# Shares of World News Articles Study

**To: Tommy Tang (& those within Mashable)**

**By Group 8: Sam Burch, Grant Edwards, Wenxuan Gu, Michael Huang, & Seok Jun Seo**

## 1 Introduction (Dataset and Research Goal)

Our research goal for Mashable was to primarily find which characteristics in our data set make a world news article more likely to be shared. In addition to our primary goal, we also had three secondary topics we wanted to analyze to see if any of the three had an impact on the share count, specifically an increase in shares. The three specific questions were the following:

*a. How the weekday of publishing (e.g., Monday vs. Wednesday vs. weekend) affects the number of shares an article gets.*

*b. How the tone and sentiment (polarity) of an article, such as whether it's positive, negative, or neutral, affects its share count.*

*c. The effect of including audio-visual components (like pictures and videos) in articles on their share count*

Before we began modeling, we looked at the data set and found that there were a total of 39,644 articles of which 8,427 belonged to the world news channel. Focusing on the set of 8,427, the data contained the response variable, shares, along with 52 different characteristics about each article that assisted us in our exploratory data analysis (EDA) and model building. After reducing our dataset to only focus on the world news articles, we took a closer look at each of the 52 different characteristics and found one discrepancy. There are two characteristics in the dataset named *rate_postive_words* and *rate_negative_words* both of which are on a scale from zero to one and indicate the amount of positive or negative words in the article. These two columns should sum up to one, and we found 439 instances where both the *rate_postive_words* and *rate_negative_words* were zero. Since these characteristics were inaccurate for those 439 articles, we thought it would be best to remove them from our dataset to avoid incorrect conclusions.

In addition to removing 439 rows in our set we also standardized each of the columns. For each of our predictors we calculated the mean and standard deviation and performed the following calculator on each row:

$$new\ column\ values\ = \frac{(old\ column\ values - mean(old\ column)}{Standard\ Deviation\ (old\ column)}$$

The reasoning behind standardizing our predictors was to allow for better comparability when we befored our modeling and our exploratory data analysis. By standardizing, we bring all of our predictors to a common scale. This ensures that the predictors that previously had larger scales

do not dominate our analysis. Below is a display of the first few rows of our dataset after we cleaned and performed standardization.

| | n_tokens_title | n_tokens_content | n_unique_tokens | n_non_stop_words | n_non_stop_unique_tokens |
|---|---|---|---|---|---|
| 10 | -0.29093 | -0.95828 | 1.38980 | -1.36095 | 1.35213 |
| 11 | -0.77197 | 1.51179 | -0.41784 | 0.91712 | 0.56759 |
| 16 | 0.67116 | 0.13710 | -0.79475 | 0.46455 | -0.59104 |

After we cleaned the dataset we then moved onto our model building procedures to find the top characteristics that would influence the number of shares in a world news article. We used several different modeling techniques including Linear Regression, Random Forests, XGBoosting, Gradient Boosting Machine, and a model ensemble. Each of these have their own strengths and weaknesses when it pertains to model building so we compared each model based on their RMSEs to see which performed the best.

To answer our three specific questions we used EDA techniques including bar charts, correlation plots, and other graphics to better visualize and analyze how shares were impacted. We found that looking at not only the total shares but also the average shares was important for these specific questions since we did have a few outliers that would potentially skew our data results. In the following pages we provide detail into how and why we performed the methods in order to construct both a model that will help increase shares and to provide feedback on the answers to the secondary questions.

## 2 Methods and Results

### 2.1 Linear Regression

The first of several modeling methods we tried was a linear regression. We started by fitting the model with all the standardized predictors from the training set. This will help compare the predictors once the model is built. We randomly split the data into 80% training and 20% testing to see how well our model does at predicting. (Note: A couple of predictors had some NA values, but we were unable to determine the issue.) Regardless, this model was okay with a RMSE of 4015. (This means our model was off by 4015 shares on average.) To drop unnecessary predictors, we applied the greedy algorithm (step function in R) with both directions allowed. This reduced the number of predictors to 21, which is a lot less than 58. The RMSE also slightly dropped to 4009. However, there are still more issues with this model (than just the accuracy). *This model is denoted on the right.*

Correlated predictors within a model can cause the model to overfit and give a biased prediction. We can reduce the impact of this – multicollinearity. The process for reducing this will be the following: look at all pairwise correlations greater than .5, create groups for each of these predictors, and drop the ones that have the smaller weight in the previous model. With such being the criteria, the groups are as follows (with the starred ones having the highest weight):

- Group 1: kw_min_max & kw_min_avg***
- Group 2: kw_avg_max & kw_max_max***
- Group 3: rate_positive_words, global_rate_positive_words, & global_sentiment_polarity***

Top Predictors (Original)

RMSE = 4009

| Name | Value[1] |
|---|---|
| self_reference_avg_sharess | 3511.20 |
| kw_avg_avg | 2758.26 |
| n_non_stop_words | -2646.49 |
| num_imgs | 1935.05 |
| kw_min_avg | -1723.32 |
| num_videos | 1635.69 |
| num_hrefs | 1537.13 |
| kw_min_min | 1481.08 |
| LDA_03 | 1334.06 |
| global_sentiment_polarity | -1224.78 |
| kw_max_max | 1209.71 |
| kw_min_max | 1114.83 |
| average_token_length | -1099.91 |
| rate_positive_words | 892.42 |
| n_non_stop_unique_tokens | -752.00 |
| global_subjectivity | 685.52 |
| global_rate_positive_words | 682.24 |
| kw_avg_max | -671.73 |
| n_tokens_title | 567.34 |
| weekday_is_wednesday | -465.85 |
| weekday_is_tuesday | -418.39 |

[1] Color indicates impact (+/-) of the predictor.

