**BIG MART SALES PRACTICE PROBLEM**

**ANALYTICS VIDHYA**

**Introduction:**

The data scientists at BigMart have collected 2013 sales data for 1559 products across 10 stores in different cities. The main objective is to understand whether specific properties of products and/or stores play a significant role in terms of increasing or decreasing sales volume. To achieve this goal, we will build a predictive model and find out the sales of each product at a particular store. This will help BigMart to boost their sales by learning optimised product organization inside stores.

**Data:**

I got the data from a renowned website for data analytics named, Analytics Vidhya. <https://datahack.analyticsvidhya.com/contest/practice-problem-big-mart-sales-iii/>

**Problem Statement:**

The data scientists at BigMart have collected 2013 sales data for 1559 products across 10 stores in different cities. Also, certain attributes of each product and store have been defined. The aim is to build a predictive model and find out the sales of each product at a particular store.

Using this model, BigMart will try to understand the properties of products and stores which play a key role in increasing sales.

**Data Description:**

We have train (8523) and test (5681) data set, train data set has both input and output variable(s). You need to predict the sales for test data set.

|  |  |
| --- | --- |
| **Variable** | **Description** |
| Item\_Identifier | Unique product ID |
| Item\_Weight | Weight of Product |
| Item\_Fat\_Content | Whether the product is low fat or not |
| Item\_Visibility | The % of total display area of all products in a store allocated to the particular product |
| Item\_Type | Category to which product belongs |
| Item\_MRP | Maximum Retail Price(list price) of the product |
| Outlet\_Identifier | Unique store ID |
| Outlet\_Establishment\_Year | The year in which store was established |
| Outlet\_Size | Size of the store |
| Outlet\_Location\_Type | The type of city in which the store is located |
| Outlet\_Type | Grocery store or some sort of supermarket |
| Item\_Outlet\_Sales | Sales of the product in the particular store. This is the outcome variable to the predicted. |

**Data Exploration:**

**Factor Attributes:**

**Item\_Identifier**: Unique Product ID.

Item\_Identifier column has 1559 unique value. So, we can say we will examine 1559 unique item properties due to the stores.

**Item\_Fat\_Content**: Whether the product is low fat or not.

Low fat items have highest rate, so we can say low fat items reside on stores generally. There is corruption about string values, such as “Low fat”/“low fat”/“LF” or “Regular” / “reg” , we know these string values represent same value and so, we should clean this column values in cleaning part.

**Item\_Type**: The category to which the product belongs.

Train dataset has 16 different Item\_type value and “Fruits and Vegetables” items reside on stores popularly.

**Outlet\_Identifier:** Unique Store ID

Outlet\_Identifier data show that train dataset includes item information of 10 different stores.

**Outlet\_Size:** The size of the store in terms of ground area covered.

Outlet\_size have some null values. So, on cleaning part we should consider about this column, also.

**Outlet\_Location\_Type**: The type of city in which the store is located

Data of Outlet\_Location\_Type column distribution is fair for each type.

**Outlet\_Type:** Whether the outlet is just a grocery store or some sort of supermarket

We have item information on “Supermarket Type1” store with the highest rate It means, our data includes “Supermarket Type1” store mostly.

**Numerical Attributes:**

**Item\_Weight**: Weight of product.

Item\_Weight column has 1463 null value, so cleaning part we should consider this column also. When we exclude the null values, the heaviest item is 21.35 gr and the lightest ite is 4.555 gr.

**Item\_Visibility:** The % of total display area of all products in a store allocated to the particular product.

There is no null value on Item\_Visibility column but from minimum value we can say there is “0” value on column.

**Item\_MRP**: Maximum Retail Price (list price) of the product.

There is no null or “0” value on ITEM\_MRP column. Max price is 266.8884 and min price is 31.29.

**Outlet\_Establishment\_Year**: The year in which store was established.

The oldest store has opened in 1985 and the newest one has opened in 2009. When we look at mean and median, we can say stores in dataset are generally old stores. (More than 17-18 years old.)

**Item\_Outlet\_Sales**: Sales of the product in the particular store. This is the outcome variable to be predicted

Item\_Outlet\_Sales column has no null values.

**Data Manipulation:**

**Train Data:**

* Recoding the Low Fat variables to 1 & Regular variables to 2 of Item\_Fat\_Content Attribute in train data.

**R Code:**

library(car)

table(train[,3])

fattrainencode <- recode(fattrain,"'LF'=1;'low fat'=1;'Low Fat'=1;'reg'=2;'Regular'=2" )

trainclass<- train['Item\_Outlet\_Sales']

table (fattrainencode)

**Output:**

>table(train[,3])

LF low fat Low Fat reg Regular

316 112 5089 117 2889

> fattrain<- train[,3]

>

> #Recoding the Low Fat variables to 1 & Regular variables to 2 of Item\_Fat\_Content Attribute in train data

>

> library(car)

Loading required package: carData

Attaching package: ‘car’

The following object is masked from ‘package:arules’:

recode

> fattrainencode <- recode(fattrain,"'LF'=1;'low fat'=1;'Low Fat'=1;'reg'=2;'Regular'=2" )

> table(fattrainencode)

fattrainencode

1 2

5517 3006

* Recoding the Tier 1 variables to 1,Tier 2 variables to 2 & Tier 3 variables to 3 of Outlet\_Location\_Type Attribute in train data

**R Code:**

table (train [, 10])

outlocationtypetrainencode <- recode (train[,10],"'Tier 1'=1; 'Tier 2'=2; 'Tier 3'=3")

table (outlocationtypetrainencode)

**Output:**

> table(train[,10])

Tier 1 Tier 2 Tier 3

2388 2785 3350

> outlocationtypetrainencode<- recode(train[,10],"'Tier 1'=1; 'Tier 2'=2; 'Tier 3'=3")

> table(outlocationtypetrainencode)

outlocationtypetrainencode

1 2 3

2388 2785 3350

* Recoding the High variables to 1, Medium variables to 2,Small variables to 3 & Empty Values(' ') are replaced by 'NA' of Outlet\_Size Attribute in train data

**R Code:**

table(train[,9])

outsizetrainencode <- recode(train[,9],"'High'=1; 'Medium'=2; 'Small'=3; ''=NA")

table(outsizetrainencode)

**Output:**

> table(train[,9])

High Medium Small

2410 932 2793 2388

> outsizetrainencode <- recode(train[,9],"'High'=1; 'Medium'=2; 'Small'=3; ''=NA")

> table(outsizetrainencode)

outsizetrainencode

1 2 3

932 2793 2388

* Converting the categorical type variables to numeric type.

**R Code:**

table(train[,5])

typetrainnum <- as.numeric(train[,5])

table(typetrainnum)

table(train[,7])

identrainnum <- as.numeric(train[,7])

table(identrainnum)

table(train[,8])

outoldtrainnum<-2018-as.numeric(train[,8])

table(outoldtrainnum)

table(train[,11])

outtypenum <- as.numeric(train[,11])

table(outtypenum)

**Output:**

> table(train[,5])

Baking Goods Breads

648 251

Breakfast Canned

110 649

Dairy Frozen Foods

682 856

Fruits and Vegetables Hard Drinks

1232 214

Health and Hygiene Household

520 910

Meat Others

425 169

Seafood Snack Foods

64 1200

Soft Drinks Starchy Foods

445 148

> typetrainnum <- as.numeric(train[,5])

> table(typetrainnum)

typetrainnum

1 2 3 4 5 6 7 8 9

648 251 110 649 682 856 1232 214 520

10 11 12 13 14 15 16

910 425 169 64 1200 445 148

>

>

>

> table(train[,7])

OUT010 OUT013 OUT017 OUT018 OUT019 OUT027

555 932 926 928 528 935

OUT035 OUT045 OUT046 OUT049

930 929 930 930

> identrainnum <- as.numeric(train[,7])

