25 October 2018
16:46

Problem Statement

The Researcher team 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.

Please note that the data may have missing values as some stores might not report all the data due to technical glitches. Hence, it will be required to treat them accordingly.

Data

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	The category to which the 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	The size of the store in terms of ground area covered
Outlet_Location_Type	The type of city in which the store is located
Outlet_Type	Whether the outlet is just a grocery store or some sort of supermarket
Item_Outlet_Sales	Sales of the product in the particular store. This is the outcome variable to be predicted.

Evaluation Metric:

Your model performance will be evaluated on the basis of your prediction of the sales for the test data (test.csv), which contains similar data-points as train except for the sales to be predicted. Your submission needs to be in the format as shown in "SampleSubmission.csv". We at our end, have the actual sales for the test dataset, against which your predictions will be evaluated. We will use the Root Mean Square Error value to judge your response.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(Predicted_i - Actual_i)^2}{N}}$$

Where,
N: total number of observations
Predicted: the response entered by user
Actual: actual values of sales
Also, note that the test data is further divided into Public (25%) and Private (75%) data. Your initial responses will be checked and scored on the Public data. But, the final rankings will be based on score on Private data set.

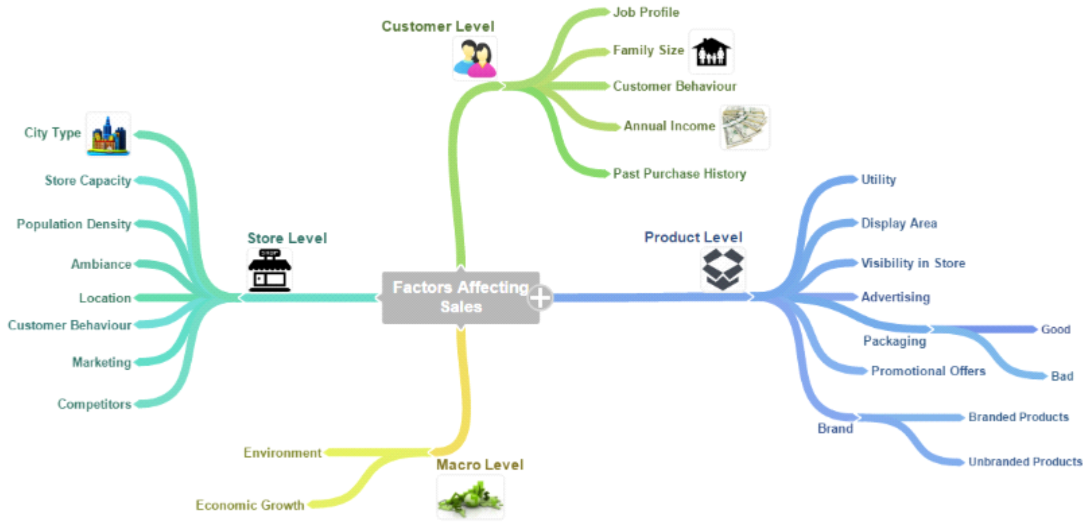
Hypothesis Generation

What is hypothesis generation?

This is a very important stage in any machine learning process. It involves understanding the problem in detail by brainstorming as many factors as possible which can impact the outcome. It is done by understanding the problem statement thoroughly and before looking at the data.

How to do hypothesis generation?

One very effective technique to generate hypotheses is by creating mindmaps. You can draw it even using a pen and paper. The general methodology is as follows: Write the main idea in the center. Draw branches from the center such they are connected with one another with final outputs shown towards the end.
Below is a simple mind map. Let’s understand it.



We can start the process by working on four levels: Store Level, Product Level, Customer Level and Macro Level.

Store Level Hypotheses

1. **City type**: Stores located in urban or Tier 1 cities should have higher sales because of the higher income levels of people there.

2. **Population Density**: Stores located in densely populated areas should have higher sales because of more demand. Store Capacity: Stores which are very big in size should have higher sales as they act like one-stop-shops and people would prefer getting everything from one place

3. **Competitors**: Stores having similar establishments nearby should have less sales because of more competition.

4. **Marketing**: Stores which have a good marketing division should have higher sales as it will be able to attract customers through the right offers and advertising.

5. **Location**: Stores located within popular marketplaces should have higher sales because of better access to customers.

6. **Ambiance**: Stores which are well-maintained and managed by polite and humble people are expected to have higher footfall and thus higher sales.

Product Level Hypotheses

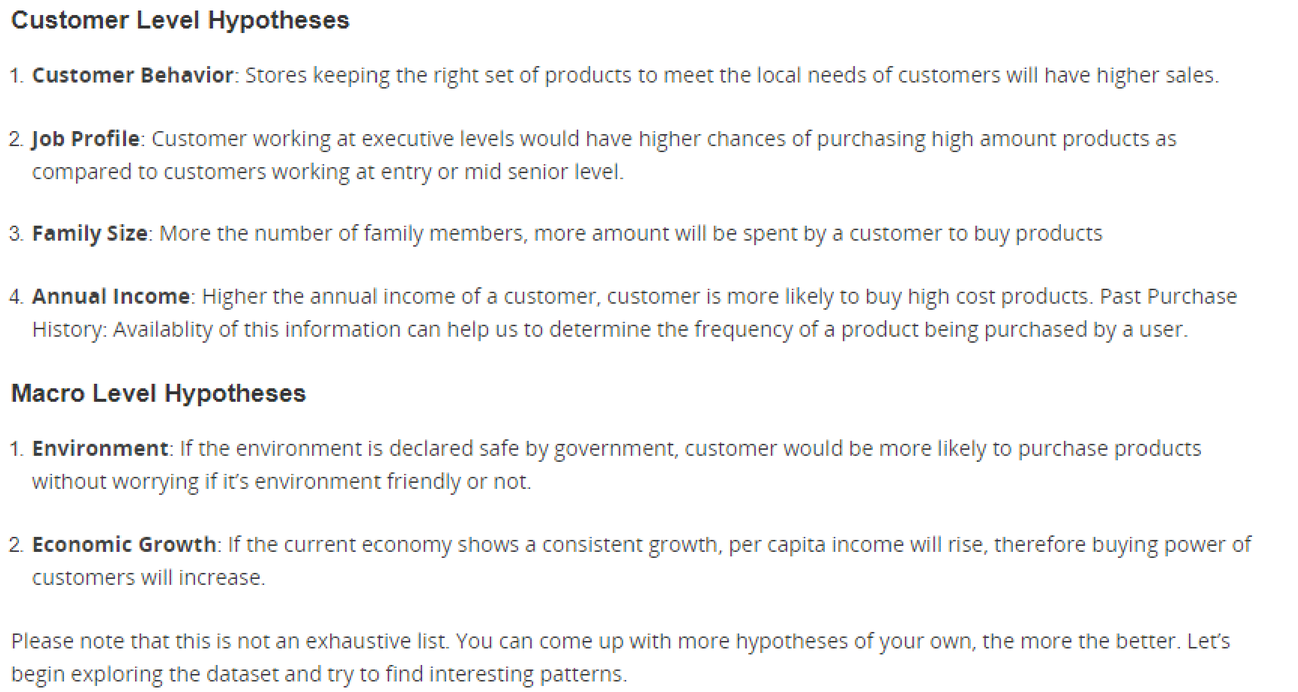
1. **Brand**: Branded products should have higher sales because of higher trust in the customer.

2. **Packaging**: Products with good packaging can attract customers and sell more.

3. **Utility**: Daily use products should have a higher tendency to sell as compared to the specific use products.

4. **Display Area**: Products which are given bigger shelves in the store are likely to catch attention first and sell more. Visibility in Store: The location of product in a store will impact sales. Ones which are right at entrance will catch the eye of customer first rather than the ones in back.

5. **Advertising**: Better advertising of products in the store will should higher sales in most cases. Promotional Offers: Products accompanied with attractive offers and discounts will sell more.



Customer Level Hypotheses

1. **Customer Behavior**: Stores keeping the right set of products to meet the local needs of customers will have higher sales.

2. **Job Profile**: Customer working at executive levels would have higher chances of purchasing high amount products as compared to customers working at entry or mid senior level.

3. **Family Size**: More the number of family members, more amount will be spent by a customer to buy products

