Unsupervised Learning: Clustering

Introduction

In this section, the goal is to cluster streams that produce similar, related or coordinated patterns using unsupervised clustering techniques, regardless of labels.
To achieve this goal, we chose to use three clustering algorithms: K-Means, Gaussian Mixture Models (GMM), and DBSCAN.

The choice of these models is motivated by their different nature and their abilities to detect structures in the data. K-Means is known for its simplicity and efficiency in dividing data into spherical clusters based on Euclidean distance. This method is useful when the clusters are well separated and regular in shape. However, it may not perform well in the presence of clusters of irregular shape or variable density.

To address the limitations of K-Means, we have included the Gaussian Mixture Model, which uses a probabilistic approach to identify elliptical clusters based on Gaussian distributions. GMM is flexible in modeling clusters with different shapes and can provide a membership probability for each point, making it useful in complex scenarios.

Finally, we chose DBSCAN because of its ability to identify clusters of arbitrary shape and to handle noise. DBSCAN does not require specifying the number of clusters a priori, but identifies dense areas of points and separates them from less dense regions, which it considers as noise. This approach is particularly advantageous when the data contain noise or have clusters of varying density.

For each algorithm, determining the optimal number of clusters is critical and can be done using methods such as the elbow method or silhouette analysis. These techniques help us understand the internal structure of the data and choose the number of clusters that best represents the data. Next, it is essential to find the best hyperparameters for each model to optimize their performance.

K-Means

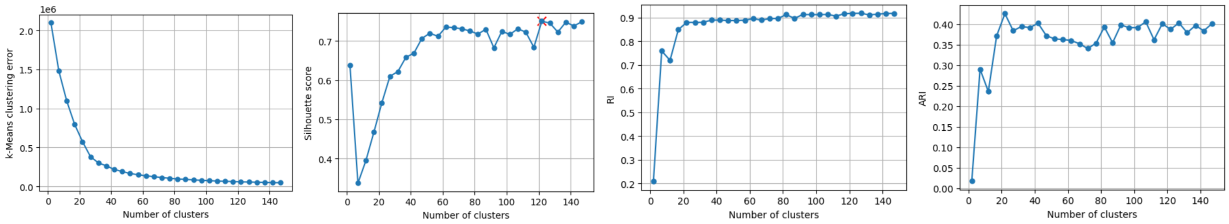
Introduction

The K-means method is a particularly effective clustering algorithm for clustering a dataset. The main objective of K-means is to minimize the sum of the distances between data points and the centroid of their respective cluster. The centroid is the midpoint of all points that belong to that cluster.

One of the critical aspects of using K-means is choosing the correct number of clusters. This number not only directly affects the quality of clustering, but can also reveal significant details about the data itself. Inappropriate choice of the number of clusters can lead to misleading results, such as over- or under-segmentation of the data. Techniques such as the elbow method or silhouette index can help determine an appropriate number of clusters, balancing between optimizing the internal variance of clusters and minimizing model complexity.

Determine the parameters of k-means

To select the optimal number of clusters for k-means clustering, it is useful to consider a combination of different evaluation metrics, as each provides a different perspective on clustering quality.



The elbow method looks for the point where the decrease in clustering error slows down, indicating that adding more clusters does not significantly improve the model. From the clustering error graph, it is observed that the elbow is around 20-40 clusters, where the rate of decrease in error slows down.

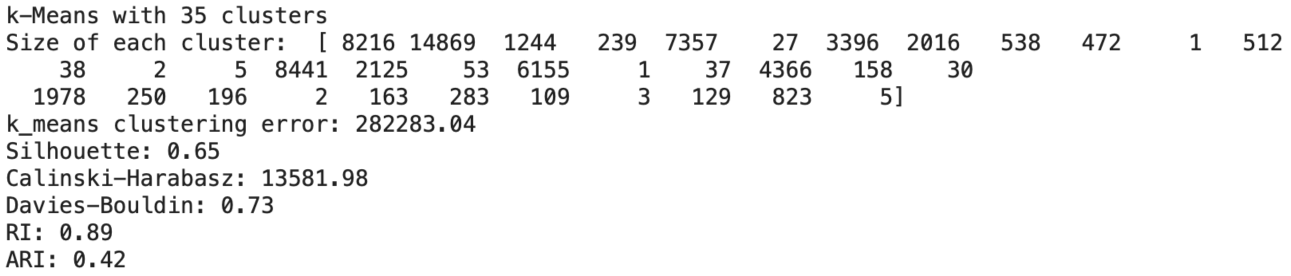
Il punteggio silhouette misura quanto un oggetto è simile al proprio cluster rispetto agli altri cluster. Un punteggio più alto indica cluster meglio definiti. Osservando il grafico del punteggio silhouette, si nota che raggiunge il massimo intorno ai 120 cluster. Tuttavia, scegliere il numero massimo di cluster basandosi esclusivamente su questo criterio potrebbe portare a selezionare un numero eccessivo di cluster, che potrebbero non essere significativi dal punto di vista pratico.

The silhouette score measures how similar an object is to its own cluster relative to other clusters. A higher score indicates better defined clusters. Looking at the silhouette score graph, it can be seen that it peaks around 120 clusters. However, choosing the maximum number of clusters based solely on this criterion could lead to selecting too many clusters, which may not be meaningful from a practical point of view.

The Rand Index (RI) and Adjusted Rand Index (ARI) both evaluate clustering similarity, but the ARI is normalized to account for chance, making it more reliable, especially in noisy datasets. Unlike RI, which ranges from 0 to 1, ARI ranges from -1 to 1, where negative values indicate agreement less than chance, enhancing its sensitivity to noise and outliers.

Combining these observations, a balance can be found between silhouette score and a practical number of clusters. Both ARI and RI suggest that around 20-40 clusters is optimal, with ARI peaking around 35 and RI stabilizing around 20. The elbow method supports these results, indicating a range of 20-40 clusters. This range offers a good compromise between clustering quality and practical significance. In light of the observed peak and stabilization points across various metrics, we have decided to utilize 35 clusters for our analysis.

Applying K-Means

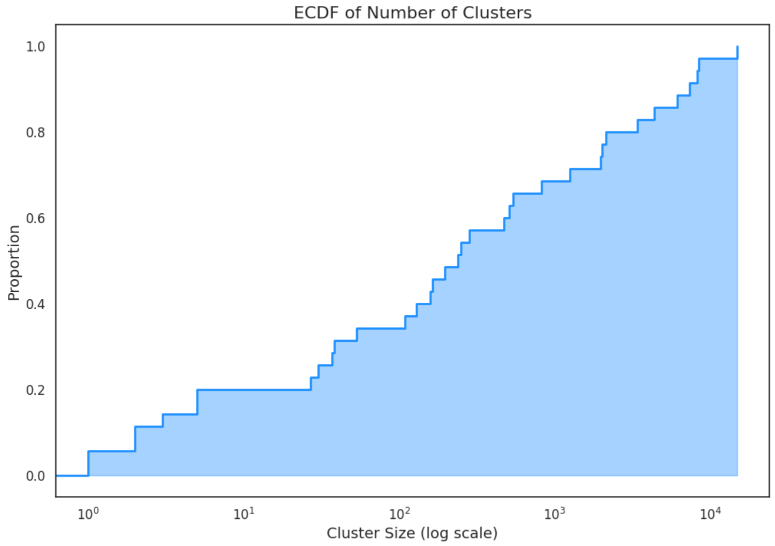


```
k-Means with 35 clusters
Size of each cluster:  [ 8216 14869  1244   239  7357    27  3396  2016   538   472     1   512
    38     2     5  8441  2125    53  6155     1    37  4366   158    30
  1978   250   196     2   163   283   109     3   129   823     5]
k_means clustering error: 282283.04
Silhouette: 0.65
Calinski-Harabasz: 13581.98
Davies-Bouldin: 0.73
RI: 0.89
ARI: 0.42
```

As can be seen, we have chosen to use two additional indices to aid in our evaluation:

- The Calinski-Harabasz index, also known as the between-cluster variance ratio, is defined as the ratio of the sum of between-cluster variance to the sum of within-cluster variance. A higher value indicates better cluster separation. An elevated value of 13581.98 suggests that the clusters are well-separated and compact, meaning that the points within each cluster are close to each other and distant from points in other clusters. This index indicates that the clustering model has successfully created distinct clusters, which is a positive sign of the quality of the clustering.
- The Davies-Bouldin index measures the average similarity between each cluster and its most similar cluster. A lower value indicates better separation of the clusters. A value of 0.73 is considered good, suggesting that the clusters are well-separated from each other. A value close to zero would be ideal, indicating that each cluster is distinct with no significant overlaps.

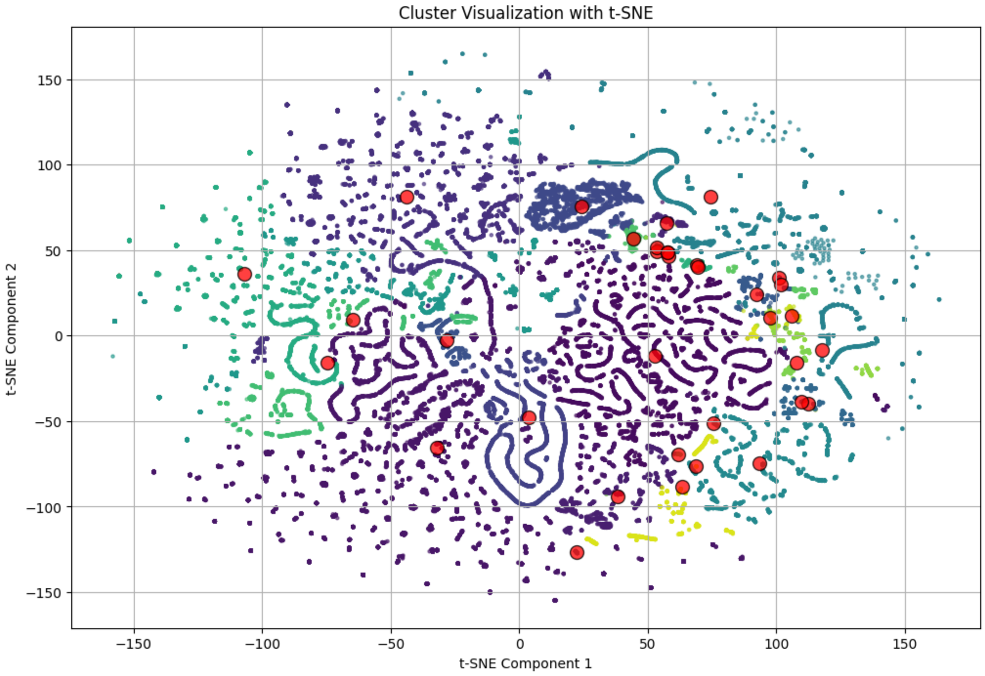
ECDF of number of cluster



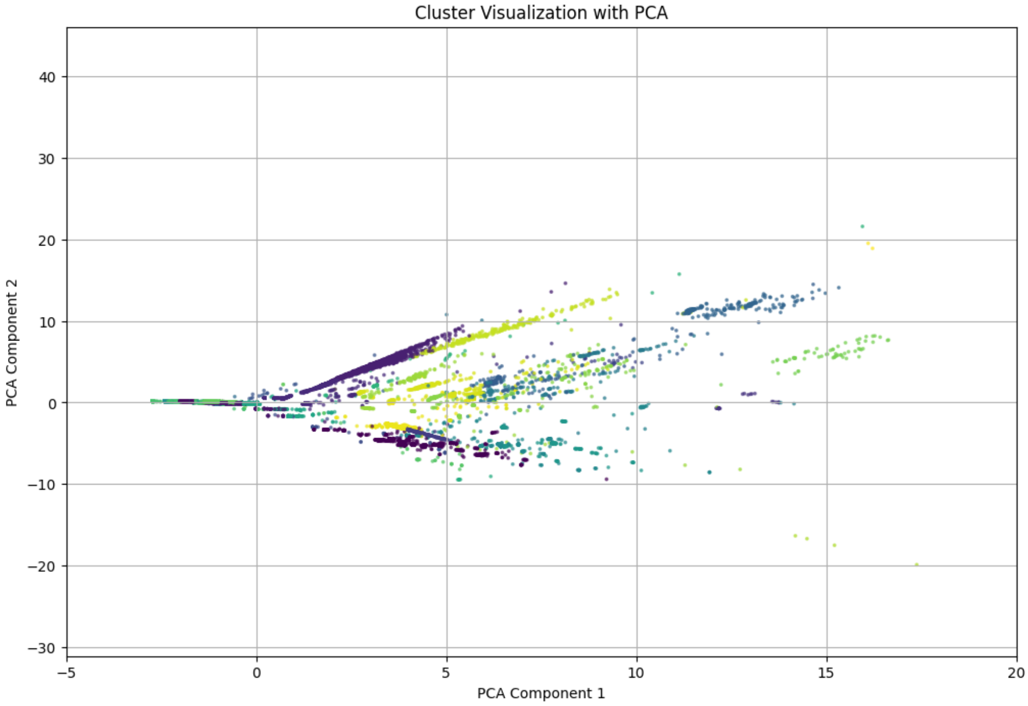
The curve starts to rise from relatively small cluster sizes (near 10^0, or 1) and continues to rise gradually. This indicates that there are many small cluster sizes that can identify noise.
The curve becomes steeper between 10^1 (10) and 10^3 (1000), suggesting that a large proportion of the clusters fall within this size range. This could indicate that most of the data are clustered in clusters of moderate size.
Toward the right edge of the graph, the curve flattens out near level 1, indicating that almost all of the data were considered in the ECDF calculation. This suggests that there are few very large clusters (near 10^4), while between 10^3 and 10^4 about 35 percent fall.

Thus the ECDF shows that much of the data is concentrated in a moderate number of medium-sized clusters, with fewer very large or very small clusters. This may indicate that the K-means method has identified some dominant patterns in the data, but also that there are many small clusters or outliers.

Cluster Visualization



Several significant aspects emerge from the image analysis of the t-SNE visualization. Some clusters appear well separated and compact, indicating effective classification of K-means for certain portions of the data. Centroids, represented by red dots, are accurately placed in the center of their respective clusters, suggesting correct identification of centers of mass for these groups. In addition, some dots are noted to appear isolated or located between multiple clusters, suggesting the presence of outliers or data that do not clearly adhere to a single cluster.



Analysis by PCA of the clustered data using the K-means algorithm reveals how the clusters are predominantly distributed along the first principal component, which is confirmed as the direction of greatest variance and informational relevance. This oriented distribution suggests a clear delineation of the most influential features of the data. The different shapes and directions observed in the clusters indicate significant structural variations. However, the proximity and partial overlap of some clusters near the origin of the components may signal that not all data sets are distinct.

Gaussian mixture model

Introduction

The Gaussian Mixture Model (GMM) is a probabilistic clustering model that assumes the generation of data by combining several Gaussian distributions, each of which corresponds to a cluster in the data set. Unlike the simpler K-means algorithm, which rigidly assigns each point to a single cluster, GMM evaluates the probability that each point belongs to several clusters. This approach, known as soft clustering, provides a richer and more detailed picture of the structure of the data, particularly when clusters intersect or overlap.

Unlike K-means, which presumes that all clusters are spherical in shape and of similar size, GMM is capable of fitting clusters of different elliptical shapes and sizes, thus allowing more complex relationships among the data to be captured.

The GMM, being a parametric model, fits well in scenarios where explicit statistical modeling of the data is essential, and it offers more flexibility in dealing with clusters of different shapes and sizes. On the other hand, its effectiveness may be limited by the choice of initial parameters and sensitivity to outliers in the data, aspects in which DBSCAN shows greater robustness due to its ability to effectively handle outliers and adapt to density variations in the data.

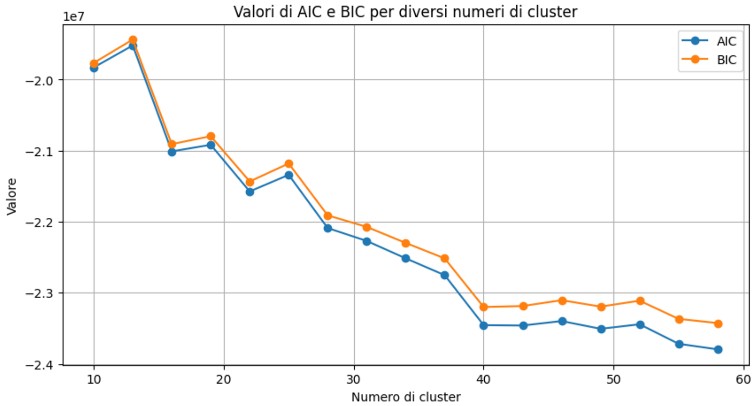
Determine the parameters of GMM

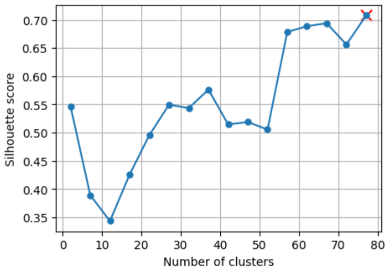
In the previous discussion, we examined the utility of the Silhouette Score, Rand Index (RI), and Adjusted Rand Index (ARI) for selecting the number of clusters. To these metrics, we now add log-likelihood, a measure assessing the probability that a Gaussian Mixture Model (GMM) generates the observed data.

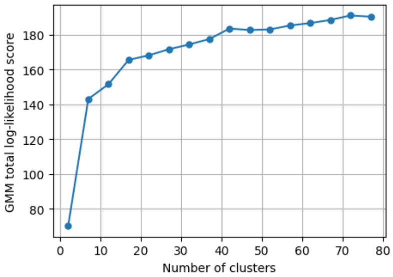
It is important to note that unlike the Elbow method commonly used in K-means analysis, this approach is not suitable for GMMs due to their probabilistic nature and intrinsic complexity. GMMs do not rely merely on distance measures like K-means but model the probabilities and covariances of the data. This shifts the focus from simply reducing the variance within clusters to a more critical sensitivity to initial parameters and the type of covariance used. Thus, applying a criterion based solely on variance change, as in the Elbow method, could yield less stable and reliable results.

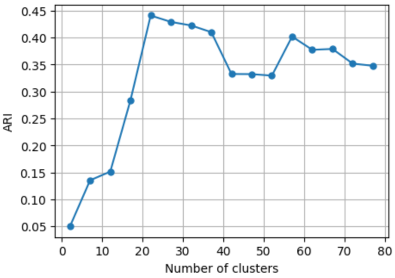
In the context of Gaussian Mixture Models (GMM), log-likelihood serves as a fundamental measure to evaluate how accurately a model represents the observed data, indicating the probability that the model has generated the observed data. A higher value suggests that the model better fits its Gaussian components to the clusters in the data, providing a direct indication of model fit.

Moreover, log-likelihood also serves as the basis for information criteria such as the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC), which integrate log-likelihood with penalties accounting for the number of parameters in the model. These criteria are designed to help balance the complexity of the model against its ability to fit the data well without overfitting, which is why it is deemed appropriate to calculate them.









We examine the information from the upper images with the goal of identifying the optimal number of clusters based on measures such as Adjusted Rand Index (ARI), log-likelihood, Silhouette score, and AIC and BIC criteria.

The Silhouette score, which assesses how accurately the data were assigned to clusters, was initially low and showed a noticeable increase to 35 clusters. Continuing beyond this number, the value continued to increase, peaking around 80 clusters and then stabilizing. As mentioned above for K-Means, it is normal that the more clusters you have the more the Silhouette goes up, but we need to find the right trade-off.

As for log-likelihood, a rapid increase up to 20 clusters was noted, after which its increase becomes marginal. This suggests that adding additional clusters beyond 20 does not make substantial improvements to the verisimilitude of the model, indicating that a number of clusters between 20 and 30 could adequately represent the data.

The ARI value showed rapid growth until it reached about 20 clusters, then fluctuated between 0.3 and 0.4 without showing any further significant trends. This index, being high when the clustering closely matches the actual clustering of the data, suggests that a high value is desirable for valid analysis, so a value between 20 and 30 seems good.

Finally, the AIC and BIC criteria, both indices that penalize increasing model parameters, showed a marked decrease up to 30-40 clusters, a point beyond which their reduction becomes less pronounced. This phenomenon suggests a good balance between model complexity and data fit.

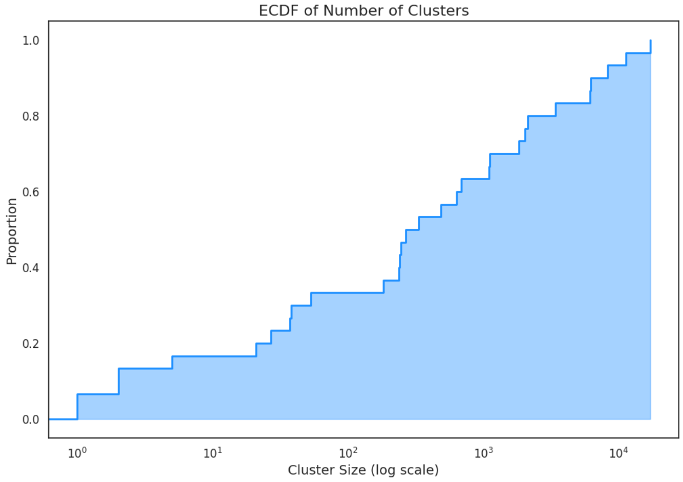
This cross-sectional analysis suggests that a range of 20-40 clusters may represent the optimal clustering compromise. This range is supported by stabilizing log-likelihood, reducing AIC and BIC penalties, and improving Silhouette score.

Using a number of clusters of 30 thus seems to be the most balanced choice, maximizing the goodness of fit of the data without making the model overly complex, thereby also facilitating the interpretation of the results.

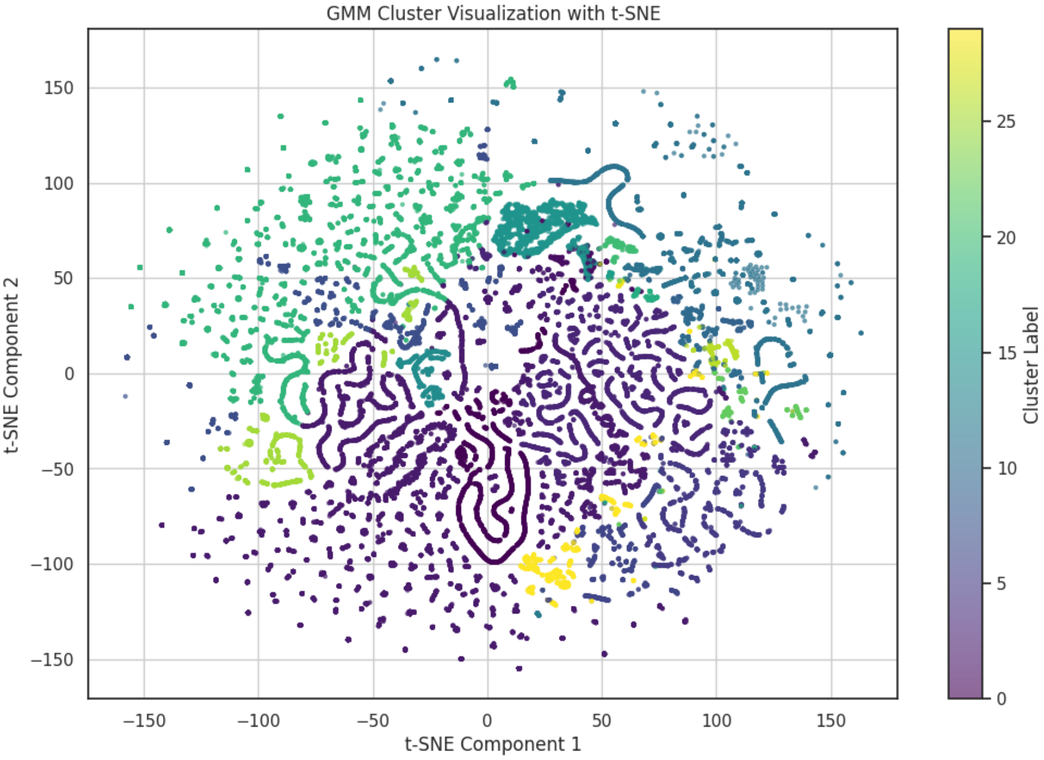
Applying GMM

```
Number of clusters:  30
Size of each cluster:  [ 3425    237 17118   6196    489   2121   1121   6113     27    246    631   8313
     38      1    686   2043     53      1    183 11358    240     21    333      2
      2   1831    268      5     37   1100]
Silhouette: 0.55
RI: 0.88
ARI: 0.43
```

ECDF of number of cluster



Il grafico rivela una variazione significativa nelle dimensioni dei cluster. Un gran numero di cluster minori suggerisce l'identificazione da parte del GMM di diverse sottogruppi nei dati, mentre la presenza di alcuni cluster di dimensioni maggiori indica che il modello ha rilevato caratteristiche dominanti che hanno attratto un maggior numero di punti. Questa distribuzione evidenzia sia la diversità dei dati che l'esistenza di tendenze predominanti.



L'analisi della visualizzazione t-SNE dei cluster formati dal Gaussian Mixture Model (GMM) rivela una serie di dinamiche interessanti nel comportamento del modello di clustering. Alcuni cluster appaiono ben definiti e distintamente separati, evidenziando la capacità del GMM di identificare gruppi con caratteristiche ben differenziate, come dimostrato dai cluster in verde chiaro, blu e giallo. Questi cluster non solo mostrano una buona separazione ma anche una compattezza che suggerisce una omogeneità interna elevata. Al contrario, altre regioni, in particolare quelle colorate di viola, presentano una maggiore sovrapposizione e una distribuzione più estesa, indicando una difficoltà del modello nel distinguere chiaramente tra alcuni gruppi di dati.

Mentre alcuni cluster piccoli e concentrati suggeriscono la presenza di nicchie di dati ben definite, altri più ampi e meno delineati potrebbero indicare gruppi di dati che condividono caratteristiche trasversali.

DBSCAN

Introduction

Among the various clustering algorithms available, Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is particularly useful in situations where the data have a complex spatial structure or contain noise (anomalous data or outliers). Unlike algorithms such as k-means, which require specifying the number of clusters a priori, DBSCAN independently determines the number of clusters based on the density of the data. This makes it particularly suitable for exploratory data analysis.

DBSCAN identifies clusters as areas of high point density separated by areas of low density.
One of the main advantages of DBSCAN is its ability to identify clusters of arbitrary shape, unlike methods such as k-means that assume a spherical shape of clusters.

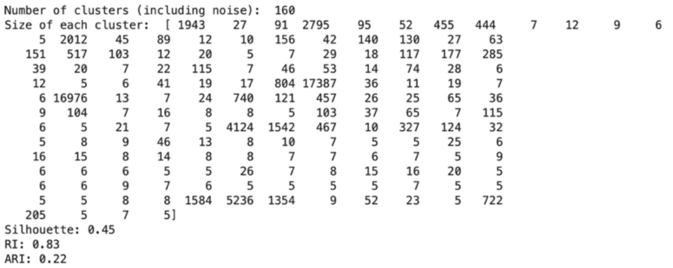
However, DBSCAN also has some disadvantages. The performance of DBSCAN is highly dependent on the choice of ε and MinPts parameters, which can be difficult to determine a priori. In addition, the algorithm may be inefficient on large or high-dimensional datasets, since it requires the evaluation of distances between all points. Another limitation is that DBSCAN may have difficulty detecting clusters in datasets with varying densities, since a single value of ε may not be adequate for all regions of the dataset.

DBSCAN uses two fundamental parameters:

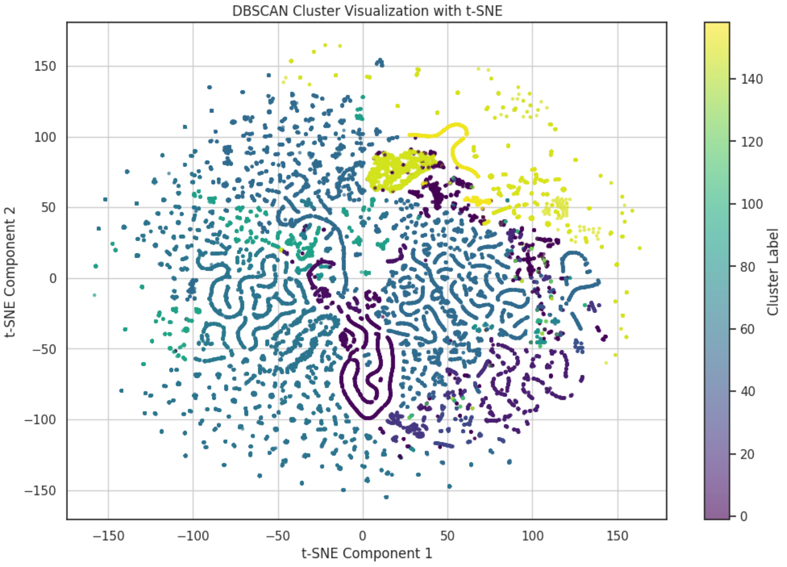
1. Epsilon (ε): The maximum distance within which to search for nearby points to consider a point as part of a cluster. A small ε creates many small clusters; a large ε can merge distinct clusters.
2. MinPts (Minimum Points): The minimum number of points required to form a cluster. A low MinPts generates small and noise-sensitive clusters; a high MinPts requires greater density to form a cluster.

A point is classified as a Core Point if it has at least MinPts points within the distance ε, as a Border Point if it is reachable from a core point but has fewer than MinPts points within the distance ε, and as a Noise Point if it is neither a core point nor a border point.

Applying DBSCAN using default parameters



```
Number of clusters (including noise):  160
Size of each cluster:  [ 1943    27    91  2795    95    52   455   444     7    12     9     6
     5  2012    45    89    12    10   156    42   140   130    27    63
   151   517   103    12    20     5     7    29    18   117   177   285
    39    20     7    22   115     7    46    53    14    74    28     6
    12     5     6    41    19    17   804 17387    36    11    19     7
     6 16976    13     7    24   740   121   457    26    25    65    36
     9   104     7    16     8     8     5   103    37    65     7   115
     6     5    21     7     5  4124  1542   467    10   327   124    32
     5     8     9    46    13     8    10     7     5     5    25     6
    16    15     8    14     8     8     7     7     6     7     5     9
     6     6     6     5     5    26     7     8    15    16    20     5
     6     6     9     7     6     5     5     5     5     7     5     5
     5     5     8     8  1584  5236  1354     9    52    23     5   722
   205     5     7     5]
Silhouette: 0.45
RI: 0.83
ARI: 0.22
```

