Tree based methods: bagging, random forest and boosting

For this exercise, the CarSeats dataset from the ISLR package is used. It is a simulated data set containing sales of child car seats at 400 different stores. The description of the variables in the dataset are as follows:

- Sales: Unit sales (in thousands) at each location
- CompPrice: Price charged by competitor at each location
- Income: Community income level (in thousands of dollars)
- Advertising: Local advertising budget for company at each location (in thousands of dollars)
- Population: Population size in region (in thousands)
- Price: Price company charges for car seats at each site
- ShelveLoc: A factor with levels Bad, Good and Medium indicating the quality of the shelving location for the car seats at each site
- Age: Average age of the local population
- Education: Education level at each location
- Urban: A factor with levels No and Yes to indicate whether the store is in an urban or rural location
- US: A factor with levels No and Yes to indicate whether the store is in the US or not

Tasks Predict sales using bagging, random forest and boosting. Evaluate and compare the performance of the models generated by these decision tree techniques.

```r
library (randomForest)
```

```
## Warning: package 'randomForest' was built under R version 3.6.3
```

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

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

```r
library(ISLR)
```

```
## Warning: package 'ISLR' was built under R version 3.6.3
```

```r
library(tree)
```

```
## Warning: package 'tree' was built under R version 3.6.3
```

```r
library(gbm)
```

```
## Warning: package 'gbm' was built under R version 3.6.3
```

```
## Loaded gbm 2.1.8
```

```r
attach(Carseats)
```

Generate test and training datasets

```r
set.seed(222)
ratio <- sample(1:nrow(Carseats),nrow(Carseats)*0.5)
```

```r
test <- Carseats[-ratio,]
train <- Carseats[ratio,]
```

Bagging There are 10 predictor variables in the dataset. Hence for bagging, we use mtry=10.

```r
set.seed(222)
bag_model <- randomForest(Sales~., data= train, mtry = 10, importance = TRUE, ntree=1000)
bag_model
```

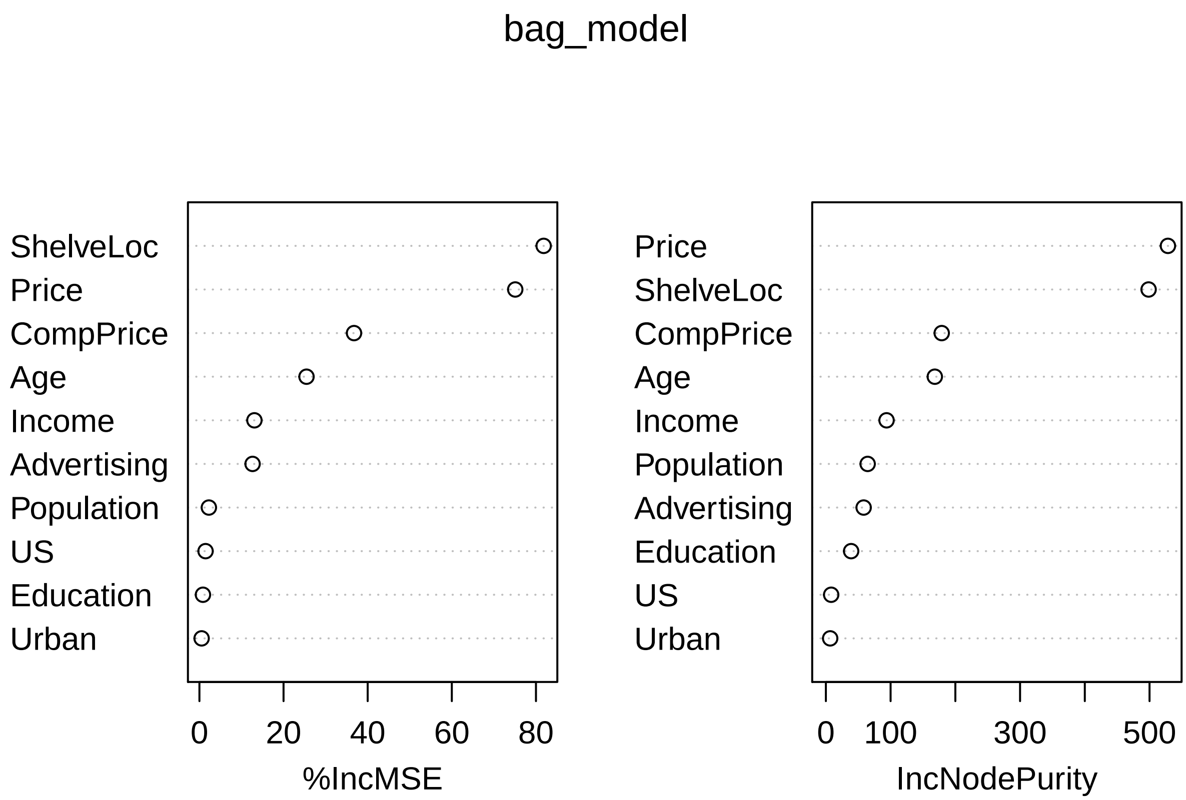
```
##
## Call:
##  randomForest(formula = Sales ~ ., data = train, mtry = 10, importance = TRUE,      ntree = 1000)
##                Type of random forest: regression
##                      Number of trees: 1000
## No. of variables tried at each split: 10
##
##           Mean of squared residuals: 2.746873
##                     % Var explained: 67.5
```