> table(identrainnum)

identrainnum

1 2 3 4 5 6 7 8 9 10

555 932 926 928 528 935 930 929 930 930

>

>

>

> table(train[,8])

1985 1987 1997 1998 1999 2002 2004 2007 2009

1463 932 930 555 930 929 930 926 928

> outoldtrainnum<-2018-as.numeric(train[,8])

> table(outoldtrainnum)

outoldtrainnum

9 11 14 16 19 20 21 31 33

928 926 930 929 930 555 930 932 1463

>

>

>

> table(train[,11])

Grocery Store Supermarket Type1

1083 5577

Supermarket Type2 Supermarket Type3

928 935

> outtypenum <- as.numeric(train[,11])

> table(outtypenum)

outtypenum

1 2 3 4

1083 5577 928 935

* Combine the modified attributes into a frame

**R Code:**

trainready <- data.frame(cbind(train[,2],fattrainencode,train[,4],typetrainnum,train[,6],identrainnum,outoldtrainnum,outsizetrainencode,outlocationtypetrainencode,outtypenum))

* Finding whether the column contains 'NA' values in train data

**R Code:**

sum(is.na(trainready[,1] ))

sum(is.na(fattrainencode))

sum(is.na(trainready[,3]))

sum(is.na(typetrainnum))

sum(is.na(trainready[,5]))

sum(is.na(identrainnum))

sum(is.na(outoldtrainnum))

sum(is.na(outsizetrainencode))

sum(is.na(outlocationtypetrainencode))

sum(is.na(outtypenum))

**Output:**

> sum(is.na(trainready[,1] ))

[1] 1463

> sum(is.na(fattrainencode))

[1] 0

> sum(is.na(trainready[,3]))

[1] 0

> sum(is.na(typetrainnum))

[1] 0

> sum(is.na(trainready[,5]))

[1] 0

> sum(is.na(identrainnum))

[1] 0

> sum(is.na(outoldtrainnum))

[1] 0

> sum(is.na(outsizetrainencode))

[1] 2410

> sum(is.na(outlocationtypetrainencode))

[1] 0

> sum(is.na(outtypenum))

[1] 0

* Some the values of 1, 8 columns in trainready are not available; they are replaced by mean and median of the attributes respectively

**R Code:**

trainready[is.na(trainready[,1]), 1] <- mean(trainready[,1], na.rm = TRUE)

trainready[is.na(trainready[,8]), 8] <- median(trainready[,8], na.rm = TRUE)

trainreadyfinal<-data.frame(cbind(trainready[,1],fattrainencode,train[,4],typetrainnum,train[,6],identrainnum,outoldtrainnum,trainready[,8],outlocationtypetrainencode,outtypenum))

**Test Data:**

* Recoding the Low Fat variables to 1 & Regular variables to 2 of Item\_Fat\_Content Attribute in test data

**R Code:**

table(test[,3])

fattrain<- test[,3]

library(car)

fattrainencode <- recode(fattrain,"'LF'=1;'low fat'=1;'Low Fat'=1;'reg'=2;'Regular'=2" )

table(fattrainencode)

**Output:**

> table(test[,3])

LF low fat Low Fat reg Regular

206 66 3396 78 1935

> fattrain<- test[,3]

>

>

> #Recoding the Low Fat variables to 1 & Regular variables to 2 of Item\_Fat\_Content Attribute in test data

>

> library(car)

> fattrainencode <- recode(fattrain,"'LF'=1;'low fat'=1;'Low Fat'=1;'reg'=2;'Regular'=2" )

> table(fattrainencode)

fattrainencode

1 2

3668 2013

* Recoding the Tier 1 variables to 1,Tier 2 variables to 2 & Tier 3 variables to 3 of Outlet\_Location\_Type Attribute in test data

