Alternatively, one can use the KDTree or BallTree classes directly to find nearest neighbors. This is the functionality wrapped by the NearestNeighbors class used above. The Ball Tree and KD Tree have the same interface; we’ll show an example of using the KD Tree here:

>>>

>>> from sklearn. neighbors import KDTree

>>> import numpy as np

>>> X = np. array([[-1, -1], [-2, -1], [-3, -2], [1, 1], [2, 1], [3, 2]])

>>> kdt = KDTree (X, leaf\_size=30, metric='euclidean')

>>> kdt.query(X, k=2, return\_distance=False)

array ([[0, 1],

[1, 0],

[2, 1],

[3, 4],

[4, 3],

[5, 4]] ...)

Refer to the KDTree and BallTree class documentation for more information on the options available for nearest neighbors searches, including specification of query strategies, distance metrics, etc. For a list of available metrics, see the documentation of the DistanceMetric class and the metrics listed in sklearn.metrics.pairwise.PAIRWISE\_DISTANCE\_FUNCTIONS. Note that the “cosine” metric uses cosine\_distances.