计算机与信息工程学院实验报告

••••••••••••••••••••••••••••••••• 密 ••••••••••••••••••••••••••••••••• 封 ••••••••••••••••••••••••••••••••• 线 •••••••••••••••••••••••••••••••••

姓名：\_\_杨佳森\_学号：\_2112080106专业：\_\_数据科学与大数据\_年级：\_ 2021 \_

课程： 数据分析与可视化 主讲教师：\_ 周黎明

实验时间： 2023 年\_5\_月\_3\_\_日

实验题目： 应用数据分析

实验目的： 1. 掌握 scikit-learn 构建聚类模型、分类模型、回归模型 2. 掌握企业数据的分析规则

实验环境（硬件和软件） PC、 Windows 操作系统、Jupyter Notebook

实验内容：

（一）scikit-learn 构建模型

1、使用 scikit-learn 转换器处理数据

2、构建聚类模型

3、构建分类模型

4、构建回归模型

（二）企业数据分析

1、航空公司客服价值分析

2、财政收入预测分析

实验步骤：

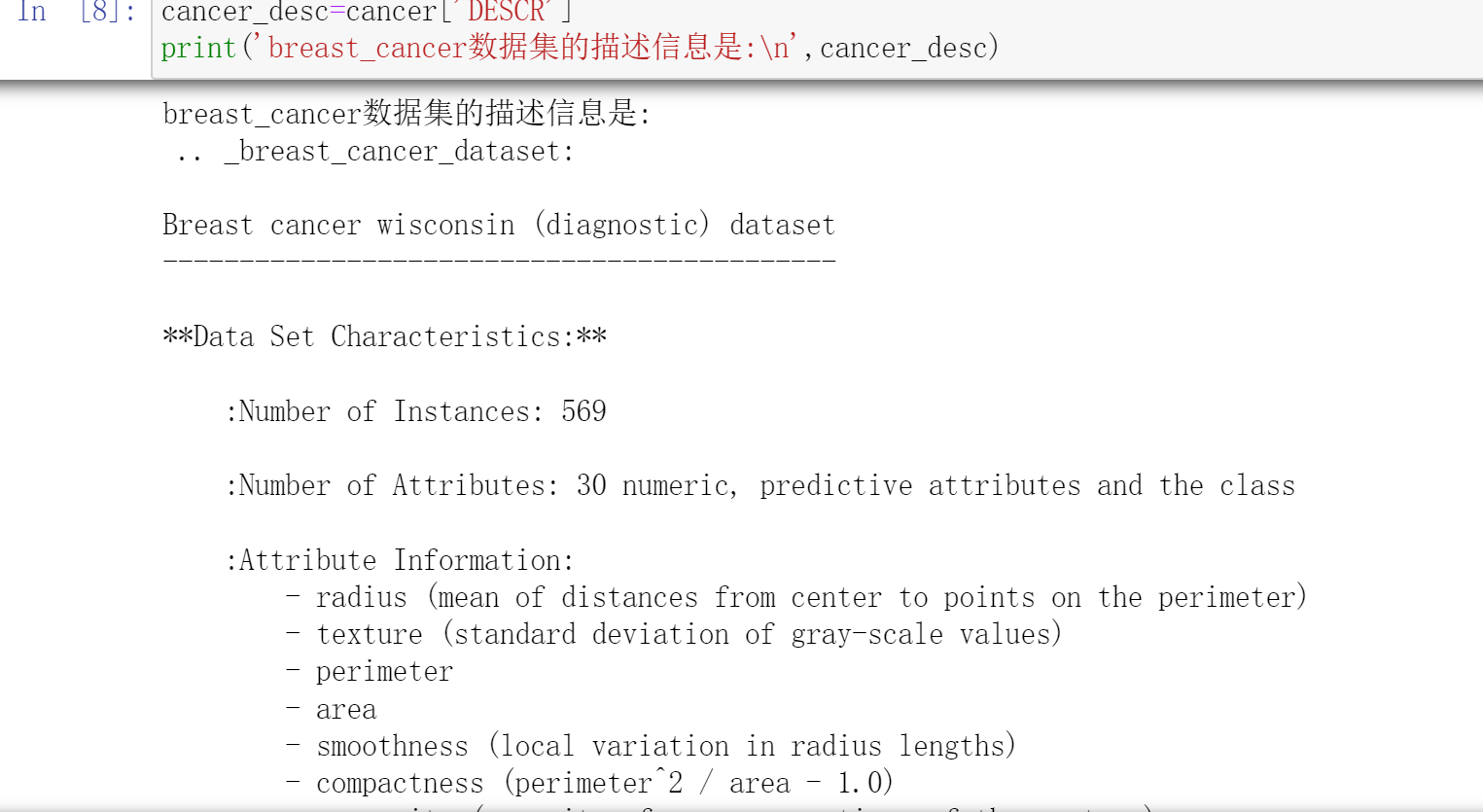
**（一）scikit-learn 构建模型**

**1、使用 scikit-learn 转换器处理数据**

Scikit-learn(sklearn)是机器学习中常用的第三方模块，对常用的机器学习方法进行了封装，包括回归(Regression)、降维(Dimensionality Reduction)、分类(Classfication)、聚类(Clustering)等方法。

sklearn 库的 datasets 模块集成了部分数据分析的经典数据集，可以使用这些数据集进行数据预处理、建模等操作，熟悉 sklearn 的数据处理流程和建模流程。datasets 模块常用数据集的加载函数与解释如表6-1所示。使用sklearn进行数据预处理会用到sklearn提供的统一接口——转换器（Transformer）。

