

Full Name: Fatema Tabassum Liza

Question1 Ans: The rules are given below

```
if (age[i] >= 54):
    result = ''
    if (gender[i] == 1):
        if (chest_pain[i] >= 2):
            if (cholesterol[i] >= 200):
                if (num_of_vessels[i] == 0):
                    result = "no"
                else:
                    result = "yes"
            else:
                result = "no"
        elif (chest_pain[i] == 2):
            if (cholesterol[i] >= 235):
                if (num_of_vessels[i] == 0):
                    result = "no"
                else:
                    result = "yes"
            else:
                result = "yes"
        else:
            if (cholesterol[i] >= 220):
                result = "yes"
            else:
                result = "no"
    else:
        if (chest_pain[i] >= 4):
            if (cholesterol[i] > 265):
                result = "yes"
            else:
                result = "no"
        else:
            result = "no"
else:
    result = 'no'
```

The above rules will lead to different paths of if-else and by doing that it will generate different combinations of attributes for predicting the result. There are many ways to generate rules for this particular dataset and here is one of the ways we can generate rules to optimize the accuracy. The above rules are giving me 75% training accuracy and 85% testing accuracy on my dataset.

Question2 Ans:

I noticed that for most of the cases where age ≥ 54 and gender == 1, People are more likely to have a heart attack. Again I noticed that the result attribute is very much related to cholesterol. High cholesterol generated more heart attack cases in this data samples. Additionally, chest_pain is also related to heart attack diseases. I analyzed different combinations of rules and after giving much thought, the above rules are generated.

Question3 Ans:

I have separated the test data in the "asn2_testdata.csv" file and the accuracy I got it shown below

```
import numpy as np
count=0
for i in range(len(y_true)):
    if (y_true[i] != y_pred[i]):
        count+=1
acc=((len(y_true)-count) / len(y_true))
print("Acc = {:.2%}".format(acc))
```

Acc = 85.71%

I have also included the test data and train data and python source file in the assignment folder.