**Unsupervised Learning:**

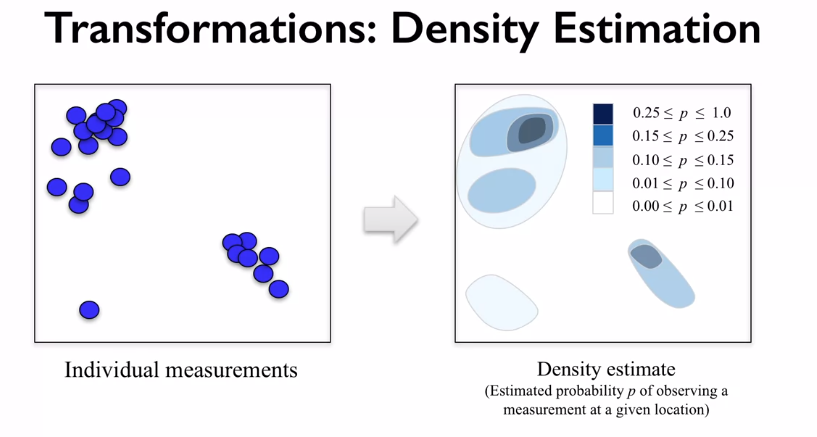
Unlike supervised learning there is no labelled target value to use for classification or regression. Unsupervised learning is used to try and capture structures or information.

Applications:

* Visualize structure of complex datasets.
* Density estimate to predict probability of events.
* Compress and summarize the data.
* Extract features for supervised learning.
* Discover important clusters or outliers.

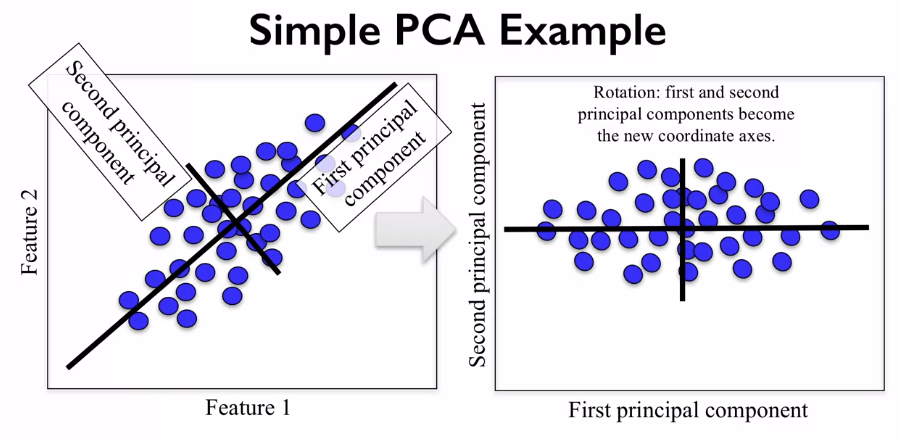
**Types of unsupervised learning covered in this document:**

1. **Transformations** – Process that extract compute information
2. **Clustering** – Find groups in the data, assign every point in the dataset to one of the groups.



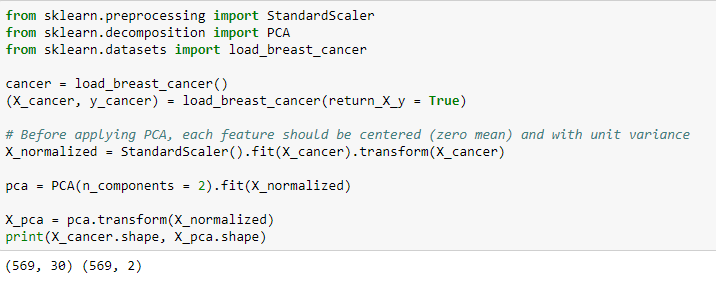
**Dimensionality Reduction:**

This is part of the transformative unsupervised learning algorithm family. This is a method that is very useful for datasets with very large features/dimensions. This method can be used to explore the dataset how certain features are grouped together. Below is a simple example of a dimensionality reduction of a 2d feature space: **PCA** combines features and tries to draw a best fit line that results in the lest loss in variance. From the line of best fit **PCA** used this line to produce new axes/features, this results in huge feature reduction but only small amounts of information loss.



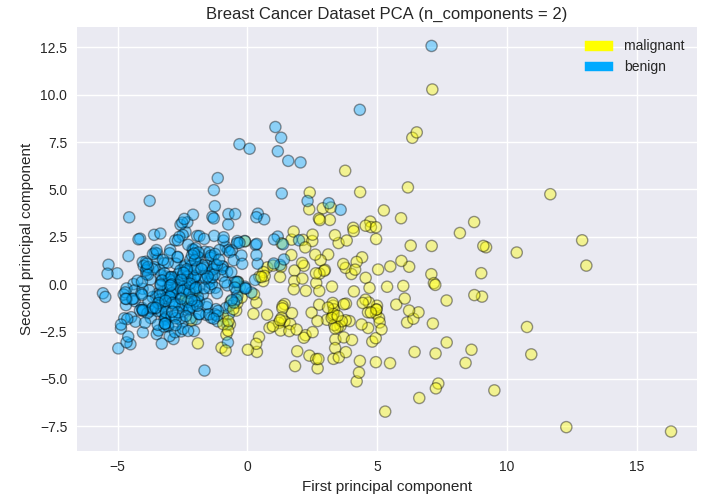
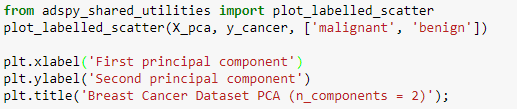
Before applying PCA, each feature should be centred (zero mean) and with unit variance.

Below is an example of PCA being used on the breast cancer dataset:

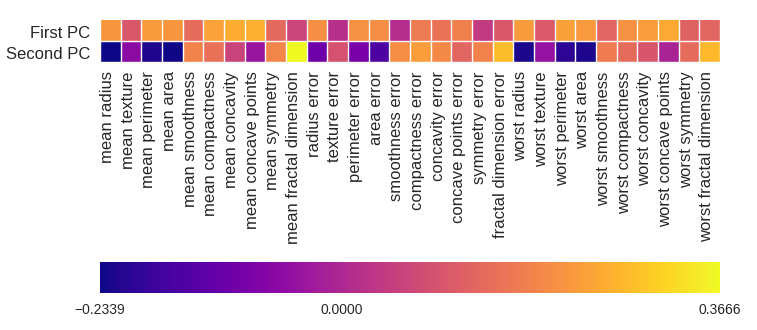
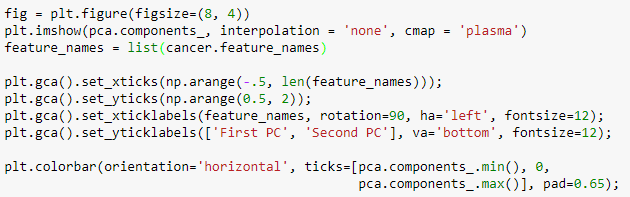


The output shows how we’ve reduced the feature space from 30 to just 2. These new 2 features represent all the data points in the original 30 feature space, but with slightly reduced information. Note how the X data is normalized to allow its use in PCA. Using “n\_componets” is an instruction to reduce the shape of features down to the given value. Because we have a feature space <= 3 we can plot this and see how well the dataset is being classified.

Below we can see how PCA can be used to form clusters in the data, applying a simple linear classifier would actually yield pretty good results! PCA can also be used to find informative features that could then be later used in a supervised learning stage.



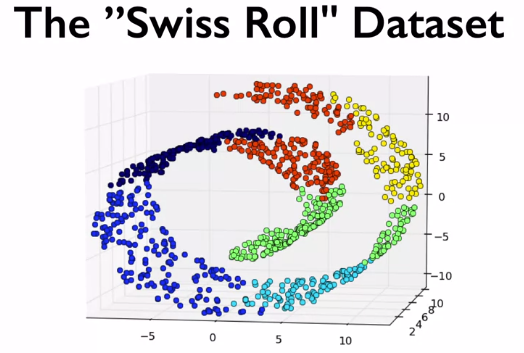
We can use heatmaps to visualize how each of the 30 features relates to the 2 principal components.



PCA is a good initial tool, but maybe not be capable of finding subtle groupings of complex datasets.

**Manifold Learning:**

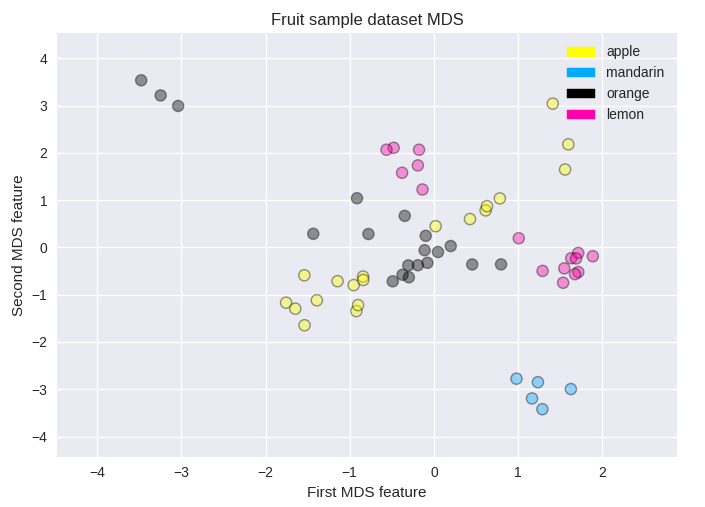
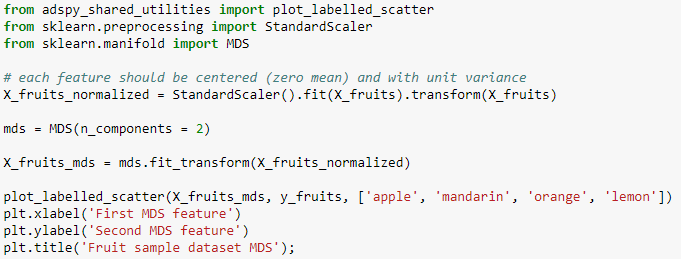
These are very good at finding low dimensional structures in high dimensional data and very useful in visualizations. Below is an example of a high dimensional dataset represented in high dimensional space:



The above shows a 2d sheet/lower dimensional space, representing different groups in a 3d/higher dimensional space. These lower dimensional sheets in higher spaces are called **manifolds**. PCA is not sophisticated enough to find this structure. One form of manifold learning is called **Multi-dimensional scaling (MDS).**

There are many forms of MDS, but they are all used to help visualize higher dimensional datasets in a way the preserves the information about the original points. In Scikit-learn this process is very similar to PCA.

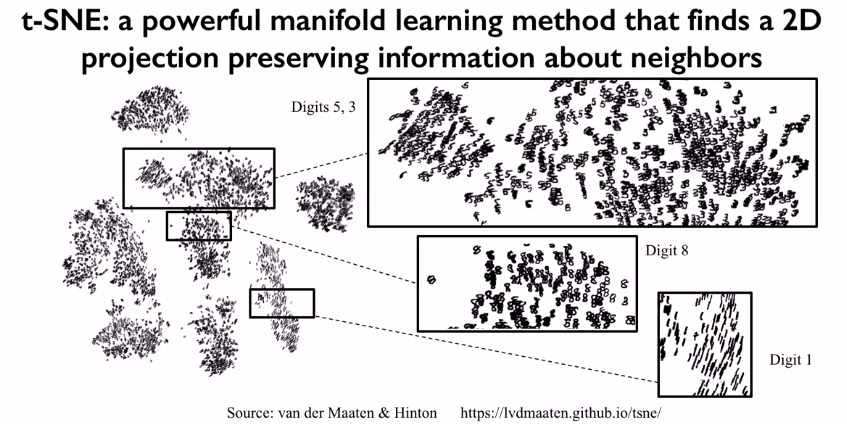
Using MDS in the Fruit dataset:



The fruits dataset originally has 4 features, and is complex to visualize this in a plot, however, using MDS we can easily see groups in our dataset. Applying something like a SVC with radial basis function would separate these groups nicely.

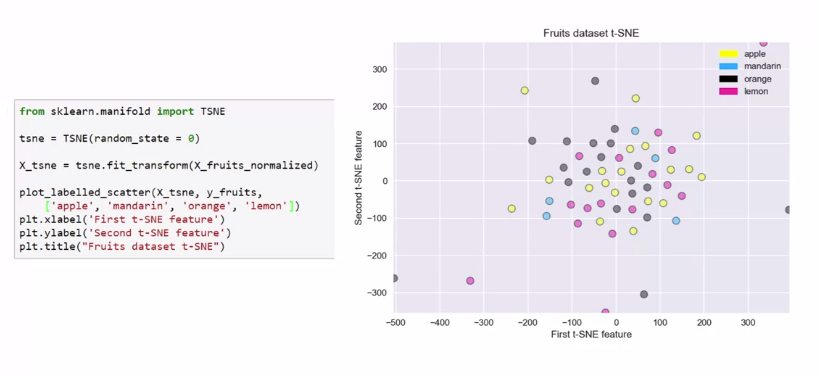
**t-SNE**:

t-SNE finds two-dimensional representations of your data, such that the distances between points in the 2d scatter plot match as closely as possible to the original high dimensional dataset. It gives more weighting to preserve distances between points that are neighbours. Below is an example of the t-SNE used for showing the digits dataset:



We can see that 5,3 and 8 all are close to each other, so this might result in misclassification of these digits for another one. E.g. predicting an 8 but its actually a 5.

Below we can see that t-SNE performs poorly on the fruit dataset, this shows us that we should try using a few different dimensionality reduction methods before deciding which one to use.



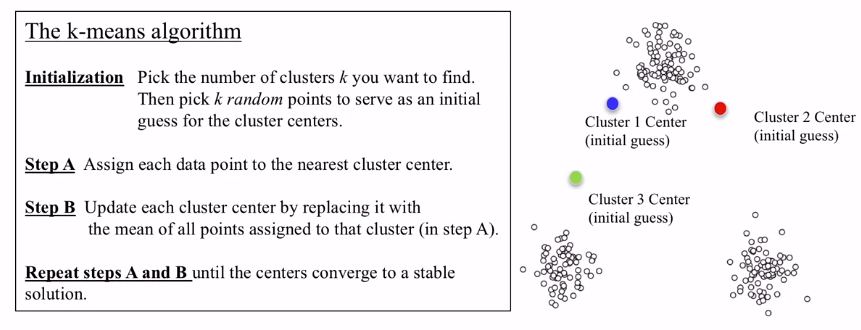
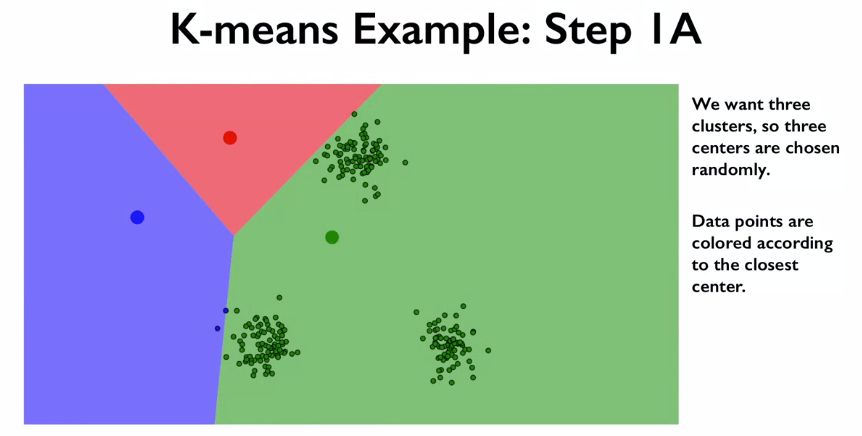
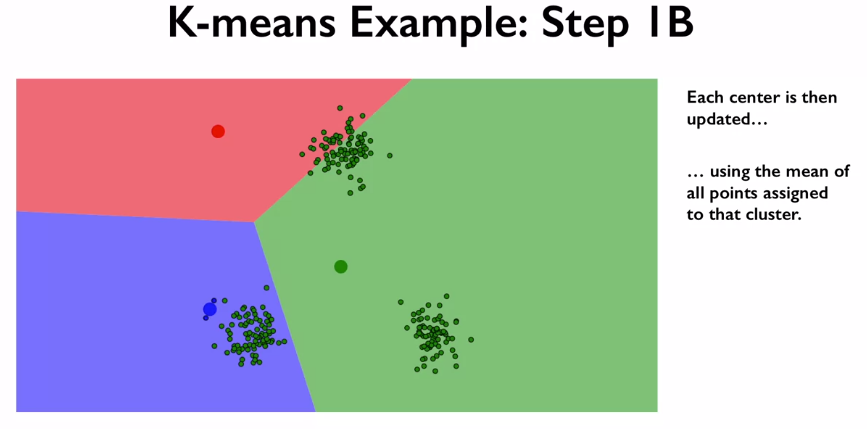
**Clustering:**

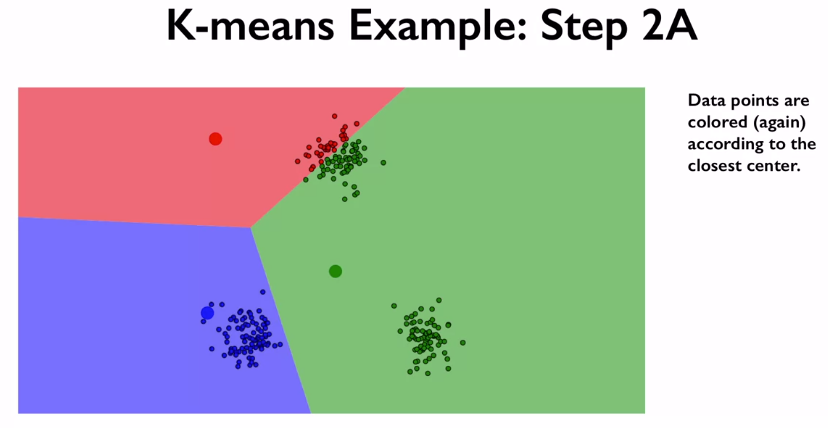
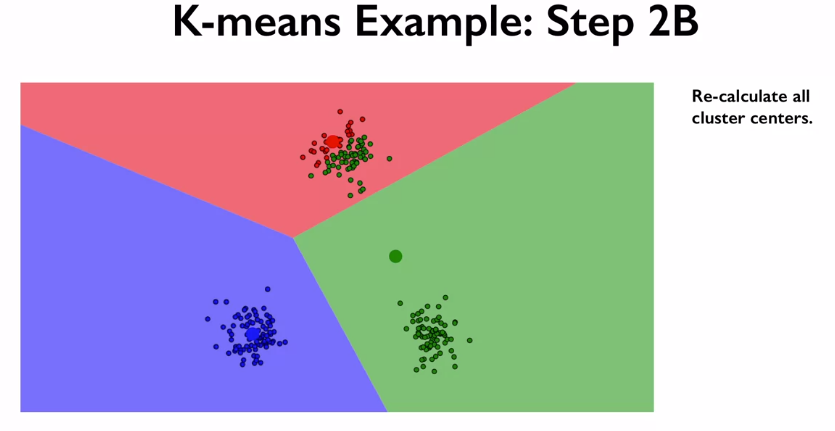
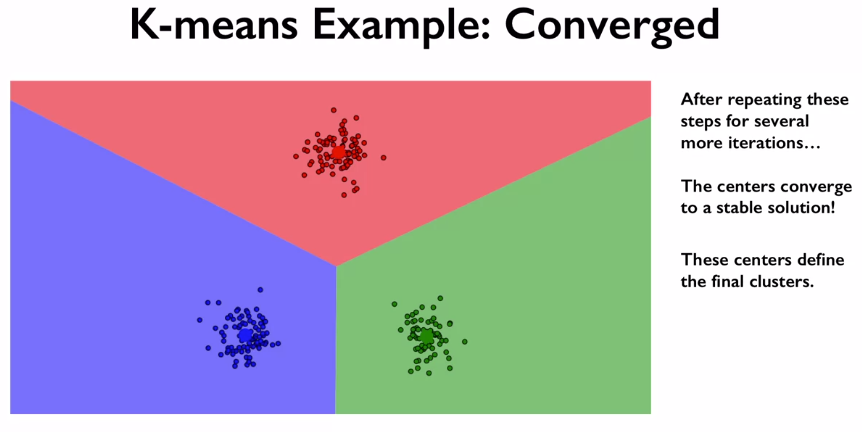
The purpose of this unsupervised learning technique is to group data together and form clusters. Some clustering algorithms can assign new data points to a specific cluster that it has already found. This is similar to classification, but the model hasn’t had access to the true labels, and we don’t know what these actual clusters are.

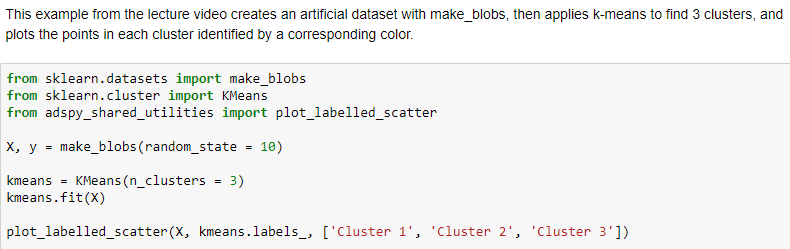
**K-Means Clustering:**

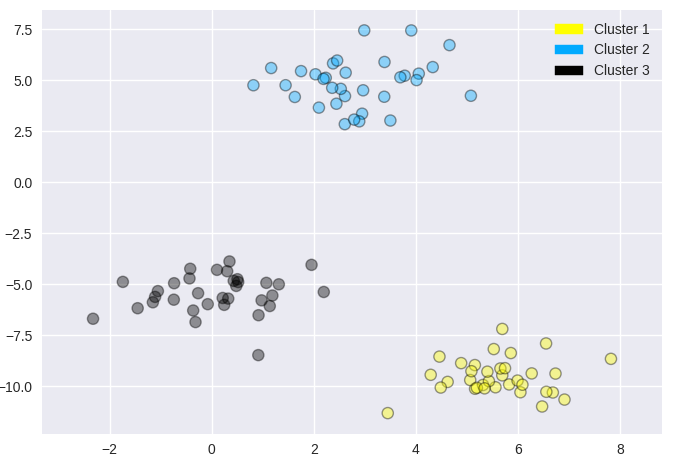
When using K-means you need to specify the number of clusters you want to generate, sometimes you might already know the classes that should come out from the data.

How does it work:

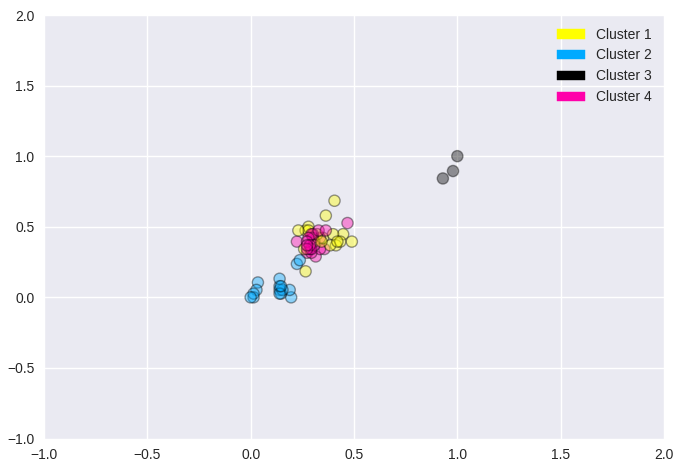
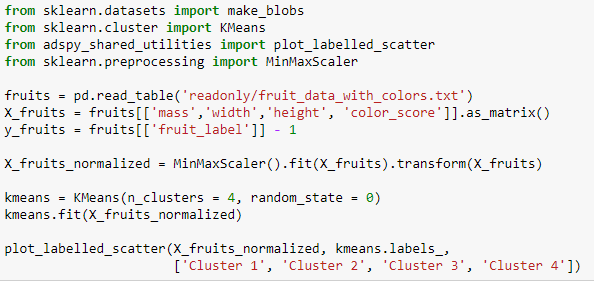








Applying the K-means algorithm to the fruit’s dataset, here we know that there are 4 different groups:

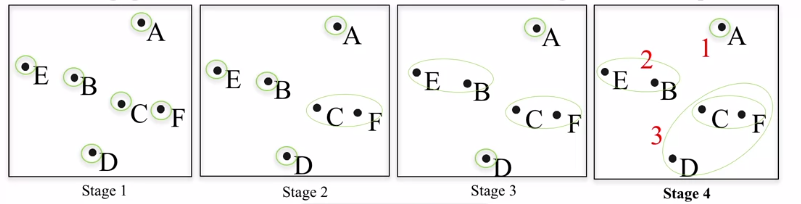


**Limitation of K-Mean Cluster:**

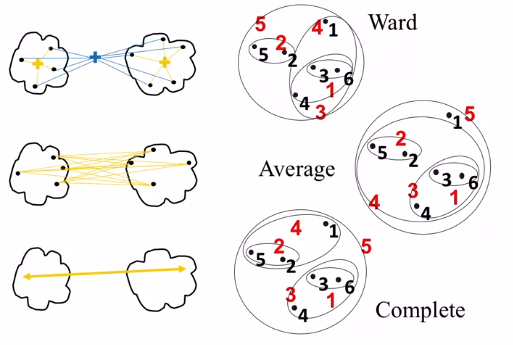
* Work well for only simple clusters that are the same size, and well-shaped.
* Doesn’t do well with complex clusters.
* Doesn’t work with categorical data sets, need to use another k-… method.

**Agglomerative Clustering:**

These work by doing an iterative bottom up approach. First each point is put into its own cluster. Then a sequence of clustering is done where the most similar clusters merge together in a new cluster. This continues until some stopping criteria is met. For Scikit-learn this it the number of clusters we define.

