Reihaneh Moghisi
Apr 24 2023

Assignment 7

The `Carseats` Dataset (Regression Trees, Random Forests)

In 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) `train/test` Split

Q: *Split the data set into a training set and a test set.*

A:

```
set.seed(2, sample.kind = "Rounding")

train_index <- sample(1:nrow(Carseats), nrow(Carseats) / 2)

train <- Carseats[train_index, ] # 200
test <- Carseats[-train_index, ] # 200
```

(b) Regression Tree Plot

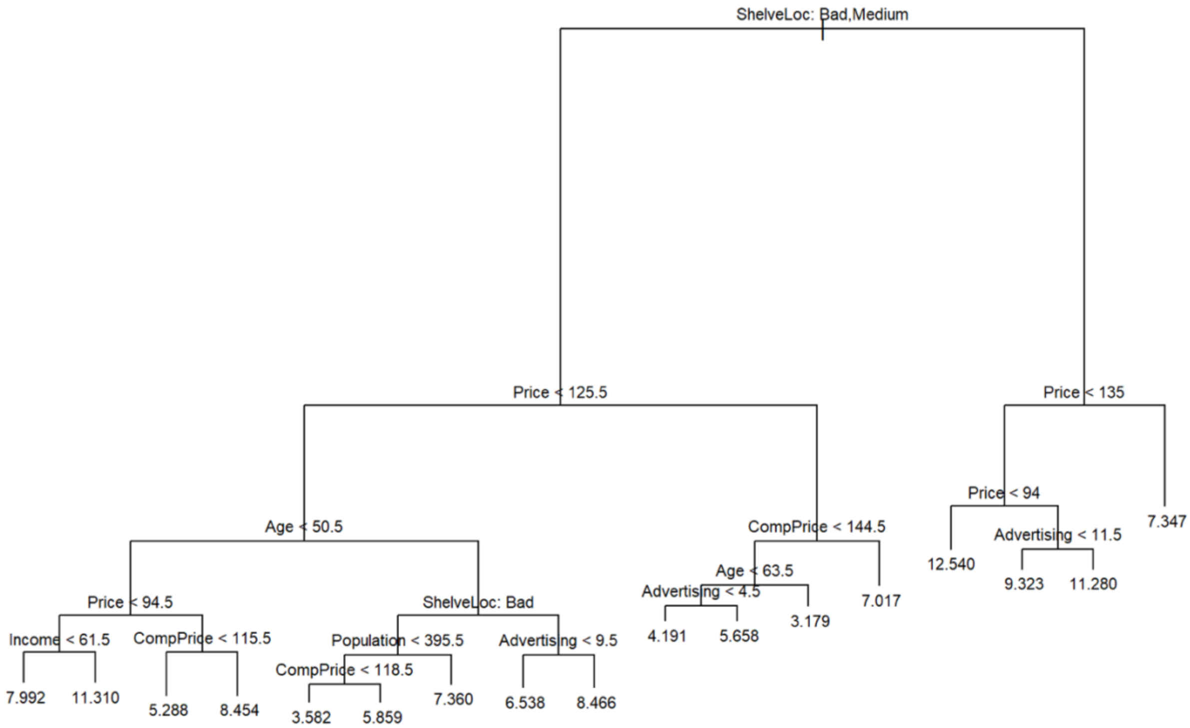
Q: *Fit a regression tree to the training set. Plot the tree, and interpret the results. What test MSE do you obtain?*

A:

The plot can be seen below:

```
tree_model <- tree(Sales ~ ., train)

plot(tree_model)
text(tree_model, pretty = 0, cex = 0.7)
```

Reihaneh Moghisi
Apr 24 2023



It can be deduced that ShelveLoc and Price are the two most crucial variables in forecasting car seat sales as they are listed at the top of the tree (since they offered the optimal data split). The tree comprises a sum of 17 end nodes:

```
summary(tree_model)
##
## Regression tree:
## tree(formula = Sales ~ ., data = train)
## Variables actually used in tree construction:
## [1] "ShelveLoc"   "Price"       "Age"         "Income"      "CompPrice"
## [6] "Population"  "Advertising"
## Number of terminal nodes:  17
## Residual mean deviance:  2.341 = 428.4 / 183
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -3.76700 -1.00900 -0.01558  0.00000  0.94900  3.58600
```

