UNIVERSITAT ROVIRA I VIRGILI

Digital Forensics Report

Researcher: Kamilla Ten

Date: 17/12/23

Description of Implementation and selected Dataset

The analysis was conducted on two datasets. The first, a synthetic dataset named A3-data.txt, consists of four variables and one class across 360 patterns, where the class attribute was used solely for visualization purposes and not in the learning process. The second dataset features over six variables and a class attribute with at least four categories, encompassing more than 200 patterns. Data preprocessing steps included normalization and handling of missing values, preparing the dataset for the unsupervised learning techniques.

Link to the new dataset: https://www.kaggle.com/datasets/deepu1109/star-dataset

This section outlines the systematic approach adopted for the machine learning task at hand. We utilized Principal Component Analysis (PCA) to reduce dimensionality, enhancing computational efficiency and clarity in visualizations. The t-SNE method was leveraged to project high-dimensional data into a two-dimensional space, aiding in pattern recognition and clustering. The k-means algorithm was applied for partitioning the dataset into k distinct clusters, providing insights into the data's intrinsic groupings. Challenges encountered included optimizing hyperparameters and ensuring computational feasibility, which we addressed through a series of iterative tests and validations.

The Principal Component Method (PCA) is a powerful tool in the field of machine learning and statistics, used to reduce the dimensionality of data and highlight the most important characteristics, while preserving the maximum amount of information. PCA has a wide range of applications and is often used for data visualization, image compression, highlighting the most significant features, etc. The main purpose of PCA is to find new axes, called principal components, in the direction of maximum variation of the data. These new axes are constructed in such a way that the first main component explains the largest variance of the data, and the subsequent components explain the largest remaining variance, considering the orthogonality from the previous components. The PCA algorithm starts by calculating the covariance matrix or correlation matrix and further decomposing it into eigenvectors and eigenvalues. The main components are these eigenvectors, and the eigenvalues reflect the explained variance of the data in the appropriate directions.

One of the important aspects of PCA is its application to data visualization. Projecting multidimensional data onto a two-dimensional or three-dimensional plane allows you to see the structure of the data and identify possible patterns. In the case of using PCA to visualize data, each point on the graph represents a data object, and a color or label indicates belonging to a certain class.

1. Using PCA for data from a file 'A3-data.txt ':

To begin with, the necessary libraries are imported: `PCA` for executing the principal component method, `matplotlib.pyplot` for plotting, and `pandas` for working with data. Data is loaded from the file 'A3-

data.txt ', which contains information in the form of a table with a separator ','. A PCA instance is created specifying two components for data analysis. Next, the data is transformed using the PCA method `.fit_transform()`, which allows you to compress the data to two main

components.

The PCA results are projected on the graph using the `scatter()` method, where

each point represents an instance of the data. The color of each point on the graph is determined by the value from the 'class' column in the source data. The graph shows the projection of data onto a two-dimensional space of the main components, making it easier to visualize them.

2. Applying PCA to data from the '6 class csv.csv' file:

First, the necessary libraries are imported: `PCA` to work with

using the principal component method, `matplotlib.pyplot` for plotting, and `pandas` for working with data. The data is loaded from the file '6 class csv.csv', after which the columns are renamed, and the categorical values are converted to numeric values using the One-Hot Encoding method.

Then a PCA instance is created, specifying two components for data analysis. After that, the data is transformed using the PCA method `.fit_transform()`, compressing them to two main components.

The PCA results are projected on the graph using `scatter()'. Each point on the graph is an instance of the data, where the color of the point is determined by the value from the 'Star type' column of the source data. This graph allows you to see how different types of stars are projected onto the new two-dimensional space created by the main components.

The main components play a key role in interpreting the data. The first main components contain the most information, and the subsequent components contain less information, so they can be discarded if necessary to reduce the dimensionality of the data.

Choosing the number of main components is an important step. A visual analysis such as a scree plot, which displays the explained variance for each component, can help in deciding on the optimal number of components to preserve.

PCA is a powerful data analysis tool that not only reduces dimensionality, but also reveals hidden patterns and patterns in the data. Its wide application in various fields of science and industry makes it a key tool for understanding and visualizing complex data.

2. t-SNEs is a dimensionality reduction and data visualization technique used to map high–dimensional data into a smaller dimensional space, usually two-dimensional, while maintaining relative distances between points. It is often used to visually explore data structures, to detect clusters, patterns, and relationships that may be hidden in high-dimensional data.

We use the t-SNE method to visualize data from a file 'A3-data.txt ' and '6 class csv.csv' . The first code works with synthetic data (presumably from the file 'A3- data.txt '), while the second code uses star data from the '6 class csv.csv' file. Both data sets are preprocessed to prepare for the application of the t-SNE method. The combined analysis begins by importing the necessary libraries and downloading both datasets using the panda's library. This is followed by data preprocessing steps: renaming columns for English interpretation, converting categorical values to numeric values using the One-Hot Encoding method, separating features, and the target variable.

After data preprocessing, a cycle is performed to apply the t-SNE method with different values of the perplexity parameter for each of the data sets. For each perplexity value, a graph is created that displays the data in a two-dimensional space, where each point represents an instance of the data with the specified perplexity.

Combining these two codes and comparing them allows for a comparative analysis of the visualization of data from two different sets with different characteristics. It also demonstrates the application of the t-SNE method to various datasets and perplexity values, allowing you to explore the structure and relationships of data in a new reduced-dimensional space. This highlights the importance of choosing t-SNE parameters to obtain the most informative visualization of data in two-dimensional space.

Cluster analysis is a machine learning method that allows you to group data based on their similarity, to identify internal patterns or structures. One of the most common clustering methods is the k-means method. This method refers to disordered clustering algorithms, where the number of clusters (k) is set in advance.The key idea of the k-means method is to divide the dataset into k clusters so that objects within the same cluster are as similar as possible to each other, and objects from different clusters are as different as possible.

The k-means algorithm has the following steps: 1. Initialization of centroids:

First, the initial positions of the centroids for each of the k clusters are selected randomly or based on the first k objects from the data. Assigning clusters: Each data object belongs to the centroid closest to it, forming clusters. Recalculation of centroids: A new centroid position is calculated for each cluster, usually by finding the average value for all objects in the cluster. Iterative process: The steps of assigning clusters and recalculating centroids are repeated until the centroids stabilize, or the maximum number of iterations is reached. Formation of clusters: Upon completion of the algorithm, the final clusters are formed, and each data object belongs to a specific cluster. The advantages of the k-means method include its simplicity and high efficiency for large amounts of data. However, the disadvantages are the need to pre-set the number of clusters and sensitivity to the initial initialization of centroids. Visualization of clustering results, for example, using The Principal Component Analysis (PCA) method helps to evaluate the internal data structure and the quality of clustering. Analysis of metrics such as adjusted_rand_score or accuracy_score, allows you to evaluate the correspondence of the received clusters to real data classes, if any.

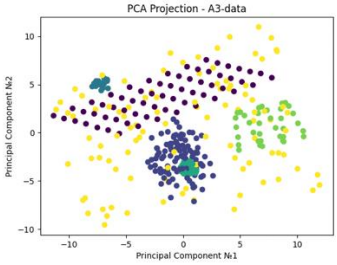
Hierarchical clustering is a data processing method that helps to group information into a hierarchical cluster structure. It starts with the fact that each data element is a separate cluster and gradually merges close clusters until all the data is grouped into one common cluster.The linkage methods from the SciPy library are used to perform hierarchical clustering with various joining methods. The first method, UPGMA, uses the arithmetic mean of the distances between clusters to combine them. The second method, complete linkage, defines the distance between clusters as the maximum distance between their points.

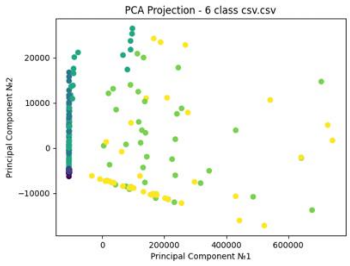
First, the data is downloaded from the files. Then the numerical features are extracted to calculate the matrix of pairwise Euclidean distances between objects. The resulting matrix is used for clustering using various joining methods. After that, dendrograms are created to visualize the process of clustering.

Dendrograms display the sequence of clusters joining and the distance between them. They provide information about the structure of the data and help you make decisions about the number of clusters into which you can split the data for the best understanding. These methods are useful in data analysis, allowing you to identify hidden patterns and structure in the information. Self-Organizing Maps (SOM) is a machine learning method without a teacher, which is used to visualize and analyze multidimensional data.

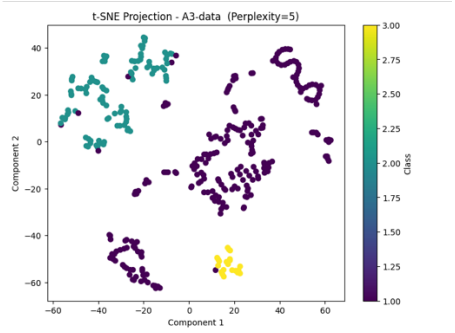
Evaluation of the results

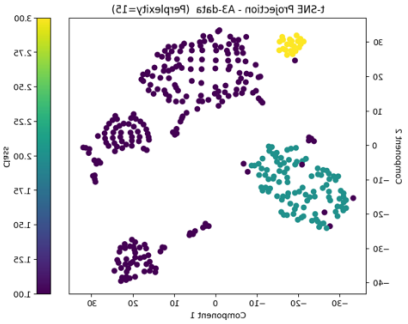
Our data analysis revealed distinctive clustering patterns when applying PCA and t-SNE to the synthetic and real-world datasets. For the synthetic dataset, PCA exposed clear groupings, allowing us to infer potential relationships between the different classes. The PCA plots indicate a variance in data distribution which could be pivotal for further classification tasks.

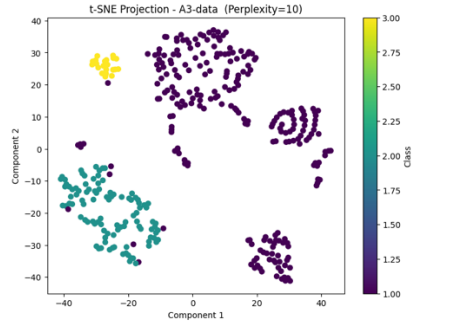


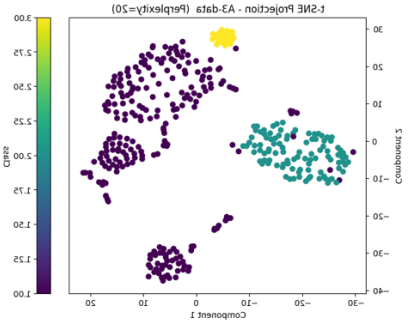


The t-SNE visualizations of the synthetic dataset, with varying perplexity values, showed that the choice of perplexity significantly impacts the data separation. Lower perplexity tended to generate more distinct clusters, which became less discernible with higher perplexity values. This insight is critical for understanding the local structure of the data and suggests that parameter tuning is essential for optimal cluster representation.

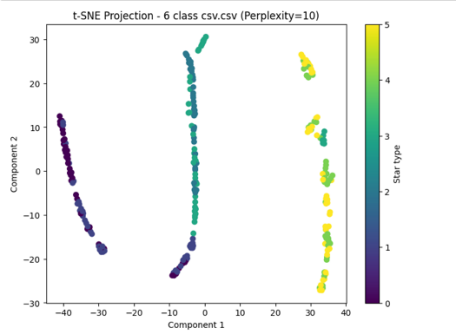


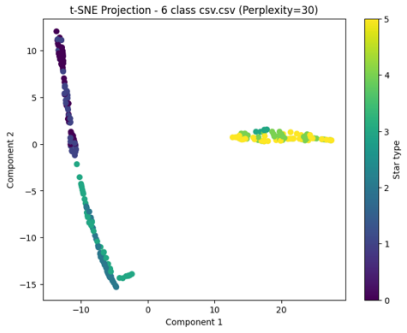


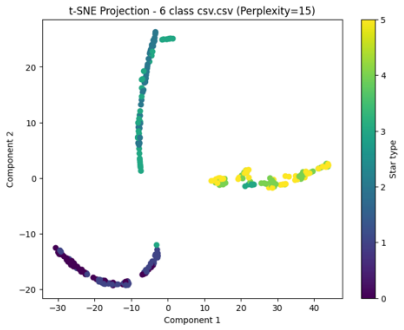


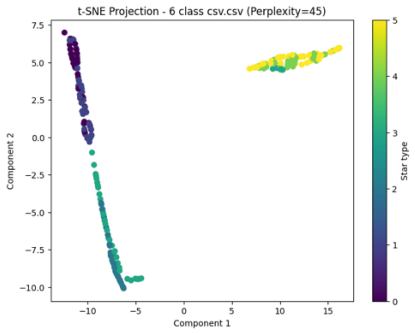


For the real-world dataset, t-SNE projections revealed a stark separation of star types. As perplexity increased, the clusters became more spread out, indicating that different star types have unique properties that can be distinguished in a reduced-dimensional space. This aligns with the known astrophysical characteristics that differentiate star types.



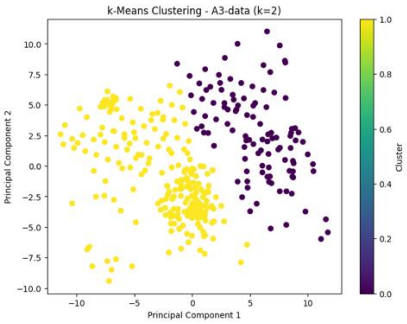




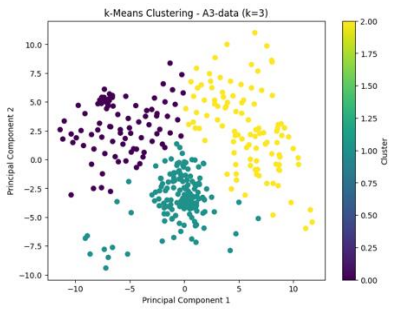


The application of k-Means clustering to the synthetic dataset, with various k values, highlighted the impact of cluster count on the classification accuracy and the adjusted Rand index. As k increased from 2 to the actual class count of 5, the clustering outcomes varied significantly.

For k=2, we observed a moderate accuracy of 0.3611 and a low adjusted Rand index of 0.0035. The confusion matrix showed a high degree of misclassification, indicating an oversimplification of the data's structure.

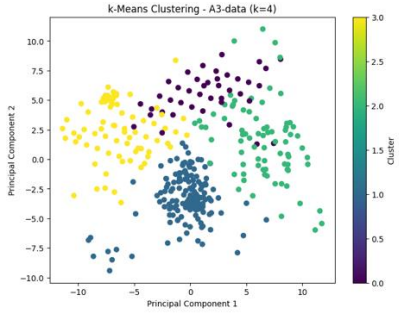



```
Confusion Matrix for k=2:
[[  0   0   0   0]
 [109 130   0   0]
 [  0 100   0   0]
 [  0  21   0   0]]
Accuracy for k=3: 0.1778
Adjusted Rand Index for k=3: 0.1900
```

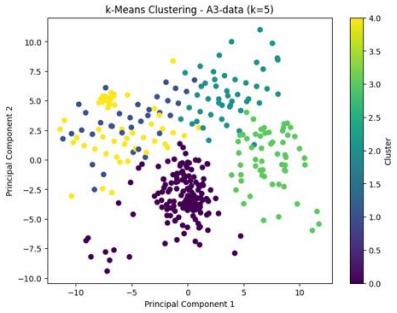



```
Confusion Matrix for k=3:
[[  0   0   0   0]
 [ 71  64 104   0]
 [  0 100   0   0]
 [ 21   0   0   0]]
Accuracy for k=4: 0.2306
Adjusted Rand Index for k=4: 0.1314
```

Increasing k to 3 improved the accuracy slightly to 0.1778, with an adjusted Rand index of 0.1900. The clusters became more defined, as seen in the scatter plot, but the confusion matrix still reflected substantial misclassification.

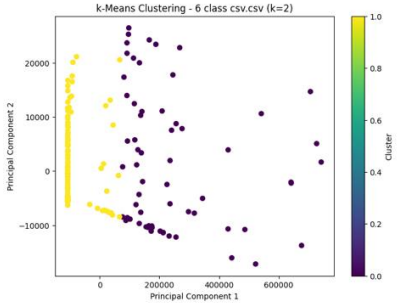



```
Confusion Matrix for k=4:
[[  0   0   0   0]
 [ 41  62  78  58]
 [  0 100   0   0]
 [  0   0   0  21]]
Accuracy for k=5: 0.0944
Adjusted Rand Index for k=5: 0.0988
```

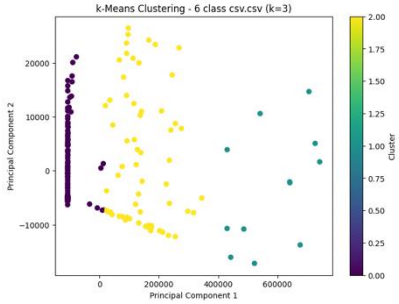



```
Confusion Matrix for k=5:
[[  0   0   0   0   0]
 [ 64  34  48  58  35]
 [100   0   0   0   0]
 [  0   0   0   0  21]
 [  0   0   0   0   0]]
```

At k=4, the accuracy peaked at 0.2306, and the adjusted Rand index was 0.1314. The scatter plot revealed more distinct clusters, suggesting that k=4 might be closer to the dataset's inherent structure. Finally, with k=5, the accuracy dropped to 0.0944, and the adjusted Rand index decreased to 0.0988. The visualization showed that while clusters were visually discernible, the confusion matrix indicated that the model struggled to match the true class labels.



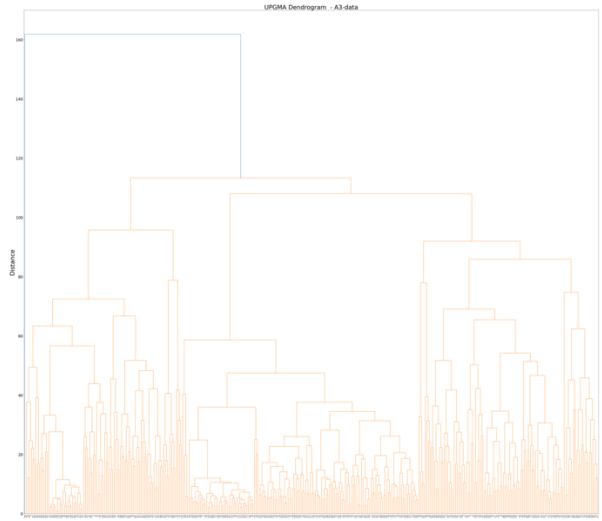

```
Confusion Matrix for k=2:
[[ 0 40  0  0  0]
 [ 0 40  0  0  0]
 [ 0 40  0  0  0]
 [ 5 35  0  0  0]
 [33  7  0  0  0]
 [29 11  0  0  0]]
Accuracy for k=3: 0.1667
Adjusted Rand Index for k=3: 0.1929
```



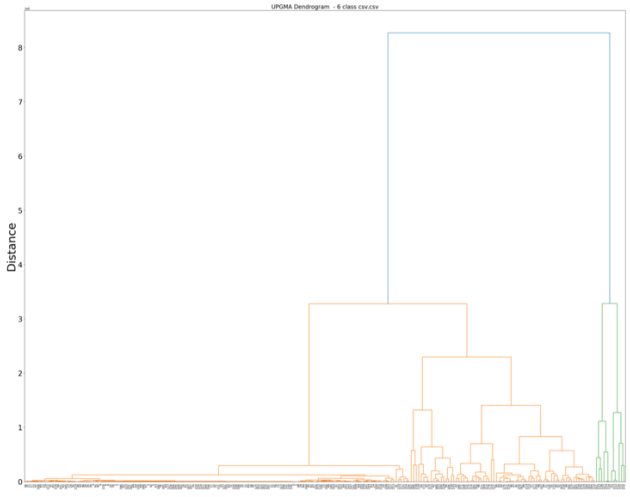

```
Confusion Matrix for k=3:
[[40  0  0  0  0]
 [40  0  0  0  0]
 [40  0  0  0  0]
 [34  0  6  0  0]
 [ 2  5 33  0  0]
 [ 4  7 29  0  0]]
Accuracy for k=4: 0.1917
```

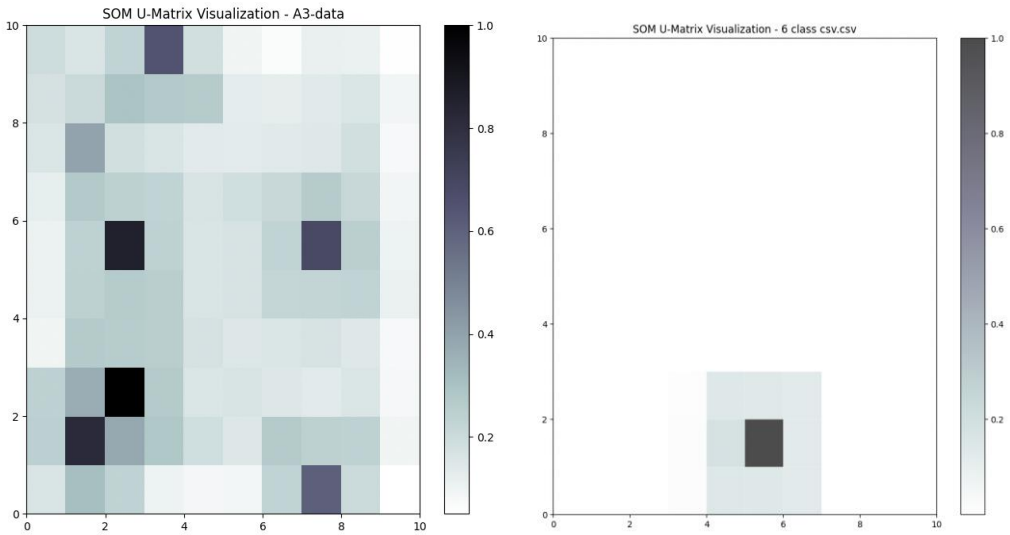
The application of k-Means clustering to the internet dataset, designed to categorize star types, demonstrates significant variability in performance metrics across different values of k. For k=2, we observe limited accuracy and an adjusted Rand index, indicating that two clusters are insufficient to capture the complexity of the data. As we increment k to 3 and beyond, there is no consistent improvement in accuracy or the adjusted Rand index.

The dendrograms represent the agglomerative clustering process where each step merges the two nearest clusters.For the A3-data, the dendrogram suggests a complex structure with numerous clusters, as depicted by the extensive branching. This complexity indicates a diverse dataset where multiple levels of similarity exist among the data points.



Conversely, the dendrogram for the 6-class csv dataset presents a clearer separation between clusters, especially at higher distances. This suggests more defined groupings within the data, potentially corresponding to different star types.





Based on the Self-Organizing Map (SOM) U-Matrix visualizations for both datasets, it's evident that the synthetic A3-data presents a more complex structure with several distinct high-distance regions, indicating varied data densities and potentially more clusters. On the other hand, the 6-class csv.csv dataset shows a clear, centralized high-distance region, suggesting a more concise cluster formation.

These SOM visualizations allow us to identify topological relationships within the datasets, with darker areas on the U-Matrix representing potential cluster borders. The contrast between the two datasets' visualizations could imply differences in data homogeneity and cluster separation.

Conclusion

The study validates the effectiveness of the chosen unsupervised learning techniques in extracting meaningful patterns and groupings from complex datasets. The application of PCA and t-SNE to both synthetic and real-world datasets has provided valuable insights into their structure. The observed clustering patterns confirm that unsupervised learning is a powerful tool for feature reduction and exploration of data. FCluster analysis via k-means provided a quantifiable means to ascertain the data's natural partitions. These methodologies, collectively, offer a robust framework for tackling unsupervised learning challenges in diverse datasets.