[Text classification](https://monkeylearn.com/what-is-text-classification/" \t "_blank) (a.k.a. text categorization or text tagging) is the task of assigning a set of predefined categories to [open-ended text](https://monkeylearn.com/blog/open-ended-questions/). Text classifiers can be used to organize, structure, and categorize pretty much any kind of text – from documents, medical studies and files, and all over the web. For example, new articles can be organized by topics; support tickets can be organized by urgency; chat conversations can be organized by language; [brand mentions can be organized by sentiment](https://monkeylearn.com/blog/brand-sentiment/); and so on.

Text classification is one of the fundamental tasks in [natural language processing](https://monkeylearn.com/natural-language-processing/) with broad applications such as [sentiment analysis](https://monkeylearn.com/sentiment-analysis/), topic labeling, spam detection, and intent detection.

Here’s an example of how it works:

“The user interface is quite straightforward and easy to use.”

A [text classifier](https://monkeylearn.com/text-classifiers/) can take this phrase as an input, analyze its content, and then automatically assign relevant tags, such as UI and Easy To Use.

How Does Text Classification Work?

Text classification can be done two different ways: manual or automatic classification. In the former, a human annotator interprets the content of text and categorizes it accordingly. This method can usually provide quality results but it’s time-consuming and expensive. The latter applies [machine learning](https://monkeylearn.com/blog/gentle-guide-to-machine-learning/), natural language processing (NLP), and other AI-guided techniques to automatically classify text in a faster, more cost-effective, and more accurate manner.

There are many approaches to automatic NLP text classification, which fall into into three types of systems:

* Rule-based systems
* Machine learning-based systems
* Hybrid systems

### **Rule-based systems**

Rule-based approaches classify text into organized groups by using a set of handcrafted linguistic rules. These rules instruct the system to use semantically relevant elements of a text to identify relevant categories based on its content. Each rule consists of an antecedent or pattern and a predicted category.

Say that you want to classify news articles into two groups: Sports and Politics. First, you’ll need to define two lists of words that characterize each group (e.g., words related to sports such as football, basketball, LeBron James, etc., and words related to politics, such as Donald Trump, Hillary Clinton, Putin, etc.). Next, when you want to classify a new incoming text, you’ll need to count the number of sport-related words that appear in the text and do the same for politics-related words. If the number of sports-related word appearances is greater than the politics-related word count, then the text is classified as Sports and vice versa.

For example, this rule-based system will classify the headline “When is LeBron James' first game with the Lakers?” as Sports because it counted one sports-related term (LeBron James) and it didn’t count any politics-related terms.

Rule-based systems are human comprehensible and can be improved over time. But this approach has some disadvantages. For starters, these systems require deep knowledge of the domain. They are also time-consuming, since generating rules for a complex system can be quite challenging and usually requires a lot of analysis and testing. Rule-based systems are also difficult to maintain and don’t scale well given that adding new rules can affect the results of the pre-existing rules.

Jfast text

-learning data set from Alexandara(one sheet from the Canada document)

-data set is very littke (precision of 0.019)

\_ Dbpedia petrained model

Company

EducationalInstitution

Artist

Athlete

OfficeHolder

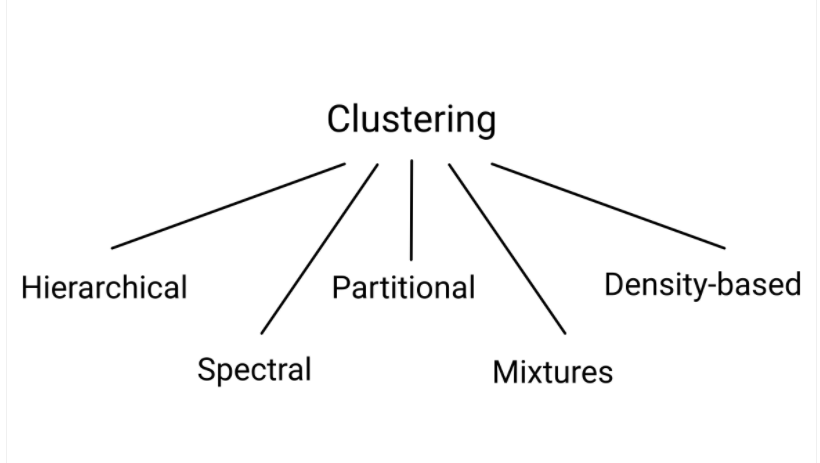
MeanOfTransportation Building NaturalPlace Village Animal Plant Album Film WrittenWork Decide to find other approach

