

Machine Learning (COL774)

Assignment #3

Tanu Kanvar
(2018EEY7537)

Google Drive Link for some of the below data -

<https://drive.google.com/drive/folders/1H6TdhCkPEiCKq9inPPGyTRbUHegY1r1m?usp=sharing>

Screenshots of the entire report can be found at this location.

Q1->Part (a)

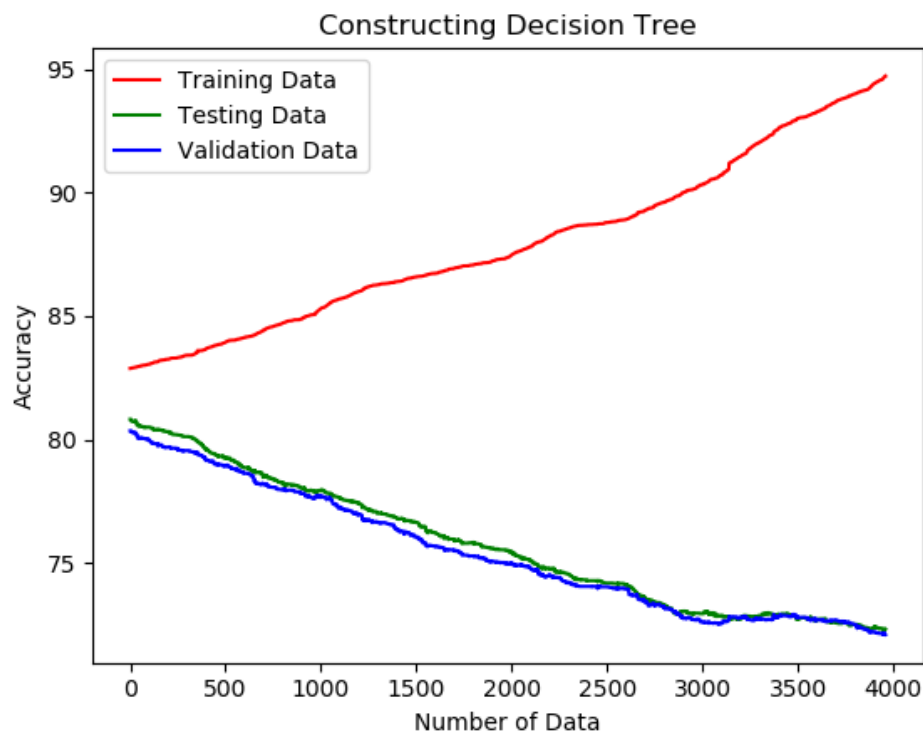
Time to run – 7 mins

of internal nodes of the decision tree = 3960

Training Data Accuracy – 94.7%

Validation Data Accuracy – 72.1%

Testing Data Accuracy – 72.3%



As is seen in the plot above, the accuracy of the decision tree increases on the training set but keeps decreasing on validation and testing data as the number of nodes of the decision tree increases.

This is happening because with each increase of nodes, the tree is overfitting on the data.

Q1->(b)

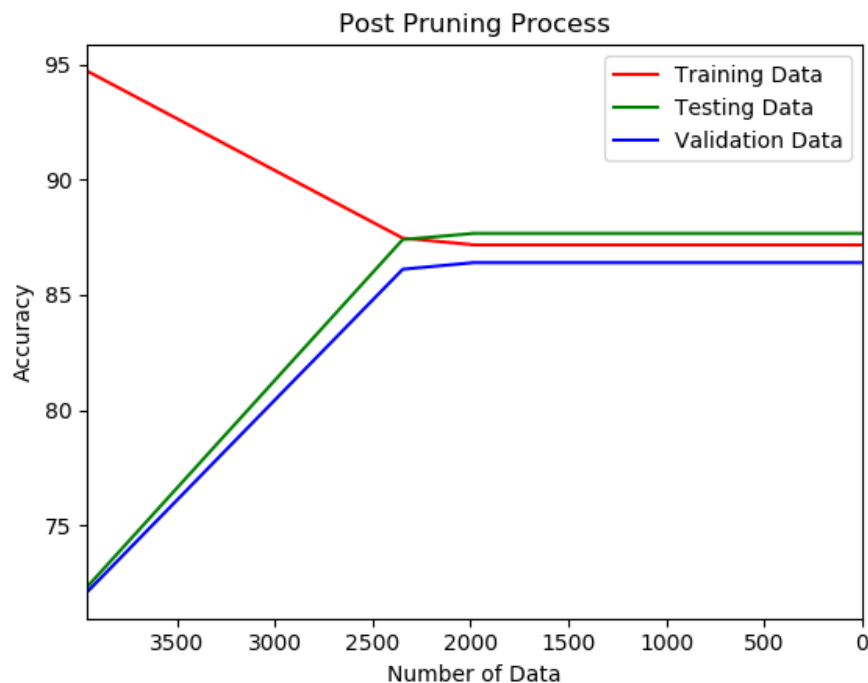
Time taken = 12 mins (Includes creating entire tree and pruning)

of internal nodes of the decision tree = 9

Training Accuracy = 87.168594

Validation Accuracy = 86.397835

Testing Accuracy = 87.667557



As is seen in the graph, as we post prune the tree, the accuracy on the training data decreases but accuracy on the validation and testing data increases. This happens till internal nodes 2200.

On further pruning, the validation accuracy remains same but doesn't decrease. We continue this till internal nodes 9 are remaining. After pruning any of the nodes in this tree, the validation accuracy is decreasing.

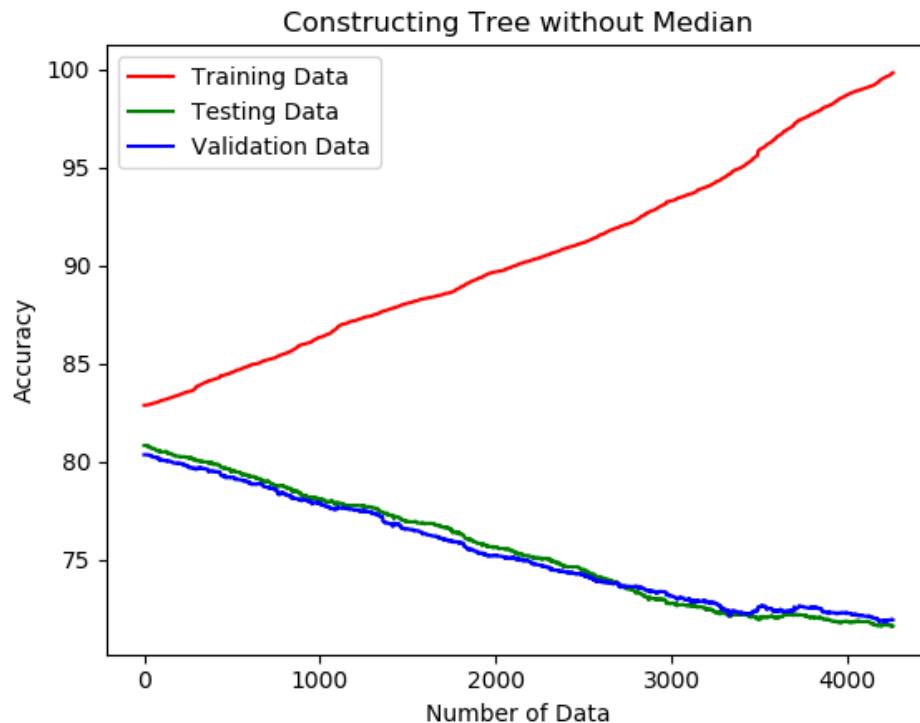
We prune the tree till accuracy doesn't start to decrease on the validation data. This leaves only those nodes in the tree removing which will cause drop on the validation data.

