Describe the effect of m, the number of variables considered at each split, on the error rate obtained.

```
> varImpPlot(bag.car)
> library(randomForest)
> set.seed(1)
> rf.car = randomForest(Sales~.,data=Car.train,mtry = 3, impor
tance = TRUE)
> yhat.rf = predict(rf.car,newdata=Car.test)
> mean((yhat.rf-Car.test$Sales)^2)
[1] 2.960559
>
```

2.We now use boosting to predict Salary in the Hitters data set.

(a) (5 points) Remove the observations for whom the salary information is unknown, and then log-transform the salaries.

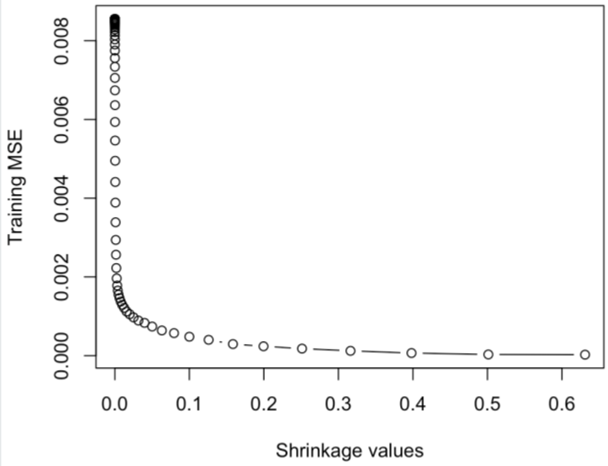
```
> Hitters = na.omit(Hitters)
> Hitters$Salary = log(Hitters$Salary)
```

(b) (5 points) Create a training set consisting of the first 200 observations, and a test set consisting of the remaining observations.

```
> Hitters = na.omit(Hitters)
> Hitters$Salary = log(Hitters$Salary)
> train = 1:200
> hitters.train = Hitters[train,]
> hitters.test = Hitters[-train,]
>
```

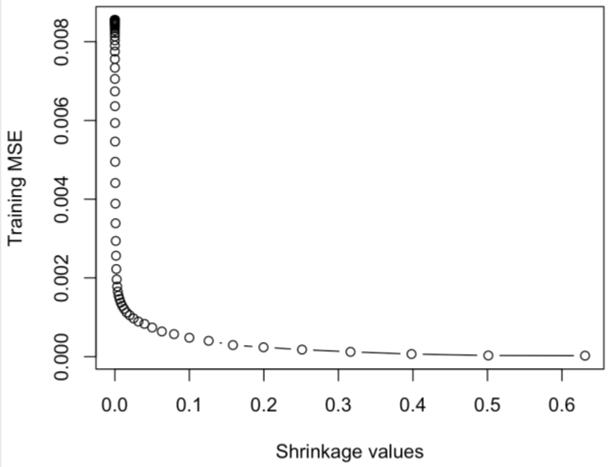
(c) (5 points) Perform boosting on the training set with 1,000 trees for a range of values of the shrinkage parameter λ. Produce a plot with different shrinkage values on the x-axis and the corresponding training set MSE on the y-axis.

```
> set.seed(1)
> pows = seq(-10, -0.2, by = 0.1)
> lambdas = 10^pows
> train.err = rep(NA, length(lambdas))
> for (i in 1:length(lambdas)) {
+     boost.hitters = gbm(Salary ~ ., data = hitters.train, di
stribution = "gaussian", n.trees = 1000, shrinkage = lambdas
[i])
+     pred.train = predict(boost.hitters, hitters.train, n.tre
es = 1000)
+     train.err[i] = mean((pred.train - hitters.train$Salary)^
2)
+ }
> plot(lambdas, train.err, type = "b", xlab = "Shrinkage value
s", ylab = "Training MSE")
>
```



(d) (5 points) Produce a plot with different shrinkage values on the x-axis and the corresponding test set MSE on the y-axis.

```
> lambdas = 10^pows
> train.err = rep(NA, length(lambdas))
> for (i in 1:length(lambdas)) {
+     boost.hitters = gbm(Salary ~ ., data = hitters.train, di
stribution = "gaussian", n.trees = 1000, shrinkage = lambdas
[i])
+     pred.train = predict(boost.hitters, hitters.train, n.tre
es = 1000)
+     train.err[i] = mean((pred.train - hitters.train$Salary)^
2)
+ }
> plot(lambdas, train.err, type = "b", xlab = "Shrinkage value
s", ylab = "Training MSE")
> set.seed(1)
> test.err <- rep(NA, length(lambdas))
> for (i in 1:length(lambdas)) {
+     boost.hitters = gbm(Salary ~ ., data = hitters.train, di
stribution = "gaussian", n.trees = 1000, shrinkage = lambdas
[i])
+     yhat = predict(boost.hitters, hitters.test, n.trees = 10
00)
+     test.err[i] = mean((yhat - hitters.test$Salary)^2)
+ }
```



(e) (5 points) Compare the test MSE of boosting to the test MSE that results from applying two of the regression approaches seen in Chapters 3 and 6.

```
> library(glmnet)
> fit1 = lm(Salary ~ ., data = hitters.train)
> pred1 = predict(fit1, hitters.test)
> mean((pred1 - hitters.test$Salary)^2)
[1] 0.005039684
> x = model.matrix(Salary ~ ., data = hitters.train)
> x.test = model.matrix(Salary ~ ., data = hitters.test)
> y = hitters.train$Salary
> fit2 = glmnet(x, y, alpha = 0)
> pred2 = predict(fit2, s = 0.01, newx = x.test)
> mean((pred2 - hitters.test$Salary)^2)
[1] 0.004676818
>
```

(f) (5 points) Which variables appear to be the most important predictors in the boosted model?

```
> boost.hitters <- gbm(Salary ~ ., data = hitters.train, distribution = "gaussian", n.trees = 1000,
 shrinkage = lambdas[which.min(test.err)])
> summary(boost.hitters)
                var    rel.inf
CAtBat       CAtBat 16.4749382
CRBI           CRBI 14.0081715
CRuns         CRuns 11.8074479
CHits         CHits 10.6095082
CWalks       CWalks  7.0162529
Walks         Walks  6.1058191
Years         Years  5.6390550
PutOuts     PutOuts  4.8779098
CHmRun       CHmRun  3.7543066
Hits           Hits  3.5288761
AtBat         AtBat  3.2808579
HmRun         HmRun  2.7361655
RBI             RBI  2.6878450
Assists     Assists  2.4552194
Runs           Runs  2.0349548
Errors       Errors  1.8469205
NewLeague NewLeague  0.5164605
Division   Division  0.4075827
League       League  0.2117085
>
```

(g) (5 points) Now apply bagging to the training set. What is the test set MSE for this approach?

```
> set.seed(1)
> bag.hitters <- randomForest(Salary ~ ., data = hitters.train, mtry = 19, ntree = 500)
> yhat.bag <- predict(bag.hitters, newdata = hitters.test)
> mean((yhat.bag - hitters.test$Salary)^2)
[1] 0.002450903
>
```