Clusturing

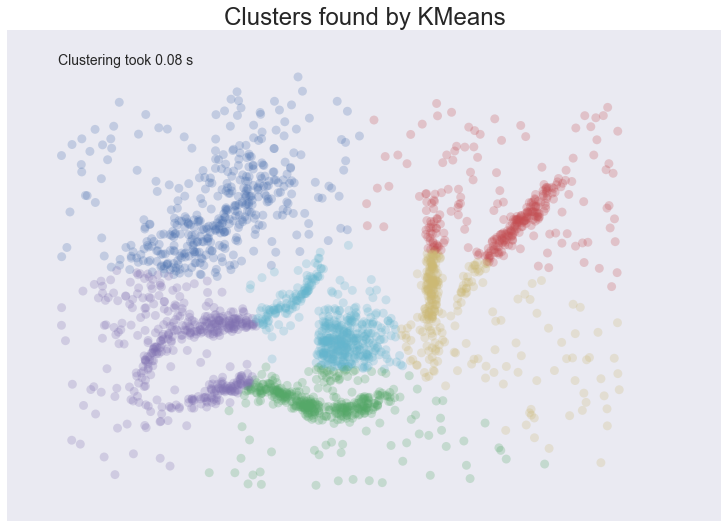
The first intuition when looking at raw data is to try to find patterns. Clustering similar information together makes it easier to understand what’s going on inside the data.

**Clustering is the task of organizing unlabelled objects in a way that objects in the same group are similar to each other and dissimilar to those in other groups**. In other words, clustering is like unsupervised classification where the algorithm models the similarities instead of the boundaries.

It has been widely used for various tasks, including: opinion mining on social media to detect a tendency in posts, image segmentation where the goal is to detect the boundaries of any object, customer profiling based on product purchases, and document clustering based on phrases or words. But you can also use it to label data in place of a person, allowing the human to go from a supervising role to a validating role.

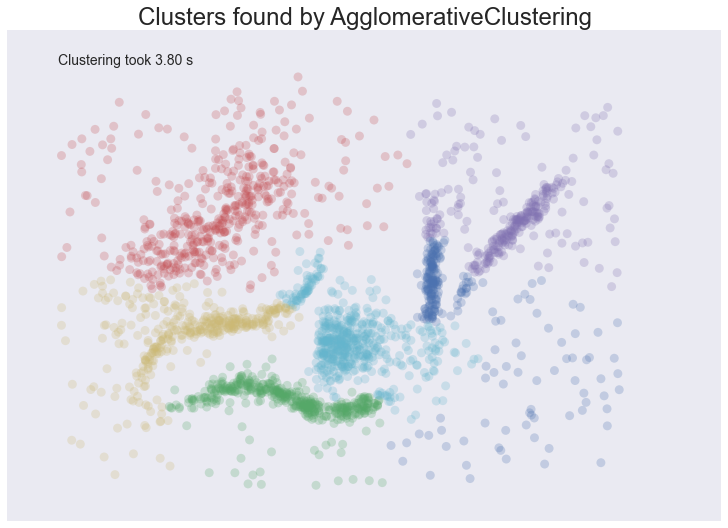


Those three requirements are the baseline of a clustering task, and the criterion function depends on the approach you selected for your clustering experiment. From now on, clustering methods will only be branching out, and covering each algorithm would be too tedious. Instead, we are going to look at three algorithms distributed over three major approaches: partitional, hierarchical and density-based.

