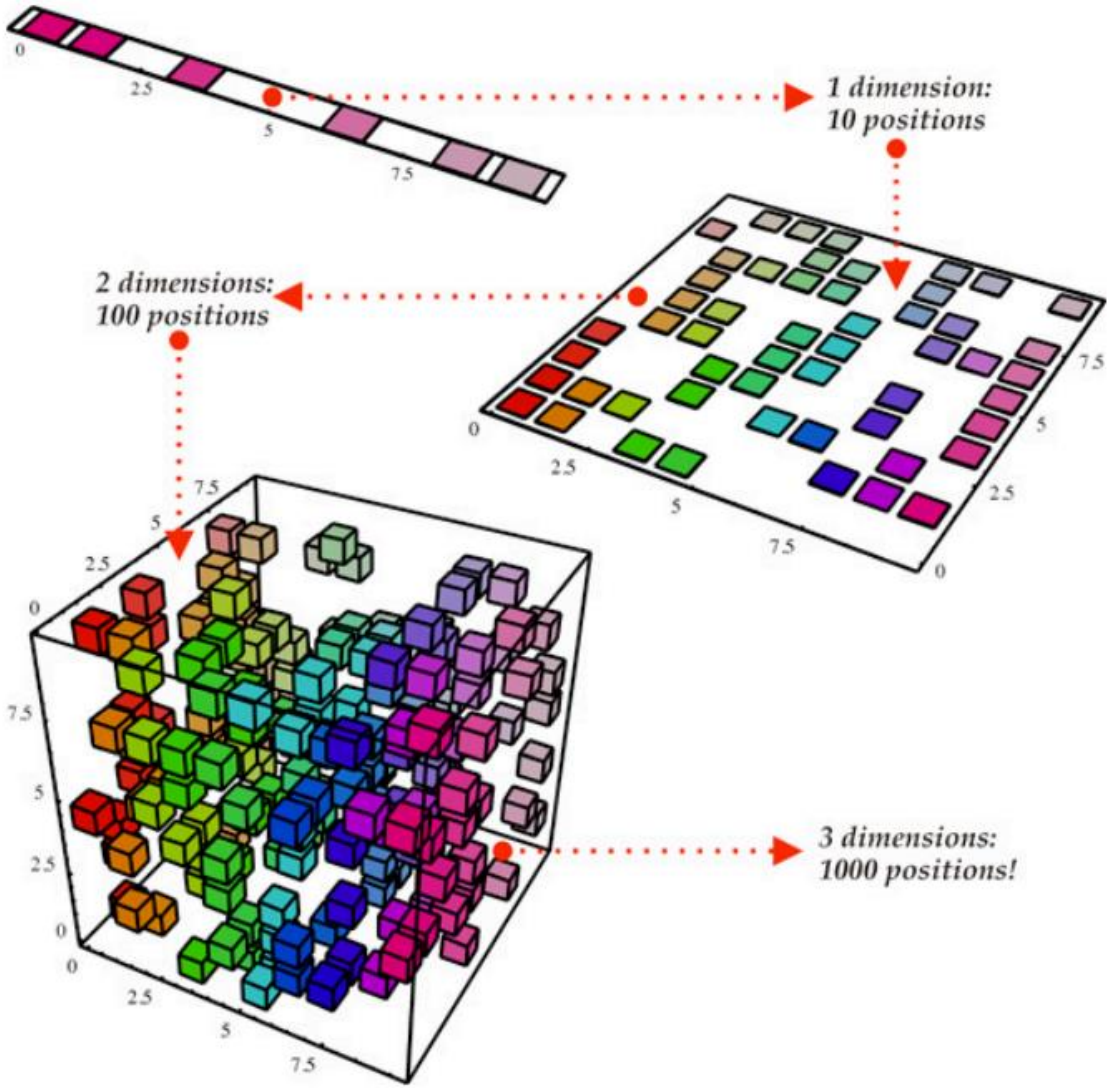
DIMESIONALITY REDUCTION

KUYE DOLUWAMU TAIWO

Motivation

- High-dimensional datasets are common in many scientific and engineering domains, such as computer vision, bioinformatics, physics, and finance. However, the curse of dimensionality, or the inherent difficulty of dealing with high-dimensional data, can cause a variety of issues such as overfitting, sparsity, and computational complexity [Ha09].



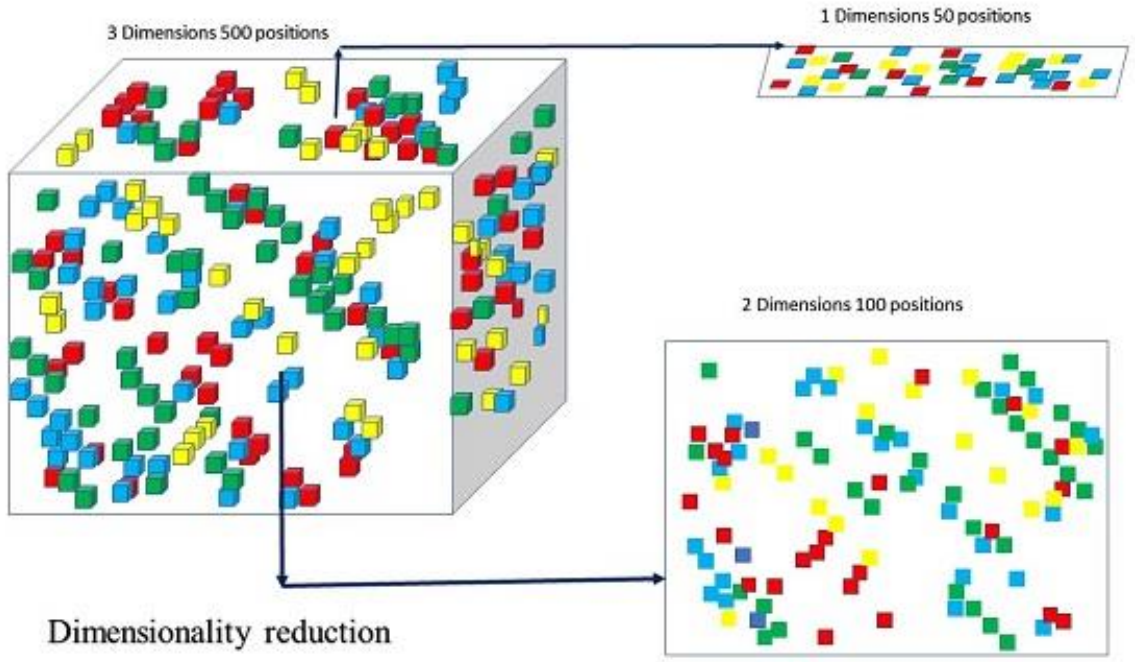
Motivation

In high-dimensional datasets, the number of features can be significantly greater than the number of samples, which can lead to overfitting and poor generalization performance. High-dimensional datasets can be difficult to visualize and analyse [AW10].

By reducing the dimensionality, we can gain a better understanding of the correlations between the features and the samples.

Dimensionality Reduction

- Dimensionality reduction is a machine learning technique that seeks to minimize the number of features or dimensions in a dataset while retaining much useful Information as feasible [Gé22].
- The importance of dimensionality reduction is roo the curse of dimensionality [Gé22].



Dimensionality Reduction

- Dimensionality reduction can be approached in two ways: feature selection and feature extraction.
- Feature selection is concerned with identifying and removing irrelevant or redundant features based on some criterion, such as correlation or mutual information.
- Feature extraction is concerned with transforming the original features into a lower-dimensional representation that maintains the most relevant information [RM17].

Application

Data visualization is one of the most common applications of dimensionality reduction. High-dimensional data can be difficult to visualize and understand, especially when the relationships between its features are complicated [LH08].

Feature selection is another application of dimensionality reduction. Not all features in the high-dimensional data are useful or informative for the task at hand in some situations [RM17].

Challenges

Technique selection: One of the most difficult aspects of dimensionality reduction is determining what technique is most appropriate for a given dataset [Va09].

Data loss: Another challenge with dimensionality reduction is the possible loss of information that can occur when a dataset's dimensionality is reduced [RM17].

Evaluation Using PCA

- To evaluate dimensionality reduction, I used the PCA technique to reduce a collection of data from the Iris dataset that had measurements of the sepal length, sepal width, petal length, and petal width of three kinds of iris flowers.
- The goal is to reduce this dataset from 4 dimensions into 2.

