Name: Sakshi Sawant
Roll No. 6957
MSC IT Part - 2

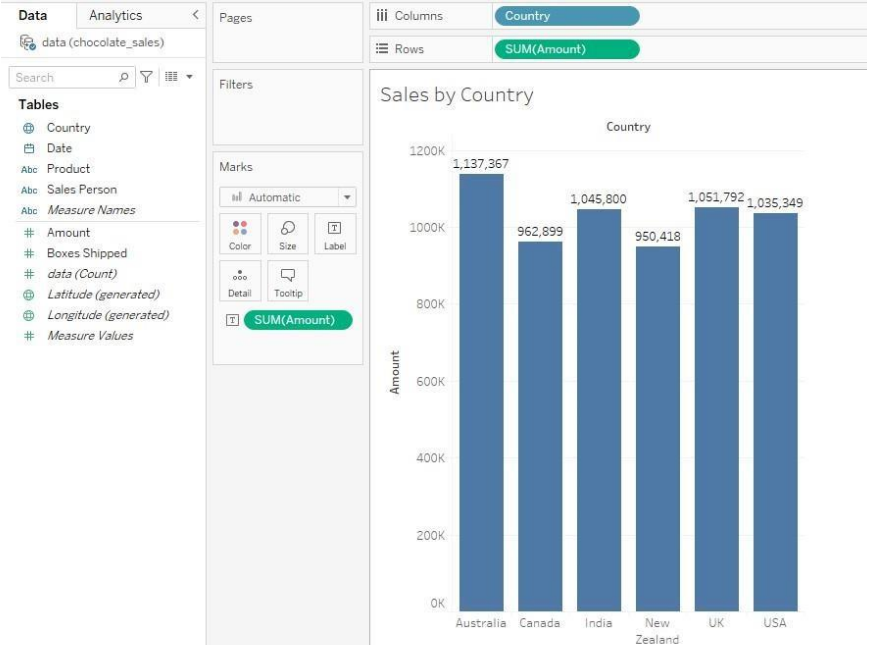
Practical 01

Aim – Descriptive analysis using Tableau.

(Chocolate factory dataset)

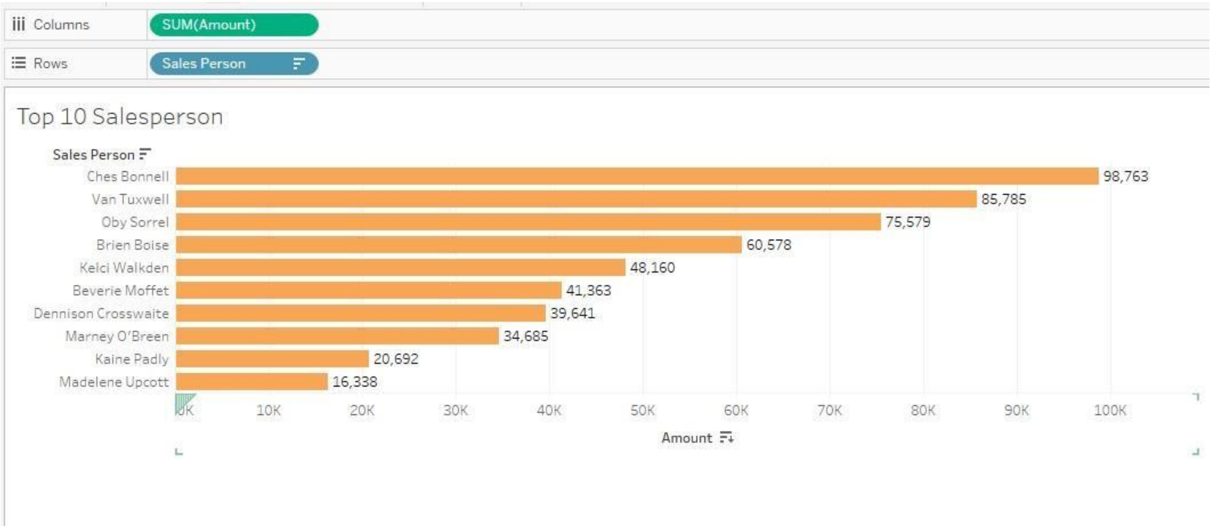
1. Analyze sales by country.

Sheet 1


Columns: Country
Rows: SUM(Amount)

