

Home x MAIN FLOW TASK 5 x +

localhost:8888/notebooks/MAIN%20FLOW%20TASK%205.ipynb? ☆ |

jupyter MAIN FLOW TASK 5 Last Checkpoint: 35 minutes ago Trusted

File Edit View Run Kernel Settings Help JupyterLab Python 3 (ipykernel)

```
[2]: #Import all the libraries that we need.
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

[3]: #importing data set
df = pd.read_csv('C:\\Users\\aadil\\Downloads\\heart.csv')

[4]: df.head()

[4]:
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0

```
[8]: df.columns.values

[8]: array(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg',
        'thalach', 'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'],
       dtype=object)

[10]: #checking for Null values
df.isna().sum()

[10]: age 0
```

Home

MAIN FLOW TASK 5

localhost:8888/notebooks/MAIN%20FLOW%20TASK%205.ipynb?

jupyter

MAIN FLOW TASK 5

Last Checkpoint: 36 minutes ago

Trusted

File Edit View Run Kernel Settings Help

dtype=object)

JupyterLab

Python 3 (ipykernel)

[10]:

#checking for Null values  
df.isna().sum()

[10]:

age 0  
sex 0  
cp 0  
trestbps 0  
chol 0  
fbs 0  
restecg 0  
thalach 0  
exang 0  
oldpeak 0  
slope 0  
ca 0  
thal 0  
target 0  
dtype: int64

[12]:

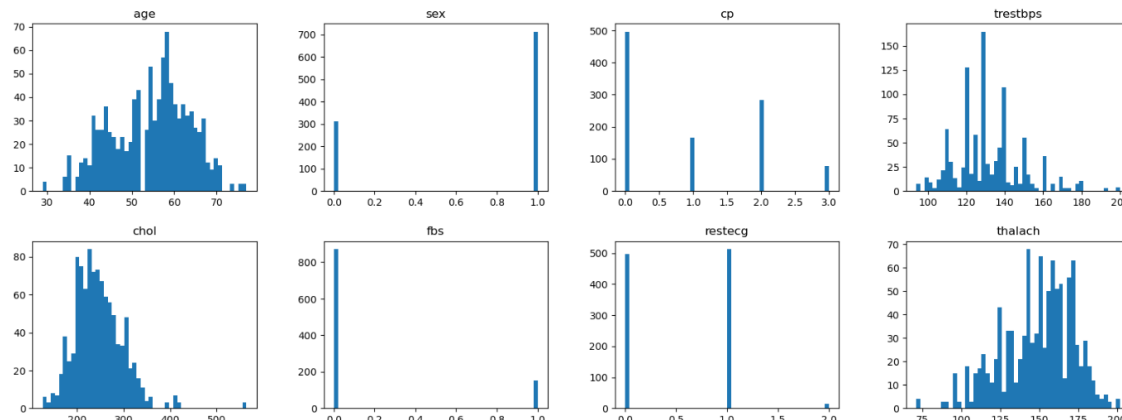
#consise summary of our dataset  
df.info()

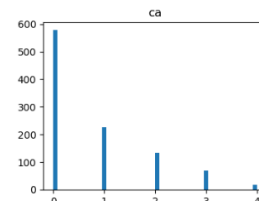
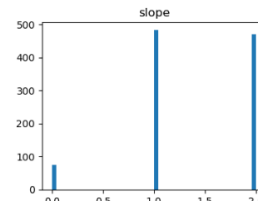
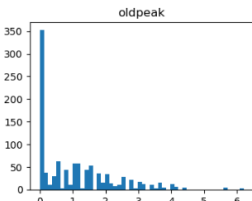
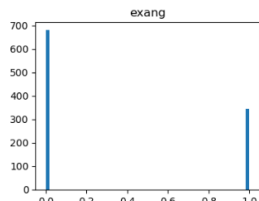
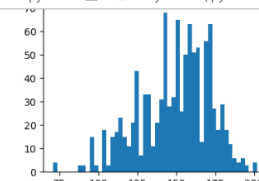
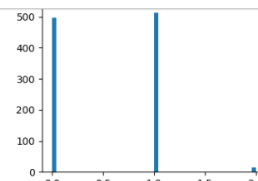
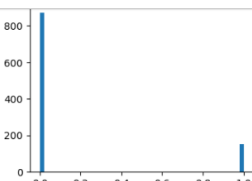
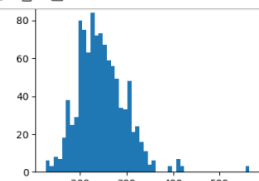
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 1025 entries, 0 to 1024  
Data columns (total 14 columns):  
# Column Non-Null Count Dtype  
---  
0 age 1025 non-null int64  
1 sex 1025 non-null int64  
2 cp 1025 non-null int64  
3 trestbps 1025 non-null int64  
4 chol 1025 non-null int64  
5 fbs 1025 non-null int64  
6 restecg 1025 non-null int64  
7 thalach 1025 non-null int64  
8 exang 1025 non-null int64



```
6 restecg 1025 non-null int64
7 thalach 1025 non-null int64
8 exang 1025 non-null int64
9 oldpeak 1025 non-null float64
10 slope 1025 non-null int64
11 ca 1025 non-null int64
12 thal 1025 non-null int64
13 target 1025 non-null int64
dtypes: float64(1), int64(13)
memory usage: 112.2 KB
```

