heartdisease

December 26, 2023

1 1) Pengumpulan Data

Data didapat dari https://archive.ics.uci.edu/dataset/45/heart+disease File yang digunakan adalah hungarian.data

2 2) Menelaah Data

[ ]:
```
import pandas as pd
import numpy as np
import re
import itertools
```

Load Data

[ ]:
```
direct = '/content/sample_data/hungarian.data'
```

Membuat iterasi untuk membaca dataset

[ ]:
```
with open(direct, encoding='Latin1') as file:
  lines= [line.strip() for line in file]

lines[0:10]
```

[ ]:
```
['1254 0 40 1 1 0 0',
 '-9 2 140 0 289 -9 -9 -9',
 '0 -9 -9 0 12 16 84 0',
 '0 0 0 0 150 18 -9 7',
 '172 86 200 110 140 86 0 0',
 '0 -9 26 20 -9 -9 -9 -9',
 '-9 -9 -9 -9 -9 -9 -9 12',
 '20 84 0 -9 -9 -9 -9 -9',
 '-9 -9 -9 -9 -9 1 1 1',
 '1 1 -9. -9. name']
```

Membuat keterangan kolom dan baris dari deskripsi dataset sebelumnya

[ ]:
```
data = itertools.takewhile(
    lambda x: len(x) == 76,
```

```
    (' '.join(lines[i:(i+10)]).split() for i in range(0, len(lines), 10))
)

df = pd.DataFrame.from_records(data)

df.head()
```

[ ]:

	0	1	2	3	4	5	6	7	8	9	…	66	67	68	69	70	71	72	73	74	\
0	1254	0	40	1	1	0	0	-9	2	140	…	-9	-9	1	1	1	1	1	-9.	-9.	
1	1255	0	49	0	1	0	0	-9	3	160	…	-9	-9	1	1	1	1	1	-9.	-9.	
2	1256	0	37	1	1	0	0	-9	2	130	…	-9	-9	1	1	1	1	1	-9.	-9.	
3	1257	0	48	0	1	1	1	-9	4	138	…	2	-9	1	1	1	1	1	-9.	-9.	
4	1258	0	54	1	1	0	1	-9	3	150	…	1	-9	1	1	1	1	1	-9.	-9.	

	75
0	name
1	name
2	name
3	name
4	name

[5 rows x 76 columns]

Menampilkan informasi dataset

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 294 entries, 0 to 293
Data columns (total 76 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   0       294 non-null    object
 1   1       294 non-null    object
 2   2       294 non-null    object
 3   3       294 non-null    object
 4   4       294 non-null    object
 5   5       294 non-null    object
 6   6       294 non-null    object
 7   7       294 non-null    object
 8   8       294 non-null    object
 9   9       294 non-null    object
 10  10      294 non-null    object
 11  11      294 non-null    object
 12  12      294 non-null    object
 13  13      294 non-null    object
 14  14      294 non-null    object
 15  15      294 non-null    object
```

```
16  16      294 non-null    object
17  17      294 non-null    object
18  18      294 non-null    object
19  19      294 non-null    object
20  20      294 non-null    object
21  21      294 non-null    object
22  22      294 non-null    object
23  23      294 non-null    object
24  24      294 non-null    object
25  25      294 non-null    object
26  26      294 non-null    object
27  27      294 non-null    object
28  28      294 non-null    object
29  29      294 non-null    object
30  30      294 non-null    object
31  31      294 non-null    object
32  32      294 non-null    object
33  33      294 non-null    object
34  34      294 non-null    object
35  35      294 non-null    object
36  36      294 non-null    object
37  37      294 non-null    object
38  38      294 non-null    object
39  39      294 non-null    object
40  40      294 non-null    object
41  41      294 non-null    object
42  42      294 non-null    object
43  43      294 non-null    object
44  44      294 non-null    object
45  45      294 non-null    object
46  46      294 non-null    object
47  47      294 non-null    object
48  48      294 non-null    object
49  49      294 non-null    object
50  50      294 non-null    object
51  51      294 non-null    object
52  52      294 non-null    object
53  53      294 non-null    object
54  54      294 non-null    object
55  55      294 non-null    object
56  56      294 non-null    object
57  57      294 non-null    object
58  58      294 non-null    object
59  59      294 non-null    object
60  60      294 non-null    object
61  61      294 non-null    object
62  62      294 non-null    object
63  63      294 non-null    object
```

