1.

a.

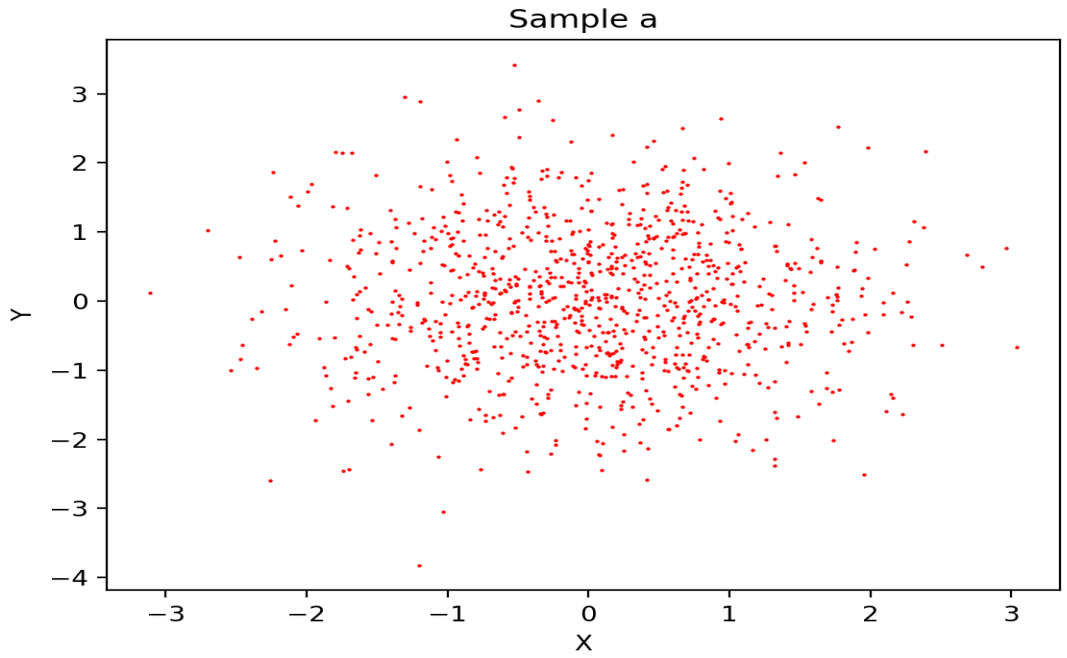


Figure 1: sample a

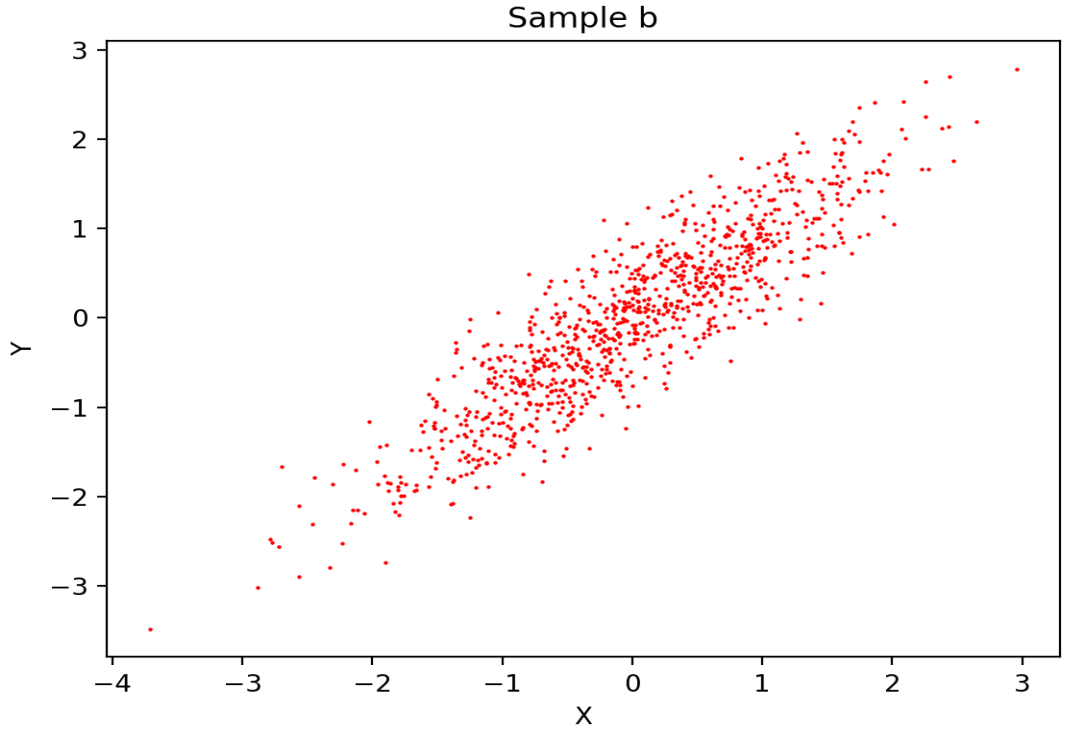


Figure 2: sample b

b.

