

EXPERIMENT 10

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

DATE : 17/04/25

AIM :

To develop a python program to implement 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 sklearn.preprocessing import StandardScaler

# Load the dataset
df = pd.read_csv('MLTempDataset (2).csv')
df['Datetime'] = pd.to_datetime(df['Datetime'])
df.set_index('Datetime', inplace=True)

# Create lagged variables to make it multivariate
df['Lag1'] = df['DAYTON_MW'].shift(1)
df['Lag2'] = df['DAYTON_MW'].shift(2)
df.dropna(inplace=True)

# Select the multivariate time series
data = df[['DAYTON_MW', 'Lag1', 'Lag2']]

# Standardize (helps with stability)
scaler = StandardScaler()
scaled_data = scaler.fit_transform(data)
scaled_df = pd.DataFrame(scaled_data, index=data.index, columns=data.columns)

# Fit the VAR model
model = VAR(scaled_df)

# Try lower lag value to avoid instability
results = model.fit(maxlags=5, ic='aic')

# Forecast 24 steps ahead
forecast_input = scaled_df.values[-results.k_ar:]
forecast = results.forecast(y=forecast_input, steps=24)

# Inverse scaling
forecast_df = pd.DataFrame(forecast, columns=scaled_df.columns)
forecast_unscaled = scaler.inverse_transform(forecast_df)
forecast_df_unscaled = pd.DataFrame(forecast_unscaled, columns=data.columns)

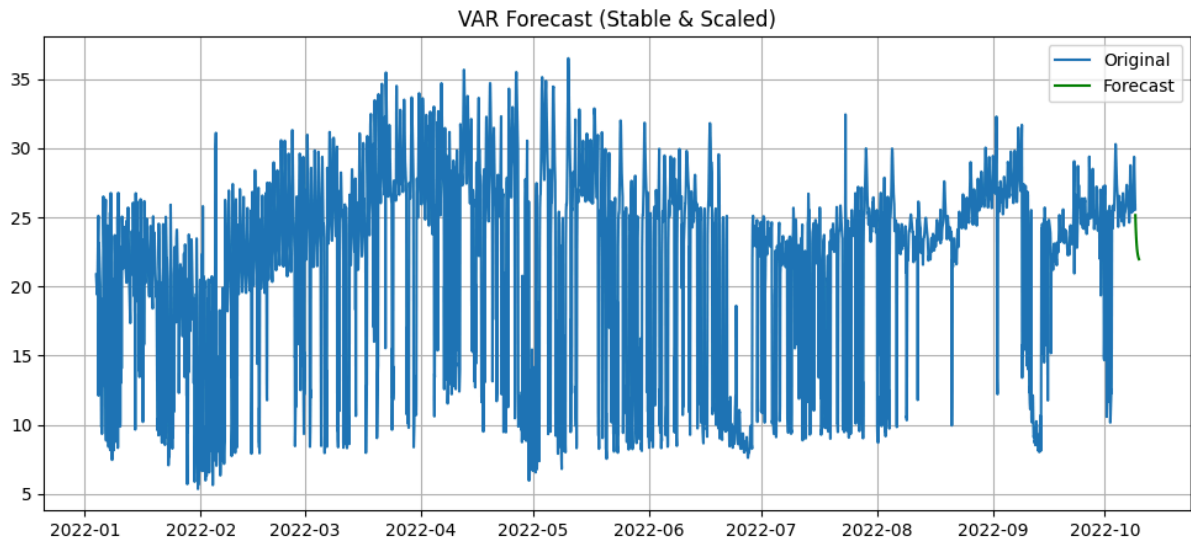
# Plot original and forecasted DAYTON_MW
plt.figure(figsize=(12, 5))
plt.plot(data['DAYTON_MW'], label='Original')
plt.plot(pd.date_range(data.index[-1], periods=25, freq='H')[1:],
```

```

forecast_df_unscaled['DAYTON_MW'],
label='Forecast', color='green')
plt.title("VAR Forecast (Stable & Scaled)")
plt.legend()
plt.grid()
plt.show()

```

OUTPUT :



RESULT :

Thus a python program to develop vector auto regression model for multivariate time series data forecasting .