实验4 MapReduce高级编程——上市公司财经新闻情感分析

实验报告

161278031 计算机-金融工程实验班 王婧

一、实验内容：

股市新闻中往往包含了大量信息，除了上市公司的财务数据外，还包括经营公告、行业动向、国家政策等大量文本信息，这些文本信息中常常包含了一定的**情感倾向**，会影响股民对公司**股票未来走势**的预期，进一步造成公司的**股价波动**。如果能够挖掘出这些新闻中蕴含的情感信息，则可以对股票价格进行预测，对于指导投资有很大的作用。

本实验尝试使用**文本挖掘技术和机器学习算法**，挖掘出新闻中蕴含的情感信息，分别将每条新闻的情感判别为“positive”、“neutral”、“negative”这三种情感中的一种，可根据抓取的所有新闻的情感汇总分析来**对股票价格做预测**。

二、实验目标与要求：

使用多种机器学习算法对文本进行情感判别，包括KNN、决策树、朴素贝叶斯、支持向量机等，学习如何进行模型训练，如何进行分类预测，要求使用至少两种分类方法。

三、实验环境：

·MapReduce编程环境：

Jdk 1.8

Hadoop 2.9.1（伪分布运行模式）

Eclipse luna

·Python编程环境：

Spyder

四、实验数据：

1、fulldata.txt：收集沪市和深市若干支股票在某时间段内的若干条财经新闻标题；

2、training\_data.zip：训练数据集，由保存在negative、neutral和positive三个文件夹下的样本数据组成，文件夹名即其中样本的分类；

3、chi\_words.txt：特证词文件，将其中的词语作为特征，并计算特征向量；

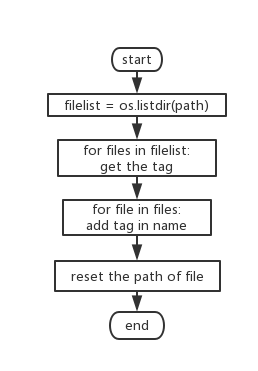
4、data.xlsx：沪市和深市若干支股票在2017年9-12月的收益率等数据，从[www.ccerdata.cn](http://www.ccerdata.cn)下载得到；

五、实验设计：

本次实验任务共分为五个阶段：

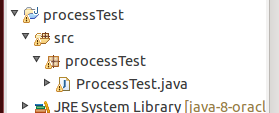
**1、预处理阶段：**

**a.** rename.py：处理training\_data，将标签加入到样本数据名中，并把样本数据放在同一个文件夹下。

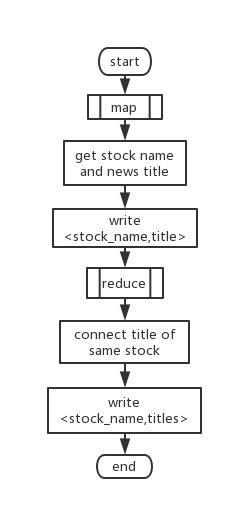


（流程图）

**b.** processTest项目（借鉴wordcount的思想）：处理fulldata.txt，将属于同一股票的新闻标题连在一起。



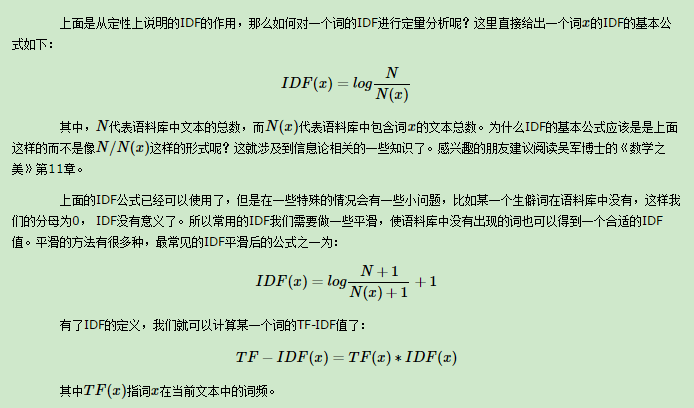
（项目结构）



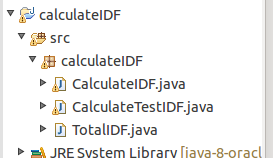
（流程图）

**2、计算tf-idf：**

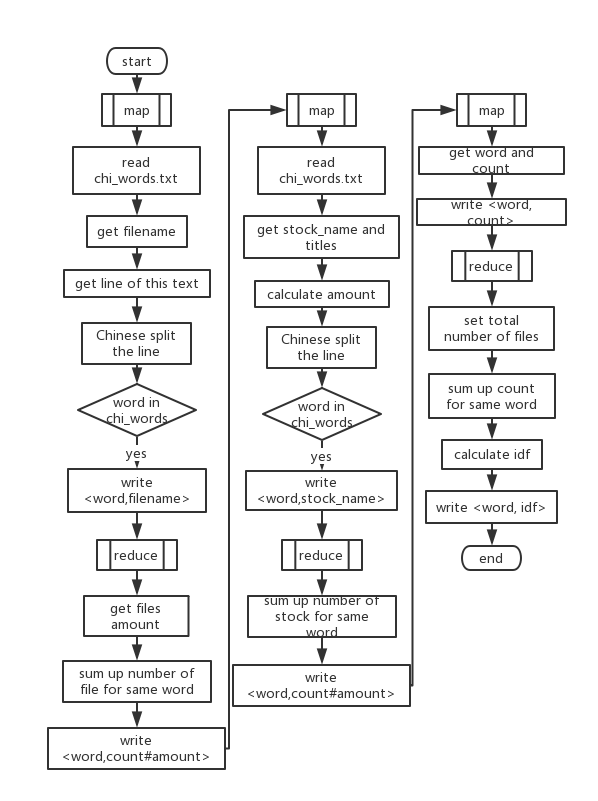
**a.** calculateIDF项目（借鉴invertedindex和wordcount思想）：计算chi\_words.txt中各词的idf值，使用公式如下：



（这里，语料库选择为训练集加测试集）

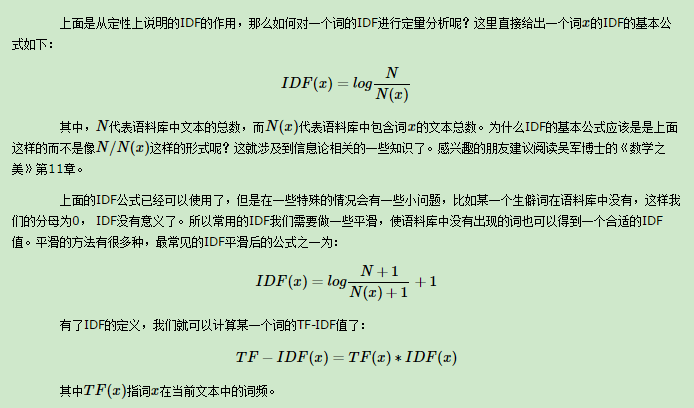


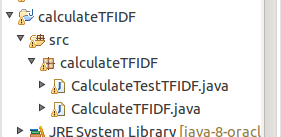
（项目结构）



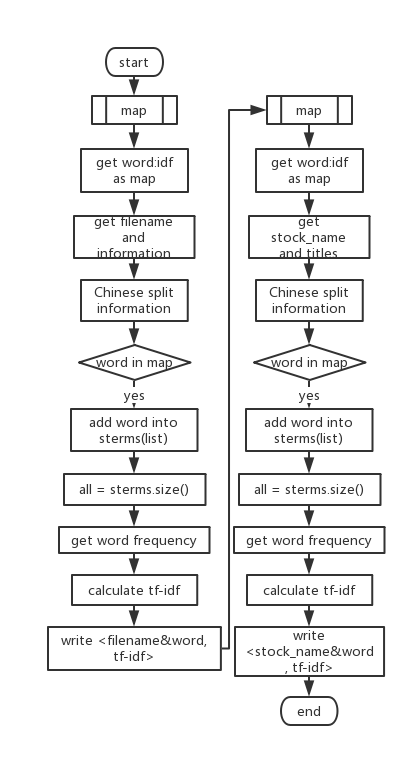
（流程图）

**b.** calculateTFIDF项目：计算训练集各文件和测试集各项各词语的tf-idf，公式如下：





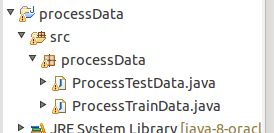
（项目结构）



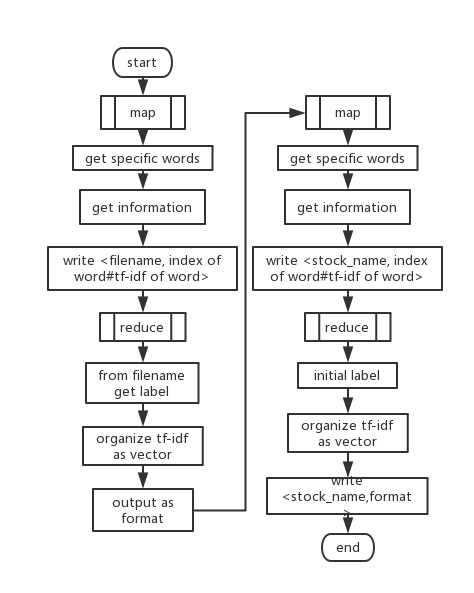
（流程图）

**3、向量化：**

**a.** processData项目：将训练集和测试集的tf-idf向量化。



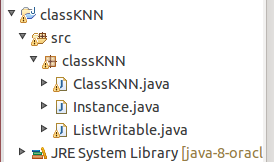
（项目结构）



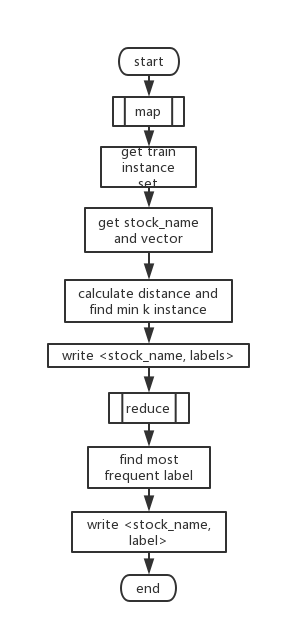
（流程图）

**4、分类：**

**a.** classKNN项目：用KNN对测试集进行分类。

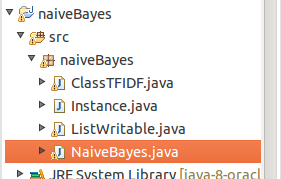


（项目结构）

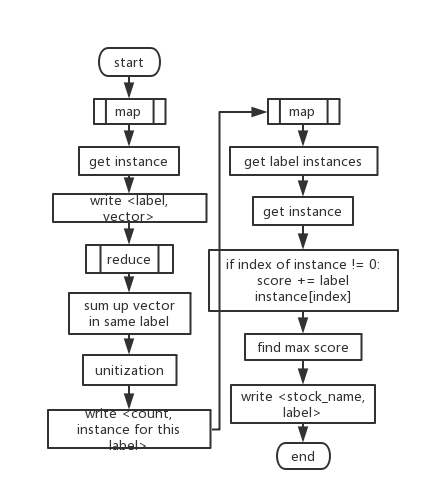


（流程图）

**b.** naiveBayes项目：用朴素贝叶斯对测试集进行分类。



（项目结构）

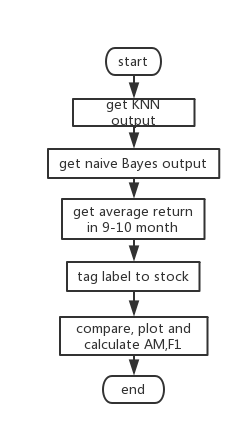


（流程图）

**5、评估：**

**a.** indexArouse.py：生成股票代码文件，供下载数据使用。

**b.** test.py：评估分类效果并可视化。

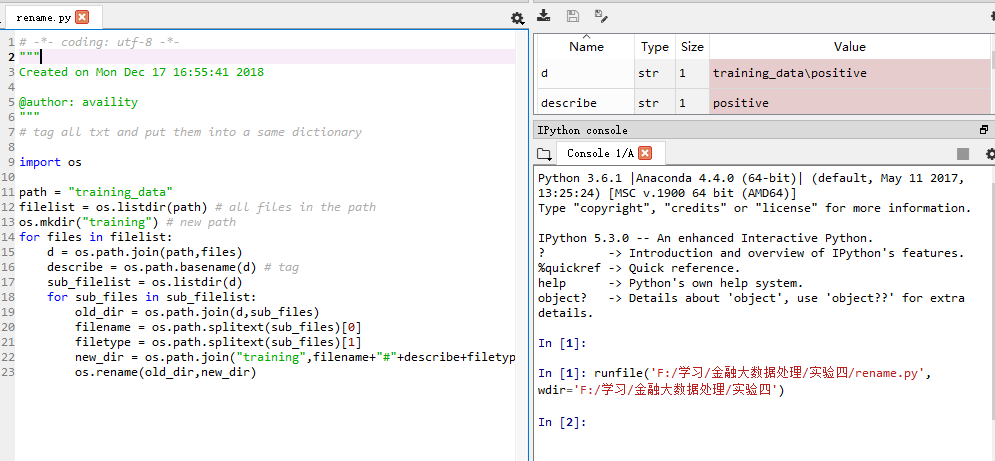


（流程图）

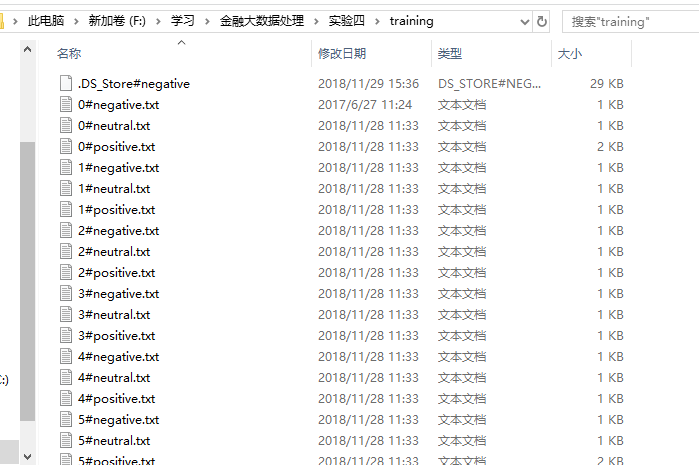
六、程序运行与实验结果：

**1、预处理阶段：**

**a.** rename.py：

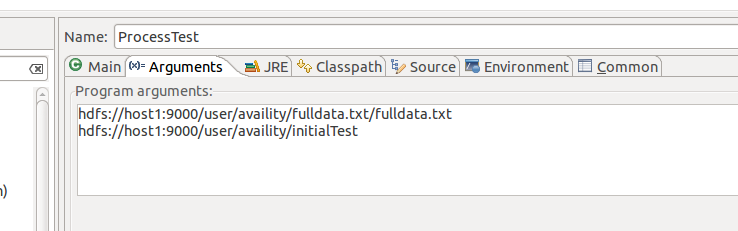


（运行）

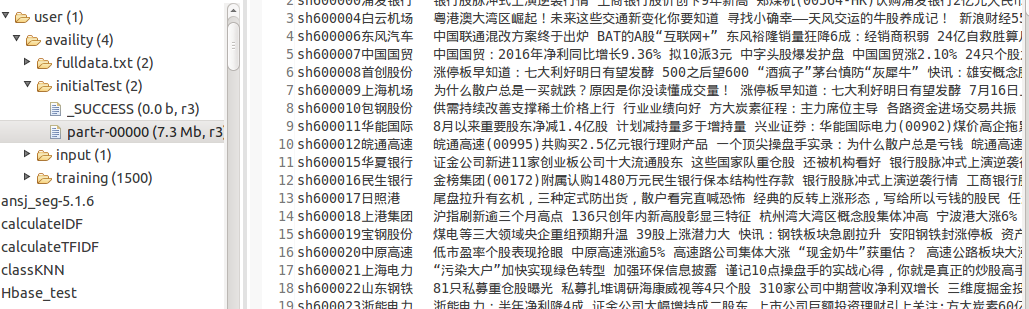


（运行结果 生成的training文件夹）

**b.** processTest项目：



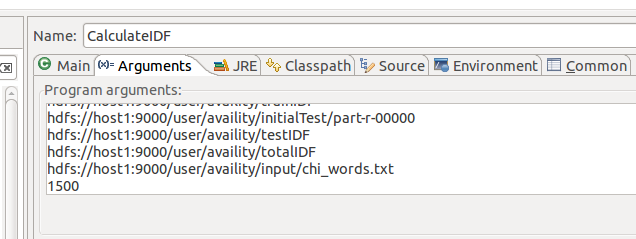
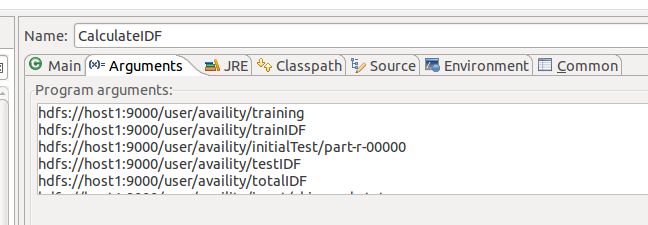
（输入）



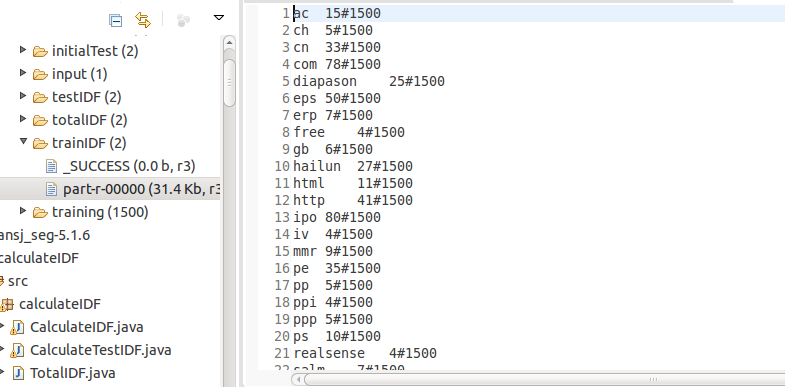
（运行结果）

**2、计算tf-idf：**

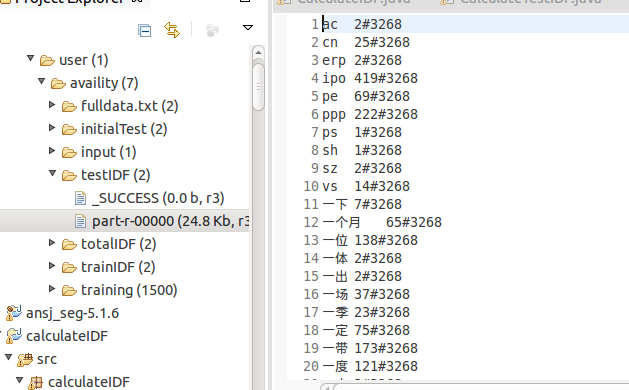
**a.** calculateIDF项目：



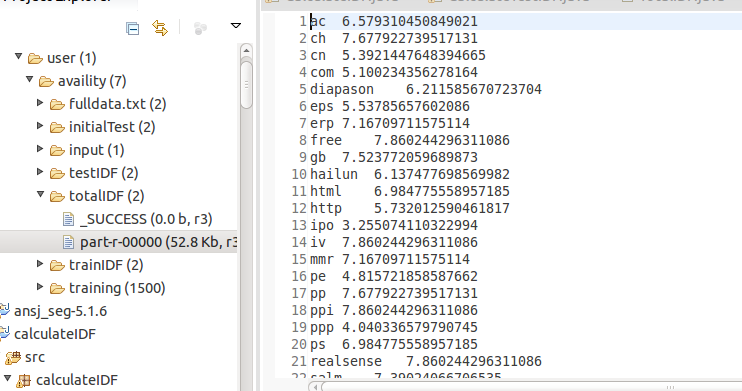
（输入）



（train idf）

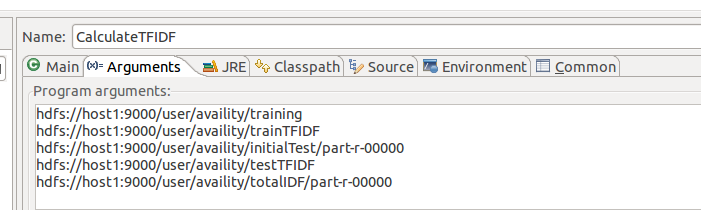


（test idf）

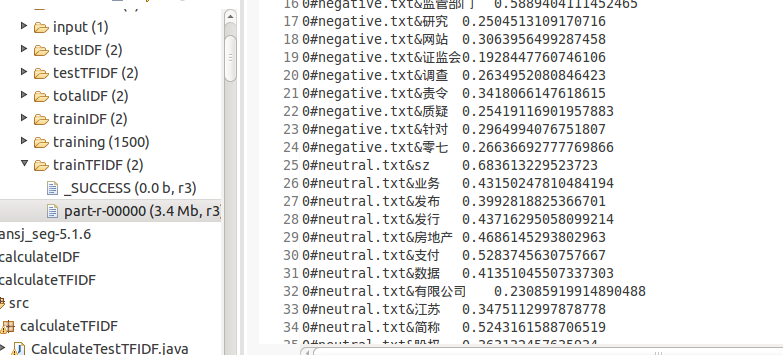


（result：word idf）

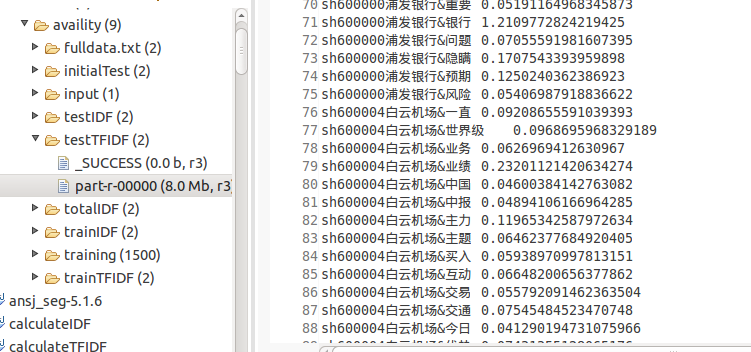
**b.** calculateTFIDF项目：



（输入）



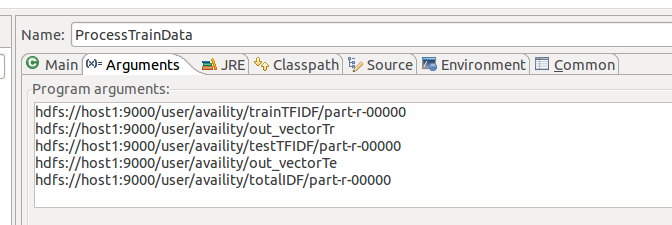
（train tf-idf）



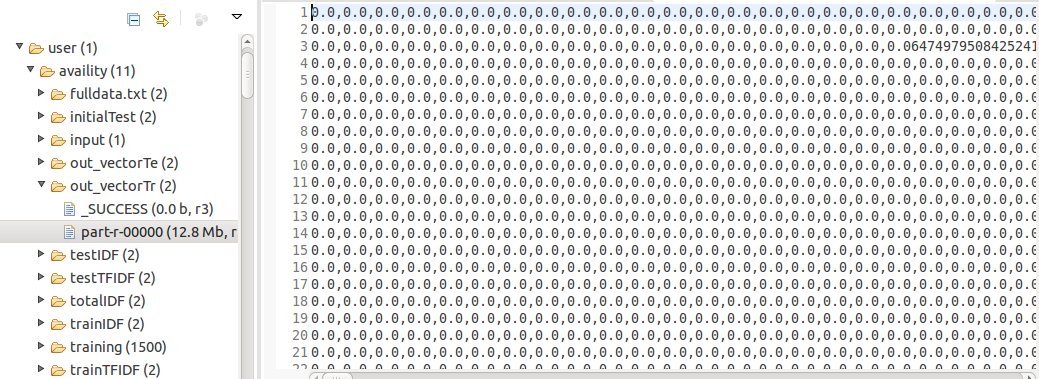
（test tf-idf）

**3、向量化：**

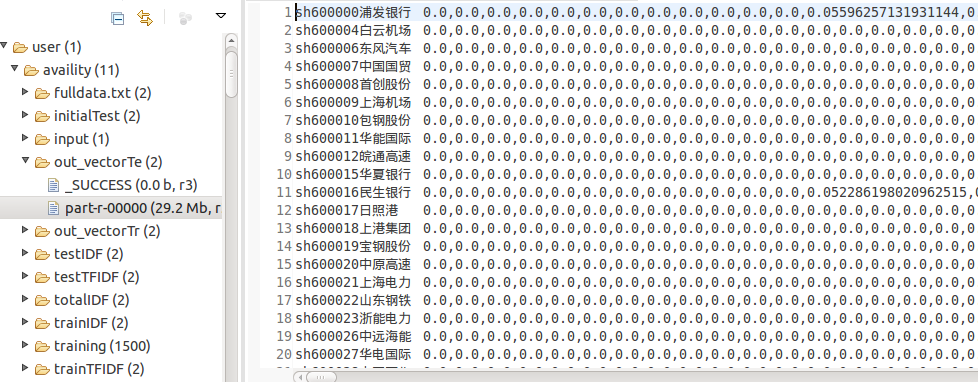
**a.** processData项目：



（输入）



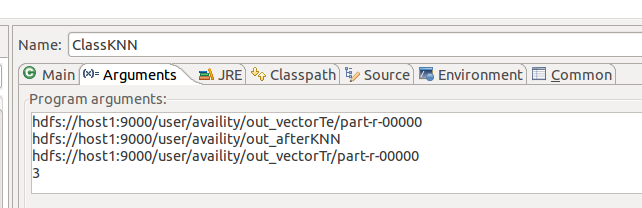
（train vectors）



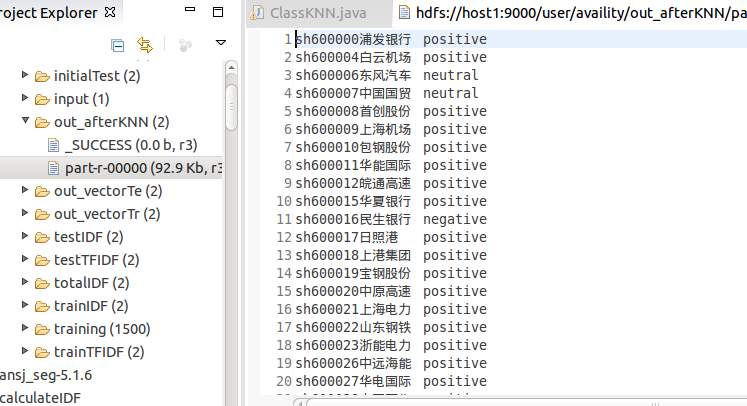
（test vectors）

**4、分类：**

**a.** classKNN项目：

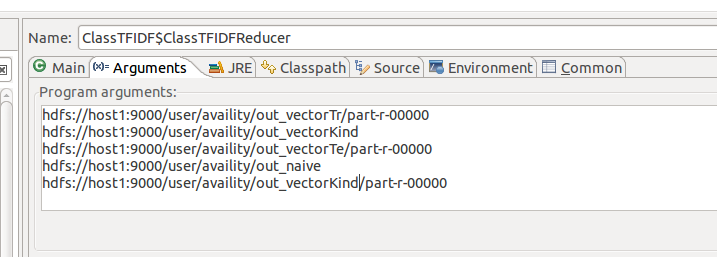


（输入）

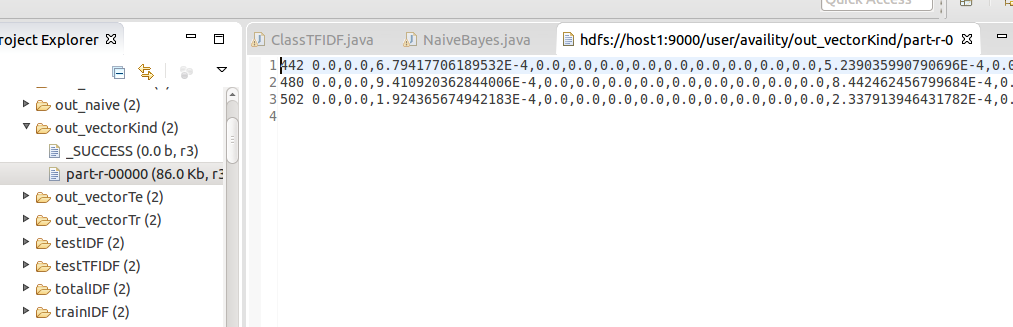


（结果文件）

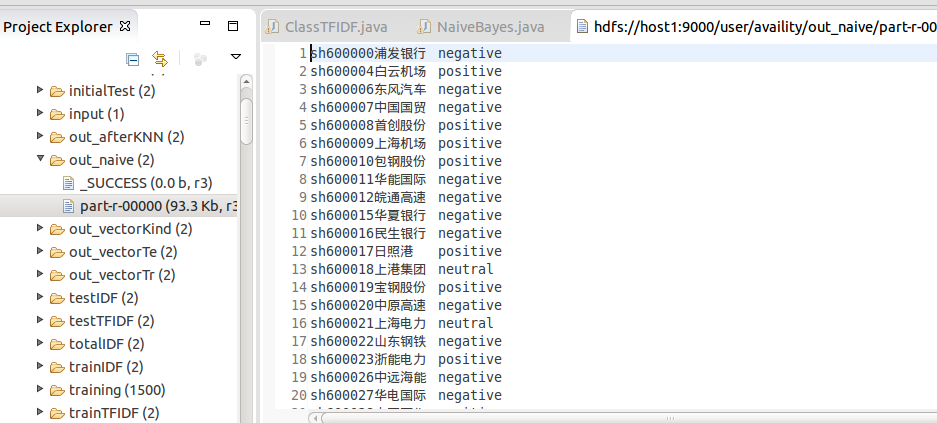
**b.** naiveBayes项目：



（输入）



（对训练集分组向量化）



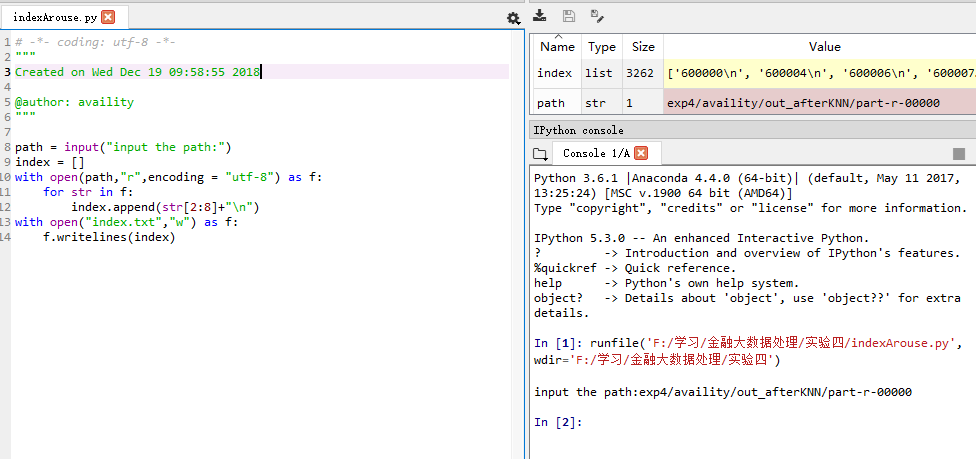
（结果文件）

**5、评估：**

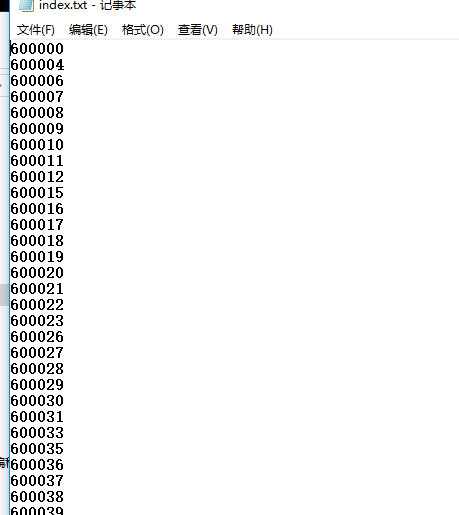
**a.** indexArouse.py：



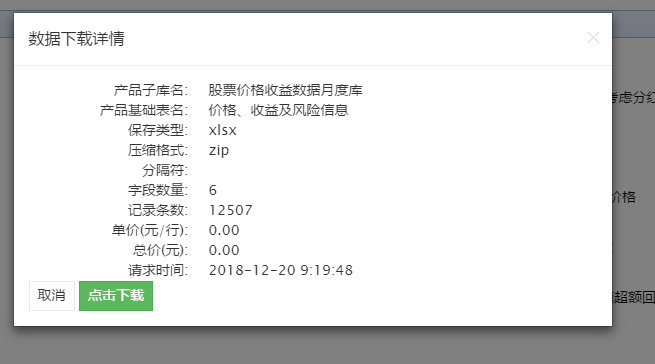
（网站页面）



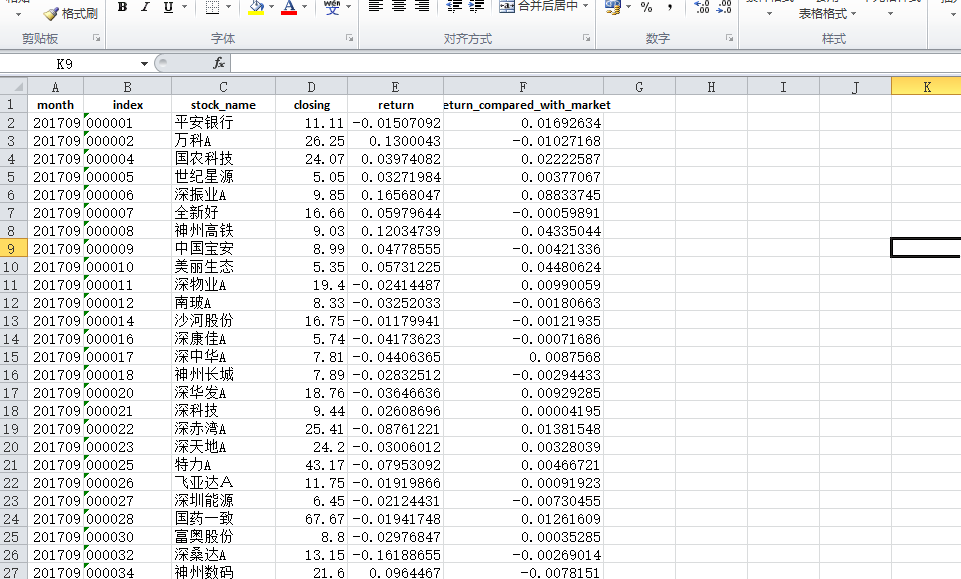
（生成股票代码程序）



（生成股票代码文件）

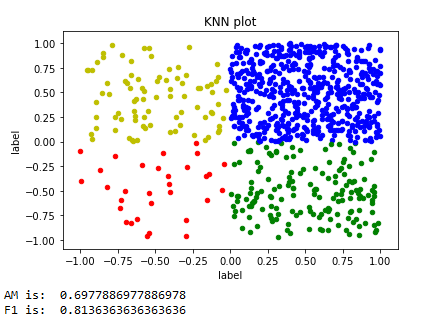


（数据下载详情）

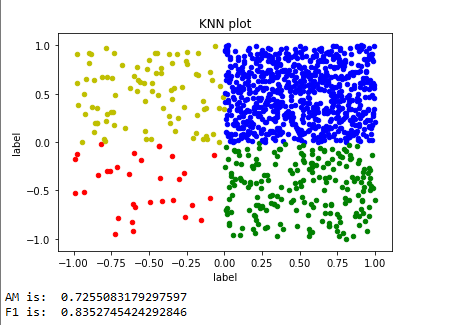


（数据下载结果）

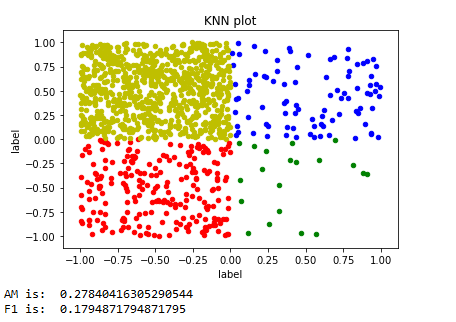
