NIDHI MAHESHWARI – CSC 8491 DATA MINING – HOMEWORK ASSIGNMENT 4

1. **Regression Prediction**

SOLUTION:

Commands Used:

```
install.packages("ISLR")
library(ISLR)
attach(Carseats)
lm.fit = lm(Sales ~ CompPrice + Income + Advertising + Price + Age, data=Carseats )
summary(lm.fit)
predict(lm.fit,
data.frame(CompPrice=c(120,95,135),Income=c(45,67,85),Advertising=c(2.1,5.3,5.8),Price=c(110,100
,127),Age=c(37,56,61)),  interval="confidence")
predict(lm.fit,
data.frame(CompPrice=c(120,95,135),Income=c(45,67,85),Advertising=c(2.1,5.3,5.8),Price=c(110,100
,127),Age=c(37,56,61)),  interval="prediction")
```

<u>F**ill in the missing sales** predictions in the following table</u>

CompPrice	Income	Advertising	Price	Age	Sales Predicted
120	45	2.1	110	37	7.397435
95	67	5.3	100	56	5.826792
135	85	5.8	127	61	7.160397

2 a. Create a factor variable called *highmpg* within the Auto data set. The variable should have the value TRUE for autos with a *mpg* of greater than 30 and a value of FALSE otherwise. **Show your code**.

Solution:

<u>**Code used**</u>

```
install.packages(“ISLR”)
library(ISLR)
attach(Auto)
?Auto
highmpg=ifelse(mpg>30, "TRUE", "FALSE")
Auto=data.frame(Auto, highmpg)
```

2 b. Build a decision tree model that can be used to predict whether an auto has a high mpg based on data other than the numeric *mpg*. The *name* variable can be safely excluded from the model as it has too many levels and won't provide any predictive value, being unique to each row. **Show your code**, as well as a **plot** of the decision tree (with text), and also supply the **% of values that were misclassified** by the model.

Solution:

<u>Commands used :</u>

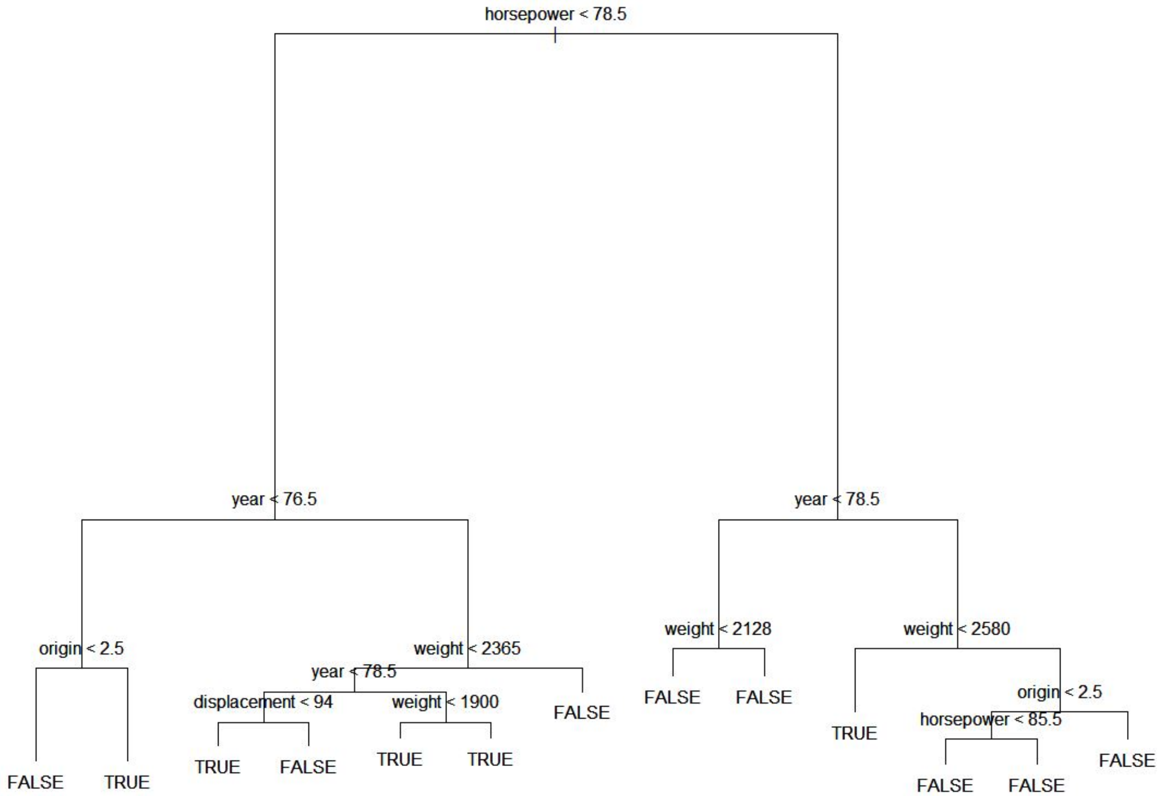
```
install.packages("tree")
library(tree)
tree.Auto=tree(highmpg~.-name-mpg,Auto)
summary(tree.Auto)
plot(tree.Auto)
text(tree.Auto,pretty=0)
```