2. Analyze top 10 sales.

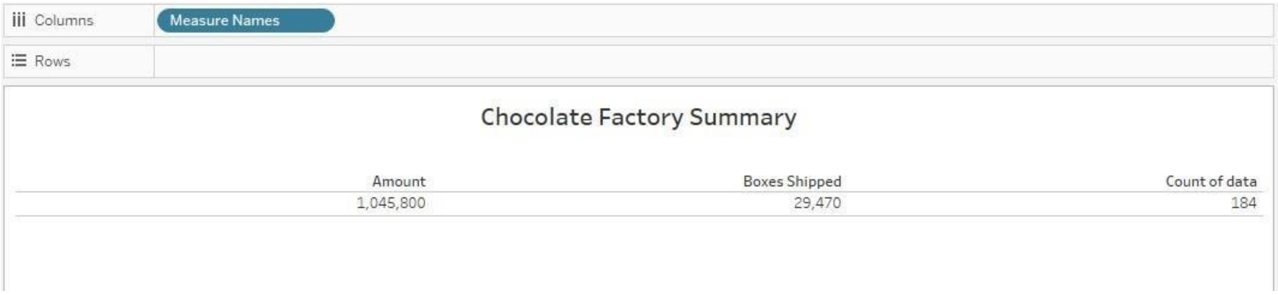
Sheet 2


Columns: SUM(Amount)
Rows: Sales Person

Name: Sakshi Sawant
Roll No. 6957
MSC IT Part - 2

3. Create an interactive dashboard with a short summary.

Sheet 3



Columns: Measure Names

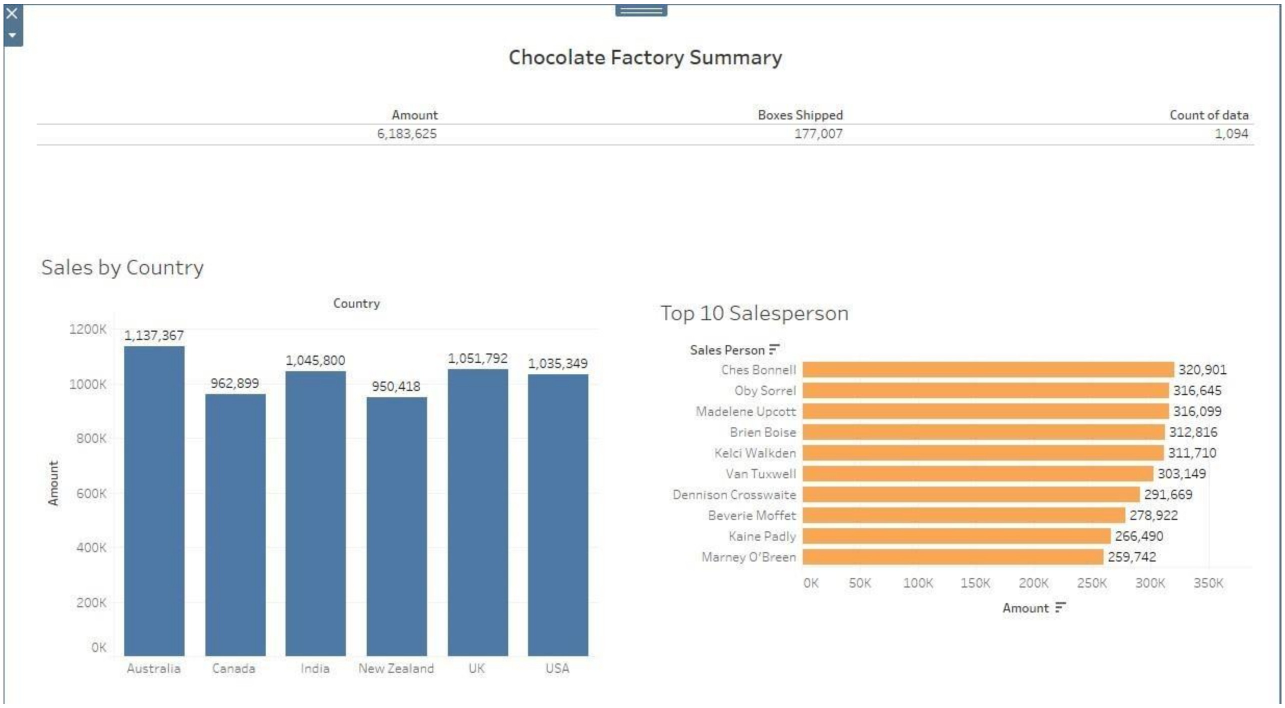
Rows

Chocolate Factory Summary

Amount	Boxes Shipped	Count of data
1,045,800	29,470	184

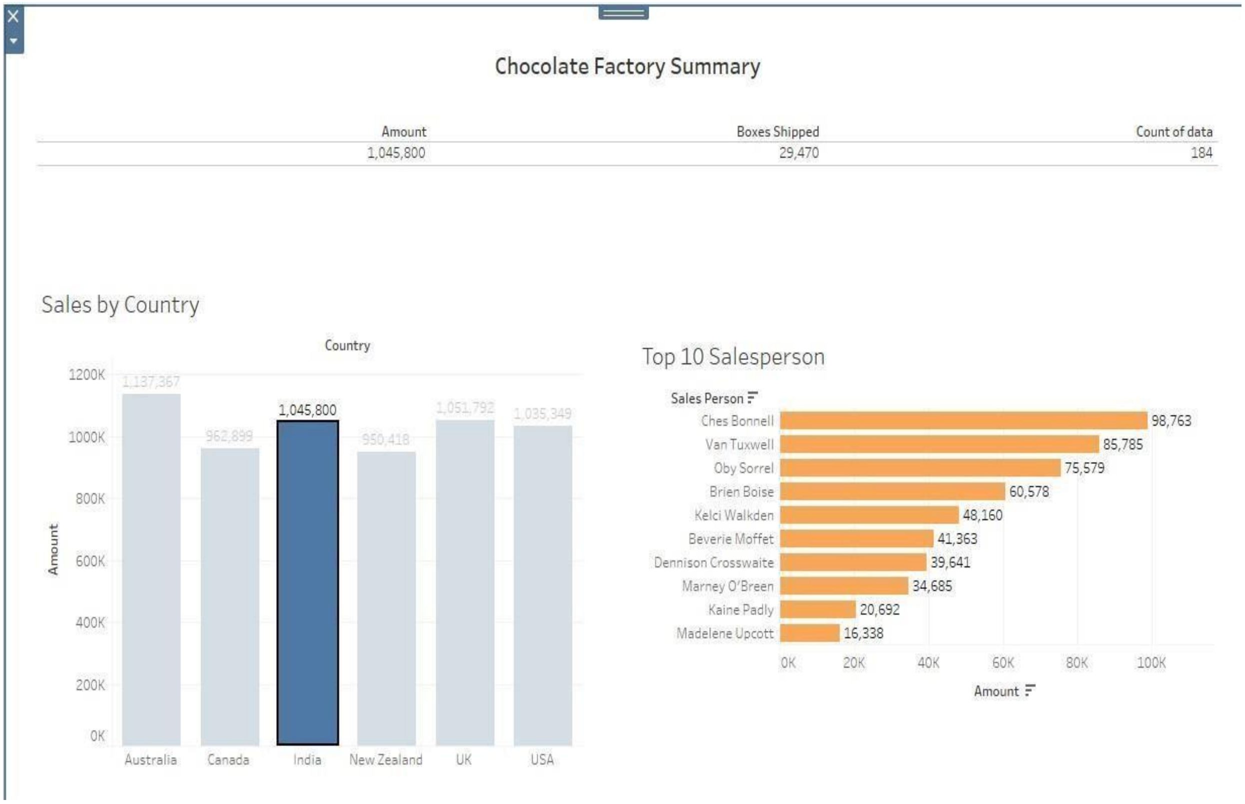
Name: Sakshi Sawant
Roll No. 6957
MSC IT Part - 2

