Big Mart Sales prediction

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 Description:

We have train (8523) and test (5681) data set, train data set has both input and output variable(s).We 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.

To solve this I have followed following steps:

1. *Data Exploration*
2. *Data Cleaning*
3. *Feature Engineering*
4. *Model Building*
5. *Conclusion based on RMSE value*

I have perform these stages with following stages viz

[Stage 1: Loading required Libraries](#_Stage_1:_Loading)

[Stage 2: Loading the data files from the local disk](#_Stage_2:_Loading)

[Stage 3: Exploring the data](#_Stage_3:_Exploring)

[Stage 4: Visualizing the Data for better understanding](#_Stage_4:_Visualizing)

[Stage 5A: Imputing NA values for continuous variables by replacing them with Mean](#_Stage_5A:_Imputing)

[Stage 5B: Imputing NA values for Categorical variables by replacing them with Mode](#_Stage_5B:_Imputing)

[Stage 6: Feature Engineering](#_Stage_6:_Feature)

[Stage 7: Model Building](#_Stage_7:_Model)

# Stage 1: Loading required Libraries

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

%matplotlib inline

#Import mode function:

from scipy.stats import mode

from sklearn.linear\_model import \*

import csv as csv

from scipy.stats import mode

from sklearn import cross\_validation, metrics

# Stage 2: Loading the data files from the local disk

Loaded files from the local disk and stored in the variables viz train and test:

train = pd.read\_csv("F://Python//train.csv")

test = pd.read\_csv("F://Python//test.csv")

# Stage 3: Exploring the data

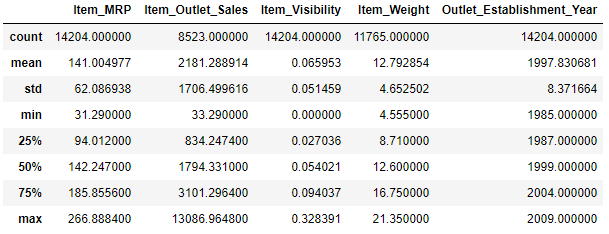
|  |  |  |
| --- | --- | --- |
| Dataset | Rows# | Columns# |
| Train | 8523 | 13 |
| Test | 5681 | 12 |
| Combined test and train as data | 14204 | 13 |

Here I have combine both train and test data sets into one, will perform feature engineering and then divide them later again. This saves the trouble of performing the same steps twice on test and train.

Checked for NA values with isnull()

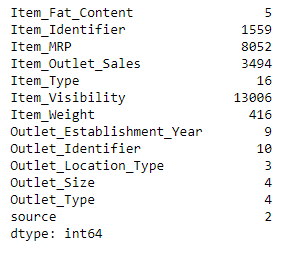
I have found NA values in Item\_Weight and Outlet\_Size that I have imputed in step 4.

I have checked basic statistics using describe()



It is cleared from above information that Outlet\_Establishment\_Years vary from 1985 to 2009. Item\_Visibility has a min value of zero. The Item is not visible cannot be sold so this cannot be zero.

With unique()I found that there are there are 1559 products and 10 outlets.



I have excluded the ID and source variables for obvious reasons. Then Filtered categorical variables and printed frequency of categories.

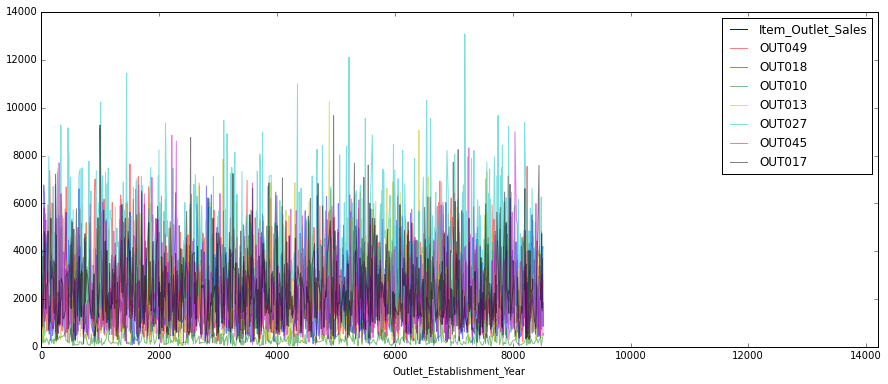
The output gives us following observations:

1. **Item\_Fat\_Content:** Some of ‘Low Fat’ values mis-coded as ‘low fat’ and ‘LF’. Also, some of ‘Regular’ are mentioned as ‘regular’.
2. **Item\_Type:** Not all categories have substantial numbers. It looks like combining them can give better results.

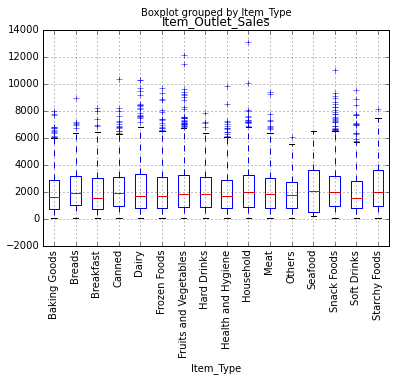
# Stage 4: Visualizing the Data for better understanding

I have got following insights.

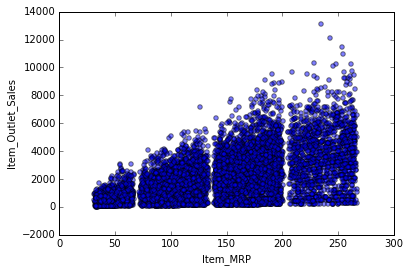
How Establishment year of an outlet year has an impact on sales of that outlet



Correlation between item type and sales for a particular store



Correlation between item prices and item outlet sales



# Stage 5A: Imputing NA values for continuous variables by replacing them with Mean

Item\_Weight

Item\_Identifier

Orignal #missing: 2439

Final #missing: 0

# Stage 5B: Imputing NA values for Categorical variables by replacing them with Mode

Mode for each Outlet\_Type:

Outlet\_Type

Grocery Store ([Small], [880.0])

Supermarket Type1 ([Small], [3100.0])

Supermarket Type2 ([Medium], [1546.0])

Supermarket Type3 ([Medium], [1559.0])

Name: Outlet\_Size, dtype: object

Orignal #missing: 4016

0

# Stage 6: Feature Engineering

I modified Item\_Visibility by considering as NA value.

Number of 0 values initially: 879

Number of 0 values after modification: 0

Then I created a broad category of Type of Item.

Food 10201

Non-Consumable 2686

Drinks 1317

Name: Item\_Type\_Combined, dtype: int64

Changed the categories of low fat and correcting the typos and differences in representation in categories of Item\_Fat\_Content variable

Original Categories:

Low Fat 8485

Regular 4824

LF 522

reg 195

low fat 178

Name: Item\_Fat\_Content, dtype: int64

Modified Categories:

Low Fat 9185

Regular 5019

Name: Item\_Fat\_Content, dtype: int64

Then marked non-consumables as separate category in low\_fat as "Non-Edible

Low Fat 6499

Regular 5019

Non-Edible 2686

Name: Item\_Fat\_Content, dtype: int64

I created a new column depicting the years of operation of a store.

count 14204.000000

mean 15.169319

std 8.371664

min 4.000000

25% 9.000000

50% 14.000000

75% 26.000000

max 28.000000

Name: Outlet\_Years, dtype: float64

This shows stores which are 4-28 years old.

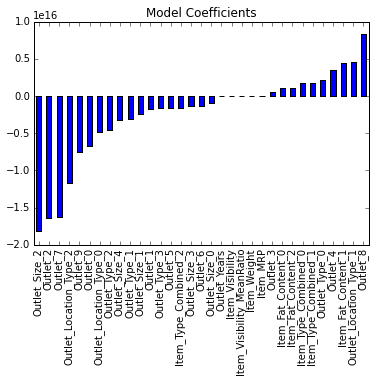
Exported this clean data into test.csv and train.csv

# Stage 7: Model Building

1. **Baseline model:**

Baseline model is the one which requires no predictive model and it’s like an informed guess. I predicted the sales as the overall average sales.

1. **Linear Regression Model:**

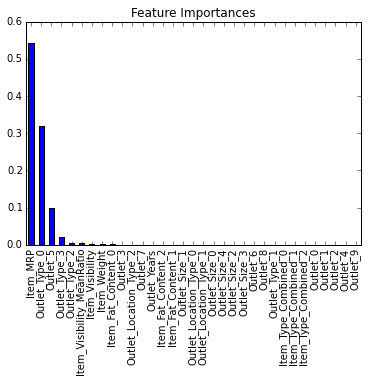


Model Report

RMSE : 1127

CV Score : Mean - 1129 | Std - 43.43 | Min - 1075 | Max – 1212

1. **Decision Tree Model:**

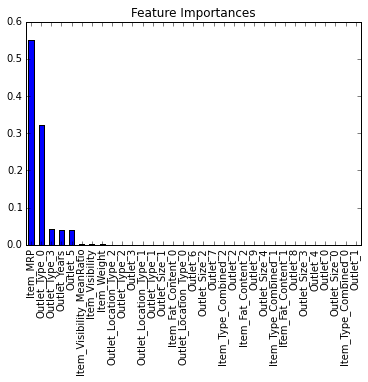


Model Report

RMSE : 1058

CV Score : Mean - 1093 | Std - 42.18 | Min - 1023 | Max - 1174

1. **Random Forest Model**:



Model Report

RMSE : 1068

CV Score : Mean - 1082 | Std - 43.05 | Min - 1021 | Max – 1160

**Conclusion:**

We have got RMSE: 1068 with Random forest model with max\_depth of 6 and 400 trees. Increasing the number of trees makes the model robust.

So this is the best Model I have found.

The output for sale is saved in CSV file and it is there in the submission.