一、预处理的一些通用方法

1、get_params([deep])：返回模型的参数。

- deep：如果为True，则可以返回模型参数的子对象。

2、set_params(**params)：设置模型的参数。

- params：待设置的关键字参数。

3、fit(X[, y])：获取预处理需要的参数（如：特征的最大值、最小值等），不同的预处理方法需要的参数不同。

- X：训练集样本集合。通常是一个NumPy数组，每行代表一个样本，每列代表一个特征。
- y：训练样本的标签集合。它与X的每一行相对应。

4、transform(X[, copy])：执行预处理，返回处理后的样本集。

- X：训练集样本集合。通常是一个NumPy数组，每行代表一个样本，每列代表一个特征。
- copy：一个布尔值，指定是否拷贝数据。

5、fit_transform(X[, y])：获取预处理需要的参数并执行预处理，返回处理后的样本集。

- X：训练集样本集合。通常是一个NumPy数组，每行代表一个样本，每列代表一个特征。
- y：训练样本的标签集合。它与X的每一行相对应。

二、预处理的一些通用参数

1、copy：一个布尔值，指定是否拷贝数据。

- 如果为False则执行原地修改。此时节省空间，但修改了原始数据。

三、特征处理

3.1 二元化

1、原型为sklearn.preprocessing.Binarizer(threshold=0.0, copy=True)

- threshold：浮点数，指定转换阈值，低于此阈值的值为0，高于此阈值的值为1。
- copy：一个布尔值，指定是否拷贝数据。

2、方法有fit、transform、fit_transform

- fit(X[, y])：不作任何事情，主要用于为流水线Pipeline提供接口。
- transform(X[, copy])：将每个样本的特征二元化。
- fit_transform(X[, y])：将每个样本的特征二元化。

3.2 独热编码

1、原型为sklearn.preprocessing.OneHotEncoder(n_values='auto', categorical_features='all', dtype=<class 'float'>, sparse=True, handle_unknown='error')

- n_vlaues：字符串‘auto’（自动从训练数据中推断特征值取值上界）或者一个整数（指定所有特征取值的上界）或者一个整数数组（每个元素依次指定每个特征取值的上界），指定样本每个特征取值的上界。
- categorical_features：字符串‘all’（所有特征都独热编码）或者下标数组（指定下标的特征独热编码）或者一个mask（对应为True的特征独热编码），指定哪些特征需要独热编码。所有非categorical_features的特征都将被安排在categorical_features特征的右边。
- dtype：一个类型，指定独热编码的数值类型，默认为np.float。
- sparse：一个布尔值，指定编码结果是否作为稀疏矩阵。
- handle_unknown：一个字符串，指定转换过程中遇到未知的categorical_features时的异常处理策略。‘error’抛出异常，‘ignore’忽略。

2、属性有active_features_、feature_indices_、n_values_

- active_features_：一个索引数组，存放转换后的特征中哪些是由独热编码而来。仅当n_values='auto'时该属性有效。
- feature_indices_：一个索引数组，存放原始特征和转换后特征位置的映射关系。第i个原始特征将被映射到转换后的[feature_indices_[i], feature_indices_[i+1]]之间的特征。
- n_values_：一个计数数组，存放每个原始特征取值的种类。一般为训练数据中该特征取值最大值加1（默认每个特征取值是从0开始的）。

3、方法有fit、transform、fit_transform

- fit(X[, y])：训练编码器。
- transform(X[, copy])：执行独热编码。
- fit_transform(X[, y])：训练编码器并执行独热编码。

3.3 标准化

3.3.1 MinMaxScaler

1、原型为sklearn.preprocessing.MinMaxScaler(feature_range=(0, 1), copy=True)

- feature_range：一个元组(min, max)，指定执行变换后特征的取值范围。
- copy：一个布尔值，指定是否拷贝数据。

2、属性有min_、scale_、data_min_、data_max_、data_range_

- min_：一个数组，给出每个特征的原始最小值的调整值。设特征j的原始最小值j(min)，原始最大值j(max)，那么特征j的原始最小值的调整值为j(min) / (j(max) - j(min))。
- scale_：一个数组，给出每个特征的缩放倍数。
- data_min_：一个数组，给出每个特征的原始最小值。
- data_max_：一个数组，给出每个特征的原始最大值。
- data_range_：一个数组，给出每个特征的原始范围。（范围=最大值-最小值）。

3、方法有fit、transform、fit_transform、inverse_transform、partial_fit

- fit(X[, y])：计算每个特征的最小值和最大值，为后续转换做准备。

- transform(X[, copy])：执行特征标准化。
- fit_transform(X[, y])：计算每个特征的最小值和最大值并执行特征标准化。
- inverse_transform(X)：逆标准化，还原成原始数据。
- partial_fit(X[, y])：学习部分数据，计算每个特征的最小值和最大值，为后续转换做准备。它支持批量学习，对内存更友好。

3.3.2 MaxAbsScaler

1、原型为sklearn.preprocessing.MaxAbsScaler(copy=True)

- copy：一个布尔值，指定是否拷贝数据。

2、属性有scale_、max_abs_、n_sample_seen_

- scale_：一个数组，给出每个特征的缩放倍数的倒数。
- max_abs_：一个数组，给出每个特征的绝对值的最大值。
- n_sample_seen_：一个整数，给出当前已处理的样本数（用于分批训练）。

3、方法参考MinMaxScaler

3.3.3 StandardScaler

1、原型为sklearn.preprocessing.StandardScaler(copy=True, with_mean=True, with_std=True)

- copy：一个布尔值，指定是否拷贝数据。
- with_mean：一个布尔值，指定是否去中心化。为True时，缩放前先将每个特征中心化（即特征值减去该特征的均值）。若元素数据是稀疏矩阵形式，则不能指定with_mean=True。
- with_std：一个布尔值，指定是否方差归一化。为True时，缩放每个特征到单位方差。

2、属性有scale_、mean_、var_、n_sample_seen

- scale_：一个数组，给出每个特征的缩放倍数的倒数。
- mean_：一个数组，给出原始数据每个特征的均值。
- var_：一个数组，给出原始数据每个特征的方差。
- n_sample_seen：一个整数，给出当前已处理的样本的数据（用于分批训练）。

3、方法参考MinMaxScaler

3.4 正则化

1、原型为sklearn.preprocessing.Normalizer(norm='l2', copy=True)

- norm：一个字符串，指定正则化方法。‘l1’采用L1范数正则化，‘l2’采用L2范数正则化，‘max’采用L∞范数正则化。
- copy：一个布尔值，指定是否拷贝数据。

2、方法有fit、transform、fit_transform

- fit(X[, y])：不作任何事情，主要用于为流水线Pipeline提供接口。
- transform(X[, copy])：将每个样本正则化为范数等于单位1。

- fit_transform(X[, y])：将每个样本正则化为范数等于单位1。