```
[14]: #Plotting histogram of all numeric values
df.hist(bins=50, grid=False, figsize=(20,15));
```



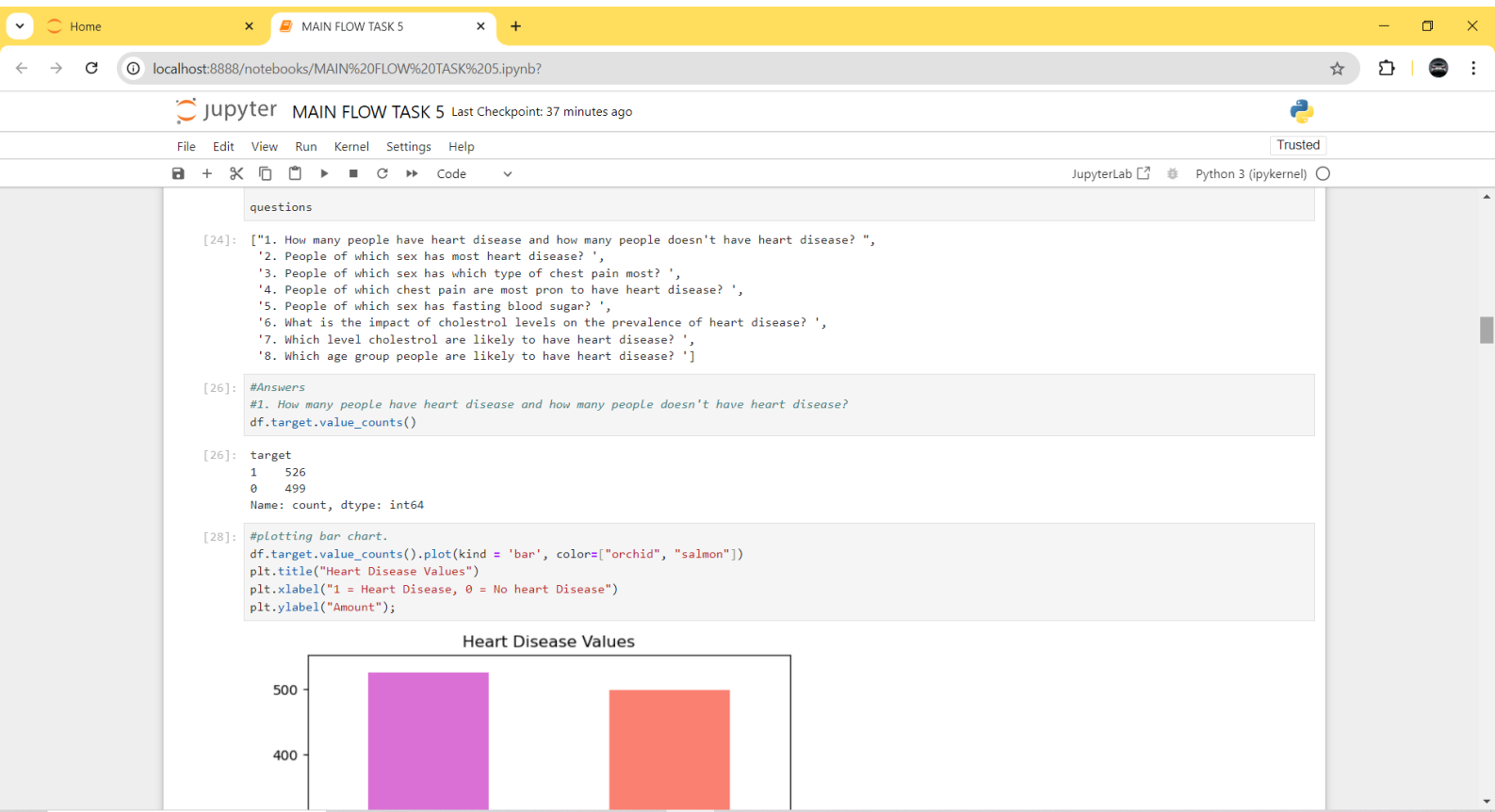


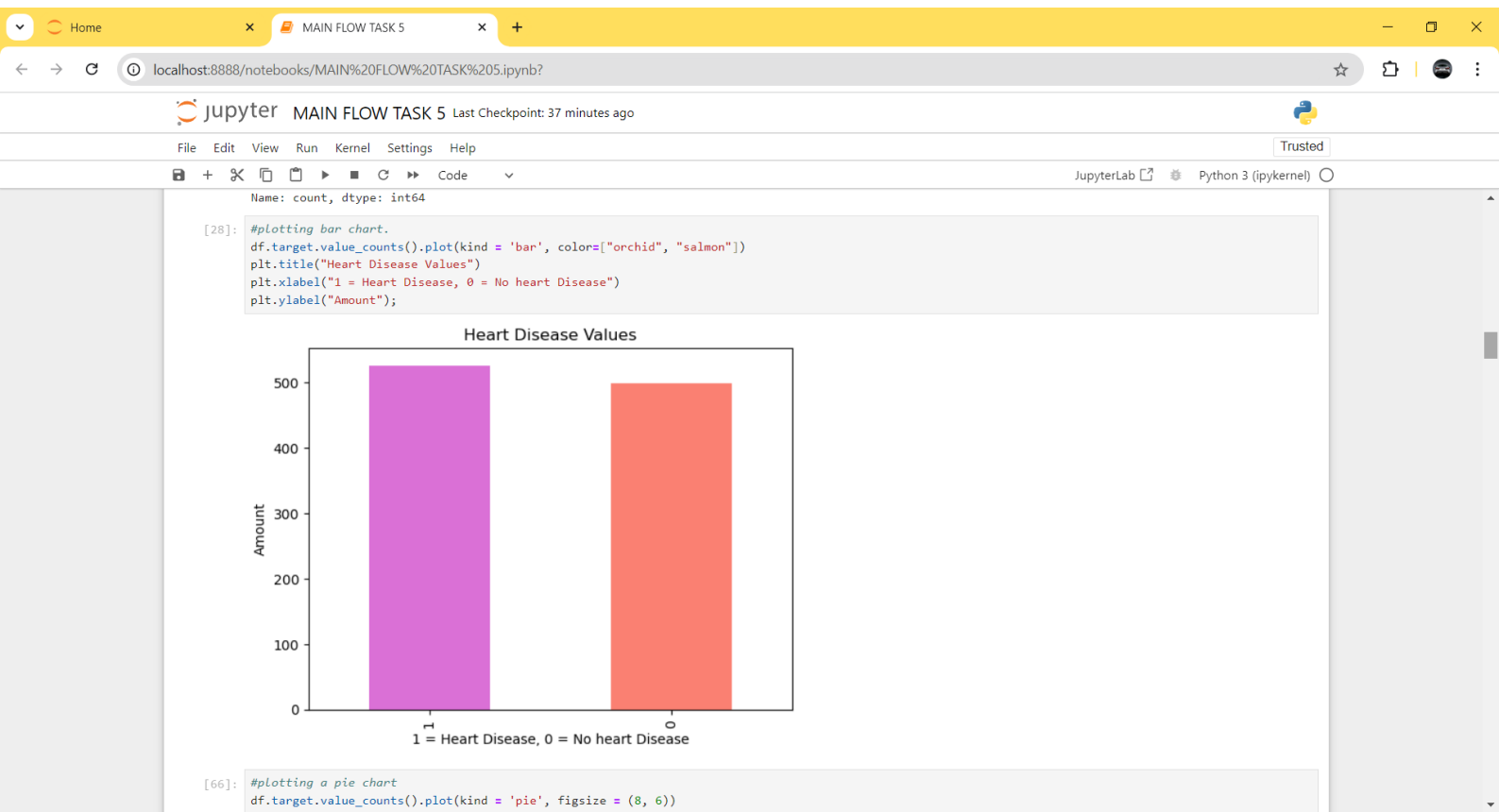
[16]: #Generate discriptive Statistics

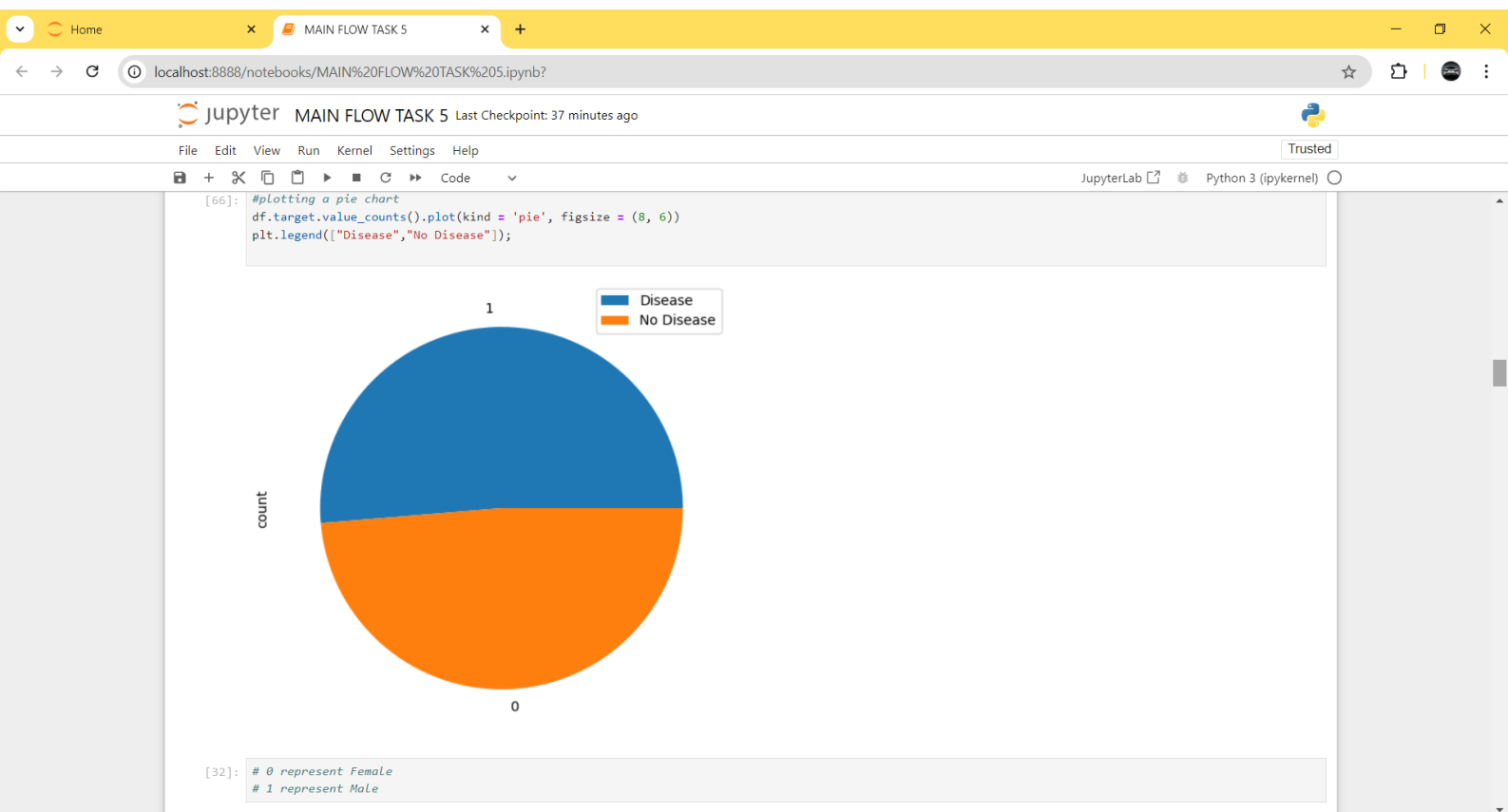
The screenshot shows a JupyterLab interface with a yellow header bar. The main area displays a Jupyter notebook with two code cells. The first cell, labeled [16], contains the code `#Generate descriptive Statistics` and `df.describe()`. Below the code, a data table is displayed with 13 columns: **count**, **age**, **sex**, **cp**, **trestbps**, **chol**, **fb**, **restecg**, **thalach**, **exang**, **oldpeak**, **slope**, and **ca**. The table shows statistical summaries for each variable. The second cell, labeled [24], contains a list of 8 questions related to heart disease, such as "1. How many people have heart disease and how many people doesn't have heart disease?".

	age	sex	cp	trestbps	chol	fb	restecg	thalach	exang	oldpeak	slope	ca
<b>count</b>	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000
<b>mean</b>	54.434146	0.695610	0.942439	131.611707	246.000000	0.149268	0.529756	149.114146	0.336585	1.071512	1.385366	0.754146
<b>std</b>	9.072290	0.460373	1.029641	17.516718	51.59251	0.356527	0.527878	23.005724	0.472772	1.175053	0.617755	1.030798
<b>min</b>	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	0.000000	0.000000	0.000000
<b>25%</b>	48.000000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	132.000000	0.000000	0.000000	1.000000	0.000000
<b>50%</b>	56.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000	152.000000	0.000000	0.800000	1.000000	0.000000
<b>75%</b>	61.000000	1.000000	2.000000	140.000000	275.000000	0.000000	1.000000	166.000000	1.000000	1.800000	2.000000	1.000000
<b>max</b>	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000	6.200000	2.000000	4.000000

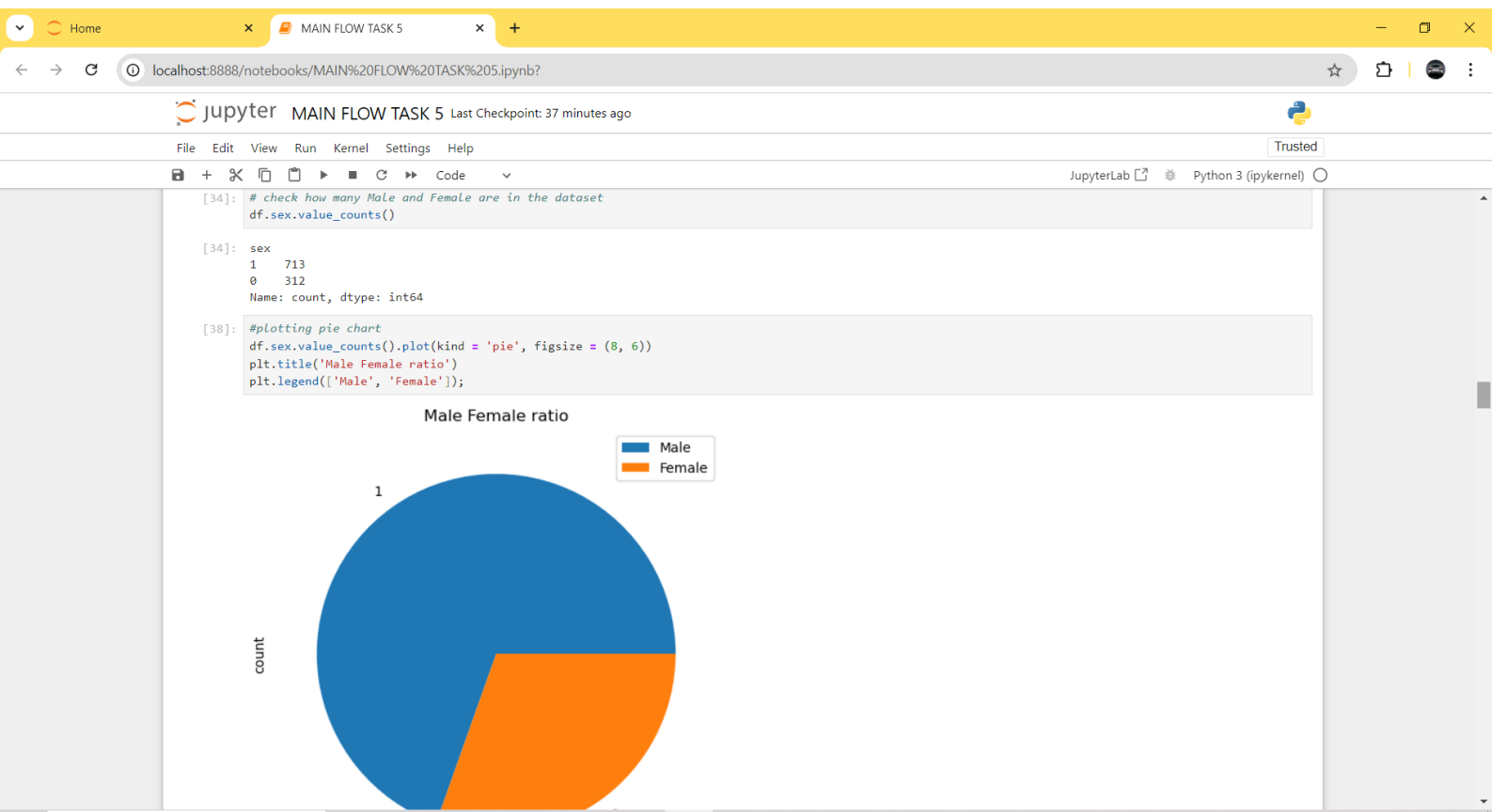
[24]: questions = ["1. How many people have heart disease and how many people doesn't have heart disease? ",  
"2. People of which sex has most heart disease? ",  
"3. People of which sex has which type of chest pain most? ",  
"4. People of which chest pain are most pron to have heart disease? ",  
"5. People of which sex has fasting blood sugar? ",  
"6. What is the impact of cholestrol levels on the prevalence of heart disease? ",  
"7. Which level cholestrol are likely to have heart disease? ",  
"8. Which age group people are likely to have heart disease? "]  
  
questions

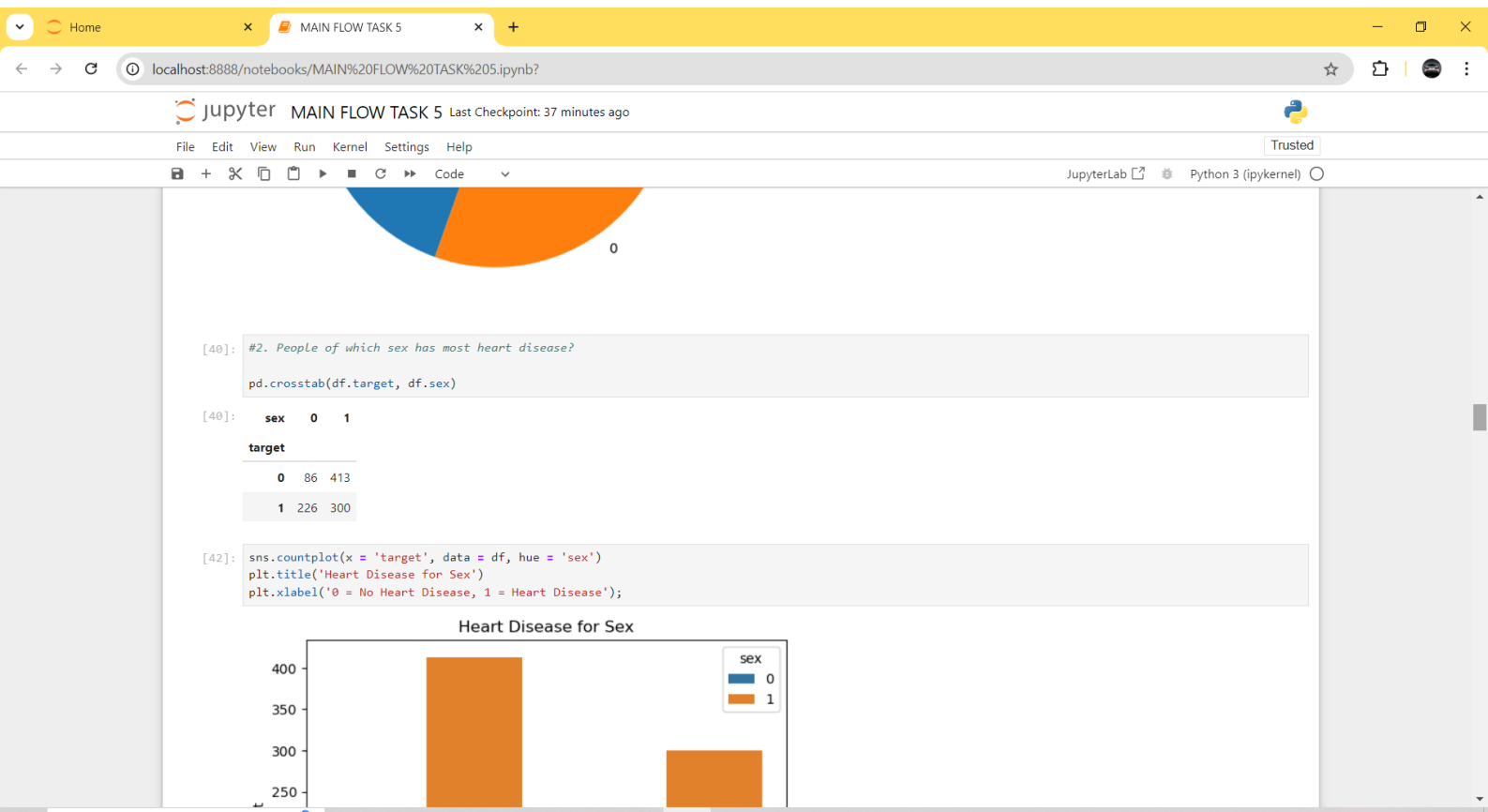


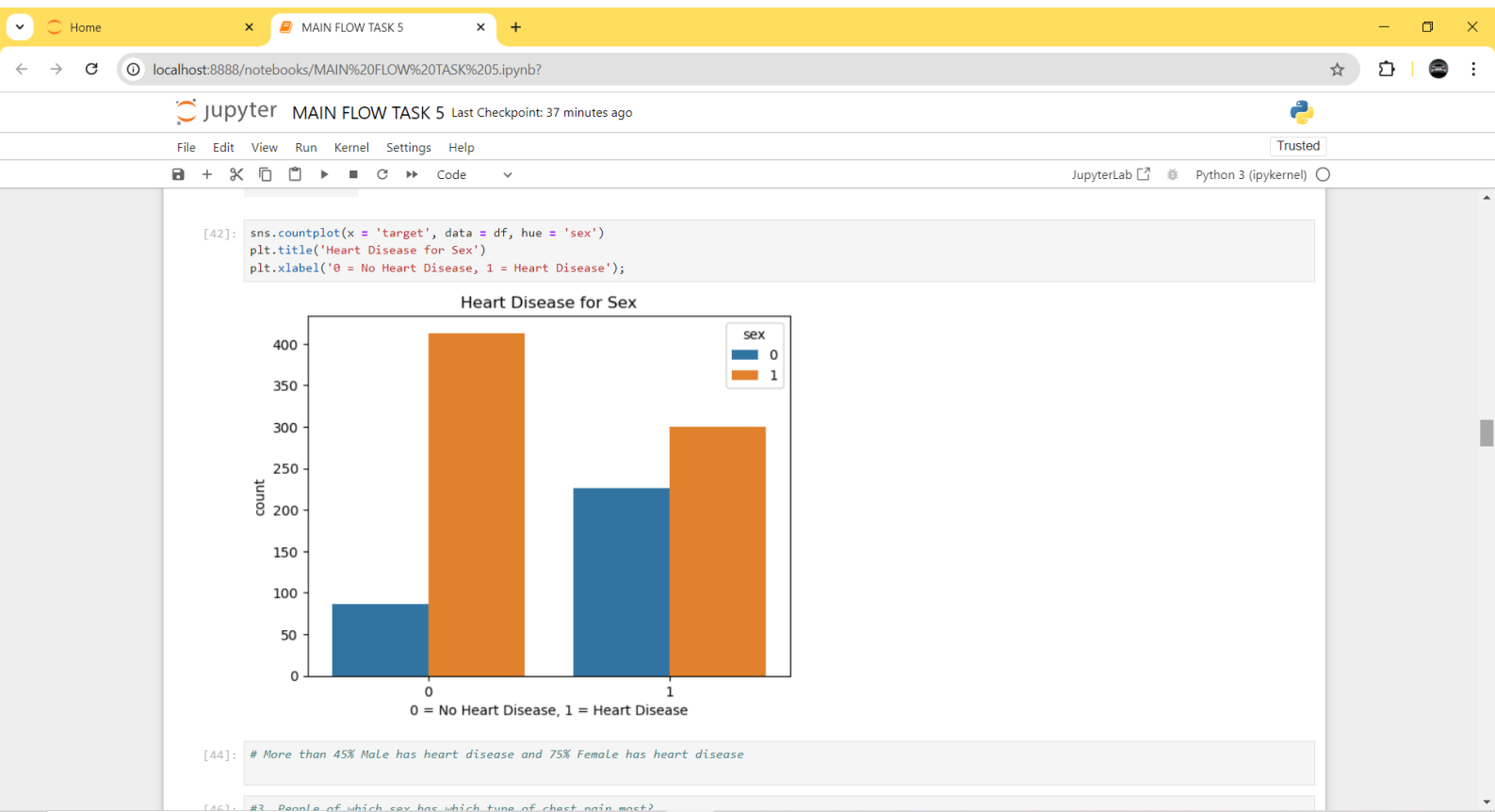


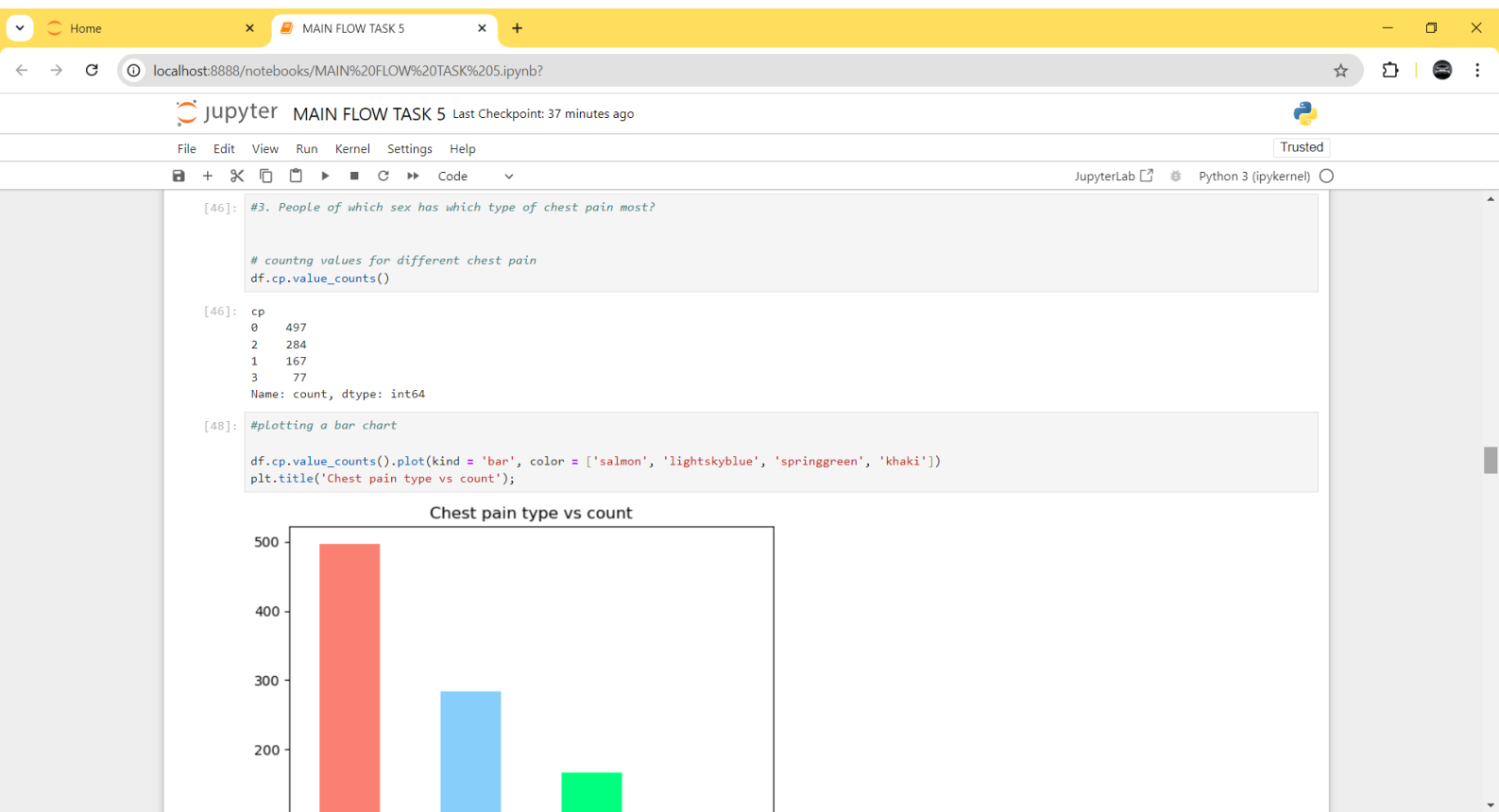


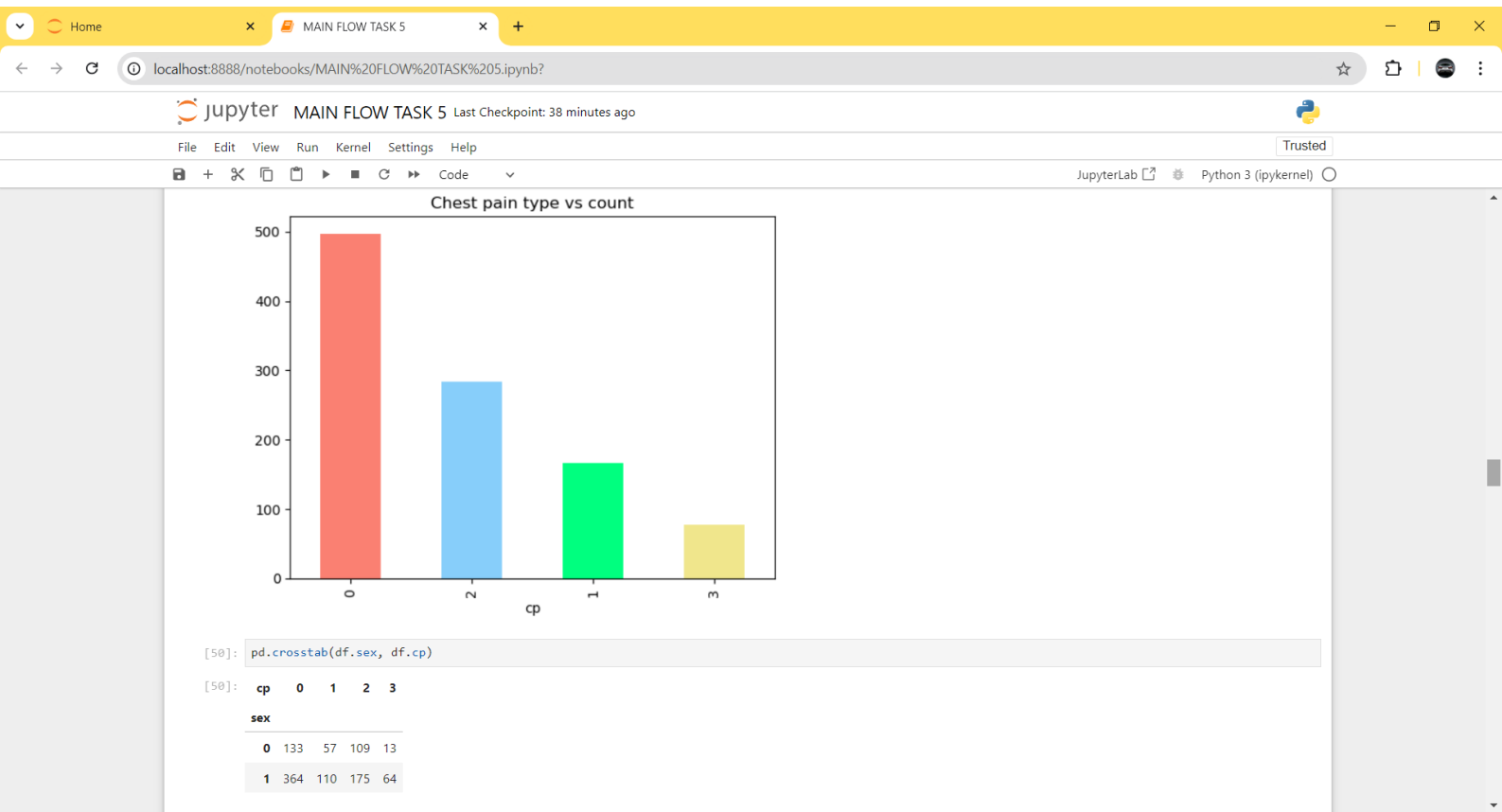


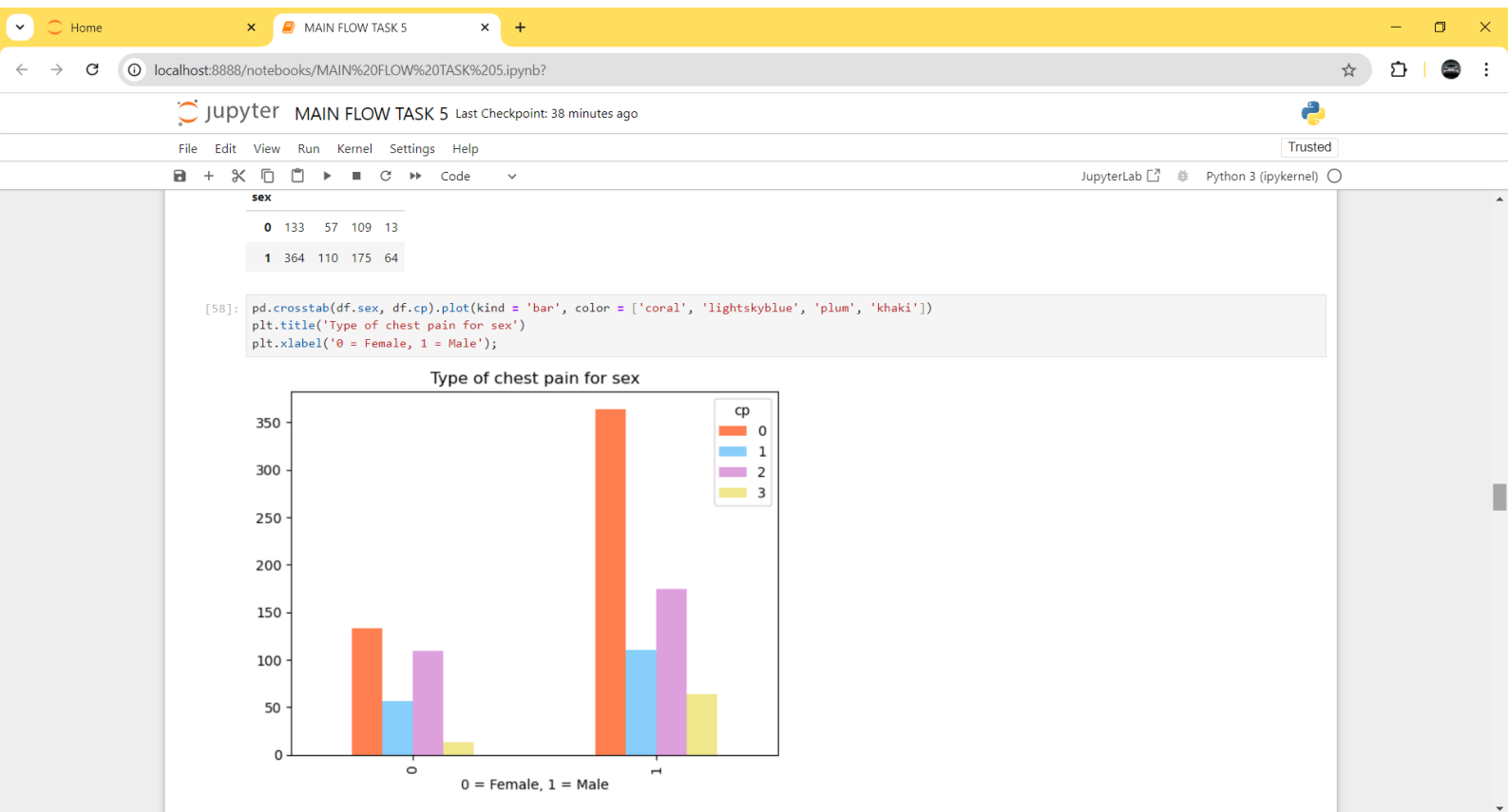


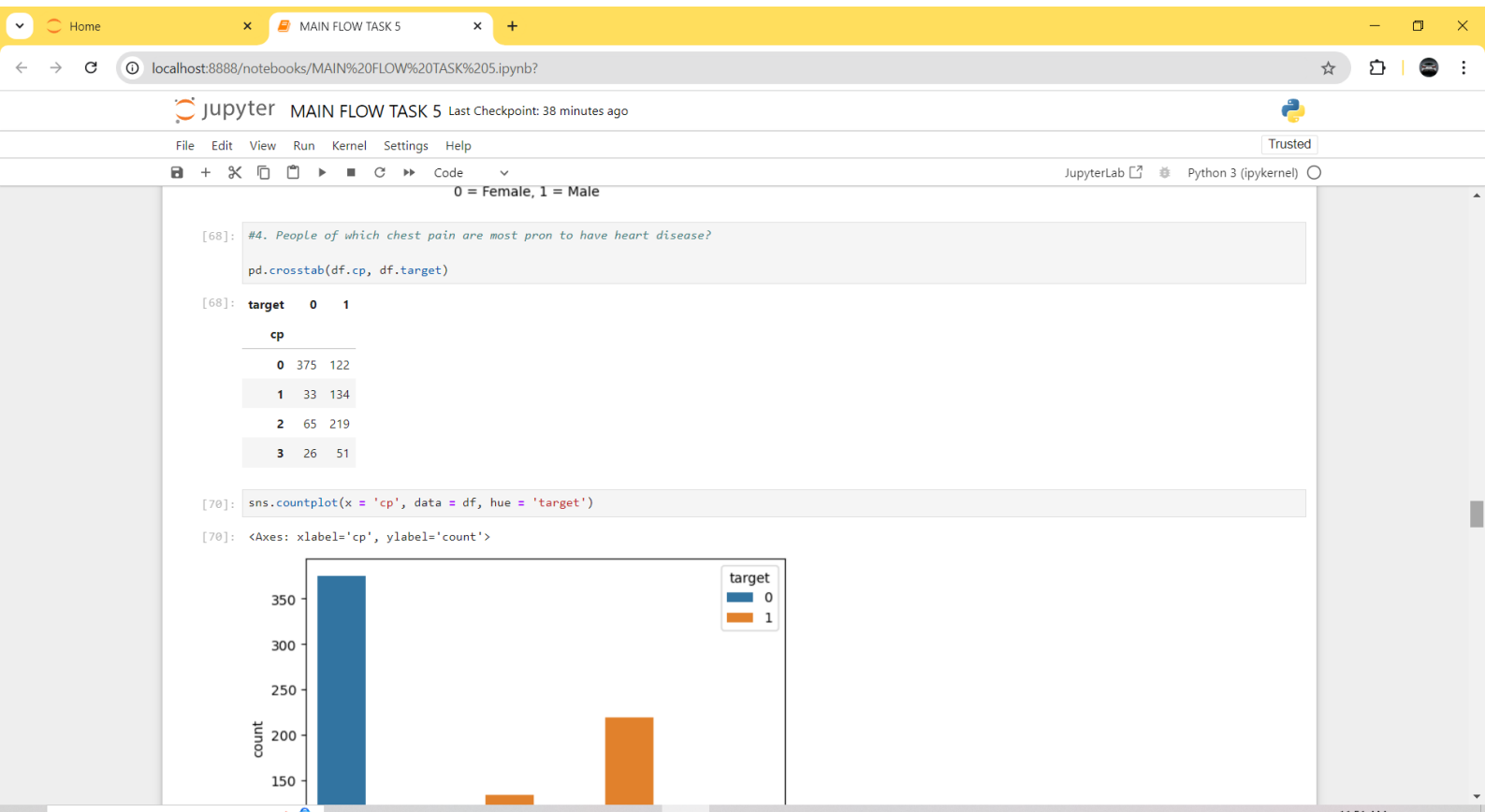


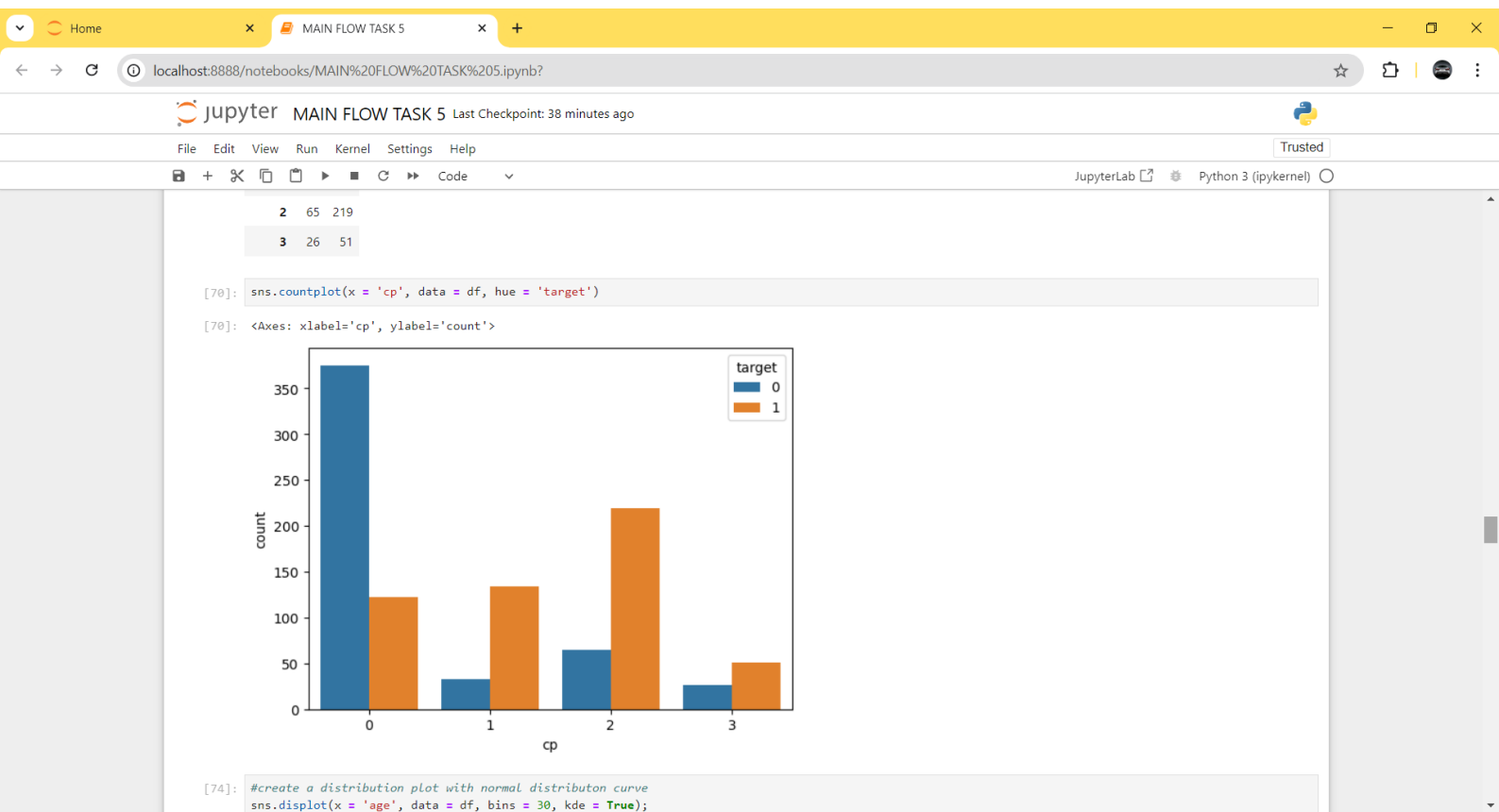




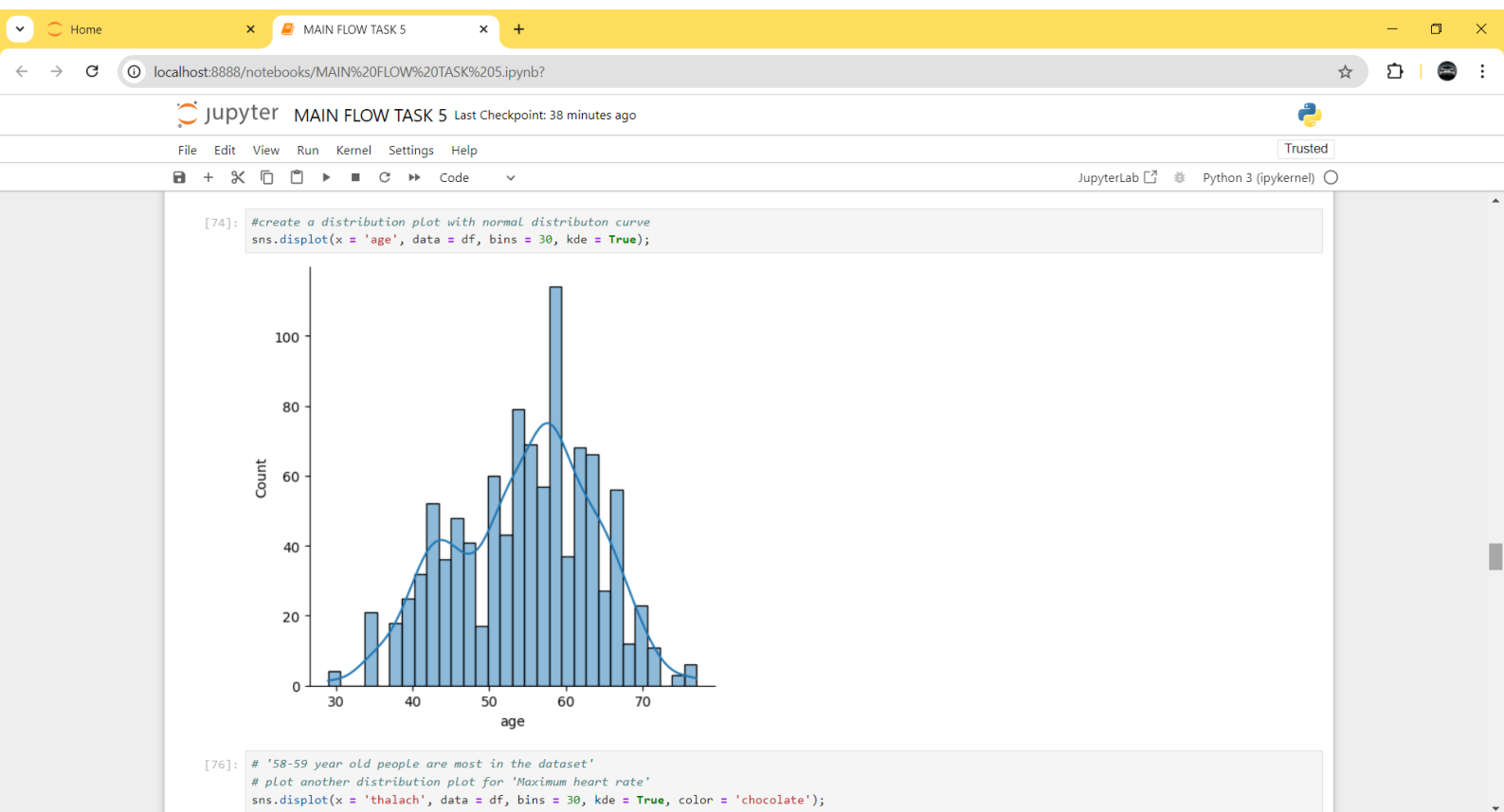


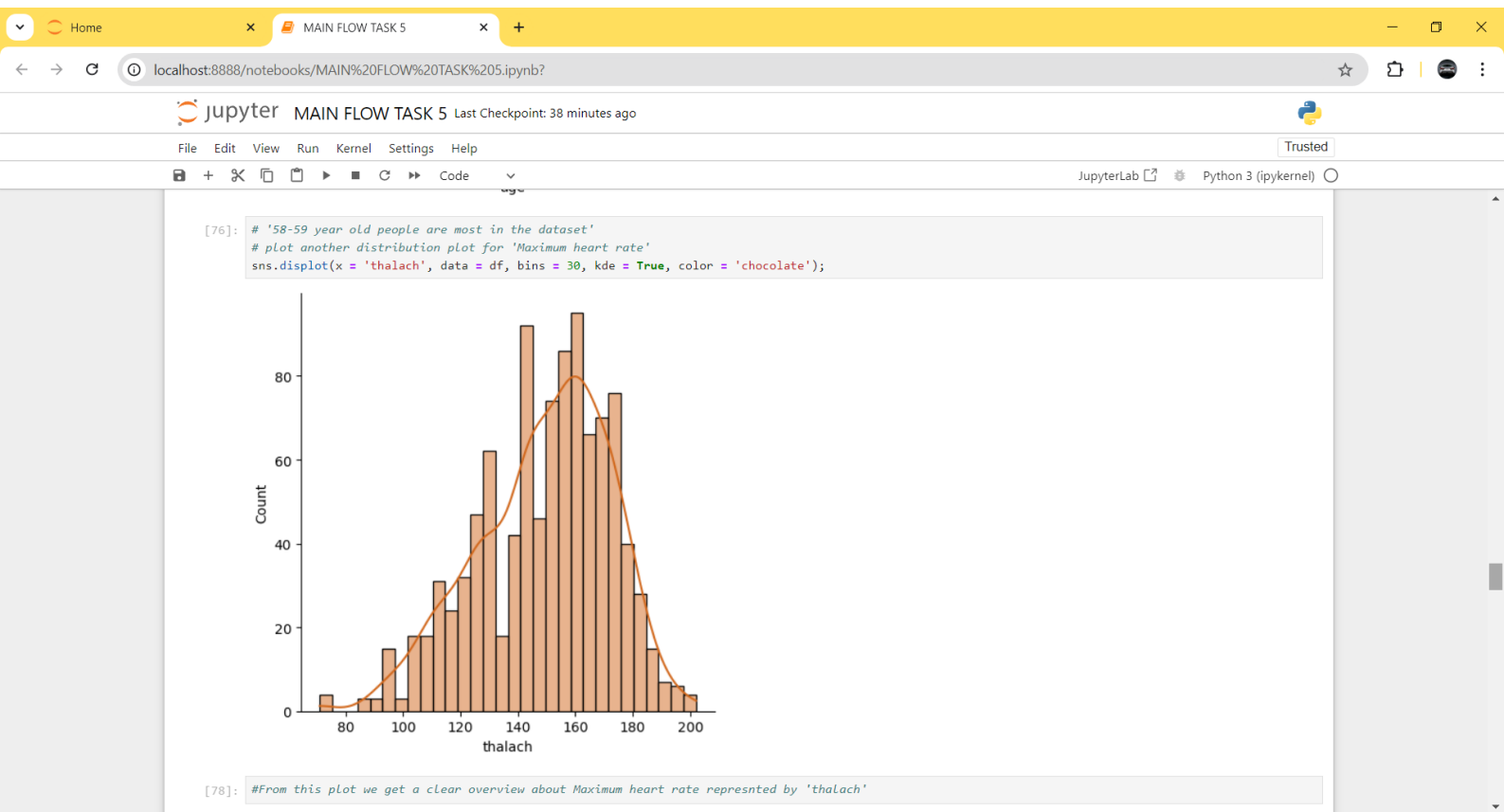


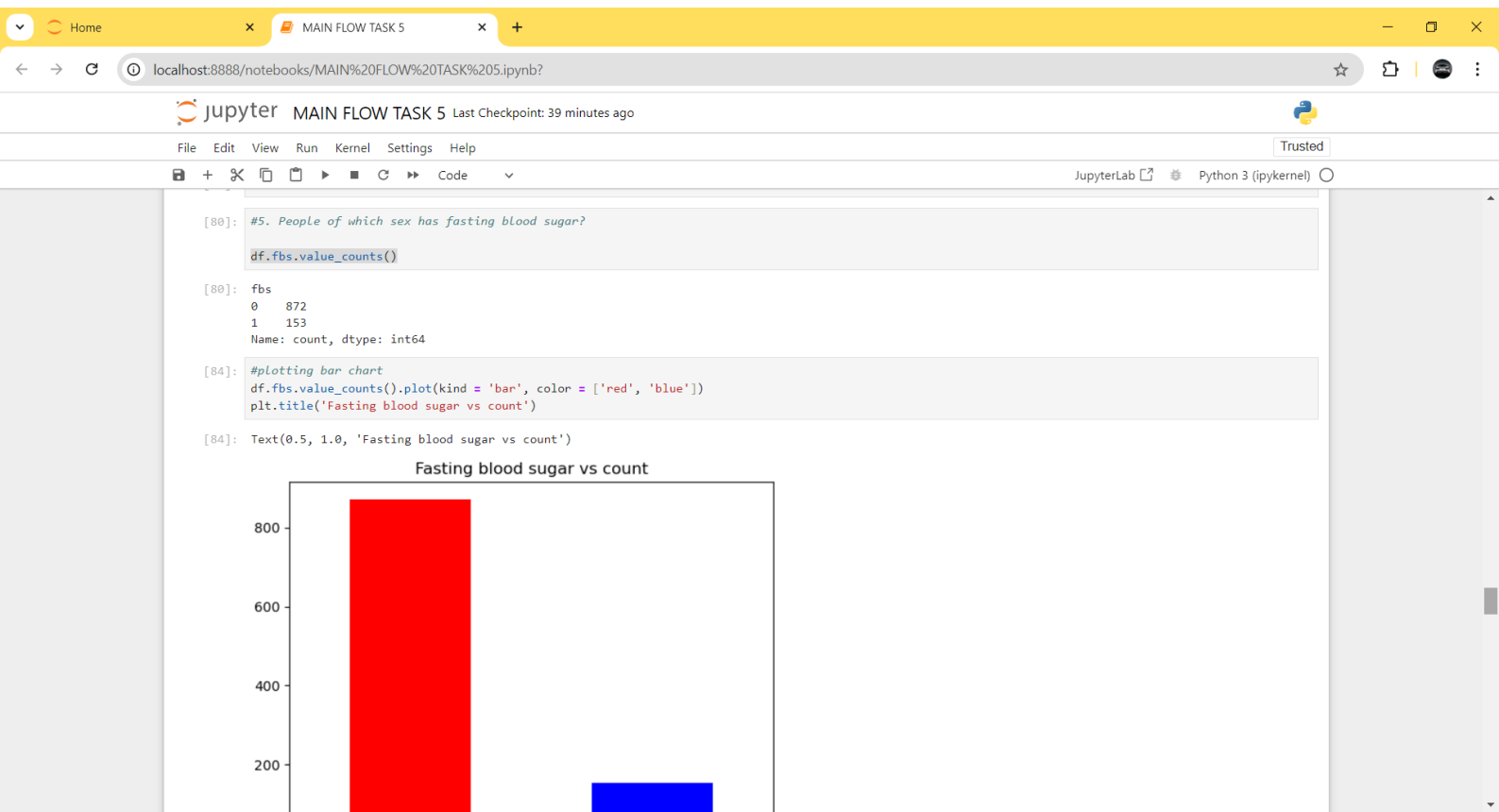




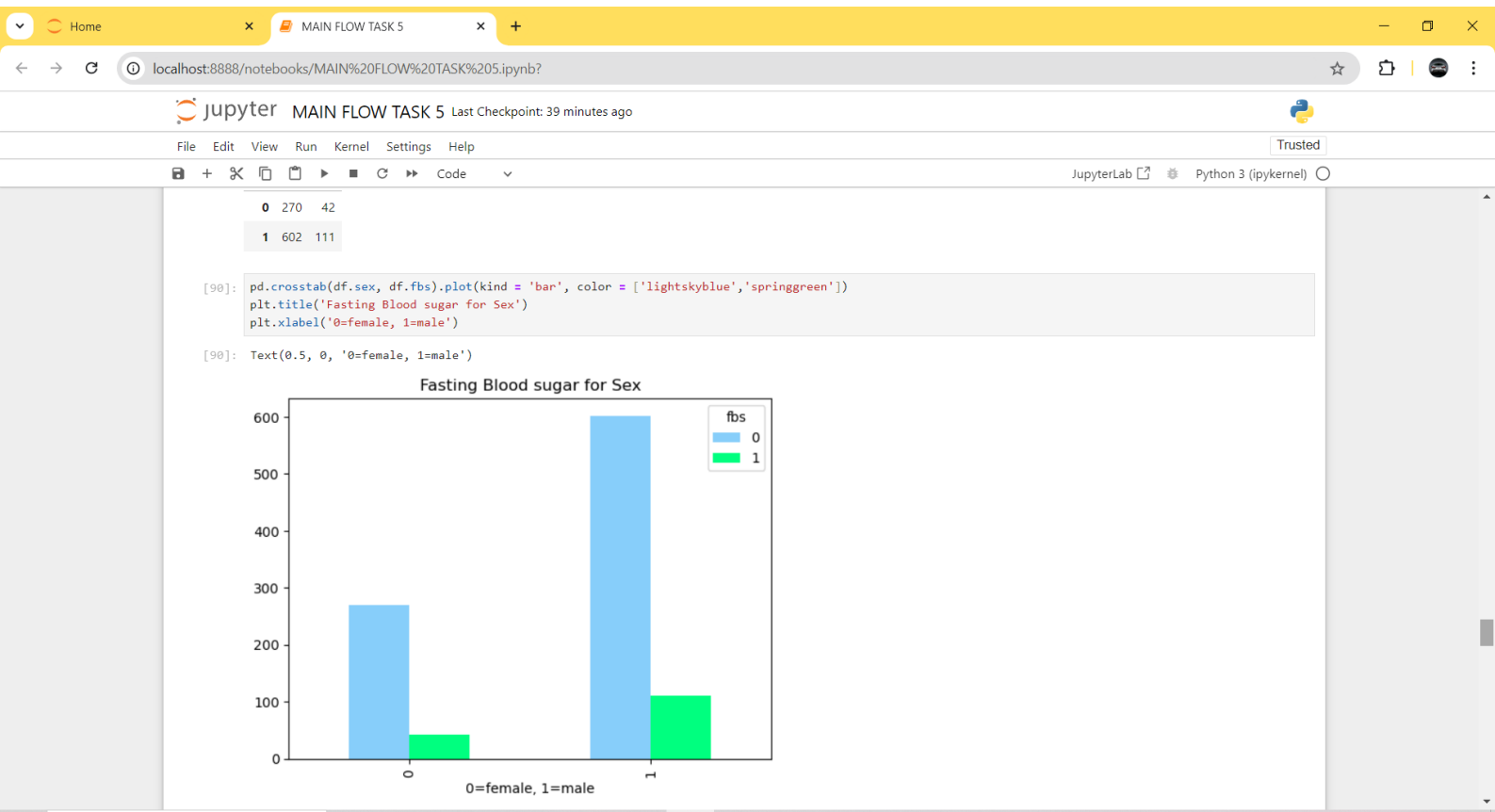


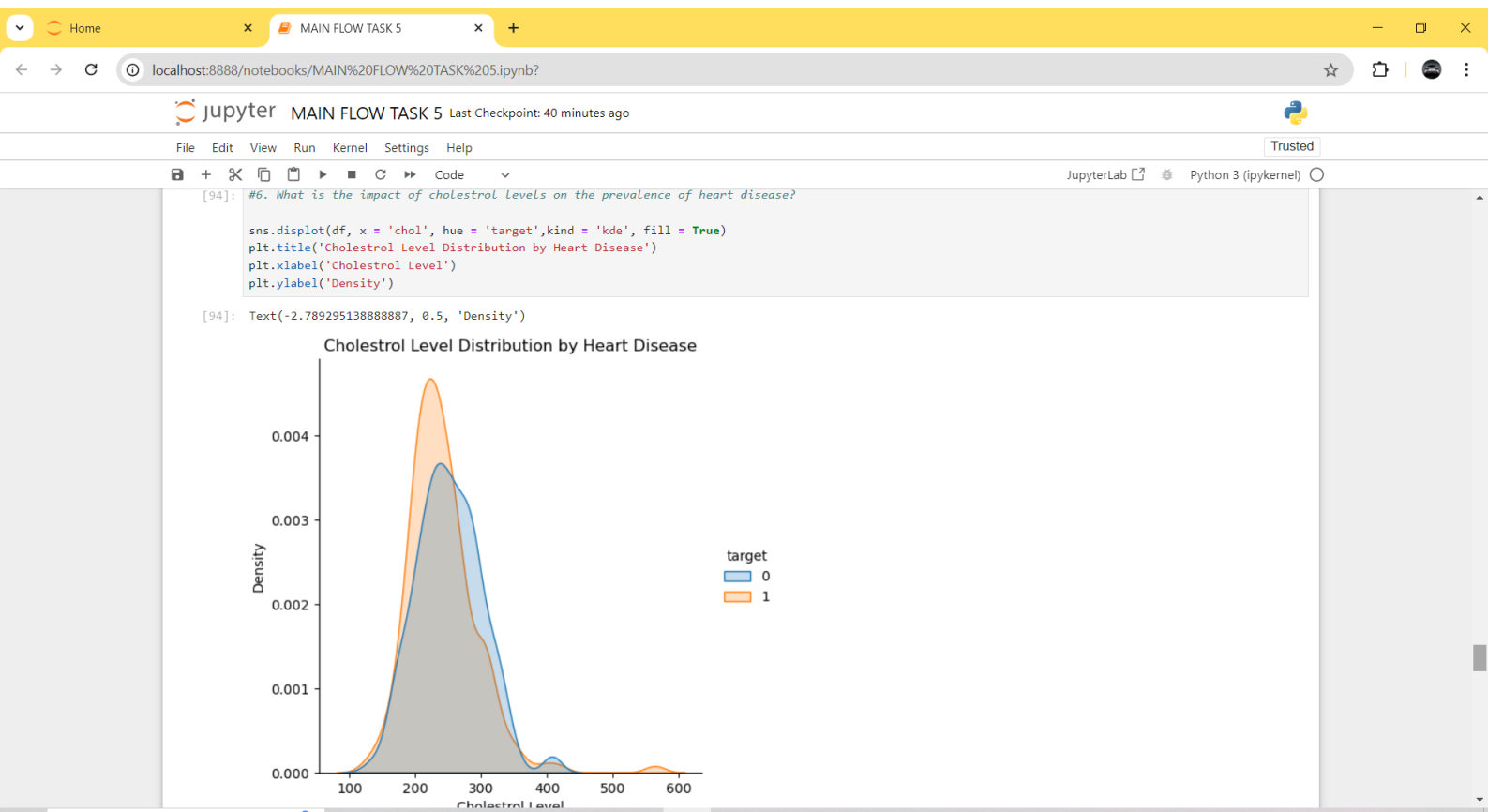


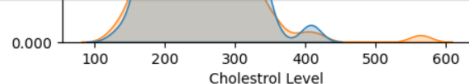










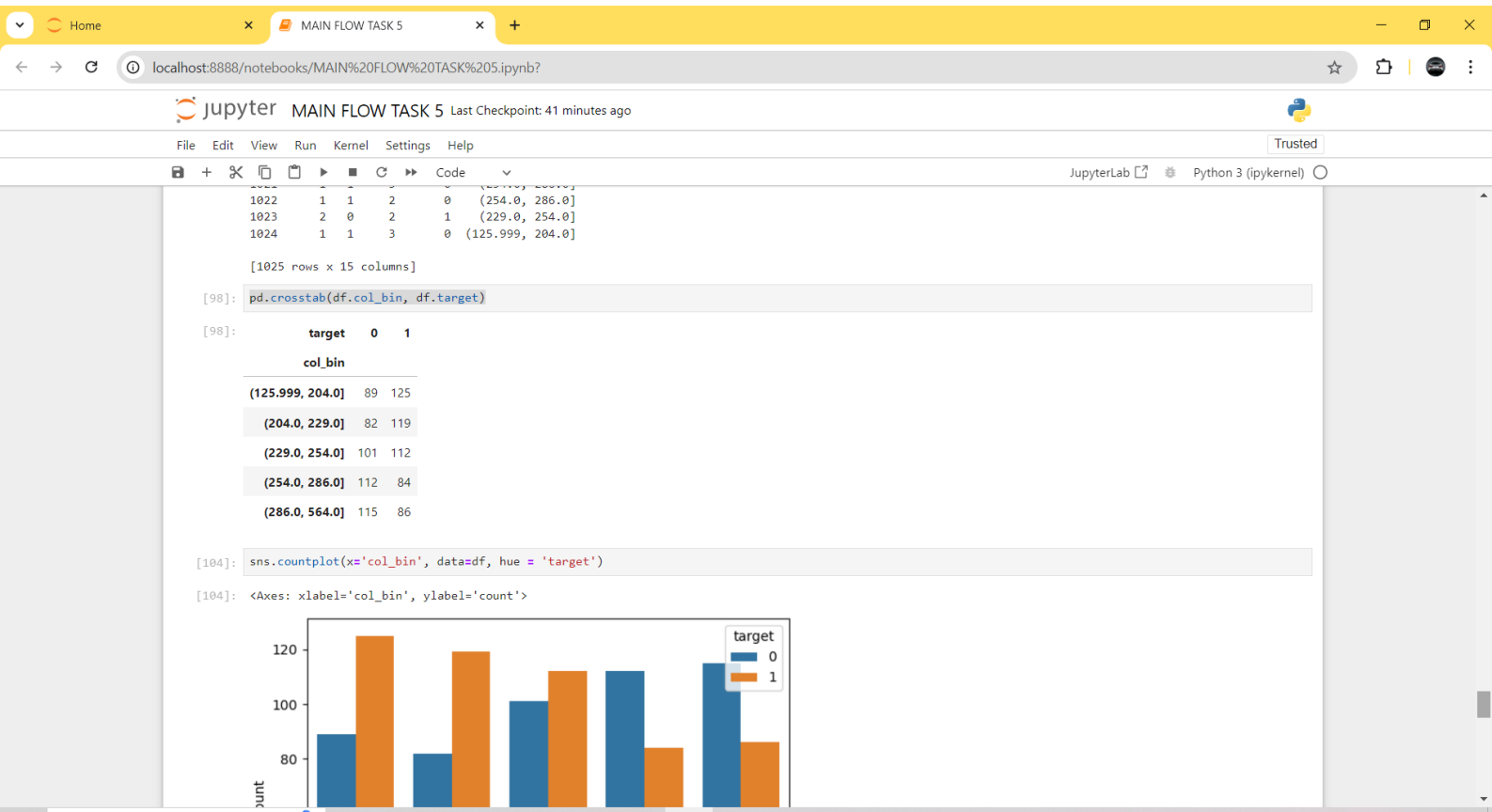


[96]: #7. Which Level cholesterol are Likely to have heart disease?

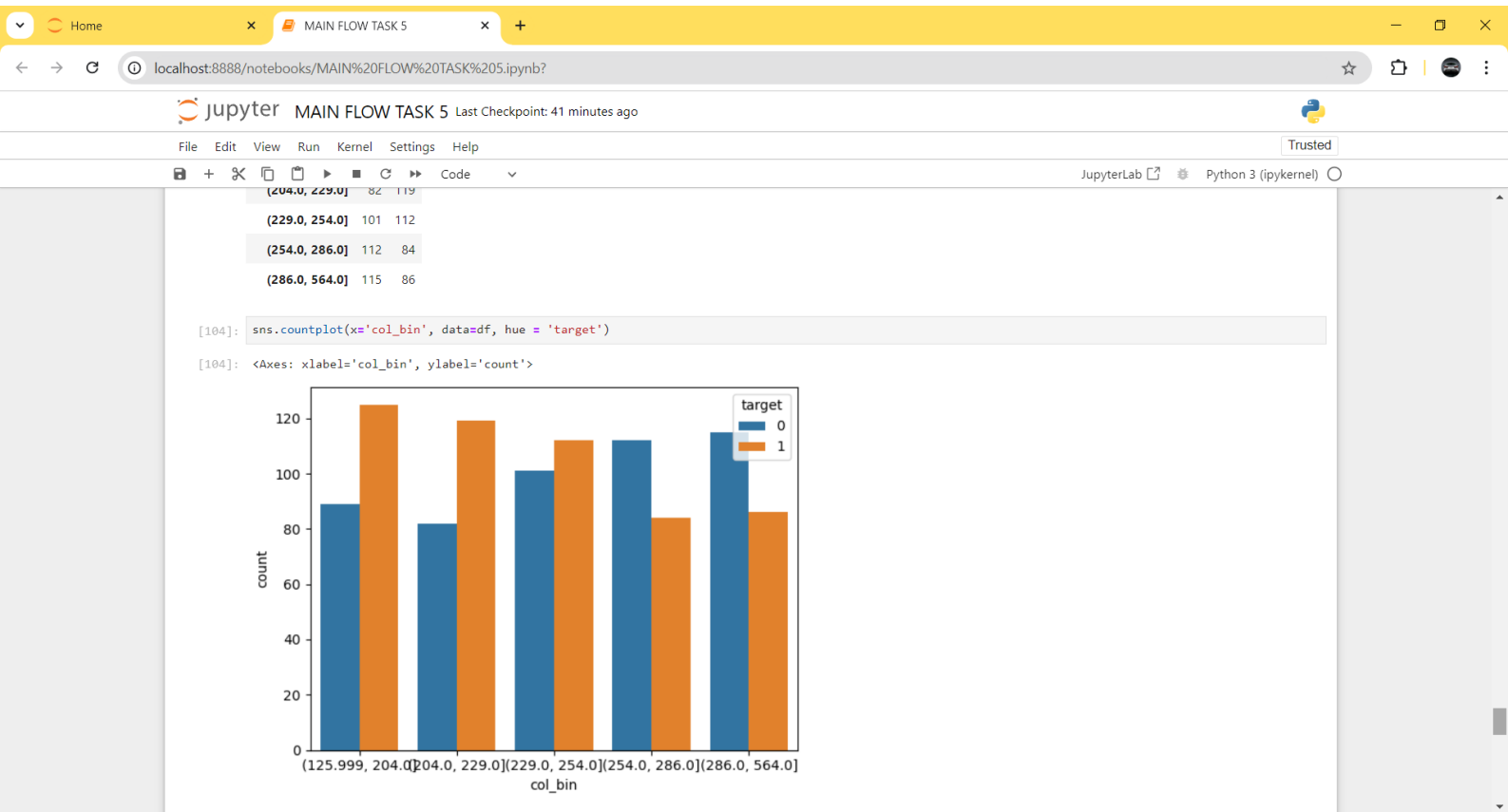
```
#grouping cholesterol using binning
df['col_bin']=pd.qcut(df['chol'],q=5)
print(df)
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	\
0	52	1	0	125	212	0	1	168	0	1.0	
1	53	1	0	140	203	1	0	155	1	3.1	
2	70	1	0	145	174	0	1	125	1	2.6	
3	61	1	0	148	203	0	1	161	0	0.0	
4	62	0	0	138	294	1	1	106	0	1.9	
...	...	...	...	...	...	...	...	...	...	...	
1020	59	1	1	140	221	0	1	164	1	0.0	
1021	60	1	0	125	258	0	0	141	1	2.8	
1022	47	1	0	110	275	0	0	118	1	1.0	
1023	50	0	0	110	254	0	0	159	0	0.0	
1024	54	1	0	120	188	0	1	113	0	1.4	