After dropping the four predictors, we fit a model with the remaining variables. Once we apply the Greedy algorithm again, two predictors are dropped. The two that were dropped were kw_max_max and global_sentiment_polarity. Thus, our final linear regression model *on the right* keeps 15 of the original 58 predictors. This final model has a RMSE of 4007. The model suggests to reference more articles with a high amount of shares, include more popular keywords, and reduce the number of non-stop words with better punctuation, grammar, and shorter sentences.

After trying to fix diagnostic issues, *in Appendix 1*, we were unable to do so. This leaves us with a model that is simply not good enough to put into use, because of the inaccuracy and diagnostic issues. Although that is the case, we can tell which predictors are likely to be important from those that were kept during the process. Hence, let us hope to reduce the error with other modeling methods.

Top Predictors (Final)

RMSE = 4007

| Name | Value[1] |
|---|---|
| self_reference_avg_sharess | 3530.09 |
| kw_avg_avg | 2713.45 |
| n_non_stop_words | -2476.60 |
| num_imgs | 2003.91 |
| num_videos | 1572.12 |
| num_hrefs | 1489.46 |
| kw_min_avg | -1478.08 |
| LDA_03 | 1213.11 |
| average_token_length | -1128.24 |
| kw_min_min | 874.51 |
| global_subjectivity | 816.90 |
| n_non_stop_unique_tokens | -664.99 |
| n_tokens_title | 579.00 |
| weekday_is_wednesday | -476.23 |
| weekday_is_tuesday | -404.41 |

[1] Color indicates impact (+/-) of the predictor.

### 2.2 XGBoosting, Random Forests, & Gradient Boosting Machine

The next model we tried was XGBoost, which is a good choice for a regression problem. We first constructed a very simple XGBoost model using all predictors and obtained a very high MSE of 86,105,124. After developing the simple XGBoost model, we attempted to tune the hyperparameters. The best selection of hyperparameters is as follows:

| Hyperparameter | Value |
|---|---|
| Booster | gbtree |
| Objective | reg:squarederror |
| Eta | 0.3 |
| Gamma | 1 |
| Max_depth | 10 |
| Min_child_weight | 1 |
| Subsample | 1 |
| Colsample_bytree | 1 |
| nrounds | 100 |

Then, we applied k-fold validation. Interestingly, we found that the MSE varied with each fold. We concluded that k-fold validation should be used here, as it would likely improve the model's ability to predict the number of shares when facing new data. After hyperparameter tuning and k-fold validation, we reduced the model's MSE to 17,500,318.
Following the hyperparameter tuning, we began variable selection. We used the greedy algorithm to identify the model with the best MSE:

| Variable | MSE |
|---|---|
| -weekdays columns | 17273442 |
| -is_weekend | 18855805 |
| -num_self_hrefs | 17020878 |
| -num_images | 17013901 |
| -num_videos | 16616056 |
| -average_token_length | 14708572 |
| -num_keywords | 14357730 |
| -kw_maxmin | 13998728 |
| Final model | 13488399 |
| Before variable selection | 17500318 |

As you can see, columns such as weekdays, number of images, and number of videos contribute little to the model. Considering the complexity of the model and the improved MSE after deletion, we decided to remove these columns from the model. Other columns like

num_self_hrefs, average_token_length, number_keywords, and kw_max_min all negatively influence the model. We decided to delete these columns to achieve a lower MSE.

After obtaining the best XGBoost model, we began to construct Random Forest and Gradient Boosting Machine models, which are both based on decision trees. Therefore, we used the same environment for these two models and also achieved a low MSE.

### 2.3 Model Ensemble

We still aimed to lower the MSE of the model to achieve better predictions. The next approach we attempted was to merge these models with each other. We made a simple attempt by calculating the average prediction made by these models, resulting in the following:

| Model | MSE |
|---|---|
| GBM+XGBoosting | 12380403 |
| GBM+XGBoosting+RF | 12158636 |

The result appears much better than the previous model. We decided to select GBM+XGBoost+RF as our final model. We also experimented with combining linear regression with these models. However, the significant increase in MSE led us to abandon this idea.

| Model | RMSE |
|---|---|
| XGBoosting(fine-tuned) | 3672 |
| Random Forest(fine-tuned) | 3577 |
| Gradient boosting machine | 3633 |
| Model Ensemble(XGB,RF,GBM) | 3486 |

Furthermore, having selected our best model, we wanted to determine which variable most influences the number of shares. We allowed these models to calculate the importance of the predictors and took the average of these values. The 'n_tokens_title' variable had a very high importance in GBM and XGBoost, but a significantly lower importance in Random Forest. Considering the low value in Random Forest, we decided to use a voting system. We allowed these three models to each select their top 5 most important predictors. The results are as follows:

| XGBoosting | Random Forest | GBM | Voting by three models |
|---|---|---|---|
| n_tokens_title | max_negative_polarity | avg_negative_polarity | LDA_02(3) |
| LDA_02 | n_tokens_content | LDA_02 | max_negative_polarity(3) |
| num_hrefs | LDA_02 | kw_max_avg | n_tokens_content(2) |
| n_tokens_content | avg_positive_polarity | LDA_01 | LDA_01(2) |
| max_negative_polarity | LDA_01 | max_negative_polarity | Others |

