DOI 10.31891/2307-5732-2023-329-6-20-27
UDC 004.82

BOYKO NATALIYA
Lviv Polytechnic National University
https://orcid.org/0000-0002-6962-9363
e-mail: Nataliya.i.boyko@lpnu.ua
CHUKLA OREST
Lviv Polytechnic National University
https://orcid.org/0009-0006-6347-4918
e-mail: orest.chukla@gmail.com

A COMPARATIVE EXAMINATION OF THE EFFICACY OF VARIOUS CLUSTERING METHODS IN ASSESSING WINE QUALITY

The paper considers a comparative analysis of the effectiveness of various clustering methods for the purpose of assessing wine quality. Various clustering algorithms have been studied, including K-means algorithms, hierarchical clustering, DBSCAN, and others. Their effectiveness and usefulness for evaluating wine quality is compared. In the work, the main attention is paid to the analysis of clustering results, computational speed and efficiency of processing large volumes of data. A comparison of the research results was made, which will help establish which clustering method is the most effective and accurate for assessing wine quality. In the study, wine data was collected to conduct a comparative analysis of the effectiveness of different clustering methods, and data on the chemical composition and physical properties of wine will be used. Wine data was processed and prepared for analysis and use in machine learning methods. Preparation and cleaning of data from inconsistencies was carried out. The effectiveness of clustering methods was evaluated. Indicators of cluster quality were used to compare the effectiveness of different clustering methods. The result was discussed. The results of cluster analysis were analyzed, conclusions were drawn about the most effective clustering method for assessing wine quality. The results of the study, which can be useful for consumers in choosing wine from a large assortment, are substantiated. Various methods of clustering and their application on the example of wine data are studied, which is an important step in using machine learning to solve practical problems. The results obtained can be useful for winemakers and experts in the field of winemaking for more accurate classification and improvement of production processes. A study was conducted that allows to determine the most effective and accurate method of clustering for the objective assessment of the quality of wines and can serve as a basis for further research in the field of winemaking and data analysis.

Keywords: cluster analysis, K-Means, DBSCAN, Hierarchical Clustering, Gaussian Mixture Models (GMM), machine learning, classifier, categorization, clustering.

БОЙКО НАТАЛІЯ
Національний університет «Львівська політехніка»
https://orcid.org/0000-0002-6962-9363
e-mail: Nataliya.i.boyko@lpnu.ua
ЧУКЛА ОРЕСТ
Національний університет «Львівська політехніка»
https://orcid.org/0009-0006-6347-4918
e-mail: orest.chukla@gmail.com

ПОРІВНЯЛЬНИЙ АНАЛІЗ ЕФЕКТИВНОСТІ РІЗНИХ МЕТОДІВ КЛАСТЕРИЗАЦІЇ ДЛЯ ОЦІНКИ ЯКОСТІ ВИНА

В роботі розглядається порівняльний аналіз ефективності різних методів кластеризації з метою оцінки якості вина. Досліджено різні алгоритми кластеризації, включаючи алгоритми K-means, ієрархічну кластеризацію, DBSCAN та інші. Порівнюється їх ефективність та корисність для оцінки якості вина. В роботі приділено основну увагу аналізу результатів кластеризації, обчислювальній швидкості та ефективності обробки великих обсягів даних. Проведено порівняння результатів дослідження, що допоможе встановити, який метод кластеризації є найбільш ефективним та точним для оцінки якості вина. В дослідженні було зібрано винні дані для проведення порівняльного аналізу ефективності різних методів кластеризації будуть використані дані про хімічний склад та фізичні властивості вина. Було оброблено та здійснено підготовку винних даних для аналізу та використання у методах машинного навчання. Здійснено підготовку та очищення даних від невідповідностей. Оцінено ефективність методів кластеризації. Використано показники якості кластерів, для порівняння ефективності різних методів кластеризації. Проведено обговорення результату. Було проаналізовано результати кластерного аналізу, зроблено висновки про найбільш ефективний метод кластеризації для оцінки якості вина. Обґрунтовано результати дослідження, які можуть бути корисні для споживачів щодо вибору вина з великого асортименту. Досліджено різні методи кластеризації та їх застосування на прикладі винних даних, що є важливим кроком у використанні машинного навчання для рішення практичних завдань. Отримано результати, які можуть бути корисні для виноробів та експертів у галузі винарства для більш точної класифікації та поліпшення виробничих процесів. Проведено дослідження, яке дозволяє визначити найбільш ефективний та точний метод кластеризації для об'єктивної оцінки якості вин та може послужити підґрунтям для подальших досліджень у галузі винарства та аналізу даних.

