A Project Report

on

<< **Sales prediction in Big Mart** >>

to be submitted in partial fulfilling of the requirements for the course on

Fundamentals of Data Analytics – CSC3005

(E1 SLOT)

by

<<Priyanka B>>   <<20BCS0124>>

<<Geetha P>>   <<20BCS0132>>

<<Sharmili P >>   <<20BCS0134>>

<<Sharmila D>>   <<20BCS0168>>

Fall Semester 2022-2023

TABLE OF CONTENTS

ABSTRACT

Introduction ………………………………………………………………. 01

Review 1 (Survey, Analysis)...…………………………………….……… 04

Review 2 (Requirement gathering & Prototype Design)……………….. 06

Review 3 (Evaluation)....…………………….…………………………… 08

Conclusion …………………………………………………...…… 11

References ……………………………………………………………. 12

ABSTRACT

Machine Learning is a category of algorithms that allows software applications to become more accurate in predicting outcomes without being explicitly programmed. The basic premise of machine learning is to build models and employ algorithms that can receive input data and use statistical analysis to predict an output while updating outputs as new data becomes available. These models can be applied in different areas and trained to match the expectations of management so that accurate steps can be taken to achieve the organization's target. In this paper, the case of Big Mart, a one-stop-shopping-center, has been discussed to predict the sales of different types of items and for understanding the effects of different factors on the items' sales. Taking various aspects of a dataset collected for Big Mart, and the methodology followed for building a predictive model, results with high levels of accuracy are generated, and these observations can be employed to take decisions to improve sales.

Keywords: Machine Learning, Sales Prediction, Big Mart, Random Forest, Linear Regression

1. INTRODUCTION

In today's modern world, huge shopping centers such as big malls and marts are recording data related to sales of items or products with their various dependent or independent factors as an important step to be helpful in prediction of future demands and inventory management. The dataset built with various dependent and independent variables is a composite form of item attributes, data gathered by means of customer, and also data related to inventory management in a data SALES PREDICTION 4 warehouse. The data is thereafter refined in order to get accurate predictions and gather new as well as interesting results that shed a new light on our knowledge with respect to the task's data. This can then further be used for forecasting future sales by means of employing machine learning algorithms such as the random forests and simple or multiple linear regression model.

2. REVIEW-1 (Survey & Analysis)

The data available is increasing day by day and such a huge amount of unprocessed data is needed to be analysed precisely, as it can give very informative and finely pure gradient results as per current standard requirements. It is not wrong to say as with the evolution of Artificial Intelligence (AI) over the past two decades, Machine Learning (ML) is also on a fast pace for its evolution. ML is an important mainstay of IT sector and with that, a rather central, albeit usually hidden, part of our life. As the technology progresses, the analysis and understanding of data to give good results will also increase as the data is very useful in current aspects. In machine learning, one deals with both supervised and unsupervised types of tasks and generally a classification type problem accounts as a resource for knowledge discovery. It generates resources and employs regression to make precise predictions about future, the main emphasis being laid on making a system self-efficient, to be able to do computations and analysis to generate much accurate and precise results.
SALES PREDICTION 5
By using statistic and probabilistic tools, data can be converted into knowledge. The statistical inferencing uses sampling distributions as a conceptual key.
ML can appear in many guises. In this paper, firstly, various applications of ML and the types of data they deal with are discussed. Next, the problem statement addressed through this work is stated in a formalized way. This is followed by explaining the methodology ensued and the prediction results observed on implementation. Various machine learning algorithms include:

• Linear Regression: It can be termed as a parametric technique which is used to predict a continuous or dependent variable on basis of a provided set of independent variables. This technique is said to be parametric as different assumptions are made on basis of data set.

• K-Nearest Neighbors (KNN): It is a learning algorithm which is based on instances and knowledge gained through them. Unlike mining in data stream scenarios, cases where every sample can simultaneously belong to multiple classes in hierarchical multi-label classification problems, k-NN is being proposed to be applied to predict outputs in structured form.

• Decision tree: It is an intuitive model having low bias and it can be adopted to build a classification tree with root node being the first to be taken into account in a top-down manner. It is a classic model for machine learning.

