• DATA ENGINEERING PROJECT REPORT

Project Title: **BIG MART SALES PREDICTION USING MACHINE LEARNING MODEL.**

Student Name: **HEENA SALIM SHAIKH.** (ROLL NO: 38)

Course Name: BSC DATA SCIENCE.

Instructor Name: MITHILESH CHAUHAN.

Date: 27-08-2024

• INTRODUCTION OBJECTIVES

OBJECTIVES:

-To Develop a predictive model for forecasting sales of Big Mart's various products and store locations with high accuracy.

-This objective ensures that the prediction model addresses key business needs and provides valuable insights for decision-making.

-This model will utilize historical sales data, seasonal trends, promotional activities, and other relevant factors to generate actionable insights,

-Helps in enabling optimized inventory management, enhanced staffing efficiency, and targeted marketing strategies.

PROBLEM STATEMENT:

-Big Mart faces challenges in accurately forecasting sales for its diverse range of products across different store locations.

-The lack of precise sales predictions leads to inefficiencies in inventory management, overstocking or stock outs, suboptimal staffing, and ineffective marketing strategies.

-This project aims to analyse historical sales data and other influencing factors to develop a robust predictive model that provides reliable sales forecasts.

-The goal is to improve operational efficiency, reduce costs, and enhance customer satisfaction by ensuring that the right products are available at the right time and in the right quantities.

DATA DESCRIPTION:

-Description: Includes historical sales transactions, often segmented by product, region, and time period.

-Data Set from Kaggle.com for Big Mart sales of various different categories includes: ProductID, Weight, FatContent, ProductVisibility, ProductType, MRP, OutletID, EstablishementYear, Outletsize, LocationType, OutletType, OutletSales.

• METHODOLOGY

TOOLS USED:

- Python Language
- Google colab
- Libraries like:

‘ Pandas
‘Matplotlib
‘SKlearn

-Importing `seaborn for preprocessiong and training,testing data`

`-XGBoost for regression`

-Bar charts

STEPS TAKEN:

1. **Open Google colab create File name it**

Installing Libraries/Dependencies

Codes:

```
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 metrics
```

2. **Data Collection and Analysis:**

Codes:

```
#loading our csv file dataset to pandas Dataframe
big_mart_data = pd.read_csv('/content/Train.csv')
#first 5 rows of dataframe
big_mart_data.head()

#number of datapoints and number of features
big_mart_data.shape

#checking missing values
big_mart_data.isnull().sum()
```

3. **Handling Missing Values:**

Mean for average values, Mode for repetative values

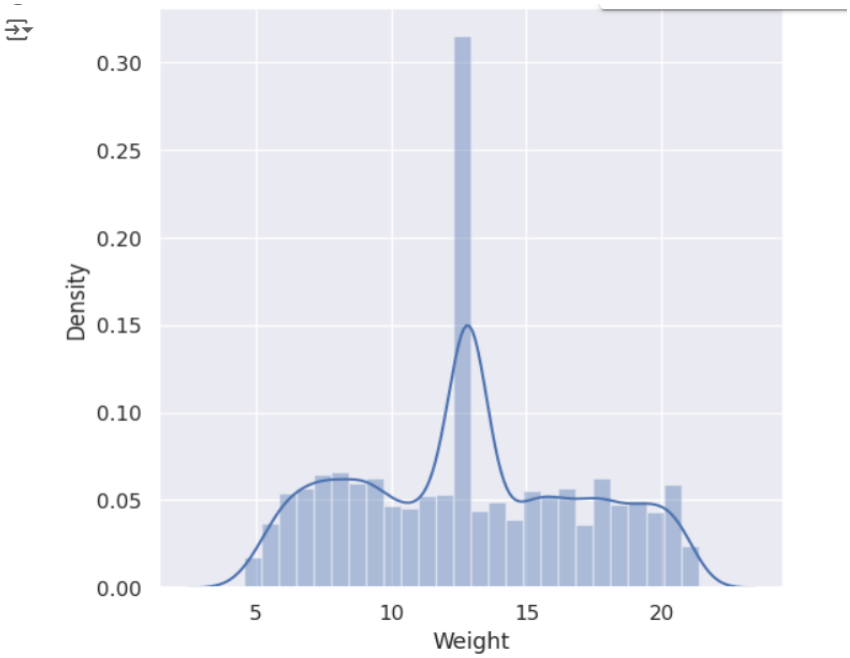
Codes:

```
#mean value for "weight column"
big_mart_data['Weight'].mean()

#filing the missing values of "weight column"
big_mart_data['Weight'].fillna(big_mart_data['Weight'].mean(),
inplace=True)

#checking missing values again
big_mart_data.isnull().sum()

#Replacing the missing value for Outletsize with mode
mode_of_outlet_size = big_mart_data.pivot_table(values='OutletSize',
columns='OutletType', aggfunc=(lambda x: x.mode()[0]))
print(mode_of_outlet_size)

missing_values = big_mart_data['OutletSize'].isnull()
print(missing_values)

big_mart_data.loc[missing_values, 'OutletSize'] =
big_mart_data.loc[missing_values, 'OutletType'].apply(lambda x:
mode_of_outlet_size[x])

#checking missing values again
big_mart_data.isnull().sum()
```

4. **Analysis of data:**

Codes:

```
#statistical measuring of data of numerical values
big_mart_data.describe()
#Numerical Features
sns.set()
#for weight column distribution
plt.figure(figsize=(6,6))
sns.distplot(big_mart_data['Weight'])
plt.show()

#OUTPUT
```



We can do this same step for every categories includes: OutletSales, ProductType, MRP etc.

