BIGMART SALES PREDICTION

By – Himanshi Garg

Date – 13 June 2023

Abstract

The BigMart Sales Prediction App project develops a machine learning model for sales prediction in BigMart stores. It utilizes historical sales data and other relevant features to build an accurate predictive model, aiding store managers in decision-making and inventory optimization. The report covers the problem statement, dataset details, methodology, and implementation of machine learning algorithms for sales prediction. Results are compared, and the best-performing model is selected for deployment. The conclusion highlights the model's effectiveness, practical implications for inventory management, and potential for improving store performance and profitability in the retail industry.

1.0 INTRODUCTION

CONTEXT

The retail industry, characterized by intense competition and evolving consumer preferences, demands accurate sales predictions to facilitate effective inventory management and optimize store performance. BigMart, a prominent retail chain, faces the challenge of accurately forecasting sales for its wide range of products across multiple stores. Inaccurate sales predictions can result in overstocking or understocking, leading to increased costs, wastage, and missed sales opportunities.

Currently, BigMart relies on manual and intuition-based methods for sales forecasting, which are time-consuming, subjective, and prone to errors. This approach hinders the ability to proactively manage inventory, identify underperforming products, and make data-driven decisions to drive sales and profitability.

Therefore, the problem at hand is to develop a robust machine learning model that leverages historical sales data and relevant features to accurately predict product sales in BigMart stores. The model should provide reliable forecasts, enabling store managers to optimize inventory levels, streamline supply chain operations, and implement effective sales strategies. By addressing this problem, BigMart can enhance its competitive edge, improve customer satisfaction, and maximize profitability.

PURPOSE

Day by day competition among different shopping malls as well as big marts is getting more serious and aggressive only due to the rapid growth of the global malls and on-line shopping. Every mall or mart is trying to provide personalized and short-time offers for attracting more customers depending upon the day, such that the volume of sales for each item can be predicted for inventory management of the organization, logistics and transport service, etc. Present machine learning algorithm are very sophisticated and provide techniques to predict or forecast the future demand of sales for an organization, which also helps in overcoming the cheap availability of computing and storage systems

OBJECTIVES

Develop an accurate predictive model, optimize inventory management, improve decision-making, enhance store performance and profitability, provide a user-friendly app interface, evaluate and compare different models, documentation and reporting that is a comprehensive report documenting the project's objectives, methodology, implementation details, and evaluation results are the essential objectives

1.1 INITIAL NEEDS

To deploy the BigMart Sales Prediction App, the following initial needs must be addressed: setting up the infrastructure for hosting the application, establishing the development environment with necessary software and tools, designing the app's architecture and user interface, integrating it with the predictive model, ensuring data integration with relevant sources, implementing security measures, conducting thorough testing and quality assurance, preparing documentation and user guides, creating a deployment plan, and setting up monitoring and maintenance procedures. By fulfilling these needs, the app can be successfully deployed, providing store managers with an intuitive interface for accessing accurate sales predictions and making informed decisions.

2.0 CUSTOMER/MARKET NEEDS

1. **Accurate Sales Forecasting**: Customers and the market require a reliable solution that can accurately predict sales for various products in BigMart stores. Accurate sales forecasting enables better decision-making, inventory management, and resource allocation.
2. **Inventory Optimization**: BigMart needs to optimize its inventory levels to prevent stockouts and overstocking. By accurately predicting sales, the app can help store managers determine the right stock levels, reduce wastage, and ensure product availability to meet customer demands.
3. **Cost Reduction**: Cost reduction is a critical need for any retail business. By having accurate sales predictions, BigMart can effectively manage inventory and reduce costs associated with excess stock, storage, and logistics.
4. **Increased Profitability**: The ultimate goal for BigMart is to increase profitability. Accurate sales predictions enable store managers to identify high-demand products, implement effective pricing strategies, and improve overall store performance to maximize profits.
5. **Data-Driven Decision-Making**: In today's data-driven world, businesses require insights to make informed decisions. The BigMart Sales Prediction App provides store managers with data-driven insights to optimize inventory, plan promotions, and allocate resources effectively.
6. **Enhanced Customer Satisfaction:** Meeting customer demands and providing a seamless shopping experience is vital for BigMart's success. Accurate sales predictions enable store managers to ensure product availability, reduce stockouts, and deliver a better customer experience.

3.0 TARGET SPECIFICATION AND CHARACTERISATION

3.1 Accuracy: The predictive model should aim for a high level of accuracy in sales forecasting. The target specification can define the desired accuracy level, such as a low margin of error or a specific evaluation metric (e.g., root mean square error, mean absolute error).

3.2 Scalability: The app should be designed to handle a large volume of data and be capable of scaling with increasing data and user demands. This includes optimizing the app's architecture, database design, and computational efficiency to ensure smooth performance.

3.3.User-Friendly Interface: The app should have an intuitive and user-friendly interface that allows store managers to input parameters, view sales predictions, and interpret the results easily. The target specifications may include considerations for clear visualizations, interactive features, and efficient user interaction.

3.4 Real-Time or Near Real-Time Prediction: Depending on the specific requirements, the app may need to provide real-time or near real-time sales predictions. The target specification can define the desired latency or frequency of updating predictions based on incoming data.

3.5 Robustness and Error Handling: The app should be robust and handle various scenarios, including missing or inconsistent data, outliers, and system failures. It should have appropriate error handling mechanisms and provide informative error messages to users when issues arise.

3.6 Security and Data Privacy: The app should adhere to security best practices to protect sensitive data and ensure data privacy. This may include encryption of data in transit and at rest, access controls, and compliance with relevant data protection regulations.

3.7 Integration with Existing Systems: If required, the app should seamlessly integrate with BigMart's existing systems, such as data warehouses, databases, or APIs. The target specifications may outline the specific integration points and compatibility requirements.

4.0 EXTERNAL SEARCHES

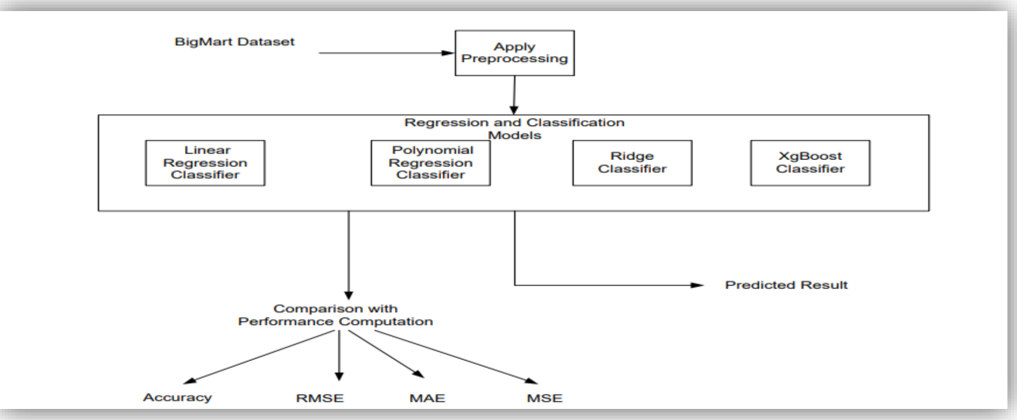
I have collected the information for visualizing and completion of the projects from the following websites and research papers –

https://www.hackersrealm.net/post/bigmart-sales-prediction-analysis-using-python

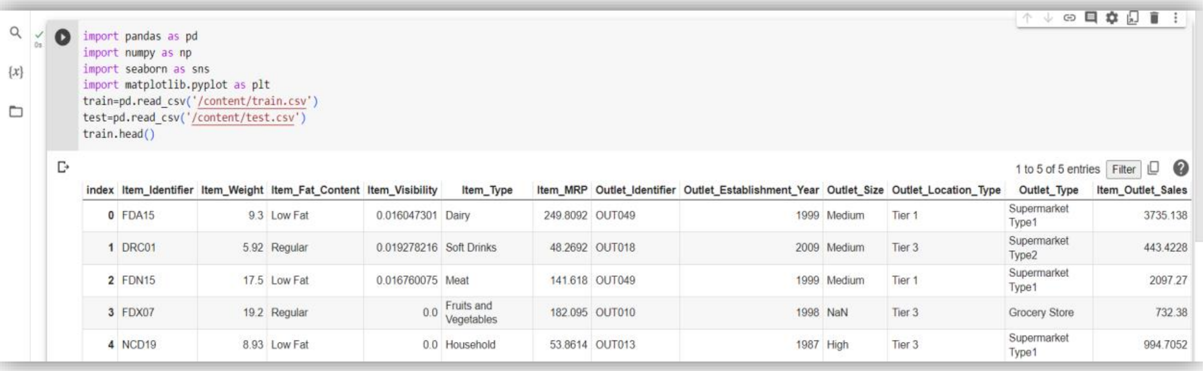
https://www.kaggle.com/code/hiralmshah/bigmart-sales-prediction

http://dx.doi.org/10.1007/978-981-15-4015-8_37

4.1 BENCHMARKING


BigMart Dataset
Apply Preprocessing
Regression and Classification Models
Linear Regression Classifier
Polynomial Regression Classifier
Ridge Classifier
XgBoost Classifier
Predicted Result
Comparison with Performance Computation
Accuracy
RMSE
MAE
MSE

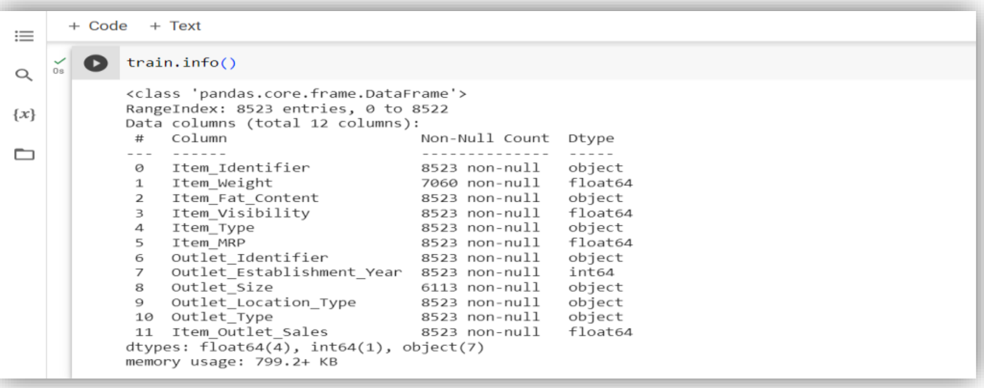
Taking a look at the dataset –



```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
train=pd.read_csv('/content/train.csv')
test=pd.read_csv('/content/test.csv')
train.head()
```

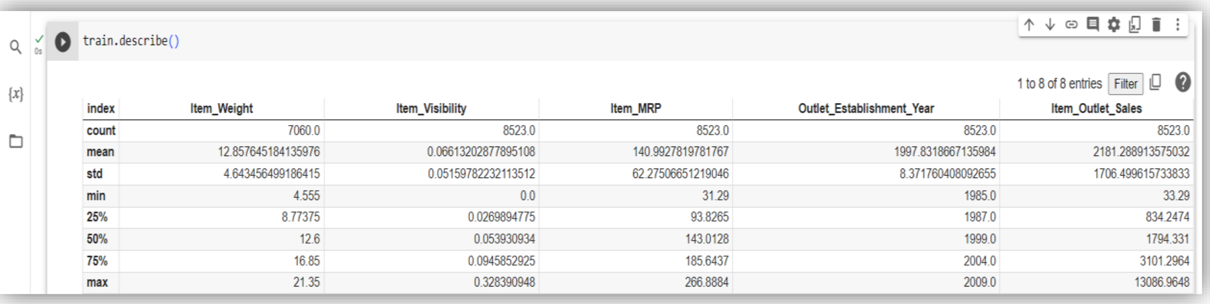
1 to 5 of 5 entries

index	Item_Identifier	Item_Weight	Item_Fat_Content	Item_Visibility	Item_Type	Item_MRP	Outlet_Identifier	Outlet_Establishment_Year	Outlet_Size	Outlet_Location_Type	Outlet_Type	Item_Outlet_Sales
0	FDA15	9.3	Low Fat	0.016047301	Dairy	249.8092	OUT049	1999	Medium	Tier 1	Supermarket Type1	3735.138
1	DRC01	5.92	Regular	0.019278216	Soft Drinks	48.2692	OUT018	2009	Medium	Tier 3	Supermarket Type2	443.4228
2	FDN15	17.5	Low Fat	0.016760075	Meat	141.618	OUT049	1999	Medium	Tier 1	Supermarket Type1	2097.27
3	FDX07	19.2	Regular	0.0	Fruits and Vegetables	182.095	OUT010	1998	NaN	Tier 3	Grocery Store	732.38
4	NCD19	8.93	Low Fat	0.0	Household	53.8614	OUT013	1987	High	Tier 3	Supermarket Type1	994.7052



```
train.info()
```

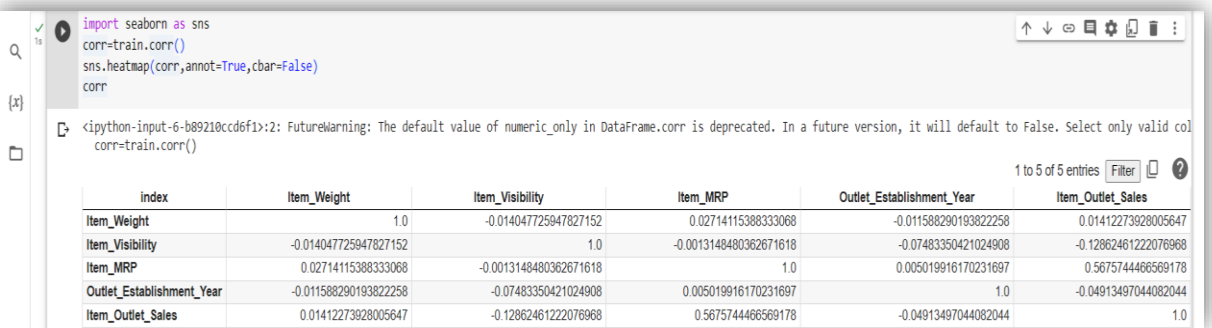
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8523 entries, 0 to 8522
Data columns (total 12 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   Item_Identifier            8523 non-null   object
 1   Item_Weight                7060 non-null   float64
 2   Item_Fat_Content           8523 non-null   object
 3   Item_Visibility            8523 non-null   float64
 4   Item_Type                  8523 non-null   object
 5   Item_MRP                   8523 non-null   float64
 6   Outlet_Identifier          8523 non-null   object
 7   Outlet_Establishment_Year  8523 non-null   int64
 8   Outlet_Size                6113 non-null   object
 9   Outlet_Location_Type       8523 non-null   object
 10  Outlet_Type                8523 non-null   object
 11  Item_Outlet_Sales          8523 non-null   float64
dtypes: float64(4), int64(1), object(7)
memory usage: 799.2+ KB
```



```
train.describe()
```

1 to 8 of 8 entries

index	Item_Weight	Item_Visibility	Item_MRP	Outlet_Establishment_Year	Item_Outlet_Sales
count	7060.0	8523.0	8523.0	8523.0	8523.0
mean	12.857645184135976	0.06613202877895108	140.9927819781767	1997.8318667135984	2181.288913575032
std	4.643456499186415	0.05159782232113512	62.27506651219046	8.371760408092655	1706.499615733833
min	4.555	0.0	31.29	1985.0	33.29
25%	8.77375	0.0269894775	93.8265	1987.0	834.2474
50%	12.6	0.053930934	143.0128	1999.0	1794.331
75%	16.85	0.0945852925	185.6437	2004.0	3101.2964
max	21.35	0.328390948	266.8884	2009.0	13086.9648

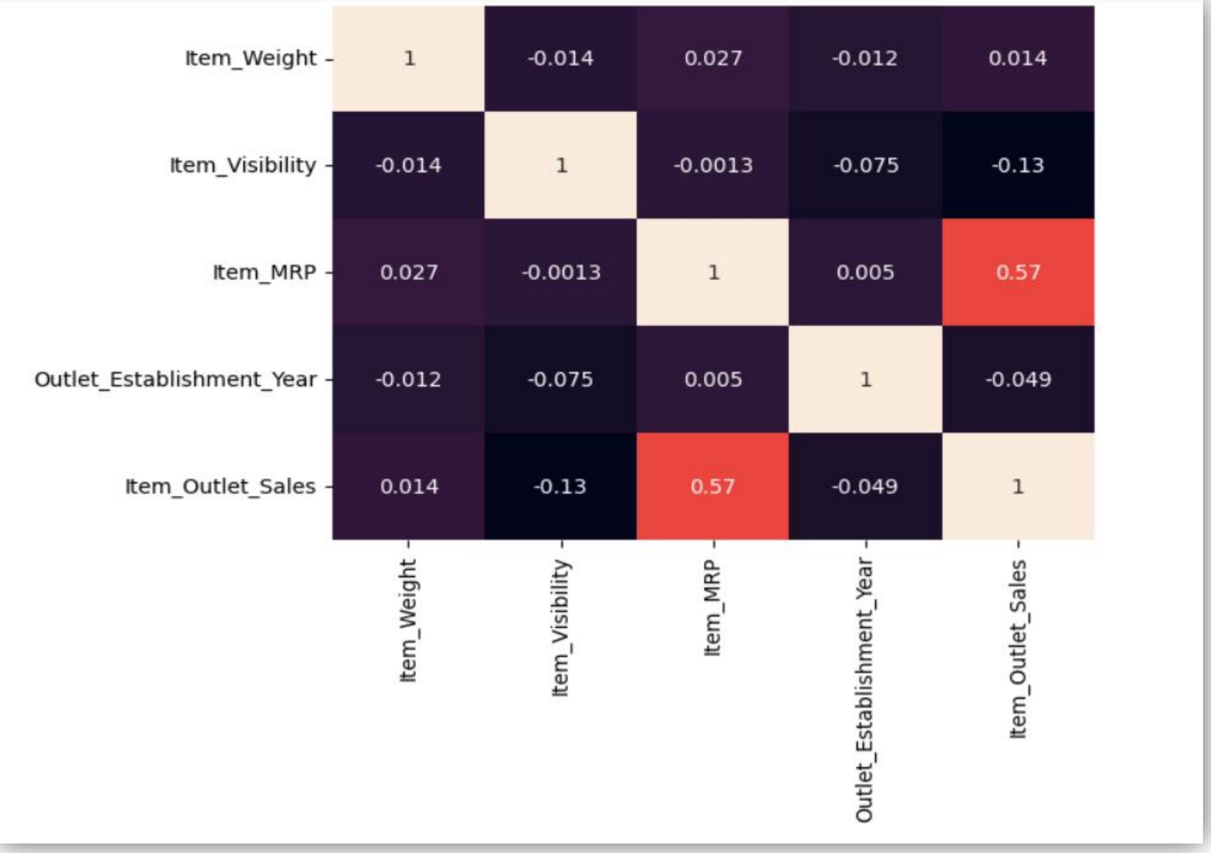


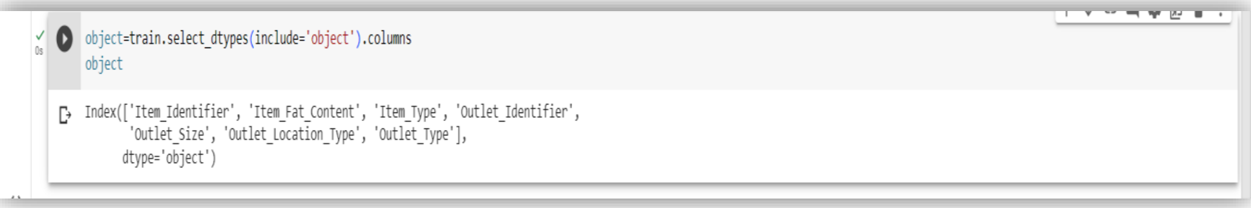
```
import seaborn as sns
corr=train.corr()
sns.heatmap(corr,annot=True,cbar=False)
corr
```

