



def normData(dataSet):

maxVals = dataSet.max(axis=0)

minVals = dataSet.min(axis=0)

ranges = maxVals - minVals

retData = (dataSet - minVals) / ranges

return retData, ranges, minVals

def kNN(dataSet, labels, testData, k):

distSquareMat = (dataSet - testData) \*\* 2 # 计算差值的平方

distSquareSums = distSquareMat.sum(axis=1) # 求每一行的差值平方和

distances = distSquareSums \*\* 0.5 # 开根号，得出每个样本到测试点的距离

sortedIndices = distances.argsort() # 排序，得到排序后的下标

indices = sortedIndices[:k] # 取最小的k个

labelCount = {} # 存储每个label的出现次数

for i in indices:

label = labels[i]

labelCount[label] = labelCount.get(label, 0) + 1 # 次数加一

sortedCount = sorted(labelCount.items(), key=opt.itemgetter(1), reverse=True)

# 对label出现的次数从大到小进行排序

return sortedCount[0][0] # 返回出现次数最大的label

熵定义为信息的期望值



