| **EXP:3**  **06/02/2025** | **LINEAR REGRESSION** |
| --- | --- |

**AIM:**

To implement a program for time series data cleaning,loading and handling time series data & preprocessing techniques and perform linear regression

**PROCEDURE:**

**1)Import the libraries**

import pandas as pd

` import numpy as np

import seaborn as sns

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, r2\_score

**2)Load & Clear the dataset**

**df = pd.read\_csv("/content/sample\_data/Car\_sales.csv")**

**df.dropna(inplace=True)**

**3)Prepare features and target**

X = pd.get\_dummies(df[['Manufacturer']], drop\_first=True)

y = df['Sales\_in\_thousands']

**4)Split the dataset**

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

**5)Train the model**

model = LinearRegression()

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

**6)Make Predictions**

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

**7)Evaluate the model**

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

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

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

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

**OUTPUT:**



**8)Plot scatter plot**

plt.figure(figsize=(25, 6))

sns.scatterplot(data=df, x='Manufacturer', y='Sales\_in\_thousands')

plt.xlabel('Manufacturer')

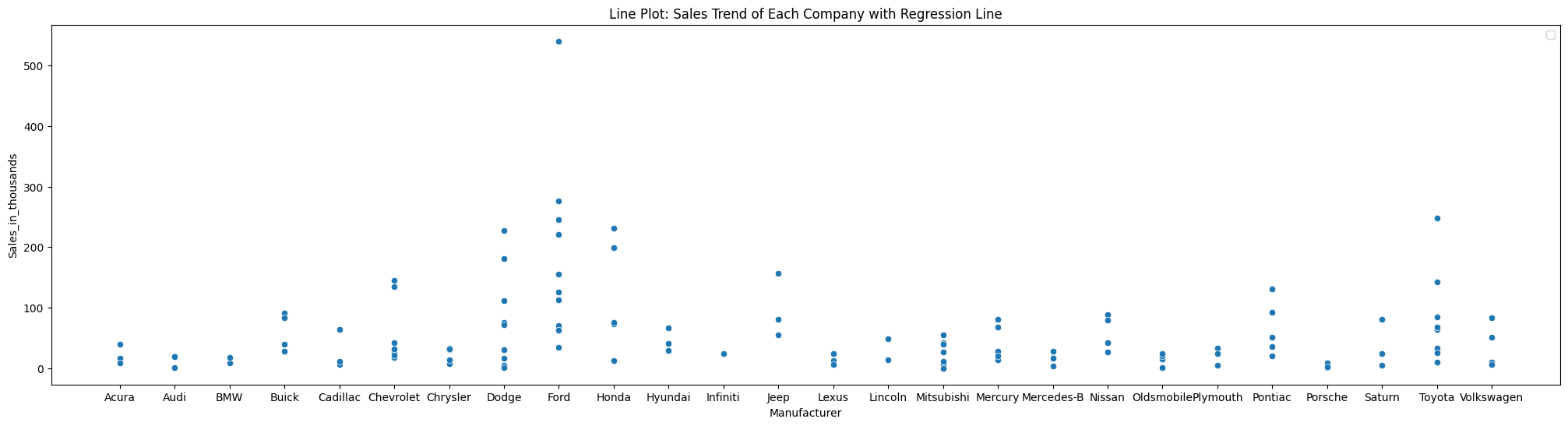
plt.ylabel('Sales\_in\_thousands')

plt.title('Line Plot: Sales Trend of Each Company with Regression Line')

plt.legend()

plt.show()

**OUTPUT:**



**9)Plot regression line**

plt.figure(figsize=(25, 6))

sns.lineplot(data=df, x='Manufacturer',y='\_\_year\_resale\_value', color='red', label='Regression Line')

plt.xlabel('Manufacturer')

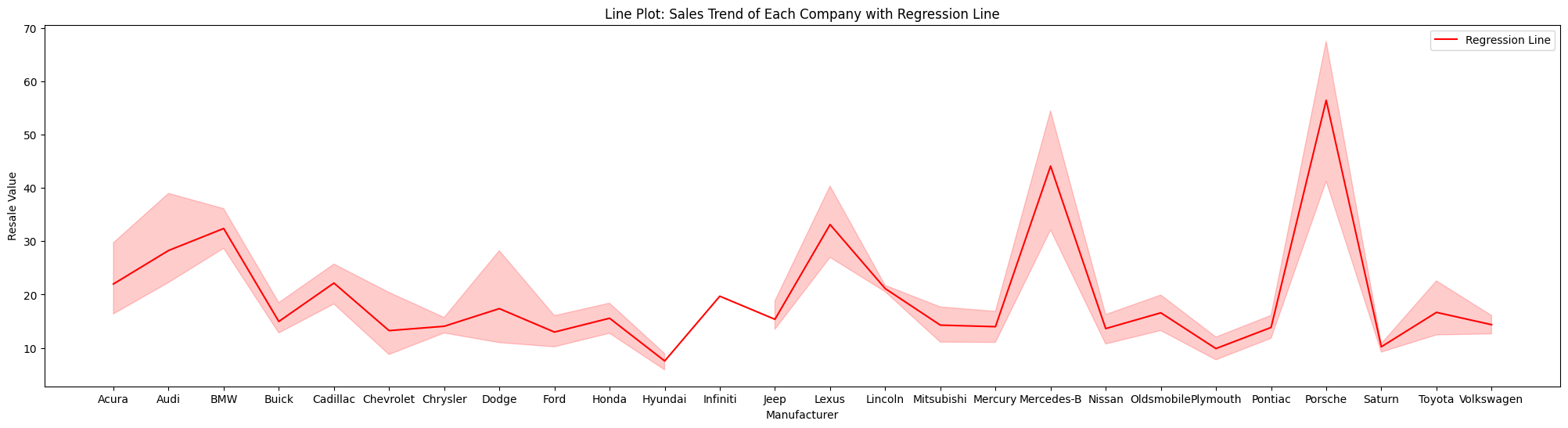
plt.ylabel('Resale Value')

plt.title('Line Plot: Sales Trend of Each Company with Regression Line')

plt.legend()

plt.show()

**OUTPUT:**

****

**RESULT:**

Thus the program has been executed successfully