From the results, we observe that LDA_02 and max_negative_polarity each received three votes. We concluded that the most important predictor lies within these two variables. Next, we compared the standardized values of importance to select the most significant one. The result is as follows:

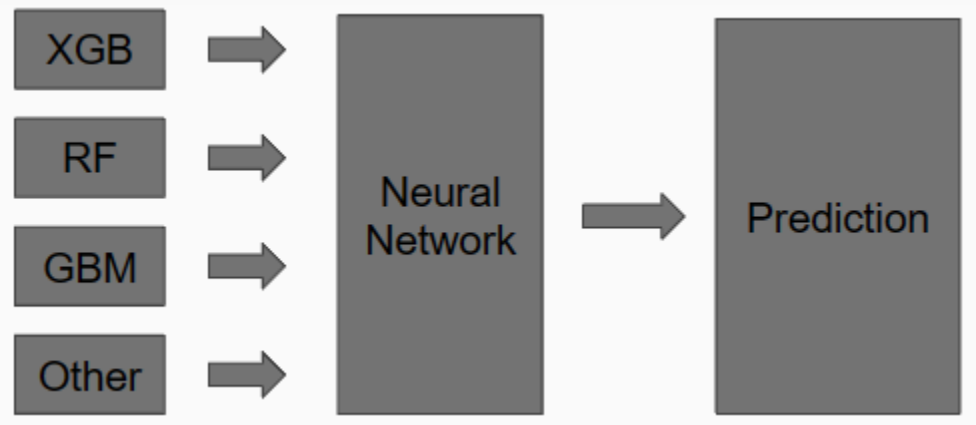
| Predictor | Standardized Importance |
|---|---|
| LDA_02 | 0.42804111 |
| max_negative_polarity | 0.35701611 |

The results show a higher standardized importance value for LDA_02. Therefore, we conclude that LDA_02 is the most important variable in our final model for this dataset. This approach is more reliable than simply calculating the average value of each predictor.

We also want to highlight some limitations of our model for future adjustments. Our model is based on decision tree methods, which means that if the distribution of the dataset changes, we should reconsider the selection of hyperparameters. The current hyperparameters may only be effective for the current distribution of the number of shares. To address this issue, we suggest applying Bayesian optimization, allowing the model to adjust its hyperparameters autonomously each time.

### 2.4 Metamodeling

After obtaining results from the final model, we continued to explore ways to achieve a lower MSE. Upon further consideration of the ensemble model, we devised a method that integrates all the models into a neural network.




This approach is known as metamodeling, often referred to as the 'model of models.' We constructed a simple neural network and integrated our existing models into this network. We experimented with two activation functions: Sigmoid and ReLU. The results are as follows:

| Activation Function | MSE | RMSE |
|---|---|---|
| Sigmoid | 11755837 | 3428 |
| ReLU | 13420674 | 3663 |

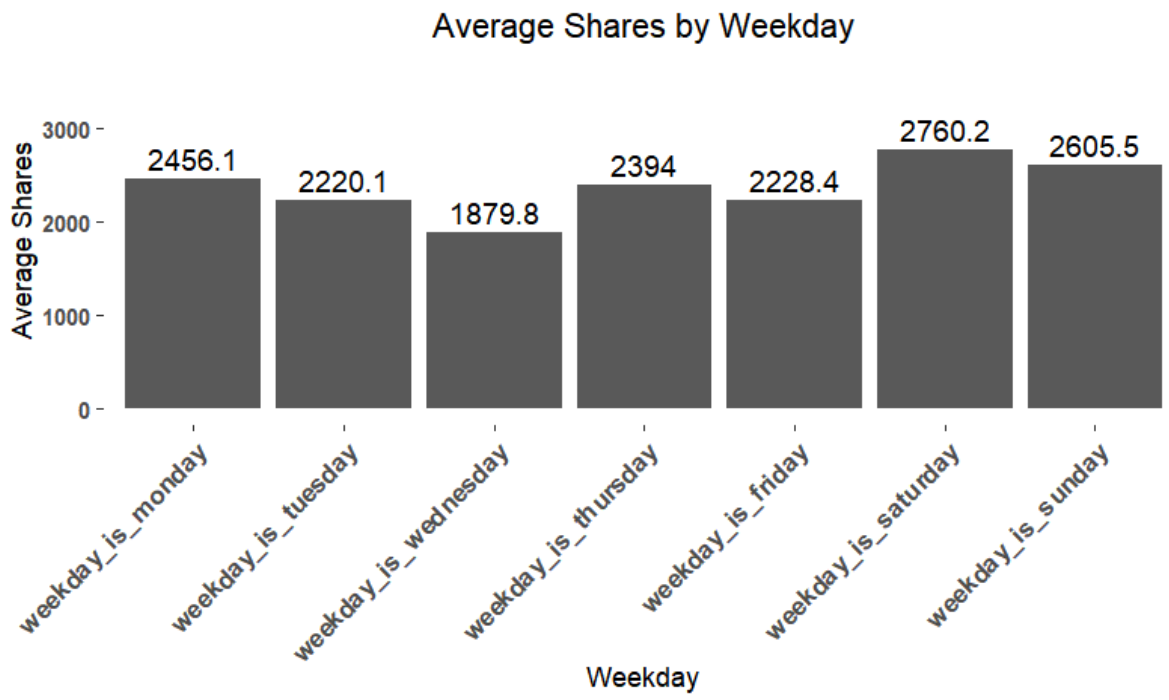
The neural network with the ReLU activation function converges very quickly. However, the network using the Sigmoid activation function achieved a better MSE. We suggest that Sigmoid is more suitable for this dataset. Here, we constructed a very simple neural network, used only a few training epochs, and observed that the training loss still decreased rapidly. This indicates that the model has not yet converged and there is considerable room for improvement.

