

K-Means Clustering on the Wine dataset

Bogdan Yanovich 109072

Aliaksandr Zaman 109163

Warsaw  
07.11.2023

Table of Contents

[1. Introduction 2](#_Toc150103962)

[2. K-Means Clustering 3](#_Toc150103963)

[2.1 Applications of the K-Means Algorithm 3](#_Toc150103964)

[3. K-Means Clustering with Wine dataset 4](#_Toc150103965)

[3.1 Data Collection, Exploration and Preparation 4](#_Toc150103966)

[3.2 Data Preparation 5](#_Toc150103967)

[3.3 Exploration 5](#_Toc150103968)

[3.3.1 Count Plot 6](#_Toc150103969)

[3.3.2 Pie Chart 7](#_Toc150103970)

[3.3.3 Correlation Matrix of Independent Variables 8](#_Toc150103971)

[3.4 Scaling The Features 8](#_Toc150103972)

[4. Analysis and Results 9](#_Toc150103973)

[4.1 Elbow Method (and Silhouette Width) to Determine The Optimal Number of Clusters 9](#_Toc150103974)

[4.2 Confusion Matrix 11](#_Toc150103975)

[5. Model Performance 12](#_Toc150103976)

[5.1 Confusion Matrix for Multi-Class Classification 12](#_Toc150103977)

[5.2 Accuracy, Precision, Recall, F1 Score and Specificity 13](#_Toc150103978)

[6. Conclusion 14](#_Toc150103979)

[7. References 15](#_Toc150103980)

# Introduction

K-Means clustering is of the most widely used unsupervised machine learning techniques. The primary objective of this algorithm is to uncover distinctive groupings within a dataset, where the number of groups is denoted by the variable K.

In this research endeavor, we embarked on a comprehensive analysis using the Wine dataset, which relies on chemical analysis to determine the geographical origin of different wines. The dataset comprises various chemical attributes linked to wines. Our primary objective was to leverage the k-means clustering algorithm to reveal patterns or clusters within the data, shedding light on the potential relationship between these attributes and the geographical origins of the wines. The ensuing sections of this report will delve into the results of our analysis and the clusters identified by the K-means clustering algorithm, offering a comprehensive discussion of our findings.

# K-Means Clustering

K-Means algorithm stands as a cornerstone in machine learning and data mining, at the same time serving as an indispensable clustering algorithm. Clustering, in essence, is a methodology that groups data points into clusters or cohorts based on their inherent similarities. The K-Means algorithm, at its core, partitions data points into a predefined number of clusters.

The K-Means algorithm adopts an iterative approach to cluster data points. It begins by associating data points with randomly selected centroid points and subsequently assigns each data point to the nearest centroid. As the process unfolds, the computed cluster centers are continually updated, refining the clustering arrangement. These steps are reiterated until the clustering process converges to a desirable solution.

K-Means employs a "centroid-based" strategy where each data point aligns itself with the closest centroid, striving to ensure that data points within each cluster exhibit strong similarity, while the dissimilarity between clusters is minimized.

This versatile algorithm finds its application in a lot of domains, including data analysis, image processing, and pattern recognition. Its efficiency and speed makes it a popular choice for addressing clustering challenges. Nevertheless, it's worth noting that K-Means relies on initial random centroid points for data point assignments, leading to variations in results. Hence, in some cases, multiple runs with different initializations may be necessary to achieve optimal outcomes.

## 2.1 Applications of the K-Means Algorithm

* **Customer Segmentation**: K-Means clustering is a valuable tool for segmenting customers based on factors such as purchase history, demographics, or behavioral patterns. This information aids businesses in tailoring their marketing strategies and offerings to cater to distinct customer groups.
* **Image Compression**: K-Means clustering can be harnessed to compress images by reducing the number of colors used. Grouping similar colors together allows the algorithm to represent images using a smaller color palette without compromising visual quality significantly.
* **Anomaly Detection**: The K-Means algorithm is adept at identifying anomalies or outliers within a dataset. By clustering data points, any points that do not belong to any cluster can be considered as anomalies, potentially signifying unusual behaviors or events.
* **Document Clustering**: Document clustering based on content similarity is a key application. This can aid in organizing vast document collections, such as news articles, customer reviews, or academic papers.
* **Recommendation Systems**: K-Means clustering is an asset in recommendation systems, facilitating the grouping of similar items or users. This enables the system to suggest items that are popular among clusters of similar users.
* **Genetic Clustering**: In genetic studies, K-Means clustering helps cluster individuals based on their genetic markers, shedding light on genetic commonalities and differences among individuals. This knowledge proves invaluable in understanding population genetics and disease susceptibility.
* **Network Analysis**: In network analysis, K-Means clustering is a potent tool for identifying communities or clusters of nodes with similar connectivity patterns. This is particularly useful in comprehending the structure and organization of intricate networks, such as social networks or biological networks.
* **Market Segmentation**: K-Means clustering is instrumental in market segmentation. Businesses employ it to segment markets based on attributes like demographics, buying behavior, or preferences, facilitating the identification of target markets and the development of tailored marketing strategies.

# K-Means Clustering with Wine dataset

The Wine dataset, rooted in chemical analysis to determine the origin of wines, is a subject of growing interest and importance. Wine origin classification plays a pivotal role in preserving the authenticity and uniqueness of wines. By detecting the region of origin, we gain insights into the distinct characteristics that make each wine unique.

Wine, celebrated for its rich diversity of flavors and aromas, carries the essence of its terroir, reflecting the geography, climate, and grape varieties of its origin. Unveiling the hidden patterns in this diversity is a quest for experts and researchers alike. Data mining techniques become indispensable in this quest to comprehend the unique attributes associated with each wine region.

Clustering, a powerful and versatile tool, is instrumental in solving this intriguing problem. It empowers us to group wines based on the characteristics derived from chemical analysis, creating clusters that represent different wine regions. In this endeavor, the K-Means algorithm emerges as a straightforward yet effective clustering technique.

K-Means clustering is a semi-parametric approach, making it well-suited for categorizing datasets into K clusters. In the realm of classification methods, we encounter parametric, semi-parametric, and nonparametric approaches. Parametric methods rely on known distributions, nonparametric methods are based on unknown distributions, while semi-parametric approaches bridge the gap by incorporating both known and unknown distributions.

The K-Means algorithm has a fundamental advantage – it can rapidly process large datasets, delivering swift results when the number of clusters is relatively small. This efficiency is particularly advantageous when working with extensive datasets, such as those encountered in wine chemical analysis.

## 3.1 Data Collection, Exploration and Preparation

Obraz zawierający tekst, Strona internetowa, Reklama internetowa, zrzut ekranu

Opis wygenerowany automatycznie

This dataset includes the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wines.

The attributes are the following:

1. **Alcohol**: This attribute measures the alcohol content in the wine, typically represented as a percentage of alcohol by volume. It influences the wine's flavor, body, and overall character.
2. **Malic Acid**: Malic acid is one of the primary organic acids found in grapes. It contributes to the wine's tartness and acidity. The Malic Acid attribute quantifies the amount of malic acid present in the wine.
3. **Ash**: The Ash attribute represents the inorganic mineral content in the wine, including elements like potassium, calcium, and magnesium. This attribute can affect the wine's texture and mouthfeel.
4. **Alcalinity of Ash**: This attribute measures the alkalinity of the inorganic content present in the wine. It is related to the wine's pH level and can impact its taste and stability.
5. **Magnesium**: Magnesium content is the quantity of magnesium present in the wine. Magnesium is an essential mineral and may have subtle effects on the wine's character and taste.
6. **Total Phenols**: Total phenols encompass various chemical compounds in wine, including antioxidants. These compounds play a role in the wine's color, flavor, and potential health benefits.
7. **Flavanoids**: Flavonoids are a subgroup of polyphenolic compounds found in wine. They contribute to the wine's color, taste, and aroma.
8. **Nonflavanoid Phenols**: Nonflavanoid phenols are another group of phenolic compounds present in wine. They have various effects on wine quality, including astringency and bitterness.
9. **Proanthocyanins**: Proanthocyanins are a type of tannin found in wine, contributing to its structure, color, and mouthfeel. They are a subset of flavonoids.
10. **Color Intensity**: This attribute quantifies the depth and richness of the wine's color. It is an important factor in wine appreciation and classification.
11. **Hue**: The hue attribute relates to the color shade of the wine. It is often described in terms of its visual appearance, such as reddish or brownish hues.
12. **OD280/OD315 of Diluted Wines**: This attribute refers to the optical density of the wine at specific wavelengths. It can provide information about the wine's clarity and protein content.
13. **Proline**: Proline is an amino acid present in grapes and wine. It can influence the wine's taste, aroma, and mouthfeel.

The “Class” attribute represents one of the 3 wine cultivars.

## 3.2 Data Preparation

All of the attributes were in “integer” or “numeric” type. We transformed the categorical variable “class”, into a factor variable. This is necessary to ensure proper handling of the categorical data during the clustering process.

## 3.3 Exploration

We’ve decided to explore data using a count plot, a pie chart, and a correlation matrix of independent variables.

### 3.3.1 Count Plot

Obraz zawierający tekst, zrzut ekranu, Prostokąt, diagram

Opis wygenerowany automatycznie

The resulting plot provides a visual representation of the distribution of wine classes in the dataset, allowing us to grasp the balance or imbalance in the number of data points in each class. This can be a valuable information when working on classification or clustering tasks as it helps us understand the dataset's real class distribution.

It's evident that the classes are reasonably well-balanced, with no extreme class imbalances. However, the second class appears to dominate the dataset, boasting the highest number of observations.

### 3.3.2 Pie Chart

Obraz zawierający tekst, diagram, zrzut ekranu, krąg

Opis wygenerowany automatycznie

The pie chart above visually represents the distribution of classes in the Wine dataset, with each slice corresponding to a class and its size proportional to the class's percentage in the dataset. This visualization provides a clear overview of the class distribution and allows us to see the relative proportions of each class at a glance.

The Pie chart confirms the conclusions we drew from the previous chart.

### 3.3.3 Correlation Matrix of Independent Variables

Obraz zawierający tekst, zrzut ekranu, numer, Czcionka

Opis wygenerowany automatycznie

The correlation matrix is a visual representation of the relationships between independent variables in the “df\_clust” dataset. Each cell in the plot contains a number representing the correlation coefficient between the corresponding pair of variables. The color and size of the numbers may be used to convey additional information about the strength and direction of the correlations. This type of visualization helps quickly identify patterns and associations between variables, which can be essential for feature selection, data exploration, and model building.

The Correlation Plot, while informative, may not always provide decisive insights for classification purposes. This is because it can reveal strong positive or negative correlations between attributes that, in the context of classification, might not be as meaningful. For instance, we may observe robust positive correlations between pairs like “Hue” and “OD280/OD315 of Diluted Wines”, “Flavanoids” and “OD280/OD315 of Diluted Wines”, “Total Phenols” and “OD280/OD315 of Diluted Wines”, as well as “Alcohol” and “Color Intensity”, or “Alcohol” and “Proline”. Similarly, notable negative correlations might be detected, such as those between “Malic Acid” and “Hue”, “Color Intensity” and “Hue”, and “Nonflavanoid Phenols” and “OD280/OD315 of Diluted Wines”. While these correlations shed light on associations within the dataset, they may not be the most significant factors when it comes to classifying wines into distinct categories. Classification tasks often require a deeper understanding of the attributes' impact on the target variable, and not all highly correlated attributes are equally relevant in this regard. Hence, further analysis and feature selection are often necessary to pinpoint the attributes that truly contribute to accurate classification.

## 3.4 Scaling The Features

Data scaling serves as an essential preprocessing step in the k-means clustering algorithm. This critical step guarantees that the data is consistently scaled, mitigating the influence of variable variations on the clustering results. By scaling the data, all variables are brought into a common range, thereby promoting greater reliability and precision of the analysis.

