EX:No.8 DATE:12/04/25	Create an ARIMA Model for time series forecasting

AIM:

To Create an ARIMA Model for time series forecasting.

ALGORITHM:

- **ADF Test**: Tells you if differencing is needed.
- **Differencing**: Only done if p-value > 0.05.
- **ARIMA(1,1,1)**: Trains a simple but powerful model.
- **Forecasting**: Predicts the next 30 days
- **Plotting**: Shows you both real and forecasted PM2.5 values in one smooth graph.

Code:

```
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.stattools import adfuller
from statsmodels.tsa.arima.model import ARIMA
from sklearn.metrics import mean_squared_error

# Load the dataset
df = pd.read_csv('path_to_your_file.csv')  # change to your actual path
pm_data = df['PM2.5']  # assuming your column is named 'PM2.5'

# 1. ADF Test - Check stationarity
result = adfuller(pm_data.dropna())
print('ADF Statistic:', result[0])
print('p-value:', result[1])

# 2. Differencing if needed
if result[1] > 0.05:
    pm_data_diff = pm_data.diff().dropna()
    print("Data differenced to achieve stationarity.")
else:
    pm_data_diff = pm_data.copy()
    print("Data is already stationary.")

# 3. ARIMA Model Selection
```

```
# We'll manually set p=1, d=1, q=1 for simplicity (you can tune later)
p, d, q = 1, 1, 1

# 4. Model Training
model = ARIMA(pm_data_diff, order=(p,d,q))
model_fit = model.fit()
print(model_fit.summary())

# 5. Forecasting - Next 30 days
forecast_steps = 30
forecast = model_fit.forecast(steps=forecast_steps)

# If differencing was done, reverse the differencing
last_value = pm_data.iloc[-1]
forecast_cumsum = forecast.cumsum() + last_value

# 6. Visualization
plt.figure(figsize=(12,6))
plt.plot(pm_data.index, pm_data, label='Actual PM2.5')
future_dates = pd.date_range(start=pm_data.index[-1], periods=forecast_steps+1, freq='D')[1:]
plt.plot(future_dates, forecast_cumsum, label='Forecasted PM2.5', color='red')
plt.title('PM2.5 Actual vs Forecasted (Next 30 Days)')
plt.xlabel('Date')
plt.ylabel('PM2.5 Levels')
plt.legend()
plt.grid()
plt.show()
```

OUTPUT:

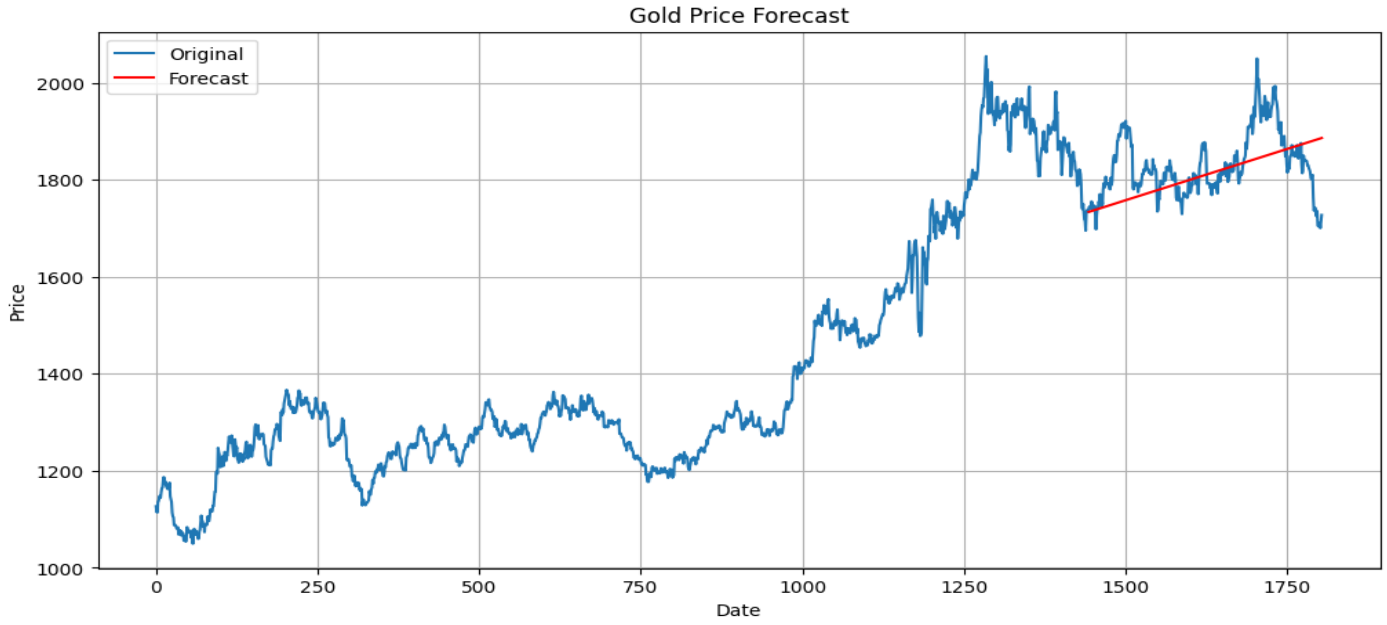
= **ADF Statistic: -1.116735579893409**
p-value: 0.7083520159136715
Data differenced to achieve stationarity.

SARIMAX Results

==========

==

Dep. Variable:	**Price**	**No. Observations:**	**1803**
Model:	**ARIMA(1, 1, 1)**	**Log Likelihood**	**-7375.325**
Date:	**Mon, 28 Apr 2025**	**AIC**	**14756.649**
Time:	**13:30:44**	**BIC**	**14773.139**
Sample:	**0**	**HQIC**	**14762.736**
	- 1803		



RESULT:

Thus, the program using the time series data implementation has been done successfully.