```
 64  64      294 non-null    object
 65  65      294 non-null    object
 66  66      294 non-null    object
 67  67      294 non-null    object
 68  68      294 non-null    object
 69  69      294 non-null    object
 70  70      294 non-null    object
 71  71      294 non-null    object
 72  72      294 non-null    object
 73  73      294 non-null    object
 74  74      294 non-null    object
 75  75      294 non-null    object
dtypes: object(76)
memory usage: 174.7+ KB
```

Menghapus kolom ke-0 atau pertama

[ ]:
```
df = df.iloc[:,:-1]
df = df.drop(df.columns[0], axis =1)
```

Mengubah type data menjadi float

[ ]:
```
df = df.astype(float)
```

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 294 entries, 0 to 293
Data columns (total 74 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   1       294 non-null    float64
 1   2       294 non-null    float64
 2   3       294 non-null    float64
 3   4       294 non-null    float64
 4   5       294 non-null    float64
 5   6       294 non-null    float64
 6   7       294 non-null    float64
 7   8       294 non-null    float64
 8   9       294 non-null    float64
 9   10      294 non-null    float64
 10  11      294 non-null    float64
 11  12      294 non-null    float64
 12  13      294 non-null    float64
 13  14      294 non-null    float64
 14  15      294 non-null    float64
 15  16      294 non-null    float64
 16  17      294 non-null    float64
 17  18      294 non-null    float64
```

```
18  19      294 non-null    float64
19  20      294 non-null    float64
20  21      294 non-null    float64
21  22      294 non-null    float64
22  23      294 non-null    float64
23  24      294 non-null    float64
24  25      294 non-null    float64
25  26      294 non-null    float64
26  27      294 non-null    float64
27  28      294 non-null    float64
28  29      294 non-null    float64
29  30      294 non-null    float64
30  31      294 non-null    float64
31  32      294 non-null    float64
32  33      294 non-null    float64
33  34      294 non-null    float64
34  35      294 non-null    float64
35  36      294 non-null    float64
36  37      294 non-null    float64
37  38      294 non-null    float64
38  39      294 non-null    float64
39  40      294 non-null    float64
40  41      294 non-null    float64
41  42      294 non-null    float64
42  43      294 non-null    float64
43  44      294 non-null    float64
44  45      294 non-null    float64
45  46      294 non-null    float64
46  47      294 non-null    float64
47  48      294 non-null    float64
48  49      294 non-null    float64
49  50      294 non-null    float64
50  51      294 non-null    float64
51  52      294 non-null    float64
52  53      294 non-null    float64
53  54      294 non-null    float64
54  55      294 non-null    float64
55  56      294 non-null    float64
56  57      294 non-null    float64
57  58      294 non-null    float64
58  59      294 non-null    float64
59  60      294 non-null    float64
60  61      294 non-null    float64
61  62      294 non-null    float64
62  63      294 non-null    float64
63  64      294 non-null    float64
64  65      294 non-null    float64
65  66      294 non-null    float64
```

```
 66  67      294 non-null    float64
 67  68      294 non-null    float64
 68  69      294 non-null    float64
 69  70      294 non-null    float64
 70  71      294 non-null    float64
 71  72      294 non-null    float64
 72  73      294 non-null    float64
 73  74      294 non-null    float64
dtypes: float64(74)
memory usage: 170.1 KB
```

3 3) Validasi Data

Mengubah value -9.0 pada setiap baris, menjadi null atau NaN

[ ]:
```
df.replace(-9.0, np.nan, inplace=True)
```

Menghitung jumlah nilai null value

[ ]:
```
df.isnull().sum()
```

[ ]:
```
1       0
2       0
3       0
4       0
5       0
     …
70      0
71      0
72      0
73    266
74    294
Length: 74, dtype: int64
```

[ ]:
```
df.head()
```

[ ]:
```
     1     2    3    4    5    6   7    8      9    10  …  65   66  67   68  \
0  0.0  40.0  1.0  1.0  0.0  0.0 NaN  2.0  140.0  0.0  …  NaN  NaN NaN  1.0
1  0.0  49.0  0.0  1.0  0.0  0.0 NaN  3.0  160.0  1.0  …  NaN  NaN NaN  1.0
2  0.0  37.0  1.0  1.0  0.0  0.0 NaN  2.0  130.0  0.0  …  NaN  NaN NaN  1.0
3  0.0  48.0  0.0  1.0  1.0  1.0 NaN  4.0  138.0  0.0  …  NaN  2.0 NaN  1.0
4  0.0  54.0  1.0  1.0  0.0  1.0 NaN  3.0  150.0  0.0  …  NaN  1.0 NaN  1.0

    69   70   71   72  73  74
0  1.0  1.0  1.0  1.0 NaN NaN
1  1.0  1.0  1.0  1.0 NaN NaN
2  1.0  1.0  1.0  1.0 NaN NaN
3  1.0  1.0  1.0  1.0 NaN NaN
```

```
4  1.0  1.0  1.0  1.0 NaN NaN

[5 rows x 74 columns]
```

