

英国芯片商店地图

曹家豪

2018年8月23日

高性能计算理学硕士

爱丁堡大学

介绍年份: 2018年

# 抽象

在互联网时代, 越来越多的餐馆通过互联网宣传餐馆。这产生了大量的网站数据, 这些数据非常混乱。过去,

这些餐馆数据可能只会帮助食客点菜。议员在大数据时代, 在数据工作者看来, 这些餐馆数据中有很多潜在的有价值的信息。"鱼和芯片" 是英国最著名的菜肴之一, 本文的重点是 "鱼和芯片" 商店 "的菜单数据, 通过菜单内容的分布来探讨英国的地区差异。

本文描述了从数据采集到区域结果获取的整个探索过程。具体而言, 它包括数据源选择、从 "鱼类和芯片" 商店网站检索数据的方法、清理网站数据集的程序、菜单内容地理分布的可视化、区域探索的过程。用于对区域和非区域内容进行分类的功能和机器学习方法。该项目最终发现了区域内容、区域菜品、区域表达的一些特点, 以及对区域效果的一些猜测。

这是一项创新的探索性研究, 没有现有的想法或方法可以参考, 所有的想法和方法都需要论文去探索和探索评价。因此, 本文所采用的方法、算法和研究结果可为今后的类似研究提供一些参考。

# 内容

第1章简介..。1

1.1 论文背景..。1

1.2 研究目标和研究重点2

1.3 研究方法..。2. 1.4 研究的价值..。3个

1.5 论文的结构..。3个

第二章背景理论..。5

2.1 网络爬网..。5

2.2 HTML 数据清理技术..。5

2.2.1 的 HTMLParser..。6

2.2.2 自然语言处理 (NLP)..。7。

2.3 地理数据可视化..。8

2.3.1 中心点计算算法..。8

2.3.2 可视化与马特普洛利布..。9

2.4 使用机器学习方法进行数据挖掘..。10

2.4.1 决策树..。10

2.4.2 逻辑回归..。12

第三章迭代 1..。14

3.1 方法..。14

3.1.1 数据采集..。14

3.1.2 数据清理..。15

3.1.3 数据可视化..。17

3.2 发现..。19

3.2.1 区域文字发现..。19

3.2.2 非区域性词语的发现..。21

3.3 评估和改进..。24

3.4 摘要和今后的工作。28

第4章迭代 2..。29

4.1 方法..。29

4.1.1 培训数据集..。29

4.2 发现..。30

4.3 评估和改进..。30

4.3.1 购物车算法的结果和评价..。32

4.4 总结和今后的工作..。33

第5章迭代 3..。34

5.1 方法..。34

5.2 发现..。35

5.3 评估..。35

5.4 摘要和未来的工作..。37第六章迭代 4..。38

6.1 名词短语..。38

6.1.1 名词短语的方法论..。38 6.1.2 名词短语决策树的发现..。39

名词短语决策树6.1.3 评价..。40

6.1.4 名词短语逻辑回归发现..。40

名词6.1.5 逻辑回归评价..。40

6.2 字对..。41

6.2.1 单词配对方法..。41

6.2.2 单词对决策树的发现..。41

6.2.3 单词对决策树的计算..。43

6.2.4 单词对逻辑回归结果..。43

6.2.5 单词对逻辑回归评估..。43

6.3 摘要..。44

第七章结论..。45

第8章未来的工作..。47

8.1 找到更多的 "鱼和芯片" 商店菜单..。47

8.2 对分类模型的算法和参数进行深入研究..。47

8.3 同义词检测..。47

8.4 将项目扩展到更广泛的范围。48

附录 A 逻辑回归概率结果..。49

引用。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。57

# 表列表

表 1: 带有 "比率" 的区域词语..。24

表 2: 带有 "比率" 的非区域词语.。25

表 3: "平均距离" 的区域单词..。26

表 4: "平均距离" 的非区域词语.。27

表 5: "比例" 的区域词语.。27

表 6: "比例" 的非区域词语.。28

表 7: 选择独立单词时的正则化选择的比较

"比例"、"比例" 和 "平均距离" 作为特征..。35

表 8: 名词短语正则化选择的比较..。40

表 9: 选择单词对的惩罚选择比较, "比例",

"比率"、"城市编号" 和 "平均距离" 作为功能..。43

# 数字一览

图 1: 工作流图..。3个

图 2: 一个 "鱼和芯片" 商店的 html 样式的 HTML 源代码

.......................................................................................................................................7。

图 3: 一个 "鱼和芯片" 商店的 HTML 源代码..。7。

图 4: 带有笛卡尔坐标系的地理坐标系..。8

图 5: Haggis 分布范围..。9

图 6: Map 脚本和减少脚本的数据操作流程图..。15

图 7: 减少脚本的结果..。17

图 8: 区域分布词的猜想图..。18

图 9: 非区域分布词的猜想图..。18

图 10: "哈吉斯" 分布 (95%)..。19

图 11: "haggis" 商店的数量随距离的变化而变化..。19

图 12: "兄弟" 分布 (95%)..。20

图 13: "兄弟" 商店的数量随距离的变化而变化..。20

图 14: "naan" 分布 (95%)..。20

图 15: "naan" 商店的数量随距离的变化..。20

图 16: "roe" 分布 (95%)..。20

图 17: "roe" 商店的数量随距离的变化而变化..。20

图 18: "晚餐" 分布 (95%)..。21

图 19: "晚餐" 商店的数量随距离的变化而变化..。21

图 20: "pakora" 分布 (95%)..。21

图 21: "pakora" 商店的数量随距离的变化而变化..。21

图 22: "芯片" 分布 (95%)..。22

图 23: "芯片" 商店的数量随距离的变化而变化..。22

图 24: "香肠" 分布 (95%)..。22

图 25: "香肠" 商店的数量随距离的变化而变化..。22 图 26: "最高" 分布 (95%)..。22

图 27: "至尊" 商店的数量随距离的变化而变化..。22

图 28: "礼品" 分布 (95%)..。23

图 29: "礼品店" 的数量随距离的变化而变化..。23

图 30: "汤" 分布 (95%)..。23

图 31: "汤" 店的数量随距离的变化而变化..。23

图32:"每日" 分布 (95%)..。23

图 33: "每日" 商店的数量随距离的变化而变化..。23

图 34: "马萨拉" 分布 (95%)..。25

图 35: "马萨拉" 商店的数量随距离的变化而变化..。25

图 36: "funghi" 分布 (95%)..。26

图 37: "funghi" 商店的数量随距离的变化而变化..。26

图38:ID3 算法决策树..。30 图 39: 购物车算法决策树..。32 图 40: 名词短语决策树..。39

图 41: 单词对决策树..。42

# 确认

首先, 我要感谢我的上司艾丽·休姆和简·肯尼迪在我的整个论文中提供的耐心咨询。当我遇到问题时, 他们总是及时给我建设性的建议, 这些建议让我在论文中学到了很多东西。有时他们会非常耐心地向我解释, 让我理解这个问题, 因为我的母语不是英语。在我写论文的过程中, 他们给了我很多反馈, 真的帮助我提高了很多。

此外, 我还要感谢所有在我研究生学习期间教我知识的老师。

此外, 我还要感谢我的初中老师马女士, 她是我一生中最重要的老师。她让我养成了努力学习的好习惯, 教会了我做一个负责任的人。没有她, 我可能就没有机会来爱丁堡学习。

最后, 我要感谢父母对我学习的无条件支持, 永远给我鼓励和爱。它们是我生命中最重要的部分, 感谢他们为我的成长所做的一切。

**第1章**

# 介绍

本章概述了本文的研究概况, 包括论文背景、本文的研究目的、本文采用的方法、研究价值和本研究的主要结构。

## 1。1 论文背景

在大数据时代, 从工业领域、业务领域、研究领域等多个领域产生了大量的混乱数据。杂乱数据是一种不能直接提供清晰可解释的信息的数据 [1]。然而, 在混乱的数据中, 有大量潜在的有价值的信息。因此, 处理混乱的数据是一项有价值的活动, 可以将混乱的数据转换为结构化数据, 以便于分析 [2], 从而帮助工程师或统计人员获得更有价值的信息。该项目主要侧重于处理餐饮领域产生的混乱数据。

在互联网和智能手机快速发展的时代, 搜索餐馆或特定食品变得更加容易。例如, 有了谷歌, 客户可以更简单、更准确地找到自己感兴趣的餐馆或食品类型。同时, 餐馆操作员可以使用这些技术更好地扩展他们的事务 [3]。发现这个商机后, 越来越多的餐馆经营者正在一些信息应用程序和网站上公布他们的菜单链接, 如谷歌和 TripAdvisor [4]。这导致了大量的餐馆信息, 如菜单和餐馆描述, 以数字形式公开。就普通食客而言, 餐馆网站上的内容只能为他们提供餐饮参考。然而, 在数据工作者看来, 从这些餐馆的数据中可以挖掘出很多潜在的有价值的信息。这意味着, 如果在餐馆的网站上使用数据挖掘技术 [5], 可以发现一些有价值的发现, 例如一个国家或地区的地区差异。本文将介绍一种获取、处理非常混乱的餐馆网站数据、挖掘区域特征和从这些数据中对区域内容进行分类的方法, 以揭示区域差异。

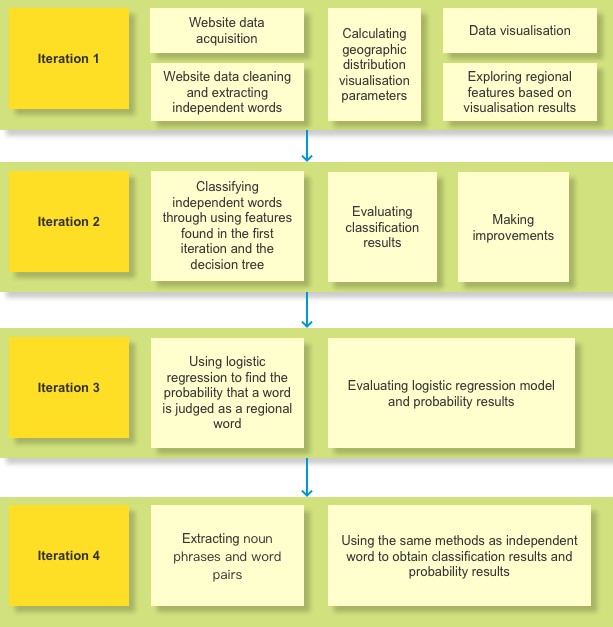
## 1。2 研究目标和研究重点

该项目希望找到未知的区域菜肴, 但该项目不希望手动找到菜单项通过使用字典。因此, 本文的目的是挖掘 "鱼和芯片" 商店的菜单数据, 并根据菜单数据内容的地理分布揭示英国的地区差异。例如,

"哈吉斯" 是苏格兰的一种传统食品, 分布广泛, 但在英格兰却很少见。根据本文采用的方法, 我们可以提供证据, 证明 ' 哈吉斯 ' 受到苏格兰人民的喜爱, 是苏格兰的一种地区菜。"鱼和芯片" 是英国最著名的食品之一, 这个国家有 1, 000多家 "鱼和芯片" 商店 [6]。为了实现该项目的目标, 我们将从其中一些 "鱼类和芯片" 商店的网站获取原始 HTML 数据, 然后使用数据清理、挖掘和可视化技术来查找与特定区域相关的内容。

## 1。3 研究方法

在数据爬网方面, 本文将说明从英国 "鱼和芯片" 商店网站上对数据进行抓取的数据来源和方法的选择。数据清理过程的重点是从网站 HTML 内容中提取和清理用于探索区域性的文本内容, 如单个独立词、名词短语和单词对。用于提取和清理 HTML 内容的方法是正则表达式、HTMLParser 和自然语言处理 (NLP) 的组合 (将在2.2 中详细介绍)。考虑到研究的数据挖掘过程, 将数据可视化技术应用于基于提取内容的地理分布的区域特征挖掘。在提取的数据集的分类 (区域内容和非区域内容) 方面, 该项目采用机器学习方法, 如决策树和回归分类器 (将在2.4 中详细说明), 以生成区域结果。具体而言, 这项研究是一个迭代过程, 包括四轮评估和改进 (如图1所示), 因为整个研究是一个探索性的过程。除了第一次迭代之外, 每个剩余的迭代都取决于最后一次迭代的结果。这意味着, 只有在获得并评估每次迭代的结果后, 项目才知道下一步要进行什么。此外, 目前还没有证明方法选择的合理性和结果正确性的标准。因此, 只有在每次迭代完成后, 项目才能知道方法的选择是否合理, 结果是否正确。此外, 可用于反映文本区域性的特点尚不清楚, 对区域内容的评价在一定程度上是以评价人员的经验为基础的。例如, 评价者知道 ' Irn bru ' 是苏格兰的饮料, 所以当这句话被判定为区域性时, 评价者可以认为这个决定是正确的。因此, 区域特征和区域结果来自于每次迭代, 包括勘探、评价和改进。在每次迭代中, 项目都可能使用或更新上一次迭代中的方法。此外, 每次迭代还将评估结果, 以确定问题, 并为下一次迭代提出改进建议。



**图 1: 工作流程图**

## 1。4 研究的价值

这项研究是一项创新的研究, 通过从混乱的菜单数据集中探索地区内容, 将看似无关的菜单信息与英国的地区差异联系起来。因此, 这个项目面临的最大挑战是, 没有现有的想法或方法可以参考, 所有的想法和方法都需要项目去探索和评价。幸运的是, 该项目成功地找到了探讨区域内容和反映区域差异的解决办法, 包括寻找区域特点和区域文字分类方法的想法。该项目中使用的方法和算法是通用的, 也可以用来发现其他国家的地区差异或在类似的研究中使用。因此, 该项目为随后的类似研究奠定了基础。例如, 如果一项研究想通过餐馆数据了解其他国家的地区差异或消费习惯, 可以参考这个项目中使用的方法和研究过程。

## 1。5 论文的结构

本文的结构结构如下: 第二章介绍了背景知识, 主要阐述了本研究中使用的主要技术和算法。第3章、第4章、第5章和第6章介绍了项目中每个迭代的详细信息, 包括方法、结果、评价、改进、总结和未来工作。第3章描述了第一次迭代,介绍了区域独立单字特征的探索程序。第四章与第二个迭代有关, 描述了一个决策树的应用, 以获得单个单词的区域结果。本章采用两种类型的决策树算法, 并对两种算法进行了比较。第五章介绍了第三轮迭代, 引入了逻辑回归, 以获得将单个独立词判断为区域词的概率。本章的重点是评价选定特征的重要性和关于概率的阈值, 以确定概率超过阈值的数量, 这将被判断为区域词。第6章是关于第四次迭代, 它展示了使用其他两种数据集 (名词短语和单词对) 的结果。第七章是对本研究的结论。第8章介绍了可在未来工作中包含的改进, 并介绍了限制和建议。

**第2章**

# 背景理论

本章重点介绍了本文所采用的技术和方法。第2.1 节说明了数据源的选择和 web 爬网的定义。第2.2 节概述了项目中使用的数据清理过程和技术。第2.3 节介绍了用于地理数据可视化的核心算法和技术。第2.4 节旨在说明项目中的数据挖掘过程以及用于区域内容分类的一些机器学习方法。

## 2。1 网络爬网

在从 "鱼和芯片" 商店的网站上抓取数据之前, 论文选择了包括食品交付网站 (如 "吃" 和独立网站) 的数据来源

"鱼和芯片" 商店 "网站。使用食品配送网站作为数据源的好处是, 通过邮政编码搜索英国每个城市的 "鱼和芯片" 商店都很方便。此外, 从食品递送网站搜索的每个商店都可以直接抓取商店有有效链接且该链接的页面具有菜单内容的数据。而在独立网站上搜索时, 可能会出现所需的内容 (如菜单) 仅在 PDF 中提供, 因此无法对其进行爬网。此外, 在独立网站上, 可能会出现网站上提供的 Url 被破坏的情况, 因此也无法对其进行爬网。因此, 论文最初计划将食品配送网站作为数据源。然而, 像 Just-eite 这样的网站明确指出, 不允许直接抓取他们的网站数据 [8]。因此, 在本文的项目中, 利用这样的食品配送网站直接抓取数据可能是非法的。此致命缺陷意味着无法使用此方法。因此, 谷歌被用来寻找 "鱼和芯片" 商店的独立网站, 这些独立网站的数据随后被用作数据来源。

网络爬网是从目标网站收集信息的机制 [9]。具体而言, Python 模块 urllib2 已被用来模拟浏览器行为下载网页和处理请求错误 [10], 以获取 "Fish & Chps" 商店的完整网站 HTML 数据。

## 2。2 HTML 数据清理技术

数据清理用于通过检测不一致和删除错误来提高用于后续处理的数据的质量 [11]。在此项目中, 需要清理的数据集是 HTML 数据。这个项目的数据清理目标是获得独立的单词 (如 "haggis")、名词短语 (如 "monday 晚餐交易 haggis") 和单词对 (如 "带有 haggis 的沙拉") 可以与商店一起分为 "沙拉与" 和 "与 haggis" 单词对 "包含坐标的 HTML 数据集和城市数据集中的坐标。

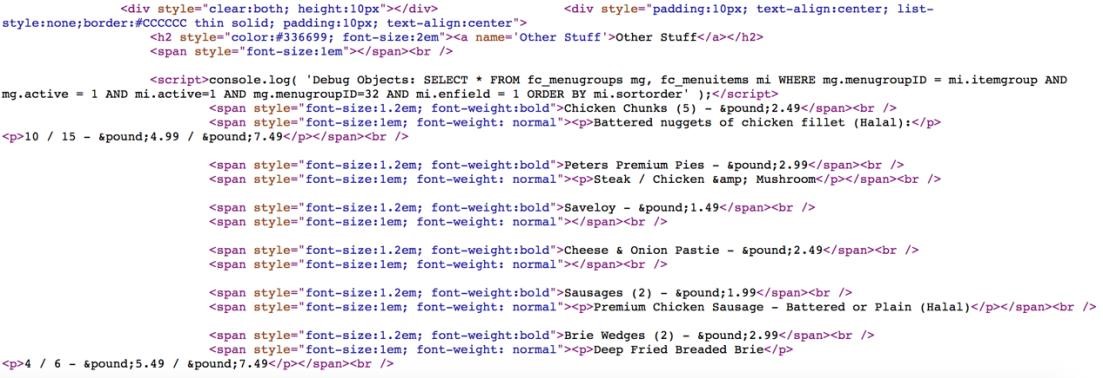
在基于 web 的数据集中, 有许多此项目不需要的内容, 如名称、HTML 标记的属性、脚本代码和特殊符号。该项目只关注用户可以在页面上看到的信息, 而不是页面的实现细节。然而, 就客户可以看到的内容而言, 有很多冗余, 比如同一个名词的单数和复数都代表同一个词。因此, 该项目不仅应过滤 HTML 数据中无用的内容, 还应将代表 "芯片" 和 "芯片" 等含义相同的词分类为同一类别 (主要侧重于对具有相同的奇异和复数名词的分类)) 的含义)。幸运的是, 正则表达式、Htmparser 和 NLP 可以帮助项目实现数据清理目标。

