**Assignment 1**

**Machine Learning**

**Name: Jasmine I. Sayyad  
PRN: 22510033**

**Batch: T2**

**A.** Generate distributions (gaussian to start with) for male and female heights (1000 samples each). Fix the mean of female heights to 152 cm and male mean height to 166 cm. label the appropriate gender for samples in each of the distribution (M or F)

**Code:** import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from scipy.stats import norm

female\_ht = pd.Series(np.random.normal(152,5,1000))

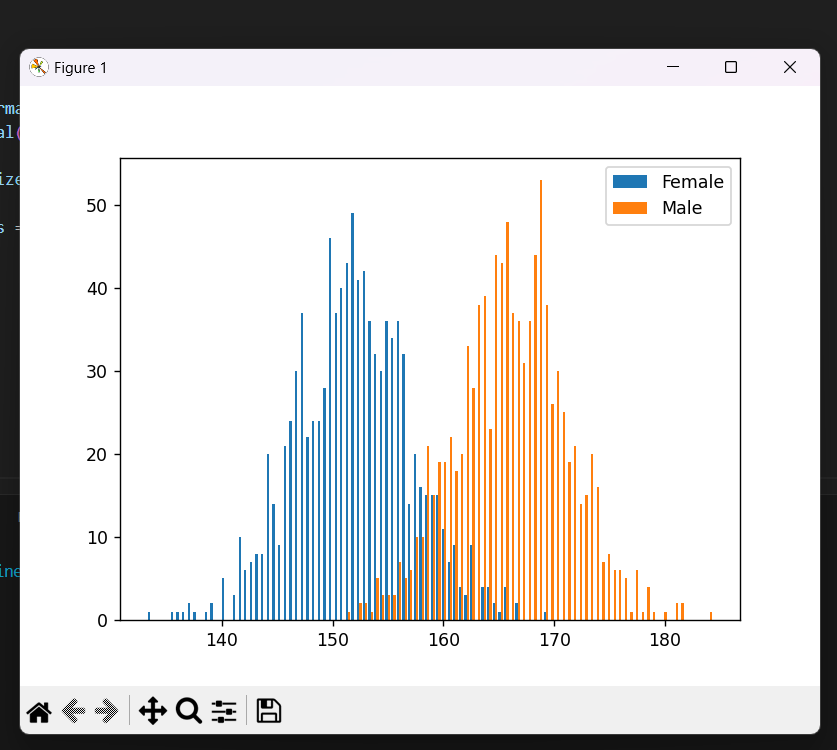
male\_ht = pd.Series(np.random.normal(166,5,1000))

total = female\_ht.size + male\_ht.size

plt.hist([female\_ht, male\_ht], bins = 100 , label = ['Female','Male'])

plt.legend(loc = 'upper right')

plt.show()



**B.** Try classification of gender using following approaches with aim to minimise misclassification.

i. Assign gender based on likelihood calculated from distribution (empirically estimated mean and sd and calculate probability assuming gaussian distributions)

ii. Derive a threshold hight to separate male female.

iii. Quantize the data at scale of 0.5 cm and empirically estimate the likelihood of male female in each segment based on majority

**Code:**

#calculating probability

F\_mean = female\_ht.mean()

M\_mean = male\_ht.mean()

F\_SD = female\_ht.std()

M\_SD = male\_ht.std()

misclassifiedmale = 0

for curr\_ht in male\_ht:

    female\_prob = norm.pdf(curr\_ht,F\_mean,F\_SD)

    male\_prob = norm.pdf(curr\_ht,M\_mean,M\_SD)

    if male\_prob < female\_prob:

        misclassifiedmale+=1

        # print("misplaced male ", curr\_ht)

misclassifiedfemale = 0

for curr\_ht in female\_ht:

    female\_prob = norm.pdf(curr\_ht,F\_mean,F\_SD)

    male\_prob = norm.pdf(curr\_ht,M\_mean,M\_SD)

    if male\_prob > female\_prob:

        misclassifiedfemale+=1

        # print("misplaced female ", curr\_ht)

print("MisplacedMales & misplacedFemales count : ", misclassifiedmale,misclassifiedfemale)

total\_misclassification = misclassifiedfemale + misclassifiedmale

rate = (total\_misclassification/total) \* 100

print("Misclassification rate : ", rate)

#Deriving threshold height to seperate male and female

max\_female\_height = int(female\_ht.max())

min\_male\_height = int(male\_ht.min())

threshold = min\_male\_height

min\_misclassification = 10\*\*15

for curr\_height in range( min\_male\_height, max\_female\_height + 1 ):

    misclassification = 0

    misclassification += sum(male\_ht < curr\_height) + sum(female\_ht > curr\_height)

    if(misclassification < min\_misclassification ):

        min\_misclassification = misclassification

        threshold = curr\_height

print("Threshold Height: ", threshold)

#quantize the data at scale of 0.5 cm and empirically estimate the likelihood of male female in each segment based on majority

def frequency(scale, height):

    quantized\_val = np.floor(height/scale)

    no\_of\_ppl = quantized\_val.value\_counts()

    return no\_of\_ppl

new\_female\_ht = frequency(0.5, female\_ht)

new\_male\_ht = frequency(0.5, male\_ht)

min\_overlap\_height = int(new\_male\_ht.index.min())

max\_overlap\_height = int(new\_female\_ht.index.max())

total\_misplaced = 0

for ht in range(min\_overlap\_height, max\_overlap\_height+1):

    # ht = float(ht)

    Female\_count = new\_female\_ht.get(ht, 0)  # Use .get() for safe access

    Male\_count = new\_male\_ht.get(ht, 0)

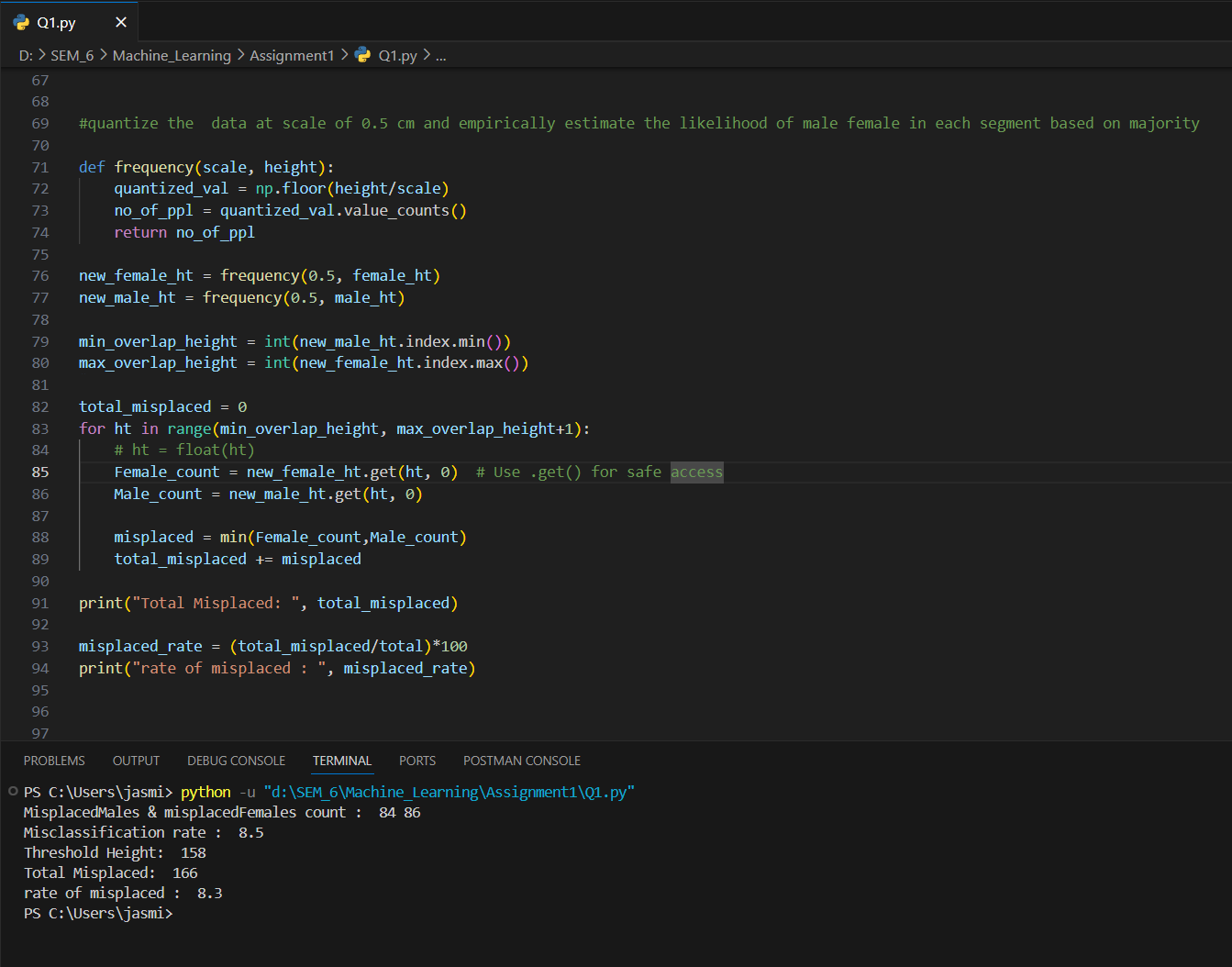
    misplaced = min(Female\_count,Male\_count)

    total\_misplaced += misplaced

print("Total Misplaced: ", total\_misplaced)

