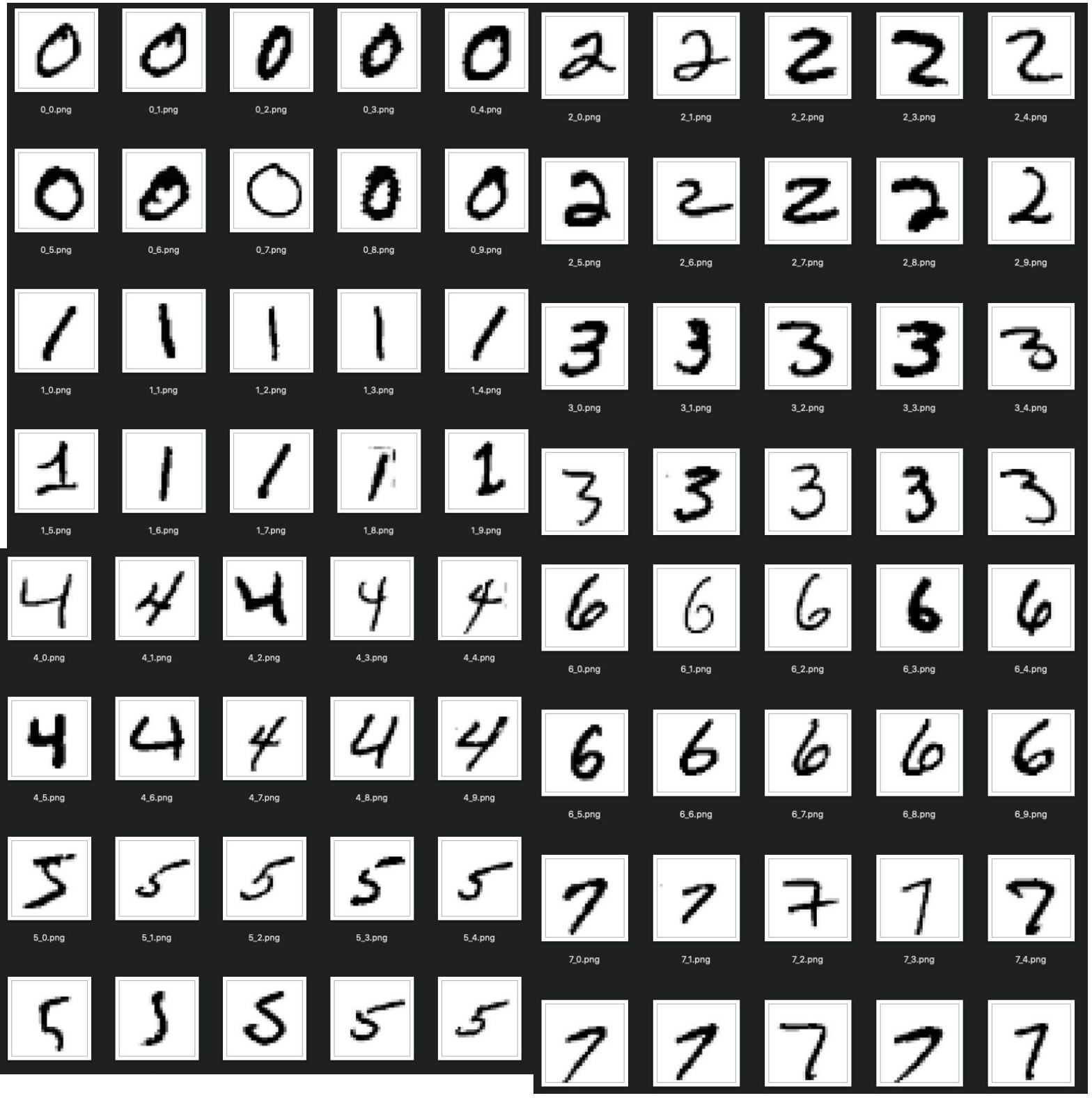
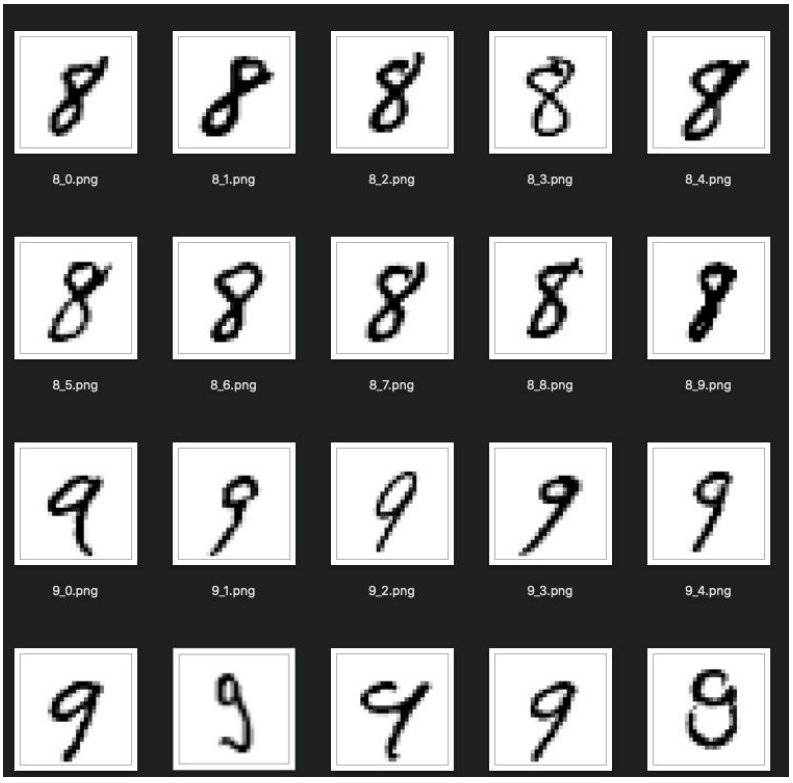
ML - Assignment 1

-Harkishan Singh (2017233)

Question 1 a):

In this question, we are supposed to visualise the data. The data is MNIST data of handwritten digits. It contains 50,000 images with ground truth labels.



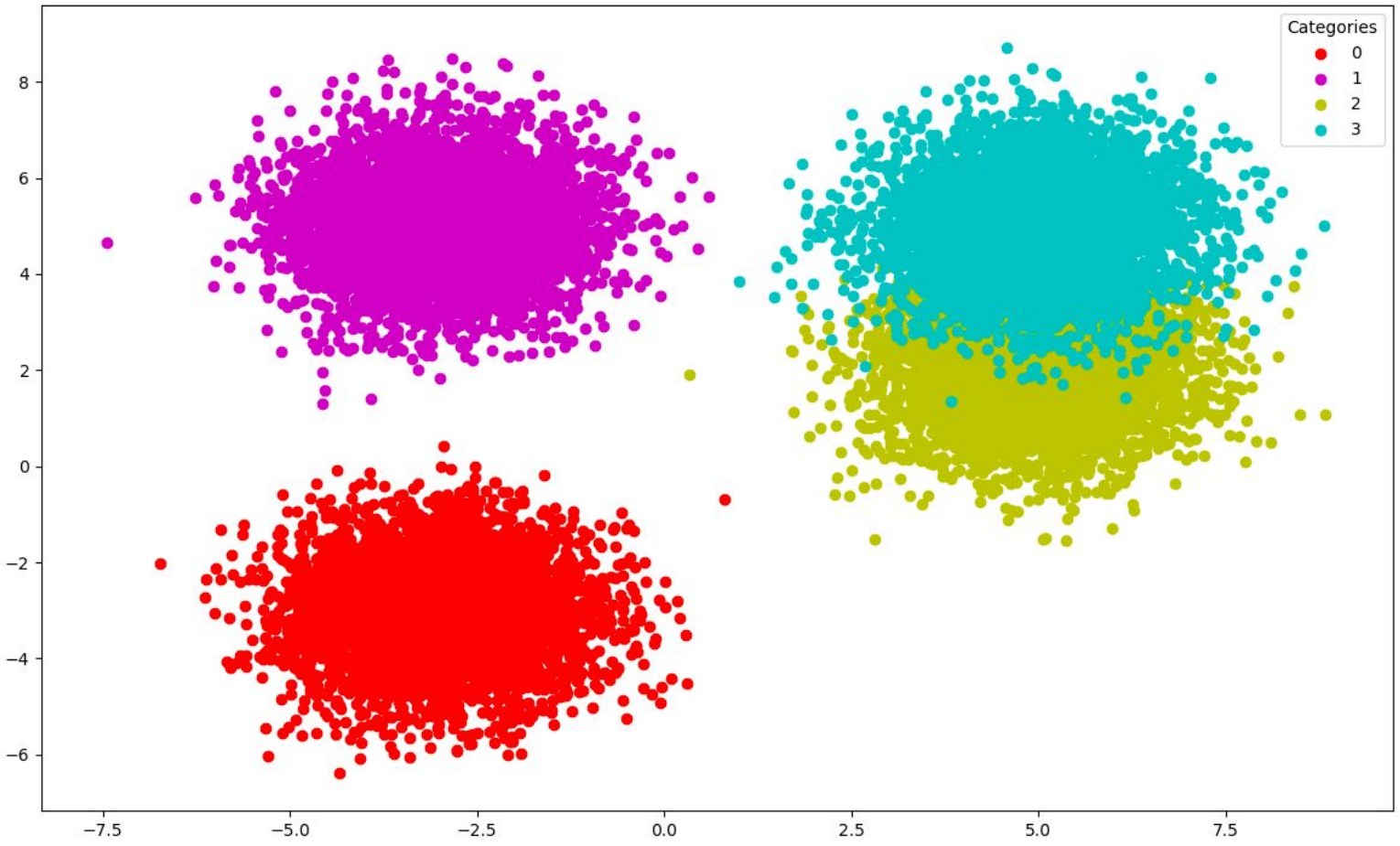


Question 1 b) on next page ⇒

(b):

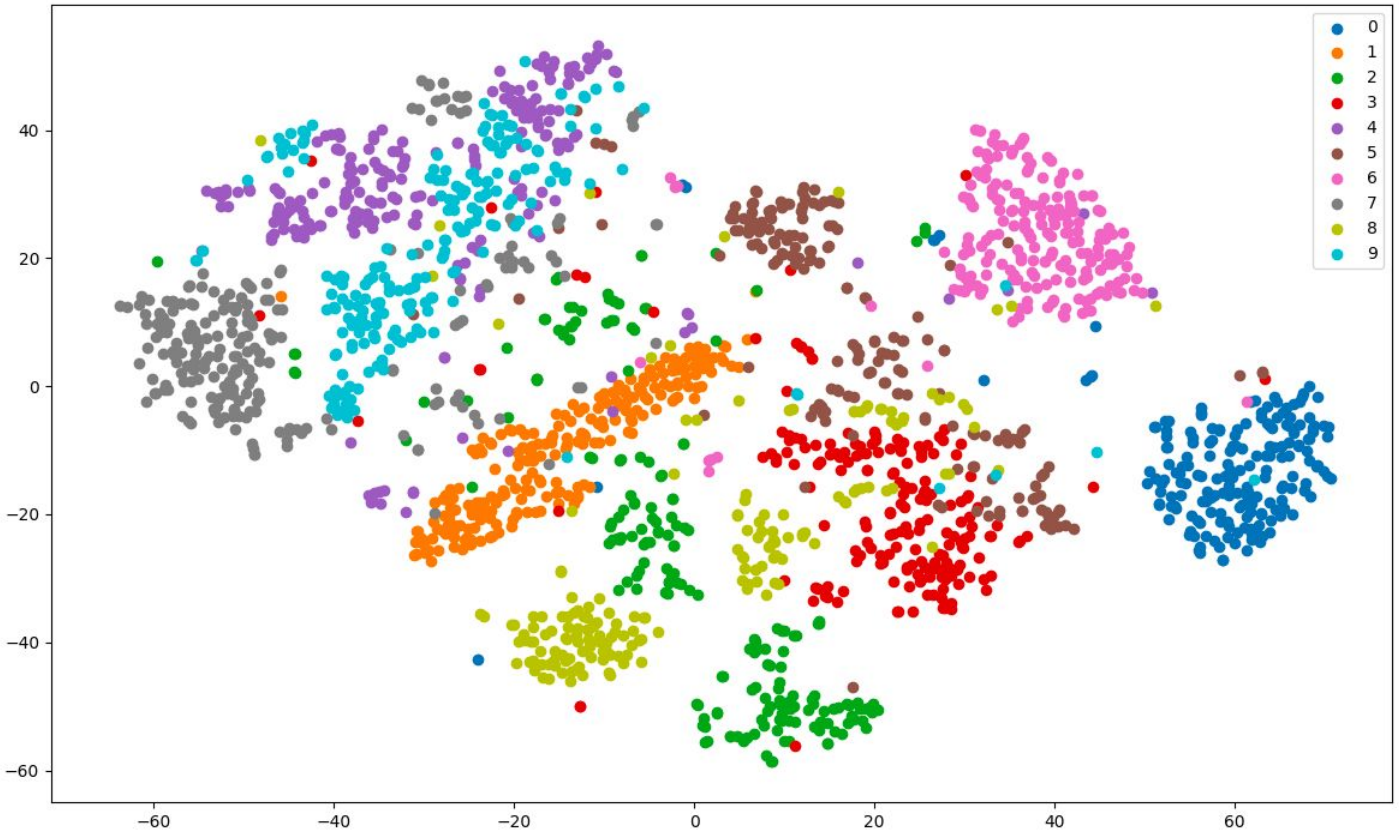
In this question, we are supposed to visualize dataset_2. I have plotted the dataset below.

Inferences : Category 0 and 1 are mutually exclusive and are easily separable whereas category 2 and 3 are overlapping which makes it difficult to separate.



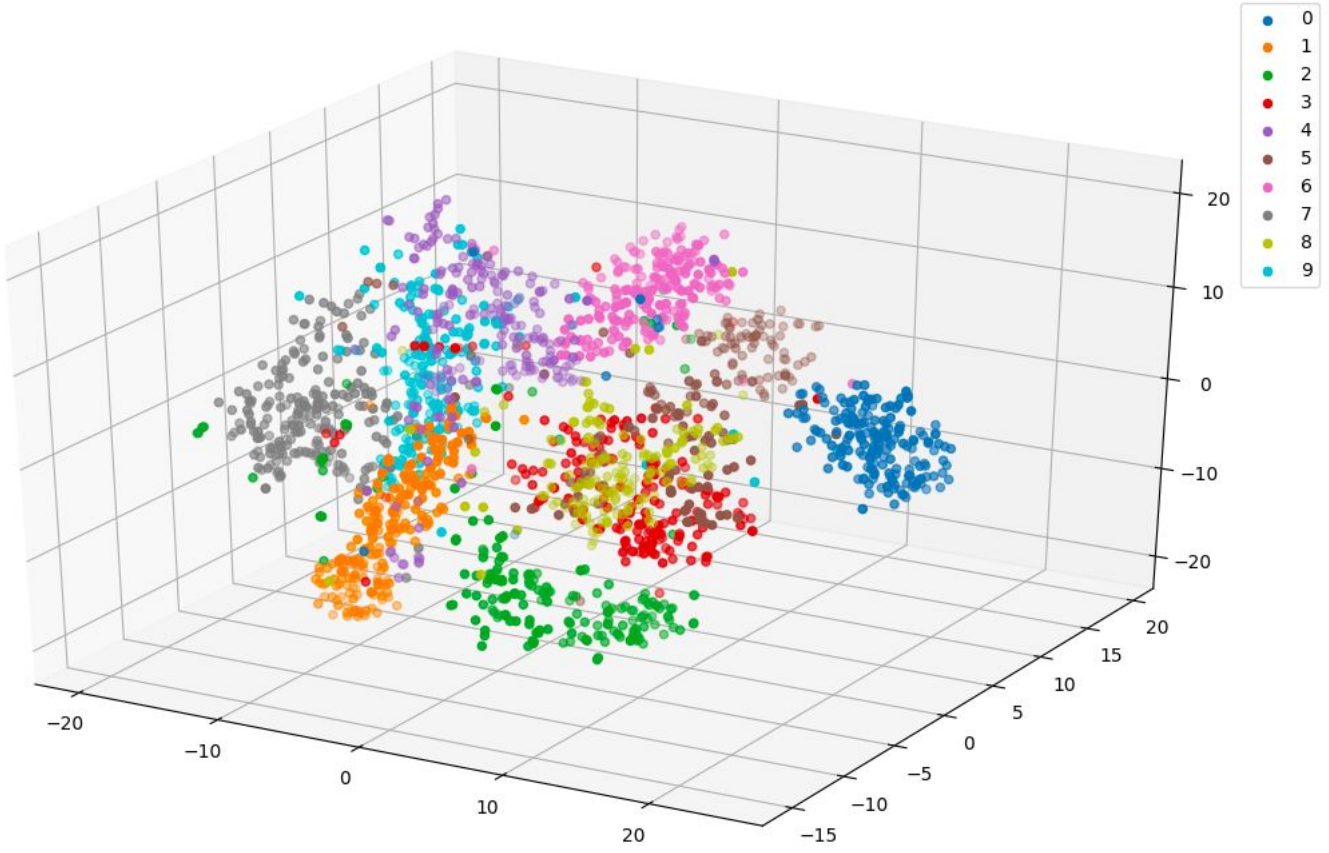
(c):

