COL774, Assignment-3

2019CS50448, Satyam Kumar Modi

1. (a) **Decision Tree Construction and Visualization**

There were missing values in the sample and they were dropped for training and testing.

Now, from the filtered data, we trained the decision tree and obtained the following accuracies :-

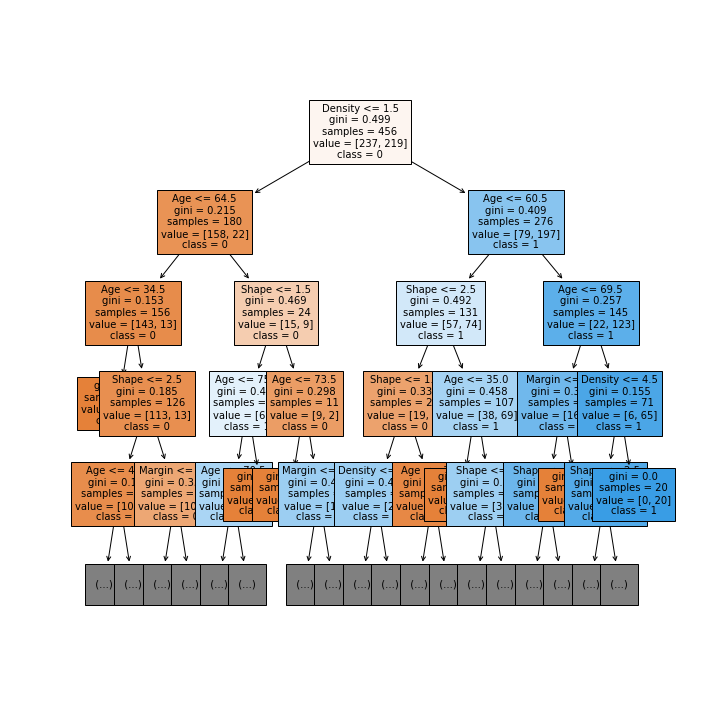
i. Training Accuracy: 92.324

ii. Test Accuracy: 69.565

iii. Validation Accuracy: 76.033

Depth of the tree: 18

Number of leaves: 146



**(b) Decision Tree Grid Search**

The Grid Search is done over the following parameters:

i. **Criterion** : [Gini, Entropy ]

ii. **Max\_depth** : [ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 ]

iii. Max\_features : [ 1, 2, 3 ]

iv. Min\_Samples\_leaf : [ 1, 2, 3, 4, 5 ]

v. Min\_Samples\_split : [ 2, 3, 4, 5, 6, 7 ]

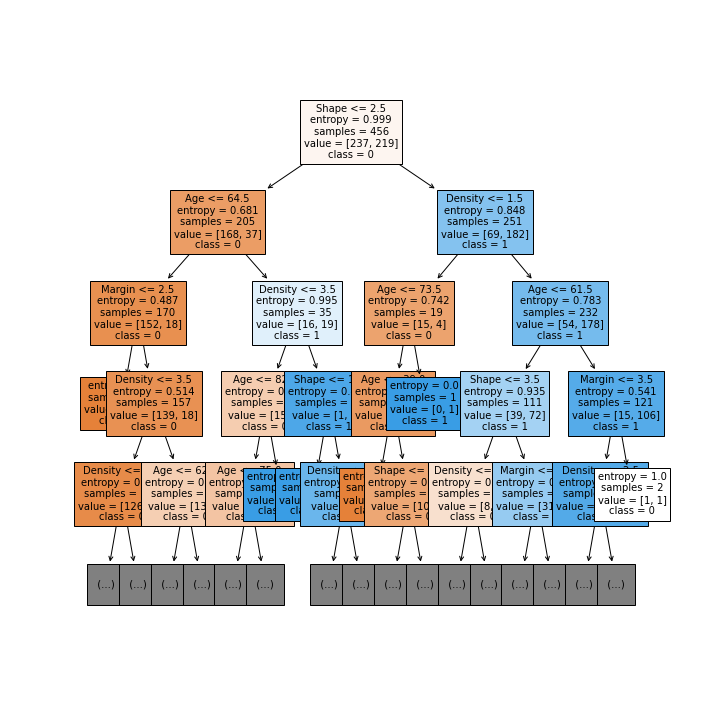
Best Parameter obtained: {'criterion': 'entropy', 'max\_depth': 6, 'max\_features': 2, 'min\_samples\_leaf': 1, 'min\_samples\_split': 3}

Accuracy on the best parameter obtained:

Training accuracy: 81.14035087719299

Test accuracy: 77.4703557312253

Validation accuracy: 89.25619834710744



Depth of the tree: 6

Number of leaves of the tree: 31

We notice that the divison on the first node in part(a) was done on the basis of **Density <= 1.5** whereas

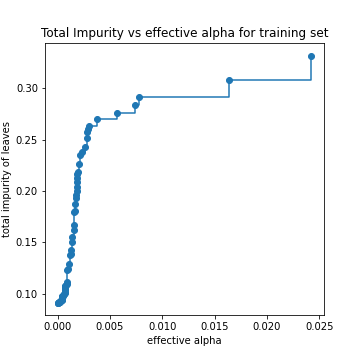
in the optimal decision tree, it is done on the basis of **Shape <= 2.5.** The entropy obtained when **Shape** is

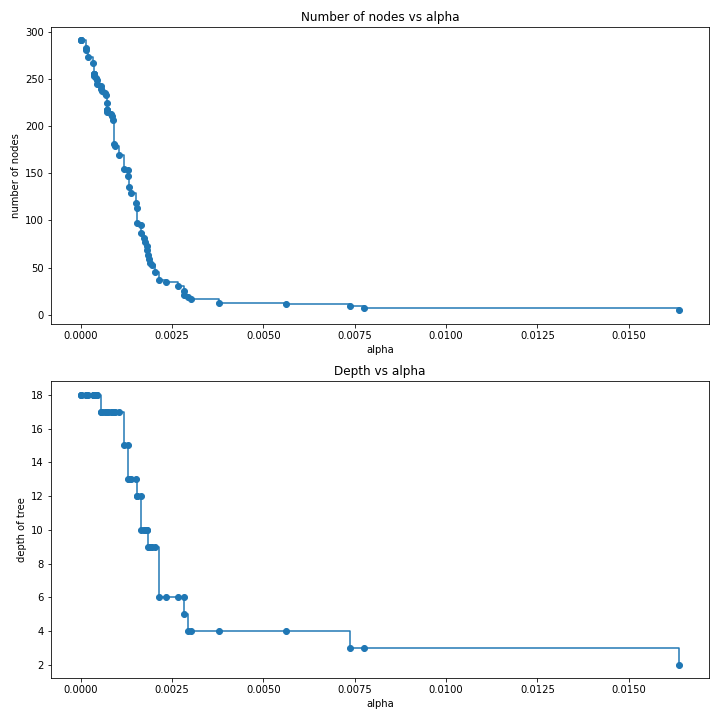
taken into account is **0.999** in Part(b) and in the next level of the optimal tree, the **Density** is taken into

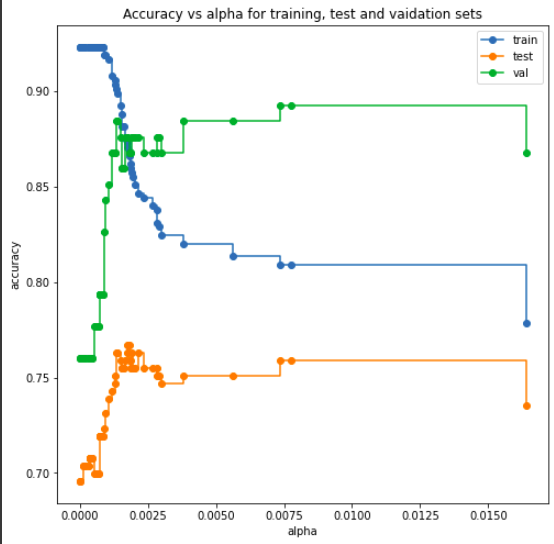
account for splitting which has an entropy of **0.848**.

**Depth** of the tree obtained in part-(a) is **16** whereas in part-(b), the optimal tree has a depth of **6**. This is reflected in the **decrement** of **training accuracy** but **increase** in the **test** and **validation accuracy.** This also gives us an indication that the tree in (a) was **overfitted**.

(c) **Decision Tree Post Pruning (Cost Complexity Pruning)**



****



The decision tree was prunned using the cost\_complexity\_pruning\_path function of the Decision tree which return the corresponding alphas and total leaf impurities at each step of the prunning process. Higher the value of alpha, the more the tree is prunned which increases the total leaf impurity. This can be noticed in the graph plotted between **impurity of leaves** vs **effective alpha.** We also notice that as the **alpha increases**, since more of tree is being prunned, this **decreases the number of nodes and the depth** of the tree as can be seen in the graph.

Best alpha so obtained: 0.007370857699805071

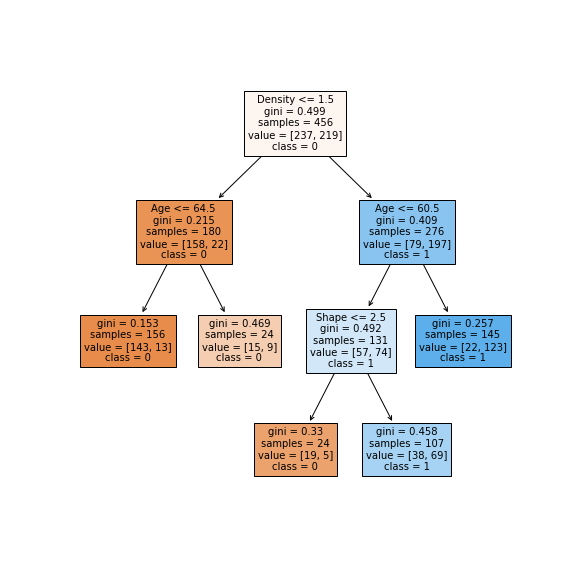
We obtained the best alpha by finding out the alpha which gave the maximum accuracy on validation set.

**Training accuracy: 80.92105263157895**

**Test accuracy: 75.8893280632411**

**Validation accuracy: 89.25619834710744**

Decision Tree so obtained had a **depth of 3 with 5 leaves.**



Since, the depth of the prunned tree has decreased, the test accuracy has decreased by 2% in compared to part-(b). The tree obtained in part(c) is a subtree of that obtained in part-(a).

(d) **Random Forests**

GridSearchCV was done in following parameters with following values:

criterion: ['gini', 'entropy']

max\_features: [1, 2, 3, 4]

min\_samples\_split: [5, 10, 15, 20, 25]

n\_estimators: [50, 60, 70, 80, 90, 100]

Best Parameter obtained: {'criterion': 'entropy', 'max\_features': 2, 'min\_samples\_split': 20, 'n\_estimators': 60}

Training accuracy: 83.552

Test accuracy: 77.865

Validation accuracy: 86.776

Oob score: 0.793

(e) When the data is imputated by median

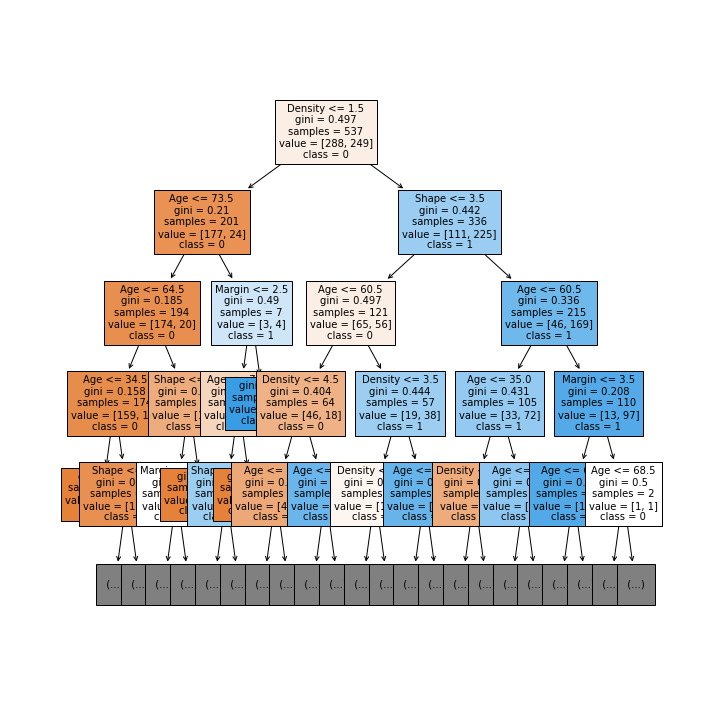
(i) Training accuracy: 91.806

Test accuracy: 73.6111

Validation accuracy: 74.074

Depth of the tree : 17

Number of leaves : 168



(ii) The GridSearch is done over the following parameters:

i. **Criterion** : [Gini, Entropy ]

ii. **Max\_depth** : [ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 ]

iii. Max\_features : [ 1, 2, 3 ]

iv. Min\_Samples\_leaf : [ 1, 2, 3, 4, 5 ]

v. Min\_Samples\_split : [ 2, 3, 4, 5, 6, 7 ]

Best Parameter obtained: {'criterion': 'entropy', 'max\_depth': 6, 'max\_features': 2, 'min\_samples\_leaf': 1, 'min\_samples\_split': 3}

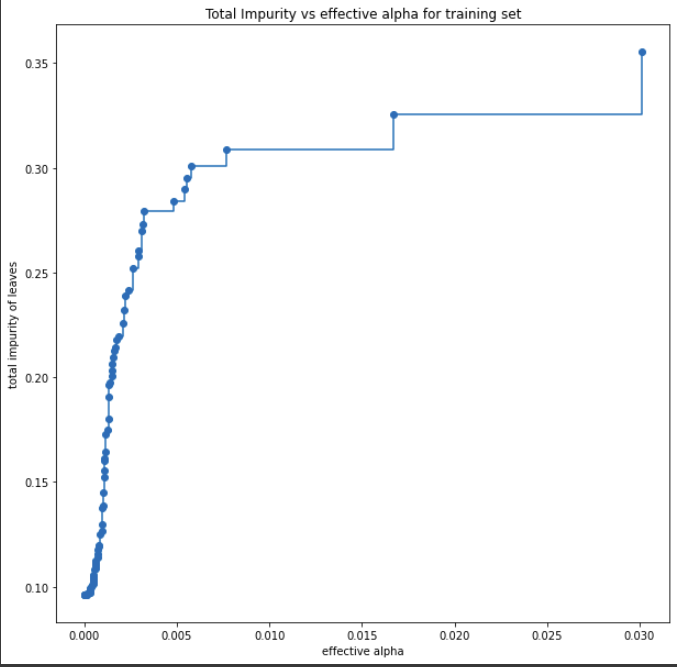
Accuracy on the best parameter obtained:

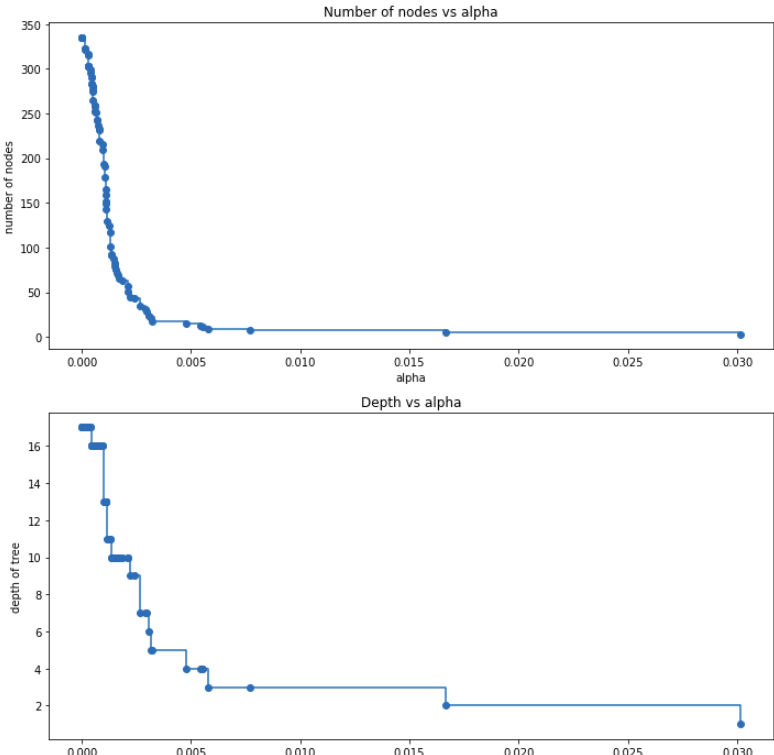
Training accuracy: 81.14035087719299

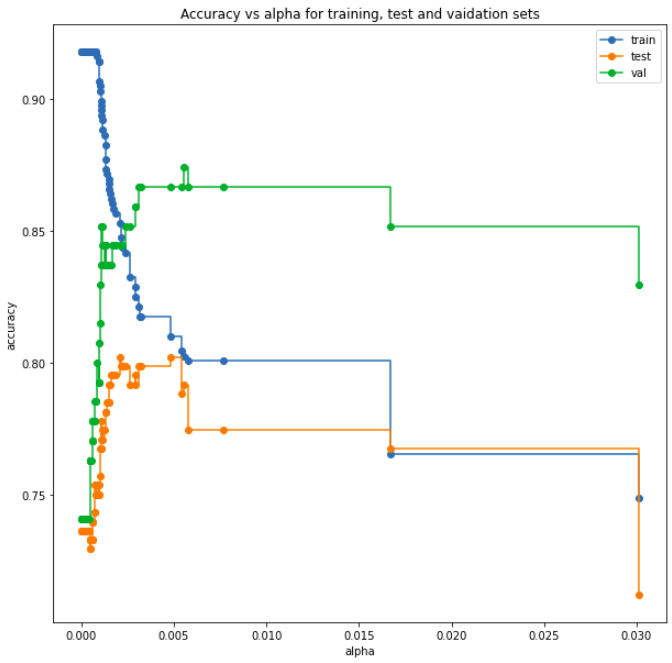
Test accuracy: 77.4703557312253

Validation accuracy: 89.25619834710744

(iii.)







The decision tree was prunned using the cost\_complexity\_pruning\_path function of the Decision tree which return the corresponding alphas and total leaf impurities at each step of the prunning process. Higher the value of alpha, the more the tree is prunned which increases the total leaf impurity. This can be noticed in the graph plotted between **impurity of leaves** vs **effective alpha.** We also notice that as the **alpha increases**, since more of tree is being prunned, this **decreases the number of nodes and the depth** of the tree as can be seen in the graph.

**Best alpha** obtained: 0.0055191

Accuracy obtained for this alpha:

**Training accuracy**: 80.260

**Test accuracy**: 79.166

**Validation accuracy**: 87.407

**Observations:**

The test accuracy has increased from **75.88% to 79.166%.** Since, the alpha obtained in this case was less than that in part (c). This means less of the decision tree was prunned in comparison to part (c). One can also attribute this difference in the accuracy because after imputation, the **size of training data increased**.

iv. Random Forest

GridSearchCV was done in following parameters with following values:

criterion: ['gini', 'entropy']

max\_features: [1, 2, 3, 4]

min\_samples\_split: [5, 10, 15, 20, 25]

n\_estimators: [50, 60, 70, 80, 90, 100]

Best Parameter obtained: {'criterion': 'entropy', 'max\_features': 4, 'min\_samples\_split': 25, 'n\_estimators': 80}

Training accuracy: 83.240

Test accuracy: 79.166

Validation accuracy: 85.925

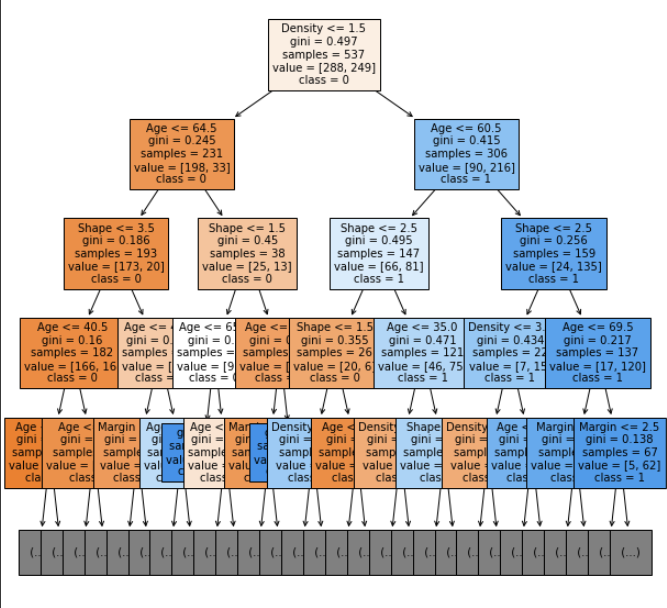
Oob score: 0.774

**For the mode imputation**

i. Training accuracy: 90.689

Test accuracy: 70.833

Validation accuracy: 77.037



ii. Grid Search was done over the following parameters:

i. **Criterion** : [Gini, Entropy ]

ii. **Max\_depth** : [ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 ]

iii. Max\_features : [ 1, 2, 3 ]

iv. Min\_Samples\_leaf : [ 1, 2, 3, 4, 5 ]

v. Min\_Samples\_split : [ 2, 3, 4, 5, 6, 7 ]

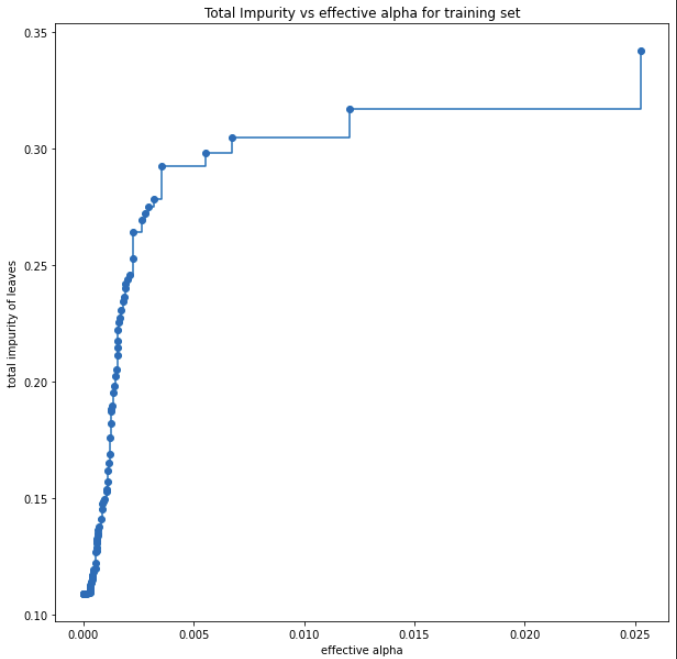
**Best param obtained: {'criterion': 'entropy', 'max\_depth': 6, 'max\_features': 2, 'min\_samples\_leaf': 3, 'min\_samples\_split': 4}**

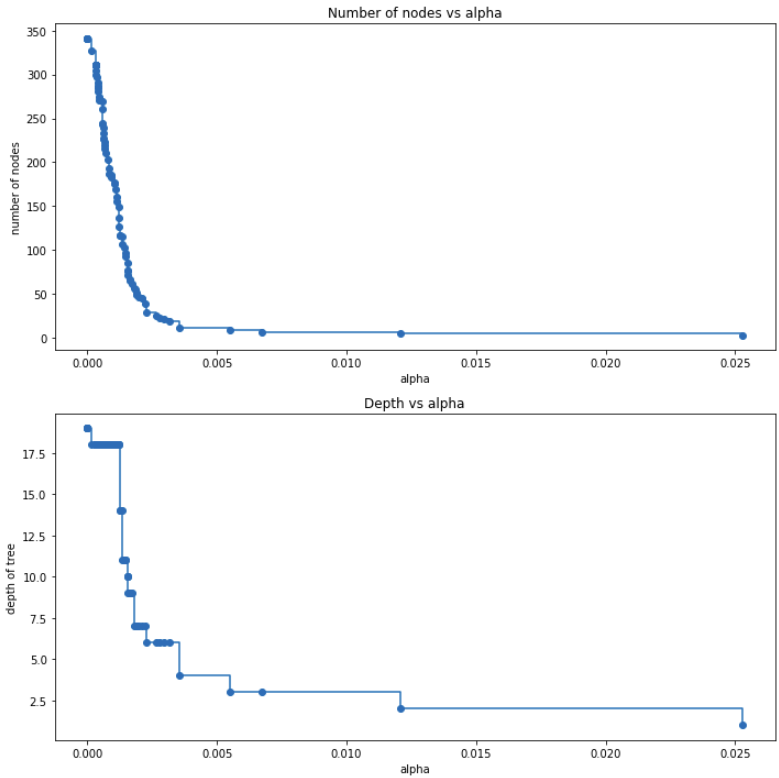
**Training accuracy**: 81.750

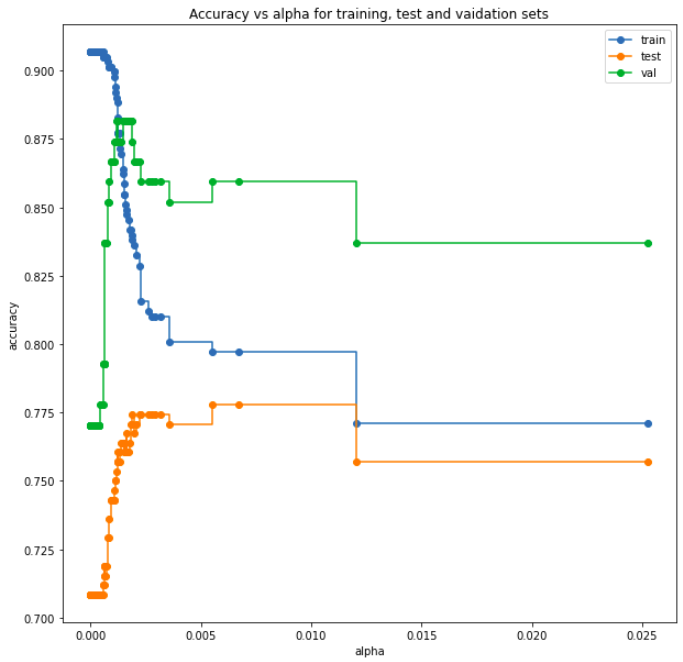
**Test accuracy**: 78.472

**Validation accuracy**: 85.925

iii. **Decision Tree Post Pruning (Cost Complexity Pruning)**







**Best alpha obtained: 0.001201**

Accuracies for the best alpha obtained:

**Training accuracy**: 89.013

**Test accuracy:** 75.347

**Validation accuracy:** 88.148

iv. **Random Forest**

GridSearchCV was done in following parameters with following values:

criterion: ['gini', 'entropy']

max\_features: [1, 2, 3, 4]

min\_samples\_split: [5, 10, 15, 20, 25]

n\_estimators: [50, 60, 70, 80, 90, 100]

Best Parameter obtained: {'criterion': 'entropy', 'max\_features': 2, 'min\_samples\_split': 20, 'n\_estimators': 70}

Training accuracy: 82.867

Test accuracy: 79.166

Validation accuracy: 86.666

Oob score: 0.769

(f) **Gradient Boosted Trees**

Grid Search was done over the following parameters in the given range:

n\_estimators = [10, 20, 30, 40, 50]

subsample = [0.1, 0.2, 0.3, 0.4, 0.5]

max\_depth = [4, 5, 6, 7, 8, 9, 10]

Result of the Grid Search :

**Training accurac**y: 81.564

**Test accuracy**: 79.166

**Validation accuracy**: 85.185

2. For Dataset 2

(a) Training accuracy : 99.641

Test accuracy: 57.091

Validation accuracy: 57.066

Depth of the tree: 826

Number of leaves: 42429

(b) We performed the Grid Search over these parameters:

max\_depth: [200, 210, 220, 230, 240]

min\_samples\_split = [5, 10]

min\_samples\_leaf = [2, 4]

Grid Search on the decision tree gave the following output:

Best Parameter obtained: {'max\_depth': 240, 'min\_samples\_leaf': 2, 'min\_samples\_split': 5}

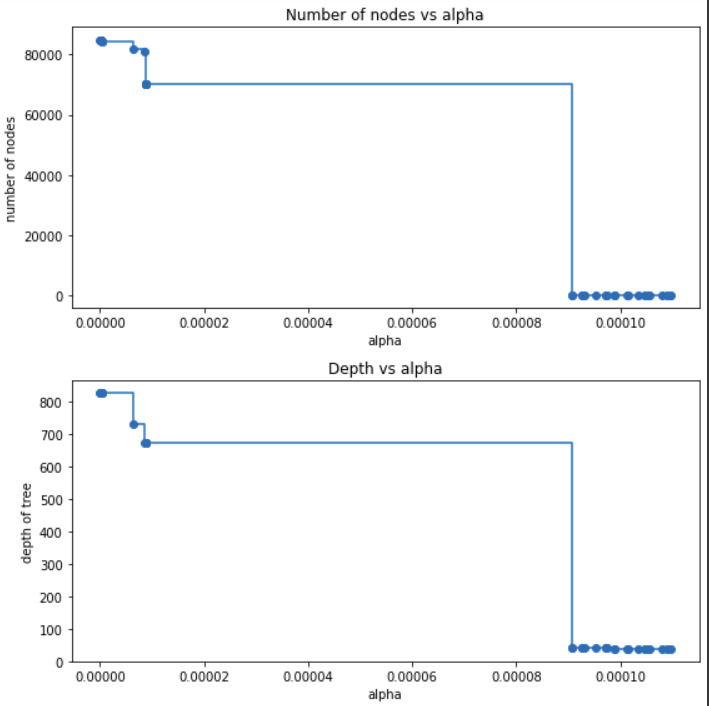
Accuracy on the best parameter obtained:

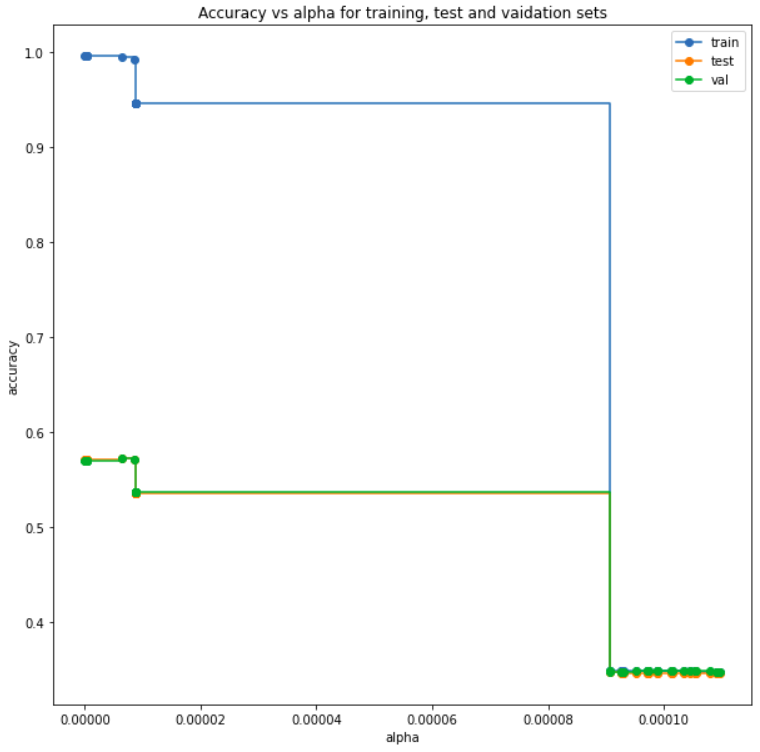
Training accuracy: 37.40

Test accuracy: 36.076

Validation accuracy: 36.3512

(c) Decision Tree Post Prunning





We performed the Prunning using Cost Complexity Prunning method. We obtained 23786 values of alpha.

We trained the model for the first and the last 20 values of alpha obtained and got the follwing results:

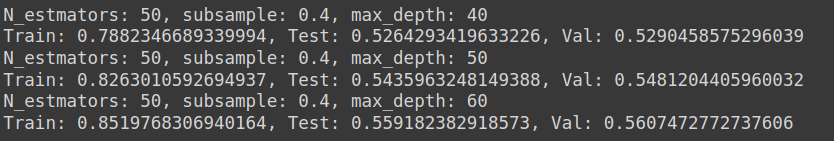
Best alpha obtained: 6.36e-6

Training accuracy: 99.5

Test accuracy: 57.29

Validation accuracy: 59.69

(d) Grid Search over Random Forest



(f) LightGBM

We trained the model using the LightGBM framework and obtained the following result:

Training accuracy: 90.74

Test accuracy: 59.39

Validation accuracy: 59.69

Parameters used are as follows:

num\_leaves: 500

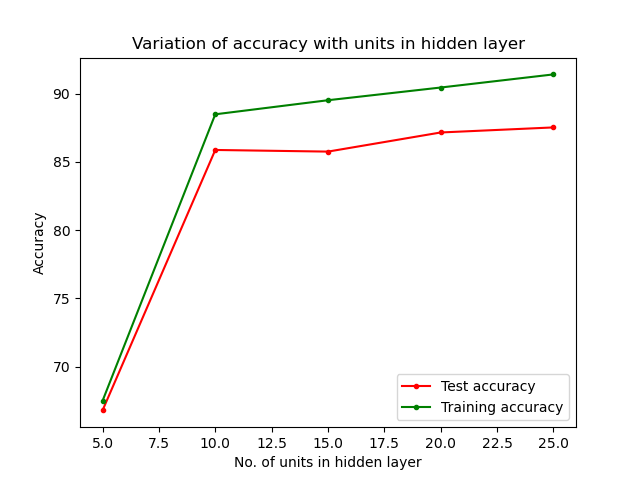
num\_class: 10

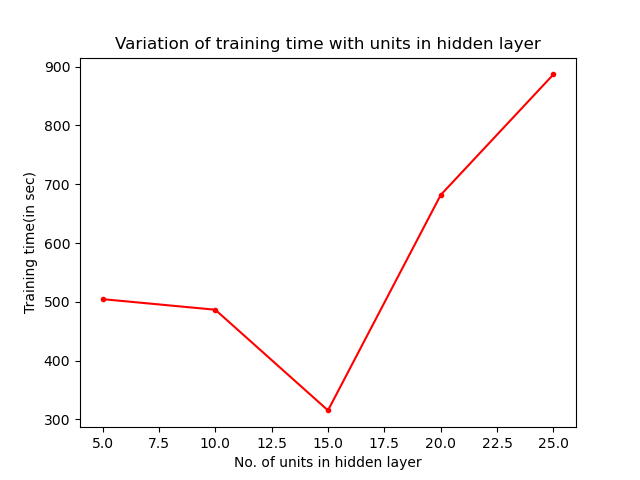
objective: ‘multiclass’

We obtained an increase of 2% in the test and validation accuracy roughly.

3. **Neural Network**

(b)





We trained the neural network by varying the number of hidden units in [5, 10, 15, 20, 25]. The learning rate was set to 0.1. We used SGD with mini-batch(size=100) processing of training data.

Stopping Criterion: When the difference in training accuracy between two consecutive epochs is < 2.5e-5.

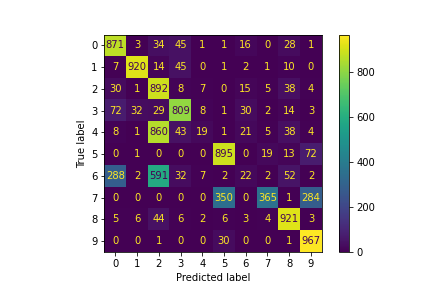
Observations:

i. We note that the training and test accuracy almost increased with increasing number of hidden units. This was because more number of hidden units are present which gives more amount of information about the data.

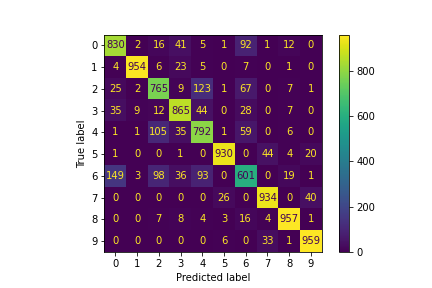
ii. The training time decreased between 5 and 15. This was due to the model taking lesser number of epochs to reach the convergence point. From 15 to 25, the training time increases.

Confusion matrix

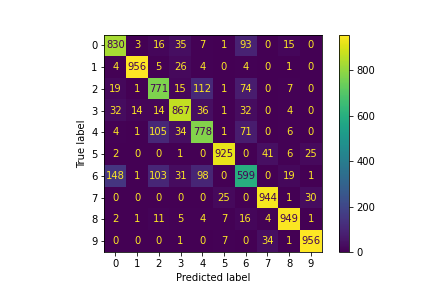
For n = 5



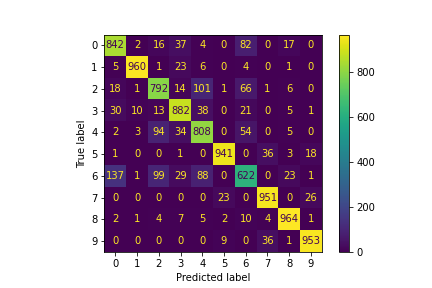
For n = 10



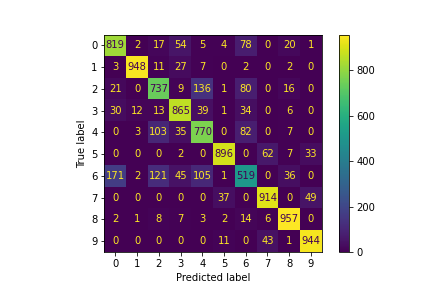
For n = 15



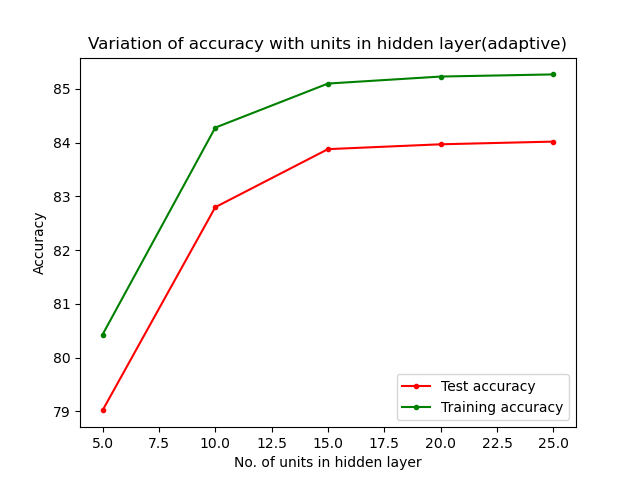
For n = 20

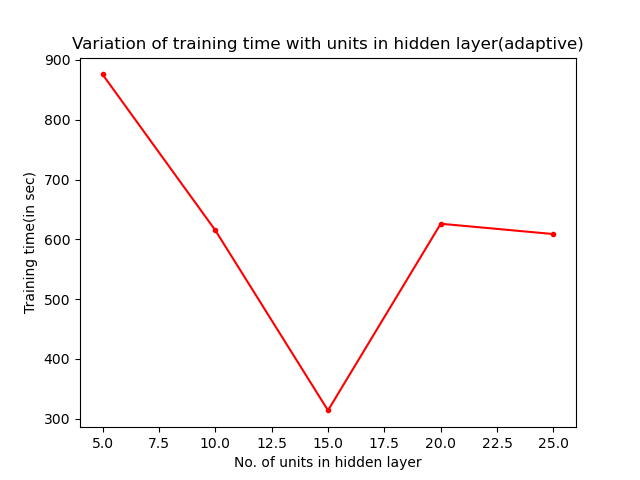


For n = 25



(c) Using adaptive learning rate





We trained the neural network by varying the number of hidden units in [5, 10, 15, 20, 25]. The learning rate was set to 0.1. We used SGD with mini-batch(size=100) processing of training data.

Stopping Criterion: When the difference in training accuracy between two consecutive epochs is < 2.5e-5.

Observations:

i. We note that the training and test accuracy almost increased with increasing number of hidden units. This was because more number of hidden units present more amount of information about the data.

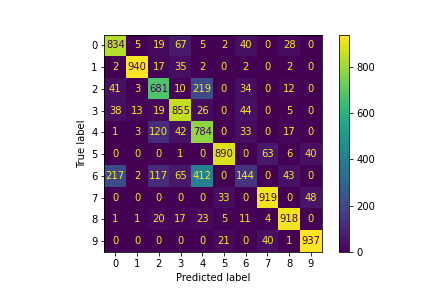
ii. The training time decreased between 5 and 15. This was due to the model taking lesser number of epochs to reach the convergence point. From 15 to 20, the training time increases.

Iii. The training speeded up for higher values of number of hidden units.

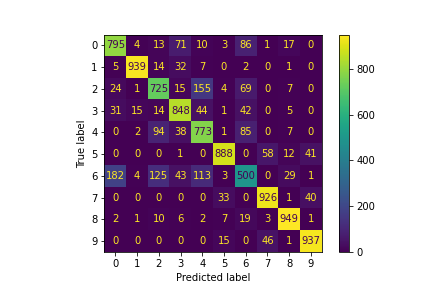
iv. The test and training accuracy has dropped from the part-(b).

Confusion Matrix

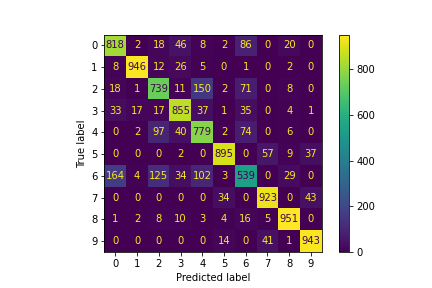
For n = 5



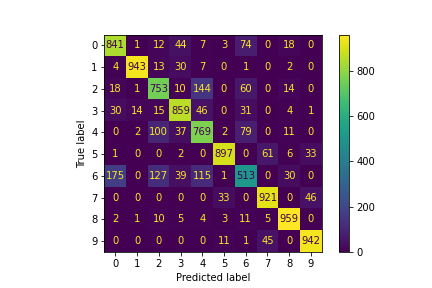
For n = 10



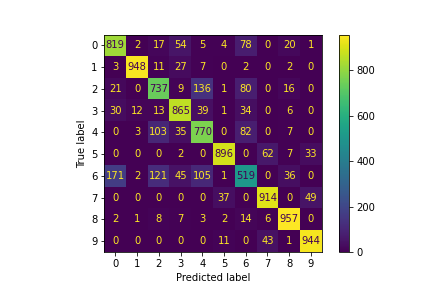
For n = 15



For n = 20

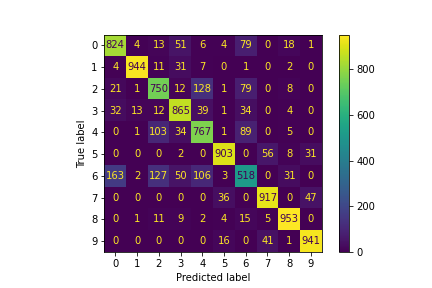


For n = 25



(d) Relu activation

We trained the network on 100 \* 100 hidden layers for both the sigmoid and relu activation functions using adaptive learning rate.

For Sigmoid:

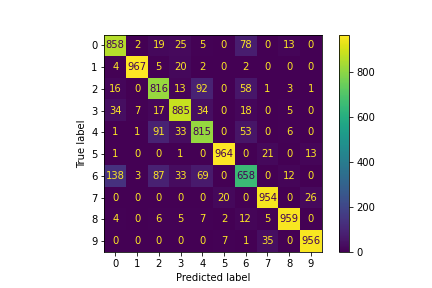
Time taken to train: 1840s

Training accuracy: 85.35

Test accuracy: 83.82

Epochs: 540

For Relu:



Time taken to train: 1014s Stopping Criterion: | L(t+1)-L(t)| < 2.5e-5

Training accuracy: 92.38

Test accuracy: 88.32

Epochs: 458

Observations:

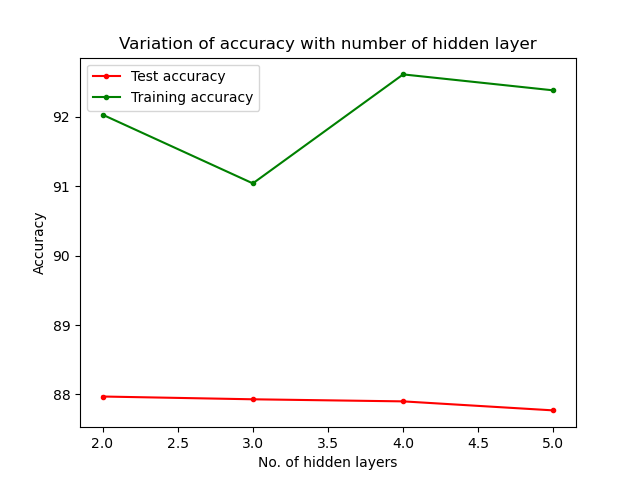
i. Relu takes lesser time to train. Also, both training and test accuracy increases with relu activation function.

ii. The Relu activation was able to learn more examples from Class-3 and Class-6 comparatively higher than Sigmoid.

Iii. The Sigmoid model overfits when trained on 100 \* 100 hidden layer. This is evident from the training accuracy increasing and test accuracy decreasing in comparison to part-(b) where a single layer with lesser number of hidden units were used.

(e) Multilayer networks

We trained the model by varying the number of hidden layers from 2 to 5 , each layer having 50 hidden units.



We notice that as the number of hidden layer increases in the multilayer network, the train accuracy decreases from 2 to 3(slight decrease) but increases from 3 to 4. The test accuracy decreases very slightly with increasing number of layers in the network. This may be due to overfitting.

(f) BCE Loss

Here, we used the binary cross entropy loss.

M k

J(θ) = ∑ ∑ (1{y(i) = j)(-log pij) + (1 - 1{y(i) = j)) (-log(1-pij)) )

i = 1 j=1

∂J = 1{y(i) = k).(-1/o(k)) + (1 - 1{y(i) = k)).(1/(1-o(k)))

∂o(k)

Used Relu as the activation unit in the hidden layers.

Trained on the 50 × 50 layers.

Training accuracy: 93.90

Test accuracy: 87.85

Epsilon: 4e-6

(g) Here, we used the scikit MLP Classifier.

Training acc: 89.246

Test acc: 83.82

We note that the training accuracy as well the test accuracy in part-(f) is more than the scikit implmentation.