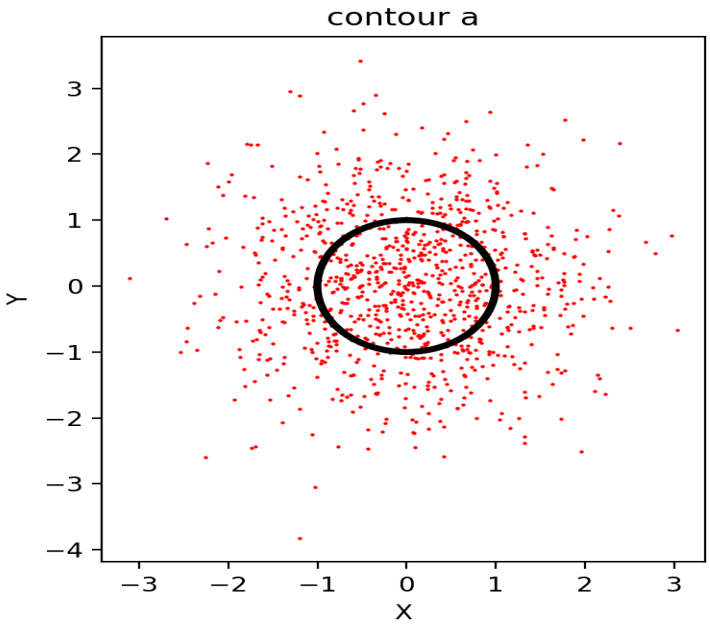


Figure 3 : contour sample a

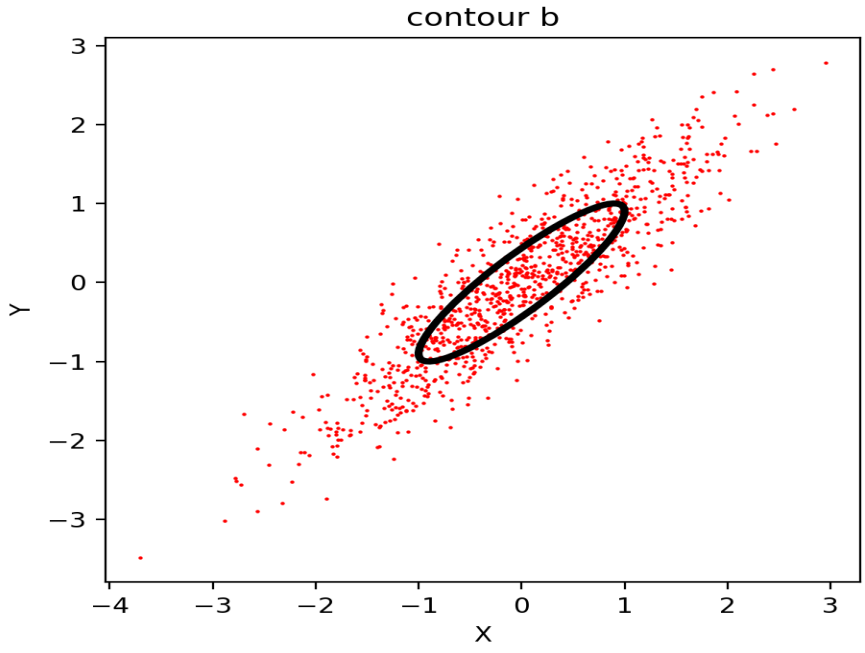


Figure 4 : contour sample b

c.
sample covariance matrices a
[[0.99185571 -0.00340674]
 [-0.00340674 1.06667082]]
sample covariance matrices b
[[0.95598692 0.87326169]
 [0.87326169 0.9890061]]
d.
The covariance matrix that we computed is a little bit different with given matrix. It is because we generate sample data with certain probability. However, we can not control what exactly it is. If we increase the amount of sample data, it will be more close to given matrix.

2.
a.