For this question, I have implemented tsne on dataset_1.mat, where I have kept number of dimensions = 2. Below is the graph :



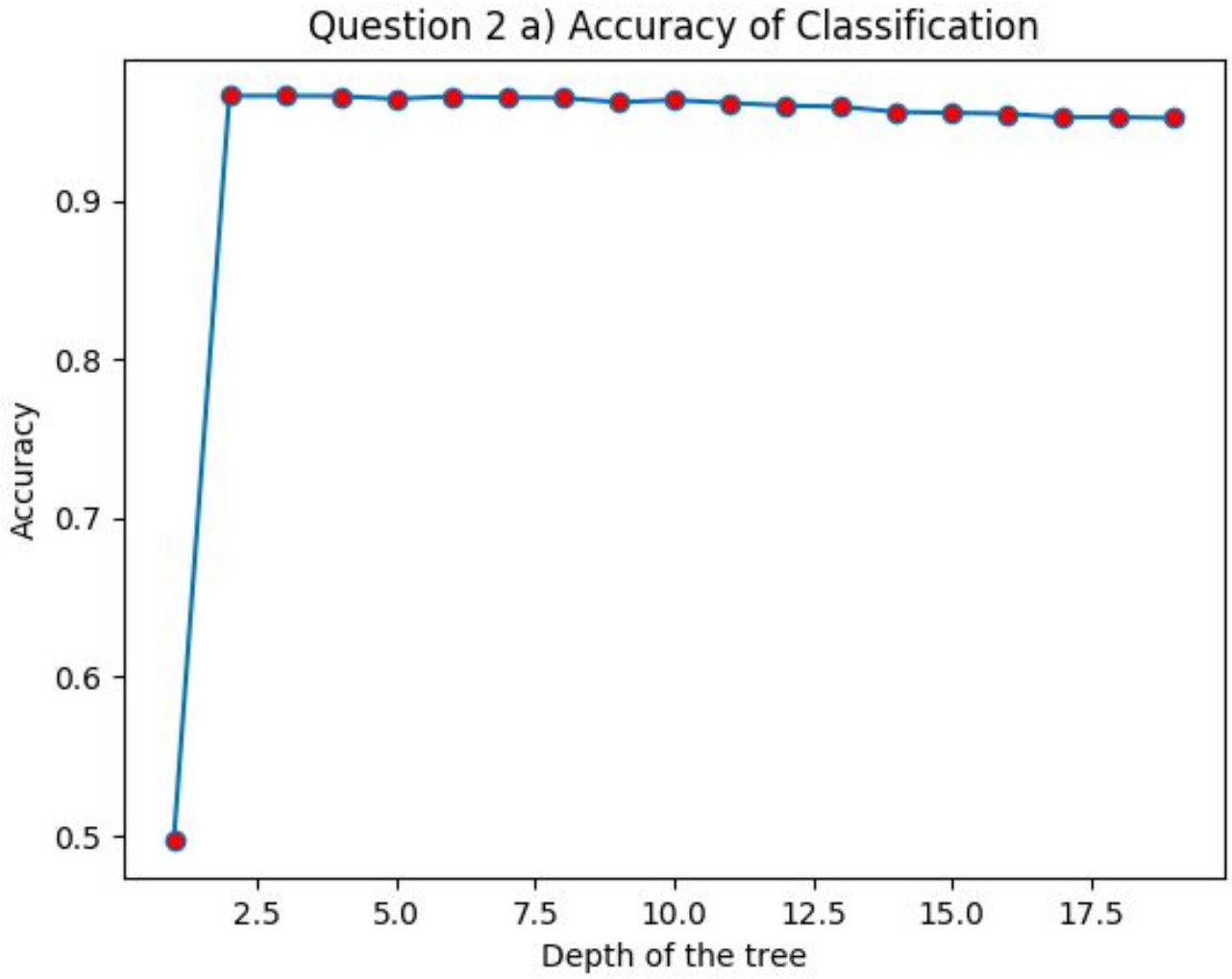
(d):

Similar to c), I have implemented tsne but this time taking 3 components into account : to plot a 3-dimensional graph, below is the graph.



Since, now the dimensions are increased, we can see more distinction between the classes than in 2 dimensional graph.

Question 2 a)



I have plotted a graph of depth of tree vs the accuracy for grid searching the depth of the tree (which is the hyperparameter). It is clear from the graph that accuracy makes a high at around very initial stages of depth of the tree. Around depth of 2 - 3, accuracy is highest, which means that a tree with depth of 2 - 3 is most accurate in predicting the class.

Question 2 b) on next page ⇒

(b)

For this question, I have made a csv file : ‘Q2_b.csv’ (see in submission folder) which contains depth of the tree and the corresponding training accuracy and validation accuracy. Following is the screenshot of the tabel.

Q2_b

Depth	Training Accuracy	Validation Accuracy
1	0.5014285714285714	0.49666666666666665
2	0.9669285714285715	0.9663333333333334
3	0.9669285714285715	0.9663333333333334
4	0.967	0.9661666666666666
5	0.9679285714285715	0.9643333333333334
6	0.9693571428571428	0.966
7	0.9703571428571428	0.9653333333333334
8	0.9727142857142858	0.965
9	0.9746428571428571	0.9626666666666667
10	0.977	0.963
11	0.9797142857142858	0.9618333333333333
12	0.9817142857142858	0.96
13	0.984	0.9596666666666667
14	0.9862857142857143	0.9571666666666667
15	0.9881428571428571	0.9556666666666667
16	0.9895	0.954
17	0.9921428571428571	0.954
18	0.9932142857142857	0.9541666666666667
19	0.9951428571428571	0.9523333333333334

It is very clear from the table that at depth of 2 and 3, validation accuracy is highest and as the the depth is increasing, the validation accuracy is decreasing (although very slightly), whereas when we look at the training accuracy, it is continually increasing with increasing the depth of the tree, which is a clear indication of overfitting.

Depth of 1 : Tree is underfit, as both the training accuracy and validation accuracy is low.

Depth of 2 to 3 : Tree is most accurate, since the validation accuracy is highest

Depth > 4 : Tree is overfit, since training accuracy is increasing but the validation accuracy is going down.

c)

For this question, I have used an inbuilt function 'sklearn.metrics.accuracy_score' instead of using my own accuracy function. I did the same thing I did in b) part but just using the inbuilt function and the results are saved in 'Q2_c.csv' file. Here is a screenshot of the same:

Q2_c

Depth	Training Accuracy	Validation Accuracy
1	0.5014285714285714	0.49666666666666665
2	0.9669285714285715	0.9663333333333334
3	0.9669285714285715	0.9663333333333334
4	0.967	0.9661666666666666
5	0.9679285714285715	0.9643333333333334
6	0.9693571428571428	0.966
7	0.9703571428571428	0.9655
8	0.9727142857142858	0.9651666666666666
9	0.9746428571428571	0.9626666666666667
10	0.977	0.963
11	0.9797142857142858	0.9615
12	0.9815714285714285	0.9603333333333334
13	0.984	0.9585
14	0.9862142857142857	0.957
15	0.988	0.9555
16	0.9897142857142858	0.9548333333333333
17	0.992	0.9525
18	0.9931428571428571	0.953
19	0.9950714285714286	0.9531666666666667

Its again very clear from the table that the accuracy values are almost similar to the b) part table.

Question 3 on next page ⇒

Question 3 a) :

In this question, we are supposed to predict the month in the PM2.5 table, for which I have loaded the data, divided the data into a split of 80:20 for training and testing respectively. Then I have used a decision tree classifier to predict the month.

2 types of decision tree are used : 1) gini index 2) Entropy .

