ASSIGNMENT – 3

1. DECISION TREES, RANDOM FORESTS AND GRADIENT BOOSTED TREES:

- **Dataset – 1**

(a) Accuracy on Training data: 92.32456 %
Accuracy on Testing data: 69.16996 %
Accuracy on Validation data: 76.03305 %

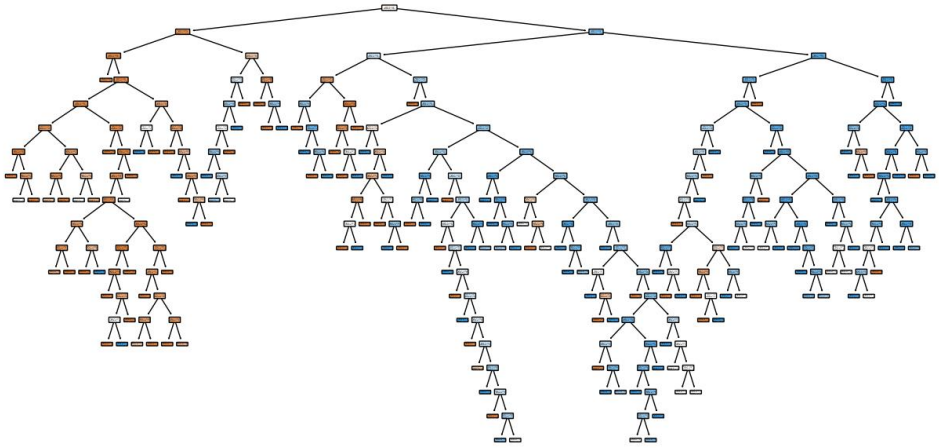


Figure 1: Decision Tree: (a)

(b) Accuracy on Training data: 83.11403 %
Accuracy on Testing data: 76.28458 %
Accuracy on Validation data: 86.77685 %

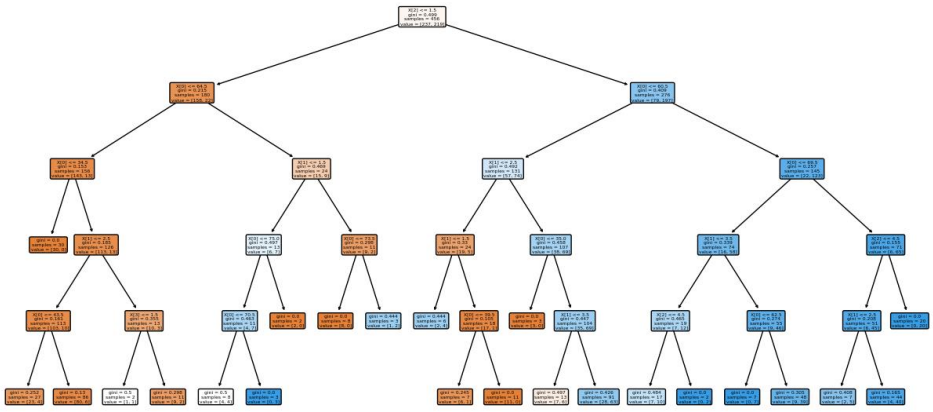


Figure 2: Decision Tree: (b)

On performing Grid Search over the mentioned parameters, the following tree was found to be the best:

DecisionTreeClassifier(max_depth=5, min_samples_leaf=2, min_samples_split=8)

The tree returned by Grid Search performs better on Testing and Validation Sets increasing the accuracies of the respective sets by 7% and 10%.

When compared to the tree visualised in part (a), this tree has stopped at a depth of 5 as minimum samples required to further split came out to be 8.

(c) Training Accuracy: 80.92105 %
Testing Accuracy: 75.88933 %
Validation Accuracy: 89.25620 %

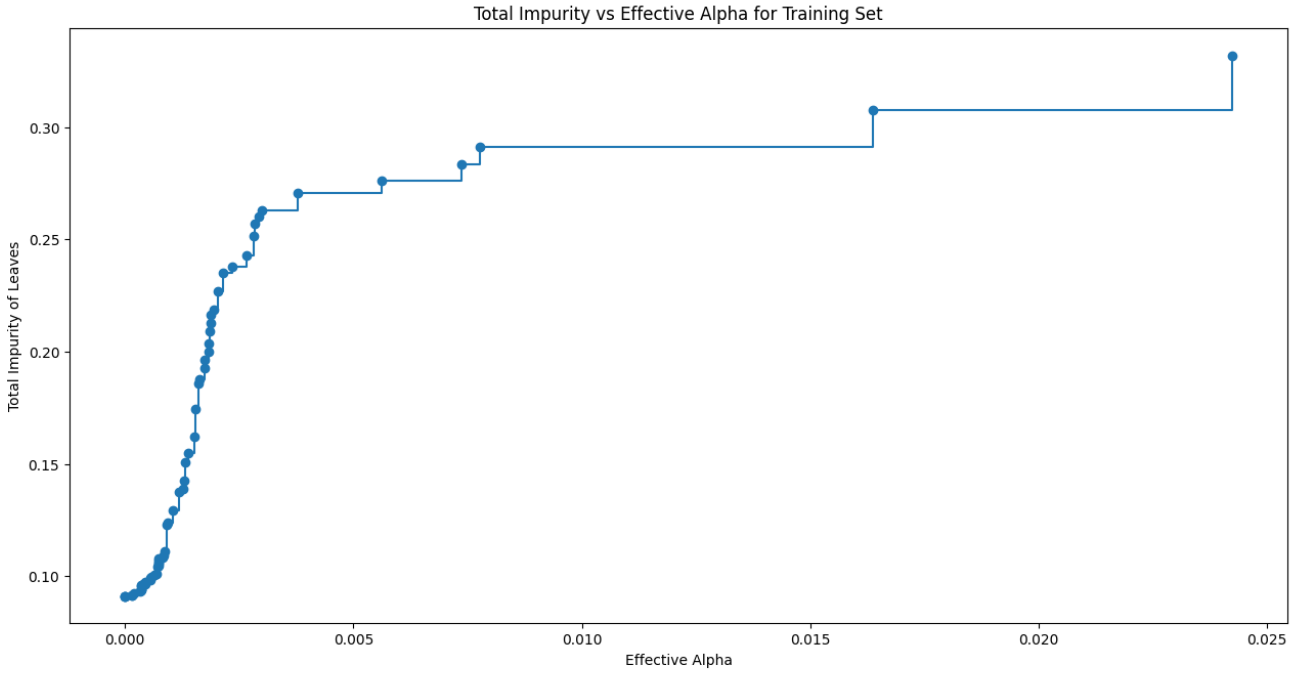


Figure 3: (c) Total Impurity vs Effective Alpha for Training Set

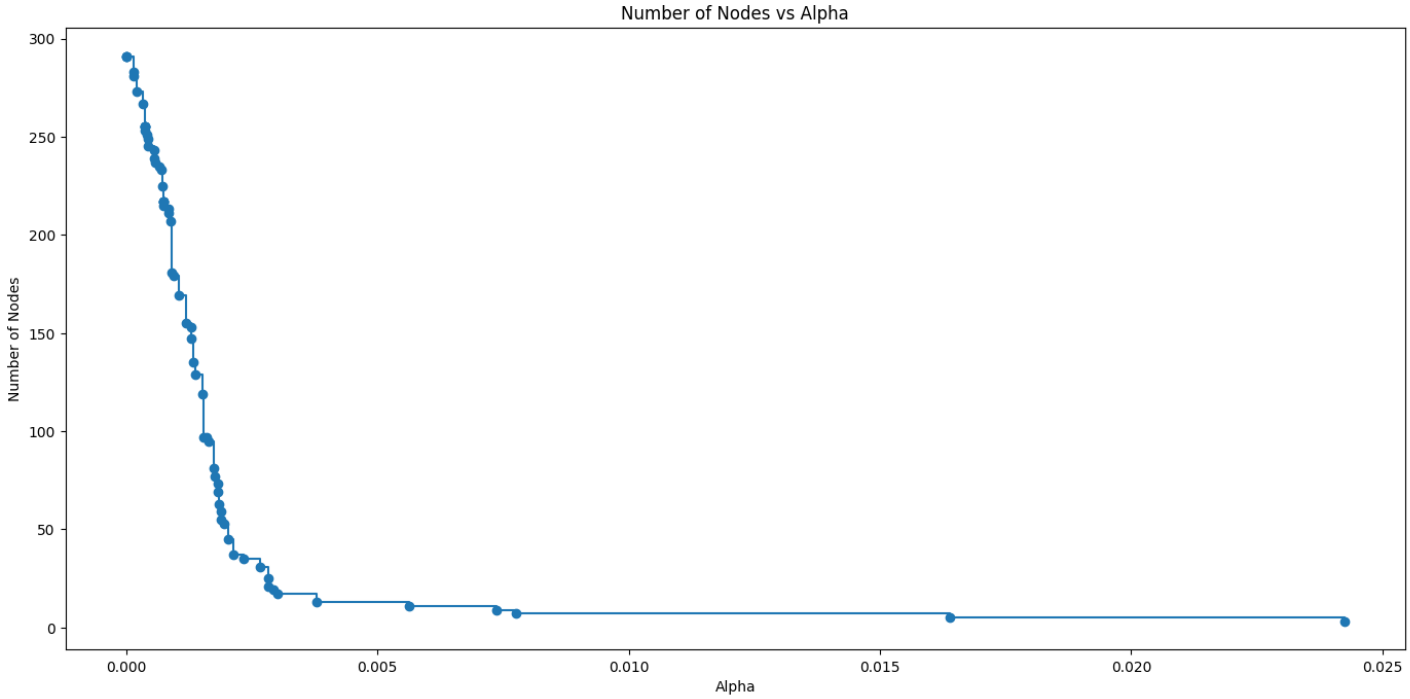


Figure 4: (c) Number of Nodes vs Alpha

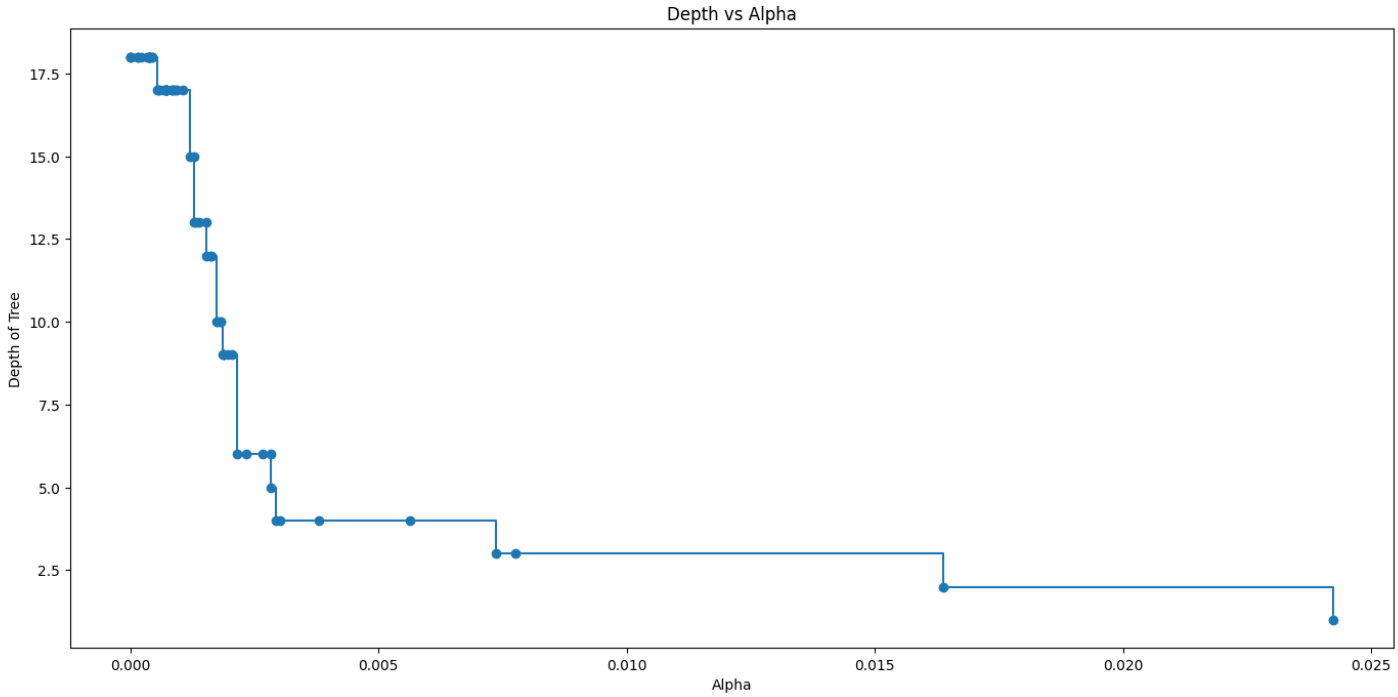


Figure 5: (c) Depth vs Alpha

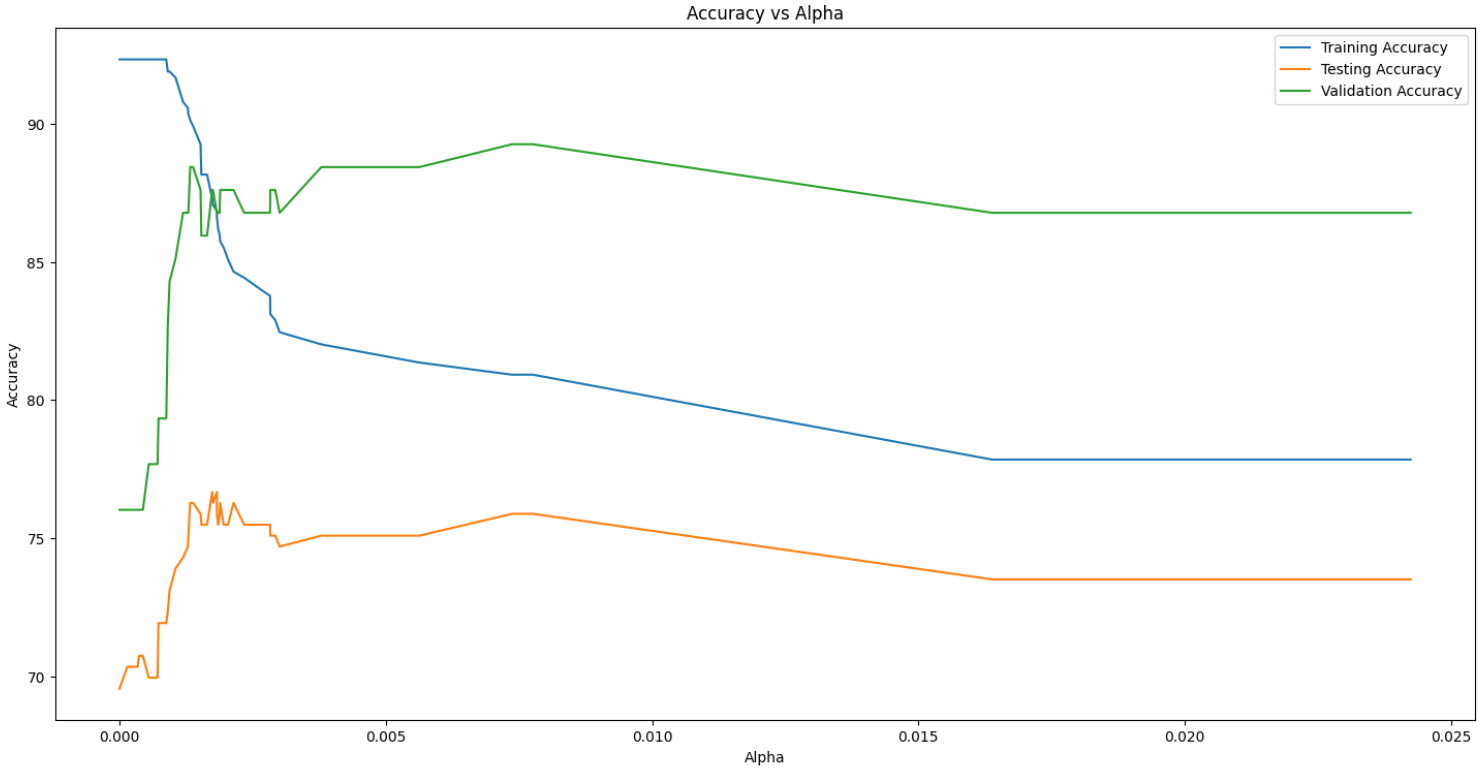


Figure 6: (c) Accuracy vs Alpha

From the above figure, it can be observed that even though training accuracy decreases, the model is now more generalised and will perform better on unseen data.

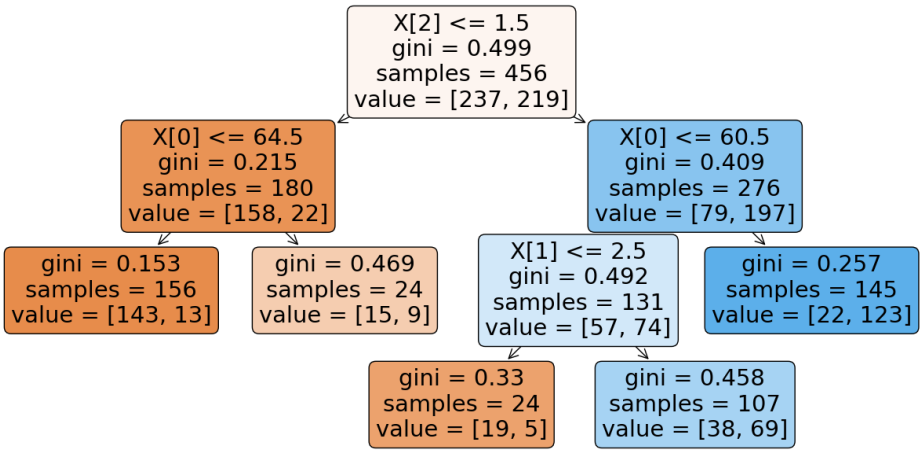


Figure 7: (c) Best Tree after Cost Complexity Pruning

(d) Accuracy on Training dataset: 92.32456 %
Accuracy on Testing dataset: 74.70356 %
Accuracy on Validation dataset: 80.99173 %
Out-Of-Bag Accuracy: 73.68421 %

(e) Dataset-1
(a) MEDIAN MODEL
Accuracy on Training dataset: 91.80633 %
Accuracy on Testing dataset: 72.91667 %
Accuracy on Validation dataset: 73.33333 %

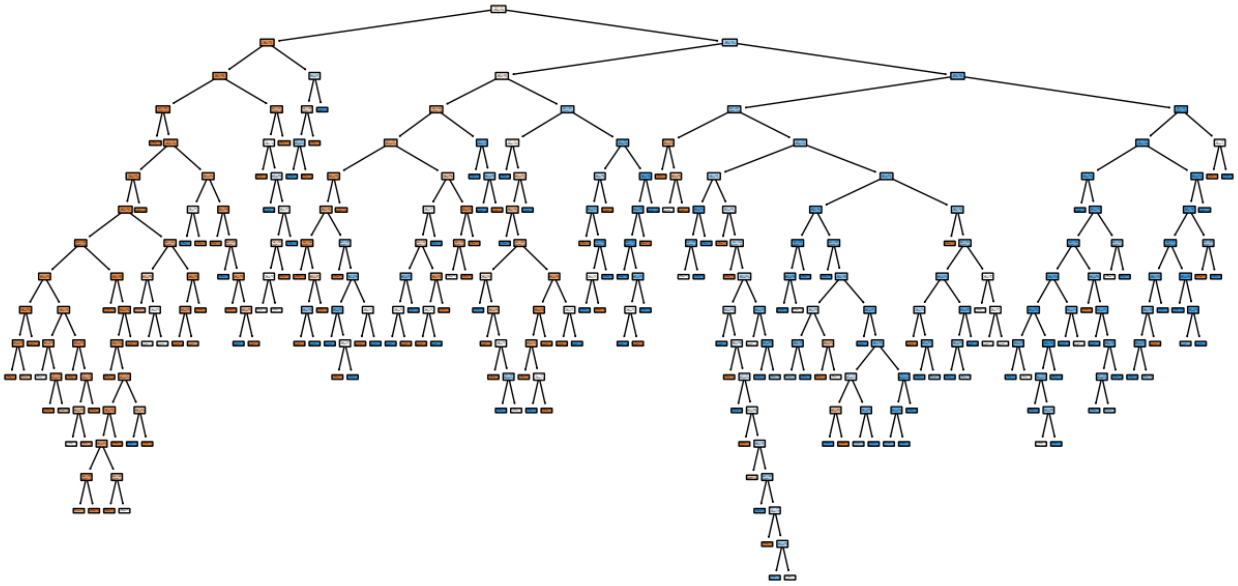


Figure 8: (e) Decision Tree (a) - Median Model

(a) MODE MODEL
Accuracy on Training dataset: 90.68901 %
Accuracy on Testing dataset: 70.83334 %
Accuracy on Validation dataset: 76.29630 %

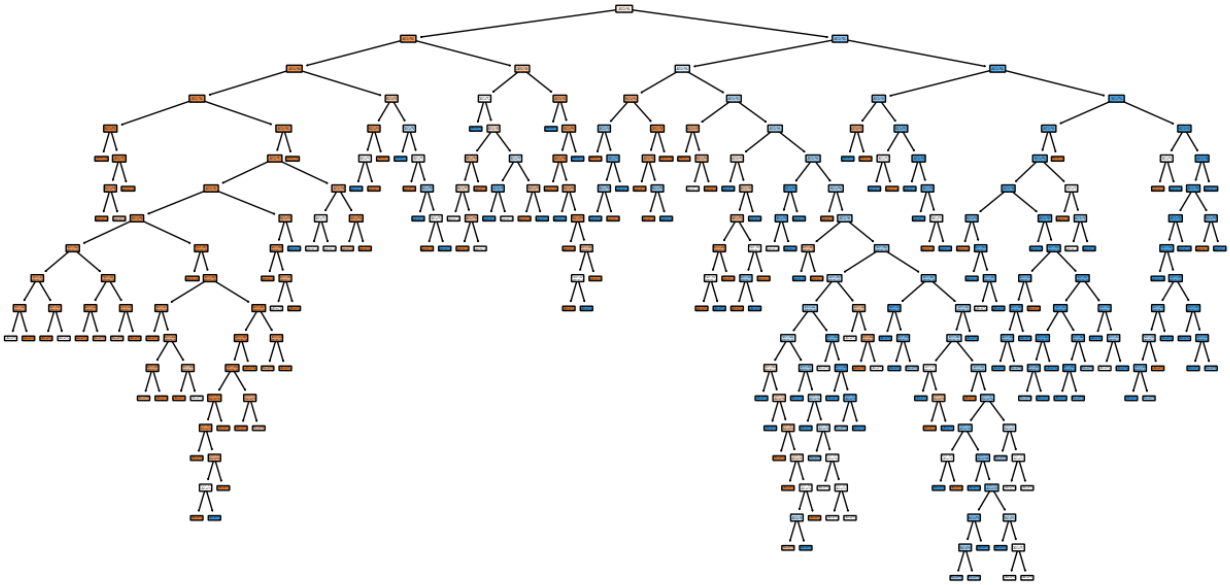


Figure 9: (e) Decision Tree (a) - Mode Model

(b) MEDIAN MODEL
Accuracy on Training dataset: 82.30912 %
Accuracy on Testing dataset: 80.20833 %
Accuracy on Validation dataset: 85.92593 %

