COL-774 ASSIGNMENT-3

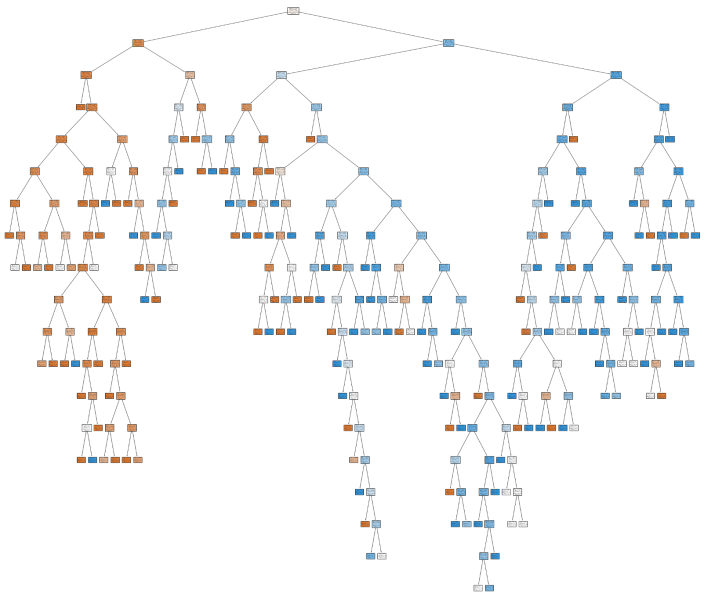
Name : Gattu Karthik **Entry No** : 2019CS10348

Mammography

a)

Trainin_accuracy	Test_accuracy	Validation_accuray
92.32	69.16	76.03

Tree_visiualization:



b)

'max_depth':[1,2,3,4,5]

'min_samples_leaf':[1,2,3,4,5]

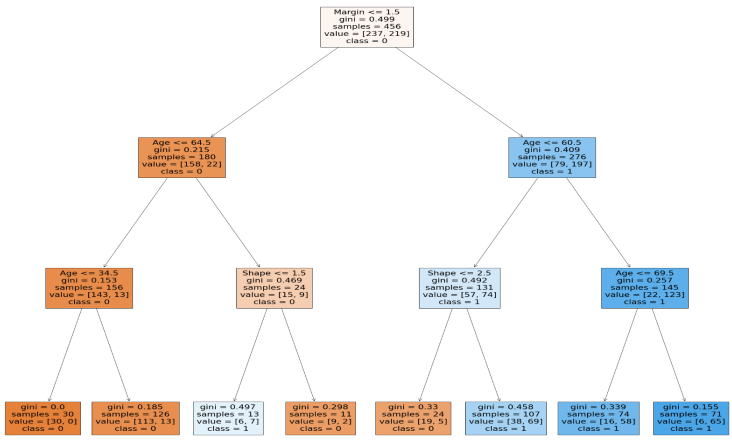
'Min_samples_split':[2,3,4,5]

Optimal _parameters = {'max_depth': 3,'min_samples_leaf': 2, 'min_samples_split': 2}

Accuracies for optimal_parameteres

Trainin_accuracy	Test_accuracy	Validation_accuray
81.14	75.49	87.60

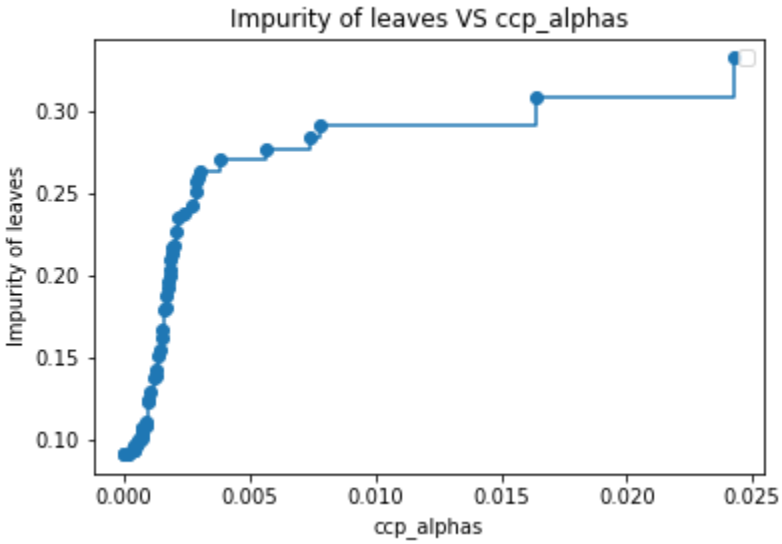
Optimal_Tree_visiualization:



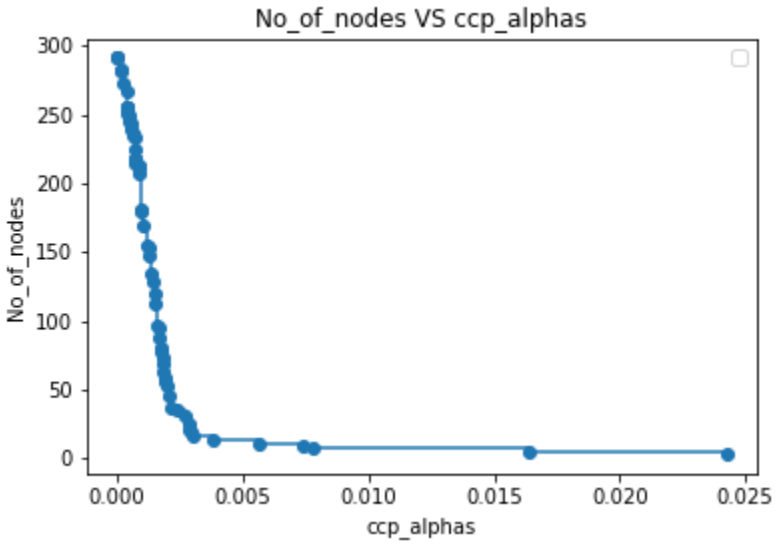
Height and no:of nodes reduced significantly when compared to part_a.

c)

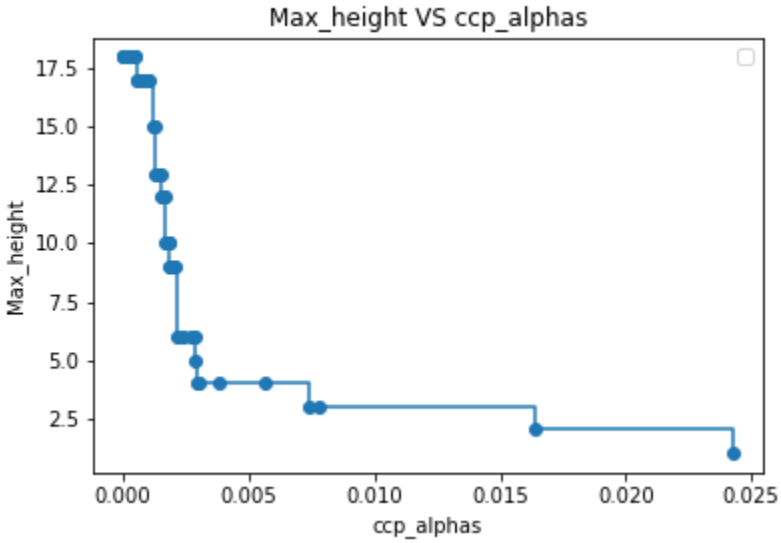
Total impurity of leaves (vs) Effective alphas of pruned tree.



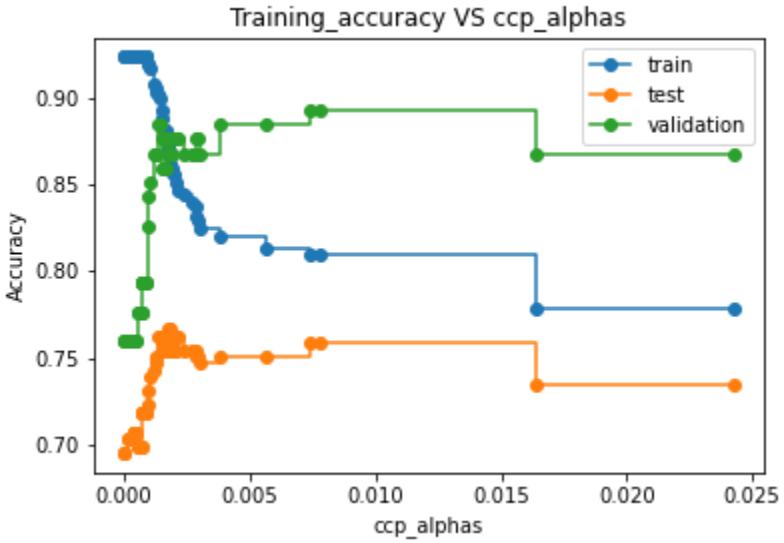
Nodes (vs) alpha



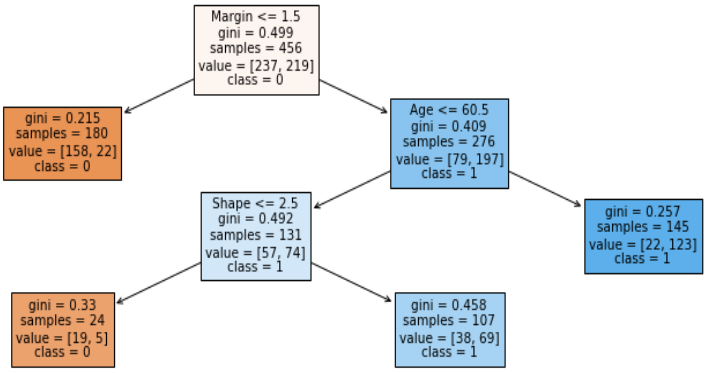
Depth of tree (vs) alpha



Training accuracy, Validation accuracy, and Test accuracy vs alpha.



Best performing tree -based on Validation split



Tree obtained is even smaller than that obatined in part_b and part_a.

Trainin_accuracy	Test_accuracy	Validation_accuray
80.92	75.88	89.25

d)

"n_estimators" : [100,200,300,400,500],

"max_features" : ['sqrt'],

"min_samples_split" : [2,3,4,5]

Optimal_parameters are : {'max_features':'sqrt', 'min_samples_split': 5, 'n_estimators': 100}

Trainin_accuracy	Test_accuracy	Validation_accuray	OOB_accuracy

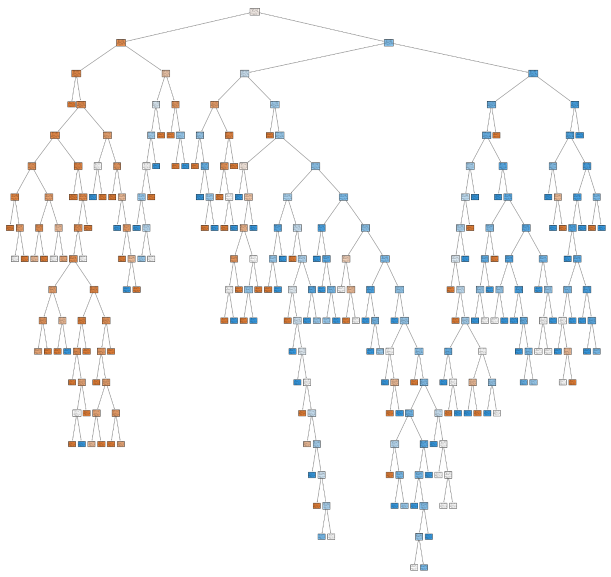
88.59	77.07	85.95	76.53

e)

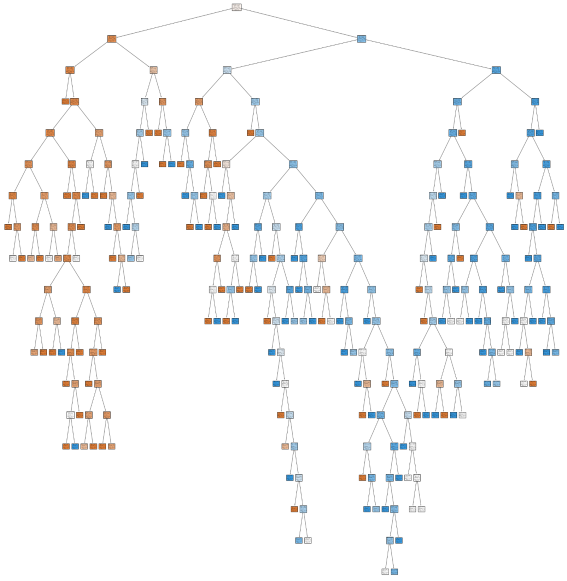
a)

Imputation_type	Trainin_accuracy	Test_accuracy	Validation_accuray
Median	92.32	69.16	76.03
Mode	92.3245	69.1699	76.0330

Tree_visiualization_median



Tree_visiualization_mode



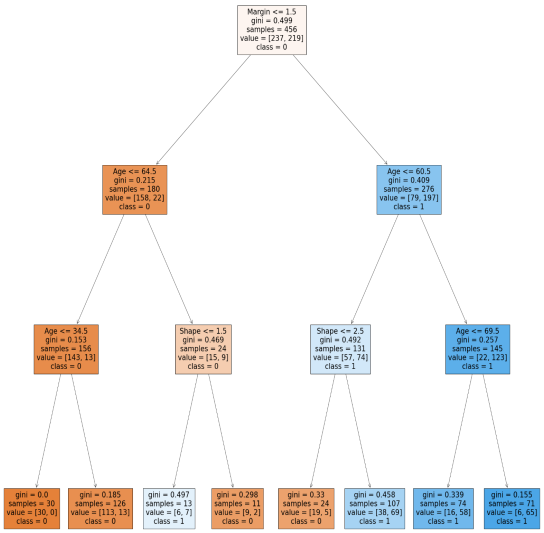
b)

optimal _parameters_median ={'max_depth': 3, 'min_samples_leaf': 2, 'min_samples_split': 2}

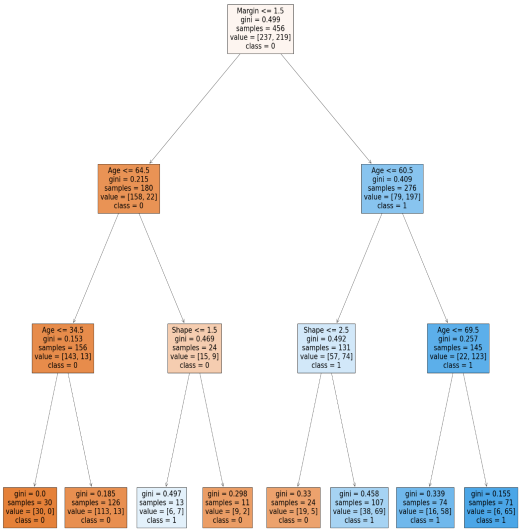
optimal _parameters_mode = {'max_depth': 3, 'min_samples_leaf': 2, 'min_samples_split': 2}

Imputation_type	Trainin_accuracy	Test_accuracy	Validation_accuray
Median	81.1403	75.4940	87.6033
Mode	81.1403	75.4940	87.6033

opt_Tree_visiualization_median



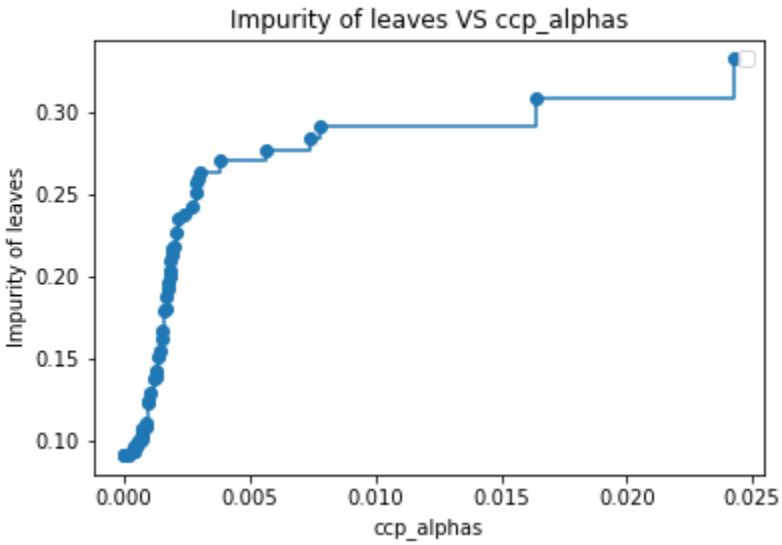
opt_Tree_visiualization_mode



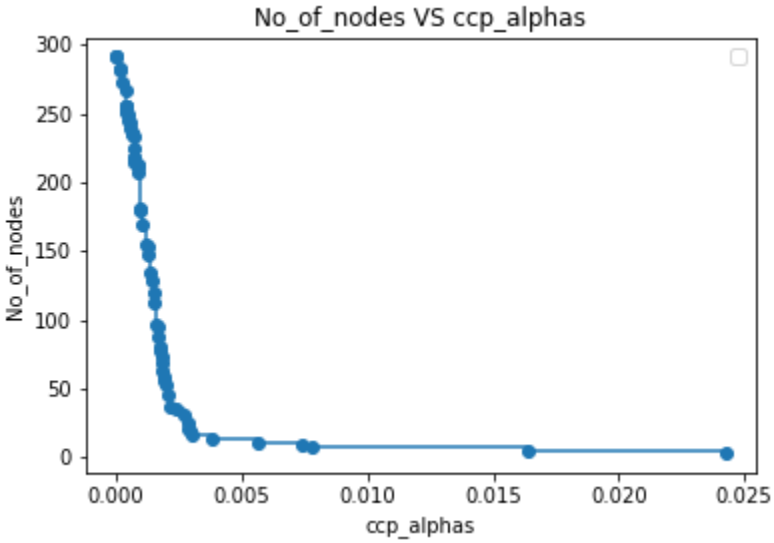
c)

MEDIAN

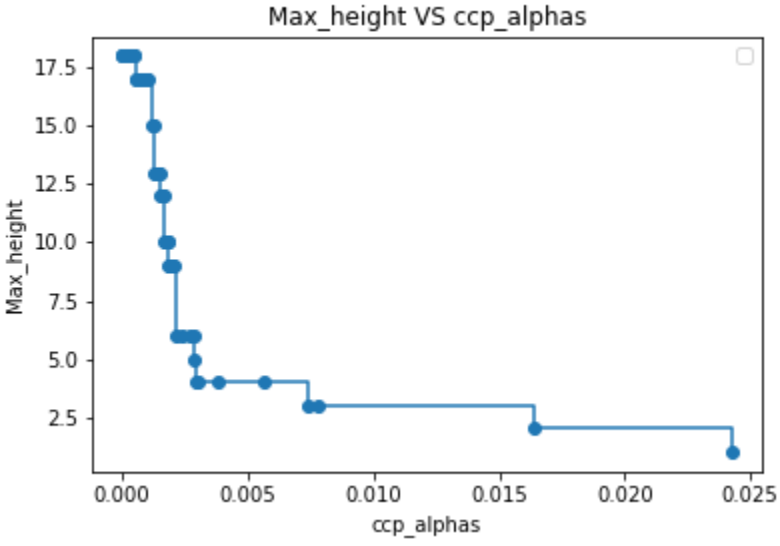
Total impurity of leaves vs Effective alphas.



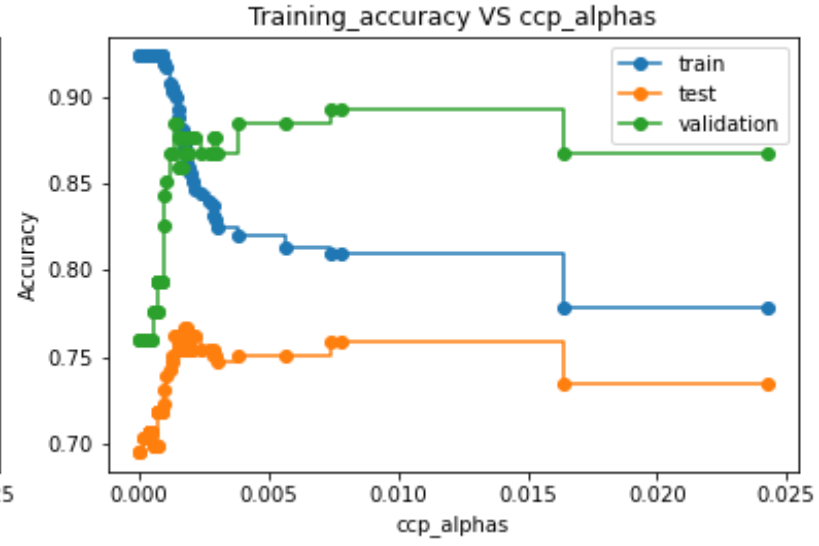
Nodes vs alpha



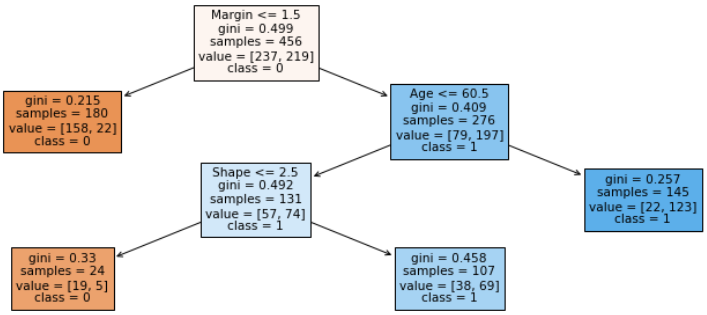
Depth of tree vs alpha



Training accuracy, Validation accuracy, and Test accuracy vs alpha.

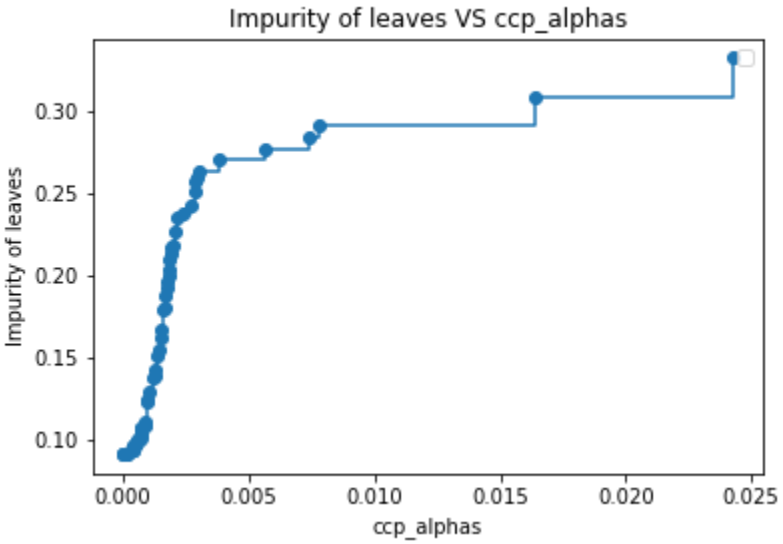


Best performing tree -based on validation split

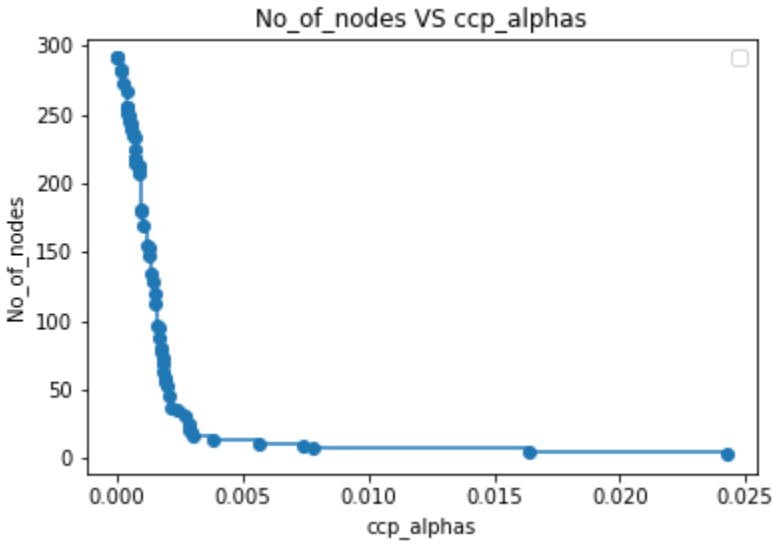


MODE

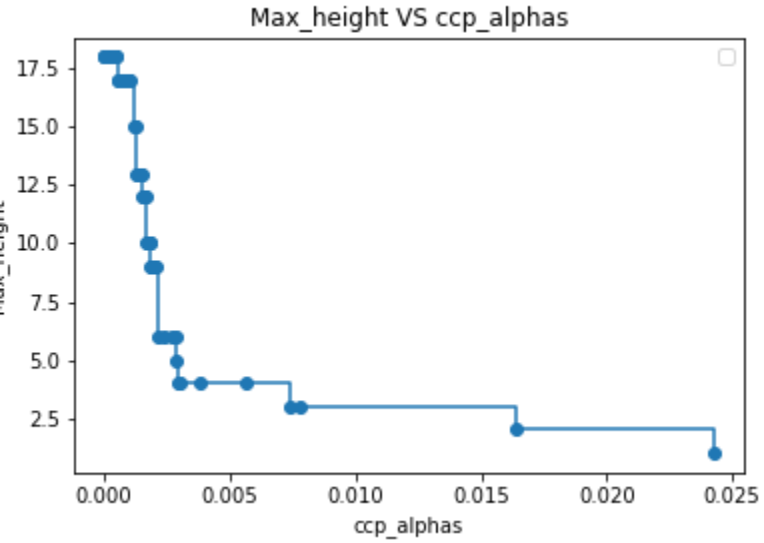
Total impurity of leaves vs Effective alphas of pruned tree.



Nodes vs alpha

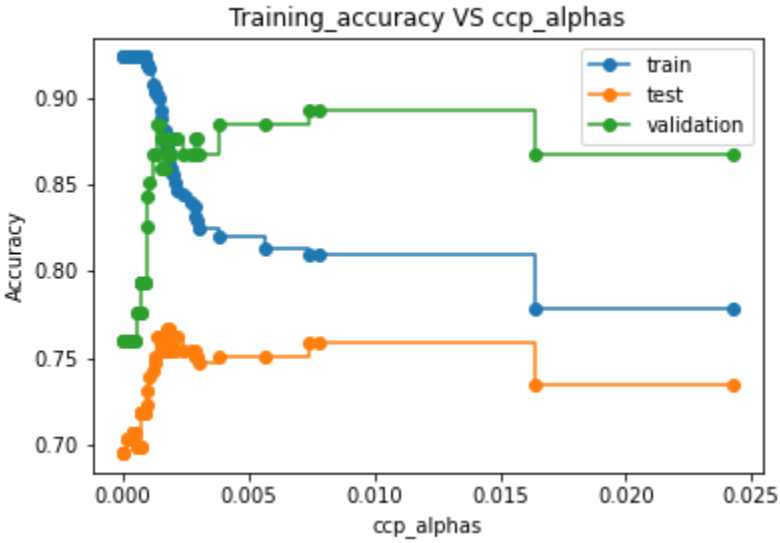


Depth of tree vs alpha

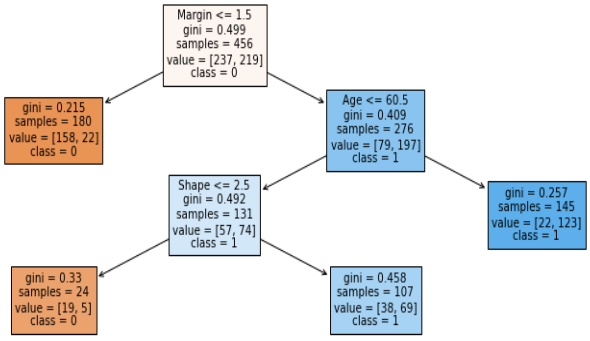


Training accuracy, Validation accuracy,

and Test accuracy vs alpha.



Best performing tree -based on validation split



Imputation_type	Trainin_accuracy	Test_accuracy	Validation_accuray
Median	80.9210	75.8893	89.2561

Mode	80.9210	75.8893	89.2561

d)

optimal _parameters_median ={'max_features':'sqrt', 'min_samples_split': 4, 'n_estimators': 100}

optimal _parameters_mode = {'max_features': 'sqrt', 'min_samples_split': 5, 'n_estimators': 100}

Imputation_type	Trainin_accuracy	Test_accuracy	Validation_accuray	OOB_accuracy
Median	90.5701	76.2845	82.6446	74.5614
Mode	89.4736	76.2845	83.4710	75.8771

The both methods of imputation gave same results as of dropping for all parts except for random forest (this also very slight deviation).

f)

"n_estimators" : [10,20,30,40,50],

"subsample" : [0.1,0.2,0.3,0.4,0.5,0.6],

"max_depth" : [4,5,6,7,8,9,10]

Optimal_parameters : {'max_depth': 10, 'n_estimators': 10, 'subsample': 0.5}

Trainin_accuracy	Test_accuracy	Validation_accuray
83.61	77.08	84.44

Drug_rating

a)

Trainin_accuracy	Test_accuracy	Validation_accuray
99.97	57.61	58.12

b)

'max_depth':[1,2,3,4,5]

'min_samples_leaf':[1,2,3,4,5]

'Min_samples_split':[2,3,4,5]

Optimal _parameters = {'max_depth': 5,'min_samples_leaf': 4, 'min_samples_split': 2}

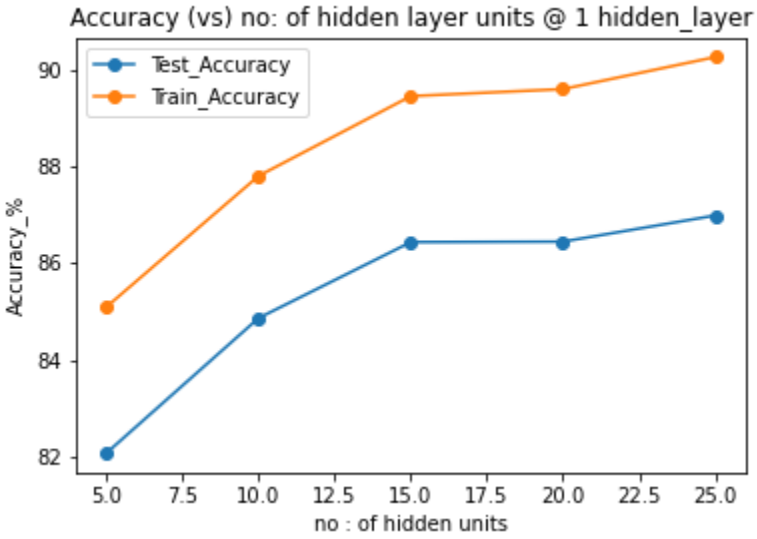
Accuracies for optimal_parameteres

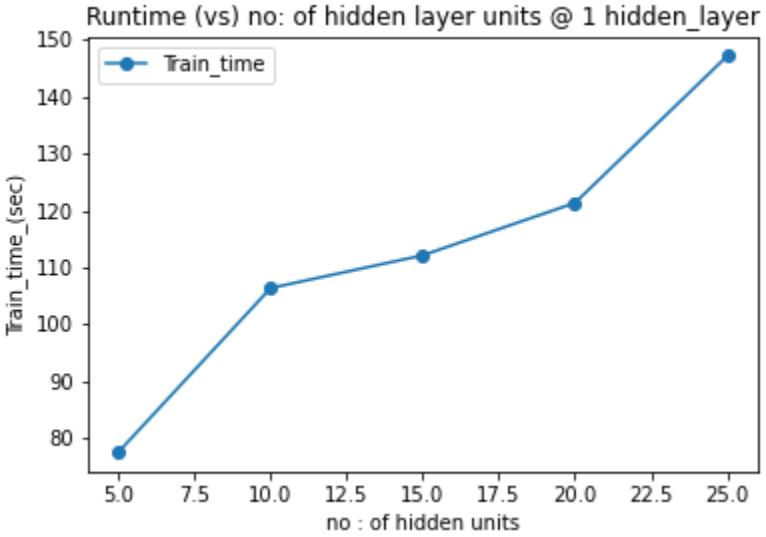
Trainin_accuracy	Test_accuracy	Validation_accuray
32.97	32.90	33.04

C,d,e,f,g- written complete code but executing is taking long time.

Neural Networks

b)





Confusion matrix for test set for each parameter

Hidden layer units = 5

```
[[790   7  44  37   5   4  94   0  18   1]
 [ 11 944  13  24   7   0   1   0   0   0]
 [ 27   1 768   5 119   1  67   0  12   0]
 [ 47  17  21 806  32   0  62   0  15   0]
 [  4   3 139  25 719   0  99   0  11   0]
 [  0   0   0   1   1 909   0  41   2  46]
 [152   4 173  28 110   0 509   0  24   0]
 [  0   0   0   0   0  50   0 907   0  43]
 [  0   2  24  17  13   4   9   4 927   0]
 [  1   0   1   0   0  30   0  40   0 928]]
```

Hidden layer units = 10

```
[[835   1  14  44   6   0  82   1  16   1]
 [  4 952   9  28   4   0   2   0   1   0]
 [ 18   3 782  13  98   2  78   0   6   0]
 [ 38  10  18 858  36   1  34   1   4   0]
 [  2   2 147  32 730   2  76   0   9   0]
 [  0   0   0   0   0 920   1  47   8  24]
 [155   1 131  38  71   0 584   0  20   0]
 [  0   0   0   0   0  40   0 934   0  26]
 [  3   1   7   8   2   5  21   4 948   1]
 [  0   0   0   0   0  16   0  39   2 943]]
```

Hidden layer units = 15

```
[[834   2  17  35   5   2  92   0  13   0]
 [  4 952   6  26   6   0   5   0   1   0]
 [ 20   3 781  11 101   1  78   0   5   0]
 [ 30  11  12 874  34   1  33   0   5   0]
 [  1   2 109  34 772   2  75   0   5   0]
 [  0   0   1   0   0 939   0  35   5  20]
 [141   1  99  36  74   0 630   0  19   0]
 [  0   0   0   0   0  28   0 947   0  25]
 [  2   0   3   4   7   3  13   5 963   0]
 [  0   0   0   0   0   8   0  40   1 951]]
```

Hidden layer units = 20

```
[[836   2  12  32   1   0 103   0  14   0]
 [  3 961   4  23   5   0   3   0   1   0]
 [ 12   1 805  11  90   1  72   0   8   0]
 [ 37  11  18 868  30   0  32   0   4   0]
 [  2   2 129  31 750   0  79   0   7   0]
 [  1   0   1   0   0 933   0  38   3  24]
 [133   0 103  28  74   0 640   0  22   0]
 [  0   0   0   0   0  29   0 944   0  27]
 [  2   1   7   9   4   2  12   5 958   0]
 [  0   0   0   0   0   9   1  41   0 949]]
```

