训练过程：

预处理=>特征提取=>权重计算=>模型训练

预测过程：

预处理=>特征提取=>权重计算=>预测类别

特征词抽取的步骤及实现：

1. 对文本分词；
2. 针对每个词，统计如下列联表（微博条数）：

|  |  |  |
| --- | --- | --- |
|  | 属于正类 | 不属于正类 |
| 包含词W | A | B |
| 不包含词W | C | D |

1. 针对每个词，计算CHI值，按大小排序并保存至文件中。

实现：

1. 统计词频：在python中，使用两个列表，每个列表包含与文本数目相同的字典, 分别保存每篇文本的正向和负向词频；
2. 计算CHI值：；
3. 选取特征词：对于二分类问题，根据上述计算公式可知，如果一个单词w有利于区分正类，那么该单词也有利于区分负类，并且两者的CHI值是相等的。所以，只要根据某一类（正类或负类）来选取n个特征词即可。

模型的训练：

输入：id label word,tf\*idf;word,tf\*idf;…

1. 使用scikit-learn的DictVectorizer生成文本集的特征数组；
2. 调用SVM的训练代码生成模型。

新文本的判别的过程：

输入：text文本

1. 分词并去除不在全单词集中的单词；
2. 调用生成特征向量的函数生成新文本的特征向量；
3. 使用新文本的特征向量进行分类预测。

实验结果分析：

1. 特征提取方法不是很好，最后产生的训练集是非常稀疏的矩阵。
2. 使用BernoulliNB模型进行语料选取时，每次选出的语料都倾向于一个主题，并且极性也偏向于一类。
3. 在进行分类预测时，如果采用BernoulliNB模型，若判别训练语料会获得很高的正确率，若是新语料则正确率会很低；如果使用MultiNormialNB模型，则在判断新语料时还能保持比较高的正确率。

改进：

1. 采用新的特征提取方法；
2. 使用多个模型进行投票判别。