**Exp no: 10 Develop vector auto regression model for multivariate time series data forecasting**

**Date: 15/4/25**

**Aim:**

To develop and evaluate a Vector AutoRegression (VAR) model for multivariate time series forecasting using the Air Passenger dataset.

**Objectives:**

The key objectives are to preprocess and normalize the Air Passenger dataset along with an additional time series variable, ensure stationarity of the data using statistical methods, build and fit a suitable VAR model, forecast future values (e.g., next 12 months), and finally visualize and evaluate the accuracy of the forecasts.

**Background/Scope:**

Time series forecasting is widely used across industries to predict future trends based on past data. The Vector AutoRegression (VAR) model is a powerful multivariate tool that captures the linear interdependencies among multiple time series. This project uses the classic Air Passenger dataset, extended with a simulated variable (e.g., weather index) to create a multivariate time series. The focus is on building a robust VAR model that not only forecasts passenger numbers but also accounts for external factors influencing air travel.

**Steps of Implementation:**

**Step 1: Load and Prepare the Multivariate Dataset**

Import the Air Passenger dataset and create a second variable (e.g., Weather Index) to make it multivariate.

import pandas as pd

import numpy as np

# Load dataset

url = 'https://raw.githubusercontent.com/jbrownlee/Datasets/master/airline-passengers.csv'

df = pd.read\_csv(url, parse\_dates=['Month'], index\_col='Month')

# Add a synthetic variable (Weather Index)

np.random.seed(0)

df['Weather\_Index'] = df['Passengers'] + np.random.normal(0, 10, len(df))

**Step 2: Normalize the Data**

We normalize the data using **MinMaxScaler** to scale the values between 0 and 1. This is essential for neural networks to perform optimally.

import matplotlib.pyplot as plt

from sklearn.preprocessing import MinMaxScaler

# Normalize

scaler = MinMaxScaler()

scaled\_data = scaler.fit\_transform(df)

df\_scaled = pd.DataFrame(scaled\_data, columns=df.columns, index=df.index)

# Visualize

df.plot(figsize=(10, 5), title="Passengers vs Weather Index")

plt.show()

A graph with blue and orange lines

AI-generated content may be incorrect.

**Step 3: Test for Stationarity and Difference the Data**

Check for stationarity using the Augmented Dickey-Fuller test and apply differencing to make the data stationary.

from statsmodels.tsa.stattools import adfuller

def test\_stationarity(timeseries):

result = adfuller(timeseries)

return result[1] # p-value

# Apply differencing

df\_diff = df\_scaled.diff().dropna()

**Step 4: Fit the VAR Model**

Use the VAR model from statsmodels to fit the differenced multivariate time series.

from statsmodels.tsa.api import VAR

# Fit VAR model

model = VAR(df\_diff)

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

**Step 5: Forecast Future Values**

Forecast future values using the trained VAR model by providing the required lagged values.

forecast\_steps = 12

forecast = results.forecast(df\_diff.values[-results.k\_ar:], steps=forecast\_steps)

forecast\_df = pd.DataFrame(forecast, index=pd.date\_range(start=df.index[-1], periods=forecast\_steps+1, freq='MS')[1:], columns=df.columns)

**Step 6: Reverse Transform and Visualize Forecast**

Reverse the differencing and normalization to get forecasted values in the original scale, then visualize the results.

# Reverse differencing

last\_values = df\_scaled.iloc[-1]

forecast\_reversed = forecast\_df.cumsum() + last\_values

# Inverse normalization

forecast\_original = pd.DataFrame(scaler.inverse\_transform(forecast\_reversed),

columns=df.columns, index=forecast\_reversed.index)

# Plot Forecast

plt.figure(figsize=(10, 5))

plt.plot(df.index, df['Passengers'], label='Original')

plt.plot(forecast\_original.index, forecast\_original['Passengers'], label='Forecast', linestyle='--')

plt.title("Air Passenger Forecast with VAR Model")

plt.xlabel("Date")

plt.ylabel("Passengers")

plt.legend()

plt.show()

**Output:**

A graph showing the growth of an airplane

AI-generated content may be incorrect.

**Result:**

Vector auto regression model for multivariate time series data forecasting for air passenger dataset has been successfully implemented.