中央财经大学

Central University of Finance and Economics

课　　程　大数据分布式计算

实验名称　基于机器学习算法的网易云音乐歌曲情感分析

指导老师　李丰

姓　　名　王思雨

实验日期　2018/01/27

一、实验目的

人们在听歌时喜欢根据自己的当时的感情来选取合适的歌曲进行感情的抒发，所以我们经常可以看到，网易云音乐上有一些专门为人们当时的感情来定制的歌单，这些歌单中存储的是都是一种情感的歌曲。歌曲除了曲调的不同来表达感情外，还有其歌词。伤感与愉悦这两个极大的情感反差的歌曲所用词汇和其中的意向相差就很远，所以打算利用网易云音乐上现有的关于感伤和愉悦的两种情感的歌单所构成的歌曲曲库作为训练样本，利用机器学习模型在分布式云平台的 spark 上构建可以通过歌词区分歌曲所表达情感的二分类器。以此可以将大量没有情感标签的歌曲进行情感分类，以达到对歌曲情感上的判别。

通过歌曲情感分析模型，我们还可以爬取某个歌手的所有歌曲，通过该歌手的具体所有歌词来判断这位歌手的大概喜欢演唱哪种情感类型的歌曲，用以分类该歌手的曲风。

二、作业相应实验环境说明

- **单机环境**

在单机上主要进行了数据的爬取，文本数据的预处理，文本数据的特征提取以及单机机器学习模型构建。所用的实验环境 windows 平台的 Anaconda3 下的 python 3.6。IDE 为 spyder。

- **分布式集群环境**

在分布式集群上主要进行的是文本特征数据的预处理以及各种机器学习模型的构建。所采用的实验环境为 Pyspark2.0.2。

三、歌词文本数据的爬取

3.1 利用 selenium 库进行数据爬取[1]

网易云音乐的网页跟普通的网页相比主要有两点不同：网页是 js 动态加载并且使用了 iframe 框架。如下图所示：

[1] 详细爬虫文件请见代码 crawl.py

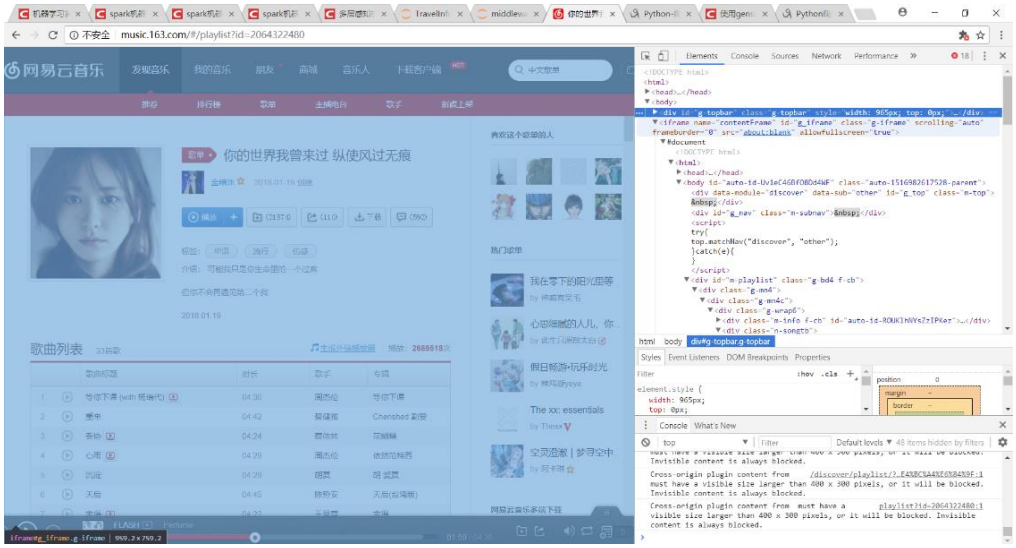

图 3-1 爬虫网页展示

所以，网页请求不能使用 requests 库，需要使用到 selenium 动态解析库以及相应的浏览器，用以模拟鼠标点击和滑动的操作，将完全操作完后的页面在抛回给 beautifulsoup 库进行解析。在此可以使用 chorme 浏览器进行操作，但是用谷歌浏览器会造成资源的大量浪费，因此本文采取的浏览器为 PhatomJS 无头浏览器用以对浏览器的模拟。这样做的好处，一是可以模拟鼠标进行操作并通过设置停顿时间，不用担心被查封 IP。二是节省计算机资源。PhatomJS 无头浏览器爬取数据过程如下图所示：

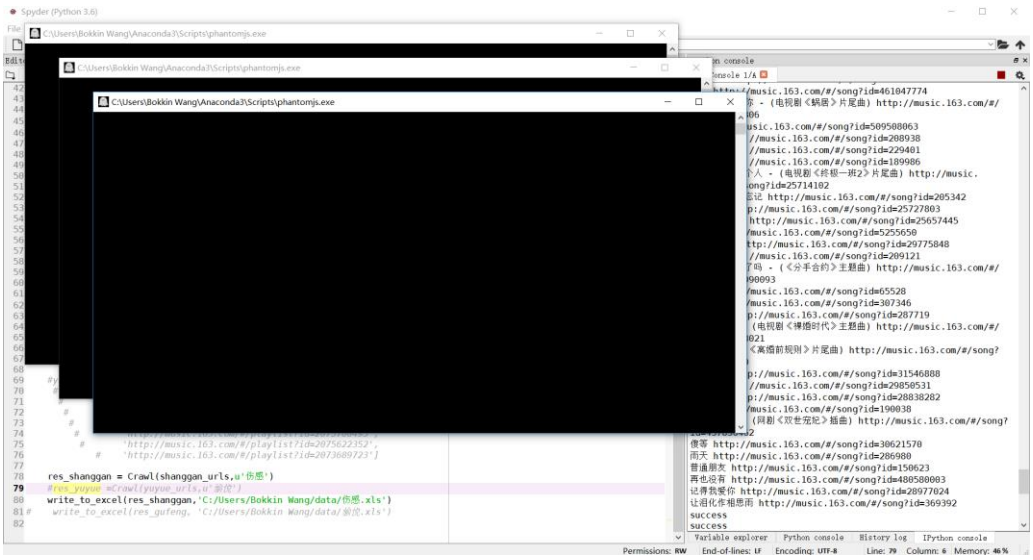

图 3-2 hatomJS 无头浏览器爬取数据过程

大体流程分为两个环节。第一环节，通过愉悦或是伤感歌单的 url 来获取具体歌曲的歌曲名称，歌曲链接以及歌曲的情感类别，这一步的 beautifulsoup 解析源码利用'html.parser'。第二个环节，通过获取的具体的歌曲链接 url 来具体获取歌曲的歌词，歌词的 beautifulsoup 解析源码利用'lxml'。最后将获取的文本数据

以歌曲名、歌词、分类的顺序储存在 xls 表格中，每一行为一首歌曲，如下图所示。获得了两个 xls 数据文件，一个是伤感的歌曲文本数据，另一个是愉悦的歌曲文本。

我们不该这样	浅紫对不起我有些累了我问我们都怎么了再不像棋逢对手般哪怕爱着恨着或酸着突然有了很
星夜的联想	午夜的星空是感觉种种层层的相思在何处留下行踪人生的旅程是路途种种种的日子在内心
今年的湖畔会	在很久很久以前湖畔有位女孩满怀默默纯情向天上星河许下了爱的愿望问着湖中恍惚的身影
想你	为什么你不再象过去那么地那么地那么地柔情长发的女孩如今哪里去为什么你不再象过去那
断点	张敬轩静静地陪你走了好远好远连眼睛红了都没有发现听着你说你现在的改变看着我依然最
他不爱我	金莎我爱他 只爱他好像只能爱到这里了我累了 太累了我终于把执着弄丢了总以为在他的心
兼葭	风 风冷冷地向我们取明的烛火瞥了一眼那乍暗而未复明的一瞬你华丽的爱情 惊惶地向我探
耶利亚女郎	很远的地方有个女郎名字叫做耶利亚有人在传说她的眼睛看了使你更年轻如果你得到她的拥
不愿见你在梦	满壶的茶水却倒不入冷冷的杯中心中的眼泪更无法在眼眶中滴出或许是我不肯将茶倒出只因
赤足走在田埂	黄昏的小村道上洒落一地细碎残阳稻草也披上柔软的金黄绸衫远处有蛙鸣悠扬枝头是蝉儿高
自动弃权	林若宁我看你笑容蛮有福开心得要哭 掩饰我孤独我替我弃权而庆祝他太风趣更显出我木独
迟到	你到我身边带着微笑带来了我的烦恼我的心中早已有个她哦 她比你先到你到我身边带着微笑
祝你幸福	李荣浩编曲：李荣浩花 早春开的花也曾摘下过几芽就放在口袋边上夏 那一年盛夏抖落鞋里