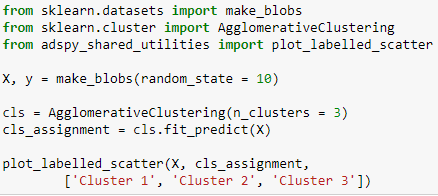


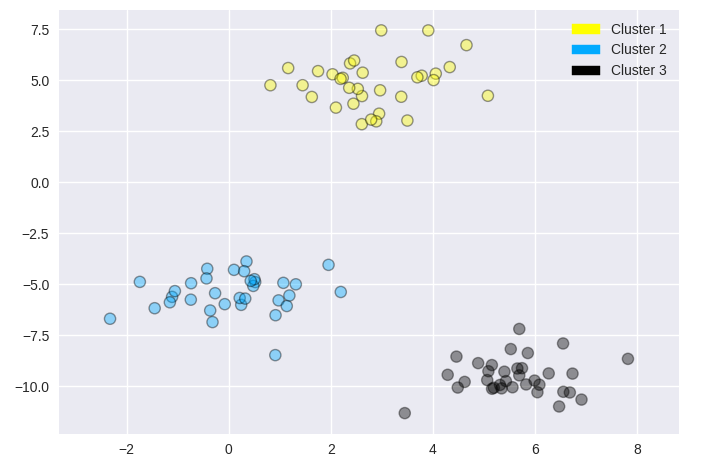
**Linkage** criteria are used to determine how similar two clusters are to each other before merging. In Scikit-learn there are 3 main types:



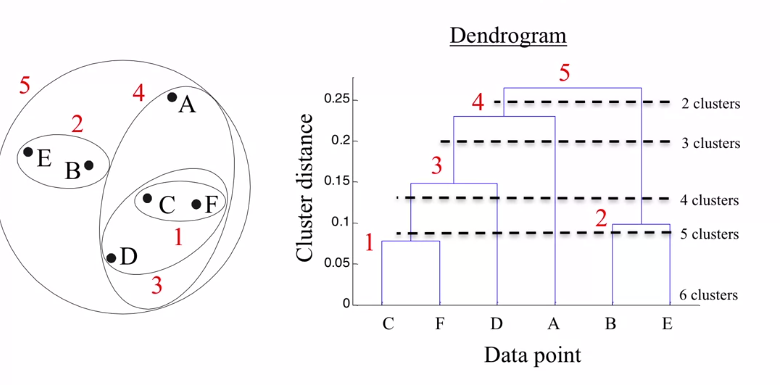
1. Ward: merges the two clusters that give the smallest increase in total variance within a cluster.
2. Average: merges two clusters that have the smallest average distance between them.
3. Complete: merges the two cluster that have the smallest maximum distance between their points.

Ward usually works well on most datasets, so this will normally be the first attempt. If there might be clusters of very different sizes, then its worth trying the other two.





Agglomerative clustering automatically sorts the data into **Hierarchical Clustering.** This can be used to visualize what’s called **dendrogram**, which shows the order in which clusters were formed. The difference in heights of the clusters tells us how far apart two clusters were when they merged, with the branch going up representing the next merge.

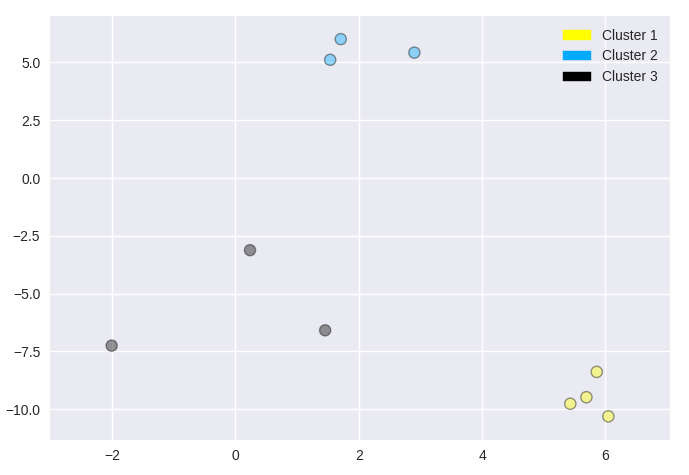


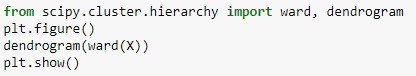
The red numbers above show us which clusters were formed on the dendrogram, and the dotted lines show us when the clusters were formed. E.g. starting at 5, only one cluster is formed, then 4 and 2 are formed, and so on…

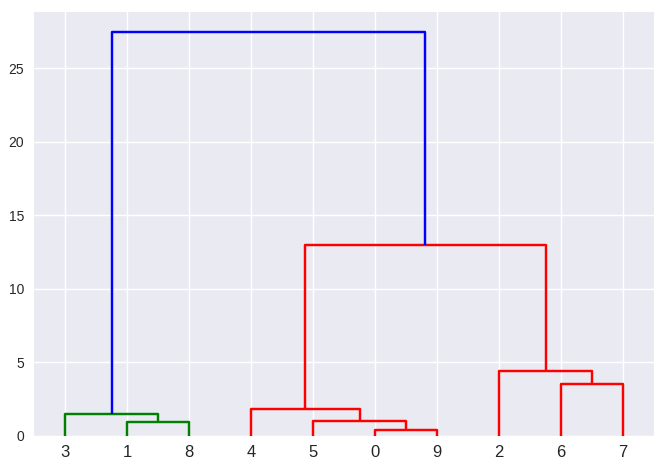
This height can help us figure out the right number of clusters. In general, we want clusters that have highly similar items within each cluster, but far apart from other clusters. For example, above we can see that going from 2 clusters to 3 happens at a fairly high Y value. This means the cluster that were merged were a significant distance apart.

