regression

November 16, 2025

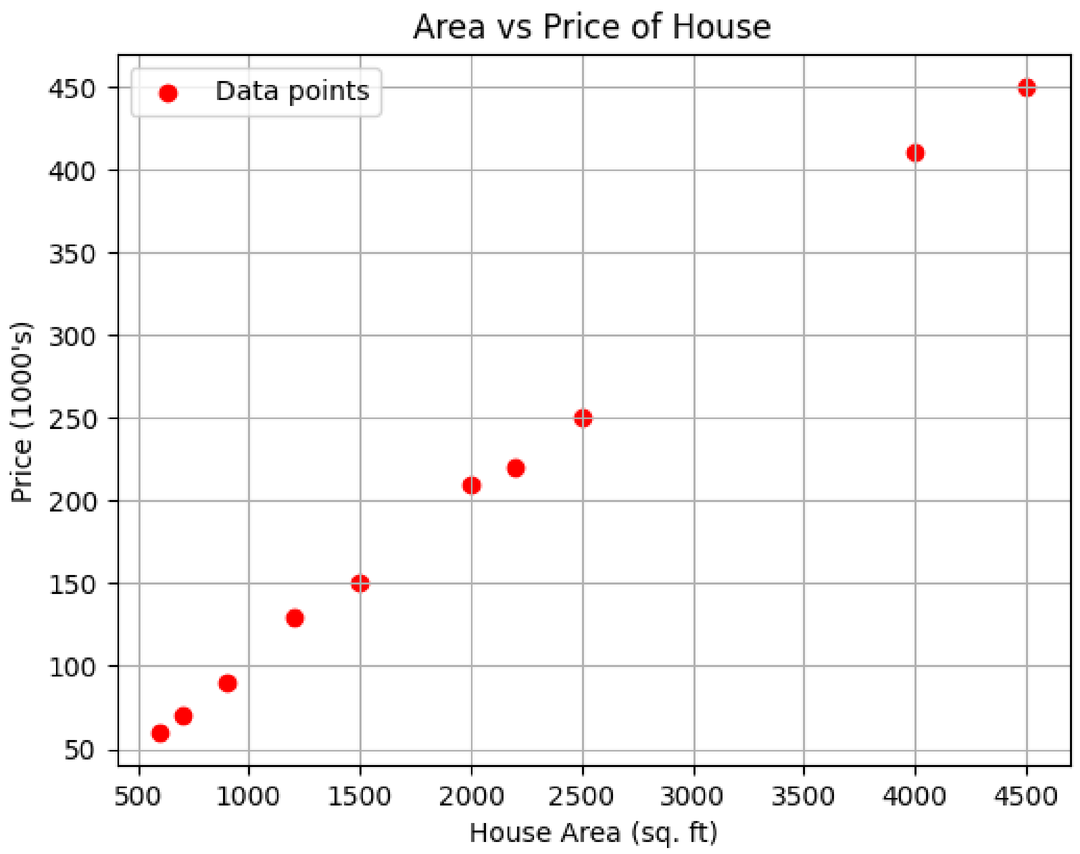
[1]:
```
# The simple linear regression
```

[2]:
```
import numpy as np
import matplotlib.pyplot as plt
```

[3]:
```
area = np.array([600, 700, 900, 1200, 1500, 2000, 2200, 2500, 4000, 4500])
price = np.array([60, 70, 90, 130, 150, 210, 220, 250, 410, 450])
```

[4]:
```
# Scatter plot of dataset

plt.scatter(x=area, y=price, color="red", label="Data points")
plt.xlabel("House Area (sq. ft)")
plt.ylabel("Price (1000's)")
plt.title("Area vs Price of House")
plt.grid(True)
plt.legend()
```

[4]:
```
<matplotlib.legend.Legend at 0x7dc7a2470c20>
```



[5]:
```
x_bar = np.mean(area)
y_bar = np.mean(price)

numerator = np.sum((area - x_bar) * (price - y_bar))
denominator = np.sum((area - x_bar) ** 2)
```

[6]:
```
beta_1 = numerator / denominator
beta_0 = y_bar - beta_1 * x_bar
```

[7]:
```
print(f"Intercept (beta_0): {beta_0:.2f}")
print(f"Slope (beta_1): {beta_1:.2f}")
```

```
Intercept (beta_0): 1.56
Slope (beta_1): 0.10
```

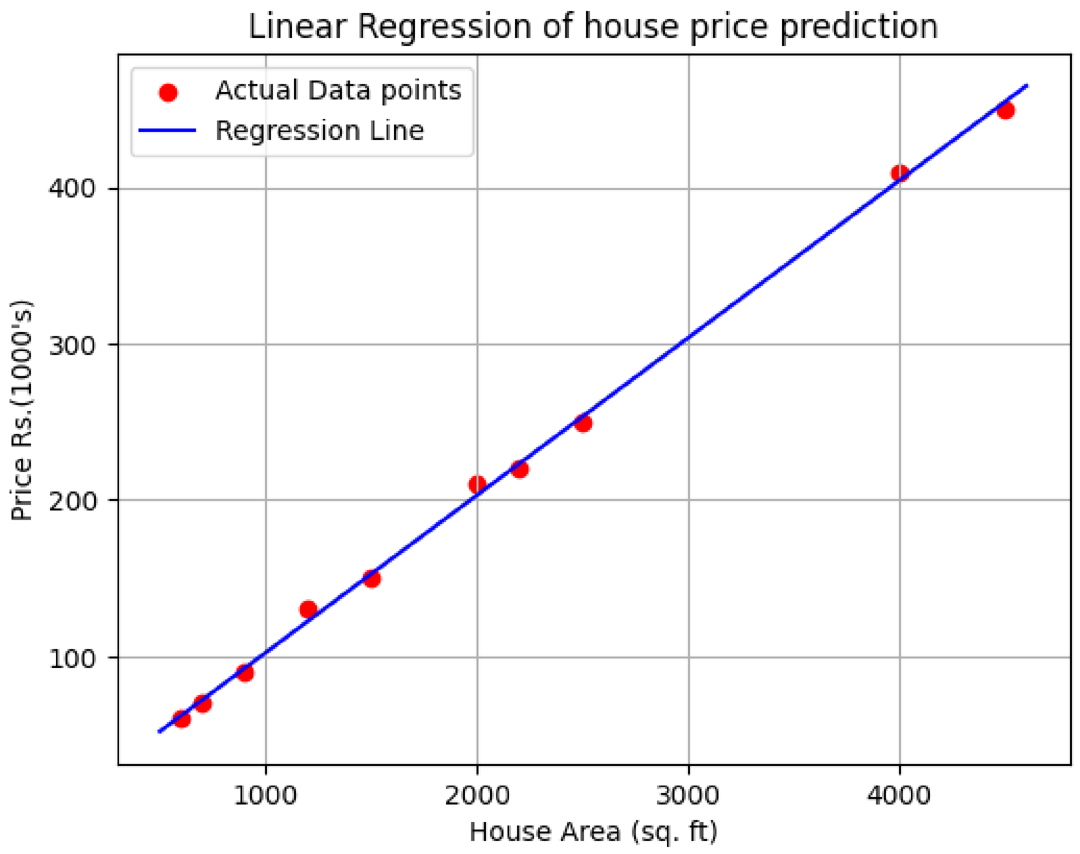
[8]:
```
def predict(area_val):
    return beta_0 + beta_1 * area_val
```

[9]:
```
area_input = np.linspace(min(area) - 100, max(area) + 100, 100)
predicted_output = predict(area_input)
```

```
# print(area_input)
# print(predicted_output)
```

[10]:
```
plt.scatter(x=area, y=price, color="red", label="Actual Data points")
plt.plot(area_input, predicted_output, color="blue", label="Regression Line")
plt.xlabel("House Area (sq. ft)")
plt.ylabel("Price Rs.(1000's)")
plt.title("Linear Regression of house price prediction")
plt.grid(True)
plt.legend()
```

[10]:
```
<matplotlib.legend.Legend at 0x7dc7a0191450>
```



[11]:
```
test_area = 5500
pred_price = predict(test_area)
print(f"Price for area {test_area} sq. ft is Rs. {pred_price:.2f}K")
```

```
Price for area 5500 sq. ft is Rs. 555.51K
```

```
[12]: import pandas as pd
```

```
[13]: df = pd.DataFrame({"Area":area, "Price":price})
```