<u>Output on console:</u>

```
Classification tree:
tree(formula = highmpg ~ cylinders + displacement + horsepower +
    weight + acceleration + year + origin, data = Auto)
Variables actually used in tree construction:
[1] "horsepower"   "year"        "origin"       "weight"       "displacement"
Number of terminal nodes:  13
Residual mean deviance:  0.2311 = 87.59 / 379
```

Misclassification error rate: 0.05867 = 23 / 392

Plot Obtained: Below


horsepower < 78.5
year < 76.5
year < 78.5
origin < 2.5
weight < 2365
year < 78.5
displacement < 94
weight < 1900
weight < 2128
weight < 2580
origin < 2.5
horsepower < 85.5
FALSE
TRUE
TRUE
FALSE
TRUE
TRUE
FALSE
FALSE
FALSE
TRUE
FALSE
FALSE
FALSE

NIDHI MAHESHWARI – CSC 8491 DATA MINING – HOMEWORK ASSIGNMENT 4

Problem C: Based on the model you built, what 3-4 variables seem to be most important in determining whether an auto has a high mpg? **List the variables.**

Solution: Horsepower, Year, Origin, Weight

Problem D: Set a seed using your 8-digitVillanova number (available in Novasis, among other places). Break the Auto data into training and test data sets using random sampling. Let the training data set be 67% of the records and the test data set be the remaining 33%. Train a new model on the training data, and then test the predictions of the new model against your test data. **Show your code** (including your Villanova number), as well as a table that **compares the predicted test values with the actuals**. State what **% of the test data was misclassified**.

Solution:
Commands Used:

```
set.seed(01642259)
train=sample(1:nrow(Auto), nrow(Auto) * .67)
Auto.test = Auto[-train,]
highmpg.test=highmpg[-train]
tree.autohighmpg = tree(highmpg~.-mpg-name,Auto, subset = train)
tree.pred=predict(tree.autohighmpg, Auto.test, type="class")
table(tree.pred, highmpg.test)
```

Output on console:

```
          highmpg.test
tree.pred FALSE TRUE
   FALSE   99    6
   TRUE     7   18
```

Percent of data misclassified= 6+7/130 =0.1=0.1*100=10%

Problem E: Determine whether your tree should be pruned. Use cross-validation to determine the ideal size of your tree. **Show your code**, along with the **output** that tells you how large your tree should be. Add a few sentences that **explain how large your tree should ideally be and how you know** that this is the best size.

Solution:
Commands Used:

```
cv.autohighmpg = cv.tree(tree.autohighmpg, FUN=prune.misclass)
names(cv.autohighmpg)
cv.autohighmpg
plot(cv.autohighmpg$size,cv.autohighmpg$dev,type="b)
plot(cv.autohighmpg$k,cv.autohighmpg$dev,type="b")
plot(cv.autohighmpg)
```

NIDHI MAHESHWARI – CSC 8491 DATA MINING – HOMEWORK ASSIGNMENT 4

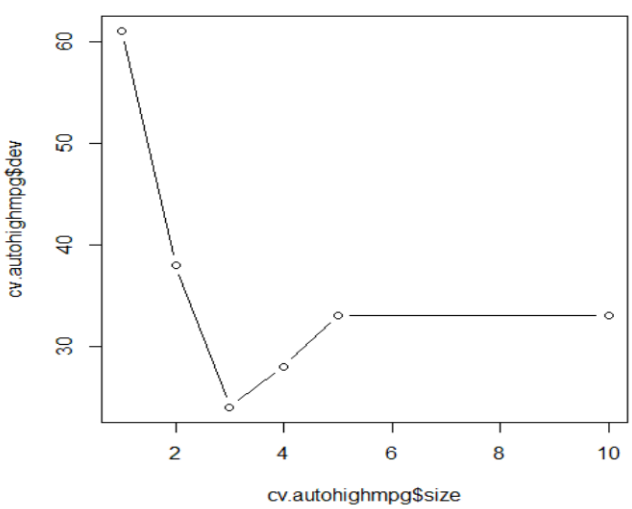
Output on Console:
[1] "size"   "dev"   "k"   "method"
$size
[1] 10 5 4 3 2 1
$dev
[1] 33 33 28 24 38 61

$k
[1] -Inf   0   1   5   13   21

$method
[1] "misclass"

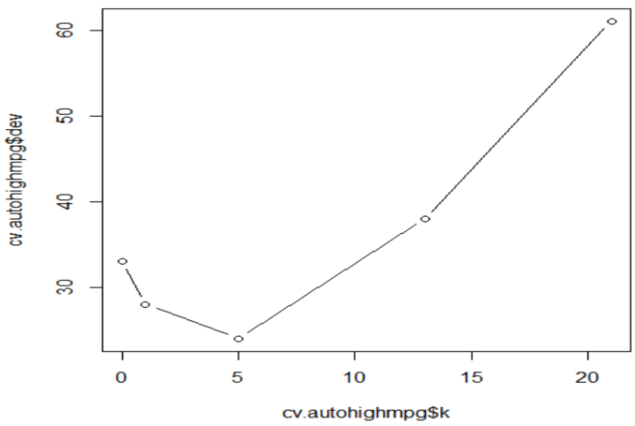
attr(,"class")
[1] "prune"   "tree.sequence"

