**EXP 6:- Implement program to apply moving average smoothing for data preparation and time series forecasting**

**AIM:**

To apply **moving average smoothing** on the **RANK** column over **YEAR** to reduce fluctuations and observe trends in the data.

**ALGORITHM:**

**Step 1: Load the Dataset**

* Import the necessary libraries (pandas, matplotlib).
* Read the dataset (trends.csv).

**Step 2: Preprocess the Data**

* Ensure the dataset has YEAR and RANK columns.
* Sort the dataset by YEAR to maintain chronological order.

**Step 3: Apply Moving Average Smoothing**

* Compute a **5-year Simple Moving Average (SMA)** for RANK.
* Compute a **5-year Exponential Moving Average (EMA)** for RANK.

**Step 4: Plot and Visualize the Data**

* Plot RANK, SMA\_5, and EMA\_5 over YEAR.
* Invert the y-axis (assuming a lower rank is better).

**Step 5: Save the Processed Data**

* Save the new dataset (trends\_smoothed.csv) with SMA and EMA values.

**PROGRAM:**

**OUTPUT:**

import pandas as pd

import matplotlib.pyplot as plt

from statsmodels.tsa.arima.model import ARIMA

# Load the dataset

file\_path = "/content/trends.csv"

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

# Sort by YEAR (if not already sorted)

df = df.sort\_values(by='year')

# Apply a 5-year simple moving average to RANK

df['SMA\_5'] = df['rank'].rolling(window=5).mean()

# Apply Exponential Moving Average (EMA) for better trend smoothing

df['EMA\_5'] = df['rank'].ewm(span=5, adjust=False).mean()

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

model = ARIMA(df["rank"], order=(1, 1, 1))

model\_fit = model.fit()

# Forecast next 5 years

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

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

# Create a dataframe for future predictions

forecast\_df = pd.DataFrame({"YEAR": future\_years, "Forecasted\_RANK": forecast})

# Append forecasts to original data

df\_forecasted = pd.concat([df, forecast\_df], ignore\_index=True)

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

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

plt.plot(df['year'], df['SMA\_5'], label="5-Year SMA", color='red')

plt.plot(df['year'], df['EMA\_5'], label="5-Year EMA", color='green')

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("Rank Forecasting using Moving Averages and ARIMA")

plt.xlabel("Year")

plt.ylabel("Rank")

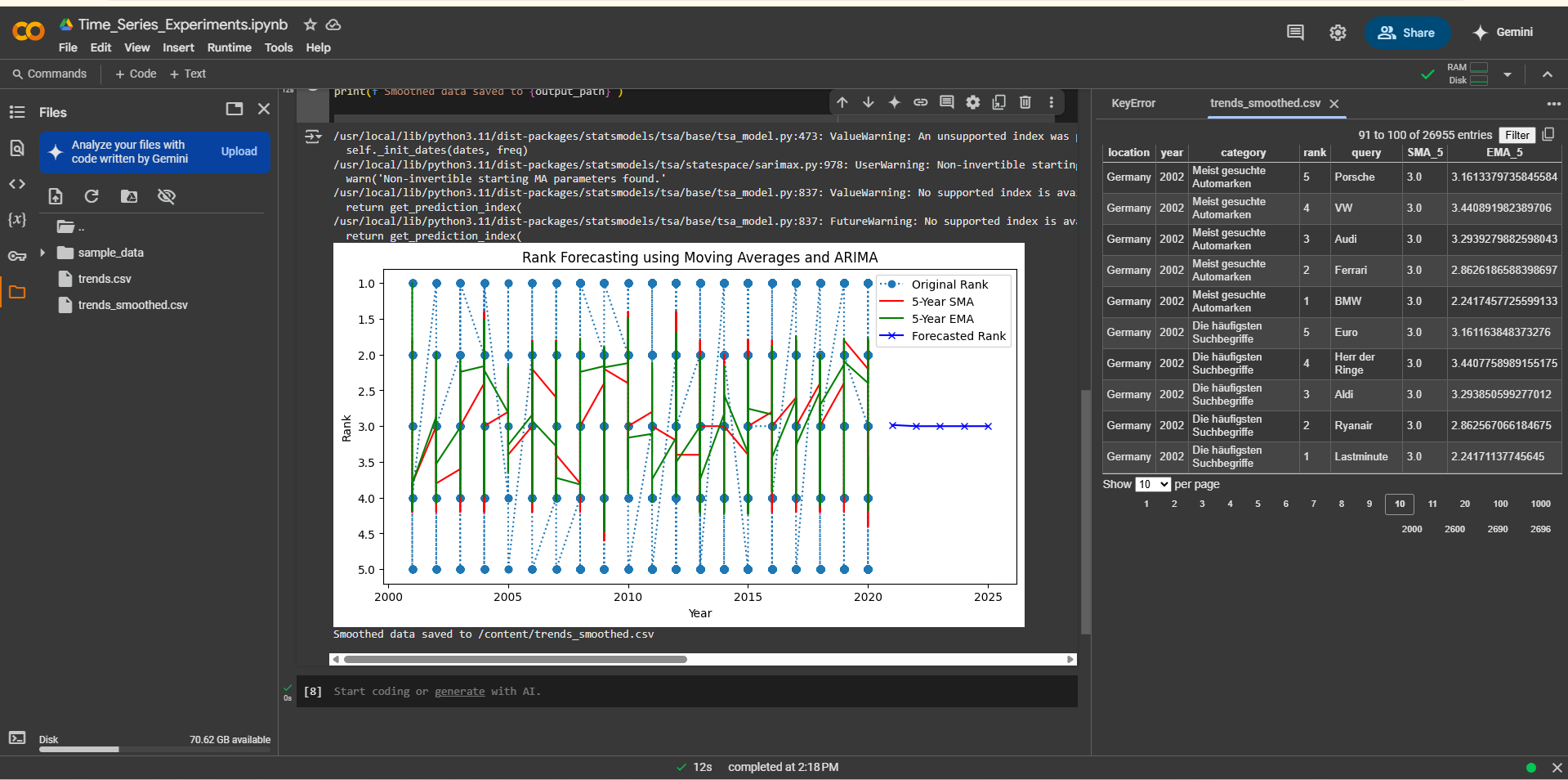
plt.show()

output\_path = "/content/trends\_smoothed.csv"

df.to\_csv(output\_path, index=False

print(f"Smoothed data saved to {output\_path}")

**OUTPUT:**



**RESULT:**

Moving average smoothing effectively reduces fluctuations in the **RANK** column over **YEAR**, highlighting overall trends. The **SMA** provides a stable long-term trend, while the **EMA** quickly adapts to recent changes, making trend analysis and forecasting more reliable.