Analyzing Article Popularity

Capstone II - Final Report

Springboard
Data Science Career Track

Tara Crutchfield
December 28, 2020

Table of Contents

Appendix

1. Objective

The objective of this project was to find what key features play into the popularity of an online news article and to determine if a model could accurately predict popularity. To answer this question, the project focused on data collected from Mashable, a multi-platform media company that publishes a wide variety of content to a global audience. Mashable was chosen not only because the data was available and easy to expand upon, but also due to its wide variety of article topics, ranging from coverage of current world issues to clickbait and entertainment.

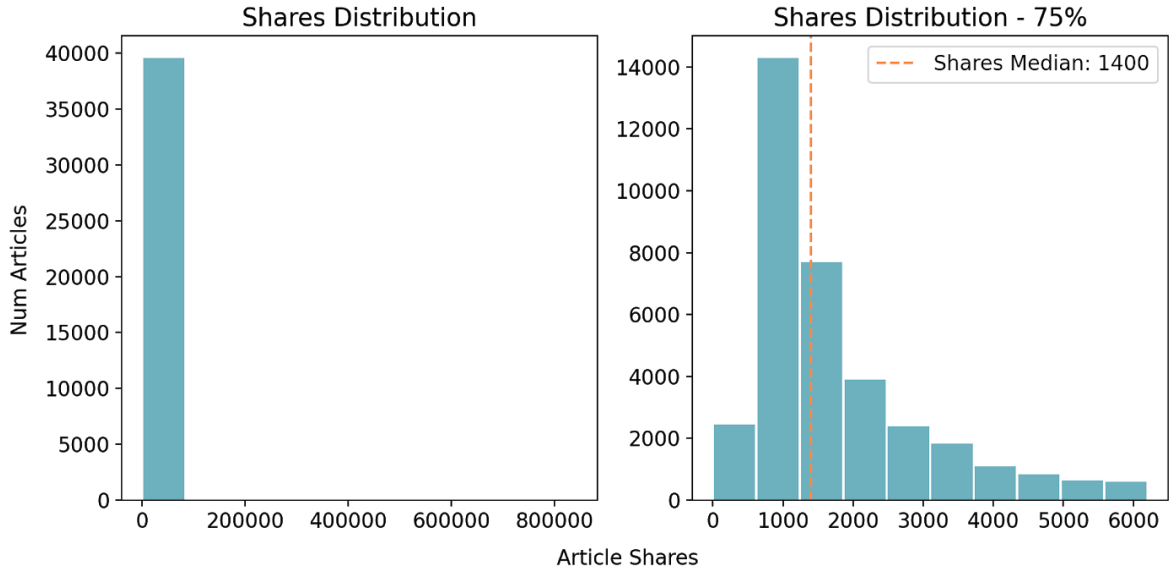
This report is broken up into eighth separate sections as well as an appendix for larger figures and tables. Section 2 covers the dataset, including origins and feature descriptions. Section 3 briefly describes the python packages and versions utilized for the project. Section 4 details the data wrangling and cleaning process and summarizes the final dataset that was used throughout analysis. Section 5 covers all exploratory data analysis and the findings of this process. Section 6 and Section 7 detail modeling and optimization, as well as the results of the final model. Lastly, Section 8 concludes with a summary of important findings.

2. Dataset

The utilized dataset was found using the University of California, Irvine's Machine Learning Repository, an archive of machine learning datasets created by David Aha in 1987. The dataset 'OnlineNewsPopularity.csv' was created by Kelwin Fernandes for a study on online news popularity done at the University of Porto, Portugal and contained 60 columns and a total of 39,644 rows. Of these columns, all were numeric except for one categorical column containing all article URLs. Table I, located in the appendix due to size, contains the name and descriptions for all 60 of these columns.

The target feature in the dataset was the **shares** column. These values have a normalized, right-tailed distribution, ranging between a minimum of 0 shares and a maximum of 843,300 shares. Figure 1 shows the distribution of these shares as well as the distribution focused on the articles below the 75th percentile threshold for better shape clarity.

Figure 1: Histogram of shares. To the left is the distribution of all shares, to the right is the distribution of shares below the 75th percentile (6200 shares).



3. Packages

Throughout the project, JupyterLab (6.1.4) was utilized for all coding purposes. Pandas (1.0.1) and Numpy (1.18.1) were used as basic packages and Matplotlib (3.1.3) and Seaborn (0.11.0) were utilized for plotting and visualizing data. For web scraping, Beautiful Soup 4 (4.8.2) and Requests (2.22.0) were used, as well as Asyncio for multithreading. Lastly, Scikit-learn (0.23.2) was used as the machine learning library.

4. Data Wrangling

The original dataset was processed in a way such that all features other than the article URLs were numerical. While at first glance the data appeared to be complete, it soon became apparent that the six separate **data_channel_is_** columns only covered 84% of all entries. There was also some concern as to the accuracy of the various keyword features as well, as for numerous entries all keyword related values were set to zero.

These factors lead me to investigate the Mashable website and collect additional features directly from each article's HTML. These features, shown in Table 2, included the data channel, publish date, article title, and all keywords for each article and were saved in the file 'Updates.csv'.

Table 2: Description of added features

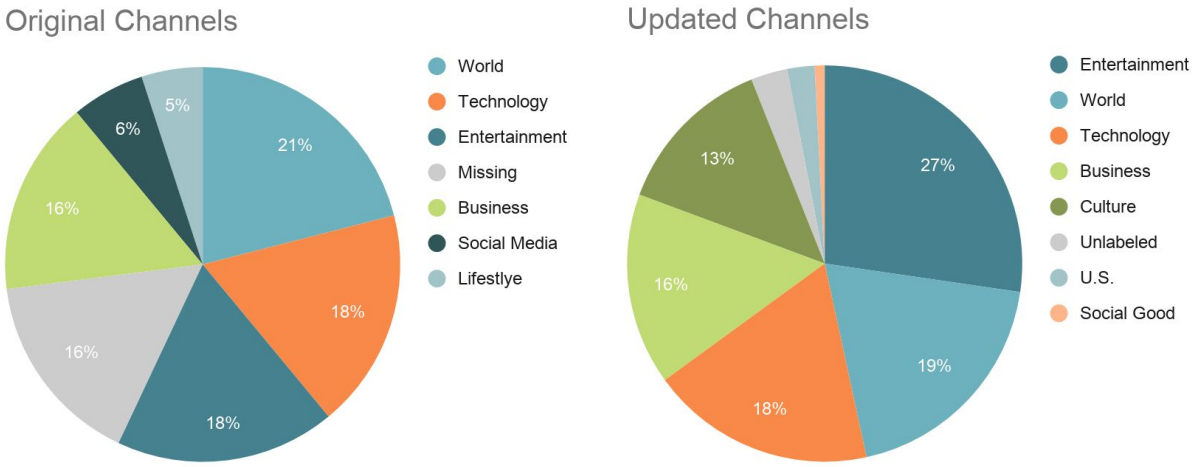
No.	Variable Name	Variable Description	Unique Values	Dtype	Non-Null
0	url	URL of articles	39644	object	100.0%
1	channel	Article data channel label	8	object	99.6%
2	date	Article publish date	738	object	99.7%
3	title	Article title	39272	object	99.1%
4	keyword	All article keywords	34545	object	99.6%

This dataset was merged with the original using an inner join on the URL's . As not all the articles were still available on the website, those missing were subsequently dropped, decreasing the dataset to 39,468 rows.

4.1 Data Channels

While some of the articles' data channels remained unlabeled, the scrape showed two key differences. The updated data channels did not include the labels *Social Media* or *Lifestyle* and instead had the labels *US, Culture,* and *Social Good*. Figure 2 presents the spread of these channels from both the original dataset as well as the updated dataset.

Figure 2: Two pie charts showing the different distributions of the article data channels before and after updating.



A majority of the articles remained under the *Entertainment, World, Technology*, and *Business* data channels. In terms of the new channels, the *Culture* label was the largest while *US* and *Social Good* only made up a small percentage. Due to the size, these two labels, as well as the unlabeled articles, were batched into a new label *Other*. With this, all **data_channel_is_** columns (No. 13-18) were dropped in favor of the updated **channel** column.

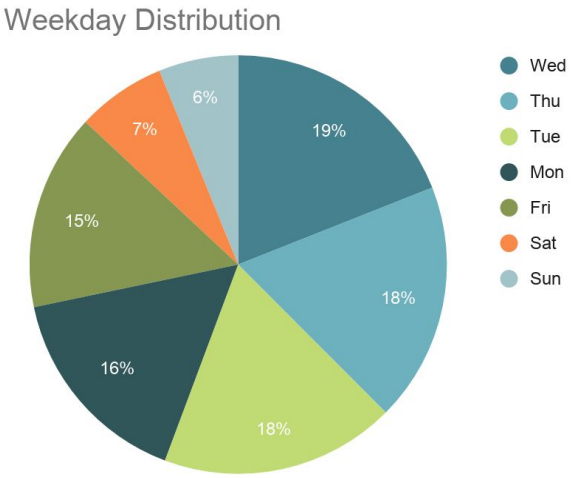
4.2 Days of the Week

The **date** column of 'Updates.csv' included both the day of the week as well as date each article was published. Thus this column was split such that the weekdays had a separate column, **weekday**, and the **date** columns could be converted to a datetime object.

While there were some discrepancies between the original dataset's **weekday_is_** columns and the new **weekday** column, the difference was not as dramatic as found with the data channels. Still, for certainty, all **weekday_is_** columns (No. 31-38) were dropped in favor of the **weekday** column.

Figure 3 shows a pie chart of the weekday distribution. Noticeably less articles are published on weekends as opposed to weekdays. To combat the imbalance, articles published on Saturday and Sundays were combined under the label *Weekend*.

Figure 3: Pie chart showing the distribution of the weekday articles are published



4.3 Keyword Features

The values of the **keyword** column consisted of a list of each article's keywords. In order to understand these values better, as well as imitate the keyword features of the original dataset (No. 19-27), the dataset was melted such that there was a separate entry for each article and keyword. These entries were then joined with the shares of the respective article.

