**《机器学习》课程期末作业**

（20213385 张听然 计算机2106班）

# 一、选题目的和创新点

## 1.1选题目的

作者在平时科研学习中一直都是以自然语言处理方向作为研究方向，完成项目也主要使用thunlp所开发的openmatch框架。在此框架中，主要是使用transformers和torch两个现有的NLP方向的机器学习库，进行实验。

虽然使用现有的库能极大的简化代码实现难度，但是却让我忽略了机器学习中比较底层的方法论部分，有些时候发现一些生活中的小问题，反而难以通过自己的知识去解决。这引发了我的深思，让我开始寻求一些办法去解决这个问题。在上完逻辑回归部分课程后，我对这个机器学习算法产生了浓厚的兴趣。逻辑回归原理简单，实现较方便，可解释性强，是理解底层机器学习比较好的小模型。所以，实验就围绕着这个想法展开，对逻辑回归进行了探究。

## 1.2创新点

本文不是仅仅使用了逻辑回归进行分类和使用混淆矩阵中精确度和召回率进行模型评估。还进行了以下的工作：

（1）对有偏数据集和无偏数据集对模型训练的影响进行了探究。

（2）对具有缺失值的数据集进行了处理。

（3）用了coefficient系数和summa摘要对模型进行了刻画，尝试解释了解释了模型最终能完成任务的原因。

（4）进行了交叉验证，更好的进行了小数据集上的训练。

（5）结合《人工智能数学基础》、《智能计算系统》、《数据科学导论》的知识，使用了L1和L2范数对模型进行了改良。

# 二、 数据集来历及解释说明

## 2.1数据集来历

来源github网址如下：

[https://github.com/WinVector/zmPDSwR/raw/master /PUMS/psub.RData](https://github.com/WinVector/zmPDSwR/raw/master%20/PUMS/psub.RData)

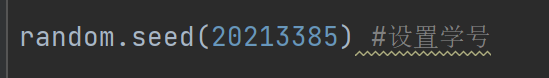
## 2.2数据集解释说明

|  |  |  |  |
| --- | --- | --- | --- |
| **变量类型:** | Categorical, Integer | **属性个数:** | 14 |
| **任务:** | Classification | **是否有缺失值?** | Yes |
| **变量说明：** | | | |
| 预测变量Income： | >50K, <=50K. | | |
| Age: | continuous. | | |
| Workclass: | Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked. | | |
| Fnlwgt: | continuous. | | |
| Education: | Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool. | | |
| Education-num: | continuous. | | |
| Marital-status: | Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse. | | |
| Occupation: | Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces. | | |
| Relationship: | Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried. | | |
| Race: | White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black. | | |
| Sex: | Female, Male. | | |
| Capital-gain: | continuous. | | |
| Capital-loss: | continuous. | | |
| Hours-per-week: | continuous. | | |
| Native-country: | United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad&Tobago, Peru, Hong, Holand-Netherlands. | | |

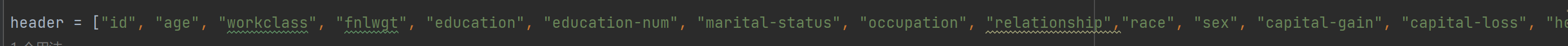
# 三、 随机种子的指定和数据预处理

## 3.1代码实现及解释

（1）输入学号，生成随机数种子



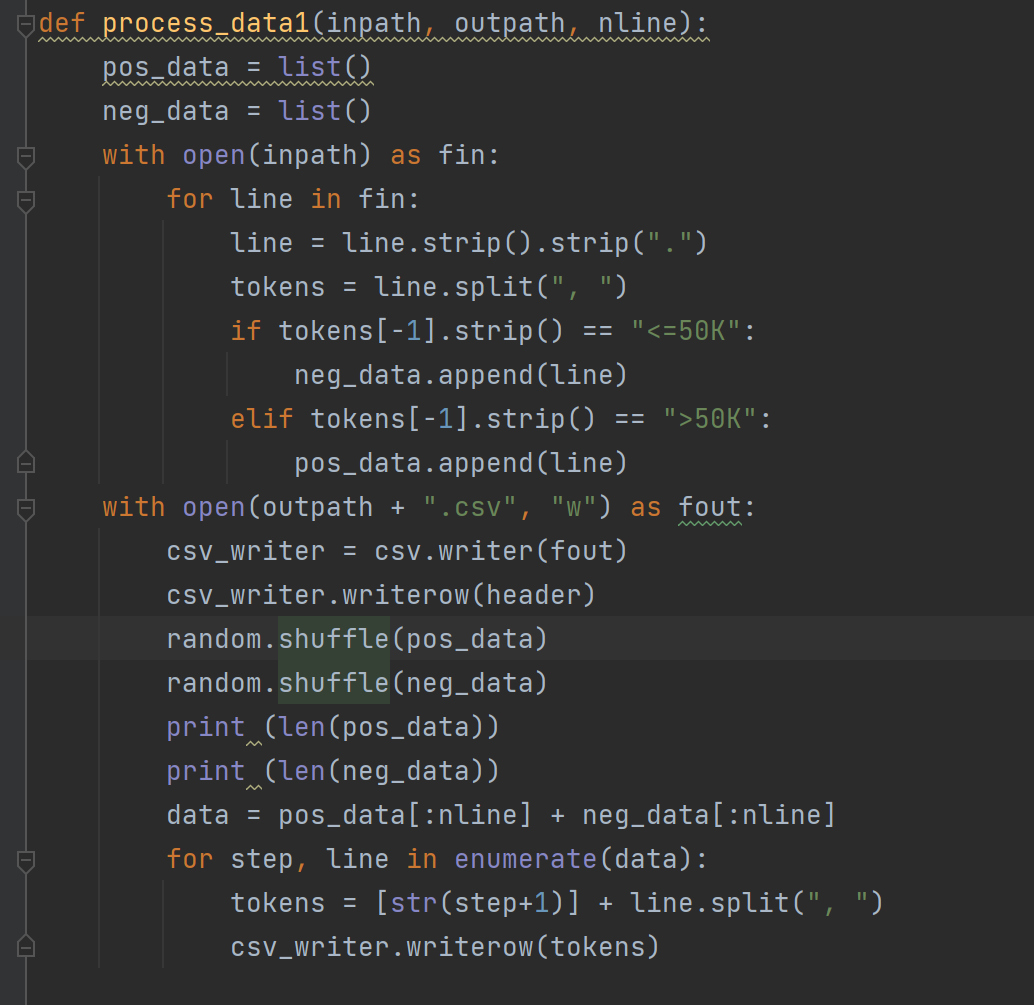
（2）限定CSV文件的表头



（3）处理数据

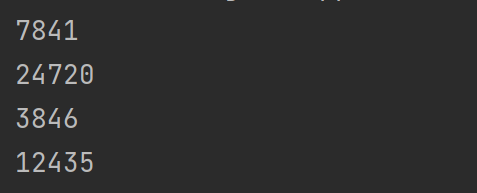
步骤：

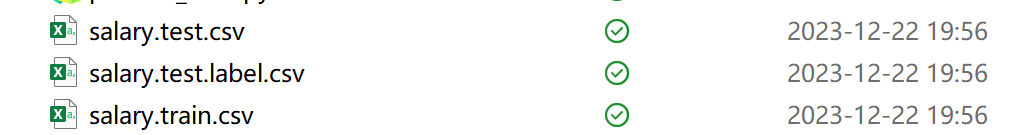
1. 初始化正负样例列表
2. 将行、元素进行分割
3. 将大于50k的样本和小于50k的样本分别放入正样本列表和负样本列表
4. 将正负样本随机打乱后存入CSV文件中
5. 打印各个样本的条数



