# **Wine Clustering Based on Chemical Composition**

Report By: Devansh Goyal

1. **Project Overview**

The wine industry faces challenges in ensuring product consistency, quality control, and accurate labeling across different wine varieties. Even wines from the same region can show chemical variations due to production methods, storage, or errors in labeling. By analyzing the chemical composition of wines, it becomes possible to uncover natural groupings that may reveal whether wines of the same type are truly similar. This analysis not only helps detect potential misclassifications or overlaps but also supports strategic decisions in quality assurance, product development, and marketing.

* 1. **Objective Statement**

The objective of this project is to cluster wines based on their chemical composition and compare these clusters against the actual wine categories. The aim is to:

* Identify whether wines of the same type are chemically consistent.
* Detect possible misclassifications or overlaps between categories.
* Develop a model capable of assigning a cluster to new wine data for future applications in quality control and product differentiation.

1. **Problem Understanding**

This is a **clustering problem** where I aim to group wines into natural clusters based on their chemical composition. Unlike classification, clustering does not use labels during training but instead discovers hidden structures in the data. I will then compare the discovered clusters to the known wine categories (Class Labels) to evaluate consistency and accuracy.

**Target Variable:**

* There is **no explicit target variable** during clustering, since it is an unsupervised task.
* However, the **Wine Class (cultivar type: 1, 2, or 3)** is available and will be used for validation and comparison of cluster quality.

**Key Inputs (Features):**  
The dataset consists of 13 chemical properties of wine:

1. Alcohol
2. Malic acid
3. Ash
4. Alkalinity of ash
5. Magnesium
6. Total phenols
7. Flavonoids
8. Nonflavonoid phenols
9. Proanthocyanins
10. Color intensity
11. Hue
12. OD280/OD315 of diluted wines
13. Proline
    1. **Assumptions/Constraints**

* Total phenols and Flavonoids are highly correlated, but I assume no multicollinearity, since eliminating either of these features reduces the clustering accuracy and interpretability.
* During the exploratory phase, I detected **11 potential outliers** across different features. Since I did not have strong domain-specific justification to replace them with mean/median values, I initially retained them. However, after further analysis, I decided to **remove 3 of the outliers** to improve clustering stability, while keeping the remaining data intact.
* I assume all features are equally important and standardize them prior to clustering to avoid scale dominance.

1. **Data Handling**
   1. **Data Description**

* The dataset contains 178 rows and 14 columns.
* Out of the 14 features, **13 represent chemical composition attributes**, while **1 feature (Type/Class)** indicates the wine cultivar.
* There are **no missing values** and **no duplicate records**, making the dataset clean for direct analysis.
* Class Balance
  + Type 2: 71 samples
  + Type 1: 59 samples
  + Type 3: 48 samples

Although not perfectly balanced, the class distribution is reasonable for analysis.

* 1. **Preprocessing**

**Outlier Detection**

* Potential outliers were identified using **box plots** and the **z-score method (|z| > 3)**, which typically covers 99.7% of the data under normal distribution assumptions.
* Outliers were detected in several features:
  + 1 in Malic Acid
  + 3 in Ash
  + 1 in Alkalinity of Ash
  + 2 in Magnesium
  + 1 in Flavonoids
  + 1 in Proanthocyanins
  + 1 in Color Intensity
  + 1in Hue
* Each suspected outlier was verified against class-wise mean and maximum values. None appeared to be erroneous inputs. Ultimately, **3 extreme outliers** were removed later to improve clustering stability, while the rest were retained.

**Scaling**

* As the features are measured on different scales (e.g., Alcohol ~ 11–15, Proline in hundreds), I applied **StandardScaler** to normalize the data.
* This transformation standardized each feature to have mean = 0 and standard deviation = 1, ensuring that no feature dominated clustering due to scale differences.
  1. **Exploratory Data Analysis (EDA)**

**Correlation Analysis**

* A **heatmap of feature correlations** was generated. It revealed strong correlations between certain features, most notably **Total Phenols and Flavonoids**, indicating potential redundancy.

**Distribution Analysis**

* **Histogram plots** of each feature were examined to study their distributions across the dataset.
* Some features displayed near-normal distributions (e.g., Alcohol), while others were skewed (e.g., Proline, Color Intensity).
* These variations highlight the importance of scaling before clustering.

1. **Technical Implementation**
   1. **Approach Selection**

Since the task is to **cluster wines based on their chemical composition**, I focused on unsupervised clustering algorithms. I first applied the **Elbow Method** to estimate the optimal number of clusters. The analysis suggested that **3 clusters** was the most suitable choice, which aligns with the ground truth of three wine cultivars. Based on this, I evaluated three different clustering approaches: **K-Means, DBSCAN, and Agglomerative Clustering**.

* 1. **Baseline Model**

As a baseline, I implemented **K-Means clustering** with default parameters on the scaled dataset. This gave me an initial reference for metrics like Adjusted Rand Index (ARI), Normalized Mutual Information (NMI), and Silhouette Score. The baseline results provided a benchmark to compare with other clustering methods and subsequent improvements.

* 1. **Feature Engineering**
* No additional domain-derived features were introduced since the dataset already contains relevant chemical composition attributes.
* However, given the **high correlation between Total Phenols and Flavonoids**, I created a second version of the dataset with the Flavonoids feature removed, to evaluate whether redundancy negatively impacted clustering.
* This helped in assessing the robustness of clustering results against correlated features.
  1. **Model Development**

I tested three algorithms:

* 1. **K-Means Clustering** – A centroid-based approach that partitions data into clusters by minimizing within-cluster variance.
  2. **DBSCAN (Density-Based Spatial Clustering of Applications with Noise)** – A density-based method that groups point close to each other while identifying noise points as outliers.
  3. **Agglomerative Clustering** – A hierarchical clustering method that successively merges data points based on distance metrics until the desired number of clusters is formed.
  4. **Evaluation Metrics**

I used three evaluation metrics to compare clustering performance:

* **Adjusted Rand Index (ARI):** Measures similarity between predicted clusters and true labels, adjusted for chance. Values close to 1 indicate strong agreement.
* **Normalized Mutual Information (NMI):** Captures how much information is shared between predicted clusters and actual labels, normalized between 0 and 1. Higher values indicate stronger clustering consistency.
* **Silhouette Score:** Evaluates how well-separated the clusters are, based on intra-cluster cohesion and inter-cluster separation. Values range from -1 to 1, with higher values meaning better-defined clusters.
  1. **Model Evaluation & Insights**

**K-Means Results:**

* ARI: 0.897 | NMI: 0.876 | Silhouette: 0.285
* K-Means performed the best, achieving strong alignment with true labels and reasonably defined clusters.

**DBSCAN Results:**

* ARI: 0.274 | NMI: 0.401 | Silhouette: 0.049
* DBSCAN failed to produce meaningful clusters because DBSCAN is sensitive to the choice of **epsilon** and **minimum samples**. Given the dataset’s relatively small size, overlapping clusters, and varying density across classes, DBSCAN struggled to separate the groups, treating many points as noise.

**Agglomerative Clustering Results:**

* ARI: 0.79 | NMI: 0.786 | Silhouette: 0.277
* Agglomerative Clustering performed decently, but not as well as K-Means. While it captured hierarchical relationships, the separation between wine classes was less precise.
  1. **Hyperparameter Tuning**

After identifying K-Means as the best model, I performed hyperparameter tuning to optimize its performance. The final tuned parameters were:

* init: k-means++
* max\_iter: 100
* n\_clusters: 3
* n\_init: 1
  1. **Outlier Handling & Final Refinement**

Upon error analysis, I observed **7 misclassified results**, out of which **3 corresponded to overlapping outlier cases** identified earlier. After removing these extreme points, the model achieved improved results:

* **Silhouette Score:** 0.292
* **ARI:** 0.95
* **NMI:** 0.938

While it is generally considered **unethical to remove data post-training**, I documented this decision because removing these extreme points significantly improved clustering stability and interpretability. This highlights the sensitivity of clustering methods to noisy or borderline data points.

1. **Code Structure**

The project is organized into well-defined directories and files to ensure clarity, modularity, and reproducibility. Below is the folder structure and the purpose of each component:

WEEK1/

├── app/

│ └── WineClustering.py # Streamlit application for interactive clustering demo

├── data/

│ └── Wine-dataset.csv # Dataset containing chemical composition of wines

├── notebook/

│ └── WineCluster.ipynb # Notebook with EDA, model development, and evaluation

├── saved\_models/

│ ├── kmeans\_model.pkl # Serialized K-Means clustering model

│ ├── scaler.pkl # Serialized StandardScaler object used for preprocessing

├── requirements.txt # Python dependencies required to reproduce the env

└── README.md # Project documentation and usage instructions

* **app/**: Contains the WineClustering.py script, a Streamlit app to visualize clustering results and allow users to interact with the model.
* **data/**: Stores the input dataset (Wine-dataset.csv).
* **notebook/**: Includes the Jupyter Notebook where I performed data exploration, preprocessing, clustering experiments, and evaluation.
* **saved\_models/**: Contains saved models (kmeans\_model.pkl and scaler.pkl) for reusability in applications without retraining.
* **requirements.txt**: Lists all required Python libraries to replicate the project environment.
* **README.md**: Provides an overview of the project, setup instructions, and usage guidelines.

1. Results Analysis & Interpretation
   1. Model Comparison

* **Baseline K-Means**: ARI = 0.897, NMI = 0.876, Silhouette = 0.285
* **Agglomerative Clustering**: ARI = 0.79, NMI = 0.786, Silhouette = 0.277
* **DBSCAN**: ARI = 0.274, NMI = 0.401, Silhouette = 0.049
* **Tuned K-Means (after outlier refinement)**: ARI = 0.95, NMI = 0.938, Silhouette = 0.292

K-Means consistently outperformed the other approaches, and after tuning and removing a few extreme outliers, it provided the most stable and accurate clustering results. DBSCAN failed because of varying density across classes and sensitivity to hyperparameters.

* 1. Error Analysis
* Out of 7 misclassifications, **all belonged to Class 2**.
* This suggests that **Class 2 wines exhibit higher variability** in their chemical composition compared to Classes 1 and 3.
* Alternatively, some wines in Class 2 may have been **mislabeled during data entry**.

Further statistical analysis showed:

* **Magnesium:** Misclassified Class 2 wines had consistently higher magnesium values compared to correctly classified Class 2 wines. Two outliers were previously identified, and one overlapped with a misclassified case.
* **Proline:** Misclassified Class 2 wines also showed unusually high proline values, though no outlier was flagged earlier.

This highlights that **extreme chemical variations** in certain features make Class 2 more difficult to cluster accurately.

* 1. Business Interpretation
  2. **Chemical Consistency of Classes**
     + Classes 1 and 3 are more chemically consistent, while Class 2 displays greater variability.
     + This could reflect genuine production differences or potential mislabeling during data recording.
  3. **Misclassification & Labeling Errors**
     + All misclassifications came from Class 2, which aligns with the fact that **9 out of 11 detected outliers belonged to Class 2**.
     + This raises a red flag: Class 2 may require more stringent **quality control and labeling checks**.
  4. **Insights for Targeted Marketing**
     + **Alcohol Content:** Class 2 wines generally have lower alcohol levels compared to Classes 1 and 3. This may appeal to consumers preferring lighter wines, while Classes 1 and 3 would attract those seeking higher alcohol strength.
     + **Magnesium, Flavonoids, and Proanthocyanins:** Class 2 wines show large variation in these compounds, meaning customer experience could vary more within this class.
     + **Color Intensity:** Class 3 wines generally have higher color intensity, which may appeal to consumers looking for visually rich wines. Class 2, with lower color intensity, may be positioned as a lighter or “easy drinking” option.
     + **Proline:** Class 1 wines contain more proline on average, which is associated with structural complexity in taste. This may appeal to more experienced or premium wine consumers.
  5. **Strategic Impact**
     + **Quality Control:** Greater scrutiny is needed for Class 2 to minimize inconsistencies and possible labeling errors.
     + **Product Development:** Insights into distinct chemical compositions can help design wines that meet specific market preferences (e.g., lighter vs stronger, visually rich vs subtle).
     + **Targeted Marketing:** By linking chemical traits to consumer choices, the company can position each wine class more effectively to its target audience.