Analysis and Classification of High-Dimensional Data:
A Comparison of Dimensionality Reduction, Clustering and Classification Methods

Fundamentals of Data Science Project

Szymon Łabędziewski

Table of contents

1. Data Summary

The dataset was successfully loaded and analysed to ensure it is suitable for further processing:

- Dataset Structure:
  - Number of samples: 10,000
  - Number of features: 784 (28x28 pixel images flattened)
  - Number of classes: 10 (e.g., T-shirt/top, Trouser, Pullover, Dress, etc.)
- Class Distribution:

The data is balanced, with each class containing approximately an equal number of samples. This ensures that subsequent machine learning models will not favour any particular class.

- Data Quality Check:

No missing values or anomalies were identified, simplifying the preprocessing stage. This confirms the dataset's readiness for dimensionality reduction and classification tasks.

- Visualization:

Sample images from each class were plotted to confirm data quality and verify class distinctions. The visualizations indicated clean and representative images for each class, supporting the clustering and classification steps described in later sections.

Analysis and Insights:

The dataset is well-structured and balanced, which is beneficial for machine learning tasks.

- The high dimensionality (784 features) necessitates dimensionality reduction to reduce computational complexity and highlight intrinsic patterns in the data.
- The balanced distribution across classes eliminates the need for techniques like oversampling or undersampling, ensuring unbiased results.

Comparison with ChatGPT Guidance:

ChatGPT's recommendations aligned with the approach taken in this step, emphasizing the importance of verifying data structure, class distribution, and quality through visualization. The guidance was instrumental in identifying these preliminary checks as critical for the project's success.

2. Dimensionality Reduction

There was applied three dimensionality reduction techniques to analyse the dataset:

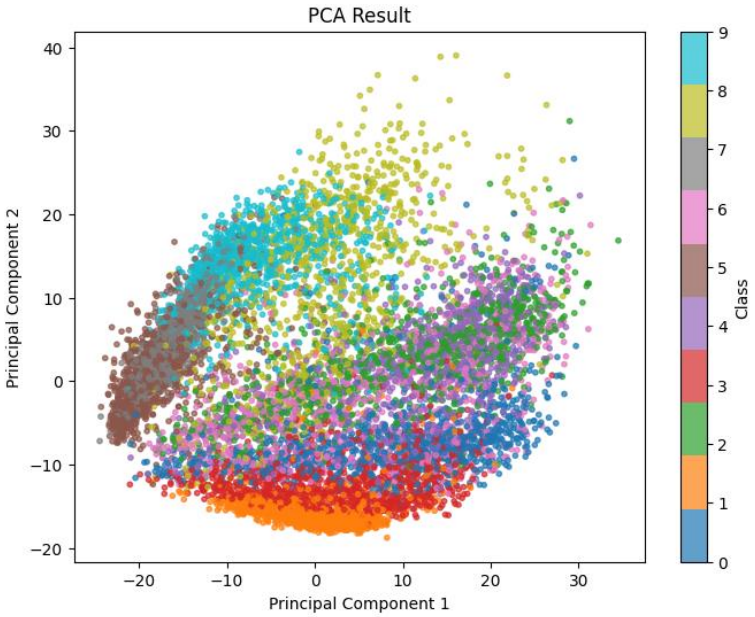
a) PCA (Principal Component Analysis):
   - Reduced the data to 2 dimensions.
   - Explained variance ratio for the two components was approximately 22% and 14%, capturing only ~36% of the data's variance.
   - PCA focuses on maximizing linear variance, which often limits its ability to separate classes with complex, non-linear relationships.

b) t-SNE:
   - Captured non-linear relationships in 2D space.
   - While computationally expensive, it preserved local structures effectively, forming distinct clusters in many cases.

c) UMAP:
   - Balanced computational efficiency with structure preservation.
   - Provided clear clustering in the reduced space, comparable to t-SNE, while being significantly faster.

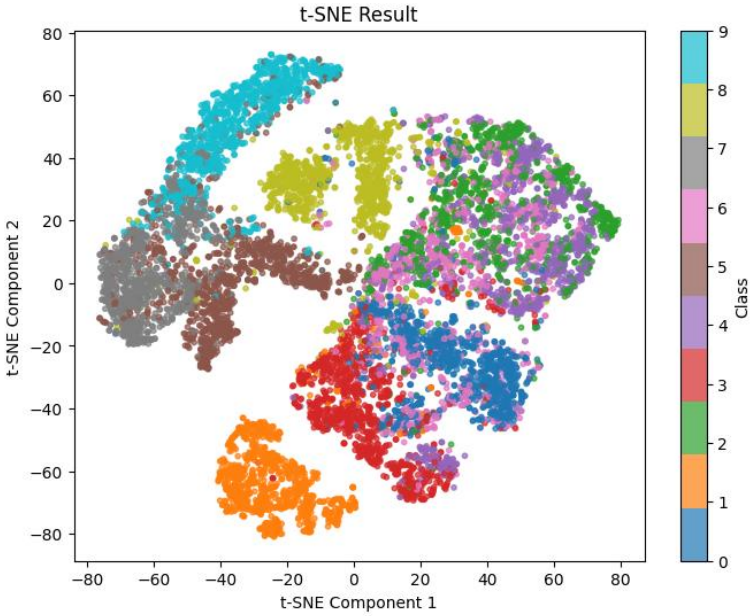
Analysis and Insights:

- PCA:



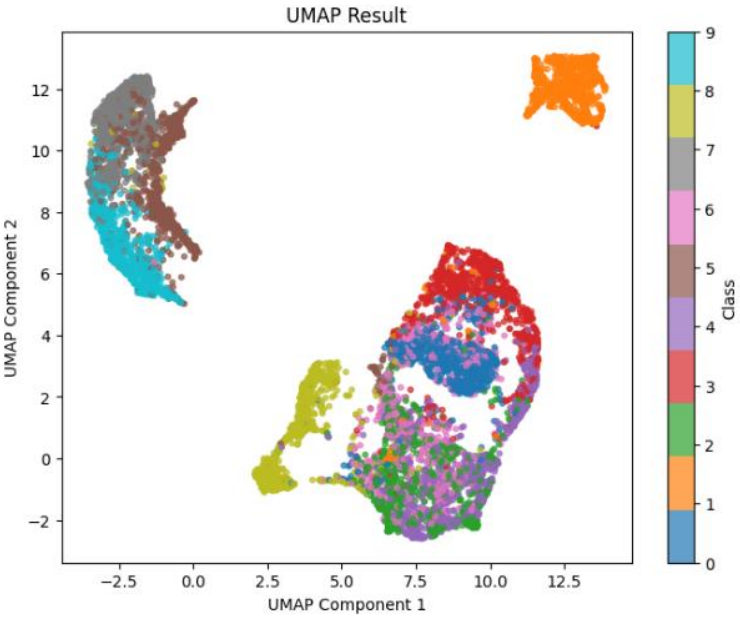
   - Visualization of PCA-reduced data revealed overlapping clusters (e.g., "shirts" and "coats"), underscoring its limitations for datasets with non-linear dependencies.
   - It showcases these overlaps, highlighting that PCA is better suited for analysing global variance than for clear class separations.

- t-SNE:



   - t-SNE provided better clustering than PCA, effectively preserving local neighbourhood structures. It highlighted distinct clusters for classes with strong differentiating features (e.g., "sandals" vs. "shirts").
   - It demonstrates tighter, more compact clusters compared to PCA, though mild overlaps persisted in some areas.

- UMAP:



  - UMAP achieved similar clustering quality to t-SNE, while maintaining a more uniform spread and preserving both local and some global structures.
  - Its efficiency makes it a practical choice for exploratory data analysis.
  - It illustrates distinct clusters with minimal overlaps, emphasizing its utility for high-dimensional datasets.

Comparison with ChatGPT Guidance:

The implemented approach aligns with the recommendations provided by ChatGPT, which suggested PCA, t-SNE, and UMAP as complementary methods for dimensionality reduction. Observations during implementation validated their utility:

- PCA was effective in identifying global variance but limited in separating complex classes.
- t-SNE and UMAP excelled in capturing non-linear structures and forming distinct clusters.

