DATA 622: PREDICTIVE ANALYTICS HW 2

Gabriel Campos

Last edited April 23, 2024

Library

```
library(caret)
library(corrplot)
library(dplyr)
library(e1071)
library(forecast)
library(ggforce)
library(ggplot2)
library(labelled)
library(Metrics)
library(mlbench)
library(ModelMetrics)
library(pROC)
library(psych)
library(RColorBrewer)
library(readr)
library(readxl)
library(randomForest)
library(rpart)
library(rpart.plot)
library(tidymodels)
library(tidyr)
library(tidyverse)
library(tsibble)
```

Decision Trees Algorithms

Pre-work

- Read this blog: https://decizone.com/blog/the-good-the-bad-the-ugly-of-using-decision-trees which shows some of the issues with decision trees
- Choose a dataset from a source in Assignment #1, or another dataset of your choice.
- Assignment work

Based on the latest topics presented, choose a dataset of your choice and create a Decision Tree where you can solve a classification problem and predict the outcome of a particular feature or detail of the data used.

Switch variables* to generate 2 decision trees and compare the results. Create a random forest and analyze the results. Based on real cases where desicion trees went wrong, and 'the bad & ugly' aspects of decision trees (https://decizone.com/blog/the-good-the-bad-the-ugly-of-using-decision-trees), how can you change this perception when using the decision tree you created to solve a real problem?

Deliverable

Essay (minimum 500 word document)

Write a short essay explaining your analysis, and how you would address the concerns in the blog (listed in pre-work) Exploratory Analysis using R or Python (submit code + errors + analysis as notebook or copy/paste to document)

Note:

1. We are trying to train 2 different decision trees to compare bias and variance - so switch the features used for the first node (split) to force a different decision tree (How did the performance change?)
2. You will create 3 models: 2 x decision trees (to compare variance) and a random forest

