Разширена линейна регресия с библиотека за статистически модели

В темата ще бъдат разгледани следните основни въпроси:

- Импортиране на пакети
- Предоставяне на данни и трансформиране на входове
- Създаване на модел и напасване
- Прогнозиране на отговора

Проект BG05M2OP001-2.016-0003„Модернизация на Национален военен университет "В. Левски" - гр. Велико Търново и Софийски университет "Св. Климент Охридски" - гр. София, в професионално направление 5.3 Компютърна и комуникационна техника“, финансиран от Оперативна програма „Наука и образование за интелигентен растеж“, съфинансирана от Европейския съюз чрез Европейските структурни и инвестиционни фондове.

Разширена линейна регресия с библиотека за статистически модели

Линейна регресия може да се приложи в Python и като се използва пакета statsmodels. Обикновено това е желателно, когато има нужда от по-подробни резултати. Процедурата е подобна на тази, използващ пакета scikit-learn.

Стъпка 1: Импортиране на пакети

Първо трябва да се импортират някои пакети. В допълнение към numpy, трябва да се импортира и пакета statsmodels.api (фиг. 8.1).

```
import numpy as np
import statsmodels.api as sm
```

Фиг. 8.1. Импортиране на пакети

Стъпка 2: Предоставяне на данни и трансформиране на входове

Можете да предоставите входовете и изходите по същия начин, както когато сте използвали scikit-learn (фиг. 8.2).

```
x = [ [0, 1], [5, 1], [15, 2], [25, 5], [35, 11], [45, 15],
[55, 34], [60, 35] ]
y = [4, 5, 20, 14, 32, 22, 38, 43]
x, y = np.array(x), np.array(y)
```

Фиг. 8.2. Предоставяне на входните и изходни данни

Проект BG05M2OP001-2.016-0003„Модернизация на Национален военен университет "В. Левски" - гр. Велико Търново и Софийски университет "Св. Климент Охридски" - гр. София, в професионално направление 5.3 Компютърна и комуникационна техника“, финансиран от Оперативна програма „Наука и образование за интелигентен растеж“, съфинансирана от Европейския съюз чрез Европейските структурни и инвестиционни фондове.

Входните и изходните масиви са създадени, но работата все още не е завършена. Трябва да се добави колоната с единици към входовете, ако е необходимо да се изчисли коефициентът b_0 чрез statsmodels. По подразбиране той не взема предвид. Това се осъществява само с едно извикване на метода .add_constant (фиг. 8.3).

```
x = sm.add_constant(x)
```

Фиг. 8.3. Добавяне на колона от единици към входния масив

Методът add_constant() взема входния масив x като аргумент и връща нов масив с колона от единици, вмъкнати в началото. Двата масива x и y могат да се отпечатат по следния начин (фиг. 8.4).

```
print(f"x:\n {x}")
print(f"y:\n {y}")
```

а)

```
x:
 [[ 1.  0.  1.]
 [ 1.  5.  1.]
 [ 1. 15.  2.]
 [ 1. 25.  5.]
 [ 1. 35. 11.]
 [ 1. 45. 15.]
 [ 1. 55. 34.]
 [ 1. 60. 35.]]
y:
 [ 4  5 20 14 32 22 38 43]
```

б)

Фиг. 8.4. Изглед на входния масив x и изходен y

а. Код на Python **б.** Изход от програмата

Проект BG05M2OP001-2.016-0003„Модернизация на Национален военен университет "В. Левски" - гр. Велико Търново и Софийски университет "Св. Климент Охридски" - гр. София, в професионално направление 5.3 Компютърна и комуникационна техника“, финансиран от Оперативна програма „Наука и образование за интелигентен растеж“, съфинансирана от Европейския съюз чрез Европейските структурни и инвестиционни фондове.

Модифицираният входен масив x има три колони: първата колона от единици, съответстваща на b_0 и заместваща пресичането (intercept), както и две колони с оригиналните данни.

Стъпка 3: Създаване на модел и напасване

Регресионният модел, базиран на пресмятането на обикновени най-малките квадрати OLS (Ordinary Least Squares), е екземпляр на класа statsmodels.regression.linear_model.OLS. Той може да се получи по начина показан на фиг. 8.5.

```
model = sm.OLS(y, x)
```

Фиг. 8.5. Създаване на регресионния модел

Тук трябва да се внимава, защото първият аргумент е изходът, следван от входа. Това е обратният ред на съответните методи в scikit-learn.

Има още няколко незадължителни параметъра. Повече информация за този клас, може да се намери на официалната страница с документация https://www.statsmodels.org/dev/generated/statsmodels.regression.linear_model.OLS.html .

След създаване на модела, той може да се напасне към данните като се приложи метода .fit() върху него (фиг. 8.6)

Проект BG05M2OP001-2.016-0003„Модернизация на Национален военен университет "В. Левски" - гр. Велико Търново и Софийски университет "Св. Климент Охридски" - гр. София, в професионално направление 5.3 Компютърна и комуникационна техника“, финансиран от Оперативна програма „Наука и образование за интелигентен растеж“, съфинансирана от Европейския съюз чрез Европейските структурни и инвестиционни фондове.

```
results = model.fit()
```

Фиг. 8.6. Напасване на регресионния модел

При извикване на метода .fit() се получава променливата обект results, която е екземпляр на класа statsmodels.regression.linear_model.RegressionResultsWrapper. Този обект съдържа много информация за регресионния модел.

Стъпка 4: Получаване на резултати

Променливата results се отнася до обекта, който съдържа подробна информация за резултатите от линейната регресия. Извличането на изчислените резултати се извършва с извикване на метода .summary(). По този начин се получава таблицата с резултатите от линейната регресия (фиг. 8.7).

Тази таблица (фиг. 8.7.б) е много изчерпателна. В нея могат да се намерят много статистически стойности, свързани с линейната регресия, включително R^2, b_0, b_1 и b_2.

В този конкретен случай се получава предупреждение, че ексцес тестът kurtosistest е валиден само за n>=20. Това се дължи на малкия брой наблюдения, дадени в примера (n = 8).

Проект BG05M2OP001-2.016-0003„Модернизация на Национален военен университет "В. Левски" - гр. Велико Търново и Софийски университет "Св. Климент Охридски" - гр. София, в професионално направление 5.3 Компютърна и комуникационна техника“, финансиран от Оперативна програма „Наука и образование за интелигентен растеж“, съфинансирана от Европейския съюз чрез Европейските структурни и инвестиционни фондове.

```
print(results.summary())
```

а)

```
C:\Users\Жана\AppData\Local\Programs\Python\Python310\lib\site-
packages\scipy\stats\_stats_py.py:1736: UserWarning: kurtosistest only valid
for n>=20 ... continuing anyway, n=8
  warnings.warn("kurtosistest only valid for n>=20 ... continuing "
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.862
Model:                            OLS   Adj. R-squared:                  0.806
Method:                 Least Squares   F-statistic:                     15.56
Date:                Mon, 30 Jan 2023   Prob (F-statistic):            0.00713
Time:                        15:53:03   Log-Likelihood:                -24.316
No. Observations:                   8   AIC:                             54.63
Df Residuals:                       5   BIC:                             54.87
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          5.5226      4.431      1.246      0.268      -5.867      16.912
x1             0.4471      0.285      1.567      0.178      -0.286       1.180
x2             0.2550      0.453      0.563      0.598      -0.910       1.420
==============================================================================
Omnibus:                        0.561   Durbin-Watson:                   3.268
Prob(Omnibus):                  0.755   Jarque-Bera (JB):                0.534
Skew:                           0.380   Prob(JB):                        0.766
Kurtosis:                       1.987   Cond. No.                         80.1
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is
correctly specified.
```

б)

Фиг. 8.7. Получаване на резултати от линейната регресия OLS

а. Код на Python **б.** Изход от програмата

Проект BG05M2OP001-2.016-0003„Модернизация на Национален военен университет "В. Левски" - гр. Велико Търново и Софийски университет "Св. Климент Охридски" - гр. София, в професионално направление 5.3 Компютърна и комуникационна техника“, финансиран от Оперативна програма „Наука и образование за интелигентен растеж“, съфинансирана от Европейския съюз чрез Европейските структурни и инвестиционни фондове.

Всяка от стойностите от таблицата по-горе можете да се извлече поотделно. Ето един пример (фиг. 8.8).

```
print(f"Coefficient of determination: {results.rsquared}")
print(f"Adjusted coefficient of determination:
{results.rsquared_adj}")
print(f"Regression coefficients: {results.params}")
```

а)

```
Coefficient of determination: 0.8615939258756776
Adjusted coefficient of determination: 0.8062314962259487
Regression coefficients: [5.52257928 0.44706965 0.25502548]
```

б)

Фиг. 8.8. Извличане на най-използваните резултати при линейна регресия

а. Код на Python **б.** Изход от програмата

Изведените резултати на линейната регресия са:

1. .rsquared съдържа R^2.
2. .rsquared_adj представлява коригирано R^2 – т.е. R^2, коригирано според броя на входните функции.
3. .params препраща към масива с b_0, b_1 и b_2.

