COL774 A3
RISHI SHAH
2019CS10394

Q1.

a) Stopping Criteria : If total data at node<5, then make the node a leaf. Or if labels of each data at node is same, make it a leaf.

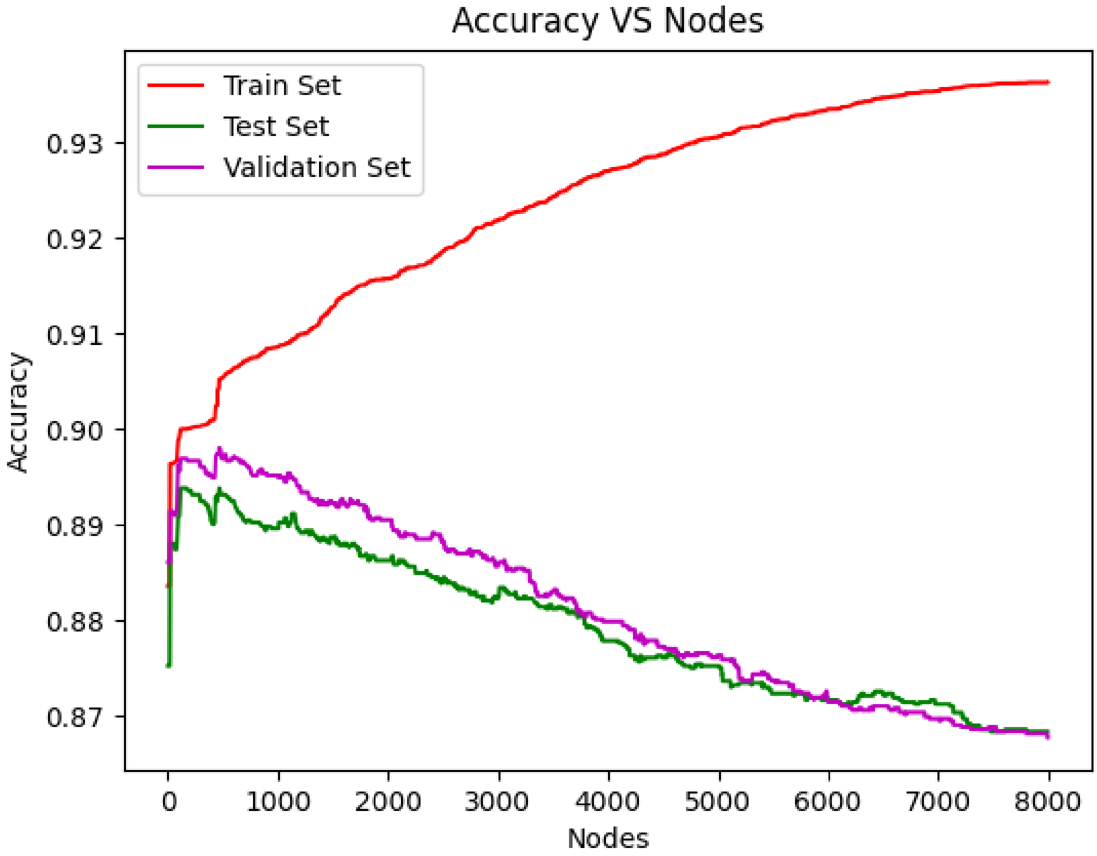
1. Multi-Way Split by Creating One Decision Tree Branch