Predicting and evaluating the test MSE:

```
test_pred <- predict(tree_model, test)
mean((test_pred - test$Sales)^2)
## [1] 4.675961
```

To provide a comparison, here is the baseline test MSE obtained by utilizing the mean of train$Sales as the forecast for all fresh test observations:

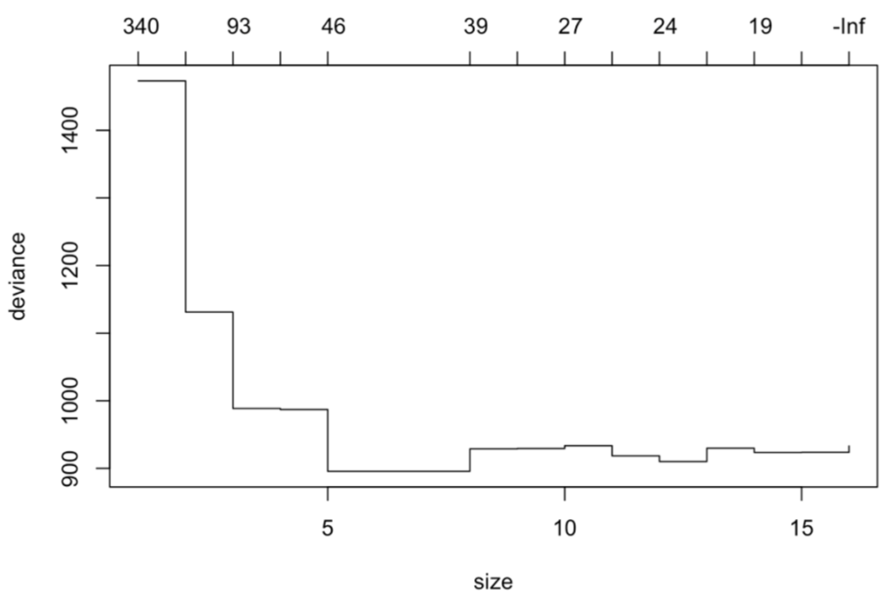
```
baseline_test_pred <- mean(train$Sales)
mean((baseline_test_pred - test$Sales)^2)
## [1] 8.745407
```

(c) Cross-Validation Pruning

Q: *Use cross-validation in order to determine the optimal level of tree complexity. Does pruning the tree improve the test MSE?*

A:

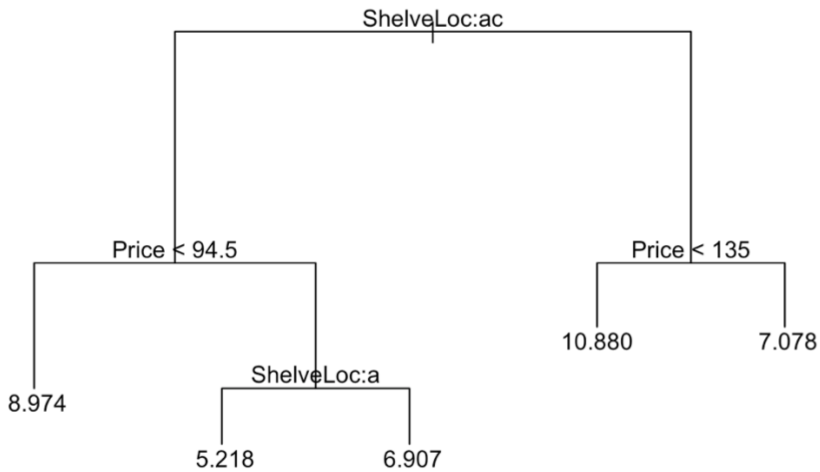
```
tree.carseats.cv=cv.tree(tree.carseats)

plot(tree.carseats.cv)
```



The selected model with the lowest cross-validation error appears to be the model with 5 terminal nodes.

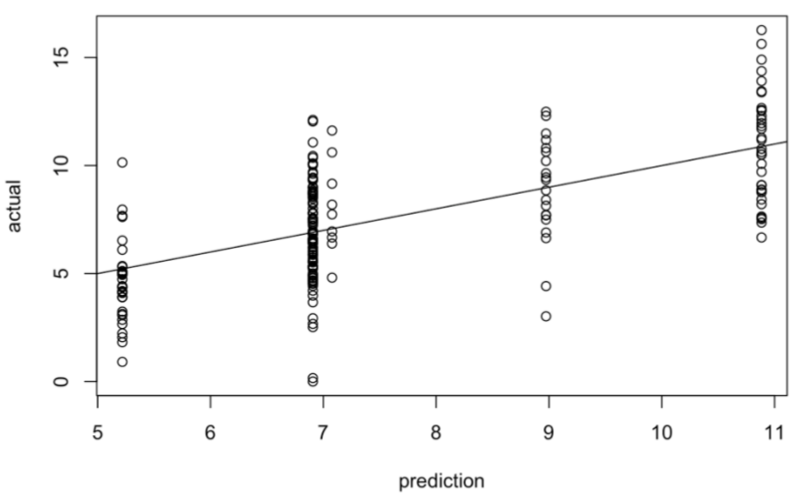
Reihaneh Moghisi
Apr 24 2023

It's clear that the test MSE in part b) is the same as the ideal tree, which is a tree that has fully grown without any pruning. However, if we want to choose a smaller sub-tree, we can use the following method. In this case, I decided to go with a tree that has 5 end nodes (best = 5), but the test MSE remains the same as in part b), as expected.




```
pruned_tree_model <- prune.tree(tree_model, best = 5)

test_pred <- predict(pruned_tree_model, test)
mean((test_pred - test$Sales)^2)
## [1] 4.675961
```

From the MSE you can observe that it does not improve the test MSE.



Reihaneh Moghisi
Apr 24 2023

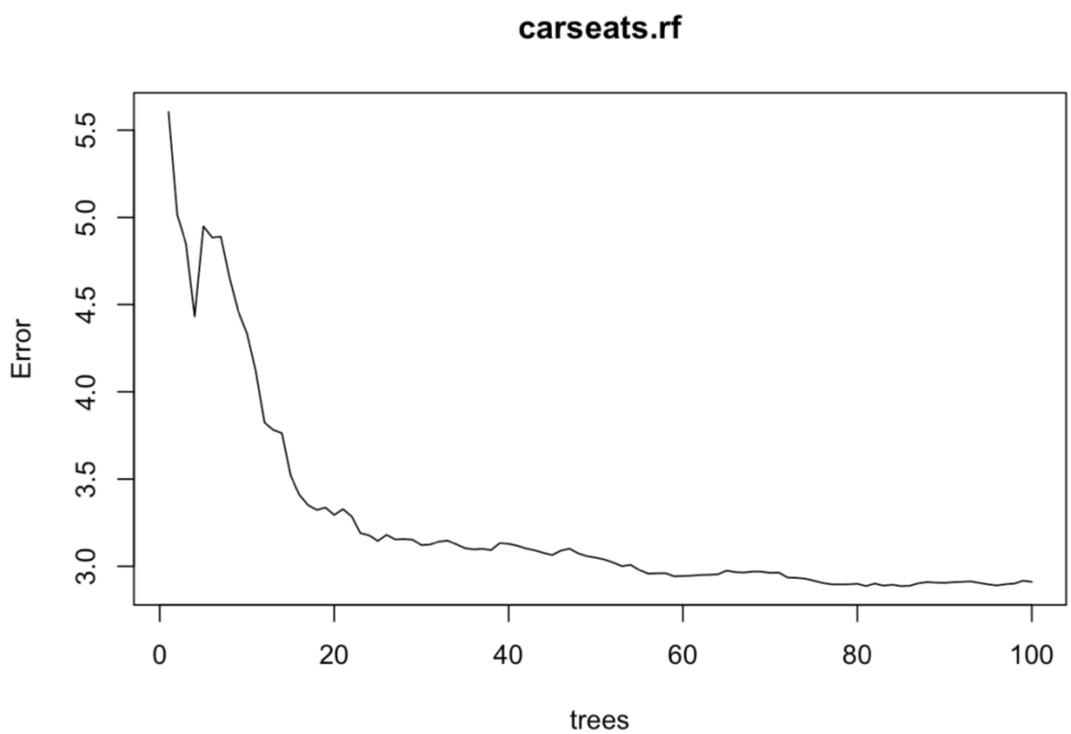
(d) Bagged Trees

Q: *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.*

A:
```
require(randomForest)

d=ncol(Carseats)-1

set.seed(42)
carseats.rf=randomForest(Sales~.,data=Carseats,subset=train,mtry=d,importance
=T,ntree=100)

tree.pred=predict(carseats.rf,Carseats[-train,])
mean((tree.pred-Carseats[-train,'Sales'])^2)
```

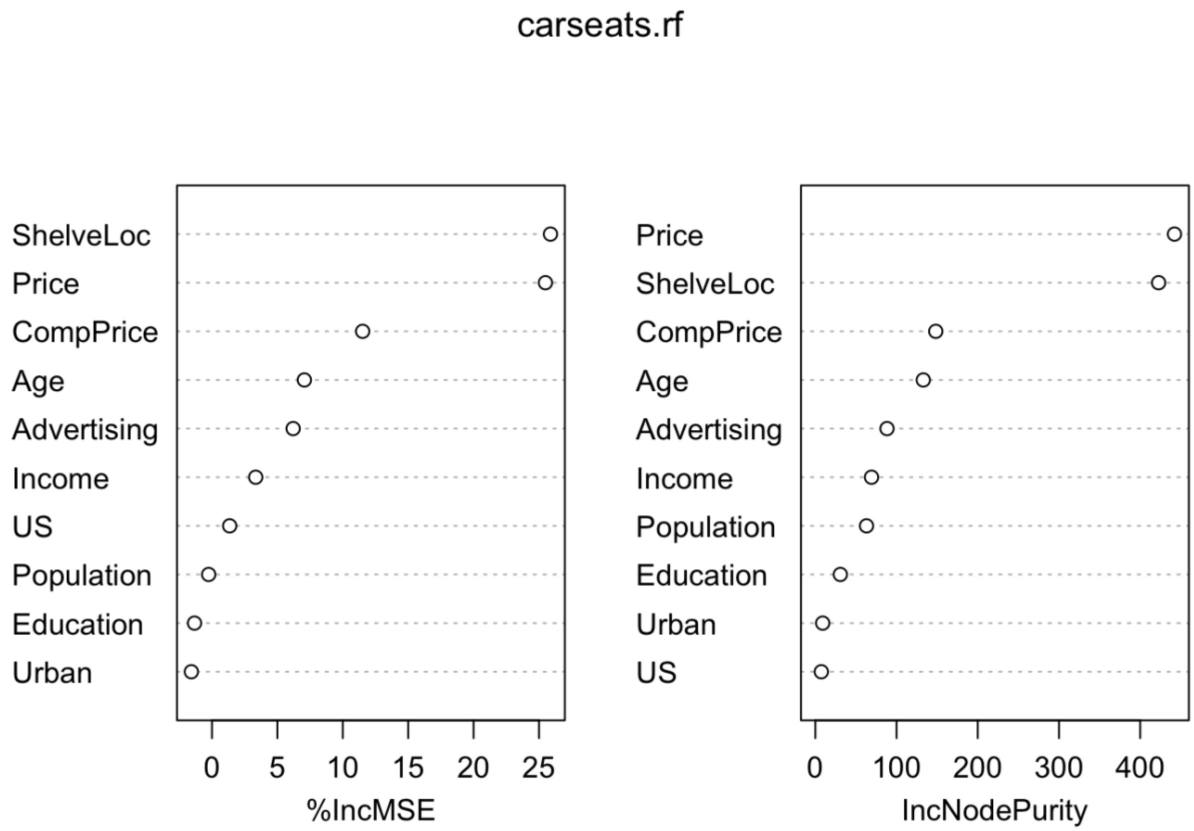
```
## [1] 2.573563
```

```
plot(carseats.rf)
```




```
varImpPlot(carseats.rf)
```

Reihaneh Moghisi
Apr 24 2023




```
kable(importance(carseats.rf))
```

	%IncMSE	IncNodePurity
CompPrice	11.5170922	148.297954
Income	3.3430145	69.192503
Advertising	6.2062812	88.251686
Population	-0.2522175	63.025904
Price	25.4921858	442.200618
ShelveLoc	25.8775065	422.649283

	%IncMSE	IncNodePurity
Age	7.0625363	132.928876
Education	-1.3343248	30.809743
Urban	-1.5756286	9.061747
US	1.3535718	7.374875

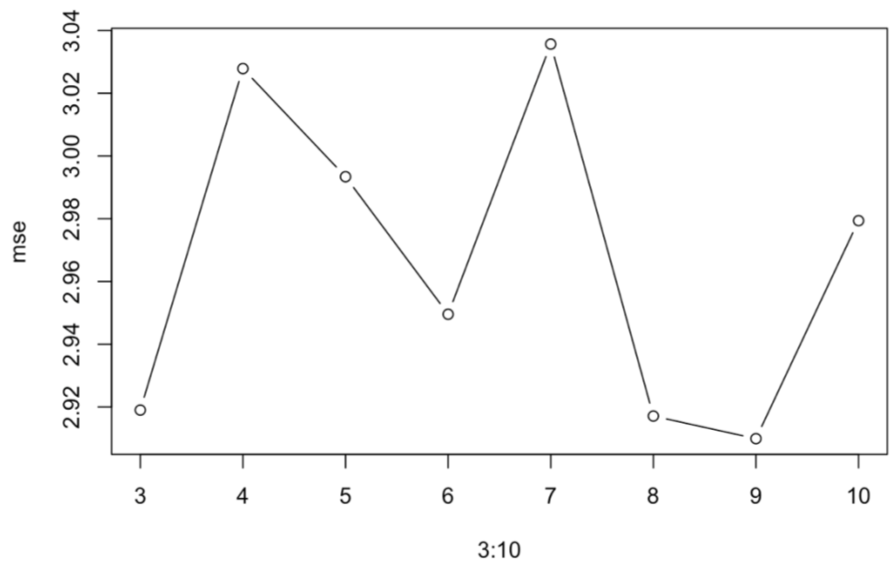
The predictor `Price` is clearly the most important predictor in predicting `Sales`. This model also achieves a much lower MSE than the previous one, with almost a half of reduction achieved in the test MSE.

(e) Random Forests

Q: *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.*
A:

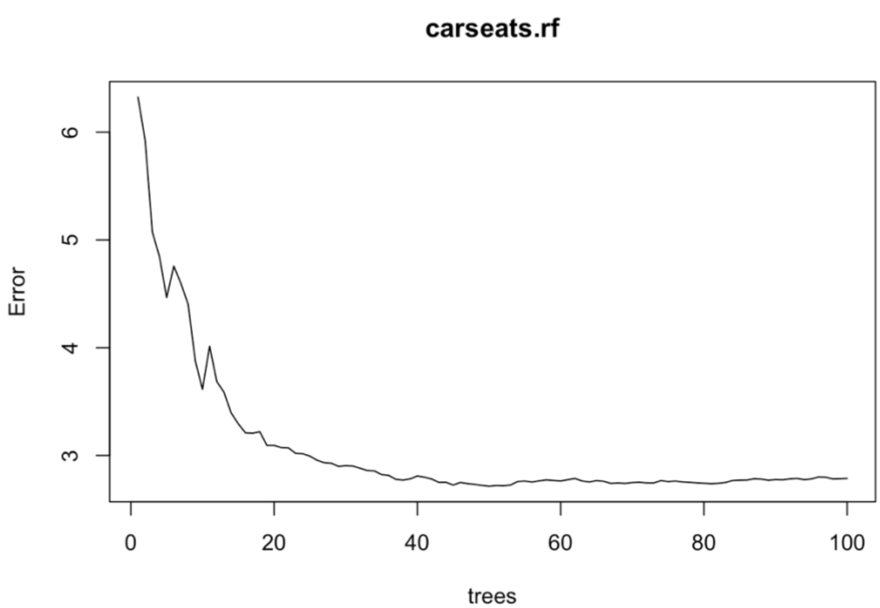
```
mse=c()
set.seed(42)


for(i in 3:10){
  carseats.rf=randomForest(Sales~.,data=Carseats,subset=train,mtry=5,importan
ce=T,ntree=100)
  tree.pred=predict(carseats.rf,Carseats[-train,])
  mse=rbind(mse,mean((tree.pred-Carseats[-train,'Sales'])^2))
}
plot(3:10,mse,type='b')
```