	slope	ca	thal	target	col_bin
0	2	2	3	0	(204.0, 229.0]
1	0	0	3	0	(125.999, 204.0]
2	0	0	3	0	(125.999, 204.0]
3	2	1	3	0	(125.999, 204.0]
4	1	3	2	0	(286.0, 564.0]
...	...	...	...	...	...
1020	2	0	2	1	(204.0, 229.0]
1021	1	1	3	0	(254.0, 286.0]
1022	1	1	2	0	(254.0, 286.0]
1023	2	0	2	1	(229.0, 254.0]
1024	1	1	3	0	(125.999, 204.0]







Home x MAIN FLOW TASK 5 x +

localhost:8888/notebooks/MAIN%20FLOW%20TASK%205.ipynb? ☆ |

jupyter MAIN FLOW TASK 5 Last Checkpoint: 41 minutes ago Trusted

File Edit View Run Kernel Settings Help JupyterLab Python 3 (ipykernel)

```
[106]: #8. Which age group people are likely to have heart disease?

#grouping age using binning
df['age_bin']=pd.qcut(df['age'],q=5)
print(df)
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	\
0	52	1	0	125	212	0	1	168	0	1.0	
1	53	1	0	140	203	1	0	155	1	3.1	
2	70	1	0	145	174	0	1	125	1	2.6	
3	61	1	0	148	203	0	1	161	0	0.0	
4	62	0	0	138	294	1	1	106	0	1.9	
...	...	...	...	...	...	...	...	...	...	...	...
1020	59	1	1	140	221	0	1	164	1	0.0	
1021	60	1	0	125	258	0	0	141	1	2.8	
1022	47	1	0	110	275	0	0	118	1	1.0	
1023	50	0	0	110	254	0	0	159	0	0.0	
1024	54	1	0	120	188	0	1	113	0	1.4	

	slope	ca	thal	target	col_bin	age_bin
0	2	2	3	0	(204.0, 229.0]	(45.0, 53.0]
1	0	0	3	0	(125.999, 204.0]	(45.0, 53.0]
2	0	0	3	0	(125.999, 204.0]	(62.0, 77.0]
3	2	1	3	0	(125.999, 204.0]	(58.0, 62.0]
4	1	3	2	0	(286.0, 564.0]	(58.0, 62.0]
...	...	...	...	...	...	...
1020	2	0	2	1	(204.0, 229.0]	(58.0, 62.0]
1021	1	1	3	0	(254.0, 286.0]	(58.0, 62.0]
1022	1	1	2	0	(254.0, 286.0]	(45.0, 53.0]
1023	2	0	2	1	(229.0, 254.0]	(45.0, 53.0]
1024	1	1	3	0	(125.999, 204.0]	(53.0, 58.0]

[1025 rows x 16 columns]

```
[108]: pd.crosstab(df.age_bin, df.target)
```

