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

**AIM:**

To apply **ARIMA (AutoRegressive Integrated Moving Average)** on COVID 19 to forecast future rankings and analyze trends in consumer brands over the years.

**ALGORITHM:**

1. **Import Libraries**:
   * Import the necessary libraries: pandas, matplotlib.pyplot, and ARIMA from statsmodels.tsa.arima.model.
2. **Load and Clean the Dataset**:
   * Load the COVID-19 dataset using pandas.read\_csv().
   * Clean the column names by stripping any leading or trailing spaces.
   * Convert the Date column to datetime format using pd.to\_datetime().
   * Set the Date column as the index for time series analysis.
3. **Compute Daily New Cases**:
   * Group the dataset by date and sum the numeric columns.
   * Calculate the **daily new cases** by taking the difference between consecutive days (diff()).
   * Drop any NaN values that arise after the diff() operation (the first row will have NaN).
4. **Resample to Monthly Data**:
   * Resample the daily data to monthly frequency using .resample("M") and compute the **average daily cases per month** using .mean().
   * Reset the index to make the Date column accessible.
   * Add a new column month to represent the month in a string format (e.g., "2020-01").
5. **Fit ARIMA Model**:
   * Fit an ARIMA model to the monthly average daily cases using ARIMA() with the specified parameters (p=2, d=0, q=2).
   * Use model.fit() to fit the model to the data.
6. **Forecast the Next 6 Months**:
   * Use the forecast() method from the ARIMA model to predict the next 6 months of daily cases.
   * Generate a date range for the next 6 months (pd.date\_range()) and format them as period-based months.
7. **Prepare Forecast Data**:
   * Create a DataFrame containing the forecasted values and their corresponding future months.
8. **Plot the Data**:
   * Plot the original monthly average daily cases with plt.plot(), using a dashed line and markers.
   * Plot the forecasted monthly average daily cases with a solid line and different markers.
   * Customize the plot by setting the title, labels, and rotating the x-axis ticks for better readability.
   * Add a legend and grid, then display the plot using plt.show().
9. **Print the Forecasted Values**:
   * Print the forecasted monthly average daily cases to the console.

**PROGRAM:**

import pandas as pd

import matplotlib.pyplot as plt

from statsmodels.tsa.arima.model import ARIMA

# Load the dataset

df = pd.read\_csv("/content/time-series-19-covid-combined.csv")

# Clean column names and parse date

df.columns = df.columns.str.strip()

df["Date"] = pd.to\_datetime(df["Date"])

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

# Group by date and compute daily new cases

daily\_df = df.groupby(df.index).sum(numeric\_only=True)

daily\_df["daily\_cases"] = daily\_df["Confirmed"].diff()

daily\_df = daily\_df.dropna(subset=["daily\_cases"])

# Resample monthly and calculate average daily cases per month

monthly\_avg = daily\_df["daily\_cases"].resample("M").mean().reset\_index()

monthly\_avg["month"] = monthly\_avg["Date"].dt.to\_period("M").astype(str)

# Fit ARIMA model

model = ARIMA(monthly\_avg["daily\_cases"], order=(2, 0, 2))

model\_fit = model.fit()

# Forecast next 6 months

forecast = model\_fit.forecast(steps=6)

future\_months = pd.date\_range(start=monthly\_avg["Date"].max() + pd.DateOffset(months=1), periods=6, freq='M')

forecast\_df = pd.DataFrame({"month": future\_months.to\_period("M").astype(str), "forecasted\_daily\_cases": forecast})

# Plotting

plt.figure(figsize=(12, 6))

plt.plot(monthly\_avg["month"], monthly\_avg["daily\_cases"], label="Avg Daily Cases", marker='o', linestyle="--")

plt.plot(forecast\_df["month"], forecast\_df["forecasted\_daily\_cases"], label="Forecasted", color="blue", marker='x')

plt.xticks(rotation=45)

plt.title("ARIMA Forecast of Monthly Avg Daily COVID-19 Cases (Global)")

plt.xlabel("Month")

plt.ylabel("Average Daily Cases")

plt.legend()

plt.tight\_layout()

plt.grid(True)

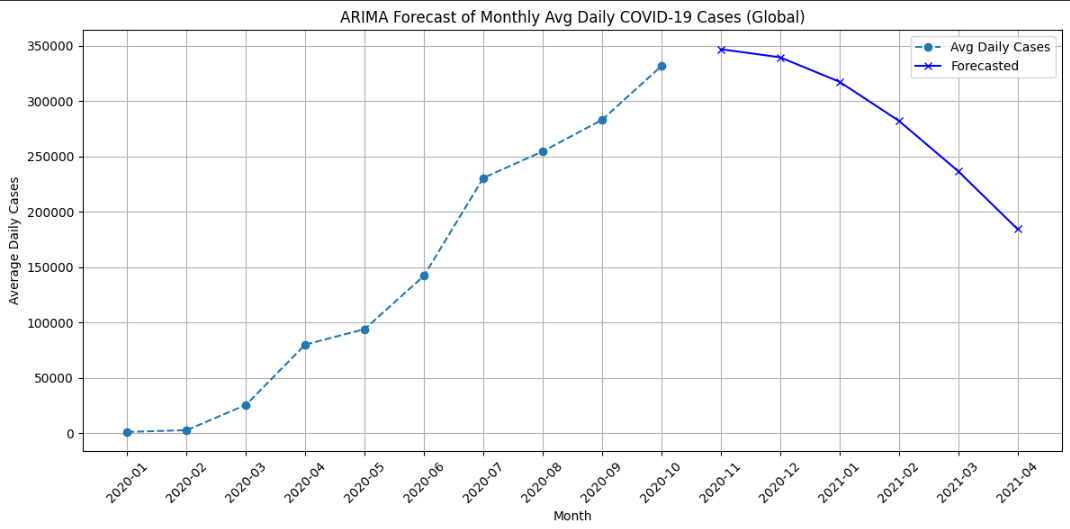
plt.show()

# Print forecasted values

print("\n📅 Forecasted Monthly Average Daily Cases:")

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