**R Code:**

table(test[,10])

outlocationtypetrainencode<- recode(test[,10],"'Tier 1'=1; 'Tier 2'=2; 'Tier 3'=3")

table(outlocationtypetrainencode)

**Output:**

> table(test[,10])

Tier 1 Tier 2 Tier 3

1592 1856 2233

> outlocationtypetrainencode<- recode(test[,10],"'Tier 1'=1; 'Tier 2'=2; 'Tier 3'=3")

> table(outlocationtypetrainencode)

outlocationtypetrainencode

1 2 3

1592 1856 2233

* Recoding the High variables to 1,Medium variables to 2,Small variables to 3 & Empty Values(' ') are replaced by 'NA' of Outlet\_Size Attribute in test data

**R Code:**

table(test[,9])

outsizetrainencode <- recode(test[,9],"'High'=1; 'Medium'=2; 'Small'=3; ''=NA")

table(outsizetrainencode)

**Output:**

> table(test[,9])

High Medium Small

1606 621 1862 1592

> outsizetrainencode <- recode(test[,9],"'High'=1; 'Medium'=2; 'Small'=3; ''=NA")

> table(outsizetrainencode)

outsizetrainencode

1 2 3

621 1862 1592

* Converting the categorical type variables to numeric type

**R Code:**

table(test[,5])

typetrainnum <- as.numeric(test[,5])

table(typetrainnum)

table(test[,8])

outoldtrainnum<-2018-as.numeric(test[,8])

table(outoldtrainnum)

table(test[,7])

identrainnum <- as.numeric(test[,7])

table(identrainnum)

table(test[,11])

outtypenum <- as.numeric(test[,11])

table(outtypenum)

**Output:**

> table(test[,5])

Baking Goods Breads

438 165

Breakfast Canned

76 435

Dairy Frozen Foods

454 570

Fruits and Vegetables Hard Drinks

781 148

Health and Hygiene Household

338 638

Meat Others

311 111

Seafood Snack Foods

25 789

Soft Drinks Starchy Foods

281 121

> typetrainnum <- as.numeric(test[,5])

> table(typetrainnum)

typetrainnum

1 2 3 4 5 6 7 8 9 10 11 12

438 165 76 435 454 570 781 148 338 638 311 111

13 14 15 16

25 789 281 121

>

> table(test[,8])

1985 1987 1997 1998 1999 2002 2004 2007 2009

976 621 620 370 620 619 620 617 618

> outoldtrainnum<-2016-as.numeric(test[,8])

> table(outoldtrainnum)

outoldtrainnum

7 9 12 14 17 18 19 29 31

618 617 620 619 620 370 620 621 976

>

> table(test[,7])

OUT010 OUT013 OUT017 OUT018 OUT019 OUT027

370 621 617 618 352 624

OUT035 OUT045 OUT046 OUT049

620 619 620 620

> identrainnum <- as.numeric(test[,7])

> table(identrainnum)

identrainnum

1 2 3 4 5 6 7 8 9 10

370 621 617 618 352 624 620 619 620 620

>

> table(test[,11])

Grocery Store Supermarket Type1

722 3717

Supermarket Type2 Supermarket Type3

618 624

> outtypenum <- as.numeric(test[,11])

> table(outtypenum)

outtypenum

1 2 3 4

722 3717 618 624

* Combine the modified attributes into a frame

**R Code:**

testready <- data.frame(cbind(test[,2],fattrainencode,test[,4],typetrainnum,test[,6],identrainnum,outoldtrainnum,outsizetrainencode,outlocationtypetrainencode,outtypenum))

* Finding whether the column contains 'NA' values in test data

**R Code:**

sum(is.na(testready[,1] ))

sum(is.na(fattrainencode))

sum(is.na(testready[,3]))

sum(is.na(typetrainnum))

sum(is.na(testready[,5]))

sum(is.na(identrainnum))

sum(is.na(outoldtrainnum))

sum(is.na(outsizetrainencode))

sum(is.na(outlocationtypetrainencode))

sum(is.na(outtypenum))