Using this melted dataset, the minimum, maximum, and mean shares associated with each keyword could be determined, thus providing a method of measuring keyword performance for individual articles. Using the mean shares of each keyword, the best and worst performing, as well as the median, keywords could be determined for each article and were added to the columns **kw_max, kw_min,** and **kw_avg** respectively. The maximum, minimum, and mean shares of these keywords were also added, replacing the original keyword features. Figure I in the appendix shows a flowchart of this process and clips of the dataset for better clarification .

4.4 Final Dataset

After data wrangling and cleaning, the final dataset included a total of 53 columns and 39,468 rows. 46 of these columns were numerical, 7 were categorical, and 1 was composed of datetime objects. Table II, located in the appendix, presents all of these columns as well as a description of each feature.

5. Exploratory Data Analysis

5.1 Statistics and Distribution

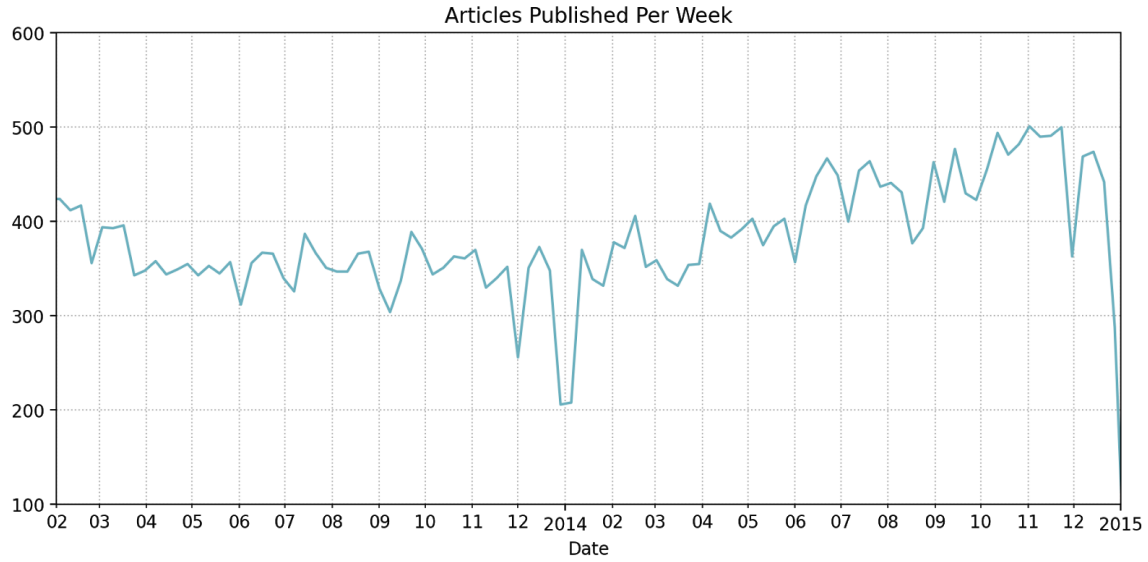
The summary statistics of the dataset are displayed in Table III. This table includes the mean, standard deviation, minimum values, and maximum values for each numeric feature. Figure II shows the distribution of these values, limited in such a way that only 90% of the data is present inorder to remove outliers and improve clarity.

A majority of these distributions appeared normal or one-tailed. The only features that differed from this pattern were some of the keyword features as well as the min/max polarity features. Overall, it appeared that a majority of Mashable articles tend to be on the positive side and slightly more subjective.

5.2 Time

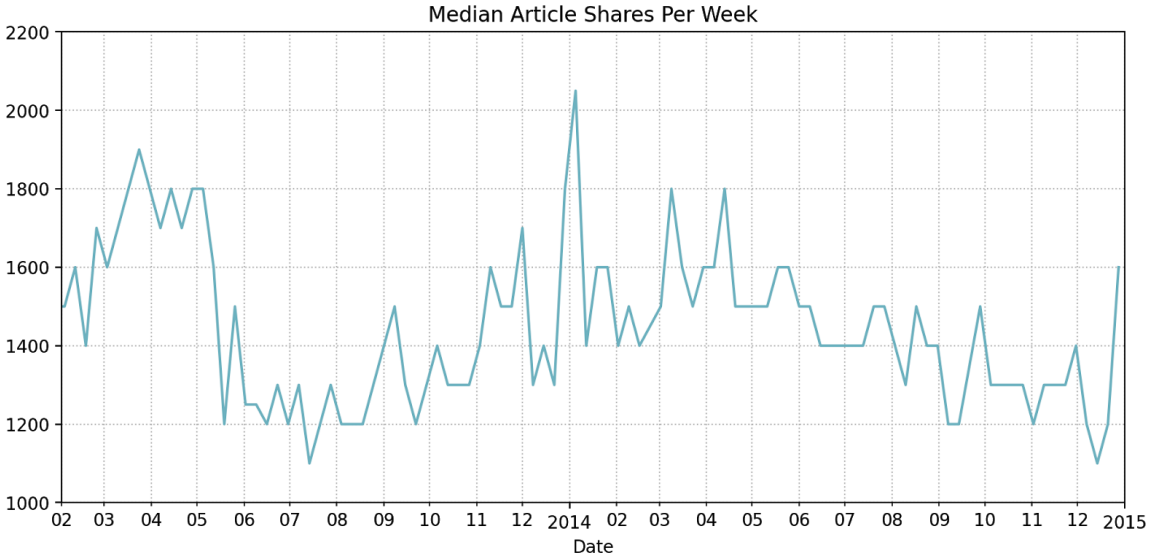
Using the date column, it was possible to plot out shares over time. Figure 4 shows the amount of articles published per week. Over the two years, this amount had increased by almost 100 articles. An interesting thing worth noting is that the amount of articles published seemed to drop by 50 articles during the last week of May as well as the first week of September. As one could expect, the amount of published articles also drastically dropped in the last weeks of November and December as writers were probably on holiday leaves.

Figure 4: Plot of the amount of articles published each week.



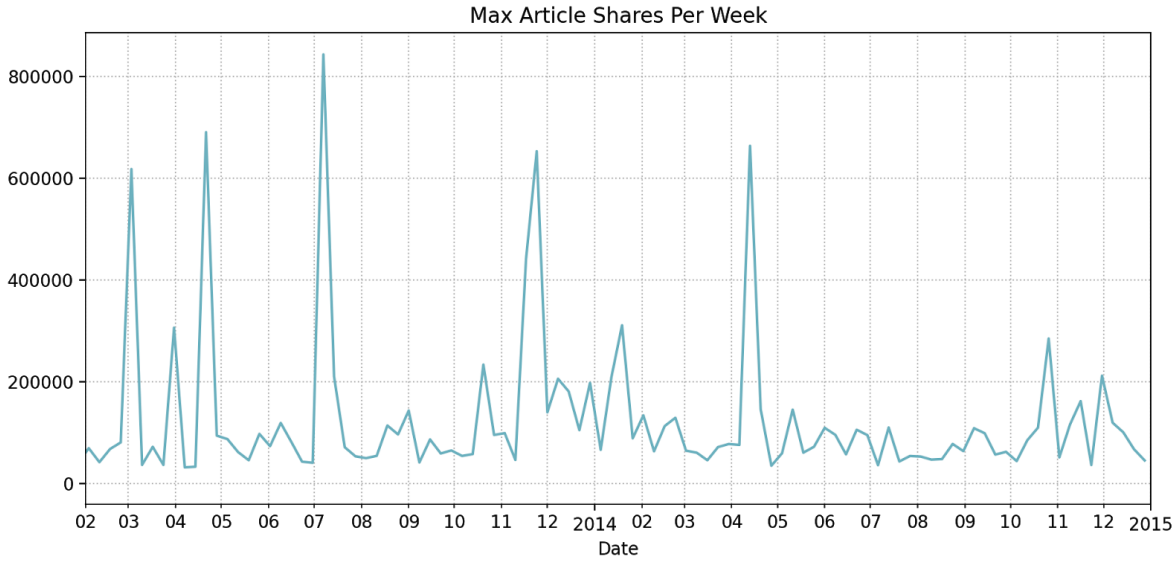
In Figure 5, the median shares of the articles published in each week are shown over time. Despite the rise in the amount of articles published, the median shares did not grow over the years, rather it appears to have fallen. Also, while it is hard to know for sure as the data only covers two years, it does appear that there may be some seasonality to the median shares: rising in the winter and falling in the summer.

Figure 5: Plot of the median shares of articles published in the week.



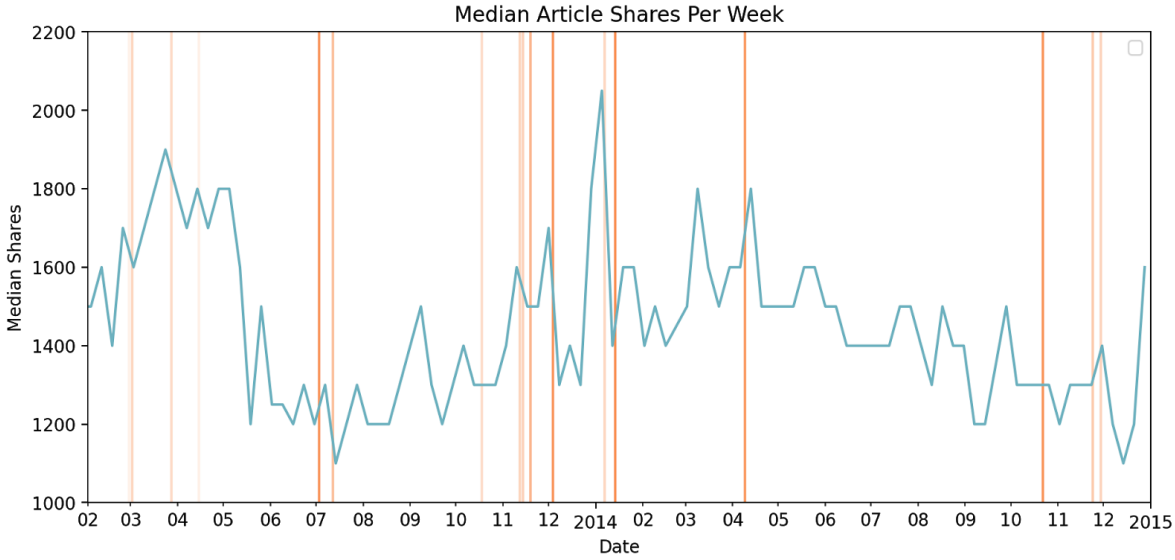
Lastly, the maximum shares of articles published in a week are plotted out in Figure 6. While a majority of these articles sit below 100,000 shares, there are a select few that really appear to jump significantly higher. These articles are considered 'viral' as they capture significantly more attention than what seems to be typical, even of popular articles.

