Assignment 3: Geometric Intelligence (SVD & PCA)

1 Instructions

In this assignment, you will move beyond simply calculating answers and focus on **geometric intuition** and **exploratory analysis**. You will deconstruct matrices to see them as geometric operators (SVD) and apply those insights to uncover structure in high-dimensional datasets (PCA).

Expectations on AI Use: Unlike Assignment 2, where you implemented algorithms from scratch, here you are encouraged to use standard libraries (`numpy`, `matplotlib`, `sklearn`). You may use LLMs to generate the boilerplate code for plotting and data loading. **However**, the grading focus shifts entirely to your ability to **interpret** the results, **construct** specific geometric scenarios, and **critique** the limitations of linear methods.

1.1 Submission Requirements

You must submit two components to Gradescope:

1. **The Report:** A plaintext Markdown document which you will paste directly into a Gradescope submission text box. This is the primary artifact for grading. **IMPORTANT:** This document must be text-only. Do not attempt to embed images. You will describe your visual findings using precise language and numerical data.

- **Character Limit:** 7,500 characters (roughly 1.5 - 2 pages of single-spaced text).

2. **Supplemental Material:** You must upload your `.ipynb` notebook containing all code, visualizations, and raw experimental results.

2 Part 1: Implementation (The Notebook)

You will author a Jupyter Notebook containing two primary experiments.

2.1 Experiment 1: The Anatomy of a Linear Transformation (SVD)

We learned in class that **every** matrix A can be decomposed into a sequence of three geometric operations: $A = U\Sigma V^T$.

$$\text{Input } x \xrightarrow{\text{Rotate/Reflect } (V^T)} \text{Aligned Space} \xrightarrow{\text{Scale/Collapse } (\Sigma)} \text{Scaled Space} \xrightarrow{\text{Rotate/Reflect } (U)} \text{Output } b$$

2.1.1 Task 1.1: The `visualize_svd` Function

Implement a function `visualize_svd(A)` that takes a 2×2 matrix A and visualizes its effect on a **unit square** defined by a grid of points (or a distinctive shape like the letter 'F').

Your function must generate a **single figure with 4 subplots** showing the progression of the data: 1. **Original Data:** The initial unit square/shape. 2. **Step 1 (V^T):** The data after multiplying by V^T. (Rotation/Reflection) 3. **Step 2 (ΣV^T):** The data from Step 1 after multiplying by Σ. (Scaling/Stretching) 4. **Step 3 ($U\Sigma V^T$):** The data from Step 2 after multiplying by U. (Final Rotation/Reflection)

Note: Use `numpy.linalg.svd` to get the components.

2.1.2 Task 1.2: The "Matrix Architect"

Using your visualization tool, you must manually construct (by defining specific U, Σ, and V^T matrices) and visualize matrices that perform the following specific geometric tasks. You cannot just pick random numbers; you must design the components to achieve the effect.

1. **The Collapse:** A matrix that projects 2D space onto a 1D line at a 45° angle.
2. **The Invertible Shear:** A matrix that shears the unit square (looking like a rhombus) but maintains full rank (no zero singular values).
3. **The Mirror:** A pure reflection across the Y-axis (no scaling, no rotation).

2.2 Experiment 2: Exploratory Data Analysis via PCA

You will use `sklearn.decomposition.PCA` to explore the "latent space" of real-world datasets. You must interpret the components not just as abstract vectors, but as "directions of maximum variance."

2.2.1 Task 2.1: Eigenfaces (The LFW Dataset)

Load the Labeled Faces in the Wild (LFW) dataset using `sklearn.datasets.fetch_lfw_people` (use `min_faces_per_person=70` to keep it small).

1. **Compute PCA:** Fit PCA to the face images.
2. **Visualize Components:** Plot the "Mean Face" and the top 5 "Eigenfaces" (principal components reshaped back into images).
3. **Reconstruction:** Show an original face and its reconstruction using $k = \{10, 50, 100, 200\}$ components.
4. **Error Analysis:** Compute the Mean Squared Error (MSE) for each reconstruction level.

2.2.2 Task 2.2: The Failure of Linearity (Swiss Roll)

PCA assumes data lies on a linear subspace (a flat sheet). Test this assumption on a non-linear manifold.

1. **Generate Data:** Use `sklearn.datasets.make_swiss_roll` to generate 1000 points.
2. **Apply PCA:** Project the 3D data down to 2D using PCA.
3. **Visualize:** Plot the 2D projection, coloring the points by their position on the roll (the univariate label).
4. **Analyze:** Does the 2D projection successfully "unroll" the Swiss Roll? Why or why not?

2.2.3 Task 2.3: Sensitivity to Outliers

PCA minimizes squared error (L_2 norm), which is notoriously sensitive to outliers.

1. **Generate Data:** Create a simple 2D dataset with a strong linear correlation (e.g., $y = x + \text{noise}$).
2. **Corrupt Data:** Add **one** massive outlier point far away from the main cluster.
3. **Compare:** Fit PCA to the "Clean" data and the "Corrupted" data.

4. **Quantify:** Compute the angle (in degrees) between the first principal component of the clean data vs. the corrupted data.

---

3 Part 2: Content Requirements (The Report)

Your report must be organized into **exactly five sections** with Markdown headers. **Do not include images.** Use text, tables, and specific numbers to describe your findings.

3.1 Section 1: Executive Summary & Key Insight

- **One-Sentence Takeaway:** A single sentence summarizing the most important geometric intuition you developed regarding how matrices manipulate data.
- **Summary Paragraph:** Briefly describe the datasets you analyzed and the key distinction you observed between linear (PCA) and non-linear data structures.

3.2 Section 2: The Geometry of SVD

- **Descriptive Analysis:** Describe the transformation of the unit square in Task 1.1 using precise geometric language. (e.g., "The square was first rotated 30 degrees clockwise, then stretched along the x-axis by a factor of 2…")
- **Matrix Architecture:** For the "Collapse" task (Task 1.2), provide the **exact numerical matrices** you constructed for U, Σ, and V^T. Explain *why* these values achieved the target effect (e.g., "I set $\Sigma_{2,2} = 0$ to collapse the second dimension, and set U to…").
  - *Format Idea:* Use a Markdown table or LaTeX matrix notation to clearly present your constructed matrices.

3.3 Section 3: PCA & Interpretability (Eigenfaces)

- **Interpreting "Ghost" Faces:** Describe the visual features captured by the first 2 Principal Components. Do NOT paste the images. Instead, use descriptive language (e.g., "PC1 appears to capture the lighting direction, showing a gradient from left to right," or "PC2 captures the difference between smiling and neutral expressions").
- **Reconstruction Analysis:** Report the **Mean Squared Error (MSE)** values for $k = \{10, 50, 100, 200\}$. Discuss the trade-off: At what k did the face become recognizable as a specific person? At what k were fine details (like glasses or teeth) resolved?

3.4 Section 4: The Limitations of Linearity

- **Swiss Roll Failure:** Describe the 2D scatter plot of the Swiss Roll projection. Did the colors (representing the manifold structure) separate cleanly, or did they overlap? Explain **geometrically** why PCA failed here (reference "Euclidean distance" vs "Geodesic distance").
- **Outlier Sensitivity:** Report the **angle of deviation** (in degrees) caused by the single outlier in Task 2.3. Explain why the L_2 norm objective function forces the principal component to tilt towards the outlier.

3.5 Section 5: Reflection

- **Geometric "Aha!" Moment:** Describe a specific moment where the code output contradicted your mental model of linear algebra.
- **LLM Usage:** How did you use LLMs for this assignment? Did the LLM struggle with the "Matrix Architect" task (constructing matrices for specific geometric effects)?

---

4 Grading Rubric

Each of the five sections is weighted equally (**20% each**).

Criterion	Excellent (5)	Good (4)	Satisfactory (3)	Okay (2)	Poor (1)
Section 1: Executive Summary	Takeaway is profound and geometrically grounded. Summary clearly contrasts linear vs non-linear behaviors.	Takeaway is clear. Summary covers the main tasks.	Takeaway is generic. Summary is present but vague.	Summary misses key elements of the analysis.	Missing.
Section 2: SVD Geometry	"Collapse" matrices are correct and explanation demonstrates deep understanding of how Σ controls rank and U controls orientation.	Explanation identifies the correct components but explanation is slightly mechanical.	Matrices are provided, but explanation of the construction is weak or relies on trial-and-error.	Matrices are incorrect or missing explanation.	Missing.
Section 3: Eigenfaces	Insightful descriptive analysis of PC features (lighting vs structure). Reconstruction discussion is grounded in specific MSE values.	Good description and reasonable discussion of features.	MSE values present. Discussion states the obvious (e.g., "error went down").	Missing MSE values or descriptions are too vague.	Missing.
Section 4: Limitations	Clearly articulates *why* PCA fails on manifolds and *why* outlier sensitivity occurs, referencing specific results (angle change).	Correctly identifies the failure modes with good evidence.	Describes the failure but struggles to explain the "why".	Interpretation is incorrect (e.g., claiming PCA worked on the Swiss Roll).	Missing.

Criterion	Excellent (5)	Good (4)	Satisfactory (3)	Okay (2)	Poor (1)
Section 5: Reflection	Honest, specific reflection connecting code to geometric theory. Critically evaluates LLM performance on geometric reasoning.	Thoughtful reflection on the learning process.	Generic reflection (e.g., "I learned about PCA").	Minimal effort.	Missing.