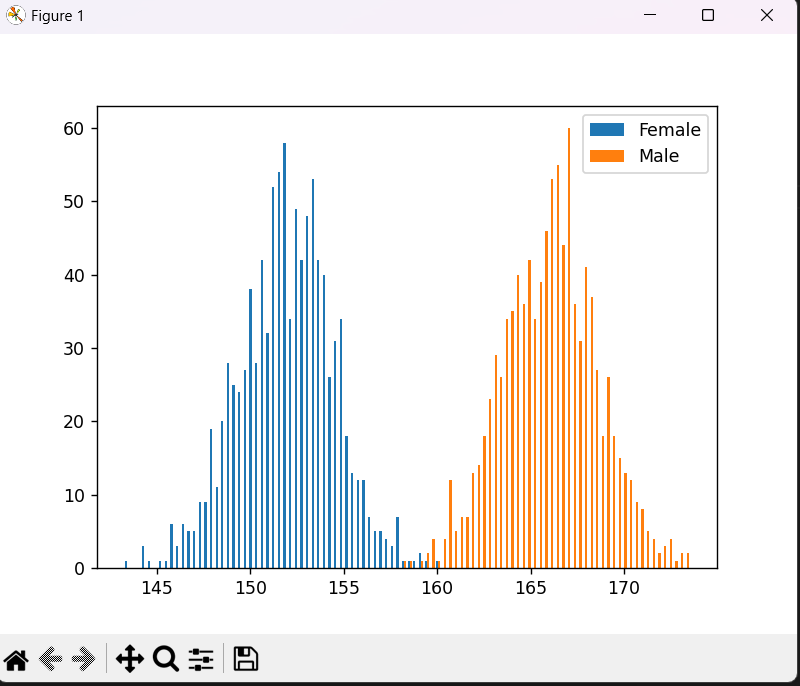
misplaced\_rate = (total\_misplaced/total)\*100

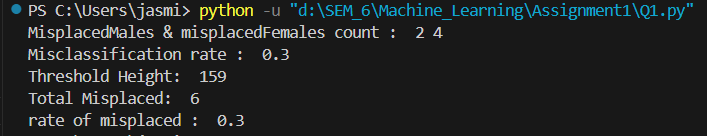
print("rate of misplaced : ", misplaced\_rate)

****

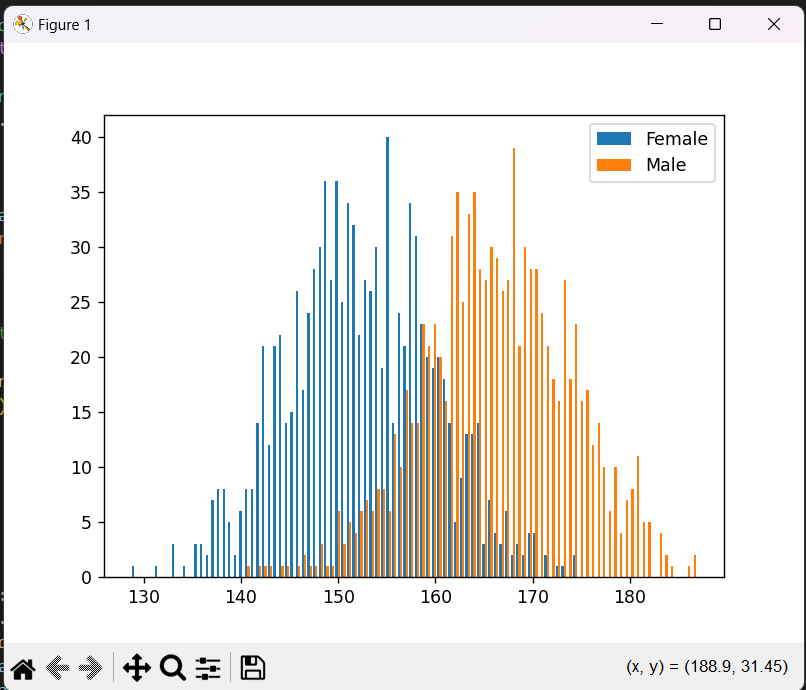
C. Try following values of sd (eg 2.5, 7.5 and 10) repeat 3.a, 3.b, 3.c, 3,d observe impact of change in sd on classification accuracy

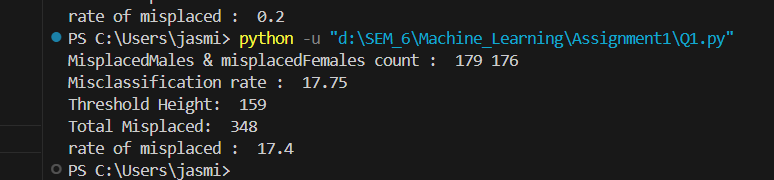
Observations:  
1) if SD = 2.5 as it is less than given SD which is 5, the overlap between male and female heights will be less therefore spread will be less. Therefore less misclassification rate and more accuracy





2) if SD = 7.5 which is greater than 5, the overlapping heights will increase i.e. more congested and therefore the misclassification will also increase, decreasing accuracy

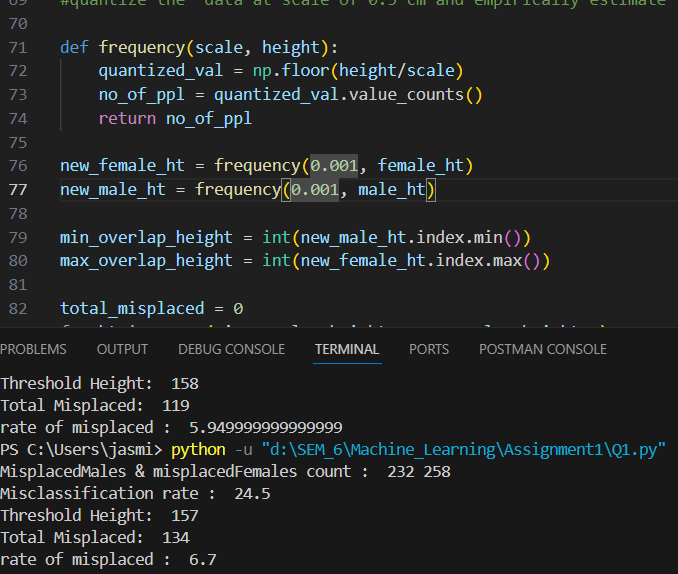




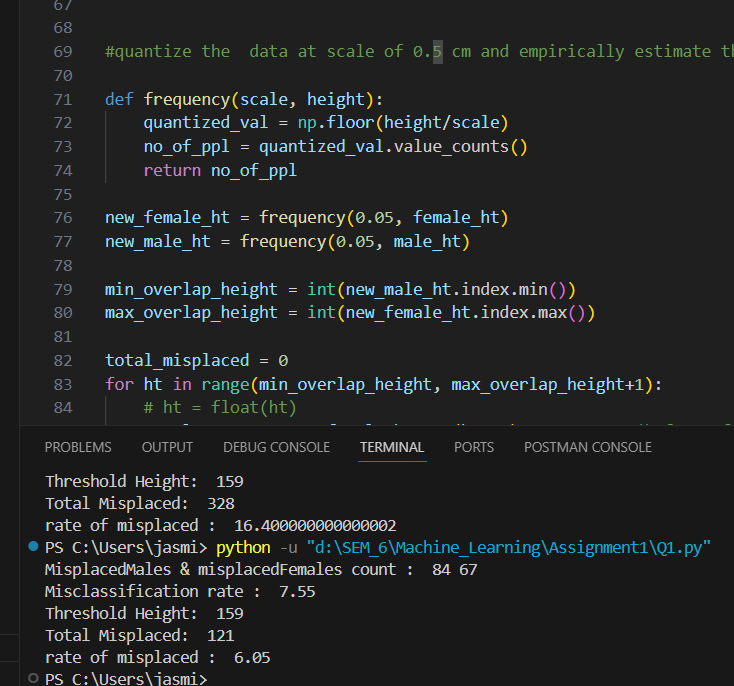
D. Change the quantization interval length (say 0.001, 0.05, 0.1, 0.3, 1, 2, 5,10 cm etc) repeat 3.a, 3.b, 3.c, 3,d observe impact of change in sd on classification accuracy

Observations:

1) If I change quantization from 0.5 to 0.001, no. of bins will increase therefore accuracy will increase. Increased computational complexity. Increased variance due to limited data in each bin i.e. some bins may have large no. of heights some may have less no. of heights



2) If I change it to 0.05 < 0.5 bins will be more , accuracy will be more



3) if I increase it to 5 the number of bins will decrease giving less accuracy but increased computational efficiency

