Online News Popularity

Pavan Vamsi Tadikonda
Sonal Seth
Amisha Shukla
Niharika Byraply Yathiraju
Dave Shivhare

13 April 13, 2022

—

Business Analytics With R
BUAN 6356

—

Dr. Jianqing Chen

CONTENTS

EXECUTIVE SUMMARY

BUSINESS BACKGROUND AND MOTIVATION

The data we have obtained belongs to Mashable, an online media and entertainment company. Online publishing houses are heavily dependent on the advertising revenue[1]. Pewresearch estimates nearly 70% of the revenue for some organizations. The popularity of each article is of paramount importance.

The problem is that too many of the articles end up not being popular and too many articles are being reviewed manually. The former hurts the ad revenue while the latter adds too much labor cost and time to the business process.

Our contribution is to predict the popularity of an article that will greatly aid the decision making process in an online publishing house. The people in charge of publishing can obtain a fuller picture with both the domain knowledge and our predictions. They can then reduce the number of articles they have to review in order to publish popular articles frequently.

SOURCE OF INFORMATION

The information is obtained from the machine learning repository (link). This is second hand information. Since the UCI repository has an excellent reputation and is held in great esteem by the community, we trust the source and the information Implicitly.

ANALYTICS SOLUTION OVERVIEW

Once an article is written, the required predictor data about the article can then be plugged into the model and its output will be used in the decision making.

Since the source of the data is a digital article, a lot of data about each article can be acquired quite painlessly. Hence, a large number of predictor variables have been considered for the model. However, most data collected is numeric and can sometimes be similar to each other, this leads to multicollinearity.

However, due to the large amount of predictor variables, the interpretibability of the model may take a slight. The final model takes the trade-offs into account and tries to strike a reasonable balance between interpretability and predictive accuracy.

The final aim would be to reduce the number of published articles that do not reach the popularity benchmark (>2500 shares).

DATA MINING OBJECTIVE

The data mining objective is to predict the popoularity of an unpublished article. Since publishing houses have a high standard integrity to meet, an article can not be pulled down and republished with improvisation. It is imperative that the prediction not be a false positive (falsely predict that an article would be popular). A false negative does not have grave consequences, the article can be reviewed manually and published. However, the huge number of articles being published makes that manual approach labour-intensive and time-ineffective. We need to build a model that can reduce the number of articles to be manually reviewed for potential non-popularity

DATA DESCRIPTION

Attribute Information:

0. url: URL of the article (non-predictive)

1. timedelta: Days between the article publication and the dataset acquisition (non-predictive)

2. n_tokens_title: Number of words in the title

3. n_tokens_content: Number of words in the content

4. n_unique_tokens: Rate of unique words in the content

5. n_non_stop_words: Rate of non-stop words in the content

6. n_non_stop_unique_tokens: Rate of unique non-stop words in the content

7. num_hrefs: Number of links

8. num_self_hrefs: Number of links to other articles published by Mashable

9. num_imgs: Number of images

10. num_videos: Number of videos

11. average_token_length: Average length of the words in the content

12. num_keywords: Number of keywords in the metadata

13. data_channel_is_lifestyle: Is data channel 'Lifestyle'?

14. data_channel_is_entertainment: Is data channel 'Entertainment'?

15. data_channel_is_bus: Is data channel 'Business'?

16. data_channel_is_socmed: Is data channel 'Social Media'?

17. data_channel_is_tech: Is data channel 'Tech'?

18. data_channel_is_world: Is data channel 'World'?

19. kw_min_min: Worst keyword (min. shares)

20. kw_max_min: Worst keyword (max. shares)

21. kw_avg_min: Worst keyword (avg. shares)

22. kw_min_max: Best keyword (min. shares)

23. kw_max_max: Best keyword (max. shares)

24. kw_avg_max: Best keyword (avg. shares)

25. kw_min_avg: Avg. keyword (min. shares)

26. kw_max_avg: Avg. keyword (max. shares)

27. kw_avg_avg: Avg. keyword (avg. shares)

28. self_reference_min_shares: Min. shares of referenced articles in Mashable

29. self_reference_max_shares: Max. shares of referenced articles in Mashable

30. self_reference_avg_sharess: Avg. shares of referenced articles in Mashable

31. weekday_is_monday: Was the article published on a Monday?

32. weekday_is_tuesday: Was the article published on a Tuesday?

33. weekday_is_wednesday: Was the article published on a Wednesday?

34. weekday_is_thursday: Was the article published on a Thursday?

35. weekday_is_friday: Was the article published on a Friday?

36. weekday_is_saturday: Was the article published on a Saturday?

37. weekday_is_sunday: Was the article published on a Sunday?

38. is_weekend: Was the article published on the weekend?
39. LDA_00: Closeness to LDA topic 0
40. LDA_01: Closeness to LDA topic 1
41. LDA_02: Closeness to LDA topic 2
42. LDA_03: Closeness to LDA topic 3
43. LDA_04: Closeness to LDA topic 4
44. global_subjectivity: Text subjectivity
45. global_sentiment_polarity: Text sentiment polarity
46. global_rate_positive_words: Rate of positive words in the content
47. global_rate_negative_words: Rate of negative words in the content
48. rate_positive_words: Rate of positive words among non-neutral tokens
49. rate_negative_words: Rate of negative words among non-neutral tokens
50. avg_positive_polarity: Avg. polarity of positive words
51. min_positive_polarity: Min. polarity of positive words
52. max_positive_polarity: Max. polarity of positive words
53. avg_negative_polarity: Avg. polarity of negative words
54. min_negative_polarity: Min. polarity of negative words
55. max_negative_polarity: Max. polarity of negative words
56. title_subjectivity: Title subjectivity
57. title_sentiment_polarity: Title polarity
58. abs_title_subjectivity: Absolute subjectivity level
59. abs_title_sentiment_polarity: Absolute polarity level
60. shares: Number of shares
61. Popularity: Did the article reach 60th percentile in shares or not <------------Target