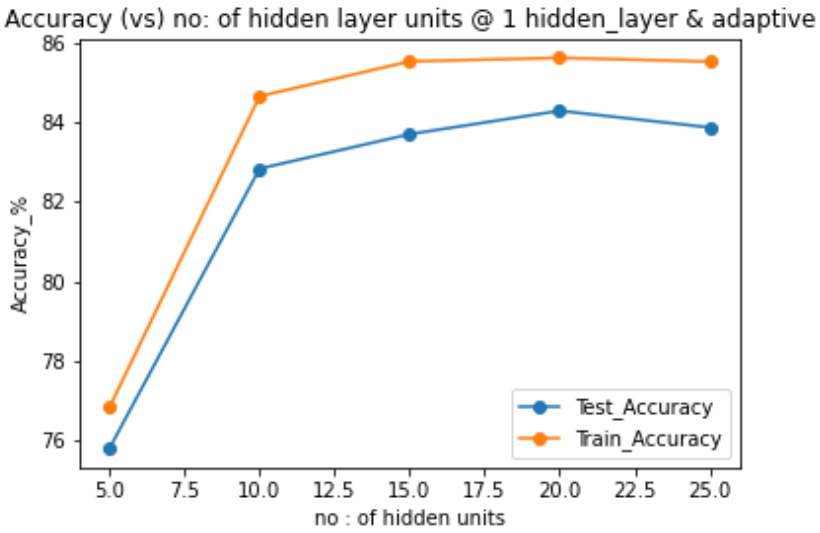
Hidden layer units = 25

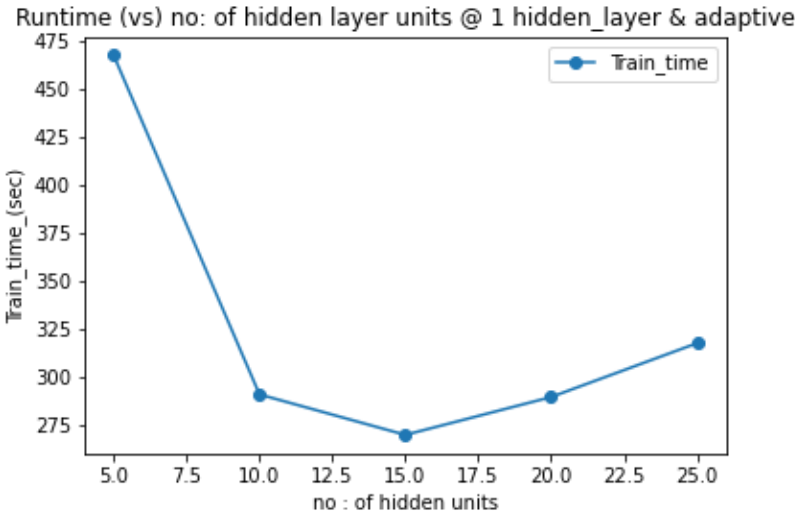
```
[[835   3   9  30   4   2 104   0  13   0]
 [  6 954   3  27   4   0   4   0   2   0]
 [ 17   1 784  13  93   1  85   0   6   0]
 [ 29  12  15 877  26   1  35   0   5   0]
 [  0   1  88  36 772   0  96   0   7   0]
 [  0   0   0   1   0 938   0  41   2  18]
 [133   1  88  30  66   0 666   0  16   0]
 [  0   0   0   0   0  27   0 953   0  20]
 [  1   1   4   7   3   2  15   4 963   0]
 [  0   0   0   0   0   7   0  36   1 956]]
```

Stopping criteria:If the validation loss is decreasing less than threshold for 5 times for sigmoid and 15 times for ReLU

c)

Adaptive Learning Rate





Confusion matrix for test set for each parameter

Hidden layer units = 5

```
[[768  15  30  91   5   2  71   0  18   0]
 [ 10 937  11  36   2   0   2   0   2   0]
 [ 24   1 806  12  28   1 117   0  11   0]
 [ 25  24  50 842  20   0  32   0   7   0]
 [  7   2 692  74  63   0 152   0  10   0]
 [  0   0   0   1   0 889   1  60   3  46]
 [185   5 209  69  21   0 473   0  38   0]
 [  0   0   0   0   0  31   0 926   1  42]
 [  2   0   8   7   3   3  39   4 933   1]
 [  0   0   0   1   0  14   1  42   0 942]]
```

Hidden layer units = 10

```
[[803   5  24  54   5   1  88   2  18   0]
 [  8 944  10  29   6   0   2   0   1   0]
 [ 23   2 725  10 140   1  88   0  11   0]
 [ 35  18  15 857  36   1  35   0   3   0]
 [  4   3 101  43 764   0  78   0   7   0]
 [  0   0   0   2   0 901   0  56   7  34]
 [180   2 141  45 123   1 477   0  31   0]
 [  0   0   0   0   0  35   0 919   0  46]
 [  1   1  10  12   3   3  16   5 949   0]
 [  0   0   0   0   0  12   0  42   1 945]]
```

Hidden layer units = 15

```
[[845   1  15  49   5   3  63   1  18   0]
 [  7 940  13  30   8   0   1   0   1   0]
 [ 16   0 739  11 157   1  66   0  10   0]
 [ 33  15  15 856  39   0  38   0   4   0]
 [  0   0  96  33 781   0  83   0   7   0]
 [  0   0   0   1   0 904   0  57   9  29]
 [205   3 133  42  99   2 487   0  29   0]
 [  0   0   0   0   0  35   0 923   0  42]
 [  2   1  12   8   4   3  10   6 954   0]
 [  0   0   0   0   0  12   0  45   1 942]]
```

Hidden layer units = 20

```
[[836   4  16  41   7   1  77   0  17   1]
 [  4 949  10  28   5   0   3   0   1   0]
 [ 19   1 758  11 133   2  64   0  12   0]
 [ 26  13  14 869  40   1  33   0   4   0]
 [  0   1 107  35 767   1  82   0   7   0]
 [  0   0   0   1   0 905   0  51   8  35]
 [162   3 132  41 103   2 530   0  27   0]
 [  0   0   0   0   0  34   0 924   0  42]
 [  2   2  10   6   6   3  16   5 950   0]
 [  0   0   0   0   0  14   0  43   1 942]]
```

