

Cluster class

January 1, 2020

1 Cluster Class

This class wraps kmeans clustering class

```
[3]: from sklearn.cluster import KMeans
from sklearn import datasets
from sklearn.datasets import make_blobs
from sklearn.model_selection import train_test_split
import numpy as np

class MyCluster:
    def __init__(self, centroids=8):
        self.c = KMeans(n_clusters=centroids)

    def fit(self, X):
        return self.c.fit(X)

    def predict(self, X_test):
        return self.c.predict(X_test)

    def score(self, X_test):
        return self.c.score(X_test)
```

1.0.1 Parameters

centroids: int, optional, default: 8.

- The number of clusters to form as well as the number of centroids to generate.

1.1 fit(self, X_train)

Compute k-means clustering.

1.1.1 Parameters

X_train: array-like or sparse matrix, shape=(n_samples, n_features)

- Training instances to cluster. It must be noted that the data will be converted to C ordering, which will cause a memory copy if the given data is not C-contiguous.

returns: Fitted estimator.

1.2 `predict(self, X_test)`

Predict the closest cluster each sample in X belongs to.

1.2.1 Parameters

X_test: {array-like, sparse matrix} of shape (n_samples, n_features)

- New data to predict.

returns: labelsarray, shape [n_samples]; Index of the cluster each sample belongs to.

1.3 `score(self, X_test)`

Opposite of the value of X on the K-means objective.

1.3.1 Parameters

X_test: {array-like, sparse matrix} of shape (n_samples, n_features)

- New data.

returns: float Opposite of the value of X on the K-means objective.

1.4 Example

```
[5]: X = np.array([[1, 2], [1, 4], [1, 0],  
                  [10, 2], [10, 4], [10, 0]])  
kmeans = MyCluster(2).fit(X)  
kmeans.labels_
```

```
[5]: array([1, 1, 1, 0, 0, 0])
```

```
[6]: kmeans.predict([[0, 0], [12, 3]])
```

```
[6]: array([1, 0])
```