**3. DEVELOP A LINEAR REGRESSION MODEL FOR FORECASTING TIME SERIES DATA**

221501118

**AIM:**

To develop a Linear Regression model for forecasting time series data using the Google Trends dataset, analyzing search trends over time, and predicting future trends based on historical data.

**PROCEDURE:**

**Import Required Libraries** – Load necessary libraries like Pandas, NumPy, Matplotlib, Seaborn, and Scikit-learn.

**Load the Dataset** – Read the Google Trends dataset into a Pandas DataFrame.

**Data Preprocessing** – Handle missing values, convert the year column to numerical format, and aggregate data if needed.

**Visualize the Data** – Use line plots to observe trends and patterns over time.

**Prepare Data for Modeling** – Split the dataset into independent (X: year) and dependent (y: search rank) variables.

**Split Data into Training and Testing Sets** – Use train\_test\_split() from Scikit-learn.

**Train the Linear Regression Model** – Fit the model using LinearRegression() from Scikit-learn.

**Make Predictions** – Use the trained model to predict future trends.

**Evaluate Model Performance** – Compute metrics like Mean Squared Error (MSE) and R-squared (R²).

**Visualize Predictions** – Plot actual vs. predicted values to assess accuracy.

**Forecast Future Trends** – Extend predictions beyond the available data for future trend forecasting.

**CODE:**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error, r2\_score

df = pd.read\_csv('google\_trends.csv')

df.dropna(inplace=True)

df['year'] = pd.to\_numeric(df['year'])

df = df.groupby('year').mean().reset\_index()

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

sns.lineplot(x=df['year'], y=df['rank'])

plt.xlabel('Year')

plt.ylabel('Search Rank')

plt.title('Google Trends Search Rank Over Time')

plt.show()

X = df[['year']]

y = df['rank']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

model = LinearRegression()

model.fit(X\_train, y\_train)

y\_pred = model.predict(X\_test)

mse = mean\_squared\_error(y\_test, y\_pred)

r2 = r2\_score(y\_test, y\_pred)

print(f"Mean Squared Error: {mse}")

print(f"R-Squared Value: {r2}")

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

plt.scatter(X\_test, y\_test, color='blue', label='Actual')

plt.plot(X\_test, y\_pred, color='red', linewidth=2, label='Predicted')

plt.xlabel('Year')

plt.ylabel('Search Rank')

plt.title('Actual vs Predicted Google Trends Search Rank')

plt.legend()

plt.show()

future\_years = np.arange(df['year'].max() + 1, df['year'].max() + 6).reshape(-1, 1)

future\_predictions = model.predict(future\_years)

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

plt.scatter(df['year'], df['rank'], color='blue', label='Historical Data')

plt.plot(future\_years, future\_predictions, color='green', linestyle='dashed', label='Future Predictions')

plt.xlabel('Year')

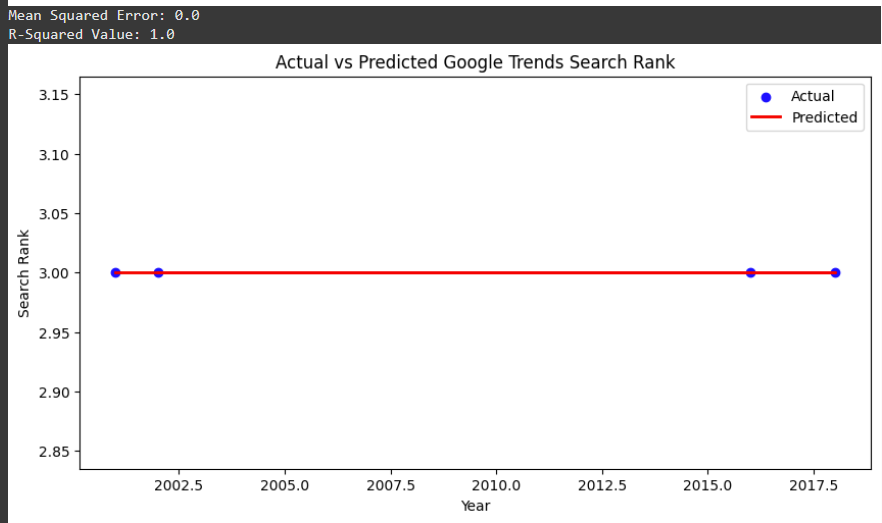
plt.ylabel('Search Rank')

plt.title('Google Trends Forecast for Future Years')

plt.legend()

plt.show()

**Output:**



**RESULT:**

Hence the linear regression model to forecast the time series data is developed Successfully