Lab Report

Problem 1

Task 0:

Synthetic dataset generation:
I created a dataset by first assuming random float values of weights (w0, w1, w2, w3, and w4) between -1 and 1, then random integer values of x1, x2, x3, and x4 were taken between -1000 and 1000, and a dataset of size 5000 was created. Labels were decided according to the value of f(x) (if >= 0, then 1 otherwise, 0). Data.txt was created consisting of 5000 samples with their labels in a space separated form and the first line containing the size of the dataset (5000).
Note that the dataset created is linearly separable.

Linear relation between weights and features:

$$f(x) = w0x0 + w1x1 + w2x2 + w3x3 + w4x4$$

Task 1:

Data Normalisation:
In order to ensure accurate training and convergence of the perceptron learning algorithm, I used L2 normalisation to normalise the train data.

Perceptron learning algorithm:
I trained the model with the normalised dataset using the perceptron learning technique. To lower the classification error, the algorithm adjusts the weights on a constant basis for a fixed number of epochs. Ultimately, the weights are adjusted to divide the linearly separable dataset. These weights are saved in weights.txt so that they can be accessed at the time of testing.

Input and Output:
The train.py file was generated, consisting of the implementation of a perceptron algorithm from scratch that takes train.txt as an argument for the train dataset and provides confirmation when weights.txt is generated and training is completed.

Task 2:

Testing code:
Using the trained weights, labels were predicted for the given test dataset (test.txt, which test.py takes as an argument) by using the above mentioned linear function f(x) such that if >= 0 label is 1 otherwise 0. Labels were then printed in a comma separated form.

Task 3:

For this task, we are expected to report the accuracy of our model by comparing it with training data that includes 20%, 50%, and 70% of our synthetic data. Note that the test data was not kept constant throughout this task but was taken as 80%, 50%, and 30% of the synthetic dataset, respectively.

Percentage of synthetic data used as training data	Accuracy
20	99.65%
50	99.44%
70	99.53%

Links:

Colab file having main.py and calAcc.py (refer the readme file)

