**NOTE: I have used two datasets in this assignment. The reason for choosing different datasets is to cover as many as concepts of data mining.**

**Dataset 1 (big-mart-sales) source:** <https://www.kaggle.com/brijbhushannanda1979/bigmart-sales-data>

**Dataset 2 (startup-success-prediction) source:** <https://www.kaggle.com/manishkc06/startup-success-prediction>

**Whole code used for this project and other resources:** ( <https://github.com/ZeeWING-Projects/Data-Mining-Operations> )

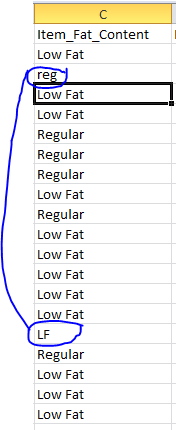
**Tools used:** Spider IDE (For using python)

# 1. Applying some pre-processing steps (Data set 1):

First we need to apply some pre-processing techniques before we process it.

## 1.1 Matching field’s values:

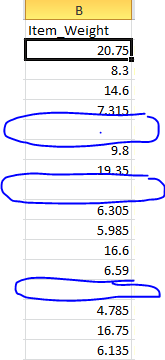
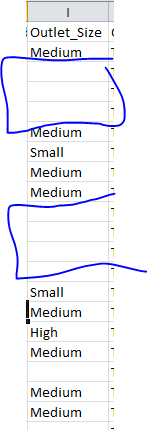
There are few fields in dataset which contain same data with different names, so we need to make the same. For example we have attribute item\_Fat\_Content which contain two labels, Low Fat and Regular, but for representing this same value “LF” and “reg” are used so we need to remove these shortcuts.



## Filling out missing values:

In our dataset we have a bunch of attributes having missing values. And we have to fill them by using well known pre-processing techniques. For example for numeric attribute we have methods like by using median, mean and mode and for ordinal attributes we will use some built in functions of python (which use this same mode technique).

For example we have some attributes with missing values.

## Implementation

So we have a real dataset having some missing values and some other issues so we need to perform above stated steps of pre-processing

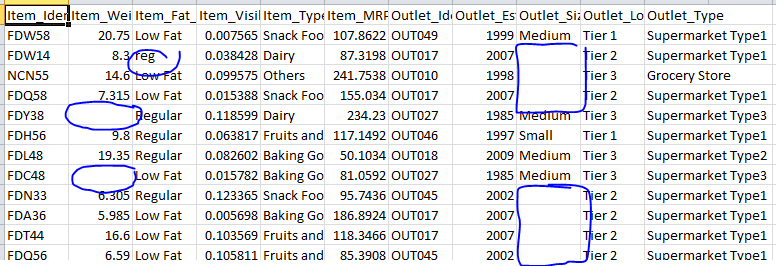
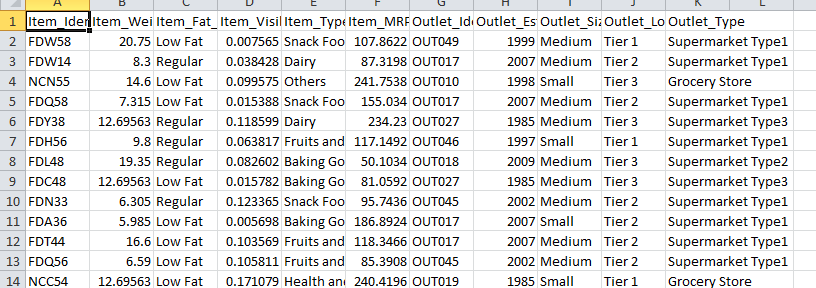
Realdataset:(<https://github.com/ZeeWING-Projects/Data-Mining-Operations/blob/main/RealDataSet/RealDataSet.csv> )

Pre-processed-dataset:(<https://github.com/ZeeWING-Projects/Data-Mining-Operations/tree/main/Pre-processed%20DataSet> )

Pre-processing-code:(<https://github.com/ZeeWING-Projects/Data-Mining-Operations/blob/main/Preprocessing-Code/Preprocessing.py>)



Dataset comparison

Now our dataset is ready to perform other operations.