图 3-2 爬取文本展示

3.2 利用 scrapy 框架项目进行数据爬取[2]

scrapy 自身为一个较为完善的爬虫框架，内置解析库为 lxml，所使用的解析方法是根据 xpath 路径解析。分为 Items，Spiders，Pipeline，Middlewares 等组件，因为网易云音乐的部分歌单内部的歌曲有的是重复的，而 scrapy 很智能可以自动逃过重复的 url。利用语句 scrapy startproject ScrapyMusic 创建 scrapy 网易云爬取项目。

在本次的爬虫项目中，Items 中规定了爬取的项目，包括歌曲的情感类别，歌名以及歌词。

Downloader 负责进行 url 网页的下载工作，包括打开歌单网页以及歌曲网页。并将结果传入到 Middlewares 组件中

Middlewares 组件主要实现的是网易云音乐的网页的动态爬取。在其中也使用了动态操作库 selenium 和无头浏览器 PhatomJS 进行网易云 iframe 框架的展开以及歌词的展开。如下图所示。在该组件中也可以在其中加入代理，且可以避免被查封 ip。最后将动态展开的结果传入到 Spiders 组建中。

[2] scrapy 数据爬取项目文件夹请见附件

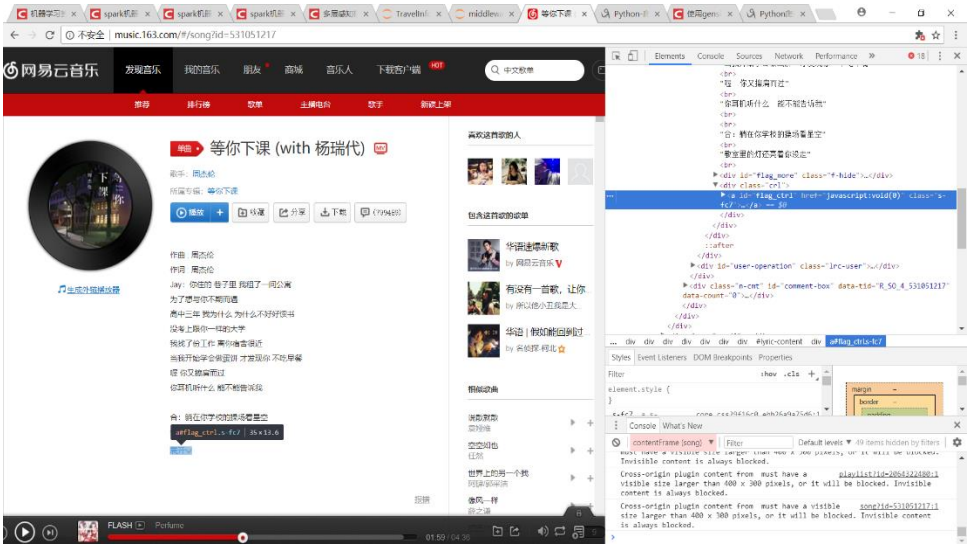

图 3-3 动态爬取文本展示

Spiders 负责解析网页，解析结果分为两种，一种为继续打开 url 的网页链接，比如从歌单中获取的歌曲 url 直接传回 Downloader 下载网页。另一种则是解析网页内容，比如获取歌曲网页中的歌词，歌名等。并将这些歌词，歌名以及歌曲情感打包送入 Pipeline。因为 spider 是并行处理每个 url 所以歌词的顺序可能与 3.1 的处理结果不尽相同。

Pipeline 在本次网易云数据的爬取中，负责了歌曲数据的写入，写入格式为 json 形式。在数据分析时在单机 python 中进行处理与整合。数据形式如下图所示。由此获得伤感歌曲.json 和愉悦歌曲.json 两个数据文件。

四、文本数据说明与预处理[3]

4.1 文本数据的清理

首先将文本中出现的空白符、'\u3000'、'\n'等文本格式字符，将文本构成单行的形式，储存在 txt 文件中，并将每一类情感的歌曲储存在一个文件夹内，便于分析。

将文本进行分词后，在分词后的直接结果中，有大量的无效项，例如空格，逗号等等。因此，一般在分词以后，还要进行预处理，将每个歌词文本中的停用词依照《哈工大停用词表》进行停用词的删除处理。

然后将文本中的纯数字，纯英文的单词进行删除。由此将每个歌曲的歌词文本都处理成了单个中文词汇的形式。最后得到的单词，其顺序是打乱的，即单词间的相关信息已经丢失

还有一些音乐是纯音乐并没有歌词，对于这样的歌曲，通过字符串大于 15 从而删除了没有歌词的歌曲。

[3] 该部分代码请见 main.py 文件

4.2 文本分词以及词袋生成

分词有很多种方法，也有很多现成的库，本次实验使用的是 jieba 库就，由 fxsjy 大神开源的一款中文分词工具，一款属于工业界的分词工具——模型易用简单、代码清晰可读，Jieba 采用的是 Unigram + HMM 算法模型，基于 TF-IDF 的关键词提取。且能为提取出的词汇进行标注词性。

jieba 分词库中支持分词的三种模式，分别为精确模式，全模式和搜索引擎模式，对不同情感歌词的分类采用的是精确模式，可以列出最优分割选项，保证较少的出现歧义现象。

生成词袋的过程主要是利用 gensim 库进行的，在将文档分割成词语之后，将这些歌词词汇，使用 dictionary.add_documents 生成歌词情感词汇词典，并可以使用 save 函数将词典持久化，以便下次调用。由于一些歌曲的歌词中有着特殊符号，和一些人名无法批量进行删除，但这并不影响分析，因为在生成词袋时，进行词频删除分析。会讲一些低词频的特殊字符进行删除，只提取一些普遍的代表的单词。

接下来要用情感 dictionary 把每个歌曲歌词从词语列表转化成用词频表示的向量。也就就是向量中的一维对应于词典中的一项。如果以词频表示，则向量中该维的值即为词典中该单词在文档中出现的频率。其实这个转化很简单，使用 dictionray.doc2bow 方法即可。到此为止，歌曲中的歌词文本已彻底转化为了词频表示的数字向量。

4.3 文本数据特征提取

将每个歌词的词频向量进行特征提取工作，需要调用 gensim.models 库先将每个歌曲的歌词词频向量利用 tf-idf 模型转化为 tfidf 向量，再利用 gensim 库中 lsi 模型提取文本的 50 个特征（也等于转成 lsi 模型以后每个歌词文档对应的向量长度。转化以后的向量在各项的值，即为该文档在该潜在指标的权重）转化为文本特征向量，由此完成歌词文本特征的提取工作。由此，lsi 的结果也可以看做该歌词文本的文档向量，用于后续的情感分析算法。

4.4 spark 分析数据生成[4]

在单机 pyhthon3.6 中生成上传到云平台上的数据。在 main.py 函数中，利用 pickle 库将生成的训练集与测试集，写出序化模型。python 的 pickle 模块实现了基本的数据序列和反序列化。通过 pickle 模块的序列化操作我们能够将程序中运行的对象信息保存到文件中去，永久存储；通过 pickle 模块的反序列化操作，保存的训练集和测试集的歌词文本特征数据是可以还原为保存前的数据形式的。

在 data_for_spark.py 中读取 pkl 格式的序化模型，并将歌词文本标签与每个

[4] spark 数据生成请见附件 data_for_spark.py 文件

文本的 50 个文本特征进行合并，行间距利用'\t'进行分割，生成 train.tsv 以及 test.tsv 两个用于 spark 分析数据。

样本总数目为 1500 首歌曲。其中训练样本 train.py 为 1200 首两种情感的歌曲，测试集的样本 test.py 为 300 首两种情感的歌曲。

五、基于 spark 平台的歌曲情感分析模型实现[5]

