



# Big Mart Sale Prediction

A Project Report

Submitted by:

**Pratyusa Kumar Dwibedy**

In fulfilment for the Internship

at

**Technocolabs**



July – August 2020

Project Directory:

GitHub: [https://github.com/pratyusa98/bigmart\\_heraku](https://github.com/pratyusa98/bigmart_heraku)

Project Website: <https://bigmartsaleproject.herokuapp.com/>

LinkedIn Profile: <https://www.linkedin.com/in/pdwibedy/>

## **Table of Contents**

- 1. Abstract**
- 2. Keywords**
- 3. Introduction**
- 4. Select a Performance Measure:**
- 5. Dataset Description of Big Mart:**
- 6. Organization of my analysis:**
  - 6.1 Exploratory data analysis (EDA)**
  - 6.2 Data Pre-processing:**
  - 6.3 Feature Transform:**
  - 6.4 Model Building:**
  - 6.5 Hyper parameter Tuning:**
  - 6.6 Use All Model To Predict**
- 7. Conclusion**
- 8. Test My Model**

# **Pratyusa Dwibedy (Technocolabs)**

## **Project Report Of Bigmart Sale Prediction**

### **1. Abstract:**

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 predict the sales of each product at a particular outlet. In this paper, I proposed a predictive model technique for predicting the sales of a company. Using this model, BigMart will try to understand the properties of products and outlets which play a key role in increasing sales.

### **2. Keywords:**

Machine Learning, Data Pre-processing, EDA, Linear Regression, Random Forest, Xgboost Regressor.

### **3. Introduction:**

According to the information provided, Bigmart is a big supermarket chain, with stores all around the country and its current board set out a challenge to all Data Scientist out there to help them create a model that can predict the sales, per product, for each store.

BigMart has collected sales data from the year 2013, for 1559 products across 10 stores in different cities. With this information the corporation hopes we can identify the products and stores which play a key role in their sales and use that information to take the correct measures to ensure success of their business.

### **4. Select a Performance Measure:**

Usually for regression problems the typical performance measure is the Root Mean Square Error (RMSE). This function gives an idea of how much error the system makes in its predictions with higher weight for large errors. Here also I use RMSE To different Model to find which model gives less rmse value.

### **5. Dataset Description of Big Mart:**

In our work i have used 2013 Sales data of Big Mart as the dataset. Where the dataset consists of 12 attributes like Item Fat, Item Type, Item MRP, Outlet Type, Item Visibility, Item Weight, Outlet Identifier, Outlet Size, Outlet Establishment Year, Outlet Location Type, Item Identifier and Item Outlet Sales. Out of these attributes response variable is the Item Outlet Sales attribute and remaining attributes are used as the predictor variables. The data-set consists of 8523 products across different cities and locations. The data-set is also based on hypotheses of store level and product level. Where store level involves attributes like: city, population density, store capacity, location, etc. and the product level hypotheses involves attributes like: brand, advertisement, promotional offer, etc. After considering all, a dataset is formed and finally the data-set was divided into two parts, training set and test set in the ratio 80 : 20.

## 6. Organization of my analysis:

For analysis of data I am divide the whole programme into 6 steps:-

1. Exploratory data analysis (EDA)
2. Data Pre-processing
3. Feature Transformation
4. Machine Modelling
5. Hyper parameter tuning
6. Use all model to predict

### 6.1 Exploratory data analysis (EDA):

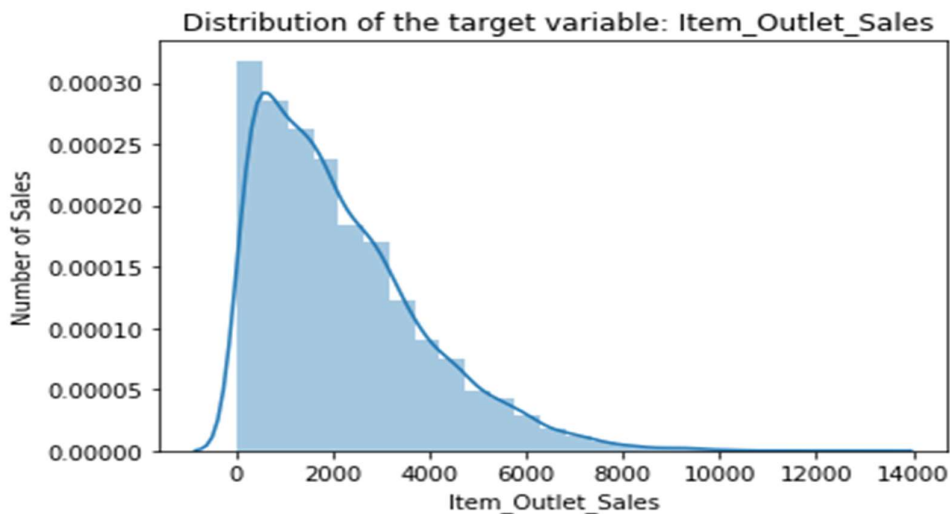
This is the First Step in my project. The goal for this section is to take a glimpse on the data as well as any irregularities.