**Output:**

> sum(is.na(testready[,1] ))

[1] 976

> sum(is.na(fattrainencode))

[1] 0

> sum(is.na(testready[,3]))

[1] 0

> sum(is.na(typetrainnum))

[1] 0

> sum(is.na(testready[,5]))

[1] 0

> sum(is.na(identrainnum))

[1] 0

> sum(is.na(outoldtrainnum))

[1] 0

> sum(is.na(outsizetrainencode))

[1] 1606

> sum(is.na(outlocationtypetrainencode))

[1] 0

> sum(is.na(outtypenum))

[1] 0

* Some the values of 1,8 columns in testready are not available; they are replaced by mean and median of the attributes respectively

**R Code:**

testready[is.na(testready[,1]), 1] <- mean(testready[,1], na.rm = TRUE)

testready[is.na(testready[,8]), 8] <- median(testready[,8], na.rm = TRUE)

testreadyfinal<-data.frame(cbind(testready[,1],fattrainencode,test[,4],typetrainnum,test[,6],identrainnum,outoldtrainnum,testready[,8],outlocationtypetrainencode,outtypenum))

**Data Visualisation:**

**Boxplots:**

**R Code:**

par(mfrow=c(2,2))

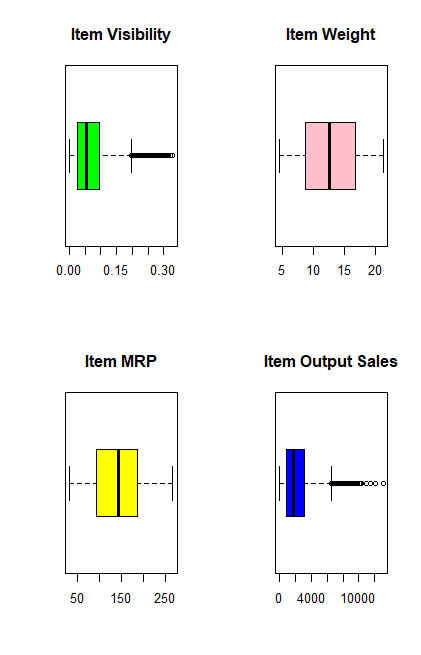
boxplot(train$Item\_Visibility,horizontal = TRUE, main="Item Visibility", col="Green")

boxplot(train$Item\_Weight,horizontal = TRUE, main="Item Weight", col="Pink")

boxplot(train$Item\_MRP,horizontal = TRUE, main="Item MRP", col="Yellow")

boxplot(train$Item\_Outlet\_Sales,horizontal = TRUE, main="Item Output Sales", col="Blue")

**Output:**



**Histograms:**

**R Code:**

par(mfrow=c(2,2))

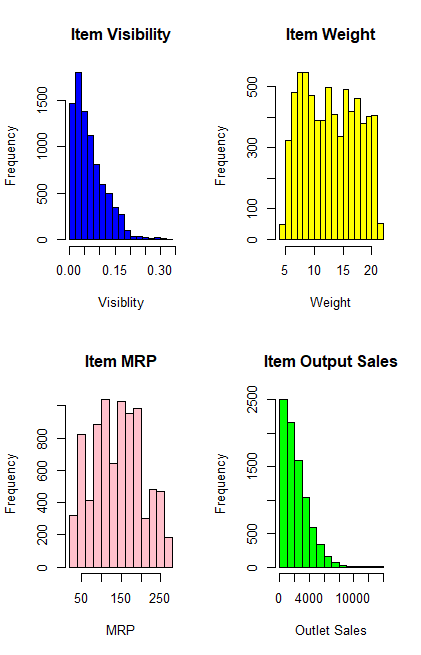
hist(train$Item\_Visibility, main="Item Visibility", col="Blue", xlab = "Visiblity" )

hist(train$Item\_Weight, main="Item Weight", col="Yellow", xlab = "Weight")

hist(train$Item\_MRP, main="Item MRP", col="Pink", xlab = "MRP")

hist(train$Item\_Outlet\_Sales, main="Item Output Sales", col="Green", xlab = "Outlet Sales")