加载datasets模块中的数据集：



将数据集划分为训练集和测试集：

sklearn的model\_selection提供了train\_test\_split函数，能够对数据集进行拆分。

train\_test\_split函数根据传入的数据，分别将传入的数据划分为训练集和测试集。如果传入的是1组数据，那么生成的就是这一组数据随机划分后训练集和测试集，总共2组。如果传入的是2组数据，则生成的训练集和测试集分别2组，总共4组。train\_test\_split是最常用的数据划分方法，在model\_selection模块中还提供了其他数据集划分的函数，如PredefinedSplit，ShuffleSplit等。

sklearn.model\_selection.train\_test\_split(train\_data, train\_target, test\_size, random\_state, shuffle)

train\_data 还未划分的数据集

train\_target 还未划分的标签

test\_size 分割比例，默认为0.25，即测试集占完整数据集的比例

random\_state 随机数种子，应用于分割前对数据的洗牌。可以是int，RandomState实例或None，默认值=None。设成定值意味着，对于同一个数据集，只有第一次运行是随机的，随后多次分割只要rondom\_state相同，则划分结果也相同。

shuffle 是否在分割前对完整数据进行洗牌（打乱），默认为True，打乱



使用sklearn转换器进行数据预处理与降维：

在数据分析过程中，各类特征处理相关的操作都需要对训练集和测试集分开操作，需要将训练集的操作规则，权重系数等应用到测试集中。如果使用pandas，则应用至测试集的过程相对烦琐，使用sklearn转换器可以解决这一困扰。

sklearn把大量特征处理的相关操作封装为转换器，放在了preprocessing模块，转换器主要包括3个方法：fit、transform、fit\_transform

fit 主要通过分析特征和目标值提取有价值的信息，这些信息可以是统计量，也可以是权值系数等

transform 该方法主要用来对特征进行转换，从可利用信息的角度分为无信息转换和有信息转换。无信息转换是指不利用任何其他信息进行转换，比如指数和对数函数转换等。有信息转换根据是否利用目标值向量又可分为无监督转换和有监督转换。无监督转换指只利用特征的统计信息的转换，比如标准化和PCA降维。有监督转换是指既利用了特征信息又利用了目标值信息的转换，比如通过模型选择特征和LDA降维等。

fit\_transform 该方法是先调用fit方法、然后调用transform方法。

离差标准化

sklearn除了提供离差标准化函数MinMaxScaler外，还提供了一系列数据预处理函数，具体如下：

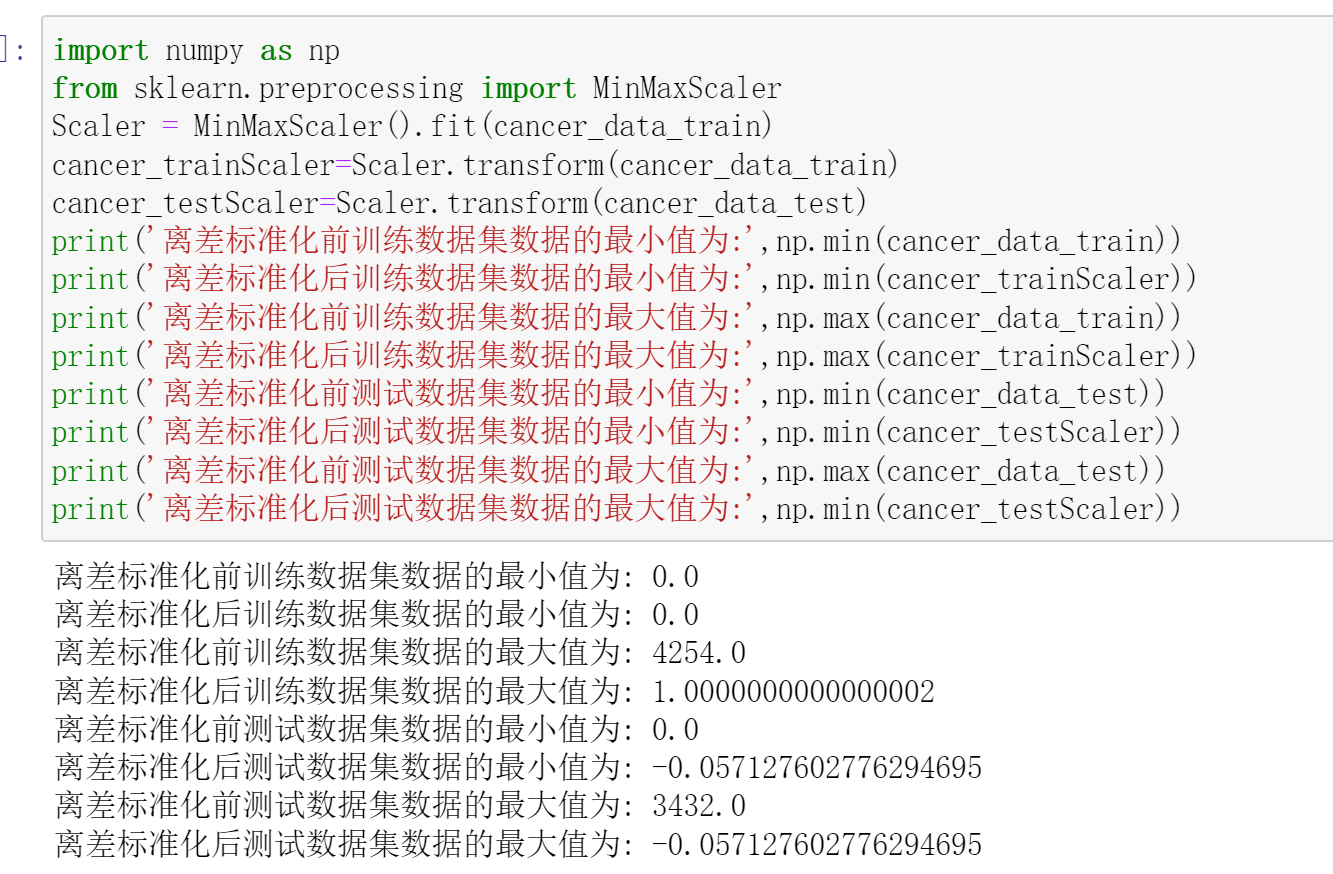
StandardScaler 对特征进行标准差标准化

Normalizer 对特征进行归一化

Binarizer 对定量特征进行二值化处理

OneHotEncoder 对定性特征进行独热编码处理

Function Transformer 对特征进行自定义函数变化



sklearn除了提供基本的特征变换函数外，还提供了降维算法，特征选择算法，这些算法的使用也是通过转换器的方式。

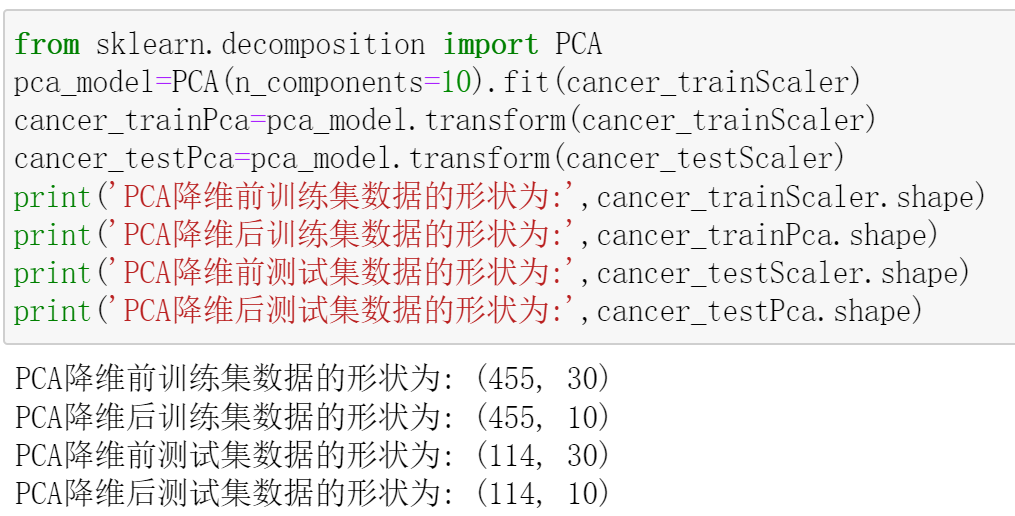
PCA降维算法函数常用参数及其作用:

n\_components: 接收None、int、float或mle。未指定时，代表所有的特征都会留下来；如果为int，则表示将原始数据降低到n个维度；如果为float，则PCA根据样本特征方差来决定降维后的维度数；赋值为mle，PCA会用MLE算法根据特征的方差分布情况自动选择一定数量的主成分特征来降维。默认为None。

Copy: 接收boolean。代表是否在运行算法时将原始数据复制一份，如果为True，则运行后，原始数据的值不会有任何变化，如果为False，则运行PCA算法后，原始数据的值会发生改变，默认为True。

Whiten: 接收boolean，表示白化，所谓白化，就是对降维后的数据的每个特征进行归一化，让方差都为1.默认为False。

svd\_solver: 接收auto、full、arpack、randomized。代表使用的SVD算法。randomized一般适用于数据量大，数据维度多，同时主成分数目比例又比较低的PCA降维，它使用了一些加快SVD的随机算法。full是使用SciPy库实现的传统SVD算法。arpack和randomized的适用场景类似，区别是，randomized使用的是sklearn自己的SVD实现，而arpack直接使用了SciPy库的sparse SVD实现。auto则代表PCA类会自动在上述3种算法中去权衡，选择一个合适的SVD算法来降维。默认为auto。



PCA算法是一种在尽可能减少信息损失的前提下，找到某种方式降低数据的维度的方法。PCA通常用于高维数据集的探索与可视化，还可以用于数据压缩，数据预处理。

通常来说，我们期望得到的结果，是把原始数据的特征空间（n个d维样本）投影到一个小一点的子空间里去，并尽可能表达的很好（就是损失信息最少）。常见的应用在于模式识别中，我们可以通过减少特征空间的维度，抽取子空间的数据来最好的表达我们的数据，从而减少参数估计的误差。

**2、构建并评价聚类模型**

使用sklearn估计器构建聚类模型

首先在处理数据的过程中我们先对我们的输入内容进行考虑，即聚类输入为一组未被进行标记的数据样本，聚类通过不同数据之间的距离和相似度将这些数据样本进行划分，被划分成若干组的数据根据样本内部距离最小化，外部距离最大化的原则进行划分，可理解成不同维度与同一维度下数据的距离。



Sklearn常用的聚类算法模块cluster提供的聚类算法及其适用范围如下:



k-Means聚类模型

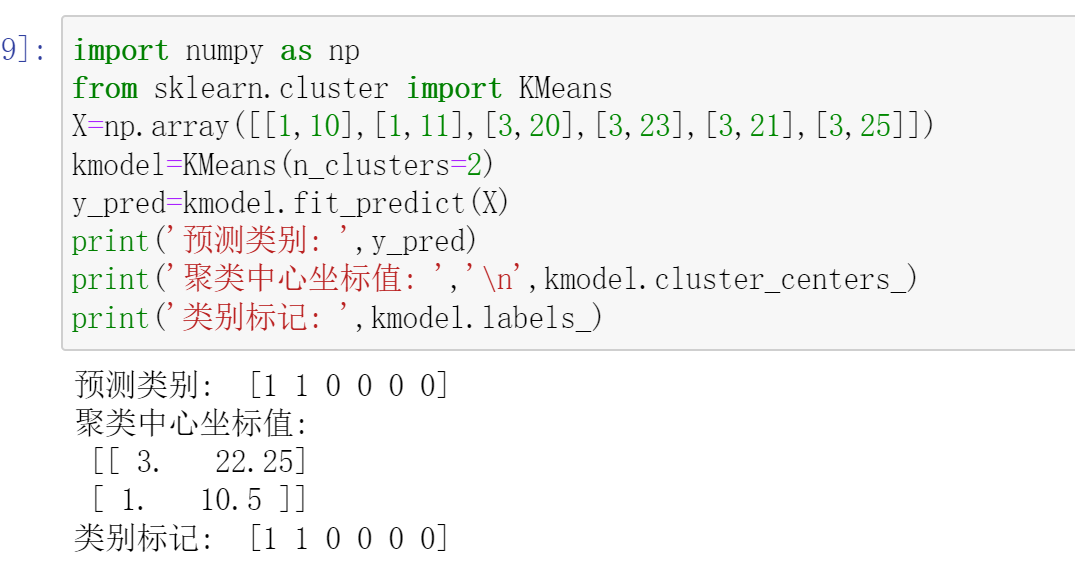
KMeans(n\_clusters=8,init=’k-means++’,n\_init=10,max\_iter=300,tol=1e-4,precompute\_distances=’auto’,verbose=0,random\_state=None,copy\_x=True,n\_jobs=None,algorithm=’auto’)

k-Means聚类模型有三个重要的参数：

n\_clusters: int, 默认: 8，表示K-Means算法中的k值，即聚类的簇数。

init: {‘k-means++’, ‘random’} or ndarray of shape (n\_clusters, n\_features), 默认: ‘k-means++’，表示初始化簇中心的方法。如果传入的值为‘k-means++’，则表示用一种启发式的方法来初始化簇中心，使它们更好地分散在整个数据集中。如果传入的值为‘random’，则表示以随机的方式选取k个数据点来作为簇中心。如果传入的值为ndarray，则表示使用传入的ndarray作为簇中心的初始值。