### 2.2。1 HTMLParser

HTMLParser 是一个开源、快速且强大的 HTML 解析工具, 用于提取和清除 HTML [12] 的内容。它可以根据用户要求自定义 HTML 标记内容提取 [13]。在本项目中, HTMLParser 主要负责数据的提取和过滤。该项目中使用的数据源是独立网站, 大多数网站的 HTML 结构在图2和图3中是不同的 (网站结构的小部分是相同的, 因为它们是由同一家公司开发的)。 图2显示了本网站在页面上写入所有 CSS 样式的网站之一的 html 源代码的一部分。图3显示了另一个网站的 HTML 源代码的一部分, 此页在 HTML 元素标记之间引入了一些 JavaScript 代码。此外, 这两个网站在菜单的 HTML 结构中完全不同。因此, 为了从具有不同结构的 HTML 源代码中提取内容, HTMLParser 发挥着重要作用。它主要关注的是 html 标签 (如 "div" 和 "脚本") 的名称, 而不是网站设计的结构。因此, 该项目可以轻松地筛选绝对无用的内容基于标记名称, 如 "脚本" 标记中的内容, 并从剩余的标记中提取潜在的有价值的内容。但是, 由于不同网页的设计风格, 提取的数据也可能包含特殊的符号, 如字段尾随空间符号, 这将干扰清洁结果。因此, 该项目还使用正则表达式, 这是一种源语言, 它可以在文本 [14] 中定位特定的字符串来筛选 HTMLParser 的结果。



### 图 2: 一个 "鱼和芯片" 商店的 html 样式的 HTML 源代码



**图 3: 一个 "鱼和芯片" 商店的 HTML 源代码**

### 2.2。2 自然语言处理 (NLP)

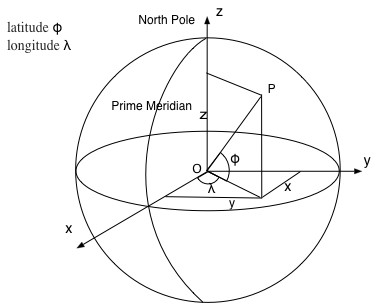
为了解决提取内容中的数据冗余问题, 该项目采用了 NLP 中的语义识别方法。自然语言处理 (NLP) 是使用计算机来理解和操作自然文本或语音来处理任务 [15]。这个项目主要是要把同一根的单数和复数名词变成奇异名词, 自然语言工具包 (nltk) 可以提供解决方案。nltk 是由 Python 编写的一个开源工具, 其中包含模块和语料库 [16]。nltk 根据一个词的词类来确定一个词的词性部分, 并封装了该词可以直接用来识别复数名词并将其转换为奇异形式的识别方法。然而, 在英语中, 有些词既可以是复数名词, 也可以是动词, nltk 会把所有的词都当作名词, 并将它们转换为单数。幸运的是, 这个项目并不关心正在转换的单词是动词还是名词。它只关心这个词是否在区域内分布。

## 2。3 地理数据可视化

该项目的目的之一是探索地域分布的特征来代表区域性, 因此该项目使用笛卡尔坐标系用于地理位置计算和数据可视化的有性图利阵。

### 2.3。1 中心点计算算法

这个项目的核心计算是一组坐标 (获取的方法将在 3.1.2) 中心点中详细说明, 中心点是包含特定内容的所有商店的中心。中心点的重要性在于, 本项目中发现的大多数区域特征都来自于该项目。该项目使用一组包含此内容的地理坐标来计算质心。本项目中用于计算中心点的算法, 通过将地理坐标系与将地球视为球体的笛卡尔坐标系结合起来 (图 4)。这种组合也被称为 ECEF ("以地球为中心, 地球固定") [17]。在笛卡尔坐标中, 地球是一个以原点为中心的球体 [18]。z 轴指向北极。x, y 轴是在赤道平面上, x 轴通过赤道和本初子午线, y 轴指向赤道在东经90度 [19]。然而, 在该方案中使用该算法获得的中心点的坐标并不准确, 因为该算法将地球视为球体, 而不是椭圆, 而不是地球本身的形状。幸运的是, 对本项目中心点坐标精度的要求并不高, 因为这个项目涉及内容的分布。



### 图4: 带有笛卡尔坐标系的地理坐标系

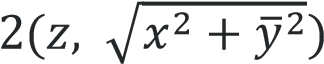
如图4所示, 点 P 表示具有纬度的地理坐标Φ和经度.一系列坐标可以表示为纬度Φ我,经度我(i = 1)...,n)。因此, 在笛卡尔坐标系, 三个方向的坐标可以表示为:

1. X我*=*r cos我科斯格普斯·格科我
2. Y我 *=*r cos我sin我
3. Z我 *=*R罪我

这些点的质心是平均的总和X我Y我Z我:

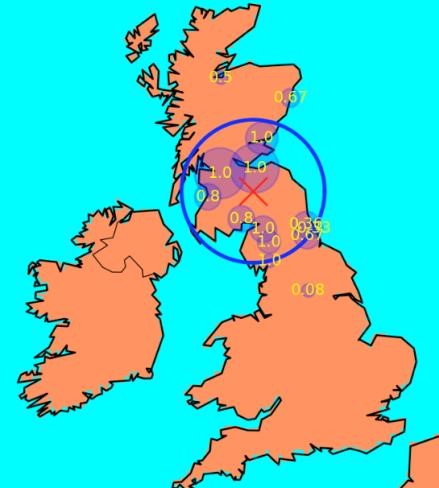
1. *(*x节, y兴, z*)*̅ n (x我Y我Z我)

的坐标。质心可以表示为:

1. 请让*=*阿坦 ̅ ̅
2. *=*atan2(x̅, 是的)

#### 2.3。2 使用 Matplotlib 可视化

在项目获得中心点和半径等其他要素 (详情将在3.1.3 中描述) 后, 该项目希望在英国地图上显示特定内容的分布情况, 以观察内容的分布情况。该项目选择的可视化工具是 Python 的 Matplotlib 包, 它是一个开源的便携式 Python 绘图包, 用于科学、工程和金融领域 [20]。它可以通过简单的封装方法实现复杂的数据可视化过程。由于此工具的便利性, 项目中的大多数数据可视化过程都是通过它实现的。具体而言, 在本项目中, Matplotlib 包主要完成地理信息分布和折线图的可视化。在地理信息分布的可视化方面, 该项目使用了一个名为 Basemap [21] 的 Matplotlib 工具包。底图提供了一种可能性, 该项目可以绘制出在真实世界地图上的 Matplotlib 绘图 [22]。这表示底图替换了 Matplotlib 的底部画布, 因此它可以实现在地图上绘制其他图形 (如半径和周长曲线) 的目标。下图 (图 5) 是使用底图可视化 "哈吉" 的分布的示例 (详细信息将在3.1.3 中说明)。



**图 5: Haggis 分布范围**

## 2。4 使用机器学习方法进行数据挖掘

这个项目是一个探索性项目, 因此, 在项目开始时, 对区域特征没有明确的定义。因此, 特征的发现和区域结果的获取是数据挖掘的过程。数据挖掘是从海量数据集中提取表示有用信息的模式的过程 [23]。在项目的早期阶段, 发现功能经历了两个阶段。首先是根据图5等内容的地域分布结果, 找到一些在地图上广泛分布的内容和一些区域分布的内容。第二阶段是寻找这些广泛分布的内容和区域内容的共性。共性可被视为内容的特点 (区域或非区域) (将在3.2 和3.3 中说明)。在项目采用上述方法获得更多功能后, 该项目尝试了两种机器学习方法对区域内容进行分类。其中之一是决策树, 另一个是逻辑回归, 两者都属于监督学习的方法。监督学习意味着培训数据既有特征, 也有标签。通过训练, 机器可以找到功能和标签之间的连接本身, 并可以判断标签时, 面对数据只有没有标签的功能 [24]。在本项目中, 培训数据集包括广泛分布的内容和区域内容。根据开发商的经验和项目的结果对它们进行了判断和添加。

### 2.4。1 决策树

决策树主要用于模型 [25] 的分类和预测, 该项目使用决策树对区域内容和广泛分布的内容进行分类。此项目中使用了两种算法。一个是 ID3 算法, 另一个是购物车算法。这两种算法都使用训练数据集创建树, 然后使用树对测试数据集进行分类 [26]。项目之所以选择 ID3 算法, 是因为项目首次获得的特征是分类的, 而不是连续的。因此, 该项目使用 ID3 算法, 该算法使用分类数据生成决策树 [27], 将内容分类为区域内容或不区域内容。但是, 使用 ID3 算法 (在4.3 中的详细信息) 存在一些缺点, 该项目尝试了另一种决策树算法--Cat 算法。

#### 2.4.1。1 ID3 算法

ID3 算法通过选择最有用的特征来构造决策树 [28]。这些功能可以使数据集的分类更加有效。因此, 该项目需要一种算法来测量要素的适用性并选择要素。熵可以测量训练数据集的杂质 [29], 即熵越大, 信息越复杂。因此, 项目可以通过添加一个特征来选择具有代表性的特征, 从而利用信息增益, 即损失的熵量。

熵: X 表示特征的集合,X = {x1X2X3,..., xn},p (x我)是发生的概率X我.

1. H ( X ) = - -n伊迪1p (x我) 日志2p (x我)

信息增益: 表示一项功能。

1. IG (X a) = H (X)-H (X a)

ID3 算法的决策树构建过程分为以下几个步骤:

* 加载训练数据集并将数据集视为第一个节点。
* 按每个要素拆分数据集, 并根据分割结果计算熵和信息增益。
* 选择具有最大信息增益的特征作为最佳分割特征。
* 根据优化分割特征将数据集分解为两个节点。
* 对每个新获取的节点重复2-4 步, 以递归方式生成树
* 样本分类。

#### 2.4.1。2 购物车算法

购物车算法使用二进制递归分区过程来拆分数据集 [30]。在分类树中, 购物车算法使用基尼索引作为属性来确定分区 [31]。基尼指数表示样品的不确定性。基尼指数越大, 样本集的不确定性就越大, 这意味着样本属于一个类的概率很低。在每个特征方面, 购物车算法将遍历所有可能的拆分方法, 并选择基尼索引最小的特征作为除法标准 [32]。下面的公式显示了基尼指数的计算。

假设有一个K类, 采样点属于K类是PK, 则吉尼指数定义为:

1. 基尼 (p) = 1-Kk=1PK2

假设CK是属于K类在D, 则吉尼指数为:

1. 基尼 C2

D

假设该功能A 个, 分割样本D, 成两个数据子集D1和D2, 然后样本的基尼指数D, 在功能下A 个是：

1. 基尼 (D, A) =DD1|吉尼 (D)1) +DD2|吉尼 (D)2)

使用购物车算法生成决策树的步骤如下所示:

* 使用每个功能A 个在样本中D,和每个可能的值A 个(A ≥a和A < a) 将样品分成两部分, 并计算基尼 (D, A)。
* 找到具有最小基尼的最佳分割特征 (D,A 个).接下来, 确定是否满足拆分停止条件。如果没有, 则输出最佳分割点。
* 递归呼叫1和2。

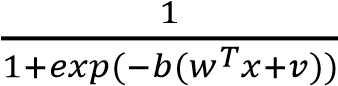
在这个项目中, 决策树的结果是二元的 (细节在4.2 和4.3 中), 这意味着内容被判断为区域性或非区域性。不过, 该项目也想知道将多少概率内容判断为区域性内容, 因为项目希望找到更多证据来验证分类结果。因此, 该项目尝试了另一种方法--逻辑回归。

### 2.4。2 逻辑回归

本项目研究分类问题, 因此模型的因变量是分类变量 (0 或 1), 模型的自因变量是非线性的。因此, 逻辑回归模型适用于该项目, 项目选择 Sklen-learn 学习包的逻辑回归模型作为分类器, 找到了每个内容被判断为区域性的可能性。

逻辑回归非常适合描述以分类结果和一个或多个分类之间的概率表示的关系 [33]。它可以适应多种分类结果。在这个项目中, 逻辑回归用于计算在多个独立特征下发生二进制事件的概率 [34]。以下模型是物流回归模型:

X表示特征变量的向量, 以及b {0, 1}表示关联的二进制输出。w表示权重向量和wT是 w 的转置矩阵。(.)是 sigmoid 函数。V是拦截。逻辑回归有以下模型:

1. Prob (b x; w) = (wTx) =

物流损耗函数 (Z是b (w)Tx + v):

1. f (z) = log (1 + exp (-z))

假设训练数据集是(x(2)Y(2))、(x(3)Y(3)),..., (x(m)Y(m))和平均物流损失:

1. 我Avg(v, w) = (半)米伊迪1f (b (w)Tx))

逻辑回归问题:

过度拟合问题: 在监督学习中, 当有许多输入特征时, 但只有少数关键特征决定了分类目标。也就是说, 当训练集数据数量不足时, 分类模型可能在训练数据集上表现良好, 但在测试数据集上表现不佳 [35]。因此, 当有许多功能时, 过度拟合将成为模型的一个问题, 除非训练集充足 [36]。为了解决这个问题, 使用了 L1 和 L2 正则化。

L1 正则化 [35]:

1. 我Avg(v, w) + "1= (半)米伊迪1f (b (w)T(x)) +n伊迪1w我|

套索 (L1) 惩罚鼓励的绝对值之和wi 是小的 [36]。它使用稀疏适合模型与许多特征 [37]。稀疏性意味着 L1 惩罚将自动筛选对分类影响较小的某些要素。L1 惩罚通过将回归系数降低到0和略微降低其他回归系数来实现滤波 [38]。 二等正则化 [35]:

1. 我Avg(v, w) + "22= (半)米伊迪1f (b (w)T(x)) +n伊迪1w我2

L2 惩罚鼓励平方的总和wi是小的 [36]。它将降低回归系数, 但不会为零 [38]。因此, L2 罚将削弱显性分类特征, 增强其他特征的影响。如果每个特征对分类都有影响, 则 L2 处罚更合适。

在此项目中, 使用逻辑回归及其 L1 和 L2 正则化不仅可以帮助项目获得概率, 还可以帮助项目查看每个功能对分类结果的影响。

**第3章**

# 迭代1

在迭代一中, 以下任务: 数据采集、数据清理和数据可视化由项目完成。数据采集程序主要负责菜单网站的检索和存储。数据清理过程主要负责从检索到的 HTML 文件中获取独立的单个单词。数据可视化短语分为两部分。其中之一是地理地图可视化, 另一个是趋势可视化 (详细信息, 请 3.1.3)。迭代一的目的是通过观察一些已知区域词的地理分布图和趋势图的结果, 发现区域词的区域特征。此外, 迭代一详细介绍了发现这些功能的过程, 并为下一次迭代制定了计划。

## 3。1 方法

迭代一主要集中在四个方面, 即网络数据采集、HTML 数据清理、地理数据和比率趋势可视化以及特征的探索。

### 3.1。1 数据采集

在该项目确定数据源是独立的 "Fish & Chps" 商店 ' 网站后, 该项目开始通过在谷歌和其他食品推荐网站上搜索发现这些网站的 Url。搜索网站的方法首先是寻找城市, 然后搜索 "鱼和芯片", 并从搜索结果中获取网站 Url。该项目开始时, 该项目收集了英国 2 1个人口较多城市的 "鱼和芯片" 商店网站, 如伦敦、曼彻斯特和格拉斯哥。然而, 该项目发现, 这些收集到的商店集中在中北部 (如爱丁堡、格拉斯哥) 和中南部 (如曼彻斯特、谢菲尔德), 北部 (如邓迪、英伦)、西南 (如普利茅斯) 和中部几乎没有商店地区 (如纽卡斯尔)。为了解决商铺分布不均的问题, 该项目增加了分布在北部、中部和西南部城市的 1 7家商铺。

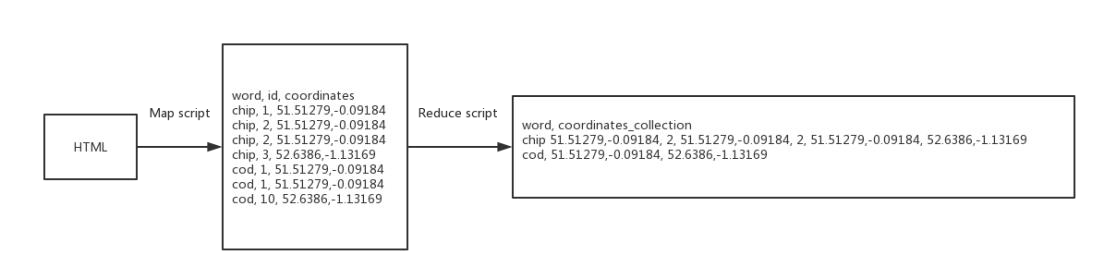
该项目的最初目标是获得一批商店, 这些商店的分销可以覆盖英国各地。不过, 在英国, 虽然有很多 "鱼和芯片" 店, 但并不是每个 ' 鱼和芯片 ' 店都在其网站上提供菜单。此外, 如2.1 所述, 其中一些 "鱼和芯片" 网站无法检索。因此, 该项目终于收集了 "鱼和芯片" 店的 2 4 0个可用网站。包含这些网站的商店的分布基本上覆盖了英国的大部分地区。然而, 尽管商店的分布覆盖了大多数城市, 但每个城市的商店数量仍然存在偏差。这意味着大多数商店集中在人口大的城市, 其他人口稀少的城市, 如英伦斯和卡莱尔, 商店较少。因此, 当项目以3.1.3 可视化商店的地理数据时, 这种不平衡就会得到反映。项目收集的每个 "鱼和芯片" 商店都将被分配 Id、城市名称和 URL, 此信息存储在 CSV 格式文件中。Id 用于唯一标识商店, 城市名称用于查找商店所在城市的坐标 (具有其坐标的城市存储在项目创建的另一个文件中)。

项目完成商店集合后, 该项目编写了一个 Python 脚本, 该脚本使用 urllib2 模块从收集到的网站检索 HTML 数据。urllib2 模块提供了一种模拟浏览器向网站发送 HTTP 请求的方法。该方法避免了一些网站因检测到异常访问而拒绝访问的问题。此外, 脚本使用具有不同城市坐标的文件, 并查找每个商店的地理坐标。(在这个项目中, 同一城市的商店的坐标是城市的坐标)。

该脚本为每个商店生成一个文件, 每个文件存储商店的 HTML 源代码, 文件名是商店的 Id (例如 1. csv)。此外, 该脚本还生成一个映射文件, 其中包含商店 Id、HTML 源代码文件的名称和该商店的坐标。此映射文件的主要作用是将商店与其网站 HTML 文件和坐标相关联。因此, 在数据采集过程之后, 项目将每个商店的 HTML 内容与其坐标连接在一起。

### 3.1。2 数据清理

此项目中的整个数据清理过程类似于 Map-locd, 该过程表明 map 方法生成一系列中间键和值对, 而理论依据方法根据相同的键值 [39] 合并 map 方法的结果。数据清理过程类似于 Mamap-设低) 的原因是, 此项目以键 (菜单字)-值 (商店 id 和一组坐标) 的形式从 HTML 源代码中提取内容, 并使用相同的键合并内容。该项目中使用了两个 Python 脚本来实现地图和减少过程, 以完成数据清理过程。Map 脚本负责提取独立的单个单词与他们的商店 id (其角色将在减少脚本中描述) 和商店坐标。减少脚本主要负责合并由 Map 脚本输出的单词。它将同一个单词的坐标添加到该单词的坐标集中, 但从不同的商店添加到该单词的坐标集中。图6是 Map 脚本的提取操作和减少脚本的合并操作的示例。



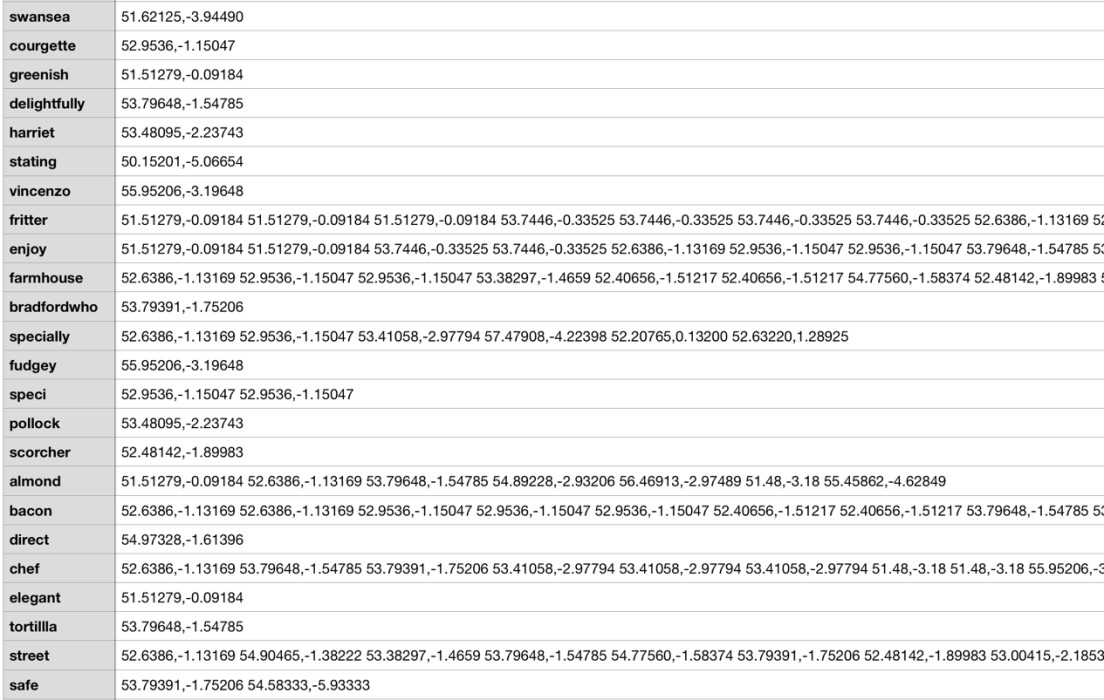
### 图 6: Map 脚本和减少脚本的数据操作流程图

Map 脚本主要侧重于过滤无用的 HTML 内容, 提取 HTML 标记中的内容, 将内容拆分为单独的独立单词, 并将所有复数名词转换为单数。首先, 脚本读取3.1.1 生成的映射文件, 并按顺序读取项目检索到的 HTML 文件, 并将 HTML 内容交给 Html 分析器进行处理。HTMLParser 首先筛选了带有项目不需要的内容的标记。无用的标签名称是由项目定义的, 在这个项目中, 无用的标签是 "脚本"、"样式"、"链接"、"头"、"a" 和 "标题", 因为这些标签中的大部分内容都是 HTML 代码。接下来, Htmparser 按顺序识别其他标记的名称及其内容, 但此项目仅侧重于内容。为了提取独立的单字, 该项目使用正则表达式和*拆分 ()*函数, 其中 HTMLParser 处理 HTML 标记的内容。该项目首先使用正则表达式来筛选数字和处理特殊符号, 如 "."、"+" 和 "-", 然后使用分割方法将内容拆分为单独的独立单词, 并将其转换为小写。

该项目最初希望在获得独立的单个单词后, 将复数名词转换为单一名词。但是, ASCII 编码的特殊空间 (如 "\ xc2\ xa0") 出现在一些独立的单个单词中, 正则表达式无法识别它们。出现这些特殊空间的原因是某些网站的字符集编码与编译器字符集编码之间的不一致。因此, 如果项目直接转换语音部分, 具有特殊空格的复数名词将被视为专有的奇异名词。此外, 该项目还发现, 这些带有特殊空格的单词在添加到列表后只显示了特殊空格的 ASCII 代码。因此, 该项目首先将 HTMLParser 的所有结果添加到输出列表中。输出列表的每一行都包括车间 Id、独立的单个单词 (可能包含特殊空格) 和商店坐标。接下来, 项目将输出数组转换为字符串, 特殊空格的 ASCII 代码作为 ASCII 字符存在, 然后项目使用正则表达式筛选这些字符。筛选特殊空格后, 项目将输出列表的字符串重新转换为列表, 并用作 Map 脚本的输出。

在 "减少" 脚本处理 Map 脚本的输出数据之前, Unix*排序 ()*方法用于对 Map 脚本输出列表中的独立单字列进行排序。这减少了 "减少" 脚本的工作负载, 在处理每一行时, "减少" 脚本不必确定当前行中的单词是否出现在之前出现的单词中。脚本只需确定当前行中的单词是否与上一行中的单词相同。如果相同, 请将当前单词的坐标添加到上一个坐标集。如果它们不同, 则为当前单词创建一个新的坐标集, 并将当前坐标添加到新的集合中。此外, 在这个项目中, 每个字只允许在一家商店出现一次。这意味着, 在具有相同的字数 (这是 Map 脚本的输出值之一) 的相同单词中, 只能向单词坐标集合中添加一组坐标。这样做的原因是添加到集合中的单词的坐标可以表示商店中所有相同单词的坐标。

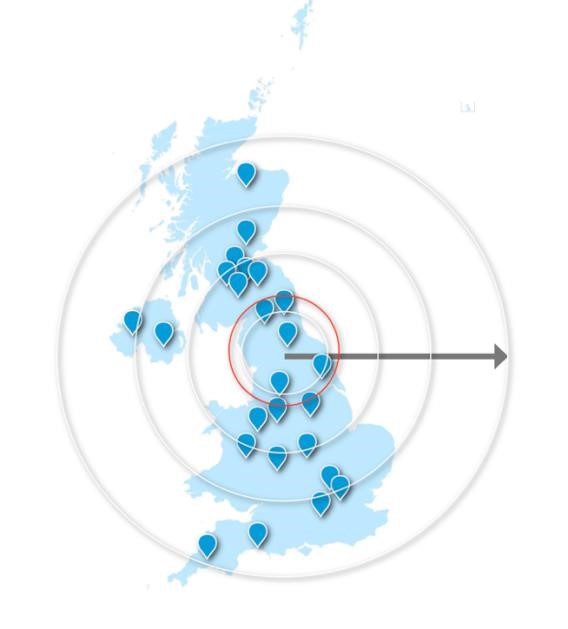
减少脚本的输出是一个 CSV 格式文件, 每行是一个独立的单个单词, 其坐标集合如图7所示。因此, 此文件可用于计算每个单词分布的中心点, 数据清理过程已完成。



**图 7: 减少脚本的结果**

### 3.1。3 数据可视化

在迭代一中, 将该项目的数据可视化分为两部分。其中之一是地理地图数据可视化, 另一个是包含特定单词的商店数量随着距离 (米) 与中心点的距离 (米) 的增加而增加的趋势。该项目希望通过这两种数据可视化结果的结果, 找到区域词汇的特征。就地理地图而言, 该项目认为这是最直观的方式, 可以看到一个词是否是一个区域词。该项目之所以想想象趋势, 是因为趋势图是项目对具有区域特征的单词分布趋势的猜测。 图8和图9是项目中区域词和非区域词的猜想图。黑色箭头指示与中心点的距离。当与中心点的距离增加时, 圆圈指示单词的分布范围。这两个图形中的红色圆圈具有相同的大小。蓝色指针代表单词分布的城市, 在每个城市中, 单词可能分布在许多商店中。该项目认为, 区域文字可能会聚集在接近其质心的地方。因此, 如果一个词是一个区域词, 随着与中心点的距离越来越大, 商店的数量就会增加到一定的价值, 然后就不再增长。此外, 当距离中心点的距离开始增加时, 区域文字的商店数量可能会迅速增加, 随着距离的增加到一定的距离, 商店数量的增长也会放缓。但是, 如果一个单词是非区域词, 则随着距离的增加, 商店的数量将继续增加, 直到距离增加到很大的距离。此外, 当距离没有增加多少 (例如图9中箭头距离的三分之一) 时, 非区域词的商店数量就不会增加到相对于商店总数的很多。因此, 该项目认为, "比率" 是指距离中心点与总商店数量除以中心点的距离小于特定距离 (图8和图9中红色圆圈的半径) 的商店数量, 可被视为区域的一个特征字。



### 图 8: 图9的猜想图: 非区域区域分布词分布词的猜想图

在可视化地理图之前, 首先, 根据单词的坐标集结果, 由项目计算单词分布的中心点。其次, 计算了单词分布的半径。计算半径的方法是计算中心点所有单词坐标之间的距离, 并找到最大的距离作为半径。中心点和每个坐标点之间的距离来自欧几里得距离, 因为底图将地球球体转换为平面。接下来, 由项目计算出全市包含这个词的商铺数量, 然后计算这些商铺在全市商铺总数中的比例。这样, 地理地图不仅可以显示单词的分布, 还可以显示每个城市商铺数量的不均匀分布。这可以帮助用户更好地了解单词分发的详细信息。

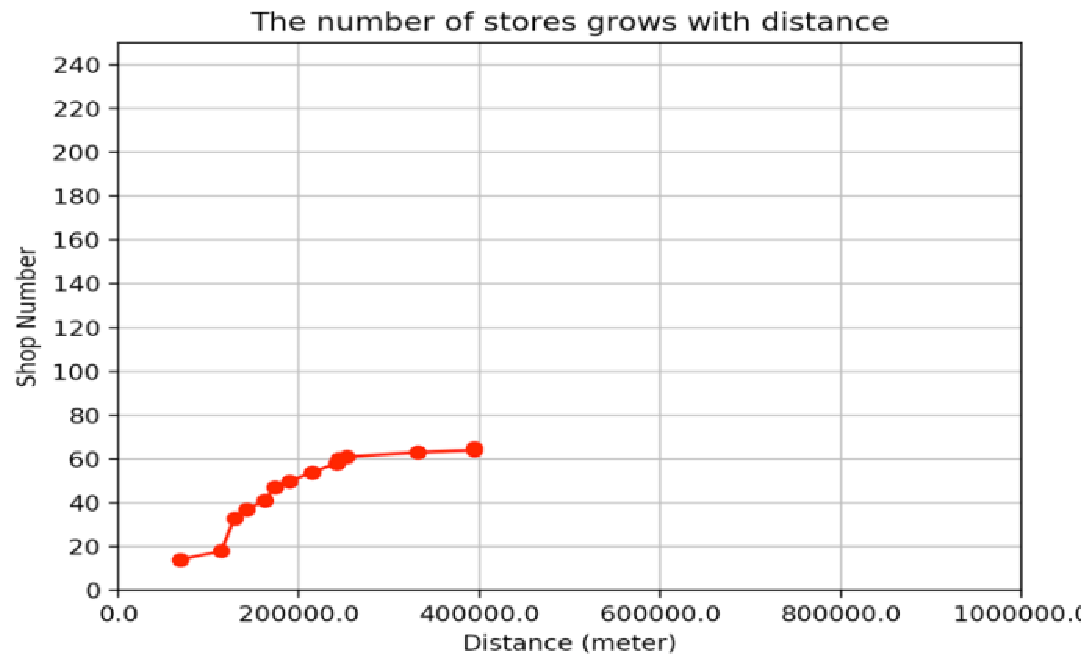
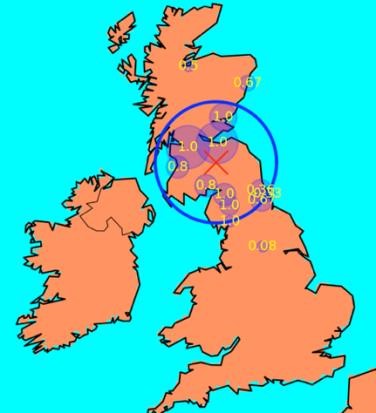
然而, 有一个问题, 一些异常值, 这意味着很少的商店远离中心点将对上述参数产生巨大的影响, 特别是在半径, 半径将变得非常大, 因为它涵盖所有的商店。例如, 如果一个词分布在英国北部, 但在英国最南端有一两家商店, 中心点将被南移。此外, 这个词的地理地图可视化结果将有很大的分布范围, 因为它包含了这个词的最南端的商店。这将给开发人员观察单词分布范围带来不便, 并影响对功能的探索。此外, 中心点的大偏差将导致该词重要特征的偏差, 因为大多数区域特征来自中心点 (详见3.2 和 3.3)。

为了过滤离群商店, 该项目首先计算包含单词的商店的中心点, 然后对所有商店的距离进行排序。接下来, 项目设置一个百分比, 即项目只占用接近中心点的商店的百分比, 然后重新计算中心点和其他参数。该项目使用 "haggis" 和 "兄弟" 作为例 (该项目事先知道这两个词是区域单词) 调整过滤百分比, 以观察分布范围的变化, 并最终决定保留95% 的商店。该项目发现, 选择 9 5% 的店铺可以过滤掉几乎所有对结果有巨大影响的异常值。此外, 这一百分比可以保留与中心点距离在合理范围内的所有正常分销店。图5是 "哈吉斯" 分布的一个例子, 其中包含95% 的商店。在图5中, "X" 表示 "哈吉斯" 分布的中心点。粗体蓝线表示分布的周长。图中许多蓝色小圆圈的大小代表了每个城市中包含 "哈格斯" 的商店数量。每个蓝色圆圈中的小数是指城市中包含单词的商店数量占城市商店总数的百分比。

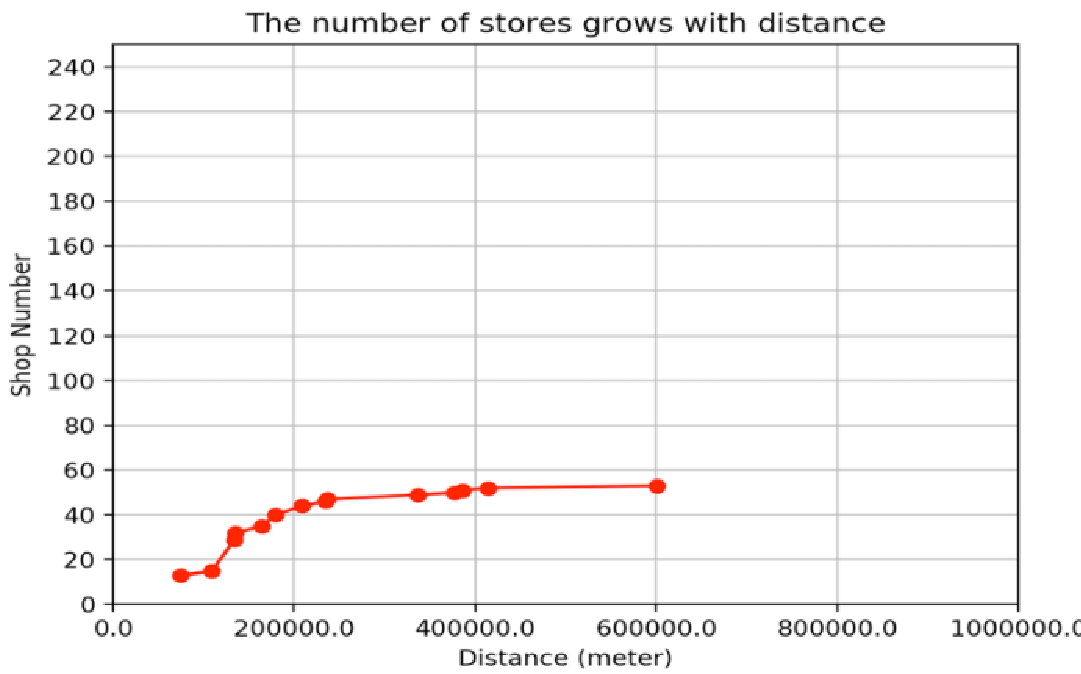
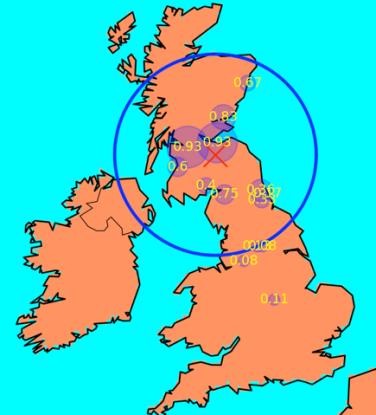
## 3。2 发现

在迭代一中, 该项目通过将已知区域词的结果与预先已知的非区域词的结果进行比较, 获得了一些区域词的特征。下图显示了地理地图可视化以及区域文字和非区域文字的趋势可视化结果。

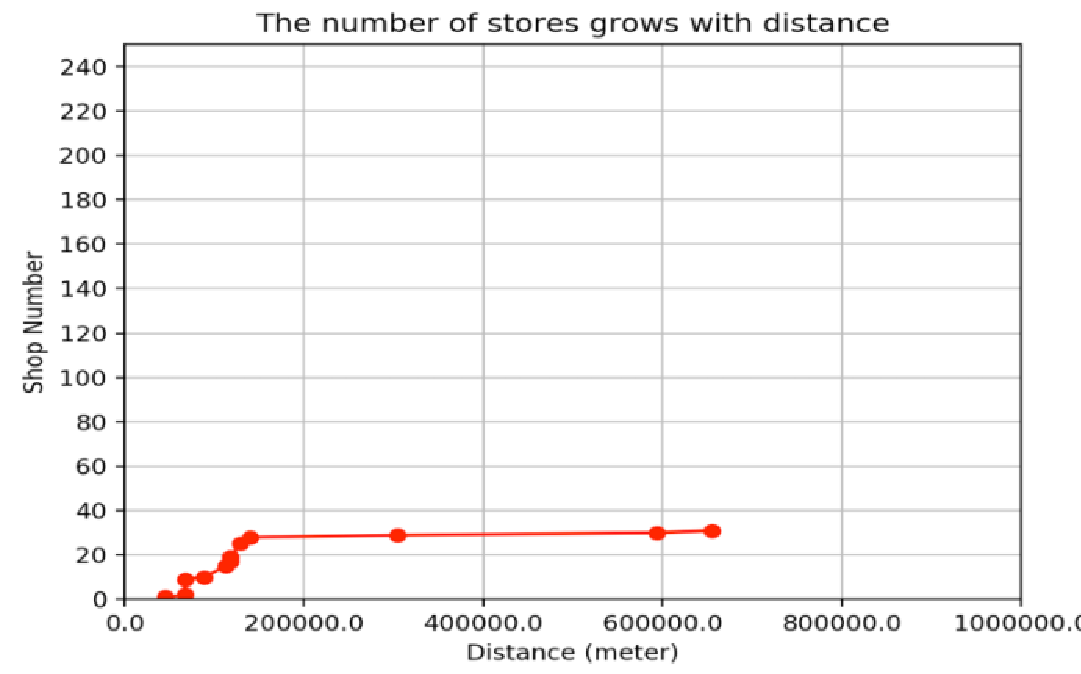
### 3.2。1 区域文字调查结果



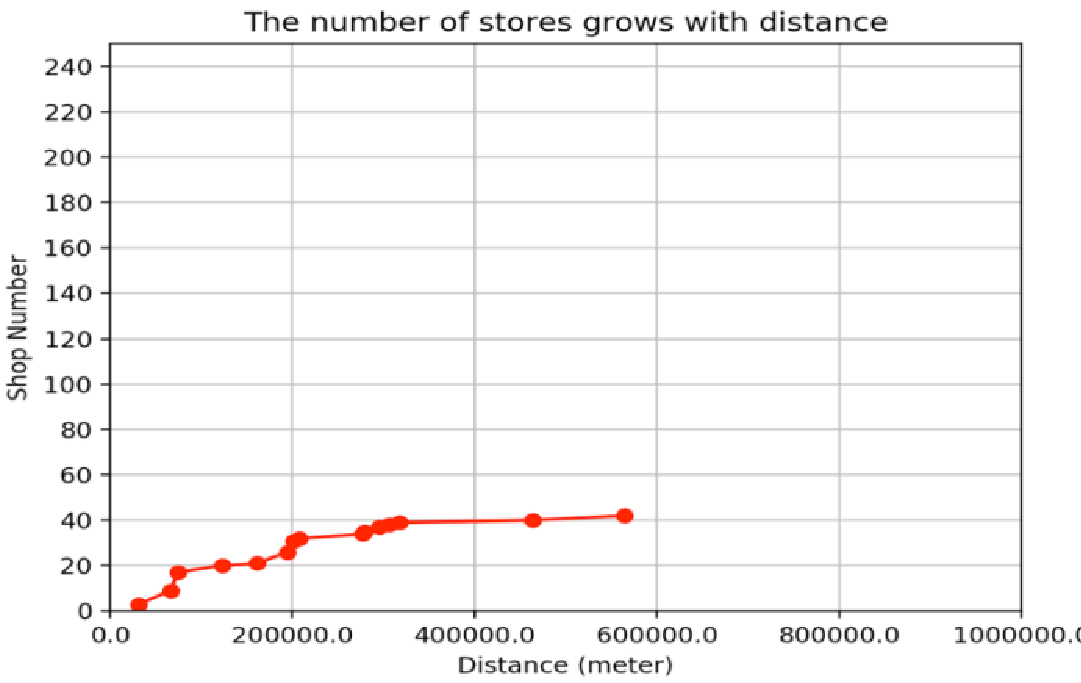
### 图 10: "哈吉斯" 图 11: "haggis" 商店的数量分布不同 (95%) 与距离



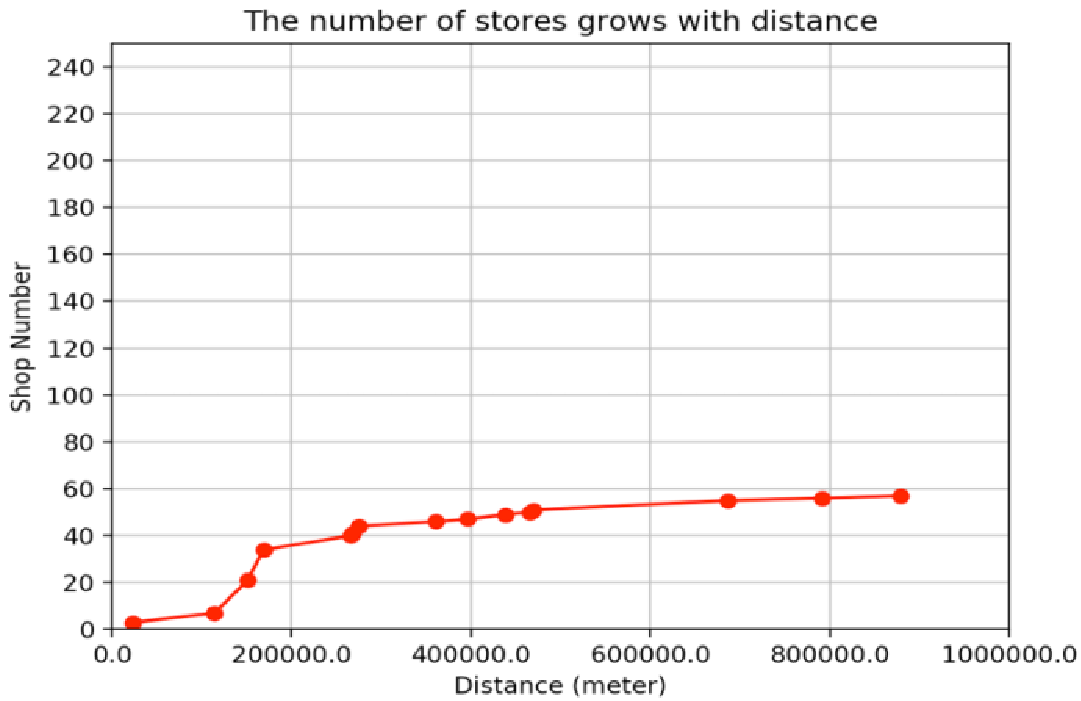
### 图 12: "兄弟" 图 13: "兄弟" 商店的数量因分布而异 (95%) 距离



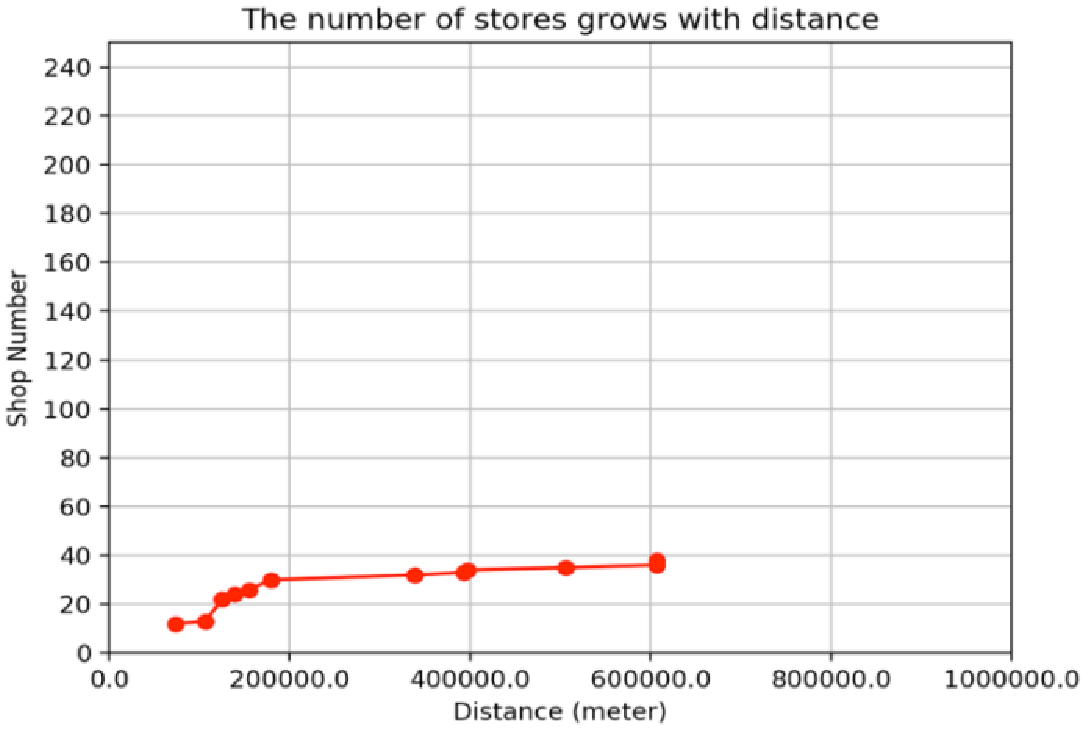
### 图 14: "naan" 图 15: "naan" 商店的数量分布不同 (95%) 与距离



### 图 16: "roe" 图 17: "roe" 商店的数量因分布而异 (95%) 距离

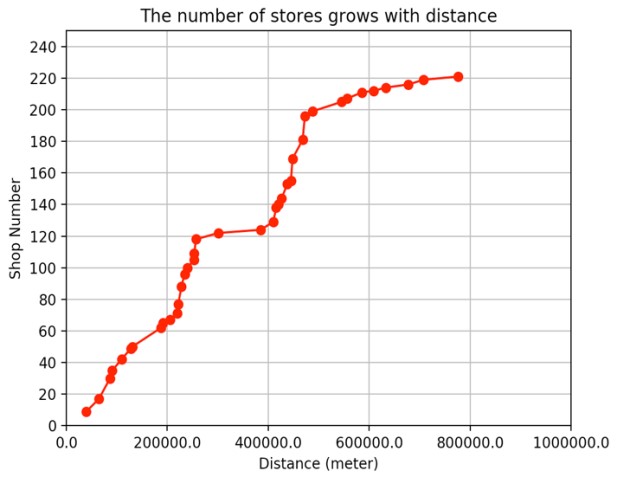


### 图 18: "晚餐" 图 19: "晚餐" 店的数量分布不同 (95%) 与距离



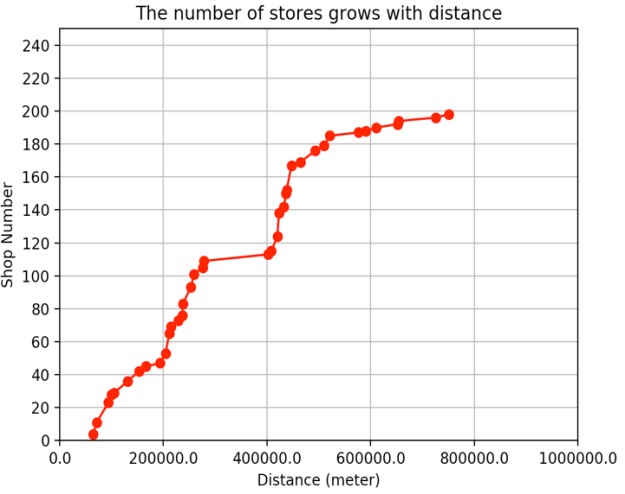
**图 20: "pakora" 图 21: "pakora" 商店的数量分布不同 (95%) 与距离**

### 3.2。2 非区域性词语调查结果



**图 22: "芯片" 图 23: "芯片" 商店的数量因**

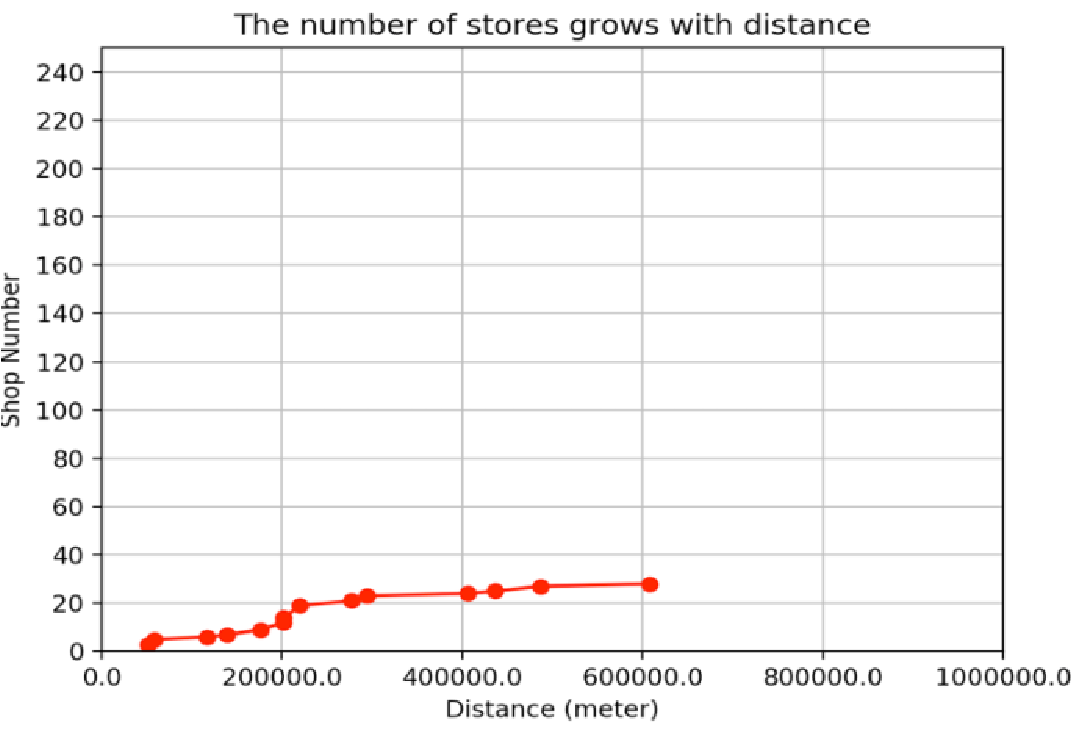
### 分配 (95%) 距离



### 图 24: "香肠" 图 25: "香肠" 店的数量分布不同 (95%) 与距离



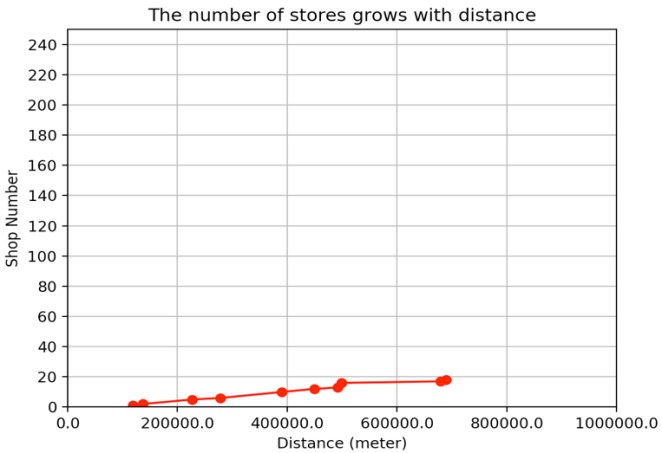
### 图 26: "至尊" 图 27: "至尊" 商店的数量分布不均 (95%) 与距离



### 图 28: "礼物" 图 29: "礼品店" 的数量因分布而异 (95%) 距离



### 图 30: "汤" 图 31: "汤" 店的数量分布不同 (95%) 与距离



### 图 32: "每日" 图 33: "每日" 商店的数量分布不同 (95%) 与距离

上图显示了三种类型的非区域词。图 22-图25显示了分布在每个城市几乎每个商店的非区域词汇的分布和趋势。

图26--图29显示了几乎分布在每个城市的单词, 但每个城市中包含这个词的商店数量并不多。

图 30-图33显示的单词不分布在许多城市, 但广泛分布在英国各地。

根据这些非区域性词汇的发现, 该项目发现, 当距离开始增长时, 商铺数量的增长不会太快。这更可以肯定的是, "比率" 是区域词语的一个特点, 而不是非区域词语。

## 3。3 评估和改进

通过对区域文字和非区域词的调查结果进行比较, 该项目发现, 距离中心点 2 0万米以内的地区文字店数量大幅增加。2 0万米后, 增长趋势呈现放缓。然而, 对于非区域性文字商店数量的趋势, 并没有这样的规则。因此, 这证实了该项目的假设, 即区域文字在一定距离内显示出快速增长趋势, 在这个项目中, 一定的距离是 2 0万米。

接下来, 该项目根据 2 0万米的距离门槛计算了上述区域词和非区域词的 "比率", 以下两个表格是 "比率" 的结果。

|  |  |
| --- | --- |
| **区域词语** | **比** |
| 哈吉斯 | 70% |
| 桥 | 75% |
| 纳安 | 90% |
| 净资产 收益 率 | 65% |
| 晚饭 | 60% |
| 帕科拉 | 79% |

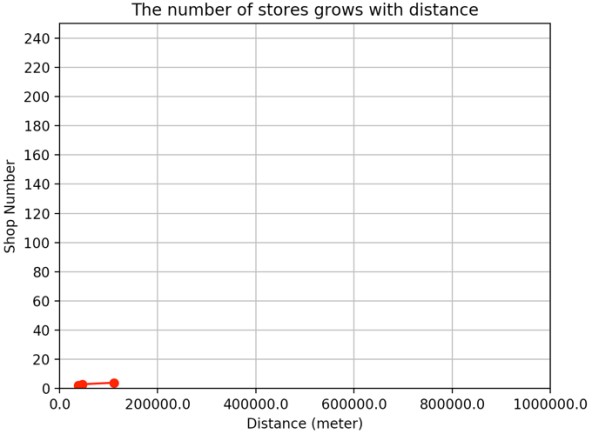
### 表 1: 带有 "比率" 的区域词语

|  |  |
| --- | --- |
| **非区域词语** | **比** |
| 芯片 | 29% |
| 香肠 | 24% |
| 最高 | 32% |
| 礼物 | 32% |
| 汤 | 32% |
| 日常 | 11% |

### 表 2: "比率" 的非区域词语

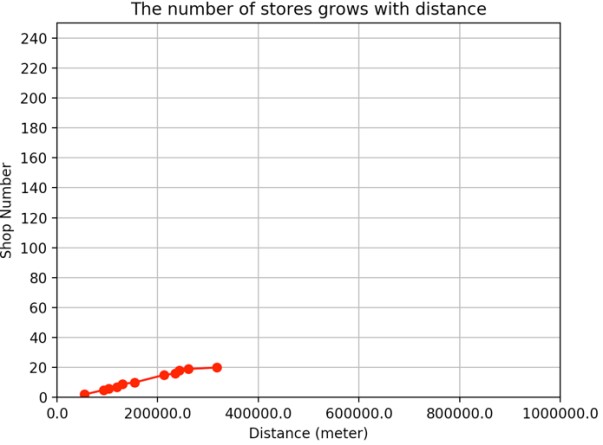
根据表1和表 2, 所有非区域词语的 "比率" 很低。然而, 从地域文字的结果来看, "晚餐" 的 "比例" 只有 "6 0%", 虽然 "晚餐" 在爱丁堡附近密集分布, 但在许多其他城市广泛分布。因此, 该项目对 "晚餐" 的地域性提出了质疑, 该项目不想将 6 0% 作为 "比例" 的门槛。从 "罗伊" 的分布来看, "罗伊" 相对于 "晚餐" 表现出很强的地域性, 因此, 该项目将 "罗伊" 的门槛设定为 6 5%。

然而, 有一个问题是, 几乎没有商店 (不到十家商店) 的单词大多分布在 2 0万米以内。此外, 这些单词中许多的 "比率" 是1。例如, 图34和图35显示了仅有四个商店的 "m夸拉" 的分布和趋势, "比率" 为1。在图34中, "马萨拉" 看起来像一个区域词, 根据图 35, "马萨拉" 具有区域词的特征。然而, "马萨拉" 的分布样本实在太小, 项目无法直接确定 "马萨拉" 是一个区域性词。因此, 该项目决定, 对于只包含十家或十家以下商店的词语, 该项目将其视为非区域词语。