I first check if the dataset suffers any duplicate values in “Item\_Identifier” feature. Since a product can exist in more than one store it is expected for this repetition to exist. There seems to be 1562 unique items only available in one single store.

Then I am do Univariate Analysis

#### 6.1.2 Univariate Analysis:-

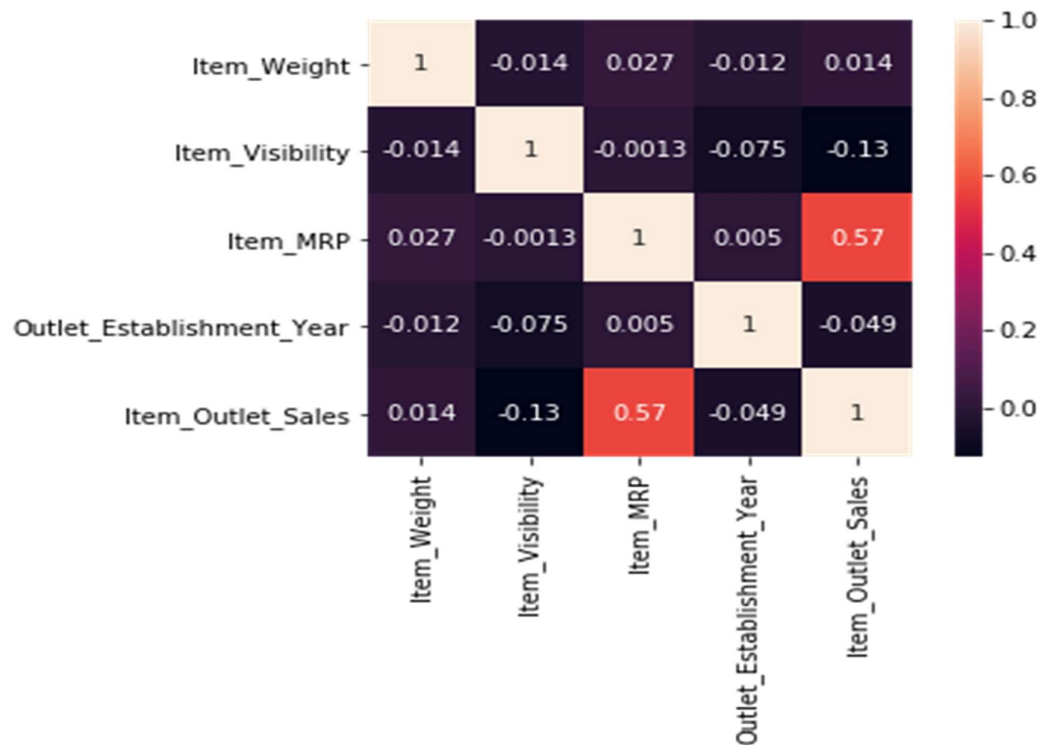
Here 1<sup>st</sup> I see the distribution of target variable as it is Gaussian or not



(Fig-1 distribution of target variable)

Here our target variable is skewed to the right, towards the higher sales, with higher concentration on lower sales.

Then I Draw the correlation matrix between feature and predictor variable.



(Fig – 2 correlation matrix between feature and predictor variable)

I can observe that the Item\_Visibility is the feature with the lowest correlation with target variable. Therefore, the less visible the product is in the store the higher the price will be. This is curious since from the initial assumptions this variables was expected to have high impact in the sales increase. Moreover, this feature has a negative correlation with all of the other features. Furthermore, the most positive correlation belongs to Item\_MRP .

By Doing EDA I concluded the following Things:-

1. Regarding the variables which were thought to have high impact on the product's sale price. Item\_Visibility does not have a high positive correlation as expected, quite the opposite. As well, there are no big variations in the sales due to the Item\_Type .
2. I see in Item Identifier Three field are 'FD' (Food), 'DR'(Drinks) and 'NC' (Non-Consumable). By use this i create a new variable.
3. In Item\_Visibility there are items with the value zero. This does not make lot of sense, since this is indicating those items are not visible on the store.
4. There seems to be 1562 unique items only available in a single store.
5. Item\_Fat\_Content has vale "low fat","reg" written in different manners.
6. For Item\_Type we try to create a new feature that does not have 16 unique values.

## **6.2 Data Pre-processing:**

For imputing missing value where the data is numeric I use mean and median to impute the missing field but where the data is object (categorical) there I used mode value to filled this in this way I impute all missing value in the data set.

## **6.3 Feature Transform:**

Some nuances were observed in the data-set during data exploration phase. So this phase is used in resolving all nuances found from the dataset and make them ready for building the appropriate model. During this phase it was noticed that the Item visibility attribute had a zero value, practically which has no sense. So the mean value item visibility of that product will be used for zero values attribute. This makes all products likely to sell. All categorical attributes discrepancies are resolved by modifying all categorical attributes into appropriate ones. In some cases, it was noticed that non-consumables and fat content property are not specified. To avoid this we create a third category of Item fat content i.e. none. In the Item Identifier attribute, it was found that the unique ID starts with either DR or FD or NC. So, we create a new attribute Item Type New with three categories like Foods, Drinks and Non-consumables. Finally, for determining how old a particular outlet is, we add an additional attribute Year to the dataset. Then I do label encoding to convert all categorical feature to numerical feature.

## **6.4 Model Building:**

After completing the previous phases, the dataset is now ready to build proposed model. Once the model is build it is used as predictive model to forecast sales of Big Mart. In my work, i am using Xgboost Regression, Linear regression, Random Forest Regression etc.

### **6.4.1 Feature Scalling**

Before give the data in to various model I do some feature scaling of data. As the data is not same range so by doing feature scaling all the data are get in same range. For feature scaling I use standard scalar. Given the code:

```
from sklearn.preprocessing import StandardScaler  
  
sc = StandardScaler()  
  
then fit the data in the scaling.
```

## 6.5 Hyper parameter Tuning:

Here I use randomize search cv for tuning the model. Here I only tune the xgboost model. So first assign all parameter in a variable like:

```
## Hyper Parameter Optimization
```

```
params={  
    "learning_rate" : [0.05, 0.10, 0.15, 0.20, 0.25, 0.30 ],  
    "max_depth"      : [ 3, 4, 5, 6, 8, 10, 12, 15],  
    "min_child_weight" : [ 1, 3, 5, 7 ],  
    "gamma"          : [ 0.0, 0.1, 0.2 , 0.3, 0.4 ],  
    "colsample_bytree" : [ 0.3, 0.4, 0.5 , 0.7 ]  
}
```

Then Import Randomize search cv

```
## Hyper parameter optimization using RandomizedSearchCV
```

```
from sklearn.model_selection import RandomizedSearchCV
```

Then initialize classifier

```
classifier=xgboost.XGBClassifier()
```

then apply randomize search cv

```
random_search=RandomizedSearchCV(classifier,param_distributions=params,n_iter=5,scoring='roc_auc',n_jobs=-1,cv=5,verbose=3)
```

Then fit the value

```
random_search.fit(X,y)
```

after that it runs and give me best parameter and score for the given dataset.

## 6.6 Use All Model To Predict

**Linear Regression:** A model which create a linear relationship between the dependent variable and one or more independent variable, mathematically linear regression is defined in Equation  $y = wT x$  where y is dependent variable and x are independent variables or attributes. In linear regression we find the value of optimal hyperplane w which corresponds to the best fitting line (trend) with minimum error. In this I got

```
Score of Training: 36.07654777886642
RMSE : 1375.0955709405644
Score of Testing: 35.817295123540404
RMSE : 1320.782611366126
Mean Absolute Error 991.8540303766889
```

**Random Forest Regressor:** it is the ensemble technique . it is extened form of decision tree. It is a tree base method. Random forest is a bagging technique and not a boosting technique. The trees in random forests are run in parallel. There is no interaction between these trees while building the trees. In this I got

```
Score of Training: 61.534008248805975
RMSE : 1066.5864051474712
Score of Testing: 60.449521636934975
RMSE : 1052.5237949756458
Mean Absolute Error 736.0231854304849
```

**XGBOOST Regressor:** XGboost (Extreme Gradient Boosting) is a modified version of Gradient Boosting Machines (GBM) which improves the performance upon the GBM framework by optimizing the system using a differentiable loss function. In this I got high accuracy and I am use this model to deploy in heraku with flask framework.

```
Score of Training: 0.9831032089018564
RMSE : 223.56555390702675
Score of Testing: 0.9834213528159617
RMSE : 212.27403701634336
Mean Absolute Error 138.69489805492816
```

**Comparison of MAE and RMSE of proposed model with other Model (Table-1)**

Model Name	MAE	RMSE
Linear Regression	991.854	1320.782
Random Forest Regressor	736.023	1052.523
XGboost Regressor	138.694	212.274



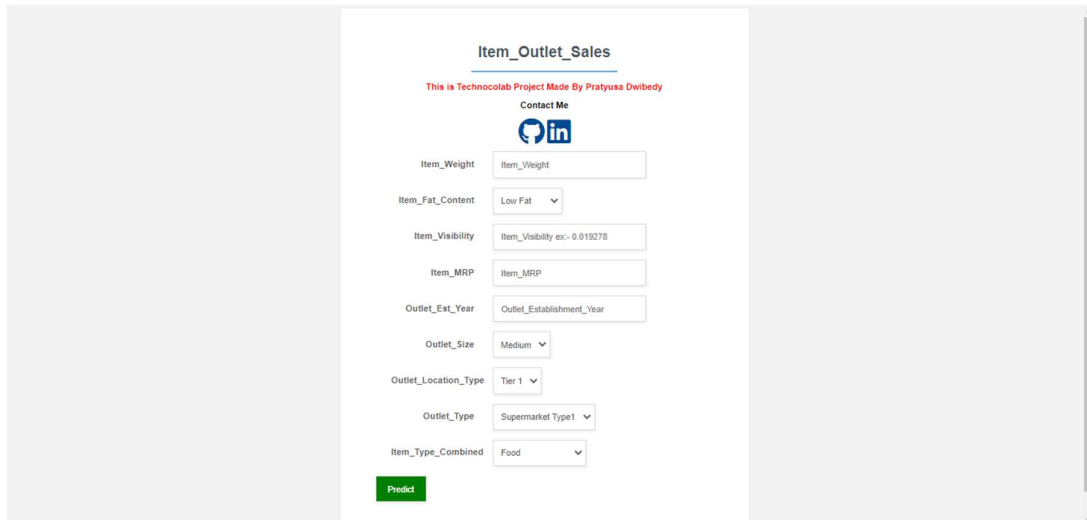
## **7. Conclusion:**

In present era of digitally connected world every shopping mall desires to know the customer demands beforehand to avoid the shortfall of sale items in all seasons. Day to day the companies or the malls are predicting more accurately the demand of product sales or user demands. Extensive research in this area at enterprise level is happening for accurate sales prediction. As the profit made by a company is directly proportional to the accurate predictions of sales, the big marts are desiring more accurate prediction algorithm so that the company will not suffer any losses. In this research work, i have designed a predictive model by modifying Gradient boosting machines as Xgboost technique and experimented it on the 2013 Big Mart dataset for predicting sales of the product from a particular outlet. Experiments support that our technique produce more accurate prediction compared to than other available techniques like decision trees, linear regression etc. Finally a comparison of different models is summarized in Table 1. From Table 1 it is also concluded that our model with lowest MAE and RMSE performs better compared to existing models.

## 8. Test My Model:

After Implement I use flask to deploy in heroku platform. By this 1<sup>st</sup> I export the xgboost model as pickle file, then by this file I made a small website by which any one can access the model efficiently.

Website Model:



(Fig-3 Website View Of My Project)

Output:

---

**Here is the result page**

**Hurray!**

**You Can Sell This Outlet At  
2907.67 price**

**[Back and Again Predict!!!](#)**

(Fig – 4 Output Show in Website)