**Create an ARIMA model for time series forecasting**

**AIM:**

To implement program to create an ARIMA model for time series forecasting

using jupyter notebook.

**ALGORITHM:**

1. Load the taxi dataset then clean and load the values

2. Fit an ARIMA model to the differenced time series and forecast future values based on autoregressive and moving average components.

3. Plot the graph and visualize the values

**CODE:**

import pandas as pd

import matplotlib.pyplot as plt

from statsmodels.tsa.arima.model import ARIMA

import warnings

warnings.filterwarnings("ignore")

# Load and prepare dataset

df = pd.read\_csv("dataset.csv")

df["datetime"] = pd.to\_datetime(df[["year", "month", "day"]]) + pd.to\_timedelta(df["hour\_of\_day"], unit='h')

df.set\_index("datetime", inplace=True)

# Aggregate to daily average

daily\_avg = df["total\_amount"].resample("D").mean().dropna()

# Fit ARIMA model (order can be tuned — here p=5, d=1, q=0 as a starting point)

model = ARIMA(daily\_avg, order=(5, 1, 0)) # (AR, differencing, MA)

model\_fit = model.fit()

# Forecast next 7 days

Forecast = model\_fit.forecast(steps=7)

# Plot original and forecast

plt.figure(figsize=(14, 6))

plt.plot(daily\_avg, label="Historical Data", alpha=0.6)

plt.plot(forecast.index, forecast, label="ARIMA Forecast (7 days)", color='red', linestyle='--')

plt.title("ARIMA Time Series Forecast")

plt.xlabel("Date")

plt.ylabel("Total Amount ($)")

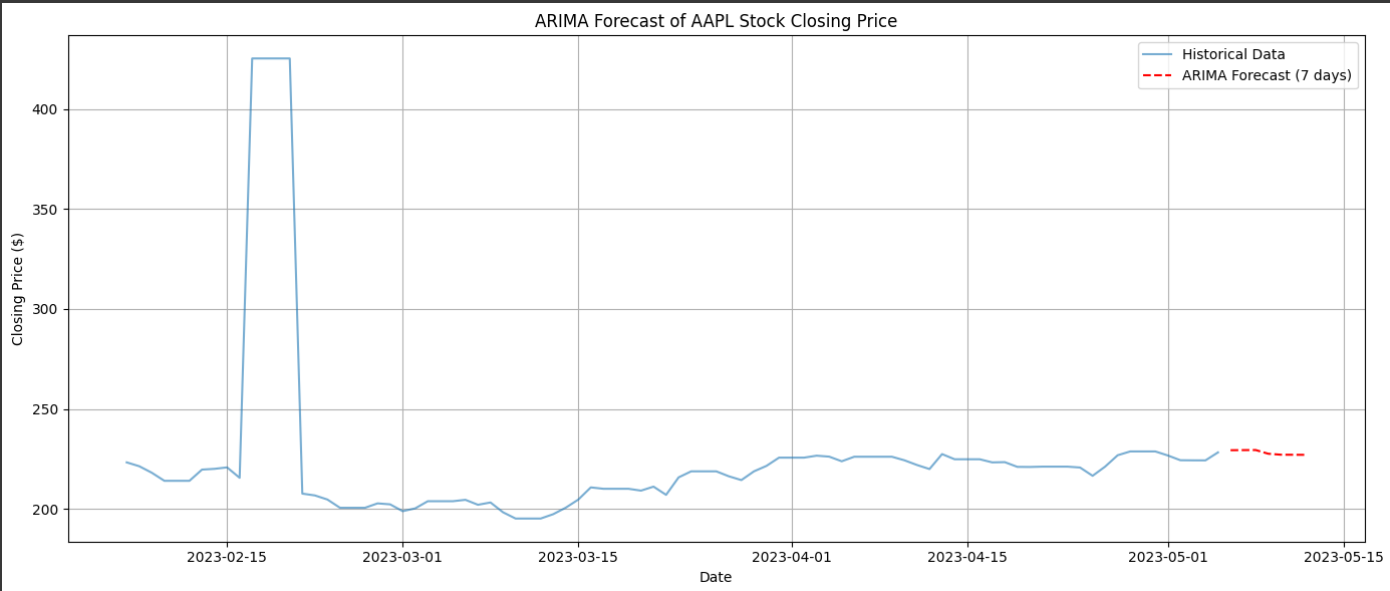
plt.legend()

plt.grid(True)

plt.tight\_layout()

plt.show()

**OUTPUT:**

****

**RESULT:**

The program to create an ARIMA model for time series forecasting completed successfully and the output is verified.