MACHINE LEARNING

Assignment- 3B
Group No- 14
Roll numbers- 19CS10037,19CS30049
Names- K Sai Santhoshi Niharika, Surkanti Akshitha

Dataset info:

- The data used in training was provided in .data format.
- The data set is related to letter recognition.
- The data had the following attributes:
    - lettr - capital letter (26 values from A to Z)
    - x-box - horizontal position of box (integer)
    - y-box - vertical position of box (integer)
    - width - width of box (integer)
    - high - height of box (integer)
    - onpix - total # on pixels (integer)
    - x-bar - mean x of on pixels in box (integer)
    - y-bar - mean y of on pixels in box (integer)
    - x2bar - mean x variance (integer)
    - y2bar - mean y variance (integer)
    - xybar - mean x y correlation (integer)
    - x2ybr - mean of x * x * y (integer)
    - xy2br - mean of x * y * y (integer)
    - x-ege - mean edge count left to right (integer)
    - xegvy - correlation of x-ege with y (integer)
    - y-ege - mean edge count bottom to top (integer)
    - yegvx -correlation of y-ege with x (integer)

The flow of events:

- Read the data from letter_recognition.data using the pandas library.
- Split the data into train and test in the ratio of 80: 20 as mentioned in the dataset information.
    - Train samples - 16,000
    - Test samples - 4,000

- All the numerical features in the dataset are scaled.
- We build the MLP classifier for
    - 0 hidden layer
    - 1 hidden layer with 2 nodes
    - 1 hidden layer with 6 nodes
    - 2 hidden layers with 2 and 3 nodes respectively
    - 2 hidden layers with 3 and 2 nodes respectively
- After initializing, we give the data to train the neural network.Then use the trained network to predict and calculate the accuracy using the accuracy function which takes a confusion matrix as input.
- For each model, we have printed accuracies for different learning rates.
- For each learning rate, a plot of Model vs accuracy is also printed.In that plot on x-axis
    - 1 - 0 hidden layer
    - 2 - 1 hidden layer with 2 nodes
    - 3 - 1 hidden layer with 6 nodes
    - 4 - 2 hidden layers with 2 and 3 nodes respectively
    - 5 - 2 hidden layers with 3 and 2 nodes respectively
- The processed data was split into train and test datasets in a 70:30 ratio.
- Trained the data using the Naive Bayes classifier and computed the accuracy.
- Trained the data using the Naive Bayes classifier with the Laplace correction and computed the accuracy.

Help Function :

- **accuracy :** It takes a confusion matrix as input and uses it to determine the accuracy which is measured as the total number of correct predictions divided by the total number of predictions.

1Q) No. of nodes in input layer - 16
No. of nodes in output layer - 26
Justification : As the no. of attributes in the given dataset other than letters are taken as input , no. of nodes in input layer is 16.And as the alphabets in the dataset to be recognised are 26 , no. of nodes in output layer are 26.
Hyperparameters we chose are learning rate, activation function.These are not actually required in the learning process but their presence affects the learning process and the accuracy produced.

Best learning rate for all the models is 0.01.
Best activation function is “logistic”.

For choosing best activation function for the hidden layers, we have checked the accuracy with learning rate 0.01

Activation	**Model**	
	1	2
logistic	0.76825	0.36125
relu	0.76825	0.345
tanh	0.76575	0.33975
identity	0.768	0.35575

So we opted for “logistic” activation

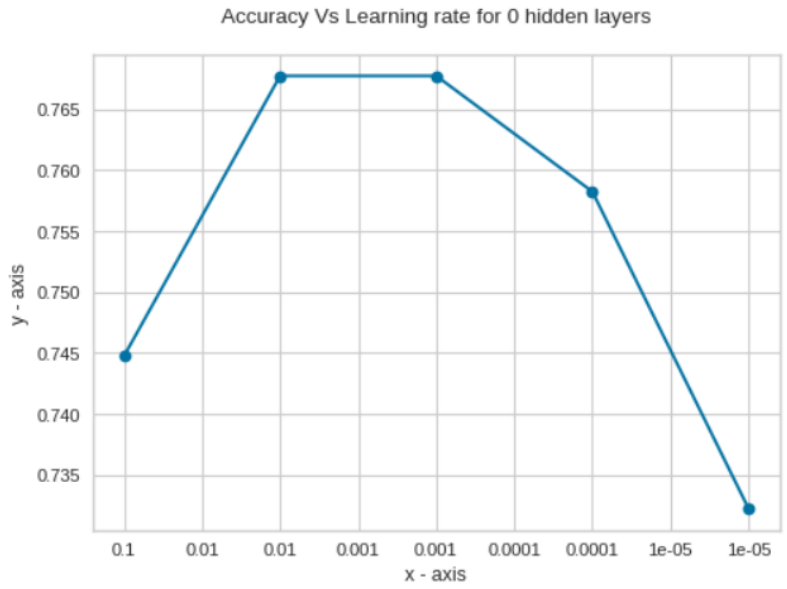
Results:

Accuracy of different models varying learning rates

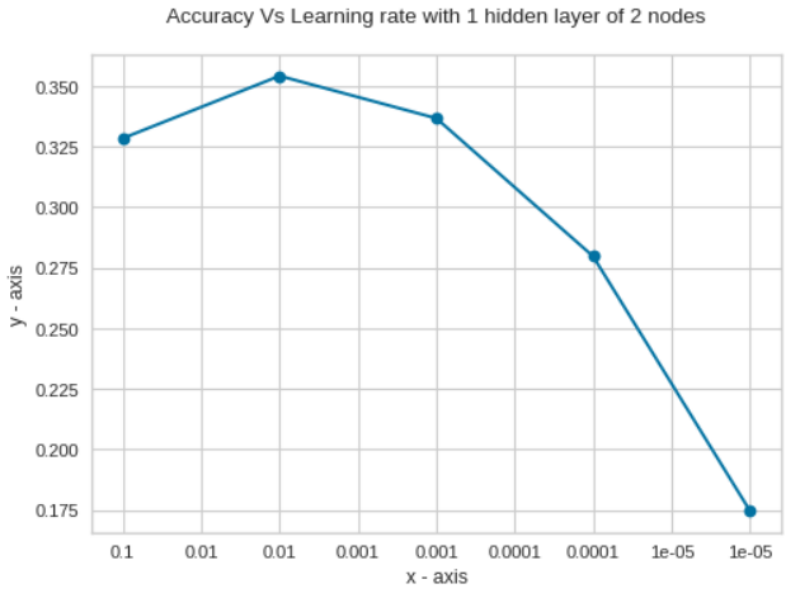
Hidden layers	**Nodes**	**Learning Rates**				
		0.1	**0.01**	**0.001**	**0.0001**	**0.00001**
0	0	0.74475	0.76775	0.76775	0.75825	0.73225
1	2	0.3285	0.35425	0.33675	0.27975	0.17475
1	6	0.68575	0.6995	0.679	0.64	0.55125
2	2,3	0.33025	0.3725	0.3395	0.2695	0.0325
2	3,2	0.36175	0.3815	0.37425	0.293	0.04475

Plots :

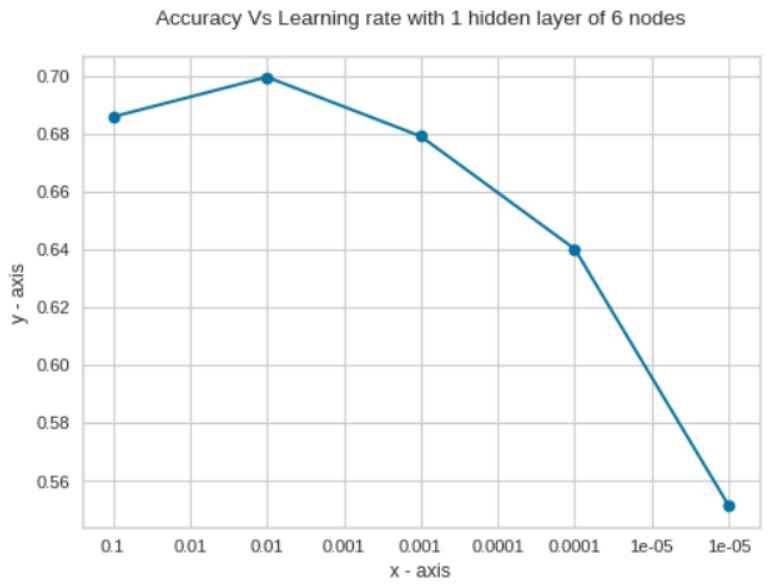
- **Accuracy vs Learning Rate**
    - Model 1 :



