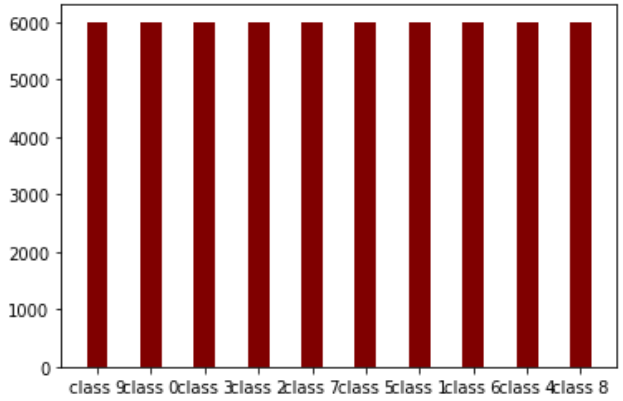
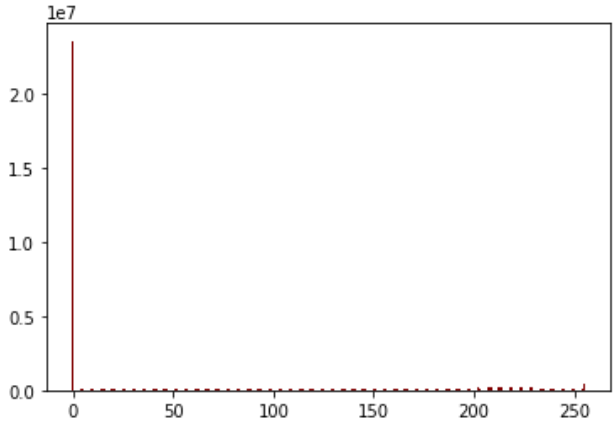
Ans 2 a)
Downloaded and loaded the data using idx2numpy library. Then applied EDA on the data. First checked by creating a list of sizes of images and it turns out that the size of list is 1 which means that all the images are of the same size and observed that they're all of (28x28). Then I checked the class imbalance in which I've looped through the train labels and created a dictionary containing key as the labels and corresponding value as the count. Observed that classes are balanced.



Applied normalization by looping through each cell of the image and dividing by 255.
Plotted pixel intensity graph by looping through all the images of the train dataset and storing the count of pixels in a list.



Ans 2 b)
In OVO, I have 10 classes so there will be 45 models to be trained. In each case, I am choosing only two labels and then classifying test images into those 2 labels only. I am doing this for 45 cases as well. Finally, the test image belongs to the most frequent label it is assigned to from the 10 labels.

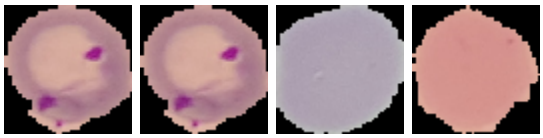
Ans 2 c)
In OVR, as I have 10 classes so there will be 10 models to be trained. In each case, one class would be treated as 1 and all other 9 classes would be treated as 0. After training, I am predicting the output for each model. Then, whichever model has the higher probability for a particular test image I am assigning that as the final predicted model.

Ans 2 d)
OVO scratch accuracy : 83.71%
OVO inbuilt accuracy : 84.09%

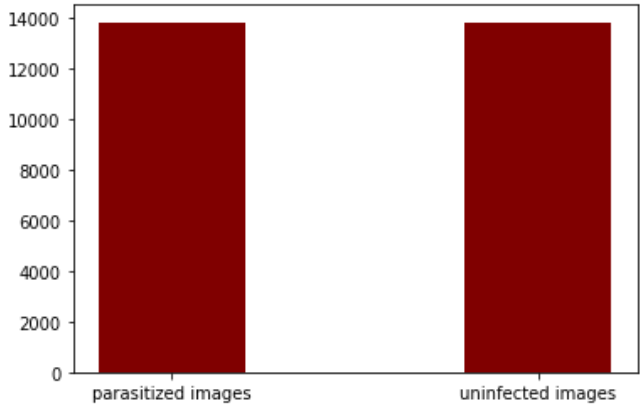
OVR scratch accuracy : 78.33%
OVR inbuilt accuracy : 83.83%

The results for the OVO case have almost the same accuracy for inbuilt as well as scratch implementation. In case of OVR, the scratch implementation and inbuilt model implementation accuracy differs by approximately 5%.