```
<ipython-input-6-b89210ccd6f1>:2: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid col
  corr=train.corr()
```

1 to 5 of 5 entries Filter

index	Item_Weight	Item_Visibility	Item_MRP	Outlet_Establishment_Year	Item_Outlet_Sales
Item_Weight	1.0	-0.014047725947827152	0.02714115388333068	-0.011588290193822258	0.01412273928005647
Item_Visibility	-0.014047725947827152	1.0	-0.0013148480362671618	-0.07483350421024908	-0.12862461222076968
Item_MRP	0.02714115388333068	-0.0013148480362671618	1.0	0.005019916170231697	0.5675744466569178
Outlet_Establishment_Year	-0.011588290193822258	-0.07483350421024908	0.005019916170231697	1.0	-0.04913497044082044
Item_Outlet_Sales	0.01412273928005647	-0.12862461222076968	0.5675744466569178	-0.04913497044082044	1.0

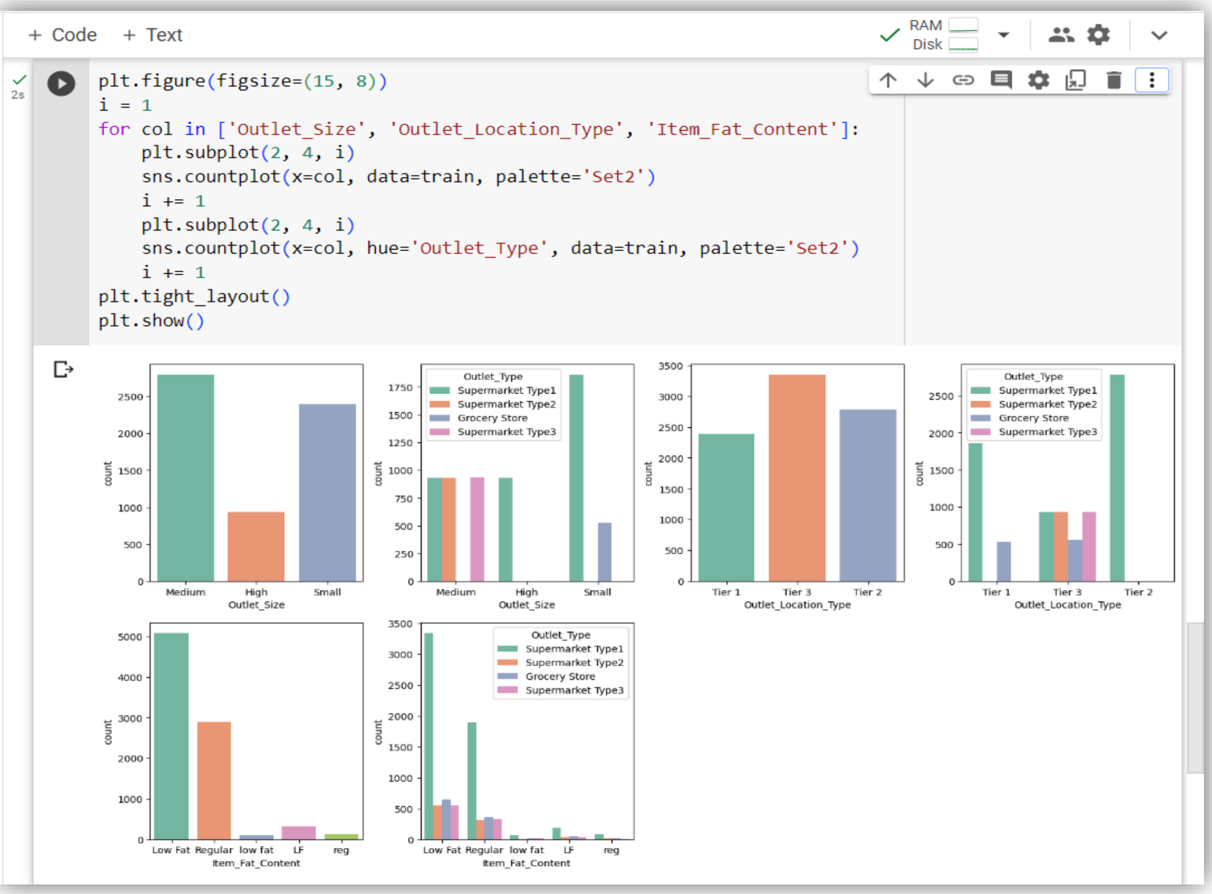




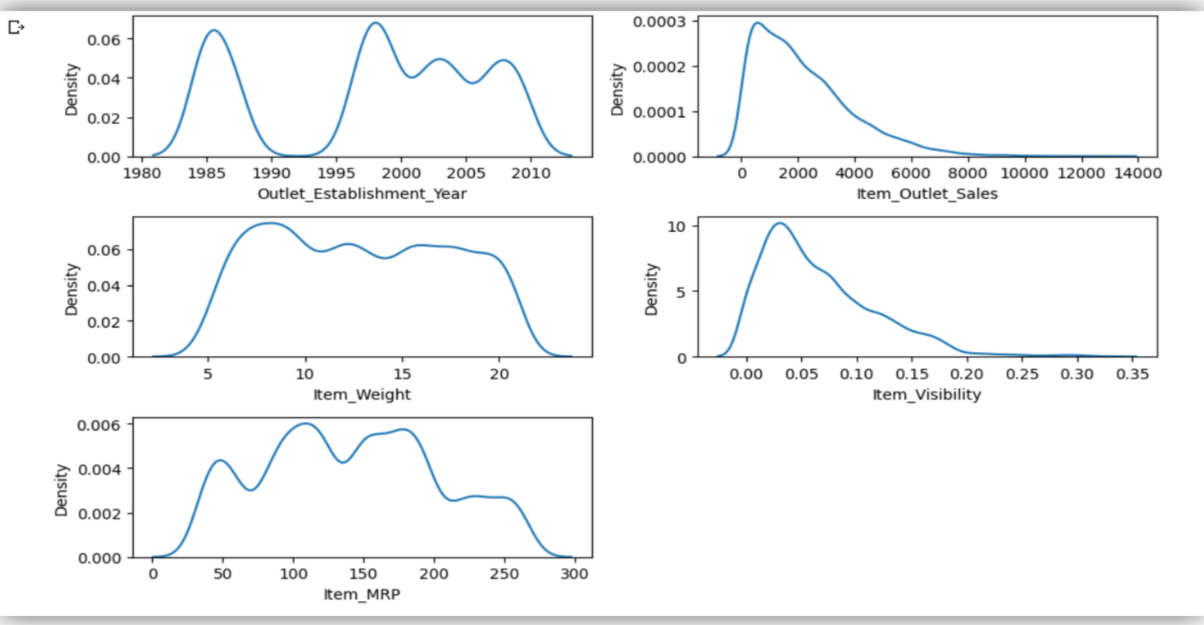
```
object=train.select_dtypes(include='object').columns
object
```

