**EXPERIMENT NO 14**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

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

from sklearn.linear\_model import LinearRegression

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

# Create sample dataset

data = {

    "Size (sq ft)": [850, 900, 1200, 1500, 1800, 2000, 2200, 2500, 2700, 3000],

    "Price (in lakhs)": [35, 40, 50, 65, 75, 85, 95, 110, 125, 140]

}

df = pd.DataFrame(data)

print(df.head())  # Display first few rows

# Summary statistics

print(df.describe())

# Visualize the data

plt.figure(figsize=(8, 5))

sns.scatterplot(x=df["Size (sq ft)"], y=df["Price (in lakhs)"])

plt.xlabel("Size (sq ft)")

plt.ylabel("Price (in lakhs)")

plt.title("House Size vs. Price")

plt.show()

# Split data into independent variable (X) and dependent variable (y)

X = df[["Size (sq ft)"]]  # Features

y = df["Price (in lakhs)"]  # Target

# Split data into training and testing sets (80% training, 20% testing)

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

# Initialize and train Linear Regression model

model = LinearRegression()

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

# Make predictions

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

# Predict price for a new house size

new\_house\_size = np.array([[700]])  # Example input

predicted\_price = model.predict(new\_house\_size)

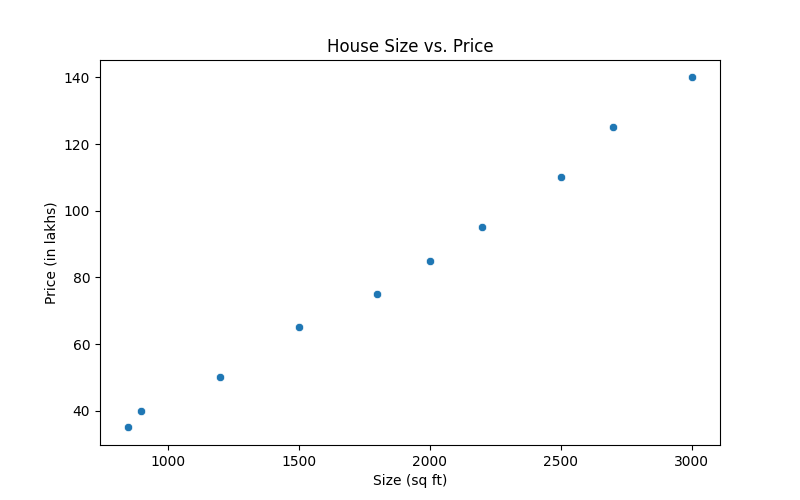
print("Predicted Price for 700 sq ft: "+str(predicted\_price[0])+ "lakhs")

new\_house\_size = np.array([[3000]])  # Example input

predicted\_price = model.predict(new\_house\_size)

# Print formatted output

print(f"Predicted Price for 3000 sq ft: {predicted\_price[0]:.4f} lakhs")

**OUTPUT:**

Size (sq ft) Price (in lakhs)

0 850 35

1 900 40

2 1200 50

3 1500 65

4 1800 75

Size (sq ft) Price (in lakhs)

count 10.000000 10.000000

mean 1865.000000 82.000000

std 749.833315 35.761556

min 850.000000 35.000000

25% 1275.000000 53.750000

50% 1900.000000 80.000000

75% 2425.000000 106.250000

max 3000.000000 140.000000