**《机器学习与数据挖掘》实验七**

实验题目： 实现拉普拉斯修正的朴素贝叶斯分类器

实验目的： 掌握朴素贝叶斯分类器的原理及应用

实验环境（硬件和软件） Anaconda/Jupyter notebook/Pycharm

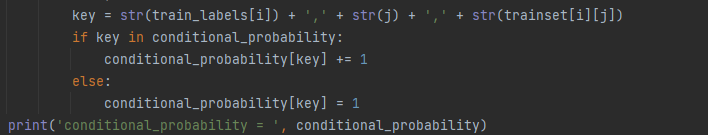
实验内容：

编码实现拉普拉斯修正的朴素贝叶斯分类器，基于给定的训练数据，对测试样本进行判别。

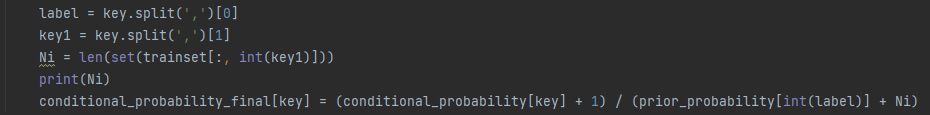
**要求：**

**一、**已经给定部分代码，补充完整的代码，需要补充代码的地方已经用红色字体标注，包括：

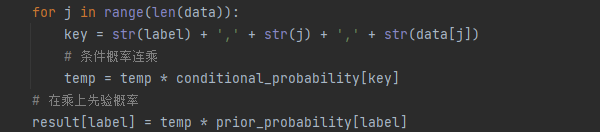
**（1）#补充计算条件概率的代码-1；**



**（2）#补充计算条件概率的代码-2；**



**（3）#补充预测代码；**



**二、**将补充完整的代码提交，并提交实验结果；（**也可以自己重写这部分的代码提交**）

import numpy as np

def loaddata():

X = np.array([[1,'S'],[1,'M'],[1,'M'],[1,'S'],

[1, 'S'], [2, 'S'], [2, 'M'], [2, 'M'],

[2, 'L'], [2, 'L'], [3, 'L'], [3, 'M'],

[3, 'M'], [3, 'L'], [3, 'L']])

y = np.array([-1,-1,1,1,-1,-1,-1,1,1,1,1,1,1,1,-1])

return X, y

def Train(trainset,train\_labels):

m = trainset.shape[0]

n = trainset.shape[1]

prior\_probability = {}# 先验概率 key是类别值，value是类别的概率值

conditional\_probability ={}# 条件概率 key的构造：类别，特征,特征值

#类别的可能取值

labels = set(train\_labels)

# 计算先验概率(此时没有除以总数据量m)

for label in labels:

prior\_probability[label] = len(train\_labels[train\_labels == label])+1

#计算条件概率

for i in range(m):

for j in range(n):

# key的构造：类别，特征,特征值

**#补充计算条件概率的代码-1；**

key = str(train\_labels[i]) + ',' + str(j) + ',' + str(trainset[i][j])  
 if key in conditional\_probability:  
 conditional\_probability[key] += 1  
 else:  
 conditional\_probability[key] = 1  
print('conditional\_probability = ', conditional\_probability)

conditional\_probability\_final = {}

for key in conditional\_probability:

**#补充计算条件概率的代码-2；**

label = key.split(',')[0]  
key1 = key.split(',')[1]  
Ni = len(set(trainset[:, int(key1)]))  
print(Ni)  
conditional\_probability\_final[key] = (conditional\_probability[key] + 1) / (prior\_probability[int(label)] + Ni)

# 最终的先验概率(此时除以总数据量m)

for label in labels:

prior\_probability[label] = prior\_probability[label]/ (m+len(labels))

return prior\_probability,conditional\_probability\_final,labels

def predict(data):

result={}

for label in train\_labels\_set:

temp=1.0

**#补充预测代码；**

for j in range(len(data)):  
 key = str(label) + ',' + str(j) + ',' + str(data[j])  
 # 条件概率连乘  
 temp = temp \* conditional\_probability[key]  
# 在乘上先验概率  
result[label] = temp \* prior\_probability[label]

print('result=',result)

#排序返回标签值

return sorted(result.items(), key=lambda x: x[1],reverse=True)[0][0]

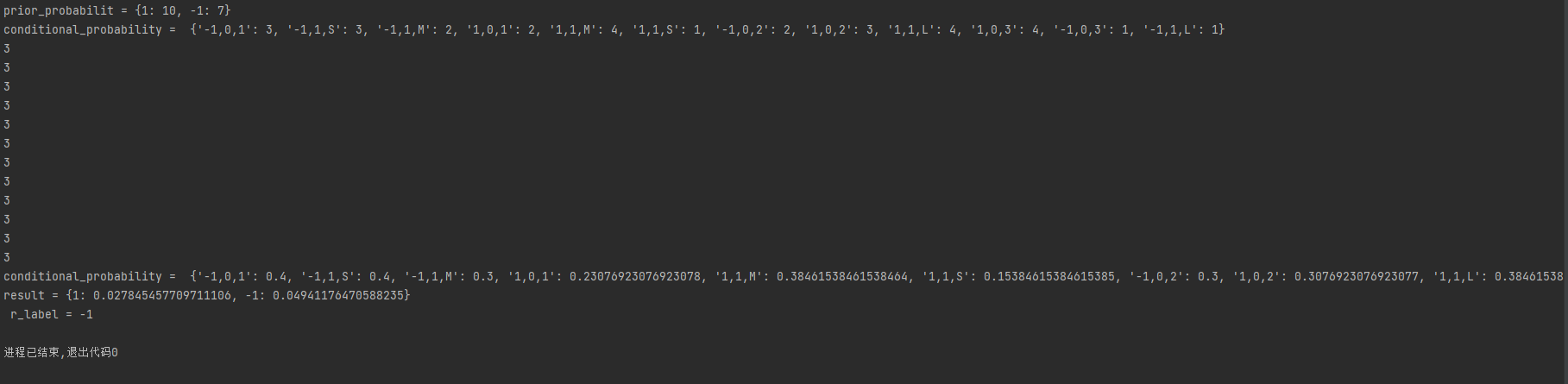
X,y = loaddata()

prior\_probability,conditional\_probability,train\_labels\_set = Train(X,y)

r\_label = predict([2,'S'])

print(' r\_label =', r\_label)

实验结果：



显示的完整内容：

prior\_probabilit = {1: 10, -1: 7}

conditional\_probability = {'-1,0,1': 3, '-1,1,S': 3, '-1,1,M': 2, '1,0,1': 2, '1,1,M': 4, '1,1,S': 1, '-1,0,2': 2, '1,0,2': 3, '1,1,L': 4, '1,0,3': 4, '-1,0,3': 1, '-1,1,L': 1}

3

3

3

3

3

3

3

3

3

3

3

3

conditional\_probability = {'-1,0,1': 0.4, '-1,1,S': 0.4, '-1,1,M': 0.3, '1,0,1': 0.23076923076923078, '1,1,M': 0.38461538461538464, '1,1,S': 0.15384615384615385, '-1,0,2': 0.3, '1,0,2': 0.3076923076923077, '1,1,L': 0.38461538461538464, '1,0,3': 0.38461538461538464, '-1,0,3': 0.2, '-1,1,L': 0.2}

result = {1: 0.027845457709711106, -1: 0.04941176470588235}

r\_label = -1