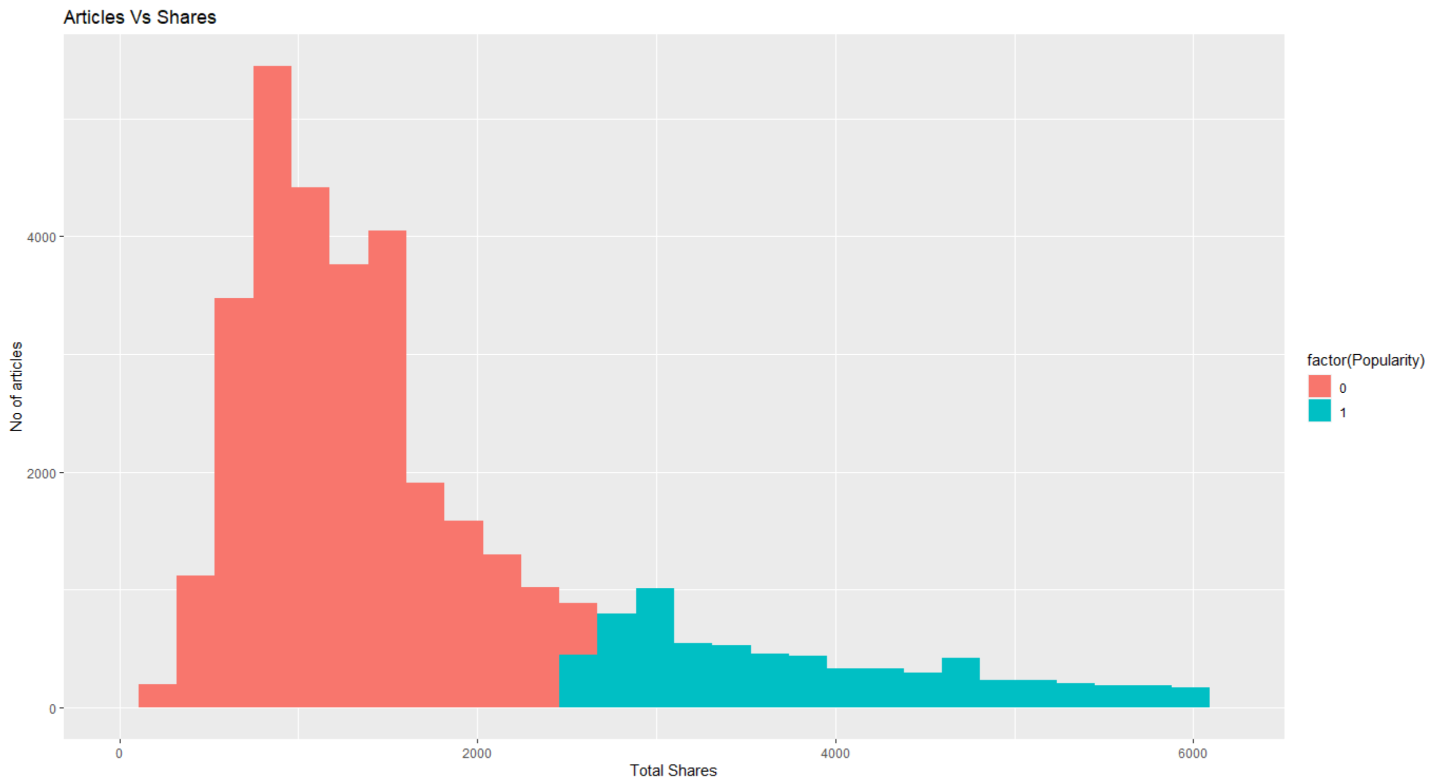
DATA PREVIEW

All the data we initially collected were numerical. There was a lot of multicollinearity, we removed the correlated variables usign the akaike information criterion. We ended up with 27 Variables.

timedelta	n_tokens_title	n_unique_tokens	n_non_stop_words	n_non_stop_unique_tokens
731	12	0.663594467	0.999999992	0.815384609
731	9	0.604743081	0.999999993	0.791946303
731	9	0.575129531	0.999999992	0.663865541
731	9	0.503787878	0.999999997	0.665634673
731	13	0.415645617	0.999999999	0.540889526

