中国地质大学(武汉)

机器学习第一次作业

姓 名：叶宇涛

专 业：计算机科学与技术

学 号：20191000595

指导老师：刘超

目录

[[3.3]实现对率回归，给出西瓜数据集3.0a结果。 3](#_Toc98670329)

[编程题目理解 3](#_Toc98670330)

[对率回归算法原理阐述 4](#_Toc98670331)

[算法设计思路 5](#_Toc98670332)

[实验流程、测试结果及分析 6](#_Toc98670333)

[代码结构，核心代码简要分析 7](#_Toc98670334)

[本次实验解决的主要问题，主要收获 8](#_Toc98670335)

[编码及内容撰写中的参考来源 8](#_Toc98670336)

# [3.3]实现对率回归，给出西瓜数据集3.0a结果。

## 编程题目理解

对率回归，是一种广义的线性模型，只是将线性回归方程中的y换成了ln(p/1-p),其中，p为p(y=1/x)。虽然是回归的方式，但是是用来做分类任务的，所以需要找一个单调可微函数，将分类任务的真实标记和线性回归模型的预测值联系起来。

和回归方法的方式一样，都是通过训练损失函数，得到最好的参数，带入到机器学习模型中。因此，通过划分西瓜数据集3.0α中的数据为训练集、测试集，计算查全率、F1-score，来对比出模型效果的好坏。

给出数据集如下：

表格

描述已自动生成

图 1 西瓜数据集3.0α

## 对率回归算法原理阐述

对率回归算法实际上是分类算法，利用线性模型函数，再代入sigmoid函数中计算出y的值从而判断分类的类别*。*因此，对数几率回归其实是广义线性模型的特例，原始线性模型公式为：

通过sigmoid函数将z值单位化，即转化为(0,1)区间上面的y值。其中，sigmoid函数如下：

将线性回归方程带入sigmoid激活函数中。其中，通过极大似然法估计线性模型的。得到的最后结果如下：

文本, 信件

描述已自动生成

其中,求解该最大化函数可用梯度下降算法或者牛顿法求解非线性函数的解。

## 算法设计思路

以这道题为例子，首先需要读取西瓜数据集3.0α，利用pandas库读取十分方便。将好瓜、坏瓜单独设置符号，画出散点图。

因此，首先需要设计激活函数sigmoid，根据书上面的公式对结果进行梯度下降，得到ω和β。根据参数，画出根据以密度为x轴，含糖率为y轴的直线，对比效果。对x进行预测，得到的y进行激活函数的转化，如果结果大于0.5，判定为好瓜，如果小于0.5，判定为坏瓜，根据原始y值，计算精确度。

## 实验流程、测试结果及分析

图示

描述已自动生成

图 2 对率回归流程

计算得到：



图 3 准确率

图表, 散点图

描述已自动生成

图 4 含糖率、密度图

其中，红点表示好瓜，蓝点表示坏瓜，直线表示对率回归拟合直线。在直线上面的点判定为好瓜，在直线下方的点判定为坏瓜。

## 代码结构，核心代码简要分析

定义激活函数，返回的值用于二分类。

def sigmoid(self,z):

'''

@param z:beta \* xi

'''

return 1/(1 + np.exp(-z))

定义梯度下降法，将参数值初始化为(0.1,0.1,0.1)，计算线性方程的z值，进行梯度下降优化参数，返回得到的参数值。

    def gradient\_descent(self,xtrain,ytrain):

        '''

        梯度下降法求解beta

        @param xtrain:(x,1) shape[N,d+1]

        @param ytrain:label shape[N,1]

        @return beta (w,b) shape [1,d+1]

        '''

        beta = np.ones((1,3)) \* 0.1

        z = np.dot(xtrain,beta.T)

        learn\_rate = 0.05

        iter\_max = 2000

        for i in range(iter\_max):

            p1 = np.exp(z) / (1 + np.exp(z))

            #shape [N, N]

            p = np.diag((p1 \* (1-p1)).reshape(-1))#生成对角阵

            #shape [N, 1] 一阶导数

            dl1 = -np.sum(xtrain \* (ytrain - p1), 0, keepdims=True) #按列相加，保持矩阵的二维性

            beta -= dl1 \* learn\_rate

            z = np.dot(xtrain,beta.T)

        return beta

最后，根据得到的参数值计算y，对y进行激活函数映射，结果大于0.5判定为好瓜，结果小于0.5判定为坏瓜。对比原来的y值，相同说明正确。最后计算得到的准确率。

print('准确率', sum(ytrain == ypredict)/len(ytrain))

## 本次实验解决的主要问题，主要收获

本次实验让我了解到了线性回归用于分类问题要如何进行。对于该问题，首先需要定义激活函数，用于分类，再对最大化函数进行梯度下降，得到参数值。最后计算准确率。

遇到的问题主要在于不熟悉numpy和pandas的操作，需要对数据集的形态进行反复的确认，不然会出现矩阵大小不匹配的情况。并且，对于公式的编码也需要反复debug测验数据。

收获在于了解了对率回归算法的实现，可以自己手动实现算法，了解了激活函数在分类问题中的作用。并且，对公式的推导更加熟悉于心。对于matplotlib库的应用更加熟悉，可以画出符合要求的图像。

## 编码及内容撰写中的参考来源

1. <https://zhuanlan.zhihu.com/p/27585172>，机器学习：对数几率回归
2. https://zhuanlan.zhihu.com/p/259848053, 对数几率回归
3. https://blog.csdn.net/llwleon/article/details/79204790, 《机器学习》周志华 课后习题3.3：编程实现对率回归,并给出西瓜数据集 3.0α 上的结果.

# [3.4]选择iris数据集，比较10折交叉验证和留一法的错误率。

## 编程题目理解

## 对率回归算法原理阐述

## 算法设计思路

## 实验流程、测试结果及分析

## 代码结构，核心代码简要分析

## 本次实验解决的主要问题，主要收获

## 编码及内容撰写中的参考来源

# [3.5]实现线性判别分析，给出西瓜数据集3.0a的结果。

## 编程题目理解

## 对率回归算法原理阐述

## 算法设计思路

## 实验流程、测试结果及分析

## 代码结构，核心代码简要分析

## 本次实验解决的主要问题，主要收获

## 编码及内容撰写中的参考来源