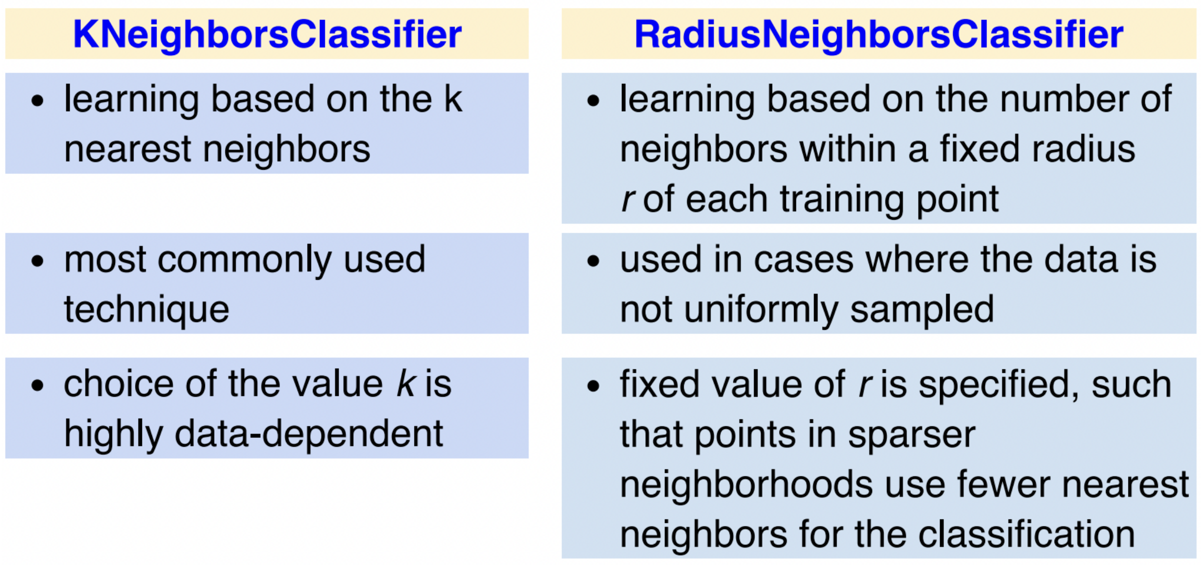
Week 7

K-nearest neighbours

- **instance-based** learning and **non-generalising** learning
  - does not attempt to construct a model
  - simply stores instances of the training data
- Classification is computed from a simple majority vote of the nearest neighbours of each point.
- Two different implementations:
  - KNeighborsClassifier
  - RadiusNeighborsClassifier

KNeighborsClassifier vs RadiusNeighborsClassifier



KNeighborsClassifier	RadiusNeighborsClassifier
• learning based on the k nearest neighbors	• learning based on the number of neighbors within a fixed radius *r* of each training point
• most commonly used technique	• used in cases where the data is not uniformly sampled
• choice of the value *k* is highly data-dependent	• fixed value of *r* is specified, such that points in sparser neighborhoods use fewer nearest neighbors for the classification

KNeighborsClassifier - Implementation

```
from sklearn.neighbors import KNeighborsClassifier
kneighbor_classifier = KNeighborsClassifier()

kneighbor_classifier.fit(X_train, y_train)
```

Hyperparameters

- **n_neighbors** (default = 5)
  - Specify the number of nearest neighbors K
  - value should be int
- **weights**
  - uniform (default)
  - distance
    - weigh points by the inverse of their distance
    - closer neighbors of a query point will have a greater influence than neighbors which are further away
  - own weight values
    - parameter also accepts a user-defined function which takes an array of distances as input, and returns an array of the same shape containing the weights.
- **algorithm**
  - ball_tree
  - kd_tree
  - brute
  - auto (default)

For 'ball_tree' and 'kd_tree' algorithms, there are some other parameters to be set.

- **leaf_size** (default = 30)
  - can affect the speed of the construction and query, as well as the memory required to store the tree

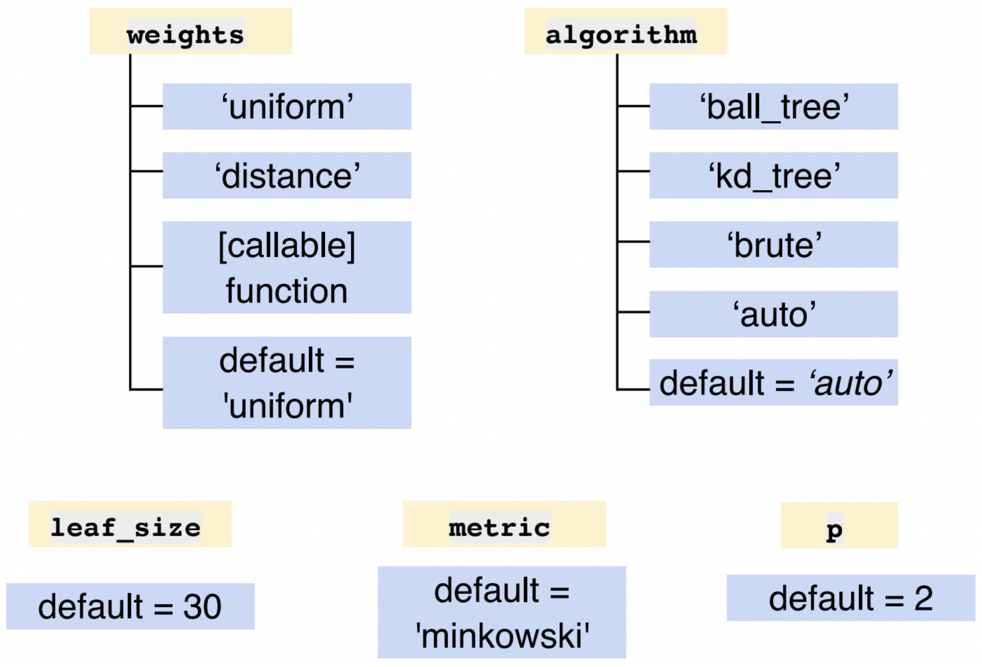
- **metric**
    - Distance metric to use for the tree
    - It is either string or callable function
        - “euclidean”, “manhattan”, “chebyshev”, “minkowski” (default), “wminkowski”,
          “seuclidean”, “mahalanobis”
- **p** (default = 2)
    - Power parameter for the Minkowski metric

RadiusNeighborsClassifier - Implementation

- The number of neighbors is specified within a fixed radius r of each training point using radius parameter.
- r is a float value

```
radius_classifier = RadiusNeighborsClassifier(radius = 1.0) #default
```

Hyperparameters


weights
‘uniform’
‘distance’
[callable] function
default = 'uniform'
algorithm
‘ball_tree’
‘kd_tree’
‘brute’
‘auto’
default = *‘auto’*
leaf_size
default = 30
metric
default = 'minkowski'
p
default = 2