Batch T4

Practical No. 1

Title of Assignment :

An univariate classifier from first principles

Student Name: Parshwa Herwade

Student PRN: 22510064

1. An univariate classifier from first principles

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)

b. Fix the sd of both the distributions to 5

c. Try classification of gender using following approaches with aim to minimise

misclassification

i. Assign gender based on likelihood calculated from distributions (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

iv. In each of the above cases output a confusion matrix for classification

d. 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

e. 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

CODE:

```
import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from scipy.stats import norm

from sklearn.metrics import confusion_matrix, accuracy_score
```

(Generating height data for males and females)

```
def gen_data(mean, sd, size, label):

   heights = np.random.normal(mean, sd, size)

   return pd.DataFrame({'height': heights, 'label': [label] * size})


# Plotting the height dist. for both males and females

def plot_hist(female, male, bins=50, label=""):

   plt.hist([female, male], bins=bins, label=['Female', 'Male'], alpha=0.7, color=['purple', 'green'])

   plt.title(f'Height Distribution {label}')

   plt.xlabel('Height (cm)')

   plt.ylabel('Frequency')

   plt.legend(loc='upper right')

   plt.show()
```

(Threshold-based classification)

```
def threshold_class(female, male, threshold):

   data = np.concatenate([female, male])

   preds = np.where(data < threshold, 'F', 'M')

   true_labels = np.concatenate([['F'] * len(female), ['M'] * len(male)])

   return true_labels, preds
```

(Prob-based classification)

```
def prob_class(female, male, f_mean, f_sd, m_mean, m_sd):
    def classify(h):
        f_prob = norm.pdf(h, f_mean, f_sd)
        m_prob = norm.pdf(h, m_mean, m_sd)
        return 'F' if f_prob > m_prob else 'M'

    data = np.concatenate([female, male])
    preds = np.array([classify(h) for h in data])
    true_labels = np.concatenate([['F'] * len(female), ['M'] * len(male)])
    return true_labels, preds
```

(**Quantized classification)**

```
def quant_class(female, male, interval):
    def quantize(data):
        return pd.Series(np.floor(data / interval)).value_counts()

    f_quant = quantize(female)
    m_quant = quantize(male)
    intervals = set(f_quant.index).union(set(m_quant.index))

    preds = []
    true_labels = []

    for interval in intervals:
        f_count = f_quant.get(interval, 0)
        m_count = m_quant.get(interval, 0)
        label = 'F' if f_count >= m_count else 'M'
        preds.extend([label] * (f_count + m_count))
        true_labels.extend(['F'] * f_count + ['M'] * m_count)

    return np.array(true_labels), np.array(preds)
```

(Evaluation if the classifier performance)

```
def evaluate(true_labels, preds, label=""):
    cm = confusion_matrix(true_labels, preds, labels=['F', 'M'])
    accuracy = accuracy_score(true_labels, preds)
    print(f"\n{label}")
    print("Confusion Matrix:")
    print(f"\n\t\tPredicted F\tPredicted M\nActual F\t{cm[0, 0]:<5} (TP)\t{cm[0, 1]:<5} (FN)\nActual M\t{cm[1, 0]:<5} (FP)\t{cm[1, 1]:<5} (TN)")
    print(f"\nAccuracy: {accuracy * 100:.2f}%")
    return cm, accuracy
```

(Main function to run other values for experimentation)

```
def run():
    f_mean = 152
    m_mean = 166
    std_devs = [2.5, 5, 7.5, 10]
    intervals = [0.001, 0.05, 0.1, 0.3, 1, 2, 5, 10]
    size = 1000
    results = []

    for sd in std_devs:
        print(f"\n--- SD = {sd} ---")
```

(Generation of synthetic data)

```
female = gen_data(f_mean, sd, size, 'F')['height'].values

male = gen_data(m_mean, sd, size, 'M')['height'].values
```

(Plotting histograms)

```
plot_hist(female, male, label=f'(SD={sd})')
```

(Threshold classification)

```
threshold = (f_mean + m_mean) / 2

true_labels, preds = threshold_class(female, male, threshold)

cm, acc = evaluate(true_labels, preds, label=f"Threshold (Threshold={threshold})")

results.append({"Method": "Threshold", "SD": sd, "Interval": None, "Accuracy": acc})
```

(Probability classification)

```
true_labels, preds = prob_class(female, male, f_mean, sd, m_mean, sd)

cm, acc = evaluate(true_labels, preds, label="Probability Classifier")

results.append({"Method": "Probability", "SD": sd, "Interval": None, "Accuracy": acc})
```

(Quantized classification)

```
for interval in intervals:

    true_labels, preds = quant_class(female, male, interval)

    cm, acc = evaluate(true_labels, preds, label=f"Quantized (Interval={interval})")

    results.append({"Method": "Quantized", "SD": sd, "Interval": interval, "Accuracy": acc})
```

(Savin results to .csv file)

```
df = pd.DataFrame(results)
```

```
    df.to_csv('results.csv', index=False)
    print("\nResults saved to 'results.csv'.")


run()
```

OBSERVATIONS:

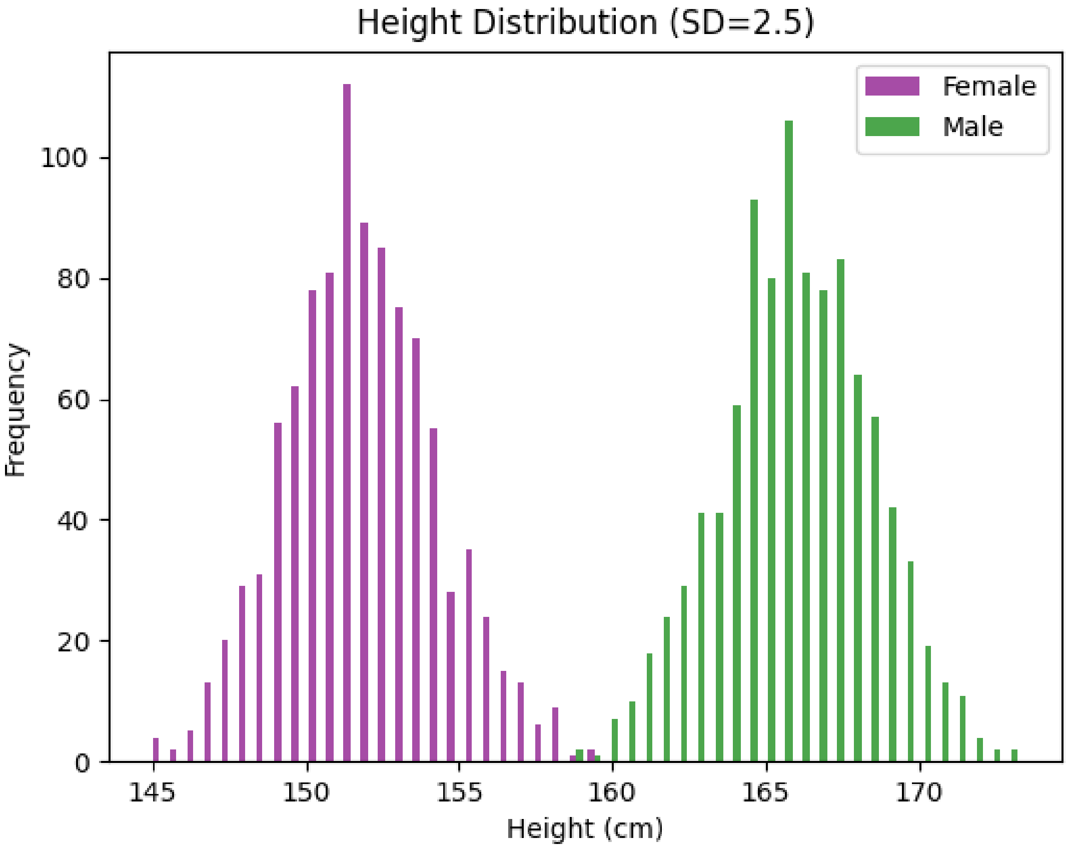
As the standard deviation (SD) increases, the accuracy of the classification methods tends to decrease. This happens because the distributions of male and female heights start overlapping more. With a smaller SD (like 2.5), the difference between the two groups is clearer, making classifiers like the Threshold-based and Probability-based methods more effective. However, as the SD increases (such as to 10), the distributions spread out, making it harder to distinguish between the genders. The Threshold-based method struggles more with this overlap, while the Probability-based method fares slightly better, as it considers the entire distribution, though its accuracy still declines as SD grows.

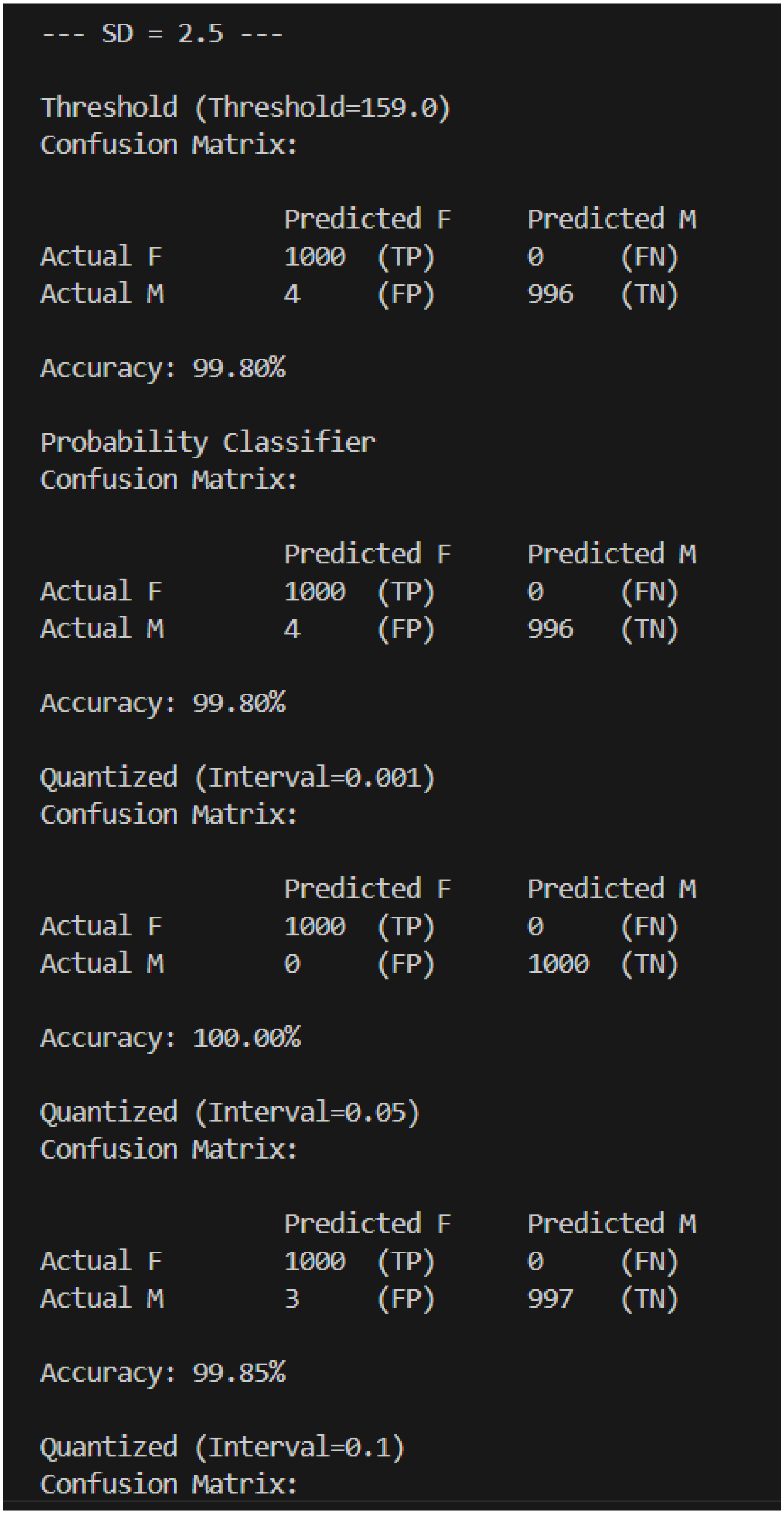
The Quantization method, which divides the data into intervals, is less sensitive to SD. It performs reasonably well even with larger SD values. This method works best when the intervals are not too small (like 0.001 cm) or too large (like 5 cm). If the intervals are too small, the method may overfit the data, while larger intervals result in rougher outcomes.

The confusion matrices reveal that as SD increases, misclassifications become more frequent. The Threshold method tends to have more False Negatives, while the Probability-based method results in more False Positives. The Quantization method remains more stable, though its performance still depends on the SD and interval size.

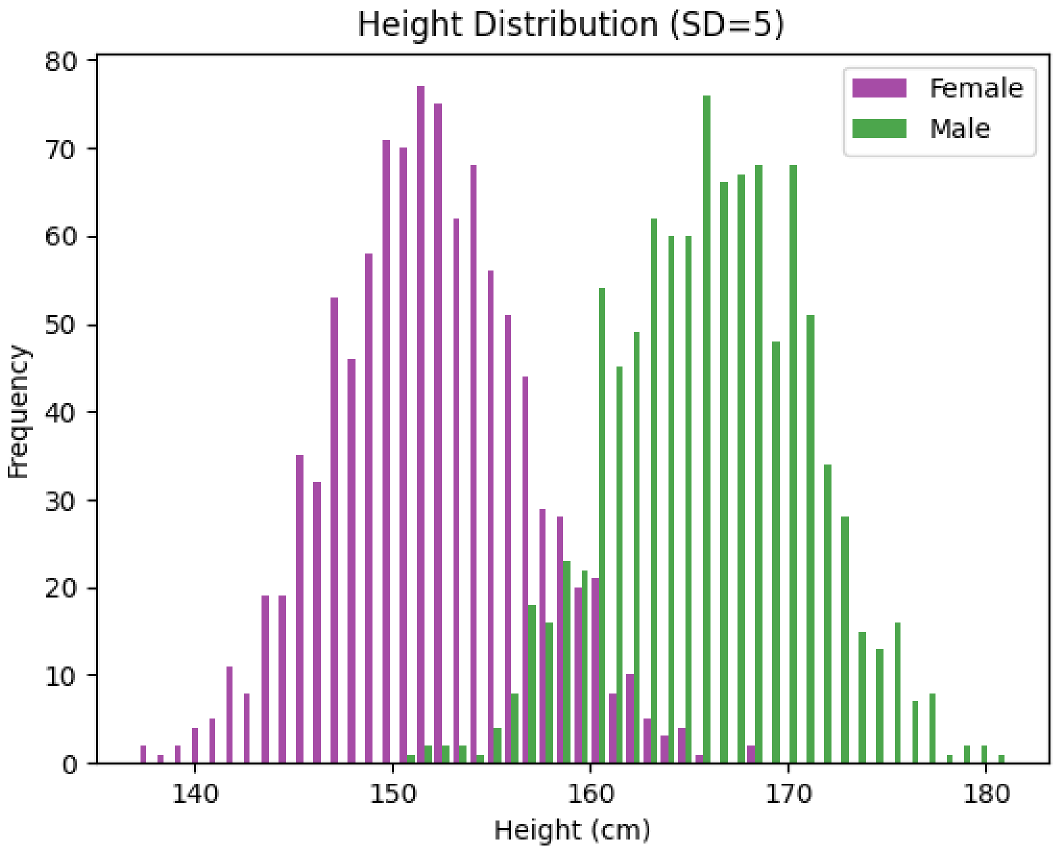
Overall, accuracy improves with lower SD values, and the Probability-based method handles SD changes most effectively. In the future, tweaking quantization intervals or utilizing machine learning models could help with cases where the distributions overlap more.

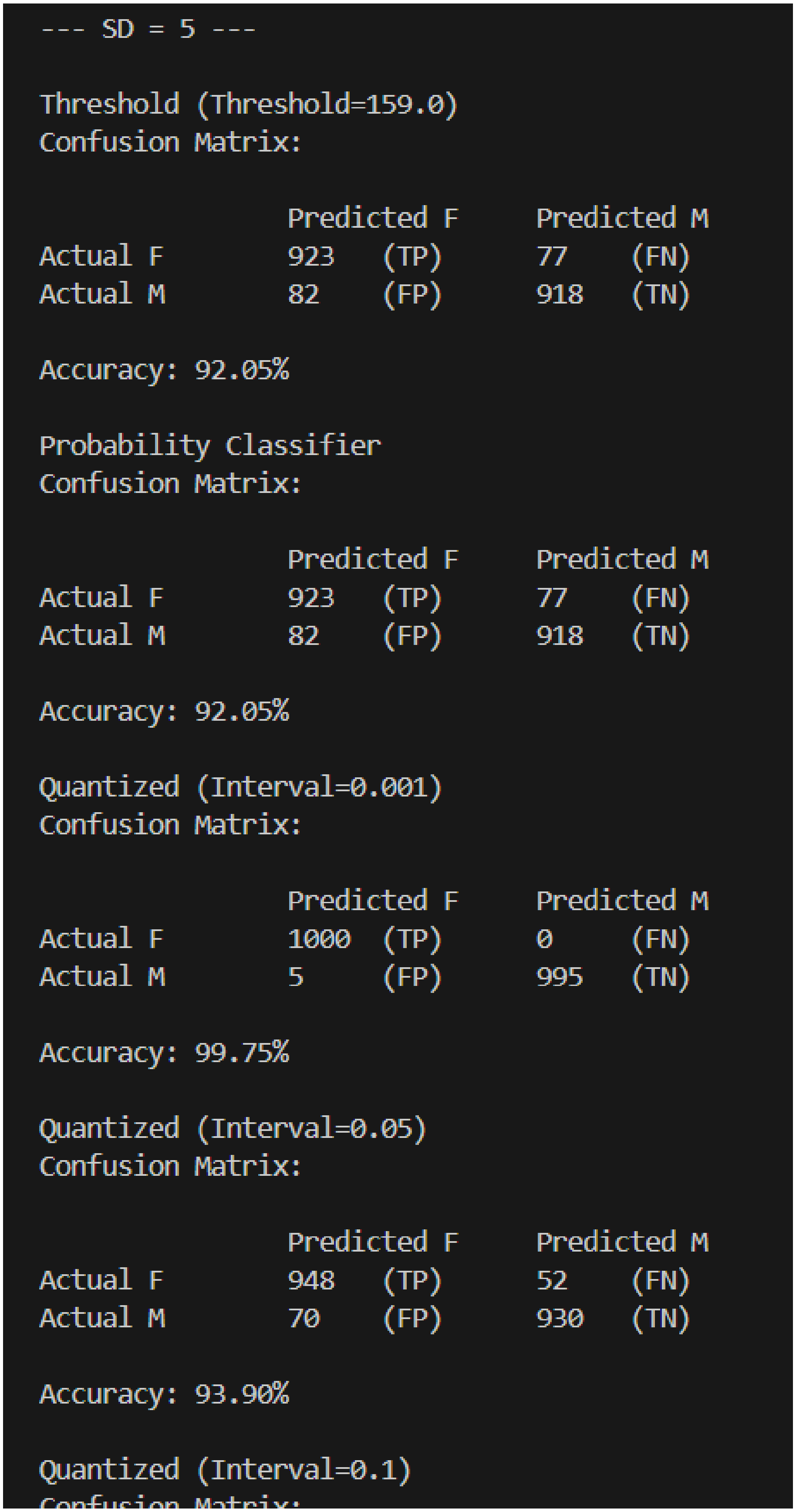
HISTOGRAMS:


Height Distribution (SD=2.5)
Female
Male
Frequency
0
20
40
60
80
100
145
150
155
160
165
170
Height (cm)

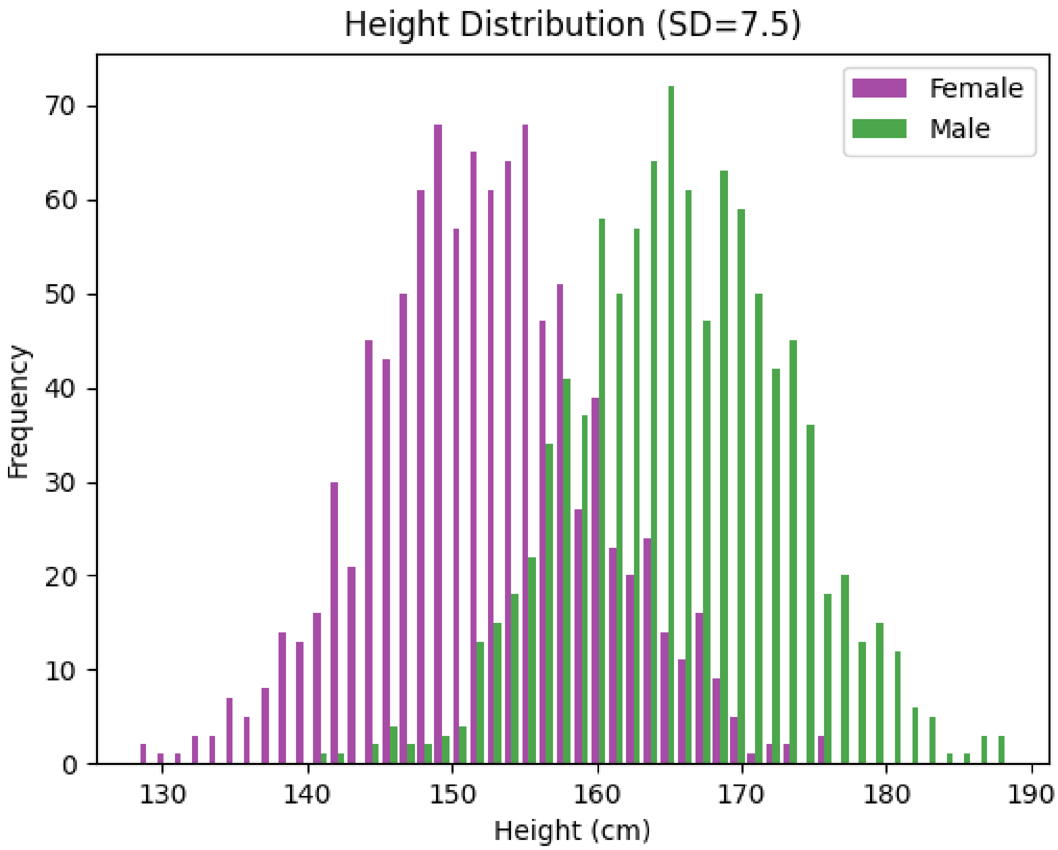


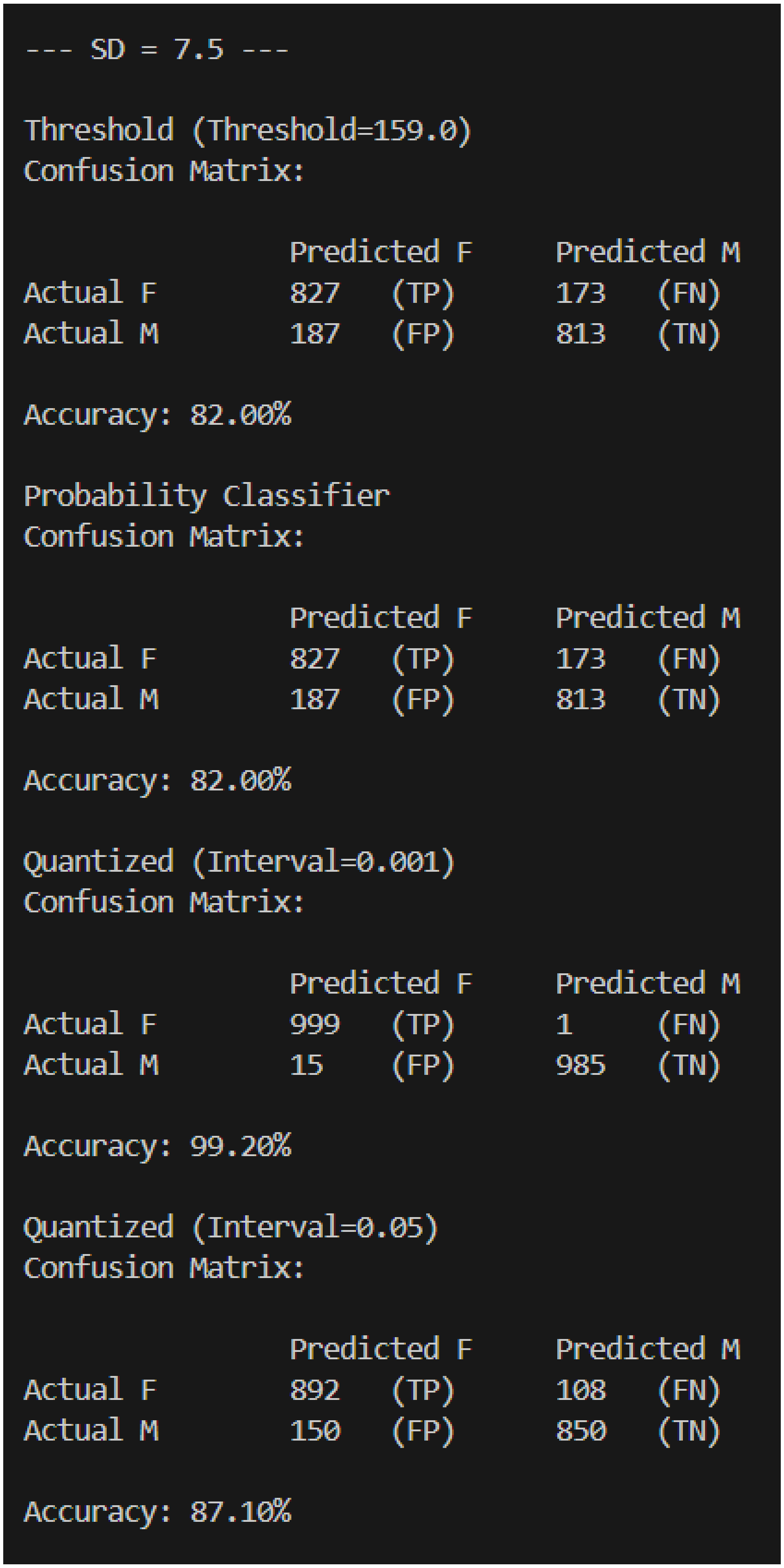
```
--- SD = 2.5 ---

Threshold (Threshold=159.0)
Confusion Matrix:

                Predicted F     Predicted M
Actual F        1000  (TP)      0     (FN)
Actual M        4     (FP)      996   (TN)

Accuracy: 99.80%

Probability Classifier
Confusion Matrix:

                Predicted F     Predicted M
Actual F        1000  (TP)      0     (FN)
Actual M        4     (FP)      996   (TN)

Accuracy: 99.80%

Quantized (Interval=0.001)
Confusion Matrix:

                Predicted F     Predicted M
Actual F        1000  (TP)      0     (FN)
Actual M        0     (FP)      1000  (TN)

Accuracy: 100.00%

Quantized (Interval=0.05)
Confusion Matrix:

                Predicted F     Predicted M
Actual F        1000  (TP)      0     (FN)
Actual M        3     (FP)      997   (TN)

Accuracy: 99.85%

Quantized (Interval=0.1)
Confusion Matrix:
```


Height Distribution (SD=5)
Female
Male
Frequency
0
10
20
30
40
50
60
70
80
140
150
160
170
180
Height (cm)

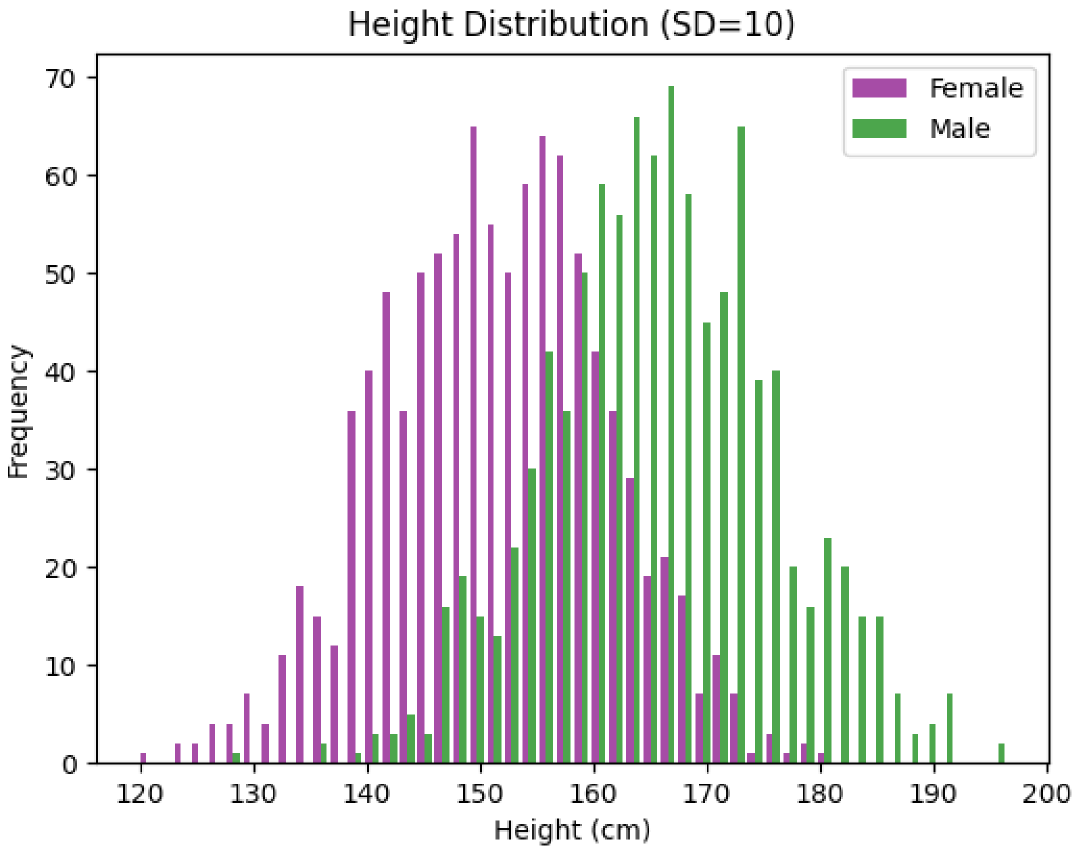


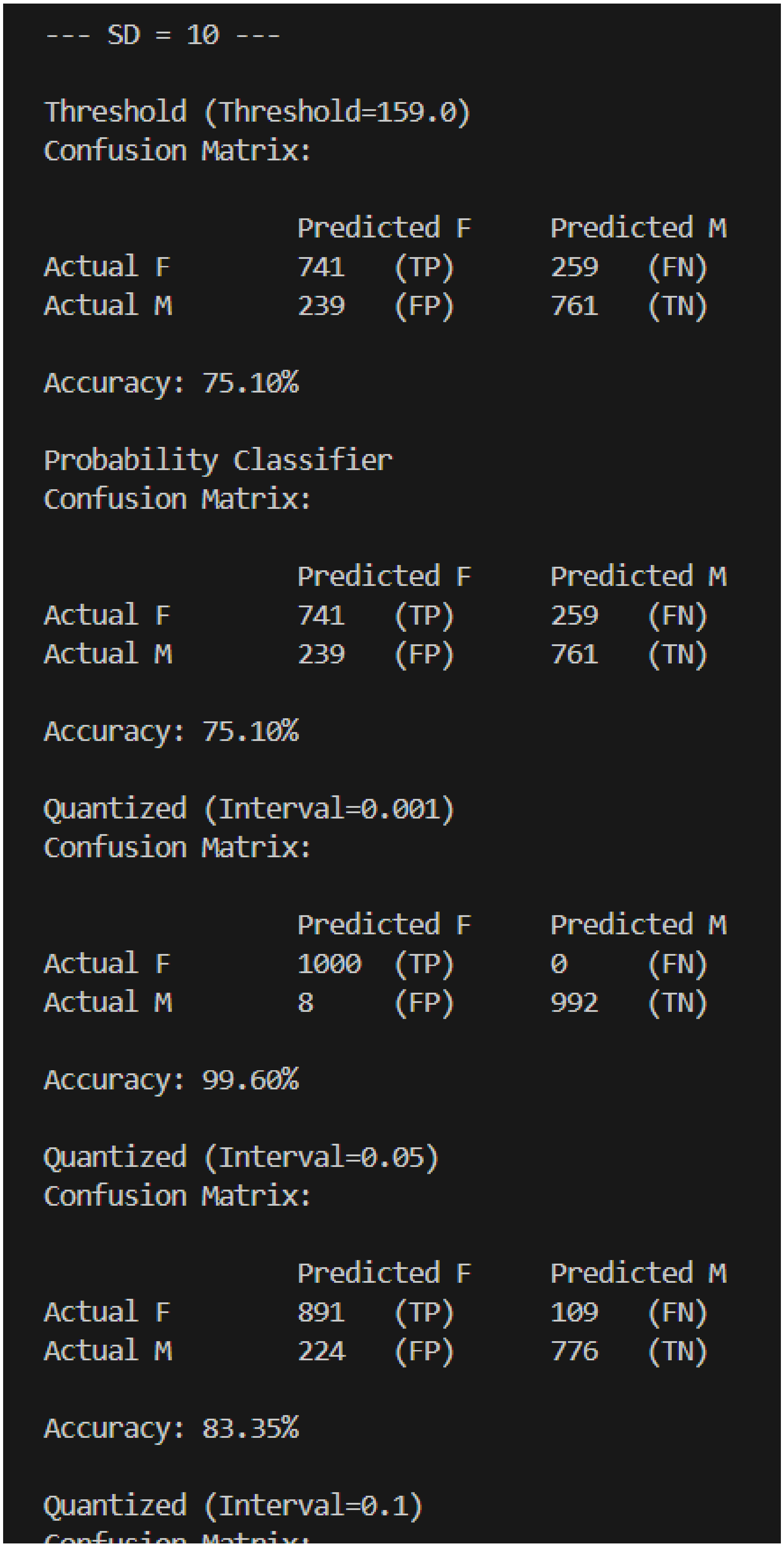
```
--- SD = 5 ---

Threshold (Threshold=159.0)
Confusion Matrix:

                Predicted F     Predicted M
Actual F        923   (TP)      77    (FN)
Actual M        82    (FP)      918   (TN)

Accuracy: 92.05%

Probability Classifier
Confusion Matrix:

                Predicted F     Predicted M
Actual F        923   (TP)      77    (FN)
Actual M        82    (FP)      918   (TN)

Accuracy: 92.05%

Quantized (Interval=0.001)
Confusion Matrix:

                Predicted F     Predicted M
Actual F        1000  (TP)      0     (FN)
Actual M        5     (FP)      995   (TN)

Accuracy: 99.75%

Quantized (Interval=0.05)
Confusion Matrix:

                Predicted F     Predicted M
Actual F        948   (TP)      52    (FN)
Actual M        70    (FP)      930   (TN)

Accuracy: 93.90%

Quantized (Interval=0.1)
Confusion Matrix:
```


Height Distribution (SD=7.5)
Female
Male
Frequency
Height (cm)
0
10
20
30
40
50
60
70
130
140
150
160
170
180
190



```
--- SD = 7.5 ---

Threshold (Threshold=159.0)
Confusion Matrix:

                Predicted F     Predicted M
Actual F        827   (TP)      173   (FN)
Actual M        187   (FP)      813   (TN)

Accuracy: 82.00%

Probability Classifier
Confusion Matrix:

                Predicted F     Predicted M
Actual F        827   (TP)      173   (FN)
Actual M        187   (FP)      813   (TN)

Accuracy: 82.00%

Quantized (Interval=0.001)
Confusion Matrix:

                Predicted F     Predicted M
Actual F        999   (TP)      1     (FN)
Actual M        15    (FP)      985   (TN)

Accuracy: 99.20%

Quantized (Interval=0.05)
Confusion Matrix:

                Predicted F     Predicted M
Actual F        892   (TP)      108   (FN)
Actual M        150   (FP)      850   (TN)

Accuracy: 87.10%
```


Height Distribution (SD=10)
Female
Male
Frequency
0
10
20
30
40
50
60
70
120
130
140
150
160
170
180
190
200
Height (cm)



```
--- SD = 10 ---

Threshold (Threshold=159.0)
Confusion Matrix:

                Predicted F     Predicted M
Actual F        741   (TP)      259   (FN)
Actual M        239   (FP)      761   (TN)

Accuracy: 75.10%

Probability Classifier
Confusion Matrix:

                Predicted F     Predicted M
Actual F        741   (TP)      259   (FN)
Actual M        239   (FP)      761   (TN)

Accuracy: 75.10%

Quantized (Interval=0.001)
Confusion Matrix:

                Predicted F     Predicted M
Actual F        1000  (TP)      0     (FN)
Actual M        8     (FP)      992   (TN)

Accuracy: 99.60%

Quantized (Interval=0.05)
Confusion Matrix:

                Predicted F     Predicted M
Actual F        891   (TP)      109   (FN)
Actual M        224   (FP)      776   (TN)

Accuracy: 83.35%

Quantized (Interval=0.1)
Confusion Matrix:
```

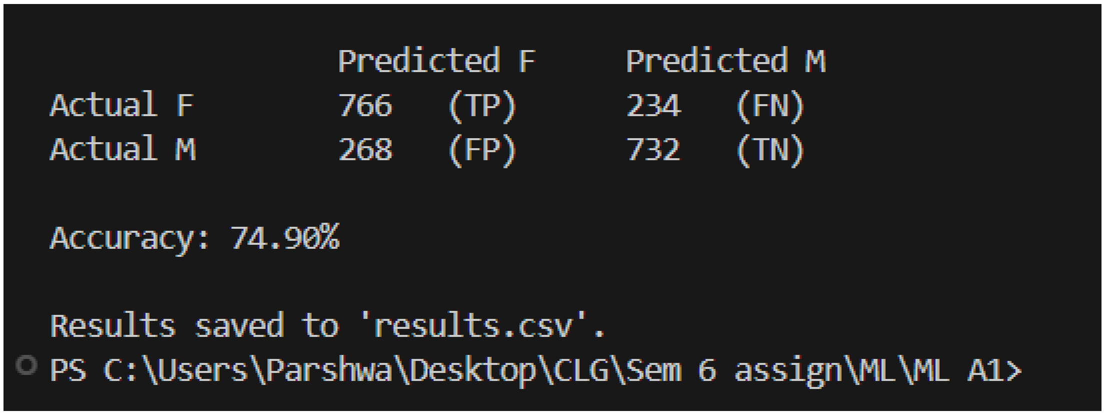
The accuracy for diff S.D's is observed as we can seee int the above screenshots.

With that the confusion matrix having True and false positives and negatives for each are observed for both male and female therby giving us the accuracy and also the accuracy goes on decreasing gradually for the S.D. and for the increment in intervals as asked for us to do in the question 0.001 , 0.1 ,1 and so on…..

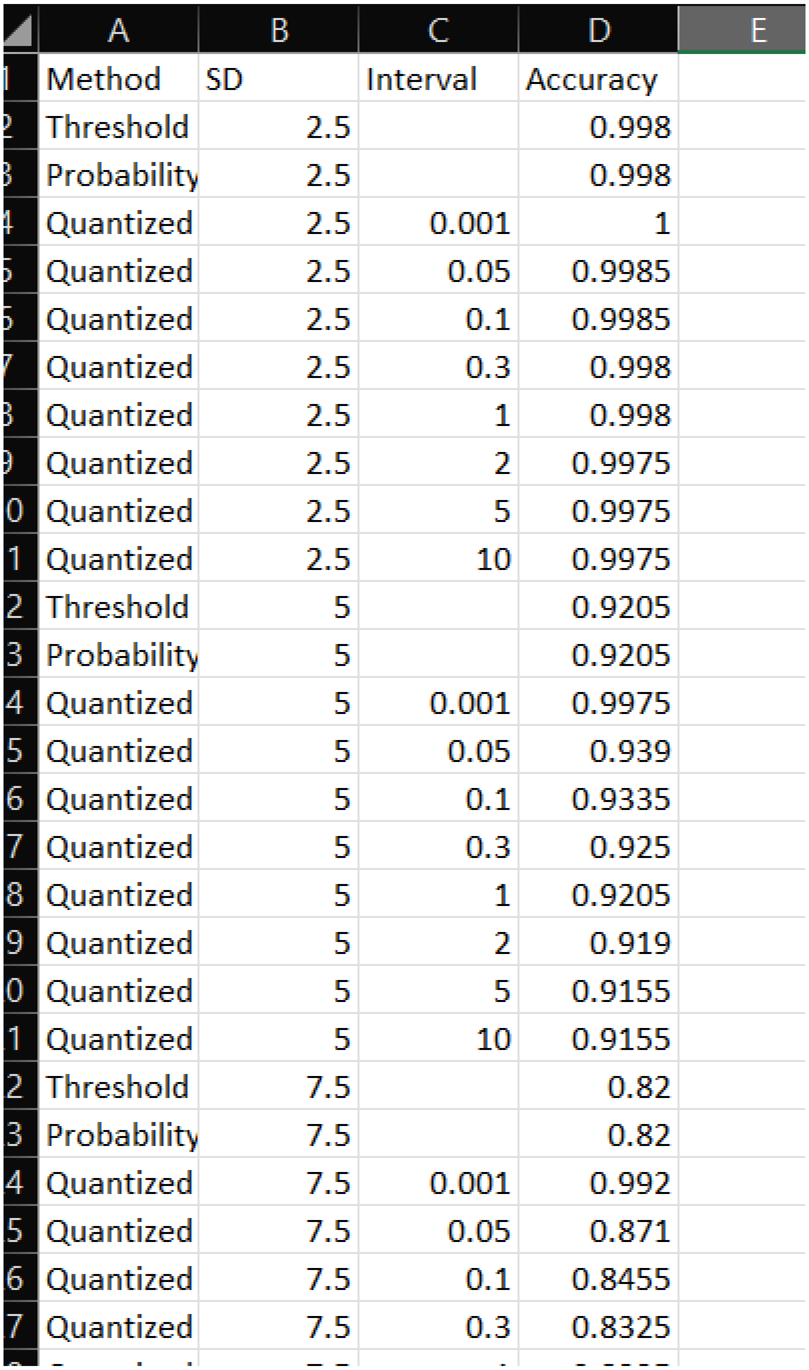
As you can see as the S.D. inceases the and the scale increase too the last image for S.D.=10 tells us most of the points lie in the middle and some points are far away thereby giving us the bell shaped kinda shape

To the left and right there is a thing observed that is there is scarcity of points as we move form the middle portion.

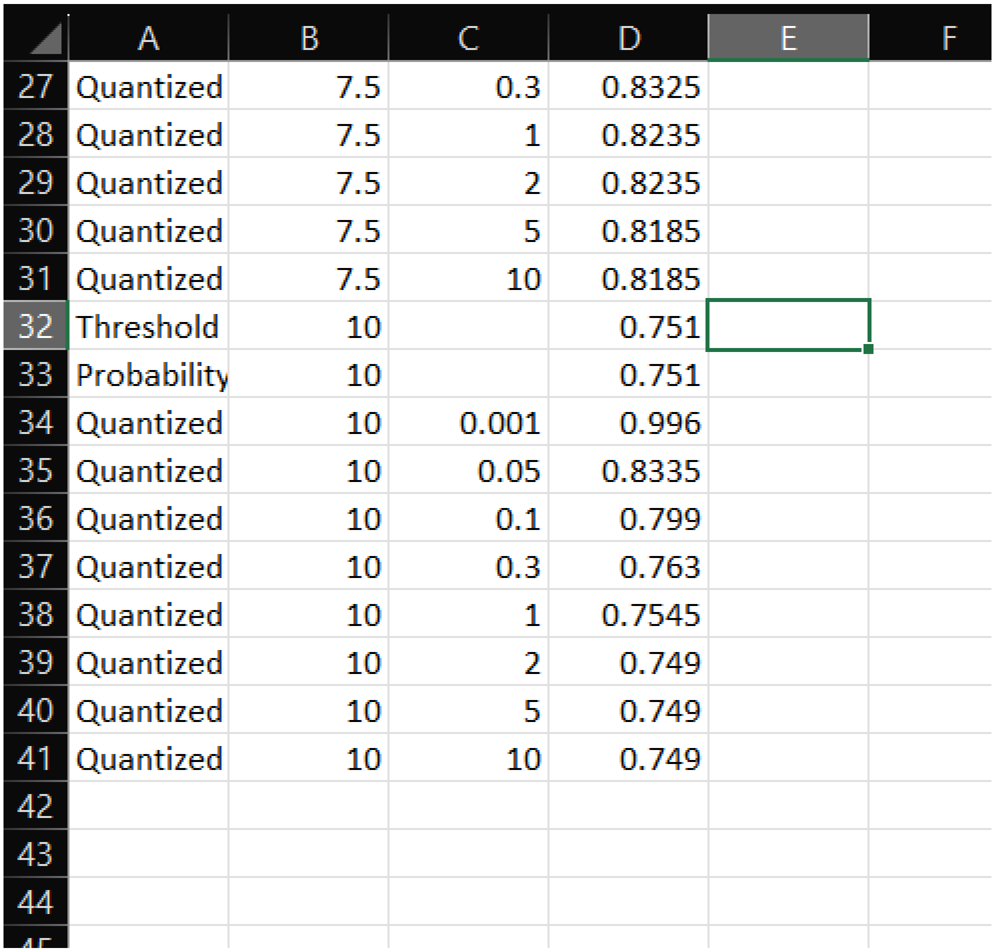
And at last the Matrix Classifier results are stored in .csv file :



```
                Predicted F     Predicted M
Actual F        766   (TP)      234   (FN)
Actual M        268   (FP)      732   (TN)

Accuracy: 74.90%

Results saved to 'results.csv'.
PS C:\Users\Parshwa\Desktop\CLG\Sem 6 assign\ML\ML A1>
```



Method	SD	Interval	Accuracy
Threshold	2.5		0.998
Probability	2.5		0.998
Quantized	2.5	0.001	1
Quantized	2.5	0.05	0.9985
Quantized	2.5	0.1	0.9985
Quantized	2.5	0.3	0.998
Quantized	2.5	1	0.998
Quantized	2.5	2	0.9975
Quantized	2.5	5	0.9975
Quantized	2.5	10	0.9975
Threshold	5		0.9205
Probability	5		0.9205
Quantized	5	0.001	0.9975
Quantized	5	0.05	0.939
Quantized	5	0.1	0.9335
Quantized	5	0.3	0.925
Quantized	5	1	0.9205
Quantized	5	2	0.919
Quantized	5	5	0.9155
Quantized	5	10	0.9155
Threshold	7.5		0.82
Probability	7.5		0.82
Quantized	7.5	0.001	0.992
Quantized	7.5	0.05	0.871
Quantized	7.5	0.1	0.8455
Quantized	7.5	0.3	0.8325



	A	B	C	D	E	F
27	Quantized	7.5	0.3	0.8325		
28	Quantized	7.5	1	0.8235		
29	Quantized	7.5	2	0.8235		
30	Quantized	7.5	5	0.8185		
31	Quantized	7.5	10	0.8185		
32	Threshold	10		0.751		
33	Probability	10		0.751		
34	Quantized	10	0.001	0.996		
35	Quantized	10	0.05	0.8335		
36	Quantized	10	0.1	0.799		
37	Quantized	10	0.3	0.763		
38	Quantized	10	1	0.7545		
39	Quantized	10	2	0.749		
40	Quantized	10	5	0.749		
41	Quantized	10	10	0.749		
42						
43						
44						