

第二十四届大学生科技节数学建模大赛

承 诺 书

我们仔细阅读了第二十四届大学生科技节数学建模大赛的有关注意事项。

我们完全明白，在竞赛开始后参赛队员不能以任何方式（包括电话、电子邮件、网上咨询等）与队外的任何人（包括教师）研究、讨论与赛题有关的问题。

我们知道，抄袭别人的成果是违反竞赛规则的，如果引用别人的成果或其他公开的资料（包括网上查到的资料），必须按照规定的参考文献的表述方式在正文引用处和参考文献中明确列出。

我们郑重承诺，严格遵守竞赛规则，以保证竞赛的公正、公平性。如有违反竞赛规则的行为，我们将受到严肃处理。

参赛队伍信息表（用电子格式填写）

| 姓名 | 学校 | 院系 | 手机 | 电子邮箱 |
|---|-----------------|-----------|-------------|-------------------|
| 易俊哲 | 华中科技大学 | 电子信息与通信学院 | 18322802285 | 781988375@qq.com |
| 齐子豪 | 华中科技大学 | 电子信息与通信学院 | 13460151341 | 2196248593@qq.com |
| 刘玥柔 | 华中科技大学 | 机械科学与工程学院 | 19398506356 | 971694657@qq.com |
| 队伍名 | 易俊哲 18322802285 | | 组别 | 兴趣组 |
| <p style="text-align: center;">注意事项</p> <p>1、队伍名为“队长姓名+队长手机号”，如“张三 12345678910”。</p> <p>2、组别为“专业组/兴趣组”</p> <p>3、请将这两页粘贴到论文的最前面，作为论文的封面（非常重要）。</p> | | | | |

以下内容请用黑色签字笔填写

| | | | | |
|--|-----|-----|-----|--|
| 本队选择的赛题为： <input type="checkbox"/> D 题 <input type="checkbox"/> E 题 <input type="checkbox"/> F 题（专业组） | | | | |
| <input type="checkbox"/> A 题 <input type="checkbox"/> B 题 <input checked="" type="checkbox"/> C 题（兴趣组） | | | | |
| 队员签字 | 刘玥柔 | 易俊哲 | 齐子豪 | |

第二十四届大学生科技节数学建模大赛

编 号 专 用 页

本页用于论文评阅，只需打印出来，不要在本页填写任何内容！

| 评委 | 评委 1 | 评委 2 | 评委 3 |
|------|------|------|------|
| 评分 | | | |
| 备注 | | | |
| 最终得分 | | | |
| 评价 | | | |

社交媒体上文本内容流行度的预测与分析

摘 要

对于网络媒体公司，建立一个准确的数学模型从海量的文本信息中预测哪些内容更容易受到大众的传播对内容推荐和广告投放等战略决策至关重要，本文针对以上问题，根据新闻文本特征属性、传播量等数据，并基于随机森林回归预测的方法，对更容易受到大众传播的新闻的属性进行了详细分析，建立了基于新闻文本和其他已知属性预测传播量的数学模型。

问题一中，用 Python 的 Pandas 库中的 **describe** 方法对数据进行了统计分析，得出了所有变量的统计描述信息，观察到数据集中已经对分类特征进行了 **One-Hot 编码**，对偏斜特征进行**对数转换**。进一步简化数据集，根据所得的变量统计描述信息，用箱线图中的**四分位距**思想处理异常值和极端值。初步筛选后分析变量和目标属性之间的相关性，引入 **Pearson 相关系数**，求解相关系数阵后将传播量一列可视化，根据 Pearson 相关系数的大小筛选接下来需要分析的变量属性。

问题二中，利用**随机森林回归算法**预测传播量，引入了**决定系数 R^2** 、**对数均方根误差 $\log\text{-RMSE}$** 、**实际均方根误差 Actual-RMSE** 用于评估模型的性能，再从庞大的预测样本库中随机抽取部分样本进行预测值和测试数据的对比可视化，得到较好的拟合效果。

问题三中，首先通过**热力图**可视化变量之间的相关性，对传播量进行 **One-Hot 编码**，将原始数据中的传播量转换为一个二元特征，再根据**特征聚合**的思想分析具有相似语义特征的变量，利用**百分比堆叠柱状图**直观显示部分变量和传播量的统计规律。

问题四中，综合所得结论，通过写信的方式给网络媒体公司提供建议。

关键词： 四分位距 随机森林回归算法 One-Hot 编码 相关系数 热力图

一、问题重述

1.1 问题背景

新闻传播的流行度预测对网络媒体至关重要。准确预测新闻内容的流行度有助于网络媒体公司更好地把握受众需求、提高内容推荐效率、优化广告投放策略，甚至为战略决策提供可靠的参考。在当今社交媒体时代，用户生成的内容（UGC）不断涌现，在影响着新闻传播的走向和影响力的同时，也加大了新闻传播的流行度预测的难度。为解决上述难题，建立一个精确的预测模型势在必行。

1.2 问题要求

附件提供的数据包括了新闻的 URL、发布时间与数据收集时间的时间差、标题和内容的单词数量、LDA 特征、链接图片数量等特征。这些特征可以用于分析新闻的流行度和传播效果。

问题一：依据所提供的数据进行数据处理与统计的工作，并且指出数据的统计特征。

问题二：利用在问题一中得到处理好的数据后进行建模，预测文章的传播量，并分析模型的表现效果。

问题三：结合前面两问的结果，回答大众更倾向于传播的新闻的类型，给出对新闻行业的启示。

问题四：假设你是在一家专注于人工智能和大数据分析的公司中，被一家网络媒体公司委托进行新闻传播的流行度预测，数据由该公司提供。综合所得成果，给该公司写一封建议信。

二、问题分析

2.1 问题一的分析

针对问题一，先观察分析附件的数据，可知这是来自 Mashable 网站的 39,643 篇新闻文章的数据集，附件中每个 URL 对应的网站都有 61 个属性，目标属性为传播量“shares”；接着对该数据进行预处理，分析出这些数据的统计特征，并对这些数据进行处理。经简单的观察与分析，易知数据过于庞大，且部分数据失真，存在个例极端的情况，若直接利用原始数据预测将会产生模型运行过载、预测准确性降低、整体的统计规律被个例忽视等消极影响。故数据处理包括：简化数据集、缺失值和不可信值的处理、深入处理异常值和极端值。

2.2 问题二的分析

针对问题二，其建立模型的目的在于预测传播量，从题目所给的数据看出，本题是一道单目标变量多特征的回归问题，因此我们选择建立随机森林的模型来对传播量进行预测。随机森林能够处理很高维度的数据，这正好是契合本题数据状态的；当然随机森林在训练的过程中完成搭建，不用人工进行特征选择，这是简便高效的；并且随机森林是多个决策树并行处理，对于这种数据量大的题目是十分受宠的；随机森林是一个过程清晰的，可以实现可视化的模型，是便于分析的。当然预测的准确度也是相当重要的，我们考虑到回归问题的实质，于是选用了评估模型性能常见的指标：决定系数 R^2 ；还有回归任务中的一种评判指标：实际均方根误差，但是考虑到本题部

分数取值范围大，并且在数据处理的时候，发现像文章中的最大关键词次数、文章中的最小关键词次数等有效特征与传播量的几何关系具有明显的偏向性，于是我们也引入了对数均方根误差进行评判模型质量。最后，由于前面的评判标准都是不可视的，缺少一定的直观性，再加上随机森林可视化的特征，可以分别做出样本预测值与实际值的变化曲线，进行直观地对比。

2.3 问题三的分析

针对问题三，引入 One-Hot 编码将原始数据中的传播量转换为一个二元特征，同时聚焦变量之间的相关性可视化热力图以及语义上的相似性，对自变量的影响有特征聚合分析，从而得出容易流行的新闻的特点。

2.4 问题四的分析

针对问题四，要求综合前面几问所得成果，给委托你预测新闻传播流行度的一家网络媒体公司写一封建议信。首先对调研背景、数据来源与分析、模型建立与验证进行简短的介绍，并由此引出建议的具体内容；接着从新闻发布的时间、类型、文风以及广告投放和战略决策几个方面提出建议；最后表达与甲方公司合作的期待以及对两家公司携手并进的展望。

