基于机器学习的评论情感分析

涂新宇 170141879

（1 上海师范大学信息与机电工程学院， 上海 201418）

*摘要*—本文使用Kaggle平台上的Google Play Store Apps数据集，利用经典的机器学习算法，例如逻辑回归，随机森林，朴素贝叶斯等算法，对谷歌应用商店的带有不同情感的评论进行分析，其中使用拉普拉斯修正对朴素贝叶斯算法进行了优化。经过实验分析得出，经过优化的朴素贝叶斯算法对评论的情感特征可以有较高准确率的准确度，制作的图形化评论分析系统可以使用三种算法对评论进行情感分析，具备一定的实用性。

关键词—机器学习；数据挖掘；朴素贝叶斯；随机森林；

# 引言

情感分析是自然语言处理（NLP）领域的一个任务，又称倾向性分析，意见抽取，意见挖掘，情感挖掘，主观分析等，它是对带有情感色彩的主观性文本进行分析、处理、归纳和推理的过程。按照处理文本的粒度不同,情感分析大致可分为词语级、句子级、篇章级三个研究层次。

文本情感分析的一个基本步骤是对文本中的某段已知文字的两极性进行分类，这个分类可能是在句子级、功能级。分类的作用就是判断出此文字中表述的观点是积极的、消极的、还是中性的情绪。更高级的“超出两极性”的情感分析还会寻找更复杂的情绪状态，比如“生气”、“悲伤”、“快乐” 等等。

在文本情感分析领域，早期做出研究贡献的有 Turney和 Pang他们运用了多种方法探测商品评论和电影影评的两极观点。此研究是建立在文档级所进行的分析。另一种文档意见的分类方式可以是多重等级的，Pang和 Snyder(among others):延伸了早先的基础两极意见研究，将电影影评分类并预测为3至4星的多重级别，而 Snyder就餐馆评论做了个深度分析，从多种不同方面预测餐馆的评分，比如食物、气氛等等 （在一个5星的等级制度上）。尽管在大多数统计方面的分类方式中，“中性” 类是经常被忽略的，因为“中性”类的文本经常是处于一个两极分类的边缘地带，但是很多研究者指出，在每个两极化问题当中，都应该识别出三个不同的类别。进一步的说，一些现有的分类方式 例如 Max Entropy和 SVMs可以证明，在分类过程中区分出“中性”类可以帮助提高分类算法的整体准确率。本文在二分类的基础上，使用消极、中性、积极三个分类，使用kaggle上的开源数据集，对词语级别情感分析的进行基础的学习和实践。

# 相关工作

## 随机森林算法

随机森林是一种灵活的、便于使用的机器学习算法，即使没有超参数调整，大多数情况下也会带来好的结果。它可以用来进行分类和回归任务。其优点主要有：

### 可以用来解决分类和回归问题：随机森林可以同时处理分类和数值特征；

### 抗过拟合能力：通过平均决策树，降低过拟合的风险性；

### 只有在半数以上的基分类器出现差错时才会做出错误的预测：随机森林非常稳定，即使数据集中出现了一个新的数据点，整个算法也不会受到过多影响，它只会影响到一颗决策树，很难对所有决策树产生影响。

## 逻辑回归

Logistic 回归是二分类任务中最常用的机器学习算法之一。它的设计思路简单，易于实现，可以用作性能基准，且在很多任务中都表现很好。Logistic 回归的工作原理是通过使用其固有的 logistic 函数估计概率，来衡量因变量（我想要预测的标签）与一个或多个自变量（特征）之间的关系。其主要优点有；

### 预测结果是界于0和1之间的概率；

### 可以适用于连续性和类别性自变量；

### 容易使用和解释。

## 朴素贝叶斯

朴素贝叶斯法是基于贝叶斯定理与特征条件独立假设的分类方法。他在贝叶斯算法的基础上进行了相应的简化，即假定给定目标值时属性之间相互条件独立。也就是说没有哪个属性变量对于决策结果来说占有着较大的比重，也没有哪个属性变量对于决策结果占有着较小的比重。虽然这个简化方式在一定程度上降低了贝叶斯分类算法的分类效果，但是在实际的应用场景中，极大地简化了贝叶斯方法的复杂性。他主要的优点为：

### 健壮性比较好，对于不同类型的数据集不会呈现出太大的差异性

### 当数据集属性之间的关系相对比较独立时，朴素贝叶斯分类算法会有较好的效果。

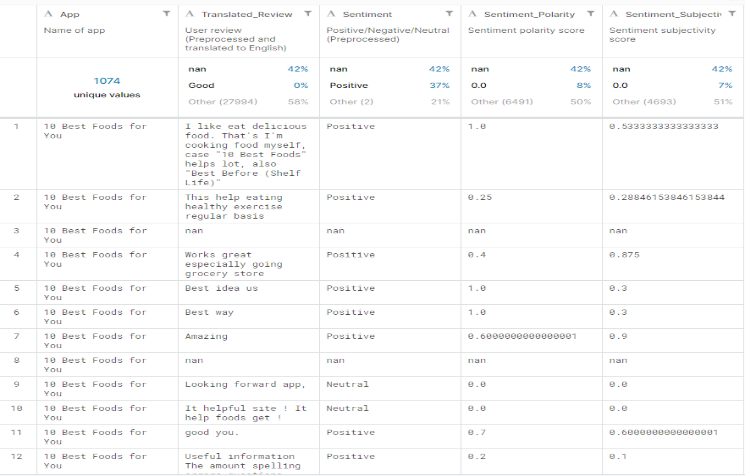
# 数据预处理

## 测试环境

为了评估算法的性能，在Windows10 专业版操作系统中安装pycharm，并采用 Python 语言编程实现了相关算法，其配置为:i5-8200U CPU@2.20 GHz，GPU 为 NVIDA GeForce 940 M，8.00 GB 内存。

## 数据展示

本文使用的具体数据集如图1所示，数据集选自kaggle的Google Play Store Apps，数据集包括9960个应用程序的类别，评分等内容，以及用户在使用不同软件后的评论、评论带有情感、主观性以及极性的数据内容，由于只是对之后用户新输入的评论的情感进行预测。因此舍弃掉应用程序的类别版本等无关内容。

图1数据展示

## 数据清洗

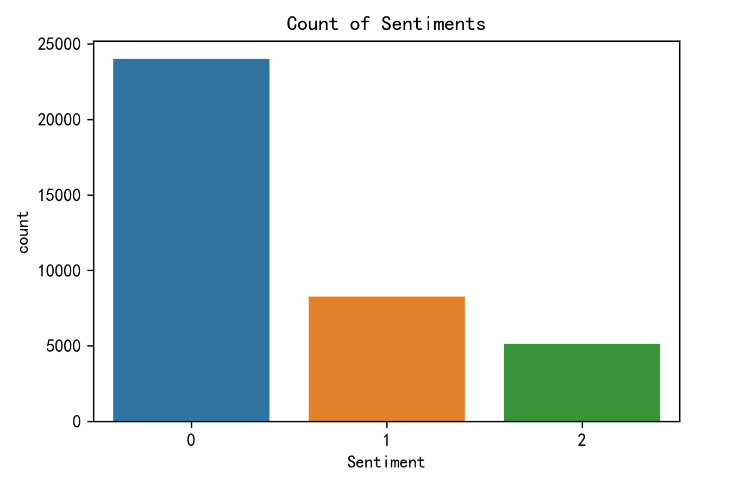
首先对收集到的所有数据的情绪进行分类并统计不同情绪的用户数量（并对标称型数据修改为数值型，将积极标记为0，消极标记为1，中立标记为2）

图2 评论属性

由于评论中含有大小写英文字符、停用词（我在后面用nltk中的库直接去除）、标点符号等与情感分类无关的内容，因此需要对整体数据集进行规格化操作。

### 数据标准化的操作内容有：

### 去除标点符号，使用正则化库中的re.sub方式将不是a-z和A-Z的字符用空格替换。

### 将所有的字母统一小写。

### 遍历所有评论，将不同的评论存放在不同的字符串中，并放入统一的列表当中。

## 分词处理

NLTK（www.nltk.org）是在处理预料库、分类文本、分析语言结构等多项操作中最长遇到的包。其收集的大量公开数据集、模型上提供了全面、易用的接口，涵盖了分词、词性标注(Part-Of-Speech tag, POS-tag)、命名实体识别(Named Entity Recognition, NER)、句法分析(Syntactic Parse)等各项 NLP 领域的功能。

因此我在对评论进行词向量映射、分词、特征对齐和去除停用词时使用了nltk.WordNetLemmatizer()和nltk.CountVectorizer()两个方法。

此处之所以选择词向量而不使用独热编码是由于不同的单词的量很多，如果使用独热编码会造成维度灾难，且会形成一个极大的稀疏矩阵。其次，为了解决评论的句子长短不一的情况，此处我拟定每一句话有1000个单词，如果没有单词，则用0进行填充，这样每一句评论都有1000个特征。

## 处理缺失值

使用dropna()将缺失值去除，因为如果没有评论的话我也无法对其情感进行分析处理。到这里我就解决了数据清洗的所有步骤。

# 算法实现

主要实现了优化的朴素贝叶斯算法，其主要步骤如下：

### 计算不同情感的评论中单词出现的次数

### 计算出现每个单词的拉普拉斯平滑后被分到不同情感中的先验概率

### 计算分类结果

## 计算不同情感的评论中单词的频次

### 将reviews中情感为积极、消极、中立的三种评价的80%分别用列表传入自定义单词出现次数的计算方法中。

### 创建字典count，遍历reviews中所有单词并对其进行计数操作。为了数据平衡，我将单词出现次数去除以该情感类别评论的总个数。

### 返回字典count，此时count中存储着传入时的情感的单词出现次数。

## 计算先验概率

### 将积极positive\_dict、消极negative\_dict、中立neutral\_dict三个字典分别传入自定义拉普拉斯平滑先验概率方法中。其中每个字典中存储着该分类情况下不同单词出现的次数。

### 定义积极概率positive\_probability、消极概率negative\_probability、中立neutral\_probability概率三个字典。其中概率字典表示，出现的单词被分为该类拉普拉斯平滑后的最大似然估计值。

### 将积极、消极、中立字典存入all\_dict列表中。

### 将积极、消极、中立概率字典存入all\_probability列表中。

### 用两个for循环遍历all\_dict中的每一个字典中的键值：

#### 如果该键值出现在另外两个字典中，则向该字典对应的概率字典中存入拉普拉斯变换后的概率值，拉普拉斯计算方式下：

#### 其中N为总分类数，Dc是一个单词被分为该类的次数，D是这个单词出现的次数。

#### 如果该键值出现在另外一个字典中，则向该字典对应的概率字典存入拉普拉斯变化后的概率，并向没有出现的那个字典中加入1/20，此处我对拉普拉斯平滑做了一些更改，因为这里的分类数只有3种，如果存入1/3可能会导致这个单词的总概率超过1很多，而如果存入一个非常小的数容易使得文章一旦出现这个词，就不可能被分到没有这个词的那一类中，选择20是一个经验值的考虑，也是我设定的超参数。

图3 预测结果

#### 以上是设置不同值时的预测结果。可以看出当数值越大，准确率逐渐攀升，然而在跨过0.01之后准确率上升减缓，因此我选择中间值0.01来作为本次拉普拉斯的参数。

#### 如果该键值对并没有出现在其余两个字典中，则在该类的概率中存入1/2，其余两类中存入1/4。

### 将all\_probability返回.

## 计算分类结果

### 将剩余20%数据和方法2中的all\_probability传入，将数据中的Translated\_Reviews存入reviews中，将数据中的Sentiment存入labels中。

### 创建评分列表score=[0,0,0]，列表中第一个值为积极的得分，第二个值为消极的得分，第三个值为中立的得分。

### 创建一个空的列表result，用于存储预测的结果。

### 遍历reviews中所有的评论，对每一条评论进行以下操作：

#### 数据标准化，即去停顿词，去标点符号，大写转换为小写以及分词操作

#### 对之下的操作循环三次，其中计数器为i，i∈（0，2），即对积极、消极和中立的概率进行计算：

遍历评论中的所有单词：

如果在传入的字典all\_probability[i]中有该单词，防止数值下溢，我不直接用乘积的形式计算概率，二十通过向score[i]加上该单词概率的对数来获得该分类的评分

### 选出最大的score并标注为对应的标签，积极为0，消极为1，中立为2

### 绘制热力图

# 实验结果

## 随机森林

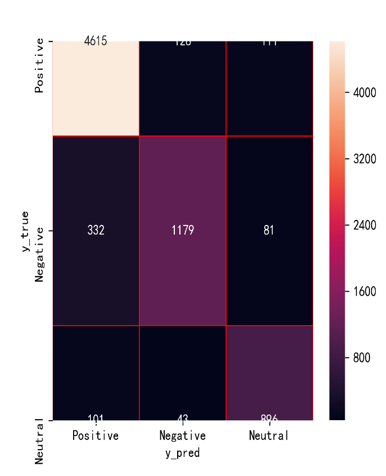
首先我调用了随机森林算法进行分类，数据集在被打乱后划分为80%训练集，20%测试集，其中随机森林使用了10个决策树，其分类后的热力图如下：

图4 随机森林热力图

准确度为0.8936

## 逻辑回归

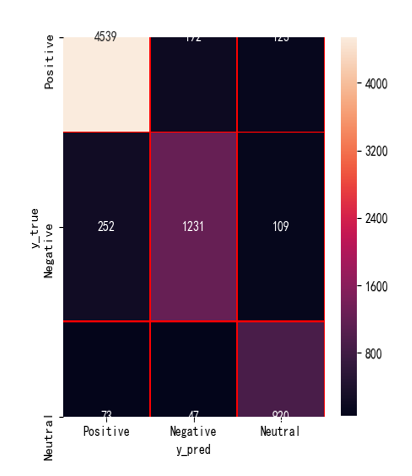
然后我调用了逻辑回归算法进行分类，数据集在被打乱后划分为80%训练集，20%测试集，其分类后的热力图如下：

图5 逻辑回归热力图

准确度为0.9010

## 朴素贝叶斯

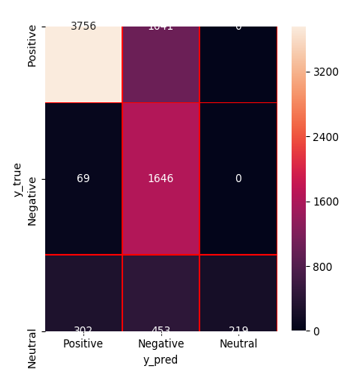
我使用了80%的数据集去构建字典，对剩余的20%的数据集进行预测，准确度为0.7339，预测后获得热力图结果如下

图6 朴素贝叶斯热力图

不难看出该算法倾向于将评论分类为积极的，为此，我对实验中的部分结果进行打印分析，其中下面的三个小数是获得的评分。在我的算法中，我选出的是评分中的最大值，也就是把helpful分类至0即积极类中，这并不是错误的

图7 部分结果打印

# 图形界面

为了更好地展示该算法地预测成果，我将模型训练的数据用sklearn库中的joblib包存成了对应的pkl文件，以方便进行预测，使用者可以通过选取算法，输入评论，点击？的预测按钮获取返回值。图7即为我的图形界面。

图8 GUI界面

# 总结与展望

本文中使用的比较算法都是在多次试验后选取的最佳模型进行对比，有很多分类模型在NLP处理上有着极大的偏差。因此即使准确率有所不如的情况下，并不能直接推导出本文自身算法有问题，为了更好的展现本文算法的优势，我再拿一个例子予以阐释，以图8为例，我发现评论关于省钱的言论也被人工分类成为中立的情绪，而在部分人的认知中这种类型的句子也可能被分类为积极情绪的句子，这只是我在7000余条数据中采样25条左右的数据获得的反馈，因此我可以发现，人工标注情感标签的时候还是会存在歧义性，或许直接按照已有的标签去进行直接分类虽然效果好，但是难以解决人工标注的误差，本文拟借此抛砖引玉，往引出更多专业人才对其进行深入研究。

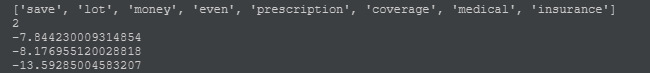
另外，在深度学习如此发达的现在，RNN和LSTM模型对于NLP处理效果相当明显，然而由于笔者技术原因未能使用RNN跑出理想的准确率，也希望在未来能够使用深度学习的相关算法继续对NLP进行深入的研究。

图9 存在分歧

##### 参考文献

[1]李鑫,李晓戈.基于深度学习的APP-Store评论情感分析[J].计算机与数字工程,2019,47(10):2417-2422+2433.

[2]喻影,陈珂,寿黎但,陈刚,吴晓凡.基于关键词和关键句抽取的用户评论情感分析[J].计算机科学,2019,46(10):19-26.

[3]杨玉娟,袁欢欢,王永利.针对评论文本的情感分析方法[J].南京理工大学学报,2019,43(03):280-285+291.

[4]魏韡,向阳,陈千.中文文本情感分析综述[J].计算机应用,2011,31(12):3321-3323.

[5]赵妍妍,秦兵,刘挺.文本情感分析[J].软件学报,2010,21(08):1834-1848.