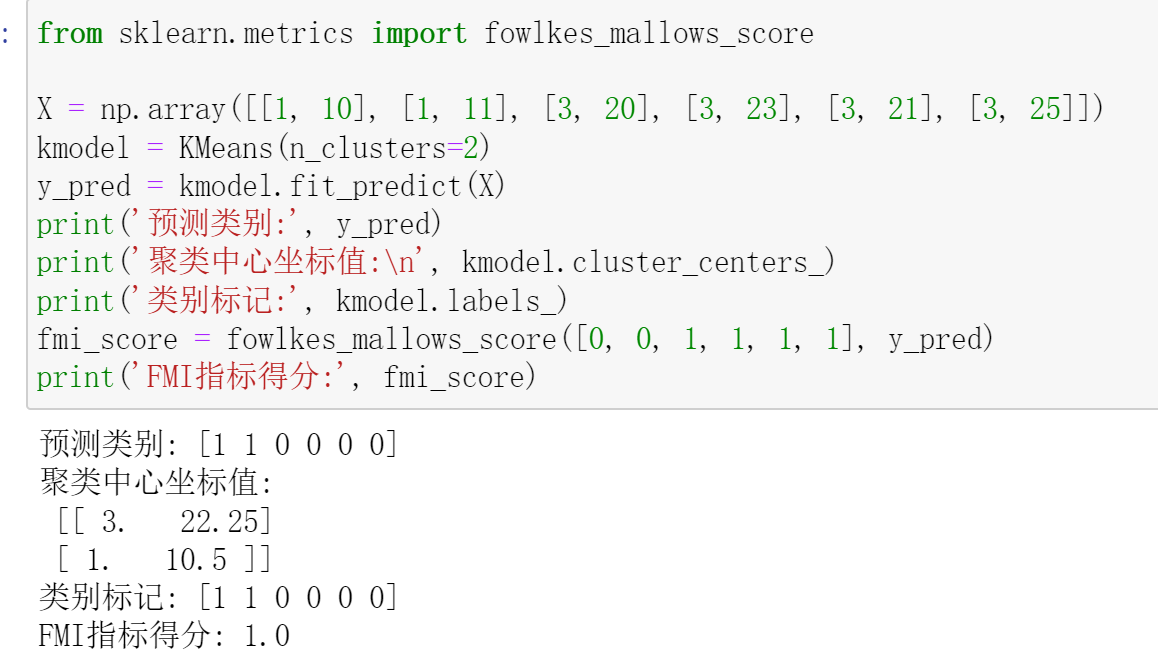
max\_iter: int, 默认: 300，表示最大迭代次数。在达到最大迭代次数或收敛前，模型会尝试调整簇中心的位置，直到聚类结果最优。



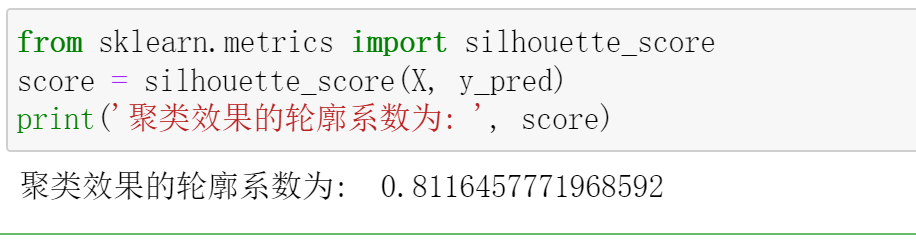
评价聚类模型

聚类评价的标准是组内的对象相互之间是相似的（相关的），而不同组中的对象是不同的（不相关的）。即组内的相似性越大，组间差别越大，聚类效果就越好。sklearn的metrics模块提供的聚类模型评价指标。

使用FMI评价法评价K-Means聚类模型



使用轮廓系数评价法评价



使用Calinski-Harabasz指数评价K-Means聚类模型



**3、构建并评价分类模型**

使用sklearn估计器构建分类模型

LinearSVR实现了线性回归支持向量机，他是根据liblinear实现的，其函数原型为：

sklearn.svm.LinearSVC(epsilon=0.0,loss='epsilon\_insensitive',dual='True',tol=0.0001,C=1.0,fit\_intercept=True,intercept\_scaling=1.0,verbose=0,random\_state=None,max\_iter=1000)

参数说明:

C：一个浮点数，为惩罚项参数。

loss：一个字符串，为损失函数。当值为epsilon\_insensitive时损失函数为L（它是标准SVR的损失函数）；值为square\_epsilon\_insensitive时表示为L的平方。

