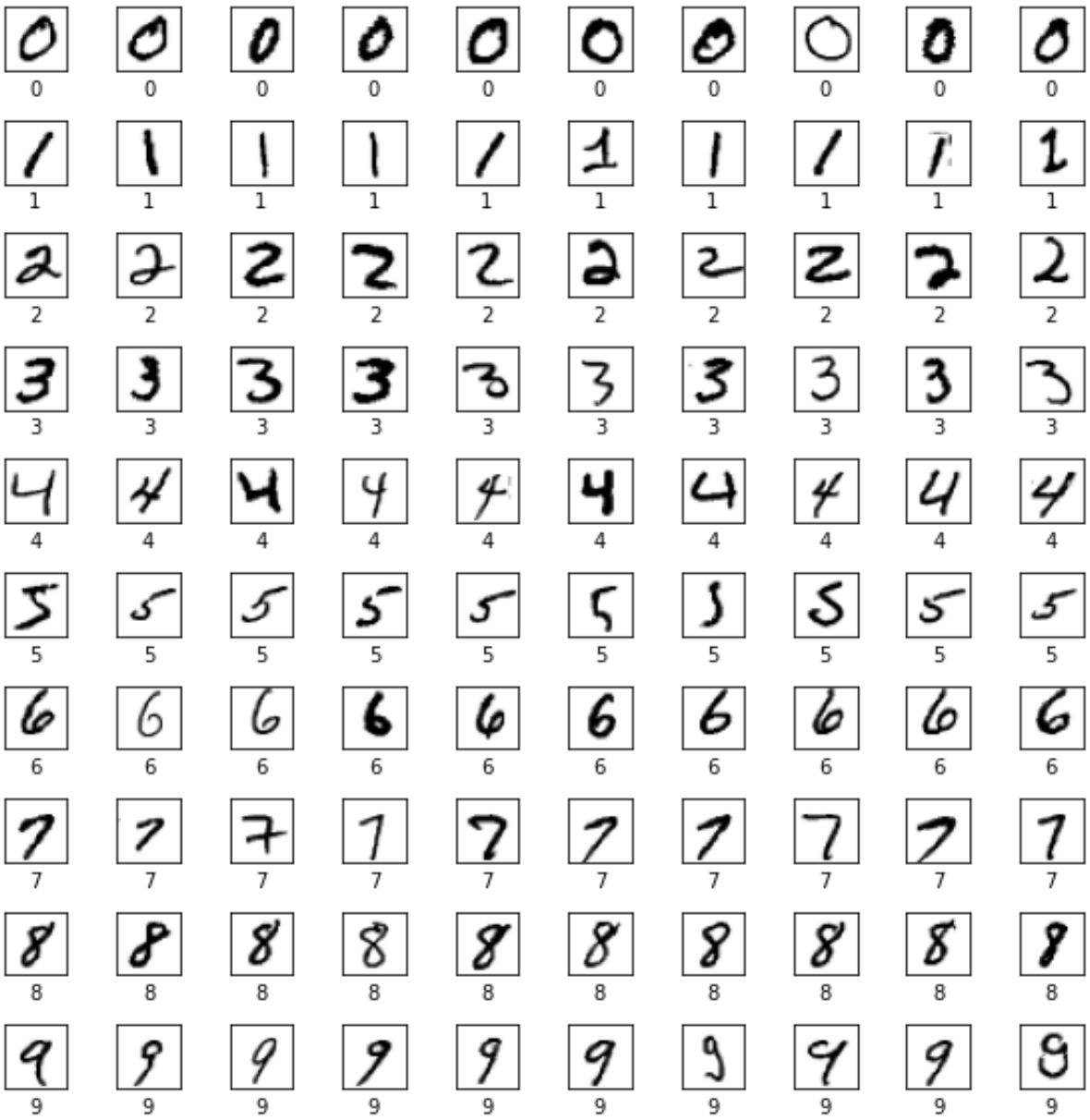
Answer 1 a)

Observations:

By looking at the dataset which is of dimensions (50000, 28, 28) I could infer that there are 50000 28*28 matrices, each representing an image(assumption).

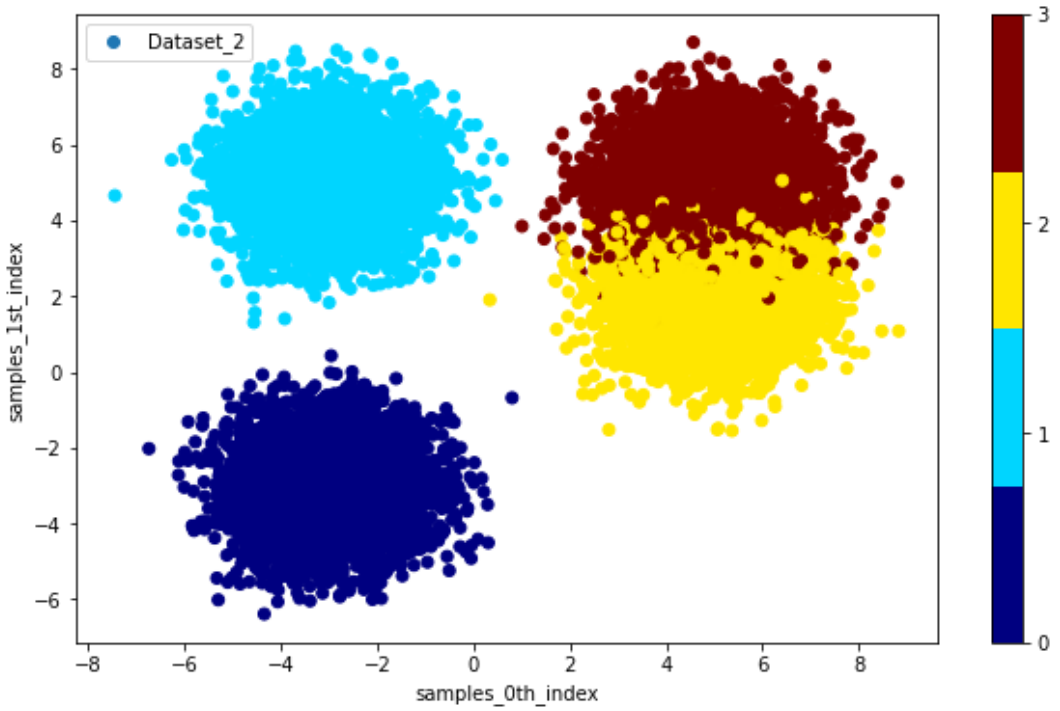
After visualization, I got 10 samples from each class i.e. total 100 out of the 50000 samples given: There are at least 10 distinct samples belonging to each class (0-9), there might be more. The image plot depicts these different samples from each class. Row 1 of the plot depicts 10 samples from class 0, row 2 shows different samples from class 1 and so on.



Answer 1 b)

Inference about the dataset-

This dataset consists of four classes (0-3). Class 0 and class 1 are quite different from each other as well as classes 2 and 3. Classes 2 and 3 overlap in a certain area, which shows that there are similarities in classes 2 and 3.



Answer 1 c)

t-SNE to reduce dataset_1 to 2 dimensions-

(t-SNE) or **T-distributed stochastic neighbour embedding** created in 2008 by **(**Laurens van der Maaten and Geoffrey Hinton) for dimensionality reduction that is particularly well suited for the visualization of high-dimensional datasets.

(t-SNE) takes a high dimensional data set and reduces it to a low dimensional graph that retains a lot of the original information. It does so by giving each data point a location in a two or three-dimensional map. This technique finds clusters in data thereby making sure that an embedding

preserves the meaning in the data. t-SNE reduces dimensionality while trying to keep similar instances close and dissimilar instances apart.