To perform data scaling, you can employ the code snippet: *df\_clust\_scale <- scale(df\_clust)*. This snippet utilizes the scale() function to standardize the data, effectively centering and scaling it. This transformation results in variables with a mean of 0 and a standard deviation of 1.

Scaling the data significantly enhances the effectiveness of the k-means algorithm in clustering the data. Since k-means relies on distances between data points, the scale and distribution of variables can exert a profound impact on the results. Scaling guarantees that variables are consistently scaled, fostering equitable and unbiased clustering outcomes.

Subsequently, the scaled data frame, denoted as “df\_clust\_scale”, will be utilized in the subsequent k-means clustering analysis, ensuring that the data is prepared for accurate and meaningful clustering insights.

# Analysis and Results

## 4.1 Elbow Method (and Silhouette Width) to Determine The Optimal Number of Clusters

The elbow method is a valuable technique employed to ascertain the ideal number of clusters within a dataset when using clustering algorithms like k-means. This method revolves around the assessment of the within-cluster sum of squares (WSS) as a function of the number of clusters.

To put this method into practice, we employ the following line of code: *fviz\_nbclust(df\_clust\_scale,kmeans,method="wss") + geom\_vline(xintercept = 3, linetype = 2) + labs(subtitle = "Elbow Method N1")*. Here, we use the "WSS" method, which calculates the sum of squared distances of data points within each cluster, to guide our choice of the optimal number of clusters for our k-means model. The “fviz\_nbclust(…)” function visualizes the results by plotting the number of clusters against their respective WSS values.

The Elbow method is instrumental in pinpointing the number of clusters that strikes a balance between minimizing the within-cluster sum of squares and avoiding excessive fragmentation. The graph below, generated by “fviz\_nbclust(…)”, presents the WSS values for varying cluster numbers. Notably, the "elbow" or inflection point in the plot signifies the optimal number of clusters, indicating a substantial reduction in WSS followed by a tapering improvement rate. This serves as a critical guidepost for cluster selection in our analysis.

Obraz zawierający tekst, linia, diagram, Wykres

Opis wygenerowany automatycznie

Another approach to determine the optimal number of clusters within a dataset is through the average silhouette width. In contrast to the elbow method, which primarily focuses on within-cluster sum of squares (WSS), the average silhouette width assesses the quality of clustering by considering both the cohesion within clusters and the separation between clusters.

Similarly, the code *fviz\_nbclust(df\_new\_scale, kmeans, method="silhouette") + labs(subtitle = "Silhouette Method N2")* uses the silhouette method which quantifies the appropriateness of each data point's placement within its assigned cluster, with values spanning from -1 to 1. Higher Silhouette coefficients indicate more suitable cluster assignments. The “fviz\_nbclust(…)” function creates a plot that correlates the number of clusters with their respective Silhouette coefficients, thus offering a visual representation of the Silhouette values for varying cluster quantities.

Determining the optimal number of clusters entails identifying the cluster count associated with the highest average Silhouette coefficient, signifying the most effective configuration for clustering in the given dataset.

A graph with blue lines and numbers

Description automatically generated

After visualizing the WSS values and Silhouette coefficients across various cluster numbers, we can discern the most appropriate number of clusters that faithfully captures the inherent structure within the dataset. In our specific analysis, it becomes evident that the optimal number of clusters is 3, offering a cohesive representation of the data's underlying patterns.

## 4.2 Confusion Matrix

Based on the analysis performed using the k-means clustering algorithm on the Wine dataset, three distinct clusters (Cluster 1, Cluster 2, and Cluster 3) were obtained. Each cluster represents wines from different regions of origin.

Cluster 1 consists of a total of 62 data points, out of which 59 belong to "Region A," 3 belong to "Region B," and 0 to "Region C."

Cluster 2 comprises 65 data points, with 0 wines from "Region A," 65 from "Region B," and 0 from "Region C."

Cluster 3 encompasses 51 data points, with 0 wines from "Region A," 3 from "Region B," and 48 from "Region C."

The k-means clustering algorithm successfully grouped the wine data into three distinct clusters. These results are valuable for wine classification and understanding the characteristics and origins of wines in each cluster.

Obraz zawierający tekst, Czcionka, zrzut ekranu, numer

Opis wygenerowany automatycznie

# Model Performance

## 5.1 Confusion Matrix for Multi-Class Classification

K-means clustering is an unsupervised learning algorithm which does not require a dependent variable in its clustering process. However, we decided to choose a dataset which has a dependent variable. This was mainly done in order to compare the clusters obtained from the K-means clustering algorithm with the real classes of wines. It will enable us to calculate descriptive statistics such as accuracy, precision, recall and other to try to measure the prediction power of the K-means clustering algorithm.

When it comes to evaluating the performance of classification models, particularly in multi-class scenarios, the calculation of True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) plays a crucial role. These metrics provide a comprehensive understanding of a model's accuracy and effectiveness in correctly classifying instances among multiple classes. However, it's important to note that the formulas for these metrics differ in multi-class confusion matrices compared to binary classification.

In binary classification, where there are only two possible classes (1 or 0, True of False etc.), the calculations are relatively straightforward. TP represents the instances correctly classified as the positive class, TN stands for the instances correctly classified as the negative class, FP signifies the instances incorrectly classified as positive when they are actually negative, and FN accounts for the instances incorrectly classified as negative when they are actually positive.

In multi-class classification, the complexity arises from dealing with more than two classes. The calculation of TP, TN, FP, and FN becomes a bit more intricate. In this context, TP refers to the instances correctly classified within a specific class, TN signifies instances correctly classified as not belonging to any class, FP represents instances incorrectly classified within the class of interest, and FN accounts for instances incorrectly classified as not belonging to the class of interest.

Example:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Class 1 | Class 2 | Class 3 |
| Class 1 | Cell 1 | Cell 2 | Cell 3 |
| Class 2 | Cell 4 | Cell 5 | Cell 6 |
| Class 3 | Cell 7 | Cell 8 | Cell 9 |

We calculated TP, FN, FP and TN the following way:  
 TP = intersection of row and column of some class  
 FN = sum of values of the class row except the TP value  
 FP = sum of values of the class column except the TP value  
 TN = sum of values of the values of all columns and rows except the values of that class that we are calculating the values for.

For example TP, FN, FP and TN for class 1 will be the following:  
 TP = Cell 1  
 FN = Cell 2 + Cell 3  
 FP = Cell 4 + Cell 7  
 TN = Cell 5 + Cell 6 + Cell 8 + Cell 9

## 5.2 Accuracy, Precision, Recall, F1 Score and Specificity

Obraz zawierający tekst, zrzut ekranu, Czcionka, linia

Opis wygenerowany automatycznie

*Accuracy* is a widely used metric for assessing the overall performance of a classification model. You can calculate accuracy by dividing the number of correct predictions by the total number of predictions. In our analysis, our model achieved an accuracy of 0.98, 0.97, 0.98 for “Class 1”, “Class 2”, “Class 3” respectively, indicating its effectiveness in correctly predicting class labels for the data. An accuracy of 0.98 for classes 1 and 3 and 0.97 for class 2 imply that the model achieved a high rate of correct predictions, accurately assigning data points to their respective classes in approximately 98%, 97%, 98% of cases.

A high accuracy value suggests that the model's predictions closely align with the true class labels. Nevertheless, it's essential to interpret accuracy in the context of the specific problem and data characteristics. Factors like class imbalances, the presence of outliers, or the application's unique requirements may influence the significance of the reported accuracy.

To ensure the reliability of the reported accuracy, it is imperative to evaluate the model's performance using additional metrics such as precision, recall, F1 score and sensitivity.

*Precision* is a metric that measures how often a machine learning model correctly predicts the positive class. You can calculate precision by dividing the number of correct positive predictions (true positives) by the total number of instances the model predicted as positive (both true and false positives). A higher precision indicates a lower false-positive rate, demonstrating that the model is more accurate at classifying wines of that class.

*Recall*, also known as the true positive rate, is a metric that measures how often a machine learning model correctly identifies positive instances (true positives) from all the actual positive samples in the dataset. You can calculate recall by dividing the number of true positives by the number of positive instances. The latter includes true positives (successfully identified cases) and false negative results (missed cases). A higher recall indicates a lower rate of false negatives, indicating that the model is better at identifying and capturing wines of that class.

The *F1 score* represents the harmonic mean of accuracy and recall. It provides a balanced evaluation of the model's performance, taking both precision and recall into consideration. It accounts for both false positives and false negatives. The F1 score is helpful for achieving a balance between precision and recall because it gives both metrics equal weight. A higher F1 score indicates improved overall performance in identifying true positives while minimizing false positives and false negatives.

*Specificity*, often referred to as the true negative rate, is a pivotal performance measure that assesses the model's precision in correctly identifying and classifying instances that do not belong to a specific class or category. It serves as a key indicator of the model's ability to exclude data points that are unrelated to a particular class.

In our analysis of the Wine dataset, the results indicate that the model demonstrates high precision, effectively capturing a significant proportion of true wines in each class while maintaining a relatively low number of false positives. With an excellent level of accuracy, the model excels at making precise predictions for wines from various classes. Moreover, the model's impressive recall indicates that it effectively identifies and captures the majority of wines in each class, minimizing false negatives. In addition to this, the model exhibits exceptional specificity, ensuring that it accurately excludes wines that do not belong to specific classes. The good F1 score underscores the model's strong overall ability to predict wines from different classes, demonstrating its reliability and effectiveness in classification tasks.

# Conclusion

In conclusion, our clustering analysis of the Wine dataset has yielded highly promising results. The dataset, comprising three distinct wine cultivars, was effectively clustered into three groups, each representing a unique class. The metrics used to assess the quality of our clustering model, including precision, recall, and F1 score, all indicate exceptional performance, underscoring the model's accuracy and reliability in classifying wines into their respective cultivars. Moreover, the model's high specificity emphasizes its precision in excluding wines that do not belong to specific classes.

One of the noteworthy achievements of this clustering analysis is the robustness of the clustering results. Visualizations of the clustering show that the vast majority of observations have been accurately assigned to their corresponding clusters, with only a very small number (a mere six observations) being misclassified. This visual representation demonstrates the high quality of the clustering, with a clear and distinct separation of wines into their respective classes. These results are not only statistically significant but also highly interpretable, providing valuable insights for various applications, including wine quality control and origin determination. Nevertheless, it’s worth to mention that our dataset contains only 178 observations, which can be limiting is case we try to extrapolate our results on a bigger dataset.

Obraz zawierający tekst, diagram, mapa

Opis wygenerowany automatycznie

In conclusion, our clustering analysis of the Wine dataset has successfully partitioned the data into three distinct clusters, aligning closely with the true wine cultivars. The impressive metrics and the minimal misclassification rate in our visualization indicate the robustness and effectiveness of our clustering model, making it a valuable tool for wine classification and related tasks.

# References

Wine dataset, UC Machine Learning Repository, 1991. The dataset is available via the following link: <https://archive.ics.uci.edu/dataset/109/wine>

Imad Dabbura, ‘*K-means Clustering: Algorithm, Applications, Evaluation Methods, and Drawbacks’*, Towards Data Science, 2018. The article is available via the following link: <https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a>

Bharathi, ‘*Latest Guide on Confusion Matrix for Multi-Class Classification’*, Analytics Vidhya, 2023. The article is available via the following link: [https://www.analyticsvidhya.com  
/blog/2021/06/confusion-matrix-for-multi-class-classification/](https://www.analyticsvidhya.com/blog/2021/06/confusion-matrix-for-multi-class-classification/)

Chris Piech, *‘K means Clustering Algorithm’*, Stanford University, 2013. The article is available via the following link: <https://stanford.edu/~cpiech/cs221/handouts/kmeans.html>

K. Harrison, *‘Accuracy, precision, and recall in multi-class classification’*, The article is available via the following link: <https://www.evidentlyai.com/classification-metrics/multi-class-metrics>