**Exp:10**

**24.04.2025**

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

**Aim:**

To develop a python program for Developing a vector auto regression model for multivariate time series data forecasting

**1. Importing Required Libraries**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from sklearn.metrics import mean\_squared\_error

**Explanation:**

We import numpy (np) is used for numerical operations, pandas (pd) for data manipulation, matplotlib.pyplot (plt) for plotting.

Adfuller for stationary check

**2. Loading the Dataset**

file\_path = "/mnt/data/gold.csv"

df = pd.read\_csv(file\_path)

**Explanation:**

We use pd.read\_csv() to load a CSV file containing Gold data.

**3. Display the first few rows to understand the structure**

print(data.head())

print(data.info())

**4.** **Preprocessing the data**

data = pd.read\_csv('/content/gold (1).csv'

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

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

### 5.Create lagged data

def create\_lagged\_features(data, lags):

    lagged\_data = {}

    for lag in range(1, lags + 1):

        lagged\_data[f"lag\_{lag}"] = data.shift(lag)

    return pd.concat(lagged\_data.values(), axis=1)

**6.Split data**

def create\_lagged\_features(data, lags):

    lagged\_data = {}

    for lag in range(1, lags + 1):

        lagged\_data[f"lag\_{lag}"] = data.shift(lag)

    return pd.concat(lagged\_data.values(), axis=1)

**7.** **Visualization**

plt.figure(figsize=(12, 8))

for i, col in enumerate(selected\_columns):

    plt.subplot(len(selected\_columns), 1, i + 1)

    plt.plot(y\_test.iloc[:, i].values, label='Actual')

    plt.plot(y\_pred[:, i], label='Predicted')

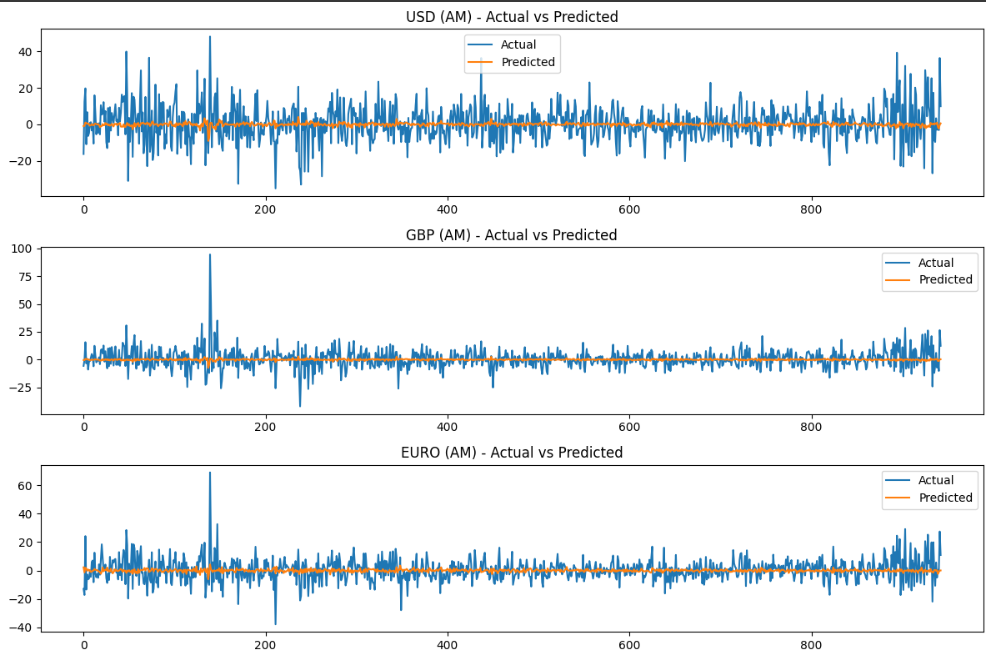
    plt.title(f"{col} - Actual vs Predicted")

    plt.legend()

plt.tight\_layout()

plt.show()

**Outputs:**

****

**Result:**

Thus the Program for developing a vector auto regression model for multivariate time series data forecasting