Amartejas Manjunath
1001742606
Data Mining

Assignment 2

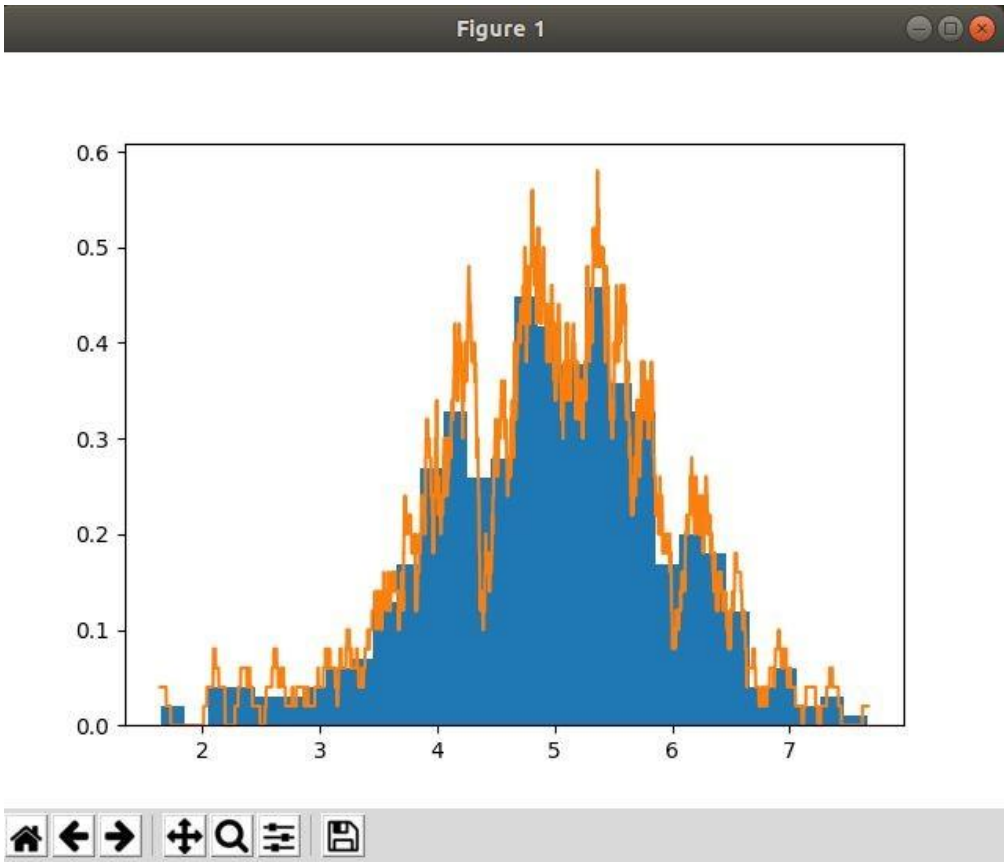
Problem 1

1.The 1D Random Gaussian data is generated using Sigma1 * np.random.randn(500) + mean1

2. The capital **X** is assumed as a discrete value from the min of x to the max of x with a step count of 0.001.

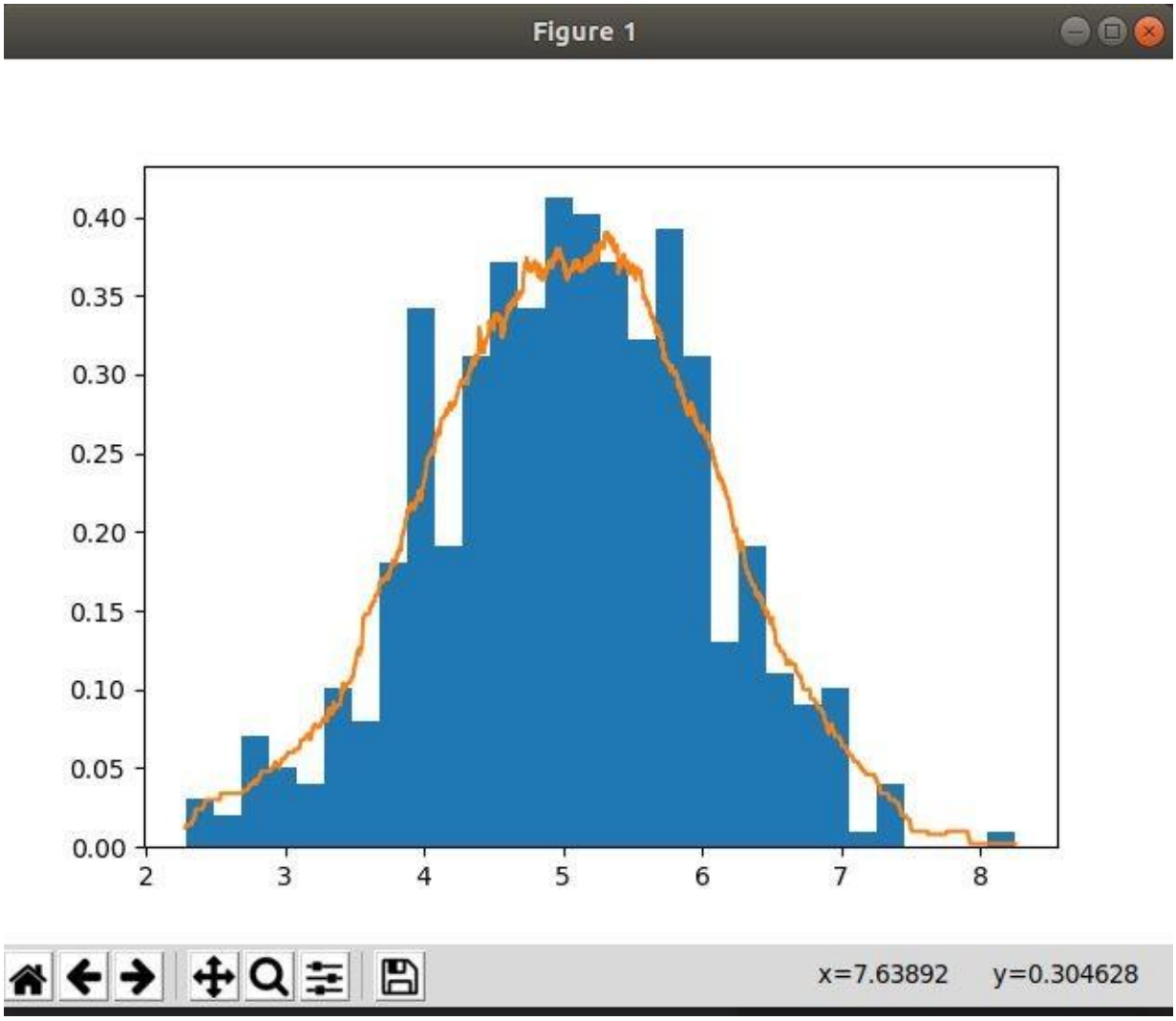
3. The h values are iterated over the entire file.

4. For part 2 of the question, the x value generated is used in the Kernel density estimation formula.

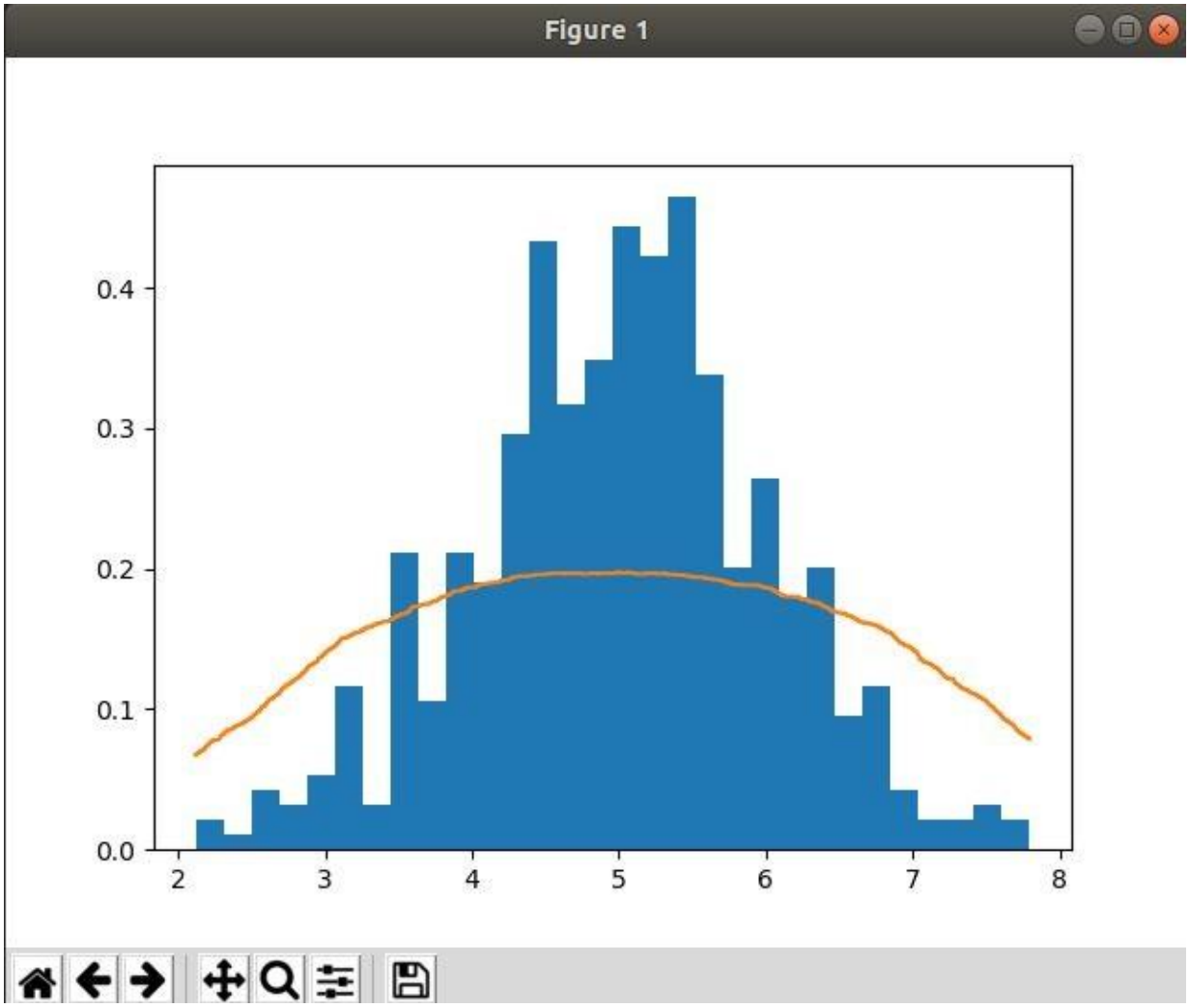


This plot is generated when mean = 5 sigma = 1 and h = 0.1

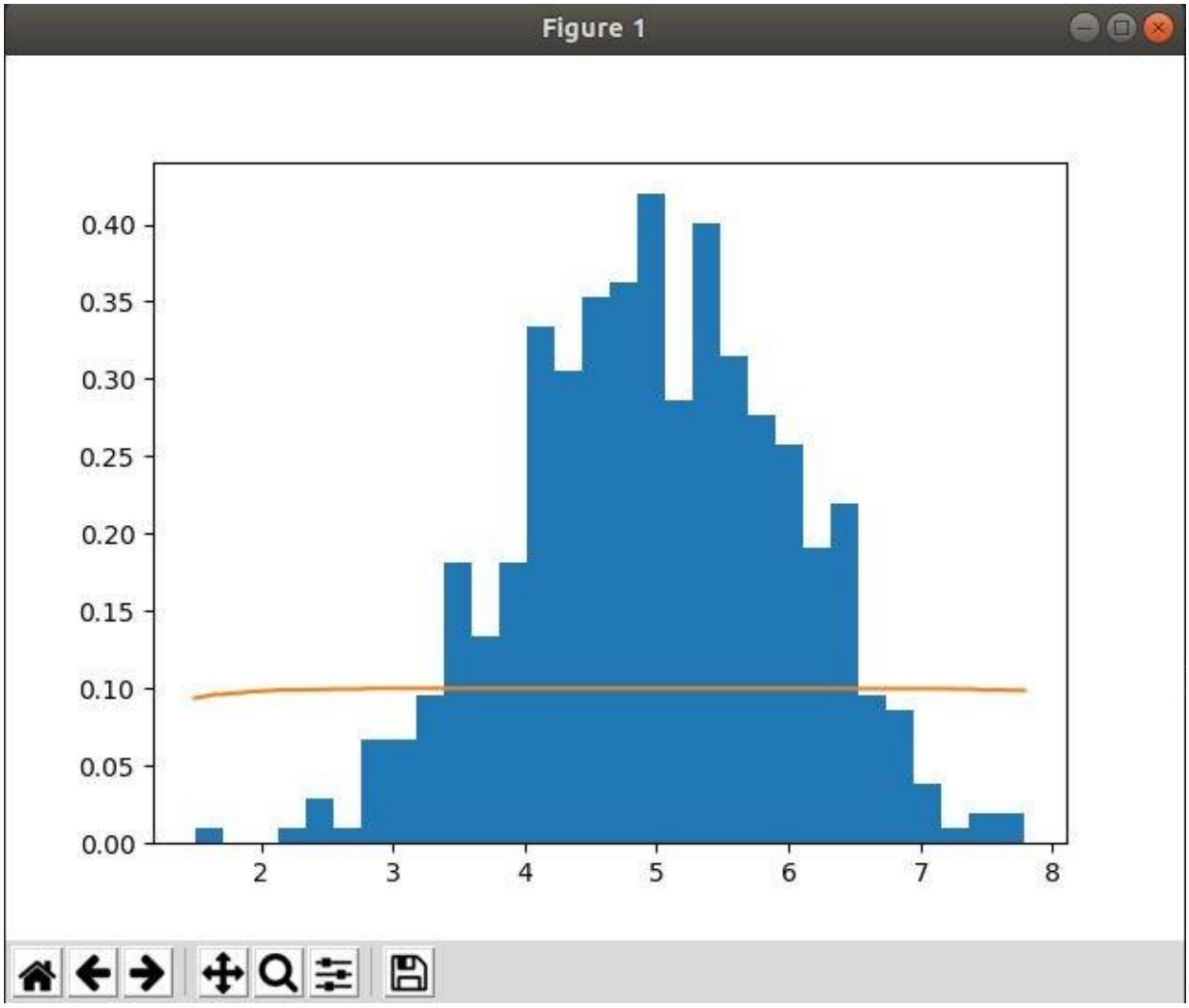
To get the next plots, we need to exit the generated plot.



This plot is generated when mean = 5 sigma = 1 and h = 1



This plot is generated when mean = 5 sigma = 1 and h = 5

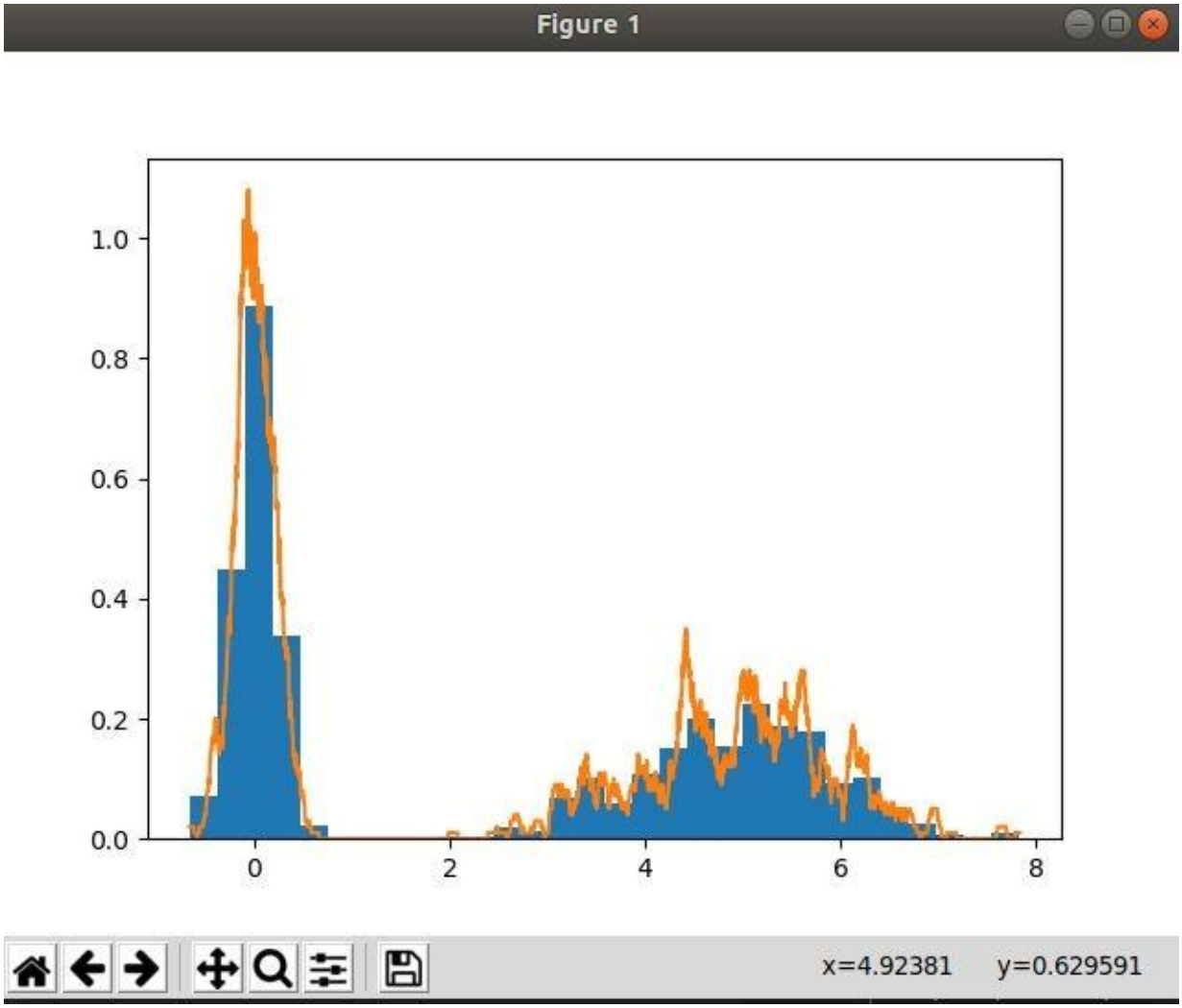


This plot is generated when mean = 5 sigma = 1 and h = 10

5. By uncommenting two lines in the file.(Line 15 and 16)

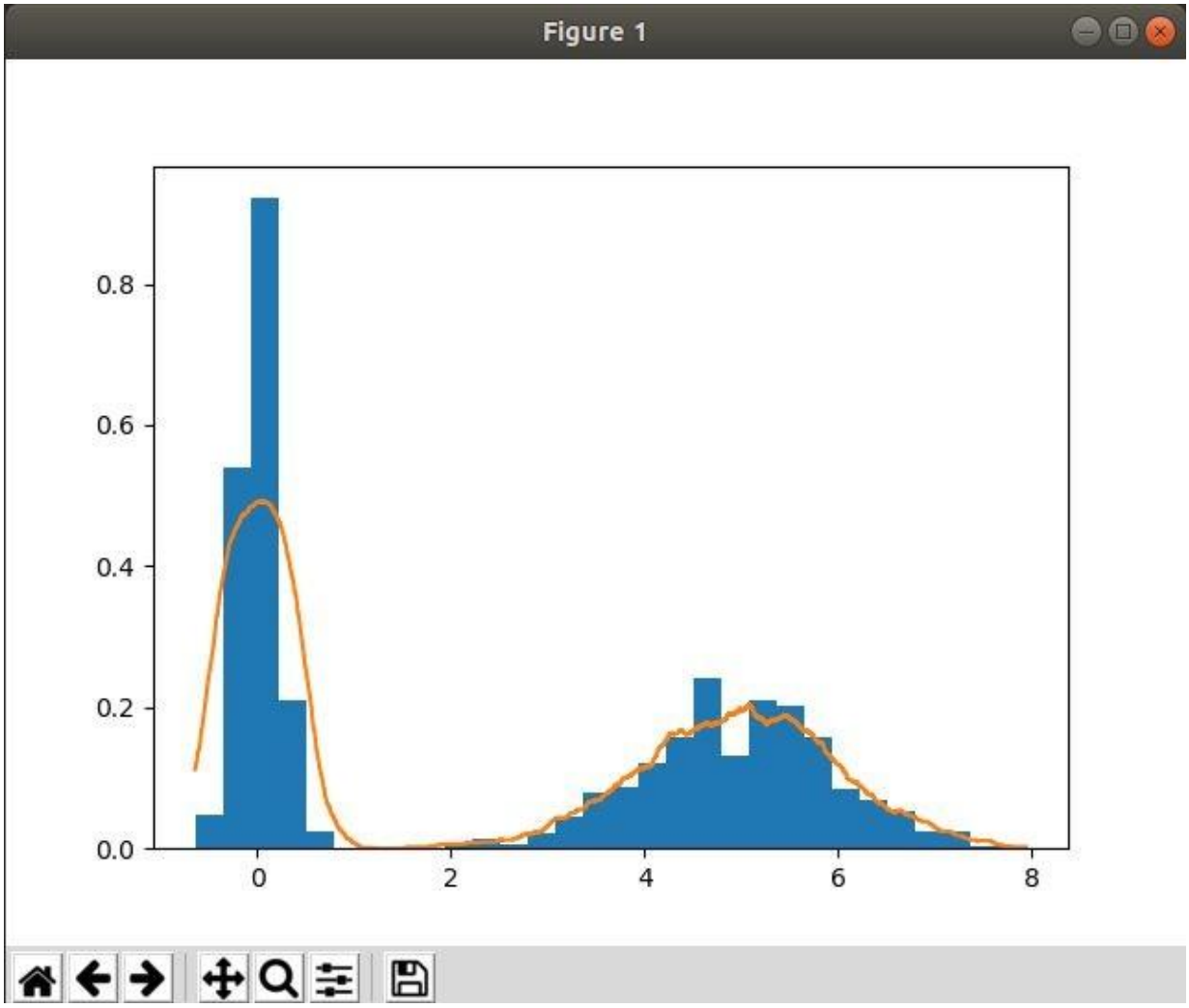
```
14      x = Sigma1 * np.random.randn(500) + mean1
15      #y = Sigma2 * np.random.randn(500) + mean2                    # 2nd set of Gaussian Data.
16      #x = np.concatenate((x, y))                                   # Concatenating the 2 sets of Gaussian Data.
17      X = np.arange(min(x),max(x) , 0.001)
```