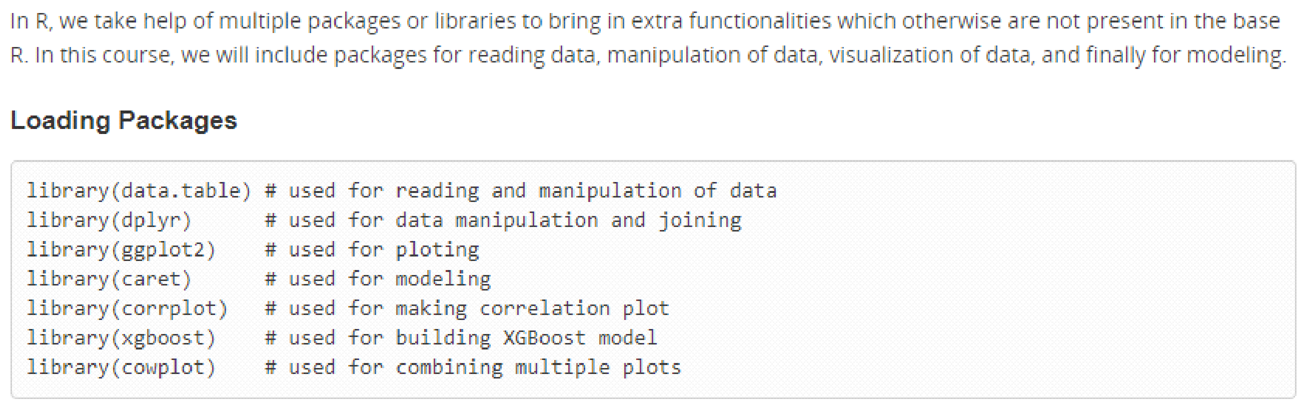
4. **Annual Income**: Higher the annual income of a customer, customer is more likely to buy high cost products. Past Purchase History: Availablity of this information can help us to determine the frequency of a product being purchased by a user.

Macro Level Hypotheses

1. **Environment**: If the environment is declared safe by government, customer would be more likely to purchase products without worrying if it's environment friendly or not.

2. **Economic Growth**: If the current economy shows a consistent growth, per capita income will rise, therefore buying power of customers will increase.

Please note that this is not an exhaustive list. You can come up with more hypotheses of your own, the more the better. Let's begin exploring the dataset and try to find interesting patterns.



In R, we take help of multiple packages or libraries to bring in extra functionalities which otherwise are not present in the base R. In this course, we will include packages for reading data, manipulation of data, visualization of data, and finally for modeling.

Loading Packages

```
library(data.table) # used for reading and manipulation of data
library(dplyr)      # used for data manipulation and joining
library(ggplot2)    # used for ploting
library(caret)      # used for modeling
library(corrplot)   # used for making correlation plot
library(xgboost)    # used for building XGBoost model
library(cowplot)    # used for combining multiple plots
```

Reading Data

- The Train file contains 11 independent variables and 1 target variable, i.e., *Item_Outlet_Sales*.
- The Test file also contains the same set of independent variables, but there is no target variable because that is what we have to predict.
- The Sample Submissions file contains the format in which we have to submit our predictions.

We will use fread() function of data.table package to read the datasets.

```
train = fread("Train_UWu5bXk.csv")
test = fread("Test_u94Q5KV.csv")
submission = fread("SampleSubmission_TmnO39y.csv")
```

Initially we should understand our raw data thoroughly, i.e., we should explore the no. of features/columns and rows, datatype of the features, feature names and so on. It helps in working with the data in the next stages.

Dimensions of Data

Let's quicky check the dimensions of our data, i.e., columns and rows.

```
dim(train);dim(test)
```

```
[1] 8523   12
[1] 5681   11
```

train dataset has 8523 rows and 12 features and test has 5681 rows and 11 columns. train has 1 extra column which is the target variable. We will predict this target variable for the test dataset later in this tutorial.

Features of Data

We will take a quick glance over the feature names of train and test datasets.

