**华中科技大学**

**计算机科学与技术学院**

**《机器学习》结课报告**



专 业： 计算机科学与技术

班 级： 计算机7班

学 号： U202215561

姓 名： 瞿明睿

成 绩：

指导教师： 何琨

**完成日期：2024 年 5 月 16 日**

目录

[1. 实验要求 2](#_Toc167121934)

[**1.1** **实验题目** 2](#_Toc167121935)

[**1.2** **实验分析** 2](#_Toc167121936)

[2. 算法设计与实现 3](#_Toc167121937)

[**2.1** **算法选择的思路以及比较预期** 3](#_Toc167121938)

[**2.2** **数据处理** 4](#_Toc167121939)

[**2.2.1** **数据的进一步分析** 4](#_Toc167121940)

[**2.2.2** **数据处理部分** 5](#_Toc167121941)

[**2.3** **具体的算法实现** 6](#_Toc167121942)

[**2.3.1** **逻辑回归的算法实现** 6](#_Toc167121943)

[**2.3.2** **MLP的算法实现** 7](#_Toc167121944)

[**2.3.3** **MLP以及逻辑回归的超参数优化** 7](#_Toc167121945)

[3. 实验环境与平台 8](#_Toc167121946)

[4. 结果与分析 9](#_Toc167121947)

[4.1 简单的结果总结 9](#_Toc167121948)

[4.2分析和比较 9](#_Toc167121949)

[5. 个人体会 11](#_Toc167121950)

# 实验要求

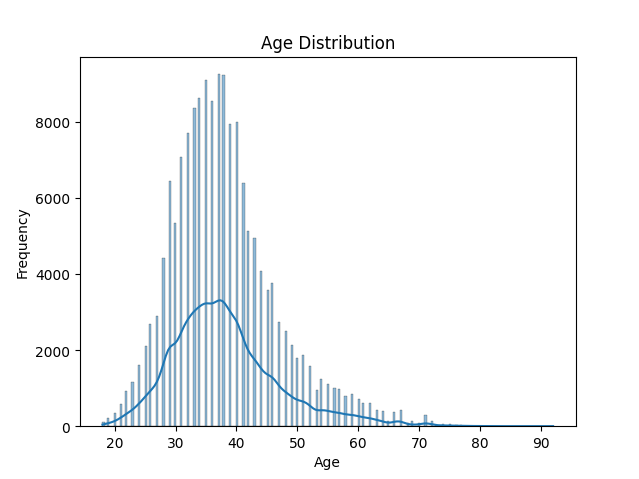
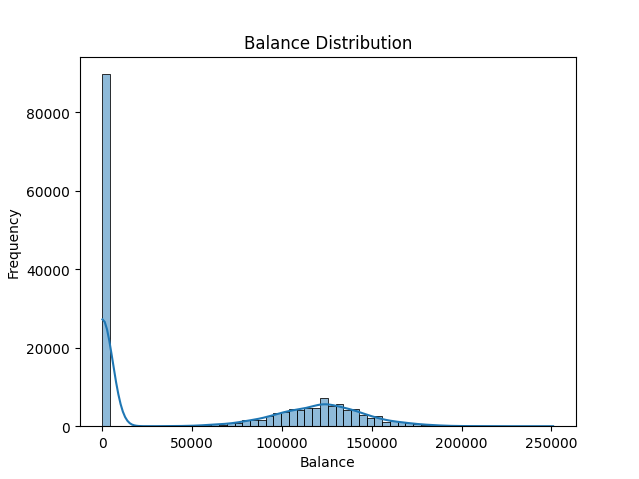
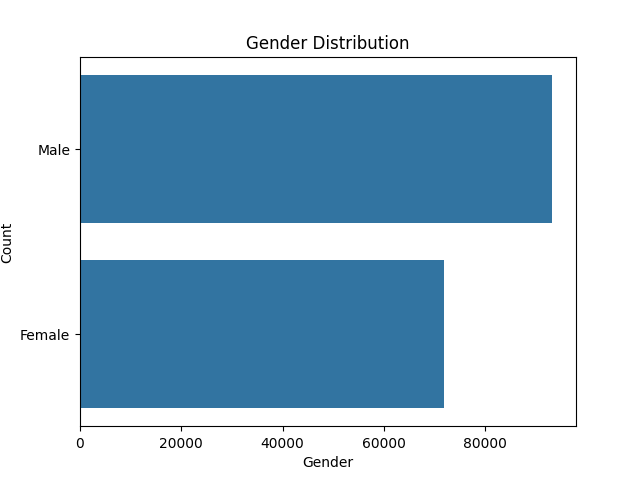
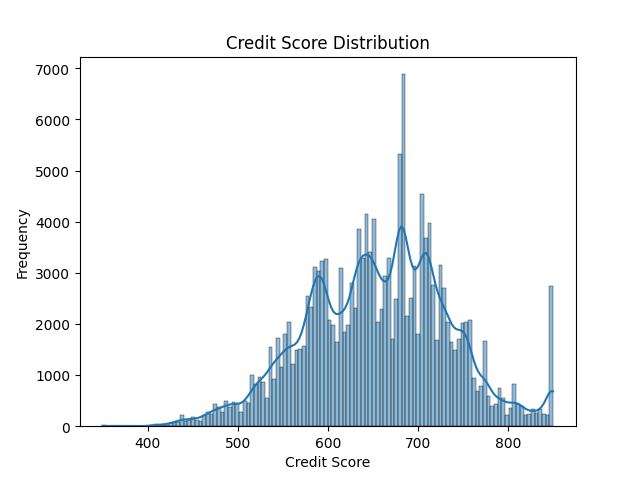
* 1. **实验题目**

Binary Classification with a Bank Churn Dataset，即使用银行的数据集进行二元分类（推荐选题1），预测客户是继续使用他们的账户还是流失。

评估方式提交在预测概率和观测目标之间的 [ROC 曲线下面积](http://en.wikipedia.org/wiki/Receiver_operating_characteristic)上进行评估，提交文件对于测试集中的每个变量，必须预测目标变量的概率。该文件应包含标头，并具有以下格式：id Exited。

* 1. **实验分析**

本次不允许使用已经实现好的模型如sklearn中的决策树，逻辑回归等，又或是torch中的隐藏层、激活函数搭建等模块，仅为实现我们需要的功能，通过我们手动实现而不是调用已经实现好的功能起到锻炼我们的能力，所以我们需要分析好我们的题目一步一步做好代码。（事实是使用这些已经搭建好的模型去做我们的任务是非常简单的）

以下为初步分析的数据分布图（并不是分类数据分布）：

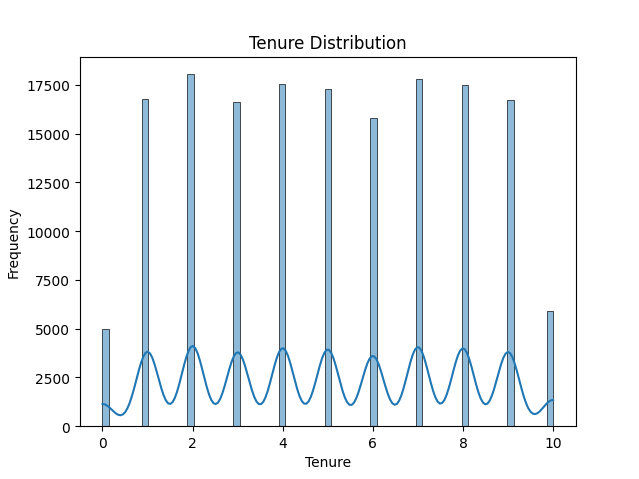
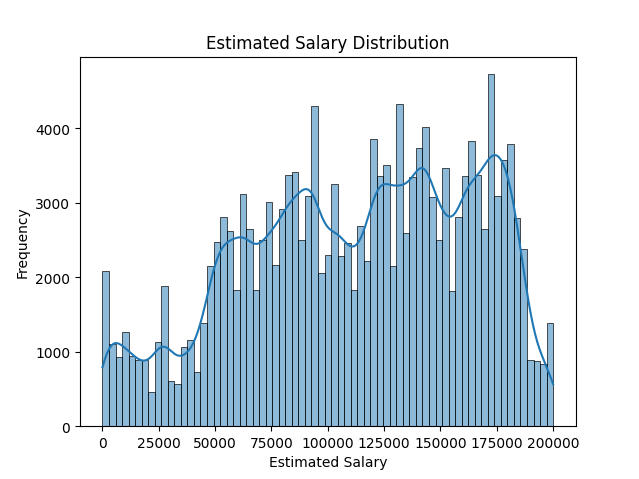
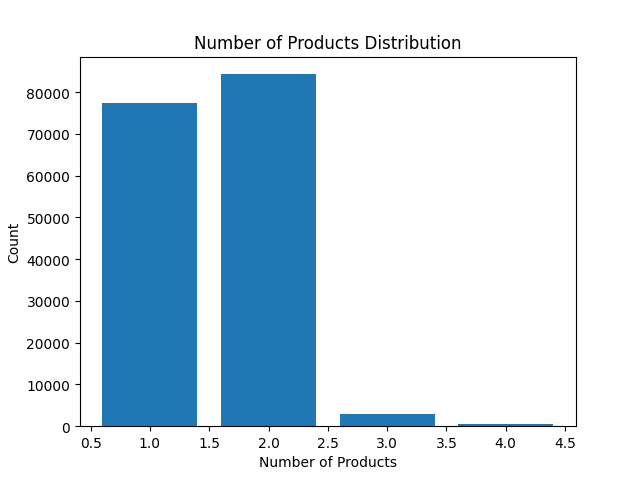


图1.1-图1.7 训练集数据分布图