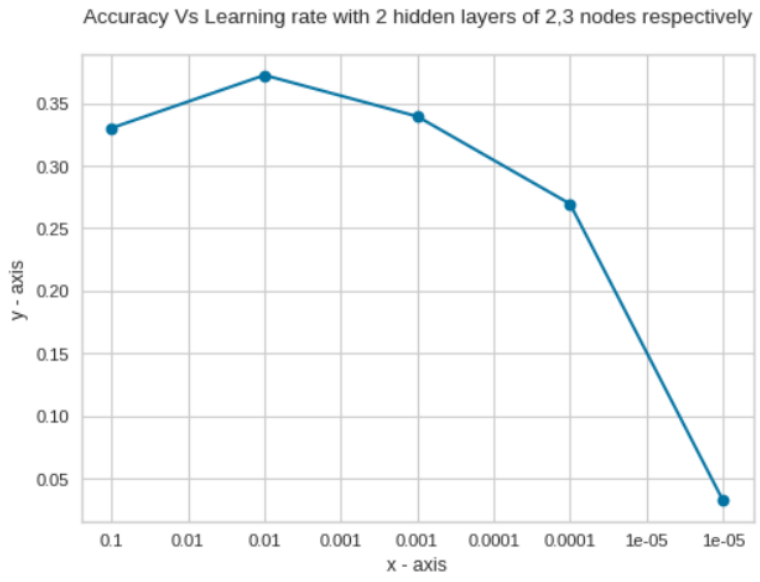
    - Model 2:



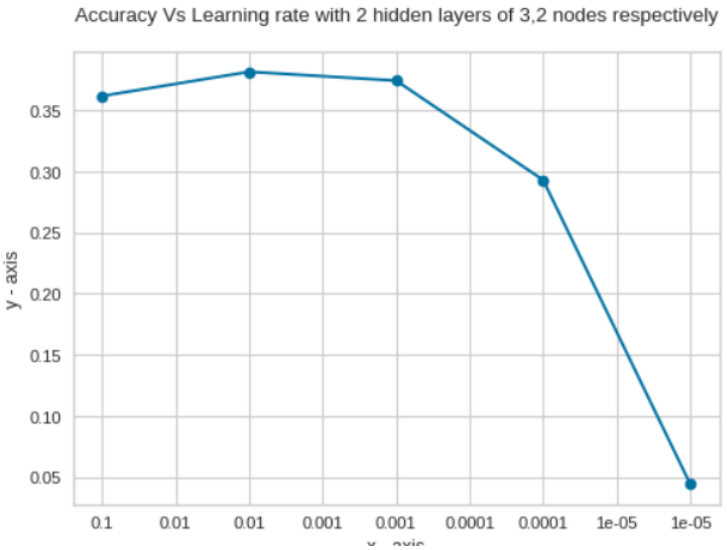
- Model 3:



- Model 4:

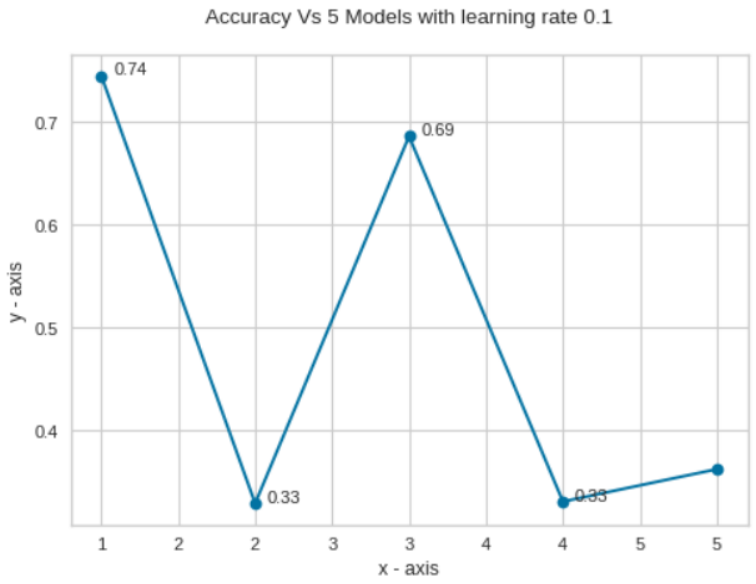


- Model 5:

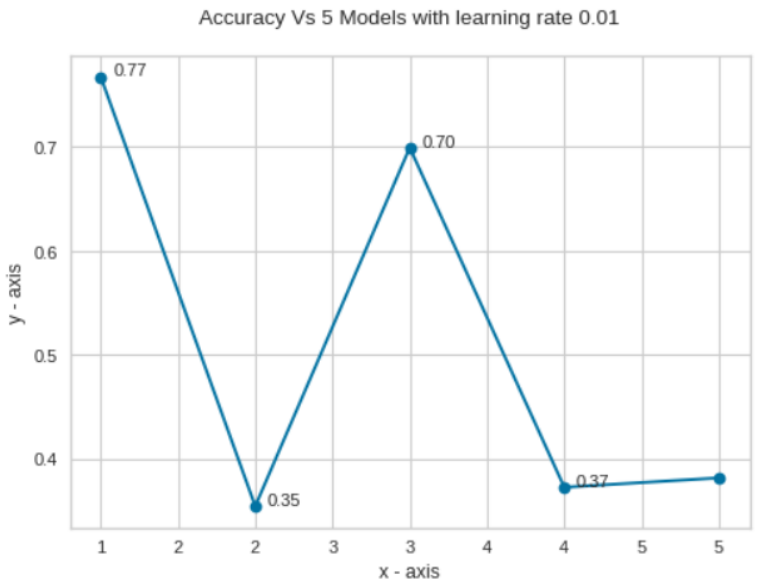


Best learning rate - 0.01

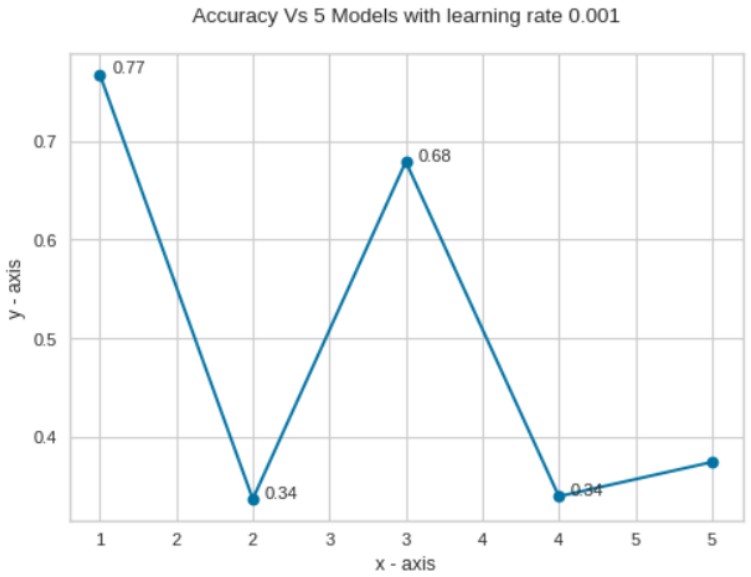
- **Model vs Accuracy with learning rate**
  - 0.1



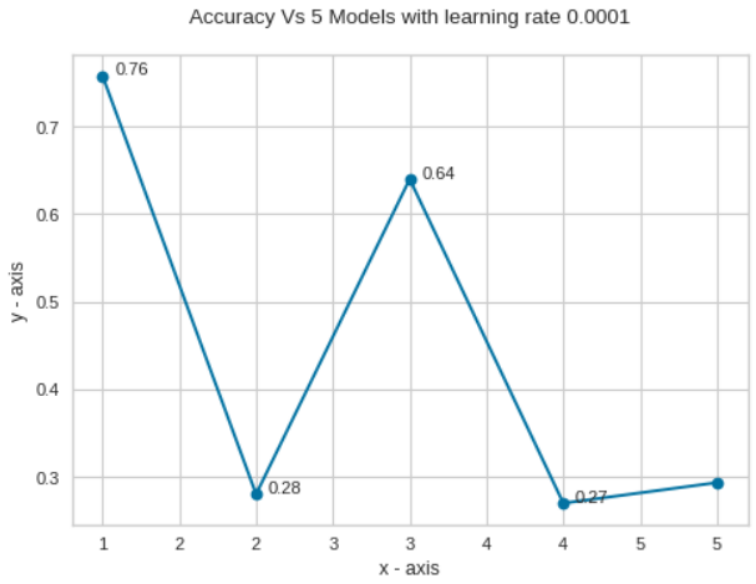
- 0.01


Accuracy Vs 5 Models with learning rate 0.01
0.77
0.70
0.35
0.37
y - axis
x - axis
0.4
0.5
0.6
0.7
1
2
2
3
3
4
4
5
5

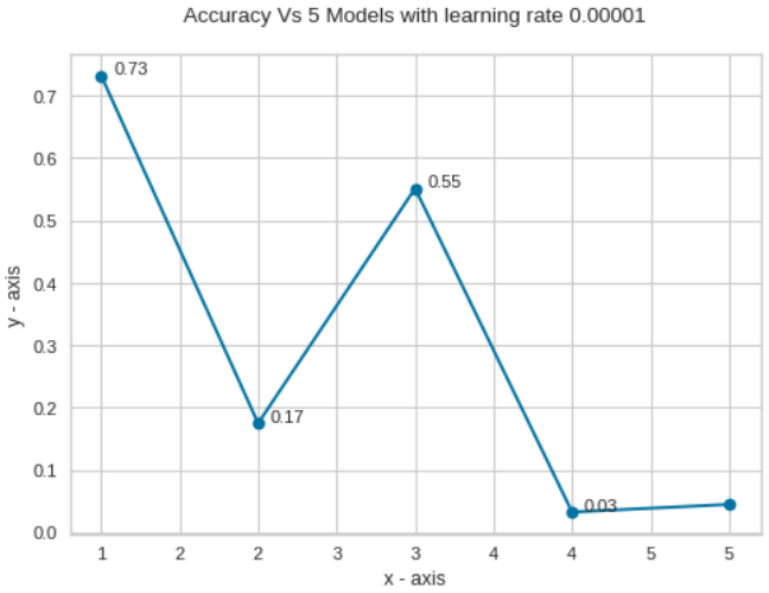
- 0.001


Accuracy Vs 5 Models with learning rate 0.001
0.77
0.68
0.34
0.34
y - axis
x - axis
0.4
0.5
0.6
0.7
1
2
2
3
3
4
4
5
5

