



辽宁工程技术大学

# 毕 业 设 计 （ 论 文 ）

题 目 亚马逊商城评论文本情感分析——基于 Kaggle 竞赛数据

学 院 名 称 工商管理学院

专 业 班 级 信息管理与信息系统17-12

学 生 学 号 1710030202

学 生 姓 名 薄靖凯

指 导 教 师 温廷新

教务处制

中文题目：亚马逊商城评论文本情感分析——基于 Kaggle 竞赛数据

外文题目：AMAZON REVIEW TEXT SENTIMENT

ANLAYSIS-BASED ON KAGGLE COMPETITION

DATA

毕业设计（论文）共 71 页（其中外文文献及译文 8 页），图纸 0 页

完成日期 2021 年 6 月

答辩日期 2021 年 6 月

辽宁工程技术大学

本科毕业设计（论文）学生诚信承诺保证书

本人郑重承诺：《亚马逊商城评论文本情感分析——基于 Kaggle 竞赛数据》毕业设计（论文）的内容真实、可靠，系本人在\_\_\_\_\_指导教师的指导下，独立完成。如果存在弄虚作假、抄袭的情况，本人承担全部责任。

学生签名：

年 月 日

辽宁工程技术大学

本科毕业设计（论文）指导教师诚信承诺保证书

本人郑重承诺：我已按学校相关规定对\_\_\_\_\_同学的毕业设计（论文）的选题与内容进行了指导和审核，确认由该生独立完成。如果存在弄虚作假、抄袭的情况，本人承担指导教师相关责任。

指导教师签名：

年 月 日

# 摘要

文本情感分析是自然语言处理领域中的研究热点之一，如何更高效的帮助人们在大数据环境下处理海量数据信息（文本信息），避免浪费大量人力物力导致低效率，提取有效数据挖掘客户对于商品评价的真实感受的问题。

本设计从模型方法角度以食品评论情感倾向性分析为研究方向，首先通过Kaggle 竞赛数据集亚马逊食品评论数据，利用数据挖掘手段对数据集进行探索性分析与数据清洗，构建分词模型进行文本特征提取；其次提出基于网格搜索优化-逻辑回归文本情感倾向性分类模型以及基于随机搜索算法优化的多层感知机文本情感倾向性分类模型；最后对比两种优化模型并进行交叉验证，模型结果测试准确率逻辑回归分类模型的正确率为 91%，93%，99%，99%，多层感知机模型的准确率为 85.86%；为食品评论情感倾向性分析提供改进模型和参考意见。

**关键词：**文本情感分析；逻辑回归；MLP；交叉验证

# Abstract

This sentiment analysis is the focus in the field of natural language processing, how to more effectively help people process massive amounts of information (text information) in a big data environment, avoid data waste, a lot of manpower and material resources leading to inefficiency, and extract effective data for the truthfulness of customer product evaluations The question of feeling.

This design is based on the analysis of the emotional tendency of food reviews from the perspective of the model. First, through the Kaggle competition data set Amazon food review data, using data mining methods to conduct exploratory analysis and data cleaning on the data set, build a word segmentation model for text feature extraction; secondly, propose a text sentiment tendency classification based on grid search optimization-logistic regression Model and a multi-layer perceptron text sentimental classification model based on random search algorithm optimization; Finally, the two optimized models are compared and cross-validated. The accuracy of the model results is tested. The accuracy of the logistic regression classification model is 91%, 93%, 99%. %, 99%, the accuracy rate of the multi-layer perceptron model is 85.86%; it provides an improved model and reference opinions for the analysis of emotional tendency of food reviews.

**Key Words:** Text sentiment analysis; logistic regression; MLP; cross-validation

# 目录

摘要.....	I
Abstract.....	II
1 绪论.....	1
1.1 研究背景及意义.....	1
1.2 国内外研究现状.....	2
1.2.1 评论文本情感倾向性分析现状.....	2
1.2.2 商品评论信息挖掘现状.....	3
1.3 设计研究框架.....	4
2 问题模型设计.....	6
2.1 商品评论情感倾向性分析问题难点及解决办法.....	6
2.2 食品评论情感倾向性分析技术路线.....	6
2.3 模型构建性能难点.....	8
3 模型构建基础.....	9
3.1 文本特征选择——词袋模型 Bag of Word.....	9
3.2 基于网格搜索优化的逻辑回归分类模型.....	10
3.2.1 网格搜索优化算法.....	11
3.2.2 基于 L1, L2 的正则化.....	12
3.3 多层感知机模型.....	13
4 基于亚马逊食品评论情感倾向性分析模型构建.....	15
4.1 数据探索 (EDA) .....	15

4.1.1 数据集.....	15
4.1.2 探索性数据分析.....	16
4.1.3 数据清洗.....	27
4.2 文本情感倾向性分类模型构建.....	29
4.2.1 词袋模型构建.....	29
4.2.2 文本分类的机器学习算法.....	30
4.2.3 文本分类的深度学习算法.....	30
4.2.4 模型优化.....	31
5 模型实验效果预测及验证分析.....	32
5.1 预测效果.....	32
5.2 模型验证——交叉验证.....	33
5.3 研究启示.....	40
6 总结与展望.....	41
6.1 总结.....	41
6.2 不足和展望.....	42
致谢.....	42
参考文献.....	43
附录 A 译文.....	45
附录 B 外文文献.....	56
附录 C.....	70

# 1 绪论

## 1.1 研究背景及意义

随着互联网的发展，网上的信息量日益增多，各大门户网站，电商网站，视频网站等逐步成为人表达情感与交流思想的线上平台。近年来，电子商务平台呈现百花齐放，百家争鸣，国外有亚马逊等电商巨头，国内有阿里巴巴，京东商城等电商平台激烈竞争，拼多多也强势崛起，2019 年全国电子商务交易额达 34.81 万亿元，比上年增长 10%。其中商品服务类电商交易额达 33.76 万亿元，比上年增长 10%。在巨大的交易量同时也产生了大量的用户评论数据。这些评论数据反映了用户对平台，商家，物流及商品的态度，如何从这些海量的数据中提取有用的信息，并给不同的用户提供个性化的服务正在成为业界和学术界关注的重点。例如，在某些购物网站上，许多的潜在用户可以通过浏览已经购买了某些产品的客户对这些产品给出的相关评价信息来评估是否继续购买这类产品：在微博中，可以通过对热点事件的相关信息挖掘、处理和分析，从而有效地对社会舆论的整体发展趋势进行监控：商业上，也有很多的生产厂商通过挖掘和分析消费者对产品给予的相关评论对该产品的未来生产计划作出相应的决定等等。对于文本信息的挖掘、处理和分析所应用到的就是文本情感分析研究中的方法。

评论文本情感分析问题广泛应用于蓬勃发展的互联网时代是自然语言处理研究的一个重要方向，用户针对产品的各项特点进行评论分析进行数据挖掘然后进一步有效的探索分析，从而将处理的信息整理归纳，这类技术代表着一个规模巨大的问题领域。作为新兴的领域，除了应用机器学习算法分类，可以结合深度学习方法进行分析，利用大数据量条件下构建神经网络模型，增加更多探索性应用，。当前文本情感分析技术是依赖于评论倾向性分析以及分词模型为基础上的，主要包含：情感的主客观分类，评论情感的态度乐观或消极，关键词的选取，分词模型等应用，文本情感分析的应用主要包括以信息抽取为基础的关键词以及评价搭配抽取等。而构建文本情感倾向性分析的模型一般是分类，回归相关模型。这类模型模型结构简单，准确率相对于较低，稳定性差，需要通过大量前期数据清洗整理，构建模型训练整理加工后的数据才会有较好的准确率。本设计的研究内容为，对亚马逊文本食品评论数据进行探索分析，并在其基础上建立机器学习模型，构建基于随机搜索算法的 MLP 模型并测试其分类效果。



设计采用机器学习和深度学习等自然语言处理手段，通过对文本内容的分析挖掘，提取有价值的信息，这些信息能够分析除消费者对商品的情感倾向，为在线市场提供提高水平的方向，有利于商店的管理和发展。同时通过满意程度也可以衡量商品的价值，在用户进行食品选购时，可以通过有效的评论选择心仪的商品。

如何正确运用文本情感分析的相关技术，互联网商品购买力爆炸式增长，随之产生的大量数据迫切需要计算机分析处理为商家以及用户提供指导帮助，这使得评论文本情感倾向性分析研究具有重要的意义。以食品商品评论情感分析为出发点展开为其他情感文本的趋势导向，开展研究其他方面的情感文本分析，通过该模型分析方法能高效挖掘文本深层价值，避免耗费过多人力物力，避免效率低下且错误率较高。

## 1.2 国内外研究现状

### 1.2.1 评论文本情感倾向性分析现状

情感分析研究现状：文本情感分析分类方法主要分为两大部分，第一部分为文本情感倾向性分析为二元分类问题，第二部分为信息的情感分类为传统二元模型以及进一步探索的多元分类模型。赵妍妍<sup>[2]</sup>等人将文本情感分析分类问题总结为包含主客观信息的二元分类，主观信息的情感倾向性分类以及文本观点分类与挖掘问题，并对文本情感分析领域研究的模型和方法进行总结整理，为该领域开拓了研究领域。

传统的文本情感倾向性分析的目标是面向文本主观情感分类，根据不同的文本粒度主要分为四类，分类的由最初的二元化也可拓展为多元化，其分类的主要依据是主题词（即在文本中重复率最高的词）传统的算法主要是分类模型方法例如 SVM, KNN 等。朱少杰<sup>[1]</sup>在研究文本情感分类使用了传统的 SVM 分析情感倾向性，并在浅层特征的基础上结合深度学习方法，提高情感分类能力。杜鹏<sup>[3]</sup>等人通过自注意力层训练文本中单词之间的关联性，通过卷积层训练特征通过构建基于自注意力和卷积神经网络商品评论文本情感倾向性分析模型并于其他神经网络模型进行对比提出了有效的文本情感分类模型。范昊<sup>[5]</sup>等人提出基于字向

量与循环神经网络构建的二元分类情感分析模型用于提高文本评论情感倾向性分析模型的效果，模型具有较好的健壮性。

当下的情感分析逐渐偏向多元化，根据语法，词性等信息分析文本所面向的情感风格，同理所需的分类算法往往更加复杂，需要模拟更多函数去构建模型。比起传统的面向主题内容情感分类，当下的研究应用更倾向于包含文本的指标之间的详细分析。胡明哲在针对酒店评论文本情感分析将其归类为二分类问题，使用逻辑回归，决策树，随机森林，支持向量机模型进行测试，分析调整情感倾向性分类的准确率，并尝试使用 CNN 和 MLP 神经网络模型在同种数据上进行测试，比对传统机器学习模型效果。刘智鹏<sup>[9]</sup>在情感倾向性分析方面的研究，将传统的机器学习和深度学习模型对特征表达方面进行改进，进行测试。作者基于现有的情感词典进而构建有关互联网电商产品评论相关的情感词典，进而提高了基于词典分类的情感倾向性模型分类效果。从构建文本情感倾向性分析模型方法角度来说，当下的主流方法为传统机器学习模型以及流行的深度学习模型，构建基于机器学习模型分析文本情感倾向性分析，主要的任务为 EDA 和文本特征工程以及分类模型构建。张明辉在分析情感倾向性在商品评论中的应用，总结了情感分析现状。分析了主要的模型方法，基于机器学习的情感倾向性分析是通过选取特征词等方法将文本数据量化矩阵化，或者根据彩带模型，进行文本特征工程，根据文本的词几何，对文本情感倾向性进行预测。深度学习在数据挖掘领域拥有巨大潜力，广泛的应用于各个行业并且带来很好的效果，由于其满足大数据量需求，随着硬件设施的完善使得其效果显著提高，深度学习模型相比传统的机器学习模型更具有自动性，不依赖于特征构建与参数调整，更倾向于自主构造特征输入从而进行情感倾向性分析。

随着多种循环神经网络和卷积神经网络被用于解决情感分析的问题中，在准确率和局限性上带来了较大的提升，情感分析的研究也进入了更深的一步。

### 1.2.2 商品评论信息挖掘现状

随着互联网产业蓬勃发展，大量商品以网络售卖方式发展，越来越多的商品被网上售卖随之产生了大量信息，其中部分信息为商品文本评论信息，应用数据挖掘手段对评论文本数据的情感分析对于帮助商家改进商品问题，预测销量，以

及帮助客户选择推荐商品具有重要意义,为自然语言处理数据挖掘领域的发展提供了很多拓展。李涵昱<sup>[4]</sup>等人通过构建过滤算法以及分类方法,构建自动化抽取商品属性和情感词的商品评论文本的情感倾向性分析模型。彭云<sup>[6]</sup>针对文本评论的特性,设计关键词与其他单词之间的语义关系规则,构建词汇的关联规则,构建基于语义约束的 Latent Dirichlet allocation 文本关键词提取模型,为探究文本词语之间潜在关系提供新思路。丁蔚<sup>[7]</sup>通过构建基于词典和机器学习模型的文本情感倾向性分析模型将情感特征量化对比各类机器学习模型算法性能,为文本情感分析方法减少了主观信息影响提出办法。宋明<sup>[8]</sup>等人基于微博文本评论数据比对多种分类模型,构建基于 Bert 的文本情感倾向性分类模型基础上提出加入损失函数的方法提高在面对困难样本的准确率和召回率。在评论文本数据中,从整体的数据挖掘后可以分析整个评论的舆论趋势,王素格<sup>[9]</sup>等人提出基于混合特征的评论文本分类模型混入时间序列方法在多个时间段训练多个模型抽取特征进行混合,在处理整体评论文本舆论趋势看具有较好的性能和现实意义。在商品评论文本情感分析中如何高效地帮助人们在大数据条件下将问题系统化构建文本情感倾向性分析模型处理现实问题,准确的分析各类型数据信息尤为重要。在自然语言处理中文本情感倾向性分析领域很多学者已经搭建了高效的模型方法从多角度问题进行分析,在处理食品文本评论的情感倾向性分析中,还缺乏很多研究。本设计在学者们的研究成果基础上,对食品评论分析这一方面进行探索研究。

### 1.3 设计研究框架

本设计提出基于机器学习和深度学习的商品评论情感倾向性分析模型,以商品评论文本数据为基准,利用数据挖掘手段对数据集进行探索性数据分析和数据清洗,构建分词模型,建立机器学习和深度学习模型进行情感分类,并通过网格搜索算法调整超参数,随机搜索算法优化深度学习模型,将两种优化模型进行对比,为商品评论情感倾向性分析提出改进模型和参考意见。

第一章,绪论 首先从研究背景及其意义思考,分析目前国内外评论文本情感倾向性分析的相关研究以及商品评论信息挖掘现状和难点。构建本设计的研究框架。

第二章，问题模型设计 构建问题模型设计，分析商品评论情感倾向性分析模型问题难点以及搭建分类模型框架，阐述解决问题模型的思路。

第三章，模型构建基础 详细的阐述本设计涉及的数据挖掘方法，数据分析手段，词袋模型，逻辑回归机器学习模型，MLP 深度学习模型以及优化模型用于评论文本分析的原理。

第四章，模型实验效果预测及验证分析 通过 Kaggle 数据集进行数据处理，探索性分析，构建分词模型，机器学习模型实现，深度学习模型实现及优化等五个方面描述本设计的实验过程。

第五章，总结与展望 通过第四章构建的基于亚马逊食品评论情感倾向性分析模型，分析训练模型的效果，以及下一步研究计划。对评论文本情感倾向性分析研究进行归纳和总结，提出展望进一步深入研究该领域的想法以及现阶段存在的不足。

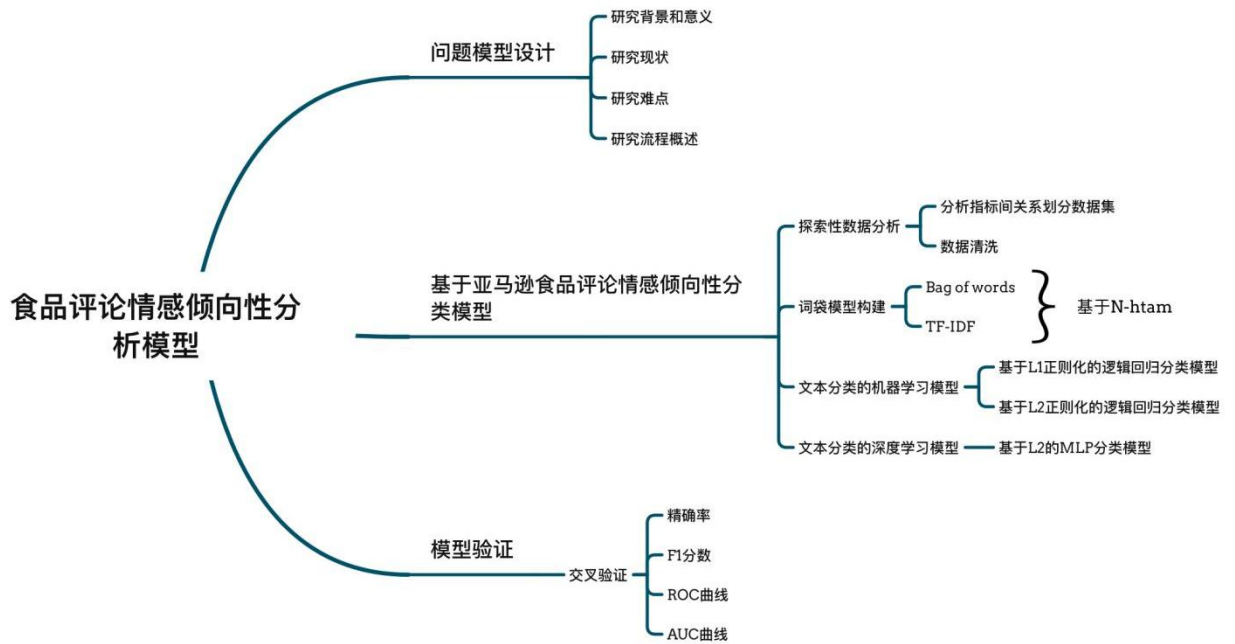


图 1-3 设计体系架构

Figure 1-1 Design system architecture

## 2 问题模型设计

### 2.1 商品评论情感倾向性分析问题难点及解决办法

商品评论情感分析问题主要有三个维度的问题，数据量的处理问题，模型的效果，以及分类效果的适用性。数据量增长快，词汇更新速度快会导致模型的稳定性减弱，复杂度过高需要更高的算力来处理。针对这一问题本设计通过构建多层感知机模型，训练少样本下的全连接神经网络分类模型，能够使用复杂多变，更新快的文本数据进行应用取得较高的分类效果。构建情感倾向性分类模型最关键的就是模型的效果，常用分类器模型其分类效果往往具有局限性，而提高模型效果的核心其实在于模型前期的特征工程，文本特征工程与结构化数据特征工程不同，首先要考虑的是定性或者定量分析。关于定性分析一般通过隶属度函数等算法确认权重的层次分析，在这方面考虑到偏序集算法进行分层，研究缺点显而易见缺乏数据的局限性，主观评价过强。关于定量分析，首先要将文本类型数据转化成结构化数据，将转化的矩阵等形式数据用模型来进行训练。本设计所采用的为定性分析方法，通过两种分词模型方法进行文本特征工程应用于基于正则化的逻辑回归模型，对比模型的效果以及稳定性，构建最佳的情感分类模型。在分类效果的适用性问题上依靠的是模型的健壮性，在评论文本情感分析领域中，评论所包含的内容涉及多个领域，词汇更新快，词汇含义也在同步更新。面临这些影响模型稳定性的因素。本设计使用正则化来对抗干扰因素，针对逻辑回归模型使用 Lasso 和岭回归进行正则化优化，针对 MLP 模型使用 L2 正则化，进而对抗样本过拟合现象。

### 2.2 食品评论情感倾向性分析技术路线

针对如何更高效的帮助人们在大数据环境下处理海量数据信息（文本信息），避免浪费大量人力物力导致低效率，提取有效数据挖掘客户对于商品评价的真实感受的问题。商品评论的信息主要涉及的是客户对在线商城所购买的商品的反馈评价，这些评价中所包含的信息如建议改进，参考意见，产品使用感受等具有大量参考价值。因此分析评论文本情感倾向性分析问题很大程度上帮助了商家了解用户的需求动向，掌握产品的发行趋势如何改进服务提高用户的满意度，帮助客户如何挑选最优商品，具有现实意义。

本设计主要针对商品评论文本的情感倾向性分析问题从模型方法角度以食

品评论情感倾向性分析为研究方向，利用 **Kaggle** 竞赛数据，以亚马逊食品评论数据为例，利用数据挖掘手段对数据集进行探索性分析与数据清洗，构建基于词袋的分词模型，提出基于网格搜索优化-逻辑回归文本情感倾向性分类模型以及基于随机搜索算法优化的多层感知机文本情感倾向性分类模型。

构建基于机器学习的评论文本情感倾向性分析重点在于前期的分词处理部分以及超参数调整，基于机器学习模型的文本情感倾向性分析流程如下图所示：

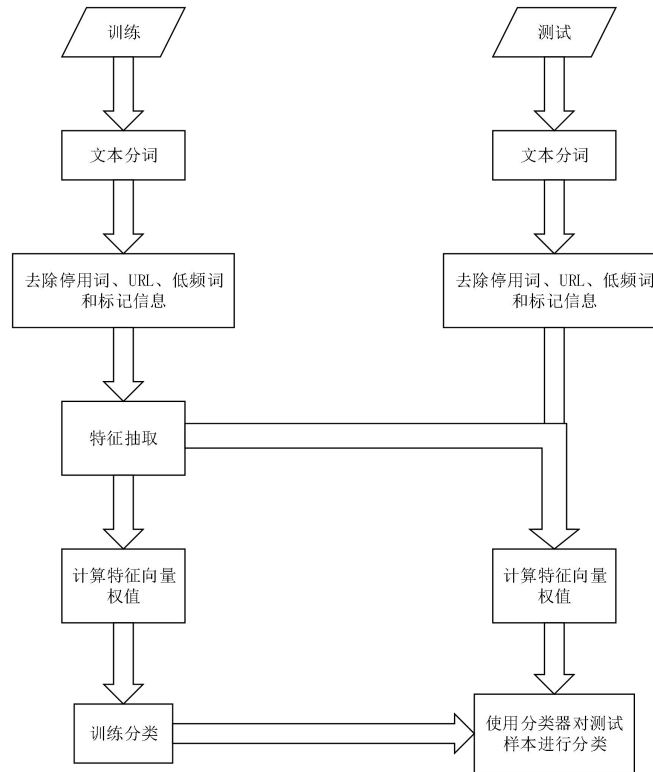


图 2 -1 基于机器学习的情感分析流程

Figure 2 -1 Sentiment analysis process based on machine learning

在深度学习设计中首先进行的是采用 **MLP** 方法进行情感倾向性判断，继而提出 **Random Search Optimizer** 进行优化提高准确率改善分类结果。最后分析结果。设计方案图如下：

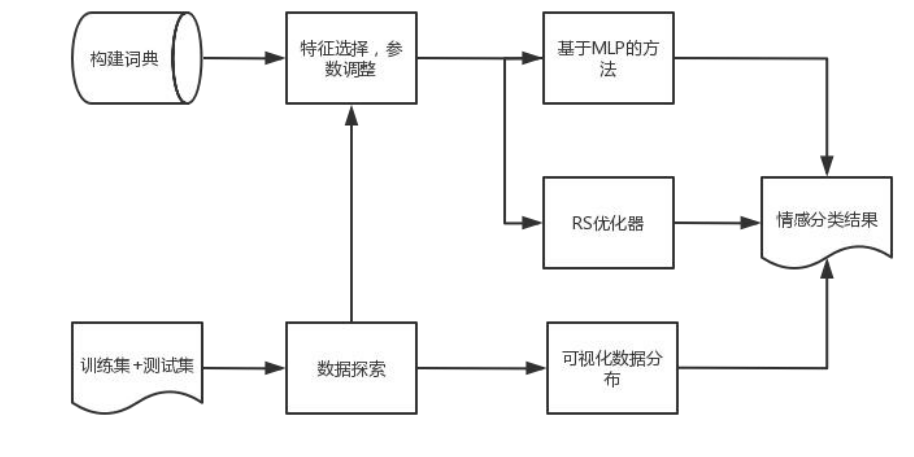


图 2-2：深度学习研究内容的整体设计方案

Figure 2-2: The overall design plan of the deep learning research content

## 2.3 模型构建性能难点

在构建文本情感倾向性分析模型中，前期训练模型的数据尤为重要，模型的性能很大程度上依赖于特征工程的思路是否包含整个问题的核心，在构建食品评论情感倾向性模型中，需要构建文本特征模型将文本数据以结构化数据形式展示，评论文本由复杂多样性的单词组成，涉及领域广，评论范围广，同时单词更新速度快，单词所表示的含义也有不同的变化，同时单词所组成的句子结构也有很多变化类型，句子长短不一导致结构化数据转化时会遇到矩阵过于稀疏等问题。这给构建一种类型的评论文本情感倾向性分析模型的稳定性增加了难度。

在面对诸多困难的电商评论文本情感倾向性分析问题，如何解决这些问题提高模型的稳定性尤为重要。

### 3 模型构建基础

本章主要介绍了本设计使用的文本情感倾向性分析的相关技术以及所涉及的相关理论，本设计处理体系从两方面入手，基于机器学习的情感倾向性分析路线主要步骤是数据处理后的文本特征选择，构建模型，正则化以及网格搜索优化；基于深度学习的情感倾向性分析路线主要是构建多层感知机模型，L2 正则化，以及随机搜索优化。如下所示对相关技术进行了详细的叙述。

#### 3.1 文本特征选择——词袋模型 Bag of Word

词袋模型是一种文本数据提取特征的模型方法，提取特征可应用于训练文本情感倾向性分析模型。这种表现方式不考虑文法以及词的顺序，然后根据袋子里装的词汇对其进行分类。

(1) 在训练集中每一个出现在任意文中的单词分配一个特定的 id（比如，通过建立一个从单词到整数索引的字典）

(2) 对于每个文档 $i$ ，计算每个单词  $w$  的出现次数并将其存储在  $X[i,j]$  中作为特征 $j$  的值，其中  $j$  是在字典中词  $w$  的索引。

通过这种方法中  $n\_features$  是在整个文章集合中不同单词的数量这个值一般来说超过 100,000。如果  $n\_samples == 10000$ ，存储  $X$  为“float32”型的 numpy 数组将会需要  $10000 * 100000 * 4 \text{ bytes} = 4\text{GB}$  内存，在当前的计算机中非常不好管理的。在  $X$  数组中大多数的值为 0，是因为特定的文档中使用的单词数量远远少于总体的词袋单词个数。因此我们可以称词袋模型是典型的 high-dimensional sparse datasets（高维稀疏数据集）。我们可以通过只在内存中保存特征向量中非 0 的部分以节省大量内存。scipy.sparse 矩阵正是能完成上述操作的数据结构，同时 scikit-learn 有对这样的数据结构的内置支持。本设计将通过 TF-IDF 和词袋模型模型进行分词训练。

TF-IDF（词频-逆文本频率）：在处理长文本和短文本时，长文本相对于短文本有更高的单词平均出现次数，尽管他们可能在描述同一主题。TF-IDF 可以避免这些潜在差异，只需将各文档中每个单词的出现次数除以该文档中所有单词的总数：这些新的特征为  $tf$ ，在词频的基础上改良是，降低在该训练文本中的很多文本中均出现的单词的权重，从而突出那些仅在该训练文集中在一小部分文档中出现的单词的信息量。



$$\text{TF(词频)}: \text{TF}(w) = \frac{\text{单词 } w \text{ 在文本中出现的次数}}{\text{文本的单词总数}} \quad (3-1)$$

$$\text{IDF (逆文本频率)} \text{ IDF}(w) = \log \left( \frac{\text{语料库中文本的总数}}{\text{包含词 } w \text{ 的文本数}+1} \right) \quad (3-2)$$

$$\text{TF-IDF 方程式: } \text{TF-IDF}(w) = \text{TF}(w) * \text{IDF}(w) \quad (3-3)$$

其中词频-逆文本频率越高文本的区分度越高，该词即可作为文本的关键词。

**N-gram:** N-gram 是一种语言模型 (Language Model)。它是基于概率与数理统计方法的判别模型，通过计算文本数据中单词的排列顺序组成该段落的概率，即集合单词的联合概率。N 代表由 N 个单词组成的集合，常见 N-gram 模型分为一元，二元，三元。元数代表组成单词的数量。本设计使用一元模型和二元模型进行分词处理。下列展示 N-gram 数学理论。

$$\text{F-gram: } P(w_1, w_2, \dots, w_m) = \prod_{i=1}^m P(w_i) \quad (3-4)$$

$$\text{BI-gram: } P(w_1, w_2, \dots, w_m) = \prod_{i=1}^m P(w_i | w_{i-1}) \quad (3-5)$$

通过 sklearn 库中的 CountVectorizer 的参数 N-gram\_range，可以调用其进行文本分析。CountVectorizer 是有 python 中 scikit-learn 库提供的接口，根据每个此在整个文本中出现的频率（计数）将给定文本转换为向量支持单词或者连续字符的 N-gram 模型的计数，一旦拟合，向量化程序就会构建一个包含特征索引的字典。

TD-IDF 表示词频-逆文档频率。通过对数因子给予 TF-IDF 低分数单词来惩罚语料库中太丰富或太稀有的单词，基于词在语料库中的频率的统计数据，关注语料库中出现的单词的频率，还提供了单词的重要性，然后通过删除对分析不太重要的词从而减少输入维度来降低模型构建的复杂性。

两种将文本数据结构化的方法各有一定的缺点 CountVectorizer 无法识别分析更重要或者不重要的词，只统计最重要的单词无法识别单词之间的关系，而 TF-IDF 无法提供有关单词的信息（真实含义，与其他单词的相似度等）。

## 3.2 基于网格搜索优化的逻辑回归分类模型

逻辑回归时一种统计模型，其基本形式使用逻辑函数对二元因变量进行建模，存在许多更复杂的扩展。在回归分析中，逻辑回归是估计逻辑模型（二元回归）的参数，它是用于分类问题的线性回归模型的扩展。

逻辑回归模型不是拟合直线或超平面，而是使用逻辑函数来挤压[0,1]之间的线性方程的输出，逻辑函数定义为：

$$\text{logistic}(\eta) = \frac{1}{1+\exp(-\eta)} \quad (3-6)$$

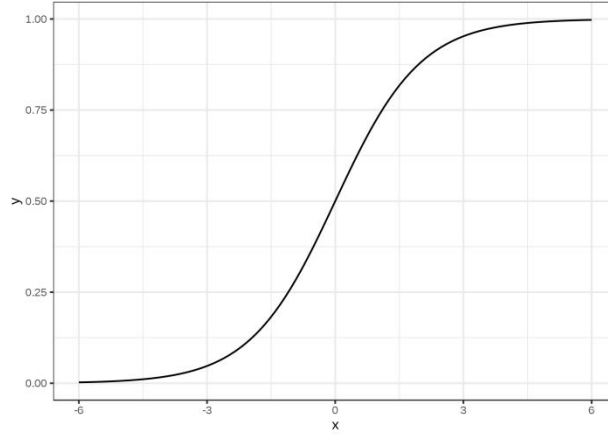


图 3-1 回归曲线

Figure 3-1 Regression curve

如上图所示：逻辑函数。它输出[0,1]之间的数字。在输入 0 处，它输出 0.5。在标准线性回归模型中，通常利用线性方程对结果和特征之间的关系进行建模

$$y^{(i)} = \beta_0 + \beta_1 x_1^{(i)} + \dots + \beta_p x_p^{(i)} \quad (3-7)$$

对于分类，可以将等式的右侧映射到逻辑函数中，此时将会强制输出仅采用[0,1]之间的值：

$$P(y^{(i)} = 1) = \frac{1}{1+\exp\left(-(\beta_0 + \beta_1 x_1^{(i)} + \dots + \beta_p x_p^{(i)})\right)} \quad (3-8)$$

在拆分 8/2 训练测试集后，我们通过逻辑回归分类器使用输入特征的加权组合并将他们传递给一个逻辑函数将所有实数输入转换为[0,1]之间的数字。在应用词袋构建模型的基础上利用 n-gram 和 TF-IDF 特征上应用逻辑回归分类器来比较他们的准确度分数。在默认参数上构建模型为我们提供准确度分数。然但是当特征指标高于数据量时，模型会出现欠拟情况。此时通过引入超参数的其他约束，尝试不同的值组合来找到具有最低错误率的模型，本设计引用 GridSearch，在逻辑回归中，GridSearch 决定了正则化的数量，较低的值会增加正则化。

### 3.2.1 网格搜索优化算法

在构建机器学习模型时，模型的性能的好坏，稳定性的强弱不仅由模型本身决定还有一些其他因素可以影响模型的性能如超参数。超参数，即预先配置的不

直接在分类器内学习的参数，在模型训练之前由模型的调用者提供。在 `scikit-learn` 包中，他们作为评估器类中构造函数的参数进行传递。如下图所示展现混入超参数建模的三个部分。

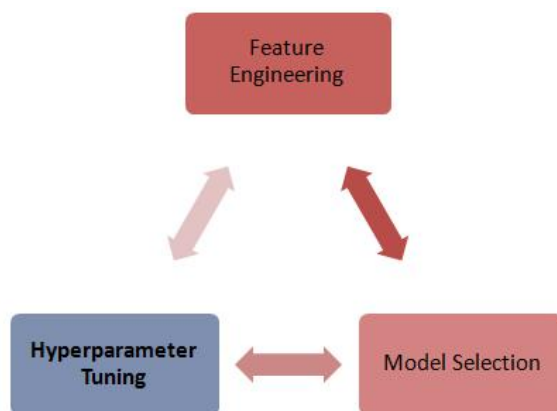


图 3-2 分类器构造关系

Figure 3-2 Classifier structure relationship

常见的超参数：C（本设计使用该超参数），Kernel，Degree，alpha 等。

网格搜索算法：遍历预定义的超参数并模型适合创建的训练集。这是对模型的特定参数值执行的相近搜索，该模型也称为评估器。通过 `GridSearchCV` 提供的网格搜索从通过 `param_grid` 参数确定的网格参数值中全面生成候选。

### 3.2.2 基于 L1，L2 的正则化

当构建机器学习模型时过拟合，欠拟合等问题很容易出现导致模型的效果和性能弱化，通常的处理手段方法为正则化，正则化是构建机器学习模型优化手段，在构建逻辑回归分类模型中通过使用两种正则化项来解决 L1,L2。在代价函数中添加惩罚项来控制模型的复杂度进而减少过拟合问题。L1 正则化（Lasso 回归）和 L2 正则化（岭回归）即为损失函数的两种惩罚项。L1 正则化可以通过参数稀疏化对模型进行特征选择提高模型的泛化能力，防止模型出现过拟合问题。本设计将会基于 Bow-ngram 的分词基础上构建逻辑回归模型，并通过网格搜索算法优化。在构建模型之前需要编辑一些 API 进行使用。

在构建深度学习模型时，同样可以使用正则化来优化模型。DNN 神经网络模型是由多个逻辑回归模型构成，多层感知机模型通常使用 `alpha` 对模型进行正则化避免过拟合问题。在食品评论文本情感倾向性模型中，本设计将构建正则化函数进行优化。

### 3.3 多层感知机模型

人工神经网络是基于人脑神经的基础上提出的模型，它从信息处理的角度出发将人脑神经元抽象化基础上建立模型，通过多个神经节点连接进而对数据之间的复杂关系模拟分析，将不同神经节点随机赋予权值（节点之间的影响力）其每个神经节点赋予一个特定函数，将隐藏层节点进行综合权重计算后输入到激活函数得到兴奋或抑制的结果。简称为神经网络模型。

设计神经网络需要考虑其节点层构造，通常神经网络节点的输入层和输出层的节点数是固定不变的，中间隐藏层可以自由指定神经结构。人工神经元模型可由下图所示：

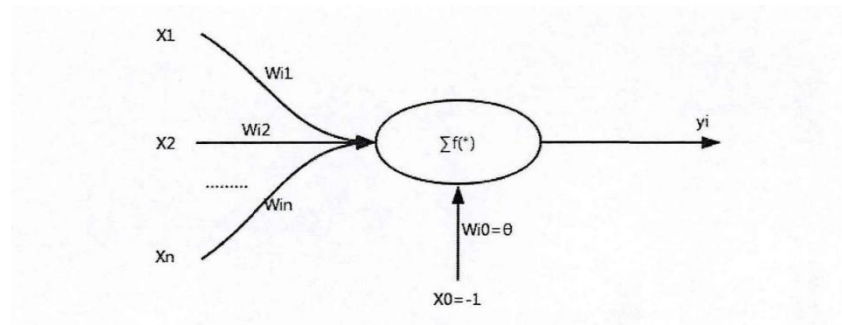


图 3-3 神经元结构

Figure 3-3 Neuron structure

人工神经元模型结构图核心是连线部分将各个神经节点连接起来，每个神经节点之间的连线对应一个权重，通过计算机训练得出最佳权值，使得整个神经网络模型预测效果达到最好。 $x_1, x_2, x_n$ 表示输入部分， $w_i$ 表示权重。有向箭头表示为信号由 $x_n$ 变成了 $x_n * w_i$ ，在神经元模型中，信号变化是随着权值变化而变化的。

**激活函数:**激活函数的作用是用来解决线性模型无法处理的问题。通过引入非线性的因素，提高模型对样本的拟合能力，使其能够逼近任意函数，使神经网络更有意义。通常，神经网络的输入层引入激活函数，使输入数据具有非线性，否则，没有激活函数，复杂的神经网络也只是输入线性组合。所有激活函数都使用非线性函数，如 sigmoid、tanh, Relu 等。

感知机是在 1957 年提出的是神经网络模型和支持向量机模型的基础，它是线性分类的二分类模型，输入数据通过与权重参数求和并加上偏置项输入到感知机中，然后激活函数激活得到输出值。激活函数一般为非线性函数。

MLP 为多层感知机模型是一种监督学习算法由多层神经元全连接构成，通过

由  $N$  个逻辑分类模型组成的全连接神经网络模型，通过逻辑分类模型的  $X * W + b$  作为运算基础，在数据集上训练来学习函数，通过使用激活函数将非结构化数据转化成线性矩阵的构造方法。将逻辑回归模型以分层全连接的方式展示，输入层为 output layer，输出分类结果的神经层为 input layer，而中间的隐藏层称之为 hidden layer，每个神经元结构以权重相乘可以通过增加隐藏层层数增加更多神经元进而增加模型的复杂度提高模型的性能。MLP 模型包含隐藏层，输入层和输出层，其至少包含一个隐藏层。多层感知机模型的层数以及隐藏层中每个神经节点都是超参数。常见的 MLP 模型如下图所示。

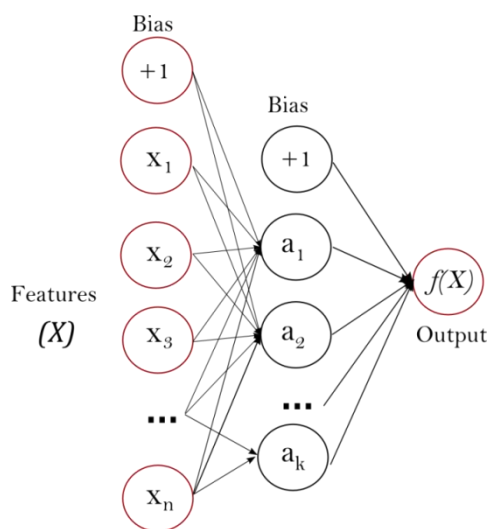


图 3-4：简单 MLP 模型

Figure 3-4: Simple MLP model

第一层为 output layer，第二层为 hidden layer，第三层为 input layer。由上图可知，输入和输出个数为  $n$  和 1，中间隐藏层包含  $k$  个神经节点。不包括输入层多层感知机的层数最低为两层。多层感知机最底层是输入层，中间是隐藏层，最后是输出层，隐藏层中神经节点与输入层神经节点完全连接，输出层神经元与隐藏层神经节点也完全连接称为全连接层。首先隐藏层与输入层是全连接的，假设输入层向量  $X$ ，隐藏层的输出就是  $f(W_1X + b_1)$ ， $W_1$  是权重（也叫连接系数）， $b_1$  是偏置，函数  $f$  可以是常用的 sigmoid 函数或 tanh 函数输出层为多类逻辑回归为  $f(W_1X + b_1)$ 。

$$\text{sigmoid}(a) = 1/(1 + e^{-a}) ; \quad \text{tanh}(a) = (e^a - e^{-a})/(e^a + e^{-a}) \quad (3-9)$$

不同的激活函数所表达的效果和泛化能力都不同，根据模型的不同选择不同的激活函数往往会产生不同的效果。

## 4 基于亚马逊食品评论情感倾向性分析模型构建

### 4.1 数据探索（EDA）

#### 4.1.1 数据集

本设计所使用的数据集来自 kaggle 竞赛，数据集包含来自亚马逊的精美食品评论，数据的时间跨度为 10 年，从 2002 年 10 月到 2012 年 10 月共 500000 条评论。数据集指标包括产品和用户信息，帮助性，评级以及文本评论。它还包括来自所有其他亚马逊类别的评论。

数据的内容主要包括两部分，Reviews.csv 和 database.splite.数据包含的信息为 1999 年 10 月至 2012 年 10 月的评论共 568454 条点评，共有 256059 位用户参与 74258 个产品的评论。其中有 260 位用户有超过 50 条评论。其中 Reviews.csv 所包含信息为用户 ID，产品 ID，用户各自类型（String），Helpfulnessnumber（评论的帮助性），HelpfulnessDeno（有多少位用户认为该评价有价值意义），分数，发布时间，评论的题目（String）以及评论内容（String）。

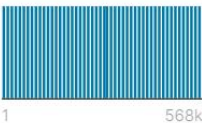

Id	ProductId	UserId	ProfileName	HelpfulnessNumber
Row Id	Unique identifier for the product	Unque identifier for the user	Profile name of the user	Number of users who found the review helpful
	<b>74258</b> unique values	<b>256059</b> unique values	<b>218418</b> unique values	
1	B001E4KFG0	A3SGXH7AUHU8GW	delmartian	1
2	B00813GRG4	A1D87F6ZCVE5NK	d11 pa	0
3	B000LQ0CH0	ABXLMWJIXXAIN	Natalia Corres "Natalia Corres"	1

图 4-1 亚马逊美食评论部分指标展示

Figure 4-1 Display of some indicators of Amazon food reviews

```
In [4]: #Printing dataset information
review.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 568454 entries, 0 to 568453
Data columns (total 10 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Id                     568454 non-null  int64
1   ProductId             568454 non-null  object
2   UserId                568454 non-null  object
3   ProfileName           568438 non-null  object
4   HelpfulnessNumerator  568454 non-null  int64
5   HelpfulnessDenominator 568454 non-null  int64
6   Score                 568454 non-null  int64
7   Time                  568454 non-null  int64
8   Summary               568427 non-null  object
9   Text                  568454 non-null  object
dtypes: int64(5), object(5)
memory usage: 43.4+ MB
```

图 4-2 各列指标数据的基本信息

Figure 4-2 Basic information of each column of index data

	Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenominator	Score	Time	Summary	Text	Helpfulness
0	1	B001E4KFG0	A3SGXH7AUHU8GW	delmartian	1	1	5	1303862400	Good Quality Dog Food	I have bought several of the Vitality canned d...	Helpful
1	2	B00813GRG4	A1D87F6ZCVE5NK	dll pa	0	0	1	1346976000	Not as Advertised	Product arrived labeled as Jumbo Salted Peanut...	No Indication
2	3	B000LQOCH0	ABXLMWJIXXAIN	Natalia Corres "Natalia Corres"	1	1	4	1219017600	"Delight" says it all	This is a confection that has been around a fe...	Helpful
3	4	B000UA0QIQ	A395BORC6FGVXV	Karl	3	3	2	1307923200	Cough Medicine	If you are looking for the secret ingredient I...	Helpful
4	5	B006K2ZZ7K	A1UQRSCLF8GW1T	Michael D. Bigham "M. Wassir"	0	0	5	1350777600	Great taffy	Great taffy at a great price. There was a wid...	No Indication

图 4-3 前五行数据展示

Figure 4-3 Data display of the first five rows

### 4.1.2 探索性数据分析

EDA 是一种数据分析方法，采用各种技术来对数据进行分析。本设计的主题为针对食品评论情感倾向性问题进行分析，评论情感倾向性所涉及文本类型等非结构化数据与数值类型数据，错综复杂。构建针对食品领域的评论情感分析，前期数据探索尤为重要。本设计通过对亚马逊食品评论数据集探索分析，为构建分词模型，分类模型架构提供理论依据和数据支撑，主要的工作是对数据进行清洗，



对其进行描述（描述统计量，图表），查看数据的分布，比较数据之间的关系，培养对数据的直觉，对数据进行总结等。在亚马逊食品评论数据集中所包含的数据不仅仅是评论文本也包含其他的结构化数据。根据本设计所使用亚马逊食品评论数据集的特点，数据探索工作将分为数据查看，统计数据集信息，数据集边界探索，指标间相关性探索，数据集统计以及可视化展示。

首先通过对数据集的查看让我们能够清楚的了解整个数据集的结构，以及包含的信息，通过调用函数可以了解数据集的指标由 9 部分组成分别为用户 id，产品 id，评论 id，食品名称，认为该评论有用的用户数，用户评分，发表评论时间，评论标题，评论文本以及帮助性。数据探索分析第一步从宏观的角度直观分析整个数据分布，如下图所示展示数据集中左右数据的直方图展示。

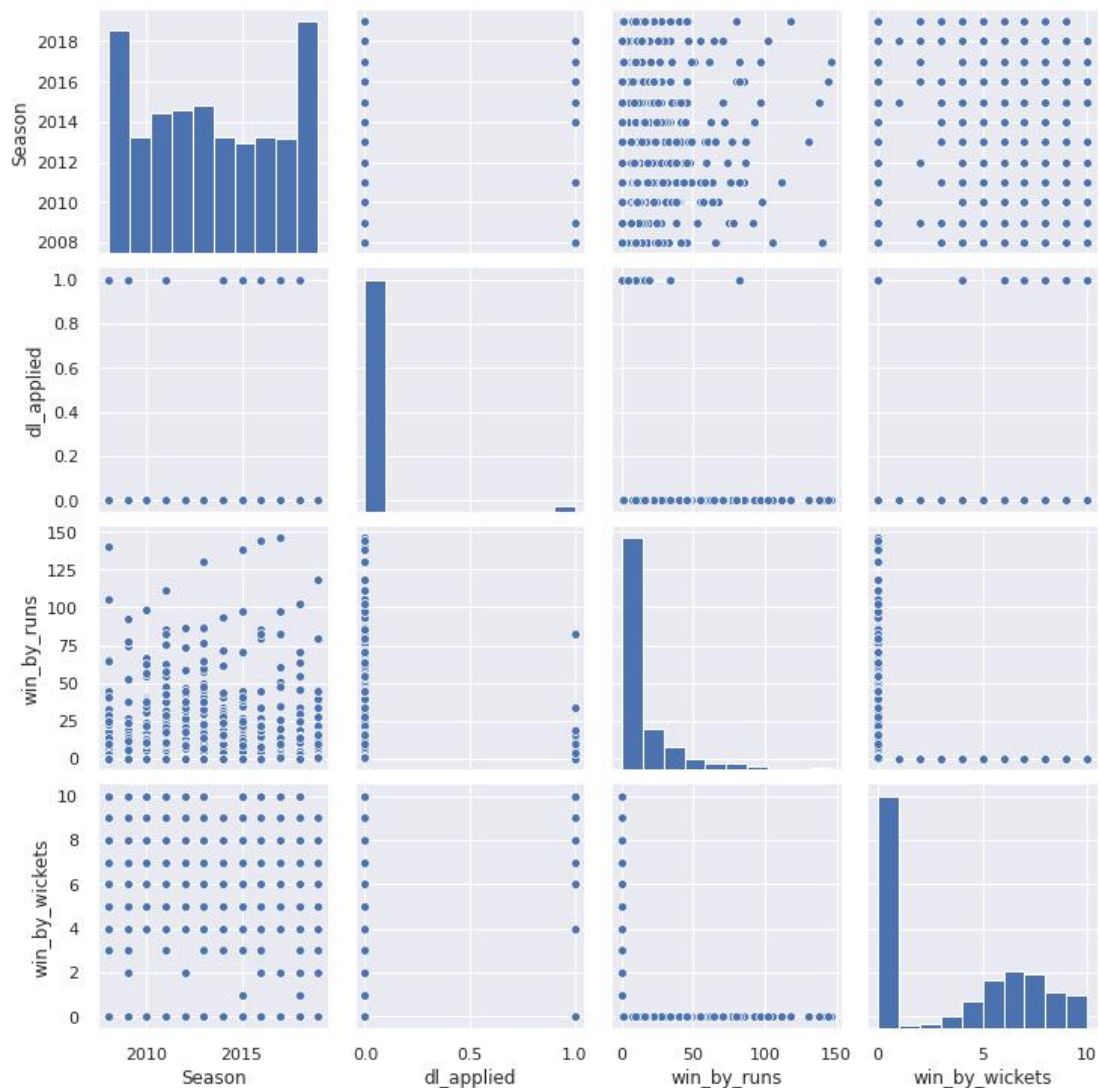


图 4-4 宏观直方图

Figure 4-4 Macro histogram



直方图是用一系列高度不等的纵向条纹或线段来表示数据的分布情况。横轴一般是数据类型，纵轴是统计特征。根据直方图以及数据类型和总数展示可以发现有一些缺失值情况，在食品名称以及评论标题所占缺失比为 0.002%，以及 0.005%。对整体影响不大因此将缺失值删掉即可。食品评论中有很多精彩的评论收到了其他用户的肯定和支持，这些评论为新用户提供了用帮助的建议，同时也为商家提供可参考的建议。其次要探索的部分为指标中 Helpfulness Numerator 和 Helpfulness Denominator 以及 score 的分布情况，以及从每年评论最多，最佳食品展示，在大多数情况下每年评论最佳的排名者情况。

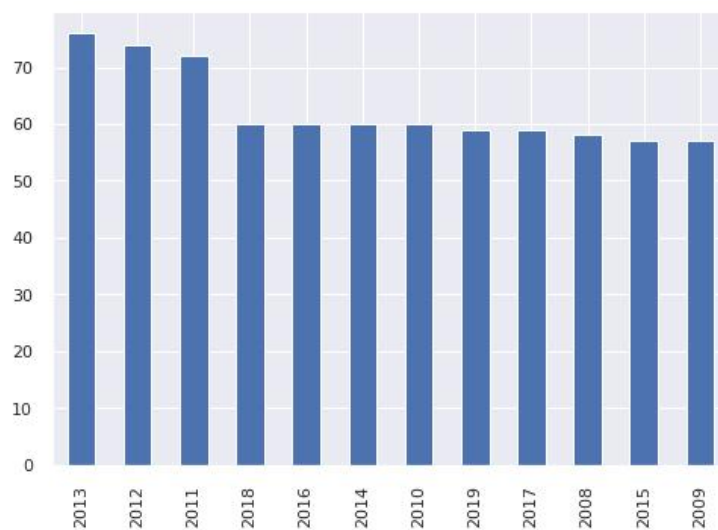


图 4-5 每年最佳评论数

Figure 4-5 Number of matches played in each year

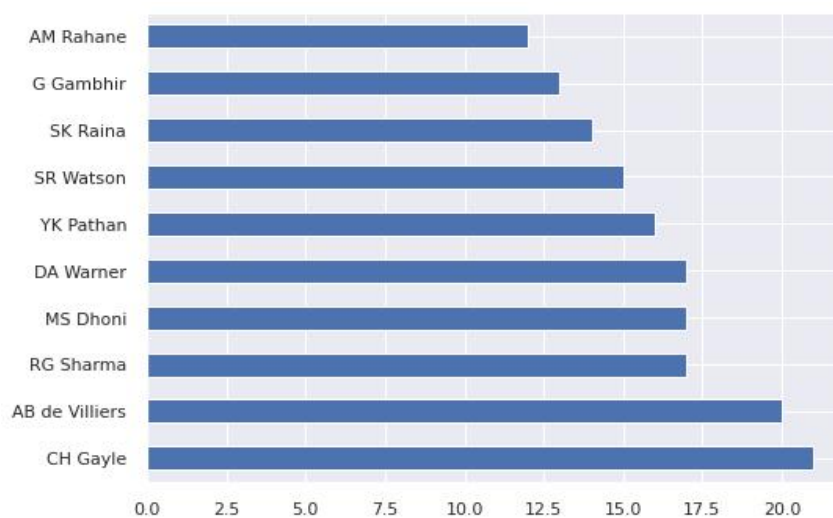


图 4-6 评论最佳前十名

Figure 4-6 Top 10 Player of the match winners over the time

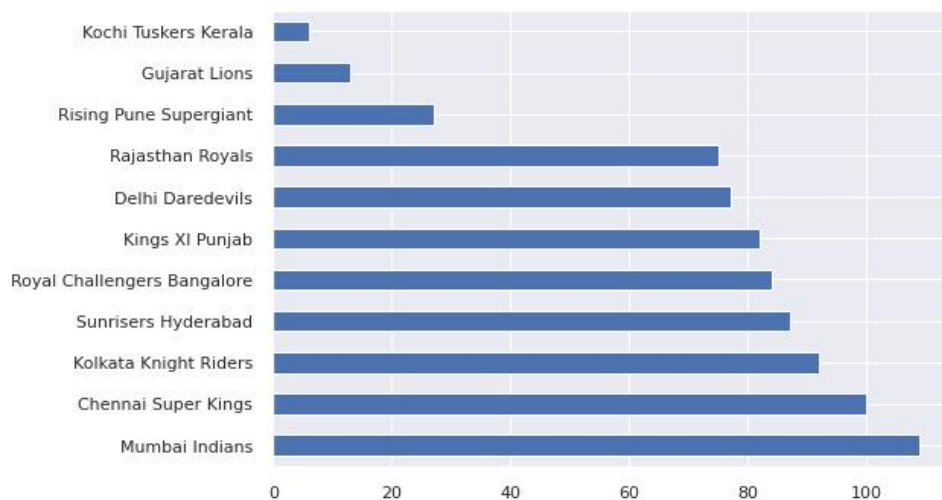


图 4-7 最大评论数食品

Figure 4-7 the most number of matches

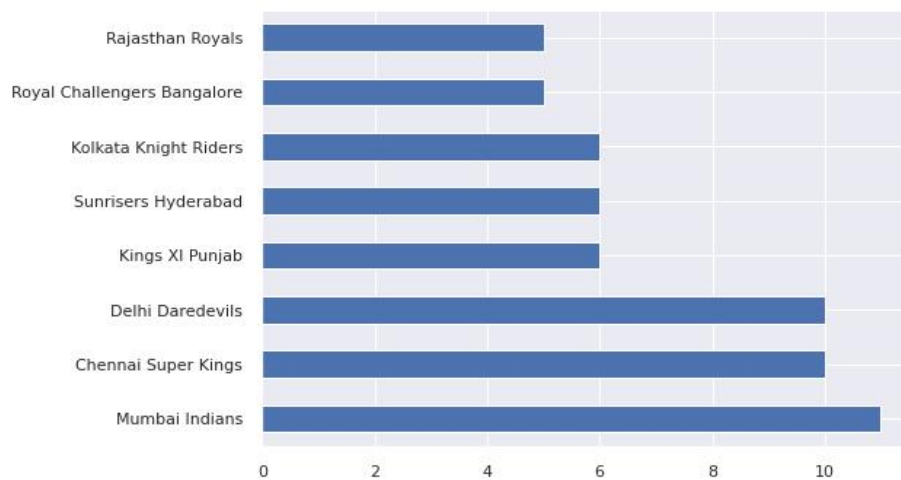


图 4-8 大多数评论最佳

Figure 4-8 Most matches won in particular season

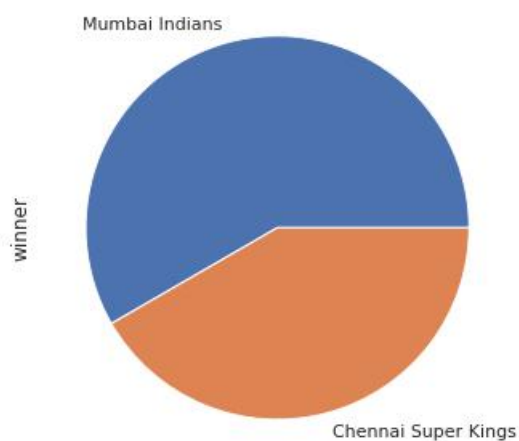


图 4-9 高分百分比

Figure 4-9 highest percentage of wins

	HelpfulnessNumerator	HelpfulnessDenominator	Score
count	393933.000000	393933.000000	393933.000000
mean	1.734399	2.204801	4.179399
std	6.872264	7.534772	1.311925
min	0.000000	0.000000	1.000000
25%	0.000000	0.000000	4.000000
50%	0.000000	1.000000	5.000000
75%	2.000000	2.000000	5.000000
max	866.000000	923.000000	5.000000

图 4-10 指标数据

Figure 4-10 Indicator data

从上述数据展示可知，在数据清洗后剩余的 393933 条评论中有价值的评论与评分存在一定的关系，这为我们进一步分析提供了一定的帮助。根据上述展示数据，在清洗过程不仅要清理缺失值也要处理重复值问题，将重复值数据全部剔除剩余数据的形状为(393933, 10)。将这三个指标分别展开分析，探究三者之间的潜在联系。其次探索部分主要为数据指标之间的关系，对数据集进行规划，从而为下一步特征工程做准备。通过评分分布可知评分分数为 5 分的所占比例远超其他评分，说明商品的总体趋势是处于积极的一面。评分 5 分的评论最多，为 250962 条，2 分的评论占比最少位 20802 条。

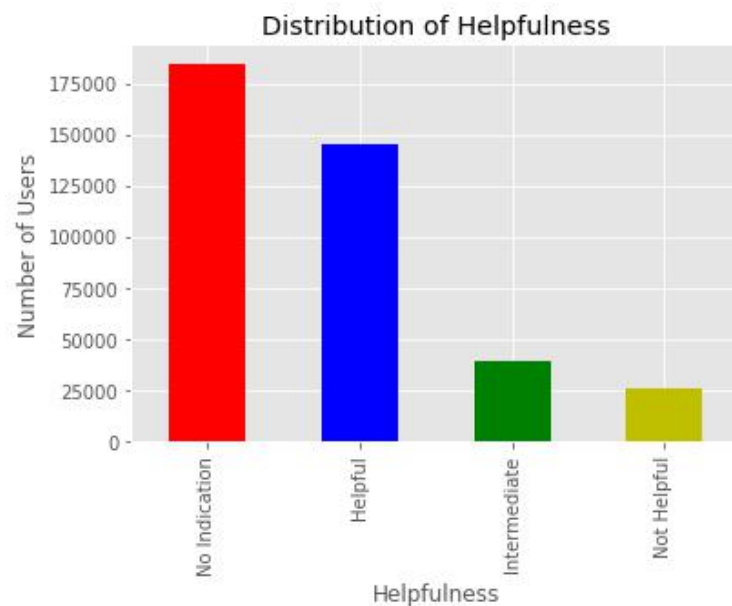


图 4-11 帮助性分布

Figure 4-11 Helping distribution

从上述数据展示可知，在数据清洗后剩余的 393933 条评论中有价值的评论与评分存在一定的关系，这为我们进一步分析提供了一定的帮助。下一步将这三个指标分别展开分析，探究三者之间的潜在联系。由评分分布可知评分分数为 5 分的所占比例远超其他评分，说明商品的总体趋势是处于积极的一面。

探索信息：评价 5 分的用户最多为 250962 人，打 2 分的用户最少为 20802 人。帮助性分布：计算帮助性，我们通过分数换算将帮助性分为四个等级 No Indication, Helpful, intermediate, Not Helpful。通过转换将清晰的筛选出评论的价值性，如下图所示展示转换结果：

下图展现帮助性分布情况，并计算各等级所占百分比：

Helpfulness	Helpful	Intermediate	No Indication	Not Helpful
Score				
1	9909	9994	9174	7229
2	5430	4551	7574	3247
3	8180	4766	13121	3702
4	19465	4516	28859	3254
5	101917	14935	125895	8215

图 4-12 帮助性在分数下分布情况

Figure 4-12 Distribution of helpability under score

探索信息：帮助性为 helpful 的用户数为 144901 占比 36.78%，帮助性为 not helpful 的用户为 25647 占比 6.51%。帮助性在各评分中的分布如下图所示：

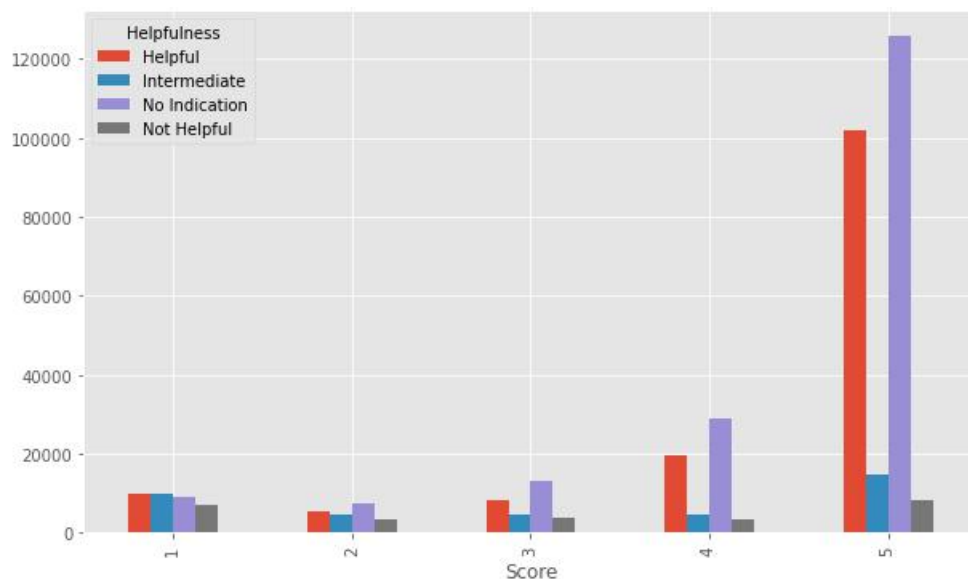


图 4-13 帮助性在分数下分布可视化

Figure 4-13 Visualization of the distribution of helpfulness under the score

随着时间的变化，客户的评论也可能存在相对应的变化，我们以月为单位分析食品评论随着时间的变化高评分所占比例逐渐增加，说明了商品越来越符合客户的需求。如下图所示：

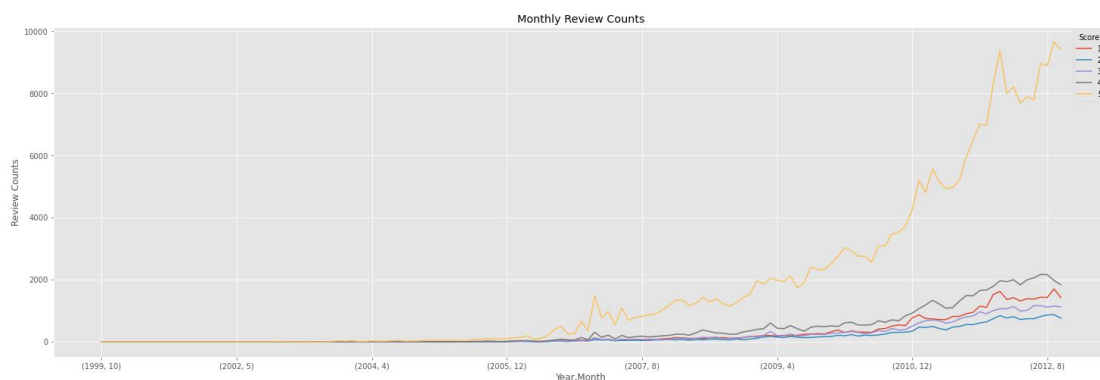


图 4-14 月单位评论数与分数关系

Figure 4-14 The relationship between the number of comments per month and the score

分析评论情感首先要分析文本的特性与其他属性之间的联系，评论文本的字数代表评论用户的用心程度，存在与评分之间的关系，下图所示展现评分等级与文本长度之间的关系，可以得知分布特征属于分数越靠近临界值评论文本字数相对应的增加，所以可以推断评论文本长度会因为评分的变化而变化。同理分析评

论文本长度与 rating 关系，以及摘要文本长度与 rating 关系，摘要文本长度与 rating 之间的关系，如毕业论文附录所示。分析文本特征可能与各指标存在的联系，下一步展示被评论商品的次数，如下图所示可以清晰的发现哪类商品的评论数较多，我们将分数编码分为 positive 和 negative 两类，可以分析乐观看法和消极看法的占比。通过 Seaborn 工具绘图箱型图来表示一组数据分散情况的统计图，其绘制过程首先找出一组数据的上下边缘，中位数以及两个四分位数。然后链接两个四分位数，画出箱体。如下图展示月为单位时间度量进行分数分配分析，评论的文本的长度与等级之间的变化关系经过分析最大评论文本长度为 3526，评论文本的平局长度为 81.5747，最小评论文本长度为 3。评论文本长度与 rating 之间的关系。摘要文本长度与 rating 之间的关系，最大摘要文本长度为 42，评论摘要文本长度为 4.1037，最小摘要文本长度为 1，以及摘要文本长度与 rating 之间的关系。

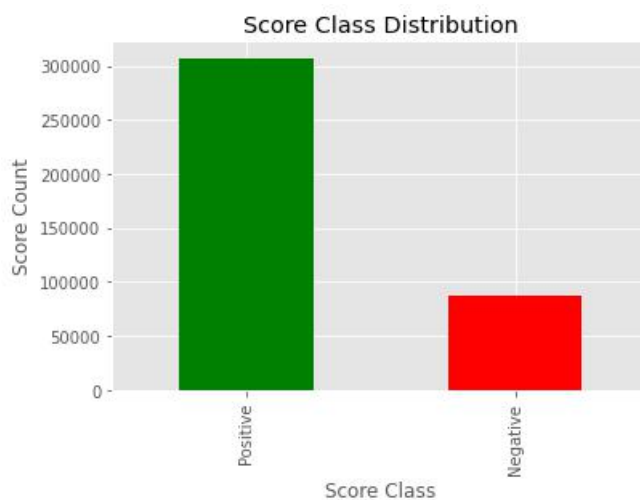


图 4-15 分类后评分分布情况

Figure 4-15 Score distribution after classification

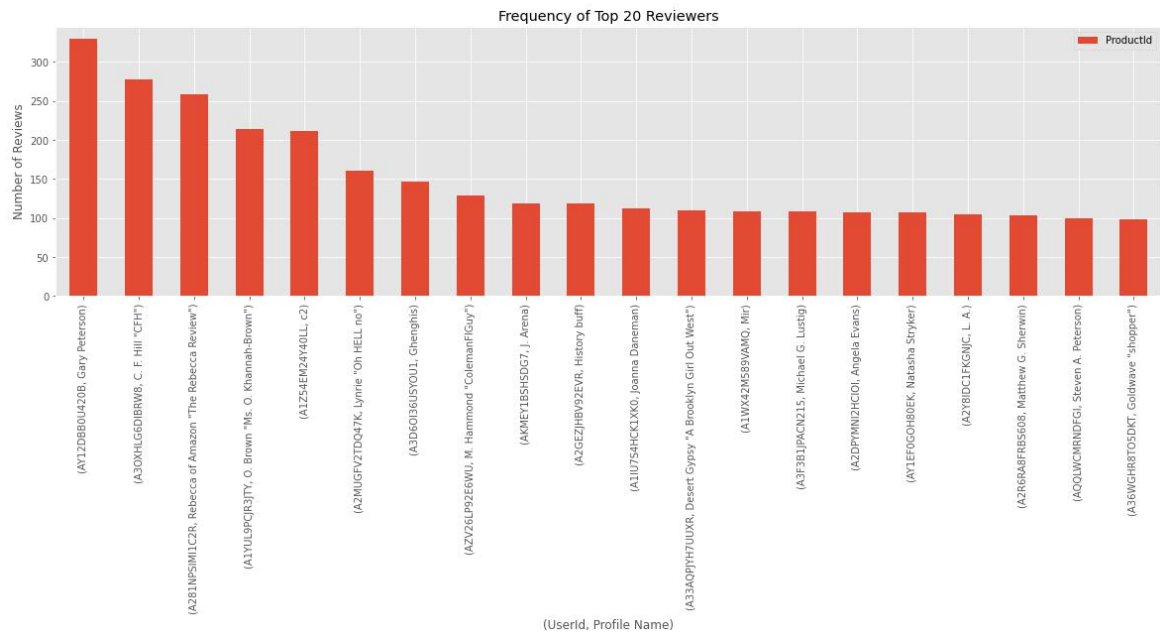


图 4-16 评论前二十商品

Figure 4-16 Top 20 products reviewed

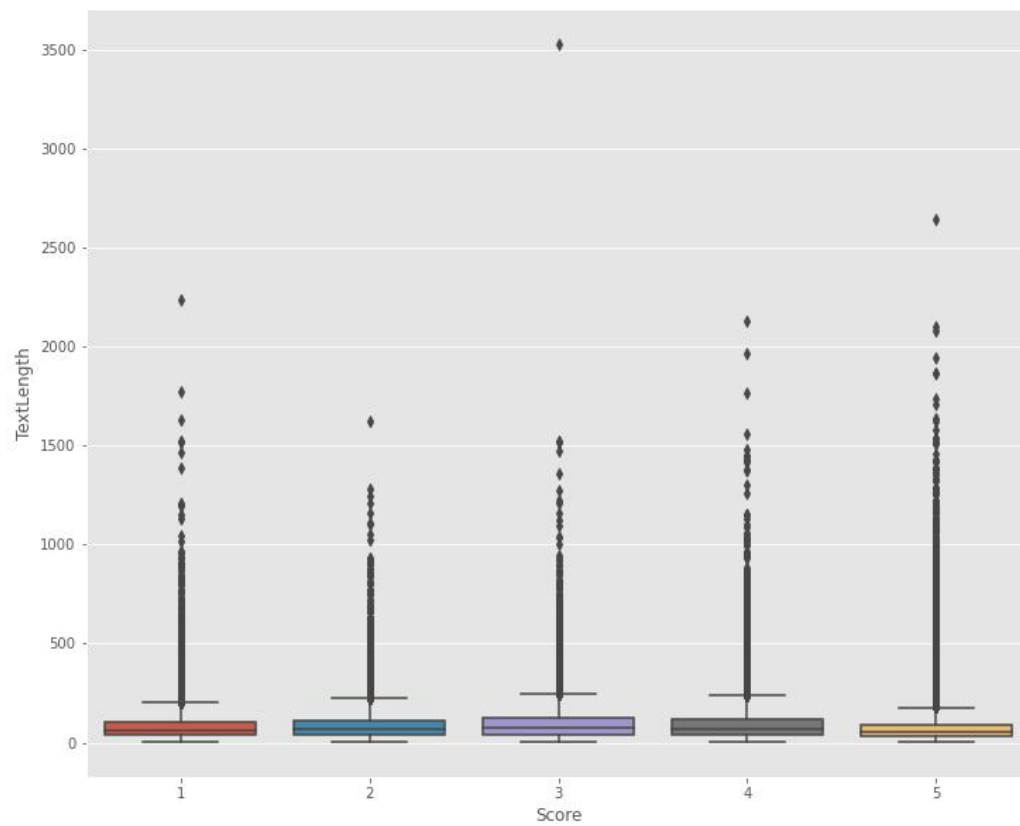


图 4-17 文本长度与分数箱型图

Figure 4-17 Box plot of text length and score

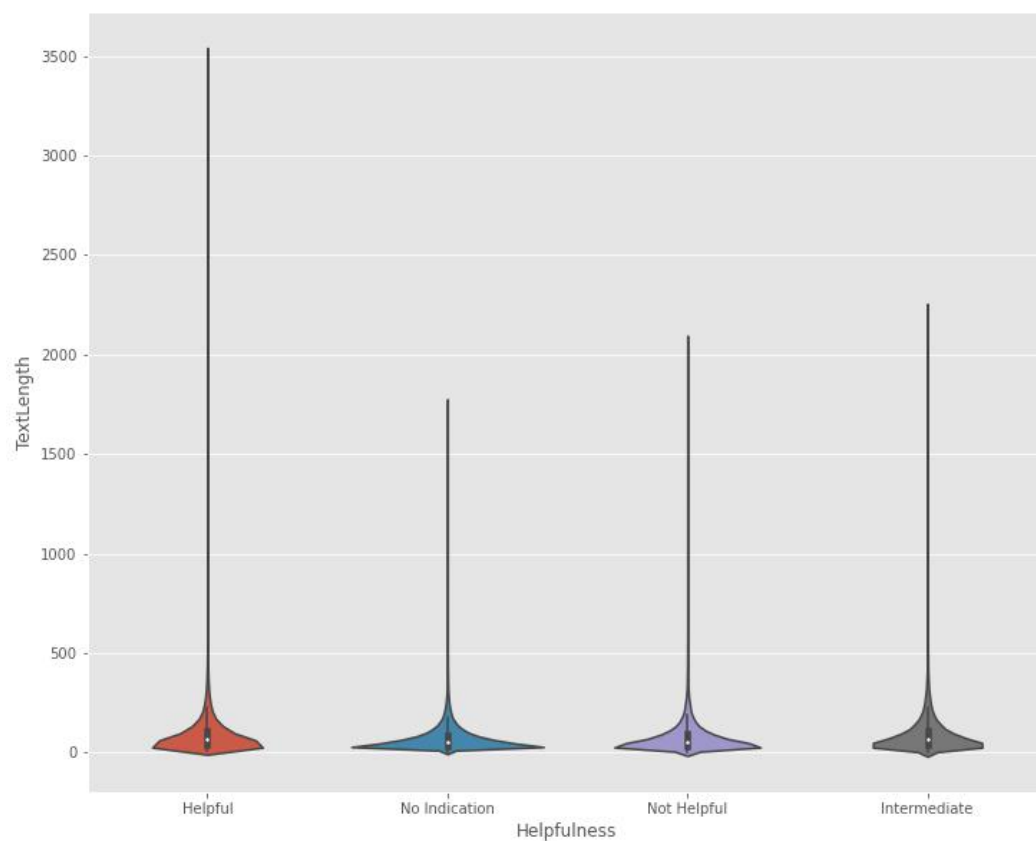


图 4-18 文本长度与帮助性小提琴图

Figure 4-18 Text length and helpful Violin illustration

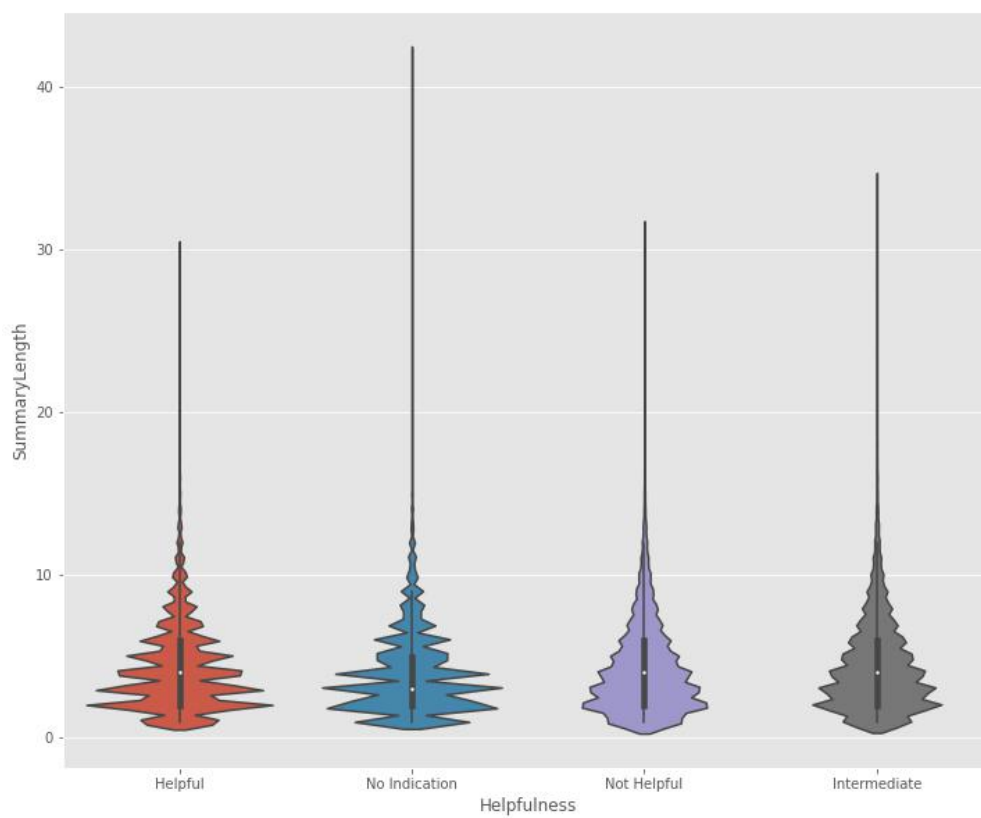




图 4-19 主题文本长度与帮助性小提琴图

Figure 4-19 The length of the topic text and the helpful Violin illustration

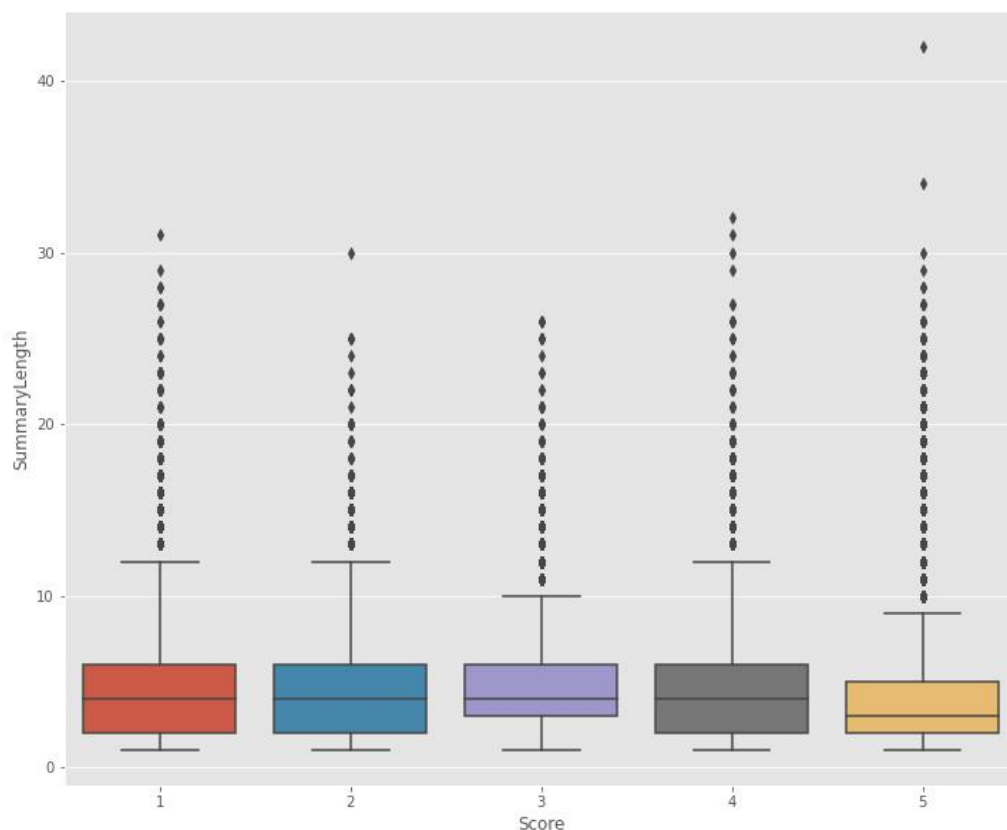


图 4-20 主题文本长度与分数箱型图

Figure 4-20 Box plot of topic text length and score

基于正负类的数据分割：通过 `splitPosNeg` 函数将清洗后的评论数据，将积极评论与消极评论数据进行汇总，并将积极评论与消极评论的以文本形式输出展示。通过数据探索可知总整体观察可了解。通过探索性数据分析了解评级文本与分数，帮助性等指标之间的关系确定了模型的参考评判标准，确定帮助性的四个等级 `helpful`, `intermediate`, `not help`, 以及 `no indication`。将数值结构化便于数据清洗部分的处理。分析评分与帮助性的关系相对应的评论文本越长帮助性所占比例相对越高，说明评论文本的长度直接影响帮助性。关于评论文本长度与分数关系说明评分与评论文本长度的关系不明显，评论文本的长度可能代表客户对于商品的用心评价，存在的价值相对应高，但存在价值的高与所评价分数并无直接关系。通过分析指标间的关系，可以科学的将分数指标分为 `positive` 和 `negative` 两类，评分大于 3 即为积极情感的评论，反之即为消极的情感评论。分类之后所占比积极评论所占 307056 条，消极评论所占 86877 条。

### 4.1.3 数据清洗

数据预处理部分是整个数据挖掘项目的重点，在自然语言数据挖掘领域中，通过计算机相关技术和方法对文本数据进行加工处理。本设计研究食品评论情感分析领域依照数据探索所提供的关键信息，了解评价与时间，分数等指标影响关系根据数据探索所分析的结论为理论依据进行数据清洗。模型的性能以及特征构造的良好依赖于数据处理过程所清洗整理的数据的好坏，在同一种模型，优化方法下，不同的处理结果往往有较大差距。自然语言处理数据挖掘的主要流程主要分为六个阶段原始文本的整理，分词模型，文本清洗，标准化以及特征提取和模型构建。如下图所示：

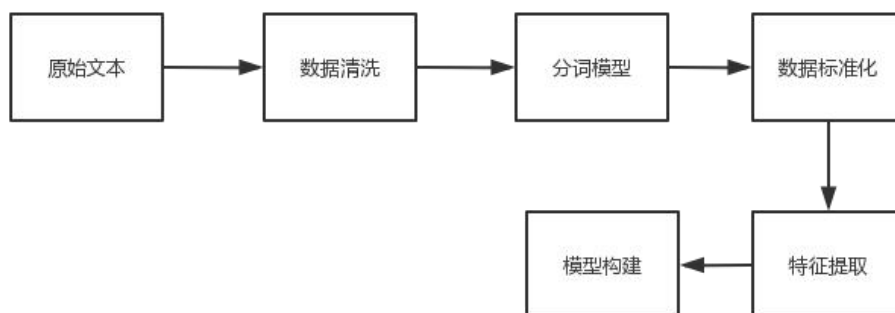


图 4-21 自然语言数据挖掘流程

Figure 4-21 Natural language data mining process

文本数据预处理主要的部分为原始文本的获取，数据清洗部分，文本数据的获取方式主要为爬虫，网络公开数据集，以及竞赛数据等。数据清洗部分的内容主要是清洗文本中非文本部分如 html 标签，数字，停用词等等。分词部分主要通过词袋模型进行处理，以及通过 python 手段去掉停用词。对于文本预处理，词干提取和词形还原为重点，词的变化形式有很多，一般词库只对正确的词进行分析处理。通过 nltk 库可以将词干提取，还原关键词。大小写转换则可以减少单词数量。

在自然语言处理中处理文本数据与机器学习中数据清洗的内容步骤不同，常见的文本内容包含多种多样，如 html，数字，url 等。处理文本类型数据，目的为清洗干扰信息，停用词，数字，标点符号等。只需保留文本内容主要的文字信

息。文本数据挖掘直接影响到模型构建的效果。将文本信息的干扰信息清洗,尽可能的减少无用文本对算法模型构建的影响。

**数据预处理：**对评论文本的预处理，主要分为七个方面。评论文本作为 x 轴为任意长度的 String 类型，将其全部转换为小写字母，去除所有 html tags，删除所有标点符号或有限的数字字符集例如 or, and, 删除所有数字，删除文本中的 url，删除所有具有三个连续重复字符的单词，删除像 zzzzzzzzzzzzzzzzzzzzzz', 'testtting', 'grrrrrreeeeetttt' etc. 保留像'looks', 'goods', 'soon', ,删除停用词 Stopwords

数据预处理考虑了整个评论中所有的单词，我们将构建的预处理函数和指令分别对正面评论和负面评论进行预处理将处理好的正面评论和负面评论输出成两个列表 `pos_data` 和 `neg_data`。最后将预处理的正面评论和负面评论的列表进行组合，将评价分数进行组合。标记数据并创建标记列表。从整个评论中获取特殊词的列表，从整个评论中保存总计特殊词输出结果。

从整个评论中加载特殊词，展示特殊单词的长度分布，单词数量与单词长度的分布图如下图所示：

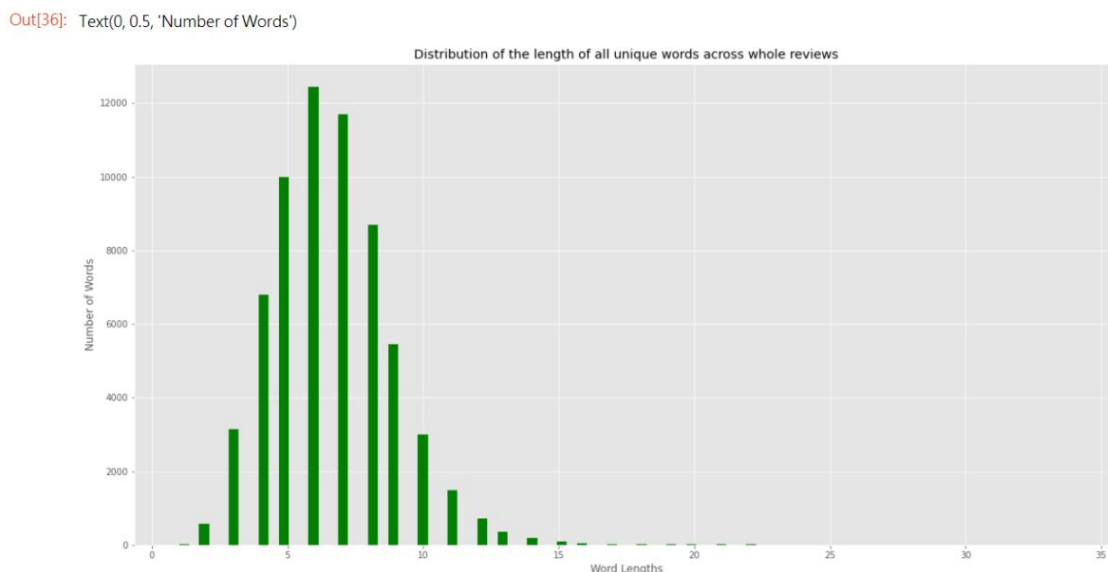


图 4-21 关键词分布

Figure 4-21 Keyword distribution

由上图可以得知，评论中大多数词的长度在 3-10 之间。长度超过 15 的单词与其他单词相比较少。所以，当处理特殊词时，需要将其从评论中删除，只保留长度大于 2 且小于 16 的特殊词。

在整个评论中仅考虑长度大于 2 且小于 16 的单词进行数据预处理，重复上

述操作。最后将最终数据和标签存储。使用分层策略将数据集划分为训练集和测试集，加载最终数据和最终标签，将数据集以 8: 2 比例分割为训练集和测试集。

首先需要分析评论是正面的还是负面的，通过 jupyter notebook 打开 database.splite 文件筛选评论评分不为 3 的所有评论，效果如下图所示：

Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenominator	Score	Time	Summary	Text	Helpfulness	
0	1	B001E4KFG0	A3SGXH7AUHU8GW	delmartian	1	1	5	1303862400	Good Quality Dog Food	I have bought several of the Vitality canned d...	Helpful
1	2	B00813GRG4	A1D87F6ZCVE5NK	dll pa	0	0	1	1346976000	Not as Advertised	Product arrived labeled as Jumbo Salted Peanut...	No Indication
2	3	B000LQOCH0	ABXLMWJIXXAIN	Natalia Corres "Natalia Corres"	1	1	4	1219017600	"Delight" says it all	This is a confection that has been around a fe...	Helpful
3	4	B000UA0QIQ	A395BORC6FGVXV	Karl	3	3	2	1307923200	Cough Medicine	If you are looking for the best...	Helpful

图 4-22 清洗后数据展示

Figure 4-22 Data display after cleaning

将小于评论打分小于三分的定义为消极（negative），大于 3 分为乐观（positive），并将分数替换为两个 string 类型的单词。这样就将数据分为二分类问题，消极或乐观（替换为文字是为了更加清晰认识，以后便于计算，可能会将代码调整为 0，1）去除停用词 HTML tags，special character（分词模型与模型构建密切相关，具体信息如模型构建展示）。

## 4.2 文本情感倾向性分类模型构建

### 4.2.1 词袋模型构建

本设计针对文本类型数据问题从定性的角度进行分析，通过数据探索和数据清洗部分将亚马逊食品评论文本的情感进行分类定义，并将文本类型非结构化数据预处理，在食品评论领域中进而构建有效的词组集合为构建词向量以及概率模型做准备。词袋模型为文本数据挖掘过程中特征工程部分中的特征提取，本设计使用两种分词模型组合 N-gram 算法进行分词。构建词袋模型的一元，二元分类，以及 TF-IDF 的一元二元分类，对于构建文本特征提取具有显著效果。

使用分层策略将数据集划分为训练集和测试集后构建训练词向量词典，总单词数为 57467，使用 GridSearchCV 库函数，便利预定义的超参数，使估计器模

型适合训练集，

BOW-Unigram 特征有'add','addendum','addict','addition','address','adept'...。该 Vectorizer 的训练集为 (315146, 12386) 矩阵，测试集为 (78787, 12386) 矩阵。同理训练 Bag of Words - Bigrams, TF-IDF - Unigram, 以及 TF-IDF - Bigram。

训练结果构建的 BOW-Bigram 向量特征矩阵训练集构成的特征矩阵为 (315146, 12386)，测试集构成的向量矩阵为(78787, 12386)。BOW-Bigrams 构成的特征向量特征值为'accord instruct', 'accord label', 'accord packag', 'accord tast'...构成训练集向量特征矩阵为 (315146, 50000)，构成的测试集向量特征矩阵为(78787, 50000)。

TF-IDF-Unigram 所构成的词典和词频数矩阵，特征值为'add', 'addendum', 'addict', 'addit', 'additon'...构成训练集特征向量矩阵为 (315146, 12386)，构成的测试集特征向量矩阵为 (78787, 12386)。

TF-IDF-Bigram 所构成的词典和词频数矩阵，特征值为'accord instruct', 'accord label', 'accord packag', 'accord tast'...构成训练集特征向量矩阵为(315146, 12386)，测试集构成的向量矩阵为(78787, 12386)。

#### 4.2.2 文本分类的机器学习算法

构建文本分类的机器学习算法，为验证基于词袋-N-gram 的 Lasso 逻辑回归分类模型，基于词袋-N-gram 岭回归的逻辑回归分类模型，基于 TF-IDF-Unigram 的 Lasso 逻辑回归分类模型，基于词袋-N-gram 岭回归的逻辑回归分类模型等四类模型在以亚马逊食品评论训练的效果。构建四类模型的程序均由 PYTHON 实现，描述下列四类的模型实现。

构建逻辑回归模型主要由三部分组成 `logisticRegression()` 函数，`GridSearchCV()`，以及评估模型性能函数。

#### 4.2.3 文本分类的深度学习算法

构建文本分类的深度学习算法模型主要由两部分组成，`RandomizedSearchCV` 以及 `MLPClassifier()`。构建多层感知机文本分类模型，构造模型需考虑参数配置。学习率，每组网络训练样本数以及 `beta` 参数值，隐藏层神经元的个数以及激活函数的选择。本设计通过构造参数空间函数，设置隐藏层数为(1024), (50,), (50,100, 50), (48,), (48, 48, 48), (96,), (144,), (192,), (96, 144, 192), (240,), (144, 192,

240)。学习率设定为 0.0001, 0.001, 0.05, 0.1, 1。使用 Adam 用于替代随机梯度下降的优化算法，学习率使用'constant','adaptive'两种学习方式。激活函数分别使用'tanh', 'logistic', 'relu'三种函数。MLPClassifier()使用参数设置为 max\_iter=10000, random\_state=42。

#### 4.2.4 模型优化

针对文本分类的机器学习模型优化方法主要分为八个步骤，标准化数据矩阵，绘制 ROC 曲线，绘制召回曲线，计算指标性能，绘制网格搜索结果，绘制，GridSearchCV 结果的 Heatmap，绘制误差与 C 值，应用 GridSearchCV 函数。

基于 MLP 的优化方法通过调整超参数进行优化，网格搜索算法通过调整超参数将所有超参数值下的模型训练效果进行比对，选出最优超参数时模型达到最优化。如下图所示构建搜索网格的 API。

该函数的解释为：一个具有线性内核并且 C 在[1,10,100,1000]中取值，通过此估计器 API，当数据集出现拟合时，参数值的所有可能的组合都会被评估，从而计算出最佳的组合。使用网格搜索进行超参数调整，主要的步骤为导入 sklearn, numpy 库，加载数据集，构建逻辑回归模型，创建超参数搜索空间，创建网格搜索，进行网格搜索，查看最佳模型的超参数值，最后使用最佳模型进行分类。

## 5 模型实验效果预测及验证分析

本章节为检验设计中提出的基于机器学习和深度学习文本情感分类方法的有效性，通过将两种分词模型以及两种正则化的逻辑回归模型进行比对，以及 L2 正则化的多层感知机分类模型效果，不同的模型超参数和参数进行对比实验，混入随机搜索和网格搜索优化算法，以训练一个最优的模型，得到最优的文本分类效果；不同的特征分类方法的对比试验，证明逻辑回归分类器和多层感知机分类器对于文本特征分类的有效性；不同文本情感分类模型对比证明两种模型的有效性。

### 5.1 预测效果

将超参数 C 赋值集合设定为[1000,100,10,1,0.1,0.01]，训练模型并拟合 10 次。基于 BOW Unigram 的 L1 正则化逻辑回归模型训练 C 的最佳超参数为 100，最佳评估得分为 0.91，基于词袋-N-gram 岭回归的逻辑回归分类模型最佳超参数为 0.1，最佳评估得分为 0.93。基于 TF-IDF-Unigram 的 Lasso 逻辑回归分类模型最佳超参数为 10，最佳评估得分为 0.99。词袋-N-gram 岭回归的逻辑回归分类模型最佳超参数为 10，最佳评估结果为 0.99。

正则化：多层感知机模型所使用的 sklearn 库通过参数  $\alpha$  作为正则化（L2 正则化）系数， $\alpha$  为 L2 正则化的参数，通过调整权重的方法避免过拟合，进而导致曲率过小产生更复杂的决策边界。

本设计将数据集进行处理展示  $\alpha$  值变化后函数变化状态。如下图所示。

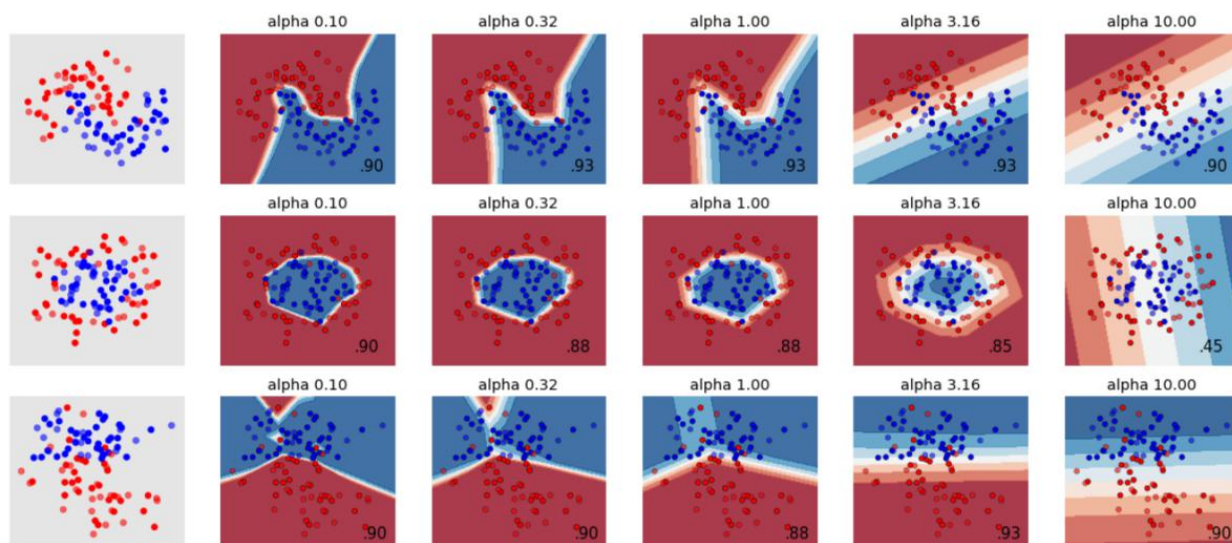


图 5-1 正则化分类边界展示图



Figure 5-1 Regularized classification boundary display diagram

由上图可知，alpha 值为 10 时模型状态稳定性最好。即将 alpha 值设为 10。接下来构造 MLP

因为神经网络所需算力庞大，无法通过个人电脑进行计算处理，通过 kaggle 提供的在线运算平台计算结果为 85.86%，绘制混淆矩阵如下图所示：

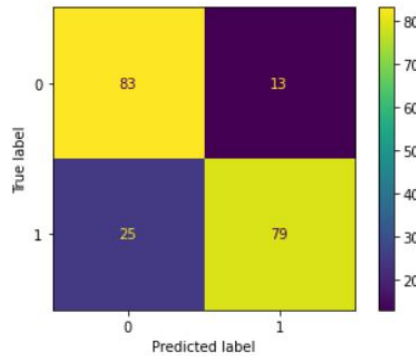


图 5-2 MLP 混淆矩阵

Figure 5-2 MLP confusion matrix

## 5.2 模型验证——交叉验证

构建逻辑回归分类器和基于随机搜索算法的多层感知机分类模型之后需要验证在处理食品评论情感倾向性问题的模型效果，为了保证模型训练结果的可靠性，分类器的实验评价对于分析模型泛化能力，需要通过混淆矩阵，准确率，精确率，召回率这些性能度量来对比模型的好坏。多层感知机模型的测试评价使用准确率和召回率作为评判依据。

准确率:预测正确的结果占总样本的百分比

$$\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN}) \quad (5-1)$$

精确率: 在所有被预测为正的样本中实际正的样本的概率.

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP}) \quad (5-2)$$

召回率: 实际为正的样本中被预测为正样本的概率

$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN}) \quad (5-3)$$

F1 分数: 精确率与召回率的关系，遍历[0, 1]之间所有的阈值，每个阈值对应的查准率和查全率，从而得到曲线。F1 分数为查准率和查全率同时达到最高，取一个平衡。



$$F1 = (2 * Precision * Recall) / (Precision + Recall) \quad (5-4)$$

$$\text{灵敏度} = TP / (TP + FN) \quad \text{特异度} = TN / (FP + TN) \quad (5-5)$$

ROC：基于混淆矩阵得出的用于评价模型的预测能力

AUC：曲线下面积对所有可能的分类阈值的效果进行综合衡量，计算 ROC 曲线上的点。基于排序的高效算法计算曲线下面积，是看作模型将某个随机正类别样本排列在某个随机负类别样本之上的概率。

观察其 ROC 曲线下面积与 C 值；其混淆矩阵如下图所示；ROC 曲线分布如下图所示：预测召回曲线如下图所示：

(1) 基于 BOW-Unigram 的 L1 正则化逻辑回归模型

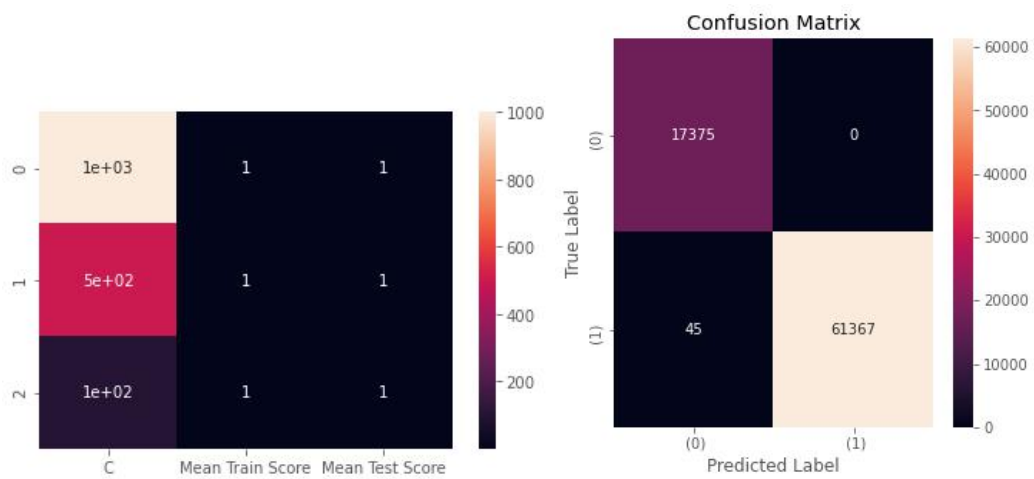


图 5-4 平均分数

Figure 5-4 mean score

图 5-5 混淆矩阵

Figure 5-5 Confusion matrix

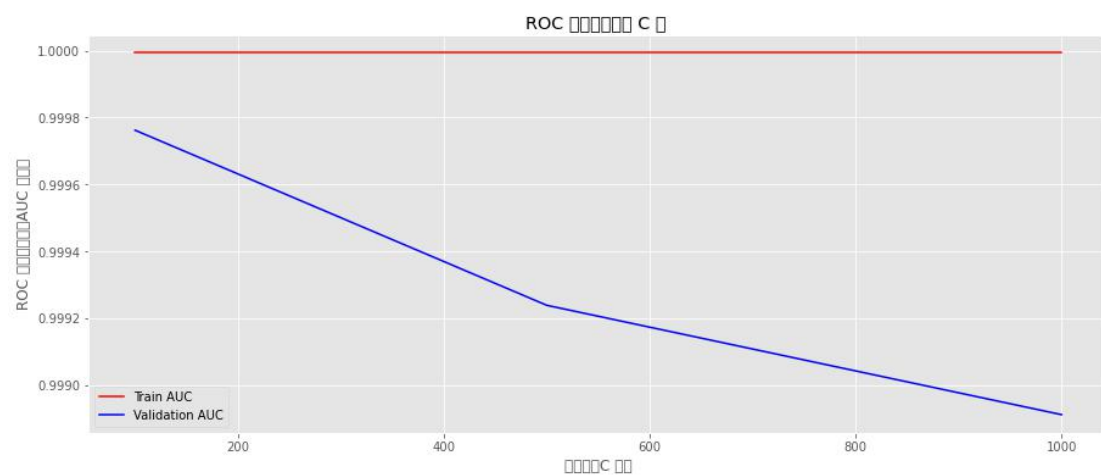


图 5-6 ROC 与 C 分布关系

Figure 5-4 Republic of China and C distribution

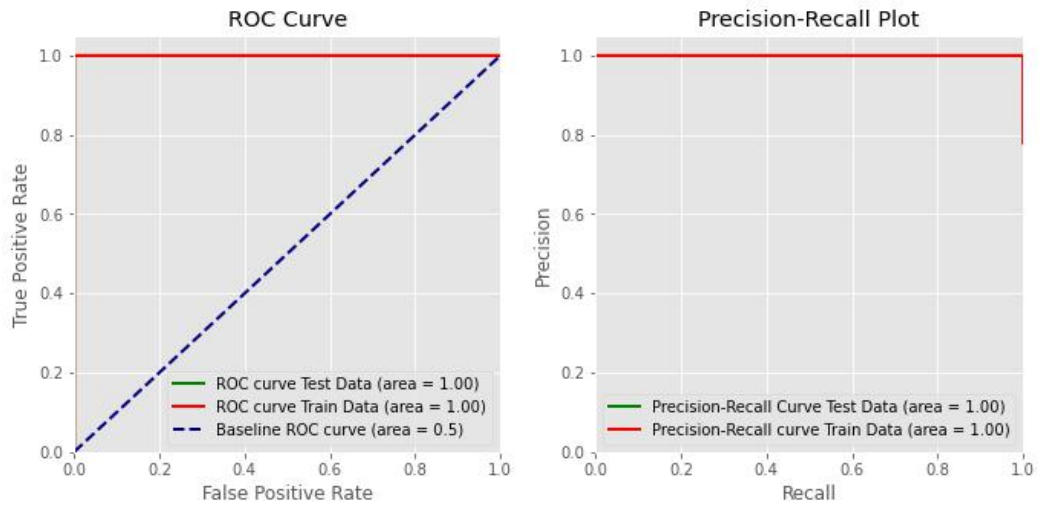


图 5-7 ROC 曲线

Figure 5-7 ROC curve

图 5-8 召回曲线

Figure 5-8 Recall curve

## (2) 基于 BOW-Unigram 的 L1 正则化逻辑回归模型

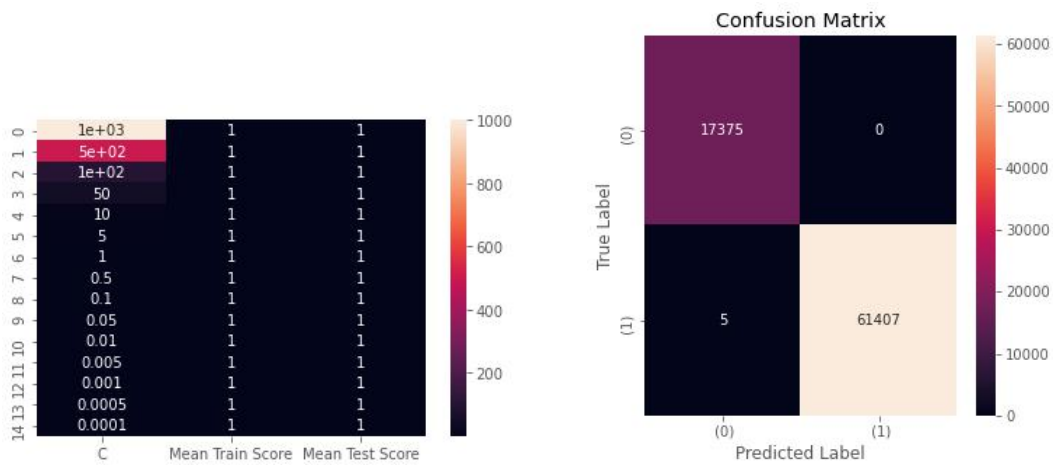


图 5-8 平均分数

Figure 5-8 mean score

图 5-9 混淆矩阵

Figure 5-9 Confusion matrix

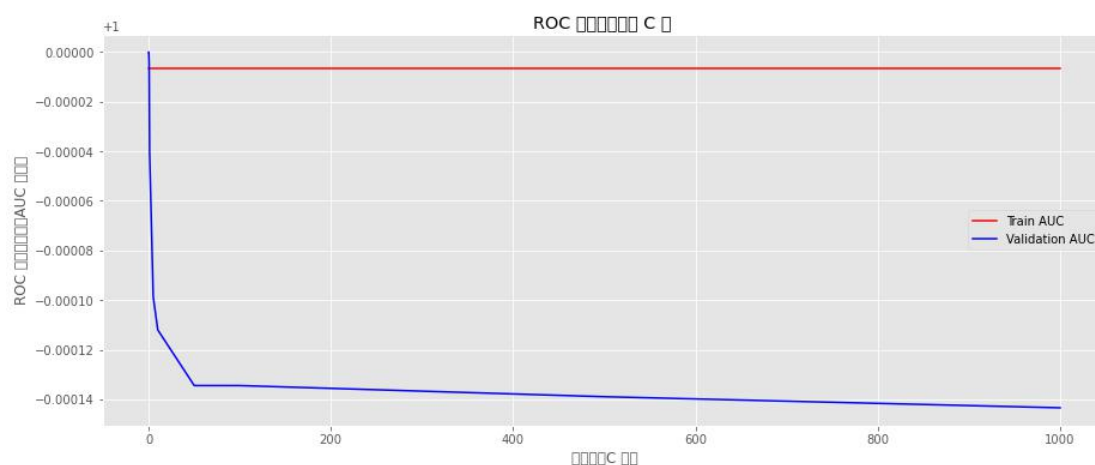


图 5-10 ROC 与 C 分布关系

Figure 5-10 Republic of China and C distribution

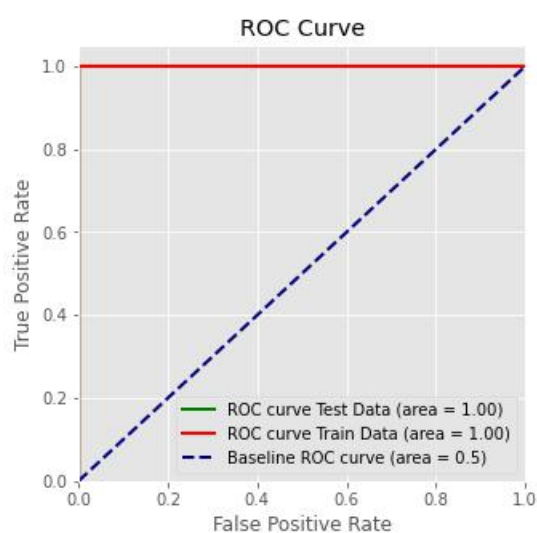


图 5-11 ROC 曲线

Figure 5-11 ROC curve

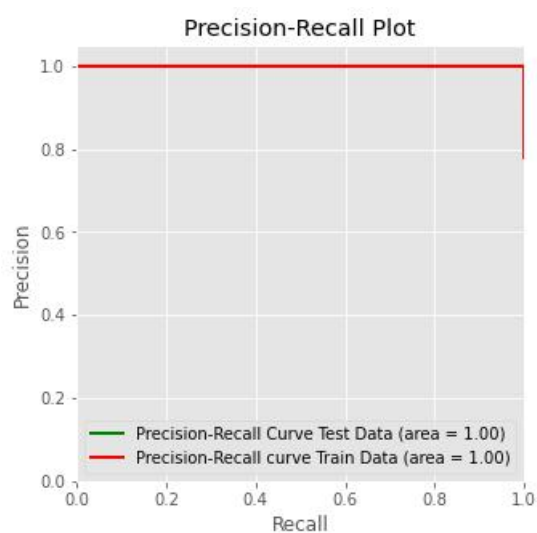


图 5-12 召回曲线

Figure 5-12 Recall curve

### (3) Logistic Regression with L2 Regularization on BOW Unigram

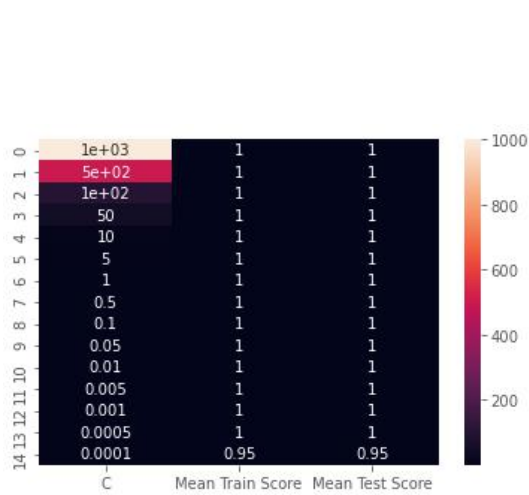


图 5-13 平均分数

Figure 5-13 mean score

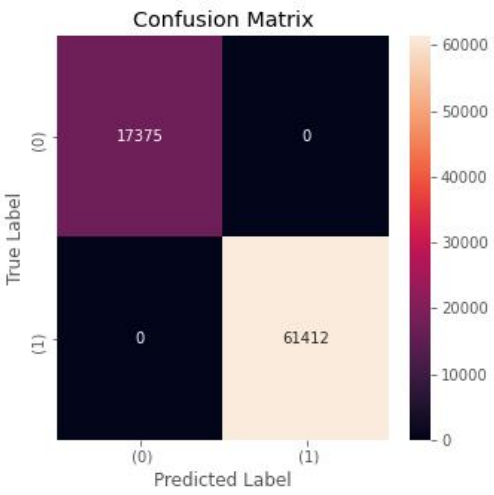


图 5-14 混淆矩阵

Figure 5-14 Confusion matrix

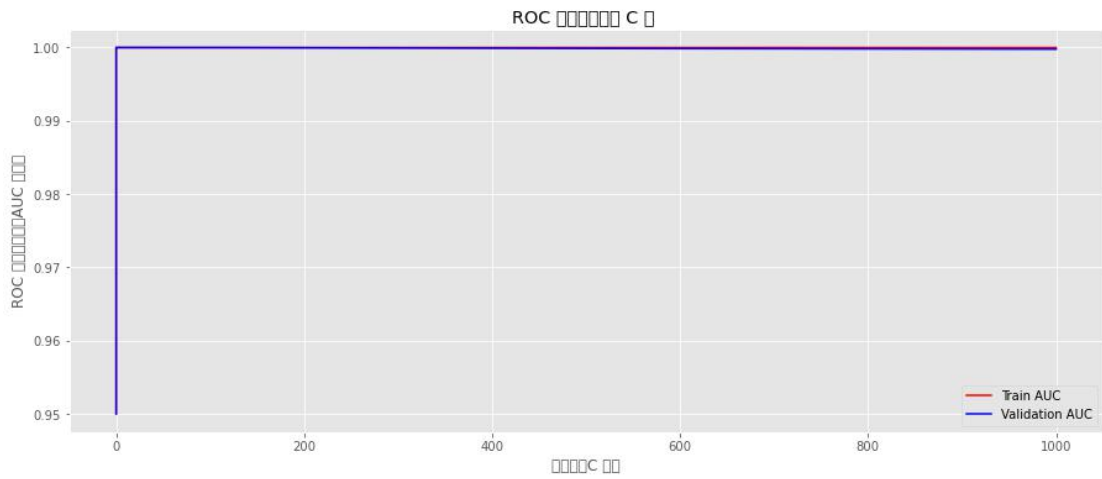


图 5-15 ROC 与 C 分布关系

Figure 5-15 Republic of China and C distribution

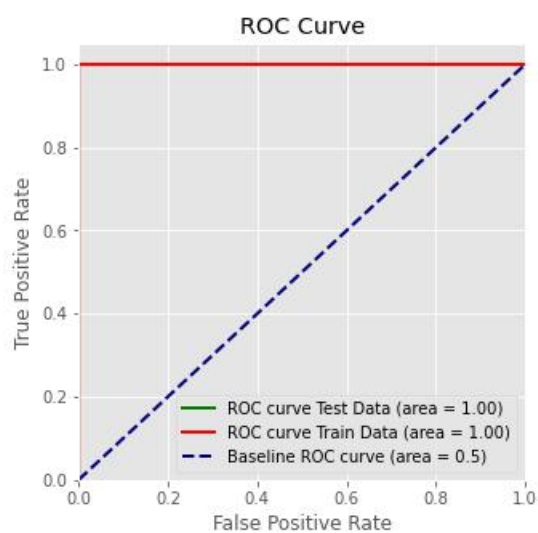


图 5-16 ROC 曲线

Figure 5-16 ROC curve

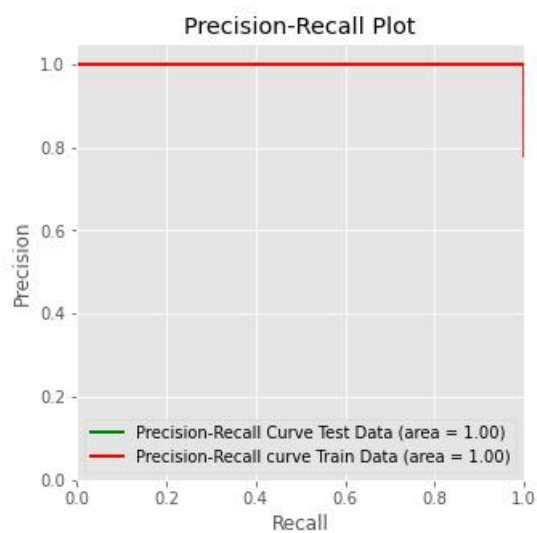


图 5-17 召回曲线

Figure 5-17 Recall curve

#### (4) Logistic Regression with L1 Regularization on BOW Bigram

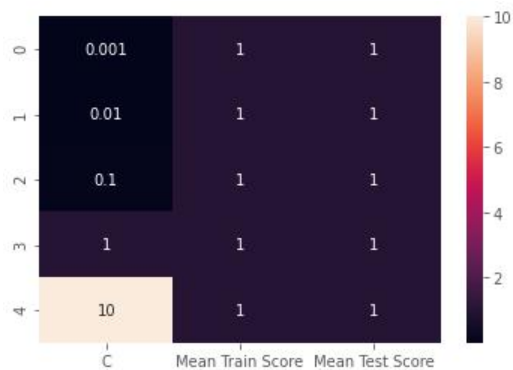


图 5-18 平均分数

Figure 5-18 mean score

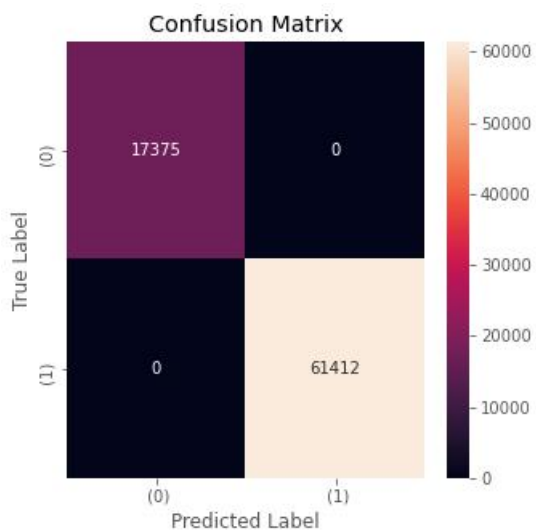


图 5-19 混淆矩阵

Figure 5-19 Confusion matrix

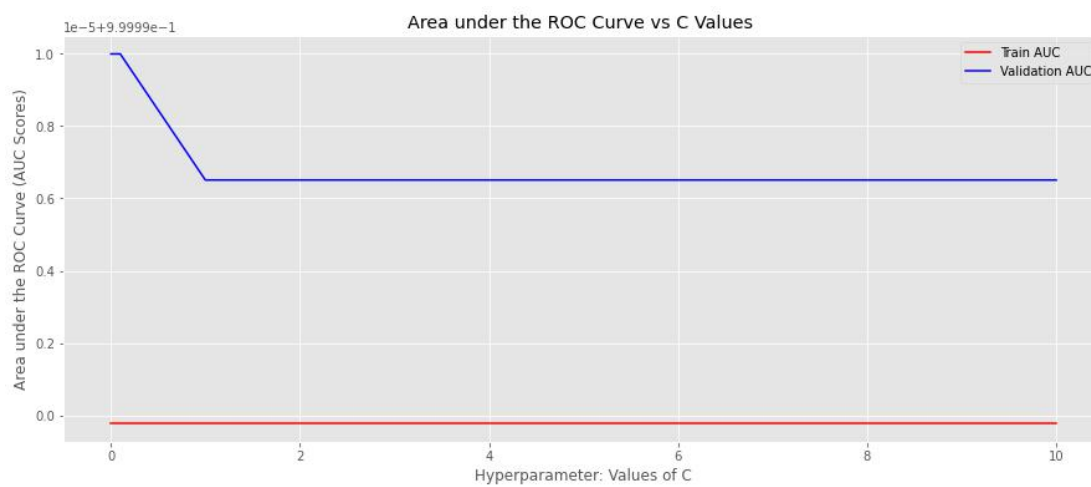


图 5-20 ROC 与 C 分布关系

Figure 5-20 Republic of China and C distribution

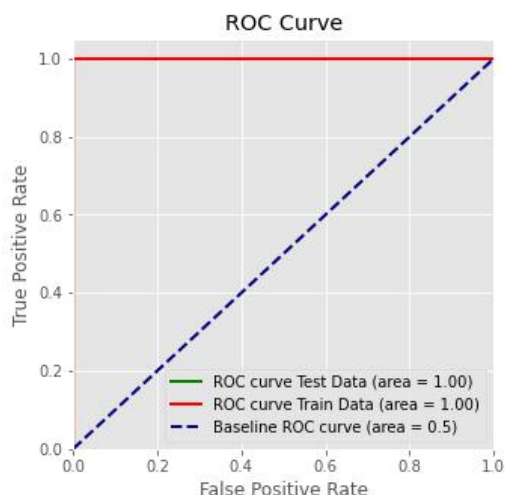


图 5-21 ROC 曲线

Figure 5-21 ROC curve

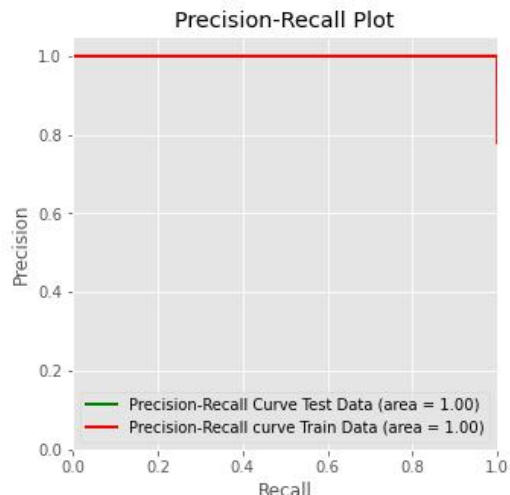


图 5-22 召回曲线

Figure 5-22 Recall curve

### 5.3 研究启示

本设计从分词模型基于 TF-IDF 和 N-gram 模型作为分词模型基础上使用两种正则化方法构建逻辑回归文本情感倾向性分类模型，以及深度学习模型展开分析，以亚马逊食品评论数据为训练集和测试集分析，都展现了较好的性能效果。从机器学习角度来看，将文本内容转化成结构化数据的过程十分关键，分词模型的不同直接影响到了模型训练的效果，考虑到食品评论独特特性，当针对食品商品的评论时该模型所展现较好的性能效果，同理根据此分类模型的构建同样可以应用于其他领域的数据集中进行个性化的分类。

本设计所使用的逻辑回归模型以及 MLP 模型在对亚马逊食品评论清洗后的数据大约 36 万条数据时，因为设计算法过于复杂在构建 MLP 时所需算力庞大。在面对大数据量条件下如何训练模型，对于深度学习算法，神经网络算法不仅仅要依赖于算力。部分模型可以根据很少的样本得出较好的模型效果，这对于减少训练成本和时间损耗更好，具有更高的现实意义。

## 6 总结与展望

### 6.1 总结

目前，在自然语言处理食品评论文本情感倾向性分析方向的研究处于空白阶段。在处理文本类型情感分类问题通常使用传统的定性方式，以及定量方式分析。定性分析存在的问题即为主观性过强，缺乏科学依据，缺少实用性强的，设计广的，健壮性强的规定体系。定量分析方向存在数据量的处理问题，模型的效果，以及模型的健壮性问题无法应对词汇数据不断更新的条件。分类器效果具有局限性，主要的工作内容如下：

- (1) 本设计针对食品评论的情感倾向性问题，使用 Kaggle 数据集提供的近十年的亚马逊食品评论数据集，以定性的角度出发探究影响评论文本的指标，进而确定食品评论数据集中指标间的关系进行数据清洗；构建两种分词模型，BOW-N-gram 模型以及 TF-IDF-N-gram 分词模型将清洗处理好的评论文本数据结构化；构建基于 L1,L2 的逻辑回归文本情感倾向性分类模型；构建基于多层感知机的文本情感倾向性分类模型；对模型进行优化测试。针对食品评论情感分析问题构建了高效的效果，其准确率均在 85%以上。
- (2) 本设计通过使用正则化的方法对构建的逻辑回归模型使用 Lasso 和岭回归进行正则化对多层感知机模型使用 L2 正则化进行模型的效果和稳定性优化测试来对抗干扰因素，避免过拟合现象。对两种分词模型进行训练比对分别使用一元，二元分类方法训练，构建了基于 TF-IDF-Bigram 的岭回归逻辑回归模型的效果最好正确率和召回率达到了 99%。在食品商品评论情感分析领域提供了较高的分类模型。
- (3) 本设计通过构建深度学习模型的文本情感分类模型，验证了深度学习模型的高效性，结构简单，将大量计算交给算力执行能够在短时间内训练出较好的分类模型，减少了大量人力物力，能够适应于大样本下的食品评论分类问题。在电商商品评论领域中涉及面广和杂，本设计的分析方法和构造的模型为其他领域商品评论提供了借鉴意义。



## 6.2 不足和展望

本设计所构建的食品评论文本情感分析方法中，所展示的定性分析并未涉及使用，在关于相关分析方法可使用偏序集决策的方法用以减少主观影响，以少量样本构建分类评价指标。从宏观角度看待食品评论情感问题，本文所探讨的方向和对应方法并未完全展示，缺少一部分研究。如何使用线性拓展链构建食品评论情感分类模型，如何使用小样本条件下预测情感倾向性分类，如何对抗困难样本多带来的波动将成为今后的研究方向。

在研究的食品评论文本情感分析领域中使用深度学习方法进行处理文本信息还处于开始阶段，对于使用神经网络算法处理相关问题还可以做进一步拓展，由于当前实验条件有限，未来研究可以使用更多方法，提高参数和训练样本数量进而提高模型的效果。在构建分词模型同样可以考虑使用神经网络算法构建分词模型，在未来研究中提供参考方向。构建完整的食品评论的情感倾向性分析体系，为商家提供更加精准的反馈信息，为用户提供更好的服务，可以作为未来的应用层研究。

## 致谢

这次毕业设计顺利进行，离不开太多人的鼓励与支持

十分感谢我的毕业设计导师温老师。在我迷茫，找不到方向的时候一直给我鼓励和支持，让我学到了很多无论是知识层面还是精神层面。我能坚持科研学习离不开温老师的教诲。在毕业设计时，老师通过丰富的指导经验给我完成毕业设计提供最重要的帮助。

十分感谢在自然语言处理情感分析领域研究的学者们，感谢吴恩达的《深度学习》课程教会我理论基础，感谢 Kaggle 平台提供的亚马逊食品评论数据集。感谢参考文献学者提供了针对特定领域构建情感倾向性分析模型的思路，感谢 scikit-learn 提供的开源代码为模型构建提供重要的帮助。感谢开发者们的无私开源为计算机科学的成长提供帮助，为我解决了开发方面的疑惑。

十分感谢我家人朋友们，感谢我的亲人对我无条件的支持和肯定，成为我有力的依靠。感谢我的朋友们给我精神和技术的支持。在计算机技术上我欠缺太多，很多都是理论上的，感谢舍友对我的技术上的帮助。

在求学的道路上道阻且长，经历了很多挫折，遇到了很多困难。感谢大家的支持让我体会到了学习的快乐。从一开始的数学建模到机器学习再到各个数据挖掘的应用领域，我切实的感受到了计算机的无穷无尽的魅力。在辽宁工程技术大学求学的四年间，信管老师们在学术上和思想上给我带来了深远的影响，在此向他们表示衷心的感谢。希望我能继续坚持奋斗，绽放理想之花。

## 参考文献

- [1] 朱少杰. 基于深度学习的文本情感分类研究[D]. 哈尔滨工业大学, 2014.
- [2] 赵妍妍, 秦兵, 刘挺. 文本情感分析[J]. 软件学报, 2010, 21(08):1834-1848.
- [3] 杜朋, 卢益清, 韩长风. 基于 Transformer 模型的商品评论情感分析[J]. 中文信息学报, 2021, 35(02):125-132.
- [4] 李涵昱, 钱力, 周鹏飞. 面向商品评论文本的情感分析与挖掘[J]. 情报科学, 2017, 35(01):51-55+61.
- [5] 范昊, 李鹏飞. 基于 FastText 字向量与双向 GRU 循环神经网络的短文本情感分析研究——以微博评论文本为例[J]. 情报科学, 2021, 39(04):15-22.
- [4] 彭云. 提取商品特征和情感词的语义约束 LDA 模型研究[D]. 江西财经大学, 2016.
- [5] 丁蔚. 基于词典和机器学习组合的情感分析[D]. 西安邮电大学, 2017.
- [6] 宋明, 刘彦隆. Bert 在微博短文本情感分类中的应用与优化[J]. 小型微型计算机系统, 2021, 42(04):714-718.
- [7] 王素格, 杨文琦, 张文跃. 融合多种特征的舆情时序文本情感分类方法[J/OL]. 山西大学学报(自然科学版)
- [8] 胡明哲. 关于酒店评论情感倾向的统计分析[D]. 华中师范大学, 2019.
- [9] 刘智鹏. 基于深度学习的商品评价情感分析与研究[D]. 重庆大学, 2018.
- [10] 刘小明, 张英, 郑秋生. 基于卷积神经网络模型的互联网短文本情感分类[J]. 计算机与现代化, 2017(04):73-77.
- [11] 崔永生. 在线评论文本挖掘对电商的影响研究[J]. 中国商论, 2018(33):17-23.
- [12] 余传明, 李浩男, 安璐. 基于多任务深度学习的文本情感原因分析[J]. 广西师范大学学报(自然科学版), 2019, 37(01):50-61.
- [13] 熊乐. 电商评论情感分析关键技术研究[D]. 南昌大学, 2018.
- [14] 宋晓雷, 王素格, 李红霞. 面向特定领域的产品评价对象自动识别研究[J]. 中文信息学报, 2010, 24(01):89-93.
- [15] 李素科, 蒋严冰. 基于情感特征聚类的半监督情感分类[J]. 计算机研究与发展, 2013, 50(12):2570-2577.

[16]. Statistics - Intelligent Data Analysis; New Intelligent Data Analysis Study Findings Recently Were Reported by Researchers at National Digital Switching Systems Engineering & Technology R&D Center (An Attention-gated Convolutional Neural Network for Sentence Classification) [J]. Network Weekly News, 2019.

## 附录 A 译文

### 英文文本的情感分析

#### 摘要

社交媒体网站越来越受欢迎，已经产生了大量的数据，吸引了研究人员、决策者和公司来调查人们在各个领域的观点和想法。情绪分析最近被认为是一个新兴的话题。决策者、公司和服务提供商以及公认的情绪分析作为一种有价值的改进工具。本文的研究目的是获得一个推文数据集，并应用不同的机器学习算法对文本进行分析和分类。本文研究了使用不同分类器进行分类平衡和不平衡数据集分类的文本分类准确性。结果发现，不同分类器的性能随数据集的大小而不同。结果还表明，朴素的 Byes 和 ID3 比其他分类器具有更好的精度水平，而平衡数据集的性能也更好。不同的分类器(K-NN、决策树、随机森林和随机树)在不平衡数据集下提供了更好的性能。

#### 1 产品简介

最近社交媒体的扩大改变了交流、分享和获取信息。除此之外，许多公司使用社交媒体通过分析对话的内容[5]来评估他们的业务表现。这包括收集客户对服务、设施和产品的意见。通过提高服务质量，探索这些数据在消费者保留中发挥着重要作用。社交媒体网站、脸书和推特提供了有价值的信息，企业主不仅可以用来跟踪和分析客户对其业务的意见，还可以分析竞争对手的意见[8-11]。此外，这些有价值的信息还吸引了那些寻求改善为[8,9,12,13]提供的服务的决策者。在这篇研究论文中，调查了几篇研究推特数据分类和分析的研究论文，以调查用于文本分类的方法和工具。本文的作者的目的是获得开源数据集，然后使用机器学习方法，应用不同的分类算法，即分类器，进行文本分类实验。作者利用几个分类器对两个版本的数据集的文本进行了分类。第一个版本是不平衡数据集，第二个版本是平衡数据集。然后，作者比较了每个使用的分类器对两个数据集文本分类的分类准确性。

#### 2 实验设置

在本节中，描述了数据集，并讨论了在实验中使用的设置和评估技术。对 tweet 类别的预测进行了两次测试，第一次在不平衡数据集上，第二次在平衡数据集上，

如下。

在六个不平衡数据集上应用了决策树、朴素贝叶斯、随机森林、K-NN、ID3 和随机树分类器。

在平衡数据集上的实验：在这个实验中，通过手工程序对不平衡数据集相关的挑战被解决，以避免有偏见的预测和误导的准确性。每个数据集中的大多数类几乎与少数类相等，即许多正、负和中性，在平衡数据集中几乎与表 3 所示的情况相同。

## 2.1 数据集说明

我们从这项工作中最大的在线数据科学社区之一 **Kaggle** 那里获得了一个数据集。它由超过 14000 条推文组成，标签要么是（正面的、负面的或中立的）。数据集也被分成六个数据集；每个数据集都包括关于六家美国航空公司（美美航空航空、达美航空、西南航空、维珍美国航空、美国航空公司）之一的推文。首先，我们总结了所得数据集的细节，如下表 1 所示。

Table 1: Summary of obtained Dataset

	American Airline Companies					
	Virgin America	United	Delta	Southwest	US Airways	American
Number of Tweets	504	3822	2222	2420	2913	2759
Positive Tweets	152	492	544	570	269	336
Negative Tweets	181	2633	955	1186	2263	1960
Neutral Tweets	171	697	723	664	381	463

## 2.2 数据集清理

在本节中，作者描述了数据集准备过程中的后续过程。作者使用 **Rapid** 次要软件进行推特分类。作者遵循以下方法：

- 1) 将数据集分割成一个训练集和测试集。
- 2) 使用 ReadExcel 加载数据集，即使用 ReadExcel 运算符将 excel 文件加载到 RapidMinor 软件中。
- 3) 利用以下操作符应用预处理。

将用例运算符转换为将文本转换为小写。

标记化运算符，将文本分成一系列标记。

过滤停止字操作符，删除停止字，如：is、t、at 等。

筛选器令牌（按长度计算）运算符：要根据长度删除令牌，在此模型中，最小字符为 3，最大字符为 20，任何其他不匹配规则的令牌都将被删除。

步骤运算符：将单词转换为基本形式。

## 2.3 数据集培训

每个数据集都被分为两部分。第一部分包含数据集推文总数的 66%，用于训练机器将数据分类在一个属性下，该属性用于将推文分类为（正或负或中性）。其余 34% 的 tweet 用于将 tweet 的属性分类为（正或负或中性），即测试集。

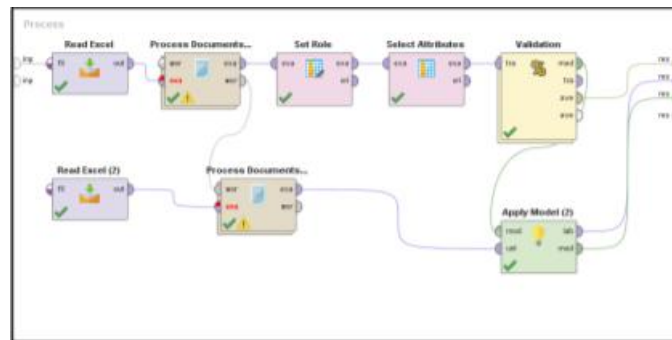


Figure 1: Summarization of the Process Model

## 2.4 数据集分类

在本节中，作者描述了推文分类技术中的步骤。设置角色运算符用于允许系统将情绪识别为目标变量。“选择属性”运算符用于删除任何缺少任何值的属性。然后在验证操作符中，数据集分为两部分（培训和测试）。我们使用三分之二的数据集来训练数据集，最后三分之一来评估模型。不同的机器学习算法被用于训练数据集(决策树、朴素贝叶斯、随机森林、KNN、ID3 和随机树)。为了测试模型，性能操作符用来测量模型的性能。

## 3 研究结论

社交媒体网站在不同年龄段的人中越来越受欢迎。推特、脸书、智能和快速聊天等平台可以让人们表达他们的想法、意见、评论和想法。因此，每天都生成大量的数据，书面文本是生成数据最常见的形式之一。企业主、决策者和研究人员越来越被社交媒体网站上生成和存储的有价值的大量数据所吸引。情绪分析是一个自然语言处理领域，它越来越吸引研究人员、政府当局、企业主、服务提

供应商和公司来改进产品、服务和研究。在这篇研究论文中，作者旨在调查情绪分析的方法。因此，对 16 篇研究推特文本分类和分析的研究论文进行了调查。作者还旨在评估不同的机器学习算法，用于分类情绪的积极或消极，或中性。本实验旨在比较在 16 篇论文中使用的不同分类器的效率和性能。这些分类器是(决策树、朴素贝叶斯、随机森林、K-NN、ID3 和随机树)。此外，作者在平衡数据集后，将数据集上应用同一分类器，研究了平衡数据集因子。目标数据集包括关于 6 家美国航空公司（联合航空、达美航空、西南航空、维珍美航空、美国航空和美国航空）的 6 个数据集；它由大约 14000 条推文组成。作者报告说，该分类器的精度结果在一些数据集中非常高，而在其他数据集中则很低。作者指出，数据集的大小是其原因。在平衡数据集上，朴素贝叶斯分类器、决策树和 ID3 大多优于其他分类器，并给出了几乎相同的精度水平。维珍美国数据集的分类器报告说，由于其体积小，精度最低。在不平衡数据集上，结果表明，当朴素的模糊识别码和 ID3 被应用于平衡数据集上时，它比其他分类器具有更好的精度水平。而(KNN、决策树、随机森林和随机树)则更好地理解不平衡的数据集。



## 附录 B 外文文献

### Sentiment Analysis in English Texts

#### ABSTRACT

The growing popularity of social media sites has generated a massive amount of data that attracted researchers, decision-makers, and companies to investigate people's opinions and thoughts in various fields. Sentiment analysis is considered an emerging topic recently. Decision-makers, companies, and service providers as well-considered sentiment analysis as a valuable tool for improvement. This research paper aims to obtain a dataset of tweets and apply different machine learning algorithms to analyze and classify texts. This research paper explored text classification accuracy while using different classifiers for classifying balanced and unbalanced datasets. It was found that the performance of different classifiers varied depending on the size of the dataset. The results also revealed that the Naive Byes and ID3 gave a better accuracy level than other classifiers, and the performance was better with the balanced datasets. The different classifiers (K-NN, Decision Tree, Random Forest, and Random Tree) gave a better performance with the unbalanced datasets.

#### 1 Introduction

The recent widening expansion of social media has changed communication, sharing, and obtaining information [1–4]. In addition to this, many companies use social media to evaluate their business performance by analysing the conversations' contents [5]. This includes collecting customers' opinions about services, facilities, and products. Exploring this data plays a vital role in consumer retention by improving the quality of services [6, 7]. Social media sites such as Instagram, Facebook, and Twitter offer valuable data that can be used by business owners not only to track and analyse customers' opinions about their businesses but also that of their competitors [8–11]. Moreover, these valuable data attracted decision-makers who seek to improve the services provided [8, 9, 12, 13]. In this research paper, several research papers that studied Twitter's data classification and analysis for

different purposes were surveyed to investigate the methodologies and approaches utilized for text classification. The authors of this research paper aim to obtain open-source datasets then conduct text classification experiments using machine learning approaches by applying different classification algorithms, i.e., classifiers. The authors utilized several classifiers to classify texts of two versions of datasets. The first version is unbalanced datasets, and the second is balanced datasets. The authors then compared the classification accuracy for each used classifier on classifying texts of both datasets.

## 2 Proposed Approach

In this work, the authors implemented and evaluated different classifiers in classifying the sentiment of the tweets. It's by utilizing RapidMiner software. Classifiers were applied on both balanced and unbalanced datasets. Classifiers used are Decision Tree, Naïve Bayes, Random Forest, K-NN, ID3, and Random Tree.

## 3 Experiment Setup

In this section, the dataset is described as well as the settings and evaluation techniques are used in the experiments have been discussed. The prediction for the tweet category is tested twice— the first time on an unbalanced data set and the second time on a balanced dataset as below.

- Experiments on the unbalanced dataset: Decision Tree, Naïve Bayes, Random Forest, K-NN, ID3, and Random Tree classifiers were applied on six unbalanced datasets.
- Experiments on the balanced dataset: In this experiment, the challenges related to unbalanced datasets were tackled by manual procedures to avoid biased predictions and misleading accuracy. The majority class in each dataset almost equalized with the minority classes, i.e., many positive, negative, and neutral, practically the same in the balanced dataset as represented in Table 3.

### 3.1 Dataset Description

We obtained a dataset from Kaggle, one of the largest online data science communities in this work. It consists of more than 14000 tweets, labeled either

(positive, negative, or neutral). The dataset was also split into six datasets; each dataset includes tweets about one of six American airline companies (United, Delta, Southwest, Virgin America, US Airways, and American). Firstly, we summarized the details about the obtained datasets, as illustrated in Table 1 below.

Table 1: Summary of obtained Dataset

	American Airline Companies					
	Virgin America	United	Delta	Southwest	US Airways	American
Number of Tweets	504	3822	2222	2420	2913	2759
Positive Tweets	152	492	544	570	269	336
Negative Tweets	181	2633	955	1186	2263	1960
Neutral Tweets	171	697	723	664	381	463

### 3.2 Dataset Cleansing

In this section, the authors described the followed procedure in the dataset preparation. The authors utilized RapidMinor software for tweet classification. Authors followed the methods described below:

- 4) Splitting the dataset into a training set and test set.
- 5) Loading the dataset, i.e., excel file into RapidMinor software using Read Excel operator.
- 6) Applying preprocessing by utilizing the below operators.
  - Transform Cases operator to transform text to lowercase.
  - Tokenize operator to split the text into a sequence of tokens.
  - Filter Stop words operator to remove stop words such as: is, the, at, etc.
  - Filter Tokens (by length) operator: to remove token based on the length, in this model, minimum characters are 3, and maximum characters are 20 any other tokens that don't match the rule will be removed.
  - Stem operator: to convert words into base form.

### 3.3 Dataset Training

Each of the datasets was divided into two-part. The first part contains 66% of the

total number of tweets of the data set, and it is used to train the machine to classify the data under one attribute, which is used to classify the tweets to either (positive or Negative or Neutral). The remaining 34% of tweets were used to classify tweets' attribute to (positive or Negative or Neutral), i.e., test set.

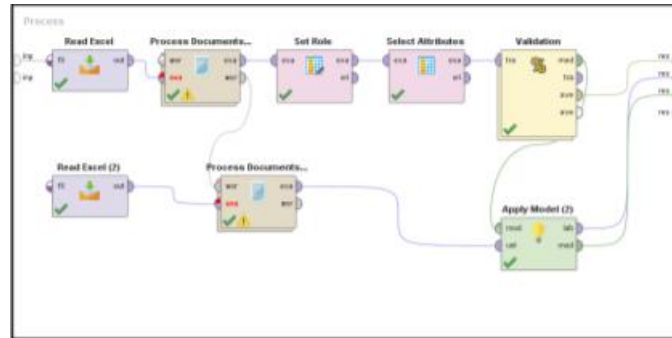


Figure 1: Summarization of the Process Model

### 3.4 Dataset Classifying

In this section, the authors described the steps in the tweet's classification techniques.

- Set Role operator is used to allow the system to identify sentiment as the target variable.
- Select Attributes operator is used to removing any attribute which has any missing values.
- Then in the validation operator, the dataset is divided into two parts (training and test). We used Two-thirds of the dataset to train the dataset and the last one-third to evaluate the model.
- Different machine learning algorithms are used for training the dataset (Decision Tree, Naïve Bayes, Random Forest, KNN, ID3, and Random Tree).
- For testing the model, the Performance operator utilized to measure the performance of the model.

## 4 Conclusions

Social media websites are gaining very big popularity among people of different ages. Platforms such as Twitter, Facebook, Instagram, and Snapchat allowed people to express their ideas, opinions, comments, and thoughts. Therefore, a huge amount of

data is generated daily, and the written text is one of the most common forms of the generated data. Business owners, decisionmakers, and researchers are increasingly attracted by the valuable and massive amounts of data generated and stored on social media websites. Sentiment Analysis is a Natural Language Processing field that increasingly attracted researchers, government authorities, business owners, services providers, and companies to improve products, services, and research. In this research paper, the authors aimed to survey sentiment analysis approaches. Therefore, 16 research papers that studied Twitter's text classification and analysis were surveyed. The authors also aimed to evaluate different machine learning algorithms used to classify sentiment to either positive or negative, or neutral. This experiment aims to compare the efficiency and performance of different classifiers that have been used in the sixteen papers that are surveyed. These classifiers are (Decision Tree, Naïve Bayes, Random Forest, K-NN, ID3, and Random Tree). Besides, the authors investigated the balanced dataset factor by applying the same classifiers twice on the dataset, one on the unbalanced and the other, after balancing the dataset. The targeted dataset included six datasets about six American airline companies (United, Delta, Southwest, Virgin America, US Airways, and American); it consists of about 14000 tweets. The authors reported that the classifier's accuracy results were very high in some datasets while low in others. The authors indicated that the dataset size was the reason for that. On the balanced dataset, the Naïve Bayes classifier, Decision Tree, and ID3 were mostly better than other classifiers and have given the almost same level of accuracy. The classifiers with Virgin America dataset reported the lowest level of accuracy due to its small size. On the unbalanced dataset, results show that the Naive Byes and ID3 gave a better level of accuracy than other classifiers when it's applied on the balanced datasets. While (KNN, Decision Tree, Random Forest, and Random Tree) gave a better understanding of the unbalanced datasets.

## 附录 C

此代码所包含包括逻辑回归分类模型构建核心函数构建代码。详见如下所示；

```
# 导入包

from sklearn.metrics import precision_recall_curve
from sklearn.model_selection import TimeSeriesSplit

# def top_features()

#标准化数据矩阵

def standardize(data, with_mean):

    scalar = StandardScaler(with_mean=with_mean)

    std=scalar.fit_transform(data)

    return (std)

# 绘制 ROC 曲线

def plot_roc_curve(clf, train_data, train_label, test_data, test_label):

    fpr = dict()

    tpr = dict()

    roc_auc = dict()

    train_prob = clf.predict_proba(train_data)

    train_label_prob = train_prob[:,1]

    fpr["Train"], tpr["Train"], threshold = roc_curve(train_label,
train_label_prob, pos_label='Positive')

    roc_auc["Train"] = auc(fpr["Train"], tpr["Train"])

    test_prob = clf.predict_proba(test_data)

    test_label_prob = test_prob[:,1]

    fpr["Test"], tpr["Test"], threshold = roc_curve(test_label,
```

```

test_label_prob, pos_label='Positive')

roc_auc["Test"] = auc(fpr["Test"], tpr["Test"])

plt.figure(figsize=(5,5))

linewidth = 2

plt.plot(fpr["Test"], tpr["Test"], color='green', lw=linewidth,
label='ROC curve Test Data (area = %0.2f)' % roc_auc["Test"])

plt.plot(fpr["Train"], tpr["Train"], color='red', lw=linewidth,
label='ROC curve Train Data (area = %0.2f)' % roc_auc["Train"])

plt.plot([0, 1], [0, 1], color='navy', lw=linewidth, linestyle='--',
label='Baseline ROC curve (area = 0.5)')

plt.xlim([0.0, 1.0])

plt.ylim([0.0, 1.05])

plt.xlabel('False Positive Rate')

plt.ylabel('True Positive Rate')

plt.title('ROC Curve')

plt.legend(loc="lower right")

plt.show()

#-----
-----
-----

# 绘制召回曲线

def plot_pr_curve(clf, train_data, train_label, test_data, test_label):

    precision = dict()

    recall = dict()

    pr_auc = dict()

    train_prob = clf.predict_proba(train_data)

```

```

train_label_prob = train_prob[:,1]

precision["Train"], recall["Train"], threshold =
precision_recall_curve(train_label, train_label_prob,
pos_label='Positive')

pr_auc["Train"] = auc(recall["Train"], precision["Train"])

test_prob = clf.predict_proba(test_data)

test_label_prob = test_prob[:,1]

precision["Test"], recall["Test"], threshold =
precision_recall_curve(test_label, test_label_prob,
pos_label='Positive')

pr_auc["Test"] = auc(recall["Test"], precision["Test"])

plt.figure(figsize=(5,5))

linewidth = 2

plt.plot(recall["Test"], precision["Test"], color='green',
lw=linewidth, label='Precision-Recall Curve Test Data (area = %0.2f)' %
pr_auc["Test"])

plt.plot(recall["Train"], precision["Train"], color='red',
lw=linewidth, label='Precision-Recall curve Train Data (area = %0.2f)' %
pr_auc["Train"])

#plt.plot([0, 1], [0, 1], color='navy', lw=linewidth, linestyle='--',
label='Baseline Precision-Recall curve (area = 0.5)')

plt.xlim([0.0, 1.0])

plt.ylim([0.0, 1.05])

plt.xlabel('Recall')

plt.ylabel('Precision')

plt.title('Precision-Recall Plot')

plt.legend(loc="lower right")

```



```

plt.show()

# 计算指标性能

def metrics_performance(grid, train_data, train_label, test_data,
test_label):

    clf = grid.best_estimator_

    clf.fit(train_data, train_label)

    test_label_pred = clf.predict(test_data)

    test_prob = clf.predict_proba(test_data)

    test_label_prob = test_prob[:,1]

    print("Accuracy : ", accuracy_score(test_label, test_label_pred,
normalize=True) * 100)

    print("Points : ", accuracy_score(test_label, test_label_pred,
normalize=False))

    print("Precision : ",
np.round(precision_score(test_label ,test_label_pred,
pos_label='Positive'),4))

    print("Recall : ", recall_score(test_label, test_label_pred,
pos_label='Positive'))

    print("F1-score : ", f1_score(test_label,test_label_pred,
pos_label='Positive'))

    print("AUC : ", np.round(roc_auc_score(test_label,
test_label_prob),4))

    print ('\nClasification Report :')

    print(classification_report(test_label,test_label_pred))

    print('\nConfusion Matrix :')

    cm = confusion_matrix(test_label ,test_label_pred)

```

```

df_cm = pd.DataFrame(cm, index = [' (0)', ' (1)'], columns = [' (0)', '
(1)'])

plt.figure(figsize = (5,5))

ax = sns.heatmap(df_cm, annot=True, fmt='d')

ax.set_xlabel("Predicted Label")

ax.set_ylabel("True Label")

ax.set_title('Confusion Matrix')

plot_roc_curve(clf, train_data, train_label, test_data, test_label)

plot_pr_curve(clf, train_data, train_label, test_data, test_label)

# 绘制网格搜索结果

def plot_gridsearch_result(grid):

    cv_result = grid.cv_results_

    auc_train = list(cv_result['mean_train_score'])

    auc_test = list(cv_result['mean_test_score'])

    params = cv_result['params']

    hp_values = [p['C'] for p in params] #获取列表 C

    # 绘制 GridSearchCV 结果的 Heatmap

    cv_result = {'C':hp_values, 'Mean Train Score':auc_train, 'Mean Test
Score':auc_test} # dataframe of cv_result

    cv = pd.DataFrame(cv_result)

    sns.heatmap(cv, annot=True)

#绘制误差与 C 值

plt.figure(figsize=(15,6))

```

```

plt.plot(hp_values , auc_train, color='red', label='Train AUC')
plt.plot(hp_values , auc_test, color='blue', label='Validation AUC')
plt.title('ROC 曲线下面积与 C 值 ')
plt.xlabel('超参数: C 的值')
plt.ylabel('ROC 曲线下面积 (AUC 分数)')
plt.legend()
plt.show()

# 应用 GridSearchCV 的函数
def gridSearchCV(train_data, train_label, regularization):
    param = {'C' : [1000,500,100]}
    model = LogisticRegression(penalty=regularization,
solver='liblinear', random_state=0)
    cv = TimeSeriesSplit(n_splits=10).split(train_data)
    grid = GridSearchCV(estimator=model, param_grid=param, cv=cv,
n_jobs=-1, scoring='roc_auc', verbose=40, return_train_score=True)
    grid.fit(train_data, train_label)

    print("最佳参数 : ", grid.best_params_)
    print("最佳得分 : ", grid.best_score_)
    print("最佳估算器 : ", grid.best_estimator_)

    return grid

# In[2]:
def logisticRegression(train_data, train_label, test_data, test_label,
regularization):
    grid = gridSearchCV(train_data, train_label, regularization)
    plot_gridsearch_result(grid)
    metrics_performance(grid, train_data, train_label, test_data,
test_label)

```

下列代码展示 MLP 模型正则优化的核心代码部分：

```

import numpy as np

from matplotlib import pyplot as plt

from matplotlib.colors import ListedColormap

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import StandardScaler

from sklearn.datasets import make_moons, make_circles,
make_classification

from sklearn.neural_network import MLPClassifier

from sklearn.pipeline import make_pipeline

h = .02 # step size in the mesh

alphas = np.logspace(-1, 1, 5)

classifiers = []
names = []

for alpha in alphas:
    classifiers.append(make_pipeline(
        StandardScaler(),
        MLPClassifier(
            solver='lbfgs', alpha=alpha, random_state=1, max_iter=2000,
            early_stopping=True, hidden_layer_sizes=[100, 100],
        )
    ))
    names.append(f"alpha {alpha:.2f}")

X, y = make_classification(n_features=2, n_redundant=0, n_informative=2,
                           random_state=0, n_clusters_per_class=1)

rng = np.random.RandomState(2)

```

```
X += 2 * rng.uniform(size=X.shape)

linearly_separable = (X, y)

datasets = [make_moons(noise=0.3, random_state=0),
            make_circles(noise=0.2, factor=0.5, random_state=1),
            linearly_separable]

figure = plt.figure(figsize=(17, 9))

i = 1

# iterate over datasets
for X, y in datasets:

    # split into training and test part
    x_train, x_test, y_train, y_test = train_test_split(X, y,
test_size=.4)

    x_min, x_max = X[:, 0].min() - .5, X[:, 0].max() + .5
    y_min, y_max = X[:, 1].min() - .5, X[:, 1].max() + .5
    xx, yy = np.meshgrid(np.arange(x_min, x_max, h),
                          np.arange(y_min, y_max, h))

    # just plot the dataset first
    cm = plt.cm.RdBu
    cm_bright = ListedColormap(['#FF0000', '#0000FF'])
    ax = plt.subplot(len(datasets), len(classifiers) + 1, i)

    # Plot the training points
    ax.scatter(x_train[:, 0], x_train[:, 1], c=y_train, cmap=cm_bright)

    # and testing points
    ax.scatter(x_test[:, 0], x_test[:, 1], c=y_test, cmap=cm_bright,
alpha=0.6)
```

```

ax.set_xlim(xx.min(), xx.max())

ax.set_ylim(yy.min(), yy.max())

ax.set_xticks(())

ax.set_yticks(())

i += 1

# iterate over classifiers
for name, clf in zip(names, classifiers):

    ax = plt.subplot(len(datasets), len(classifiers) + 1, i)

    clf.fit(x_train, y_train)

    score = clf.score(x_test, y_test)

    # Plot the decision boundary. For that, we will assign a color to
each
    # point in the mesh [x_min, x_max] x [y_min, y_max].
    if hasattr(clf, "decision_function"):
        Z = clf.decision_function(np.c_[xx.ravel(), yy.ravel()])
    else:
        Z = clf.predict_proba(np.c_[xx.ravel(), yy.ravel()])[:, 1]

    # Put the result into a color plot
    Z = Z.reshape(xx.shape)
    ax.contourf(xx, yy, Z, cmap=cm, alpha=.8)

    # Plot also the training points
    ax.scatter(x_train[:, 0], x_train[:, 1], c=y_train,
cmap=cm_bright,
               edgecolors='black', s=25)

    # and testing points

```

```
ax.scatter(x_test[:, 0], x_test[:, 1], c=y_test, cmap=cm_bright,
           alpha=0.6, edgecolors='black', s=25)

ax.set_xlim(xx.min(), xx.max())
ax.set_ylim(yy.min(), yy.max())
ax.set_xticks(())
ax.set_yticks(())
ax.set_title(name)
ax.text(xx.max() - .3, yy.min() + .3, ('%.2f' % score).lstrip('0'),
        size=15, horizontalalignment='right')

i += 1

figure.subplots_adjust(left=.02, right=.98)
plt.show()
```