Hidden layer units = 25

```
[[835   4  12  40   6   4  81   0  18   0]
 [  6 951   7  26   6   0   2   0   2   0]
 [ 17   3 740  11 144   2  72   0  11   0]
 [ 33  15  13 855  43   1  36   0   4   0]
 [  0   2 102  39 773   0  78   0   6   0]
 [  0   0   0   2   0 902   0  56   7  33]
 [180   2 133  38  98   1 516   0  32   0]
 [  0   0   0   0   0  36   0 918   0  46]
 [  2   1  10   7   4   4  14   5 953   0]
 [  0   0   0   0   0  12   0  42   1 945]]
```

d)

Used adaptive learning rate , 100x100 hidden layer

Accuracies

Activation	Training Accuracy	Test Accuracy
Sigmoid	85.39	83.78
ReLU	86.76	84.24

Confusion matrix for Sigmoid

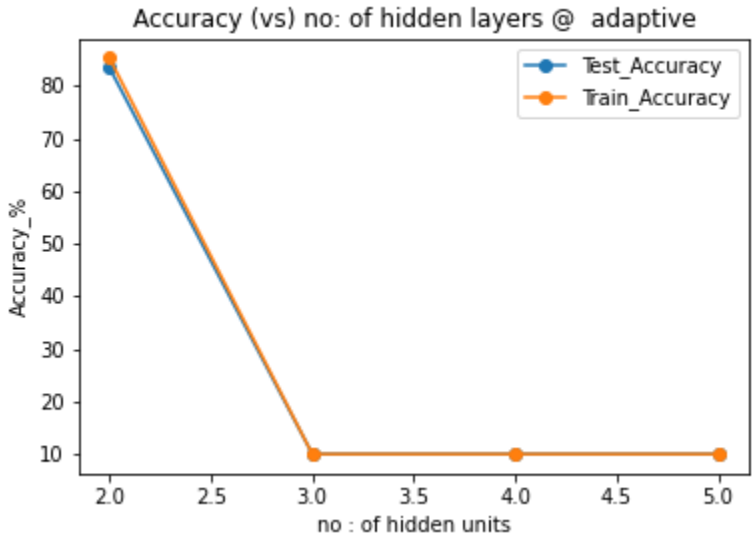
```
[[826   3  13  51   4   5  81   0  17   0]
 [  7 938  14  31   8   0   1   0   1   0]
 [ 16   1 747  11 134   1  82   0   8   0]
 [ 36  12  14 843  39   2  50   0   3   1]
 [  1   0 104  32 765   0  91   0   7   0]
 [  0   0   0   1   0 908   0  53   7  31]
 [162   2 123  45 102   1 535   0  30   0]
 [  0   0   0   0   0  34   0 923   0  43]
 [  0   1   7   9   2   6  21   6 948   0]
 [  0   0   0   0   0  15   0  39   1 945]]
```

Confusion matrix for ReLU

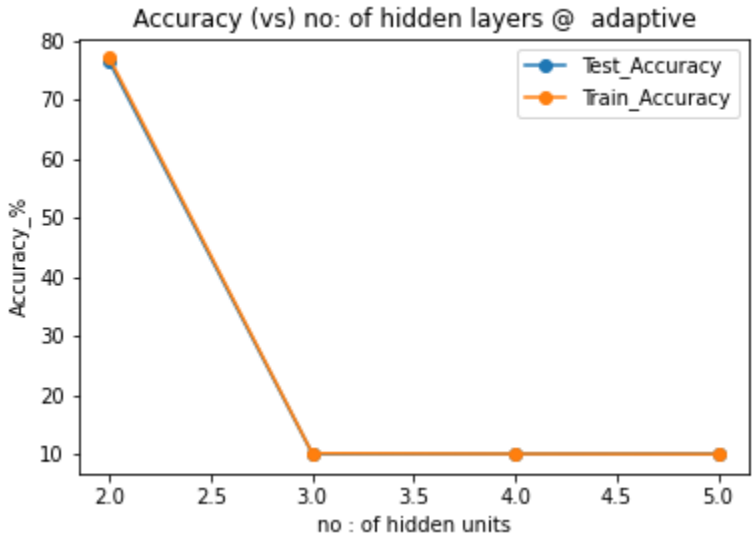
```
[[836   4  16  41   7   1  77   0  17   1]
 [  4 949  10  28   5   0   3   0   1   0]
 [ 19   1 758  11 133   2  64   0  12   0]
 [ 26  13  14 869  40   1  33   0   4   0]
 [  0   1 107  35 767   1  82   0   7   0]
 [  0   0   0   1   0 905   0  51   8  35]
 [162   3 132  41 103   2 530   0  27   0]
 [  0   0   0   0   0  34   0 924   0  42]
 [  2   2  10   6   6   3  16   5 950   0]
 [  0   0   0   0   0  14   0  43   1 942]]
```

e)

ReLU



Sigmoid



Best Arhcitecture from above analysis : based on max test accuracy criteria is [50,50] hidden layer with ReLU.

f)

Trainin_accuracy	Test_accuracy
89.06	85.94

g)

Trainin_accuracy	Test_accuracy
85.62	81.33

In my case the performance of the scikit learn function is low interms of accuracies but the execution was fast while using it.

Libraries used during implementation of code:

Pandas,Numpy

time,

matplotlib.pyplot as plt

sklearn.neural_network import MLPClassifier

sklearn.metrics import confusion_matrix

sklearn.tree import DecisionTreeClassifier

sklearn.model_selection import GridSearchCV

sklearn.ensemble import RandomForestClassifier

sklearn import tree

Xgboost

TfidfVectorizer

Nltk.corpus - stopwords

String

Scipy.sparse - hstack,csr_matrix

lightgbm import LGBMClassifier