Figure 6: Plot of the shares of the most popular article per week.



Comparing the max and median weekly article shares, there appeared to be a pattern in which the amount of viral articles within a given time period seems to align with a raise in the overall median of articles published in a week. Figure 7 shows the graph from Figure 5 with orange lines marking the publish dates of viral articles.

Figure 7: Plot of the median shares of articles published in the week. The orange lines represent viral articles with opacity corresponding with the amount of shares, with darker lines representing higher shares.



Where the orange lines are most concentrated, the weekly median tended to be high or on the rise. It also appears that viral articles may be more common within the winter seasons. It also seems like the more saturated lines, or viral articles with higher shares, tend to be followed by

dips in the median. This may indicate that viral articles can potentially direct attention away from articles published in the weeks after it.

5.3 Keywords

Altogether there were 16,724 difference keywords among all these articles. Of these keywords, less than half (8035) were actually used for more than one article. The most common keywords found were all associated with the most popular data channels: *world, tech, entertainment, culture,* and *business.*

In terms of performance, Table 3 shows the most common keywords found for each keyword feature. Many of these are the data channel keywords as they are so commonly used and cover such a wide range of articles. Besides these common keywords, it appears that *sports* is a recurring worst performing keyword, while *gadgets* and *viral video* are some of the best performing keywords

Table 3: Most common keywords found in kw_min, kw_avg, and kw_max. These entries are in order of occurrences, with the highest count on the top.

Worst Performing	Average	Best Performing
world	world	gadgets
business	entertainment	tech
sports	tech	entertainment
television	business	u.s.
entertainment	culture	viral video

5.4 Viral Articles

Going back to viral articles, I was curious to see whether these articles had any overarching similarities or gained popularity though other, more random, means. Figure III shows a heatmap of the correlation matrix focused on how the shares of these viral articles relate to its other features. In this image, the redder shades represent positive correlation while the bluer shades represent negative correlation.

From this figure it's evident that many of these features are correlated in some manner. Of the positive correlation, it appears that viral articles typically have well performing keywords, as well as a higher amount of shares in its lowest performing keyword. Other notable features include overall article content positivity, a larger number of references, and longer articles.

In terms of negative correlation, features that stand out are long, subjective titles and the large number of keywords. Interestingly it appears that the shares of the least shared article in the best performing keyword also negatively correlated with the amount of shares of a viral article.

6. Machine Learning Models

In order to ensure that all features were numeric, the categorical columns **weekday** and **channel** were converted to dummy features. The columns **date** and **timedelta** were dropped to remove the element of time, as well as the columns **kw_max, kw_min, kw_avg** as there were too many keywords to convert to dummy features. In all, this expanded the dataset such that there were 58 columns, all numerical besides the urls used as the index.

6.1 Unsupervised Learning

In order to explore the unseen trends present in the data, k-means clustering was utilized to bunch points into categories. To determine the amount of clusters to best explain the data the 'elbow method' was used. This process included creating ten separate models with n-clusters ranging between 1 and 10 clusters. The inertia, or within-cluster sum-of-squares, of each of these models were collected and plotted against the amount of clusters that the respective model had, as seen in Figure 9. Through this method it was determined that five clusters would best encompass the variance of the dataset.

Figure 9: Example of the 'elbow method'. The orange line marks the inertia of the model with 5 clusters. A model with any more clusters would be overfitting to the dataset.

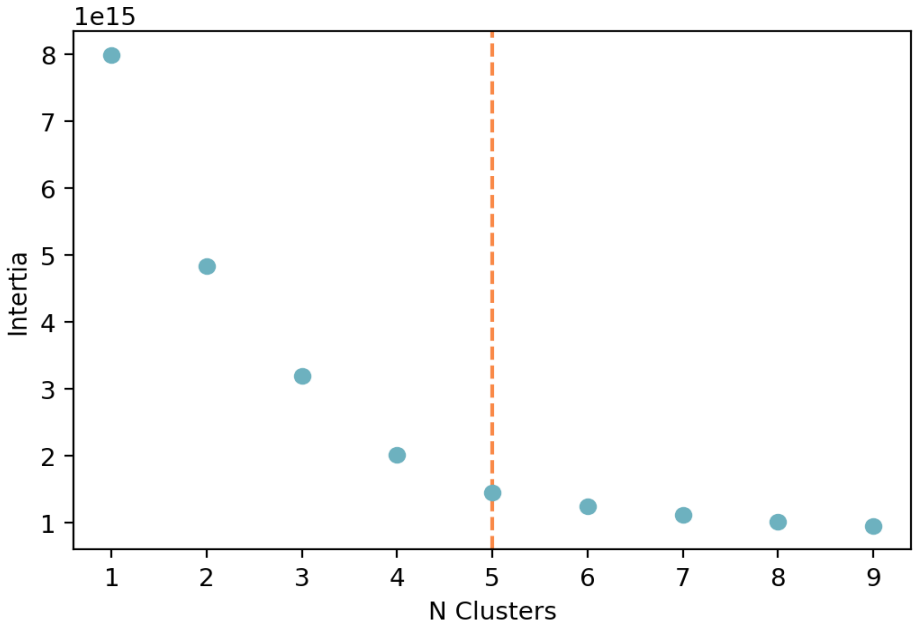


Figure IV in the appendix shows the different distribution between these clusters. These histograms show a lot of overlap between clusters, the main differentiating features being text features and polarity. Table 4 shows the amount of entries in each cluster, as well as their defining

features. Of these clusters, Cluster 2 held the majority of the viral articles suggesting that simple, positive articles are more likely to be popular.

Table 4: Clusters Descriptions

Cluster	No. Entries	Description
Cluster 0	128	Indistinguishable
Cluster 1	1174	Articles with no text (only singular video)
Cluster 2	10345	Long yet simple articles that are slightly more subjective and positive
Cluster 3	13010	Short yet more complex articles that are negative
Cluster 4	14809	Short yet more complex articles that are positive

6.2 Predicting Shares

When testing different models, it quickly became apparent that the dataset was not fit for predicting article shares. Even when rounding shares to the closest thousand, both logarithmic regression and random forest models fared equally poorly. This shifted the focus not on predicting articles shares but rather article 'popularity' which was determined by a percentile threshold.

6.3 Popularity - Multi-Classification

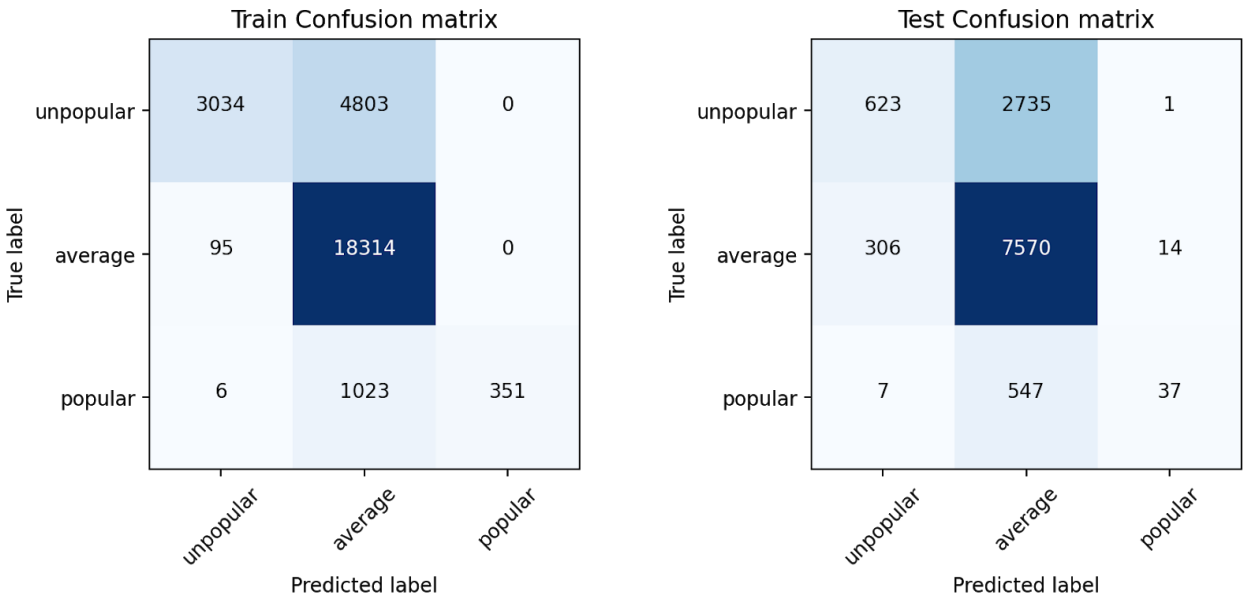
The first of these models included three separate bins with the labels *Popular*, *Average*, and *Unpopular*. Articles considered *Popular* had shares above the 95th percentile (10800 shares), while those labeled *Unpopular* had shares below the 30th percentile (1000 shares). The data was split in a stratified manner such that the training set contained 70% of the total point, with the remaining 30% as the testing set . Details regarding these sets can be found in Table 5.

