Decision Trees and Random Forests

Problem Statement: *This project involves building decision tree and random forest models to answer a number of questions. We will use the Carseats dataset that is part of the ISLR package.*

Loading Required Libraries

```
library(ISLR)

library(dplyr)

library(glmnet)

library(caret)

library(rpart)

library(rpart.plot)

library(rattle)
```

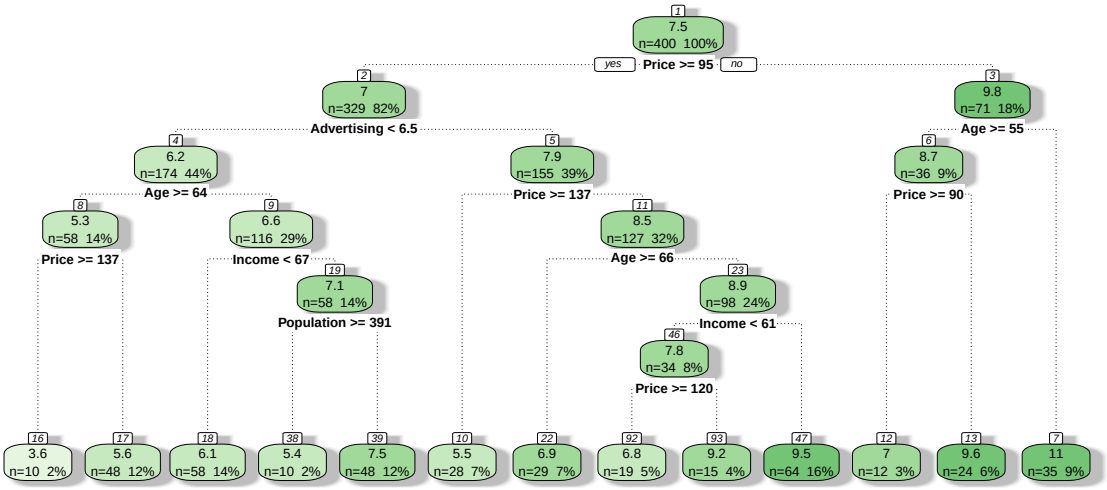
Selecting required attributes

```
Carseats_Filtered <- Carseats %>% select("Sales", "Price",
"Advertising","Population","Age","Income","Education")
```

Building a decision tree regression model to predict Sales based on all other attributes ("Price", "Advertising", "Population", "Age", "Income" and "Education").

```
model_1<- rpart(Sales~.,data=Carseats_Filtered,method = 'anova')
```

```
fancyRpartPlot(model_1)
```



Rattle 2023−May−14 15:21:21 dniha

Price attribute is used at the top of the tree (root node) for the splitting.

Considering the following input: Sales=9,Price=6.54,Population=124,Advertising=0,Age=76,Income=110,Education=10.Estimating Sales for this record using the decision tree model

```
prediction_data = data.frame(Price=6.54 ,Population=124,Advertising=0,Age=76
,Income= 110, Education= 10)


prediction<- predict(model_1,prediction_data)

prediction
```

```
##       1
## 9.58625
```

Predicted sales value for this record is 9.58625.

Using the caret function to train a random forest (method='rf') for the same dataset.

```
set.seed(123)
model_2 <- train(Sales~.,
                 data= Carseats_Filtered,
                 method = "rf")

# Print the results
model_2
```

```
## Random Forest
##
## 400 samples
##   6 predictor
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 400, 400, 400, 400, 400, 400, ...
## Resampling results across tuning parameters:
##
##   mtry  RMSE      Rsquared   MAE
##   2     2.405819  0.2852547  1.926801
##   4     2.421577  0.2790266  1.934608
##   6     2.447373  0.2681323  1.953147
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was mtry = 2.
```

Best results are obtained when mtry value is set to be 2.

Customizing the search grid by checking the model's performance for mtry values of 2, 3 and 5 using 3 repeats of 5-fold cross validation.

```
set.seed(123)

#Cross-Validation
control <- trainControl(method = "repeatedcv",
                        repeats = 3,
                        number = 5)

#Defining search grid with mtry values of 2,3,5
search_grid <- expand.grid(mtry = c(2, 3, 5))


# Training the model using the search grid and cross-validation
model <- train(Sales~., Carseats_Filtered,method = "rf",tuneGrid = search_grid,trControl = control)
print(model)
```

```
## Random Forest
##
## 400 samples
##   6 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold, repeated 3 times)
## Summary of sample sizes: 320, 321, 319, 320, 320, 319, ...
## Resampling results across tuning parameters:
##
##   mtry  RMSE      Rsquared   MAE
##   2     2.405235  0.2813795  1.930855
##   3     2.401365  0.2858295  1.920612
##   5     2.417771  0.2821938  1.934886
##
## RMSE was used to select the optimal model using the smallest value.
```

```
## The final value used for the model was mtry = 3.
```