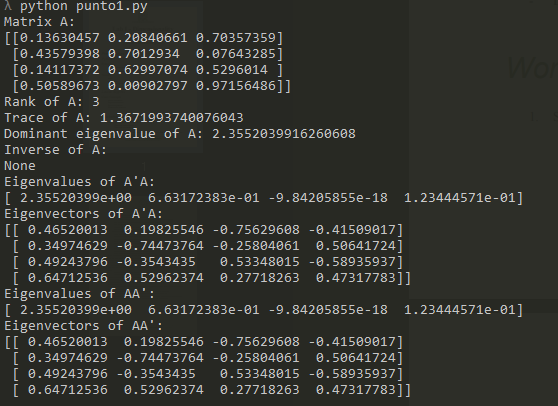
TALLER 1 REDUCCIÓN DE DIMENSIONALIDAD

ANDRES FELIPE ORREGO QUINTERO

1041204802

1. 

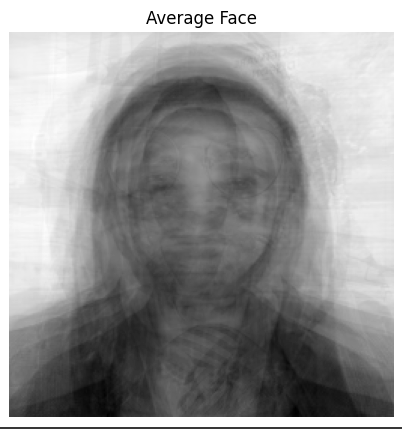
It is not posible to have the inverse of a matrix which is nos square.

As it´s seen, EigenVectors and Eigenvalues of A’A and AA’ are the same, it opens the possibilities to do more things with matrixes.

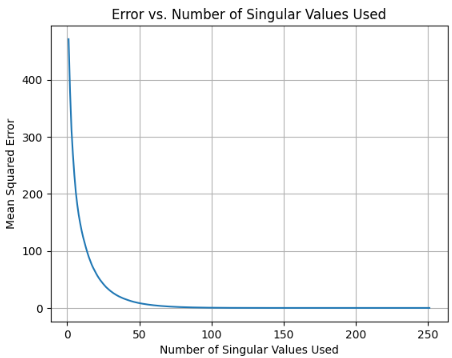
1. This is my own photo in grayscale:

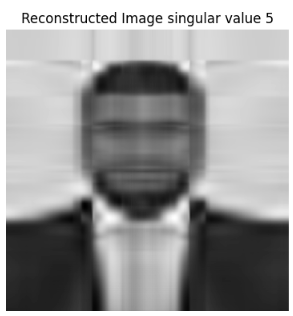


I calculate the average of all the faces of the course and have this answer:

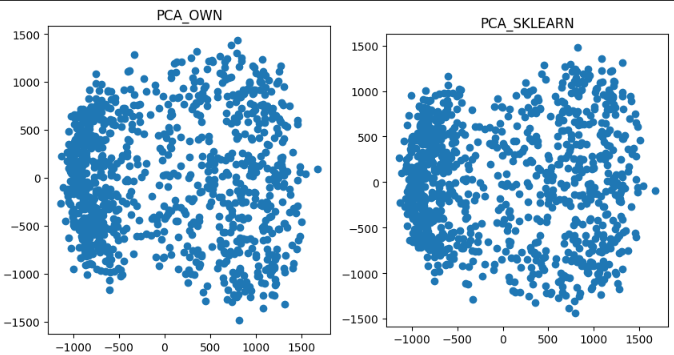


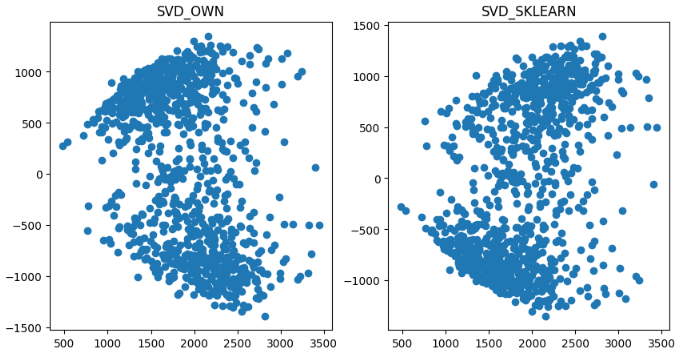
1. The library is in the code
2. Applying SVD and changing the singular values, I plot the next figured.



When the mean squared error is near to zero, the image looks like the original, I plot different photos with different singular values and this is the result. 

1. The code is in the repository
2. Points 6 and 7 are in the same code These are the results





8. To make PCA algorithm more robust, there are different things to try.

1. - **Data Preprocessing:**
   * **Standardization:** Prior to applying PCA, standardize the data by subtracting the mean and scaling to unit variance (z-score normalization). This step ensures that all features have a similar scale and prevents features with large variances from dominating the principal components.
   * **Outlier Handling:** Robustly handle outliers in the data, as outliers can have a significant impact on PCA results. Consider using robust statistical methods or outlier detection techniques to identify and mitigate the influence of outliers on the covariance matrix.
2. **Regularization:**
   * **Shrinkage Estimators:** Incorporate regularization techniques, such as shrinkage estimators (e.g., Ledoit-Wolf or OAS), when estimating the covariance matrix. Regularization helps stabilize eigenvalue computations, especially when dealing with high-dimensional data or small sample sizes. It can also improve robustness against noise and multicollinearity.
3. **Cross-Validation for Component Selection:**
   * **Cross-Validation:** Use cross-validation to select the optimal number of principal components. Instead of arbitrarily choosing the number of components, cross-validation helps identify the most informative components while avoiding overfitting. Implement techniques like k-fold cross-validation or leave-one-out cross-validation to determine the appropriate dimensionality reduction.

