Wine Quality Prediction: A Data-Driven Approach

Master Degree in Computer Science, Universidade NOVA de Lisboa
Data Analytics and Mining project, part 1
Academic Year: 2024-2025
Group 10

Alberto Dicembre
a.dicembre@campus.fct.unl.pt
Roll number: 72624

Ricardo Rodrigues
rf.rodrigues@campus.fct.unl.pt
Roll number: 72054

1 Introduction

1.1 Motivation for Choosing the Dataset

Wine quality assessment is a crucial aspect of the wine industry, influencing both consumer satisfaction and commercial success. The dataset related to Portuguese wine (Red and White) offers relationships between physicochemical properties and sensory-based quality evaluation. By leveraging this dataset, we aim to develop a predictive model that can classify or estimate wine quality based on measurable attributes. This analysis is valuable for wine producers, sommeliers, and retailers to improve product consistency and optimize production methods.

1.2 Descriptive Information of the Dataset

The dataset comprises two subsets representing red and white variants of wine from northern Portugal. The inputs include objective tests (e.g. PH values) and the output is based on sensory data (median of at least 3 evaluations made by wine experts). Each expert graded the wine quality between 0 (very bad) and 10 (very excellent). The variables are:

- **Number of Instances:** 6497 (1599 in the Red Wine dataset, 4898 in the White Wine dataset)
- **Number of Features:** 11
- **Target Variable:** Wine quality, rated on a scale from 0 to 10
- **Link for consultation:** Wine Quality - UCI Machine Learning Repository

The features are all **real-numbered**, except for the target value which is an integer score from 1 to 10.

1.3 Feature Description

The input features are:

- **Fixed Acidity:** Primarily tartaric and malic acids that remain constant during fermentation. These contribute to the wine's sour taste and are essential for stability and aging.
- **Volatile Acidity:** Refers mainly to acetic acid (vinegar-like smell). Low levels are normal, but high concentrations can indicate spoilage or poor fermentation control.
- **Citric Acid:** A minor acid in wine, it adds freshness and a citrus note. Winemakers may add it to enhance acidity and flavor balance.
- **Residual Sugar:** The sugar left after fermentation. It determines the sweetness of the wine. Dry wines have low residual sugar, while sweet wines have more.
- **Chlorides:** Represents the salt content in wine. High levels may give a salty taste and are often considered a fault, usually due to contamination or water quality.
- **Free Sulfur Dioxide:** Acts as an antimicrobial and antioxidant. It protects wine from spoilage and oxidation but must be balanced to avoid affecting flavor.
- **Total Sulfur Dioxide:** Sum of free and bound forms. Excess can affect aroma and safety, while insufficient amounts risk spoilage.
- **Density:** Relates to the sugar and alcohol content. Higher density usually means higher sugar or lower alcohol, which can help monitor fermentation progress.
- **pH:** Measures acidity. Lower pH (more acidic) helps preserve wine and enhances freshness, while higher pH can lead to instability and dull flavor.
- **Sulphates:** Contribute to the preservation and flavor of the wine. Higher levels can enhance the wine's body and bitterness but need careful control.
- **Alcohol:** Result of fermentation. It affects the body, warmth, and mouthfeel of the wine. Typically measured as a percentage by volume.

Note that, as reported in the dataset description, due to privacy and logistic issues, some features are missing, such as the type of grapes and the selling price of the wines. Also, there is no information on the unit measure of the feature values.

1.4 Problem Domain and Proposed Exploration

Wine quality is determined by multiple factors, including acidity, sugar levels, and alcohol content. The challenge lies in modeling these relationships effectively for prediction and classification tasks. Several potential research problems can be addressed:

1. **Predicting Wine Quality:** Regression models can be used to estimate the wine's quality based on physicochemical properties.

2. **Classifying Wine into Quality Categories:** Machine learning classification techniques can categorize wines as high, medium, or low quality.

3. **Feature Selection for Quality Assessment:** Identifying the most influential attributes can help improve wine production quality control.

4. **Outlier Detection in Wine Quality:** Identifying exceptional or defective wines based on physicochemical anomalies.

We propose to explore **Predicting Wine Quality**, as it has significant practical implications. A reliable predictive model can assist wine producers in understanding the impact of chemical compositions on sensory evaluation, ultimately leading to process optimization and better-quality wines.

2 Experimental study

2.1 Exploratory Data analysis

The dataset is divided in two partitions: one related to red wine, one related to white wine. It was decided to perform analyses both on the two separated versions and one one that includes both.

- **Duplicates**: the *Red Wine* Dataset contains 240 duplicate rows, and the *White Wine* contains 937.

- **Missing values**: The datasets do not contain missing values, therefore there's no need to address that problem.

The dataset, both for red and white wine, is generally well-behaved. Most features show distributions centered around their medians with only minor skewness and limited outliers. The main exceptions are:

- **Citric Acid** in the red wine dataset – this feature shows a concentration of values at the origin (0), resulting in a slightly negatively skewed distribution.

- **Residual Sugar** in the white wine dataset – similarly concentrated near zero, with noticeable outliers.

2.2 Feature Correlations

Figures 1, 2 and show the correlation heatmaps for the Red and White Wine dataset respectively.

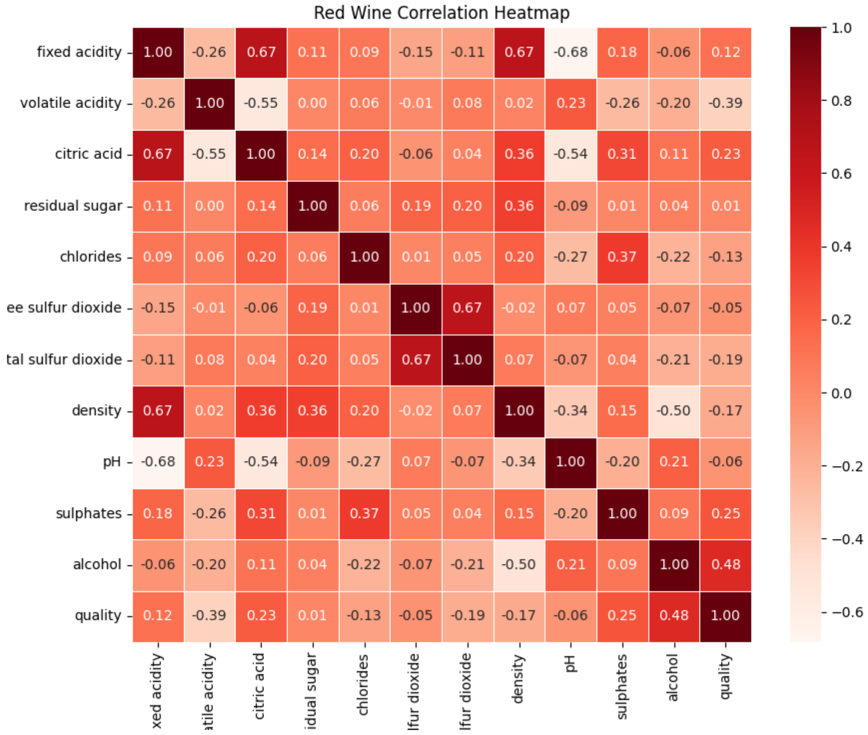

Figure 1: Red Wine Dataset Correlation Heatmap

In both **Red** and **White Wine** datasets, **Density** shows a negative correlation with **Quality**, though this relationship is more pronounced in the **White Wine** dataset (**-0.31**) compared to the **Red Wine** dataset (**-0.11**). This suggests that higher density is associated with lower quality in both wine types, but the effect is stronger in **White Wine**.

Residual Sugar is a significant predictor of high **Density** in both datasets, though it has a stronger relationship with **Density** in the **White Wine** dataset (**0.84**) compared to the **Red Wine** dataset (**0.36**). This indicates that **Residual Sugar** plays a crucial role in determining **Density** in **White Wine**, contributing more to lower quality.

In **Red Wine**, **Fixed Acidity** is the most important predictor of high **Density** (with a strong correlation of **0.67**). However, high **Density** in **Red Wine** is still associated with lower **Quality**, though the correlation is weaker compared to **White Wine**. Despite **Fixed Acidity's** strong correlation with **Density**, it has no significant correlation with **Quality**, which suggests it might not directly influence the overall quality of **Red Wine**, but could become more relevant when merging both datasets.

For **Sulfur Dioxide** (both **Free** and **Total**), there is a marked difference between **Red** and **White** wines. In **Red Wine**, **Free Sulfur Dioxide** and **Total Sulfur Dioxide** show minimal correlations with **Density**, making them less useful predictors. However, in **White Wine**, **Total Sulfur Dioxide** and **Free Sulfur Dioxide** are heavily correlated with **Residual Sugar** (**0.40** and **0.30**, respectively) and **Density** (**0.53** and **0.29**, respectively), and they are also correlated with each other (**0.62**). Therefore, both sulfur dioxide types are useful predictors for **Density**, which in turn is associated with low-quality **White Wine**. **Total**

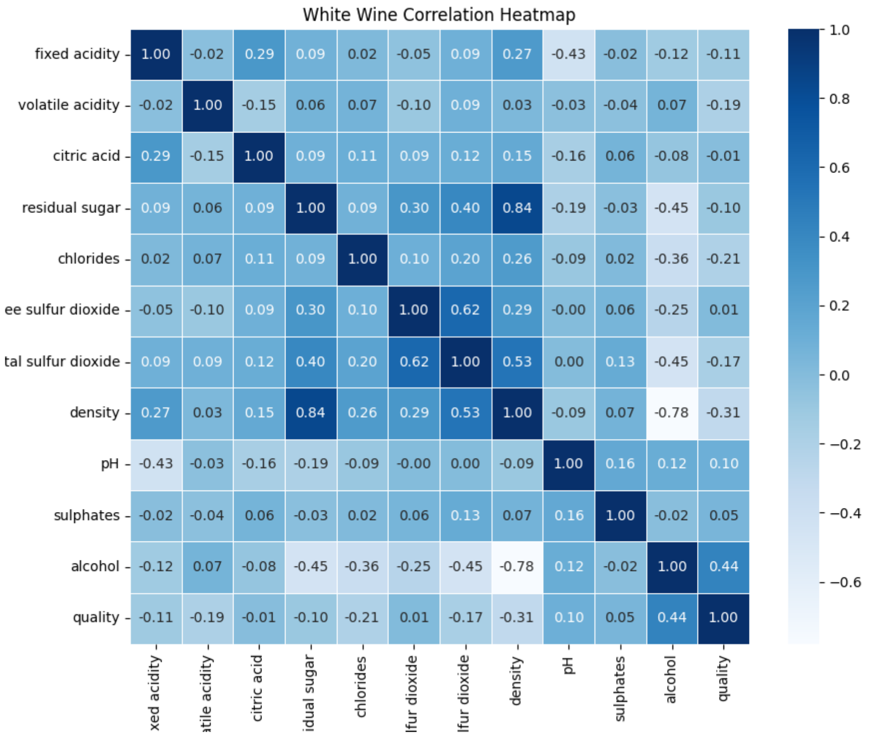

Figure 2: White Wine Dataset Correlation Heatmap

Sulfur Dioxide has stronger correlations with **Density** and **Residual Sugar** compared to **Free Sulfur Dioxide**, making it a slightly more reliable predictor in **White Wine**.

Alcohol is a strong positive predictor of **Quality** in both datasets. In **Red Wine**, **Alcohol** has a correlation of **0.48** with **Quality**, and in **White Wine**, it has a similar correlation of **0.44**. This suggests that wines with higher alcohol content tend to be of better quality in both types. Additionally, in **Red Wine**, **Sulphates** (**0.25**) and **Citric Acid** (**0.23**) are also useful predictors for **Quality**. In contrast, in **White Wine**, only **Alcohol** shows a meaningful relationship with **Quality**, followed by a small positive correlation with **pH** (**0.10**) and **Sulphates** (**0.05**). However, **Sulphates** does have a moderate correlation with **pH** (**0.16**), so one could argue that **Sulphates** is a marginally relevant predictor for **Quality** in both datasets.

2.2.1 Feature Distribution

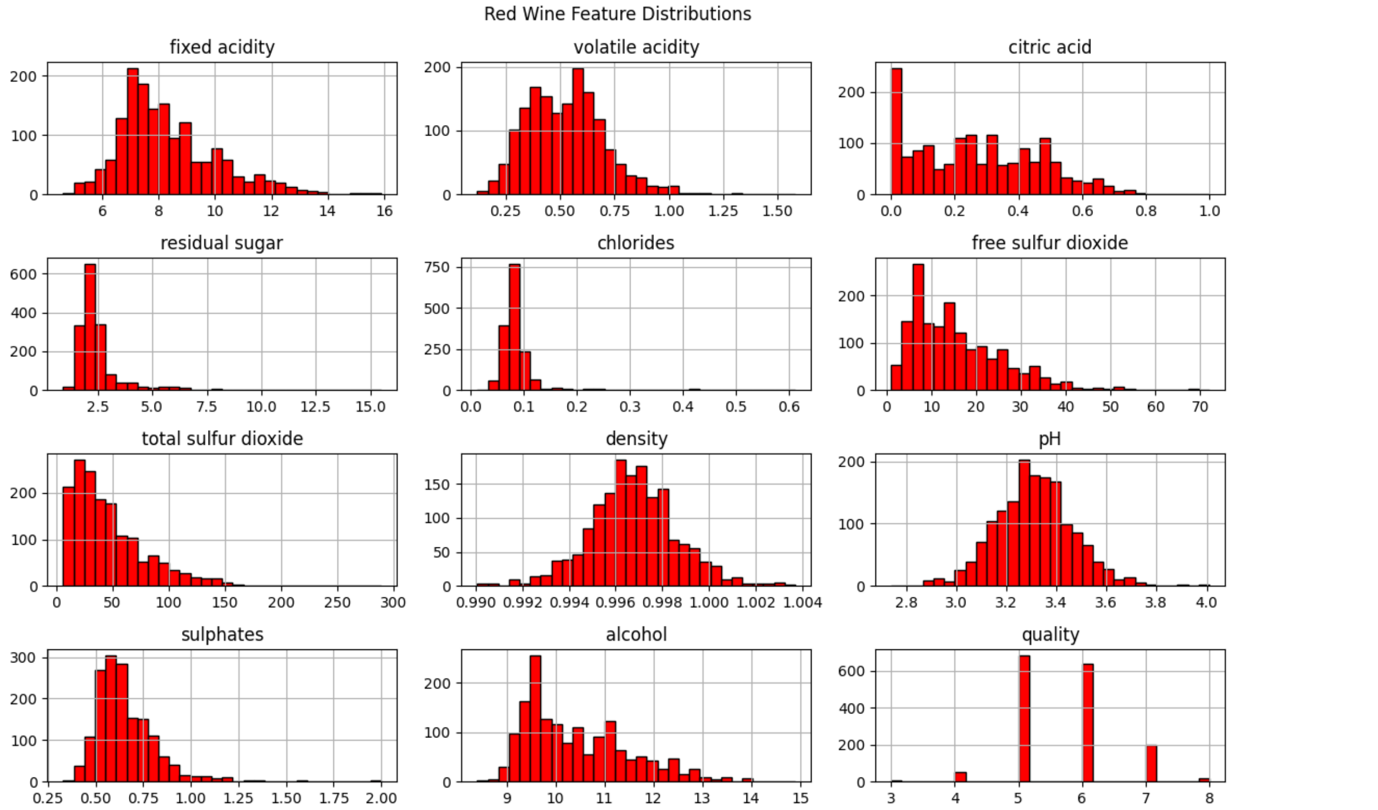

Figure 3: Histograms for the feature distribution of the Red Wine dataset

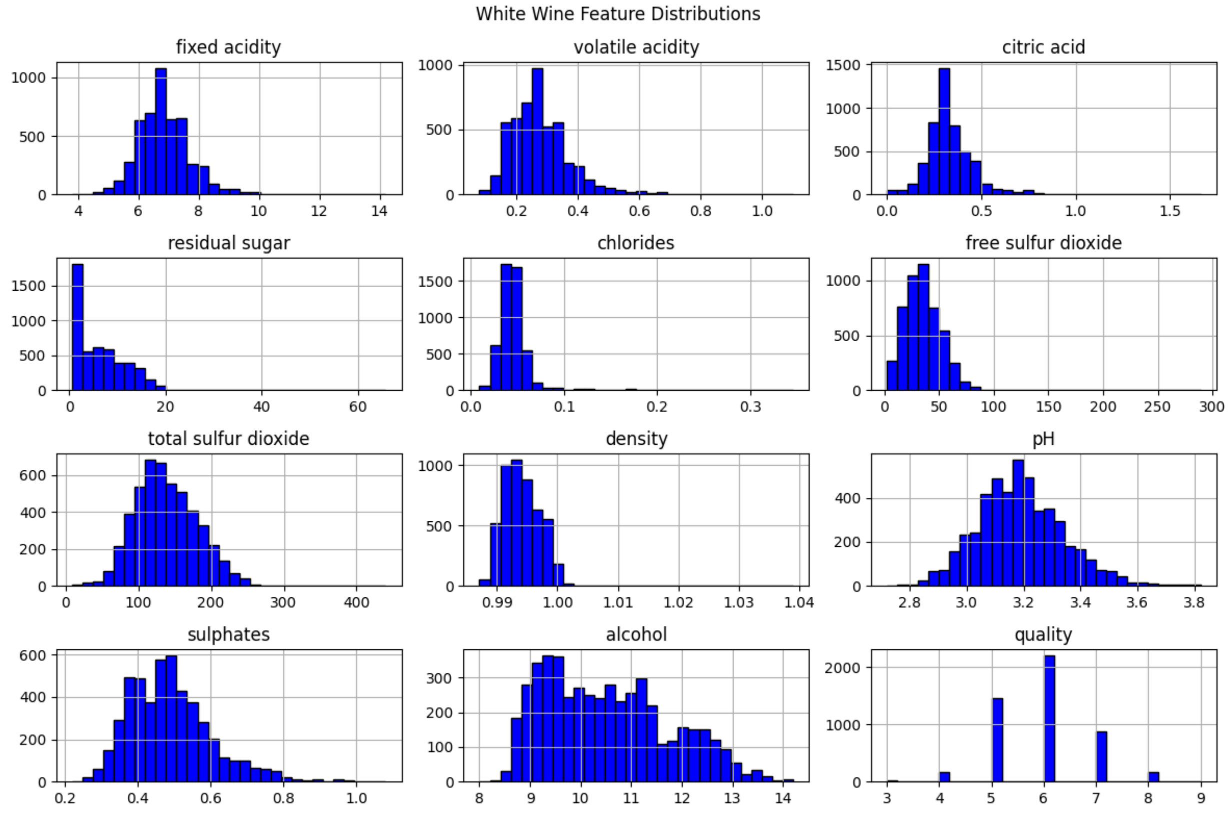

Figure 4: Histograms for the feature distribution of the White Wine dataset

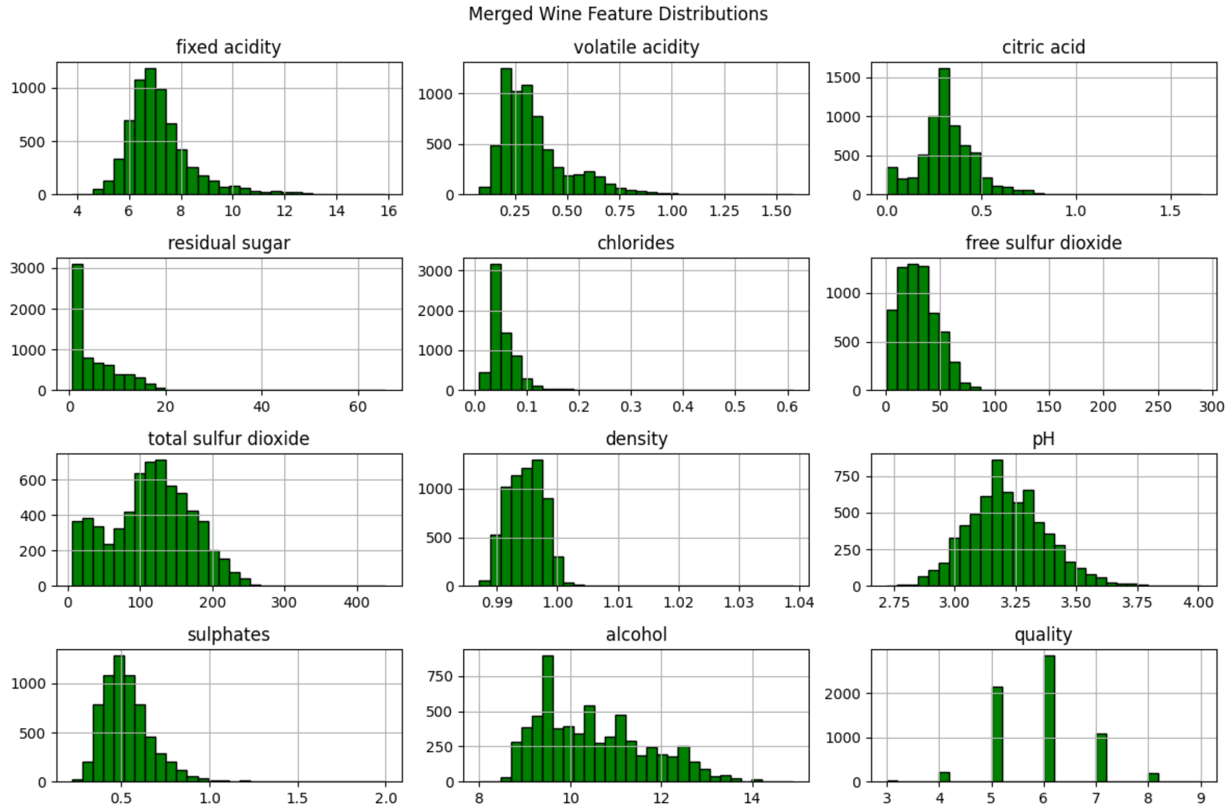

Figure 5: Histograms for the feature distribution of the Merged dataset

By looking at the histograms for the distribution of the features of the datasets (Figures 3, 4, 5), we can notice that neither of the three datasets is balanced, as the *Quality* label distribution is not homogeneous. In particular, the Red Wine dataset has more average quality values compared to the White Wine, which is more Gaussianly Distributed. We can also observe some differences between the two kinds of wine in the other features. For example the *Sulfure Dioxide* (both total and free), and the *Sulphates* are right-skewed in the Red Wine data, whereas they are not in the White Wine data. In general, the White Wine data seems to be more symmetric, and with more outliers, probably due to the fact that it has more entries. The merged dataset reflects the properties of the White Wine dataset: having more entries, it's more influential.

2.2.2 Feature Differences

By inspecting the dataset statistics, we can notice some differences in the trends of the values of the wines. In particular, white wine seems to have much more *Total Sulfure Dioxide* (mean value of ≈ 120 vs. ≈ 45 of the red wine), while having less *Fixed* and *Volatile Acidity* values (mean of respectively ≈ 6.79 and ≈ 0.27 vs. ≈ 8.30 and ≈ 0.52 of the red wine.

2.3 Fuzzy Clustering

2.3.1 Cost function analysis

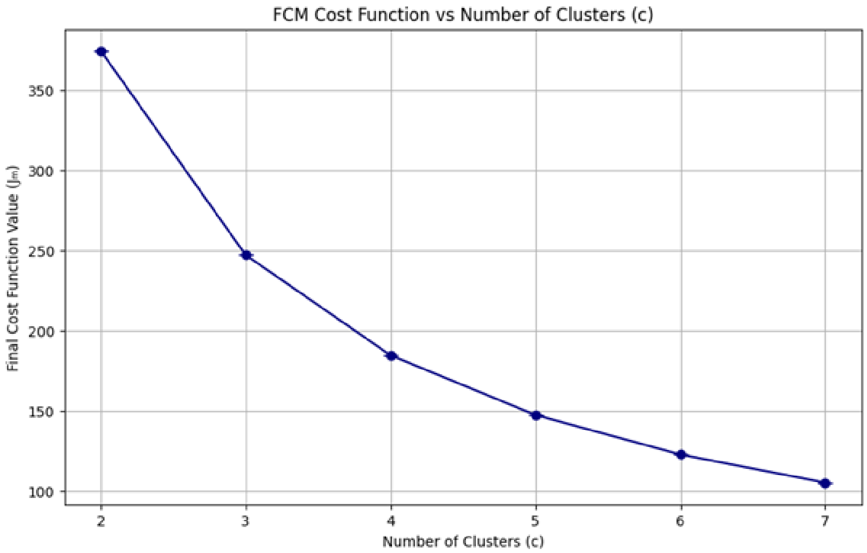

Figure 6: FCM values per number of clusters

From the FCM clustering criterion analysis (6), we observe a clear "elbow" between c = 2 and c = 3, where the cost function J_m experiences its largest drop. A smaller but still noticeable decrease occurs between c = 3 and c = 4. However, after c = 4, the reductions in cost become increasingly marginal, indicating diminishing returns in terms of clustering quality. In *fuzzy c-means*, a decrease in the cost function suggests that the algorithm is finding tighter and more well-defined clusters, with less fuzzy overlap between them. This is a desirable outcome, as it typically reflects more coherent groupings within the data. However, if the cost keeps decreasing only slightly as we increase the number of clusters, it often signals that the algorithm is fitting noise or over-segmenting the data. Although the elbow method should be used cautiously and not as a standalone criterion, this behavior suggests that **c = 2**, **3**, and **4** are the most informative cluster counts. Beyond c = 4, the cost function plateaus, reinforcing the idea that increasing the number of clusters further yields minimal gain in clustering performance.

2.3.2 Validation indices analysis

The validation indices confirm the trend observed in the cost function analysis, diminishing returns after **c = 4**. Notably, the only cluster counts that yield meaningful values across all three validation metrics (**Xie-Beni Index, Silhouette Score**, and **ARI**) are c = 2, 3, and 4. Beyond c = 4, the Xie-Beni Index sharply increases (jumping to values as high as 20 or more), and the scores become unstable, indicating potential overfitting or incoherent clusters. Among the cluster counts analyzed, c = 3 emerges as the most balanced and optimal choice, achieving the best Xie-Beni score and Silhouette score, while also being a close second in ARI. Given this consistent performance, we select c = 3 as the optimal

Clusters	**Xie-Beni**	**Silhouette**	**ARI**
2	1.3664	0.2071	**0.0604**
3	**0.6995**	**0.2501**	0.0595
4	0.9068	0.1548	0.0536
5	28.3104	0.1032	0.0530
6	20.3117	0.0960	0.0518

Table 1: Comparison of clustering metrics for different cluster counts

number of clusters for subsequent analysis, particularly for the Anomalous Pattern (AP) detection task. Additionally, all tests were conducted across multiple seeds: 1, 42, 70, 100, and 200. Interestingly, the results across all validation indices remained identical for c = 2 to c = 4, and only began to diverge for c $\geq$ 5. This stability across seeds aligns with an earlier observation made by my teacher, who described the dataset as "*well-behaved*", meaning it yields consistent clustering results regardless of initialization. As a result, we do not expect significant variability when running the **Anomalous Pattern Clustering** Algorithm.

2.4 Anomalous Clustering

As predicted, after implementing the **Anomalous Clustering Algorithm** as the Initialization for our **Fuzzy C-Means (FCM)** clustering, we observed no significant differences in the resulting clusters. The chosen threshold was **25**, which, given our dataset size of **6,497 samples**, ensures that no resulting cluster will have fewer than 25 data points. This represents a minimum cluster size of approximately **0.38%** of the total data, which is acceptable for this dataset, especially considering the natural density and distribution observed in earlier exploratory data analysis.

2.4.1 3-Cluster PCA Visualization

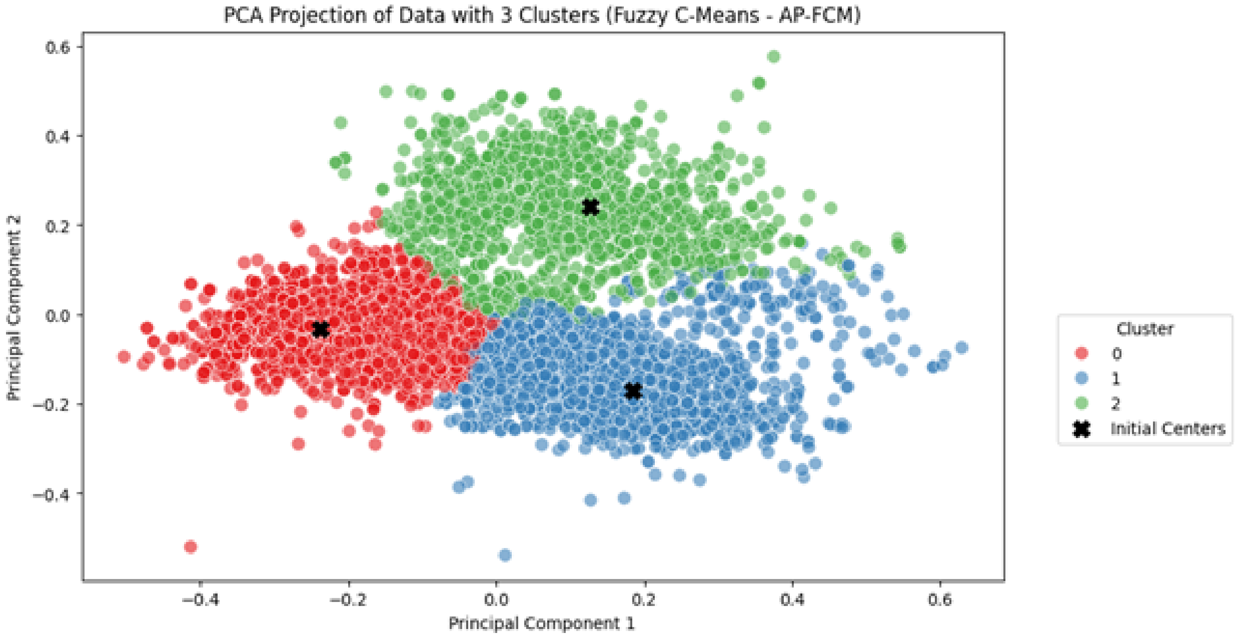

Figure 7: 3-Cluster PCA Visualization

From the PCA visualization with 3 clusters, our choice of c, backed up by the previous analysis of our validation indices and cost function, it becomes clear that the initial centroids discovered via Anomalous Pattern Clustering are already closely aligned with the natural centers of the three clusters. This further reinforces the notion that the dataset has an inherent cluster tendency, a desirable trait in clustering applications. In well-behaved datasets, initial centroids, even those selected through advanced strategies like Anomalous Clustering, do not significantly outperform random seeds.

2.4.2 2-Cluster PCA Visualization

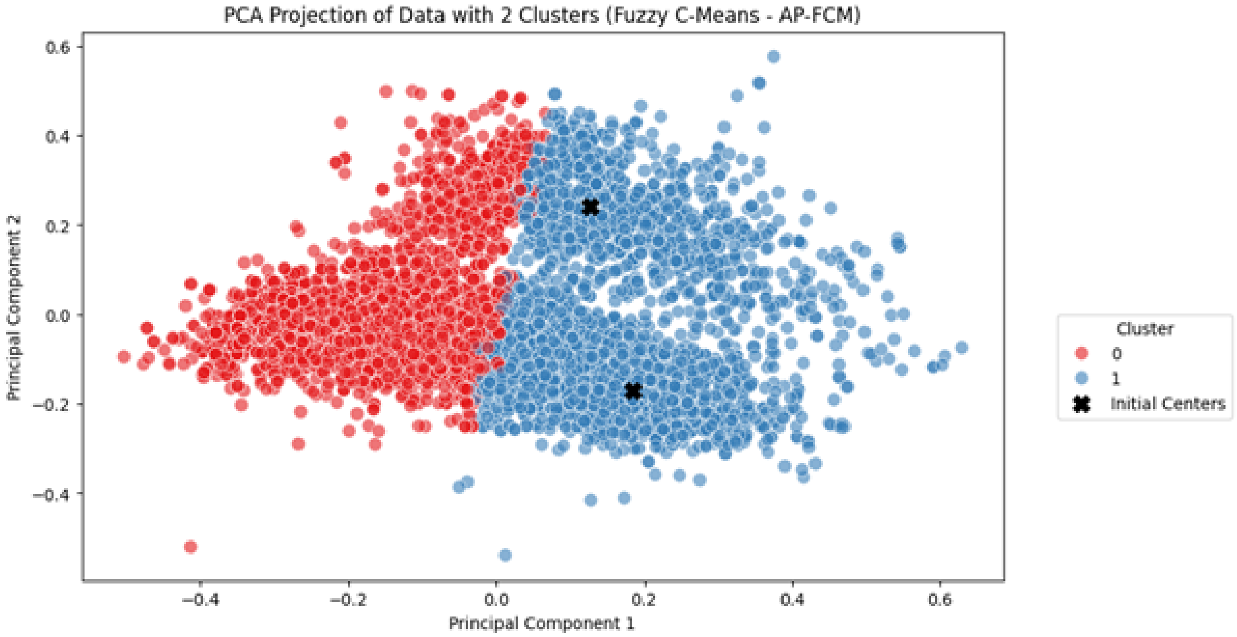

Figure 8: 2-Cluster PCA Visualization

However, an interesting observation arises with c = 2, where one of the initial centroids is noticeably distant from the eventual center of the red cluster (cluster 0). This suggests that forcing the data into only two clusters results in a more complex optimization path for centroid movement. This longer path to convergence indicates that c = 2 might not sufficiently capture the underlying structure, leading to a lower Silhouette Score and high Xie-Beni Index score (compared to c=3 and 4), as previously observed.

2.4.3 4-Cluster PCA Visualization

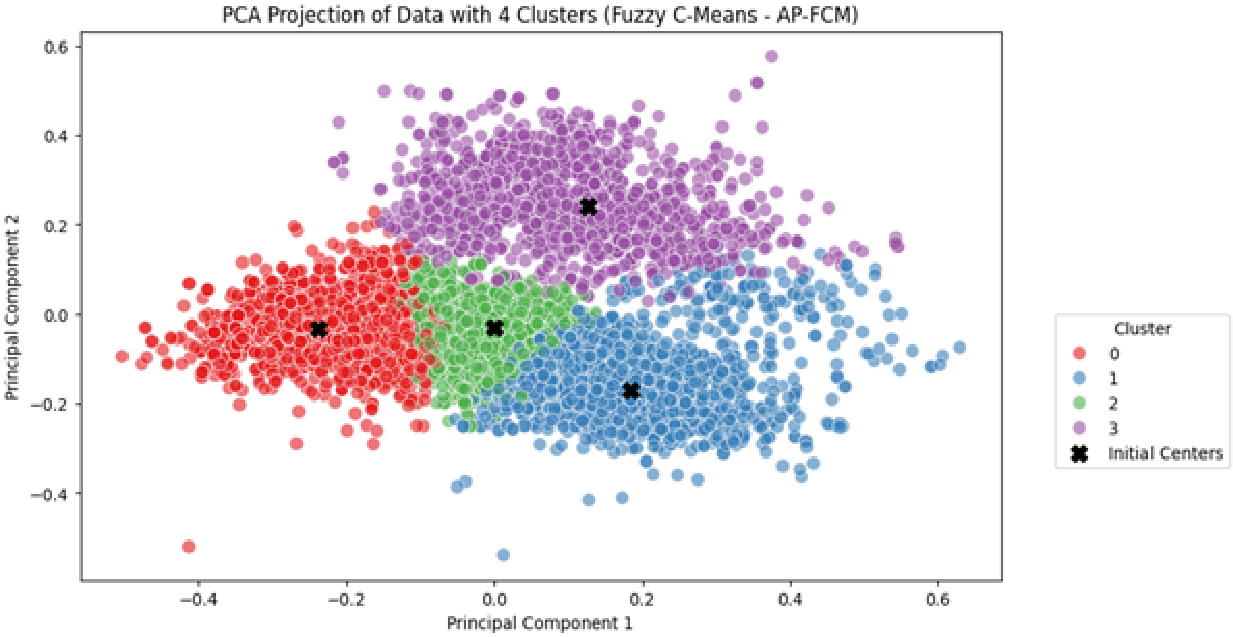

Figure 9: 4-Cluster PCA Visualization

On the other hand, for c = 4, the initial centroids remain well-centered, and the overall clustering quality, while slightly lower in terms of validation metrics compared to c = 3, still reflects the dataset's robustness.

2.4.4 Final considerations

In all tested configurations (c = 2, 3, 4), the validation indices remained consistent with the earlier experiments using standard FCM. This further supports the assessment that the Wine Quality Dataset is well-behaved, exhibiting high inter-cluster separation and low intra-cluster variance, characteristics indicative of clear and cohesive clustering structures. In such cases, different clustering initializations and parameter settings tend to converge to similar local optima. Consequently, all validation indices yield stable results, and advanced initialization techniques (like Anomalous Clustering) act more as validation tools than necessary improvements. Figures 10 and 11 show the 3D PCA and SVD visualizations for the final choice of **c** (=3).

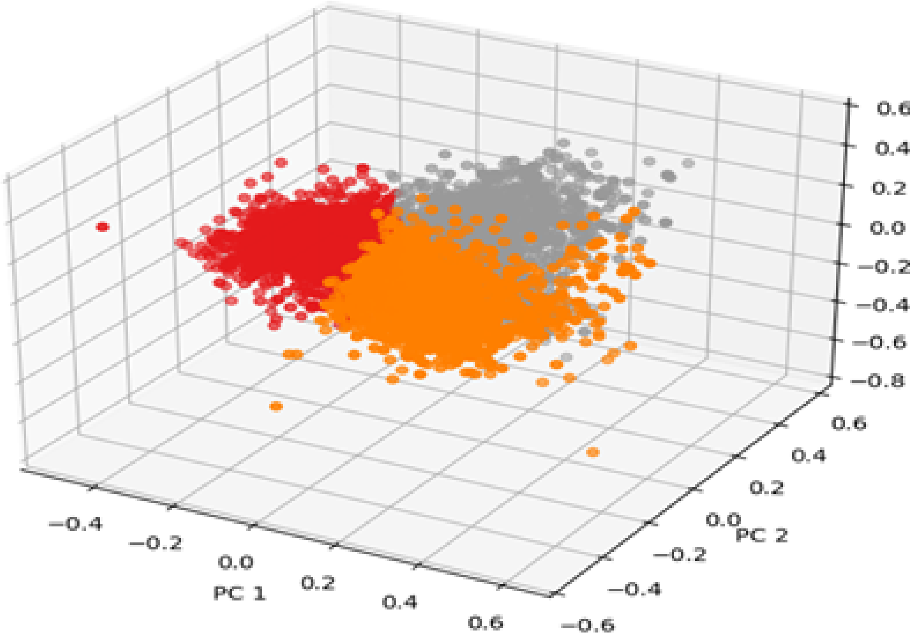

Figure 10: 3D PCA visualization for optimal number of clusters (3) and Anomalous Clustering as initialization method

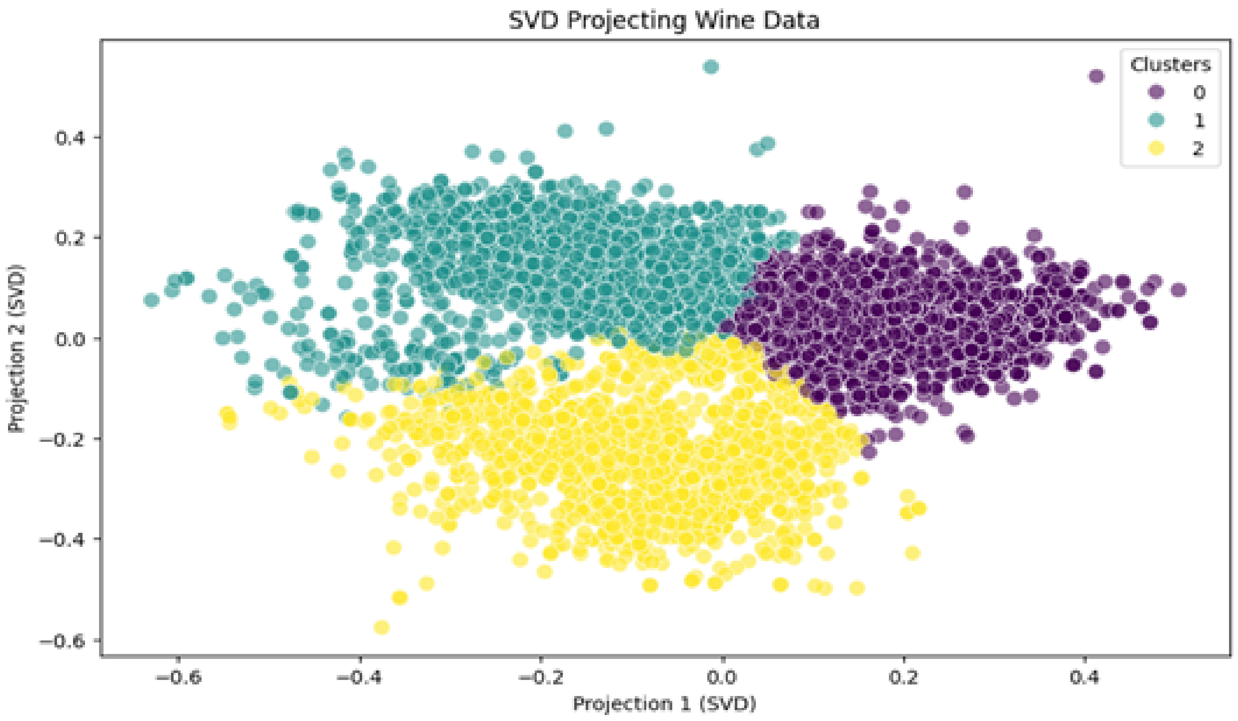

Figure 11: SVD visualization for optimal number of clusters (3) and Anomalous Clustering as initialization method

2.5 Normalization and Comparative Analysis

For the clustering experiments, we tested both Range Normalization and Z-score Normalization. After evaluating the results from both methods, **Range Normalization** was selected as the best fit for our dataset. This decision was supported by the fact that the points seem to "cluster together" more effectively, as shown visually in the PCA and SVD visualizations. Specifically, the Range Normalization produced a clustering structure where the data points within each cluster are more closely grouped, exhibiting stronger cohesion and clearer separation between clusters compared to Z-score Normalization. This choice was further confirmed by the stability of the clustering results across different seeds, as well as the consistent performance of the validation indices.

c	Cost	Xie-Beni	Silhouette	ARI
2	35729.45	22.6983	0.1740	0.0019
3	23786.37	20.3566	0.2158	0.0362
4	17668.91	321.5452	0.1576	0.0353
5	14109.28	698.5475	0.1244	0.0351

Table 2: Clustering results for the Z-score normalized data

Table 2 shows the clustering results for the Z-score normalized data. When comparing the results from Range Normalization with those obtained using Z-score Normalization, the differences are significant. The Xie-Beni index, which measures compactness and separation, shows a dramatic increase in values with z-score Normalization, jumping from **0.6995** at **c=3** in range normalization (1) to **22.6983** for **c=2** in Z-score Normalization. This highlights a significant deterioration in clustering quality. Similarly, the Silhouette score drops from **0.2501** for **c=3** with Range normalization to **0.2158** in Z-score Normalization, which suggests that the clusters have a less distinct boundary and are less cohesive. Additionally, the **ARI** (Adjusted Rand Index), which measures the agreement between the clusters and the ground truth, also suffers a decline from **0.0604** (Range) to **0.0362** (Z-score), further confirming the suboptimal performance of Z-score when compared to Range Normalization. The cost function values are significantly higher with Range Normalization (e.g., **35729.45** for **c=2**), which, although expected due to the scale of the data, still indicates a higher level of within-cluster variance compared to the results obtained with Z-score normalization. Visually, the clusters in the Z-score Normalization case appear more overlapped, with reduced separation between them. Figures (12 and (13) show clustering results for 4 clusters, using Z-score normalization and Range Normalization respectively.

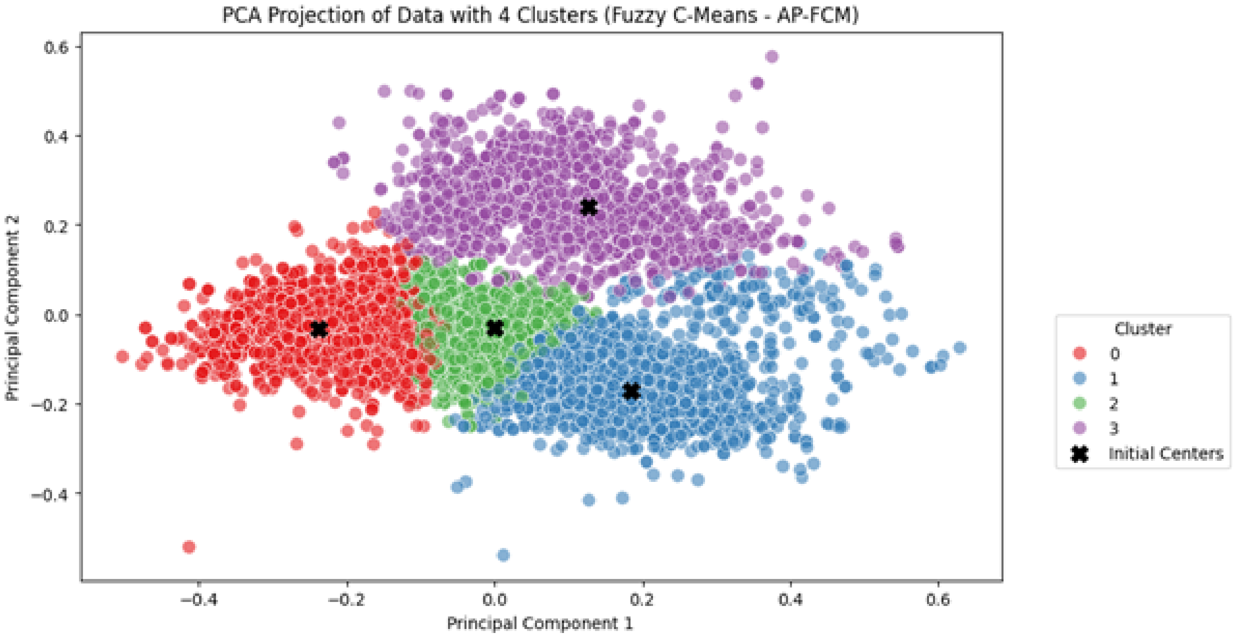

Figure 13: Range Normalization with Fuzzy C-Means for 4 Clusters

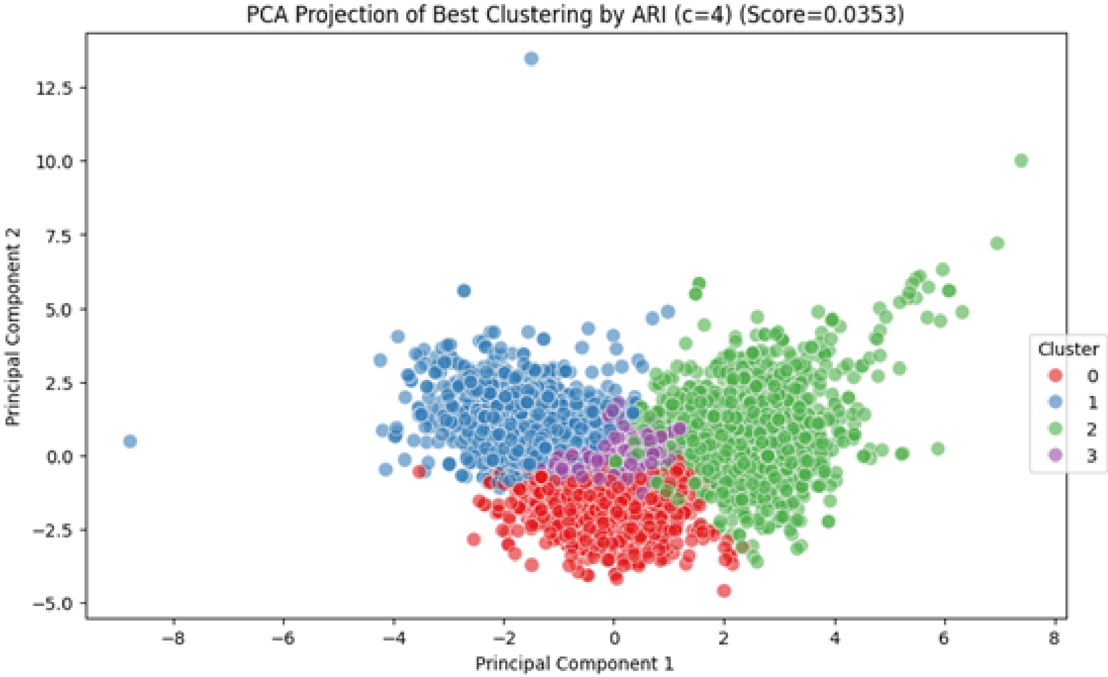

Figure 12: Z-score Normalization with Fuzzy C-Means for 4 Clusters

In the Z-score normalization plot, we observe a decrease in inter-cluster separation, with the clusters appearing more similar to each other. This suggests higher intra-cluster similarity, which is undesirable as it indicates that the clustering algorithm struggles to clearly distinguish between the clusters. In contrast, with Range Normalization, the clusters are more distinctly separated, with clear boundaries between them. This separation is particularly noticeable when compared to the Z-score Normalization case, where one of the clusters is almost indistinguishable from the others. In fact, in the Z-score case, this cluster appears to be positioned in the middle of the other three, making it difficult to identify clearly.

In summary, **Range Normalization** provided a better fit for the dataset, allowing the clustering algorithm to perform more effectively. It resulted in much clearer and more meaningful clustering outcomes, both visually and quantitatively (via validation indices). Z-score normalization, on the other hand, led to poorer clustering performance, with higher intra-cluster similarity and reduced inter-cluster separation, making it less suitable for this dataset.

2.6 Principal Component Analysis

For this study case, we selected a subset of 4 features from our data. They are: **Fixed Acidity**, **Volatile Acidity**, **Citric Acid**, **pH**.

This selection allowed us to have a good amount of strictly related attributes relative to the same phenomenon: the *acidity profile* of the wine. Concerning the choice of the pre-defined groups, it was decided to aggregate the entries based on the quality value: more precisely, data points with a quality in [3,4] were assigned a score of ”low”, points with a quality in [5,6,7] were assigned a score of ”medium”, and points with a quality in [8,9] were assigned a score of ”high”. Therefore the analytical task we aimed to address with the following study consisted in evaluating the possibility of defining quality-based clusters on this manually-reduced set of features. For the visualization with dimensionality reduction, the SVD method was chosen because of its greater space efficiency compared to PCA.

2.6.1 SVD Visualization

For the visualization and analysis of this study case, both normalization approaches were used: namely *range normalization* 14, and *z-score normalization* 15.

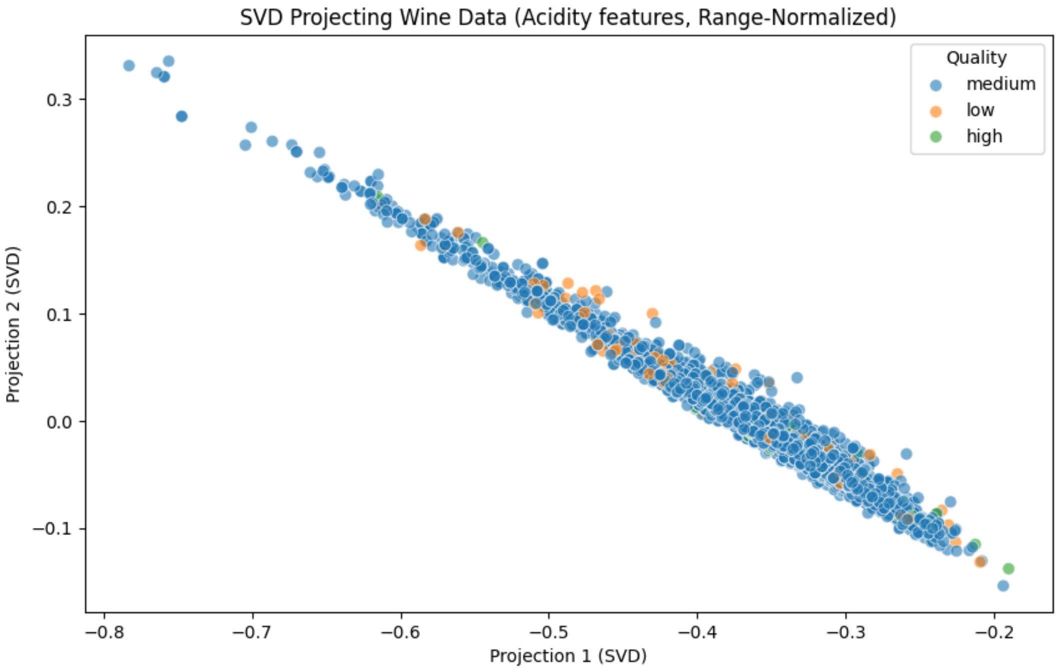

Figure 14: SVD projection over a subset of features (Acidity-related, Range-normalized)

Eigenvalue	Explained Variance (Cumulative)
λ_1	97.72%
λ_2	99.83%
λ_3	99.95%
λ_4	100%

Table 3: Eigenvalues of SVD projection over a subset of features (Acidity-related, Range-normalized)

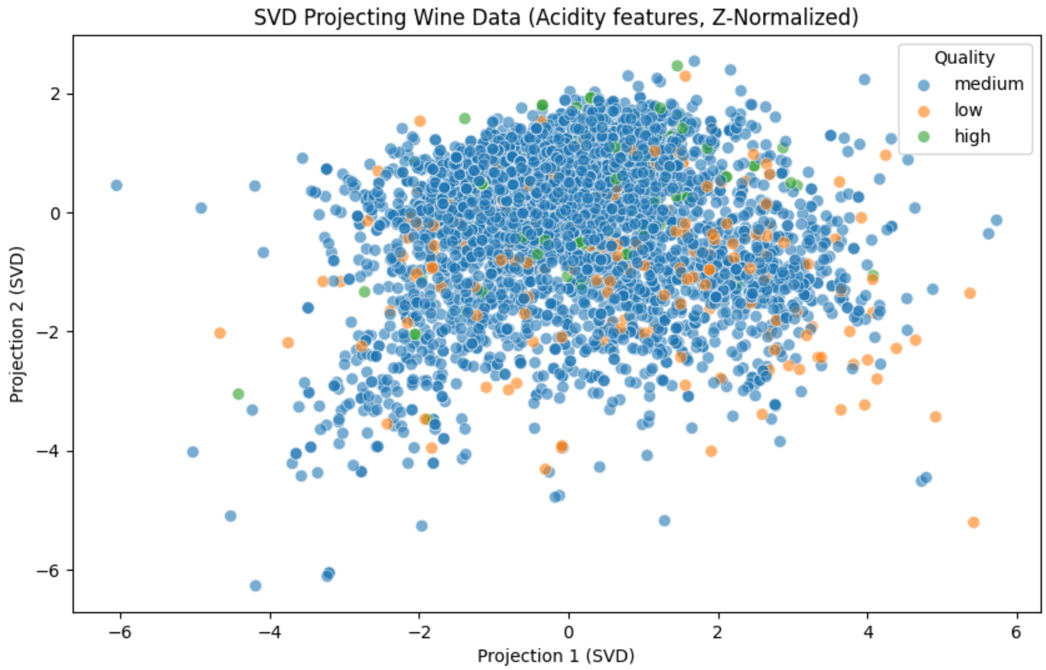

Figure 15: SVD projection over a subset of features (Acidity-related, Z-normalized)

Eigenvalue	**Explained Variance (Cumulative)**
λ_1	43.17%
λ_2	73.63%
λ_3	90.83%
λ_4	100%

Table 4: Eigenvalues of SVD projection over a subset of features (Acidity-related, Z-normalized)

Range Normalization: **PC1** explains almost the totality of the variance. This means that that single component is able to capture the majority of the structure. Therefore, **PC2** has little utility in our case. Also, the two components seem inversely (and linearly) correlated: they might be capturing tradeoffs between the features. This is not surprising though, as the feature choice voluntarily concerned strongly correlated features.

Z-score Normalization: The PCA behaviour is very different, as we need more principal components to capture a considerable amount of variance: our data, when reduced to zero mean and unitary variance does not lay almost flat on a single dimension. Concerning the quality score, high-quality wine entries seem to be slightly more concentrated on the top and top-right area, but are not easily separable anyway.

Experimenting with a different feature subset: to support the aforementioned claims, the experiment was repeated on a different subset of features, this time relative to *Sugar/fermentation* characteristics. They are: **"residual sugar", "alcohol", "volatile**

acidity" and **"density"**. This different subset includes "alcohol", therefore we expect to see a more determinant split in the three pre-selected quality clusters (this because, as discussed in 2.2, alcohol seems to be quite correlated with quality). Also, since the features are less "similar" to each other (in the "acidity" subset they all referred *directly* to acidity, whereas in this setting they are more weakly linked), we expect to see less of a "linear correlation" plot with the Principal Components.

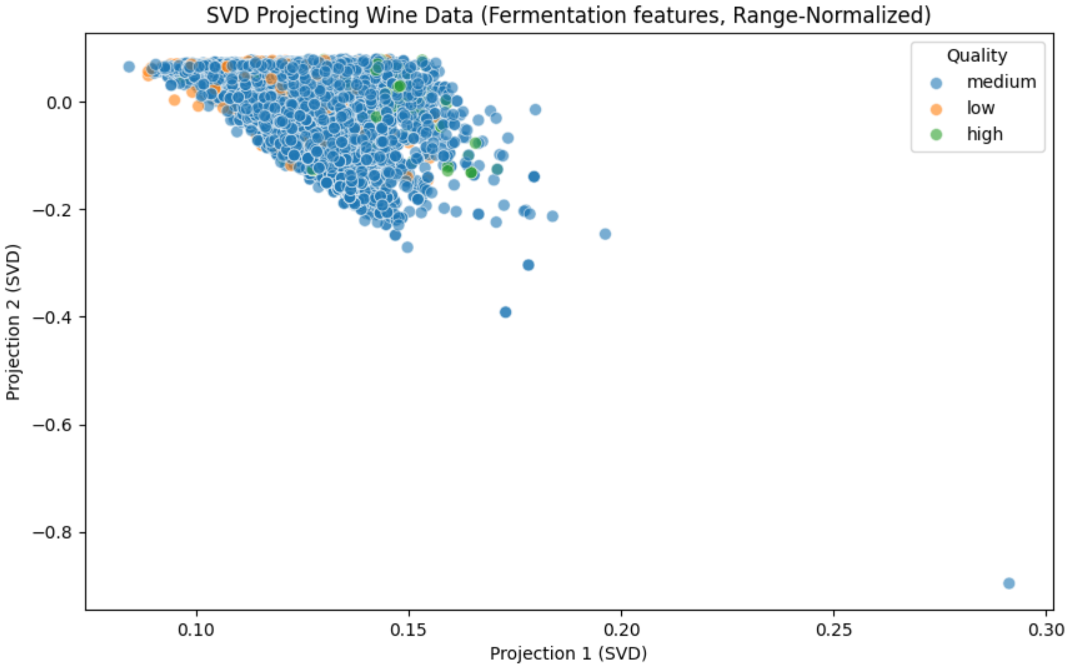

Figure 16: SVD projection over a subset of features (Fermentation-related, Range-normalized)

Eigenvalue	**Explained Variance (Cumulative)**
λ_1	74.35%
λ_2	99.45%
λ_3	99.99%
λ_4	100%

Table 5: Eigenvalues of SVD projection over a subset of features (Fermentation-related, Range-normalized)

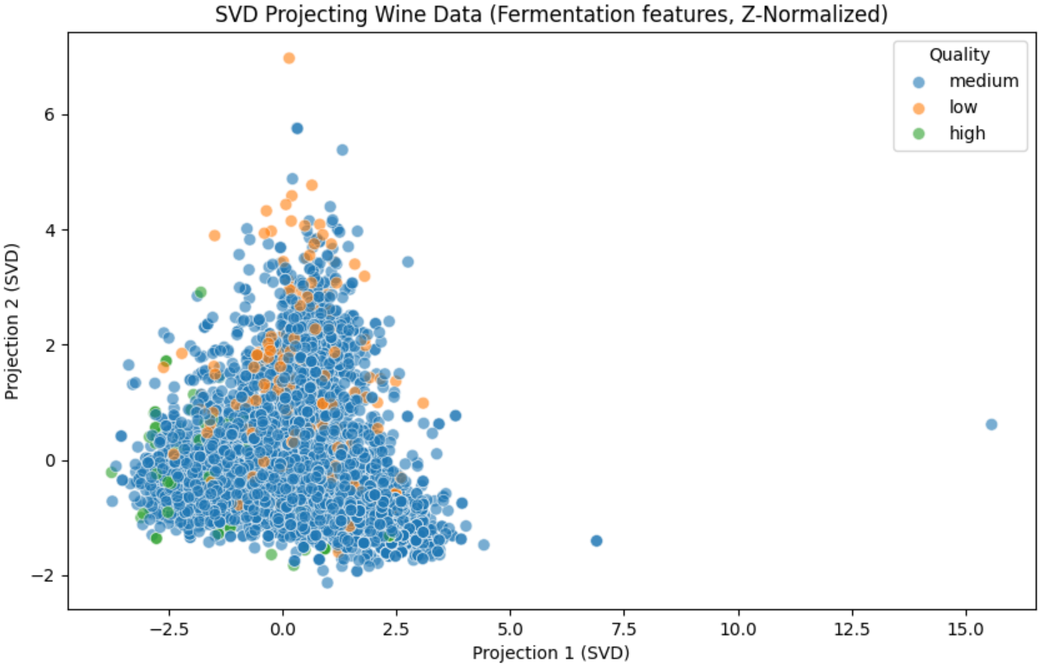

Figure 17: SVD projection over a subset of features (Acidity-related, Z-normalized)

Eigenvalue	Explained Variance (Cumulative)
λ_1	52.13%
λ_2	80.88%
λ_3	95.33%
λ_4	100%

Table 6: Eigenvalues of SVD projection over a subset of features (Fermentation-related, Z-normalized)

And, as visible in figures 16 and 17, this is in fact the case: the ”high” and ”low” quality data points are clearly more clustered together and easier to tell apart from the ”medium”, in both the normalizations, while also being in opposite directions. Also, the explained variance in the *range-normalized* experiment is not nearly as high in the first component, compared to the one done on the Acidity feature subset. Nevertheless, it still cumulates to **99.45** with the addition of the second PC. Concerning the explained variance in the *z-normalized* case, it is more balanced between different components, similarly to the previous setting.

2.6.2 Conclusive thoughts

In conclusion, by looking at the results of the different experiments, we have assessed the importance and variability that dimensionality-reduced visualizations can have. By performing a ”manual” dimensionality reduction, we have seen that choosing strictly-related features clearly translates into a ”linear correlation” type of PCA plot, whereas using a more diverse subset of features results in a more ”spherical” and more nuanced visualization. Also, these experiments showed how clearly sensitive PCA is to the choice of the normalization method: this because variance is directly affected by the scale of the data.

2.7 Spectral Clustering

2.7.1 Football Dataset

Figures 18, 19, 20, 21 show the NMI and Modularity values obtained with Normalized and Unnormalized Laplacian use, respectively.

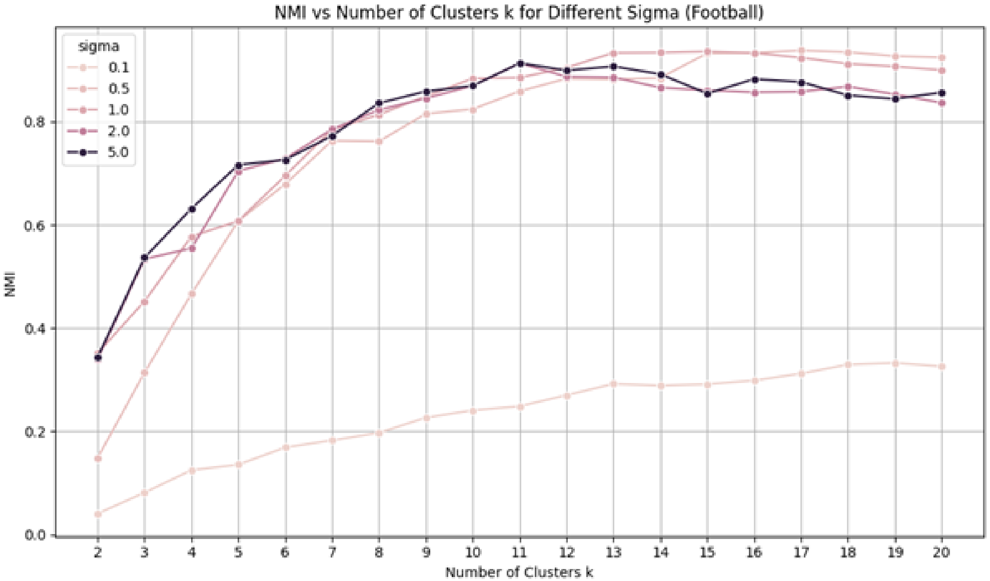

Figure 18: NMI values for Normalized Laplacian on Football dataset

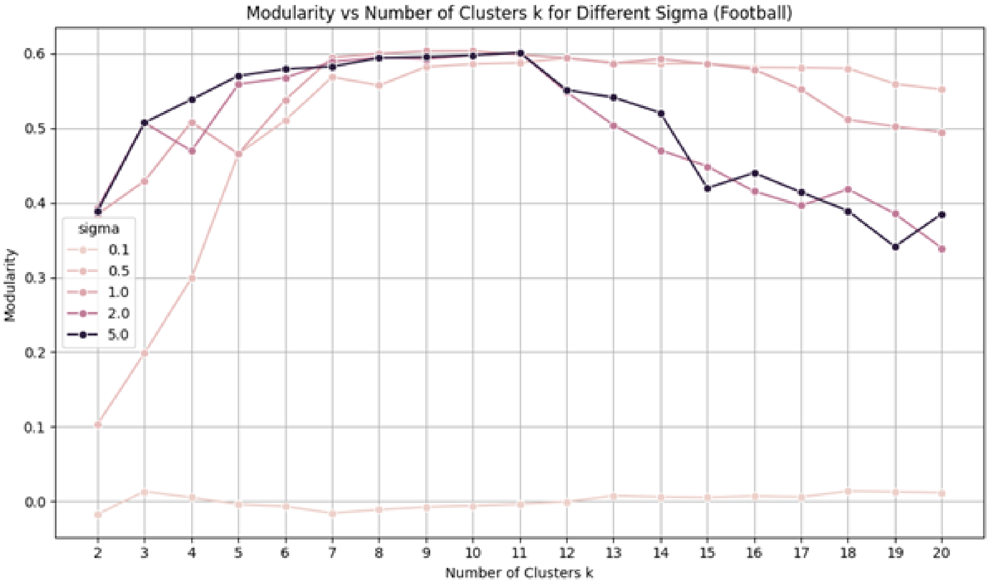

Figure 19: Modularity values for Normalized Laplacian on Football dataset

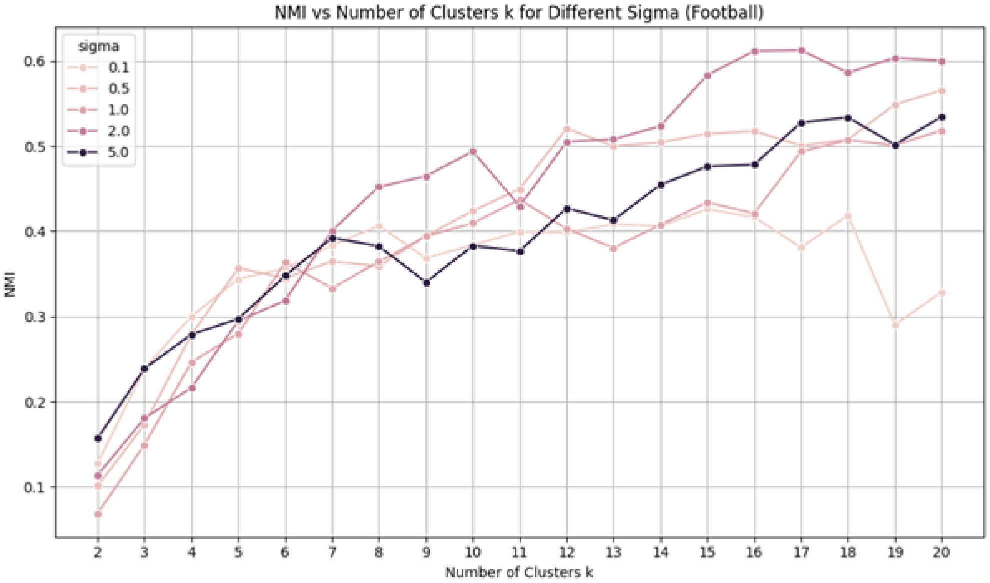

Figure 20: NMI values for Unnormalized Laplacian on Football dataset

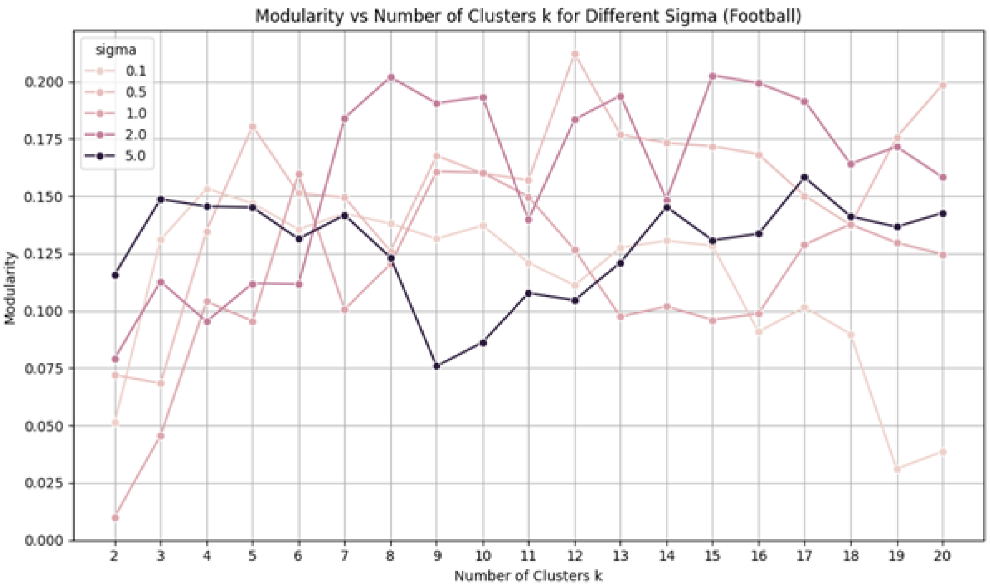

Figure 21: Modularity values for Unnormalized Laplacian on Football dataset

Sigma(σ)	NMI (Normalized)	Modularity (Normalized)	NMI (Unnormalized)	Modularity (Unnormalized)
0.1	0.332 (k=19)	0.014 (k=18)	0.472 (k=19)	0.092 (k=18)
0.5	0.938 (k=17)	0.593 (k=12)	0.965 (k=17)	0.741 (k=12)
1.0	0.936 (k=15)	0.603 (k=10)	0.962 (k=15)	0.752 (k=10)
2.0	0.913 (k=11)	0.601 (k=11)	0.940 (k=11)	0.751 (k=11)
5.0	0.913 (k=11)	0.601 (k=11)	0.940 (k=11)	0.751 (k=11)

Table 7: Best performance metrics across different Gaussian σ values on Football dataset

Table 7 synthesizes the best performances obtained with different values of σ. The best performing configuration was found at σ=**0.5** with **k=14** clusters, yielding a Normalized Mutual Information **(NMI)** of **0.934** and modularity of **0.592**. This combination strikes an effective balance for capturing meaningful community structure within the dataset.

Influence of the number of clusters (k): When using the normalized Laplacian, both NMI and modularity consistently improve as the number of clusters k increases from 2 up to roughly 11 to 17 clusters. Beyond this range, the metrics tend to plateau or decline slightly, indicating that adding more clusters results in over-segmentation, where communities are broken down into excessively small and fragmented groups. This suggests an optimal range for k in terms of producing coherent and well-defined clusters. In contrast, the unnormalized Laplacian exhibits less stable behavior across varying k. Although NMI and modularity generally improve up to a certain point, the trends show frequent fluctuations and sharp dips. This erratic pattern reflects instability in the clustering outcomes, making it challenging to reliably select the optimal number of clusters based on these metrics when using the unnormalized Laplacian.

Influence of the Gaussian Kernel (σ): At σ=**0.1**, both NMI and modularity remain consistently low across all cluster counts, demonstrating that the similarity graph constructed at this scale is overly sparse. This sparsity prevents the clustering algorithm from effectively identifying meaningful communities, as few connections exist between nodes to form coherent groups. On the other hand, σ values of **0.5** and **1.0** yield the highest NMI (**0.938** and **0.936** respectively) and modularity scores (**0.593** and **0.603**). These values indicate an optimal balance between capturing local neighborhood information and broader global structure in the graph. This balance allows the algorithm to detect well-defined communities that align closely with the underlying data structure. Increasing σ further to **2.0** and **5.0** causes a modest decline in the best achievable NMI (≈**0.913**) and modularity (≈**0.601**). Moreover, modularity exhibits a sharper drop-off when the number of clusters surpasses the optimal range. This pattern is characteristic of oversmoothing in the similarity graph: as σ grows large, the Gaussian kernel effectively blurs distinctions between communities, causing previously separate clusters to merge and reducing the overall quality of the clustering.

Influence of Laplacian type: The normalized Laplacian produces clear and smooth trends, with NMI and modularity steadily increasing as k grows and peaking within a consistent range of cluster counts. This stability holds across the different σ values tested, making the normalized Laplacian more interpretable and reliable for spectral clustering. By comparison, the unnormalized Laplacian results in scattered and inconsistent outcomes, characterized by multiple spikes and sudden drops in both NMI and modularity. While it occasionally attains higher peak NMI values (around **0.96**), these peaks are less reliable as modularity scores are generally less stable and often lower. This inconsistency complicates the interpretation and reduces the practical usefulness of the unnormalized Laplacian in this context. To summarize, the normalized Laplacian offers more stable and interpretable clustering performance, whereas the unnormalized Laplacian is more sensitive to the graph structure and prone to noisier, less predictable results despite occasional strong peaks.

Interpretation of identified communities: Using the best-performing configuration (normalized Laplacian, σ = 0.5, k = 14), spectral clustering produces communities that closely align with the known conference affiliations in the data. While no explicit mapping between clusters and conferences is required to compute NMI or modularity, the high scores alone indicate a strong structural match. Given that the network represents games played between Division football teams, and that most games occur within conferences (as per league scheduling), the clustering method successfully groups teams that are more densely connected, that are from the same conference. The algorithm captures this naturally via spectral clustering, as evidenced by the high modularity (>0.58) and tight clustering correspondence (NMI > 0.93) at optimal parameters.

Conclusive thoughts: While spectral clustering demonstrates strong performance in uncovering meaningful community structure, particularly when using the normalized Laplacian with well-chosen σ and k values, its effectiveness is highly sensitive to parameter selection. The stability and interpretability of results rely heavily on careful tuning, especially of the Gaussian kernel width. Moreover, the method struggles with graph sparsity (as seen at low σ) and oversmoothing (at high σ), which can obscure community boundaries. The unnormalized Laplacian, while occasionally yielding higher peak scores, suffers from instability and unpredictability, limiting its practical applicability. Overall, spectral clustering proves powerful but requires a nuanced and data-informed approach to parameterization, and its sensitivity may present challenges in less structured or noisier datasets.

2.7.2 US Political Books Dataset

Figures 22, 23, 24, 21 show NMI and Modularity values obtained with Normalized and Unnormalized Laplacian use, respectively.

Table 8 synthesizes the best performances obtained with different values of σ.

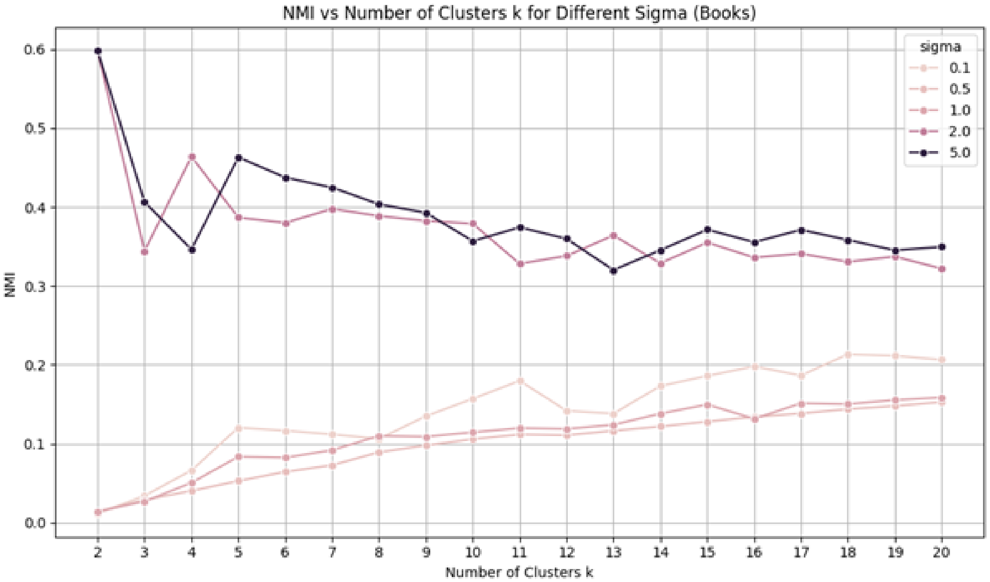

Figure 22: NMI values for Normalized Laplacian on Books dataset

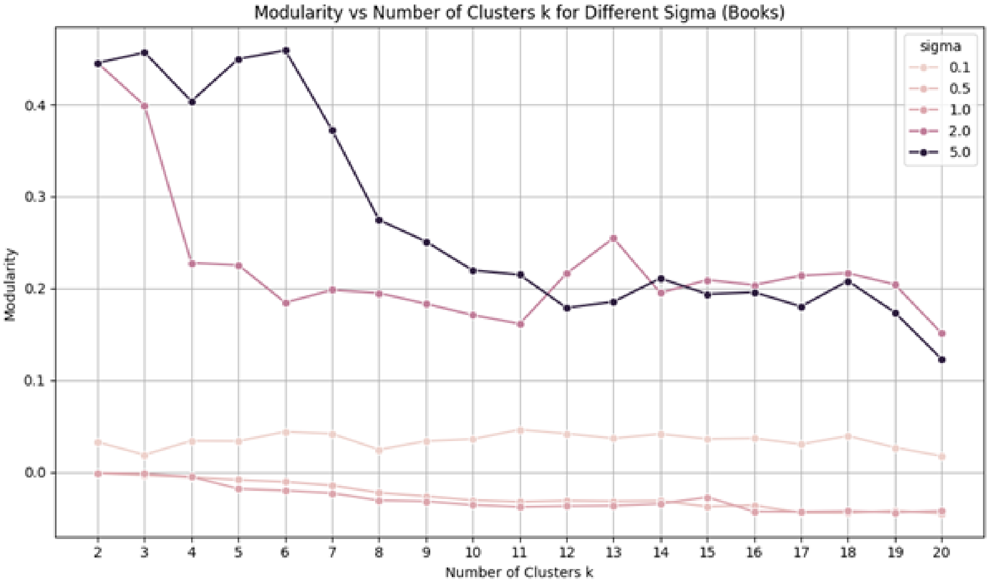

Figure 23: Modularity values for Normalized Laplacian on Books dataset

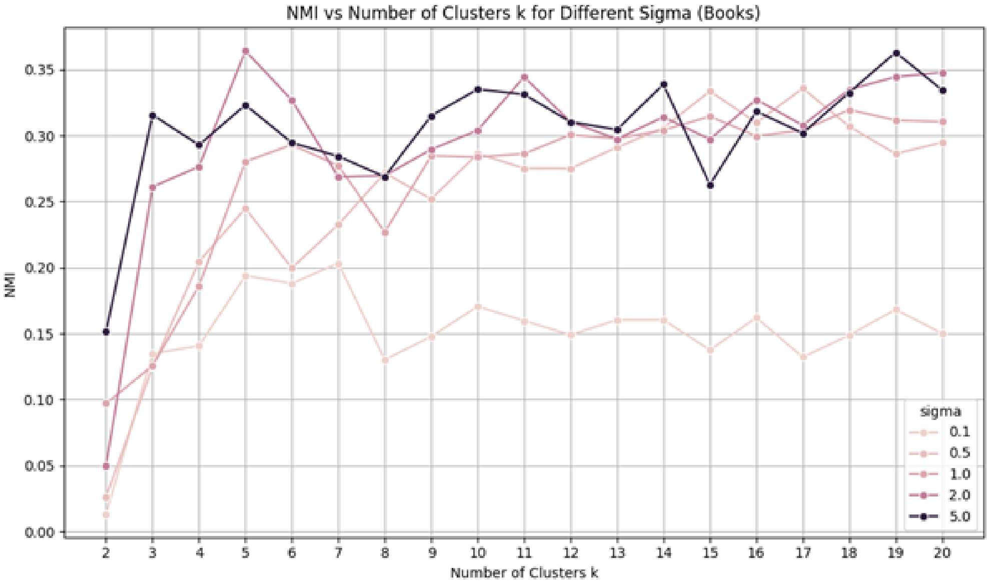

Figure 24: NMI values for Unnormalized Laplacian on Books dataset

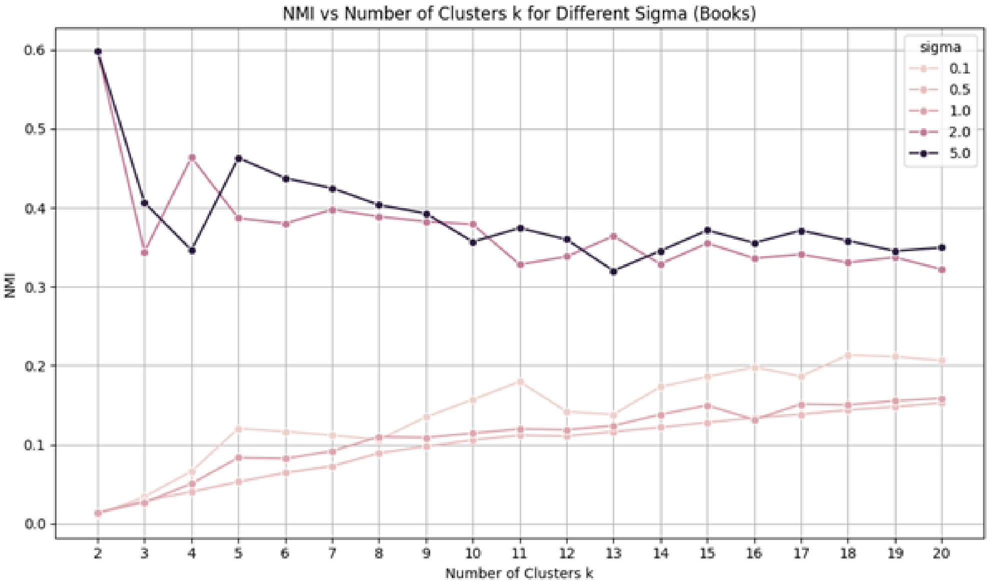

Figure 25: Modularity values for Unnormalized Laplacian on Books dataset

Sigma(σ)	**NMI (Normalized)**	**Modularity (Normalized)**	**NMI (Unnormalized)**	**Modularity (Unnormalized)**
0.1	0.214 (k=18)	0.046 (k=11)	0.194 (k=5)	0.075 (k=5)
0.5	0.153 (k=20)	-0.001 (k=2)	0.334 (k=17)	0.139 (k=5)
1.0	0.159 (k=20)	-0.001 (k=2)	0.319 (k=18)	0.136 (k=15)
2.0	0.598 (k=2)	0.445 (k=2)	0.348 (k=20)	0.166 (k=20)
5.0	0.597 (k=2)	0.456 (k=3)	0.363 (k=19)	0.241 (k=19)

Table 8: Best performance metrics across different Gaussian σ values on Books dataset

The best performing configuration was found at $\sigma = 2.0$ with k = 2 clusters, yielding a Normalized Mutual Information (NMI) of 0.598 and modularity of 0.445. This aligns closely with the natural political divide (liberal vs. conservative), indicating that community structure in the dataset is dominated by this binary polarization.

Influence of the Number of Clusters (k): In contrast to the football dataset, the Books network exhibits a strong bimodal community structure, reflected in the highest NMI and modularity being achieved at k = 2 for the normalized Laplacian. This makes intuitive sense given the dataset represents US political ideology, republics or democrats. For small σ values ($\sigma \leq 1.0$), increasing k slightly improves NMI up to k $\approx$ 20, but both NMI and modularity remain low, indicating weak or noisy cluster formation due to insufficient graph connectivity. With larger σ values ($\sigma = 2.0$ and $\sigma = 5.0$), NMI and modularity stay relatively low and stable across a range of k values(around 0.35 and 0.40 for NMI and 0.20 for modularity), but the peak still clearly occurs at k = 2(with NMI 0.59 and modularity 0.44). This reinforces the idea that the most meaningful split is not in over segmenting the data but in revealing the primary ideological divide. Meanwhile, the unnormalized Laplacian exhibits significantly greater variability across different values of k. While its highest NMI (0.364) appears at k = 19 for $\sigma = 5.0$, the associated modularity is notably lower, making the clustering less meaningful. This inconsistency, especially at higher k, underscores the unnormalized Laplacian's tendency toward overfitting and unstable segmentation. As seen in the football dataset, its performance is often erratic, with frequent dips and unpredictable trends, making it less reliable for detecting well-defined community structures, and thus harder to determine the optimal number of clusters with confidence.

Influence of Gaussian Kernel (σ): At $\sigma = \mathbf{0.1}$, both normalized and unnormalized Laplacians perform poorly, as the similarity graph becomes too sparse. This leads to many disconnected or weakly connected nodes, degrading the algorithm's ability to infer community structure. $\sigma = \mathbf{2.0}$ and **5.0** yield the highest performance under the normalized Laplacian, with modularity $\approx$ 0.44 and **NMI** near **0.60**, both peaking at $\mathbf{k = 2}$. This suggests that a moderately broad kernel width allows the algorithm to integrate meaningful global structure without overly blurring the distinctions between liberal and conservative communities. The performance at $\sigma = \mathbf{0.5}$ and σ **1.0** is relatively poor by comparison. These values seem to be in a transitional zone, too wide to preserve local signal, yet too narrow to capture global structure. This echoes the pattern seen in the football dataset, where tuning σ appropriately was crucial for optimal clustering.

Influence of Laplacian type: The normalized Laplacian once again provides more stable and interpretable clustering results, with clear performance peaks and smoother trends. Its NMI and modularity curves rise predictably with k for small σ and peak decisively at low k for higher σ, revealing clear structural signals in the data. By contrast, the unnormalized Laplacian produces noisier and less reliable interpretations. Although it occasionally reaches high NMI values, the associated modularity scores vary considerably. This behavior mirrors what was observed in the football dataset.

Interpretation of identified communities: Using the best-performing configuration (**normalized Laplacian, $\sigma = 2.0$, k = 2**), spectral clustering identifies the main ideological divide in the Books network, likely corresponding to liberal and conservative books. The clustering aligns with the known labels, reflected in a moderate NMI of about **0.6** and modularity around **0.45**. However, this NMI is notably lower than that achieved for the

Football dataset ($\approx$ **0.93**), indicating that the community structure in the Books dataset is less clear-cut or noisier. Unlike the Football dataset, which required many clusters (**k** $\approx$ **14–17**) to capture detailed conference groupings, the Books network's best partition is the simpler two-cluster split reflecting the primary political polarization. This shows spectral clustering's ability to adapt to different network structures but also highlights that the Books dataset has a weaker and less distinct community signal compared to the Football dataset.

Conclusive thoughts: Spectral clustering performs well on the Books dataset, especially with the normalized Laplacian and σ values between 2.0 and 5.0, revealing a clearly defined and interpretable binary community structure that reflects the partisan nature of political book co-purchases. However, similar to the football dataset, performance remains highly sensitive to parameter choices, especially the Gaussian kernel width σ. Overly small or large values lead to sparsity or oversmoothing, respectively. While the unnormalized Laplacian occasionally yields competitive scores, its volatility across k and σ values limits its practical applicability. Overall, the findings support the conclusion that normalized spectral clustering is both robust and adaptive for revealing community structure in real-world graphs, be it sports or politically based.

3 Conclusion

Throughout this project, we explored several clustering techniques and preprocessing strategies on the Wine Quality dataset, drawing key insights into the behavior of the data and the performance of different methods. **Fuzzy clustering**, across various configurations ($c = 2, 3, 4$), consistently produced stable validation indices, reinforcing the notion that the dataset is "well-behaved," characterized by high inter-cluster separation and low intra-cluster variance. The choice of normalization played a critical role: **range normalization** enhanced clustering clarity and performance, while Z-score normalization led to poorer results due to its impact on intra- and inter-cluster separations. Our experiments on dimensionality reduction further emphasized how the selection of features and normalization methods influences **PCA** visualizations, with manual feature selection providing more interpretable, structured projections. **Spectral clustering** revealed itself as a powerful but highly sensitive technique, performing best with careful parameter tuning. However, it also demonstrated limitations related to graph sparsity and oversmoothing, underscoring the need for cautious application in noisy or less structured datasets.

Several potential improvements could strengthen the analysis further. Incorporating a preliminary data cleaning step—such as the removal of outliers and duplicates—would likely enhance cluster definition and stability. Expanding the dataset, both in terms of the number of entries and the diversity of features, could also provide richer information for clustering and better generalizability of the models. A more balanced dataset in terms of quality classes could prevent clustering biases and improve the interpretability of the clusters. Additionally, future work could include a comparative analysis between red and white wine subsets, offering insights into possible structural differences between the two types. Finally, extending the comparison between standard k-means and fuzzy clustering would further illuminate the advantages and trade-offs of adopting soft versus hard clustering approaches for this type of data.

A Function Headers and Parameters

This appendix lists the function headers, their purposes, and the types of their parameters.

- `center_(x: ndarray, cluster: list or ndarray)`
  Computes the mean (center) of a cluster.

- `distNorm(x: ndarray, remains: list, ranges: ndarray, p: ndarray)`
  Computes normalized squared distances between points and a centroid.

- `separCluster(x: ndarray, remains: list, ranges: ndarray, a: ndarray, b: ndarray)`
  Assigns points to the closer of two centroids.

- `anomalousPattern(x: ndarray, remains: list, ranges: ndarray, centroid: ndarray, me: ndarray)`
  Iteratively refines a cluster starting from a centroid using anomalous clustering logic.

- `dist(x: ndarray, remains: list, ranges: ndarray, p: ndarray)`
  Computes normalized squared distances between data points and a point.

- `xie_beni_index(U: ndarray, centers: ndarray, X: ndarray)`
  Computes the Xie-Beni index, a validity measure for fuzzy clustering.

- `pca_application(x_norm: ndarray, y_values: ndarray, init_centroids: ndarray (optional))`
  Applies PCA manually to a normalized dataset, and optionally transforms centroids.

- `plot_clustering(x: ndarray, labels: ndarray, centers: ndarray, title: str, score: float)`
  Plots the clustering results (2D) with cluster centers.

- `spectral_clustering_dense(graph: NetworkX graph, k: int, sigma: float = 1.0, laplacian_type: str = "normalized")`
  Spectral Clustering using Full Eigen Decomposition for small graphs.

- `evaluate_clustering(graph: NetworkX graph, labels: list or ndarray, ground_truth: list or ndarray)`
  Evaluates clustering using Normalized Mutual Information (NMI) and Modularity Score.

- `load_dataset(path: str, label_attr: str (optional))`
  Loads a GML dataset and extracts ground-truth labels from a given attribute.

- `run_spectral_experiments(graph: NetworkX graph, ground_truth: list or ndarray, dataset_name: str, sigmas: list, ks: list)`
  Runs spectral clustering on a graph with multiple σ and k settings.

- `run_full_experiment(dataset_path: str, dataset_name: str, label_attr: str)`
  Runs the full pipeline for spectral clustering experiments on a given dataset (loads the dataset, experiments with different parameters, and plots NMI and Modularity Values)

- `plot_2d_3d_pca(data_pca: DataFrame, fuzzy_membership: array-like, c: int, title: str, score: float, initial_centroids: ndarray (optional))`
  Generates 2D and 3D PCA scatter plots of the dataset colored by fuzzy cluster membership, optionally highlighting initial centroids. Displays clustering score in the title.

- `plot_svd_projection(x_norm: ndarray, x_values: ndarray, y_values: ndarray, fuzzy_membership: array-like)`
  Performs SVD on normalized data and visualizes the 2D projection and approximations.

- `plot_metric_vs_param(df: pandas DataFrame, metric: str, param: str, title: str, xlabel: str)`
  Plots a clustering metric (NMI or Modularity) versus a parameter (sigma or k).

- `map_quality(quality: int or float)`
  Maps a numerical wine quality score to a categorical label.

Note: The code responsible for running experiments with varying parameters and plotting the results is implemented outside of function bodies in both the Fuzzy and IAP Clustering files. This script-based code serves the same purpose and functionality as the `run_spectral_experiments` function, which tests the algorithm across different parameters, and the `run_full_experiment` function, which orchestrates the execution of relevant functions and the presentation of results.