**中山大学数据科学与计算机学院**

**移动信息工程专业-人工智能**

**本科生实验报告**

**（2017-2018学年秋季学期）**

课程名称：**Artificial Intelligence**

|  |  |  |  |
| --- | --- | --- | --- |
| 教学班级 |  | 专业（方向） | **移动互联网** |
| 学号 |  | 姓名 | **Jw** |

# 实验题目

**K近邻与朴素贝叶斯 -------- 分类和回归**

# 实验内容

1. 算法原理

**K近邻：**

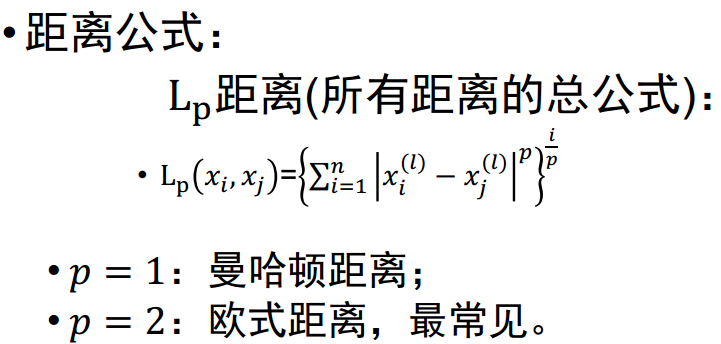
K近邻算法是数据挖掘分类与回归算法中最简单的一种。首先我们假定给出一篇包含n篇文本的训练集，我们将每个文本用一个特征向量表示，这个特征向量的每一维表示一个单词，常用的求特征向量的方法有onehot，tf，和tfidf。然后对每一个测试文本，求它与每一个训练文本的特征向量之间的距离(可以是欧式距离、曼哈顿距离或余弦距离)。在

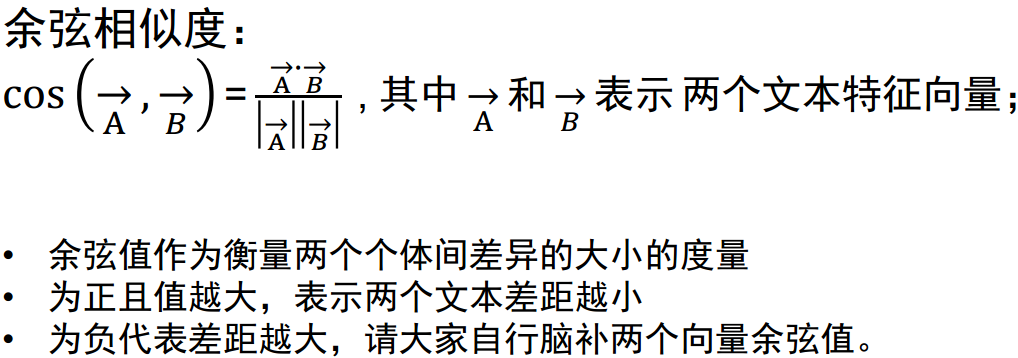
分类问题上，如果一个样本在特征空间中的K个最相邻的样本中的大多数属于某个类别，则该样本也属于这个类别。即:

the most common（{label | label belong to the Top K nearest trains}）

在回归问题上，训练集样本不是单纯地打上标签，而是知道它属于每种标签的概率(概率和为1), 那么对于每个测试集样本，取它在特征空间中的K个最相邻的样本的距离（余弦距离为正且值越大为越相近，为负且值越小为越远，因为是根据两个向量之间的夹角看的），将距离的倒数作为权重，计算属于每个标签的概率。例如:

补充知识：

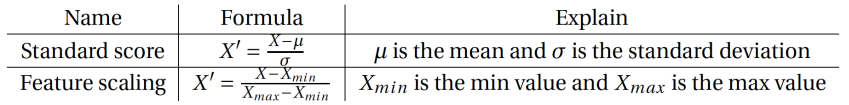




**结果影响因素：**不同的K值，一般取sqrt（样本数）以内；

不同的距离度量方式

对权值或者特征向量归一化：



**朴素贝叶斯(Naïve Bayes):**

用朴素贝叶斯去解决分类和回归问题首先要知道贝叶斯定理，贝叶斯定理主要是提供了一个算法解决这样一个问题:已知某条件概率，如何求事件交换后的条件概率，即已知P(A|B), 求P(B|A)。定理如下:

其中 P(A|B) =

上式中，P(A|B)表示事件B已经发生的前提下事件A发生的概率，P(AB)表示A和B同时发生的概率。而当A和B两个事件是互斥的，完全独立的没有交集的两件事，此时P(AB)=P(A)P(B)，这时就是朴素的

分类问题:朴素贝叶斯求解分类问题就是对于给出的待分类项，算出它属于每个标签的概率，哪个概率最大，就是哪个标签。类似KNN，首先我们要求出每个样本的特征向量x，对给定的样本特征x，样本属于类别ei的概率是

假设x的维数是K，由于是朴素的，特征条件互斥，根据全概率公式有：

由于对于同一个文本来讲，P(x)是一样的，所以预测它属于哪个分类，只要算上式的分子部分即可。对于一个给定的测试样本，分子部分常用的有两种模型：

1.伯努利模型(Bernoulli Model)：

=

其中表示情感ei出现的训练集样本数， 表示xi出现在了多少个情感为ei的样本中。N表示训练集总样本数。不难发现这个模型中x向量是一个二进制数

2.多项式模型(Multinomial Model)

=

其中，表示在训练集中，情感为ei的文章中，单词xi出现了多少次(一篇文章中出现多次按多次算),表示训练集中情感为ei的文章包含多少个单词(不去重)

回归问题: 与KNN中一样的是同样结果要输出每种情感的概率。它的特征向量为当前文本的词频，即特征空间应处理成TF矩阵。在回归问题中，同样只需要关注如何计算.设训练集中第j篇文本为dj，特征向量维数为K,则的计算方法如下:

这里，如果测试集中出现了训练集没有的单词，我们选择忽略。最后算完需要将一个测试文本的情感概率归一化，保证概率和为1

补充知识：**平滑。**当测试文本中存在一个词x，x是训练集中的单词，但是x没有出现在当前dj文本，那么会=0从而导致=0，此时，我们可以通过平滑解决。

回归：

分类：

伯努利模型：

多项式模型：

以回归的平滑为例，不难看出对于测试集x，的分母为保持不变，而分子，若count(xk)为0，则为alpha，否则为count(xk)+alpha, 这样保证了，即向量中各维所表示的单词的概率和为1，同时避免了乘上了1个概率为0的数导致最后结果为0

在上面的式子中，0<alpha<=1, 当alpha=1时，称为拉普拉斯平滑(Laplace)；当0<alpha<1时称为Lidstone平滑

1. 伪代码

**KNN 分类**

|  |
| --- |
| Step1. 读入测试集、训练集、验证集的数据  Step2. 选择进行验证集调参还是对测试集进行预测，若选择调参则进行Step3,否则  进行Step8;无论选择哪一步，均需先指定三个参数:  MatrixType(特征空间的矩阵类型)：onehot、TFIDF  Standard(特征向量类型):original(原始)、normalization(Z-Score 标准化)  Distype(距离类型)：Manhattan、Euclidean、Cosine  Step3. 根据MatrixType的到对应的矩阵，onehot的单词集合包括训练集和验证集所  有不重复的单词，TFIDF的单词集合只算训练集的单词  Step4. 一个循环尝试不同的k值，对于某个k值ki，计算测试数据与各个训练数据  之间的距离，  Step5. 排序后选取距离最小的ki个点（若是余弦距离则选取距离最大的ki个点）  Step6. 确定前ki个点所在类别的出现频率，选择ki个点中出现频率最高的类别作  为测试数据的分类  Step7 对于当前ki，求当前ki下的准确率，然后跳回Step4 继续使用下一个ki值  去做预测，若循环结束则跳到Step9  Step8 使用通过验证集得到的最优的参数K、MatrixType以及Standard去做预测，  方法更预测验证集时一样，不再累述。预测完将结果输出，不用算准确率。  Step9 程序结束 |

**KNN 回归**

|  |
| --- |
| 流程跟KNN分类时一样，其他不再累述，唯一不同的地方只有下面这两步  Step6:.使用最近的K个点去算测试数据是每种分类的概率，假设最近的K个文本的编  号为1-k则使用非余弦距离时计算公式如下：  计算时注意，距离d即分母为0，说明测试文本和训练文本一模一样，此时训练文本的各项感情的概率直接使用该训练文本的概率即可。  若使用余弦距离则公式如下：  计算时如果遇到距离d为1的，明测试文本和训练文本一模一样，此时训练文本的各项感情的概率直接使用该训练文本的概率即可。最后需要对情感各概率归一化  Step7 预测完之后，将结果输出到文件中。(使用相关性验证的excel文档去做相关性计算) |

**NB分类**

|  |
| --- |
| Step1. 读入测试集、训练集、验证集的数据，并计算p(ei)  Step2. 选择要预测的数据（测试集 or 验证集）、要计算的模型（默认多项式模型）  以及alpha值(默认1)  Step3. 根据模型先计算的分子部分 num，以及分母部分den，存在数组里  Step4. 遍历每一个需预测文本test[i]  初始化该文本所有情感概率p[j]为1  遍历每一个情感lable[j]  遍历test[i]的每一个单词word  如果word不是训练集中的词就忽略  否则根据选择的模型加上平滑计算概率  P[j] \*=  P[j] \*= p(ej)  对P作归一化保证概率和为1  Step5. 选择p中概率最大的情感输出 |

**NB回归**

|  |
| --- |
| Step1. 读入测试集、训练集、验证集的数据，并计算p(ei)  Step2. 选择要预测的数据（测试集 or 验证集）、以及alpha值(默认1)  Step3. 遍历每一个需预测文本test[i]  初始化该文本所有情感概率p[j]为0  遍历每一个情感lable[j]  遍历每一个训练文本 train[k]  temp = 1.0  遍历test[i]的每一个单词word  如果word不是训练集中的词就忽略  否则加上平滑计算概率  temp \*=  temp \*=  P[j] += temp  对P作归一化保证概率和为1  Step5. 输出所有预测文本的所有情感的概率 |

1. 关键代码截图（带注释）

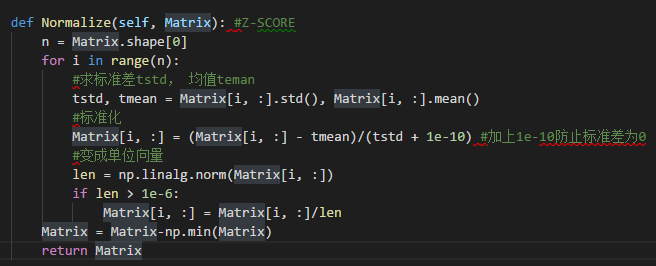
**KNN：**

1.标准化。根据公式对每行进行Z-Score 标准化x = （x-均值mean） / 标准差std

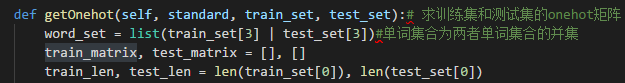
由于标准差有可能为0，这里我的解决办法是在分母加上一个极小的数字1e-10

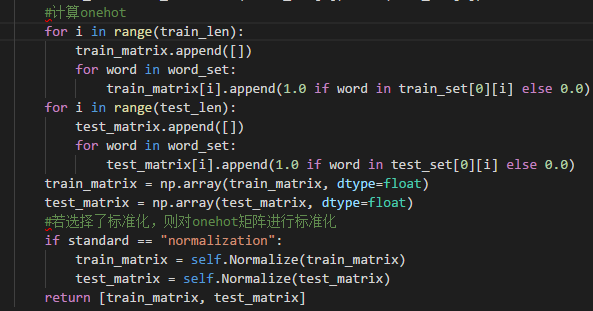
最后变成单位向量并通过将矩阵的值减去负数而变成全整数。选择Z-Score的原

因是可以将数据正太分布，使“近的更近，远的更远”。

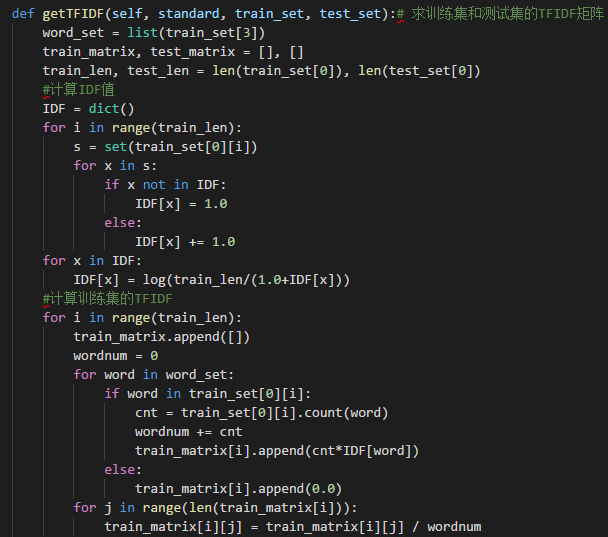


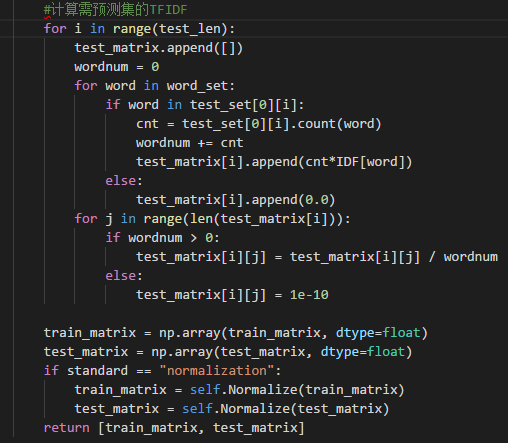
2. 计算onehot矩阵



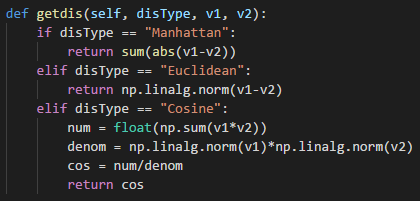


3. 计算TFIDF矩阵

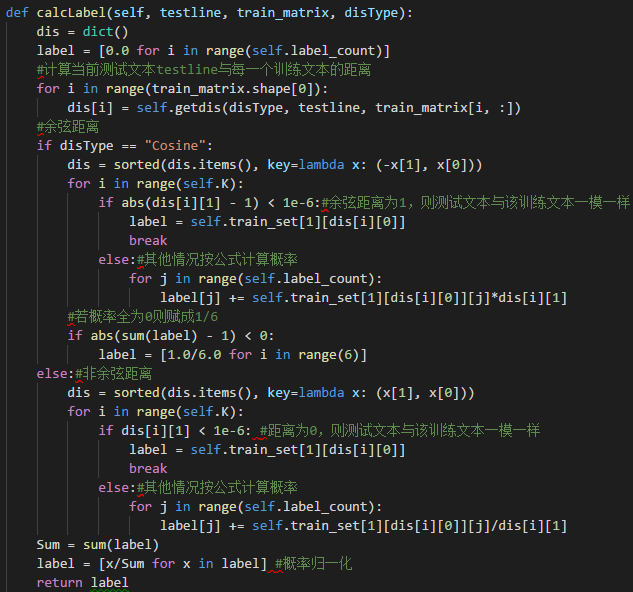




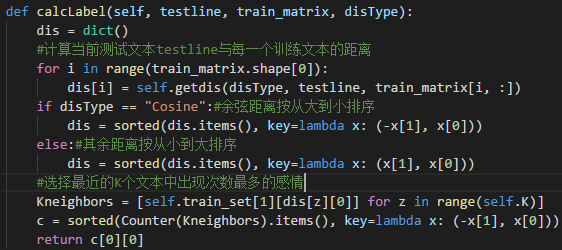
4. 计算两个向量的不同距离



5. KNN回归的概率计算

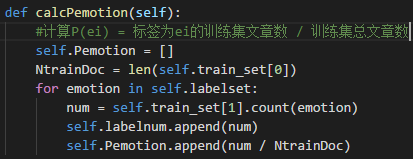


6. KNN分类的概率计算



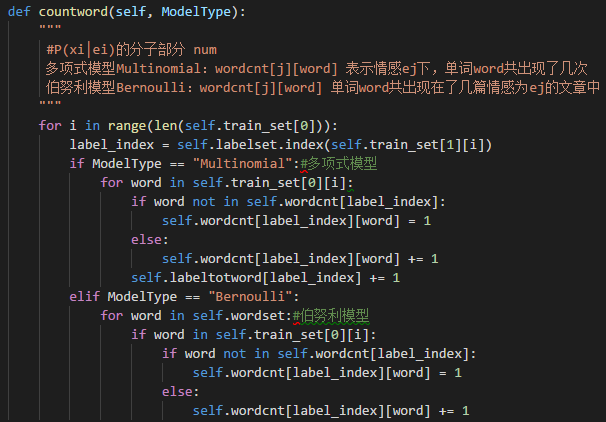
**NB分类：**

1.计算P(ei)，其中self.labelnum可以同时作为伯努利模型的分母部分

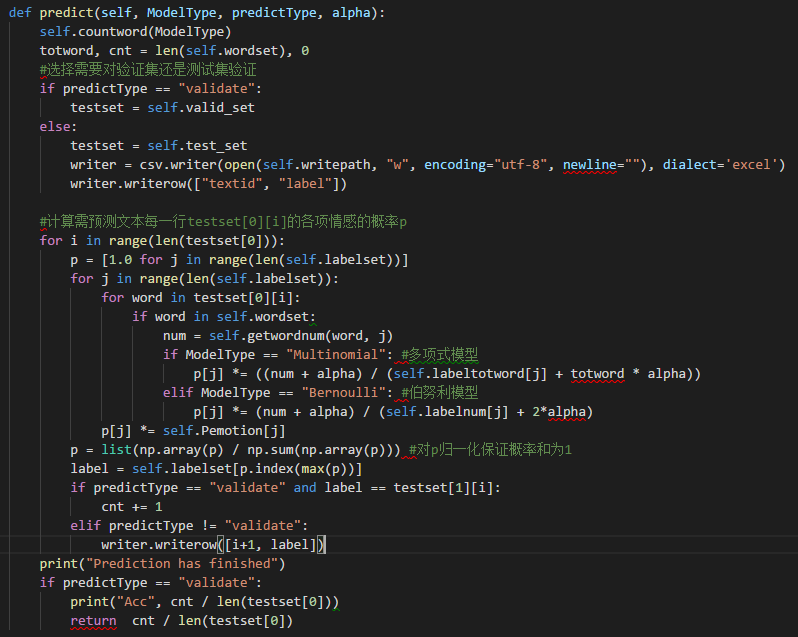


2.计算P(xi | ei) 的分子部分num，若是多项式模型，则self.labeltotword可为分母

部分

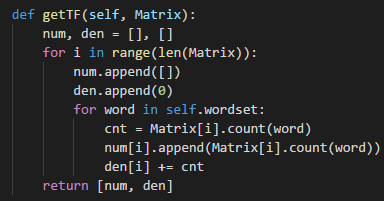


3. 分类的预测概率计算部分



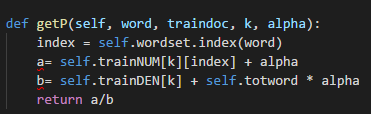
**NB回归：**

1.得到TF矩阵





2.计算



3.预测



1. 创新点&优化（如果有）

**KNN：**

1.Z – Score 标准化的时候为了防止除0，在分母加上了一个极小的数1^(-10)

