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

**EX.No:10**

**DATE:**

**AIM:**

To build a VAR model for forecasting multiple related time series variables.

**ALGORITHM:**

1. Load and preprocess multivariate time series data.
2. Split the data into training and testing sets.
3. Fit the VAR model on the training data.
4. Forecast future values using the fitted model.
5. Evaluate and visualize the forecast results.

**CODE:**

import pandas as pd

import numpy as np

from statsmodels.tsa.api import VAR

from sklearn.metrics import mean\_squared\_error

import matplotlib.pyplot as plt

df = pd.read\_csv('Plant\_1\_Generation\_Data.csv')

df['DATE\_TIME'] = pd.to\_datetime(df['DATE\_TIME'], dayfirst=True)

df = df.set\_index('DATE\_TIME').resample('h').mean(numeric\_only=True).fillna(0)

data = df[['DC\_POWER', 'AC\_POWER']]

n\_obs = int(len(data) \* 0.8)

train, test = data[:n\_obs], data[n\_obs:]

model = VAR(train)

model\_fitted = model.fit(maxlags=15, ic='aic')

forecast = model\_fitted.forecast(train.values[-model\_fitted.k\_ar:], steps=len(test))

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

rmse = np.sqrt(mean\_squared\_error(test['DC\_POWER'], forecast\_df['DC\_POWER']))

print(f'RMSE (DC\_POWER): {rmse:.2f}')

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

plt.plot(test['DC\_POWER'], label='Actual DC\_POWER')

plt.plot(forecast\_df['DC\_POWER'], label='Forecast DC\_POWER')

plt.legend()

plt.title('VAR Forecast: DC\_POWER')

plt.show()

**OUTPUT:**



**RESULT:**

Thus the program has been completed and verified successfully.