Superstore Sales Data Projections

August 20, 2023

1 Superstore Sales Data Forecasting

1.1 Objective

Based on the analysis of the quarterly sales of the dataset on average there is an increase of \$12,780.33 in quartely sales, a 18% increase. The objective of this forecast is to forecast the quarterly sales total of Q1 of 2019.

[45]:
```
# Importing packages

import pandas as pd
import numpy as np
from prophet import Prophet
```

[33]:
```
#Import csv
df1 = pd.read_csv(r"/Users/scipio/Downloads/Sales_Dataset_Project.csv")

#Converting 'Order Date' to datetime format
df1['Order Date'] = pd.to_datetime(df1['Order Date'])

df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9800 entries, 0 to 9799
Data columns (total 18 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Row ID         9800 non-null   int64
 1   Order ID       9800 non-null   object
 2   Order Date     9800 non-null   datetime64[ns]
 3   Ship Date      9800 non-null   object
 4   Ship Mode      9800 non-null   object
 5   Customer ID    9800 non-null   object
 6   Customer Name  9800 non-null   object
 7   Segment        9800 non-null   object
 8   Country        9800 non-null   object
 9   City           9800 non-null   object
 10  State          9800 non-null   object
 11  Postal Code    9789 non-null   float64
 12  Region         9800 non-null   object
 13  Product ID     9800 non-null   object
 14  Category       9800 non-null   object
 15  Sub-Category   9800 non-null   object
 16  Product Name   9800 non-null   object
 17  Sales          9800 non-null   float64
dtypes: datetime64[ns](1), float64(2), int64(1), object(14)
memory usage: 1.3+ MB
```

```
/var/folders/3k/bzmghyyj1j51lkx1mc36njjw0000gn/T/ipykernel_40862/2354983070.py:5
: 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.
  df1['Order Date'] = pd.to_datetime(df1['Order Date'])
```

[36]:
```
#Dropping Columns

col_drop = ['Row ID', 'Order ID', 'Ship Date', 'Ship Mode',
       'Customer ID', 'Customer Name', 'Segment', 'Country', 'City', 'State',
       'Postal Code', 'Region', 'Product ID', 'Category', 'Sub-Category',
       'Product Name']

df1 = df1.drop(columns = col_drop)
```

[51]:
```
#Renaming Columns
new_columns = {'Order Date':'ds', 'Sales':'y'}

df1.rename(columns = new_columns, inplace = True)

df1.head()
```

[51]:
```
          ds         y
0 2017-08-11  261.9600
1 2017-08-11  731.9400
2 2017-12-06   14.6200
3 2016-11-10  957.5775
4 2016-11-10   22.3680
```

[52]:
```
#Instantiating Prophet Object

m = Prophet()

#fitting model

m.fit(df1)
```

```
13:39:33 - cmdstanpy - INFO - Chain [1] start processing
13:39:34 - cmdstanpy - INFO - Chain [1] done processing
```

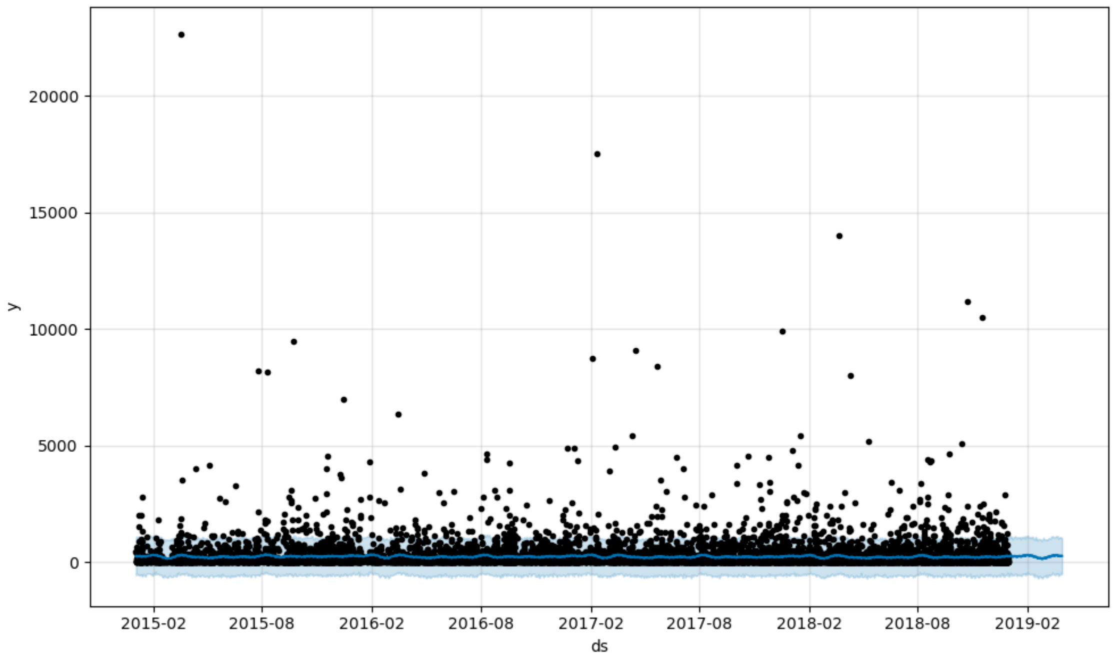
```
[52]: <prophet.forecaster.Prophet at 0x7f9c5811ded0>
```

```
[164]: #Predicting the next quarter
       future = m.make_future_dataframe(periods = 90)
```

```
[187]: #Forecasting
       forecast = m.predict(future)
```

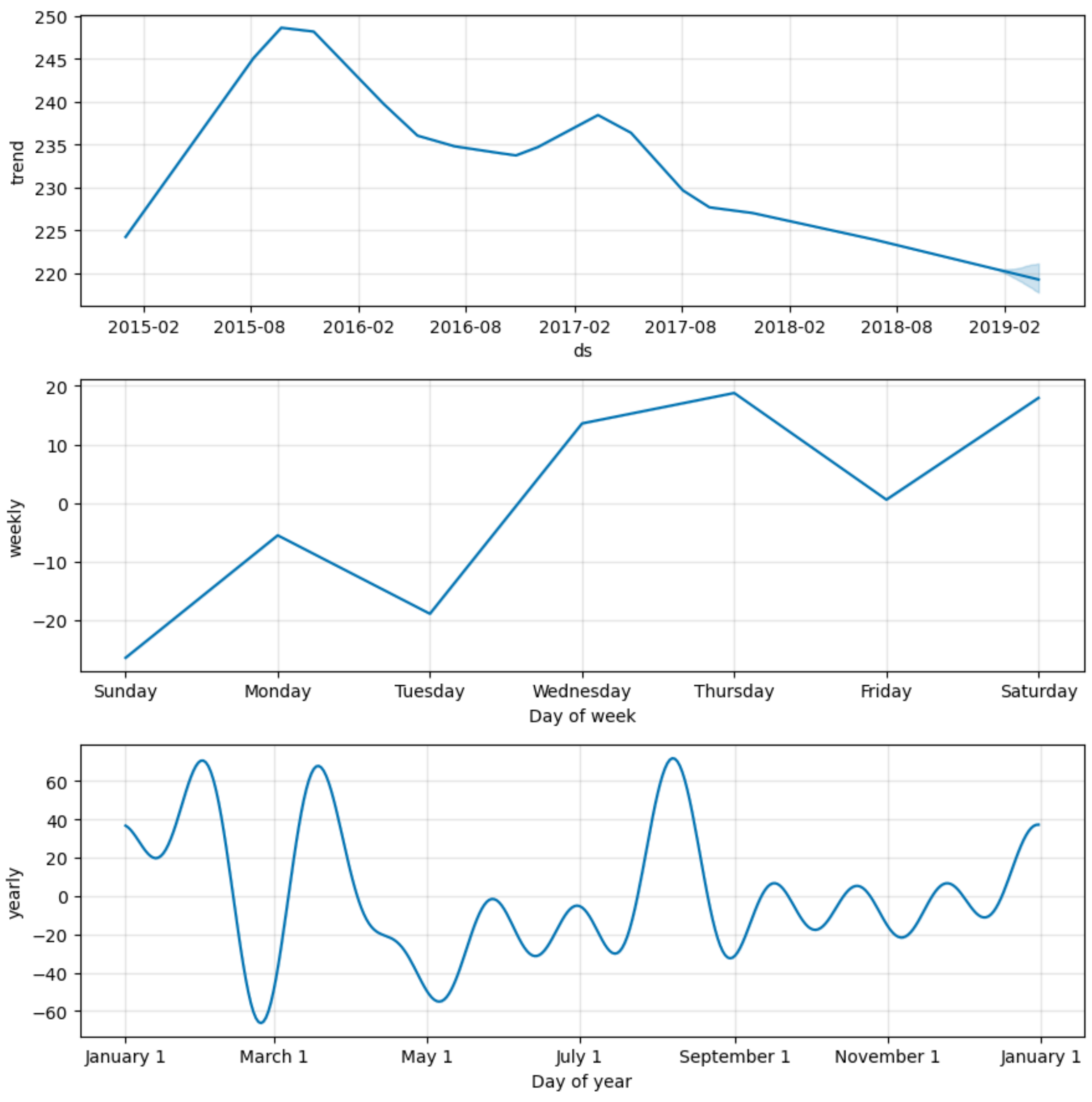
1.2 Scatter Plot of Daily Sales

```
[186]: fig1 = m.plot(forecast)
```



1.3 Daily, Weekly, and Yearly Trends

```
[184]: fig2 = m.plot_components(forecast)
```



There is a positive sales trend.The trend of sales is highest between the weekdays of Tuesday and Saturday. There is a spike in yeraly trend data between July and September. Lastly, there is a signigficant drop in yearly sales trend between January and March.

1.4 Forecasting Q1 2019

[170]:
```python
#Creating Month Column
forecast['Month'] = forecast['ds'].dt.month

#Creating Year Column
forecast['Year'] = forecast['ds'].dt.year
```

[189]:
```
#Margin of Error
margin_of_error = (Q1_Forecast_2019['yhat_upper'].sum() -␣
 ↪Q1_Forecast_2019['yhat_lower'].sum())/2

#Calculating Confidence Interval
lower_bound = round(Q1_Forecast_2019['yhat'].sum() - margin_of_error,2)
upper_bound = round(Q1_Forecast_2019['yhat'].sum() + margin_of_error,2)

print(f"Forecast Interval: [{lower_bound}, {upper_bound}]")
```

```
Forecast Interval: [-49565.6, 92596.46]
```

The value of the sales total of Q1 2019 is projected to be between -$49,565.60 and +$92,596.46

[171]:
```
#Creating Q1_Forecast_2019 df
Q1_Forecast_2019 = forecast[forecast['Year']== 2019]

#Adding Qtr Columns
Q1_Forecast_2019['Qtr'] = Q1_Forecast_2019['ds'].dt.quarter

#Monthly Forecasting Data
round(Q1_Forecast_2019.groupby(['Year','Month'])['yhat'].sum(),2)
```

```
/var/folders/3k/bzmghyyj1j51lkx1mc36njjw0000gn/T/ipykernel_40862/61217769.py:5:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  Q1_Forecast_2019['Qtr'] = Q1_Forecast_2019['ds'].dt.quarter
```

[171]:
```
Year  Month
2019  1        7990.15
      2        6045.16
      3        7480.12
Name: yhat, dtype: float64
```

[172]:
```
#Quarterly Forecast
round(Q1_Forecast_2019.groupby(['Year','Qtr'])['yhat'].sum(),2)
```

[172]:
```
Year  Qtr
2019  1      21515.43
Name: yhat, dtype: float64
```

The projected sales total of Q1 2019 is +$21,515.43