Pipeline的steps参数设定数据处理流程。格式为('key', value)。key为该step设定的名称，value是对应的处理类，最后通过list将其传入。前n-1个step中的类必须有transform函数，最后一个可有可无，且Pipeline继承了最后一个类的所有方法。

|  |
| --- |
| 1. **from** sklearn.pipeline **import** Pipeline 2. **from** sklearn.pipeline **import** make\_pipeline 3. **from** sklearn.linear\_model **import** LinearRegression 4. **from** sklearn.decomposition **import** PCA 5. **from** sklearn.datasets **import** load\_iris 7. iris = load\_iris() 8. pipe = Pipeline([('pca',PCA()), ('linear',LinearRegression())]) 9. pipe.fit(iris.data, iris.target) |

make\_pipeline是Pipeline的简单实现只需传入每个step的类实例即可。

FeatureUnion对特征进行并行化处理，每一个step分开计算，最后将计算得到的结果合并到一块并返回一个数组。

|  |
| --- |
| 1. **from** sklearn.pipeline **import** FeatureUnion 2. **from** sklearn.preprocessing **import** StandardScaler 3. **from** sklearn.preprocessing **import** FunctionTransformer 4. **from** numpy **import** log1p 6. iris = load\_iris() 7. steps = FeatureUnion(transformer\_list=[('standard', StandardScaler()), ('ToLog', FunctionTransformer(log1p))]) 8. data = steps.fit\_transform(iris.data) |