REPORT

SVM CLASSIFIER

Flow of program:
We used SVC function of sklearn library to implement Support Vector Machine Classifier for above dataset.First we divided the data into 80:20 ratio of train and test datasets,normalised the data using train data.Then fitted the SVC model for different values of C as described above and produced train and test accuracies for all three linear ,quadratic and rbf kernels. Finally displayed tables using matplotlib library showing train and test accuracies for each C vs kernel.

After implementing theSVM classifier of scikit learn library using linear,quadratic and radial basis function kernels, following accuracies are observed.

Observed Accuracies :

Train Accuracies

Kernel → ---------- C value ↴	Linear	Quadratic	Radial Basis Function
0.0001	84.8404255319149	84.8404255319149	84.8404255319149
0.01	84.8404255319149	84.8404255319149	84.8404255319149
1	84.8404255319149	87.23404255319149	85.90425531914894
50	84.8404255319149	91.22340425531915	95.21276595744681
150	84.8404255319149	91.48936170212766	95.74468085106383

Test Accuracies

Kernel → ---------- C value ↴	Linear	Quadratic	Radial Basis Function
0.0001	86.17021276595744	86.17021276595744	86.17021276595744
0.01	86.17021276595744	86.17021276595744	86.17021276595744
1	86.17021276595744	86.17021276595744	86.17021276595744
50	86.17021276595744	72.3404255319149	78.72340425531915
150	86.17021276595744	76.59574468085106	81.91489361702128

Observed Best Test Accuracies:

For linear kernel,test accuracy is same for various values of C.
So best accuracy from above data is 86.17 for C=150
For quadratic kernel, test accuracy is highest and same for all values of C less than 1.
So best accuracy from above data is 86.17 for C=1.
For radial basis function kernel, test accuracy is highest and same for all values of C less than 1.
So best accuracy from above data is 86.17 for C=1.

Effect of C on accuracy:

Greater the value of C ,greater will be the train accuracy. So for C = infinity,the model tries to get a train accuracy of 100 and hence the model overfits for large values of C.Also the time of execution(fitting the data for a classifier) increases exponentially for large values of C(say greater than 100).
The dataset provided has 469 samples out of which 400 samples are of one category and 69 others are of the other class. So the dataset is completely biased and hence

Observations:

- ★ For linear kernel,for all values of C ,the test accuracy is same which suggests that as the data is biased the model is assigning a single label to all the samples and hence the accuracy is not changing with C.This suggests that even though C is very high due to less complexity of linear kernel ,100% training accuracy can never be reached for given dataset.
- ★ For quadratic kernel, for all values of C less than 1 ,test accuracy is same which suggest that model is assigning a single label to all samples.For C value greater than 1,model complexity increases and tries to overfit data there by increasing training accuracy and gradually changing test accuracy. From above table,test accuracy trends as constant till C=1 and then suddenly decreases and then gradually increases for large values of C.
- ★ For rbf kernel,pattern of change in test accuracy is similar to that of quadratic kernel.It remains constant in the beginning till C value around 1,then suddenly decreases and then gradually increases for larger values of C.