```
Index(['Item_Identifier', 'Item_Fat_Content', 'Item_Type', 'Outlet_Identifier',
       'Outlet_Size', 'Outlet_Location_Type', 'Outlet_Type'],
      dtype='object')
```

DATA VISUALISATION –



```
plt.figure(figsize=(15, 8))
i = 1
for col in ['Outlet_Size', 'Outlet_Location_Type', 'Item_Fat_Content']:
    plt.subplot(2, 4, i)
    sns.countplot(x=col, data=train, palette='Set2')
    i += 1
    plt.subplot(2, 4, i)
    sns.countplot(x=col, hue='Outlet_Type', data=train, palette='Set2')
    i += 1
plt.tight_layout()
plt.show()
```



5.0 BUSINESS OPPORTUNITY

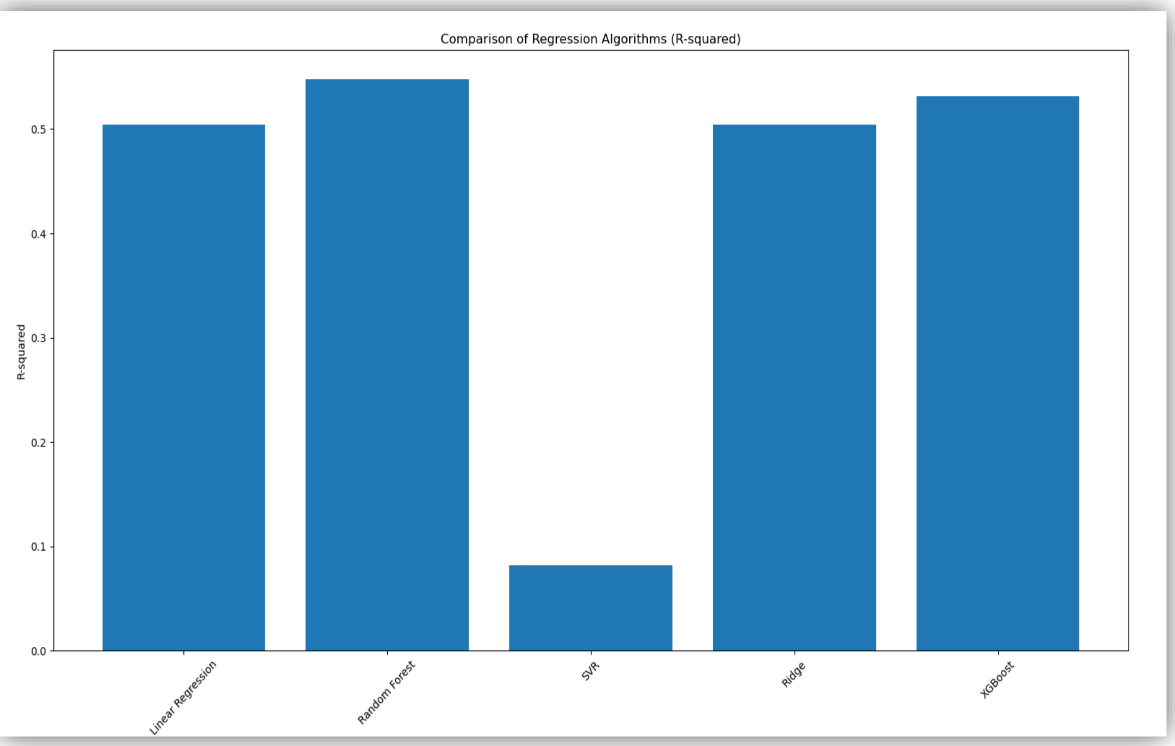
The BigMart Sales Prediction App presents a significant business opportunity for BigMart by enabling better inventory management, cost reduction, improved decision-making, competitive advantage, enhanced customer experience, scalability, and potential data monetization.

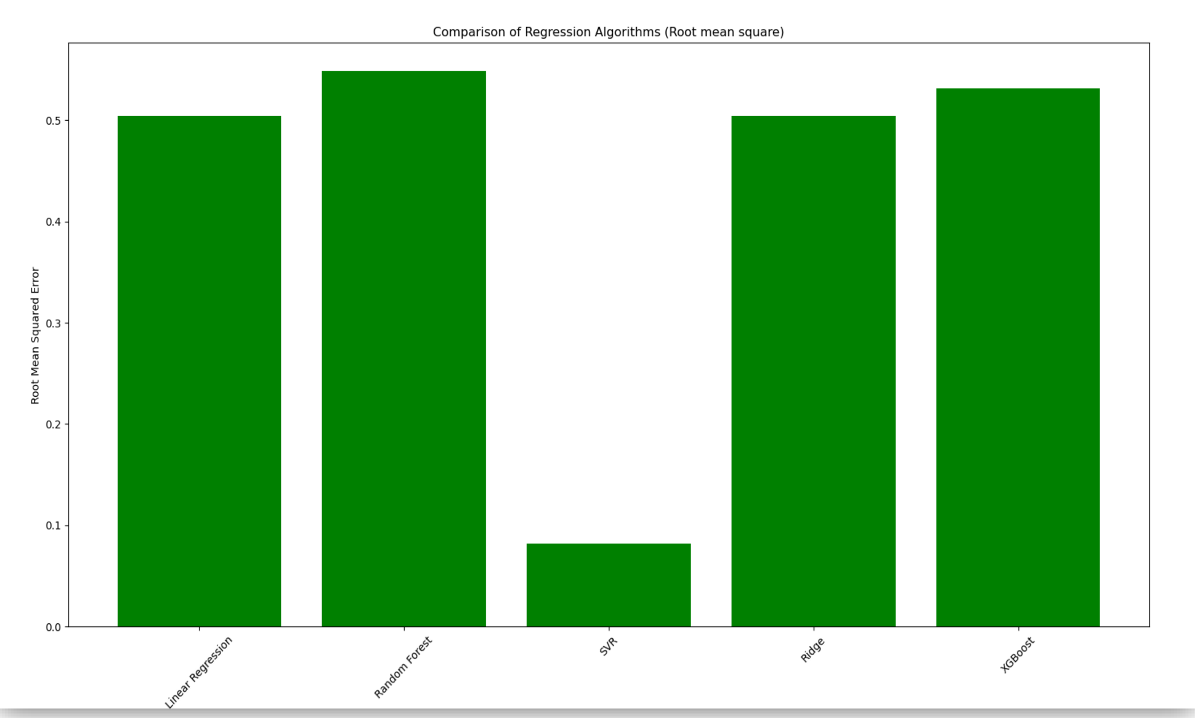
6.0 CONCEPT GENERATION

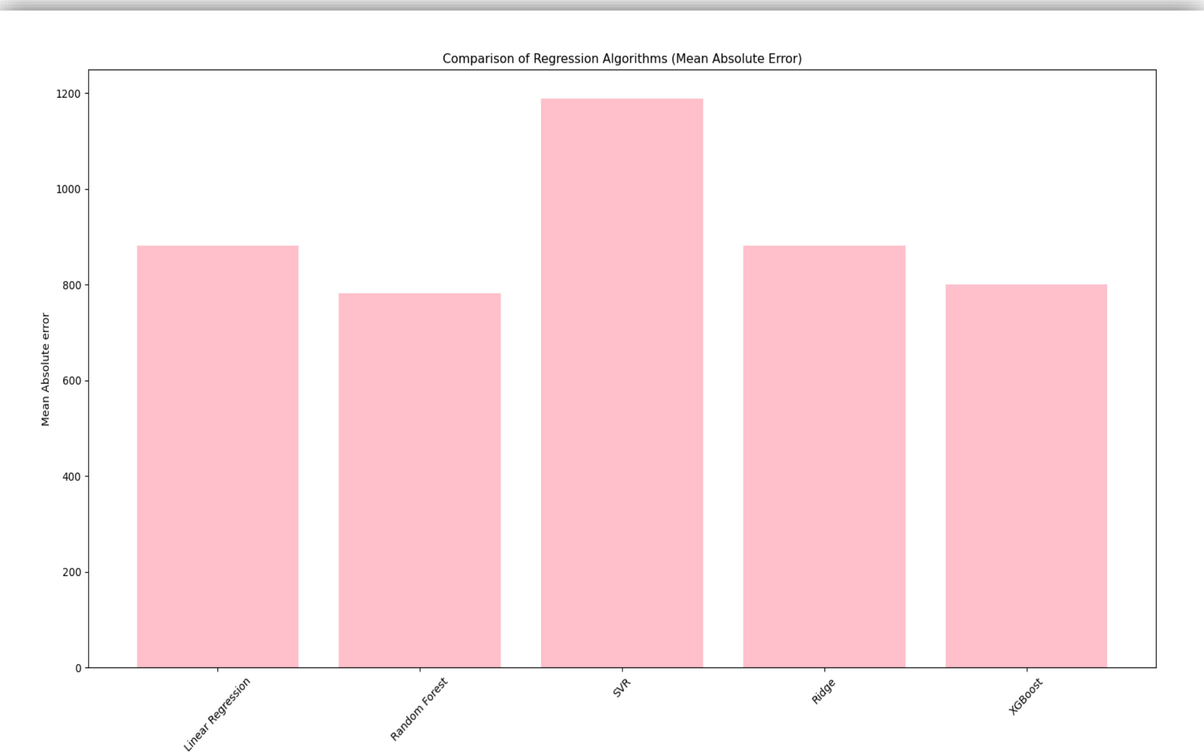
For the code implementation, I did a study for Big Mart Sales project using 5 algorithms –

- Linear Regression
- Random Forest Regression
- Ridge Regression
- Support Vector Regression
- XGBoost Regression

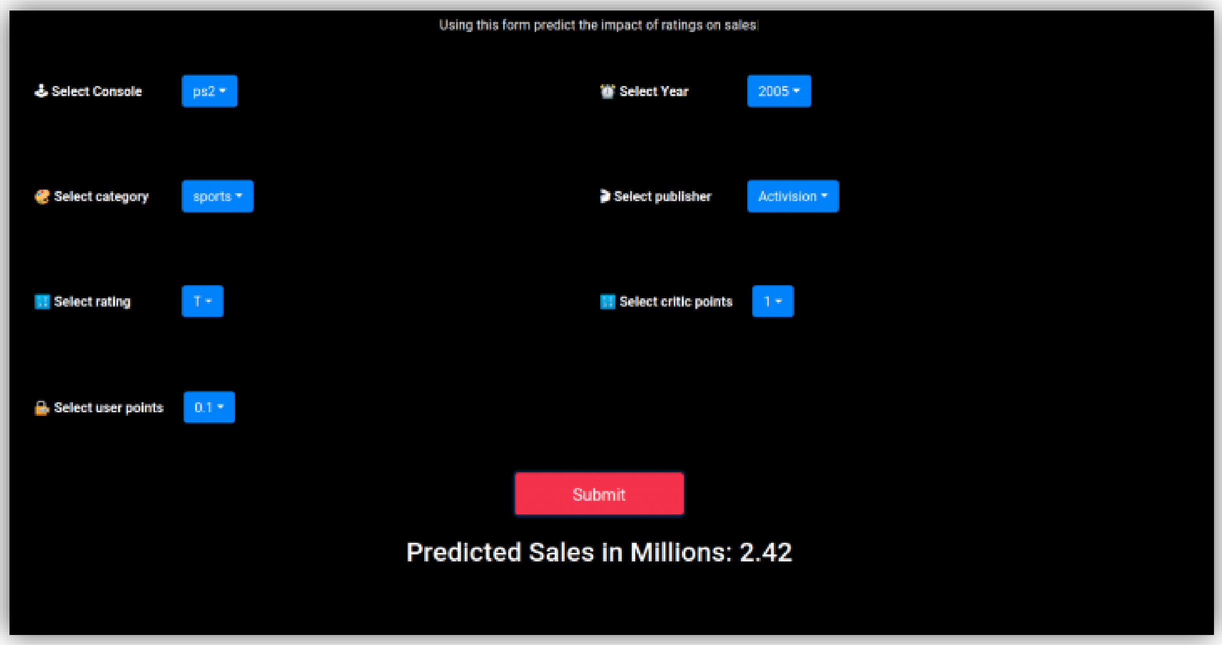
On a comparative study the following results were obtained –




Comparison of Regression Algorithms (Root mean square)
Root Mean Squared Error
0.0
0.1
0.2
0.3
0.4
0.5
Linear Regression
Random Forest
SVR
Ridge
XGBoost