**Output:**

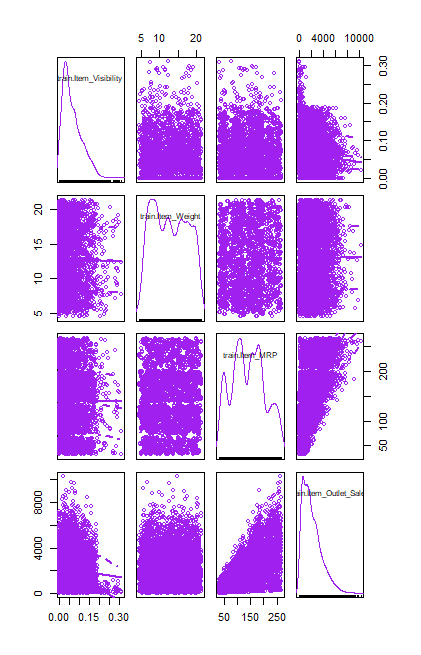


**Scatter Plot:**

**R Code:**

scatterplotMatrix(~train$Item\_Visibility+train$Item\_Weight+train$Item\_MRP+train$Item\_Outlet\_Sales,col="Purple")

**Output:**



From above plots we can observe that Item\_Output\_Sales, Item\_Visibility are skewed curves towards right. Item\_Weight & Item\_MRP are most probably following normal distribution.

**Correlation Matrix:**

**R Code:**

Corr\_Matrix <- train[,c(2,4,6,8,12)]

cor(Corr\_Matrix)

**Output:**

Item\_Weight

Item\_Weight 1

Item\_Visibility NA

Item\_MRP NA

Outlet\_Establishment\_Year NA

Item\_Outlet\_Sales NA

Item\_Visibility

Item\_Weight NA

Item\_Visibility 1.000000000

Item\_MRP -0.001314848

Outlet\_Establishment\_Year -0.074833504

Item\_Outlet\_Sales -0.128624612

Item\_MRP

Item\_Weight NA

Item\_Visibility -0.001314848

Item\_MRP 1.000000000

Outlet\_Establishment\_Year 0.005019916

Item\_Outlet\_Sales 0.567574447

Outlet\_Establishment\_Year

Item\_Weight NA

Item\_Visibility -0.074833504

Item\_MRP 0.005019916

Outlet\_Establishment\_Year 1.000000000

Item\_Outlet\_Sales -0.049134970

Item\_Outlet\_Sales

Item\_Weight NA

Item\_Visibility -0.12862461

Item\_MRP 0.56757445

Outlet\_Establishment\_Year -0.04913497

Item\_Outlet\_Sales 1.00000000

library(corrplot)

corrplot(corr=cor(Corr\_Matrix),method="ellipse")

**Data Modelling:**

**Random Forest Classification:**

**R Code:**

library(randomForest)

modelrf<-randomForest(trainreadyfinal,train[,12],importance=TRUE, ntree=100)

Item\_Outlet\_Sales<-predict(modelrf,testreadyfinal)

soln<-cbind(test[,'Item\_Identifier'],test[,'Outlet\_Identifier'],data.frame(Item\_Outlet\_Sales))

write.csv(soln,file="C:/Users/PhaniRohitha Kaza/Desktop/BigMart/SampleSubmission.csv")

**Decision Tree Classification:**

**R Code:**

library(caret)

library(onehot)

library(ggplot2)

library(Metrics)

library(tree)

library(rpart)

library(h2o)

dtc = rpart.control(maxdepth = 4,minsplit = 20,minbucket = 7)

dt = rpart(train[,12]~. , data = trainreadyfinal,parms = c(split = "gini"),control = dtc)