**b.** test.py：



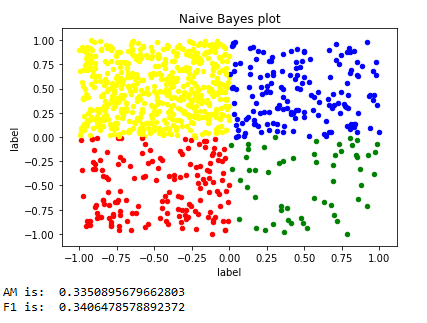
（KNN k=3）



（KNN k=10）



（KNN k=50）



（朴素贝叶斯）

可以得到：KNN方法在k=10左右有较好的表现，而k偏大或偏小表现较为欠佳，朴素贝叶斯表现不太好；同时，分类算法结果总是倾向于一个方向，如KNN（K较小时）倾向于将测试集判断为积极，而朴素贝叶斯倾向于把测试集判断为消极。

整体效果还有待改进。

七、不足和可能的改进之处：

1、选取更为优良的原始数据，可能对分类效果有一定的提升；

2、虽然整个实验按照功能分块，逻辑清晰，但存在一定的代码冗余，可以进行进一步的细致设计；

3、新闻内容直接进行分词，可以先进行一定的预处理，比如去停词等工作；

4、特征词直接选择了chi\_words.txt，可以进一步利用tf-idf筛选出特证词，这对分类的效果也会有提升；

5、模型可以通过交叉验证法进行进一步优化；

6、评估可视化可以进行一定的改进。