三、模型假设

1. 假设人们的思维方式已经定型，即数据集中的两年时间段内任何时间同一个人对于不同新闻的偏好程度一致，人们的观念不会随时间发生改变。
2. 假设数据集中的两年时间段内这些新闻面向的读者基数不变。
3. 假设每条新闻不受推广策略影响。
4. 假设每条数据对应新闻的流行性相互独立。

四、符号说明

| 符号 | 说明 |
|-------------------------|--|
| x | 自变量向量 |
| Y | 因变量向量 |
| D | 自变量 x 的集合 |
| y | 因变量 y 的集合 |
| T | 训练总轮数，通常是一个不太大的常数 |
| t | 当前训练轮数 |
| D_t | 表示 $h_t(\cdot)$ 实际使用的训练样本集 |
| $\theta(\cdot)$ | 指示函数，在 \cdot 为真和假时分别取值为 1, 0 |
| $h_t(\cdot)$ | 表示在 D_t 内该基学习器下对应的因变量 |
| $H^{\text{oob}}(\cdot)$ | 对样本的包外预测，表示 $h_t(\cdot)$ 在 $\cdot \notin D_t$ 时拟合最好的 |
| ϵ^{oob} | Bagging 泛化误差的包外估计 |
| FVU | 未解释方差比例 |
| RSS | 残差平方和 |
| TSS | 总平方和 |
| \hat{y} | 变量 y 的预测值 |

五、模型的建立与求解

5.1 问题一解答

5.1.1 数据统计特征分析

用 Python 数据分析可以得到如下的统计特征：

| | timedelta | n_tokens_title | n_tokens_content | n_unique_tokens | n_non_stop_words | n_non_stop_unique_tokens | num_hrefs | num_self_hrefs |
|--------|-----------|----------------|------------------|-----------------|------------------|--------------------------|-----------|----------------|
| 均值 | 354.53 | 10.40 | 546.51 | 0.55 | 1.00 | 0.69 | 10.88 | 3.29 |
| 标准差 | 214.16 | 2.11 | 471.11 | 3.52 | 5.23 | 3.26 | 11.33 | 3.86 |
| 最小值 | 8.00 | 2.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 最大值 | 731.00 | 23.00 | 8474.00 | 701.00 | 1042.00 | 650.00 | 304.00 | 116.00 |
| 第一四分位点 | 164.00 | 9.00 | 246.00 | 0.47 | 1.00 | 0.63 | 4.00 | 1.00 |
| 第二四分位点 | 339.00 | 10.00 | 409.00 | 0.54 | 1.00 | 0.69 | 8.00 | 3.00 |
| 第三四分位点 | 542.00 | 12.00 | 716.00 | 0.61 | 1.00 | 0.75 | 14.00 | 4.00 |

| num_imgs | num_videos | average_token_length | num_keywords | lifestyle | entertainment | bus | socmed | tech | world |
|----------|------------|----------------------|--------------|-----------|---------------|------|--------|------|-------|
| 4.54 | 1.25 | 4.55 | 7.22 | 0.05 | 0.18 | 0.16 | 0.06 | 0.19 | 0.21 |
| 8.31 | 4.11 | 0.84 | 1.91 | 0.22 | 0.38 | 0.36 | 0.23 | 0.39 | 0.41 |
| 0.00 | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 128.00 | 91.00 | 8.04 | 10.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 0.00 | 4.48 | 6.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 1.00 | 0.00 | 4.66 | 7.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 4.00 | 1.00 | 4.85 | 9.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |

| kw_min_min | kw_max_min | kw_avg_min | kw_min_max | kw_max_max | kw_avg_max | kw_min_avg | kw_max_avg | kw_avg_avg | self_reference_min_shares |
|------------|------------|------------|------------|------------|------------|------------|------------|------------|---------------------------|
| 26.11 | 1153.95 | 312.37 | 13612.35 | 752324.07 | 259281.94 | 1117.15 | 5657.21 | 3135.86 | 3998.76 |
| 69.63 | 3857.99 | 620.78 | 57986.03 | 214502.13 | 135102.25 | 1137.46 | 6098.87 | 1318.15 | 19738.67 |
| -1.00 | 0.00 | -1.00 | 0.00 | 0.00 | 0.00 | -1.00 | 0.00 | 0.00 | 0.00 |
| 377.00 | 298400.00 | 42827.86 | 843300.00 | 843300.00 | 843300.00 | 3613.04 | 298400.00 | 43567.66 | 843300.00 |
| -1.00 | 445.00 | 141.75 | 0.00 | 843300.00 | 172846.88 | 0.00 | 3562.10 | 2382.45 | 639.00 |
| -1.00 | 660.00 | 235.50 | 1400.00 | 843300.00 | 244572.22 | 1023.64 | 4355.69 | 2870.07 | 1200.00 |
| 4.00 | 1000.00 | 357.00 | 7900.00 | 843300.00 | 330980.00 | 2056.78 | 6019.95 | 3600.23 | 2600.00 |

| self_reference_max_shares | self_reference_avg_shares | Monday | tuesday | wednesday | thursday | Friday | Saturday | sunday | weekend | LDA_00 |
|---------------------------|---------------------------|--------|---------|-----------|----------|--------|----------|--------|---------|--------|
| 10329.21 | 6401.70 | 0.17 | 0.19 | 0.19 | 0.18 | 0.14 | 0.06 | 0.07 | 0.13 | 0.18 |
| 41027.58 | 24211.33 | 0.37 | 0.39 | 0.39 | 0.39 | 0.35 | 0.24 | 0.25 | 0.34 | 0.26 |
| 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 843300.00 | 843300.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 0.93 |
| 1100.00 | 981.19 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.03 |
| 2800.00 | 2200.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.03 |
| 8000.00 | 5200.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.24 |

| LDA_01 | LDA_02 | LDA_03 | LDA_04 | global_subjectivity | global_sentiment_polarity | global_rate_positive_words | global_rate_negative_words |
|--------|--------|--------|--------|---------------------|---------------------------|----------------------------|----------------------------|
| 0.14 | 0.22 | 0.22 | 0.23 | 0.44 | 0.12 | 0.04 | 0.02 |
| 0.22 | 0.28 | 0.30 | 0.29 | 0.12 | 0.10 | 0.02 | 0.01 |
| 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | -0.39 | 0.00 | 0.00 |
| 0.93 | 0.92 | 0.93 | 0.93 | 1.00 | 0.73 | 0.16 | 0.18 |
| 0.03 | 0.03 | 0.03 | 0.03 | 0.40 | 0.06 | 0.03 | 0.01 |
| 0.03 | 0.04 | 0.04 | 0.04 | 0.45 | 0.12 | 0.04 | 0.02 |
| 0.15 | 0.33 | 0.38 | 0.40 | 0.51 | 0.18 | 0.05 | 0.02 |

| rate_positive_words | rate_negative_words | avg_positive_polarity | min_positive_polarity | max_positive_polarity | avg_negative_polarity | min_negative_polarity |
|---------------------|---------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| 0.68 | 0.29 | 0.35 | 0.10 | 0.76 | -0.26 | -0.52 |
| 0.19 | 0.16 | 0.10 | 0.07 | 0.25 | 0.13 | 0.29 |
| 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | -1.00 | -1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 0.00 | 0.00 |
| 0.60 | 0.19 | 0.31 | 0.05 | 0.60 | -0.33 | -0.70 |
| 0.71 | 0.28 | 0.36 | 0.10 | 0.80 | -0.25 | -0.50 |
| 0.80 | 0.38 | 0.41 | 0.10 | 1.00 | -0.19 | -0.30 |

| max_negative_polarity | title_subjectivity | title_sentiment_polarity | abs_title_subjectivity | abs_title_sentiment_polarity | shares |
|-----------------------|--------------------|--------------------------|------------------------|------------------------------|-----------|
| -0.11 | 0.28 | 0.07 | 0.34 | 0.16 | 3395.38 |
| 0.10 | 0.32 | 0.27 | 0.19 | 0.23 | 11626.95 |
| -1.00 | 0.00 | -1.00 | 0.00 | 0.00 | 1.00 |
| 0.00 | 1.00 | 1.00 | 0.50 | 1.00 | 843300.00 |
| -0.13 | 0.00 | 0.00 | 0.17 | 0.00 | 946.00 |
| -0.10 | 0.15 | 0.00 | 0.50 | 0.00 | 1400.00 |
| -0.05 | 0.50 | 0.15 | 0.50 | 0.25 | 2800.00 |