### 图 34: "马萨拉" 分布图 35: "马萨拉" 商店的数量 (95%) 随距离的变化

通过以上发现, 可以解释, "比率" 65% 可以说是区域词的特征之一, 但仅仅依靠 "比率" 来判断所有的区域词, 如 "funghi", 是不够的。根据图36和图37是 "funghi" 的分布和趋势, 该项目发现, 含有 "funghi" 的商店数量上升顺利, "funghi" 的 "比率" 仅为50%。然而, 如图36所示, 有许多商店在一个小区域内包含这个词, 它看起来像一个区域词。因此, 如果 ' funghi ' 只以 ' 比率 ' 来判断, 就必须将其视为一个非区域性的词。因此, 该项目需要发现更多的功能, 以做出更准确的决策。



### 图 36: "funghi" 分布 图 37: "funghi" 商店的数量 (95%) 随距离的变化

根据 "funghi" 的结果, 该项目首先想到了计算商店的平均距离 (平均距离), 其中包含了从中心点的单词。这是因为与中心点的平均距离较小, 这意味着单词分布范围不会很大。区域和非区域词语结果的结果见下表。

|  |  |
| --- | --- |
| **区域词语** | **平均距离** |
| 哈吉斯 | 216926.14 |
| 桥 | 266813.33 |
| 纳安 | 214933.34 |
| 净资产 收益 率 | 239904.54 |
| 晚饭 | 387295.02 |
| 帕科拉 | 306999.72 |
| 蘑菇 | 176748.19 |

### 表 3: "平均距离" 的区域单词

|  |  |
| --- | --- |
| **非区域词语** | **平均距离** |
| 芯片 | 351005.86 |
| 香肠 | 353703.24 |
| 最高 | 308121.37 |
| 礼物 | 262870.84 |
| 汤 | 361040.31 |
| 日常 | 389090.17 |

### 表 4: "平均距离" 的非区域词语

根据表3和表 4, 大多数区域文字的 "平均距离" 小于 300, 000 米, 大多数非区域文字的平均距离大于 300 000 米。因此, 该项目最初决定使用 "平均距离" < 300 000 字作为区域文字的一个特点。

在项目的区域特色探索过程中, 项目的灵感来自于离中心点越近的现象, 店面分布就越密集。因此, 该项目决定使用包含单词的城市的中位距离, 试图发现新的特征。首先, 计算和排序的距离, 从中心点的所有城市包含该单词的项目。接下来, 中位距离

("中位") 被项目发现, 然后该项目发现距离低于中位的城市, 并计算了这些城市中包含的商店数量 ("num \_ less")。此外, 该项目还发现距离大于或等于中位距离的城市, 并计算了这些城市的商店数量 ("num \_ 上")。通过将区域词和非区域词分别比较 "中值"、"num \_ less" 和 "num \_ on", 该项目没有发现区分区域和非区域词的任何特征。然而, 该项目发现, "比例", 这意味着 ' num \_ less \_ 除以该词的总车间数量, 可以区分区域和非区域, 下表就是结果。

|  |  |
| --- | --- |
| **区域词语** | **比例** |
| 哈吉斯 | 7692% |
| 桥 | 83.0% |
| 纳安 | 74.19 |
| 净资产 收益 率 | 6780% |
| 晚饭 | 77.19 |
| 帕科拉 | 78.94 |
| 蘑菇 | 45% |

### 表 5: "比例" 的区域词语

|  |  |
| --- | --- |
| **非区域词语** | **比例** |
| 芯片 | 55.2% |
| 香肠 | 5505% |
| 最高 | 59.46 |
| 礼物 | 50.14% |
| 汤 | 42.10% |
| 日常 | 55.56 |

### 表 6: "比例" 的非区域词语

根据表5和表 6, 该项目发现, 除 "funghi" 外, 所有区域词的 "比例" 大于 67%, 所有非区域词的 "比例" 低于60%。因此, 该项目将 "比例" > 67% 界定为区域词的一个特点。

此外, 该项目还推测, 如果一个词分布在许多城市, 可能是一个区域词。为了验证这一想法, 该项目计算了所有单词的上述五个要素的值, 并将其存储在 CSV 格式文件中 (单词、比率、比例、平均距离、城市编号、商店数量)。通过这个文件, 该项目发现, 当一个词的 ' 城市号 ' > 19, 该项目可以直接得出结论, 它是一个非区域性的词, 因为 ' 城市数字 ' > 19 的单词有低 ' 比例 ', 低 ' 比例 ' 和大 ' 平均距离 '。然而, 分布在18个城市的 "糊状" 显示了地域性。因此, 该项目将 "城市编号" < 19 定义为区域词的特点之一。

## 3。4 摘要和今后的工作

通过观察和比较地理地图和已知的区域和非区域词的趋势, 迭代1获得了五个特征 (' 比率 ' > 65%, ' 商店数量 ' > 10, ' 平均距离 ' ' 300, 000 米 ', ' 比例 ' 67, ' 城市编号 ' < 19)。区域文字。该项目希望利用这些功能对所有分离的独立词语做出区域判断。但是, 项目不能在一个有条件的语句中使用所有这些功能来判断区域单词, 因为某些单词 (如 "m夸拉") 只满足其中的一些特征。因此, 项目只能通过一次选择一个要素将数据集划分为两个部分, 然后将划分的结果通过另一个要素进行划分。然而, 该项目无法判断每个划分是否为最佳划分, 这意味着目前选定的特征可以最大限度地区分区域词和非区域词。幸运的是, 决策树中的 ID3 算法可以帮助项目解决 ID3 算法利用熵选择特征以实现最优划分的问题。因此, 在下一次迭代中, 项目将使用决策树使用这些功能来获得区域单词的结果。

**第4章**

# 迭代2

迭代二的目的是利用迭代一中发现的决策树和特征对独立的单字进行分类。在此迭代中, 项目将完成以下任务: 生成决策树的训练集, 使用 ID3 算法生成决策树, 评估 ID3 算法和 ID3 决策树的区域结果。此外, 在对 ID3 算法进行评估后, 该迭代引入了另一种决策树算法 (购物车算法), 并对算法和区域结果进行了评估, 并对下一次迭代提出了建议。

## 4。1 方法

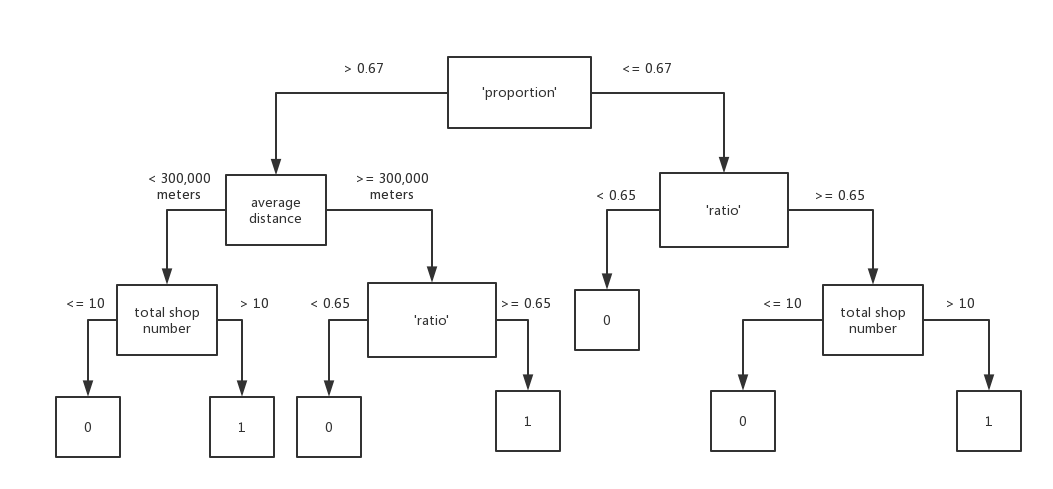
在本节中, 该项目重点介绍了训练数据集和 ID3 算法的生成过程。

### 4.1。1 培训数据集

培训数据集是由项目生成的, 其中包含一些已知的区域单词和非区域单词, 该项目最初希望使用总数据的 20% (5289 字) 作为培训数据集。根据迭代一产生的 "带功能的单词" 文件, 该项目发现, 在这些词的商店数量中, 只有五分之一的单词超过十个, 在这些单词中, 大多数单词显示出明显的非区域特征。例如, 非区域性的单词 "饮料" 和 "bean" 的 "比率" 较低,

"比例", 和非常大的 "平均距离"。因此, 找到足够数量的非区域词作为培训数据是很容易的。但是, 通过观察 "带功能的单词" 文件中的数据集, 很难找到足够数量的区域单词。原因是, 查找区域文字不仅需要观察地理地图的可视化结果, 还受到开发人员主观标准的影响。在首次生成训练集的过程中, 开发者的主观标准成为选择区域词的最大障碍, 因为英语中常用的单词很多, 比如 "安全" 和 "即时"一些地区特色。虽然这些词虽然有一些区域词的特点, 但项目并不具备所有的功能。因此, 该项目没有将这些单词及其特征添加到培训数据集中。最终, 该项目刚刚定义了 2 5个地区单词, 其中大部分是菜名, 还有一些是 "约克郡" 等地名。此外, 训练数据集中还增加了70个非区域词, 然后该项目利用该数据集生成 ID3 算法决策树。

## 4。2 发现



### 图38:id3 算法决策树

图38是 ID3 算法的决策树结果, 其中 "0" 表示非区域单词, "1" 表示区域单词。有54个以下独立词被判断为区域词。用这些话说, "哈吉"、"irn"、"b鲁"、"pakora"、"pakora"、"roe"、"yorkshire"、"kiev"、"地狱"、"紧缩"、"通心粉" naan ', "pattie", "膝盖", "意大利面", "balty", "haagen", "dazs", "cob", "parmesan", "plaice", "咸味", "圆马", "臀部"在训练数据集中使用。"溜冰", "博洛尼亚", "汉堡包", "肋骨", "卡纳拉",

"肉质"、"黄瓜"、"番石榴"、"帕特"、"牛角"、"飞溅"、"蒲公英"、"扇贝"、"扇贝"、"凯马 ', ' 萨莫萨 ', ' 给 ', ' 烟熏 ', ' 科尼什 '," 位 "," 四兆 "," 脸谱 "," 选择 "," 价值 "," 安全 "," 即时 "," 摇滚 "," 射击 "," 公牛 "是新的成立。

## 4。3 评估和改进

虽然这个项目得到了区域词, 但除了菜词, 这些词很多还在英语中常用, 比如 ' 位 '、' 给 '、' 值 '。为了理解为什么这些词被判断为区域词, 项目决定找到这些词的背景, 找出这些词在网站上的用法。为了实现这一目标, 该项目编写了一个 Python 脚本, 其输入是这些区域单词。此脚本负责在迭代一中获得的所有 HTML 文件中搜索这些区域单词的上下文, 并生成这些单词及其上下文的文件。因此, 该项目可以根据这个文件找到为什么这些词被判断为区域词的原因。

根据 "上下文" 一词, 该项目发现, 以下词语代表一种菜或菜名的一部分: "哈吉斯"、"irn"、"b鲁"、"kiev"、"地狱",

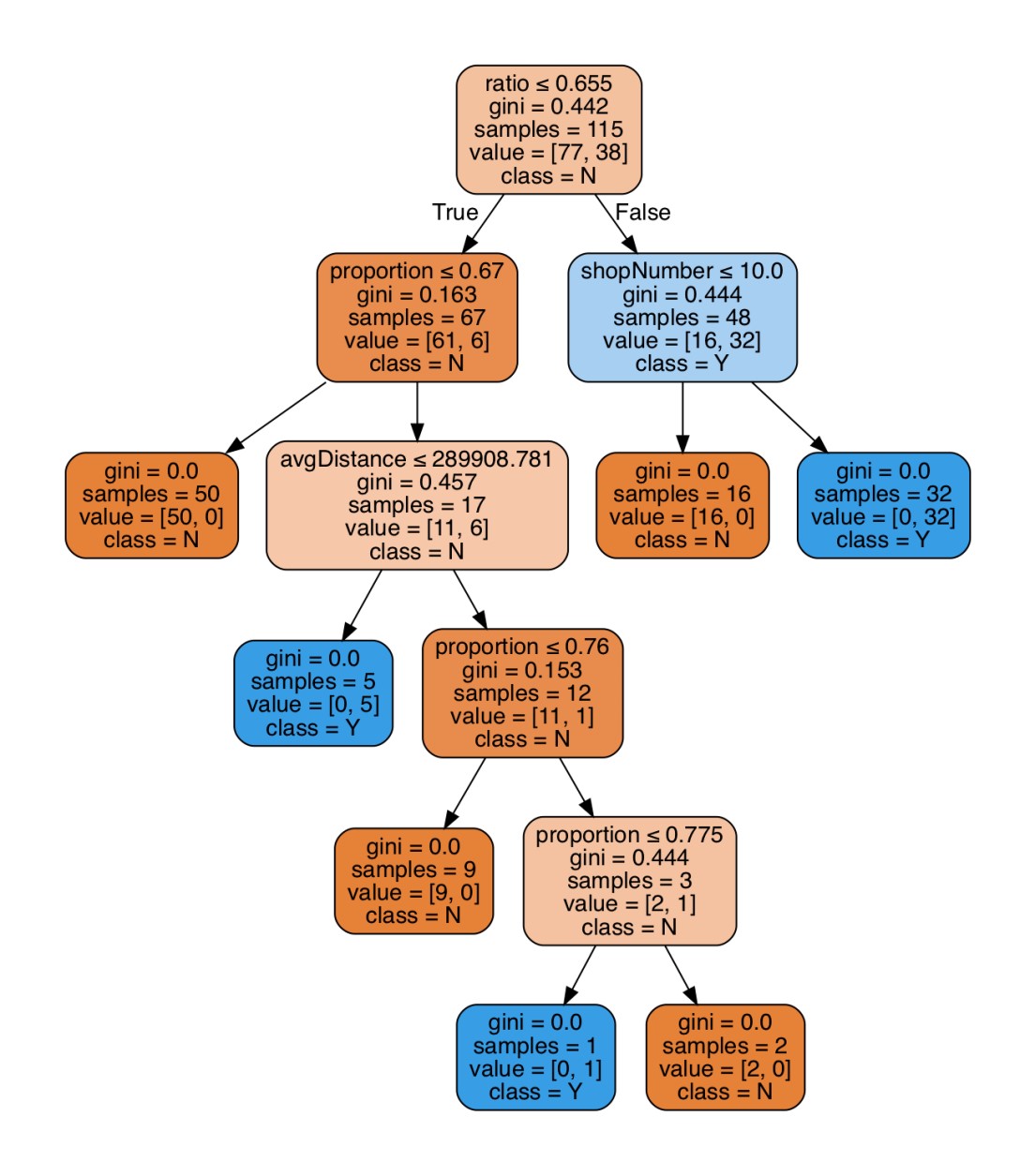
"紧缩", "溜冰", "博洛尼亚", "通心粉", "naan", "汉堡包", "辫子", "肋骨", "肾", "意大利面", "意大利面", "" 碳酸酯 "," 面糊 "," 酸 "," 肉 "," 黄瓜 "," 椰子 "," 番石榴 "," pakora "," pukka "," 咸味 "," pattie "," burdock "," parmesan "," parmesan "," 帕玛森 "," 帕梅桑 "," 帕梅桑 "," 肉食 "," 肉食 "," 肉食 "," 肉 "," 肉 "," 肉 "," 肉食 "," 帕梅桑 "," 肉食 "," 肉食 "," 肉食 "," 肉食 "," 肉食 "," 帕梅桑 "," 帕玛坦 "," 肉无 "," 肉酸 "," 肉食 "," 肉酸 "," 肉食 "," 帕佩坦 "," 帕佩桑 "," 帕佩森 "," 肉用 "," 肉用 "," 帕玛蒂 ",飞溅 "," 蒲公英 "," 扇贝 "," 凯马 "," 萨莫萨 "," 粗糙 "," 臀部 "," das "," 岩石 "," 射击 "," haagen "," 公牛 ",

"通心粉", "科尼什", "夸特特罗", "激情"。值得一提的是, 在这些地区的词汇中, "伊恩兄弟" 总是一起使用, 代表着苏格兰的饮料;"哈根" 和 "大" 代表地区词汇, 因为在这方面, 哈根达斯可能与商家有更多的贸易联系;"摇滚" 总是和 "鳗鱼" 一起使用。"石鳗鱼" 代表一种鱼;"射门" 总是与 "热" 和 "热射门" 一起使用, 代表一种菜;"公牛" 总是与 "红" 和 "红牛" 一起使用, 是一种饮料, "公牛" 在区域分布的原因可能与 "黑根" 相同;"油" 总是与 "糊状" 和 "油松饼" 是一道菜;"通心粉" 总是与 "奶酪" 一起使用, 而 "通心粉奶酪" 则是一道菜;"夸特洛" 总是与 "stagioni" 一起使用, 而 "quattro stagioni" 是一种比萨饼;"激情" 代表水果或菜肴。也许在那个地区这种水果很有名或者卖得很好。

除了上面提到的区域菜词外, 英语中常用的一些词也被判断为区域词, 项目根据这些词的语境找到了这样做的原因。"安全" 主要与 "与" 和 "安全在线支付" 一起使用。此外, 该项目发现, 当 "安全" 与 "与" 一起使用时, 所有使用此用法的网站都具有相同的风格。同样, 所有使用 "安全在线支付" 的网站都有相同的风格。这可能是因为该地区商店的网站是由同一家公司开发的。因此, "安全" 出现在区域内。同样, "选择" (总是与 "味道" 一起使用) 和 "即时" (总是与 "芯片店外卖-在线即时订购") 被判断为区域词语的原因与 "安全" 相同。就 "约克郡" 而言, 大多数 "约克郡" 代表的是一个名为 "约克郡" 的地方。因此, ' 约克郡 ' 是一个代表地名的地区词。"给予" 被用作动词。也许在那个地区, 人们习惯了用这种方式来表达自己的菜, 比如给人最好的味道。"烟熏" 被用作形容词, 通常与 "bbq" 或 "香肠" 结合使用。"位" 通常用作程度副词。"值" 总是与代表菜肴的 "盒子" 或 "餐食" 一起使用。在 "脸谱" 方面, 该项目发现, 一些网站提供脸谱账户, 包含 "脸谱" 的商店分布在一个小区域。

根据 ID3 算法的结果, 虽然该项目根据区域特征成功地确定了 "地域性" 一词, 但也发现了一个问题。这个问题是, 每个功能的阈值可能不准确, 因为这些阈值是由开发人员通过在迭代一中观察有限数量的数据来定义的。因此, 项目希望使用算法自动定义要素阈值。但是, 项目不能使用 ID3 算法来查找特征阈值, 因为 ID3 算法的局限性在于它只能处理离散值 [40]。这意味着要素值必须根据数值变量进行分类, 并且项目必须将每个训练数据的特征标记为数字变量。例如, "haggis" 的 "平均距离" 为 216 926 米, 该项目将其标记为 "< 300, 000"。幸运的是, Python 提供了一个工具包 (Sklearn-war判别), 它集成了各种机器学习算法, 用于监视和非监督问题 [41]。该工具包可以帮助项目实现自动查找阈值的目标。因此, 该项目使用了 Sklearn-learn 学习包决策树算法, 该算法使用一种优化的 Cart 算法 [42] 生成决策树, 包括分类树和回归树。在本项目中, 分类树更合适, 因为决策树的目标是二进制的。

### 4.3。1 购物车算法的结果和评估



### 图39:cart 算法决策树

图39是使用购物车算法的决策树结果。在生成购物车算法决策树之前, 该项目根据 ID3 算法的结果, 在训练数据集中添加了一些区域词。虽然特征阈值不准确, 但通过项目验证了 ID3 算法分类的结果是区域性的。因此, 该项目从以前分类的区域结果中增加了一些新的区域词语。这样做的目的是为了得到一个能够更准确地分类区域单词的树。

从购物车算法的分类结果来看, 虽然购物车算法生成了更多的树分支, 分支条件变得更加精确, 但分类结果与 ID3 算法差别不大。在这56个单词中, 有56个单词被判断为区域词, 除了 "薄荷", 有55个是 ID3 算法的结果。"薄荷" 这个词在爱丁堡附近广泛分布。

然而, 在使用 ID3 算法时, 由于其 "比例" 大于 67%, 但其 "比例" 低于 65%, 因此被判断为非区域性词。

## 4。4 摘要和今后的工作

在这个迭代中, 项目首先获得了区域词。此外, 该项目还了解了为什么根据单词出现的背景将这个词判断为一个区域词的原因。此外, 该项目还发现, "城市号" 对决策树没有任何影响, 因此 "城市号" 对项目来说是一个无用的功能。不过, 该项目希望继续探讨每一个分类结果的可能性, 比如 "哈格斯" 被归类为区域内容的可能性。遗憾的是, 项目无法通过使用决策树获得概率, 因为在此项目中, 决策树的结果是二进制的。因此, 在下一次迭代中, 项目希望使用 Sklearn-learn 学习包的回归分类器来获取特定内容在类别中的概率。

**第5章**

# 迭代3

迭代三的目的是使用逻辑回归来寻找一个词被判断为一个区域词的概率。在迭代三过程中, 项目还发现了其他有价值的信息, 如每个特征对分类结果的影响、使用 L1 和 L2 对分类结果的影响, 以及一些新发现的区域词。迭代三分为四个部分: 方法论、研究结果、评价和总结以及今后的工作。方法部分的重点是描述训练数据集的改进和特征选择的变化。研究结果部分主要讨论了逻辑回归模型的结果和概率结果。评价部分的重点是评价结果。总结和未来的工作部分负责总结迭代三, 并安排迭代四的计划。

## 5。1 方法

该项目编写了一个 Python 脚本来实现逻辑回归。回归中使用的训练数据集是一个改进的数据集, 它在决策树中添加了更多的结果。此外, 该项目还使用了 "城市号" 和 "店号" 以外的功能进行逻辑回归。考虑到 "城市号", 决策树已经证实, 它对分类没有影响。就 "商店编号" 而言, 不将其用作功能的原因是项目的第一个逻辑回归分类结果。对于第一个分类, 该项目使用了除 "城市号" 以外的所有其他功能, 发现逻辑回归不能将 "商店编号" 小于10的单词直接分类为非区域单词, 如决策树。相反, 逻辑回归将这些词的区域概率设置得非常高, 这意味着这些词成为区域词的概率很高。在逻辑回归中, 一个词属于一个类别的概率会受到要素权重的影响。在这个项目中, "商店数量" 的权重小于其他功能, 几乎接近零。因此, 拥有十家或十家以下商店的词可能很有可能被赋予。为了筛选 "商店号" 小于10的单词, 项目在脚本中筛选这些单词, 而不将其用作生成逻辑回归模型的培训集。此外, 该项目也没有对不到十家店铺的文字做出区域判断, 因为该项目只关心区域文字, 而这些文字的过滤对区域文字没有影响。最后, 该项目决定用不到 1 0家店铺的文字进行过滤, 而不是将 "店号" 作为功能。

该项目最初希望使用逻辑回归来获得将单词判断为区域词的概率。但是, 在分类过程中发现了其他一些发现, 例如使用 L1 和 L2 规则化之间的差异。

## 5。2 发现

下表使用 "比例"、"比例" 和 "平均距离" 作为特征生成。它比较了使用 L1 和 L2 规则化的差异。分类率是指训练数据集中的数据量, 由模型正确分类, 除以训练集总数据量。百分比阈值意味着, 如果一个词的区域概率高于这个阈值, 这个词将被判断为一个区域单词, 它是通过观察概率结果得到的。该项目每10% 间隔观察一次, 从高概率到低概率。如果间隔中有五个以上的区域单词 (在第二次迭代中决定), 则此间隔的下限将被视为阈值。

|  |  |  |
| --- | --- | --- |
| **正规化** | **L1** | **L2** |
| 分类率 | 83.13 | 53.57% |
| 百分比门槛值 | 50% | 无法分辨 |
| 系数  ("比例"、"比例" 和  "平均距离") | 1.50840108 和 + 00  4.48321094 和 + 00  -1.5 15233889 E-05 | 3.44111057-12  8.66 3444565 e-12  -1.7 7054477 E-06 |
| 区域单词的数量 | 59 | 无法分辨 |
| 有概率的区域词 | 见附录 A-a.1 中的区域词语完整清单 | 毫无 意义 |

### 表 7: 正规化的比较 独立的选择 选择 "比例"、"比例" 和 "平均距离" 作为特征时的单词

单词 ' 概率 (大于 50%) 的区域结果使用 L1 正则化时显示在附录 A.1 中。与决策树的结果相比, 有些词被新定义为区域词 (概率大于 5 0%)。

它们是 "高级"、"垃圾邮件"、"希腊"、"电话"、"小食"、"服务"、"饼干"、"农场"、"下"。然而, 决策树中定义的六个区域词的概率较低, 如 "博洛尼亚"、"斑块"、"肾"、"臀部"、"公牛"、"薄荷"。

遗憾的是, 当使用 L2 正则化时, 所有单词的区域概率都小于 50%, 这意味着逻辑模型无法判断每个单词的区域性。因此, 在表7中, 项目无法说明门槛值和区域文字数。

## 5。3 评价

根据表7的结果, 使用不同的调整导致分类率的差异非常大。在使用 L1 正则化时, L1 正则化降低了对分类影响较小的特征的权重。通过系数的结果, 项目知道, 在项目选择的三个特征中, "比率" 对分类结果的影响最大, 其次是 "平均距离" 和 "比例"。"平均距离" 系数关闭为零的原因并不是因为 L1 正则化认为此属性不重要, 并将其减少到零。这是因为 "平均距离" 的值很大, 如 300, 000, 因此该模型降低了 "平均距离" 系数, 使逻辑回归模型的概率输出为零到1。此外, 将逻辑回归模型视为 "平均距离" 的一个重要特征。

该项目推测, 出现在新的地区词汇之上的原因是 "平均距离"。为了核实这种猜测, 该项目观察了这些新的区域词的特点, 发现这些词的 "平均距离" 很小, "比例" 和 "比例" 很低。因此, 在决策树中, 由于 "比例" 和 "比例" 较低, 这些词被判断为非区域词。然而, 在逻辑回归模型中, 这些词被判断为区域词, 因为它们的 "平均距离" 很小, 这意味着它们的店铺分布在小区域。从决策树判断为区域词但逻辑回归模型将这些词分为两类进行分析的概率较低的词。第一类包括 "博洛尼亚"、"斑块"、"肾" 和 "薄荷"。这些词的 "比例" 很低, "平均距离" 也不是很小, 但 "比例" 很高。然而, 在逻辑回归模型中, "比例" 对结果的影响最小。因此, 这些词的概率不到 5 0%, 但接近 5 0%。第二类包括 "公牛" 和 "臀部"。它们具有较高的 "比率" 和较高的 "概率", 但非常大的 "平均距离"。因此, 逻辑回归模型使他们的概率接近 50%, 但低于50%。

在使用 L2 正则化时, 逻辑回归模型将所有特征的系数降低到接近零。这意味着模型只是使用了 "平均距离" 功能来判断区域单词。该项目推测, L2 正则化希望适合所有这三个特征, 但它过度适应了 "平均距离" 的特征, 导致其他两个参数接近零。因此, 所有单词的区域概率都低于 5 0%, 模型无法做出决定。这就是为什么分类率在使用不同的规则时显示出很大的差异的原因。

该项目还发现了新出现的区域词汇的背景, 并发现了以下发现: "希腊" 总是与 "沙拉" 或 "比萨饼" 一起使用, 代表一种菜。几乎所有的 "垃圾邮件" 都与 "折腾" 一起使用, "垃圾邮件" 是一种菜。"溢价" 通常与三文鱼和鳕鱼等成分或沙司 (如沙司) 一起使用。"电话" 具有较高的区域概率的原因是, 该地区所有带有 "电话" 的商店网站都是由同一公司开发的, 每个网站都有 "欢迎电话订单"。"小食" 总是与 "盒子" 一起使用, "小食盒" 是苏格兰的一种快餐。"服务" 主要集中在英格兰中部地区, 经常与 "质量" 一起使用, 以显示良好的成分。这可能是区域用法。"饼干" 的区域性可能是由于该地区的饼干销售良好。所有的 "农场" 都与 "房子" 一起使用, "农舍" 是一个地区品牌的比萨饼。有许多用法的 "下", 但最出现的是 "12岁以下", 商店包含这种用法可以打折的人12岁以下, 他们集中在英格兰中部。

## 5。4 总结和今后的工作

迭代三得到了区域词的概率。此外, 通过逻辑回归模型, 该项目发现了一些新的区域词, 并分析了这些词被判断为区域词的原因。此外, 通过分析后勤模型系数, 项目知道 "比率" 对分类的影响最大, 其次是 "平均距离" 和 "比例"。此外, 该项目还推测了 L1 和 L2 分类率差异较大的原因。然而, 对于独立的单字, 有很多词表现出地域性, 因为它们与其他词一起使用。因此, 该项目决定使用其他类型的内容, 如名词短语和单词对, 以找到内容的区域性。

**第6章**

# 迭代4

此迭代的目的是使用名词短语和单词对作为数据集, 以查找内容区域性。本迭代主要分为两部分。第一个是名词短语部分, 第二个是单词对部分, 两者都包括方法论、结果和评价。在本章结尾处的总结中, 本项目比较了三种类型内容 (独立词、名词短语和单词对) 的分类结果。

## 6。1 名词短语

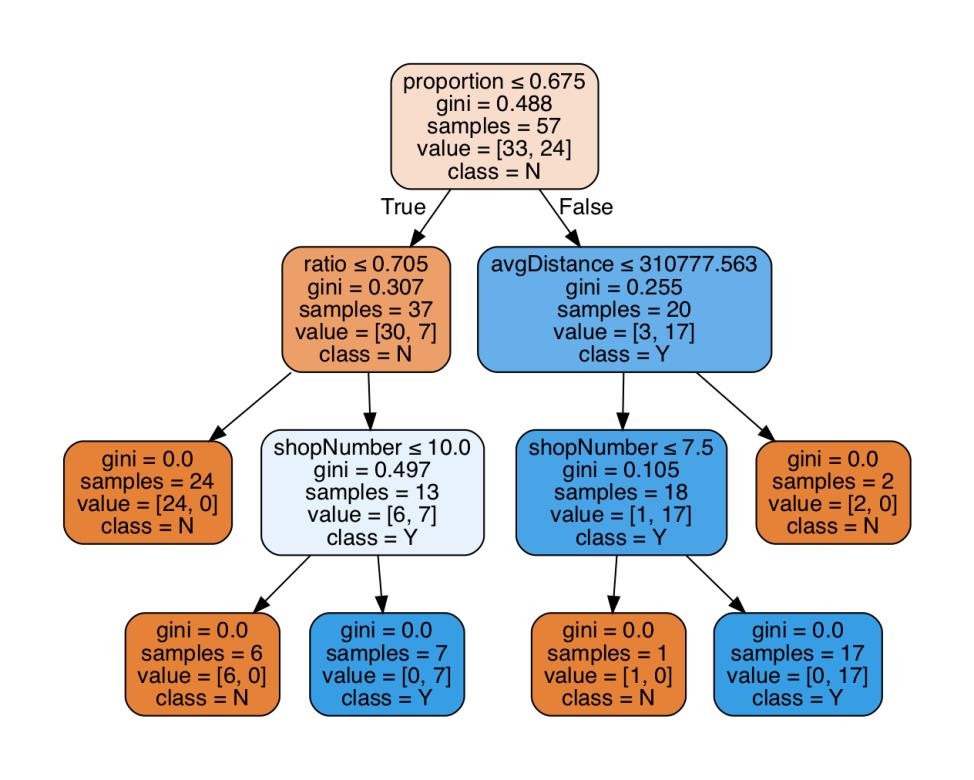
本部分主要介绍了名词短语分类结果的生成方法和评价结果。

### 6.1。1 名词短语方法

该项目首先需要从 HTML 文件中提取名词短语。用于实现此的方法与用于提取独立单字的方法相同。然而, 项目使用 Spasy 提取名词短语是不同的, Spasy 是一个开源 NLP 工具包 [43], 它封装了可以直接识别段落或句子中的名词短语的方法。此外, 在名词短语方面, 该项目不需要将名词复数转换为单数。具体而言, 该项目创建了一个新脚本, 该脚本是在原始脚本的基础上构建的, 用于提取独立单词, 并且项目对其进行了一些更改。在 Htmparser 识别 HTML 标记的内容并筛选特殊符号后, 该项目使用封装函数提取名词短语。然而, 该项目发现, 由 Spasy 提取的名词短语有时包含一些动词和形容词。因此, 该项目继续使用 Spasy 中的语音识别方法部分来筛选名词短语中的动词和形容词。例如, "大鸡肉汉堡" 和 "小鸡肉汉堡" 是相同的菜, 但不同的大小。因此, 该项目需要删除形容词 "大" 和 "小", 以确保这两个名词短语可以分为一个类别。脚本生成的名词短语既包含名词短语, 也包含独立词, 因为某些 HTML 标记内容是一个独立的词。接下来, 该项目使用与独立单词相同的步骤来生成区域结果。值得一提的是, 在生成名词短语训练数据集时, 该项目指的是独立词的结果。这意味着项目优先级从包含区域单词 (如 "饮食兄弟" 和 "蔬菜 pakora") 的短语中找到区域名词短语, 因为这些名词短语更有可能是区域性的。此外, 该项目还使用地理地图可视化结果和趋势图来验证上述名词短语的区域性。名词短语训练数据集包含24个区域名词短语和33个非区域名词短语。在项目为培训数据集做好准备后, 该项目首先使用 "比例"、"比例"、"平均距离"、"商店编号" 和 "城市编号" 作为特征, 以生成决策树。在发现 "城市号" 对于名词短语分类也是无用的 (详见 6.1.2) 后, 该项目采用了 "比例"、"比例" 和 "平均距离" 等特征, 得到了逻辑回归结果。

### 6.1。2 名词短语决策树发现

下图是名词短语决策树的结果, 它使用了卡特算法。



**图 40: 名词短语决策树**有52个名词短语被判断为区域短语:

"肾派", "鳕鱼", "帕特", "布鲁", "芝士汉堡四分之一磅", "蔬菜 pakora '," funghi "," 鸡肉 pakora "," haggis "," pagy "," 馅饼晚餐 "," 布丁晚餐 "," 通心粉奶酪 "," 地狱 "," 汉堡包 "," pakora "," 意大利面 "," 意大利面 ","公牛 "," 饮食兄弟 "," 苏伊卷 "," 罗 "," 薄荷 "在训练数据集中。

"米饭", "流行", "馅饼", "炸鱼片", "饼干", "晚饭", "鸡肉肉", "射击", "王肋", "面食", "pollo", "dandelion", "芝士汉堡晚餐", "香肠晚餐", "牛角饼", "位", "有点", "土豆派", "奶酪", "啤酒," 芯片卷 ","新发现了奶酪番茄、"披萨晚餐"、"鸡胸肉晚餐"、"芯片店外卖单"、"饮食可乐"、"鸡腿晚餐"、"菠萝圈"、"披萨紧缩"、"汉堡晚餐"。

### 6.1。3 名词短语决策树评估

根据名词短语的发现, 在独立区域词出现的语境中出现了 "蔬菜 pakora" 和 "鳕鱼" 等短语。然而, 独立词的结果中出现的 "摇滚"、"溜冰" 等词并没有在上述名词短语中被删除。这是因为这些独立词出现的短语有很多种, 每个名词短语的商店数量不到十个商店。因此, 这些短语被分为非区域短语。

例如, "摇滚" 出现在独立词的结果中, 经常被用作 "石鳗鱼", 但 "石鳗鱼" 的商店数量还不到十倍。因此, "摇滚" 消失在名词短语的结果中。此外, 还有一些新词被判断为 "流行"、"波波" 等地区词, 项目找到了出现的原因。以 "流行" 为例, "流行" 集中在英国中部 (曼彻斯特周边), 代表着一种饮料。然而, 它也出现在名词短语, 如 "瓶流行" 和 "流行百事可乐"。因此, 名词短语形式中 "流行" 的 "比例" 高于独立词形式中的 "流行"。因此, ' 流行 ' 满足了地区词的条件。就短语 "芯片店外卖订单" 而言, 它被判定为区域性短语的原因是包含此短语的网页具有相同的样式, 并且是由同一公司开发的。

### 6.1。4 名词短语逻辑回归结果

下表比较了名词短语逻辑回归模型的 L1 和 L2 正则。

|  |  |  |
| --- | --- | --- |
| **正规化** | **L1** | **L2** |
| 分类率 | 89.74。 | 89.74。 |
| 百分比门槛值 | 50% | 50% |
| 系数  ("比例"、"比例" 和  "平均距离") | 1.43506897 和 + 00  4.21097118 和 + 00  -1.42088335 E-05 | 1.27 526880 和 + 00  1.74 406725 和 + 00  -1.25081676 E-05 |
| 区域名词短语的数量 | 64 | 66 |
| 有概率的区域名词短语 | 参见附录 A-a.2 中区域名词短语的完整列表 | 参见附录 A-a.3 中区域名词短语的完整列表 |

**表 8: 正规化的比较** **名词短语的选择**

### 6.1。5 名词短语逻辑回归评估

根据表 8, 该项目发现, 就分类结果而言, 使用 L1 惩罚和 L2 规范化之间没有太大区别。在该项目分析了名词短语的训练数据集后, 该项目发现, 使用这三个特征中的任何一个 ("比率"、"比例" 和 "平均距离") 都可以成功地对大多数名词短语进行分类, 尽管 "比率" 对名词短语的影响仍然最大。结果。因此, 当使用 L2 正则化时, 这三个特征的系数几乎是相同的。

此外, 与决策树的结果相比, 在使用逻辑回归时, 出现了一些新的短语。该项目推测, 逻辑回归中出现新短语的原因是训练数据集不完善导致特征阈值不足。因此, 一些短语, 如 "垃圾邮件抖动", 其 "比例" 和 "比例" 是低, 但它也有低的 "平均距离" 被判断为非区域短语的名词短语决策树。此外, 该项目还发现, 这些名词短语被判断为区域短语的原因与上述原因相同 (包括区域词的原因和区域名词短语的原因)。

