

计算机学院

课程设计报告

（ 20 21 ~20 22 学年度 第 一学期 ）

|  |  |
| --- | --- |
| 课程名称 | 机器学习 |
| **设计名称** | **基于XGBoost的服装分类** |

|  |  |  |  |
| --- | --- | --- | --- |
| 姓名 | 刘良柏 | 学号 | 20200011704 |

|  |  |  |  |
| --- | --- | --- | --- |
| 专业 | 人工智能 | 班级 | 软智2001班 |

|  |  |  |  |
| --- | --- | --- | --- |
| 地点 | 雨母楼204 | 教师 | 熊东平 |

目录

[一、课题设计背景 3](#_Toc529765059_WPSOffice_Level1)

[1.1赛题介绍](#_Toc1804859298_WPSOffice_Level1) [3](#_Toc1804859298_WPSOffice_Level1)

[1.1.1 背景及任务](#_Toc820320477_WPSOffice_Level2) [4](#_Toc820320477_WPSOffice_Level2)

[1.2比赛基本数据集](#_Toc281243199_WPSOffice_Level1) [4](#_Toc281243199_WPSOffice_Level1)

[1.2.1 训练数据集---fashion-mnist\_train.csv](#_Toc1063707611_WPSOffice_Level2) [4](#_Toc1063707611_WPSOffice_Level2)

[1.2.2 测试数据集---fashion-mnist\_test\_data.csv](#_Toc2079940449_WPSOffice_Level2) [4](#_Toc2079940449_WPSOffice_Level2)

[1.3打分指标](#_Toc242938546_WPSOffice_Level1) [5](#_Toc242938546_WPSOffice_Level1)

[1.4分类的意义与目的 5](#_Toc1417930374_WPSOffice_Level1)

[二、实现原理概述 5](#_Toc529765059_WPSOffice_Level1)

[2.1数据处理 5](#_Toc294146151_WPSOffice_Level2)

[1. 载入各种数据科学以及可视化库: 5](#_Toc200501501_WPSOffice_Level3)

[2. 载入数据： 5](#_Toc426885164_WPSOffice_Level3)

[3. 数据总览: 5](#_Toc2063570368_WPSOffice_Level3)

[4. 判断数据缺失和异常 5](#_Toc566275926_WPSOffice_Level3)

[2.2 XGBoost原理介绍 6](#_Toc207004463_WPSOffice_Level2)

[1. 特点: 6](#_Toc200501501_WPSOffice_Level3)

[2. 回归树介绍： 6](#_Toc426885164_WPSOffice_Level3)

[3. XGBoost原理: 7](#_Toc2063570368_WPSOffice_Level3)

[2.3实验预期目标 1](#_Toc1685266320_WPSOffice_Level2)1

[三 、具体代码实现 1](#_Toc523125651_WPSOffice_Level1)1

[3.1 数据处理 1](#_Toc374765539_WPSOffice_Level2)2

[3.2 算法参数设置 1](#_Toc1811811102_WPSOffice_Level2)3

[1.各参数含义 1](#_Toc1938560501_WPSOffice_Level3)3

[3.3 XGBoost模型训练 14](#_Toc1811811102_WPSOffice_Level2)

[3.3.1 代码实现 14](#_Toc2045843316_WPSOffice_Level3)

[3.3.2 特征重要程度展示 14](#_Toc2045843316_WPSOffice_Level3)

[3.3.3 在训练集上准确率展示 15](#_Toc1127939895_WPSOffice_Level3)

[3.4 模型测试](#_Toc1037767629_WPSOffice_Level2) [14](#_Toc1037767629_WPSOffice_Level2)

[3.4.1 代码实现](#_Toc2045843316_WPSOffice_Level3) [14](#_Toc2045843316_WPSOffice_Level3)

[3.3.2 在测试集上结果展示](#_Toc1127939895_WPSOffice_Level3) [15](#_Toc1127939895_WPSOffice_Level3)

[四、使用GridSearch对模型调参及优化](#_Toc1381844603_WPSOffice_Level1) [19](#_Toc1381844603_WPSOffice_Level1)

[4.1 模型参数介绍](#_Toc1774083963_WPSOffice_Level2) [19](#_Toc1774083963_WPSOffice_Level2)

[4.2 模型调参](#_Toc1366211193_WPSOffice_Level2) [19](#_Toc1366211193_WPSOffice_Level2)

[4.2.1寻找最佳的 n\_estimators：](#_Toc1016367027_WPSOffice_Level3) [19](#_Toc1016367027_WPSOffice_Level3)

[4.2.2寻找最佳的 max\_depth：](#_Toc995694551_WPSOffice_Level3) [19](#_Toc995694551_WPSOffice_Level3)

[4.2.3寻找最佳的 min\_child\_weight](#_Toc1445741233_WPSOffice_Level3) [20](#_Toc1445741233_WPSOffice_Level3)

[4.2.4寻找最佳的gamma值：](#_Toc1942920873_WPSOffice_Level3) [20](#_Toc1942920873_WPSOffice_Level3)

[4.2.5寻找最佳的subsample值： 2](#_Toc34776229_WPSOffice_Level3)0

[4.2.6寻找最佳的colsample\_bytree值： 20](#_Toc1942920873_WPSOffice_Level3)

[4.2.7寻找最佳的 eta 值： 2](#_Toc34776229_WPSOffice_Level3)0

[4.2 模型调参后结果展示 2](#_Toc1366211193_WPSOffice_Level2)0

[五、 比赛心得](#_Toc147667628_WPSOffice_Level1) [21](#_Toc147667628_WPSOffice_Level1)

写在前面：代码和相关资源在Github上

Github账号：[2350792144@qq.com](mailto:2350792144@qq.com)

密码：LOVE2350792144u

题目网站：

[服装分类 竞赛 - DataFountain](https://www.datafountain.cn/competitions/490/submits?view=submit-records)

**一、课题设计背景**

**1.1**赛题介绍

**赛题名：**服装分类练习

**赛道：**训练赛道

**背景：**在日常生活中，随着人们生活水平的日益提高，服装变得越来越多样化，在大型服装工厂中如何做到服装分类变得尤为重要，需要投入大量人力物力，本项目旨在设计一款基于XGBoost的服装分类器，从而能够减少劳动力的投入。

分类器步骤：第一，服装图片采集并进行特征提取。

第二，基于提取的特征进行模型学习和预测

其中包含10种分类：T恤、裤子、套头衫、连衣裙、大衣、凉鞋、衬衫、运动鞋、包、短靴。

**任务：**本任务旨在构建一种机器学习算法模型，建立振动信号和“亚健康”状态之间的关系，通过一系列手段，使得模型具有更高的准确率、更好的鲁棒性和泛化性。

**这是一个多分类的问题**。通过这道赛题来引导大家了解XGBoost在生活中的应用，帮助竞赛新人进行自我练习、自我提高。

**1.2**比赛基本数据集

一般而言，对于数据在比赛界面都有对应的数据概况介绍（匿名特征除外），说明列的性质特征。了解列的性质会有助于我们对于数据的理解和后续分析。

数据集来源：**比赛平台提供的数据记录**，总数据量超过6万。

1.2.1 训练数据集---fashion-mnist\_train.csv

- pixel1至pixel784为特征列

- label 为服装类别（0、1、2、3、4、5、6、7、8、9）

1.2.2 测试数据集---fashion-mnist\_test\_data.csv

- pixel1至pixel784为特征列

需要预测服装类别。

**1.3打分指标**

本练习赛采用正确率（Accuracy）为评测标准。  
在分类任务中，正确率是更为直观的一种衡量方法，即统计样本预测值与实际值一致的情况占整个样本的比例（衡量样本被正确标注的数量），即score = 正确数/总数。

**1.4分类的意义与目的**

希望参赛选手可以通过本次比赛完成对服装的基本分类，对于服装的分类有助于减轻劳动力，并且可以提高分类的准确度。

1. **实现原理概述**

**2.1 数据处理**

1. 载入各种数据科学以及可视化库:

- 数据科学库 pandas、numpy、scipy；

- 可视化库 matplotlib、seabon；

2. 载入数据：

- 载入训练集和测试集；

- 简略观察数据(head()+shape)；

3. 数据总览:

- 通过describe()来熟悉数据的相关统计量

- 通过info()来熟悉数据类型

4. 判断数据缺失和异常

- 查看每列的存在nan情况

- 异常值检测

这里我们使用pandas读取csv文件，提取pixel1至pixel784为特征列作为特征输入，提取label作为分类标签。

**2.2 XGBoost原理介绍**

**(1).特点**

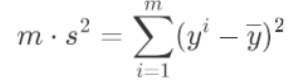
属于GBDT的一种，利用前向分布算法，学习到包含K颗回归树的加法模型。

**(2).首先介绍回归树：**

**回归树的叶节点分裂指标**

通常在CART回归树中，样本的标签是一系列的连续值的集合，不能再使用基尼指数作为划分树的指 标。在回归问题中我们可以发现，对于连续数据， 当数据分布比较分散时，各个数据与平均数的差的平方和较大，方差就较大；当数据分布比较集中时，各个数据与平均数的差的平方和较小。方差越大，数 据的波动越大；方差越小，数据的波动就越小。因此，对于连续的数据，可以使用样本与平均值的差的 平方和作为划分回归树的指标。

