**Nearest Neighbors Transformer**

Many scikit-learn estimators rely on nearest neighbors: Several classifiers and regressors such as **[KNeighborsClassifier](https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html" \l "sklearn.neighbors.KNeighborsClassifier" \o "sklearn.neighbors.KNeighborsClassifier)** and **[KNeighborsRegressor](https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsRegressor.html" \l "sklearn.neighbors.KNeighborsRegressor" \o "sklearn.neighbors.KNeighborsRegressor)**, but also some clustering methods such as [**DBSCAN**](https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html#sklearn.cluster.DBSCAN) and **[SpectralClustering](https://scikit-learn.org/stable/modules/generated/sklearn.cluster.SpectralClustering.html" \l "sklearn.cluster.SpectralClustering" \o "sklearn.cluster.SpectralClustering)**, and some manifold embeddings such as [**TSNE**](https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html#sklearn.manifold.TSNE) and **[Isomap](https://scikit-learn.org/stable/modules/generated/sklearn.manifold.Isomap.html" \l "sklearn.manifold.Isomap" \o "sklearn.manifold.Isomap)**.

All these estimators can compute internally the nearest neighbors, but most of them also accept precomputed nearest neighbors [sparse graph](https://scikit-learn.org/stable/glossary.html#term-sparse-graph), as given by **[kneighbors\_graph](https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.kneighbors_graph.html" \l "sklearn.neighbors.kneighbors_graph" \o "sklearn.neighbors.kneighbors_graph)** and **[radius\_neighbors\_graph](https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.radius_neighbors_graph.html" \l "sklearn.neighbors.radius_neighbors_graph" \o "sklearn.neighbors.radius_neighbors_graph)**. With mode mode='connectivity', these functions return a binary adjacency sparse graph as required, for instance, in **[SpectralClustering](https://scikit-learn.org/stable/modules/generated/sklearn.cluster.SpectralClustering.html" \l "sklearn.cluster.SpectralClustering" \o "sklearn.cluster.SpectralClustering)**. Whereas with mode='distance', they return a distance sparse graph as required, for instance, in [**DBSCAN**](https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html#sklearn.cluster.DBSCAN). To include these functions in a scikit-learn pipeline, one can also use the corresponding classes **[KNeighborsTransformer](https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsTransformer.html" \l "sklearn.neighbors.KNeighborsTransformer" \o "sklearn.neighbors.KNeighborsTransformer)** and **[RadiusNeighborsTransformer](https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.RadiusNeighborsTransformer.html" \l "sklearn.neighbors.RadiusNeighborsTransformer" \o "sklearn.neighbors.RadiusNeighborsTransformer)**. The benefits of this sparse graph API are multiple.

First, the precomputed graph can be re-used multiple times, for instance while varying a parameter of the estimator. This can be done manually by the user, or using the caching properties of the scikit-learn pipeline:

>>>

**>>> import** **tempfile**

**>>> from** **sklearn.manifold** **import** Isomap

**>>> from** **sklearn.neighbors** **import** KNeighborsTransformer

**>>> from** **sklearn.pipeline** **import** make\_pipeline

**>>> from** **sklearn.datasets** **import** make\_regression

**>>>** cache\_path = tempfile.gettempdir() *# we use a temporary folder here*

**>>>** X, \_ = make\_regression(n\_samples=50, n\_features=25, random\_state=0)

**>>>** estimator = make\_pipeline(

**...**  KNeighborsTransformer(mode='distance'),

**...**  Isomap(n\_components=3, metric='precomputed'),

**...**  memory=cache\_path)

**>>>** X\_embedded = estimator.fit\_transform(X)

**>>>** X\_embedded.shape

(50, 3)