Train Accuracy: 0.936

Test Accuracy: 0.868

Validation Accuracy: 0.866

Plot -

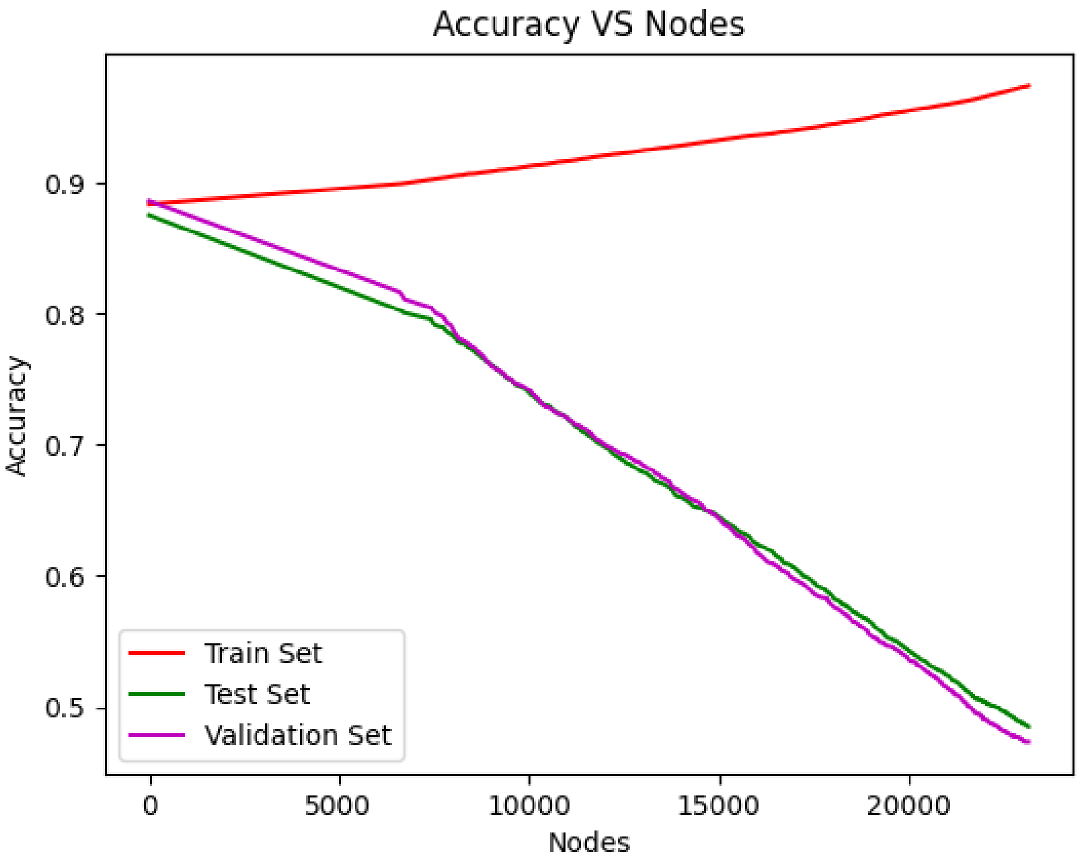


Total Nodes - 7995

Observations - As I change the convergence criteria the results vary. If I set that total data at node<10 then split, test and validation accuracy increases while the train decreases hence it shows that the model is overfitting currently.

Now, as the number of nodes increases the model is learning better on train set, however it is overfitting as validation and test accuracy decreases.

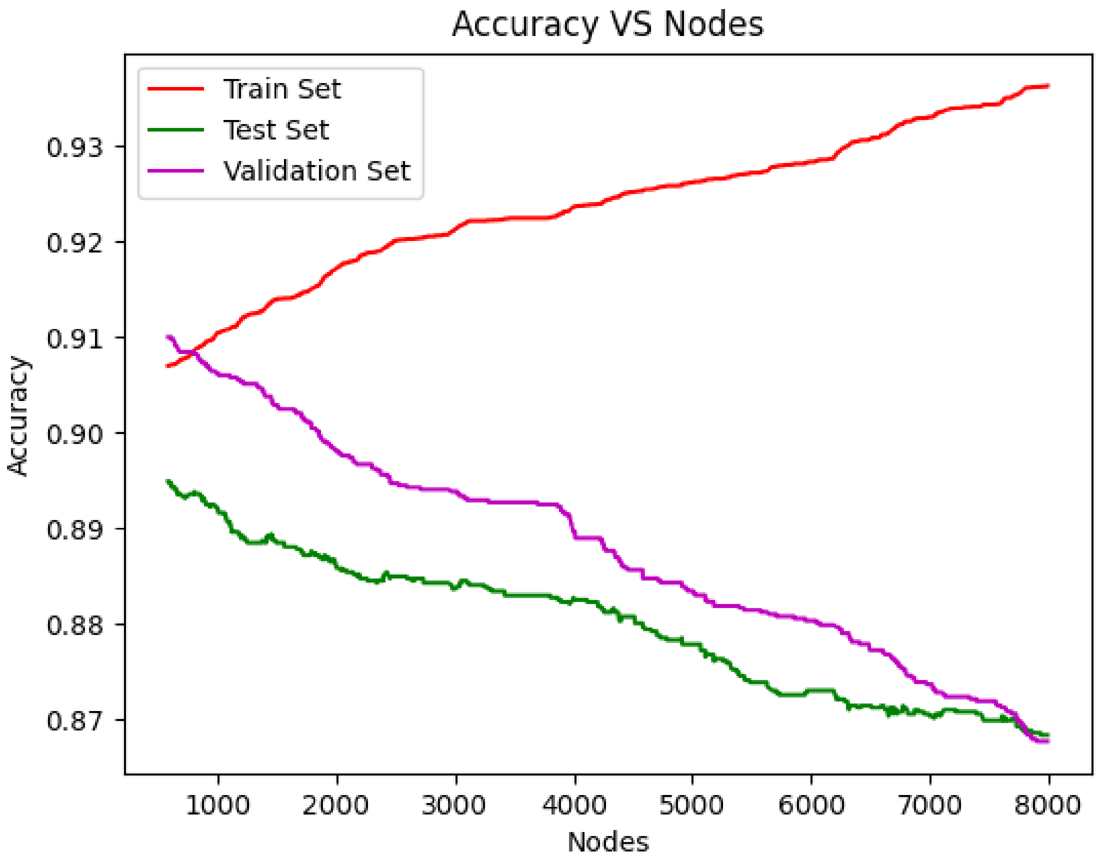
2. Two Way Split by First Converting the Attribute to a One-Hot Encoding
Train Accuracy: 0.974
Test Accuracy: 0.48
Validation Accuracy: 0.473
Plot -



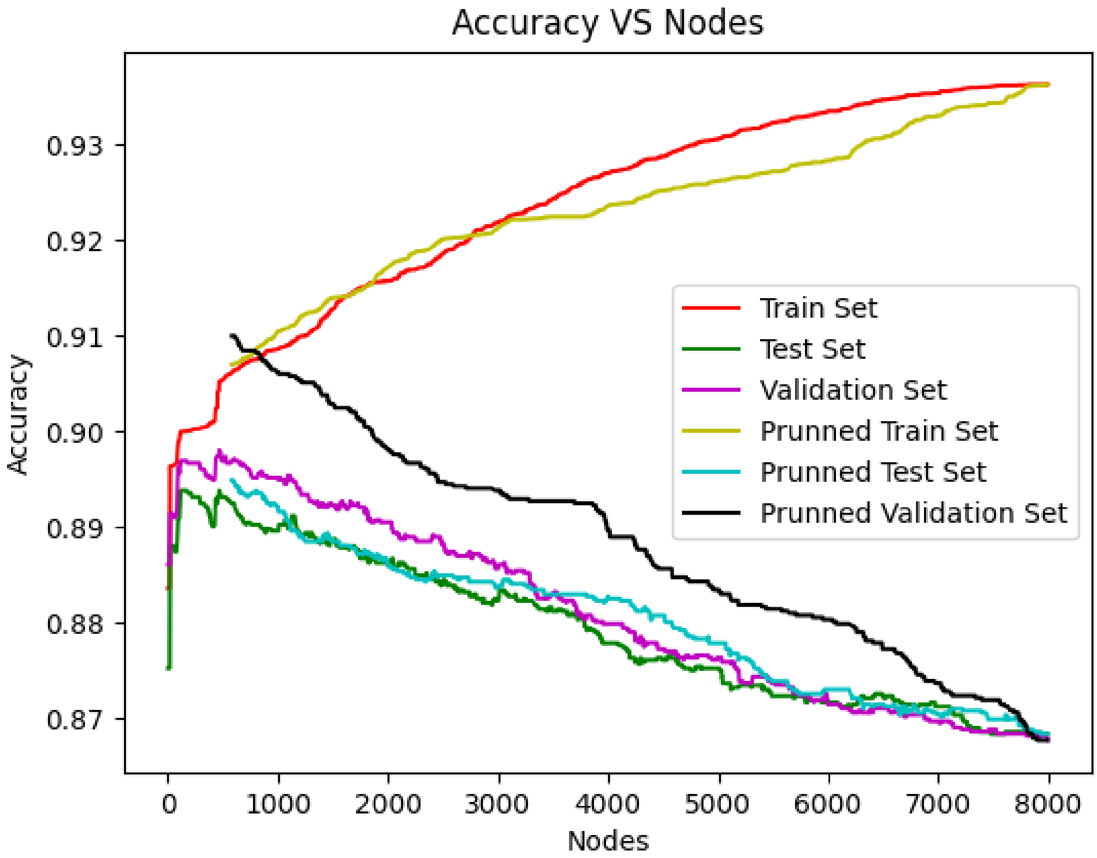
Total Nodes: 23160
Observations - In the encoded data, the number of features are more. So there is more chance of the model overfitting to the train set. Hence the train set accuracy is very large however validation and test set accuracy decrease significantly.

b) Method for Pruning: Compare the validation accuracy if the current node is kept as the leaf, with the actual leaves of the tree. If it is better at the current node then prun.
Train Accuracy: 0.907

Test Accuracy: 0.894
Validation Accuracy: 0.9099
Plot(only pruning, post growing tree) -



Note that the graph is from left to right, as the total nodes decrease.
Plot(both grow+prun):



Total Nodes: 580

Observations - The behavior of the graphs are as expected. As we prun the nodes, we try to reduce the overfitting caused by the decision tree. Now as seen in the graph, train set accuracy decreases as nodes decreases . While the test and validation set increases as node decreases. The model is able to achieve close to 0.9 accuracy in test and validation so it is fairly good. However, note that the data is skewed in the way that a lot of labels are marked 0, hence learning is very less by the model.

c) Best Parameters :

N_Estimators- 450

Max Featuers- 0.9

Min_Samples_Split- 10

Accuracy :

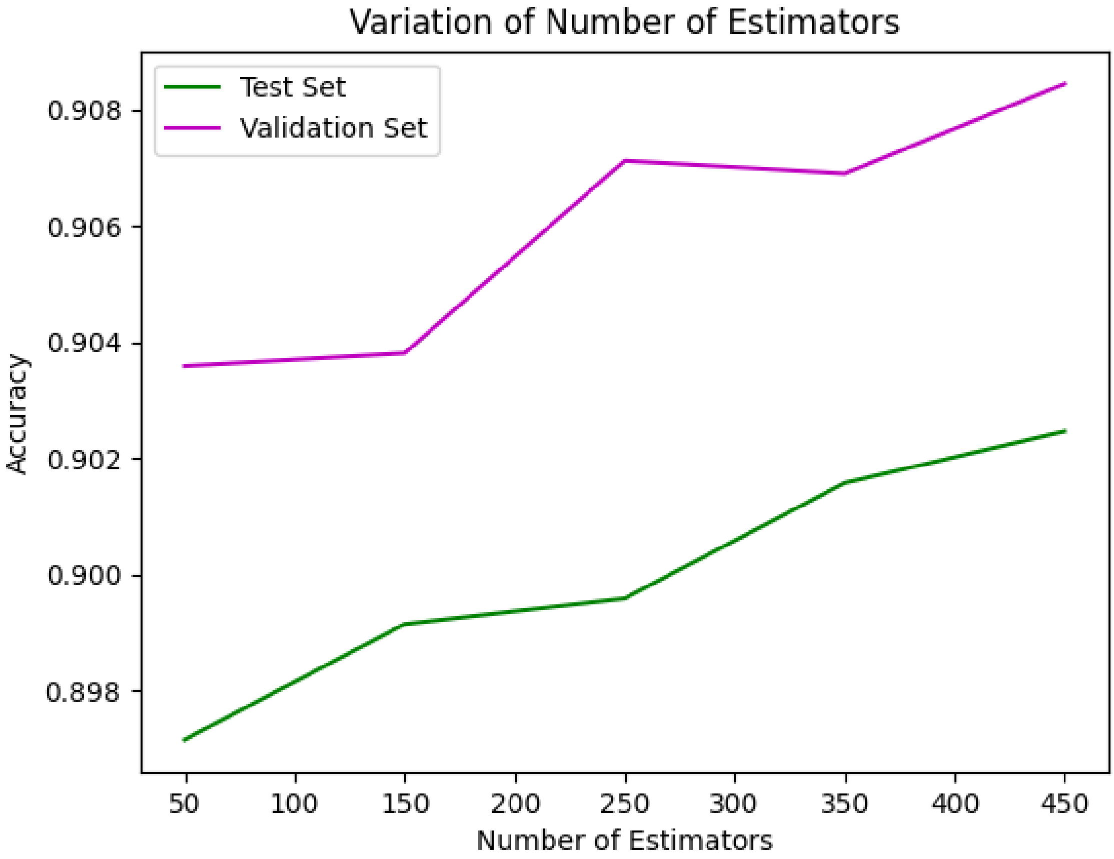
Training - 0.98

Test - 0.90

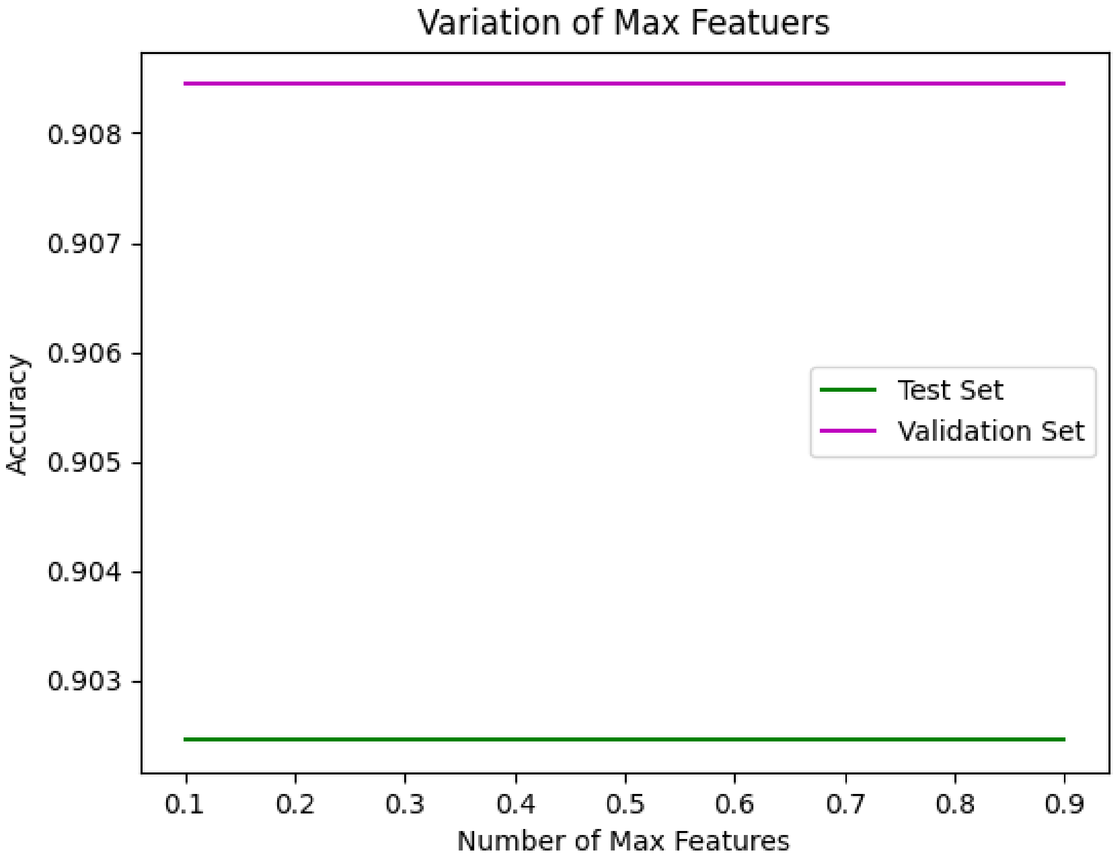
Validation 0.90

Comparison : By comparing the accuracy with pruning accuracy, we see that training accuracy is high in case of sklearn model. However both validation and test set accuracy are very close to model training post pruning(both 0.9). Hence if we consider sklearn to be a good model, our model after pruning performs very well on test and validation data set.

