DS3030: Data Analytics Lab

Assignment 8

Date: 22nd September, 2025
Timing: 2:00 to 5:00 PM
Max Marks: 18

Instructions

- Submit one `.ipynb` file containing all answers named as: **[student name]_assignment8.ipynb**
- Write the questions in **separate text blocks** before the answers.
- Write **justifications/comments** as required.

Part I: PCA Using EVD and SVD

Q1: PCA using Eigenvalue Decomposition (EVD)

Given a dataset(healthcare_dataset.csv) in CSV format, write a Python program to perform Principal Component Analysis (PCA) using the Eigenvalue Decomposition approach. Your code should:

1. Load the dataset and keep only numeric columns.
2. Center the data by subtracting the mean of each feature.[0.5]
3. Compute the covariance matrix and perform eigenvalue decomposition.[0.25]
4. Sort the eigenvalues and eigenvectors in descending order.[0.25]
5. Print the top 2 eigenvalues, explained variance ratio, and execution time for the PCA process.[1]

Q2: PCA with Singular Value Decomposition (SVD)

Given the same dataset(healthcare_dataset.csv), write a Python program to perform Principal Component Analysis (PCA) using the Singular Value Decomposition (SVD) approach. Your code

should:

1. Load the dataset and preprocess it in the same way as above.
2. Apply SVD on the centered data.[0.5]
3. Compute eigenvalues[0.5].
4. Print the top 2 eigenvalues, explained variance ratio, and execution time for the PCA process using SVD.[1]
5. Report the difference in execution time between the EVD and SVD approaches.
6. Explain which method is faster and why.[2]

Part II: Word Embeddings Analysis: PCA and t-SNE

- Use the vocabulary.txt file shared with you which contains words for various categories.

```
# Download and install the spaCy model
!python -m spacy download en_core_web_md
import spacy
# Load SpaCy's medium English model
nlp = spacy.load("en_core_web_md")
# Filter words that actually exist in the SpaCy model and have vectors
Educational_vocabulary = [] #get words from a text file shared with you.
selected_words = []
missing_words = []

# Shuffle the words to avoid bias in selection
# Take first 300 words for manageable analysis
# Extract word vectors (EMBEDDINGS)
```

Q3: PCA Analysis on Word Embeddings

1. Apply PCA to reduce dimensionality to 50 dimensions
2. Plot explained variance ratio for each principal component
3. Compare word relationships (cosine similarity) before and after PCA reduction for these word pairs:

```
word_pairs = [
     ('king', 'queen'), ('man', 'woman'), ('good', 'bad'),
     ('big', 'small'), ('happy', 'sad'), ('dog', 'cat'),
     ('car', 'vehicle'), ('book', 'read'), ('water', 'drink'),
     ('computer', 'technology')
 ]
```

Q4.t-SNE Analysis on Word Embeddings

Using the 50-dimensional word embeddings from the previous question, perform t-SNE dimensionality reduction.

1. Parameter Exploration and Optimization:

- Experiment with different perplexity values: [5, 30, 50, 100]
- Try different learning rates: [10, 200, 1000]
- Based on your exploration, select the best combination of parameters and justify your choice

2. Comparison Analysis:

- Compare word relationships (cosine similarity) before and after t-SNE reduction for the same word pairs from previous question.
- Create a comparison table showing: Original similarity, PCA similarity, t-SNE similarity.

Optional

2D Visualization:

- Further reduce the 50-dimensional embeddings to 2 dimensions using PCA

1. Create a scatter plot showing words in 2D space
2. Label each point with its corresponding word
3. Analyze and discuss the clustering patterns of semantically related words

- Analysis:

1. Identify groups of similar words that cluster together in the 2D space
2. Discuss how well the 2D PCA representation preserves semantic relationships

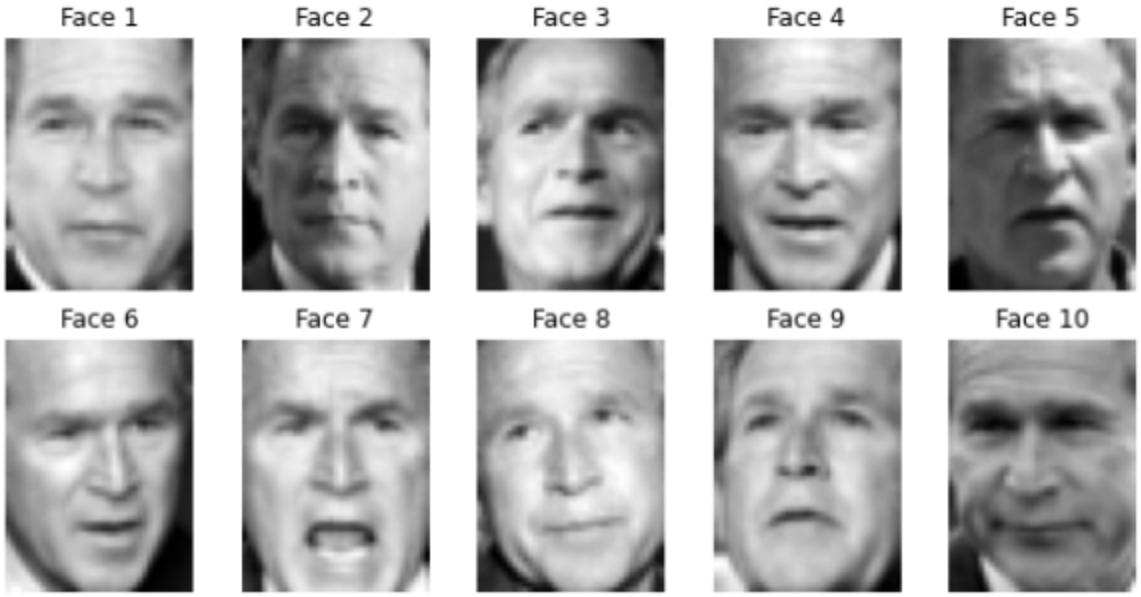
Part III: EXPLORING PCA IN VISION DATASETS

Q5: Eigenfaces Generation (3 marks)

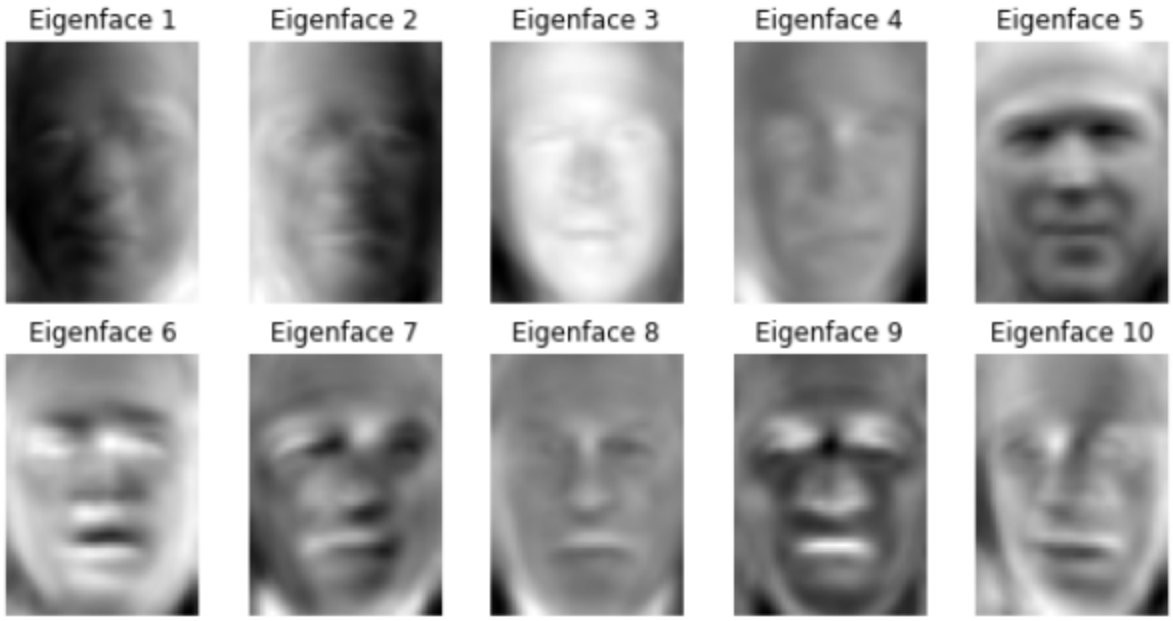
Generate and visualize the first (top) 10 eigenfaces of the classic vision dataset LFW.

WORKFLOW :

1. Load the dataset from sklearn and visualize 10 original images from it. (You may load with min_faces_per_person=500). The output should be similar to the following :



   (HINT: Use the images component of the loaded dataset)
2. Apply PCA on the corresponding flattened array with your choice for the appropriate number of components.
3. Visualize the first 10 principal components by reshaping them. The output should be similar to the following :



Q6: PCA RECONSTRUCTION (3 marks)

Transform a sample of 5 images to the PCA space and reconstruct them back using the inverse transformation. Visualize original and reconstructed images side by side

WORKFLOW :

1. Transform any set of 5 randomly chosen images (flattened array) using PCA
2. Inverse transform the images, reshape, and visualize. The output should be similar to the following :

Original Reconstructed


Original Reconstructed


Original Reconstructed


Original Reconstructed


Original Reconstructed