**BigMart Sales Prediction**

**Big-Data Systems and Intelligence Analytics**

**INFO 7245 - SPRING 2018**

PROFESSOR:

**Nicholas Brown**

TEAM MEMBERS:

**TEJESH RATHI (**<https://github.com/Tejesh0711>**)**

(<https://linkedin.com/in/tejesh-rathi>)

Table of Contents

[Problem Statement 3](#_Toc481448732)

[Part 1: Data wrangling and exploratory data analysis 5](#_Toc481448733)

[Data cleaning 3](#_Toc481448735)

Data Preprocessing………………………………………………………………………………………………………………………5

Exploratory data analysis in Python……………………………………………………………………………………………….7

[Exploratory data analysis in Tableau 9](#_Toc481448736)

[Part 2: Building and Evaluating Models 25](#_Toc481448738)

Feature Selection…………………………………………………………………………………………………………………………12

[Model Building and Prediction 1](#_Toc481448739)3

[References 1](#_Toc481448740)3

# **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.

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 particulat store. This is the outcome variable to be predicted. |

<https://www.kaggle.com/devashish0507/big-mart-sales-prediction>

# **Part 1: Data wrangling and exploratory data analysis**

In this section, we will perform the data the following operations:

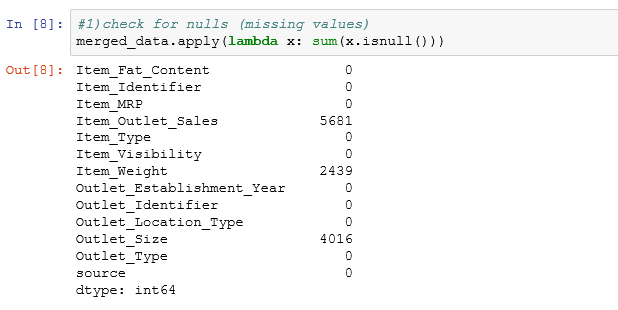
* Data cleaning
* Exploratory data analysis in Python
* Exploratory data analysis in Tableau

As per the problem statement, we have two files, train csv to train our model and test csv to test our model and predict the sales outcome. But, before we begin to clean our data or impute new features and values, we combined both the files onto one so that we don’t have to clean the data twice differently.

## **Data cleaning**

This section comprises of handling missing data. We get the concatenated .CSV file from the previous part, on which we perform data cleaning steps.

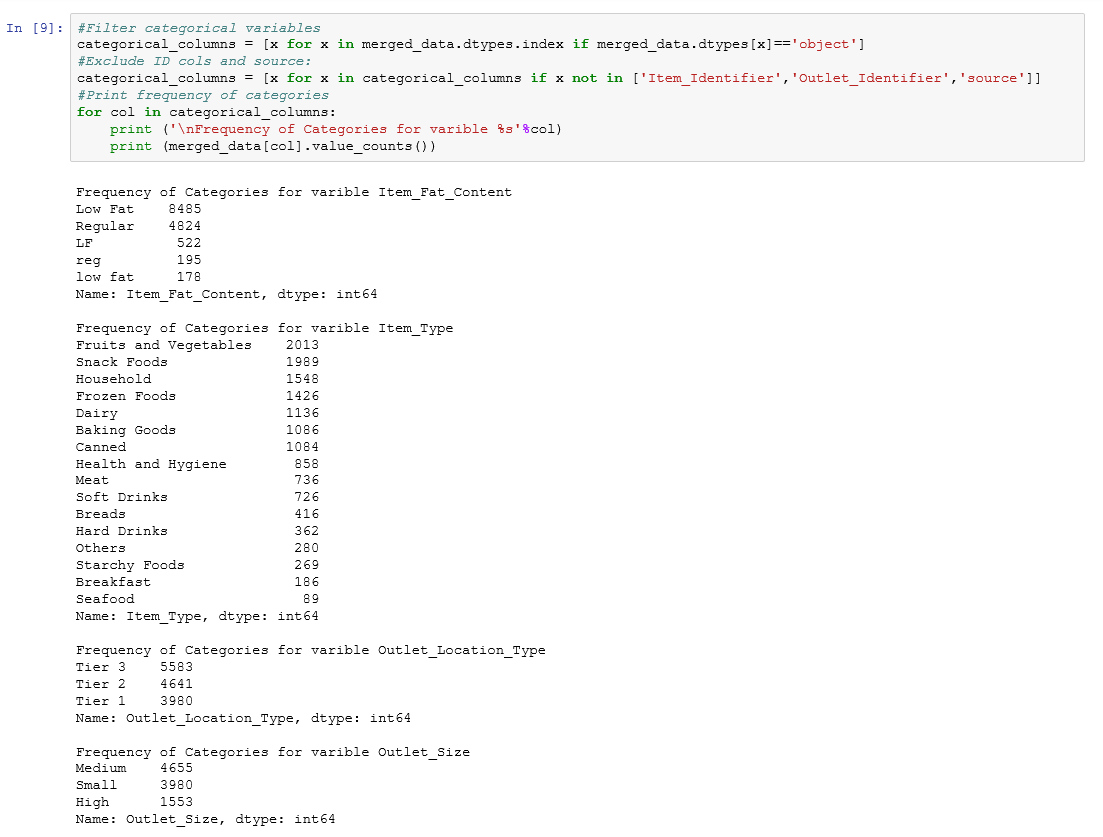
I first analyzed all the columns having empty values as shown below:



I have implemented this by writing a lambda function, which gives us a .CSV file having the count of null values for each variable.

I will clean these null values in the upcoming section

Prior to cleaning, just to understand how the data is flowing, so I got a count of all the different categories in all the variables.



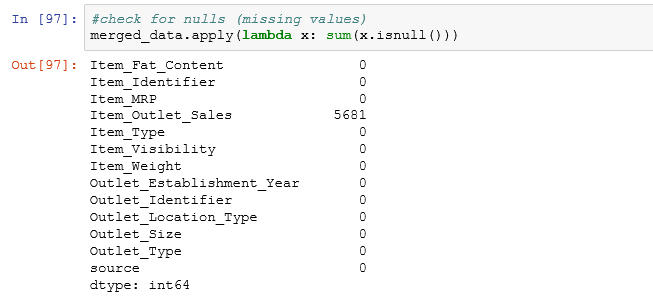
**Columns Updated and cleaned:**

* **Item\_Weight** – Ideally, a weight of an item can never be zero. So, replacing all the nulls with the average weight of a particular item across all outlets.



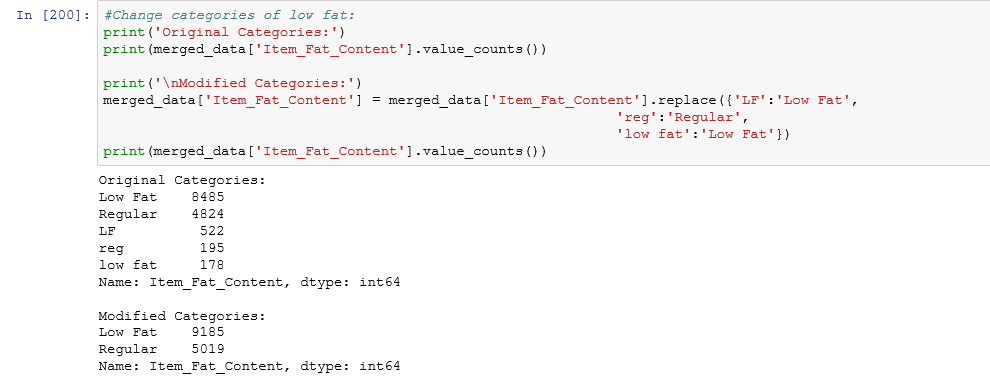
* **Outlet\_Size** –
* **Item\_Outlet\_Sales** – We need not clean these, as these are nulls from the test file, as we need to predict these values based on our model

We check for nulls again after cleaning and below are the results:

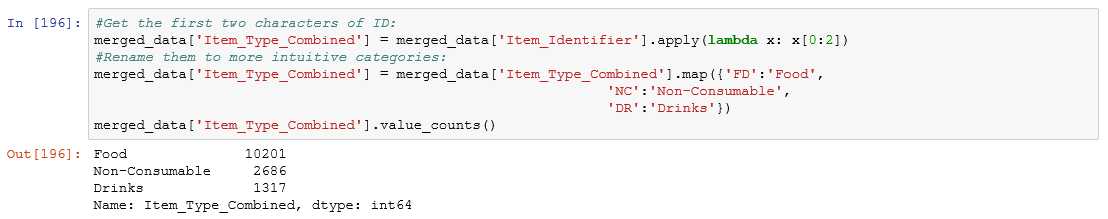


**Data pre-processing**

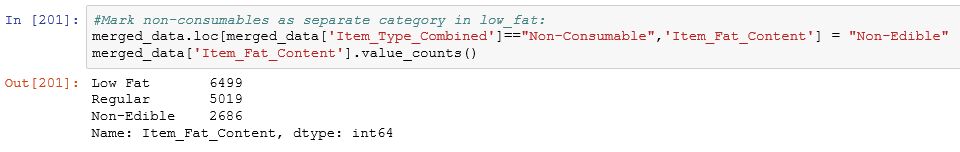
* On observing, I found out that there are 5 different categories present, out of which 3 are similar and all combined falls under two categories in the data. So treated this bad data and combined the groups together as below:



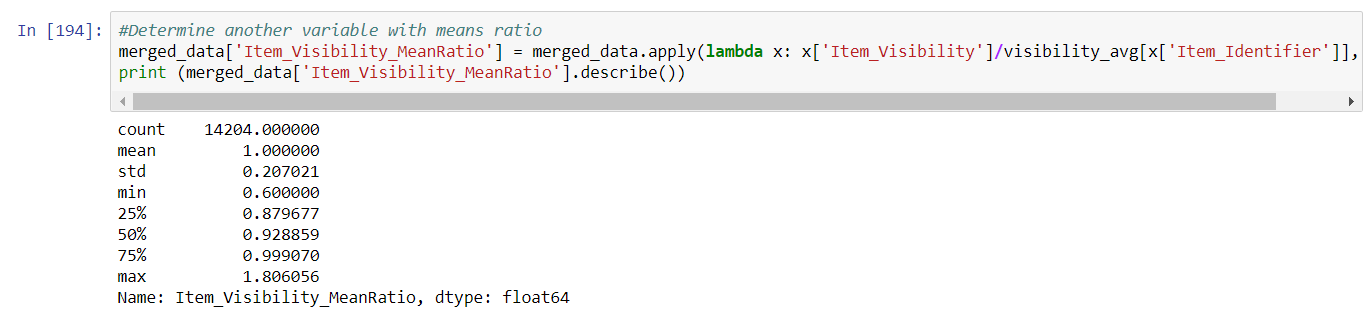
* The data dictionary says there are three types of items – Food items, Non-consumable and Drinks.



* The data says one category as Non consumable, but still filters it using Regular and Nonfat. Since, some are non-consumable, it doesn’t fall under any fat/nonfat category, so replacing all non-consumable items as “Non-edible”



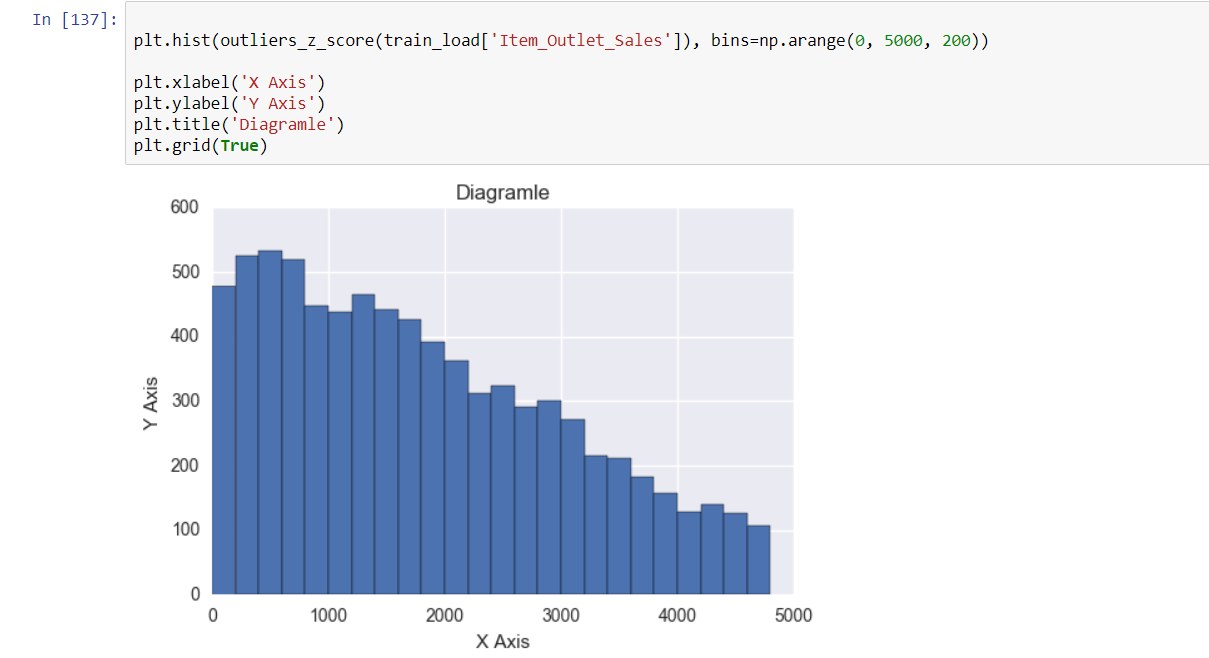
* Determine visibility ratio. (Standardize the visibility ratio)



**Exploratory data analysis in Python**

**Analysis 1:** The distribution of sales data

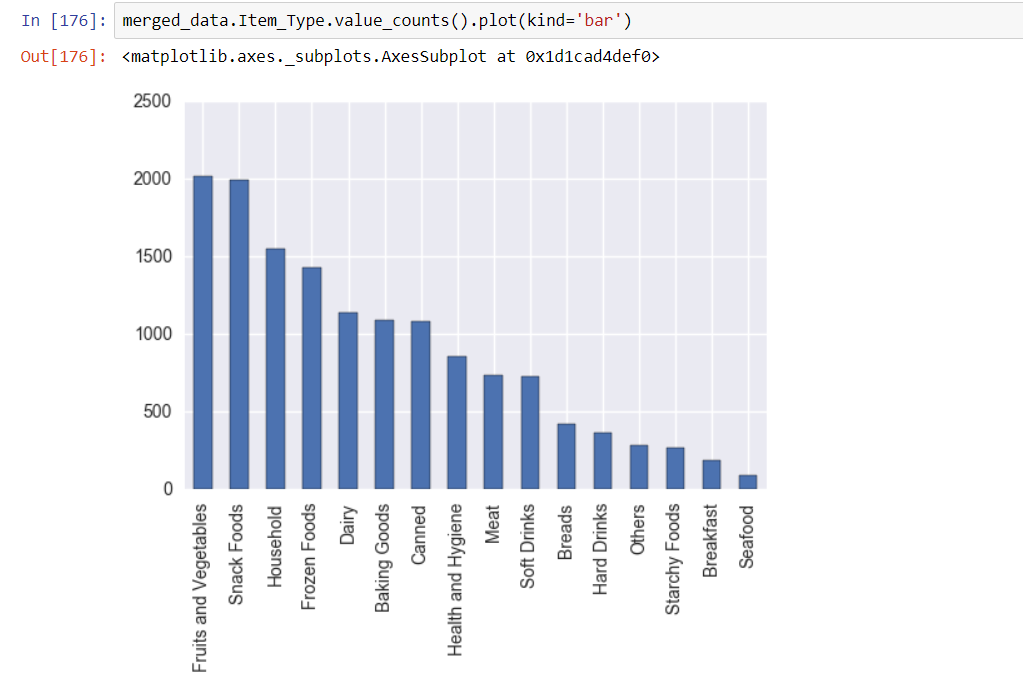
**Approach:**



**Conclusion:** The data is left skewed based on the total sum of sales

**Analysis 2:** Sum of sales based on different categories of Item\_Type

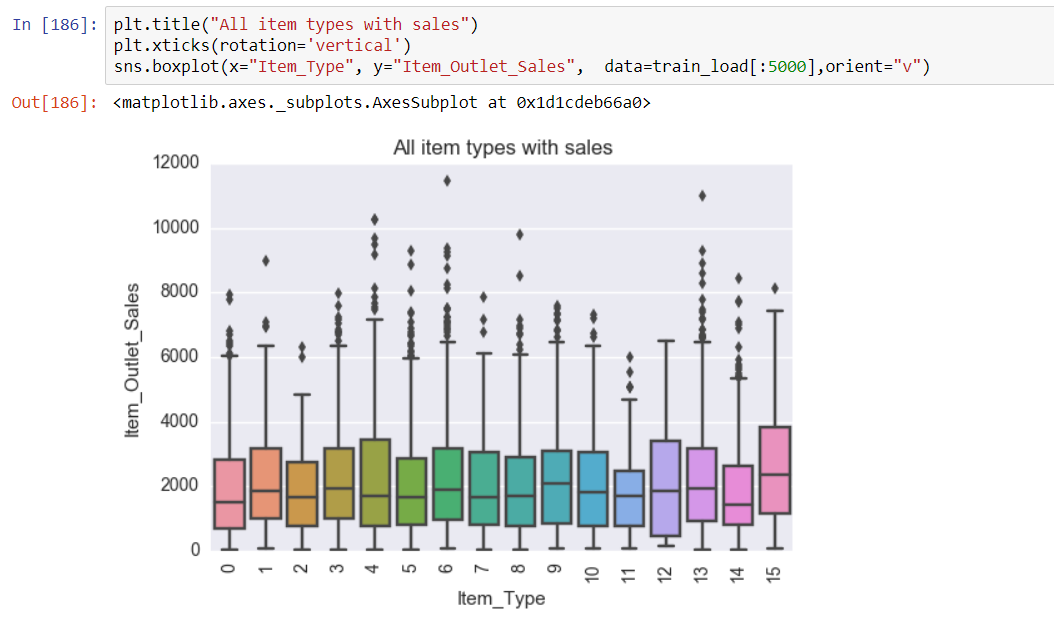
**Approach:**



**Analysis 3:** Box plot view for all sales items

**Approach:**

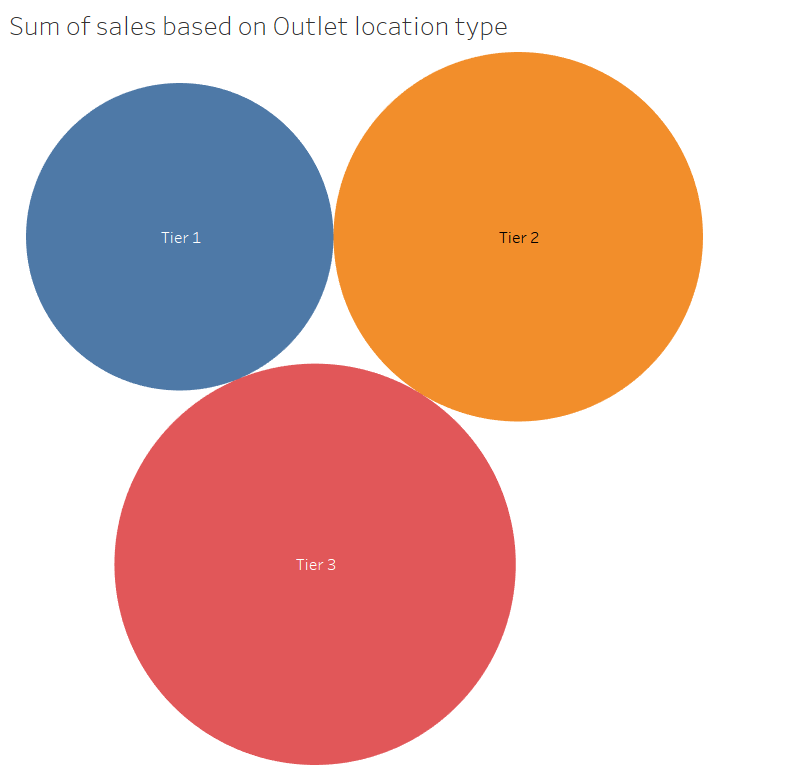
The dots above each categorical box plots shows the outliers and this has skewed the data toward the left side.



