vclsi-01-mueller

April 10, 2019

In [2]:
```
import pandas as pd
import numpy as np
```

In [3]:
```
data = pd.read_excel("breast-cancer-wisconsin.xlsx")
data.head()
```

Out[3]:

	code	thickness	uniCelS	uniCelShape	marAdh	epiCelSize	bareNuc	\
0	1000025	5	1	1	1	2	1.0	
1	1002945	5	4	4	5	7	10.0	
2	1015425	3	1	1	1	2	2.0	
3	1016277	6	8	8	1	3	4.0	
4	1017023	4	1	1	3	2	1.0	

	blaChroma	normNuc	mitoses	class
0	3	1	1	2
1	3	2	1	2
2	3	1	1	2
3	3	7	1	2
4	3	1	1	2

In [4]:
```
#data.loc[:,"thickness"] #Get all values in column thickness
#data.loc[:,["thickness", "mitoses"]] #Get all values in column thickness, mitoses
#data.loc[2:5, :] #row 2-5, all columns
#data.loc[:,"class"]==2 #index of all rows where class equals 2
#data.loc[data.loc[:,"class"]==2, ["thickness", "mitoses"]] #column thickness, mitoses o
#data.iloc[:, [0, -2, -1]] #all rows showing colums with last index and second last
#data.iloc[:5,[0, -2, -1]] #first 5 rows showing colums with last index and second last
#data[np.logical_and(data.thickness > 3, data.epiCelSize<7)] #all rows + columns where t
#data.loc[:, ["mitoses", "thickness"]].iloc[:100, :] #first 100 rows with columns "mitso
```

In [5]:
```
#data.class #class is a reseved keyword
```

In [6]:
```
data.isnull().sum()
```

Out[6]:
```
code           0
thickness      0
uniCelS        0
uniCelShape    0
```

```
marAdh          0
epiCelSize      0
bareNuc        16
blaChroma       0
normNuc         0
mitoses         0
class           0
dtype: int64
```

In [7]:
```
data.isnull().sum().sum()
```

Out[7]:
```
16
```

In [8]:
```
data[data.isnull().any(axis=1)] #axis = 1 -> rows, axis = 0 -> columns
```

Out[8]:

	code	thickness	uniCelS	uniCelShape	marAdh	epiCelSize	bareNuc	blaChroma	normNuc	mitoses	class
23	1057013	8	4	5	1	2	NaN	7	3	1	4
40	1096800	6	6	6	9	6	NaN	7	8	1	2
139	1183246	1	1	1	1	1	NaN	2	1	1	2
145	1184840	1	1	3	1	2	NaN	2	1	1	2
158	1193683	1	1	2	1	3	NaN	1	1	1	2
164	1197510	5	1	1	1	2	NaN	3	1	1	2
235	1241232	3	1	4	1	2	NaN	3	1	1	2
249	169356	3	1	1	1	2	NaN	3	1	1	2
275	432809	3	1	3	1	2	NaN	2	1	1	2
292	563649	8	8	8	1	2	NaN	6	10	1	4
294	606140	1	1	1	1	2	NaN	2	1	1	2
297	61634	5	4	3	1	2	NaN	2	3	1	2
315	704168	4	6	5	6	7	NaN	4	9	1	2
321	733639	3	1	1	1	2	NaN	3	1	1	2
411	1238464	1	1	1	1	1	NaN	2	1	1	2
617	1057067	1	1	1	1	1	NaN	2	1	1	2

```
        411          2        1        1      2
        617          1        1        1      2
```

In [9]:
```
data.drop("code", axis=1)
```

Out[9]:

	thickness	uniCelS	uniCelShape	marAdh	epiCelSize	bareNuc	blaChroma	\
0	5	1	1	1	2	1.0	3	
1	5	4	4	5	7	10.0	3	
2	3	1	1	1	2	2.0	3	
3	6	8	8	1	3	4.0	3	
4	4	1	1	3	2	1.0	3	
5	8	10	10	8	7	10.0	9	
6	1	1	1	1	2	10.0	3	
7	2	1	2	1	2	1.0	3	
8	2	1	1	1	2	1.0	1	
9	4	2	1	1	2	1.0	2	
10	1	1	1	1	1	1.0	3	
11	2	1	1	1	2	1.0	2	
12	5	3	3	3	2	3.0	4	
13	1	1	1	1	2	3.0	3	
14	8	7	5	10	7	9.0	5	
15	7	4	6	4	6	1.0	4	
16	4	1	1	1	2	1.0	2	
17	4	1	1	1	2	1.0	3	
18	10	7	7	6	4	10.0	4	
19	6	1	1	1	2	1.0	3	
20	7	3	2	10	5	10.0	5	
21	10	5	5	3	6	7.0	7	
22	3	1	1	1	2	1.0	2	
23	8	4	5	1	2	NaN	7	
24	1	1	1	1	2	1.0	3	
25	5	2	3	4	2	7.0	3	
26	3	2	1	1	1	1.0	2	
27	5	1	1	1	2	1.0	2	
28	2	1	1	1	2	1.0	2	
29	1	1	3	1	2	1.0	1	
..	...	...	...	...	...	...	...	
669	5	10	10	8	5	5.0	7	
670	3	10	7	8	5	8.0	7	
671	3	2	1	2	2	1.0	3	
672	2	1	1	1	2	1.0	3	
673	5	3	2	1	3	1.0	1	
674	1	1	1	1	2	1.0	2	
675	4	1	4	1	2	1.0	1	
676	1	1	2	1	2	1.0	2	
677	5	1	1	1	2	1.0	1	
678	1	1	1	1	2	1.0	1	
679	2	1	1	1	2	1.0	1	

680	10	10	10	10	5	10.0	10
681	5	10	10	10	4	10.0	5
682	5	1	1	1	2	1.0	3
683	1	1	1	1	2	1.0	1
684	1	1	1	1	2	1.0	1
685	1	1	1	1	2	1.0	1
686	1	1	1	1	2	1.0	1
687	3	1	1	1	2	1.0	2
688	4	1	1	1	2	1.0	1
689	1	1	1	1	2	1.0	1
690	1	1	1	3	2	1.0	1
691	5	10	10	5	4	5.0	4
692	3	1	1	1	2	1.0	1
693	3	1	1	1	2	1.0	2
694	3	1	1	1	3	2.0	1
695	2	1	1	1	2	1.0	1
696	5	10	10	3	7	3.0	8
697	4	8	6	4	3	4.0	10
698	4	8	8	5	4	5.0	10

	normNuc	mitoses	class
0	1	1	2
1	2	1	2
2	1	1	2
3	7	1	2
4	1	1	2
5	7	1	4
6	1	1	2
7	1	1	2
8	1	5	2
9	1	1	2
10	1	1	2
11	1	1	2
12	4	1	4
13	1	1	2
14	5	4	4
15	3	1	4
16	1	1	2
17	1	1	2
18	1	2	4
19	1	1	2
20	4	4	4
21	10	1	4
22	1	1	2
23	3	1	4
24	1	1	2
25	6	1	4
26	1	1	2

```
27          1        1      2
28          1        1      2
29          1        1      2
..        ...      ...    ...
669        10        1      4
670         4        1      4
671         1        1      2
672         1        1      2
673         1        1      2
674         1        1      2
675         1        1      2
676         1        1      2
677         1        1      2
678         1        1      2
679         1        1      2
680        10        7      4
681         6        3      4
682         2        1      2
683         1        1      2
684         1        1      2
685         1        1      2
686         1        1      2
687         3        1      2
688         1        1      2
689         1        8      2
690         1        1      2
691         4        1      4
692         1        1      2
693         1        2      2
694         1        1      2
695         1        1      2
696        10        2      4
697         6        1      4
698         4        1      4

[699 rows x 10 columns]
```

```
In [10]: interpolated_data = data.interpolate(method ='linear', limit_direction ='backward', lim
         interpolated_data.isnull().sum().sum()
```

```
Out[10]: 0
```

```
In [11]: cur_column = "code"
         groups = data.groupby("class")
         results = {}

         for column in data.columns:
             if column == "class":
```

```
            continue
        mean = data[column].mean()
        group_means = groups[column].mean()
        group_variances = groups[column].var()

        F = ((group_means[2] - mean)**2 + (group_means[4] - mean)**2) / (group_variances[2]
        print(f"F score for column {column}: {F}")
        results[column] = F
```

```
F score for column code: 0.009469207736910977
F score for column thickness: 1.1317247225583984
F score for column uniCelS: 1.8363064117529115
F score for column uniCelShape: 1.8986351151927807
F score for column marAdh: 0.8488204387140027
F score for column epiCelSize: 0.8084139123636951
F score for column bareNuc: 1.9368428273239722
F score for column blaChroma: 1.3013895528628854
F score for column normNuc: 0.9282550362104417
F score for column mitoses: 0.18783851825754008
```

In [ ]: