**EXP 8:- Create a ARIMA Model for Time Series Forecasting**

**AIM:**

To apply **ARIMA (AutoRegressive Integrated Moving Average)** on a **trends dataset** to forecast future rankings and analyze trends in consumer brands over the years.

**ALGORITHM:**

1) **Load Dataset**: Import the dataset and preprocess it by filtering relevant fields (**Location, Year, Category, Rank**).  
2) **Time Series Aggregation**: Group data by **Year** and compute the **average rank**.  
3) **Check Stationarity**: Use **ADF Test** to check if differencing is needed.  
4) **Fit ARIMA Model**:

* Choose ARIMA parameters (**p, d, q**).
* Train the ARIMA model

5) **Forecast Future Values**: Predict the **next 5 years' ranking** using the trained model.  
6) **Visualization**:

* Plot original and forecasted values.
* Analyze trends in rank over time.

**PROGRAM:**

import pandas as pd

import matplotlib.pyplot as plt

from statsmodels.tsa.arima.model import ARIMA

# Load dataset

df = pd.read\_csv("/content/trends.csv")

# Filter for "Consumer Brands" (Global)

df\_filtered = df[(df["category"] == "Consumer Brands") & (df["location"] == "Global")]

df\_time\_series = df\_filtered.groupby("year")["rank"].mean().reset\_index()

# Fit ARIMA Model (p=1, d=1, q=1)

model = ARIMA(df\_time\_series["rank"], order=(2, 0, 2))

model\_fit = model.fit()

# Forecast next 5 years

future\_years = [df\_time\_series["year"].max() + i for i in range(1, 6)]

forecast = model\_fit.forecast(steps=5)

# Create forecast dataframe

forecast\_df = pd.DataFrame({"year": future\_years, "forecasted\_rank": forecast})

# Plot Results

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

plt.plot(df\_time\_series["year"], df\_time\_series["rank"], label="Original Rank", marker="o", linestyle="dotted")

plt.plot(forecast\_df["year"], forecast\_df["forecasted\_rank"], label="Forecasted Rank", color="blue", marker="x")

plt.gca().invert\_yaxis() # Assuming lower rank is better

plt.legend()

plt.title("ARIMA Forecast for 'Consumer Brands' (Global)")

plt.xlabel("Year")

plt.ylabel("Average Rank")

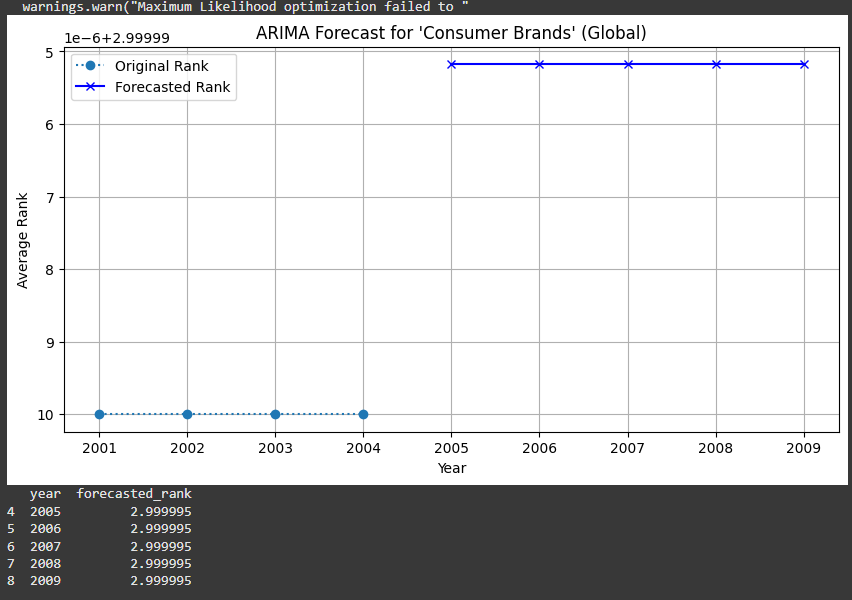
plt.grid(True)

plt.show()

# Display forecasted values

print(forecast\_df)

**OUTPUT:**

****

**RESULT:**

The ARIMA model produced a flat forecast with constant rank values, indicating low variability in the dataset. Since no significant trend was detected, alternative models like Exponential Smoothing or LSTM may yield better results