Interpretation of model generated by bagging

```r
importance(bag_model)
```

```
##               %IncMSE IncNodePurity
## CompPrice   36.7548348    178.955061
## Income      13.0720561     93.767106
## Advertising 12.6356763     58.343387
## Population   2.2491144     64.539213
## Price       75.0771808    528.406917
## ShelveLoc   81.8304059    498.573434
## Age         25.4589586    168.287918
## Education    0.8381906     39.019997
## Urban        0.5233544      6.646045
## US           1.4605611      8.192876
```

```r
varImpPlot(bag_model)
```



The model generated from bagging show that ShelveLoc and Price are the important variables that affect sales.

The %IncMSe values of both variables which indicates the mean decrease in the accuracy of the model if these variables are excluded from the model. It can be seen that the %IncMSE values for ShelveLoc and price are very high thus signifying their importance in the model.

Likewise, the IncNodePurity shows the total decrease in node impurity associated to the variables. The values for ShelveLoc and price are very high indicating that these variables contribute significantly to node purity in the decision trees.

Therefore, the model generated with bagging technique is as follows: $Sales \sim ShelveLoc + Price$

Predictions and assessment of bagging model performance

```
predictions_bag <- predict(bag_model,newdata = test)
MSE_bagging <- mean((predictions_bag - test$Sales)^2)
```

Random Forest Since the response variable is quantitative (we will be generating random forest of regression trees), we will use p/3 predictor variables. hence, mtry = 3

```
set.seed(222)
rf_model <- randomForest(Sales~., data= train, mtry = 4, importance = TRUE,ntree=1000)
rf_model
```

```
##
## Call:
##  randomForest(formula = Sales ~ ., data = train, mtry = 4, importance = TRUE,      ntree = 1000)
##                Type of random forest: regression
##                      Number of trees: 1000
```