Ключові слова: кластерний аналіз, K-Means, DBSCAN, Hierarchical Clustering, Gaussian Mixture Models (GMM), машинне навчання, класифікатор, групування, кластеризація.

Problem overview

Wine production stands as a pivotal sector within the realm of the food industry, where product quality assumes paramount significance. One of the methods employed for assessing wine quality is cluster analysis, enabling the classification of wines based on their characteristics and properties. In light of this, a comparative analysis of the effectiveness of diverse clustering methods can lend valuable assistance in the development of novel viniculture technologies and the enhancement of wine quality.

The relevance and novelty of this research lie in the fact that wines constitute a significant commodity of global importance. Evaluating wine quality poses a complex challenge, as it hinges on numerous factors such as alcohol content, acidity, flavor profiles, and much more. Hence, a comparative analysis of the efficacy of various clustering methods for wine quality assessment holds great significance in addressing this task.

Investigating this avenue will reveal the most efficacious clustering methods for wine quality assessment, which can find applications in industry and entrepreneurship. Moreover, the incorporation of cutting-edge machine learning techniques for clustering will yield more precise and reliable results.

Machine learning emerges as a potent tool for conducting cluster analysis on vinous data. To attain maximum effectiveness in cluster analysis, the selection of the most suitable clustering method is imperative. Thus, the objective of this work is to undertake a comparative analysis of different clustering methods for wine quality assessment.

The research object pertains to the process of clustering wines with the aim of assessing their quality and identifying common characteristics among them.

The research subject encompasses the comparative analysis of the effectiveness of various clustering methods for wine quality assessment.

Analysis of recent sources

To ensure the utmost objectivity and precision of results, an essential prerequisite was the comprehensive analysis of as many sources as possible. Among the primary sources to be employed in this work on the topic of 'A Comparative Analysis of the Effectiveness of Various Clustering Methods for Wine Quality Assessment,' one can include scientific articles and research published in specialized journals, as well as books and other publications within this domain. Furthermore, data from various open sources can be utilized to obtain a more realistic assessment of the effectiveness of clustering methods for wine quality assessment. Below, a selection of the most significant sources is noted:

- The article titled 'Clustering performance comparison using K-means and expectation maximization algorithms' is dedicated to comparing the efficacy of two clustering algorithms - K-means and expectation maximization - based on their ability to cluster wines according to their characteristics. The research was conducted through the analysis of wine data collected via chemical analysis and sensory evaluation. The article provides a detailed description of K-means and expectation maximization clustering methods, their advantages and disadvantages, as well as methodologies for selecting the number of clusters and assessing clustering quality. Subsequently, the authors conducted experiments with these algorithms on wine data and compared their effectiveness in terms of accuracy and clustering speed.
- The article 'Enhancing the wine tasting experience using a greedy clustering wine recommender system' describes the creation of a wine recommendation system based on a clustering algorithm. The authors argue that such a system can significantly enhance the wine tasting experience by helping users select wines that best match their taste preferences. To build the recommendation system, the authors used data clustering with a greedy algorithm. Initially, data preprocessing was carried out, reducing data dimensionality and minimizing noise using principal component analysis. Then, data clustering was performed to group the data and cluster similar wines together.
- The article 'Enhancement Clustering Evaluation Result of Davies-Bouldin Index with Determining Initial Centroid of K-Means Algorithm' is dedicated to improving the quality assessment of clustering by enhancing the Davies-Bouldin Index through the determination of initial centroids in the K-means method. The article proposes an algorithm to enhance clustering quality assessment, relying on finding the best initial centroids for the K-means method. Genetic algorithm and random search methods are utilized for this purpose. The results of experiments demonstrated that the proposed algorithm enhances clustering quality and reduces execution time, particularly on large datasets.

Formulation of the goals of the article

The aim of this work is to conduct a comparative analysis of the efficacy of different clustering methods for the assessment of wine quality. The research is focused on studying and contrasting various clustering algorithms, including K-means, hierarchical clustering, DBSCAN, and others, in terms of their effectiveness and suitability for wine quality assessment.

In order to conduct a comparative analysis of the effectiveness of various clustering methods for wine quality assessment, it is essential to comprehend the fundamental principles of clustering and its application in the realm of winemaking. This research will delve into the primary clustering methods, their advantages and disadvantages in the context of wine quality assessment, and analyze practical examples of clustering utilization in the wine industry to enhance the efficiency of wine quality assessment.

So, the task is to compare the effectiveness of various clustering methods for wine quality assessment.

To address this task, it is necessary to gather a dataset of wines containing information about their attributes (such as alcohol content, sugar, acidity, fruitiness, etc.). Subsequently, it is required to compare the effectiveness of different clustering methods, such as K-Means, DBSCAN, Spectral Clustering, Hierarchical Clustering, and others, using clustering quality metrics like Silhouette score, Davies-Bouldin index, and others. For each clustering method, experiments should be conducted by varying algorithm parameters to determine the most optimal parameter values that yield the best results. Following that, the effectiveness of different clustering methods can be compared using a diagram in which each cluster is represented by a different color.

Presentation of the main material

There exists a plethora of clustering methods, yet depending on the nature of the input data and the task at hand, different methods may prove more or less effective. To compare the effectiveness of various clustering methods for categorizing wines based on their characteristics, we can consider the following four methods:

1. K-Means: This is one of the most prevalent and straightforward clustering methods used to partition input data into clusters where objects are close to each other in certain features. In the K-Means method, each cluster is represented by its center, and each data point is assigned to the nearest center. The algorithm iteratively updates the cluster centers until convergence is reached, meaning the clustering stabilizes.

2. DBSCAN: DBSCAN is a density-based clustering method that identifies clusters based on the data density. The algorithm separates data into clusters with high data density, distinguishing them from other data points with low density. DBSCAN can discover clusters of various shapes, does not require a predetermined number of clusters, and helps identify noise points.

3. Hierarchical Clustering: This clustering method employs a tree-like structure to divide input data into clusters. It can be of two types: agglomerative and divisive. In the agglomerative type, each data point initially represents an individual cluster, and the algorithm progressively merges the two closest clusters until a single cluster remains. In the divisive type, all data points start in a single cluster, and at each step, the algorithm divides the cluster into two smaller ones.

4. Gaussian Mixture Models (GMM): GMM is a statistical model used for clustering data by determining the probability of each data point belonging to each cluster. GMM is based on the assumption that data within each cluster follows a Gaussian probability density function. The model consists of several Gaussian distributions, each representing a distinct data cluster. As each cluster can have its own distribution, this model can identify clusters of varying shapes and sizes.

Let us commence with the first clustering method - K-Means.

The K-Means method stands as one of the most popular and simplest clustering algorithms, employed for the partitioning of a dataset into clusters based on their characteristics. In this section, we shall delve into the principles of the K-Means method, elucidating its merits and demerits, and elucidating instances of its application in the assessment of wine quality.

Data classification is predicated upon the formation of clusters, wherein each cluster comprises objects possessing akin characteristics. The determination of the requisite number of clusters is undertaken prior to algorithm execution, typically achieved through estimation or visual data analysis. The K-Means method segregates the dataset into a specific number of clusters (K), which maximize the sum of intra-cluster distances, i.e., distances between objects within the same cluster. The optimal allocation of objects to clusters is attained via iterative recalculation of cluster centroids and the assignment of each object to the nearest centroid.

The mathematical description of the method entails the division of n observations into k clusters in such a manner that each observation is assigned to the cluster with the closest mean. This method is grounded in the minimization of the sum of squared distances between each observation and the center of its cluster, as reflected in the following function (Formula 1):

$$J = \sum_{i=1}^{k} \sum_{x_j \in S_i} (x_j - \mu_i)^2, \qquad 1)$$

where k represents the number of clusters, x_j represents an observation, μ_i denotes the center of a cluster, and S_i represents the obtained clusters.

Let's move on to the second clustering method - DBSCAN.

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is one of the most widely used clustering methods for partitioning objects into clusters. DBSCAN is based on the concept of point density in space, which allows it to discover clusters of any shape and size.

The DBSCAN method has two main parameters that need to be configured: the radius ε and the minimum number of points m. The radius ε defines the sphere's radius around each point, which is used to determine whether this point belongs to a specific cluster. The minimum number of points m determines the minimum number of points that should be within the radius ε to form a cluster.

The main idea of DBSCAN is that points with a density higher than a certain threshold are considered part of clusters, while points with low density are considered noise. Point density is defined as the number of points within a radius ε of a given point. If the number of points within the radius ε is greater than or equal to the minimum count m, then the point is considered a cluster core point. If a point is not a core point but is within the radius ε of a core point, it is considered part of the cluster; otherwise, it is considered noise. In other words, points that do not fall into any density core are regarded as noise points. However, the nearest neighbors of noise points may be part of clusters, so they can be joined to the corresponding clusters if they meet certain conditions, such as the minimum number of points required to form a cluster and the maximum distance between neighboring points within the cluster.

Mathematically, the DBSCAN method uses two parameters:

1. Epsilon (ε): This is the radius that defines the maximum distance between two points to consider them neighbors. It allows you to define the region that includes neighboring points.

2. MinPoints: This is the minimum number of points that must be found within the ε radius for a point to be

considered the core of a cluster.

The DBSCAN method is based on the following concepts:

1. Core Points: A point is a core point if it has at least MinPoints neighbors within the ε radius.
2. Border Points: A point is a border point if it has fewer neighbors than MinPoints but is within the ε radius of a core point.
3. Noise Points: A point is noise if it is neither a core point nor a border point.

Let's move on to the third clustering method - Hierarchical Clustering.

Hierarchical Clustering is a clustering method that initially assigns each object to a separate cluster and then gradually merges them into larger clusters based on similarity. The idea is to find the closest objects and merge them into clusters until all objects are combined into a single cluster.

There are two types of Hierarchical Clustering: Agglomerative and Divisive. Agglomerative starts with each object as a separate cluster and then merges them, while Divisive starts with the entire dataset as one cluster and divides it into smaller clusters.

Also, there are several methods for calculating the distance between clusters, such as Single Linkage, Complete Linkage, and Average Linkage.

- Single Linkage - This method calculates the distance between two clusters by using the distance between the two closest points, one belonging to the first cluster and the other to the second (Formula 2):

$$D(r,\ s) = Min\{d(i,j)\}, \tag{2)}$$

where *i* is a point from cluster *r*, and *j* is a point from cluster *s*.

- Complete Linkage - This method calculates the distance between two clusters by using the distance between the two farthest points, one belonging to the first cluster and the other to the second (Formula 3).

$$D(r,\ s) = Max\{d(i,j)\}\ , \tag{3)}$$

where *i* represents a point from cluster *r*, and *j* represents a point from cluster *s*.

- Average Linkage - computes the distance between two clusters by taking the average of distances between all pairs of points, where one point belongs to the first cluster and the other belongs to the second cluster (Formula 4).

$$D(r,\ s) = \frac{T_{rs}}{N_r * N_s} \tag{4)}$$

where $T_{rs} -$ is the sum of all pairwise distances between cluster *r* and cluster *s*, N_r – is the size of cluster *r*, a N_s – is the size of cluster *s*.

Let us now delve into the fourth method of clustering - Gaussian Mixture Models (GMM).

Gaussian Mixture Models (GMM) is a clustering method founded on the principles of probability theory. It models the data distribution as a sum of several Gaussian distributions (normal distributions).

GMM offers considerably more flexibility compared to K-Means, as it allows for modelling data distributions of various shapes, without being confined solely to circular clusters.

The GMM model consists of multiple Gaussian distributions, where each Gaussian distribution corresponds to an individual cluster. Each Gaussian distribution is described by three parameters: the mean value (*μ*), the covariance matrix (Σ), and a weight or probability (*p*), which represents the likelihood of a data point belonging to the cluster. Consequently, for *k* clusters, the GMM model encompasses *k* Gaussian distributions.

The probability of a sample x belonging to the k-th cluster is calculated using the following Formula 5:

$$P(x|k) = \frac{1}{2\pi^{\frac{D}{2}} * |\Sigma|^{\frac{1}{2}}} * exp\left(-\frac{1}{2} * (x-\mu)^T * \Sigma^{-1} * (x-\mu)\right), \tag{5)}$$

where *x* represents our data points, *D* is the number of dimensions for each data point, *μ* and Σ are the mean and covariance matrix, respectively.

Additionally, the overall probability of *x* belonging to the GMM model is calculated as (Formula 6):

$$P(x) = \sum_{i=1}^{M} p_i P(x|k_i), \tag{6)}$$

where *x* - represents a data vector consisting of *D* dimensions, $P(x|k_i)$ - denotes the probability of vector *x* given cluster *k*, μ_i - signifies the vector of mean values for component *i*, Σ_i - represents the covariance matrix for component *i*.

After analyzing various clustering methods, one may wonder how to compare the results of each clustering method and determine the most effective one. To compare the effectiveness of different clustering methods, various metrics can be employed. Some of the most common metrics include:

- The Silhouette Coefficient is a metric that evaluates the quality of clustering by measuring how well objects fit into their own cluster compared to other clusters. The Silhouette Coefficient can take values from -1 to 1, where values closer to 1 indicate better clustering quality.

- The Davies-Bouldin index is a metric that measures how well each cluster is distinct from other clusters. It depends on the average distance between cluster centers and the sizes of clusters. Smaller values of the Davies-Bouldin index indicate better clustering.
- The Adjusted Rand Index (ARI) is a metric that measures the similarity between the clustering obtained by the algorithm and the ground truth clustering. If ARI equals 0, the clustering is random, and if it equals 1, the clustering is perfect.
- The Calinski-Harabasz index is a metric that measures how well clusters are separated from each other. It depends on the average distance between cluster centers and the sizes of clusters. Higher values of the Calinski-Harabasz index indicate better clustering.
- The Sum of Squared Distances (SSD) metric measures how far points are from the centers of their clusters. Methods that minimize SSD, such as K-Means, typically work well with data where clusters have simple geometry. The smaller the SSD value, the more accurately the clustering reflects the data's structure.

Indeed, to compare different clustering methods, you can use each of these metrics and compare the values obtained as a result of execution.

To mathematically calculate the Silhouette Coefficient, you should follow these steps:

1. Calculate the average distance between an object and all other objects within its cluster. After calculating all such average distances for an object, you will obtain the value *a(i)* for each object i. $a(i)=\frac{1}{|C_i|-1}\sum_{j\in C_i,\, i\neq j} d(i,j)$, where $|C_i|$ is the number of points belonging to cluster C_i, and d(i, j) — is the distance between points i and j within cluster C_i.

2. Calculate the average distance between an object and all the objects in the nearest cluster to it. This value will be denoted as *b(i)* for each object *i*. $b(i)=\min_{J\neq I}\frac{1}{|C_J|}\sum_{j\in C_J} d(i,j)$, where $|C_j|$ is the number of points belonging to cluster C_J, and *d(i, j)* — is the distance between points *i* and *j* in cluster C_j.

3. Calculate silhouette values for each object *i* by dividing the difference between *b(i)* and *a(i)* by the greater of the two values, as follows $\begin{cases} 1-\frac{a(i)}{b(i)},\ if\ a(i)<b(i) \\ 0,\ if\ a(i)=b(i) \\ \frac{b(i)}{a(i)}-1,\ if\ a(i)>b(i) \end{cases}$. From the provided definition, it is clear that $-1\leq s(i)\leq 1$

Next metric to consider is the Davies-Bouldin index. To mathematically calculate the Davies-Bouldin index, the following steps should be performed:

1. Calculate the similarity between clusters *i* and *j* using the following formula: $R_{i,j}=\frac{S_i+S_j}{M_{i,j}}$, where *s(i)* and *s(j)* - represent the dispersion of clusters *i* and *j*, *d(i, j)* - is the distance between the centroids of these clusters.

2. For each cluster *i,* find the maximum similarity value between cluster *i* and all other clusters: $D_i=\max_{j\neq i} R_{i,j}$

3. Calculate the Davies-Bouldin index: $DB=\frac{1}{N}\sum_{i=1}^{N} D_i$, where *N* is the number of clusters, D_i is the maximum similarity value between cluster i and all other clusters

The next metric is the Calinski-Harabasz index. To mathematically calculate the Calinski-Harabasz index, you should follow these steps:

1. Calculate the intra-cluster variability (the sum of squares of deviations within clusters), which equals the sum of the squares of distances between objects within a cluster and its centroid: $\sum W_k=\sum_{k=1}^{K}\sum_{i=1}^{n_k} d_i-c_k^{\,2}$, where d_i-c_k represents the distance between object d_i and the centroid of cluster c_k, and the sum is calculated for all objects i and all clusters *k*.

2. Determine the between-cluster variability (the sum of squared distances between the centroid of individual clusters and the overall centroid, multiplied by the cluster size): $\sum B_k=\sum_{K=1}^{K} n_k c_k-c^2$, were n_k represents the size of cluster *k,* c_k-c denotes the distance between the centroid of cluster c_k and the overall

centroid *c*, and the sum is calculated across all clusters *k*.

3. Calculate the Calinski-Harabasz index using the formula: $CH = \frac{\sum B_k}{K-1} - \frac{\sum W_k}{N-K}$, where *K* is the number of clusters, and *N* is the number of objects.

Another metric is the Adjusted Rand Index (ARI). To mathematically calculate the Adjusted Rand Index, you should follow these steps:

1. To calculate the Unadjusted Rand Index (Rand Index, RI) using the formula: $RI = \frac{a+b}{a+b+c+d}$, where: *a* - the number of objects that belong to the same cluster in both *A* and *B*, *b* - the number of objects that belong to different clusters in both *A* and *B*, *c* - the number of objects that belong to the same cluster in *A* but different clusters in *B*, *d* - the number of objects that belong to different clusters in *A* but the same cluster in *B*.

2. Calculate the expected value of *RI* according to the hypothesis of random dependence: $E(RI) = \frac{(a+c)*(a+b)+(b+d)*(c+d)}{(a+b+c+d)^2}$, where: a is the number of objects that belong to the same cluster in both *A* and *B*, *b* is the number of objects that belong to different clusters in both *A* and *B*, *c* is the number of objects that belong to the same cluster in *A* but different clusters in *B*, *d* is the number of objects that belong to different clusters in *A* but the same cluster in *B*.

3. Calculate the Adjusted Rand Index (*ARI*) by adjusting the *RI* for the probability of random cluster overlap $ARI = \frac{RI - E(RI)}{1 - E(RI)}$, where *RI* is the unadjusted Rand Index, and *E(RI)* is the expected value of *RI* under the hypothesis of random dependence.

The final metric to consider is the Sum of Squared Distances (SSD). To mathematically compute the Sum of Squared Distances, the following steps should be followed:

1. Calculate the squared distance between each object and its cluster centroid $d^2(x,\ C) = x - C^2$, where *x* represents the object, and *C* is the centroid of the cluster to which *x* belongs, $x - C$ denotes the Euclidean distance between *x* and *C*.

2. Sum up the squared distances for all objects and their respective clusters $SSD = \sum d^2(x,\ C)$, where $d^2(x,\ C)$ represents the squared distance between each object and its cluster centroid.

Now, to compare these metrics let's implement methods of clustering. Let's start with the implementation of the first clustering method, K-Means.

To implement this clustering method, we need to import the KMeans class from the sklearn.cluster library. Clustering will be performed using the fit() method. The fit() method in the sklearn.cluster.KMeans library is used to find clusters in the input data. It takes as input a data matrix where each row represents one sample and performs the KMeans model training on this data.

After executing the fit() method, the KMeans model object contains cluster centers and the membership of each point to a cluster. In other words, the model has "learned" to cluster the input data based on the selected parameter n_clusters, which specifies the number of clusters to be found.

So, after performing all the operations, we obtain the metric results for the K-Means clustering method (Table 1):

Table 1

The metric results for the K-Means clustering method

	3 clusters	6 clusters
Silhouette Coefficient	0.504	0.391
Davies-Bouldin index	0.641	0.843
Adjusted Rand Index	0.298	0.153
Calinski-Harabasz index	13900.541	13195.037
Sum of Squared Distances	4309868.868	2037658.902

When it comes to comparison, the K-Means clustering demonstrated good results in terms of Silhouette Coefficient and Calinski-Harabasz index for both cases. This indicates a strong separation and compactness of clusters. However, the K-Means method struggles with cluster overlap, as evidenced by the low Adjusted Rand Index and high Davies-Bouldin index values. K-Means can encounter issues with cluster overlap due to its nature, where each point belongs to only one cluster. This limitation can lead to incorrect cluster boundary determination and a deterioration of Adjusted Rand Index and Davies-Bouldin index metrics in cases of cluster overlap.

Now, let's proceed to the implementation and results for the third clustering method, Hierarchical Clustering.

To implement Hierarchical Clustering, we need to import the Agglomerative Clustering class from the sklearn.cluster library. We will use the fit_predict() function to create our clusters. The fit_predict() method is used to train the model and obtain clusters using Agglomerative Clustering. It takes the feature matrix on which clustering will be performed as its input.

The method returns an array where each element corresponds to the cluster to which the respective sample belongs (Table 2).

Table 2

Results of the metrics for the Hierarchical Clustering method

	3 clusters, single linkage	3 clusters, complete linkage	3 clusters, average linkage	6 clusters, single linkage	6 clusters, complete linkage	6 clusters, average linkage
Silhouette Coefficient	0.688	0.609	0.664	0.230	0.431	0.469
Davies-Bouldin index	0.220	0.389	0.405	0.362	0.554	0.432
Adjusted Rand Index	0.001	0.001	0.001	0.001	-0.004	-0.001
Calinski-Harabasz index	41.107	83.499	58.516	24.694	4304.506	4776.398

When comparing the Hierarchical Clustering method, it showed varying results depending on the linkage method used. Overall, the Single Linkage method yielded the best results in terms of Silhouette Coefficient, Davies-Bouldin index, and Adjusted Rand Index for 3 clusters.

For the implementation and results of the Gaussian Mixture Models (GMM) clustering method:

To implement Gaussian Mixture Models (GMM) clustering, we need to import the GaussianMixture class from the sklearn.mixture library. We will perform clustering using the fit() method. The fit() method is used to train a GMM model on the given data. It finds the parameters that best fit the input data, including the Gaussian distribution parameters determined by the number of components, their weights, means, and covariance matrices. The fit() method modifies the model and returns the modified gmm object (Table 3).

Table 3

Results for the GMM clustering method

	3 clusters	6 clusters
Silhouette Coefficient	0.182	0.043
Davies-Bouldin index	19.147	14.293
Adjusted Rand Index	0.742	0.329
Calinski-Harabasz index	2767.007	1977.159
Sum of Squared Distances	-4.341	-3.594

When comparing the metrics for all other clustering methods above, the Gaussian Mixture Models (GMM) method demonstrated the best results for 3 clusters, with a high Adjusted Rand Index and moderate values for Silhouette Coefficient and Calinski-Harabasz index. This suggests that GMM performed well in terms of capturing the underlying clusters in the data while maintaining a reasonable balance between compactness and separation of clusters.

Let's proceed with the implementation and results for the second clustering method, DBSCAN.

To implement the DBSCAN clustering method, we need to import the DBSCAN class from the sklearn.cluster library and the StandardScaler class from the sklearn.preprocessing library. After that, we will use the DBSCAN class from the sklearn.cluster library to cluster the data with parameters such as eps and min_samples. We will perform clustering using the fit_predict() method.

The fit_predict() method takes as input the feature matrix X and optional parameters of the DBSCAN algorithm, such as eps, min_samples, and metric, and returns an array of cluster labels corresponding to each sample in the input feature matrix X. Clusters are labelled with integers starting from 0, and labels of -1 indicate that a particular sample was not assigned to any cluster.

Similarly, we obtain the metric results for DBSCAN (Table 4):

Table 4

The metric results for DBSCAN

	eps = 7, min_samples = 5	eps = 1.5 та min_samples = 5
Silhouette Coefficient	0.749	-0.214
Davies-Bouldin index	1.360	1.958
Adjusted Rand Index	1.000	0.155
Calinski-Harabasz index	25.191	18.791

Also, comparing the results, it can be said that DBSCAN showed very good results for the first case with a high Silhouette Coefficient, low Davies-Bouldin index, and a high Adjusted Rand Index. This indicates well-separated clusters and their consistency with the real labels. However, for the second case with a smaller eps value and a low Silhouette Coefficient, the results were less satisfactory. A low Silhouette Coefficient suggests cluster overlap and incorrect cluster identification. The Davies-Bouldin index also showed high values, indicating a lack of well-defined clusters.

In summary, when considering the Silhouette Coefficient and Calinski-Harabasz index, the K-Means method performed the best for both cases of cluster count.

DBSCAN yielded good results in terms of Silhouette Coefficient and Adjusted Rand Index but unfairly with respect to the Davies-Bouldin index when parameters were incorrectly set.

The Hierarchical Clustering method with Single Linkage showed favorable results for Silhouette Coefficient, Davies-Bouldin index, and Adjusted Rand Index for 3 clusters.

GMM demonstrated the most promising outcomes for 3 clusters, with a high Adjusted Rand Index and average Silhouette Coefficient and Calinski-Harabasz index values.

Overall, there is no universal clustering method that consistently performs best for all metrics and cluster count variations. Each method has its advantages and limitations, and the choice of method depends on the specific context and research objectives.

Conclusions

In this study, we conducted an analysis of the effectiveness of different clustering methods for evaluating wine quality. The purpose of the research was to determine the most suitable clustering method for use in dietetics to personalize nutrition based on patients' physical conditions.

We justified the relevance and significance of the topic for the wine industry, emphasizing that clustering can assist winemakers in classifying wines by quality and identifying clusters with similar characteristics.

We presented four clustering methods: K-Means, DBSCAN, Hierarchical Clustering, and GMM. Each method underwent a detailed analysis, and we provided descriptions of their working principles and characteristics.

To perform a comprehensive analysis, we utilized diverse sources of information. This included scientific articles from journals, which provided fundamental knowledge about clustering methods and their applications in the wine industry. We also used a dataset of wines containing real data on chemical characteristics and quality ratings. Leveraging these sources contributed to a deeper understanding of the topic and added scientific rigor to the comparative analysis of clustering methods for wine quality evaluation.

Subsequently, a comparative analysis of the results was conducted using various metrics, such as the Adjusted Rand Index, Silhouette Coefficient, Calinski-Harabasz index, and Davies-Bouldin index. The influence of each clustering method on the quality of wine dataset clustering was investigated.

Based on the obtained results, it was revealed that the GMM method demonstrated the best performance for 3 clusters, with a high Adjusted Rand Index and average Silhouette Coefficient and Calinski-Harabasz index values. This indicates that GMM is capable of finding more accurate and distinct clusters within the wine data.

Therefore, based on this analysis, it is recommended to utilize the GMM method for wine quality evaluation. Its ability to model Gaussian distributions in the data allows for the identification of more precise and separable clusters.

However, it's worth noting that the choice of a specific clustering method should depend on the characteristics of the research problem and the researcher's requirements. I also recommend conducting additional experiments with different methods and their parameters to achieve the best results in a specific context.

References

1. Jung Y.G., Kang M.S., Heo J. (2014). Clustering performance comparison using K-means and expectation maximization algorithms. Biotechnology & Biotechnological Equipment, volume 28, pages. 44-48 URL: https://doi.org/10.1080/13102818.2014.949045

2. Katarya R., Saini R. (2021). Enhancing the wine tasting experience using greedy clustering wine recommender system. Multimedia Tools and Applications, volume 81, pages 807–840. URL: https://doi.org/10.1007/s11042-021-11300-5

3. Sitompul B.D., Opim Salim Sitompul O.S., Sihombing P. (2019). Enhancement Clustering Evaluation Result of Davies-Bouldin Index with Determining Initial Centroid of K-Means Algorithm. Journal of Physics: Conference Series, volume 1235, pages 123 – 128. URL: https://doi.org/10.1088/1742-6596/1235/1/012015

4. Nawrin S., Rahman M.R., Akhter S. (2017). Exploreing K-Means with internal validity indexes for data clustering in traffic management system. International Journal of Advanced Computer Science and Applications, volume 8(3), pages 264-272.URL: https://doi.org/10.14569/IJACSA.2017.080337#sthash.jmnh8QU0.dpuf

5. Celebi M.E., Hassan A.K., Patricio A.V. (2017). A comparative study of efficient initialization methods for the k-means clustering algorithm. Expert Systems with Applications, volume 40, pages 200-210. URL: https://doi.org/10.1016/j.eswa.2012.07.021

6. Junjie W., Jian C., Hui X., Ming X. (2008). External validation measures for K-means clustering: A data distribution perspective. Expert Systems with Applications, volume 36, pages 6050-6061. URL: https://doi.org/10.1016/j.eswa.2008.06.093

7. Arbelaitz O., Gurrutxaga I., Muguerza J., Perona I. (2013). An extensive comparative study of cluster validity indices. Pattern Recognition, volume 46, pages 243-256. URL: https://doi.org/10.1016/j.patcog.2012.07.021