3. Visualization of Reduced Dataset

To better understand the relationships within the dataset, we applied three dimensionality reduction techniques: PCA (Principal Component Analysis), t-SNE, and UMAP. Each method was used to reduce the high-dimensional data to 2D, enabling us to visually assess clustering and class separability.

a) PCA Visualization:
   - PCA aimed to capture the global variance of the dataset by reducing it to two dimensions.
   - Although PCA was effective at revealing the global structure, it struggled with separating classes that have similar features (e.g., "shirts" vs. "coats").
   - The PCA visualization shows overlapping clusters, indicating the limitations of PCA in handling non-linear relationships between data points.

b) t-SNE Visualization:
   - t-SNE is particularly effective for capturing local relationships and preserving neighbourhood structure.
   - The resulting visualization showed better separation of the classes compared to PCA, but some mild overlaps were still present.
   - t-SNE is more focused on local clustering, which makes it well-suited for identifying distinct groups, though it may distort global relationships between clusters.

c) UMAP Visualization:
   - UMAP, like t-SNE, is designed to preserve local structure but is also more efficient in maintaining global relationships between clusters.
   - UMAP provided clear clusters with minimal overlaps and demonstrated a more uniform spread of the data.
   - UMAP is also computationally faster than t-SNE, making it particularly useful for handling larger datasets.

Analysis and Insights:

- Clustering Quality:
  - Both t-SNE and UMAP provided better clustering than PCA, showing clearer separation between classes.
  - PCA revealed some global patterns but was not as effective at separating complex classes that have similar features.
  - t-SNE and UMAP both excelled at capturing local structures and distinct clusters, with UMAP offering a more uniform spread and fewer overlaps than t-SNE.
- Dimensionality Reduction Objective:
  - PCA excels at identifying global variance but is limited in separating local class structures.
  - In contrast, t-SNE and UMAP focus on preserving local structures, which is more useful for identifying clusters within complex datasets.
  - While both t-SNE and UMAP performed well, UMAP emerged as the more efficient method for exploratory analysis in this project due to its faster computation and clearer cluster separation.

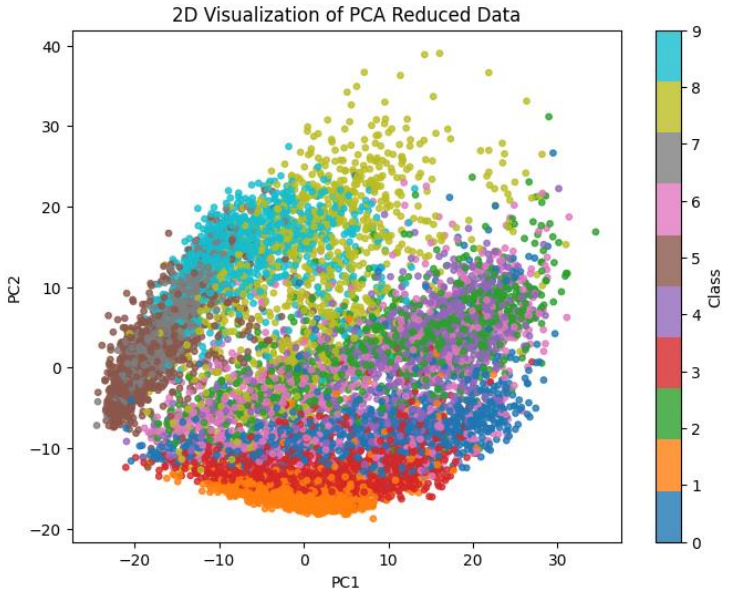
Comparison with ChatGPT Guidance:

- The applied dimensionality reduction techniques—PCA, t-SNE, and UMAP—aligned well with ChatGPT's guidance, which suggested these methods as suitable for this type of analysis.
- Practical implementation confirmed their utility:
    - PCA was useful for understanding global variance but showed limitations in separating classes with non-linear relationships.
    - t-SNE and UMAP excelled in capturing local neighbourhood structures and clustering, with UMAP providing a more balanced approach to preserving both local and global relationships.

Visualizations of Reduced Datasets:

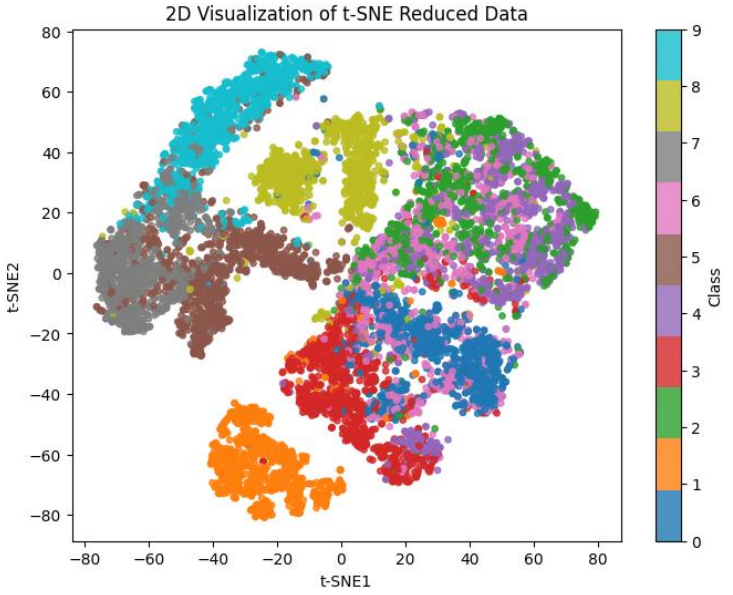
<u>PCA Visualization:</u>

- **Analysis**: PCA provided a reasonable separation of classes with distinct features (e.g., "sandals" vs. "shirts") but had difficulty with overlapping classes such as "shirts" and "coats", which share similar characteristics.
- The PCA visualization clearly shows these overlaps, highlighting the limitations of PCA in reducing high-dimensional data with complex, non-linear relationships.



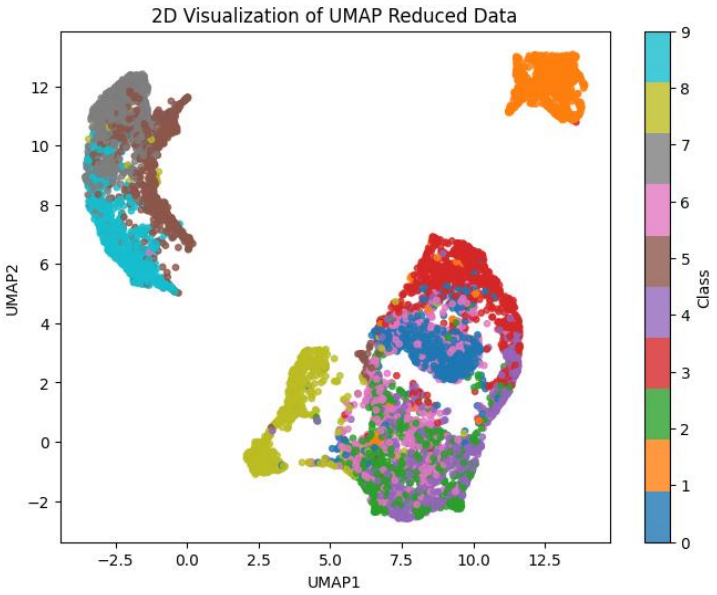
<u>t-SNE Visualization:</u>

- **Analysis**: t-SNE provided a clearer clustering than PCA, effectively grouping classes with similar features into distinct clusters. While some mild overlaps remained, t-SNE showed better separability, particularly for classes such as "sandals" and "coats".
- The t-SNE visualization reveals how well local structures are preserved, even though global relationships might be slightly distorted.



UMAP Visualization:

- **Analysis**: UMAP produced similar clustering quality to t-SNE but with fewer overlaps and a more uniform spread of clusters. It was better at preserving both local and global structures in the reduced space.
- The UMAP visualization demonstrates how the method captures both local relationships and the overall distribution of classes effectively, while maintaining computational efficiency.



Key Takeaways:

- PCA is useful for understanding global variance but has limitations when dealing with non-linear relationships between features, especially for similar classes.
- Both t-SNE and UMAP outperform PCA in terms of visualizing and separating clusters, with UMAP providing a more uniform spread and better computational efficiency.
- For large-scale datasets, UMAP's computational efficiency and the clarity of clustering make it the preferred method, especially for exploratory analysis.
- t-SNE excels in preserving local structures but may struggle with scalability on large datasets. UMAP provides a more balanced approach, preserving both local and global structures while being more computationally efficient.