Table 5: Distribution of training and testing sets from modeling

Set	Unpopular	Average	Popular	Total Entries
Train	7837	18409	1380	27626
Test	3359	7890	591	11840
Total	11196	26299	1971	39466

Five fold cross-validation was performed utilizing model accuracy to determine a best max depth of 13. The prediction results of this model can be visualized as a confusion matrix as seen in Figure 10.

Figure 10: Confusion matrix for both the training and test sets. Darker colors represent a higher density of points.



Due to the volume of *Average* labeled articles, the model predicted that the majority of articles were of average performance. Due to this imbalance the overall accuracy was 70% despite the fact that the average recall was only 0.40. Focusing particularly on the *Popular* label, the recall was as low as 0.06 despite a precision of 0.71.

In order to determine how the imbalance of data played into the model's predictions, the training set entries were limited such that there were an equal amount of each entry, with the removed entries then added to the test set. This revised distribution can be found in Table 6.

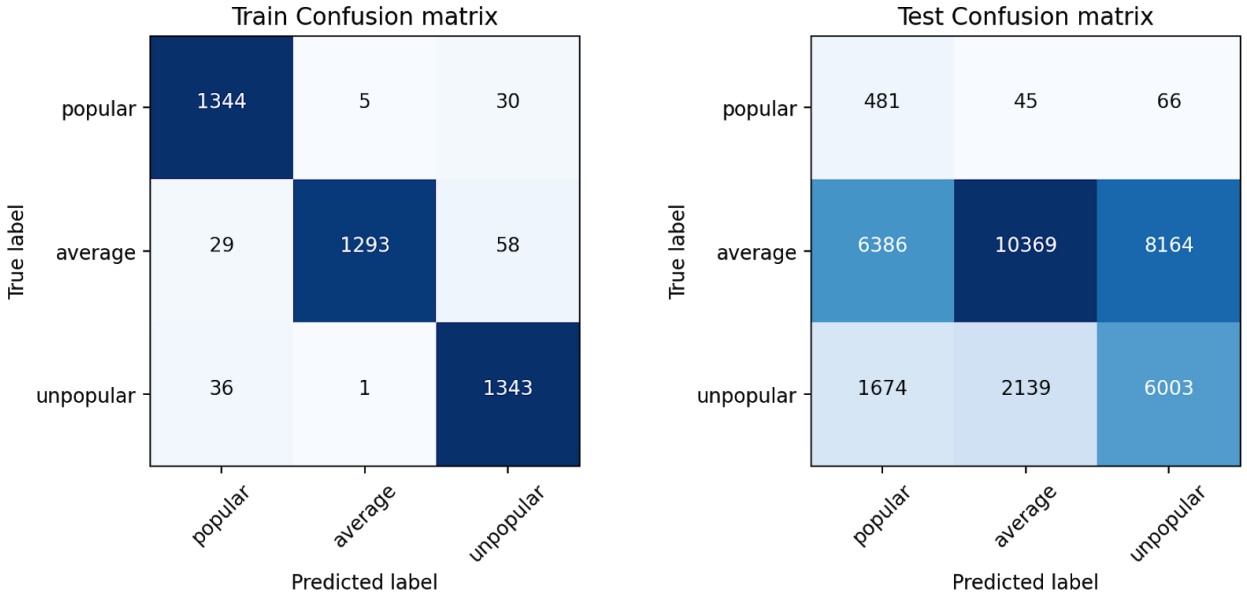
Table 6: Distribution of entries with a limited training set.

Set	Unpopular	Average	Popular	Total Entries
Train	1380	1380	1379	4139
Test	9816	26299	592	39466

Using this training set, cross-validation was performed again to determine a best max depth of 13. Figure 11 presents the results of this model as a confusion matrix. At a glance it appears that the distribution of these points have improved yet despite the increase in the

accuracy of the training set, the overall accuracy of the test set decreased to 48%. While the average recall did increase to 0.60, it came with a large hit to the average precision.

Figure 11: Confusion matrix for both the training and test sets. Despite the higher accuracy in the training set, the test set accuracy fell by more than 20%



Isolating the *Popularity* label, the results were the exact opposite of the first model. In this case the recall increased to 0.81 while the precision decreased 0.05. While this is technically a higher net positive outcome, higher precision is ultimately the desired result.

As it appeared that the models had a trade off between precision and recall, this multiclass model was shifted into a binary classification model to better monitor these qualities.

6.4 Popularity - Binary Classification

For the binary classification models, the threshold dividing popularity shifted to the 90th percentile (6200 shares). Similar to before, two models were experimented with: one leaving the training set as is and another limiting the points such that both entries were equal in the training set. These two training and test set distributions are shown in Table 7.

Table 7: Tables featuring the two separate train/test distributions for the binary classification models

a) Distribution of entries without limiting Unpopular articles in train set

Set	Unpopular	Popular	Total Entries
Train	24879	2747	27626
Test	10663	1177	11840
Total	35542	3924	39466

b) Distribution of entries in equal Unpopular and Popular articles in train set

Set	Unpopular	Popular	Total Entries
Train	2746	2747	5493
Test	32796	1177	33973

Both of these models used five fold cross-validation to determine the best max depth from a range of values between 5 and 15. The f1 score was selected as the method of determining the best depth.

The model trained on the normal training set performed with an overall accuracy of 90% and f1 score of 0.15. Figure 12 presents both the confusion matrix and ROC curves for this model. In all, the model had a precision of 0.79 and recall of 0.08, both of which were higher than the scores of the *Popular* label from the multiclass model.

Figure 12: Confusion matrix and ROC curve for the model trained on the normal training set.

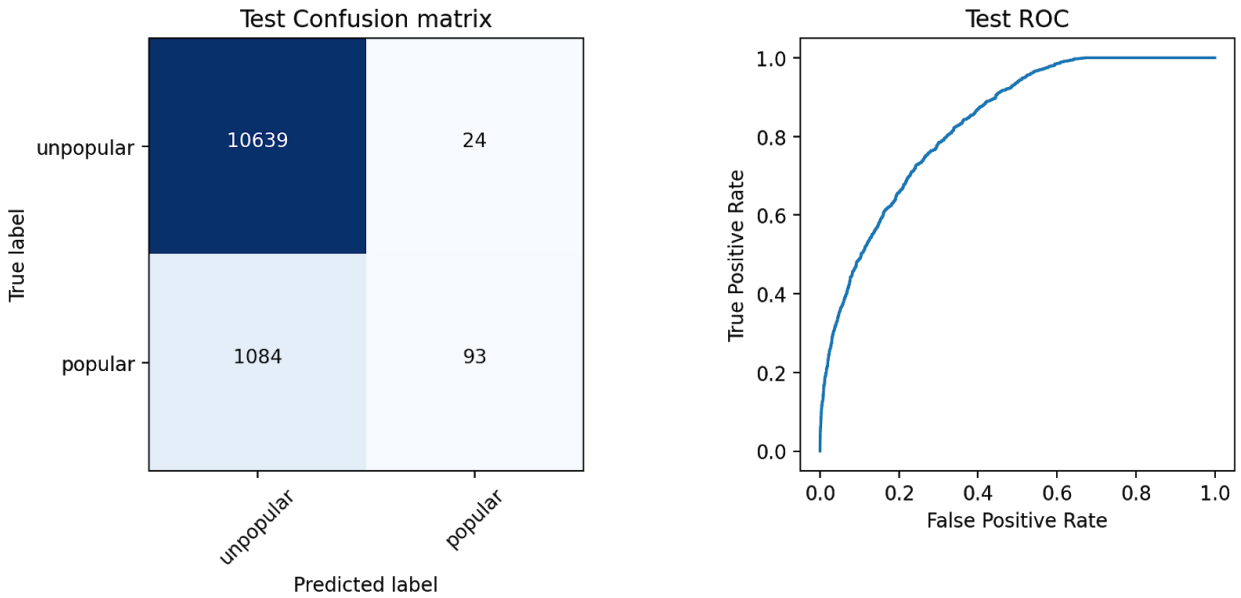
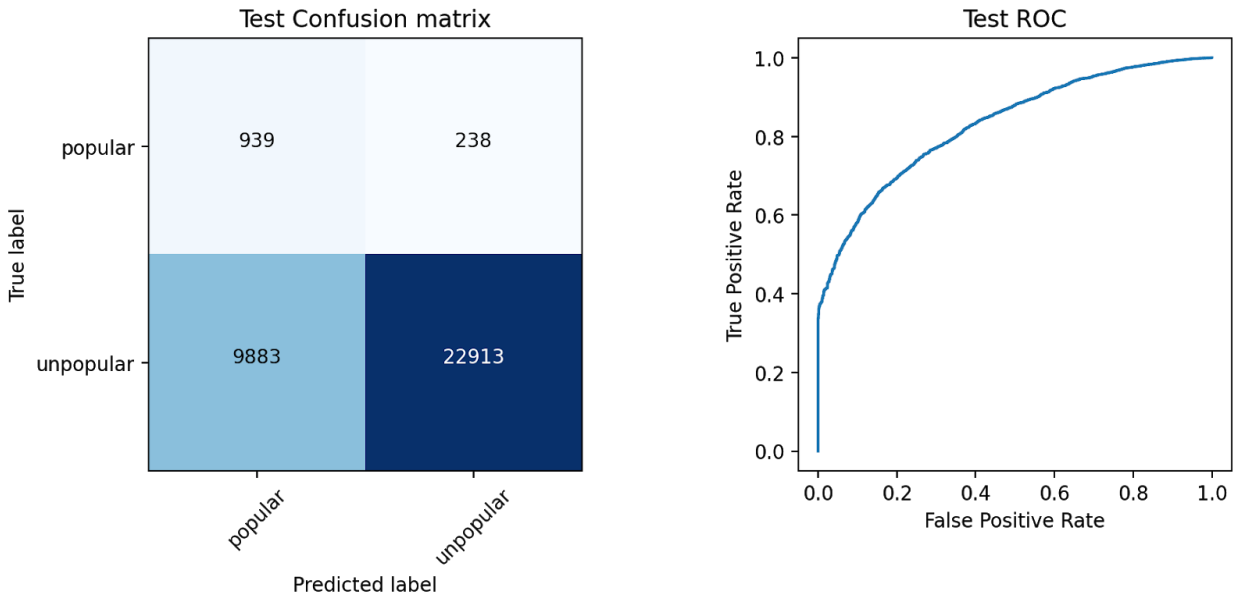


Figure 13 shows the confusion matrix and ROC curve of the model trained on the limited training set. This model had an accuracy of 70% and f1 score of 0.16. The recall and precision of this model were almost exactly the opposite as those of the earlier model: precision dropping to 0.09 and recall increasing to 0.80 Compared to the *Popular* label of the similar multiclass model, the recall fell by 0.01 but the precision increased by 0.04.

Figure 13: Confusion matrix and ROC curve for the model trained on the limited training set.



A summary of these four models can be found in Table 8. Over all, while the binary classification models suffer from the same issues as the multiclass models, they do perform slightly better and will help simplify the optimization process.

Table 8: Summary of the test scores for all four models. Here 'MC' means the multiclass models while 'BC' represents the binary classification models.

Model	Accuracy	F1 Score	Precision	Recall
MC - No Limit	70%	0.11	0.71	0.06
MC - Limited	48%	0.10	0.05	0.81
BC - No Limit	90%	0.15	0.79	0.08
BC - Limited	70%	0.16	0.09	0.80

7 Optimization

Moving forward with the binary classification model, there were two main components that needed to be optimized in order to find a balance between precision and recall: the ratio of *Unpopular* to *Popular* points in the training data and various model hyperparameters. In terms of criterion, precision was ultimately the priority as even if the model can not identify all popular articles, there should be confidence in the articles it did label as popular.

Figure 14 shows the confusion matrix and ROC curve of an unoptimized model. The training and testing tests, seen in Table 9, were split 80/20 and no changes were made to limit the training set. The model was then created with an arbitrarily set max depth of 10 with 100 estimators. Overall, this unoptimized model has an accuracy of 90% and an f1 score of 0.12. It's precision was 0.67 and it had a recall of 0.06.

Figure 14: Confusion matrix and ROC curve of the initial, unoptimized model.

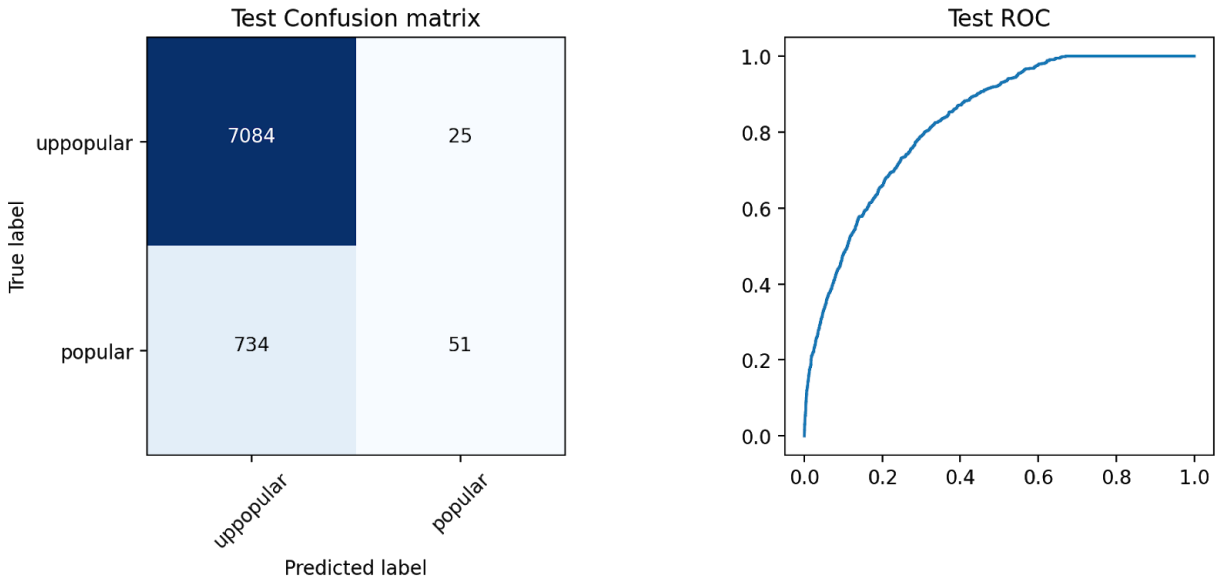


Table 9: Distribution of training and testing sets.

Set	Unpopular	Popular	Total
Train	28434	3140	31574
Test	7109	785	7894

7.1 Testing Set Distribution

The first of these important factors was the distribution of entries in the training set. As seen in the earlier models, not limiting the *Unpopular* points gave a greater total precision while limiting these points increased the recall. In order to test what distribution of points was ideal, or if a change was even necessary, different ratios were tested and compared.

The dataset had almost 10 times more *Unpopular* points than *Popular.* Using an arbitrarily chosen max depth of 10, a total of 10 models with ratios ranging between a 1:1 and no ratio (roughly 10:1) were tested 10 times each. The recall, precision, and f1 scores of the ratios were then averaged between the models. The best training set distribution was chosen by first taking all models with a precision above 0.70, then selecting the one with the highest f1 score. Table 9 shows the various scores for these five models. Using this method, it was determined that ultimately the limit was unnecessary. While limiting the training set 9:1 had a higher precision the F1 score was overall higher when the training set remained unaltered.

Table 9: Precision, recall, and f1 scores for the five models with the highest precision. These ratios are sorted by highest f1 score. The entries in blue are those that met the precision threshold.

Ratio	Precision	Recall	F1 Score
9:1	0.75	0.06	0.12
None (~10:1)	0.72	0.07	0.13
8:1	0.65	0.08	0.15
7:1	0.53	0.10	0.17
6:1	0.44	0.13	0.20

7.2 Hyperparameter Tuning

Since the most time was spent on finding the ideal distribution for the testing set, hyperparameter tuning was kept simple. Using cross-validation, the ideal max depth and amount of estimators were selected from a range of values. For max depth this range was composed of all integers between 10 and 25, for the amount of estimators the options were 50, 100, and 150. These parameters were then selected based on the f1 score to help increase the recall without repercussions on the precision. In all, hyperparameter tuning determined the best max depth to be 20 and the best amount of estimators to be 50.

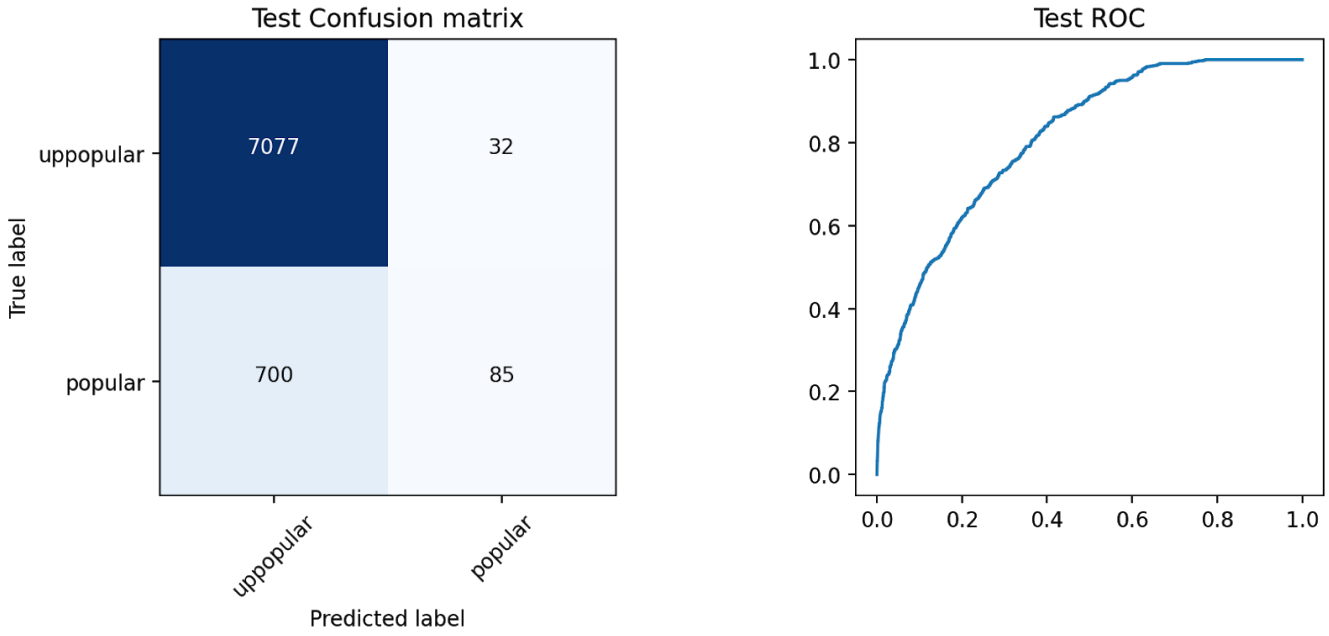
7.3 Optimized Model

Figure 15 shows the confusion matrix and ROC curve of the optimized model and Table 10 presents the corresponding scores. Not only did the model precision increase, but also the recall, almost doubling that of the unoptimized model.

Tabel 10: Scores of the unoptimized and optimized models set

Model	Precision	Recall	F1 Score	Accuracy
Unoptimized	0.67	0.06	0.12	90%
Optimized	0.73	0.11	0.19	91%

Figure 15: Confusion matrix and ROC of the final, optimized model



While optimization increased all of the scores, the recall is still quite low. The amount of false negatives imply that the model can only identify a certain group of popular articles. This suggests that while some popular articles stand out amongst the majority, there are many overlapping similarities between the two labels that challenge the models ability to accurately classify them.