Dashboard



Chocolate Factory Summary

Amount	Boxes Shipped	Count of data
6,183,625	177,007	1,094



Chocolate Factory Summary

Amount	Boxes Shipped	Count of data
1,045,800	29,470	184

Name: Sakshi Sawant
Roll No. 6957
MSC IT Part - 2

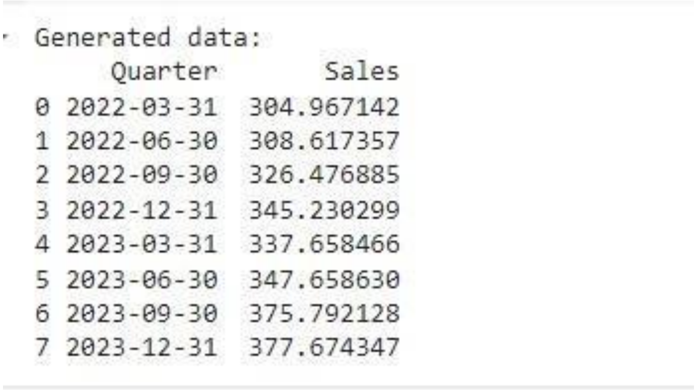
Practical 02

AIM: Predictive Analysis - Forecasting the next quarter's Sales from the previous 2 years data

Code:-

```
import pandas as pd import

numpy as np

import matplotlib.pyplot as plt
 from statsmodels.tsa.arima.model import
ARIMA



#Step 1: Generate Sample Historical Sales Data

np.random.seed(42)

quarters = pd.date_range(start='2022-01-01', periods=8, freq='Q')
 sales = [300, 310, 320, 330, 340, 350, 360, 370] +
np.random.normal(0,
10, 8) #noise added
 data = pd.DataFrame({'Quarter': quarters, 'Sales':
sales})

 print("Generated
data:") print(data)
```

Output:-

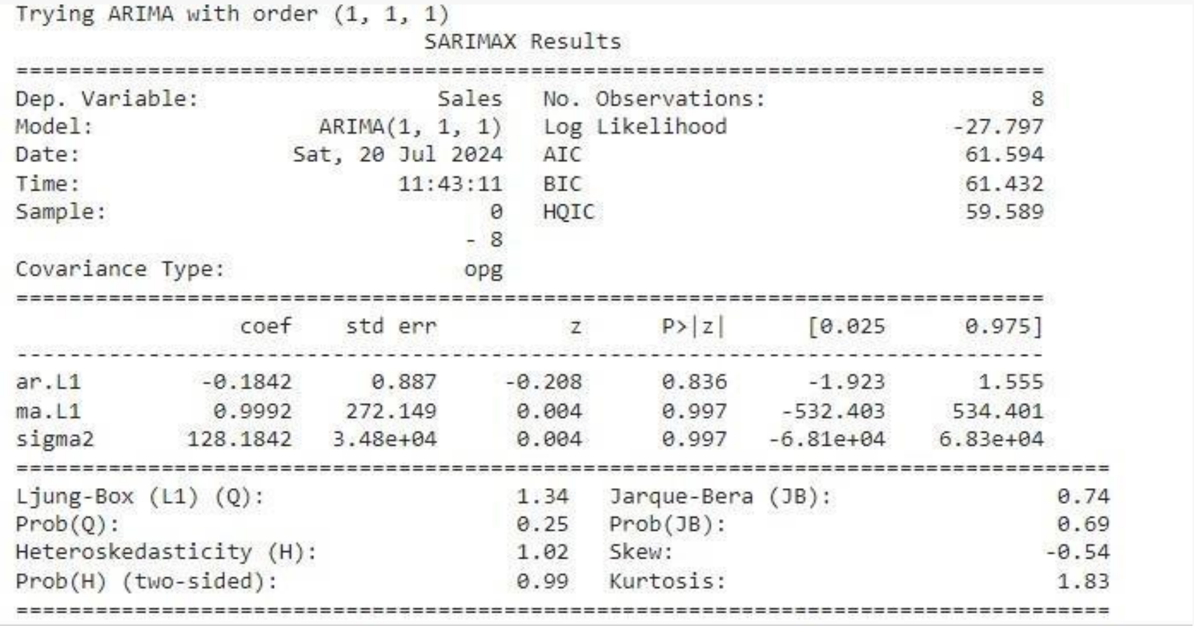


```
Generated data:
     Quarter       Sales
0 2022-03-31  304.967142
1 2022-06-30  308.617357
2 2022-09-30  326.476885
3 2022-12-31  345.230299
4 2023-03-31  337.658466
5 2023-06-30  347.658630
6 2023-09-30  375.792128
7 2023-12-31  377.674347
```

Code:-

```
#Train      ARIMA      model

successful_fit = False

arima_orders = [(1, 1, 1), (2, 1, 2), (0, 1, 1), (1, 0, 1)]
 for order in
arima_orders:

    try:

         print(f"Trying   ARIMA   with   order

         {order}")      model      =      ARIMA

         (data['Sales'], order=order) model_fit

         =  model.fit()  successful_fit  =  True

         break

    except Exception as e:
 print(f"Error in fitting ARIMA model with order {order}: {e}")

if successful_fit:



    t
    r
    y
    :
 print(model_fit.summary())  except  Exception  as

    e:  print("Error  in  printing  model  summary:

    {e}")
```

Output:-

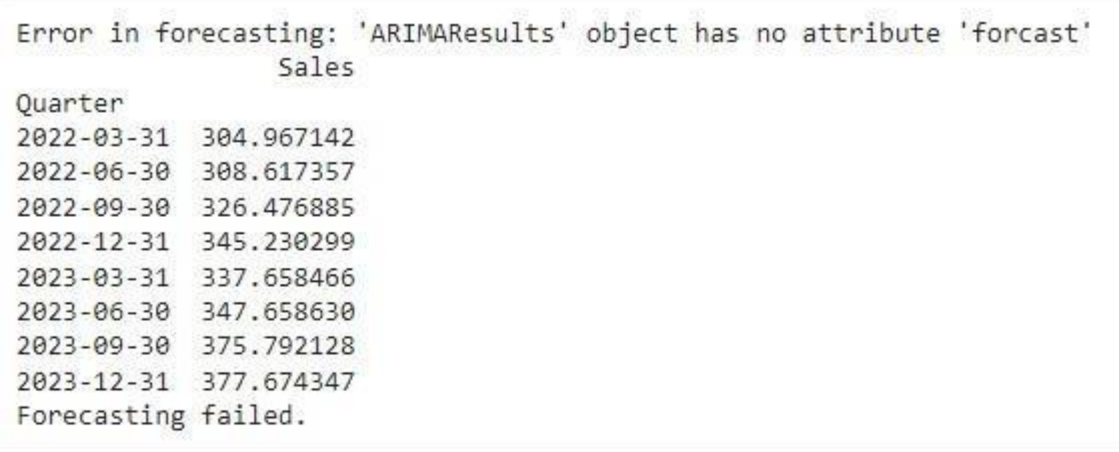


```
Trying ARIMA with order (1, 1, 1)
                               SARIMAX Results
==============================================================================
Dep. Variable:                  Sales   No. Observations:                    8
Model:                 ARIMA(1, 1, 1)   Log Likelihood                 -27.797
Date:                Sat, 20 Jul 2024   AIC                             61.594
Time:                        11:43:11   BIC                             61.432
Sample:                             0   HQIC                            59.589
                                  - 8
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -0.1842      0.887     -0.208      0.836      -1.923       1.555
ma.L1          0.9992    272.149      0.004      0.997    -532.403     534.401
sigma2       128.1842   3.48e+04      0.004      0.997   -6.81e+04    6.83e+04
===================================================================================
Ljung-Box (L1) (Q):                   1.34   Jarque-Bera (JB):                 0.74
Prob(Q):                              0.25   Prob(JB):                         0.69
Heteroskedasticity (H):               1.02   Skew:                            -0.54
Prob(H) (two-sided):                  0.99   Kurtosis:                         1.83
===================================================================================
```

Code:-

```
#Forcast next Quarter Sales try:

  forecast = model_fit.forcast(steps=1)

  print("Forcasting result:", forecast)

  next_quarter_sales = forecast.iloc[0]

except Exception as e:
 print(f"Error in forecasting: {e}")
  next_quarter_sales = None



#sample   data   &   forecast   sales

data.set_index('Quarter', inplace=True)

print(data)

if next_quarter_sales is not None:
 print(f"Frocast Sales for the next quarter:
{next_quarter_sales:.2f}")
 else:
 print("Forecasting failed.")
```

Ouput:-

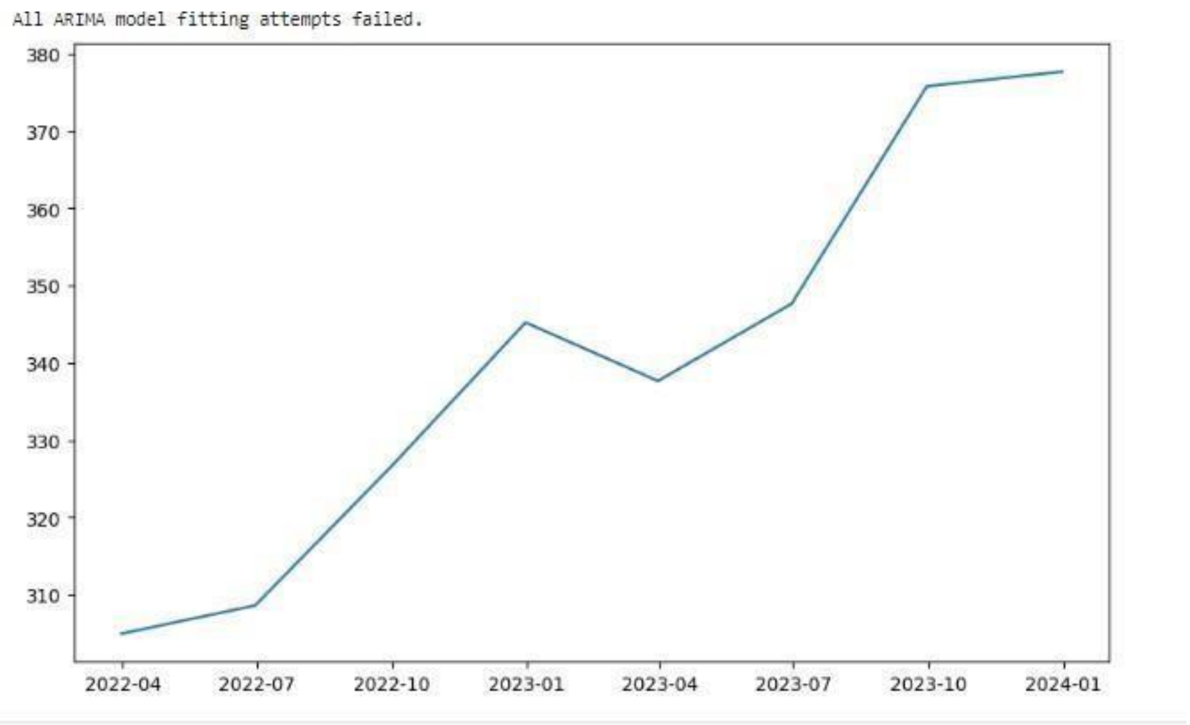


```
Error in forecasting: 'ARIMAResults' object has no attribute 'forcast'
                 Sales
Quarter
2022-03-31  304.967142
2022-06-30  308.617357
2022-09-30  326.476885
2022-12-31  345.230299
2023-03-31  337.658466
2023-06-30  347.658630
2023-09-30  375.792128
2023-12-31  377.674347
Forecasting failed.
```

