Capstone Project – Walmart Sales Prediction

Table of Contents

1. Problem Statement:

A retail store that has multiple outlets across the country are facing issues in managing the inventory - to match the demand with respect to supply. You are a data scientist, who has to come up with useful insights using the data and make prediction models to forecast the sales for X number of months/years.

1. Using the above data, come up with useful insights that can be used by each of the stores to improve in various areas.
2. Forecast the sales for each store for the next 12 weeks.

2. Project Objective:

The objectives of the project are:

1. To visualize which store has highest weekly sales.
2. To visualize the distribution of features.
3. To visualize which features are affecting the weekly sales.
4. To visualize the feature affecting the weekly sales per year and comparing them.
5. To find the best evaluation model.
6. To predict the model using prophet with regressors.
7. To verify the data obtained from prophet with the best evaluation model
8. To forecast the weekly sales for 12 weeks

3. Data Description

The available dataset is of Walmart sales which contains 6435 rows and 8 columns.

Feature Name	Description
Store	Store Number
Date	Week of Sales
Weekly_Sales	Sales for the given store in that week
Holiday_Flag	If it is a holiday week
Temperature	Temperature on the day of the sale
Fuel_price	Cost of the fuel in the region
CPI	Consumer Price Index
Unemployment	Unemployment Rate

4. Data Preprocessing Steps and Inspiration:

Data preprocessing is an essential step in preparing data for analysis and machine learning tasks. The preprocessing of the data included the following steps:

1. Importing libraries:

```
# importing Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
```

2. Loading Libraries:

```
#loading Dataset
df=pd.read_csv('Walmart (1).csv')
df
```

	Store	Date	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	CPI	Unemployment
0	1	05-02-2010	1643690.90	0	42.31	2.572	211.096358	8.106
1	1	12-02-2010	1641957.44	1	38.51	2.548	211.242170	8.106
2	1	19-02-2010	1611968.17	0	39.93	2.514	211.289143	8.106
3	1	26-02-2010	1409727.59	0	46.63	2.561	211.319643	8.106
4	1	05-03-2010	1554806.68	0	46.50	2.625	211.350143	8.106
...	...	...	...	...	...	...	...	...
6430	45	28-09-2012	713173.95	0	64.88	3.997	192.013558	8.684
6431	45	05-10-2012	733455.07	0	64.89	3.985	192.170412	8.667
6432	45	12-10-2012	734464.36	0	54.47	4.000	192.327265	8.667
6433	45	19-10-2012	718125.53	0	56.47	3.969	192.330854	8.667
6434	45	26-10-2012	760281.43	0	58.85	3.882	192.308899	8.667

6435 rows × 8 columns

3. Reframing the column in dataset:

   Reframing the ‘Date’ column into ‘weekday’, ‘month’ & ‘year’ for better evaluation.

```
#Reframing Date into weekday, month & year
df.Date=pd.to_datetime(df.Date)

df['weekday'] = df.Date.dt.weekday
df['month'] = df.Date.dt.month
df['year'] = df.Date.dt.year
df.drop(columns=['Date'],axis=1,inplace=True)
df.head()
```

	Store	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	CPI	Unemployment	weekday	month	year
0	1	1643690.90	0	42.31	2.572	211.096358	8.106	6	5	2010
1	1	1641957.44	1	38.51	2.548	211.242170	8.106	3	12	2010
2	1	1611968.17	0	39.93	2.514	211.289143	8.106	4	2	2010
3	1	1409727.59	0	46.63	2.561	211.319643	8.106	4	2	2010
4	1	1554806.68	0	46.50	2.625	211.350143	8.106	0	5	2010

