Master in Artificial Intelligence

INTRODUCTION TO MACHINE LEARNING

Work 1

Authors:
Ruizhe Yu Xia
Jan Rodríguez Miret
Jose Juan Galán Montero

Course 2020-2021

Contents

1 Introduction

In this first practical work of the subject, our goal is to implement several clustering algorithms for applying them to three different datasets.

All of these algorithms are divisive and unsupervised, we don't use the true classification for training our models. In addition, each dataset needs to be prepossessed before applying any of the clustering algorithm.

Apart from that, the performance of each algorithm will be measured and compared in order to extract conclusions about their accuracy.

The datasets selected are 'adult', 'waveform' and 'connect-4', which after the preprocessing they have 45222, 5000 and 67557 samples respectively.

2 Preprocessing

In our particular case, the adult dataset was large enough and had a relatively small number of missing values. Therefore, when finding missing values, we chose to delete the entire sample. We thought we could afford this loss of information and focus on other things rather than finding reasonable approximations for the missing values. The other datasets did not contain any missing value.

In order to have all features with the same weight when calculating distances between samples, we used the MinMaxScaler provided by sklearn library for general numeric normalization.

And finally, we used the pandas method get_dummies to encode all categorical values to a one-hot-encoded ones. In our case, the categorical values were equally different to any other possible values of the field, that is, the categories of a feature did not reflect any order, so it is nonsensical to attribute a numerical scale to them. Instead, this approach gives any different value the same distance between them. This encoding was not necessary for the waveform dataset, as it is contains only numerical values.

This encoding increased a lot the number of dimensions of these datasets: adult got from 14 to 104 columns and connect-4 from 42 to 126.

On the other hand, to encode the real classes for adult and cn4 datasets, which were categorical, we used a LabelEncoder. The reason is that in this case the value is not used to compute distances, only to group the samples together.

3 DBSCAN

The DBSCAN algorithm is a density based one, for executing it we need to fix some parameters. The algorithm can be customized in a certain degree choosing the desired metric and algorithm variant.

With the waveform dataset, the cosine distance and the brute algorithm were the only ones able to find some clusters (with any value of minPts and eps).

In the case of the adult dataset, the cosine distance hardly finds any cluster and the euclidean distance achieves finding some clusters, but not the rest of the metrics.

For the connect-4 one, we don't obtain good results in any of the configurations.

The first one is the minPts, that determines the minimum number of similar points required to from a cluster. This parameter is difficult to choose and there is not a single method to chose it. However, a very common used estimation takes the natural logarithm of the amount of points to analyze. This is what we chose to use here. The other one is epsilon, that determines how similar two points have to be to be considered of the same cluster. This parameter is also different for each dataset and for estimating it we can use the Nearest Neighbors function of sklearn, which will tell us how far are each one of the points to its nearest neighbours. We can use the point of where the curvature changes to approximate the value of epsilon.

Using this method to approximate and trying manually some values after, we arrive to "accurate" cluster number, but the results for any of the dataset are poor.

Table 1: DBSCAN metrics for adult dataset (eps=0.09, euclidean metric)

Silhouette score	-0.2876
Davies-Bouldin score	2.0839
Calinski-Harabasz score	2.3981
Adjusted rand score	0.0002
Fowlkes-Mallows score	0.7912

Table 2: DBSCAN metrics for waveform dataset (eps=0.29, cosine metric)

Silhouette score	-0.0919
Davies-Bouldin score	2.6275
Calinski-Harabasz score	16.8092
Adjusted rand score	0.00003
Fowlkes-Mallows score	0.5665

Table 3: DBSCAN metrics for connect-4 dataset (eps=0.005, cosine metric)

Silhouette score	0.7892
Davies-Bouldin score	0.1984
Calinski-Harabasz score	723.114
Adjusted rand score	0.0013
Fowlkes-Mallows score	0.70920

4 K-Means

Regarding the implementation of the basic K-means algorithm, there are some designing decisions that had to be made.

The first one is about the initial selection of the seeds. Assuming we have no external information about the distribution of the clusters and what should be the original seeds, the initial centroids can be chosen randomly.

This initialization can also be performed according to the K-means++ approach, that tries to get a centroid with higher probability of being far from the others, but in our particular cases it does not seem to help very much, because euclidean distance is not very representative of the difference of our examples. The results using this initialization were identical than when using random seeds, so we preferred to leave it that way.

```
for i in 0..n_seeds:
    new_seed = random_seed()
    if new_seed not in seeds:
        seeds.append(new_seed)
```

For calculating the distance between the centroids and every one of the points the metric chosen is the Euclidean one. We tested the cosine distance too, but for some datasets the calculation of this metric required more memory and produced no better results. Anyways,

for the datasets in which we could apply this metric, the results were very similar to the euclidean distance.

For the stopping criteria, we have chosen to stop the algorithm when the assignments of the points to the clusters stop changing, so the new assignment is exactly the same than in the last iteration. This guarantees that we are arriving at a local minimum of the error function (sum of the square of the distances between points).

Assuming that we have no external ground truth, the only way to get a proper k value is trying different ones and choosing the one that gets better internal measurements using internal metrics.

For each dataset we tried the following values of k: 2, 3, 4, 5, 6, 8, 10, 15 and 20.

4.1 Adult dataset

For the adult dataset we obtained these results.

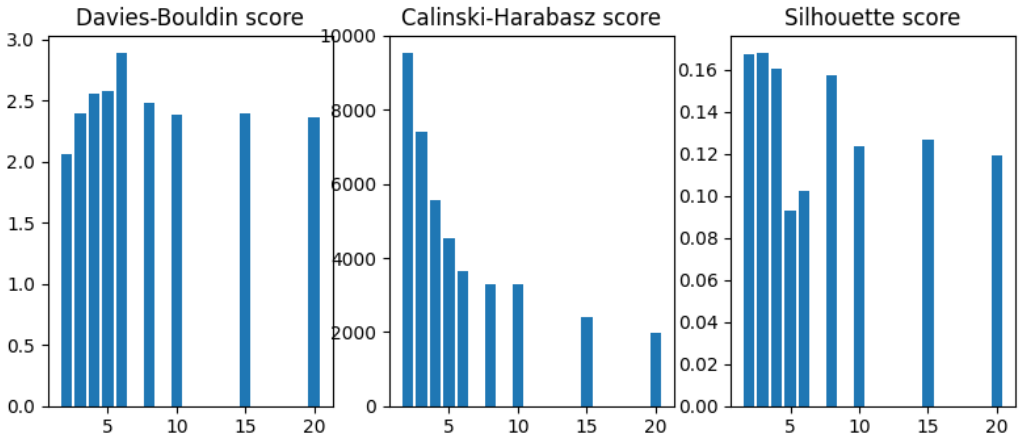

Figure 1: Internal metrics for the adult dataset

Both the Calinski-Harabasz and the Silhouette scores points to k=2 being the best value, while the Davies-Bouldin also points this value as the best (lower values are better for this index). Finally, the silhouette score also gives the best values for k=2 and k=3. Having this into account, k=2 seems to be the best value.

We obtain these values for external metrics (comparing with the true classes)

For k=2:

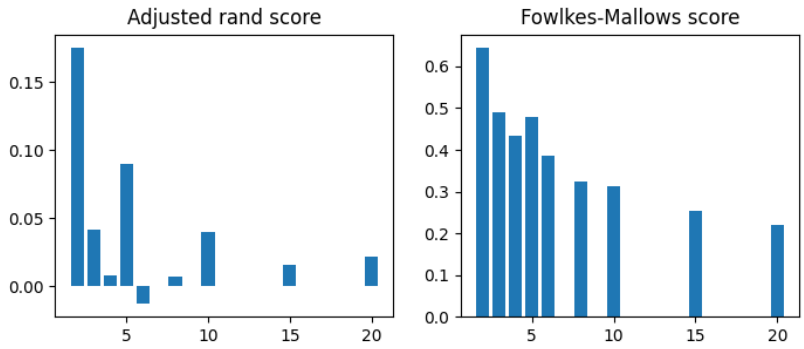

Figure 2: External metrics for the adult waveform

Adjusted rand score: 0.175 Fowlkes-Mallows score: 0.644

Which are not so good results.

4.2 Waveform dataset

For the waveform dataset we obtain different results. The Davies-Bouldin score is the best for k=2 or k=3

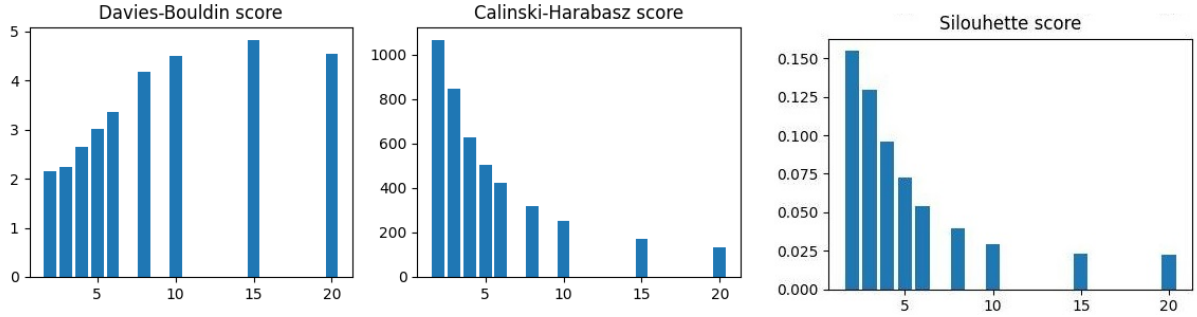

Figure 3: Internal metrics for the waveform dataset

When comparing with the ground truth, we obtain better results for k=2, even when this is not true. This can probably explained by the high dimensionality of the data.

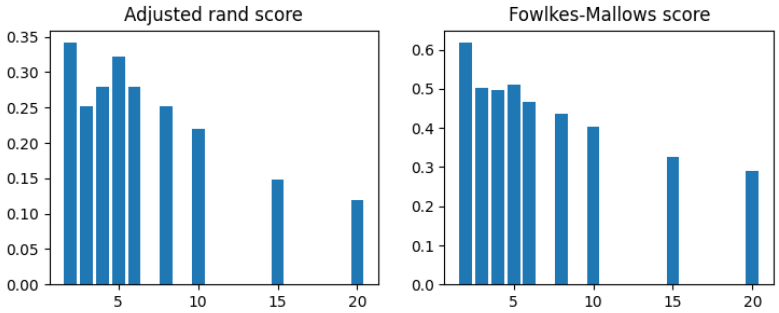

Figure 4: External metrics for the waveform dataset

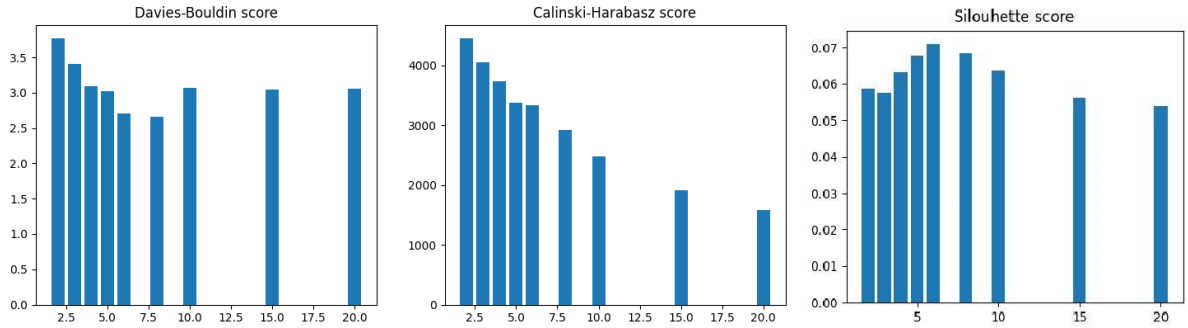

Figure 5: External metrics for the connect-4 dataset

4.3 Connect-4 dataset

Finally, the connect-4 dataset reported the less clarifying results.

The different internal metrics point in to different results so no conclusion can be clearly extracted.

Even when applying external validation, results are terrible even for the true values of k (3). The accuracy of the k-means algorithm for this dataset is very poor. This can be caused by the full nominality of the attributes.

Once chosen the k parameter for each dataset, we can perform an external validation to check the performance of our algorithms, since we indeed have available the true classes.

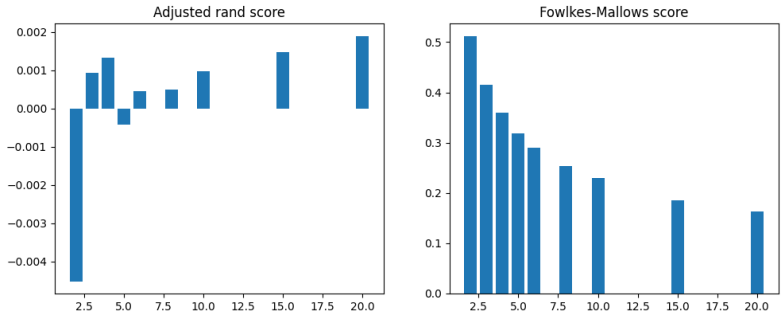

Figure 6: External metrics for the connect-4 dataset

5 Bisecting K-Means

The Bisecting K-Means algorithm is a hierarchical version of the K-Means that consists in splitting a selected cluster, according to some metric, in two using the K-Means algorithm.

In our case, we began by bisecting the largest cluster, as it is one of the easiest ways to determine which cluster to select. However, we quickly found out that this was not a good criterion for our datasets because we had a dominant class and therefore the one that is split first. Hence, we modified our code to select the most heterogeneous cluster, that is, the one with the highest average distance between the samples and the centroid of its cluster.

Once the algorithm was implemented, we tested it with our three datasets. For each dataset, we are trying to get the K values that maximizes the internal metrics and then evaluate the best K values with some external metrics, by comparing the real classes for the samples with the predicted ones.

5.1 Adult Dataset

For the adult dataset the best results are obtained with a number of clusters equal to 2, as you can see in Figure 7 below. It makes sense because it is the number of different classes that we have in our labelled data ('<=50k' and '>50k'). We were looking for the K that has a lower Davies-Bouldin score and a higher Calinski-Harabasz and Silhouette scores.

If we evaluate the algorithm for k=2 and k=3 with the selected external metrics, we can see that the results match what we should expect: k=2 fits better the data, as it is the real number of clusters of the labelled data. The difference between both values of K is large enough to state that.

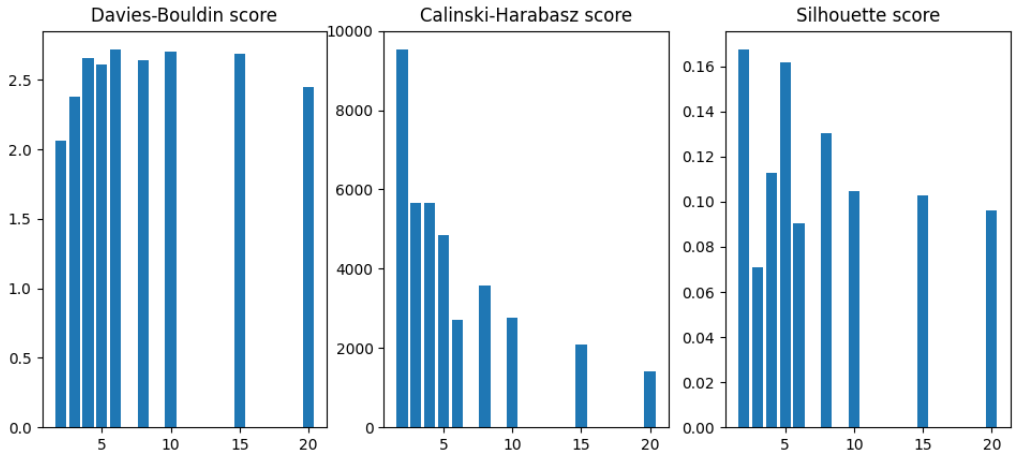

Figure 7: Internal metrics for bisecting K-Means and adult for different k values

The results, though, are not very accurate. This means that the algorithm did not find strong patterns that actually exist in the data.

	ARS	FMS
k=2	0.17546788	0.6443945
k=3	0.0637507	0.5087057

Table 4: External metrics for adult dataset

5.2 Waveform Dataset

For the waveform dataset, we obtained curious results because it shows a better performance in all metrics when we use a k=2, which is different than the real one (k=3), even for external metrics.

By looking at the internal metrics, it is clear that the samples are best arranged with only 2 groups, showing a clear tendency that the more clusters, the lower the metrics. This leaded us to the following hypothesis: the data may be very homogeneous without any clear defined group.

The results for external metrics between 2 and 3 clusters are very similar, but with k=2 being slightly better. Once again, the algorithms did not find a model that fits well the

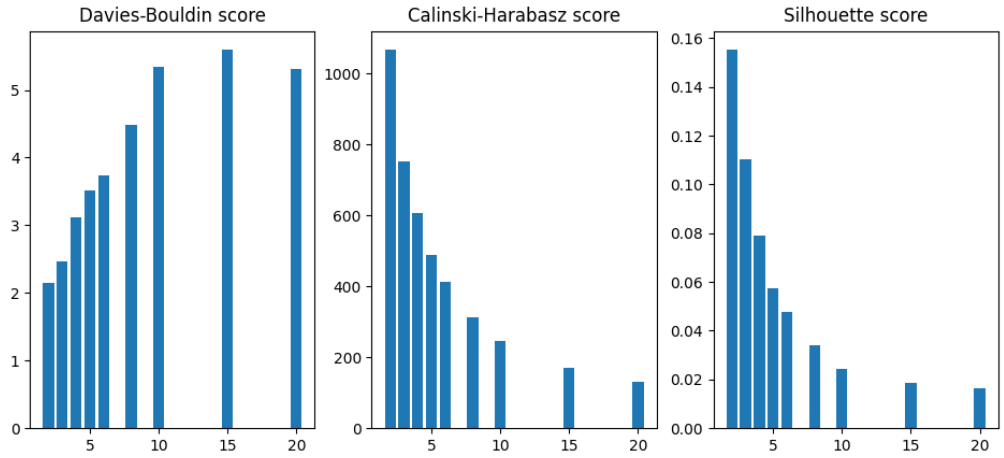

Figure 8: Internal metrics for bisecting K-Means and waveform for different k values

	ARS	FMS
k=2	0.34186588	0.61801443
k=3	0.32988581	0.56591171

Table 5: External metrics for waveform dataset

dataset, although the external metrics are higher than the adult's.

5.3 Connect-4 dataset

The Figure 9 displays the internal metrics for this algorithm, which are quite bad compared to what we obtained in the other datasets. It is not so clear which is the best K value. We chose k=2 and k=3 again as best because they have the higher Calinski-Harabasz and Silhouette score and an roughly the same Davies-Bouldin score than the rest.

As expected, we got very poor results from the external metrics, too. The score for the Adjusted Random Index tells us that the output clusters are nearly random (close to 0.0). The fact that we got better results with k=2 (though extremely poor too) is an indicator that the algorithm probably did not work quite well.

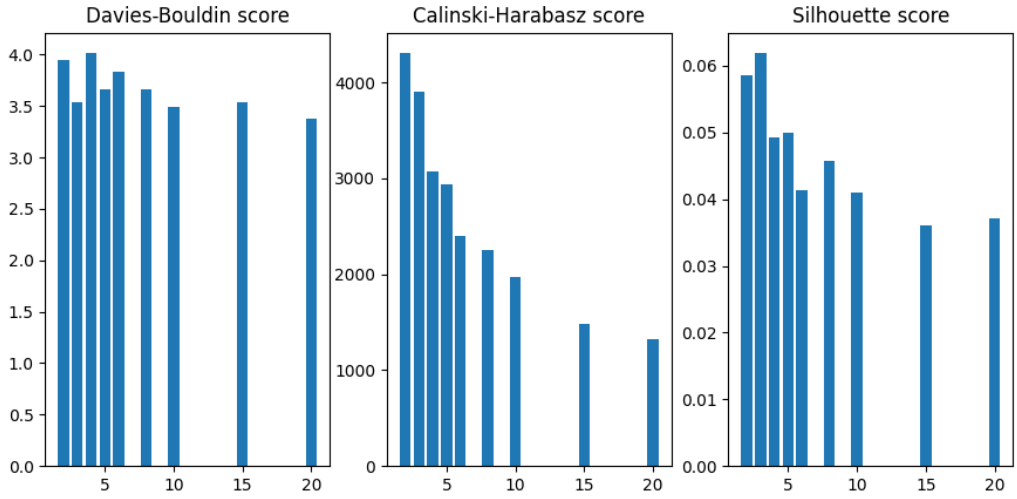

Figure 9: Internal metrics for bisecting K-Means and connect-4 for different k values

	ARS	FMS
k=2	0.00562575	0.52412939
k=3	0.00205232	0.42973191

Table 6: External metrics for Connect-4 dataset

6 K-Harmonic Means

The K-Harmonic Means (KHM) algorithm is a variation of the K-Means algorithm whereby the performance metric is the harmonic mean of the p-distances to all centroids. Accordingly, each point belongs to each centroid with a certain degree of membership.

Aside from the k parameter, KHM takes a p parameter for the used p-distance. It also includes parameter for the a maximum number of iterations and another one for the minimum change in performance before stopping (tolerance). The two former parameters will be considered fixed at 100 and 10^{-12} respectively. Evaluation has been preformed for k values varying within $\{2, 3, 4, 5, 6\}$ and p within $\{2, 3, 4\}$. To initialize the centroids, k random samples of the training data are taken, as recommended by the author.

6.1 Adult Dataset

The internal metrics are gathered in figure 10. From this figure, it can be inferred that the best values would be $k = 2$, $p = 3$. This is because it minimizes the DBS and maximizes both CHS and Silhouette. Thus we obtain the real number of labels.

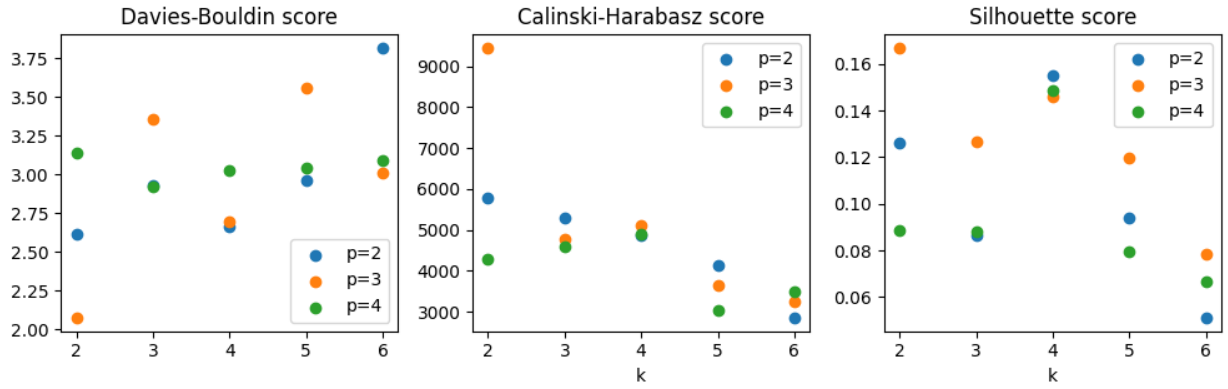

Figure 10: Internal metrics for KHM applied to adult with different k and p values

The external metrics are gathered in figure 11. The selected values $k = 2$, $p = 3$ performs best for the FHS, but it is outperformed in ARS by $k = 3$, $p = 2$.

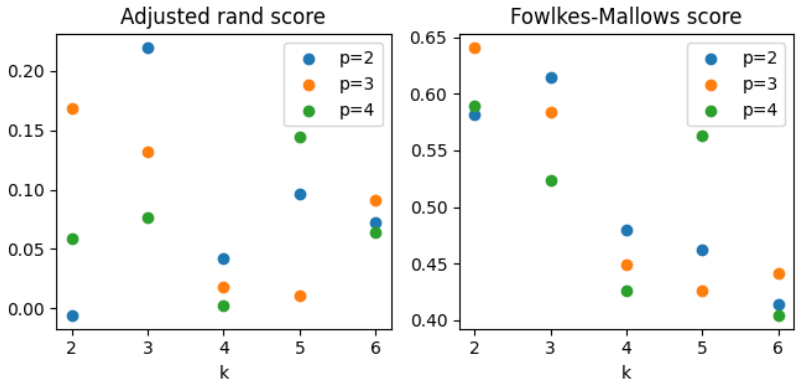

Figure 11: External metrics for KHM applied to adult with different k and p values

6.2 Waveform Dataset

The internal metrics are gathered in figure 12. From this figure, it can be inferred that the best values would be $k = 2$, $p = 2$. However, $k = 2$, $p = 3$ is closely behind. Neither of them is the correct number of labels.

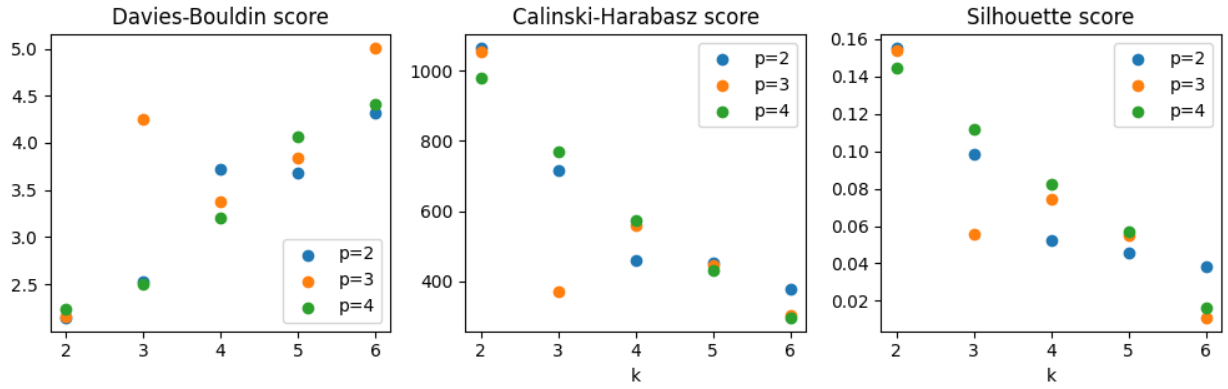

Figure 12: Internal metrics for KHM applied to waveform with different k and p values

The external metrics are gathered in figure 13. For the pairs of values selected above it can be seen that $k = 2$, $p = 3$ outperforms $k = 2$, $p = 2$. This is not surprising, since the two values' internal scores are close. What is surprising is the ARS for $k = 6$, $p = 3$, which is above the one of the selected values.

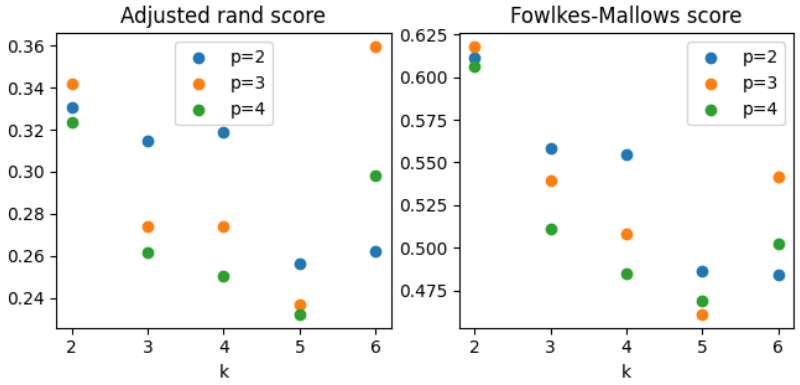

Figure 13: External metrics for KHM applied to waveform with different k and p values

6.3 Connect-4 Dataset

The internal metrics are gathered in figure 10. Conflicting values are obtained. However if one were to choose a pair of values, a good choice would be $k = 2$, $p = 3$.

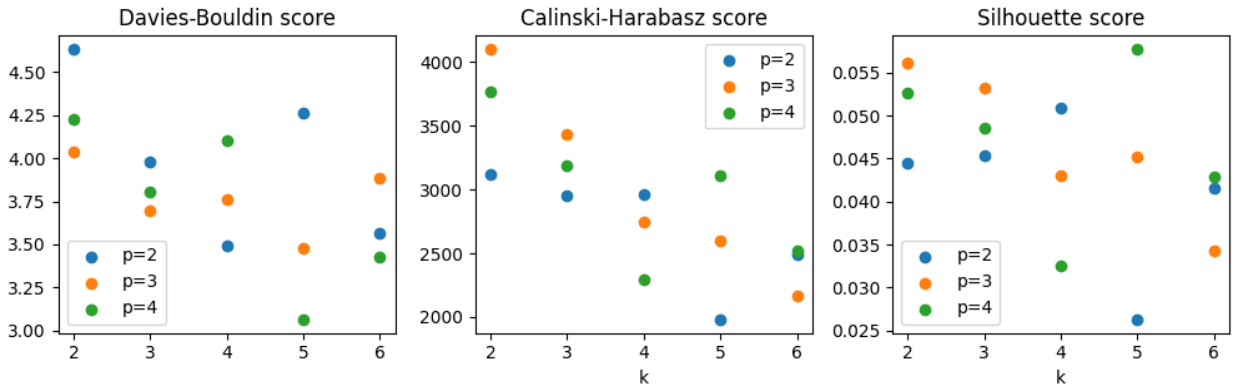

Figure 14: Internal metrics for KHM applied to Connect-4 with different k and p values.

The external metrics are gathered in figure 13. The selected values $k = 2$, $p = 3$ perform best for the FHS, but it is one of the worst in ARS.

As we have seen with the algorithms above, the Connect-4 dataset may not be suitable for unsupervised learning using our design decisions. This may also explain the inconsistencies in internal metrics and low external scores.

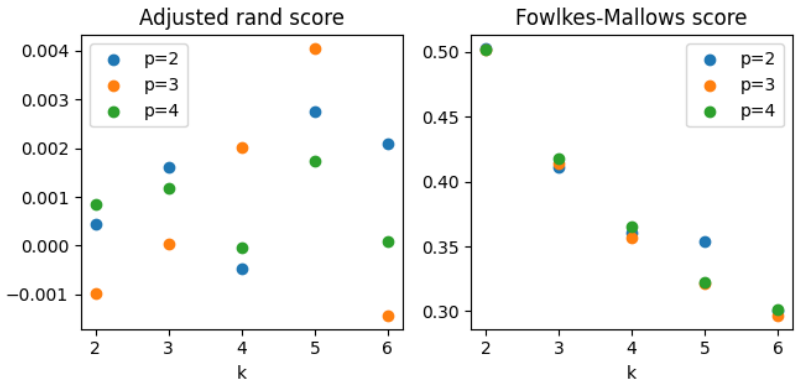

Figure 15: External metrics for KHM applied to Connect-4 with different k and p values.

7 Fuzzy C-Means

The Fuzzy C-Means (FCM) algorithm is a variation of the K-Means algorithm whereby the clusters are fuzzy sets. Accordingly, each point belongs to each centroid with a certain degree of membership.

Aside from the k parameter, FCM takes an p parameter for certain power within the algorithm. It also includes parameter for the a maximum number of iterations and another one for the minimum change in the universe matrix before stopping (tolerance). The two former parameters will be considered fixed at 100 and 10^{-2} respectively. Evaluation has been preformed for k values varying within $\{2, 3, 4, 5, 6\}$ and p within $\{2, 3, 4\}$. The universe matrix is initialized randomly and normalize per sample.

7.1 Adult Dataset

The internal metrics are gathered in figure 16. The DBS has values that are conflicting with the other two. If we ignore this value, we obtain similar CHS's and Silhouettes for all $k = 2$ and for $k = 3, p = 3$.

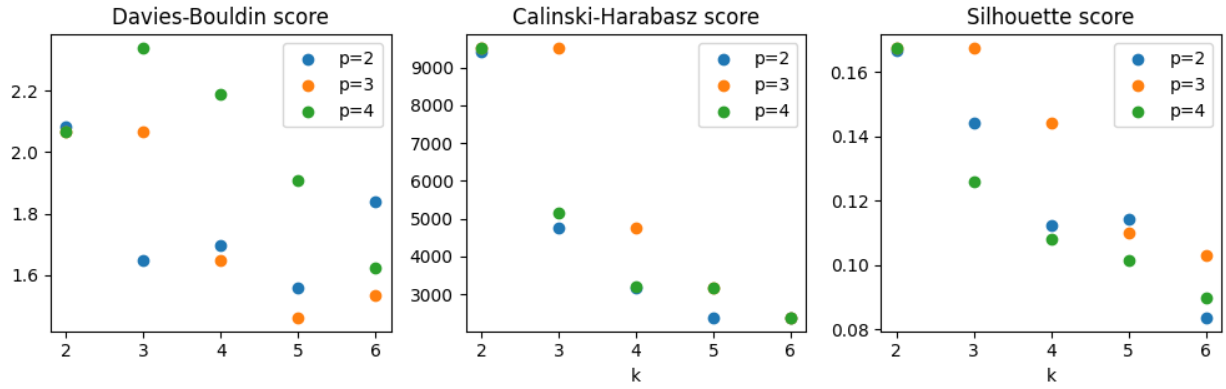

Figure 16: Internal metrics for FCM applied to adult with different k and p values

The external metrics are gathered in figure 17. The selected values obtain similar scores for this dataset.

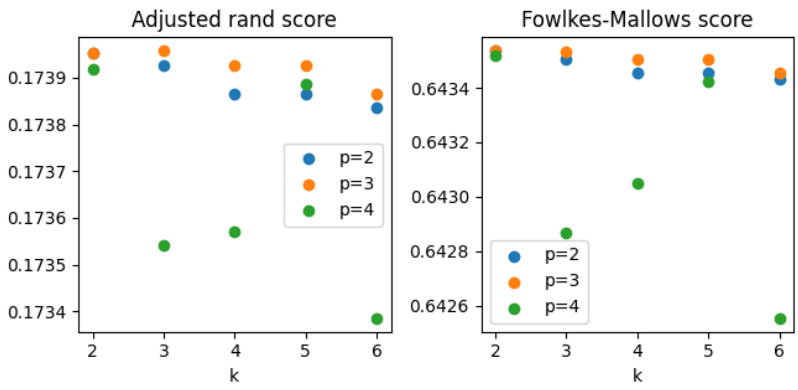

Figure 17: External metrics for FCM applied to adult with different k and p values

7.2 Waveform Dataset

The internal metrics are gathered in figure 18. From this figure, it can be inferred that the best values would be any of the $k = 2$. Again, the true number of labels is not selected.

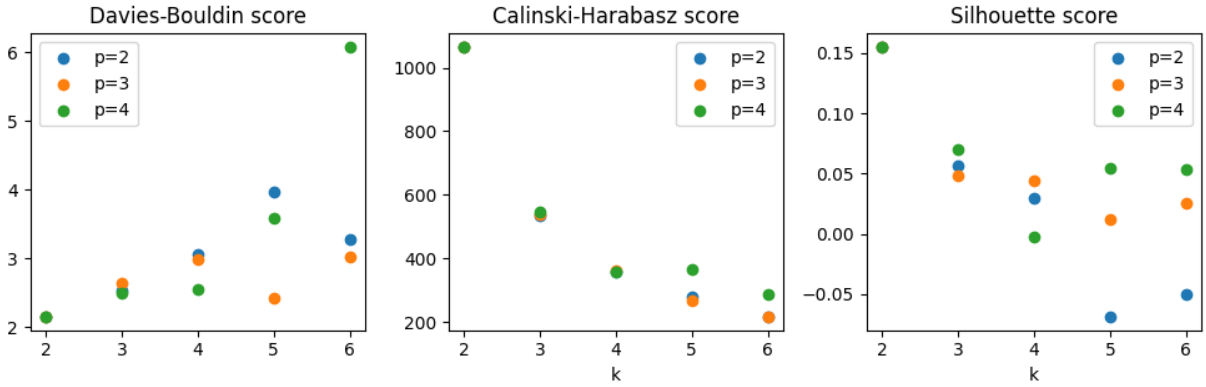

Figure 18: Internal metrics for FCM applied to waveform with different k and p values

The external metrics are gathered in figure 19. The selected values are clearly outperformed by $k = 4, p = 3$ in ARS, but obtain high FMS's.

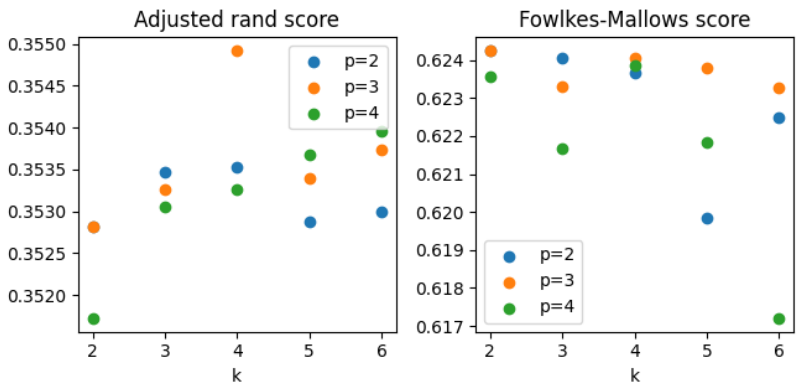

Figure 19: External metrics for FCM applied to waveform with different k and p values

7.3 Connect-4 Dataset

The internal metrics are gathered in figure 20. From this figure, the DBS shows some conflicting values when compared with the other two. If we ignore DBS, CHS and Silhouette point to $k = 2, p = 3$ y $k = 2, p = 4$

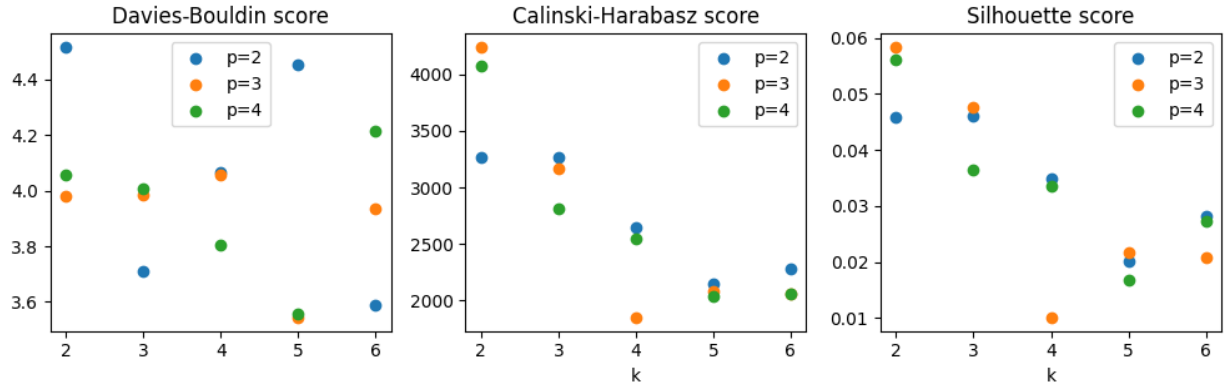

Figure 20: Internal metrics for FCM applied to Connect-4 with different k and p values.

The external metrics are gathered in figure 21. The selected values perform best for the FHS, but in ARS it is not clear which value is best. Even so, as we found on all the algorithms above, the ARS is extremely low for this dataset.

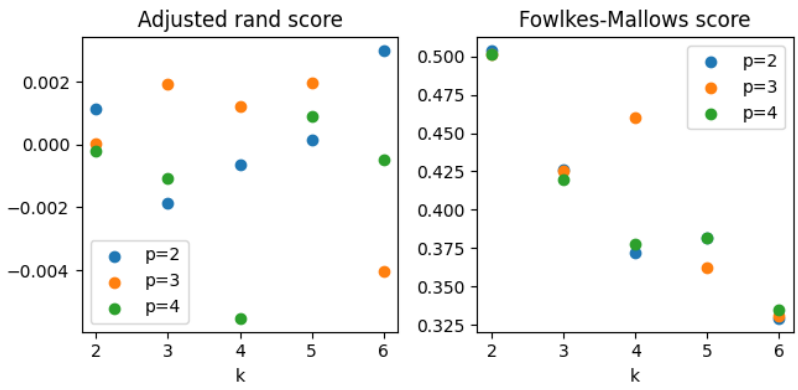

Figure 21: External metrics for FCM applied to Connect-4 with different k and p values.

8 Conclusions

Barely any useful information can be obtained by using this unsupervised algorithms, the main idea that we extract is that they are not effective against our data.

The DBSCAN algorithm shows nearly zero effectiveness when applied to any of the dataset. Density-based algorithms do not perform well in high dimensional data, and all our datasets can subscript to this definition.

All algorithms produce more or less the same results, with not a good performance in any of the metrics, but pointing in the same direction. With every algorithm, the best number of clusters seems to be two for the adult dataset and also for the waveform one (although three gets closer). The Connect-4 dataset does not behave well under any of the algorithms and produces non-concluding results.

The differences between the results are not very big. We might point out that the variations of the base k-means (harmonic and bisecting) achieve some better results using both internal and external metrics. However they are far from being a substantial improvement.

The explanation of this results can probably be caused by the high dimensionality of our data, in addition to the great number of nominal attributes. Partitional algorithms like these have a reduced accuracy when dealing with this two specific issues.

9 Execution Guide

In order to test the different algorithms you can use the Python main.py file by entering the following on the command line:

```
python main.py
```

Once the main is executed, you will be asked to specify the dataset and the algorithm you want to test. The algorithm selected will be executed with the best parameters values we found during our research.

Additionally to the main file, there are some other tests files that can be executed and configured through commenting the desired part. These test files were used to generate the graphics and can be used to check detailed results.