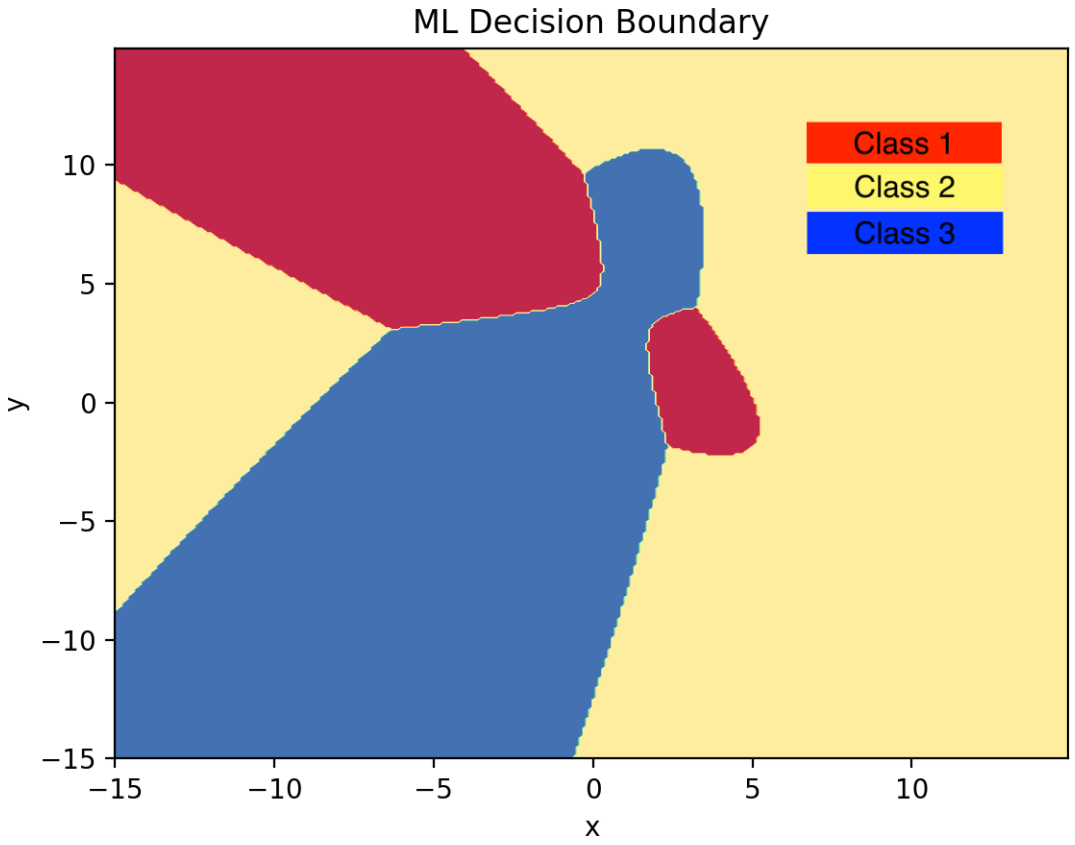


figure 5 : ML Decision Boundaries

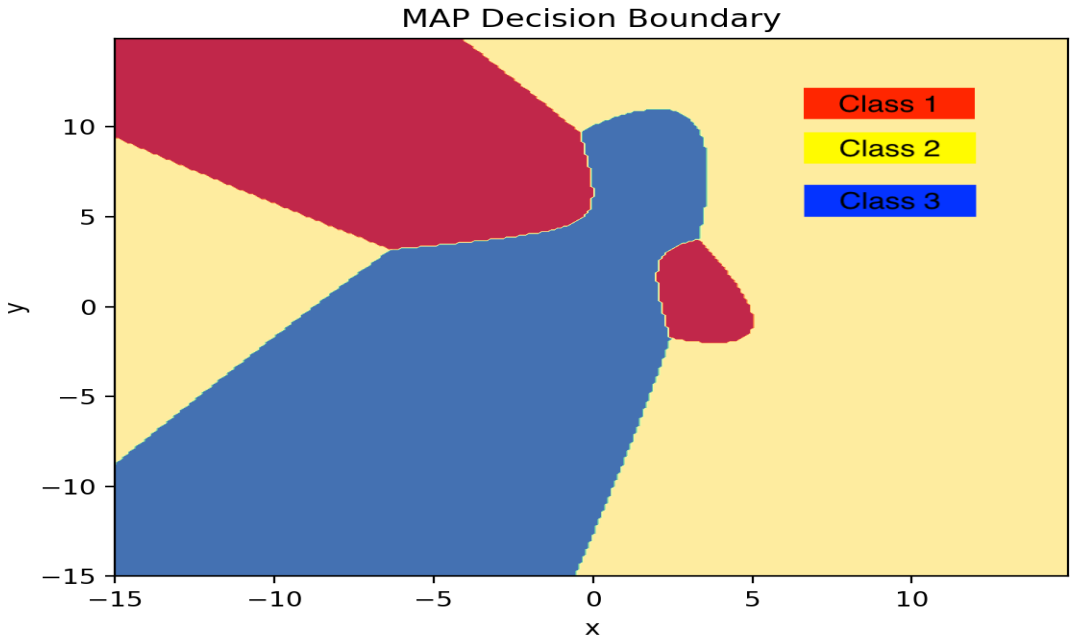
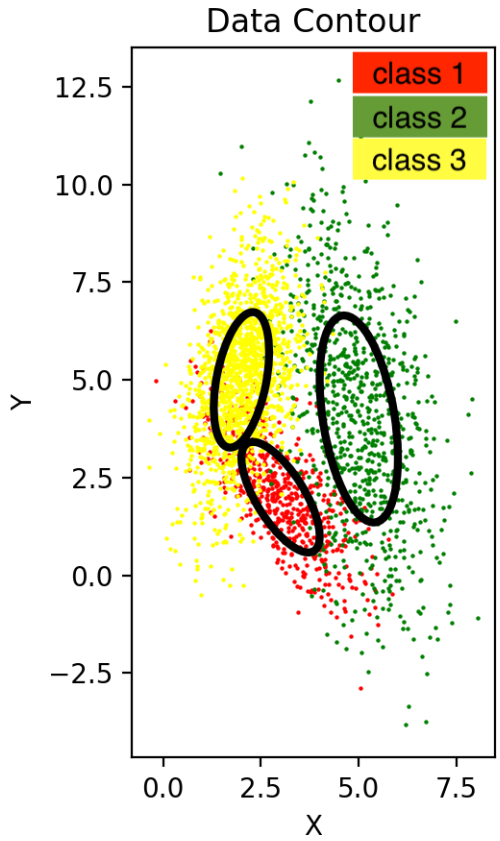


Figure 6 : MAP Decision Boundary



differences between the decision boundaries
The decision boundary has a little bit different between two methods. We can see from the calculation, the difference between them is prior probability. We can consider that, if the probability for each case is same, the ML decision boundary will be more like MAP decision boundary.

b.
ML confusion matrix
[[466 20 114]
 [48 811 41]
 [126 33 1341]]
P(error) = 0.12733333333333333

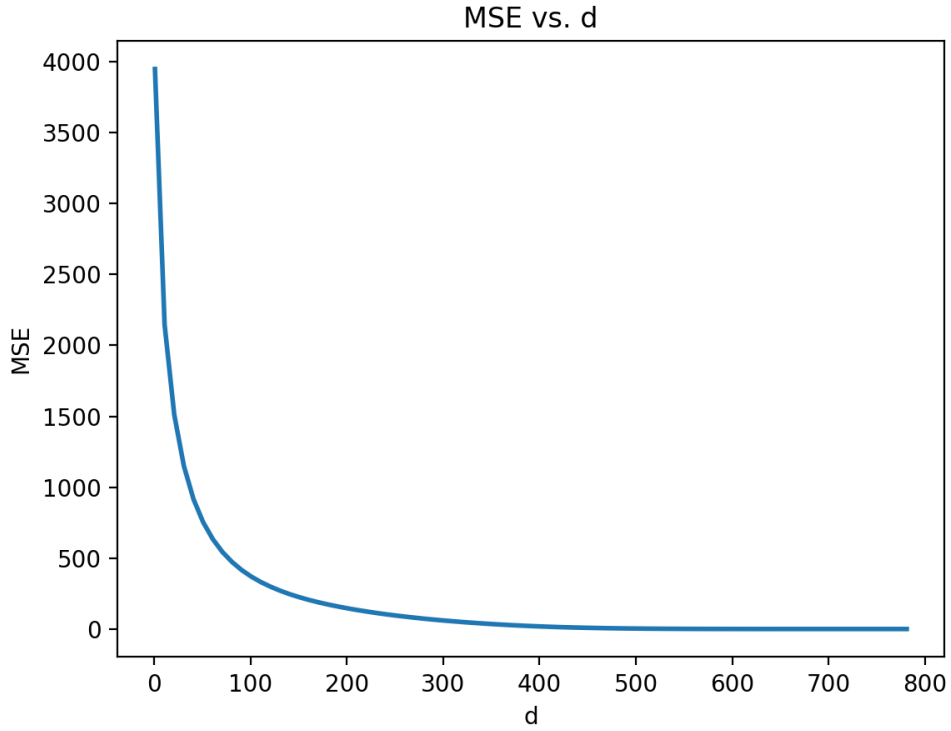
MAP confusion matrix
[[403 29 168]
 [42 804 54]
 [52 22 1426]]
P(error) = 0.12233333333333334

From the result, we can see that MAP has less error than ML because it consider the prior probability. We can conclude that MAP is more accurate than ML in most of cases.

