EX:No.5 221501042

**Develop a linear regression model for forecasting time series data**

**Aim:**

Write a program to develop a linear regression model for forecasting time series data

**Algorithm:**

1. **Import Required Libraries**
   * Load necessary libraries: numpy, pandas, matplotlib, sklearn.
2. **Load the Dataset**
   * Read the dataset (weather\_data.csv).
   * Set the 'Date' column as the **index**.
3. **Prepare the Data**
   * Create a **time index** (Time\_Index) as a numerical feature for regression.
   * Define **features (X)** as the time index.
   * Define **target variable (y)** as the column to be forecasted (e.g., Temperature).
4. **Split the Data into Training and Testing Sets**
   * Use an **80-20 split** (train\_test\_split), ensuring the order is **not shuffled**.
5. **Train the Linear Regression Model**
   * Initialize the LinearRegression() model.
   * Fit the model using **training data (X\_train, y\_train)**.
6. **Make Predictions on the Test Set**
   * Use model.predict(X\_test) to generate forecasts.
7. **Evaluate the Model's Performance**
   * Compute **Mean Absolute Error (MAE)** and **Root Mean Squared Error (RMSE)**.
8. **Visualize the Results**
   * Plot the **actual values** vs. **predicted values** on a time-series graph.

**Code:**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

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

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_absolute\_error, mean\_squared\_error

# Load the dataset

df = pd.read\_csv(r"C:\Users\harsh\Downloads\cleaned\_weather.csv", parse\_dates=['date'], index\_col='date')

# Selecting a time series column

df['Time\_Index'] = np.arange(len(df)) # Create a numerical index for time

X = df[['Time\_Index']] # Features (Time)

y = df['p'] # Target variable

# Split data into training and testing sets

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

# Train a Linear Regression model

model = LinearRegression()

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

# Make predictions

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

# Evaluate the model

mae = mean\_absolute\_error(y\_test, y\_pred)

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

rmse = np.sqrt(mse)

print(f"MAE: {mae:.2f}")

print(f"RMSE: {rmse:.2f}")

# Plot actual vs predicted values

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

plt.plot(df.index[len(X\_train):], y\_test, label='Actual', color='blue')

plt.plot(df.index[len(X\_train):], y\_pred, label='Predicted', color='red', linestyle='dashed')

plt.xlabel('Date')

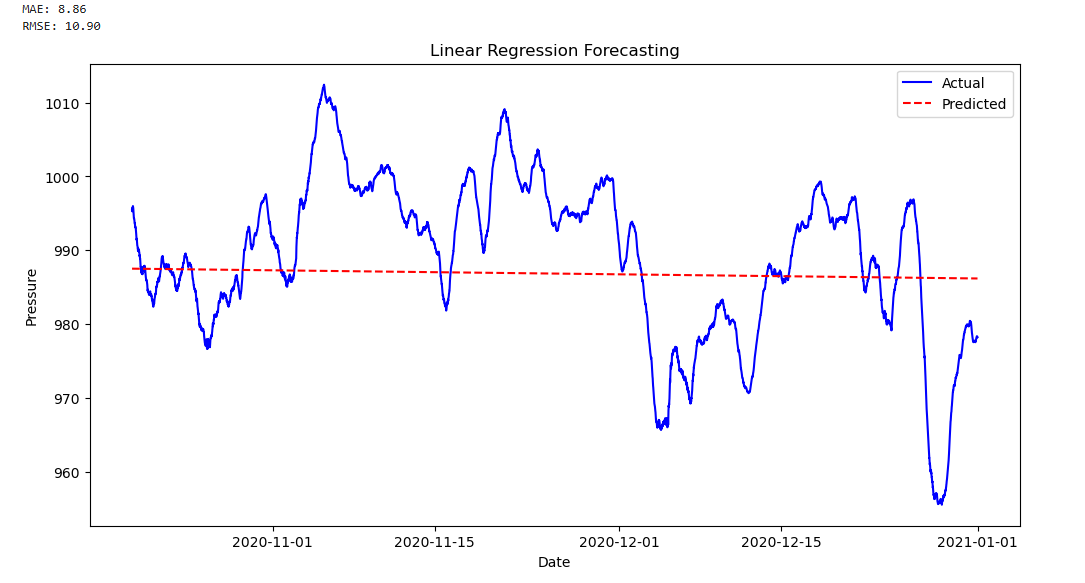
plt.ylabel('Pressure')

plt.title('Linear Regression Forecasting')

plt.legend()

plt.show()

**Output:**

****

**Result:**

Thus, the program to develop a linear regression model for forecasting time series data was done.