5.1.2 对数据进行筛选

对于数据集每个 URL 所对应的网站有 61 个属性，其中包括 1 个目标属性(传播量 “shares”), 2 个非预测特征(文章的 URL “url” 和文章发布与数据集获取之间的天数 “timedelta”)和 58 个预测特征，数据集的原收集者已经进行了初步预处理。例如，对分类特征(如发布日期、文章类别)通过 one-hot 编码模式进行了转换，现考虑对偏斜特征(文章字数)进行**对数转换**，由图 1 的对比可以明显看到，经过对数转换重塑了 X 轴后文章字数总和与传播量的关系规律更加明显。

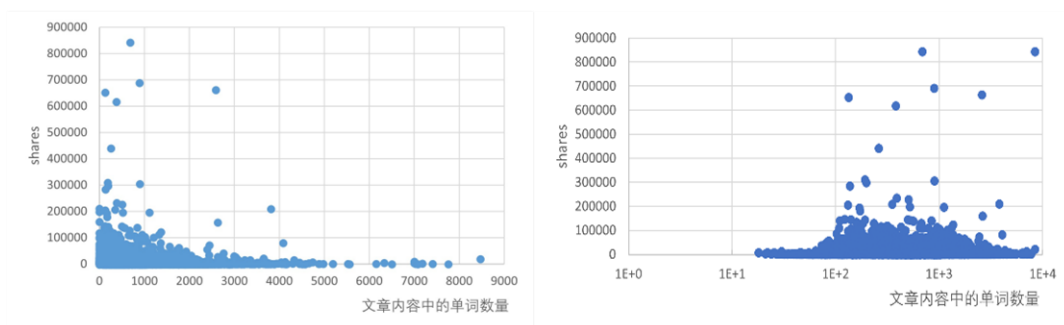


图 1：经过对数转换前后文章内容单词数量和传播量关系

观察数据后可以发现有必要对数据进行进一步的处理，以确保数据集的质量并便于进一步分析。按照如下步骤处理：

(1) 简化数据集:首先，删除 “url” 和 “timedelta” 列。因为 “url” 没有为我们的研究问题提供任何相关信息，故删除 “url”。又根据基本假设：数据集中的两年时间段内任何时间同一个人对于不同新闻的偏好程度一致，人们的观念不会随时间发生改变，“timedelta” 列的数据不会对预测传播量造成影响，故删除 “timedelta”。这一步降低了维数，有利于简化分析。

(2) 缺失值处理：对数据进行检查，没有任何特征包含丢失的数据。

(3) 不可信值处理:在异常值处理过程中，根据特征的性质为每个特征定义可信范围，在数据集中删去超出可信范围的几组数据。

(4) 深入处理异常值和极端值：结合图 2 和传播量的统计特征我们可以看出：传播量的最大值为 84330，最小值为 1，而均值为 3395，可以看出传播量数据中存在很多极端值，这些极端值会对结果产生负面影响，所以要对传播量数据进行筛选，借鉴**箱线图**的思想，用一种简单的标准筛决定哪些值是极端的，即四分位数距离。在描述性统计中，**四分位数范围 (IQR)** 是衡量数据传播或分散程度的一种度量，记传播量数据的上四分位点、中位数、下四分位点为 Q_3 、 Q_2 和 Q_1 ，记四分位距 $IQR=Q_3-Q_1$ 。

记正常分布区间为:

$$[Q1-1.5 \times IQR, Q3+1.5 \times IQR]$$

筛选过后的传播量数据如图 3 所示,可以明显地看出,筛选过后的传播量数据没有出现极端值,更有利于统计规律。

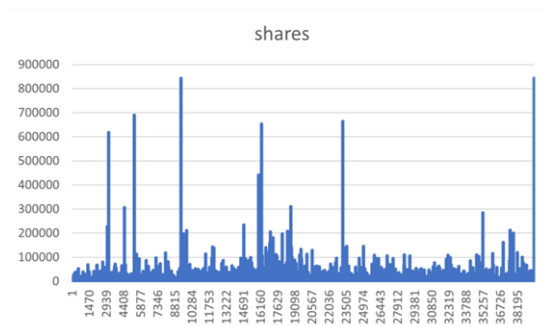


图 2: 筛选前传播量数据

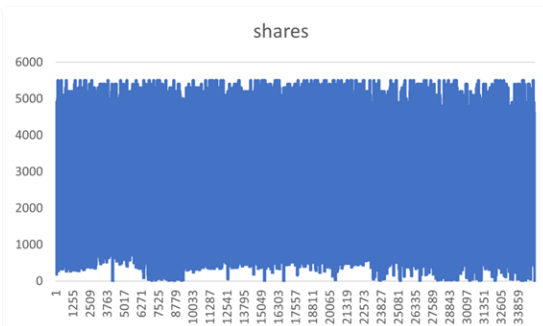


图 3: 筛选后传播量数据

实现对传播量进行准确预测,需要分析特征之间的相关性,考虑变量和目标属性传播量之间的相关性。引入 **Pearson 相关系数**, Pearson 相关系数是一种用于衡量两个变量之间线性相关程度的统计量,通常用符号 r 表示。它的取值范围在-1 到 1 之间。Pearson 相关系数的计算公式如下:

$$r = \frac{Cov(X,Y)}{\sigma(X) \cdot \sigma(Y)}$$

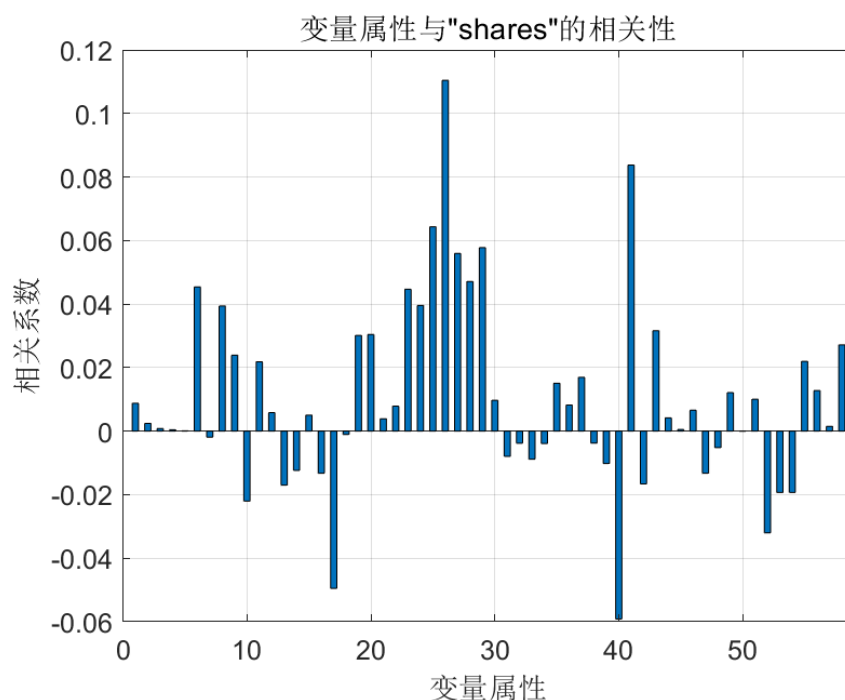


图 4: 变量和传播量之间的相关系数可视化

从相关系数阵可视化的图 4 可以看出一部分变量和传播量之间的相关性很小,可以考虑选取相关系数在 0.01 以上的变量作为最终预测属性。

5.2 问题二解答

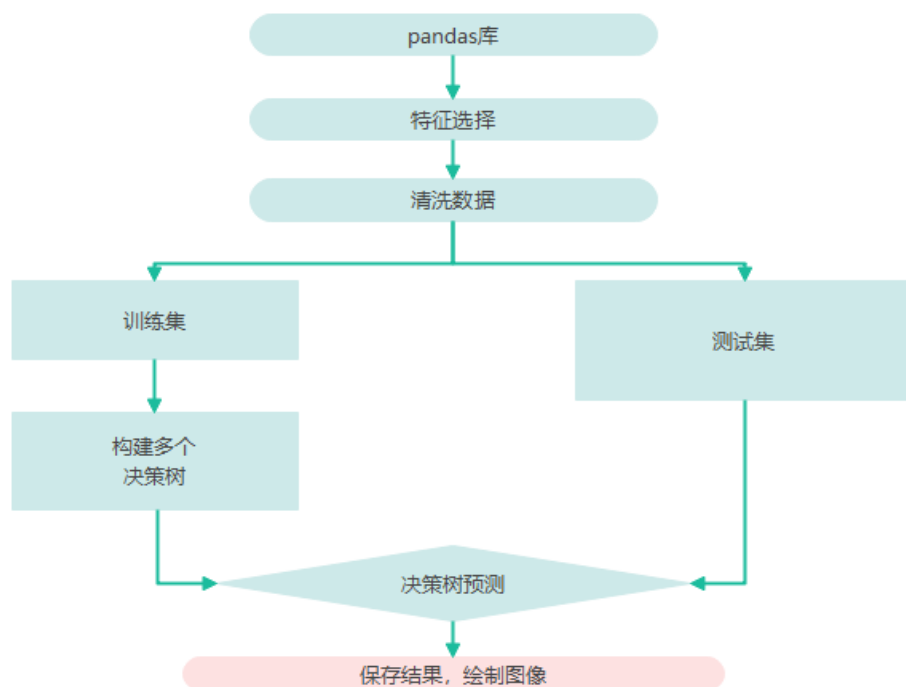


图 5：问题二解决思路流程图

5.2.1 随机森林与 Bagging 思想

问题二是一个预测回归问题，决策树是一种基本的回归方法，但根据处理好的数据可以发现，数据集的特征很多，样本也很多，若采用单个决策树则很容易产生过拟合，因此单个决策树很难适用于本题的数据集，**随机森林**是集成了很多树模型的集成模型，它的基本单元是决策树，所用到的思想是 **Bagging 思想**，从直观的角度来解释，每颗决策树都是一个分类器，随机森林集成了所有的分类投票结果，将投票次数最多的类别指定为最终的输出。

因为随机森林取样的随机性，对于一个训练集，随机森林每次都从 n 个样本中随机挑选样本进行训练，可以计算得到某个样本在 n 次随机选取中被选到的概率为：

$$pi = 1 - \left(1 - \frac{1}{n}\right)^n$$

某一样品在 n 次随机抽取的过程中至少被抽到一次的对立事件是 n 次全都没被抽到，后者的概率为：

$$p1 = \left(1 - \frac{1}{n}\right)^n$$

当样本数量充分大时：

$$pi \approx 1 - \lim_{n \rightarrow \infty} \left(1 - \frac{1}{n}\right)^n \approx 1 - \frac{1}{e} \approx 0.632$$

在 Bagging 的每轮随机采样中，每个基学习器只使用了初始训练集中约 63.2% 的样本，还剩下 36.8% 的样本可以用作验证集来对泛化性能进行“包外估计”。这部分

数据也被称为袋外数据。

$$H^{oob}(x) = \arg \max_{y \in \gamma} \sum_{t=1}^T \vartheta(h_t(x) = y) \cdot \vartheta(x \notin D_t)$$

上式中, $h_t(x)$ 表示 x 在 D_t 内该基学习器下对应的因变量, $\vartheta(h_t(x) = y)$ 表示当 $h_t(x)$ 预测的因变量与实际因变量 y 相等时为 1, 反之为 0; 同理, $x \notin D_t$ 时为 1, 反之为 0。故只有满足 $h_t(x) = y$ 且 $x \notin D_t$ 时相乘才为 1, $\arg \max$ 则取式子 $\sum_{t=1}^T \vartheta(h_t(x) = y) \cdot \vartheta(x \notin D_t)$ 达到最大时 x 向量的取值。故 $H^{oob}(x)$ 为对样本的包外预测, 表示 $h_t(x)$ 在 $x \notin D_t$ 时拟合最好的 x 。

$$\epsilon^{oob} = \frac{1}{|D|} \sum_{(x,y) \in D} \vartheta(H^{oob}(x) \neq y)$$

上式中 x 预测为 $H^{oob}(x)$, $H^{oob}(x)$ 是所有未使用 x 作为训练数据的基学习器 $h_t(x)$ 对 x 做出的最多预测。最后计算所有样本在上一步中的预测错误的占总样本的比例, 即为包外估计。

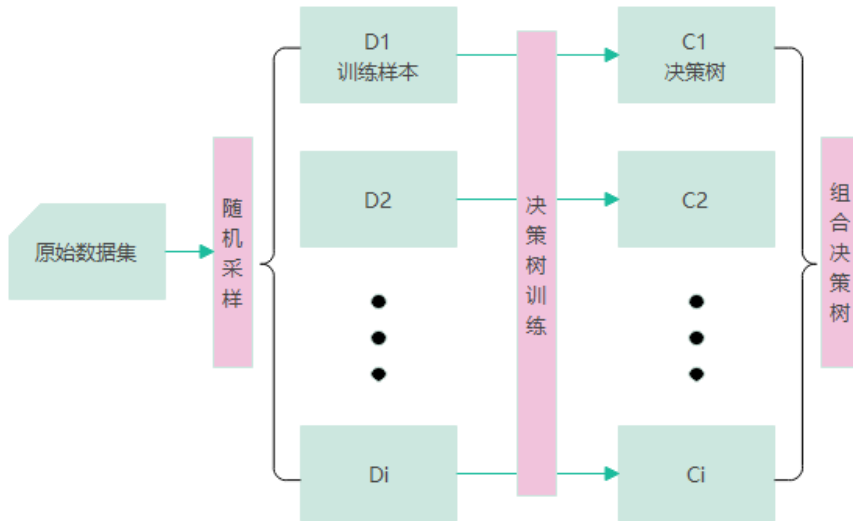


图 6: 随机森林流程图

首先, 通过 Pandas 库读取附件, 并进行一系列数据预处理操作, 包括重命名列名、删除不需要的列、筛选有效数据、对目标变量进行对数转换等。然后, 将数据集划分为训练集和测试集, 测试集所占的比例为 20%。

为检验和阐述模型的表现效果, 我们引入了几个用于评估模型性能的常见指标:

(1) 决定系数 R^2

决定系数 R^2 一般用在回归模型用于评估预测值和实际值的符合程度, R^2 定义如下:

$$R^2 = 1 - FVU = 1 - \frac{RSS}{TSS} = 1 - \frac{\sum_i (y_i - f_i)^2}{\sum_i (y_i - \hat{y})^2}$$

上式中 y_i 是实际值， f_i 是预测值。FVU 为未解释方差比例，RSS 为残差平方和，TSS 为总平方和。一般地， R^2 越接近 1，表示回归分析中自变量对因变量的解释越好。

(2) 对数均方根误差 log-RMSE

对数均方根误差 log-RMSE 是回归任务中的一种评判指标，在某些情况下，特别是当目标变量的范围很大或者分布偏斜时，使用对数误差来衡量模型性能比原始误差可能更合适，其表达式为：

$$\log - RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (\log y_i - \log \hat{y}_i)^2}$$

其中 n 表示样本数量； \log 表示自然对数； y_i 表示第 i 个样本的真实目标值； \hat{y}_i 表示第 i 个样本的预测目标值。

(3) 实际均方根误差 Actual-RMSE

实际均方根误差是衡量模型在原始数据尺度上的预测误差大小的指标。其计算方式与对数均方根误差类似，只是不进行对数转换。

随机森林的各项评估值指标如下：

| 评估指标 | 值 |
|-----------------|---------|
| 训练集 R^2 Score | 0.88 |
| 训练集 Log RMSE | 0.05 |
| 训练集 Actual-RMSE | 504.88 |
| R^2 Score | 0.13 |
| Log RMSE | 0.33 |
| Actual RMSE | 1001.68 |
| 基准 RMSE | 1053.89 |

表 1：随机森林回归性能指标

5.2.2 预测结果

创建一个随机森林回归模型，设置决策树的棵数为 300 颗，使用训练集进行训练。对训练集进行标准化处理，使用训练集进行预测，并计算训练集的决定系数 R^2 得分、对数均方根误差 (RMSE) 和实际均方根误差 (RMSE)。然后，对测试集进行预测，并计算测试集的 R^2 得分、log-RMSE 和 Actual-RMSE。最后，将预测结果和真实结果输出并绘制散点图如下。

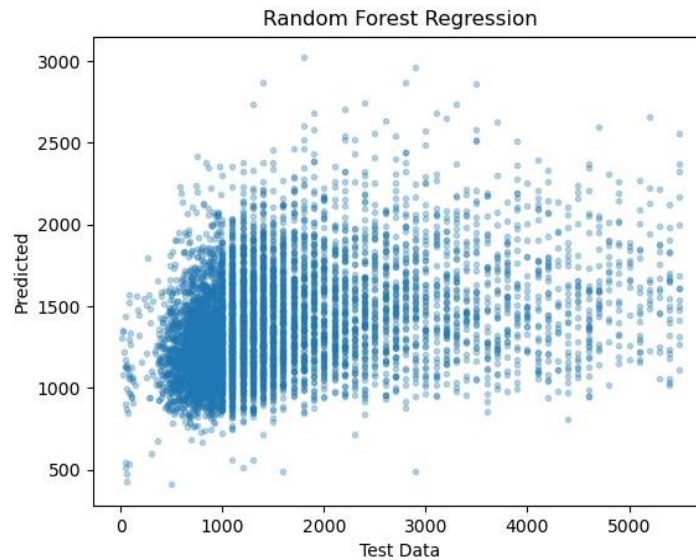


图 7：随机森林回归预测散点图

对于输出结果的进一步可视化：因为输出的预测样本结果很多，所以用 Matlab 中 `datasample` 函数从输出的结果中无放回地随机抽取 150 个样本进行可视化，得到的图形如下：

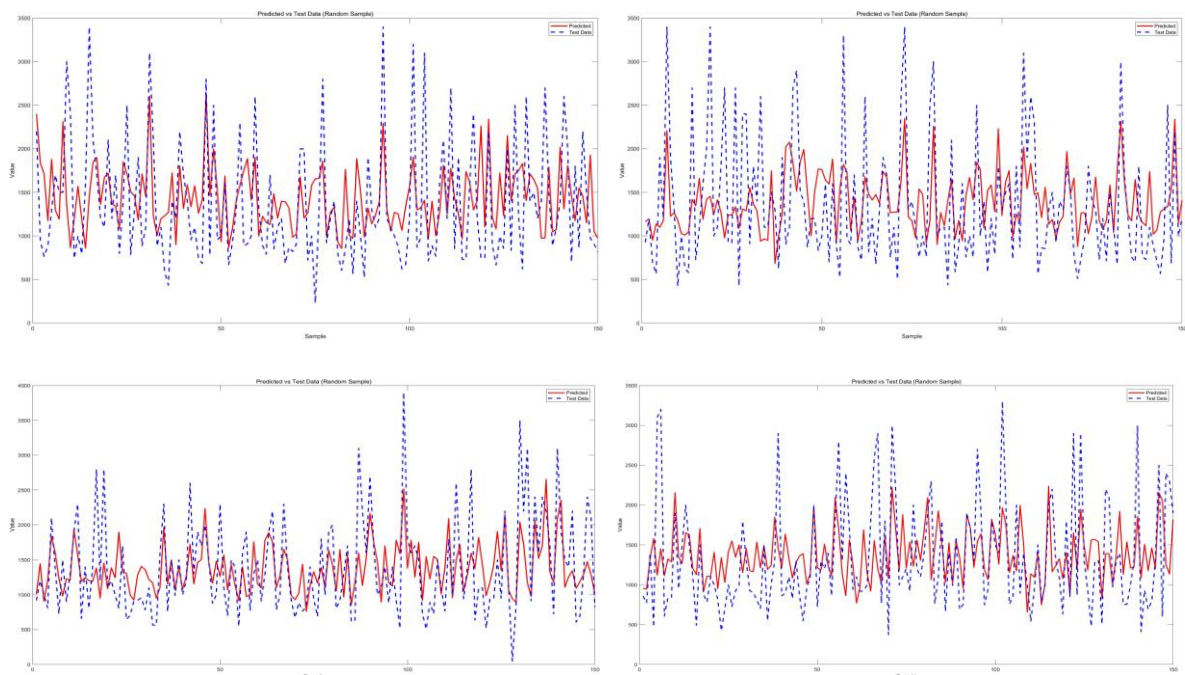


图 8：随机抽样的预测样本和测试集对比图

从以上的随机抽样的预测样本和测试集可视化图可以看出，预测值和测试集值拟合得较好。所以可以直观地看出随机森林回归模型表现效果较好。

5.3 问题三解答

本问要求得出什么样的新闻更容易受到大众的传播这一结论，与第二问中准确预

测传播量的定量研究不同，本问属于定性研究，在问题一中已经对数据进行了初步的处理，除了预测属性和目标属性传播量之间的相关性，还需对预测属性之间的相关性进行分析。



图 9：变量之间的相关性可视化热力图

根据热力图，我们发现文章标题用词数和文章内容用词数之间存在着很强的相关性。同时，文章链接数量和停用词之间、图片数量与正面情感词数之间也存在相关性。但由于数据集较大，如果要探究属性之间的具体相关程度仍需要进行简洁而有价值的分析。

首先观察到数据集中有很多语义特征相似的预测变量，例如文章发布的在一个星期中的日期、文章隶属的频道等，考虑借鉴**特征聚合**的思想把这些语义强相似的预测变量结合起来一起分析，有助于得出什么样的新闻更容易受到大众的传播的结论。

如果直接使用传播量的原始数据作为研究对象，由第一问传播量的原始数据的统计特征：中位数为 1400，均值为 3395，最大值为 843300，可以明显地看出中位数两边的数据值偏离严重，以研究新闻是否周末发布和传播量的关系为例，用原始数据作出图 10，可以看出极端值严重影响了数据分析结果的可视化，而若基于用于第二问中对传播量进行经过四分位距筛选后的数据集研究则会丢失一部分极其流行的数据，故有必要对传播量的数据进行变换。

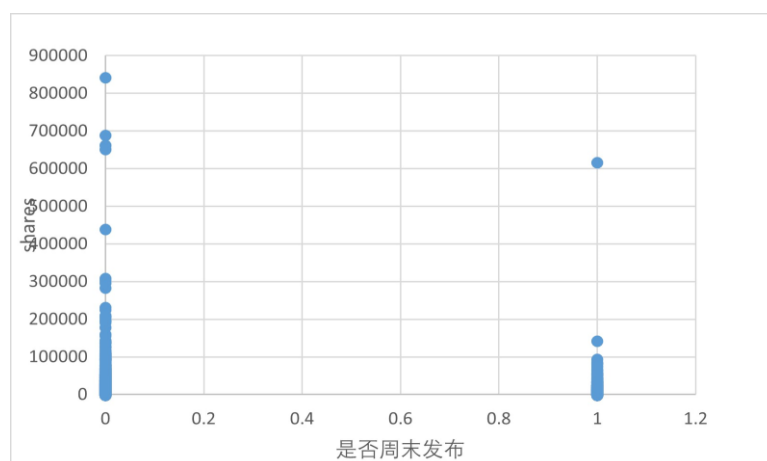


图 10: 新闻是否周末发布与传播量原数据的关系图

考虑到在本问题中对已有传播量的数据只要进行分类：即是否达到了容易受到大众的传播的标准，引入 **One-Hot 编码**，0 代表不流行，1 代表流行。这样可以将原始数据中的传播量转换为一个二元特征，有助于在统计分析和机器学习任务中更好地处理数据。