主要利用的是 MLilb 库进行处理，利用 MLilb 库中的各种分类模型来讨论每个模型对于歌词文本数据的情感分类状况的处理效果。从而选择最优的模型来进行歌曲情感的分析模型。

5.1 Pyspark 分布式计算数据预处理

在设置标签时本文统一将愉悦的歌曲标签设置为 0，而伤感歌曲的标签设置为 1。在 Pyspark 的机器学习库 MLilb 中的核心是将数据处理成 LabeledPoint 格式。

首先在 Pyspark 分布式中利用 SparkContext 函数，将 train.tsv 和 test.tsv 进行处理，处理成 RDD 的弹性数据集。然后利用 LabeledPoint 和 Vectors.dense 函数将数据处理成，MLilb 库能够识别的格式。

对于朴素贝叶斯模型的处理，有所不同的是。朴素贝叶斯模型要求特征值非负,否则碰到负的特征值程序会抛出错误。因此,需要为朴素贝叶斯模型构建一份输入特征向量的数据,将负特征值设为 0。其他的处理和其他处理相同。

现在已经从歌曲歌词的数据集中提取了基本的特征并且创建了 RDD,接下来开始训练各种模型吧。为了比较不同模型的性能,我们将训练逻辑回归、SVM、朴素贝叶斯、决策树以及随机森林模型。每个模型的训练方法几乎一样,因为都是构建在 spark 的 MLilb 库中的，不同的是每个模型都有着自己特定可配置的模型参数。MLlib 大多数情况下会设置明确的默认值,但实际上,最好的参数配置需要通过评估技术来选择。

5.2 逻辑回归情感分类模型

逻辑回归是一个概率模型,也就是说该模型的预测结果的值域为[0,1]。对于二分类来说,逻辑回归的输出等价于模型预测某个数据点属于正类的概率估计。逻辑回归是线性分类模型中使用最广泛的一个。

逻辑回归模型的参数，因为提取了 50 个特征，只显示其中前 5 个特征的回归系数。但因为文本特征都进行了抽象化的 lsi 文本特征提取，所以这些回归系数并不具有实际意义。回归权重如下表 5-1 所示：

[5] pyspark 分布式处理代码全部存放在 exam.py

表 5-1 逻辑回归模型权重

文本特征	权重 weight
第 1 特征	0.09745
第 2 特征	-0.19362
第 3 特征	-0.00953
第 4 特征	0.00573
第 5 特征	-0.08252

通常在二分类中使用的评估方法包括:预测正确率和错误率、准确率和召回率、准确率–召回率曲线下方的面积即 PR、ROC 曲线下的面积 AUC。对于歌词文本情况的效果如下表所示：

表 5-2 逻辑回归模型评价

Accuracy	AUC	Recall	PR
0.8031	0.8163	0.7726	0.8041

对于情感分类的二分类效果上来看，逻辑分类模型的准确率达到了 80%以上，效果较好，能够将大部分的歌曲情感识别出来，AUC 面积达到 0.81 比较接近与 1。召回率较低，说明真阳性率较低。但无关紧要，因为本次的模型的目的只是分辨歌曲的情感，所以召回率的影响并不是很大。准确率-召回率(PR)曲线,表示给定模型随着决策阈值的改变,准确率和召回率的对应关系。PR 曲线下的面积为平均准确率。PR 曲线下的面积为 1 等价于一个完美模型,此时，准确率和召回率达到 100%，PR 值达到 80%，总体的效果还是不错的。

5.3 支持向量机情感分类模型

SVM 在分类方面是一个强大且流行的技术。和逻辑回归不同,SVM 并不是概率模型,但是可以基于模型对正负的估计预测类别。SVM 的连接函数是一个对等连接函数，因此,当 w^Tx 的估计值大于等于阈值 0 时,SVM 对数据点标记为 1,否则标记为 0(其中阈值是 SVM 可以自适应的模型参数)。SVM 是一个最大间隔分类器,它试图训练一个使得类别尽可能分开的权重向量。SVM 不仅表现得性能突出,而且对大数据集的扩展是线性变化的。尤其对于本次试验的歌曲小样本数量,多变量特征的情况更为适合。