方差是度量数据分布离散程度最常用的一种指标，对于包含m个训练样本的数据集D{(X(1),y(1)), (X(2),y(2)),…,(X(m),y(m))}，则指标为数据集D中所有样本标签与均值的差的平方和：



先定义样本被划分到左右子树的过程函数，原理为根据特征fea位置处的特征，按照值value将样本 划分到左右子树中，当样本在特征fea处的值大于或者等于value时，将其划分到右子树中；否则，将其 划分到左子树中。

另外需要定义计算当前叶子节点的值，计算的方法是使用划分到该叶子节点的所有样本的标签均值。

在按照特征对上述的数据进行划分的过程中，需要设置划分的终止条件和分类树比较类似。其构建过程可以分为以下几个步骤：

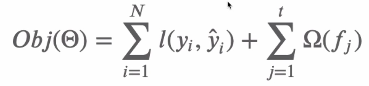
对于当前训练数据集，遍历所有特征及其对应的所有可能切分点，寻找最佳切分特征及其最佳切分 点，使得切分之后的各子集方差和最小，利用该最佳切分特征及其最佳切分点将训练数据集切分成 两个子集，分别对应判别结果为左子树和判别结果为右子树。

重复以下的步骤直至满足停止条件：为每一个叶子节点寻找最佳切分特征及其最佳切分点，将其划 分为左右子树。

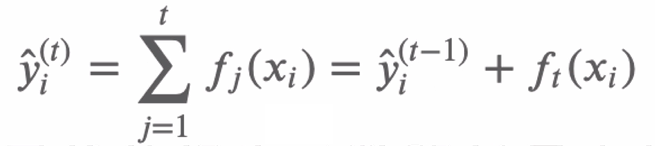