# Data Visualization

For understating the data graphically we can use different types of graphs. For this dataset analysis we will use some well know graphs to analyze few features of dataset.

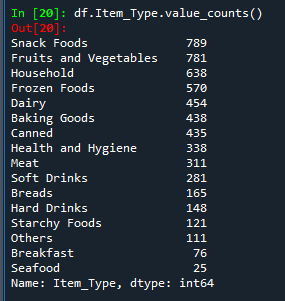
## 2.1 Box-Plot

Def: A Box Plot is also known as Whisker plot is created to display the summary of the set of data values having properties like minimum, first quartile, median, third quartile and maximum. In the box plot, a box is created from the first quartile to the third quartile; a vertical line is also there which goes through the box at the median. Here x-axis denotes the data to be plotted while the y-axis shows the frequency distribution. (geeksforgeeks). So this is used for quick summary of data. we have following points to observe by using this box-plot.

* Detect outlier values.
* Mean tendency of values.
* Symmetry of data.

Assume for example we are plotting item\_weight of different types of items (item\_type)

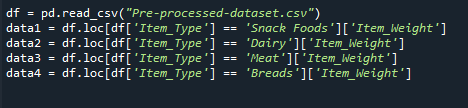
For this we need item weights of each individual item\_type. we have following types of unique items



For example we are taking 4 items type.

1. Snack foods
2. Dairy
3. Meat
4. Breads

Now we need to get separate data of each item



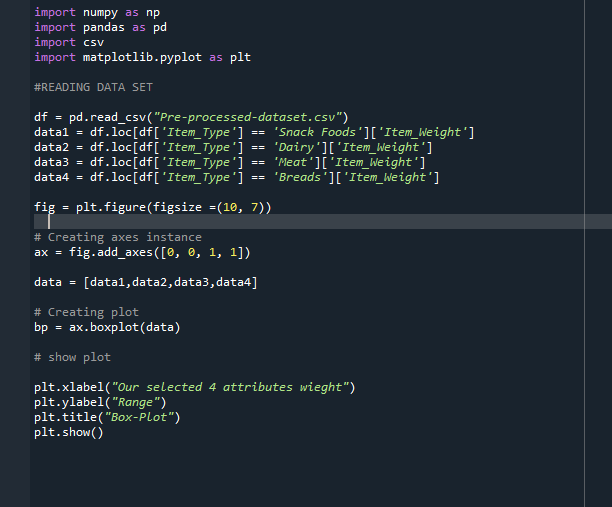
Now we will use this data to plot box-plot

### 2.1.1 Analysis of box-plot

1. Snack foods:
   1. Average weight of this item is around 12.5 or a bit greater.
   2. Its max average value is around 15.5 (approximately)
   3. Its min average value is around 9.7 (approximately)
   4. its symmetric line shows that there is no skewness, which means no there are approximately equal number of values greater than mean and less than mean.
   5. There is no out liar value.
2. Dairy:
   1. Average weight of this item is around 12.
   2. Its max average value is around 16.25 (approximately)
   3. Its min average value is around 10 (approximately)
   4. its symmetric line shows that there is right skewned because it is toward lower half, the **number of items is greater** whose value is less than than mean.
   5. There is no out liar value.
3. Meat:
   1. Average weight of this item is around 12.5 or a bit greater.
   2. Its max average value is around 15.5 (approximately)
   3. Its min average value is around 9.8 (approximately)
   4. its symmetric line shows that there is no skewness, which means no there are approximately equal number of values greater than mean and less than mean.
   5. There is no out liar value.
4. Breads:
   1. Average weight of this item is around 11.25 or a bit greater.
   2. Its max average value is around 13.25 (approximately)
   3. Its min average value is around 6.5 (approximately)
   4. its symmetric line shows that there is left skewned because it is toward upper half, the **number of items is greater** whose value is greater than mean.
   5. There is no out liar value.

### 2.1.1 Implementation box-plot

Code for box-plot : (<https://github.com/ZeeWING-Projects/Data-Mining-Operations/blob/main/Graphs-code/Graph-ploting-box-plot.py>)



2.2