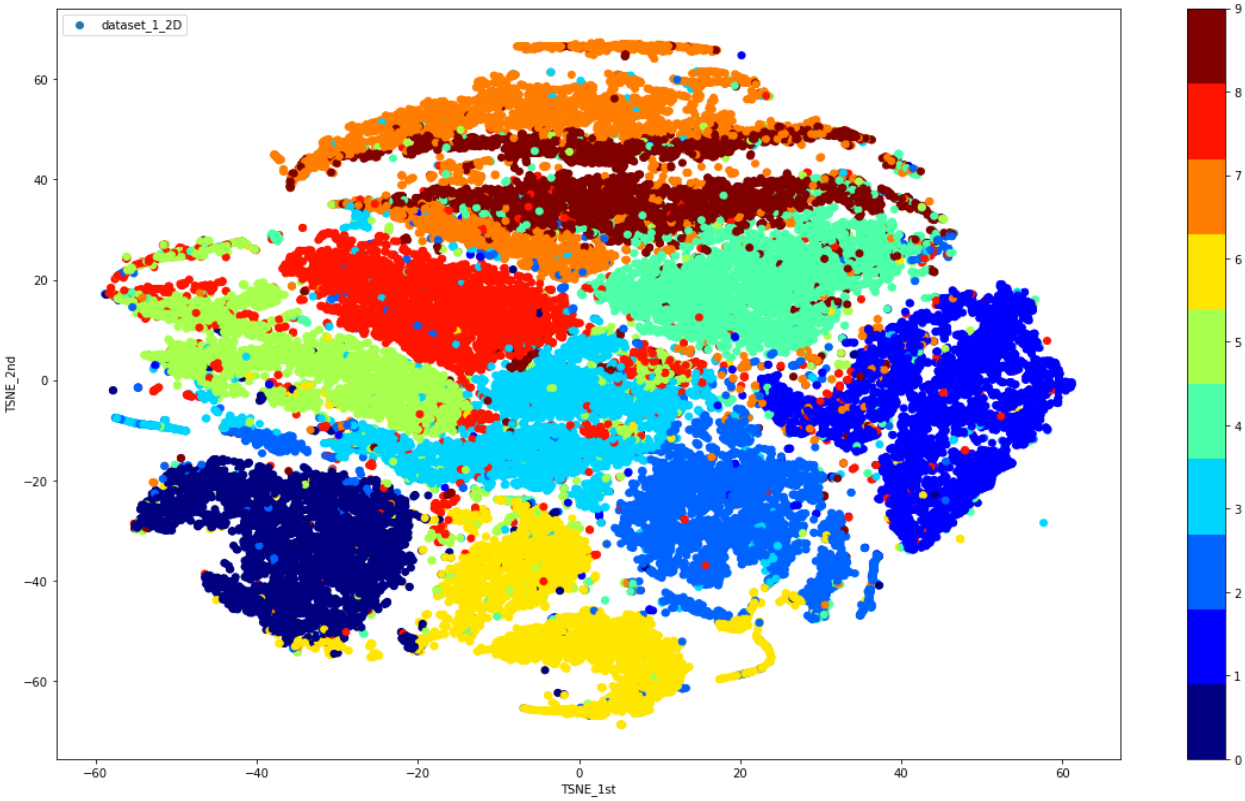
Inference from the plot regarding class separation:

There are 10 classes (0-9).

Though they are all different classes, but there are similarities in some classes , like class 3 and class 8 and class 5 are similar so they are clustered together and have a slight overlap, class 4, class9 and class 7 are similar so they occur in clusters adjacent to each other.

Class 0, class 2, class 1 and class 6 do not overlap with any other class.

Visualization of given dataset through t-SNE 2D.

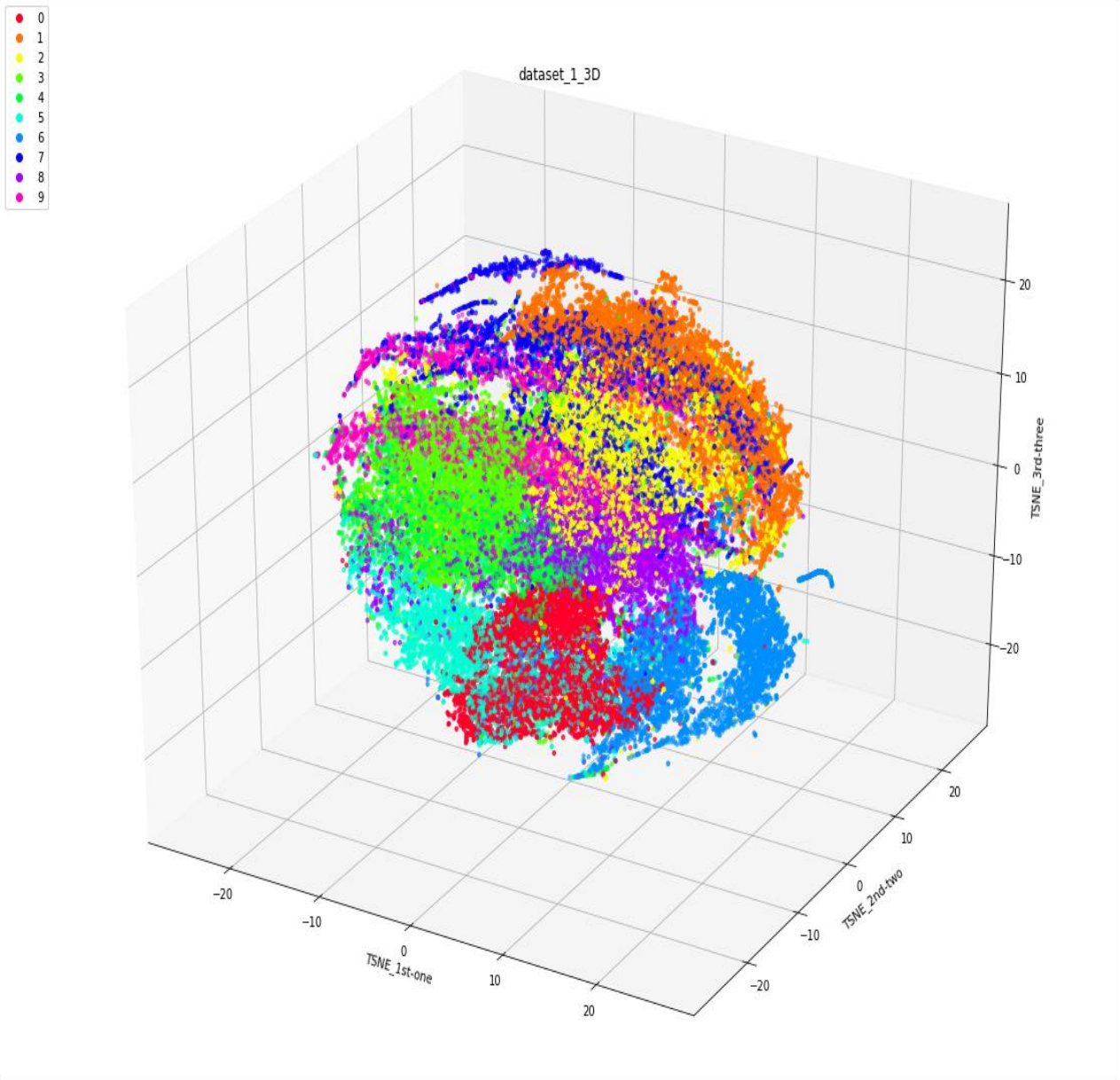


Answer 1 d)

Reduce original dataset_1 to 3 dimensions :

The 3D t-SNE has a richer structure than 2D t-SNE but it does not give more information as compared to above 2D plot. We added an extra dimension to see if it helps recognise clusters more easily than the 2D plot. But it seems there is an added problem of finding the best rotation (viewpoints) to interpret the visualization.

Important distinction in inference as compared to c above- It takes much more computational effort and time .

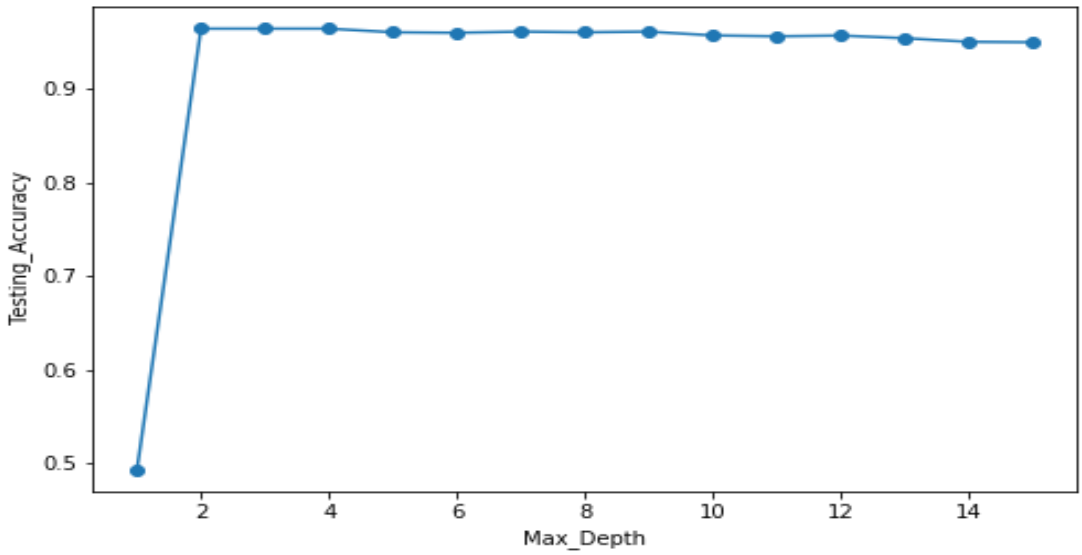


Answer 2 a)

Effect of depth on the performance of the classifier,- the depths considered are from 1 to 15. Tthe accuracy of the classifier is maximum at

depth= 2, 3, 4. It is least at depth=1. After depth =4, the accuracy starts decreasing.

And curve between depth and testing accuracy



The best performance is consistent with the visualization in 1.b, maximum there are four classes in the dataset, therefore after obtaining the best accuracy at depth=2, there is not much difference in performance as class 0 and class 1 can be identified easily as seen from the plot but class 2 and class 3 can't be identified uniquely as they overlap as evident from the scatter plot.

Answer 2 b)

Table with train accuracy and test accuracy for each value of depth from 1 to 15.

Comments on overfitting and underfitting for each entry in the table-

The tree performs best at depth 2, 3, 4.

For depths greater than 3, the training accuracy keeps on increasing but the test accuracy keeps on decreasing, which indicates that the tree starts overfitting for depths greater than 4.

The highest training accuracy is at depth 15 which gives a lower test accuracy, thus indicating overfitting.

	Depth	Train Accuracy	Test Accuracy
0	1.0	0.502429	0.494167
1	2.0	0.966786	0.965833
2	3.0	0.966786	0.965833
3	4.0	0.966857	0.965833
4	5.0	0.967357	0.965333
5	6.0	0.968286	0.965333
6	7.0	0.969000	0.965167
7	8.0	0.970357	0.963667
8	9.0	0.971714	0.962167
9	10.0	0.973714	0.961333
10	11.0	0.975571	0.960500
11	12.0	0.978286	0.959500
12	13.0	0.980214	0.958333
13	14.0	0.982786	0.957667
14	15.0	0.984714	0.957500

Answer 2c)

With sklearn accuracy_score function –Is there any deviation?-No. The values are same as the ones obtained in part b.

	Depth	Train Accuracy	Test Accuracy
0	1.0	0.502429	0.494167
1	2.0	0.966786	0.965833
2	3.0	0.966786	0.965833
3	4.0	0.966857	0.965833
4	5.0	0.967357	0.965333
5	6.0	0.968286	0.965333
6	7.0	0.969000	0.965167
7	8.0	0.970357	0.963667
8	9.0	0.971714	0.962167
9	10.0	0.973714	0.961333
10	11.0	0.975571	0.960500
11	12.0	0.978286	0.959500
12	13.0	0.980214	0.958333
13	14.0	0.982786	0.957667
14	15.0	0.984714	0.957500

Answer 3a)

Trained decision tree on both Gini Index and Entropy:

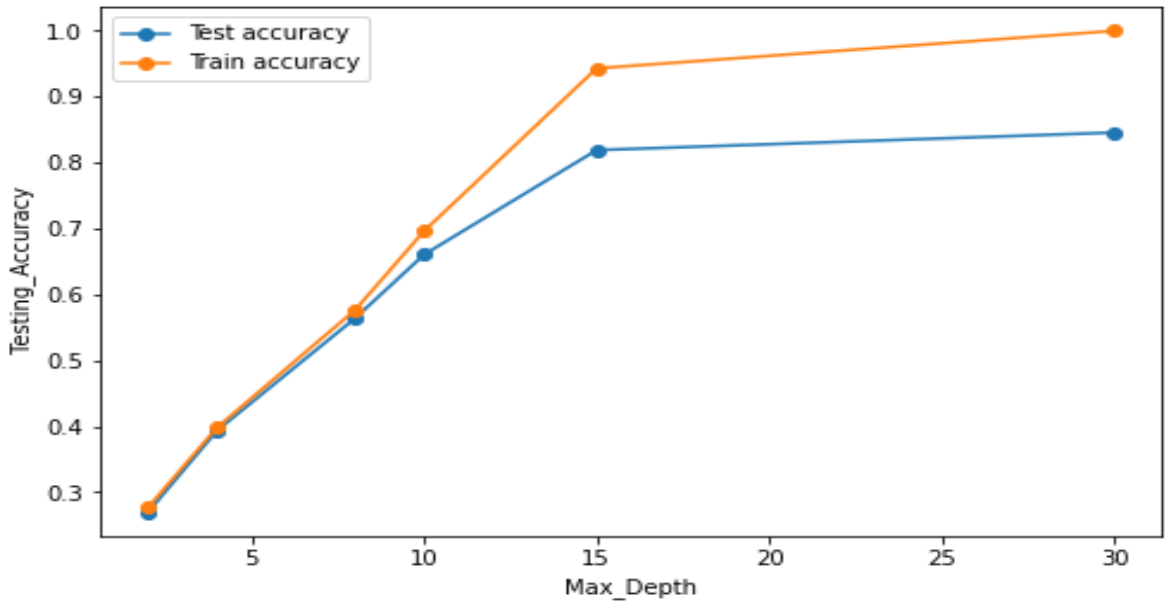
```
Performance criteria = gini  0.8390188248716486
Performance criteria = entropy  0.8450656018254421
```

Better accuracy is given by Entropy.

Answer3 b)

Train decision tree with different depths [2, 4, 8, 10, 15, 30], plot curve between training accuracy and testing accuracy and depth-

	Depth	Train Accuracy	Test Accuracy
0	2.0	0.275650	0.276098
1	4.0	0.397302	0.410839
2	8.0	0.576400	0.562578
3	10.0	0.700334	0.654649
4	15.0	0.946633	0.818597
5	30.0	1.000000	0.844723



Best value of depth as clear from plot is = 30. The tree performs best at depth 30 but also it starts overfitting at depths greater than 2.

Answer 3c)

100 different decision stumps were created of max depth 3. Each stump was trained on 50% randomly selected training data. Now the majority vote was taken from the output labels of the decision stumps to predict the test samples' labels.