3.
a. PCA method(see the code):
1. calculate of data average and use data minus mean as normal data
2. calculate the covariance
3. calculate the eigenvalue
4. rank the eigenvalue and selected first d features
5. final data = normal_data * selected_feature.T

b.
when d = 154, POV = 0.95

c.



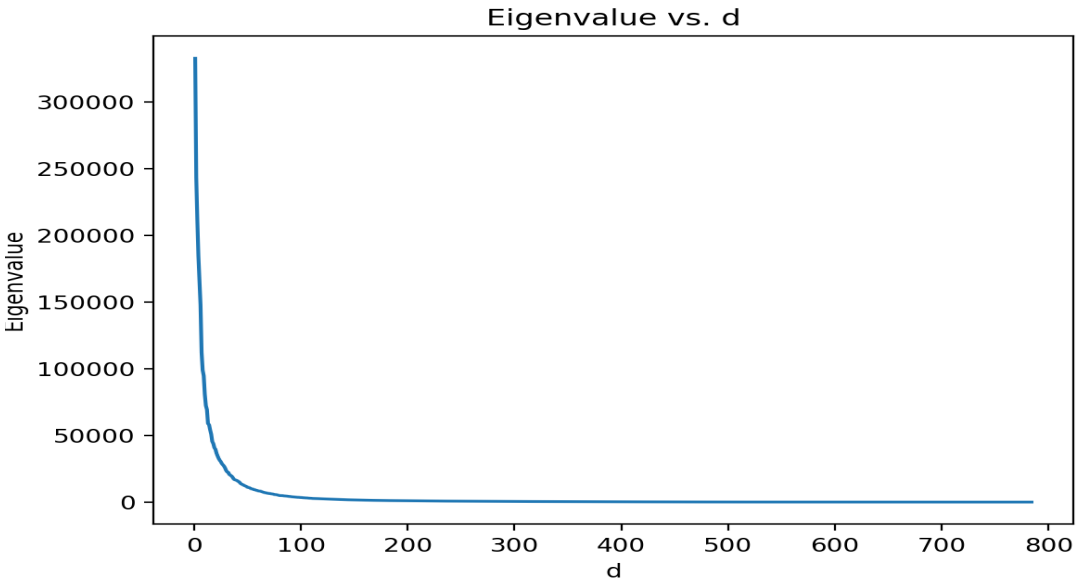
We can observe from plot that the MSE is large when d is less than 100. It decreases smoothly after 100. When d gets 400, the MSE is almost 0. As a result, we can conclude that if we want an extreme small MSE, d should be greater than 400.

d.

d = 1 d= 10 d=50 d= 250 d = 784

The first picture is not clear to classify what is the number. Too many information lost when d is 1 and 10. Starting from d equals 50, the picture is clear. With the d increasing, the reconstruct picture is more like the origin picture. When may conclude that d should be greater than 50 in order that it can keep enough information.

e.



We can see that eigenvalue is large when d is less than 100. It shows some significant features that can provide much information for classifier. After d = 100, the eigenvalue is almost zero. They may not that help for for the classification. The graph helps us to know we need at least 70 feature in order that we will not loss important feature.

4.

a.

k value	1	3	5	11
accuracy	0.9691	0.9705	0.9688	0.9668

b.

k/d	1	3	5	11
5	0.6985	0.7269	0.7459	0.7596
50	0.974	0.9749	0.975	0.9732
100	0.9714	0.9733	0.9723	0.9695
500	0.9691	0.9707	0.9687	0.9669

c.
From the result, we can see that the accuracy is low when d is 5. It is because POV is too low after PCA feature subtraction. Some important features may loss when the dimension is 5. In addition, we got a highest accuracy number when d is 5 and k is 50. With increasing d, accuracy decreased. It may because some useless feature makes a bad effect on classification.
As result, d =5 and k = 50 can provide best result from this data. Comparing with a, it can increase accuracy and reduce the execute time.