### 2.5 Day of the Week

The table contains data about shares for each day of the week. The plot below represents the average values for each weekday, which are all quite similar. We see that **Wednesday has the lowest** number of shares, averaging about **1879.79**. Saturday is the best day, with approximately **2760.20** shares.

| Weekday | Average Shares | Total Shares |
|---|---|---|
| Monday | 2,456.05 | 3,330,409 |
| Tuesday | 2,220.14 | 3,432,328 |
| Wednesday | 1,879.79 | 2,941,868 |
| Thursday | 2,394.01 | 3,756,199 |
| Friday | 2,228.41 | 2,908,077 |
| Saturday | 2,760.20 | 1,432,545 |
| Sunday | 2,605.48 | 1,477,309 |

However, the difference in average shares on any given weekday is less than 1000. This suggests that individual weekdays are not strong predictors. Despite this observation, if posting during weekdays is necessary, it is advisable to **avoid Wednesdays and instead favor weekends**, as weekends tend to be more advantageous for shares.



For further analysis, we could use a method called **Time Series Analysis**.This method requires data recorded at set times, like every day or week, to see how things change and spot trends. Time Series Analysis has several advantages. It helps us understand trends over time, like when articles get the most shares.

This method also **lets us predict future patterns, guiding us on the best times to publish**. Plus, it can reveal the impact of special events or seasons on article popularity, making our strategy more effective. Sadly, our data does not include time details, so we cannot do this detailed analysis. For better analysis, data should be collected regularly and include dates and times. It also should note any special events, like Thanksgiving, that might affect the results. In the future, if your internal teams can gather and share such time-specific data, it would substantially improve our ability to provide more detailed insights.

We recommend that your internal team focus on collecting specific data to improve the Time Series Analysis. Start by recording the '**Date of Publication** ' of each article in a standard format like YYYY-MM-DD, as this will allow us to track changes in article popularity over time. Additionally, gathering the **'Day of the Week'** each article is published, which can be linked from the publication date, this variable is significant to understanding weekly trends in reader engagement. Also, the **'Time of Publication'** should be noted for each article. Knowing the exact time when articles are published can provide patterns into how different times of day affect reader interaction and shares. Lastly, it is important to note any '**Special Events or Holidays**' that occur around the time of publication. This will help us analyze how these events might influence reader behavior and interest in the content. Once we have these new time related details, we can start seeing the trends over time. This will help us guess what might happen in the future and guide us to choose the best times for publishing articles to enhance the shares.

**2.6 Polarity**

We used the following questions to guide our analysis of variables relating to polarity:

1. Are polarizing articles more likely to go viral?
2. How does the intensity of polarization affect shares
3. Do positive or negative articles differ?
4. Can polarizing sentences or phrases influence shares?
5. How does subjectivity in world news impact share count?

For general analysis of polarity, we focused on the variables avg_positive_polarity and avg_negative_polarity and split the dataset into two: a non-viral dataset and a viral dataset. The variables are measured on a scale from 0 (neutral) to 1 (polarizing), with the negative polarity variable having been applied an absolute value transformation to standardize measure of intensity. The cutoff for virality was set at 3,500 shares from the IQR upper-whisker limit. There were 7,301 non-viral articles and 865 viral articles. We then categorized the intensity of the positive and negative polarization by 3 distinct buckets: weakly negative (less than 0.33), moderately negative (0.33 - 0.66), and highly negative (greater than 0.66). We then calculated the average share per article in each bucket. We will first examine the dataset with non-viral articles.

| Non-viral Articles (<3,500 Shares) | Weakly Negative (<0.33) | Moderately Negative (0.33 - 0.66) | Highly Negative (>0.66) |
|---|---|---|---|
| Number of Articles | 5,911 | 1,962 | 28 |
| Sum of Shares | 7,312,791 | 1,682,809 | 37,139 |
| Shares per Article | 1,237 | 1,235 | 1,326 |

Negative Polarity of Non-viral Articles

| Non-viral Articles (<3,500 Shares) | Weakly Positive (<0.33) | Moderately Positive (0.33 - 0.66) | Highly Positive (>0.66) |
|---|---|---|---|
| Number of Articles | 3,695 | 3,606 | 7 |
| Sum of Shares | 4,492,843 | 4,539,896 | 9,541 |
| Shares per Article | 1,215 | 1,258 | 1,363 |