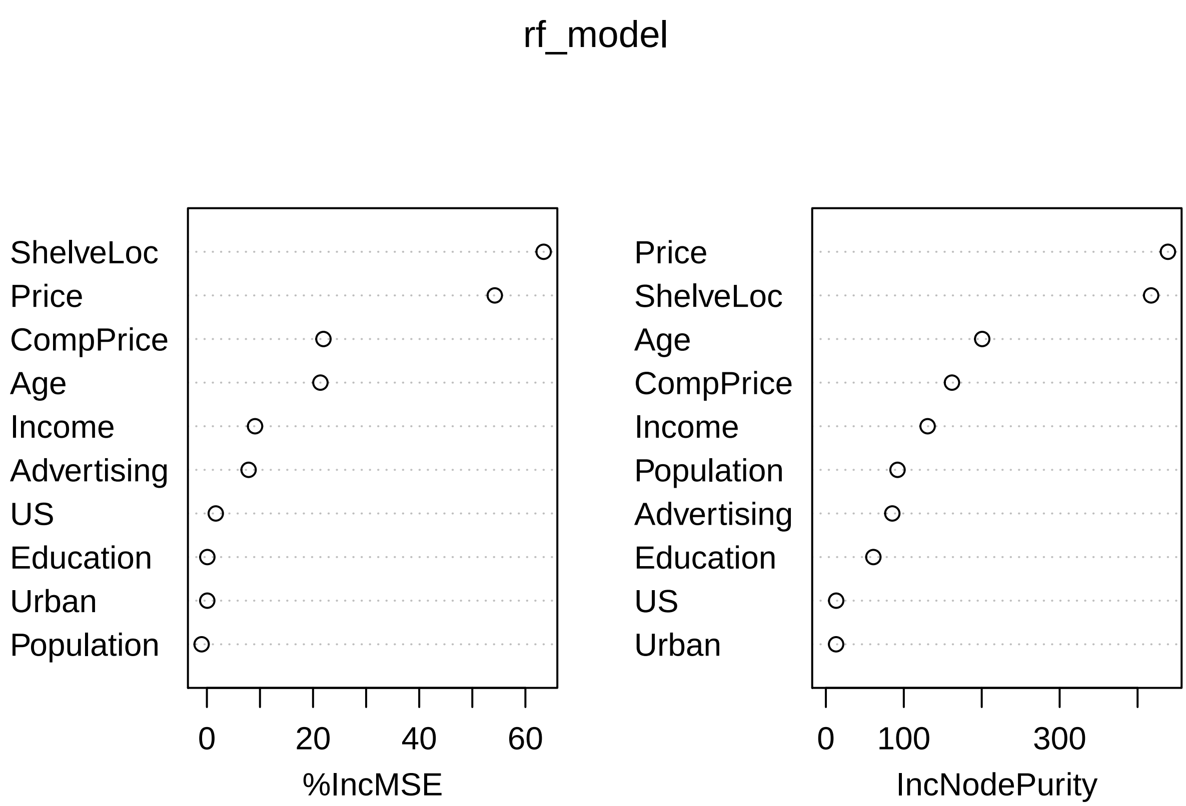
```
## No. of variables tried at each split: 4
##
##           Mean of squared residuals: 3.052206
##                     % Var explained: 63.89
```

Interpretation of model generated by Random forest

```r
importance(rf_model)
```

```
##                   %IncMSE IncNodePurity
## CompPrice   21.95576022     161.80562
## Income       9.07353174     130.57718
## Advertising  7.85329952      85.25187
## Population  -0.99416627      91.95462
## Price       54.23283831     438.92205
## ShelveLoc   63.44034053     417.39050
## Age         21.38360452     200.59966
## Education    0.08825133      60.94746
## Urban        0.07212251      13.10908
## US           1.67232230      13.22129
```

```r
varImpPlot(rf_model)
```



The model generated from random forest show that ShelveLoc and Price are the important variables that affect sales.

The %IncMSe values of both variables which indicates the mean decrease in the accuracy of the model if these variables are excluded from the model. It can be seen that the %IncMSE values for ShelveLoc and price are very high thus signifying their importance in the model.

Likewise, the IncNodePurity shows the total decrease in node impurity associated to the variables. The values for ShelveLoc and price are very high indicating that these variables contribute significantly to node purity in the decision trees.