G9

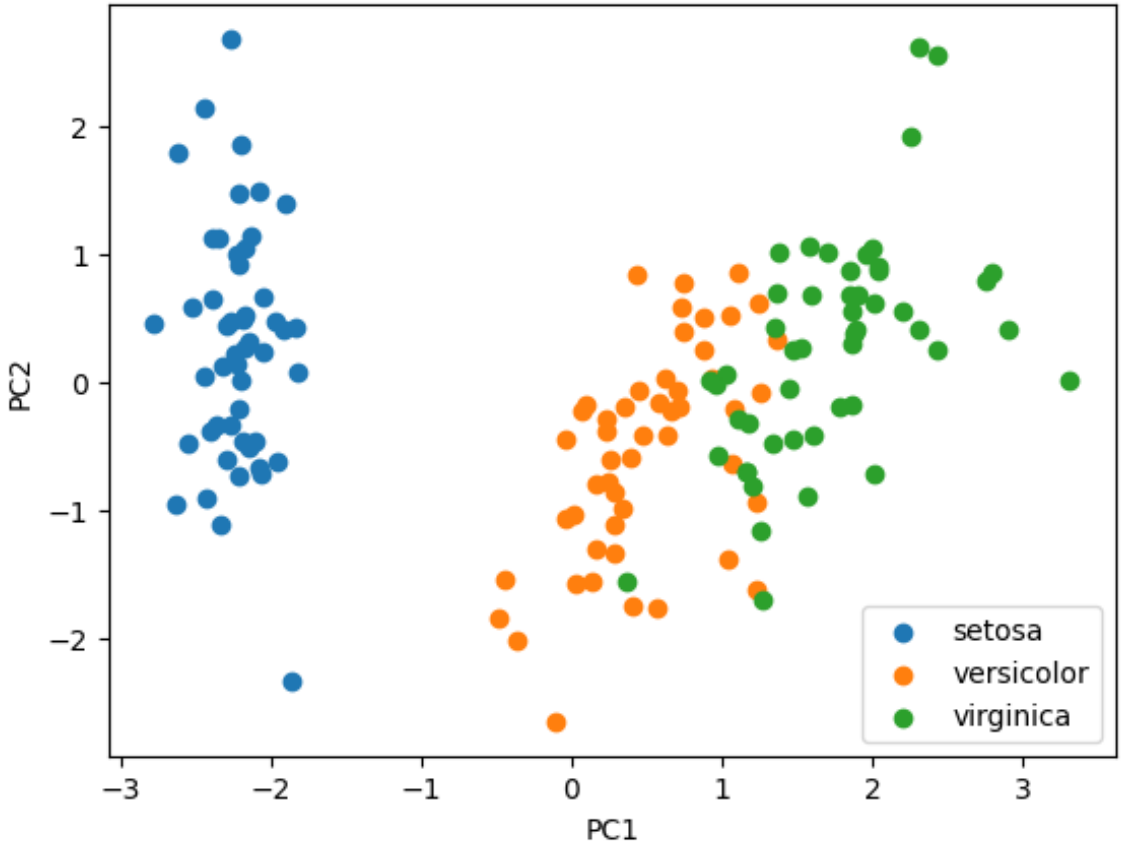
	A	B	C	D	E
1	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
2	5.1	3.5	1.4	0.2	setosa
3	4.9	3	1.4	0.2	setosa
4	4.7	3.2	1.3	0.2	setosa
5	4.6	3.1	1.5	0.2	setosa
6	5	3.6	1.4	0.2	setosa
7	5.4	3.9	1.7	0.4	setosa
8	4.6	3.4	1.4	0.3	setosa
9	5	3.4	1.5	0.2	setosa
10	4.4	2.9	1.4	0.2	setosa
11	4.9	3.1	1.5	0.1	setosa
12	5.4	3.7	1.5	0.2	setosa
13	4.8	3.4	1.6	0.2	setosa
14	4.8	3	1.4	0.1	setosa
15	4.3	3	1.1	0.1	setosa
16	5.8	4	1.2	0.2	setosa
17	5.7	4.4	1.5	0.4	setosa
18	5.4	3.9	1.3	0.4	setosa
19	5.1	3.5	1.4	0.3	setosa
20	5.7	3.8	1.7	0.3	setosa

Iris Dataset

```python
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA

# Load the Iris dataset
iris = load_iris()

# Standardize the data
scaler = StandardScaler()
X = scaler.fit_transform(iris.data)

# Apply PCA to reduce the dimensionality of the data to 2 dimensions
pca = PCA(n_components=2)
X_reduced = pca.fit_transform(X)

# Get unique target labels and corresponding names
target_labels = np.unique(iris.target)
target_names = iris.target_names[target_labels]

# Plot the reduced data with labeled points
for target_label, target_name in zip(target_labels, target_names):
    plt.scatter(X_reduced[iris.target == target_label, 0],
                X_reduced[iris.target == target_label, 1],
                label=target_name)

plt.xlabel('PC1')
plt.ylabel('PC2')
```

Iris Dataset Reduction Using PCA

- This procedure was carried out utilizing the SCIKIT learning environment and the necessary libraries.
- Importing the libraries,
  - Standardizing the feature values (essential for more accurate results),
  - Instantiating our PCA, and visualizing the results are the main steps.