表 5-3 支持向量机模型权重

文本特征	权重 weight
第 1 特征	0.19887
第 2 特征	-0.39029
第 3 特征	-0.01916
第 4 特征	0.01160
第 5 特征	-0.16584

支持向量机的模型评价如下表所示:

表 5-4 支持向量机模型评价

Accuracy	AUC	Recall	PR
0.8531	0.8363	0.8526	0.8441

由于在 main.py 中也进行了单机的支持向量机的模型构建，与 spark 集群上的结果相比来说，单机上的 SVM 的运行速度上稍显较慢，但是准确率却达到了 90%左右，相比集群上的实验结果，利用单机 python 所能够构建的模型更为准确。但是支持向量机在 spark 的结果也已经超越的逻辑回归的表现。

5.4 朴素贝叶斯情感分类模型

朴素贝叶斯是一个概率模型,通过计算给定数据点在某个类别的概率来进行预测。朴素贝叶斯模型假设每个特征分配到某个类别的概率是独立分布的（假定各个特征之间条件独立），这也是朴素贝叶斯叫法的原因。

基于这个假设,属于某个类别的概率表示为若干概率乘积的函数,其中这些概率包括某个特征在给定某个类别的条件下出现的概率(条件概率),以及该类别的概率(先验概率)。这样使得模型训练非常直接且易于处理。类别的先验概率和特征的条件概率可以通过数据的频率估计得到。分类过程就是在给定特征和类别概率的情况下选择最可能的类别。

另外还有一个关于特征分布的假设,即参数的估计来自数据。MLlib 实现了多项朴素贝叶斯(multinomial naive Bayes),其中假设特征分布是多项分布,用以表示特征的非负频率统计,并且普遍用于文本分类，是比较适合此次的歌词情感分析的需求的。

表 5-5 朴素贝叶斯模型评价

Accuracy	AUC	Recall	PR
0.8514	0.8179	0.8422	0.8330

由于朴素贝叶斯模型在数据预处理时将一些负值的特征进行了归零处理的粗糙预处理，所以不是很妥当，剔除了一些可能很有用的信息，但分类效果还是不错的，情感分析准确率达到 85%左右。

5.5 决策树情感分类模型

决策树是一个强大的非概率模型,它可以表达复杂的非线性模式和特征相互关系。决策树在很多任务上表现出的性能很好,相对容易理解和解释,可以处理类属或者数值特征,同时不要求输入数据归一化或者标准化。

决策树算法是一种自上而下始于根节点(或特征)的方法,在每一个步骤中通过评估特征分裂的信息增益,最后选出分割数据集最优的特征。信息增益通过计算节点不纯度(即节点标签不相似或不同质的程度)减去分割后的两个子节点不纯度的加权和。对于分类任务,这里有两个评估方法用于选择最好的分割：基尼系数和熵。

决策树情感分类模型的处理结果如下表所示：

表 5-6 决策树模型评价

Accuracy	AUC	Recall	PR
0.8402	0.8336	0.8022	0.8290

可以看到,决策树的模型结果是介于朴素贝叶斯模型与逻辑回归模型之间的,效果不错，已经达到了 84%正确率。

5.6 随机森林情感分类模型

决策树非常适合应用集成方法(ensemble method),比如多个决策树的集成,称为决策树森林，相对于决策树，随机森林主要降低了模型预测的方差项。

首先，简单介绍一下随机森林：由多个决策树构成的森林，算法分类结果由这些决策树投票得到。其本质是基于决策树的 bagging 算法。决策树在生成的过程当中分别在行方向和列方向上添加随机过程,行方向上构建决策树时采用放回抽样(bootstrapping)得到训练数据，列方向上采用无放回随机抽样得到特征子集,并据此得到其最优切分点，除此之外，所有的决策树的训练都是同样的。

