Report Part 1

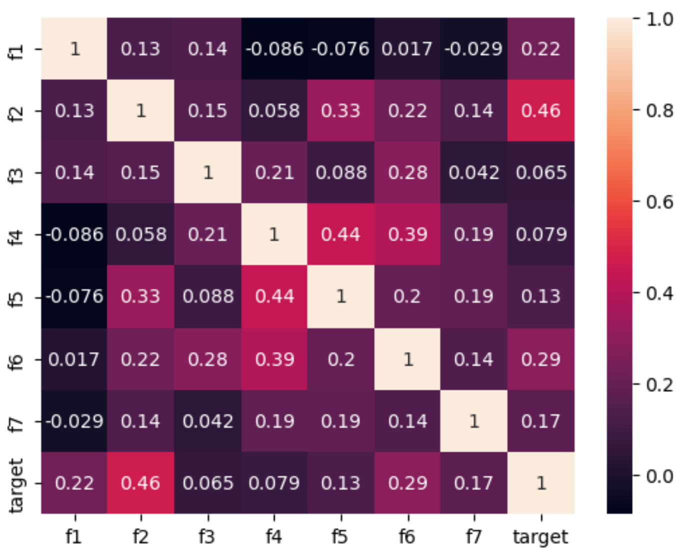
Question 1

The dataset given is a collection of values, where each value is of two classes i.e.; 1 or 0. The type of data present is of numerical data. Each data point has seven features and one target. The model which is developed would be a binary classifier. There were 766 entries, out of which 6 were removed as these entries were improper, i.e.; where expecting numerical data but had categorical or character data.

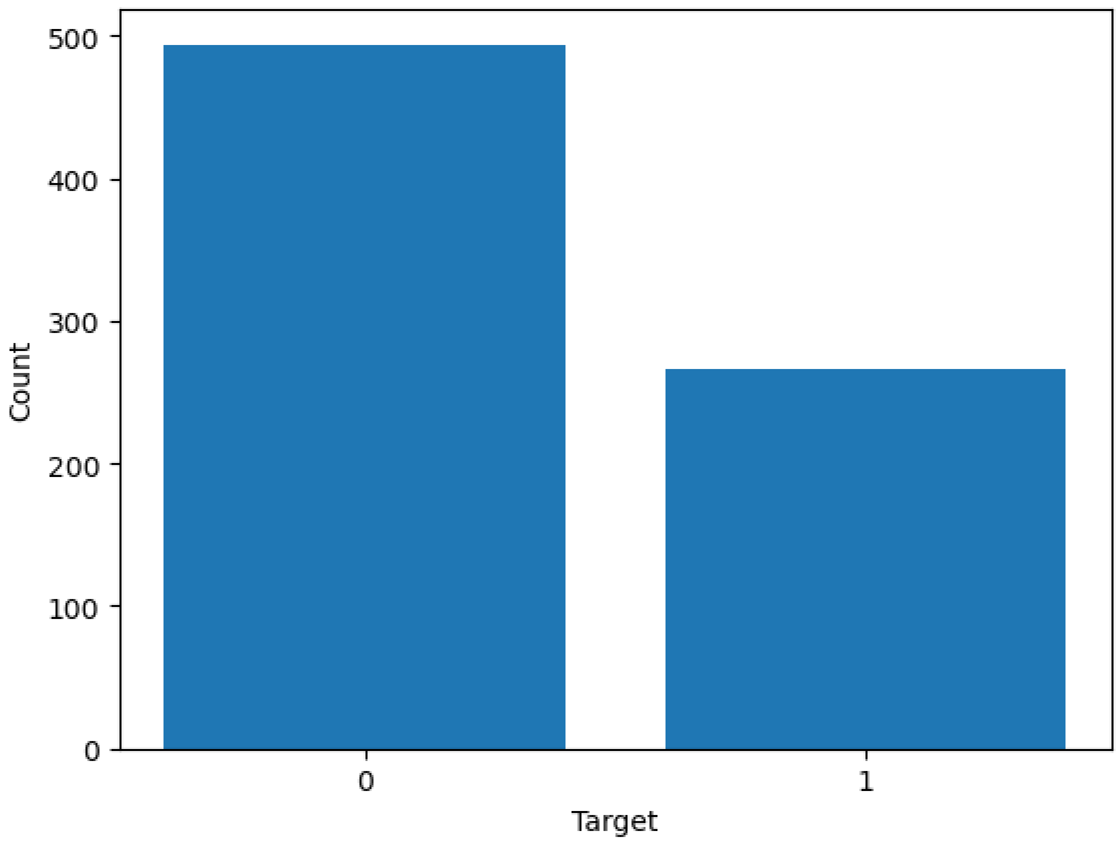
The following table has statistics of the data.

	F1	F2	F3	F4	F5	F6	F7	target
Count	760	760	760	760	760	760	760	760
Mean	3.83	120.97	69.12	20.61	80.23	32	0.47	0.35
Std	3.37	32.02	19.45	15.96	115.58	7.9	0.33	0.48
Min	0	0	0	0	0	0	0.078	0
25%	1	99	63.5	0	0	27.3	0.24	0
50%	3	117	72	23	36	32	0.38	0
75%	6	141	80	32	128.25	36.6	0.63	1
Max	17	199	122	99	846	67.1	2.42	1

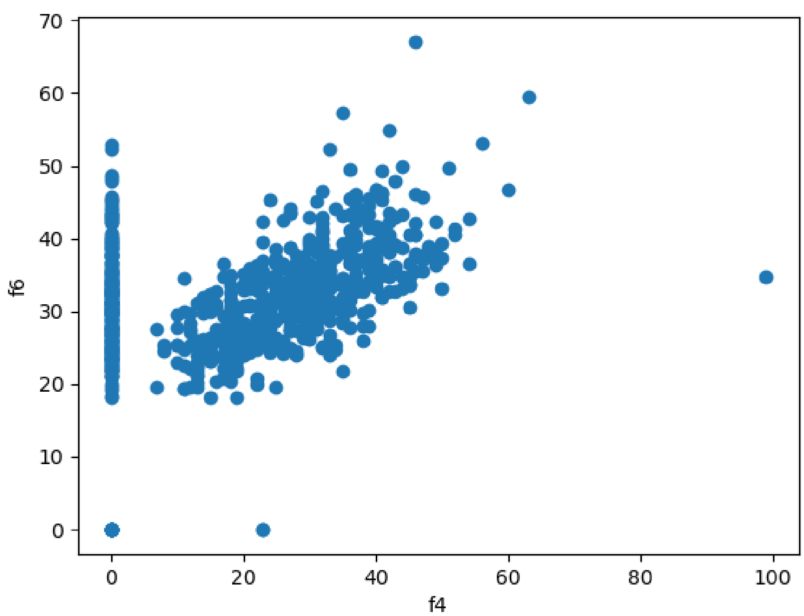
Question 2



The above shows the co-relation of columns with other columns, from which we can see that they have very less co-relation. This means we would need to use all the features to get the best accuracy.



The above graph is we can see that target 0 has almost twice the number as target 1. We need to make sure that we see similar ration when we split the dataset into test and train so the model does not skew towards target 0.



The above graph is look at the relation between f4 and f6 features. We can see some co-relation but due to the outliers we can get very low co-relation.

Question 3

In preprocessing we have deleted rows with missing values and wrong values i.e.; where we were expecting numerical data but had categorical or character data. We have also scaled the data to have mean close to 0 and standard deviation close to 1, to help with processing speed due to easier calculation by scaling.

	f1	f2	f3	f4	f5
count	7.600000e+02	7.600000e+02	7.600000e+02	7.600000e+02	7.600000e+02
mean	-7.756953e-17	2.278879e-17	-2.836912e-16	6.719771e-17	6.719771e-18
std	1.000659e+00	1.000659e+00	1.000659e+00	1.000659e+00	1.000659e+00
min	-1.140270e+00	-3.780041e+00	-3.556770e+00	-1.285961e+00	-6.946361e-01
25%	-8.428760e-01	-6.865065e-01	-2.891809e-01	-1.285961e+00	-6.946361e-01
50%	-2.480889e-01	-1.240456e-01	1.482128e-01	1.562691e-01	-3.829623e-01
75%	6.440919e-01	6.259022e-01	5.598776e-01	7.206199e-01	4.157018e-01
max	3.915421e+00	2.438276e+00	2.721117e+00	4.921898e+00	6.629698e+00

	f6	f7
count	7.600000e+02	7.600000e+02
mean	-1.623701e-16	1.615667e-16
std	1.000659e+00	1.000659e+00
min	-4.053275e+00	-1.190303e+00
25%	-5.951826e-01	-6.911439e-01
50%	1.666711e-04	-2.943761e-01
75%	5.828490e-01	4.645270e-01
max	4.446286e+00	5.862677e+00