We can plot these dendrograms using Scipy!









The ward function returns an array showing the distances spanned during the agglomerative clustering. This Ward array is then passed into the dendrogram to output the hierarchical plot. We can see that the most suitable clustering is 3 as going down further results in very small changes in Y.

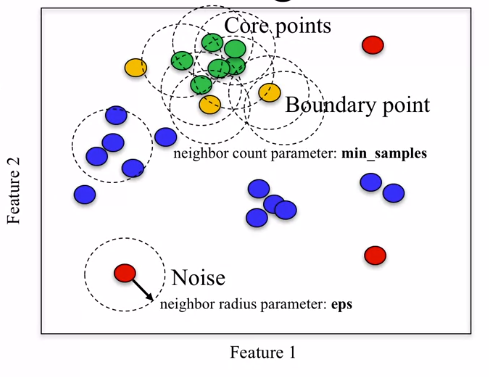
This type of clustering is particularly useful in biological data and genetics.

**DBSCAN Clustering:**

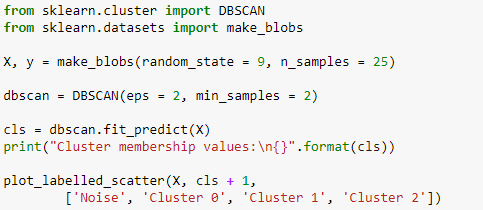
Density-based spatial clustering of applications with noise = DBSCAN.

The main advantage of this method is that you don’t need to specify the number of clusters in advance. It also works well with datasets that have more advanced cluster shapes. It is also good at finding data points that are outliers and shouldn’t be assigned to any clusters.

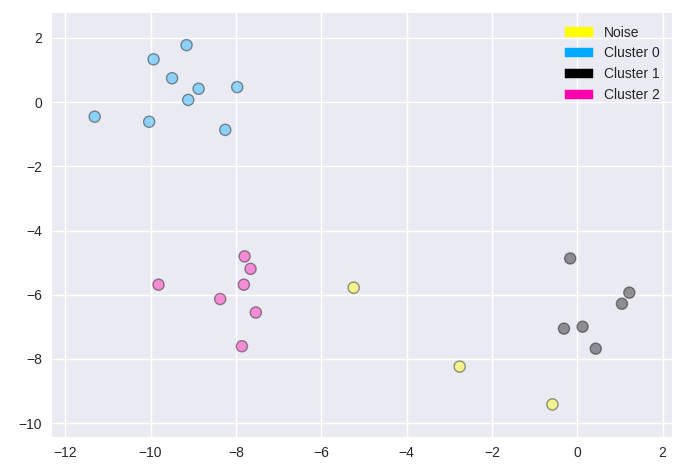
The main idea of DBSCAN is that clusters represent areas of more dense data points, while being separated by regions that are empty or at least less densely populated.



The two parameters of DBSCAN are min samples and eps. All points that lie in a denser are called core samples. For a given data point, if there are min sample of other data points that lie within a distance of eps, that given data points is labelled as a core sample. Then all core samples that are with a distance of eps units apart are put into the same cluster.







**Summary:**

Unlike supervised learning, where we have existing labels or target values to use for evaluating the effectiveness of the learning method, it can be difficult to evaluate unsupervised learning algorithms automatically.

Since there's typically no ground truth to compare against.

In some cases, as in the breast cancer example, we may have existing labels that can be used to evaluate the quality of the clusters by comparing the assignment of a data point to a cluster with the label assigned to the same data point.

But there are many cases where labels are not available. In addition, in the case of clustering, for example, there's ambiguity, in a sense that there are typically multiple clustering’s that could be plausibly assigned to a given data set. And none of them is obviously better than another unless we have some additional criteria. Such as, performance on the specific application task that does have an objective evaluation to use as a basis for comparison.

For example, in cases where the results of the clustering are used as features for supervised learning, we could use the overall classifier accuracy gain from adding these clustering-based features as a measure of success for the underlying clustering.

Another issue with evaluating clustering algorithms is that it can be hard to automatically interpret or label the meaning of the clusters that are found. And this is still a step that requires human expertise to judge.