SPRINT 8: Tasca Feature Engineering

Descripció Aprèn a gestionar paràmetres amb Python.

NIVELL 1

Exercici 1

Agafa un conjunt de dades de tema esportiu que t'agradi i normalitza els atributs categòrics en dummy. Estandaritza els atributs numèrics amb StandardScaler.

Per a realitzar aquest sprint utilitzo el mateix conjunt de dades que els sprints anteriors, relacionat amb les jugadores del mundial de futbol femení 2019.

In [2]:
```
# Crido a les llibreries que necessito
# Faig entrar l'arxiu CSV gràcies a pandas

import pandas as pd
import numpy as np
import random
import matplotlib.pyplot as plt
import seaborn as sns
import imblearn

women = pd.read_csv("C:\\Users\\Anna\DataScience\\SPRINTS\\SPRINT 5\\Womens Squads.csv", e

display(women)
```

	Squad no.	Country	Pos.	Player	DOB	Age	Caps	Goals	Club
0	1	US	GK	Alyssa Naeher	20-abr-88	31	43.0	0.0	Chicago Red Stars
1	2	US	FW	Mallory Pugh	29-abr-98	21	50.0	15.0	Washington Spirit
2	3	US	MF	Sam Mewis	09-oct-92	26	47.0	9.0	North Carolina Courage
3	4	US	DF	Becky Sauerbrunn	06-jun-85	34	155.0	0.0	Utah Royals
4	5	US	DF	Kelley O'Hara	04-ago-88	30	115.0	2.0	Utah Royals
...	...	...	...	...	...	...	...	...	...
547	19	France	DF	Griedge Mbock Bathy	26-feb-95	24	49.0	4.0	Lyon
548	20	France	FW	Delphine Cascarino	05-feb-97	22	11.0	1.0	Lyon
549	21	France	GK	Pauline Peyraud-Magnin	17-mar-92	27	1.0	0.0	Arsenal
550	22	France	DF	Julie Debever	18-abr-88	31	2.0	0.0	Guingamp
551	23	France	MF	Maéva Clémaron	10-nov-92	26	3.0	0.0	Fleury

552 rows × 9 columns

In [3]:
```
women.count()
```

Out[3]:
```
Squad no.    552
Country      552
```

```
Pos.          552
Player        552
DOB           552
Age           552
Caps          520
Goals         520
Club          552
dtype: int64
```

In [4]:

```
# Com que en l'anterior punt veiem que les columnes "Caps" i "Goals" tenen menys quantita

print(women.isnull())

print("_____________________________________________________________________________")

print(women.count())

print("_____________________________________________________________________________")

print(women.isnull().sum())
```

```
     Squad no.  Country   Pos.  Player    DOB    Age   Caps  Goals   Club
0        False    False  False   False  False  False  False  False  False
1        False    False  False   False  False  False  False  False  False
2        False    False  False   False  False  False  False  False  False
3        False    False  False   False  False  False  False  False  False
4        False    False  False   False  False  False  False  False  False
..         ...      ...    ...     ...    ...    ...    ...    ...    ...
547      False    False  False   False  False  False  False  False  False
548      False    False  False   False  False  False  False  False  False
549      False    False  False   False  False  False  False  False  False
550      False    False  False   False  False  False  False  False  False
551      False    False  False   False  False  False  False  False  False

[552 rows x 9 columns]
_____________________________________________________________________________
Squad no.    552
Country      552
Pos.         552
Player       552
DOB          552
Age          552
Caps         520
Goals        520
Club         552
dtype: int64
_____________________________________________________________________________
Squad no.     0
Country       0
Pos.          0
Player        0
DOB           0
Age           0
Caps         32
Goals        32
Club          0
dtype: int64
```

In [5]:

```
# Eliminem les files que contenen algun valor nul i comprobem que s'han eliminat, de manei

women = women.dropna(subset=["Caps", "Goals"])

women.count()
```

Out[5]:
```
Squad no.     520
Country       520
Pos.          520
Player        520
DOB           520
Age           520
Caps          520
Goals         520
Club          520
dtype: int64
```

In [11]:
```
women.describe()
```

Out[11]:

	Squad no.	Age	Caps	Goals
count	520.000000	520.000000	520.000000	520.000000
mean	11.867308	26.178846	43.661538	7.348077
std	6.609365	3.996715	43.674846	15.541727
min	1.000000	16.000000	0.000000	0.000000
25%	6.000000	23.000000	11.750000	0.000000
50%	12.000000	26.000000	29.500000	1.500000
75%	18.000000	29.000000	62.000000	8.250000
max	23.000000	41.000000	282.000000	181.000000

In [13]:
```
women.dtypes
```

Out[13]:
```
Squad no.      int64
Country       object
Pos.          object
Player        object
DOB           object
Age            int64
Caps         float64
Goals        float64
Club          object
dtype: object
```

CONVERTIR A DUMMY

Anem a normalitzar els atributs categòrics en dummy, això vol dir que convertirem els valors en 0 i 1, creant un nou dataset que concatenarem amb el dataset original per poder-los treballar més fàcilment.