Positive Polarity of Non-viral Articles

Here we notice that for highly polarizing articles, there simply isn't enough of them to make any conclusive statements. However, for weakly and moderately polarizing articles, we note that polarization doesn't seem to influence shares at all. All the shares are essentially the same across

both positive and negative measurements as well. One thing to note is that there seems to be a high prevalence of weakly negative articles at almost 6,000. Positive articles seem to be uniformly distributed among weak and moderate buckets. We conclude that polarization does not influence shares of nonviral articles. We proceed to examine viral articles.

| Viral Articles (>3,500 Shares) | Weakly Negative (<0.33) | Moderately Negative (0.33 - 0.66) | Highly Negative (>0.66) |
|---|---|---|---|
| Number of Articles | 675 | 185 | 5 |
| Sum of Shares | 66,449,700 | 2,585,500 | 59,900 |
| Shares per Article | 9,851 | 13,975 | 11,980 |

Negative Polarity of Viral Articles

| Viral Articles (>3,500 Shares) | Weakly Positive (<0.33) | Moderately Positive (0.33 - 0.66) | Highly Positive (>0.66) |
|---|---|---|---|
| Number of Articles | 380 | 485 | 1 |
| Sum of Shares | 4,116,800 | 5,178,300 | 6,500 |
| Shares per Article | 10,833 | 10,676 | 6,500 |

Positive Polarity of Viral Articles

For viral articles, we notice that similar to non-viral articles, highly polarizing articles are too far and too few to conclude anything statistically sound. The distribution of weak and moderate articles also matches the non-viral dataset as well, with positive articles having a roughly uniform distribution while negative articles are heavily skewed to be weakly negative. When looking at shares per article, we see that moderately negative articles perform the best with an share per article count at almost 14,000. Moderately negative viral articles perform nearly 50% better than the weakly negative, weakly positive, and moderately positive articles. We also note the weak performance of weakly negative articles as its share count is the smallest. However, we note that weakly negative articles are most likely to go positive, signaling the neutral to weak polarization of articles perform the best when seeking virality.

After we looked at the viral and non-viral share distribution of our articles we also thought it would be important to break down the data even further, to see if what was occurring on the extreme ends of either very high or very low polarity. We first took a look at the rate of polarizing words, specifically the rate of positive words in our articles and computed the average

shares of each group. You can see below from the table that the data was broken down into buckets of length 0.2 based on the rate of positive words in each article.

| Title | Very Neg Avg. | Mod Neg Avg. | Neutral Avg. | Mod Pos Avg. | Very Pos Avg. |
|---|---|---|---|---|---|
| Rate | 0.0-0.2 | 0.2-0.39 | 0.4-0.59 | 0.6-0.79 | 0.8-1.0 |
| Avg. Shares | 3,458 | 1,585 | 1,998 | 2,190 | 2,506 |

Rate of Positive Words Average Share Breakdown

From this table we were able to conclude that the articles with a very low positive word rate (Very Neg Avg.) produced the greater amount of average shares compared to the other categories. Similarly, we also constructed the same table for the global subjectivity of the entire article. We used identical buckets of length 0.2.

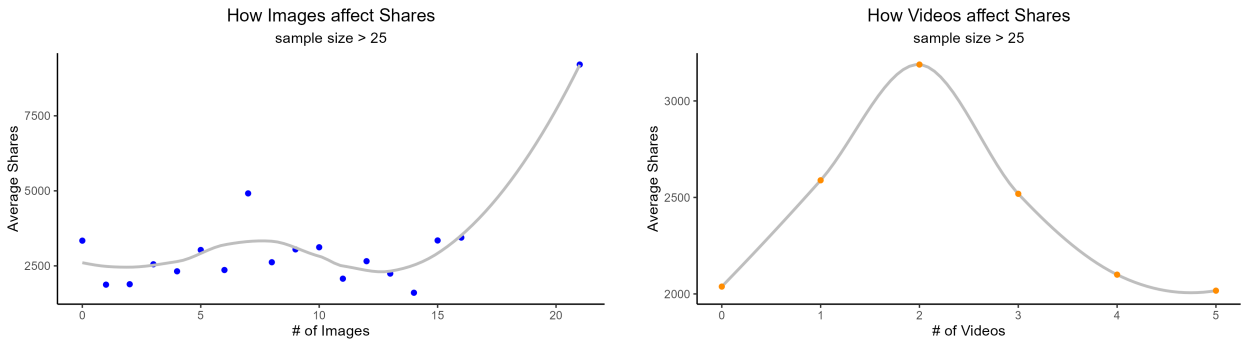
| Title | Very Obj GS | Mod Obj GS | Neutral GS | Mod Sub GS | Very Sub GS |
|---|---|---|---|---|---|
| Rate | 0.0-0.2 | 0.2-0.39 | 0.4-0.59 | 0.6-0.79 | 0.8-1.0 |
| Avg. Shares | 1,398 | 1,854 | 2,327 | 3,310 | 1,350 |

Global Subjectivity Average Share Breakdown

Here we can see that when we look at the subjectivity of an article it is more beneficial to be neutral or slightly subjective from the author's point of view. We tend to see the lowest amount of average shares when the articles are written with a very objective or subjective tone, so we recommend staying away from both extremes. Staying neutral allows for the reader to make their own decisions based on the article and not to be influenced by the author’s tone, this may result in them not sharing the article with others.

