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。 4](#_Toc98692367)

[编程题目理解 4](#_Toc98692368)

[对率回归算法原理阐述 5](#_Toc98692369)

[算法设计思路 6](#_Toc98692370)

[实验流程、测试结果及分析 6](#_Toc98692371)

[代码结构，核心代码简要分析 7](#_Toc98692372)

[本次实验解决的主要问题，主要收获 9](#_Toc98692373)

[编码及内容撰写中的参考来源 9](#_Toc98692374)

[[3.4]选择iris数据集，比较10折交叉验证和留一法的错误率。 10](#_Toc98692375)

[编程题目理解 10](#_Toc98692376)

[对率回归算法原理阐述 10](#_Toc98692377)

[十折交叉验证、留一法算法设计思路 11](#_Toc98692378)

[实验流程、测试结果及分析 11](#_Toc98692379)

[代码结构，核心代码简要分析 13](#_Toc98692380)

[本次实验解决的主要问题，主要收获 13](#_Toc98692381)

[编码及内容撰写中的参考来源 13](#_Toc98692382)

[[3.5]实现线性判别分析，给出西瓜数据集3.0a的结果 14](#_Toc98692383)

[编程题目理解 14](#_Toc98692384)

[线性判别分析算法原理阐述 14](#_Toc98692385)

[算法设计思路 15](#_Toc98692386)

[实验流程、测试结果及分析 16](#_Toc98692387)

[代码结构，核心代码简要分析 17](#_Toc98692388)

[本次实验解决的主要问题，主要收获 18](#_Toc98692389)

[编码及内容撰写中的参考来源 19](#_Toc98692390)

# [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

需要对于iris数据集，比较十折交叉验证和留一法的错误率。对于模型，可以选用对率回归或者其他方法进行验证。查看iris数据集，发现要预测的species存在三种数据，而对率回归只能够完成二分类任务。如果需要完成多分类任务，需要指定分类方式：ovr即one-vs-rest(OvR)，multinomial是many-vs-many(MvM)。由于对于速度没有要求，选择multinomial进行分类。题目中的错误率是一种指标，公式如下：

## 对率回归算法原理阐述

十折交叉验证是一种用来测试算法准确性常用的算法，将数据集分成十份，轮流将其中9份作为训练数据，1份作为测试数据，进行试验。每次实验会得出相应的正确率，将十次的正确率的平均值作为对算法精度的估计。本题目中的错误率实际上就是1-正确率。

留一法也是同样的评估方法，实际上是交叉验证法的特例。每次都只留下一个样本作为测试集，剩下的样本作为训练集。最后对该样本进行预估。比如对于iris数据集，将149个样本作为训练集，将1个样本作为测试集，对模型训练完之后，对该样本进行预测测试。重复进行150次。这种方法一般比较准确，但是训练的时间比较长，时间开销大对于数据量大的数据集是无法忍受的。

## 十折交叉验证、留一法算法设计思路

将iris数据集导入，画出每个特征之间的关系，查看每个特征对于分类的影响。导入sklearn中的对率回归模型、十折交叉验证以及留一法。对于十折交叉验证，采用cross\_val\_predict()函数对模型进行训练，计算出准确率。对于留一法，分150次划分训练集，测试集，通过拟合模型，对剩下的一个样本进行测试，如果分类错误，错误数量+1.最后，计算出错误率。

## 实验流程、测试结果及分析

图示

描述已自动生成

图 5 3.4实验流程

图表

描述已自动生成

图 6 iris数据集各个特征之间关系

图形用户界面, 文本, 应用程序, Word, 电子邮件

描述已自动生成

图 7 十折交叉验证的错误率

图形用户界面, 文本

描述已自动生成

图 8 留一法错误率

对比可以看出，留一法的错误率比较高，而留一法一般来说，准确率会高一点，可以相信该指标对于该模型的评估作用。但是留一法的性能消耗太大，一般还是采用十折交叉验证进行评估。

## 代码结构，核心代码简要分析

导入数据集，画出特征之间的关系：

sns.set(style="white", color\_codes=True)

iris = sns.load\_dataset('iris')#导入数据集

sns.pairplot(iris,hue='species') #画出散点图

进行十折交叉验证，计算错误率。

y\_predict = cross\_val\_predict(lr,X,y,cv=10)#交叉验证

print('十折交叉验证错误率:',1-metrics.accuracy\_score(y,y\_predict))#错误率

进行留一法验证，计算错误率。

accuracy = 0  # 初始化

for train, test in loo.split(X):  # 每次取一个样本

    X\_train, y\_train = X.iloc[train, :], y.iloc[train]  # 训练集

    X\_test = X.iloc[test, :]  # 测试集

    lr.fit(X\_train, y\_train)  # 训练

    y\_predict = lr.predict(X\_test)  # 预测

    if y.iloc[test] in y\_predict:  # 如果预测结果和测试集一样，则正确率加一

        accuracy += 1

print('留一法错误率：', 1-accuracy / X.shape[0])  # 错误率

## 本次实验解决的主要问题，主要收获

本次实验主要解决的问题是逻辑回归对于多分类问题应该如何处理。并且，对于十折交叉验证和留一法相关的数据输入需要特别注意，不然会导致报错。主要收获是了解了十折交叉验证和留一法的基本原理，了解了错误率的计算公式，可以利用sklearn中的相关函数对iris数据集进行测试，对比错误率。

## 编码及内容撰写中的参考来源

1. https://blog.csdn.net/weixin\_44350982/article/details/102667884, 周志华《机器学习》课后习题3.410折交叉验证法和留一法UCI实例比较
2. <https://baike.baidu.com/item/%E7%95%99%E4%B8%80%E6%B3%95/22499706>,留一法简介
3. <https://baike.baidu.com/item/%E5%8D%81%E6%8A%98%E4%BA%A4%E5%8F%89%E9%AA%8C%E8%AF%81>,十折交叉验证简介

# [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

线性判别分析是一种经典的线性学习方法，本题目要求调用该算法对西瓜数据集3.0α中的数据进行分析，得出结果。因此，需要进行该方法的实现，并且根据该方法对西瓜数据集进行判别，画出相应图像。因此，需要导入西瓜数据集，并且根据书上面的公式拟合直线。

## 线性判别分析算法原理阐述

线性判别分析要求将样本投影到一条直线上，并且，要求同类样本的距离尽可能近，不同样本尽可能远。对于新的样本，进行投影之后，根据投影点的位置确定新样本的类别。其中，直线的方程为：

为了使同类样本距离尽可能近，不同样本距离尽可能远，给出了最大化目标：

其中，利用拉格朗日乘子法有：

最后得到的公式如下，其中为平均值。

文本, 信件

描述已自动生成

我们的目标就是求出ω.有了公式，求代码就会变得比较简单。

## 算法设计思路

首先输入数据集，定义LDA函数，划分X\_0,X\_1为坏瓜，好瓜，根据坏瓜，好瓜样本的平均值，协方差，求出，根据以及求出最后的直线参数，我们要求的直线斜率可以根据的值进行求解。

然后，根据原来X\_0,X\_1的值，根据映射公式，映射到直线上，画出新映射点。

最后，画出原来的X\_0,X\_1值，直线.

## 实验流程、测试结果及分析

图示

描述已自动生成

图 9 3.5实验流程

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图 10 ω计算结果

图表, 散点图

描述已自动生成

图 11 好瓜、坏瓜、映射点、投影直线

可以看出，效果不是很好，好瓜、坏瓜都粘合在了一起，好瓜与坏瓜的投影点不够远离，坏瓜与坏瓜之间的投影点不够聚集。原因可能是数据集的数量太少，不能很好的拟合直线。

## 代码结构，核心代码简要分析

读取数据，划分好瓜、坏瓜

data = pd.read\_csv('3.0a.csv')  # 读取数据

X\_0 = np.array(data[data['好瓜'].isin([0])].iloc[:, 1:3],

               dtype='float')  # 取出坏瓜的数据

X\_1 = np.array(data[data['好瓜'].isin([1])].iloc[:, 1:3],

               dtype='float')  # 取出好瓜的数据

X\_0, X\_1  # 查看数据

输入X0,X1，求出均值、协方差，得出w。

def LDA(X0, X1):

    miu0 = np.mean(X0, axis=0).reshape((-1, 1))  # 均值miu0

    miu1 = np.mean(X1, axis=0).reshape((-1, 1))  # 均值miu1

    # 求协方差

    cov0 = np.cov(X0, rowvar=False)  # 协方差X0

    cov1 = np.cov(X1, rowvar=False)  # 协方差X1

    # 求出w

    S\_w = np.mat(cov0 + cov1)  # sigma\_w

    Omiga = S\_w.I \* (miu0 - miu1)  # 协方差矩阵的逆乘以均值差

return Omiga  # 返回w

映射函数，将原来的X映射到直线上，返回新点的坐标。

def mapping(w, X):#映射函数

    k = -w[0] / w[1]#求出k

    new\_X = ((k \* X[:, 1] + X[:, 0]) / (k \* k + 1)).reshape(1, X.shape[0])#求出新的X

    new\_X = np.insert(new\_X, values=k \* new\_X, obj=1, axis=0)# 插入y值

return new\_X.T#返回新的X

画图，画出直线、映射前的点、映射后的点。

ax.plot([0, 1], [0, -w[0] / w[1]], label='y=wx')  # 绘制y=wx

ax.scatter(new\_yes[:, 0].tolist(), new\_yes[:, 1].tolist(), marker='^',

           c='b', label='new\_Yes')  # 画出新的好瓜

ax.scatter(new\_no[:, 0].tolist(), new\_no[:, 1].tolist(), marker='\*',

           c='r', label='new\_No')  # 画出新的坏瓜

ax.scatter(yes['密度'], yes['含糖率'], marker='o', c='b', label='Yes')  # 画出好瓜

ax.scatter(no['密度'], no['含糖率'], marker='x', c='r', label='No')  # 画出坏瓜

## 本次实验解决的主要问题，主要收获

这次实验主要收获是了解了线性判别分析的算法原理，可以自己推导、计算得到最后的公式。并且，更加熟悉了numpy中的matrix数据结构的用法，对matplotlib的理解更加深刻。了解了numpy中常见的函数用法。了解了线性模型具有一定的局限性，对于圆圈数据不能很好的划分。

主要的问题是对于scatter的函数用法不太熟悉，需要反复确认验证数据，查阅资料，才可以画出图像。

## 编码及内容撰写中的参考来源

1. <https://blog.csdn.net/A993852/article/details/80099258>，周志华《机器学习》3.5答案-编程实现线性判别分析，并给出西瓜数据集3.0α上的结果
2. 周志华 机器学习[M] 清华大学出版社 2016 ：P60
3. 周志华 机器学习[M] 清华大学出版社 2016：P61
4. 周志华 机器学习[M] 清华大学出版社 2016：P62
5. 周志华 机器学习[M] 清华大学出版社 2016：P29-30