

Analyzing Article Popularity

Capstone II - Final Report

Springboard

Data Science Career Track

Tara Crutchfield

December 28, 2020

Table of Contents

1. [Objective](#)
2. [Dataset](#)
3. [Packages](#)
4. [Data Wrangling](#)
5. [Exploratory Data Analysis](#)
6. [Modeling](#)
7. [Model Optimization](#)
8. [Conclusions](#)

Appendix

- I. [Tables](#)
- II. [Figures](#)

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.

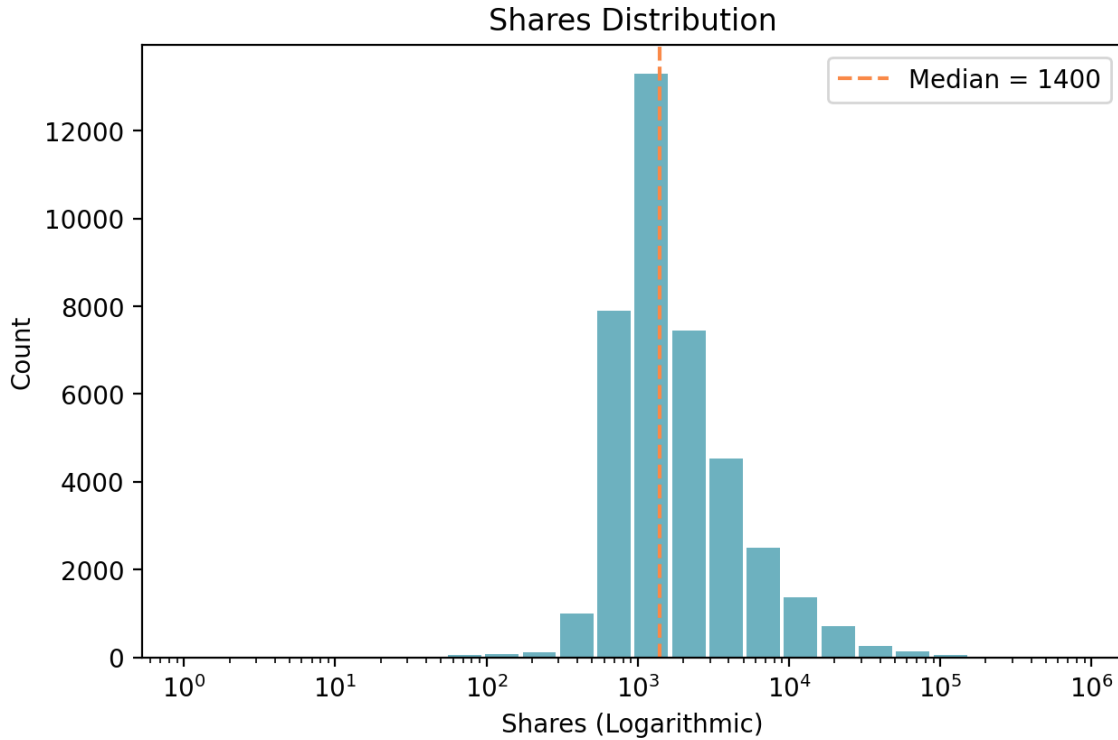
2. Dataset

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.

This dataset did not contain a feature for popularity, but rather **shares** which represented the amount of times an article was shared by users on social media. Figure 1 shows the logarithmic distribution of the **shares** column. These shares ranged from a minimum of 1 and a maximum of 843,300 with a median of 1400 shares.

While shares are indicative of popularity, a threshold had to be determined in order to properly distinguish popular articles from the rest. For this project, an article was labeled *Popular* if the shares were above the 90th percentile (6200 shares). With this definition, the initial dataset had a total of 3,946 *Popular* articles, an example of which can be seen in [Figure I](#).

Figure 1: Histogram of shares with logarithmic scale.



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, BeautifulSoup 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 1, included the data channel, publish date, article title, and all keywords for each article and were saved in the file 'Updates.csv'. Snippets of the code where these features were found can be seen in [Figure II](#).

Table 1: Description of added features

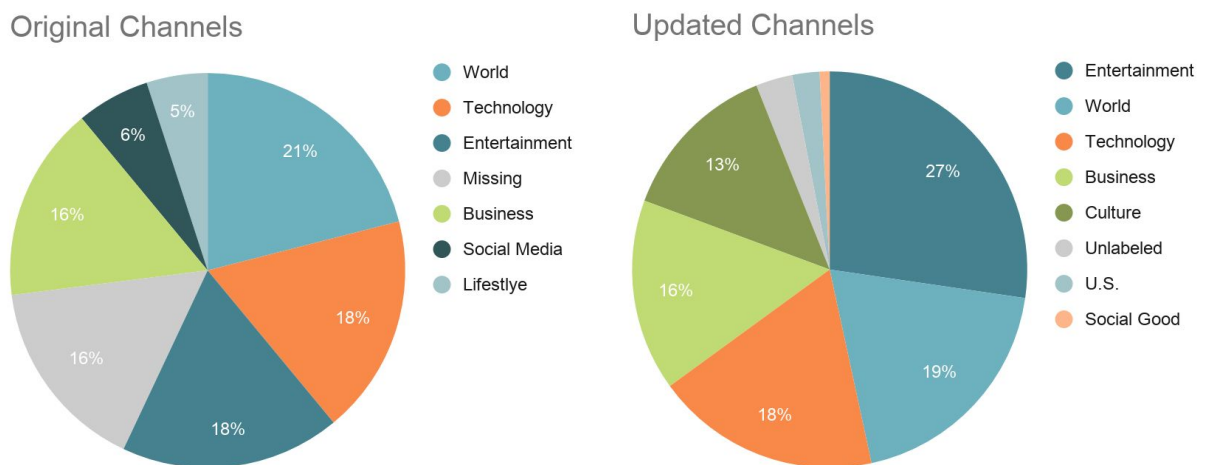
No.	Variable Name	Variable Description	Unique Values	Dtype	Non-Null
0	url	URL of articles	39644	object	100.0%
1	channel	Article data channel label	8	object	99.6%
2	date	Article publish date	738	object	99.7%
3	title	Article title	39272	object	99.1%
4	keyword	All article keywords	34545	object	99.6%

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 new data 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.

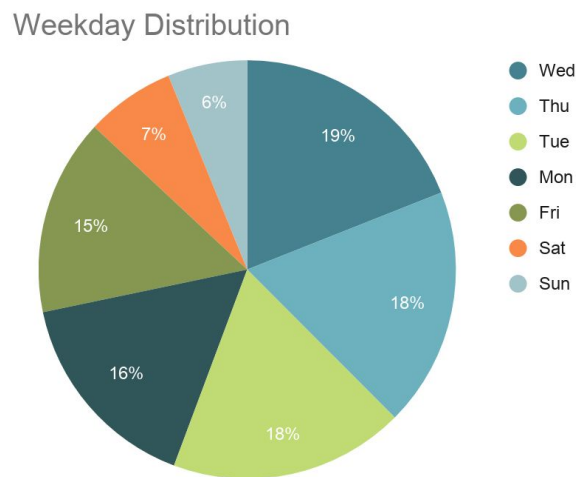
4.2 Days of the Week

The **date** column of 'Updates.csv' included both the date and day of the week each article was published. This column was split into two separate columns: **date**, consisting of the date as a datetime object, and **weekday**, containing the publish day.

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 newer **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 III](#) 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 IV](#) shows the distribution of these values. In order to adjust the plots for outliers, if the maximum was larger than three standard deviations from the mean, the range of the graph was adjusted to only contain points within the 90th percentile.

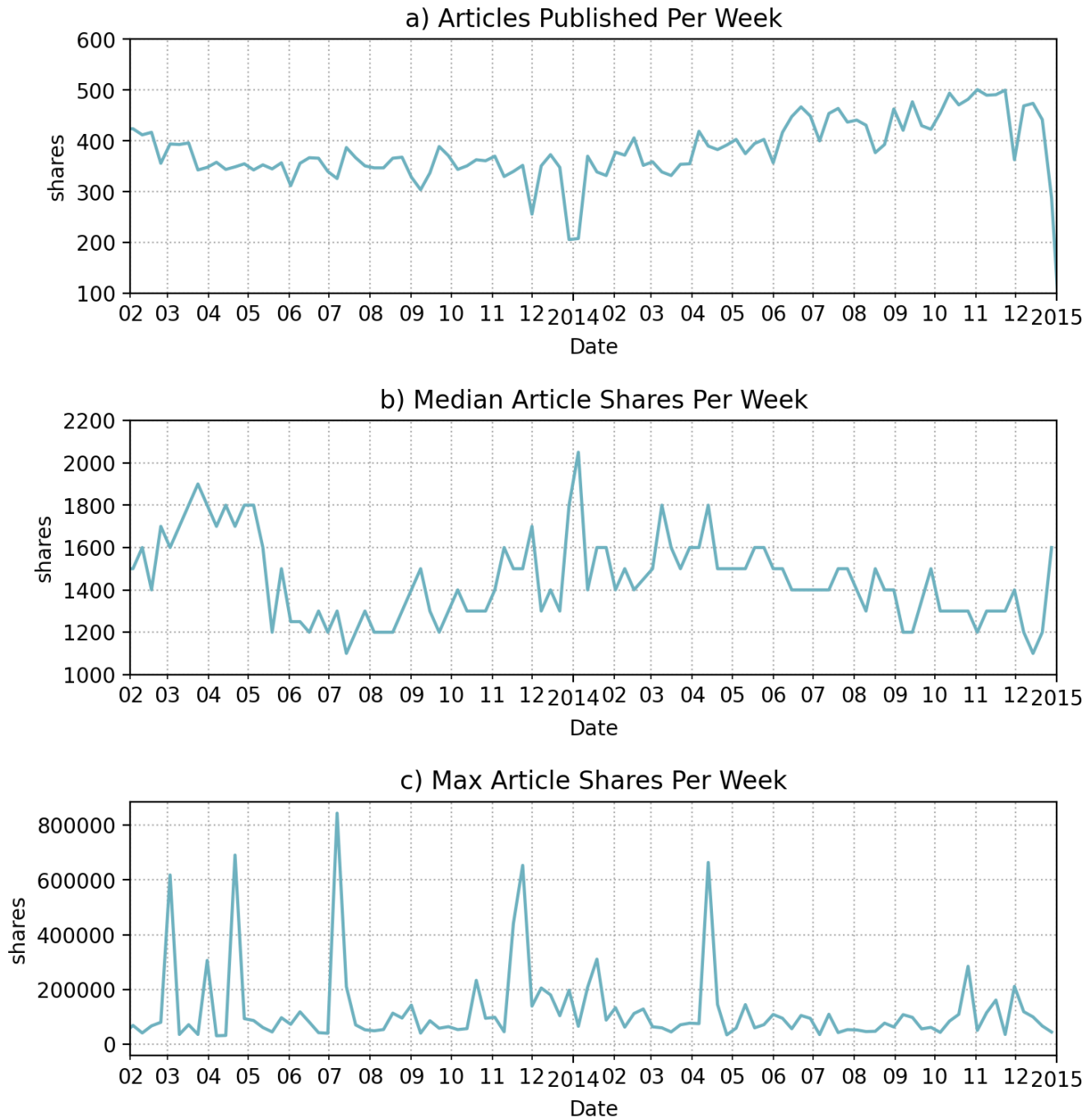
A majority of these distributions appeared normal or one-tailed. The only features that differed from this pattern were some of the keyword features as well as the min/max polarity features. Overall, it appeared that a majority of Mashable articles tend to be on the positive side and slightly more subjective.

5.2 Time

Using the date column, it was possible to plot out shares over time. Figure 4a 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.

In Figure 4b, 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 4: Three separate graphs displaying a) the amount of articles published each week, b) the median shares of articles published each week, and c) the shares of the most popular article per week.

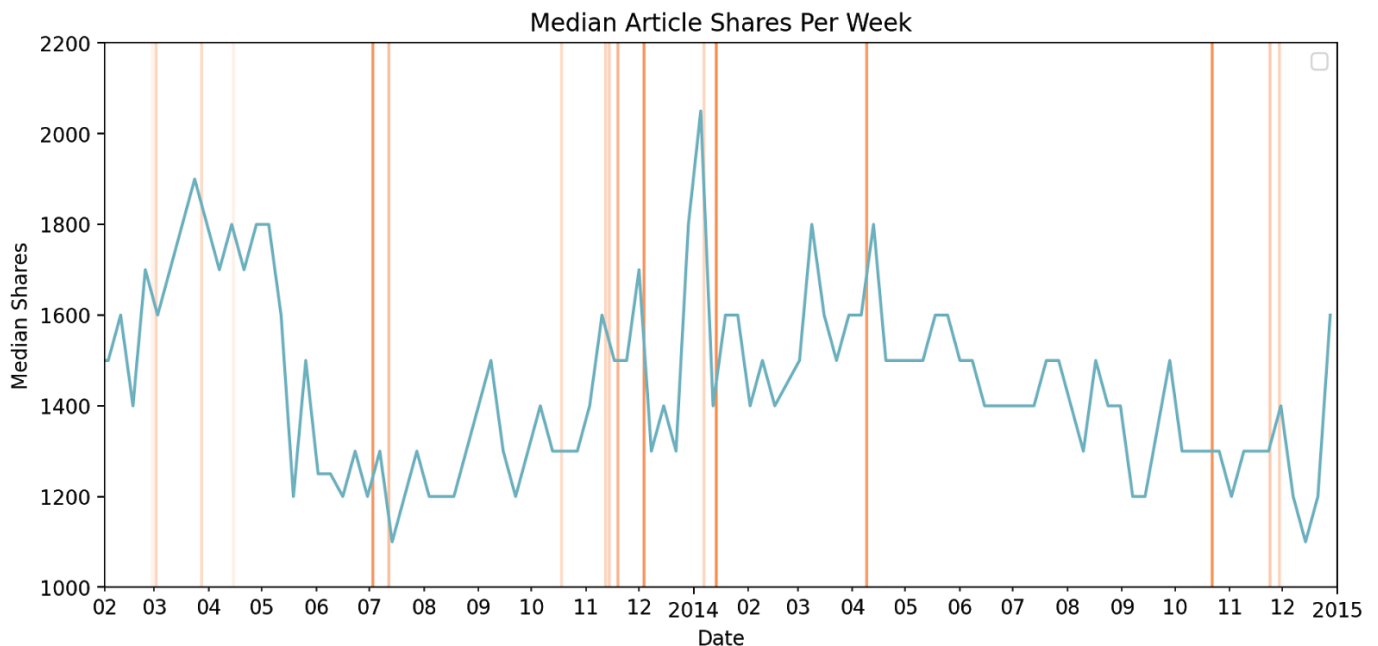


Lastly, the maximum shares of articles published in a week are plotted out in Figure 4c. All of the articles that make up the points in this graph are *Popular*, yet a select few cause some large spikes in the pattern. This subset of *Popular* articles are defined as *Viral* as they manage to capture a significantly greater audience than the typical *Popular* article. While there are not enough *Viral*

articles present in the dataset to study in detail, they do appear to make an impact on the shares of other articles.

This can be seen when the publish dates of *Viral* articles are superimposed on the median weekly shares, seen in Figure 5. Comparing the placement of *Viral* articles with the shape of the weekly median, it appears that higher concentrations of *Viral* articles give a boost to the overall median shares. Paying attention to the value of the *Viral* articles, represented by the opacity of these lines, it also seems that *Viral* articles with higher shares tend to be followed by small dips in the median. This may indicate that while a lot of *Viral* articles may boost the weekly median, they could also possibly direct attention away from articles published in the weeks after it.

Figure 5: 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.



5.3 Keywords

Altogether there were 16,724 different 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 2 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

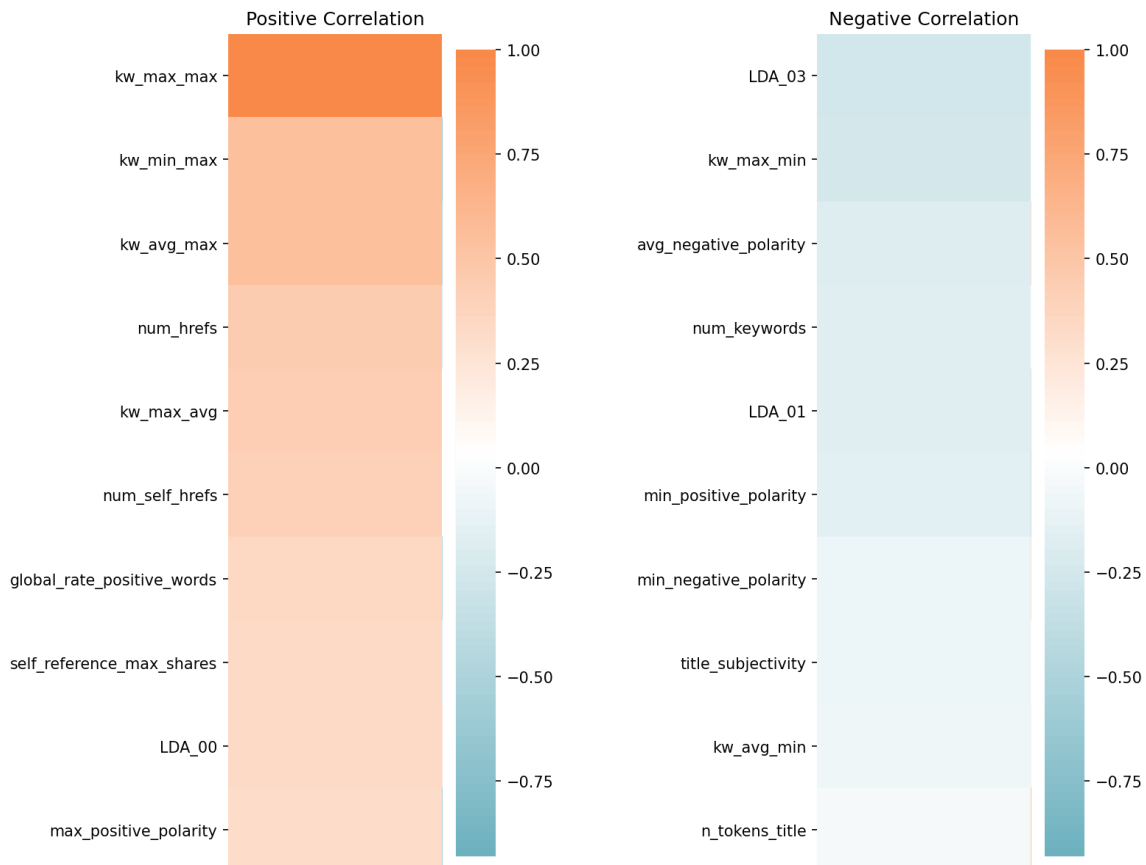
Table 2: Most common keywords found in *kw_min*, *kw_avg*, and *kw_max*. These entries are in order of occurrences, with the highest count on the top.

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

5.4 Correlation

A correlation matrix was created using all the *Popular* articles in order to find what features appeared to be related to shares. No strong correlations were found, so this process was repeated, focusing on the *Viral* subset of the *Popular* articles. This resulted in dramatically clearer correlations, which can be seen in the heatmap provided in Figure 6.

Figure 6: Heatmap of the shares column of the correlation matrix.



Positive correlation, represented by the orange hues, can be seen with the maximum keyword features, the amount of references, and positive sentiment. Focusing on the keywords, it appears that an article's shares are more likely to be higher if they are related to well performing keywords.

The values for negative correlation, represented by the blue hues, were not as extreme. Nonetheless, features that stand out are negative sentiment as well as long, subjective titles. Interestingly it also appears that the shares are negatively correlated with the amount of keywords and the shares of the least shared article in the best performing keyword.

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 further analyze the relationships between these articles and their features, K-means clustering was utilized in order to find trends in the data that would either be too time consuming or too small to distinguish by eye. This process involved a machine comparing the data points such that it would group them into a given amount of clusters. While the clusters themselves are quite telling of the unobvious similarities between articles, comparing the clusters and their relationship with article shares could help distinguish whether there is an underlying trend that affects the popularity of articles. In order to prevent the clusters from correlating with shares, the **shares** column was removed before modeling.

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 7. Through this method it was determined that four clusters would best encompass the variance of the dataset.

When clustering, it became apparent that two articles in particular always got their own clusters. Upon further inspection it appeared that these two were both incomplete entries with each column value set to zero. These two entries were discarded, leaving a total of 39466 entries.

Figure 7: Example of the ‘elbow method’. The orange line marks the inertia of the model with 4 clusters. A model with any more clusters would be overfitting to the dataset.

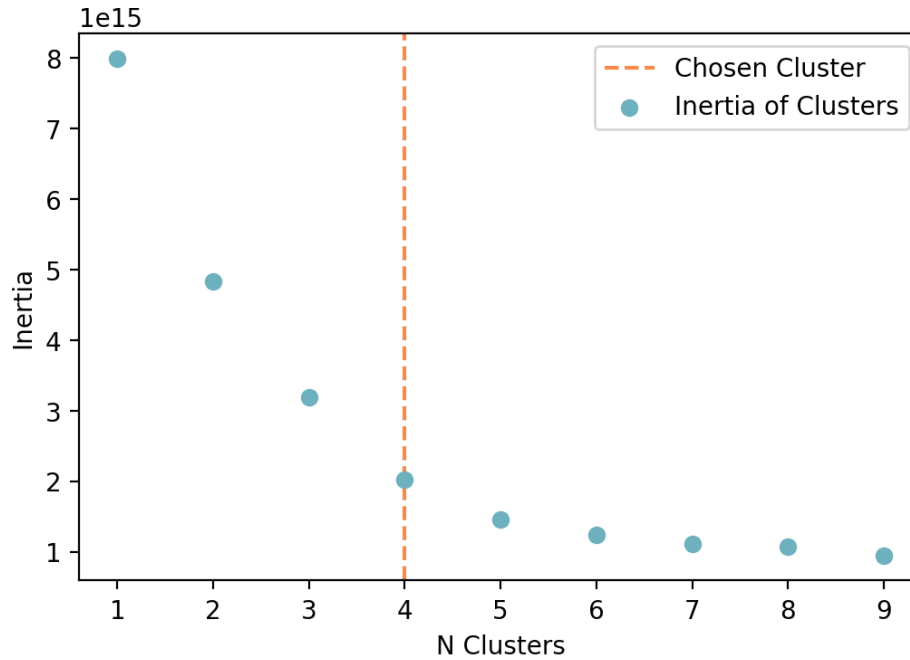


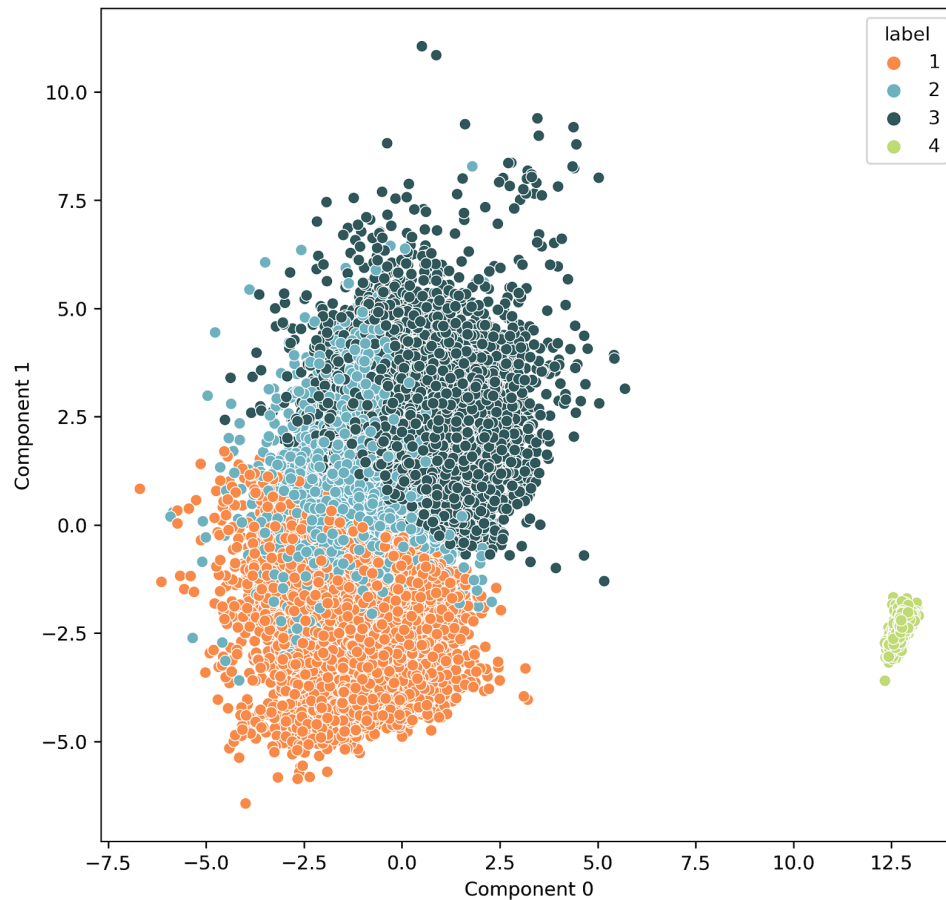
Table 3 shows the results of the clustering as well as a brief summary of the defining features based on the distributions which can be seen in [Figure V](#) in the appendix. While there was a lot of overlap between the clusters, it appeared the main differentiating factors were the text features and polarity. Figure 8 gives an example visual of these clusters using the top two components found using principal component analysis. Here it is apparent that Cluster 1, Cluster 2, and Cluster 3 all appear to overlap yet Cluster 4 is distinctly separate.

While all the clusters contained some *Popular* articles, Cluster 2 had the most overall, followed closely by Cluster 1. This may imply that *Popular* articles are generally on the more positive side, and that Mashable may want to focus on creating articles with more words and references.

Table 3: Clusters descriptions including the amount of entries in each cluster

Cluster	No. Entries	No. Pop Entries	Description
Cluster 3	13071	974	Negative sentiment, less words
Cluster 1	15092	1248	Positive sentiment, less words
Cluster 4	1174	192	Articles with little to no text
Cluster 2	10129	1510	More words, more references, positive sentiment

Figure 8: Plot of the two top PCA components with the clusters colored separately.



6.2 Supervised Learning

Supervised learning was utilized to determine whether the popularity of an article could be modeled by its given features alone. Depending on the success of these models, more information as to determining the qualities that add to the popularity of an article could be found when looking at the model's feature importance. In the best case scenario, this model could even be used to predict whether a new article would be popular before publishing.

6.2.1 Predicting Shares

Before beginning with classification, I was curious as to whether a model could accurately predict the value of article shares. When testing different models, it quickly became apparent that the dataset was not fit for this. Even when rounding shares to the closest thousand, both linear regression and random forest models fared equally poorly.

6.2.2 Popularity - Multi-Classification

The first classification model was made with the goal of classifying articles into three separate labels. These labels included the already determined *Popular* label which was made up of articles with shares above the 90th percentile (6200 shares). The remaining articles were split into the labels *Average* or *Unpopular*: *Average* if greater than or equal to the 30th percentile (1000 shares), *Unpopular* if the shares were below this threshold. The data was then split in a stratified manner such that there was a training set containing 80% of the data points and a testing set composed of the remaining 20%. Table 4 shows the distribution of these sets.

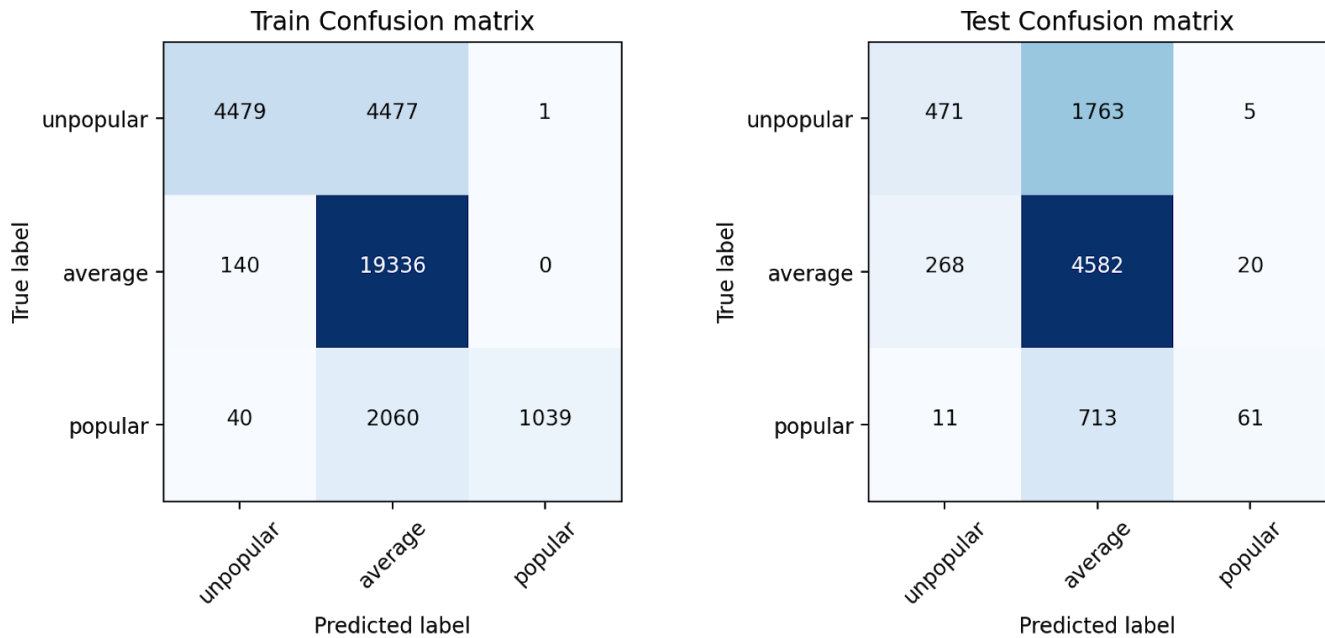
Table 4: Distribution of training and testing sets from modeling

Set	Unpopular	Average	Popular	Total Entries
Range	<i>(0, 1000)</i>	<i>[1000, 6200]</i>	<i>(6200, 843300]</i>	<i>(0, 843300]</i>
Train	8957	19476	3139	31572
Test	2239	4870	785	7894
Total	11196	24346	3924	39466

Five fold cross-validation was performed utilizing model accuracy to determine a best max depth of 14 for the model. The prediction results of this model can be visualized as a confusion matrix as seen in Figure 9.

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 65% with an average recall and precision of 0.41 and 0.69. Focusing particularly on the *Popular* label, the recall was as low as 0.08 despite a precision of 0.80.

Figure 9: Confusion matrix for both the training and test sets. Darker colors represent a higher density of points.



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 5.

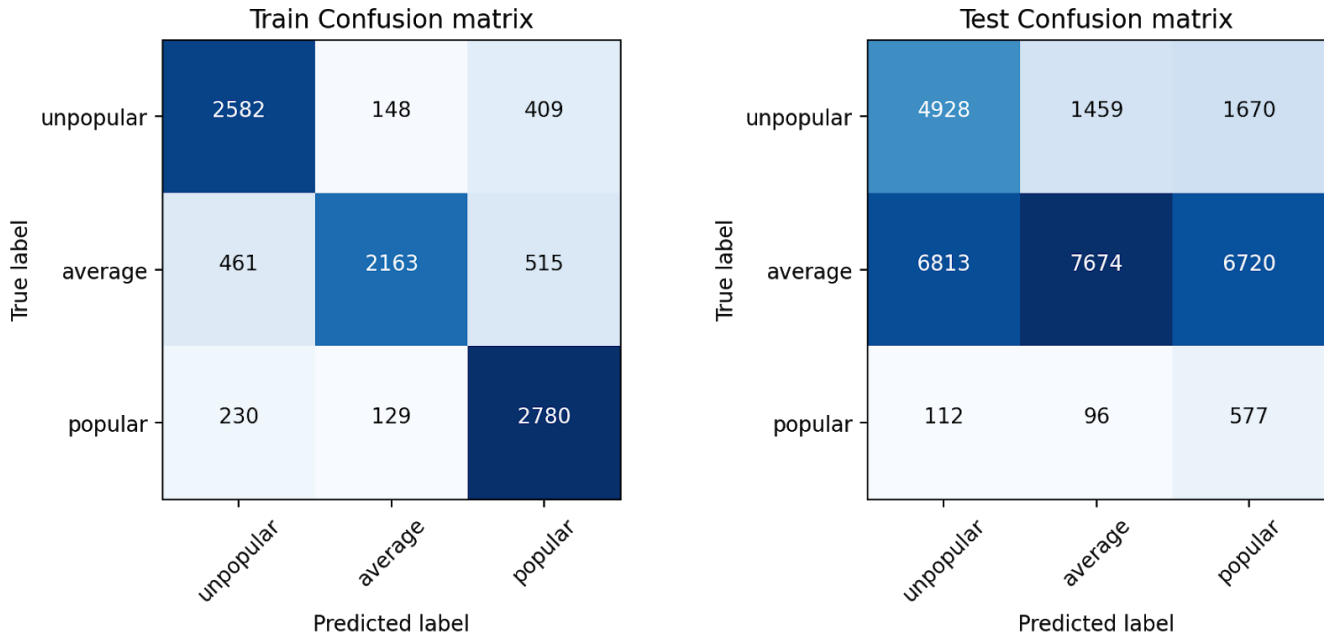
Table 5: Distribution of entries with a limited training set.

Set	Unpopular	Average	Popular	Total Entries
Train	3139	3139	3139	9417
Test	8057	21207	785	30049

Using this training set, cross-validation was performed again to determine a best max depth of 10. Figure 10 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 45%. While the average recall did increase to 0.57, it came with a large hit to the precision, averaging out to 0.43.

Isolating the *Popularity* label, the results were the opposite of the first model. In this case the recall increased to 0.74 while the precision decreased 0.06. While an increase in recall is desired, the fall in precision is far too large to warrant such an exchange.

Figure 10: 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%



6.2.3 Popularity - Binary Classification

As it appeared that the models had a trade off between precision and recall, the multiclass model was shifted in favor of a binary classification model to better monitor these qualities. For this model, the threshold dividing popularity remained at the 90th percentile (6200 shares). Articles with shares above the 6200 shares were labeled as *Popular* and articles with shares equal to or below were designated *Unpopular*. 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 6.

Table 6: 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
Range	[0, 6200]	[6200, 843300]	[0, 843300]
Train	28433	3139	31572
Test	7109	785	7994
Total	35542	3924	39466

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

Set	Unpopular	Popular	Total Entries
Train	3139	3139	6278
Test	32403	785	33188

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 test accuracy of 90% and f1 score of 0.15. Figure 11 presents both the confusion matrix and ROC curves for this model. In all, the model had a precision of 0.80 and recall of 0.09, both of which were higher than the scores of the *Popular* label from the multiclass model.

Figure 11: Confusion matrix and ROC curve for the model trained on the normal training set.

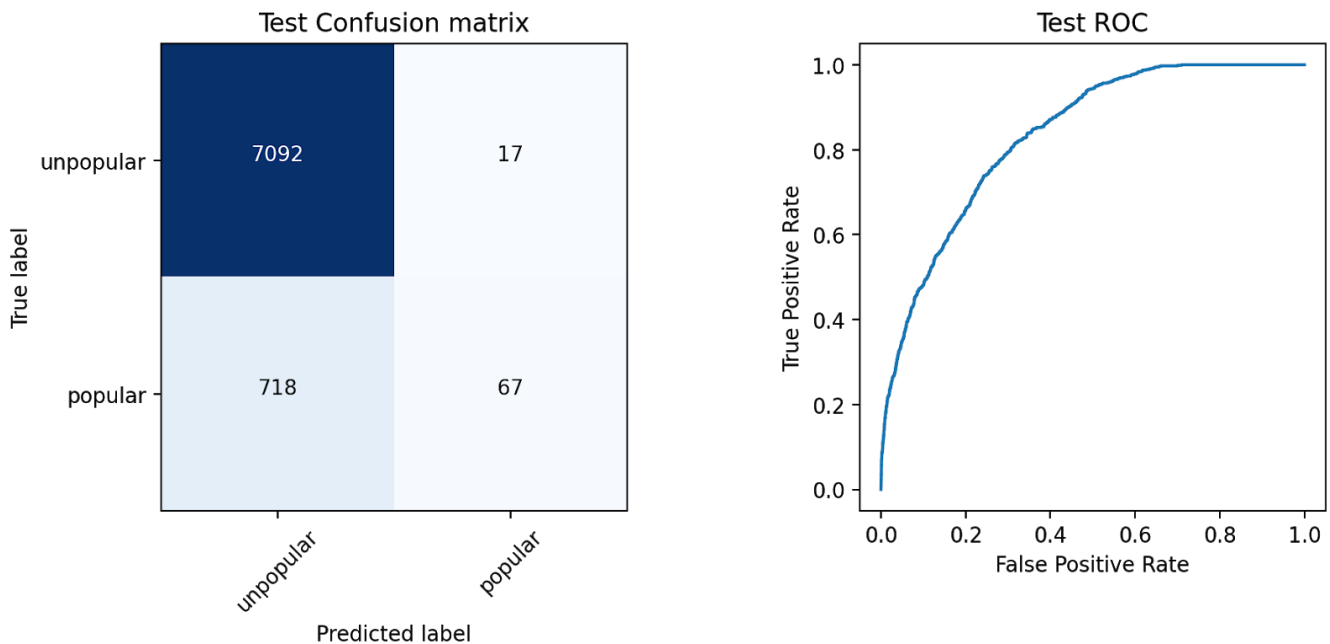
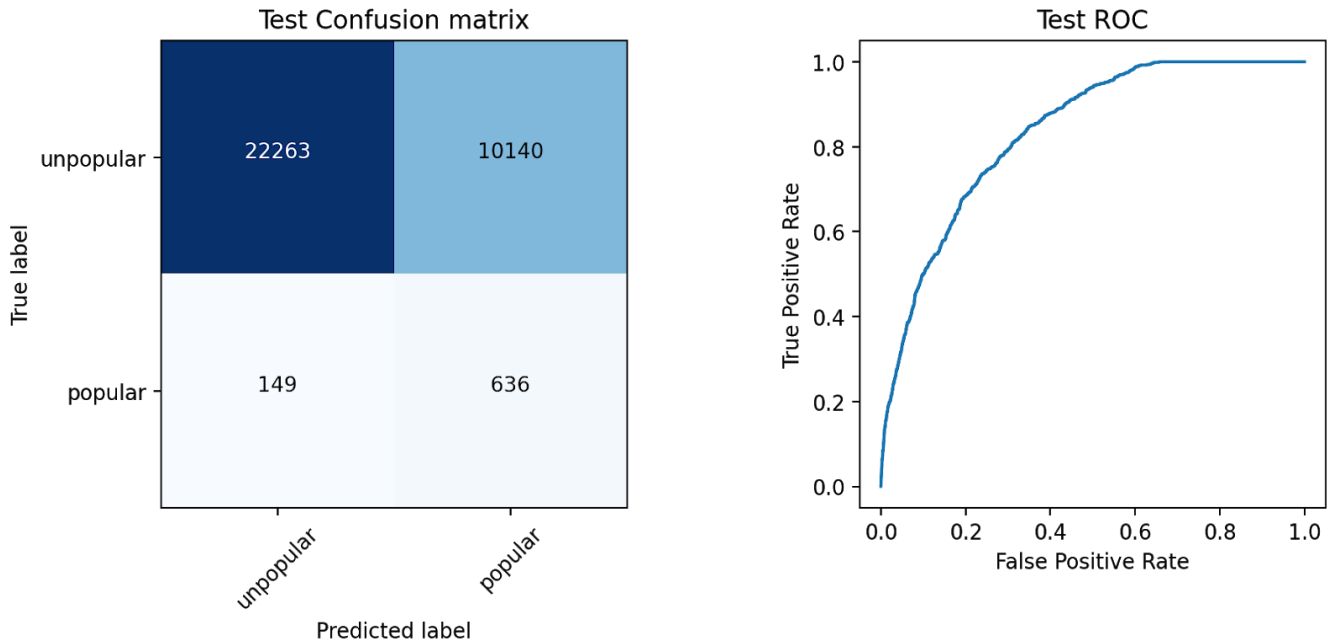


Figure 12 shows the confusion matrix and ROC curve of the model trained on the limited training set. This model had an accuracy of 68% and f1 score of 0.11. The recall and precision of this model were almost exactly the opposite as those of the earlier model: precision dropping to 0.06 and recall increasing to 0.81.