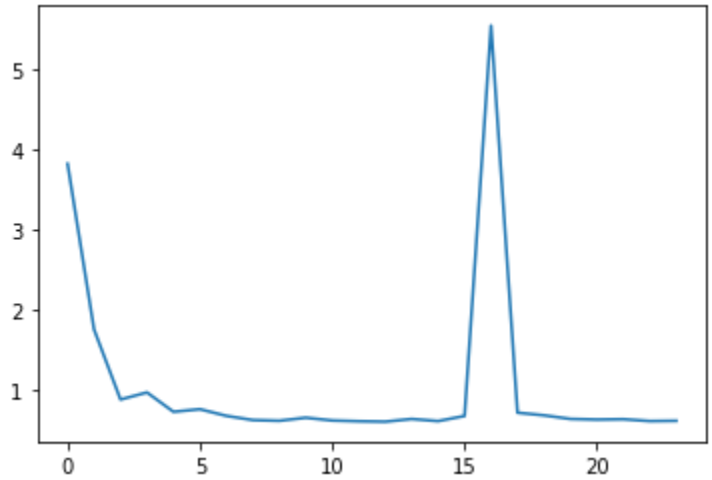
Ans 3 a) 2 images of each class



Class imbalance

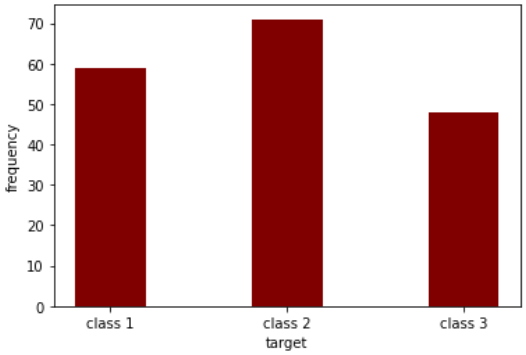


For image size uniformity, I have changed image sizes to be 64x64.
Ans 3 D)
Training loss vs epoch

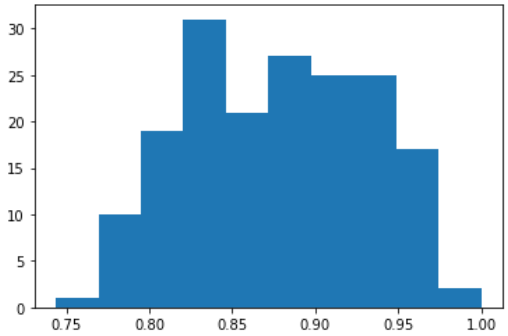


The training starts with a loss of around 3.8 and decreases to 0.67 and then there is finally a spike to 5.5 and then again the loss settles down to 0.61. The epochs finally stop when loss is 0.61 which is very less and this shows that the trained model is good because the loss is very less when the program stops.

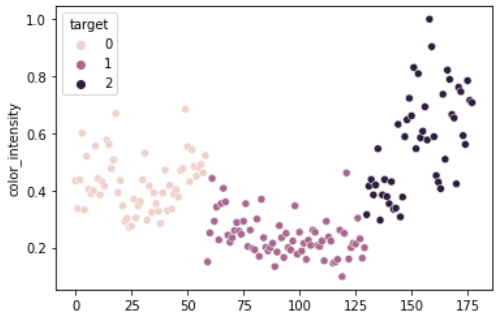
Ans 4 a)
In EDA, firstly I have normalized the data by calculating the max of each column and then dividing each column cell by their maximum. Then I have checked for class imbalance, for that I have created a dictionary and I am storing count of each class as a value in that dictionary and then plotting the bar graph.



Then, I am plotting the histogram of the column ‘alcohol’.



After that I am plotting a scatter plot where the y axis is the color_intensity column and the plot is class wise divided.



Ans 4)

CV technique	Accuracy
Hold Out Cross-Validation	94.44
4-fold cross-validation	81.5
Stratified 3-fold Cross-Validation	94.416
Monte Carlo Cross-Validation	95.06
Leave P Out Cross-Validation	96.06

Hold out cross validation is fast but at the same time its limitation is that it requires a certain percentage of data to be training.
In 4 fold cross validation the accuracy is low because the classes are imbalanced and therefore k-fold cross validation does not give a good accuracy.
In stratified 3-fold cross validation, the accuracy is better than k-fold because it overcomes the problem of class imbalances and gives good accuracy.
In leave p out cross validation, the accuracy is higher but as we increase the value of “p” the run time of code increases exponentially.

For our dataset, I think **stratified 3-fold cross validation** works the best because it gives good accuracy and is not so time consuming and also it works for imbalanced class data.