Results of Evaluation

The plot demonstrates how the original high-dimensional data (4 features: sepal length, sepal width, petal length, and petal width) was translated into a lower-dimensional space (two primary components: PC1 and PC2).

The principal components are linear combinations of the original features that capture the most significant patterns and variations in the data.

The scatter plot depicts the projections of the original data onto the dataset's two most significant directions of variation, PC1, and PC2.

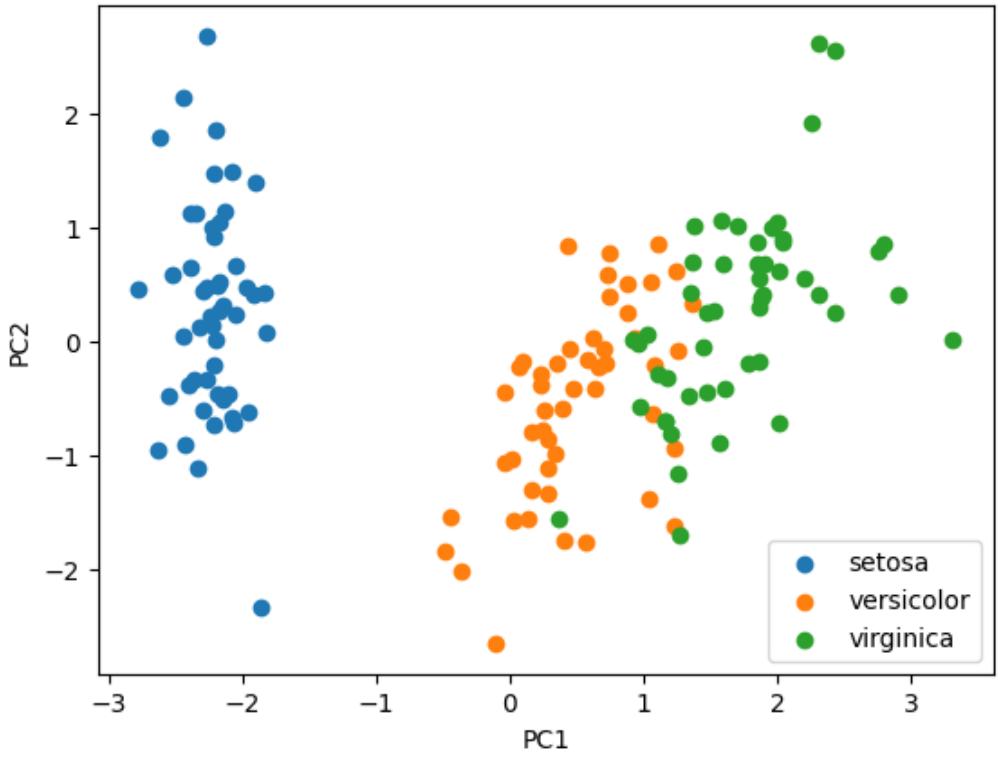
Results of Evaluation

My simulation’s explained variance ratio is [0.72962445, 0.22850762].

It is essential to note that the explained variance is a key factor for determining how much of the variance in data is retained after reduction.

The percentage of my PC1 and PC2 is approximately 95.81 percent, indicating that a significant amount of variance is still retained after reduction.

It is critical that the percentage of explained variance be substantial in order for data reduced to still be a precise representation of the dataset.



Conclusion

- Dimensionality reduction is a significant technique in machine learning with numerous applications. It alleviates the curse of dimensionality by eliminating redundant or noisy features, increasing computing efficiency, and reducing overfitting.

References

- [AW10] Abdi, Hervé; Williams, Lynne J: Principal component analysis. Wiley interdisciplinary reviews: computational statistics, 2(4):433–459, 2010.
- [Gé22] Géron, Aurélien: Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow. Ö'Reilly Media, Inc.", 2022.
- [Ha09] Hastie, Trevor; Tibshirani, Robert; Friedman, Jerome H; Friedman, Jerome H: The elements of statistical learning: data mining, inference, and prediction, volume 2. Springer, 2009.
- [RM17] Raschka, Sebastian; Mirjalili, Vahid: Python Machine Learning. Packt Publishing, 2nd edition, 2017.
- [Va09] Van Der Maaten, Laurens; Postma, Eric; Van den Herik, Jaap et al.: Dimensionality reduction: a comparative. J Mach Learn Res, 10(66-71):13, 2009.
- [LH08] LJPvd, Maaten; Hinton, GE: Visualizing high-dimensional data using t-SNE. J Mach Learn Res, 9(2579-2605):9, 2008.