Machine Learning : COL 774

Assignment 3

Q1. Decision Tree (and Random Forest)

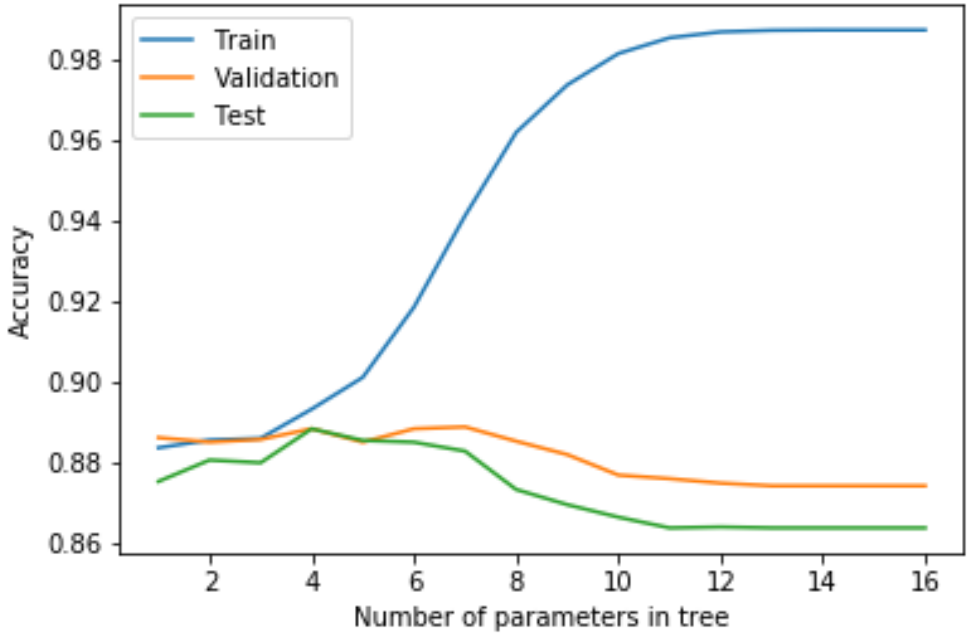
A.

