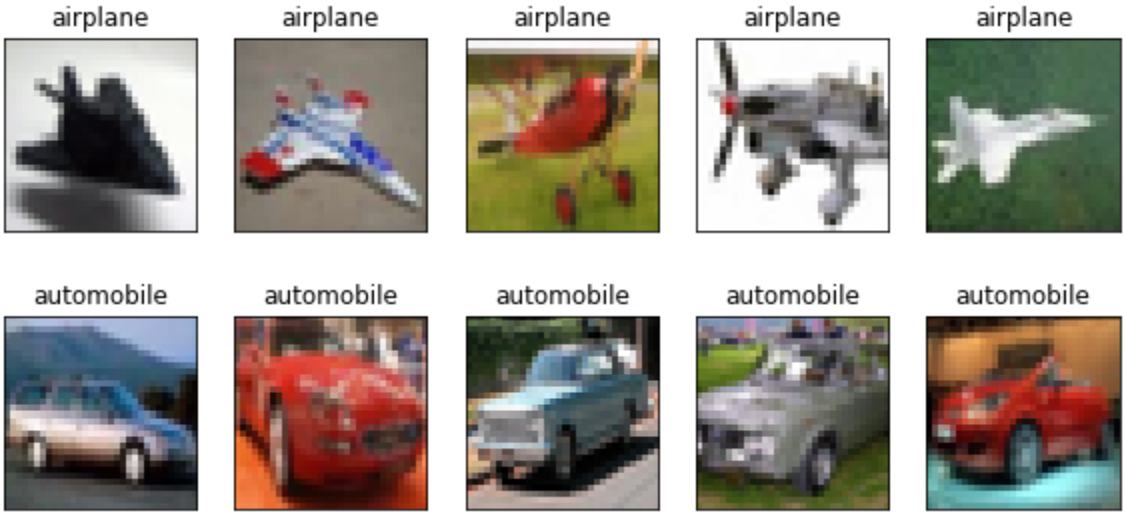
Report

Question 1:

Part 1

Steps

1. Unpickling the data
2. Reshape and transpose an image/array into (32,32,3) i.e, 32x32 =1024 entries and 3 for RGB channels.
3. Visualizing images



Similarly for other classes.

Part 2

Steps

1. Merging batches to create training and testing dataset
2. Applying LDA on the merged batches

Part 3

Steps:

1. Calculated accuracy and class-wise accuracy for the testing dataset

```
Accuracy  0.3713

Class-wise accuracy
Label 0  -  0.463
Label 1  -  0.415
```

```
Label 2  -  0.255
Label 3  -  0.245
Label 4  -  0.271
Label 5  -  0.329
Label 6  -  0.413
Label 7  -  0.404
Label 8  -  0.494
Label 9  -  0.424
```

Question 2:

Steps

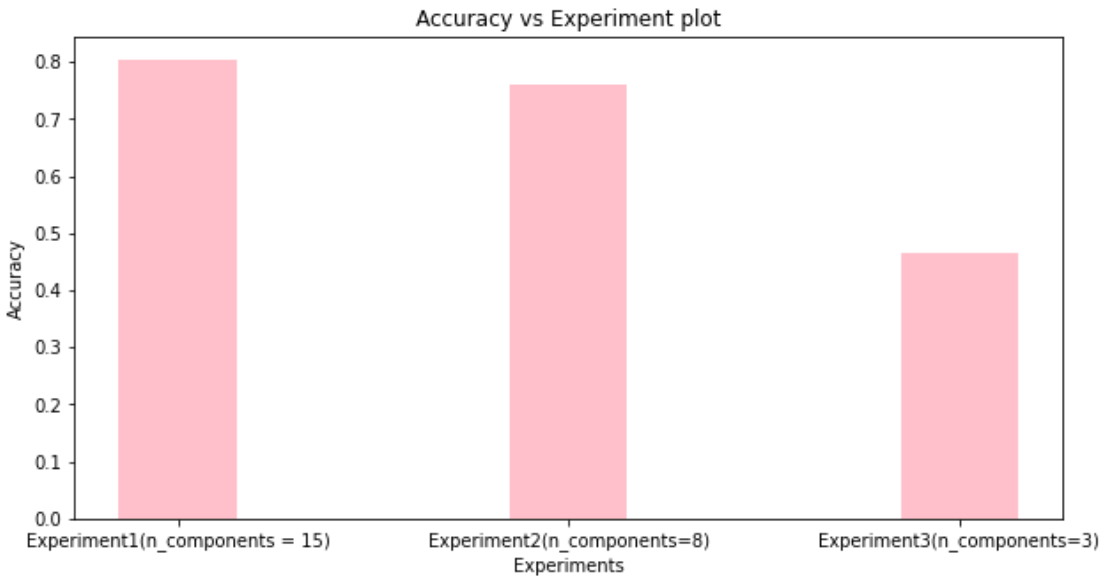
1. Loaded the dataset using idx2numpy
2. Reshaping the data from 3d to 2d
3. Applying PCA on the given data for n_components = 15, 8 and 3
4. Applying LDA on the transformed data
5. Calculated accuracy
6. Plotted accuracy of all three experiments on the testing dataset

```
Classification metrics for PCA(n_componetns = 15)
              precision    recall  f1-score   support

           0       0.90      0.92      0.91       962
           1       0.91      0.79      0.85      1318
           2       0.74      0.85      0.79       899
           3       0.79      0.77      0.78      1044
           4       0.81      0.81      0.81       978
           5       0.65      0.72      0.68       808
           6       0.83      0.85      0.84       927
           7       0.84      0.88      0.86       976
           8       0.75      0.74      0.74       989
           9       0.79      0.72      0.75      1099

    accuracy                           0.80     10000
   macro avg       0.80      0.80      0.80     10000
weighted avg       0.81      0.80      0.80     10000


Classfication metrics for PCA(n_components = 8)
              precision    recall  f1-score   support

           0       0.82      0.89      0.85       906
           1       0.96      0.75      0.84      1453
           2       0.69      0.78      0.73       916
```