对于分割节点，最好的分割点是通过量化分割后类的纯度来确定的，目前有三种纯度计算方式，分别是 Gini 不纯度、熵(Entropy)及错误率。

分类采用的就是投票机制，每个决策树都会有一个分类的结果，根据得票最多，就是哪类。随机森林的模型评价结果，如下表所示：

表 5-7 随机森林模型评价

Accuracy	AUC	Recall	PR
0.8718	0.8592	0.8407	0.8444

相对于决策树模型虽然准确率提升的并不多，但也有所提高，是这几种机器学习模型中效果最好的，可以准确率达到 87%左右，在排除音乐曲调状况下还能达到这样的效果已经实属不错了。而且随机森林在集群 spark 上的运行速度也是很快的，这相对于以往的单机上的随机森林速度已经提升很多。

六、情感分析结论与展示

本文是一篇完整的文本数据处理分析报告。分别通过 scrpy 框架和 request 库进行了两种网络爬虫的动态抓取数据，爬取到了网易云音乐的网页信息，并将歌词数据返回到本地，进行了文本数据的处理，包括歌词的文本清理，歌词分词去重，歌词文本词袋的生成以及歌曲歌词文本的特征向量化处理，最后又将歌词的特征数据放入到 spark 集群上进行多模型分类器的比较。是一套较为完善的文本处理和分布式集群应用的实例。

文本爬取和前期处理主要是以单机为主，分类模型的应用主要是在 spark 分布式集群上进行。通过对比我们主要可以得出两个方面的结论。

首先对于网络爬虫的应用，scrapy 框架更加适于对大量文本数据的爬取，因为较原始的 beautifulsoup 库来讲，scrapy 项目提供了爬取数据的多个组件，使用的检索方式为 Xpath，且利用并行的方式，使用更加灵活，效率更高。在 middleware 组件中还可以进行各种动态方式的设定，比如加载 selenium 库等，使得 ip 被查封的可能性被降低了。

其次对于模型的选择上，由是位于分布式上的 spark 进行的操作，所以所耗时间都是比较少的，时间成本降低。对于歌词情感分析这一个样本的对比中，随机森林分类模型的效果是最好的，正确率已经达到了 87%，是一个比较可以信赖的“伤心/愉悦”歌曲的分类器。还有就是朴素贝叶斯以及支持向量机模型的分类效果也是比较不错的，正确率都在 85%左右。

七、拓展与应用

本歌曲情感分析 NLP 模型主要的作用是通过歌曲的歌词解析一首歌曲所含有的情感。也就是间接地为歌曲进行分类，贴上情感标签。由此可以将不同情感的歌曲推荐给不同感情需求的人，是一个比较实用化的歌曲情感分类模型。

但由于时间紧迫，所以有很多功能并不完善。比如本情感解析器只能进行愉

悦和伤感两种极端的感情的歌曲分类。虽然正确率比较高，但是在实际应用上比较欠缺。因为实际的根据感情的歌曲推荐是包含多种情感，比如思念，兴奋等比较模糊的情感，所以在后续的完善中会加入更多的情感分类，构造更多特征变量。

另一个是将搜集的其他语言的歌曲都删除了，只进行了中文歌曲的情感分析，但实际上 不同种语言的歌曲同样给人们传递出相同的感情，且现在很多年轻人更喜欢尝试不同语言的歌曲，甚至中文歌曲中还会混合一些外文。所以在后续改善中会加入更多语言的分类，扩展所构造出的词典，加入更多的外文词汇。

在生成歌词情感字典时，只是进行了单机的词频统计以及字典的生成。如果所用的文本较少，单机处理也能操作的过来。但是，如果文本信息较多，则可以利用 MapReduce，采用边分词边统计的方法。将文本分批读取分词，然后用之前生成的词典加入新内容的统计结果，由于文本之间的顺序变不重要。可以将此任务交给集群进行词频统计与生成字典。

八、参考网址：

分布式：
http://blog.csdn.net/u013719780/article/details/51784452
http://blog.csdn.net/u013719780/article/details/51768720
爬虫参考：
https://www.cnblogs.com/Albert-Lee/p/6276847.html?utm_source=itdadao&utm_medium=referral
https://www.cnblogs.com/shaosks/p/7278634.html
https://www.cnblogs.com/kongzhagen/p/6549053.html
文本分析参考：
http://blog.csdn.net/u014595019/article/details/52515616
http://blog.csdn.net/u010105243/article/details/53352155
http://blog.csdn.net/questionfish/article/details/46739207