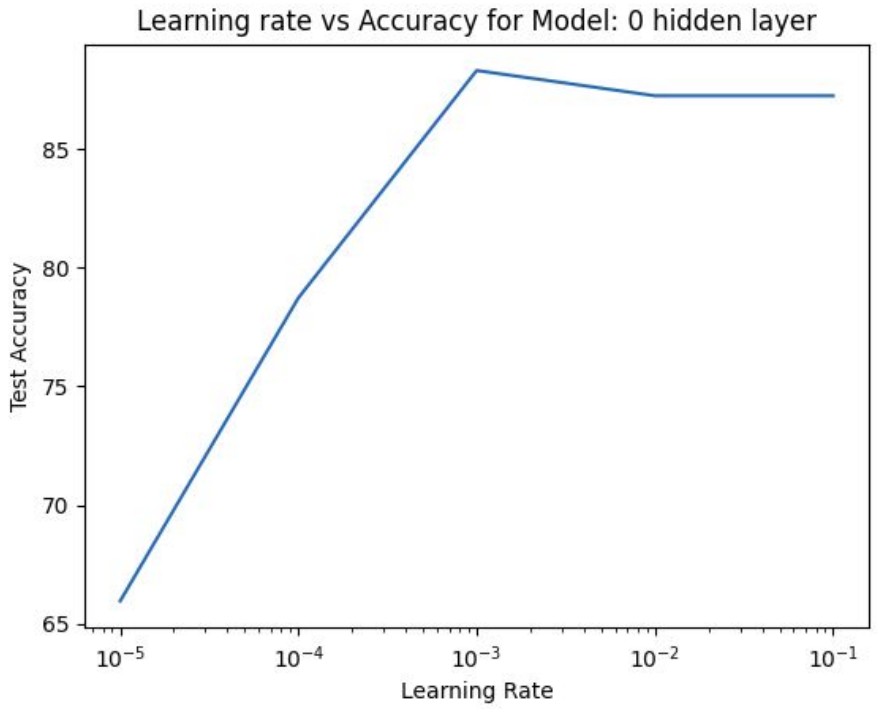
So to prove this variation in trend ,we chose C values as 0.0001,0.01,1(Till which accuracies remain constant for all kernels),50,150(To demonstrate variation in change of test accuracies for quadratic and rbf kernels).

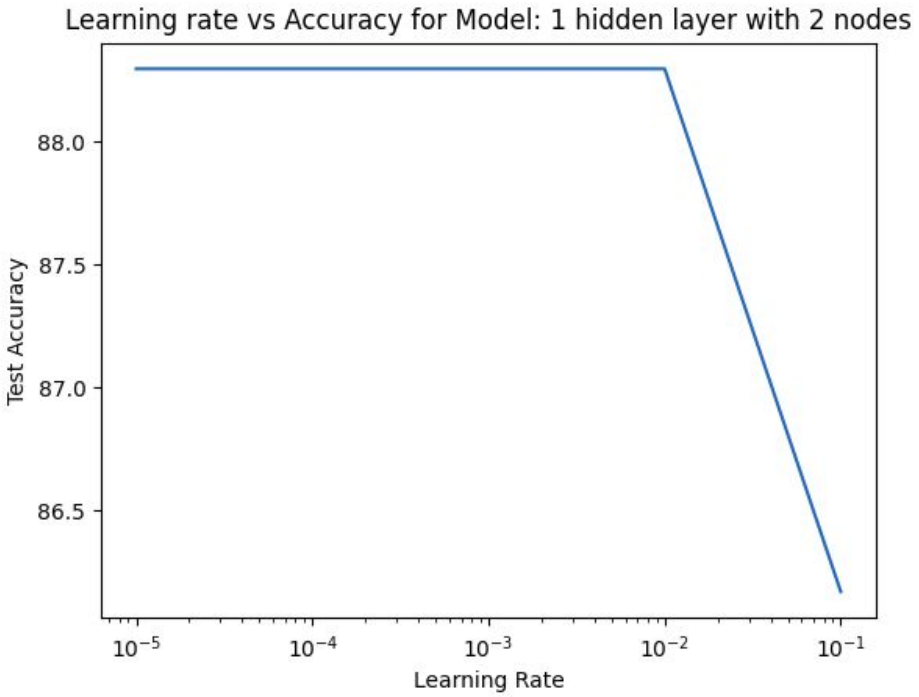
Out of linear,quadratic and rbf kernels, the rbf kernel is better to fit the given dataset as the model tries to reach higher training accuracies for greater values of C and provides better test accuracies than other kernels.

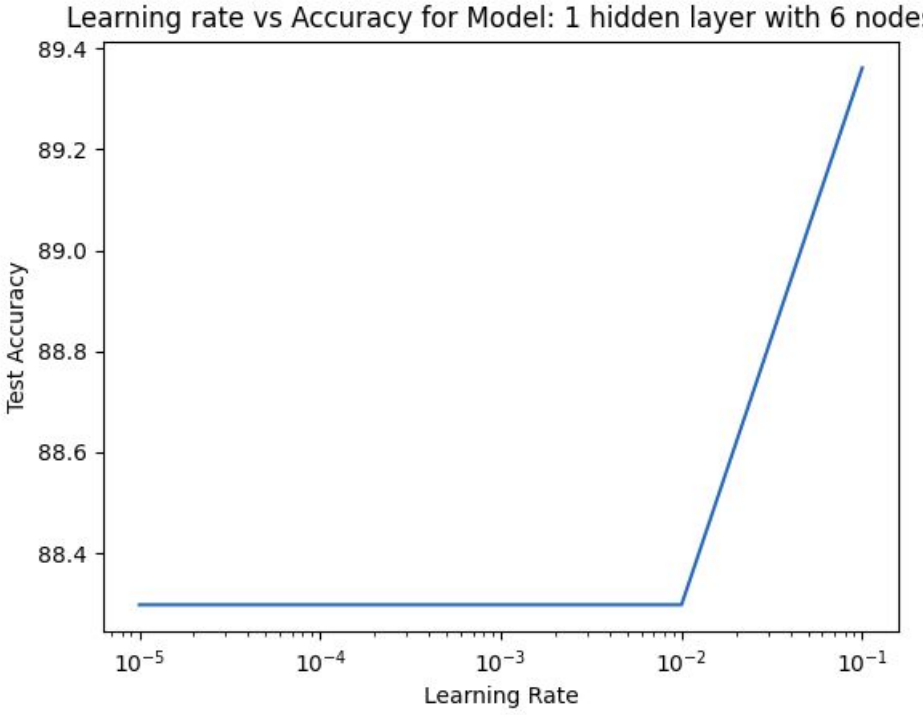
MLP CLASSIFIER

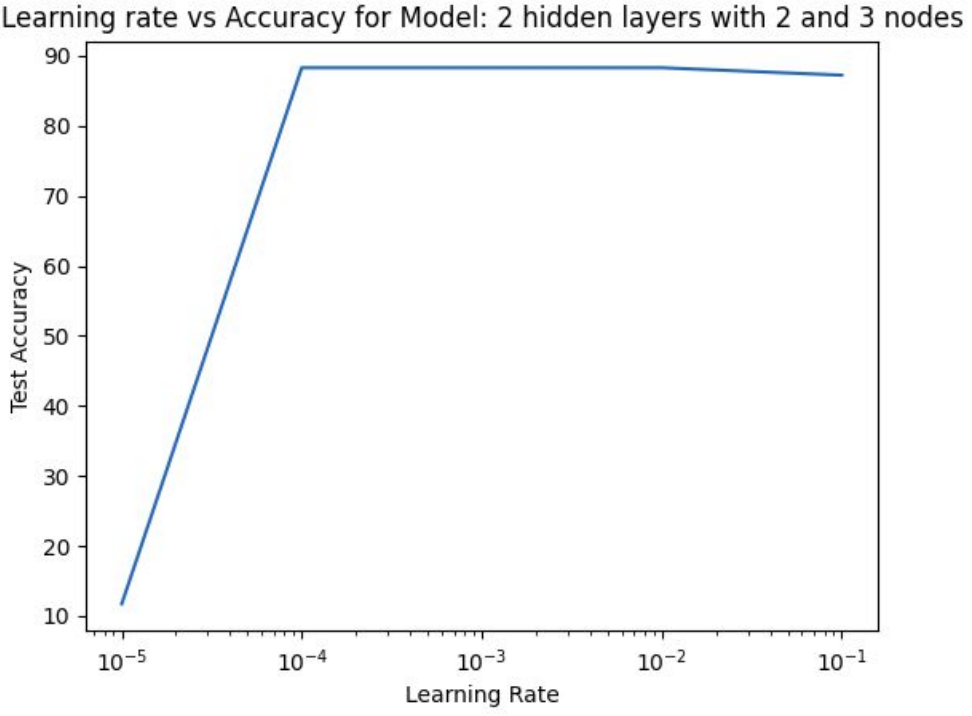
In the MLP classifier for the given dataset the number of nodes in input layer is 16 and in output layer is 1. This is because the number of features excluding the target attribute is 16 and as the target attribute has 2 classes, namely True or False, so only 1 node in the output layer.

