网络新闻的流行度预测

专　　业：　信管创新班 20-1 班

姓名：史康威 徐梦理 高纪铭 贾其豪 金硕

老　　师：　王刚 王钊 任刚

时　　间：　2023 年 4 月

摘要

本文针对社交媒体上的文章转发量进行了预测，数据包括 38000 条新闻，其中包括 58 个预测变量和 1 个因变量。本次实验采样 35000 条数据作为训练和验证集，3000 条数据作为测试集。

首先通过对数据进行观察，发现数据存在偏态状况，通过使用采取小样本的方式选取了转发数小于 10000 的样本作为模型的训练样本。为加快模型的训练速度采用了卡方检验和 PCA 的方法进行特征选择。在模型调优时，利用参数调优的方式对选取的特征和降维的维度进行调优。

在回归模型方面，利用多种算法进行回归预测，在选定的样本集上采用了随机森林回归、XGBoost 回归、支持向量回归等多种回归算法进行建模。为了获得最佳预测效果，对模型参数进行了调优，并使用交叉验证和 RMSE 评估指标来验证模型的性能。通过结果发现，回归预测的结果并不好，利用可视化发现，结果的分布呈现出明显的分类。基于此本文使用了逻辑回归的方式对该任务进行预测，最终获得较好的结果。最好的 RMSE 得分为 0.5132。

一、题目背景

网络新闻作为一种新兴的信息传播渠道，已经成为人们获取信息的重要途径之一。许多研究者对网络新闻的流行度进行了研究，认为流行度反映了网络新闻在社交媒体和互联网上的影响力和受欢迎程度。因此，预测网络新闻的流行度（例如，转发数）是非常有意义的。本文旨在使用随机森林回归、XGBoost 回归、支持向量回归、逻辑回归四种回归算法，通过网络新闻的属性预测网络新闻的流行度。

二、研究问题分析

1. 数据来源与说明

本次实验的数据包括 38000 条新闻，其中包括 58 个预测变量和 1 个因变量（转发数）。本次实验采样 35000 条数据作为训练和验证集，3000 条数据作为测试集。数据集特征在附录中可见。数据集来源：[Online News Popularity Data Set]

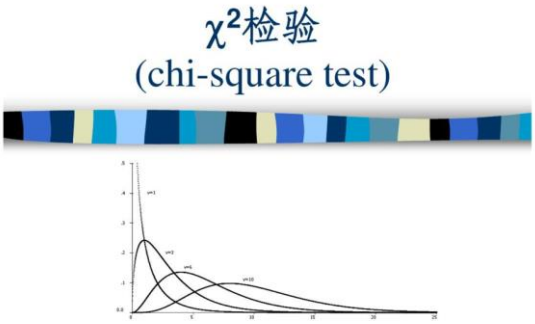
2. 研究问题

本次实验旨在利用网络新闻的属性来预测网络新闻的流行度，即转发数。在研究过程中，需要考虑不同的预测变量对流行度的影响，通过选择适合的算法模型进行回归分析，并对模型进行评估验证。

三、原理介绍

1. 特征选择

在该数据集中有 58 个特征，样本数据量大，为简化计算，首先对样本数据进行特征选择，在这里介绍卡方检验的原理。



卡方检验是一种常见的统计方法，主要用于比较变量之间的关联性，常用于特征选择中。卡方检验的原理是计算观测值和期望值之间的差异，从而判断两个变量是否独立。在特征选择中，通过计算每个特征与类别之间的卡方值，来评估特征的重要性，选择与类别相关性较高的特征。计算期望频数表方法和卡方值如下所示

$$E_{ij} = \frac{R_i \times C_j}{N}$$

$$\chi^2 = \sum_{i=1}^{n} \sum_{j=1}^{k} \frac{(O_{ij} - E_{ij})^2}{E_{ij}}$$

其中，

- χ^2表示卡方值；
- O_{ij}表示观测频数表中第i行第j列的观测值；

- E_{ij}表示期望频数表中第i行第j列的期望值；
- n表示行数；
- k表示列数。

2. 随机森林回归原理

随机森林是一个集成学习方法，由多个决策树组成，每个决策树都是基于随机抽样的样本特征（属性）集训练出来的。因此，随机森林在不同的样本分布和特征子集下，可以产生多个不同但相关度较低的决策树，最终通过对多个决策树的投票或平均值来进行预测，从而提高整体模型的泛化能力和鲁棒性。

3. MAPE

MAPE（Mean Absolute Percentage Error，平均绝对百分比误差）是一种用于回归模型评估的常见指标，用于评估模型的平均误差大小，可以用百分比来表示误差大小，是一种相对误差的度量。

MAPE 的公式为：

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n} \frac{|y_i - \hat{y_i}|}{|y_i|}$$

其中，

- n 是样本数量；
- y_i 是第i个样本的真实值；
- $\hat{y_i}$ 是第i个样本的预测值。

MAPE 的计算方法是，将每个真实值与其预测值之间的绝对误差除以其真实值，并将它们的平均值乘以 100%。由于 MAPE 使用百分比来度量误差，因此可以更好地表达误差的严重程度。

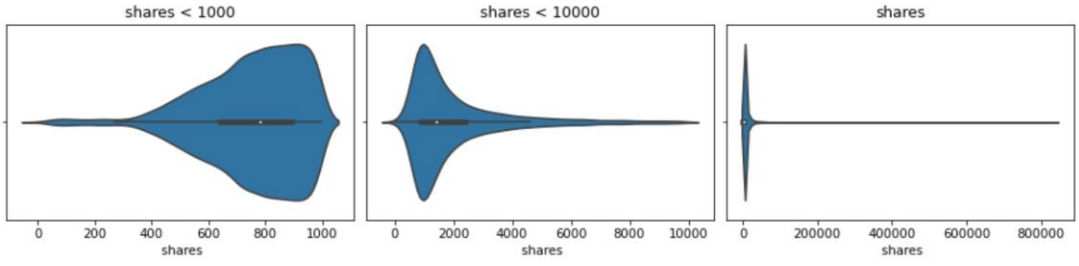
MAPE 的优点是可以避免由于数据范围变化而产生的量纲问题，并且可以直观地反映误差的相对大小，是一种通用的评价指标。但是，MAPE 存在的问题是，当真实值为 0 时，分母为 0，此时 MAPE 无法计算。此外，MAPE 对于异常值敏感，如果存在异常值，MAPE 的值会受到影响。

总之，MAPE 是一种常见的用于回归模型评估的指标，适用于数据范围变化较大的情况，但需要注意其在某些情况下的局限性。

四、数据分析方案

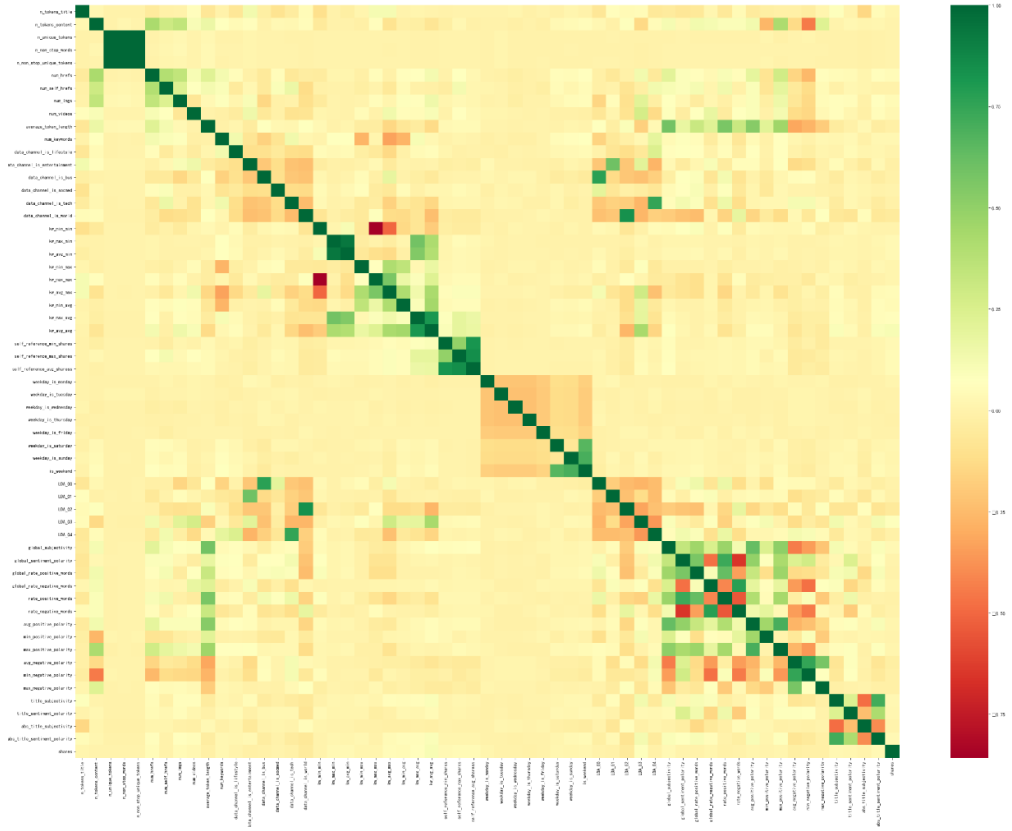
1. 数据预处理

通过对特征分布的观察发现，目标 shares 存在偏态，为了降低偏态降低的影响，在模型训练时采用了部分数据集即利用数据分布密集的部分。在本文中使用的是转发次数小于 10000 次的样本集。



对样本进行处理后，采用特征选择选取具体的特征作为训练的特征。本文采用卡方检验

和 PCA 降维的方法对数据的特征进行选择和数据降维。



2. 模型选择

为了得到更好的回归结果，本文采用了多个回归算法进行回归预测。同时，为了预测结果更加泛化，我们采用了逻辑回归算法进行观测，描述整体样本的分布，并作为预测的结果。

我们选择四种回归算法来构建模型：随机森林回归、XGBoost 回归、支持向量回归、逻辑回归。这些算法被广泛使用于回归问题，其中随机森林回归和 XGBoost 回归原理基于决策树，逻辑回归 主要用于二元分类，支持向量回归是基于支持向量机（SVM） 的一个重要应用场景，适用于非线性问题。

3. 参数选择

在模型选择后，需要对算法进行参数调优，以获得最佳的预测效果。在本次实验中，我们使用交叉验证方法对模型进行优化。具体来说，我们使用 GridSearchCV 对特征选择、降维维度和四种算法的一些主要参数进行交叉验证，以确定最佳参数的组合。为了避免过度拟合，我们还使用 RMSE 指标对模型进行评估验证。

五、结果分析和结论

采用不同特征选择算法和回归算法的 mape 结果如下图所示：

算法	样本规模	特征选择方法	MAPE
XGBoost	0.1	卡方	1.5958

RF	0.1	卡方	1.5987
SVR	0.1	PCA	0.6382
LR	1	卡方（15）+PCA(5)	0.5132

通过结果发现，使用逻辑回归进行回归的 mape 值最小，这是因为数据的范围大，对数据进行预测时导致预测精度差，从而出现了分类的效果。

数据样本本身的波动性较大，数据集整体呈现偏态，利用常见的机器学习回归算法难以很好的拟合。从分类的角度将数据样本划分为多类，将每个类别的固定值作为划分区间的预测值，能够有效减小偏态的影响。

六、总结

通过本次数据分析时间，我们学习了如何对社交媒体上的文章转发量进行预测，并且在实践中掌握了各种回归算法和数据处理技巧。此次比赛考验了我们对模型的理解能力和实践能力，也让我们深刻认识到团队协作的重要性。在数据处理方面，我们发现数据存在偏态分布现象，为了减小偏态的影响，我们采取了选取小样本的方法。在特征选择方面，我们尝试了卡方检验和 PCA，最终确定了特征的数量和重要性，提高了模型的训练速度和精度。在模型选择方面，我们尝试了多种回归算法，最终选定了随机森林回归和 XGBoost 回归。在模型调优方面，我们利用交叉验证和 RMSE 评估指标来评价模型的性能，并进行了参数调整，最终得到了较为准确的预测结果。通过这个竞赛，我们加深了对模型建立和数据处理技巧的理解，也体验到了团队合作的重要性和价值。在竞赛的过程中，我们互相学习、互相帮助，在团队的配合下完成任务，增强了我们的合作精神和凝聚力。同时，也让我们对未来的学习和工作有了更深入的认识和规划。感谢本次竞赛，让我们获得了宝贵的经验和成长的机会。

七、附录

1. 数据集特征

特征名称及含义

n_tokens_title: Number of words in the title
n_tokens_content: Number of words in the content
n_unique_tokens: Rate of unique words in the content
n_non_stop_words: Rate of non-stop words in the content
n_non_stop_unique_tokens: Rate of unique non-stop words in the content
num_hrefs: Number of links
num_self_hrefs: Number of links to other articles published by Mashable
num_imgs: Number of images
num_videos: Number of videos