[](https://mk0caiblog1h3pefaf7c.kinstacdn.com/wp-content/uploads/2017/01/recast-ai-kmeans.png)

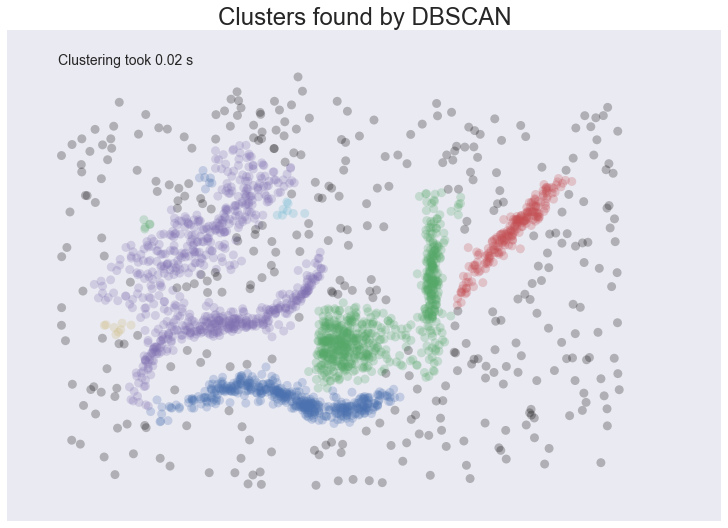
Clusters obtained with K-means on a toy dataset, courtesy of Leland McInnes, John Healy, Steve Astels

K-means is THE go-to clustering algorithm. Fast, available and easy to wrap your head around, it requires you to know the number of clusters your expect. One downside is that K-means tends to assume that your clusters will be simple, due to its partitional approach: it tries to decompose the dataset into non-overlapping subsets. Expect quick results, but with noise.

[](https://mk0caiblog1h3pefaf7c.kinstacdn.com/wp-content/uploads/2017/01/recast-ai-agglomerative-clustering.png)

Clusters obtained with Agglomerative Clustering on a toy dataset, courtesy of Leland McInnes, John Healy, Steve Astels

On the other hand, hierarchical clustering tends to model the dataset into clusters organized as a tree, where you can go up from one document to the whole corpus. This can be done from top to bottom (divisive) or bottom to top (agglomerative). Agglomerative clustering is an approach that yields a dendrogram (a tree diagram) of your dataset, for you to cut at the desired threshold. This leaves you with the freedom to view the assumed clusters before selecting them, but they will still be noisy.

[](https://mk0caiblog1h3pefaf7c.kinstacdn.com/wp-content/uploads/2017/01/recast-ai-dbscan.png)

Clusters obtained with DBSCAN on a toy dataset, courtesy of Leland McInnes, John Healy, Steve Astels

Finally, density-based clustering will create clusters on the denser regions of your dataset. DBSCAN (and its improvement HDBSCAN) combines the best of agglomerative clustering with the capacity of removing noisy documents. The only parameter you have to select is the minimal distance to consider two documents as similar, and DBSCAN will do the rest.

If you want a more complete comparison of different clustering methods, you can find one [here](http://hdbscan.readthedocs.io/en/latest/comparing_clustering_algorithms.html), with visuals too!

Here we covered the basics of clustering and listed several algorithms that can be used to find the best groups in our corpus. But for the algorithms to work, it is important to feed it the data under the right light. In the next part, we’re going to talk about preprocessing techniques, which will help reduce noise, and facilitate data visualization.

## ****What is topic modeling?****

Topic modeling is a form of unsupervised learning that identifies hidden themes in data.

Being unsupervised, topic modeling doesn’t need labeled data. It can be applied directly to a set of text documents to extract information.

Topic modeling works in an exploratory manner, looking for the themes (or topics) that lie within a set of text data. There is no prior knowledge about the themes required in order for topic modeling to work. It discovers topics using a probabilistic framework to infer the themes within the data based on the words observed in the documents.

## ****How topic modeling works****

### **Latent Dirichlet Allocation**

LDA was [developed in 2003](https://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf) by researchers David Blei, Andrew Ng and Michael Jordan. Its simplicity, intuitive appeal and effectiveness have supported its strong growth.

LDA topic modeling discovers topics that are hidden (latent) in a set of text documents. It does this by inferring possible topics based on the words in the documents. It uses a generative probabilistic model and Dirichlet distributions to achieve this.

The inference in LDA is based on a Bayesian framework. This allows the model to infer topics based on observed data (words) through the use of conditional probabilities.

## Bayesian Probability

Bayesian probability is an interpretation of the concept of probability, in which, instead of frequency or propensity of some phenomenon, probability is interpreted as reasonable expectation representing a state of knowledge or as quantification of a personal belief.

[Topic Modeling with LDA: An Intuitive Explanation - HDS (highdemandskills.com)](https://highdemandskills.com/topic-modeling-intuitive/)