Q1->(c)

Time taken = 12 mins

of internal nodes of the decision tree = 4260

Training Accuracy = 99.838890
Validation Accuracy = 71.931290
Testing Accuracy = 71.599998



We observe increase in accuracy as compared to part(a) because in part(a) we are treating continuous data as Boolean and so important information is lost.

Also because continuous is treated Boolean in part(a), such an attribute can come only once in the branch. This reduced the accuracy.

The continuous attributes that appeared in the tree and number of times they appeared and the conditions on which those attributes are split can be found in the file "**q1PartCAttr.txt**" on the google drive

Q1->(d)

Default parameters

of nodes – 5015

Tree height 43

Training Accuracy: 99.961

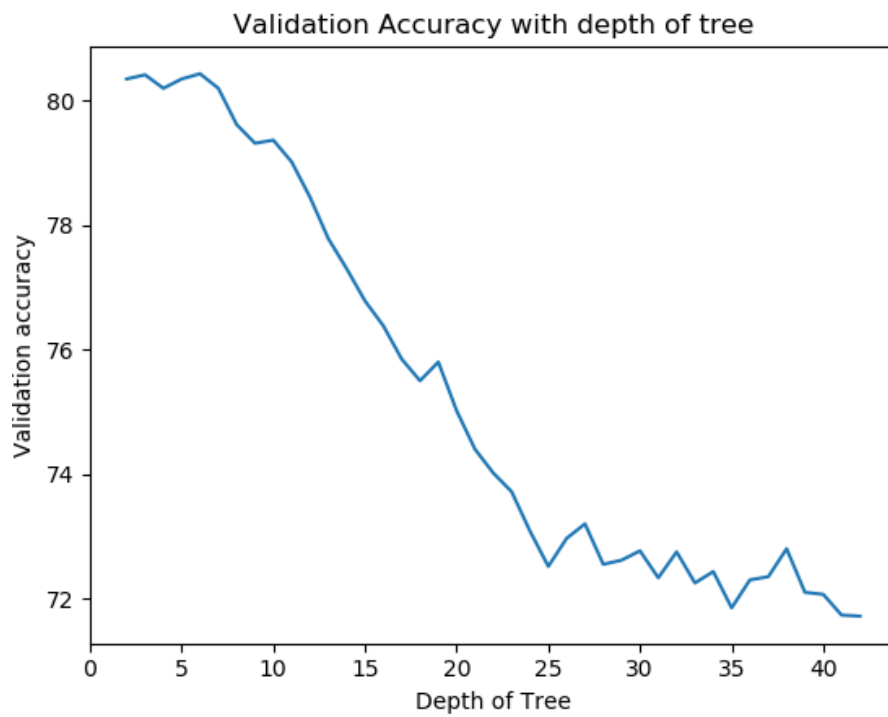
Validation Accuracy: 72.083

Testing Accuracy: 72.117

The minor difference is because scikit treats categorical data is treated as continuous and so there can be multiple nodes in the same branch for the same attribute. This increases the accuracy.

Depth of tree variation –

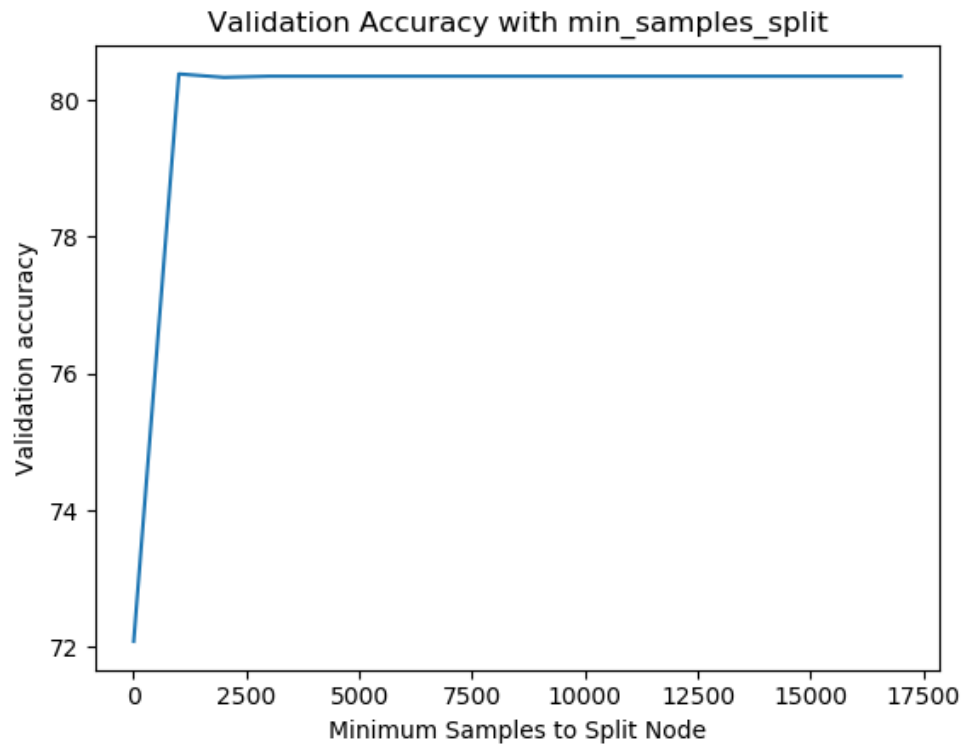
Range = 2 - 43



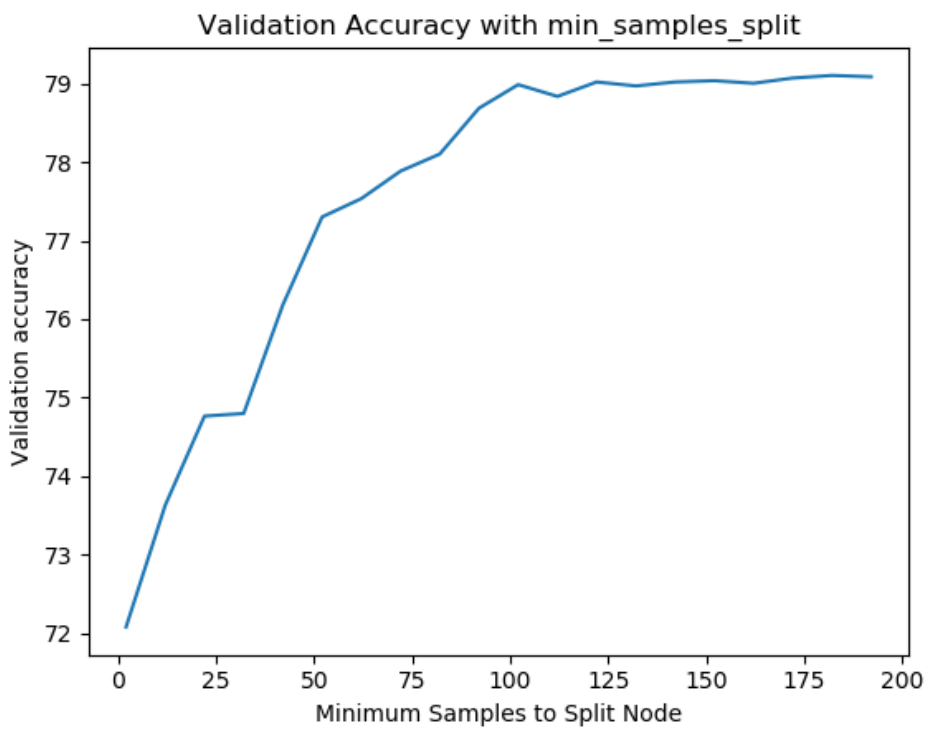
Min samples split

Minimum number of samples needed to split node

Range = 2 - 18000 taken at every 100 steps

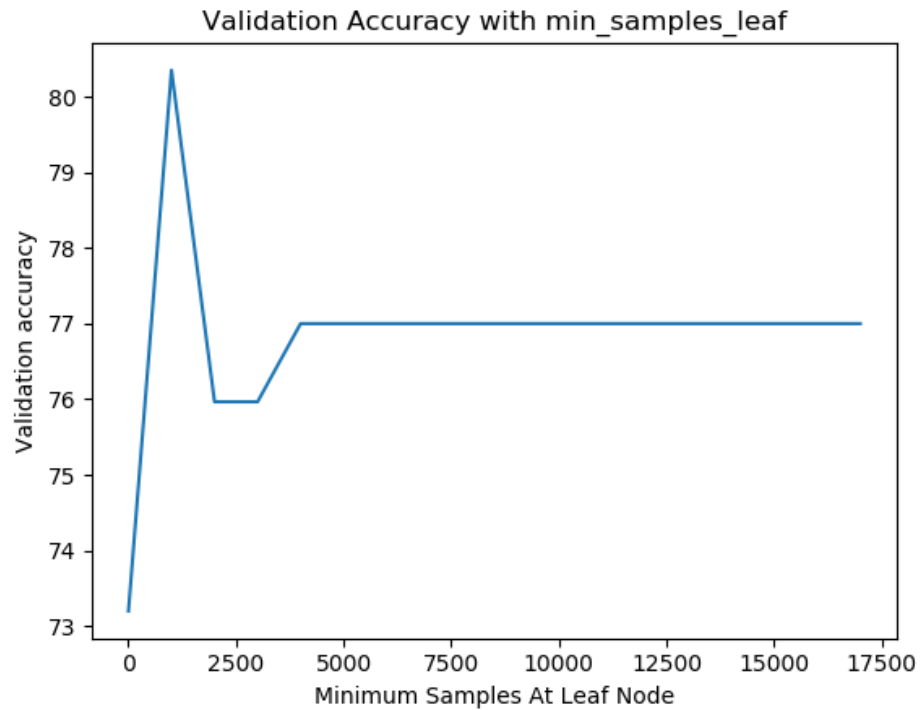


Range = 2 - 200 taken at every 10 steps

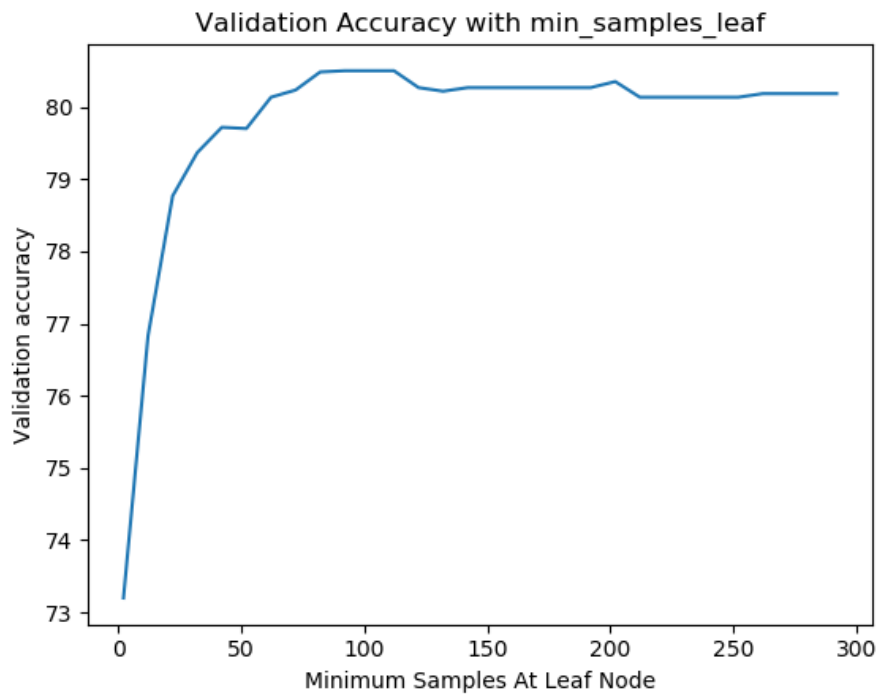


Min samples leaf

Range = 2 – 18000 taken at every 1000 steps



Range = 2 – 300 taken at every 10 steps



Depth of tree variation, Min_samples_split, Min_samples_leaf

Ran all the above ranges together in a for loop and found accuracy of validation

Found max out of all. **Max parameters** were

Validation Accuracy = 80.517

[leaf, split, depth] = [100, 802, 9]

min_samples_leaf = 100, min_samples_split = 802, max_depth = 9

Training Accuracy: 83.111

Validation Accuracy: 80.517

Testing Accuracy: 80.967

Part (b) has higher accuracy on all 3 cases. Reason – There might be some other parameter changing which could improve the results.

Part (c) training accuracy is pretty high as compared to these accuracies and test and validation accuracies are pretty low as compared to these accuracies. Reason – Overfitting in part(c)

Q1->(e)

One hot encoding

Training Accuracy: 83.200

Validation Accuracy: 80.067

Testing Accuracy: 80.467

Part(b) data performs better than this one maybe because there needs to be some additional parameter changes which could improve the accuracy.

It is comparable to part (d). One-hot encoding is useful in algorithms where categorical data is strings and algorithms cannot handle them.

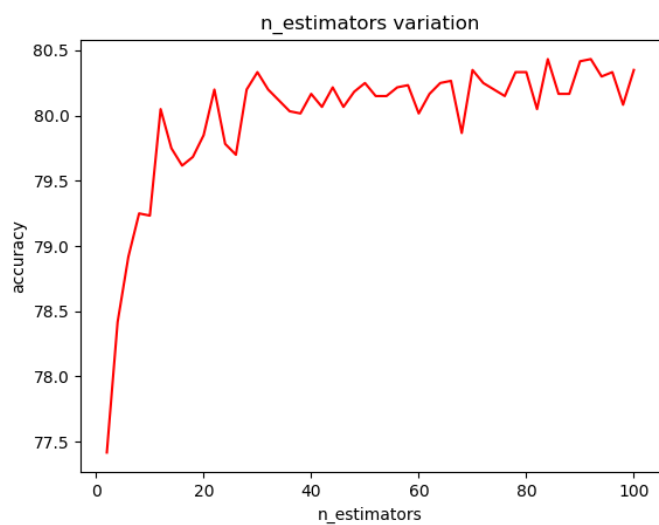
Hence not much difference.

For part(c), the difference is because decision tree in part (c) is overfit.

Q1->(f)

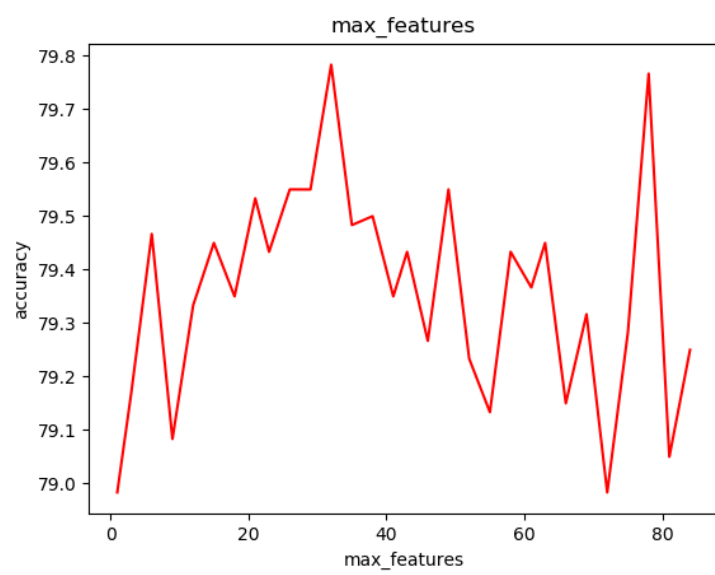
Estimators

Range = 2 - 100



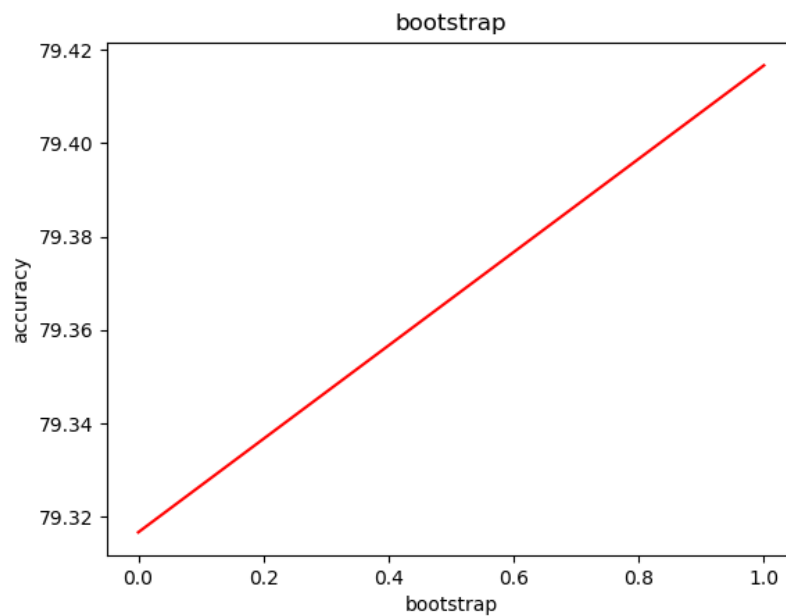
Max Features

Range = 3 – 90



Bootstrap

Values = True, False



Best Parameter Setting

max_feature: 32
Bootstrap: 1
n_estimators: 84

Train Accuracy: 99.95
Test Accuracy: 80.21666666666667
Validation Accuracy: 80.10000000000001

The Test and Validation remain same as above except for part(b), which shows higher accuracy. But Training accuracy increases as compared to all the above parts. Reason – Averaging of the decisions of multiple decision Trees.

Q2->(a)

Number of Features = 85

One hot encoding Training file can be found in google drive (at the location specified at the top of report) with the name **"q2partAtrain.txt"** and testing file one hot encoding is with the name **"q2partAtest.txt"**

Q2->(b)

Even after increasing maxepocs, it continued to converge but there was not much difference in the accuracy

Conditions:

maxepocs = 500
batchSize = 100
noOfInputs = 85
hiddenLayers = [10]
noOfOutputs = 10
threshold = 1e-5
learning_rate = 0.1

Training Accuracy = 75.20591763294682%

Testing Accuracy = 72.4711%

Q2->(c)

Condition:

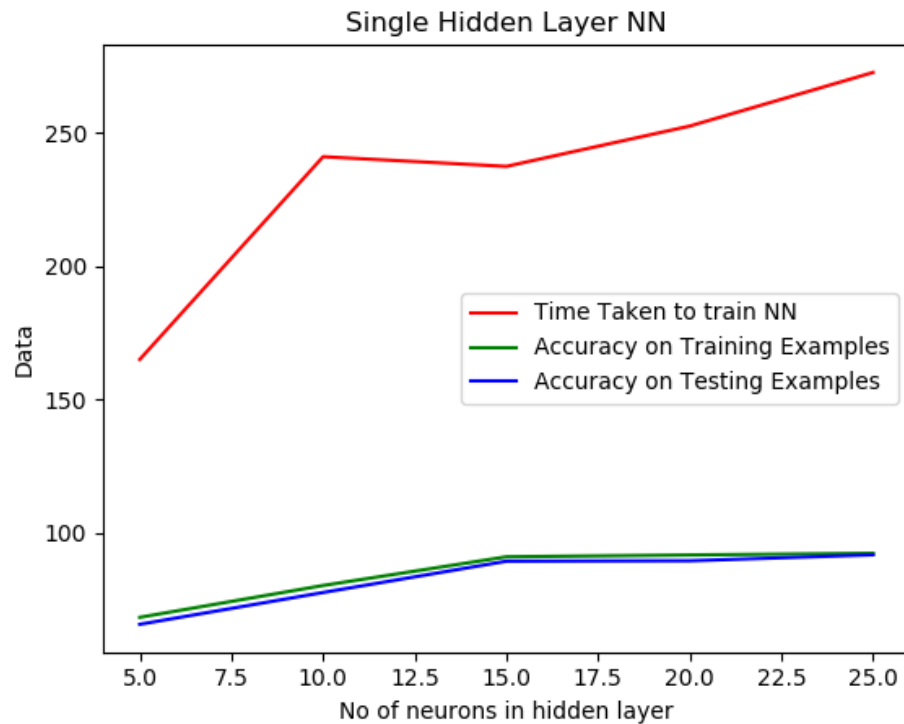
maxepocs = 3000
batchSize = 100
noOfInputs = 85
hiddenLayersArr = [[5],[10],[15],[20],[25]]
noOfOutputs = 10
threshold = 1e-5
learning_rate = 0.1

<u>S.No.</u>	<u>Number of Neurons</u>	<u>Time Taken to train NN (s)</u>	<u>Training Accuracy</u>	<u>Testing Accuracy</u>
1.	5	165	68.428628548580	65.8126
2.	10	241	80.36385445821	77.7358
3.	15	237	91.12355057976	89.4601
4.	20	238	91.803278688524	89.6135
5.	25	270	92.423030787684	91.8634

Time to learn the network increases with increase in the number of neurons.

Accuracy of the network also increases with the increase in the number of neurons.

This observation can be seen in the below graph.



S.No.	Number of Neurons	Confusion Matrix Training	Confusion Matrix Testing
1.	5	Predicted 0 1 All Actual 0 10118 2375 12493 1 3603 6996 10599 2 247 959 1206 3 75 438 513 4 84 9 93 5 39 15 54 6 5 31 36 7 0 6 6 8 5 0 5 9 5 0 5 All 14181 10829 25010	Predicted 0 1 All Actual 0 392995 108214 501209 1 157367 265131 422498 2 9775 37847 47622 3 3501 17620 21121 4 3477 408 3885 5 1560 436 1996 6 144 1280 1424 7 4 226 230 8 11 1 12 9 3 0 3 All 568837 431163 1000000
2.	10	Predicted 0 1 3 All Actual 0 11959 534 0 12493 1 2459 8140 0 10599 2 90 1116 0 1206 3 58 455 0 513 4 68 24 1 93 5 50 4 0 54 6 2 34 0 36	Predicted 0 1 3 All Actual 0 468842 32367 0 501209 1 113959 308516 23 422498 2 3651 43965 6 47622 3 2878 18243 0 21121 4 2941 944 0 3885 5 1873 123 0 1996 6 72 1352 0 1424

		7 0 6 0 6 8 5 0 0 5 9 2 3 0 5 All 14693 10316 1 25010	7 7 223 0 230 8 8 4 0 12 9 2 1 0 3 All 594233 405738 29 1000000
3.	15	Predicted 0 1 All Actual 0 12429 64 12493 1 238 10361 10599 2 6 1200 1206 3 14 499 513 4 77 16 93 5 52 2 54 6 0 36 36 7 1 5 6 8 3 2 5 9 4 1 5 All 12824 12186 25010	Predicted 0 1 4 All Actual 0 489853 11356 0 501209 1 17749 404748 1 422498 2 131 47491 0 47622 3 446 20675 0 21121 4 3619 266 0 3885 5 1941 55 0 1996 6 4 1420 0 1424 7 8 222 0 230 8 11 1 0 12 9 2 1 0 3 All 513764 486235 1 1000000
4.	20	Predicted 0 1 3 All Actual 0 12465 28 0 12493 1 104 10495 0 10599 2 0 1206 0 1206 3 26 487 0 513 4 76 16 1 93 5 50 4 0 54 6 0 36 0 36 7 1 5 0 6 8 4 1 0 5 9 5 0 0 5 All 12731 12278 1 25010	Predicted 0 1 3 5 All Actual 0 490311 10894 1 3 501209 1 16667 405824 3 4 422498 2 21 47601 0 0 47622 3 550 20570 0 1 21121 4 3523 361 1 0 3885 5 1952 44 0 0 1996 6 0 1424 0 0 1424 7 18 212 0 0 230 8 12 0 0 0 12 9 3 0 0 0 3 All 513057 486930 5 8 1000000
5.	25	Predicted 0 1 4 5 All Actual 0 12493 0 0 0 12493 1 0 10599 0 0 10599 2 24 1182 0 0 1206 3 63 449 1 0 513 4 46 25 22 0 93 5 46 7 0 1 54 6 1 35 0 0 36 7 2 4 0 0 6 8 4 1 0 0 5 9 3 2 0 0 5 All 12682 12304 23 1 25010	Predicted 0 1 4 5 7 All Actual 0 499157 1966 86 0 0 501209 1 3049 419397 39 13 0 422498 2 496 47126 0 0 0 47622 3 1907 19126 13 69 6 21121 4 3531 274 80 0 0 3885 5 1975 19 2 0 0 1996 6 1 1423 0 0 0 1424 7 79 151 0 0 0 230 8 9 3 0 0 0 12 9 3 0 0 0 0 3 All 510207 489485 220 82 6 1000000

As the number of neurons in the hidden layer increases, the number of classes learnt also increases. E.g. for neurons 5, only 2 classes (0 and 1) are learnt. For neurons 10, 15, 20, 3 output classes are learnt. For 25 neurons, 5 classes are learnt.

This means as the neurons increases, the learned model becomes more accurate.

Q2->Part(d)

Condition:

maxepocs = 3000
 batchSize = 100
 noOfInputs = 85
 hiddenLayersArr = [[5,5],[10,10],[15,15],[20,20],[25,25]]
 noOfOutputs = 10
 threshold = 1e-5
 learning_rate = 0.1

S.No.	Number of Neurons	Time Taken to train NN (s)	Training Accuracy	Testing Accuracy
1.	5,5	177	61.395441823270	59.018
2.	10,10	270	76.929228308676	71.2716
3.	15,15	300	92.299080367852	89.8488
4.	20,20	330	96.489404238304	94.9187
5.	25,25	360	49.952019192323	50.1209

Time to learn the network increases with increase in the number of neurons.

Accuracy of the network also increases with the increase in the number of neurons.

However in the last case [25,25], the accuracy suddenly decreases. Reason might be it might not be able to converge.

As can be seen, the accuracy of the network increased with 2 hidden layers as compared to 1 hidden layer in the previous part.

