

伊斯法罕理工大学

电气与计算机工程系

推特情绪对股市趋势的影响

作者：
Melvin Mokhtari（melvin.mokhtari@ec.iut.ac.ir）

Ali Seraj
Niloufar Saeedi
Adel Karshenas

作者声明

我们声明，上述论文所包含的工作为原创，是文中所列作者开展研究所得的成果，且我们是该项工作的唯一作者。我们谨此确认，本出版物不存在已知的利益冲突，且本研究未获得可能影响其结论的财务资助。

我们还确认，该项工作未侵犯他人的任何版权或财产权（包括许可），且不存在剽窃行为。若要使用受版权保护作品（包括网站）中的图、表格、动画或文本引文，均已获得版权持有人的必要许可。通过此声明，我们保证已遵守有关知识产权的规定。

摘要

网络是一个广阔的虚拟空间，人们可以在其中分享观点，影响生活的方方面面，并对营销和沟通产生影响。由于社交媒体普及广泛且发布消息十分便捷，人们可以在其中获取最新、最全面的信息。因此，社交媒体被视为精准预测市场的重要资源。尤其是推特已发展成为了解用户情感的有力工具。

本学位论文研究推文对股票代码趋势的影响。我们分析了三个月内推特上标准普尔500指数中排名前五的股票代码的成交量、情感和提及情况。本研究采用了长短期记忆网络、伯努利朴素贝叶斯和随机森林这三种算法。研究结果揭示，股价与Twitter情绪之间存在显著相关性。

关键词： 自然语言处理、机器学习、情感分析、数据挖掘、社交媒体、推特、股票市场

目录

作者声明 **iii**

摘要与关键词 v

目录 vii

图目录 xi

xiii

表目录 **xiii**

1 引言 1

1.1 概述.1 1.2 动机与目标.2 1.3 方法论.3 1.4 文档结构.4

2 相关工作 **5**

2.1 情绪与情感之间的类比.5 2.2 情感分析.7 2.2.1 情感分析层级.7 2.2.1.1 文档级情感分类.7 2.2.1.2 句子级情感分类.7 2.2.1.3 实体与方面级情感分类 . . . . . 8 2.2.2 情感分类技术 .8 2.2.2.1 基于词典的方法 .8 2.2.2.2 机器学习方法 .9

监督学习.10 回归.10 分类.10 无监督学习.10 聚类.10 关联.10 2.2.2.3 混合方法.10 2.2.3 算法方法.11 2.2.3.1 朴素贝叶斯分类器.11 朴素贝叶斯分类器的优点.12 朴素贝叶斯分类器的缺点.12 2.2.3.2 随机森林分类器.13 随机森林分类器的优点.13 随机森林分类器的缺点.14 2.2.3.3 长短期记忆模型（LSTM） . . . . . 14 2.3 情感分析的应用 .15 股票市场 .15 电子商务 .15 2.3.1 推特情感分析 .16 政治.16 商业.16 2.4 数据收集与准备.16 2.4.1 推特数据收集方法.17 2.4.1.1 Twitter API.17 2.4.1.2 Tweepy.17 2.4.1.3 Twint.17 2.4.1.4 预处理数据集.18

3 方法 学方法与技术框架 19

3.1 数据.19

3.1.1 Twitter数据收集.19 3.1.2 金融数据收集.20 3.1.3 数据预处理.20 3.1.4 标注.23

3.2 算法实现.23

3.2.1 随机森林分类器.24 3.2.2 朴素贝叶斯（伯努利）分类器.25 3.2.3 长短期记忆模型（长短期记忆网络）.25

4 性能评估 29

4.1 性能评估的图解结构.29 4.2 结果详细评估.30 4.2.1 Apple股票（$AAPL）评估.30 4.2.2 Facebook股票（$FB）评估.36 4.2.3 Microsoft股票（$MSFT）评估.39 4.2.4 特斯拉股票（$TSLA）评估.42 4.2.5 亚马逊股票（$AMZN）评估.45 4.2.6 总体比较.47

5 结论 49

5.1 结果与讨论.49 5.2 展望与未来工作.50

参考文献 **51**

图目录

2.1 情绪的二维结构.6 2.2 情感分析的不同层级.8 2.3 不同的情感分类技术.9 2.4 K-means倾向于形成大小相同的簇.11 2.5 朴素贝叶斯分类器中的条件独立性假设 . 12 2.6 随机森林分类器工作原理示例.13 2.7 LSTM模型的架构.15 3.1 独热编码示例.22 3.2 Sigmoid和ReLU函数.26 4.1 AAPL看涨情绪图.31 4.2 AAPL的LSTM模型图.33 4.3 AAPL的朴素贝叶斯分类器图.34 4.4 AAPL的随机森林分类器图.35 4.5 脸书看涨情绪图.36 4.6 脸书的LSTM模型图.37 4.7 脸书的朴素贝叶斯分类器图.37 4.8 脸书的随机森林分类器图.38 4.9 MSFT看涨情绪图.39 4.10 MSFT的LSTM模型图.40 4.11 MSFT的朴素贝叶斯分类器图.40 4.12 MSFT的随机森林分类器图.41 4.13 TSLA看涨情绪图.42 4.14 TSLA的LSTM模型图.43 4.15 TSLA的朴素贝叶斯分类器图.43 4.16 TSLA的随机森林分类器图.44

4.17 AMZN看涨情绪图.45 4.18 AMZN的LSTM模型图.46 4.19 AMZN的朴素贝叶斯分类器图.46 4.20 AMZN的随机森林分类器图.47

表目录

2.1 Th情绪与情感之间的区别.6

第1章

引言

本章介绍研究的动机，描述问题，并界定研究范围和理论方法。随后介绍研究目标。最后，概述本学位论文的主要贡献，并说明文档的结构。

1.1 概述

近期的技术革命引发了前所未有的数据洪流，从根本上改变了我们看待社会科学与经济学的方式。如今，计算机和互联网已成为几乎每个人日常生活的一部分。互联网使用量持续增长，也推动了线上活动的类似增长，例如商业或新闻相关活动。由于这一趋势的影响，社交媒体的兴起为网络用户提供了一个平台，让他们能够表达对各种主题的观点。因此，过去几年里，这些平台上的事件和文本文件的创建数量显著增加；由此，技术系统之间的交互产生了海量数据集，以前所未有的方式记录集体行为。最后，我们可以从这个庞大的互联网活动数据库中，了解全球人口对各种经济、政治和文化现象的兴趣、担忧与意图。

推特是一个广泛使用的微博和社交网络平台，人们可以在这里阅读、撰写、发布和更新称为推文的短文本消息 [1–4]。2021年，推特的全球用户群约为4.2979亿。此外，预计到2025年，全球用户数将达到4.9748亿[5]。因此，这一微博和社交网络服务引起了政治、医疗保健和金融等多个领域研究人员的兴趣。

情感分析是对这些推文语义内容进行科学研究的统称。一般而言，情感分析是一种识别和分类

文本极性的方法，旨在按照标准分类确定某份特定文档的倾向是正面还是负面。一些研究人员认为，在金融市场背景下处理推特用户观点，能够让所有人获取与股票市场相关的信息，并利用这些信息预测股票价格的变化。事实上，自股票市场诞生以来，投资者始终寻求获得竞争优势。他们并非通过内幕交易等非法手段，而是谨慎且节制地利用信息。

本文以金融市场为案例研究，这是数据收集、分析和建模广泛应用的众多领域之一。根据我们的分析，社交层面的社交网络指标对于理解金融交易额尤其有帮助。事实上，羊群行为等集体现象常常会引发金融危机，展现出金融体系的复杂性质。因此，政策制定者迫切希望预判投资者中异常的集体行为，以便在必要时尽早采取干预措施。

在金融市场投资本身固有的风险和复杂性，促使人们开发出能够最大限度减少投资者损失并提高收益率的技术。这些技术利用机器学习算法，寻找个人对金融市场的看法与股票价格走势之间的关联，从而有可能预测市场的变化。

1.2 研究动机与目标

本学位论文聚焦于推特对金融市场的影响。研究将重点考察有关某家公司的推文如何影响其股价走势。

早期对股票市场进行预测的尝试，旨在确定股票价格是否可以预测。对此有两种观点：随机游走理论和有效市场假说（EMH） [6]。根据有效市场假说，股票指数主要反映现有投资者带来的消息，而非当前和历史价格。然而，随机游走理论认为，由于消息会在无法预测的时间出现，因此预测永远无法确保准确。

根据 Qian 等人的研究，有效市场假说和随机游走模型的基本假设已得到确认和阐释[7]。这些理论表明，利用不同的经济和商业指标可以进行一定程度的预测。根据有效市场假说强势版本的广泛应用，价格汇总了公开可获取的信息，并立即反映最新信息 [8]。人们普遍认为，新闻会影响市场的宏观经济走势。然而，研究表明，社交媒体热度会对微观经济层面产生显著影响，尤其是在标准普尔500指数和道琼斯工业平均指数等主要指数中 [9–12]。早期研究发现

投资者情绪和决策最终会推动市场 [7]。因此，考虑情感的模型将能提供更好的微观经济预测。

交易者在市场中持有头寸的时间很短，有时甚至只有片刻，因此许多投资者可能每天都会迅速加入或退出这些头寸。因此，许多投资者会寻找任何能让自己在进行市场押注时占据优势的因素。一些优秀的交易者开发了基于人工智能的机器人，用来识别社交媒体上的趋势，却没有深入理解其中所蕴含的信息。

如果我们收集与目标股票代码相关的推文和金融数据，利用框架处理数据以推断推文的情绪分数，然后将结果与股票价格变化进行比较，就能为股票市场创建一个实用的动量指标。

这些框架会根据极性对推文进行分类，以解决这些问题。推文的情感可根据其极性分为正面、中性或负面。推文分类后，会计算每日分数。

我们尤其关注以下主要研究问题：

- 关于某家公司的推文可能会对其股票未来走势产生哪些影响?
- 能否利用情感分析结果，根据动量判断某只股票的价格走势是强劲还是疲弱?

因此，了解情感能否用于预测股价趋势方向，是这项研究的重要组成部分。

1.3 方法论

正如前文所述，本研究的方法是收集推文数据，然后根据其情绪极性进行分类。然而，由于全球股票市场的范围极其广泛，本文将聚焦于标准普尔500指数中出现频率最高的一些公司。作出这一选择有多方面原因，其中包括：标准普尔500指数是全球应用最广泛且最具相关性的指数之一，被视为衡量美国大型股表现的最佳指标；其成分公司也广受社交媒体平台用户关注。

本研究使用 Python 编程语言完成。此外，在各个模块中创建的脚本均使用 Jupyter Notebook 和 PyCharm 集成开发环境实现。

可以使用官方 Twitter API 收集推文。然而，鉴于我们所需的数据量庞大，这种方式存在诸多限制，并不适用于我们的情况。可供选择的方案

很少，这使得寻找推特数据成为本研究面临的最大挑战之一。尽管还有其他选项，例如 Twint （一款全面且可靠的 Python 网络爬虫）或 Tweepy （一款用于提取推文的 Python 库），但我们使用了 Kaggle 网站上的预处理数据集 [13]来规避这些限制。

我们使用了标准普尔500指数中排名前五的成分股公司、话题标签“# STOCK”、现金标签“$ STOCK”，以及一个经过精心整理和标注的推文数据集，该数据集涵盖2020年4月9日至7月16日的推文，以说明股价回报与推文所反映情绪之间的关系。

1.4 文档结构

本文档共分为五章，分别介绍了实现前文所述目标所需的步骤。除包含一些引言内容的第1章外，本学位论文其余部分的安排如下。

第2章介绍了情感分析领域中的若干概念，并详细说明了这些概念的不同层次和类型。其关键组成部分包括相关理念、推特在这一领域的应用，以及各种算法的工作方法，尤其是机器学习方法。本章还介绍了方法论、数据收集、用于情感分析的数据集准备，以及模拟设置的简要概述。

第3章介绍了支撑对股市数据进行情感分析实践的基础理论和方法。此外，还阐述了许多概念、技术，以及算法实现和数据获取这两个模块的详细内容。

第4章详细评估了针对选定的五只股票开展的试验。此外，还通过图表讨论了金融数据与情感看涨情绪之间的关系，以阐明研究结果。

最后，第5章将呈现本学位论文的总体结论与摘要，以及有助于进一步发展的潜在未来研究方向。

第2章

相关工作

本章将探讨情感分析的最新进展。首先介绍背景及其基本理念，随后探讨如何将推特用作情感分析的有效工具。之后，我们将简要介绍各种算法策略，特别是机器学习方法。最后，我们将讨论多种推特数据收集方法，包括各自的优缺点，以及我们所选择的方法。

2.1 情绪与情感之间的类比

在日常生活中，“情绪”和“情感”这两个术语经常被互换使用。然而，在心理学中，它们被视为两个截然不同的概念。Watson和Tellegen提出的著名心理学模型在图2中描绘了情绪二维模型。1 [14]从心理学角度来看，情绪在激发行为方面发挥着关键作用。著名心理学家 Paul Ekman 提出了六种基本情绪：快乐、惊讶、悲伤、愤怒、恐惧和厌恶 [15]。后来，其他心理学家在这些基本情绪的基础上，又加入了兴奋、轻蔑、自豪、尴尬和羞耻。与之相对，情感描述的是由情绪形成的精神状态。这一定义强调，情感表达了一个人的想法，而这些想法会受到其情绪的影响。与情绪相比，情感的涵盖范围更广。情感具有高度的组织性，通常会将行动与某种情绪联系起来。