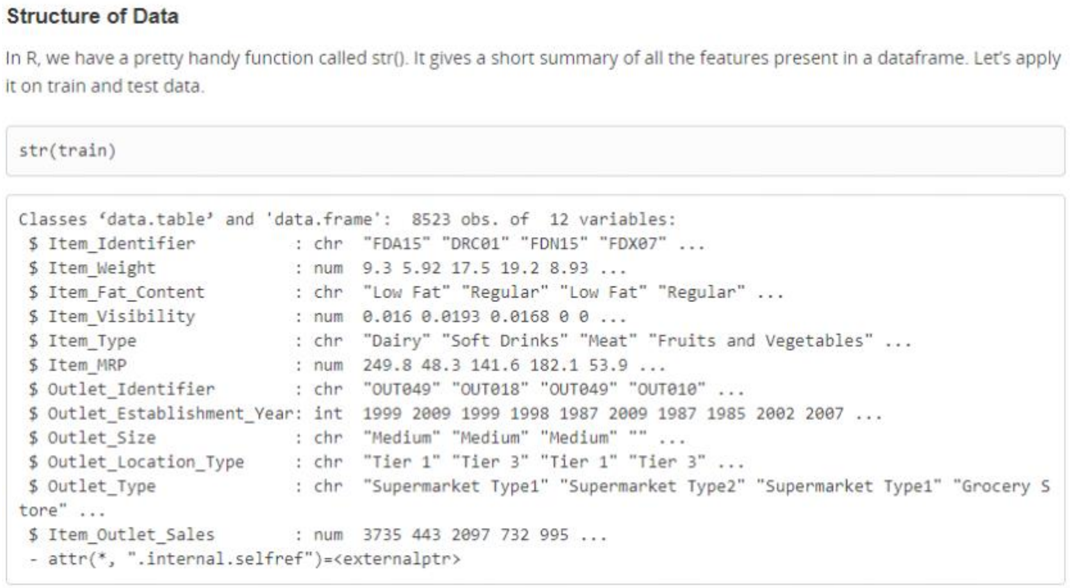
```
names(train)
```

```
 [1] "Item_Identifier"           "Item_Weight"               "Item_Fat_Content"          "Item_Visibilit
y"
 [5] "Item_Type"                 "Item_MRP"                  "Outlet_Identifier"         "Outlet_Establis
hment_Year"
 [9] "Outlet_Size"               "Outlet_Location_Type"      "Outlet_Type"               "Item_Outlet_Sal
es"
```

```
names(test)
```

```
 [1] "Item_Identifier"           "Item_Weight"               "Item_Fat_Content"          "Item_Visibilit
y"
 [5] "Item_Type"                 "Item_MRP"                  "Outlet_Identifier"         "Outlet_Establis
hment_Year"
 [9] "Outlet_Size"               "Outlet_Location_Type"      "Outlet_Type"
```

Item_Outlet_Sales is present in train but not in test dataset because this is the target variable that we have to predict.

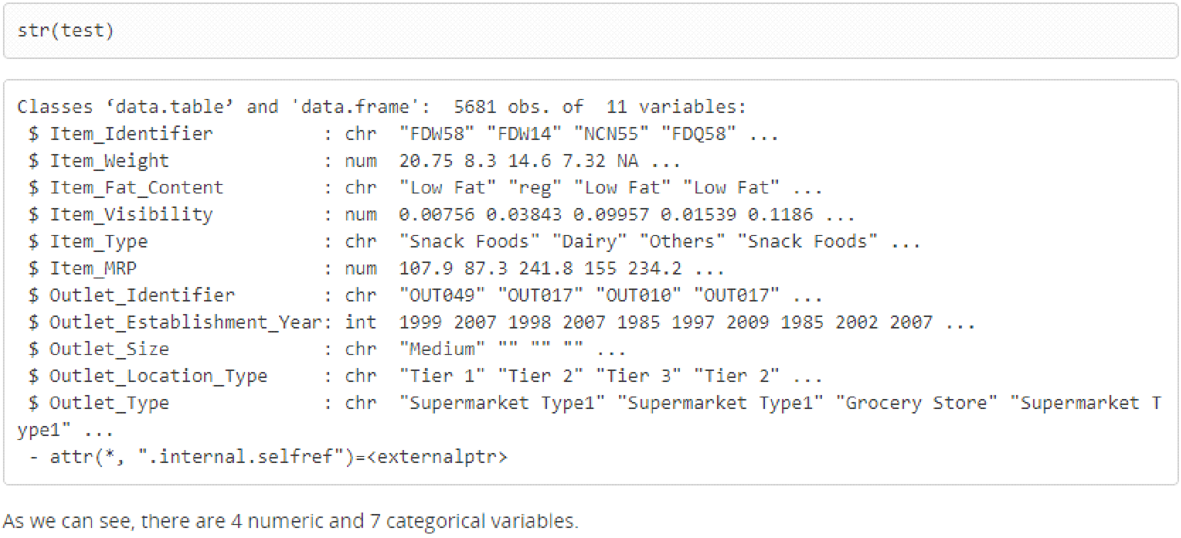


Structure of Data

In R, we have a pretty handy function called str(). It gives a short summary of all the features present in a dataframe. Let's apply it on train and test data.

```
str(train)
```