epsilon：浮点数，用于loss中的sigma参数。

dual：布尔值。如果为True，则解决对偶问题，如果为False，则解决原始问题，当n\_samples>n\_features时，倾向于采用False。

tol：浮点数，指定终止迭代的阈值。

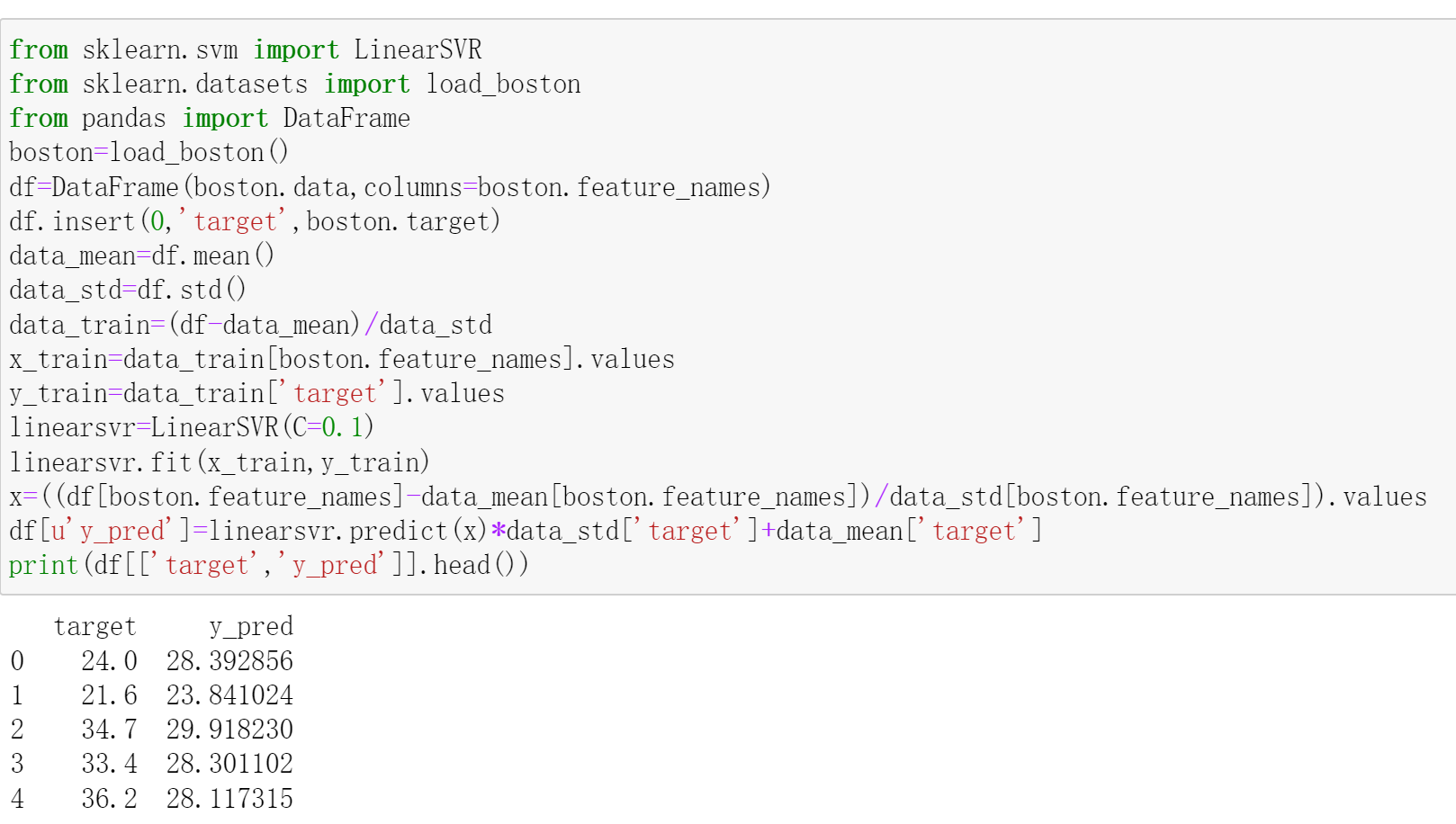
fit\_intercept：布尔值，如果为True，则计算截距，即决策函数中的常数项；否则忽略截距。

intercept\_scaling:浮点值，如果提供了，则实例x变成了向量[x,intercept\_scaling],此时相当于添加了一个人工特征，该特征对所有实例都是常数值。这个时候截距变成了intercept\_scaling\*人工特征的权重Ws；人工特征也参与了惩罚项的计算。

verbose：一个整数，表示是否开启verbose输出

randomstate：一个整数或者一个RandomState实例，或者为None；如果为整数，则他指定随机数生成的种子，如果为RandomState，则指定随机数生成器。如果为None，则指定随机数生成器。

max\_iter：一个整数，指定最大的迭代次数。



**4、构建并评价回归模型**

使用sklearn估计器构建回归模型



最小二乘法回归

sklearn.linear\_model.LinearRegression(fit\_intercept=True, normalize=False,copy\_X=True, n\_jobs=1)

参数说明：

fit\_intercept：boolean,optional,default True。是否计算截距，默认为计算。如果使用中心化的数据，可以考虑设置为False,

不考虑截距。注意这里是考虑，一般还是要考虑截距。

normalize：boolean,optional,default False。标准化开关，默认关闭；该参数在fit\_intercept设置为False时自动忽略。如果为True,回归会标准化输入参数：(X-X均值)/||X||，当然啦，在这里还是建议将标准化的工作放在训练模型之前；若为False，在训练模型前，可使用sklearn.preprocessing.StandardScaler进行标准化处理。

copy\_X：boolean,optional,default True。默认为True, 否则X会被改写。

n\_jobs：整型，代表CPU工作效率的核数，默认为1，-1与CPU核数一致。

主要属性：

coef\_：array,shape(n\_features, ) or (n\_targets, n\_features)。回归系数(斜率)。

intercept\_: 截距

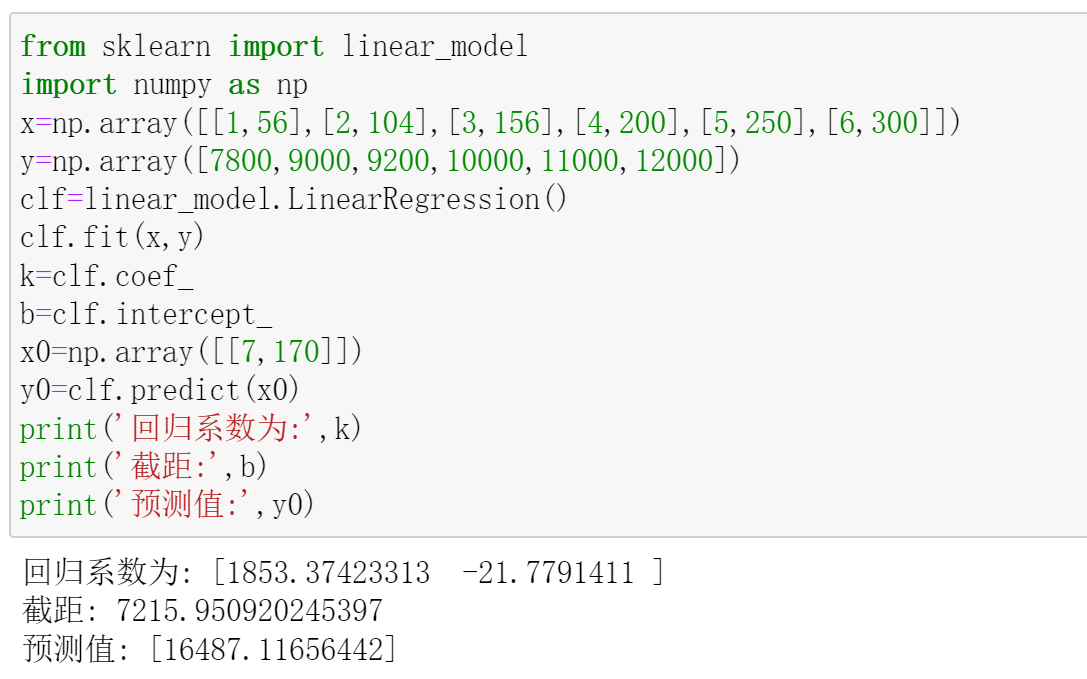
主要方法：

fit(X,y,sample\_weight=None) 拟合线性模型

predict(x):预测方法，将返回值y\_pred

get\_params(deep=True)： 返回对regressor 的设置值

score(X,y,sample\_weight=None)：评分函数，将返回一个小于1的得分，可能会小于0。



岭回归

linear\_model.Ridge(alpha=1.0, fit\_intercept=True, normalize=False,

copy\_X=True,max\_iter=None,tol=0.001,solver=’auto’,random\_state=None)

参数说明:

alpha：正则化系数，float类型，默认为1.0。正则化改善了问题的条件并减少了估计的方差。较大的值指定较强的正则化。

fit\_intercept：是否需要截距，bool类型，默认为True。也就是是否求解b。

normalize：是否先进行归一化，bool类型，默认为False。如果为真，则回归X将在回归之前被归一化。 当fit\_intercept设置为False时，将忽略此参数。 当回归量归一化时，注意到这使得超参数学习更加鲁棒，并且几乎不依赖于样本的数量。 相同的属性对标准化数据无效。然而，如果你想标准化，请在调用normalize = False训练估计器之前，使用preprocessing.StandardScaler处理数据。

copy\_X：是否复制X数组，bool类型，默认为True，如果为True，将复制X数组; 否则，它覆盖原数组X。

max\_iter：最大的迭代次数，int类型，默认为None，最大的迭代次数，对于sparse\_cg和lsqr而言，默认次数取决于scipy.sparse.linalg，对于sag而言，则默认为1000次。

tol：精度，float类型，默认为0.001。就是解的精度。

solver：求解方法，str类型，默认为auto。可选参数为：auto、svd、cholesky、lsqr、sparse\_cg、sag。

auto根据数据类型自动选择求解器。

svd使用X的奇异值分解来计算Ridge系数。对于奇异矩阵比cholesky更稳定。

cholesky使用标准的scipy.linalg.solve函数来获得闭合形式的解。

sparse\_cg使用在scipy.sparse.linalg.cg中找到的共轭梯度求解器。作为迭代算法，这个求解器比大规模数据（设置tol和max\_iter的可能性）的cholesky更合适。