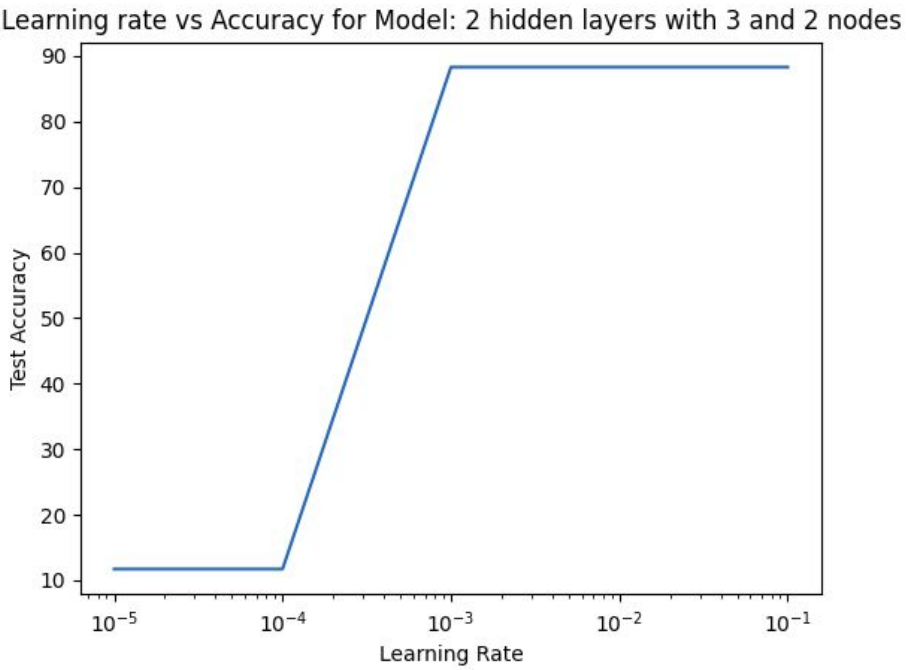
Plots for Learning rate vs Accuracy for different Models:





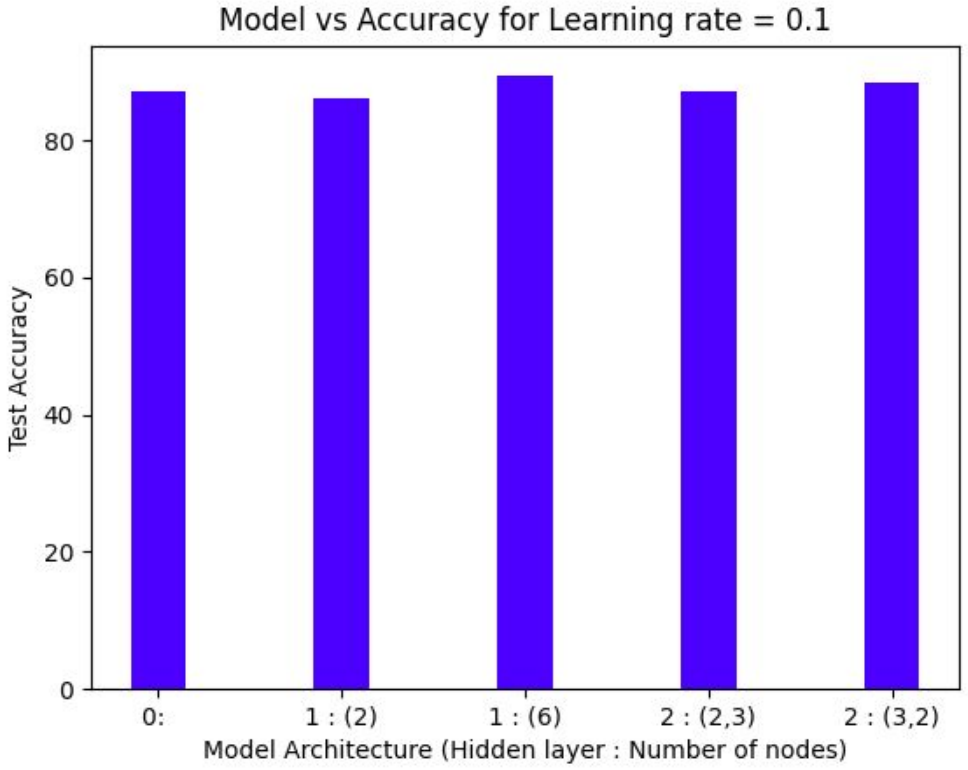

Learning rate vs Accuracy for Model: 1 hidden layer with 6 nodes
89.4
89.2
89.0
88.8
88.6
88.4
Test Accuracy
10^{-5}
10^{-4}
10^{-3}
10^{-2}
10^{-1}
Learning Rate

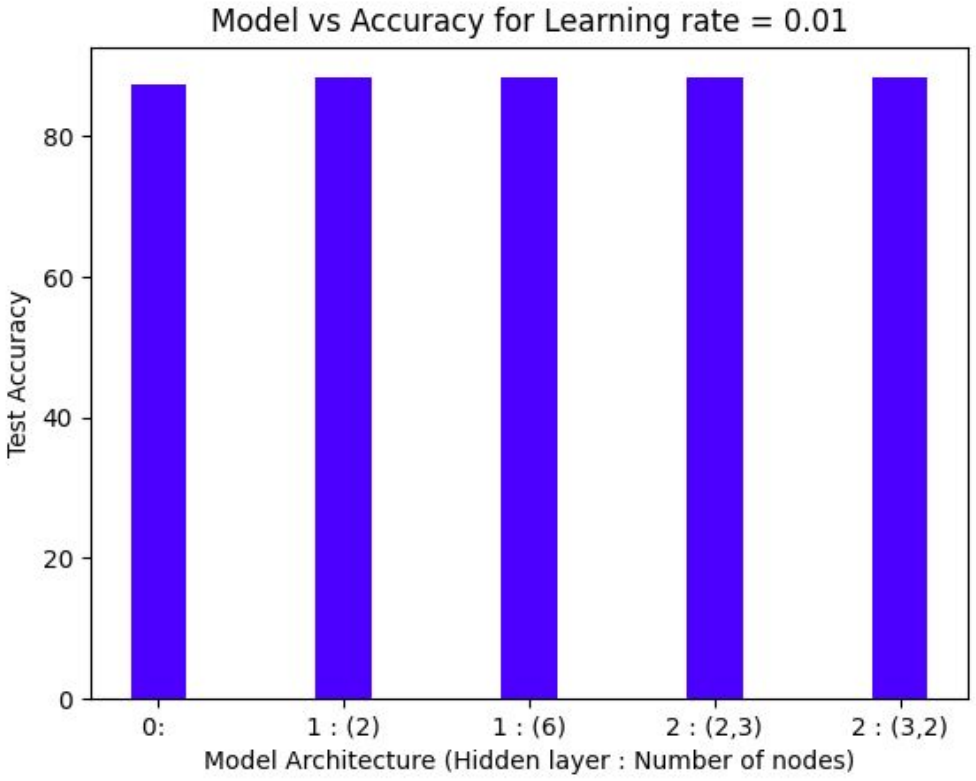

Learning rate vs Accuracy for Model: 2 hidden layers with 2 and 3 nodes
90
80
70
60
50
40
30
20
10
Test Accuracy
10^{-5}
10^{-4}
10^{-3}
10^{-2}
10^{-1}
Learning Rate

