**天津工业大学**

**机器学习实验报告**

**实验课程名称：使用 Python 实现对数几率回归模型**

**学生姓名： 王丽娜**

**学号： 1611640416**

**班级： 计算机1604**

**实验时间： 2018年10月10日**

**一、实验题目**

利用 Python 实现对数几率回归模型，并对 Iris 数据集进行分类以验证模型。

1. **问题描述**

利用numpy和pandas和matplotlib.pyplot库对鸢尾花数据进行数据分析。选取鸢尾花数据集前100个数据作为此次实验数据。

程序需要完成的模块：

sigmoid:映射到概率的函数

model:返回预测结果值

cost：根据参数计算损失

gradient：计算每个参数的梯度方向

descent：进行参数更新

accuracy：计算精度

sigmoid函数：g(z)=1/(1+e^-z)

def sigmoid(z):  
 return 1/(1+np.exp(-z))

model函数：插入一列值都为1的数，把数值运算转变为矩阵运算，x和theta相乘的值给T在返回给sigmoid函数

def model(x,theta):  
 return sigmoid(np.dot(x, theta.T))

cost函数：损失函数

将对数似然函数去负号



求平均损失



def cost(x,y,theta):  
 left=np.multiply(-y,np.log(model(x,theta)))  
 right=np.multiply(1-y,np.log(1-model(x,theta)))  
 return np.sum(left-right)/(len(x))

gradient函数：计算梯度



def gradient ( x , y , theta):  
 grad = np.zeros(theta.shape) #进行占位  
 error = (model(x, theta)-y).ravel()  
 for j in range(len(theta.ravel())): #对每一个theta参数求导  
 term = np.multiply(error, x[:, j])  
 grad[0, j] = np.sum(term)/len(x)  
 return grad

RunExpe：功能函数

def runExpe(data, theta, batchSize, thresh, alpha):  
 theta, iter, costs, grad, dur = decent(data, theta, batchSize, thresh, alpha)  
 print("theta:{}-Iter:{}-last cost:{:03.2f}-Duration:{:03.2f}s".format(theta, iter, costs[-1], dur))  
 fig, ax = plt.subplots(figsize=(12, 4))  
 ax.plot(np.arange(len(costs)), costs, 'r')  
 ax.set\_xlabel('Iterations')  
 ax.set\_ylabel('Cost')  
 ax.set\_title('Costs with Iteration')  
 plt.show()  
 return theta,costs  
  
n = 100#基于所有样本进行梯度下降  
runExpe(data1, theta, n, thresh=500000, alpha=0.000001)

descent：设定阈值（概率值改变成类别值）

def predict(x, theta):  
 return [1 if x >= 0.5 else 0 for x in model(x, theta)]

accuracy：计算精度

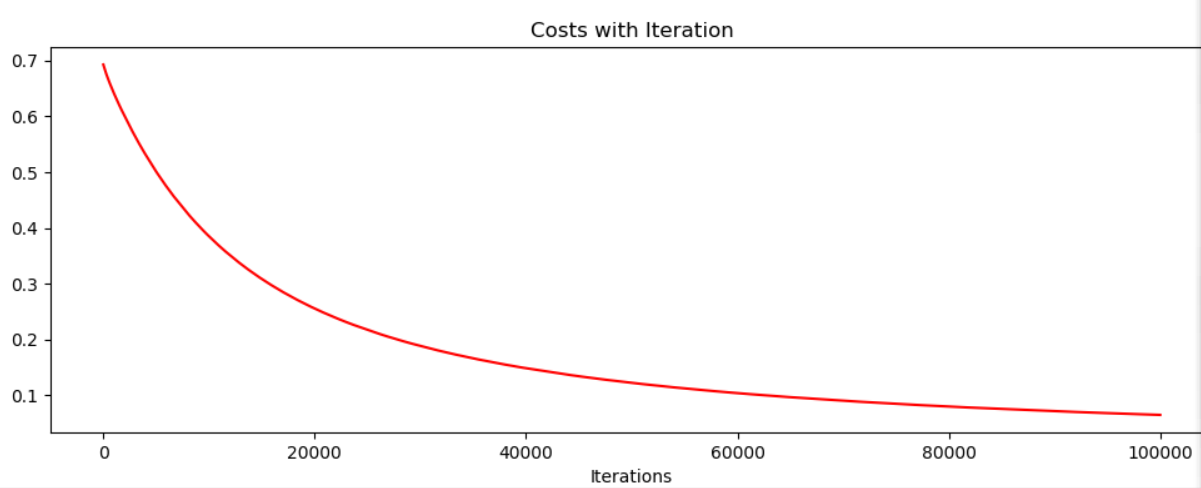
batchSize=20  
theta,costs=runExpe(data\_train, theta, batchSize, 100000, 0.0001)  
#print('MinCosts',costs[100000])  
predictions=predict(x\_exam,theta)  
correct=[1 if((a == 1 and b == 1) or (a == 0 and b == 0)) else 0 for(a, b) in zip(predictions, y\_exam)]  
accuracy=sum(map(int, correct))/len(correct)  
print('Accuracy:', accuracy) #计算准确率

**三、数据集描述**

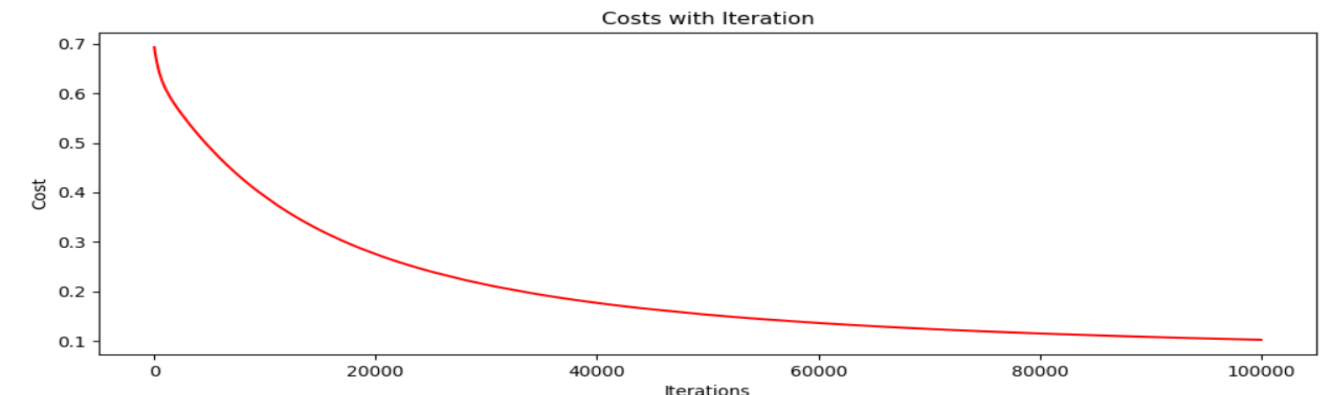
鸢尾花数据集实验通过花萼长度，花萼宽度，花瓣长度，花瓣宽度4个属性预测鸢尾花卉属于（Setosa，Versicolour，Virginica）三个种类中的哪一类。该数据集一共包含4个特征变量，1个类别变量。共有150个样本，此次实验我选取前100组样本作为实验数据，即预测鸢尾花卉属于（Setosa，Versicolou）两个种类中的哪一类。

1. **实验结果图**
2. 将数据集的 50%作为训练集，50%作为测试集，检验模型在测试集上的分类正确率

data\_train = np.vstack((data1[1:25, :],data1[76:100, :])) #人工将训练集和测试集分开  
data\_exam = np.vstack((data1[26:50, :],data1[51:75, :] ))

CCUA$9O)8HVB(R5_4T_W3O6

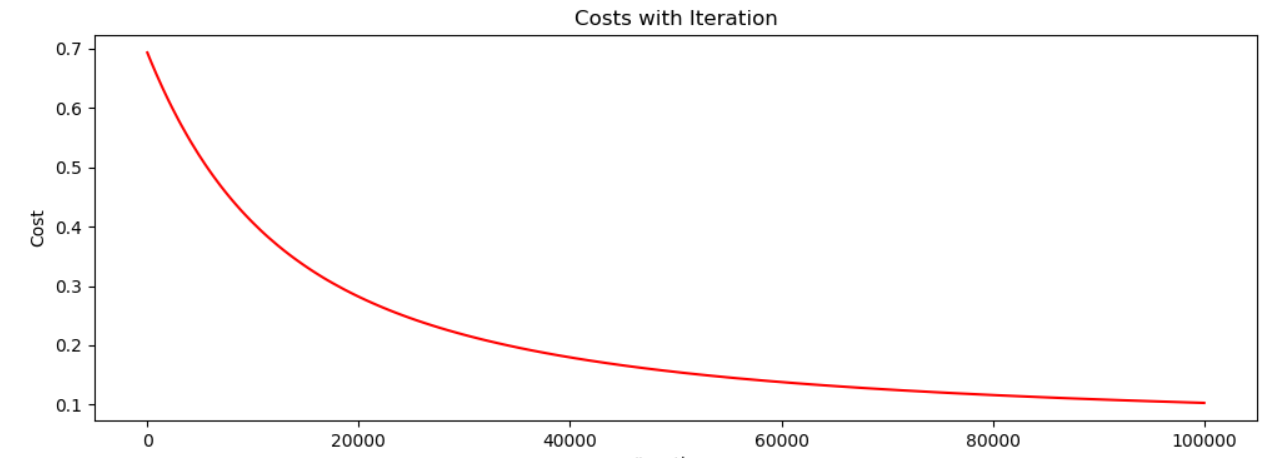
1. 将数据集的 70%作为训练集，30%作为测试集，检验模型在测试集上的分类正确率

VD%UQC`KLMLD(OM{Y$P6]RP

)63(D$@P_ZPDY0)N890D69H

1. 将数据集的 90%作为训练集，10%作为测试集，检验模型在测试集上的分类正确率

data\_train = np.vstack((data1[1:70, :],data1[81:100, :]))  
data\_exam = np.vstack((data1[71:80, :] ))

**@MS66X6MH$LGO]{ITV5GGU0**

1. **实验结果分析**

随着训练次数增加，损失函数值减小。

将数据集的 50%作为训练集，50%作为测试集时，损失值趋近于零，精度约为9.79。

将数据集的 70%作为训练集，30%作为测试集时，损失值趋近于零，精度为1。

将数据集的 90%作为训练集，10%作为测试集时，损失值趋近于零，精度为1。