**Ex : 03**

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

**AIM**

To write a program for linear regression model for forecasting time series data

**PROCEDURE**

 **Import Necessary Libraries**  
Load essential libraries such as Pandas, NumPy, Matplotlib, and Scikit-learn.

 **Load the Dataset**  
Read the dataset from a CSV file into a Pandas DataFrame.

 **Convert Date Column to Datetime Format**  
Ensure the date column is in the correct datetime format for further processing.

 **Extract Time Features**  
Extract Year and Month from the datetime column to use as features.

 **Aggregate Data by Month**  
Group the dataset by Year and Month to compute total car sales per month.

 **Prepare Features and Target Variable**  
Define the feature variables (Year, Month) and target variable (Car sales count).

 **Split the Data into Training and Testing Sets**  
Divide the dataset into training and testing sets without shuffling.

 **Train a Linear Regression Model**  
Fit a Linear Regression model to the training data.

 **Make Predictions**  
Use the trained model to predict car sales on the test set.

 **Evaluate the Model**  
Compute the Mean Squared Error (MSE) to assess prediction accuracy.

 **Visualize Results**  
Plot actual and predicted sales over time to analyze performance.

**PROGRAM**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from sklearn.linear\_model import LinearRegression

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

from sklearn.metrics import mean\_squared\_error

*# Load data (replace with the correct path)*

df = pd.read\_csv("Car Sales.xlsx - car\_data (2).csv")

*# Convert the date column to datetime (adjust 'Date' column name if needed)*

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

*# Extract Year, Month, and Day from the 'date' column*

df['Month'] = df['date'].dt.month

df['Year'] = df['date'].dt.year

*# Group data by Year and Month to get the total sales per month*

df\_sales = df.groupby(['Year', 'Month'])['Car\_id'].count().reset\_index()

df\_sales['Date'] = pd.to\_datetime(df\_sales[['Year', 'Month']].assign(DAY=1))

*# Define features (Year, Month) and target (Car sales count)*

X = df\_sales[['Year', 'Month']]

y = df\_sales['Car\_id']

*# 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 the model*

model = LinearRegression()

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

*# Make predictions on the test set*

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

*# Evaluate the model*

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

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

*# Plot actual vs predicted sales*

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

plt.plot(df\_sales['Date'], df\_sales['Car\_id'], label='Actual Sales', color='blue')

plt.plot(pd.to\_datetime(X\_test[['Year', 'Month']].assign(DAY=1)), y\_pred, label='Predicted Sales', color='red')

plt.xlabel('Date')

plt.ylabel('Number of Car Sales')

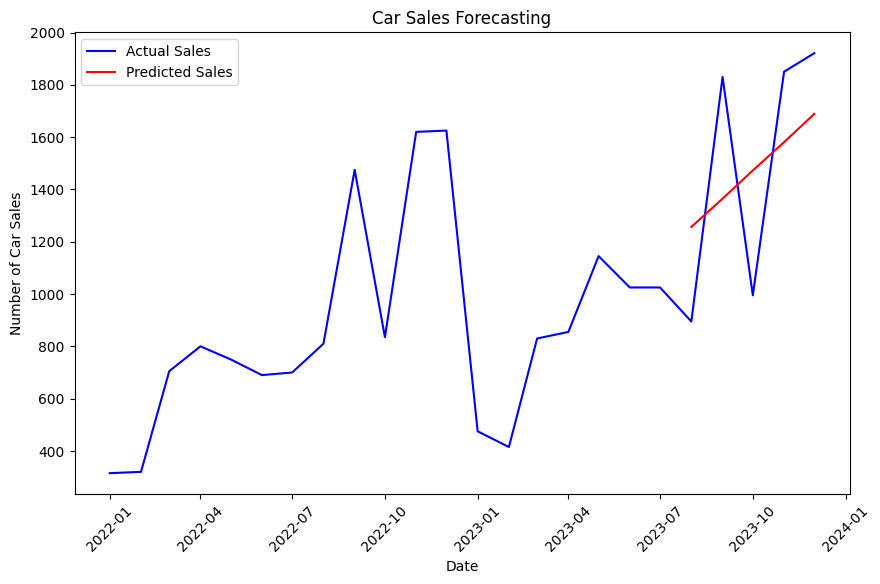
plt.title('Car Sales Forecasting')

plt.xticks(rotation=45)

plt.legend()

plt.show()

Mean Squared Error: 140384.90685312016

****

**RESULT**

Thus the program has been implemented successfully.