S.No.	Number of Neurons	Confusion Matrix Training	Confusion Matrix Testing
1.	5	Predicted 0 1 2 All Actual 0 6406 6087 0 12493 1 1691 8888 20 10599 2 53 1092 61 1206 3 55 458 0 513 4 44 49 0 93 5 30 24 0 54 6 1 34 1 36 7 0 6 0 6 8 2 3 0 5 9 0 5 0 5 All 8282 16646 82 25010	Predicted 0 1 2 All Actual 0 246019 255150 40 501209 1 77933 342904 1661 422498 2 2275 44090 1257 47622 3 2422 18695 4 21121 4 2043 1842 0 3885 5 1052 944 0 1996 6 63 1337 24 1424 7 8 222 0 230 8 6 6 0 12 9 0 3 0 3 All 331821 665193 2986 1000000

2.	10	Predicted 0 1 2 All Actual 0 11099 1394 0 12493 1 2582 7995 22 10599 2 159 901 146 1206 3 86 424 3 513 4 69 24 0 93 5 45 9 0 54 6 1 33 2 36 7 0 6 0 6 8 4 1 0 5 9 5 0 0 5 All 14050 10787 173 25010	Predicted 0 1 2 All Actual 0 418510 82691 8 501209 1 131142 289330 2026 422498 2 5792 36954 4876 47622 3 3411 17502 208 21121 4 2971 914 0 3885 5 1657 339 0 1996 6 93 1165 166 1424 7 8 221 1 230 8 9 3 0 12 9 3 0 0 3 All 563596 429119 7285 1000000
3.	15	Predicted 0 1 2 3 All Actual 0 12301 192 0 0 12493 1 737 9777 68 17 10599 2 41 171 936 58 1206 3 7 218 218 70 513 4 91 2 0 0 93 5 53 1 0 0 54 6 0 4 31 1 36 7 0 0 6 0 6 8 4 1 0 0 5 9 5 0 0 0 5 All 13239 10366 1259 146 25010	Predicted 0 1 2 3 All Actual 0 487224 13950 33 2 501209 1 39708 375768 6165 857 422498 2 1400 9808 33560 2854 47622 3 239 10100 8846 1936 21121 4 3786 99 0 0 3885 5 1949 47 0 0 1996 6 1 176 1228 19 1424 7 0 27 184 19 230 8 12 0 0 0 12 9 3 0 0 0 3 All 534322 409975 50016 5687 1000000
4.	20	Predicted 0 1 2 3 All Actual 0 12342 151 0 0 12493 1 38 10542 10 9 10599 2 0 45 1147 14 1206 3 0 355 57 101 513 4 92 1 0 0 93 5 52 2 0 0 54 6 0 4 30 2 36 7 0 4 2 0 6 8 5 0 0 0 5 9 5 0 0 0 5 All 12534 11104 1246 126 25010	Predicted 0 1 2 3 All Actual 0 490880 10329 0 0 501209 1 4972 413658 2045 1823 422498 2 75 3677 42276 1594 47622 3 13 15921 2814 2373 21121 4 3803 82 0 0 3885 5 1956 40 0 0 1996 6 2 53 1327 42 1424 7 0 134 73 23 230 8 11 1 0 0 12 9 3 0 0 0 3 All 501715 443895 48535 5855 1000000
5.	25	Predicted 0 All Actual 0 12493 12493 1 10599 10599 2 1206 1206 3 513 513 4 93 93 5 54 54 6 36 36	Predicted 0 All Actual 0 501209 501209 1 422498 422498 2 47622 47622 3 21121 21121 4 3885 3885 5 1996 1996 6 1424 1424

		7	6	6	7	230	230
		8	5	5	8	12	12
		9	5	5	9	3	3
		All	25010	25010	All	1000000	1000000

As the number of neurons in the hidden layer increases, the number of classes learnt also increases. E.g. for neurons 5, only 3 classes (0, 1 and 2) are learnt. For neurons 10, 15, 20, 4 output classes are learnt. For 25 neurons, the accuracy decrease as explained above.

As can be seen, the number of predicted output classes increased with 2 hidden layers as compared to 1 hidden layer in the previous part.

This means as the neurons and the hidden layer increases, the learned model becomes more accurate. However as can be seen in 25,25 case, the accuracy decreased. So there is a limit to increase.

Q2->(e)

As the neurons and layers increased the rate of learning is fast and soon it reaches to minima. So for lower number of neurons or layers, the learning rate started decreasing late (i.e. the tolerance reached late) as compared to more number of neurons or layers, where the learning rate started decreasing in very less number of iterations (i.e. the tolerance was reached in less number of iterations).

With $10e-4$ and $10e-5$ the learning rate decreases very fast as a result in 5,10,15 case observed marginal decrease in accuracy. In 25, 20, and all double layered case, it couldn't even start learning and it stopped merging because the learning rate became close to 0 very fast as a result the network couldn't learn and we got 0.2% accuracy in one of the cases.

Tolerance – $10e-6$

S.No.	Number of Neurons	Time Taken to train NN (s)	Training Accuracy	Testing Accuracy
1.	5	200	66.025589764094	63.5762
2.	10	230	83.702518992403	82.0755
3.	15	250	90.891643342662	89.6458
4.	20	300	92.363054778088	92.0077
5.	25	30	0.0199920031987	0.0003
6.	5,5	300	49.95201919232	50.1209
7.	10,10	1	49.952019192323	50.1209
8.	15,15	1	49.952019192323	50.1209
9.	20,20	1	49.952019192323	50.1209
10.	25,25	1	49.952019192323	50.1209

S.No.	Number of Neurons	Confusion Matrix Training	Confusion Matrix Testing
1.	5	Predicted 0 1 All	Predicted 0 1 All