Data Load

**NOTE: originally attempted with 100k data set but randomforest function would not compute.

EDA

Initial Exploration

```r
head(df_1k)
```

```
##                          Region Country  Item.Type Sales.Channel Order.Priority
## 1 Middle East and North Africa   Libya  Cosmetics       Offline              M
## 2                 North America  Canada Vegetables        Online              M
## 3 Middle East and North Africa   Libya  Baby Food       Offline              C
## 4                          Asia   Japan     Cereal       Offline              C
## 5            Sub-Saharan Africa    Chad     Fruits       Offline              H
## 6                        Europe Armenia     Cereal        Online              H
##   Order.Date  Order.ID  Ship.Date Units.Sold Unit.Price Unit.Cost Total.Revenue
## 1 10/18/2014 686800706 10/31/2014       8446     437.20    263.33    3692591.20
## 2  11/7/2011 185941302  12/8/2011       3018     154.06     90.93     464953.08
## 3 10/31/2016 246222341  12/9/2016       1517     255.28    159.42     387259.76
## 4  4/10/2010 161442649  5/12/2010       3322     205.70    117.11     683335.40
## 5  8/16/2011 645713555  8/31/2011       9845       9.33      6.92      91853.85
## 6 11/24/2014 683458888 12/28/2014       9528     205.70    117.11    1959909.60
##   Total.Cost Total.Profit
## 1  2224085.2   1468506.02
## 2   274426.7    190526.34
## 3   241840.1    145419.62
## 4   389039.4    294295.98
## 5    68127.4     23726.45
## 6  1115824.1    844085.52
```

```r
describe(df_1k)
```

```
##                 vars    n         mean           sd       median      trimmed
## Region*            1 1000         4.30         2.03         4.00         4.37
## Country*           2 1000        92.69        53.82        93.50        92.64
## Item.Type*         3 1000         6.53         3.57         7.00         6.53
## Sales.Channel*     4 1000         1.48         0.50         1.00         1.48
## Order.Priority*    5 1000         2.49         1.12         3.00         2.49
## Order.Date*        6 1000       424.45       244.13       425.00       425.14
## Order.ID           7 1000 549681324.74 257133358.84 556609713.50 550135672.60
## Ship.Date*         8 1000       423.04       241.00       419.50       423.80
## Units.Sold         9 1000      5053.99      2901.38      5184.00      5066.66
## Unit.Price        10 1000       262.11       216.02       154.06       241.99
## Unit.Cost         11 1000       184.97       175.29        97.44       164.10
## Total.Revenue     12 1000   1327321.84   1486514.56    754939.18   1044429.63
## Total.Cost        13 1000    936119.23   1162570.75    464726.06    690338.42
## Total.Profit      14 1000    391202.61    383640.19    277225.98    326888.48
##                          mad         min          max        range  skew
## Region*                 1.48 1.00000e+00         7.00         6.00 -0.09
## Country*               68.94 1.00000e+00       185.00       184.00  0.01
## Item.Type*              4.45 1.00000e+00        12.00        11.00 -0.02
## Sales.Channel*          0.00 1.00000e+00         2.00         1.00  0.08
## Order.Priority*         1.48 1.00000e+00         4.00         3.00 -0.02
## Order.Date*           317.28 1.00000e+00       841.00       840.00 -0.02
## Order.ID        328478990.09 1.02928e+08 995529830.00 892601824.00 -0.02
## Ship.Date*            310.60 1.00000e+00       835.00       834.00 -0.01
## Units.Sold           3766.55 1.30000e+01      9998.00      9985.00 -0.05
## Unit.Price            150.07 9.33000e+00       668.27       658.94  0.79
## Unit.Cost              91.89 6.92000e+00       524.96       518.04  0.95
## Total.Revenue      868353.20 2.04325e+03   6617209.54   6615166.29  1.63
## Total.Cost         548404.86 1.41675e+03   5204978.40   5203561.65  1.79
## Total.Profit       305473.93 5.32610e+02   1726181.36   1725648.75  1.40
##                 kurtosis          se
## Region*            -1.08        0.06
## Country*           -1.20        1.70
## Item.Type*         -1.30        0.11
## Sales.Channel*     -2.00        0.02
## Order.Priority*    -1.37        0.04
## Order.Date*        -1.22        7.72
## Order.ID           -1.19 8131270.76
## Ship.Date*         -1.20        7.62
## Units.Sold         -1.22       91.75
## Unit.Price         -0.74        6.83
## Unit.Cost          -0.61        5.54
## Total.Revenue       2.04    47007.72
## Total.Cost          2.53    36763.72
## Total.Profit        1.58    12131.77
```

```r
str(df_1k)
```

```
## 'data.frame':    1000 obs. of  14 variables:
##  $ Region        : chr  "Middle East and North Africa" "North America" "Middle East and North Africa
##  $ Country       : chr  "Libya" "Canada" "Libya" "Japan" ...
```

```
##  $ Item.Type     : chr  "Cosmetics" "Vegetables" "Baby Food" "Cereal" ...
##  $ Sales.Channel : chr  "Offline" "Online" "Offline" "Offline" ...
##  $ Order.Priority: chr  "M" "M" "C" "C" ...
##  $ Order.Date    : chr  "10/18/2014" "11/7/2011" "10/31/2016" "4/10/2010" ...
##  $ Order.ID      : int  686800706 185941302 246222341 161442649 645713555 683458888 679414975 208630
##  $ Ship.Date     : chr  "10/31/2014" "12/8/2011" "12/9/2016" "5/12/2010" ...
##  $ Units.Sold    : int  8446 3018 1517 3322 9845 9528 2844 7299 2428 4800 ...
##  $ Unit.Price    : num  437.2 154.06 255.28 205.7 9.33 ...
##  $ Unit.Cost     : num  263.33 90.93 159.42 117.11 6.92 ...
##  $ Total.Revenue : num  3692591 464953 387260 683335 91854 ...
##  $ Total.Cost    : num  2224085 274427 241840 389039 68127 ...
##  $ Total.Profit  : num  1468506 190526 145420 294296 23726 ...
```

```
summary(df_1k)
```

```
##     Region            Country            Item.Type         Sales.Channel
##  Length:1000        Length:1000        Length:1000        Length:1000
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##  Order.Priority      Order.Date           Order.ID          Ship.Date
##  Length:1000        Length:1000        Min.   :102928006   Length:1000
##  Class :character   Class :character   1st Qu.:328074026   Class :character
##  Mode  :character   Mode  :character   Median :556609714   Mode  :character
##                                        Mean   :549681325
##                                        3rd Qu.:769694483
##                                        Max.   :995529830
##    Units.Sold     Unit.Price       Unit.Cost      Total.Revenue
##  Min.   :  13   Min.   :  9.33   Min.   :  6.92   Min.   :   2043
##  1st Qu.:2420   1st Qu.: 81.73   1st Qu.: 56.67   1st Qu.: 281192
##  Median :5184   Median :154.06   Median : 97.44   Median : 754939
##  Mean   :5054   Mean   :262.11   Mean   :184.97   Mean   :1327322
##  3rd Qu.:7537   3rd Qu.:421.89   3rd Qu.:263.33   3rd Qu.:1733503
##  Max.   :9998   Max.   :668.27   Max.   :524.96   Max.   :6617210
##    Total.Cost        Total.Profit
##  Min.   :   1417   Min.   :    532.6
##  1st Qu.: 164932   1st Qu.:  98376.1
##  Median : 464726   Median : 277226.0
##  Mean   : 936119   Mean   : 391202.6
##  3rd Qu.:1141750   3rd Qu.: 548456.8
##  Max.   :5204978   Max.   :1726181.4
```

```
glimpse(df_1k)
```

```
## Rows: 1,000
## Columns: 14
## $ Region         <chr> "Middle East and North Africa", "North America", "Middl~
## $ Country        <chr> "Libya", "Canada", "Libya", "Japan", "Chad", "Armenia",~
## $ Item.Type      <chr> "Cosmetics", "Vegetables", "Baby Food", "Cereal", "Frui~
## $ Sales.Channel  <chr> "Offline", "Online", "Offline", "Offline", "Offline", "~
## $ Order.Priority <chr> "M", "M", "C", "C", "H", "H", "H", "M", "H", "H", "M", ~
```

```
## $ Order.Date     <chr> "10/18/2014", "11/7/2011", "10/31/2016", "4/10/2010", "~
## $ Order.ID       <int> 686800706, 185941302, 246222341, 161442649, 645713555, ~
## $ Ship.Date      <chr> "10/31/2014", "12/8/2011", "12/9/2016", "5/12/2010", "8~
## $ Units.Sold     <int> 8446, 3018, 1517, 3322, 9845, 9528, 2844, 7299, 2428, 4~
## $ Unit.Price     <dbl> 437.20, 154.06, 255.28, 205.70, 9.33, 205.70, 205.70, 1~
## $ Unit.Cost      <dbl> 263.33, 90.93, 159.42, 117.11, 6.92, 117.11, 117.11, 35~
## $ Total.Revenue  <dbl> 3692591.20, 464953.08, 387259.76, 683335.40, 91853.85, ~
## $ Total.Cost     <dbl> 2224085.18, 274426.74, 241840.14, 389039.42, 68127.40, ~
## $ Total.Profit   <dbl> 1468506.02, 190526.34, 145419.62, 294295.98, 23726.45, ~
```

```r
look_for(df_1k)
```

```
##  pos variable       label col_type missing values
##  1   Region         -     chr      0
##  2   Country        -     chr      0
##  3   Item.Type      -     chr      0
##  4   Sales.Channel  -     chr      0
##  5   Order.Priority -     chr      0
##  6   Order.Date     -     chr      0
##  7   Order.ID       -     int      0
##  8   Ship.Date      -     chr      0
##  9   Units.Sold     -     int      0
##  10  Unit.Price     -     dbl      0
##  11  Unit.Cost      -     dbl      0
##  12  Total.Revenue  -     dbl      0
##  13  Total.Cost     -     dbl      0
##  14  Total.Profit   -     dbl      0
```

```r
apply(df_1k, 2, function(x) sum(is.na(x)))
```

```
##         Region        Country      Item.Type  Sales.Channel Order.Priority
##              0              0              0              0              0
##     Order.Date       Order.ID      Ship.Date     Units.Sold     Unit.Price
##              0              0              0              0              0
##      Unit.Cost  Total.Revenue     Total.Cost   Total.Profit
##              0              0              0              0
```

```r
unique(df_1k$Region)
```

```
## [1] "Middle East and North Africa"      "North America"
## [3] "Asia"                              "Sub-Saharan Africa"
## [5] "Europe"                            "Central America and the Caribbean"
## [7] "Australia and Oceania"
```

```r
#unique(df_1k$Country)
```

```r
length(unique(df_1k$Country))
```

```
## [1] 185
```

```r
table(df_1k$Item.Type)
```

```
## 
##       Baby Food       Beverages          Cereal         Clothes       Cosmetics 
##              87             101              79              78              75 
##          Fruits       Household            Meat Office Supplies   Personal Care 
##              70              77              78              89              87 
##          Snacks      Vegetables 
##              82              97
```

```r
table(df_1k$Sales.Channel)
```

```
## 
## Offline  Online 
##     520     480
```

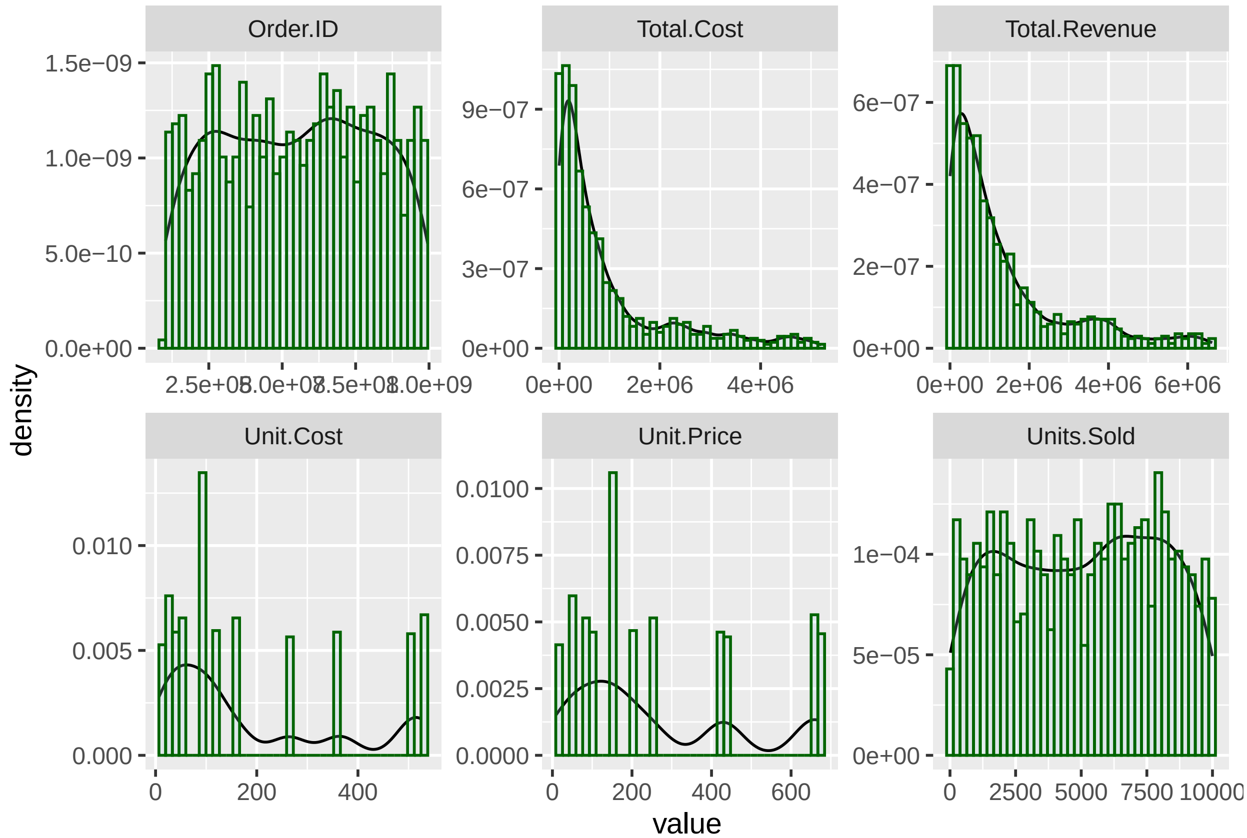
```r
unique(df_1k$Order.Priority)
```

```
## [1] "M" "C" "H" "L"
```

```r
#select numeric columns 1k
df_1k_num <- df_1k %>%
  keep(is.numeric)

#stats
describe(df_1k_num, fast=TRUE) %>%
  select(c(-vars,-n))
```

```
##                      mean           sd         min          max        range
## Order.ID      549681324.74 257133358.84 1.02928e+08 995529830.00 892601824.00
## Units.Sold        5053.99      2901.38 1.30000e+01      9998.00      9985.00
## Unit.Price         262.11       216.02 9.33000e+00       668.27       658.94
## Unit.Cost          184.97       175.29 6.92000e+00       524.96       518.04
## Total.Revenue   1327321.84   1486514.56 2.04325e+03   6617209.54   6615166.29
## Total.Cost       936119.23   1162570.75 1.41675e+03   5204978.40   5203561.65
## Total.Profit     391202.61    383640.19 5.32610e+02   1726181.36   1725648.75
##                      se
## Order.ID      8131270.76
## Units.Sold        91.75
## Unit.Price         6.83
## Unit.Cost          5.54
## Total.Revenue   47007.72
## Total.Cost      36763.72
## Total.Profit    12131.77
```

```r
#distributions
df_1k_num %>%
  pivot_longer(cols = 1:6, names_to = "variable", values_to = "value") %>%
  ggplot(aes(value)) +
    facet_wrap(~variable, scales = "free") +
    geom_density() +
    geom_histogram(aes(y = after_stat(density)), bins = 40, alpha = 0.2, fill = "lightblue", color = "da
```



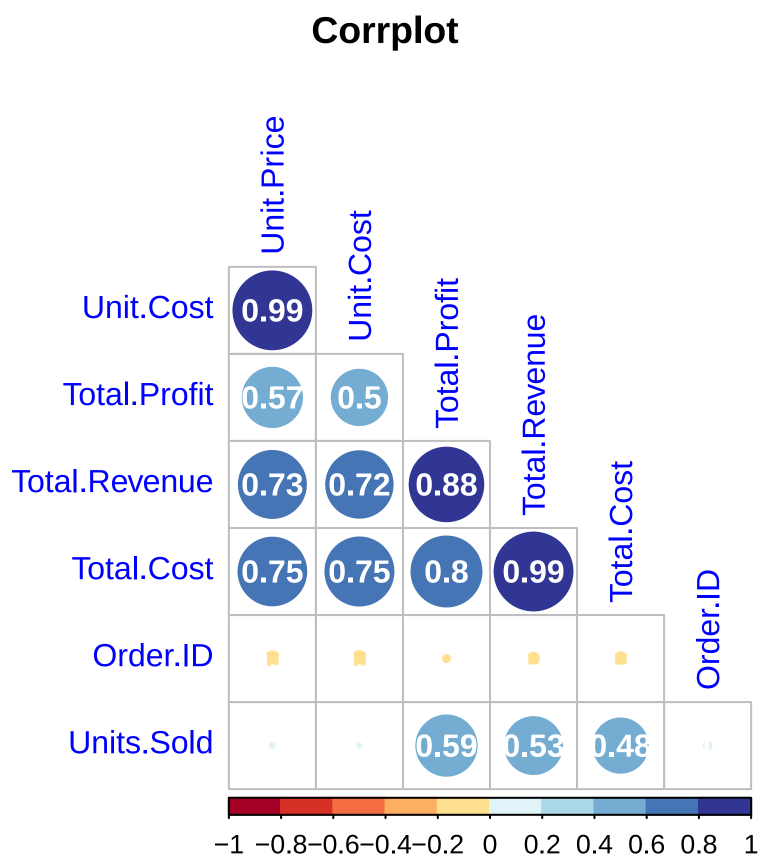
From the initial EDA we see the following:

- The data set is 1,000 rows and 14 columns
- No labels are found in the variables
- High range among the integers and doubles
- Variable types include:
  - 2 integers, 5 doubles and 7 character types
- 5 regions are noted with 185 countries associated with it
- Priority is categorized C(Critical), H(High), M(Medium), and L(Low)
- No variables seem to be missing values
- Dependencies among the variables are as follows:
  - $Total.Cost = Units.Sold \times Unit.Cost$
  - $Total.Revenue - Units.Sold \times Unit.Price$
  - $Total.Profit - Total.Revenue - Total.Cost$
  - $Total.Cost$ and $Total.Revenue$ depends on $Units.Sold, Units.Cost\ and\ Unit.Price$
- Distribution of the data is noted with several skewed variables which will need transformation and normalizing

Correlation

```
corr_matrix <- cor(df_1k_num)
corrplot(corr_matrix,
```

```
        type = "lower",
        order = "hclust",
        tl.col = "blue",
        addCoef.col = "white",
        diag = FALSE,
        title = "Corrplot",
        mar = c(0, 0, 1, 0),
        col = brewer.pal(10, "RdYlBu"))
```



Looking at the correlation plot we see the following:

- Weak correlation between Unit.Price, Unit.Cost and Units.Sold
- Mild correlation between Total.Profit, Total.Revenue, Total.Cost and Units.Sold
- Mild correlation between Unit.Price, Unit.Cost and Total.Profit
- High correlation between Unit.Price and Unit.Cost
- High correlation between Total.Profit and Total Revenue
- High correlation between Total,Cost and Total.Revenue

I suspect multicollinearity but will use and additional method to confirm.

VIF

```r
set.seed(321)

sample_1k_train <- df_1k_num$Total.Revenue %>%
  createDataPartition(p = 0.8, list = FALSE)
df_train_1k  <- df_1k_num[sample_1k_train, ]
df_test_1k <- df_1k_num[-sample_1k_train, ]


model<- lm(Total.Revenue~., data=df_train_1k )

vif_values<-car::vif(model)

print(vif_values)
```

```
##     Order.ID   Units.Sold   Unit.Price    Unit.Cost   Total.Cost Total.Profit
##     1.002970     3.041373   167.637725   167.273600    11.445468    14.149909
```

The values interpret as: * Order.ID has low multicollinearity * Units.Sold low multicollinearity * Unit.Price and Unit.Cost has high levels of multicollinearity * Total.Cost and Total.Profit has moderate levels of multicollinearity.