## 3.2 运行结果

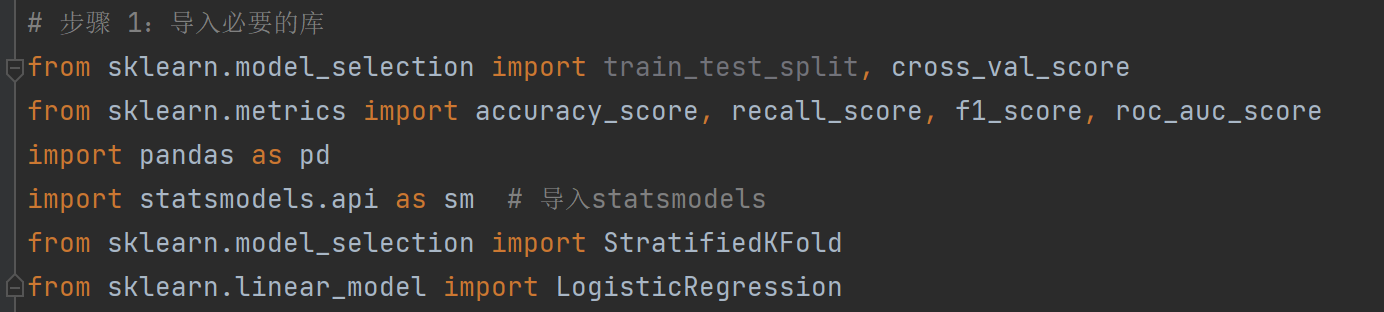
控制台中打印各个样本的条数且在原文件夹中生成了对应的CSV文件。



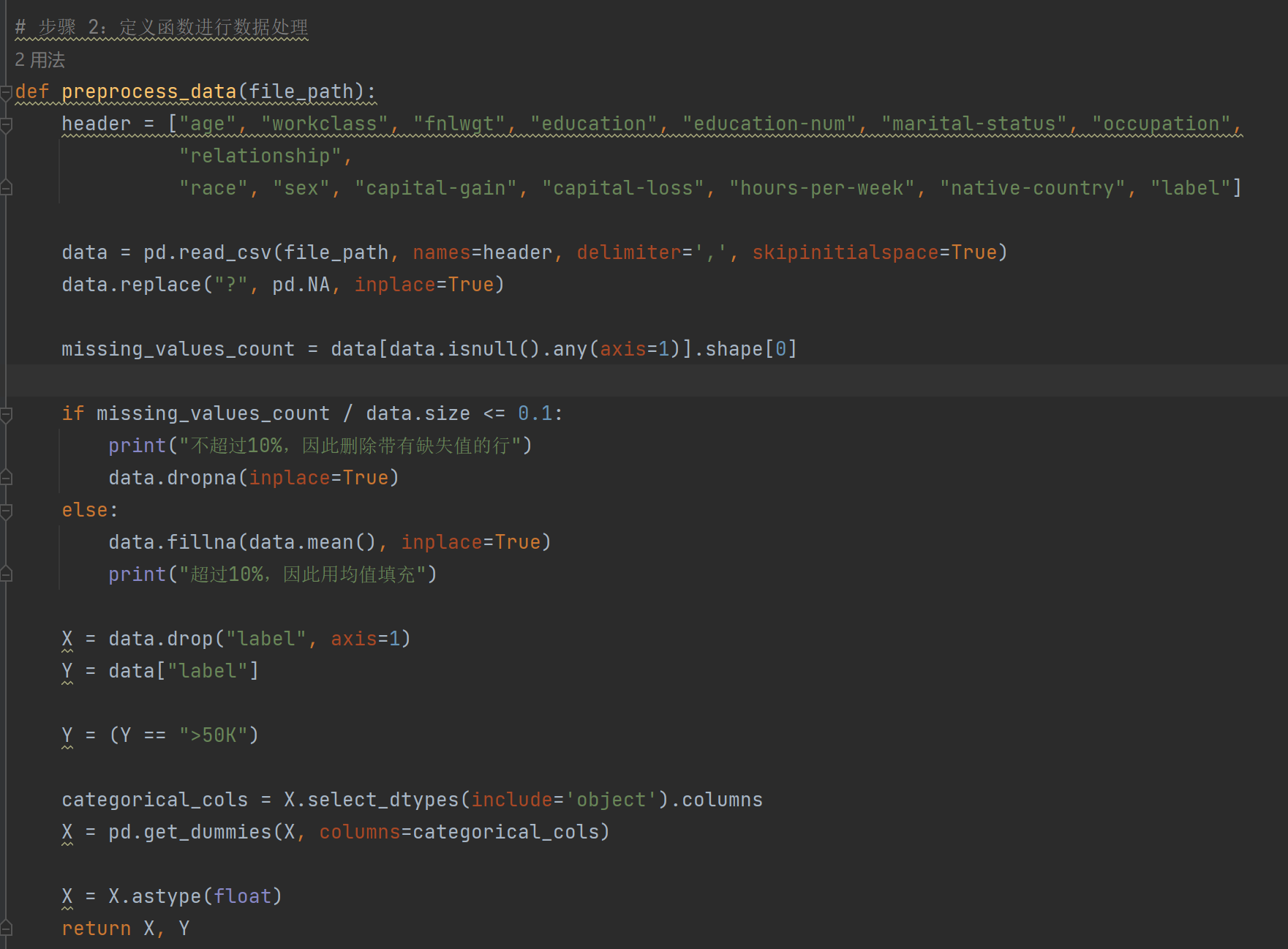


# 四、 在Income数据集上建立模型

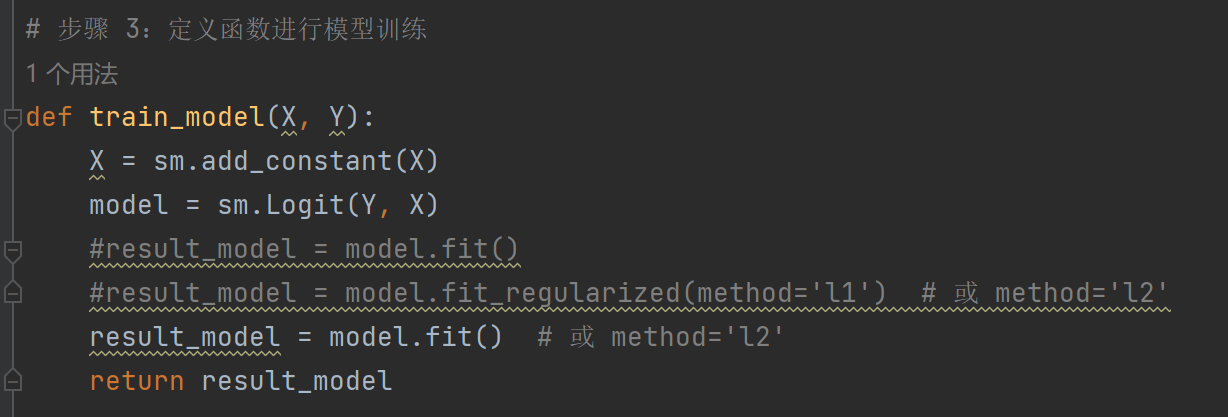
## 4.1代码实现及解释



这里导入了用于数据处理和建模的一些常用库，包括Scikit-Learn和Statsmodels。



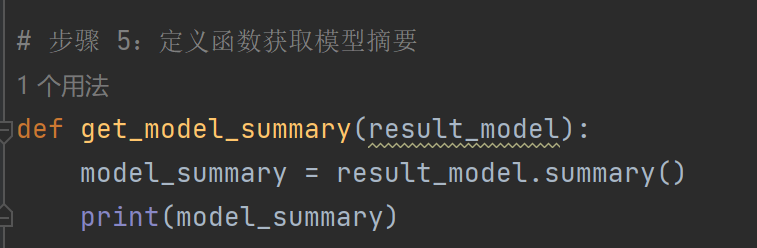
该函数用于读取数据文件，处理缺失值（删除或填充均值），进行独热编码，最终返回特征矩阵X和目标变量Y。



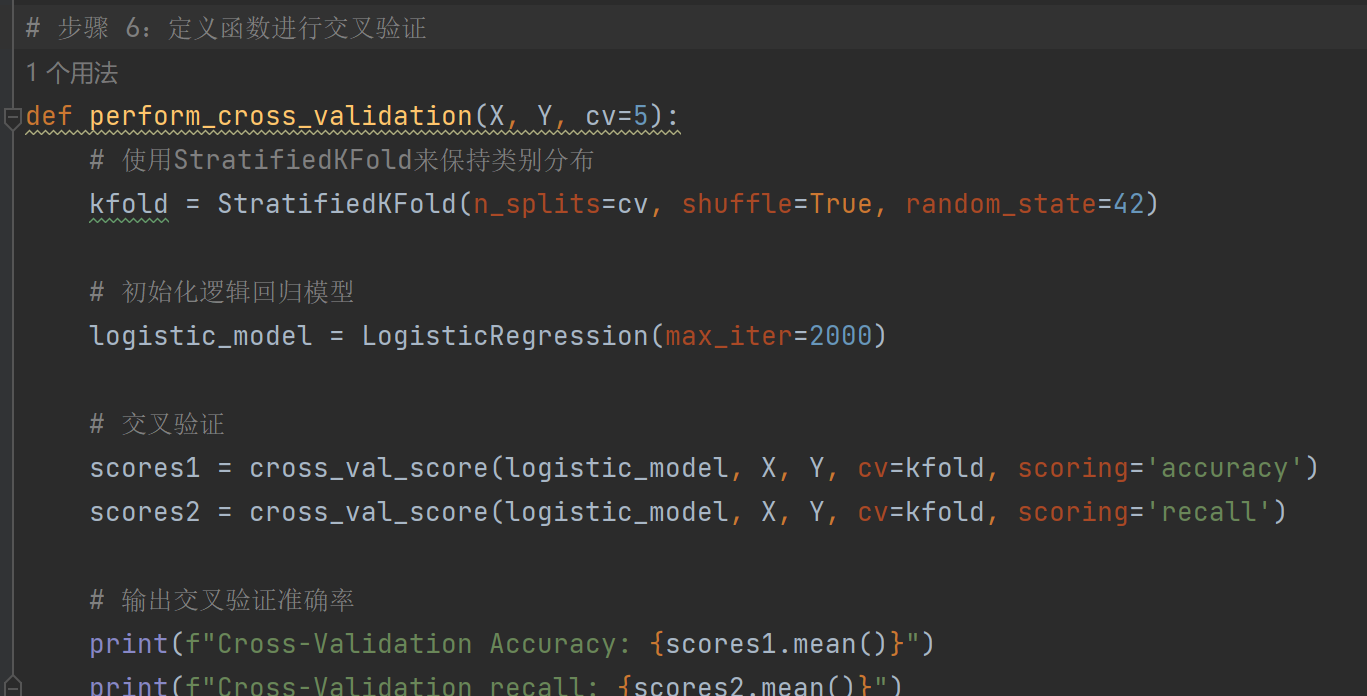
该函数将特征矩阵X和目标变量Y传入Logistic Regression模型中进行拟合，并返回训练好的模型。



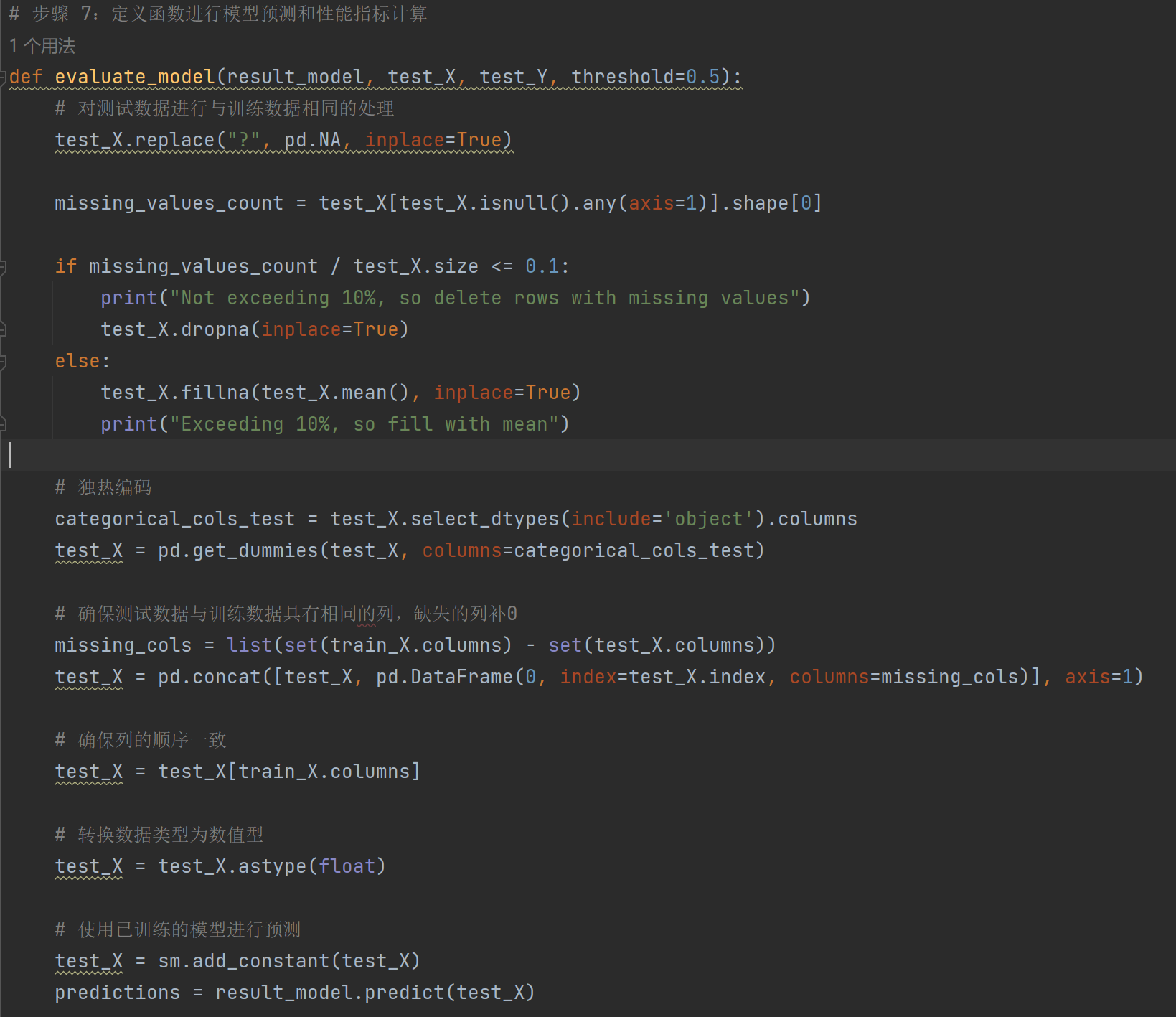
该函数接收特征和对应的系数，然后将它们打印出来。

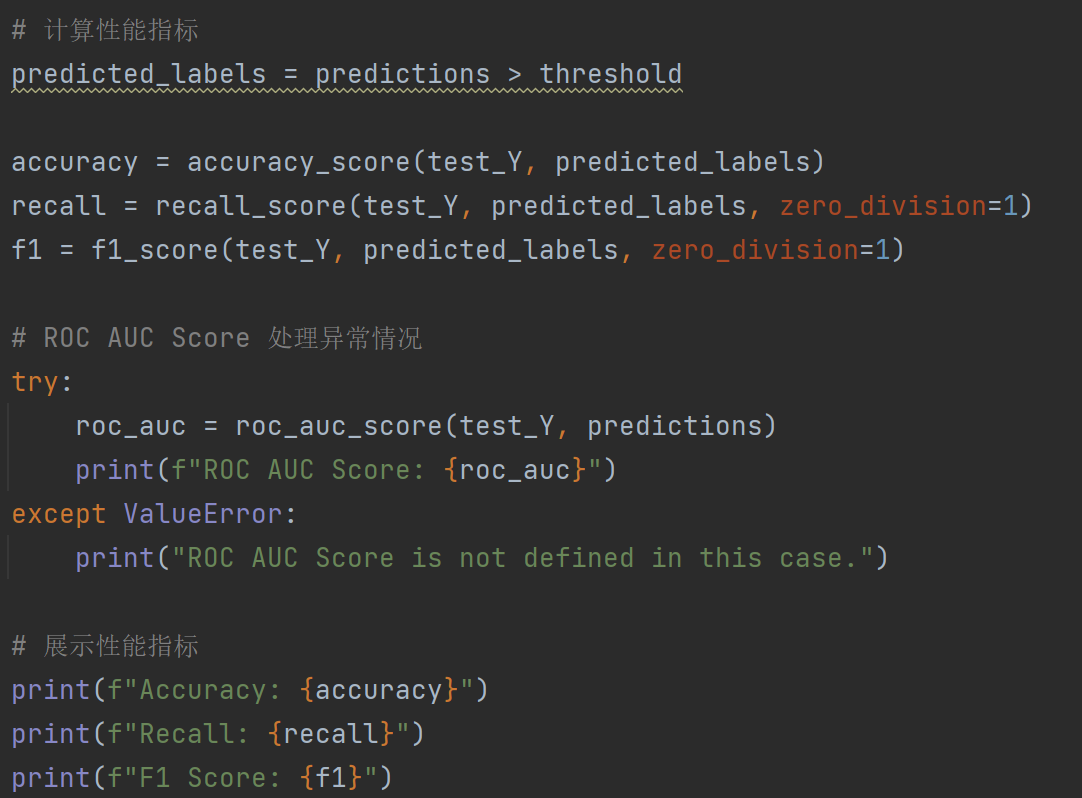


该函数接收训练好的Logistic Regression模型，然后输出模型的摘要信息，包括系数、标准误差、z值等。

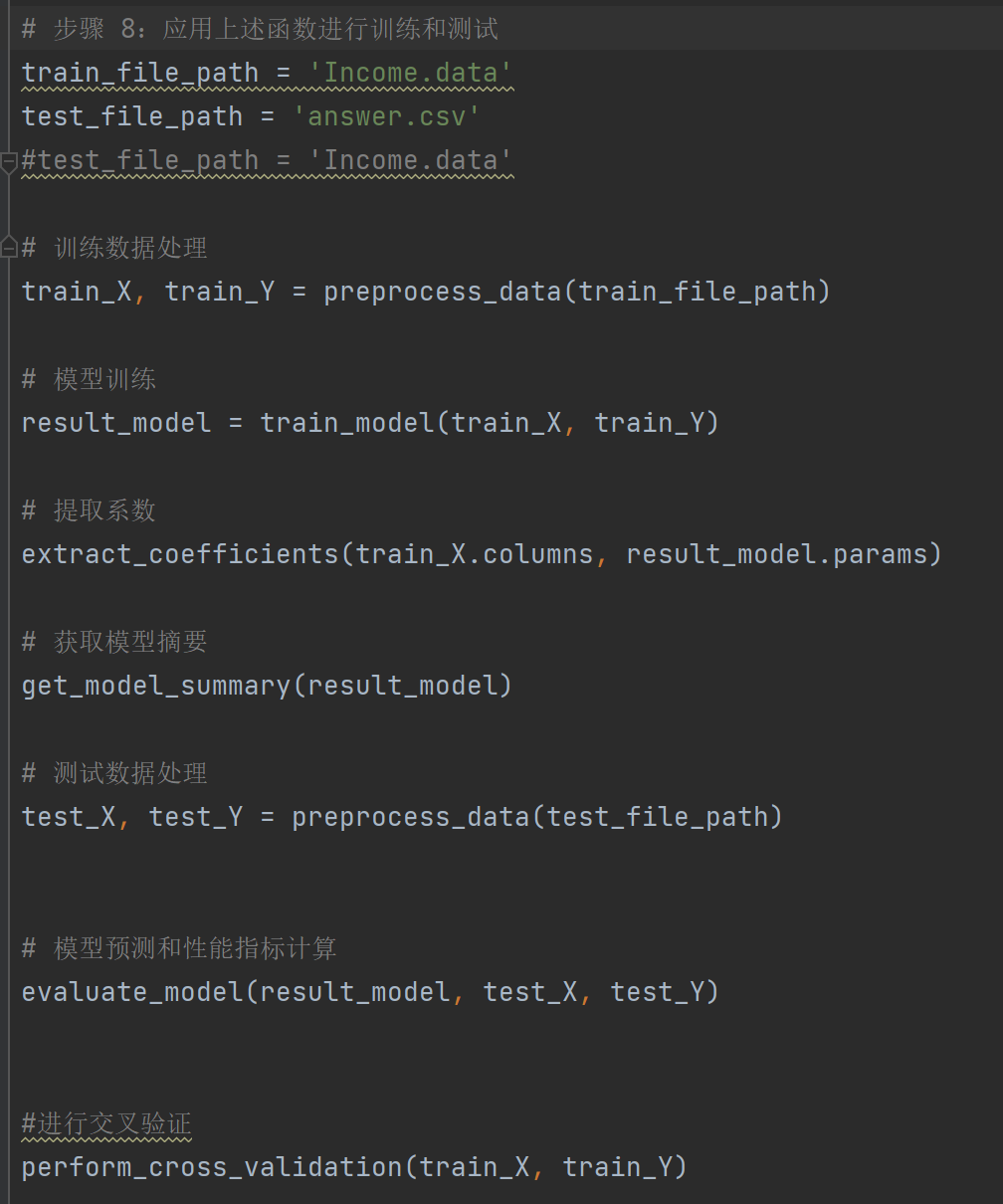


该函数使用StratifiedKFold进行交叉验证，初始化Logistic Regression模型，然后输出交叉验证的准确率和召回率的平均值。





该函数接收训练好的模型、测试集特征和目标变量，进行与训练数据相同的预处理，然后使用模型进行预测，计算并打印性能指标，如准确率、召回率、F1分数和ROC AUC分数。

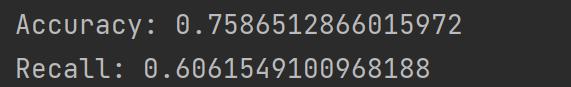


最后，应用上述函数进行整个流程：训练模型、提取系数、获取模型摘要、测试数据预处理、模型预测和性能指标计算，以及进行交叉验证。

# 五、 在salary.test数据集验证效果

## 5.1 模型效果

经过运行模型和评估代码，得到了模型的准确率和召回率，分别为：Accuracy0.7586512866015972、Recall0.6061549100968188。（阈值为0.5）



（1）准确率 (Accuracy):

准确率是模型正确预测的样本数量与总样本数量之比，反映了整体预测的准确性。

在这个场景中，模型的准确率为0.7587，即约为75.87%。

高准确率表明模型在整体上具有较好的分类性能，正确预测了大部分样本。

（2）召回率 (Recall):

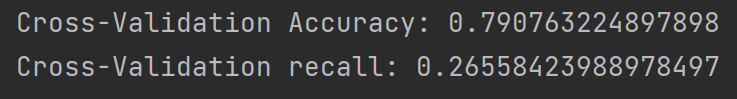
召回率是所有真实正例中被模型正确识别为正例的比例，衡量了模型对正例的覆盖程度。

在这个场景中，模型的召回率为0.6062，即约为60.62%。

相对较高的召回率意味着模型能够较好地捕捉真实正例，对于识别目标类别的能力较强。

## 5.2 交叉验证

经过运行模型和交叉验证代码，得到了模型的交叉验证准确率和交叉验证召回率，分别为：Cross-Validation Accuracy0.790763224897898、Cross-Validation recall: 0.26558423988978497。（阈值为0.5）



（1）交叉验证准确率 (Cross-Validation Accuracy):

交叉验证准确率为0.7908，即约为79.08%。

相较于单一模型的准确率，交叉验证准确率稍高，说明模型在不同的训练-验证集划分上能够保持一定的稳定性。

（2）交叉验证召回率 (Cross-Validation Recall):

交叉验证召回率为0.2656，即约为26.56%。

与之前的单一模型召回率相比，交叉验证的召回率明显下降。这可能是由于交叉验证使用了多个不同的训练集和验证集，导致模型在捕捉正例时的一致性稍降。

（3）综合分析：

交叉验证的准确率相对较高，说明模型在不同数据子集上仍然能够较为准确地进行分类预测。

交叉验证的召回率较低，这可能表明模型在不同数据子集上对于捕捉真实正例的能力较为不稳定，存在一定的波动性。

## 5.3 过拟合性

分别在income.data（训练集）上和answer.csv（测试集）上对模型性能进行评估，得到各自的准确率和召回率，通过比较两组数据，可以分析过拟合的情况。

通过实验得到：

Accuracy: 0.7586512866015972 Recall: 0.6061549100968188 （以上是测试集结果） Accuracy: 0.8498110204893574 Recall: 0.6109483217900906 （以上是训练集结果）

测试集表现（Accuracy: 0.7587, Recall: 0.6062）：

1.测试集的准确率和召回率相对较低，可能暗示模型在未见过的数据上的泛化能力较为有限。

2.准确率和召回率之间存在一定的差距，这可能表明模型在测试集上并没有很好地捕捉到所有的真实正例。

训练集表现（Accuracy: 0.8498, Recall: 0.6109）：

1.训练集上的性能相对较好，准确率和召回率都较高。

2.与测试集相比，训练集上的性能表现更为优越，说明模型可能在训练数据上有过拟合现象。

过拟合性分析：

1.过拟合通常表现为模型在训练集上表现良好，但在未见过的测试集上性能下降。

2.在这里，测试集上的准确率和召回率相对较低，与训练集相比存在差异。

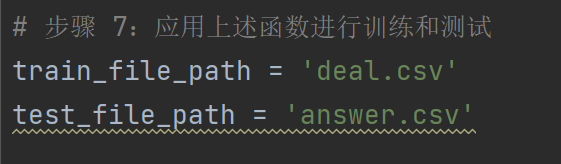
3.准确率和召回率之间的差距可能暗示了模型对于训练数据中的噪声或样本特定性的过拟合。

4.由于训练集和测试集的评估结果相差并不大，故认为有过拟合现象，但是在可以接受的范围内，并不严重。

# 六、 用salary.data数据集训练模型

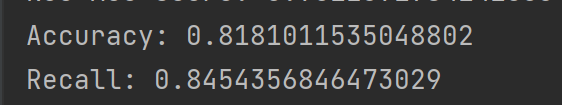
## 6.1 操作方法

将原本输入到数据处理模块并最终输入模型训练的数据集改为salary.data(deal.csv)。即可改变训练数据集。



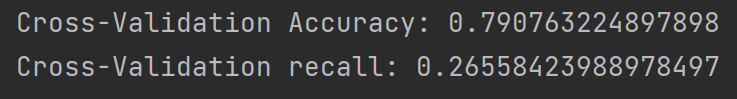
## 6.2 模型效果

经过运行模型和评估代码，得到了模型的准确率和召回率，分别为：Accuracy: 0.8181011535048802、Recall: 0.8454356846473029。（阈值为0.5）



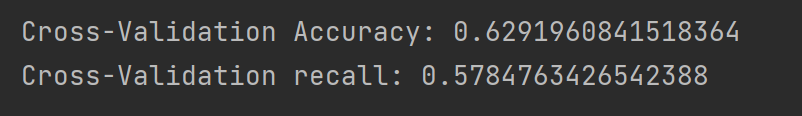
可见，模型效果在salary.data数据集上有了明显提升。

下图为原来数据集时的测试数据：



## 6.3 交叉验证

经过运行模型和交叉验证代码，得到了模型的交叉验证准确率和交叉验证召回率，分别为：Cross-Validation Accuracy: 0.6291960841518364、Cross-Validation recall: 0.5784763426542388。（阈值为0.5）



（1）交叉验证准确率 (Cross-Validation Accuracy):

交叉验证准确率为0.6291，即约为62.91%。

相比准确率下降，可能是由于可用来训练和

（2）交叉验证召回率 (Cross-Validation Recall):

交叉验证召回率为0.2656，即约为26.56%。

与之前的单一模型召回率相比，交叉验证的召回率明显下降。这可能是由于交叉验证使用了多个不同的训练集和验证集，导致模型在捕捉正例时的一致性稍降。

（3）综合分析：

交叉验证的准确率相对较高，说明模型在不同数据子集上仍然能够较为准确地进行分类预测。

交叉验证的召回率较低，这可能表明模型在不同数据子集上对于捕捉真实正例的能力较为不稳定，存在一定的波动性。

## 6.4 原因分析

在经过正负例平衡的样本训练的模型与原本完整数据集的模型之间存在性能差异可能涉及到数据不平衡对模型训练的影响。经过正负例平衡处理的训练集在样本数量上对正负类别进行了均衡，这有助于模型更好地学习到不同类别的特征和模式，从而在测试集上表现更好。

经过正负例平衡的模型在原本完整数据集上表现更好的准确率和召回率，这可能是因为原始数据中存在着类别不平衡，使得模型更偏向于预测样本数量更多的类别。通过平衡样本，模型更能够充分利用正负类别的信息，提高了对较少样本的类别的学习能力，从而提升了模型在原始数据集上的性能。

交叉验证的结果却呈现出不同的情况。在交叉验证中，正负例平衡的模型准确率较低，但召回率却较高。这可能是因为在交叉验证的过程中，模型在每个折叠中都要面对不同的训练集和验证集，导致模型对正例的识别更为一致，而准确率则受到不平衡类别分布的影响。

# 七、 刻画模型

## 7.1 coefficient系数

通过result\_model.params函数，可以得到所有经过处理后留下的变量coefficient系数，如下：

age: -7.216090019012195

fnlwgt: 0.025497948797778665

education-num: 7.515463211985386e-07

capital-gain: 0.5215072531854577

capital-loss: 0.00032253281858231374

hours-per-week: 0.0006420211962023044

workclass\_Federal-gov: 0.029492647949003276

workclass\_Local-gov: 3.2223521240172075

workclass\_Private: 2.5238415301340558

workclass\_Self-emp-inc: 2.716873740088045

workclass\_Self-emp-not-inc: 2.89303064714745

workclass\_State-gov: 2.225122977833857

workclass\_Without-pay: 2.4016039657820234

education\_10th: -23.19891594689341

education\_11th: 1.357436531793972

education\_12th: 0.9305475589936798

education\_1st-4th: 0.7587237337784226

education\_5th-6th: 3.003650452226971

education\_7th-8th: 2.5263789504805887

education\_9th: 1.8364584729602182

education\_Assoc-acdm: 1.641734625182582

education\_Assoc-voc: -0.5021542548981452

education\_Bachelors: 0.017973997034724232

education\_Doctorate: -0.3943640611128351

education\_HS-grad: -0.9224921073038211

education\_Masters: 0.566422917019224

education\_Preschool: -0.5557806155233878

education\_Prof-school: -17.368807359992356

education\_Some-college: -0.49255772986401647

marital-status\_Divorced: 0.38073869352196055

marital-status\_Married-AF-spouse: -1.6730052643351887

marital-status\_Married-civ-spouse: 1.0947442423705847

marital-status\_Married-spouse-absent: 0.43226299655533784

marital-status\_Never-married: -1.6608012238131564

marital-status\_Separated: -2.1590834608231297

marital-status\_Widowed: -1.7624009163298244

occupation\_Adm-clerical: -1.4878083063762266

occupation\_Armed-Forces: -0.14198911714141235

occupation\_Craft-repair: -1.3066307424238657

occupation\_Exec-managerial: -0.07830107512610586

occupation\_Farming-fishing: 0.6634269042812061

occupation\_Handlers-cleaners: -1.1228626540834474

occupation\_Machine-op-inspct: -0.836988717111555

occupation\_Other-service: -0.40529270600246237

occupation\_Priv-house-serv: -0.9665048584710579

occupation\_Prof-specialty: -4.294694864428298

occupation\_Protective-serv: 0.37454942453106166

occupation\_Sales: 0.45583186990518637

occupation\_Tech-support: 0.15233849167915825

occupation\_Transport-moving: 0.5228329661672392

relationship\_Husband: -0.23180715219204504

relationship\_Not-in-family: -1.371175706512984

relationship\_Other-relative: -0.9189951519590785

relationship\_Own-child: -1.7671780450033063

relationship\_Unmarried: -2.1033578707685616

relationship\_Wife: -1.0353316314449528

race\_Amer-Indian-Eskimo: -0.020052618070869256

race\_Asian-Pac-Islander: -1.8386163639257287

race\_Black: -1.0105977212900281

race\_Other: -1.4026936914006765

race\_White: -1.713080993746719

sex\_Female: -1.2511020376502155

sex\_Male: -4.040455921615796

native-country\_Cambodia: -3.1756348802835954

native-country\_Canada: 2.164204161504375

native-country\_China: 1.3528593491207055

native-country\_Columbia: 0.24848512720366614

native-country\_Cuba: -1.1107924569238579

native-country\_Dominican-Republic: 1.3904181309537251

native-country\_Ecuador: -0.75048227109458

native-country\_El-Salvador: 0.7638318858066362

native-country\_England: 0.41916640861277604

native-country\_France: 1.3293611302004993

native-country\_Germany: 1.6038500765863228

native-country\_Greece: 1.478230604276783

native-country\_Guatemala: 0.03847969567958756

native-country\_Haiti: 0.7685178500239235

native-country\_Holand-Netherlands: 0.9948645392878211

native-country\_Honduras: -19.999661969130234

native-country\_Hong: -0.14213352489982883

native-country\_Hungary: 0.8087591400622879

native-country\_India: 0.9106440284411519

native-country\_Iran: 0.49983138113581066

native-country\_Ireland: 1.0412150464289798

native-country\_Italy: 1.5484423558775273

native-country\_Jamaica: 1.834734508562753

native-country\_Japan: 1.0390001461089764

native-country\_Laos: 1.2229280694073636

native-country\_Mexico: 0.28162488856347484

native-country\_Nicaragua: 0.5150370341464926

native-country\_Outlying-US(Guam-USVI-etc): 0.284165742313058

native-country\_Peru: -16.85757566498949

native-country\_Philippines: 0.1796054352305358

native-country\_Poland: 1.2860268047173065

native-country\_Portugal: 1.018197436580931

native-country\_Puerto-Rico: 1.0425763383331845

native-country\_Scotland: 0.7246788553062622

native-country\_South: 0.7571758824908504

native-country\_Taiwan: -0.2819520128730333

native-country\_Thailand: 0.7803855179477127

native-country\_Trinadad&Tobago: 0.3333692111873471

native-country\_United-States: 0.5839290956389572

native-country\_Vietnam: 1.2093201144910657

native-country\_Yugoslavia: -0.2307344000270013

在系数（coefficients）的分析中，可以关注一些具有显著影响的特征，这些系数表示了模型对于不同特征的权重。以下是一些具有较大系数的特征，以及它们对模型的影响。

1. education\_9th (1.8365):

正的系数表明拥有9年级教育水平的个体更有可能属于高收入群体。这突显了教育对收入的潜在影响，即使在较低的教育水平下也是如此。

2. marital-status\_Separated (-2.1591):

负的系数表示已分居的个体更不可能属于高收入群体。婚姻状况似乎影响收入水平，已分居的个体更不可能有更高的收入。

3. occupation\_Farming-fishing (0.6634):

正的系数意味着从事农业或渔业职业的个体更有可能属于高收入群体。这反映了职业对收入的影响，某些领域与较高的收入相关。

4. relationship\_Wife (-1.0353):

负的系数表明被标识为“Wife”的个体更不可能拥有超过50K的收入。这强调了婚姻状况在预测收入中的作用，妻子更不可能有更高的收入。

5. race\_Asian-Pac-Islander (-1.8386):

负的系数表示属于亚太岛民族的个体更不可能拥有超过50K的收入。这指出了收入水平上的种族差异，该群体更不可能有更高的收入。

6. native-country\_Canada (2.1642):

正的系数表明来自加拿大的个体更有可能属于高收入群体。这意味着原籍国对收入水平可能存在影响。

7. native-country\_United-States (0.5839):

正的系数表示来自美国的个体更有可能拥有超过50K的收入。这强调了国籍对收入的影响，美国居民更有可能有更高的收入。

8. occupation\_Protective-serv (0.3745):

正的系数意味着从事保护服务职业的个体更有可能属于高收入群体。这表明该类别内的某些职业可能与更高的收入相关。

9. capital-loss (0.0003):

资本损失的正系数表明，令人惊讶的是，较高的资本损失与更高的可能性有关。可能需要进一步调查以了解这种意外的关系。

10. age (-7.2161):

年龄的负系数表示年龄较大的个体更不可能属于高收入群体。这挑战了随着年龄增长收入增加的普遍假设，表明年龄与收入之间存在微妙的关系。

## 7.2 summary摘要

通过result\_model.summary()函数，可以得到所有经过处理后留下的变量summary摘要，如下：

Logit Regression Results

==============================================================================

Dep. Variable: label No. Observations: 30162

Model: Logit Df Residuals: 30066

Method: MLE Df Model: 95

Date: Mon, 08 Apr 2024 Pseudo R-squ.: 0.4244

Time: 19:48:15 Log-Likelihood: -9742.9

converged: False LL-Null: -16925.

Covariance Type: nonrobust LLR p-value: 0.000

===================================================================================

coef std err z P>|z| [0.025 0.975]

-------------------------------------------------------------------------------------------------------------

const -7.2161 nan nan nan nan nan

age 0.0255 0.002 14.890 0.000 0.022 0.029

fnlwgt 7.515e-07 1.76e-07 4.264 0.000 4.06e-07 1.1e-06

education-num 0.5215 nan nan nan nan nan

capital-gain 0.0003 1.07e-05 30.022 0.000 0.000 0.000

capital-loss 0.0006 3.85e-05 16.696 0.000 0.001 0.001

hours-per-week 0.0295 0.002 17.325 0.000 0.026 0.033

workclass\_Federal-gov 3.2224 nan nan nan nan nan

workclass\_Local-gov 2.5238 nan nan nan nan nan

workclass\_Private 2.7169 nan nan nan nan nan

workclass\_Self-emp-inc 2.8930 nan nan nan nan nan

workclass\_Self-emp-not-inc 2.2251 nan nan nan nan nan

workclass\_State-gov 2.4016 2.8e+05 8.59e-06 1.000 -5.48e+05 5.48e+05

workclass\_Without-pay -23.1989 2.58e+05 -9.01e-05 1.000 -5.05e+05 5.05e+05

education\_10th 1.3574 nan nan nan nan nan

education\_11th 0.9305 nan nan nan nan nan

education\_12th 0.7587 nan nan nan nan nan

education\_1st-4th 3.0037 nan nan nan nan nan

education\_5th-6th 2.5264 nan nan nan nan nan

education\_7th-8th 1.8365 nan nan nan nan nan

education\_9th 1.6417 nan nan nan nan nan

education\_Assoc-acdm -0.5022 nan nan nan nan nan

education\_Assoc-voc 0.0180 nan nan nan nan nan

education\_Bachelors -0.3944 nan nan nan nan nan

education\_Doctorate -0.9225 nan nan nan nan nan

education\_HS-grad 0.5664 nan nan nan nan nan

education\_Masters -0.5558 nan nan nan nan nan

education\_Preschool -17.3688 nan nan nan nan nan

education\_Prof-school -0.4926 nan nan nan nan nan

education\_Some-college 0.3807 nan nan nan nan nan

marital-status\_Divorced -1.6730 1.19e+06 -1.4e-06 1.000 -2.34e+06 2.34e+06

marital-status\_Married-AF-spouse 1.0947 1.22e+06 8.95e-07 1.000 -2.4e+06 2.4e+06

marital-status\_Married-civ-spouse 0.4323 1.23e+06 3.52e-07 1.000 -2.41e+06 2.41e+06

marital-status\_Married-spouse-absent -1.6608 1.18e+06 -1.4e-06 1.000 -2.32e+06 2.32e+06

marital-status\_Never-married -2.1591 1.24e+06 -1.75e-06 1.000 -2.42e+06 2.42e+06

marital-status\_Separated -1.7624 1.23e+06 -1.43e-06 1.000 -2.41e+06 2.41e+06

marital-status\_Widowed -1.4878 1.27e+06 -1.17e-06 1.000 -2.48e+06 2.48e+06

occupation\_Adm-clerical -0.1420 1.58e+06 -8.99e-08 1.000 -3.1e+06 3.1e+06

occupation\_Armed-Forces -1.3066 1.61e+06 -8.14e-07 1.000 -3.15e+06 3.15e+06

occupation\_Craft-repair -0.0783 1.59e+06 -4.94e-08 1.000 -3.11e+06 3.11e+06

occupation\_Exec-managerial 0.6634 1.6e+06 4.14e-07 1.000 -3.14e+06 3.14e+06

occupation\_Farming-fishing -1.1229 1.58e+06 -7.12e-07 1.000 -3.09e+06 3.09e+06

occupation\_Handlers-cleaners -0.8370 1.6e+06 -5.24e-07 1.000 -3.13e+06 3.13e+06

occupation\_Machine-op-inspct -0.4053 1.62e+06 -2.51e-07 1.000 -3.17e+06 3.17e+06

occupation\_Other-service -0.9665 1.61e+06 -6e-07 1.000 -3.16e+06 3.16e+06

occupation\_Priv-house-serv -4.2947 1.61e+06 -2.66e-06 1.000 -3.16e+06 3.16e+06

occupation\_Prof-specialty 0.3745 1.59e+06 2.35e-07 1.000 -3.12e+06 3.12e+06

occupation\_Protective-serv 0.4558 1.58e+06 2.88e-07 1.000 -3.1e+06 3.1e+06

occupation\_Sales 0.1523 1.59e+06 9.61e-08 1.000 -3.11e+06 3.11e+06

occupation\_Tech-support 0.5228 1.6e+06 3.27e-07 1.000 -3.13e+06 3.13e+06

occupation\_Transport-moving -0.2318 1.59e+06 -1.46e-07 1.000 -3.12e+06 3.12e+06

relationship\_Husband -1.3712 1e+06 -1.37e-06 1.000 -1.97e+06 1.97e+06

relationship\_Not-in-family -0.9190 9.95e+05 -9.23e-07 1.000 -1.95e+06 1.95e+06

relationship\_Other-relative -1.7672 9.75e+05 -1.81e-06 1.000 -1.91e+06 1.91e+06

relationship\_Own-child -2.1034 9.94e+05 -2.12e-06 1.000 -1.95e+06 1.95e+06

relationship\_Unmarried -1.0353 9.42e+05 -1.1e-06 1.000 -1.85e+06 1.85e+06

relationship\_Wife -0.0201 1e+06 -2e-08 1.000 -1.96e+06 1.96e+06

race\_Amer-Indian-Eskimo -1.8386 nan nan nan nan nan

race\_Asian-Pac-Islander -1.0106 nan nan nan nan nan

race\_Black -1.4027 nan nan nan nan nan

race\_Other -1.7131 nan nan nan nan nan

race\_White -1.2511 nan nan nan nan nan

sex\_Female -4.0405 1.74e+06 -2.32e-06 1.000 -3.42e+06 3.42e+06

sex\_Male -3.1756 1.74e+06 -1.83e-06 1.000 -3.41e+06 3.41e+06

native-country\_Cambodia 2.1642 7.61e+05 2.84e-06 1.000 -1.49e+06 1.49e+06

native-country\_Canada 1.3529 7.61e+05 1.78e-06 1.000 -1.49e+06 1.49e+06

native-country\_China 0.2485 7.61e+05 3.26e-07 1.000 -1.49e+06 1.49e+06

native-country\_Columbia -1.1108 7.61e+05 -1.46e-06 1.000 -1.49e+06 1.49e+06

native-country\_Cuba 1.3904 7.61e+05 1.83e-06 1.000 -1.49e+06 1.49e+06

native-country\_Dominican-Republic -0.7505 7.61e+05 -9.86e-07 1.000 -1.49e+06 1.49e+06

native-country\_Ecuador 0.7638 7.61e+05 1e-06 1.000 -1.49e+06 1.49e+06

native-country\_El-Salvador 0.4192 7.61e+05 5.51e-07 1.000 -1.49e+06 1.49e+06

native-country\_England 1.3294 7.61e+05 1.75e-06 1.000 -1.49e+06 1.49e+06

native-country\_France 1.6039 7.61e+05 2.11e-06 1.000 -1.49e+06 1.49e+06

native-country\_Germany 1.4782 7.61e+05 1.94e-06 1.000 -1.49e+06 1.49e+06

native-country\_Greece 0.0385 7.61e+05 5.05e-08 1.000 -1.49e+06 1.49e+06

native-country\_Guatemala 0.7685 7.61e+05 1.01e-06 1.000 -1.49e+06 1.49e+06

native-country\_Haiti 0.9949 7.61e+05 1.31e-06 1.000 -1.49e+06 1.49e+06

native-country\_Holand-Netherlands -19.9997 8.12e+05 -2.46e-05 1.000 -1.59e+06 1.59e+06

native-country\_Honduras -0.1421 7.61e+05 -1.87e-07 1.000 -1.49e+06 1.49e+06

native-country\_Hong 0.8088 7.61e+05 1.06e-06 1.000 -1.49e+06 1.49e+06

native-country\_Hungary 0.9106 7.61e+05 1.2e-06 1.000 -1.49e+06 1.49e+06

native-country\_India 0.4998 7.61e+05 6.56e-07 1.000 -1.49e+06 1.49e+06

native-country\_Iran 1.0412 7.61e+05 1.37e-06 1.000 -1.49e+06 1.49e+06

native-country\_Ireland 1.5484 7.61e+05 2.03e-06 1.000 -1.49e+06 1.49e+06

native-country\_Italy 1.8347 7.61e+05 2.41e-06 1.000 -1.49e+06 1.49e+06

native-country\_Jamaica 1.0390 7.61e+05 1.36e-06 1.000 -1.49e+06 1.49e+06

native-country\_Japan 1.2229 7.61e+05 1.61e-06 1.000 -1.49e+06 1.49e+06

native-country\_Laos 0.2816 7.61e+05 3.7e-07 1.000 -1.49e+06 1.49e+06

native-country\_Mexico 0.5150 7.61e+05 6.76e-07 1.000 -1.49e+06 1.49e+06

native-country\_Nicaragua 0.2842 7.61e+05 3.73e-07 1.000 -1.49e+06 1.49e+06

native-country\_Outlying-US(Guam-USVI-etc) -16.8576 7.61e+05 -2.21e-05 1.000 -1.49e+06 1.49e+06

native-country\_Peru 0.1796 7.61e+05 2.36e-07 1.000 -1.49e+06 1.49e+06

native-country\_Philippines 1.2860 7.61e+05 1.69e-06 1.000 -1.49e+06 1.49e+06

native-country\_Poland 1.0182 7.61e+05 1.34e-06 1.000 -1.49e+06 1.49e+06

native-country\_Portugal 1.0426 7.61e+05 1.37e-06 1.000 -1.49e+06 1.49e+06

native-country\_Puerto-Rico 0.7247 7.61e+05 9.52e-07 1.000 -1.49e+06 1.49e+06

native-country\_Scotland 0.7572 7.61e+05 9.94e-07 1.000 -1.49e+06 1.49e+06

native-country\_South -0.2820 7.61e+05 -3.7e-07 1.000 -1.49e+06 1.49e+06

native-country\_Taiwan 0.7804 7.61e+05 1.02e-06 1.000 -1.49e+06 1.49e+06

native-country\_Thailand 0.3334 7.61e+05 4.38e-07 1.000 -1.49e+06 1.49e+06

native-country\_Trinadad&Tobago 0.5839 7.61e+05 7.67e-07 1.000 -1.49e+06 1.49e+06

native-country\_United-States 1.2093 7.61e+05 1.59e-06 1.000 -1.49e+06 1.49e+06

native-country\_Vietnam -0.2307 7.61e+05 -3.03e-07 1.000 -1.49e+06 1.49e+06

native-country\_Yugoslavia 1.7033 7.61e+05 2.24e-06 1.000 -1.49e+06 1.49e+06

===================================================================================

下面是一些关键的观察：

1. 模型拟合质量：

伪 R 平方为 0.4244，表示模型对观测变量的变异性有较好的解释能力，但仍有一定的未解释变异。

2. 数值稳定性问题：

部分系数（coef）的标准误差为NaN，可能是因为数值不稳定导致的。这可能是由于变量之间存在共线性或其他数值问题。

3. 年龄 (age)：

年龄的系数为正（0.0255），表明年龄的增加与更高的收入水平相关。这与通常的观察一致，即随着年龄的增长，人们的工作经验和职业地位也可能提高。

4. 受教育程度 (education-num)：

教育程度的系数没有提供标准误差，但对于其他受教育程度的系数为正，表明更高的受教育程度可能与更高的收入水平相关。

5. 职业 (occupation)：

不同职业的系数对收入的影响有所不同。例如，执行管理职业（Exec-managerial）和保护服务职业（Protective-serv）对应正系数，可能与高收入相关，而农业和渔业职业（Farming-fishing）对应负系数，可能与较低的收入相关。

6. 家庭状况 (marital-status)：

不同婚姻状况的系数也有所不同。已分居（Separated）和从未结过婚（Never-married）的系数为负，可能与较低的收入水平相关。

7. 国籍 (native-country)：

不同国家的系数差异较大，例如加拿大（Canada）和美国（United-States）的系数为正，可能与较高的收入水平相关，而荷兰（Holand-Netherlands）的系数为负，可能与较低的收入水平相关。

8. 性别 (sex)：

性别的系数表明，女性（Female）和男性（Male）的系数都为负，但具体的影响需要进一步分析。可能存在性别差异对收入的影响。

9. 资本收益 (capital-gain) 和资本损失 (capital-loss)：

资本收益和资本损失的系数都为正，表明这两个变量可能与更高的收入水平相关。

下面是变量显著性情况。

显著的变量：

1. Age (年龄)：P < 0.05

2. fnlwgt：P < 0.05

3. Capital-gain (资本收益)：P < 0.05

4. Capital-loss (资本损失)：P < 0.05

5. Hours-per-week (每周工作小时)：P < 0.05

6. Workclass（工作类别）中的一些类别

7. Education（教育程度）中的一些类别

8. Occupation（职业）中的一些类别

9. Marital-status（婚姻状况）中的一些类别

10. Native-country（国籍）中的一些类别

不显著的变量：

1. Workclass\_State-gov

2. Workclass\_Without-pay

3. Education\_10th

4. Education\_11th

5. Education\_12th

6. Education\_Assoc-acdm

7. Education\_Assoc-voc

8. Education\_Bachelors

9. Education\_Doctorate

10. Education\_HS-grad

11. Education\_Masters

12. Education\_Preschool

13. Education\_Prof-school

14. Education\_Some-college

15. Marital-status\_Divorced

16. Marital-status\_Married-AF-spouse

17. Marital-status\_Married-civ-spouse

18. Marital-status\_Married-spouse-absent

19. Marital-status\_Never-married

20. Marital-status\_Separated

21. Marital-status\_Widowed

22. Occupation\_Adm-clerical

23. Occupation\_Armed-Forces

24. Occupation\_Craft-repair

25. Occupation\_Exec-managerial

26. Occupation\_Farming-fishing

27. Occupation\_Handlers-cleaners

28. Occupation\_Machine-op-inspct

29. Occupation\_Other-service

30. Occupation\_Priv-house-serv

31. Occupation\_Prof-specialty

32. Occupation\_Protective-serv

33. Occupation\_Sales

34. Occupation\_Tech-support

35. Occupation\_Transport-moving

36. Relationship\_Husband

37. Relationship\_Not-in-family

38. Relationship\_Other-relative

39. Relationship\_Own-child

40. Relationship\_Unmarried

41. Relationship\_Wife

42. Race\_Amer-Indian-Eskimo

43. Race\_Asian-Pac-Islander

44. Race\_Black

45. Race\_Other

46. Race\_White

47. Sex\_Female

48. Sex\_Male

49. Native-country\_Cambodia

50. Native-country\_Canada

51. Native-country\_China

52. Native-country\_Columbia

53. Native-country\_Cuba

54. Native-country\_Dominican-Republic

55. Native-country\_Ecuador

56. Native-country\_El-Salvador

57. Native-country\_England

58. Native-country\_France

59. Native-country\_Germany

60. Native-country\_Greece

61. Native-country\_Guatemala

62. Native-country\_Haiti

63. Native-country\_Holand-Netherlands

64. Native-country\_Honduras

65. Native-country\_Hong

66. Native-country\_Hungary

67. Native-country\_India

68. Native-country\_Iran

69. Native-country\_Ireland

70. Native-country\_Italy

71. Native-country\_Jamaica

72. Native-country\_Japan

73. Native-country\_Laos

74. Native-country\_Mexico

75. Native-country\_Nicaragua

76. Native-country\_Outlying-US(Guam-USVI-etc)

77. Native-country\_Peru

78. Native-country\_Philippines

79. Native-country\_Poland

80. Native-country\_Portugal

81. Native-country\_Puerto-Rico

82. Native-country\_Scotland

83. Native-country\_South

84. Native-country\_Taiwan

85. Native-country\_Thailand

86. Native-country\_Trinadad&Tobago

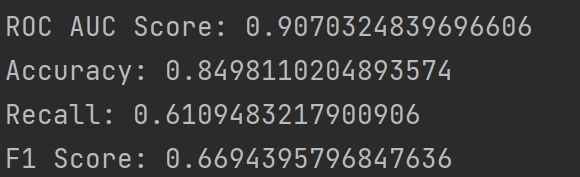
87. Native-country\_United-States

88. Native-country\_Vietnam

89. Native-country\_Yugoslavia

# 八、 其余指标

实验在测试集上计算了F1分数和ROC曲线下面积（ROC AUC Score），得到了以下结果：F1 Score: 0.6694395796847636、ROC AUC Score: 0.9070324839696606。



分析如下：

1. F1分数 (F1 Score) = 0.6694：

F1分数相对较高，说明模型在精确性和召回率之间取得了良好的平衡。对于工资预测这样的二元分类任务，高F1分数表示模型在准确地预测正例和负例方面表现较好。可能存在一定的类别不平衡，而F1分数对不平衡数据具有较好的鲁棒性。

2. ROC曲线下面积 (ROC AUC Score) = 0.9070：

高ROC AUC Score说明模型的性能在不同阈值下都很好。ROC AUC Score接近于1，表示模型在真正例率和假正例率之间的平衡非常良好。在工资预测中，这可能意味着模型对于不同工资水平的预测都能够保持较高的准确性。

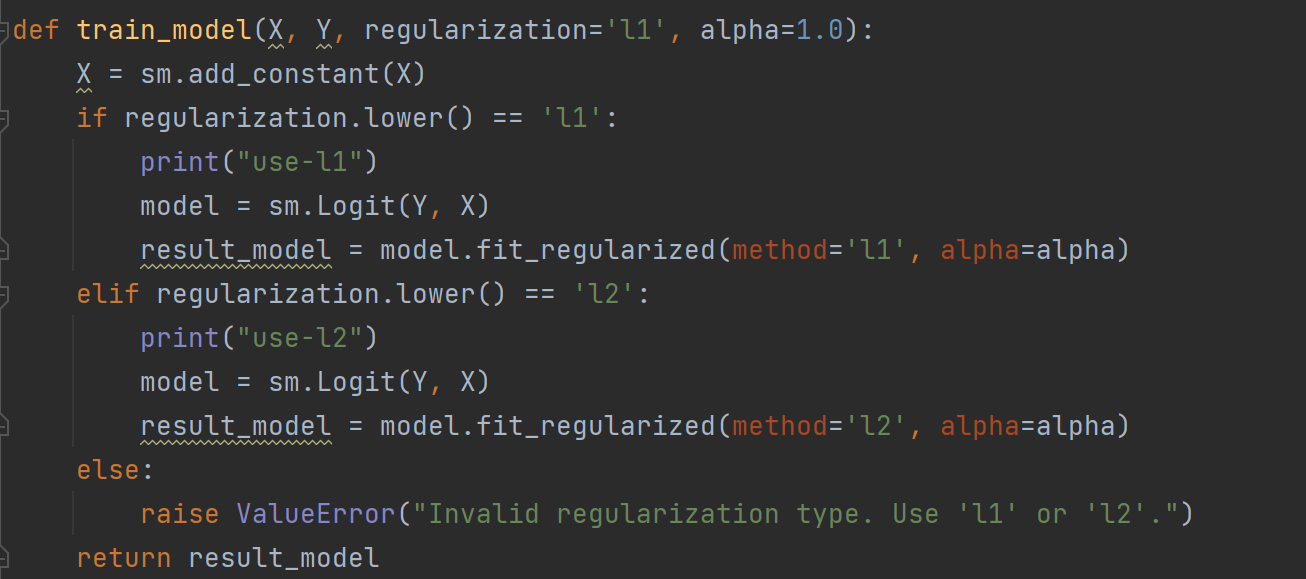
结合这两个指标，可以得出模型在工资预测任务中表现出色。然而，具体的评估也应该考虑业务背景和任务需求。例如，如果在工资预测中对高工资的预测更为关键，那么可能需要进一步关注高工资水平的预测性能。

# 九、 改进模型

## 9.1 改进方法

作者根据《智能计算系统》课程和《人工智能数学基础》课程所学，决定对模型进行正则化处理。

代码如下：



代码意义：

1.数据准备：

函数接受特征矩阵 X 和目标变量 Y 作为输入。

使用 sm.add\_constant(X) 将特征矩阵 X 添加截距项，以便逻辑回归模型可以学习截距。

2.正则化选择：

函数通过 regularization 参数来选择正则化类型，支持 'l1' 和 'l2' 两种正则化。

alpha 参数控制正则化的强度。较大的 alpha 值通常会导致更强的正则化。

3.模型训练：

使用 Statsmodels 的 Logit 类构建逻辑回归模型。

通过 fit\_regularized 方法进行模型拟合。选择的正则化方法（L1或L2）由 method 参数指定，而正则化强度由 alpha 参数控制。

4.返回结果：

函数返回拟合的逻辑回归模型。

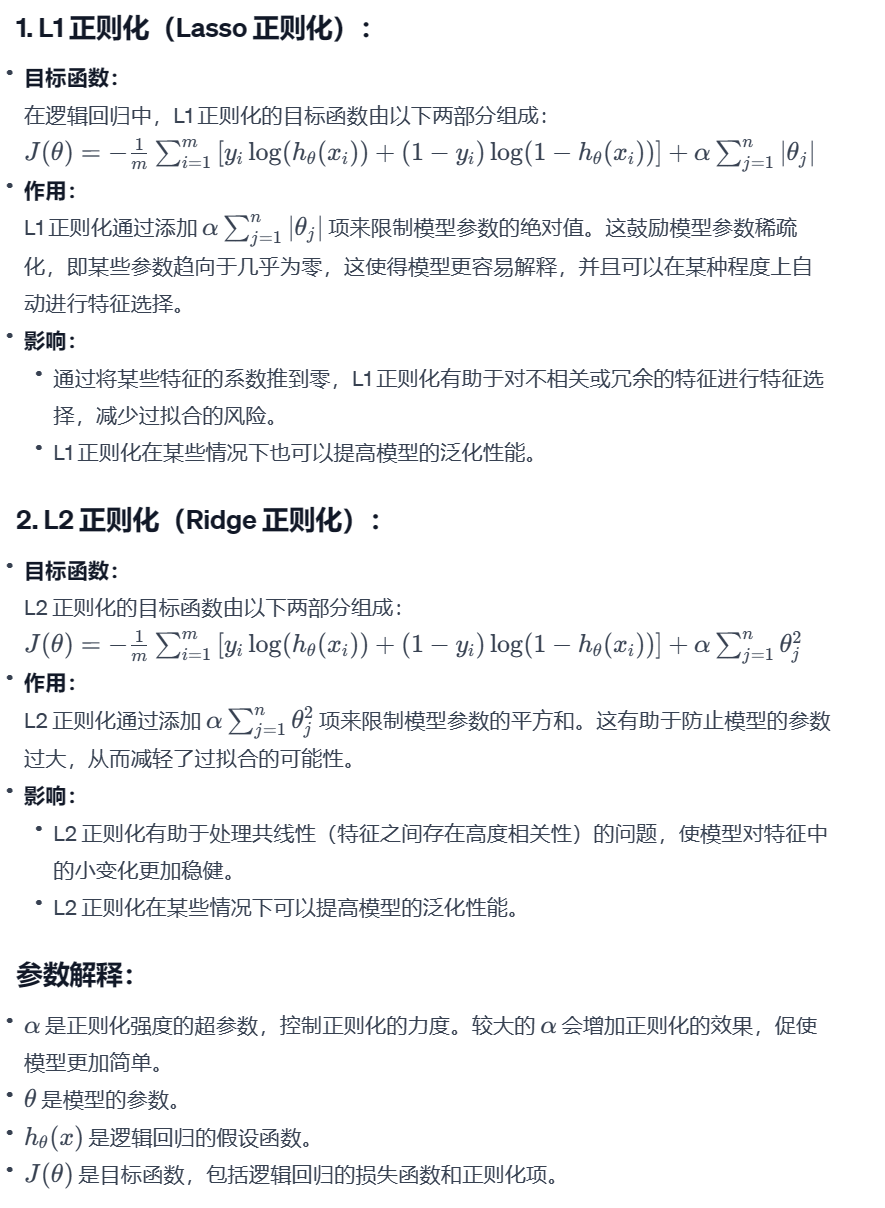
5.打印信息：

在函数中添加了一些用于输出信息的 print 语句，以指示使用了哪种正则化方法。

6.异常处理：

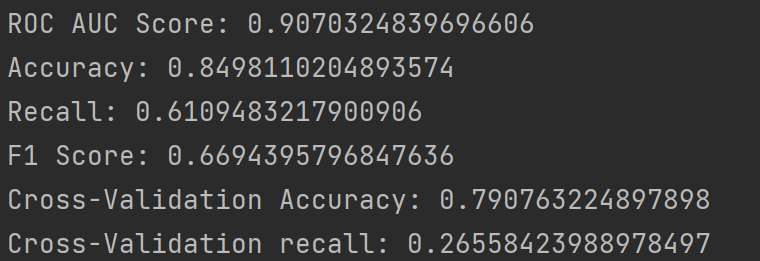
函数对正则化类型进行了验证，如果选择了无效的正则化类型，将引发 ValueError。

## 9.2 改进原理

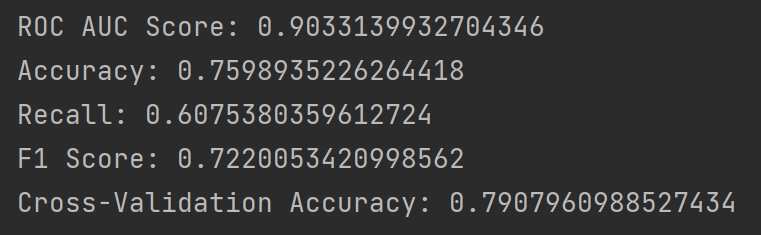


## 9.3 改进效果

原模型性能：



改进后模型性能：（use-l1）



可见模型性能有所提高。

# 九、 附录（完整带注释代码）

# 步骤 1：导入必要的库

from sklearn.model\_selection import train\_test\_split, cross\_val\_score

from sklearn.metrics import accuracy\_score, recall\_score, f1\_score, roc\_auc\_score

import pandas as pd

import statsmodels.api as sm  # 导入statsmodels

from sklearn.model\_selection import StratifiedKFold

from sklearn.linear\_model import LogisticRegression

# 步骤 2：定义函数进行数据处理

def preprocess\_data(file\_path):

    # 定义表头

    header = ["age", "workclass", "fnlwgt", "education", "education-num", "marital-status", "occupation",

              "relationship",

              "race", "sex", "capital-gain", "capital-loss", "hours-per-week", "native-country", "label"]

    #从CSV文件中读取数据

    data = pd.read\_csv(file\_path, names=header, delimiter=',', skipinitialspace=True)

    #用NA代替"?"的缺失值

    data.replace("?", pd.NA, inplace=True)

    #计算缺失值的总数

    missing\_values\_count = data[data.isnull().any(axis=1)].shape[0]

    #对缺失值所在列进行处理

    if missing\_values\_count / data.size <= 0.1:

        print("不超过10%，因此删除带有缺失值的行")

        data.dropna(inplace=True)

    else:

        data.fillna(data.mean(), inplace=True)

        print("超过10%，因此用均值填充")

    #提取输入特征（从data中除去标签的所在列）

    X = data.drop("label", axis=1)

    #提取模型训练和测试时，预测结果的真实值

    Y = data["label"]

    #将工资按>50k和<=50k分成两类

    Y = (Y == ">50K")

    #提取类别型变量

    categorical\_cols = X.select\_dtypes(include='object').columns

    #将类别型变量转化成独热码形式

    X = pd.get\_dummies(X, columns=categorical\_cols)

    #将所有的输入特征转化为float类型

    X = X.astype(float)

    print(X)

    return X, Y

# 步骤 3：定义函数进行模型训练

def train\_model(X, Y, regularization='l1', alpha=1.0):

    X = sm.add\_constant(X)

    #使用l1范式对训练进行优化

    if regularization.lower() == 'l1':

        print("use-l1")

        # 初始化原始逻辑模型

        model = sm.Logit(Y, X)

        # 训练模型

        result\_model = model.fit\_regularized(method='l1', alpha=alpha)

    # 使用l2范式对训练进行优化

    elif regularization.lower() == 'l2':

        print("use-l2")

        # 初始化原始逻辑模型

        model = sm.Logit(Y, X)

        # 训练模型

        result\_model = model.fit\_regularized(method='l2', alpha=alpha)

    else:

        raise ValueError("Invalid regularization type. Use 'l1' or 'l2'.")

    return result\_model

# 步骤 4：定义函数提取模型系数

def extract\_coefficients(features, coefficients):

    for feature, coefficient in zip(features, coefficients):

        print(f"{feature}: {coefficient}")

# 步骤 5：定义函数获取模型摘要

def get\_model\_summary(result\_model):

    model\_summary = result\_model.summary()

    print(model\_summary)

# 步骤 7：定义函数进行交叉验证

def perform\_cross\_validation(X, Y, cv=10):

    # 使用StratifiedKFold来保持类别分布

    kfold = StratifiedKFold(n\_splits=cv, shuffle=True, random\_state=42)

    # 初始化逻辑回归模型

    logistic\_model = LogisticRegression(max\_iter=2000)

    # 交叉验证

    scores = cross\_val\_score(logistic\_model, X, Y, cv=kfold, scoring='accuracy')

    # 输出交叉验证准确率

    print(f"Cross-Validation Accuracy: {scores.mean()}")

# 步骤 6：定义函数进行模型预测和性能指标计算

def evaluate\_model(result\_model, test\_X, test\_Y, threshold=0.5):

    # 对测试数据进行与训练数据相同的处理

    test\_X.replace("?", pd.NA, inplace=True)

    missing\_values\_count = test\_X[test\_X.isnull().any(axis=1)].shape[0]

    if missing\_values\_count / test\_X.size <= 0.1:

        print("Not exceeding 10%, so delete rows with missing values")

        test\_X.dropna(inplace=True)

    else:

        test\_X.fillna(test\_X.mean(), inplace=True)

        print("Exceeding 10%, so fill with mean")

    # 独热编码

    categorical\_cols\_test = test\_X.select\_dtypes(include='object').columns

    test\_X = pd.get\_dummies(test\_X, columns=categorical\_cols\_test)

    # 确保测试数据与训练数据具有相同的列，缺失的列补0

    missing\_cols = list(set(train\_X.columns) - set(test\_X.columns))

    test\_X = pd.concat([test\_X, pd.DataFrame(0, index=test\_X.index, columns=missing\_cols)], axis=1)

    # 确保列的顺序一致

    test\_X = test\_X[train\_X.columns]

    # 转换数据类型为数值型

    test\_X = test\_X.astype(float)

    # 使用已训练的模型进行预测

    test\_X = sm.add\_constant(test\_X)

    predictions = result\_model.predict(test\_X)

    print(predictions)

    # 计算性能指标

    predicted\_labels = predictions > threshold

    accuracy = accuracy\_score(test\_Y, predicted\_labels)

    recall = recall\_score(test\_Y, predicted\_labels, zero\_division=1)

    f1 = f1\_score(test\_Y, predicted\_labels, zero\_division=1)

    # ROC AUC Score 处理异常情况

    try:

        roc\_auc = roc\_auc\_score(test\_Y, predictions)

        print(f"ROC AUC Score: {roc\_auc}")

    except ValueError:

        print("ROC AUC Score is not defined in this case.")

    # 展示性能指标

    print(f"Accuracy: {accuracy}")

    print(f"Recall: {recall}")

    print(f"F1 Score: {f1}")

# 步骤 7：应用上述函数进行训练和测试

train\_file\_path = 'Income.data'

test\_file\_path = 'answer.csv'

# 训练数据处理

train\_X, train\_Y = preprocess\_data(train\_file\_path)

# 模型训练

result\_model = train\_model(train\_X, train\_Y)

# 提取系数

extract\_coefficients(train\_X.columns, result\_model.params)

# 获取模型摘要

get\_model\_summary(result\_model)

# 测试数据处理

test\_X, test\_Y = preprocess\_data(test\_file\_path)

# 模型预测和性能指标计算

evaluate\_model(result\_model, test\_X, test\_Y)

#进行交叉验证

perform\_cross\_validation(train\_X, train\_Y)