生成回归树。

**(3).XGBoost原理：**

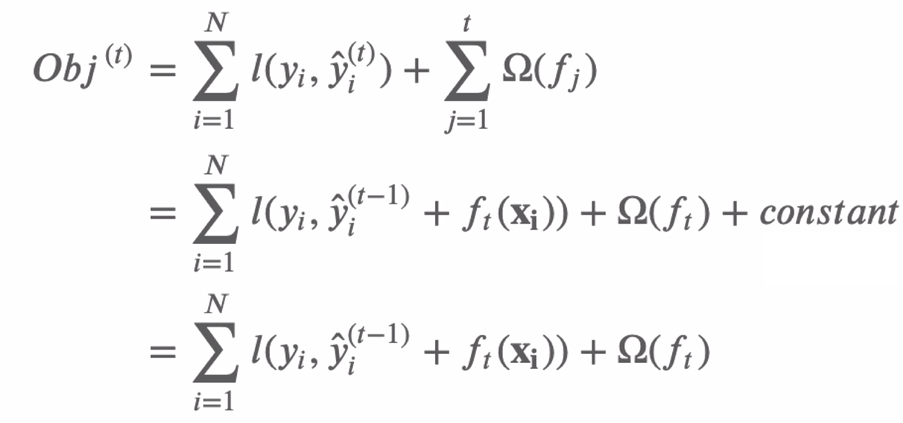
**定义目标函数，包含正则项，确定打分项W：**



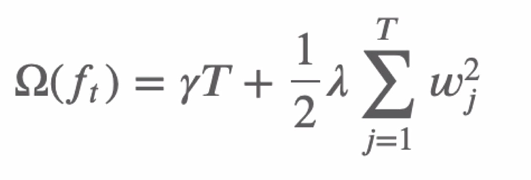
**考虑第K颗回归树的构建：**



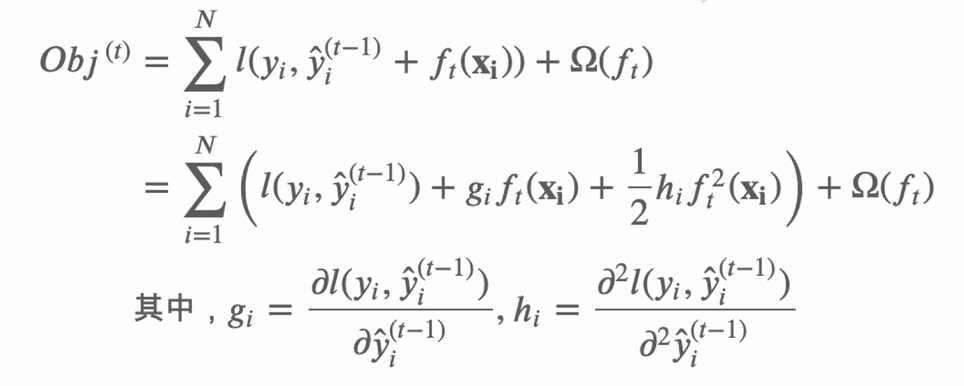
**第K棵树的目标函数为（由损失函数加上正则化项）：**



**正则化项如下，T为回归树叶子节点个数，w为叶子节点的打分项**



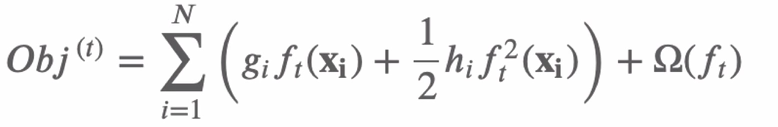
**对目标函数进行泰勒展开后得到**



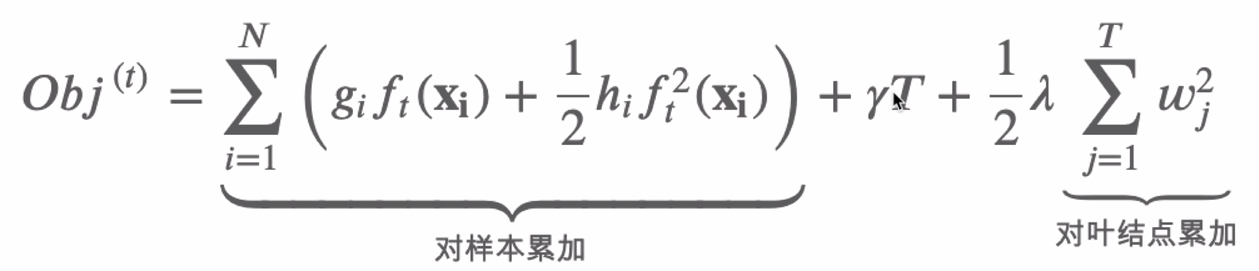
**移除对于第t轮迭代来说的常数项**



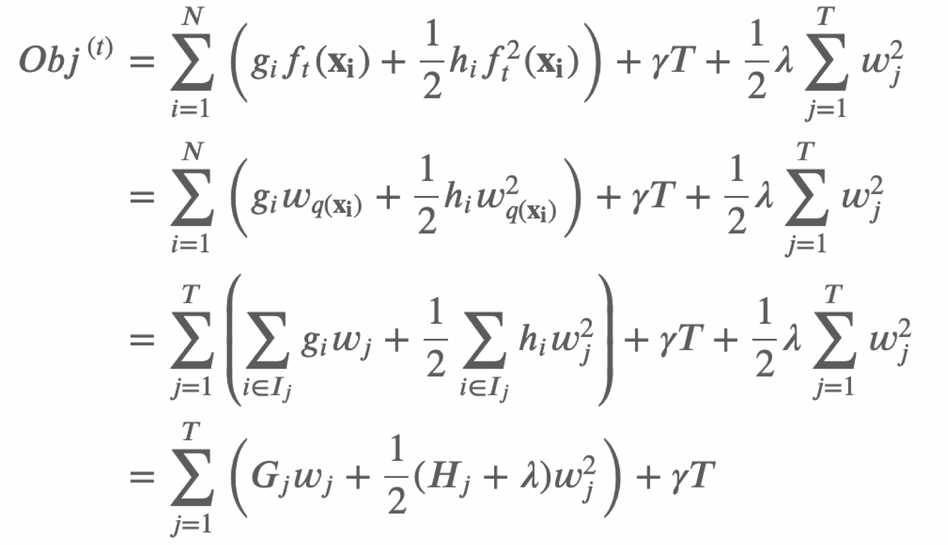
**之后得到：**



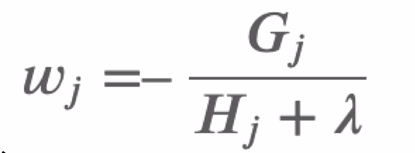
**所以目标函数只依赖于每条数据在误差函数上的一阶导数和二阶导数。目标函数改写为：**



