目录

[一、实验目的 3](#_Toc28534250)

[二、实验内容 3](#_Toc28534251)

[三、算法原理与分析 4](#_Toc28534252)

[四、实验设计 6](#_Toc28534253)

[4.1 数据预处理 6](#_Toc28534254)

[4.2 模型训练 9](#_Toc28534255)

[五、实验结果与分析 9](#_Toc28534256)

[六、心得体会 11](#_Toc28534257)

# 一、实验目的

通过本次大作业，学习并练习使用Python搭建机器学习模型。具体任务为：对adult\_train.csv文件数据集进行处理与模型训练，对adult\_test.csv内的数据进行预测，进行准确度评估并求出混淆图矩阵，分析实验结果。

# 二、实验内容

要求针对提供的收入统计数据集（adult），预测年收入是否超过50k$。

该数据从美国1994年人口普查数据库中抽取而来，因此也称作“人口普查收入”数据集，共包含48842条记录，年收入大于50k$的占比23.93%，年收入小于50k$的占比76.07%，数据集已经划分为训练数据32561条和测试数据16281条。该数据集类变量为年收入是否超过50k$，属性变量包括年龄、工种、学历、职业等14类重要信息，其中有8类属于类别离散型变量，另外6类属于数值连续型变量，具体见表1。

表 1 adult数据集属性

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 属性名 | 类型 | 含义 |
| 1 | Age | Continuous | 年龄 |
| 2 | Workclass | Discrete | 工作类别 |
| 3 | Fnlwgt | Continuous | 序号 |
| 4 | Education | Discrete | 受教育程度 |
| 5 | education-num | Continuous | 受教育时间 |
| 6 | marital-status | Discrete | 婚姻状况 |
| 7 | Occupation | Discrete | 职业 |
| 8 | Relationship | Discrete | 社会角色 |
| 9 | Race | Discrete | 种族 |
| 10 | Sex | Discrete | 性别 |
| 11 | capital-gain | Continuous | 资本收益 |
| 12 | capital-loss | Continuous | 资本支出 |
| 13 | hours-per-week | Continuous | 每周工作时间 |
| 14 | native-country | Discrete | 国籍 |

本次大作业，将提供两个文件adult\_train和adult\_test，建立一个收入的预测模型。注意在训练模型时，不得使用测试样本。

设计者可以采用任何分类算法及数据预处理方法。

报告中请给出运行结果的混淆图confusion matrix，并分析原因

|  |  |  |
| --- | --- | --- |
| % | 估计样本0 | 估计样本1 |
| 真实样本0 |  |  |
| 真实样本1 |  |  |

# 三、算法原理与分析

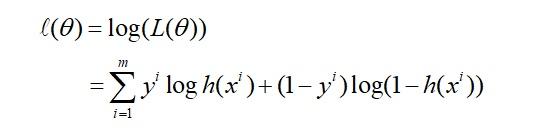
此处主要讲解建模的四种方法原理：包括逻辑回归，决策树，adaboost，和xgboost的原理。

1）逻辑回归

逻辑回归是一种广义线性回归分析模型，其中最常见的为二分类的多项式回归。其公式为：

https://pic.rmb.bdstatic.com/2caf3c84ebb361da259d528394610e732929.gif

X为特征，b为偏置值，θ为各特征的权值，逻辑回归通过sigmod函数归一到（0,1）间，然后构建似然函数，最终推出θ的迭代更新表达式：



逻辑回归的优缺点如下所示；

优点：

（1）预测结果是介于0和1之间的概率；

（2）可以适用于连续性和类别性自变量；

（3）容易使用和解释。

缺点：

（1）对模型中自变量多重共线性较为敏感，例如两个高度相关自变量同时放入模型，可能导致较弱的一个自变量回归符号不符合预期，符号被扭转。需要利用因子分析或者变量聚类分析等手段来选择代表性的自变量，以减少候选变量之间的相关性；

（2）预测结果呈“S”型，因此从log(odds)向概率转化的过程是非线性的，在两端随着log(odds)值的变化，概率变化很小，边际值太小，slope太小，而中间概率的变化很大，很敏感。 导致很多区间的变量变化对目标概率的影响没有区分度，无法确定阀值。

2）决策树

决策树也是一种有监督的机器学习算法，其组织结构为树，其内部节点表示在某属性上的判断，通过对样本的学习，可以得到一棵决策树，并根据该决策树对新的数据进行分类，其原理是基于信息熵和信息增益。。常用的决策树包括ID3,C4.5和CART。

决策树的优缺点如下所示：

优点:

(1)速度快: 计算量相对较小, 且容易转化成分类规则. 只要沿着树根向下一直走到叶, 沿途的分裂条件就能够唯一确定一条分类的谓词.

(2)准确性高: 挖掘出来的分类规则准确性高, 便于理解, 决策树可以清晰的显示哪些字段比较重要, 即可以生成可以理解的规则.

(3)可以处理连续和种类字段

(4)不需要任何领域知识和参数假设

(5)适合高维数据

缺点:

(1)对于各类别样本数量不一致的数据, 信息增益偏向于那些更多数值的特征

(2)容易过拟合

(3)忽略属性之间的相关性

3）Adaboost

AdaBoost，是英文"Adaptive Boosting"（自适应增强）的缩写，由Yoav Freund和Robert Schapire在1995年提出。adaboost算法首先为训练集中的每个样本赋予一个均分权重，构成向量D；接着训练第一个弱分类器，计算其错误率e，再调整样本权重，提高误分类样本权值，进行下一轮训练。并且，由于最终结果是由各分类器投票产生，所以还需为每个分类器计算权重值α，α是根据错误率e更新的，e越小，α越大。训练过程不断持续，直至e为0或分类器数目达到指定值时算法停止。

Adboost的优缺点如下所示：

优点：

(1) 泛化错误率低，无需参数调整

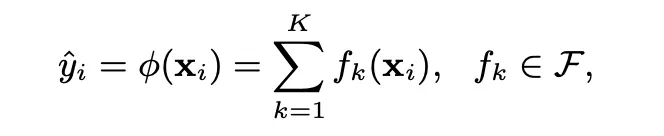
缺点：

(2) 对离群点敏感

4）XGBoost

XGBoost全名叫（eXtreme Gradient Boosting）极端梯度提升，经常被用在一些比赛中，其效果显著。它是大规模并行boosted tree的工具，它是目前最快最好的开源boosted tree工具包。XGBoost 所应用的算法就是 GBDT（gradient boosting decision tree）的改进，既可以用于分类也可以用于回归问题中，本质模型为加法模型，基函数为决策树，迭代拟合标注和模型的残差，来不断逼近损失函数最小化。更一般的情况，以当前模型在L负梯度方向的值，作为残差的近似。

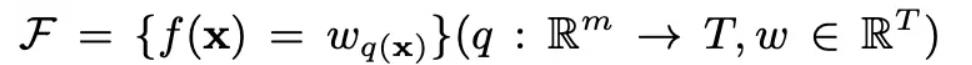
他将K（树的个数）个树的结果进行求和，作为最终的预测值。即：



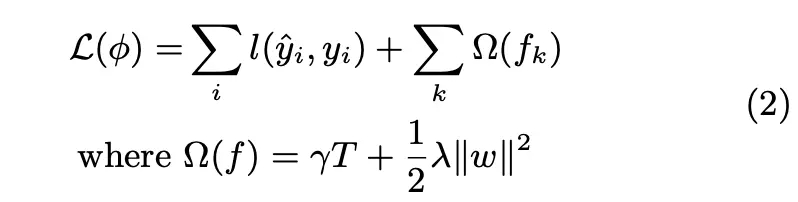
假设给定的样本集有n个样本，m个特征，则：



其中 xi 表示第i个样本，yi 表示第i个类别标签，回归树（CART树）的空间F为：



其中q代表每棵树的结构，他将样本映射到对应的叶节点；T是对应树的叶节点个数；f(x)对应树的结构q和叶节点权重w。我们的目标是学习这k个树，所以我们最小化下面这个带正则项的目标函数：



Xgboost的优点在于：

1.使用许多策略去防止过拟合，如：正则化项、Shrinkage and Column Subsampling等。

2. 目标函数优化利用了损失函数关于待求函数的二阶导数

3.支持并行化，这是XGBoost的闪光点，虽然树与树之间是串行关系，但是同层级节点可并行。具体的对于某个节点，节点内选择最佳分裂点，候选分裂点计算增益用多线程并行。训练速度快。

4.添加了对稀疏数据的处理。

5.交叉验证，early stop，当预测结果已经很好的时候可以提前停止建树，加快训练速度。

6.支持设置样本权重，该权重体现在一阶导数g和二阶导数h，通过调整权重可以去更加关注一些样本。

# 四、实验设计

首先，我们了解数据集的基本结构与数据情况。数据集14个数据属性，8类离散型变量，6类数值型变量。总共包含48842条记录，其中32561条为训练集，16281条作为测试集。之后的实验设计，我们分为数据预处理和模型训练两个部分进行设计：

### 4.1 数据预处理

我们可以看到，部分数据存在缺失，并以’ ?’表示，所以第一步我们要寻找缺失数据并对数据集进行缺失值填充。我们首先搜索数据集中的缺失值并对其缺失数量进行统计与排序，其结果如图4-1所示：



图4-1 缺失数据类型

我们发现，只有三类属性中存在缺失值，接下来我们使用Counter()函数统计一下在存在缺失值中各类别的数量，如图4-2所示：

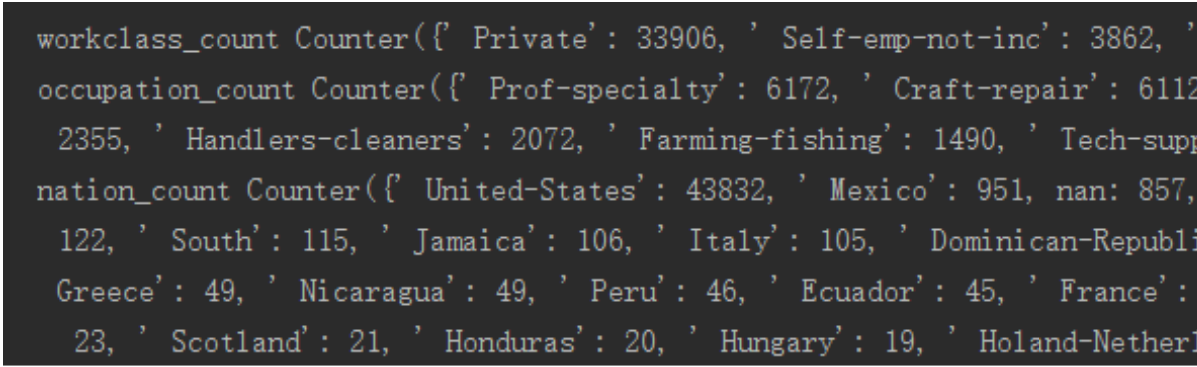


图4-2统计缺失类内数据

针对不同的缺失类型确定不同的填充方式。通过观察我们可以发现：

1. 在国籍属性（nation）和工作类别（workclass）中，' United-States'和‘ private’占了绝大多数，所以将缺失国籍全部填充为' United-States'与’private’。

2. 而在职业类别（occupation）中，最多的一类数据量为6172，而缺失值几乎要超过其类内样本最多的属性的1/3的数量，所以不适合使用众数填充，所以我们暂时标注为‘un-known’，以进行下一步处理。

在填充完缺失值后，我们就要对其中的离散属性进行特征编码和标准化处理，我们这里可以直接使用pd.dummies()函数将离散值进行0ne-hot编码处理，然后对其进行标准化处理以统一量纲，这里我们尝试了z-score标准化和MinMax标准化两种方式，如图4-3所示，并根据其实际效果，最终选择了MinMax标准化方式。这样，我们的数据集就已经转化成为计算机可以处理的数据了。

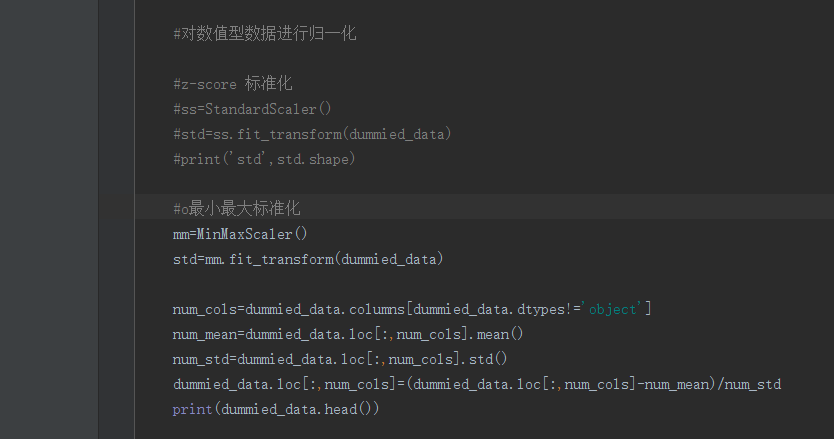


图4-3 标准化

但在训练之前，我们还需要进行特征选择。否则过多的特征会导致模型复杂度提高和泛化能力的下降。特征选择方法分为三类：filter，wrapper和embedded方法。在这里我们选择使用embedded方法中的Lasso方法进行特征选择，通过Lasso方法可以得到一个稀疏矩阵，其中的权值大小可以体现出特征的重要性，我们得到了特征重要性的统计图如图4-4所示：

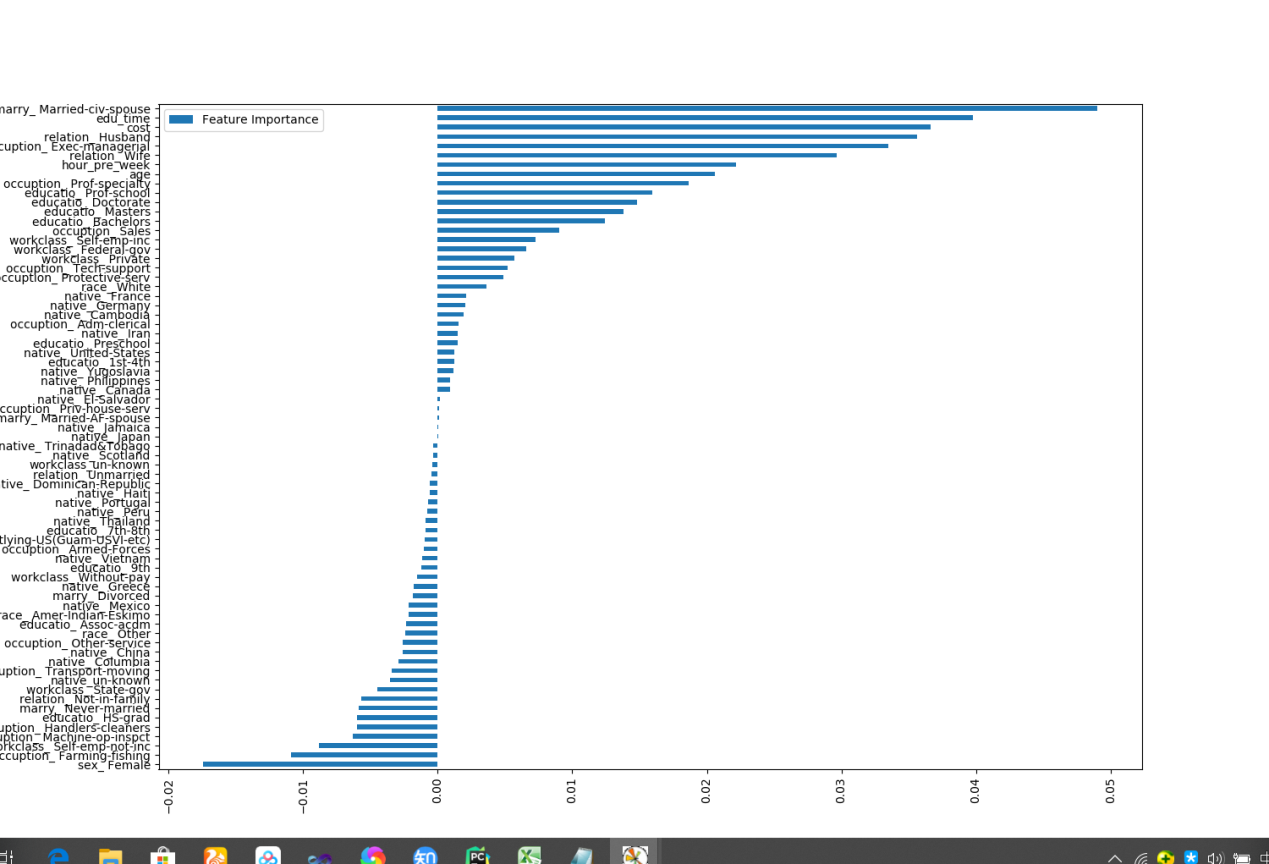


图4-4 初始特征重要性

我们保留其中重要性强的特征，而抛弃重要性小，权值低的特征，然后重新使用Lasso方法进行再次选择，通过多次选择，以及后期的调整，我们最终选择的特征如图4-5所示：

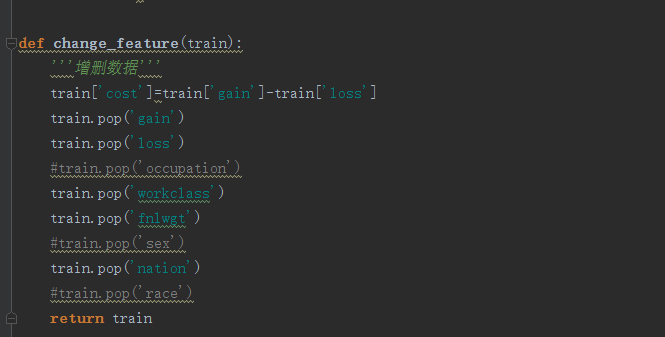


图4-5 最终特征选择结果

选择好特征后，我们进入数据预处理的最后一步，使用PCA对数据进行处理。我们挑选得到的数据特征之间可能存在较大的关联性，而PCA可以用来去除结果的相关性，可以进一步提高我们的训练效率。

## 4.2 模型训练

经过处理的数据被分为了特征属性X\_train和结果标签y\_train两组数据，我们使用不同的模型对X\_train进行拟合，并通过y\_train得到损失函数，来更新训练中得到的权值与各类超参数。最后，在测试集X\_test上进行模型预测，并得到最终对y\_test的预测准确率。我们在此选择了四种模型：逻辑回归，SVM，决策树和集成学习模型xgboost对相关数据进行拟合训练，并简单对其部分参数进行了微调，以期望得到更高的识别准确率。

# 五、实验结果与分析

我们使用了四种模型来预测分类准确率，其结果图像依次如下所示：

逻辑回归准确率最高为85.34%，绘制的混淆图如表5-1所示，图5-1为实验截图：

表5-1

|  |  |  |
| --- | --- | --- |
| % | 估计样本0 | 估计样本1 |
| 真实样本0 | 11896 | 469 |
| 真实样本1 | 1678 | 610 |

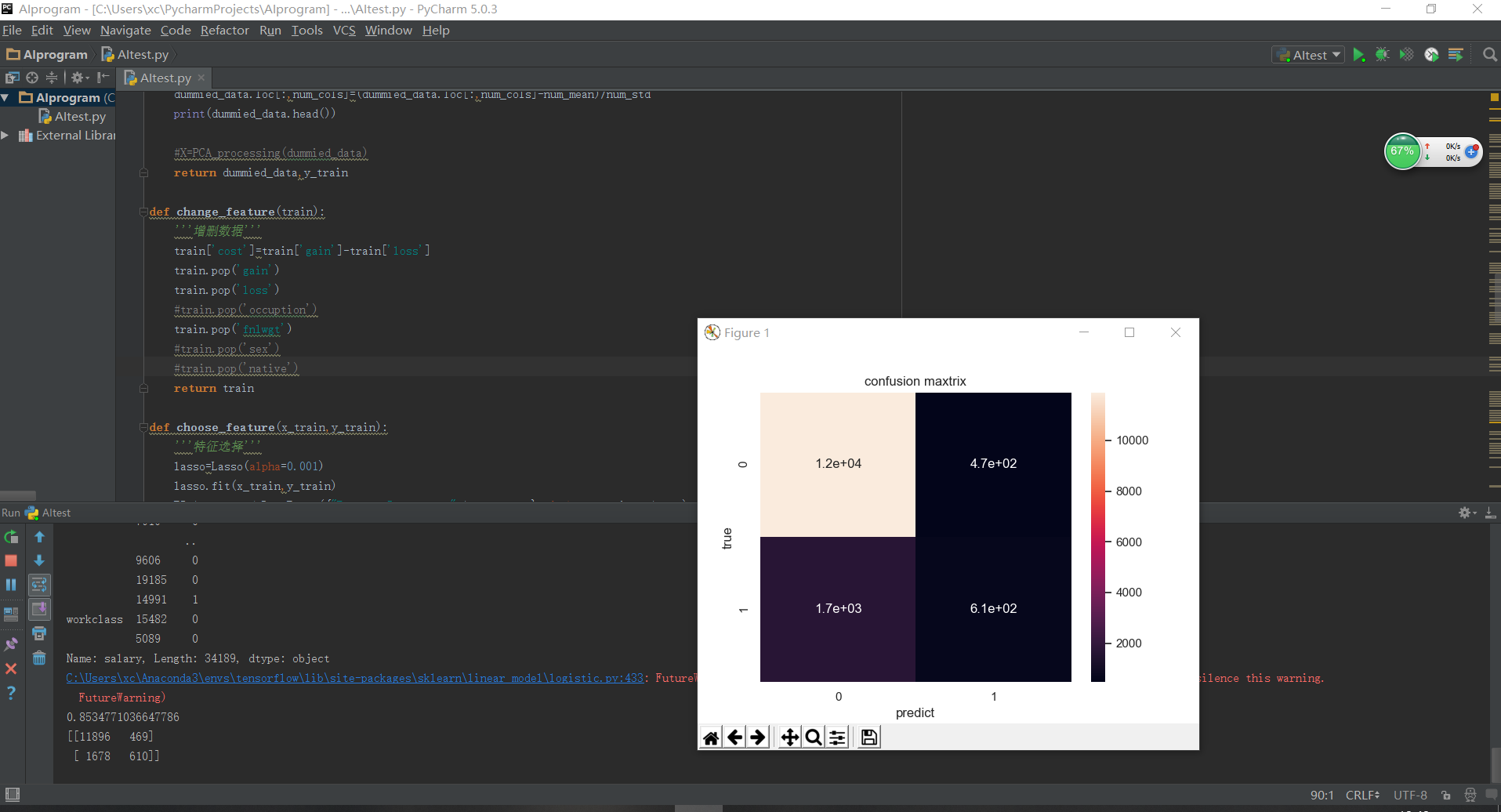


图5-1 逻辑回归实验截图

决策树准确率最高为82.52%，绘制的混淆图如表5-2所示，图5-2为实验截图：

表5-2

|  |  |  |
| --- | --- | --- |
| % | 估计样本0 | 估计样本1 |
| 真实样本0 | 11250 | 1152 |
| 真实样本1 | 1694 | 2185 |

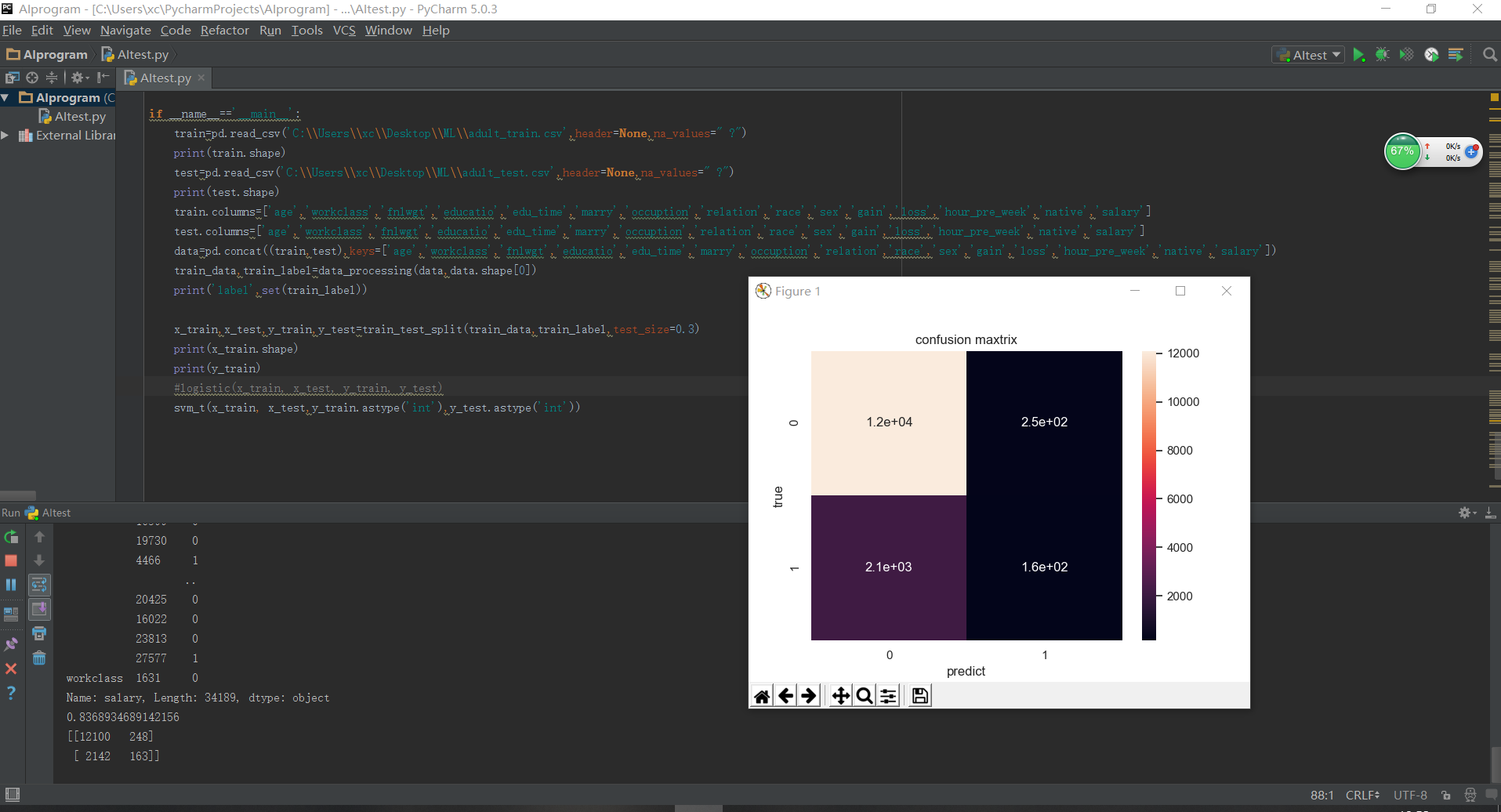


图5-2 决策树实验结果

adaboost准确率最高为86.09%，但由于绘制的混淆图如表5-3所示，图5-3为实验截图：

表5-3

|  |  |  |
| --- | --- | --- |
| % | 估计样本0 | 估计样本1 |
| 真实样本0 | 11618 | 766 |
| 真实样本1 | 1499 | 2398 |

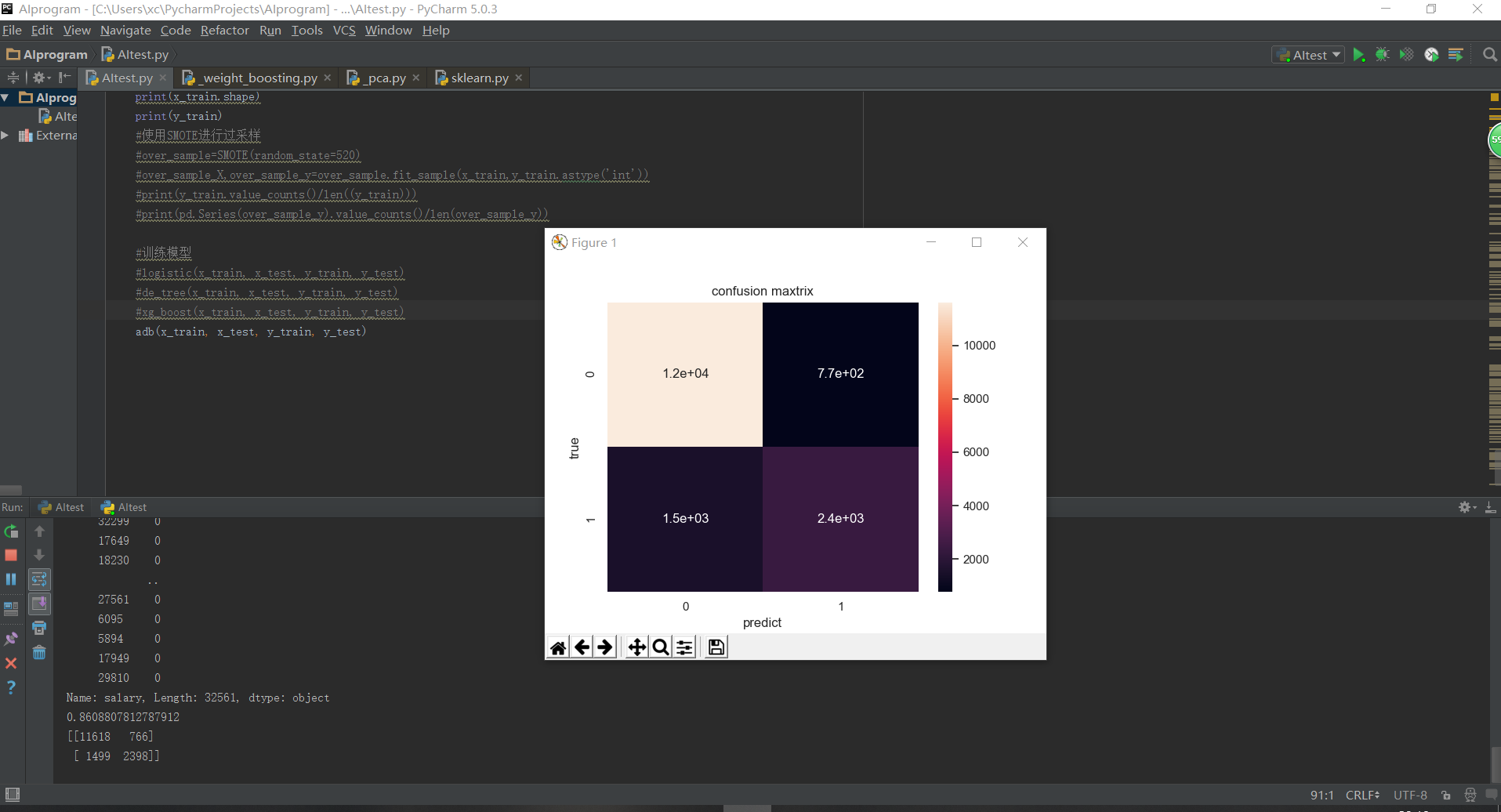


图5-3 Adaboost实验结果

最后使用了集成学习xgboost方式进行预测，其效果最好，逻辑回归准确率最高为87.10%，但由于绘制的混淆图如表5-4所示，图5-4为实验截图：

表5-4

|  |  |  |
| --- | --- | --- |
| % | 估计样本0 | 估计样本1 |
| 真实样本0 | 11802 | 638 |
| 真实样本1 | 1462 | 2379 |

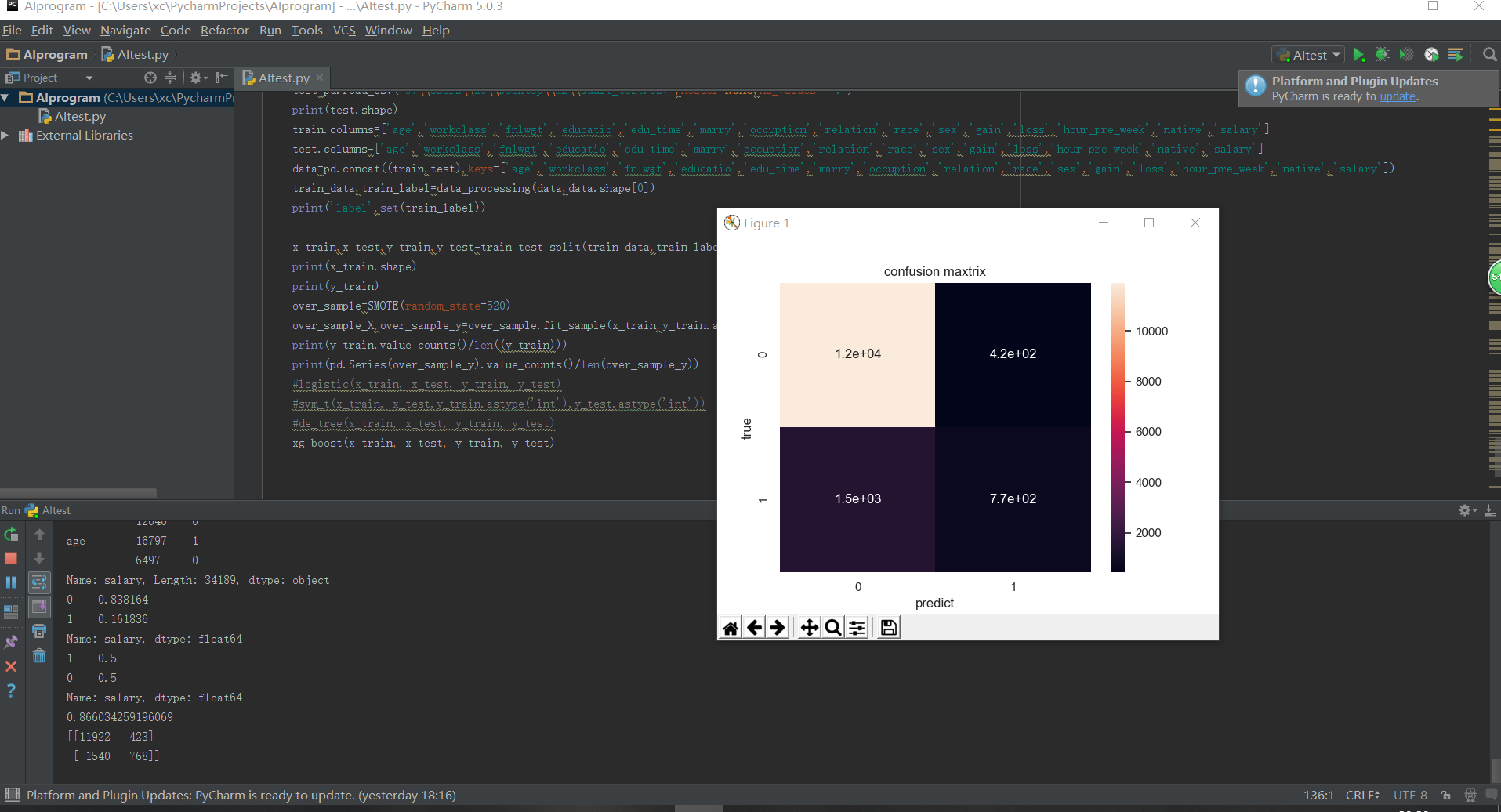


图5-4 XGBoost实验结果

结果分析：

在本次大作业中，经过多种模型的测试与简单调参后，发现集成学习模型表现出了比其他学习模型更强的训练拟合能力，最终精确度达到了87.10%

思考了不能达到更高的准确度的可能原因：

1）特征工程方面还有较大改进空间。

2）在调用训练模型时，基本使用默认参数，可以在从参数调整方面进一步加强。

3）不同标签数据量存在较大差异，在这里，我们曾尝试使用SMOTE对数据进行过采样以平衡数据，但发现结果并不理想，所以放弃使用，可以再对数据进行其他采样方式。

# 六、心得体会

通过本次大作业，我们对机器学习整体流程有了更为清晰的理解，对不同的机器学习模型的优缺点有了更加深入的理解和体会，同时锻炼了我们的代码能力，提升了我们使用Python编写程序的能力，学习了sklearn库的使用，并深刻理解了特征工程在机器学习中的重要性，了解了可能影响模型精度的原因，同时对模型调参方式以及手段有了一定的理解。