

# Assignment – 3

## COL774 Machine Learning

Anirudha Kulkarni

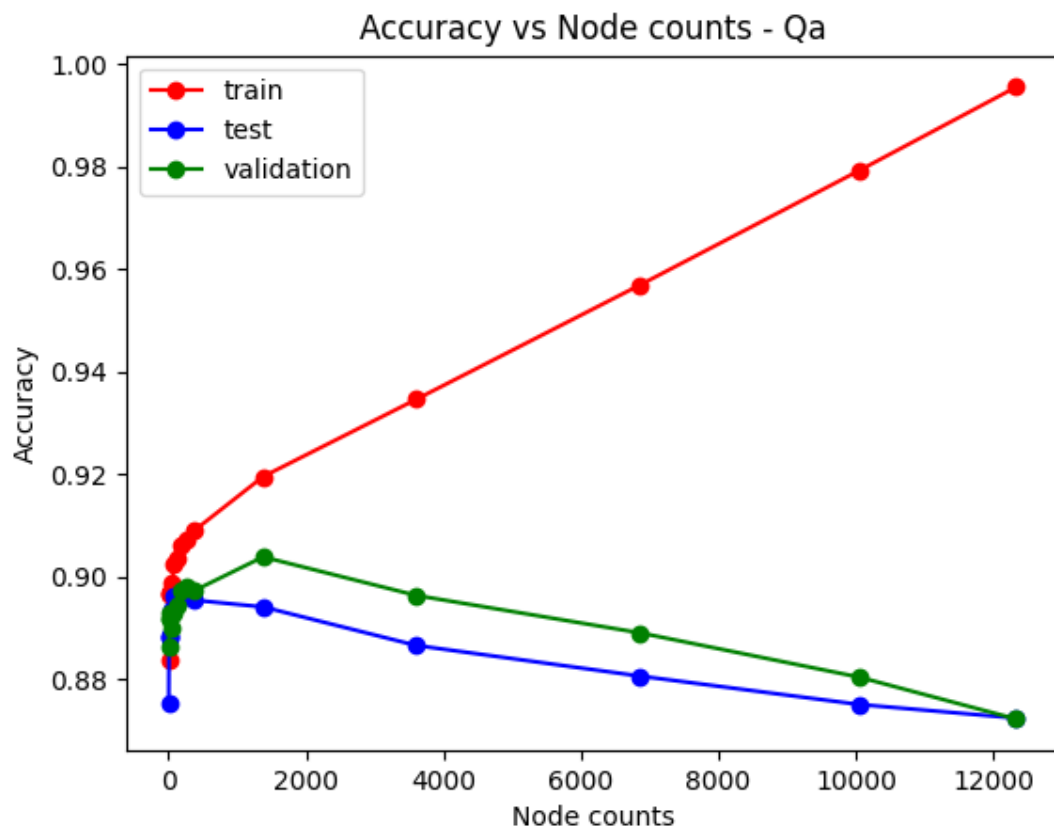
### Q1 Decision Tree

a)

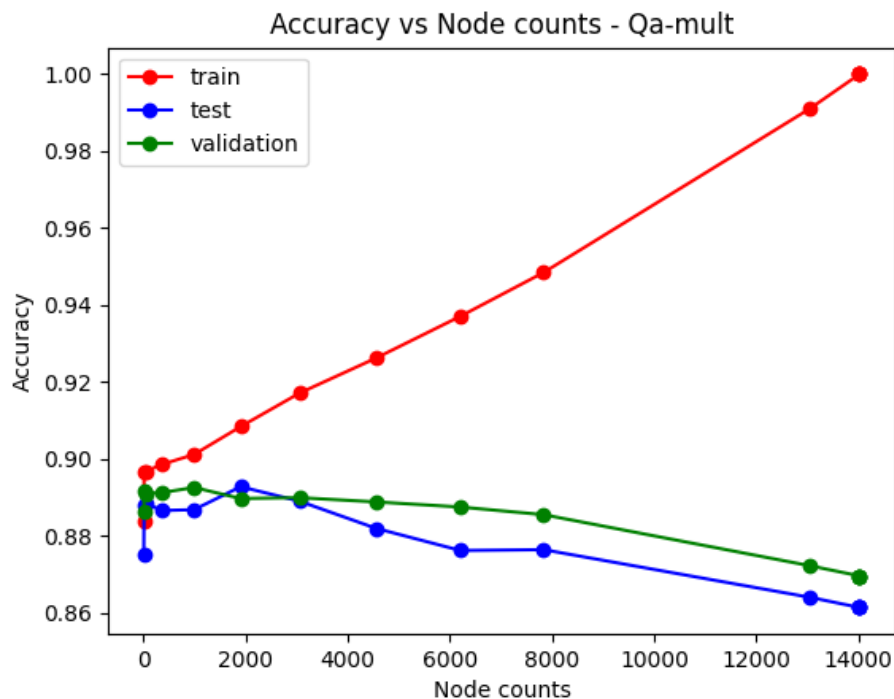
Datapoints:

Attached in datapoints.txt

One Hot encoded:



### Multi way division:



### Comment:

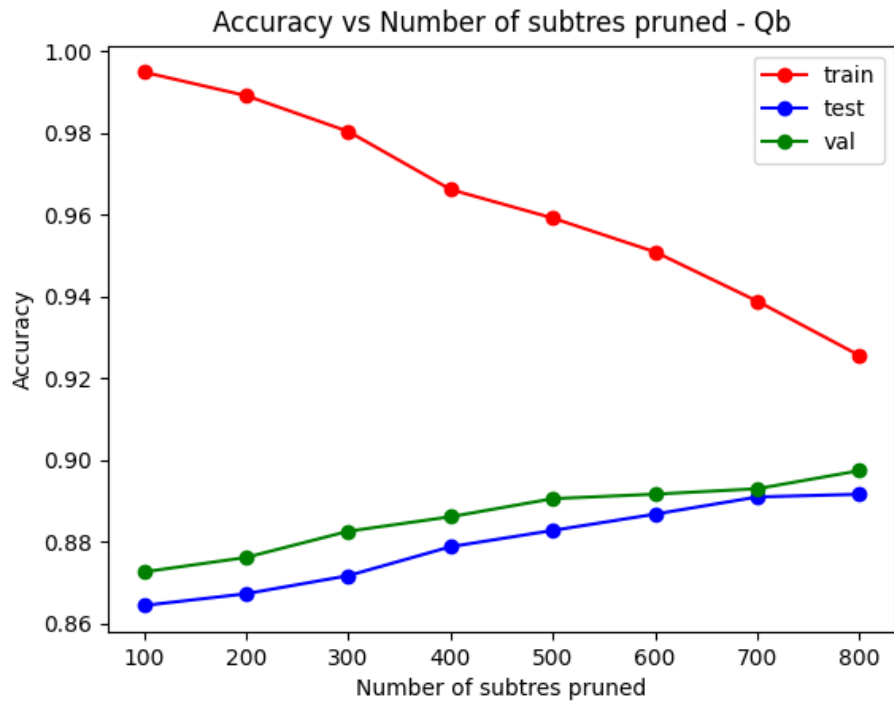
1. One hot encoding vs Multi way division: One hot encoding performs slightly better than multi way division. Multi way division produces large number of nodes compared with one hot encoding even at same depth.

2. As number of nodes increase the train accuracy increases but the validation and test accuracies increase upto a point and then start to decline. This is a case of overfitting. The model fits data correctly till 1000 nodes but after that it fits the noise as well and hence lead to overfit and testing accuracy declines.

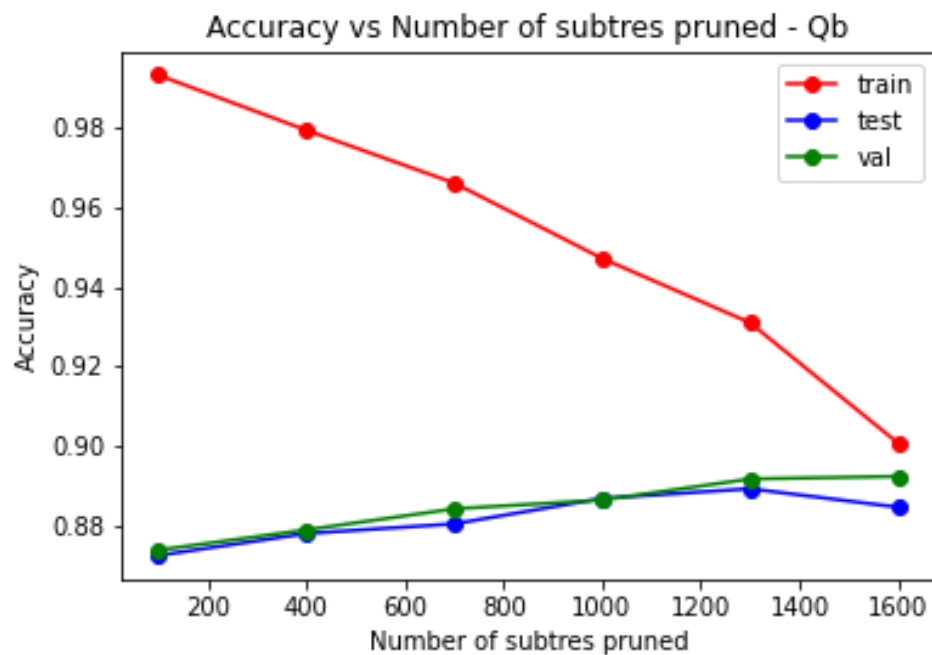
3.

b)

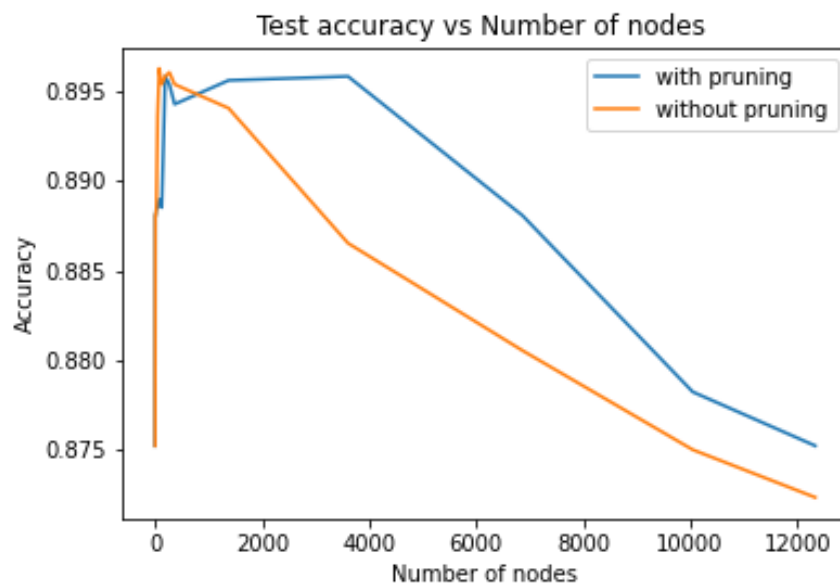
Comparing Accuracy vs Number of trees pruned for **Multi way division**



Comparing Accuracy vs Number of trees pruned for **One hot encoded**



### Test accuracy after maximum possible pruning



#### Comments:

1. increase in pruning increases the validation accuracy as expected as we pruned in way to increase validation accuracy.
2. Increase in pruning decreases training accuracy significantly but this is counteracted by increase in test accuracy which increases with validation accuracy. This is expected as the noise is learnt less as we start to prune and hence lead to fitting model to the underlying patterns only
3. Test accuracy decreases with more number of total nodes in the tree but is more when done with pruning than without pruning.

**c)**

Optimal parameters: n\_estimators: 350, max\_features:0.3, min\_samples\_split: 10

oob score: 0.8907597876575979

Train accuracy: 0.9064642778146428

Test accuracy: 0.8843176288431763

Val accuracy: 0.8900928792569659

Previous model after pruning:

Train accuracy: 0.9006856890068569

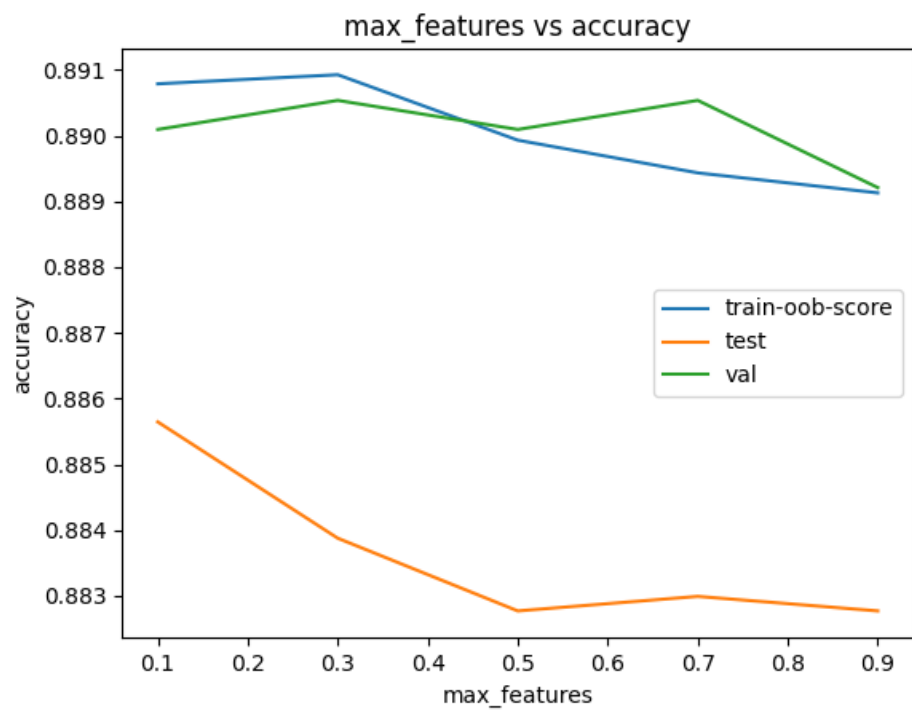
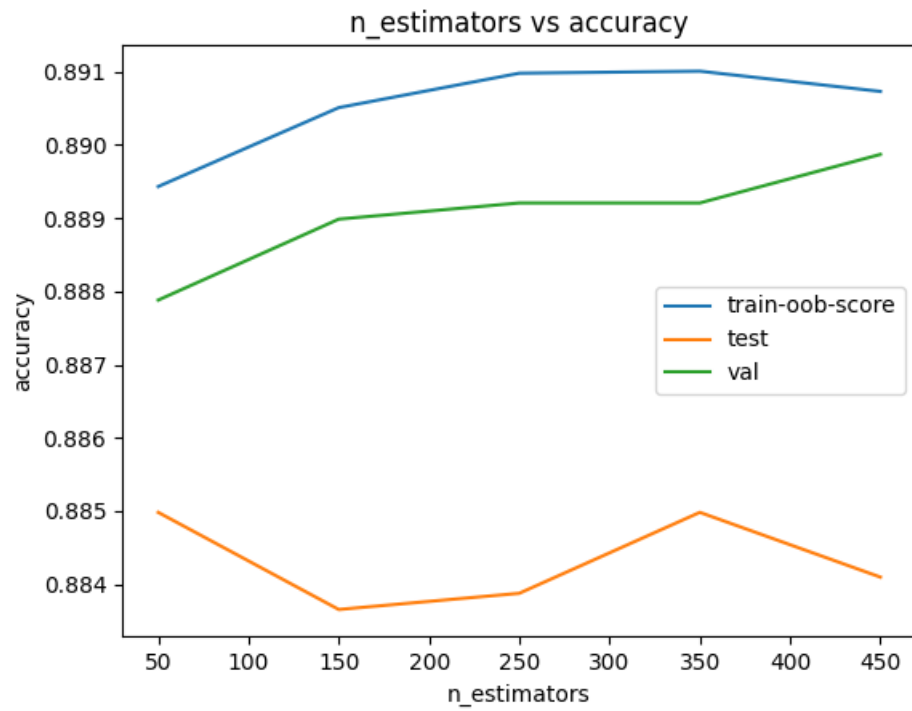
Test accuracy: 0.8845388188453882

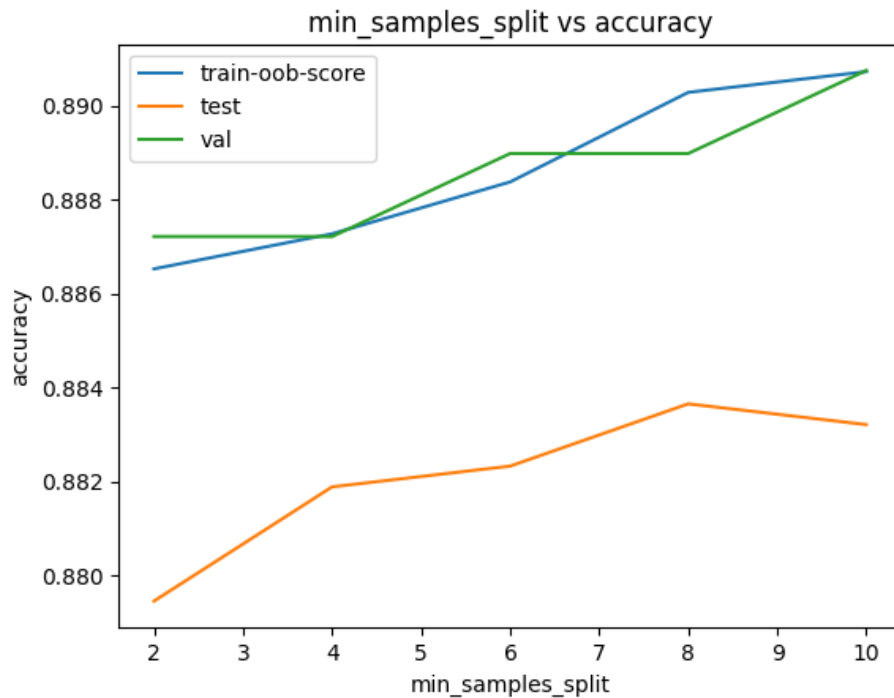
Val accuracy: 0.8923042901371074

**Comments:**

- 1. Both performs almost similarly with Random Forest performing better in training and Pruning performing better in testing and validation. Still the difference is less than 0.02 percent**
- 2. out of bag score is a good indicator of performance on test and validation data. Though the train accuracy is higher than oob score, test accuracy and validation accuracy match with oob score**

d)





### Observations:

We see the peak at optimal parameters in oob score in each graph but test and validation accuracies vary over a limit.

In case of `n_estimators` we see the test and validation follows similar trend as oob score and are maximum at 350 estimators.

In case of `max_features` test accuracy decreases as we increase the `max_features`. Though the optimal is at 0.3 and validation also occurs at 0.3, test accuracy is maximum at 0.1. This can be attributed to noise as the difference in accuracies is less than 0.1%

In case of `min_samples_split` we see validation and train accuracy is max at 10 and test accuracy decreases by 0.1% after 9. Still the overall trend is upwards and matches with oob score trend

### Sensitivity:

`n_estimators` is the least sensitive to absolute parameter change. It varies over by 0.1 percent over entire range of `n_estimator`. `max_feature` is the moderately sensitive to absolute parameter change. It varies by 0.3 percent over entire range. `min_samples_split` is most sensitive with 0.4% variation across range.

## Neural Network:

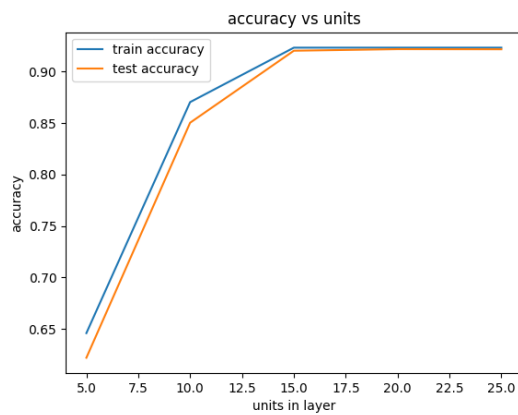
a) attached in the code

b) attached in the code

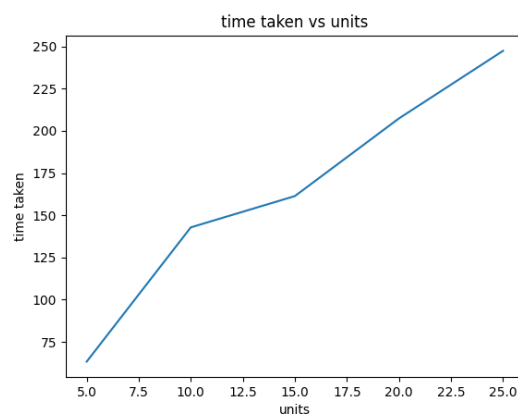
c)

Units	Train time	Train accuracy	Test accuracy
5	63.26	0.6457	0.6278
10	142.82	0.87	0.85
15	161.37	0.92	0.92
20	207.40	0.9233	0.9218
25	247.45	0.9233	0.9217

train accuracy and test accuracy vs units

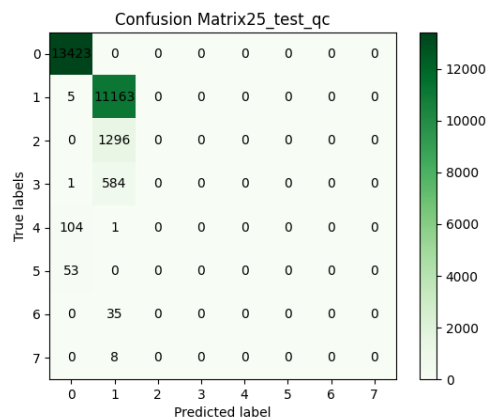
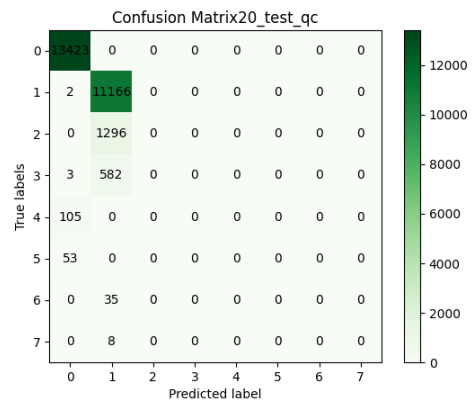
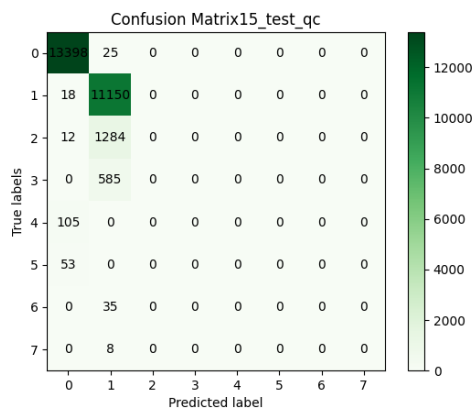
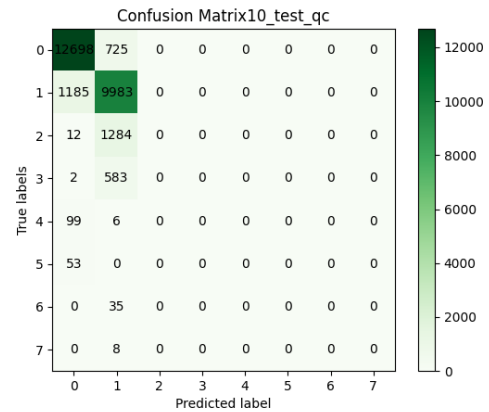
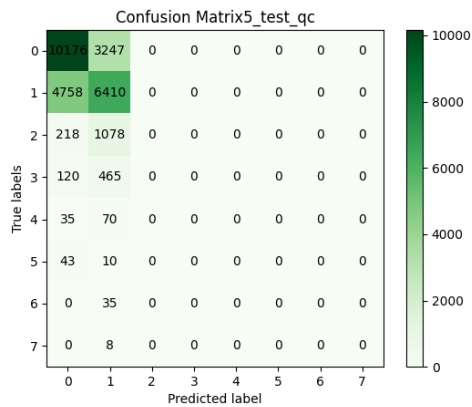


Time taken to train



Stopping criteria: if change in train acc after an epoch is less than epsilon or if max iterations are exceeded where **Epsilon = 10-5, Max iterations = 1400**



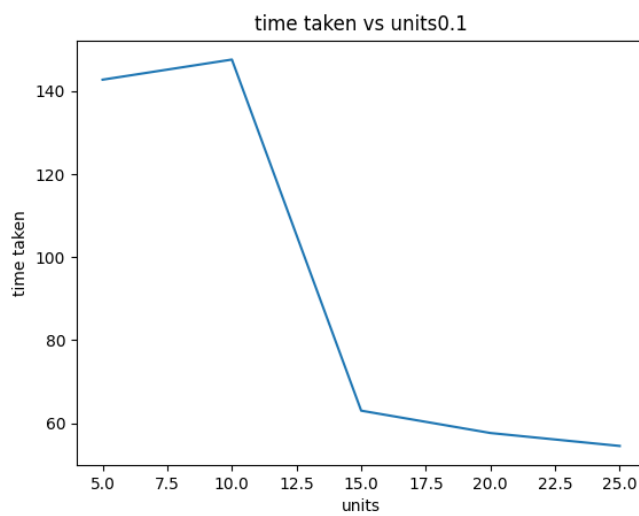
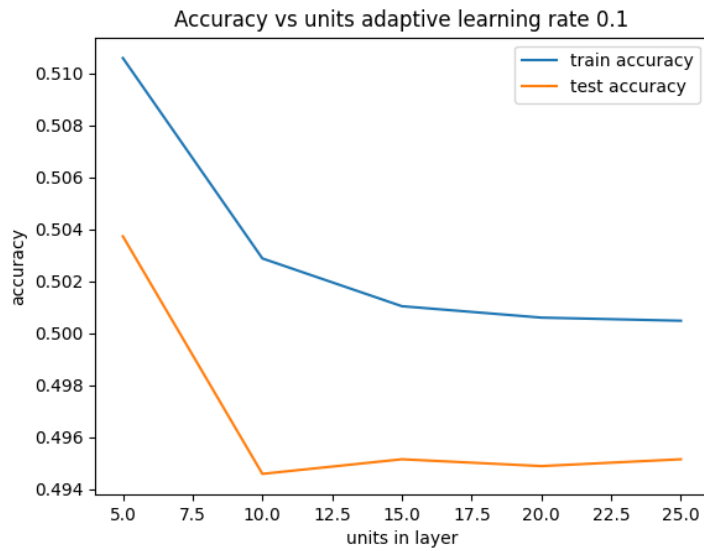


## Observations:

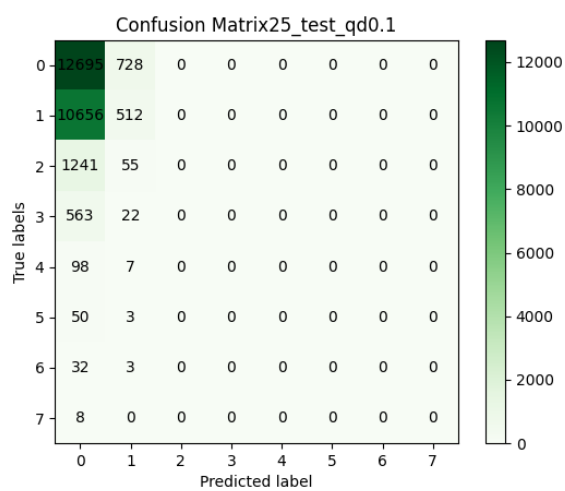
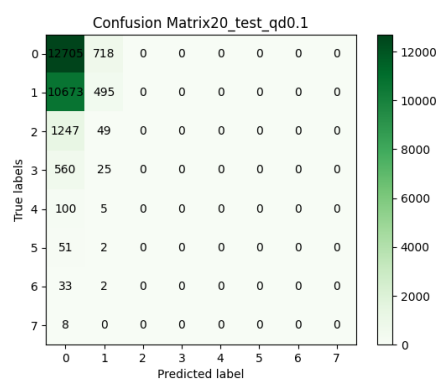
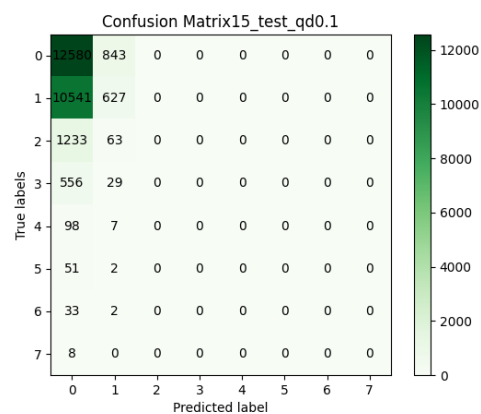
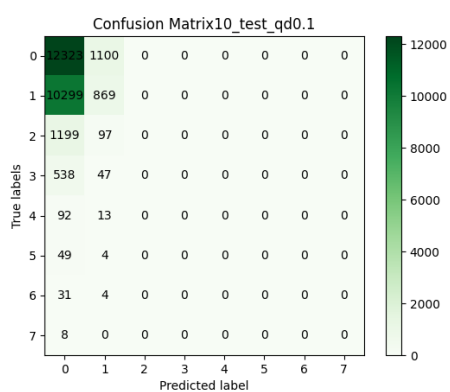
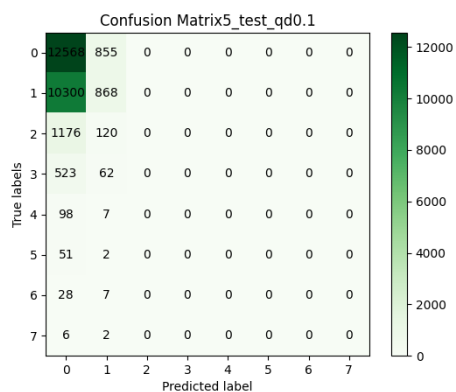
1. Accuracy increases as the number of units in the layer are increased. This is intuitive as more units allows more complex relation to learn. But there is not significant progress after 15 units in the layer. Handful of examples are predicted correctly but time required to train is significant.

d)

When  $\alpha_0 = 0.1$



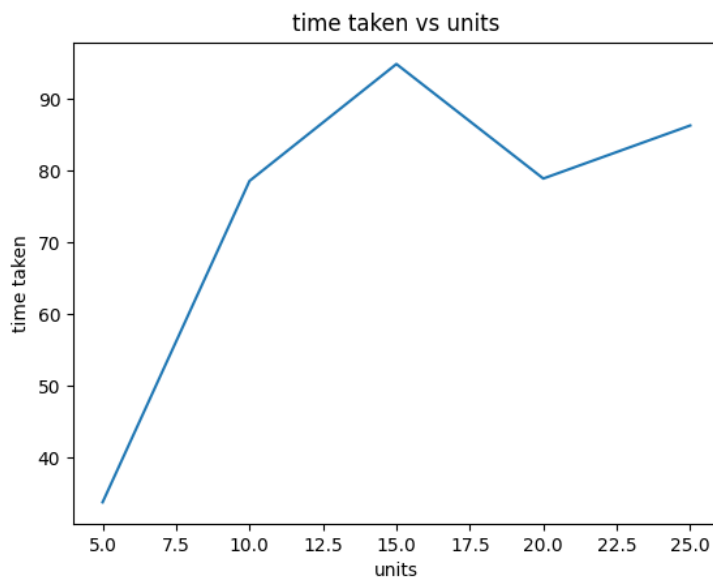
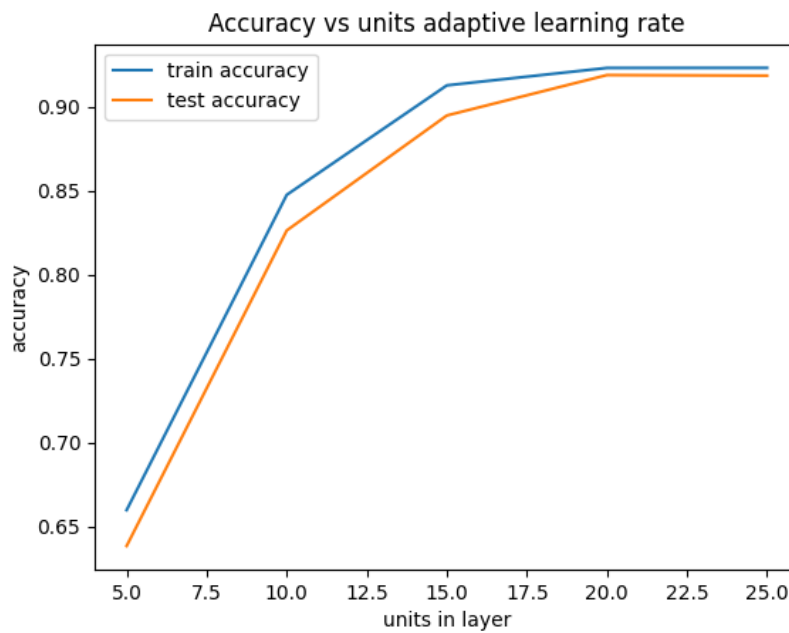
Stopping criteria: if change in train acc after an epoch is less than epsilon or if max iterations are exceeded where **Epsilon =  $10^{-6}$** , **Max iterations = 1400** as the progress is very slow



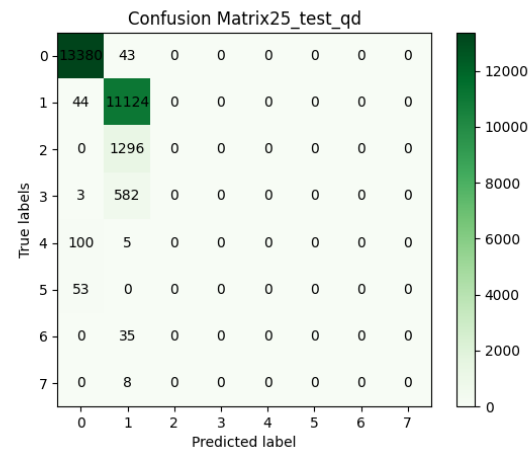
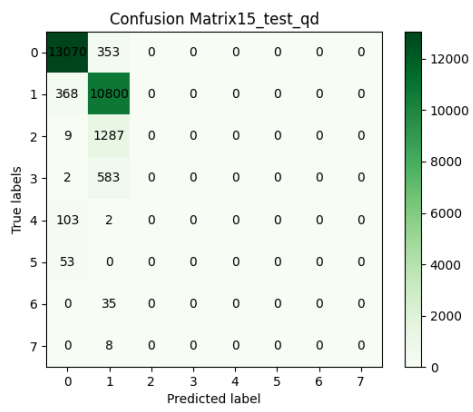
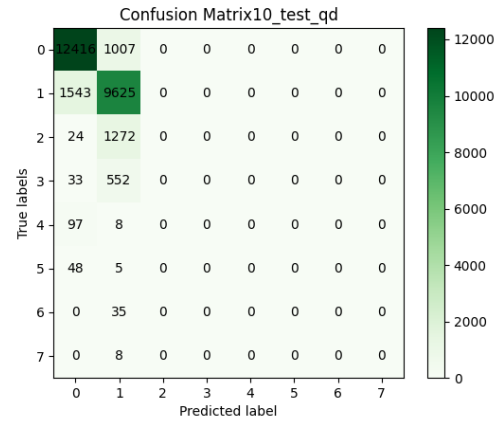
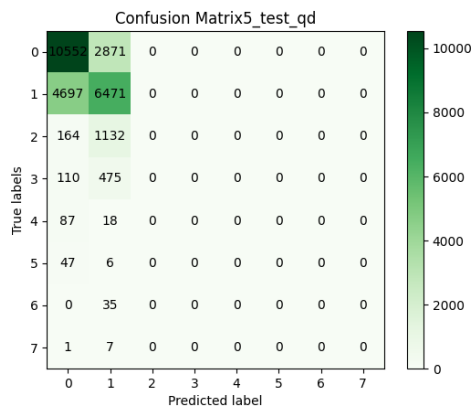
When  $\alpha_0 = 10$

**Train Accuracies:** [0.6724110355857656, 0.8502199120351859, 0.9145541783286686, 0.9233106757297082, 0.9233106757297082]

**Test Accuracies:** [0.649683200239943, 0.8287781651857684, 0.8956997713043152, 0.918269411014884, 0.9177445356727777]



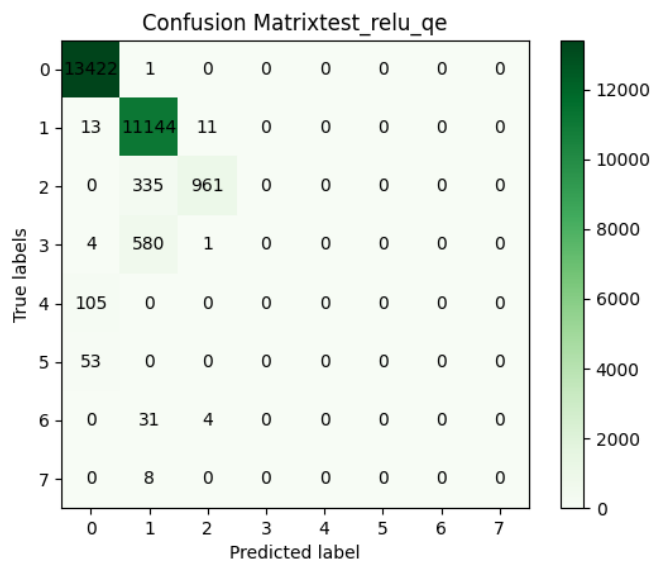
if change in train acc after an epoch is less than epsilon or if max iterations are exceeded where **Epsilon = 10-5, Max iterations = 1400**



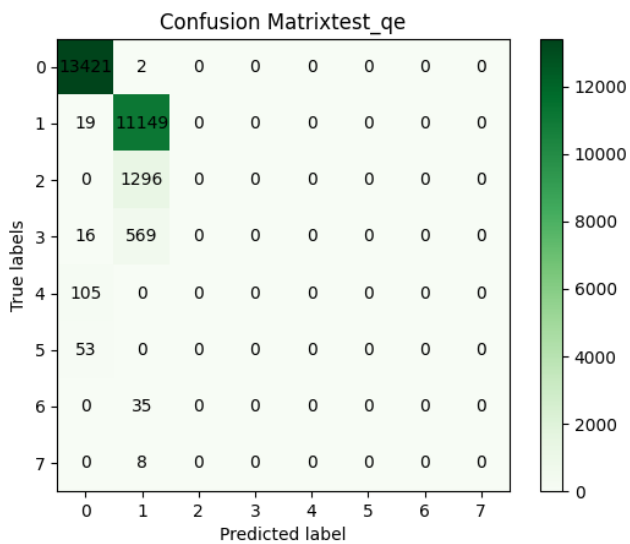
e)

	Sigmoid	Relu
Train time	442.49	51.66
Training accuracy	0.9233106757297082	0.9616953218712515
Test accuracy	0.9211562253964684	0.957035204139017

Relu:



Sigmoid



Comments:

1. Test accuracy and confusion matrix comparison:

Relu performs significantly better in terms of overall accuracy. Also, relu correctly classifies 3<sup>rd</sup> prediction where sigmoid didn't predict any correctly. Difference in training time is also significant. Relu converges much faster

2. single hidden layer with sigmoid vs 2 layers

Single hidden layer could not achieve more than 92% accuracy in all the cases. Whereas 100,100 combinations provided more than 95% test accuracy. Also the accuracy was still high in single layer as only 2 predictions were majority. Single layer could not learn any prediction corresponding to class 3. Hence if the data was without heavy bias then there would have been larger difference in accuracies.

f)

Non adaptive

Training time: 84.9031867980957

Test accuracy: 0.9474374835976456

Adaptive:

Training time: 80.86073112487793

Test accuracy: 0.9474374835976456

Custom implementation performed slightly better than the library implementation. The difference might be due to difference in error function and stopping criteria and learning rate