总而言之，如表 2.1 所示，有四个因素可以区分情绪与情感。此外，所有情绪都可以看作是这四个影响情感的变量综合作用的结果。因此，分析观点并识别情感至关重要。

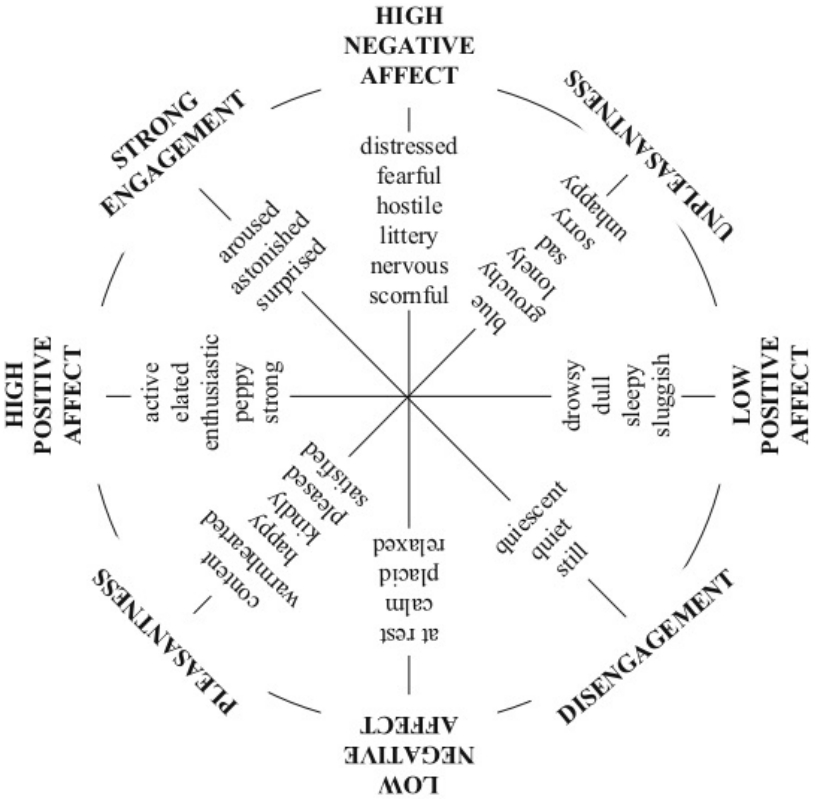

图 2.1 情绪的二维结构[16]

	定义	关联	维度	性质
情绪	心理状态	—	心理的	自然的
情感	精神状态	情绪表达	社会性	有组织的

表 2.1 情绪与情感的区别

[16]

2.2 情感分析

情感描述某人对某个主题所持的感受，可以是正面的，也可以是负面的。观点挖掘是情感分析的另一个名称。这是一种自然语言处理技术，研究人们对某个实体（如社会事件、产品或服务）的情感、观点和态度。文本挖掘是将无结构的原始文本转换为结构化数据，以提取有用信息的过程。因此，研究能够扫描文本并将其分类为不同类别的算法受到了广泛关注。

由于推特、Facebook 和 YouTube 等社交媒体平台的普及，包含个人观点和信息的评论数量大幅增加；因此，情感分析已成为学者和专业人士关注的热门趋势主题。情感分析帮助我们根据他人的观点对各种主题作出决策。许多公司利用情感分析来提供更好的产品和服务、开展营销活动以及分享促销优惠。

2.2.1 情感分析的层级

如图 2.2 所示，情感分析包含三个粒度层级：

- 文档级
- 句子级
- 实体与方面级

2.2.1.1 文档级情感分类

这一层级将整个文档视为一个整体，以判断其整体概念在此处是正面还是负面。这种方法假设一篇文档只讨论一个主题。例如，它会判断一部电影或一篇书评总体上是正面还是负面。

2.2.1.2 句子级情感分类

在此阶段，系统会将每个句子分开，并将其观点归类为正面、负面或中性。与文档级不同，每个文档对多个实体可能持有不同的观点。每个句子都会被归类为客观或主观类型 [18]。主观句子基于个人视角和观点，例如“她的头发很漂亮。”而客观句子不包含任何人的观点，而是以事实为依据，例如“I have black hair.”

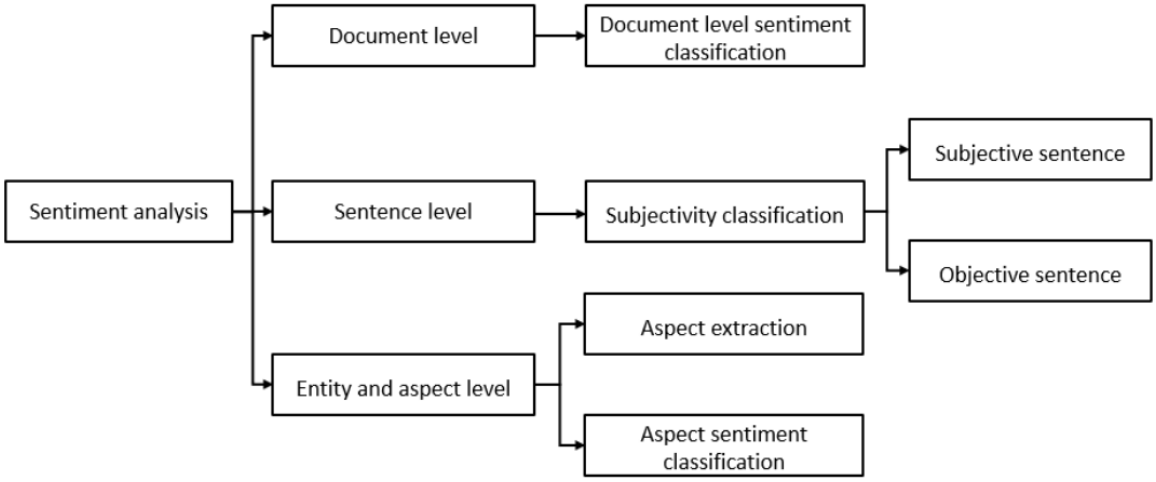

图 2.2 情感分析的不同层级[17]

2.2.1.3 实体和方面级情感分类

这一层级也称为基于特征的层级。与文档级和句子级不同，它直接处理观点。其目标是发现观点，并确定这些观点针对某个主题或实体所表达的情感，这些情感可能是正面、负面或中性的。

2.2.2 情感分类技术

我们之前已经了解过情感分析，并知道对实体相关观点进行分类称为情感分析。人们会采用不同策略来收集观点及其相关情感；有些策略相当复杂且耗时，有些则简单快捷。

如图 2.3 所示，我们将逐一探讨：

- 基于词典的方法
- 机器学习方法
- 混合方法

2.2.2.1 基于词典的方法

基于词典的技术是最早用于情感分析的技术之一。在基于词典的系统中，每个词都会被赋予一个极性分数；系统处理完文本后，

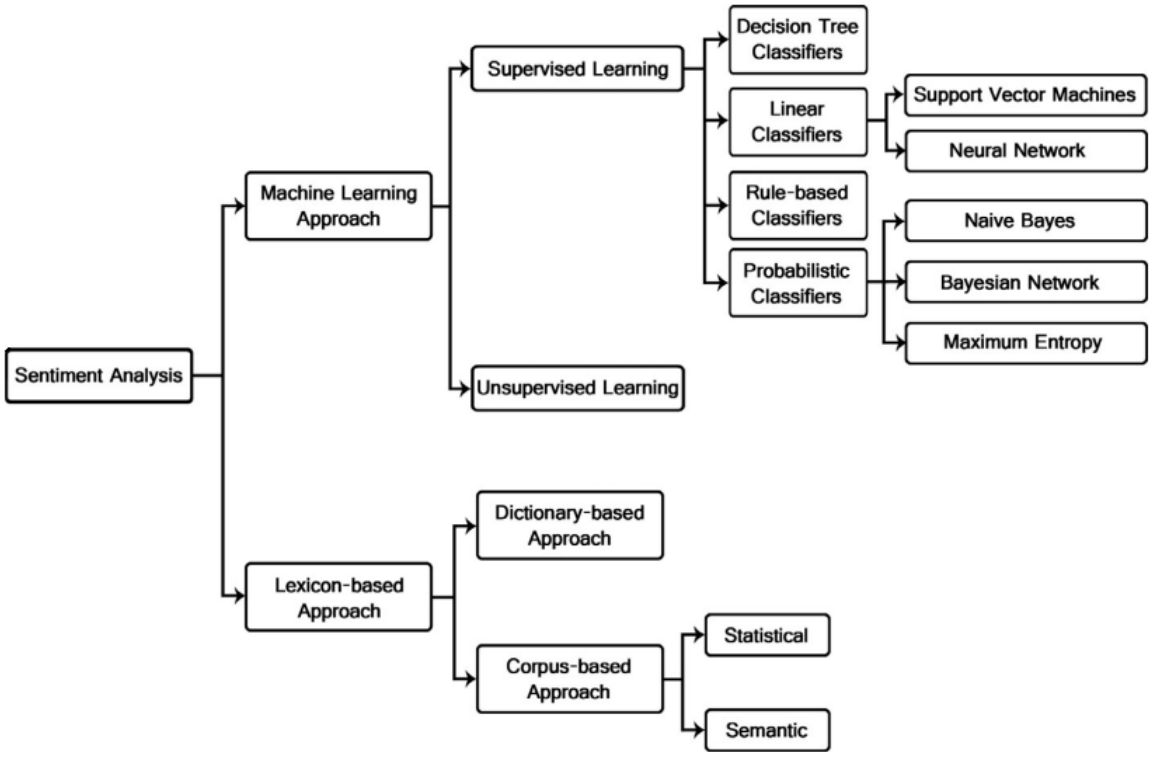

图2.3 不同的情感分类技术[19]

会为该文本给出一个情绪分数。这种方法的优点是无需数据自动化 [20]。

基于词典的技术分为两种子技术，即基于词典和基于语料库的方法。基于词典的方法使用类似于 WordNet[21]的术语词典来实现。另一方面，基于语料库的方法不需要预先定义的词典，而是采用 K近邻等统计分析方法。

2.2.2.2 机器学习方法

机器学习是人工智能的一个分支，它以数据作为输入来做出更好的判断，而不是被明确指示应该做什么。

情感分析中的机器学习方法 [22] 可以推导出观点及其对某个实体的情感。为此，人们采用了各种方法，包括支持向量机（SVM）和朴素贝叶斯。

通常，机器学习技术分为两类：

- 监督学习
- 无监督学习

监督学习 这项技术的目标是在不确定的情况下预测结果。

模型首先使用具有已知输出和标签的数据进行训练。然后，系统将从这些数据中学习，以做出最佳决策。

监督学习技术分为两种：

回归 目标是根据不同的独立输入预测一个实际值。例如，根据销售额确定预算。

分类 将各种项目归入不同类别。这是我们用于情感分析的方法。

有多种算法用于对对象进行分类。例如，支持向量机将每个对象分为两个不同的类别，这称为二分类。它试图找到一条分界线，将不同类别的成员分开。随后，它根据每条数据更新该直线的斜率和参数值，以找到最适合这种分类的直线。

无监督学习 无监督学习不同于监督学习，因为它不会基于预训练模型做出决策。相反，它通过在没有先验知识的情况下尝试提取数据之间的模式和相似性，分析原始数据以做出最优判断。

无监督学习技术也分为两个子类别：

聚类 该技术根据数据项之间的相似性和差异，将无标签数据划分为具有一定结构的组。聚类算法可能是层次式的、重叠式的或概率式的。

K均值聚类是聚类的一个示例，其中数据被分为 K 组，每组中距离某个质心最近的数据都归入该类别。通过调整 K，可以改变每个组的大小和详细程度，如图 2.4 所示。

关联 它是从大量数据中识别众多元素之间的模式和关联的过程。

2.2.2.3 混合方法

混合情感分析方法同时采用统计方法和基于知识的方法进行极性检测。因此，它兼具基于词典的方法和机器学习所带来的稳定性与出色的准确率[24]。

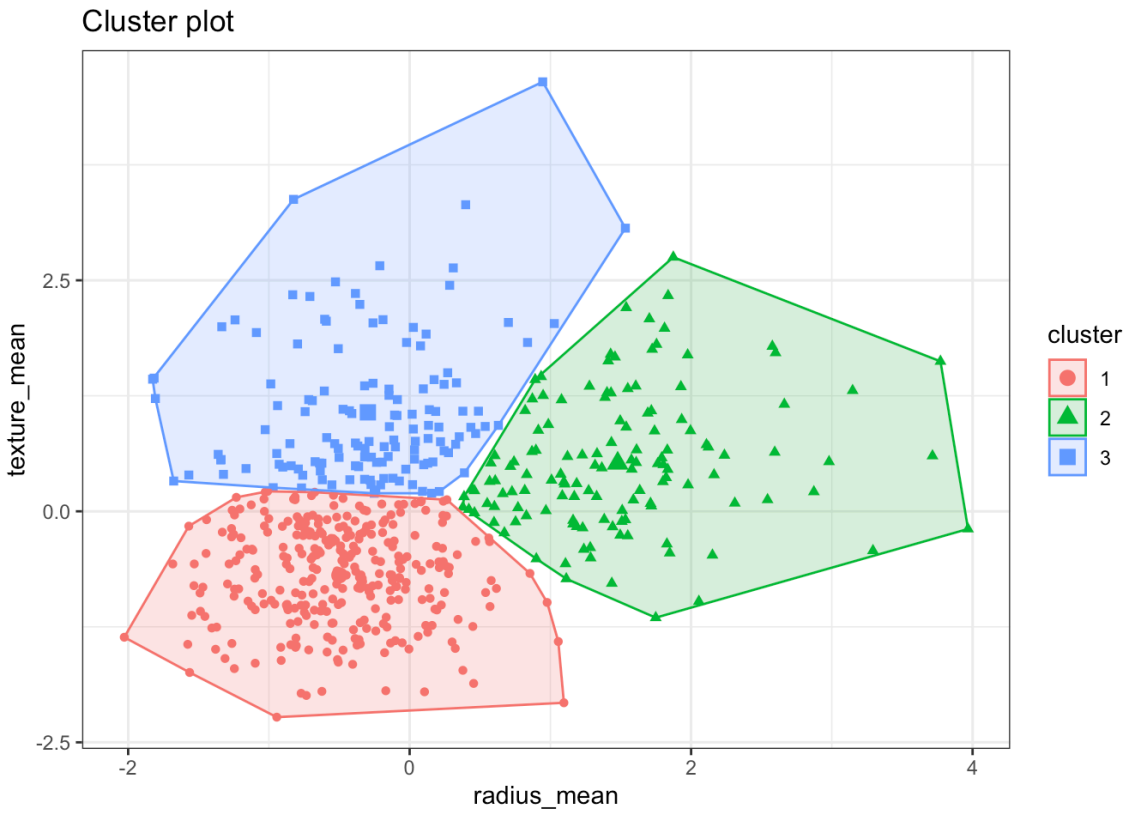

图2.4 K-means 倾向于形成大小相同的簇[23]