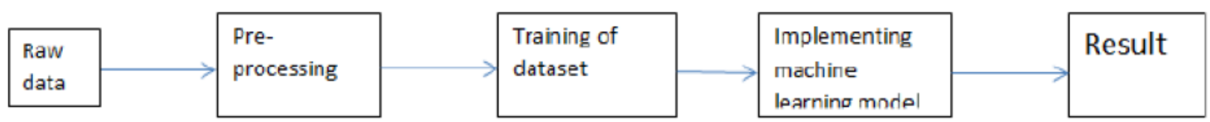
• Naïve Bayes classifiers: These are based on Bayes theorem and a collection of classification algorithms where classification of every pair is independent of each other. Bayesian learning can provide SALES PREDICTION 6 predictions with readable reasons by generating an if-then form of list of rules.

• Random Tree: It is an efficient algorithm for achieving scalability and is used in identification problems for building approximate system. The decisions are taken considering the choices made on basis of possible consequences, the variables which are included, input factor. Other algorithms can include SVM, xgboost, logistic regression and so on.

• K-means clustering: This algorithm is used in unsupervised learning for creating clusters of related data based on their closeness to the centroid value.

3. REVIEW-2 (Requirement gathering & Prototype Design)

To find out what role certain properties of an item play and how they affect their sales by understanding Big Mart sales." In order to help Big Mart achieve this goal, a predictive model can be built to find out for every store, the key factors that can increase their sales and what changes could be made to the product or store's characteristics.



Python is a general purpose, interpreted-high level language used extensively nowadays for solving domain problems instead of dealing with complexities of a system. It is also termed as the 'batteries included language' for programming. It has various libraries used for scientific purposes and inquiries along with number of third-party libraries for making problem solving efficient.

In this work, the Python libraries of Numpy, for scientific computation, and Matplotlib, for 2D plotting have been used. Along with this, Pandas tool of Python has been employed for carrying out data analysis. Random forest regressor is used to solve tasks by ensembling random forest method. As a development platform, Jupyter Notebook, which proves to work great due to its excellence in 'literate programming', where human friendly code is punctuated within code blocks, has been used.

4. REVIEW-3 (Evaluation)

In this section, the programming language, libraries, implementation platform along with the data modeling and the observations and results obtained from it are discussed. SALES PREDICTION 8

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.preprocessing import LabelEncoder

from sklearn.model_selection import train_test_split

from xgboost import XGBRegressor

from sklearn import metric

DATA COLLECTION AND PROCESSING

loading the data from csv file to Pandas DataFrame

big_mart_data = pd.read_csv('/content/Train.csv')

first 5 rows of the dataframe

big_mart_data.head()

	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	9.30	Low Fat	0.016047	Dairy	249.8092	OUT049	1999	Medium	Tier 1	Supermarket Type1	3735.1380
1	DRC01	5.92	Regular	0.019278	Soft Drinks	48.2692	OUT018	2009	Medium	Tier 3	Supermarket Type2	443.4228
2	FDN15	17.50	Low Fat	0.016760	Meat	141.6180	OUT049	1999	Medium	Tier 1	Supermarket Type1	2097.2700
3	FDX07	19.20	Regular	0.000000	Fruits and Vegetables	182.0950	OUT010	1998	NaN	Tier 3	Grocery Store	732.3800
4	NCD19	8.93	Low Fat	0.000000	Household	53.8614	OUT013	1987	High	Tier 3	Supermarket Type1	994.7052

number of data points & number of features

big_mart_data.shape

```
(8523, 12)
```

getting some information about thye dataset

Big_mart data.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
```

CATEGORICAL FEATURES:

• Item_Identifier

• Item_Fat_Content

• Item_Type

• Outlet_Identifier

• Outlet_Size

• Outlet_Location_Type

• Outlet_Type

SALES PREDICTION 10

checking for missing values

big_mart_data.isnull().sum()

```
Item_Identifier                 0
Item_Weight                  1463
Item_Fat_Content                0
Item_Visibility                 0
Item_Type                       0
Item_MRP                        0
Outlet_Identifier               0
Outlet_Establishment_Year       0
Outlet_Size                  2410
Outlet_Location_Type            0
Outlet_Type                     0
Item_Outlet_Sales               0
dtype: int64
```

mean value of "Item_Weight" column

big_mart_data['Item_Weight'].mean()