5. DATA PREPROCESSING:

We will process data for FatContents as it is showing similar name multiple values

Codes:

```
big_mart_data.head()

big_mart_data['FatContent'].value_counts()

#OUTPUT
```

FatContent	count
Low Fat	5089
Regular	2889
LF	316
reg	117
low fat	112

dtype: int64

```
big_mart_data.replace({'FatContent': {'Low Fat':0, 'Regular':1, 'LF':0,
'reg':1, 'low fat':0}}, inplace=True)

big_mart_data['FatContent'].value_counts()

#OUTPUT AFTER NEW CHANGES
```

FatContent	count
0	5517
1	3006

dtype: int64

6. **Label Encoding:**

Codes:

```
encoder = LabelEncoder()

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

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

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

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

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

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

big_mart_data.head()
```

7. Splitting Features And Target:

Codes:

```
X = big_mart_data.drop(columns='OutletSales', axis=1)
Y = big_mart_data['OutletSales']
print(X)
print(Y)

#Splitting training and 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)

print(Y.shape, Y_train.shape, Y_test.shape)
```

8. TRANING MACHINE LEARNING MODEL:

XGboost regressor

Codes:

```
regressor = XGBRegressor()
X_train = X_train.drop('OutletID', axis=1)

regressor.fit(X_train, Y_train)
```

9. ERROR HANDLING:

TYPE The error message "ValueError: DataFrame.dtypes for data must be int, float, bool or category" indicates that the XGBoost regressor is encountering a column ('OutletID') in your training data (X_train) that has a data type ('object') it cannot handle. XGBoost primarily works with numerical and boolean data, and sometimes categorical data if explicitly enabled.

10. EVALUATING OUR MODEL:

Codes:

```
#Prediction based on Training Data
training_data_prediction = regressor.predict(X_train)
#R square values
r2_train = metrics.r2_score(Y_train, training_data_prediction)
print('R square value = ', r2_train)

#OUTPUT FOR TRAINING
R square value =  0.8737186757682781

# Create a copy of X_test before dropping 'OutletID' (if you haven't
already)AS before we were getting error
X_test_original = X_test.copy()
```

```
# Drop 'OutletID' from X_test if it exists
if 'OutletID' in X_test.columns:
    X_test = X_test.drop('OutletID', axis=1)

# Select the same columns in X_test as in X_train
X_test = X_test_original[X_train.columns]

# Now try predicting again
test_data_prediction = regressor.predict(X_test)

#R square values
r2_test = metrics.r2_score(Y_test, test_data_prediction)
print('R square value = ', r2_test)

#OUTPUT FOR TEST DATA
R square value =  0.5142059291987224
```

- # RESULTS

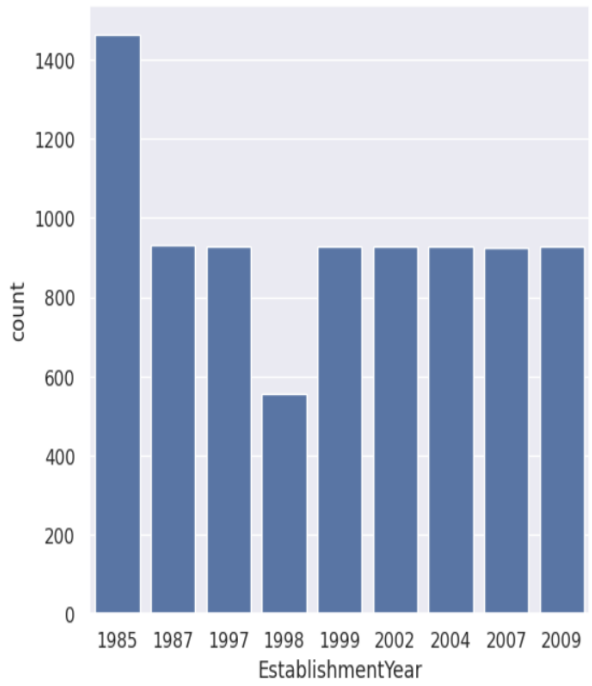
KEY FINDINGS:

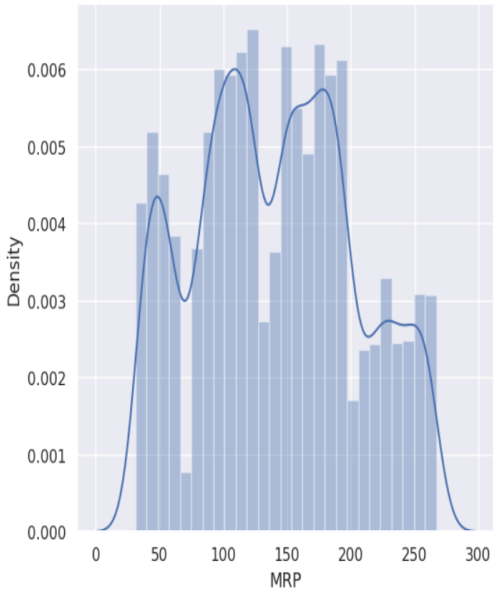
-The report identifies that higher MRP significantly boosts sales, with larger supermarkets outperforming smaller stores, particularly in Tier 1 cities.

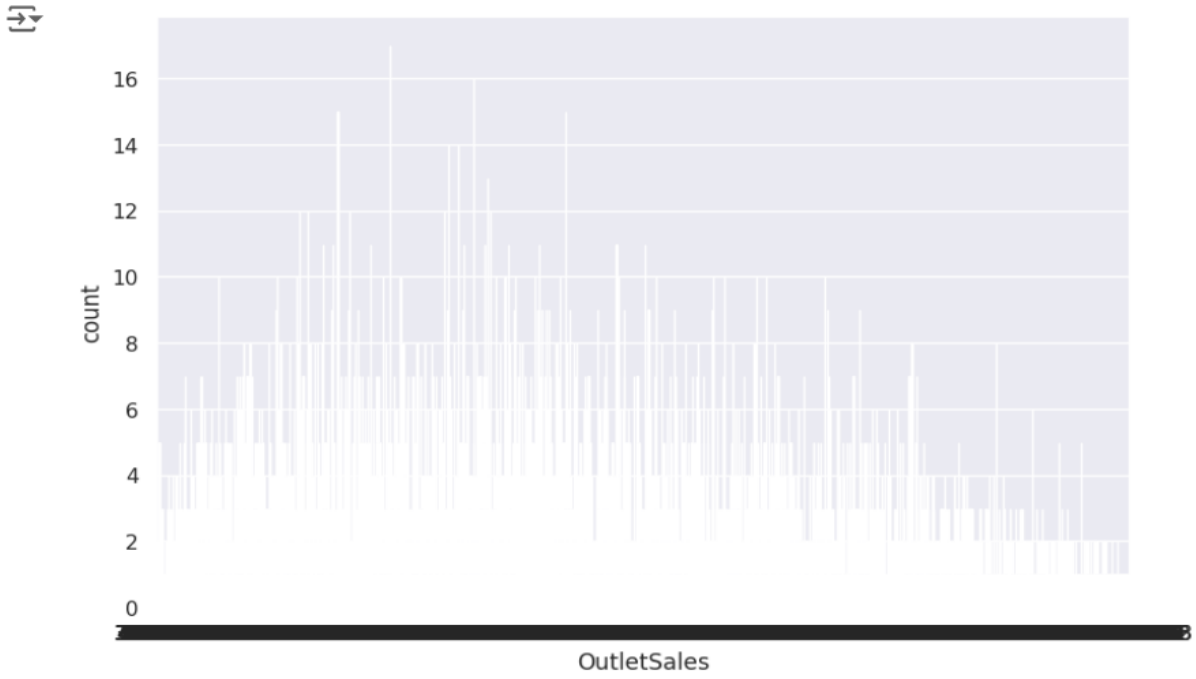
-While item visibility has a minimal impact, factors like product type and pricing are more crucial. Seasonal trends also play a vital role in driving sales, especially during holidays, underscoring the importance of targeted promotions and inventory strategies.

VISUALS:

-Histograms and Bar Charts







- **CONCLUSION**

SUMMARY:

-The Big Mart sales prediction project successfully identifies key drivers of sales, such as Item MRP, outlet type, and location. The analysis reveals that higher-priced items and larger supermarkets in urban areas consistently generate more sales.

- Seasonal trends and promotional strategies also play a crucial role in boosting sales during peak periods. The predictive model developed in this project offers valuable insights that can guide pricing strategies, inventory management, and marketing efforts, ultimately helping Big Mart optimize sales and enhance profitability across its stores.

SUGGESTIONS:

-Enhance Data Quality: Address missing values and inconsistencies in the dataset to improve model accuracy. Consider gathering more granular data, such as daily sales or customer demographics, to enrich the analysis.

-Incorporate External Factors: This can help in capturing more contextual insights.

-Advanced Modeling Techniques: Experiment with advanced machine learning algorithms like ensemble methods (e.g., XGBoost, Random Forest) or deep learning techniques to potentially increase prediction accuracy.

-Visual Dashboard: Develop an interactive dashboard for stakeholders to easily visualize sales trends, key metrics, and predictions. Tools like Tableau or Power BI could be useful for this purpose.

• REFERENCES

- Kaggle, Google, and Github