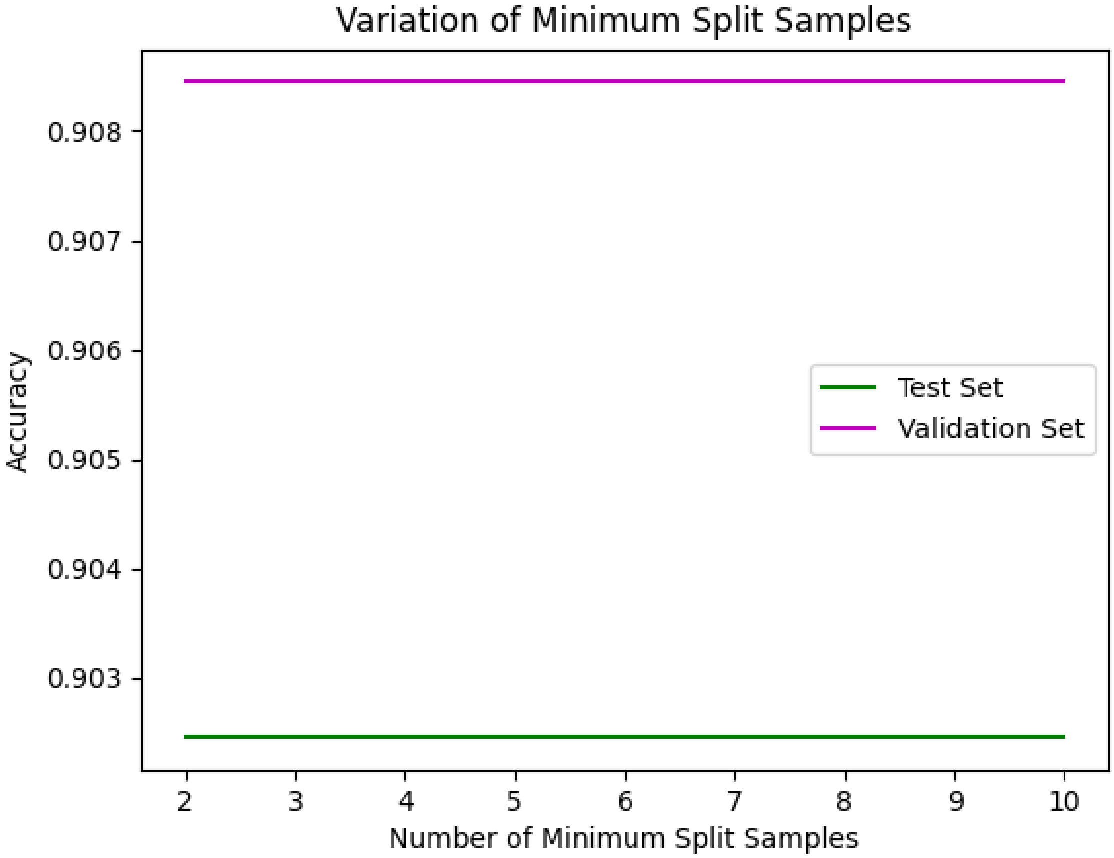
d) Plot for varying number of estimators -



Plot for varying maximum features -



Plot for varying minimum sample split -



Hence, the number of estimators is the most sensitive paramter.

Q2.

a) One hot encoding was used to get 85 features.
b) Implemented SGD for any given input parameters.
c) Stopping Criteria :

    Avg loss of previous epoch - Avg loss of next epoch < Delta

    Delta was chosen as 0.001. Mini-batch size was 100 and learning rate 0.1. Initialing the W(theta) was critical. I tried random initialsing from numpy(0.01*np.random.rand(shape), but the results were not at all satisfying. Model was predicting 0 all the time and was not deviating from it. However difference between 0 and 1 prediction was very low, so I went for different initialsation. I have used uniform initialisation(commonly known as He).

    Activation Unit: Sigmoid
    Hidden Layer Units :
    5,10,15,20,25:

Train Accuracy: 0.4995(almost same in all)

Test Accuracy: 0.501(almost same in all)

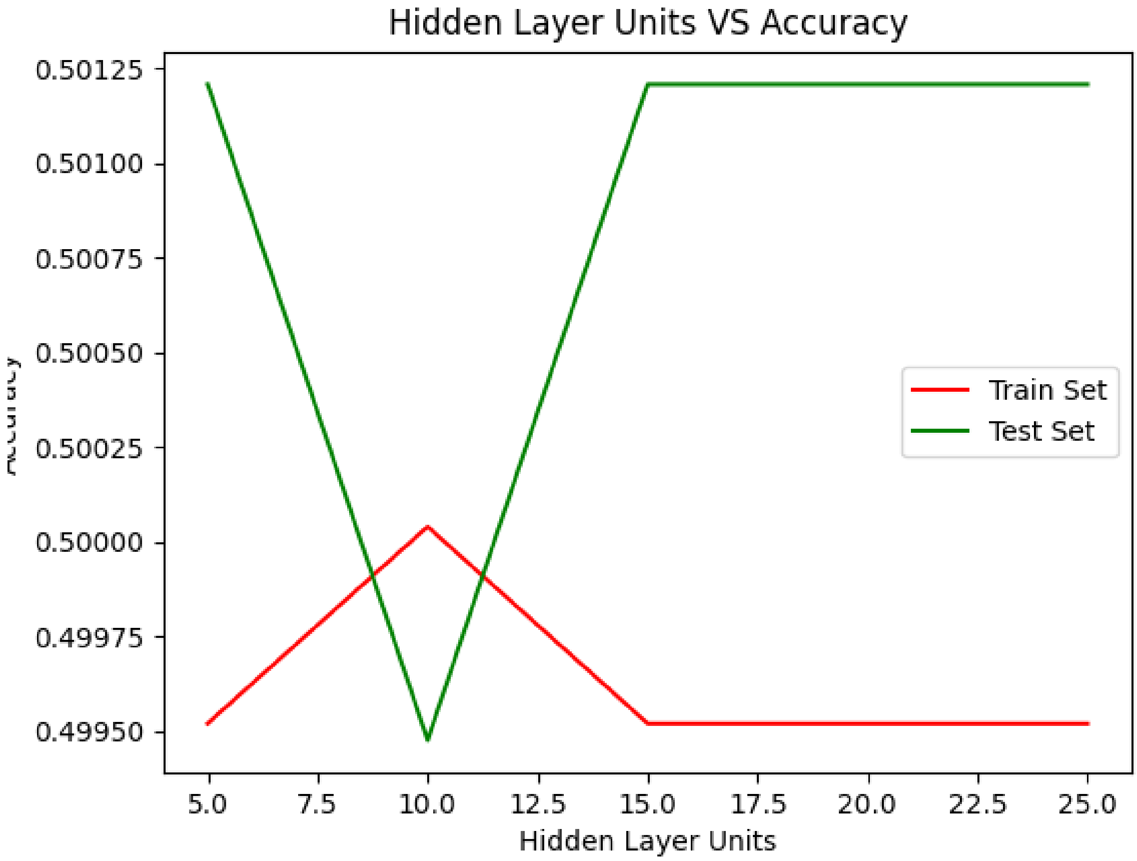
Time Taken(sec) : 5.47,8.06,6.05,1.31,1.47

Epochs : 92,117,95,18,19

Confusion Matrix :

```
[[501209 422498  47622  21121   3885   1996   1424    230     12      3]
 [     0      0      0      0      0      0      0      0      0      0]
 [     0      0      0      0      0      0      0      0      0      0]
 [     0      0      0      0      0      0      0      0      0      0]
 [     0      0      0      0      0      0      0      0      0      0]
 [     0      0      0      0      0      0      0      0      0      0]
 [     0      0      0      0      0      0      0      0      0      0]
 [     0      0      0      0      0      0      0      0      0      0]
 [     0      0      0      0      0      0      0      0      0      0]
 [     0      0      0      0      0      0      0      0      0      0]]
```

Plot:



Observations: The accuracy of train and test set is nearly same even if we change the number of hidden units in case of sigmoid. Almost all of the predictions are 0 as seen from the confusion matrix. However, as number of hidden layers increase the number of epochs to converge is less and less time is taken for the model to converge.

As results were not satisfying, I tried keeping the activation function for hidden layer as 'Relu'.

Activation Unit : 'Relu'

1. 5 :
   Train Accuracy: 0.49
   Test Accuracy: 0.46
   Time Taken : 1.13 sec
   Epochs : 21
   Confusion Matrix :

```
[[217371 175356  19050   8075   1664   1078    512     68      7      2]
 [283838 247142  28572  13046   2221    918    912    162      5      1]
 [     0      0      0      0      0      0      0      0      0      0]
 [     0      0      0      0      0      0      0      0      0      0]
 [     0      0      0      0      0      0      0      0      0      0]
 [     0      0      0      0      0      0      0      0      0      0]
 [     0      0      0      0      0      0      0      0      0      0]
 [     0      0      0      0      0      0      0      0      0      0]
 [     0      0      0      0      0      0      0      0      0      0]
 [     0      0      0      0      0      0      0      0      0      0]]
```

2. 10:
   Train Accuracy : 0.66
   Test Accuracy: 0.65
   Time Taken : 5.88 sec
   Epochs : 110
   Confusion Matrix :

```
[[426508 195164  10557   4866   2712   1696    128     15      8      1]
 [ 74701 227328  37049  16255   1173    300   1292    214      4      2]
 [     0      6     16      0      0      0      4      1      0      0]
[     0      0      0      0      0      0      0      0      0      0]
[     0      0      0      0      0      0      0      0      0      0]
[     0      0      0      0      0      0      0      0      0      0]
[     0      0      0      0      0      0      0      0      0      0]
[     0      0      0      0      0      0      0      0      0      0]
[     0      0      0      0      0      0      0      0      0      0]
[     0      0      0      0      0      0      0      0      0      0]]
```

3. 15:

   Train Accuracy : 0.79

   Test Accuracy : 0.78

   Time Taken : 6.67 sec

   Epochs : 105

   Confusion Matrix :

   ```
   [[442268  82036   1386    532   2744   1761      7      0      8      2]
    [ 58941 340462  46236  20589   1141    235   1411    211      4      1]
    [     0      0      0      0      0      0      0      0      0      0]
    [     0      0      0      0      0      0      6     19      0      0]
    [     0      0      0      0      0      0      0      0      0      0]
    [     0      0      0      0      0      0      0      0      0      0]
    [     0      0      0      0      0      0      0      0      0      0]
    [     0      0      0      0      0      0      0      0      0      0]
    [     0      0      0      0      0      0      0      0      0      0]
    [     0      0      0      0      0      0      0      0      0      0]]
   ```

4. 20:

   Train Accuracy : 0.90

   Test Accuracy : 0.89

   Time Taken : 7.63

   Epochs : 116

   Confusion Matrix :

