

# Lesson-5 Linear Regression

June 24, 2024

```
[8]: # import the dataframe
import pandas as pd
# dataset
data = pd.read_csv('data.csv')
```

```
[9]: data.head()
```

```
[9]:   total_sqft  price
0      1056.0   39.07
1      2600.0  120.00
2      1440.0   62.00
3      1521.0   95.00
4      1200.0   51.00
```

```
[10]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12993 entries, 0 to 12992
Data columns (total 2 columns):
#   Column          Non-Null Count  Dtype
---  -
0   total_sqft      12993 non-null  float64
1   price           12993 non-null  float64
dtypes: float64(2)
memory usage: 203.1 KB
```

```
[11]: # input and output values
x = data.drop('price', axis=1)
y = data.price
```

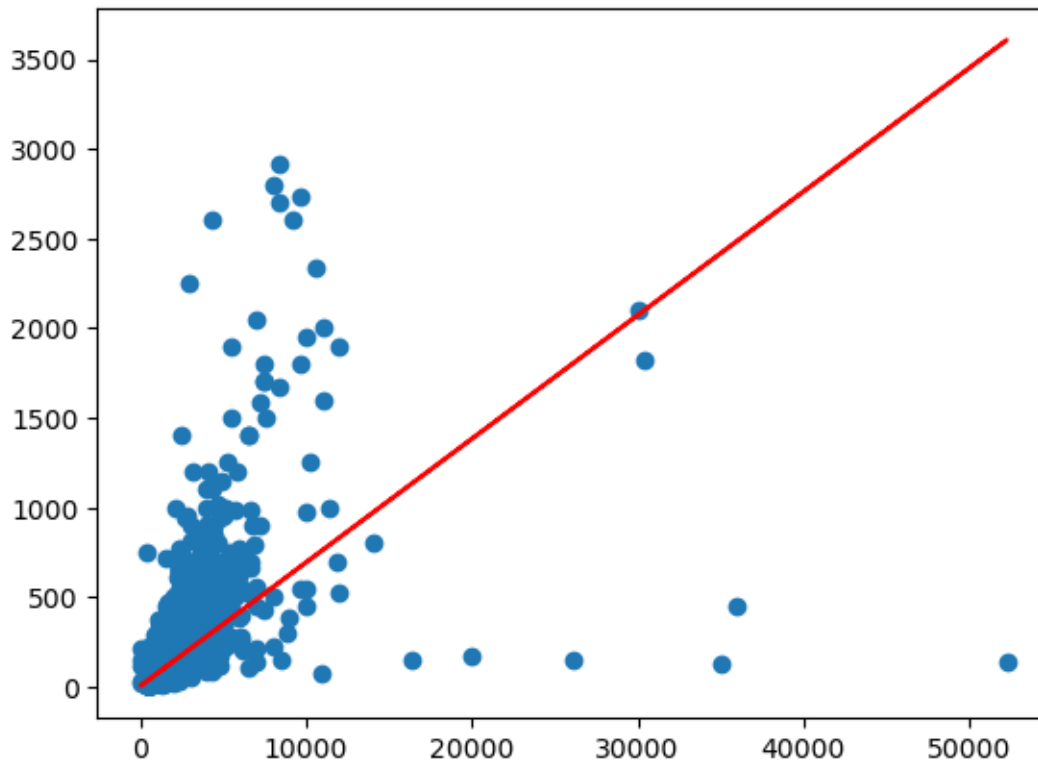
```
[12]: from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.25)
```

```
[13]: from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(X_train, y_train)
```

```
[13]: LinearRegression()
```

```
[14]: # visualize the model
```

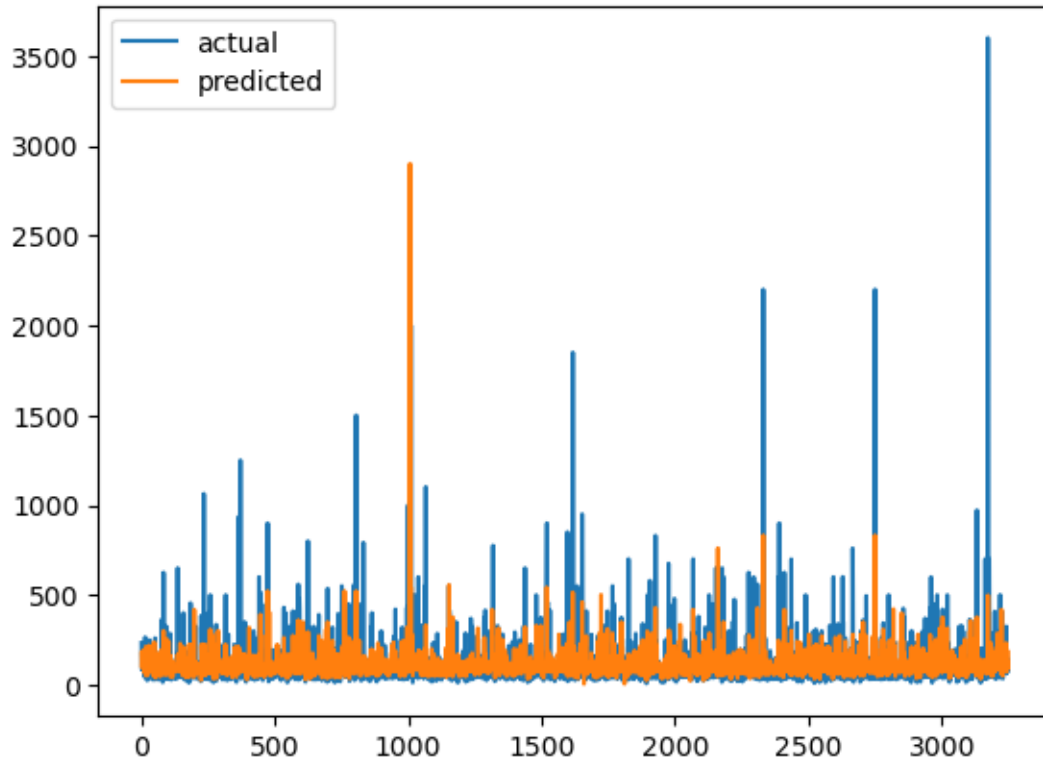
```
[19]: import matplotlib.pyplot as plt
plt.scatter(X_train['total_sqft'], y_train)
plt.plot(X_train['total_sqft'], model.predict(X_train), c='r')
plt.show()
```



```
[25]: from sklearn.metrics import r2_score
r2_score(y_test, model_pred)
```

```
[25]: 0.29751332137559994
```

```
[23]: # make prediction
model_pred = model.predict(X_test)
# show prediction and actual values on the chart
plt.plot([i for i in range(len(y_test))], y_test, label='actual')
plt.plot([i for i in range(len(y_test))], model_pred, label='predicted')
plt.legend()
plt.show()
```



[ ]: