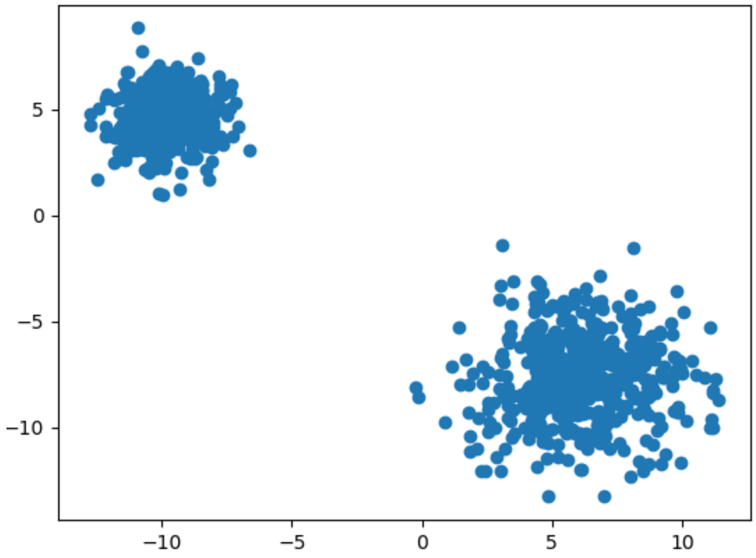
Machine Learning

Lab 5. Unsupervised learning

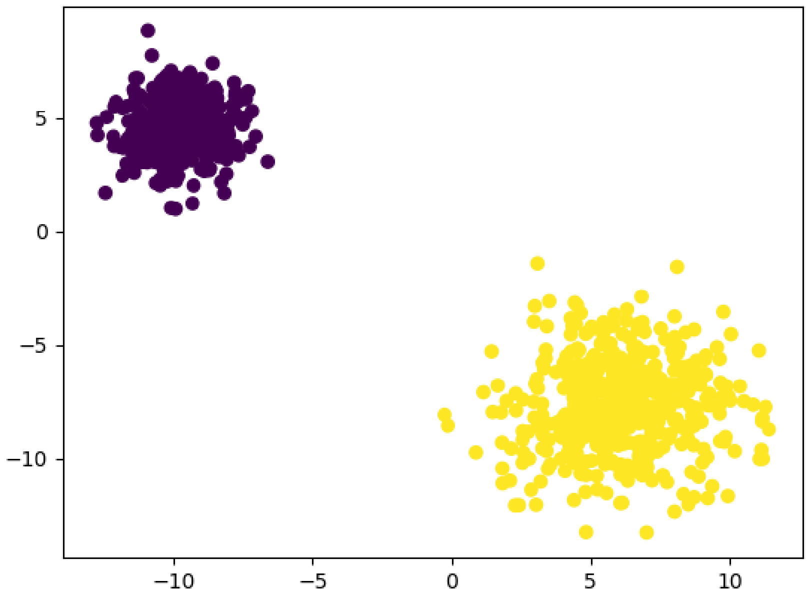
Unsupervised learning involves searching for patterns in data. It is also called unsupervised learning, as opposed to supervised learning, which is also known as learning with a teacher. In supervised learning, the "teacher" is the set of reference data - class labels in the training set for classification, reference values for dependent variables for regression, and so on. In unsupervised learning, no reference data are used. An example of this is the PCA method - principal component analysis.

There are many algorithms in the group of unsupervised methods, each with different goals. For example, PCA aims to decompose data/signals into components based on certain assumptions that were described earlier. There are also other decomposition algorithms, such as independent component analysis and dictionary learning. There are also unsupervised neural networks, such as restricted Boltzmann machines, whose learning is inspired by thermodynamics. Another important type of unsupervised convolutional neural network is autoencoders, whose architecture includes reducing the resolution of the image and then increasing it to the input level. The first layers can be considered as an encoder that compresses the image, and the last ones as decoders that reconstruct it from the encoded form. Such networks learn a hidden representation of the data and are used in dimensionality reduction, image processing, machine translation, and drug discovery.

A large group of supervised methods are clustering methods. They essentially involve classifying samples based on certain criteria, but without knowledge of their actual class membership. Let's imagine a data set consisting of 2,000 samples, each described by two features. We can visualize this set as a cloud of points.



In supervised learning, sample classification would be based on knowledge of their class membership. However, in this case, we only have sample features. Cluster analysis methods analyze the structure of the data set. When applied appropriately, they are able to identify regularities, such as in the above example where samples form two dense clusters. The result of clustering using, for example, the k-means method, would give the following result:



Samples have been automatically assigned to one of two classes, which in the case of clustering methods are usually referred to as clusters, to emphasize the nature of these methods. The aforementioned k-means method is one of the simplest and well-known methods for several decades. It groups samples into k clusters, where k is usually a predetermined number. However, there are methods that allow for the evaluation of the "quality" of clustering, which can be determined for different values of k and find the best result.

Task 1: Go to the website

https://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html

and run a program that tests different clustering methods. Analyze the features of these methods - their parameters, typical applications, and metrics they use - summarized in a table in section 2.3.1 under the link: https://scikit-learn.org/stable/modules/clustering.html. Finally, try to add to this code the calculation of clustering quality using the Davies-Bouldin index, described in section 2.3.10.7 under the above link. Display the results in the form of a bar chart. Which method gave the best result according to this measure for different types of clusters? Do visual inspection of clustering correctness and Davies-Bouldin index give consistent results?

Task 2: Familiarize yourself with the k-means algorithm (k-means, https://en.wikipedia.org/wiki/K-means_clustering). Choose two other clustering methods and familiarize yourself with their operation in detail.

Task 3: Implement the k-means method on your own (without using ready-made implementations from the internet). To do this, generate artificially clustered data using the sklearn.datasets.make_blobs function. You can decide on the number, variance, and location of the generated clusters, as well as the number of samples in each of them. The implementation should allow you to change the target number of clusters k and work for any number of features - however, for convenient visualization, check the correctness of the method's operation for two or three features. You can generate initial cluster centroids using the np.random.uniform function, taking into account the low and high parameters of this function according to the range of variability of features in the data set (the goal is to generate centroids that will be located somewhere within the data points).

This instruction was based on, amongst others:

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

https://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html

https://en.wikipedia.org/wiki/Unsupervised_learning

https://en.wikipedia.org/wiki/Autoencoder

ChatGPT

This instruction was written in Polish by Jakub Jurek, then translated by ChatGPT.