**9.** UMAP, which stands for Uniform Manifold Approximation and Projection, is a dimensionality reduction and manifold learning technique. It is based on several mathematical principles and is useful for various applications in machine learning and data analysis. Here are the underlying mathematical principles behind UMAP and its common uses:

Mathematical Principles Behind UMAP:

1. Manifold Learning: UMAP is built upon the foundation of manifold learning. It assumes that high-dimensional data often lies on or near a lower-dimensional manifold within the high-dimensional space. The goal is to learn a faithful representation of this manifold.

2. Fuzzy Topological Structure: UMAP aims to preserve the fuzzy topological structure of data points. It uses a local fuzzy set membership function to measure how connected data points are in both the high-dimensional and low-dimensional spaces. This fuzziness allows UMAP to capture complex relationships.

3. Graph-Based Approach: UMAP constructs a graph representation of the data, where data points are nodes, and edges represent the connectivity or similarity between points. It uses a metric such as Euclidean distance or a more complex distance metric like a geodesic distance on the manifold.

4. Optimization: UMAP formulates the problem as an optimization task. It seeks to optimize the low-dimensional representation of the data such that the fuzzy topological structure is preserved. This is done through an optimization process that balances the preservation of local and global relationships.

5. Stochastic Gradient Descent (SGD): UMAP utilizes stochastic gradient descent to iteratively optimize the low-dimensional embedding. This allows it to find a suitable representation efficiently.

Usefulness of UMAP in Machine Learning:

1. Dimensionality Reduction: UMAP is primarily used for dimensionality reduction. It can reduce the number of dimensions in high-dimensional data while retaining its essential structure. This is especially useful for visualizing and exploring complex datasets.

2. Visualization: UMAP is valuable for data visualization tasks. It can be used to create 2D or 3D visualizations of high-dimensional data, making it easier to interpret and analyze complex datasets.

3. Clustering and Pattern Recognition: UMAP can help improve clustering and pattern recognition tasks by providing a more informative representation of the data. It can be used as a preprocessing step for clustering algorithms, and it often outperforms traditional dimensionality reduction methods like PCA for such tasks.

4. Anomaly Detection: UMAP can be employed for anomaly detection by reducing high-dimensional data to a lower-dimensional space and then identifying data points that deviate significantly from the norm.

5. Bioinformatics: UMAP has found significant applications in bioinformatics, particularly in the analysis of single-cell RNA sequencing data. It helps visualize and cluster cells in high-dimensional gene expression datasets, enabling researchers to understand cellular heterogeneity.

6. Recommendation Systems: UMAP can be used in recommendation systems to reduce the dimensionality of user-item interaction data, making it easier to suggest relevant items to users.

In summary, UMAP is a versatile dimensionality reduction and manifold learning technique that leverages mathematical principles related to preserving fuzzy topological structure in data. Its primary utility lies in visualizing and exploring high-dimensional data, improving clustering and pattern recognition, and finding applications in various domains, including bioinformatics and recommendation systems.

**10**. LDA, which stands for Linear Discriminant Analysis, is a dimensionality reduction and classification technique used in machine learning and statistics. It is based on several mathematical principles and is useful for various applications in machine learning. Here are the underlying mathematical principles behind LDA and its common uses:

Mathematical Principles Behind LDA:

1. Maximizing Class Separation: The primary goal of LDA is to maximize the separation between classes in the data. It does this by finding a projection (linear transformation) of the data points onto a lower-dimensional space such that the distances between the class means are maximized, and the scatter (variance) within each class is minimized.

2. Between-Class Scatter and Within-Class Scatter: LDA quantifies class separation by considering two types of scatter:

- Between-Class Scatter: This measures the spread of class means in the projected space. LDA aims to maximize this scatter, as larger separation between class means leads to better class discrimination.

- Within-Class Scatter: This measures the spread of data points within each class in the projected space. LDA aims to minimize this scatter to ensure that data points within each class are tightly clustered.

3. Eigenvector Decomposition: LDA involves solving an eigenvalue problem to find the eigenvectors and eigenvalues of a matrix formed by the combination of the within-class scatter and between-class scatter matrices. These eigenvectors correspond to the directions (linear combinations of features) in the original space that maximize class separation.

Usefulness of LDA in Machine Learning:

1. Dimensionality Reduction: LDA is commonly used for dimensionality reduction. It reduces the number of dimensions in the data while preserving the most discriminative information. It projects the data onto a lower-dimensional subspace that maximizes class separability.

2. Classification: LDA is often used as a classification algorithm, especially in cases with multiple classes. It uses the learned linear transformation to project new data points into the lower-dimensional space and assigns them to the class with the nearest mean.

3. Feature Selection: LDA can serve as a feature selection technique. It ranks features based on their discriminative power and selects the top features for use in classification or other tasks.

4. Data Visualization: LDA can be employed for data visualization by projecting high-dimensional data onto a lower-dimensional space (usually 2D or 3D). This helps visualize the separability of classes and may aid in exploratory data analysis.

5. Reducing Overfitting: In some cases, LDA can help reduce overfitting in machine learning models by reducing the dimensionality of the input features and focusing on the most relevant information.

6. Face Recognition: LDA has been used extensively in face recognition applications. It helps identify facial features that are most discriminative for distinguishing between different individuals.

7. Speech Processing: LDA has applications in speech processing, particularly in speaker recognition and speech classification tasks.

In summary, LDA is a mathematical technique that focuses on maximizing class separation while reducing within-class scatter. It is useful for dimensionality reduction, classification, feature selection, data visualization, and improving the performance of machine learning models, especially in cases with multiple classes and high-dimensional data.