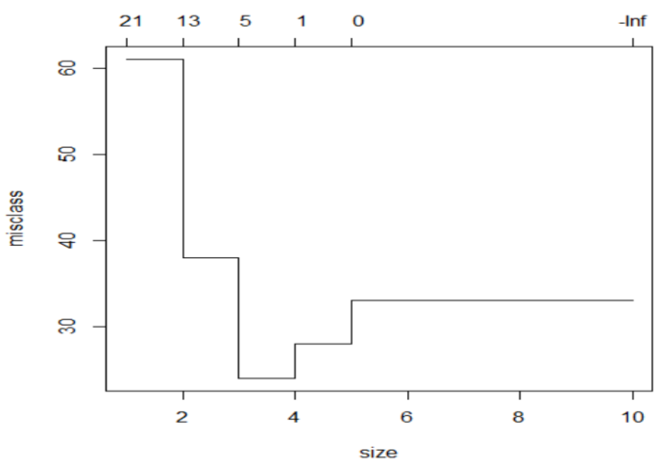
Plots Obtained



Mininmum deviance is 24. So best size is the size corresponding to this deviance which is 3.

NIDHI MAHESHWARI – CSC 8491 DATA MINING – HOMEWORK ASSIGNMENT 4





In the above plot, the lower X axis is the number of terminal nodes and the upper X axis is the number of folds (# of pieces the data is split) in the cross validation.
It shows how the misclassification error varies against these. So, this plot is very useful in determining the optimal number of terminal nodes at which the decision tree should be pruned.

In the above plot, 3 and 5 the two options (# terminal nodes) at which you want to prune the data.

Ideally, it is best keep the tree as simple as possible (lesser number of nodes) and the misclassification error as low as possible.

Given a choice of number of terminal nodes between 3 - 5, 3 terminal nodes should be the first choice.

NIDHI MAHESHWARI – CSC 8491 DATA MINING – HOMEWORK ASSIGNMENT 4

f) Create a pruned version of your tree based on the ideal size you determined in the previous step. Use this pruned version to make predictions for your test data set. **Show your code,** as well as a table that **compares the predicted test values with the actuals**. State what **% of the test data was misclassified**. Explain **whether or not the pruning improved** your predictive accuracy.

Solution:

Commands Used

```
prune.auto=prune.misclass(tree.autohighmpg,best=3)
prune.predict = predict(prune.auto, Auto.test, type="class")
table(prune.predict, highmpg.test)
highmpg.test
prune.predict FALSE  TRUE
FALSE           103   9
TRUE              2   16
```

% of the test data was misclassified= 9+2/130
=0.085
=8.5 %

Yes **% of the test data was misclassified** decreases from 10 % to 8.5% after Pruning. So pruning improved the predictive accuracy.

g) Try to improve your results using either bagging or a random forest. **State** which alternative you are using. **Show your code**, as well as your **OOB error rate**. Note again that *name* should not be part of your analysis.

Solution: **install.packages("randomForest")**

```
library(randomForest)
set.seed(01642259)
rf.Auto=randomForest(highmpg~. -mpg-name, data=Auto, subset = train, importance=TRUE)
rf.Auto

Call:
randomForest(formula = highmpg ~ . - mpg - name, data = Auto,      importance = TRUE, subset = train)
Type of random forest: classification
Number of trees: 500
No. of variables tried at each split: 2

OOB estimate of error rate: 8.02%
Confusion matrix:
        FALSE TRUE class.error
FALSE  196      8       0.03921569
TRUE    13      45      0.22413793
```

importance(rf.Auto)

TRUE	Mean	DecreaseAccuracy	Mean	DecreaseGini
cylinders	**5.521281**	**5.603101**	**7.79734**	**3.044625**
displacement	**9.137109**	**18.107183**	**18.36833**	**15.238174**
horsepower	**11.845740**	**18.484905**	**22.12942**	**19.699102**
weight	**11.464012**	**22.215984**	**23.43002**	**18.658696**
acceleration	**9.904066**	**7.468129**	**12.53212**	**8.686559**
year	**21.618763**	**36.802977**	**37.77017**	**18.371809**
origin	**9.404817**	**16.764425**	**17.14757**	**5.879648**

varImpPlot(rf.Auto)

2h)

Solution 3. P(H|E) = P(E|H) * P(H)/ P(E)
Where P(H|E)= Probability of Hypothesis given some evidence
P(E|H)= Probability of evidence given some Hypothesis
P(H)=Probability of hypothesis
P(E)=Probability of Evidence

probability that person laptop will fail within warrenty period. Given that person belongs to sales department	**=**	**Probability that a person belongs to sales department. Given that there laptop fail within warranty period**	*****	**probability that someone laptop will fail within warranty period**

/ probability of being from sales department