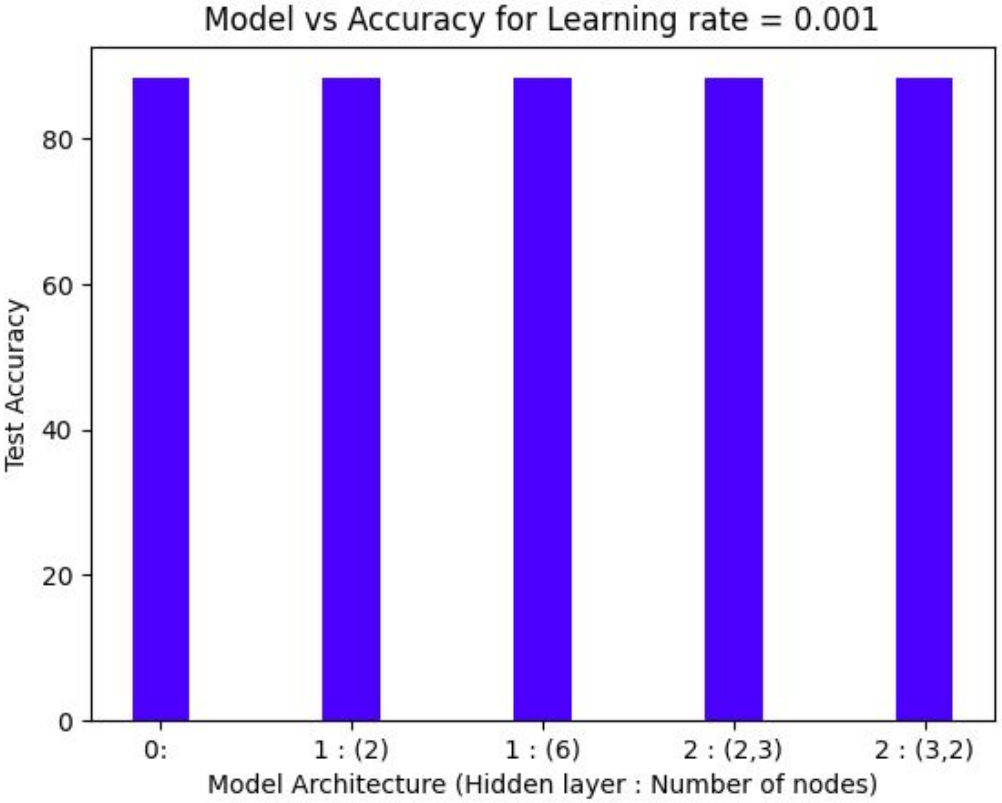


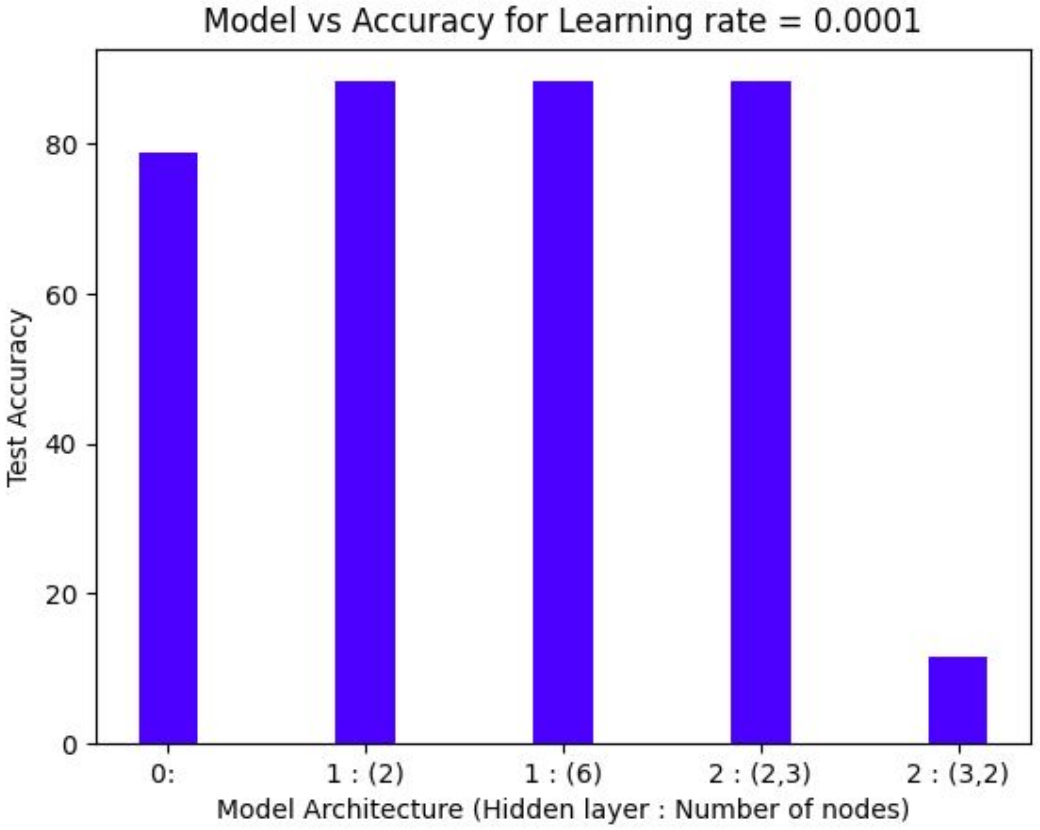
Plots for Model vs Accuracy for