Figure 12: 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 7. Over all, while the binary classification models have the same precision in the *Popular* label as their multiclassification counterparts, the recall does appear to be slightly higher.

Table 7: Summary of the test scores for all four models. Here 'MC' means the multiclass models while 'BC' represents the binary classification models.

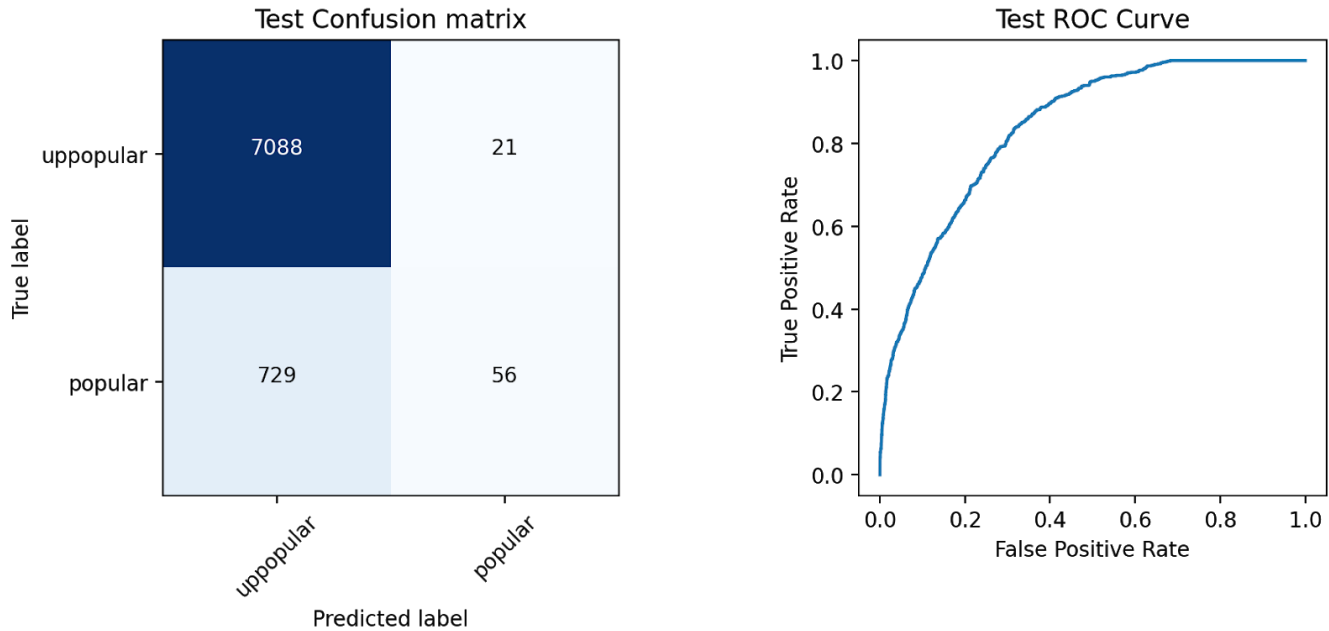
Model	Accuracy	F1 Score	Precision	Recall
MC - No Limit	65%	0.11	0.80	0.08
MC - Limited	45%	0.10	0.06	0.74
BC - No Limit	90%	0.15	0.80	0.09
BC - Limited	68%	0.11	0.06	0.81

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.

The unoptimized model was built on the same training data as seen in Table 6b. The model had an arbitrarily set max depth of 10 and n-estimators of 100, and predictions were made with a probability threshold of 0.5. Figure 13 shows the results of this model, both as a confusion matrix and ROC curve. Overall, it had an accuracy of 90% and an f1 score of 0.13. It's precision was 0.72 and it had a recall of 0.07.

Figure 13: Confusion matrix and ROC curve of the initial, unoptimized model.



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*. 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 8 shows the various scores for these five models. Using this method, it was determined that a ratio of 9 *Unpopular* articles to *Popular* articles in the training set provided the best results.

Table 8: 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
None (~10:1)	0.78	0.07	0.132
9:1	0.75	0.07	0.133
8:1	0.64	0.09	0.156
7:1	0.55	0.12	0.192
6:1	0.47	0.15	0.226

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 seen in Table 9. For max depth this range was composed of all even integers between 10 and 25, and for the amount of estimators the options were 50, 100, 150, and 200. 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 22 and the best amount of estimators to be 200.

Figure 9: Table displaying the range of values tested during hyperparameter tuning.

Parameter	Values
max_depth	10, 12, 14, 16, 18, 20, 22, 24
n_estimators	50, 100, 150, 200

7.3 Optimized Model

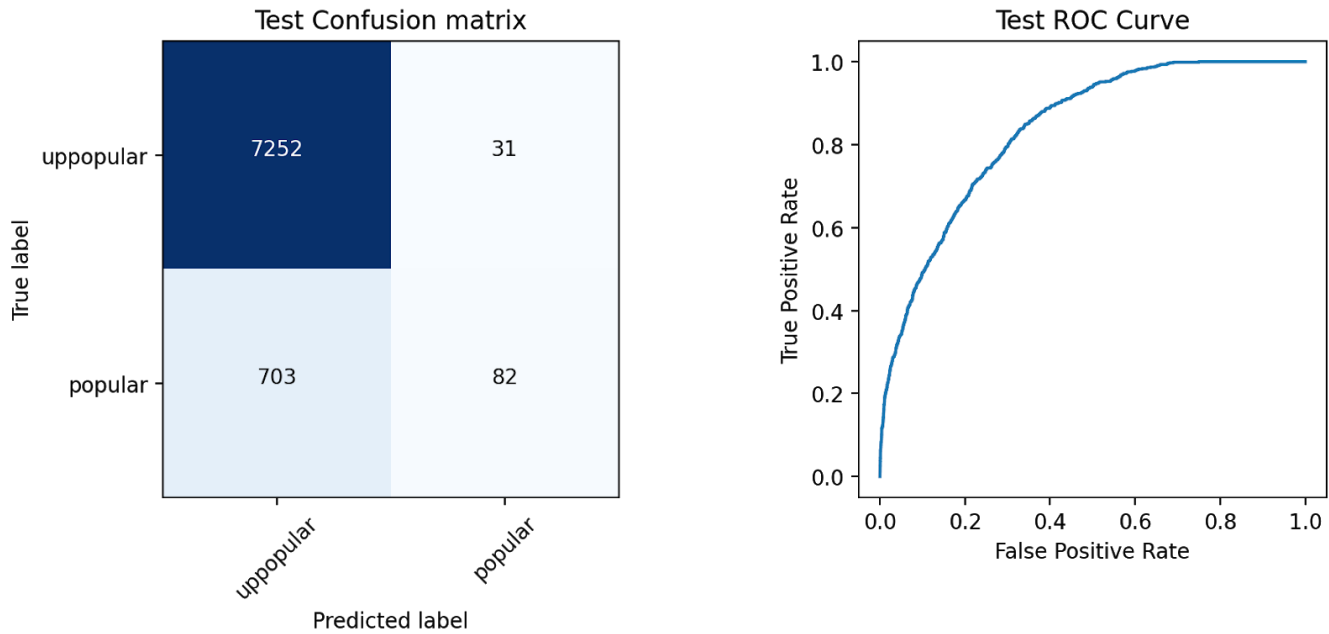
When creating the optimized model, the probability threshold was experimented with to find the best certainty of which to determine labels. Thresholds ranging from 0.3 to 0.7 in increment of 0.1 were tested and it was determined that the default 0.5 had the best balance of precision and recall.

Figure 14 shows the confusion matrix and ROC curve of the optimized model and Table 10 presents the corresponding scores. Overall the increase was quite small, but nonetheless successful: not only did the recall increase, but also the precision.

Tabel 10: Scores of the unoptimized and optimized models set

Model	Precision	Recall	F1 Score	Accuracy
Unoptimized	0.72	0.07	0.12	90%
Optimized	0.73	0.10	0.18	91%

Figure 14: Confusion matrix and ROC of the final, optimized model



In the end the trade-off between recall and precision was too large to change significantly. 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 a ratio of 9 *Unpopular* to 1 *Popular* article was best for balancing the training data and that the optimal hyperparameters were a max depth of 22 and a total of 50 estimators. Figure 15 shows the training set confusion matrix with a probability threshold of 0.5. Overall, this model had a training precision of 1.0, recall of 0.87, F1 score of 0.93, and accuracy of 99%.

Figure 15: Training confusion matrix of the final model.