Transformation

Only transformation needed are: * date values to Month, Day and Year * levels for categorical values. * scaling for pre-processing for modelling * Attribute selection of relevant data will also be best

```r
df_1k[['Order.Date']] <- as.Date(df_1k[['Order.Date']], "%m/%d/%Y")
df_1k[['Ship.Date']] <- as.Date(df_1k[['Ship.Date']], "%m/%d/%Y")

df_1k[['Sales.Channel']] <- as.factor(df_1k[['Sales.Channel']])

df_1k[['Order.Priority']] <- as.factor(df_1k[['Order.Priority']])

df_1k[['Item.Type']] <- as.factor(df_1k[['Item.Type']])

df_1k[['Region']] <- as.factor(df_1k[['Region']])

df_1k[['Country']] <- as.factor(df_1k[['Country']])

df_1k[['Order.ID']] <- as.character(df_1k[['Order.ID']])

df_1k_norm<-predict(preProcess(df_1k, method=c("center", "scale")),df_1k)
```

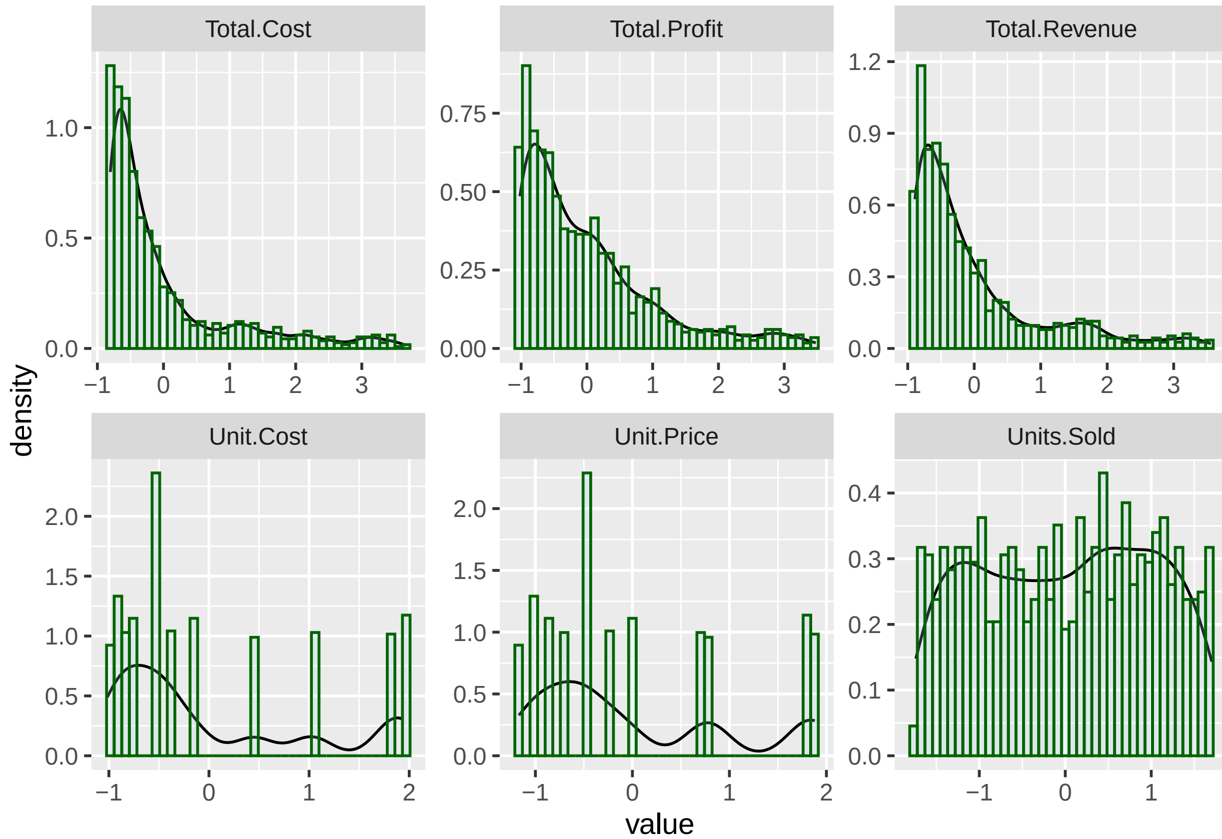
```r
df_1k_norm %>%
  keep(is.numeric) %>%
  describe(fast=TRUE) %>%
  select(-c(vars,n))
```

```
##                 mean sd   min  max range   se
## Units.Sold         0  1 -1.74 1.70  3.44 0.03
```

```
## Unit.Price       0  1 -1.17 1.88  3.05 0.03
## Unit.Cost        0  1 -1.02 1.94  2.96 0.03
## Total.Revenue    0  1 -0.89 3.56  4.45 0.03
## Total.Cost       0  1 -0.80 3.67  4.48 0.03
## Total.Profit     0  1 -1.02 3.48  4.50 0.03
```

```r
df_1k_norm %>%
  select(where(is.numeric)) %>%   # keep numeric columns
  {list(summary = summary(.),
        plot = ggplot(tidyr::pivot_longer(., cols = everything()),
                      aes(value)) +
               facet_wrap(~name, scales = "free") +
               geom_density() +
               geom_histogram(aes(y=after_stat(density)), alpha=0.2, fill = "lightblue",
                              color="darkgreen", position="identity", bins = 40))
  }
```

```
## $summary
##    Units.Sold         Unit.Price        Unit.Cost        Total.Revenue
##  Min.   :-1.73745   Min.   :-1.1701   Min.   :-1.0157   Min.   :-0.8915
##  1st Qu.:-0.90775   1st Qu.:-0.8350   1st Qu.:-0.7319   1st Qu.:-0.7037
##  Median : 0.04481   Median :-0.5002   Median :-0.4993   Median :-0.3851
##  Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.85572   3rd Qu.: 0.7397   3rd Qu.: 0.4471   3rd Qu.: 0.2732
##  Max.   : 1.70402   Max.   : 1.8802   Max.   : 1.9396   Max.   : 3.5586
##    Total.Cost        Total.Profit
##  Min.   :-0.8040   Min.   :-1.0183
##  1st Qu.:-0.6633   1st Qu.:-0.7633
##  Median :-0.4055   Median :-0.2971
##  Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.1769   3rd Qu.: 0.4099
##  Max.   : 3.6719   Max.   : 3.4798
##
## $plot
```




```
df_1k_norm <- df_1k_norm %>%
  select(-c(Country,Order.ID,))
```

