**华中科技大学计算机科学与技术学院**

**《机器学习》结课报告**



专 业： 计算机科学与技术

班 级： ACM1901班

学 号： U201915035

姓 名： 邹雅

成 绩：

指导教师： 黄宏

**完成日期： 2021年 11月 28日**

目录

[新闻数据情感分析 2](#_Toc89121272)

[一、实验题目：新闻数据情感分析 2](#_Toc89121273)

[二、实验要求 2](#_Toc89121274)

[三、算法设计 2](#_Toc89121275)

[四、实验环境与平台 2](#_Toc89121276)

[五、程序实现 3](#_Toc89121277)

[六、实验结果 3](#_Toc89121278)

[七、结果分析 3](#_Toc89121279)

# 新闻数据情感分析

## 实验题目：新闻数据情感分析

使用机器学习相关知识完成新闻数据的情感分析。数据集分为训练集和测试集。其中训练集的标签只有0和1，即这是一个二分类问题。0指新闻不客观，1指新闻客观。

## 二、实验要求

**2.1题目背景**

随着信息技术的进步，新闻和事件的发布媒介越来越快。信息技术也充斥着大量的数据，每天每分钟都有数百万用户以评论、博客、通过博客分享新闻的形式发布，社交媒体微博网站等。面对如此庞大且富有情绪表达的文本信息，完全可以考虑通过探索他们潜在的价值为人们服务。因此近年来情绪分析受到计算机语言学领域研究者们的密切关注，成为一项进本的热点研究任务。情感分析或意见挖掘是一种发现书面文本中表达的意见(正面或负面)的极性或强度的方法。从新闻标题文本中提取特征，分析新闻标题的情感极性。研究得出结论，标题的极性对新闻文章的受欢迎程度有很大的影响。研究发现，负面和正面的新闻标题比中性基调的新闻标题更能吸引读者的兴趣。

本题目标为在庞大的数据集中区分文本的客观性，将数据进行二分类。面对浩如烟海的新闻信息，精确识别蕴藏在其中的情感倾向，对舆情有效监控、预警及疏导，对舆情生态系统的良性发展有着重要的意义。

**2.2数据集**

手动标注的2200条多语言新闻训练数据和907条多语言测试数据，以json格式提供，其中训练集格式如下：

{“content”:“新闻文本”,“label”:0}

测试集格式如下：

{“content”:“新闻文本”}

**2.3任务描述**

使用机器学习相关知识完成新闻数据的情感分析。其中将新闻数据进行二分类，0指新闻不客观，1指新闻客观。编码采用UTF-8编码。

为完成任务首先要对数据进行清洗，其次要将文本转化为词向量，最后思考如何预测文本的情感。

**2.4评测标准**

将预测的测试集标签输入educoder平台，平台在后台测试对比，并返回两个值micro-f1和macro-f1。为了更好的学习和构造代码，了解这两个测评值。

f1-score是统计学中用来衡量二分类模型精确度的一种指标，用于测量不均衡数据的精度。它同时兼顾了分类模型的精确率和召回率。f1-score可以看作是模型精确率和召回率的一种加权平均，它的最大值是1，最小值是0。微观f1(micro-f1)和宏观f1(macro-f1)都是f1合并后的结果，这两个f1是用在分类任务中的评价指标，是两种不一样的求f1均值的方式。micro-F1和macro-F1的计算方法有差异，得出来的结果也略有差异。

micro-f1的计算方法：先计算所有类别的总的precision和recall，然后计算出来的f1值即为micro-f1。micro-f1的使用场景是在计算公式中考虑到了每个类别的数量，所以适用于数据分布不平衡的情况；但同时因为考虑到数据的数量，所以在数据极度不平衡的情况下，数量较多数量的类会较大的影响到F1的值。

marco-f1的计算方法：将所有类别的precision和recall求平均，然后计算f1值作为macro-F1。marco-f1的使用场景是没有考虑到数据的数量，所以会平等的看待每一类（因为每一类的precision和recall都在0-1之间），会相对受高precision和高recall类的影响较大。

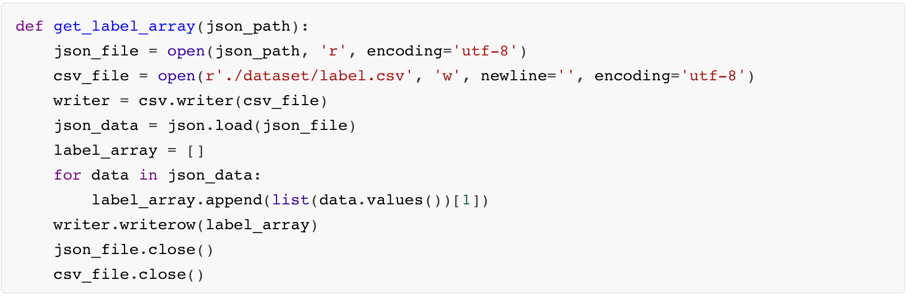
## 三、算法设计

**3.1数据处理**

1.首先将从educoder网站上下载的训练集和测试集的json格式转化成csv格式。转换代码如下所示。

****

2.将训练集的label单独取出来，写入一个csv文件，以便于后续的使用，转换的代码如下所示。



3.一开始没有想到要对数据进行翻译，所以找了19种语言的停用词。大致思路是将新闻文本进行分词，然后消除停用词。

4.在意识到多语言新闻数据应该先进行翻译后，查找翻译的方法。决定采用百度翻译的API进行翻译。在<http://api.fanyi.baidu.com/api/>网站注册登录并申请翻译开发者，并完成翻译部分python代码。

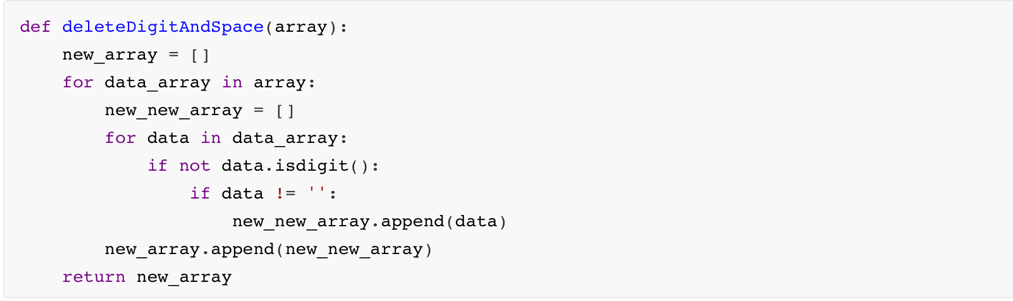
****

可以看到为了爬取数据要定义一系列参数：appId、secretKey是在网站上由注册的账号获取的；其次定义翻译的语言类型、爬取数据的URL等数据。由requests模块对相应的网址发起post请求，得到的结果转化成json类型并读取得到。

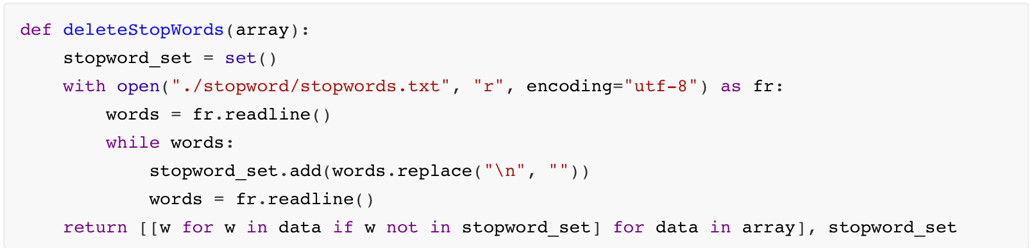
5.利用re模块的split()函数进行数据集的分词。



6.过滤掉分词后有完全由数字组成和空格的词。



7.读取停用词文件得到停用词集合，并对分词进行过滤。

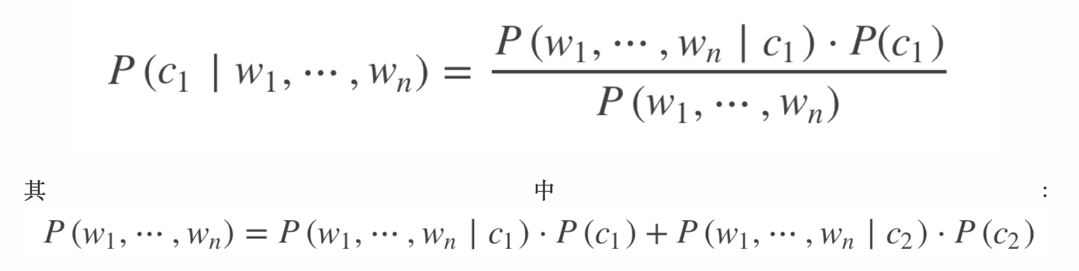


8.读取停用词文件得到停用词数组，在后续算法中使用。

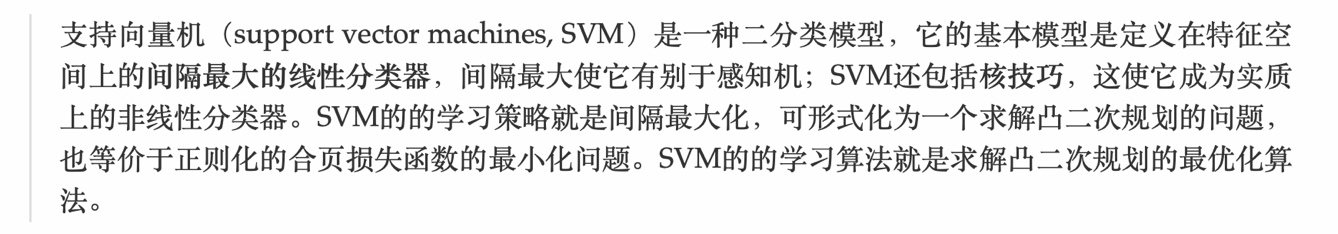


**3.2模型算法**

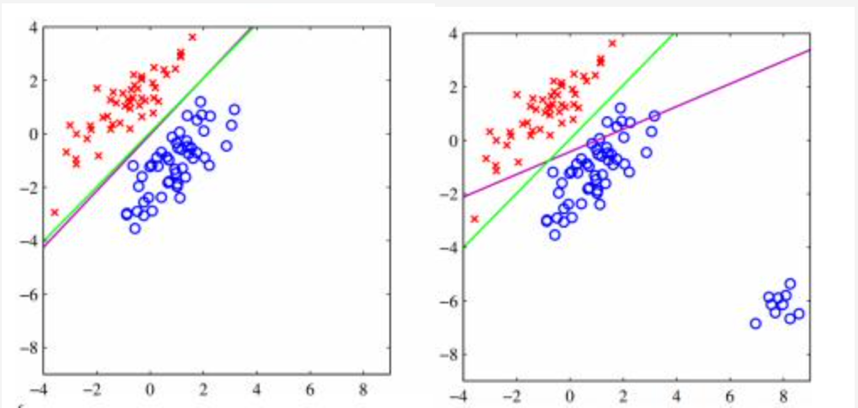
1.情感分类的基本模型是贝叶斯分类。对于有两个类别C1和C2的分类问题来说，其特征为w1、w2、…、wn，特征之间是相互独立的，属于类别C1的贝叶斯模型的基本过程为：



2.

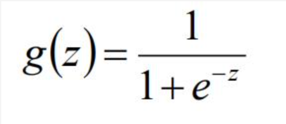


3. 逻辑回归（Logistic Regression）与线性回归（Linear Regression）都是一种广义线性模型（generalized linear model）。logistic回归的因变量可以是二分类的，也可以是多分类的，但是二分类的更为常用，也更加容易解释，多类可以使用softmax方法进行处理。实际中最为常用的就是二分类的logistic回归。Logistic 回归的本质是：假设数据服从这个分布，然后使用极大似然估计做参数的估计。

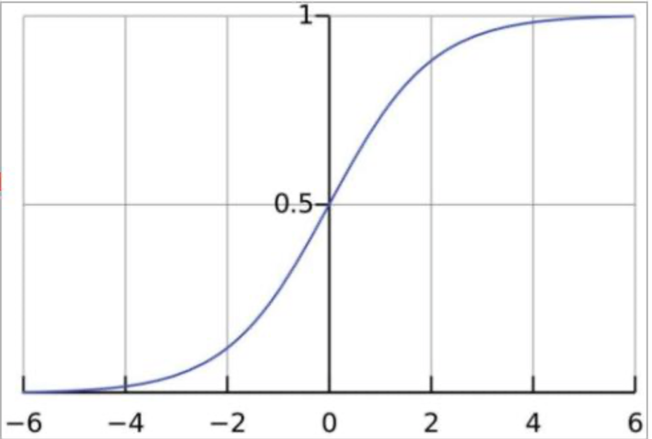


对于线性回归，在角落上加上一块蓝色点之后，线性回归的线会向下倾斜，参考上图右中紫色的线，但是logistics回归仍旧为绿色的线，分类还是很准确。

logistics回归会用到Sigmoid函数，如下所示：



Sigmoid函数的图形如下所示：



然后基于改进的上升梯度算法确定最佳回归系数，其后的公式推导略去，见参考文献。

**3.3模型评估**

通过NLP来进行文本情感分析在市面上已经有了非常成熟的体系，SVM、Logistics、Neural Network等算法都是非常好的方案。而sklearn是一个Python第三方提供的非常强力的机器学习库，它包含了从数据预处理到训练模型的各个方面。所以整体都采用sklearn中提供的模型来进行训练测试，对朴素贝叶斯分类器、逻辑回归和线性分类支持向量机三种模型进行测试。

## 四、实验环境与平台

处理器：2.3 GHz Quad-Core Intel Core i5

操作系统：macOS 11.3.1

实验 IDE：pycharm 2021.2.3

编程语言及其版本：python3.8

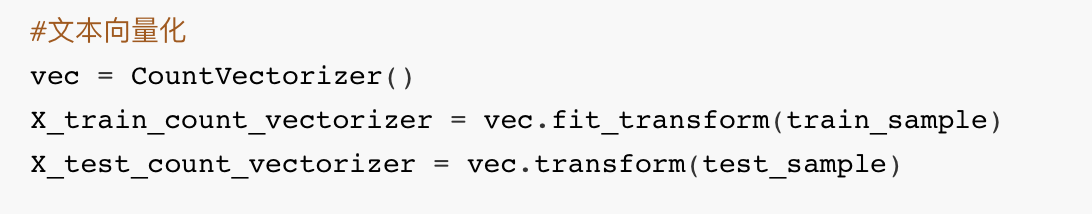
## 五、程序实现

1.朴素贝叶斯实现：

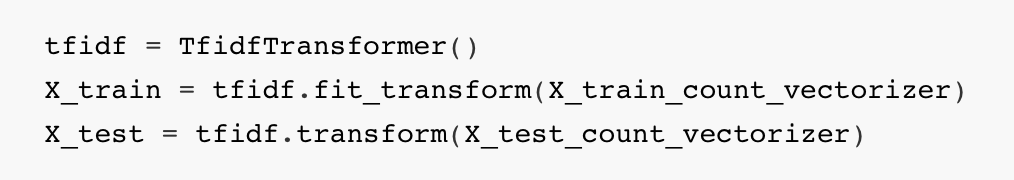


首先对数据进行处理，读取翻译后的数据集和测试集的文本、label的文本，并且都转化成ndarray类型，ndarray是numpy中的n维数组对象，同时删去其中不对应的数据文本。

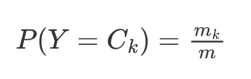
其次进行文本向量化。因为数据集中每一条数据都是很长的一个字符串，我们需要对数据进行向量化的处理，将其中的词变成向量数组。sklearn提供了实现词频向量化功能的CountVectorizer类。代码如下所示：



但是仅仅通过统计词频的方式来将文本转换成向量会出现一个问题，那就是长的文章词语出现的次数会比短的文章要多，但是实际上两篇文章可能谈论的都是同一个主题。为了解决这个问题，我们可以使用tf-idf来构建文本向量。tf-idf是一种用于信息检索与数据挖掘的常用加权技术。tf-idf的主要思想是：如果某个词或短语在一篇文章中出现的频率TF高，并且在其他文章中很少出现，则认为此词或者短语具有很好的类别区分能力，适合用来分类。tf-idf实际上是：TF \* IDF，TF词频(Term Frequency)，IDF逆向文件频率(Inverse Document Frequency)。TF表示词条在文档中出现的频率。IDF的主要思想是如果包含词条t的文档越少，也就是n越小，IDF越大，则说明词条t具有很好的类别区分能力。sklearn中已经提供了tf-idf的接口，构建代码如下所示：



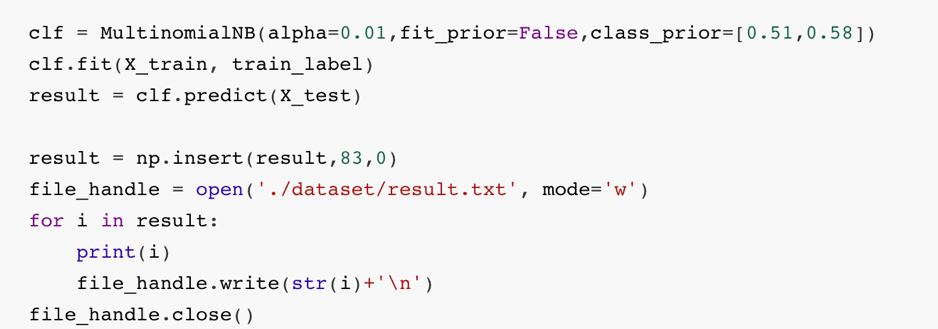
MultinomialNB是sklearn中多项分布数据的朴素贝叶斯算法的实现，并且是用于文本分类的经典朴素贝叶斯算法。在MultinomialNB实例化时alpha是一个常用的参数。alpha为平滑因子。当等于1时，做的是拉普拉斯平滑；当小于1时做的是Lidstone平滑；当等于0时，不做任何平滑处理。fit\_prior为布尔型可选参数，默认为True。布尔参数fit\_prior表示是否要考虑先验概率，如果是False，则所有的样本类别输出都有相同的类别先验概率。否则可以自己用第三个参数class\_prior输入先验概率，或者不输入第三个参数class\_prior，让MultinomialNB自己从训练集样本来计算先验概率，此时的先验概率为



其中m为训练集样本总数量， 为输出为第k类别的训练集样本数。

MultinomialNB类中的fit函数实现了朴素贝叶斯分类算法训练模型的功能，predict函数实现了法模型预测的功能。其中fit函数的参数如下：X的类型为ndarry，存放训练样本；Y值为整型，大小为样本数量的ndarray，存放训练样本的分类标签。而predict函数有一个向量输入：X的类型为ndarry，存放预测样本。

最后用MultinomialNB类进行训练和预测，以及得到结果并写入txt文档代码如下所示：



2.逻辑回归实现：

在读取数据集部分代码同上，不同的是使用不同的类来进行文本处理。

首先利用TfidfVectorizer进行文本向量化，TfidfVectorizer并不适用[朴素贝叶斯算法](https://www.zhihu.com/search?q=%E6%9C%B4%E7%B4%A0%E8%B4%9D%E5%8F%B6%E6%96%AF%E7%AE%97%E6%B3%95&search_source=Entity&hybrid_search_source=Entity&hybrid_search_extra=%7B%22sourceType%22%3A%22article%22%2C%22sourceId%22%3A67883024%7D)，TfidfVectorizer对词项用idf值进行改进，也就是考虑了词项在文档间的分布，也适用于线性模型，同时由于通常线性模型要求输入向量的模为1，因此TfidfVectorizer默认行向量是单位化后的。idf值对频次表示的文本向量进行了改进，它不仅考虑了文本中词项的频次，同时考虑了词项在一般文本上的出现频率，词项总是在一般的文本中出现，表示它可提供的分类信息较少，比如虚词 “的”、“地”、“得”等。TfidfVectorizer在构建词汇表（特征词表）时考虑了词语文档频次，可以通过设置[min\_df](https://www.zhihu.com/search?q=min_df&search_source=Entity&hybrid_search_source=Entity&hybrid_search_extra=%7B%22sourceType%22%3A%22article%22%2C%22sourceId%22%3A67883024%7D)和max\_df来实现通过文档频次进行特征选择。

使用sklearn线性模型中的LogisticRegression类来进行训练分类。LogisticRegression的参数**solver是**优化算法选择参数，只有五个可选参数，即newton-cg,lbfgs,liblinear,sag,saga。默认为liblinear。solver参数决定了我们对逻辑回归损失函数的优化方法。在这里我选择使用 lbfgs。lbfgs为拟牛顿法的一种，利用损失函数二阶导数矩阵即海森矩阵来迭代优化损失函数。

代码如下所示：



3.线性分类支持向量机实现：

对文本的处理同上。

LinearSVC实现了线性分类支持向量机，它是给根据liblinear实现的，可以用于二类分类，也可以用于多类分类。其原型为：class Sklearn.svm.LinearSVC(penalty=’l2’, loss=’squared\_hinge’, dual=True, tol=0.0001, C=1.0, multi\_class=’ovr’, fit\_intercept=True, [intercept\_scaling](https://www.zhihu.com/search?q=intercept_scaling&search_source=Entity&hybrid_search_source=Entity&hybrid_search_extra=%7B%22sourceType%22%3A%22article%22%2C%22sourceId%22%3A57162092%7D)=1, class\_weight=None, verbose=0, random\_state=None, max\_iter=1000)。

我也采用SVC类进行测试但是其结果没有LinearSVC来的高。

代码如下所示：

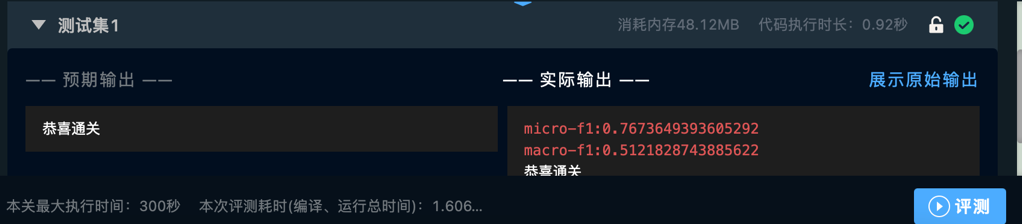


## 六、实验结果

1.朴素贝叶斯测试结果：



2.逻辑回归测试结果：

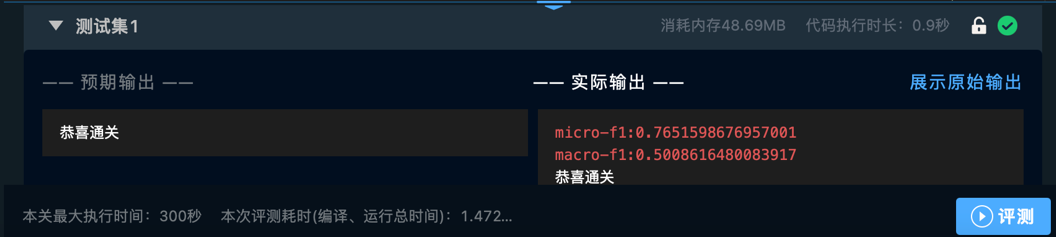


3.支持向量机测试结果：

（1）LinearSVC



（2）SVC



## 七、结果分析

从最后结果来看，其macro-f1最高能达到0.737左右，而SVC表现得最差只能达到0.50左右。首先这和调整参数有一部分的关系，在逻辑回归和支持向量机中我都没有特别去优化参数，因为参数毕竟还是面向训练集和测试集的。其次这也和模型本身有关，这提醒我们在新闻情感分析过程中，选取一个合适的模型是非常重要的。

其实机器学习来分析文本的情感有些时候是个伪命题，某句话的情感在不同的人心中可能是完全不一样的。所以人工打标签的时候也是受制于个人的偏向影响的。但是在这一门课的选题，帮助我认识到了分类器是如何工作的事，通过贝叶斯等分类方法来进行特征的分类与聚合，将视线从纯粹的代码逻辑拉回到数学逻辑，这让机器学习这门课的意义在我们这里意义变得更高了。

## 八、参考文献

1.逻辑回归：<https://www.freesion.com/article/13841252046/>

2. LogisticRegression-参数说明：https://blog.csdn.net/jark\_/article/details/78342644