Following is the accuracy for the above described decision trees :

Accuracy of decision tree with :

1) Gini index = 0.3414750957854406 = 34%
2) Entropy = 0.37248563218390807 = 37%

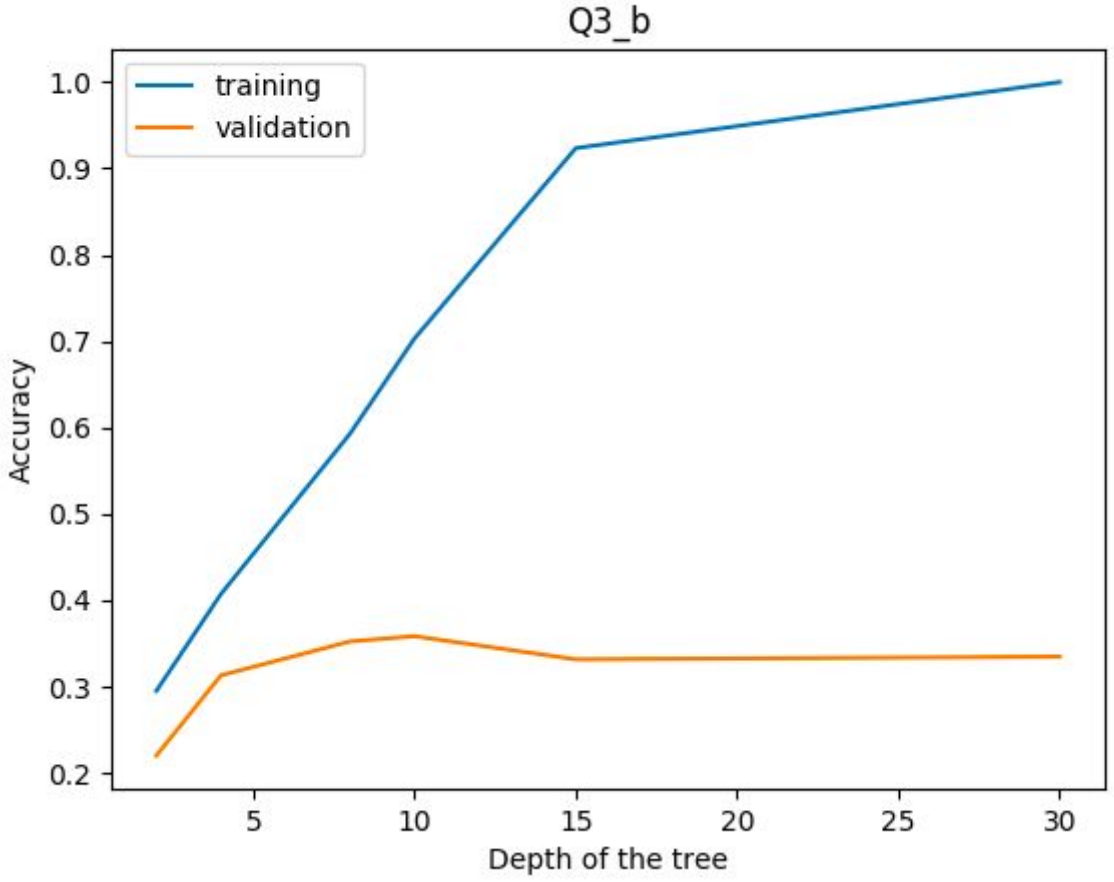
So, looking at the above results, we can say that decision tree with criterion = entropy is performing better than the decision tree with criterion = gini index.

Part b on next page⇒

b)
Here is the accuracy table when decision tree classifier is trained with the described depths:

Q3_b

Depth	Training Accuracy	Validation Accuracy
2	0.29507558748690316	0.22018678160919541
4	0.40709474629546477	0.31297892720306514
8	0.5927256398742703	0.35225095785440613
10	0.703038467295315	0.3545258620689655
15	0.9236341864990271	0.3400383141762452
30	1	0.33321360153256707



Clearly from the above table and graph, depth of 10 is best suited according to validation accuracy.

c) For this question, I have made a list of 100 decision tree classifiers, and then choose the labels based on majority voting. The accuracy has increased in the previous cases.

d)
Similar to the previous part, in this part I have made the decision tree of the given length int the question. Best accurate decision tree is of depth : 10