2.2.3 算法方法

人们提出了许多算法，用于提取有关实体或主题的情感。情感分析中最常用的算法包括朴素贝叶斯、随机森林和长短期记忆网络。下面我们将快速详细回顾这些算法。

2.2.3.1 朴素贝叶斯分类器

对于包含数百万条记录的大数据数据集而言，朴素贝叶斯是最简单、时间复杂度最低的监督算法之一。朴素贝叶斯认为，某个类别中一个字符的出现与另一特征的出现之间不存在相关性。因此，它被称为“朴素”算法。

朴素贝叶斯算法基于图2.6所示的贝叶斯定理，该定理使我们能够计算后验概率，例如，$P(H|E)$，使用 $P(E|H)$、$P(H)$ 和 $P(E)$，其中 $P(H|E) = P(e_1|H) * P(e_2|H) * ... * P(e_n|H) * P(H)$。

为了更好地理解朴素贝叶斯算法的工作原理，我们可以考虑这样一种情形：扫描每封电子邮件并提取其中的词语和

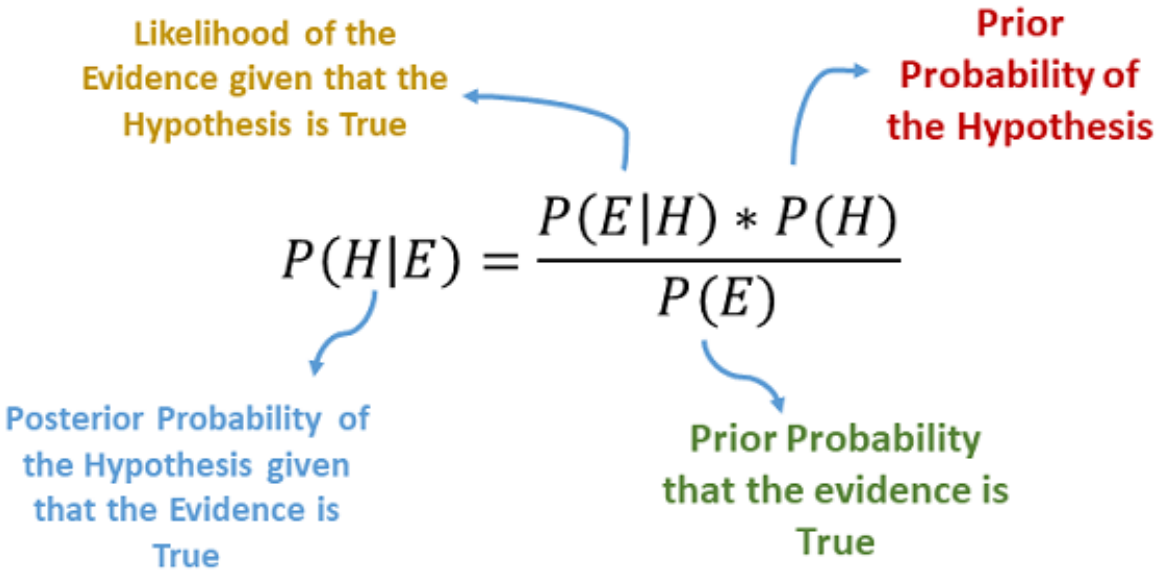

图 2.5 朴素贝叶斯分类器中的条件独立性假设[25]

短语，以区分垃圾邮件和正常邮件。为了判断一封电子邮件是否为垃圾邮件，我们创建一个表格，记录某个词出现在邮件中的似然。例如，已知这封电子邮件是垃圾邮件时，我们计算其中出现“offer”一词的似然。此外，我们还必须考虑这封电子邮件并非垃圾邮件的可能性。请记住，我们使用训练数据集计算了这些概率。

Th以下是朴素贝叶斯算法的优点和缺点 :

朴素贝叶斯分类器的优点

- 它的时间复杂度低于其他机器学习模型。
- 它在分类问题中表现更好。
- 它需要的训练数据较少。

朴素贝叶斯分类器的缺点

- 零频率问题会导致忽略在训练数据中出现次数为零且概率为零的类别的预测。
- 它是用于数值预测和估计的糟糕模型。
- 这种分类器往往会做出在现实中大多不成立的假设。

2.2.3.2 随机森林分类器

随机森林是另一种常见的分类算法。“随机森林”指的是一种构建大量随机决策树，并依据其中大多数树的预测来确定结果的系统。每棵决策树所使用的特征和数据都是随机选择的。

如图2.6所示，数据集被划分为若干彼此独立且完全随机的决策树；在尝试预测某个项目时，算法采用多数投票作为最终预测结果。此外，还可以通过计算每棵决策树预测结果的均值或平均值来解决回归问题。

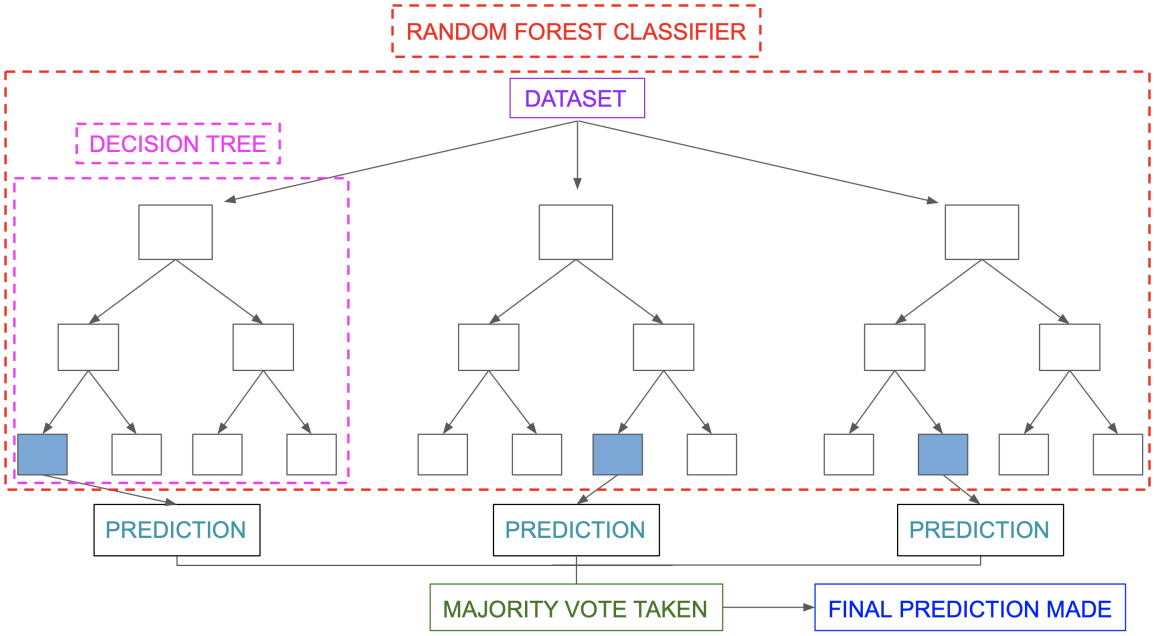

图2.6 随机森林分类器工作原理示例[26]

与其他算法一样，该算法也有若干优缺点：

随机森林分类器的优势

- 做出合理的预测
- 处理海量数据集
- 在预测准确率方面优于决策树
- 比其他算法更擅长处理缺失数据
- 在过拟合方面比决策树模型更不容易产生问题

随机森林分类器的缺点

- 时间复杂度高
- 最高价空间复杂度

2.2.3.3 长短期记忆模型（LSTM）

长短期记忆模型是一种循环神经网络，在处理大量序列数据方面非常有效。新型循环神经网络与传统循环神经网络的不同之处在于，它们可以利用先前输出的反馈来生成当前输出。LSTM 的主要特征是能够长时间存储信息，而这通过细胞状态实现。细胞状态的主要作用是高效存储过去的信息[27]。

在每次迭代中，LSTM 必须决定：

- 要向细胞状态添加哪些信息
- 从状态中删除无用信息。
- 根据输入生成输出，并将其报告给细胞。

LSTM模型在进行预测时不会受到异常值的影响，并且能够训练其参数。图2.7展示了该模型的一种简单机制。

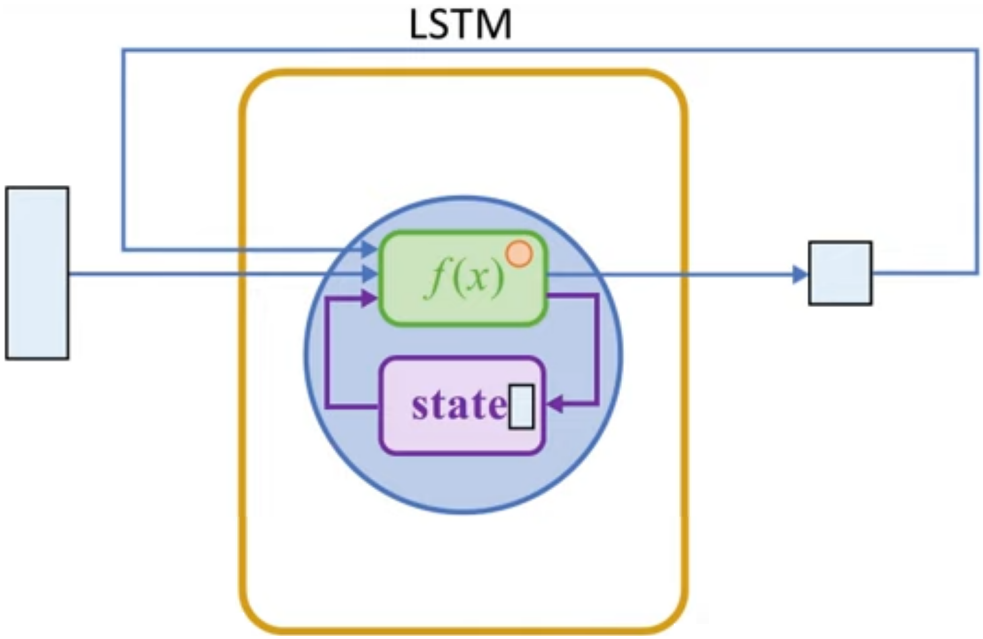

图2.7 长短期记忆网络模型的架构[28]

2.3 情感分析的应用

使用社交媒体的人数逐年增加。因此，数百万人通过推特等应用分享观点，包含人们对各种主题看法的数据量也在迅速增长。因此，利用情感分析提取观点及其极性已成为一项热门研究，并拥有众多应用。

以下是情感分析的一些应用：

股票市场 股票市场对每个国家的经济都至关重要。预测股票未来的表现有助于我们更好地进行资金投资。情感分析通过收集和处理不同的数据集，提取有关股票的观点，并尝试找出这些观点与股票之间的相关性。借助情感分析，我们可以在数据收集与准备之后尝试预测股票的走势。

电子商务 各种电子商务企业会利用不同类型的反馈，例如关于其产品的评论和留言，来提升产品质量和客户服务。推荐系统也会使用情感分析来分析人们对实体的感受，并改进推荐结果。例如，亚马逊曾

u使用情感分析来提升其所提供各项服务的质量 s.

因此，情感分析之所以广受欢迎，是因为这些应用展现出了浓厚的兴趣。

2.3.1 推特情感分析

推特是全球使用最广泛的社交媒体应用之一；自2006年推出以来，每月活跃用户已超过4亿。创建推特的主要目的是让用户能够通过推文表达对不同主题的观点，而推文是一种少于280个字符的文本。因此，推特用户的构成差异很大，从普通人到杰夫·贝索斯这样的极具影响力的人物，以及巴拉克·奥巴马这样的政治人物。

推特与其他社交媒体应用的主要区别在于，你在推特上发布的所有内容都向公众开放，即使他们没有推特账号也能查看；这一特征使推特上的情感分析非常有效。医疗保健和政治是推特情感分析的应用领域之一。

政治 随着推特的兴起，许多研究尝试用推特情感分析取代电话民意调查和网络民意调查等传统调查方法。研究表明，支持向量机分类器等算法有助于我们以很高的准确率预测民意调查结果。识别针对候选人的反讽是最具挑战性的部分。在这种情况下，正面情感会被视为负面。

商业 许多公司会利用推文直接或间接提取人们对其产品的情感和感受。这有助于公司改进产品并提供更好的服务。

2.4 数据收集与准备

数据收集与准备是从多个来源收集原始数据，并将其转换为结构化数据，以用于情感分析。数据准备的主要工作是将数据清理并统一为单一格式。

对于推特，数据准备分为五个步骤 [29–31]:

1. 移除图片和网址
2. 删除用户名

3. 移除特殊字符：此处必须移除标点符号。另一种需要移除并替换为空格的特殊字符是表情符号。尽管表情符号似乎有助于发现推文的情感，但它们过于复杂，难以解读并用于情感分析。

4. 删除不包含任何情绪的字母，例如“a”和“the”，这些字母也称为停用字符

5. 将所有内容转换为小写

2.4.1 推特数据收集方法

有多种方法可用于收集推文，供情感分析模型使用。本研究将探讨不同数据整理方法的优缺点，例如 Twitter API、Twint 网页爬虫、Tweepy 和预处理数据集。

2.4.1.1 Twitter API

我们可以使用官方 Twitter API 来收集数据。Twitter API 分为标准版、高级版和企业版 3 种订阅，其中标准版可免费使用。Twitter API 的主要限制是只能获取过去七天内的 3200 条推文，并且使用时还需要一个账户。为克服这些限制，我们可以订阅高级服务，但这些服务并非免费。这就是我们更倾向于使用 Twint 和 Tweepy 等库和工具的原因。

2.4.1.2 Tweepy

Tweepy 是一个用于收集推特数据的 Python 库。它使用 Twitter API 来收集和汇总推文。Tweepy 提供从入门级到高级的完整文档。此外，它允许用户发布推文和转推，甚至还可以为推文点赞。

尽管如此，它是一款付费应用，其免费版本也具有与 Twitter API 相同的限制。

2.4.1.3 Twint

Twint 使用网页抓取来收集推文，不依赖官方 Twitter API。借助 Twint，您可以不受限制地收集无限量的推文。不过，与推特近期的更新相比，Twint 的文档并不完整且已经过时，因此无法用于发布推文或点赞。

Twint 与 Tweepy 的主要区别如下：

• Tweepy 使用 Twitter API 收集推文，而 Twint 使用网络爬虫收集推文。

• Tweepy 在可收集的推文数量方面与 Twitter API 存在相同的限制，而 Twint 则没有收集推文数量的限制。

• Tweepy 的文档非常完善，因此更容易学习和使用。另一方面，Twint 的文档不够完善，因此学习和使用其功能可能会比较麻烦。

• Twint 仅限于收集推文和获取信息，而 Tweepy 还可用于发布推文和点赞。

2.4.1.4 预处理数据集

使用预处理数据集是 Twitter API 和其他第三方应用程序的替代方案。使用预处理数据集可以省去收集推文的麻烦，但灵活性不如推文收集工具。

例如，Kaggle 网站提供了许多可用于不同研究的预处理数据集。

我们为数据选择了这种方法，并使用了来自 Kaggle 网站的半预处理数据集。

第 3 章

方法论方法与技术框架

3.1 数据

在本节中，我们将说明所使用数据集的详细信息、这些特定数据集的用途、为使数据可供使用而执行的操作，以及数据标注流程。

3.1.1 Twitter 数据收集

我们使用了两个从推特收集的数据集。

第一个数据集由 160 万条带标签的记录组成，其中包含不一定与特定主题相关的推文。

我们从 deepnote.com 获取了这个数据集，用于训练自然语言处理模型。其中一半记录标记为 0，另一半标记为 4。这些标签表示推文的情感倾向。标签 0 表示负面情绪，标签 4 表示正面态度。

在构建并测试分类器模型后，我们使用这些模型为来自 Kaggle 网站的第二个数据集添加标签。该数据集包含约 90,000 条与股票市场相关的推文记录，以及三列信息：ID、创建日期和推文全文。这些数据是在三个月内收集的，时间为 2020 年4月9日至7月16日。

该数据集中的推文包含 13000 多只股票的名称。我们仅提取了与五个股票市场相关的数据：Facebook（最近已更名为 Meta）、Microsoft、特斯拉、Apple 和亚马逊。它们是最受欢迎且交易最活跃的股票。

```
top5_dataframe_noLable = data[data['full_text'].str.contains('|'.join(
  top5_tickers))]
```

清单 1 提取前 5 只股票的数据

随后，我们对数据进行了预处理，并为每只股票分别创建了一个数据框：

```
AAPL_DF = stockDF[stockDF['full_text'].str.contains('aapl')]
AMZN_DF = stockDF[stockDF['full_text'].str.contains('amzn')]
FB_DF = stockDF[stockDF['full_text'].str.contains('fb')]
MSFT_DF = stockDF[stockDF['full_text'].str.contains('msft')]
TSLA_DF = stockDF[stockDF['full_text'].str.contains('tsla')]
```

清单 2 分别提取每只股票的数据

3.1.2 金融数据收集

我们通过雅虎财经 API 收集了五只股票的市场数据：Meta、Microsoft、特斯拉、Apple 和亚马逊。

雅虎财经 API 是一个免费的 API，提供了大量关于不同市场的信息和新闻。从中提取数据也非常简单。

```
import yfinance as yf
start = AAPL_DF['Date'].iloc[0]
end = AAPL_DF['Date'].iloc[-1]
apple_data = yf.download("AAPL", start=start, end=end)
meta_data = yf.download("META", start=start, end=end)
micro_data = yf.download("MSFT", start=start, end=end)
tesla_data = yf.download("TSLA", start=start, end=end)
amazon_data = yf.download("AMZN", start=start, end=end)
```

列表 3 从雅虎财经下载数据

这些数据包含六个属性：开盘价、最高价、最低价、收盘价、调整后收盘价和成交量。我们只使用收盘价，分别用于计算 *Return* 的数量并检查其与推特数据的相关性。

3.1.3 数据预处理

数据需要进行预处理：既要在用于构建监督学习模型之前进行预处理，也要在由该模型进行标注之前进行预处理。

我们将说明对两个推特数据集进行预处理的具体方法。

文本预处理的首要重要前提，是清除推文文本中的代词、静态动词、网址、表情符号、用户名、数字和标点符号。我们将代词和静态动词放入一个名为 停用词列表 的列表中，并据此清理了推文文本：

```
STOPWORDS = set(stopwordlist)
def cleaning_stopwords(text):
return " ".join([word for word in str(text).split() if word not in
 STOPWORDS])
```

清单 4：清理数据中的冗余词语

我们移除了标点符号：

```
english_punctuations = string.punctuation
punctuations_list = english_punctuations
def cleaning_punctuations(text):
translator = str.maketrans('', '', punctuations_list)
return text.translate(translator)
```

清单 5：移除标点符号

移除重复字符，例如将 *hiiiii* 转换为 *hi*：

```
def cleaning_repeating_char(text):
return re.sub(r'(.)1+', r'1', text)
```

代码清单 6：清理重复字符

已移除网址和数字：

```
def cleaning_URLs(data):
return re.sub('((www.[^s]+)|(https?://[^s]+))',' ',data)
def cleaning_numbers(data):
return re.sub(r'[\W_]+',' ',data)
```

下一步是对文本中的词语进行分词，以便更轻松地检查和处理。我们将每条推文的文本转换成了词向量。

```
tokenizer = RegexpTokenizer(r'\w+')
```

我们还会对词语进行词干提取。这意味着将词根相同的不同词语转换为同一个词根。例如，payment、pay 和 paying 这些词的词根都是 pay。使用词干词而不是这些不同形式，有助于简化处理和学习过程，同时保留主要含义和上下文。

```
import nltk
st = nltk.PorterStemmer()
def stemming_on_text(data):
text = [st.stem(word) for word in data]
```

```
5   return data
```

清单 7 词干提取

另一项必要步骤是对词语进行词形还原，也就是说，使用 NLTK 内置函数 *WordNet 词形还原器()*，统一同一个词的不同形式。

```
1   nltk.download('wordnet')
    lm = nltk.WordNetLemmatizer()
3   def lemmatizer_on_text(data):
    text = [lm.lemmatize(word) for word in data]
5   return data
```

清单 8 词形还原

我们使用以下语法，通过 Lambda 表达式调用所有已创建的函数：

```
data = data.apply(lambda x: function(x))
```

清单 9 应用预处理函数

此外，我们还对每个词进行了独热编码，将每个词转换为向量，以供后续处理。

独热编码（图 3.1）用一个包含 N 个值的向量表示每个单词，其中 N 是数据库中的单词总数。该向量中的所有值均为 0，只有索引对应目标单词的值例外。由于数据库中的单词数量极其庞大，我们省略了出现频率最低的单词，只保留出现频率最高的单词，以便对大型数据库进行处理。

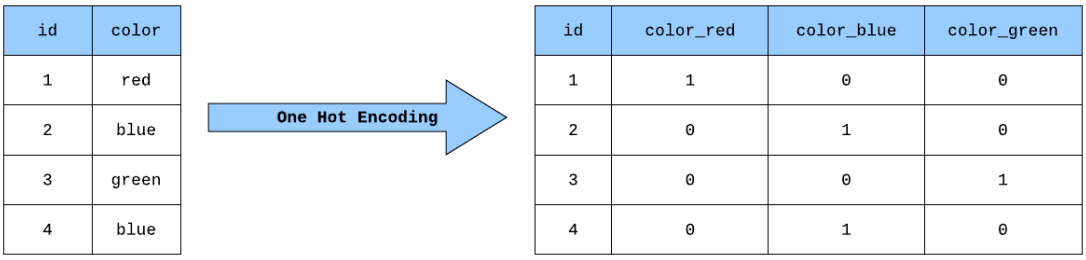

图 3.1 独热编码示例[32]

预处理完成后，我们就可以将模型拟合到数据中了。为此，我们将数据划分为训练集和测试集，并构建了分类器模型。

3.1.4 标注

我们使用基于第一个数据集构建的分类器模型，为从第二个数据集中提取的数据添加标签。如前所述，训练数据的标签为 0 和 4。为简化起见，我们将标签 4 替换为 1。

使用随机森林模型进行标注：

```
rfPred = np.where(rfPred==4,1,rfPred)
stockDF = dup_top5_noLable.copy()
stockDF['Label_rf']=rfPred
```

清单 10 使用随机森林模型进行标注

使用朴素贝叶斯模型进行标注：

```
nbPred = BNBmodel.predict(X_test)
nbPred = np.where(nbPred==4,1,nbPred)
stockDF['Label_nb']=nbPred
```

清单 11 使用朴素贝叶斯模型进行标注

使用长短期记忆网络算法进行标注：

```
test_sequences = tokenizer.texts_to_sequences(dup_top5_noLable['
 full_text'])
test_padded = pad_sequences(test_sequences, padding='post', maxlen=
 max_length)
pred_labels = []
prediction = model.predict(test_padded)
for i in prediction:
if i >= 0.5:
pred_labels.append(1)
else:
pred_labels.append(0)
pred_labels
stockDF['Label_lstm'] = pred_labels
```

清单 12 使用LSTM模型进行标注

3.2 算法实现

本节将详细介绍分类器模型的构建，包括代码和参数、超参数调优时的考虑因素、所实现模型的结构、准确率的检查，以及理解代码所需的相关概念。

3.2.1 随机森林分类器

我们使用了 *sklearn* 中的随机森林分类器模型。我们想说明传递给该分类器模型的两个参数：估计器的数量和最大深度。

估计器的数量表示我们希望生成的决策树数量。树的数量越多，预测越准确，但会增加计算量，进而减慢处理速度。因此，该参数的最佳取值是处理器所能承受的最大值。

最大深度表示每棵树的最大深度。如果不设置最大深度，算法就会持续生成叶节点，直到样本耗尽，或所有叶节点都变为纯节点。

还有其他一些关键参数也可以调整，例如最大_特征 和最小_样本_叶节点，它们与*n*_估计器一样，会同时影响性能和速度。还有一些参数可以直接影响训练速度，包括*n*_作业、随机_状态以及袋外_分数。

```
model = RandomForestClassifier(n_estimators=200,max_depth=60)
model.fit(X_train,y_train)
predictions = model.predict(X_test)
```

图表 13 随机森林模型的构建

我们进一步使用混淆矩阵来观察结果。混淆矩阵是一个包含四个单元格的表格：真正例（TP）、真负例（TN）、假正例（FP）和假负例（FN）。

- TP 表示被准确预测为正面的值的数量。
- TN 表示被预测为负面且实际也是负面的值的数量。
- FP 统计被预测为正面的负面值的数量。
- FN 表示被错误预测为负面的正面值的数量。

```
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test,predictions)
```

清单 14：混淆矩阵

并检查了准确率得分，该得分表示模型准确预测标签的记录所占的百分比。

```
from sklearn.metrics import accuracy_score
accuracy_score(y_test,predictions)*100
```

清单15 随机森林模型的预测准确率

3.2.2 朴素贝叶斯（伯努利）分类器

只需调用伯努利模型内置的 fit 函数，使用训练数据和测试数据进行拟合、预测并检查准确率，即可轻松完成实现。

该算法最关键的优化参数是*alpha*、拟合_先验和类别_先验。

```
from sklearn.Naïve_bayes import BernoulliNB
BNBmodel = BernoulliNB()
BNBmodel.fit(X_train, y_train)
y_pred1 = BNBmodel.predict(X_test)
accuracy_score(y_test,y_pred1)*100
```

清单16 构建朴素贝叶斯模型

3.2.3 长短期记忆网络模型（长短期记忆网络）

我们构建了一个由 4 层组成的顺序模型，输入端有 200 个特征，输出端为 1。第一层是一个神经网络，包含 200 个起始节点，这些节点的值会传递到中间节点，进而得到 100 个末端节点的值，并将其作为输出返回。显然，我们需要模型只返回一个值，也就是预测值。

嵌入层的工作方式类似于查找表，可将每个单词嵌入数据集中所有单词构成的向量中。它为每个单词指定一个整数值，并将其映射到稠密向量。嵌入层有助于将高维数据嵌入到较低维的空间中。

双向层由两个循环神经网络组合而成：一个从数据序列的开头开始处理数据，另一个则沿相反方向处理数据。这种遍历方式的必要性在文本处理中显而易见：一个单词的含义不仅取决于它前面的单词，也取决于它后面的单词。

