International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 09 Issue: 12 | Dec 2022 www.irjet.net p-ISSN: 2395-0072

BIG MART SALES PREDICTION USING MACHINE LEARNING

[1]Kasireddy Raghuvardhan Reddy, [2]Kolipaka Rajesh, [3]Bhukya Pavan Kalyan, [4] V. Prabhakar

[4]Assistant Professor, Department of Computer Science and Engineering, SNIST, Hyderabad-501301, India
[1,2,3]B. Tech Scholars, Department of Computer Science and Engineering, SNIST, Hyderabad-501301, India

---

Abstract: *- Supermarkets and their franchises are increasing a lot in recent times. At this moment to increase their sales, they have to predict the sales of the items. Thereby they can protect themselves from losses and they can generate profits. So, this analysis will require a lot of time and effort. So, we proposed a machine learning model that will use the XGBoost Regressor to predict the sales of the items. Thereby marts can plan their recruitment strategy, perceive challenges early, motivate the sales team, predict revenue, aid future marketing plans, and helps in many more ways.*

I. INTRODUCTION

At present, there are many supermarkets and there is high competition between them. If any mart wants to win the competition it has to make more sales than any other competitor. The mart can increase sales by knowing products/items which will generate more sales and those that will generate low sales. This analysis is tedious. So with the proposed system, the mart can predict the sales, thereby marts can take necessary steps. This is a user-friendly system. When the user submitted details of a particular item, the system will predict sales generated by that item. Hence, this leads to winning in the competition and an increase in sales.

II. LITERATURE SURVEY

Earlier we are having different methods for predicting the sales such as:

Expert's Opinion Method: Here, marketing and sales professionals will make sales predictions. Analysis and sales forecasting are labor-intensive tasks. However, predictions can also be inaccurate for a variety of reasons, including a lack of enthusiasm for the task being performed and other problems.

Survey of Buyer's Expectations: In this case, the store will poll customers based on their preferences, enabling it to forecast which products would result in the greatest increase in sales. The actual purchases could differ from the declared aims, though. Unfortunately, it caused incorrect sales predictions at the time.

To overcome those drawbacks we proposed a model which was developed using machinelearning. In this, we used XGBoost Regressor for predicting the sales based on the customer purchases but not on the expert's opinion and survey results. We would get accurate results compared to the previous system.

III. DATASET

A collection of data points that a computer may use to analyze and anticipate a situation as a whole. Internet information was gathered for the Kaggle.com website. The test data set used in this study comprises 8542 rows as well as 12 categories, which have been trained to deliver the most accurate prediction outcomes.

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.

Fig:1 Data set used

IV. METHODOLOGY

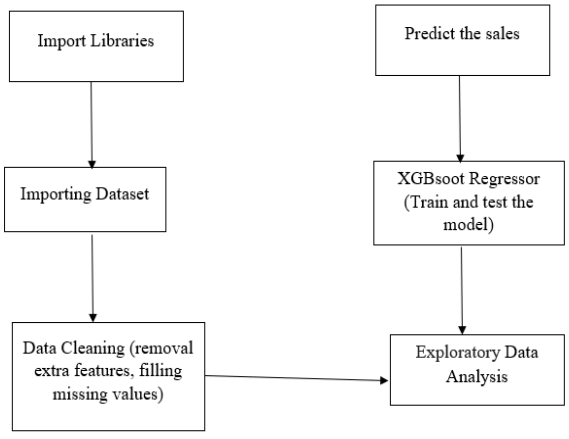

Fig: 2 Data flow Diagram

V. DATA PREPROCESSING

5.1 Filling Missing Values

In data preprocessing, we will find and fill the missing values with either mean, mode, or median.

```
dTrain['Item_Weight'].fillna(dTrain['Item_Weight'].mean(),inplace=True)
dTrain['Outlet_Size'].fillna(dTrain['Outlet_Size'].mode()[0],inplace=True)
dTest['Item_Weight'].fillna(dTest['Item_Weight'].mean(),inplace=True)
dTest['Outlet_Size'].fillna(dTest['Outlet_Size'].mode()[0],inplace=True)
dTrain.drop(['Item_Identifier','Outlet_Identifier'],axis=1,inplace=True)
dTest.drop(['Item_Identifier','Outlet_Identifier'],axis=1,inplace=True )
```

Fig:3 Filling missing values

5.2 Removing Unwanted Attributes

Those attributes which will not involve in the prediction can be removed.

```
dTrain.drop(['Item_Identifier','Outlet_Identifier'],axis=1,inplace=True)
dTest.drop(['Item_Identifier','Outlet_Identifier'],axis=1,inplace=True )
```

Fig 4: Removing Unwanted Attributes

5.3 Exploratory Data Analysis

Exploratory data analysis is the crucial process of doing preliminary analyses of data in order to find patterns, identify anomalies, test hypotheses, and double-check assumptions with the use of summary statistics and graphical representations.

Data Visualization

By putting the data in a graphical framework, such as graphs, and visualization of the data provides a better understanding of what that implies.

This makes it easier for us to interpret the data intuitively and to spot patterns, trends, and abnormalities in large datasets.

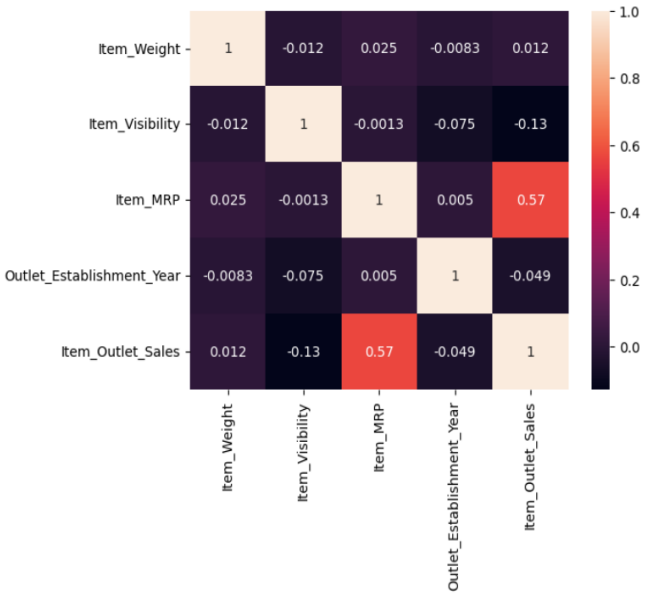

Fig 5 : heatmap for data visualization

5.4 Data Cleaning

Cleaning the dataset by making names of attributes from initial capital letters to small letters.

```
: import klib
klib.clean_column_names(dTrain)
```

Fig 6: cleaning the data

5.5 Label Encoding

Label encoding is the process of transforming labels into a numeric form so that they may be read by machines. The operation of such labels can then be better determined by machine learning techniques. For the structured dataset in supervised learning, it is a crucial pre-processing step.

```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
dTrain['item_fat_content']=le.fit_transform(dTrain['item_fat_content'])
dTrain['item_type']=le.fit_transform(dTrain['item_type'])
dTrain['outlet_size']=le.fit_transform(dTrain['outlet_size'])
dTrain['outlet_location_type']=le.fit_transform(dTrain['outlet_location_type'])
dTrain['outlet_type']=le.fit_transform(dTrain['outlet_type'])
```

Fig 7: Label Encoding

5.6 Splitting the training and testing data

The training and testing sets for the dataset were going to be separated. Two separate datasets are not imported for train and testing in order to prevent the overfitting process. The same dataset is thus divided into train and test sets. The datasets utilized to train and test our model are referred to

collectively as the training dataset and the testing dataset, respectively.

```
X=dTrain.drop('item_outlet_sales',axis=1)
Y=dTrain['item_outlet_sales']
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test= train_test_split(X,Y,random_state=101,test_size=0.2)
```

Fig 8 : splitting training and testing data

5.7 Standardization

By utilizing the mean and standard deviation as the starting point to obtain particular values, data values are rescaled to fit the distribution between 0 and 1. This is known as standardization.

```
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
X_train_std=sc.fit_transform(X_train)
X_test_std=sc.fit_transform(X_test)
```

Fig 9: Standardization

5.8 Evaluation Metrics

The assessment of the model is a vital stage in creating a successful machine-learning model. It is essential to create a model and get metrics suggestions from it as a consequence.

It will happen and keep happening until we achieve high accuracy in line with the value achieved through metric improvements.

Evaluation metrics are used to describe the output of one model. The ability of the assessment metrics to distinguish between various model outputs is an important feature. This assessment approach made use of the Root Mean Squared Error (RMSE)metric.

R-squared is a statistical indicator of how well a regression model fits the data. R-square should be set to a value of 1. The fit of the model is improved by the r-square value being near 1.

VI. MODEL BUILDING

After data preprocessing, the dataset is now ready to be used to build a predictive model. Training data is provided to the algorithm to train it to forecast values. The model creates a target variable to forecast before receiving input from the testing data. The XGBoost Regressor is used to construct the prediction model.

XGBOOST:

Decision trees and gradient boosting are combined in the XG Boost method. The method's development was intended to maximize the efficiency of memory and processing resources. The ensemble idea serves as the foundation for the sequential "boosting" procedure. The accuracy rate is increased as a consequence, and a group of slow learners is added. At each instant t, the weights of the model variables are determined by the impact of the instant before. Outcomes that are correctly computed have a lower weight than results that are incorrectly computed, which have a greater weight.

Internally, the XGBoost model implements stepwise ridge regression using this method, which automatically selects features and eliminates repeated regressions.

```
!pip install xgboost
from xgboost import XGBRegressor
xrg=XGBRegressor(n_estimators=1000,learning_rate=0.02,max_depth=10,
                 colsample_bytree=1,subsample=0.95,seed=1,min_child_weight=2)
xrg.fit(X_train_std,Y_train)
predictions = xrg.predict(X_test_std)
```

Fig 10: Building the model

VII. RESULT

After testing the model with testing data, we will find the RMSE (Root Mean Square Error), r2_score for accuracy, and mean absolute error.

```
from sklearn import metrics
rmse = np.sqrt(metrics.mean_squared_error(Y_test,predictions))
rmse
```

```
1142.5375
```

```
print(metrics.r2_score(Y_test,predictions))
```

```
0.5210212103560086
```

```
print('Mean Absolute Error:', metrics.mean_absolute_error(Y_test, predictions))
```

```
Mean Absolute Error: 798.8241
```

Fig 11: Results

VIII. CONCLUSION

We are predicting the XG Boost Regressor's accuracy. Large markets benefit from our projections by improving their methods and tactics, which boosts their profitability. The expected outcomes will be highly helpful for the company's leaders to understand their sales and profitability. This will also inspire them to create more Bigmart Stores or branches.

IX. REFERENCES

[1] Ayesha Syed, Asha Jyothi Kalluri, Venkateswara Reddy Pocha, Venkata Arun Kumar Dasari, B.Ramasubbaiah (2020, FEB). "BIGMART SALES USING MACHINE LEARNING WITH DATA ANALYSIS". In JES (Vol 11,Issue 2).

[2] Aaditi Narkhede, Mitali Awari, Suvarna Gawali, Prof.Amrapal Mhaisgawali. "BigMart Sales Prediction Using Machine Learning Techniques" Naveenraj R and Vinayaga Sundharam R

[3] "PREDICTION OF BIG MART SALES USING MACHINE LEARNING" in IRJMETS (Volume:03/Issue:09/September-2021)

[4] Rohit Sav, Pratiksha Shinde, Saurabh Gaikwad "BIG MART SALES PREDICTION USING MACHINE LEARNING"

[5] Predicitive Analysis for Big Mart Sales Using Machine Learning Algorithm" in IJRASET Volume 10 Issue VIII August 2022

[6] Nayana R, Chaithanya G, Meghana T, Narahari K S, Sushma M "Predictive Analysis for Big Mart Sales using Machine Learning Algorithms" in IJERT RTCSIT – 2022 (Volume 10 – Issue 12)