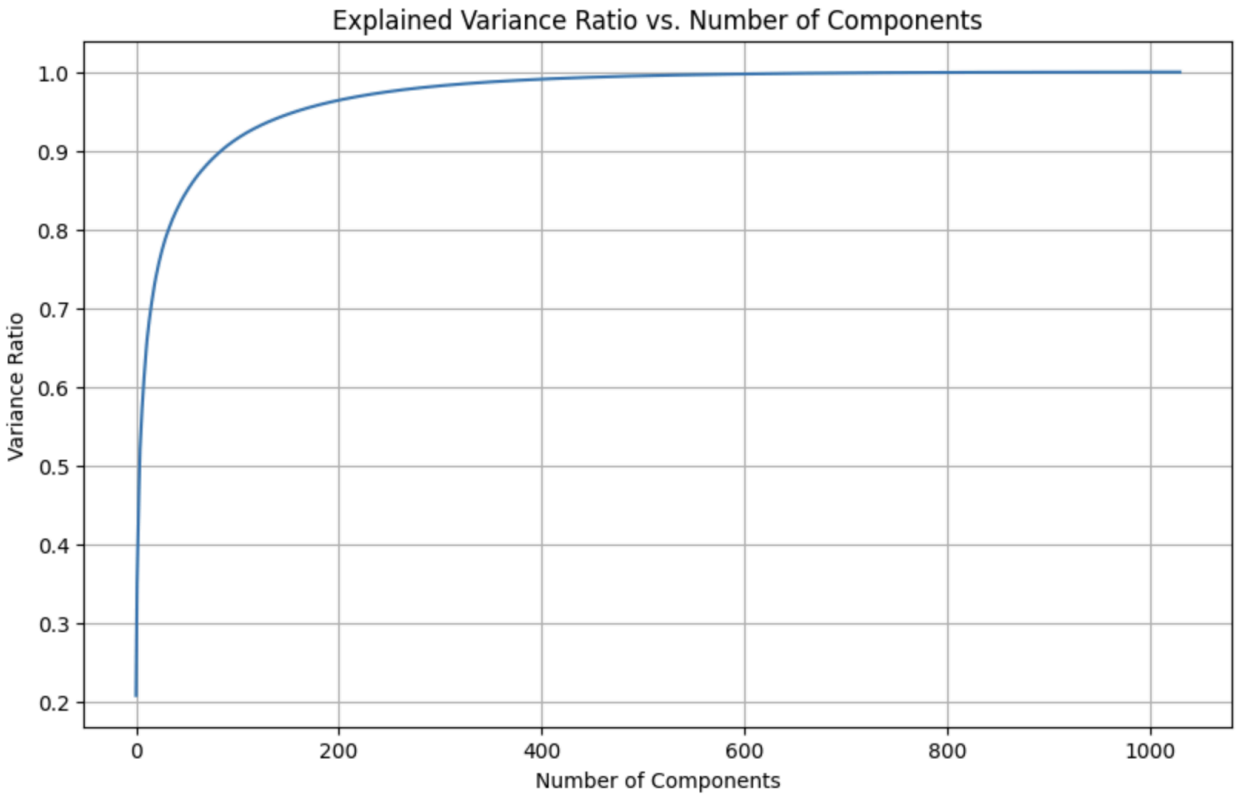
PROBLEM 2

Task 1:

LFW Dataset was loaded, threshold of minimum 70 images per target was set and images were resized to 40% of their original size. Then the dataset was splitted using 80-20 split.

Task 2:

In this task, we are supposed to implement PCA and for that we first need to decide the value of n_componenets for dimensionality reduction.
To decide this value I plotted a graph of Explained variance ratio v/s number of components. Usually the value of n_components which takes around 95% of the variance into account should be chosen. Hence 140 was chosen as the value and the graph for the same is attached below.



Initially the curve has a sharp incline and then the variance ratio doesn't increase much after reaching value around 95% and hence we take the value of n_component around 95% into account.

After that PCA is implemented on training data and the eigenfaces are derived from it which depicts the patterns found in the images for face recognition. Then we transform the training and test data using the fitted model of PCA.

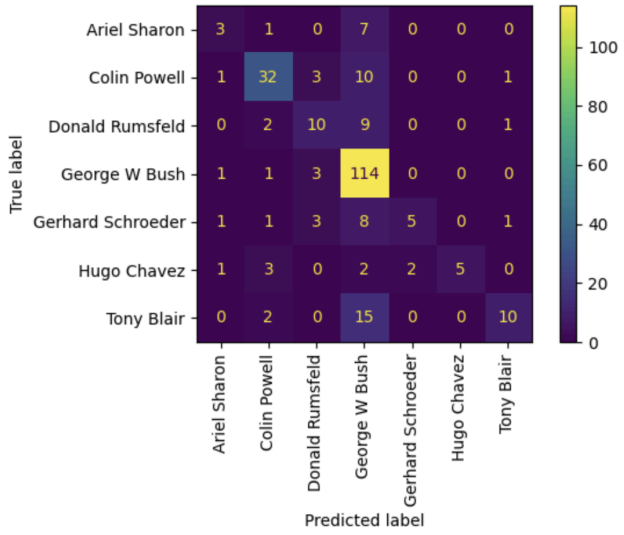
Task 3:

I choose KNN as the classifier as it is well suited for tasks like face recognition because its decision boundary method allows us to identify the patterns. Value of the nearest neighbour was chosen as 6 (K=6) and then the classifier was trained using the transformed data.

Task 4:

By using the transformed test data model was evaluated using classification report and confusion matrix as shown below.

	precision	recall	f1-score	support
Ariel Sharon	0.43	0.27	0.33	11
Colin Powell	0.76	0.68	0.72	47
Donald Rumsfeld	0.53	0.45	0.49	22
George W Bush	0.69	0.96	0.80	119
Gerhard Schroeder	0.71	0.26	0.38	19
Hugo Chavez	1.00	0.38	0.56	13
Tony Blair	0.77	0.37	0.50	27
accuracy			0.69	258
macro avg	0.70	0.48	0.54	258
weighted avg	0.70	0.69	0.67	258



Now, a subset of 10 eigenfaces were visualised and below are the observations, images on which model might fail and the ways to improve the model.

Observations:

1. Eigenfaces that capture lighting changes: Some eigenfaces may primarily represent lighting variations across face photos. These eigenfaces frequently show as patterns of light and dark patches, and they are critical for the model's ability to generalise across varied illumination environments.
2. Eigenfaces that capture facial traits include changes in the position of the eyes, nose, and mouth, as well as the general form and orientation of the face. These eigenfaces are critical for encoding structural distinctions between faces.

3. Eigenfaces can capture noise or unimportant fluctuations in training data. While these components may not have an impact on face recognition accuracy, they remain part of the eigenface representation.

Test images where model might fail:

1. Occluded faces: If test photos include partially occluded faces (e.g., sunglasses, hands, or hair), the model may have difficulty recognising them effectively. Eigenfaces are constrained by the variances in the training data, and if occlusions were not adequately captured during training, the model may fail to generalise to such circumstances.