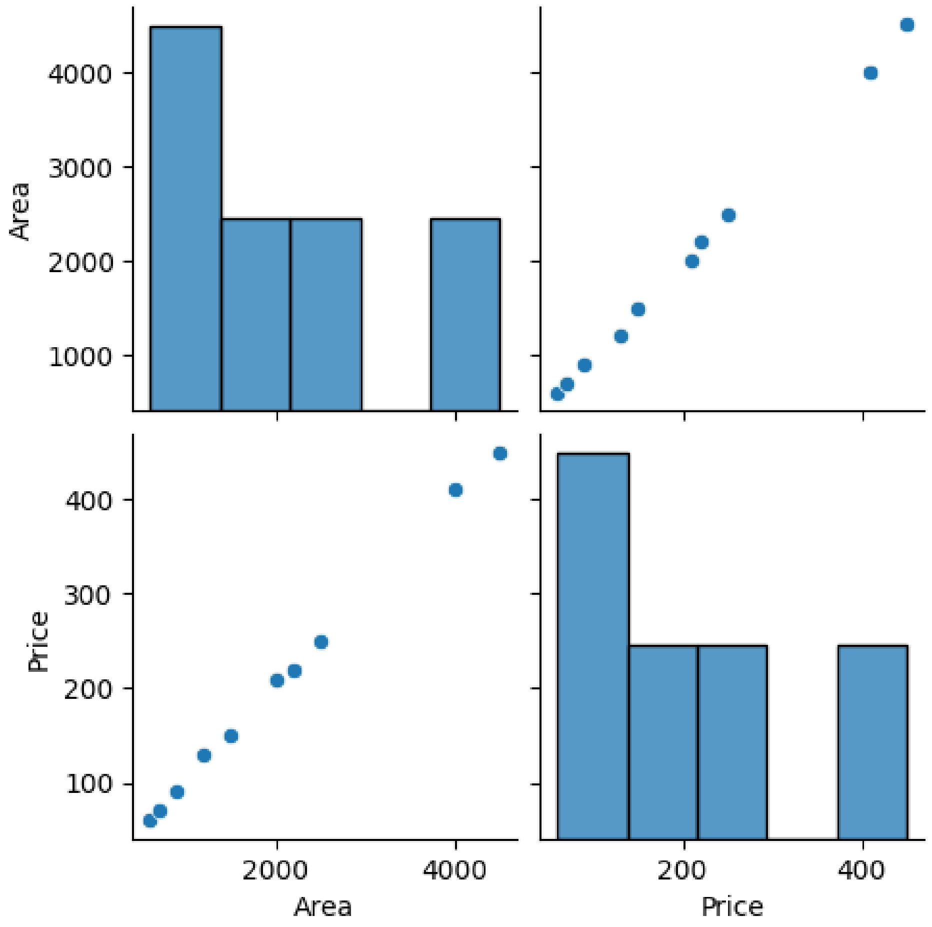
```
[14]: df
```

```
[14]:    Area  Price
      0   600     60
      1   700     70
      2   900     90
      3  1200    130
      4  1500    150
      5  2000    210
      6  2200    220
      7  2500    250
      8  4000    410
      9  4500    450
```

```
[15]: import seaborn as sns
```

```
[16]: sns.pairplot(df)
```

```
[16]: <seaborn.axisgrid.PairGrid at 0x7dc784f02900>
```



```
[17]: # ! pip install scikit-learn
```

```
[18]: X = df.drop("Price", axis=1)
      y = df.Price
```

```
[19]: from sklearn.model_selection import train_test_split

      X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=42,␣
       ↪test_size=0.3)
```

```
[20]: X_train.shape
```

```
[20]: (7, 1)
```

```
[21]: # import model

      from sklearn.linear_model import LinearRegression
```

