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

EX.N0 : 10	**Develop vector auto regression model for multivariate time series data forecasting.**
DATE : 12/04/2025	

AIM:

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

PROGRAM:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from statsmodels.tsa.api import VAR
from statsmodels.tsa.stattools import adfuller

file_path = r"D:/221501507/TIME SERIES ANALYSIS AND FORECASTING/EX06/archive (1)
(1)/FINAL_USO.csv"  # ✅ Replace with your multivariate data path
df = pd.read_csv(file_path, parse_dates=['Date'], index_col='Date')
df.columns = df.columns.str.strip()

# ✅ Example: Only select relevant numeric columns for modeling
features = ['Adj Close', 'SP_Ajclose', 'DJ_Ajclose', 'EU_Price', 'OF_Price', 'OS_Price', 'SF_Price',
'USB_Price']
data = df[features].dropna()

def make_stationary(df):
    df_diff = df.copy()
    for col in df.columns:
        result = adfuller(df[col])
        if result[1] > 0.05:
            print(f"{col} is non-stationary. Applying first differencing.")
            df_diff[col] = df[col].diff().dropna()
    return df_diff.dropna()

stationary_data = make_stationary(data)

n_obs = 10  # Days to forecast
train = stationary_data[:-n_obs]
test = stationary_data[-n_obs:]

model = VAR(train)
lag_order = model.select_order(maxlags=15)
selected_lag = lag_order.aic  # you can use 'bic', 'fpe', etc.
```

```python
model_fitted = model.fit(selected_lag)
print(model_fitted.summary())

forecast_input = train.values[-selected_lag:]
forecast_result = model_fitted.forecast(y=forecast_input, steps=n_obs)

forecast_df = pd.DataFrame(forecast_result, index=test.index, columns=train.columns)

last_obs = data.iloc[-n_obs - 1]
reconstructed = forecast_df.cumsum() + last_obs
reconstructed.index = test.index

plt.figure(figsize=(14, 6))
for col in train.columns:
    plt.plot(data[col].iloc[-n_obs:], label=f'Actual {col}', linestyle='--')
    plt.plot(reconstructed[col], label=f'Forecast {col}')
plt.title("VAR Model Forecast vs Actual")
plt.xlabel("Date")
plt.ylabel("Value")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```
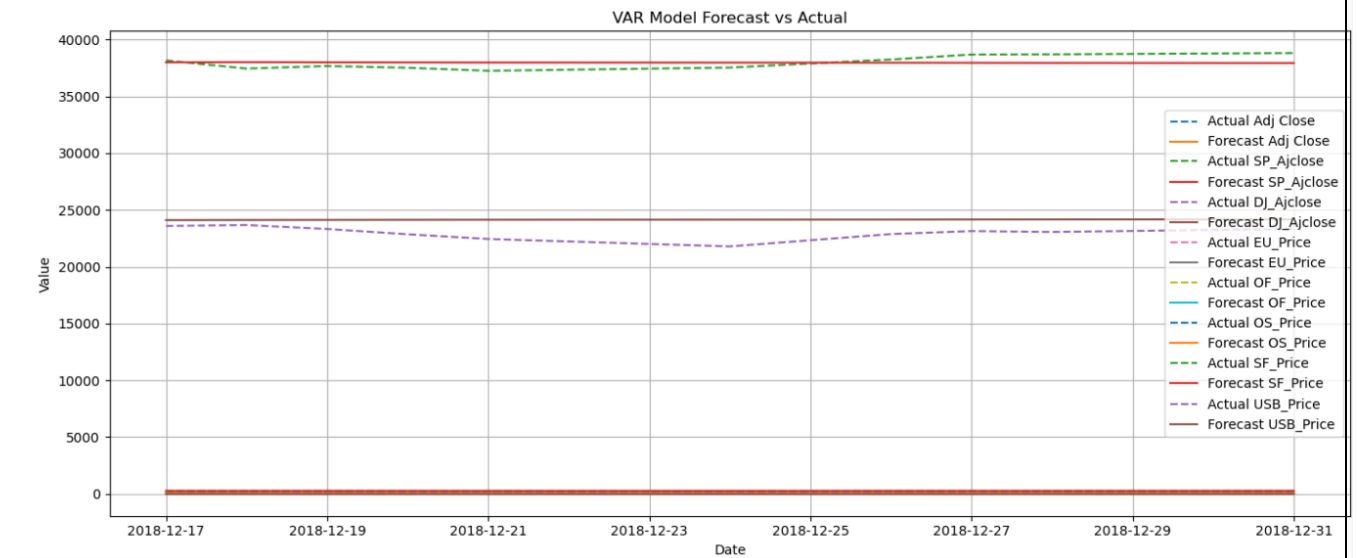
OUTPUT:

VAR Model Forecast vs Actual

RESULT:

Thus, Development vector auto regression model for multivariate time series data forecasting.
is executed successfully.