- 0.0001



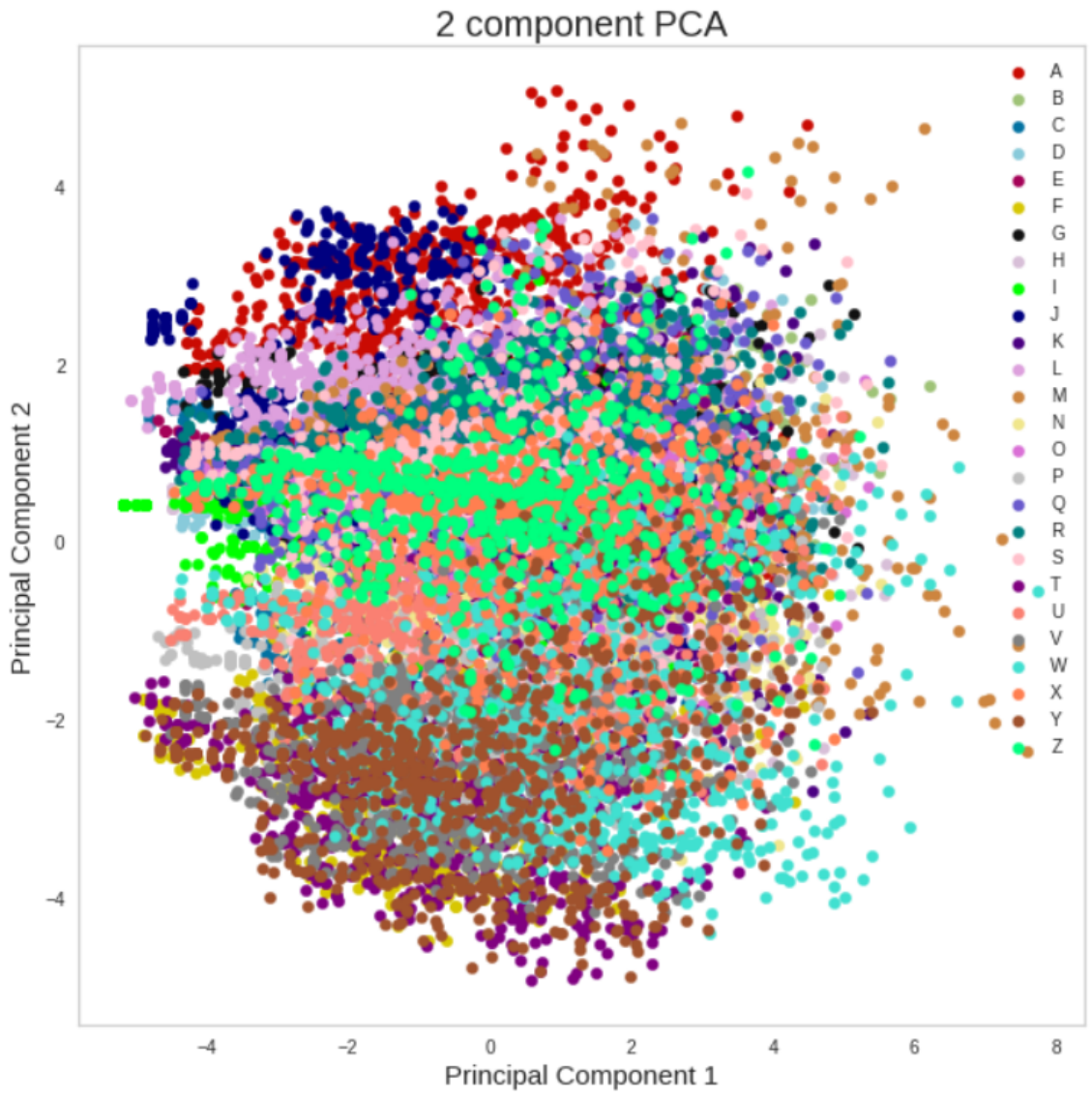
- 0.00001



4Q) The best architecture is of 0 hidden layers where the accuracy is highest of all other architectures mentioned in the assignment.
Hyperparameters
Learning rate - 0.01
Activation function - logistic.(justification provided above in Q1)

Justification :A higher learning rate makes the model learn faster, but it may miss the minimum loss function and only reach the surroundings of it. A lower learning rate gives a better chance to find a minimum loss function.May be as a tradeoff we got 0.01 as the best learning rate.

Plot of reduced dimensional data in 2D Plane using PCA



After applying a multi-layer perceptron classifier on this reduced feature space, the accuracy obtained is 15.925 which is drastically low compared to the output accuracy obtained in step-2 i.e, 76.775.
Justification : PCA is an algorithm that does not consider the response variable / prediction target into account. PCA will treat the feature that has large variance as important features, but the feature that has large variance can have nothing to do with the prediction target.

How to run the code :

- Download ipynb file and data set into your local machine
- Upload the above-downloaded files into the drive
- Open them in google collab from your drive
- Run every part in the notebook from the beginning to the end and the later parts will give you the desired results.
- U will observe the output of each part individually