厦 门 大 学

XIAMEN UNIVERSITY

课 程 论 文

COURSE PAPER

基于外卖评论的文本情感分析与 LDA 主题分析

一、引言

1、背景

新冠疫情对餐饮堂食带来了巨大冲击，而外卖的市场规模则在稳步扩大。截至 2022 年底，我国网上外卖用户规模已达到 5.21 亿，外卖餐饮行业市场规模同比增长 16%。随着越来越多的消费者通过外卖点餐，因此关于外卖的点评数据也变得愈加庞大，而海量的外卖评论背后也蕴藏着巨大的商业价值。对于消费者而言，由于外卖这一线上下单线下配送的方式无法直接与商家接触，因此容易出现信息偏差，故而主要是通过评论对外卖商品的口味进行判断；对于商家而言，由于平台的外卖商家变多、竞争激烈，因此通过评论获取顾客最直观的消费体验从而提高竞争力也尤为重要；对于平台而言，顾客对店铺的评价不仅可以反映出加盟店铺的优劣、从而优化推荐算法，同时顾客还对外卖平台的优惠机制、配送速度等服务进行评价，外卖平台可以通过这些评价优化平台服务从而提高竞争力。因此，挖掘外卖评论中蕴含的信息、尤其是情感信息是十分必要的。

2、研究意义

基于用户的文本评论数据进行情感分析，有重要的实际意义，可以帮助商家优化外卖商品的口味、质量、服务等，提升店铺的竞争力，在激烈的竞争中脱颖而出；此外，顾客关于平台政策、机制、配送等方面的评价还可以为外卖平台提供建议，从而优化用户体验，提高用户在平台内的活跃程度。

3、研究思路

本报告旨在对某外卖平台中某家店铺的评论文本进行情感分析，主要研究思路如下：

首先，利用文本挖掘技术，对碎片化、非结构化的外卖评论数据进行清洗与处理，如分词、词性标注、去停用词等，从而转化为结构化数据；

其次，参考知网所发布的情感分析词汇集，计算评论数据的正负情感指数，从而进行基于情感词典的情感分析，并通过词云图直观查看正面和负面评价的关

键词；

第三，构建模型进行情感标签预测，主要比较了决策树和逻辑斯蒂回归两种分类器的分类效果；

第四，采用 LDA 主题模型提取评论关键信息，进一步了解顾客的意见、外卖产品及平台的优缺点等。

二、数据描述与预处理

1、数据基本描述

报告选取中文外卖评论数据集，该数据集是从某外卖平台的某商铺中进行爬取，包括外卖订单的评论及对每条评论手工设定的情感标签，其中“pos”代表积极情感，“neg”代表消极情感。表 1 对该外卖评论数据集的前五行数据和后五行数据进行了展示。从图 1 中可以看出，该数据集共包含 11987 条数据，且没有缺失值。积极情感的评论与消极情感的评论数量大约比值为 1:2，数据集存在数据不平衡的情况。

2、数据预处理

首先对数据集进行去重操作，从而删除外卖平台自动为客户所做出的评论。经去重后，发现数据数量没有发生变化，仍为 11987 条数据，说明不存在重复数据。对数据去重后进行数据清洗，将数字、字母及外卖平台等字样都进行删除。

表 1 外卖评论数据集展示

	label	review
0	pos	很快，好吃，味道足，量大
1	pos	没有送水没有送水没有送水
2	pos	非常快，态度好。
3	pos	方便，快捷，味道可口，快递给力
4	pos	菜味道很棒！送餐很及时！

	label	review
11982	neg	以前几乎天天吃，现在调料什么都不放，
11983	neg	昨天订凉皮两份，什么调料都没有放，就放了点麻油，特别难吃，丢了一份，再也不想吃了
11984	neg	凉皮太辣,吃不下都
11985	neg	本来迟到了还自己点！！！
11986	neg	肉夹馍不错，羊肉泡馍酱肉包很一般。凉面没想象中好吃。送餐倒是很快。

```
RangeIndex: 11987 entries, 0 to 11986
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   label   11987 non-null  object
 1   review  11987 non-null  object
```

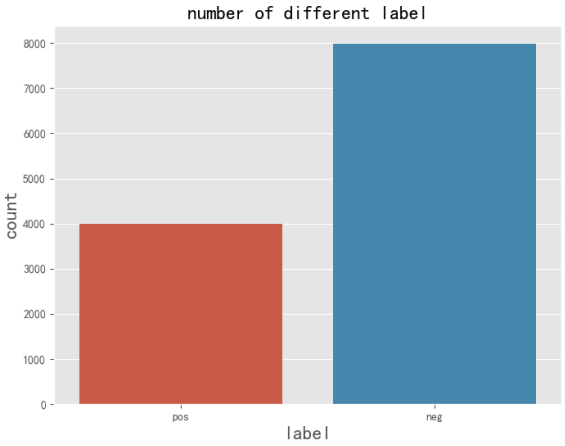

图 1 外卖评论数据集分布描述

```
清洗之前：
今天师傅是不是手抖了，微辣格外辣！
----------
送餐快,态度也特别好,辛苦啦谢谢
----------
超级快就送到了，这么冷的天气骑士们辛苦了。谢谢你们。麻辣香锅依然很好吃。
----------
经过上次晚了2小时，这次超级快，20分钟就送到了……
----------
清洗之后：
今天师傅是不是手抖了，微辣格外辣！
----------
送餐快,态度也特别好,辛苦啦谢谢
----------
超级快就送到了，这么冷的天气骑士们辛苦了。谢谢你们。麻辣香锅依然很好吃。
----------
经过上次晚了小时，这次超级快，分钟就送到了……
----------
```

图 2 数据清洗前后对比

其次进行分词处理，分词处理将原数据集的 11987 条长评论切分成了 203031 个词汇并输出一个数据框，其中包含分词的词汇、对应的词性、分词词汇所在原评论的编号、分词词汇所在原评论的情感类型，从而将外卖评论这一非结构化数据转化为结构化数据。输出分词操作的第一步结果，通过图 3 可以看出第一条评论被划分为 9 个词，分别为“很快”、“，”、“好吃”、“，”、“味道”、“足”、“，”、“量”、“大”，每个分词词汇后面的字母即为对应的词性，此处所用词性表为《结巴分词词性对照表》。最终分词结果如表 2 所示。

```
[('很快', 'd'), (', ', 'x'), ('好吃', 'v'), (', ', 'x'), ('味道', 'n'), ('足', 'a'), (', ', 'x'), ('量', 'n'), ('大', 'a')]
----------
[('没有', 'v'), ('送水', 'v'), ('没有', 'v'), ('送水', 'v'), ('没有', 'v'), ('送水', 'v')]
----------
[('非常', 'd'), ('快', 'a'), (', ', 'x'), ('态度', 'n'), ('好', 'a'), ('。', 'x')]
----------
[('方便', 'a'), (', ', 'x'), ('快捷', 'a'), (', ', 'x'), ('味道', 'n'), ('可口', 'v'), (', ', 'x'), ('快', 'a'), ('递给', 'v'), ('力', 'n')]
----------
[('菜', 'n'), ('味道', 'n'), ('很棒', 'a'), ('！', 'x'), ('送餐', 'v'), ('很', 'd'), ('及时', 'c'), ('！', 'x')]
----------
```

图 3 分词结果

表 2 分词结果

	index_content	word	nature	content_type
0	1	很快	d	pos
1	1	,	x	pos
2	1	好吃	v	pos
3	1	,	x	pos
4	1	味道	n	pos

	index_content	word	nature	content_type
203026	11987	送餐	v	neg
203027	11987	倒	v	neg
203028	11987	是	v	neg
203029	11987	很快	d	neg
203030	11987	。	x	neg

从图中可见分词词汇仍包含大量标点符号和停用词等，通过导入停用词词汇表来将标点符号和停用词进行去除，最终得到 88128 个具有实义的分词词汇，如表 3 所示。表格的第一列为该分词在 203031 个原始分词中所在的位置编号，第二列 index_content 为该分词词汇所在评论的编号，第三列 word 为分词词汇，第四列 nature 为该分词词汇在结巴分词词性对照表中的词性，第五列 content_type 为该分词词汇所在评论的情感，第六列 index_word 为分词词汇在评论中的位置。对分词词汇进行词频统计，并用词云图对分词词汇进行展示，图 4 中左图为全部分词词汇的词云图，右图为仅挑选名词后的分词词汇词云图。。

表 3 数据预处理结果

	index_content	word	nature	content_type	index_word
4	1	味道	n	pos	3
7	1	量	n	pos	5
18	3	态度	n	pos	1
25	4	味道	n	pos	2
30	4	力	n	pos	5

图 4 词云图展示数据预处理后的分词词汇

三、对情感分类进行预测

1、基于情感词典的情感分类预测

采用知网发布的情感分析词语集，包括“正面评价词语”、“负面评价词语”、“正面情感词语”、“负面情感词语”四个 txt 文件，其中分别含有 3743、3138、833、1251 个词汇。将正向情感的词汇的权重记为 1，负向情感的词汇的权重记为-1，将情感词汇合并到分词词汇中并赋予对应权重；若在情感词汇表中找不到某个分词词汇，则将该分词词汇的权重赋值为 0。

接下来，需对分词词汇进行情感倾向修正。若某条评论具有多重否定词组，则该词组若有奇数个否定词则分词词汇为否定意味，若有偶数个否定词则为肯定意味。我们需要找到某个分词词汇在其所在评论语句的前两个词汇，通过判断这两个词汇中否定词的数量来对分词词汇进行情感倾向修正。若该分词词汇位于句首，则算作没有否定词汇；若位于第二个词，则只需判断前一个词是否为否定词。具体的否定词也由知网发布的否定词词汇进行导入，包括“不”、“没”、

“无”、“非”、“莫”等。进行情感倾向修正后，可以得到每个分词词汇的情感值如图 7 所示，其中 amend_weight 列代表的即为情感倾向修正后的情感值得分。

表 4 情感倾向修正后的情感值得分

	index_content	word	nature	content_type	index_word	weight	amend_weight
0	1	很快	d	pos	1	1.0	1.0
1	1	好吃	v	pos	2	0.0	0.0
2	1	味道	n	pos	3	0.0	0.0
3	1	足	a	pos	4	0.0	0.0
4	1	量	n	pos	5	0.0	0.0

表 5 添加新词后的情感值得分及正负情感判断

	index_content	word	nature	content_type	index_word	weight	amend_weight	ml_type
0	1	很快	d	pos	1	1.0	1.0	pos
1	1	好吃	v	pos	2	1.0	1.0	pos
2	1	味道	n	pos	3	0.0	0.0	pos
3	1	足	a	pos	4	0.0	0.0	pos
4	1	量	n	pos	5	0.0	0.0	pos

但从表 4 中可以看出，许多情感词汇没有被正确判断，如“好吃”的权重为 0，而很明显这一分词词汇是积极情感词汇，这说明我们所采用的情感词典不够全面，因此我们选择通过手动添加积极词汇的方法来完善情感词典从而进行校正，并得到表 5 所示结果。将 amend_weight 按照 Index_content 相加即可得到每条评论的情感值得分，若情感值得分大于 0 则判断为积极情感，ml_type 列记为“pos”，若情感值得分小于 0 则判断为消极情感“neg”，若情感值得分为 0 则不对该评论的 ml_type 赋值。最后输出混淆矩阵和预测结果分析，包括精确率、召回率、f1 值等数据。从分析结果可以看到，预测的结果并不十分理想，尽管我们所构建的情感词典对消极评论预测的准确率达到了 91%，但对积极评论预测的准确率仅有 67%，仅仅略胜于随机分类。这一结果很可能是因为我们的情感词典不够完善。在实际应用中，情感词典很难对所研究领域的特定词汇进行全面覆盖。同时由于汉语表达的复杂性、多样性、丰富性，由于部分用户评论中错别字的存在，由于网络热词正呈爆炸式增长的态势，构建全面的情感词典愈加困难，而手动添加情感词的方法也不适用与庞大的数据集，因此这一方法仍有改进和优

化的空间。

表 6 基于情感词典进行情感分类预测的混淆矩阵

ml_type content_type	neg	pos	All
neg	1376	933	2309
pos	128	1860	1988
All	1504	2793	4297

	precision	recall	f1-score	support
neg	0.91	0.60	0.72	2309
pos	0.67	0.94	0.78	1988
accuracy			0.75	4297
macro avg	0.79	0.77	0.75	4297
weighted avg	0.80	0.75	0.75	4297

图 5 基于情感词典进行情感分类预测的分析结果

图 6 积极情感词汇词云图

图 7 消极情感词汇词云图

2、基于决策树的情感分类预测

决策树是通过一系列问题的诊断结果来进行决策的，决策树的生成一般是从根节点开始，选择对应的特征，然后选择该节点特征的分裂点，根据分裂点分裂节点。

将每条评论的分词词汇和情感标签分别保存在 X 和 Y 两个变量中，并进行训练集和测试集的划分。构建决策树模型，通过输出结果可以得到决策树在训练集上的准确率为 0.76，在测试集上的准确率为 0.75，表现较为稳定。将决策树在测试集上的预测分类表现通过图 8 进行展示，可以看出决策树模型对消极情感的预测分类结果要优于积极情感的预测分类结果，这可能是由于原始数据中消极情感的评论数量远远多于积极情感的评论数量，因此对于积极情感分类的训练程度不够。

	precision	recall	f1-score	support
neg	0.77	0.89	0.83	1609
pos	0.67	0.45	0.54	781
accuracy			0.75	2390
macro avg	0.72	0.67	0.68	2390
weighted avg	0.74	0.75	0.73	2390

图 8 基于决策树进行情感分类预测的分析结果

通过 python 的第三方库 graphviz 将决策树模型通过可视化的决策树图形进行直观展示。由于决策树图形较宽、难以全部在报告中进行展示，故仅截取一半决策树图形进行展示，完整的决策树图形将以 svg 格式作为附件附在文件夹中。决策树的节点所出现的词语为较为重要的分词词汇，若满足节点处的判断条件则进入左子树，否则进入右子树。节点同样展示了左子树和右子树分别的样本个数，样本量最大的一个子树为不包含“不错”、“好吃”、“很快”、“辛苦”、“小时”等关键词的评论。

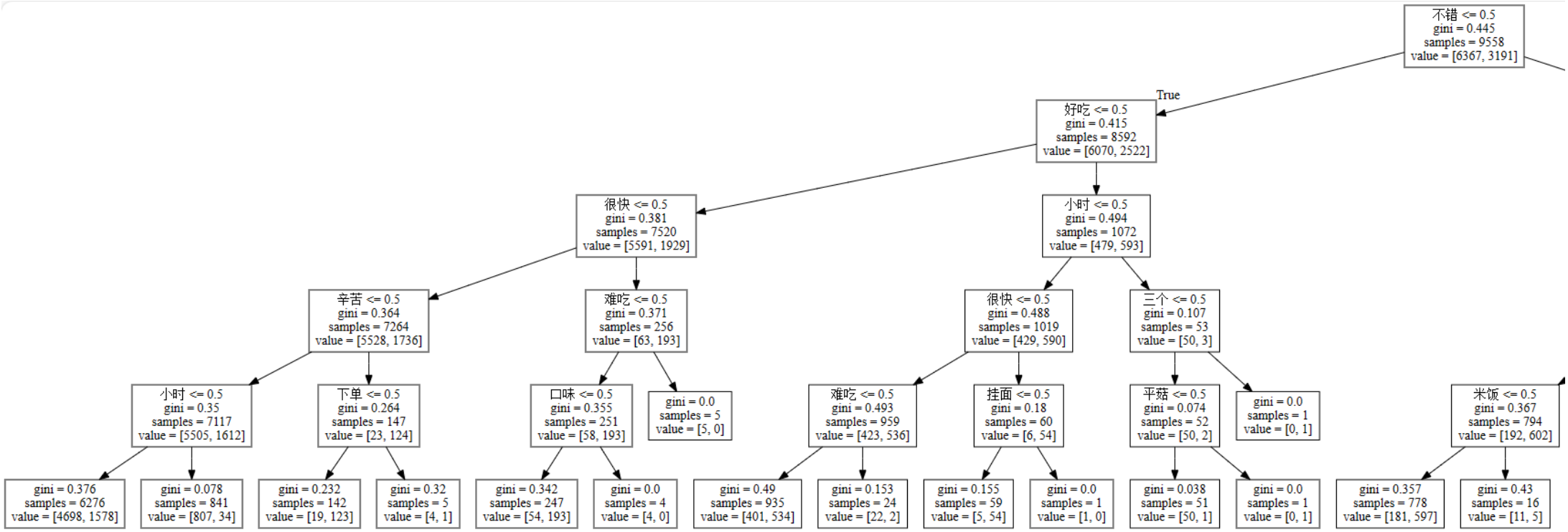
不错 <= 0.5
gini = 0.445
samples = 9558
value = [6367, 3191]
True
好吃 <= 0.5
gini = 0.415
samples = 8592
value = [6070, 2522]
很快 <= 0.5
gini = 0.381
samples = 7520
value = [5591, 1929]
小时 <= 0.5
gini = 0.494
samples = 1072
value = [479, 593]
辛苦 <= 0.5
gini = 0.364
samples = 7264
value = [5528, 1736]
难吃 <= 0.5
gini = 0.371
samples = 256
value = [63, 193]
很快 <= 0.5
gini = 0.488
samples = 1019
value = [429, 590]
三个 <= 0.5
gini = 0.107
samples = 53
value = [50, 3]
小时 <= 0.5
gini = 0.35
samples = 7117
value = [5505, 1612]
下单 <= 0.5
gini = 0.264
samples = 147
value = [23, 124]
口味 <= 0.5
gini = 0.355
samples = 251
value = [58, 193]
gini = 0.0
samples = 5
value = [5, 0]
难吃 <= 0.5
gini = 0.493
samples = 959
value = [423, 536]
挂面 <= 0.5
gini = 0.18
samples = 60
value = [6, 54]
平菇 <= 0.5
gini = 0.074
samples = 52
value = [50, 2]
gini = 0.0
samples = 1
value = [0, 1]
米饭 <= 0.5
gini = 0.367
samples = 794
value = [192, 602]
gini = 0.376
samples = 6276
value = [4698, 1578]
gini = 0.078
samples = 841
value = [807, 34]
gini = 0.232
samples = 142
value = [19, 123]
gini = 0.32
samples = 5
value = [4, 1]
gini = 0.342
samples = 247
value = [54, 193]
gini = 0.0
samples = 4
value = [4, 0]
gini = 0.49
samples = 935
value = [401, 534]
gini = 0.153
samples = 24
value = [22, 2]
gini = 0.155
samples = 59
value = [5, 54]
gini = 0.0
samples = 1
value = [1, 0]
gini = 0.038
samples = 51
value = [50, 1]
gini = 0.0
samples = 1
value = [0, 1]
gini = 0.357
samples = 778
value = [181, 597]
gini = 0.43
samples = 16
value = [11, 5]

图 9 决策树部分图形

3、基于逻辑斯蒂回归进行情感分类预测

首先要先确定最优的惩罚项和对应的惩罚系数，分别测试 11 惩罚项和 12 惩罚项在不同惩罚系数下的预测表现，并绘制图 10。从图 10 中可以看出，12 惩罚项在测试集上的表现要优于 11 惩罚项，但仅仅从图中难以看出最优的惩罚项系数。输出 11 惩罚项在不同惩罚项系数下的准确率最大值与 12 惩罚项在不同惩罚项系数下的准确率最大值并进行比较，可以看出 12 惩罚项在测试集的最优表现的确优于 11 惩罚项在测试集的最优表现，输出 12 惩罚项在测试集的最优表现对应的惩罚项系数。

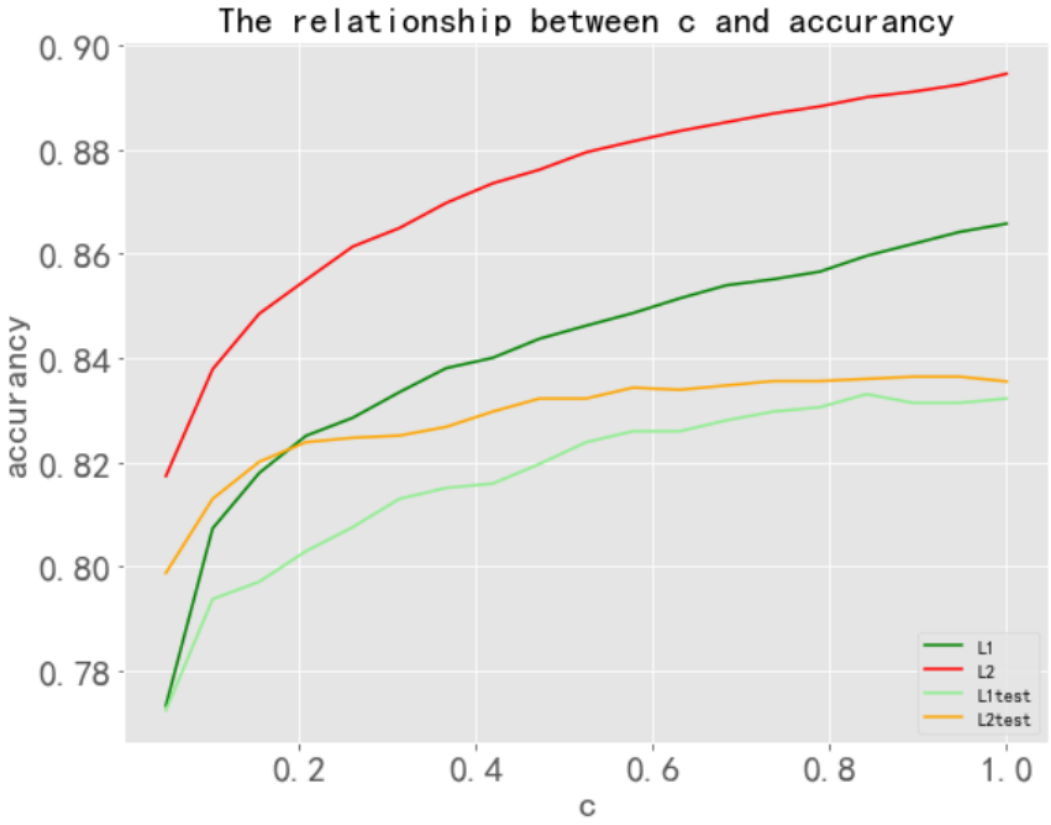

图 10 惩罚项系数与预测准确率的关系

```
max value of accurancy with l1 penalty:0.8330
max value of accurancy with l2 penalty:0.8384
the best value of c with l2 penalty:0.9
```

图 11 11 惩罚项最大值、12 惩罚项最大值与最优惩罚项系数

```
accuracy_train:0.8911
accuracy_test:0.8368
precision:0.7931
recall:0.6773
f1:0.7307
```

图 12 逻辑斯蒂回归的准确率、精确度、召回率、F1 分数值

分析逻辑斯蒂回归分类器表现的情况。首先，逻辑斯蒂回归分类器在训练集上的准确率为 0.8911，在测试集上的准确率为 0.8368，表现优于决策树分类器。精确率（precision）、召回率（recall）和 F1 分数值的计算公式如下，其中真阳性 TP 表示预测为正、实际为正，假阳性 FP 表示预测为正、实际为负，假阴性 FN 表示预测为负、实际为正。

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

$$F_1 = \frac{2 * precision * recall}{precision + recall}$$

从公式可以看出，精确率衡量的是预测为正的个体中正确预测的个数，召回率衡量的是实际为正的个体中正确预测的个数，精确率和召回率这两个指标为相互矛盾的指标，因为如果我们希望精确率提升，即降低误报率，但这样一来就会提升漏报率，导致召回率下降。PR 曲线绘制描述了精确率和召回率的关系。因此，我们需要 F1 分数值来综合衡量这两个指标，F1 分数值其实是精确率和召回率的谐波平均值，其值在[0,1]之间，越接近 1 表示分类器性能越好。

绘制 ROC 曲线，其对应的下面积（AUC 值）也可评估模型性能，越接近 1 说明分类器性能越好。同时还可通过分类分析结果来具体查看模型性能。

将训练好的逻辑斯蒂回归分类器应用到全部数据集上进行情感预测分类，得到分类分析结果及混淆矩阵如图 15、图 16 所示。

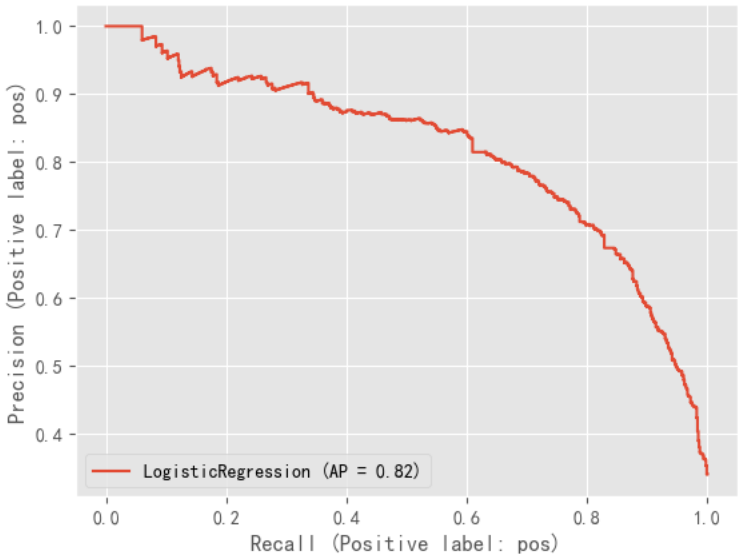

图 13 PR 曲线

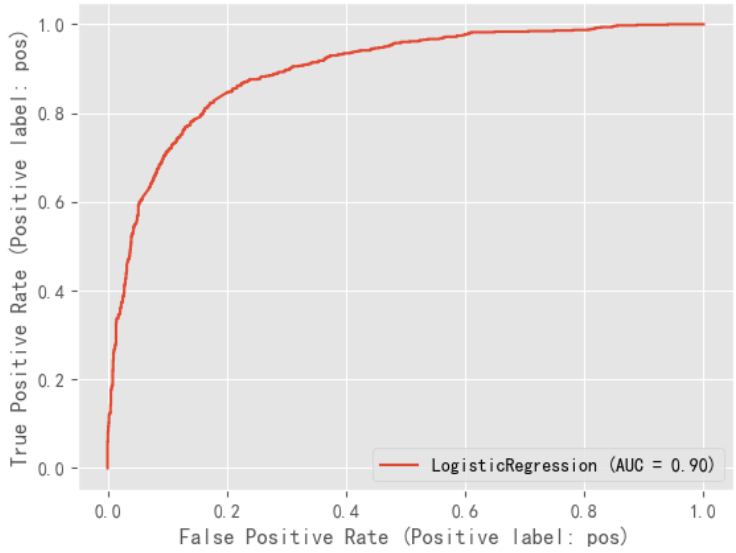

图 14 ROC 曲线

	precision	recall	f1-score	support
neg	0.85	0.91	0.88	1609
pos	0.79	0.68	0.73	781
accuracy			0.84	2390
macro avg	0.82	0.80	0.81	2390
weighted avg	0.83	0.84	0.83	2390

图 15 分类分析结果

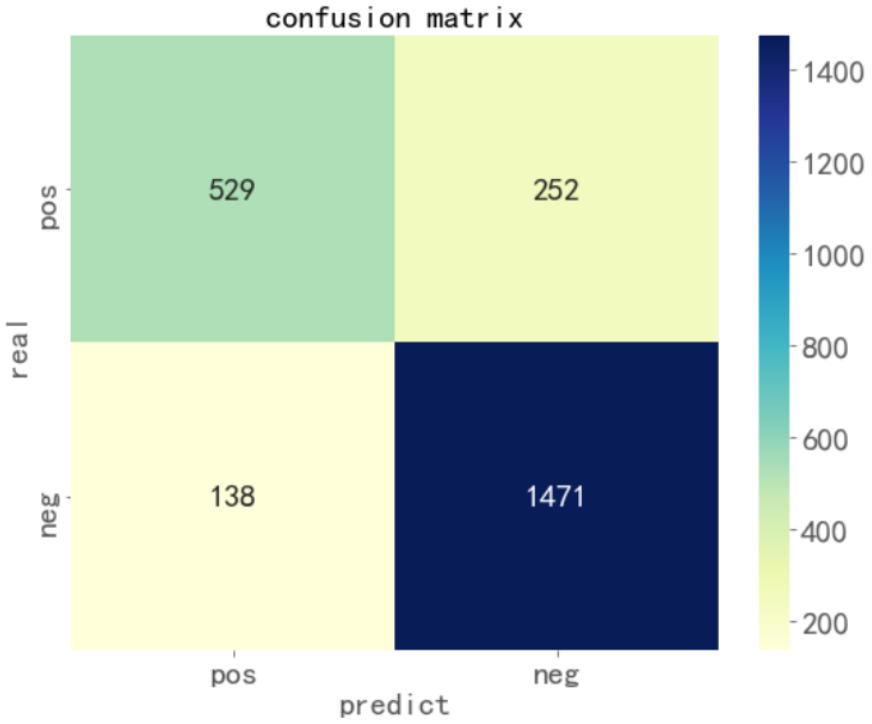

图 16 逻辑斯蒂分类器混淆矩阵

四、基于 LDA 的主题分析

LDA（Latent Dirichlet Allocation, LDA）主题模型是一种基于贝叶斯思想的无监督聚类算法，广泛运用于文本分析等场景。作为一种生成式主题模型，LDA 认为每一篇文档的每一个词都是通过一定的概率选择某个主题，并从这个主题中以一定的概率选择了某个词语，因此 LDA 主题模型包括了文档（d）、主题（z）、词语（w）三层结构。LDA 算法的输入是一个文档的集合 $D=\{d_1,d_2,d_3,\cdots,d_n\}$，同时还需要输入主题的类别数量 k，LDA 算法会给出每一篇文档在所有 Topic 上的对应概率值，即得到概率的集合 $d_i=\{d_{p1},d_{p2},\cdots,d_{pk}\}$，表示文档 d_i 在 k 个 topic 上的概率值；算法同样会给出文档中每个词对应每个 Topic 的概率，$w_i=\{w_{p1},w_{p2},\cdots,w_{pk}\}$，这样就以主题为中介得到了两个矩阵，一个矩阵为从文档到主题的概率分布矩阵，另一个矩阵为从词汇到主题的概率分布矩阵。

LDA 主题模型的核心公式为

$$p(w|d)=p(w|t)*p(t|d),$$

其中 $p(w|d)$ 是可观测到的，而等号右边的两个条件概率则是无法直接观测到的。直观而言，这个核心公式就是以主题作为中间层，将一个条件概率转化为两个条件概率。这两个条件概率分别可以通过 θ_d 和 φ_t 作为参数进行计算，其中 $p(t|d)$ 利用 θ_d 计算得到， $p(w|t)$ 利用 φ_t 计算得到。因此，利用当前的 θ_d 和 φ_t，我们可以为某个文档中的某个单词计算它对应任意一个主题时的 $p(w|d)$，然后根据这些结果来更新这个词所对应的主题。然后，如果这个更新改变了这个单词所对应的主题，就会反过来影响 θ_d 和 φ_t 的取值。

LDA 算法开始时，先随机地对所有的 d 和 t 给 θ_d 和 φ_t 赋值，不断重复上述过程，直到最终收敛，而收敛的结果就是 LDA 的输出结果。这样 LDA 算法，就将文档和词语投射到了一组主题上，试图通过主题找出文档与词语间、文档与文档间、词语与词语之间潜在的关系。由于 LDA 属于无监督算法，每个主题并没有指定条件，在聚类后，通过统计出各个主题上词语的概率分布，最终得到在某个主题上概率较高的词，这些词语可以非常好地描述该主题的意义。

进行 LDA 主题分析时，首先要对分词词汇进行去重从而建立词库和语料库。在建立词库和语料库后，由于 LDA 要求提供主题数作为参数，因此需要寻找最优主题数。具体步骤为：

（1）取初始主题数 k，得到初始模型，计算各主题之间相似度，其中相似度用平均余弦距离衡量，计算公式为：

$$\cos\theta = \frac{\sum_{i=1}^{n} A_i B_i}{\sum_{i=1}^{n} (A_i)^2 \sum_{i=1}^{n} (B_i)^2} = \frac{AB}{|AB|} \quad ;$$

（2）增加 k 值，重新训练模型，再次计算各主题之间相似度；

（3）若不满足停止条件，则重复步骤（2）；

（4）比较不同 k 值下的主题相似度，选择最优的 k 值作为最优主题数。

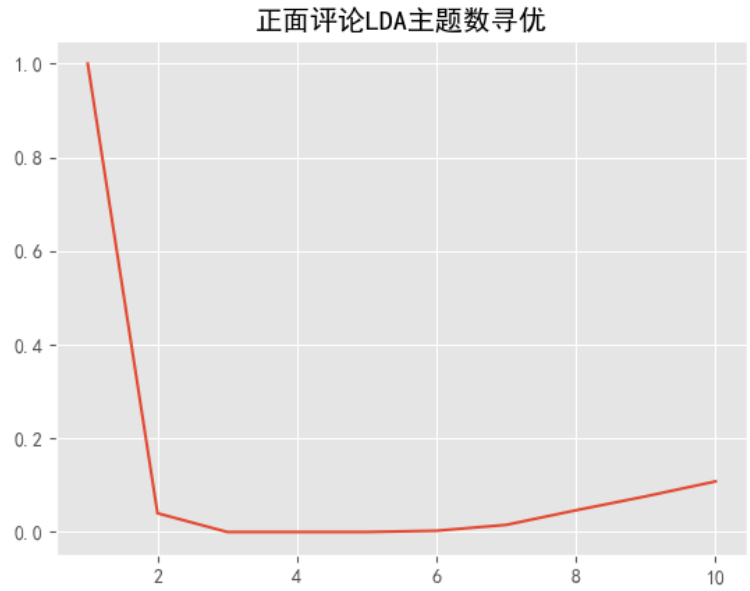

图 17 正面评论 LDA 主题数寻优

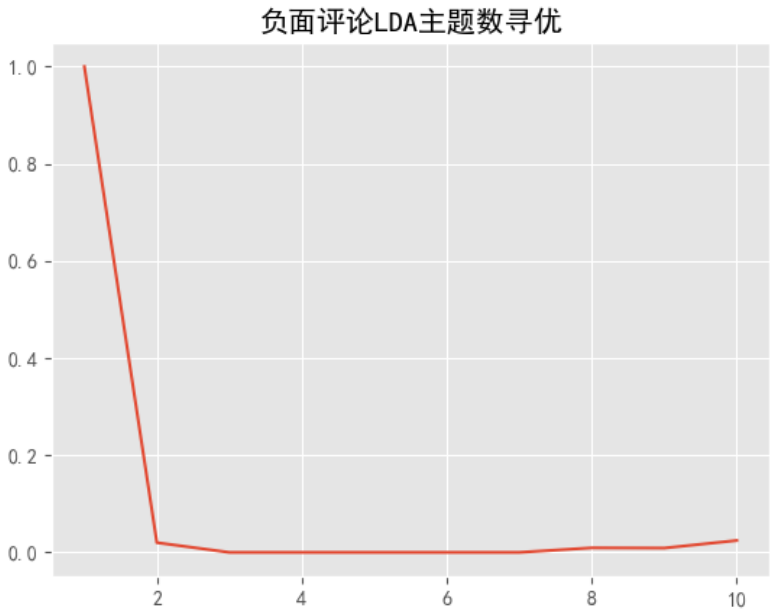

图 18 负面评论 LDA 主题数寻优

图 17 与图 18 分别展示了积极情感与消极情感主题数与余弦相似度之间的关系，可以看出积极情感评论与消极情感评论的最优主题数均为 2。将最优主题数作为参数输入并重新建模，得到四组主题模型，其中前两个主题模型为积极情感的主题模型，后两个主题模型为消极情感的主题模型。可以看出，第一个主题模型主要是围绕送餐时间快，第二个主题模型主要是围绕食物味道好，均为积极情

感；第三个主题模型主要围绕送餐速度慢、送餐态度差，第四个主题模型主要是围绕食品味道差、品质差，均为消极情感。

积极情感主题模型：

模型1= 0.028×“送”+ 0.027×“吃”+ 0.016×“速度”+
0.015×“饭”+ 0.014×“满意”+ 0.013×“外卖”+
0.012×“送到”+ 0.011×“辛苦”+ 0.010×“时间”
0.009×“很快”

模型2= 0.110×“好吃”+ 0.030×“味道”+ 0.028×“太”+
0.026×“菜”+ 0.026×“美味”+ 0.018×“点”+
0.016×“不错”+ 0.016×“店家”+ 0.013×“还好”
0.012×“小时”

消极情感主题模型：

模型3= 0.045×“送餐”+ 0.043×“小时”+ 0.038×“送”+
0.025×“点”+ 0.020×“速度”+ 0.018×“太慢”+
0.018×“态度”+ 0.015×“时间”+ 0.015×“凉”
0.014×“打电话”

模型4= 0.069×“味道”+ 0.045×“慢”+ 0.021×“难吃”+
0.019×“不好”+ 0.018×“太”+ 0.014×“垃圾”+
0.014×“送来”+ 0.013×“饭”+ 0.012×“坑”
0.012×“包装”

五、总结

本报告围绕外卖评论进行了文本挖掘和分析，采用了三种不同的方法进行情感分类预测，并通过 LDA 主题模型提取了不同主题的关键词进行语义描述。报

告内容较为完整，但仍有很大的改进空间。

首先，本文采用的情感词典不够完善，导致很多分词词汇无法正确判断情感的政府。其次，本文采用的分类器均较为简单且种类偏少，还可以采用高斯贝叶斯分类器、支持向量机、随机森林、Xgboost 等分类器或集成方法进行训练和学习。第三，本文在 LDA 主题模型部分仅仅是简单的对 LDA 主题模型进行调用，目前有很多方法对 LDA 主题模型进行了改进，并结合了机器学习的方法，可以得到表现更优的结果，