2_Regresion Lineal Multiple

July 6, 2025

Creado por:

Isabel Maniega

1 Regresión Lineal Múltiple

[1]:
```
# pip install scikit-learn
```

[2]:
```
import numpy as np
from sklearn import datasets, linear_model
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.metrics import mean_squared_error, r2_score
```

[3]:
```
dataset = datasets.load_diabetes()
print(dataset.DESCR)
# Crear un DataFrame con los datos
data = pd.DataFrame(dataset.data, columns=dataset.feature_names)
data['level'] = dataset.target
```

```
.. _diabetes_dataset:

Diabetes dataset
----------------

Ten baseline variables, age, sex, body mass index, average blood
pressure, and six blood serum measurements were obtained for each of n =
442 diabetes patients, as well as the response of interest, a
quantitative measure of disease progression one year after baseline.

**Data Set Characteristics:**

:Number of Instances: 442

:Number of Attributes: First 10 columns are numeric predictive values

:Target: Column 11 is a quantitative measure of disease progression one year
after baseline
```

```
:Attribute Information:
    - age     age in years
    - sex
    - bmi     body mass index
    - bp      average blood pressure
    - s1      tc, total serum cholesterol
    - s2      ldl, low-density lipoproteins
    - s3      hdl, high-density lipoproteins
    - s4      tch, total cholesterol / HDL
    - s5      ltg, possibly log of serum triglycerides level
    - s6      glu, blood sugar level

Note: Each of these 10 feature variables have been mean centered and scaled by
the standard deviation times the square root of `n_samples` (i.e. the sum of
squares of each column totals 1).

Source URL:
https://www4.stat.ncsu.edu/~boos/var.select/diabetes.html

For more information see:
Bradley Efron, Trevor Hastie, Iain Johnstone and Robert Tibshirani (2004) "Least
Angle Regression," Annals of Statistics (with discussion), 407-499.
(https://web.stanford.edu/~hastie/Papers/LARS/LeastAngle_2002.pdf)
```

[4]:

```
data
```

[4]:

```
          age       sex       bmi        bp        s1        s2        s3  \
0    0.038076  0.050680  0.061696  0.021872 -0.044223 -0.034821 -0.043401
1   -0.001882 -0.044642 -0.051474 -0.026328 -0.008449 -0.019163  0.074412
2    0.085299  0.050680  0.044451 -0.005670 -0.045599 -0.034194 -0.032356
3   -0.089063 -0.044642 -0.011595 -0.036656  0.012191  0.024991 -0.036038
4    0.005383 -0.044642 -0.036385  0.021872  0.003935  0.015596  0.008142
..        …         …         …         …         …         …         …
437  0.041708  0.050680  0.019662  0.059744 -0.005697 -0.002566 -0.028674
438 -0.005515  0.050680 -0.015906 -0.067642  0.049341  0.079165 -0.028674
439  0.041708  0.050680 -0.015906  0.017293 -0.037344 -0.013840 -0.024993
440 -0.045472 -0.044642  0.039062  0.001215  0.016318  0.015283 -0.028674
441 -0.045472 -0.044642 -0.073030 -0.081413  0.083740  0.027809  0.173816

           s4        s5        s6  level
0   -0.002592  0.019907 -0.017646  151.0
1   -0.039493 -0.068332 -0.092204   75.0
2   -0.002592  0.002861 -0.025930  141.0
3    0.034309  0.022688 -0.009362  206.0
4   -0.002592 -0.031988 -0.046641  135.0
```

```
..         ...       ...       ...    ...
437 -0.002592  0.031193  0.007207  178.0
438  0.034309 -0.018114  0.044485  104.0
439 -0.011080 -0.046883  0.015491  132.0
440  0.026560  0.044529 -0.025930  220.0
441 -0.039493 -0.004222  0.003064   57.0

[442 rows x 11 columns]
```

[5]:
```
# Load the diabetes dataset
diabetes_X, diabetes_y = datasets.load_diabetes(return_X_y=True)
```