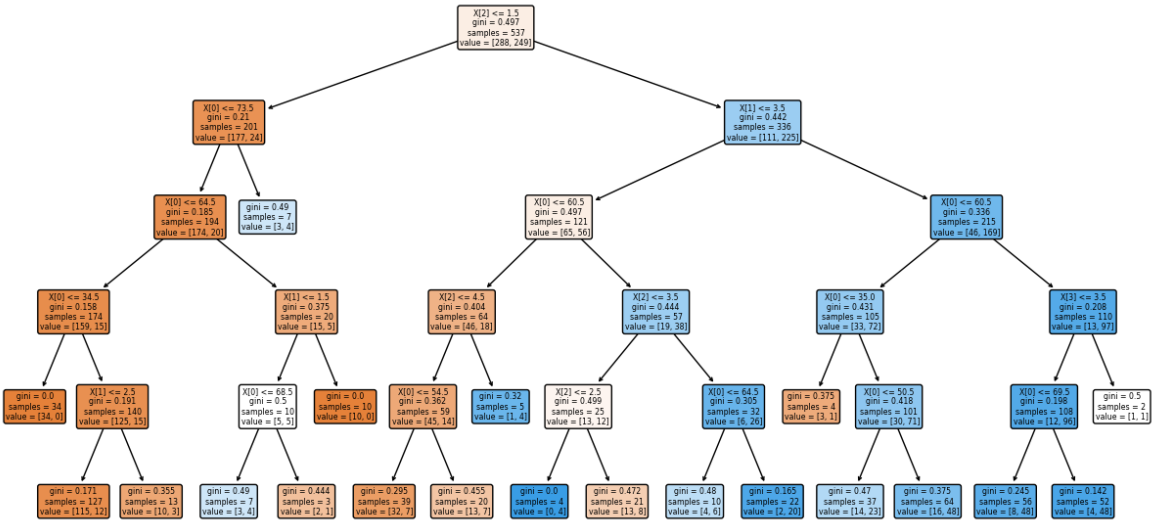


Figure 10: (e) Decision Tree (b) - Median Model

(b) MODE MODEL
Accuracy on Training dataset: 81.75046 %
Accuracy on Testing dataset: 76.04167 %
Accuracy on Validation dataset: 85.92593 %

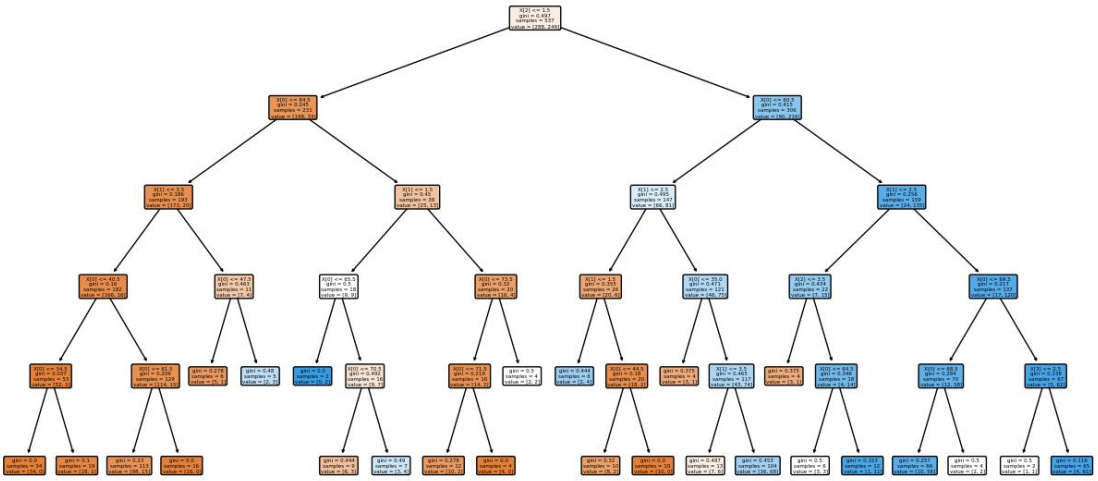


Figure 11: (e) Decision Tree (b) - Mode Model

(c) MEDIAN MODEL
Training Accuracy: 80.26071 %
Testing Accuracy: 79.16667 %
Validation Accuracy: 87.40741 %

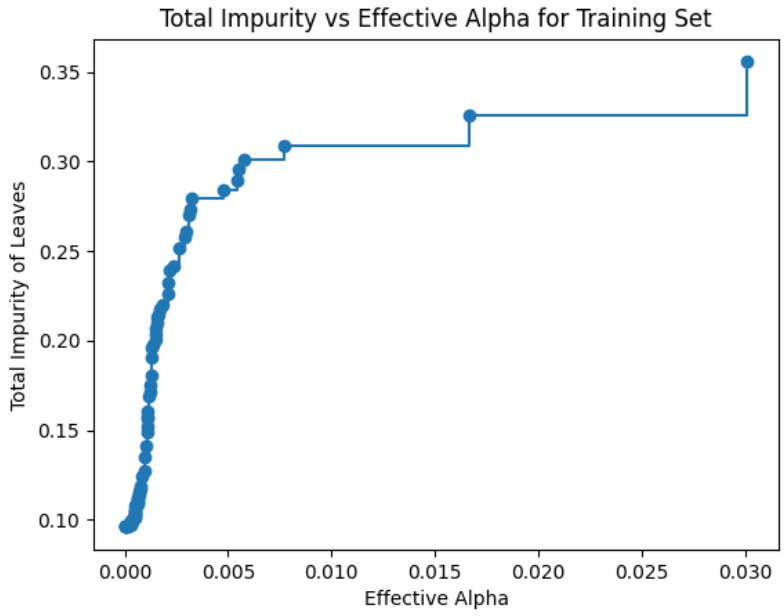


Figure 12: (e) Total Impurity vs Alpha - Median Model

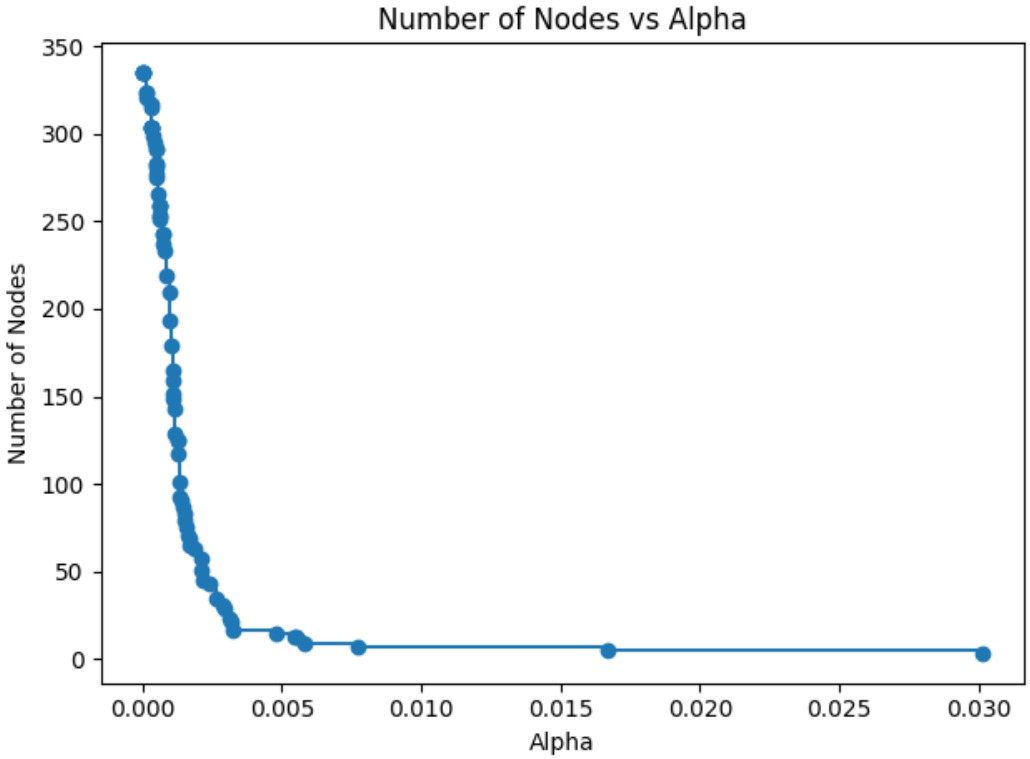


Figure 13: (e) Number of Nodes vs Alpha - Median Model

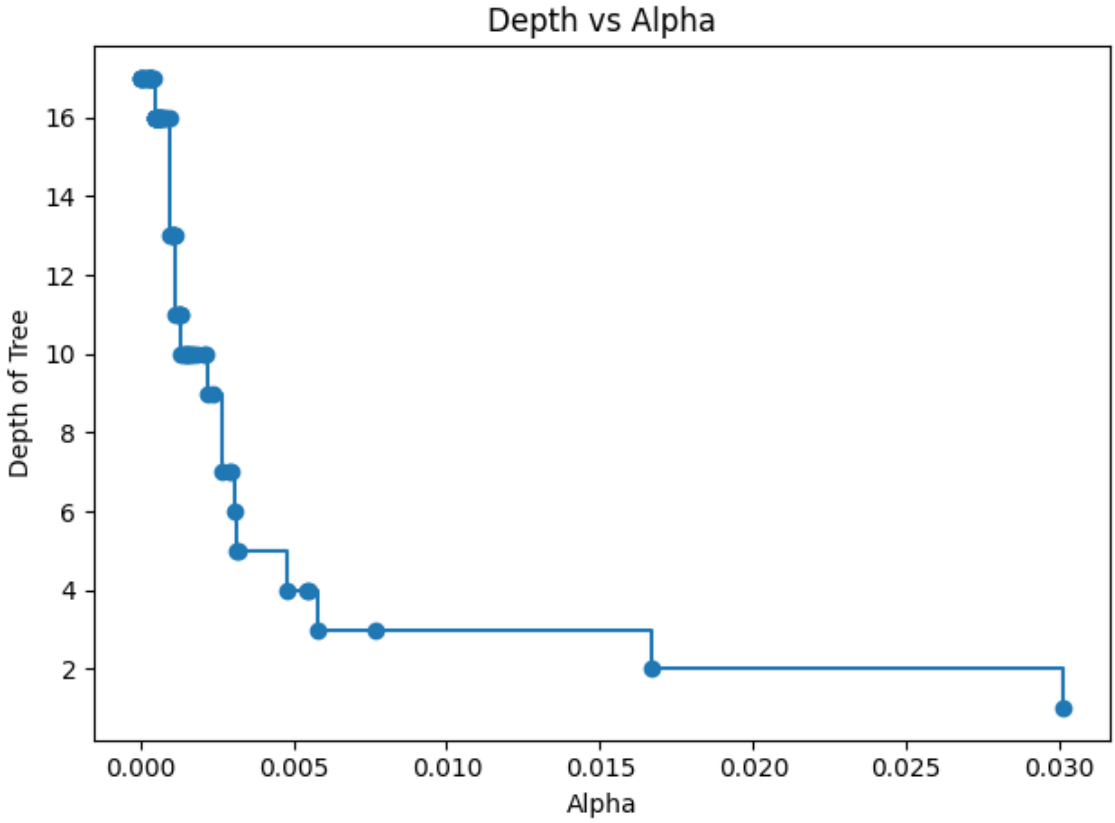


Figure 14: (e) Depth vs Alpha - Median Model

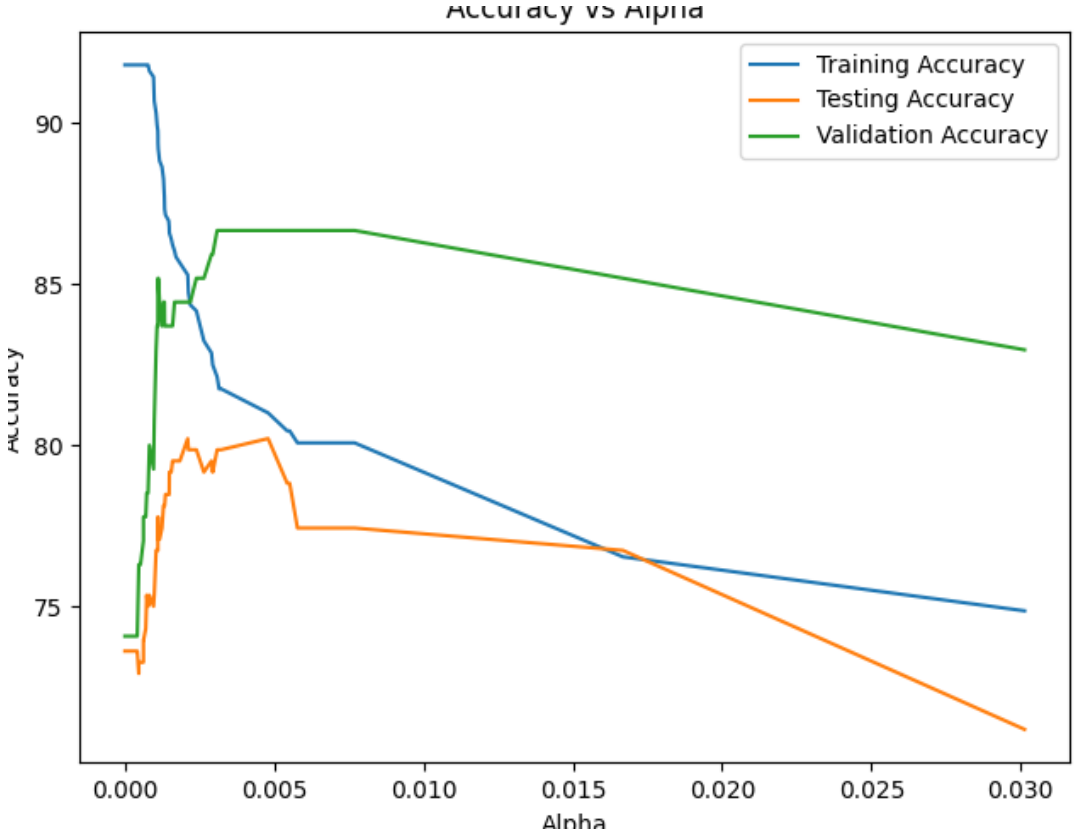


Figure 15: (e) Accuracy vs Alpha - Median Model

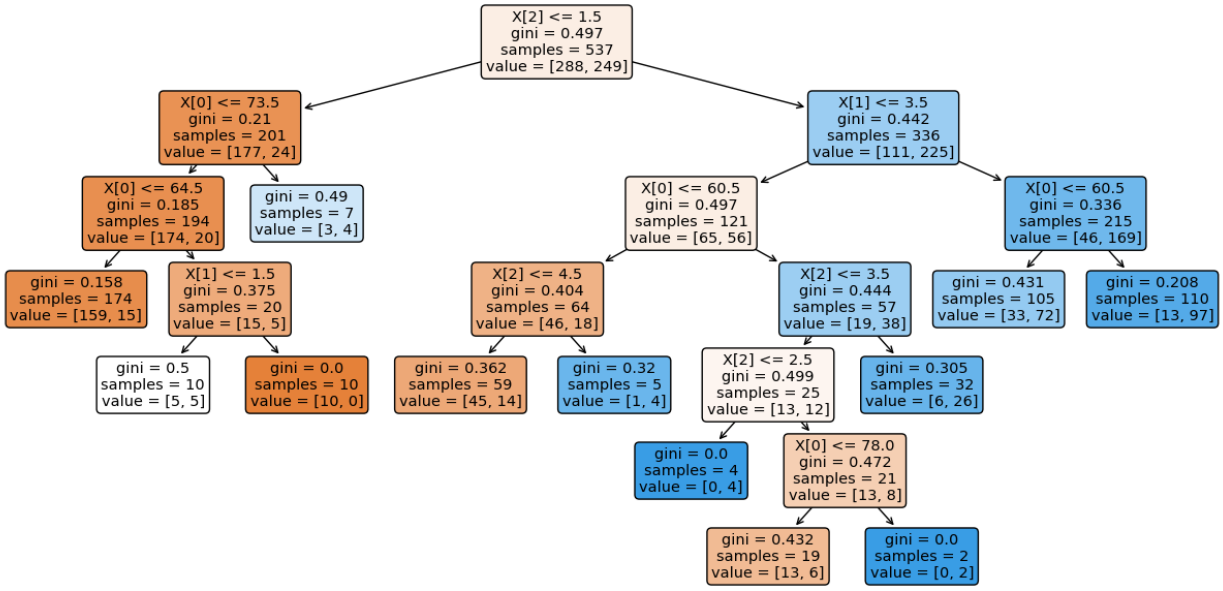


Figure 16: (e) Best Tree - Median Model

(c) MODE MODEL
Training Accuracy: 89.01304 %
Testing Accuracy: 75.34722 %
Validation Accuracy: 88.14815 %

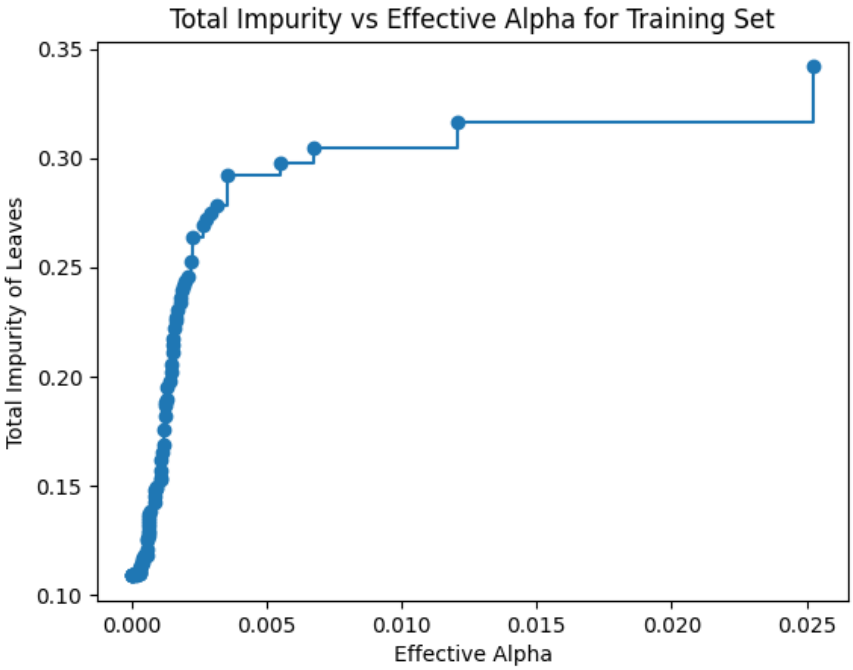


Figure 17: (e) Total Impurity vs Alpha - Mode Model

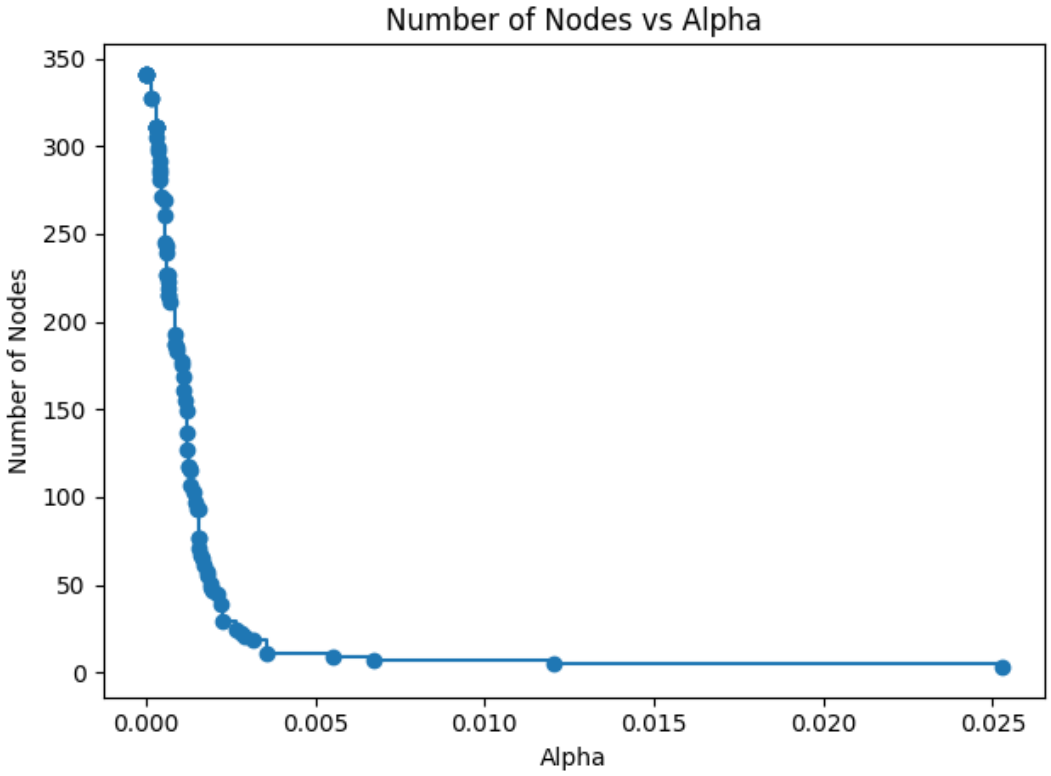


Figure 18: (e) Number of Nodes vs Alpha - Mode Model

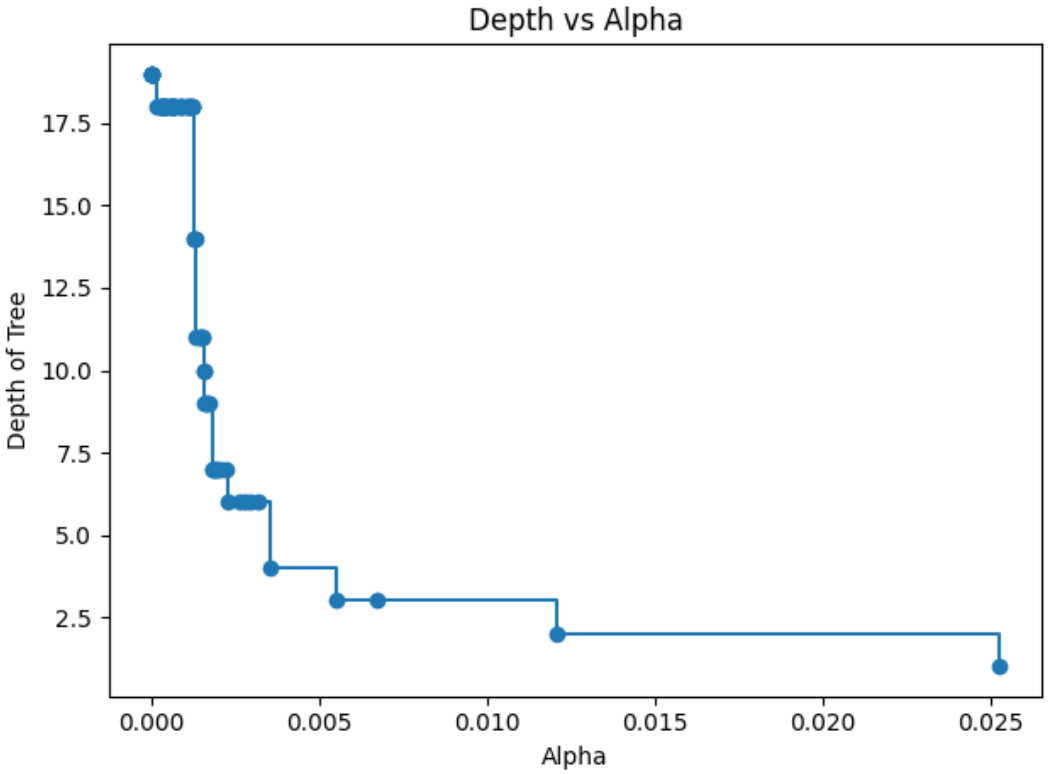


Figure 19: (e) Depth vs Alpha - Mode Model

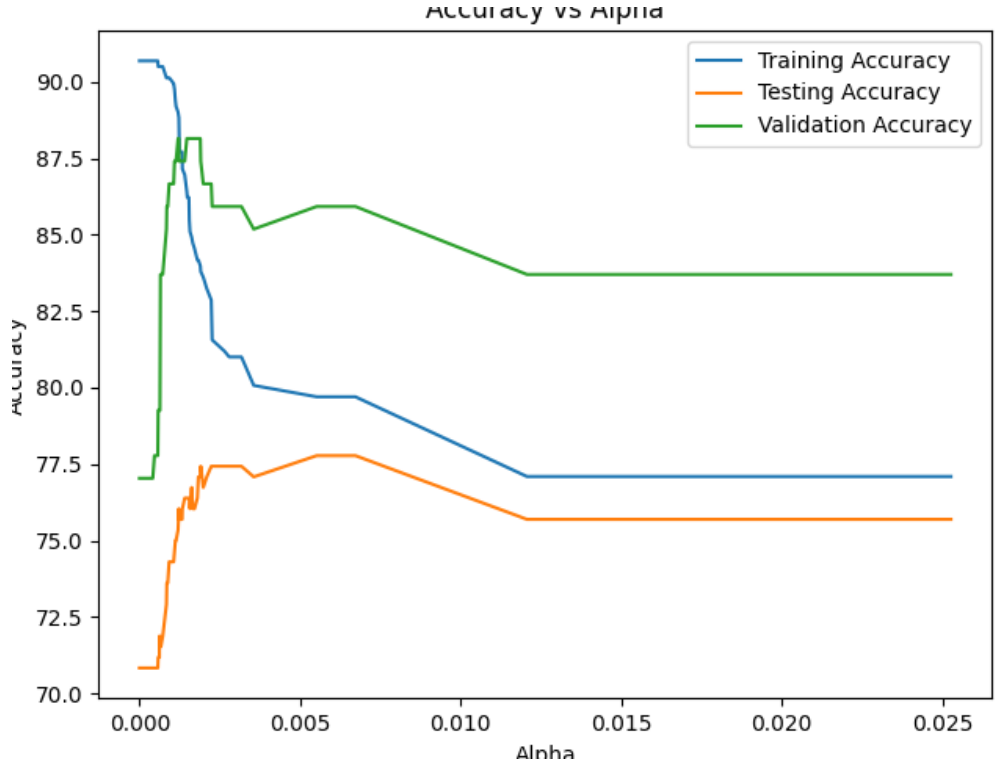


Figure 20: (e) Accuracy vs Alpha - Mode Model

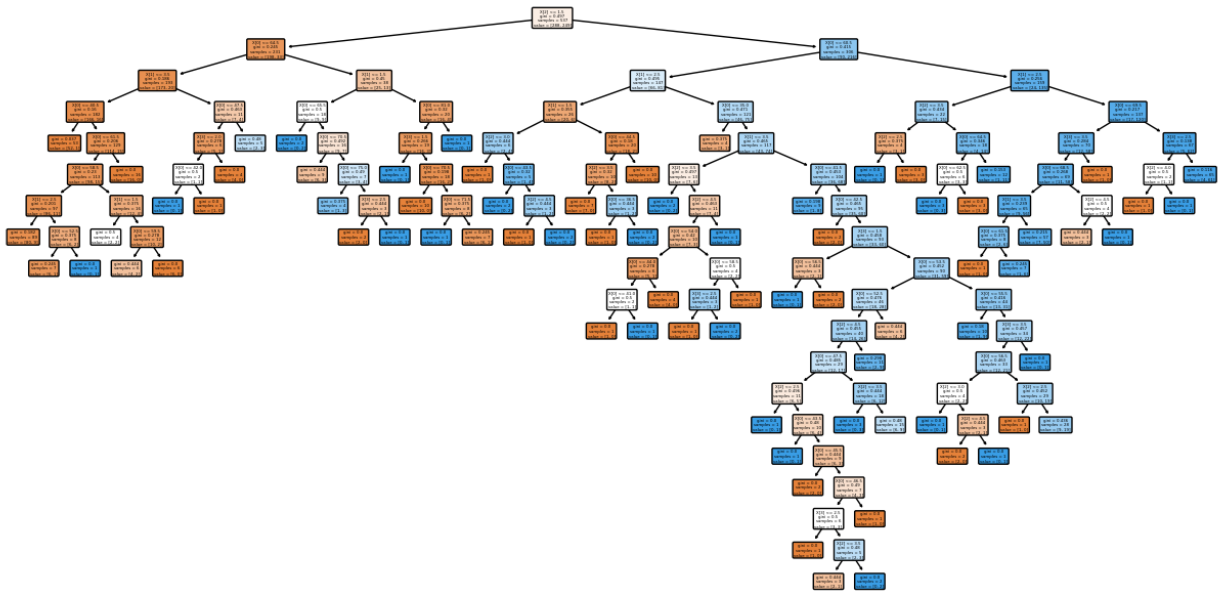


Figure 21: (e) Best Decision Tree (c) - Mode Model

(d) MEDIAN MODEL
Accuracy on Training dataset: 91.80633 %
Accuracy on Testing dataset: 75.69444 %
Accuracy on Validation dataset: 82.96296 %
Out-Of-Bag Accuracy: 72.81192 %
(d) MODE MODEL
Accuracy on Training dataset: 90.68901 %
Accuracy on Testing dataset: 74.65277 %
Accuracy on Validation dataset: 82.22222 %
Out-Of-Bag Accuracy: 72.06704 %

(f) Accuracy on Training dataset: 83.61266 %
Accuracy on Testing dataset: 77.08333 %
Accuracy on Validation dataset: 84.44445 %

- **Dataset – 2**

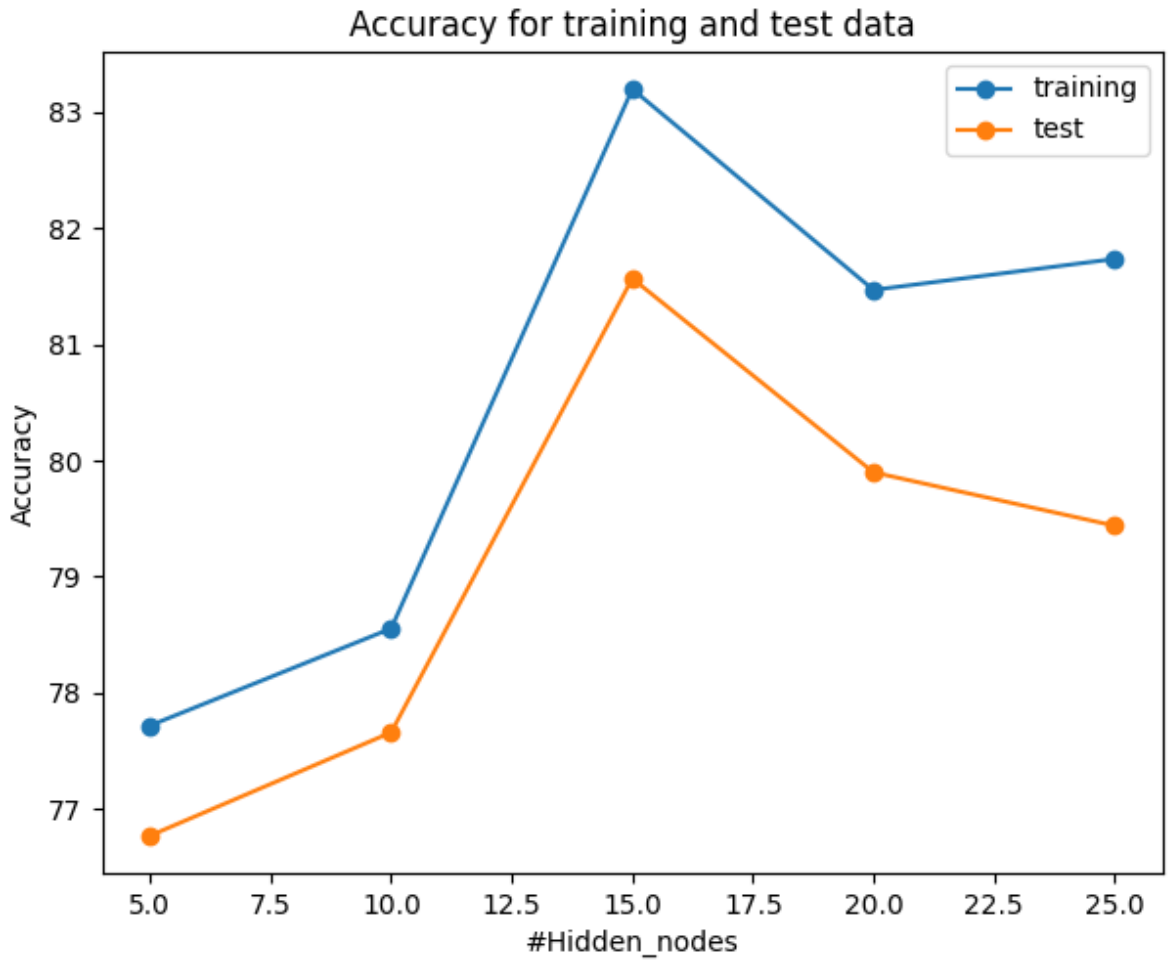
(a) Accuracy on Training dataset: 99.96900 %
Accuracy on Testing dataset: 57.61820 %
Accuracy on Validation dataset: 57.82719 %
(b) Accuracy on Training dataset: 96.68845 %
Accuracy on Testing dataset: 56.55060 %
Accuracy on Validation dataset: 56.55211 %

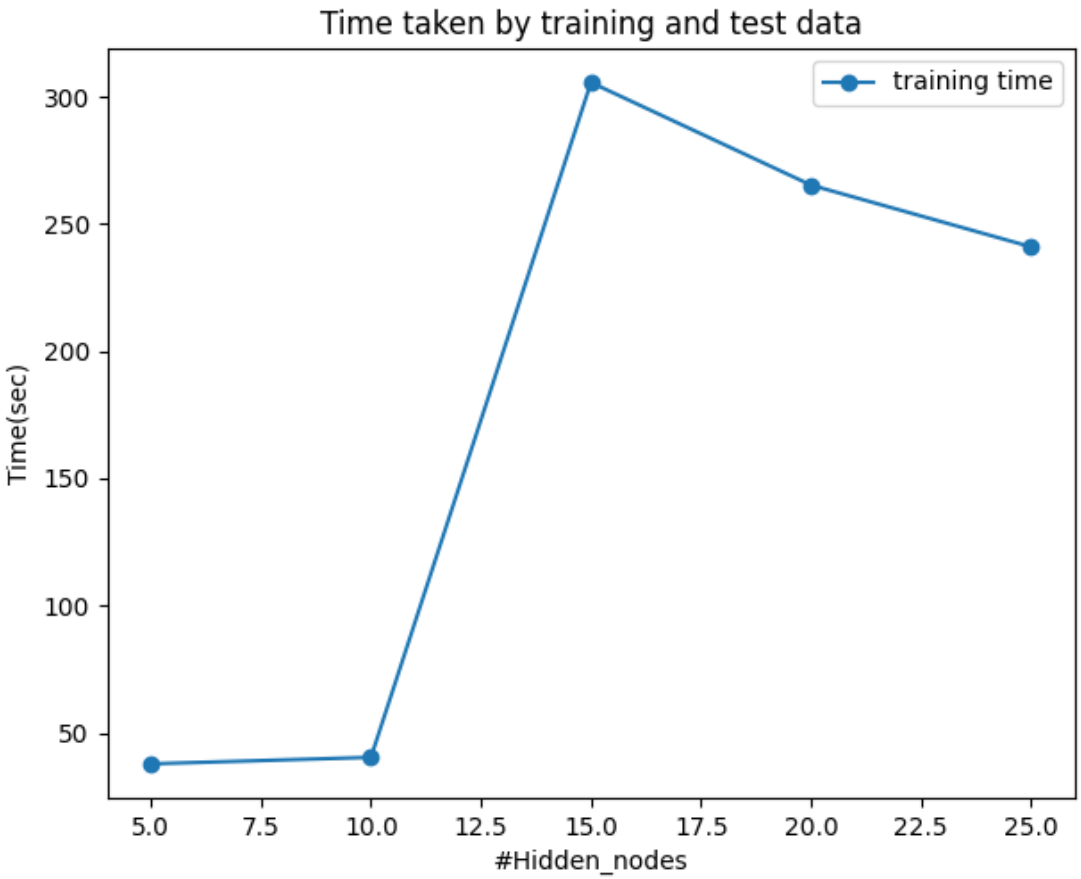
2) Neural Networks

a) Implemented the neural network interface as mentioned in the question
nn(feature_layer=<integer>, hidden_layer=<list>, costFunction=['bce','sgd'], output_layer=<integer>, batchSize=<integer>, activationFunctions=dict{layer_number:['sigmoid','relu']})

Convergence criteria: we set aside a fixed validation set of 5000 instance and after each epoch we calculate the network's performance, when there is no significant change in the accuracy of the subsequent epoch we stop learning

b) Hidden layer nodes: [5,10,15,20,25]




Time taken by training and test data
training time
300
250
200
150
100
50
Time(sec)
5.0
7.5
10.0
12.5
15.0
17.5
20.0
22.5
25.0
#Hidden_nodes

model 5 hidden layers

Time taken to train: 37.8497838973999

Training Accuracy: 77.71629527158785%

Test Accuracy: 76.76767676767676%

----------confusion matrix----------

[[791 14 26 105 12   3 201   0   2   0]
 [2 904   0   8   0   0   0   0   1   0]
 [10 12 503   5 70   0 106   0   9   0]
 [37 50 12 741 22   0 42   0   2   0]
 [33 15 276 104 724   0 119   0   8   2]
 [2   0   1   1   0 887   0 140   6 43]
 [92   4 168 28 160   0 488   0 11   0]
 [0   0   0   0   0 57   0 835   6 96]
 [32   1 13   8 11 13 44   1 946   1]
 [1   0   1   0   1 40   0 24   9 857]]

---

model 10 hidden layers

Time taken to train: 40.40649962425232

Training Accuracy: 78.55297588293138%

Test Accuracy: 77.65776577657766%

----------confusion matrix----------

[[838 12 34 65 7 0 259 0 12 0]

[5 941 1 12 5 1 2 0 0 0]

[17 10 581 7 66 0 121 0 19 1]

[82 25 19 837 47 0 60 0 3 0]

[11 8 309 38 824 0 395 0 7 0]

[2 0 1 1 1 839 1 25 12 18]

[28 3 44 33 44 0 130 0 2 0]

[0 0 0 0 0 92 0 942 7 79]

[17 1 11 7 6 14 32 0 935 3]

[0 0 0 0 0 54 0 33 3 898]]

---

model 15 hidden layers

Time taken to train: 305.5929822921753

Training Accuracy: 83.19971999533325%

Test Accuracy: 81.56815681568156%

----------confusion matrix----------

[[719 3 6 34 1 0 127 0 2 0]

[6 927 0 9 1 1 1 0 0 0]

[9 15 627 11 72 0 76 0 11 0]

[84 41 15 850 24 2 62 0 9 1]

[9 8 197 52 771 0 129 0 8 0]

[2 0 0 0 0 906 1 48 10 13]

[159 5 148 39 123 0 582 0 26 1]

[0 0 0 0 0 52 1 915 4 53]

[12 1 7 3 8 4 21 0 929 1]

[0 0 0 2 0 35 0 37 1 930]]

---

model 20 hidden layers

Time taken to train: 265.26688027381897

Training Accuracy: 81.46969116151936%

Test Accuracy: 79.89798979897989%

----------confusion matrix----------

[[810 5 45 51 9 0 233 0 12 0]

[1 936 5 17 0 0 5 0 1 0]

[21 16 784 16 229 0 197 0 13 0]

[57 28 11 796 37 1 45 0 3 0]

[19 9 118 49 653 0 107 0 5 0]

[2 0 0 2 0 894 1 69 6 30]

[73 6 25 64 64 0 384 0 4 0]

[0 0 1 0 0 44 0 864 4 43]

[17 0 8 3 8 17 25 1 943 1]

[0 0 3 2 0 44 3 66 9 925]]

---

model 25 hidden layers

Time taken to train: 241.0044286251068

Training Accuracy: 81.73636227270454%

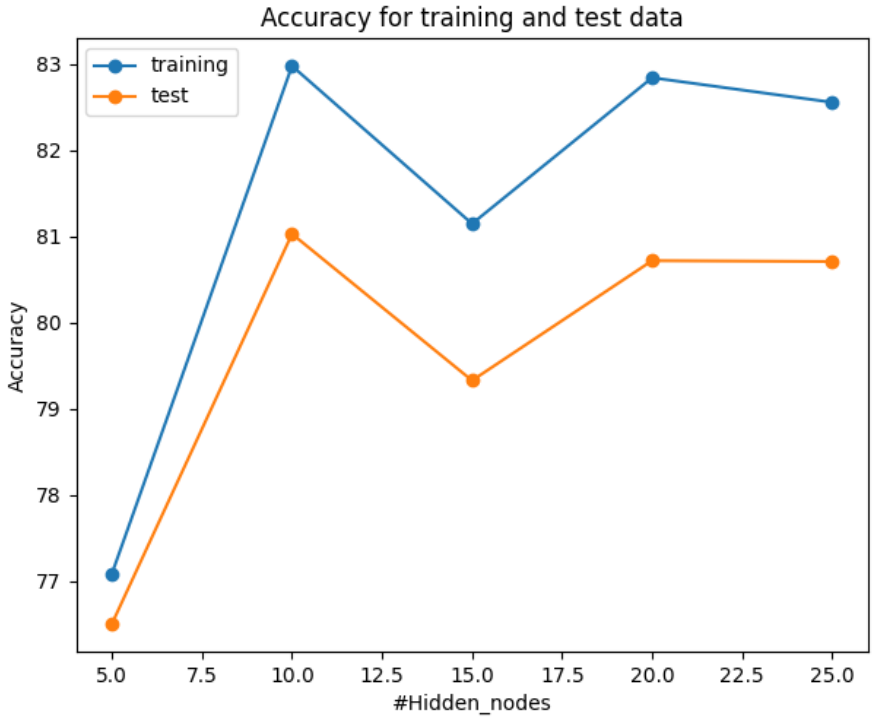
Test Accuracy: 79.43794379437944%

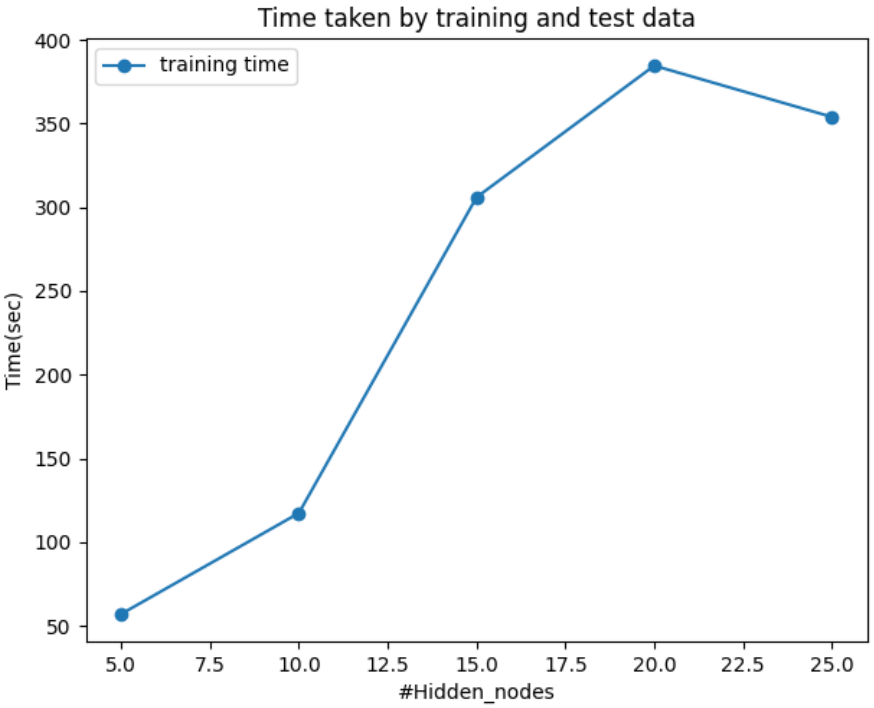
----------confusion matrix----------

```
[[756   4  12  45  10   0 200   0   4   0]
 [  3 933   2  19   5   1   5   0   1   0]
 [ 78  15 842  12 277   0 246   0  56   1]
 [ 57  32  12 818  39   0  39   0   5   0]
 [ 19  13  82  43 585   0 105   0   6   0]
 [  2   0   2   4   0 899   1  58   9  24]
 [ 73   3  42  56  75   0 393   0  15   0]
 [  0   0   0   0   0  44   0 886   5  35]
 [ 11   0   5   2   9   9  11   0 892   0]
 [  1   0   1   1   0  47   0  56   7 939]]
```