6. The following graphs are generated after uncommenting the two lines.

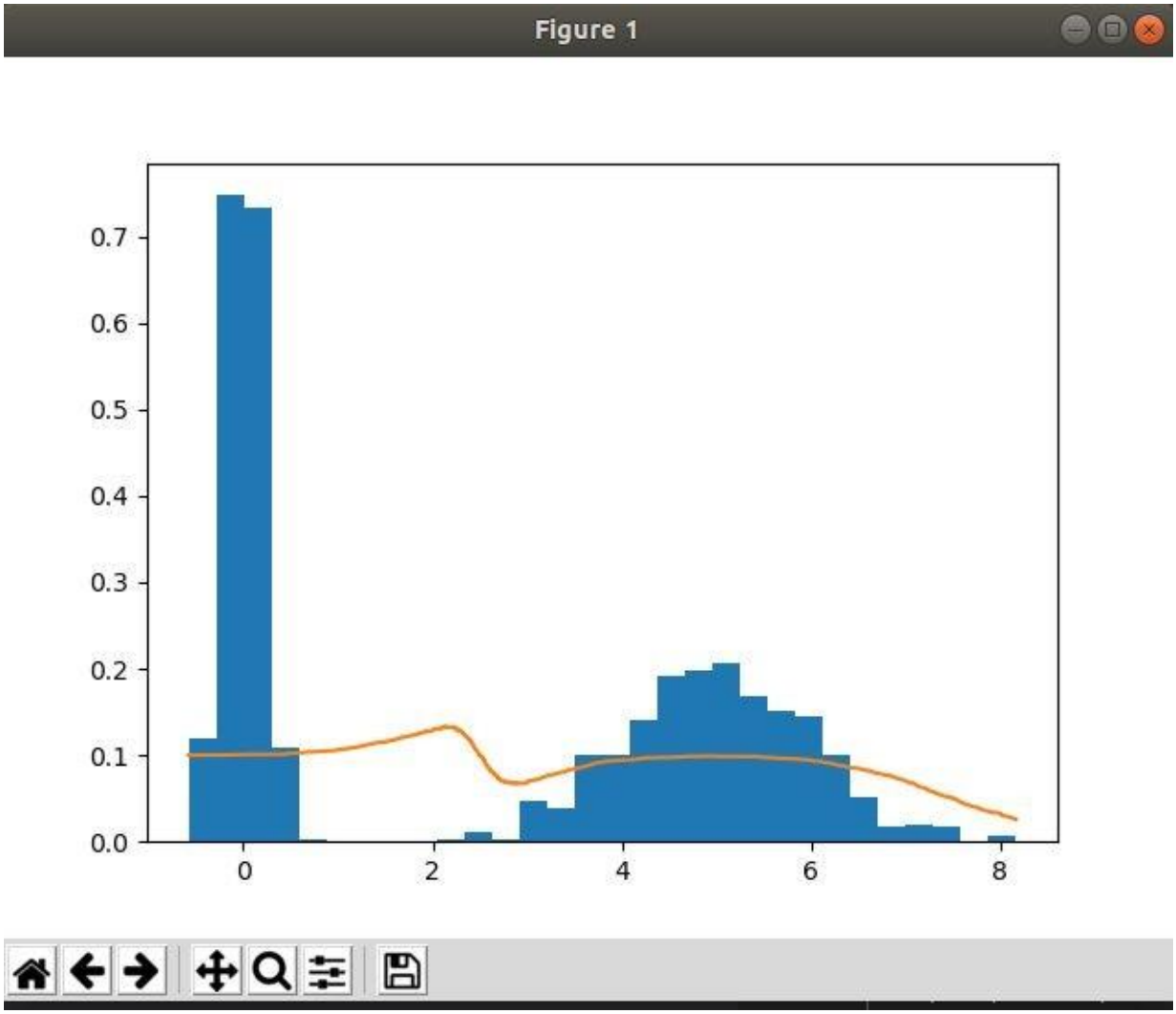


This plot is generated by combining two data sets, where the mean1 = 5, sigma1 = 1 and mean1 = 0, sigma1 = 2 and h =0.1

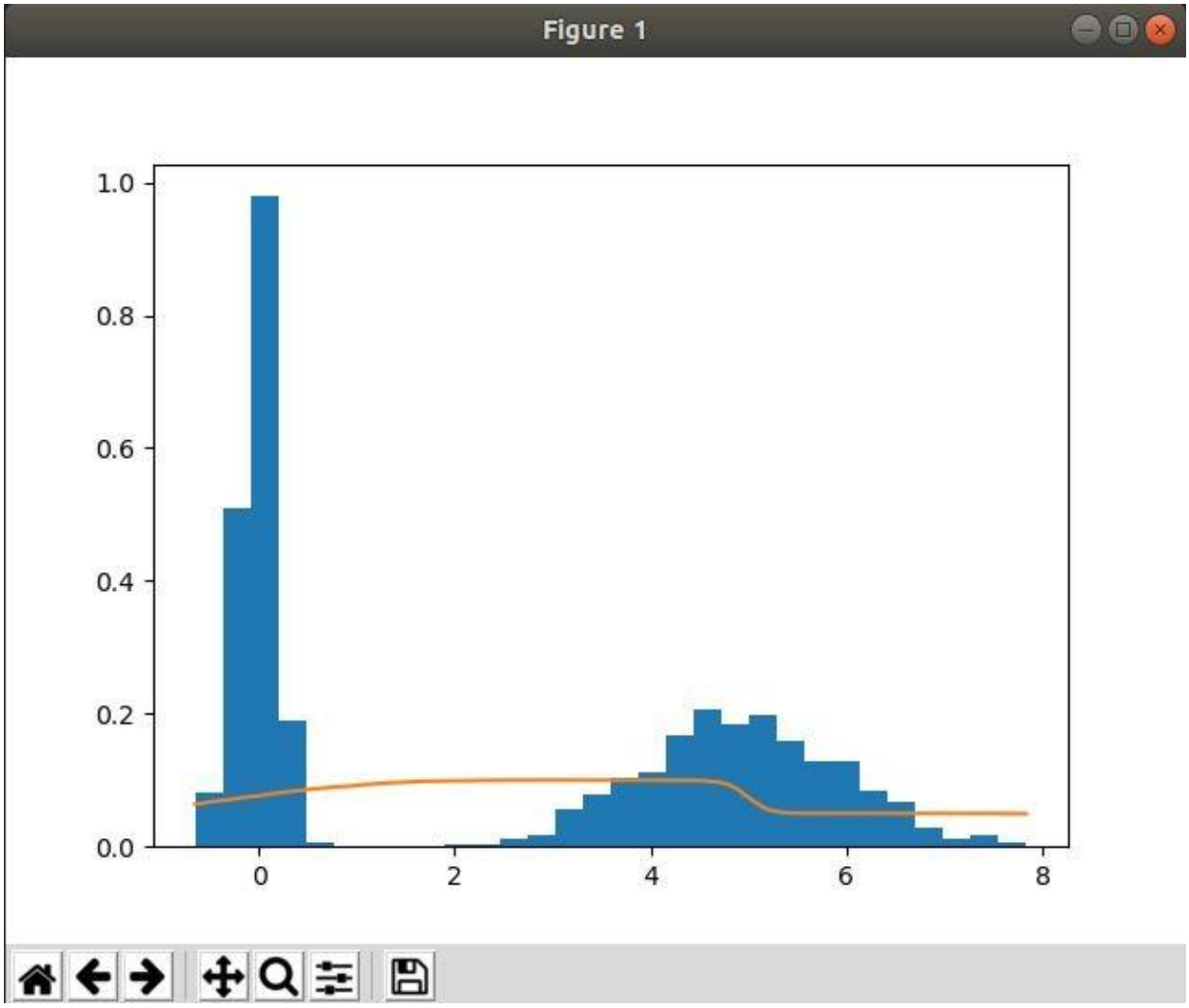
To get the next plots, we need to exit the generated plot.



This plot is generated by combining two data sets, where the mean1 = 5, sigma1 = 1 and mean1 = 0, sigma1 = 2 and h =1



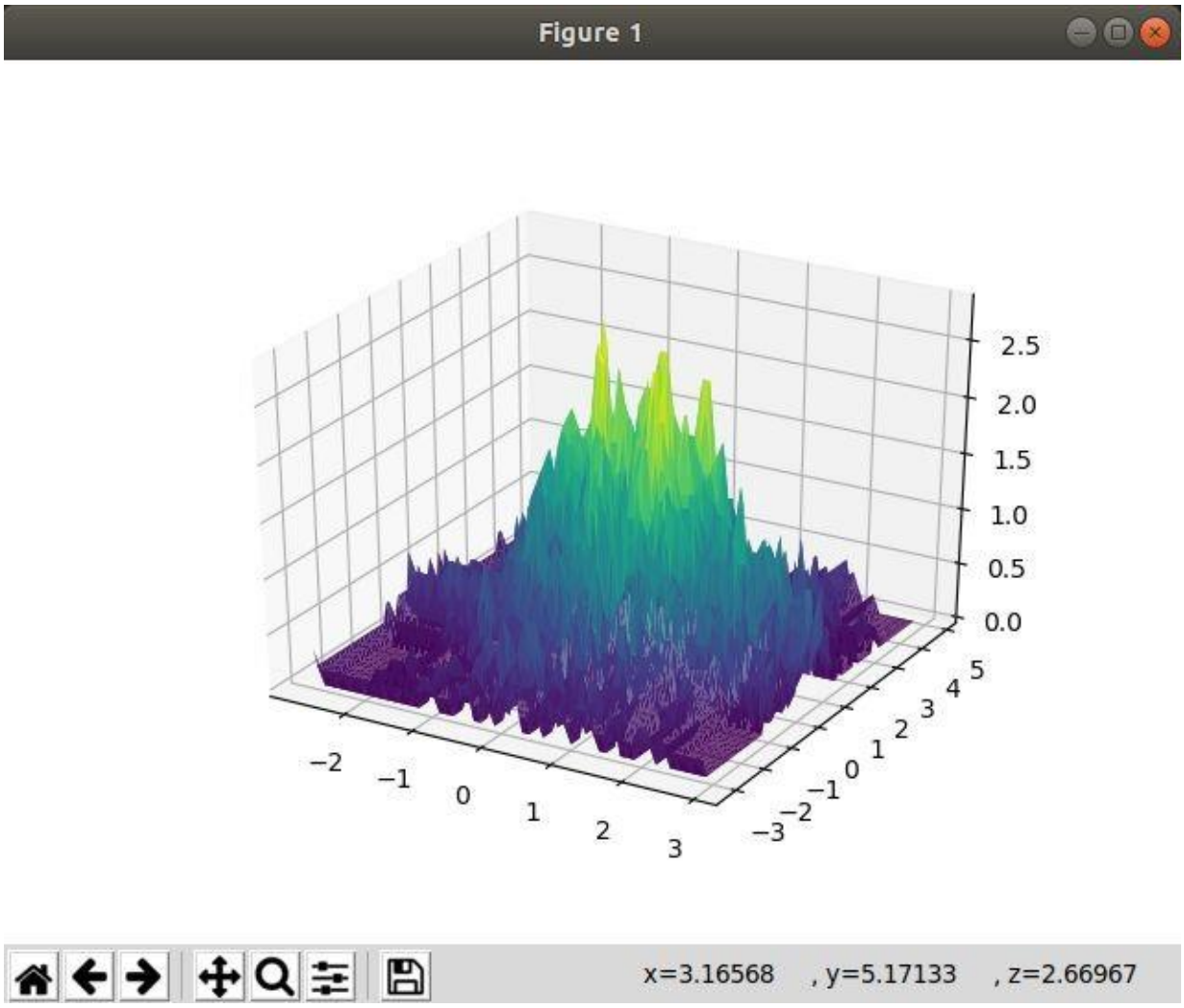
This plot is generated by combining two data sets, where the mean1 = 5, sigma1 = 1 and mean1 = 0, sigma1 = 2 and h =5



This plot is generated by combining two data sets, where the mean1 = 5, sigma1 = 1 and mean1 = 0, sigma1 = 2 and h =10

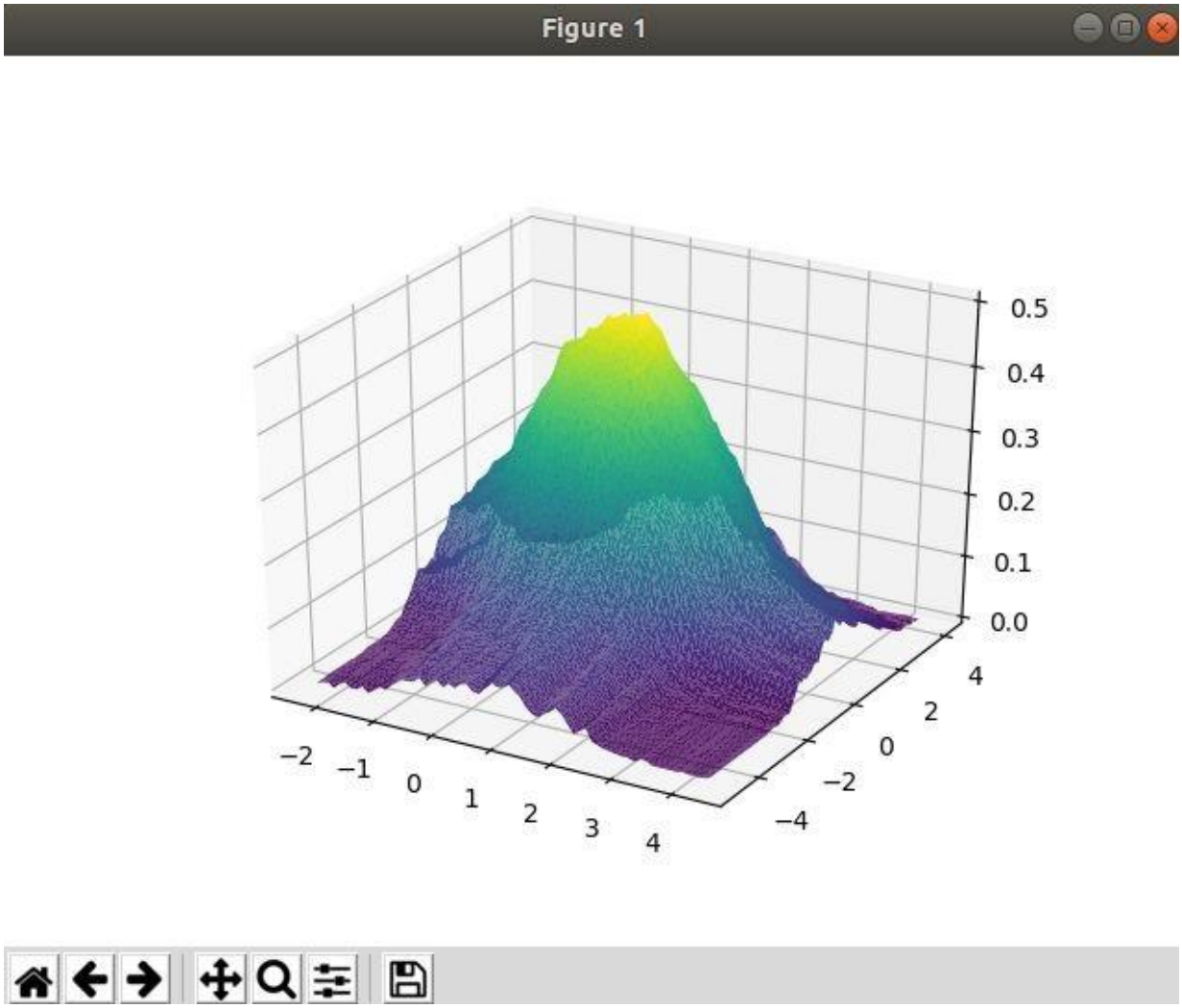
7. Execute filename problem 1.2.py

8.



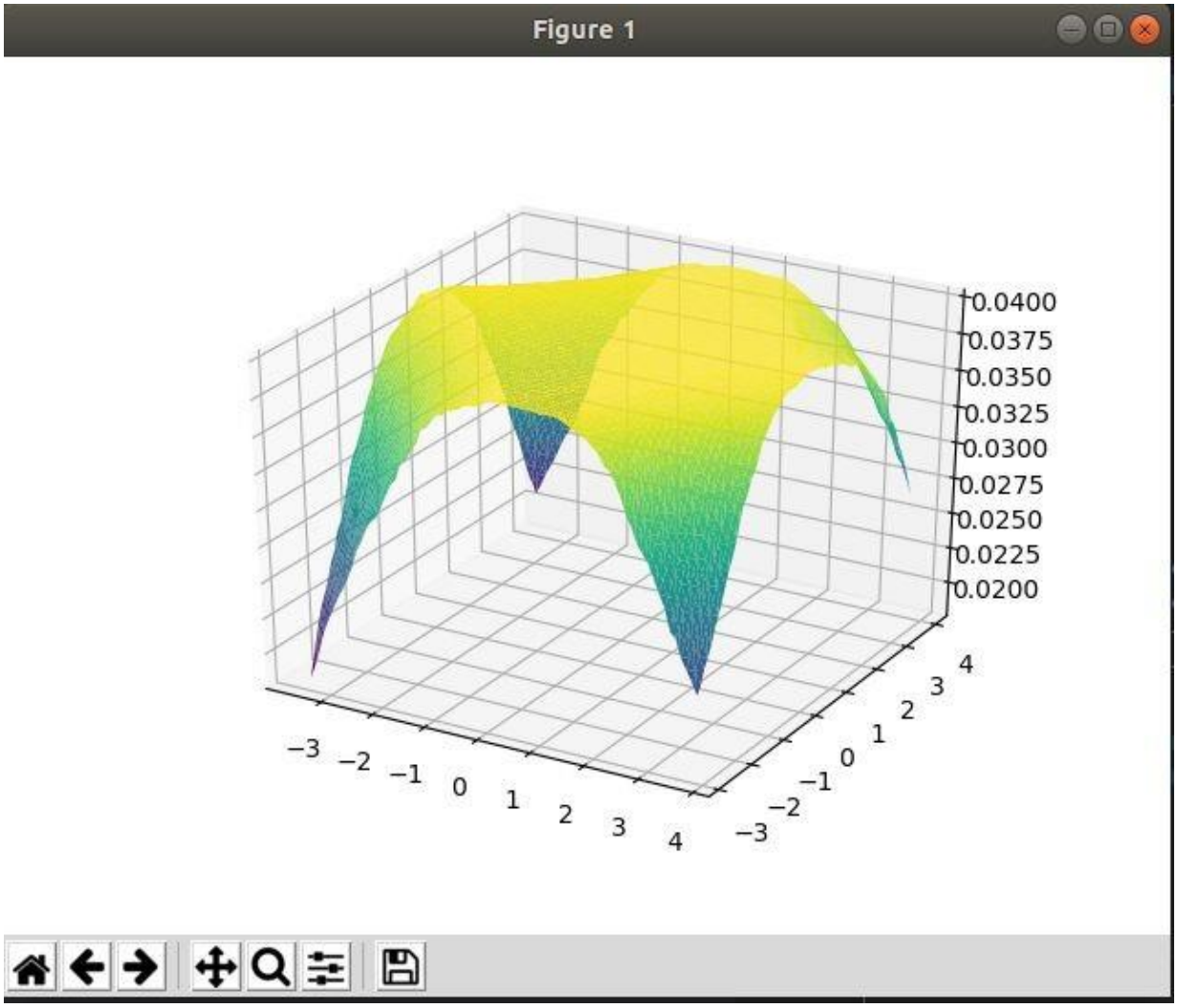
This plot is generated by combining two data sets, where the μ 1 = [1, 0], μ 2 = [0, 1.5],

Σ 1 =[0.9, 0.4; 0.4,0.9], Σ 2 =[0.9, 0.4; 0.4,0.9] and h = 0.1



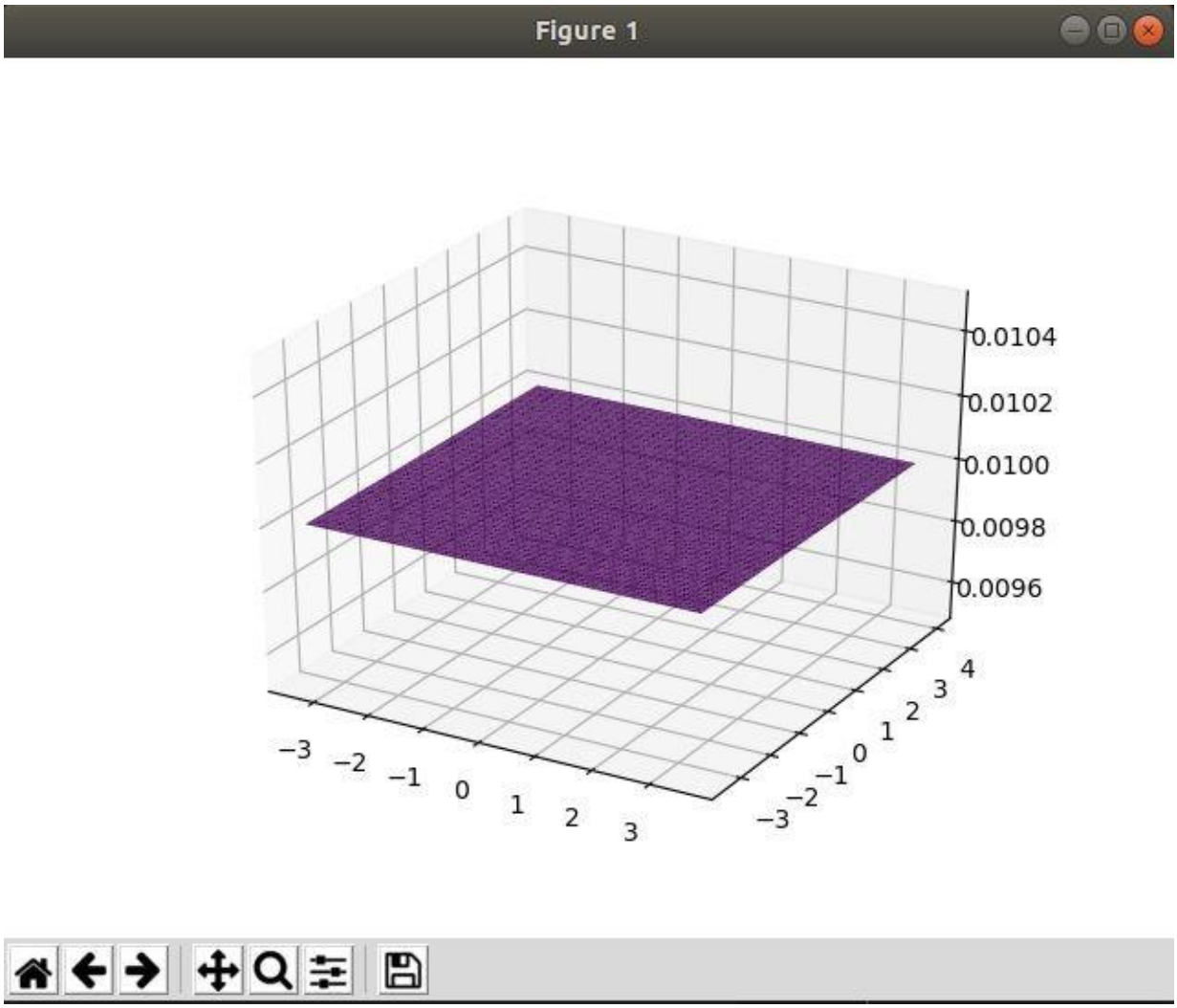
This plot is generated by combining two data sets, where the μ 1 = [1, 0], μ 2 = [0, 1.5],