### 2.7 Images / Videos

To get a good idea of how images and videos affect the number of shares, let's look at their relationship with the average number of shares. We will restrict the sample size (number of articles with the specific amount of images or videos) to greater than 25. This reduces the impact of outliers.




Clearly, there is a positive relationship for images, however it is not strictly increasing. This means adding more and more pictures isn't going to constantly increase the average number of shares in general. Instead, there seems to be a couple areas to target – 8ish and 15ish. For videos, there are only 6 data points when we set the sample size restriction. So, all we can see here is a couple videos are better than nothing.

To dig deeper, we can look at several splits. Let us start with images:

Effect of Images by Categories

| Images | Average | Median | 10th Pct. | 25th Pct. | 75th Pct. | 90th Pct. | Sample Size |
|---|---|---|---|---|---|---|---|
| 0 | 3340 | 1300 | 680 | 863 | 2575 | 6030 | 738 |
| 1 | 1874 | 1100 | 589 | 783 | 1700 | 3200 | 4668 |
| 2 | 1888 | 1100 | 689 | 846 | 1700 | 3000 | 943 |
| 3+ | 2990 | 1300 | 728 | 924 | 2300 | 4600 | 1639 |

These four columns suggest not including images in articles on average, but that's not the end all be all. The median for 0 and 3+ are actually equal – suggesting the average is skewed right. Thus, while articles with no images have a high ceiling, 3+ have a higher floor. By only this

analysis, one should not include images if they want a higher ceiling, but they should add several for a higher floor. It is not as simple as this though. Hence, we recommend to only include images if they are truly meaningful and add to audience engagement. Articles without images can do well too. Therefore, you don't need to add an image or two if they don't bring anything new to the article.

Now, let us look at videos:

Effect of Videos by Categories

| Videos | Average | Median | 10th Pct. | 25th Pct. | 75th Pct. | 90th Pct. | Sample Size |
|---|---|---|---|---|---|---|---|
| 0 | 2038 | 1100 | 622 | 811 | 1700 | 3400 | 5481 |
| 1 | 2589 | 1100 | 629 | 827 | 2000 | 4500 | 1757 |
| 2+ | 2900 | 1300 | 733 | 924 | 2200 | 4510 | 750 |

When looking at splits for videos, we can see our prior is just confirmed here. A couple is better than nothing for videos. This shows that having one or two videos (that are also of great quality) can truly add to that audience engagement. In an age of YouTube and Tiktok, people like to consume information from videos. Having one that summarizes your article or adds a new touch is certainly beneficial to the article's success.

| **Variable** | **Value** | **Significance** |
|---|---|---|
| Images | 89.1 | 2.4*10^-9*** |
| Videos | 141.9 | 0.0029** |
| Interaction | -2.7 | 0.86 |

With regards to studying a potential interaction, we built more models. Above is the output for running a linear regression (with interaction) on the **number** of images and videos. We already know that the number of images and videos are important – from the linear regression analysis earlier on. For the interaction, we can see it is not statistically significant. This means there is no added benefit or disadvantage when adding an increase of one image and one video together.

| Variable | Value | Significance |
|---|---|---|
| Images | -279.1 | 0.44 |
| Videos | 1686.3 | 0.00018*** |
| Interaction | -1317.1 | 0.0057** |

When we change the images and videos variable to either **0 or 1+**, this shows a different interaction. Simply put, this interaction suggests that including both images and videos has an added disadvantage for the article's shares; however, this can be misleading. When we take a look back on the *effect of images by categories* table, there is a bias towards including just one or two images. These articles perform so poorly and at a large sample size that they skew the significance of the interaction term here. Thus, the result above should mainly be ignored.

## 3 Conclusion

As we finish our analysis, there are many takeaways to be had. Our linear regression model suggested top predictors include self-referenced articles average shares, average keyword average shares, and number of non-stop words. Because of their different weights, this means to reference articles with a high amount of shares (on average), use important keywords (on average), and limit the number of non-stop words with shorter sentences and good punctuation / grammar. Our better performing models suggested the LDA_02 is the most important variable, followed by max_negative_polarity. When we were doing variable selection for our models. We found that weekdays columns, number of images and number of videos contribute a little to the model. By considering the complexity of the model, it will be better to delete these columns. We also suggest considering whether it is the weekend because of the increasing MSE. These suggestions are all based on the current best model. If the distribution of the data changes, variable selection should be reconsidered.

For the day of the week, we noticed that Wednesday usually has the lowest number of people sharing the news, about 1879.79 shares on average. Saturday is better for sharing, with around 2760.20 shares. The difference in shares between any two days is not very big, usually less than 1000 shares. This means the day of the week might not significantly predict how popular the news will be. But if you have to choose a day to post news, it is better to avoid Wednesday and try the weekend, because weekends usually get more shares. To understand this better, we could use a method called 'Time Series Analysis'. This method looks at data collected regularly, like every day or week, to find patterns over time. We suggest collecting more detailed data (like the Date of Publication, Day of the Week, Time of Publication, Special Events or Holidays) to make this analysis better.

For polarization, we find that shares of global news are not impacted by polarization for non-viral articles and moderately polarizing articles perform the best when viral. We find that most articles are weakly to moderately polarizing and recommend experimenting and writing more polarizing articles to see if there is a relationship between highly polarizing articles and shares. We also find that highly polarizing sentences sprinkled in the article are a characteristic of highly shared articles. Subjectivity should not be maximized; articles should avoid being strongly subjective, but contain moderate amounts of subjectivity. We conclude that polarizing plays a small role in shares for articles and recommend to focus on moderately negative articles and moderately subjective articles to increase shares.

