**Pattern Recognition**

**Lab 1 Report**

| **Abdelrhman Ahmed Yousry Elnainay** | **19015889** |
| --- | --- |
| **Mahmoud Ibrahim Gad Abou Alwafa** | **19016532** |
| **Gamal Abdul Hameed Nasef Newysar** | **19015550** |

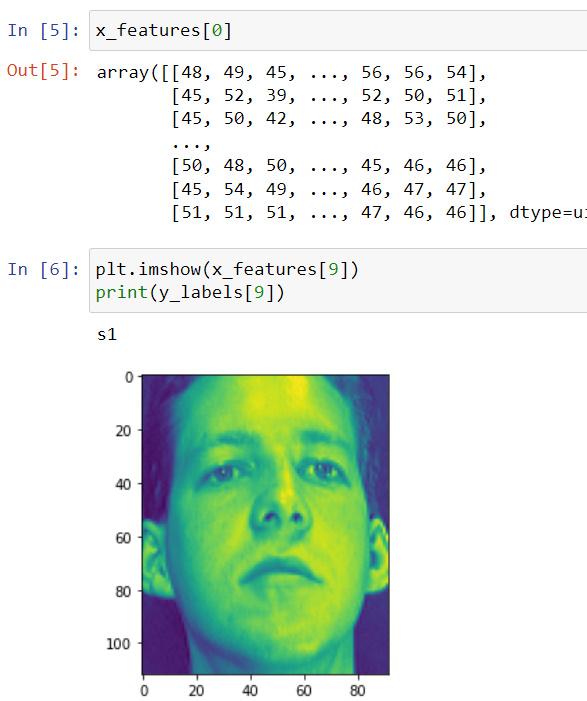
1. **Download the Dataset and Understand the Format (10Points)**

**a.ORL dataset is available at the following link .https://www.kaggle.com/kasikrit/att-database-of-faces/**

**b.The dataset has 10 images per 40 subjects. Every image is grayscale image of size 92x112.**

**The data consists of 40 classes each class contains 10 samples .**

**The data size is small and loaded in no time**

****

**Sample from the data**

****

**Each image has shape (112,92)**

**Each image is of type ndarray (n dimensional array)**

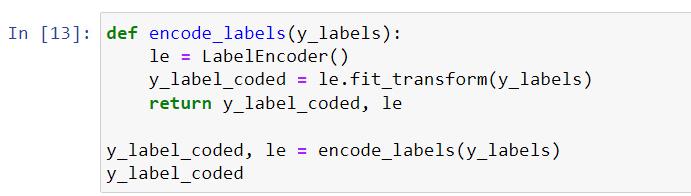
1. **Generate the Data Matrix and the Label vector (10 Points)**

**a.Convert every image into a vector of 10304 values corresponding to the image size.**

**b.Stack the 400 vectors into a single Data Matrix D and generate the label vector y**

****

**.The labels are integers from 1:40 corresponding to the subject id.**

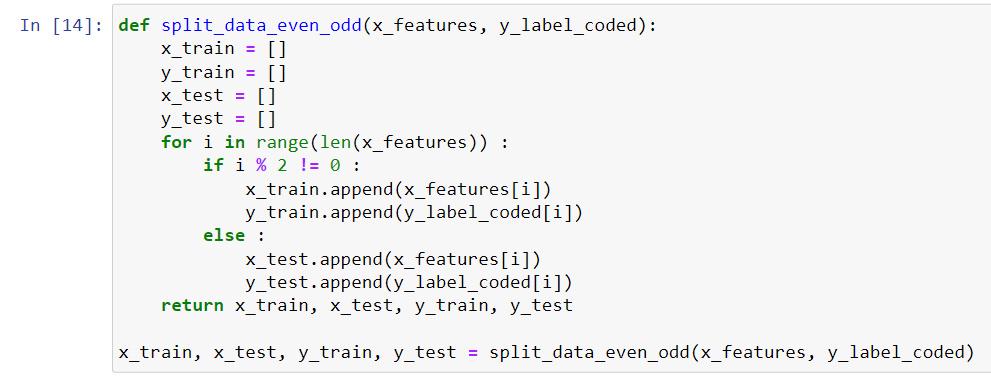
****

**We used label encoder to encode the data subjects**

1. **Split the Dataset into Training and Test sets (10 Points)**

**a.From the Data Matrix D400x10304 keep the odd rows for training and the even rows for testing. This will give you 5 instances per person for training and 5 instances per person for testing.**

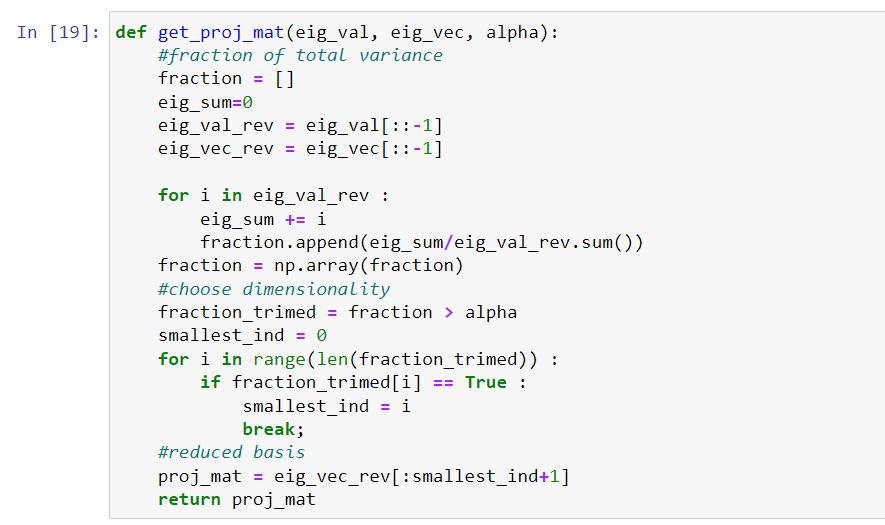
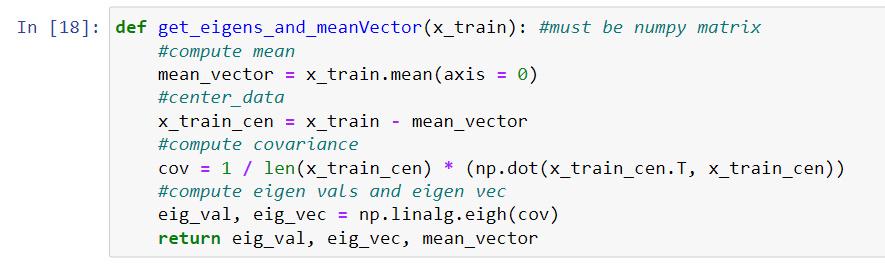
**b.Split the labels vector accordingly.**

****

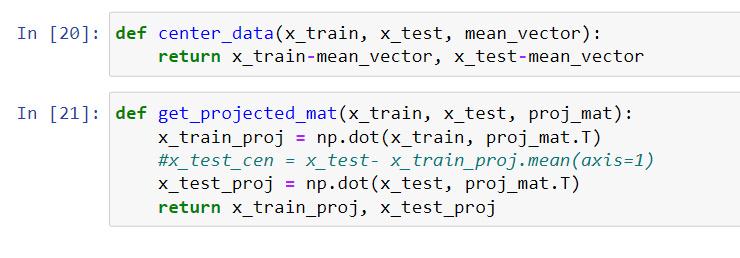
1. **Classification using PCA(30 points)**

**a.Use the pseudo code below for computing the projection matrix U.**

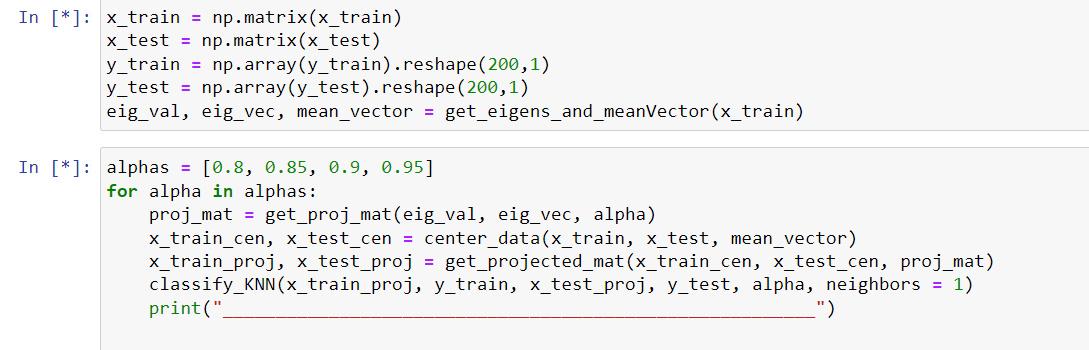
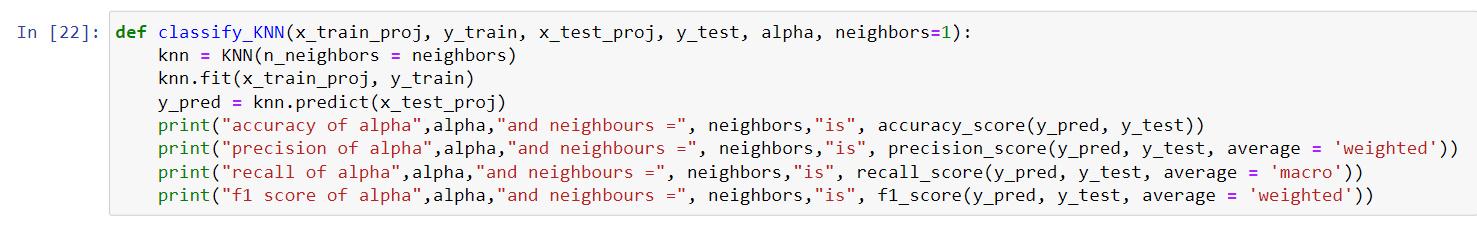
**Define the alpha = {0.8,0.85,0.9,0.95}**

****

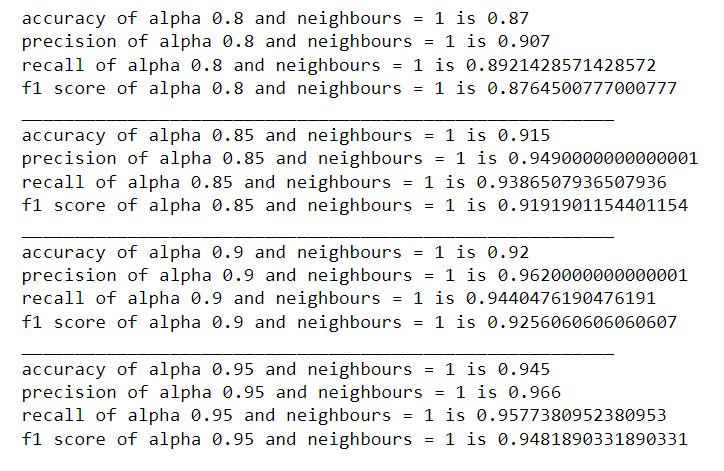
**b.Project the training set, and test sets separately using the same projection matrix.**

****

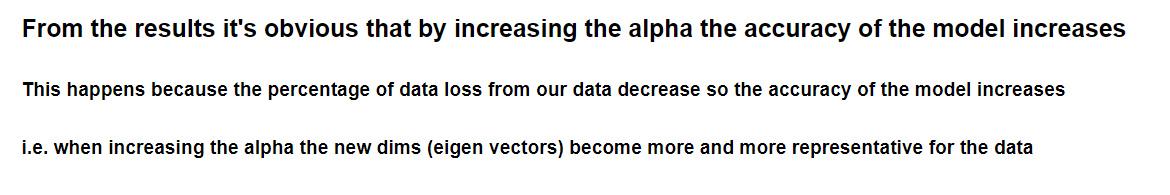
**c .Use a simple classifier (first Nearest Neighbor to determine the class labels).**

****

**d. Report Accuracy for every value of alpha separately.**

****

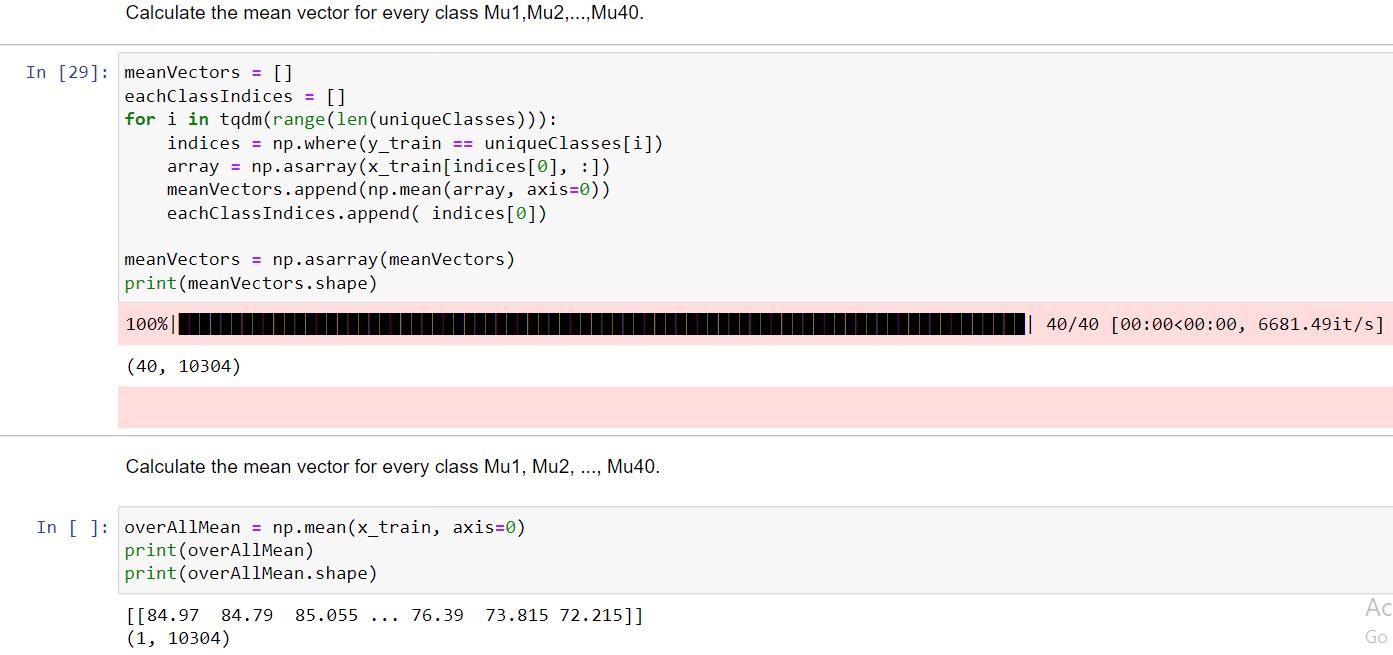
**e. Can you find a relation between alpha and classification accuracy?**

****

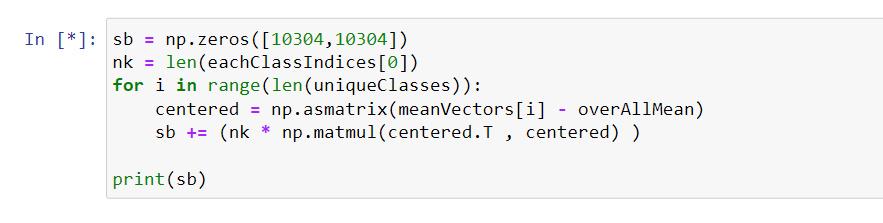
1. **Classification Using LDA (30 Points)**
2. **Use the pseudo code below for LDA. We will modify few lines in**

