

|                                      |   |
|--------------------------------------|---|
| <b>EXP:8</b><br><br><b>17/4/2025</b> | <b>ARIMA model for time series forecasting.</b> |
|--------------------------------------|---|

**AIM:**

To implement a program for ARIMA model for time series forecasting.

**PROCEDURE:**

```
# Step 1: Install required library
!pip install statsmodels

# Step 2: Import libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from statsmodels.tsa.arima.model import ARIMA

# Step 3: Upload the CSV
from google.colab import files
uploaded = files.upload()

# Step 4: Load dataset
filename = list(uploaded.keys())[0]
df = pd.read_csv(filename)
df.columns = ['Month', 'Sales']
df['Sales'] = pd.to_numeric(df['Sales'], errors='coerce')
df = df.dropna()

# Step 5: Fix datetime index
df['Month'] = pd.date_range(start='2001-01-01', periods=len(df),
                             freq='M')
df.set_index('Month', inplace=True)

# Step 6: Fit ARIMA model (p=1, d=1, q=1)
model = ARIMA(df['Sales'], order=(1, 1, 1))
arima_result = model.fit()
```

```
# Step 7: Forecast next 12 months
forecast = arima_result.forecast(steps=12)
forecast_index = pd.date_range(df.index[-1] + pd.DateOffset(months=1),
                                periods=12, freq='M')

# Step 8: Plot actual and forecasted sales
plt.figure(figsize=(12, 6))
plt.plot(df.index, df['Sales'], label='Actual Sales')
plt.plot(forecast_index, forecast, label='Forecasted Sales',
         linestyle='--')
plt.title('ARIMA Forecast for Shampoo Sales')
plt.xlabel('Date')
plt.ylabel('Sales')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()

# Step 9: Display model summary
print(arima_result.summary())
```

## **OUTPUT:**

