Capstone Project – Walmart

1. *Problem Statement :*

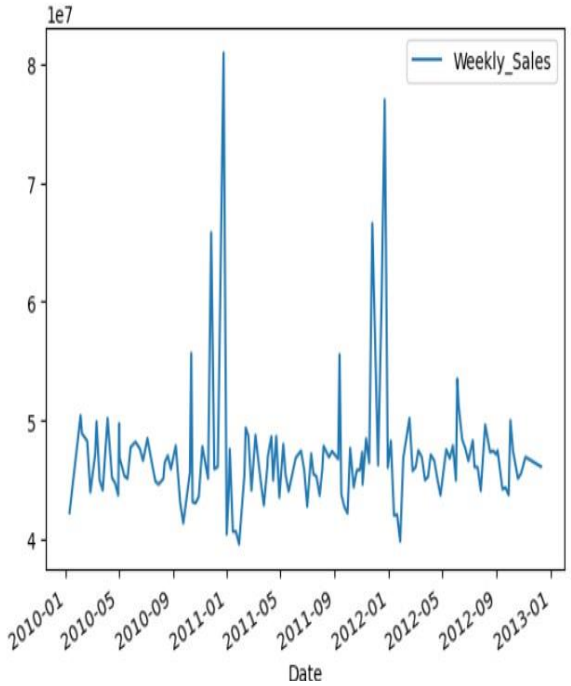
A retail store that has multiple outlets across the country are facing issues in managing the inventory - to match the demand with respect to supply. You are a data scientist, who has to come up with useful insights using the data and make prediction models to forecast the sales for X number of months/years.

Problem Statement

In [74]:
```
#reading the Walmart.csv file
df = pd.read_csv('Walmart.csv')
df['Date'] = pd.to_datetime(df['Date'])
df.groupby('Date')[['Weekly_Sales']].sum().sort_values(by='Date', ascending=True).plot()
```

```
C:\Users\anilk\AppData\Local\Temp\ipykernel_24868\1308824369.py:3: UserWarning: Parsing dates in DD/MM/YYYY format when dayfirst=False (the default) was specified. This may lead to inconsistently parsed dates! Specify a format to ensure consistent parsing.
  df['Date'] = pd.to_datetime(df['Date'])
```

Out[74]: <Axes: xlabel='Date'>



2. *Project Objective*

The objective of this project is to help my manager get forecasting insights of weekly sales onwards 2013-01 to at least 1 year, because the company want to sale unsold product in the start of the year. They have underutilized or undersold items that they want to push to shelves and sell it as soon as possible and they need forecast so that production team can cut down the supply based on future demands.

3. Data Description

1. df.isnull().sum()
2. df.duplicated().sum()
3. df.info()
4. df.describe().T
5. df.shape

You are provided with the weekly sales data for their various outlets. Use statistical analysis, EDA, outlier analysis, and handle the missing values to come up with various insights that can give them a clear perspective on the following:

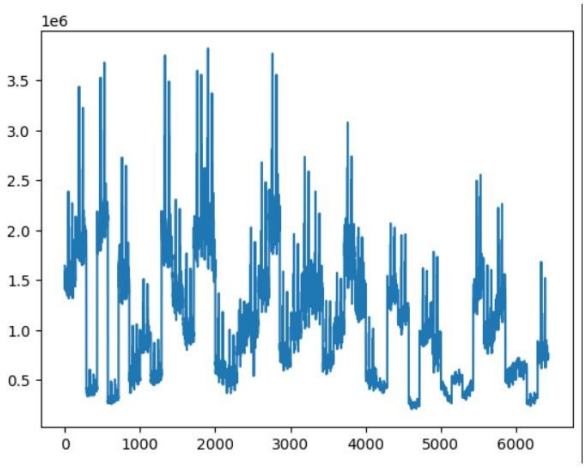
In [135]:
```
#a. If the weekly sales are affected by the unemployment rate, if yes - which stores are suffering the most?
#Weekly_Sales has -0.106176 correlation with Unemployment.
grouped = df.groupby('Store')['Unemployment'].mean().reset_index()
minvalue = grouped.min()[1]
grouped[grouped['Unemployment'] == minvalue]
#As you can store 23 and store 40 is the most suffering stores weekly sales affected by unemployment rate.
```

Out[135]:

	Store	Unemployment
22	23	4.796014
39	40	4.796014

In [136]:
```
#b. If the weekly sales show a seasonal trend, when and what could be the reason?
df['Weekly_Sales'].plot()
#Since the weekly_sales showing downward seasonal trend, it can mean there is a population shift or it might be a competititor
#effect.
```

Out[136]: <Axes: >



In [139]:
```
#c. Does temperature affect the weekly sales in any manner?
df.corr(numeric_only=True)
#As you can see temperature has weak negative correlation, there is no correlation we can conclude
#this, it means temperature has no effect on the weekly sales in any manner
```

Out[139]:

	Store	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	CPI	Unemployment
Store	1.000000e+00	-0.335332	-4.386841e-16	-0.022659	0.060023	-0.209492	0.223531
Weekly_Sales	-3.353320e-01	1.000000	3.689097e-02	-0.063810	0.009464	-0.072634	-0.106176
Holiday_Flag	-4.386841e-16	0.036891	1.000000e+00	-0.155091	-0.078347	-0.002162	0.010960
Temperature	-2.265908e-02	-0.063810	-1.550913e-01	1.000000	0.144982	0.176888	0.101158
Fuel_Price	6.002295e-02	0.009464	-7.834652e-02	0.144982	1.000000	-0.170642	-0.034684
CPI	-2.094919e-01	-0.072634	-2.162091e-03	0.176888	-0.170642	1.000000	-0.302020
Unemployment	2.235313e-01	-0.106176	1.096028e-02	0.101158	-0.034684	-0.302020	1.000000

In [138]:
```
#d. How is the Consumer Price index affecting the weekly sales of various stores?
df.corr(numeric_only=True)
#As you can see Consumer Price Index has no affect on the weekly sales of various stores. The correlation is -0.072634
#it means CPI is not affecting the weekly sales of various stores.
```

Out[138]:

	Store	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	CPI	Unemployment
Store	1.000000e+00	-0.335332	-4.386841e-16	-0.022659	0.060023	-0.209492	0.223531
Weekly_Sales	-3.353320e-01	1.000000	3.689097e-02	-0.063810	0.009464	-0.072634	-0.106176
Holiday_Flag	-4.386841e-16	0.036891	1.000000e+00	-0.155091	-0.078347	-0.002162	0.010960
Temperature	-2.265908e-02	-0.063810	-1.550913e-01	1.000000	0.144982	0.176888	0.101158
Fuel_Price	6.002295e-02	0.009464	-7.834652e-02	0.144982	1.000000	-0.170642	-0.034684
CPI	-2.094919e-01	-0.072634	-2.162091e-03	0.176888	-0.170642	1.000000	-0.302020
Unemployment	2.235313e-01	-0.106176	1.096028e-02	0.101158	-0.034684	-0.302020	1.000000

In [115]:
```
#e. Top performing stores according to the historical data.
grouped = df.groupby('Store')[['Store','Weekly_Sales']].sum()
a = maxvalue.max()
a = a.values[1]
ab = grouped.eq(301397792.46)
ab[ab['Weekly_Sales']==True]
print(grouped.loc[20:20])
#Store No. 20 is the top performing store according to the historical data.
```

Out[115]:

Store	Store	Weekly_Sales
20	2860	3.013978e+08

In [124]:
```
#f. The worst performing store, and how significant is the difference between the highest and
#lowest performing stores.
grouped = df.groupby('Store')[['Store','Weekly_Sales']].sum()
a = grouped.max()
a = a.values[1]
ab = grouped.eq(301397792.46)
ab[ab['Weekly_Sales']==True]
print(grouped.loc[20:20])
print(grouped.min()[0])
#Store 20 is the highest performing store
#Store 143 is the lowest performing store
```

```
       Store  Weekly_Sales
Store
20      2860  3.013978e+08
143.0
```

