|  |  |
| --- | --- |
| **EX.N0 : 8** | **Create an ARIMA model for time series forecasting.** |
| **DATE : 24/03/2025** |

**AIM:**

# Create an ARIMA model for time series forecasting.

# ALGORITHM:

# Step 1: Import Libraries Step 2: Load the Dataset Step 3: Preprocess the Data Step 4: Fit ARIMA Model Step 5: Forecast Future Values Step 6: Plot Forecast

# PROGRAM:

# Step 1: Import Libraries

import pandas as pd

import matplotlib.pyplot as plt

from statsmodels.tsa.arima.model import ARIMA

import warnings

warnings.filterwarnings('ignore')

# Step 2: Load the Dataset

file\_path = '/mnt/data/climate\_change\_data.csv' # Change path if needed

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

# Step 3: Preprocess the Data

df.columns = [col.strip() for col in df.columns]

date\_col = df.columns[0]

df[date\_col] = pd.to\_datetime(df[date\_col], errors='coerce')

df.set\_index(date\_col, inplace=True)

# Select numeric columns only

df\_numeric = df.select\_dtypes(include='number')

if df\_numeric.empty:

raise ValueError("No numeric columns found!")

# Use first numeric column

value\_col = df\_numeric.columns[0]

data = df\_numeric[value\_col].dropna()

# Optional: Plot original data

plt.figure(figsize=(12, 5))

plt.plot(data, label='Original Data')

plt.title('Original Time Series')

plt.grid(True)

plt.legend()

plt.show()

# Step 4: Fit ARIMA Model

# (p,d,q) values should ideally be determined via AIC or grid search; we use a sample here

model = ARIMA(data, order=(1, 1, 1)) # Change to (p,d,q) as needed

model\_fit = model.fit()

# Step 5: Forecast Future Values

forecast\_steps = 12 # Forecast next 12 periods

forecast = model\_fit.forecast(steps=forecast\_steps)

# Create index for forecast

last\_date = data.index[-1]

forecast\_index = pd.date\_range(start=last\_date, periods=forecast\_steps+1, freq='M')[1:]

forecast\_series = pd.Series(forecast, index=forecast\_index)

# Step 6: Plot Forecast

plt.figure(figsize=(12, 5))

plt.plot(data, label='Original')

plt.plot(forecast\_series, label='Forecast (ARIMA)', color='red', linestyle='--')

plt.title('ARIMA Forecast')

plt.grid(True)

plt.legend()

plt.show()

**OUTPUT:**

A blue line graph on a white background

AI-generated content may be incorrect.

A blue line graph on a white background

AI-generated content may be incorrect.

# RESULT:

Thus, the program for Implement programs for a time series data is executed successfully.