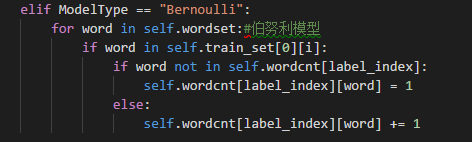
2.使用了TFIDF矩阵 + 余弦距离优化结果

3.如果出现了跟测试文本完全一样的训练文本，则直接使用它的概率。

**NB:**

1.分类实现了伯努利模型

计算伯努利模型的分子：



计算伯努利模型的



由于计算过程没有和多项式模型完全分开，在前面关键代码展示中已经全部展示过，这里就不再重复展示。

2. 通过调整alpha值来提供准确率（回归的话是相关度）

# 实验结果及分析

1. 实验结果展示示例（可图可表可文字，尽量可视化）

KNN分类：

set:

|  |  |
| --- | --- |
| Words | label |
| Good thanks | joy |
| No impressive thanks | Sad |
| Impressive good | Joy |
| No thanks | ? |

处理成onehot矩阵:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Good | Thanks | No | Impressive | Lable |
| train1 | 1 | 1 | 0 | 0 | Joy |
| train2 | 0 | 1 | 1 | 1 | Sad |
| train3 | 1 | 0 | 0 | 1 | Joy |
| test1 | 0 | 1 | 1 | 0 | ? |

取K=1，欧几里得距离跑出的结果为:



验证结果:

d(test1, train1) =

d(test1, train2) = 1

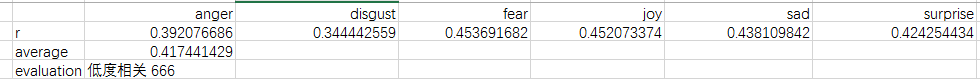
d(test1, train3) = 2

则应取第二个文本的标签，即sad，所以程序得出的结果准确

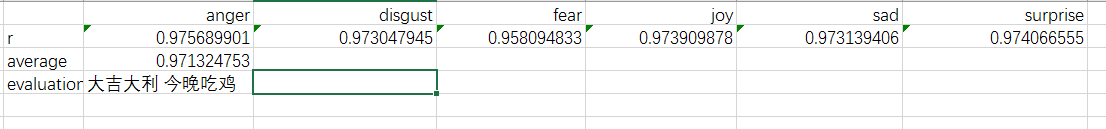
KNN回归：

跑的是实验课给的数据

使用所有优化，Z-SCORE标准化 + TFIDF + 余弦距离 + 特判完全一样的文本



与其他同学对拍了答案的相关度:



相关度达到了97，程序基本正确。

NB分类：

使用KNN分类中的小数据集，取alpha=1，多项式模型得出结果:



验证答案:

P(joy) = 2/3

P(sad) =1/3

P(test1 | joy) = (2/11) \* (1/11) \*(2/3)= 0.011

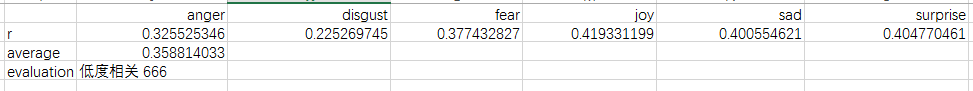
P(test1 | sad) = (1/5) \* (1/5) \*(1/3)= 0.013

则P(test1 | joy) < P(test1 | sad)

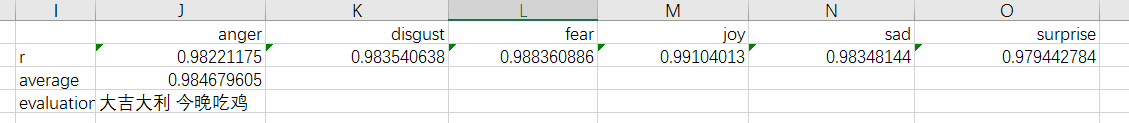
结果为sad，正确

NB回归：

取alpha=0.032跑实验课给的数据集的结果如下：



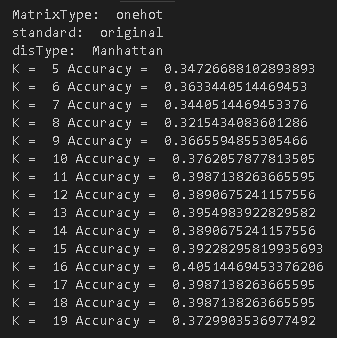
与其他同学对拍了答案的相关度:



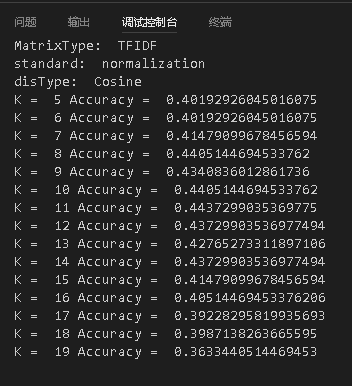
**2. 评测指标展示即分析（如果实验题目有特殊要求，否则使用准确率）**

**KNN分类：**

1. 最普通的onethot + 曼哈顿距离 + 不标准化：



2. 使用所有优化，Z-SCORE标准化 + TFIDF + 余弦距离

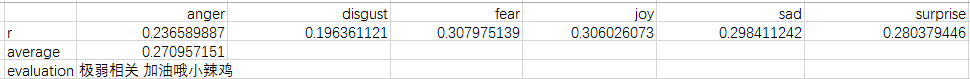


可以发现普通做法，准确率最高只有0.405，而优化了之后，当K=11时，准确率达0.4437。说明优化的确可以帮助提高预测准确率

**KNN回归：**

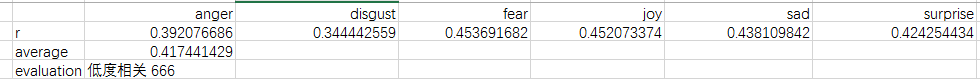
使用分类得到的最优K=11来预测

1. 最普通的onethot + 曼哈顿距离 + 不标准化：



可以发现相关度非常低，只有0.27，其中对disgust的预测最不准确

2. 使用所有优化，Z-SCORE标准化 + TFIDF + 余弦距离 + 特判完全一样的文本



可以看到相关度提高了很多，说明优化很有作用。

**NB分类：**

NB主要调节的是alpha的值，由于我跑了(0,1] 之间，间隔0.001的所有值去当alpha，结果太多，下面节选一些alpha值的结果作为展示

多项式模型:

|  |  |
| --- | --- |
| Alpha | Accuracy |
| 0.001 | 0.42765273311897106 |
| 0.055 | 0.4340836012861736 |
| 0.16 | 0.4437299035369775 |
| 0.2 | 0.4405144694533762 |
| 0.35 | 0.45016077170418006 |
| 0.42 | 0.4662379421221865 |
| 0.55 | 0.47266881028938906 |
| 0.6 | 0.4694533762057878 |
| 0.7 | 0.45980707395498394 |
| 0.8 | 0.4565916398713826 |
| 0.9 | 0.4565916398713826 |
| 1.0 | 0.4565916398713826 |

由表可知，当alpha = 0.55 时，准确率最高，为0.4727

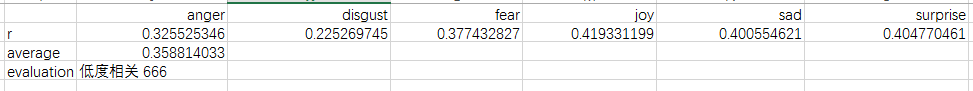
伯努利模型:

|  |  |
| --- | --- |
| Alpha | Accuracy |
| 0.01 | 0.4115755627009646 |
| 0.05 | 0.40192926045016075 |
| 0.1 | 0.38263665594855306 |
| 0.2 | 0.3311897106109325 |
| 0.3 | 0.2861736334405145 |
| 0.4 | 0.2282958199356913 |
| 0.5 | 0.2090032154340836 |
| 0.6 | 0.17684887459807075 |
| 0.7 | 0.14469453376205788 |
| 0.8 | 0.13504823151125403 |
| 0.9 | 0.13504823151125403 |
| 1.0 | 0.12861736334405144 |

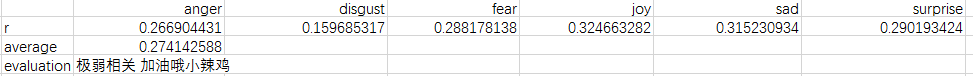
由数据可以看出，伯努利的预测效果远不及多项式模型，多项式模型比较稳定，伯努利模型的正确率与alpha基本成负相关。

**NB回归：**

Alpha = 0.032



Alpha = 1



# 思考题

**KNN：**

1. 回归问题中，为什么要将距离的倒数作为权重？

答：KNN回归问题是将K个邻居的平均属性赋值给当前测试样本，但这样不够准确，应该根据距离的远近，让邻居对这个样本的影响程度有所不同，距离越近的影响越大，因此想到根据距离来加权，取倒数就满足距离越小，权重越大的要求。

2. 同一测试样本的各个情感概率总和应该为1，如何处理？

答：只要在最后对情感概率归一化处理，同比放大即可。

3. 在矩阵稀疏程度不同的时候，曼哈顿距离和欧式距离的表现有什么区别，为什么？

答：我个人觉得，没有什么区别。我认为这两者的表现的区别不在于矩阵的稀疏，而在于向量特征值的差别。假设一个测试样本的特征向量跟很多训练样本的特征向量的欧式距离相同，可能会没办法好好得出正确的类别，不过这个时候虽然欧式距离相同，当曼哈顿距离不一定相同，也许使用曼哈顿距离会更好。而曼哈顿距离如果向量中存在一个特征值相差较大的话，可能会掩盖其他特征值的变化特征。

硬要说矩阵稀疏程度不同两者表现有什么区别的话，假如使用了三元组表示矩阵，随着矩阵越来越密集，欧式距离的计算时间会更大。

**NB：**

1. 分类问题中，伯努利模型和多项式模型有什么优缺点？

答：伯努利模型是文档型的模型，它的特征向量是单词有没有出现过，是一个只含（0,1）两种取值的向量，是以单词出现的文档数占总文档数的概率去做预测，是一种从全局特征考虑的预测方式，实现起来比较简单，缺点就是由于考虑的是全局特征，从而忽略了局部特征的影响，比如同样一个词，在一篇感情为ei的文章中出现了很多次，但是在一篇感情为ej的文章中只出现了1次的话，那么我们直观上应该将这个词归为ei这类，然后在伯努利模型中，这个词是感情ei或ej的概率都是1/2，这样的预测就不够精确。

多项式模型是单词型的模型，它的特征向量是词频，也就是说它将单词出现的频率考虑了进去，这样就可以解决上述伯努利模型中的忽略局部特征影响的问题。缺点是当特征是连续变量时，使用多项式模型会造成后验概率为0的情况，即时使用了平滑处理也不能很好地反映真实情况。

2. 如果测试集中出现了一个之前全词典中没有出现过的词该如何解决？

答：这种情况下我选择忽略这个词,当成测试集中不存在它。原因是由于全词典中没有这个词，那么就不知道跟它有关的各种概率，无法去参与后验概率计算，既然如此，不如选择舍弃。