Code:-

```
#plot the historical data & the forecast

plt.figure(figsize=(10, 6)) plt.plot(data.index,

data['Sales'], label='Historical Sales') if

next_quarter_sales is not None:

    plt.plot(data.index[-1] + pd.DateOffset(months=3),
next_quarter_sales, 'ro', label='Forecasted Sales')

    plt.title('Quarterly Sales Forecast using ARIMA

    Model') plt.xlabel('Quarter') plt.ylabel('Sales')

    plt.legend() plt.grid(True) plt.show()

else:
 print("All ARIMA model fitting attempts failed.")
```

Output:-



```
All ARIMA model fitting attempts failed.
```

Name: Sakshi Sawant
Roll No. 6957
MSC IT Part - 2

Name: Sakshi Sawant
Roll No. 6957
MSC IT Part - 2

Practical 03

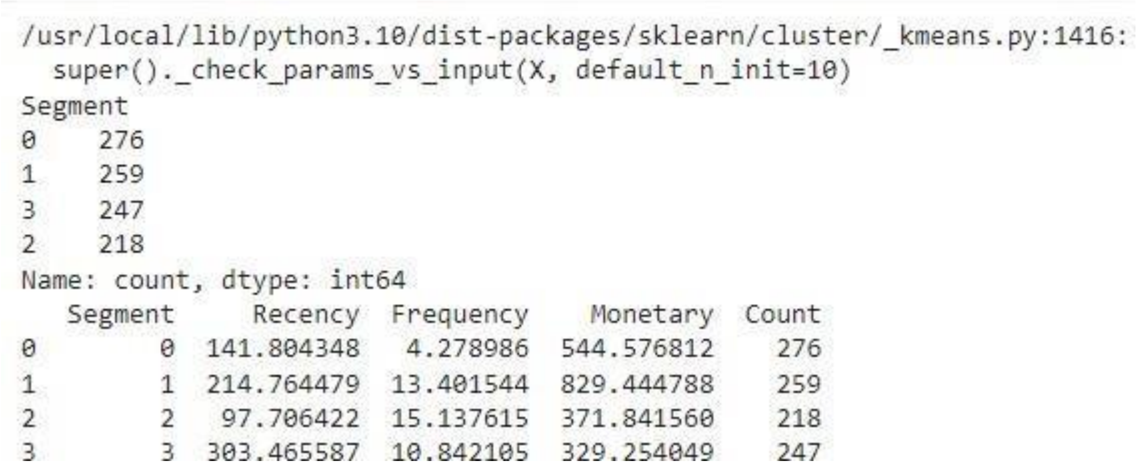
Aim:-Marketing Analysis: Segmenting Customers Based on Behavior

Code:-

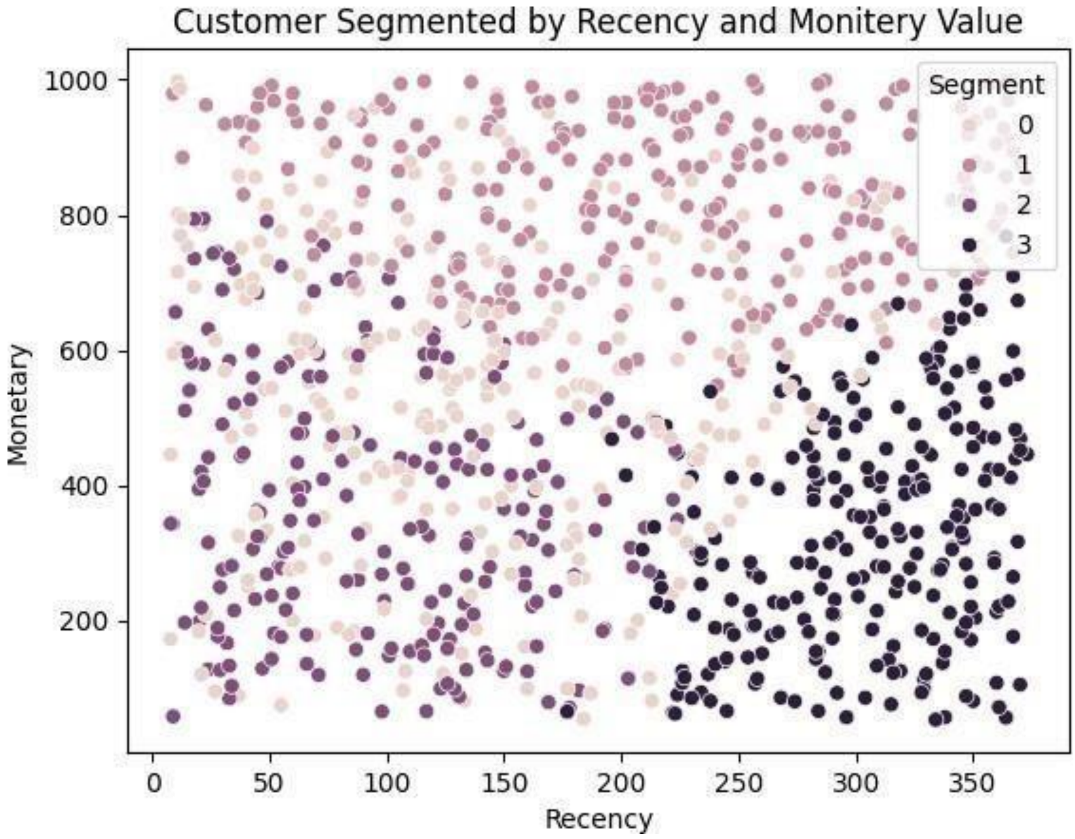
```
import pandas as pd
 data = pd.read_csv('/content/customer_behavior_data
- customer_behavior_data.csv')
 from     datetime     import

datetime    current_date    =

datetime.now()

data['LastPurchaseDate'] = pd.to_datetime(data['LastPurchaseDate'])
 data['Recency'] = (current_date -
data['LastPurchaseDate']).dt.days
 data = data.rename(columns={'PurchaseFrequency':
'Frequency',
'MonetaryValue':        'Monetary'})        from

sklearn.preprocessing import StandardScaler

scaler       =       StandardScaler()       rfm_scaled       =
scaler.fit_transform(data[['Recency',              'Frequency',
'Monetary']])
 from sklearn.cluster import
KMeans
 kmeans  = KMeans(n_clusters=4,  random_state=42)

data['Segment'] = kmeans.fit_predict(rfm_scaled)

print(data['Segment'].value_counts())

segment_analysis = data.groupby('Segment').agg({
    'Recency': 'mean',
    'Frequency': 'mean',
    'Monetary': 'mean',
    'CustomerID': 'count'
}).rename(columns={'CustomerID': 'Count'}).reset_index()

print(segment_analysis)

import matplotlib.pyplot as plt import
seaborn as sns
```

```
sns.scatterplot(x='Recency', y='Monetary', hue='Segment',data=data)
plt.title('Customer  Segmented  by  Recency  and  Monitery  Value')
plt.show()
```