lsqr使用专用的正则化最小二乘常数scipy.sparse.linalg.lsqr。它是最快的，但可能在旧的scipy版本不可用。它是使用迭代过程。

sag使用随机平均梯度下降。它也使用迭代过程，并且当n\_samples和n\_feature都很大时，通常比其他求解器更快。

random\_state：sag的伪随机种子。



**（二）企业数据分析**

**1、航空公司客服价值分析**

主要包括4个步骤：

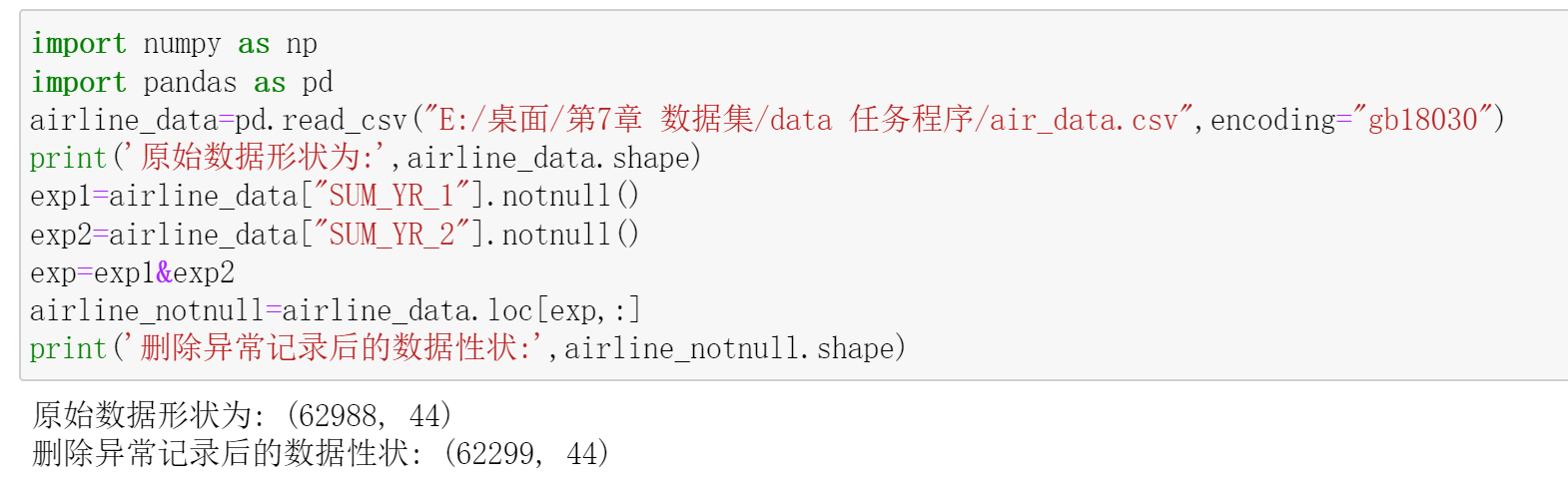
（1）抽取航空公司2012年4月1日至2014年3月31日的数据。

（2）对抽取的数据进行数据清洗、特征构建和标准化等操作。

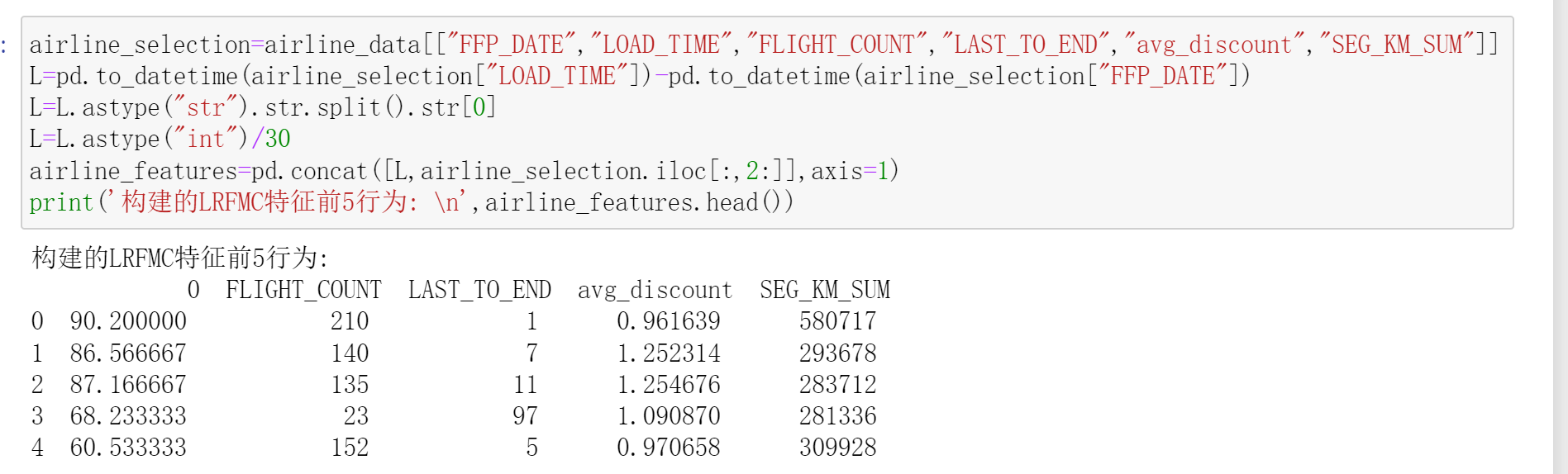
（3）基于RFM模型，使用K-Means算法进行客户分群。

（4）针对模型结果得到不同价值的客户，采用不同的营销手段，提供定制化的服务。

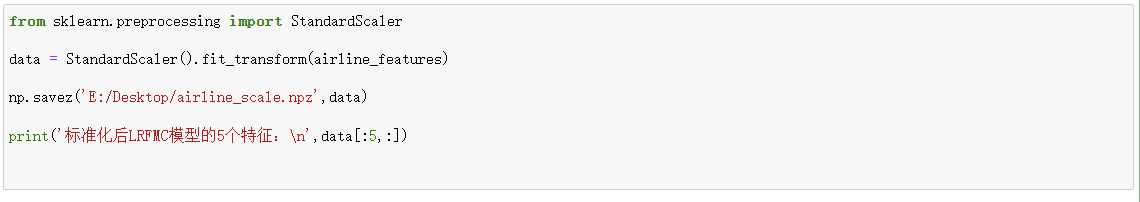
缺失值与异常值处理：



选取并构建LRFMC模型的特征：



标准化LRFMC模型的特征：



**2、财政收入预测分析**