		Actual 0 10786 1707 12493 1 4872 5727 10599 2 248 958 1206 3 168 345 513 4 83 10 93 5 50 4 54 6 4 32 36 7 0 6 6 8 5 0 5 9 5 0 5 All 16221 8789 25010	Actual 0 420714 80495 501209 1 207450 215048 422498 2 10327 37295 47622 3 7459 13662 21121 4 3643 242 3885 5 1678 318 1996 6 139 1285 1424 7 45 185 230 8 12 0 12 9 2 1 3 All 651469 348531 1000000
2.	10	Predicted 0 1 3 All Actual 0 12189 304 0 12493 1 1855 8744 0 10599 2 45 1161 0 1206 3 62 450 1 513 4 85 8 0 93 5 52 2 0 54 6 2 34 0 36 7 0 6 0 6 8 5 0 0 5 9 5 0 0 5 All 14300 10709 1 25010	Predicted 0 1 3 All Actual 0 483115 18094 0 501209 1 84815 337633 50 422498 2 1888 45731 3 47622 3 2636 18478 7 21121 4 3431 454 0 3885 5 1934 62 0 1996 6 23 1401 0 1424 7 0 230 0 230 8 11 1 0 12 9 3 0 0 3 All 577856 422084 60 1000000
3.	15	Predicted 0 1 3 4 5 All Actual 0 12471 22 0 0 0 12493 1 355 10244 0 0 0 10599 2 18 1183 4 0 1 1206 3 27 463 17 4 2 513 4 79 14 0 0 0 93 5 52 2 0 0 0 54 6 0 34 2 0 0 36 7 1 4 1 0 0 6 8 4 1 0 0 0 5 9 4 1 0 0 0 5 All 13011 11968 24 4 3 25010	Predicted 0 1 3 4 5 All Actual 0 496272 4935 1 1 0 501209 1 22043 400060 265 47 83 422498 2 314 47196 85 0 27 47622 3 829 20079 126 40 47 21121 4 3658 225 2 0 0 3885 5 1985 11 0 0 0 1996 6 60 1352 7 0 5 1424 7 20 193 7 2 8 230 8 12 0 0 0 0 12 9 2 1 0 0 0 3 All 525195 474052 493 90 170 1000000
4.	20	Predicted 0 1 4 5 All Actual 0 12493 0 0 0 12493 1 0 10599 0 0 10599 2 10 1196 0 0 1206 3 57 429 8 19 513 4 67 18 8 0 93 5 54 0 0 0 54 6 0 36 0 0 36 7 4 2 0 0 6	Predicted 0 1 4 5 6 All Actual 0 500044 1103 60 2 0 501209 1 2480 419920 71 27 0 422498 2 102 47520 0 0 0 47622 3 1581 18681 366 480 13 21121 4 3408 360 113 4 0 3885 5 1982 12 2 0 0 1996 6 0 1424 0 0 0 1424 7 76 130 11 5 8 230

		8 2 2 1 0 5 9 3 1 1 0 5 All 12690 12283 18 19 25010	8 10 2 0 0 0 12 9 2 1 0 0 0 3 All 509685 489153 623 518 21 1000000
5.	25	Predicted 9 All Actual 0 12493 12493 1 10599 10599 2 1206 1206 3 513 513 4 93 93 5 54 54 6 36 36 7 6 6 8 5 5 9 5 5 All 25010 25010	Predicted 9 All Actual 0 501209 501209 1 422498 422498 2 47622 47622 3 21121 21121 4 3885 3885 5 1996 1996 6 1424 1424 7 230 230 8 12 12 9 3 3 All 1000000 1000000
6.	5,5	Predicted 0 All Actual 0 12493 12493 1 10599 10599 2 1206 1206 3 513 513 4 93 93 5 54 54 6 36 36 7 6 6 8 5 5 9 5 5 All 25010 25010	Predicted 0 All Actual 0 501209 501209 1 422498 422498 2 47622 47622 3 21121 21121 4 3885 3885 5 1996 1996 6 1424 1424 7 230 230 8 12 12 9 3 3 All 1000000 1000000
7.	10,10	Predicted 0 All Actual 0 12493 12493 1 10599 10599 2 1206 1206 3 513 513 4 93 93 5 54 54 6 36 36 7 6 6 8 5 5 9 5 5 All 25010 25010	Predicted 0 All Actual 0 501209 501209 1 422498 422498 2 47622 47622 3 21121 21121 4 3885 3885 5 1996 1996 6 1424 1424 7 230 230 8 12 12 9 3 3 All 1000000 1000000
8.	15,15	Predicted 0 All Actual 0 12493 12493 1 10599 10599 2 1206 1206 3 513 513	Predicted 0 All Actual 0 501209 501209 1 422498 422498 2 47622 47622 3 21121 21121

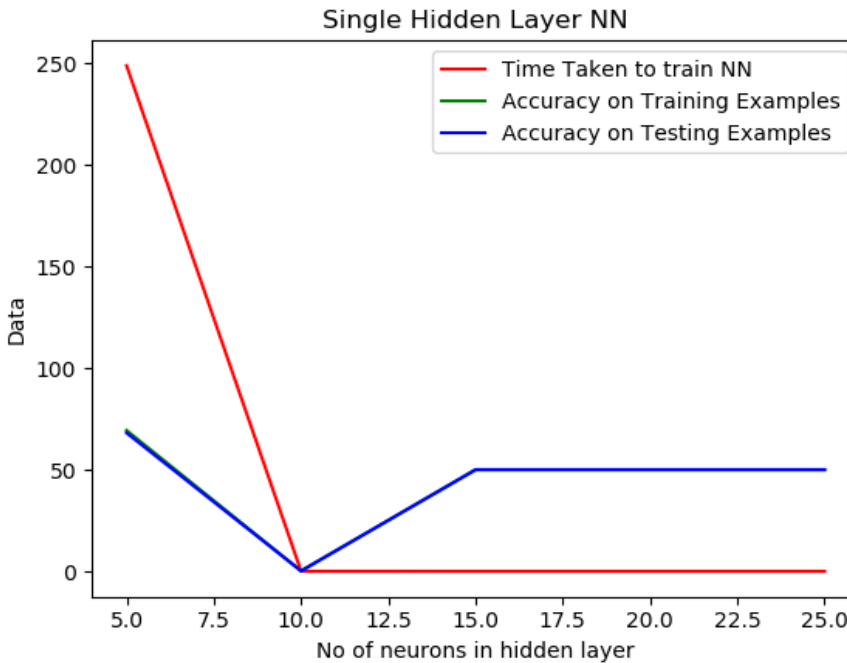
		4 93 93 5 54 54 6 36 36 7 6 6 8 5 5 9 5 5 All 25010 25010	4 3885 3885 5 1996 1996 6 1424 1424 7 230 230 8 12 12 9 3 3 All 1000000 1000000
9.	20,20	Predicted 0 All Actual 0 12493 12493 1 10599 10599 2 1206 1206 3 513 513 4 93 93 5 54 54 6 36 36 7 6 6 8 5 5 9 5 5 All 25010 25010	Predicted 0 All Actual 0 501209 501209 1 422498 422498 2 47622 47622 3 21121 21121 4 3885 3885 5 1996 1996 6 1424 1424 7 230 230 8 12 12 9 3 3 All 1000000 1000000
10	25,25	Predicted 0 All Actual 0 12493 12493 1 10599 10599 2 1206 1206 3 513 513 4 93 93 5 54 54 6 36 36 7 6 6 8 5 5 9 5 5 All 25010 25010	Predicted 0 All Actual 0 501209 501209 1 422498 422498 2 47622 47622 3 21121 21121 4 3885 3885 5 1996 1996 6 1424 1424 7 230 230 8 12 12 9 3 3 All 1000000 1000000

25 neuron hidden layer has such low accuracy because it didn't get time to learn. The tolerance reached $10e-6$ after a few iterations and hence learning rate decreased so fast that the system stopped learning. It is predicting 9 always maybe because the training examples on which it trained initially had output 9.

When I increased number of layers, the speed of convergence decreases faster than single layer system. Hence the tolerance reaches faster and so in most of the cases in this case the accuracy is 50% i.e. it is predicting 0 for all cases.