Output:-



```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:1416:
  super()._check_params_vs_input(X, default_n_init=10)
Segment
0    276
1    259
3    247
2    218
Name: count, dtype: int64
   Segment     Recency  Frequency   Monetary  Count
0        0  141.804348   4.278986  544.576812    276
1        1  214.764479  13.401544  829.444788    259
2        2   97.706422  15.137615  371.841560    218
3        3  303.465587  10.842105  329.254049    247
```

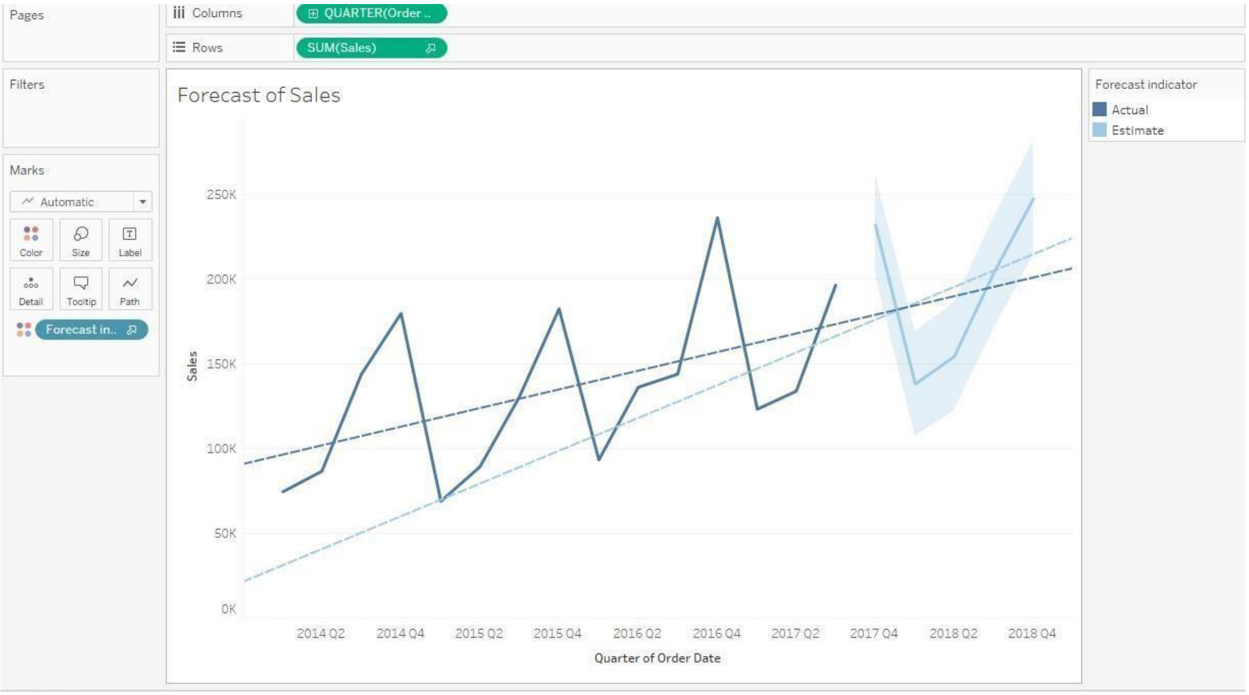


Name: Sakshi Sawant
Roll No. 6957
MSC IT Part - 2