Performance effected as compared to part a)

Though the criteria used was entropy but the accuracy at depth 3 that could be achieved from decision stumps is around 35-36%(sometimes 34% too due to random selection of training data each time). In part a) depth was not considered, only Entropy and gini index were considered and accuracy was very high as compared to the accuracy obtained here.

and part b)- the accuracy obtained was close to 34%

Thus, there is just slight difference of 1-2%% at max if we don't do ensembling at depth =3.

Answer 3d)

Tuning decision stumps by changing max-depth and number of trees:

Best performance depth achieved from part b) =30

Here I tuned the decision trees at different depths- 4, 8, 10, 15, 20, 30.

And different number of trees - [100,150,200,250,300]

Observations:

The test accuracy at depth 4 is almost same(41%) or sometimes less due to random selection of data for each stump every time.

The test accuracy at depth 8 has a significant increase due to the ensembling technique- from 56% to 61%.

The test accuracy at depth 10 also increases from 65% to 75% due to ensembling.

The test accuracy at depth 15 also increases from 81% to 91% due to ensembling.

The test accuracy achieves its maximum at depth 15 which is 92 %.

Also, it is noticed that for depth =15 and greater ,the accuracy increases with increasing number of decision stumps, but the increase is not very significant.

	Srno	Depth	Number of Trees	Train Accuracy	Test Accuracy
0	1.0	4.0	100.0	0.401808	0.396920
1	2.0	4.0	150.0	0.404604	0.397946
2	3.0	4.0	200.0	0.403834	0.399886
3	4.0	4.0	250.0	0.404375	0.398973
4	5.0	4.0	300.0	0.404632	0.398745
5	6.0	8.0	100.0	0.638124	0.614604
6	7.0	8.0	150.0	0.638951	0.615288
7	8.0	8.0	200.0	0.640691	0.613349
8	9.0	8.0	250.0	0.641661	0.615060
9	10.0	8.0	300.0	0.641747	0.614832
10	11.0	10.0	100.0	0.806269	0.751284
11	12.0	10.0	150.0	0.808922	0.753679
12	13.0	10.0	200.0	0.807724	0.751626
13	14.0	10.0	250.0	0.808066	0.751854
14	15.0	10.0	300.0	0.809293	0.752310
15	16.0	15.0	100.0	0.994267	0.911694
16	17.0	15.0	150.0	0.994581	0.914661
17	18.0	15.0	200.0	0.994295	0.910439
18	19.0	15.0	250.0	0.994552	0.911922
19	20.0	15.0	300.0	0.995379	0.912835
20	21.0	20.0	100.0	0.999658	0.923788
21	22.0	20.0	150.0	0.999829	0.922875
22	23.0	20.0	200.0	0.999886	0.924016
23	24.0	20.0	250.0	0.999943	0.925271
24	25.0	20.0	300.0	0.999857	0.926298
25	26.0	30.0	100.0	0.999572	0.921962
26	27.0	30.0	150.0	0.999772	0.925613
27	28.0	30.0	200.0	0.999857	0.924815
28	29.0	30.0	250.0	0.999886	0.926184
29	30.0	30.0	300.0	0.999943	0.924130

Comparison of the results of the classification models created above on the test set.-

- Ensembling with 100 or more decision stumps at depth=20 gives the best accuracy with overfitting.
- Entropy criteria gives the second best accuracy.
- Gini Index gives the third best accuracy.
- If only depth is to be considered then the fourth best accuracy is at depth= 30 without ensembling which is 84%, but the tree shows overfitting as seen from the training accuracy.

Therefore, the above models can be ranked in the following way:

1. Ensembling with depth =20 and 100 decision stumps or more as ensembling doesn't perform significantly better for depths less than 8.
2. Tree modelled on the criteria of Entropy.
3. Tree modelled on the criteria of Gini Index.
4. Tree modelled by just tuning various depths.