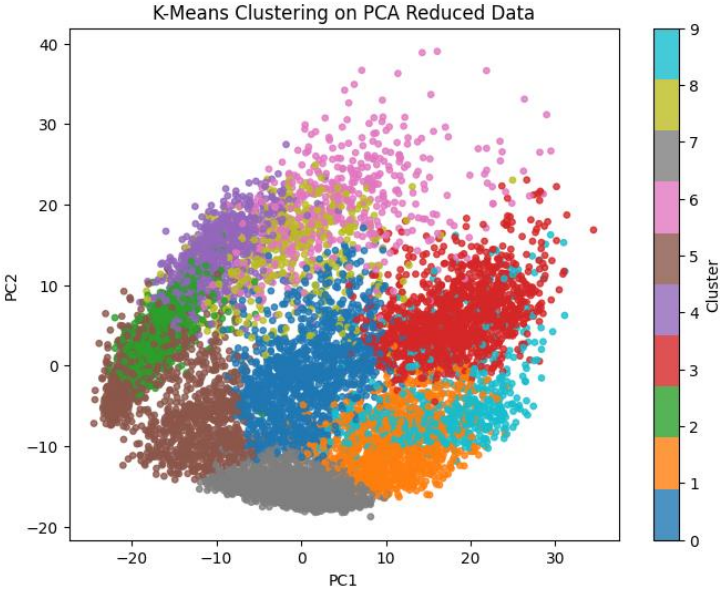
4. Clustering Data

K-Means Results:

- Adjusted Rand Index (ARI): 0.3387
- The ARI of 0.3387 indicates **moderate alignment** between the predicted clusters and the true labels. However, it also highlights the challenges in effectively separating certain classes due to **overlapping classes** and **nonlinear structures** in the data.
- Confusion Matrix:

$$\begin{bmatrix} 160 & 154 & 2 & 14 & 0 & 131 & 14 & 18 & 0 & 507 \\ 20 & 60 & 0 & 8 & 0 & 15 & 1 & 895 & 0 & 1 \\ 295 & 35 & 0 & 523 & 0 & 107 & 15 & 1 & 0 & 24 \\ 50 & 399 & 0 & 7 & 0 & 104 & 0 & 434 & 0 & 6 \\ 157 & 211 & 0 & 560 & 0 & 54 & 4 & 14 & 0 & 0 \\ 7 & 1 & 232 & 0 & 72 & 635 & 1 & 0 & 52 & 0 \\ 298 & 121 & 0 & 251 & 1 & 194 & 18 & 8 & 0 & 109 \\ 0 & 0 & 771 & 0 & 142 & 85 & 0 & 0 & 2 & 0 \\ 304 & 8 & 50 & 30 & 24 & 79 & 497 & 1 & 4 & 3 \\ 6 & 0 & 42 & 0 & 582 & 31 & 0 & 0 & 339 & 0 \end{bmatrix}$$

- Analysis:
    - The **K-Means clustering** algorithm struggles to separate certain classes, as shown by the confusion matrix. For example, **class 0** is spread across multiple clusters, indicating difficulty in distinguishing between similar classes.
    - **Cluster 7** aligns well with **class 1**, but many other classes show significant **misclassifications**.
    - The method assumes spherical clusters and linear separability, which are not ideal for this dataset that has overlapping and non-linear relationships.
    - Visualization of K-Means Clustering on PCA Reduced Data:



- The PCA reduction reveals that the **clusters formed by K-Means overlap**, especially for classes with similar characteristics (e.g. **"shirts" vs. "coats"**). This reinforces the challenge that K-Means faces in this project.
- It clearly shows overlapping clusters, indicating that the assumption of spherical clusters does not align well with the data's true structure.

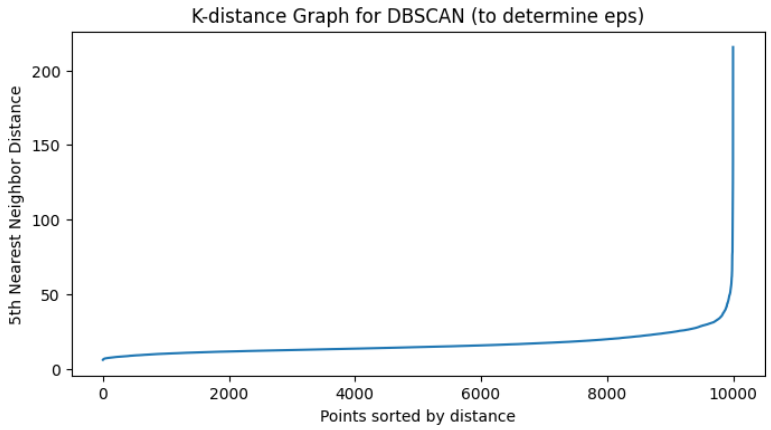
DBSCAN Results:

- Adjusted Rand Index (ARI): 0.03096
- The ARI for DBSCAN is very low (0.03096), suggesting that DBSCAN struggled significantly to cluster the data correctly.
- Confusion Matrix:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 823 & 0 & 177 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 247 & 748 & 2 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 901 & 0 & 62 & 0 & 0 & 22 & 0 & 1 & 1 & 11 & 0 & 0 & 2 & 0 \\ 829 & 0 & 20 & 136 & 0 & 0 & 0 & 0 & 0 & 0 & 15 & 0 & 0 & 0 \\ 942 & 0 & 35 & 0 & 0 & 6 & 0 & 3 & 10 & 2 & 0 & 0 & 2 & 0 \\ 953 & 0 & 0 & 0 & 3 & 0 & 30 & 0 & 0 & 0 & 0 & 0 & 0 & 14 \\ 764 & 0 & 200 & 1 & 0 & 0 & 0 & 17 & 0 & 0 & 0 & 0 & 18 & 0 \\ 852 & 0 & 2 & 0 & 122 & 0 & 0 & 0 & 0 & 0 & 0 & 24 & 0 & 0 \\ 999 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1000 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

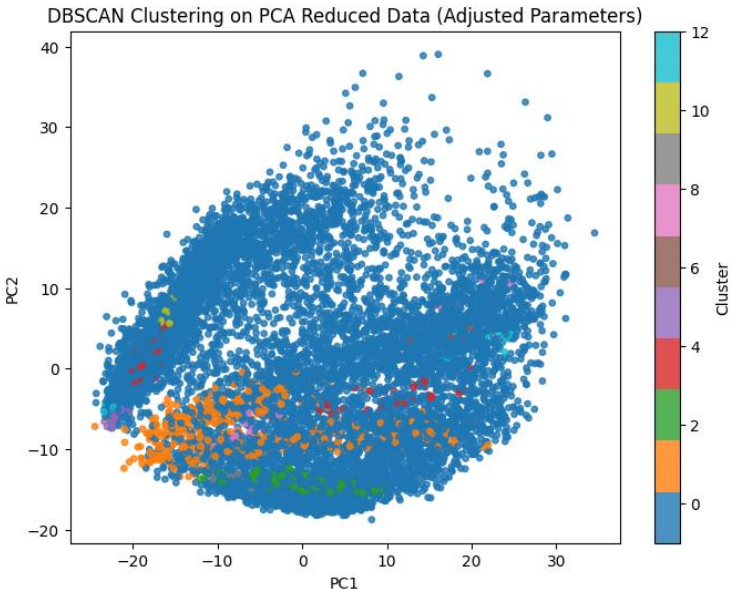
- Analysis:
  - **DBSCAN** had difficulty clustering the data due to its sensitivity to **density**. The **confusion matrix** reveals that **many points** are considered **noise** (represented by zero values in the matrix), indicating that DBSCAN is struggling to form proper clusters.
  - DBSCAN works well with datasets that have **irregular density distributions**, but this dataset has a mix of dense and sparse regions, leading to **incorrect cluster assignments**.

  - K-Distance Graph for DBSCAN (to determine eps):



    - It illustrates how the optimal eps parameter can be determined using the k-distance graph.
    - This graph is crucial for selecting the eps value, which controls the neighbourhood size for clustering. By choosing the correct eps, the density-based clustering can be improved, reducing the number of outliers and improving cluster separation.

  - DBSCAN Clustering on PCA-Reduced Data (Adjusted Parameters):



    - It illustrates how clusters are distributed in the reduced 2D space.
    - **Cluster Formation**: A few well-defined clusters are visible, but there is a significant number of points categorized as noise (8310), reflecting the challenges of cluster density separation.
    - **Outliers**: The high number of outliers indicates **sparsity** in parts of the dataset and the difficulty in separating classes with varying densities.
    - **Adjusted Parameters**: After tuning the parameters, there was **better distribution** of clusters, but a **high outlier count remains**, showing that DBSCAN is sensitive to **parameter settings**.

DBSCAN effectively identifies **local groupings**, but its performance is hindered by the dataset's complexity, especially after dimensionality reduction.

Parameter adjustments are critical for improving clustering performance, but **DBSCAN still struggles** with high-dimensional data, even after adjusting the parameters.

Key Takeaways:

- **K-Means** and **DBSCAN** both struggle with this dataset due to assumptions about cluster shapes (spherical in K-Means and density-based in DBSCAN).
- For non-linear and overlapping datasets, methods like GMM, Spectral Clustering, or Agglomerative Clustering may provide more accurate results.
- **DBSCAN's performance** heavily depends on the **eps** parameter, and using the **k-distance graph** can help optimize this choice for better clustering performance.

5. Classification Results on Train-Test Split

Key Metrics Across Models:

1. Random Forest:
    - Accuracy: 88.28%
    - Precision: 88.15%
    - Recall: 88.28%
    - F1-Score: 88.12%
    - **Observation**: Random Forest shows strong performance with balanced precision and recall, making it reliable for tasks requiring consistent predictions. Additionally, its interpretability allows for the analysis of feature importance, making it useful for understanding model behaviour.
2. k-Nearest Neighbours (k-NN):
    - Accuracy: 86.11%
    - Precision: 86.34%
    - Recall: 86.11%
    - F1-Score: 86.08%
    - **Observation**: k-NN performs slightly worse than Random Forest, indicating its limitations with potentially overlapping data distributions. Despite its simplicity, it may struggle with high-dimensional data due to its reliance on distance-based metrics.
3. Support Vector Machine (SVM):
    - Accuracy: 89.43%
    - Precision: 89.35%
    - Recall: 89.43%
    - F1-Score: 89.36%
    - **Observation**: SVM outperforms both Random Forest and k-NN in all metrics, showcasing its robustness in handling high-dimensional data and complex decision boundaries, making it ideal for non-linear classification tasks.

Comparison of Classifiers:

- Best Model: SVM stands out as the top performer in terms of all evaluated metrics, suggesting its suitability for this classification problem.
- Trade-offs: While Random Forest is slightly less accurate, it may be preferred in scenarios requiring better interpretability or faster predictions on large datasets.

Conclusion:
The classification models achieved high accuracy overall, with SVM emerging as the most effective choice. These results underscore the value of experimenting with diverse classifiers to achieve optimal outcomes in terms of accuracy, precision, recall, and F1-score.

Model Comparison Table:

Model	Accuracy	Precision	Recall	F1-Score
Random Forest	88.28%	88.15%	88.28%	88.12%
k-Nearest Neighbours	86.11%	86.34%	86.11%	86.08%
Support Vector Machine	89.43%	89.35%	89.43%	89.36%

6. Conclusions

This project explored dimensionality reduction, clustering, and classification methods to analyse and model the provided dataset. Key findings include:

- Dimensionality Reduction:
  - **PCA** effectively captured **linear variance**, but methods like **t-SNE** and **UMAP** demonstrated better clustering of data points in lower dimensions, particularly for **non-linear patterns**.
  - While **PCA** was useful for capturing global variance, **t-SNE** and **UMAP** were better suited for preserving local structures and effectively separating classes with non-linear relationships.
- Clustering:
  - Both **DBSCAN** and **K-Means** faced challenges with this dataset due to overlapping features and **irregular density**. Future work could explore alternative clustering methods, such as **Gaussian Mixture Models (GMM)**, which can model clusters as ellipses and handle overlapping data more effectively.
- Classification:
  - **SVM** achieved the highest **accuracy (0.8943)** among the classification models, demonstrating its suitability for **high-dimensional data** with complex decision boundaries.
  - **Random Forest**, while slightly less accurate (**0.8828**), offers greater **interpretability** and is useful for **feature analysis**.
  - The **k-Nearest Neighbours (k-NN)** model performed well as a **baseline**, achieving an accuracy of **0.8611**, but its performance drops for datasets with high dimensionality or overlapping classes.

7. References:

- **Scikit-learn Documentation**

https://scikit-learn.org/stable/documentation.html

- **UMAP Documentation**

https://umap-learn.readthedocs.io/en/latest/

- **Scientific Articles for t-SNE**

van der Maaten, L., & Hinton, G. (2008). Visualizing Data using t-SNE. Journal of Machine Learning Research, 9(Nov), 2579-2605.

https://www.jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf

- **Python Library Documentation**

Matplotlib and Seaborn for visualization:

https://matplotlib.org/stable/contents.html
https://seaborn.pydata.org/