Observations: Training time is increasing as we expected with the increase in hidden nodes as the network becomes more complex, accuracy increased with increased in number of hidden layers and get saturated around 80% after that further increase in hidden nodes did not increase the accuracy.

c) Adaptive learning





model 5 hidden layers

Time taken to train: 57.03181791305542

Training Accuracy: 77.08961816030268%

Test Accuracy: 76.5076507650765%

----------confusion matrix----------

[[857 5 48 91 13 0 335 0 8 0]

[4 926 7 8 3 1 2 0 2 0]

[18 16 691 25 136 0 227 0 7 0]

[45 40 10 823 45 0 29 0 7 1]

[14 6 214 10 692 0 271 0 14 0]

[2 0 0 1 0 863 1 78 12 33]

[39 5 13 38 104 0 94 0 5 0]

[0 0 0 1 0 79 0 895 7 91]

[21 2 17 3 7 16 41 0 937 2]

[0 0 0 0 0 41 0 27 1 872]]

---

model 10 hidden layers

Time taken to train: 117.32068991661072

Training Accuracy: 82.97638293971566%

Test Accuracy: 81.02810281028103%

----------confusion matrix----------

[[803 4 45 47 4 0 237 0 14 0]

[2 929 7 16 4 0 2 0 0 0]

[31 18 719 16 122 0 140 0 12 0]

[78 34 14 854 28 1 61 1 9 0]

[8 4 138 37 752 0 128 0 4 0]

[3 0 2 1 1 911 0 67 10 26]

[51 7 58 24 83 1 392 0 12 0]

[0 0 0 0 0 39 0 871 7 28]

[24 2 17 5 6 12 40 2 932 6]

[0 2 0 0 0 36 0 59 0 939]]

---

model 15 hidden layers

Time taken to train: 305.9366545677185

Training Accuracy: 81.14635243920732%

Test Accuracy: 79.32793279327933%

----------confusion matrix----------

[[764 2 19 30 1 1 172 0 3 0]

[2 934 4 18 1 0 0 0 3 1]

[28 10 682 17 156 0 122 0 15 0]

[81 38 16 862 69   1 65   0   7   0]

[15 11 103 33 659   1 147   0 25   0]

[2   1   0   5   1 872   1 49 10 37]

[94   0 172 27 95   2 476   0 20   2]

[1   1   0   3   3 67   0 853   3 40]

[12   3   4   5 15   8 16   3 912   1]

[1   0   0   0   0 48   1 95   2 918]]

---

model 20 hidden layers

Time taken to train: 384.5056827068329

Training Accuracy: 82.83971399523325%

Test Accuracy: 80.71807180718072%

----------confusion matrix----------

[[765 10 12 60 21   0 193   0   2   0]

[5 932   3 14   1   0   0   0   1   0]

[15 10 675 10 116   0 99   0   8   1]

[38 29   9 796 28   1 30   0   3   0]

[15 11 144 47 725   0 141   0   7   0]

[3   0   1   4   0 890   2 28 16 29]

[141   6 145 65 100   0 511   0 26   0]

[0   0   1   0   0 62   0 923   3 47]

[18   2 10   2   9   7 24   0 933   1]

[0   0   0   2   0 40   0 49   1 921]]

---

model 25 hidden layers

Time taken to train: 353.88060450553894

Training Accuracy: 82.55470924515409%

Test Accuracy: 80.7080708070807%

----------confusion matrix----------

[[822 3 43 72 25 0 241 0 10 0]

[1 939 4 12 6 0 4 0 1 0]

[25 20 791 14 191 0 173 0 20 0]

[54 30 10 816 33 3 35 0 4 1]

[8 7 97 40 659 0 120 0 8 0]

[2 1 0 2 0 893 2 58 8 24]

[72 0 47 32 74 2 394 0 5 0]

[0 0 0 1 1 45 0 889 6 41]

[16 0 7 11 10 15 31 2 937 3]

[0 0 1 0 1 42 0 51 1 930]]

Observation: Results are almost similar when compared with above cases, learning time slightly increased but not too much.

d) Several activation units: Sigmoid and ReLU

model sigmoid [100, 100] hidden layers

Time taken to train: 2944.3284707069397

Training Accuracy: 77.20628677144619%

Test Accuracy: 76.58765876587658%

----------confusion matrix----------

[[811 4 36 39 1 0 263 0 5 0]

[7 931 5 12 8 1 3 0 1 0]

[12 14 678 7 197 0 223 0 9 0]

[111 42 12 870 60 1 73 0 17 0]

[12 7 209 30 682 0 240 0 2 0]

[10 0 6 4 5 836 17 64 16 28]

[14 0 34 34 37 0 138 0 1 0]

[0 0 0 0 0 103 1 852 8 48]

[23 2 20 4 10 8 42 1 938 1]

[0 0 0 0 0 51 0 83 3 922]]

ReLU model [100, 100] hidden layers

Time taken to train: 573.2644910812378

Training Accuracy: 86.76311271854532%

Test Accuracy: 84.85848584858486%

----------confusion matrix----------

[[815 4 15 29 0 0 139 0 0 0]

[4 955 3 19 1 1 5 0 1 0]

[10 5 762 13 103 0 103 0 7 0]

[48 27 10 861 34 1 42 0 5 0]

[4 6 125 37 763 0 79 0 6 0]

[6 0 1 1 0 912 0 36 7 21]

[101 2 78 35 92 0 610 0 17 0]

[0 0 0 0 0 55 1 922 4 44]

[12 1 6 4 7 5 21 1 952 1]

[0 0 0 1 0 26 0 41 1 933]]

Observation: ReLU activation function performs better accuracy increased by 10% when compared with sigmoid, when compared with 2b, accuracy increased by 5% however as expected training time also increased.

e) Varying the number of hidden layers

ReLU

model [50, 50] hidden layers

Time taken to train: 687.1628496646881

Training Accuracy: 83.64806080101334%

Test Accuracy: 82.34823482348234%

---

model [50, 50, 50] hidden layers

Time taken to train: 949.8115842342377

Training Accuracy: 84.51474191236521%

Test Accuracy: 83.1983198319832%

---

model [50, 50, 50, 50] hidden layers

Time taken to train: 807.3541448116302

Training Accuracy: 84.23973732895548%

Test Accuracy: 82.67826782678267%

---

model [50, 50, 50, 50, 50] hidden layers

Time taken to train: 971.501805305481

Training Accuracy: 84.59640994016567%

Test Accuracy: 83.1083108310831%

Sigmoid

model [50, 50] hidden layers

Time taken to train: 1145.8908560276031

Training Accuracy: 74.76291271521193%

Test Accuracy: 74.22742274227423%

---

model [50, 50, 50] hidden layers

Time taken to train: 2317.100964307785

Training Accuracy: 47.79412990216504%

Test Accuracy: 47.814781478147815%

---

model [50, 50, 50, 50] hidden layers

Time taken to train: 2650.133682012558

Training Accuracy: 27.13378556309272%

Test Accuracy: 27.322732273227324%

---

model [50, 50, 50, 50, 50] hidden layers

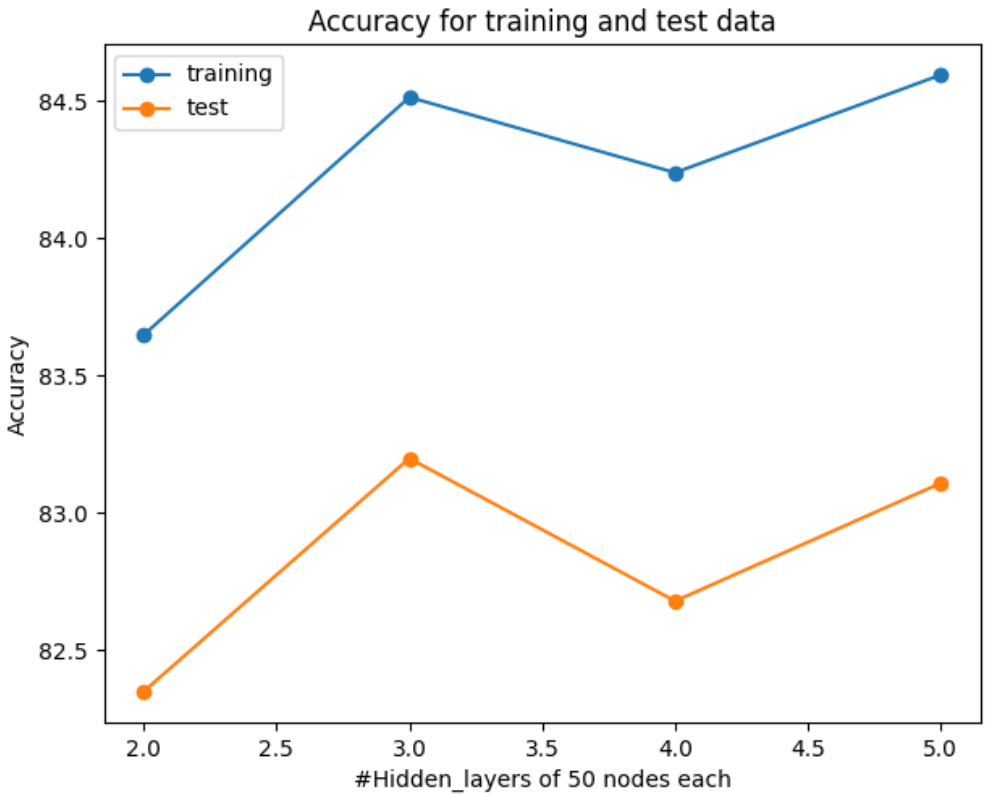
Time taken to train: 2852.7096073627472

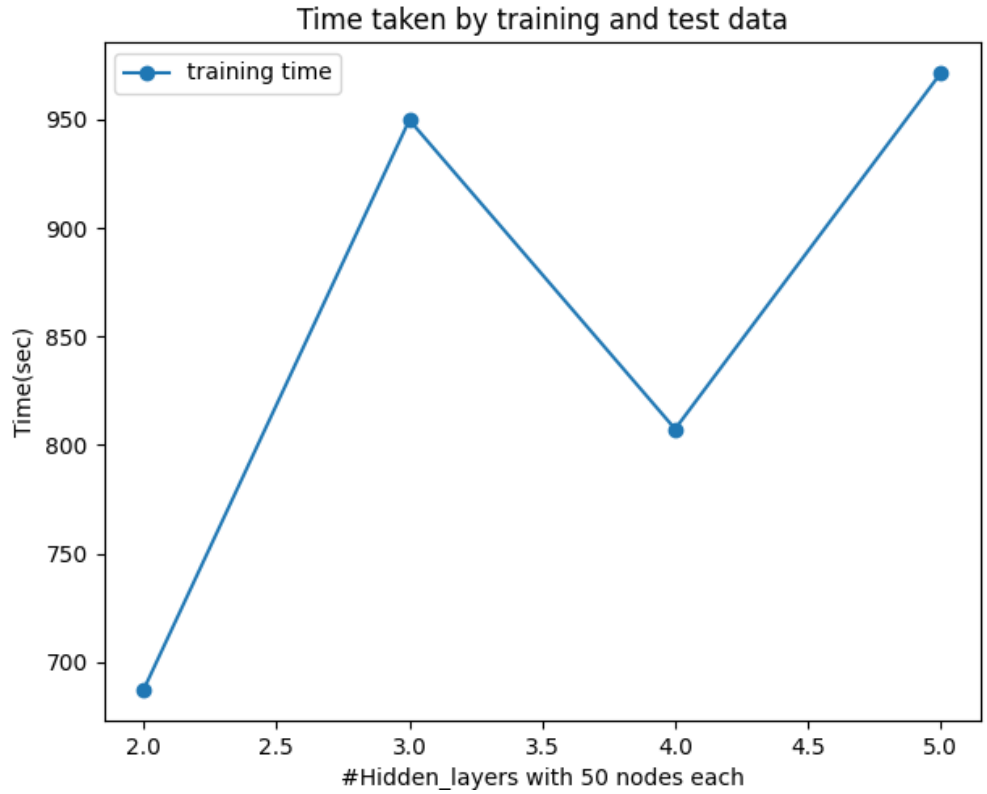
Training Accuracy: 10.051834197236621%

Test Accuracy: 10.021002100210021%

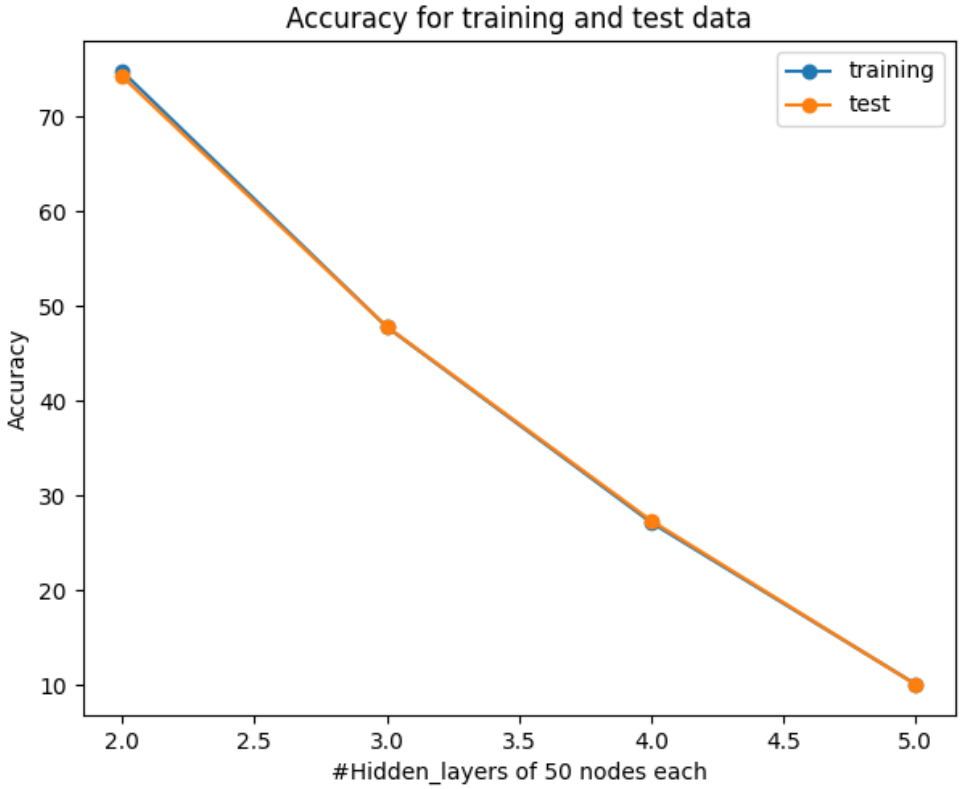
Best performing model is **ReLU** with 3 **[50, 50, 50]** hidden layers. As we start increasing the number of layers in sigmoid model it's performance started deteriorating.

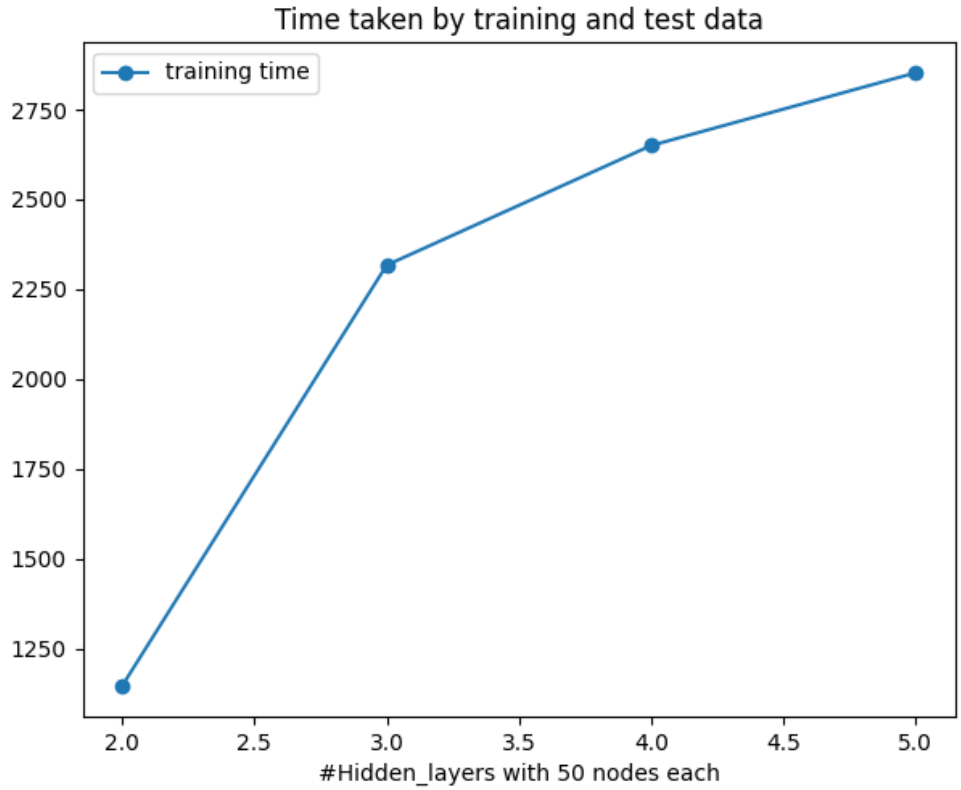
Graphs for **ReLU** activation function in Hidden layers





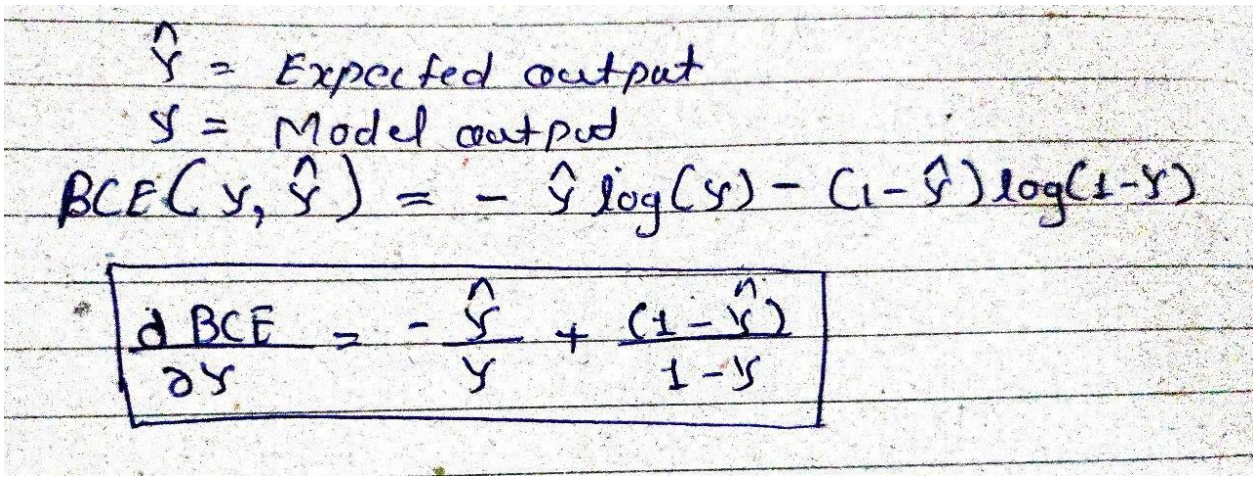
Graphs for **Sigmoid** activation function in Hidden layers





Observation: network with ReLU activation layer accuracy increased and saturated around 86% with increasing number of hidden layers. In case of sigmoid accuracy actually start decreasing with increasing hidden layers because of diminishing gradient problem I guess.

f) Binary-Cross Entropy (BCE) loss



Best model of 2e) three [50, 50, 50] hidden layers

Time taken to train: 1273.2066054344177

Training Accuracy: 86.60811013516891%

Test Accuracy: 84.74847484748474%

Training set

----------confusion matrix----------

[[4940 10 68 169 8 1 793 0 15 0]

[16 5783 11 63 10 0 18 0 3 2]

[95 39 4715 43 517 1 656 0 23 0]

[330 135 68 5378 212 5 225 0 42 0]

[16 14 680 197 4837 0 544 0 20 0]

[8 0 4 0 1 5566 0 139 19 96]

[541 16 427 133 390 3 3687 0 103 1]

[0 0 1 0 0 275 3 5603 24 189]

[53 3 25 15 24 29 73 17 5747 3]

[1 0 1 2 1 120 1 241 4 5708]]

Test set

----------confusion matrix----------

[[802 2 16 24 0 0 145 0 0 0]

[2 953 3 14 2 0 1 0 0 0]

[14 9 760 5 109 0 121 0 5 0]

[59 29 17 874 40 1 49 0 6 0]

[6 4 115 36 779 0 94 0 10 0]

[1 0 1 1 0 911 2 28 10 15]

[104 3 85 41 64 0 573 0 16 1]

[0 0 0 0 0 52 0 933 4 43]

[11 0 3 4 6 5 15 1 949 0]

[1 0 0 1 0 31 0 38 0 940]]

---

g) MLPClassifier from scikit-learn library

Iteration 200, loss = 0.00206089

Time taken to train: 906.4067075252533

model [50, 50, 50] hidden layers

Training Accuracy: 99.99%

Test Accuracy: 86.75%

Observation: when compared with 2f sklearn implementation is slightly Faster and training accuracy is significantly better then my implementation but the test accuracy is same in both case i.e, around 86%.