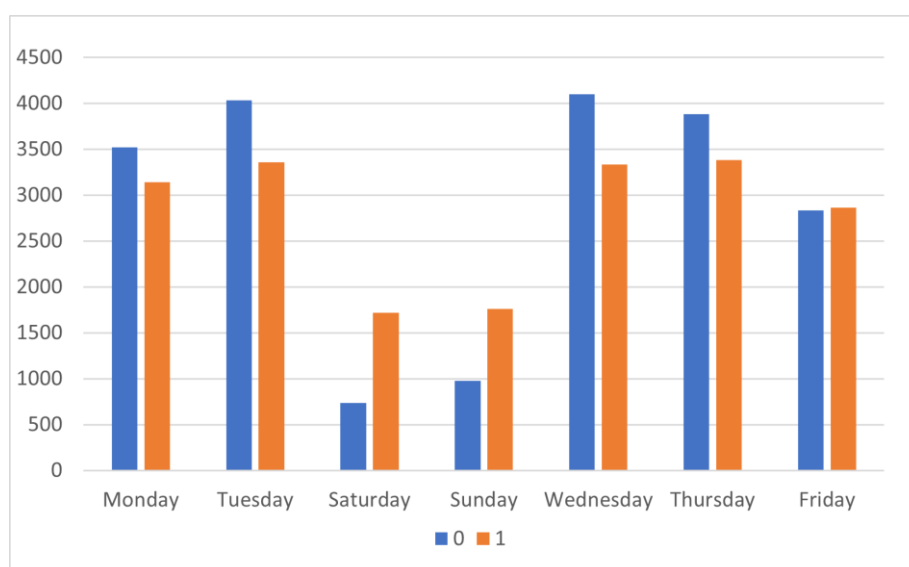


图 11: 在不同日期流行文章和不流行文章统计图

结合图 11 可以看出周末发布新闻数要少，但是在周末发布的文章中得到流行的文章的比例较高。接下来分析容易受到大众传播的即编码为 1 的文章隶属的频道。考虑到文章隶属的频道属于分类特征，用饼状图表示较为清晰。

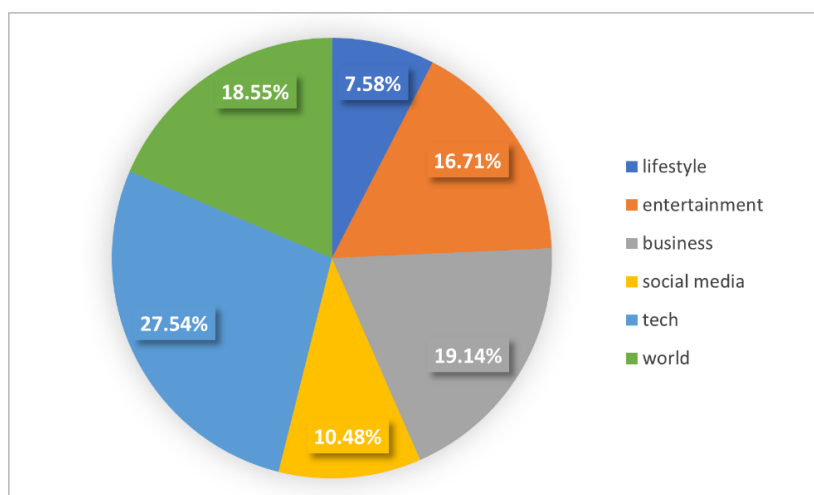


图 12: 容易受到传播的文章隶属频道饼状图

由图 12 可以看出：世界类、科技类、经济类、娱乐类的新闻更容易受到大众的传播。

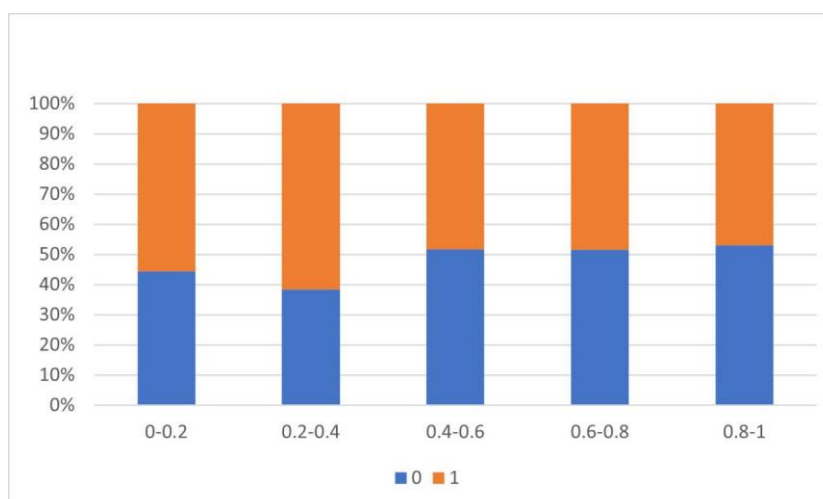


图 13: 文章唯一单词比例和文章是否流行的百分比堆叠柱状图

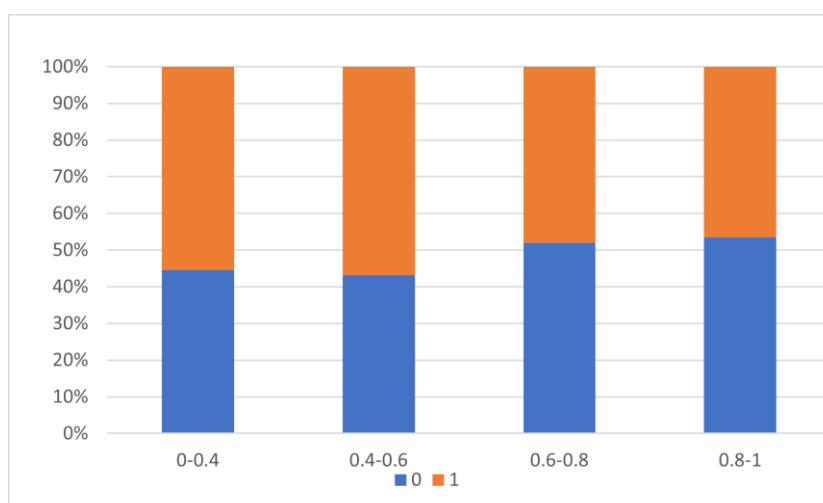


图 14: 文章中非停用唯一单词比例百分比堆叠柱状图

由图 13 和图 14 可以看出文章唯一单词比例在 0.4 以下, 文章中非停用唯一单词比例在 0.6 以下的文章更倾向于受到人们的传播。

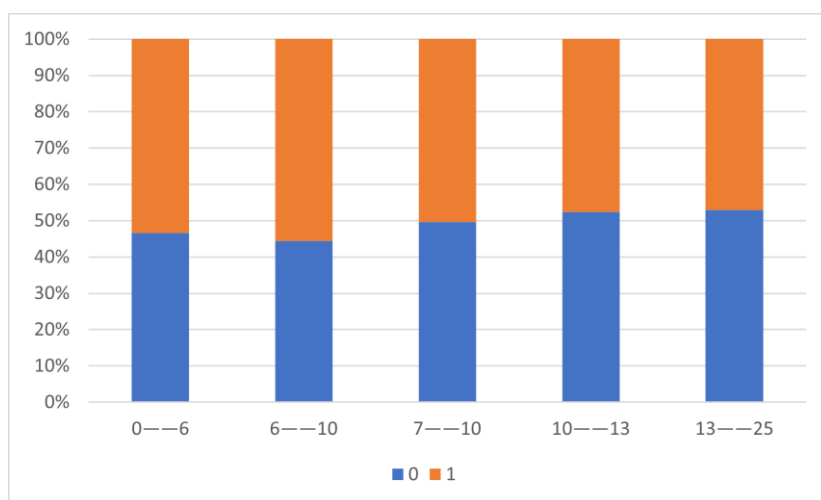


图 15: 文章标题单词数和是否流行的百分比堆叠柱状图

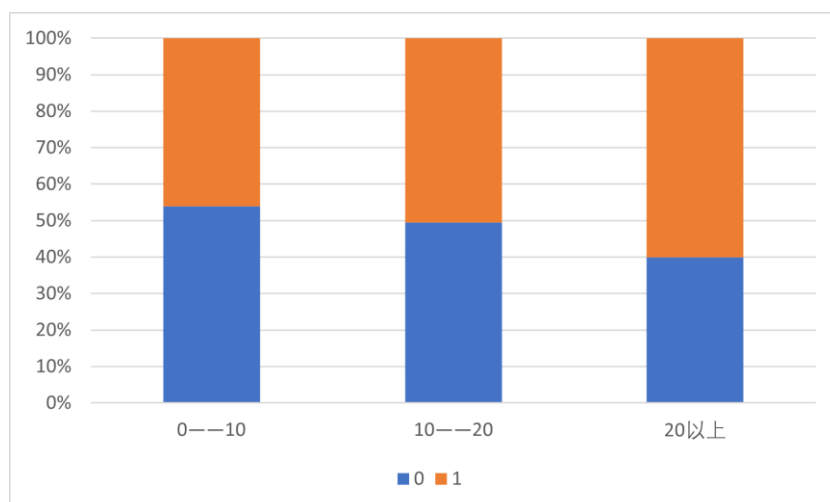


图 15: 文章超链接数和是否流行的百分比堆叠柱状图

从图 14 和图 15 可以看出: 标题中单词数量在 10 个以内、文章包含的超链接数量在 20 个以上的新闻更容易受到大众的传播。

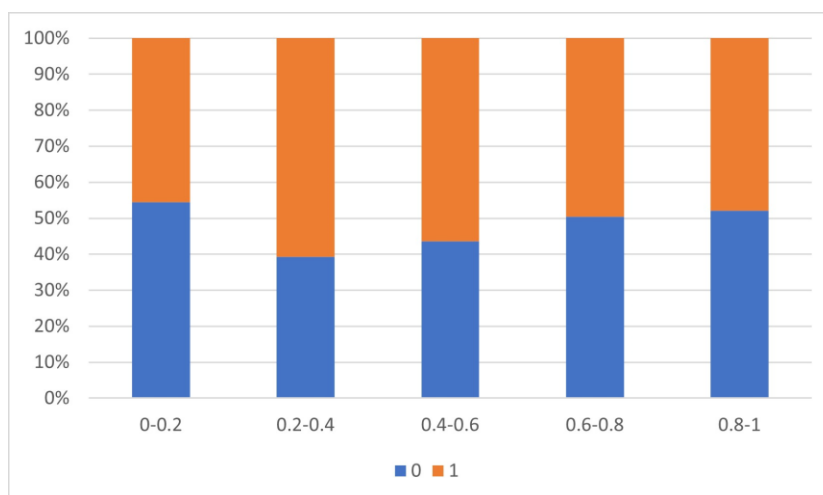


图 6：文章积极单词比例和是否流行的百分比堆叠柱状图

可以看出，文章中积极单词比例介于 0.2 和 0.6 之间的新闻更容易流行。

综合以上的分析可以得到对新闻行业的启示，更容易流行的文章通常含有以下特征：发布日期为工作日，主题与世界、科技、经济、娱乐相关，文章唯一单词比例在 0.4 以下，文章中非停用唯一单词比例在 0.6 以下，标题中单词数量在 10 个以内，文章包含的超链接数量在 20 个以上，文章中积极单词比例介于 0.2 和 0.6 之间。

5.4 问题四的解答

给某某网络媒体公司关于新闻传播的建议信

新闻传播的流行度预测在当今社交媒体时代具有重要意义，我们利用贵公司提供的大量新闻传播数据进行分析，同时采用了随机森林算法进行建模，并对模型进行了验证和评估，从而精确地预测出新闻传播的流行度，并以此为依据得出流行新闻的普遍特点。我代表我们公司在新闻传播的流行度预测项目中得到的综合成果方面，向贵公司提出一些建议。

新闻发布日期上：应尽量选择工作日。数据研究表明，周末发布的新闻数较工作日要少，传播量也不及工作日；且根据常理，人们在工作日更加活跃，故而事件发生率较高、投入新闻的精力也较多。但是新闻讲究时效性，故意拖到工作日发布是为本末倒置，娱乐类新闻便是最好的例子。

新闻发布种类上：在传播比重较低的社交媒体类与生活方式类上应“对症下药”。社交媒体类传播量较少有些出乎意料，究其原因，是社交媒体与网络新闻的赛道不同。所以在社交平台大行其道的当今，我们建议贵公司可以在社交媒体赛道的拓展方面下功夫。而生活方式类上，不可否认它的确受关注较少，但中老年群体却一向青睐于此，所以打造专属于中老年群体的生活类新闻或许是新趋势。另一方面，贵公司作为网络媒体巨头，不应只看重眼前的大众关注，而应肩负起社会责任，主动向大众宣扬健康的生活方式、科普当前的世界格局。毕竟某种程度上，观众“吃”什么，取决于媒体“喂”什么。

新闻发布文风上：贵公司应在文章内容上把好关，既不能过分简洁，也不要长篇大论。务必做到简洁、客观、准确、清晰和全面。

新闻向来是内容为王，故以上三点均以内容为切入口。但广告投放和战略决策同样也不能忽视。

广告投放上可适当迎合大众的喜好，聚焦社交媒体类、科技类、经济类、娱乐类

来吸引大众，形成客户黏性。但总体上的战略决策必须始终坚持“守正创新”。“守正”，即坚持正向价值观导向，切忌因短期利益忽视长期利益与持续性发展，切忌为了制造噱头而放弃原则。“创新”，即应勇于、敢于、善于在新闻内容、新闻形式、新闻载体上推陈出新，争做时代的弄潮儿。

最后，希望这些建议能为贵公司带来一些启发并对未来的发展有所助益。非常感激贵公司对我们公司的信任与支持，本次合作非常愉快，我们也非常期待下次与贵公司的合作！

六、模型的评价、改进与推广

6.1 模型的优点

数据处理方面：本模型对数据的预处理针对数据过于庞大、部分数据失真、存在个别极端值的情况进行筛选，从而防止过载、预测准确性降低、整体的统计规律被个别忽视等情况的发生。

模型建构方面：本模型采用的随机森林是一种强大的机器学习算法，具有高准确性、可处理大量特征、可估计特征重要性、鲁棒性和可并行化处理等优点。

结果分析方面：本模型根据变量之间的相关性可视化热力图以及语义上的相似性对自变量的影响进行分类分析，针对性与逻辑性兼具。

6.2 模型的缺点

数据处理方面：仅筛去相关性系数较低的属性，没有处理到彼此相关性较强的属性对预测造成的影响。

模型建构方面：随机森林训练集的准确度高但是测试集的决定系数 R^2 较低，可能是数据过拟合导致的。

6.3 模型的改进

数据处理方面：可以通过进一步特征选择，主成分分析，引入正则化，对数据进行标准化、归一化、离散化，以及预测间隔等方法改进。

模型建构方面：可以通过调整模型参数、处理样本不平衡、限制树的数量和深度、合理选择特征、增加叶子节点的最小样本数等方法改进。

6.4 模型的推广

本模型还可运用于现实生活中更多样、更复杂、可量化属性对新闻流行度的预测；不受限于新闻流行性，可运用于社会生活中各类高维、大型、不平衡、有噪声、非线性数据集预测问题。

七、参考文献

【1】邱锡鹏，神经网络与深度学习，机械工业出版社，157-192 页，2021 年

【2】周志华，机器学习，清华大学出版社，178-181 页，2016 年

附录 1

第一问代码

#Python 数据统计特征分析 运行环境: Anaconda3

```
import pandas as pd
df=pd.read_csv("C:\\Users\\86183\\Desktop\\数模校赛\\C 题附件.csv")
description = df.describe()
print(df.describe())
# 指定导出的文件路径和文件名
output_file = 'C:\\Users\\86183\\Desktop\\描述统计信息.xlsx'
# 将描述统计信息导出为 Excel 表格
description.to_excel(output_file, index=False)
```

%Matlab 计算相关系数阵，可视化

```
% 读取 CSV 文件数据
data = readmatrix("C:\\Users\\86183\\Desktop\\数模校赛\\C 题附件.csv");

% 提取变量属性和"shares"的数据列
variables = data(:, 3:end-1);
shares = data(:, end);

% 计算相关性
correlation = corrcoef([variables shares]);

% 提取变量属性与"shares"的相关性
variable_correlation = correlation(1:end-1, end);

% 打印相关系数
for i = 1:numel(variable_correlation)
    fprintf('变量属性 %d 与"shares"的相关系数: %.2f\n', i, variable_correlation(i));
end

% 绘制条形图
figure;
bar(variable_correlation, 'FaceColor', '#0072BD', 'BarWidth', 0.5);
title('变量属性与"shares"的相关性', 'FontSize', 14);
xlabel('变量属性', 'FontSize', 12);
ylabel('相关系数', 'FontSize', 12);
grid on;
set(gca, 'FontSize', 12);
% 添加网格线
yline(-1:0.3:1, '--', 'Color', 'gray', 'LineWidth', 0.5);
hold off;
```

附录 2

第二问代码

#Python 随机森林回归代码 运行环境: Anaconda3

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestRegressor

# 数据预处理
def preprocess():
    global df, X, Y, X_train, X_test, Y_train, Y_test, baseline
    # 读取 CSV 文件
    df = pd.read_csv("C:\\Users\\86183\\Desktop\\数模校赛\\C 题附件.csv")
    # 移除列名两端的空格
    df = df.rename(columns = lambda x: x.strip())
    # 删去'url', 'timedelta' 列
    df = df.drop(columns = ['url', 'timedelta'])
    # 过滤'shares'在(0, 5581)范围内的数据
    df = df[df['shares'] > 0]
    df = df[df['shares'] < 5581]
    # 删除不需要的列
    df = df.drop(columns = ['n_non_stop_words',
                            'n_non_stop_unique_tokens', 'rate_positive_words',
                            'rate_negative_words',
                            'global_sentiment_polarity', 'global_rate_positive_words',
                            'weekday_is_monday', 'weekday_is_tuesday',
                            'global_rate_negative_words',
                            'weekday_is_wednesday', 'weekday_is_thursday',
                            'weekday_is_friday',
                            'LDA_00', 'LDA_01', 'LDA_02', 'LDA_04',
                            'n_tokens_title', 'n_tokens_content',
                            'n_unique_tokens', 'num_self_hrefs',
                            'data_channel_is_entertainment', 'data_channel_is_bus',
                            'data_channel_is_socmed', 'data_channel_is_tech',
                            'data_channel_is_world',
                            'abs_title_subjectivity', 'title_subjectivity', ])
    # 对目标变量'shares'取对数
    df['shares'] = np.log(df['shares'])

    X = df[df.columns]
```

```

Y = X['shares'].values
X = X.drop('shares', axis = 1).values

# 将数据拆分为测试集和训练集
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.20, shuffle =
False)
baseline = np.exp(Y_train)
baseline = baseline.mean()

# 计算RMSE的函数
def actualRMSE(diff):
    diff_squared = diff * diff
    diff_squared_sum = sum(diff_squared)
    diff_squared_sum_by_n = (diff_squared_sum * 1.0)/ len(diff)
    rmse = np.sqrt(diff_squared_sum_by_n)
    return rmse

# 散点图函数
def scatterplot(model, data):
    x = data['Test Data']
    y = data['Predicted']
    plt.scatter(x, y, marker = '.', alpha = 0.3)
    plt.title(model)
    plt.xlabel('Test Data')
    plt.ylabel('Predicted')
    plt.savefig(model + '.png', dpi=100)
    plt.show()

# 随机森林函数
def rfr():
    print ("\nRandom Forest Regression Model")
    # 数据标准化
    scaler = StandardScaler().fit(X_train)
    rescaled_X_train = scaler.transform(X_train)
    # 创建随机森林回归模型
    model = RandomForestRegressor(n_estimators = 300)
    model.fit(rescaled_X_train, Y_train)
    # 训练集上的预测和评估
    train_predictions = model.predict(rescaled_X_train)
    train_r2score = model.score(rescaled_X_train, Y_train)
    actual_y_train = np.exp(Y_train)
    actual_train_predictions = np.exp(train_predictions)
    train_diff = actual_y_train - actual_train_predictions

```

```

print ('Train R2 Score = ', round(train_r2score, 2))
print ('Train Log RMSE = ', round(mean_squared_error(Y_train, train_predictions),
2))
print ('Train Actual RMSE = ', round(actualRMSE(train_diff), 2))
# 测试集上的预测和评估
rescaled_X_test = scaler.transform(X_test)
predictions = model.predict(rescaled_X_test)
r2score = model.score(rescaled_X_test, Y_test)

actual_y_test = np.exp(Y_test)
actual_predicted = np.exp(predictions)
diff = actual_y_test - actual_predicted
diff_baseline = actual_y_test - baseline

print ('R2 Score = ', round(r2score, 2))
print ('Log RMSE = ', round(mean_squared_error(Y_test, predictions), 2))
print ('Actual RMSE = ', round(actualRMSE(diff), 2))
print ('Baseline RMSE = ', round(actualRMSE(diff_baseline), 2))
# 创建一个包含实际值、预测值和差异的 DataFrame
compare_actual = pd.DataFrame({'Test Data': actual_y_test,
                              'Predicted': actual_predicted,
                              'Difference': diff})

compare_actual = compare_actual.astype(int)
# 将数据导出到 CSV 文件
compare_actual.to_csv('C:\\Users\\86183\\Desktop\\compare_actual.csv',
index=False)
# 绘制散点图
scatterplot('Random Forest Regression', compare_actual)

return compare_actual

# 清空屏幕并运行
print(chr(27) + "[2J")
preprocess()
rfr()

%Matlab 进一步可视化随机森林回归结果

% 读取 CSV 文件
data = readmatrix('C:\\Users\\86183\\Desktop\\数模校赛\\compare_actual.csv');

% 提取数据列
difference = data(:, 3);

```



```
predicted = data(:, 2);
testData = data(:, 1);

% 随机抽样 100 组数据
numSamples = 150;
indices = datasample(1:length(predicted), numSamples, 'Replace', true);
predicted_sample = predicted(indices);
testData_sample = testData(indices);

% 绘制数据
plot(predicted_sample, 'r-', 'LineWidth', 2); % 绘制预测值
hold on;
plot(testData_sample, 'b--', 'LineWidth', 2); % 绘制测试数据
hold off;

% 设置图形标题和轴标签
title('Predicted vs Test Data (Random Sample)');
xlabel('Sample');
ylabel('Value');

% 添加图例
legend('Predicted', 'Test Data');
```

附录 3

第三问代码

#Python 绘制热力图代码 运行环境: Anaconda3

```
from numpy import array, sqrt, argsort, std
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib
import seaborn as sns
from matplotlib.pyplot import figure
from sklearn import preprocessing
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression, LogisticRegression
from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import mean_squared_error as mse
from sklearn.metrics import *
from sklearn.tree import *
from sklearn.ensemble import *
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis,
QuadraticDiscriminantAnalysis
from sklearn.model_selection import RandomizedSearchCV, GridSearchCV
from sklearn.naive_bayes import GaussianNB, MultinomialNB
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
dataset = pd.read_csv("C:\\Users\\86183\\Desktop\\数模校赛\\C 题附件.csv")

dataset = dataset.drop_duplicates()
dataset.shape
dataset.columns = dataset.columns.str.lstrip()
dataset.columns
dataset = dataset.drop(['url', 'timedelta'], axis=1)
pd.set_option("display.max_rows", None, "display.max_columns", None)
dataset.head()

dataset = dataset.dropna()
dataset.shape










dataset.describe()

figure(num=None, figsize=(12, 12), dpi=80, facecolor='w', edgecolor='k')
corr = dataset.corr()
ax = sns.heatmap(
```

```
corr,  
    vmin=-1, vmax=1, center=0,  
    cmap=sns.diverging_palette(20, 220, n=200),  
    square=True  
)  
ax.set_xticklabels(  
    ax.get_xticklabels(),  
    rotation=45,  
    horizontalalignment='right'  
);  
plt.show()
```

附录 4

支撑材料

| | | | |
|---|------------------|----------------------|-----------|
|  compare_actual | 2023/12/10 21:39 | Microsoft Excel 逗... | 104 KB |
|  corroration_matrix.py | 2023/12/11 19:15 | Python.File | 2 KB |
|  correlation_matrix | 2023/12/11 19:06 | MATLAB Code | 1 KB |
|  C题附件 | 2023/12/7 18:59 | Microsoft Excel 逗... | 16,518 KB |
|  describe.py | 2023/12/11 19:10 | Python.File | 1 KB |
|  random_forest_regression.py | 2023/12/11 19:00 | Python.File | 5 KB |
|  rfr_result | 2023/12/11 18:58 | MATLAB Code | 1 KB |
|  描述统计信息 | 2023/12/9 15:05 | Microsoft Excel 逗... | 4 KB |
|  去除相关性再0.01以下的列 | 2023/12/11 19:04 | Microsoft Excel 工... | 6,184 KB |