**EXP : 3**

Aim: Implement linear regression model in a time series data

Program:

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

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

from sklearn.linear\_model import LinearRegression

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

# Load the dataset

file\_path = "C:/Users/Lenovo/Downloads/DailyDelhiClimateTrain.csv"

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

# Convert date to datetime

data['date'] = pd.to\_datetime(data['date'])

# Select features and target variable

X = data[['humidity', 'wind\_speed', 'meanpressure']]

y = data['meantemp']

# Split into training and testing sets

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

# Train a linear regression model

model = LinearRegression()

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

# Make predictions

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

# Evaluate the model

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

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

print(f"Mean Squared Error: {mse:.2f}")

print(f"R-squared: {r2:.2f}")

# Visualize actual vs predicted values

plt.figure(figsize=(10, 6))

plt.scatter(y\_test, y\_pred, alpha=0.7, color='b')

plt.plot([y.min(), y.max()], [y.min(), y.max()], '--r', linewidth=2)

plt.xlabel("Actual Mean Temperature")

plt.ylabel("Predicted Mean Temperature")

plt.title("Actual vs Predicted Mean Temperature")

plt.grid()

plt.show()

Output:

