《机器学习基础》实验报告

实验题目	对数几率回归算法实践		
实验时间	2024/04/07	实验地点	DS3401

一、实验目的

掌握线性模型、对率回归算法原理。

二、实验项目内容

1. 理解对率回归算法原理。
2. 编程实现对数几率回归算法。
3. 将算法应用于西瓜数据集、鸢尾花数据集或其它数据集分类问题。

三、实验过程或算法（源程序）

```
import csv
import numpy as np
from sklearn.model_selection import train_test_split

#读取西瓜数据集中的数据并返回
def loadDataset1(filename):
    dataset = []
    labelset = []
    with open(filename, 'r', encoding='utf-8') as csvfile:
        csv_reader = csv.reader(csvfile)
        header = next(csv_reader)
        for row in csv_reader:
            data_row = [float(row[1]), float(row[2])]
            dataset.append(data_row)
            labelset.append(int(row[3]))
    return dataset, labelset

#读取鸢尾花数据集中的数据并返回（2 和 2p2 分别对应两种分类方式）
def loadDataset2(filename):
    dataset = []
    labelset = []
    with open(filename, 'r', encoding='utf-8') as csvfile:
        csv_reader = csv.reader(csvfile)
        header = next(csv_reader)
        for row in csv_reader:
            data_row = [float(row[1]), float(row[2]),float(row[3]),float(row[4])]
            dataset.append(data_row)
            if row[5]=='Iris-setosa' or row[5]=='Iris-versicolor':
```

```
                labelset.append(1)
            else:
                labelset.append(0)
    return dataset, labelset

def loadDataset2p2(filename):
    dataset = []
    labelset = []
    with open(filename, 'r', encoding='utf-8') as csvfile:
        csv_reader = csv.reader(csvfile)
        header = next(csv_reader)
        for row in csv_reader:
            data_row = [float(row[1]), float(row[2]),float(row[3]),float(row[4])]
            if row[5]=='Iris-setosa':
                labelset.append(1)
                dataset.append(data_row)
            elif row[5]=='Iris-versicolor':
                labelset.append(0)
                dataset.append(data_row)
    return dataset, labelset

#定义 sigmoid 函数
def sigmoid(z):
 return 1.0 / (1 + np.exp(-z))

#计算准确率
def test(dataset,labelset,w):
    data=np.mat(dataset)

    y=sigmoid(np.dot(data,w))
    b,c=np.shape(y)
    #功能是查看矩阵或者数组的维数。
    rightcount=0

    for i in range(b):
        flag=-1
        if y[i,0]>0.5:
            flag=1
        elif y[i,0]<0.5:
            flag=0
        if labelset[i] == flag:
            rightcount+=1

    rightrate=rightcount/len(dataset)
    #print(rightrate)
    return rightrate

```

```
#迭代求w
def training(dataset,labelset,highw):
    data=np.mat(dataset).astype(float)
    label=np.mat(labelset).transpose()
    w = np.ones((len(dataset[0]),1))

    #步长
    n=0.0001

    # 每次迭代计算一次正确率（在测试集上的正确率）
    # 达到highw的正确率，停止迭代
    rightrate=0.0
    while rightrate<highw:
        c=sigmoid(np.dot(data,w))
        b=c-label
        change = np.dot(np.transpose(data),b)
        w=w-change*n
        #预测，更新准确率
        rightrate = test(dataset,labelset,w)
    return w

#西瓜
dataset=[]
labelset=[]
filename = 'D:\zjw\demo\machine learning\watermelon_3a.csv'
dataset,labelset=loadDataset1(filename)
X_train, X_test, y_train, y_test = train_test_split(dataset, labelset, test_size=0.1, random_state=42)
w=training(X_train,y_train,0.90)
print("西瓜数据集：")
print("若使得准确率大于90%，则此时的w为：\n",w)
accuracy = test(X_test, y_test, w)
print("测试集上的准确率：%f" % (accuracy * 100) + "%")

#鸢尾花
dataset=[]
labelset=[]
filename = 'D:\zjw\demo\machine learning\Iris-data.csv'
dataset,labelset=loadDataset2(filename)
X_train, X_test, y_train, y_test = train_test_split(dataset, labelset, test_size=0.2, random_state=42)
w1=training(X_train,y_train,0.95)
print("鸢尾花数据集：")
print("对于第一次分类，准确率大于0.95，则此时的w为：\n",w1)
r1= test(X_test, y_test, w1)
print("测试集上的准确率为:%f"%(r1*100)+"%")

dataset,labelset=loadDataset2p2(filename)
```

```
X_train, X_test, y_train, y_test = train_test_split(dataset, la
belset, test_size=0.2, random_state=42)
w2=training(X_train,y_train,0.95)
print("对于第二次分类，准确率大于 0.95，则此时的 w 为：\n",w2)
r2 = test(X_test, y_test, w2)
print("测试集上的准确率为:%f"%(r2*100)+"%")
```

四、实验结果及分析

1.控制台输出：

```
西瓜数据集：
若使得准确率大于90%，则此时的w为：
 [[-0.27394248]
 [ 0.90898622]]
测试集上的准确率： 100.000000%
鸢尾花数据集：
对于第一次分类，准确率大于0.95，则此时的w为：
 [[ 0.22718722]
 [ 0.93813229]
 [-0.86834889]
 [ 0.06346749]]
测试集上的准确率为:86.666667%
对于第二次分类，准确率大于0.95，则此时的w为：
 [[-0.28867293]
 [ 0.61324918]
 [-0.35930641]
 [ 0.5280983 ]]
测试集上的准确率为:100.000000%
```

2.实验分析

（1）对数几率算法原理

对数几率回归是一种广泛应用于二分类问题的统计学习方法。它通过假设数据服从伯努利分布，并利用 Sigmoid 函数将线性模型的输出转化为样本属于某一类别的概率。具体来说，对数几率回归计算给定输入特征下样本属于正类的概率，并基于这个概率和真实标签构建损失函数。通过最小化损失函数，模型可以学习到最优的权重和偏置项，迭代得到最优解。

（2）程序结果分析

鸢尾花数据集多分类问题处理：将三个类进行多次分类，以二叉树的形式对进行两次分类，每次分类就是一次单独的分类；

准确率偏高：可能是测试集样本数量较小，仅有的几个测试用例全部预测成功，同时也说明了模型训练效果好。