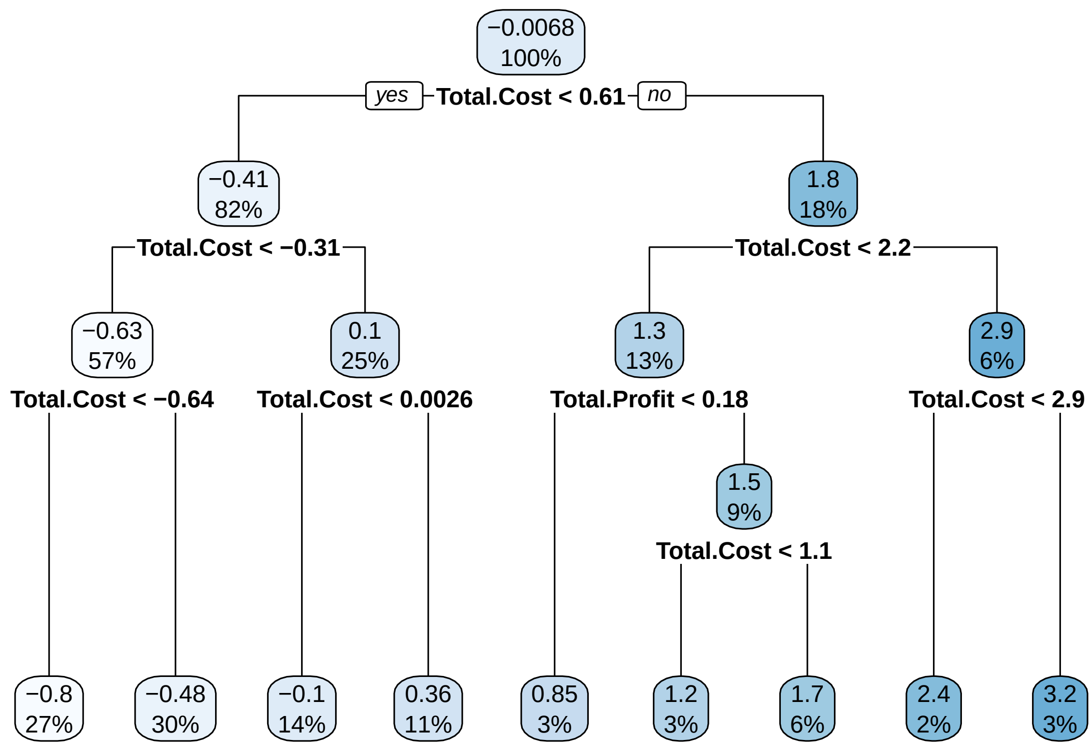
Models

Regression trees

Model 1

```
set.seed(1234)

df1k_norm1 <- df_1k_norm

#split
training_1k_samples <- df1k_norm1$Total.Revenue %>%
  createDataPartition(p = 0.8, list = FALSE)

train_1k1  <- df1k_norm1[training_1k_samples, ]
test_1k1 <- df1k_norm1[-training_1k_samples, ]

#train using rpart, cp- complexity, smaller # = more complexity,
#method- anova is for regression
```

```r
tree_1k1 <- rpart(Total.Revenue ~., data = train_1k1, cp = 0.004,  method = 'anova')

#visualize
rpart.plot(tree_1k1)
```




```r
print(tree_1k1)
```

```
## n= 800
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 800 789.9383000 -0.006752507
##    2) Total.Cost< 0.6147312 654 108.3324000 -0.410574000
##      4) Total.Cost< -0.3106351 456  16.8432000 -0.634194800
##        8) Total.Cost< -0.6429517 216   1.1617710 -0.802689100 *
##        9) Total.Cost>=-0.6429517 240   4.0300370 -0.482550000 *
##      5) Total.Cost>=-0.3106351 198  16.1706800  0.104431600
##       10) Total.Cost< 0.002629725 109   1.8064560 -0.101871600 *
##       11) Total.Cost>=0.002629725 89   4.0434160  0.357095100 *
##    3) Total.Cost>=0.6147312 146  97.2280900  1.802146000
##      6) Total.Cost< 2.244415 102  19.0550000  1.348138000
##       12) Total.Profit< 0.1788319 27   1.5098580  0.846133500 *
##       13) Total.Profit>=0.1788319 75   8.2913770  1.528860000
##         26) Total.Cost< 1.124162 25   1.4624880  1.217733000 *
```

```
##         27) Total.Cost>=1.124162 50   3.1988910  1.684424000 *
##      7) Total.Cost>=2.244415 44   8.4097410  2.854619000
##       14) Total.Cost< 2.891512 18   0.9282189  2.399053000 *
##       15) Total.Cost>=2.891512 26   1.1594990  3.170012000 *
```

Predictions

```r
predictions <- predict(tree_1k1, newdata = test_1k1) %>%
  bind_cols(test_1k1 )

predictions$...1 <- as.numeric(predictions$...1)
```

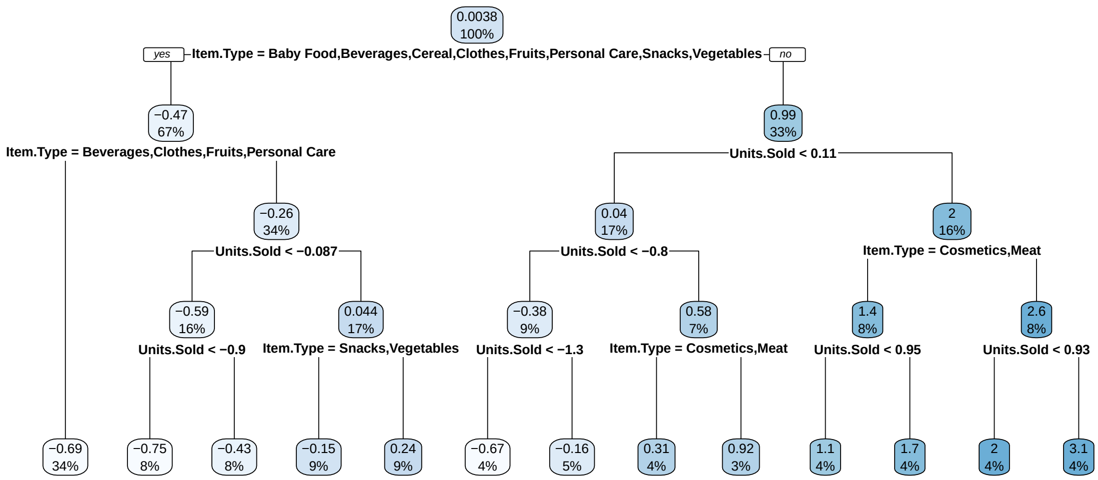
Performance

```r
decision_tree_model <- data.frame(Model = "Decision Tree 1",

MAE = ModelMetrics::mae(predictions$Total.Revenue, predictions$...1),
#rmse Root Mean Squared Error
RMSE = ModelMetrics::rmse(predictions$Total.Revenue, predictions$...1),
#r squared
R2 = caret::R2(predictions$Total.Revenue, predictions$...1)
)

decision_tree_model
```

```
##             Model       MAE      RMSE        R2
## 1 Decision Tree 1 0.1215621 0.1660607 0.9737615
```