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 294 entries, 0 to 293
Data columns (total 74 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   1       294 non-null    float64
 1   2       294 non-null    float64
 2   3       294 non-null    float64
 3   4       294 non-null    float64
 4   5       294 non-null    float64
 5   6       294 non-null    float64
 6   7       0 non-null      float64
 7   8       294 non-null    float64
 8   9       293 non-null    float64
 9   10      293 non-null    float64
 10  11      271 non-null    float64
 11  12      12 non-null     float64
 12  13      1 non-null      float64
 13  14      0 non-null      float64
 14  15      286 non-null    float64
 15  16      21 non-null     float64
 16  17      1 non-null      float64
 17  18      293 non-null    float64
 18  19      294 non-null    float64
 19  20      294 non-null    float64
 20  21      294 non-null    float64
 21  22      293 non-null    float64
 22  23      292 non-null    float64
 23  24      293 non-null    float64
 24  25      293 non-null    float64
 25  26      293 non-null    float64
 26  27      285 non-null    float64
 27  28      292 non-null    float64
 28  29      104 non-null    float64
 29  30      292 non-null    float64
 30  31      293 non-null    float64
 31  32      293 non-null    float64
 32  33      293 non-null    float64
 33  34      293 non-null    float64
 34  35      293 non-null    float64
 35  36      293 non-null    float64
```

```
 36  37      293 non-null    float64
 37  38      292 non-null    float64
 38  39      294 non-null    float64
 39  40      104 non-null    float64
 40  41      293 non-null    float64
 41  42      294 non-null    float64
 42  43      4 non-null      float64
 43  44      0 non-null      float64
 44  45      0 non-null      float64
 45  46      0 non-null      float64
 46  47      3 non-null      float64
 47  48      0 non-null      float64
 48  49      2 non-null      float64
 49  50      28 non-null     float64
 50  51      27 non-null     float64
 51  52      17 non-null     float64
 52  53      0 non-null      float64
 53  54      294 non-null    float64
 54  55      294 non-null    float64
 55  56      294 non-null    float64
 56  57      294 non-null    float64
 57  58      19 non-null     float64
 58  59      58 non-null     float64
 59  60      48 non-null     float64
 60  61      18 non-null     float64
 61  62      59 non-null     float64
 62  63      9 non-null      float64
 63  64      23 non-null     float64
 64  65      5 non-null      float64
 65  66      50 non-null     float64
 66  67      25 non-null     float64
 67  68      294 non-null    float64
 68  69      294 non-null    float64
 69  70      294 non-null    float64
 70  71      294 non-null    float64
 71  72      294 non-null    float64
 72  73      28 non-null     float64
 73  74      0 non-null      float64
dtypes: float64(74)
memory usage: 170.1 KB
```

4 4) Menentukan Object Data

Mengambil 14 fitur, sesuai instruksi dari file heart-disease.names

[ ]:
```
df_selected = df.iloc[:, [1, 2, 7, 8, 10, 14, 17, 30, 36, 38, 39, 42, 49, 56]]
```

[ ]:
```
df_selected.head()
```

[ ]:

	2	3	8	9	11	15	18	31	37	39	40	43	50	57
0	40.0	1.0	2.0	140.0	289.0	0.0	0.0	172.0	0.0	0.0	NaN	NaN	NaN	0.0
1	49.0	0.0	3.0	160.0	180.0	0.0	0.0	156.0	0.0	1.0	2.0	NaN	NaN	1.0
2	37.0	1.0	2.0	130.0	283.0	0.0	1.0	98.0	0.0	0.0	NaN	NaN	NaN	0.0
3	48.0	0.0	4.0	138.0	214.0	0.0	0.0	108.0	1.0	1.5	2.0	NaN	NaN	3.0
4	54.0	1.0	3.0	150.0	NaN	0.0	0.0	122.0	0.0	0.0	NaN	NaN	NaN	0.0

[ ]:
```
df_selected.tail()
```

[ ]:

	2	3	8	9	11	15	18	31	37	39	40	43	50	\
289	48.0	0.0	2.0	NaN	308.0	0.0	1.0	NaN	NaN	2.0	1.0	NaN	NaN	
290	36.0	1.0	2.0	120.0	166.0	0.0	0.0	180.0	0.0	0.0	NaN	NaN	NaN	
291	48.0	1.0	3.0	110.0	211.0	0.0	0.0	138.0	0.0	0.0	NaN	NaN	6.0	
292	47.0	0.0	2.0	140.0	257.0	0.0	0.0	135.0	0.0	1.0	1.0	NaN	NaN	
293	53.0	1.0	4.0	130.0	182.0	0.0	0.0	148.0	0.0	0.0	NaN	NaN	NaN	

	57
289	0.0
290	0.0
291	0.0
292	0.0
293	0.0

[ ]:
```
df_selected.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 294 entries, 0 to 293
Data columns (total 14 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   2       294 non-null    float64
 1   3       294 non-null    float64
 2   8       294 non-null    float64
 3   9       293 non-null    float64
 4   11      271 non-null    float64
 5   15      286 non-null    float64
 6   18      293 non-null    float64
 7   31      293 non-null    float64
 8   37      293 non-null    float64
 9   39      294 non-null    float64
 10  40      104 non-null    float64
 11  43      4 non-null      float64
 12  50      28 non-null     float64
 13  57      294 non-null    float64
dtypes: float64(14)
```

```
memory usage: 32.3 KB
```

Mengganti 14 nama kolom sesuai instruksi

[ ]:
```
column_mapping = {
    2: 'age',
    3: 'sex',
    8: 'cp',
    9: 'trestbps',
    11: 'chol',
    15: 'fbs',
    18: 'restecg',
    31: 'thalach',
    37: 'exang',
    39: 'oldpeak',
    40: 'slope',
    43: 'ca',
    50: 'thal',
    57: 'target'
}
df_selected.rename(columns=column_mapping, inplace=True)
```

```
<ipython-input-103-b484e5bfe3ce>:17: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_selected.rename(columns=column_mapping, inplace=True)
```

[ ]:
```
df_selected.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 294 entries, 0 to 293
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       294 non-null    float64
 1   sex       294 non-null    float64
 2   cp        294 non-null    float64
 3   trestbps  293 non-null    float64
 4   chol      271 non-null    float64
 5   fbs       286 non-null    float64
 6   restecg   293 non-null    float64
 7   thalach   293 non-null    float64
 8   exang     293 non-null    float64
 9   oldpeak   294 non-null    float64
 10  slope     104 non-null    float64
 11  ca        4 non-null      float64
 12  thal      28 non-null     float64
```

```
 13  target    294 non-null    float64
dtypes: float64(14)
memory usage: 32.3 KB
```

Menghitung jumlah fitur pada dataset

[ ]:
```
df_selected.value_counts()
```

[ ]:
```
age   sex  cp   trestbps  chol   fbs  restecg  thalach  exang  oldpeak  slope
ca   thal  target
47.0  1.0  4.0  150.0     226.0  0.0  0.0      98.0     1.0    1.5      2.0
0.0  7.0   1.0       1
dtype: int64
```

5 5) Membersihkan Data

Menghitung nilai null pada setiap kolom

[ ]:
```
df_selected.isnull().sum()
```

[ ]:
```
age          0
sex          0
cp           0
trestbps     1
chol        23
fbs          8
restecg      1
thalach      1
exang        1
oldpeak      0
slope      190
ca         290
thal       266
target       0
dtype: int64
```

Dikarenakan kolom slope, ca, dan thal memiliki banyak nilai null, maka akan didrop atau dihapus

[ ]:
```
columns_to_drop = ['ca', 'slope', 'thal']
df_selected = df_selected.drop(columns_to_drop, axis=1)

df_selected.isnull().sum()
```

[ ]:
```
age          0
sex          0
cp           0
trestbps     1
chol        23
```

```
fbs          8
restecg      1
thalach      1
exang        1
oldpeak      0
target       0
dtype: int64
```

Pengisian nilai null pada beberapa fitur, dengan mencari nilai mean di setiap kolomnya

[ ]:
```
meanTBPS = df_selected['trestbps'].dropna()
meanChol = df_selected['chol'].dropna()
meanfbs = df_selected['fbs'].dropna()
meanRestCG = df_selected['restecg'].dropna()
meanthalach = df_selected['thalach'].dropna()
meanexang = df_selected['exang'].dropna()
```

[ ]:
```
meanTBPS = meanTBPS.astype(float)
meanChol = meanChol.astype(float)
meanfbs = meanfbs.astype(float)
meanthalach = meanthalach.astype(float)
meanexang = meanexang.astype(float)
meanRestCG = meanRestCG.astype(float)
```

[ ]:
```
meanTBPS = round(meanTBPS.mean())
meanChol = round(meanChol.mean())
meanfbs = round(meanfbs.mean())
meanthalach = round(meanthalach.mean())
meanexang = round(meanexang.mean())
meanRestCG = round(meanRestCG.mean())
```

Mengubah nilai null menjadi nilai mean

[ ]:
```
fill_values = {'trestbps': meanTBPS, 'chol': meanChol, 'fbs': meanfbs,
'thalach':meanthalach,'exang':meanexang,'restecg':meanRestCG}
dfClean = df_selected.fillna(value=fill_values)
dfClean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 294 entries, 0 to 293
Data columns (total 11 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       294 non-null    float64
 1   sex       294 non-null    float64
 2   cp        294 non-null    float64
 3   trestbps  294 non-null    float64
 4   chol      294 non-null    float64
```

```
 5   fbs       294 non-null    float64
 6   restecg   294 non-null    float64
 7   thalach   294 non-null    float64
 8   exang     294 non-null    float64
 9   oldpeak   294 non-null    float64
 10  target    294 non-null    float64
dtypes: float64(11)
memory usage: 25.4 KB
```

[ ]:
```
dfClean.isnull().sum()
```

[ ]:
```
age         0
sex         0
cp          0
trestbps    0
chol        0
fbs         0
restecg     0
thalach     0
exang       0
oldpeak     0
target      0
dtype: int64
```

Pengecekan duplikasi data

[ ]:
```
duplicate_rows = dfClean.duplicated()
dfClean[duplicate_rows]
```

[ ]:
```
      age  sex   cp  trestbps   chol  fbs  restecg  thalach  exang  oldpeak  \
163  49.0  0.0  2.0     110.0  251.0  0.0      0.0    160.0    0.0      0.0

     target
163     0.0
```

[ ]:
```
print("All duplicate rows:")
dfClean[dfClean.duplicated(keep=False)]
```

```
All duplicate rows:
```

[ ]:
```
      age  sex   cp  trestbps   chol  fbs  restecg  thalach  exang  oldpeak  \
90   49.0  0.0  2.0     110.0  251.0  0.0      0.0    160.0    0.0      0.0
163  49.0  0.0  2.0     110.0  251.0  0.0      0.0    160.0    0.0      0.0

     target
90      0.0
163     0.0
```

Hapus data yang sama

[ ]:
```
dfClean = dfClean.drop_duplicates()
print("All duplicate rows:")
dfClean[dfClean.duplicated(keep=False)]
```

```
All duplicate rows:
```

[ ]:
```
Empty DataFrame
Columns: [age, sex, cp, trestbps, chol, fbs, restecg, thalach, exang, oldpeak,
target]
Index: []
```

[ ]:
```
dfClean.head()
```

[ ]:
```
    age  sex   cp  trestbps   chol  fbs  restecg  thalach  exang  oldpeak  \
0  40.0  1.0  2.0     140.0  289.0  0.0      0.0    172.0    0.0      0.0
1  49.0  0.0  3.0     160.0  180.0  0.0      0.0    156.0    0.0      1.0
2  37.0  1.0  2.0     130.0  283.0  0.0      1.0     98.0    0.0      0.0
3  48.0  0.0  4.0     138.0  214.0  0.0      0.0    108.0    1.0      1.5
4  54.0  1.0  3.0     150.0  251.0  0.0      0.0    122.0    0.0      0.0

   target
0     0.0
1     1.0
2     0.0
3     3.0
4     0.0
```

[ ]:
```
dfClean['target'].value_counts()
```

[ ]:
```
0.0    187
1.0     37
3.0     28
2.0     26
4.0     15
Name: target, dtype: int64
```

[ ]:
```
import seaborn as sns
import matplotlib.pyplot as plt
```

Mencari korelasi antar fitur

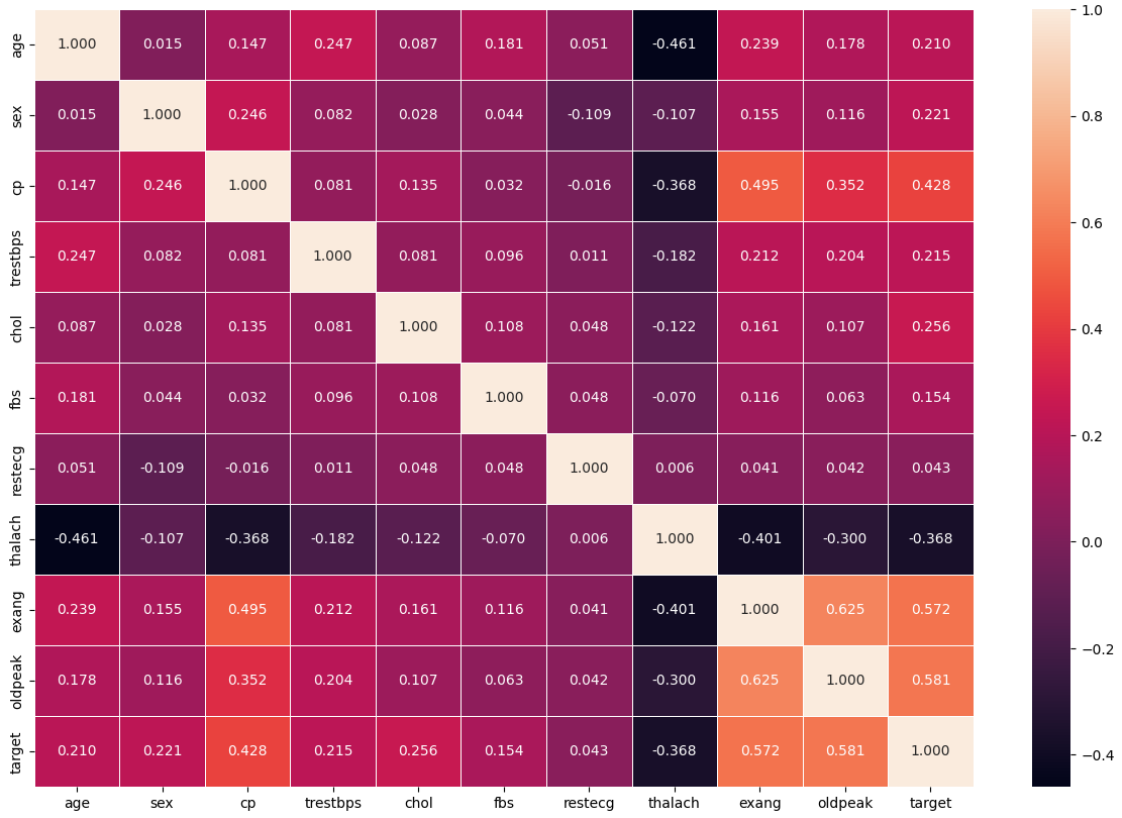
[ ]:
```
dfClean.corr()
```

[ ]:
```
              age       sex        cp  trestbps      chol       fbs  \
age      1.000000  0.014516  0.146616  0.246571  0.087101  0.181130
sex      0.014516  1.000000  0.245769  0.082064  0.027695  0.044372
```

```
cp        0.146616  0.245769  1.000000  0.081293  0.134697  0.031930
trestbps  0.246571  0.082064  0.081293  1.000000  0.080818  0.096222
chol      0.087101  0.027695  0.134697  0.080818  1.000000  0.107686
fbs       0.181130  0.044372  0.031930  0.096222  0.107686  1.000000
restecg   0.050672 -0.108656 -0.016372  0.011256  0.048081  0.047988
thalach  -0.460514 -0.106959 -0.367819 -0.181824 -0.122038 -0.069722
exang     0.239223  0.154925  0.494674  0.211507  0.161055  0.115503
oldpeak   0.178172  0.115959  0.351735  0.204000  0.106743  0.063179
target    0.210429  0.220732  0.427536  0.214898  0.256027  0.154319

            restecg   thalach     exang   oldpeak    target
age        0.050672 -0.460514  0.239223  0.178172  0.210429
sex       -0.108656 -0.106959  0.154925  0.115959  0.220732
cp        -0.016372 -0.367819  0.494674  0.351735  0.427536
trestbps   0.011256 -0.181824  0.211507  0.204000  0.214898
chol       0.048081 -0.122038  0.161055  0.106743  0.256027
fbs        0.047988 -0.069722  0.115503  0.063179  0.154319
restecg    1.000000  0.006084  0.041290  0.042193  0.042643
thalach    0.006084  1.000000 -0.400508 -0.300458 -0.367525
exang      0.041290 -0.400508  1.000000  0.624965  0.571710
oldpeak    0.042193 -0.300458  0.624965  1.000000  0.580732
target     0.042643 -0.367525  0.571710  0.580732  1.000000
```

[ ]:
```
cor_mat = dfClean.corr()
fig,ax = plt.subplots(figsize=(15,10))
sns.heatmap(cor_mat, annot=True, linewidths=0.5, fmt=".3f")
```

[ ]:
```
<Axes: >
```



6 6) Konstruksi Data

[ ]:
```
dfClean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 293 entries, 0 to 293
Data columns (total 11 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       293 non-null    float64
 1   sex       293 non-null    float64
 2   cp        293 non-null    float64
 3   trestbps  293 non-null    float64
 4   chol      293 non-null    float64
 5   fbs       293 non-null    float64
 6   restecg   293 non-null    float64
 7   thalach   293 non-null    float64
 8   exang     293 non-null    float64
 9   oldpeak   293 non-null    float64
 10  target    293 non-null    float64
```

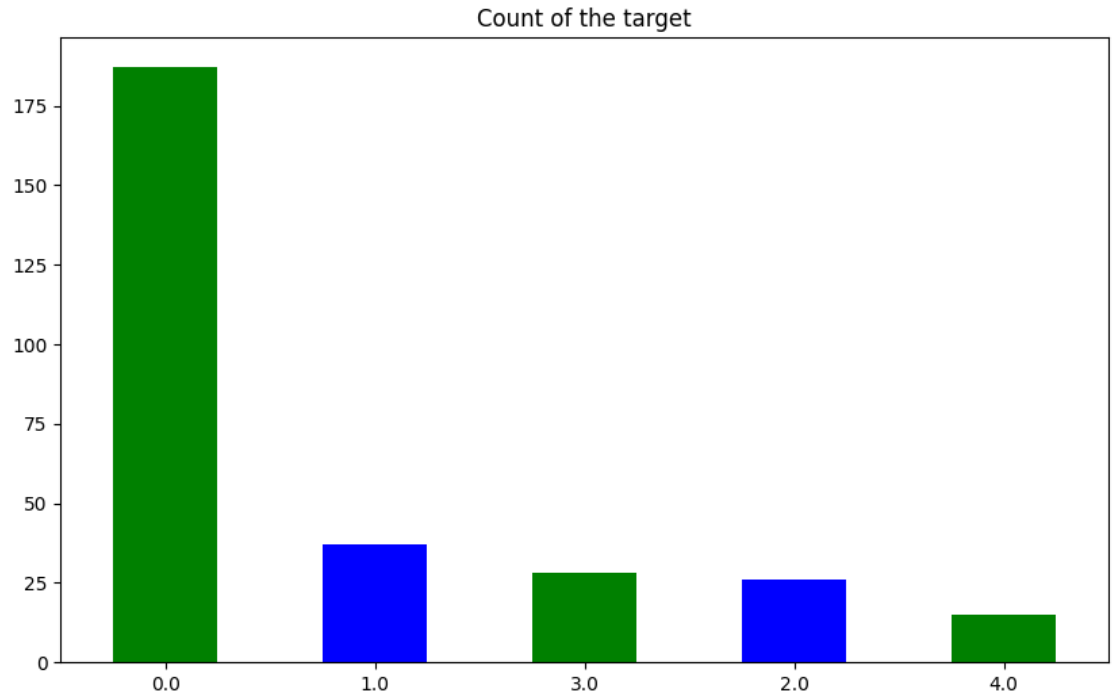
```
dtypes: float64(11)
memory usage: 27.5 KB
```

[ ]:
```
dfClean.head(5)
```

[ ]:
```
    age  sex   cp  trestbps   chol  fbs  restecg  thalach  exang  oldpeak  \
0  40.0  1.0  2.0     140.0  289.0  0.0      0.0    172.0    0.0      0.0
1  49.0  0.0  3.0     160.0  180.0  0.0      0.0    156.0    0.0      1.0
2  37.0  1.0  2.0     130.0  283.0  0.0      1.0     98.0    0.0      0.0
3  48.0  0.0  4.0     138.0  214.0  0.0      0.0    108.0    1.0      1.5
4  54.0  1.0  3.0     150.0  251.0  0.0      0.0    122.0    0.0      0.0

   target
0     0.0
1     1.0
2     0.0
3     3.0
4     0.0
```

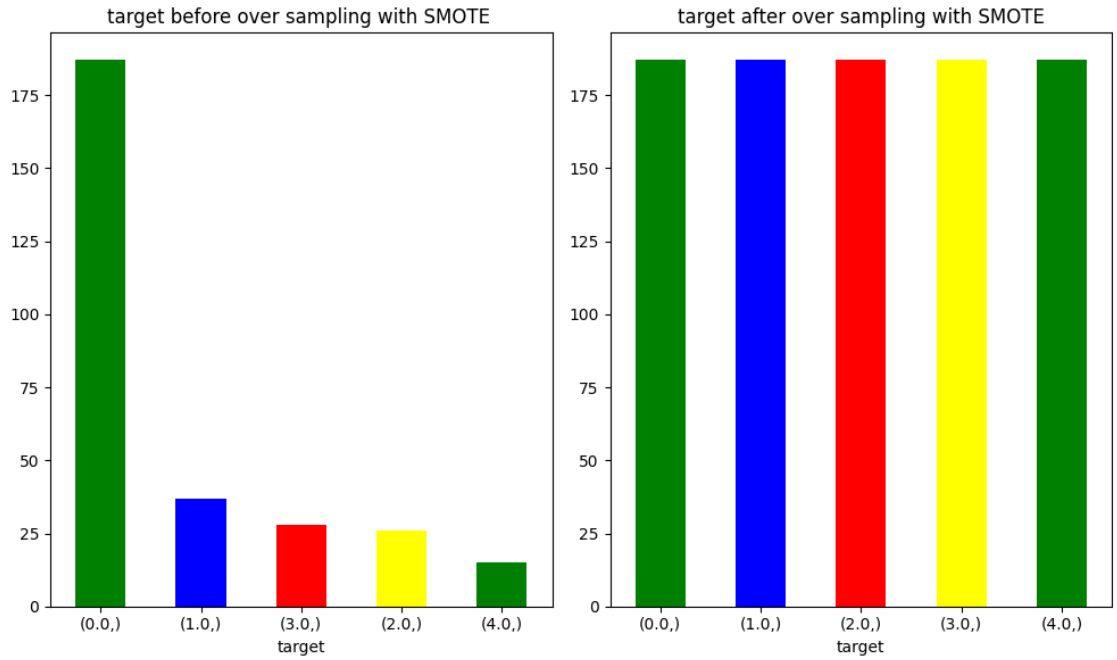
[ ]:
```
X = dfClean.drop("target", axis=1).values
y = dfClean.iloc[:, -1]
```

[ ]:
```
dfClean['target'].value_counts().
  ↪plot(kind='bar',figsize=(10,6),color=['green','blue'])
plt.title("Count of the target")
plt.xticks(rotation=0);
```



[ ]:
```
from imblearn.over_sampling import SMOTE

# oversampling
smote = SMOTE(random_state=42)
X_smote_resampled, y_smote_resampled = smote.fit_resample(X, y)
```

[ ]:
```
plt.subplot(1, 2, 1)

new_df1 = pd.DataFrame(data=y)

plt.subplot(1, 2, 1)
new_df1.value_counts().plot(kind='bar',figsize=(10,6),color=['green', 'blue',␣
 ↪'red', 'yellow'])
plt.title("target before over sampling with SMOTE ")
plt.xticks(rotation=0);

plt.subplot(1, 2, 2)
new_df2 = pd.DataFrame(data=y_smote_resampled)

new_df2.value_counts().
 ↪plot(kind='bar',figsize=(10,6),color=['green','blue','red','yellow'])
plt.title("target after over sampling with SMOTE")
plt.xticks(rotation=0);

plt.tight_layout()
plt.show()
```



[ ]:
```
new_df1 = pd.DataFrame(data=y)
new_df1.value_counts()
```

[ ]:
```
target
0.0       187
1.0        37
3.0        28
2.0        26
4.0        15
dtype: int64
```

[ ]:
```
# oversampling
new_df2 = pd.DataFrame(data=y_smote_resampled)
new_df2.value_counts()
```

[ ]:
```
target
0.0       187
1.0       187
2.0       187
3.0       187
4.0       187
dtype: int64
```

[ ]:
```
dfClean.describe()
```

[ ]:
```
              age         sex          cp    trestbps        chol         fbs  \
count  293.000000  293.000000  293.000000  293.000000  293.000000  293.000000
mean    47.822526    0.726962    2.986348  132.662116  250.860068    0.068259
std      7.824875    0.446282    0.965049   17.576793   65.059069    0.252622
min     28.000000    0.000000    1.000000   92.000000   85.000000    0.000000
25%     42.000000    0.000000    2.000000  120.000000  211.000000    0.000000
50%     49.000000    1.000000    3.000000  130.000000  248.000000    0.000000
75%     54.000000    1.000000    4.000000  140.000000  277.000000    0.000000
max     66.000000    1.000000    4.000000  200.000000  603.000000    1.000000

          restecg     thalach       exang     oldpeak      target
count  293.000000  293.000000  293.000000  293.000000  293.000000
mean     0.218430  139.058020    0.303754    0.588055    0.795222
std      0.460868   23.558003    0.460665    0.909554    1.238251
min      0.000000   82.000000    0.000000    0.000000    0.000000
25%      0.000000  122.000000    0.000000    0.000000    0.000000
50%      0.000000  140.000000    0.000000    0.000000    0.000000
75%      0.000000  155.000000    1.000000    1.000000    1.000000
max      2.000000  190.000000    1.000000    5.000000    4.000000
```

[ ]:
```
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()

X_smote_resampled_normal = scaler.fit_transform(X_smote_resampled)
```

[ ]:
```
len(X_smote_resampled_normal)
```

[ ]:
```
935
```

[ ]:
```
dfcek1 = pd.DataFrame(X_smote_resampled_normal)
dfcek1.describe()
```

[ ]:
```
                0           1           2           3           4           5  \
count  935.000000  935.000000  935.000000  935.000000  935.000000  935.000000
mean     0.563739    0.842507    0.818224    0.403413    0.341027    0.094277
std      0.174873    0.332492    0.274211    0.147493    0.110990    0.252030
min      0.000000    0.000000    0.000000    0.000000    0.000000    0.000000
25%      0.473283    1.000000    0.666667    0.305556    0.267954    0.000000
50%      0.578947    1.000000    1.000000    0.387952    0.330240    0.000000
75%      0.683363    1.000000    1.000000    0.487481    0.393811    0.000000
max      1.000000    1.000000    1.000000    1.000000    1.000000    1.000000

                6           7           8           9
count  935.000000  935.000000  935.000000  935.000000
mean     0.117938    0.453354    0.598398    0.227015
std      0.199527    0.197232    0.450288    0.201293
```

min	0.000000	0.000000	0.000000	0.000000
25%	0.000000	0.312720	0.000000	0.000000
50%	0.000000	0.440606	0.962447	0.200000
75%	0.201473	0.593629	1.000000	0.386166
max	1.000000	1.000000	1.000000	1.000000