Images and videos both are important. In fact, both had a positive relationship, suggesting as you add an image or video, you should expect an increase in the number of shares (on average). It is not this simple when we take a deeper look. For images, articles including none or at least several performed the best. Therefore, we suggest to only include images if they add engagement

and quality information to the article. An image is not required in every article for it to do well. For videos, we found that articles with a couple or more perform well. Hence, adding a quality video in the article should help with engagement. There was too little data on articles with high amounts of images and videos to do in-depth analysis on. We suggest (if the images and videos add something significant to the article) a high amount might lead to better shares; take this information with a grain of salt though.

We hope this analysis has provided Mashable with sufficient evidence on how to increase the number of shares for world news articles. For more information on the analysis, the next section and attached files provide a more in-depth look at our process. (These include coding files, diagnostics, etc.)

# 4 Appendices

## 4.1 Linear Regression Diagnostics

Now, we must check for diagnostics. We will check to see if there are any multicollinearity issues first. After applying the VIF test, we can see there is not much of a multicollinearity effect. This is good, so we can say there are no collinearity issues to worry about.

*Top-5 Sqrt. VIFs*

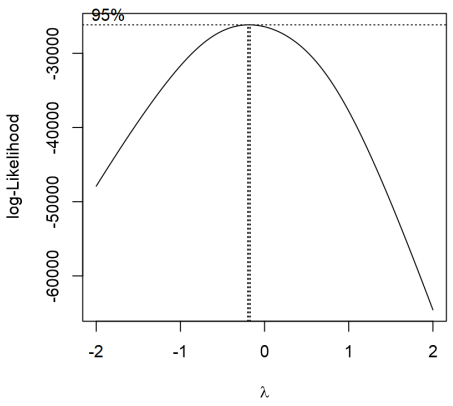
| n_non_stop_words | n_non_stop_unique_tokens | kw_avg_avg | kw_min_avg | num_hrefs |
|---|---|---|---|---|
| 1.56 | 1.53 | 1.30 | 1.22 | 1.16 |

As we look at the cook's distances, the top three are 0.36, 0.16, and 0.08. There are no data points exceeding 1. This means there are no highly influential points (HIPs) to note.

Next, we will check for constant variance. To do this, we will apply the Breusch-Pagan test which assumes constant variance. With a p-value ~ 0.00086 < .05 = α, we reject the null and conclude constant variance is not satisfied. We will try to fix this later.

Finally, we need to check for normality. To do so, we will use the Kolmogorov-Smirnov test which assumes normality. With a p-value ~ 0 < .05 = α, we reject the null and conclude normality is not satisfied.

Let us first try to fix the constant variance by doing a weighted least squares. Our weights will be defined as 1 / residuals^2. By doing such, we increase our p-value ~ 1 for the Breusch-Pagan test. This suggests that we fail to reject the null and conclude that constant variance is satisfied. With that being done, let us recheck the other diagnostics. Collinearity looks worse, but should be okay – top-3 have values of 3.83, 3.76, and 2.26. Also, we have two HIPs, since their cook's distances are above 1 (1042 & 3.14). On top of that, the Kolmogorov-Smirnov test fails again. With the interpretability falling, more diagnostics failing, and only a slight decrease in the RMSE (4006) we will not choose this model.




Perhaps we can fix the diagnostic issues with a box-cox transformation *on the right*. However, when we check for which lambda is optimal, the plot suggests no transformations. Thus, we cannot fix the constant variance and normality issues that popped up during our final model.

### 4.2 Grant's R Code

```
data=read.csv("OnlineNewsPopularity.csv") ## reading the data
newdat= subset(data,data$data_channel_is_world==1) ## filtering to only include world articles
newdat = subset(newdat, newdat$rate_positive_words!=0&newdat$rate_negative_words!=0) ## removing rows in data where rate positive and rate negative or both 0
newdat = newdat[,-c(1,2,14:18)] ## removing indicator columns, and the URL & indices
newdat[,c(1:11,13:24,33:54)] = scale(newdat[,c(1:11,13:24,33:54)]) ## scaling the continuous variables in order to have better prediction results


newdat = subset(newdat,newdat$shares<100000) ## removing the shares about 100,000 so we aren't heavily skewed
nushares = newdat$shares[(newdat$rate_positive_words>=.40) &(newdat$rate_positive_words<=.6)]
modpositiveshares = newdat$shares[(newdat$rate_positive_words>.6) &(newdat$rate_positive_words<.8)]
vpositiveshares = newdat$shares[newdat$rate_positive_words>=.8]
modnegativeshares = newdat$shares[(newdat$rate_positive_words<.4) &(newdat$rate_positive_words>.2)]
vnegativeshares = newdat$shares[newdat$rate_positive_words<=.20]
## created buckets for  positive word rates


nugs = newdat$shares[(newdat$global_subjectivity>=.40) &(newdat$global_subjectivity<=.6)]
modposgs = newdat$shares[(newdat$global_subjectivity>.6) &(newdat$global_subjectivity<.8)]
vposgs = newdat$shares[newdat$global_subjectivity>=.8]
modneggs = newdat$shares[(newdat$global_subjectivity<.4) &(newdat$global_subjectivity>.2)]
vneggs = newdat$shares[newdat$global_subjectivity<=.2]
## created buckets for  global subjectivity


means=
rbind(c(mean(vnegativeshares),mean(modnegativeshares),mean(nushares),mean(modpositiveshares),mean(vpositive shares)))
meansGS = rbind(c(mean(vneggs),mean(modneggs),mean(nugs),mean(modposgs),mean(vposgs)))
## calculated the mean of each of the buckets


means = rbind(c("Very Neg Avg","Mod Neg Avg","Neutral Avg","Mod Pos Avg","Very Pos Avg"),c("0.0-0.2","0.2-0.39","0.4-0.59","0.6-0.79","0.8-1.0"),round(means,0))
meansGS=rbind(c("Very Obj GS","Mod Obj GS","Neutral GS","Mod Sub GS","Very Sub GS"),c("0.0-0.2","0.2-0.39","0.4-0.59","0.6-0.79","0.8-1.0"),round(meansGS,0))
## constructed a table of the means with the labels of each bucket

rownames = c("Title","Rate","Avg Shares")
meanratetable = cbind(rownames,means)
meanGStable = cbind(rownames,meansGS)
## added another column to display the titles for the rows
```