[6]:
```
from sklearn.model_selection import train_test_split
# Separo los datos de "train" entrenamiento y "test" prueba para probar los␣
↪algoritmos

X_train, X_test, y_train, y_test = train_test_split(diabetes_X, diabetes_y,␣
↪test_size=0.2)
```

[7]:
```
lr_multiple = linear_model.LinearRegression()
```

[8]:
```
lr_multiple.fit(X_train, y_train)
```

[8]:
```
LinearRegression()
```

[9]:
```
y_pred = lr_multiple.predict(X_test)
y_pred
```

[9]:
```
array([198.92942131, 114.84534815, 241.55686724, 199.64125541,
       112.61590082, 107.1934254 , 107.96941096, 137.59993048,
        69.27761389, 222.33109311, 150.86244978, 145.40001303,
       255.34710296, 163.35076062, 142.81825513, 126.70611647,
       238.31372899, 186.66772647, 153.05948515, 244.01602178,
        65.51277224, 209.49948887,  96.87859477, 159.76543679,
       112.6826685 , 209.19870258, 105.15518047, 117.03331812,
        97.82531017,  78.06915422, 185.62907937, 218.51218553,
       157.96980347, 179.05651855, 137.4350862 , 117.71019316,
       255.82640598,  66.88811217, 195.9950507 ,  86.4545535 ,
       254.58156976, 166.56049251, 253.55564875,  83.4338512 ,
       163.32869193,  95.08668772, 228.58659933, 237.04679853,
       105.48363302,  60.39818583,  83.11228058, 169.84090478,
       253.0277734 , 126.99207723, 132.12846921, 176.93674279,
       259.91047517, 120.86320122, 138.05650514, 104.2942936 ,
       190.68577909, 197.46431645, 161.17951365,  76.62152224,
       148.65018355, 143.67764528, 151.82205271,  79.41488423,
       164.88841741, 168.62825415, 166.3294742 , 190.99375231,
       242.50128692,  91.6317138 , 200.91842072, 157.80520745,
```

```
 92.35210501, 233.11088782, 120.85966053, 104.91747567,
176.31137634, 210.97548991, 164.59101327,  90.39905682,
177.96704449, 144.60286349, 123.53437884,  99.76696947,
221.45945529])
```

[10]:
```python
y_test
```

[10]:
```
array([220.,  88., 259., 265., 199., 118.,  63.,  40., 134., 259., 246.,
        97., 273.,  58., 103.,  53., 270., 164.,  86., 252., 158., 310.,
        87., 113., 102., 297.,  65.,  53.,  49.,  59., 217., 173., 155.,
       143., 170., 214., 310.,  52., 202.,  60., 233., 141., 281., 181.,
       206.,  69.,  99., 272.,  88.,  57.,  71., 258., 243.,  42.,  49.,
        91., 242.,  78., 115., 142., 170.,  48., 185.,  75., 150., 185.,
        85.,  42., 178., 242., 131., 232., 132.,  88., 186., 276., 101.,
       236., 182., 135., 126., 121., 120.,  64., 311., 202.,  71., 118.,
       180.])
```

[11]:
```python
print('DATOS DEL MODELO REGRESIÓN LINEAL MULTIPLE')
print()
print('Valor de las pendientes o coeficientes "a":')
print(lr_multiple.coef_)
print('Valor de la intersección o coeficiente "b":')
print(lr_multiple.intercept_)
```

```
DATOS DEL MODELO REGRESIÓN LINEAL MULTIPLE

Valor de las pendientes o coeficientes "a":
[   6.89443248 -226.02422036  529.1542349   344.9279648  -886.09501482
  530.84496929  209.30837657  238.80440808  765.83681449   74.64827602]
Valor de la intersección o coeficiente "b":
152.84136793062558
```

[12]:
```python
print("Precisión del modelo:")
print(lr_multiple.score(X_train, y_train))
```

```
Precisión del modelo:
0.5314676813743083
```

Creado por:

Isabel Maniega