4. Data Preprocessing Steps and Inspiration

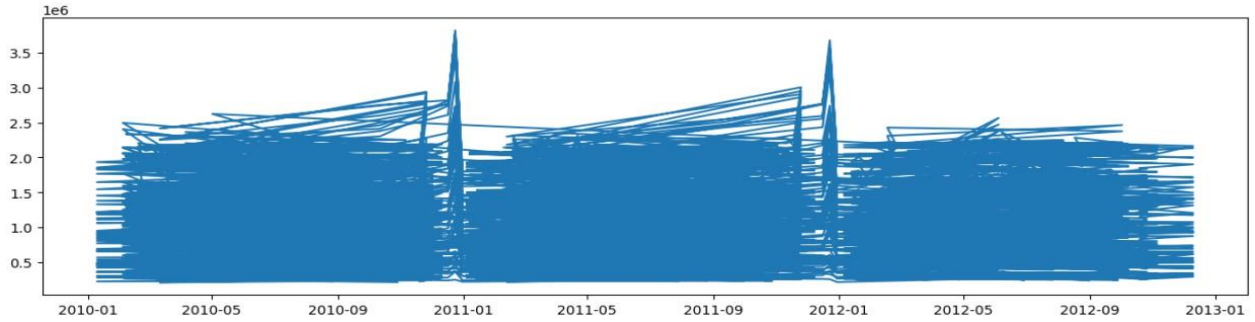
1 . df.drop(columns=['Store', 'Holiday_Flag', 'Temperature', 'Fuel_Price', 'CPI', 'Unemployment'], inplace=True, axis = 1)

2. df['Date'] = pd.to_datetime(df['Date'])

3. df.columns = ['ds', 'y']

Data Preprocessing Steps and Inspiration

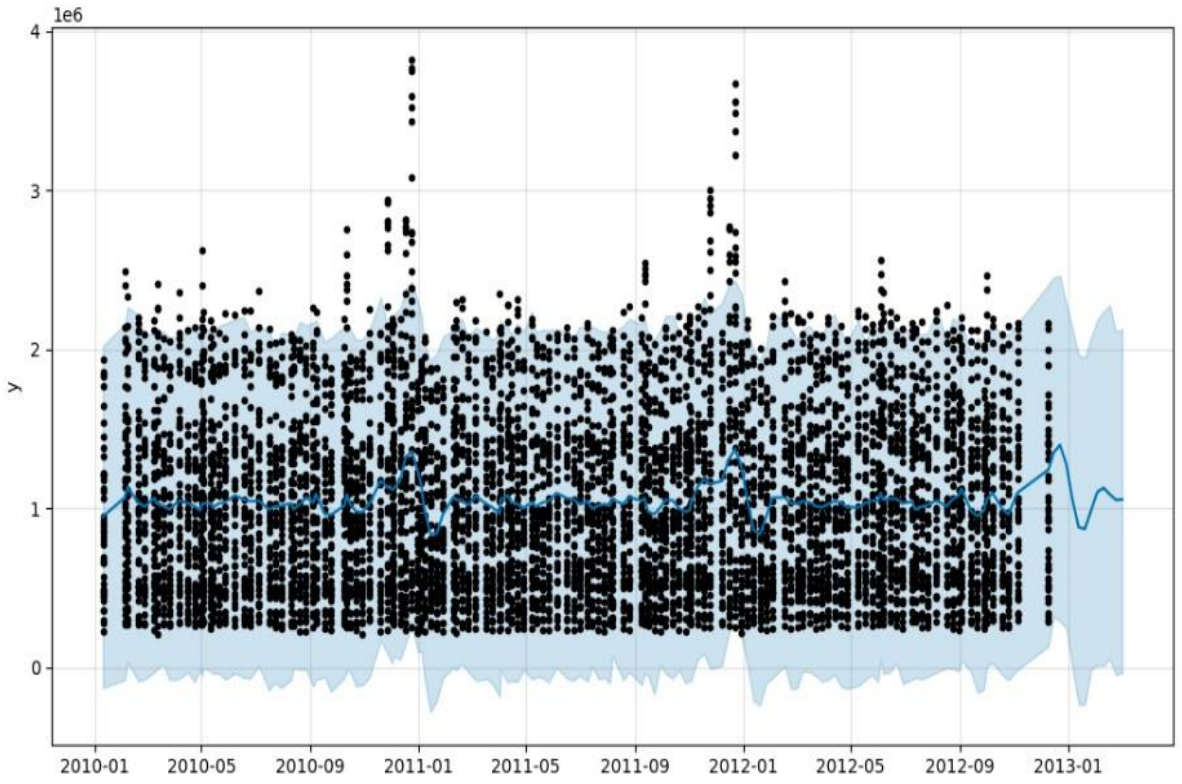
```
In [48]: df.drop(columns=['Store', 'Holiday_Flag', 'Temperature', 'Fuel_Price', 'CPI', 'Unemployment'], inplace=True, axis = 1)

In [52]: df['Date'] = pd.to_datetime(df['Date'])

In [56]: df.index = df['Date']
         df.drop(columns = ['Date'], inplace=True)

In [68]: plt.figure(figsize=(15,4))
         plt.plot(df['Weekly_Sales'])

Out[68]: [<matplotlib.lines.Line2D at 0x2075ee09050>]
```