average_token_length: Average length of the words in the content
num_keywords: Number of keywords in the metadata
data_channel_is_lifestyle: Is data channel 'Lifestyle'?
data_channel_is_entertainment: Is data channel 'Entertainment'?
data_channel_is_bus: Is data channel 'Business'?
data_channel_is_socmed: Is data channel 'Social Media'?
data_channel_is_tech: Is data channel 'Tech'?
data_channel_is_world: Is data channel 'World'?
kw_min_min: Worst keyword (min. shares)
kw_max_min: Worst keyword (max. shares)
kw_avg_min: Worst keyword (avg. shares)
kw_min_max: Best keyword (min. shares)
kw_max_max: Best keyword (max. shares)
kw_avg_max: Best keyword (avg. shares)
kw_min_avg: Avg. keyword (min. shares)
kw_max_avg: Avg. keyword (max. shares)
kw_avg_avg: Avg. keyword (avg. shares)
self_reference_min_shares: Min. shares of referenced articles in Mashable
self_reference_max_shares: Max. shares of referenced articles in Mashable
self_reference_avg_sharess: Avg. shares of referenced articles in Mashable
weekday_is_monday: Was the article published on a Monday?
weekday_is_tuesday: Was the article published on a Tuesday?
weekday_is_wednesday: Was the article published on a Wednesday?
weekday_is_thursday: Was the article published on a Thursday?
weekday_is_friday: Was the article published on a Friday?
weekday_is_saturday: Was the article published on a Saturday?
weekday_is_sunday: Was the article published on a Sunday?
is_weekend: Was the article published on the weekend?
LDA_00: Closeness to LDA topic 0
LDA_01: Closeness to LDA topic 1
LDA_02: Closeness to LDA topic 2
LDA_03: Closeness to LDA topic 3
LDA_04: Closeness to LDA topic 4
global_subjectivity: Text subjectivity
global_sentiment_polarity: Text sentiment polarity
global_rate_positive_words: Rate of positive words in the content
global_rate_negative_words: Rate of negative words in the content
rate_positive_words: Rate of positive words among non-neutral tokens
rate_negative_words: Rate of negative words among non-neutral tokens
avg_positive_polarity: Avg. polarity of positive words
min_positive_polarity: Min. polarity of positive words
max_positive_polarity: Max. polarity of positive words
avg_negative_polarity: Avg. polarity of negative words
min_negative_polarity: Min. polarity of negative words

max_negative_polarity: Max. polarity of negative words
title_subjectivity: Title subjectivity
title_sentiment_polarity: Title polarity
abs_title_subjectivity: Absolute subjectivity level
abs_title_sentiment_polarity: Absolute polarity level
shares: Number of shares (target)

2. 模型调参

模型调参结果展示

算法	样本规模	特征选择方法	十折交叉?	MAPE	预测值处理
XGBoost	0.1	无·	否·	1.8153	
XGBoost	1	无·	否·	1.87	
XGBoost	1	无·	是	2.1169	
XGBoost	0.1	无·	是	1.6154	
XGBoost	0.1	卡方	是	1.5958	
RF	0.1	卡方	是	1.5958	
SVR	1	卡方	是	0.6689	
RF	0.1	PCA	是	1.7914	
RF	0.1	卡方	是	1.5987	
SVR	0.1	卡方	是	0.6351	
LR	0.1	PCA	是	0.542943	没处理
LR	0.1	PCA	是	0.542875	加 abs
LR	0.1	卡方	是	0.55244	没处理
LR	0.1	卡方	是	0.55244	加 abs
LR	0.1	PCA	是	0.542943	<0=0
SVR	0.1	PCA	是	0.6382	
LR	0.1	PCA+卡方	是	0.536016	
LR	0.1	卡方（15）+PCA(5)	是	0.533923	
LR	1	卡方（15）+PCA(10)	手动降维	0.5145	大于 10000 的都删去了
LR	1	卡方（15）+PCA(10)	无手动降维	0.5149	大于 10000 的都删去了
LR	1	卡方（15）+PCA(5)	无手动降维	0.5132	大于 10000 的都删去了
LR	1	卡方（15）+PCA(5)	手动降维	0.515	大于 10000 的都删去了

3. 核心代码展示

逻辑回归及交叉验证

```
from sklearn.feature_selection import SelectKBest, chi2
selector = SelectKBest(chi2, k=10)
```

```
X_new = selector.fit_transform(X_scaled, y)

# 训练模型并进行十折交叉验证
n_splits = 10
kf = KFold(n_splits=n_splits, shuffle=True, random_state=0)
mae_list = []
rmse_list = []
mape_list = []
for train_index, test_index in kf.split(X_new):
    X_train, X_test = X_new[train_index], X_new[test_index]
    y_train, y_test = y.iloc[train_index], y.iloc[test_index]
    lr_model = LogisticRegression()
    lr_model.fit(X_train, y_train)
    y_pred = lr_model.predict(X_test)
    mae_list.append(mean_absolute_error(y_test, y_pred))
    rmse_list.append(np.sqrt(mean_squared_error(y_test, y_pred)))
    mape_list.append(np.mean(np.abs((y_test - y_pred) / y_test)) * 100)

# 显示结果
print('MAE: %.4f' % np.mean(mae_list))
print('RMSE: %.4f' % np.mean(rmse_list))
print('MAPE: %.4f%%' % np.mean(mape_list))
```