6.5. Unsupervised dimensionality reduction

If your number of features is high, it may be useful to reduce it with an unsupervised step prior to supervised steps. Many of the Unsupervised learning methods implement a `transform` method that can be used to reduce the dimensionality. Below we discuss two specific example of this pattern that are heavily used.

Pipelining

The unsupervised data reduction and the supervised estimator can be chained in one step. See Pipeline: chaining estimators.

6.5.1. PCA: principal component analysis

`decomposition.PCA` looks for a combination of features that capture well the variance of the original features. See Decomposing signals in components (matrix factorization problems).

Examples

- Faces recognition example using eigenfaces and SVMs

6.5.2. Random projections

The module: `random_projection` provides several tools for data reduction by random projections. See the relevant section of the documentation: Random Projection.

Examples

- The Johnson-Lindenstrauss bound for embedding with random projections

6.5.3. Feature agglomeration

`cluster.FeatureAgglomeration` applies Hierarchical clustering to group together features that behave similarly.

Examples

- Feature agglomeration vs. univariate selection
- Feature agglomeration

Feature scaling

Note that if features have very different scaling or statistical properties, **`cluster.FeatureAgglomeration`** may not be able to capture the links between related features. Using a **`preprocessing.StandardScaler`** can be useful in these settings.

Toggle Menu