Exploring clinical heterogeneous data using unsupervised machine learning

Gabriel WARDE

Supervised by Dr. Adán JOSÉ-GARCÍA

In collaboration with Pr. DHAENENS and Pr. SOBANSKI

at the CRIStAL & INFINITE Laboratories

M1 DATA SCIENCE

August 26, 2025

Contents

Abstract. This research project focuses on exploring clinical heterogeneous data using unsupervised machine learning techniques. Clustering was performed on different types of datasets, including numeric, categorical, and mixed-typed heterogeneous clinical datasets. Algorithms such as K-means, K-modes, and K-prototypes were employed for clustering. The advantages, limitations, and applications of each of these clustering algorithms. The results demonstrated good performance, indicating the effectiveness of the clustering algorithms. Additionally, scatter plots and box plots were generated to provide visual insights and aid in better understanding the clustered data. Interested in exploring or learning some of the work conducted during my research? Feel free to visit the GitHub repository dedicated to this project: https://github.com/GabrielWarde/ResearchProject1.

1 Introduction

In recent years, the integration of unsupervised machine learning techniques has reshaped the landscape of medicine, introducing a new era of data-driven healthcare. One of the pivotal aspects of this transformation lies in the utilization of clustering algorithms, which play a crucial role in uncovering hidden patterns and structures within complex medical datasets. This research project explores the deep impact of unsupervised machine learning, particularly clustering algorithms, on modernizing the medical field.

Unsupervised machine learning techniques, unlike their supervised counterparts, operate without labeled data, making them particularly well-suited for analyzing vast and heterogeneous datasets that are commonly encountered in medicine. Within this domain, clustering algorithms stand out as powerful tools capable of grouping similar data points together based on inherent patterns, thus enabling healthcare professionals to gain deeper insights into diseases, treatments, and patient outcomes.

Moreover, the real-world nature of medical data presents huge challenges, often characterized by the presence of diverse datasets with varying types of information. Clinical datasets, for instance, may include numerical measurements, categorical variables, textual descriptions, and even imaging data. Dealing with such heterogeneous datasets requires sophisticated techniques capable of handling different data types smoothly. Unsupervised machine learning, particularly clustering algorithms, offers a flexible solution by accommodating various types of data and extracting meaningful insights from them, thus enhancing our understanding of diseases and treatment modalities.

Unsupervised machine learning techniques, especially clustering algorithms, play a pivotal role in deciphering complex relationships within these diverse datasets to identify distinct patient subgroups with unique disease manifestations, treatment responses, and prognoses. By uncovering these nuanced patterns, personalized medicine can offer tailored interventions that maximize efficacy while minimizing adverse effects, leading to a new era of patient-centered care.

Throughout this report, We will explore the functionality of the most popular clustering algorithms, such as the K-means, K-modes, and K-prototypes, their mechanisms, strengths, and limitations. Each of these algorithms offers unique strengths and applicability in uncovering patterns within diverse datasets encountered in the medical domain. By illuminating the intersection of unsupervised machine learning and medicine, this research project aims to underscore the pivotal role of data-driven approaches in driving innovation and improving patient outcomes in the 21st century.

2 Clustering Algorithms

According to [Jai10], the goal of data clustering, also known as cluster analysis, is to discover the natural grouping(s) of a set of patterns, points, or objects. Webster (Merriam-Webster Online Dictionary, 2008) defines cluster analysis as "a statistical classification technique for discovering whether the individuals of a population fall into different groups by making quantitative comparisons of multiple characteristics." The reader is referred to some recent surveys in the specialized literature, such as [ESA+]. The following summarizes the most relevant clustering algorithms used in this research project.

2.1 K-means

2.1.1 Definition

K-means clustering is a popular unsupervised machine learning algorithm used for clustering data points into groups or clusters based on their similarity. The term "k-means" was first used by James MacQueen in 1967. The standard algorithm was first proposed by Stuart Lloyd of Bell Labs in 1957 as a technique for pulse-code modulation, although it was not published as a journal article until 1982. It is attractive in practice because it is simple and it is generally very fast. The algorithm aims to minimize the within-cluster variance, which is the sum of squared distances between each data point and the centroid of its assigned cluster and assigns inputs to the nearest centroid.

2.1.2 The principles of the K-means Algorithm

1. Choose the number of clusters (k) and randomly place initial centroids in the feature space.
2. Assign each data point to the nearest centroid, forming initial clusters.
3. The algorithm calculates the mean of the data points in each cluster to obtain updated centroids.
4. Reassign each data point to the nearest updated centroid.
5. Repeat steps 3 and 4 until convergence, where the centroids no longer significantly change or for a predefined number of iterations.

2.1.3 K-means Measuring

K-means assigns data points to the nearest cluster centroid based on the Euclidean distance measure. The Euclidean distance calculates the shortest distance between two points in Euclidean space.

Euclidean distance $= \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2}$

Given a set of observations $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$ and let be $\mathbf{C} = \{C_1, C_2, \ldots, C_k\}$ a clustering solution, the minimized cost function J for the K-means algorithm is given by:

$$J(\mathbf{C}) = \sum_{i=1}^{k} \sum_{\mathbf{x}_j \in C_i} ||\mathbf{x}_j - \mu_i||^2$$

Where:

J is the minimized cost function,

k is the number of clusters,

n is the total number of data points,

$\mathbf{x}_j$ represents the j^{th} data point,

μ_i represents the centroid of the i^{th} cluster,

$||\mathbf{x}_j - \mu_i||^2$ is the squared Euclidean distance between data point $\mathbf{x}_j$ and centroid μ_i.

2.1.4 Advantages of K-means Algorithm

1. Simplicity, Efficiency, and Scalability: K-means is straightforward to understand and implement, making it computationally efficient and suitable for large datasets.

2. Interpretability: The resulting clusters from K-means are easy to interpret, especially when clusters are compact and well-separated. Each cluster is represented by its centroid, allowing for an intuitive understanding of cluster characteristics.

3. Flexibility: K-means can be applied to various types of data and across different domains. It works only with numerical data; however, it can also be adapted to handle categorical data by encoding them appropriately.

4. Linear Time Complexity: The time complexity of K-means is linear with respect to the number of data points, making it efficient for large datasets.

5. Ease of Implementation: It's available in many libraries and software packages, making it easily accessible for implementation in various programming languages.

2.1.5 Limitations of K-means Clustering

1. Sensitive to Initial Centroid Selection: K-means performance can vary significantly based on the initial placement of centroids. Random initialization may lead to different clustering results in each run, potentially resulting in suboptimal solutions.

2. Requires Specifying the Number of Clusters (K): Determining the optimal number of clusters (K) can be challenging and may require domain knowledge or trial and error. Selecting an inappropriate K value can lead to poor clustering results.

3. Sensitive to Outliers: Outliers can significantly impact the centroids' positions and deform the cluster boundaries. K-means is sensitive to outliers, potentially resulting in clusters that do not accurately represent the underlying data distribution.

4. May Converge to Local Optima: K-means is prone to converging to local optima, especially when initialized with random centroids. It may fail to find the globally optimal solution, particularly in high-dimensional or complex datasets.

5. Does Not Handle Non-Numeric Data: K-means is designed for numerical data and cannot handle categorical or mixed data types directly. Preprocessing categorical variables into numerical representations may lead to loss of information or introduce bias in the clustering results.

2.1.6 The main applications of the K-means Algorithm

According to [Jai10], K-means clustering is widely used in various domains, such as applications such as Internet search, digital imaging, and video surveillance.

2.1.7 Practical tips for using K-means

When using the K-means clustering algorithm in practice, several tips can help ensure its effectiveness and efficiency:

1. Choose the Right K: Selecting the appropriate number of clusters (K) is crucial. Consider using techniques such as the elbow method or silhouette analysis to determine the optimal K value.

2. Preprocess Data: Prepare the data by scaling or normalizing features to ensure that all dimensions contribute equally to the clustering process. Additionally, address missing values or outliers, as they can significantly impact cluster formation.

3. Iterative Refinement: Run the algorithm multiple times with different initializations to reduce the risk of getting stuck in suboptimal solutions. Evaluate the clustering results using appropriate metrics and select the best-performing clustering solution.

4. Consider Scalability: K-means can struggle with large datasets or high-dimensional data due to computational complexity and sensitivity to outliers. Consider using variants like mini-batch K-means or employing dimensionality reduction techniques before clustering.
5. Interpret Results: After clustering, analyze the resulting clusters to understand their characteristics and interpretability. Visualize clusters using scatter plots, heatmaps, or t-SNE embeddings to gain insights into the underlying data structure.

2.2 K-modes

2.2.1 Definition:

KModes clustering is an algorithm used for clustering categorical data. In 1998, Huang [Hua98] proposed the k-modes algorithm for categorical data clustering. It is a variation of the popular K-means clustering algorithm, which is designed for numerical data. KModes works by identifying the modes or most frequent values within each cluster to determine its centroid. This allows it to effectively group similar data points based on their categorical attributes.

2.2.2 The principles of the K-modes Algorithm

1. Pick K observations at random and use them as leaders/clusters.
2. Calculate the dissimilarities and assign each observation to its closest cluster.
3. Define new modes for the clusters.
4. Repeat steps 2-3 until there is no re-assignment required.

2.2.3 K-modes Measuring

In K-modes algorithm, we use the dissimilarity measure to check the dissimilarity between data points. The dissimilarity measure is based on the number of differing categorical attributes between two data points. It counts the number of categorical attributes that are different between the data points, providing a measure of dissimilarity suitable for categorical data.

Mode Dissimilarity$(x, y) = \sum_{i=1}^{n} \delta(x_i, y_i)$

$$\delta(x_i, y_i) = \begin{cases} 0 & \text{if } x_i = y_i \\ 1 & \text{otherwise} \end{cases}$$

The minimized cost function J for the K-modes algorithm is given by:

$$J = \sum_{i=1}^{n} \sum_{j=1}^{k} \sum_{l=1}^{m} \delta(\mathbf{x}_{il}, \mu_{jl})$$

Where:

J is the minimized cost function,
n is the total number of data points,
k is the number of clusters,
$\mathbf{x}_i$ represents the i^{th} data point,
μ_j represents the mode of the j^{th} cluster,
$\delta(\mathbf{x}_{il}, \mu_{jl})$ is the dissimilarity measure between $\mathbf{x}_{il}$ and μ_{jl}.

2.2.4 Advantages of KModes Clustering

1. Handling Categorical Data: K-modes is specifically designed for clustering categorical data, making it suitable for datasets with non-numeric attributes. It can effectively handle nominal, ordinal, and binary categorical variables.

2. Interpretability: K-modes produces clusters that are easily interpretable, particularly in the context of categorical data. The resulting clusters consist of similar categorical patterns, allowing for straightforward interpretation and actionable insights.

3. Robustness to Outliers: Unlike K-means, which is sensitive to outliers in numerical data, K-modes is more robust to outliers in categorical data. Outliers may not significantly impact the clustering results, as dissimilarity is primarily based on categorical attribute differences.

4. Efficient Computation: K-modes is computationally efficient, particularly for large datasets with categorical attributes. It can handle high-dimensional categorical data without significantly increasing computational complexity.

5. No Assumption of Numeric Continuity: K-modes does not assume any numeric continuity in the data, making it suitable for datasets where categorical attributes are not naturally ordered or do not have a meaningful numerical representation.

2.2.5 Limitations of KModes Clustering

1. Sensitive to Initial Centroid Selection: Similar to K-means, K-modes clustering is sensitive to the initial placement of centroids. Different initializations can lead to different clustering results, impacting the quality of the final clusters.

2. Difficulty Handling High Cardinality Categorical Data: K-modes may struggle with datasets containing categorical variables with a high number of unique values (high cardinality). The dissimilarity measure used in K-modes is based on individual attribute differences, which can be less effective for high cardinality attributes.

3. Assumption of Equal Attribute Importance: K-modes assumes that all categorical attributes have equal importance in determining dissimilarity. However, in real-world scenarios, some attributes may be more informative or relevant for clustering than others.

4. Difficulty with Varying Data Scales: K-modes treats all categorical attributes equally, regardless of their scale or magnitude. This can lead to suboptimal clustering results when attributes have different levels of importance.

5. Scalability: While K-modes is generally efficient, it may struggle with scalability for large datasets with many categorical attributes or high-dimensional categorical data. The computational complexity can increase significantly with the number of unique attribute values and the number of clusters.

2.2.6 The main applications of the K-modes Algorithm

K-modes clustering is commonly implemented in fields such as marketing, healthcare, text mining and Natural Language Processing (NLP), E-commerce and recommendation Systems, fraud Detection, and Anomaly Detection.

2.2.7 Practical tips for using KModes

1. Data Preparation: Preprocess categorical data by encoding it appropriately. Use techniques such as one-hot encoding, label encoding, or frequency encoding to represent categorical variables as numerical values suitable for K-modes clustering.

2. Feature Selection: Carefully select relevant categorical features for clustering. Exclude irrelevant or redundant attributes that may not contribute significantly to cluster formation or interpretation.

3. Optimal Number of Clusters (K): Determine the optimal number of clusters (K) using evaluation metrics like silhouette score or elbow method. Experiment with different K values to find the most meaningful partitioning of the data.

4. Iterative Refinement: Run the K-modes algorithm multiple times with different initializations to reduce the risk of converging to local optima. Evaluate clustering results using different validation metrics and select the best-performing solution.

5. Visualize Results: Visualize clustering results using techniques such as cluster profiles, cluster heatmaps, or parallel coordinate plots to gain insights into cluster characteristics and facilitate interpretation.

2.3 K-prototypes

2.3.1 Definition:

K-Prototypes clustering is a clustering algorithm that is specifically designed to handle datasets with mixed data types, including both numerical and categorical variables. It is an extension of the K-Means and K-Modes clustering algorithms. The K-Prototypes algorithm was proposed by Huang [Hua98] as a solution to the problem of clustering data that contains both numerical and categorical variables. It is based on the partitioning approach used in K-Means and K-Modes, but incorporates a cost function that takes into account the different data types. This cost function allows the algorithm to handle mixed data types effectively.

2.3.2 The principles of the K-prototypes algorithm

1. Randomly select initial prototypes for numerical and categorical attributes.

2. Calculate Euclidean distances and dissimilarities and assign data points to the nearest prototypes.

3. Update prototypes using mean for numerical attributes and mode for categorical attributes.

4. Repeat assignment and prototypes update until convergence criteria are met.

2.3.3 K-prototypes Measuring

According to [Hua98], the K-prototypes Dissimilarity formula (Equation 2.3.3):

$$\text{K-prototypes Dissimilarity} = \sum_{j=1}^{p}(x_j, y_j)^2 + \gamma \sum_{j=p+1}^{m} \delta(x_j, y_j)$$

$$\delta(x_i, y_i) = \begin{cases} 0 & \text{if } x_j = y_j \\ 1 & \text{if } x_j \neq y_j \end{cases}$$

γ is the weighting factor for the categorical attributes. Adjusting the value of γ allows you to control the relative importance of the categorical attributes compared to the numerical attributes in the clustering process.

$$\text{Minimized Cost function} = \sum_{l=1}^{k} \left(\sum_{i=1}^{n} \sum_{j=1}^{p} (x_{i,j} - q_{l,j})^2 + \gamma \sum_{i=1}^{n} \sum_{j=p+1}^{m} \delta(x_{i,j}, q_{l,j}) \right)$$

Where:

k is the number of clusters,
n is the total number of data points,
m is the total number of attributes,
p is the number of numerical attributes,
$x_{i,j}$ represents the j^{th} numerical attribute of the i^{th} data point,
$q_{l,j}$ represents the j^{th} numerical attribute of the centroid of the l^{th} cluster,
γ is a weighting factor for the categorical attributes.

2.3.4 Advantages of K-prototypes Clustering

1. Handling Mixed Data Types: K-prototypes algorithm can handle datasets with both numerical and categorical attributes, making it suitable for real-world datasets that often contain a mix of different types of variables.

2. Improved Cluster Interpretability: By incorporating both numerical and categorical attributes, K-prototypes can create more interpretable clusters that capture diverse characteristics of the data. This allows for deeper insights and a better understanding of cluster patterns.

3. Preservation of Information: By considering both numerical and categorical attributes simultaneously, K-prototypes preserves more information present in the dataset compared to algorithms that handle only one type of variable. This can lead to more accurate and meaningful clustering results.

4. Robustness to Outliers: K-prototypes is robust to outliers in the data, particularly when the dissimilarity measure is appropriately weighted for numerical and categorical attributes. This ensures that clusters are less likely to be influenced by irrelevant or anomalous data points.

5. Scalability: With efficient implementation, K-prototypes can handle large-scale datasets with mixed data types, making it suitable for clustering tasks in big data analytics and data mining applications.

2.3.5 Limitations of K-prototypes Clustering

1. Complexity of Tuning Parameters: K-prototypes requires tuning parameters such as the number of clusters (K), weighting factors for numerical and categorical attributes, and the dissimilarity measure parameter (gamma). Determining optimal values for these parameters can be challenging and may require experimentation or domain expertise.

2. Sensitive to Initialization: Similar to other clustering algorithms, K-prototypes clustering is sensitive to the initialization of centroids. Different initializations can lead to different clustering results, impacting the quality and stability of the clusters obtained.

3. Difficulty Handling High-Dimensional Data: K-prototypes may struggle with high-dimensional datasets containing a large number of numerical and categorical attributes. As the number of dimensions increases, the computational complexity and memory requirements of the algorithm also increase, potentially leading to scalability issues.

4. Interpretability of Clusters: While K-prototypes can create more interpretable clusters by incorporating both numerical and categorical attributes, interpreting clusters with mixed data types may still be challenging. The interpretation of clusters may require careful examination of the contribution of each attribute type to cluster formation.

5. Subjectivity in Attribute Selection: Selecting relevant attributes for clustering can be subjective and dependent on domain knowledge. Including irrelevant or redundant attributes may lead to suboptimal clustering results and decreased interpretability of the clusters obtained.

2.3.6 The main applications of the K-prototypes Algorithm

K-prototypes clustering is implemented in various fields and industries, including such as marketing, healthcare, E-commerce, finance and cybersecurity, manufacturing and Text Mining and NLP

2.3.7 Practical tips for using K-prototypes

1. Data Preprocessing: Preprocess your data by encoding categorical variables appropriately. Use techniques like one-hot encoding, label encoding, or frequency encoding for categorical attributes before applying K-prototypes clustering.

2. Attribute Selection: Carefully select numerical and categorical attributes relevant to your analysis objectives. Include attributes that capture meaningful information and are likely to contribute to cluster formation.

3. Parameter Tuning: Experiment with different values for parameters such as the number of clusters (K), weighting factors for numerical and categorical attributes, and dissimilarity measure parameter (gamma). Use validation techniques to identify optimal parameter settings. Pay attention to the initialization of centroids, as it can significantly impact clustering results.

4. Handling Missing Values: Decide on a strategy for handling missing values in both numerical and categorical attributes. Choose appropriate imputation methods or consider creating a separate category for missing values.

5. Interpretability: Interpret resulting clusters by analyzing the predominant numerical and categorical attributes within each cluster. Use visualization techniques to gain insights into cluster characteristics and facilitate interpretation.

3 Experiments

In this report, the application of these three prominent clustering algorithms in unsupervised machine learning: K-means, K-modes, and K-prototypes are explored across various datasets. The accuracy score and adjusted rand score were calculated and served as evaluation metrics to measure the clustering performance and compare it against ground truth labels, that were available for comparing purposes.

3.1 Datasets

Iris Dataset

This dataset consists of 150 samples of iris flowers, where each sample is described by four numerical attributes: sepal length, sepal width, petal length, and petal width. The dataset is commonly used in machine learning and pattern recognition tasks due to its simplicity and clarity. The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. The dataset used can be found at: https://archive.ics.uci.edu/dataset/53/iris.

Digits Dataset

In addition to the Iris Dataset, the K-means algorithm was also implemented on the Digits dataset. The Digits dataset comprises 1797 samples of hand-written digits, containing 10 clusters, each representing each number from 0 to 9, with each digit represented by a 64-dimensional feature vector. After loading the dataset, we extracted the feature matrix X containing the numerical attributes of the digit images. This dataset serves as a classic benchmark in the field of machine learning, often used for evaluating clustering algorithms' performance in recognizing and grouping similar handwritten digits. To use this dataset, you need to import it using this approach:
`from sklearn.datasets import load_digits`.

```
data, labels = load_digits(return_X_y=True)
```

Vote Dataset

Moving on to the next stage, the K-modes algorithm was implemented on the vote dataset. This dataset contains information about the voting records of members of the U.S. House of Representatives on 16 different issues. It has 435 samples and the number of clusters k was specified to be 2. Unlike the Iris Dataset, the vote dataset is categorical (with 'yes', or 'no'), but it's been encoded, with each attribute representing a vote (e.g., '1' for yes, '2' for no). By using K-modes clustering, we aimed to uncover patterns in the voting behavior of legislators and identify groups of representatives with similar voting patterns. The dataset used can be found at:
https://archive.ics.uci.edu/dataset/105/congressional+voting+records.

Soybean Dataset

Another K-modes implementation was used on the Soybean dataset. The small soybean dataset, contains 47 samples of soybean plants. Each sample represents a single plant and is characterized by a set of categorical attributes, including leaf shape, size, color, texture, and the presence or absence of specific symptoms or conditions. The dataset encompasses 4 distinct classes, denoted as D1, D2, D3, and D4, which represent different types of diseases or conditions affecting soybean plants. With its focus on categorical attributes and class labels, this dataset serves as a valuable resource for developing and evaluating classification algorithms tailored to agricultural applications, particularly in the realm of plant pathology and disease diagnosis. The dataset used can be found at: https://archive.ics.uci.edu/dataset/91/soybean+small.

Autism Dataset

The Autism dataset, according to [Tha17], is a crucial resource in the domain of Autistic Spectrum Disorder (ASD) screening for toddlers. ASD is a neurodevelopmental condition associated with substantial healthcare costs, and early diagnosis can significantly reduce these expenses. It comprises ten behavioral features, derived from the Q-Chat-10 questionnaire, along with other individual characteristics that have proven effective in detecting ASD cases from controls in behavioral science. The dataset, with 1054 samples and 18 attributes including 2 classes variable (yes, no), supports predictive and descriptive tasks, with attributes encompassing nominal/categorical, binary, and continuous data types. It serves as a valuable resource for classification, clustering, association, and feature assessment analyses in the medical, health, and social science domains. Notably, missing values are not present in the dataset, enhancing its reliability for analysis purposes. The dataset used can be found at: https://www.kaggle.com/datasets/fabdelja/autism-screening-for-toddlers?resource=download.

Dermatology Dataset

The Dermatology dataset, according to Ilter,Nilsel and Guvenir,H. [IG98], aims to determine the type of Eryhemato-Squamous Disease. This database contains 34 attributes, 33 of which are linear valued and one of them is nominal.Dataset Characteristics: Multivariate, Subject Area being Health and Medicine, associated tasks: Classification, feature type: categorical, integer. There are 366 instances, 34 features and 6 classes. The differential diagnosis of erythemato-squamous diseases is a real problem in dermatology. They all share the clinical features of erythema and scaling, with very little differences. The diseases in this group are psoriasis, seboreic dermatitis, lichen planus, pityriasis rosea, cronic dermatitis, and pityriasis rubra pilaris. Usually a biopsy is necessary for the diagnosis but unfortunately these diseases share many histopathological features as well. Another difficulty for the differential diagnosis is that a disease may show the features of another disease at the beginning stage and may have the characteristic features at the following stages. Patients were first evaluated clinically with 12 features. Afterwards, skin samples were taken for the evaluation of 22 histopathological features. The values of the histopathological features are determined by an analysis of the samples under a microscope. In the dataset constructed for this domain, the family history feature has the value 1 if any of these diseases has been observed in the family, and 0 otherwise. The age feature simply represents the age of the patient. Every other feature (clinical and histopathological) was given a degree in the range of 0 to 3. Here, 0 indicates that the feature was not present, 3 indicates the largest amount possible, and 1, 2 indicate the relative intermediate values. The names and id numbers of the patients were recently removed from the database. The dataset used can be found at: https://archive.ics.uci.edu/dataset/33/dermatology.

Heart Dataset

According to [JAS+88], the Heart dataset, encompassing data from four databases (Cleveland, Hungary, Switzerland, and the VA Long Beach), serves as a vital resource in the field of health and medicine, particularly for the classification of heart disease. With its multivariate nature and categorical, integer, and real feature types, the dataset facilitates various classification tasks related to heart disease diagnosis. Despite containing 76 attributes, published experiments focus on a subset of 14 attributes, particularly emphasizing the Cleveland database, which has been extensively used by machine learning researchers. The "goal" field within the dataset indicates the presence of heart disease in patients, ranging from 0 (no presence) to 4, with experiments typically concentrating on distinguishing between presence (values 1, 2, 3, 4) and absence (value 0) of heart disease. Notably, the dataset has undergone anonymization, with patient names and social security numbers replaced by dummy values to ensure privacy. With 303 instances, 13 features, and 2 classes, the dataset presents a comprehensive overview of heart disease characteristics across different populations and healthcare settings. Despite the presence of missing values, the dataset remains a valuable asset for researchers and practitioners in the field of cardiology, providing insights into risk factors and diagnostic indicators associated with heart disease. The dataset used can be found at: https://archive.ics.uci.edu/dataset/45/heart+disease.

Datasets	Number of samples	Number of classes	Number of attributes	Dataset type
Iris	150	3	4	Numerical
Digits	1797	10	64	Numerical
Vote	435	2	16	Categorical
Soybean	47	4	35	Categorical
Autism	1054	2	18	Heterogeneous
Dermatology	366	6	34	Heterogeneous
Heart	303	2	13	Heterogeneous

Table 1: Table that describes all the mentioned datasets

3.2 Performance Metrics

In this study, two metrics were used, Accuracy Score and Adjusted Rand Score, to evaluate the performance of various clustering algorithms on the datasets under consideration. The Accuracy Score assesses the proportion of correctly classified instances among all instances in the dataset, providing insight into the clustering algorithm's ability to accurately assign data points to their respective clusters. On the other hand, the Adjusted Rand Score measures the similarity between the clustering results and the ground truth labels, accounting for chance agreement between clusters. By using these two metrics across different datasets and clustering algorithms, a comprehensive assessment of the clustering methods' effectiveness in capturing the underlying structures and patterns within the data was achieved. This approach allowed for informed comparisons and insights into the relative strengths and weaknesses of each algorithm in handling the specific characteristics of the datasets and their respective clustering tasks.

3.3 Preprocessing of Datasets

In preparing the datasets for clustering algorithms, various preprocessing techniques can be applied to handle different data types effectively. For numerical datasets, normalization or standardization methods can be employed if needed, to ensure uniform scaling of features, thereby preventing any particular attribute from dominating the clustering process due to its scale. Concerning the Categorical datasets, they underwent encoding procedures such as Label Encoding or One Hot Encoding to transform categorical variables into numerical representations that clustering algorithms could interpret. Label Encoding assigned a unique integer to each category, while One Hot Encoding created binary columns for each category, indicating its presence or absence. Additionally, for datasets containing mixed types of data, discretization techniques were utilized to convert continuous variables into categorical ones, facilitating their integration into the clustering process for the the K-modes algorithm. Notably, the K-prototypes algorithm stands out in this context, as it inherently accommodates mixed data types without requiring any preprocessing or encoding steps. This unique capability enables K-prototypes to seamlessly handle diverse datasets encompassing both numerical and categorical attributes, simplifying the clustering process and enhancing its applicability across a wide range of real-world scenarios.

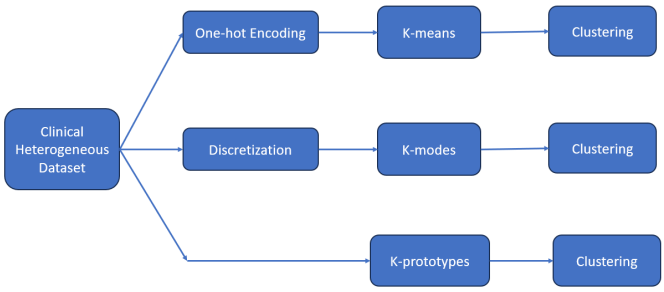
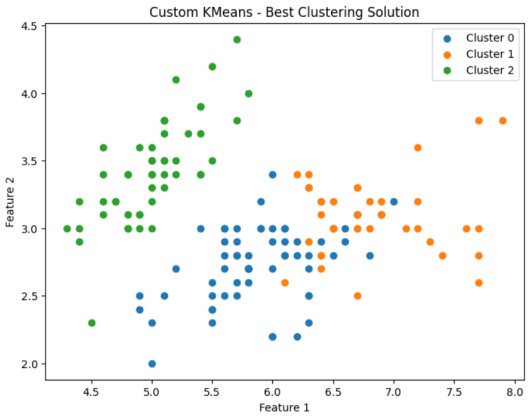


Figure 2: A graph describing the different possible implementations on a Heterogeneous dataset

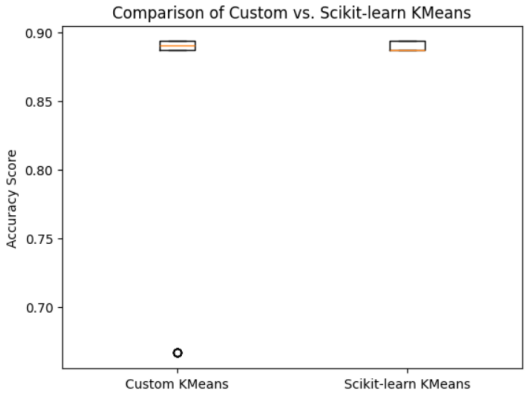
4 Results

4.1 Experiment I: Results on Numerical datasets

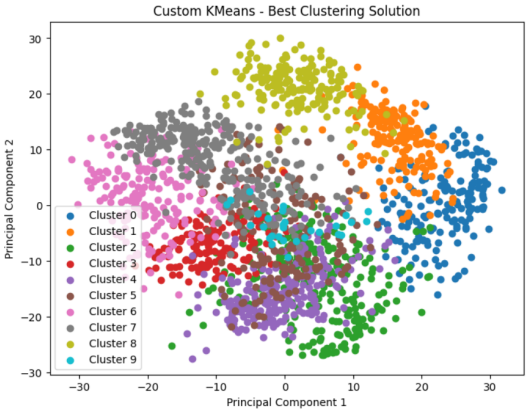
We aimed to group samples from the 'Iris' & 'Digits' datasets using the K-means algorithm, into distinct clusters based on their feature similarities, with the ultimate goal of understanding the natural grouping of different iris species & digits present in the datasets. We plotted some graphs too to get some visualizations and get better understandings. The accuracy score and adjusted rand score were utilized as metrics to assess the clustering performance and compare it against ground truth labels that were available after repeating the K-means 50 times iteratively on these datasets.



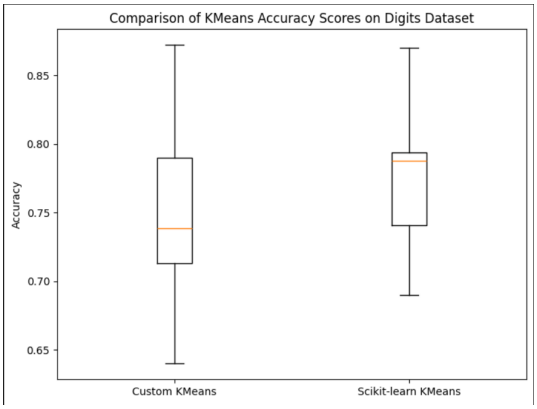
My K-means results on the Iris dataset



My K-means results on the Iris dataset



My K-means results on the Digits dataset



My K-means results on the Digits dataset

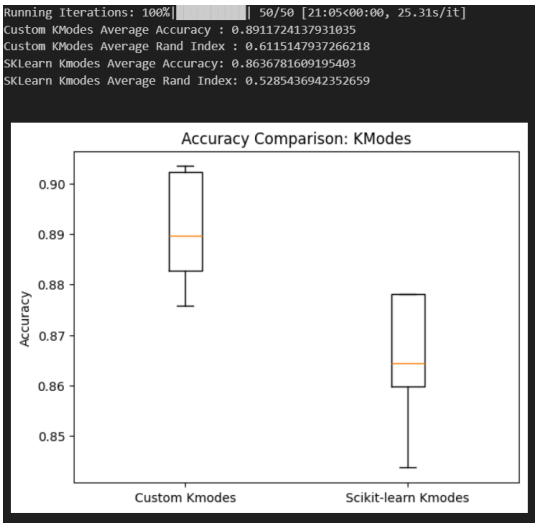
Figure 3: Scatter and Box plots of the Iris & Digits datasets respectively, using my K-means implementation

K-means	Accuracy Score	Adjusted Rand score
Iris Dataset	0.8539 (± 0.082)	0.6917 (± 0.070)
Digits Dataset	0.7479 (± 0.041)	0.6208 (± 0.036)

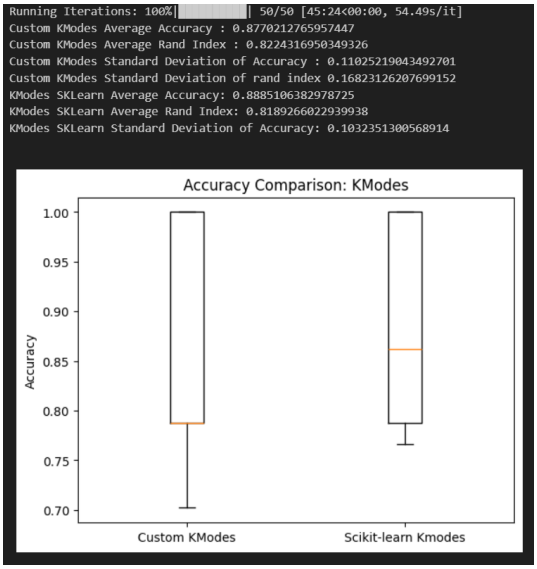
Table 2: Table describing my obtained K-means Accuracies and Adjusted Rand scores on Numerical datasets

4.2 Experiment II: Results on Categorical datasets

we also applied the K-modes algorithm to group samples from the 'Vote' and 'Soybean' datasets, aiming to recognize distinct clusters based on categorical feature similarities. The primary objective was to reveal inherent patterns within the datasets, shedding light on natural groupings present in the political voting records and soybean plant attributes, respectively. After running the K-modes algorithm iteratively 50 times to ensure robust results, we plotted boxplots to gain deeper insights into the clustering accuracies and compared my custom K-modes implementation accuracies with the scikit-learn accuracies. These metrics facilitated a comprehensive evaluation by comparing the clustering results against the ground truth labels available in the datasets. By employing rigorous iterations and robust evaluation metrics, this study aims to provide valuable insights into the clustering performance and the underlying structures of the 'Vote' and 'Soybean' datasets.



My K-modes results on the Vote dataset



My K-modes results on the Soybean dataset

Figure 4: Boxplots of the Vote & Soybean datasets respectively, using my K-modes implementation

K-modes	Accuracy Score	Adjusted Rand score
Vote Dataset	0.8912 ($\pm$ 0.010)	0.6115 ($\pm$ 0.031)
Soybean Dataset	0.8770 ($\pm$ 0.110)	0.8224 ($\pm$ 0.168)

Table 3: Table describing my obtained K-modes Accuracies and Adjusted Rand scores on Categorical datasets

4.3 Experiment III: Results on Medical Heterogeneous datasets

Finally, we employed the K-prototypes algorithm to cluster samples from heterogeneous multi-typed datasets, on the 'Autism', 'Dermatology', and 'Heart' datasets. The objective was to find distinct clusters based on the combined numerical and categorical features present in these datasets. By leveraging the unique capability of K-prototypes to handle mixed data types, we aimed to uncover inherent patterns within the datasets, shedding light on natural groupings in autism diagnostic records, dermatological conditions, and heart disease diagnoses. To ensure robust and reliable results, the K-prototypes algorithm was executed iteratively 50 times. The clustering performance was quantitatively evaluated using metrics such as accuracy score and adjusted rand score, enabling a comprehensive assessment by comparing the clustering results against the ground truth labels available in the datasets. Through rigorous iterations and robust evaluation metrics, this study endeavors to offer valuable insights into the clustering performance and the hidden relationships of the 'Autism', 'Dermatology', and 'Heart' datasets.

K-prototypes	Accuracy score	Adjusted Rand score
Autism Dataset	0.6974 (± 0.031)	0.0205 (± 0.095)
Dermatology Dataset	0.3704 (± 0.029)	0.0643 (± 0.025)
Heart Dataset	0.5771 (±0.005)	0.0229 (± 0.003)

Table 4: Table describing my obtained K-prototypes Accuracies and Adjusted Rand scores on Heterogeneous clinical datasets

Using different approaches to cluster clinical multi-typed datasets

Accuracy Score	Autism Dataset	Dermatology Dataset	Heart Dataset
K-means	0.5522 (± 0.00)	0.3704 (± 0.00)	0.5771 (± 0.00)
K-modes	0.7818 (± 0.00)	0.5492 (± 0.00)	0.7844 (± 0.00)
K-prototypes	0.6974 (± 0.031)	0.3704 (± 0.029)	0.5771 (±0.005)

Table 5: Table representing the my obtained Accuracy Scores after 50 runs on each clustering algorithm on Clinical datasets

5 Conclusion

In conclusion, this research project has been a remarkable experience through the fascinating realm of unsupervised machine learning. Working with datasets deprived of labels has provided invaluable insights into the complexities of clustering algorithms such as K-means, K-modes, and K-prototypes. Through this exploration, I have gained a deeper understanding of how these algorithms operate and their applicability to datasets of diverse data types. I am grateful for the guidance and support of my research professor, Dr. Adán JOSÉ-GARCÍA, whose expertise and assistance have been crucial in navigating this complex field. His persistent support and mentorship made this learning experience both smooth and enriching. Additionally, I extend my gratitude to the esteemed members of the INFINITE team, including Pr. Vincent Sobanski, Pr. Clarisse Dhaenens, and PhD student Clement Chauvet, for welcoming me at the CHU de Lille and engaging in insightful discussions about the practical applications of unsupervised machine learning clustering algorithms in the medical domain. Their expertise and feedback have further broadened my perspective and enriched my understanding of the field.

In addition to the enriching experience gained from this research project, I found the intersection of artificial intelligence and medicine to be particularly captivating and promising. The potential for AI to revolutionize healthcare by aiding in diagnostics, treatment planning, and patient care is truly remarkable. This fascination led me to pursue an internship opportunity at the CHU de Lille, where we will be coding deep learning neural network models for the classification of medical images captured through microscopes. This hands-on experience will further deepen my understanding of the practical applications of AI in the medical domain and contribute to the advancement of healthcare technology.

Interested in exploring or learning some of the work I've conducted during my research? Feel free to visit my GitHub page where I've shared my projects and findings. You can find it at https://github.com/GabrielWarde/ResearchProject1. There, you'll discover the code, data, and insights from my research experience. We hope you find it insightful and valuable for your own interests or projects.

References

[ESA+] Absalom E. Ezugwu, Amit K. Shukla, Moyinoluwa B. Agbaje, Olaide N. Oyelade, Adán José-García, and Jeffery O. Agushaka. Automatic clustering algorithms: a systematic review and bibliometric analysis of relevant literature. DOI: https://doi.org/10.1007/s00521-020-05395-4.

[Hua98] Zhexue Huang. Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery*, 1998.

[IG98] Nilsel Ilter and H. Guvenir. Dermatology. UCI Machine Learning Repository, 1998. DOI: https://doi.org/10.24432/C5FK5P.

[Jai10] Anil K. Jain. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8):651–666, 2010. Award winning papers from the 19th International Conference on Pattern Recognition (ICPR).

[JAS+88] Janosi, Andras, Steinbrunn, William, Pfisterer, Matthias, Detrano, and Robert. Heart Disease. UCI Machine Learning Repository, 1988. DOI: https://doi.org/10.24432/C52P4X.

[Tha17] F. Thabtah. Autism spectrum disorder screening: Machine learning adaptation and dsm-5 fulfillment. In *Proceedings of the 1st International Conference on Medical and Health Informatics 2017*, pages 1–6, Taichung City, Taiwan, 2017. ACM.