7.4 Final Model

In creating the final model, all points were used in the training set. Using the same optimization methods as above, it was determined that no limit was necessary for the training data and that the best hyperparameters were a max depth of 20 and a total of 50 estimators.

Figure 16 shows the training set confusion matrix and ROC curve. Overall, this model had a precision of 1.0, recall of 0.78, F1 score of 0.87, and accuracy of 98%.

Figure 16: Training confusion matrix and ROC curve of the final model.

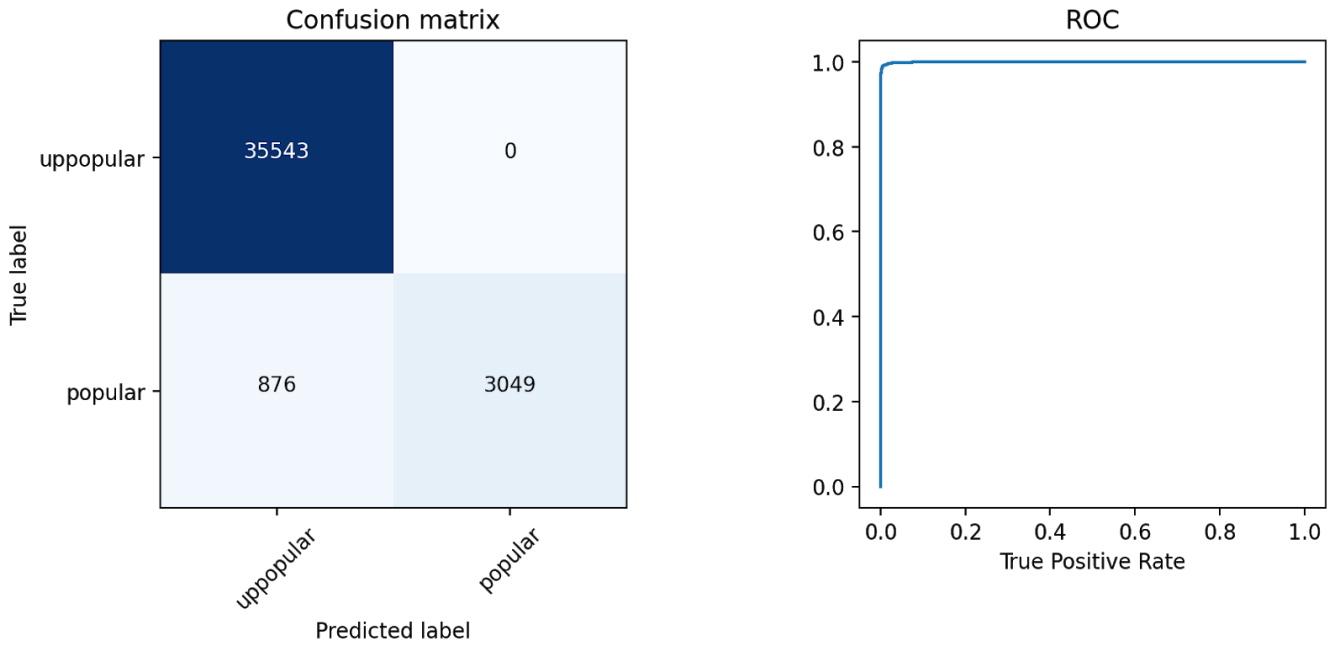
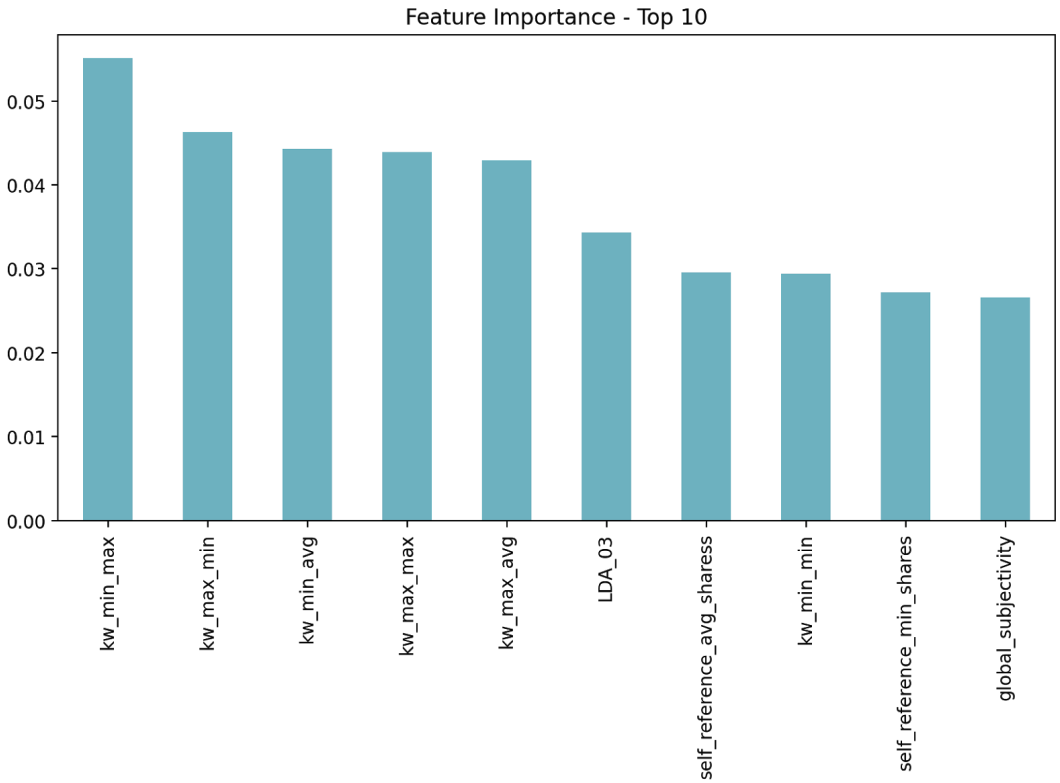


Figure 17 shows a bar graph of the top ten most important features according to the model. From this graph, it is evident that the keyword features have the most weight in the model's classification process, as well as the shares of other Mashable articles referenced in the text. This shows that article popularity is very much tied to the popularity of the articles associated with it, whether directly referenced or within the same topic.

Figure 17: Bar graph of the top 10 most important features according to the model.



8 Conclusions

Mashable's article shares range between a minimum of 0 shares and a maximum of 843,300 shares with a mean and median of 3,392 and 1,400 shares respectively. Throughout analysis popular articles were classified as those with shares above the 90% percentiles (above 6200 shares) and viral articles were those over 190,000 shares.

Initial analysis showed that, while these articles had a wide spread of features, a majority tended to be subjective and positive in polarity. Looking over the time scale, it appears that a high frequency of viral articles boosts the median shares of other articles published within the same time frame and that Mashable fairs better during the winter season. This could indicate that viral articles bring attention to Mashable as a whole, thus raising the viewership and shares of other articles. In terms of keywords, it appeared that the worst performing was *sports* and the best performing were *gadgets* and *viral videos*. Lastly, focusing on viral articles, these articles were all similar in that they had high keyword performances, with a lot of positive text.

In terms of modeling, clustering categorized articles by the polarity, length, and complexity of the text. While a majority of articles fell into the short, yet more articulate, and positive text category, almost all viral articles were within the simple and positive text category. This further suggests that positivity and longer articles appear to be popular among readers and makes for good shareability.

Of classification models, binary classification was best at determining the popularity of models. After optimization, the final model had a training precision of 1.0, recall of 0.78, F1 score of 0.87, and accuracy of 98% . According to this model's feature importance, the keyword features as well as the shares of referenced articles held the most significance in discerning the popularity of articles. This points to the importance of the interconnectedness articles, popular articles bring attention to the articles affiliated with it through references and keywords.

Overall, it appears that Mashable articles that are positive and more wordy tend to have higher shares. There is also a great importance on how the articles are connected to each other, suggesting that Mashable should put more emphasis on how they cluster their articles. This could possibly include devoting resources towards a more advanced recommendation engine to better enhance this flow of share from article to article or creating a more concrete system of keywords.

Appendix

I. Tables

Table I: Description of dataset. For all int *columns, a 0 represents 'no' while a 1 represents 'yes'.*

No.	Variable Name	Variable Description	Unique Values	Dtype
0	url	URL of articles	39644	object
1	timedelta	Days between the article publication and the dataset acquisition	724	float
2	n_tokens_title	Number of words in title	20	float
3	n_tokens_content	Number of word in content	2406	float
4	n_unique_tokens	Rate of unique word in the article	27281	float
5	n_non_stop_words	Rate of non-stop words in the content	1451	float
6	n_non_stop_unique_tokens	Rate of unique non-stop words in the content	22930	float
7	num_hrefs	Number of links	133	float
8	num_self_hrefs	Number of links to other articles published by Mashable	59	float
9	num_imgs	Number of images	91	float
10	num_videos	Number of videos	53	float
11	average_token_length	Average length of the words in the content	30136	float
12	num_keywords	Number of keywords in article metadata	10	float
13	data_channel_is_lifestyle	Is data channel labeled as Lifestyle?	2	int
14	data_channel_is_entertainment	Is data channel labeled as Entertainment?	2	int
15	data_channel_is_bus	Is data channel labeled as Business?	2	int
16	data_channel_is_socmed	Is data channel labeled as Social Media?	2	int
17	data_channel_is_tech	Is data channel labeled as Tech?	2	int
18	data_channel_is_world	Is data channel labeled as World?	2	int
19	kw_min_min	Min shares of worst keyword	26	float
20	kw_max_min	Avg shares of worst keyword	1076	float
21	kw_avg_min	Max shares of worst keyword	17003	float
22	kw_min_max	Min shares of best keyword	1021	float
23	kw_max_max	Avg shares of best keyword	35	float

24	kw_avg_max	Max shares of best keyword	30834	float
25	kw_min_avg	Min shares of average keyword	15982	float
26	kw_max_avg	Avg shares of average keyword	19438	float
27	kw_avg_avg	Max shares of average keyword	39300	float
28	self_reference_min_shares	Min shares of referenced articles in Mashable	1255	float
29	self_reference_max_shares	Max shares of referenced articles in Mashable	1137	float
30	self_reference_avg_sharess	Avg shares of referenced articles in Mashable	8626	float
31	weekday_is_monday	Was the article published on a Monday?	2	int
32	weekday_is_tuesday	Was the article published on a Tuesday?	2	int
33	weekday_is_wednesday	Was the article published on a Wednesday?	2	int
34	weekday_is_thursday	Was the article published on a Thursday?	2	int
35	weekday_is_friday	Was the article published on a Friday?	2	int
36	weekday_is_saturday	Was the article published on a Saturday?	2	int
37	weekday_is_sunday	Was the article published on a Sunday?	2	int
38	is_weekend	Was the article published on a Weekend?	2	int
39	LDA_00	Closeness to LDA topic 0	39337	float
40	LDA_01	Closeness to LDA topic 1	39098	float
41	LDA_02	Closeness to LDA topic 2	39525	float
42	LDA_03	Closeness to LDA topic 3	38963	float
43	LDA_04	Closeness to LDA topic 4	39370	float
44	global_subjectivity	Text subjectivity	34501	float
45	global_sentiment_polarity	Text sentiment polarity	34695	float
46	global_rate_positive_words	Rate of positive words in the content	13159	float
47	global_rate_negative_words	Rate of negative words in the content	10271	float
48	rate_positive_words	Rate of positive words among non-neutral tokens	2284	float
49	rate_negative_words	Rate of negative words among non-neutral tokens	2284	float
50	avg_positive_polarity	Avg polarity of positive words	27301	float
51	min_positive_polarity	Min polarity of positive words	33	float

52	max_positive_polarity	Max polarity of positive words	38	float
53	avg_negative_polarity	Avg polarity of negative words	13841	float
54	min_negative_polarity	Min polarity of negative words	54	float
55	max_negative_polarity	Max polarity of negative words	49	float
56	title_subjectivity	Title subjectivity	673	float
57	title_sentiment_polarity	Title polarity	813	float
58	abs_title_subjectivity	Absolute subjectivity level	532	float
59	abs_title_sentiment_polarity	Absolute polarity level	653	float
60	shares	Amount of shares (at time of dataset acquisition)	1454	int

Table II: Description of dataset after cleaning

No.	Variable Name	Variable Description	Unique Values	Dtype
0	url	URL of articles	39644	object
1	timedelta	Days between the article publication and the dataset acquisition	724	float
2	n_tokens_title	Number of words in title	20	float
3	n_tokens_content	Number of word in content	2404	float
4	n_unique_tokens	Rate of unique word in the article	27197	float
5	n_non_stop_words	Rate of non-stop words in the content	1450	float
6	n_non_stop_unique_tokens	Rate of unique non-stop words in the content	22862	float
7	num_hrefs	Number of links	132	float
8	num_self_hrefs	Number of links to other articles published by Mashable	59	float
9	num_imgs	Number of images	91	float
10	num_videos	Number of videos	53	float
11	average_token_length	Average length of the words in the content	30025	float
12	num_keywords	Number of keywords in article metadata	10	float
13	self_reference_min_shares	Min shares of referenced articles in Mashable	1255	float
14	self_reference_max_shares	Max shares of referenced articles in Mashable	1137	float
15	self_reference_avg_sharess	Avg shares of referenced articles in Mashable	8599	float
16	LDA_00	Closeness to LDA topic 0	39165	float
17	LDA_01	Closeness to LDA topic 1	38928	float
18	LDA_02	Closeness to LDA topic 2	39350	float
19	LDA_03	Closeness to LDA topic 3	38794	float
20	LDA_04	Closeness to LDA topic 4	39197	float
21	global_subjectivity	Text subjectivity	34349	float
22	global_sentiment_polarity	Text sentiment polarity	34548	float
23	global_rate_positive_words	Rate of positive words in the content	13121	float
24	global_rate_negative_words	Rate of negative words in the content	10250	float
25	rate_positive_words	Rate of positive words among non-neutral tokens	2280	float

26	rate_negative_words	Rate of negative words among non-neutral tokens	2280	float
27	avg_positive_polarity	Avg polarity of positive words	27188	float
28	min_positive_polarity	Min polarity of positive words	33	float
29	max_positive_polarity	Max polarity of positive words	38	float
30	avg_negative_polarity	Avg polarity of negative words	13795	float
31	min_negative_polarity	Min polarity of negative words	54	float
32	max_negative_polarity	Max polarity of negative words	49	float
33	title_subjectivity	Title subjectivity	672	float
34	title_sentiment_polarity	Title polarity	808	float
35	abs_title_subjectivity	Absolute subjectivity level	532	float
36	abs_title_sentiment_polarity	Absolute polarity level	650	float
37	shares	Amount of shares (at time of dataset acquisition)	1452	int
38	channel	Data channel of article	6	object
39	date	Publish Date	737	datetime
40	title	Article title	39229	object
41	weekday	Weekday article was published	6	object
42	kw_min	Worst performing keyword	5324	object
43	kw_min_min	Min shares of worst keyword	825	int
44	kw_min_avg	Avg shares of worst keyword	2183	int
45	kw_min_max	Max shares of worst keyword	556	int
46	kw_avg	Median keyword	1559	object
47	kw_avg_min	Min shares of average keyword	637	int
48	kw_avg_avg	Avg shares of average keyword	1168	int
49	kw_avg_max	Max shares of average keyword	468	int
50	kw_max	Best performing keyword	2098	object
51	kw_max_min	Min shares of best keyword	701	int
52	kw_max_avg	Avg shares of best keyword	1547	int
53	kw_max_max	Max shares of best keyword	505	int

Table III: Summary statistics of numerical columns

Column Names	Mean	STD	Minimum	Maximum
timedelta	355.609	213.930	8.000	731.000
n_tokens_title	10.396	2.113	2.000	23.000
n_tokens_content	546.398	471.047	0.000	8474.000
n_unique_tokens	0.548	3.529	0.000	701.000
n_non_stop_words	0.997	5.243	0.000	1042.000
n_non_stop_unique_tokens	0.689	3.272	0.000	650.000
num_hrefs	10.883	11.309	0.000	304.000
num_self_hrefs	3.297	3.860	0.000	116.000
num_imgs	4.547	8.317	0.000	128.000
num_videos	1.253	4.116	0.000	91.000
average_token_length	4.548	0.844	0.000	8.042
num_keywords	7.225	1.909	1.000	10.000
self_reference_min_shares	3975.765	19467.546	0.000	843300.000
self_reference_max_shares	10304.821	40836.816	0.000	843300.000
self_reference_avg_sharess	6375.495	23953.428	0.000	843300.000
LDA_00	0.185	0.263	0.000	0.927
LDA_01	0.141	0.220	0.000	0.926
LDA_02	0.216	0.282	0.000	0.920
LDA_03	0.224	0.295	0.000	0.927
LDA_04	0.234	0.289	0.000	0.927
global_subjectivity	0.443	0.117	0.000	1.000
global_sentiment_polarity	0.119	0.097	-0.394	0.728
global_rate_positive_words	0.040	0.017	0.000	0.155
global_rate_negative_words	0.017	0.011	0.000	0.185
rate_positive_words	0.682	0.190	0.000	1.000

rate_negative_words	0.288	0.156	0.000	1.000
avg_positive_polarity	0.354	0.105	0.000	1.000
min_positive_polarity	0.095	0.071	0.000	1.000
max_positive_polarity	0.757	0.248	0.000	1.000
avg_negative_polarity	-0.259	0.128	-1.000	0.000
min_negative_polarity	-0.522	0.290	-1.000	0.000
max_negative_polarity	-0.107	0.095	-1.000	0.000
title_subjectivity	0.282	0.324	0.000	1.000
title_sentiment_polarity	0.071	0.266	-1.000	1.000
abs_title_subjectivity	0.342	0.189	0.000	0.500
abs_title_sentiment_polarity	0.156	0.226	0.000	1.000
shares	3392.228	11623.226	1.000	843300.000
kw_min_min	400.377	361.269	1.000	4500.000
kw_min_avg	2441.377	775.508	406.000	11650.000
kw_min_max	93429.993	177515.965	446.000	843300.000
kw_avg_min	92.863	211.751	1.000	5800.000
kw_avg_avg	3437.082	426.394	1029.000	40771.000
kw_avg_max	398775.956	282780.684	1200.000	843300.000
kw_max_min	285.105	570.033	1.000	15000.000
kw_max_avg	4742.799	4036.744	1334.000	100459.000
kw_max_max	339648.822	297141.065	3800.000	843300.000

II. Figures

Figure I: Flow chart of keyword feature processing.