Item\_Outlet\_Sales = predict(dt,newdata = testreadyfinal)

soln<-cbind(test[,'Item\_Identifier'],test[,'Outlet\_Identifier'],data.frame(Item\_Outlet\_Sales))

write.csv(soln,file="C:/Users/PhaniRohitha Kaza/Desktop/BigMart/SampleSubmission1.csv")

dtc = rpart.control(maxdepth = 4,minsplit = 20,minbucket = 7)

dt = rpart(train[,12]~. , data = trainreadyfinal,parms = c(split = "entropy"),control = dtc)

Item\_Outlet\_Sales = predict(dt,newdata = testreadyfinal)

soln<-cbind(test[,'Item\_Identifier'],test[,'Outlet\_Identifier'],data.frame(Item\_Outlet\_Sales))

write.csv(soln,file="C:/Users/PhaniRohitha Kaza/Desktop/BigMart/SampleSubmission3.csv")

**Linear Regression with multiple variables:**

**R Code:**

lr =lm(train[,12]~ . ,data =trainreadyfinal)

Item\_Outlet\_Sales <- predict(lr,newdata = testreadyfinal)

soln<-cbind(test[,'Item\_Identifier'],test[,'Outlet\_Identifier'],data.frame(Item\_Outlet\_Sales))

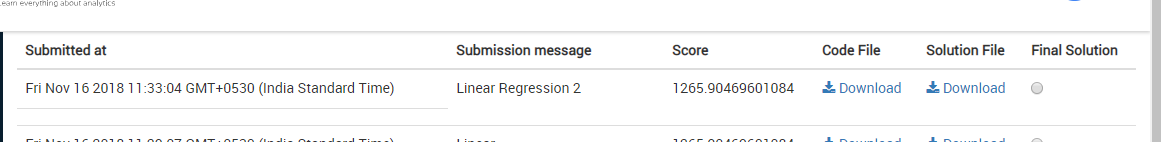
write.csv(soln,file="C:/Users/PhaniRohitha Kaza/Desktop/BigMart/SampleSubmission2.csv")

**Conclusion:**

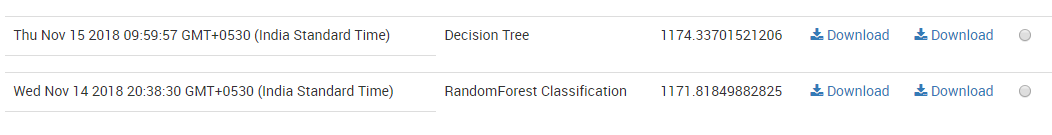
Score predicted using three different Classifiers:

Minimum Score of the competition is 1131.3755

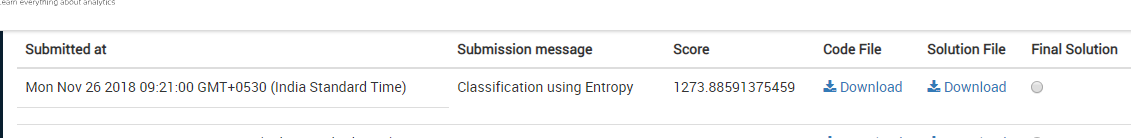
**Linear Regression:**

****

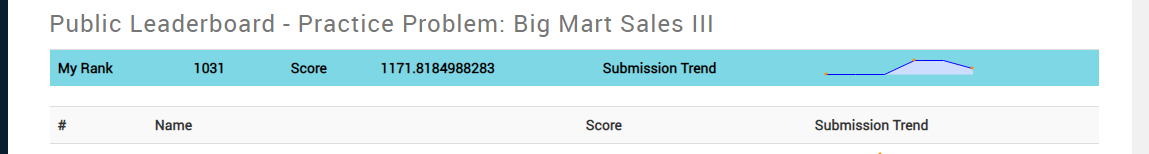
**Decision Tree & Random Forest:**



**Decision Tree using Entropy:**

****

**My Rank on Leader Board:**

****

**K. Phani Rohitha**

**Roll No.:411634**