[22]:
```
lr = LinearRegression()
lr.fit(X_train, y_train)
```

[22]:
```
LinearRegression()
```

[23]:
```
lr.intercept_
```

[23]:
```
np.float64(2.7248104008668292)
```

[24]:
```
lr.coef_
```

[24]:
```
array([0.09932286])
```

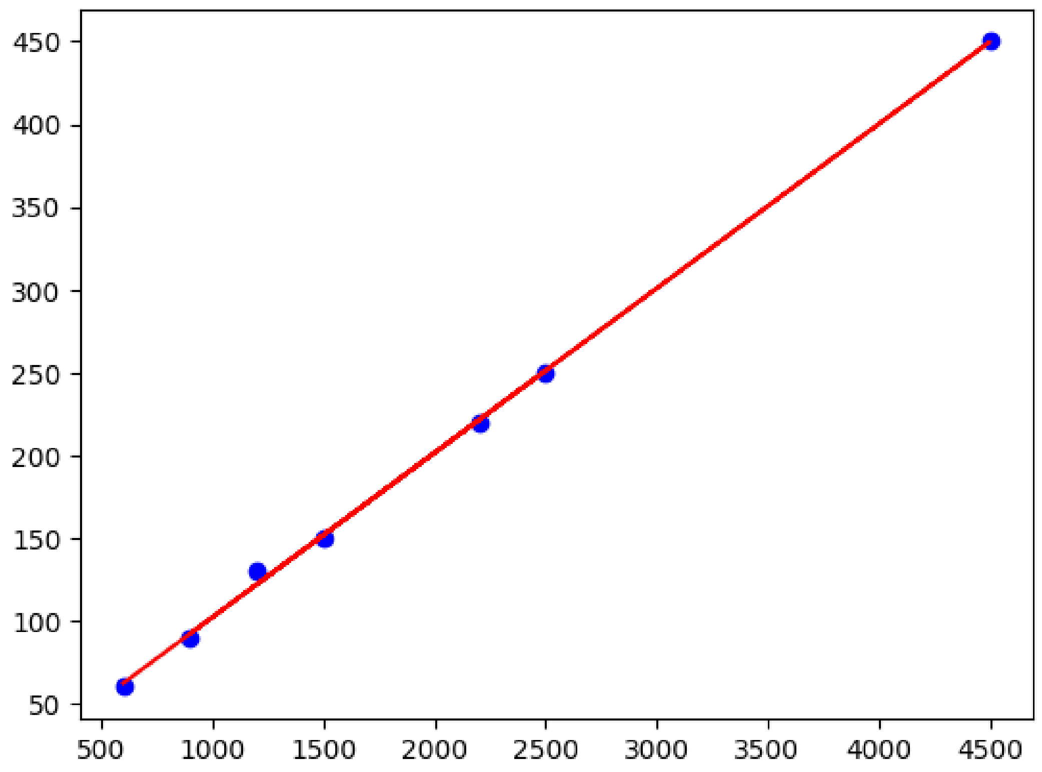
[25]:
```
y_pred = lr.predict(X_test)
```

[26]:
```
print(y_test)
print(y_pred)
```

```
8    410
1     70
5    210
Name: Price, dtype: int64
[400.01625135  72.25081257 201.37053088]
```

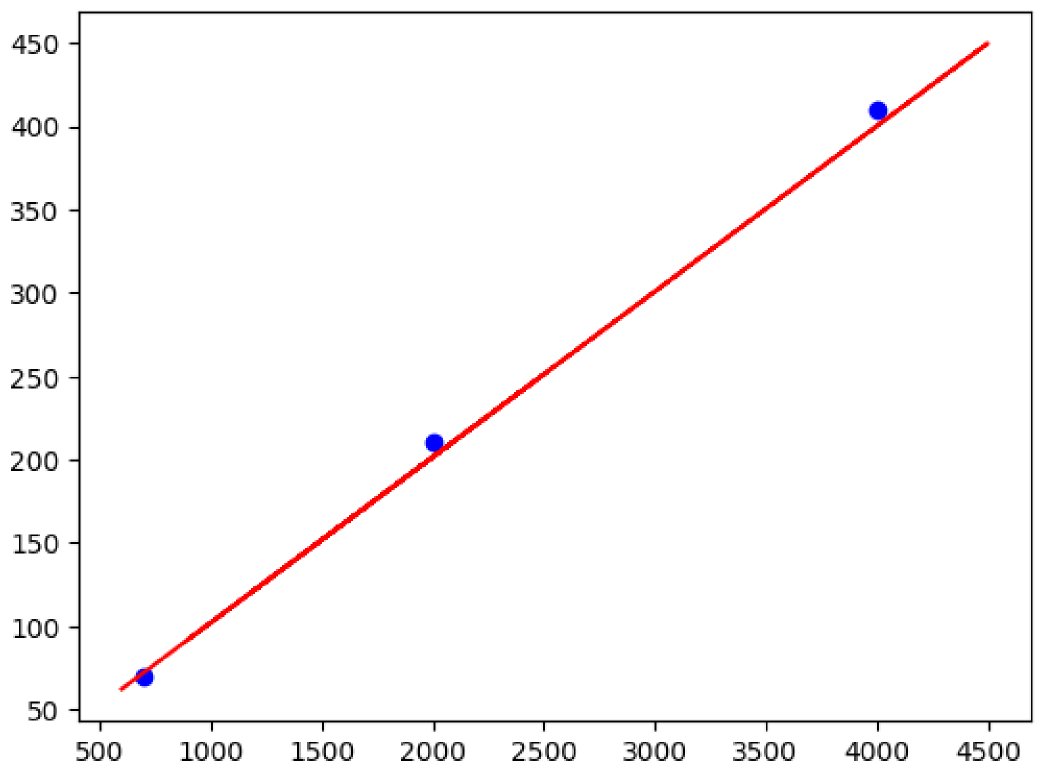
[27]:
```
plt.scatter(X_train, y_train, color="blue")
plt.plot(X_train, lr.predict(X_train), color="red")
```

[27]:
```
[<matplotlib.lines.Line2D at 0x7dc78396e5d0>]
```



[28]:
```
plt.scatter(X_test, y_test, color="blue")
plt.plot(X_train, lr.predict(X_train), color="red")
```

[28]:
```
[<matplotlib.lines.Line2D at 0x7dc78380b250>]
```

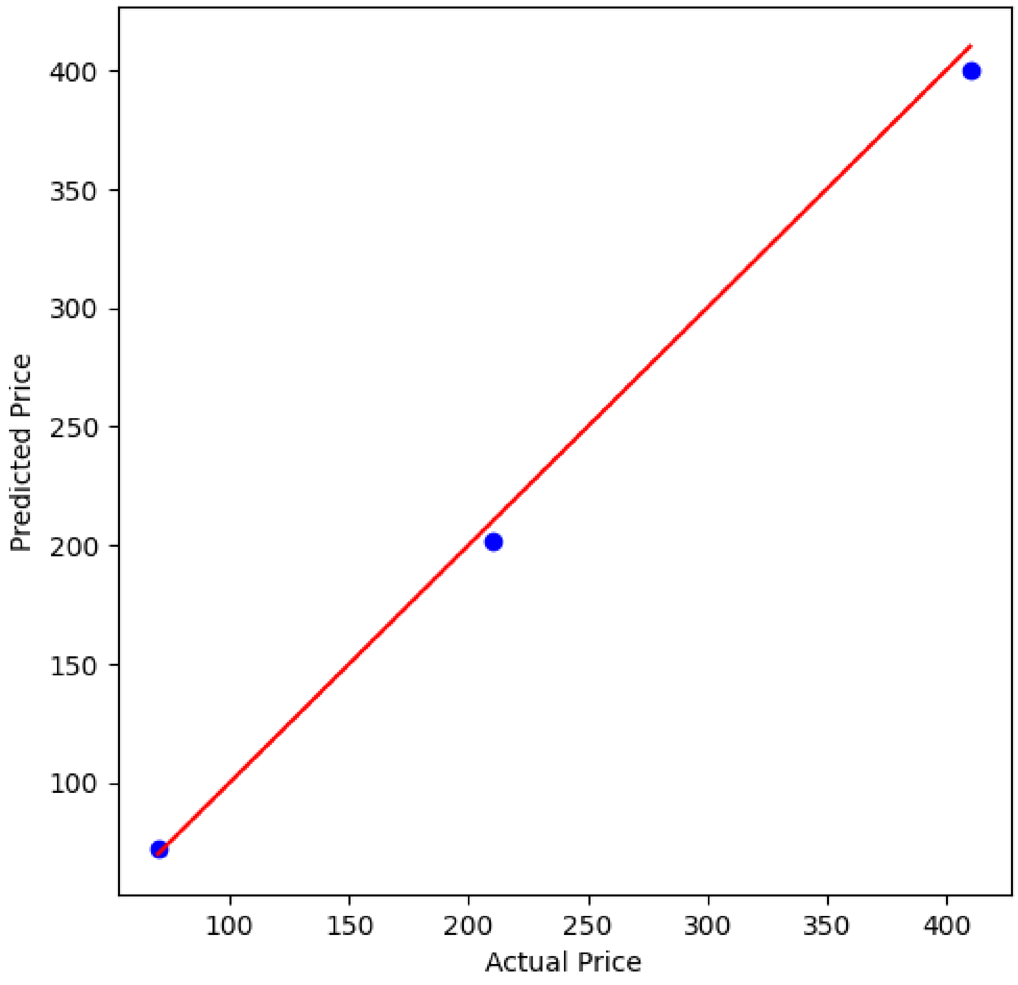


[30]:
```
from sklearn.metrics import r2_score, mean_squared_error

print(r2_score(y_test, y_pred))
print(mean_squared_error(y_test, y_pred))
```

```
0.9969313504868009
59.73637719027488
```

[31]:
```
import matplotlib.pyplot as plt
plt.figure(figsize=(6, 6))
plt.scatter(x=y_test, y=y_pred, color="blue")
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()],␣
 ↪color="red")

plt.xlabel("Actual Price")
plt.ylabel("Predicted Price")
```

[31]:
```
Text(0, 0.5, 'Predicted Price')
```


Predicted Price
Actual Price
400
350
300
250
200
150
100
100
150
200
250
300
350
400