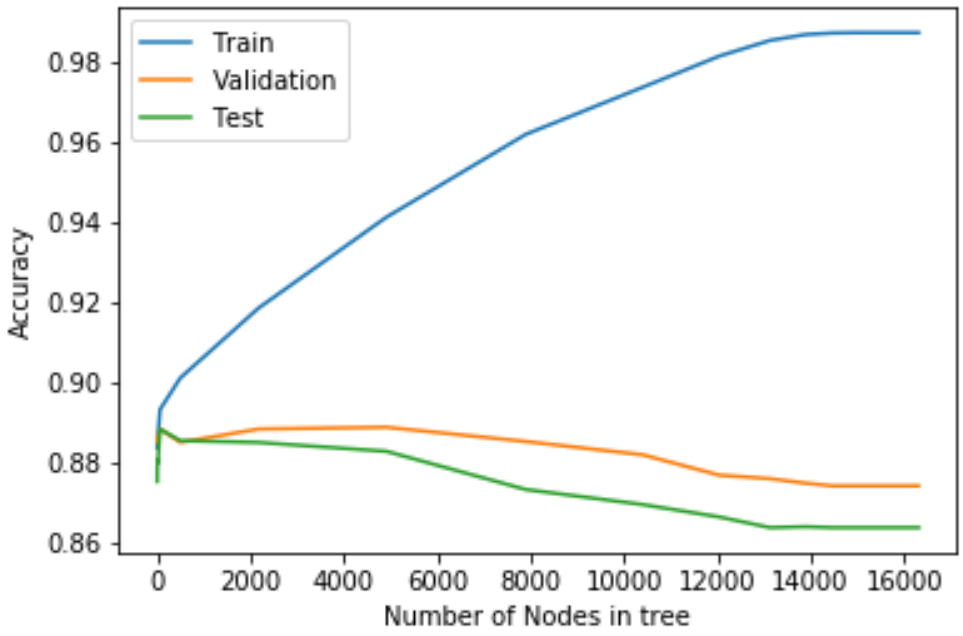
a) Multi-way Split decision tree

Training Accuracy 98.71%

Validation Accuracy 87.41%

Testing Accuracy 86.37 %





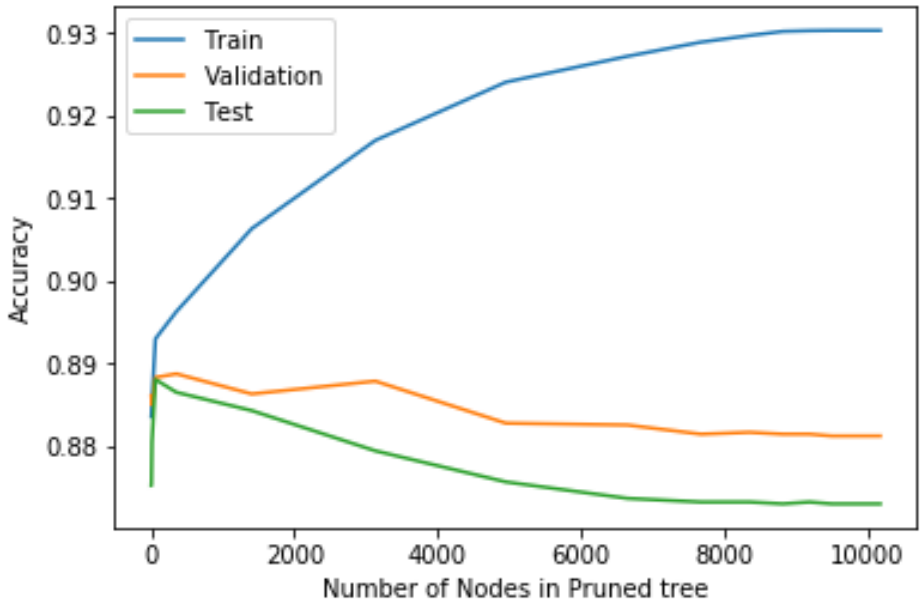
b) One-hot encoding decision tree

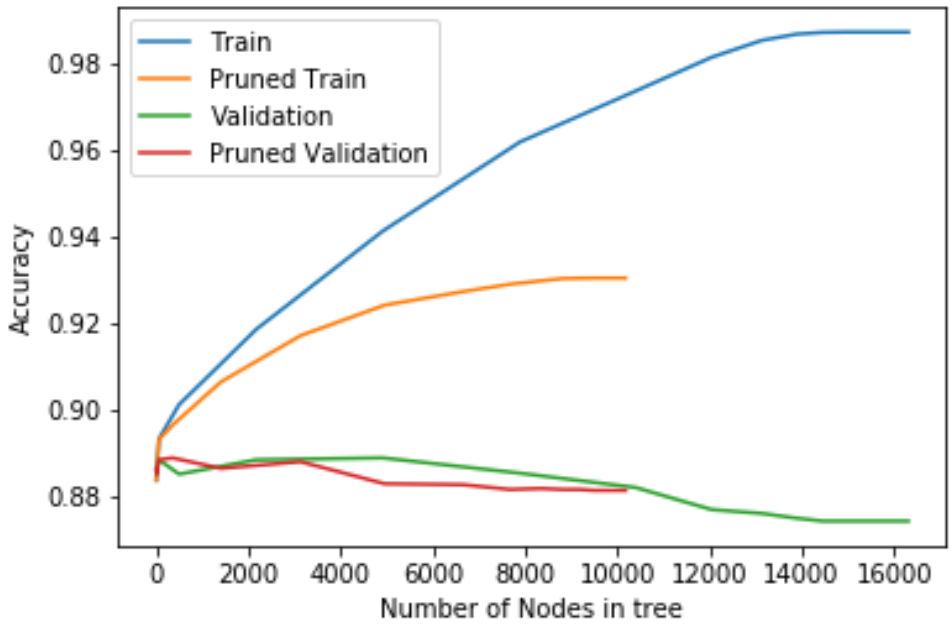
Training Accuracy 89.64%
Validation Accuracy 89.16%
Testing Accuracy 88.80 %

Observations:

- **Multi-way split decision tree** seems **overfitted** on the **training data,** while One-hot encoding tree doesn't much
- As number of nodes increasing in tree, the **Validation accuracy** is **dropping**

B. Post Pruning Decision Tree





Train Accuracy 92.60% (decreased)

Validation Accuracy 88.10% (increased)

Test Accuracy 87.48 % (Increased)

Comments:

a. As we see from the plots, in the **pruned** decision tree, number of nodes has decreased.
b. The **Validation** accuracy has **increased** as compared to the original tree (i.e. without pruning)
c. The **Training** accuracy has **decreased** significantly, which indicates **reduction in overfitting**. Hence, **pruning is successful.**

C. Random Forest

a. **Training accuracy 94.16%**
b. **Out-of-bag accuracy 89.84%**
c. **Validation accuracy 89.89%**
d. **Test accuracy 89.07%**

Optimal Parameters:

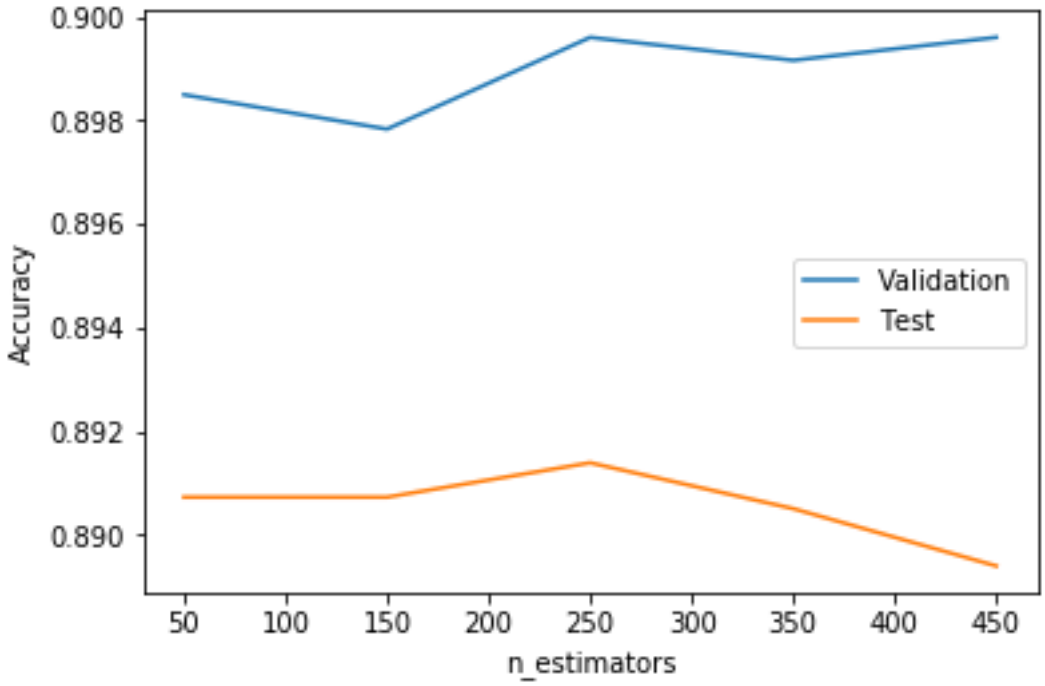
1. **n_estimators = 350**
2. **max_features = 0.3**
3. **min_samples_split = 10**

Observation:

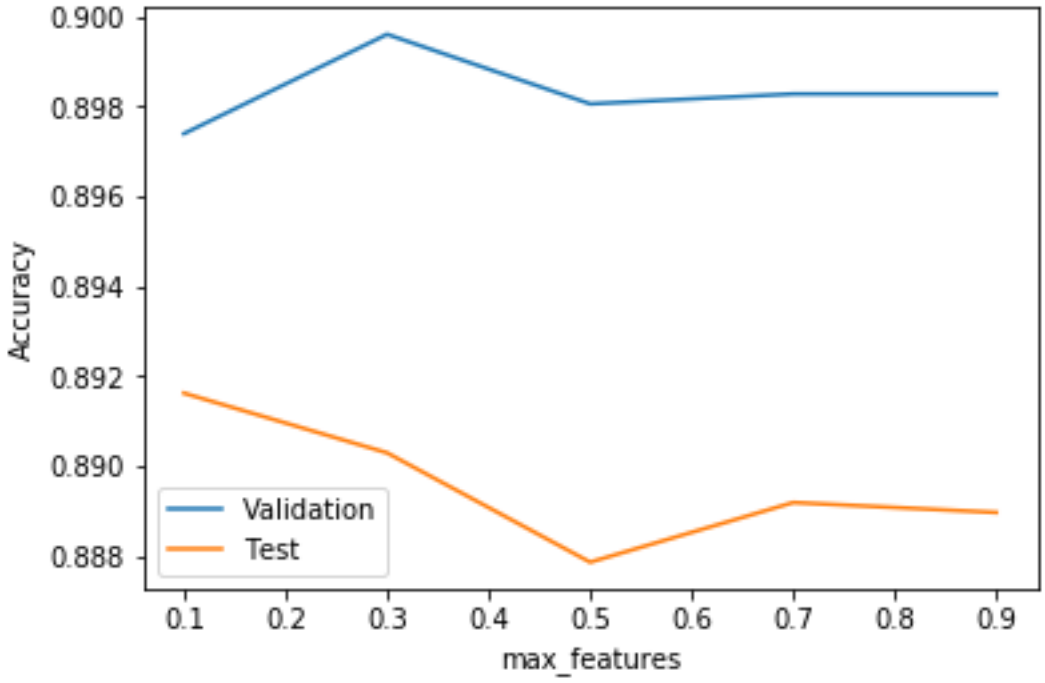
a. **Accuracies as compared to part b are near about similar.**
b. **But on part c, they are on slightly higher side**
c. **Time is faster for calculation**

D. Random Forest – Parameter Sensitivity Analysis

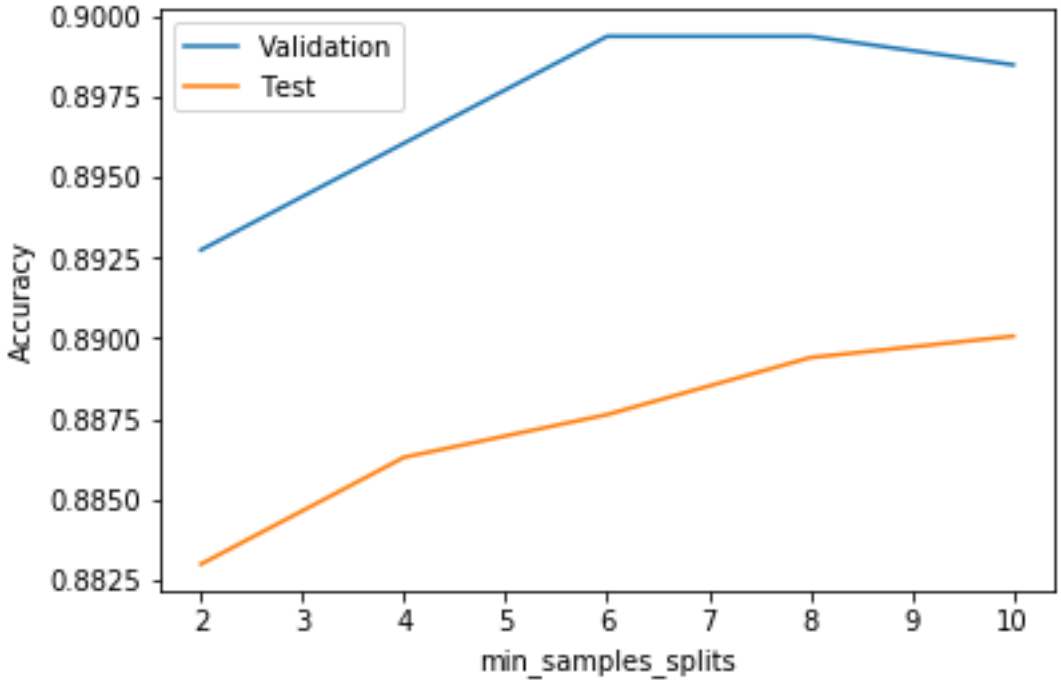
a. max_features = 0.3 and min_samples_split = 10



b. n_estimators = 350 and min_samples_split = 10



c. **max_features = 0.3 and n_estimators = 350**



Observations:

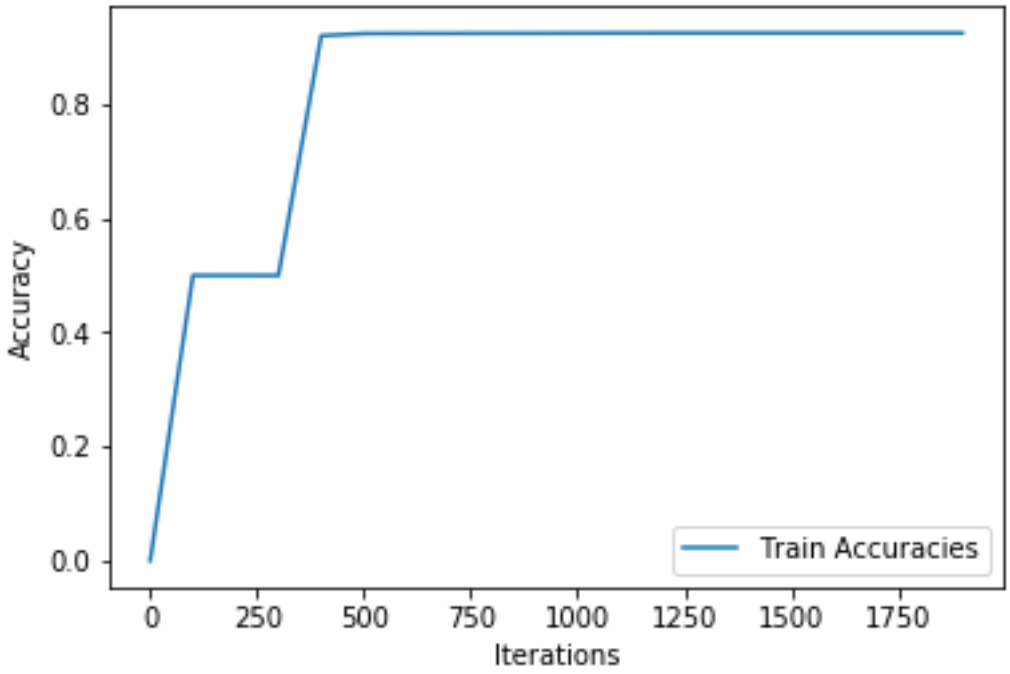
- **'min_sample_splits' is most sensitive as compared to the other two. It also increases as increase in accuracy**
- **'max_features' mediocre range gives best accuracy realtively.**
- **'n_estimators' as increases reduces test acuuracy but increases validation accuracy. Could be over-fitting over validation set.**

Q2. Neural Networks

A. **Converted data to one-hot encoding**
B. **Implemented customizable NN**

C. **Stoppage Criteria:**

Loss change should be **less than '0.000001'**
Got best accuracy at layer **25.**
Time taken to Train **~2 mins**
Train accuracy **92.27 %**
Test accuracy **91.15 %**



Confusion Matrix:

1. Train

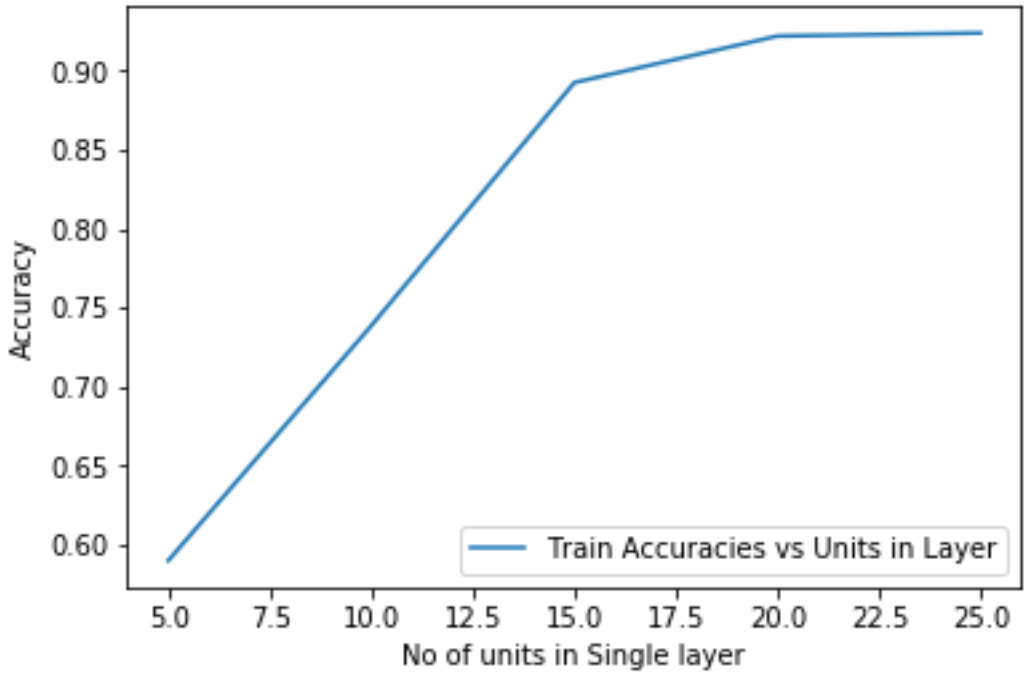
	0	1	2	3	4	5	6	7	8	9
0	10599	0	0	0	0	0	0	0	0	0
1	1206	0	0	0	0	0	0	0	0	0
2	452	0	0	0	0	6	0	0	1	54
3	13	0	0	18	0	0	0	0	0	62
4	12	0	0	0	0	0	0	0	0	42
5	36	0	0	0	0	0	0	0	0	0
6	2	0	0	0	0	0	0	0	0	4
7	3	0	0	0	0	0	0	0	0	2
8	1	0	0	0	0	0	0	0	0	4
9	0	0	0	0	0	0	0	0	0	12493

2. Test

	0	1	2	3	4	5	6	7	8	9
0	421449	0	0	79	0	14	0	0	0	956
1	47618	0	0	0	0	0	0	0	0	4
2	19327	0	0	1	0	73	0	0	18	1702
3	377	0	0	137	0	1	0	0	0	3370
4	27	0	0	0	0	0	0	0	0	1969
5	1424	0	0	0	0	0	0	0	0	0
6	117	0	0	0	0	0	0	0	1	112
7	2	0	0	1	0	0	0	0	0	9
8	1	0	0	0	0	0	0	0	0	2
9	1508	0	0	75	0	0	0	0	0	499626

For each value of Single layer units:

Accuracies varying with respect to the single layer units.



Observation:

As number of **units in single layer increases**, the **accuracy is increasing.**

D. Adaptive learning rate

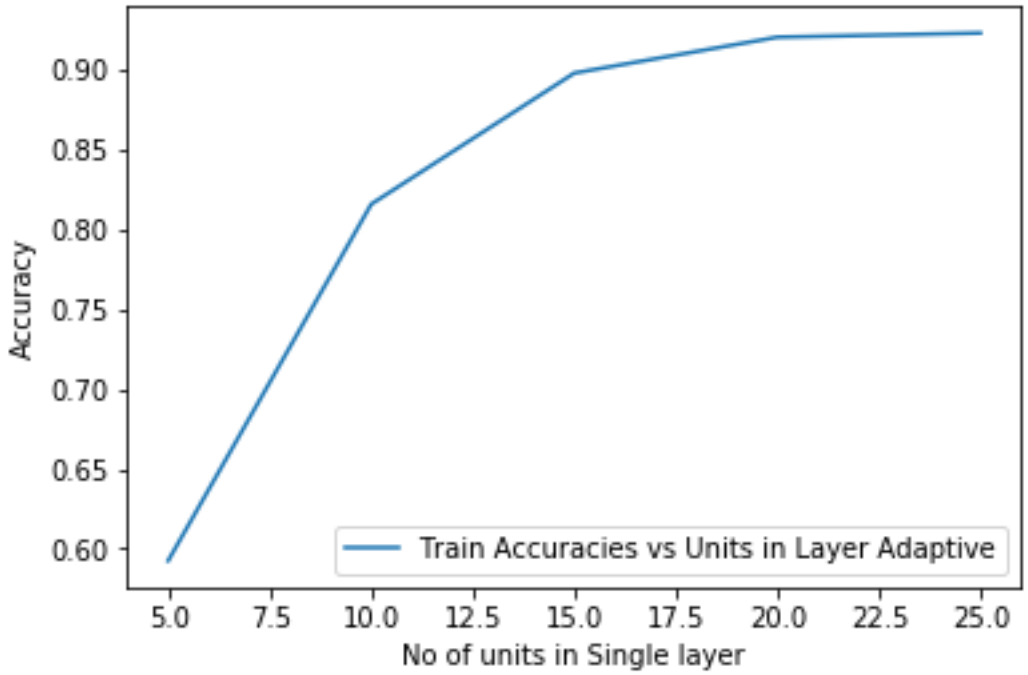
Stopping criteria is **'0.00001'**

Have to change stoppage criteria as loss curve is smoother.

Training accuracy **92.32%**

Test accuracy **91.96%**

Adaptive accuracies vs Number of units in single layer:



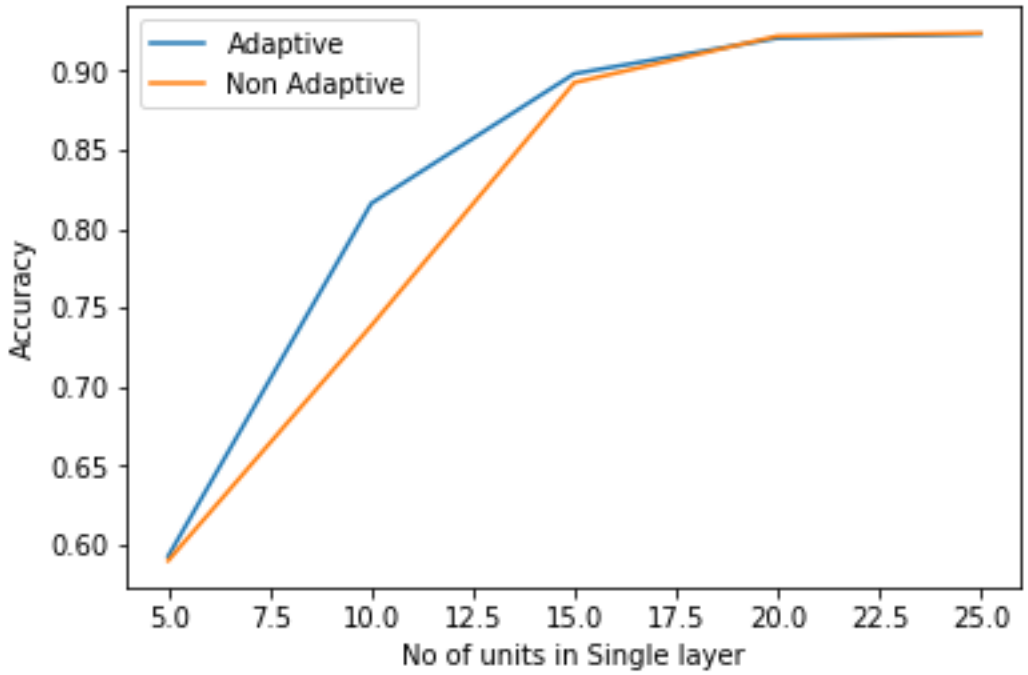
Confusion Matrix:

d. Train

	0	1	2	3	4	5	6	7	8	9
0	10599	0	0	0	0	0	0	0	0	0
1	1206	0	0	0	0	0	0	0	0	0
2	480	0	0	0	0	0	0	0	0	33
3	4	0	0	0	0	0	0	0	0	89
4	2	0	0	0	0	0	0	0	0	52
5	36	0	0	0	0	0	0	0	0	0
6	6	0	0	0	0	0	0	0	0	0
7	3	0	0	0	0	0	0	0	0	2
8	2	0	0	0	0	0	0	0	0	3
9	0	0	0	0	0	0	0	0	0	12493

e. Test

	0	1	2	3	4	5	6	7	8	9
0	422217	0	0	0	0	0	0	0	0	281
1	47622	0	0	0	0	0	0	0	0	0
2	20365	0	0	0	0	0	0	0	0	756
3	189	0	0	0	0	0	0	0	0	3696
4	14	0	0	0	0	0	0	0	0	1982
5	1424	0	0	0	0	0	0	0	0	0
6	160	0	0	0	0	0	0	0	0	70
7	0	0	0	0	0	0	0	0	0	12
8	1	0	0	0	0	0	0	0	0	2
9	255	0	0	0	0	0	0	0	0	500954



Comments:

a. As compared to the part **c**, the curve in this part is lot smoother.
b. Accuracy is also slightly increased for the layers with lesser number of units.
c. Time to train is also less, around **~2mins**

E. ReLU

Training accuracy **92.22 %**
Testing accuracy **89.22 %**
Confusion matrix:

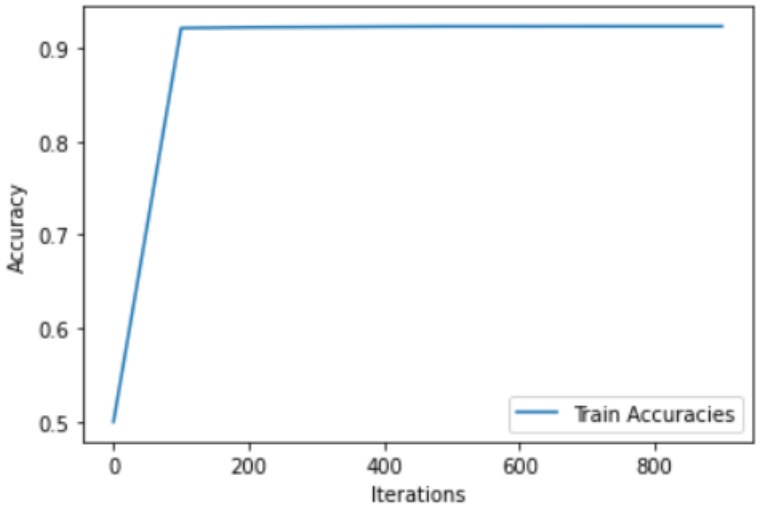
1. **Adaptive**
   a. Train

	0	1	2	3	4	5	6	7	8	9
0	10593	0	0	0	0	0	0	0	0	6
1	1206	0	0	0	0	0	0	0	0	0
2	513	0	0	0	0	0	0	0	0	0
3	33	0	0	0	0	1	0	0	0	59
4	8	0	0	0	0	0	0	0	0	46
5	36	0	0	0	0	0	0	0	0	0
6	6	0	0	0	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0	0	4
8	2	0	0	0	0	0	0	0	0	3
9	3	0	0	0	0	0	0	0	0	12490

b. Test

	0	1	2	3	4	5	6	7	8	9
0	407069	6	1	0	0	0	0	0	0	15422
1	47616	0	0	0	0	0	0	0	0	6
2	21121	0	0	0	0	0	0	0	0	0
3	604	0	0	0	0	0	0	0	1	3280
4	31	0	0	0	0	0	0	0	0	1965
5	1424	0	0	0	0	0	0	0	0	0
6	230	0	0	0	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0	0	11
8	2	0	0	0	0	0	0	0	0	1
9	16049	0	1	0	0	0	0	0	8	485151

Train accuracy plot vs iterations:



2. Non-Adaptive

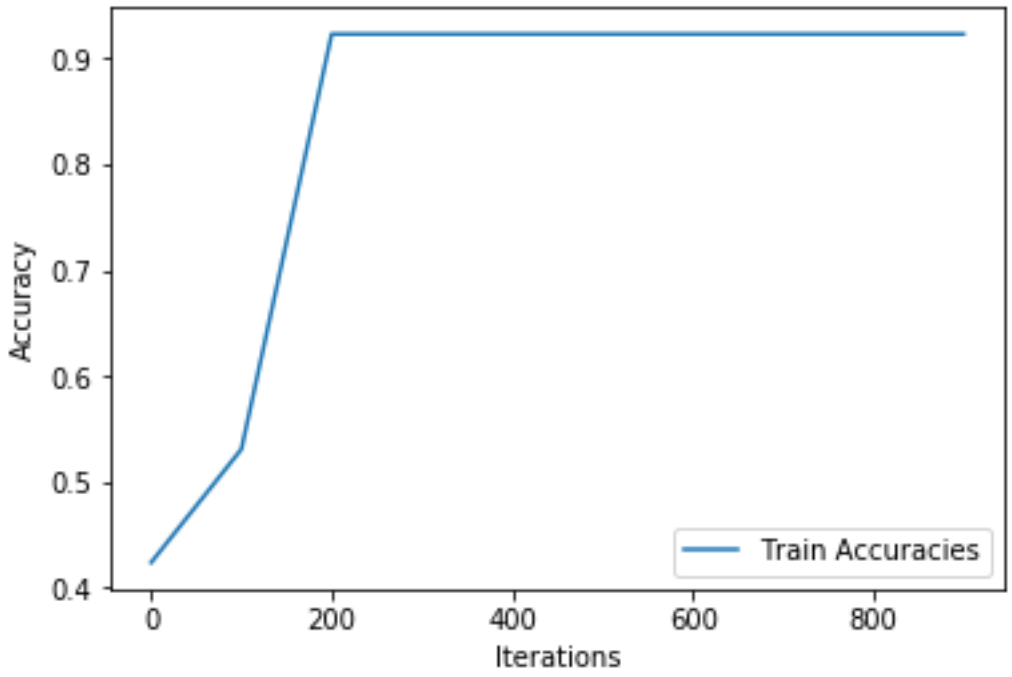
a. Train

	0	1	2	3	4	5	6	7	8	9
0	10599	0	0	0	0	0	0	0	0	0
1	1120	0	0	0	0	0	0	0	0	86
2	512	0	0	0	0	0	0	0	0	1
3	5	0	0	0	0	0	0	0	0	88
4	1	0	0	0	0	0	0	0	0	53
5	36	0	0	0	0	0	0	0	0	0
6	6	0	0	0	0	0	0	0	0	0
7	2	0	0	0	0	0	0	0	0	3
8	0	0	0	0	0	0	0	0	0	5
9	0	0	0	0	0	0	0	0	0	12493

b. Test

	0	1	2	3	4	5	6	7	8	9
0	418671	0	0	0	0	0	0	0	0	3827
1	45535	0	0	0	0	0	0	0	0	2087
2	21086	0	0	0	0	0	0	0	0	35
3	70	0	0	0	0	0	0	0	0	3815
4	4	0	0	0	0	0	0	0	0	1992
5	1414	0	0	0	0	0	0	0	0	10
6	230	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	12
8	0	0	0	0	0	0	0	0	0	3
9	1300	0	0	0	0	0	0	0	0	499909

Train accuracy plot vs Iterations:



Observations:

1. As compared to part **c** , RelU has **converged faster,** even for the lesser valued layer.
2. Time taken is more as there are two 100 layers. But, relatively relu is faster as compared to sigmoid because of the lesser computation.
3. Adaptive learning rate, accuracy curve is steeper than non-adaptive one. Which means learning is faster in Adaptive case. Accuracy is near about same in both cases.

F. MLP

Training accuracy **96.04%**

Testing accuracy **93.78%**

Comment:

1. As compared to part **e,** the accuracy is really great.
2. Also, the time taken in MLP is also really less.

G. Added extra sample to **train data** set of **Royal Flush** category.

Training accuracy 92.56%

Testing accuracy 92.20%

Confusion matrix:

a. Train

	0	1	2	3	4	5	6	7	8	9
0	10599	0	0	0	0	0	0	0	0	0
1	1206	0	0	0	0	0	0	0	0	0
2	442	0	0	0	0	0	0	0	7	64
3	23	0	0	17	0	0	0	0	6	47
4	6	0	0	0	0	0	0	0	2	46
5	36	0	0	0	0	0	0	0	0	0
6	5	0	0	0	0	0	0	0	0	1
7	1	0	0	0	0	0	0	0	0	4
8	0	0	0	0	0	0	0	0	485	0
9	0	0	0	0	0	0	0	0	0	12493

b. Test

	0	1	2	3	4	5	6	7	8	9
0	421397	0	11	4	0	0	0	0	27	1059
1	47622	0	0	0	0	0	0	0	0	0
2	19198	0	2	3	0	0	0	0	194	1724
3	344	0	0	169	0	0	0	0	155	3217
4	15	0	0	0	0	0	0	0	2	1979
5	1424	0	0	0	0	0	0	0	0	0
6	144	0	0	0	0	0	1	0	19	66
7	1	0	0	1	0	0	0	0	0	10
8	0	0	0	0	0	0	0	0	3	0
9	644	0	0	62	0	0	0	0	41	500462

Comments:

a. There is significant **increase** in the **testing accuracy** due to addition of the extra examples of minority class.