**机器学习文本分类数据报告**

**钟东彤 2020200883**

1. **数据爬取**

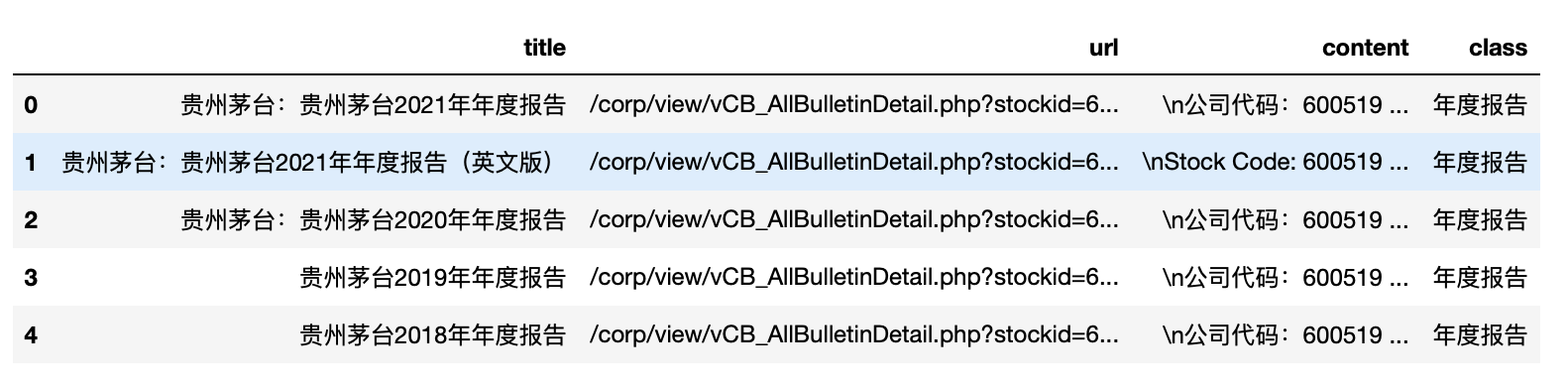
在数据爬取的这部分，我综合和改进了第一次作业的爬虫程序，首先设定好要爬取的公告类型，保存到一个列表中，然后通过读取其中的公告类型id，配合上一个固定的该类别下的公告集合的url地址，就能够得到不同类别下的公告标题和内容，在这个过程中顺便也将各个公告自己的url爬取下来了，为后续的爬取公告文本做好准备。在写好一般性的爬取程序之后，只需要更改不同的股票代码，就能够爬取不同的股票了。

为了各个股票的数据能够叠加，我重复使用了同一个列表来存放各个数据（公告url，class，title），这里只需要设置一下循环条件就可以了，我设置为了100，200，300和400，这是为了各个股票的公告数量尽量均匀，最后每个股票都爬取到了100多条数据，这初步符合了要求。

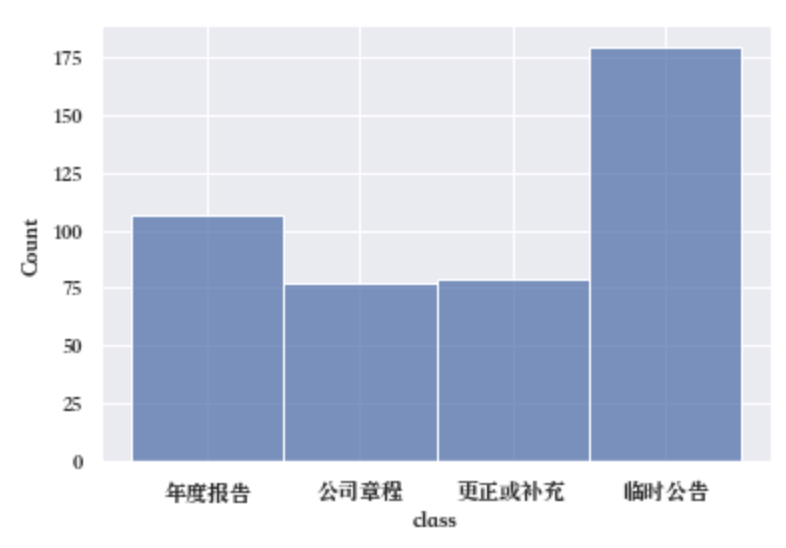
然后就是对汇总的400多条公告url进行循环遍历，得到各自的公告文本，并将其写入列表中。最后汇总几个数据列表为字典，再将其转换为DataFrame类型，至此数据爬取部分的工作就完成了，大概得到DataFrame如下所示：



然后为了后面的展示方便，还需要将各个类别转变为中文名称，目前还是拼音缩写，转换后得到下面的数据：



爬取到的公告类型分布结构如下图所示，可以看出临时公告的数量明显高于其他公告，这与正常上市公司的公告类型分布是基本一致的，因为临时公告的更新频率往往是最高的。



然后写入规定路径的csv文件就可以了。

1. **数据预处理**

接下来是数据预处理的部分，由于爬取到的数据文本比较大，因此需要对其进行分词之后才能抽取特征。同时也为了后续的模型分类学习方便，我另外设置了对应类别的数字类型的class\_id。

然后我再加入一些公司名称之类的特有名词，就可以用上作业3中的分词函数对公告文本进行遍历分词了，分好的词用空格隔开，另外放在一行中，为下一步的文本向量化提供了基础。

接着就是设定好分类的标签和分类的数据，这里用的就是分词后的公告文本，对数据文本进行打乱，然后给分类的学习模型设置好测试集和训练集，这里是2:8的比例。

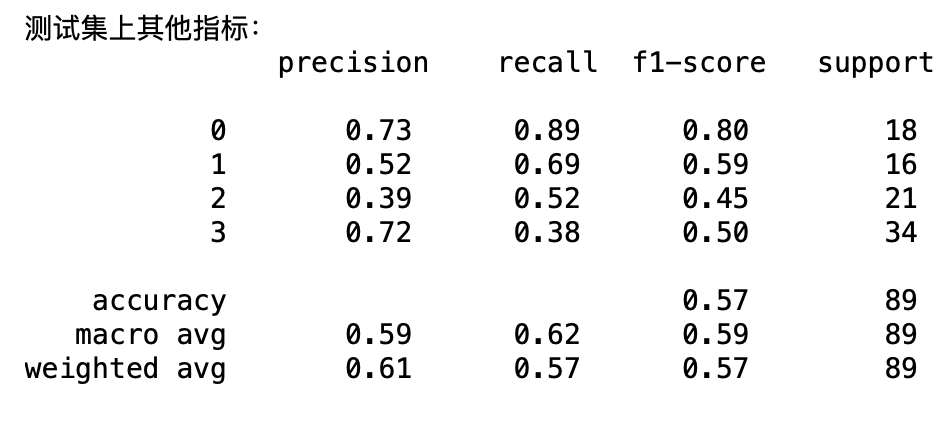
1. **特征抽取**

对数据做好处理之后，就可以对文档进行特征抽取了，这里用的是TfidfVectorizer和CountVectorizer两种向量化的方法，通用词的设定还是使用经过特别改动的hit的那个stopwards，我加了一些上市公司公告里面常见的词汇。同时给词汇设定频率的阈值，上限是文档频率为0.95，下限是至少出现5次，我感觉下限可能还有调高的空间，不过5次也差不多了。

这里分别可以得到用文档中词汇频率和tf-idf值为分量的两组特征向量，维度是1000，因为之前设定了1000个特征。

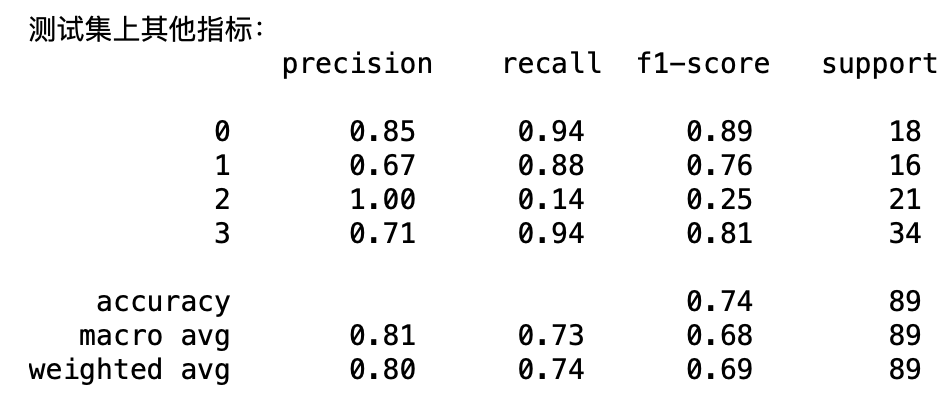
1. **模型效果**

得到特征之后，首先我采用了朴素贝叶斯分类器来进行训练，分别采用两组特征向量，首先是基于tf的特征向量，其训练结果如下图所示：



看的出来效果很一般，基本上属于瞎蒙的程度。

然后是基于tf-idf的，其训练结果如下图所示：

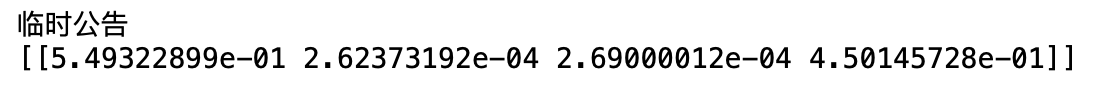
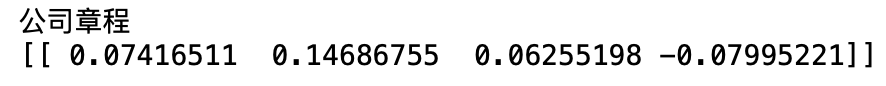


看的出来比前者好了很多，说明tf-idf在对文本特征的概括上表现更强，因为其本身蕴含的信息更综合多样，更方便计算机来识别不同的文档。

接着，我还采用了两个主题分析模型：LDA模型和基于截断的SVD模型。

通过上述两个模型，我能够得到1000个特征下，各个公告类型的首要关键词，以及不同的文档在四个类型维度的上的得分，正是这个得分决定了计算机对该公告文本的分类。

通过拿第一个公告文本进行测试，我发现SVD模型得到了正确的结果而LDA模型得到了错误的结果，如下所示：



其中前者是SVD模型结果，后者是LDA模型结果，实际上该文档属于公司章程，说明基于tf的特征抽取和分类的准确率还是会低于基于tf-idf的模型。

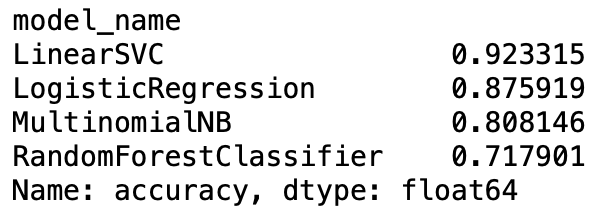
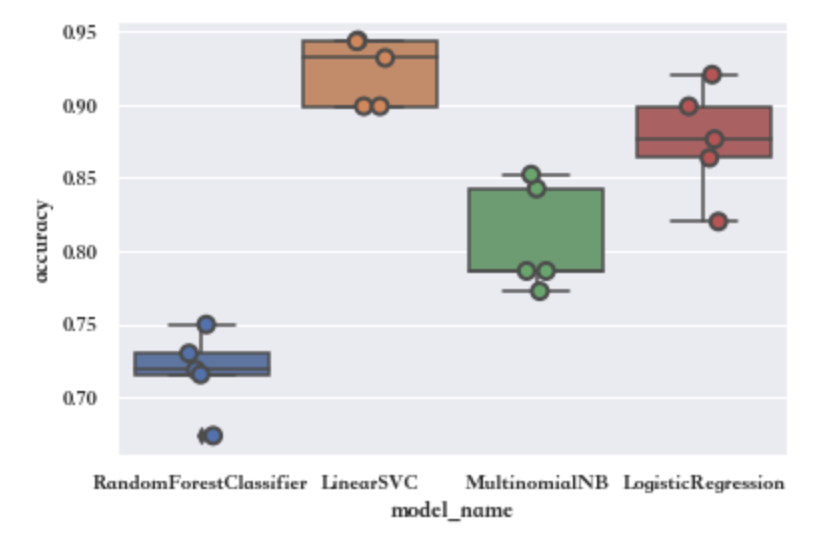
下面是对LDA模型的预测结果展示：



可以看出，效果很一般，准确率只有不到50%，分类效果很差。说明对于高维的文本数据，单一的tf特征指标已经不足以胜任分类的功能，同时，明显偏少的数据量也成为了LDA模型分类效果不佳的重要原因。

1. **不同分类方法比较与评估**

接下来，我还采用了各种不同的分类方法，有logistic回归算法，随机森林算法以及线性支持向量机算法，并在accuracy的维度上进行了分类效果的比较，在使用不同的分类模型进行了多次训练和测试集检验之后，我得到了具体模型的分类准确率分布如下图所示：



可以看到，针对这一数据集，linear SVC的效果最好，准确率达到了92%，其次是logistic回归和朴素贝叶斯，最后是随机森林分类器。可以看到，集成分类器对于处理高纬度的特征向量并不占优势。综合上述，目前来看，对于这一文本数据，最佳的分类方法就是线性分类支持向量机模型。

1. 改进方法

综合全文，不难看出，要改进分类的效果，有以下几个方法：首先，是获取尽可能多的数据，通过查阅网上的资料，我发现好的分类器训练都离不开海量数据的支持，本次所用模型分类效果一般很大程度上源于数据量的不足，450条左右的数据显然还是太少；其次，是是做好数据的清洗，尽可能得使得数据包含的信息足够凝练，排除无关数据特征对分类学习的干扰，以数字特征为佳；最后，是根据数据特征选择合适的分类算法，这部分的知识我由于不太熟悉，因此就只能一个个模型的尝试，并汇总其效果来得到最佳的选项，如果对算法的认识达到一定程度的话，相比效率会大大提高。