**Develop vector auto regression model for multivariate time series data forecasting.**

**EX:No.10 DATE:12/04/25**

# AIM:

To Develop vector auto regression model for multivariate time series data forecasting.

# ALGORITHM:

## **Load and preprocess data**: Import the dataset, convert 'Year' and 'Month' to datetime, set it as index, and drop unnecessary columns.

1. **Handle missing values**: Fill any missing values using forward fill to maintain continuity.

## **Split dataset**: Divide the data into training (80%) and testing (20%) sets.

1. **Fit VAR model**: Initialize the VAR model on training data, select optimal lag order (e.g., using AIC), and fit the model.
2. **Forecast future values**: Use the fitted model to forecast the same number of steps as in the test set.
3. **Evaluate and visualize**: Plot actual vs forecasted values for each variable to assess model performance.

# CODE:

import pandas as pd

import numpy as np

from statsmodels.tsa.api import VAR

import matplotlib.pyplot as plt

filename = list(uploaded.keys())[0]

data = pd.read\_csv(Superstoredata.csv, encoding='latin1')

data['Order Date'] = pd.to\_datetime(data['Order Date'])

data.set\_index('Order Date', inplace=True)

monthly\_data = data.resample('M').sum()[['Sales', 'Profit', 'Quantity']]

monthly\_data = monthly\_data.fillna(method='ffill')

train\_size = int(len(monthly\_data) \* 0.8)

train = monthly\_data.iloc[:train\_size]

test = monthly\_data.iloc[train\_size:]

model = VAR(train)

lag\_order = model.select\_order().aic # you can use .bic or .hqic instead

model\_fitted = model.fit(lag\_order)

forecast\_input = train.values[-lag\_order:]

forecast\_steps = len(test)

forecast = model\_fitted.forecast(y=forecast\_input, steps=forecast\_steps)

forecast\_df = pd.DataFrame(forecast, index=test.index, columns=monthly\_data.columns)

for col in monthly\_data.columns:

plt.figure(figsize=(10, 4))

plt.plot(test.index, test[col], label='Actual')

plt.plot(forecast\_df.index, forecast\_df[col], label='Forecast')

plt.title(f'{col} Forecast')

plt.xlabel('Date')

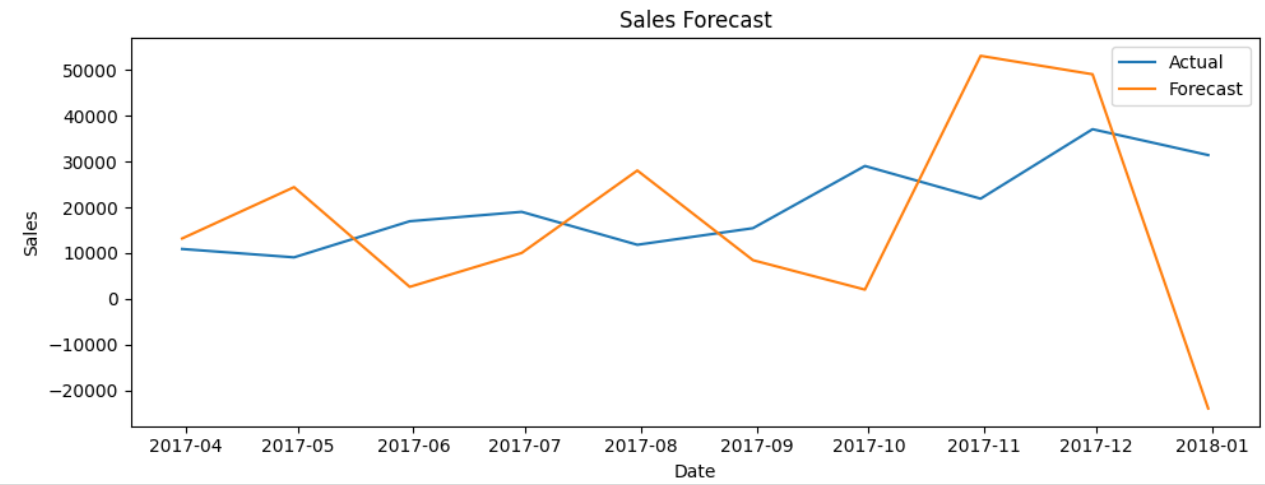
plt.ylabel(col)

plt.legend()

plt.tight\_layout()

plt.show()

# OUTPUT:



**RESULT:**

Thus, the program using the time series data implementation has been done successfully.



Time Series Analysis and Forecasting



221501047