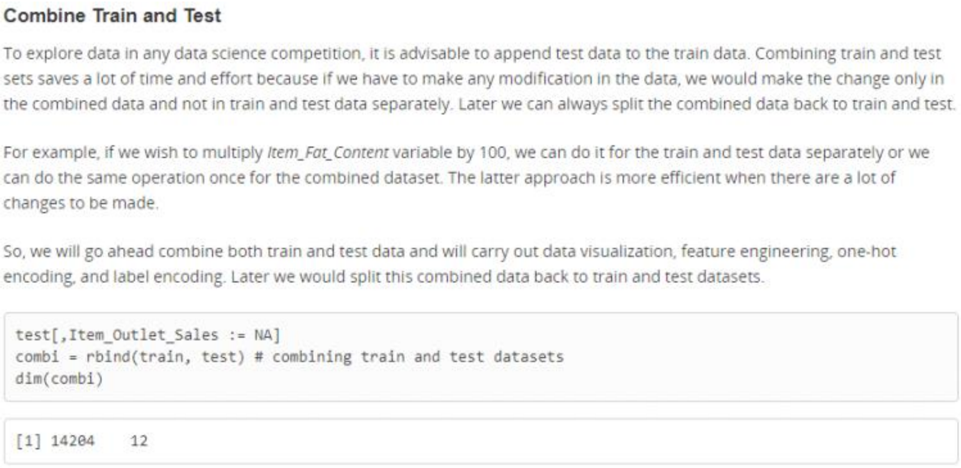
```
Classes ‘data.table’ and 'data.frame':  8523 obs. of  12 variables:
 $ Item_Identifier          : chr  "FDA15" "DRC01" "FDN15" "FDX07" ...
 $ Item_Weight              : num  9.3 5.92 17.5 19.2 8.93 ...
 $ Item_Fat_Content         : chr  "Low Fat" "Regular" "Low Fat" "Regular" ...
 $ Item_Visibility          : num  0.016 0.0193 0.0168 0 0 ...
 $ Item_Type                : chr  "Dairy" "Soft Drinks" "Meat" "Fruits and Vegetables" ...
 $ Item_MRP                 : num  249.8 48.3 141.6 182.1 53.9 ...
 $ Outlet_Identifier        : chr  "OUT049" "OUT018" "OUT049" "OUT010" ...
 $ Outlet_Establishment_Year: int  1999 2009 1999 1998 1987 2009 1987 1985 2002 2007 ...
 $ Outlet_Size              : chr  "Medium" "Medium" "Medium" "" ...
 $ Outlet_Location_Type     : chr  "Tier 1" "Tier 3" "Tier 1" "Tier 3" ...
 $ Outlet_Type              : chr  "Supermarket Type1" "Supermarket Type2" "Supermarket Type1" "Grocery S
tore" ...
 $ Item_Outlet_Sales        : num  3735 443 2097 732 995 ...
 - attr(*, ".internal.selfref")=<externalptr>
```



```
str(test)
```

```
Classes ‘data.table’ and 'data.frame':  5681 obs. of  11 variables:
 $ Item_Identifier          : chr  "FDW58" "FDW14" "NCN55" "FDQ58" ...
 $ Item_Weight              : num  20.75 8.3 14.6 7.32 NA ...
 $ Item_Fat_Content         : chr  "Low Fat" "reg" "Low Fat" "Low Fat" ...
 $ Item_Visibility          : num  0.00756 0.03843 0.09957 0.01539 0.1186 ...
 $ Item_Type                : chr  "Snack Foods" "Dairy" "Others" "Snack Foods" ...
 $ Item_MRP                 : num  107.9 87.3 241.8 155 234.2 ...
 $ Outlet_Identifier        : chr  "OUT049" "OUT017" "OUT010" "OUT017" ...
 $ Outlet_Establishment_Year: int  1999 2007 1998 2007 1985 1997 2009 1985 2002 2007 ...
 $ Outlet_Size              : chr  "Medium" "" "" "" ...
 $ Outlet_Location_Type     : chr  "Tier 1" "Tier 2" "Tier 3" "Tier 2" ...
 $ Outlet_Type              : chr  "Supermarket Type1" "Supermarket Type1" "Grocery Store" "Supermarket T
ype1" ...
 - attr(*, ".internal.selfref")=<externalptr>
```

As we can see, there are 4 numeric and 7 categorical variables.



Combine Train and Test

To explore data in any data science competition, it is advisable to append test data to the train data. Combining train and test sets saves a lot of time and effort because if we have to make any modification in the data, we would make the change only in the combined data and not in train and test data separately. Later we can always split the combined data back to train and test.

For example, if we wish to multiply *Item_Fat_Content* variable by 100, we can do it for the train and test data separately or we can do the same operation once for the combined dataset. The latter approach is more efficient when there are a lot of changes to be made.

So, we will go ahead combine both train and test data and will carry out data visualization, feature engineering, one-hot encoding, and label encoding. Later we would split this combined data back to train and test datasets.

```
test[,Item_Outlet_Sales := NA]
combi = rbind(train, test) # combining train and test datasets
dim(combi)
```

```
[1] 14204    12
```