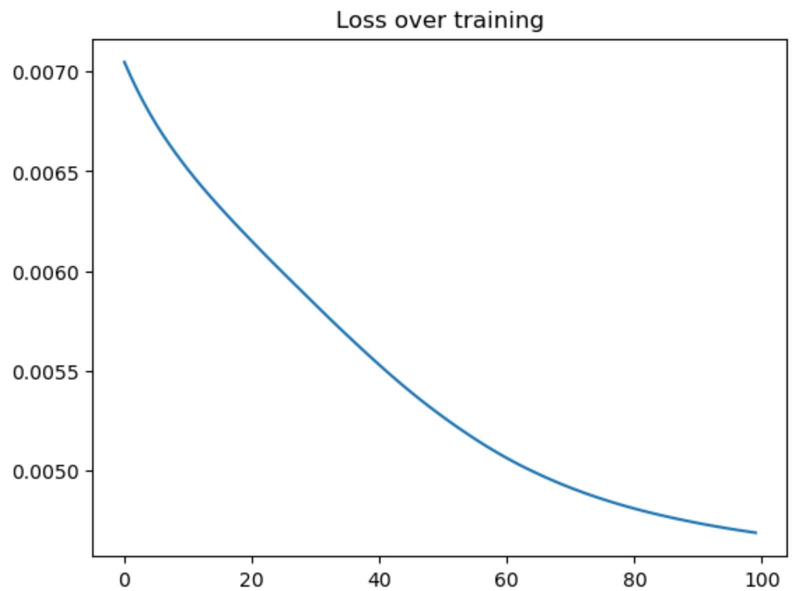
Question 4

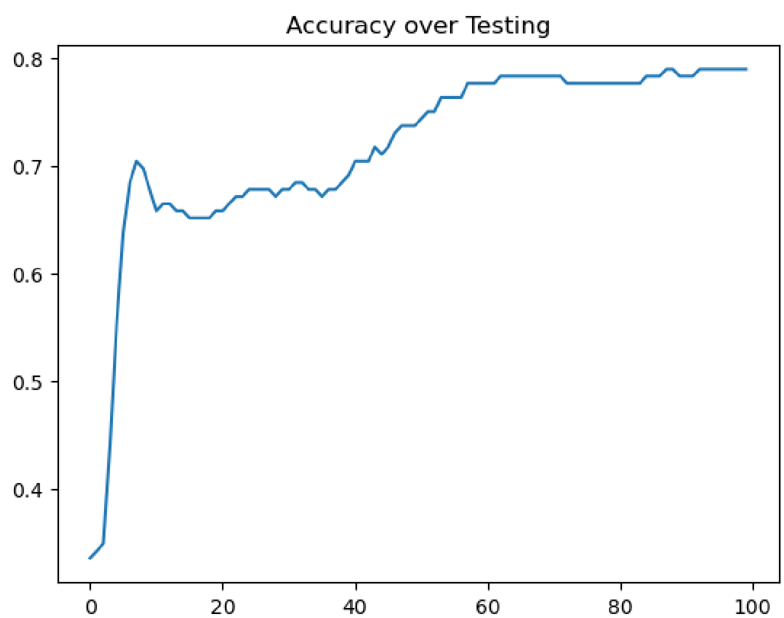
```
class NeuralNetwork(nn.Module):
    def __init__(self):
        super(NeuralNetwork, self).__init__()
        self.fc1 = nn.Linear(input_size, 128)
        self.fc2 = nn.Linear(128, 64)
        self.fc3 = nn.Linear(64, output_size)
        self.relu = nn.ReLU()
        # self.sigmoid = nn.Sigmoid()

    def forward(self, x):
        x = self.fc1(x)
        x = self.relu(x)
        x = self.fc2(x)
        x = self.relu(x)
        x = self.fc3(x)
        # x = self.sigmoid(x)
        return x
```

In architecture structure of Neural network used has 3 fully connected layers. First layer has 7 inputs that is number of features of the dataset and 128 output nodes. In second layer 128 inputs and 64 output nodes.

The last layer has 64 inputs and 1 output node. Each layer has Rectified Linear Unit as activation function.

1. Provide graphs that compares test and training accuracy on the same plot, test and training loss on the same plot. Thus, in total two graphs with a clear labeling.





Report for Part 2

Question 1

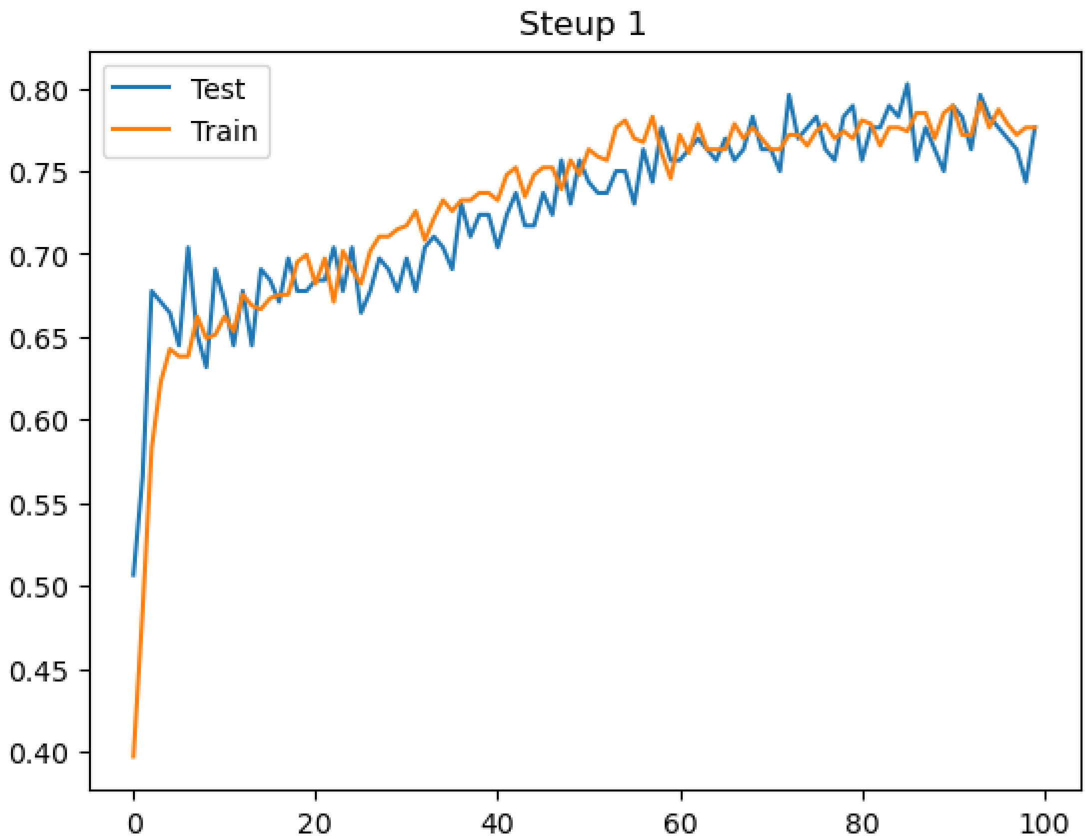
	Setup1	Test Accuracy	Setup2	Test Accuracy	Setup3	Test Accuracy
Dropout	0.1	77.63	0.2	77.63	0.3	73.03
Optimizer	SGD		SGD		SGD	
Activation Function	ReLU		ReLU		ReLU	
Initializer	Xavier Normal		Xavier Normal		Xavier Normal	

	Setup4	Test Accuracy	Setup5	Test Accuracy	Setup6	Test Accuracy
Dropout	0.1	70.39	0.1	67.11	0.1	73.68
Optimizer	Adam		RMS prop		Adagrad	
Activation Function	ReLu		ReLu		ReLu	
Initializer	Xavier Normal		Xavier Normal		Xavier Normal	

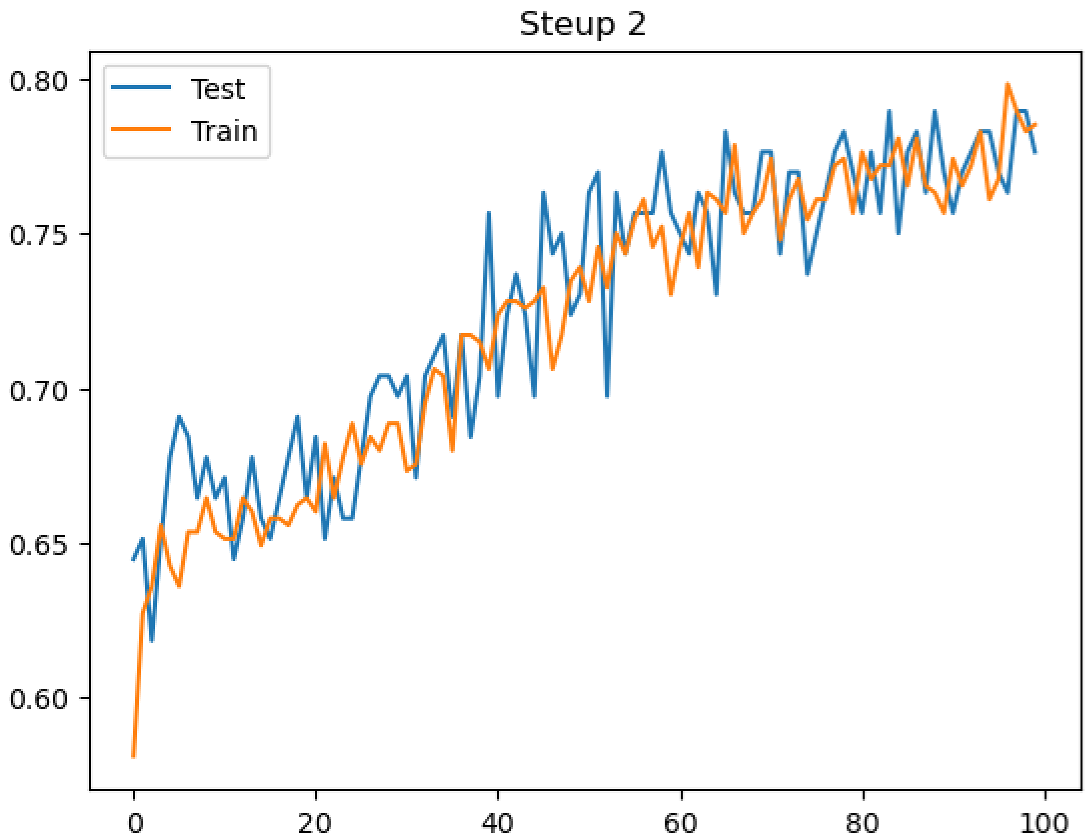
	Setup7	Test Accuracy	Setup8	Test Accuracy	Setup9	Test Accuracy
Dropout	0.1	79.61	0.1	66.45	0.1	78.29
Optimizer	SGD		SGD		SGD	
Activation Function	Leaky ReLU		Sigmoid		Tanh	
Initializer	Xavier Normal		Xavier Normal		Xavier Normal	

	Setup10	Test Accuracy	Setup11	Test Accuracy	Setup12	Test Accuracy
Dropout	0.1	72.37	0.1	66.45	0.1	69.08
Optimizer	SGD		SGD		SGD	
Activation Function	Leaky ReLU		Leaky ReLU		Leaky ReLU	
Initializer	Uniform		Normal		Ones	

Question 2

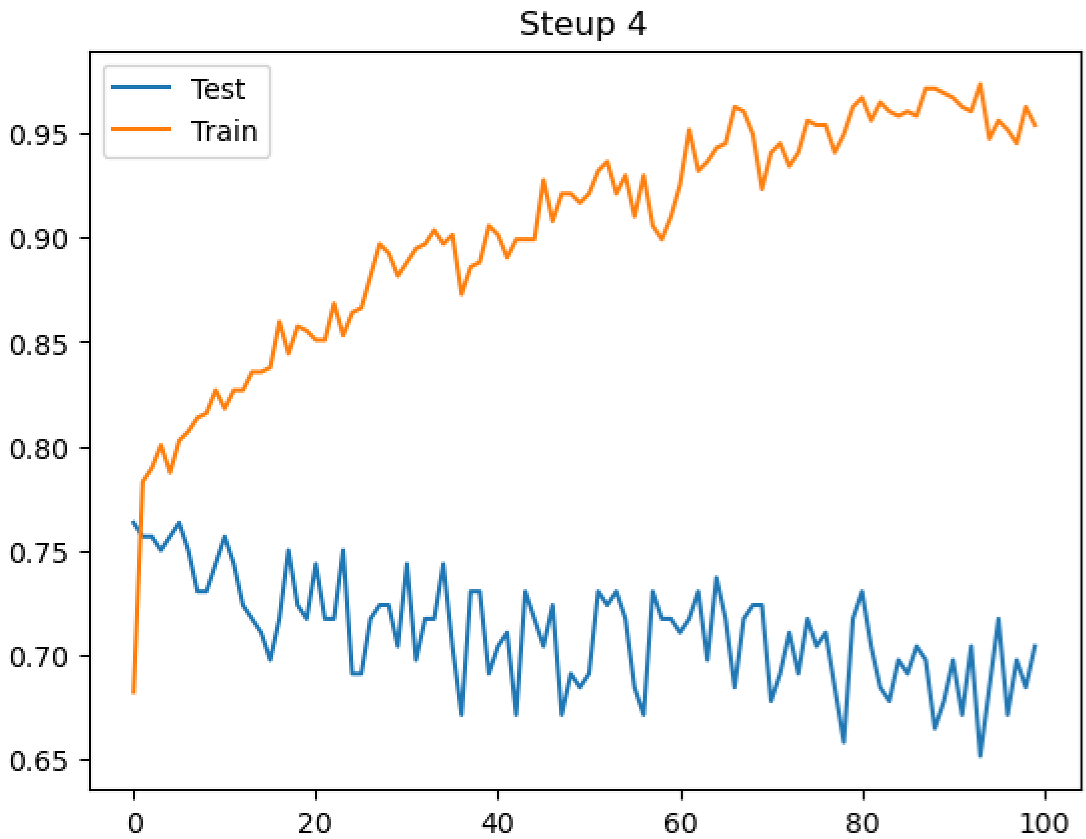


This setup has Test accuracy as 77.63% and train accuracy as 77.52%.

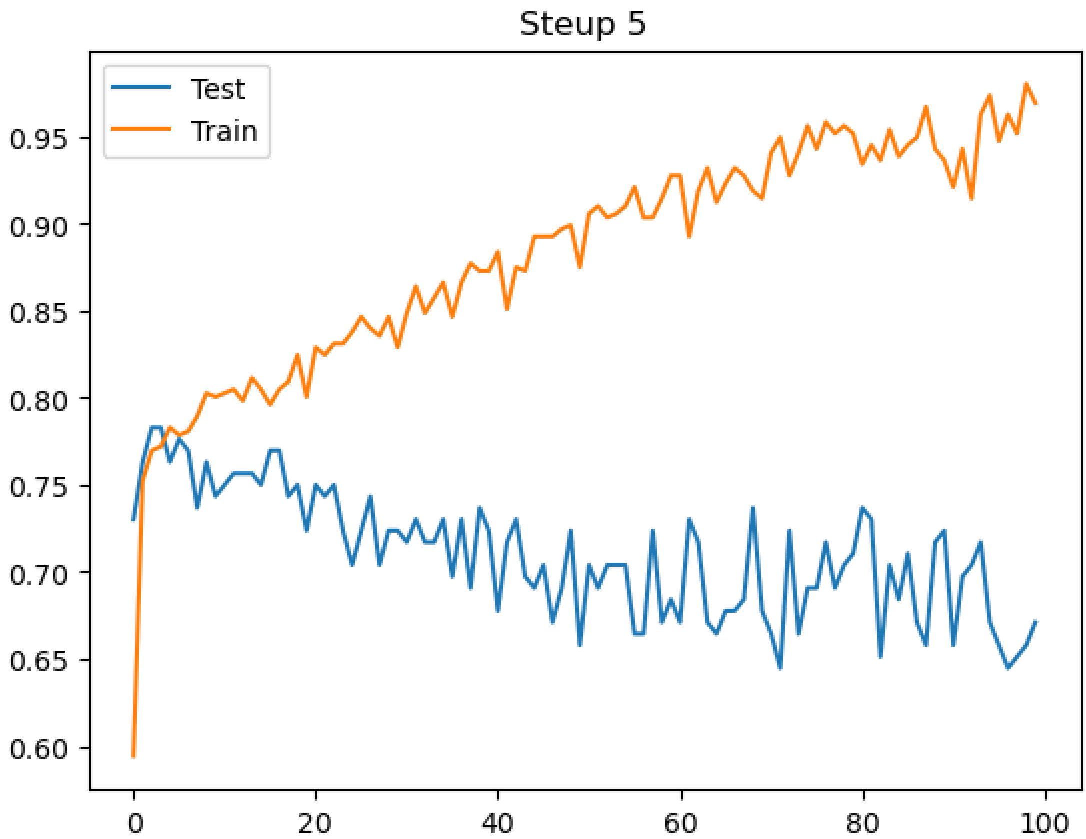


This setup has Test accuracy as 77.63% and train accuracy as 77.84%.

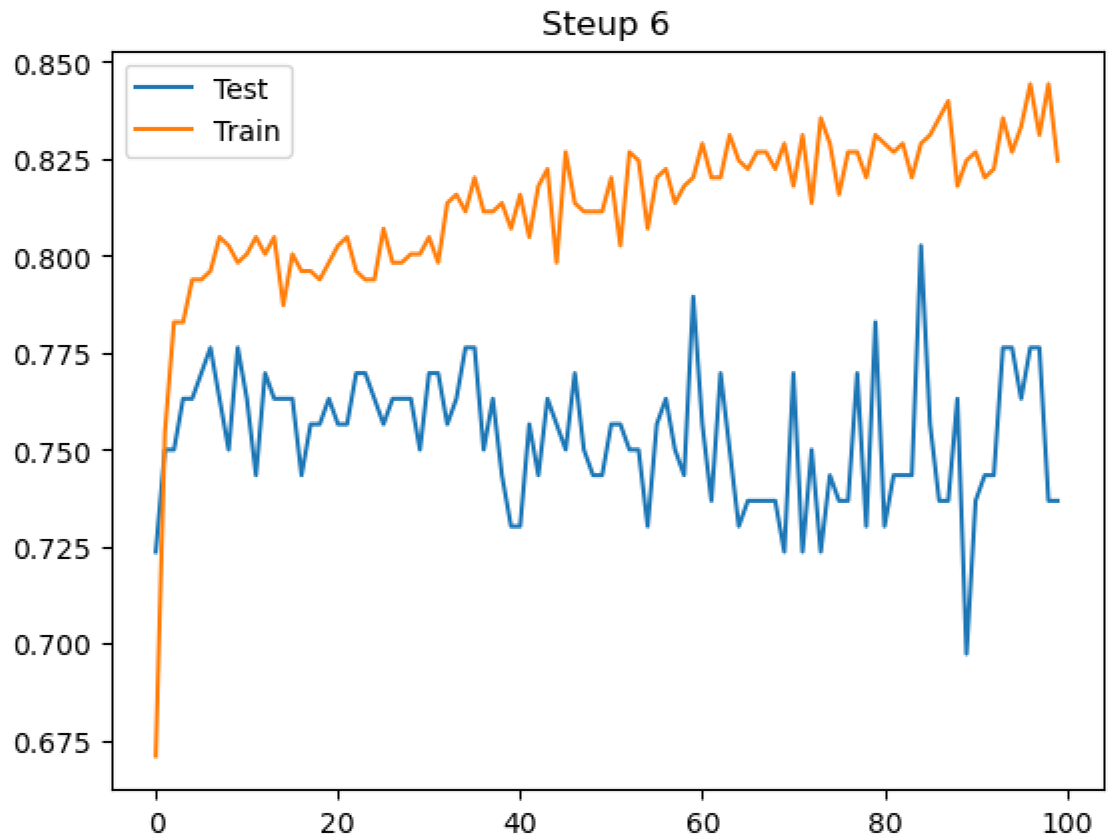
This setup has Test accuracy as 73.03% and train accuracy as 71.28%.



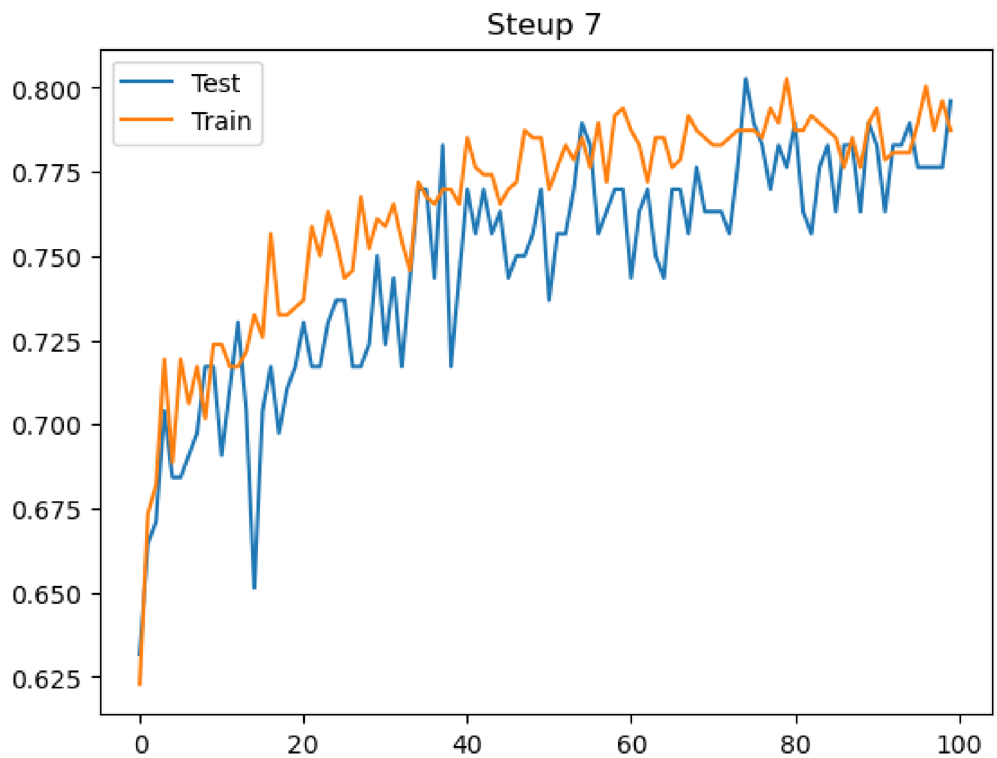
This setup has Test accuracy as 70.03% and train accuracy as 95.04%.



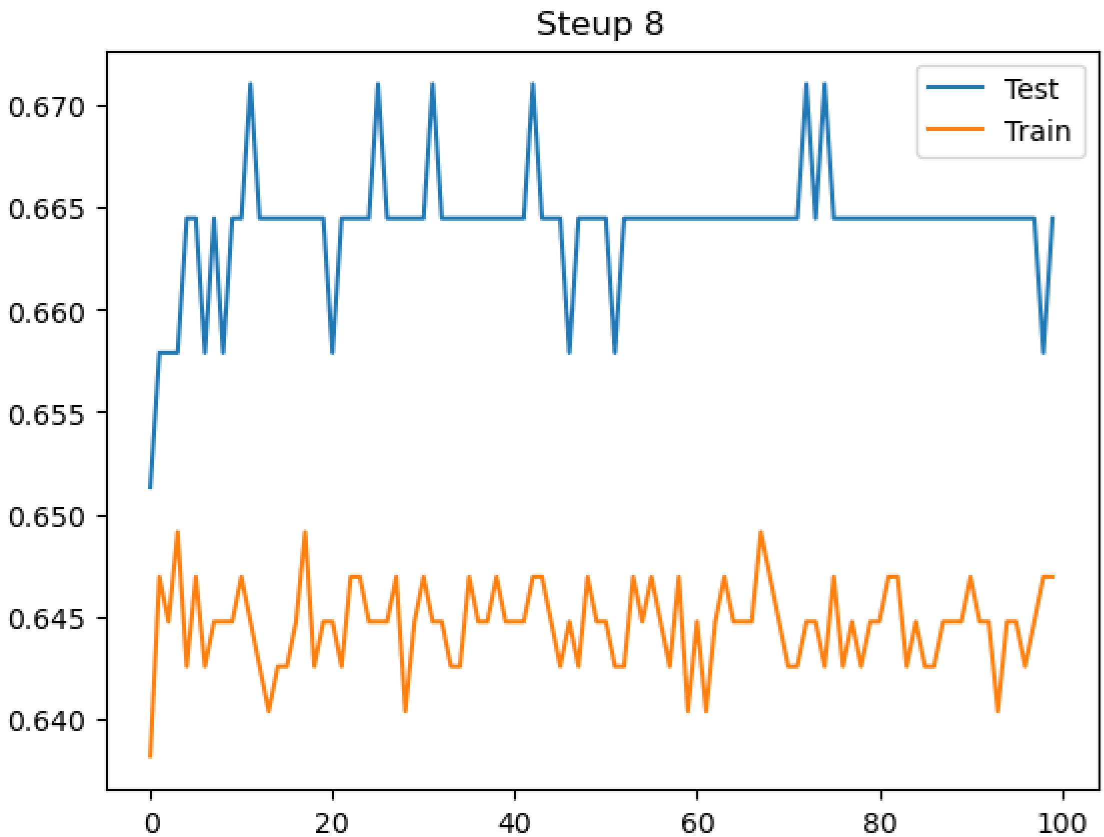
This setup has Test accuracy as 67.11% and train accuracy as 96.12%.



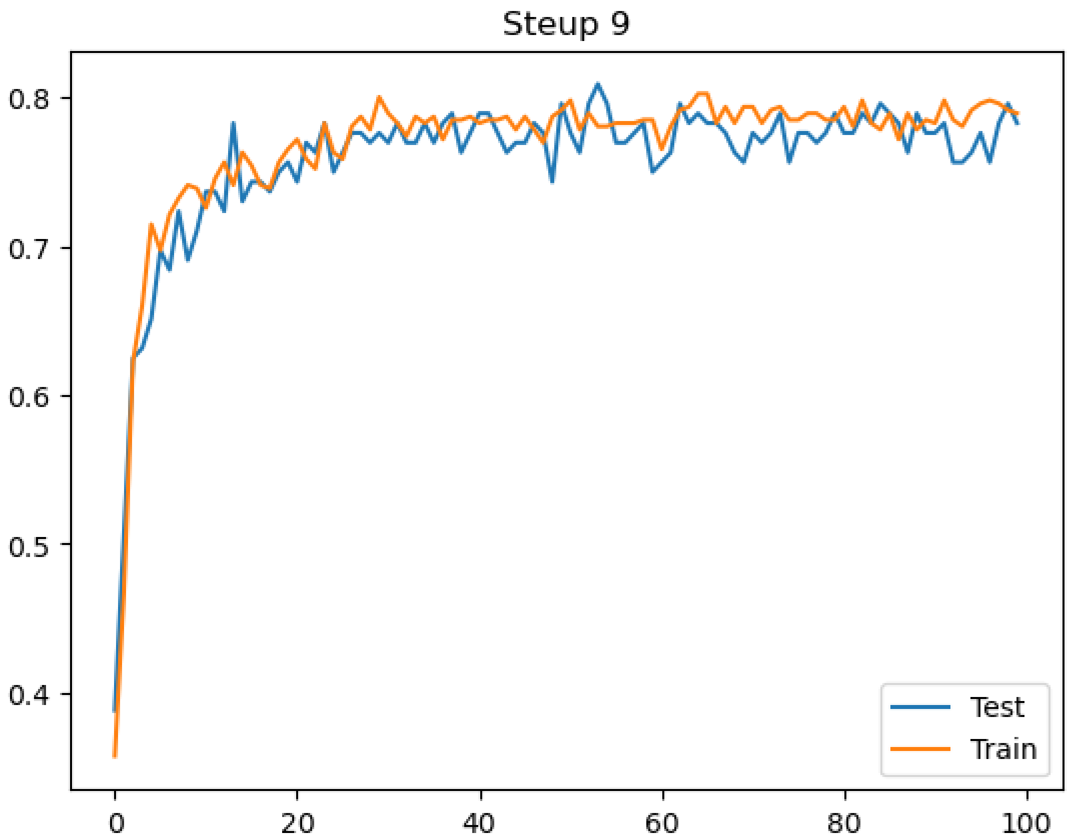
This setup has Test accuracy as 73.68% and train accuracy as 82.46%.



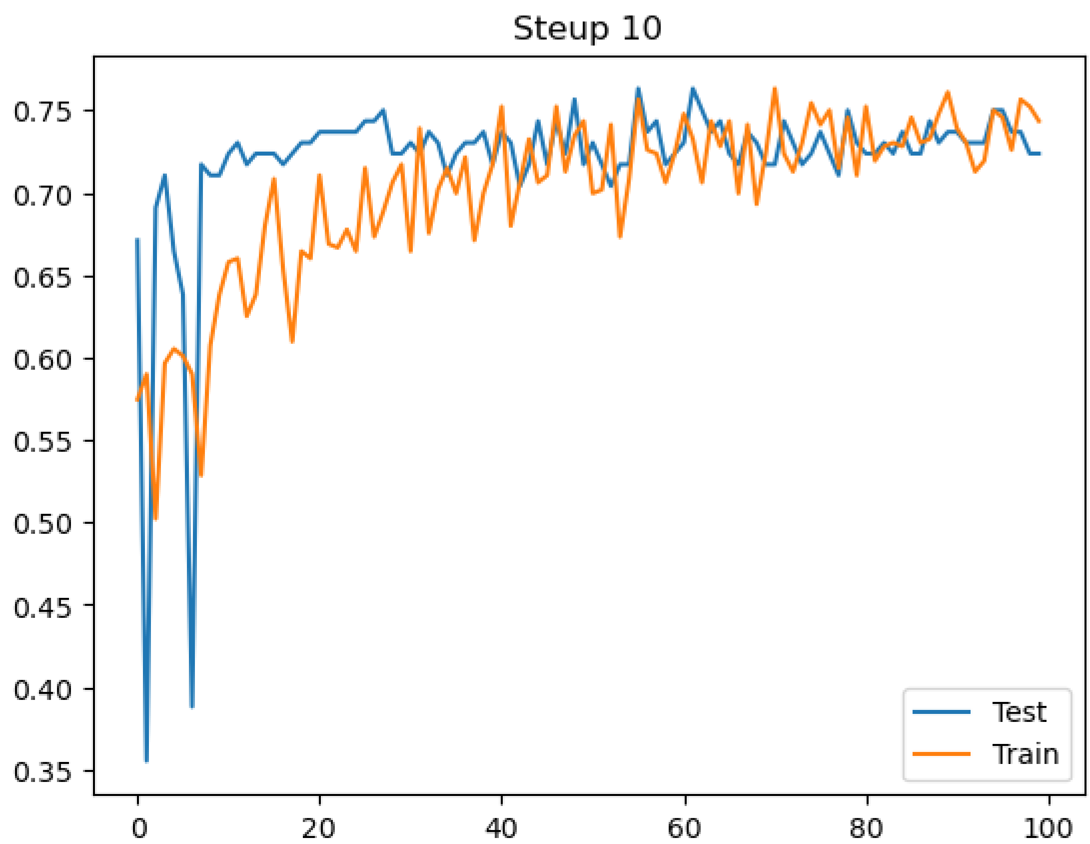
This setup has Test accuracy as 79.61% and train accuracy as 78.84%.



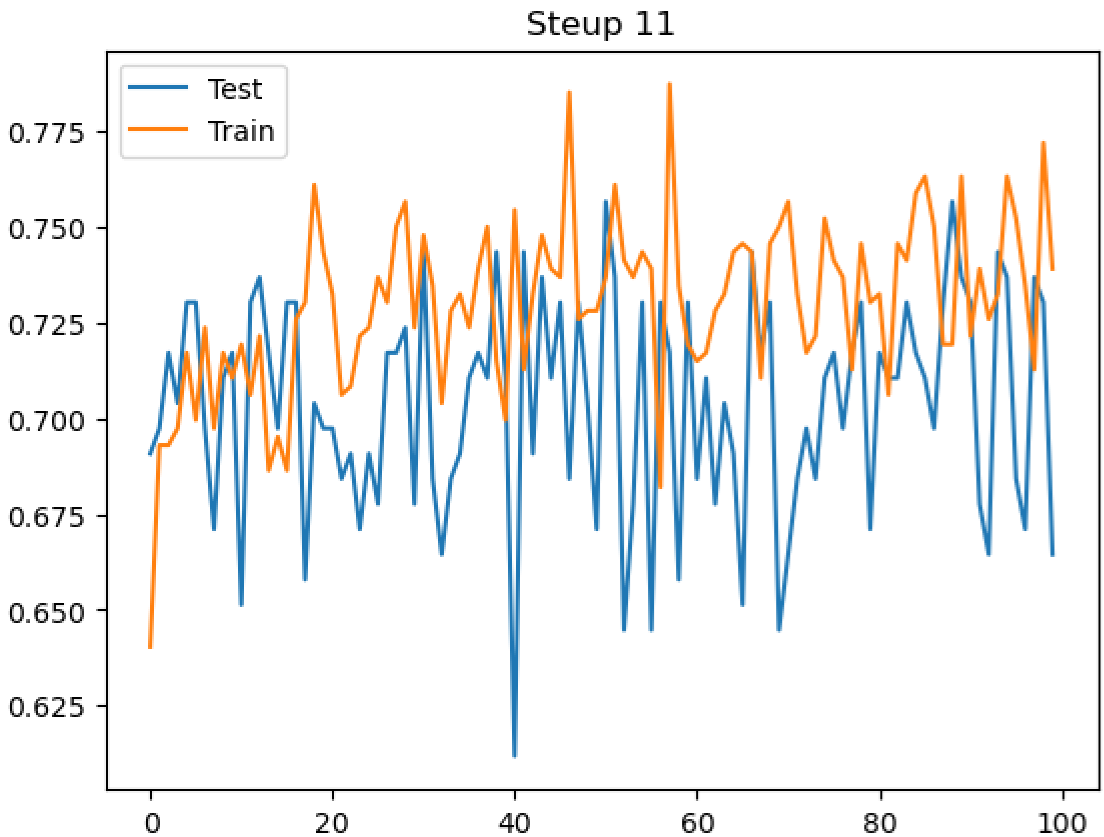
This setup has Test accuracy as 66.45% and train accuracy as 66.5%.



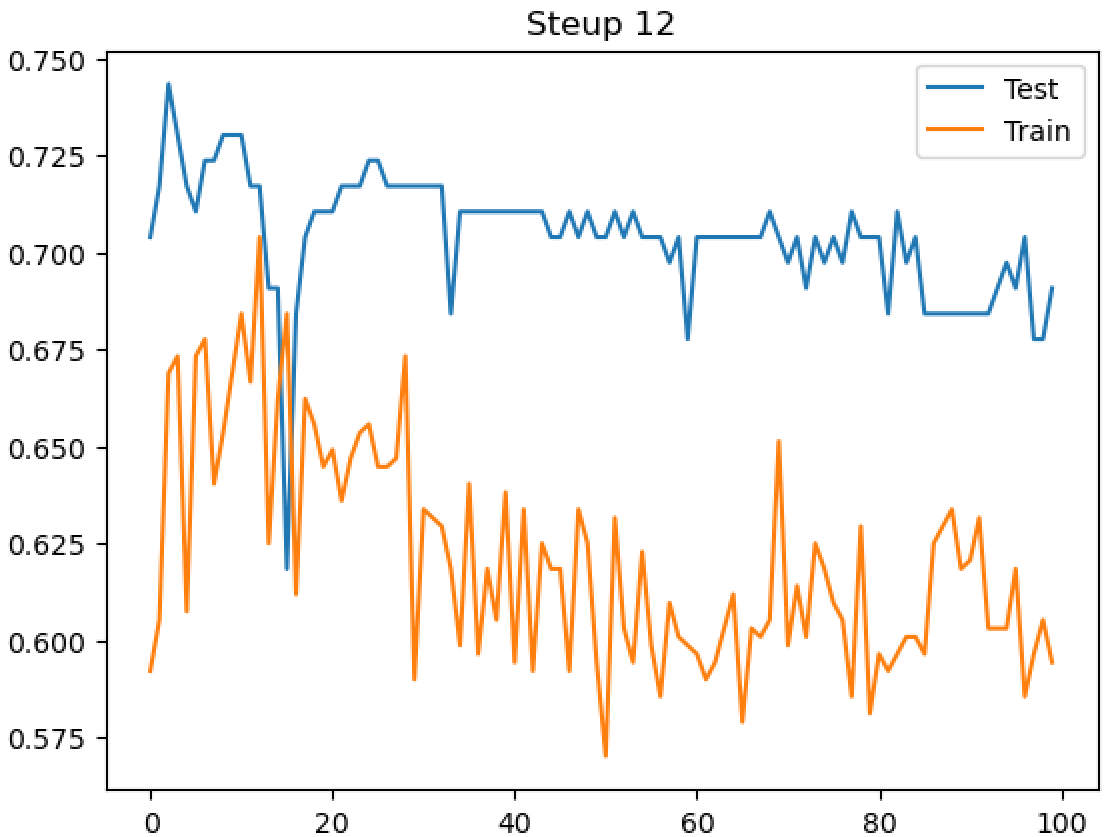
This setup has Test accuracy as 78.29% and train accuracy as 77.98%.



This setup has Test accuracy as 72.37% and train accuracy as 74.62%.



This setup has Test accuracy as 66.45% and train accuracy as 73.74%.



This setup has Test accuracy as 69.08% and train accuracy as 60.05%.

Question 3

We have totally tried 12 different setups, following the analysis of it:

Model 1: model setup of dropout value 0.1, SGD optimizer, Xavier Normal initializer and ReLU activation function we got test accuracy of 77.63%. As a result, training accuracy is high and generalization to testing data is poor and overall accuracy is high could improve with more tunning.

Model 2: model setup of dropout value 0.2, SGD optimizer, Xavier Normal initializer and ReLU activation function we got test accuracy of 77.63%. As a result, training accuracy is high and generalization to testing data is poor and overall accuracy is high could improve with more tunning.

Model 3: model setup of dropout value 0.3, SGD optimizer, Xavier Normal initializer and ReLU activation function we got test accuracy of 73.03%. As a result, training accuracy is high and generalization to testing data is poor and overall accuracy is high could improve with more tunning.

Model 4: model setup of dropout value 0.1, Adam optimizer, Xavier Normal initializer and ReLU activation function we got test accuracy of 70.39%. As a result, training accuracy is high and generalization to testing data is poor and overall accuracy is low.

Model 5: model setup of dropout value 0.1, RMS prop optimizer, Xavier Normal initializer and ReLU activation function we got test accuracy of 67.11%. As a result, training accuracy is high and generalization to testing data is poor and overall accuracy is low.

Model 6: model setup of dropout value 0.1, Adagrad optimizer, Xavier Normal initializer and ReLU activation function we got test accuracy of 73.68%. As a result, training accuracy is high and generalization to testing data is poor and overall accuracy is high could improve with more tunning.

Model 7: model setup of dropout value 0.1, SGD optimizer, Xavier Normal initializer and Leaky ReLU activation function we got test accuracy of 79.61%. As a result, training accuracy is lower than testing and generalization to testing data is high and overall accuracy is best and selected.

Model 8: model setup of dropout value 0.1, SGD optimizer, Xavier Normal initializer and Sigmoid activation function we got test accuracy of 66.45%. As a result, training accuracy is low and generalization to testing data is high but overall accuracy is low.

Model 9: model setup of dropout value 0.1, SGD optimizer, Xavier Normal initializer and Tanh activation function we got test accuracy of 78.29%. As a result, training accuracy is high and generalization to testing data is poor and overall accuracy is high could improve with more tunning.

Model 10: model setup of dropout value 0.1, SGD optimizer, Uniform initializer and Leaky ReLu activation function we got test accuracy of 72.37%. As a result, training accuracy is high and generalization to testing data is poor and overall accuracy is low.

Model 11: model setup of dropout value 0.1, SGD optimizer, Normal initializer and Leaky ReLu activation function we got test accuracy of 66.45%. As a result, training accuracy is high and generalization to testing data is poor and overall accuracy is low.

Model 12: model setup of dropout value 0.1, SGD optimizer, Ones initializer and Leaky ReLu activation function we got test accuracy of 69.08%. As a result, training accuracy is low and generalization to testing data is good but overall accuracy is low.

Question 4

We have used 4 techniques to improve the accuracy of the model, which are as follows:

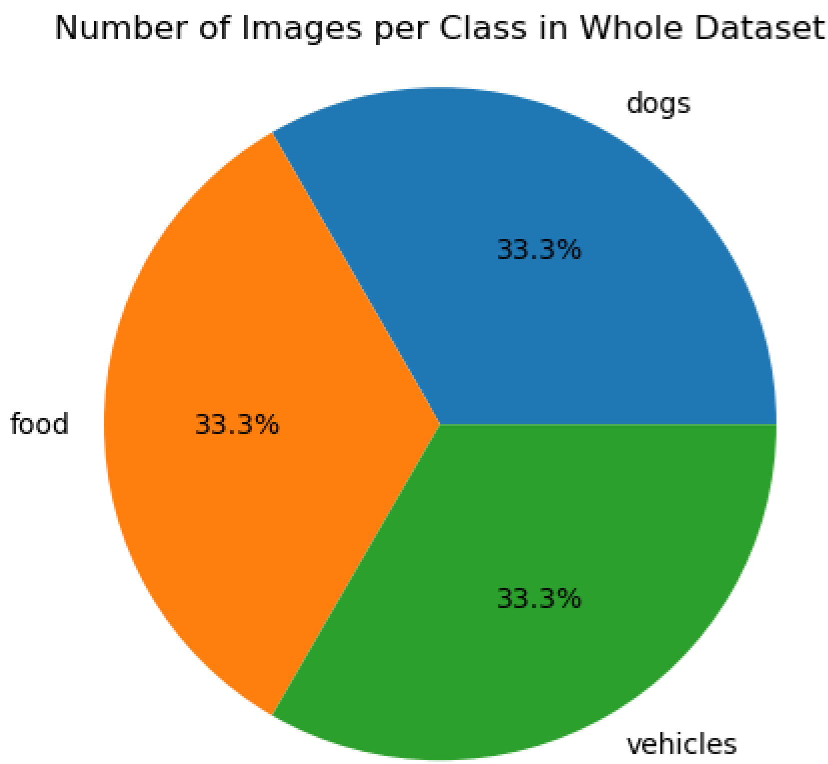
1. Learning Rate Scheduler: A learning rate scheduler is a predetermined structure that modifies the learning rate between epochs or iterations. Setting the learning rate to a higher number at the start of the training to enable faster convergence aids in the algorithm's swift convergence to an optimal state.
2. Gradient Clipping: Gradient clipping, is a method for preventing exploding gradients in very networks. Before sending the error derivatives back across the network, it requires capping them. The weights are then updated using the capped gradients, leading to reduced weights. As a result, problems with disappearing and exploding gradients during the training of recurrent neural networks can be avoided1.
3. K-Fold: A resampling technique called K-fold cross-validation is used to assess machine learning models on a small data sample. The process contains a single parameter, k, that designates how many groups should be created from a given data sample. When a particular value for k is selected, it may be substituted for k in the model's reference, such as when k=10 becomes 10-fold cross-validation.
4. Early Stopping: Early stopping is a type of regularization that helps prevent overfitting when a learner is being trained iteratively, such with gradient descent. With each repetition, these techniques update the learner to improve its fit with the training data. This enhances the learner's performance on data not in the training set, but only to a certain extent. However, after that time, generalization error increases while the learner's fit to the training data improves. It is possible to perform a certain number of iterations before the learner starts to over-fit, according to early stopping guidelines.

Report Part 3

Question 1

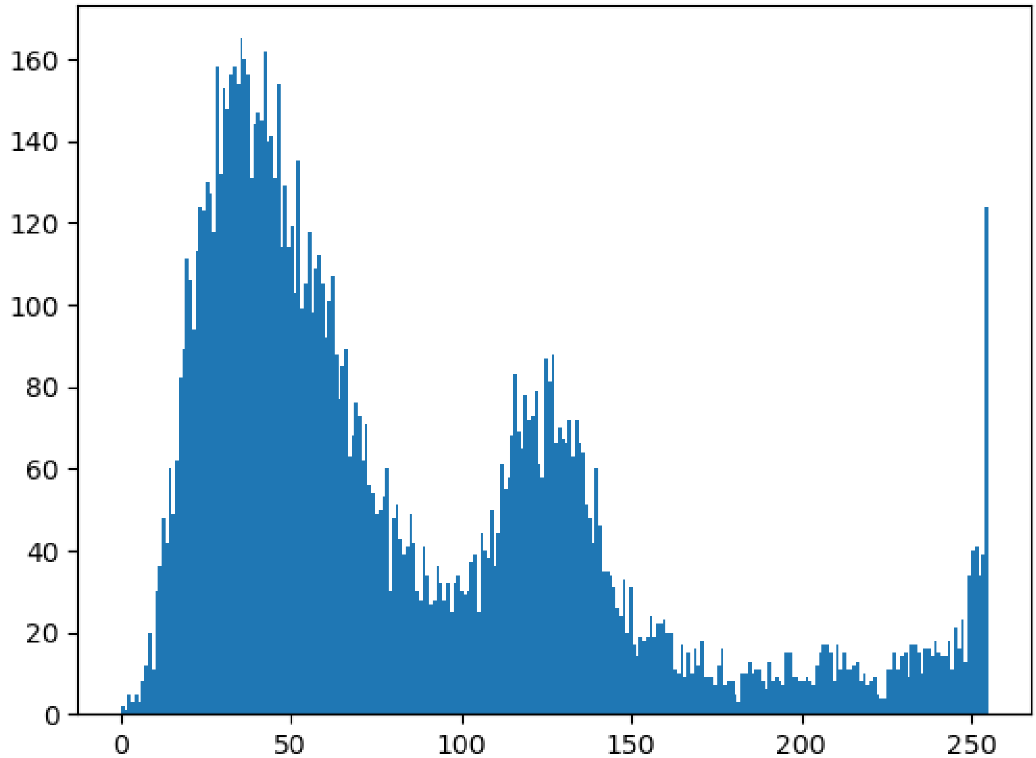
The given dataset is a collection of images, where each image belongs to one of three classes - dogs, vehicles, and food. The type of data we are encountering is image data. Each image is a collection of pixels, and the features of the image are derived from these pixels. The dataset comprises 3,000 entries, with 1,000 images for each class. In terms of variables, each image can have multiple variables depending on the features you are extracting.

Three visualizations:-



Visualization 1

The above pie chart shows that the dataset provided has equal number of photos distributed among the 3 classes.



Visualization 2

We have plotted a histogram for one of the images. This shows the distribution of pixel intensities in the image. The x-axis represents the pixel intensity values ranging from 0 to 255, and the y-axis represents the frequency of occurrence of each intensity value.

From the above plot, we can conclude that the image has a wide range of pixel values, with most of the pixel values concentrated in the lower range (0-100) and few concentrated in the higher range (above 250). There are two prominent peaks, one near 45 and the other near 125, which may correspond to different types of features in the image. The first peak near 45 indicates the presence of dark regions in the image, while the second peak near 125 may indicate the presence of mid-range brightness regions. Overall, the histogram can provide insights into the distribution of pixel values in the image.

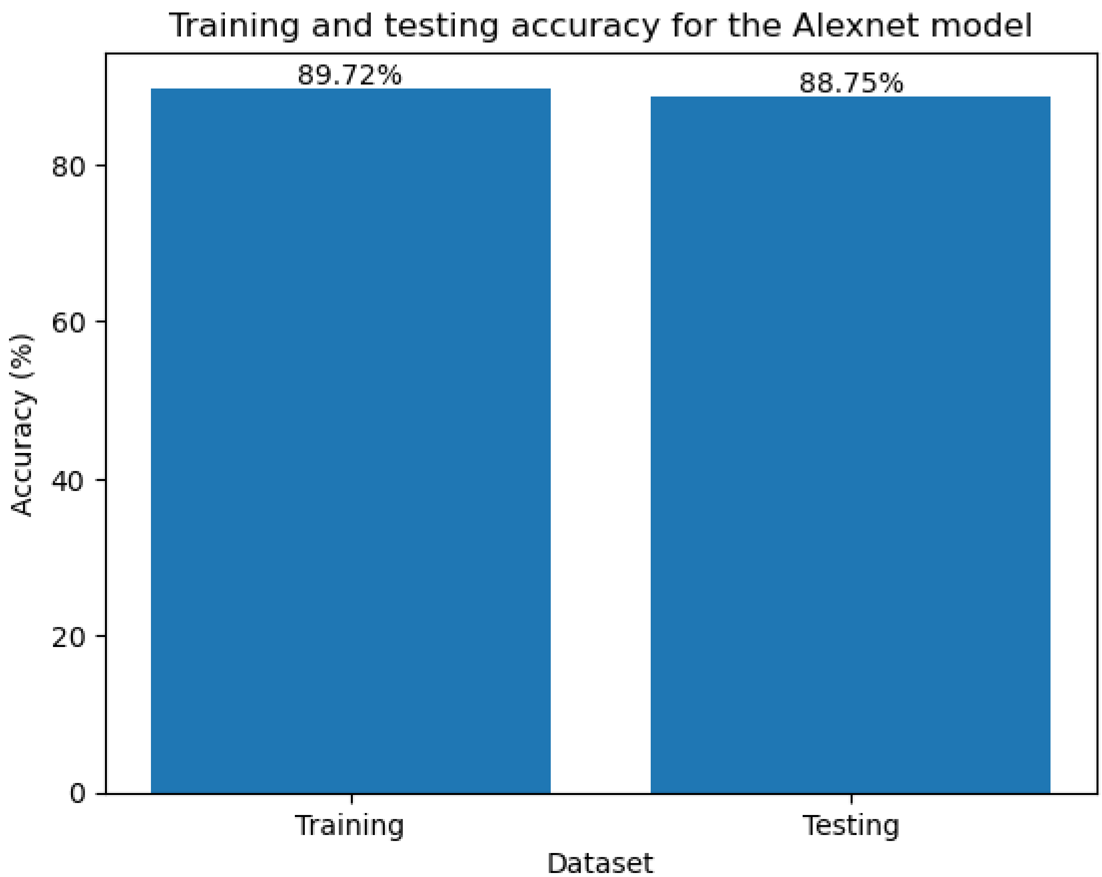
Visualization 3

Here we were potting images from each class. By plotting them, we confirmed that the images were loaded properly in the 3-channel format. This helped us ensure the data was ready for further processing and analysis."

Question 2:

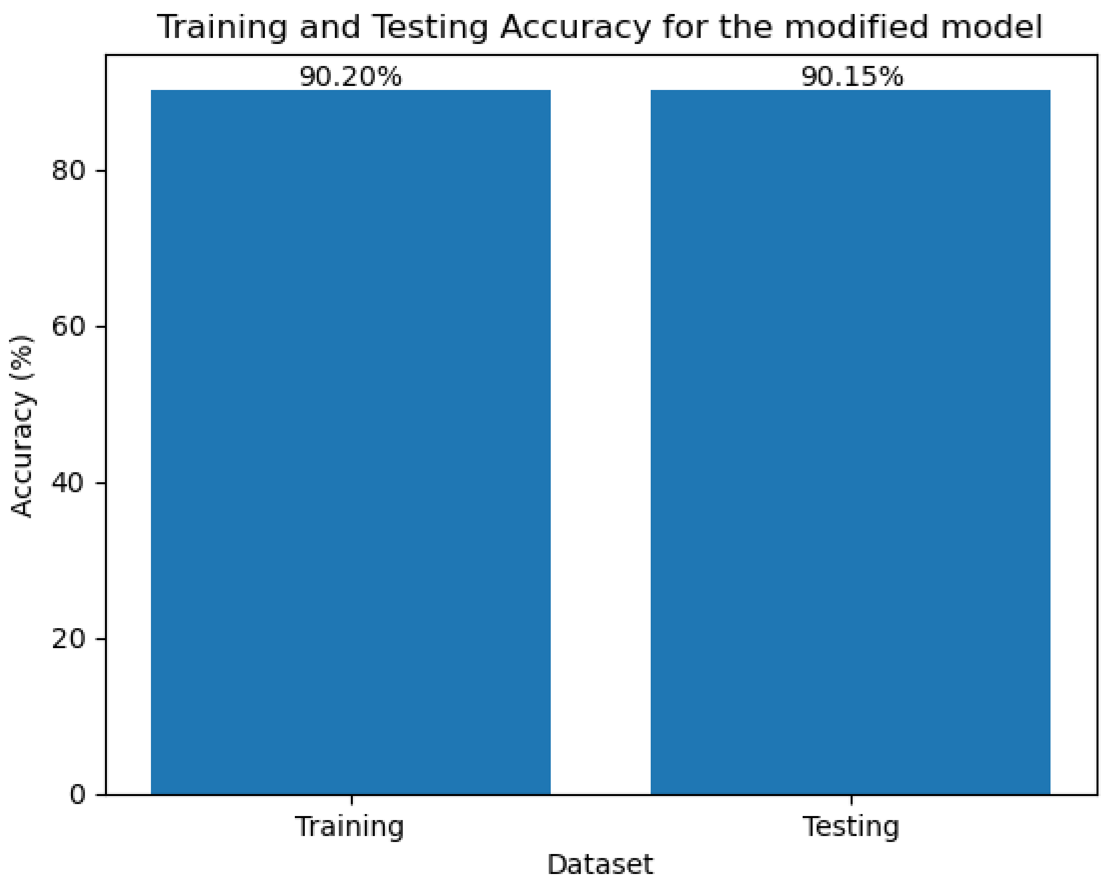
The AlexNet model which we coded is used to classify images into 3 different categories. The model uses convolutional layers with filters of different sizes to create feature maps that help to identify patterns in the images. The output of the convolutional layers is passed through a max pooling layer and then an adaptive average pooling layer to reduce the output size. The output is then passed through fully connected layers with ReLU activation functions and dropout to prevent overfitting. The final output is a probability distribution over the 3 classes- dogs, food and vehicles.

Our model achieved an accuracy of 89.72% on the training set and 88.75% on the testing set.



Question 3:

The AlexNet model was modified to improve its performance in training accuracy and training time. Two convolutional layers and one fully connected layer were removed from the model, and the number of epochs was increased from 10 to 15. The removal of layers helped to reduce the model's complexity, reduce overfitting, and improve the training time. Increasing the number of epochs allowed the model to learn more from the training data and improve the training accuracy. The modifications resulted in an improvement in both the training and testing accuracy of the model as we achieved an accuracy of 90.2% on the training set and 90.15% on the testing set.