Σ 1 =[0.9, 0.4; 0.4,0.9], Σ 2 =[0.9, 0.4; 0.4,0.9] and h = 1



This plot is generated by combining two data sets, where the μ 1 = [1, 0], μ 2 = [0, 1.5],

Σ 1 =[0.9, 0.4; 0.4,0.9], Σ 2 =[0.9, 0.4; 0.4,0.9] and h =5



This plot is generated by combining two data sets, where the μ 1 = [1, 0], μ 2 = [0, 1.5],

Σ 1 =[0.9, 0.4; 0.4,0.9], Σ 2 =[0.9, 0.4; 0.4,0.9] and h =10

Problem 2

Run python file named problem2.py

We are assuming that the labels remain to 0 and 1, and the training and testing datasets are created in the file and not provided by the user.

Functions created:

naïve_bayes(trainingset0_size,trainingset1_size)

trainingset0_size,trainingset1_size are the sizes of the training set of label 0 and label 1 respectively. This function is called in the main function and the sizes of the training sets can be adjusted.

find_roc(tpr,fpr)

Tpr and fpr are columns of a pandas data frame that are passed to the function to calculate and draw the roc curve. The function call is commented by default under the naïve_ bayes function and **needs to be uncommented** to get roc graphs.

Area_curve(maxx, maxxy)

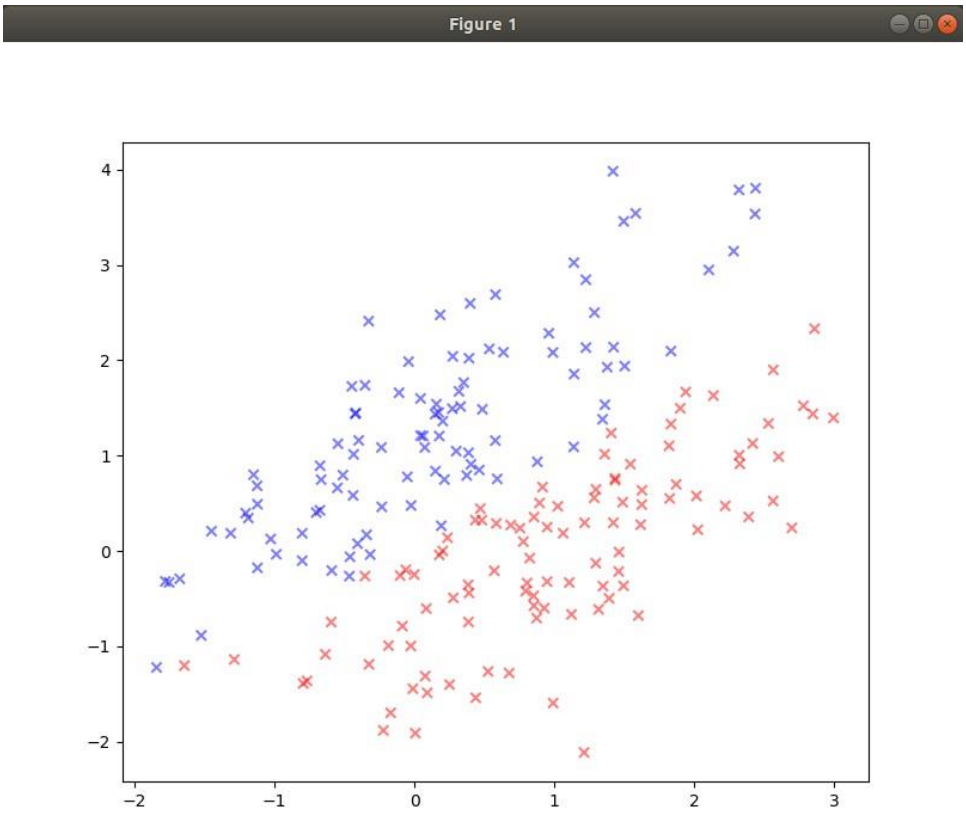
Maxx and maxy are the max value of tpr and fpr respectively. They are required to calculate the area under the curve. We are approximating the area by taking the area of a trapezoid. This function call is commented by default under find_roc and **needs to be uncommented** to get the AUC

Results

The following testing dataframe is generated(but not printed) upon runnng of the program.

	x	y	actual	post0	post1	pred	predicted	color	...	false_positive	false_negative	tp_cum	tn_cum	fn_cum	fp_cum	tpr	fpr
0	1.203588	0.503616	0	0.062167	0.036644	0.062167	0.0	r	...	0.0	0.0	0.0	1.0	0.0	0.0	NaN	0.000000
1	0.393617	-0.270154	0	0.059328	0.031020	0.059328	0.0	r	...	0.0	0.0	0.0	2.0	0.0	0.0	NaN	0.000000
2	-0.595779	-1.182829	0	0.012666	0.005086	0.012666	0.0	r	...	0.0	0.0	0.0	3.0	0.0	0.0	NaN	0.000000
3	0.517485	-0.083609	0	0.064750	0.037350	0.064750	0.0	r	...	0.0	0.0	0.0	4.0	0.0	0.0	NaN	0.000000
4	-0.125787	-1.392933	0	0.017424	0.003818	0.017424	0.0	r	...	0.0	0.0	0.0	5.0	0.0	0.0	NaN	0.000000
5	-0.464367	-2.420860	0	0.002069	0.000161	0.002069	0.0	r	...	0.0	0.0	0.0	6.0	0.0	0.0	NaN	0.000000
6	1.996205	0.748534	0	0.034549	0.012666	0.034549	0.0	r	...	0.0	0.0	0.0	7.0	0.0	0.0	NaN	0.000000
7	-1.753235	-2.456138	0	0.000173	0.000032	0.000173	0.0	r	...	0.0	0.0	0.0	8.0	0.0	0.0	NaN	0.000000
8	0.099109	0.113013	0	0.050520	0.050167	0.050520	0.0	r	...	0.0	0.0	0.0	9.0	0.0	0.0	NaN	0.000000
9	0.474059	0.832699	0	0.047217	0.071096	0.071096	1.0	b	...	0.0	1.0	0.0	9.0	1.0	0.0	0.000000	0.000000
10	2.416477	0.977122	0	0.018163	0.005562	0.018163	0.0	r	...	0.0	0.0	0.0	10.0	1.0	0.0	0.000000	0.000000
11	-0.378568	-0.572056	0	0.027191	0.018699	0.027191	0.0	r	...	0.0	0.0	0.0	11.0	1.0	0.0	0.000000	0.000000
12	-0.726987	-2.478082	0	0.001263	0.000110	0.001263	0.0	r	...	0.0	0.0	0.0	12.0	1.0	0.0	0.000000	0.000000
13	1.133890	0.598004	0	0.060173	0.041373	0.060173	0.0	r	...	0.0	0.0	0.0	13.0	1.0	0.0	0.000000	0.000000
14	0.352569	0.072128	0	0.059975	0.046606	0.059975	0.0	r	...	0.0	0.0	0.0	14.0	1.0	0.0	0.000000	0.000000
15	0.372434	-0.111109	0	0.060347	0.038059	0.060347	0.0	r	...	0.0	0.0	0.0	15.0	1.0	0.0	0.000000	0.000000
16	1.191838	0.523678	0	0.061763	0.037521	0.061763	0.0	r	...	0.0	0.0	0.0	16.0	1.0	0.0	0.000000	0.000000
17	0.837792	0.063377	0	0.070504	0.036343	0.070504	0.0	r	...	0.0	0.0	0.0	17.0	1.0	0.0	0.000000	0.000000
73	0.621767	2.118210	1	0.009367	0.039053	0.039053	1.0	b	...	0.0	0.0	68.0	96.0	4.0	6.0	0.944444	0.058824
74	-0.052468	2.365392	1	0.003800	0.033217	0.033217	1.0	b	...	0.0	0.0	69.0	96.0	4.0	6.0	0.945205	0.058824
75	-0.309053	1.171600	1	0.018809	0.072222	0.072222	1.0	b	...	0.0	0.0	70.0	96.0	4.0	6.0	0.945946	0.058824
76	0.934517	1.508826	1	0.026193	0.050023	0.050023	1.0	b	...	0.0	0.0	71.0	96.0	4.0	6.0	0.946667	0.058824
77	1.114423	2.583550	1	0.003717	0.014745	0.014745	1.0	b	...	0.0	0.0	72.0	96.0	4.0	6.0	0.947368	0.058824
78	0.496190	2.220862	1	0.007349	0.036929	0.036929	1.0	b	...	0.0	0.0	73.0	96.0	4.0	6.0	0.948052	0.058824
79	-0.247207	1.118238	1	0.021271	0.074166	0.074166	1.0	b	...	0.0	0.0	74.0	96.0	4.0	6.0	0.948718	0.058824
80	0.738282	2.459554	1	0.004838	0.024028	0.024028	1.0	b	...	0.0	0.0	75.0	96.0	4.0	6.0	0.949367	0.058824
81	-0.215324	1.253768	1	0.019107	0.073638	0.073638	1.0	b	...	0.0	0.0	76.0	96.0	4.0	6.0	0.950000	0.058824
82	-1.453126	1.037128	1	0.003205	0.024673	0.024673	1.0	b	...	0.0	0.0	77.0	96.0	4.0	6.0	0.950617	0.058824
83	0.226030	1.176947	1	0.030366	0.077201	0.077201	1.0	b	...	0.0	0.0	78.0	96.0	4.0	6.0	0.951220	0.058824
84	1.173534	1.019352	1	0.044319	0.044125	0.044319	0.0	r	...	1.0	0.0	78.0	96.0	4.0	7.0	0.951220	0.067961
85	0.761616	1.220729	1	0.036379	0.062069	0.062069	1.0	b	...	0.0	0.0	79.0	96.0	4.0	7.0	0.951807	0.067961
86	1.233298	1.933646	1	0.013302	0.028279	0.028279	1.0	b	...	0.0	0.0	80.0	96.0	4.0	7.0	0.952381	0.067961
87	-0.625514	0.382869	1	0.021422	0.048725	0.048725	1.0	b	...	0.0	0.0	81.0	96.0	4.0	7.0	0.952941	0.067961
88	-0.611393	1.599099	1	0.007572	0.053494	0.053494	1.0	b	...	0.0	0.0	82.0	96.0	4.0	7.0	0.953488	0.067961
89	0.105160	0.861676	1	0.036933	0.076926	0.076926	1.0	b	...	0.0	0.0	83.0	96.0	4.0	7.0	0.954023	0.067961
90	1.727749	2.355454	1	0.004754	0.009157	0.009157	1.0	b	...	0.0	0.0	84.0	96.0	4.0	7.0	0.954545	0.067961
91	-2.322193	-1.562704	1	0.000184	0.000150	0.000184	0.0	r	...	1.0	0.0	84.0	96.0	4.0	8.0	0.954545	0.076923