## 6。2 单词对

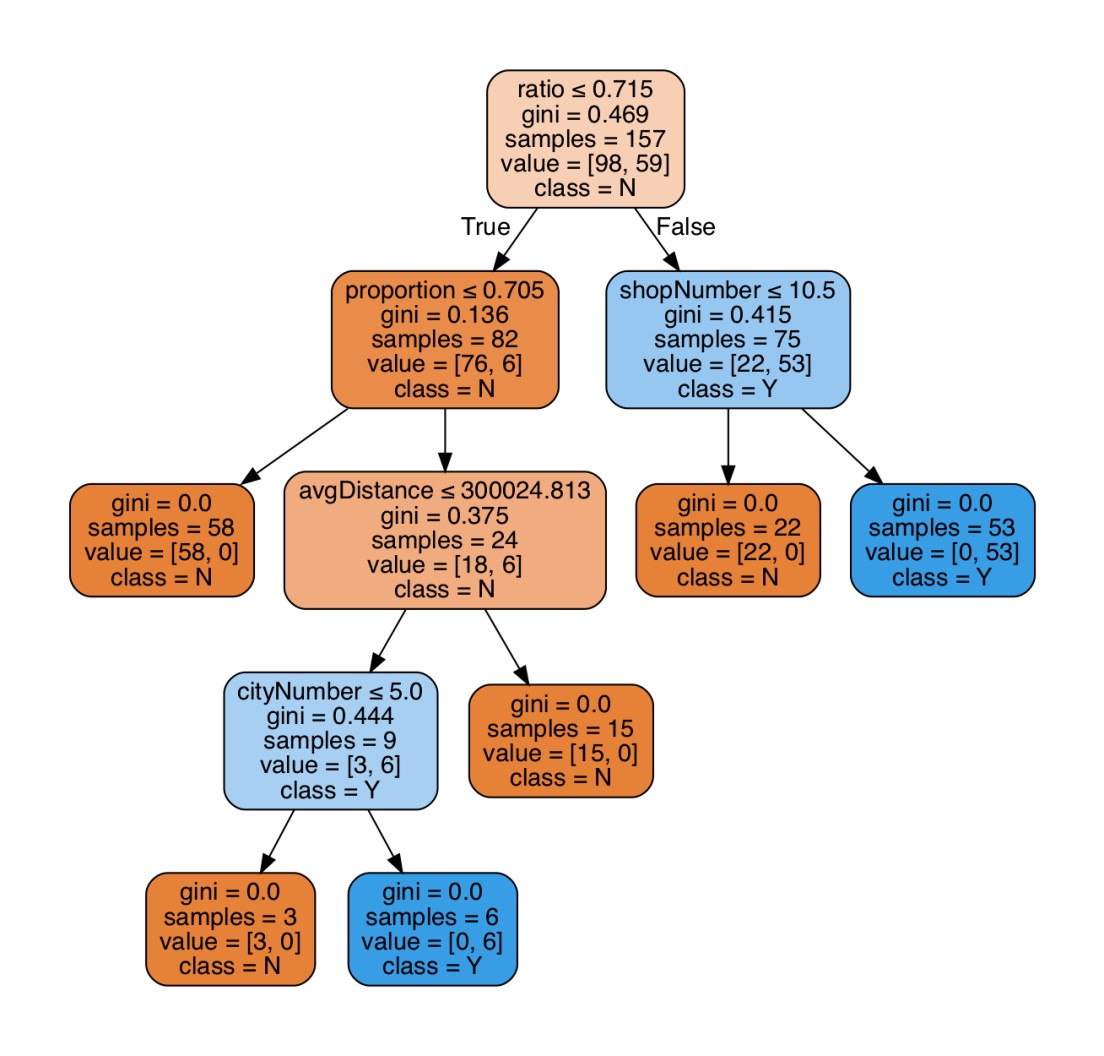
本部分主要介绍了单词对分类结果的生成方法和结果的评价。

### 6.2。1 单词对方法

用于生成单词对的脚本是基于生成名词短语的脚本构建的。脚本中的更改是, 在 HTMLParser 识别标记的内容并筛选特殊符号后, 脚本按空格将内容拆分为单词对。例如, "我是 XXX" 将被拆分为 "我是" 和 "我是 XXX"。单词对脚本的结果包含独立的单词、名词短语和单词对, 因为某些内容本身是一个单词或名词短语。单词对脚本的其余部分与名词短语相同。

### 6.2。2 单词对决策树发现

"对决策树" 一词使用 "比率"、"比例"、"平均距离"、"商店号" 和 "城市号" 作为功能。



### 图 41: 单词对决策树

有117词对被判断作为地方词对。(黑布丁, 在皮塔, 鳕鱼卵, 伊恩布鲁, 香肠大, 鱼片, 豌豆小, 或肉汁, 薄荷馅饼, 鸡肉, 鸡肉, 特殊的鱼, 钙质, 油松皮, 鸡肉 pakora, 馅饼, 大香肠, 大哈达克, 混合肉, 蔬菜 pakora, 网上订购,鱼晚餐, 哈吉斯, 熏制香肠, 奶酪西红柿, 豌豆咖喱, 香肠晚餐, 南炸, 面包, 小鱼, 大豌豆, 烤肉, 鸡肉片, 半披萨, 红牛, 金排骨, 沙拉或, 扇贝, 包裹餐, 果汁, 鸡尾酒, 盐, 肉, 肉, 烤土豆,热火, 多纳, 可乐毫升, 布鲁尔, 可乐, 薯条, 薯条, 薯条, 所有牛排, 红宝石番石榴, 烤土豆, 纳安面包, 烟熏香肠, 桑皮晚餐, 香肠单, 意大利香肠酱, 混合波科拉, 馅饼晚餐, 咬和, 汉堡单, 意大利面面包店, 布丁晚餐, 奶酪馅饼, 芯片卷, 芯片牛排, 比萨饼晚餐, 黄原渣, 比萨饼单, 炸披萨, 布鲁兄弟奶酪, 乳房晚餐, 哈格斯晚餐, 白布丁, 鸡肉巴迪, 桑皮单, 晚餐, 烤肉串, 肉饭,方塔毫升, 饮食熨斗, 在纳安, 馅饼单, 金块晚餐, 金块单, 鸡晚餐, 布罗摩毫升, 鱼屁股, 腿单, 鱼单, 半油炸, 肋骨单, 鸡单, 现在在线, 外卖订单, 溜冰鞋, 苏伊卷, 乳房单, 在线瞬间, 布丁单, 腿晚餐, 安全与, 菠萝戒指, 牛排晚餐, 帕科拉蔬菜, 商店外卖, 辛辣的哈格斯, 牛排单, 汉堡晚餐, 哈格斯单, 比萨饼紧缩, 牛排, 汉堡单)。

### 6.2。3 单词对决策树计算

在这些单词对中, 它们中的许多包含区域独立的单词, 如 "安全地与" 和 "红宝石番石榴"。此外, 其中许多出现在名词短语的结果, 如 "奶酪番茄" 和 "蔬菜 pakora"。此外, 一些单词对, 如 "牛排单", "可乐毫升" 和 "半比萨饼" 可能反映了区域营销习惯, 在这些地区, 居民可能喜欢标记饮料在毫升, 或他们将出售比萨饼一半。此外, 根据单词对决策树, 该项目发现 "城市号" 特征影响了分类结果。因此, 在单词对逻辑回归模型中, 项目将使用此功能生成模型。

### 6.2。4 单词对逻辑回归结果

|  |  |  |
| --- | --- | --- |
| **正规化** | **L1** | **L2** |
| 分类率 | 88.77 | 74.48 |
| 百分比门槛值 | 无法分辨 | 无法分辨 |
| 系数  ("比例", "比例",  "城市号码" 和  "平均距离") | 1.83013399 和 + 00  6.2 2791230 和 + 00  -2.69416502 E-01-, 797070787 E-06 | 1.01 66688 e-01  1.80860335 e-01  -1.26984839 E-01  5.8837008 e-06 |
| 区域单词的数量 | 无法分辨 | 无法分辨 |
| 有概率的区域词 | 毫无 意义 | 毫无 意义 |

### 表 9: 选择单词对、"比例"、"比例"、"城市数字" 和 "平均距离" 作为特征时的惩罚选择比较

有200个单词对, 当使用 L1 正则化时, 区域概率超过 50%, 在使用 L2 正则化时, 有684个单词对的区域概率超过50%。

### 6.2。5 单词对逻辑回归评估

根据单词对的逻辑回归结果, 虽然使用 L1 正则化时的分类率较高, 但在区域概率大于50% 的200个单词对结果中, 该项目无法找到区域阈值。该项目发现, 在这200个单词对中, 有许多单词对不是通过地理可视化结果和趋势图进行的区域单词对。此外, 通过对这些非区域词对的分析, 发现这些词对具有高 "比"、大 "平均距离" 和低 "比例" 的相同特征。通过观察单词对训练数据集, 项目知道训练集中所有区域单词对的 "比率" 高于 59%, 但其他三个要素不能判断某个阈值。此外, 根据使用 L1 正则化时每个特征的系数, 该项目发现 "比率" 对结果的影响最大, "平均距离" 的影响较小。因此, 这导致了一个糟糕的分类结果。但是, 该项目认为, 模型分类不良的根本原因是训练数据集样本不够。

使用 L2 正则化时, "平均距离" 对分类结果的影响最大。因此, 这就是为什么有更多的非区域单词对被标识为区域单词对。

在上述描述的基础上, 项目认为利用逻辑回归获得的单词对的分类结果毫无意义。因此, 该项目只对结果进行了讨论, 没有显示结果细节。

## 6。3 总结

通过比较这三种数据类型的分类结果, 使用独立的分类词可以找到大多数区域词, 但需要项目继续找到它们的使用背景, 以确定为什么它们被判断为区域词词。在使用名词短语进行分类时, 项目可以直接从大多数结果中找到将其归类为区域名词短语的原因, 因为大多数名词短语都是菜名。使用单词对数据集时, 区域分类结果 (由决策树生成) 比独立词和名词短语的结果更多。其主要原因是, 在项目拆分内容后, 项目生成的单词对数介于独立单词和名词短语的数量之间。此外, 一些在生成独立词和名词短语的过程中被过滤的单词出现在单词对中。

例如, 在生成独立单词和名词短语的提取过程中, "ml" 和 "ltr" 将被过滤, 但它们被保留在单词对中。因此, 当这些词与区域独立词 (如 "兄弟"、"兄弟毫升" 和 "兄弟 ltr") 一起使用时, 被判断为区域单词对。此外, 由于单词对训练数据集的不完善, 逻辑回归模型的分类效果不好, 回归模型的结果没有参考价值。

**第7章**

# 结论

该项目最大的成功是, 该项目成功地探索了将 "鱼和芯片" 商店菜单信息与英国地区差异联系起来的方法。通过四个探索迭代, 该项目成功地探索了餐馆网站中潜在的区域信息。第一次迭代发现了获取数据和清理数据的方法, 包括数据源选择、数据检索、从 HTML 文件中提取独立单词, 最后获得具有坐标的独立单词。 此外, 通过可视化单词的分布和商店数量的趋势, 随着距离 (米) 从中心点的增加, 该项目确定了一些区域文字, 并成功地确定了区域文字的特点。基于这些特点, 该项目探讨了区域文字的分类方法。第二次迭代通过使用决策树成功地对单词进行了分类, 这是项目第一次看到区域结果。然而, 在独立词语方面, 项目很难直接解释为什么它们表现出区域性。因此, 该项目确定了这些词语出现的背景, 并成功地解释了这些词语的区域原因。通过对第三次迭代的探索, 成功地发现了词语呈现区域性的概率以及各特征对分类结果的影响。在成功获得独立词的区域结果后, 本项目在第四次迭代中探索了名词短语和词对的区域性。

通过对项目区域成果的评价, 发现了一些区域差异。第一个是所有差异中最明显的差异, 那就是饮食习惯的差异。比如, 住在苏格兰喜欢吃和喝, 英格兰人民喜欢吃和。第二个是购物习惯的差异。例如, 在伦敦及其周边城市, 有许多 "鱼和芯片" 餐馆出售哈根达斯冰淇淋。这可能意味着, 那个地区的人在 "鱼" 消费时, 喜欢购买哈根达斯

& 芯片的商店。第三是 "鱼和芯片" 店销售习惯的差异。例如, 在曼彻斯特和周边城市, 许多 "鱼和芯片" 商店为12岁以下的儿童提供折扣, 有许多商店出售一半的菜, 如 "半披萨"。最后的区别反映在语言的用法上, 比如苏格兰人喜欢用 ' 晚餐 ' 来形容晚餐。

但是, 该项目也有局限性。首先是在三种类型的内容中出现的不完善的训练数据集。这可能会导致项目获得的分类结果出现偏差, 甚至分类不正确, 例如单词对的逻辑回归结果。第二个限制是, 有限的时间导致对结果的分析有限, 项目认为还有其他原因可以用来解释为什么逻辑回归模型不能很好地对区域和非区域内容进行分类。此外, 项目还希望研究更多的分类模型参数, 以优化分类模型, 但由于时间的原因, 项目没有做到。第三个限制是, 项目选择的城市在纽卡斯尔附近的一些地区分布不均。该项目发现, 在该地区, 几个选定的城市分布密集, 这可能会对区域成果造成一些偏见。

虽然该项目有这些局限性, 但该项目最初探讨了这项研究的可行方法, 该项目回答了在哪里获得数据、如何检索数据、如何清理数据、如何从数据中挖掘要素以及如何对数据进行分类。这些方法可以被随后类似研究的研究人员参考或重用。此外, 该项目取得的区域成果可作为随后类似研究的评价参考。

**第8章**

# 今后的工作

对于这个项目, 未来还有很多事情可以做。

## 8。8 寻找更多的 "鱼和芯片" 商店菜单

该项目只使用谷歌和 TripAdvisor 搜索 "鱼和芯片" 店, 但该项目认为, 还有许多其他网站可以用来搜索更多的 "鱼和芯片" 商店菜单。此外, 该项目可以找到一种方法来提取一些 PDF 菜单页面的信息。通过这些方式, 项目可以向数据集添加更多的内容示例, 这样做将带来两个好处。第一个好处是, 该项目可能会发现更多的区域内容, 包括新出现的内容 (在本项目中区域或非区域的结果中不存在) 和最初被认为是非区域的内容 (商店数量低于10) 到区域内容。第二个好处是, 更多的样本将使内容的区域或非区域特征更加不同。这意味着将有更多的区域内容集中在中心点, 从中心点将有更多的非区域内容分布在更广泛的地区。因此, 训练数据集可以得到改进, 分类模型可以通过更精确的系数生成更准确的。 最后, 该项目可以获得更准确的分类结果。

## 8。8 分类模型的算法和参数的深入研究

由于时间有限, 该项目只尝试了两种分类模型, 只分析了模型的几个参数。在决策树方面, 项目可以尝试使用更多参数来构建树。例如, 项目可以设置一个最小样本大小, 如果分支至少有此大小的样本, 则可以对分支进行划分, 而不是构建一个巨大的树来满足所有训练样本。考虑到逻辑回归, 项目可以可视化回归模型, 因此项目可以更好地了解模型如何划分数据集。这有助于分析逻辑回归模型的结果。此外, 该项目还可以分析更多的逻辑回归参数, 如学习率和损失函数优化方法, 以更好地解释本项目逻辑回归获得的概率结果。

## 8。8 同义词检测

将来, 项目可以尝试使用新方法合并相同的内容, 该方法将确定内容之间的相似度。如果相似度较高, 类似的内容将被划分为一个类别, 并具有相同的内容名称。例如, "芯片" 和 "芯片" 的相似概率可能很高。因此, 该项目可以将它们合并为 "芯片"。通过这种方式, 项目可以避免同一含义的内容因字符串不匹配而被划分为不同的内容。

## 8。8 将项目扩展到更宽的范围

今后, 该项目可以尝试从其他类型的餐馆收集数据和地雷信息, 而不仅仅是关注 "鱼和芯片" 商店。这可以增加内容样本中的数据量, 以便内容样本能够更好地反映其特征, 从而提高分类的准确性。此外, 该项目还可以使用同样的方法挖掘其他国家的餐馆数据, 而不仅仅是在英国, 以探讨其他国家的地区差异。此外, 该项目还可以根据菜单上的菜品或酱汁的频率, 从菜单数据中挖掘其他信息, 如预测当地人的口味。通过这种方式, 该项目可以为想在当地开餐馆或销售食材的商人提供合理的建议。

**附录 A**

## 逻辑回归概率结果

A.1 独立单词分类结果 (具有 "比率"、"比例"、"平均距离" 特征的 L1 惩罚)。

|  |  |
| --- | --- |
| **词** | 非区域和区域 |
| **Cob** | [0.01 187646404012133, 0.98 1234349598779] |
| **达兹** | [0.0356 65746464686514, 0.9643421462531349] |
| **立即** | [0.03880521046799243, 0.96 4747595203076] |
| **哈根达斯** | [0.042161688886797874, 0.95 47868611132021] |
| **汉堡** | [0.0453101010101313992514, 0.95 46485948600600-75] |
| **约克 郡** | [0.067 26020112458264, 0.9327397988754174] |
| **卡尔博纳拉** | [0.07 4865454909691001, 0.0748655490961001] |
| **地狱** | [0.07 8209914149680193, 0.92 1790455031981] |
| **牛蒡** | [0.080610448440063548, 0.919389512259936445] |
| **蒲公英** | [0.080610448440063548, 0.919389512259936445] |
| **飞溅** | [0.0885926617249821, 0.91 14440724827275018] |
| **干酪** | [0.10952897901919191857, 0.10952897901931857] |
| **四** | [0.1148709886937076, 0.88 5129011362924] |
| **帕蒂** | [0.11670735823124456, 0.88 392641768735354] |
| **纳安** | [0.12525699647456212, 0.87 4744300352543779] |
| **岩石** | [0.12849691511617833, 0.87 15030848388217] |
| **给** | [0.14043323848824363, 0.14043323848824363] |
| **煮** | [0.1525299788 2747477, 0.8470021722552] |
| **通心粉** | [0.18069696474742066, 0.81973300352524793] |
| **季节** | [0.1855 24357047070105, 0.18552435704470105] |

|  |  |
| --- | --- |
| **康沃尔** | [0.18743374731899054, 0.812256626810095] |
| **肉** | [0.18850927708171743, 0.81 14907292988257] |
| **滑冰** | [0.19105596269953462, 0.80 8944373004654] |
| **哈吉斯** | [0.2000 666988938408545, 0.799 330121216169145] |
| **番 石榴** | [0.210533340420163552, 0.7894646495998365] |
| **烟熏** | [0.2188211503446117, 0.78 117882946553883] |
| **扇贝** | [0.23 742272727360933, 0.76 25 772372626339067] |
| **激情** | [0.2422077949410029497, 0.757792105899705] |
| **拍摄** | [0.247684284444102064, 0.75 231515151538994] |
| **肋** | [0.24915028186127286, 0.75084971818827271] |
| **紧缩** | [0.2595798061837933, 0.7404203838162067] |
| **基辅** | [0.29097630707112465, 0.29097630707112465] |
| **普卡** | [0.2987 至206171735254, 0.29878720611735254] |
| **苏伊** | [0.3070038321413041, 0.6929967878786959] |
| **位** | [0.3167334820875807, 0.68 26266517179124193] |
| **下** | [0.3168042925668779, 0.68 31957074331221] |
| **农场** | [0.3227256066834283, 0.67727393939333316557] |
| **巴尔蒂** | [0.3348704966159057, 0.66 5192929503840943] |
| **价值** | [0.301 117228726148, 0.649829771283852] |
| **桥** | [0.352 421779327335, 0.6487578252067266] |
| **伊恩** | [0.352 421779327335, 0.6487578252067266] |
| **糊 状** | [0.35460931 105113624, 0.6459090688488838] |
| **黄瓜** | [0.365 614270875 258, 0.637 438438412442] |
| **安全** | [0.38772979177115596, 0.38772979177115596] |
| **饼干** | [0.39519298167919703, 0.39519298167919703] |
| **意大利 面** | [0.3957820821304686, 0.3957820821304686] |
| **Facebook** | [0.42024142622620053, 0.57 975737737995] |
| **服务** | [0.4248062839850639, 0.57 51937160149361] |
| **净资产 收益 率** | [0.426228245944002295, 0.57 7171754059977] |
| **蒙奇** | [0.4370817175920020039, 0.56329108244077961] |
| **蘑菇** | [0.438454209134041226, 0.56477059595274] |
| **电话** | [0.465500101032222067, 0.53499648989677933] |
| **选择** | [0.4743260958093295, 0.52567390904196605] |
| **帕科拉** | [0.47695369015475175, 0.52 30430309842482] |
| **希腊语** | [0.4793352062612427, 0.4793352062612427] |
| **垃圾 邮件** | [0.4842444012413775, 0.5175464659858625] |
| **溢价** | [0.4826230587021527, 0.517373769412978473] |
| **萨莫萨** | [0.4909144128010813, 0.509085578619819] |

A.2 名词短语分类结果 (具有 "比率"、"比例"、"平均距离" 特征的 L1 惩罚)。

|  |  |
| --- | --- |
| **名词短语** | 非区域和区域 |
| **奶酪馅饼** | [0.013309049172268272, 0.98986690904082771717] |
| **汉堡晚餐** | [0.0186291426474251, 0.98 137085753525749] |
| **比萨饼紧缩** | [0.02 2596262613137979788, 0.977 403838386130202] |
| **土豆派** | [0.023568146763607967, 0.97 64646318323692] |
| **地狱** | [0.032204851371333258, 0.967795141486246674] |
| **纳安** | [0.03268870363025178, 0.9672812969697482] |
| **菠萝圈** | [0.034 4812327403224, 0.96 55567672591268] |
| **布丁晚餐** | [0.03671211066273117, 0.96 3288289332686] |
| **比萨饼晚餐** | [0.9620667230727336, 0.9620667230727336] |
| **通心粉奶酪** | [0.0384770449522916, 0.96 15229295504771] |
| **减肥兄弟** | [0.04 103344031818725975, 0.958966259681274] |

|  |  |
| --- | --- |
| **芯片店外卖订单** | [0.049003760243348515, 0.95099697566515] |
| **减肥可乐** | [0.05 2748284463645713, 0.94727151553632929] |
| **汉堡** | [0.053531717178504022, 0.996464628282814995] |
| **帕蒂** | [0.06093147368861507, 0.9390685263113849] |
| **芯片辊** | [0.06 282885877899259, 0.93716104122210074] |
| **蒲公英** | [0.0642077996357829794, 0.9357920364217021] |
| **鸡肉** | [0.067 10259897974517, 0.93 328974018902548] |
| **苏伊卷** | [0.07 10555101051131, 0.998 9444489480087] |
| **国王肋骨** | [0.07 476846265165665, 0.923077343414] |
| **位** | [0.77 6340456464334098, 0.92 2236595435353565959] |
| **鸡** | [0.0806512191335388068, 0.9 344806464461193] |
| **派晚餐** | [0.08337077826346695, 0.08337077826346695] |
| **拍摄** | [0.09 24513132840321, 0.9075 4486778668] |
| **牛蒡** | [0.11106739501985041, 0.88 89326049449801496] |
| **奶酪番茄** | [0.1334193658886207, 0.86858061113793] |
| **牛肉洋葱饼** | [0.1360888 11022031546, 0.1360885812031546] |
| **哈吉斯** | [0.14537848581225887, 0.85 42641515151877411] |
| **蘑菇** | [0.15603603785303422, 0.843969696262669658] |
| **鸡胸肉晚餐** | [0.16533612727008, 0.83446308787959999] |
| **糊 状** | [0.16887846051290278, 0.8311215394870972] |
| **晚饭** | [0.1828044881275689, 0.817171: 55511724311] |
| **炸鱼片** | [0.19210810099 80167, 0.80 778889999 1988] |
| **芝士汉堡半磅** | [0.20097439977528253, 0.79 909025600247175] |
| **芝士汉堡季度磅** | [0.20097439977528253, 0.79 909025600247175] |
| **鸡肉帕科拉** | [0.24954773 7788309, 0.750452626031691] |
| **饼干** | [0.2592799667107383, 0.740720020032892617] |
| **牛** | [0.26262262620247466, 0.7336771717179725253] |
| **桥** | [0.279724131022031, 0.72026277786377969] |
| **鸡块粉** | [0.2976763913638301, 0.297673913638301] |
| **掘金** | [0.310712909060468216, 0.31071290604648216] |
| **蔬菜帕科拉** | [0.3141616446386699668, 0.685855656313300332] |
| **帕西蒂** | [0.32791435320038853, 0.67 20885646969611115] |
| **意大利 面** | [0.3284135989436908, 0.67 15864040105660992] |
| **薯条沙拉** | [0.33646537799041176, 0.66353462 20095882] |
| **扇贝** | [0.3597985894131531, 0.604 2014105868469] |
| **流行** | [0.3661374073969268, 0.6368692626030732] |
| **帕科拉** | [0.380 938337612429626, 0.619016238757037] |
| **网站** | [0.382906709066207, 0.617709292929292933793] |
| **面食** | [0.392441367677101085, 0.3924413678971085] |
| **烤肉串包装** | [0.4040777030550764441, 0.4060773055076441] |
| **英寸马格里塔** | [0.4082 24409512546662, 0.59 590448433838] |
| **香肠晚餐** | [0.437 1353690771911, 0.56646292929278089] |
| **片断** | [0.44206894811490416, 0.579. 710010518850958] |
| **英寸面包** | [0.44448106111373529, 0.55 5188882626471] |
| **薄** | [0.445 16408225252588, 0.55 43591717741212] |
| **馅饼片** | [0.455055646034040344997, 0.544435359659651] |
| **鳕鱼** | [0.460536260973207, 0.53939439442679] |
| **春卷** | [0.4624339277638466, 0.53756672232661534] |
| **Facebook** | [0.4765 96222514401, 0.52340380777485599] |
| **垃圾邮件切割机** | [0.4841356562629295611, 0.51586433797704389] |
| **肾** | [0.49 192244884715, 0.50 77555115285] |

A.3 名词短语分类结果 (具有 "比率"、"比例"、"平均距离" 特征的 L2 惩罚)。

|  |  |
| --- | --- |
| **名词短语** | 非区域和区域 |
| **奶酪馅饼** | [0.04 302663367675612, 0.95697373663202439] |
| **汉堡晚餐** | [0.05669634713763139, 0.9433036528623686] |
| **比萨饼紧缩** | [0.06500442003732525, 0.93499595979962698] |
| **土豆派** | [0.06658411254621566, 0.9334158874537843] |
| **地狱** | [0.77 5083674474747169, 0.92 24999255210883] |
| **菠萝圈** | [0.08145369004390668, 0.91854630995660933] |
| **纳安** | [0.0867078283991937, 0.08670872839951937] |
| **通心粉奶酪** | [0.09 43291414144692861, 0.905678785330714] |
| **布丁晚餐** | [0.095259936198881202, 0.90474063801188] |
| **比萨饼晚餐** | [0.09 8204949567760055, 0.901795044322223994] |
| **减肥兄弟** | [0.0998603121828819, 0.900 139685817118] |
| **芯片店外卖订单** | [0.103838264445965, 0.88 91561375401313] |
| **减肥可乐** | [0.115201424734668147, 0.88 4744987526533185] |
| **苏伊卷** | [0.11605131791091994, 0.88 39486820890001] |
| **帕蒂** | [0.12619217081624445, 0.87 380 78291837555] |
| **汉堡** | [0.12126611165498678, 0.87069383834501322] |
| **鸡肉** | [0.132938387878728997, 0.13293857088732897] |
| **蒲公英** | [0.133 32525541675 4494, 0.86 80488324551] |
| **芯片辊** | [0.1387 4956035593355, 0.86 12504396444664] |
| **位** | [0.1430330050784575, 0.85696699499211443] |
| **鸡** | [0.15360521749090306, 0.84639415090969] |
| **国王肋骨** | [0.1544355521144186, 0.846304447454282814] |
| **拍摄** | [0.153262613458673, 0.840673838186413227] |
| **牛蒡** | [0.164813161616989252, 0.16458133161698252] |
| **派晚餐** | [0.177601010926262627819, 0.1776010926627819] |

|  |  |
| --- | --- |
| **蘑菇** | [0.1948 3683998675805, 0.80 51616160013242] |
| **奶酪番茄** | [0.2070747074747423562, 0.79 2929252979257644] |
| **牛肉洋葱饼** | [0.20907048034755127, 0.790909299196524487] |
| **糊 状** | [0.02 20804004000819970743, 0.7779599916800926] |
| **哈吉斯** | [0.22384433767294343, 0.777 6155662327220566] |
| **芝士汉堡半磅** | [0.24157381600573946, 0.758426263939262605] |
| **芝士汉堡季度磅** | [0.24157381600573946, 0.758426263939262605] |
| **鸡胸肉晚餐** | [0.243395646468951732, 0.75664331048268] |
| **炸鱼片** | [0.2505011589087093, 0.749498841099907] |
| **晚饭** | [0.2820484701165006, 0.717951529292934994] |
| **饼干** | [0.29260688 44445514, 0.7073931155654486] |
| **牛** | [0.31996 264499926, 0.3199562641957926] |
| **鸡块粉** | [0.32125571887427606, 0.67 844442811257239] |
| **鸡肉帕科拉** | [0.32258630530369103, 0.677 4136694696309] |
| **掘金** | [0.33 8887826877777, 0.66 1821571240402] |
| **帕西蒂** | [0.3434913363840276, 0.3434913363840276] |
| **薯条沙拉** | [0.334 544474739232, 0.6562452582600708] |
| **扇贝** | [0.37 1331077719297, 0.37133107972519297] |
| **流行** | [0.375661140709595 1995, 0.624385929048] |
| **网站** | [0.3828808892295264, 0.6171191277042736] |
| **桥** | [0.3870878662375089, 0.61212121313624911] |
| **英寸面包** | [0.3949447590422569, 0.3949447590422569] |
| **英寸马格里塔** | [0.4051898339090944395, 0.59 484166009556] |
| **烤肉串包装** | [0.4092167362183604, 0.4092167362183604] |
| **意大利 面** | [0.4148 65666000986, 0.51348033999014] |
| **蔬菜帕科拉** | [0.42 201666362053063, 0.5779837379499337] |
| **馅饼片** | [0.422254 第227 56608205, 0.42219252756608205] |
| **香肠晚餐** | [0.43141384413909924, 0.56858615151515609208] |
| **鳕鱼** | [0.4401889430287208, 0.55 94495697 12792] |
| **鳕鱼** | [0.44418610097133, 0.55 58138997202867] |
| **鲁比孔芒果** | [0.4468756085262042, 0.55 31244394377958] |
| **垃圾邮件切割机** | [0.45 105422928562, 0.5484457777071438] |
| **鱼咬伤** | [0.43042008974705, 0.54679991025033] |
| **春卷** | [0.443704464898912, 0.545629553011988] |
| **片断** | [0.45 698 8310102715691, 0.54 37016167288284309] |
| **Facebook** | [0.4754315709258996, 0.52 245629290741004] |
| **帕科拉** | [0.47 8014161604129306, 0.52 19858385870169] |
| **薄** | [0.4788787474808592726, 0.52 11220251414707 27] |
| **肾** | [0.4799 76488914886 03, 0.52 20023311068514] |
| **英寸肉宴** | [0.88 99211444644704, 0.4899211474645704] |
| **面食** | [0.49 921432925156173, 0.5007856707484383] |

## 引用

1. Wickham, H. (2014年)。紧张的数据。*统计软件杂志*,*59*(10), 1-23。
2. Hafley, W. L., & Lewis, J. s. (1963年)。分析混乱的数据。*工业和工程化学*,*55*(4), 37-39。
3. 瓦苏米塔··阿阿拉什(2013年12月26日)。TastyKhana 推出了网站的谷歌地图功能。(互联网)。*经济时报*P。《经济时报》, 2013年12月26日。
4. 奥康纳, P. (2010年)。管理酒店在 TripAdvisor 上的形象。*酒店营销与管理杂志*,*19*(7), 754-772。
5. Hand, D. J. (2007年)。数据挖掘的原则。*药物安全*,*30*(7), 621-622。
6. 鱼和薯条(2010年)。*营养和食品科学,* *40*(6), 157-165。
7. *只是吃*.在线访问<https://www.just-eat.co.uk/>(参考 19/08/2018)。
8. *只需佩戴本网站的条款和条件。*在线访问 [https://www.justeat.co.uk/termsandconditions](https://www.just-eat.co.uk/termsandconditions) (参考 19/08/2018)。
9. 卡斯蒂略, C. (2005年, 6月)。有效的网络爬网。在*行为信息论坛*(第39卷, 第1期, 55-56 页)。Acm。
10. Goerzen, J. (2004年)。Web 客户端访问。在*Python 网络编程的基础*(第113-126 页)。加州伯克利。
11. Rahm, E., & Do, H. h. (2000年)。数据清理: 问题和当前
12. LI, W, & HUANG, Y. (2007年)。基于 HtmlPaser 的网络信息提取 [J]。*军械工业自动化*,*7*, 024。
13. 林, S., 和 Hu, Y. (2010年, 7月)。一种基于 html 解析器的网络信息提取方法。在*信息技术与计算机科学 (ITCS), 2010年第二次国际会议*(第2284-287 页)。Ieee。
14. 汤普森, K. (1968)。编程技术: 正则表达式搜索算法。*含石棉材料的来文*,*11*(6), 419-422。
15. Chowdhury, G. G. (2003年)。自然语言处理。*信息科学和技术年度审查*,*37。*(1), 51-89。
16. 伯德, S., & Loper, E. (2004年, 7月)。NLTK: 自然语言工具包。在*2004年 ACL 会议记录互动海报及示范会议*(第31页)。计算语言学协会。
17. 朱, J. (1994年)。以地球为中心的地球固定坐标转换为大地测量坐标。*航空航天和电子系统方面的 IEEE 交易*,*30*(3), 957-961。
18. Clynch, J. R. (2006年)。地球坐标。*电子文件, 2月*.
19. Montenbruck, o., Gill, E. & Terzibaschian, T. (2000年)。关于 BIRD ACS 参考框架的说明。
20. Barrett, p., hunter, J., Miller, J. t., su, j. c. & Greenfield, P. (2005年, 12月)。一个便携式的 Python 绘图包。在*天文数据分析软件和系统*(第347卷, 第91页)。
21. Whitaker, J. (2011年)。Matplotlib 底图工具包用户指南。*Matplotlib Basemap 工具包文档, 2月*.
22. 没错, S. (2009年)。*适用于 Python 开发人员的 Matplotlib*. 包业出版有限公司。
23. Han, J., Pei, J., & Kamber, M. (2011年)。*数据挖掘: 概念和技术*.埃尔塞维耶
24. 郭, & 科尔施。监督学习对音乐句法处理与事件相关的潜在因素的影响。*脑研究1626*232246。
25. Jin, C., De-Lin, L., & Fen-hi, M. (2009年, 7月)。一种改进的 ID3 决策树算法。在*计算机科学与教育, 2009年。ICCSE. 09。第四届国际会议*(127-130 页)。Ieee。
26. Hssina, B., Merbouha, a., Ezzikouri, H. & Erritali, m. (2014年)。决策树 ID3 和 C4 的比较研究。5。*国际高级计算机科学与应用杂志*,*4*(2)。
27. Umanol, m., okamoto, H., Hatono, i., Tamura, H. i. r. o. k. i., Kawachi, f., Umedzu, s, & Kinoshita, J. (1994年, 6月)。模糊 ID3 算法在诊断系统中的应用。在*模糊系统, 1994年。IEEE 计算智能世界大会, 第三届 IEEE 大会会议记录*(第22113-2118 页)。Ieee。
28. Quinlan, J. R. (1986年)。决策树的感应。*机器学习*,*1*(1), 81106。
29. 彭, 陈, j., & 周, H. (2009年)。一种 ID3 决策树学习算法的实现。*从网络上。拱。乌西德埃杜。。pdf 检索日期: 5月*,*13*.
30. Steinberg, D., & Colla, P. (2009年)。购物车: 分类和回归树。*数据挖掘中的十大算法*,*9*, 179。
31. Rutkowski, l., Pietruczuk, l., duda, p. & Jaworski, M. (2013年)。基于 M齐尔米德的绑定挖掘数据流的决策树。*关于知识和数据工程的 IEEE 事务*,*25*(6), 1272-1279。
32. 尚, 黄, h., 朱, h., 林, y., Qu, y., & wang, Z. (2007年)。一种新的文本分类特征选择算法。*具有应用的专家系统*,*33*(1), 1-5。
33. Peng, C. y. j., lee, K. l., & Ingersoll, G. m. (2002年)。逻辑回归分析和报告简介。*教育研究杂志*,*96*(1), 3-14。
34. Walker, S. h., & Duncan, D. b. (1967年)。事件概率的估计作为几个自变量的函数。*生物识别技术*,*54*(1-2), 167-179。
35. Koh, K., kim, S. j., & Boyd, S. (2007年)。大规模 l1 正则化逻辑回归的内点方法。*机器学习研究杂志*,*8*(7月), 1519-1555。
36. Ng, A. Y. (2004年, 7月)。特征选择, l1 vs. l2 正则化, 旋转不变性。在*第21届机器学习问题国际会议记录*(第78页)。Acm。
37. Tibshirani, R. (2011年)。回归收缩和选择通过套索: 回顾。*皇家统计学会杂志: b 系列 (统计方法)*,*73*(3), 273-282。
38. Goeman, J., Meijer, r., & Chaturvedi, N. (2012年)。L1 和 L2 惩罚回归模型。*克兰。r-项目。或*.
39. Dean, J., & Ghemawat, S. (2010年)。MapReduce: 一种灵活的数据处理工具。*含石棉材料的来文*,*53*(1), 72-77。
40. 徐, L., Liu, G., & Chen, Z. (2012年12月)。基于决策树算法的半速率阈值设置优化模型研究。在*计算机科学与网络技术 (ICCSNT), 2012年第二届国际会议*(316-320 页)。Ieee。
41. Vanderplas, f., Varoquaux, g., Gramfort, A., Michel, v., Thirion, b., Grisel, O., & Vanderplas, J. (2011年)。科学学习: Python 中的机器学习。*机器学习研究杂志*,*12*(10月), 2825-2830。
42. *决策树*.在线访问 [http://scikit-民营. org/stlelele/meodeses\ tree. html](http://scikit-learn.org/stable/modules/tree.html) (参考 19/08/2018)。
43. Wester, A., Øvrelid, L., Velldal, E., & hammer, H. l. (2016年)。在线讨论中的威胁检测。在*第七次主体性、情感和社会媒体分析计算方法研讨会论文集*(66-71 页)。