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

**AIM:**

To implement program for linear regression model for forecasting time series data using jupyter notebook.

**ALGORITHM:**

1. Load the taxi dataset then clean and load the values

2. Fit a linear model where time is the independent variable and the target (e.g.,total\_amount) is the dependent variable to forecast trends.

3. Plot the graph and visualize the values

**CODE:**

import pandas as pd

import matplotlib.pyplot as plt

from sklearn.linear\_model import LinearRegression

import numpy as np

# Load and prepare the data

df = pd.read\_csv("taxi dataset.csv")

df["datetime"] = pd.to\_datetime(df[["year", "month", "day"]]) + pd.to\_timedelta(df["hour\_of\_day"], unit='h')

df.set\_index("datetime", inplace=True)

# Resample total\_amount daily

daily\_avg = df["total\_amount"].resample("D").mean().dropna()

# Convert datetime index to ordinal for regression

X = daily\_avg.index.map(pd.Timestamp.toordinal).values.reshape(-1, 1)

y = daily\_avg.values

# Train linear regression model

model = LinearRegression()

model.fit(X, y)

# Predict using the model

y\_pred = model.predict(X)

# Plot actual vs predicted

plt.figure(figsize=(14, 6))

plt.plot(daily\_avg.index, y, label="Actual", alpha=0.6)

plt.plot(daily\_avg.index, y\_pred, label="Linear Trend", color="red", linewidth=2)

plt.title("Linear Regression for Time Series Forecasting")

plt.xlabel("Date")

plt.ylabel("Total Amount ($)")

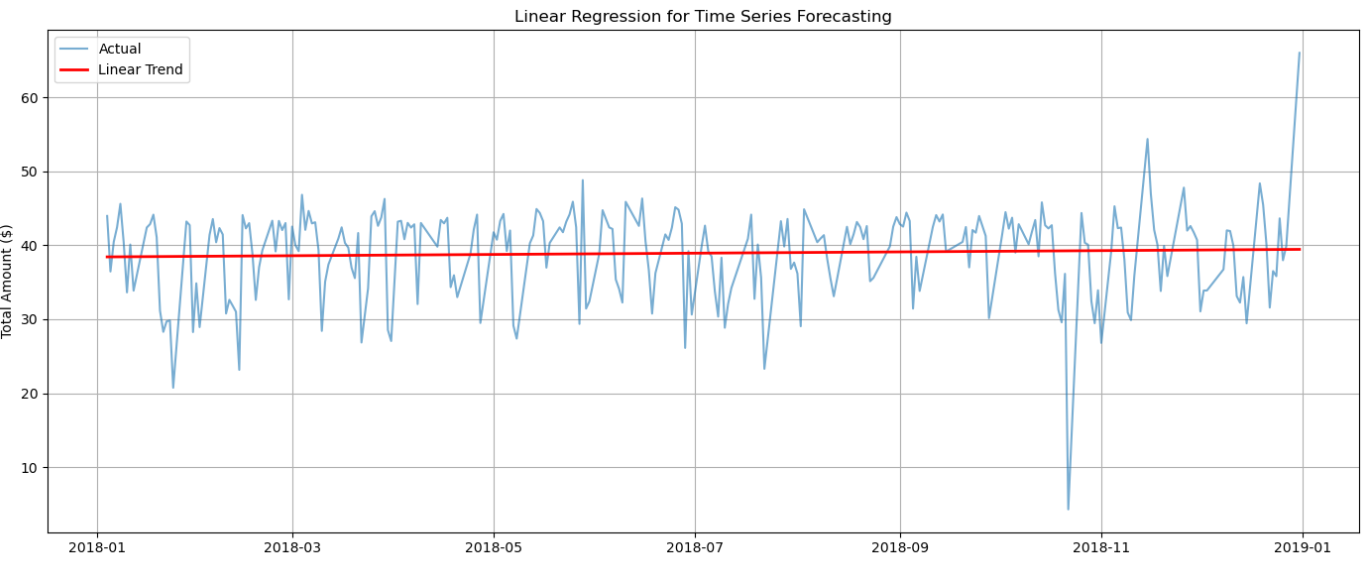
plt.legend()

plt.grid(True)

plt.tight\_layout()

plt.show()

**OUTPUT:**

****

**RESULT:**

The program to execute the time series data cleaning, loading and handling time series data and preprocessing techniques completed successfully and the output is verified.