实现LR分类器

在这个练习中，你将实现一个LR分类器，以预测司机在驾驶时是否足够警觉。

本练习数据来自Kaggle比赛，Ford公司针对多名司机进行数据收集，并对每一行数据做了标注。

**输入：**

data.csv数据包括三类：分别是车辆（V）、环境(E)和司机(P)的生理数据，每一类的数据包含多列。数据集中的第一列数据是司机是否警觉（1表示警觉）。

在这个例子中，没有区分训练数据和测试数据，所以需要使用train\_test\_split进行训练集和测试集划分。同时因为每次训练和测试数据不同，精确度也会不同。

**任务：**

1. 实现LR\_alert.py中的gradDescent，classifyVector，classifytest等函数。
2. 测试预测准确率。

**提示：**

可以参考课程中所演示代码实现逻辑回归。

另，随机森林对本数据集可以达到较高的正确率，可以尝试

**代码检查标准：**

1. 代码能否正确运行。要求没有语法和逻辑错误，同时便于他人理解。请写好相应的注释，便于老师理解。
2. 预测的正确率，请尽可能提高预测正确率。

**代码：**

**import** pandas **as** pd  
**import** numpy **as** np  
**from** sklearn.model\_selection **import** train\_test\_split  
  
iter\_num = 10  
  
**def** readData():  
 data = pd.read\_csv(**'train.csv'**)  
  
 X = data.iloc[:,1::]  
 Y = data.iloc[:,0]  
  
 Xtrain, Xtest, ytrain, ytest = train\_test\_split(X, Y, test\_size=0.20)  
  
 col\_list = [**'P1'**, **'P2'**, **'P3'**, **'P4'**, **'P6'**, **'P7'**,  
 **'E1'**, **'E2'**,**'E3'**, **'E4'**,**'E5'**, **'E6'**, **'E7'**, **'E8'**, **'E9'**,  
 **'V1'**, **'V2'**, **'V3'**, **'V4'**, **'V5'**, **'V6'**, **'V7'**, **'V8'**]  
 #对数据进行简化处理  
 **for** i **in** col\_list:  
 Xtrain[i] = (Xtrain[i] - Xtrain[i].mean())/(Xtrain[i].max() - Xtrain[i].min())  
  
 **for** i **in** col\_list:  
 Xtest[i] = (Xtest[i] - Xtest[i].mean())/(Xtest[i].max() - Xtest[i].min())  
  
 **return** Xtrain, ytrain, Xtest, ytest  
  
  
  
*# Logistic Regression***def** sigmoid(inX):  
 **return** 1.0/(1+np.exp(-inX))  
  
*# 随机梯度下降算法***def** gradDescent(dataMatrix, classLabels):  
 #实现随机梯度下降

*# 分类决策函数***def** classifyVector(inX,weights):  
 #根据随机梯度下降训练的参数，对测试数据进行预测

**def** classifytest(dataMatrix, classLabels, weights):  
 #基于classifyVector，统计预测的正确率  
  
Xtrain,ytrain,Xtest,ytest = readData()  
weights = gradAscent(np.array(Xtrain.iloc[1:,:]), np.array(ytrain.iloc[1:]))  
print(weights)  
classifytest( np.array(Xtest.iloc[1:,:]), np.array(ytest.iloc[1:]), weights)