**Develop neural network-based time series forecasting model.**

**Aim:**

To develop an **neural network model** for forecasting the **rank trends** in a Google Trends dataset using time series analysis.

**Procedure:**

1. Load the Google Trends dataset, convert the year column to datetime, and filter by category.
2. Normalize the rank values using MinMaxScaler to improve neural network training.
3. Transform the time series data into a supervised learning format using a sliding window approach.
4. Build and train an LSTM-based neural network using TensorFlow/Keras.
5. Predict future rank values, inverse transform the predictions, and compare them with actual data.
6. Plot the actual vs. predicted trends to evaluate forecasting accuracy.

**Code :**

!pip install tensorflow --quiet

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import tensorflow as tf

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import LSTM, Dense

from sklearn.preprocessing import MinMaxScaler

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

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

df['year'] = pd.to\_datetime(df['year'], format='%Y')

category = "Movies" # Change this as needed

df\_category = df[df['category'] == category]

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

df\_category.set\_index('year', inplace=True)

scaler = MinMaxScaler(feature\_range=(0, 1))

df\_category['rank\_scaled'] = scaler.fit\_transform(df\_category[['rank']])

def create\_dataset(data, time\_steps=3):

X, y = [], []

for i in range(len(data) - time\_steps):

X.append(data[i : i + time\_steps])

y.append(data[i + time\_steps])

return np.array(X), np.array(y)

time\_steps = 3

X, y = create\_dataset(df\_category['rank\_scaled'].values, time\_steps)

X = X.reshape((X.shape[0], X.shape[1], 1))

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, shuffle=False)

model = Sequential([

LSTM(50, activation='relu', return\_sequences=True, input\_shape=(time\_steps, 1)),

LSTM(50, activation='relu'),

Dense(1)

])

model.compile(optimizer='adam', loss='mse')

model.fit(X\_train, y\_train, epochs=50, batch\_size=8, validation\_data=(X\_test, y\_test), verbose=1)

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

y\_pred\_actual = scaler.inverse\_transform(y\_pred)

y\_test\_actual = scaler.inverse\_transform(y\_test.reshape(-1, 1))

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

plt.plot(df\_category.index[-len(y\_test):], y\_test\_actual, label="Actual Rank", marker='o')

plt.plot(df\_category.index[-len(y\_test):], y\_pred\_actual, label="Predicted Rank", linestyle='dashed', marker='s')

plt.xlabel("Year")

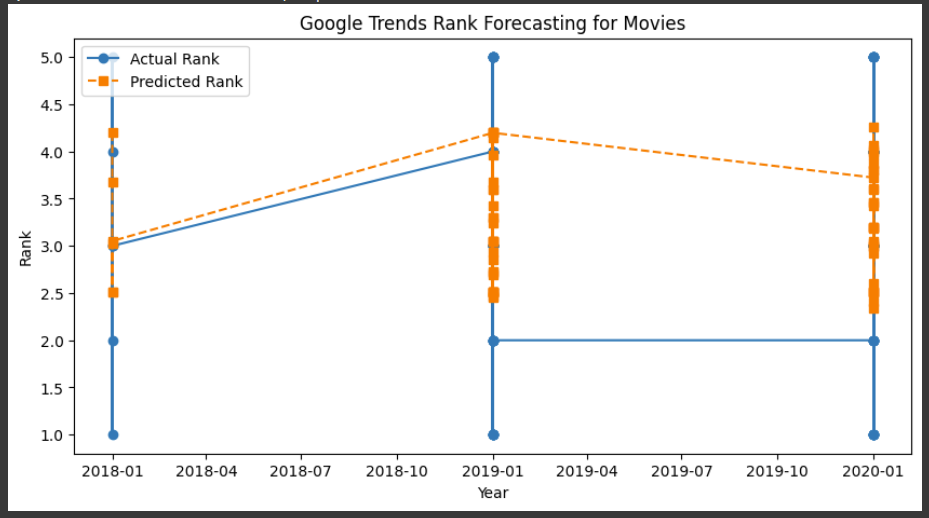
plt.ylabel("Rank")

plt.title(f"Google Trends Rank Forecasting for {category}")

plt.legend()

plt.show()

**Output:**

****

**Result:**

The program to develop a neural network based time series forecasting model has been successfully implemented