Actividad2, Técnicas Multivariantes

Jorge A. Balsells Orellana

May 17, 2021

En esta actividad vas a profundizar en las distintas técnicas que se pueden aplicar para abordar un problema de regresión. Además, profundizarás en tus conocimientos sobre las librerías statsmodels y scikit-learn de Python.

0.1 Importando librerías

[1]:
```
from sklearn.model_selection import train_test_split
from sklearn.datasets import make_regression
from sklearn import linear_model
import matplotlib.pyplot as plt
import statsmodels.api as sm
import sklearn as sk
import numpy as np
import pandas as pd
```

1 Creando dataset

El primer paso consiste en crear un conjunto de datos ficticio. Para garantizar que cada alumno obtiene uno distinto se va a emplear el documento de identidad de cada uno para crear el conjunto de datos. Para que sean comparables entre todos, si el número de identidad tiene menos de 8 cifras replicaremos las primeras hasta obtener exactamente 8. Además, para evitar los dígitos 0 y 1, si alguna de las cifras es menor que 2 la sustituiremos por ese número. Aplicando estos cambios tendremos el número del documento de identidad preparado para la resolución de la actividad. Numero de identificación personal: 166315389-0101, equivale a 26632538

[2]:
```
data = sk.datasets.make_regression(
    n_samples=200+10*2,
    n_features=10+6+6,
    n_informative=10+6,
    n_targets=1,
    noise=10*3,
    shuffle=False,
    bias=2,
    random_state=None,effective_rank=None,
    tail_strength=0.5,  coef=False
)
```

2 Agregando títulos

[3]:
```python
data_fr = pd.DataFrame(data[0])
data_fr.columns=['c0','c1','c2','c3','c4','c5','c6','
c7','c8','c9','c10','c11','c12','c13','c14','c15','c16',
'c17','c18','c19','c20','c21']
print(data_fr)
```

```
           c0        c1        c2        c3        c4        c5        c6  \
0   -1.200619  0.363886  0.527277  1.776005  1.562085 -0.998717 -1.445892
1   -0.370290 -0.721044 -0.004600  1.988354 -1.187087  1.929338  1.106362
2   -0.083060 -0.848515 -0.666145  2.155664  1.217626  0.442125 -0.066560
3    0.547946  0.137935  0.803414  1.647611 -0.406032 -2.838325  0.998711
4   -0.478050  0.939166  1.590225 -2.444970 -1.545710  0.086851 -0.821011
..        ...       ...       ...       ...       ...       ...       ...
215  0.076349 -0.294829  0.649279 -1.841271  0.515866 -0.785015  0.919299
216 -1.447078  0.530184  0.876954 -0.646612  0.652381 -0.302676 -1.117376
217 -0.197006 -2.568405  0.944425  1.424370  1.913104  0.213528  0.841183
218  0.148215  1.986257  1.706012  1.239836 -0.204384 -1.498575  1.656196
219  0.101630  1.352317  0.540504  1.114844 -0.140460 -1.433073  1.174183

           c7        c8        c9  ...       c12       c13       c14  \
0    1.007802 -0.210287 -0.315635  ... -0.846384  0.751240 -0.831247
1   -1.119966  1.421816 -0.070631  ... -0.386086  0.305581  1.619579
2   -0.238667  0.858434 -0.115024  ...  0.949134 -1.379160  1.224498
3   -1.523243 -0.242737 -1.103062  ...  0.449596 -2.452147 -1.325093
4    0.187369 -0.064385  0.240592  ...  0.141086 -0.828289  1.132770
..        ...       ...       ...  ...       ...       ...       ...
215 -0.460281  1.771298 -0.550230  ... -0.181780 -1.344381 -0.942128
216 -0.952387  0.646239 -0.520901  ... -0.118848 -0.840821  0.817320
217  1.971822  1.805175 -1.793488  ...  0.081922 -0.748347 -1.119656
218  1.336042 -0.428932  0.776098  ...  2.081237 -0.391987 -0.439679
219  1.167033 -0.296465 -1.497535  ...  0.890979 -0.105300  1.564012

          c15       c16       c17       c18       c19       c20       c21
0   -2.234769 -1.388944  0.341197  1.684392 -0.810816  0.172313 -0.943393
1    0.406285 -1.096114 -0.904892 -1.189931 -0.551215  0.579500 -2.862278
2    0.048189 -1.182454 -1.991184 -0.456665 -0.371789 -0.618690 -0.531200
3   -0.034923  0.227892  0.140545 -0.679035 -1.118085  0.325487 -1.751562
4   -2.100005 -0.793645  1.414209  0.430947 -1.358468  1.061865  0.168035
..        ...       ...       ...       ...       ...       ...       ...
215 -0.763008  0.103196  0.059206  0.813484  0.177341  0.376273  0.912228
216  0.154747  0.249033  0.124586  0.790569  1.317087  1.068314 -0.180360
217 -1.136287 -1.632205 -0.185916  0.315124  0.350710  0.078135 -1.662309
218 -0.825360 -0.954144  0.378729  1.972238 -0.975027 -0.040878 -0.707584
219  0.717811 -0.886581  0.015919  0.702115  0.673015 -1.029820 -0.050169
```

```
[220 rows x 22 columns]
```

3 Dividiendo Dataframe a través de un Split

Divide el conjunto de datos en 200 observaciones para el entrenamiento y el resto para realizar la validación de los distintos métodos de regresión aplicados.

[4]:
```
train_set, test_set, y_train, y_test = train_test_split(data_fr, data[1],␣
 ↪test_size=0.09, random_state=42)

print(f"tamaño entrenamiento features {train_set.shape}")
print(f"tamaño test features {test_set.shape}")
print(f"tamaño entrenamiento respuesta {y_train.shape}")
print(f"tamaño test respuesta {y_test.shape}")
```

```
tamaño entrenamiento features (200, 22)
tamaño test features (20, 22)
tamaño entrenamiento respuesta (200,)
tamaño test respuesta (20,)
```

Describe tu conjunto de datos (transformalo en un data.frame, aplica los métodos .info(), .describe() y obtén el histograma de todas las variables (predictoras y la variable respuesta).

[5]:
```
train_set.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 200 entries, 189 to 102
Data columns (total 22 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   c0      200 non-null    float64
 1   c1      200 non-null    float64
 2   c2      200 non-null    float64
 3   c3      200 non-null    float64
 4   c4      200 non-null    float64
 5   c5      200 non-null    float64
 6   c6      200 non-null    float64
 7   c7      200 non-null    float64
 8   c8      200 non-null    float64
 9   c9      200 non-null    float64
 10  c10     200 non-null    float64
 11  c11     200 non-null    float64
 12  c12     200 non-null    float64
 13  c13     200 non-null    float64
 14  c14     200 non-null    float64
 15  c15     200 non-null    float64
 16  c16     200 non-null    float64
 17  c17     200 non-null    float64
```

```
 18   c18      200 non-null    float64
 19   c19      200 non-null    float64
 20   c20      200 non-null    float64
 21   c21      200 non-null    float64
dtypes: float64(22)
memory usage: 35.9 KB
```

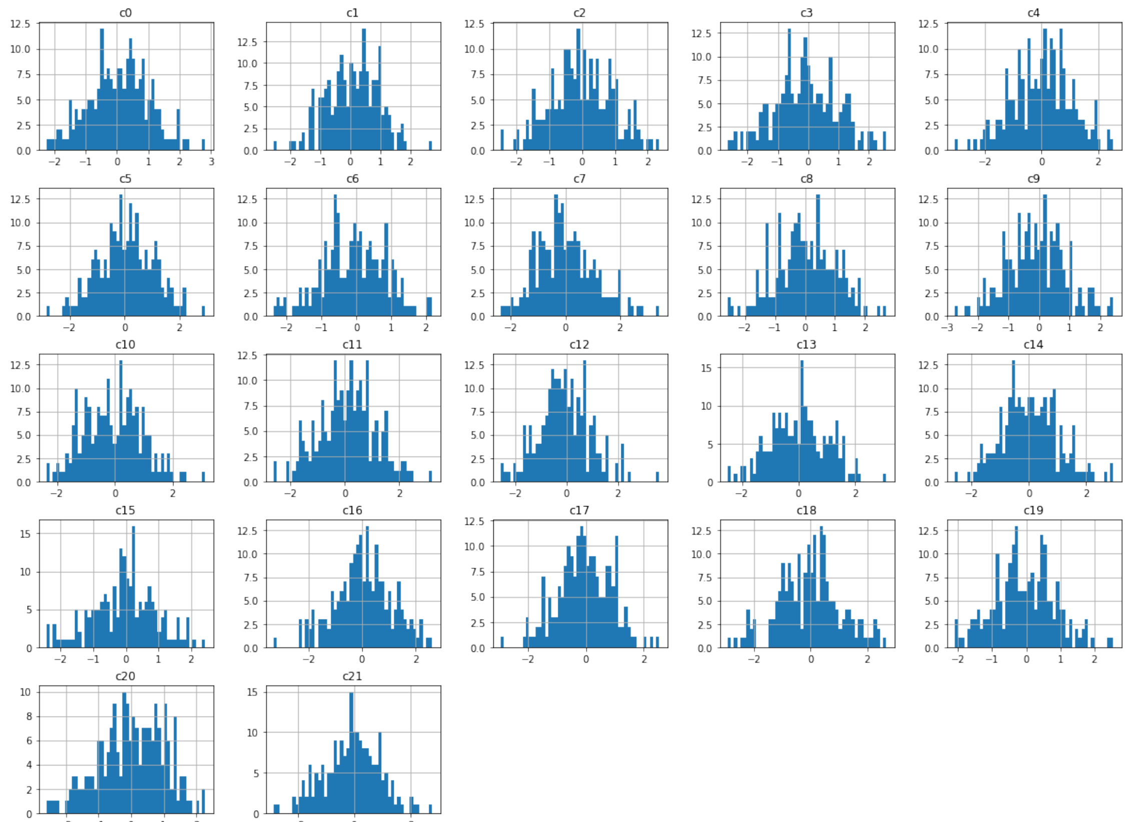
[6]:
```
train_set.describe()
```

[6]:
```
               c0          c1          c2          c3          c4          c5  \
count  200.000000  200.000000  200.000000  200.000000  200.000000  200.000000
mean     0.061775    0.068480   -0.029100   -0.079289    0.024882    0.058193
std      0.951104    0.859848    0.935541    1.045438    1.029154    0.990542
min     -2.257323   -2.568405   -2.482018   -2.660958   -3.065924   -2.838325
25%     -0.533241   -0.546414   -0.606245   -0.740814   -0.688340   -0.597589
50%      0.102723    0.133160   -0.073856   -0.105574    0.089689    0.072999
75%      0.744515    0.681227    0.627146    0.691111    0.691641    0.776711
max      2.827125    2.731737    2.353466    2.578257    2.533175    2.936126

               c6          c7          c8          c9  ...         c12  \
count  200.000000  200.000000  200.000000  200.000000  ...  200.000000
mean    -0.061592    0.020946    0.024503   -0.093435  ...   -0.072567
std      0.898575    1.046160    0.983612    0.951497  ...    1.004430
min     -2.377142   -2.372431   -2.582735   -2.757625  ...   -2.571439
25%     -0.645426   -0.744354   -0.657205   -0.675275  ...   -0.668238
50%     -0.062011   -0.085418    0.019527   -0.079579  ...   -0.108140
75%      0.584005    0.664773    0.690342    0.540392  ...    0.602127
max      2.180832    3.437404    2.713194    2.434474  ...    3.605012

              c13         c14         c15         c16         c17         c18  \
count  200.000000  200.000000  200.000000  200.000000  200.000000  200.000000
mean     0.018591    0.064096   -0.042001    0.051858   -0.066966   -0.010057
std      1.008430    0.987427    0.961506    1.068868    0.910952    1.071814
min     -2.511117   -2.603918   -2.434906   -3.314557   -2.959647   -2.939178
25%     -0.699544   -0.597825   -0.652038   -0.604215   -0.633148   -0.712945
50%      0.082356    0.013870   -0.005546    0.034842   -0.064684    0.023465
75%      0.686061    0.741414    0.602292    0.708587    0.621575    0.614373
max      3.105076    2.967300    2.438551    2.630969    2.541617    2.675915

              c19         c20         c21
count  200.000000  200.000000  200.000000
mean    -0.048863    0.036259   -0.090818
std      0.893632    0.995174    0.973091
min     -2.101485   -2.593781   -2.862278
25%     -0.590866   -0.631175   -0.696132
50%     -0.114663    0.061491   -0.049538
75%      0.520805    0.803799    0.533666
```

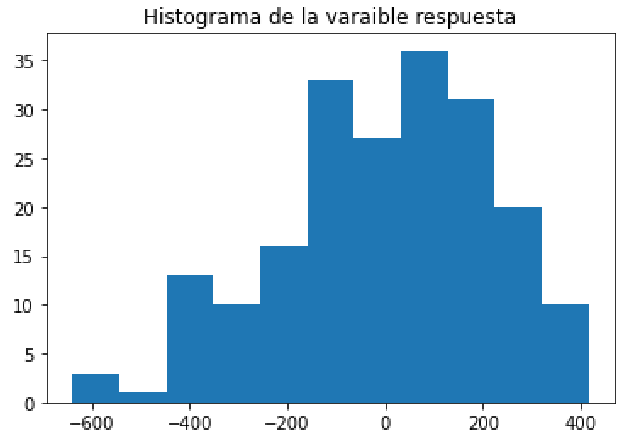
```
max      2.559372    2.276325    2.769567

[8 rows x 22 columns]
```

4 Histogramas de features

[7]:
```python
train_set.hist(bins=50, figsize=(20,15))
plt.show()
```



[8]:
```python
plt.hist(y_train, bins='auto')
plt.title("Histograma de la varaible respuesta")
plt.show()
```



5 Regresión lineal múltiple

Obtén un modelo de regresión lineal múltiple. ¿Son todas las variables predictoras significativas? Utiliza la librería statsmodels. **No son todas las variables predictoras significativas, ya que algunas no cumplen con $P > |t|$.**

[9]:
```
train_data = train_set.copy()
train_data['y'] = y_train
mat_corr = train_data.corr()
mat_corr['y']
```

[9]:
```
c0     0.364644
c1     0.250325
c2    -0.107469
c3     0.424363
c4    -0.037278
c5     0.351267
c6     0.322329
c7     0.231789
c8    -0.018025
c9     0.251766
c10    0.130534
c11    0.168569
c12    0.118699
c13    0.161102
c14    0.437383
c15    0.233611
c16    0.062251
c17   -0.133591
c18   -0.017822
c19    0.024090
```

```
c20   -0.070258
c21    0.031895
y      1.000000
Name: y, dtype: float64
```

6 Entrenamiento modelo 1

[11]:
```
train_set = sm.add_constant(train_set, prepend=True)
model1 = sm.OLS(endog=y_train, exog=train_set)
model1 = model1.fit()
```

[12]:
```
print(model1.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.982
Model:                            OLS   Adj. R-squared:                  0.979
Method:                 Least Squares   F-statistic:                     430.1
Date:                Mon, 17 May 2021   Prob (F-statistic):          2.78e-141
Time:                        01:34:36   Log-Likelihood:                -959.93
No. Observations:                 200   AIC:                             1966.
Df Residuals:                     177   BIC:                             2042.
Df Model:                          22
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.2033      2.326     -0.087      0.930      -4.794       4.388
c0            71.3972      2.483     28.758      0.000      66.498      76.297
c1            65.7305      2.717     24.189      0.000      60.368      71.093
c2             2.6603      2.580      1.031      0.304      -2.431       7.751
c3            96.9798      2.290     42.354      0.000      92.461     101.499
c4            17.8190      2.313      7.705      0.000      13.255      22.383
c5            55.8696      2.350     23.774      0.000      51.232      60.507
c6            65.0975      2.617     24.878      0.000      59.934      70.261
c7            42.9200      2.204     19.472      0.000      38.570      47.270
c8             7.8005      2.340      3.334      0.001       3.183      12.418
c9            62.1009      2.541     24.439      0.000      57.086      67.116
c10           22.1543      2.280      9.719      0.000      17.656      26.653
c11           18.4234      2.227      8.272      0.000      14.028      22.819
c12           31.9253      2.329     13.706      0.000      27.329      36.522
c13           39.7516      2.318     17.148      0.000      35.177      44.326
c14           98.2821      2.323     42.301      0.000      93.697     102.867
c15           45.2187      2.473     18.288      0.000      40.339      50.098
c16            2.7655      2.168      1.276      0.204      -1.513       7.044
c17            2.6992      2.632      1.025      0.307      -2.496       7.894
```

```
c18              2.1383      2.263      0.945      0.346     -2.328       6.604
c19              0.7532      2.523      0.299      0.766     -4.225       5.731
c20             -3.2485      2.373     -1.369      0.173     -7.931       1.434
c21             -0.7817      2.521     -0.310      0.757     -5.756       4.193
==============================================================================
Omnibus:                        6.271   Durbin-Watson:                   1.991
Prob(Omnibus):                  0.043   Jarque-Bera (JB):                9.922
Skew:                           0.045   Prob(JB):                      0.00701
Kurtosis:                       4.087   Cond. No.                         2.01
==============================================================================

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

7 Aplicando Step-Wise

Realiza una selección de variables mediante un algoritmo de tipo step-wise, donde en cada paso elimines la variable predictora menos significativa atendiendo a su p.valor hasta que todas las variables del modelo sean significativas (p.valor < 0.05).

En este caso, el valor de $P > |t|$ debe cumplir, siendo P menor a $1.35 * e^{-142}$... A continuación se eliminan todas las columnas del train-set que correspondan a valores mayores de P. En este caso, corresponde inicialmente a la columna **C19**, **C21**, **C18**, **C17**, **C20**, **C2** y **C16**.

[13]:
```python
train_set.drop('c19', axis='columns', inplace=True)
train_set = sm.add_constant(train_set, prepend=True)
model2 = sm.OLS(endog=y_train, exog=train_set)
model2 = model2.fit()
```

[14]:
```python
print(model2.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.982
Model:                            OLS   Adj. R-squared:                  0.979
Method:                 Least Squares   F-statistic:                     452.9
Date:                Mon, 17 May 2021   Prob (F-statistic):          1.35e-142
Time:                        01:34:53   Log-Likelihood:                -959.98
No. Observations:                 200   AIC:                             1964.
Df Residuals:                     178   BIC:                             2037.
Df Model:                          21
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.2604      2.313     -0.113      0.910     -4.824       4.303
c0            71.4167      2.476     28.849      0.000     66.532      76.302
```

```
c1            65.7940      2.702     24.349      0.000      60.462      71.126
c2             2.6727      2.573      1.039      0.300      -2.404       7.750
c3            96.9954      2.283     42.480      0.000      92.490     101.501
c4            17.8809      2.297      7.783      0.000      13.347      22.414
c5            55.8882      2.343     23.851      0.000      51.264      60.512
c6            65.0868      2.610     24.940      0.000      59.937      70.237
c7            42.8624      2.190     19.571      0.000      38.540      47.184
c8             7.8136      2.333      3.349      0.001       3.209      12.418
c9            62.0935      2.534     24.500      0.000      57.092      67.095
c10           22.1471      2.274      9.741      0.000      17.660      26.634
c11           18.4483      2.220      8.310      0.000      14.067      22.829
c12           31.8587      2.313     13.776      0.000      27.295      36.422
c13           39.7645      2.312     17.200      0.000      35.202      44.327
c14           98.2918      2.317     42.418      0.000      93.719     102.865
c15           45.2141      2.466     18.333      0.000      40.347      50.081
c16            2.8038      2.158      1.299      0.196      -1.456       7.063
c17            2.6858      2.625      1.023      0.308      -2.495       7.867
c18            2.1186      2.256      0.939      0.349      -2.334       6.571
c20           -3.2184      2.364     -1.361      0.175      -7.884       1.447
c21           -0.7839      2.514     -0.312      0.756      -5.746       4.178
==============================================================================
Omnibus:                        6.547   Durbin-Watson:                   1.988
Prob(Omnibus):                  0.038   Jarque-Bera (JB):               10.683
Skew:                           0.033   Prob(JB):                      0.00479
Kurtosis:                       4.130   Cond. No.                         1.99
==============================================================================

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

[15]:

```
train_set.drop('c21', axis='columns', inplace=True)
train_set = sm.add_constant(train_set, prepend=True)
model2 = sm.OLS(endog=y_train, exog=train_set)
model2 = model2.fit()
```

[16]:

```
print(model2.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.982
Model:                            OLS   Adj. R-squared:                  0.980
Method:                 Least Squares   F-statistic:                     478.0
Date:                Mon, 17 May 2021   Prob (F-statistic):          6.39e-144
Time:                        01:35:05   Log-Likelihood:                -960.03
No. Observations:                 200   AIC:                             1962.
Df Residuals:                     179   BIC:                             2031.
Df Model:                          20
```

```
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.1809      2.293     -0.079      0.937      -4.705       4.343
c0            71.2484      2.410     29.566      0.000      66.493      76.004
c1            65.8150      2.694     24.426      0.000      60.498      71.132
c2             2.7500      2.554      1.077      0.283      -2.290       7.790
c3            97.1219      2.241     43.332      0.000      92.699     101.545
c4            17.9285      2.286      7.841      0.000      13.417      22.440
c5            55.9942      2.313     24.213      0.000      51.431      60.558
c6            65.1156      2.602     25.030      0.000      59.982      70.249
c7            42.8445      2.184     19.619      0.000      38.535      47.154
c8             7.8812      2.317      3.401      0.001       3.308      12.454
c9            61.9033      2.454     25.228      0.000      57.061      66.745
c10           22.2037      2.261      9.822      0.000      17.743      26.665
c11           18.4284      2.214      8.325      0.000      14.060      22.796
c12           31.8425      2.306     13.807      0.000      27.292      36.393
c13           39.7847      2.305     17.259      0.000      35.236      44.333
c14           98.2064      2.295     42.789      0.000      93.677     102.735
c15           45.3165      2.438     18.587      0.000      40.505      50.128
c16            2.7531      2.147      1.282      0.201      -1.483       6.990
c17            2.6668      2.618      1.019      0.310      -2.499       7.833
c18            2.0328      2.234      0.910      0.364      -2.375       6.441
c20           -3.1425      2.346     -1.340      0.182      -7.772       1.487
==============================================================================
Omnibus:                        6.356   Durbin-Watson:                   1.988
Prob(Omnibus):                  0.042   Jarque-Bera (JB):               10.201
Skew:                           0.030   Prob(JB):                      0.00609
Kurtosis:                       4.105   Cond. No.                         1.88
==============================================================================

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

[17]:
```python
train_set.drop('c18', axis='columns', inplace=True)
train_set = sm.add_constant(train_set, prepend=True)
model2 = sm.OLS(endog=y_train, exog=train_set)
model2 = model2.fit()
```

[18]:
```python
print(model2.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.982
Model:                            OLS   Adj. R-squared:                  0.980
Method:                 Least Squares   F-statistic:                     503.6
```

```
Date:                 Mon, 17 May 2021   Prob (F-statistic):          4.24e-145
Time:                         01:35:19   Log-Likelihood:                -960.50
No. Observations:                  200   AIC:                             1961.
Df Residuals:                      180   BIC:                             2027.
Df Model:                           19
Covariance Type:             nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.3254      2.286     -0.142      0.887      -4.836       4.186
c0            71.3017      2.408     29.611      0.000      66.550      76.053
c1            65.8078      2.693     24.435      0.000      60.494      71.122
c2             3.0360      2.534      1.198      0.232      -1.964       8.036
c3            97.3354      2.228     43.689      0.000      92.939     101.732
c4            17.9433      2.285      7.852      0.000      13.434      22.453
c5            56.0758      2.310     24.278      0.000      51.518      60.633
c6            64.8721      2.586     25.081      0.000      59.768      69.976
c7            42.7109      2.178     19.611      0.000      38.413      47.008
c8             8.0906      2.305      3.510      0.001       3.542      12.639
c9            61.4269      2.396     25.636      0.000      56.699      66.155
c10           21.9495      2.242      9.789      0.000      17.525      26.374
c11           18.5601      2.208      8.407      0.000      14.204      22.916
c12           31.4831      2.271     13.863      0.000      27.002      35.964
c13           40.0765      2.282     17.565      0.000      35.574      44.579
c14           98.2379      2.294     42.828      0.000      93.712     102.764
c15           45.5399      2.425     18.783      0.000      40.756      50.324
c16            2.8142      2.145      1.312      0.191      -1.418       7.047
c17            2.5329      2.613      0.969      0.334      -2.622       7.688
c20           -2.7757      2.310     -1.202      0.231      -7.334       1.782
==============================================================================
Omnibus:                        6.022   Durbin-Watson:                   1.989
Prob(Omnibus):                  0.049   Jarque-Bera (JB):                9.415
Skew:                           0.007   Prob(JB):                      0.00903
Kurtosis:                       4.063   Cond. No.                         1.78
==============================================================================

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

[19]:

```
train_set.drop('c17', axis='columns', inplace=True)
train_set = sm.add_constant(train_set, prepend=True)
model2 = sm.OLS(endog=y_train, exog=train_set)
model2 = model2.fit()
```

[20]:

```
print(model2.summary())
```

```
                            OLS Regression Results
```

```
==============================================================================
Dep. Variable:                      y   R-squared:                       0.981
Model:                            OLS   Adj. R-squared:                  0.980
Method:                 Least Squares   F-statistic:                     531.7
Date:                Mon, 17 May 2021   Prob (F-statistic):          2.89e-146
Time:                        01:35:32   Log-Likelihood:                -961.02
No. Observations:                 200   AIC:                             1960.
Df Residuals:                     181   BIC:                             2023.
Df Model:                          18
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.5507      2.274     -0.242      0.809      -5.037       3.936
c0            71.0005      2.387     29.739      0.000      66.290      75.711
c1            65.4634      2.669     24.526      0.000      60.197      70.730
c2             2.9783      2.533      1.176      0.241      -2.019       7.975
c3            96.8253      2.165     44.732      0.000      92.554     101.096
c4            18.0813      2.280      7.929      0.000      13.582      22.581
c5            56.1468      2.308     24.325      0.000      51.592      60.701
c6            64.8758      2.586     25.087      0.000      59.773      69.978
c7            42.9235      2.166     19.813      0.000      38.649      47.198
c8             7.9689      2.301      3.463      0.001       3.428      12.509
c9            61.4172      2.396     25.637      0.000      56.690      66.144
c10           21.9190      2.242      9.778      0.000      17.496      26.342
c11           18.8089      2.192      8.579      0.000      14.483      23.135
c12           31.2239      2.255     13.847      0.000      26.775      35.673
c13           40.1395      2.280     17.603      0.000      35.640      44.639
c14           98.2658      2.293     42.851      0.000      93.741     102.791
c15           45.4753      2.423     18.767      0.000      40.694      50.257
c16            2.6426      2.137      1.237      0.218      -1.574       6.860
c20           -2.6741      2.307     -1.159      0.248      -7.226       1.878
==============================================================================
Omnibus:                        5.772   Durbin-Watson:                   1.997
Prob(Omnibus):                  0.056   Jarque-Bera (JB):                8.784
Skew:                           0.024   Prob(JB):                       0.0124
Kurtosis:                       4.026   Cond. No.                         1.74
==============================================================================

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

[21]:

```
train_set.drop('c20', axis='columns', inplace=True)
train_set = sm.add_constant(train_set, prepend=True)
model2 = sm.OLS(endog=y_train, exog=train_set)
model2 = model2.fit()
```

[22]:
```python
print(model2.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.981
Model:                            OLS   Adj. R-squared:                  0.980
Method:                 Least Squares   F-statistic:                     561.8
Date:                Mon, 17 May 2021   Prob (F-statistic):          2.34e-147
Time:                        01:35:44   Log-Likelihood:                -961.76
No. Observations:                 200   AIC:                             1960.
Df Residuals:                     182   BIC:                             2019.
Df Model:                          17
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.6684      2.274     -0.294      0.769      -5.155       3.818
c0            70.9421      2.389     29.694      0.000      66.228      75.656
c1            65.6803      2.665     24.645      0.000      60.422      70.939
c2             3.1229      2.532      1.233      0.219      -1.873       8.119
c3            97.1546      2.148     45.233      0.000      92.917     101.392
c4            18.5179      2.251      8.225      0.000      14.076      22.960
c5            56.0162      2.308     24.275      0.000      51.463      60.569
c6            64.7596      2.587     25.037      0.000      59.656      69.863
c7            42.8632      2.168     19.772      0.000      38.586      47.141
c8             8.0017      2.303      3.474      0.001       3.457      12.546
c9            61.2205      2.392     25.595      0.000      56.501      65.940
c10           22.0990      2.238      9.873      0.000      17.683      26.515
c11           18.8339      2.194      8.583      0.000      14.504      23.164
c12           31.1036      2.255     13.795      0.000      26.655      35.552
c13           40.0868      2.282     17.566      0.000      35.584      44.589
c14           98.4499      2.290     42.994      0.000      93.932     102.968
c15           45.7302      2.415     18.932      0.000      40.964      50.496
c16            2.8738      2.130      1.349      0.179      -1.329       7.076
==============================================================================
Omnibus:                        6.004   Durbin-Watson:                   1.997
Prob(Omnibus):                  0.050   Jarque-Bera (JB):                9.310
Skew:                           0.034   Prob(JB):                      0.00952
Kurtosis:                       4.055   Cond. No.                         1.68
==============================================================================

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

[23]:
```python
train_set.drop('c2', axis='columns', inplace=True)
train_set = sm.add_constant(train_set, prepend=True)
```

```python
model2 = sm.OLS(endog=y_train, exog=train_set)
model2 = model2.fit()
```

[24]:
```python
print(model2.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.981
Model:                            OLS   Adj. R-squared:                  0.979
Method:                 Least Squares   F-statistic:                     595.1
Date:                Mon, 17 May 2021   Prob (F-statistic):          2.01e-148
Time:                        01:35:58   Log-Likelihood:                -962.59
No. Observations:                 200   AIC:                             1959.
Df Residuals:                     183   BIC:                             2015.
Df Model:                          16
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.6630      2.277     -0.291      0.771      -5.155       3.829
c0            70.6606      2.382     29.669      0.000      65.962      75.360
c1            65.4699      2.663     24.581      0.000      60.215      70.725
c3            97.0777      2.150     45.152      0.000      92.836     101.320
c4            18.0597      2.224      8.122      0.000      13.672      22.447
c5            55.5940      2.285     24.327      0.000      51.085      60.103
c6            64.7329      2.590     24.992      0.000      59.623      69.843
c7            42.8066      2.170     19.723      0.000      38.524      47.089
c8             7.7933      2.300      3.388      0.001       3.255      12.332
c9            61.5895      2.376     25.916      0.000      56.901      66.278
c10           22.2372      2.239      9.933      0.000      17.820      26.654
c11           19.2248      2.174      8.841      0.000      14.935      23.515
c12           31.1868      2.257     13.819      0.000      26.734      35.640
c13           39.5467      2.243     17.633      0.000      35.122      43.972
c14           98.3032      2.290     42.927      0.000      93.785     102.821
c15           45.5222      2.413     18.865      0.000      40.761      50.283
c16            2.7093      2.129      1.273      0.205      -1.491       6.909
==============================================================================
Omnibus:                        5.220   Durbin-Watson:                   2.021
Prob(Omnibus):                  0.074   Jarque-Bera (JB):                7.417
Skew:                           0.052   Prob(JB):                       0.0245
Kurtosis:                       3.938   Cond. No.                         1.61
==============================================================================

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

[26]:
```python
train_set.drop('c16', axis='columns', inplace=True)
train_set = sm.add_constant(train_set, prepend=True)
model2 = sm.OLS(endog=y_train, exog=train_set)
model2 = model2.fit()
```

[27]:
```python
print(model2.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.981
Model:                            OLS   Adj. R-squared:                  0.979
Method:                 Least Squares   F-statistic:                     632.6
Date:                Mon, 17 May 2021   Prob (F-statistic):          1.75e-149
Time:                        01:36:18   Log-Likelihood:                -963.47
No. Observations:                 200   AIC:                             1959.
Df Residuals:                     184   BIC:                             2012.
Df Model:                          15
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.4821      2.276     -0.212      0.832      -4.973       4.009
c0            70.6325      2.386     29.609      0.000      65.926      75.339
c1            65.7698      2.657     24.749      0.000      60.527      71.013
c3            97.0603      2.154     45.069      0.000      92.811     101.309
c4            18.0374      2.227      8.098      0.000      13.643      22.432
c5            55.5705      2.289     24.276      0.000      51.054      60.087
c6            64.6963      2.594     24.938      0.000      59.578      69.815
c7            43.0078      2.168     19.835      0.000      38.730      47.286
c8             7.7346      2.304      3.358      0.001       3.190      12.279
c9            61.6922      2.379     25.931      0.000      56.998      66.386
c10           22.0828      2.239      9.862      0.000      17.665      26.501
c11           19.0877      2.175      8.774      0.000      14.796      23.380
c12           31.6224      2.234     14.152      0.000      27.214      36.031
c13           39.3717      2.242     17.558      0.000      34.948      43.796
c14           98.2735      2.294     42.844      0.000      93.748     102.799
c15           45.8517      2.403     19.080      0.000      41.110      50.593
==============================================================================
Omnibus:                        5.633   Durbin-Watson:                   2.036
Prob(Omnibus):                  0.060   Jarque-Bera (JB):                8.378
Skew:                           0.046   Prob(JB):                       0.0152
Kurtosis:                       3.999   Cond. No.                         1.60
==============================================================================

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

8 Regresión mediante la red elástica

Realiza una regresión mediante la red elástica. Prueba distintos valores de r y obtén el valor óptimo de r y de mediante validación cruzada.

[28]:
```
from sklearn.model_selection import RepeatedKFold
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import mean_squared_error
from sklearn.linear_model import ElasticNet
```

[29]:
```
parameters = {
    'alpha':np.arange(0.005,0.9,0.01),
    'l1_ratio':np.arange(0.1,0.5,0.01)
}
print(parameters)
```

```
{'alpha': array([0.005, 0.015, 0.025, 0.035, 0.045, 0.055, 0.065, 0.075, 0.085,
       0.095, 0.105, 0.115, 0.125, 0.135, 0.145, 0.155, 0.165, 0.175,
       0.185, 0.195, 0.205, 0.215, 0.225, 0.235, 0.245, 0.255, 0.265,
       0.275, 0.285, 0.295, 0.305, 0.315, 0.325, 0.335, 0.345, 0.355,
       0.365, 0.375, 0.385, 0.395, 0.405, 0.415, 0.425, 0.435, 0.445,
       0.455, 0.465, 0.475, 0.485, 0.495, 0.505, 0.515, 0.525, 0.535,
       0.545, 0.555, 0.565, 0.575, 0.585, 0.595, 0.605, 0.615, 0.625,
       0.635, 0.645, 0.655, 0.665, 0.675, 0.685, 0.695, 0.705, 0.715,
       0.725, 0.735, 0.745, 0.755, 0.765, 0.775, 0.785, 0.795, 0.805,
       0.815, 0.825, 0.835, 0.845, 0.855, 0.865, 0.875, 0.885, 0.895]),
'l1_ratio': array([0.1 , 0.11, 0.12, 0.13, 0.14, 0.15, 0.16, 0.17, 0.18, 0.19,
0.2 ,
       0.21, 0.22, 0.23, 0.24, 0.25, 0.26, 0.27, 0.28, 0.29, 0.3 , 0.31,
       0.32, 0.33, 0.34, 0.35, 0.36, 0.37, 0.38, 0.39, 0.4 , 0.41, 0.42,
       0.43, 0.44, 0.45, 0.46, 0.47, 0.48, 0.49])}
```

[30]:
```
cv = RepeatedKFold(n_splits=5, n_repeats=2, random_state=42)
print(cv)
```

```
RepeatedKFold(n_repeats=2, n_splits=5, random_state=42)
```

[31]:
```
elasticNet = ElasticNet()
Srch = GridSearchCV(elasticNet, parameters, scoring='neg_mean_squared_error',
 ↪cv=cv, n_jobs=-1)
result = Srch.fit(train_set,y_train)
```

[32]:
```
print(result)
print(result.best_score_)
print(result.best_params_)

best_elastic = result.best_estimator_
print(best_elastic)
```

```
GridSearchCV(cv=RepeatedKFold(n_repeats=2, n_splits=5, random_state=42),
             estimator=ElasticNet(), n_jobs=-1,
             param_grid={'alpha': array([0.005, 0.015, 0.025, 0.035, 0.045,
0.055, 0.065, 0.075, 0.085,
       0.095, 0.105, 0.115, 0.125, 0.135, 0.145, 0.155, 0.165, 0.175,
       0.185, 0.195, 0.205, 0.215, 0.225, 0.235, 0.245, 0.255, 0.265,
       0.275, 0.285, 0.295, 0.305, 0.315, 0.325, 0.335, 0.345, 0.355,
       0.36...
       0.725, 0.735, 0.745, 0.755, 0.765, 0.775, 0.785, 0.795, 0.805,
       0.815, 0.825, 0.835, 0.845, 0.855, 0.865, 0.875, 0.885, 0.895]),
                         'l1_ratio': array([0.1 , 0.11, 0.12, 0.13, 0.14, 0.15,
0.16, 0.17, 0.18, 0.19, 0.2 ,
       0.21, 0.22, 0.23, 0.24, 0.25, 0.26, 0.27, 0.28, 0.29, 0.3 , 0.31,
       0.32, 0.33, 0.34, 0.35, 0.36, 0.37, 0.38, 0.39, 0.4 , 0.41, 0.42,
       0.43, 0.44, 0.45, 0.46, 0.47, 0.48, 0.49])},
             scoring='neg_mean_squared_error')
-1047.2692562635616
{'alpha': 0.005, 'l1_ratio': 0.48999999999999977}
ElasticNet(alpha=0.005, l1_ratio=0.48999999999999977)
```

9 Comprobación de error cuadrático medio

Comprueba con la muestra de validación con cuál de los tres modelos se obtiene un menor error cuadrático medio.

9.1 Modelo 1

[33]:
```
test_set = sm.add_constant(test_set, prepend=True)
y_predict_model_1 = model1.predict(test_set)
Mean_Sq_Error = mean_squared_error(y_predict_model_1, y_test)
print(Mean_Sq_Error)
```

```
750.5664746002329
```

9.2 Modelo 2

[34]:
```
test_set.drop(columns=['c2','c16','c17','c18','c19','c20','c21'],inplace=True)
y_predict_model_2 = model2.predict(test_set)
```

[36]:
```
Mean_Sq_Error_2 = mean_squared_error(y_predict_model_2, y_test)
print(Mean_Sq_Error_2)
```

```
803.4930275424474
```

9.3 Modelo 3

[37]:
```
y_predict_model_3 = best_elastic.predict(test_set)
```

[39]:
```
Mean_Sq_Error_3 = mean_squared_error(y_predict_model_3, y_test)
print(Mean_Sq_Error_3)
```

```
796.5599792561268
```

10 Resultados de predicciones

[48]:
```
resultados_dict ={
    'y_true' : y_test,
    'pred_modelo_1': y_predict_model_1,
    'pred_modelo_2': y_predict_model_2,
    'pred_modelo_3': y_predict_model_3,
}

resultados_df = pd.DataFrame(resultados_dict)

resultados_df
```

[48]:

	y_true	pred_modelo_1	pred_modelo_2	pred_modelo_3
132	333.020602	337.500686	337.793499	336.921428
148	48.171506	70.577423	68.063862	67.811020
93	-2.486599	6.473093	12.315322	12.395744
180	117.144811	98.533880	100.483649	100.065695
15	219.582261	179.860269	178.777864	178.464100
115	118.285486	93.841975	92.082986	91.826115
172	35.675697	92.450567	99.545055	99.347175
209	-165.223912	-159.573230	-164.636243	-164.098968
75	-102.658248	-111.493527	-112.536782	-112.142209
142	78.210423	44.007023	51.301069	51.217902
100	136.480734	87.929544	86.054299	85.798261
30	239.090672	248.634253	246.631635	245.976291
190	-41.514453	-31.521500	-20.886600	-20.791891
9	-395.573814	-401.397174	-395.046826	-394.028853
67	31.789654	39.031722	38.286235	38.367161
218	373.228970	407.431108	400.094889	398.936383
175	-336.495524	-362.971016	-365.631178	-364.613181
18	-99.469821	-152.123828	-155.469751	-155.049234
197	-139.163819	-137.783906	-149.326964	-149.080743
66	-107.825197	-118.043561	-112.738986	-112.551307

Según los resultados mostrados, el mejor modelo fué el Modelo 1, ya que tiene el valor de mean squared error más bajo, luego dió mejor resultado el modelo 3, teniendo en consideración el uso de elastic net. Por último el modelo que tenía mas alto el valor de mean squared error es del modelo 2. Posiblemente estos resultados pueden mejorar teniendo una mayor cantidad de datos, ya que la cantidad de datos utilizados en realidad han sido pocos.

References

[1] Aurélien Géron. *Hands-on Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media.

[2] Scikit-Learn. Gridsearchcv. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html.

[3] Scikit-Learn. Repeatedkfold. https://sklearn.org/modules/generated/sklearn.model_selection.RepeatedKFold.html.