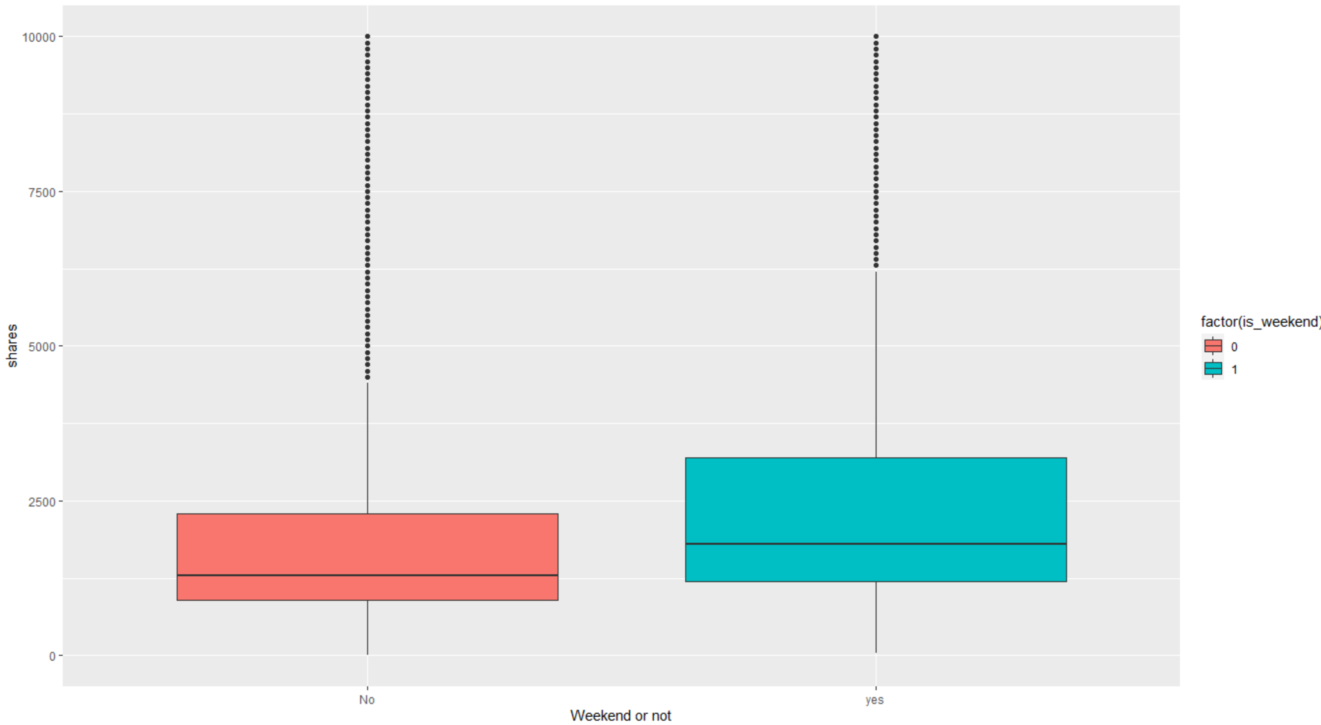
INTERESTING FINDINGS

The original data measured the popularity in the number of shares. But that is unnecessary amount of granularity for the decision makers. We have used a minimum threshold of 2500 shares to be considered as popular



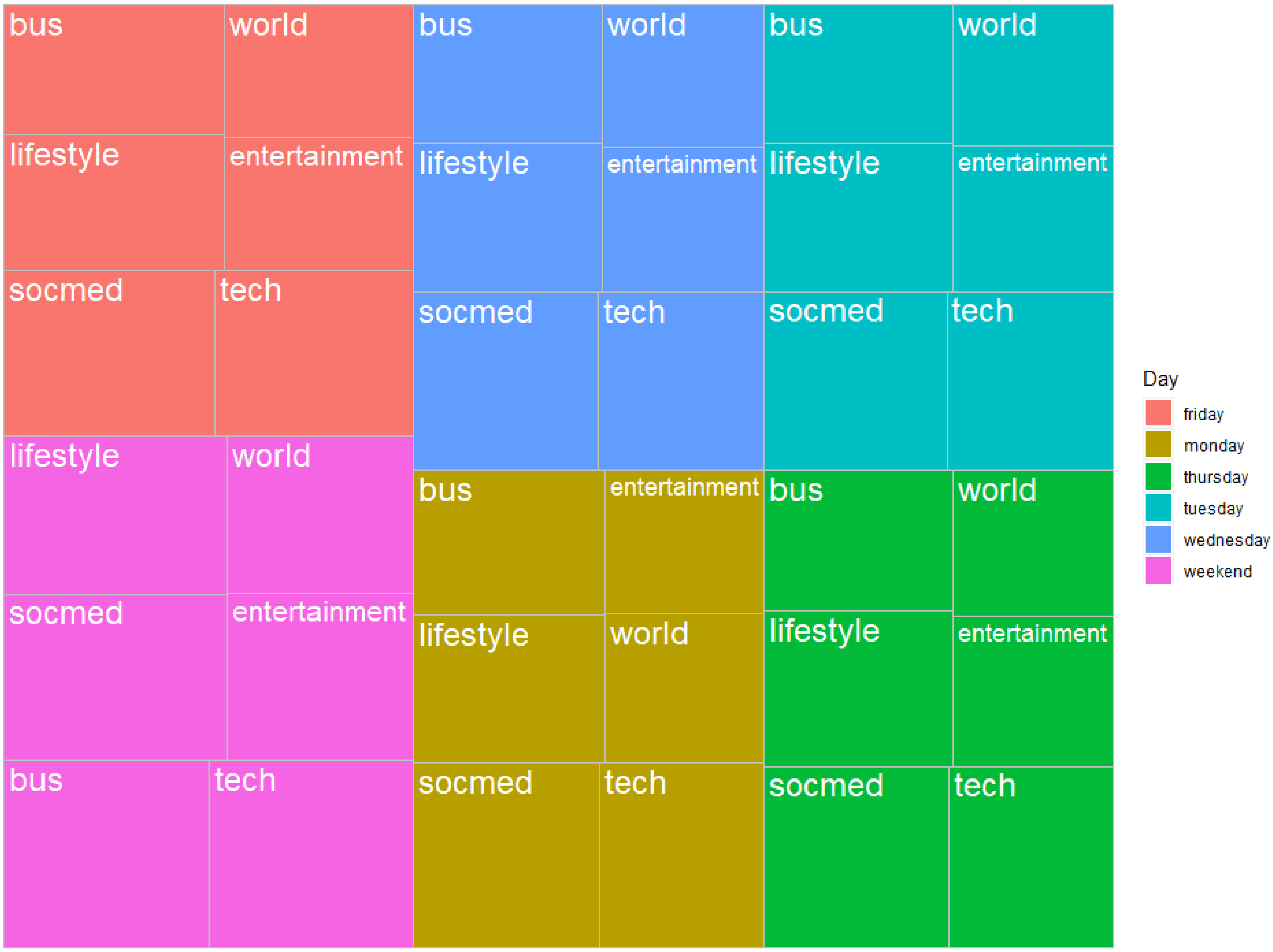
As we can see, most of the articles do not reach the popularity benchmark.

Articles written in the weekend are generally more popular



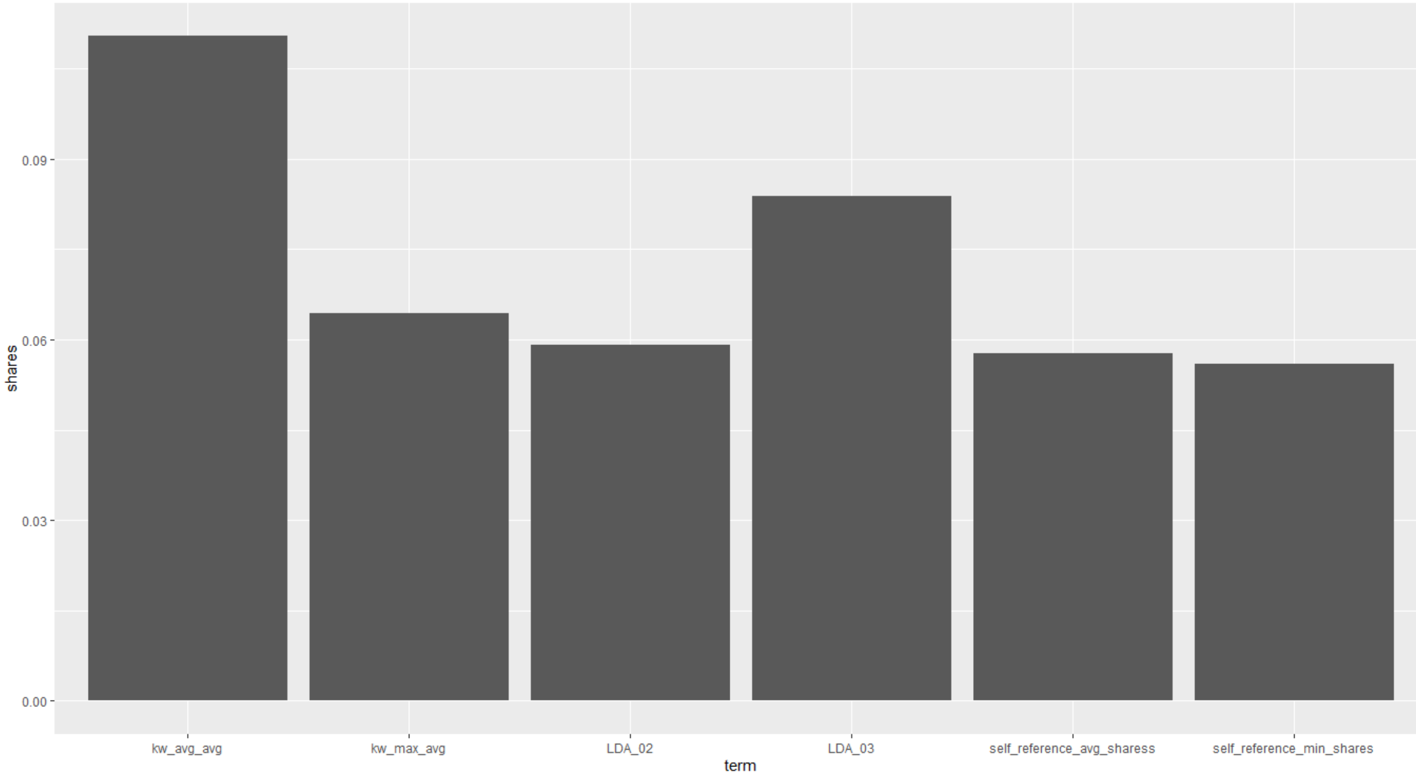
However, a causal relationship can not be established without further investigation and or a statistical experiment. We believe there is a significant relationship between time and the popularity of the article.

However, we didn't find any particular channel type that is way more popular during the weekends than in the weekdays. The following tree map shows how each channel type does on each day of the week



This would mean that each article is slightly more popular during the weekends than in the weekdays thus leading to a higher grand average of shares during the weekend.
But as you can see from the two boxplots, there are *viral* articles that come from both weekends and weekdays.

There is no one variable which has very high correlation with the shares variable:



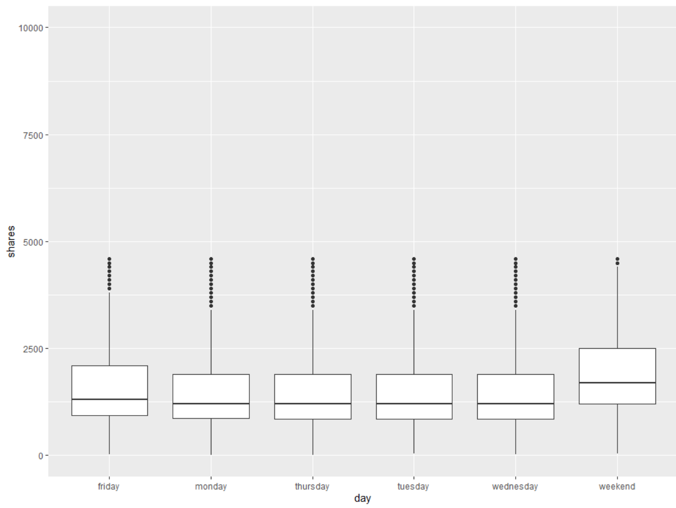
That's why we cannot reduce the number of features in the data much at all. However, using 60 features in our model can lead to several challenges:

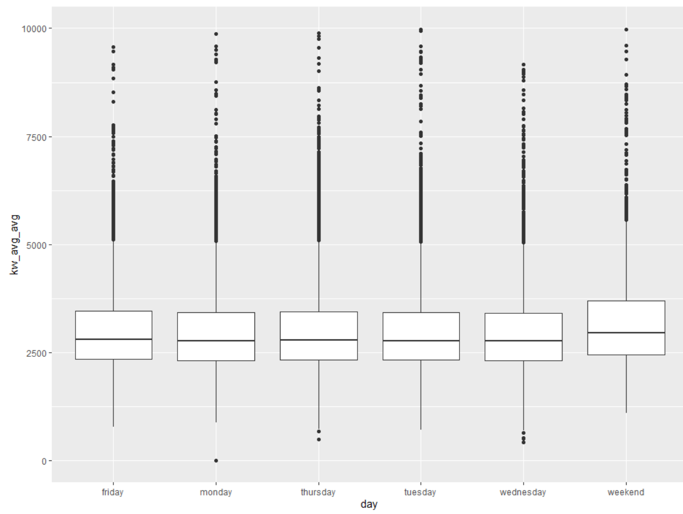
1. Tough to collect all variables from each article
2. Chances of missing data goes high
3. Chances of having outlier values goes up
4. Higher chance of noise creeping in

We used the AKAIKE information criterion in order to reduce the number of features. We were able to cut the number of features by nearly half. This translates to lower cost of model deployment and usage.

As we can see, the average number of keywords being used follows the same trend as the number of shares.

But the number of keywords used follows the trend of increased shares during the weekend:





Shares Each Day

Keywords Each Day

The other keyword attributes that measured the keywords showed similar trends but not as pronounced.

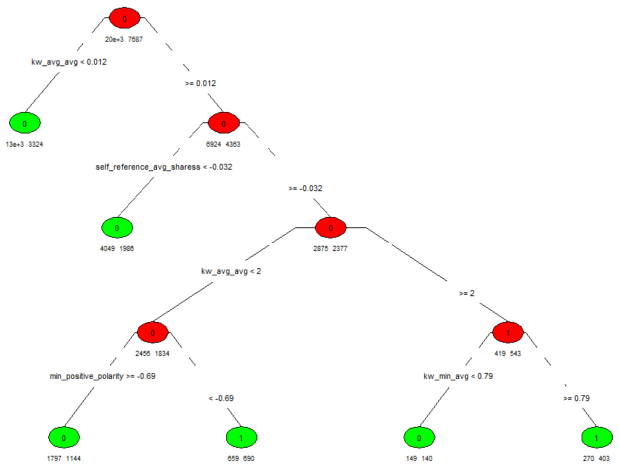
The above graph could indicate the importance of using keywords in your articles. For two reasons: one- people search according to keywords and two- search engines use keywords to rank the pages. Which implies that the more number of keywords in our article the more number of people will see it in their search results

MODELS USED

DECISION TREE

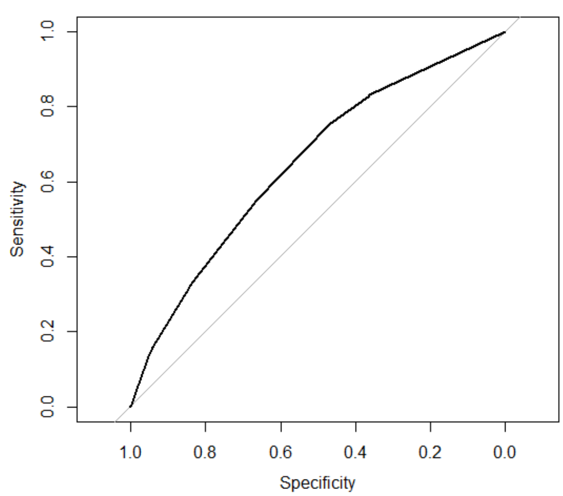
UNPRUNED TREE

We fit a decision tree model on the cleaned data.

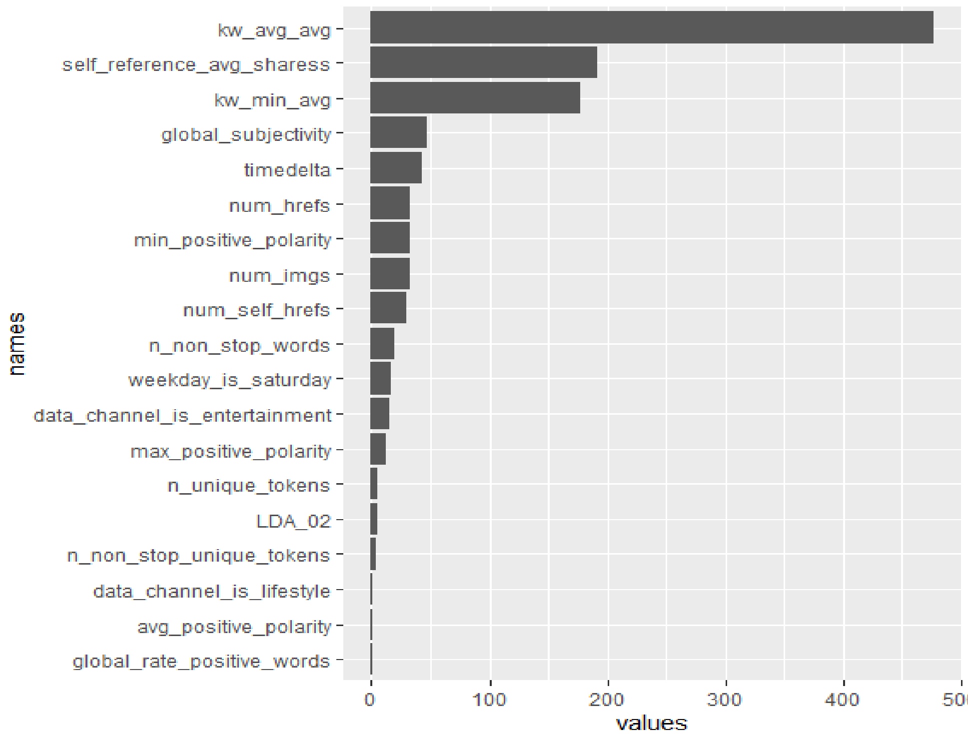



```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 8264 2722
         1  450  457

               Accuracy : 0.7333
                 95% CI : (0.7252, 0.7412)
    No Information Rate : 0.7327
    P-Value [Acc > NIR] : 0.4471

                  Kappa : 0.1192

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.14376
            Specificity : 0.94836
         Pos Pred Value : 0.50386
         Neg Pred Value : 0.75223
```

Our model scored high in specificity (95%). This is in line with our expectation that a False Positive is far more costly than a false negative. However, we were able to simplify the model further without much difference in the specificity at all. The AUC score is 0.65.



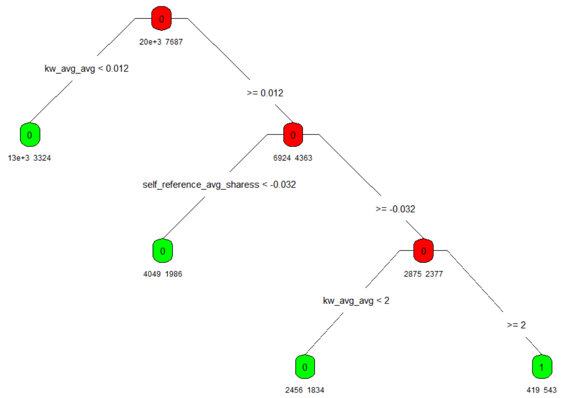
The variable importance plot



The variable importance plot re-iterates the trend shown previously about the keywords features. The average number of keywords used is the most important factor in the model

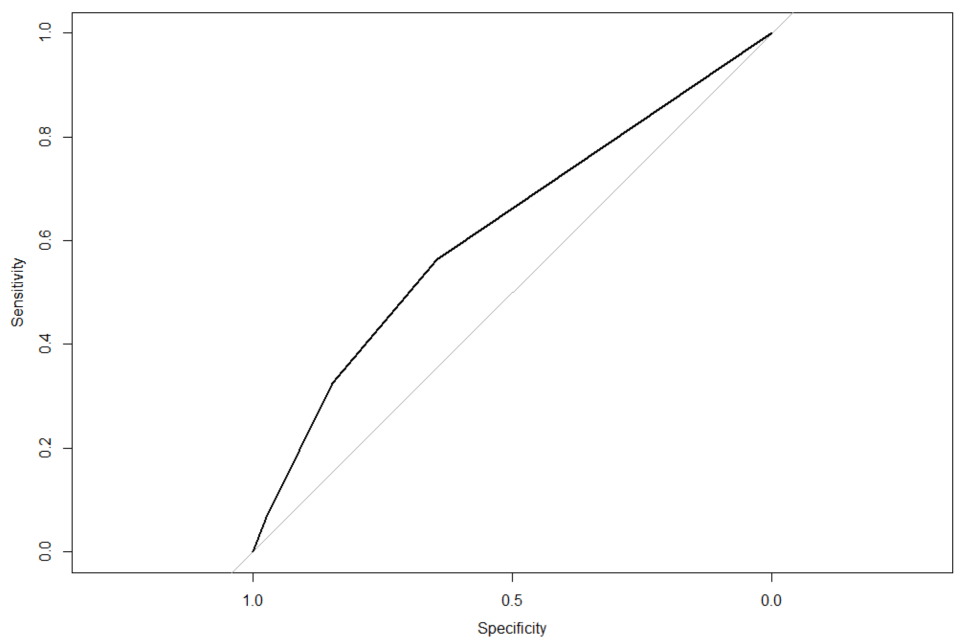
PRUNED TREE

The pruned tree performs at a similar level. The specificity is in fact slightly higher than before and the overall accuracy didn't drop much at all. For all practical purposes, the pruned and unpruned trees are the same.