2. Extreme lighting circumstances: If test images have extreme lighting conditions (for example, strong shadows or overexposure), the model may struggle to reliably recognise faces. While eigenfaces can capture some lighting fluctuations, they may not perform well under severe situations that were not sufficiently represented in the training data.

3. Non-frontal faces: Eigenfaces are typically trained using a dataset that contains primarily frontal faces. If test images include faces shot at various angles or orientations, the model may struggle to recognise them correctly. Eigenfaces have a limited ability to capture pose variations, therefore alternative techniques such as 3D modelling or extra training with augmented data may be required to increase performance in these circumstances.

Ways to improve the model:

1. Augment training data: Adding photographs with differences like as occlusions, harsh lighting situations, and varied stances can help the model learn to recognise faces under a variety of conditions.

2. Use more sophisticated techniques: While eigenfaces are effective for basic facial recognition tasks, more advanced techniques, such as deep learning-based approaches (e.g. CNN), may provide better performance, particularly when dealing with complex variations in facial appearance.

3. Combining eigenfaces with other facial recognition methods, such as local feature extraction algorithms (e.g. SIFT, SURF) or landmark-based approaches, can boost resilience and accuracy, particularly in difficult instances like occlusions and extreme poses.

4. Fine-tune parameters: Adjusting parameters like the number of main components kept during PCA and the regularisation strength might assist improve the model's performance for certain datasets and applications.

Task 5:

In this task we were supposed to observe the accuracy for different values of n_components and below given are the classification report for values other than 140 of n_components.

N_components = 150

	precision	recall	f1-score	support
Ariel Sharon	0.60	0.27	0.37	11
Colin Powell	0.79	0.72	0.76	47
Donald Rumsfeld	0.59	0.45	0.51	22
George W Bush	0.68	0.97	0.80	119
Gerhard Schroeder	0.71	0.26	0.38	19
Hugo Chavez	1.00	0.38	0.56	13
Tony Blair	0.83	0.37	0.51	27
accuracy			0.71	258
macro avg	0.74	0.49	0.56	258
weighted avg	0.72	0.71	0.68	258

N_components = 130

	precision	recall	f1-score	support
Ariel Sharon	0.25	0.18	0.21	11
Colin Powell	0.77	0.70	0.73	47
Donald Rumsfeld	0.43	0.41	0.42	22
George W Bush	0.69	0.92	0.78	119
Gerhard Schroeder	0.71	0.26	0.38	19
Hugo Chavez	1.00	0.38	0.56	13
Tony Blair	0.80	0.44	0.57	27
accuracy			0.68	258
macro avg	0.66	0.47	0.52	258
weighted avg	0.69	0.68	0.66	258

N_components = 160

	precision	recall	f1-score	support
Ariel Sharon	0.57	0.36	0.44	11
Colin Powell	0.74	0.74	0.74	47
Donald Rumsfeld	0.55	0.55	0.55	22
George W Bush	0.70	0.96	0.81	119
Gerhard Schroeder	0.83	0.26	0.40	19
Hugo Chavez	1.00	0.23	0.38	13
Tony Blair	0.80	0.30	0.43	27
accuracy			0.70	258
macro avg	0.74	0.49	0.54	258
weighted avg	0.72	0.70	0.67	258

N_components = 200

	precision	recall	f1-score	support
Ariel Sharon	0.60	0.27	0.37	11
Colin Powell	0.87	0.55	0.68	47
Donald Rumsfeld	0.67	0.27	0.39	22
George W Bush	0.59	0.97	0.73	119
Gerhard Schroeder	0.83	0.26	0.40	19
Hugo Chavez	1.00	0.23	0.38	13
Tony Blair	0.75	0.22	0.34	27
accuracy			0.64	258
macro avg	0.76	0.40	0.47	258
weighted avg	0.70	0.64	0.59	258

N_components = 100

	precision	recall	f1-score	support
Ariel Sharon	0.44	0.36	0.40	11
Colin Powell	0.68	0.77	0.72	47
Donald Rumsfeld	0.48	0.55	0.51	22
George W Bush	0.72	0.89	0.79	119
Gerhard Schroeder	0.89	0.42	0.57	19
Hugo Chavez	1.00	0.15	0.27	13
Tony Blair	0.83	0.37	0.51	27
accuracy			0.69	258
macro avg	0.72	0.50	0.54	258
weighted avg	0.72	0.69	0.67	258

Links: Colab file