**pseudocode to handle multiclass LDA.**

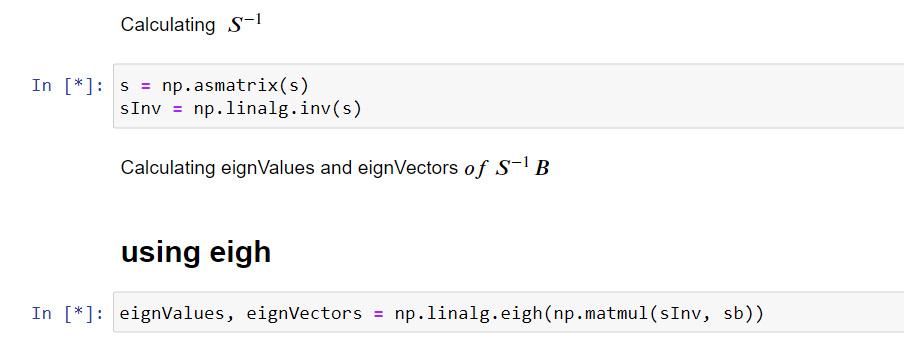
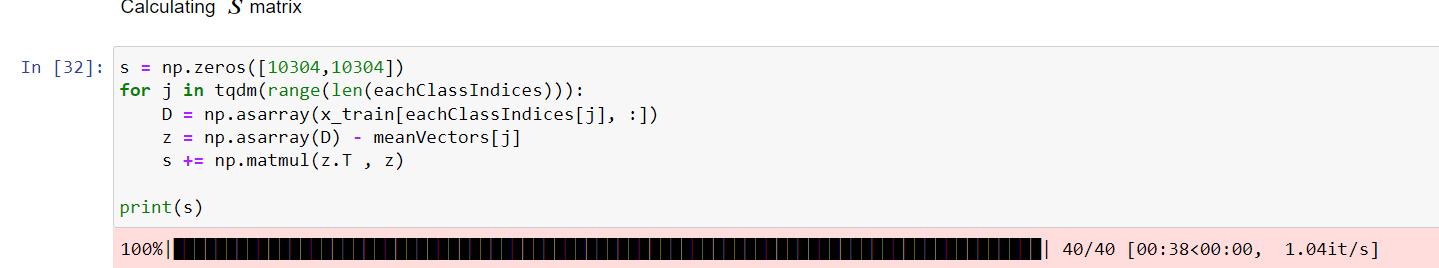
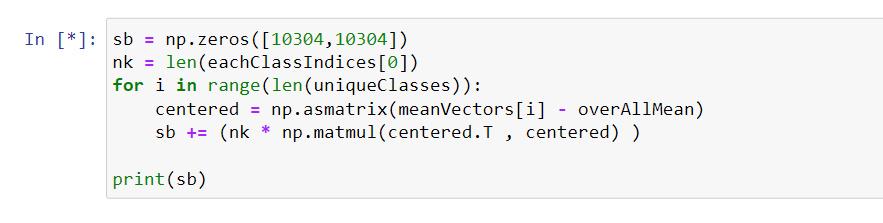
**i.Calculate the mean vector for every class Mu1,Mu2,..,Mu40.**

****

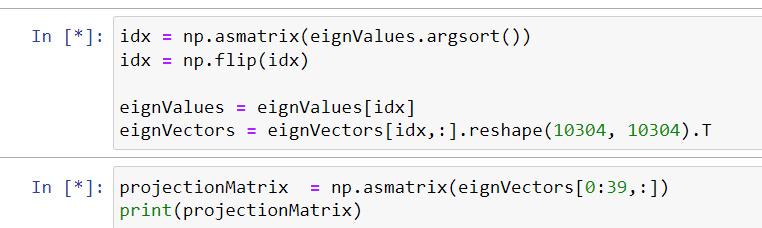
**ii.Replace B matrix by Sb .𝑆𝑏=∑𝑛𝑘(𝜇𝑘−𝜇)(𝜇𝑘−𝜇)𝑇**

****

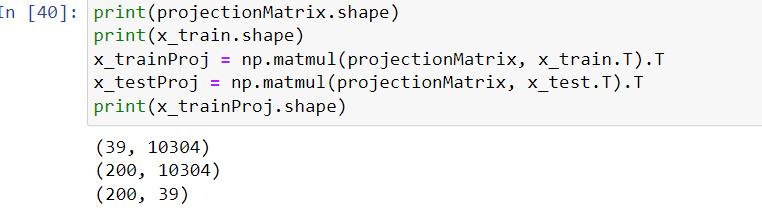
**iii.S matrix remains the same, but it sums S1,S2,S3,...S40.**

****

**iv.Use 39 dominant eigenvectors instead of just one. You will have a projection matrix U39x10304.**

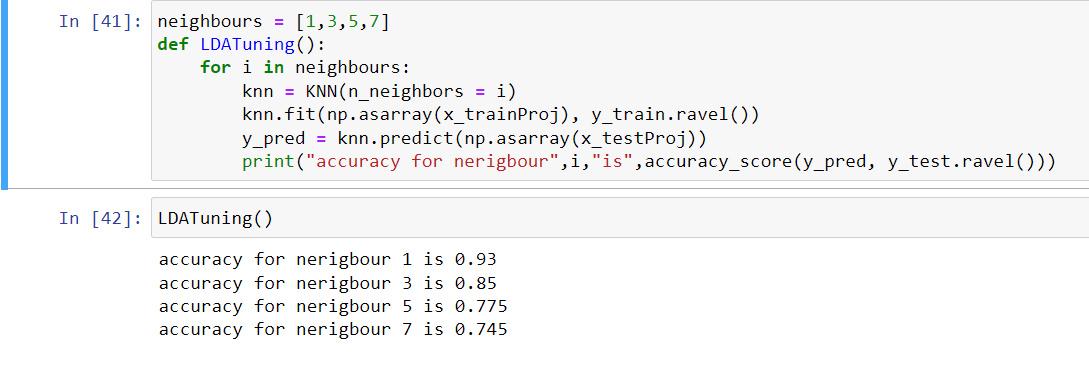
****

**b.Project the training set, and test sets separately using the same projection matrix U. You will have 39 dimensions in the new space.**

****

**c.Use a simple classifier (first Nearest Neighbor to determine the class labels).**

**d. Report accuracy for the multiclass LDA on the face recognition dataset.**

****

1. **Classifier Tuning (20 Points)**

**Plot (or tabulate) the performance measure (accuracy) against the K value. This is to be done for PCA and LDA as well.**

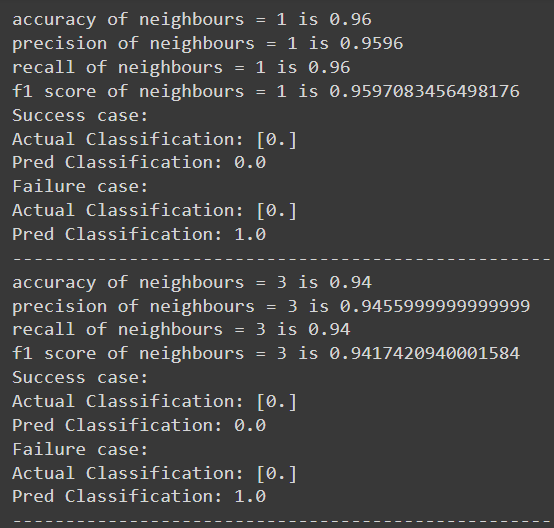
| **K** | **PCA** | **LDA** |
| --- | --- | --- |
| **1** | **94.5** | **93** |
| **3** | **83.5** | **85** |
| **5** | **78** | **77.5** |
| **7** | **73.5** | **74.5** |

