**华中科技大学计算机科学与技术学院**

**机器学习报告**



专 业： 计算机科学与技术

班 级： 计算机1807班

学 号： U201814683

姓 名： 荣烁

成 绩：

指导教师： 邹复好

**完成日期： 2020年 6月 20**

## 一、实验题目：个人收入预测

## 二、实验要求

给定训练集income.csv，要求根据每个人的属性值来判断此人年收入是否大于50K。

训练集介绍：

（1）CSV文件，大小为4000行×59列;

（2）4000行数据对应着4000个人，ID编号从1到4000;

（3）59列数据中，第一列为ID，最后一列label(1或0)表示年收入是否大于50K，中间的57列为57种属性值。

任务描述：

（1）将数据中前3000项作为训练集，后1000项作为测试集，使用logistic回归进行二分类，实现语言要求为Python；

（2）在使用梯度下降法时，调整学习率的固定值，有能力的同学可以学习并使用动态调整学习率的方法，探究不同学习率的选择对训练误差收敛速度的影响，绘制misclassification rate曲线进行比较并分析。

评测标准：

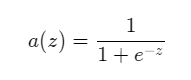
（1）要求计算出准确率。

（2）要求画出训练和测试loss曲线。

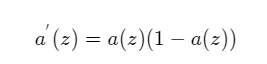
（3）要求调整多个学习率和正则化参数后给出上面的结果。

## 三、算法设计

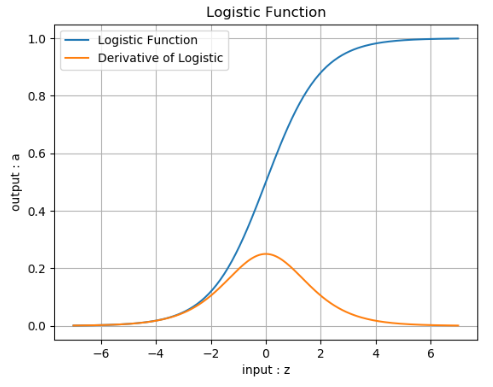
使用对率函数(Logistic Functio)作为二分类函数。对率函数公式如下所示。



对率函数导数如下所示。

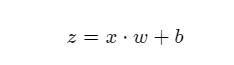


对率函数与其导数图象如下所示。



使用对率函数进行概率计算，对实数域的数字进行压缩刀(0,1)之间，并且返回一个概率值。在这个概率值接近1时因为是正例，否则是负例。在本次实验中涉及到的样本都是多特征，所以在计算预测值时需要使用矩阵运算。假设x表示特征值向量，w表示权重，b表示偏置，z表示预测值，a表示分类计算后的结果。那么可以得到预测值计算和分类计算的公式如下表示。

预测值计算。



分类计算。

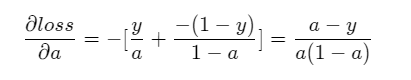
C:\Users\荣烁\Documents\Tencent Files\905869875\FileRecv\MobileFile\Image\A_H1HAY(9T@TA9()[3G7_FR.png

在此基础上，使用二分类交叉熵损失函数作为损失函数。二分类交叉熵损失函数公式如下表示。

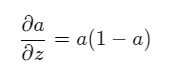
C:\Users\荣烁\Documents\Tencent Files\905869875\FileRecv\MobileFile\Image\KL7@F1XV{PV6}DT2[6K_XZS.png

**梯度下降法:**

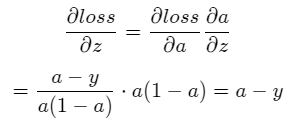
对二分类交叉熵损失函数，可以计算其对a的梯度。



计算损失函数a对z的梯度



利用链式法则计算损失函数loss对z的梯度。



根据上述推导易得loss对b的梯度与loss对w的梯度。

Loss对b的梯度在形式上与loss对z的梯度相同。Loss对w的梯度如下形式。

C:\Users\荣烁\Documents\Tencent Files\905869875\FileRecv\MobileFile\Image\G5H9RI`X{X5OP@L_94L~]OI.png

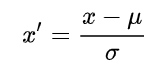
根据公式推导得到了loss对w和loss对b的梯度。那么利用梯度下降法进行求解目标的w和b两个参数，来得到最终的目标函数。每次更新后的w和b可以表示为(dw和db表示w和b梯度)。w = w + dw以及b = b + db。

**随机梯度下降法:**

因为使用梯度下降法时，每次迭代都要遍历全部样本，来对w和b进行更新，在面对大样本容量时速度较慢，这里采用每次都随机选取一个样本进行更新参数的方式进行。

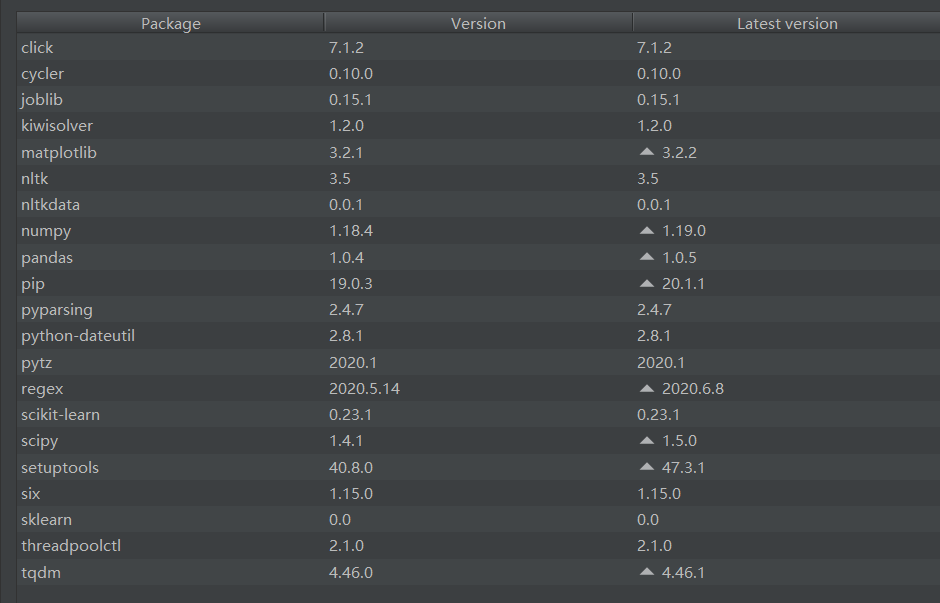
**对样本特征数据的事先处理，mean-std归一化:**

为了提高模型的收敛速度和模型的精度。在这里事先对样本集的样本特征数据进行归一化处理。这里采用mean-std归一化。下图的公式可以描述这一过程



## 四、实验环境与平台

本次实验是基于python语言实现的，使用的ide是pycharm。配置环境如图所示。



主要是使用了numpy、pandas以及sklearn三个库。

Numpy主要是用于数据的处理等过程。使用pandas库进行数据集csv文件的读取操作。Sklearn主要是使用其中包含的trian\_test\_split以及standardscaler函数，分别用来实现数据集的划分和标准化过程。并且使用了sklearn中的losisticregression模型与自己实现的模型进行对比。

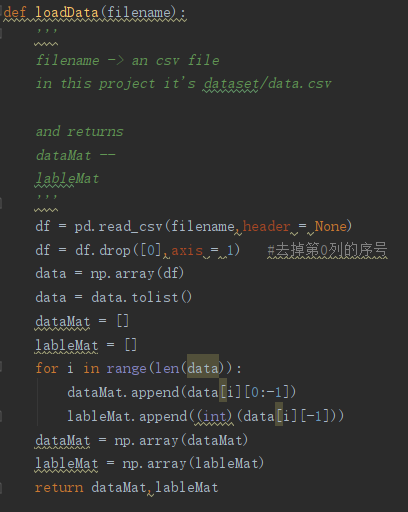
创建的代码文件夹中包含两个py文件以及用于保存数据集的文件夹dataset。如下所示。



其中，LogisticRegression.py文件中使用python3实现了logistic regression模型。Sklearn-LogisticRegression.py文件中利用sklearn的模型实现了logistic regreesion模型。

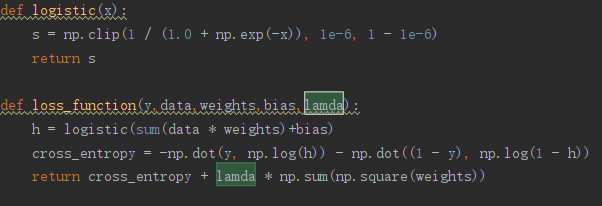
## 五、程序实现

(1)数据集的读取loadData



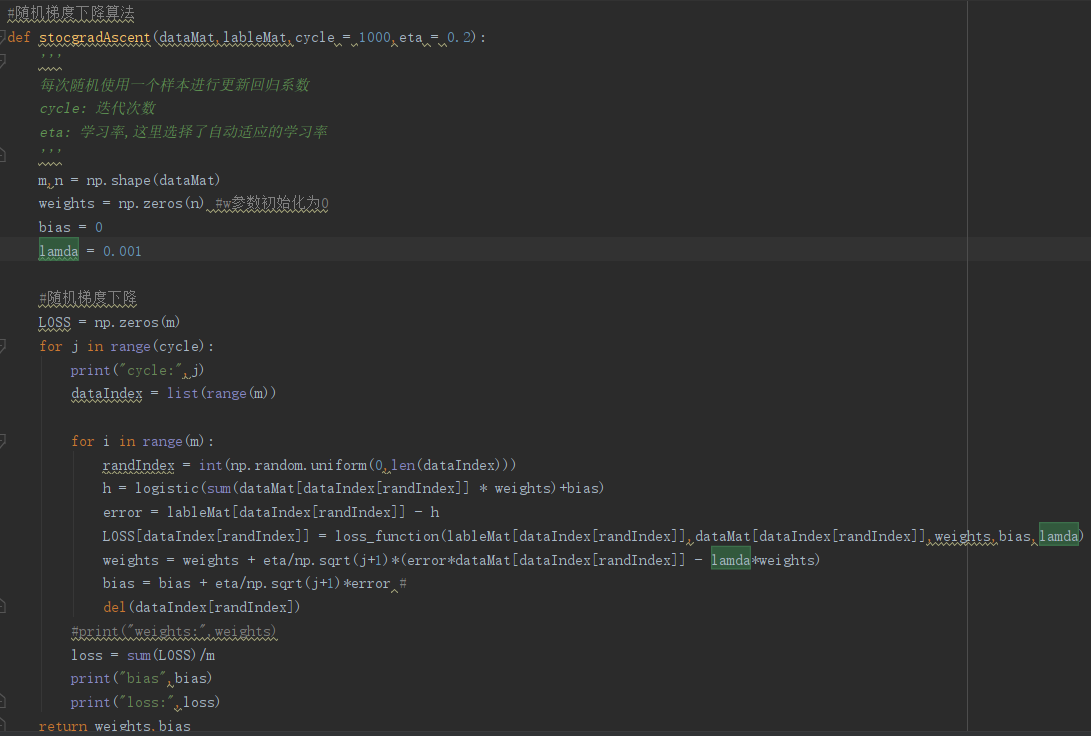
这里采用了pandas里的read\_csv函数来读取csv文件。并且因为数据集中的第一列的数据其实是每个数据的序号，所以选择去掉。去掉后将标签与特征分开，就得到了数据集的样本和样本对应的标签。这dataMat和lablaMat作为返回值。

(2)分类函数和损失函数



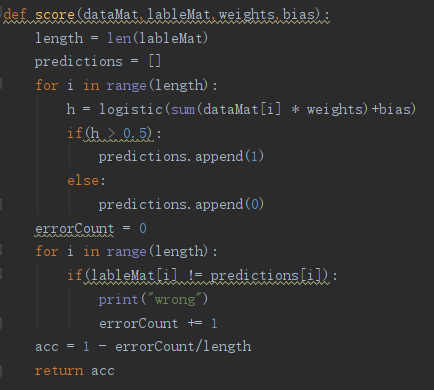
Logistic函数就是对率函数。Loss\_function就是二分类交叉熵损失函数。这里的loss\_function中有lamda作为正则化惩罚过拟合的参数。使用了clip函数防止到达边界时，log0的出现。

(3)随机梯度下降算法



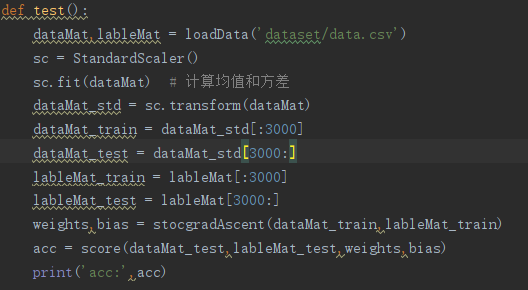
其中参数cycle是表示迭代次数。eta参数是初始学习率。学习率会随着迭代次数的增加而下调。bias是偏置，lamda参数是正则化系数。在梯度下降的开始，bias和weights矩阵参数都被初始化为0，lamda可以调整。程序在每一次迭代中，都随机选取一个样本进行系数的更新，直到每个样本都被拿来更新系数。梯度的计算公式在上述已经提及，不再赘述。最终所有迭代都进行完之后，将返回根据梯度下降法得到的参数weights和bias。

(3)模型的评估



根据迭代的得到的weights和bias参数来进行预测。并由预测值和真实值得到准确率。函数最后返回准确率。

(4)运行功能



整合上述所有的函数，实现logistic regression功能进行分类。对原数据进行了标准化。并且没有随机划分，而是采用固定的前3000为训练集，后1000为测试集。

(5)使用sklearn实现



直接调用sklearn中现有的函数即可。同样最后会输出一个准确度。

## 六、实验结果

(1)使用sklearn提供的函数进行测试，发现最终准确率的大致范围。

C:\Users\荣烁\Documents\Tencent Files\905869875\FileRecv\MobileFile\Image\FZNCB52JU]%WPU}IYIUG]EX.png

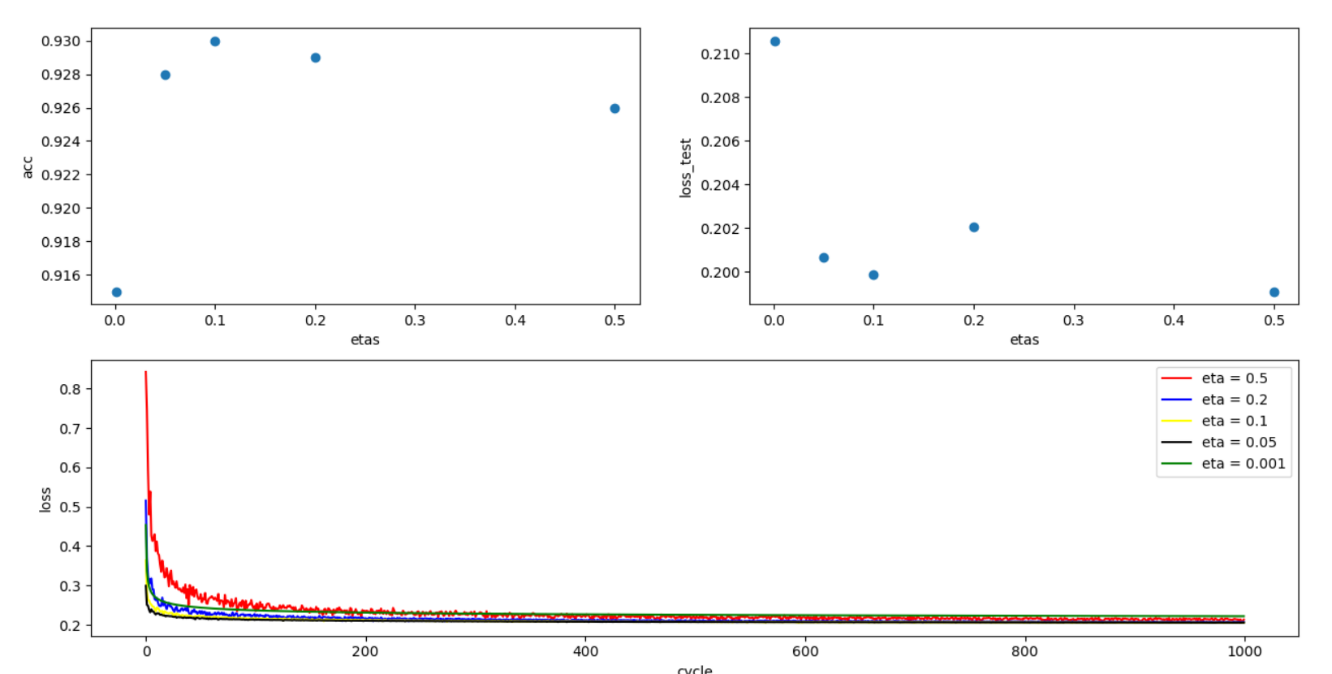
运行后发现，准确率大致就在0.927。

(2)使用自己编写的logistic regression进行测试，并且调整不同的参数来看对训练结果的影响。

1.不同学习率对准确率、测试loss和训练loss的影响。

这里选用的学习率分别为eta = 0.5,0.2,0.1,0.05,0.001,分别进行训练。每次迭代1000次。其中第一张图是选用不同学习率时，最后的准确率。第二张图是选用不同学习率时，最后的test loss。第三张图是在训练时，选用不同的学习率时，训练过程中loss的变化情况。

根据前两个图像展示，在学习率不是太小的情况下，准确率相近，且test loss都不是很大，但学习率为0.001时，准确率和test loss 都不是很好。而且根据第三个图像可以看出。学习率最小时，在迭代接近1000次时，收敛程度不如其他学习率。并且在这些学习率中，采用eta = 0.05时loss收敛最快。



2.不同正则化系数lamda对准确率、测试loss和训练loss的影响

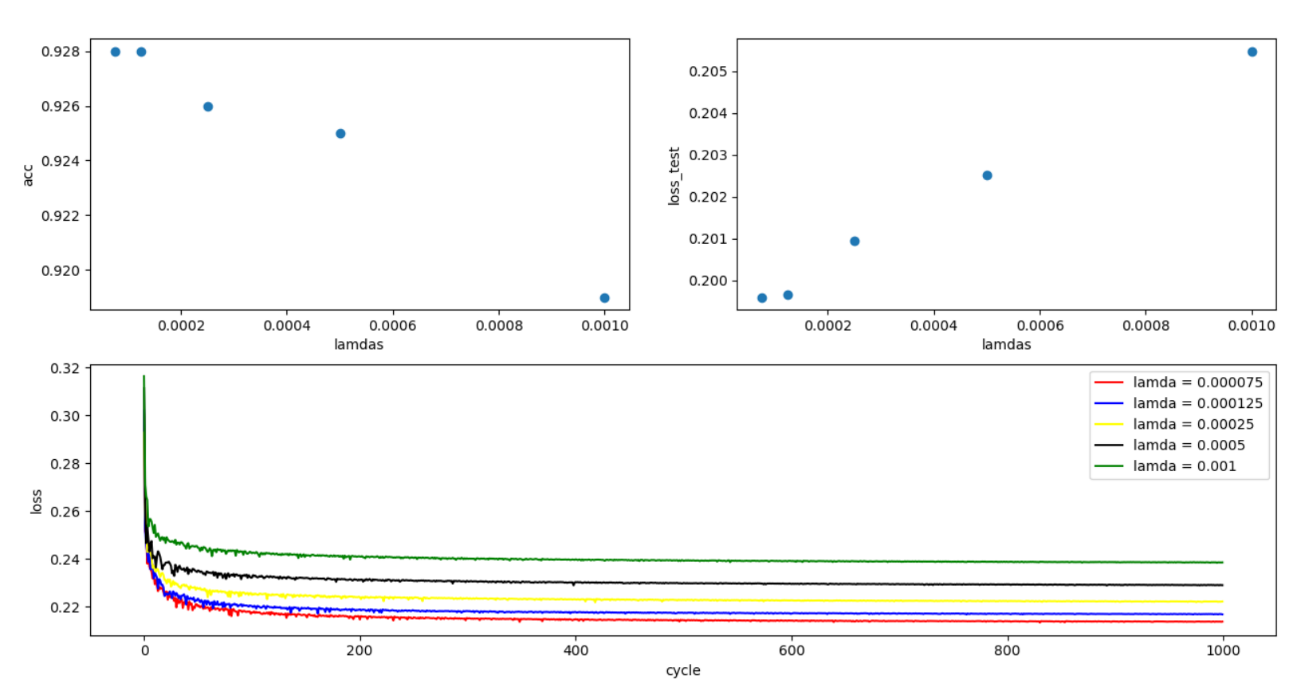
这里进行了三组迭代。每次都有五个不同的lamda系数。这里选用的正则化系数分别为

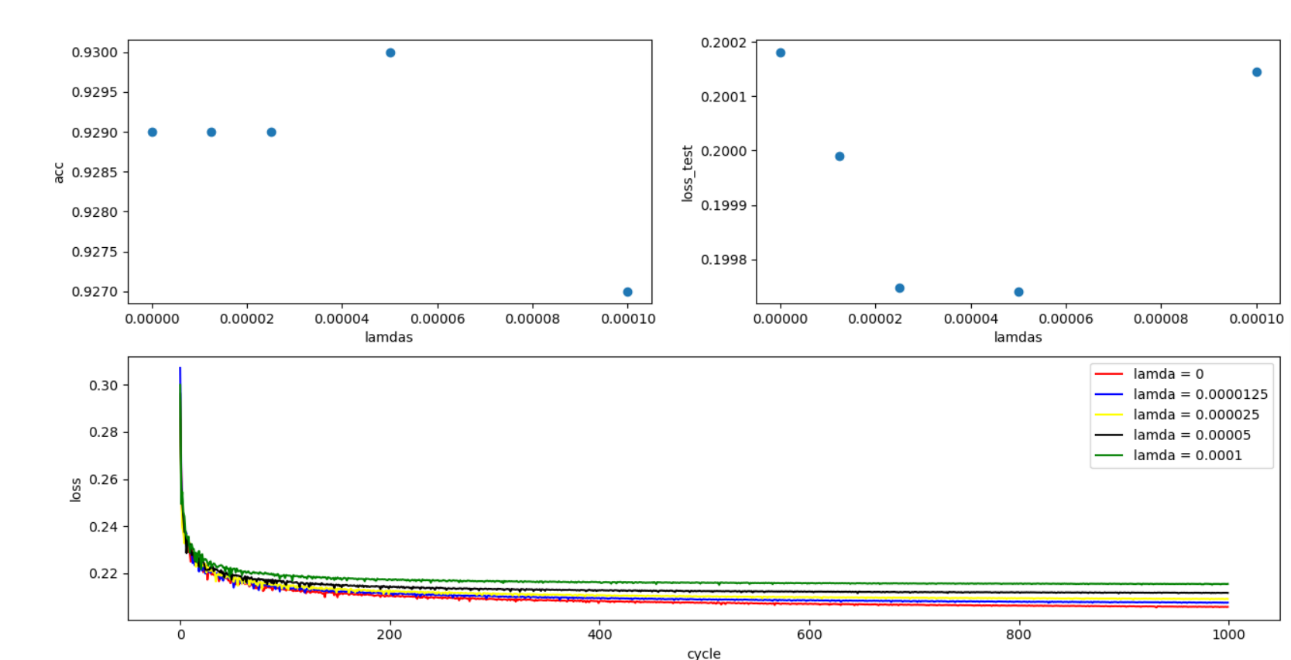
(1)lamda = 0.001,0.0005,0.00025,0.000125,0.000075

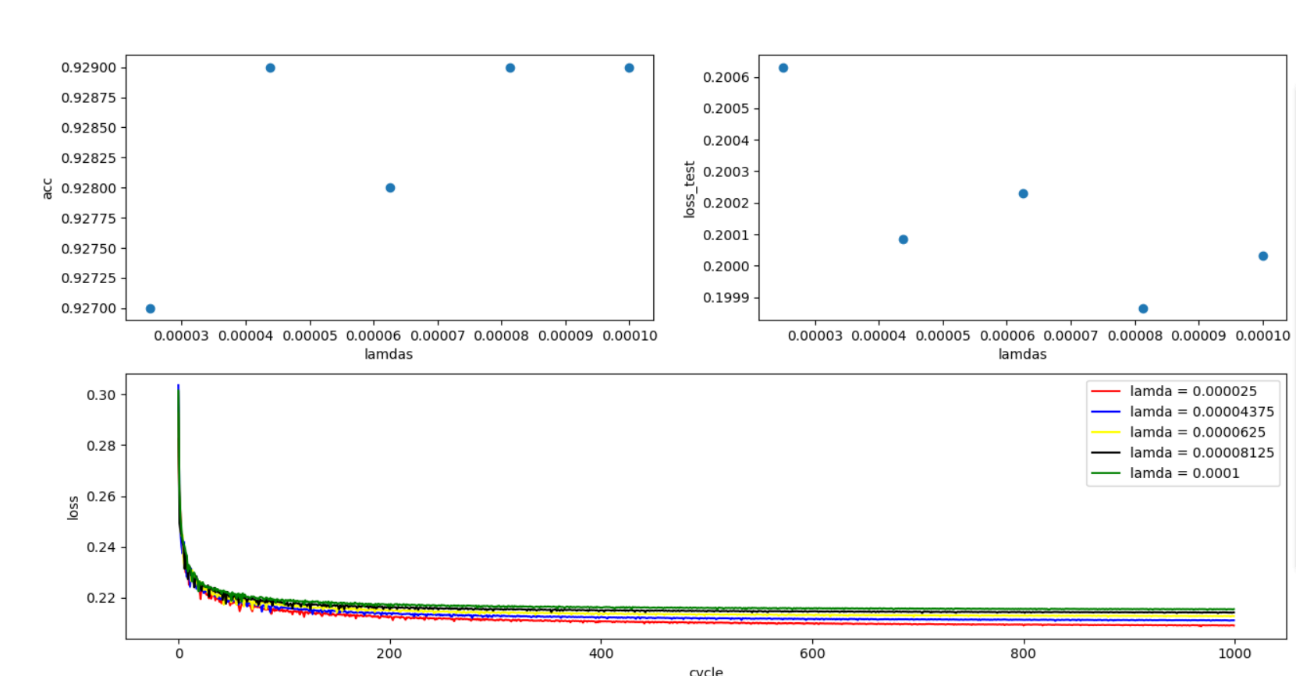
(2)lamda = 0.0001,0.00005,0.000025,0.0000125,0

(3)lamda=0.0001,0.00008125,0.0000625,0.00004375,0.000025

分别进行训练,每次迭代1000次。其中每张大图的左上角是选用不同正则化系数时，最后模型的准确率。左上角图像是选用不同正则化系数时，最后的test loss。下面的图是在训练时，选用不同的正则化系数时，训练过程中loss的变化情况。







由图可以看出，因为选用了随机梯度下降的方式，在选用同一正则化系数时，可能会有不用的准确率。但根据图可以看出，在0到0.0001之间选用正则化系数时,表现效果都很好。

## 七、结果分析

根据实验结果可以得到一些信息。使用logistic regression进行收入预测的最好准确率在0.93附近。与使用sklearn提供的函数训练得到的准确率相比相差不大。

并且通过调整学习率eta。发现在初始学习率较低时，模型的loss收敛不是很好，导致收敛率不如其他初始学习率。原因很好分析。因为在训练过程中采用了学习率随迭代次数动态减小的方法，在初始学习率很大的情况下，也不会影响后续的模型训练。在初始学习率较小的情况下，模型在迭代过程中收敛速度慢，导致模型最后没有很好的收敛。

在调整正则化系数的过程中，发现在选用0到0.0001之间的正则化系数时，都能得到较好的准确率。因为使用正则化系数可以减小模型过拟合的情况，这里推测因为迭代次数比较而言不算太多，所以模型没有达到会产生过拟合的情况。所以在使用较小的正则化系数时，能达到较好的结果，而正则化系数较大时，反而不如不使用正则化。

# 参考文献

[1]周志华.机器学习[M].北京:清华大学出版社,2020年:53-63.