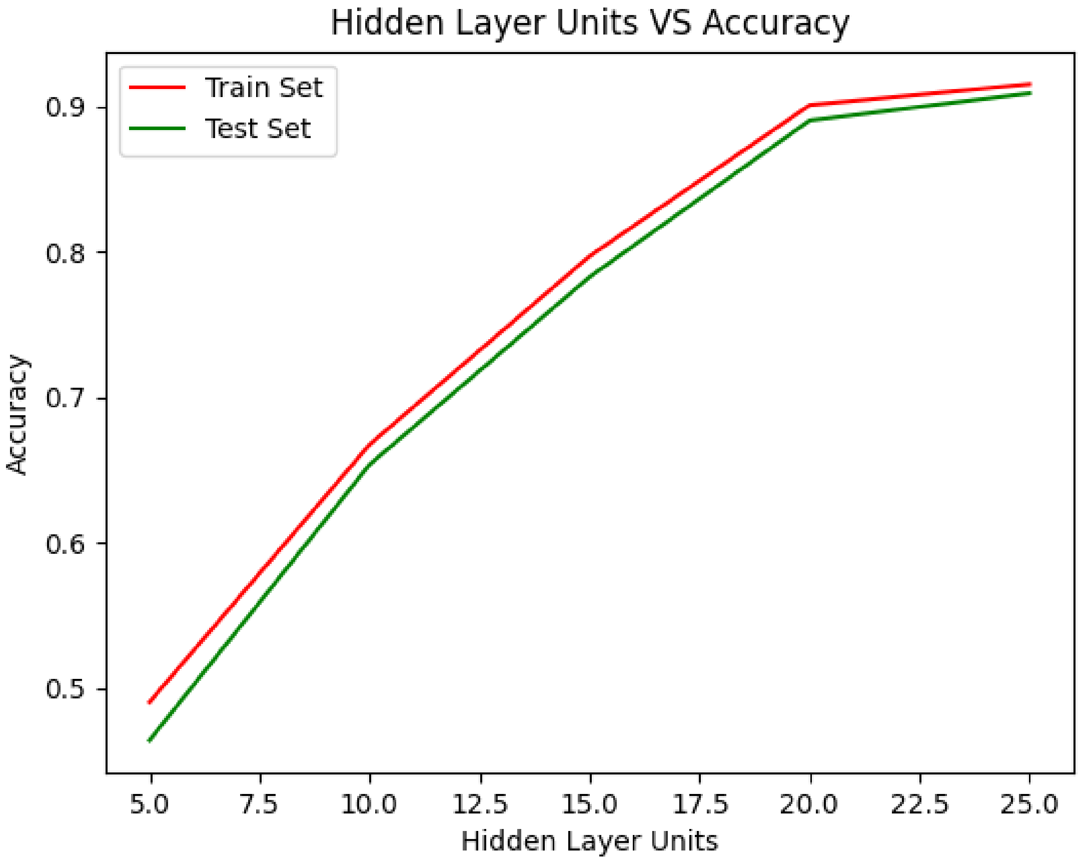
   ```
   [[481939  14189    137      0   3294   1883      0      0     11      3]
    [ 19270 408309  47485  21121    591    113   1424    230      1      0]
    [     0      0      0      0      0      0      0      0      0      0]
    [     0      0      0      0      0      0      0      0      0      0]
    [     0      0      0      0      0      0      0      0      0      0]
   ```

```
 [     0      0      0      0      0      0      0      0      0      0]
 [     0      0      0      0      0      0      0      0      0      0]
 [     0      0      0      0      0      0      0      0      0      0]
 [     0      0      0      0      0      0      0      0      0      0]
 [     0      0      0      0      0      0      0      0      0      0]]
```

5. 25:
   Train Accuracy : 0.915
   Test Accuracy : 0.908
   Time Taken : 15.01
   Epochs : 229
   Confusion Matrix :

```
[[496439  10057      0      0   3773   1970      0      0     11      3]
 [  4770 412320  47502  21061    112     26   1398    220      1      0]
 [     0    105    119     51      0      0     26     10      0      0]
 [     0     16      1      9      0      0      0      0      0      0]
 [     0      0      0      0      0      0      0      0      0      0]
 [     0      0      0      0      0      0      0      0      0      0]
 [     0      0      0      0      0      0      0      0      0      0]
 [     0      0      0      0      0      0      0      0      0      0]
 [     0      0      0      0      0      0      0      0      0      0]
 [     0      0      0      0      0      0      0      0      0      0]]
```

Plot:



Observations: In case of relu, the predictions have deviated from 0 as it does not allow underfitting(as it passes the Z through it by using max(0,Z) function). Also as the number of hidden units increases, both training and test accuracy increases. And model takes more epochs and more time to train. So the results are as expected, and model is performing better.

d) Learning Rate was kept as mentioned in the problem statment. Stopping Criteria was kept the same.

Activation unit: Sigmoid.