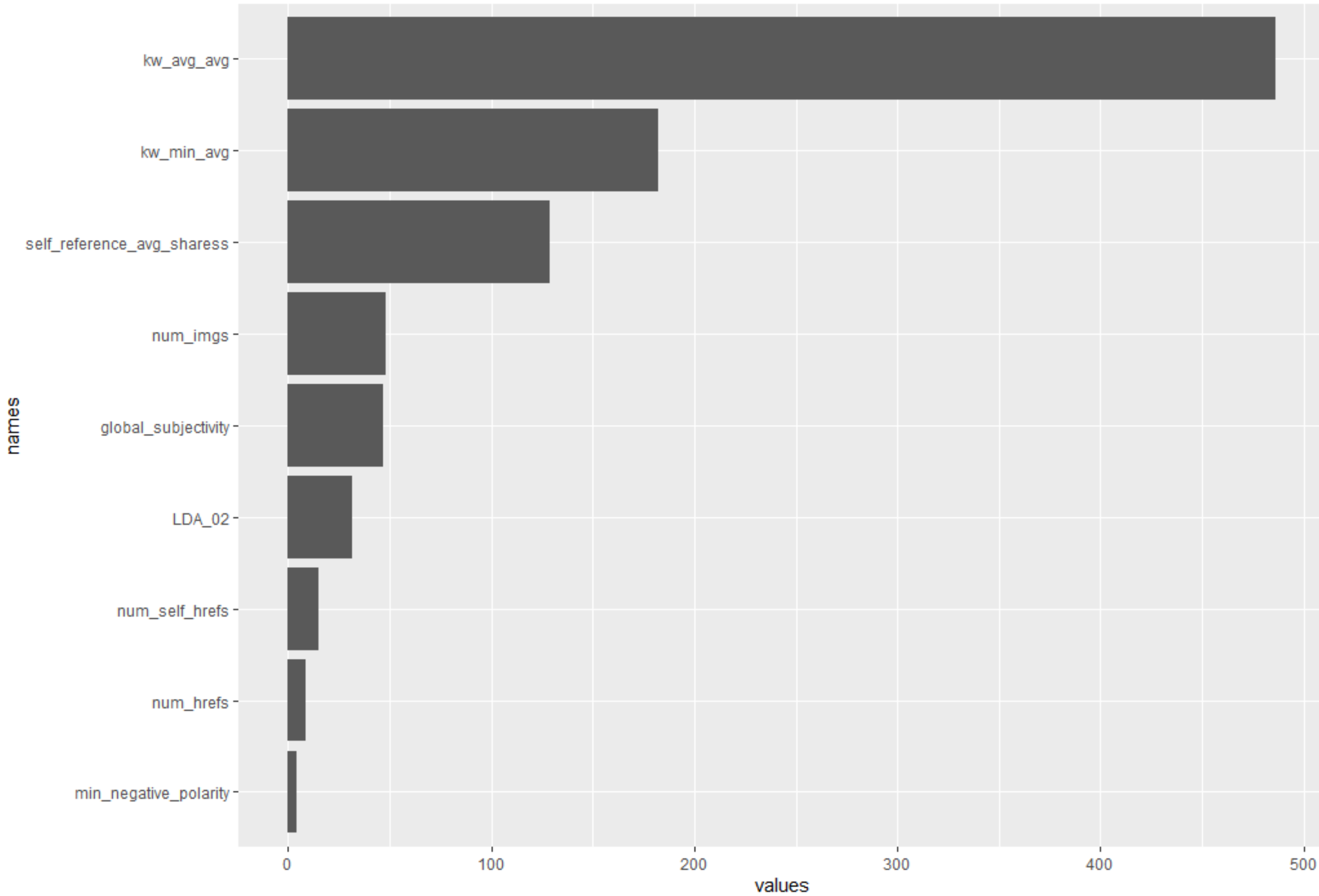



```
> confusionMatrix(default.ct.pred.test, test
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 8502 2966
         1  212  213

               Accuracy : 0.7328
                 95% CI : (0.7247, 0.7407)
    No Information Rate : 0.7327
    P-Value [Acc > NIR] : 0.4965

                  Kappa : 0.0589

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.06700
            Specificity : 0.97567
```

The accuracy is 72.1% (down from 72.6%) but the specificity is 96% (up from 95%).



The variable importance plot of the smaller tree. The smaller tree highlights the same trend as the bigger one but uses fewer features.



LOGISTIC MODEL

A logistic model over the data yielded the following result

```
Call:
glm(formula = Popularity ~ ., family = "binomial", data = training)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-1.9306  -0.7929  -0.6244   1.0994   2.3518  

Coefficients:
                                Estimate Std. Error z value Pr(>|z|)    
(Intercept)                    -0.994590   0.017098 -58.171  < 2e-16 ***
timedelta                       0.172795   0.015181  11.382  < 2e-16 ***
n_tokens_title                  0.005165   0.014145   0.365 0.715019    
n_unique_tokens                -0.274128   0.040301  -6.802 1.03e-11 ***
n_non_stop_words               -0.106088   0.022769  -4.659 3.17e-06 ***
n_non_stop_unique_tokens        0.124984   0.037098   3.369 0.000754 ***
num_hrefs                       0.093067   0.017115   5.438 5.40e-08 ***
num_self_hrefs                 -0.072708   0.015281  -4.758 1.96e-06 ***
num_imgs                        0.030231   0.016087   1.879 0.060212 .  
num_videos                      0.038978   0.014165   2.752 0.005927 ** 
data_channel_is_lifestyle1     -0.153382   0.058302  -2.631 0.008518 ** 
data_channel_is_entertainment1 -0.482376   0.039274 -12.282  < 2e-16 ***
kw_min_avg                     -0.039873   0.015472  -2.577 0.009964 ** 
kw_avg_avg                      0.392657   0.018549  21.169  < 2e-16 ***
self_reference_avg_sharess      0.269073   0.013661  19.696  < 2e-16 ***
weekday_is_monday1              0.044586   0.036153   1.233 0.217483    
weekday_is_saturday1            0.473672   0.052034   9.103  < 2e-16 ***
LDA_02                         -0.158973   0.017107  -9.293  < 2e-16 ***
global_subjectivity             0.081268   0.018021   4.510 6.49e-06 ***
global_rate_positive_words      0.028480   0.017399   1.637 0.101649    
avg_positive_polarity          -0.041860   0.022201  -1.886 0.059358 .  
min_positive_polarity          -0.063755   0.017371  -3.670 0.000242 ***
max_positive_polarity           0.007807   0.024600   0.317 0.750962    
min_negative_polarity          -0.009475   0.016783  -0.565 0.572362    
max_negative_polarity          -0.007190   0.014813  -0.485 0.627390    
abs_title_subjectivity          0.036354   0.015366   2.366 0.017985 *  
abs_title_sentiment_polarity    0.030450   0.014998   2.030 0.042329 *  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 36356  on 30833  degrees of freedom
Residual deviance: 33696  on 30807  degrees of freedom
AIC: 33750

Number of Fisher Scoring iterations: 4
```

However, the specificity was very low. We tried to improve the model by removing the variables which do not have a significant contribution to the model using the p-values.