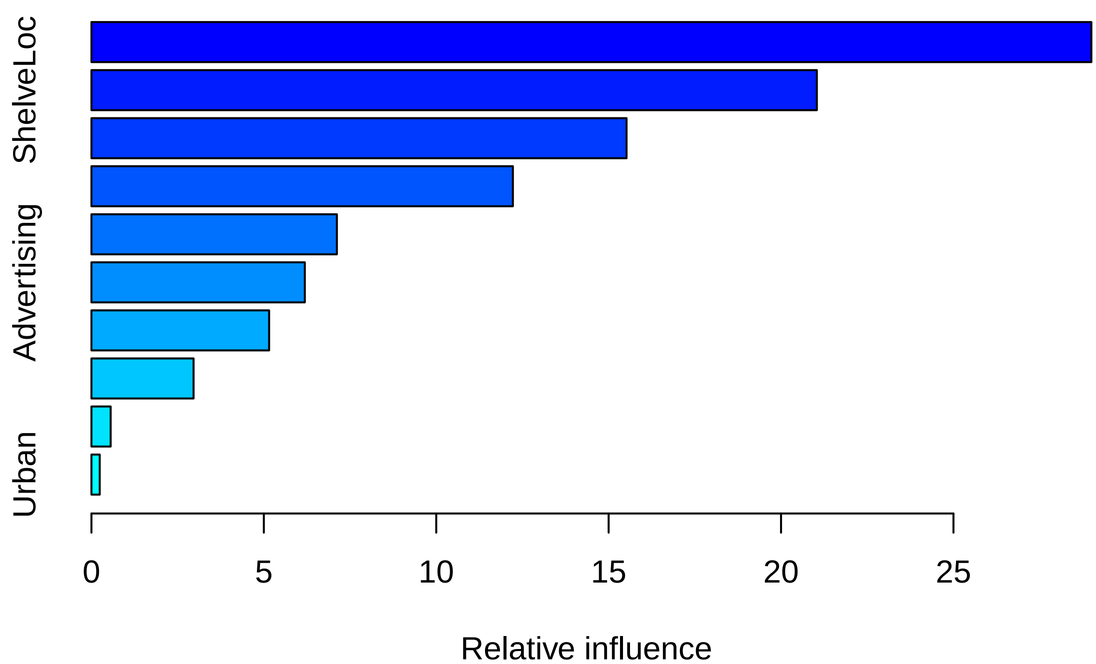
Therefore, the model generated with the random forest technique is as follows: *Sales* ~ *ShelveLoc* + *Price*

Predictions and assessment of random forest model performance

```
predictions_rf <- predict(rf_model,newdata = test)
MSE_rf <- mean((predictions_rf - test$Sales)^2)
```

Boosting In the boosting classifier, we use distribution ="gaussian" for regression trees or distribution = "bernoulli" for classification trees.

```
set.seed(222)
boost_model =gbm(Sales~.,data=train, distribution="gaussian",n.trees =1000 , interaction.depth =4)
summary(boost_model)
```




```
##                     var    rel.inf
## Price             Price 28.9995253
## ShelveLoc     ShelveLoc 21.0385095
## CompPrice     CompPrice 15.5195899
## Age                 Age 12.2223220
## Income           Income  7.1180899
## Advertising Advertising  6.1883562
## Population   Population  5.1533006
## Education     Education  2.9618317
## US                   US  0.5578466
## Urban             Urban  0.2406282
```

The boosting model indicate that the important variables affecting sales are price and ShelveLoc as seen in the influence plot. The model is *Sales ~ Price + ShelveLoc*

Predictions and assessment of boosting model performance

```
predictions_boosting <- predict(boost_model,newdata = test)
```

```
## Using 1000 trees...
```

```
MSE_boosting <- mean((predictions_boosting - test$Sales)^2)
```

Comparison of model performance

```
data.frame(MSE_bagging,MSE_rf, MSE_boosting)
```

```
##   MSE_bagging   MSE_rf MSE_boosting
## 1    2.154229 2.443522     1.720936
```

The error rate of the boosting model is the lowest as expected, followed by that of the bagging model while the random forest model had the worst performance. Unexpectedly, the bagging model performed better than the random forest model in this analysis.