```
12.857645184136183
```

filling the missing values in "Item_weight column" with "Mean" value

big_mart_data['Item_Weight'].fillna(big_mart_data['Item_Weight'].mean(), inplace=True)

mode of "Outlet_Size" column

big_mart_data['Outlet_Size'].mode()

```
0    Medium
dtype: object
```


filling the missing values in "Outlet_Size" column with Mode

mode_of_Outlet_size=big_mart_data.pivot_table(values='Outlet_Size', columns='Outlet_Type', aggfunc=(lambda x: x.mode()[0]))

print(mode_of_Outlet_size)

Outlet_Type	Grocery Store	Supermarket Type1	Supermarket Type2	Supermarket Type3
Outlet_Size	Small	Small	Medium	Medium

miss_values = big_mart_data['Outlet_Size'].isnull()

print(miss_values)

```
0       False
1       False
2       False
3        True
4       False
        ...
8518    False
8519     True
8520    False
8521    False
8522    False
Name: Outlet_Size, Length: 8523, dtype: bool
```

big_mart_data.loc[miss_values,'Outlet_Size'] = big_mart_data.loc[miss_values,'Outlet_Type'].apply(lambda x: mode_of_Outlet_size[x])

checking for missing values

big_mart_data.isnull().sum()

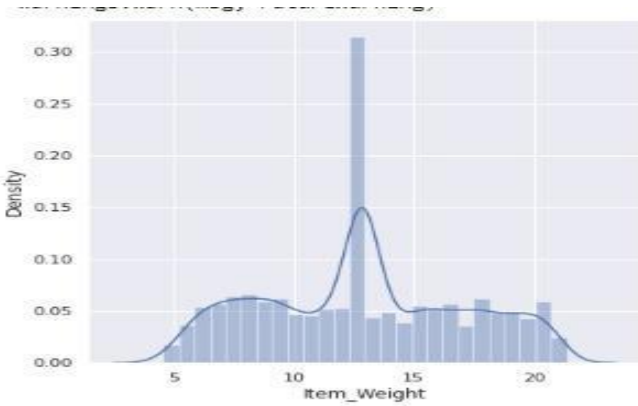
```
Item_Identifier              0
Item_Weight                  0
Item_Fat_Content             0
Item_Visibility              0
Item_Type                    0
Item_MRP                     0
Outlet_Identifier            0
Outlet_Establishment_Year    0
Outlet_Size                  0
Outlet_Location_Type         0
Outlet_Type                  0
Item_Outlet_Sales            0
dtype: int64
```


big_mart_data.describe()

	Item_Weight	Item_Visibility	Item_MRP	Outlet_Establishment_Year	Item_Outlet_Sales
count	8523.000000	8523.000000	8523.000000	8523.000000	8523.000000
mean	12.857645	0.066132	140.992782	1997.831867	2181.288914
std	4.226124	0.051598	62.275067	8.371760	1706.499616
min	4.555000	0.000000	31.290000	1985.000000	33.290000
25%	9.310000	0.026989	93.826500	1987.000000	834.247400
50%	12.857645	0.053931	143.012800	1999.000000	1794.331000
75%	16.000000	0.094585	185.643700	2004.000000	3101.296400
max	21.350000	0.328391	266.888400	2009.000000	13086.964800

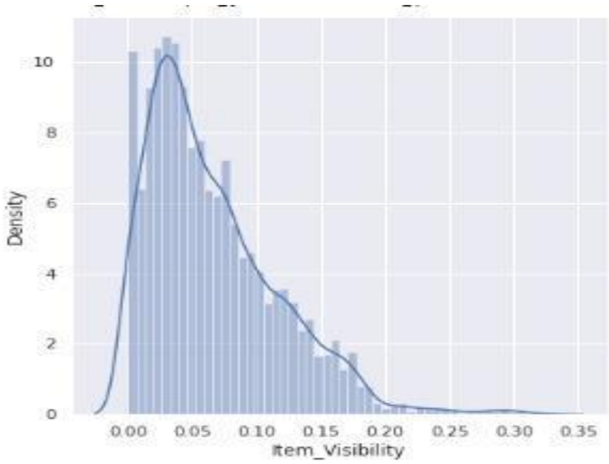
NUMERICAL FEATURES