different Learning rates:

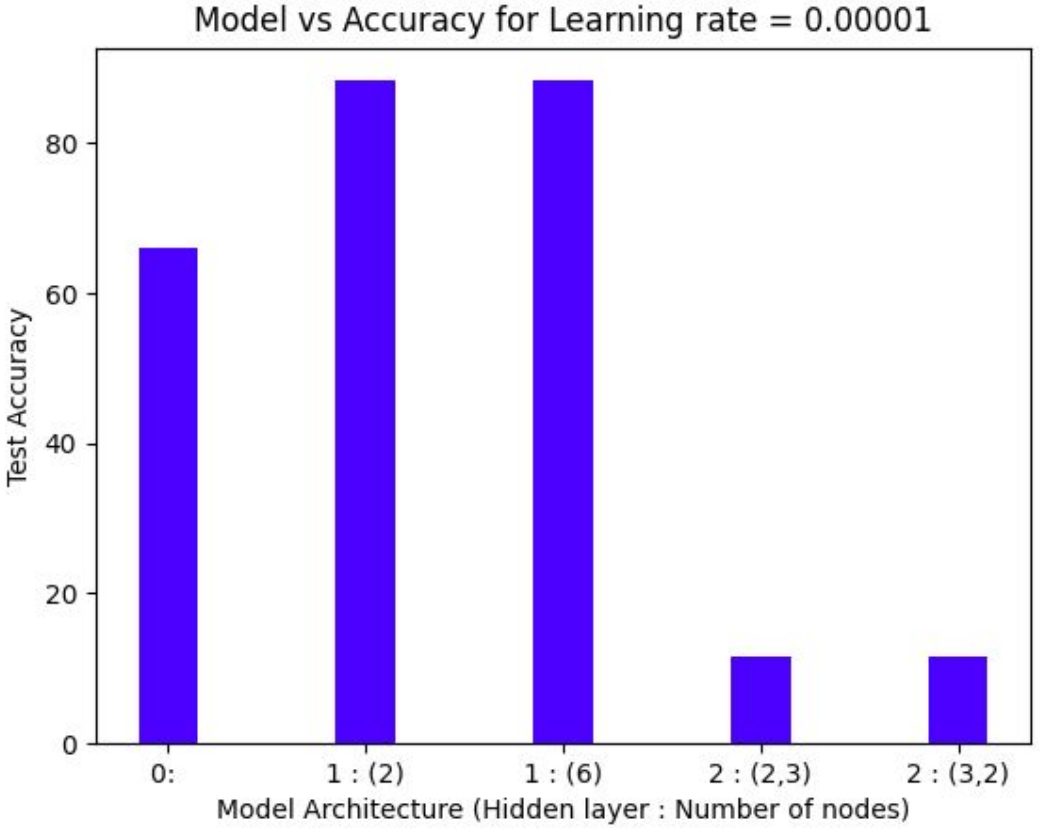
Format for x-axis: Hidden Layer Size: Number of nodes