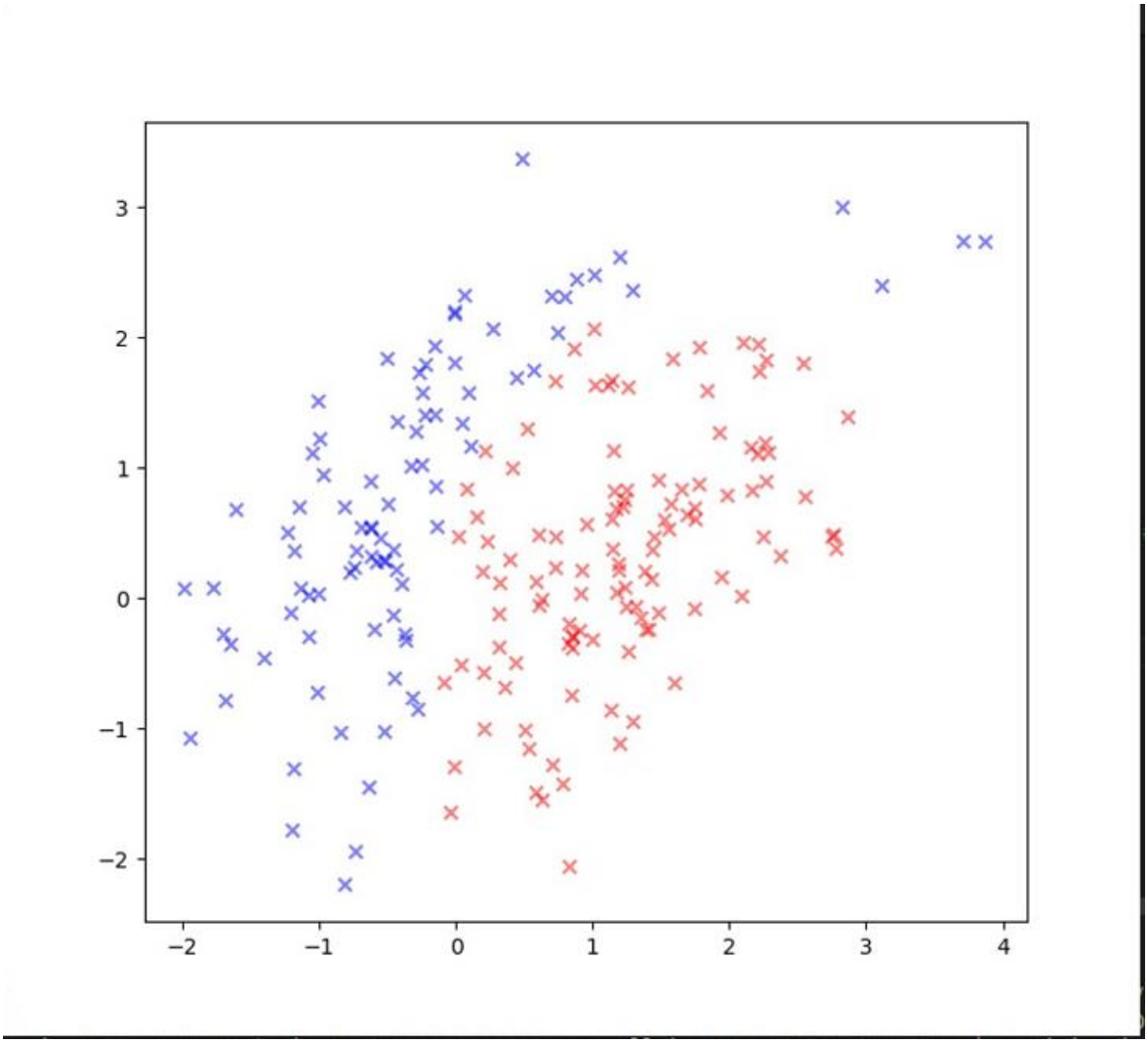
Posterior probability and pred are generated in the dataframe. The error is calculated after the generation of the dataframe.




```
accuracy is 0.93,error rate is 0.06999999999999995,recall is 0.9387755102040817 and precision is 0.92 True positive is 92.0 true negative is 94.0 False positive is 8.0 Fa
lse negative is 6.0
```

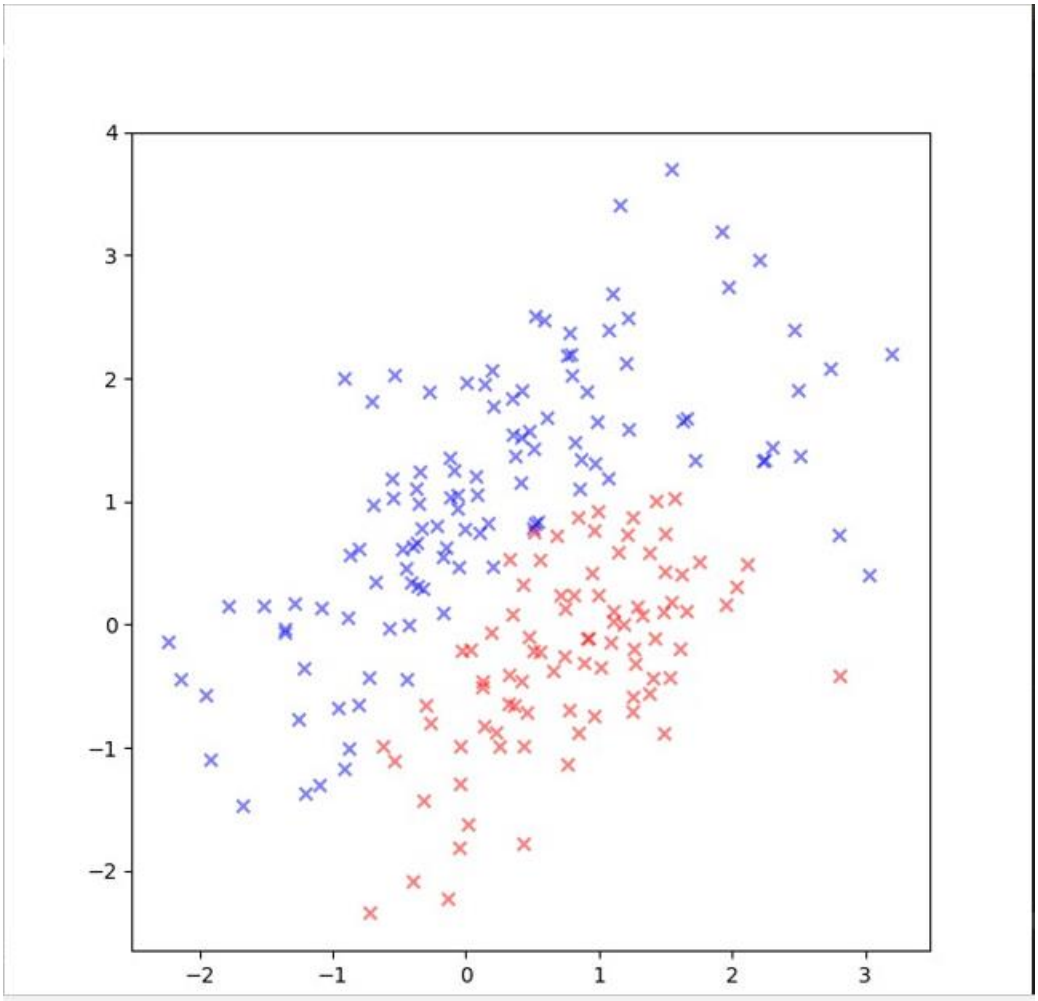
Scatter plot, accuracy and confusion matrix when training set is 500,500.

Red points are 0, blue points are 1.



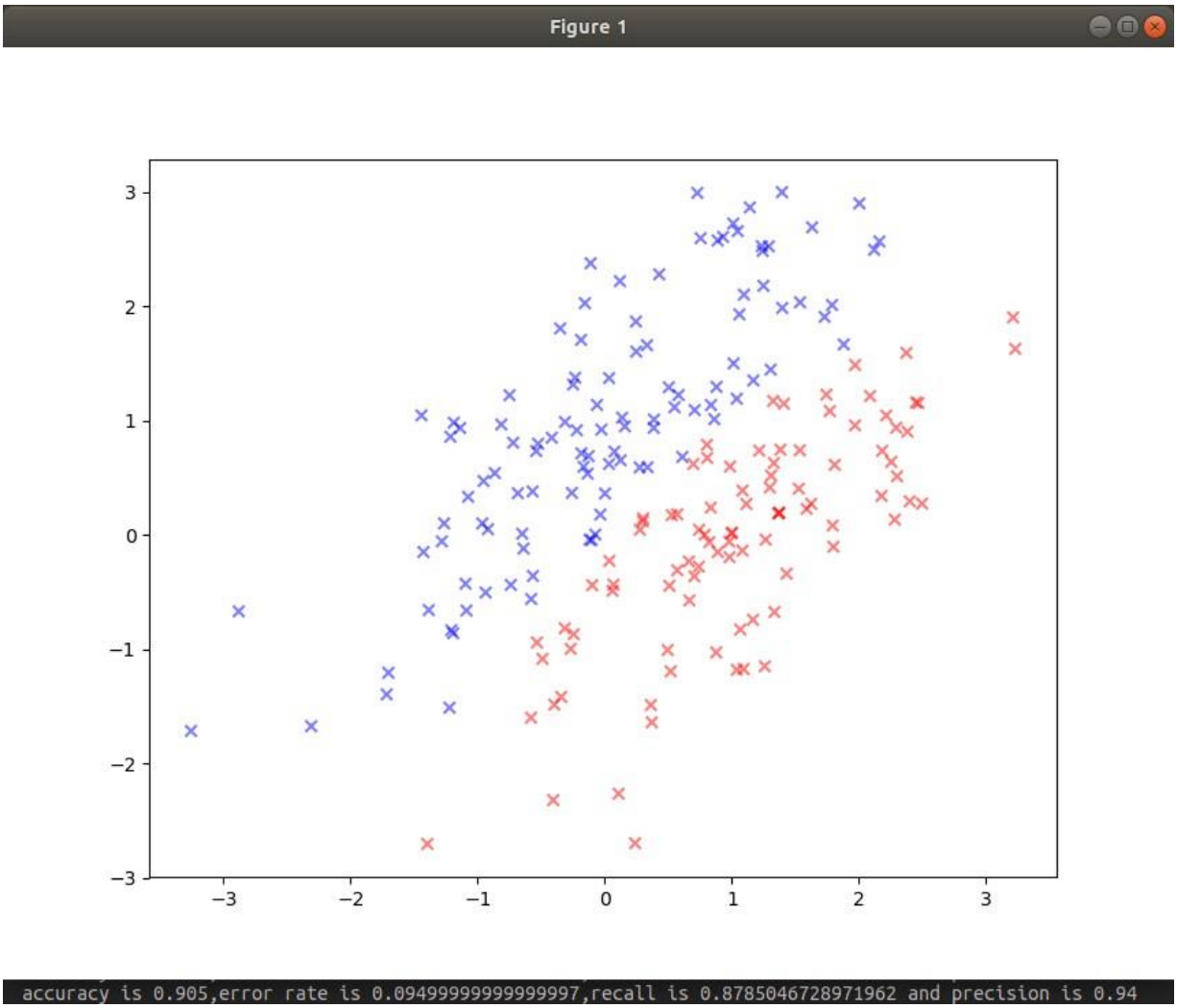
Scatter plot, accuracy and confusion matrix when training set is 10,10.