Insights: *Last month and first month of every year, i.e. December-January there is a surge on weekly sales. This seasonal pattern indicates a festival celebration, i.e. Christmas season. Once the Christmas is over, the weekly sales seems to take a dip in sales.*

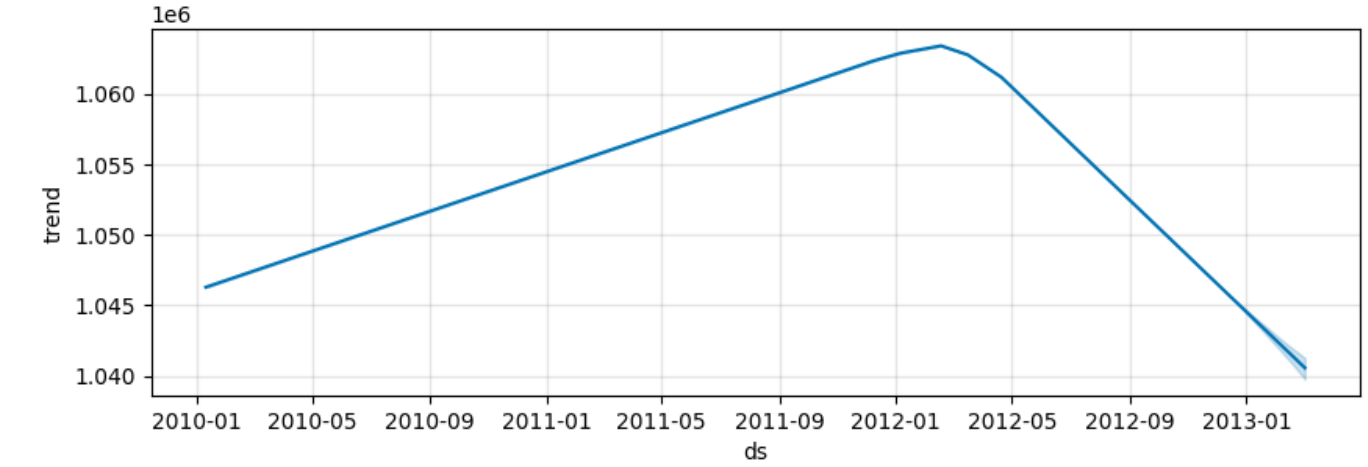
5. Choosing The Algorithm For The Project

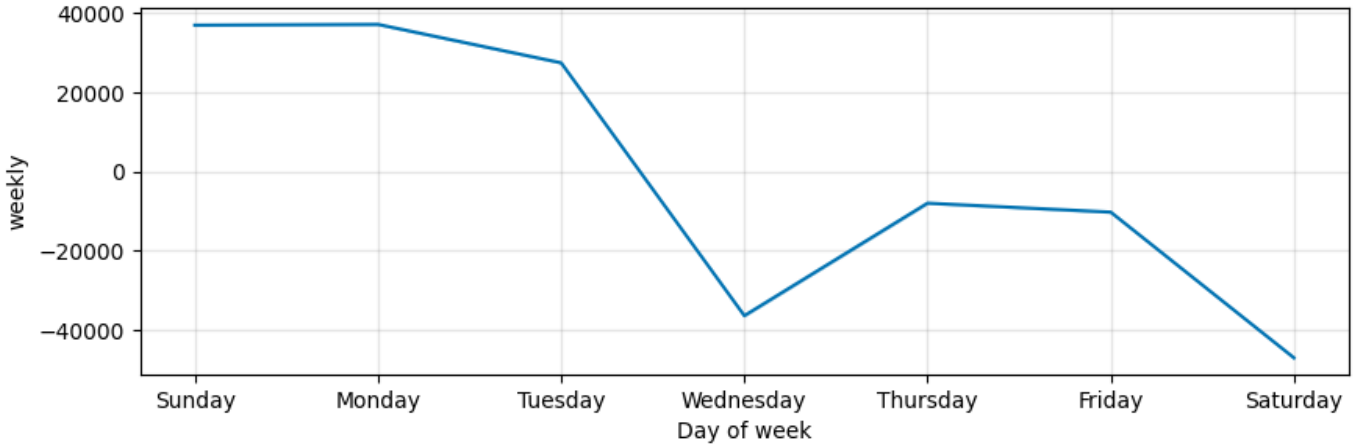
```
In [94]: from prophet import Prophet

In [95]: model = Prophet(interval_width=0.95)

In [ ]: model.fit(df)

In [143]: future_dates = model.make_future_dataframe(periods=12, freq='W')

In [144]: forecast = model.predict(future_dates)

In [145]: sales_forecast = model.plot(forecast)
```

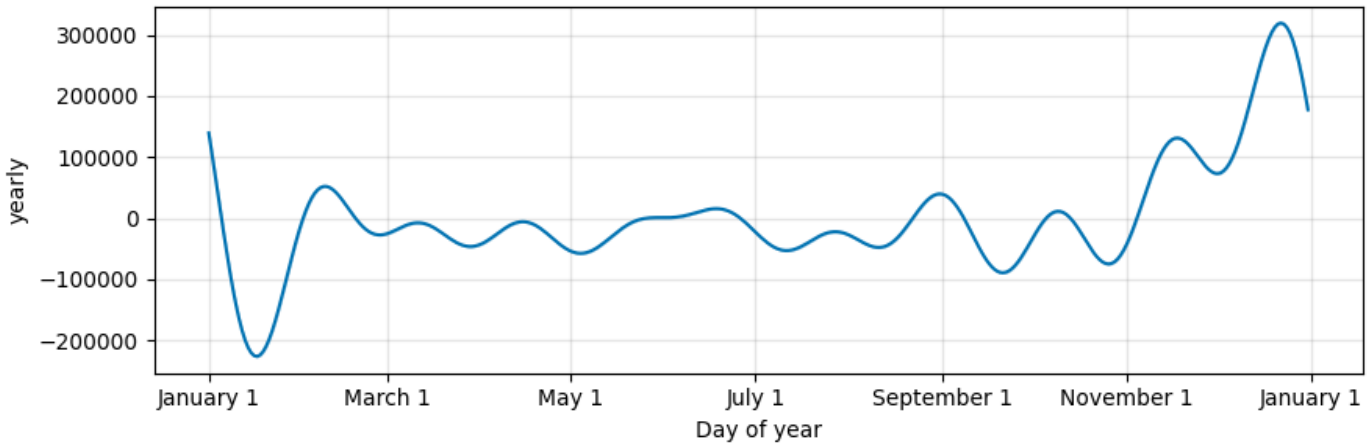


Since my manager wants me to forecast weekly sales for 12 weeks and we have date and weekly sales which makes the strong candidates to tell it's a 'time series' problem, hence used time series algorithm.

As per weekly trend of the forecast, it clearly says, every Wednesday there is lowest sale mark in any given week and on every Sunday the sales are the highest.







6. *Assumptions*

1. The problem of this project is to forecast sales for 12 weeks, I assumed it to be a forecasting concern hence used Time Series.
2. Since my manager wants me to give insights and improve sales in various areas, I did descriptive analysis on Data Description and described the data.
3. The stores have issue managing the inventory, I proposed my time series forecasting to help make stores make informed decisions and reduce/increase supply based on future demand.

7. Model Evaluation And Technique

Not applicable for this model. I used prophet library.

8. Inferences From The Project

1. The model is forecasting good, good because it's following the seasonality of the data properly. There is no trend in the data.
2. This forecast can be used to predict for many years, however there can be some noise it can lead to low sales or high sale surge for future.
3. This project yields sales information on weekly basis for Walmart store in US States.

9. Future Possibilities

1. This model is limited to only Walmart Sales forecast for US States.
2. I will try to evaluate this model, since I have ‘yhat’ and I will try to forecast in a way that I already have k months data, I will evaluate for 1, 2, 3, k-3 months and forecast for k-2 and k-1. This way I will be able to evaluate the model. I will do my research

10. Conclusion

This project is build to do time series forecasting keeping in mind the simplicity and difficulty level of the model. I used prophet library which is convenient and easy, because using prophet we don't have to make our data stationary and all good reasons why I picked prophet. Otherwise using ARIMA and SARIMA model is also good for this. I wanted to use prophet to deliver simplicity in forecasting time series.

It was my pleasure developing this capstone project. I have many more to do with more attractive graphs and more simpler code and better design and customization.

My final words: This project is concise and apt to do time series for Walmart Sales forecasting. I will conclude here by saying 'I am grateful to do this project and happy to share with Intellipaat as my capstone project'.