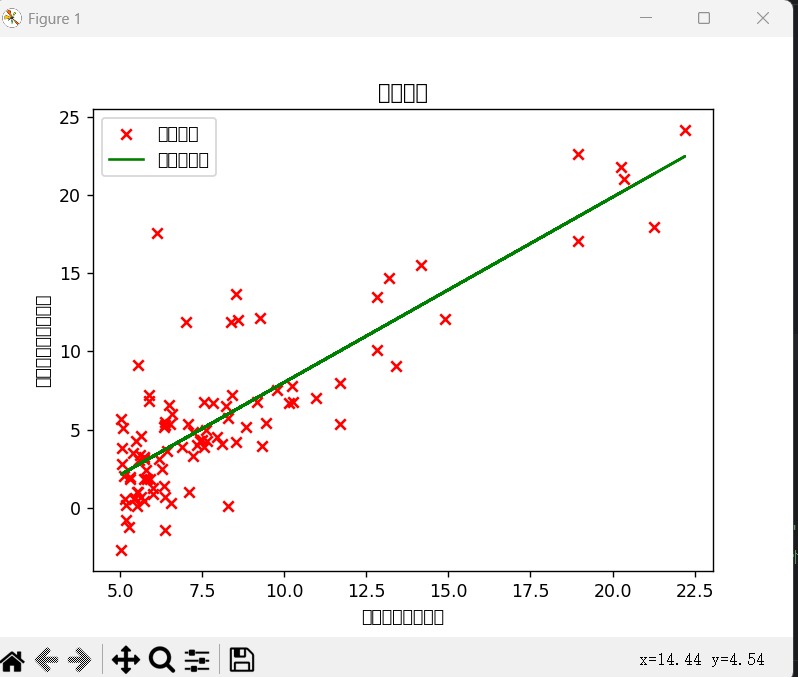
***线性回归***

**代码部分**

import numpy as np  
import matplotlib.pyplot as plt  
  
data=np.genfromtxt("D:\桌面\线性模型练习\线性回归\ex1data1.txt",delimiter=',')  
# 提取txt文件中的第一列和第二列的数据  
x=data[:,0]  
y=data[:,1]  
  
x=np.stack([np.ones(x.shape[0]),x],axis=1)  
# 初始化参数,y=ax+b  
a=0  
b=0  
  
# 定义学习率和迭代次数  
"""learning\_rate = 0.01  
iterations = 1500"""  
learning\_rate=float(input("请您输入学习率"))  
iterations=int(input("请您输入迭代次数"))  
  
# 采用梯度下降的方法  
for \_ in range(iterations):  
 # 计算得出预测值  
 prediction=x.dot([b,a])  
 # 计算得到误差  
 error=prediction-y  
 # 更新参数，同步更新  
 b-=learning\_rate \* np.sum(error) /len(x)  
 a-=learning\_rate \* np.dot(error,x[:,1]) /len(x)  
  
# 绘图  
plt.scatter(x[:,1],y,color='red',marker='x',label='训练数据')  
plt.plot(x[:,1],x.dot([b,a]),color='green',label='线性回归线')  
plt.xlabel('城市人口数（万）')  
plt.ylabel('餐馆利润（万美元）')  
plt.title('线性回归')  
plt.legend()  
plt.show()  
  
print(f"梯度下降法得到的参数：a={a}，b={b}")

**该绘图为学习率0.01，迭代次数2000的图**

****

***逻辑回归***

**代码部分**

import matplotlib.pyplot as plt  
import numpy as np  
import matplotlib.pyplot as pl  
from sklearn.linear\_model import LogisticRegression  
  
# 读取txt文件中的数据  
"""f=open("D:\桌面\线性模型练习\逻辑回归\ex2data1.txt","r",encoding="utf-8")  
x=f.read()"""  
data=np.genfromtxt('D:\桌面\线性模型练习\逻辑回归\ex2data1.txt', delimiter=',')  
  
# 从txt中读取的数据前两列为特征值向量  
x=data[:,:2]  
# 第三列为目标  
y=data[:,2]  
  
model=LogisticRegression()  
model.fit(x,y)  
  
# 绘图  
plt.scatter(x[y==0][:,0],x[y==0][:,1],color='red',label='non-admitted')  
plt.scatter(x[y==1][:,0],x[y==1][:,1],color='green',label='admitted')  
  
# 绘制决策边界  
# model.intercept\_ 和 model.coef\_:  
# model.intercept\_ 是逻辑回归模型的截距项。  
# model.coef\_ 是模型的系数数组，其中 model.coef\_[0][0] 是第一门成绩的系数，model.coef\_[0][1] 是第二门成绩的系数。  
# np.dot 函数计算两个数组的点积。在这里，它将第一门成绩的系数与x轴的范围（x\_values）相乘。  
  
x\_value=[np.min(x[:,0]),np.max(x[:,0])]  
y\_value=-(model.intercept\_+np.dot(model.coef\_[0][0],x\_value))/model.coef\_[0][1]  
plt.plot(x\_value,y\_value,label='决策边界')  
plt.xlabel('the first class-grade')  
plt.ylabel('the second class-grade')  
  
plt.show()

**实验结果部分：**