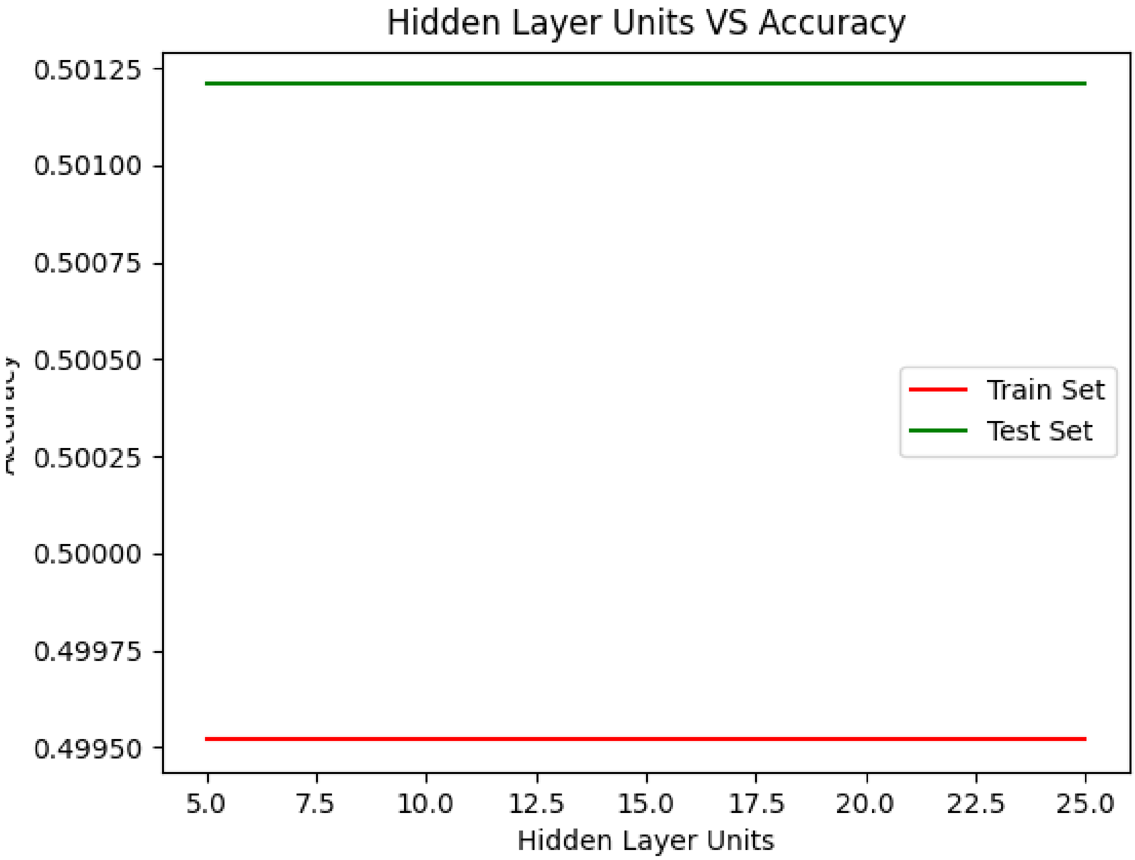
Train Accuracy : 0.4995

Test Accuracy : 0.5012

Time Taken : 1.71,4.79,2.60,0.71,41.53

Epochs : 21,73,56,14,20

Plot :



```
Confusion Matrix :
[[501209 422498  47622  21121   3885   1996   1424    230     12      3]
 [     0      0      0      0      0      0      0      0      0      0]
 [     0      0      0      0      0      0      0      0      0      0]
 [     0      0      0      0      0      0      0      0      0      0]
 [     0      0      0      0      0      0      0      0      0      0]
 [     0      0      0      0      0      0      0      0      0      0]
 [     0      0      0      0      0      0      0      0      0      0]
 [     0      0      0      0      0      0      0      0      0      0]
 [     0      0      0      0      0      0      0      0      0      0]
 [     0      0      0      0      0      0      0      0      0      0]]
```

Observations: By changing the learning rate to adaptive, there is not much difference in the train and test accuracy. However, the time taken and number of epochs have decreased significantly. The reason being that adaptive learning rate helps to converge faster(and loss is alsl fluctuating less in this case).

e) Activation Unit: Relu, Hidden Layer Unit: [100,100]

Without Adaptive-

Train Accuracy : 0.993

Test Accuracy : 0.990

Time Taken : 39.49 sec

Epochs : 217

Confusion Matrix :

```
[[501205      8      0      0   3863   1996      0      0     12      3]
 [     4 422368   1225      2     22      0      0      0      0      0]
 [     0    122  46352    726      0      0     21      0      0      0]
 [     0      0     45  20381      0      0    857    230      0      0]
 [     0      0      0      0      0      0      0      0      0      0]
 [     0      0      0      0      0      0      0      0      0      0]
 [     0      0      0     12      0      0    546      0      0      0]
 [     0      0      0      0      0      0      0      0      0      0]
 [     0      0      0      0      0      0      0      0      0      0]
 [     0      0      0      0      0      0      0      0      0      0]]
```

With Adaptive:

Train Accuracy : 0.95

Test Accuracy : 0.94

Time Taken : 73.86 sec

Epochs : 394

Confusion Matrix :

```
[[500962    507      0      0   3885   1996      0      0     12      3]
 [   247 421991  28751     80      0      0      6      0      0      0]
 [     0      0  18818  19297      0      0    863     12      0      0]
 [     0      0     53   1744      0      0    555    218      0      0]
 [     0      0      0      0      0      0      0      0      0      0]
 [     0      0      0      0      0      0      0      0      0      0]
 [     0      0      0      0      0      0      0      0      0      0]
 [     0      0      0      0      0      0      0      0      0      0]
 [     0      0      0      0      0      0      0      0      0      0]
 [     0      0      0      0      0      0      0      0      0      0]]
```

Comparison : This model is actually performing very well. In the case where learning rate is not adaptive train and test accuracy are very high.Earlier it was not deviating from 0, but now learning is better. So it is not overfitting nor underfitting. However, in case of adaptive, it is taking more epochs and still not converging enough, so as per me the adaptive criteria is making the learning slow and it needs to be such that learning can be bit fast. The non-adaptive model is infact working better than the MLP classifier(solved in next part), so that is a good result.

f) All activation units were set Relu. Other parameters :
Hidden Layer Size: (100,100)
Activation : Relu
Solver : sgd
Learning Rate Init: 0.01
Batch Size ; 100
Learning Rate: Adaptive
Max Iterations: 1000
Random State: 2
Shuffle: True

Train Accuracy: 0.9996
Test Accuracy: 0.9742
In terms of performance, it is working better than the adaptive learning rate in part (e). However, the model without adaptive learning rate in part (e) works better than this on test set(0.99 as compared to 0.97).