preprocessing

January 6, 2021

1 Preparar datos

[1]:
```
import pandas as pd
import re

import numpy as np

from dataloader import DataLoader
```

1.1 Datos no nulos

Como existen suficientes datos (existe al menos un dato para cada comunidad) se optó por eliminar las filas nulas. La eliminación se realizará en el momento de selección en la clase *DataLoader*, con el fin de mantener la máxima cantidad de datos.

[2]:
```
df = pd.read_csv('energy-usage-2010.csv')
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 67051 entries, 0 to 67050
Data columns (total 73 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   COMMUNITY AREA NAME                   67051 non-null  object
 1   CENSUS BLOCK                          66974 non-null  float64
 2   BUILDING TYPE                         66974 non-null  object
 3   BUILDING_SUBTYPE                      66974 non-null  object
 4   KWH JANUARY 2010                      66180 non-null  float64
 5   KWH FEBRUARY 2010                     66180 non-null  float64
 6   KWH MARCH 2010                        66180 non-null  float64
 7   KWH APRIL 2010                        66180 non-null  float64
 8   KWH MAY 2010                          66180 non-null  float64
 9   KWH JUNE 2010                         66180 non-null  float64
 10  KWH JULY 2010                         66180 non-null  float64
 11  KWH AUGUST 2010                       66180 non-null  float64
 12  KWH SEPTEMBER 2010                    66180 non-null  float64
 13  KWH OCTOBER 2010                      66180 non-null  float64
 14  KWH NOVEMBER 2010                     66180 non-null  float64
```

```
15  KWH DECEMBER 2010                     66180 non-null  float64
16  TOTAL KWH                             66180 non-null  float64
17  ELECTRICITY ACCOUNTS                  66180 non-null  object
18  ZERO KWH ACCOUNTS                     67051 non-null  int64
19  THERM JANUARY 2010                    64821 non-null  float64
20  THERM FEBRUARY 2010                   62819 non-null  float64
21  THERM MARCH 2010                      65569 non-null  float64
22  THERM APRIL 2010                      65476 non-null  float64
23  THERM MAY 2010                        65194 non-null  float64
24  THERM JUNE 2010                       65284 non-null  float64
25  THERM JULY 2010                       65231 non-null  float64
26  THERM AUGUST 2010                     65143 non-null  float64
27  THERM SEPTEMBER 2010                  64769 non-null  float64
28  THERM OCTOBER 2010                    65329 non-null  float64
29  THERM NOVEMBER 2010                   65492 non-null  float64
30  THERM DECEMBER 2010                   65507 non-null  float64
31  TOTAL THERMS                          65755 non-null  float64
32  GAS ACCOUNTS                          65755 non-null  object
33  KWH TOTAL SQFT                        65901 non-null  float64
34  THERMS TOTAL SQFT                     65378 non-null  float64
35  KWH MEAN 2010                         66180 non-null  float64
36  KWH STANDARD DEVIATION 2010           57095 non-null  float64
37  KWH MINIMUM 2010                      66180 non-null  float64
38  KWH 1ST QUARTILE 2010                 66180 non-null  float64
39  KWH 2ND QUARTILE 2010                 66180 non-null  float64
40  KWH 3RD QUARTILE 2010                 66180 non-null  float64
41  KWH MAXIMUM 2010                      66180 non-null  float64
42  KWH SQFT MEAN 2010                    65901 non-null  float64
43  KWH SQFT STANDARD DEVIATION 2010      51666 non-null  float64
44  KWH SQFT MINIMUM 2010                 65901 non-null  float64
45  KWH SQFT 1ST QUARTILE 2010            65901 non-null  float64
46  KWH SQFT 2ND QUARTILE 2010            65901 non-null  float64
47  KWH SQFT 3RD QUARTILE 2010            65901 non-null  float64
48  KWH SQFT MAXIMUM 2010                 65901 non-null  float64
49  THERM MEAN 2010                       65755 non-null  float64
50  THERM STANDARD DEVIATION 2010         56821 non-null  float64
51  THERM MINIMUM 2010                    65755 non-null  float64
52  THERM 1ST QUARTILE 2010               65755 non-null  float64
53  THERM 2ND QUARTILE 2010               65755 non-null  float64
54  THERM 3RD QUARTILE 2010               65755 non-null  float64
55  THERM MAXIMUM 2010                    65755 non-null  float64
56  THERMS SQFT MEAN 2010                 65378 non-null  float64
57  THERMS SQFT STANDARD DEVIATION 2010   51367 non-null  float64
58  THERMS SQFT MINIMUM 2010              65378 non-null  float64
59  THERMS SQFT 1ST QUARTILE 2010         65378 non-null  float64
60  THERMS SQFT 2ND QUARTILE 2010         65378 non-null  float64
61  THERMS SQFT 3RD QUARTILE 2010         65378 non-null  float64
62  THERMS SQFT MAXIMUM 2010              65378 non-null  float64
```

```
 63  TOTAL POPULATION                      67037 non-null  float64
 64  TOTAL UNITS                           67037 non-null  float64
 65  AVERAGE STORIES                       67051 non-null  float64
 66  AVERAGE BUILDING AGE                  67051 non-null  float64
 67  AVERAGE HOUSESIZE                     67037 non-null  float64
 68  OCCUPIED UNITS                        67037 non-null  float64
 69  OCCUPIED UNITS PERCENTAGE             64606 non-null  float64
 70  RENTER-OCCUPIED HOUSING UNITS         67037 non-null  float64
 71  RENTER-OCCUPIED HOUSING PERCENTAGE    64433 non-null  float64
 72  OCCUPIED HOUSING UNITS                67037 non-null  float64
dtypes: float64(67), int64(1), object(5)
memory usage: 37.3+ MB
```

[3]:
```
df.describe()
```

[3]:

	CENSUS BLOCK	KWH JANUARY 2010	KWH FEBRUARY 2010	KWH MARCH 2010	\
count	6.697400e+04	6.618000e+04	6.618000e+04	6.618000e+04	
mean	1.703140e+14	1.758159e+04	1.737651e+04	1.624212e+04	
std	2.776392e+09	3.482508e+05	3.351910e+05	3.164713e+05	
min	1.703101e+14	0.000000e+00	0.000000e+00	0.000000e+00	
25%	1.703116e+14	1.370000e+03	1.613000e+03	1.586000e+03	
50%	1.703133e+14	3.481500e+03	3.814000e+03	3.681500e+03	
75%	1.703167e+14	7.157000e+03	7.410250e+03	7.059000e+03	
max	1.703198e+14	5.298534e+07	4.787976e+07	4.413646e+07	

	KWH APRIL 2010	KWH MAY 2010	KWH JUNE 2010	KWH JULY 2010	\
count	6.618000e+04	6.618000e+04	6.618000e+04	6.618000e+04	
mean	1.595696e+04	1.906623e+04	2.300485e+04	2.482891e+04	
std	3.118232e+05	3.634429e+05	3.988582e+05	4.135955e+05	
min	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	
25%	1.579000e+03	1.957000e+03	2.700000e+03	3.203000e+03	
50%	3.646000e+03	4.528000e+03	6.295000e+03	7.389000e+03	
75%	7.010000e+03	8.943500e+03	1.282850e+04	1.465750e+04	
max	4.222055e+07	4.861925e+07	5.256908e+07	5.513983e+07	

	KWH AUGUST 2010	KWH SEPTEMBER 2010	…	TOTAL POPULATION	\
count	6.618000e+04	6.618000e+04	…	67037.000000	
mean	2.267526e+04	1.856410e+04	…	105.180169	
std	3.940989e+05	3.486212e+05	…	801.339175	
min	0.000000e+00	0.000000e+00	…	0.000000	
25%	2.837750e+03	2.027000e+03	…	37.000000	
50%	6.413500e+03	4.573000e+03	…	64.000000	
75%	1.229800e+04	8.634000e+03	…	105.000000	
max	5.158912e+07	4.450386e+07	…	67388.000000	

	TOTAL UNITS	AVERAGE STORIES	AVERAGE BUILDING AGE	AVERAGE HOUSESIZE	\
count	67037.000000	67051.000000	67051.000000	67037.000000	

```
mean      48.375897         1.887592             71.593006          3.454721
std      426.941305         1.957215             34.168384         26.146208
min        0.000000         1.000000              0.000000          0.000000
25%       15.000000         1.140000             53.000000          2.150000
50%       25.000000         1.750000             80.000000          2.700000
75%       42.000000         2.000000             96.500000          3.320000
max    51372.000000       110.000000            158.000000       2061.920000

       OCCUPIED UNITS  OCCUPIED UNITS PERCENTAGE  \
count    67037.000000                64606.000000
mean        42.347495                    0.880365
std        371.024953                    0.130937
min          0.000000                    0.000000
25%         13.000000                    0.833200
50%         22.000000                    0.914600
75%         37.000000                    0.967700
max      43222.000000                    1.000000

       RENTER-OCCUPIED HOUSING UNITS  RENTER-OCCUPIED HOUSING PERCENTAGE  \
count                    67037.000000                         64433.000000
mean                        25.438952                             0.511679
std                        251.193571                             0.288431
min                          0.000000                             0.000000
25%                          3.000000                             0.286000
50%                         11.000000                             0.537900
75%                         23.000000                             0.733000
max                      28335.000000                             1.000000

       OCCUPIED HOUSING UNITS
count            67037.000000
mean                42.347495
std                371.024953
min                  0.000000
25%                 13.000000
50%                 22.000000
75%                 37.000000
max              43222.000000

[8 rows x 68 columns]
```

[4]:
```
missing_energy = df[DataLoader(df).energy_cols].isna().sum()
missing_energy_per = missing_energy * 100 / len(df)
missing_energy_per, missing_energy_per.mean()
```

[4]:
```
(KWH JANUARY 2010      1.299011
 KWH FEBRUARY 2010     1.299011
 KWH MARCH 2010        1.299011
```

```
KWH APRIL 2010         1.299011
KWH MAY 2010           1.299011
KWH JUNE 2010          1.299011
KWH JULY 2010          1.299011
KWH AUGUST 2010        1.299011
KWH SEPTEMBER 2010     1.299011
KWH OCTOBER 2010       1.299011
KWH NOVEMBER 2010      1.299011
KWH DECEMBER 2010      1.299011
dtype: float64,
1.2990112004295238)
```

[5]:
```
missing_gas = df[DataLoader(df).gas_cols].isna().sum()
missing_gas_per = missing_gas * 100 / len(df)
missing_gas_per, missing_gas_per.mean()
```

[5]:
```
(THERM JANUARY 2010      3.325827
 THERM FEBRUARY 2010     6.311614
 THERM MARCH 2010        2.210258
 THERM APRIL 2010        2.348958
 THERM MAY 2010          2.769534
 THERM JUNE 2010         2.635307
 THERM JULY 2010         2.714352
 THERM AUGUST 2010       2.845595
 THERM SEPTEMBER 2010    3.403380
 THERM OCTOBER 2010      2.568194
 THERM NOVEMBER 2010     2.325096
 THERM DECEMBER 2010     2.302725
 dtype: float64,
 2.9800698970435437)
```

1.2 Sustituir categóricos

Existen ciertas columnas que mezclan valores numéricos y categóricos, como *ELECTRICITY ACCOUNTS* y *GAS ACCOUNTS*, donde se muestra el número de contadores de energía o gas de cada zona. Se pasarán los valores categóricos a numéricos, fijando arbitrariamente un valor único para estas filas.

Sustituimos todos los valores con *Less than 4* en la columna *ELECTRICITY ACCOUNTS* por 3.

[6]:
```
acc = df['ELECTRICITY ACCOUNTS'].unique()
res_match = [re.match(r"\D.*", e) for e in acc if e is not np.nan]
res_match = [e for e in res_match if e is not None]
res_match
```

[6]:
```
[<re.Match object; span=(0, 11), match='Less than 4'>]
```

[7]:
```
df['ELECTRICITY ACCOUNTS'] = df['ELECTRICITY ACCOUNTS'].str.replace('Less than␣
↪4', '3')
```

[8]:
```
df['ELECTRICITY ACCOUNTS'] = pd.to_numeric(df['ELECTRICITY ACCOUNTS'])
```

[9]:
```
df['ELECTRICITY ACCOUNTS']
```

[9]:
```
0         NaN
1         8.0
2         NaN
3         NaN
4         NaN
         ...
67046     6.0
67047     9.0
67048     7.0
67049     7.0
67050    12.0
Name: ELECTRICITY ACCOUNTS, Length: 67051, dtype: float64
```

Sustituimos todos los valores con *Less than 4* en la columna *GAS ACCOUNTS*

[10]:
```
acc = df['GAS ACCOUNTS'].unique()
res_match = [re.match(r"\D.*", e) for e in acc if e is not np.nan]
res_match = [e for e in res_match if e is not None]
res_match
```

[10]:
```
[<re.Match object; span=(0, 11), match='Less than 4'>]
```

[11]:
```
df['GAS ACCOUNTS'] = df['GAS ACCOUNTS'].str.replace('Less than 4', '3')
```

[12]:
```
df['GAS ACCOUNTS'] = pd.to_numeric(df['GAS ACCOUNTS'])
```

[13]:
```
df['GAS ACCOUNTS']
```

[13]:
```
0        11.0
1         NaN
2         4.0
3         3.0
4         3.0
         ...
67046     9.0
67047     8.0
67048     5.0
67049     5.0
67050    13.0
Name: GAS ACCOUNTS, Length: 67051, dtype: float64
```

Ya se tienen todos los datos con los tipos de datos acorde a la información que aportan. A continuación salvamos el dataset en un fichero csv nuevo.

[14]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 67051 entries, 0 to 67050
Data columns (total 73 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   COMMUNITY AREA NAME                   67051 non-null  object
 1   CENSUS BLOCK                          66974 non-null  float64
 2   BUILDING TYPE                         66974 non-null  object
 3   BUILDING_SUBTYPE                      66974 non-null  object
 4   KWH JANUARY 2010                      66180 non-null  float64
 5   KWH FEBRUARY 2010                     66180 non-null  float64
 6   KWH MARCH 2010                        66180 non-null  float64
 7   KWH APRIL 2010                        66180 non-null  float64
 8   KWH MAY 2010                          66180 non-null  float64
 9   KWH JUNE 2010                         66180 non-null  float64
 10  KWH JULY 2010                         66180 non-null  float64
 11  KWH AUGUST 2010                       66180 non-null  float64
 12  KWH SEPTEMBER 2010                    66180 non-null  float64
 13  KWH OCTOBER 2010                      66180 non-null  float64
 14  KWH NOVEMBER 2010                     66180 non-null  float64
 15  KWH DECEMBER 2010                     66180 non-null  float64
 16  TOTAL KWH                             66180 non-null  float64
 17  ELECTRICITY ACCOUNTS                  66180 non-null  float64
 18  ZERO KWH ACCOUNTS                     67051 non-null  int64
 19  THERM JANUARY 2010                    64821 non-null  float64
 20  THERM FEBRUARY 2010                   62819 non-null  float64
 21  THERM MARCH 2010                      65569 non-null  float64
 22  THERM APRIL 2010                      65476 non-null  float64
 23  THERM MAY 2010                        65194 non-null  float64
 24  THERM JUNE 2010                       65284 non-null  float64
 25  THERM JULY 2010                       65231 non-null  float64
 26  THERM AUGUST 2010                     65143 non-null  float64
 27  THERM SEPTEMBER 2010                  64769 non-null  float64
 28  THERM OCTOBER 2010                    65329 non-null  float64
 29  THERM NOVEMBER 2010                   65492 non-null  float64
 30  THERM DECEMBER 2010                   65507 non-null  float64
 31  TOTAL THERMS                          65755 non-null  float64
 32  GAS ACCOUNTS                          65755 non-null  float64
 33  KWH TOTAL SQFT                        65901 non-null  float64
 34  THERMS TOTAL SQFT                     65378 non-null  float64
 35  KWH MEAN 2010                         66180 non-null  float64
 36  KWH STANDARD DEVIATION 2010           57095 non-null  float64
 37  KWH MINIMUM 2010                      66180 non-null  float64
```

```
 38  KWH 1ST QUARTILE 2010                66180 non-null  float64
 39  KWH 2ND QUARTILE 2010                66180 non-null  float64
 40  KWH 3RD QUARTILE 2010                66180 non-null  float64
 41  KWH MAXIMUM 2010                     66180 non-null  float64
 42  KWH SQFT MEAN 2010                   65901 non-null  float64
 43  KWH SQFT STANDARD DEVIATION 2010     51666 non-null  float64
 44  KWH SQFT MINIMUM 2010                65901 non-null  float64
 45  KWH SQFT 1ST QUARTILE 2010           65901 non-null  float64
 46  KWH SQFT 2ND QUARTILE 2010           65901 non-null  float64
 47  KWH SQFT 3RD QUARTILE 2010           65901 non-null  float64
 48  KWH SQFT MAXIMUM 2010                65901 non-null  float64
 49  THERM MEAN 2010                      65755 non-null  float64
 50  THERM STANDARD DEVIATION 2010        56821 non-null  float64
 51  THERM MINIMUM 2010                   65755 non-null  float64
 52  THERM 1ST QUARTILE 2010              65755 non-null  float64
 53  THERM 2ND QUARTILE 2010              65755 non-null  float64
 54  THERM 3RD QUARTILE 2010              65755 non-null  float64
 55  THERM MAXIMUM 2010                   65755 non-null  float64
 56  THERMS SQFT MEAN 2010                65378 non-null  float64
 57  THERMS SQFT STANDARD DEVIATION 2010  51367 non-null  float64
 58  THERMS SQFT MINIMUM 2010             65378 non-null  float64
 59  THERMS SQFT 1ST QUARTILE 2010        65378 non-null  float64
 60  THERMS SQFT 2ND QUARTILE 2010        65378 non-null  float64
 61  THERMS SQFT 3RD QUARTILE 2010        65378 non-null  float64
 62  THERMS SQFT MAXIMUM 2010             65378 non-null  float64
 63  TOTAL POPULATION                     67037 non-null  float64
 64  TOTAL UNITS                          67037 non-null  float64
 65  AVERAGE STORIES                      67051 non-null  float64
 66  AVERAGE BUILDING AGE                 67051 non-null  float64
 67  AVERAGE HOUSESIZE                    67037 non-null  float64
 68  OCCUPIED UNITS                       67037 non-null  float64
 69  OCCUPIED UNITS PERCENTAGE            64606 non-null  float64
 70  RENTER-OCCUPIED HOUSING UNITS        67037 non-null  float64
 71  RENTER-OCCUPIED HOUSING PERCENTAGE   64433 non-null  float64
 72  OCCUPIED HOUSING UNITS               67037 non-null  float64
dtypes: float64(69), int64(1), object(3)
memory usage: 37.3+ MB
```

[15]:
```
df.to_csv("energy-usage-2010-clean.csv", index=False)
```