我们在全连接层中使用激活函数。

```
import keras
model = keras.Sequential([

```

```
  keras.layers.Bidirectional(keras.layers.LSTM(64)),
5
7
```

图表 17 顺序长短期记忆层

激活函数对于深度模型正确学习是必不可少的，因为它们会限制神经元值，防止其变得过大或过小而失去作用。常见的激活函数包括 ReLU、Sigmoid 和正切函数。

ReLU 函数被认为是层间使用最可靠的激活函数。除非输入为负数，否则它会返回输入值本身；若输入为负数，则返回零。Sigmoid 和正切函数用于分类问题。

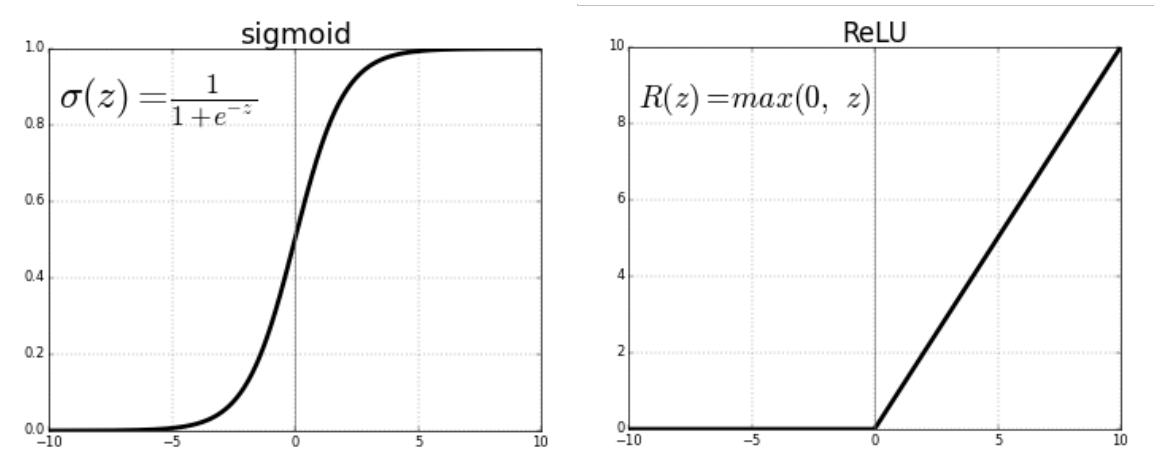

图 3.2 Sigmoid 和 ReLU 函数[33]

为了继续讲解代码，我们需要定义损失函数和优化器。

损失函数将模型的预测输出与期望值（即实际值）进行比较，并为每个观测值返回一个数值，即损失。损失越小，模型训练效果越好。

最常用的损失函数包括均方误差（MSE）、分类交叉熵和二元交叉熵。它们分别适用于不同类型的问题。

均方误差最适用于回归问题，而交叉熵最适用于分类问题。由于我们预期输入只有两个取值，因此使用二元交叉熵。

优化器是利用损失函数的梯度更新神经元权重的算法。GSD 和 Adam 是两种最稳定的优化器。优化器唯一

可能需要调整的参数是学习率。如果学习率过低，算法会以极小的步长搜索损失函数的极小值，导致学习过程极其缓慢。反之，如果学习率过高，算法起初可能学习得很快，但由于搜索时采用的步长过大，可能无法找到极小值，从而导致训练过程不稳定。

```
model.compile(loss='binary_crossentropy',
```

代码清单 18 编译 LSTM 模型

批量大小指定了每个训练步骤中传递给网络的观测值数量。训练轮数表示向神经网络呈现完整训练数据的次数。

```
num_epochs = 2
```

代码清单 19 将 LSTM 模型拟合到数据

设置 *verbose* 属性后，运行时会显示进度条，而*validation_split* 表示用于验证的数据量占训练数据总量的比例。

最后，我们测试了模型的准确率，并将标签 4 替换为 1。

```
prediction = model.predict(test_padded)
```

清单 20 检查 LSTM模型的准确率

第4章

性能评估

4.1 性能评估的图解结构

我们定义并计算了一个变量看涨情绪（B_t），该变量根据推特数据特征 [34],提取，计算如下:

$$B_t = \ln(\frac{1 + M_t^{Positive}}{1 + M_t^{Negative}})$$

```
def bullishness(data):
newData = pd.DataFrame(columns=['Date','lstm','rf','nb'])
newData.Date = data.created_at.unique()
grouped = data.groupby(data.created_at)
for date in newData.Date:
dfTemp = grouped.get_group(date)
lstmPos = (dfTemp.Label_lstm==1).sum()
lstmNeg = (dfTemp.Label_lstm==0).sum()
newData.loc[newData.Date == dfTemp.created_at.values[0],"lstm"]=math.
 log((1+lstmPos)/(1+lstmNeg))
rfPos = (dfTemp.Label_rf==1).sum()
rfNeg = (dfTemp.Label_rf==0).sum()
newData.loc[newData.Date == dfTemp.created_at.values[0],"rf"]=math.log
 ((1+rfPos)/(1+rfNeg))
nbPos = (dfTemp.Label_nb==1).sum()
nbNeg = (dfTemp.Label_nb==0).sum()
newData.loc[newData.Date == dfTemp.created_at.values[0],"nb"]=math.log
 ((1+nbPos)/(1+nbNeg))
return newData
```

代码清单21 看涨情绪的计算

并从金融数据的特征中提取了一个变量 *Return Close* 。

股票的开盘价是指股票在一天开始时的交易价格，而收盘价是指当天所有交易完成后达成的股票价格，或者简单来说，就是该股票当天最后一笔交易的价格。

为了了解某只股票的价值相较前一天在一天内上涨或下跌了多少，我们计算每天的一个变量 收益率 。 t [34]。

$$Return_t = (\ln Close_t - \ln Close_{t-1}) \times 100$$

```
def calc_Return(data):
newData = pd.DataFrame(columns=['Date','Values'])
for i in range(1,len(data.index)):
newData.loc[i-1] = [data.index[i],(math.log(data.Close.iloc[i])-math.
 log(data.Close.iloc[i-1]))*100]
return newData
```

清单 22 收益值的计算

最后，我们分析了金融特征“收益率”与推特数据特征“看涨情绪”之间的相关性。

4.2 结果的详细评估

本节将展示分析，并探寻金融数据与推特数据之间的相关性。

4.2.1 苹果股票（$AAPL）的评估

我们将三个月期间计算得出的看涨情绪与各分类模型的输出之间的差异进行了可视化。

```
AAPL_B = bullishness(AAPL_DF)
import matplotlib.pyplot as plt
plt.plot(AAPL_B.Date,AAPL_B.lstm)
plt.plot(AAPL_B.Date,AAPL_B.nb)
plt.plot(AAPL_B.Date,AAPL_B.rf)
plt.legend(["LSTM","Naïve base","Random forest"])
```

列表 23：绘制某只股票的三项预测



图4.1 AAPL 看涨情绪图

如图4.1所示，朴素贝叶斯和长短期记忆网络模型的结果相当接近，但与随机森林模型预测的看涨情绪相差甚远。仔细观察可以发现，在大多数时间间隔内，所有分类器都呈现出相似的模式。

以下是根据从雅虎财经统计数据中提取的数据计算出的收益率图示（图形结果见图**??**）：

```
AAPL_R = calc_Return(apple_data)
plt.plot(AAPL_R.Date,AAPL_R.Values)
plt.show
```

代码清单24 单只股票收益率的绘图

Then w我们使用 *FinRes* 将金融和推特的结果合并到一个数据框中：

```
import timeimport datetimedef finalRes(return_data,bullishness_data):
```

```
    break
fin = pd.DataFrame(l,columns=['Date','lstm','rf','nb','Values'])
return fin
res = finalRes(AAPL_R,AAPL_B)
```

代码清单 25 将预测和收益值合并到一个数据集中

我们进一步使用合并后的数据，通过可视化将每个分类器输出与实际市场事件进行比较。

```
def drawAllForStock(data,str):
bull = bullishness(data)
start = bull['Date'].iloc[0]
end = bull['Date'].iloc[-1]
stock_data = yf.download(str, start=start, end=end)
stock_data.drop(apple_data.columns.difference(['Close']), 1, inplace=
 True)
ret = calc_Return(stock_data)
res = finalRes(ret,bull)
drawPlot(res.Date,res.lstm,res.Values,"lstm model")
drawPlot(res.Date,res.rf,res.Values,"Random forest model")
drawPlot(res.Date,res.nb,res.Values,"Naïve Bayes model")
```

代码清单 26 绘制收益率和所有预测的比较图

并在一个代码块中绘制并保存了这些图:

```
drawAllForStock(AAPL_DF,'AAPL')
figs = [plt.figure(n) for n in plt.get_fignums()]
i=0
for fig in figs:
fig.savefig(f"apple_RB{i}.png", dpi=300)
i = i+1
```

代码清单 27 绘制并保存开放图形

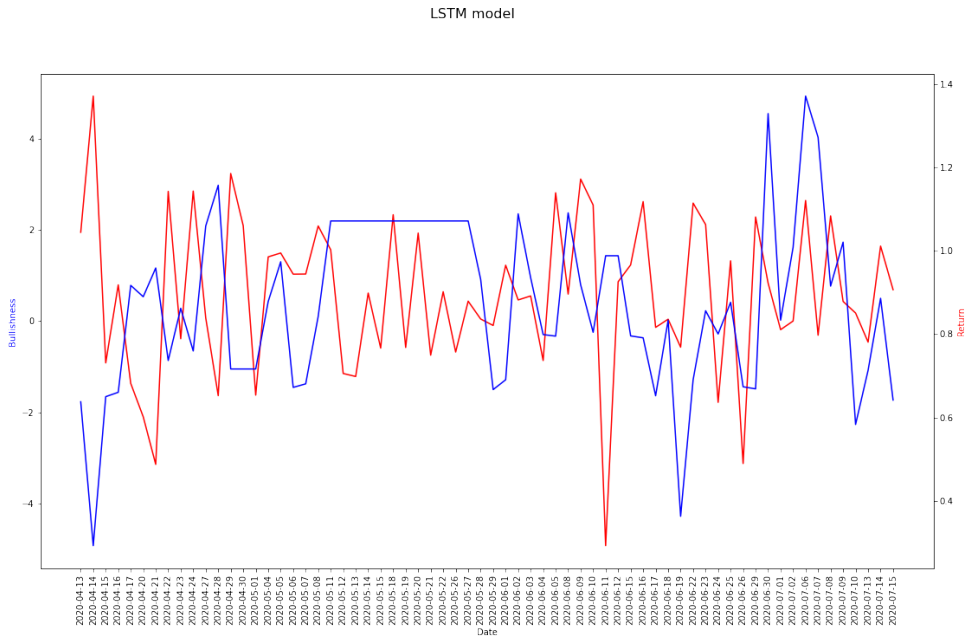

图4.2 AAPL的LSTM模型图

从图4.2可以看出，大多数日子里，收益率的正负与市场看涨情绪的预测相符；有些日子的预测也相当接近，例如6月25日至7月2日。这之后，预测值与实际收益率之间的差距是完全可以理解的。由于股票一直在快速上涨，人们会预期它会继续保持这一势头，因此出现了许多正面评论。然而，收益率却意外下跌了。这一差距之后，直至7月15日，预测也相当接近。

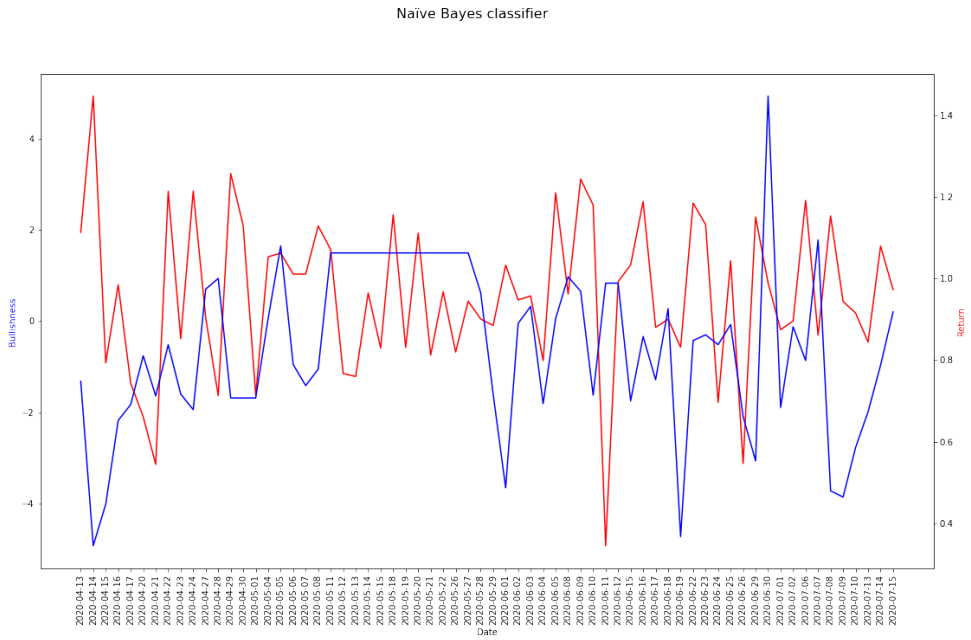

图 4.3 AAPL 朴素贝叶斯分类器图

图4.3显示，朴素贝叶斯分类器可能做出了最差的预测。尽管该模型的预测与朴素贝叶斯的预测相当相似，但该分类器的同期差距更大。值得注意的是，有些日子的收益率与对次日的预测非常接近，这意味着人们的积极情绪与当日收益率之间的关系，比与次日收益率之间的关系更为密切。



图4.4 AAPL 随机森林分类器图

随机森林的预测最令人满意，尤其是从6月18日起。（图4.4）

4.2.2 Facebook股票（$脸书）的评估

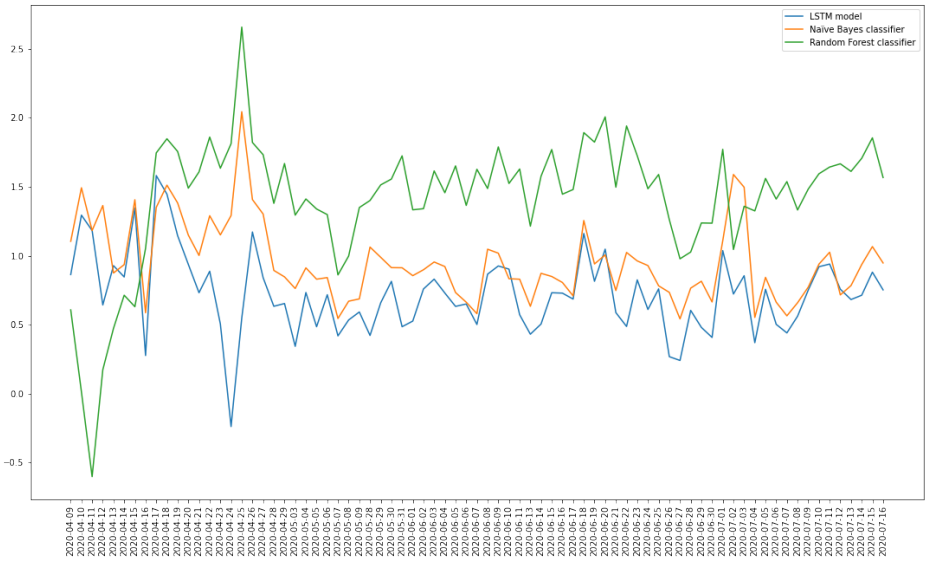

图 4.5 脸书看涨情绪图

如图4.5所示，与苹果股票类似，朴素贝叶斯和长短期记忆网络的输出相比随机森林更加相似。我们可以观察到这些模式之间存在一些相似之处。

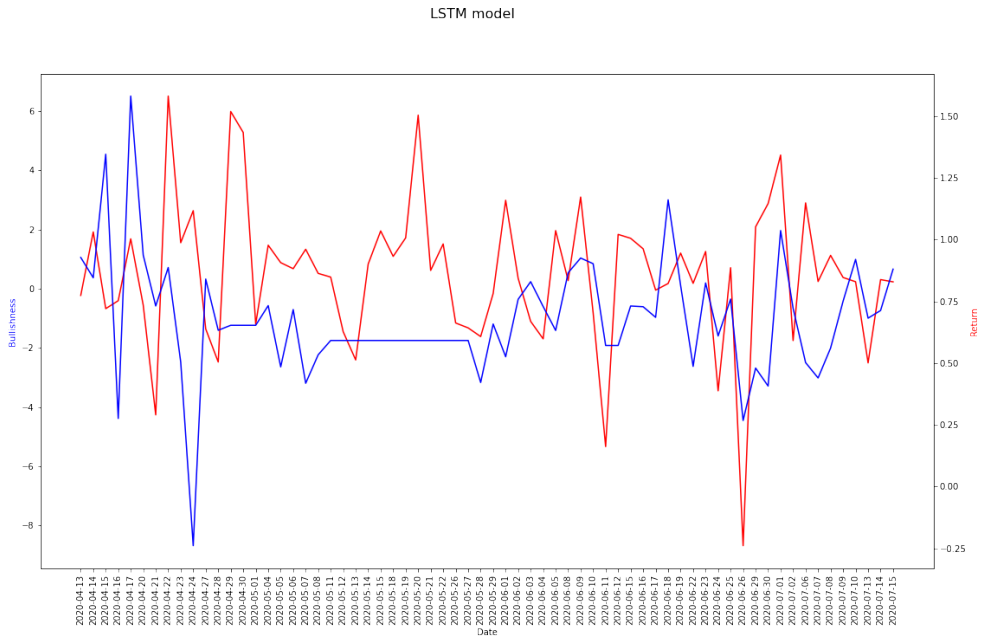

图4.6 脸书的LSTM模型图

在图4.6中，我们可以看到有些幸运的日子，预测结果较为接近，但在大多数情况下，预测结果相当不可靠。最接近的时间区间与Apple的类似，为6月23日至29日。

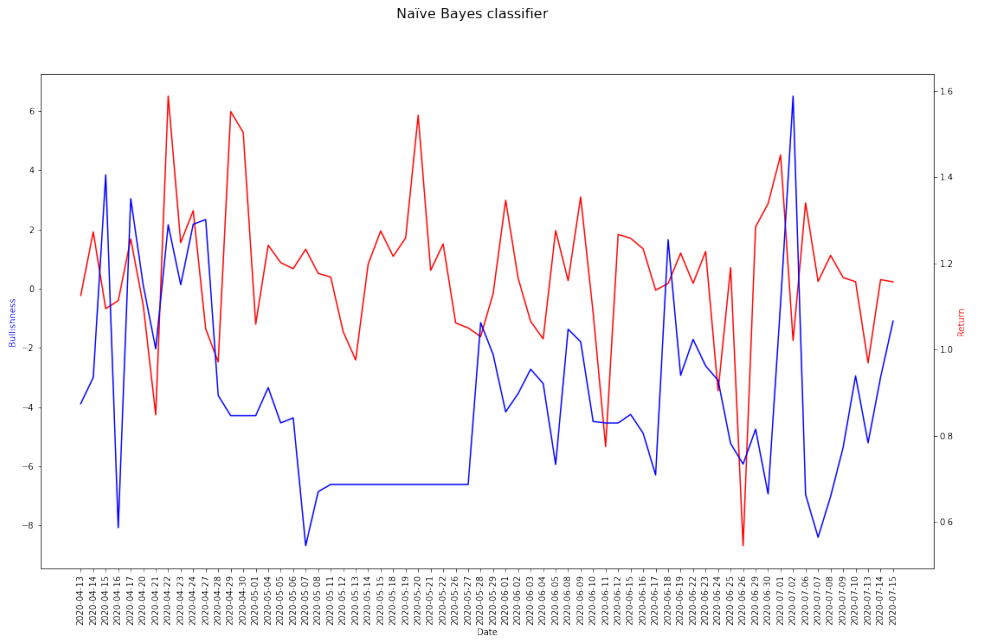

图4.7 脸书的朴素贝叶斯分类器图

朴素贝叶斯的预测（图4.7）比长短期记忆网络的预测略为接近。结果在5月1日至5日以及6月24日至7月7日期间最为令人满意。

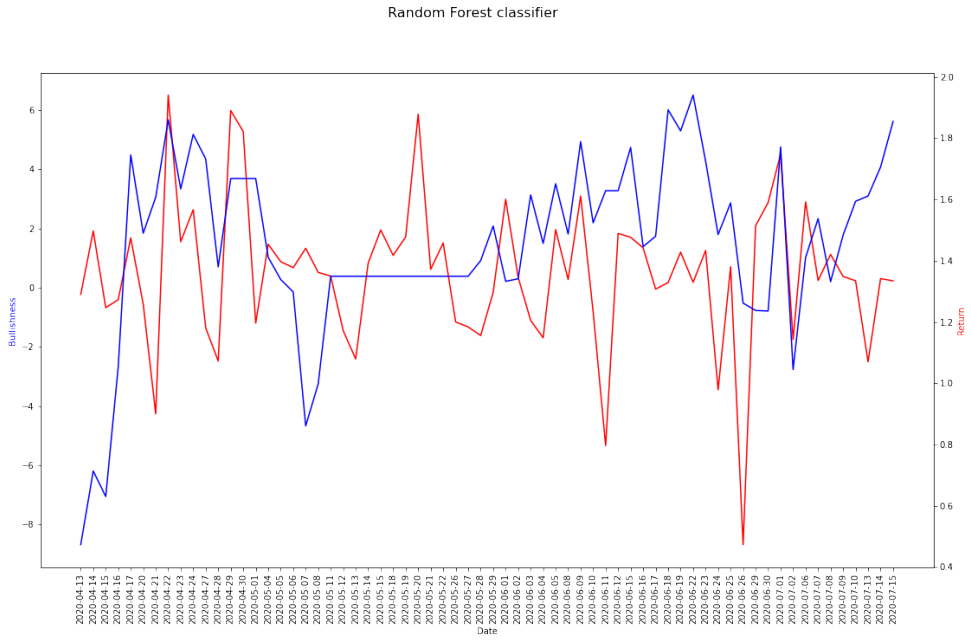

图4.8 脸书随机森林分类器图

如图 4.8 所示，随机森林的许多预测值与实际股票收益率相差甚远，只有少数预测值较为接近，因此难以评估其表现。最接近实际值的预测出现在 7 月初的几天里；在这些日子里，其他分类器的预测值（甚至针对其他股票的预测值）也都比较接近。

4.2.3 Microsoft股票（$MSFT）的评估

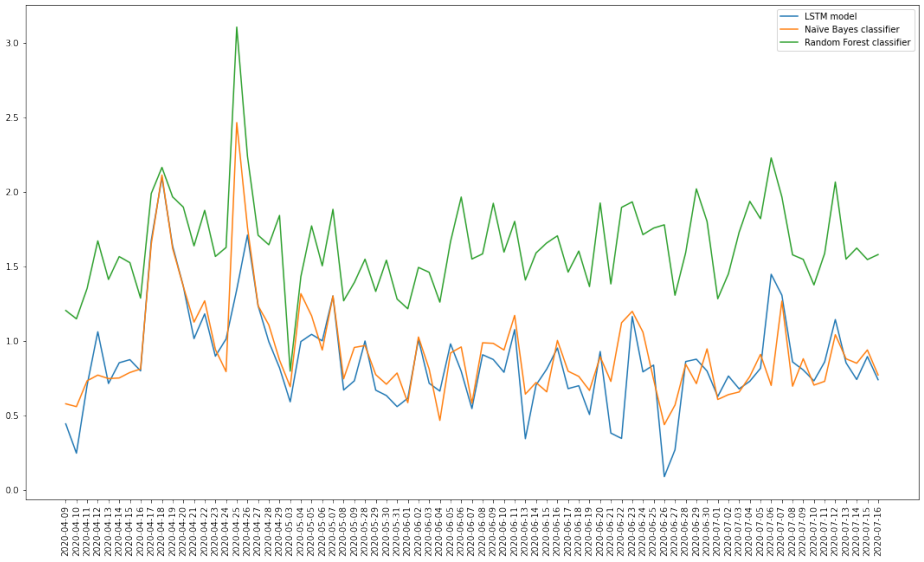

图 4.9 MSFT看涨情绪图

与随机森林相比，朴素贝叶斯和长短期记忆网络的输出更为相似（图 4.9）。三种分类器的模式之间都能观察到许多相似之处，而且它们对峰值点的判断一致。

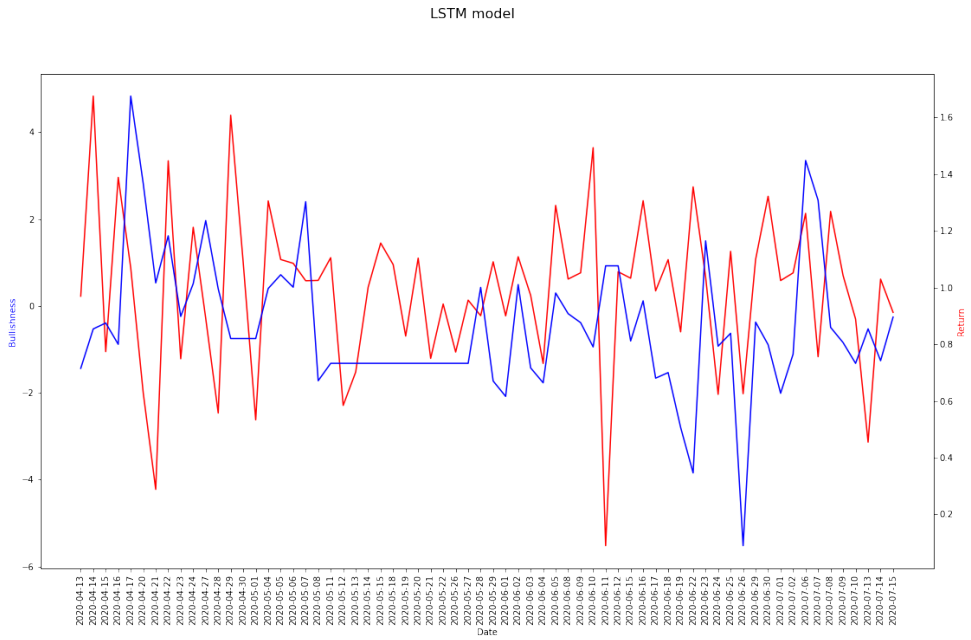

图4.10 MSFT的LSTM模型图

图4.10显示，LSTM模型对某些模式的预测较为准确。然而，有些预测值与实际值相差甚远。

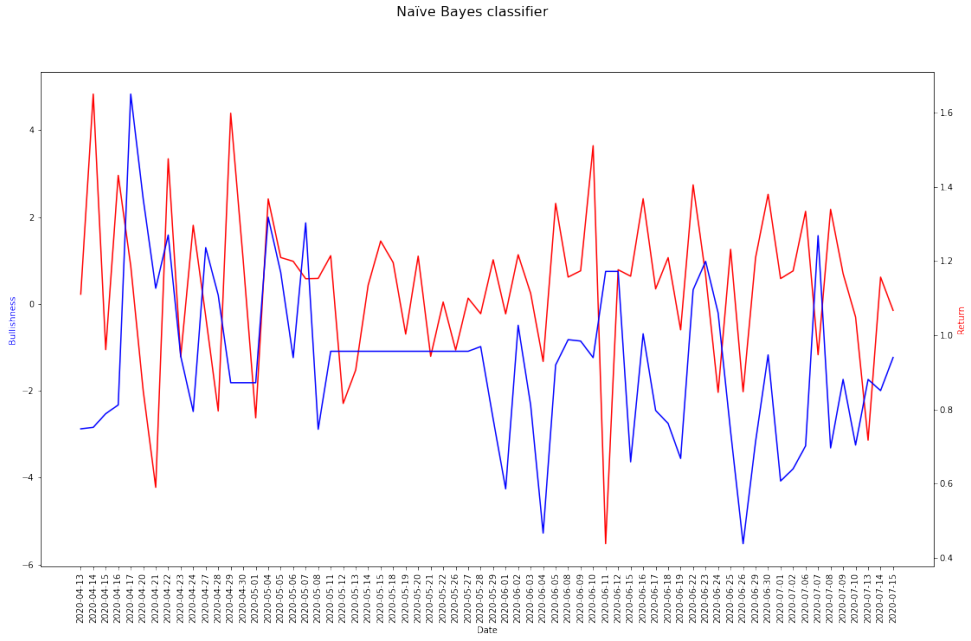

图4.11 MSFT的朴素贝叶斯分类器图

如图4.11所示，朴素贝叶斯分类器的大多数预测

都是错误的，即使是对收益率积极或消极情绪的判断也是如此。

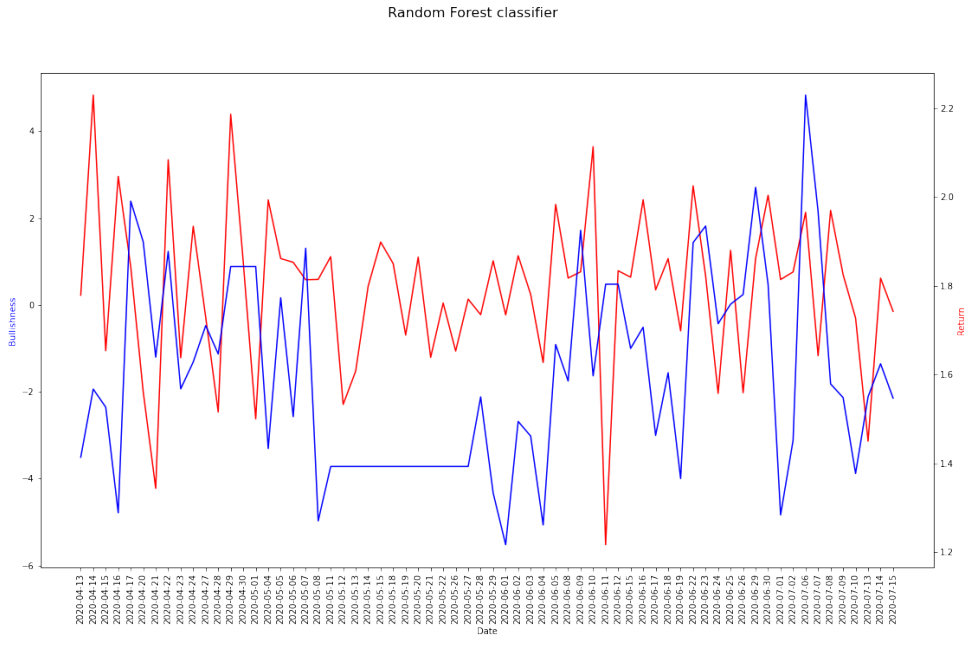

图4.12 MSFT随机森林分类器的图示

随机森林分类器的预测结果不太令人满意（图4.12），但比另外两种更接近。

4.2.4 特斯拉股票（$TSLA）的评估

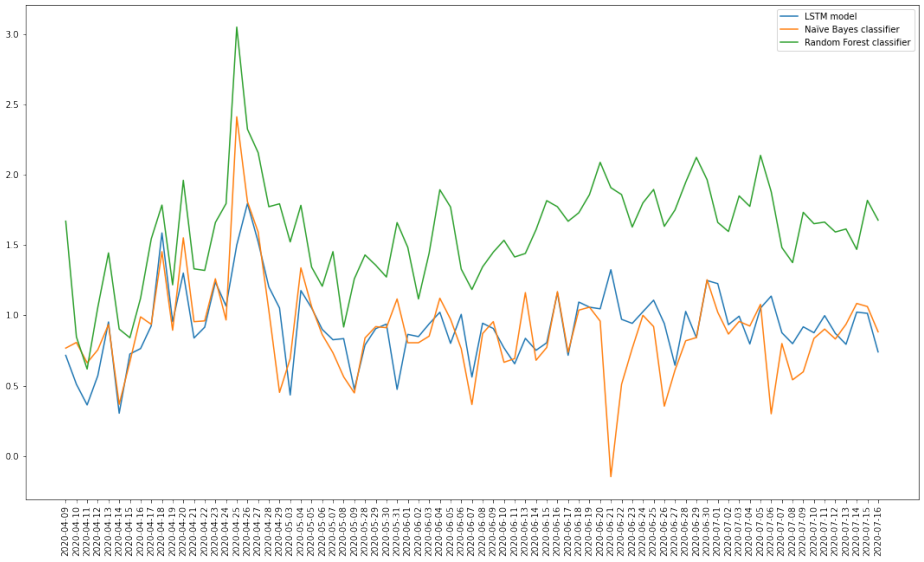

图4.13 TSLA看涨情绪图

从图4.13可以看出，与随机森林相比，朴素贝叶斯和长短期记忆网络的输出更为相似。三者的模式中都可以观察到许多相似之处。所有分类器对峰值点的判断一致。

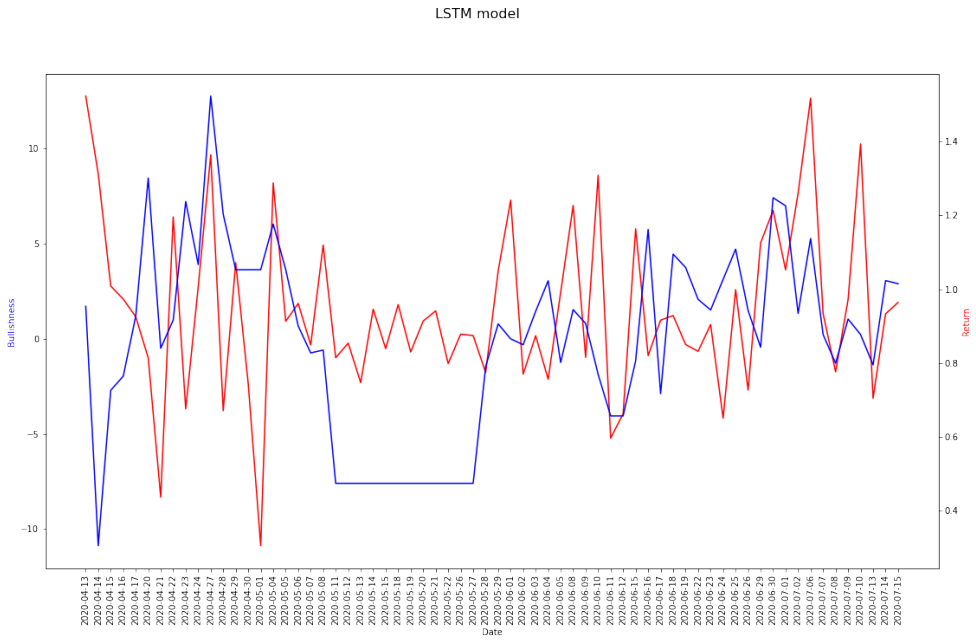

图4.14 TSLA 的 LSTM模型图

长短期记忆网络模型中有少数几处模式的预测较为准确（图4.14）。然而，大多数预测值与实际值相差甚远。

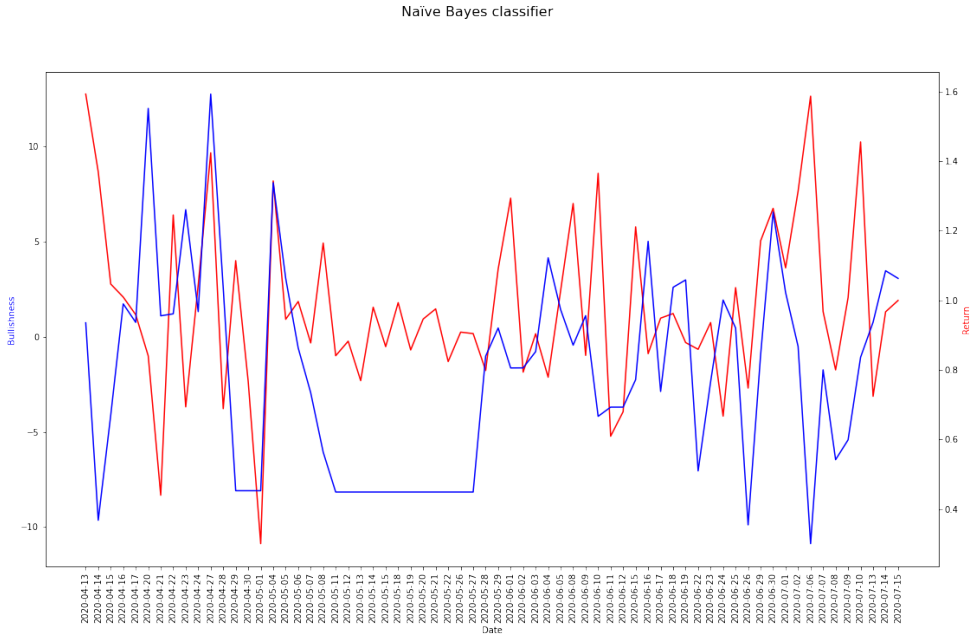

图4.15 TSLA 的朴素贝叶斯分类器图

通过分析（图4.15），我们可以看出，朴素贝叶斯的表现优于

长短期记忆网络；在这种情况下，大多数峰值和谷值都得到了准确预测。

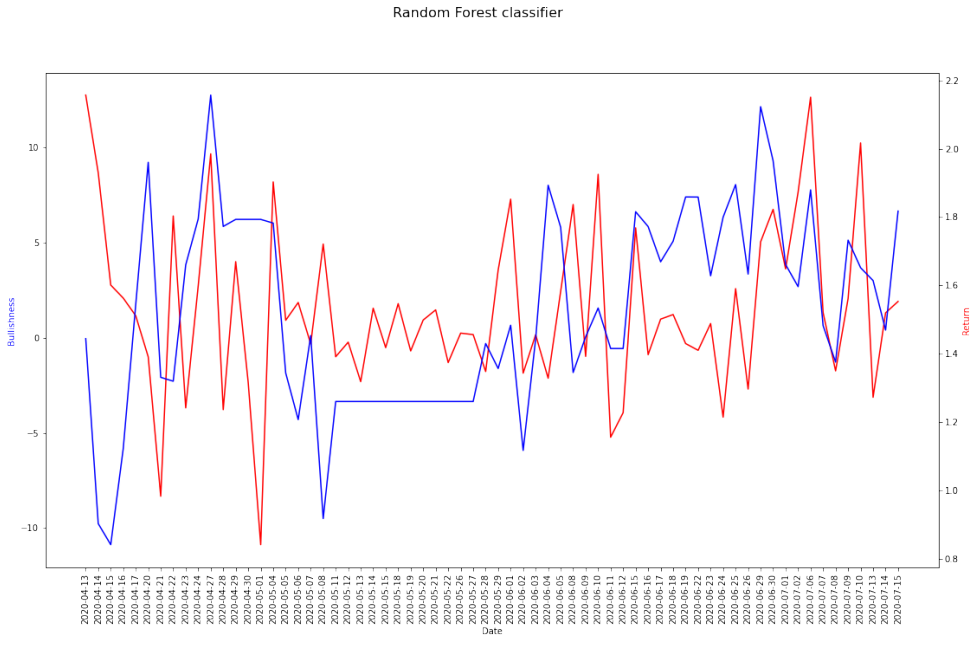

图4.16 TSLA 随机森林分类器图

随机森林的预测最接近实际值（图4.16），该图的预测结果也是所有模型中最接近实际值的，只有少数错误。

4.2.5 亚马逊股票（$AMZN）评估

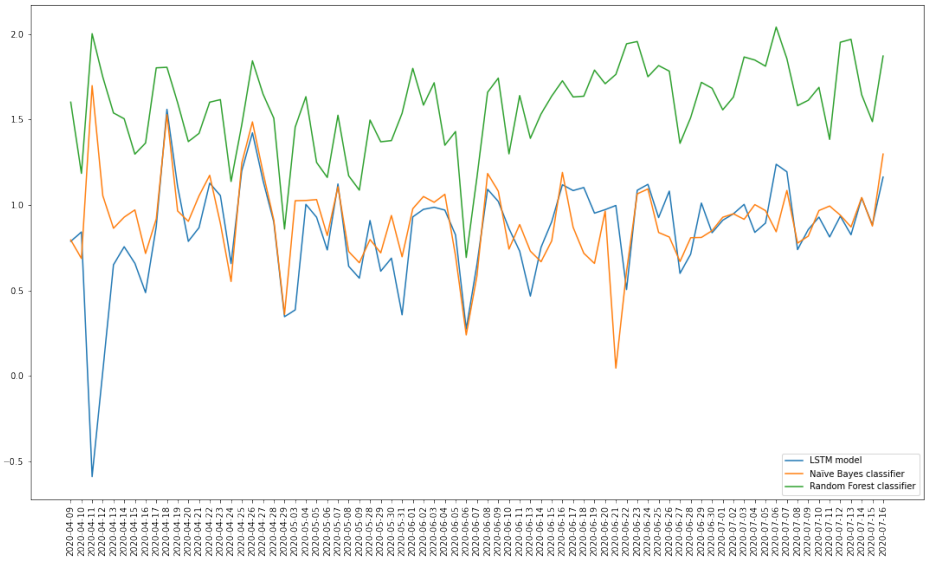

图4.17 AMZN看涨情绪图

如图4.17所示，三个模型对大多数上升或下降走势的判断一致。随机森林的预测最乐观，朴素贝叶斯的预测最悲观，长短期记忆网络则介于两者之间。

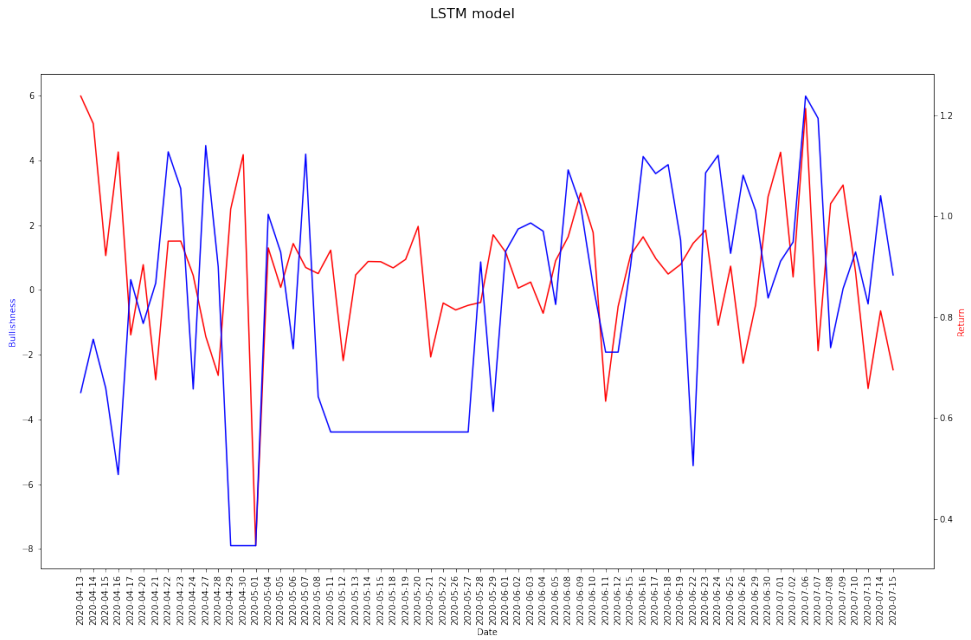

图4.18 AMZN 的 LSTM模型图

LSTM模型能够非常准确地预测某些模式（图4.18）。然而，有些预测值与实际值相差甚远。

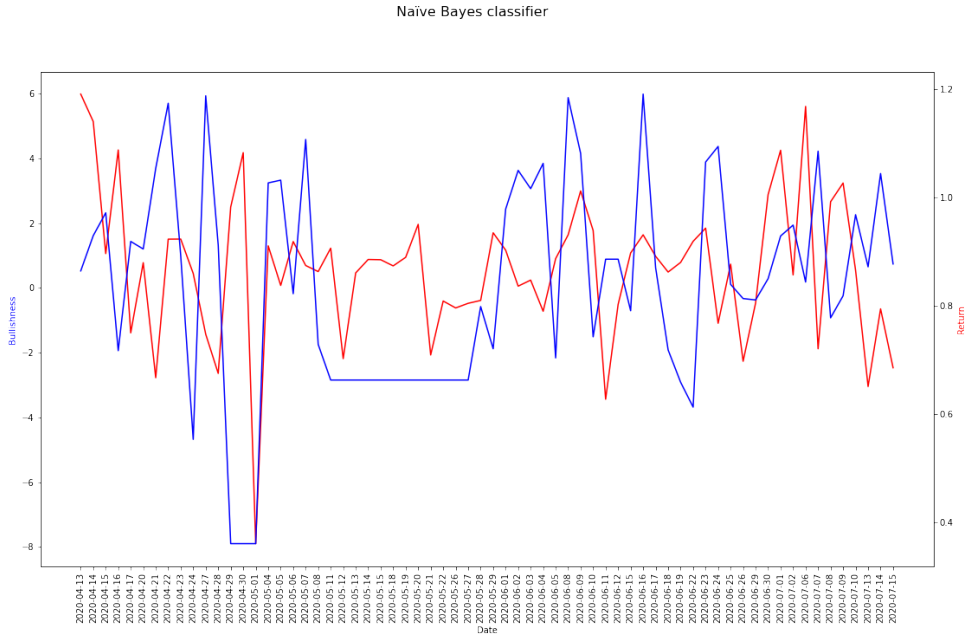

图4.19 AMZN 的朴素贝叶斯分类器图

如图4.19所示，朴素贝叶斯的预测似乎并不那么令人满意，但

仍然优于长短期记忆网络的预测。

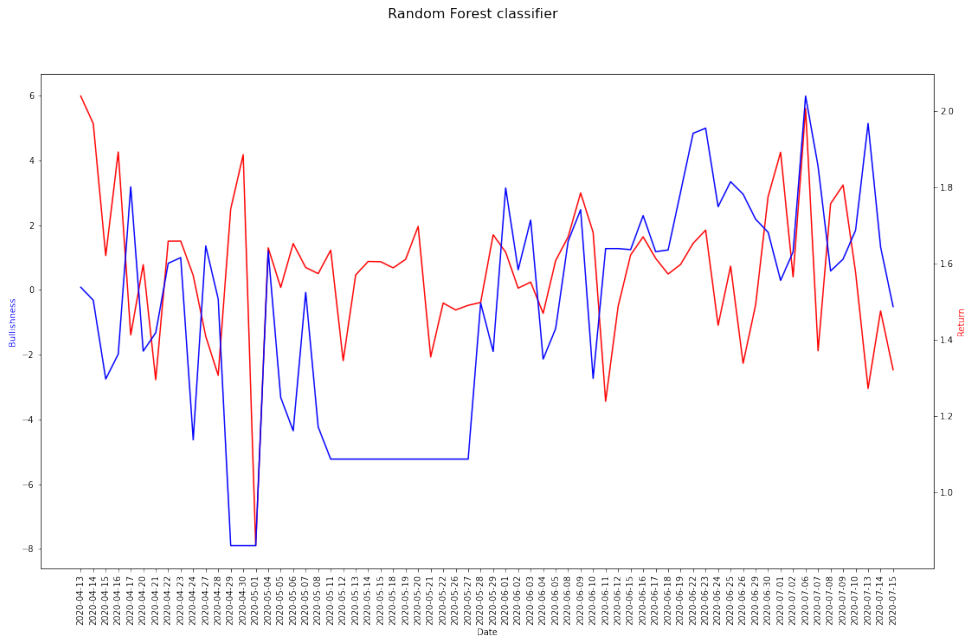

图4.20 AMZN随机森林分类器图

图4.20显示，随机森林的表现与朴素贝叶斯相似。有些交易日朴素贝叶斯的结果更接近实际值，而另一些交易日则是随机森林的结果更接近。

4.2.6 总体比较

随机森林的预测最为接近实际值。长短期记忆网络和朴素贝叶斯的表现相当接近。有时长短期记忆网络表现更好，有时则是朴素贝叶斯更胜一筹。综合所有情况来看，朴素贝叶斯略胜一筹。需要考虑到，在构建模型时，我们并未十分精确地设定参数，数据或参数计算中也可能存在其他误差。

第五章

结论

本章概述了本学位论文的主要研究结果以及未来研究的潜在方向。

5.1 结果与讨论

本文回顾了情感分析这一日益热门领域中的许多杰出研究，并探讨了各类前沿成果。

在此，我们通过分析推特上有关市场帖文的极性，预测美国股市未来的走势。我们介绍了各种情感分析技术的不同阶段和策略。此外，我们探讨了情感分析研究中可能使用的多种数据形式和工具，并提出了关于其优缺点的建议。总体而言，本研究旨在确定公众意见评分与股价变化之间是否存在关联。

本研究还探讨了在此类情境中应采用哪些最合适的情感分析方法，以及确定预测股价波动最相关的变量。然而，数据部分的数据集筛选和处理过程困难且耗时。例如，相当一部分推文有时含有垃圾邮件。因此，我们也花了大量时间对推文进行标注和分词。

在数据经过正确的整理和清理后，我们使用了各种情感分析模型和算法，并对其进行了评估。

具体而言，我们使用长短期记忆网络、伯努利朴素贝叶斯和随机森林分类器，分析了该推特数据集的平均情感值。随后，我们比较了结果，以确定哪个模型最准确；结果表明，所有这些算法之间都存在相关性。

最后，第3章总结了主要评估方案，并对模型在五个股票代码上的结果进行了全面分析。

根据第4章对股票的评估结果，推文的情感会显著影响股价，而算法的结果也表明，其对金融市场的趋势有着显著影响。

换句话说，我们得出的结论是，关于特定公司的推文可能会影响其股价表现。当然，这并不意味着两者总是存在直接联系，但数据表明，推文情感往往会导致股票价格和市值波动。

5.2 前景与未来工作

虽然学位论文的总体目标已经实现，但仍有一些方面可以改进。若能进一步拓展这项研究，以完善一种用作股票市场动量信号的情绪指标，将会很有意义。

换句话说，我们认为这些机器学习方法的准确性还有提升空间。以下是我们认为可能有助于此的建议。

- 尽管推特拥有许多国际用户，我们却只关注英文句子。我们的方法也应能对其他语言的情感进行分类。
- 不能忽视中性推文；中性情绪需要得到充分考虑。为了获得更准确的结果，音频文件、电影和照片也必须采用相同的标准进行考量。
- 当前的算法会对推文的总体语气进行分类。推文的极性可能因对推文的解读不同而有所变化。例如，推文“英格兰击败德国”中的情感对英格兰而言是正面的，而对比利时而言则是负面的。在这里，语义学可能会发挥作用。根据语义角色标注器的结果，应依据与动词主要相连的名词来进行分类。这样一来，“德国击败英格兰”与“英格兰击败德国”可能会被归入不同的类别。
- 延长研究时段十分重要，这样可以涵盖更广泛的推文和事件，从而克服 COVID-19 大流行等特定时期事件所带来的局限性。通过在不同经济状况的背景下考察这些模型，例如经济繁荣期和经济衰退期，也可以更好地了解模型的有效性。
- 人工神经网络可用于更准确地分析带有情感的推文。

参考文献

