COL774 Assignment 3

Aman Hassan
2021CS50607

November 2023

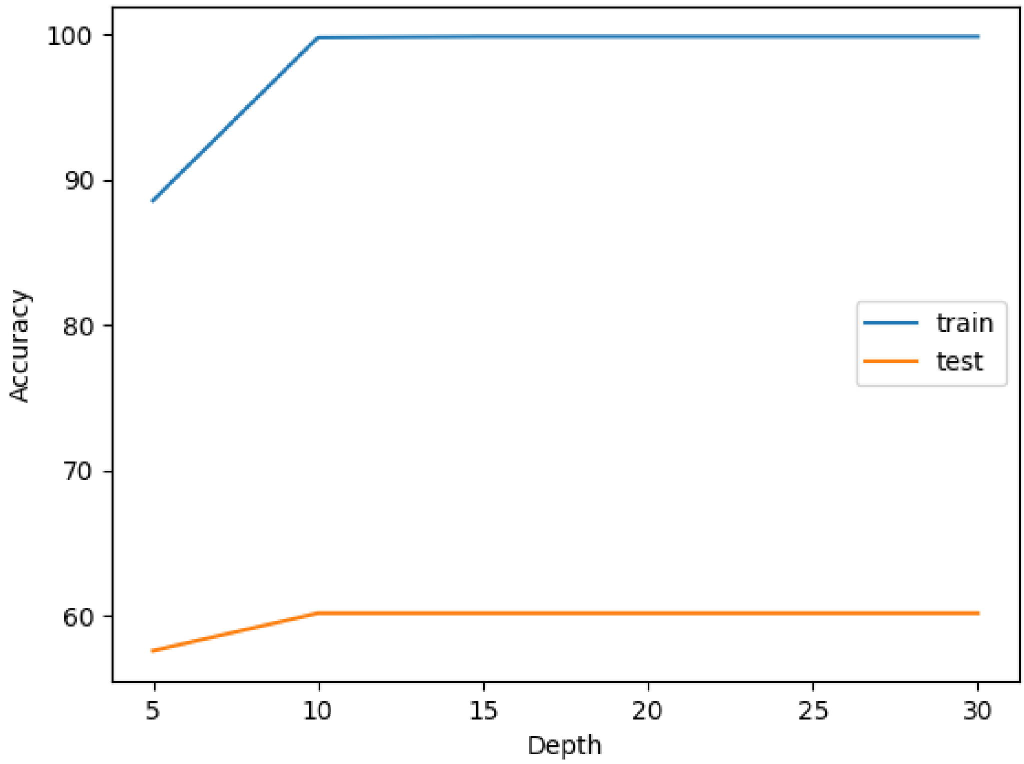
1 Decision Tree

(a) Here we constructed the Decision Tree for varying depths where features to split are determined using highest mutual information metric

i.
- Only Win:
  - Accuracy for in prediction type of only win on training set is 50.3380
  - Accuracy for in prediction type of only win on test set is 49.6380
- Only Loss
  - Accuracy for in prediction type of only loss on training set is 49.6614
  - Accuracy for in prediction type of only loss on test set is 50.3619
- DT with varying depths on training set:
  - Accuracy for depth 5 on training set is 88.5652
  - Accuracy for depth 10 on training set is 99.7828
  - Accuracy for depth 15 on training set is 99.8466
  - Accuracy for depth 20 on training set is 99.8466
  - Accuracy for depth 25 on training set is 99.8466
  - Accuracy for depth 30 on training set is 99.8466
- DT with varying depths on test set:
  - Accuracy for depth 5 on test set is 57.6008
  - Accuracy for depth 10 on test set is 60.1861
  - Accuracy for depth 15 on test set is 60.1861
  - Accuracy for depth 20 on test set is 60.1861
  - Accuracy for depth 25 on test set is 60.1861
  - Accuracy for depth 30 on test set is 60.1861

From the data obtained we find that single type prediction(only win, only loss) performs worse compared to Decision Tree classification (DT is almost 2x better in training prediction). As we expect the training accuracy is much better than test accuracy. We also find that the accuracy is almost the same after depth 10-15 (This can be attributed to the aggressive terminating conditions applied on grow_tree / fit function)

ii. The following Accuracy vs depth graph was obtained

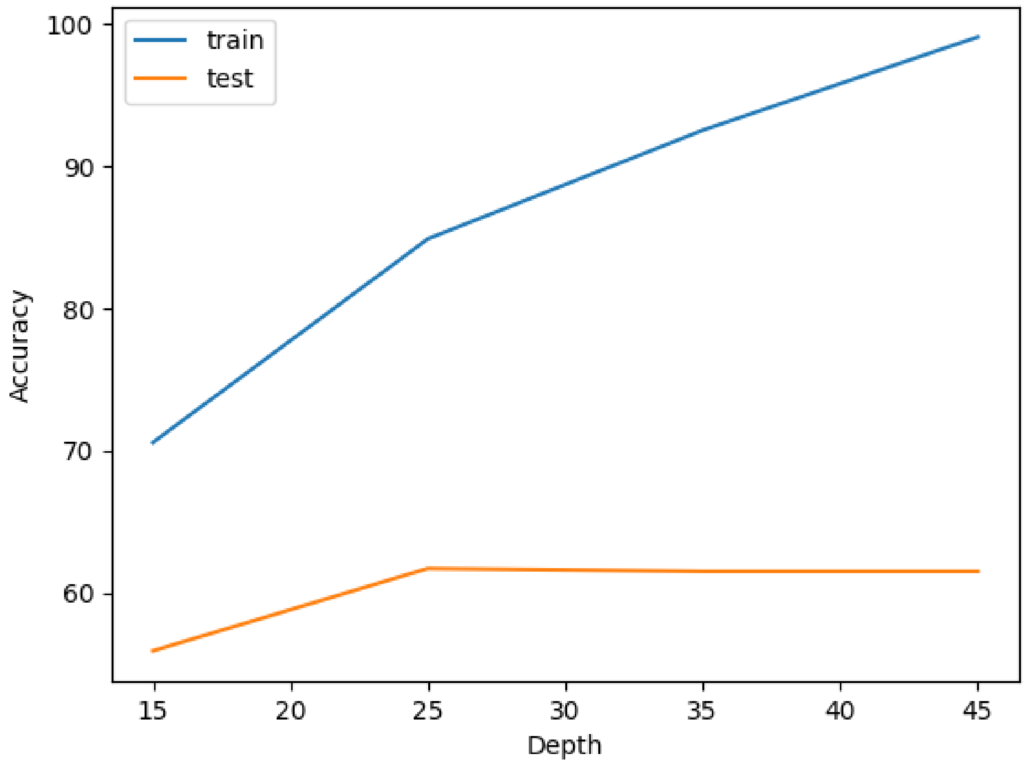


(b) Using one-hot encoding we obtain the following results

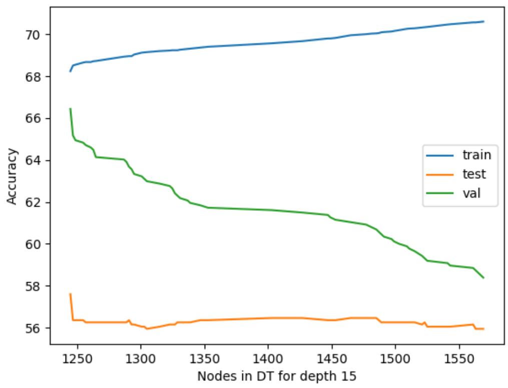
i. • Only Win:
    – Accuracy for in prediction type of only win on training set is 50.3386
    – Accuracy for in prediction type of only win on test set is 49.6381
  • Only Loss
    – Accuracy for in prediction type of only loss on training set is 49.6614
    – Accuracy for in prediction type of only loss on test set is 50.3619
  • DT with varying depths on training set:
    – Accuracy for depth 15 on training set is 70.6018
    – Accuracy for depth 25 on training set is 84.9112
    – Accuracy for depth 35 on training set is 92.5514
    – Accuracy for depth 45 on training set is 99.1057
  • DT with varying depths on test set:
    – Accuracy for depth 15 on test set is 55.9462
    – Accuracy for depth 25 on test set is 61.7373
    – Accuracy for depth 35 on test set is 61.5305
    – Accuracy for depth 45 on test set is 61.5305

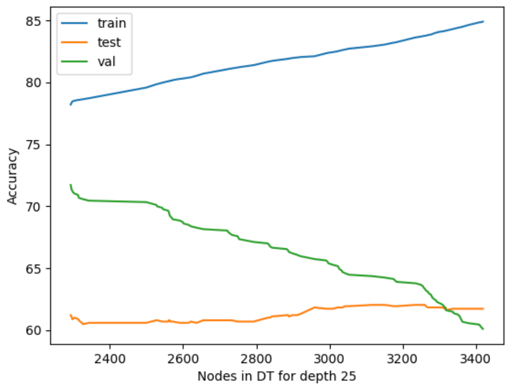
From the data obtained we find that Decision Tree classification performs better compared to single type prediction(only win, only loss). As we expect the training accuracy is much better than test accuracy for a given depth. However contrary to part (a) we find that here the accuracies significantly increase as we increase the depth for the case of training set and for the test set, it increases from 15 to 25, decreases from 25 to 35 and then remains same for the next increment of depth.

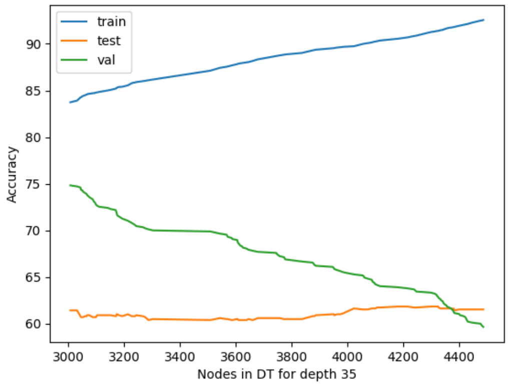
ii. The following Accuracy vs depth graph was obtained

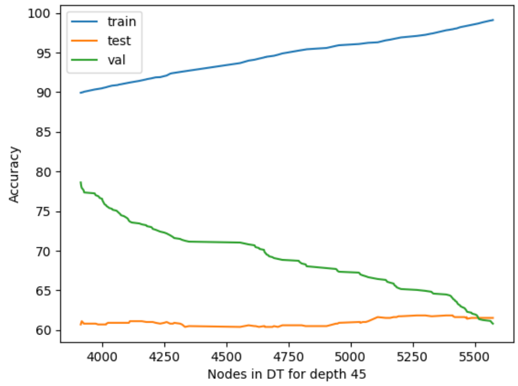


(c) The following Accuracy vs nodes graphs were obtained upon performing reduced error pruning for various depths









Some observations to note:

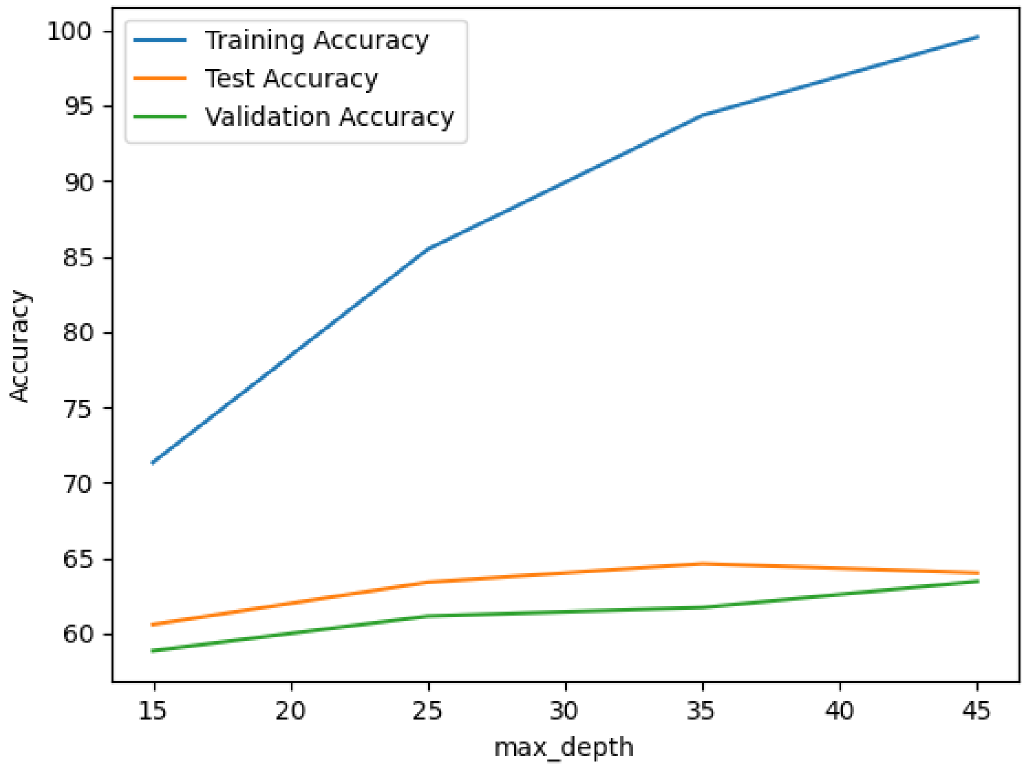
- In all graphs the training accuracy decreases as number of nodes reduce in the graph (which suggests that more nodes could possibly lead to overfitting)
- Both the validation increases as number of nodes reduce in the graph
- we find that the test accuracies remain around the same value throughout the process
- In the initial stage of the DT (before pruning) we find the following order of accuracies:
  - For depth 15: train > val > test accuracy
  - For all other depths: train > test > val accuracy
- As more nodes are pruned we find the following order of accuracies: train > val > test

(d) Decision Tree using sci-kit learn

i. Varying Max-Depth

- Training Set Accuracies:
  - Training Accuracy for max_depth = 15 is 71.3428
  - Training Accuracy for max_depth = 25 is 85.4734
  - Training Accuracy for max_depth = 35 is 94.3529
  - Training Accuracy for max_depth = 45 is 99.5528
- Test Set Accuracies:
  - Test Accuracy for max_depth = 15 is 60.5998
  - Test Accuracy for max_depth = 25 is 63.3919
  - Test Accuracy for max_depth = 35 is 64.6329
  - Test Accuracy for max_depth = 45 is 64.0124

The obtained graph is as follows:

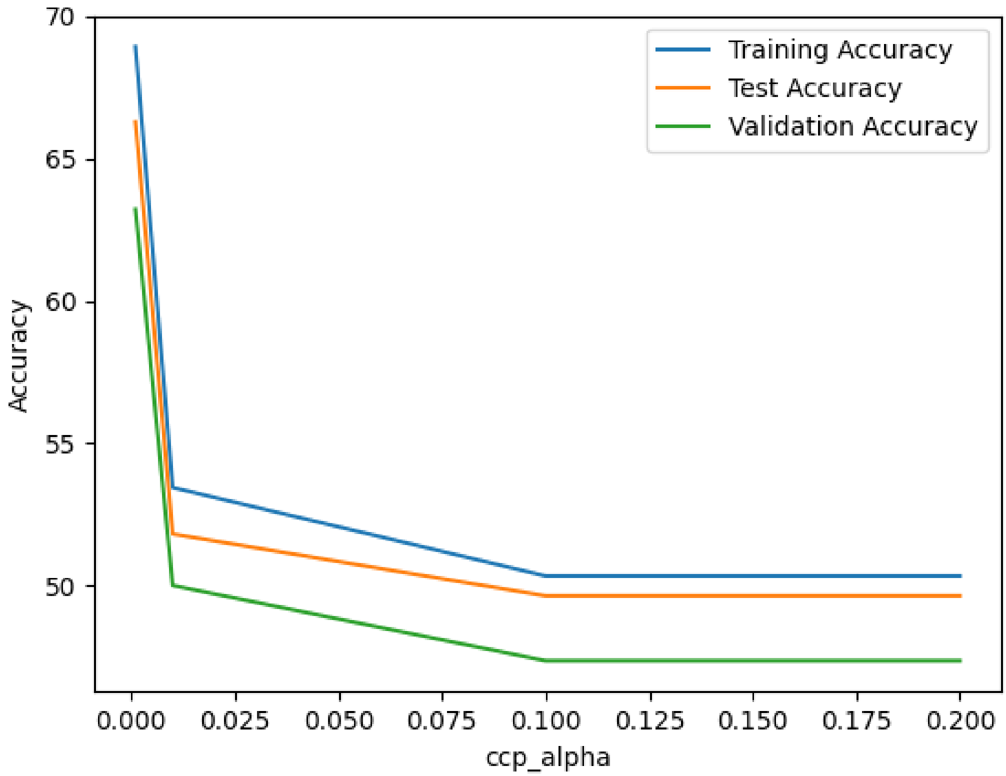


We find that the best max_depth obtained using the validation set is 45

ii. Varying ccp_alpha

- Training Set Accuracies:
  - Training Accuracy for ccp_alpha = 0.001 is 68.9408
  - Training Accuracy for ccp_alpha = 0.01 is 53.4432
  - Training Accuracy for ccp_alpha = 0.1 is 50.3386
  - Training Accuracy for ccp_alpha = 0.2 is 50.3386
- Test Set Accuracies:
  - Test Accuracy for ccp_alpha = 0.001 is 66.2875
  - Test Accuracy for ccp_alpha = 0.01 is 51.8097
  - Test Accuracy for ccp_alpha = 0.1 is 49.6381
  - Test Accuracy for ccp_alpha = 0.2 is 49.6381

The obtained graph is as follows:



We find that the best ccp_alpha obtained using the validation set is 0.001

iii. Observations to note:

- We find that the training data prediction accuracy is lesser in the sci-kit learn model compared to the model developed in both part b and c
- On the other hand we find that the test data prediction accuracy is higher in the sci-kit model as compared to the model developed in both b and c

(e) Random Forests: Using out of box accuracies and grid search over the parameter space we observe the following result:

- Best Parameters:
  - max_features: 0.7
  - min_samples_split: 8
  - n_estimators: 150
- Out of Box Accuracy: 71.8922
- Training Accuracy: 98.7990
- Test Accuracy: 71.7684
- Validation Accuracy: 69.5402

Compared to the previous parts, we find that the training, test and validation accuracy obtained here is higher than what was obtained in part d. However validation accuracy is lesser than what was obtained in part c (Training and Test accuracy are higher)

2 Neural Networks

(a) A general Neural network architecture was created with configurable parameters such as:

- Mini-batch size: M
- Number of features: n
- Hidden layer architecture: [h1, h2, h3, ...]
- Target labels: r (number of target labels [1, 2, 3, ... r])
- Activation function: sigmoid, relu
- Learning rate, η: constant, adaptive

The terminating condition was fixed as 200 epochs. An intermediate condition of moving average of loss value was also kept with the threshold as 0.001.(Although this intermediate condition was less often used since error nearly never reached 0.001 difference)

(b) Varying number of units in 1 layer: The following tables were obtained for various number of units:

i. 1 unit

Class	Precision	Recall	F1Score	Support
1	0.98	0.73	0.84	2657
2	0.52	0.69	0.60	1491
3	0.49	0.62	0.55	1548
4	0.25	0.49	0.33	1030
5	0.93	0.59	0.72	3274

Table 1: Training

Class	Precision	Recall	F1Score	Support
1	0.98	0.75	0.85	302
2	0.50	0.70	0.58	141
3	0.45	0.62	0.52	143
4	0.29	0.48	0.36	113
5	0.94	0.58	0.72	301

Table 2: Test

ii. 5 units

Class	Precision	Recall	F1Score	Support
1	0.92	0.86	0.89	2098
2	0.66	0.73	0.69	1788
3	0.58	0.58	0.58	1969
4	0.58	0.51	0.54	2288
5	0.68	0.76	0.72	1857

Table 3: Train

Class	Precision	Recall	F1Score	Support
1	0.92	0.90	0.91	233
2	0.65	0.70	0.67	182
3	0.54	0.58	0.56	184
4	0.64	0.49	0.55	247
5	0.60	0.73	0.66	154

Table 4: test

iii. 10 units

Class	Precision	Recall	F1Score	Support
1	0.91	0.87	0.89	2066
2	0.72	0.70	0.71	2020
3	0.62	0.58	0.60	2097
4	0.55	0.52	0.53	2136
5	0.63	0.78	0.70	1681

Table 5: Train

Class	Precision	Recall	F1Score	Support
1	0.91	0.90	0.90	231
2	0.70	0.69	0.69	201
3	0.58	0.58	0.58	197
4	0.59	0.49	0.53	228
5	0.57	0.74	0.64	143

Table 6: test

iv. 50 units

Class	Precision	Recall	F1Score	Support
1	0.91	0.87	0.89	2068
2	0.68	0.73	0.70	1849
3	0.58	0.59	0.59	1923
4	0.50	0.54	0.52	1861
5	0.79	0.72	0.75	2299

Table 7: Train

Class	Precision	Recall	F1Score	Support
1	0.91	0.90	0.90	231
2	0.66	0.71	0.68	182
3	0.57	0.60	0.58	188
4	0.56	0.51	0.53	206
5	0.72	0.69	0.71	193

Table 8: test

v. 100 units

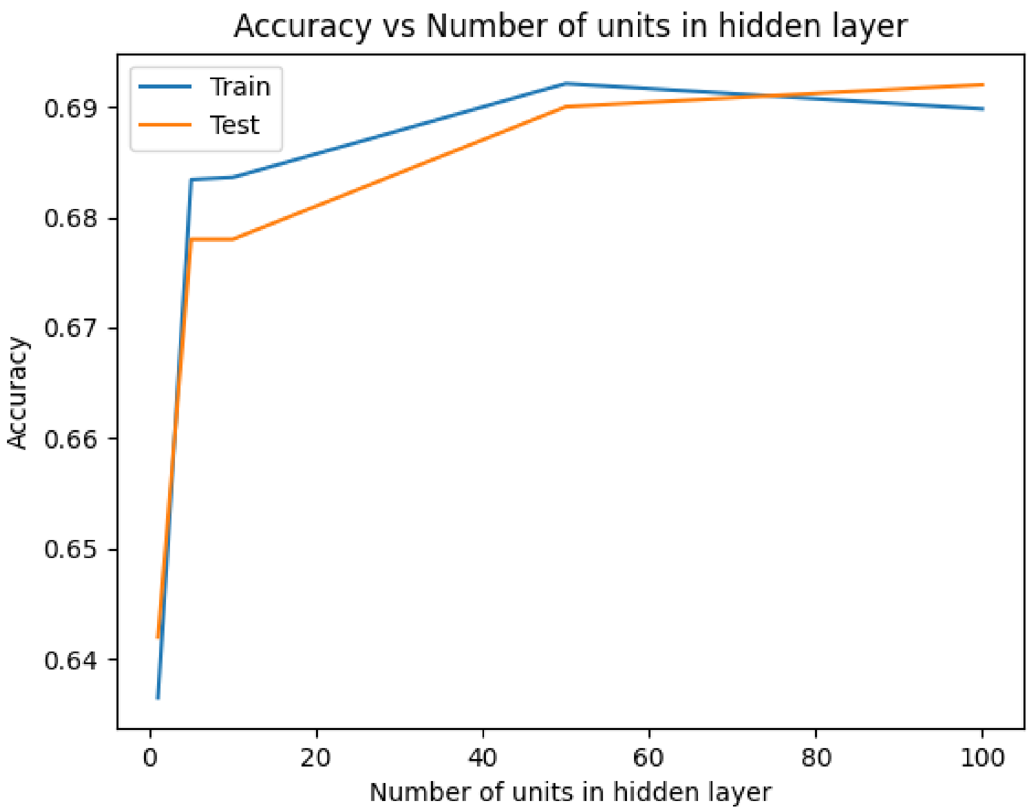
Class	Precision	Recall	F1Score	Support
1	0.88	0.90	0.89	1937
2	0.68	0.72	0.70	1870
3	0.58	0.58	0.58	1941
4	0.51	0.53	0.52	1914
5	0.80	0.71	0.75	2338

Table 9: Train

Class	Precision	Recall	F1Score	Support
1	0.89	0.93	0.91	220
2	0.67	0.72	0.69	185
3	0.56	0.60	0.58	186
4	0.58	0.51	0.54	212
5	0.73	0.69	0.71	197

Table 10: test

We obtain the following graph:



(c) Varying number of layers: The following tables were obtained for various number of layers:

i. 1 layer

Class	Precision	Recall	F1Score	Support
1	0.91	0.88	0.89	2039
2	0.70	0.71	0.71	1955
3	0.60	0.58	0.59	1989
4	0.53	0.53	0.53	1980
5	0.73	0.75	0.74	2037

Table 11: train

Class	Precision	Recall	F1Score	Support
1	0.91	0.90	0.91	230
2	0.67	0.70	0.69	190
3	0.55	0.58	0.57	190
4	0.57	0.49	0.53	218
5	0.66	0.72	0.69	172

Table 12: test

ii. 2 layers

Class	Precision	Recall	F1Score	Support
1	0.93	0.86	0.89	2123
2	0.72	0.70	0.71	2035
3	0.57	0.59	0.58	1909
4	0.51	0.53	0.52	1938
5	0.71	0.74	0.72	1995

Table 13: train

Class	Precision	Recall	F1Score	Support
1	0.93	0.90	0.91	236
2	0.71	0.68	0.70	206
3	0.52	0.58	0.55	178
4	0.56	0.49	0.52	211
5	0.65	0.72	0.68	169

Table 14: test

iii. 3 layers

Class	Precision	Recall	F1Score	Support
1	0.89	0.89	0.89	1985
2	0.71	0.72	0.71	1961
3	0.56	0.60	0.58	1845
4	0.42	0.53	0.47	1584
5	0.84	0.67	0.75	2625

Table 15: train

Class	Precision	Recall	F1Score	Support
1	0.90	0.92	0.91	223
2	0.70	0.71	0.70	197
3	0.54	0.61	0.57	175
4	0.47	0.50	0.48	176
5	0.80	0.66	0.72	229

Table 16: test

iv. 4 layers

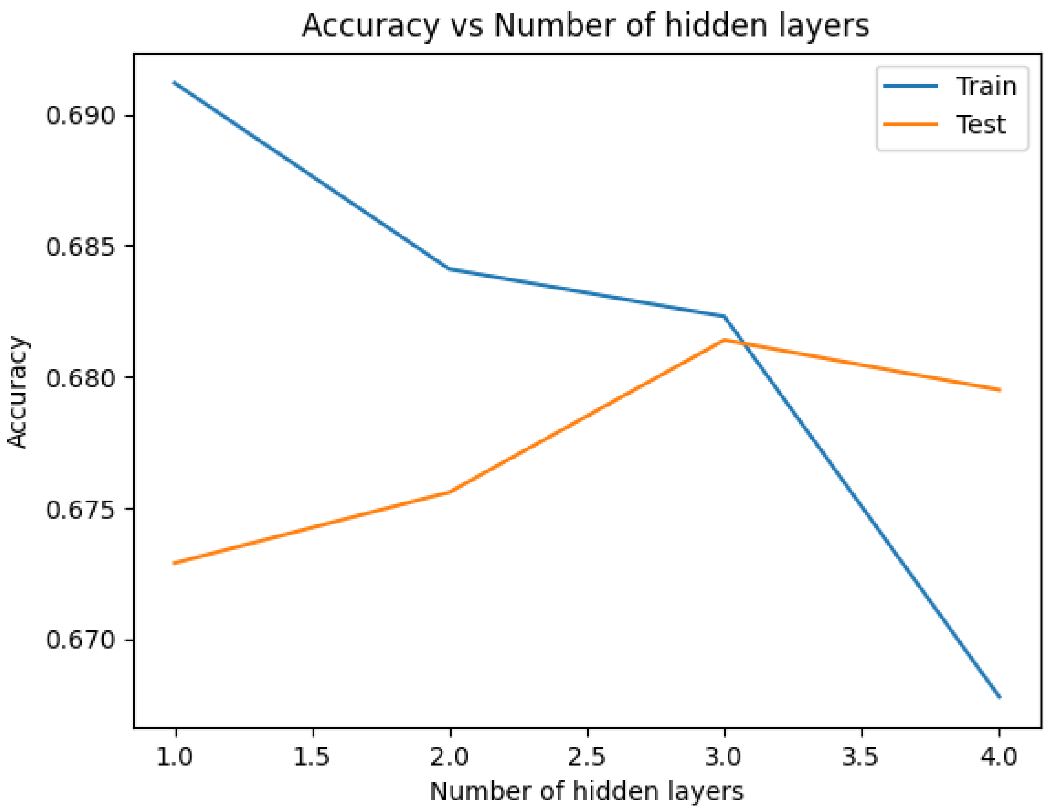
Class	Precision	Recall	F1Score	Support
1	0.85	0.91	0.88	1847
2	0.67	0.70	0.68	1893
3	0.51	0.57	0.54	1750
4	0.84	0.66	0.74	2674
5	0.84	0.67	0.75	2625

Table 17: train

Class	Precision	Recall	F1Score	Support
1	0.86	0.93	0.90	212
2	0.66	0.70	0.68	187
3	0.51	0.59	0.55	174
4	0.51	0.49	0.50	195
5	0.81	0.66	0.73	232

Table 18: test

We obtain the following graph:



(d) Adaptive Learning Rate The following tables were obtained for various number of layers (Note in this case the max number of epochs had to be increased to 500 since the theta updates became smaller as epochs increased):

i. 1 layer

Class	Precision	Recall	F1Score	Support
1	0.91	0.83	0.87	2152
2	0.62	0.66	0.64	1843
3	0.50	0.52	0.51	1873
4	0.44	0.48	0.46	1848
5	0.74	0.68	0.71	2284

Table 19: train

Class	Precision	Recall	F1Score	Support
1	0.90	0.85	0.88	241
2	0.58	0.64	0.61	181
3	0.47	0.53	0.50	177
4	0.49	0.45	0.47	204
5	0.70	0.66	0.68	197

Table 20: test

ii. 2 layers

Class	Precision	Recall	F1Score	Support
1	0.90	0.81	0.85	2177
2	0.61	0.63	0.62	1895
3	0.47	0.50	0.49	1831
4	0.38	0.45	0.42	1688
5	0.75	0.65	0.70	2409

Table 21: train

Class	Precision	Recall	F1Score	Support
1	0.90	0.82	0.86	250
2	0.56	0.61	0.58	180
3	0.47	0.52	0.50	180
4	0.42	0.44	0.43	179
5	0.71	0.63	0.67	211

Table 22: test

iii. 3 layers

Class	Precision	Recall	F1Score	Support
1	0.89	0.89	0.89	1985
2	0.71	0.72	0.71	1961
3	0.56	0.60	0.58	1845
4	0.42	0.53	0.47	1584
5	0.84	0.67	0.75	2625

Table 23: train

Class	Precision	Recall	F1Score	Support
1	0.90	0.92	0.91	223
2	0.70	0.71	0.70	197
3	0.54	0.61	0.57	175
4	0.47	0.50	0.48	176
5	0.80	0.66	0.72	229

Table 24: test

iv. 4 layers

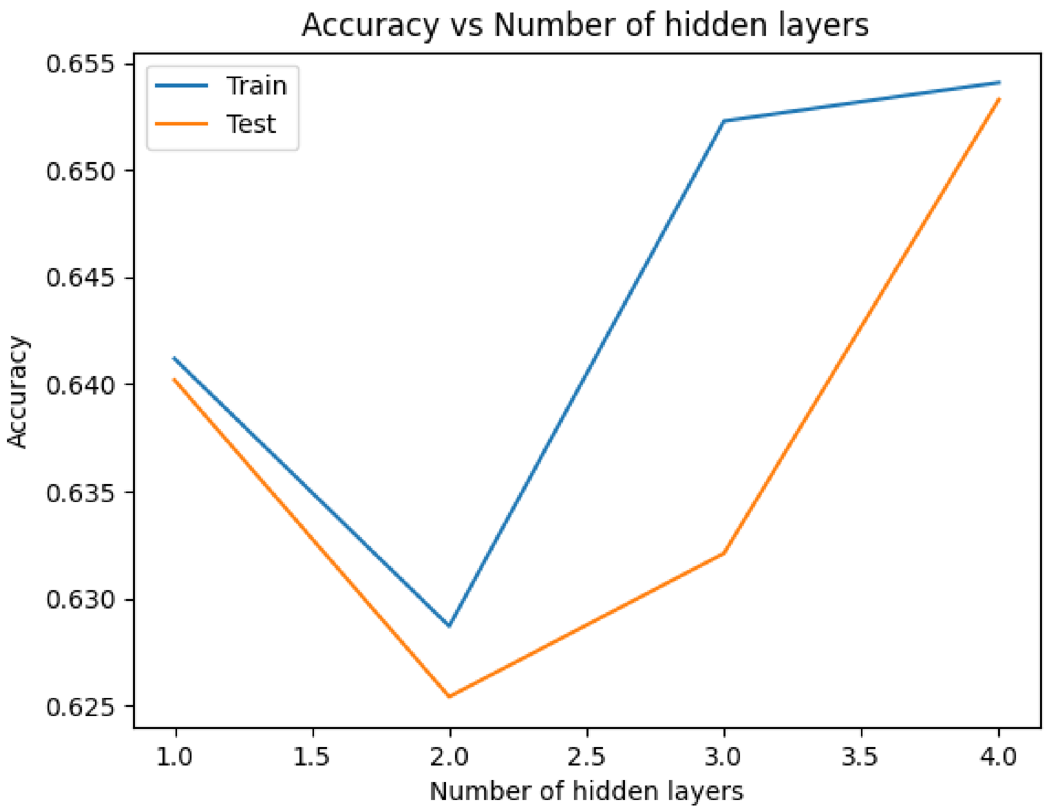
Class	Precision	Recall	F1Score	Support
1	0.85	0.91	0.88	1847
2	0.67	0.70	0.68	1893
3	0.51	0.57	0.54	1750
5	0.84	0.66	0.74	2674
5	0.84	0.67	0.75	2625

Table 25: train

Class	Precision	Recall	F1Score	Support
1	0.86	0.93	0.90	212
2	0.66	0.70	0.68	187
3	0.51	0.59	0.55	174
4	0.51	0.49	0.50	195
5	0.81	0.66	0.73	232

Table 26: test

We obtain the following graph:



We find that training takes a bit more time with adaptive learning rate. It was also observed that the accuracy was slightly worse with adaptive learning rate (Possible reason could be that the learning rate was too small and hence the model was not able to converge to a good solution in the given number of epochs)

(e) ReLU Activation function The following tables were obtained for various number of layers:

i. 1 layer

Class	Precision	Recall	F1Score	Support
1	0.92	0.86	0.89	2092
2	0.68	0.71	0.70	1912
3	0.56	0.58	0.57	1880
4	0.53	0.53	0.53	1880
5	0.75	0.74	0.74	2108

Table 27: train

Class	Precision	Recall	F1Score	Support
1	0.92	0.89	0.91	237
2	0.66	0.70	0.68	185
3	0.53	0.58	0.55	181
4	0.56	0.48	0.51	218
5	0.66	0.69	0.68	179

Table 28: test

ii. 2 layers

Class	Precision	Recall	F1Score	Support
1	0.95	0.88	0.91	2134
2	0.74	0.75	0.74	1947
3	0.64	0.61	0.63	2025
4	0.52	0.57	0.54	1813
5	0.75	0.75	0.75	2081

Table 29: train

Class	Precision	Recall	F1Score	Support
1	0.95	0.91	0.93	238
2	0.71	0.73	0.72	191
3	0.60	0.59	0.60	200
4	0.55	0.52	0.54	197
5	0.68	0.73	0.70	174

Table 30: test

iii. 3 layers

Class	Precision	Recall	F1Score	Support
1	0.98	0.93	0.95	2084
2	0.84	0.84	0.84	1974
3	0.71	0.71	0.71	1974
4	0.58	0.63	0.60	1865
5	0.79	0.79	0.79	2103

Table 31: train

Class	Precision	Recall	F1Score	Support
1	0.98	0.96	0.97	234
2	0.82	0.82	0.82	198
3	0.67	0.69	0.68	194
4	0.62	0.58	0.60	200
5	0.71	0.76	0.74	174

Table 32: test

iv. 4 layers

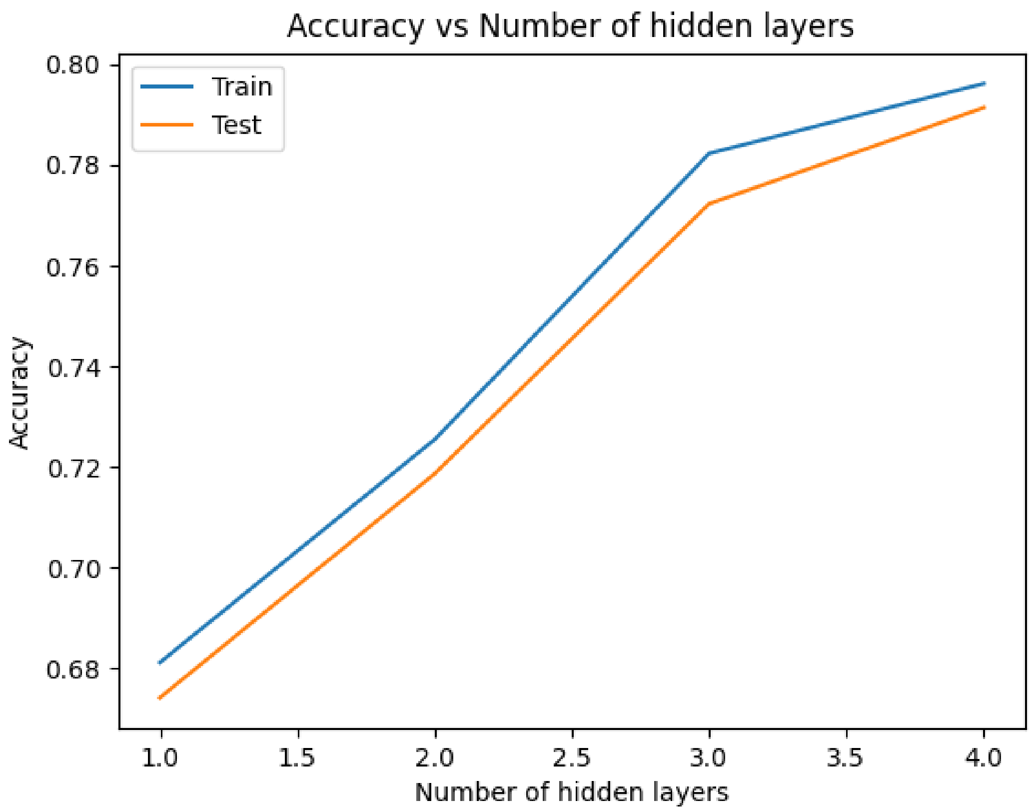
Class	Precision	Recall	F1Score	Support
1	0.92	0.99	0.95	1824
2	0.84	0.86	0.85	1938
3	0.73	0.72	0.73	1981
4	0.59	0.64	0.61	1865
5	0.86	0.75	0.80	2392

Table 33: train

Class	Precision	Recall	F1Score	Support
1	0.93	0.99	0.96	216
2	0.80	0.83	0.81	191
3	0.72	0.72	0.72	198
4	0.64	0.62	0.63	193
5	0.81	0.75	0.78	202

Table 34: test

We obtain the following graph:



We find that with relu and a epoch limit of 600, we are able to get higher accuracy rates but training time increases due to the higher number of epochs.

(f) MLPClassifier The following tables were obtained for various number of layers:

i. 1 layer

Class	Precision	Recall	F1Score	Support
1.0	0.70	0.91	0.79	1971
2.0	0.52	0.44	0.48	1978
3.0	0.47	0.34	0.40	1952
4.0	0.44	0.33	0.38	2008
5.0	0.58	0.79	0.67	2091

Table 35: train

Class	Precision	Recall	F1Score	Support
1	0.91	0.90	0.74	230
2	0.67	0.70	0.69	190
3	0.55	0.58	0.57	190
4	0.57	0.49	0.53	218
5	0.66	0.72	0.69	172

Table 36: test

ii. 2 layers

Class	Precision	Recall	F1Score	Support
1.0	0.76	0.90	0.83	1971
2.0	0.58	0.53	0.55	1978
3.0	0.46	0.41	0.43	1952
4.0	0.46	0.36	0.40	2008
5.0	0.63	0.78	0.70	2091

Table 37: train

Class	Precision	Recall	F1Score	Support
1	0.93	0.90	0.71	236
2	0.71	0.68	0.70	206
3	0.52	0.58	0.55	178
4	0.56	0.49	0.52	211
5	0.65	0.72	0.68	169

Table 38: test

iii. 3 layers

Class	Precision	Recall	F1Score	Support
1.0	0.80	0.89	0.84	1971
2.0	0.61	0.58	0.59	1978
3.0	0.50	0.44	0.47	1952
4.0	0.46	0.42	0.44	2008
5.0	0.65	0.74	0.69	2091

Table 39: train

Class	Precision	Recall	F1Score	Support
1	0.90	0.92	0.91	223
2	0.70	0.71	0.70	197
3	0.54	0.61	0.57	175
4	0.47	0.50	0.48	176
5	0.80	0.66	0.72	229

Table 40: test

iv. 4 layers

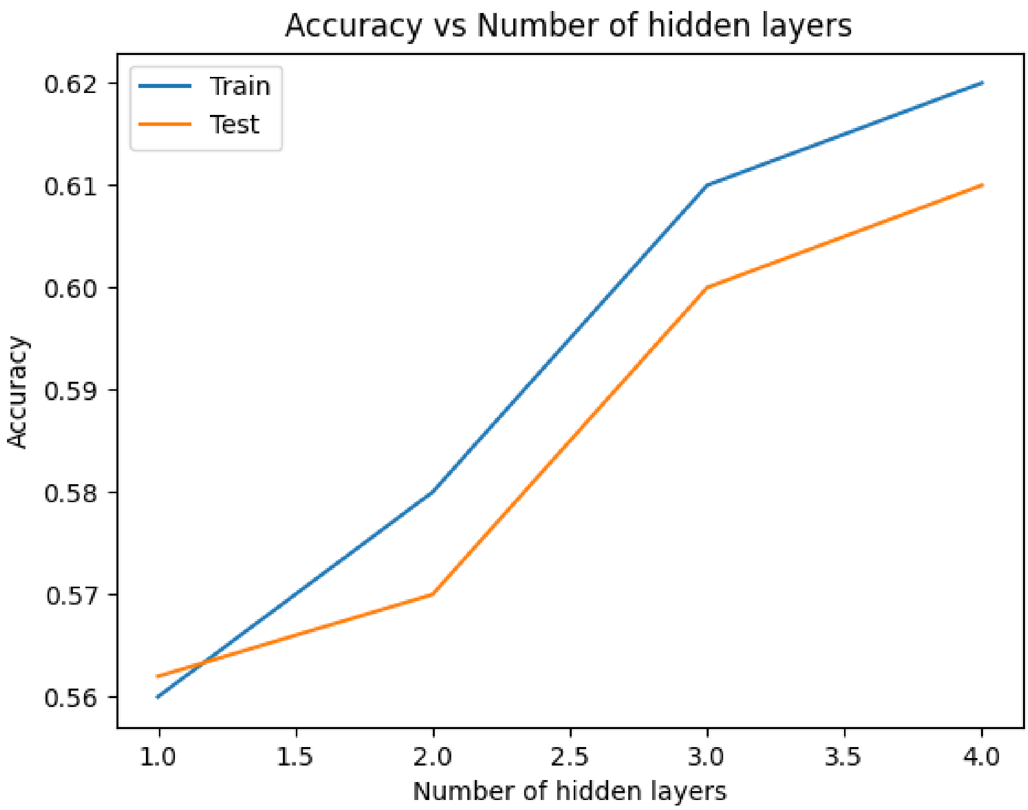
Class	Precision	Recall	F1Score	Support
1.0	0.82	0.88	0.85	1971
2.0	0.62	0.62	0.62	1978
3.0	0.50	0.45	0.47	1952
4.0	0.46	0.43	0.45	2008
5.0	0.66	0.73	0.69	2091

Table 41: train

Class	Precision	Recall	F1Score	Support
1	0.86	0.93	0.90	212
2	0.66	0.70	0.68	187
3	0.51	0.59	0.55	174
4	0.51	0.49	0.50	195
5	0.81	0.66	0.73	232

Table 42: test

We obtain the following graph:



We find that although sklearn is slightly faster than the self implementation in part b,c,d. The accuracy is slightly worse due to smaller number of epochs