In [15]:
```
# Imprimint els tipus en la línia anterior, sabem quins hem de convertir en dummys, ho fai
dummyCountry = pd.get_dummies(women["Country"])
dummyCountry.head()
```

Out[15]:

	Argentina	Australia	Brazil	Cameroon	Canada	Chile	China PR	England	France	Germany	...	New Zealand	Nigeria
0	0	0	0	0	0	0	0	0	0	0	...	0	0
1	0	0	0	0	0	0	0	0	0	0	...	0	0
2	0	0	0	0	0	0	0	0	0	0	...	0	0
3	0	0	0	0	0	0	0	0	0	0	...	0	0

	Argentina	Australia	Brazil	Cameroon	Canada	Chile	China PR	England	France	Germany	...	New Zealand	Nigeria
4	0	0	0	0	0	0	0	0	0	0	...	0	0

5 rows × 24 columns

In [18]:
```
dummyPosition = pd.get_dummies(women["Pos."])
dummyPosition.head()
```

Out[18]:

	DF	FW	GK	MF
0	0	0	1	0
1	0	1	0	0
2	0	0	0	1
3	1	0	0	0
4	1	0	0	0

In [17]:
```
dummyPlayer = pd.get_dummies(women["Player"])
dummyPlayer.head()
```

Out[17]:

	Abbie McManus	Abby Dahlkemper	Abby Erceg	Adriana Leon	Adriana Sachs	Adrianna Franch	Agustina Barroso	Ainon Phancha	Aitana Bonmatí	Aivi Luik	...	Yessenia López	Ha
0	0	0	0	0	0	0	0	0	0	0	...	0	
1	0	0	0	0	0	0	0	0	0	0	...	0	
2	0	0	0	0	0	0	0	0	0	0	...	0	
3	0	0	0	0	0	0	0	0	0	0	...	0	
4	0	0	0	0	0	0	0	0	0	0	...	0	

5 rows × 520 columns

In [19]:
```
dummyDOB = pd.get_dummies(women["DOB"])
dummyDOB.head()
```

Out[19]:

	01-abr-95	01-abr-97	01-ago-87	01-dic-94	01-dic-95	01-feb-88	01-feb-89	01-jul-90	01-jul-91	01-jul-98	...	30-sep-85	30-sep-94	31-ago-99	31-dic-96	31-ene-90	31-ene-91	31-ene-97	31-jul-87	31-mar-95
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0

5 rows × 493 columns

In [20]:
```
dummyClub = pd.get_dummies(women["Club"])
```

```
dummyClub.head()
```

Out[20]:

	1. FFC Frankfurt	3B da Amazônia [pt]	AC Nagano Parceiro	ADO Den Haag	AWA Yaoundé	Air Force United	Ajax	Albirex Niigata	Amazone FAP	Ambilly [fr]	...	Vittsjö	Vä
0	0	0	0	0	0	0	0	0	0	0	...	0	
1	0	0	0	0	0	0	0	0	0	0	...	0	
2	0	0	0	0	0	0	0	0	0	0	...	0	
3	0	0	0	0	0	0	0	0	0	0	...	0	
4	0	0	0	0	0	0	0	0	0	0	...	0	

5 rows × 180 columns

Veiem que hi ha atributs que s'han convertit en 520 columnes, això passa perquè per exemple, els només de les jugadores són únics. No acabo d'entendre quina finalitat podria tenir convertir aquest atribu en dummy. Per exemple, però, si que podria entendre el de la data de naixament (DOB) o el del club, ja que ens podria ajudar a treure estadísitques i conclusions interessants (per exemple, la majoria d'esportistes professionals són nascuts entre el gener i el juny, això passa perquè eren els grans de la seva edat, destacaven més quan eren petits i per això solen arribar més lluny, per un cúmul d'aptetar-los/animar-los més, per les capacitats que tenien, per la confiança en ells, etc)

Concatenaré els nous dataframes amb l'original, per veure com canvia el format de l'original, sobretot en la forma i tamany.

In [24]:
```
women1 = pd.concat([women, dummyCountry,dummyPosition, dummyPlayer, dummyDOB, dummyClub],
display(women1)
```

	Squad no.	Country	Pos.	Player	DOB	Age	Caps	Goals	Club	Argentina	...	Vittsjö	Växjö	Vålerenga
0	1	US	GK	Alyssa Naeher	20-abr-88	31	43.0	0.0	Chicago Red Stars	0	...	0	0	
1	2	US	FW	Mallory Pugh	29-abr-98	21	50.0	15.0	Washington Spirit	0	...	0	0	
2	3	US	MF	Sam Mewis	09-oct-92	26	47.0	9.0	North Carolina Courage	0	...	0	0	
3	4	US	DF	Becky Sauerbrunn	06-jun-85	34	155.0	0.0	Utah Royals	0	...	0	0	
4	5	US	DF	Kelley O'Hara	04-ago-88	30	115.0	2.0	Utah Royals	0	...	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
547	19	France	DF	Griedge Mbock Bathy	26-feb-95	24	49.0	4.0	Lyon	0	...	0	0	