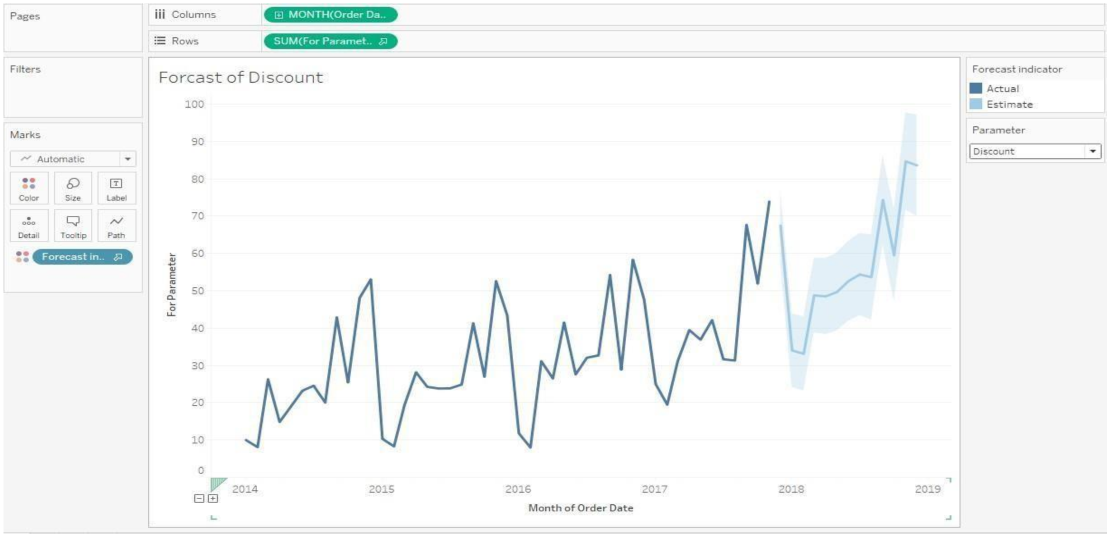
Practical 04

Aim – Understanding analytics in business intelligence tool .

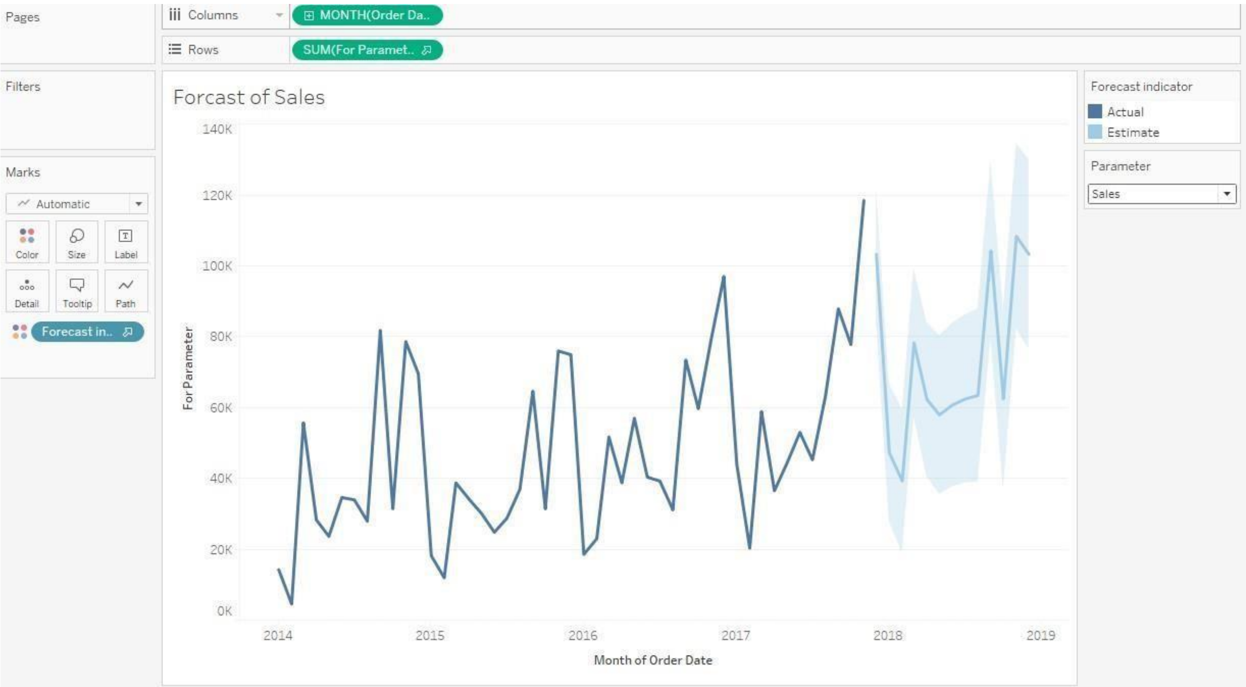
1) Predictive analysis – forcasting the sales of next 4 quarters .


Columns: QUARTER(Order ..
Rows: SUM(Sales)
Marks: Automatic
Forecast in..
Forecast of Sales
Forecast indicator: Actual, Estimate

2) Using a parameter to change the measure (user-defined)


Columns: MONTH(Order Da..
Rows: SUM(For Paramet..
Marks: Automatic
Forecast in..
Forcast of Discount
Forecast indicator: Actual, Estimate
Parameter: Discount