### 4.3 Jun's Code (Weekdays)

[Table for average shares by weekday]

```
library(ggplot2)
library(dplyr)
library(tidyr)

data <- read.csv("OnlineNewsPopularity.csv")
 # Choose only the parts of the data where 'data_channel_is_world' equals 1. This means we are looking at world news only.
world_news <- data %>% filter(data_channel_is_world == 1)
# Transform the dataset: Combine all different weekday columns into one new column named 'weekday'.
av_day <- world_news %>%
  gather(key = "weekday", value = "value",
         weekday_is_monday, weekday_is_tuesday, weekday_is_wednesday,
         weekday_is_thursday, weekday_is_friday, weekday_is_saturday,
         weekday_is_sunday) %>%
 # Keep only the rows where the news was published on that specific weekday (indicated by the number 1).
  filter(value == 1) %>%
 # Change the 'weekday' column into a category type and arrange the days of the week in a specific order.
mutate(weekday = factor(weekday, levels =
c("weekday_is_monday","weekday_is_tuesday","weekday_is_wednesday", "weekday_is_thursday",
"weekday_is_friday", "weekday_is_saturday", "weekday_is_sunday"))) %>%
  group_by(weekday) %>%
# For each weekday, calculate the average number of shares and the total number of shares.
  summarize(average_shares = mean(shares), total_shares = sum(shares))
print(av_day)
```

[Plot for average shares by weekday]

```
#Define the order of weekdays starting from Monday to Sunday
week_level <- c("weekday_is_monday", "weekday_is_tuesday", "weekday_is_wednesday",
                "weekday_is_thursday", "weekday_is_friday", "weekday_is_saturday",
                "weekday_is_sunday")
# Change the 'weekday' column in 'av_day' to a type that has categories, using the order of days we listed earlier.
av_day$weekday <- factor(av_day$weekday, levels = week_level)
ggplot(av_day, aes(x = weekday, y = average_shares)) +
# Create bars on the graph to show the average number of shares for each day of the week.
  geom_bar(stat = "identity") +
  geom_text(aes(label = round(average_shares, 1)), vjust = -0.4, size = 4) +
# Make sure the y-axis (vertical line) starts at 0 and goes up to 30% more than the highest average share value.
  ylim(0, max(av_day$average_shares) * 1.3) +
# Make the graph look nice: set the title in the center, make the text bold and easy to read, and set margins.
  theme(plot.title = element_text(hjust = 0.5), axis.text.x = element_text(angle = 45, hjust = 1,size = 10, face =
"bold"),axis.text.y = element_text(face = "bold"),
  panel.background = element_rect(fill = "white",
  colour = "white"),plot.margin = margin(1, 1, 1, 1, "cm")) +
  labs(title = "Average Shares by Weekday", x = "Weekday", y = "Average Shares")
```

**4.4 Our Contributions**

Michael contributed to the polarity analysis and was responsible for the analysis done on avg_postive_polarity and avg_negative_polarity variables. Michael and Grant worked closely together to tackle the general direction and analysis of variables relevant to polarity. We both discussed issues related to modeling with polarity variables, issues related to data variables measuring incorrectly, and the relationship between polarity and shares. R code can be found in "MichaelHuangRCode.R".

Sam built the linear regression model and completed the images/videos analysis. Sam also participated in general conversations and decisions (e.g. how to clean the code). Coding file is listed as "sam-burch-project.R".

Grant contributed to the polarity analysis including the rate positive polarity and global sentiment tables. Also produced the introduction of the dataset and research goals/methods portion of the report. My code can be found in section 4.2 labeled "Grant's R Code".

Wenxuan constructed the GBM, RF, XGBoosting, and metamodeling. Also tuned the hyperparameter and did the variable selection. Wenxuan also participated in weekday stuff analysis. Code file is "Wenxuan Gu 443 CODE".

Seok Jun conducted research and analysis on weekday shares, including time series and further analysis. He also executed the necessary coding for this investigation (4.3 & separate file).