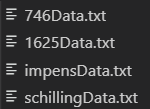
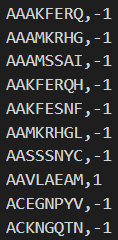
逻辑斯蒂回归

1. 思路
2. 处理数据
3. 分析数据

首先拿到数据集，分析得共四个数据集，



每个数据集包括两列，第一列是8个字符组成的名称，第二列是标签

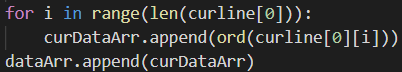


1. 处理名称

这个数据很规范，长度一定，为8位，又是只有字母组成，所以我的第一选择是把字符串看成数组，然后把每个字母用其ASCII码表示，这样就得到一个长为8的纯数字数组。

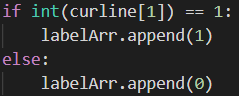
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| A | A | A | K | F | E | R | Q |
| 65 | 65 | 65 | 75 | 70 | 69 | 82 | 81 |

则从字符串AAAKFERQ得数组[65,65,65,75,70,69,82,81]。



1. 处理标签

考虑到逻辑斯蒂回归函数图像，我把标签1处理为数字1，标签-1处理为0



1. 处理结果

“AAAKFERQ，-1“就成了

1. 逻辑斯蒂回归函数

初始化w数组为[0]\*9

写一个循环来做梯度上升

循环结束后返回w，这就相当于是训练好了w

1. 测试函数

其实就是测试算出来的结果跟实际结果是不是一样的

1. 实验
2. 训练数据与测试数据的选择

一共四个数据集，拿一个做训练集，另外三个做测试集，测试结果准确率

|  |  |  |  |
| --- | --- | --- | --- |
| 746Data.txt | 1625Data.txt | impensData.txt | schillingData.txt |
| 训练集 | 76.9231 | 84.2661 | 86.7359 |
| 53.8874 | 训练集 | 15.7339 | 13.2641 |
| 49.8660 | 28.6769 | 训练集 | 26.2531 |
| 46.1126 | 76.9231 | 84.2661 | 训练集 |