Name: Sakshi Sawant
Roll No. 6957
MSC IT Part - 2

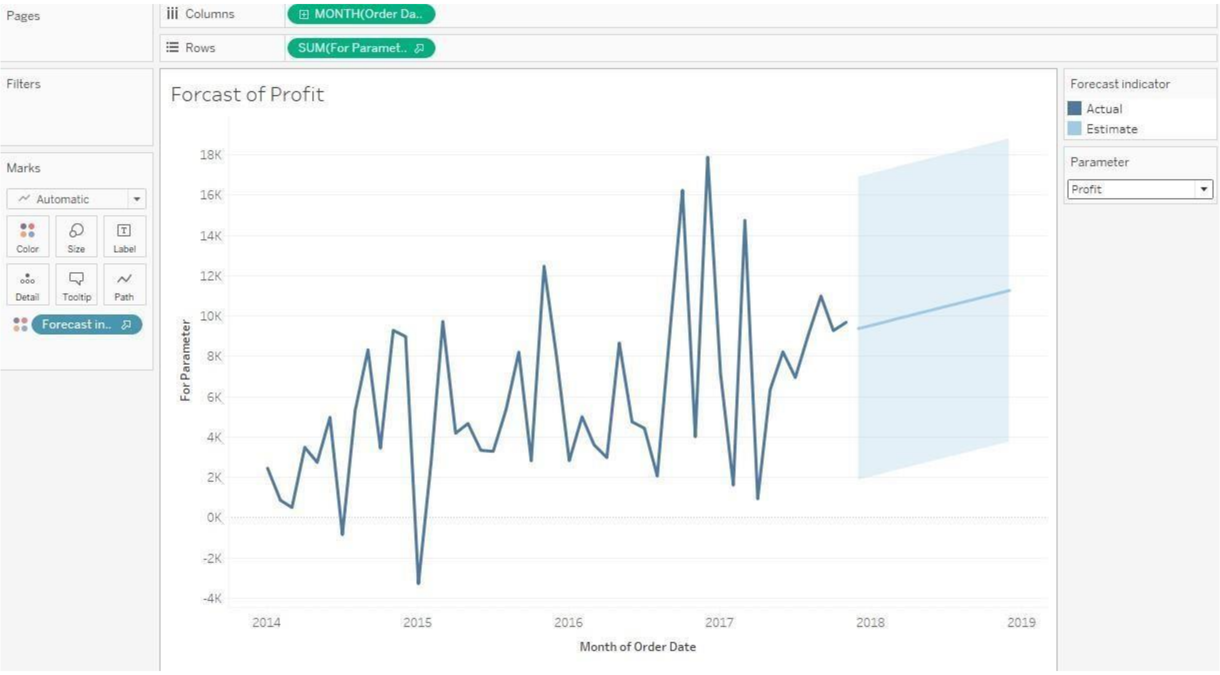


Columns: MONTH(Order Da..

Rows: SUM(For Paramet..

Forecast indicator: Actual, Estimate

Parameter: Sales

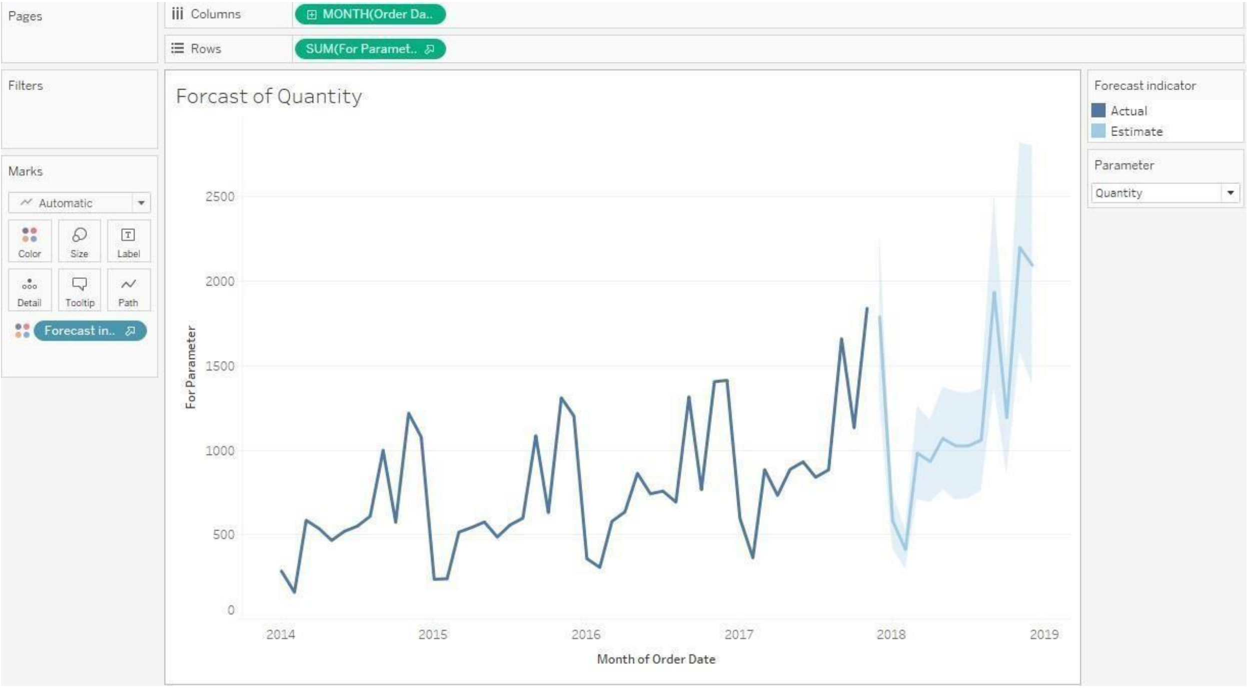


Columns: MONTH(Order Da..

Rows: SUM(For Paramet..

Forecast indicator: Actual, Estimate

Parameter: Profit



Columns: MONTH(Order Da..

Rows: SUM(For Paramet..

Marks: Automatic; Color: Forecast in..

Parameter: Quantity

3. Taking interpretations out of the visualizations .

The visualizations provide insights into sales trends, highlighting patterns like growth, decline, or seasonality. Identifying these trends can help forecast future sales. Outliers should be analyzed carefully as they may represent unique events impacting sales. Comparative analysis of different metrics, such as regions or products, can reveal areas of strength or weakness. Using predictive models like regression will help project sales for the next four quarters, allowing for informed decision-making based on historical data.

Name: Sakshi Sawant
Roll No. 6957
MSC IT Part - 2