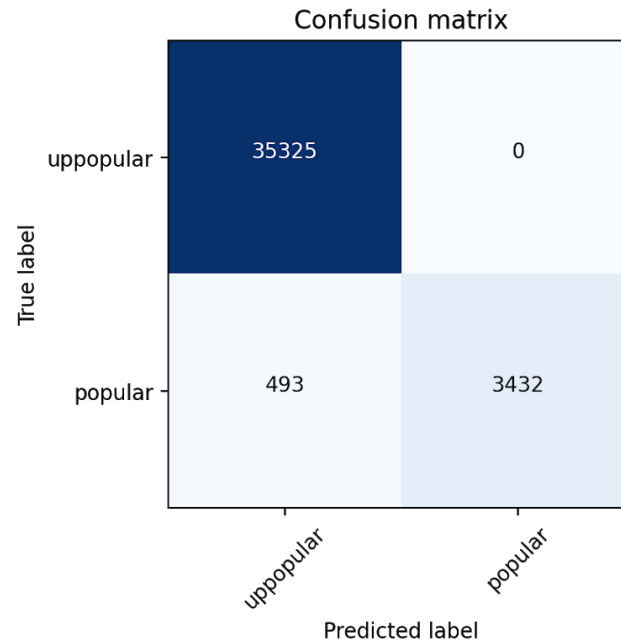
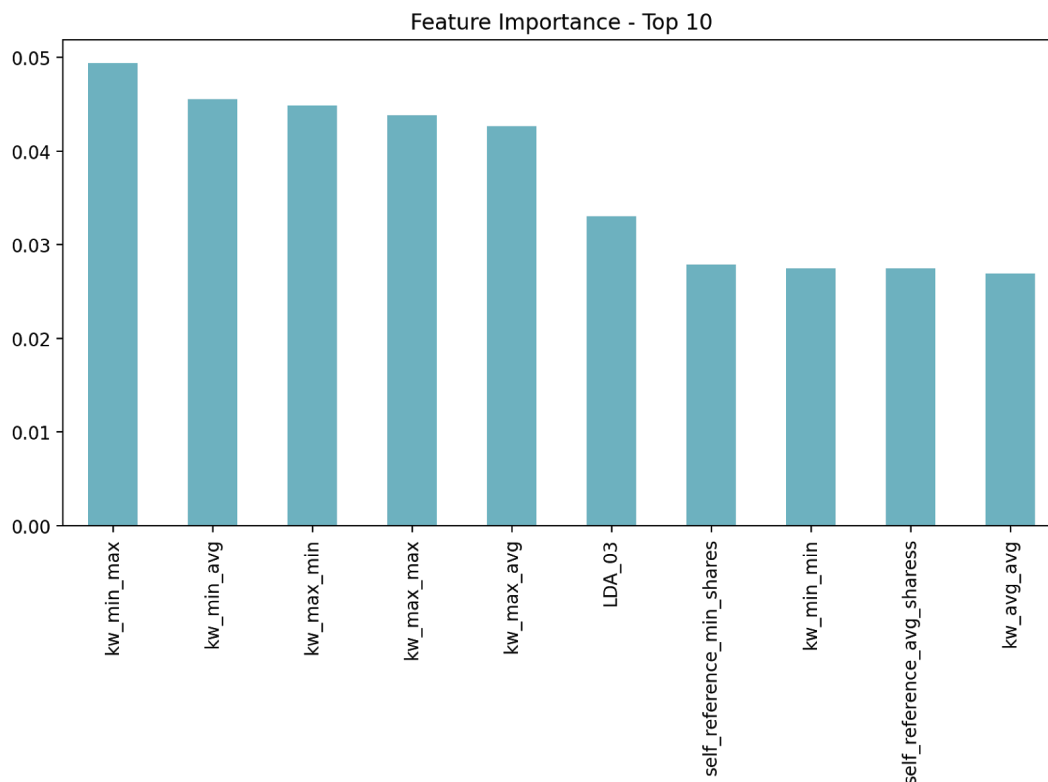


Figure 16 shows a bar graph of the top ten more important features in determining popularity according to the model. It appears that the keyword features hold the most weight, followed by the shares of other Mashable articles referenced within the text of an article. These factors could imply that the greatest factor in an article's popularity is its relationship to other articles, either by direct references or shared keywords.

Figure 16: Bar graph of the top 10 most important features according to the model.



8 Conclusions

The shares of Mashable articles range between a minimum of 1 share 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) while the higher subclass of *Viral* articles were those over 190,000 shares.

Overall it appeared that the greatest indications of popularity was an article's connection to other popular articles. The features seen most consistently associated with popularity were the keyword features, which specified the overall performance of other articles in the target article's associated keywords. A majority of *Popular* articles had overall high performing keywords, and these features were also deemed the most important in determining popularity according to the final model. Another common feature were those associated with the shares of other Mashable articles mentioned in the text of the target article.

The importance of the interconnected qualities of the articles can also be inferred by comparing the median shares over time with the frequency of *Viral* articles. Overall the median tends to rise when more *Viral* articles are published, indicating that the popularity of a singular article can greatly affect the overall popularity of other articles on the website.

In terms of actual article content, there was no equation for the perfect popular article. There were some common features, though, that stood out in the results of both clustering and a glance at the correlation matrix of *Viral* articles. The first was that positivity seemed to be a big commonality between popular articles, while negativity was greatly associated with lower shares. The second was article length: longer articles with more words seemed to perform better overall, especially compared to articles with little to no text.

8.1 Modeling

Popularity is not easily modeled. The final optimized model struggled to recognize a majority of *Popular* articles, often mislabeling them as *Unpopular*. Overall the model could only discern a particular subset of *Popular* articles, suggesting that there is much overlap between the features of popular and unpopular articles.

8.2 Suggestions

My personal suggestions to Mashable would be to look deeper at the relationships between articles and use these connections to find better ways of promoting the flow of popularity. An example would be to clean up the current keyword system. As mentioned before, this dataset contained 16,724 separate keywords with less than half being used for more than one article. Limiting keywords to a concrete list of topics would help solidify the connections between articles through keywords and could also aid recommendation engines to provide better suggestions to viewers based on previous reading habits.

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_shares	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

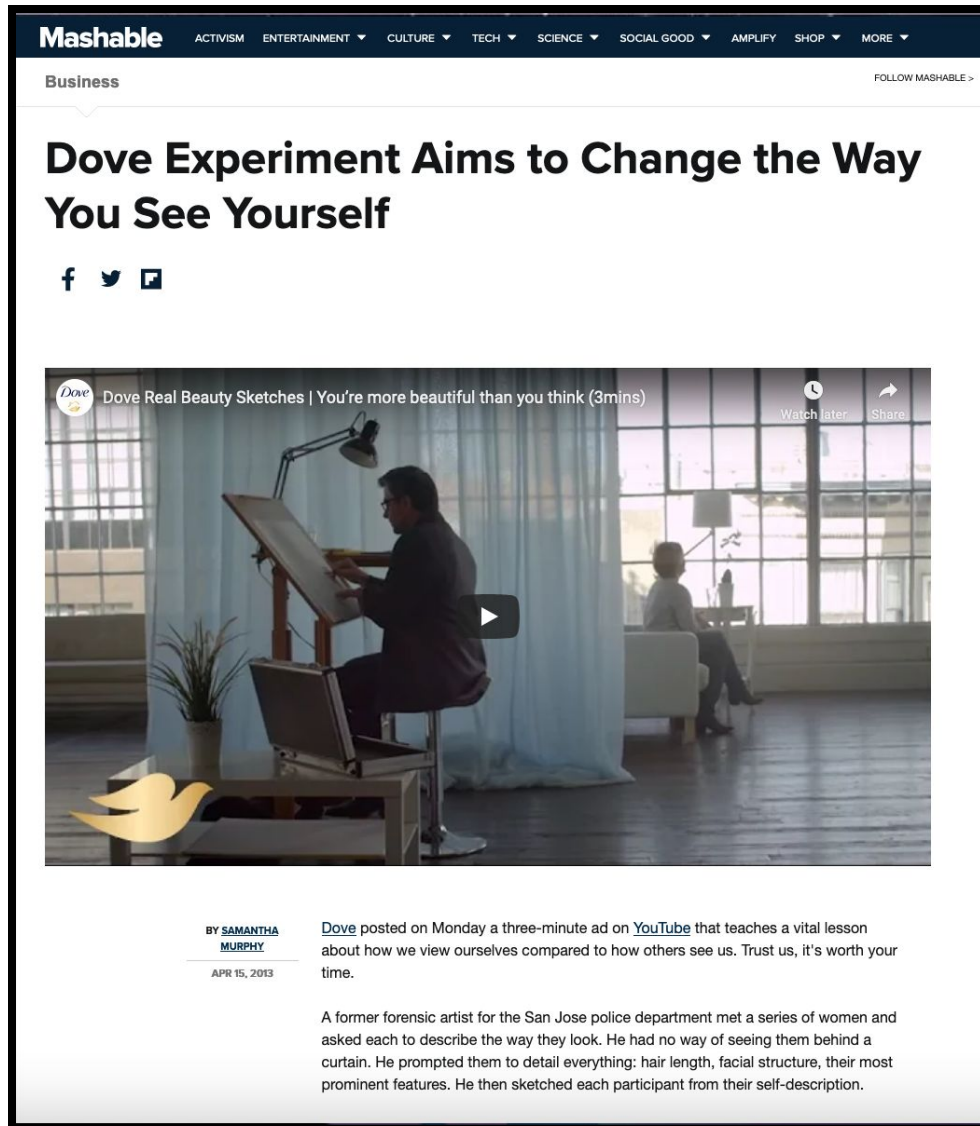
Column Names	Mean	STD	Minimum	Maximum
timedelta	355	213	8	731
n_tokens_title	10.3	2.11	2	23
n_tokens_content	546	471	0	8474
n_unique_tokens	0.548	3.53	0	701
n_non_stop_words	0.997	5.24	0	1042
n_non_stop_unique_tokens	0.689	3.27	0	650
num_hrefs	10.8	11.3	0	304
num_self_hrefs	3.29	3.86	0	116
num_imgs	4.54	8.31	0	128
num_videos	1.25	4.11	0	91
average_token_length	4.54	0.844	0	8.04
num_keywords	7.22	1.91	1	10
self_reference_min_shares	3975	19467	0	843300
self_reference_max_shares	10304	40836	0	843300
self_reference_avg_sharess	6375	23953	0	843300
LDA_00	0.185	0.263	0	0.927
LDA_01	0.141	0.220	0	0.926
LDA_02	0.216	0.282	0	0.920
LDA_03	0.224	0.295	0	0.927
LDA_04	0.234	0.289	0	0.927
global_subjectivity	0.443	0.117	0	1
global_sentiment_polarity	0.119	0.097	-0.394	0.728
global_rate_positive_words	0.040	0.017	0	0.155
global_rate_negative_words	0.017	0.011	0	0.185
rate_positive_words	0.682	0.190	0	1

rate_negative_words	0.288	0.156	0	1
avg_positive_polarity	0.354	0.105	0	1
min_positive_polarity	0.095	0.071	0	1
max_positive_polarity	0.757	0.248	0	1
avg_negative_polarity	-0.259	0.128	-1	0
min_negative_polarity	-0.522	0.290	-1	0
max_negative_polarity	-0.107	0.095	-1	0
title_subjectivity	0.282	0.324	0	1
title_sentiment_polarity	0.071	0.266	-1	1
abs_title_subjectivity	0.342	0.189	0	0.50
abs_title_sentiment_polarity	0.156	0.226	0	1
shares	3392	11623	1	843300
kw_min_min	400	361	1	4500
kw_min_avg	2441	775	406	11650
kw_min_max	93429	177515	446	843300
kw_avg_min	92	211	1	5800
kw_avg_avg	3437	426	1029	40771
kw_avg_max	398775	282780	1200	843300
kw_max_min	285	570	1	15000
kw_max_avg	4742	4036	1334	100459
kw_max_max	339648	297141	3800	843300

II. Figures

Figure 1: Screenshots of articles on the Mashable website

a) A popular Mashable article ([Link Here](#)) with 690,400 shares as of the date the data was collected

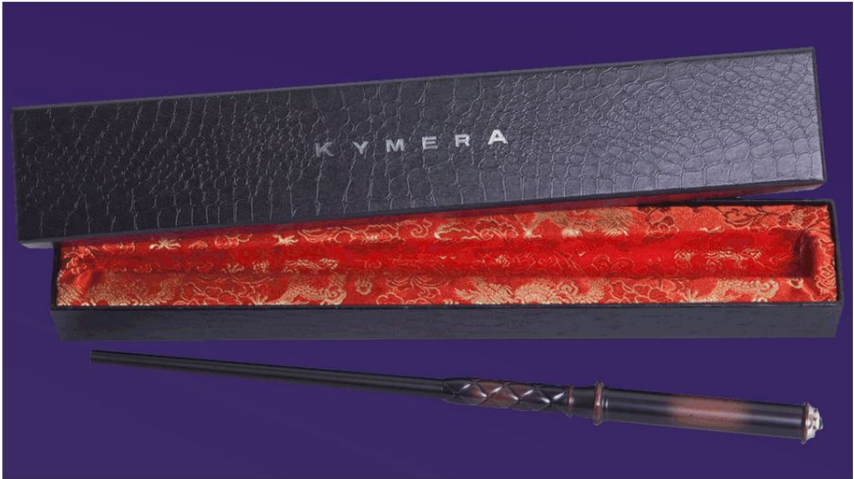


b) An unpopular Mashable article ([Link Here](#)) with only 1 share as of the date the data was collected

Mashable
ACTIVISM
ENTERTAINMENT
CULTURE
TECH
SCIENCE
SOCIAL GOOD
AMPLIFY
SHOP
MORE

Impress Your Muggle Friends With a Magic Wand TV Remote

f t



BY [TAYLOR CASTI](#)
DEC 09, 2013

Product Name: Magic Wand TV Remote **Price:** \$78.99 **Who would like this?:** Harry Potter enthusiasts, D&D superfans and other would-be wizards; television junkies; those afraid of buttons.

Remote controls are for Muggles. If you're still convinced that your Hogwarts acceptance letter got lost in the Owl Post, then you *need* this Magic Wand TV Remote from Kymera, as seen on [ThinkGeek](#). Made of springy willow and containing the heartstring of one particularly saucy dragon, it's the perfect addition to any living room. Instead of having to push tiresome buttons, you can swish and flick your way through hours of your favorite entertainment (we're thinking *Bewitched*, *Hocus Pocus*, *Lord of the Rings*, and of course all eight *Harry Potter* Movies).

Figure II: Screenshots of Mashables html and the locations of web-scraped features

a) Data channel

```

392 <div class='post-content'>
393 <script>
394   window.variationsData = []
395 </script>
396 <div class='post-slider'><article class='full post story' data-channel='Business'
397 <header class='article-header'>
398 <h1 class='title' href='https://mashable.com/2013/04/15/dove-ad-beauty-sketches/'>
399 <script>
400   window.variationsTitleElem = 'article header h1.title';
401 </script>

```

b) Keywords

```

634 <footer class='article-topics'>
635 Topics:
636 <a href="/category/advertising/">Advertising</a>, <a href="/category/business/">Business</a>, <a href="/cat

```

c) Title

```

14 </script>
15 <title>Dove Experiment Aims to Change the Way You See Yourself</title>
16 <link href="/assets/app-6be72717c63cc7300b02ad22d641e6ee9cf4d06a73beb6a38e1ca18a38fda358.css"

```

d) Date

```

443 <div class='article-image'></div>
444 <div class='article-info'><span class='byline'><span class='author_name'>By <a href="/author/samantha-murphy/">Samantha Murphy</a></span><time datetime='Mon, 15 Apr 2013

```

Figure III: Flow chart of keyword feature processing.

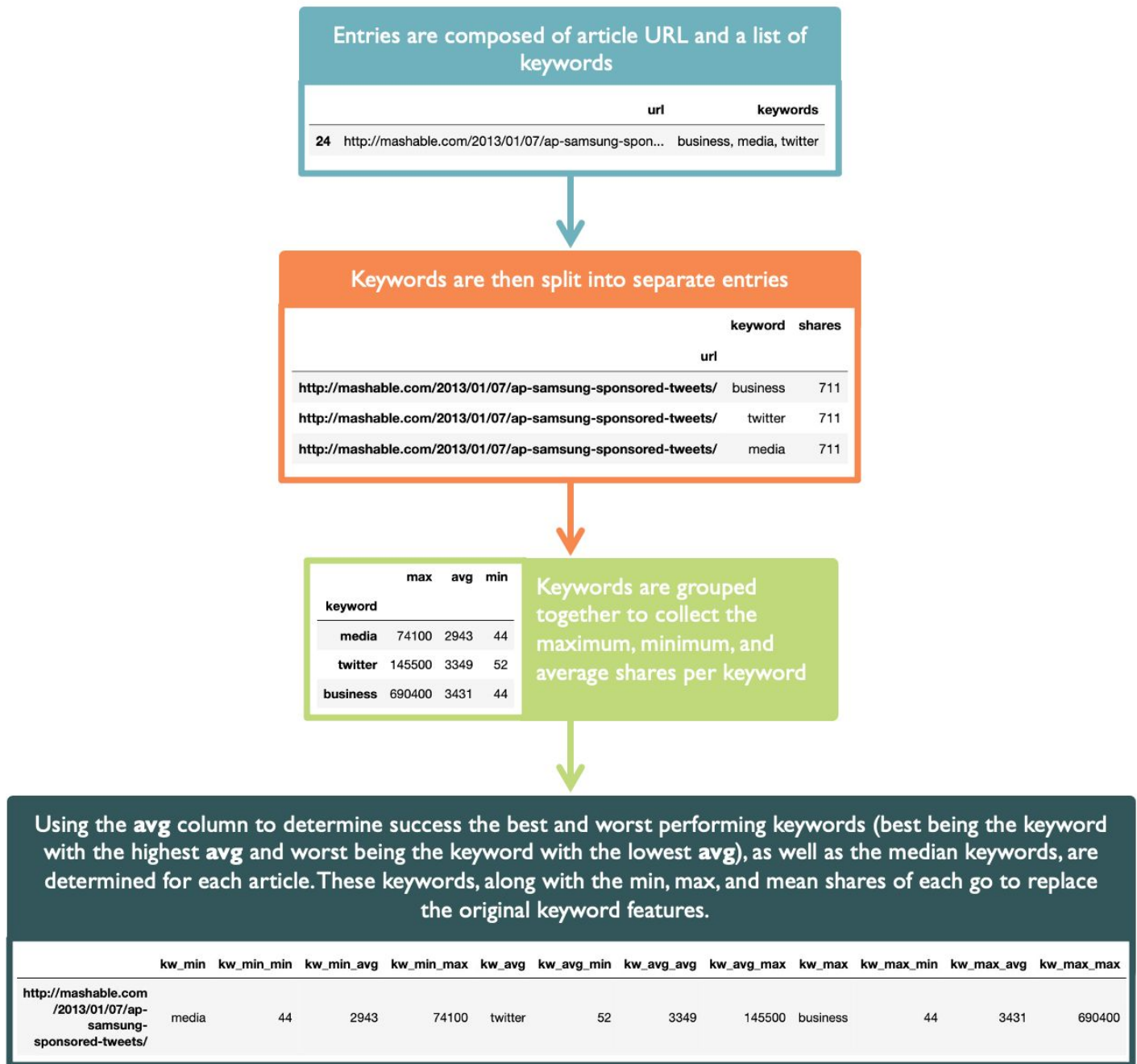
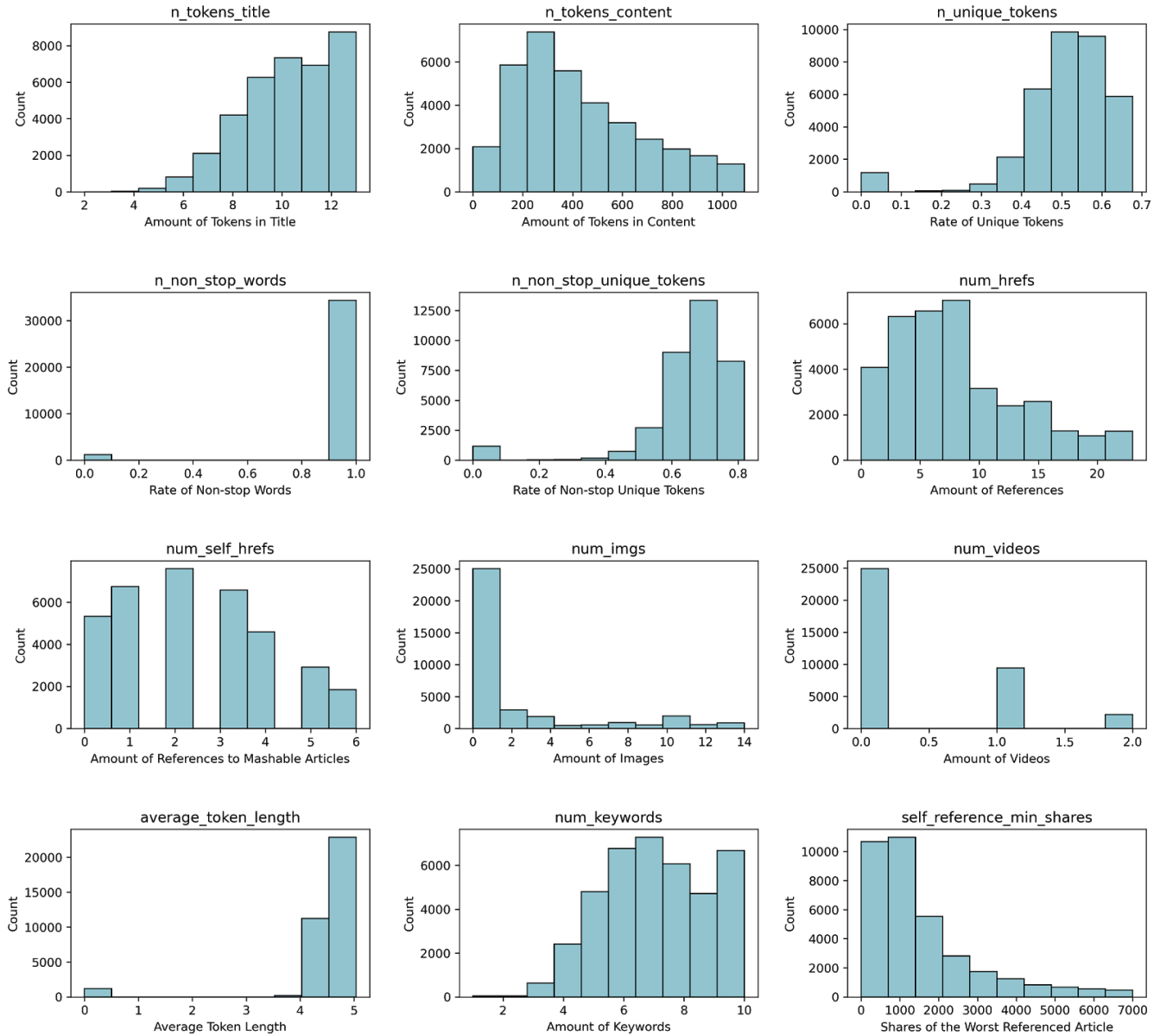
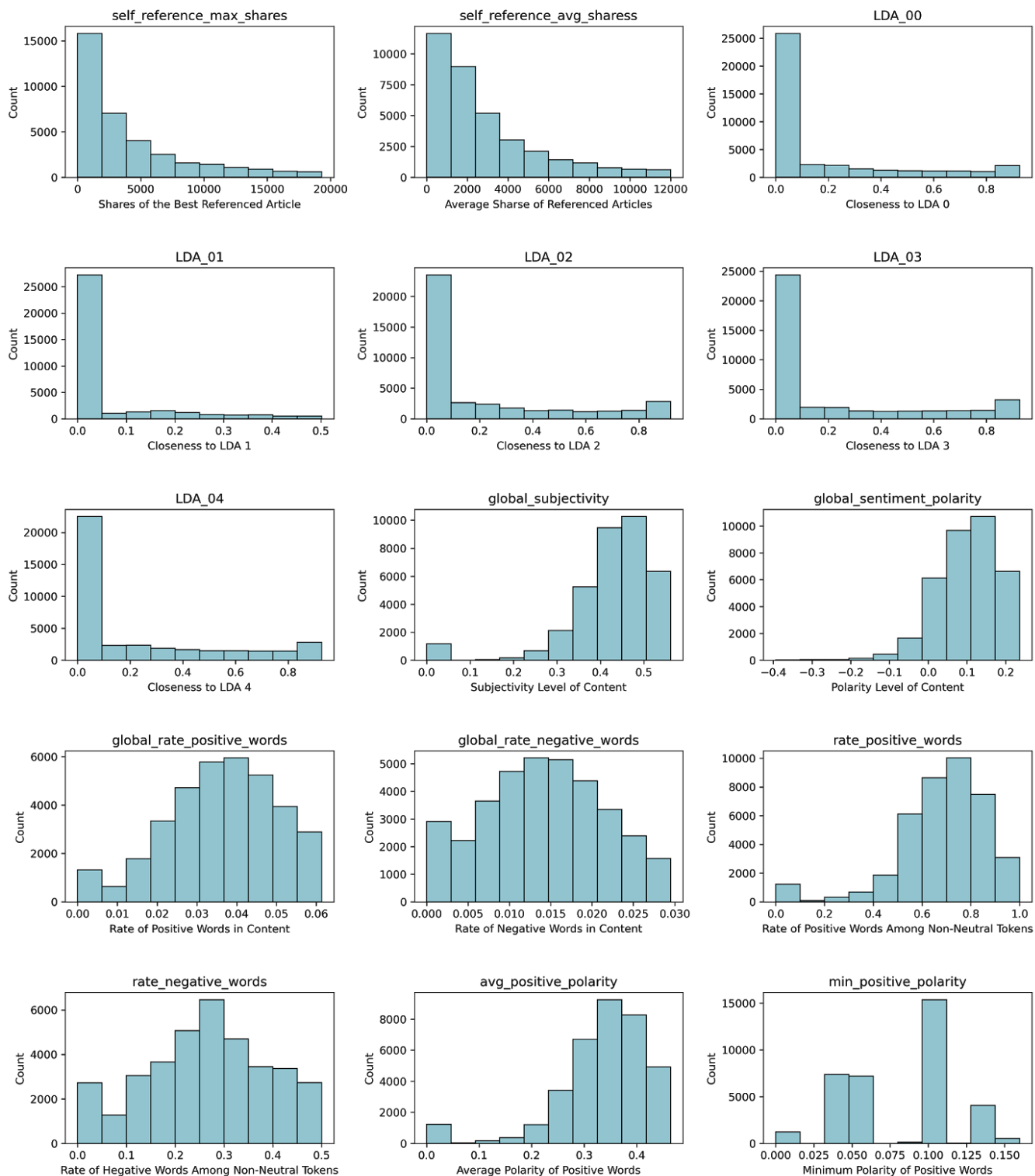
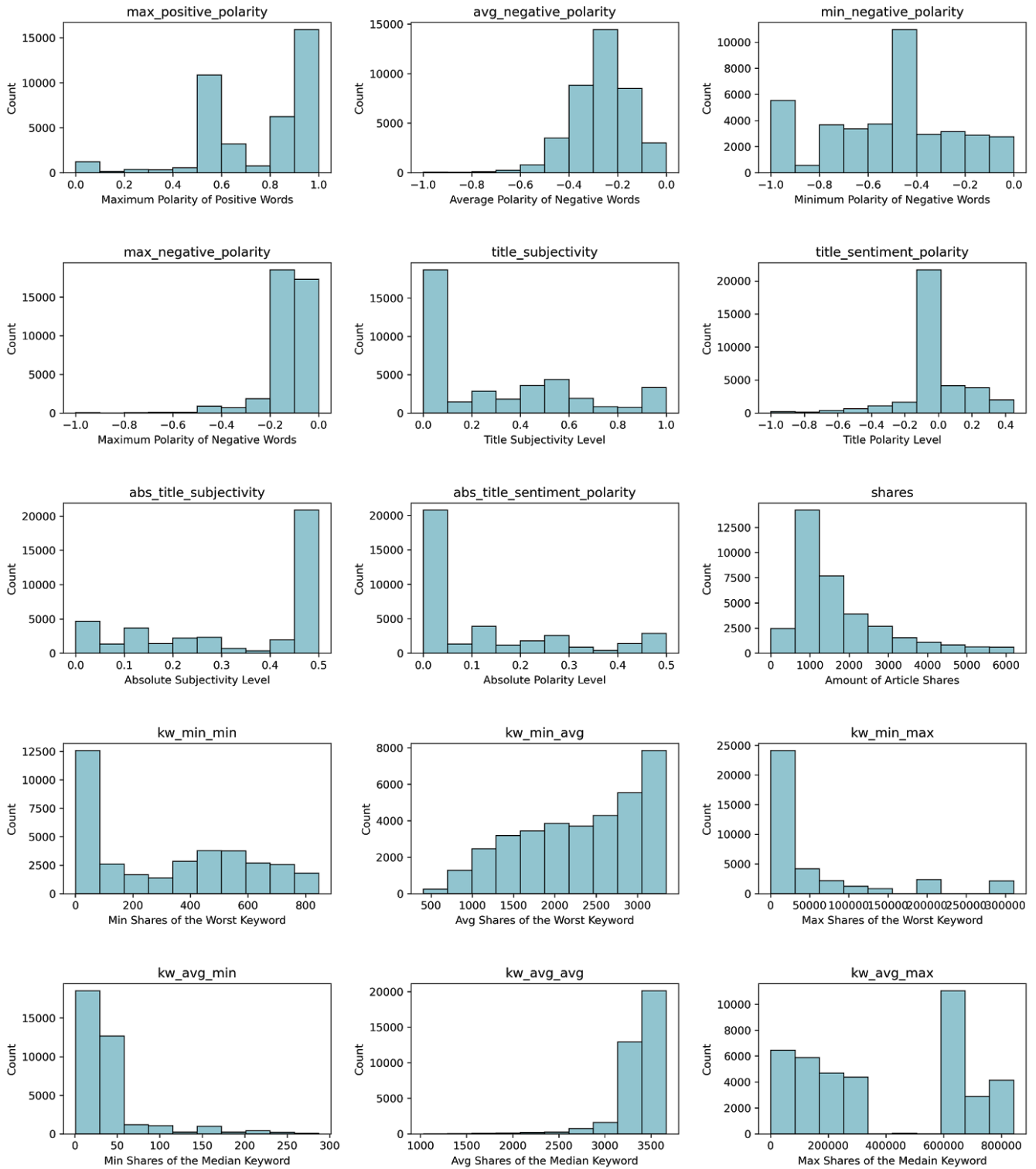


Figure IV: 45 graphs displaying the distribution of each feature. The title of each graph is the column name and the x label underneath presents the column description. For all graphs with a maximum larger than three times the standard deviation from the mean, the range has been cut to only include points within the 90th percentile.







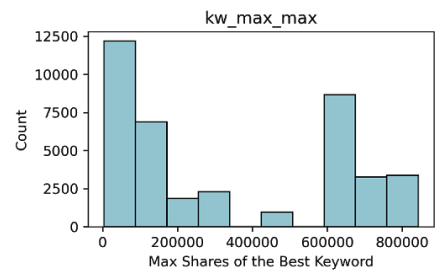
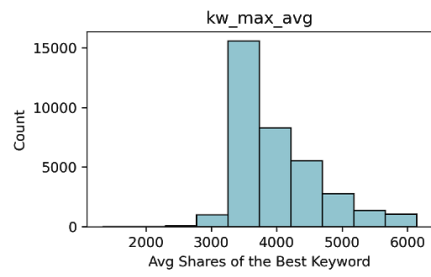
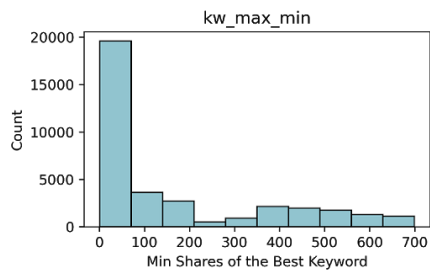


Figure V: 44 graphs displaying the distribution of each feature with different lines representing one of the four clusters. The title of each graph is the column name and the x label underneath presents the column description. For all graphs with a maximum larger than three times the standard deviation from the mean, the range has been cut to only include points within the 90th percentile.

