Department of Mathematics
Bar-Ilan University

May 2020

Final Project: Wine Clustering

Itai Schwed 208616664
Aviv Weinstein 315140475

Supervisor: Prof. Yoram Louzoun

Contents

1 Introduction

1.1 Dataset

The world of oenology is a vast-ranging world. Many parameters affect the quality and type of wine. The dataset (Cortez et al., 2009) that we chose includes 6497 samples of wines from Portugal. The dataset is divided into two wine types, red wine with 4898 samples and white wine with 1599 samples. Each sample, both "red" and "white", is composed mostly of various physicochemical properties of wine, and also of wine color and quality:

- **fixed acidity** (g(tartaric acid)$/dm^3$) - The amount of non volatile acid in wine
- **volatile acidity** (g(acetic acid)$/dm^3$) - The steam distillable acids present in wine
- **citric acid** (g/dm^3) - The amount of citric acid in the wine
- **residual sugar** (g/dm^3) - The amount of sugar remaining after fermentation stops
- **chlorides** (g(sodium chloride)$/dm^3$) - The amount of chloride in the wine
- **free sulfur dioxide** (mg/dm^3) - A measure of the amount of SO^2 that is not bound to other molecules
- **total sulfur dioxide** (mg/dm^3) - A measure of both the free and bound forms of SO^2
- **density** (g/cm^3) - The density of the wine
- **pH** - A scale used to specify how acidic or basic (or alkaline) the wine is
- **sulphates** (g(potassium sulphate)$/dm^3$) - The amount of sulphate in the wine
- **alcohol** (% $vol.$) - The volume of alcohol in the wine
  - **wine color/type** - The color/type of the wine
  - **quality** (score between 0 and 10) - Median of at least 3 evaluations made by wine experts

The majority of features appearing in the dataset are continuous, excluding wine color/-type and quality, which are discrete. These discrete features have more significant potential for unsupervised classification, than the continuous ones. In the original (Cortez et al., 2009), the authors goal was to classify the wine's quality using supervised methods, but we chose a different perspective.

1.2 Goals

The first goal of our project is to successfully classify wine type based on other properties without any supervision. To achieve that, we examined the use of different kinds of clustering algorithms.

Our second goal is to detect anomalies, i.e., red wine samples with a physicochemical composition of white wine and vice versa. Thus, we used various anomaly detection algorithms over our above-mentioned clustering algorithms.

2 Data Analysis

The first step of our project will be to understand the data, including how do the features distribute, and what is the connection between the different features.

2.1 Visualization

Understanding the data must start with understanding the features. Here are some interesting graphs about various features distribution:

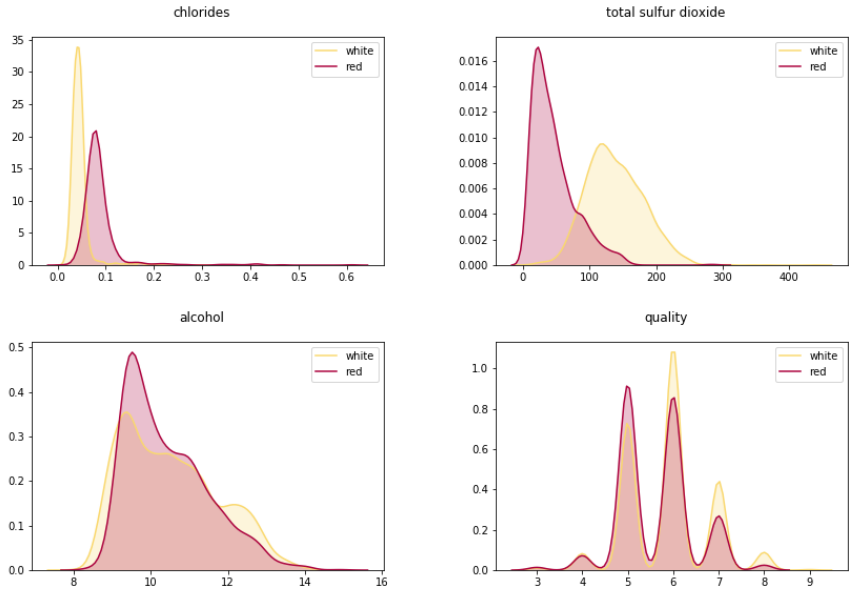

Figure 1: Features Distribution

From the graphs on figure 1 and similar graphs appropriate to other features, the following conclusions can be drawn. First of all, as we can see, most of the features are pseudo-normally distributed. Moreover, while on some of the features, there is a noticeable difference between the white and red samples, on others, there is almost no difference. This conclusion will guide us in the "Data Pre-processing" section.

Another way to compare the features related to wine type involves a boxplot graph, which is a method for graphically depicting groups of numerical data through their quartiles. The boxplots in figure 2 exhibit several more "important" features for our clustering, i.e., features that their values vary based on the wine type. For example, in the right boxplot, the "white" median is about 7, although the "red" median is 8.

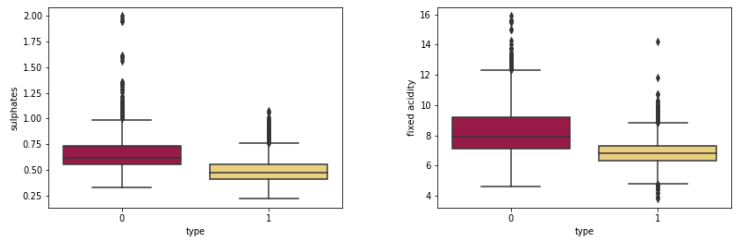

Figure 2: Additional Features Distribution

Finally, we used a dimension reduction algorithm, called PCA (Principal Component Analysis), for representing the feature vectors (excluding the wine type feature), both in 2D and 3D dimensional graphs. The two graphs in figure 2 present the resulted data points concerning the wine type.

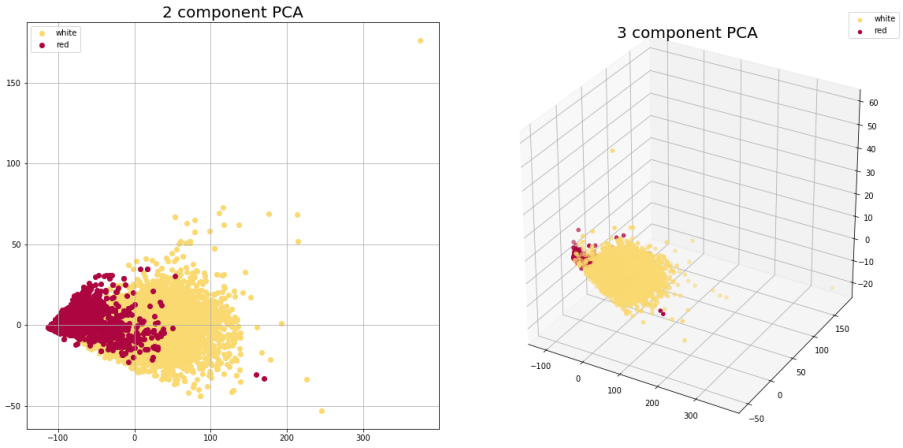

Figure 3: PCA Visualization

2.2 Feature Correlation

Once we have visualized the data, we can now explore the correlation between the features. We will use the Pearson Correlation Coefficient (PCC), which measures the strength of the linear relationship between any two variables.

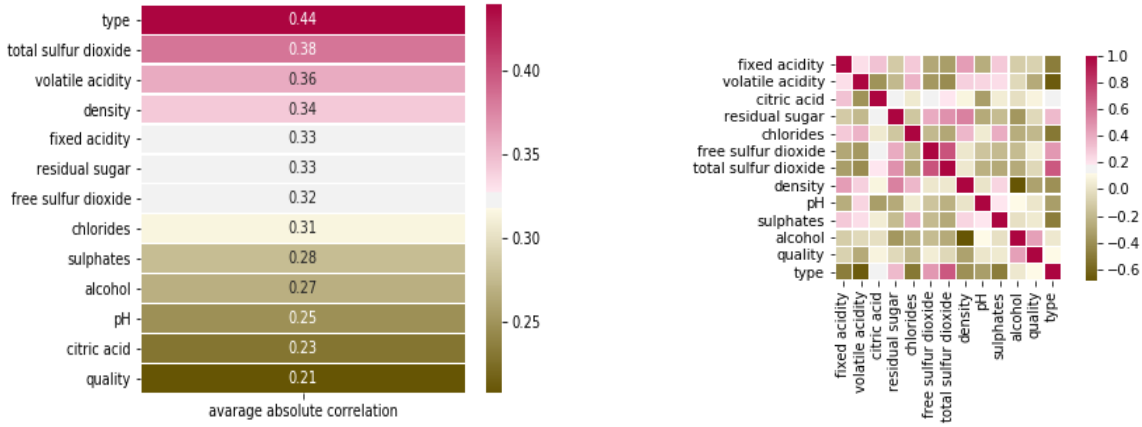

Figure 4: Features Correlation

As appears in figure 4, the total and free sulfur dioxide are strongly correlated. It is just natural since the first one is composed of the second one, among other components. However, most of the other features don't significantly correlate. One may also notice that the correlation between the quality feature and the rest of the properties is relatively insignificant in comparison to the correlation between the type feature and the rest of the features which is much higher, based on the average correlation (as illustrated in the left chart of figure 4). Therefore, this observation reinforces our decision to predict the type of wine rather than the quality of wine.

2.3 Data Pre-processing

In this section, we will describe our pre-processing procedure, based on the knowledge which was gained from the dataset so far.

When cleaning any dataset, the first stage is to check for NaN values, i.e., missing values, in any of the features. Our dataset was complete with valid value for every feature in each sample. Another thing to consider is balancing between the target labels in the dataset. Fortunately, the target labels, white and red for the feature wine type, were balanced enough for our purposes as will be seen in the following section. Here, the technical cleaning process ended.

Afterward, we dived more deeply into the influence of each feature on our classification problem. Considering the graphs in figure 1, we can observe that the features of alcohol and quality are very similar for the two wine types, red and white. According to that, we concluded that those two features would have a relatively small effect on the type classification compared to other features. Thus, those features are considered redundant in our project. Moreover, considering the features free and total sulfur dioxide, which we have shown to have a substantial correlation, we inferred that only the more informative feature should remain. In the "Clustering" section, we will determine whether or not these decisions where justified.

Finally, we will standardize each feature in the dataset based on its values, such that the features will be normally distributed as $N(0, 1)$. Although the features are already pseudo-normally distributed, as mentioned before, the standardization is used nevertheless for two reasons. First, the features in the dataset have different ranges, which may cause an undesired effect on a distance-based clustering algorithm, like k-means and spectral clustering. Second, we wanted to ensure that each feature will be distributed as a normal variable, since it is required for the k-means algorithm, and it is also recommended for other clustering algorithms.

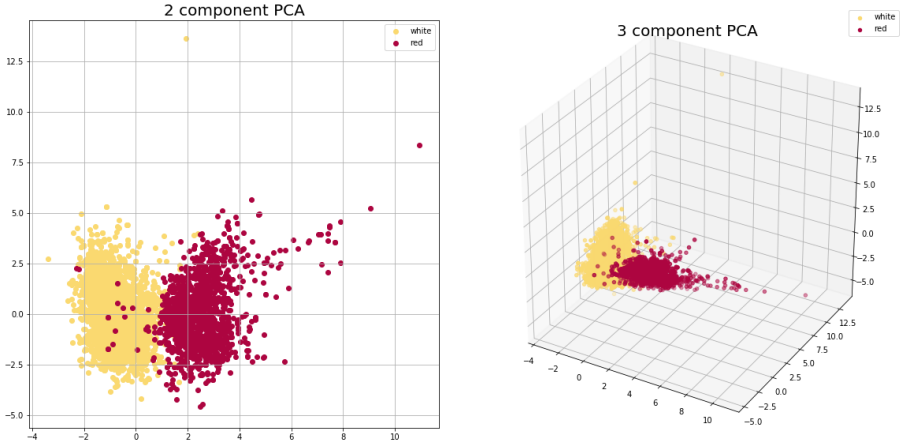

Figure 5: PCA Visualization after Performing Pre-processing

In the resulted representation in figure 5, the white and red wines can be easily interpreted as two different clusters, which are now more separated than has been seen before, in figure 3. It suggests that separated clusters may appear at higher dimensions as well, and it motivates our clustering goal. In order to achieve that goal, those graphs will be used as a ground truth reference in the "Clustering" section.

The pre-processing procedure ended here with 9 remained features. Theoretically, we could have also removed outlier samples from the dataset, which can be interpreted as the

points with feature values distributed well outside the bulk of values around the median (the black marks in figure 2). However, we elected to avoid that in the pre-processing stage, for several reasons. First, we avoided removing data points before performing clustering, due to the relatively small given dataset. Additionally, this phenomenon will be discussed in the "Anomaly Detection" section, where we will operate more complex algorithms over the clustering results, aiming to detect some interesting anomalies in the dataset. There, we will also test the success of particular clustering algorithms after removing the outlier samples.

3 Clustering

In this section, we will present several clustering algorithms and compare their performance after data pre-processing. The clustering algorithms are sorted here based on their success in classifying the wine type, starting from the least successful one. For each algorithm, we will discuss the most suitable number of clusters according to different evaluation metrics. Then, we will compare the algorithm's predictions and the ground truth in more depth, especially when the number of predicted clusters is the same as the number of labels in the ground truth, meaning two labels, "red" and "white". The comparison will be done using evaluation metrics, which will be applied over the data as a whole, and over different test-sets (Cross-Validation). Our results will be conveyed with p-value, which will support the significance of the results (all the p-values were calculated using $10k$ shuffling of the ground truth labels).

3.1 Gaussian Mixture Model

As we reach the clustering stage, the main goal is to adjust an algorithm that suits the data by its properties, such as variance, distribution, etc. Considering the PCA representation in figure 5, it appears that each cluster distributes pseudo-normally, which motivated us to fit our data using normally distributed models, such as *Gaussian Mixture Model (GMM)*.

Specifically for the GMM model, a normalization was applied, which transform each sample vector x into a vector with $\|x\| = 1$, over the resulted vectors in the pre-processing procedure. This decision wasn't derived from any guiding principle, but it slightly improved our GMM results, from 0.69 NMI to 0.79 NMI and from 0.93 F1 to 0.96 F1.

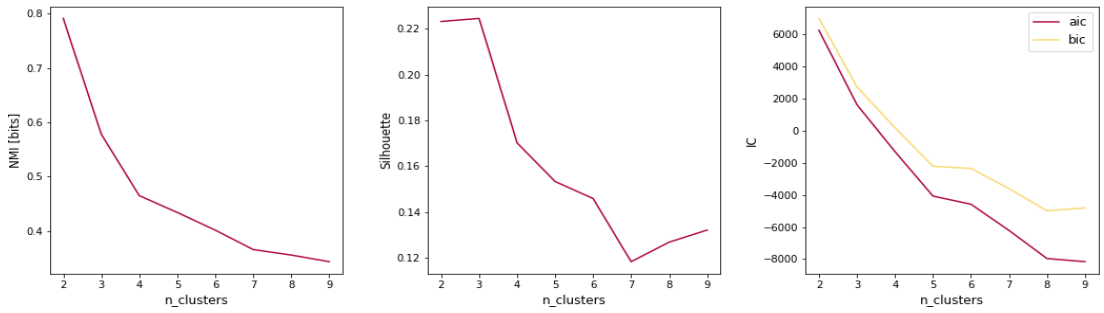

Figure 6: GMM - Scores as function of number of clusters

First of all, we searched for the number of clusters in which the EM (Expectation-Maximization) algorithm can separate most effectively, with a GMM model as a prior.

The algorithm was fed with $n\_clusters \in \{2, ..., 9\}$, and its results were evaluated based on various metrics, Silhouette, NMI (normalized mutual information), and IC (information criterion). According to the results in figure 6, **with respect to the ground truth labels**, the best value is $n\_cluster = 2$, since the NMI decreased dramatically when $n\_cluster > 2$. However, **with no respect to the ground truth labels**, the value $n\_clusters = 4$ is preferred, based on the ELBOW method for the IC scores, and the silhouette value that is still relatively high.

Next, we constrained the number of clusters to be $n\_cluster = 2$ and we ran GMM algorithm. The outcome clusters are shown in figure 7 with a comparison to the ground truth labels, and the evaluation metrics that we used are shown in table 1.

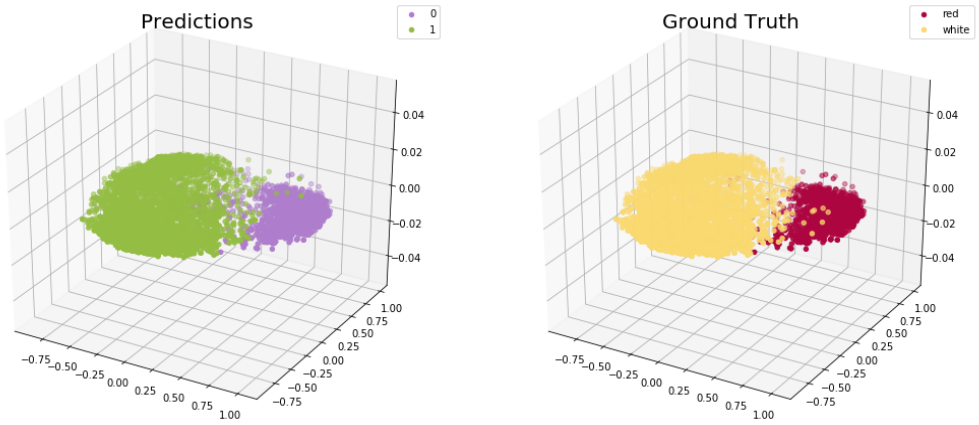

Figure 7: GMM Algorithm Results

	All Data	10-Fold Cross-Validation
Silhouette	0.2231	-
NMI	0.7911*	-
Accuracy	0.9723*	0.9721
F1	0.9634*	0.9631

	RED	WHITE
0	1559	40
1	140	4758

Table 1: GMM Scores [* p-value < 0.0001]

3.2 K-Means

Consequently to the success of GMM, an attempt with the K-Means algorithm was inevitable. The spherical-like clusters of the ground truth labels (shown in figure 5), also supported the potential of the k-means model. Furthermore, we calculated the variance of the ground truth of each target label, and discovered that they were remarkably similar. Thus, we have hoped that a centroid-based algorithm will perform even better.

It is essential to notice that k-means algorithm depends on the distances between the samples, so it will be incorrect to normalize the data. This is why this time we used the standardized data, which is crucial when using Euclidean distance as k-means does, without further normalization. As in previous attempt, we tried to find the best number of clusters, according to the graphs in figure 8. The best parameter **with respect to the ground truth** is definitely $n\_clusters = 2$, based on the NMI, similarly to GMM.

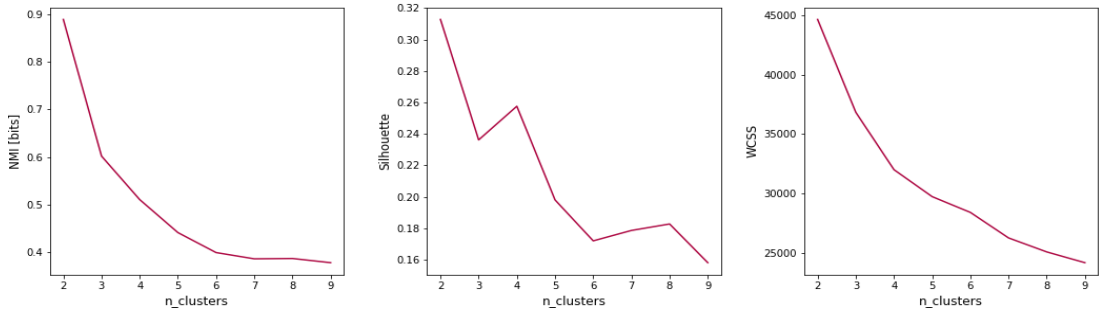

Figure 8: K-Means - Scores as function of numbers of clusters

In addition, **with no respect to the ground truth** $n\_clusters = 4$ value is preferred once again, based on the ELBOW of WCSS (Within Cluster Sum of Squares) and the improvement in the silhouette graph after $n\_cluster = 3$.

As we predicted, the k-means indeed achieved better scores than the previous model, as shown in table 2.

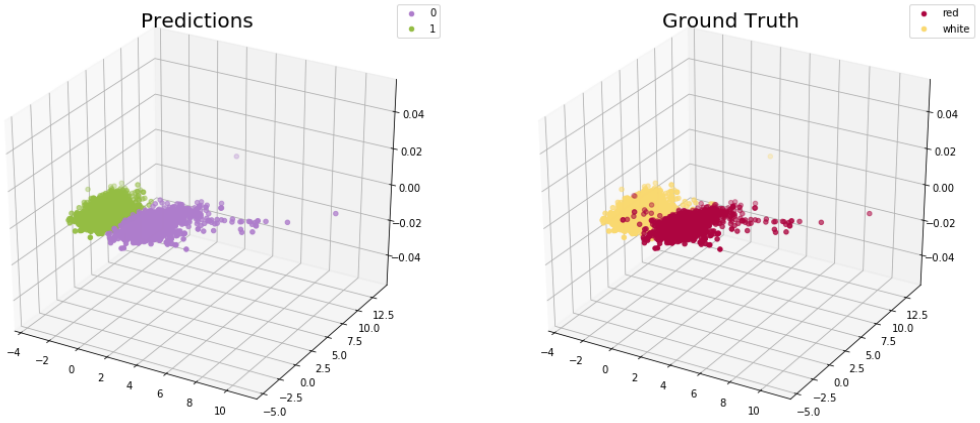

Figure 9: K-Means Algorithm Results

	All Data	10-Fold Cross-Validation
Silhouette	0.3128	-
NMI	0.8888*	-
Accuracy	0.9880*	0.9881
F1	0.9870*	0.9841

	RED	WHITE
0	1575	24
1	54	4844

Table 2: K-means Scores [* p-value < 0.0001]

3.3 Spectral Clustering

Another interesting algorithm that we attempted to apply to our problem is *Spectral Clustering*. The spectral algorithm, which was initially intended for cluster problems with exactly two labels, seemed to have a high potential for solving our problem. The overall results of the algorithm were the best ones that we achieved in this project, with better outcomes in most of the evaluation scores.

The spectral clustering partitioned given connected graphs into separated graphs. Our dataset is composed of data samples that were represented as vectors; in order to use this algorithm, we had to generate a graph from these vectors. We started with a simple graph that contained the distances between any two vectors, which resulted with extremely high running time. This is why, we chose to use a distance graph over the k-nearest neighbors and received the following results.

Similarly to the algorithms mentioned before, the spectral algorithm can also determine the number of most suitable number of clusters, which is represented by the data; Whenever more than two clusters are required, the algorithm uses a hierarchical clustering technique. However, as opposed to previous algorithms, there is another hyper-parameter that affects the performance of the algorithm, which is the number of neighbors that were used to generate the graph. Based on the resulted 3-dimensional graph figure 10, we can conclude that **with respect to the ground truth labels**, it is best to use $n\_cluster = 2$ and $n\_neighbors >= 8$ (for $n\_neighbors < 8$ the knn graph isn't connected). One may also notice that the number of clusters had a much greater effect on the evaluated scores, than the number of neighbors.

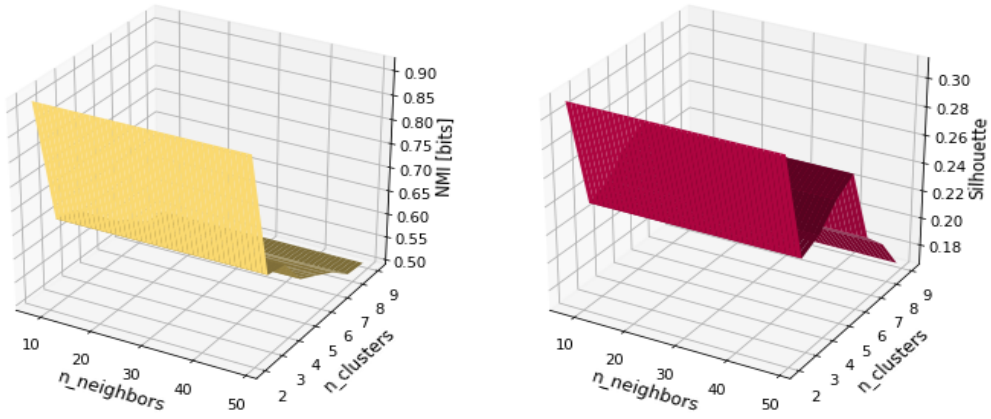

Figure 10: Spectral Clustering - Scores as function of hyper parameters

Hence, we evaluated the spectral model on our data when $n\_cluster = 2$ and compared it to the ground truth labels. Many options for the number of nearest neighbors were examined, and best results were achieved for $n\_neighbors = 8$, which are summed up in table 3.

	All Data	10-Fold Cross-Validation
Silhouette	0.3111	-
NMI	0.9177*	-
Accuracy	0.9918*	0.9881
F1	0.9890*	0.9854

	RED	WHITE
0	1577	22
1	31	4867

Table 3: Spectral Clustering Scores [* p-value < 0.0001]

After achieving such distinctive results, we also wanted to check whether or not the pre-processing of the data improved the results as we thought it would. Therefore, we performed the spectral algorithm again on the original representation of the data and received a much worse outcome: Silhouette=0.5208, NMI=0.5183, accuracy=0.919, f1-

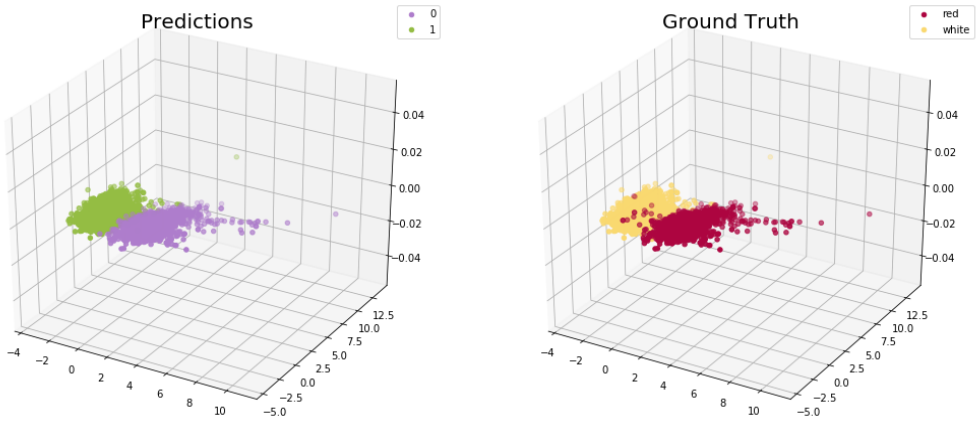

Figure 11: Spectral Clustering Algorithm Results

score=0.89. Even though the silhouette score increased; all the other scores that are compared with the ground truth labels were dramatically decreased.

As was explained in the beginning of this section, we calculated the *p-value* of each score to contradict the H_0 hypothesis that our results are accidental. For instance, we calculated the NMI of our predictions over 10k randomly shuffled ground truth labels as appears in figure 12. Then, we compared our NMI score (0.9881 for spectral clustering) to the other scores, and counted the number of times (if any) that the "random" NMI was higher than the original NMI. Under the common assumption of p-value threshold of 0.05, we can definitely state that H_0 is contradicted and the results are considered significant. This process has also been repeated for all other algorithms as well in order to retrieve their p-values.

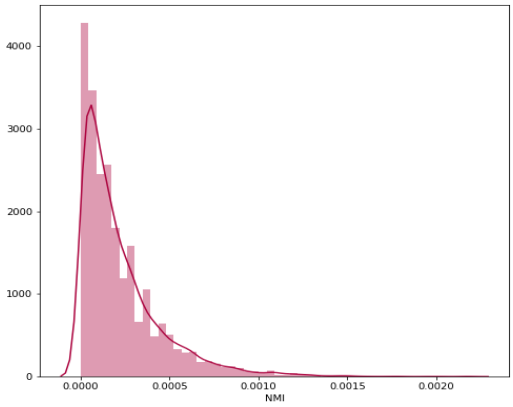

Figure 12: Spectral Clustering P-value Histogram for NMI

3.4 Other Algorithms

During our research, we tried to run several other algorithms, including *Fuzzy C-Means*, *DBSCAN*, etc. Those algorithms achieved poor results after testing many hyper-parameters, and hence, we avoided elaborating on them.

4 Anomaly Detection

One of our goals in this project as mentioned in the introduction is to detect anomalies on top of the clustering algorithms described in the last section. Since the anomaly detection subject constituted a tiny fraction of the course, we will briefly outline our results here.

In order to detect anomalies, we used the silhouette metric, which was computed for each sample in the dataset. The silhouette value of a sample is a number between -1 and 1, where 1 is considered as the best match of a sample to its target cluster, and -1 is considered as worst match. Thus, the aim of any clustering problem is, among other things, to get higher silhouette values for all the samples. However, there will always be some problematic data points that have the smallest silhouette. These points aren't matched well with their suitable clusters, meaning they are considered *anomalies*.

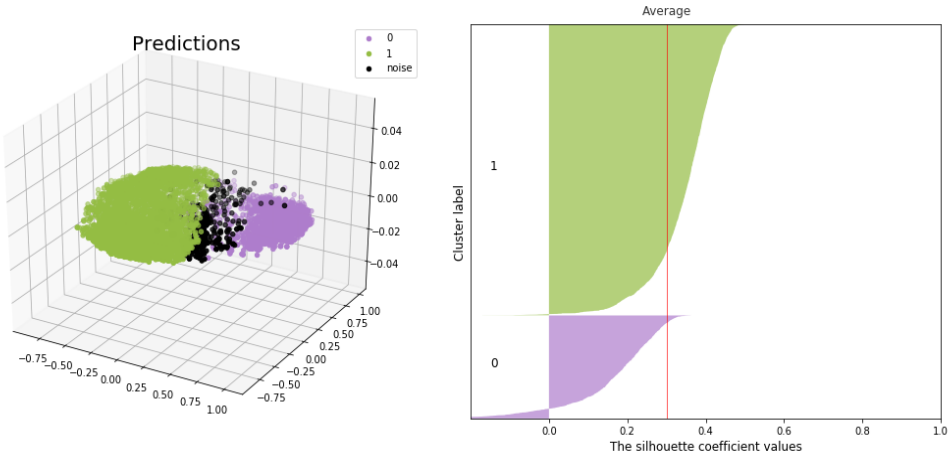

Figure 13: Spectral Clustering Silhouette-based Anomalies

The distinctive silhouette threshold value, for determining whether a sample is an anomaly or not, that we used is a commonly used one, 0.0. We used the same value for the clustering algorithms that we mentioned above and received 53 anomalies for spectral clustering, 65 for k-means, and 339 for GMM. As expected, the number of anomalies are higher as the algorithm performance is worsen. In figure 13, the anomalies are marked as noise for the algorithm GMM. Those anomalies, all mutually appear between the clusters (at least after PCA), which is not surprising, since that area is usually the hardest to classify correctly. The anomalies may refer to wine samples from one type that have properties which are more suitable for wine samples of other type, but we left that further exploration and conclusions out of the bounds of our work.

5 Conclusion

In this project we presented a dataset of physicochemical features of wine samples, and have explained our goals to cluster the wine samples by their type and to find anomalies. We showed the distribution of the aforementioned features and drew practical conclusions based on them. The preliminary stage was to decide which features can be disposed of, and what pre-processing techniques we should perform over our dataset.

Afterward, we applied different clustering algorithms over the dataset, including EM according to GMM, k-means and Spectral Clustering, and evaluated them using silhou-

ette, NMI, accuracy and F1 (based on confusion matrix). We attempted to find the optimal number of clusters both **with and without respect to the ground truth**. The resulted evaluations for all the algorithms showed a decisively insurance in $n\_clusters = 2$ **with respect to the ground truth labels**. However, GMM and k-means algorithms determined that **with no respect to the ground truth labels** $n\_clusters = 4$ is preferred. From that, we can infer that the wine samples is divided by more than just the wine type, e.g. by the wine dryness, etc. Then, we compared the success of the algorithms specifically when $n\_clusters = 2$ (with respect to the ground truth labels) and concluded that for our purpose *Spectral Clustering Algorithm* achieved the best results.

Finally, we explored a specific anomaly detection technique that we chose, using the *Silhouette score*, and showed how it was applied over the aforementioned clustering algorithms.

In this project we tried to present our decision-making process from analyzing the dataset, to choosing the right clustering algorithms and achieving our goals.

References

Paulo Cortez, António Cerdeira, Fernando Almeida, Telmo Matos, and José Reis. 2009. Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, 47(4):547–553.