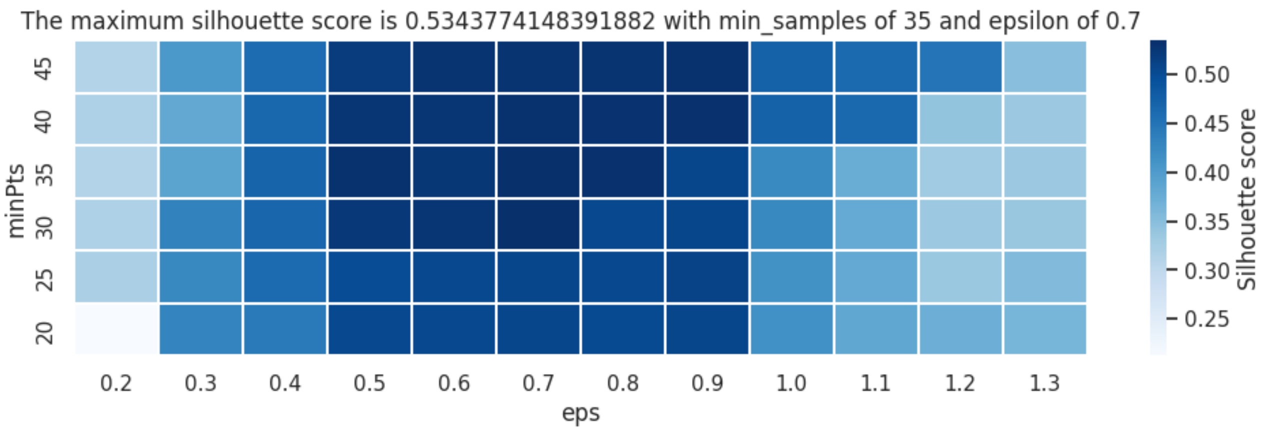


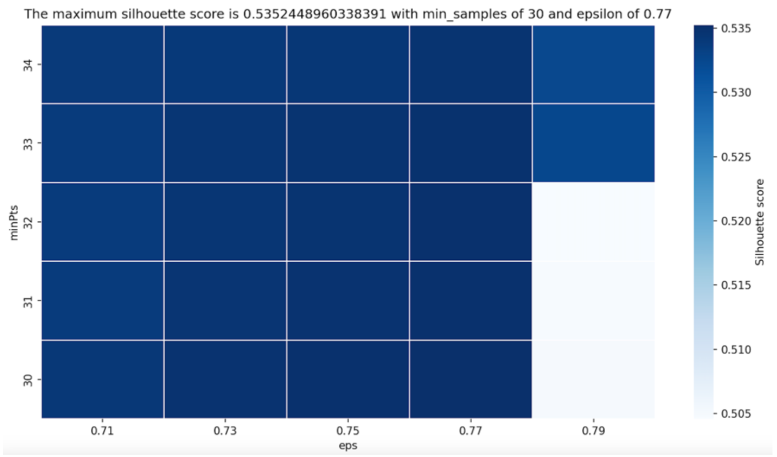
We chose to apply the DBSCAN algorithm without pre-selecting optimal parameters in order to emphasize the importance of parameter choice in the clustering process. Specifically, the analysis produced a total of 160 clusters, a considerably high number that emphasizes an overly fragmented distribution of the data. This phenomenon is further evidenced by relatively low values for both the silhouette coefficient and the Adjusted Rand Index (ARI), both indicators of suboptimal clustering quality.

Looking at the graph regarding the distribution of samples in the various clusters, it can be seen that two clusters predominate, aggregating most of the samples. This situation is not conducive to a balanced and effective representation of the heterogeneity of the data, compromising the effectiveness of the clustering model in capturing the structural subtleties of the analyzed dataset.

Determine best parameters of DBSCAN

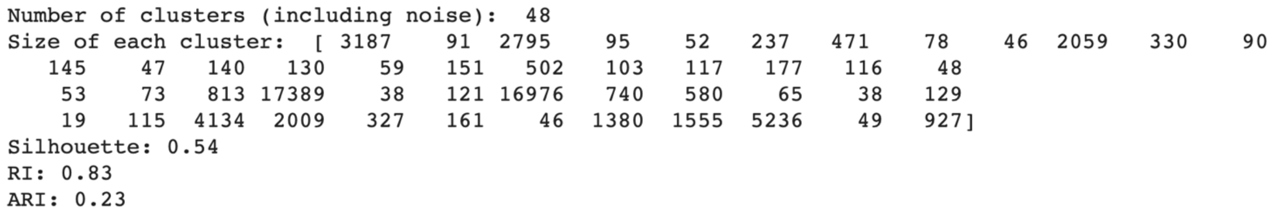
Per affinare la selezione dei parametri epsilon e min_samples del DBSCAN, abbiamo scelto di condurre una grid search in due fasi. Nella prima fase, abbiamo esplorato un ampio intervallo per entrambi i parametri: epsilon variava da 0.2 a 1.3 e min_samples da 20 a 45, aumentando di 5 unità per passo. Questo ci ha permesso di identificare un'area di interesse preliminare in base alla formazione dei cluster. Successivamente, nella seconda fase, abbiamo ristretto la ricerca focalizzandovi su un intervallo più specifico per epsilon, da 0.71 a 0.8 con incrementi di 0.02, e considerando solo valori per min_samples, da 30 a 34. Questa metodologia a due livelli non solo ci ha aiutato a ottimizzare i tempi di elaborazione ma anche a concentrare l'analisi sugli intervalli di parametri più promettenti.


The maximum silhouette score is 0.5343774148391882 with min_samples of 35 and epsilon of 0.7


The maximum silhouette score is 0.5352448960338391 with min_samples of 30 and epsilon of 0.77

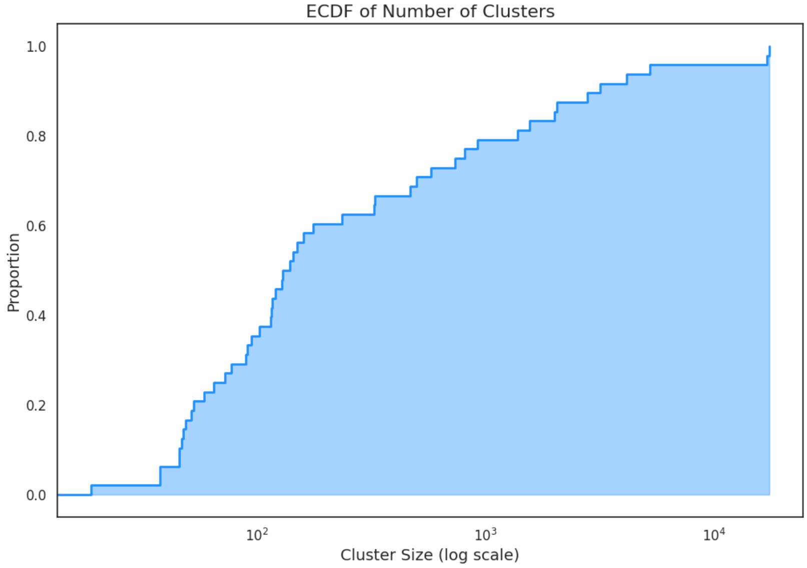
Following this detailed cross analysis, we selected the parameters with the highest silhouette value, namely min_sample equal to 30 and epsilon equal to 0.77.

Applying DBSCAN using best parameters



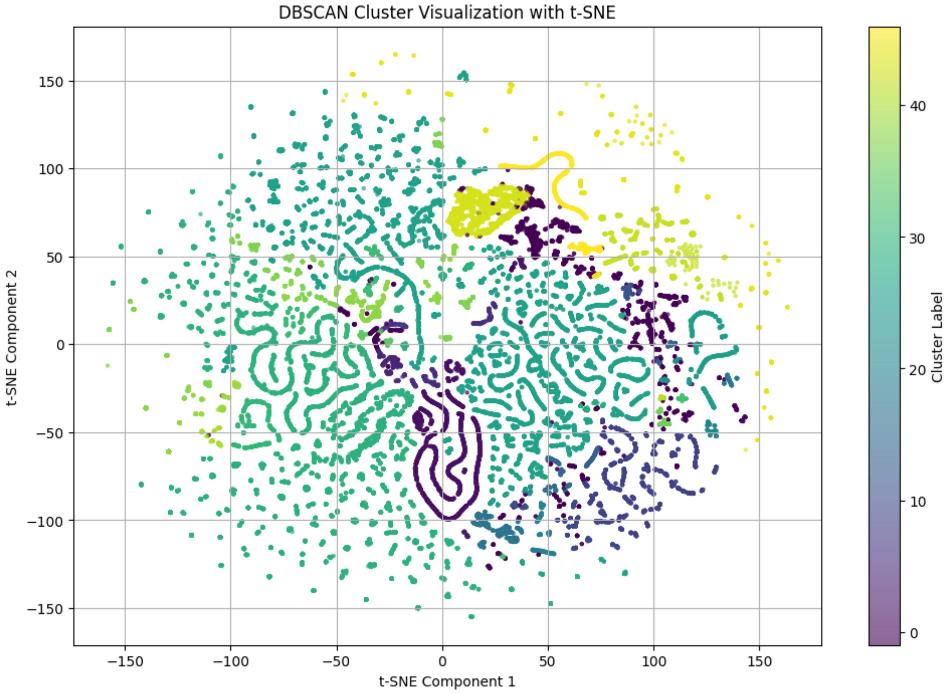
```
Number of clusters (including noise):  48
Size of each cluster:  [ 3187    91  2795    95    52   237   471    78    46  2059   330    90
   145    47   140   130    59   151   502   103   117   177   116    48
    53    73   813 17389    38   121 16976   740   580    65    38   129
    19   115  4134  2009   327   161    46  1380  1555  5236    49   927]
Silhouette: 0.54
RI: 0.83
ARI: 0.23
```

ECDF of number of cluster



The graph shows considerable variability in cluster size, revealing the presence of both many very small clusters and some very large ones. In particular, the initial part of the curve shows a rapid succession of increments, indicating that a large number of clusters contain few points, suggesting possible excessive fragmentation or the presence of many outliers. Toward the end of the curve, the gradual flattening of the ECDF reveals that a few large clusters enclose a significant proportion of the data points.

Cluster Visualization



From the analysis of the DBSCAN cluster visualization with t-SNE, key characteristics emerge that outline the effectiveness and peculiarities of this clustering approach. DBSCAN has distinguished a considerable variety of groups, indicating its ability to identify density-based clusters in a complex dataset. The presence of numerous clusters suggests that the method has been able to recognize various data densities, isolating areas of high concentration. Moreover, DBSCAN is effective at signaling noise points, which are data that do not aggregate into dense clusters, highlighting its utility for identifying anomalies or atypical data. These results indicate that the parameters `min_samples=30` and `epsilon=0.77` have effectively contributed to delineating complex data structures, although they may require further optimization to refine cluster separation or reduce noise.

We note that the clusters in the t-SNE visualization described in the previous paragraphs, resulting from the application of DBSCAN with default parameters, appear scattered and less defined, with a prevalence of noise characteristic of suboptimal parameters for distinguishing data densities. In contrast, the t-SNE with optimized parameters shows well-separated and distinct clusters. This indicates that a proper selection of values for epsilon and min_samples has significantly improved DBSCAN's ability to identify and separate clusters based on density, reducing noise and enhancing the readability of the data structure.