Model 2

```r
set.seed(4321)

df_1k_norm2 <- df_1k_norm %>%
  select(-c("Unit.Price","Unit.Cost","Total.Cost", "Total.Profit"))

#split
training_1k_samples2 <- df_1k_norm2$Total.Revenue %>%
  createDataPartition(p = 0.8, list = FALSE)

train_1k2  <- df_1k_norm2[training_1k_samples2, ]
test_1k2 <- df_1k_norm2[-training_1k_samples2, ]

#train using rpart, cp- complexity, smaller # = more complexity,
#method- anova is for regression
tree_1k2 <- rpart(Total.Revenue ~., data = train_1k2, cp = 0.004, method = 'anova')

#visualize
rpart.plot(tree_1k2)
```




```r
print(tree_1k2)
```

```
## n= 800
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 800 822.8269000  0.003789579
##    2) Item.Type=Baby Food,Beverages,Cereal,Clothes,Fruits,Personal Care,Snacks,Vegetables 539  75.41
##      4) Item.Type=Beverages,Clothes,Fruits,Personal Care 270   8.8690330 -0.685383800 *
##      5) Item.Type=Baby Food,Cereal,Snacks,Vegetables 269  42.5309600 -0.263248700
##       10) Units.Sold< -0.08719589 131   4.9589340 -0.587279500
##         20) Units.Sold< -0.9037052 64   0.4596006 -0.751204600 *
##         21) Units.Sold>=-0.9037052 67   1.1367960 -0.430694400 *
##       11) Units.Sold>=-0.08719589 138  10.7607700  0.044345760
##         22) Item.Type=Snacks,Vegetables 69   1.2076570 -0.148450800 *
##         23) Item.Type=Baby Food,Cereal 69   4.4235870  0.237142300 *
##    3) Item.Type=Cosmetics,Household,Meat,Office Supplies 261 369.1487000  0.991951000
##      6) Units.Sold< 0.1121923 133  47.2663700  0.039783430
##       12) Units.Sold< -0.7953083 75   6.9743660 -0.381010600
##         24) Units.Sold< -1.301275 32   0.6975396 -0.671731000 *
##         25) Units.Sold>=-1.301275 43   1.5595200 -0.164660400 *
##       13) Units.Sold>=-0.7953083 58   9.8394350  0.583913600
##         26) Item.Type=Cosmetics,Meat 32   1.5942220  0.310831200 *
##         27) Item.Type=Household,Office Supplies 26   2.9217760  0.920015000 *
```

```
##      7) Units.Sold>=0.1121923 128  76.0103700  1.981313000
##       14) Item.Type=Cosmetics,Meat 63   7.7936750  1.394445000
##         28) Units.Sold< 0.9454178 28   0.7889117  1.056832000 *
##         29) Units.Sold>=0.9454178 35   1.2600410  1.664536000 *
##       15) Item.Type=Household,Office Supplies 65  25.4882900  2.550122000
##         30) Units.Sold< 0.9302526 32   3.0820170  1.987527000 *
##         31) Units.Sold>=0.9302526 33   2.4563190  3.095669000 *
```

Predictions

```r
predictions2 <- predict(tree_1k2, newdata = test_1k2) %>%
  bind_cols(test_1k2)

predictions2$...1 <- as.numeric(predictions2$...1)
```

Performance

```r
decision_tree_model2 <- data.frame(Model = "Decision Tree 2",
#mean absolute error
MAE = ModelMetrics::mae(predictions2$Total.Revenue, predictions2$...1),
#rmse Root Mean Squared Error
RMSE = ModelMetrics::rmse(predictions2$Total.Revenue, predictions2$...1),
#r squared
R2 = caret::R2(predictions2$Total.Revenue, predictions2$...1)
)

decision_tree_model2
```

```
##             Model       MAE      RMSE        R2
## 1 Decision Tree 2 0.1645788 0.2060202 0.9540491
```

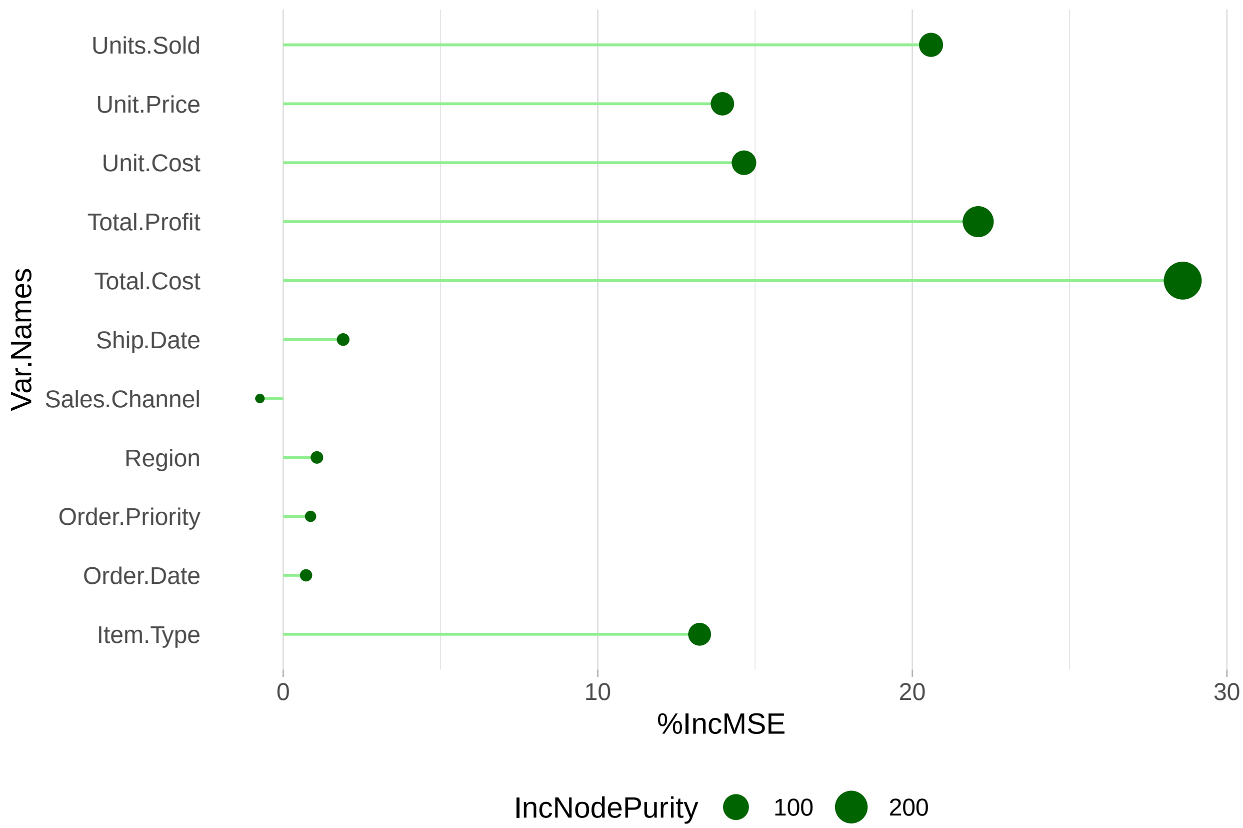
Random Forest Regression Tree

```r
set.seed(222)
rf <- randomForest::randomForest(formula = Total.Revenue ~ .,
                   data = train_1k1, importance=TRUE)
```

```r
rf
```