4. Checking Null Values:

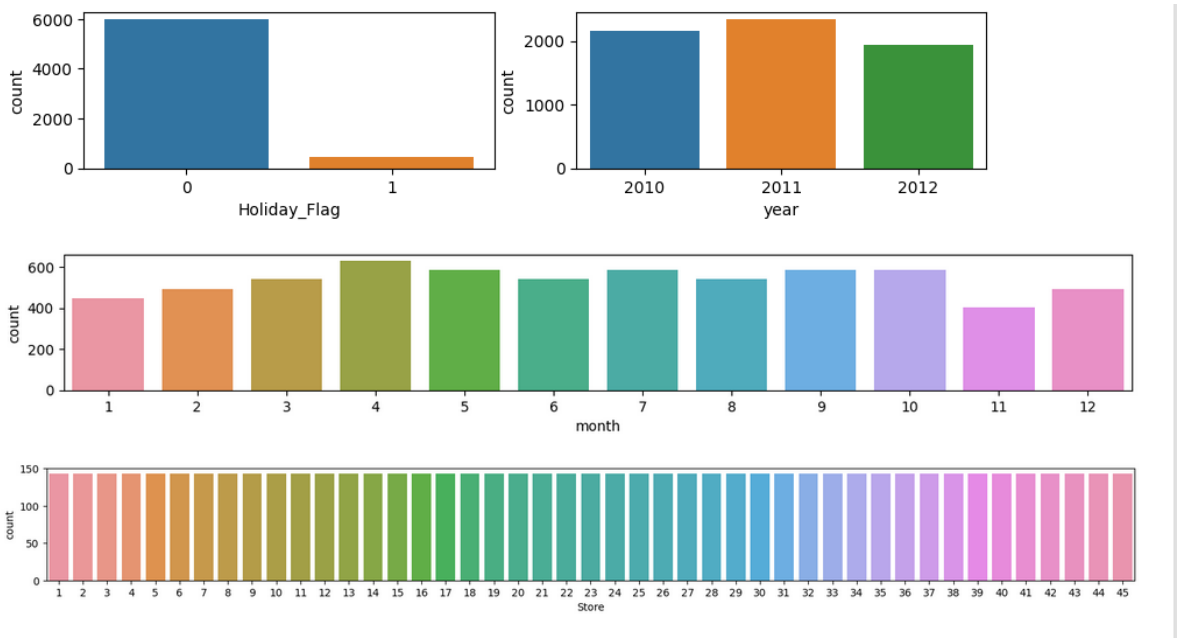
```
: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6435 entries, 0 to 6434
Data columns (total 10 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Store         6435 non-null   int64
 1   Weekly_Sales  6435 non-null   float64
 2   Holiday_Flag  6435 non-null   int64
 3   Temperature   6435 non-null   float64
 4   Fuel_Price    6435 non-null   float64
 5   CPI           6435 non-null   float64
 6   Unemployment  6435 non-null   float64
 7   weekday       6435 non-null   int64
 8   month         6435 non-null   int64
 9   year          6435 non-null   int64
dtypes: float64(5), int64(5)
memory usage: 502.9 KB
```

5. Handling Categorical Data:

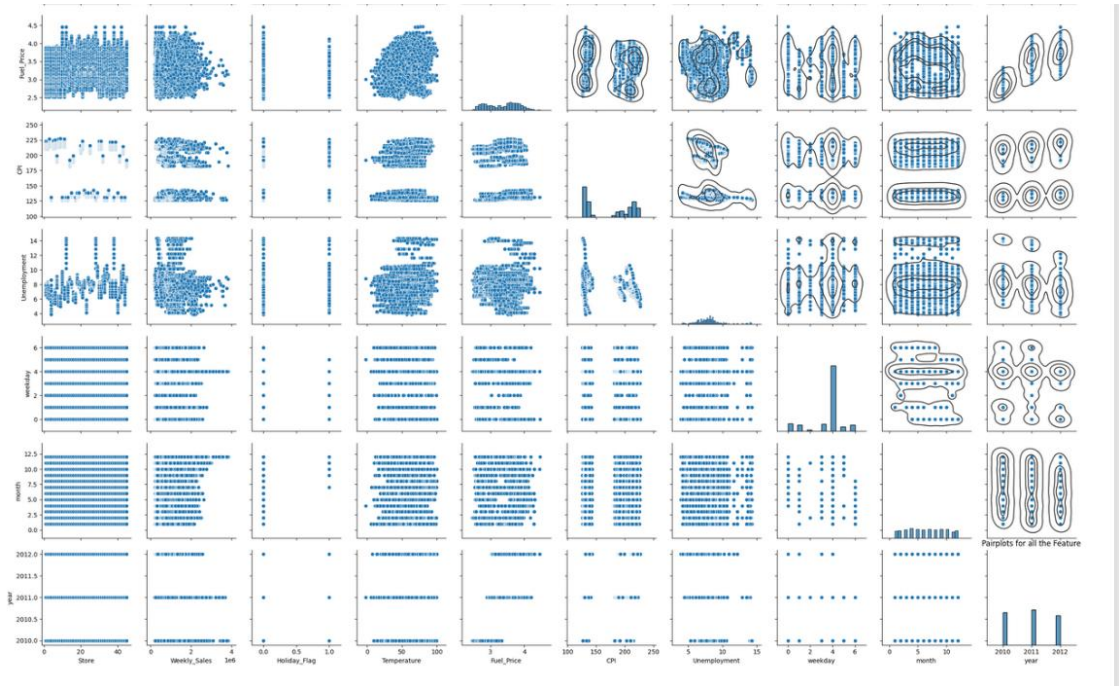
   In order to check the categorical features details, visualization helps. In the given data there are not much categorical features which affects the target variable.

```
# Visualizing the categorical features
plt.figure(figsize=[10,8])
plt.subplot(4,2,1)
sns.countplot(x=df['Holiday_Flag'],data=df)
plt.subplot(4,2,2)
sns.countplot(x=df['year'],data=df)

plt.figure(figsize=[30,8])
plt.subplot(4,2,3)
sns.countplot(x=df['month'],data=df)

plt.figure(figsize=[30,8])
plt.subplot(4,2,4)
sns.countplot(x=df['Store'],data=df)

plt.tight_layout()
```



6. Relationship between all the features:

```
#Understanding the relationship between all the features
g = sns.pairplot(df)
plt.title('Pairplots for all the Feature')
g.map_upper(sns.kdeplot, levels=4, color=".2")
plt.show()
```

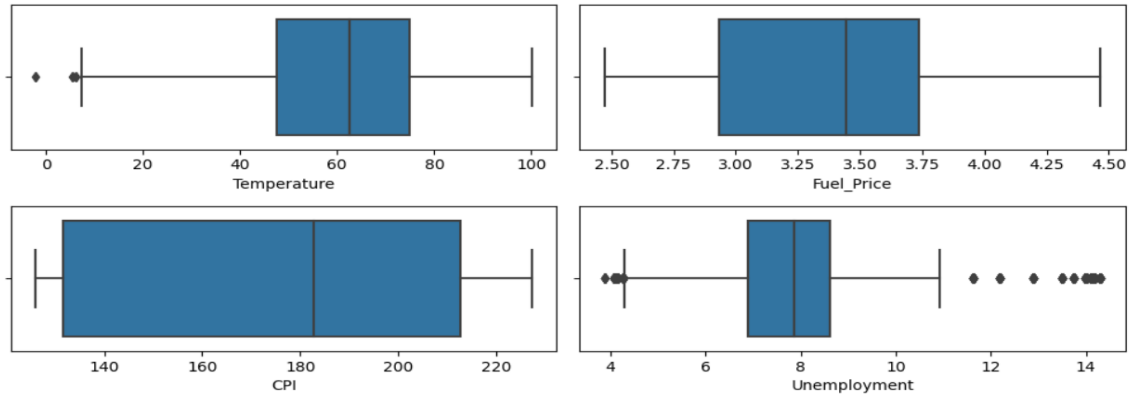


7. Handling outliers:

Outliers are seen in Temperature, Unemployment & Weekly_Sales features.

```
for i in df.columns:
    if df[i].dtypes=="float64" or df[i].dtypes=="int64":
        sns.boxplot(df[i])
        print(i)
        plt.show()
        plt.tight_layout()
```

These outliers have been removed using Interquartile Method.



