CLUSTERIZAÇÃO

O que é Clusterização?

“Clusterização é a classificação não-supervisionada de dados, formando agrupamentos ou clusters. Ela representa uma das principais etapas de processos de análise de dados, denominada análise de clusters (JAIN et al., 1999).

A análise de clusters envolve, portanto, a organização de um conjunto de padrões (usualmente representados na forma de vetores de atributos ou pontos em um espaço multidimensional – espaço de atributos) em clusters, de acordo com alguma medida de similaridade.

Intuitivamente, padrões pertencentes a um dado cluster devem ser mais “similares” entre si do que em relação a padrões pertencentes a outros clusters.

Dependendo da disponibilidade de modelos apropriados para os fenômenos responsáveis pela produção dos dados, a análise de dados pode ser exploratória (formulação de hipóteses e tomada de decisão) ou confirmatória (validação de modelos).

A clusterização está normalmente associada com a análise exploratória, pois envolve problemas em que há pouca informação a priori acerca dos dados (por exemplo, modelos estatísticos), e poucas hipóteses podem ser sustentadas.

É justamente a clusterização que pode fornecer novas hipóteses a respeito dos inter-relacionamentos dos dados e de sua estrutura intrínseca.”

(ftp://ftp.dca.fee.unicamp.br/pub/docs/vonzuben/ia368_02/topico5_02.pdf)

Validação do resultado de um processo de clusterização

“Como avaliar a qualidade da saída produzida por um processo de clusterização? O que caracteriza bons e maus processos de clusterização? O fato é que todo algoritmo de clusterização vai produzir clusters a partir dos padrões de entrada. Sendo assim, caso os dados de entrada não contenham clusters, então eles não devem ser processados por um algoritmo de clusterização.”

(ftp://ftp.dca.fee.unicamp.br/pub/docs/vonzuben/ia368_02/topico5_02.pdf)

FERRAMENTAS

https://scikit-learn.org/stable/modules/clustering.html

COMPARAÇÃO ENTRE OS ALGORITMOS

Method name	Parameters	Scalability	Usecase	Geometry (metric used)
K-Means	number of clusters	Very large `n_samples`, medium `n_clusters` with MiniBatch code	General-purpose, even cluster size, flat geometry, not too many clusters	Distances between points
Affinity propagation	damping, sample preference	Not scalable with n_samples	Many clusters, uneven cluster size, non-flat geometry	Graph distance (e.g. nearest-neighbor graph)
Mean-shift	bandwidth	Not scalable with `n_samples`	Many clusters, uneven cluster size, non-flat geometry	Distances between points
Spectral clustering	number of clusters	Medium `n_samples`, small `n_clusters`	Few clusters, even cluster size, non-flat geometry	Graph distance (e.g. nearest-neighbor graph)
Ward hierarchical clustering	number of clusters	Large `n_samples` and `n_clusters`	Many clusters, possibly connectivity constraints	Distances between points
Agglomerative clustering	number of clusters, linkage type, distance	Large `n_samples` and `n_clusters`	Many clusters, possibly connectivity constraints, non Euclidean distances	Any pairwise distance
DBSCAN	neighborhood size	Very large `n_samples`, medium `n_clusters`	Non-flat geometry, uneven cluster sizes	Distances between nearest points
Gaussian mixtures	many	Not scalable	Flat geometry, good for density estimation	Mahalanobis distances to centers
Birch	branching factor, threshold, optional global clusterer.	Large `n_clusters` and `n_samples`	Large dataset, outlier removal, data reduction.	Euclidean distance between points

https://scikit-learn.org/stable/modules/clustering.html

CÓDIGO

Vamos executar o código da página:
https://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html

```
    # ============
    # Create cluster objects
    # ============
    ms = cluster.MeanShift(bandwidth=bandwidth, bin_seeding=True)
    two_means = cluster.MiniBatchKMeans(n_clusters=params['n_clusters'])
    ward = cluster.AgglomerativeClustering(
        n_clusters=params['n_clusters'], linkage='ward',
        connectivity=connectivity)
    spectral = cluster.SpectralClustering(
        n_clusters=params['n_clusters'], eigen_solver='arpack',
        affinity="nearest_neighbors")
    dbscan = cluster.DBSCAN(eps=params['eps'])
    affinity_propagation = cluster.AffinityPropagation(
        damping=params['damping'], preference=params['preference'])
    average_linkage = cluster.AgglomerativeClustering(
        linkage="average", affinity="cityblock",
        n_clusters=params['n_clusters'], connectivity=connectivity)
    birch = cluster.Birch(n_clusters=params['n_clusters'])
    gmm = mixture.GaussianMixture(
        n_components=params['n_clusters'], covariance_type='full')

    clustering_algorithms = (
        ('MiniBatchKMeans', two_means),
        ('AffinityPropagation', affinity_propagation),
        ('MeanShift', ms),
        ('SpectralClustering', spectral),
        ('Ward', ward),
        ('AgglomerativeClustering', average_linkage),
        ('DBSCAN', dbscan),
        ('Birch', birch),
        ('GaussianMixture', gmm)
    )
```

EXERCÍCIO

Epileptic Seizure Recognition Data Set

Download: Data Folder, Data Set Description

Abstract: This dataset is a pre-processed and re-structured/reshaped version of a very commonly used dataset featuring epileptic seizure detection.

Data Set Characteristics:	Multivariate, Time-Series	**Number of Instances:**	11500	**Area:**	Life
Attribute Characteristics:	Integer, Real	**Number of Attributes:**	179	**Date Donated**	2017-05-24
Associated Tasks:	Classification, Clustering	**Missing Values?**	N/A	**Number of Web Hits:**	50488

Agora que já entendemos um pouco mais sobre clusterização, vamos fazer uma clusterização do dataset https://archive.ics.uci.edu/ml/datasets/Epileptic+Seizure+Recognition# . Neste dataset tempos a coluna Y que indica a classe [1,2,3,4,5]. Devemos retirar esta coluna e trabalhar apenas com os dados restantes. Ao final, devemos comparar o resultado da clusterização com o resultado real.