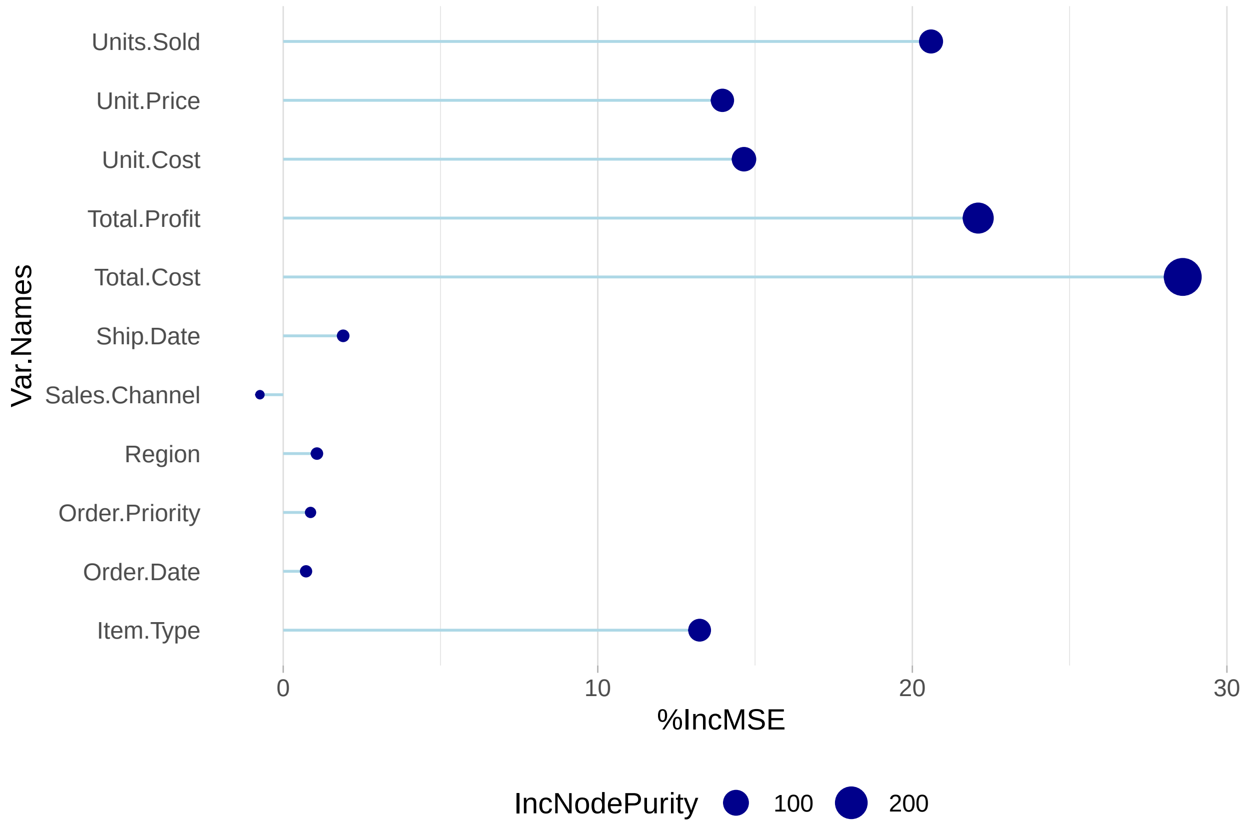
```
##
## Call:
##  randomForest(formula = Total.Revenue ~ ., data = train_1k1, importance = TRUE)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##           Mean of squared residuals: 0.001567821
##                     % Var explained: 99.84
```

```
ImpData <- as.data.frame(importance(rf))
ImpData$Var.Names <- row.names(ImpData)

ggplot(ImpData, aes(x=Var.Names, y=`%IncMSE`)) +
  geom_segment( aes(x=Var.Names, xend=Var.Names, y=0, yend=`%IncMSE`), color="lightgreen") +
  geom_point(aes(size = IncNodePurity),  color="darkgreen", alpha=1) +
  theme_light() +
  coord_flip() +
  theme(
    legend.position="bottom",
    panel.grid.major.y = element_blank(),
    panel.border = element_blank(),
    axis.ticks.y = element_blank()
  )
```




```
ggplot(ImpData, aes(x=Var.Names, y=`%IncMSE`)) +
  geom_segment( aes(x=Var.Names, xend=Var.Names, y=0, yend=`%IncMSE`), color="lightblue") +
  geom_point(aes(size = IncNodePurity),  color="darkblue", alpha=1) +
  theme_light() +
  coord_flip() +
  theme(
    legend.position="bottom",
    panel.grid.major.y = element_blank(),
    panel.border = element_blank(),
    axis.ticks.y = element_blank()
```

```
)
```



Predictions

```
predictions3 <- predict(rf, newdata = test_1k1) %>%
  bind_cols(test_1k1)

predictions3$...1 <- as.numeric(predictions3$...1)
```

Performance

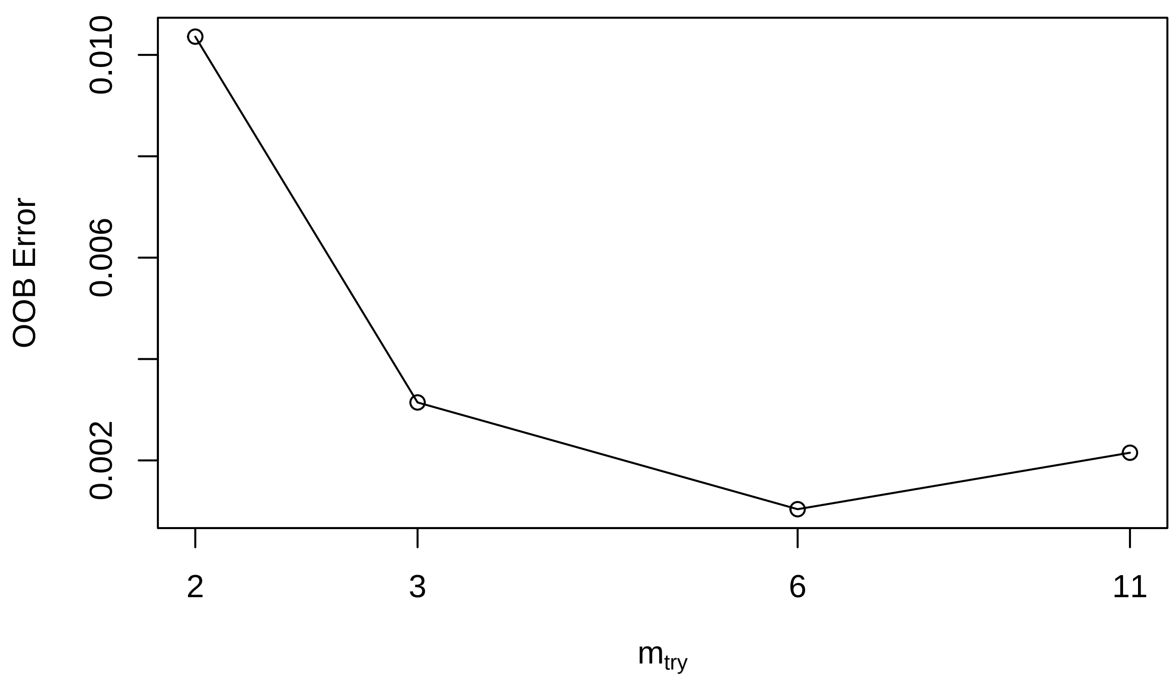
```
random_forest_model <- data.frame(Model = "Random Forest",
#mean absolute error
MAE = ModelMetrics::mae(predictions3$Total.Revenue, predictions3$...1),
#rmse Root Mean Squared Error
RMSE = ModelMetrics::rmse(predictions3$Total.Revenue, predictions3$...1),
#r squared
R2 = R2(predictions3$Total.Revenue, predictions3$...1)
)

random_forest_model
```

```
##           Model       MAE       RMSE        R2
## 1 Random Forest 0.0209061 0.04107028 0.9988368
```

Tuned Random Forest Regression Tree

```
set.seed(333)

train_tuned_rf <- train_1k1 %>%
  select(-Total.Revenue)

bestmtry <- tuneRF(train_tuned_rf,train_1k1$Total.Revenue, stepFactor = 2, improve = 0.01,
                   trace=T, plot= T, doBest=TRUE, importance=TRUE)
```

```
## mtry = 3  OOB error = 0.003143373
## Searching left ...
## mtry = 2     OOB error = 0.01035999
## -2.29582 0.01
## Searching right ...
## mtry = 6     OOB error = 0.001037668
## 0.6698873 0.01
## mtry = 11    OOB error = 0.002151529
## -1.073428 0.01
```