8. Feature Selection:

Heatmap has been plotted for the feature selection in order to get the relationships between features in a given dataset.

```
# Feature Selection
plt.figure(figsize=(10,5))
sns.heatmap(df.corr(),annot=True,cmap='coolwarm')
plt.show()
```

9. Feature Scaling:

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

scaler.fit(df.drop('Weekly_Sales',axis=1))
```

5. Choosing the Algorithm for the Project: (Motivation and Reasons)

XGBoost Regressor can be a suitable algorithm for analyzing Walmart Sales data, It is specifically designed for regression tasks, where the goal is to predict continuous numerical values.

I have chosen XGBoost Regressor for this project because of the following reasons:

1. Gradient boosting: XGBoost uses a gradient boosting framework, which is a powerful technique that leverages the gradients of the loss function to improve model performance. It minimizes the loss function by optimizing the model's predictions.
2. Regularization: XGBoost provides various regularization techniques to prevent overfitting. These include L1 and L2 regularization terms, as well as a "gamma" parameter that controls the minimum loss reduction required for creating additional tree nodes.
3. Tree pruning: XGBoost employs a technique called tree pruning to reduce the complexity of individual decision trees. Pruning helps to prevent overfitting and improves generalization.
4. Handling missing values: XGBoost can handle missing values by learning how to best treat them during model training. It automatically learns the best direction to assign missing values in the decision tree.
5. Cross-validation: XGBoost supports built-in cross-validation functionality, enabling you to perform model selection and hyperparameter tuning more effectively.
6. Speed and scalability: XGBoost is designed to be efficient and scalable. It includes parallelization techniques that make use of multi-core CPUs and distributed computing frameworks for faster model training and prediction.
7. Feature importance: XGBoost provides a feature importance metric, which helps identify the most influential features in the dataset. This information can be useful for feature selection or understanding the model's decision-making process.

Overall, the XGBoost Regressor is a highly regarded algorithm in the field of machine learning, known for its excellent performance and ability to handle complex regression problems.

6. Assumptions:

The following assumptions were made in order to create the model for Walmart Sales Project:

1. The Random Forest regression and XG Boost regression model used here is a very plain and simple model, no extra parameters are added.
2. Out of all the models for forecasting, one model was selected based on the accuracy. With the help of the regressors added to the prophet model, the model will make predictions. Those predictions will be evaluated by using the selected evaluation algorithm.
3. After evaluating the forecasting model, if the accuracy of the evaluation algorithm is good, then the obtained forecasting result will be best.

7. Model Evaluation and Technique:

The following techniques and steps were involved in the evaluation of the model:

1. Splitting the Data:

```
x_train, x_test, y_train, y_test = train_test_split(scaled_features,df1['Weekly_Sales'],test_size=0.30,random_state=42)
```

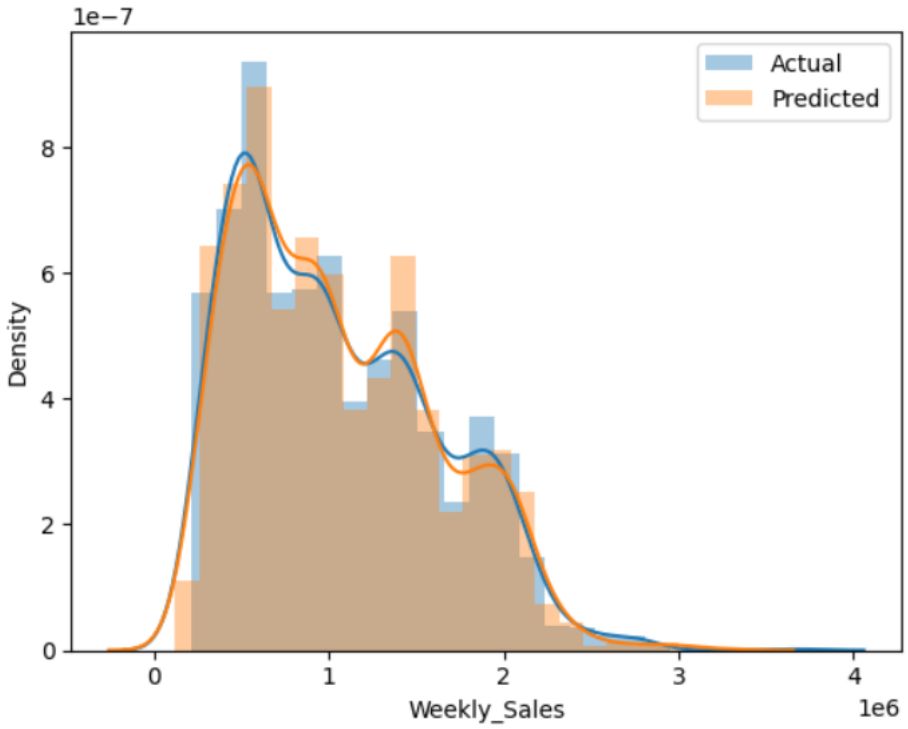
2. Fitting the Model:

```
from xgboost import XGBRegressor
xgb= XGBRegressor()
xgb.fit(x_train,y_train)
y_pred4= xgb.predict(x_test)

difference4=pd.DataFrame(np.c_[y_test,y_pred4],columns=['Actual_Value','Predicted_Value'])
print(difference4)
print('--------------------------------------------------')
```

3. Evaluating the Performance Metrics:

```
# Performance parameters
print('XGBoost Regressor Performance Parameters')
print('--------------------------------------------------')
from sklearn.metrics import r2_score
xgb_pred1=r2_score(y_test,y_pred4)
print('r2_score is:',xgb_pred1)
print('--------------------------------------------------')
from sklearn import metrics
xgb_pred2=mean_abs_error=metrics.mean_absolute_error(y_test,y_pred4)
print('MAE:',xgb_pred2)
print('--------------------------------------------------')
from sklearn.metrics import mean_absolute_percentage_error
xgb_pred3=mean_absolute_percentage_error(y_test,y_pred4)
print('MAPE:',xgb_pred3)
print('--------------------------------------------------')
xgb_pred4=mean_sq_error=metrics.mean_squared_error(y_test,y_pred4)
print('MSE:',xgb_pred4)
print('--------------------------------------------------')
xgb_pred5=root_mean_sq_error=np.sqrt(metrics.mean_squared_error(y_test,y_pred4))
print('RMSE',xgb_pred5)
```

4. Graphical Representation of the prediction: (Visualization)



The evaluation report suggests the following:

Inferences from the evaluation:

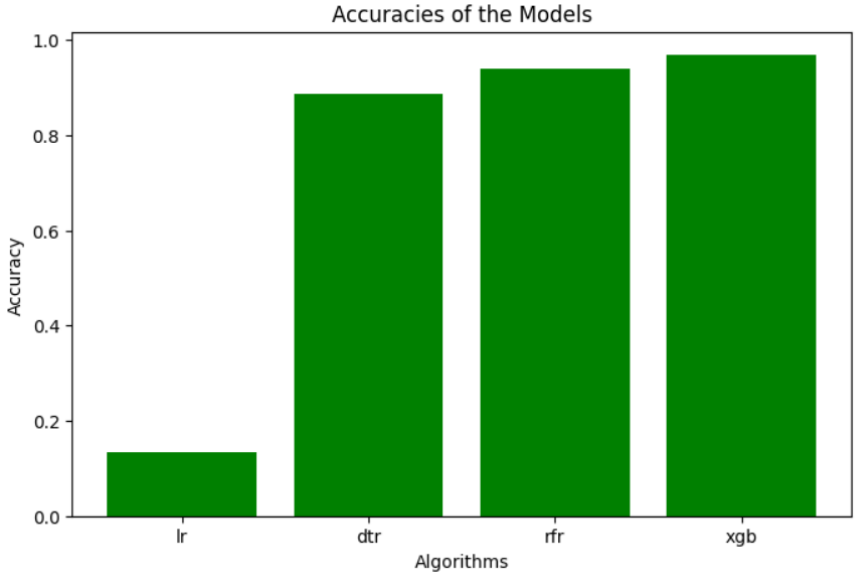
Evaluation Metrics for XGBoost Regressor are as follows:

i)	Accuracy (r2_score)	: 96.88%
ii)	Root Mean Squared Error (RMSE)	: 106632.6644
iii)	Mean Absolute Percentage Error (MAPE)	: 7.02%

Accuracy(r2_score) is highest among all algorithms i.e., **96.88%.** So, this model is the best among all with respect to the performance metrics.

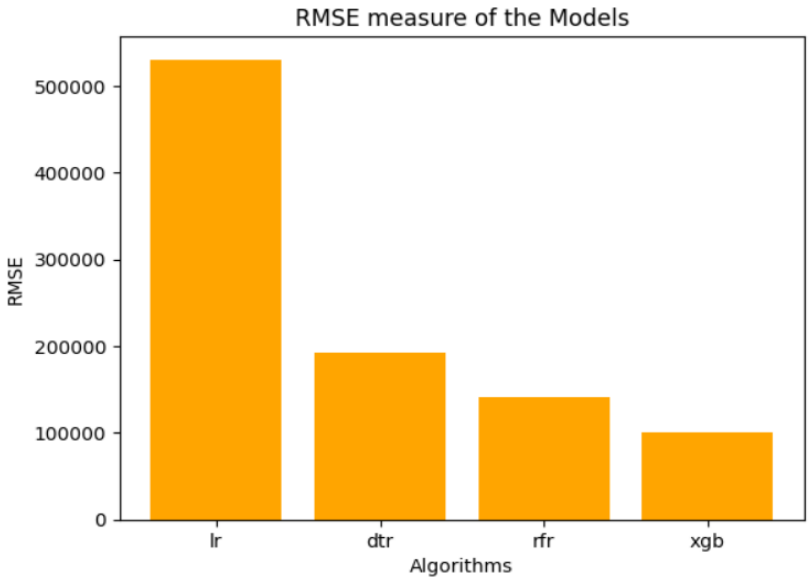
8. Inferences from the Project:

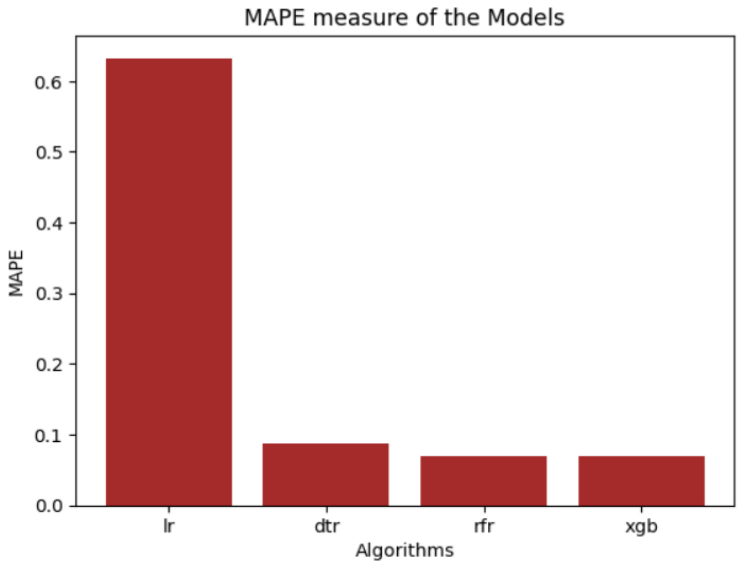
The model performance inferences are as follows, For walmart sales forecasting, following models has been built to check the accuracy, Linear Regression, Decision Tree Regressor, Random Forest Regressor and XGBoost Regressor.



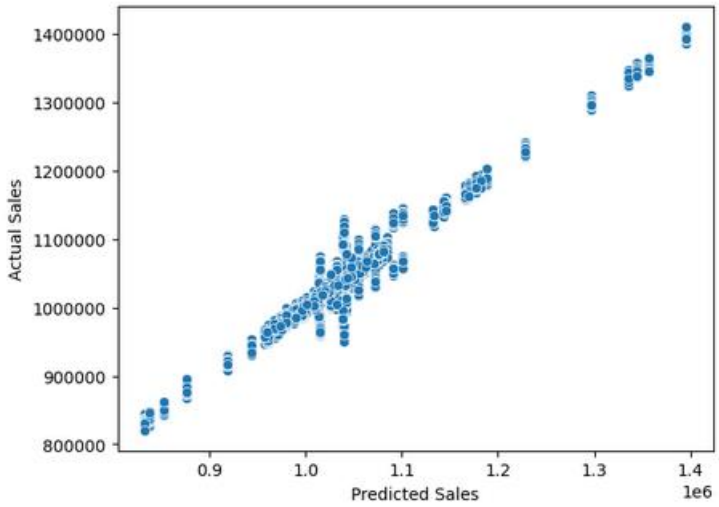
Among all these models' evaluation metrics has been compared to check the performance of the model.

Visualization of the comparison of different evaluation metrics is:

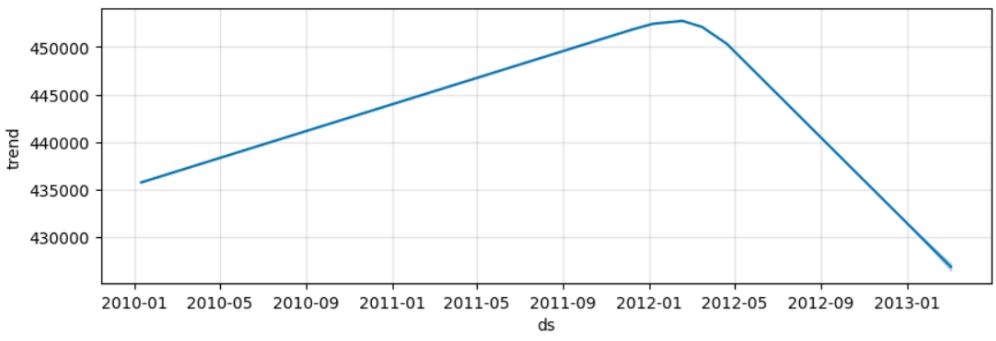




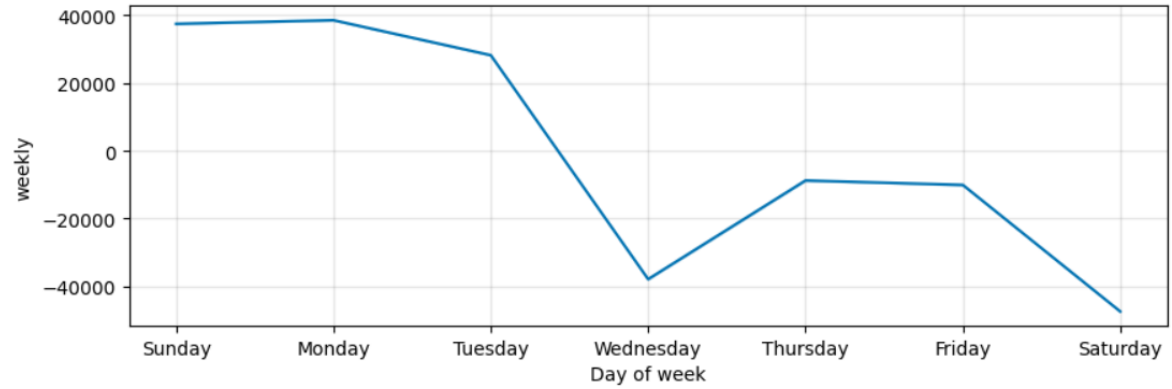
After forecasting the sales of store using time series-prophet model on XGBoost Regressor, following insights has been obtained. The Actual & Predicted Sales relation obtained after evaluating the model is linear.



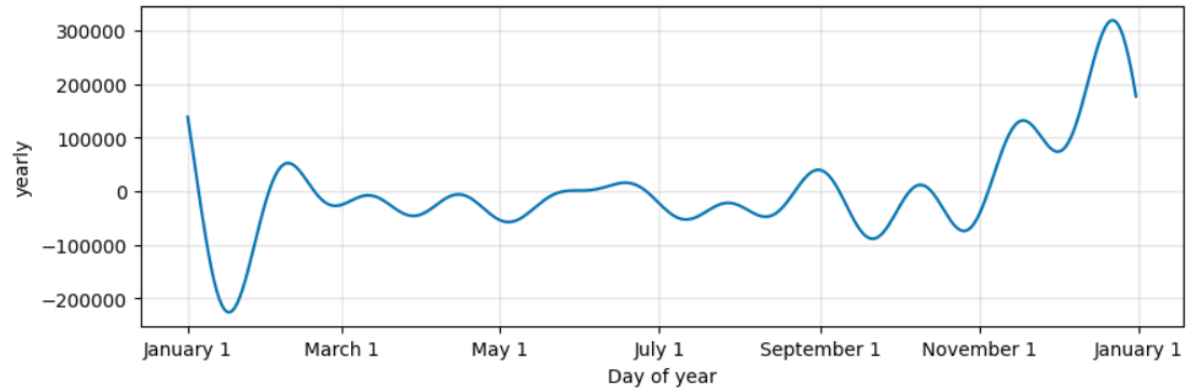
After forecasting sales of each store for the next 12 weeks, using time-series-prophet model, following trend has been seen with respect to time-data:



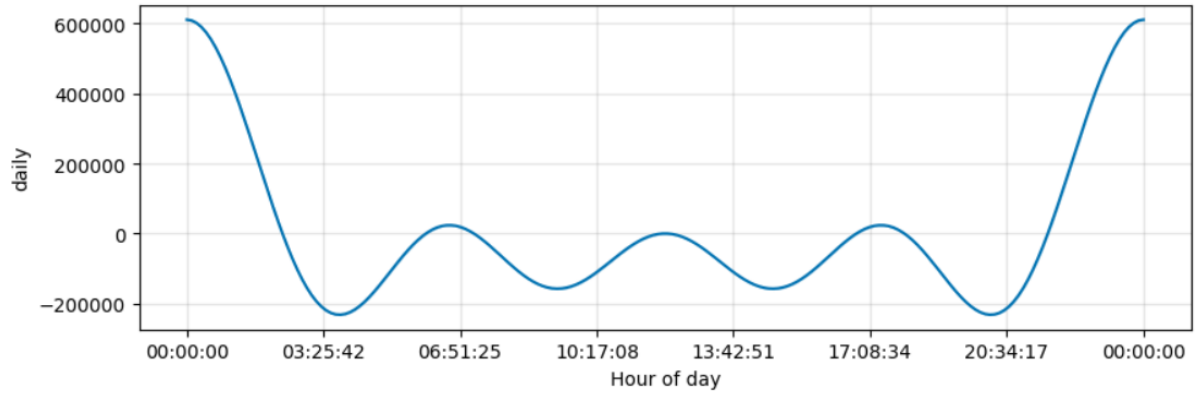
Weekly Trend:



Yearly Trend:



Daily Trend:



As per the trend, the sales has been increased till 2012 and then 2013 it starts reducing.

9. Future Possibilities:

Developing a robust and accurate sales prediction model can bring numerous benefits and opportunities to a retail business. In order to achieve it need to focus on following points:

1. Strategic Planning
2. Cost Optimization
3. Effective Pricing Strategies
4. Improved Customer Satisfaction
5. Effective Pricing Strategies

These points can contribute to the overall success and profitability of a retail business.

10. Conclusion:

Here, we can conclude that,

1. XG Boost Regression is not only fast but also efficient for time series model evaluation. It provides best results, and the company can trust the forecasting.
2. High sales in the prediction were observed in the end of the year 2012, and for the upcoming weeks, sales will fall down but recover quickly.
3. This forecast tells that there is a presence of seasonality. Every end of the year is good for sales, and every start of the year brings some fall in the sales and then recovery.

11. References:

The reference file where the project has been executed (google colab) is as follows:

https://colab.research.google.com/drive/1RvAZZIbGsW3Hi2VDF9Qse-edmgvd4GiX