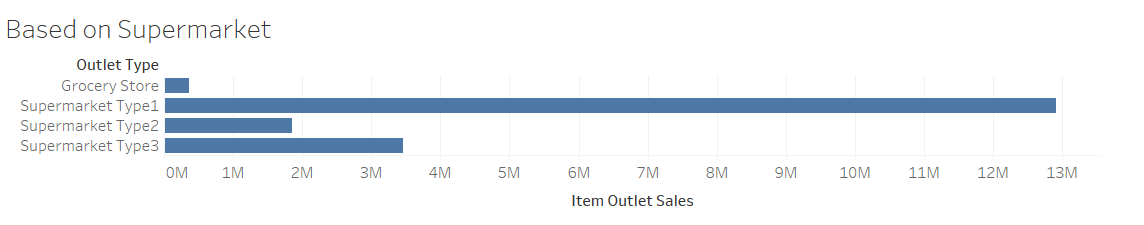
## **Exploratory data analysis in Tableau**

**Analysis 4:** Sum of sales based on Outlet Location Type

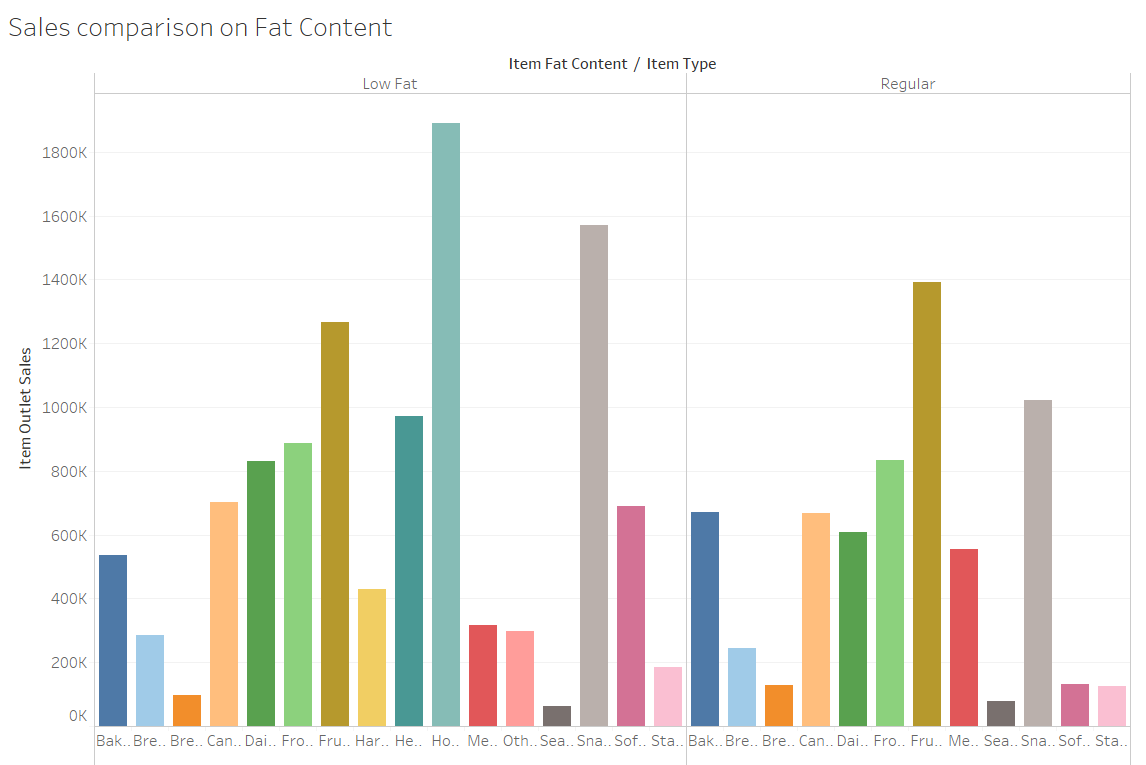
**Approach:**



**Analysis 5:** Sales based on Supermarket type



**Analysis 6:** Sales comparison based on Fat Content



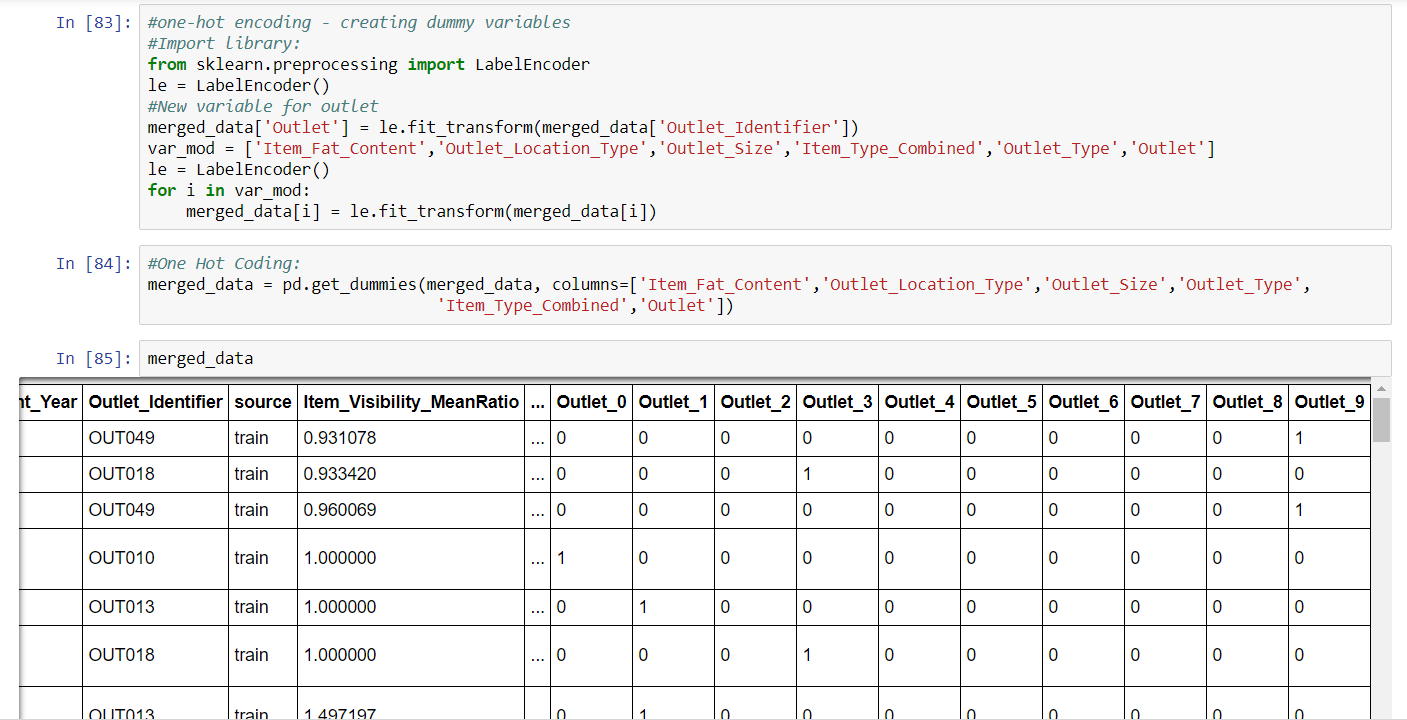
**Part 2: Building and Evaluating Models**

**Feature Selection**

Our main aim is to predict the sales of the BigMart outlets across cities for the test data. So predicting the sales means, it is a part of regression. After cleaning the data thoroughly, we proceed to feature Engineering and feature selection.

We created various new features like Item Visibility Mean Ratio. So, the next step is to convert all non-numeric data into numeric data and dummy variables using one-hot encoding.

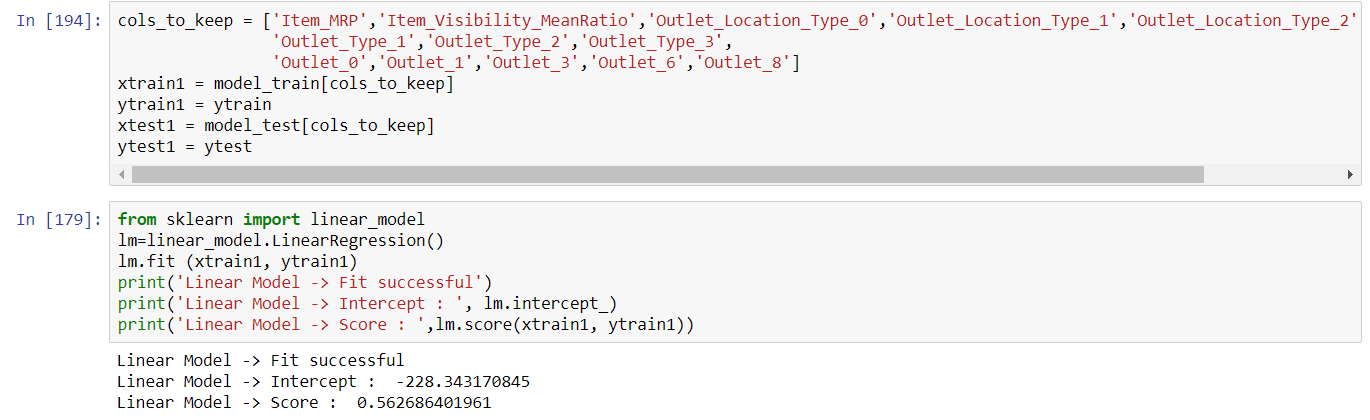
One-Hot-Coding refers to creating dummy variables, one for each category of a categorical variable. For example, the Item\_Fat\_Content has 3 categories – ‘Low Fat’, ‘Regular’ and ‘Non-Edible’. One hot coding will remove this variable and generate 3 new variables. Each will have binary numbers – 0 (if the category is not present) and 1(if category is present). This can be done using ‘get\_dummies’ function of Pandas.



Once our data is ready with all dummy variables and numeric variables, we proceed further to select the features which would have an impact in predicting the sales of the outlet. Based on correlation and p-values, I selected the features using back propagation and exhaustive method, trying out all values with different combination to get the maximum accuracy.

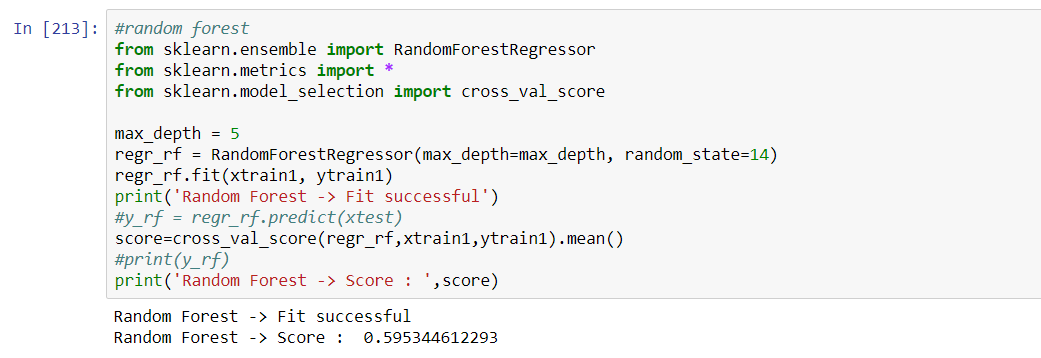
**Model Building and Prediction:**

Once we have the features ready, we create a model. I always start off with linear regression, which gives us an idea of the threshold value and if not always, more often than not we end up having a better accuracy than we get in Linear Regression.



The accuracy has been 56% while working on Linear Regression, which gives us a fair idea that this is the threshold and our actual accuracy rate would be near this percentage. If there is a large difference between two models, it would assure us that the model is overfitting, which is not a good sign for us.

We later implemented Random Forest Algorithm to compare the accuracy and we would consider the better of the two. It is always good to implement more than algorithm to get an idea of which algorithm would suit on a given dataset.



The accuracy of 60% might seem less overall, but given the dataset and the amount of skew shows us that the data is inclined mostly on a few items or outlets and hence, the prediction of sales based on just 8000 records would not be a good way, but gives us a fair chance to know the approximate value of the sales for each BigMart. Infact, this is the quiz dataset posted on Kaggle with zero kernels and probably the first solution to this competition. Also, we had the test dataset sales completely empty and hence we could not calculate the RMSE or any sort of error but the accuracy gives us an idea of how well the model performed for us.

**References:**

<https://www.kaggle.com/devashish0507/big-mart-sales-prediction>

<https://www.analyticsvidhya.com/blog/2016/02/bigmart-sales-solution-top-20/>