Получените резултати са идентични с тези, получени със scikit-learn за същия проблем.

Стъпка 5: Прогнозиране на отговора

Прогнозираният отговор на входните стойности, използвани за създаване на модела, могат да се получат като се използват .fittedvalues или .predict() с входния масив като аргумент (фиг. 8.9).

Проект BG05M2OP001-2.016-0003„Модернизация на Национален военен университет "В. Левски" - гр. Велико Търново и Софийски университет "Св. Климент Охридски" - гр. София, в професионално направление 5.3 Компютърна и комуникационна техника“, финансиран от Оперативна програма „Наука и образование за интелигентен растеж“, съфинансирана от Европейския съюз чрез Европейските структурни и инвестиционни фондове.

```
print(f"Predicted response:\n{results.fittedvalues}")
print(f"Predicted response:\n{results.predict(x)}")
```

а)

```
Predicted response:
[ 5.77760476  8.012953   12.73867497 17.9744479  23.97529728
29.4660957 38.78227633 41.27265006]
Predicted response:
[ 5.77760476  8.012953   12.73867497 17.9744479  23.97529728
29.4660957 38.78227633 41.27265006]
```

б)

Фиг. 8.9. Прогнозиране на отговора

а. Код на Python **б.** Изход от програмата

Това е прогнозираният отговор за известни входове. Ако е необходимо да се направят прогнози с нови регресори, може също да се приложи .predict() с нови данни като аргумент (фиг. 8.10).

```
x_new = sm.add_constant(np.arange(10).reshape((-1, 2)))
y_new = results.predict(x_new)
print(f"x_new:\n{x_new}")
print(f"y_new:\n{y_new}")
```

а)

```
x_new:
[[1., 0., 1.],
 [1., 2., 3.],
 [1., 4., 5.],
 [1., 6., 7.],
 [1., 8., 9.]])
y_new:
[5.77760476, 7.18179502, 8.58598528, 9.99017554, 11.3943658]
```

б)

Фиг. 8.9. Прогнозиране на отговора за ново данни

а. Код на Python **б.** Изход от програмата

Проект BG05M2OP001-2.016-0003„Модернизация на Национален военен университет "В. Левски" - гр. Велико Търново и Софийски университет "Св. Климент Охридски" - гр. София, в професионално направление 5.3 Компютърна и комуникационна техника“, финансиран от Оперативна програма „Наука и образование за интелигентен растеж“, съфинансирана от Европейския съюз чрез Европейските структурни и инвестиционни фондове.

Прогнозираните резултати са същите като тези, получени със scikit-learn за същия проблем.

Линейната регресия понякога не е подходяща, особено за нелинейни модели с висока сложност. Съществуват и други техники за регресия, подходящи за случаите, когато линейната регресия не работи добре. Някои от тях са опорни векторни машини, дървета на решенията, произволна гора и невронни мрежи.

Има множество библиотеки на Python за регресия, използваща тези техники. Повечето от тях са безплатни и с отворен код. Това е една от причините Python да е сред основните програмни езици за машинно обучение.

Пакетът scikit-learn предоставя средства за използване на други техники за регресия по много подобен начин на това, което бе разгледано до сега. Той съдържа класове за опорни векторни машини [https://scikit-learn.org/stable/modules/svm.html], дървета на решенията [https://scikit-learn.org/stable/modules/tree.html], случайна гора [https://scikit-learn.org/stable/modules/ensemble.html#forests-of-randomized-trees] и други с методите .fit(), .predict(), .score() и т.н.

В обобщение може да се каже, че линейната регресия се прилага със пакетите:

- NumPy се използва за обработка на масиви.

Проект BG05M2OP001-2.016-0003„Модернизация на Национален военен университет "В. Левски" - гр. Велико Търново и Софийски университет "Св. Климент Охридски" - гр. София, в професионално направление 5.3 Компютърна и комуникационна техника“, финансиран от Оперативна програма „Наука и образование за интелигентен растеж“, съфинансирана от Европейския съюз чрез Европейските структурни и инвестиционни фондове.

- scikit-learn се прилага, ако не са необходими подробни резултати и е необходимо да се използва подхода, съвместим с други техники за регресия.
- statsmodels се прилага, ако има нужда от разширени статистически параметри на модела.

Проект BG05M2OP001-2.016-0003„Модернизация на Национален военен университет "В. Левски" - гр. Велико Търново и Софийски университет "Св. Климент Охридски" - гр. София, в професионално направление 5.3 Компютърна и комуникационна техника“, финансиран от Оперативна програма „Наука и образование за интелигентен растеж“, съфинансирана от Европейския съюз чрез Европейските структурни и инвестиционни фондове.