Red points are 0, blue points are 1.



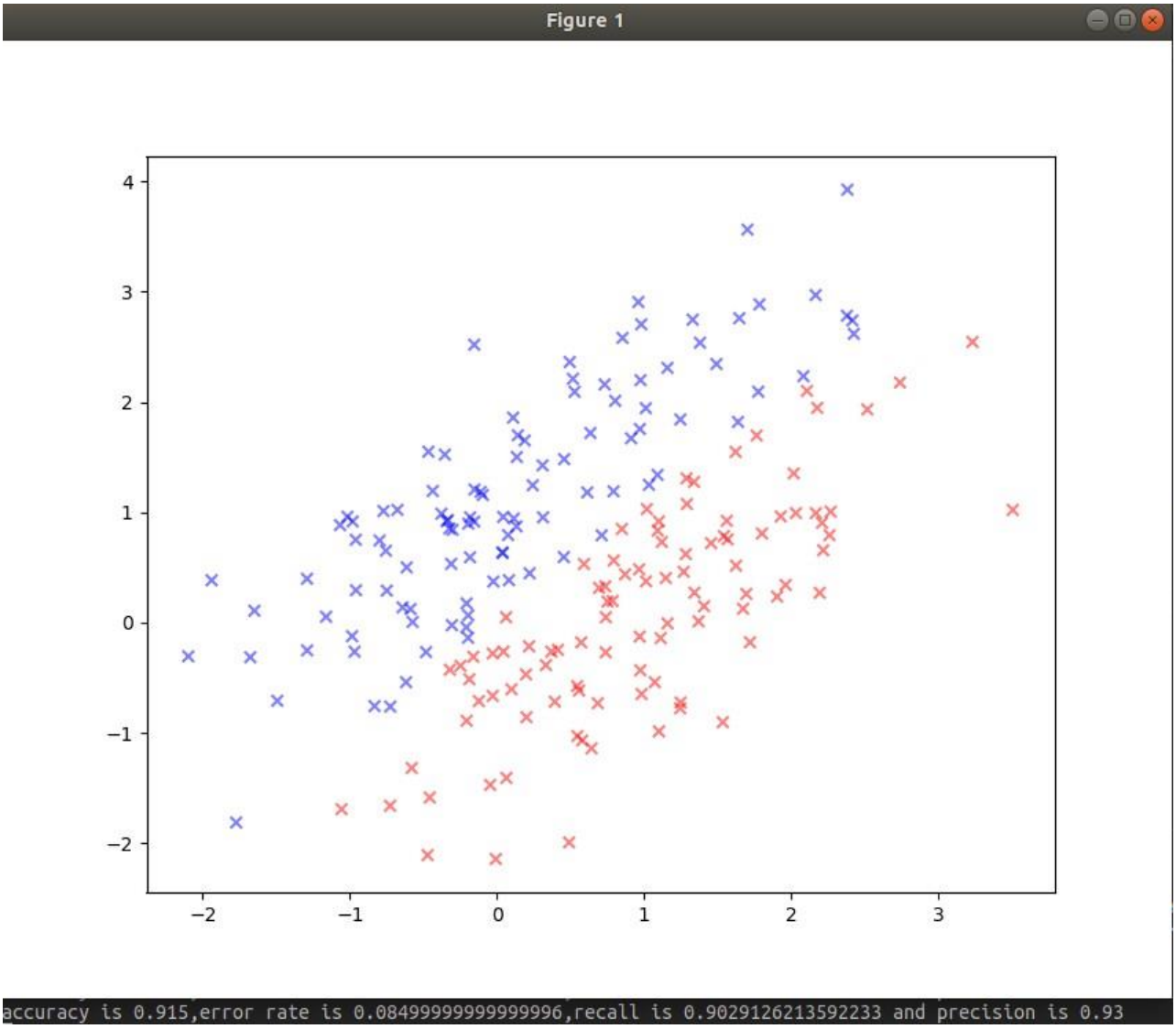
Scatter plot, accuracy and confusion matrix when training set is 20,20.

Red points are 0, blue points are 1.



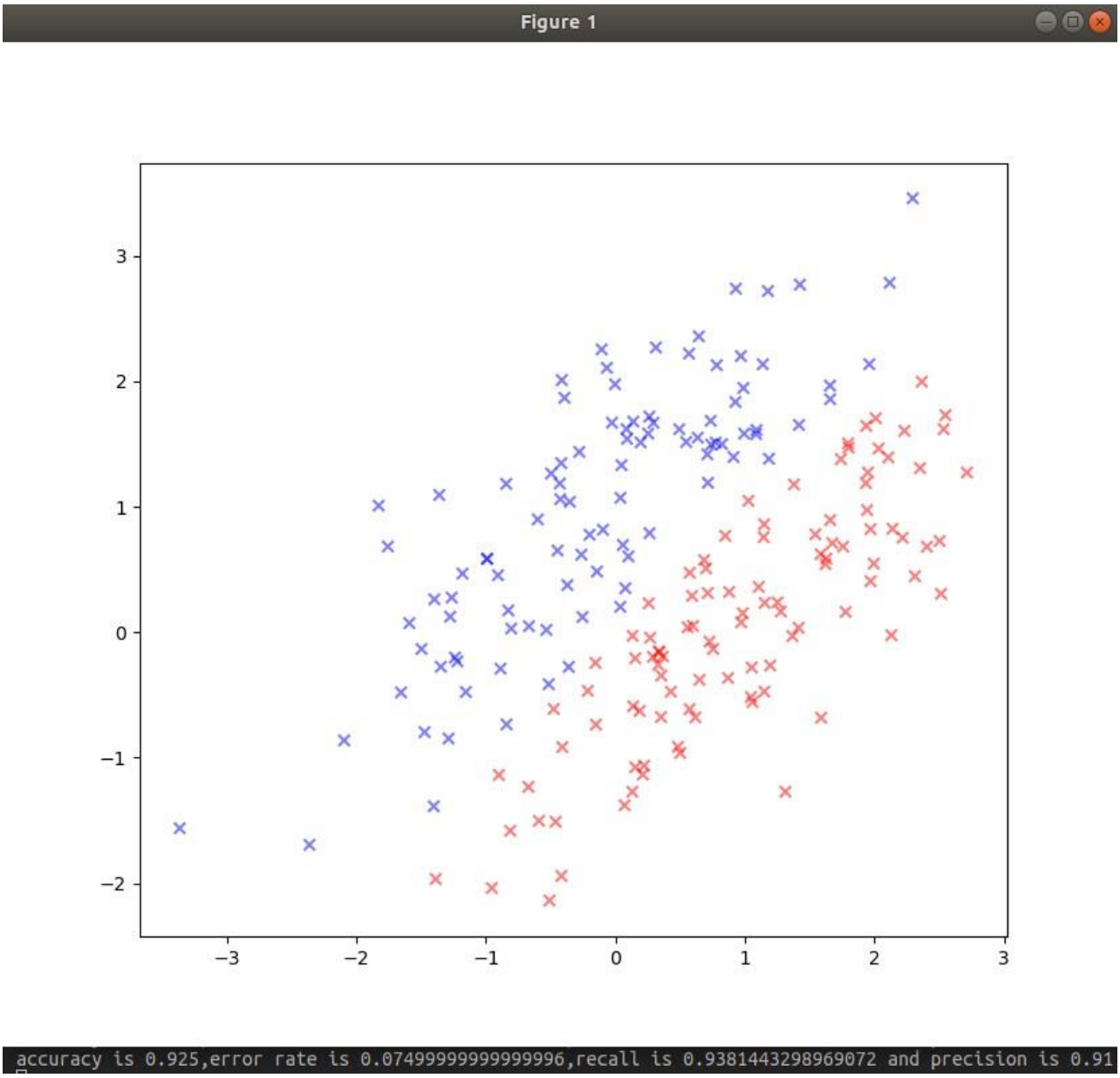
Scatter plot, accuracy and confusion matrix when training set is 50,50.

Red points are 0, blue points are 1.



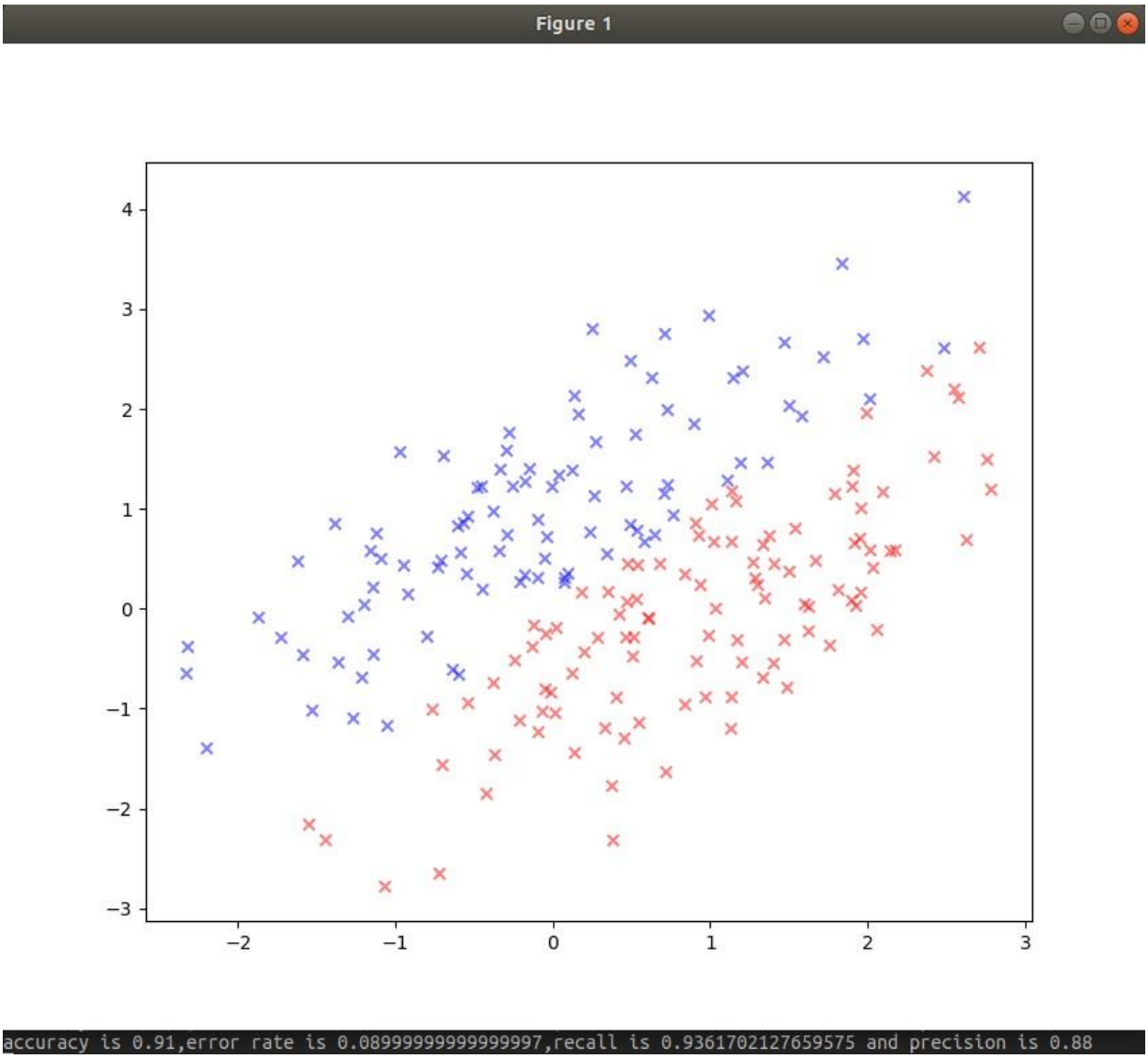
Scatter plot, accuracy and confusion matrix when training set is 100,100.