1. **Compare vs Non-Face Images (15 Points)**

**Download non-face images and make them of the same size 92x112.**

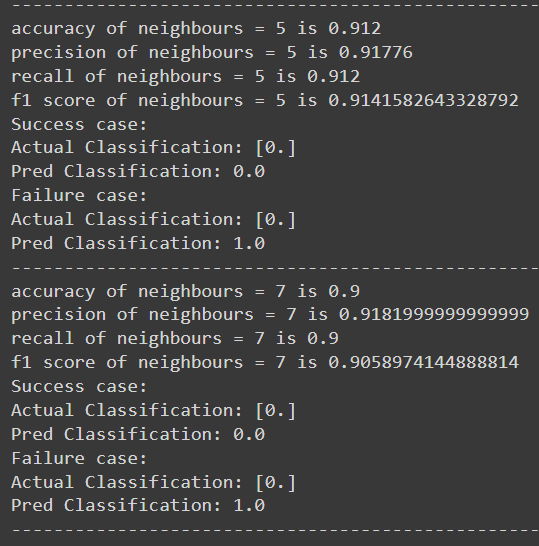
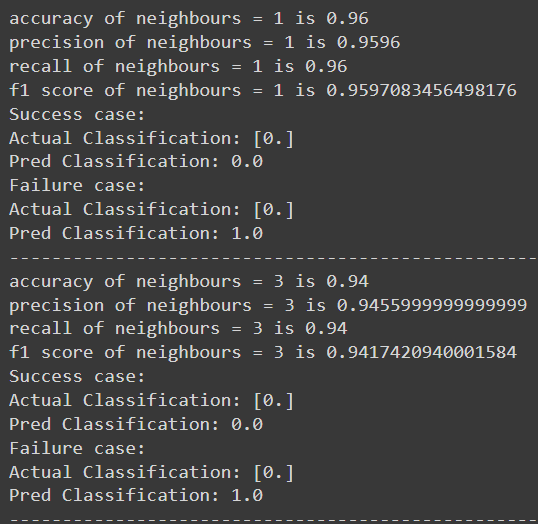
The data contains images of airplanes, cars, cats, dogs, fruits, flowers and motorbikes.

**i. Show failure and success cases.**

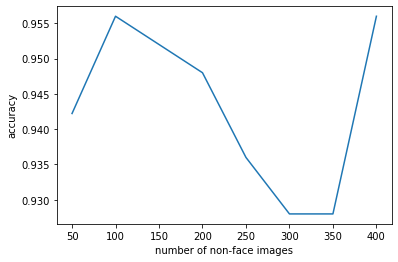
****

**ii. How many dominant eigenvectors will you use for the LDA solution?**

**Number of dominant eigenvectors = 10, as it has the highest accuracy, recall, precision and F1 score between the other numbers of dominant eigenvectors.**

****

**iii. Plot the accuracy vs the number of non-faces images while fixing the number of face images.**

****

**iv. Criticize the accuracy measure for large numbers of non-faces images in the training data.**

As we saw from the plot, the accuracy decreases when we increase the size of the non-face images whlile fixing the number of face images.

1. Bonus
2. **[5 points] Use different Training and Test splits. Change the number of instances per subject to be 7 and keep 3 instances per subject for testing. compare the results you have with the ones you got earlier with 50% split.**

| **Algorithm** | **Accuracy** |
| --- | --- |
| **PCA (50%)** | **94.5** |
| **PCA (70%)** | **92.4** |
| **LDA (50%)** | **94.5** |
| **LDA (70%)** | **95.8** |

1. **[10 points] There are other variations of PCA and LDA beyond the original algorithms. Please use one of the variations of PCA and one variations of LDA other than the original ones. Compare the time complexity and accuracy between the 2 different PCA and LDA models.**

| **Algorithm** | **Accuracy** | **Time(seconds)** |
| --- | --- | --- |
| **PCA** | **94.5** | **0.11 (without calculating eign vectors and values)**  **136.83 (total time including all steps)** |
| **LDA** | **93** | **0.01 (without calculating eign vectors and values)**  **363.1 (total time including all steps)** |
| **sklearn.PCA** | **97.5** | **0.24** |
| **sklearn.LDA** | **95** | **0.45** |