```
sns.set()
# Item_Weight distribution
plt.figure(figsize=(6,6))
sns.distplot(big_mart_data['Item_Weight'])
plt.show()
```



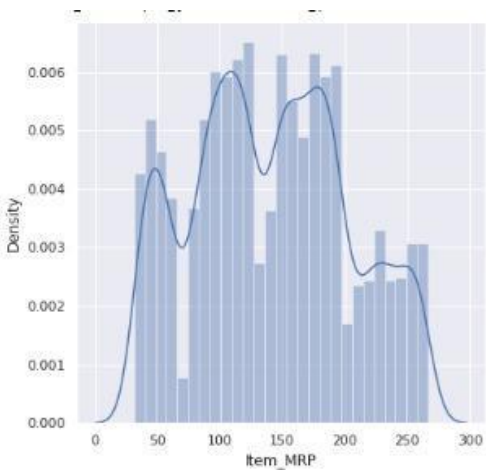
SALES PREDICTION 13

```
# Item Visibility distribution
plt.figure(figsize=(6,6))
sns.distplot(big_mart_data['Item_Visibility'])
plt.show()
```

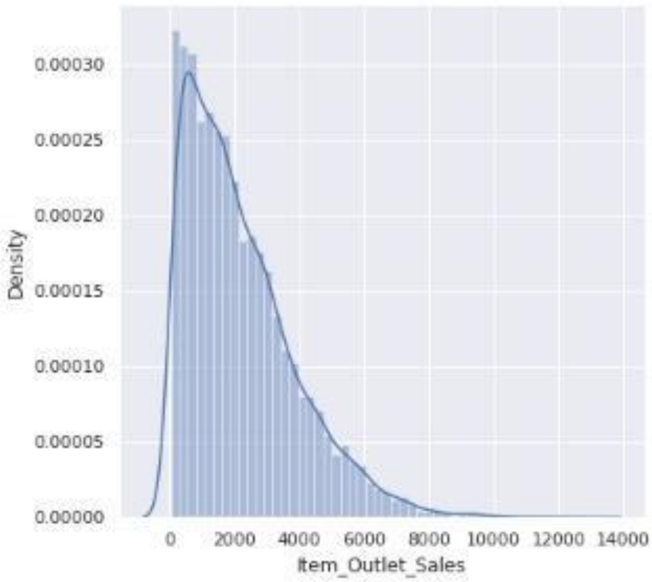


SALES PREDICTION 14

```
# Item MRP distribution
plt.figure(figsize=(6,6))
sns.distplot(big_mart_data['Item_MRP'])
plt.show()
```

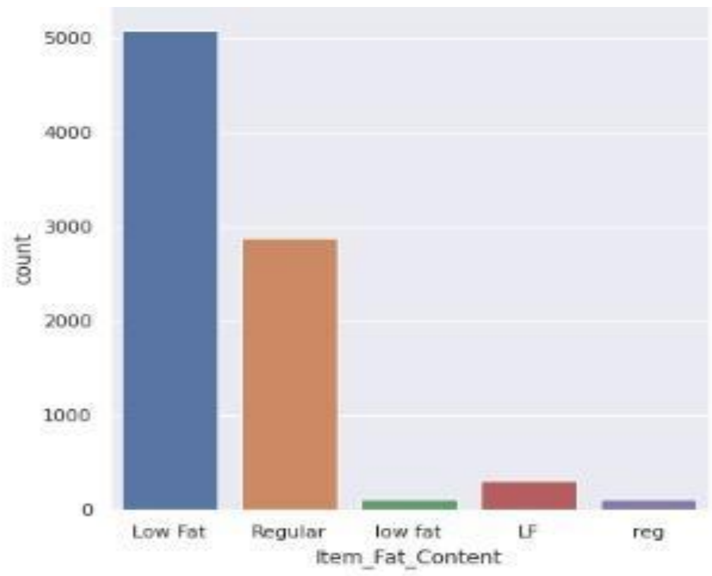