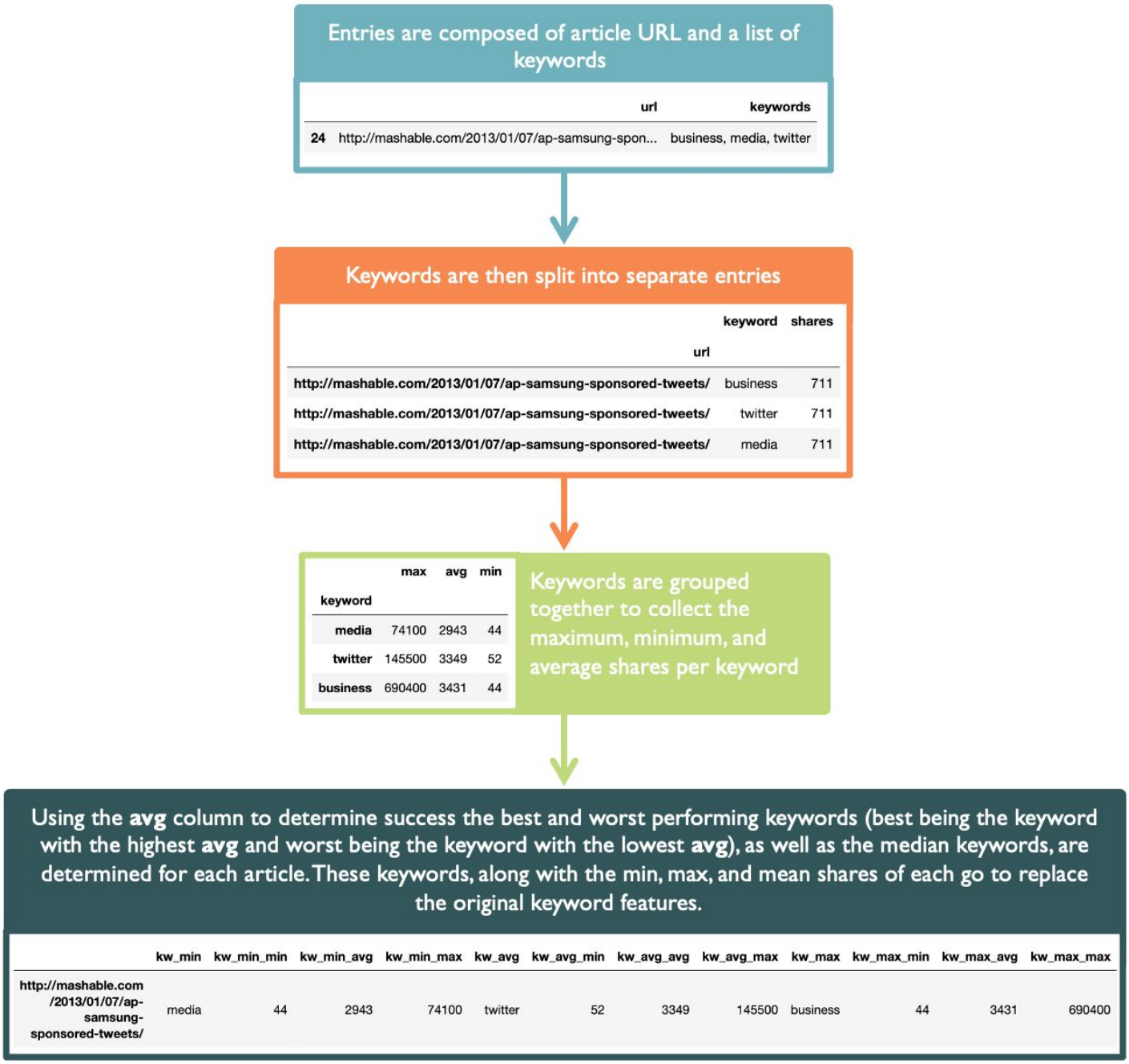


Figure II: Distribution of features

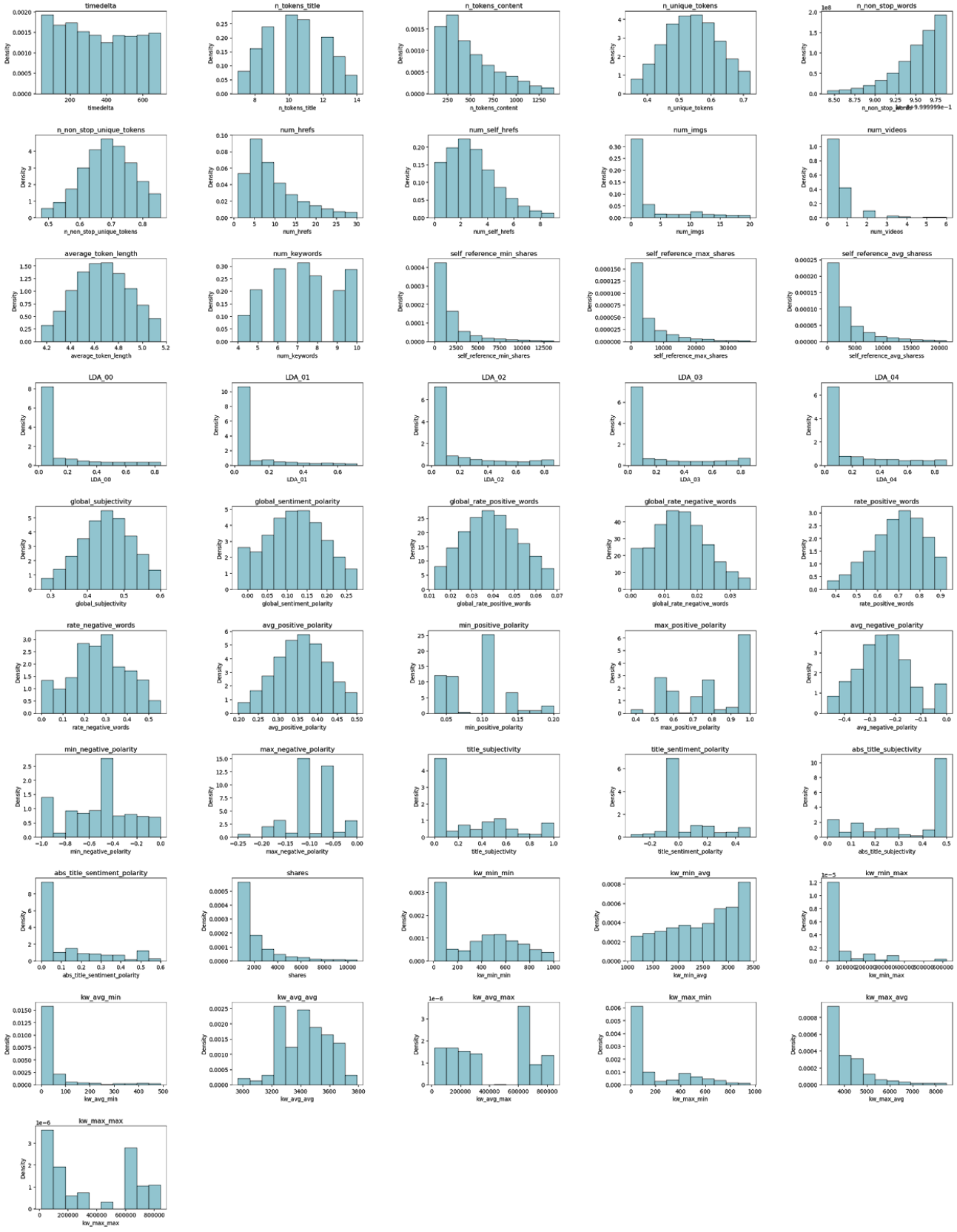


Figure III: Heatmap displaying the correlation viral articles have with shares

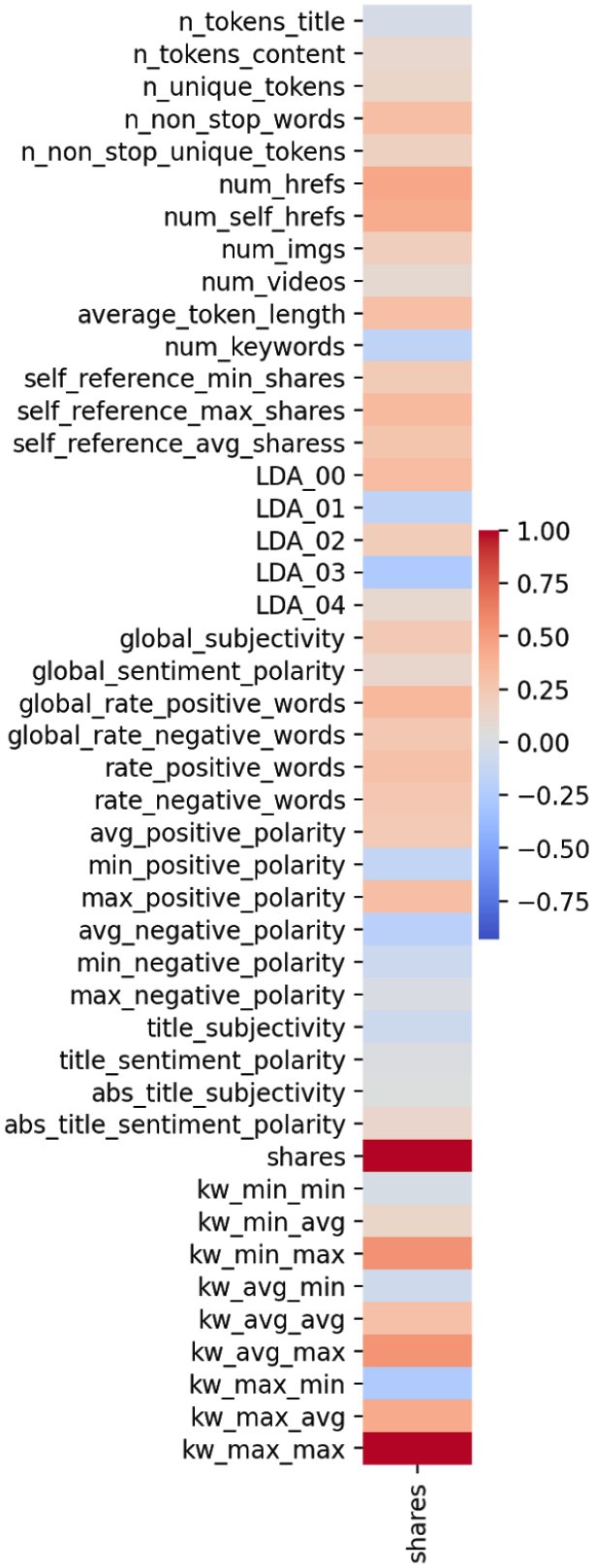


Figure IV: Distributions of each feature separated by cluster