In some cases like hidden layer 1 with 10 neurons, the accuracy has increases as compared to part(c) and part(d).

So adaptive learning is good when data is not converging and it is oscillating back and forth. In that case reducing the learning rate will help converge.



Q2->(f)

ReLU converged very fast. However the results were not good because the function is $\max(0, z)$. So if by chance z comes to 0 then it will stop progressing as the gradient is 0.

Part of the problem was also because of tolerance issue as described in the above part

S.No.	Number of Neurons	Time Taken to train NN (s)	Training Accuracy	Testing Accuracy
1.	5	167	58.560575769692	58.4413
2.	10	250	0.4878048780487	0.4851
3.	15	330	49.952019192323	50.1209
4.	20	1	49.860055977608	50.062
5.	25	1	49.952019192323	50.1209
6.	5,5	200	42.35505797680	42.2861
7.	10,10	1	49.952019192323	50.1209
8.	15,15	1	49.952019192323	50.1209
9.	20,20	1	49.952019192323	50.1209
10.	25,25	1	49.952019192323	50.1209

S.No.	Number of Neurons	Confusion Matrix Training	Confusion Matrix Testing

1.	5	Predicted 0 1 All Actual 0 12491 2 12493 1 8444 2155 10599 2 823 383 1206 3 399 114 513 4 91 2 93 5 54 0 54 6 25 11 36 7 5 1 6 8 5 0 5 9 5 0 5 All 22342 2668 25010	Predicted 0 1 All Actual 0 500701 508 501209 1 338786 83712 422498 2 32643 14979 47622 3 17392 3729 21121 4 3870 15 3885 5 1989 7 1996 6 1027 397 1424 7 159 71 230 8 12 0 12 9 2 1 3 All 896581 103419 1000000
2.	10	Predicted 2 4 5 6 7 All Actual 0 1029 11 3286 5044 3123 12493 1 875 9 2793 4222 2700 10599 2 89 0 339 475 303 1206 3 44 0 124 203 142 513 4 8 0 16 49 20 93 5 5 0 12 22 15 54 6 1 0 6 21 8 36 7 0 0 1 5 0 6 8 0 0 1 2 2 5 9 0 0 2 1 2 5 All 2051 20 6580 10044 6315 25010	Predicted 2 4 5 6 7 All Actual 0 41059 419 132690 201152 125889 501209 1 33726 271 112744 168797 106960 422498 2 3759 35 12685 19026 12117 47622 3 1694 8 5573 8482 5364 21121 4 314 4 965 1802 800 3885 5 165 1 463 804 563 1996 6 117 0 395 553 359 1424 7 11 0 52 95 72 230 8 0 0 4 8 0 12 9 0 0 1 1 1 3 All 80845 738 265572 400720 252125 1000000
3.	15	Predicted 0 All Actual 0 12493 12493 1 10599 10599 2 1206 1206 3 513 513 4 93 93 5 54 54 6 36 36 7 6 6 8 5 5 9 5 5 All 25010 25010	Predicted 0 All Actual 0 501209 501209 1 422498 422498 2 47622 47622 3 21121 21121 4 3885 3885 5 1996 1996 6 1424 1424 7 230 230 8 12 12 9 3 3 All 1000000 1000000
4.	20	Predicted 0 1 All Actual 0 12392 101 12493 1 10521 78 10599 2 1198 8 1206	Predicted 0 1 All Actual 0 497209 4000 501209 1 419087 3411 422498 2 47272 350 47622

		3 510 3 513 4 90 3 93 5 53 1 54 6 36 0 36 7 6 0 6 8 5 0 5 9 5 0 5 All 24816 194 25010	3 20929 192 21121 4 3840 45 3885 5 1957 39 1996 6 1420 4 1424 7 225 5 230 8 11 1 12 9 3 0 3 All 991953 8047 1000000
5.	25	Predicted 0 All Actual 0 12493 12493 1 10599 10599 2 1206 1206 3 513 513 4 93 93 5 54 54 6 36 36 7 6 6 8 5 5 9 5 5 All 25010 25010	Predicted 0 All Actual 0 501209 501209 1 422498 422498 2 47622 47622 3 21121 21121 4 3885 3885 5 1996 1996 6 1424 1424 7 230 230 8 12 12 9 3 3 All 1000000 1000000
6.	5,5	Predicted 0 1 7 All Actual 0 116 12375 2 12493 1 120 10477 2 10599 2 8 1198 0 1206 3 5 508 0 513 4 3 90 0 93 5 0 54 0 54 6 0 36 0 36 7 0 6 0 6 8 0 5 0 5 9 0 5 0 5 All 252 24754 4 25010	Predicted 0 1 7 All Actual 0 4679 496466 64 501209 1 4256 418182 60 422498 2 520 47094 8 47622 3 232 20885 4 21121 4 64 3821 0 3885 5 13 1983 0 1996 6 16 1407 1 1424 7 2 228 0 230 8 1 11 0 12 9 0 3 0 3 All 9783 990080 137 1000000
7.	10,10	Predicted 0 All Actual 0 12493 12493 1 10599 10599 2 1206 1206 3 513 513 4 93 93 5 54 54 6 36 36 7 6 6 8 5 5 9 5 5 All 25010 25010	Predicted 0 All Actual 0 501209 501209 1 422498 422498 2 47622 47622 3 21121 21121 4 3885 3885 5 1996 1996 6 1424 1424 7 230 230 8 12 12 9 3 3 All 1000000 1000000
8.	15,15	Predicted 0 All	Predicted 0 All

		Actual 0 12493 12493 1 10599 10599 2 1206 1206 3 513 513 4 93 93 5 54 54 6 36 36 7 6 6 8 5 5 9 5 5 All 25010 25010	Actual 0 501209 501209 1 422498 422498 2 47622 47622 3 21121 21121 4 3885 3885 5 1996 1996 6 1424 1424 7 230 230 8 12 12 9 3 3 All 1000000 1000000
9.	20,20	Predicted 0 All Actual 0 12493 12493 1 10599 10599 2 1206 1206 3 513 513 4 93 93 5 54 54 6 36 36 7 6 6 8 5 5 9 5 5 All 25010 25010	Predicted 0 All Actual 0 501209 501209 1 422498 422498 2 47622 47622 3 21121 21121 4 3885 3885 5 1996 1996 6 1424 1424 7 230 230 8 12 12 9 3 3 All 1000000 1000000
10	25,25	Predicted 0 All Actual 0 12493 12493 1 10599 10599 2 1206 1206 3 513 513 4 93 93 5 54 54 6 36 36 7 6 6 8 5 5 9 5 5 All 25010 25010	Predicted 0 All Actual 0 501209 501209 1 422498 422498 2 47622 47622 3 21121 21121 4 3885 3885 5 1996 1996 6 1424 1424 7 230 230 8 12 12 9 3 3 All 1000000 1000000