The new model after removing the frivolous variables:

```
Call:
glm(formula = Popularity ~ . - n_tokens_title - weekday_is_monday - 
    global_rate_positive_words - max_positive_polarity - min_negative_polarity - 
    max_negative_polarity, family = "binomial", data = training)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-1.9326  -0.7929  -0.6246   1.1002   2.3406  

Coefficients:
                               Estimate Std. Error z value Pr(>|z|)    
(Intercept)                    -0.98664    0.01581 -62.410  < 2e-16 ***
timedelta                       0.17421    0.01476  11.802  < 2e-16 ***
n_unique_tokens                -0.26683    0.03853  -6.926 4.34e-12 ***
n_non_stop_words               -0.09168    0.01920  -4.774 1.81e-06 ***
n_non_stop_unique_tokens        0.13033    0.03611   3.610 0.000307 ***
num_hrefs                       0.09171    0.01696   5.406 6.43e-08 ***
num_self_hrefs                 -0.07073    0.01523  -4.646 3.39e-06 ***
num_imgs                        0.03010    0.01606   1.874 0.060917 .  
num_videos                      0.03956    0.01408   2.810 0.004950 ** 
data_channel_is_lifestyle1     -0.15260    0.05823  -2.620 0.008780 ** 
data_channel_is_entertainment1 -0.48161    0.03886 -12.395  < 2e-16 ***
kw_min_avg                     -0.03924    0.01546  -2.539 0.011125 *  
kw_avg_avg                      0.39041    0.01846  21.145  < 2e-16 ***
self_reference_avg_sharess      0.26880    0.01364  19.707  < 2e-16 ***
weekday_is_saturday1            0.46783    0.05160   9.066  < 2e-16 ***
LDA_02                         -0.16379    0.01683  -9.734  < 2e-16 ***
global_subjectivity             0.09289    0.01675   5.547 2.90e-08 ***
avg_positive_polarity          -0.03407    0.01736  -1.963 0.049662 *  
min_positive_polarity          -0.07197    0.01669  -4.313 1.61e-05 ***
abs_title_subjectivity          0.03174    0.01508   2.104 0.035338 *  
abs_title_sentiment_polarity    0.03073    0.01498   2.051 0.040263 *  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 36356  on 30833  degrees of freedom
Residual deviance: 33702  on 30813  degrees of freedom
AIC: 33744
```

The confusion matrix of the best logit model :

```
> confusionMatrix
            predicted_values
actual_value FALSE TRUE
           0  4659 1808
           1  1096 1247
> 
```

The specificity is only 75%

RECOMMENDED MODEL

We recommend the decision tree model as it has higher specificity and reduces the number of False positives. This is done because a False positive would result in publishing an unpopular article, this is disastrous for online media houses wherein the likeability and popularity are paramount to their survival.

Also, a false negative would simply mean that the media house has to review the article again before publishing. This has labour costs associated with it but is far less costly than losing the customer base due to unpopular articles.

MANAGERIAL INSIGHTS AND RECOMMENDATIONS

POPULARITY IS A FUNCTION OF TIME

Although popularity is higher during the weekend, the articles published on weekdays aren't too far behind. If an article must be published and its popularity is under question try to publish it during the weekend

IMPORTANCE OF KEYWORD

The average number of keywords used in the title and the body of the article have a big impact on the popularity of the article. Try to maximise the number of relevant keywords

However, closeness to the actual topic is a close second factor. So including frivolous keywords could backfire.

DISCRETION IS ADVISED

We have tried to reduce the number of False positives as much as possible while keeping the overall accuracy over 70%. However, the situation may change in the future. For instance, if the labour costs become too high, it may be more profitable to reduce the specificity.

Right now, however, the potential alienation of customer base and the resulting loss of ad revenue far outweigh the labour costs. It is sensible to review articles than to leave them to chance.

COST BENEFIT ANALYSIS

The ad revenue generated by per hour by a website such as New york times is roughly $87,000[2]. In contrast the average hourly rate for a newswriter is $22[3] . Given this data, it makes sense to prioritize the ad revenue over the labour costs.

It is difficult to obtain the information needed at the required granularity. For instance, NY times announced it's operation cost and cost of labour to be $250 million but this also includes their non-writing staff as well (which includes computer scientists, network specialists, cameramen, anchors, electricians and so on). Because of which a cost benefit analysis can not be done accurately. However, an educated guess can be made.

REFERENCES

1. https://www.pewresearch.org/journalism/2014/03/26/revenue-sources-a-heavy-dependence-on-advertising/: Portion of revenue given by advertisements
2. https://nytco-assets.nytimes.com/2022/02/NYT-Press-Release-12.26.2021-PpCb082.pdf: Revenue over the quarter for NYT
3. zippia.com/news-writer-jobs/salary/ : Average salary for a newswriter