Reihaneh Moghisi
Apr 24 2023



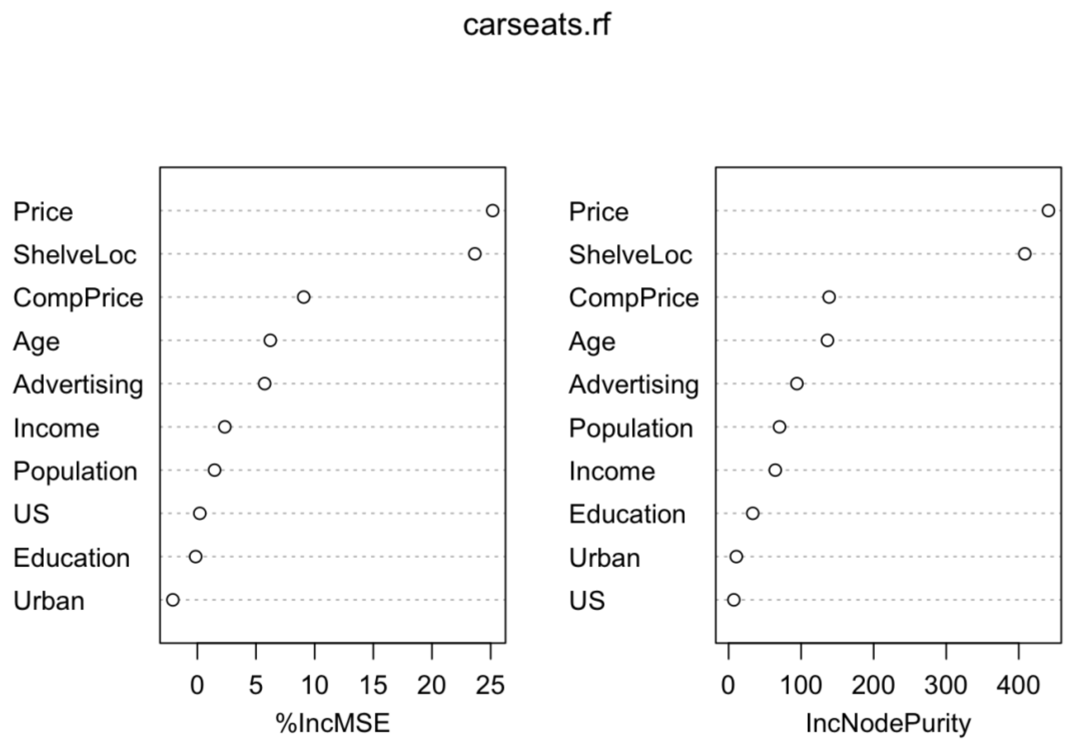
The plot above displays the effect of `mtry` in the test MSE.

```
require(randomForest)


set.seed(42)

carseats.rf=randomForest(Sales~.,data=Carseats,subset=train,mtry=9,importance
=T,ntree=100)

plot(carseats.rf)
```



Reihaneh Moghisi
Apr 24 2023

```
varImpPlot(carseats.rf)
```




```
kable(importance(carseats.rf))
```

	%IncMSE	IncNodePurity
CompPrice	9.0678902	138.620007
Income	2.3427697	64.333859
Advertising	5.7404558	94.148559
Population	1.4682674	70.206086
Price	25.1750997	440.957346

	%IncMSE	IncNodePurity
ShelveLoc	23.6551259	408.393880
Age	6.2210635	136.281793
Education	-0.1427487	33.300827
Urban	-2.0875662	10.667281
US	0.2117698	7.227559

From the data above you can see that `ShelveLoc` is now the most important predictor in terms of MSE (whose absence most increase the training MSE). Moreover, while considering only 9 predictors for training each tree achieves a lower training MSE the test MSE is higher than the bagging approach.