TÉCNICO LISBOA

STATISTICAL METHODS IN DATA MINING

INSTITUTO SUPERIOR TÉCNICO

1^{st} SEMESTER - $2021/2022$

2^{nd} Project

Group 6:

Catarina Alexandra Saleiro Rodrigues **(45%)**	92434
Cristiano José Monte Mendonça **(0%)**	102355
Filipa Barros Costa **(45%)**	92626
Matheus Monteiro Casagrandi **(0%)**	101763
Natalija Stanojlovic **(10%)**	101644

Professor:

Conceição Amado

February 4, 2022

Contents

1 Introduction

This project is a continuation of the first part. The initial stage of the study contains an analysis of the dataset including two different types of dry beans in order to build an appropriate supervised binary classification algorithm for predicting dry bean varieties. However, since one has abstracted from the idea of having a labeled dataset and is simply interested in the feature space, the problem is now viewed from a new perspective. This way, the goal is to see if there are any interesting patterns or relationships between the bean observations using clustering algorithms, which will lead to the conclusion that some observations are related in some way and hence belong together. Cluster Analysis is based on the idea of identifying groups (or clusters) of items that are as similar as possible while differing from observations from other groups.

To validate the resulting clusters, external assessment indices (Accuracy, Sensitivity, Specificity, Balanced Accuracy, Precision, F1-score) and the confusion matrices are used to quantify and validate the decisions taken. It is worth noting that one will be using the real beans classes to subsequently validate the discovered clusters.

The next step is to apply the two best classifiers from Part 1 to this data, treating the best clustering solution as the new "classes" of the response variable, and compare the classes predicted by classification algorithms when using the clustering data and the real data. Thus, in the first part of this project, a supervised learning study was performed, from which it was concluded that the classifiers that best fit the proposed dataset are: KNN and XGBoost. In this second part, an unsupervised evaluation of the problem will be performed by applying the following clustering methods: Hierarchical (Single Linkage, Complete Linkage, Average Linkage and Ward's Method), Partitioning (K-Means, K-Medoids), Density Based (DBSCAN), Graph Based (Mst-Knn), Spectral.

In this manner, one chooses a strategy to achieve objectives of the project. The datasets that will be used to run the clustering methods are described in section 2. Section 3 introduces the technique used for determining the appropriate clustering algorithm as well as the dataset that will be utilized for classification. In section 4, the approach for comparing the results produced from classifiers trained on real classes to classifiers trained on clustered classes is proposed. By section 5 one begin to describe each clustering method used and some of the results obtained, in section 6, one provide the final results of methods to answer our problem and and in the last section one can find the conclusions.

2 Datasets

In this section, the datasets that are considered throughout the project are presented. In the first place, it is worth noting that, in contrast to the first part of the project, it was made the decision to abandon the idea of using under and oversampling. In particular, studying the clustering behavior of the same observation does not make sense for oversampling because it will never be assigned to a new cluster, leaving only the original dataset.

Another thing to note is that two new datasets have been introduced to this second part. The first dataset came from a Pearson correlation analysis between the independent variables, in which the one with the highest mean absolute correlation was excluded from a pair of highly correlated variables.

The second dataset, which was obtained from a previous dataset, contained eight variables, each of which had a point-bisserial correlation of greater than 95% with the response variable. Now, among these atributes, it was determined to eliminate those with a pearson correlation of greater than 95%, resulting in only two variables: ShapeFactor1 and ShapeFactor2.

This makes sense since, when looking at the histograms from Part 1, one can see that these two variables have a high separability while also having a modest representation of both classes.

Consequently, this study is left with 5 different datasets:

- Dataset1 - Normalized, with all variables;
- Dataset2 - Normalized + Pearson correlation analysis between independent variables (among two highly correlated variables, the one with the largest mean absolute correlation was eliminated) **(new)**;

- Dataset3 - Normalized + Pearson correlation analysis between independent variables (among two highly correlated variables, the one with the lowest point-bisserial correlation with the response variable was eliminated);
- Dataset4 - Normalized + Point Bisserial correlation analysis between independent variables and response variable;
- Dataset5- Normalized + Point Bisserial + Pearson Correlation Analysis **(new)**.

3 Methodology to choose "The Best" Clustering Technique

In an ideal world, every clustering algorithm would be used to every dataset and the best one would be chosen. Due to the impossibility of this strategy, the following procedure was utilized to choose the best clustering technique:

1. Choose one dataset. In this case, it was decided to choose Dataset1, since it presented all variables.
2. Split the chosen dataset into training and test sets;
3. Apply the clustering methods to the training set, from which the following classes result $\hat{y}_{cluster}^{treino}$;
4. Assign the new labels, applying a distance, to the data in the test set, using clusters obtained in step 2, from which the following classes $\hat{y}_{cluster}^{test}$ resulted. The label corresponding to each data in the test set is the one whose observation has the smallest distance to the centroid of each cluster obtained in step 2;
5. It is feasible to choose the "Best" Clustering Method based on the findings reported in items (3) and (4) by assessing the performance measures on the train and test sets when compared to the true classes.
6. The foregoing items (2), (3), (4), (5) were run in all the datasets available using only the "Best" clustering algorithm to see which datasets perform best with this clustering method.
7. From this, it is possible to extract the best clustering method from the chosen dataset.

4 Methodology to compare the results from the classifier applied to the original labels vs the cluster labels

Following the selection of the "best" Clustering Technique, it is required to run the two best classifiers from Part 1 using the clustering classes and compare their performance to that of merely utilizing the original classes. The following list explains how this comparison is made:

- Conduct the clustering with the dataset and clustering method that produced the best results, but without splitting the dataset into train and test. As a result, the clustering labels for observations will emerge: $\hat{Y}_{cluster}$;
- Divide the dataset with the new classes $\hat{Y}_{cluster}$ into train and test sets;
- Train the 2 best classifiers, obtained in the first part of the project, with the training set;
- Apply the 2 best classifiers to the test set;
- Compare the results of applying the classifier to the dataset using the original classes against the cluster classes.

5 Clustering Methods

It was decided to use three techniques for clustering: Hierarchical Clustering, K-means, and K-medoids. Several approaches were used to determine the best number of clusters (k), including the elbow method, the average silhouette width, and the between and within sum of squares. Some indices, such as the Davies Bouldin-index, the Dunn-index, and the C-index were also investigated. Using the normalized dataset, the algorithms were computed for the values of $k = 2, ..., 7$. In general, the above approaches showed that k = 2 was the best one for Hierarquical Clustering, K-means and K-medoids (since the other methods, Dbscan, Mst-Knn and Spectral, do not allow the choice of this parameter), as it can be noticed in the Figure 1 below:

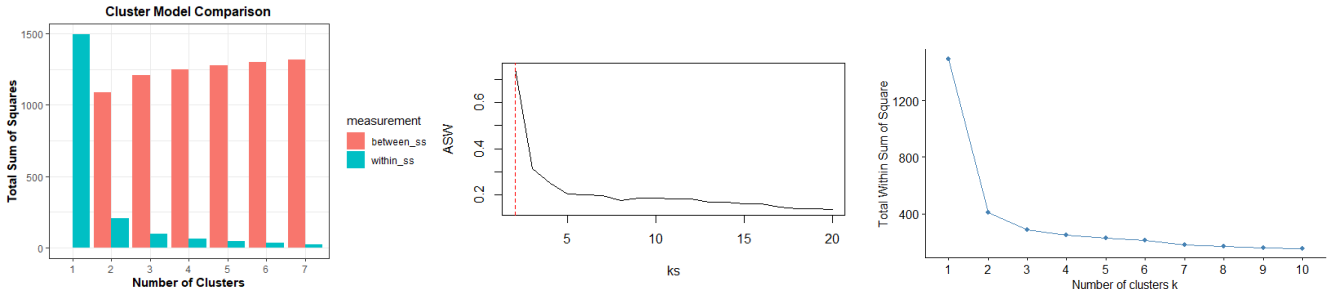

Figure 1: Number of Clusters vs Between-SS and Within-SS (for k-Means) on the left, Number of Clusters vs Average Silhouette Width (for Hierarquical Clustering with Ward's Method) in the middle, Number of Clusters vs Total Within-SS (for k-Medoids) in the right

It is easy to see in the Figure 1 above that between $k = 1$ to $k = 2$, the between-ss increases tremendously while the within-ss reduces significantly (in k-Means). This is no longer the case with the transition from $k = 2$ to $k = 3$. One can also observe that when going from $k = 1$ to $k = 2$, the total within-ss (in k-Medoids) and the average silhouette width (in Hierarchical Clustering) both fall considerably, but the opposite happens when going from $k = 2$ to $k = 3$, and an elbow appears. Considering these asssertions, and given the fact that the type of beans variable has two alternative outcomes (BOMBAY and DERMASON), these three algorithms were computed with k = 2 so that the clustering results could be externally validated. As a result, the choice of two clusters corresponds to the number of classes in the response variable under investigation. When it comes to Spectral Clustering, DBSCAN and Graph Cluster, the number of cluster where above two. Since one is working with continuous real variables, one attempted to conduct k-medoids with the Euclidian and Manhattan distances. Since the methods for determining the ideal k yielded similar results performing both distances, the Euclidian distance was chosen to perform the clustering algorithms.

5.1 Hierarquical Methods

For the Hierarquical Clustering, four different approaches were experimented, with the Euclidian distance: Single Linkage, Complete Linkage, Average Linkage and Ward's Method. These four hierarquical clustering techniques were performed in the normalized dataset to see which one better describes the data.

Table 1: Optimal number of clusters for Single Linkage and Complete Linkage, performed on the normalized train set, considering different internal indexes

Single Linkage			
Number of Clusters	**C-Index**	**Davies-Bouldin**	**Dunn-Index**
2	0.0014	0.4093	0.4335
3	0.0011	0.3445	0.3278
4	0.0011	0.3398	0.2559
5	0.0011	0.3321	0.2445
6	0.0011	0.3350	0.2154
7	0.0022	0.3863	0.1884

Complete Linkage			
Number of Clusters	**C-Index**	**Davies-Bouldin**	**Dunn-Index**
2	0.0014	0.4093	0.4335
3	0.0132	0.9522	0.0352
4	0.0809	1.2950	0.0189
5	0.0802	1.0720	0.0215
6	0.0797	1.1932	0.0230
7	0.0795	1.1219	0.0239

Table 2: Optimal number of clusters for Average Linkage and Ward's Method, performed on the normalized train set, considering different internal indexes

Average Linkage			
Number of Clusters	**C-Index**	**Davies-Bouldin**	**Dunn-Index**
2	0.0014	0.4093	0.4335
3	0.0011	0.3445	0.3278
4	0.0011	0.4793	0.2154
5	0.0012	0.5074	0.0909
6	0.0220	0.7585	0.0597
7	0.0228	0.9570	0.0597

Ward's Method			
Number of Clusters	**C-Index**	**Davies-Bouldin**	**Dunn-Index**
2	0.0014	0.4093	0.4335
3	0.0818	1.0111	0.0190
4	0.0726	1.1543	0.0190
5	0.0662	1.3207	0.0241
6	0.0580	1.3832	0.0241
7	0.0535	1.3213	0.0241

Single Linkage and Average Linkage can be eliminated from the tables 1 and 2 above since, according to the Davies-Bouldin index, $k = 3$ provides the best number of clusters. However, the best number of clusters, according to the Dunn-Index, using the same approaches is $k = 2$. As a result, the findings reached are inconsistent, and since the optimal k for the remaining clustering methods is $k = 2$, the only option is to choose between Complete Linkage and Ward's Method.

The graph 2, which examines the average silhouette width as a function of the number of clusters, was used to make this decision:

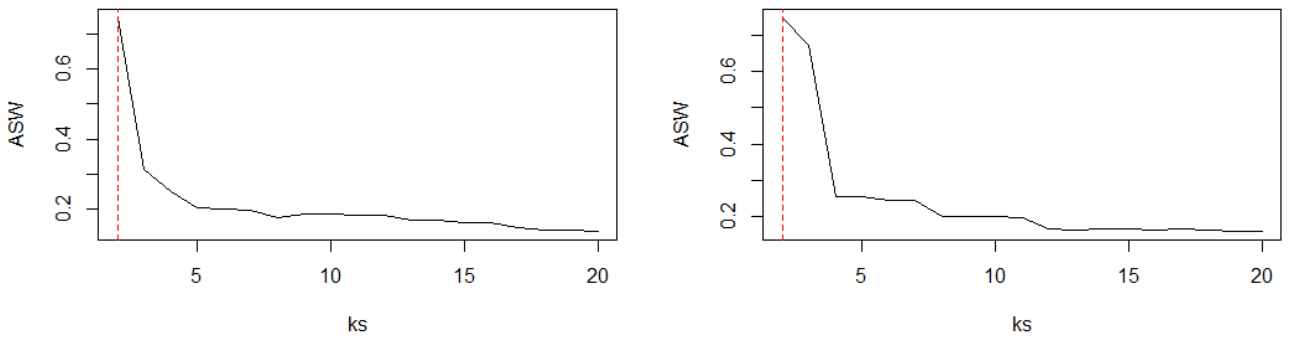

Figure 2: Number of Clusters vs Average Silhouette Width (for Hierarquical Clustering with Ward's Method) (left). Number of Clusters vs Average Silhouette Width (for Hierarquical Clustering with Complete Linkage) (right)

From the graphs above 2, the Ward's approach is the one with the biggest decline between $k = 2$ and $k = 3$ in the Average Silhouette Width. As a result, Ward's Method was picked as the best Hierarchical Clustering algorithm.

Table 3: Performance Measures for the normalized dataset, with 16 variables

Performance Measures - 16 variables				
Normalization	**Train data**		**Test data**	
Classes	*BOMBAY*	*DERMASON*	*BOMBAY*	*DERMASON*
Cluster 1	423	0	99	0
Cluster 2	0	2831	0	715
Accuracy	1.000		1.000	
Sensitivity	1.000		1.000	
Specificity	1.000		1.000	
Balanced Accuracy	1.000		1.000	
Precision	1.000		1.000	
F1-Score	1.000		1.000	
Entropy	0.3864		-	

As it is noticeable in the confusion matrices displayed in Table 3, the Ward's Method did not miss-classified any observation and performed exceptionally well across all performance metrics (for example, it got an accuracy of 100% on both train and test sets). Finally, when considering the normalized dataset and choosing the Ward's Method, a dendogram was constructed as shown in Figure 3:

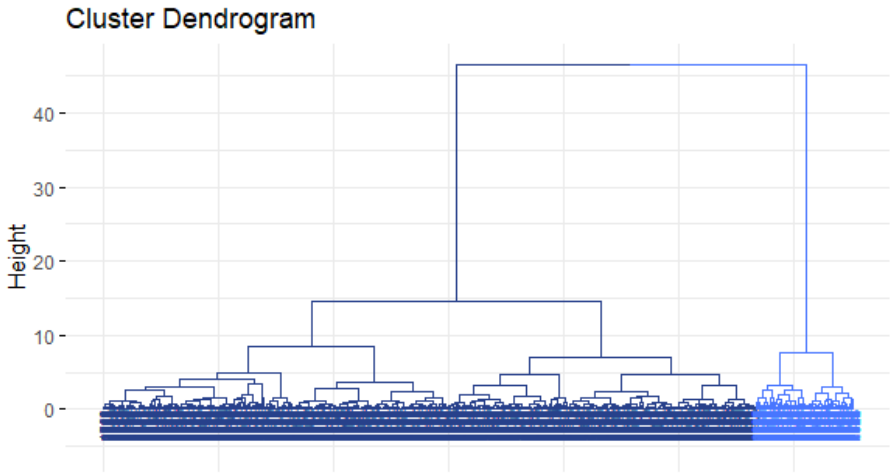

Figure 3: Dendogram for the Normalized Dataset with Ward's Method

In this dendogram, the dark blue part corresponds to the cluster of the DERMASON class and the light blue part to the cluster of the BOMBAY class. It is possible to notice that all the observations were correctly assigned.

5.2 Partitioning Methods (k-Means and k-Medoids)

A cluster is represented by a center vector or centroid in partitioning clustering, which can be a position that was not observed. Typically, this is accomplished using the k-means approach, which divides the data into k clusters with the condition that each observation within each cluster is closer to that centroid than the centroid of another cluster. The primary difference between k-Means and k-Medoids is that in the first, the centroid is determined by the euclidian distance, and the centroid must be the mean, whereas in the second, the centroid can be the median (as is gonna be considered in this project), or another location measure. In either instance, the clusters k value must be chosen in advance. Figure 1 depicts the decision-making process. All three indices agree that k = 2 is the best number of clusters in k-means and k-medoids. The silhouette is plotted as a function of the number of clusters considered in Figure 4. For this, a cycle was run from k = 2 to k = 7, with the greatest total Within-SS being recorded. Similarly, two clusters is the optimal result.

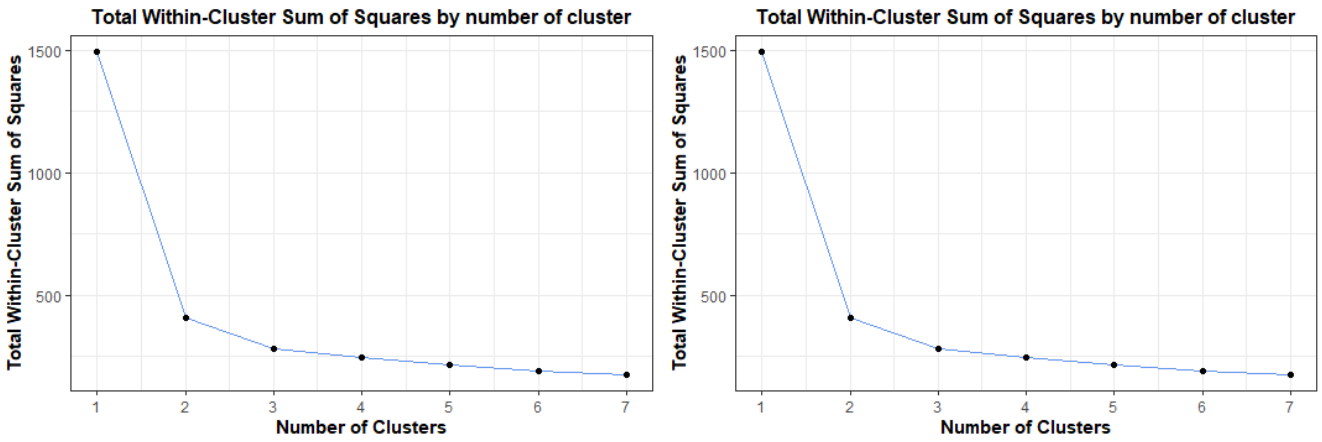

Figure 4: Number of Clusters vs Total Within-SS (for k-Means) (left). Number of Clusters vs Total Within-SS (for k-Medoids) (right)

Finally, when looking at the performance measures, the result is the same as the ones when applying Ward's Method 3. As it can be observed, there was none observation assigned improperly to the cluster where they should be. One also analysed the accuracy values for each clustering technique and the accuracy for both methods is 100%. Since k-Medoids has a complexity of O(N2CT), where N, C, and T denote the number of data points, clusters, and iterations, respectively, while k-Means has a complexity of O(NCT) [1], k-Means is better suitable for larger datasets.

5.3 DBSCAN

DBSCAN is an efficient technique for vast amounts of spatial data that detects clusters in arbitrary ways. As a result, it entails establishing a set of clusters that represents the underlying density of observations, i.e. grouping points that are relatively close to each other into a cluster so that each cluster is a "dense" subset of the point cloud. Core points are observations that have a minimum number of points (MinPts) within a radius (Eps) of a certain radius. Thus, it is important to define the minimum number of points (MinPts) within a given radius (Eps), in the beginning.

In order to get the best value of Eps, the average of all points' distances from their k nearest neighbours was calculated and graphed in ascending order. Following this idea, the optimal value of Eps is found in the elbow of the graph in Figure 5. The MinPts value was set to be the number of variables+1 [2].

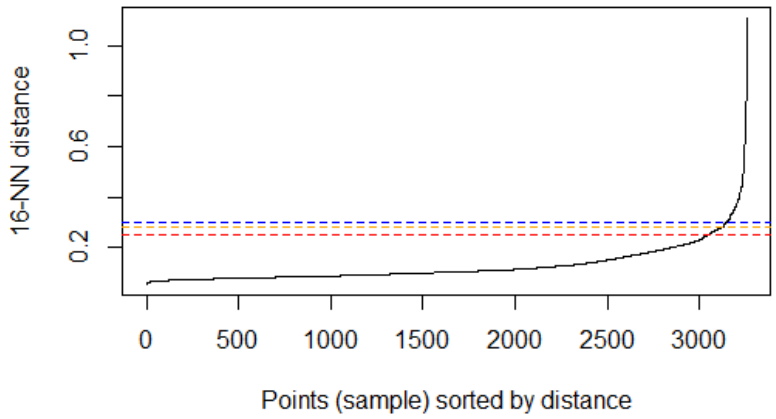

Figure 5: Average distance of all points to its 16 nearest neighbors, ordered in ascending order. Eps=0.3 (in blue) ,Eps=0.28 (in red), Eps=0.25 (in orange)

According to Figure 5, one can conclude that the optimal value of Eps is around 0.3.
After determining the initial parameters, the number of clusters is determined automatically.
Figure 6 depicts the results obtained from applying the DBSCAN algorithm:

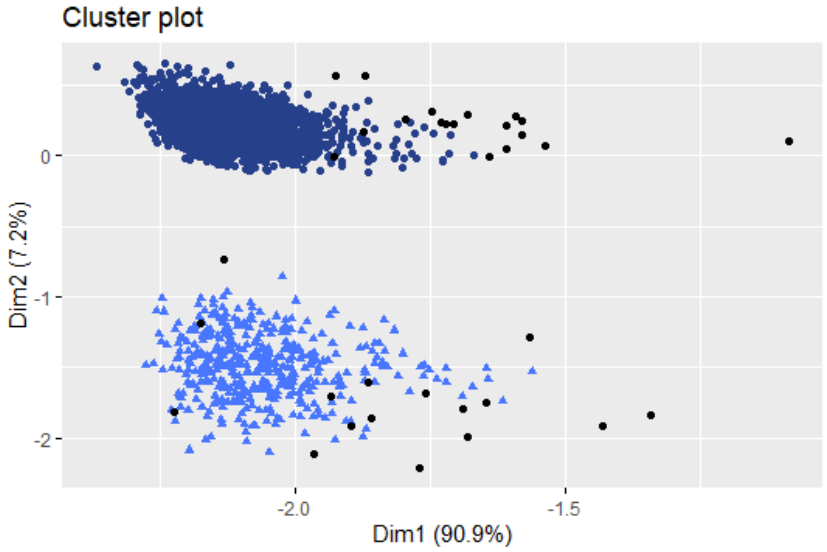

Figure 6: Cluster Plot resulting from the application of the DBSCAN method, using the 2 principal components (Dim1 and Dim2) that represent the highest variability

According to this algorithm, the resulting number of clusters was 2, however, there were 34 points (in black) that were not clustered at all. The points in dark blue are refering to the DERMASON clsutering class, while the ones in light blue are refering to BOMBAY. Therefore, since the methods previously presented have a better performance, this algorithm

will not be used in subsection 5.6.

5.4 Graph Based

The MST-KNN technique is a weighted graph algorithm that defines each vertex as an observation and each edge as the weight determined by the dissimilarity of the two vertices (observations) it connects. The purpose is to discover vertices clusters. This entails calculating the graph's spanning tree with the smallest weight - minimum spanning tree (MST). Also known as the KNN graph, this graph connects the vertex a to the vertex b if and only if b is one of a's k-nearest observations. Again, the selection of the number of clusters is automatic, because k is the smallest number of neighbors required for the KNN graph to be connected.

The results using this type of clustering were not so good since it resulted in 37 different clusters and so it will not be used in subsection 5.6.

5.5 Spectral Methods

The Spectral clustering method uses the eigenvalues of data's own similarity matrix to perform dimensionality reduction before clustering.

The Spectrum method of the Spectrum package was applied and 3 clusters were obtained, that are represented in Figure 7.

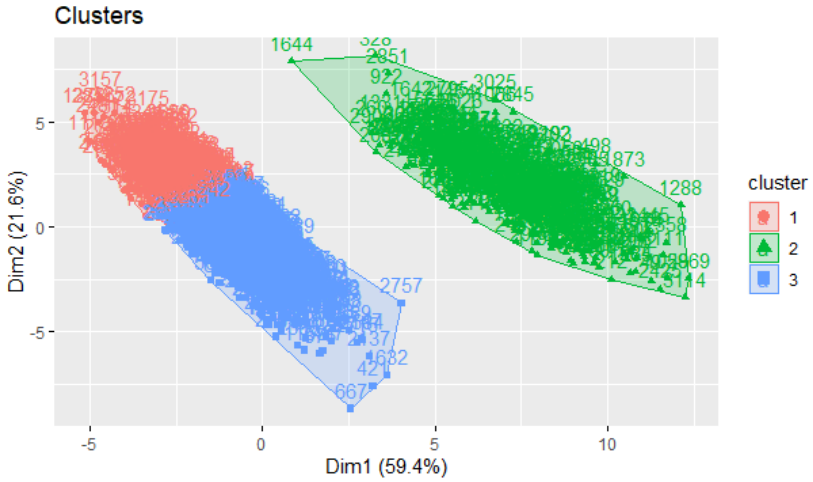

Figure 7: Cluster Plot resulting from the application of the Spectral Clustering method, using the 2 principal components (Dim1 and Dim2) that represent the highest variability

Evidently, the cluster with the largest(2111) (smallest (423)) size in blue (green), corresponds to the DERMASON (BOMBAY) class, respectively. Thus, it is possible to conclude that there were 720 observations from DERMASON class that were misclassified, since the smallest cluster has the same number of observations (423) that the BOMBAY class has in the training set. This way, since the other clustering algorithms performed better, this will not be considered in subsection 5.6.

5.6 "The Best" Clustering Result

For the selected methods, k-Means, k-Medoids and Ward's Method presented the same performance measures for the normalized dataset. However, since k-Means is a less computational complex algorithm (and thus, faster), it was

chosen as the best classifier to perform with the remaining datasets. After applying K-means to the remaining datasets, all of the datasets from section 2 produced identical results (100% in every performance metric and perfect confusion matrices), strongly implying that Dataset 5 produces the best results because it only has two variables. This indicates that not all variables are relevant to the model, and that ShapeFactor1 and ShapeFactor2 are crucial in identifying the type of beans.

Because of this, in Table 4, one will only present the performance measures when the Dataset5 was used.

Table 4: Performance Measures for Dataset5, with 2 variables

Performance Measures - 2 variables				
-	**Train data**		**Test data**	
Classes	*BOMBAY*	*DERMASON*	*BOMBAY*	*DERMASON*
Cluster 1	423	0	99	0
Cluster 2	0	2831	0	715
Accuracy	1.000		1.000	
Sensitivity	1.000		1.000	
Specificity	1.000		1.000	
Balanced Accuracy	1.000		1.000	
Precision	1.000		1.000	
F1-Score	1.000		1.000	
Entropy	0.3864		-	

As it be can seen in the Graph 8 below, the elbow method is represented with the Dataset5. Through the graph one can conclude that the optimal k value is indeed two (as was said at the beginning of this chapter).

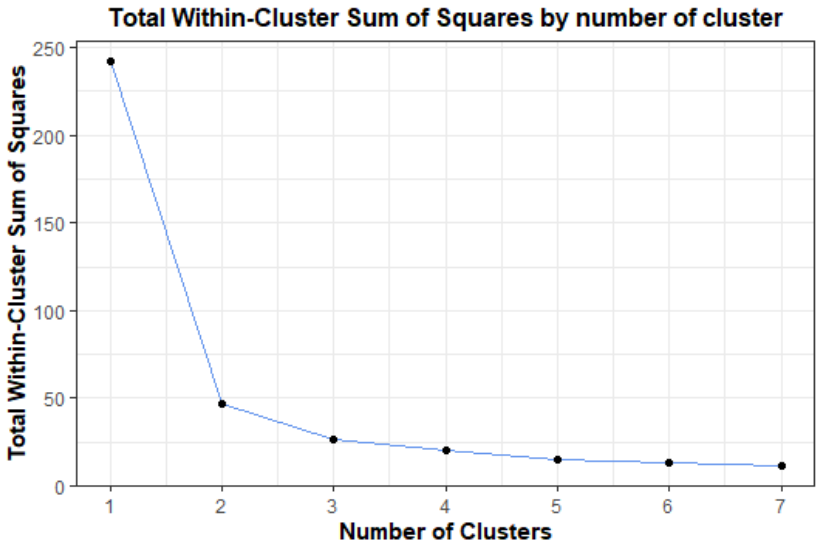

Figure 8: Elbow Method for optimal value of k in K-Means for the Dataset5

Luckily, when eliminating variables based on Point Bisserial Correlation, not only were achieved the best clustering results of all the preprocessed datasets but also one was able to obtain a dataset with only two explanatory variables which meant that it is possible to obtain a 2D plot of the data (Figure 9).

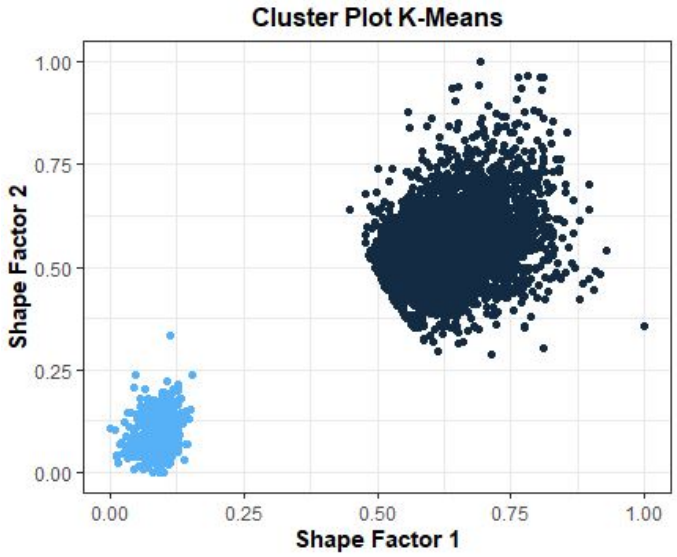

Figure 9: 2D scatter plot of the clustering results in Dataset5, using k-Means

The DERMASON beans are represented by dark blue points, whereas the BOMBAY beans are represented by light blue points. Because no misclassification occurred, it is clear that there is no overlap between the two categories of beans. Finally, hierarchical clustering methods can be used to build a dendogram. When considering the Dataset5 and choosing the Ward's Method, we were able to obtain a dendogram, shown in Figure 10 that, for our amusement, correctly classified all the observations.

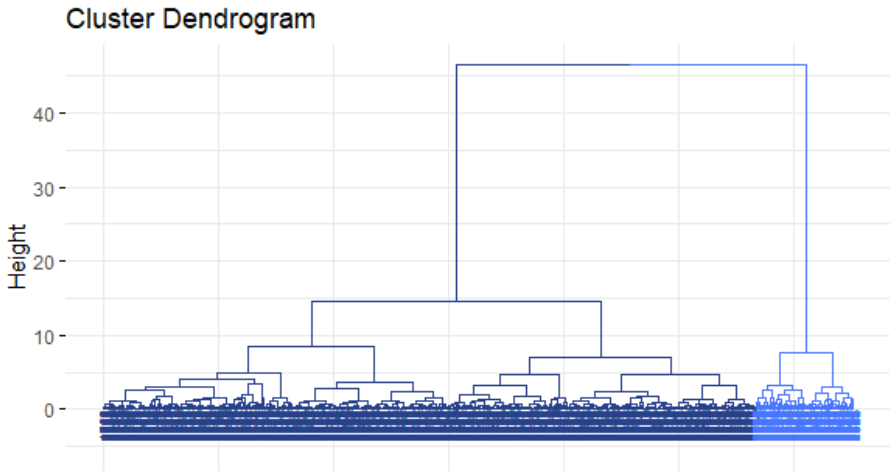

Figure 10: Dendogram for the Dataset5 with Ward's Method

When analysing these results, the best explanation for this incredible results with this dataset is again that only two variables are present in this dataset which "cleans" the space for clustering methods.

6 Supervised Learning with Clustering Results

In this section one applied the two best classifiers from Part 1 (KNN and XGBoost) to the data, using the best clustering solution as the new "classes" of the response variable, and comparing the classes predicted by classification algorithms using clustering (in this case `ClusterClass`) data and real data (`TrueClass`).

6.1 KNN

KNN (K-Nearest Neighbors) is a supervised learning that examines the labels of a chosen number of data points surrounding a target data point, in order to make a prediction about the class that the data point falls into. In part 1 it was already seen that the best values were obtained with $k = 1$ and with Manhattan metric. The number $k = 1$ was chosen because small and large k produce overfitting problems, according to the theory. Because the scores for each k in this scenario are all the same, a smaller k was chosen to simplify the model. Given that the Manhattan distance is the measure that leads to the highest performing classification (according to part 1), one looks at the different values of k for this distance and uses accuracy as a score because all of the scores had 100%, thus it was irrelevant to use another score.

Once again, the optimal k value was 1. The plot is shown below for the TrueClass dataset:

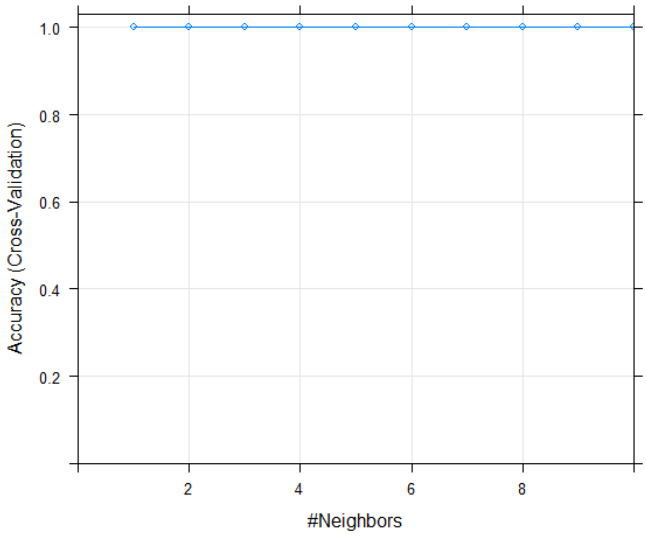

Figure 11: KNN Classifier scores for different $k = \#Neighbors$ values for the `TrueClass` dataset

As it can be seen in 11 the accuracy is always 1.0 for the different k values. The same result was obtained for the ClusterClass dataset.

First, one applied the KNN classifier to the TrueClass dataset and the results obtained were as expected. In the table bellow are the prediction results obtained:

Performances Measures	
Model	KNN
Accuracy	1.000
Sensitivity/Recall	1.000
Specificity	1.000
Balanced Accuracy	1.000
Precision	1.000
F1-Score	1.000

Table 5: Performance Measures for KNN with $k = 1$ in the `TrueClass` dataset and `metric:Manhattan`.

$$\begin{bmatrix} 99 & 0 \\ 0 & 715 \end{bmatrix}$$

Figure 12: Confusion Matrix for KNN with `TrueClass` dataset

The table 5 shows that the classifier application resulted in a precision of 100% for both classes (DERMASON and BOMBAY), such as the recall and the F1-score, which means that the rate of true positives among all positives and the harmonic mean between precision and sensitivity is always 1.0, implying that all observations were correctly classified.

Aside from that, the accuracy of the macro average and the weighted average are both 1.0. Finally, by examining the confusion matrix, it is clear that no class has not been misclassified, implying that the KNN algorithm for the `TrueClass` has correctly predicted 100% of the test set observations (as shown in Figure 12), owing to the good results achieved.

The exactly same results were obtained for the ClusterClass dataset. In conclusion, the results obtained with `ClusterClass` and `TrueClass` were the same so it can be concluded that, in this case, using cluster techniques is not necessary. Also in this part of the project one only used three variables (`ShapeFactor1`, `ShapeFactor2`, and TrueClass or `ClusterClass`) and the results obtained were also the same, in comparison with the first part of the project, therefore it can be also concluded that the variables removed were not necessary.

6.2 XGBoost

XGBoost is an implementation of Gradient Boosted decision trees. In this algorithm, decision trees are created in sequential form. Weights are assigned to all the independent variables which are then fed into the decision tree which predicts results. The weight of variables predicted wrong by the tree is increased and these variables are then fed to the second decision tree. These individual classifiers/predictors then ensemble to give a strong and more precise model. It can work on regression, classification, ranking, and user-defined prediction problems.

As was done with the KNN classifier, first, one applied the XGBoost to the TrueClass dataset. The performance measure results obtained are in the table 6:

Performances Measures	
Model	XGBoost
Accuracy	1.000
Sensitivity/Recall	1.000
Specificity	1.000
Balanced Accuracy	1.000

Table 6: Performance Measures for XGBoost in the `TrueClass` dataset.

$$\begin{bmatrix} 99 & 0 \\ 0 & 715 \end{bmatrix}$$

Figure 13: Confusion Matrix for XGBoost with `TrueClass` dataset

As with KNN, the classifier application resulted in a precision of 100% for both classes (DERMASON and BOMBAY). By examining the confusion matrix, it is clear that no class has not been misclassified, implying that the XGBoost algorithm for the `TrueClass` has correctly predicted 100% of the test set observations (as shown in Figure 13). The same results were obtained for the `ClusterClass` dataset.

The SHAP values were also calculated. SHAP (SHapley Additive explanations) assesses the influence of variables while accounting for their interactions with other variables. "Shapley values calculate the importance of a feature by comparing what a model predicts with and without the feature. However, since the order in which a model sees features can affect its predictions, this is done in every possible order, so that the features are fairly compared."[3] So, the SHAP values for the `TrueClass` dataset obtained were:

ShapeFactor1	ShapeFactor2
1.6973	0.0000

Table 7: SHAP values for the `TrueClass` dataset

In plots like the one in figure 14 the y-axis indicates the variable's name, in order of importance from top to bottom, so as it can be seen in the figure bellow, the `ShapeFactor1` is the most influence variable and the value next to them is the mean SHAP value, which is 1.6973 for `ShapeFactor1` and 0.000 for `ShapeFactor2`. On the x-axis, one can find the SHAP value, which indicates how much is the change in log-odds and from this number the probability of success can be extracted. Gradient color indicates the original value for that variable. In booleans, it will take two colors, but in number it can contain the whole spectrum. Each point represents a row from the original dataset. The plot obtained for the `ClusterCLass` dataset was exactly the same, as well as the shape values.

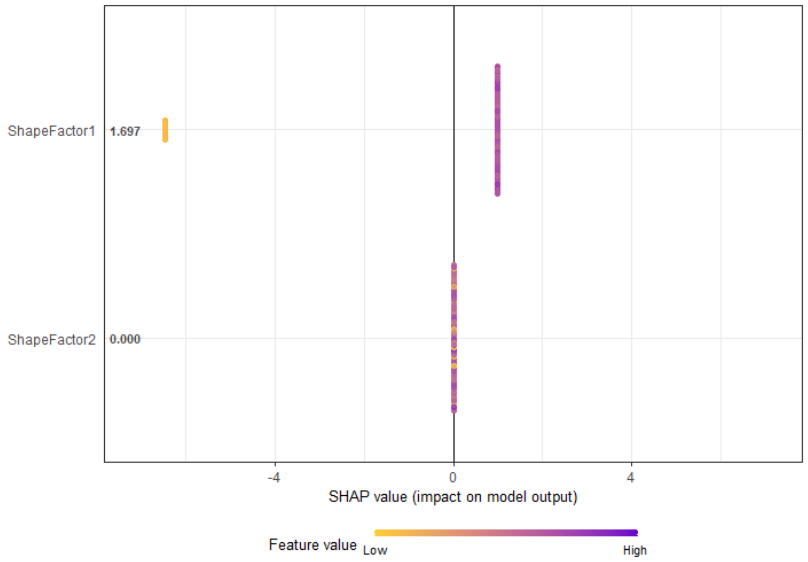

Figure 14: SHAP values for the `TrueClass` dataset

Concluding, the results obtained with `ClusterClass` and `TrueClass` were the same so, as said before in the section

of KNN, it can be concluded that using cluster techniques is dispensable and that the variables removed were not necessary. Regarding shap values, it can be seen that `ShapeFactor1` is the most important variable between the variables used. So having the model without the variable `ShapeFactor1` can affect all the predictions. The same is not true for `ShapeFactor2` because as seen before its shape value is 0.000, so this variable do not have influence in the model.

7 Discussion and conclusions

To begin with, in this second part of the project one can conclude that the data is well separable. The conclusion is drawn because one had a very high accuracy of the class predictions in both parts of the project, when predicting bean type (Part 1) and clustering class (Part 2). This is very feasible and desirable when dealing with and interpreting results, since clear characteristics can be interpreted and solid conclusions can be made about the groups/predictions.

To start with, it is worth noting that one can compare real labels to those derived through grouping and prediction. However, if this were not the case, the problem would be considerably more difficult to solve. It is interesting that the models almost always found the optimal number of clusters to be 2, however, not all clustering methods allow the implementation of the chosen approach. In these cases, the number of clusters is determined automatically, which, despite being considered an advantage, prevents the comparison with the best classifier obtained in the first part of the project whenever the number of clusters is different from two.

This leads to the conclusion that the beans may be classified into two very distinct bean kinds, and when compared to the true classes, it is evident that the clustering methods were successful in finding the two response variable classes, clearly distinguishing the BOMBAY and DERMASON bean types. To make this assumption, one claims that where the higher values appear is where the true class is, which means that if a cluster has more BOMBAY observations linked with it, that cluster is more likely to be the BOMBAY class cluster, and vice versa for DERMASON.

So, depending on the need and the "question to be answered" of the data mining analysis, one could also use the two clusters to divide the beans instead of the typical "by type" division. A scenario like this could be that one want to create a "bean mixture", but want to have a good proportion of the bean size, then one could use two clusters and thus simplify the separation.

Another interesting foundation when exploring the clustering methods was that the graph-based method, the MST-kNN technique, clustered the beans into 37 different clusters, which indicates that the technique is too complex for this type of problem. This is reasonable, since graph-based models are often used in social networks or for tracking the spread of diseases. These types of models are also suitable when the structure of the data is unknown, which is not the case for this dataset.

Finally, a supervised analysis was carried out, which compared the classes obtained by the kNN and XGBoost classifiers to the clustering results. The classifier was able to efficiently categorize the data linked with the various clustering approaches, particularly the K-Means method, which yielded the best results, according to the performance measures studied. Since the dataset under study provides the real class regarding the type of bean, the results produced in a semi-supervised analysis, where the real classes were compared to the classes obtained by the classifier, are the same as those obtained in the first part of the project. This is consistent with the clustering algorithms' good performance against true classes, as well as the classifier's strong performance versus the clusters' suggested classes.

When comparing these results to the ones acquired in the first part of the project, one notices that they are very similar. That is, classifiers can be used in either a supervised or unsupervised context, because clustering algorithms do not show any observations that have been misclassified. It is indeed worth noting that the use of clustering algorithms has been proved to be equally effective, and that all of the methods used in both parts of the project have outstanding performance measurements.

References

[1] Analytics Indiamag. 2021. *Comprehensive Guide to k-Medoids Clustering Algorithm*. https://analyticsindiamag.com/comprehensive-guide-to-k-medoids-clustering-algorithm/

[2] KDnuggets. 2021. *DBSCAN Clustering Algorithm*. https://www.kdnuggets.com/2020/04/dbscan-clustering-algorithm-machine-learning.html

[3] SHAP. 2019. *A gentle introduction to SHAP values in R*. https://blog.datascienceheroes.com/how-to-interpret-shap-values-in-r/