**为了将两者统一起来，作用对象都落在叶结点上。定义函数q将输入x映射到某个叶子节点上，则有：**

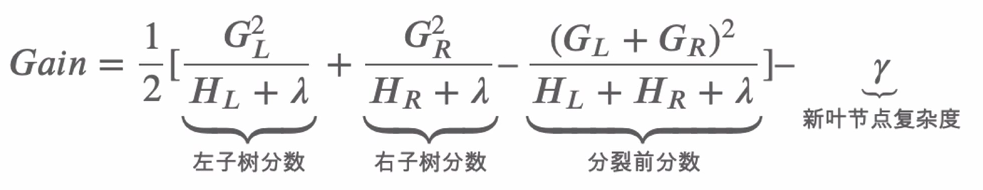


**将目标函数对W求导得到，并令倒数为0，得到：**





**叶节点如何分裂？引入信息增益概念，通过信息增益来衡量划分的优劣：**



**2.3实验预期目标**

matlibplot对数据特征重要程度进行可视化展示，输出预测分类结果准确率。

1. **具体代码实现**

**3.1 数据处理**

利用pandas读入csv文件，并进行相应的特征提取，转换为narray类型矩阵。代码实现如图3-1：



图3-1

输出结果如图3-2：

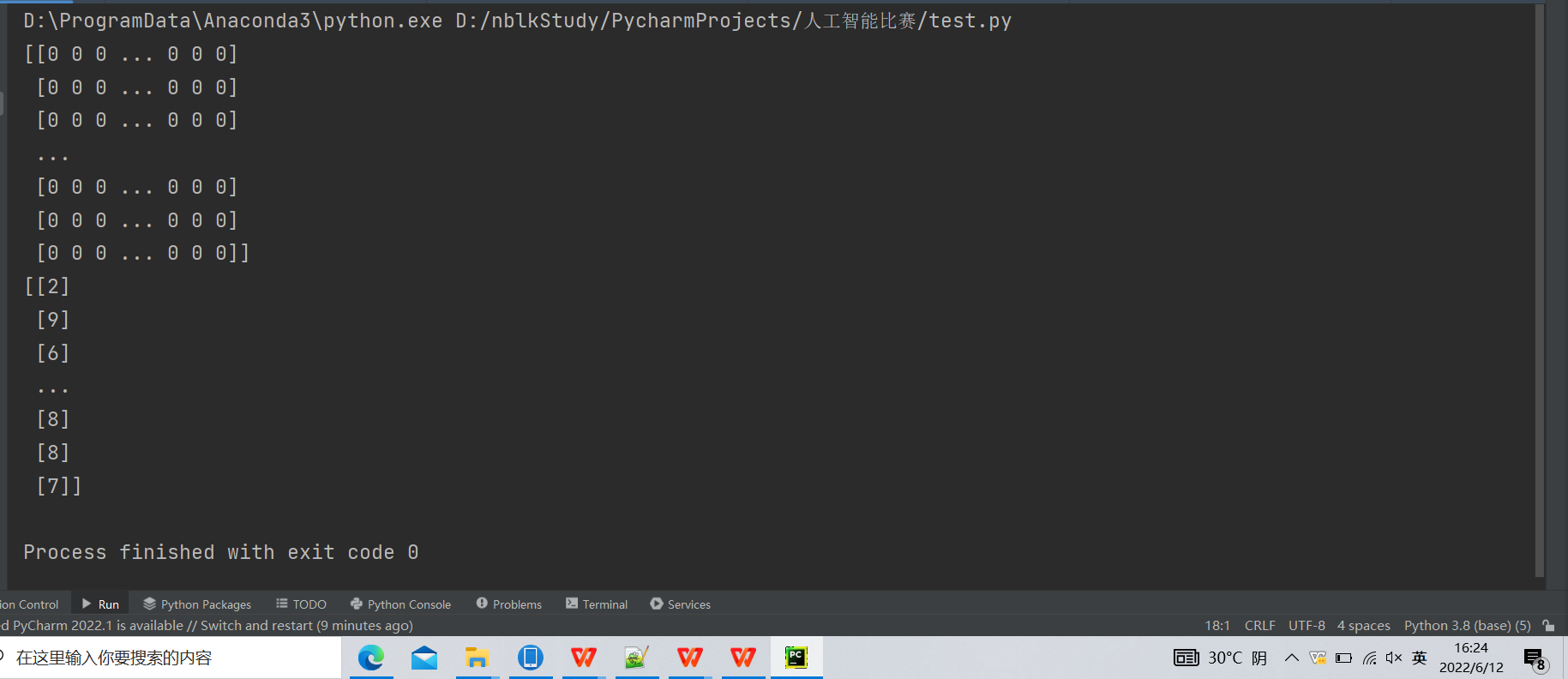


图3-2

**3.2 算法参数设置**  
eta：和GBM中的learning rate参数类似。通过减少每一步的权重，可以提高模型的稳定性。 典型值为 0.01-0.2。比如我们目标值是4 ，第一棵树权重为3.3 ，第二颗树权重只能学0.7，这样后面的树就学不了什么了，所以可以设置eta，来防止过拟合，设置eta=0.3 表示剩下的权重为4-3.3\*0.7。削弱每棵树的影响，让后面有更大的学习空间。  
min\_child\_weight：最小叶子节点权重和，如果在一次分裂中，叶子节点上所有样本的权重和小于min\_child\_weight則停止分裂，能够有效的防止过拟合，防止学到特殊样本。  
max\_depth：树的最大深度，典型值：3-10  
max\_leaf\_nodes：树上叶子节点数。  
gamma ：惩罚项那个和叶子节点结合的项  
subsample：每棵树随机采样的样本的比例，减小这个参数的值，算法会更加保守，避免过拟合。但是，如果这个值设置得过小，它可能会导致欠拟合。 典型值：0.5-1  
colsample\_bytree：用来控制每棵随机采样的列数的占比 (每一列是一个特征)。 典型值：0.5-1  
lambda：权重的 L2 正则化项。  
alpha：权重的 L1 正则化项  
objective：定义损失函数，常用的值  
reg:linear –线性回归。  
reg:logistic–逻辑回归。  
binary:logistic –二分类的逻辑回归问题，输出为概率。  
multi:softmax –让XGBoost采用softmax目标函数处理多分类问题，同时需要设置参数num\_class（类别个数）

如图3-3所示：

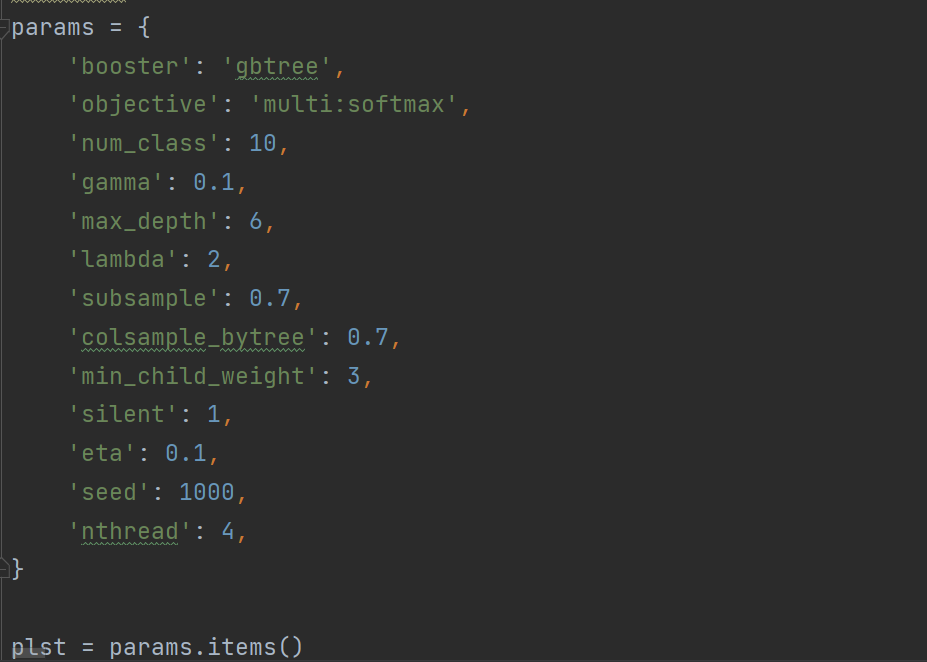
****

图3-3

**3.3 XGBoost模型训练**

3.3.1 模型训练代码如图3-4所示：

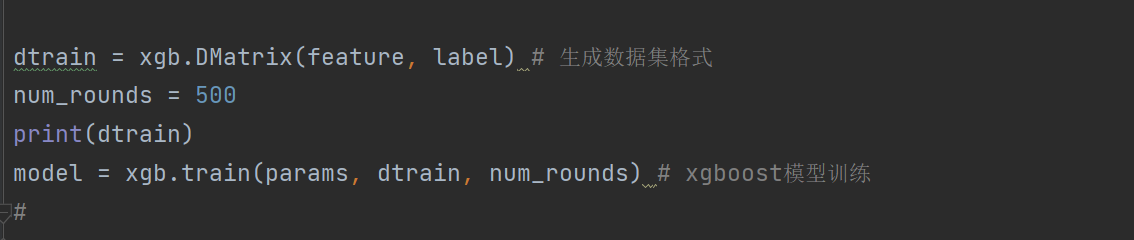


图3-4

3.3.2 基于训练数据，进行主成分分析（特征重要程度），通过matplotlib进行特征重要程度展示，如图3-5：

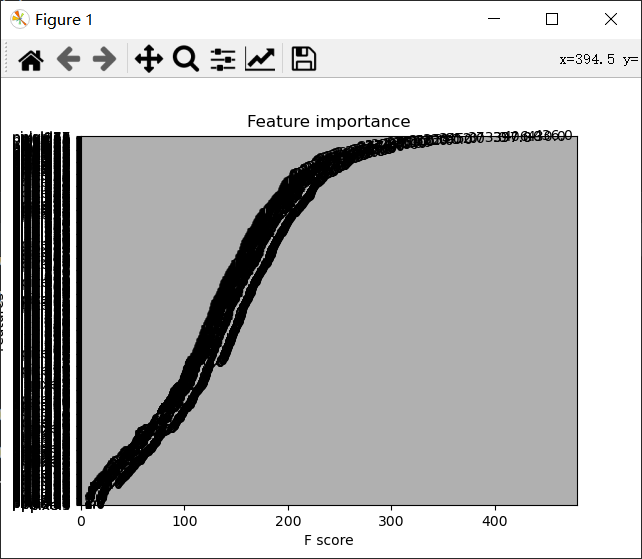


图3-5

3.3.3 在训练集上测试准确率，结果如3-6所示：

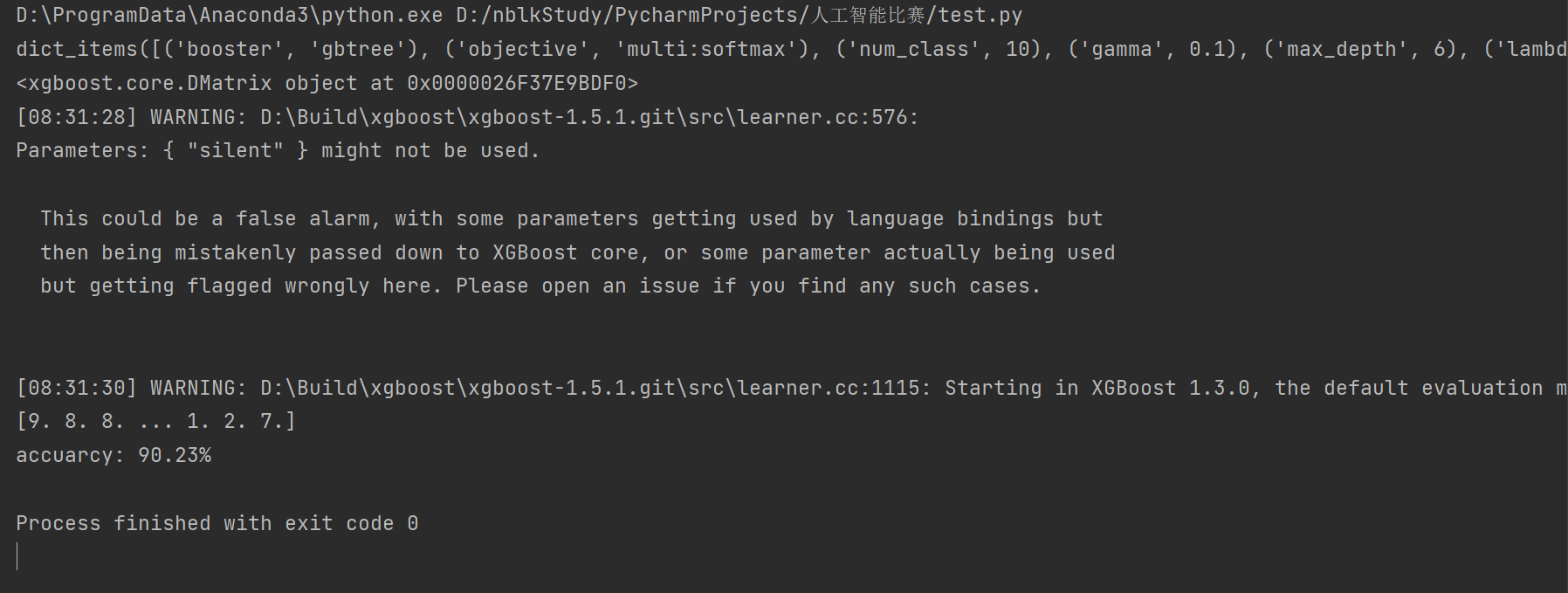


图3-6

**3.4 模型测试**

3.4.1 模型测试代码如图3-7所示：



图3-7

3.4.2 模型测试结果如图3-8、3-9所示：



图3-8



图3-9

1. **模型调参及优化**

**4.1 模型参数介绍**

eta ： 默认是0.3，别名是 leanring\_rate，更新过程中用到的收缩步长，在每次提升计算之后，算法会直接获得新特征的权重。 eta通过缩减特征的权重使提升计算过程更加保守；[0,1]

gamma：默认是0，别名是 min\_split\_loss，在节点分裂时，只有在分裂后损失函数的值下降了（达到gamma指定的阈值），才会分裂这个节点。gamma值越大，算法越保守（越不容易过拟合）；[0，∞]

max\_depth：默认是6，树的最大深度，值越大，越容易过拟合；[0，∞]

min\_child\_weight：默认是1，决定最小叶子节点样本权重和，加权和低于这个值时，就不再分裂产生新的叶子节点。当它的值较大时，可以避免模型学习到局部的特殊样本。但如果这个值过高，会导致欠拟合。[0，∞]

max\_delta\_step：默认是0，这参数限制每颗树权重改变的最大步长。如果是 0 意味着没有约束。如果是正值那么这个算法会更保守，通常不需要设置。[0，∞]

subsample：默认是1，这个参数控制对于每棵树，随机采样的比例。减小这个参数的值算法会更加保守，避免过拟合。但是这个值设置的过小，它可能会导致欠拟合。 (0,1]

colsample\_bytree：默认是1，用来控制每颗树随机采样的列数的占比； (0,1]

colsample\_bylevel：默认是1，用来控制的每一级的每一次分裂，对列数的采样的占比； (0,1]

lambda：默认是1，别名是reg\_lambda，L2 正则化项的权重系数，越大模型越保守；

alpha：默认是0，别名是reg\_alpha，L1 正则化项的权重系数，越大模型越保守；

seed：随机数种子，相同的种子可以复现随机结果，用于调参！

n\_estimators：弱学习器的数量

**4.2 模型调参**

**4.2.1 第一步：寻找最佳的 n\_estimators**

cv\_params = {'n\_estimators': np.linspace(100, 1000, 10, dtype=int)}

regress\_model = xgb.XGBRegressor(\*\*other\_params) # 注意这里的两个 \* 号！

gs = GridSearchCV(regress\_model, cv\_params, verbose=2, refit=True, cv=5, n\_jobs=-1)

gs.fit(X, y) # X为训练数据的特征值，y为训练数据的label

# 性能测评

print("参数的最佳取值：:", gs.best\_params\_)

print("最佳模型得分:", gs.best\_score\_)

**注意：每次调完一个参数，要把 other\_params对应的参数更新为最优值！**

**4.2.2 第二步：寻找最佳的 max\_depth**

cv\_params = {'max\_depth': np.linspace(1, 10, 10, dtype=int)}

**4.2.3 第三步：寻找最佳的 min\_child\_weight**

cv\_params = {'min\_child\_weight': np.linspace(1, 10, 10, dtype=int)}

传入的参数列表为等差数列[1,2,3,…,10]，其他部分不变，

**4.2.4 第四步：寻找最佳的gamma值**

cv\_params = {'gamma': np.linspace(0, 1, 10)}

传入的参数列表为等差数列[0，0.1，0.2，…,1.0]，其他部分不变。

**4.2.5 第五步：寻找最佳的subsample值**

cv\_params = {'subsample': np.linspace(0, 1, 11)}

传入的参数列表为等差数列[0，0.1，0.2，…,1.0]，其他部分不变。

**4.2.6 第六步：寻找最佳的colsample\_bytree值**

cv\_params = {'colsample\_bytree': np.linspace(0, 1, 11)[1:]}

**4.2.7 第七步：寻找最佳的 eta 值**  
一般这时候要调小学习率来测试，这里用的是一个等比数列；

cv\_params = {'eta': np.logspace(-2, 0, 10)}

**4.3 模型调参后结果展示**

****

1. **比赛心得**

通过这一个学期的机器学习的学习，掌握了很多比较强的分类预测方法，其中包括SVM,adboost,线性判别分析，随机森林决策树，以及XGBoost等。通过这个竞赛题目，我学会使用XGBoost去解决生活中的分类问题。