```
# Item_Outlet_Sales distribution
plt.figure(figsize=(6,6))
sns.distplot(big_mart_data['Item_Outlet_Sales'])
plt.show()
```



SALES PREDICTION 15

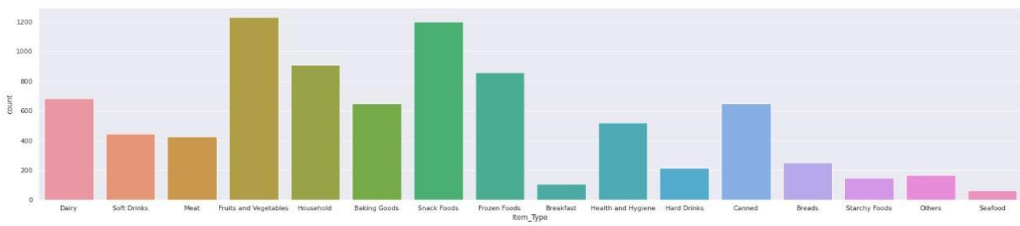
```
# Item_Fat_Content column
plt.figure(figsize=(6,6))
sns.countplot(x='Item_Fat_Content', data=big_mart_data)
plt.show()
```



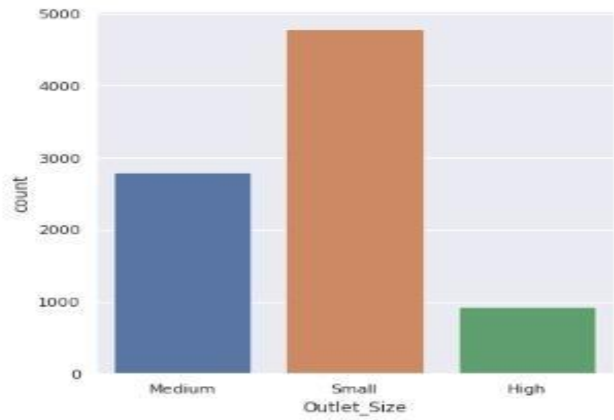
SALES PREDICTION 16

```
# Item_Type column
plt.figure(figsize=(30,6))
```

```
sns.countplot(x='Item_Type', data=big_mart_data)
plt.show()
```




```
# Outlet_Size column
plt.figure(figsize=(6,6))
sns.countplot(x='Outlet_Size', data=big_mart_data)
plt.show()
```



SALES PREDICTION 17

DATA PRE-PROCESSING

```
big_mart_data.head()
```

	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	9.30	Low Fat	0.016047	Dairy	249.8092	OUT049	1999	Medium	Tier 1	Supermarket Type1	3735.1380
1	DRC01	5.92	Regular	0.019278	Soft Drinks	48.2692	OUT018	2009	Medium	Tier 3	Supermarket Type2	443.4228
2	FDN15	17.50	Low Fat	0.016760	Meat	141.6180	OUT049	1999	Medium	Tier 1	Supermarket Type1	2097.2700
3	FDX07	19.20	Regular	0.000000	Fruits and Vegetables	182.0950	OUT010	1998	Small	Tier 3	Grocery Store	732.3800
4	NCD19	8.93	Low Fat	0.000000	Household	53.8614	OUT013	1987	High	Tier 3	Supermarket Type1	994.7052

```
big_mart_data['Item_Fat_Content'].value_counts()
```

```
Low Fat     5089
Regular     2889
LF           316
reg          117
low fat      112
Name: Item_Fat_Content, dtype: int64
```


big_mart_data.replace({'Item_Fat_Content':{'low fat':'Low Fat','LF':'Low Fat', 'reg':'Regular'}}, inplace=True)

big_mart_data['Item_Fat_Content'].value_counts()

```
Low Fat     5517
Regular     3006
Name: Item_Fat_Content, dtype: int64
```

LABEL ENCODING

encoder = LabelEncoder()

big_mart_data['Item_Identifier'] = encoder.fit_transform(big_mart_data['Item_Identifier'])

big_mart_data['Item_Fat_Content'] = encoder.fit_transform(big_mart_data['Item_Fat_Content'])

big_mart_data['Item_Type'] = encoder.fit_transform(big_mart_data['Item_Type'])

big_mart_data['Outlet_Identifier'] = encoder.fit_transform(big_mart_data['Outlet_Identifier'])

big_mart_data['Outlet_Size'] = encoder.fit_transform(big_mart_data['Outlet_Size'])

big_mart_data['Outlet_Location_Type'] = encoder.fit_transform(big_mart_data['Outlet_Location_Type'])

big_mart_data['Outlet_Type'] = encoder.fit_transform(big_mart_data['Outlet_Type'])

big_mart_data.head()

	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	156	9.30	0	0.016047	4	249.8092	9	1999	1	0	1	3735.1380
1	8	5.92	1	0.019278	14	48.2692	3	2009	1	2	2	443.4228
2	662	17.50	0	0.016760	10	141.6180	9	1999	1	0	1	2097.2700
3	1121	19.20	1	0.000000	6	182.0950	0	1998	2	2	0	732.3800
4	1297	8.93	0	0.000000	9	53.8614	1	1987	0	2	1	994.7052

SPLITTING FEATURES AND TARGET

X = big_mart_data.drop(columns='Item_Outlet_Sales', axis=1)

Y = big_mart_data['Item_Outlet_Sales']

print(X)

```
      Item_Identifier  Item_Weight  ...  Outlet_Location_Type  Outlet_Type
0                 156        9.300  ...                     0            1
1                   8        5.920  ...                     2            2
2                 662       17.500  ...                     0            1
3                1121       19.200  ...                     2            0
4                1297        8.930  ...                     2            1
...               ...          ...  ...                   ...          ...
8518              370        6.865  ...                     2            1
8519              897        8.380  ...                     1            1
8520             1357       10.600  ...                     1            1
8521              681        7.210  ...                     2            2
8522               50       14.800  ...                     0            1

[8523 rows x 11 columns]
```

print(Y)

```
0       3735.1380
1        443.4228
2       2097.2700
3        732.3800
4        994.7052
          ...
8518    2778.3834
8519     549.2850
8520    1193.1136
8521    1845.5976
8522     765.6700
Name: Item_Outlet_Sales, Length: 8523, dtype: float64
```


SPLITTING THE DATA INTO TRAINING DATA & TESTING DATA

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=2)

```
print(X.shape, X_train.shape, X_test.shape)
```

```
(8523, 11) (6818, 11) (1705, 11)
```

MACHINE LEARNING MODEL TRAINING

XGBOOST REGRESSOR

```
regressor = XGBRegressor()
regressor.fit(X_train, Y_train)
```

EVALUATION

```
# prediction on training data
training_data_prediction = regressor.predict(X_train)
# R squared Value
r2_train = metrics.r2_score(Y_train, training_data_prediction)
print('R Squared value = ', r2_test
```

```
R Squared value =  0.6364457030941357
```

```
# prediction on test data

test_data_prediction = regressor.predict(X_test)

# R squared Value

r2_test = metrics.r2_score(Y_test, test_data_prediction)

print('R Squared value = ', r2_test)
```

```
R Squared value =  0.5867640914432671
```

5. CONCLUSION

Sales prediction is a critical part of the strategic planning process and allows a company to predict how their company will perform in the future. It allows them to not only plan for new opportunities, but also allows them to avert negative trends that appear in the forecast. A mission statement is important because it allows an organization to know exactly why they exist and serves as a guide for decisions. Both concepts are important to the success of the company and should not be overlooked throughout the strategic planning process.

6. REFERENCES

https://youtu.be/epI9W3MZ3Ts

https://www.embedded-robotics.com/forecast-sales-using-machine-learning/

https://rpubs.com/ngwx/819233

https://thecleverprogrammer.com/2022/03/01/future-sales-prediction-with-machine-learning/

https://www.netsuite.com/portal/resource/articles/financial-management/predictive-modeling.shtml

https://www.analyticsvidhya.com/blog/2020/08/building-sales-prediction-web-application-using-machine-learning-dataset/

https://www.researchgate.net/publication/344099746_SALES_PREDICTION_MODEL_FOR_BIG_MART

https://www.opensourceforu.com/2021/01/using-python-to-predict-sales/

https://www.academia.edu/43174670/Sales_Analysis_and_Prediction_Using_Python

--End--