Model vs Accuracy for Learning rate = 0.01
Test Accuracy
0
20
40
60
80
0:
1 : (2)
1 : (6)
2 : (2,3)
2 : (3,2)
Model Architecture (Hidden layer : Number of nodes)


Model vs Accuracy for Learning rate = 0.001
Test Accuracy
0
20
40
60
80
0:
1 : (2)
1 : (6)
2 : (2,3)
2 : (3,2)
Model Architecture (Hidden layer : Number of nodes)


Model vs Accuracy for Learning rate = 0.0001
Test Accuracy
0
20
40
60
80
0:
1 : (2)
1 : (6)
2 : (2,3)
2 : (3,2)
Model Architecture (Hidden layer : Number of nodes)


Model vs Accuracy for Learning rate = 0.00001
Test Accuracy
0
20
40
60
80
0:
1 : (2)
1 : (6)
2 : (2,3)
2 : (3,2)
Model Architecture (Hidden layer : Number of nodes)

Remarks on the plots obtained:

In general as the learning rate is lesser the model should slowly get trained on the training data set, so it usually takes many iterations for the MLP classifier to converge but gives better accuracy. But for higher learning rates it converges early but the accuracy will be slightly lower, as the data was biased (85% of the target attribute was from '*False*' class), we can observe that here most of the graphs are following the general trend. But because of the aforementioned bias the best accuracy was found near 0.1 learning rate, this is not exactly surprising as the other models also show significantly high accuracy for 0.1 learning rate.

As the model architecture gets complex it can be seen that the accuracies are slightly higher for learning rates 0.1, 0.01, 0.001, for the others it is different because of time vs accuracy trade off. The increase in the number of hidden layers is clearly a better method to get good accuracy but it obviously takes more time and iterations to train the set. These kinds of trade offs are clearly observable in the plots.

If clearly observed then some irregularities in the graphs can be observed like very low test accuracy for lower learning rate for complex models. This kind of behaviour is because of no chance of learning anything from the model as there is an accuracy vs time trade off. Even though the Max number of iterations and the difference in losses (the parameter 'tol' in MLP classifier) were altered by a significant amount the change in accuracy was very less but it took a lot of time to execute. This is clearly due to the low learning rate for a complex model like 2 hidden layers. All the exact values for accuracies have been reported while running the code. Finally as we are training the data set on the train data we are expecting a general curve, but in reality we have found similar plots with some differences.

Final Result:

Highest found Test Accuracy = 89.36%
Best Found Model: 1 hidden layer with 6 nodes
Corresponding hyper parameters:
-> Hidden Layer Size = [6] (meaning 1 hidden layer with 6 nodes0
-> Learning rate = 0.1
-> Optimizer = Stochastic Gradient Descent
-> Activation function = logistic
-> Max Iterations = 500
-> Batch Size = 1 (as the optimizer is SGD)
-> alpha = 0.0001 (default)
Some other parameters are reported while running the code.

Comparing the performances of both the classifiers:

Best test accuracy incase of SVM classifier found for following C values:
For C = 150 for Linear Kernel with a test accuracy of 86.17%
For C = 1 for Quadratic Kernel with a test accuracy of 86.17%
For C = 1 for Radial Basis Kernel with a test accuracy of 86.17%

Best Model architecture incase of MLP classifier is 1 hidden layer with 6 nodes with 0.1 learning rate, and having a test accuracy of 89.36%.

So after comparison we have found that MLP classifier is giving better results as compared to SVM classifier.