

**机器学习实验报告**

**课程名称： 机器学习**

**专业班级： 计卓1701**

**学 号： U201714466**

**姓 名： 张淇**

**指导老师： 凌贺飞**

**报告日期： 2020年6月25日**

**计算机科学与技术学院**

目录

[1 任务 3](#_Toc44092847)

[2 算法介绍 3](#_Toc44092848)

[3 具体实现步骤与源码分析 4](#_Toc44092849)

[4 实验测试 6](#_Toc44092850)

[5 结论 8](#_Toc44092851)

# 任务

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。

（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。

**要求：**

1. 计算出准确率。
2. 画出训练和测试loss曲线。
3. 调整多个学习率和正则化参数后给出上面的结果。

# 算法介绍

逻辑回归由于存在易于实现、解释性好以及容易扩展等优点，被广泛应用于点击率预估（CTR）、计算广告（CA）以及推荐系统（RS）等任务中。逻辑回归虽然名字叫做回归，但实际上却是一种分类学习方法。

线性回归完成的是回归拟合任务，而对于分类任务，我们同样需要一条线，但不是去拟合每个数据点，而是把不同类别的样本区分开来。

对于二分类问题，y∈{0,1}，1表示正例，0表示负例。逻辑回归是在线性函数输出预测实际值的基础上，寻找一个假设函数函数, 将实际值映射到到0~1之间，如果，则预测y=1，即y属于正例；如果，，则预测y=0,即y属于负例。

逻辑回归中选择对数几率函数（logistic function）作为激活函数，对数几率函数是Sigmoid函数（形状为S的函数）的重要代表

图片包含 文字, 大, 白色

描述已自动生成

函数图示

则逻辑回归输出的预测函数数学表达式为

其中是参数向量。对于的直观解释是：对于给定的输入,表示其对应类标y=1,即属于正例的概率。

如果说线性回归是对于特征的线性组合来拟合真实标记的话，那么逻辑回归是对于特征的线性组合来拟合真实标记为正例的概率的对数几率.

Adagrad是解决不同参数应该使用不同的更新速率的问题。Adagrad自适应地为各个参数分配不同学习率的算法。其公式如下：

图片包含 游戏机

描述已自动生成

随着我们更新次数的增大，我们是希望我们的学习率越来越慢。因为我们认为在学习率的最初阶段，我们是距离损失函数最优解很远的，随着更新的次数的增多，我们认为越来越接近最优解，于是学习速率也随之变慢。

# 具体实现步骤与源码分析

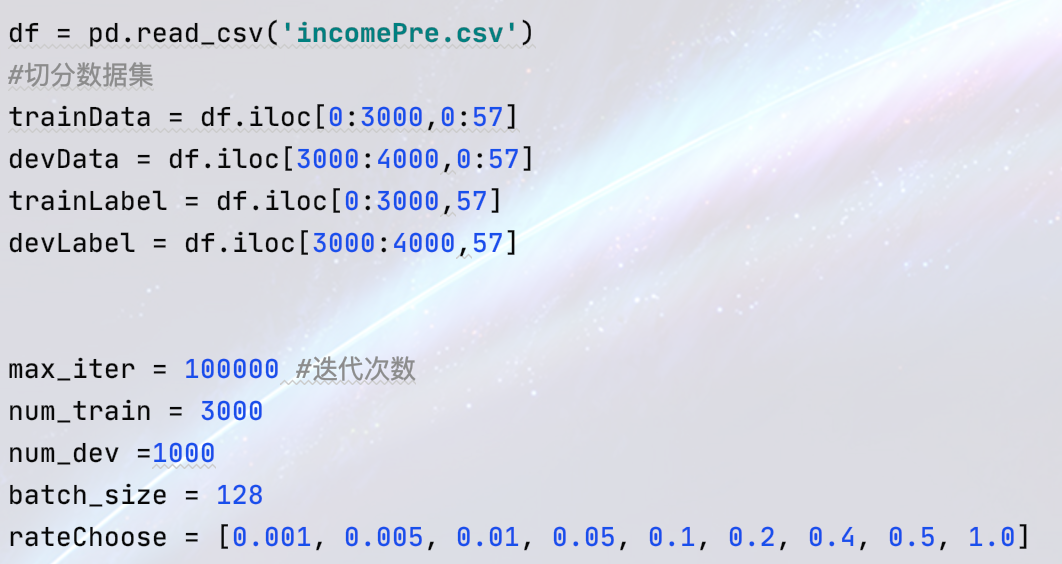
首先使用python对数据进行读入，这里采用python内置的pandas库中的read\_csv对数据进行读入，通过shape[0]可以获得一共有多少行数据，shape[1]表示有多少列，为了计算的正确性，需要对数据进行归一化，否则数值大的数据会将数值小的数据的影响抹去，导致错误的结果。对于归一化，一种做法是使用最大值和最小值进行归一化，比如x就被归一化为(x-min)/(max-min)，另一种做法是使用平均值和标准差进行归一化，比如x就被归一化为(x-mean)/std，在本次试验中我们会对这两种方式进行比较。

归一化代码如下，我们将归一化后的数据存入新的csv文件进行保存。

一些文字和图片的手机截图

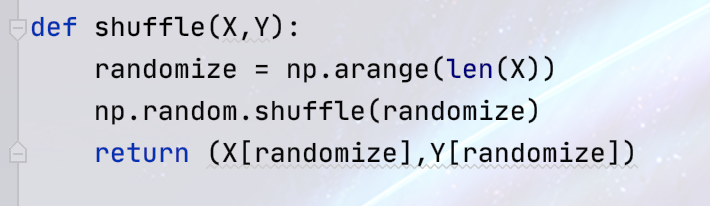
描述已自动生成

在进行数据归一化之后，就可以切分数据集并设置学习率和迭代次数了。在pandas中使用iloc可以进行块的切分，比如df.lioc[a:b,x:y]就表示从第a行到第b行的从x列到y列的所有数据，这样我们就可以很轻松的把数据划分为训练集、测试集、训练集标签和测试集标签。

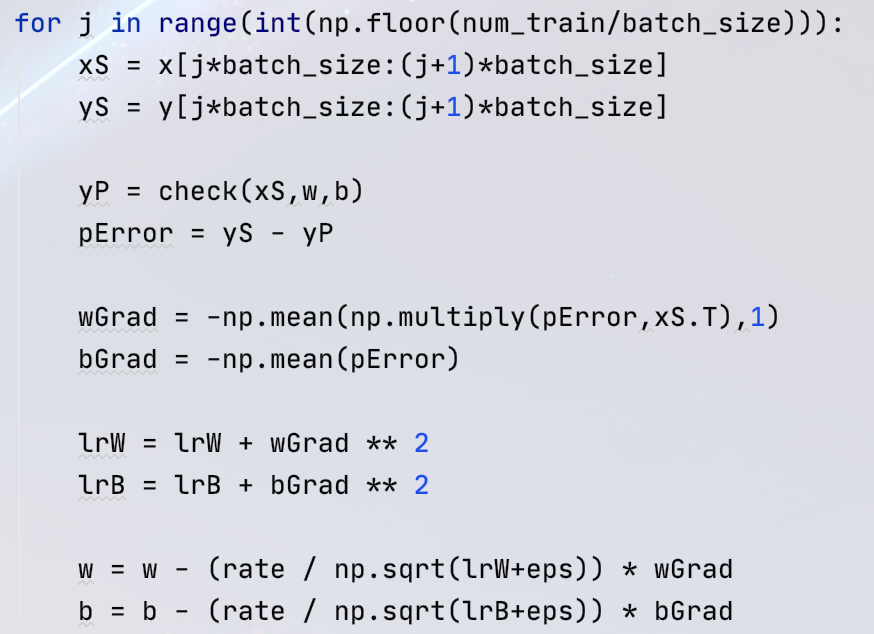


之后就可以把数据传入train函数进行训练了。在训练时有一些要点需要注意：

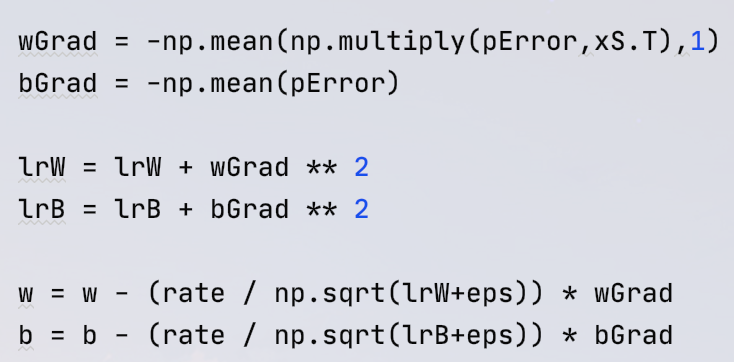
1. 每次迭代时应该将数据打乱，避免由于数据顺序带来的对于训练结果的影响



1. 对于每次迭代时，逐个数据进行迭代的做法效率比较低，可以选取一批数据（通常是2的n次幂）同时进行迭代，这样其实相当于对数据做了分块，现在每一小块中进行训练，最后对所有块进行训练，这样可以提高每次迭代的效率。



1. 因为采用了分批训练的方法，所以直接使用矩阵乘法进行梯度的运算，并且进行更新，这里采用的是Adagrad算法对w和b进行更新。



1. 对于结果的检测，loss函数的运算和准确率的运算代码如下：

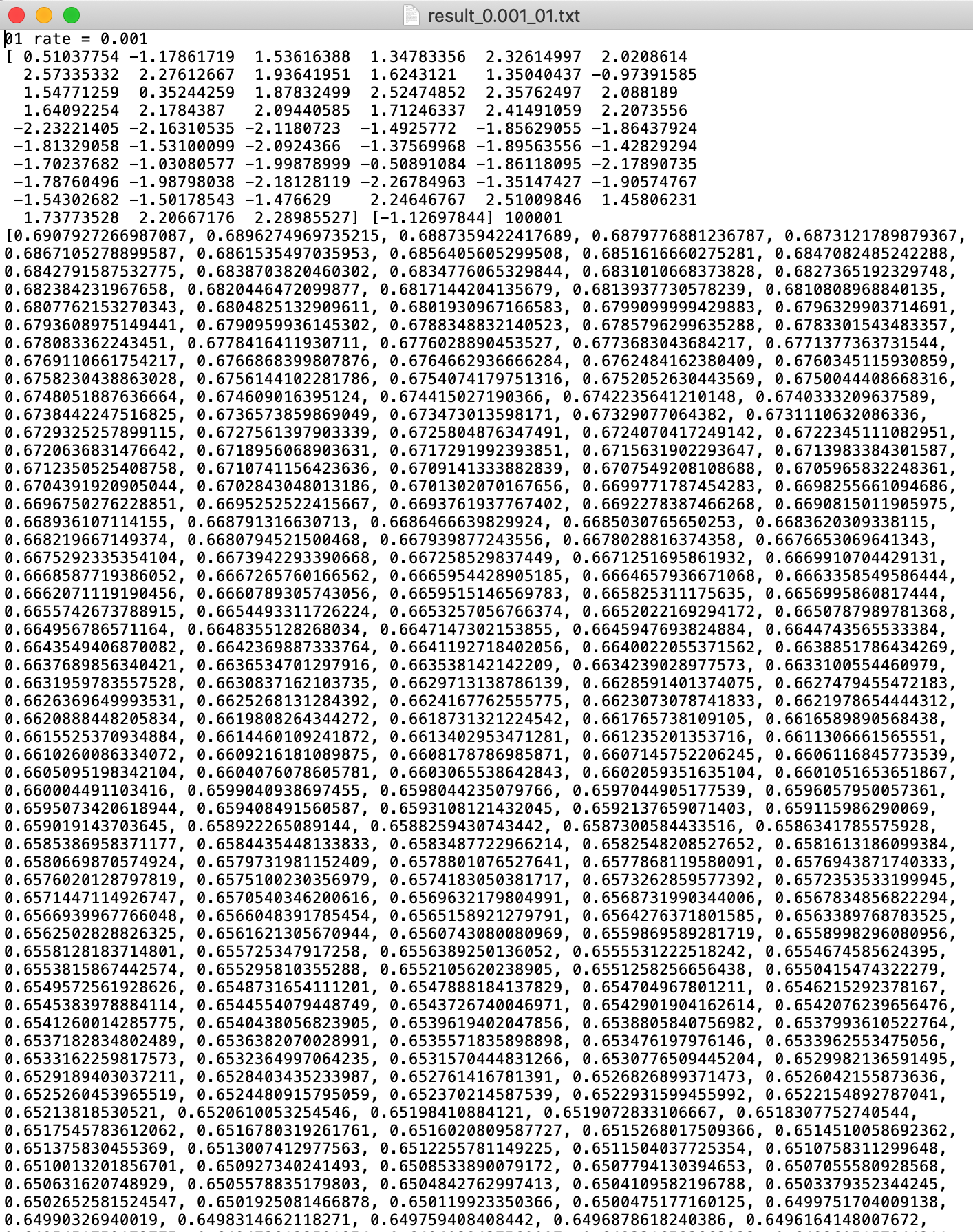
手机屏幕截图

描述已自动生成

最后将结果进行输出即可。

输出文件和输出文件内容如下：

电脑屏幕的照片上有文字

描述已自动生成

# 实验测试

具体输出文件可以在<https://github.com/copyrightpoiiiii/HUST-ML-2020>中找到。

程序采用了两种数据归一化的方法，一种是使用最大值、最小值进行归一化，另一种是使用平均数和标准差进行归一化。

对于Adagrad算法，初始学习率选取了0.001、0.01、0.1、0.2、0.5、1.0进行测试

经过100000次迭代后，准确率如下：

|  |  |  |
| --- | --- | --- |
|  | 训练集正确率 | 测试集正确率 |
| Maxmin\_0.001 | 87.06% | 87.00% |
| Maxmin\_0.01 | 92.16% | 92.00% |
| Maxmin\_0.1 | 93.06% | 92.90% |
| Maxmin\_0.2 | 93.16% | 92.80% |
| Maxmin\_0.5 | 93.16% | 92.70% |
| Maxmin\_1.0 | 93.13% | 92.80% |
| Meanstd\_0.001 | 92.46% | 92.20% |
| Meanstd\_0.01 | 92.96% | 93.00% |
| Meanstd\_0.1 | 93.20% | 92.80% |
| Meanstd\_0.2 | 93.20% | 92.80% |
| Meanstd\_0.5 | 93.20% | 92.70% |
| Meanstd\_1.0 | 93.16% | 92.80% |

选取最大值、最小值进行归一化时正确率最高的学习率0.1的数据，进行loss曲线绘制，选取平均数和标准差进行归一化时正确率最高的学习率0.01的数据，进行loss曲线绘制。因为在2000次迭代之后，loss函数的变化可以忽略不计，因此只取前2000次的迭代数据。

手机屏幕截图

描述已自动生成

**采用最大值、最小值进行归一化时的loss函数**

手机屏幕截图

描述已自动生成

**采用平均值和标准差进行归一化时的loss函数**

在最终的数据上，采用最大值、最小值进行归一化时的loss函数的值为0.22，采用平均值和标准差进行归一化时的loss函数的值为0.19，两者比较接近，并且loss函数是递增的，符合预期。

同时，根据得到的数据，得到不同学习率的选择对训练误差收敛速度的影响，并绘制得到misclassification rate曲线。

手机屏幕截图

描述已自动生成

# 结论

经过测试，选用平均值和标准差时，有着更高的准确率和效率。同时可以看到随着学习率的增加，正确率先出现了上升，达到峰值后会下降，之后出现下降的原因应该是出现了过拟合，因为在训练集的数据变化并不大。

同时因为学习率其实代表的是每次迭代在向量空间中朝目标点移动的步长，我们通过收敛速度可以看出，选择较小的的学习率，其收敛速度较慢，因为需要更多步才能移动到目标点。而选择较大的学习率可以快速逼近目标点，收敛速度变快，但是当学习率进一步增大时，也会带来一个新的问题，就是有可能会跳过目标点，而进入了一个局部最优点，在实验中我们可以看到，当学习率变得很大时，正确率反而出现了下降。

而对于损失函数，我们可以看到随着迭代次数的增加，我们构造的逻辑回归函数会更贴近于真实数据，因此损失函数会逐渐下降。

通过本次试验可以得出对于个人工资预测，最好使用平均值和标准差进行归一化，对于该数据来说，最佳学习率为0.01。