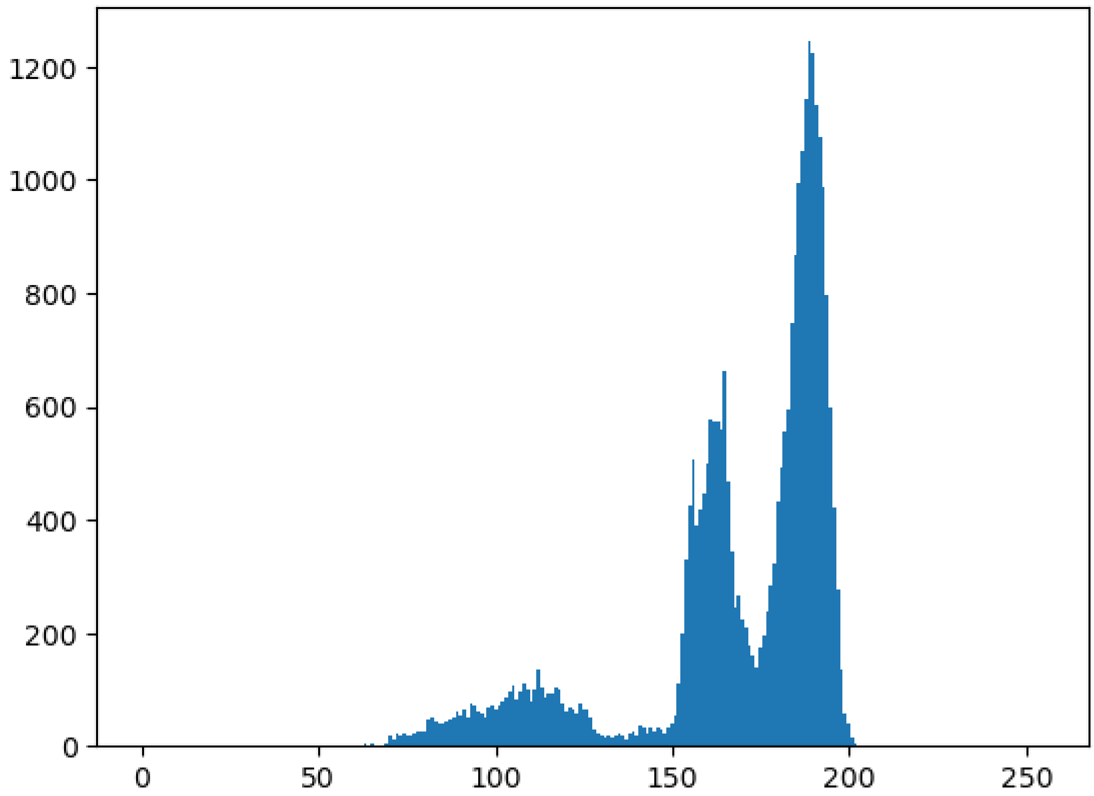


Report Part 4

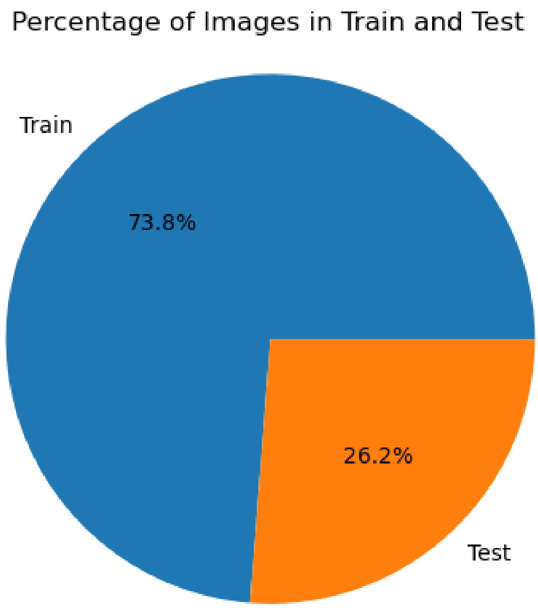
Question 1

The given dataset is collection of images which contains house number take from street view which is a great dataset used to bench mark a model for identifying street numbers. There are a total of 99,289 images in this dataset. Out of there are 26,032 number of images used for testing and 73,257 number of images used for training the model.

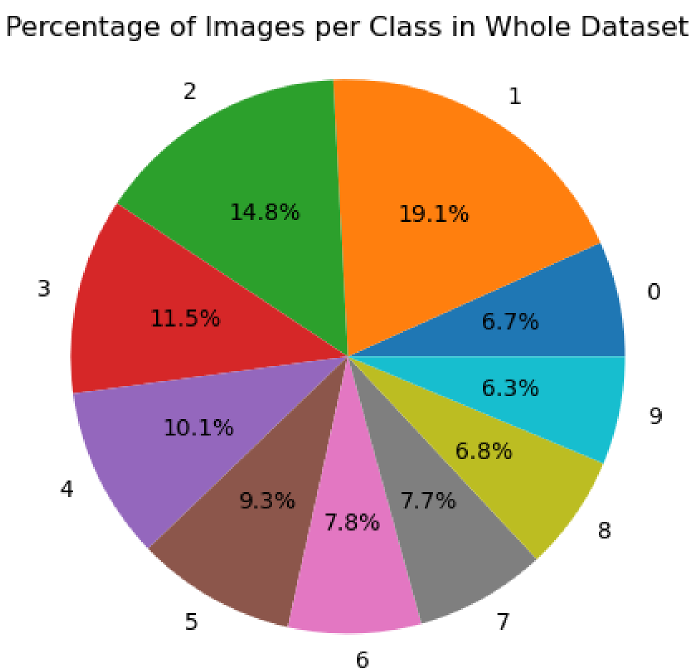
Question 2



The above plot is a histogram of an image from test dataset of SVHN dataset. Here we can see most of the values are concentrated in the mid-range. So, we would need to train the model to differentiate the background from number we need .



The above pie chart shows how the data is split between test and train dataset and we can see a standard train to test split ration of about 0.7 to train and 0.3 to test.

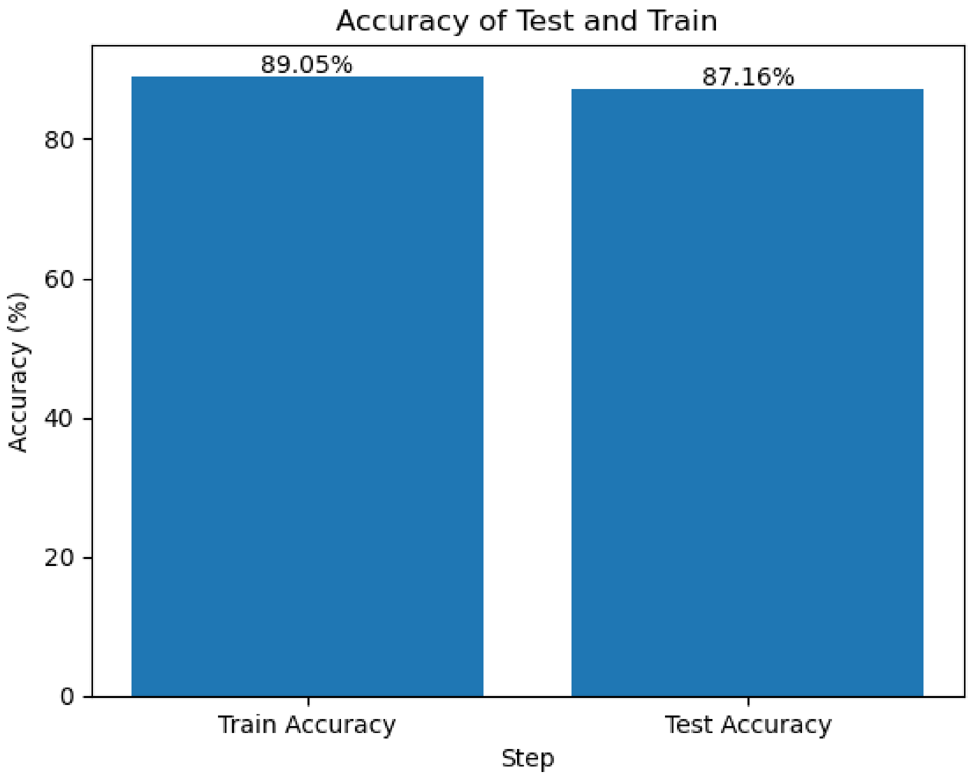


The above pie chart shows us the split in data between different classes and we can see almost 35% of the data is for classes 1 and 2. This could skew the model to predict classes 1 and 2 then other classes.

Question 3

We have used RandomRotation and RandomPosterize functions of torchvision.transforms package. This helps us to train our model to predict accurately even if the image is tilted or not leveled and also in situations where the color of the numbers have been worn over time.

Question 4



We have used AlexNet as our base model and modified it to improve it accuracy for this dataset. Here we can see that training accuracy is 89.05% and testing accuracy is 87.16%. If the model and training is tunned future then the accuracy could improve but would need a lot of time to train the model multiple times.