[1] M. Graham、S. A. Hale、D. Gaffney，你身在何处？推特中的地理定位与语言识别，《专业地理学家》66（2014）568–578。[2] V. D. Nguyen、B. Varghese、A. Barker，2013年皇室新生儿诞生：利用推特分析并可视化英国公众情绪，载于：2013年IEEE国际大数据会议，电气电子工程师学会，2013年，第46–54页。[3] A. Java、X. Song、T. Finin、B. Tseng，我们为何使用推特：了解微博的使用情况与社群，载于：第九届WebKDD与第一届SNA-KDD网络挖掘和社交网络分析研讨会论文集，2007年，第56–65页。[4] G. Ranco、D. Aleksovski、G. Caldarelli、M. Grčar、I. Mozetič，推特情绪对股票价格回报的影响，《PLOS ONE》10（2015）e0138441。[5] J. Degenhard，2017年至2025年全球推特用户数量预测，2021年。网址：https://www.statista.com/forecasts/1146722/twitterusersintheworld。[6] H. Miao、S. Ramchander、J. K. Zumwalt，信息驱动的价格跳跃与交易策略：来自股指期货的证据，SSRN电子图书馆（2011）。[7] B. Qian、K. Rasheed，使用多个分类器进行股市预测，《应用智能》26（2007）25–33。[8] B. G. Malkiel，有效市场假说及其批评者，《经济展望杂志》17（2003）59–82。[9] H. Mao、S. Counts、J. Bollen，预测金融市场：比较调查、新闻、推特和搜索引擎数据，arXiv预印本 arXiv:1112.1051（2011）。[10] T. O. Sprenger、A. Tumasjan、P. G. Sandner、I. M. Welpe，推文与交易：股票微博的信息含量，《欧洲金融管理》20（2014）926–957。[11] E. Gilbert、K. Karahalios，普遍的担忧与股票市场，载于：第四届AAAI国际博客与社交媒体会议，2010年。[12] J. Bollen、H. Mao、X. Zeng，推特情绪可预测股票市场，《计算科学杂志》2（2011）1–8。[13]B. Taborda、A. De Almeida、J. C. Dias、F. Batista、R. Ribeiro，股市推文数据，2021年。网址：https://www.kaggle.com/datasets/utkarshxy/stock-markettweets-lexicon-data。[14] D. Watson、A. Tellegen，迈向情绪的共识结构，《心理学通报》98（1985）219。[15] P. Ekman，是否存在基本情绪？（1992）。

[16] L. Yue、W. Chen、X. Li、W. Zuo、M. Yin，《社交媒体情感分析综述》，Knowledge and Information Systems 60 (2019) 617–663。[17] J. M. G. F. F. Sacramento，《基于推特数据的股票市场情感分析》，博士论文，ISCTE-Instituto Universitario de Lisboa（葡萄牙），2021。[18] A. Yadav、D. K. Vishwakarma，《基于深度学习架构的情感分析：综述》，Artificial Intelligence Review 53 (2020) 4335–4385。[19] W. Medhat、A. Hassan、H. Korashy，《情感分析算法及其应用：综述》，Ain Shams engineering journal 5 (2014) 1093–1113。[20] Z. Drus、H. Khalid，《社交媒体中的情感分析及其应用：系统性文献综述》，Procedia Computer Science 161 (2019) 707–714。[21] A. Agarwal、B. Xie、I. Vovsha、O. Rambow、R. J. Passonneau，《推特数据的情感分析》，载于：社交媒体语言研讨会论文集（LSM 2011），2011，第30–38页。[22] N. C. Dang、M. N. Moreno-García、F. De la Prieta，《基于深度学习的情感分析：比较研究》，Electronics 9 (2020) 483。[23]A. Teixeira-Pinto、J. Harezlak，2021。网址：https://bookdown.org/tpinto_home/Unsupervised-learning/k-means-clustering.html。[24] I. El Alaoui、Y. Gahi、R. Messoussi、Y. Chaabi、A. Todoskoff、A. Kobi，《一种用于大型社交数据情感分析的新型自适应方法》，Journal of Big Data 5 (2018) 1–18。[25] B. Gamal，朴素贝叶斯算法，2021。网址：https://medium.com/analytics-vidhya/na%C3%AFve-bayes-algorithm-5bf31e9032a2。[26]O. Mbaabu，2020。网址：https://www.section.io/engineering-education/introduction-to-random-forest-in-machine-learning/。[27] K. Gopalakrishnan、F. M. Salem，《使用简化长短期记忆循环神经网络进行情感分析》，arXiv预印本 arXiv:2005.03993 (2020)。[28] MATLAB，YouTube，2018。网址：https://www.youtube.com/watch?v=5dMXyiWddYs。[29] A. Go、R. Bhayani、L. Huang，《使用远程监督进行推特情感分类》，CS224N项目报告，斯坦福大学 1 (2009) 2009。[30] S. V. Kolasani、R. Assaf，《利用神经网络对推特信息流进行情感分析以预测股票走势》，Journal of Data Analysis and Information Processing 8 (2020) 309–319。[31] G. A. Ruz、P. A. Henríquez、A. Mascareño，《通过贝叶斯网络分类器分析重大事件期间的推特数据情感》，Future Generation Computer Systems 106 (2020) 92–104。[32]G. Novack，《使用 TensorFlow 构建独热编码层》，2020。网址：https://towardsdatascience.com/building-a-one-hot-encoding-layer-with-tensorflow-f907d686bf39。[33]S. Sharma，《神经网络中的激活函数》，2021。网址：https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6

[34] T. Rao、S. Srivastava，《推特情感分析：如何在股票市场中对冲风险》，载于《社会网络分析的前沿应用》，施普林格，2014年，第227–247页。