如图所示，我们从画出的训练集数据分布图可以看出，首先用户是否退出的问题在性别上并不是均等的，无法忽略性别的影响。其次Tenure即用户余期应该典型的中间值多，两头少的数据特征。然后就是年龄分布，薪水分布，信誉分数分布和存款分布，可以发现这几个数据指标都是有特定的分布。我们这次任务是做二分类，但是如果做其他的工作，比如预测用户会如何进行资金操作，又或者理解用户行为，是非常有用的。最后就是用户产品持有数量，大部分也是在1.0和2.0，3.0用户很少。

# 算法设计与实现

* 1. **算法选择的思路以及比较预期**

在选择算法的过程中，由于需要手动实现，我选择了几个方向：svm，逻辑回归，以及多层感知机。

在初次尝试的过程中，svm性能表现很差，在训练集上花费的时间非常多（cuda12.1 cupy 很初级的模型训练都非常非常慢，完全比不上剩下两个的速度，可能是因为训练集的数据比较大，导致了这种问题），通常十分钟都无法进行一次训练，这使得使用cuda成为一种必须，但是这花费太大，故放弃。

第二次尝试使用逻辑回归，在实现的初步既可以得到一定的效果，但是在寻找最佳参数的时候，在不使用任何优化方法的情况下，正确率一直在0.65徘徊无法提升，给出贝叶斯寻找1000次找到的可能的最佳的参数：Learning Rate = 0.1478, Iterations = 1409, Validation AUC = 0.6512。所以我选择了逻辑回归并且决定在寻找一种方式使得能够达到更高的准确度。（事实证明逻辑回归的思路并不是不正确，而是需要做工作需要大量优化，花费的时间比其他思路多很多，使用超参数优化的方法也是很费时间的）

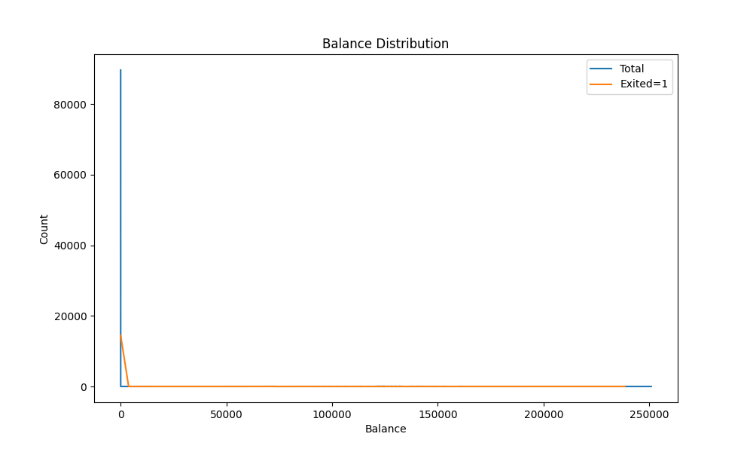
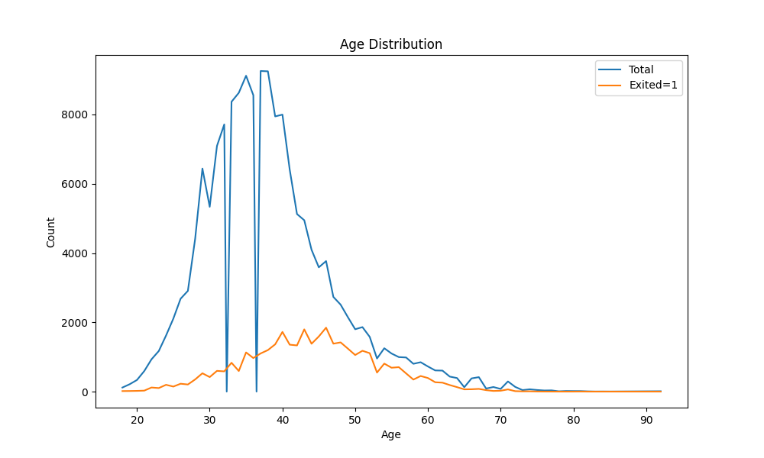
第三次尝试我选择了多层感知机，因为室友的思路分享启发了我，我也可以通过几个技术方向解决这个问题，我也想知道不同的处理对结果影响究竟有多大，在初步尝试中，我得到了相当高的准确度，某种意义上薄纱了逻辑回归，这也是符合这样的问题，因为逻辑回归其实并不太适合这样的具有一定的概率分布的数据，所以其上限是不如svm等路线的，但是我还是对其进行了优化。

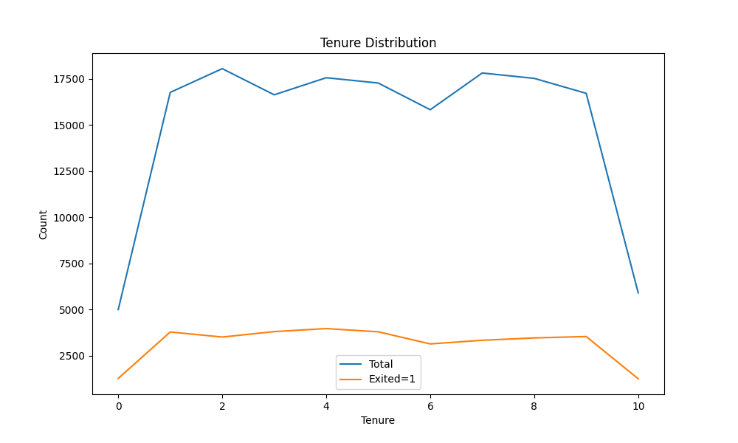
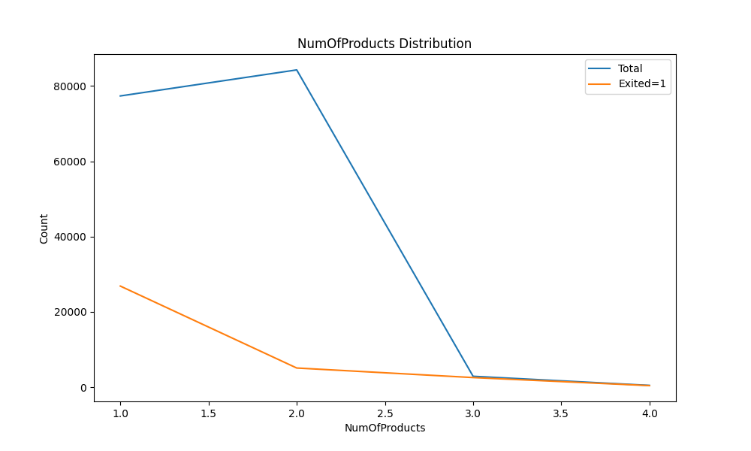
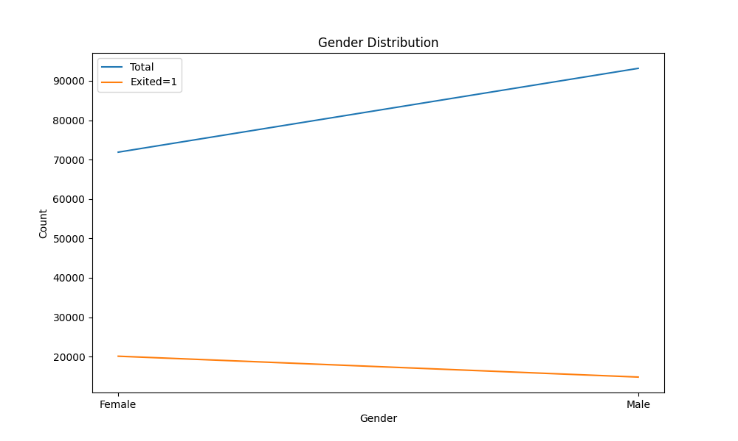
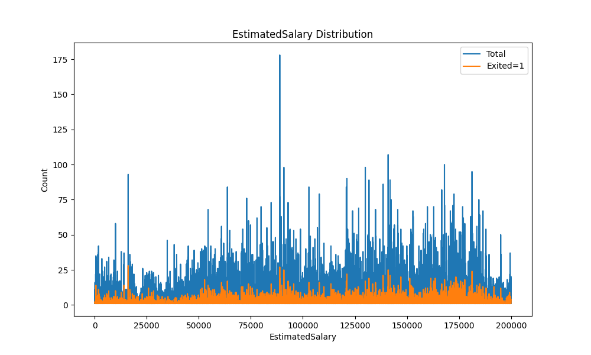
我期待的是多层感知机应该在设计完成后调试的难度上应该会比逻辑回归这种方法简单很多。

* 1. **数据处理**
     1. **数据的进一步分析**

我们在引入问题时分析过，那么我将挑其中重要的几个数据重新分析一下。

首先就是性别这样的问题，我们应该采取独热编码还是单纯的赋值？实践证明，应该采取独热编码，使用独热编码后，未经过优化的逻辑回归模型准确度提高了2%-3%，这还是调整后没有动参数的情况下。

剩下的具有特定分布的数据，是否该采取行动？我觉得并不应该，如图：

图2.1-图2.6 数据分析图

可以发现我们的数据分布并没有像我们预期的一样，但是我们可以从中分析到首先薪水变化不好做处理，因为这样的分布不是很有特征，其次就是持有产品的数量，是很明显的下降趋势，性别则是即使男性数量更多的情况下还是女性离开的数量更多，余期的则与整个用户群体的趋向使符合的。这样我们就不好对数据做什么处理去消除一些极端数据，只能通过算法实现优化了。

* + 1. **数据处理部分**

在这里先给出训练集第一个用户的的数据样式。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| id | customerID | Surname | CreditScore | Geography | HasCreCard |
| 0 | 15674932 | Okwudilichukwu | 668 | France | 1.0 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Gender | Age, | Tenure | Balance, | NumofProducts | IsactiveMember | EstimatedSalary |
| Male | 33.0 | 3 | 0.0 | 2 | 0.0 | 181449.97 |

表2.1 训练集用户数据

可以看出，有些数据是本身就是数据化的，有些数据是需要处理的，我们的处理如下，提取客户的id以在讲预测的结果和id相关联，如何我们需要将性别也转换为数值型，将男性表为1，女性表为0，再其次将地理位置也类似于性别处理，比如将法国设置为0，德国设置为1，西班牙设置为2。最后再将无用的客户id以及字符串姓名去除得到我们需要的数据。但是通过理性分析，我们真的除了这些数据就没有地方可以提取数据吗？并不是，对于一个银行用户来说，分析是否会推出，显然，如果是一个目标用户，其用户画像应该是，拥有非常多项目，同时活跃，同时应该是有经常性的资金往来，这些东西数据集中都有提及。我们也会将不把其他因素考虑进来和全考虑做一个对比。

但是这样的数据处理其实有很大的问题，其原因在于我们和模型的思考并不一样，我们对于0、1和2的感觉并不是权重，而模型很有可能当成权重而影响准确率，但是实际上三个地理本身并没有什么区别或者可比性。

* 1. **具体的算法实现**
     1. **逻辑回归的算法实现**

我这样描述函数实现：定义一个逻辑回归类，该类使用随机梯度下降（SGD）进行优化，并包含L2正则化和Nesterov Momentum。这个类有两个主要的方法：fit（用于训练模型）和predict（用于预测）。

首先我们需要定义一个逻辑回归类，其中我们需要首先初始化好模型参数，在这里我是用了L2正则化和Nesterov Momentum，所以一共4个参数，分别是学习率learning\_rate，迭代次数 n\_iterations，L2正则化的强度regularization\_strength，Nesterov Momentum的因子momentum\_factor。此外就是一些比如weights和bias是模型的参数，weight\_momentums和bias\_momentum是参数的动量。

其次我们需要定义训练模型的方法，在这里我命名为fit。在这里我们需要初始化模型的参数和动量。然后，进行迭代。在每次迭代中，先计算下一步的权重和偏置，然后更新动量，最后更新权重和偏置。

然后我们需要针对我们要面对的问题作出一些调整或者针对：即我们做的是二分类，我需要输出1或者0，这里我们使用概率的方式处理，即我们设计一个函数来讲我们函数的输出成为概率，再将概率输出为0或者1，这里我设计的方法就是如果概率大于0.5则输出1反之则输出0。

这里和原始的逻辑回归不太一样，一是使用了L2正则化方法，二是使用了Nesterov Momentum，L2正则化意义在于防止模型过拟合，在训练数据上表现得过于完美，以至于在新的、未见过的数据上表现得很差。这通常是因为学习了数据中的噪声，我们在明星的损失函数中添加一个正则项来工作，这个正则项是模型所有权重的平方和。这意味着模型的权重不能太大，否则损失函数的值就会增加。这防止模型过于依赖某一个特征。Nesterov Momentum则是用于加速梯度下降，可以提供更快的收敛速度和减少震荡。

这里提出为什么放弃了SVM，因为标准的SVM并不支持提供概率预测，需要使用PLATT缩放，在原始训练不使用SGD的情况下已经很慢，那我就认为能够直接输出概率的模型，比如逻辑回归或者随机森林的模型更好。最后我依然会贴出进行比较。

* + 1. **MLP的算法实现**

多层感知器是一种前馈神经网络，包含一个输入层、一个或多个隐藏层和一个输出层。每一层都由多个神经元组成，每个神经元都与下一层的所有神经元相连。这些连接被称为权重 。

首先我们需要初始化，设定好我们的输入大小，隐藏层大小和输出大小，此外还有权重和偏执，其次就是我们的结构建立，forward和backward函数，前者股则向前传播，计算每一层的激活值然后应用激活函数，如果是最后一层，则应用映射函数输出0或者1，如果不是最后一层则继续使用ReLU激活函数，后者实现的是反向传播，计算权重和偏执，首先需要计算出曾的误差然后在反向遍历每一层，计算每一层的误差，如果是最后一层，使用映射函数的倒数，如果不是最后一层则使用ReLU激活函数的导数。其中还需要定义好我们的fit函数训练模型，对于每一个epoch，执行向前传播，计算梯度更新参数。

* + 1. **MLP以及逻辑回归的超参数优化**

通过和很多语言模型的询问我得知有很多超参数优化的方式：网格搜索、随机搜索、贝叶斯优化等等，在这里我经过实验决定使用贝叶斯优化，一是因为贝叶斯优化有理论上最少的步骤数，而是因为我们优化的范围并不大，参数可取值不如。此外强化学习也是一种很不错的想法，但是感觉时间不足以完成遂放弃。其实现比较简单：我们首先需要定义我们需要的参数，比如我们使用的逻辑回归中有四个参数，我们需要定义参数的范围然后定义一个函数，该函数接受超参数作为输入，并返回模型的负AUC分数 ，在这里返回负AUC分数，因为我们希望最小化这个函数，输出最佳的超参数，最后使用最佳的超参数来训练模型

这里附上逻辑回归贝叶斯优化的结果，可以发现优化得出的超参数并不一定是固定的，但是结果却是相似的，我怀疑有些错误但是确实如此，如下图：

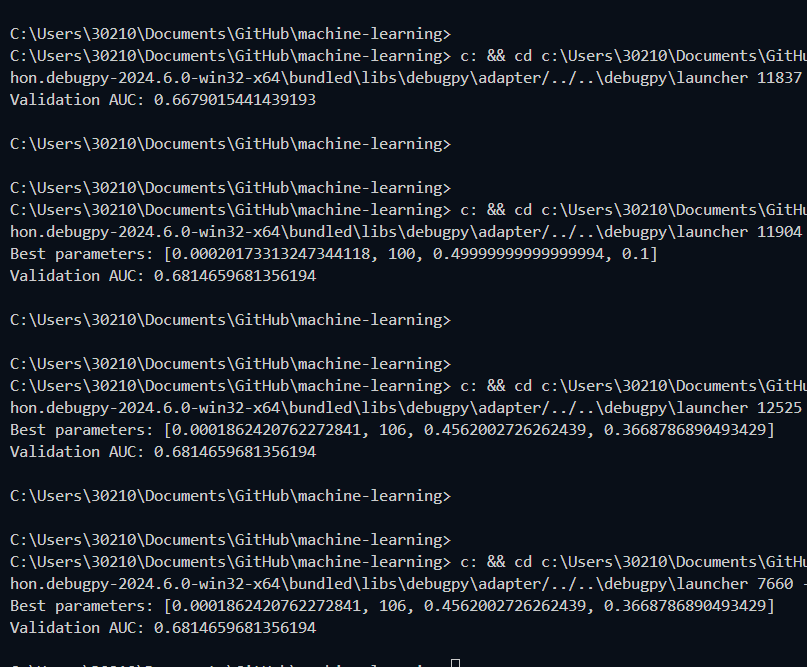


图2.7超参数优化

# 实验环境与平台

平台1：

Windows10操作系统内部版本 19045.4412

处理器 11th Gen Intel(R) Core(TM) i9-11900KF

Gpu: 2080ti 11g

平台2：

Windows 11 专业工作站版 操作系统版本 22631.3447

处理器 AMD Ryzen 7 6800HS with Radeon Graphics

Gpu: 3060 laptop 6g

所用库：numpy, bayesian-optimization, pandas, scikit-learn, skopt, matplotlib, torch, joblib, xgboost, multiprocessing

基于cuda12.1

# 结果与分析

## 4.1 简单的结果总结

在本次实验中，我们对数据进行了详细的预处理。我们将数据中的类别变量 Geography 和 Gender 转换为数值表示，并且应用了独热编码（One-Hot Encoding）以避免数值映射带来的潜在影响。独热编码将类别变量扩展为多维特征向量，从而使模型能够更好地处理类别特征。

原始特征：

- Geography: France, Germany, Spain

- Gender: Male, Female

独热编码后的特征：

- Geography\_France, Geography\_Germany, Geography\_Spain

- Gender\_Male, Gender\_Female

为了提高模型的训练效果，我们对所有数值特征进行了标准化处理。标准化处理将特征的均值调整为0，标准差调整为1，从而使特征具有相同的尺度。这有助于加速模型的收敛并提高模型的性能。

在验证集上，我们通过预测概率来评估模型的AUC值：

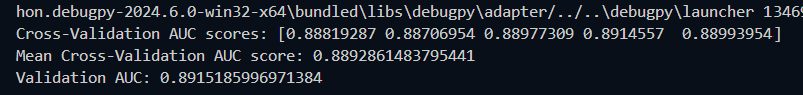
在对测试集进行预测后，我们生成了最终的提交文件。模型对测试集进行了预测，并输出了每个样本的预测概率

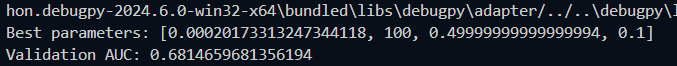
通过以上步骤，我们实现了一个从数据预处理、特征工程到模型训练和评估的完整机器学习流程。使用逻辑回归模型，我们在验证集上达到了的AUC值，证明了模型的高效性和稳定性。独热编码和标准化处理确保了特征处理的一致性，为模型提供了良好的基础数据。

## 4.2分析和比较

逻辑回归、mlp以及svm是三种不太一样的方向，svm和mlp都不太好直接输出某种数值用作概率直接输出，但是逻辑回归可以。

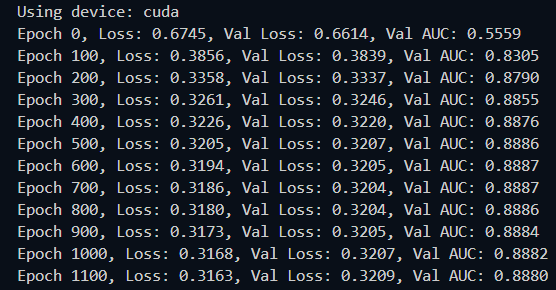
在没有使用SGD之前，如果使用SVM，我采用的是对数据集中的每个样本进行迭代，这实在是太慢了，当我使用SGD去处理的时候发现快了很多（因为我一直没有注意到训练集一共160000行，我用2080ti跑了好久都没结果一度认为我写什么写出出问题了） 但是准确度和和没怎么优化一样，感觉非常好笑。

其次就是逻辑回归，我第一次接触觉得很简单，但是但我尝试使用打包好的逻辑回归去练随随便便就能得到一个很高的准确度，我同时也尝试了别的模型比如xgboost，准确度如下图：



两个实现的准确度比较确实高下立判。

在mlp实现的过程中，我还利用过torch的定义，即import torch.nn as nn，准确度如下：



由本人自行写成的mlp准确度非常糟糕，感到非常失落。

可以发现三种模型都需要非常完备的数据处理，加上合理的模型写成，最后能在交叉验证的部分取得很好的结果，可以说缺一不可。

从问题的角度出发，最适用的就是SVM，但是我这里数据处理方法并不合适，总的来说，逻辑回归作为一种统计学习方法，很适合二分类问题。通过函数将线性回归的输出转换为概率，然后根据这个概率进行分类。模型简单，易于理解和实现，但在处理如本题这样的大概率不是线性的问题时，上限并不高，表现不佳。SVM作为一种二分类方法可以处理线性和非线性问题，但在大数据集上可能会很慢，那就比如本题。MLP 神经网络，由一个输入层、一个或多个隐藏层和一个输出层组成。每一层都由多个神经元组成，处理复杂的非线性问题，且可以通过增加隐藏层和神经元的数量来增加模型的复杂性。但是容易过拟合，需要调整参数实现更高的准确度。

# 5. 个人体会

个人第一次接触python处理这样的问题，从学习pip，py的语法以及各种各样的模型，打包好的数据处理方法都让我觉得十分新奇，也觉得人工智能是一个非常有意思的方向。

我突然明白了调参的意义，在模型的接口，在模型的参数上做调整，我貌似能做的只有这些，这也让我感觉到了一些无力感，我做的算法都是经过验证的，Github copilot学习了也能做，我觉得写出的代码真不如他自动生成的，虽然帮助我解决了很多错误，但是我还是觉得有点失落。第一次实际用py发生了很多问题，有很多不会的问题，但是我很庆幸都解决了，总体上是一次历练。