3	0.79	0.76	0.78	1057
4	0.74	0.67	0.70	1081
5	0.61	0.66	0.64	825
6	0.83	0.83	0.83	950
7	0.82	0.89	0.85	949
8	0.70	0.76	0.73	888
9	0.61	0.63	0.62	975
accuracy			0.76	10000
macro avg	0.76	0.76	0.76	10000
weighted avg	0.77	0.76	0.76	10000

Classification metrics for PCA(n_components = 3)

	precision	recall	f1-score	support
0	0.76	0.61	0.67	1220
1	0.98	0.72	0.83	1538
2	0.24	0.35	0.28	694
3	0.69	0.48	0.57	1457
4	0.41	0.40	0.40	1016
5	0.12	0.24	0.16	422
6	0.41	0.35	0.38	1144
7	0.57	0.40	0.47	1440
8	0.12	0.29	0.17	402
9	0.26	0.39	0.31	667
accuracy			0.47	10000
macro avg	0.46	0.42	0.42	10000
weighted avg	0.56	0.47	0.50	10000



Reasoning: Experiment1(n_components =15) has given the best result.
We can see that n_components = 15 gives the best accuracy as compared to 8 and 3. This is because there is more a number of features for the data to be trained on and hence it gives better accuracy.

Question 3:

Steps:

1. loaded the dataset
2. sorted the dataset by label
3. splitting into features and label
4. implemented FDA function and return the coefficient vector W
5. projecting training data using W for both training and testing data
6. applying LDA on the projected data Y for classifying the testing samples
7. Calculated accuracy and class-wise accuracy for the testing dataset

```
Accuracy  0.5817

Class-wise accuracy
Label 0  -  0.661
Label 1  -  0.782
Label 2  -  0.402
Label 3  -  0.48
Label 4  -  0.562
Label 5  -  0.579
Label 6  -  0.195
Label 7  -  0.745
```

```
Label 8  -  0.596
Label 9  -  0.815
```

Question 4:

Steps:

1. loaded the dataset
2. Reshaping the data from 3d to 2d
3. Applying PCA for the best value reported in question-2 i.e n_components = 15
4. implemented FDA function and return the coefficient vector W
5. projecting training data using W for both training and testing data
6. applying LDA on the projected data Y for classifying the testing samples
7. Calculated accuracy and class-wise accuracy for the testing dataset

```
Accuracy  0.6877

Class-wise accuracy
Label 0  -  0.7887755102040817
Label 1  -  0.8898678414096917
Label 2  -  0.6841085271317829
Label 3  -  0.7029702970297029
Label 4  -  0.5936863543788188
Label 5  -  0.4349775784753363
Label 6  -  0.7651356993736952
Label 7  -  0.7772373540856031
Label 8  -  0.6878850102669405
Label 9  -  0.5004955401387512
```