Red points are 0, blue points are 1.



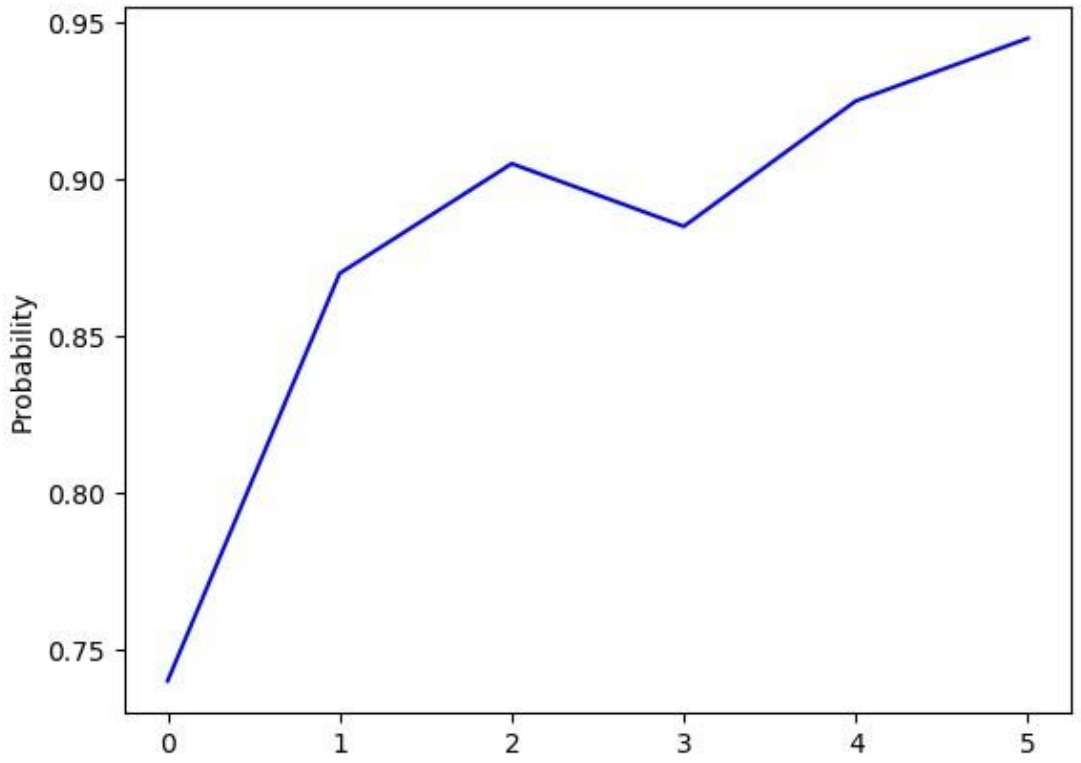
Scatter plot, accuracy and confusion matrix when training set is 300,300.

Red points are 0, blue points are 1.



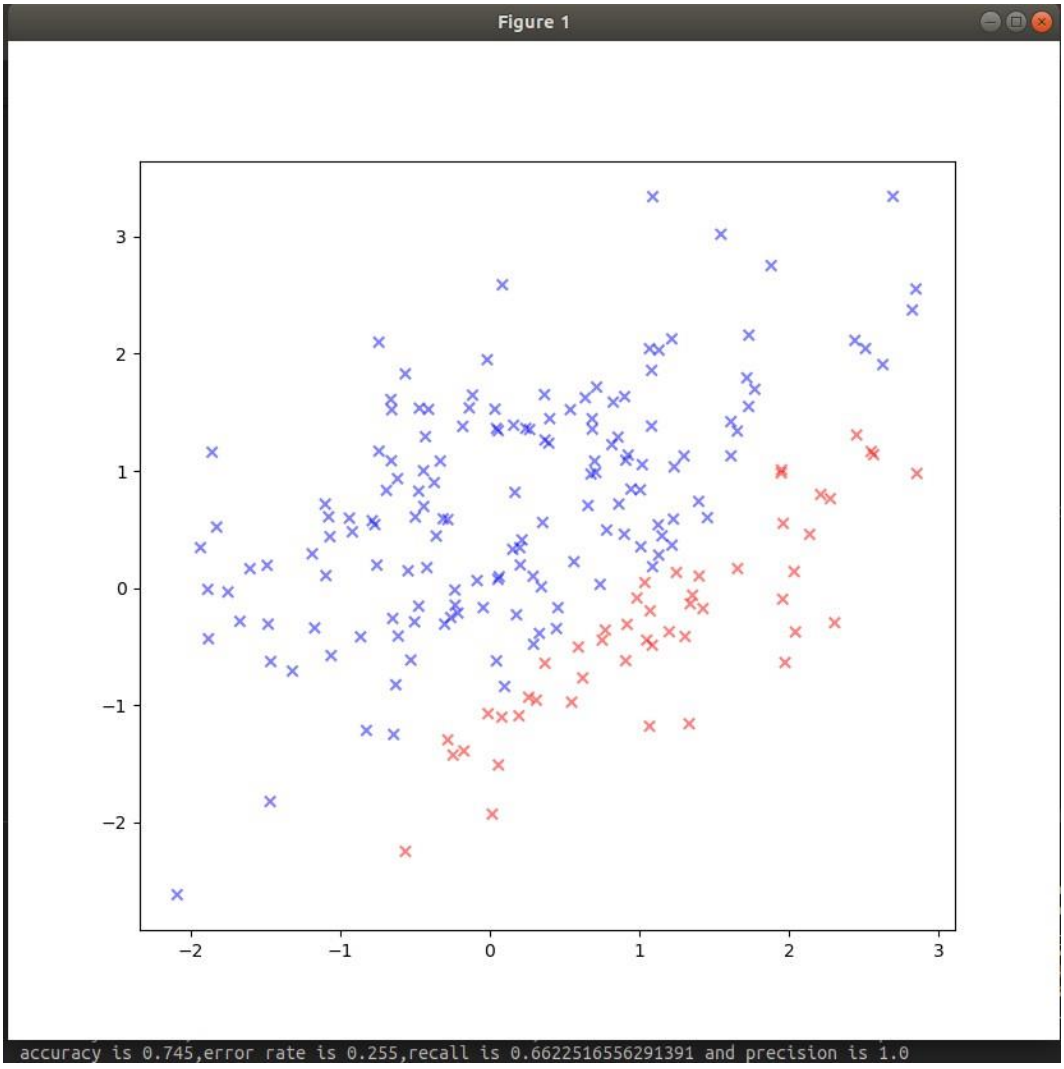
Scatter plot, accuracy and confusion matrix when training set is 500,500.

Red points are 0, blue points are 1.



Plot of the accuracies.

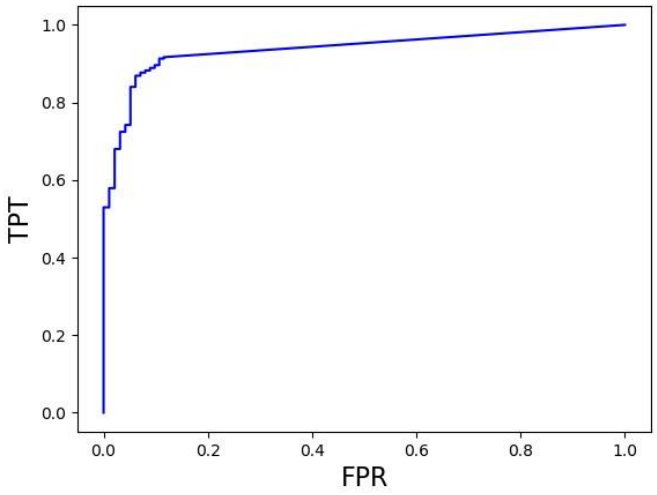
The accuracy for the most part tends to increase with the increasing size of the dataset. This can be contributed to the part that we would be getting gaussian curve and likelihood with a higher training dataset.



Scatter plot, accuracy and confusion matrix when training set is 300,700.

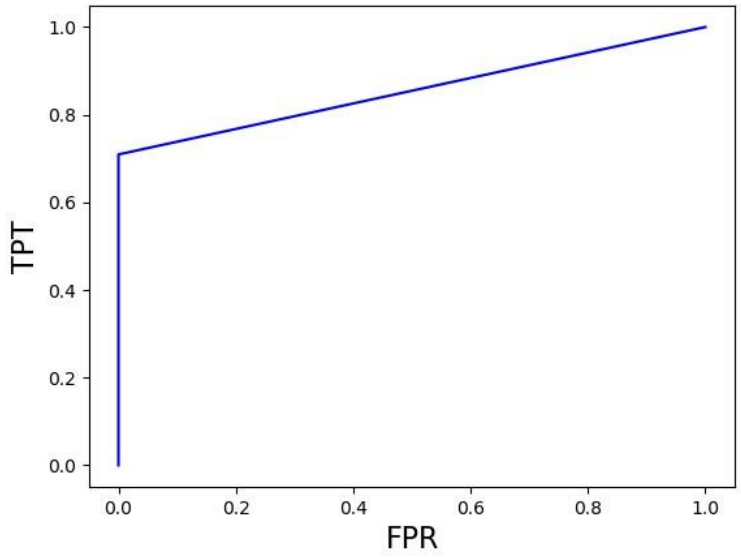
Red points are 0, blue points are 1.

The accuracy decreases, in comparison to the 500,500 dataset. This might be because, the training model was skewed towards the label 1. Hence, we end up with a lower accuracy. Most of the labels were erroneously labelled 1.



ROC for 500, 500 dataset.

```
accuracy is 0.9,error rate is 0.09999999999999998,recall is 0.9166666666666666 and precision is 0.88
Area under the curve is 0.8655770913857789
```



ROC for 300, 700 dataset.

```
accuracy is 0.795,error rate is 0.20499999999999996,recall is 0.7092198581560284 and precision is 1.0
Area under the curve is 0.7239073714424649
```