Comparison of Regression Algorithms (Mean Absolute Error)
Mean Absolute error
0
200
400
600
800
1000
1200
Linear Regression
Random Forest
SVR
Ridge
XGBoost

7.0 CONCEPT DEVELOPMENT



Using this form predict the impact of ratings on sales

Select Console: ps2

Select Year: 2005

Select category: sports

Select publisher: Activision

Select rating: T

Select critic points: 1

Select user points: 0.1

Predicted Sales in Millions: 2.42

8.0 FINAL REPORT PROTOTYPE

Front-end Development:

1. Choose a suitable front-end technology stack such as HTML, CSS, and JavaScript or consider using front-end frameworks like React, Angular, or Vue.js.
2. Design the user interface (UI) of the app, including layouts, forms, and visual elements.
3. Implement the UI using your chosen front-end technology and ensure a user-friendly and intuitive experience.
4. Connect the front-end to the back-end API to fetch and display data.

Back-end Development:

1. Select an appropriate back-end technology stack such as Python with frameworks like Flask or Django, Node.js with Express.js, or any other server-side language or framework you are comfortable with.
2. Create the necessary routes or endpoints to handle incoming requests from the front-end and provide the required responses.
3. Implement the logic to interact with the machine learning model for sales prediction.
4. Integrate with a database, if needed, to store and retrieve data for the app.
5. Ensure the back-end is secure and handles input validation, authentication, and authorization if required.

Integration:

1. Connect the front-end and back-end by making API requests from the front-end to the appropriate endpoints on the back-end.

2. Implement data exchange between the front-end and back-end, such as sending user input from the front-end to the back-end for prediction and receiving the prediction results back to display on the front-end.

Testing and Deployment:

1. Conduct thorough testing of both the front-end and back-end components to ensure functionality, performance, and user experience.
2. Deploy the back-end API on a suitable hosting platform or server.
3. Continuously monitor and maintain the deployed app, addressing any issues or bugs that may arise.

9.0 CODE IMPLEMENTATION

I have implemented the code and can be evaluated through the github link – https://github.com/Ghimanshigit03/BigMarketSalesPrediction

10.0 CONCLUSION

In this work, the effectiveness of various algorithms on the data on revenue and review of, best performance-algorithm, here propose a software to using regression approach for predicting the sales centred on sales data from the past the accuracy of linear regression prediction can be enhanced with this method. So, we can conclude Random Forest and Xgboost regression gives the better prediction with respect to Accuracy, MAE and RMSE.

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