KNeighborsClassifier() parameters

**n\_neighbors : *int, default=5***

Number of neighbors to use by default for [**kneighbors**](https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html#sklearn.neighbors.KNeighborsClassifier.kneighbors) queries.

**weights*{‘uniform’, ‘distance’} or callable, default=’uniform’***

weight function used in prediction. Possible values:

* ‘uniform’ : uniform weights. All points in each neighborhood are weighted equally.
* ‘distance’ : weight points by the inverse of their distance. in this case, closer neighbors of a query point will have a greater influence than neighbors which are further away.
* [callable] : a user-defined function which accepts an array of distances, and returns an array of the same shape containing the weights.

**algorithm*{‘auto’, ‘ball\_tree’, ‘kd\_tree’, ‘brute’}, default=’auto’***

Algorithm used to compute the nearest neighbors:

* ‘ball\_tree’ will use [**BallTree**](https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.BallTree.html#sklearn.neighbors.BallTree)
* ‘kd\_tree’ will use [**KDTree**](https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KDTree.html#sklearn.neighbors.KDTree)
* ‘brute’ will use a brute-force search.
* ‘auto’ will attempt to decide the most appropriate algorithm based on the values passed to [**fit**](https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html#sklearn.neighbors.KNeighborsClassifier.fit) method.

Note: fitting on sparse input will override the setting of this parameter, using brute force.

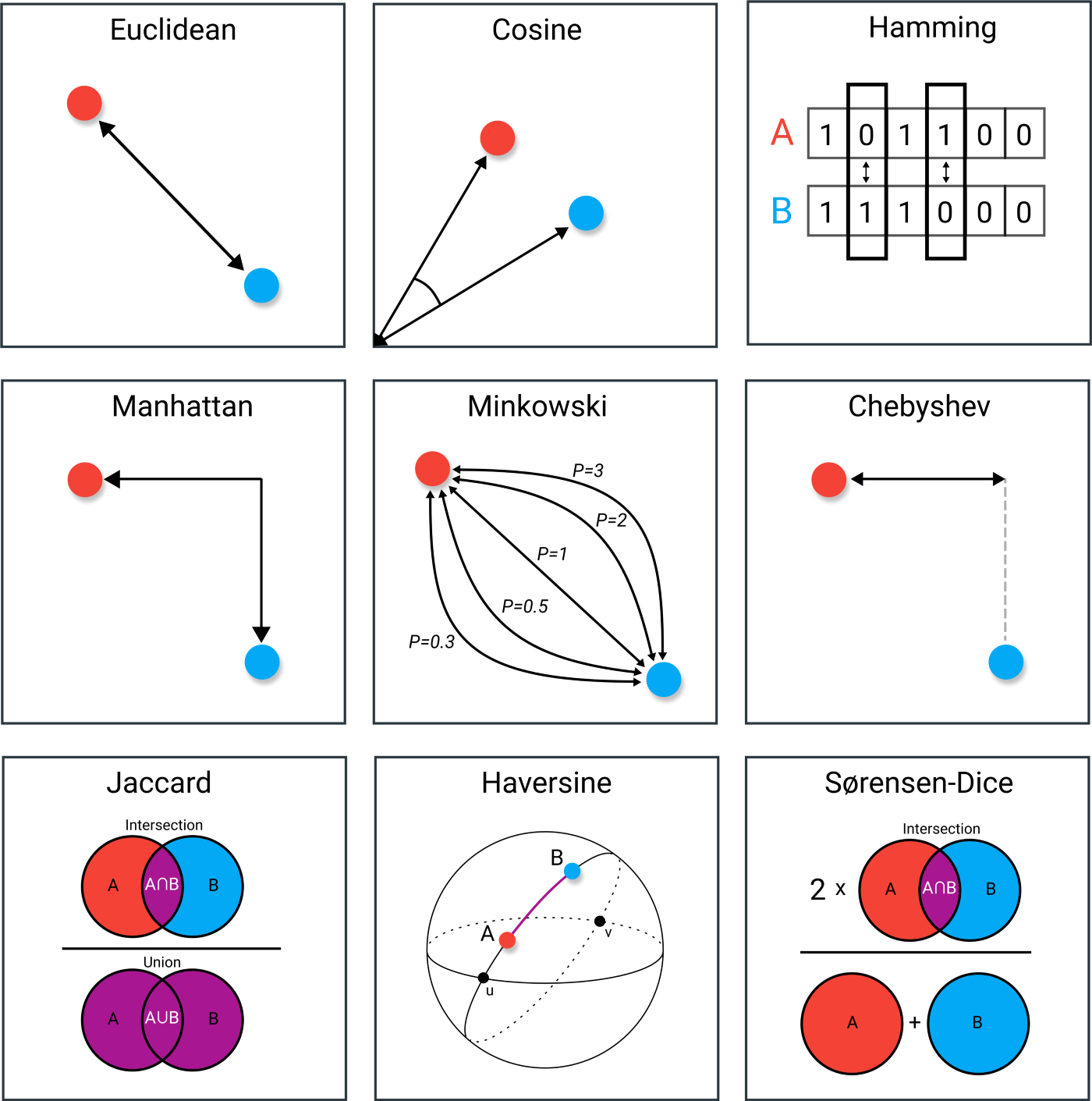
<https://www.youtube.com/watch?v=FnJj28u7rF0>

**leaf\_size : *int, default=30***

Leaf size passed to BallTree or KDTree. This can affect the speed of the construction and query, as well as the memory required to store the tree. The optimal value depends on the nature of the problem.

**p : *int, default=2***

Power parameter for the Minkowski metric. When p = 1, this is equivalent to using manhattan\_distance (l1), and euclidean\_distance (l2) for p = 2. For arbitrary p, minkowski\_distance (l\_p) is used.



**metric : *str or callable, default=’minkowski’***

the distance metric to use for the tree. The default metric is minkowski, and with p=2 is equivalent to the standard Euclidean metric. See the documentation of [DistanceMetric](https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.DistanceMetric.html#sklearn.neighbors.DistanceMetric) for a list of available metrics. If metric is “precomputed”, X is assumed to be a distance matrix and must be square during fit. X may be a [sparse graph](https://scikit-learn.org/stable/glossary.html#term-sparse-graph), in which case only “nonzero” elements may be considered neighbors.

**metric\_params : *dict, default=None***

Additional keyword arguments for the metric function.

**n\_jobs : *int, default=None***

The number of parallel jobs to run for neighbors search. None means 1 unless in a [**joblib.parallel\_backend**](https://joblib.readthedocs.io/en/latest/parallel.html#joblib.parallel_backend) context. -1 means using all processors. See [Glossary](https://scikit-learn.org/stable/glossary.html#term-n_jobs) for more details. Doesn’t affect [**fit**](https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html#sklearn.neighbors.KNeighborsClassifier.fit) method.

**Attributes:**

**classes\_ : *array of shape (n\_classes,)***

Class labels known to the classifier

**effective\_metric\_ : *str or callble***

The distance metric used. It will be same as the metric parameter or a synonym of it, e.g. ‘euclidean’ if the metric parameter set to ‘minkowski’ and p parameter set to 2.

**effective\_metric\_params\_ : *dict***

Additional keyword arguments for the metric function. For most metrics will be same with metric\_params parameter, but may also contain the p parameter value if the effective\_metric\_ attribute is set to ‘minkowski’.

**n\_samples\_fit\_ : *int***

Number of samples in the fitted data.

**outputs\_2d\_ : *bool***

False when y’s shape is (n\_samples, ) or (n\_samples, 1) during fit otherwise True.