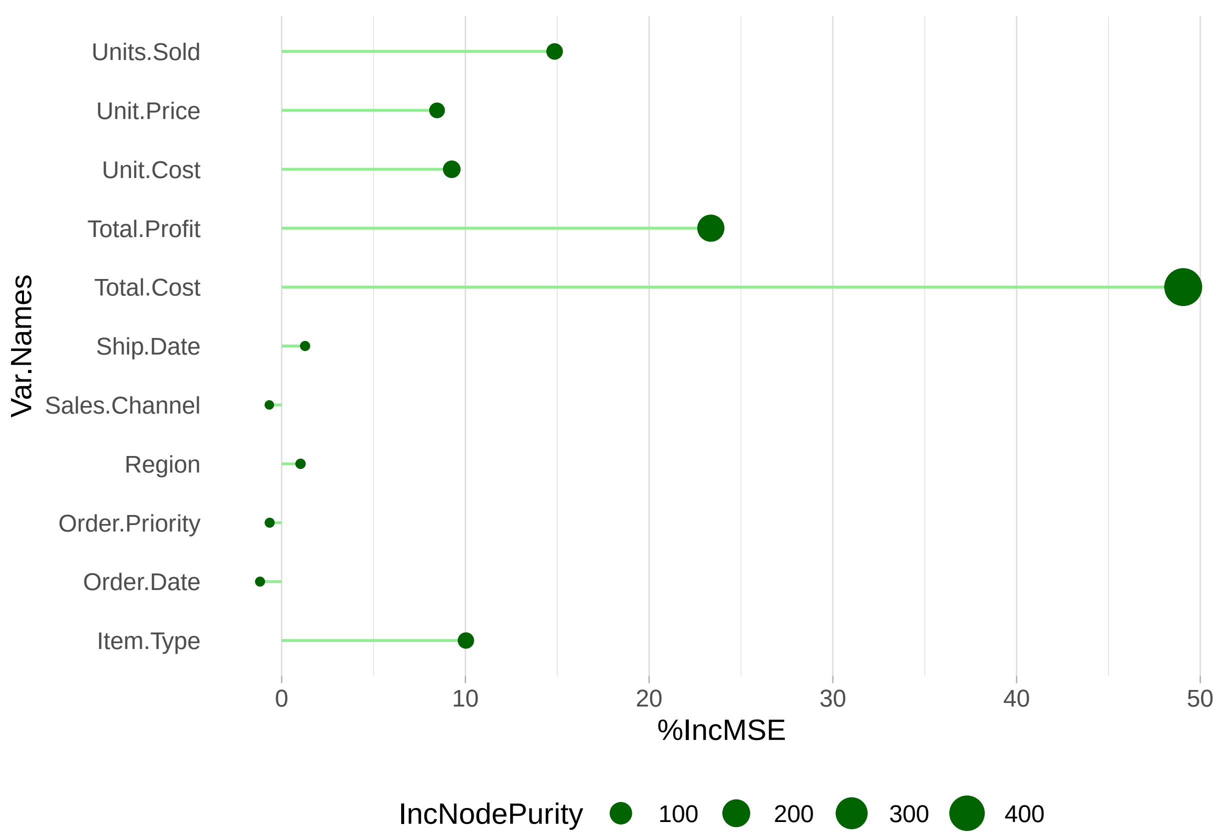

```
bestmtry
```

```
##
```

```
## Call:
##  randomForest(x = x, y = y, mtry = res[which.min(res[, 2]), 1],      importance = TRUE)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 6
##
##           Mean of squared residuals: 0.0005806283
##                     % Var explained: 99.94
```

```r
#importance(bestmtry)

# Get variable importance from the model fit
ImpData <- as.data.frame(importance(bestmtry))
ImpData$Var.Names <- row.names(ImpData)
```

```r
ggplot(ImpData, aes(x=Var.Names, y=`%IncMSE`)) +
  geom_segment( aes(x=Var.Names, xend=Var.Names, y=0, yend=`%IncMSE`), color="lightgreen") +
  geom_point(aes(size = IncNodePurity), color="darkgreen", alpha=1) +
  theme_light() +
  coord_flip() +
  theme(
    legend.position="bottom",
    panel.grid.major.y = element_blank(),
    panel.border = element_blank(),
    axis.ticks.y = element_blank()
  )
```



Predictions

```
predictions4 <- predict(bestmtry, newdata = test_1k1) %>%
  bind_cols(test_1k1)

predictions4$...1 <- as.numeric(predictions4$...1)
```

Model Performance

```
random_forest_tuned_model <- data.frame(Model = "Tuned Random Forest",
#mean absolute error
MAE = ModelMetrics::mae(predictions4$Total.Revenue, predictions4$...1),
#rmse Root Mean Squared Error
RMSE = ModelMetrics::rmse(predictions4$Total.Revenue, predictions4$...1),
#r squared
R2 = caret::R2(predictions4$Total.Revenue, predictions4$...1)
)

random_forest_tuned_model
```

```
##                 Model        MAE       RMSE        R2
## 1 Tuned Random Forest 0.01196328 0.02322624 0.9995862
```

Essay

This assignment is a build-on to HW1, with an implementation of randomrorest algorithm. Originally my goal for the assignment was to incorporate the 100k dataset with 100k observations used for HW1. The initial plan was to use to assess performance and practicality or the randomforest and decision tree, after assessing the best way to transform the data. Afterwards, for my benefit I would compare to my original assignment and learn from the experience. An issue that arose was with the randomforest method and the large data set. The size created to big a computation load and cause the function to cycle with not result. Due to the submission deadline, I chose to utilize the 1k dataset for this assignment as a result. For my own benefit, I will rerun the function on my own time, to get a gauge on time needed for the computation to complete. Understanding the time needed for this method, would be useful if I chose to use randomforest again in the future. In this assignment I also utilized VIF scores to better assess the level of multicollinearity, which in the HW1 was only assessed with a correlation plot. During the EDA stage of the data, a few transformations were identified before moving to the modelling for this data. Categorical data was ranked, and the dates were defined as dates before proceeding. The distribution of the data was shown to be skewed in some case and the correlation plot showed, that numeric values would be best to utilize with my model. There was very little correlation with the categorical or data values and so those attributes were removed. All numerical data was used regardless if they showed multicollinearity which we identified using VIF. Preprocess function was used for scaling. The motivation behind using this function, was to ensure the values would contribute equally to the analysis, which can be impacted if ranges vary more among the attributes. For models 1 & 2 a decision tree was use. For Model 2, highly correlated variables were removed to assess the impact. I expected Model 2 to out perform on all levels, but it only retained a higher R2 value, which means a higher proportion of the variance is explained by the model, however Model 1 had a higher MAE and RMSE indicating better precision and accuracy. Random forest also had similar results, with a higher R2 but also higher RMSE and MAE, indicating a larger proportion of the dependent variable is explained, while technically being lower in accuracy and precision. Tuning random forest gave some improvement in the area of precision and accuracy, RMSE and MAE, while performing the best as indicated by the R2. However, when compared to the decision tree its RMSE and MAE values is slightly higher. I imagine this data and results would differ if a larger dataset was used, and I intend to rerun this work on my own after the assignment it submitted.