	Squad no.	Country	Pos.	Player	DOB	Age	Caps	Goals	Club	Argentina	...	Vittsjö	Växjö	Vålerenga
548	20	France	FW	Delphine Cascarino	05-feb-97	22	11.0	1.0	Lyon	0	...	0	0	
549	21	France	GK	Pauline Peyraud-Magnin	17-mar-92	27	1.0	0.0	Arsenal	0	...	0	0	
550	22	France	DF	Julie Debever	18-abr-88	31	2.0	0.0	Guingamp	0	...	0	0	
551	23	France	MF	Maéva Clémaron	10-nov-92	26	3.0	0.0	Fleury	0	...	0	0	

520 rows × 1230 columns

In [23]:
```
print(women.shape)
print("_____________________________________________________________________________")
print(women1.shape)
```

```
(520, 9)
_____________________________________________________________________________
(520, 1230)
```

Hem passat de tenir un dataset de 9 atributs amb 520 files, a tenir un dataset de 1230 atributs i 520 files

ESTANDARITZAR AMB STANDARDSCALER

Ara procedim a estandaritzar els atributs numèrics amb StandardScaler. Tornaré a utilitzar el dataset original per a una millor comprensió dels valors. Una transformació d'escala estàndard és mapar les dades de l'escala original a una escala entre zero i un. Això normalment s'anomena normalització de dades.

In [26]:
```
# Imprimim els tipus de dades per saber amb quines hem de treballar
women.dtypes
```

Out[26]:
```
Squad no.      int64
Country       object
Pos.          object
Player        object
DOB           object
Age            int64
Caps         float64
Goals        float64
Club          object
dtype: object
```

In [30]:
```
# Només tinc 4 atributs numèrics, i els "extrec" del dataset
numerics = women.iloc[: , [0,5,6,7]]
print(numerics)
```

```
     Squad no.  Age    Caps  Goals
0            1   31    43.0    0.0
1            2   21    50.0   15.0
2            3   26    47.0    9.0
3            4   34   155.0    0.0
```

```
4            5   30  115.0    2.0
..         ...  ...    ...    ...
547         19   24   49.0    4.0
548         20   22   11.0    1.0
549         21   27    1.0    0.0
550         22   31    2.0    0.0
551         23   26    3.0    0.0

[520 rows x 4 columns]
```

In [29]:
```
# Importo la llibreria StandardScaler
from sklearn.preprocessing import StandardScaler
```

In [31]:
```
# Calculo la mitjana de les columnes

numerics.mean()
```

Out[31]:
```
Squad no.    11.867308
Age          26.178846
Caps         43.661538
Goals         7.348077
dtype: float64
```

In [32]:
```
# Calculo la desviació estandar de les columnes

numerics.std()
```

Out[32]:
```
Squad no.     6.609365
Age           3.996715
Caps         43.674846
Goals        15.541727
dtype: float64
```

In [33]:
```
# Creo un objecte anomenat scaler que contingui els atributs numèrics del meu dataset per

scaler = StandardScaler().fit(numerics)
print(scaler)
```

```
StandardScaler()
```

In [35]:
```
# Calculo la mitjana de les columnes amb scaler per veure si em surt el mateix
scaler.mean_
```

Out[35]:
```
array([11.86730769, 26.17884615, 43.66153846,  7.34807692])
```

In [37]:
```
# Calculo la desviació estandar de les columnes amb scaler per veure si em surt el mateix
scaler.scale_
```

Out[37]:
```
array([ 6.60300692,  3.9928704 , 43.63283059, 15.52677576])
```

In [42]:
```
# Anem a estandaritzar els valors

numericScaled = scaler.transform(numerics)
print (numericScaled)
```

```
[[-1.64581195  1.2074406  -0.01516148 -0.47325195]
 [-1.4943658  -1.29702335  0.14526817  0.49282112]
 [-1.34291964 -0.04479137  0.07651261  0.10639189]
```

```
...
[ 1.38311112  0.20565502 -0.97773942 -0.47325195]
[ 1.53455727  1.2074406  -0.9548209  -0.47325195]
[ 1.68600343 -0.04479137 -0.93190238 -0.47325195]]
```

In [46]:
```
# Calculo la mitjana de les columnes escalades
print (numericScaled.mean(axis=0)) #marco l'axis=0 perquè se m'imprimeixin totes les colur
```

```
[-2.04964251e-17 -4.09928501e-16 -4.09928501e-17  8.54017711e-18]
```

In [48]:
```
# Calculo la desviació estandard de les columnes escalades
print (numericScaled.std(axis=0)) #marco l'axis=0 perquè se m'imprimeixin totes les columr
```

```
[1. 1. 1. 1.]
```

Veiem doncs, com hem convertit les mitjanes en un número 0 (el número negatiu final ens indica els 0 que té al davant) i les desviacions típiques ara són 1.

NIVELL 2

Exercici 2

Continua amb el conjunt de dades de tema esportiu que t'agradi i aplica l'anàlisi de components principals.

In [ ]:

In [ ]:

NIVELL 3

Exercici 3

Continua amb el conjunt de dades de tema esportiu que t'agradi i normalitza les dades tenint en compte els outliers.

In [ ]:

In [ ]: