extract_data1

May 3, 2025

```python
[3]: import pandas as pd
     import numpy as np
```

```python
[5]: path_file1 = '/home/darkcover/Documentos/Gan/Data/UJIndoorLoc/trainingData.csv'
     data1 = pd.read_csv(path_file1)

     data1.head()
```

```
[5]:    WAP001  WAP002  WAP003  WAP004  WAP005  WAP006  WAP007  WAP008  WAP009  \
     0     100     100     100     100     100     100     100     100     100
     1     100     100     100     100     100     100     100     100     100
     2     100     100     100     100     100     100     100     -97     100
     3     100     100     100     100     100     100     100     100     100
     4     100     100     100     100     100     100     100     100     100

        WAP010  …  WAP520  LONGITUDE      LATITUDE  FLOOR  BUILDINGID  SPACEID  \
     0     100  …     100 -7541.2643  4.864921e+06      2           1      106
     1     100  …     100 -7536.6212  4.864934e+06      2           1      106
     2     100  …     100 -7519.1524  4.864950e+06      2           1      103
     3     100  …     100 -7524.5704  4.864934e+06      2           1      102
     4     100  …     100 -7632.1436  4.864982e+06      0           0      122

        RELATIVEPOSITION  USERID  PHONEID   TIMESTAMP
     0                 2       2       23  1371713733
     1                 2       2       23  1371713691
     2                 2       2       23  1371714095
     3                 2       2       23  1371713807
     4                 2      11       13  1369909710

     [5 rows x 529 columns]
```

```python
[6]: data1.describe()
```

```
[6]:              WAP001        WAP002   WAP003   WAP004        WAP005  \
     count  19937.000000  19937.000000  19937.0  19937.0  19937.000000
     mean      99.823644     99.820936    100.0    100.0     99.613733
     std        5.866842      5.798156      0.0      0.0      8.615657
     min      -97.000000    -90.000000    100.0    100.0    -97.000000
```

	WAP001	WAP002	WAP003	WAP004	WAP005
25%	100.000000	100.000000	100.0	100.0	100.000000
50%	100.000000	100.000000	100.0	100.0	100.000000
75%	100.000000	100.000000	100.0	100.0	100.000000
max	100.000000	100.000000	100.0	100.0	100.000000

	WAP006	WAP007	WAP008	WAP009	WAP010 \
count	19937.000000	19937.000000	19937.000000	19937.000000	19937.000000
mean	97.130461	94.733661	93.820234	94.693936	99.163766
std	22.931890	30.541335	33.010404	30.305084	12.634045
min	-98.000000	-99.000000	-98.000000	-98.000000	-99.000000
25%	100.000000	100.000000	100.000000	100.000000	100.000000
50%	100.000000	100.000000	100.000000	100.000000	100.000000
75%	100.000000	100.000000	100.000000	100.000000	100.000000
max	100.000000	100.000000	100.000000	100.000000	100.000000

	…	WAP520	LONGITUDE	LATITUDE	FLOOR	BUILDINGID \
count	…	19937.0	19937.000000	1.993700e+04	19937.000000	19937.000000
mean	…	100.0	-7464.275947	4.864871e+06	1.674575	1.212820
std	…	0.0	123.402010	6.693318e+01	1.223078	0.833139
min	…	100.0	-7691.338400	4.864746e+06	0.000000	0.000000
25%	…	100.0	-7594.737000	4.864821e+06	1.000000	0.000000
50%	…	100.0	-7423.060900	4.864852e+06	2.000000	1.000000
75%	…	100.0	-7359.193000	4.864930e+06	3.000000	2.000000
max	…	100.0	-7300.818990	4.865017e+06	4.000000	2.000000

	SPACEID	RELATIVEPOSITION	USERID	PHONEID \
count	19937.000000	19937.000000	19937.000000	19937.000000
mean	148.429954	1.833024	9.068014	13.021869
std	58.342106	0.372964	4.988720	5.362410
min	1.000000	1.000000	1.000000	1.000000
25%	110.000000	2.000000	5.000000	8.000000
50%	129.000000	2.000000	11.000000	13.000000
75%	207.000000	2.000000	13.000000	14.000000
max	254.000000	2.000000	18.000000	24.000000

	TIMESTAMP
count	1.993700e+04
mean	1.371421e+09
std	5.572054e+05
min	1.369909e+09
25%	1.371056e+09
50%	1.371716e+09
75%	1.371721e+09
max	1.371738e+09

[8 rows x 529 columns]

```
[7]: data1.columns
```

```
[7]: Index(['WAP001', 'WAP002', 'WAP003', 'WAP004', 'WAP005', 'WAP006', 'WAP007',
            'WAP008', 'WAP009', 'WAP010',
            ...
            'WAP520', 'LONGITUDE', 'LATITUDE', 'FLOOR', 'BUILDINGID', 'SPACEID',
            'RELATIVEPOSITION', 'USERID', 'PHONEID', 'TIMESTAMP'],
           dtype='object', length=529)
```

```
[8]: data2 = data1[['LONGITUDE', 'LATITUDE', 'FLOOR', 'BUILDINGID']]
     data2.head()
```

```
[8]:    LONGITUDE      LATITUDE  FLOOR  BUILDINGID
     0 -7541.2643  4.864921e+06      2           1
     1 -7536.6212  4.864934e+06      2           1
     2 -7519.1524  4.864950e+06      2           1
     3 -7524.5704  4.864934e+06      2           1
     4 -7632.1436  4.864982e+06      0           0
```

```
[9]: data2.describe()
```

```
[9]:             LONGITUDE      LATITUDE         FLOOR    BUILDINGID
     count  19937.000000  1.993700e+04  19937.000000  19937.000000
     mean   -7464.275947  4.864871e+06      1.674575      1.212820
     std      123.402010  6.693318e+01      1.223078      0.833139
     min    -7691.338400  4.864746e+06      0.000000      0.000000
     25%    -7594.737000  4.864821e+06      1.000000      0.000000
     50%    -7423.060900  4.864852e+06      2.000000      1.000000
     75%    -7359.193000  4.864930e+06      3.000000      2.000000
     max    -7300.818990  4.865017e+06      4.000000      2.000000
```

```
[ ]: # Carregar base
     df = pd.read_csv(path_file1)

     # Substituir valor ausente 100 por -110
     rssi_columns = df.columns[:520]
     df[rssi_columns] = df[rssi_columns].replace(100, -110)

     # Filtro para prédio 1, andar 2
     df_filtered = df[(df['BUILDINGID'] == 1) & (df['FLOOR'] == 2)].copy()

     # Selecionar RSSI + coordenadas
     X_rssi = df_filtered[rssi_columns].values
     y_coords = df_filtered[['LONGITUDE', 'LATITUDE']].values
```