# ORIE 4741 Project Report

Mina Huh (mh859), Matthew Danbury (mjd358)

December 5, 2021

## 1 Background and Motivation

Throughout the 20th century, the business model for journalistic publications consisted primarily of securing advertising revenue, with actual sales of physical papers and magazines accounting for a much smaller portion of a publication's profitability. In recent decades, the ways in which people stay informed have been transformed beyond recognition by the advent of internet search engines, mobile devices, and social media platforms with algorithmically curated newsfeeds. It is somewhat unclear what specific roles large journalistic publications will come to fill in this rapidly evolving information ecosystem, but what is certain is that their old business models are no longer viable. With the rise of digital advertising, we have seen a proportionate decline in the revenue of print advertising that underpinned these old business models (Figure 1), and in this new digital ad market, journalistic publications must compete with every other high-traffic website for ad revenue. Moreover, current research shows that a majority of Americans get their news from social media [1]. Consequently a significant portion of any news outlet's advertising revenue hinges on traffic directed to their articles from social media sites. While the precise details of any particular platform's newsfeed algorithm are unknown, it is no secret that

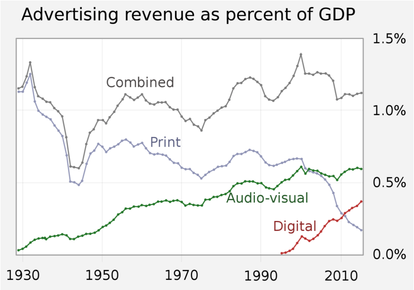

Figure 1: Trends in Advertising Revenue. Figure from 'Measuring the "Free" Digital Economy Within the GDP and Productivity Accounts' https://www.bea.gov/index.php/system/files/papers/WP2017-9.pdf

posts with higher engagement in the form of reactions, comments, and shares reach more users on these platforms. It is therefore crucial to be able to understand and predict how a particular piece of content will perform on a social media platform in terms of these metrics. The authors believe that being able to predict the popularity of an article before publication will help journalistic publications to both maintain profitability in the short term, and better understand how readers engage with news media online, which could in turn inform decisions on which direction to take the publication in in the long term.

## 2 Dataset

For this project we are using UCI's *Online News Popularity Data set* [2] that is compromised of 60 features in addition to the number of shares which we will use as our metric for an article's popularity. We chose this data set as it focuses on one news platform with attributes that could be tracked across different news platforms. This is important as we found that data sets containing multiple different news platforms were not consistent across different platforms when it came to what data is available and the formatting. In addition, while number of shares might not fully represent an article's popularity, we wanted to focus on a simple engagement metric that is available on most online news platforms for the sake of making our model not platform specific.

Additionally, to keep the possibility of exploring different models open we decided to experiment with different representations of the engagement metric. As the number of shares is numerical it can be predicted with a regression model as it is. In the case of classification as we need labels, we decided on dividing the entries for number of shares into quartiles, assigning the label "low engagement" if the number of shares of an article was below 1000, the label "average engagement" if the number of shares fell between 1000 and 3000, and the "high engagement" if the number of shares exceeded 3000. With this decided, we next focus on what sort of features can be utilized to predict engagement/number of shares.

In some of the earlier data sets we considered we found missing or nonsensical values in a significant portion of the full data set (around 28 percent). To check for this in the UCI set, we first started examining our data by using .describe() to obtain some general statistics. Through doing so we found unreasonable minimum/maximum values for 'n tokens content', 'n unique tokens', 'kw avg min', 'kw min avg', and 'kw min min'. As 'n tokens content' refers to the number of words in the content of the article, we felt that negative values and the value of 0 did not make sense and filtered them out. 'kw avg min', 'kw min avg', and 'kw min min' all refer to the number of shares for the shares associated with an article's keywords so similarly will need to filter out negative values. In the case of 'n unique tokens' we noticed a max value of 701 even though the field is measuring the rate of unique words in the content of the article. After filtering this max value out we checked the histogram and found that the remaining values correctly fell between 0 and 1.

While doing the filtering we also realized that the names of each column had an extra space at the front so we used a replace function to modify each column name to not have the extra spacing. After applying these filters, our data set still maintained most of its data and was reduced to 38462 records, keeping around 97 percent of the initial data set.

After performing a 75-25 train-test split, we made a histogram of the shares in the training set. The first thing we noticed was that on a linear scale, the distribution is very skewed to the right. However, after applying a log scale we can see that it more closely resembles a normal distribution.

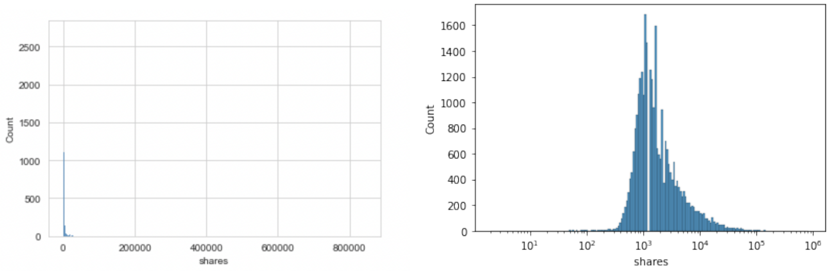

Figure 2: Linear vs Log Scale Histograms

# 3 Model Selection and Hyper-Parameter Tuning

Going into this project, we saw predicting the number of shares an article receives as a regression task. After partitioning our data 75-25 into a training set and a test set, we attempted to fit a linear least squares regression model to our data. This model struggled to predict the number of shares for examples that had particularly high or particularly low numbers of shares. For us, this was a deal-breaker as understanding which articles perform unusually poorly or unusually well in terms of engagement is likely what a journalistic publication would be most interested in knowing. We also recognized that predicting the absolute number of shares is less important than predicting whether a given article performs relatively good or bad compared to other articles. These insights inspired us to shift directions and think of this as a classification problem instead of a regression one.

To do this, we had to transform our training and test set labels from numbers of shares (integers) into relative performance categories (ordinals). We did this at two levels of granularity: the first simply being whether the number of shares was above or below the mean number of shares in the training set (a binary classification problem), the second whether the number of shares fell into the first quartile, the middle two quartiles, or the upper quartile of number of shares in the training set (a multi-class classficication problem). It is worth emphasizing that none of the test examples or their summary statistics were used in deriving these boundaries so as to prevent data leakage in our model validation.

To proceed, we further partitioned the training set 80-20 to have a validation set for tuning model parameters. Then, for each classification problem, we trained a random forest classifier and a gradient boosted decision tree classifier and compared results. We gravitated towards these tree based models both because their non-linearity could hopefully model some of the non-linearity inherent in the data-generating process, and because the ability to extract feature importance from the final model would provide greater interpretability.

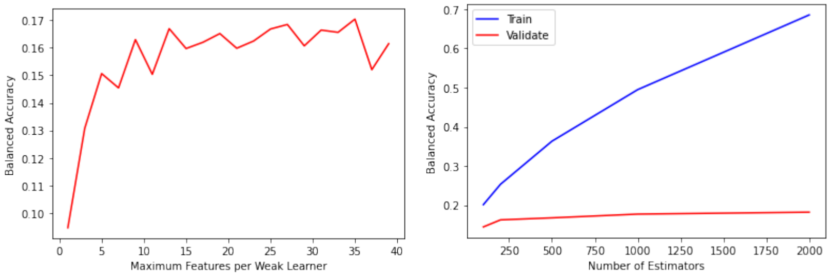

Figure 3: Hyperparameter tuning for the multiclass classification problem.

For the binary classification problem, tuning hyperparameters did relatively little to improve results. For multiclass classification some initial exploration revealed that for the random forest classifier, the hyperparameter which most significantly influenced the model's performance was the maximum number of features per weak learner. To determine the optimal value, we performed a grid search, the results of which are shown in Figure 3. For the gradient boosted decision tree ensemble, we used a similar grid search on the number of estimators (Figure 3) and determined that the marginal benefit to using more estimators tapers off after about 200 estimators, and any more just lead to over fitting.

# 4 Results and Discussion

For the binary classification problem, the random forest classifier and the gradient boosted tree ensemble classifier achieved the same mean accuracy of 67 percent on the test set. In other words, two thirds of the time, our classifier will be able to accurately determine whether an article will get more or less shares than ”the average” article from the publication. This is an accuracy we would feel comfortable using in production to influence decisions at a journalistic publication.

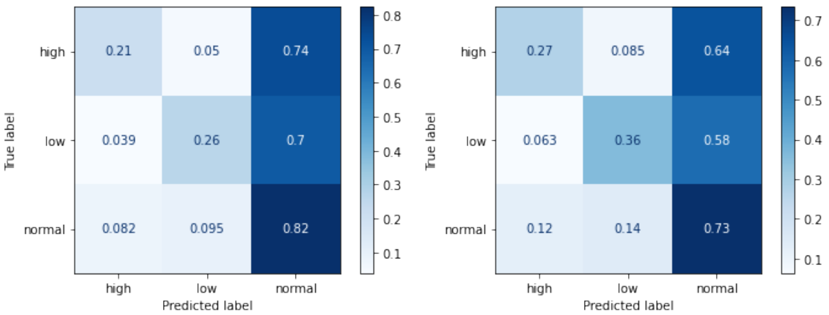

Figure 4: The confusion matrices for the random forest multiclass classifier (left) and gradient boosted tree ensemble multiclass classifier (right)

Unfortunately, if we want to know ”how many more or less shares than average?” we are much less confident. When evaluating the performance of multiclass classification, one must be cognizant of the fact that mean accuracy is a misleading statistic when one class is more dominant than another, as the classifier could achieve ”good” performance by simply assigning every example it encounters the dominant label. It is considered best practice to use confusion matrices [3], which show a percentage breakdown of the predicted labels assigned to the examples of each true label. The confusion matrices for our optimal multiclass classifiers are shown in Figure 4. While the gradient boosted tree ensemble outperforms the random forest classifier in in this case, it only manages to correctly identify 27 percent of the examples whose number of shares lie in the upper most quartile, and 36 percent of the examples whose number of shares lie in the lower most quartile. With such unreliable performance, we would not feel comfortable recommending that a journalistic publication use this model in production.

While we may only be able to make very coarse predictions confidently, we are able to understand what factors go into these predictions by determining feature importances in our model using mean decrease in impurity. In Figure 5, we show a bar graph of the mean decrease in impurity for each feature our model uses:

In this figure we can see that ’kw avg avg’ has the highest importance and is followed by ’kw max avg’ and ’kw avg max’. These features all relate to the keywords in an article, implying that the keywords in an article play a big part in predicting the popularity of an article. Another interesting finding is that our model did not assign much importance to the specific day of the week that the article was published although the distinction between weekday and weekend did hold some importance. We can also evaluate how the model made splits on polarity/sentiment and the token counts. In the case of polarity/sentiment related features, the model placed more importance on the average polarity of positive and negative words than the min/max and sentiment polarities.

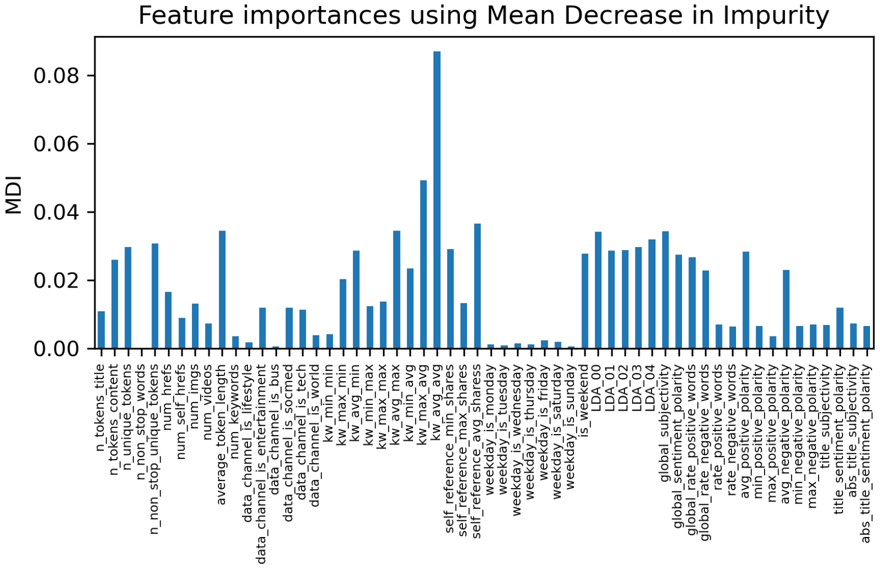

Figure 5: Caption

# 5 Conclusion and Outlook

In summary, we have demonstrated that it can be possible to predict with moderate accuracy whether an article will receive more or less engagement than other articles from a given publication. Since content engagement affects the baseline of most journalistic publications, those with the resources to invest in developing these data technologies could find it advantageous to do so. That said, we want to end on a word of caution about potential negative consequences of going this route.

Throughout this project we considered how fairness might apply to our model and the potential impact that predicting popularity in news could have. While the project could help in aiding one's understanding of the factors that contribute to popularity, there is also the potential to create a pernicious feedback cycle in the way that almost all predictions have the potential to do. Especially within the increasingly competitive attention economy such predictive models can lead to methods of engagement optimization that reinforce existing expectations rather than innovate. Already, one often sees critique about the way content creation is fixated on engagement optimization over the content itself. In this sense we can see how applying such predictive models could negatively impact or restrict what news gets published and given a platform. Because of this, it is important to make the distinction between the popularity we are predicting and the quality of an article.

Another consideration we had was regarding the data set we chose which focuses specifically on Mashable, a very well known digital media and news platform. We want to make sure to take into account how the model was trained on articles from a high profile company, and as a result may not work well for other news platforms that may have different audience demographics or different levels of viewership.

Finally, when thinking about ways that we might further develop our model we had a few ideas. In an attempt to find additionally useful features we had spent some time experimenting with the Beautiful Soup package to parse through the HTML given by the article URL. In theory, we wanted to obtain the textual data for article content/title to see how they could play a role in predicting popularity as concepts such as ”clickbait” and wording definitely play a role in attracting engagement. This could open up the possibility of also integrating NLP techniques for further exploration. Upon attempting this on such a large data set we found the time it took to scrape each web page for specific information was not surprisingly way too slow for the scope of this project. However, seeing what improvements could be had using more sophisticated NLP techniques could still be an interesting line of inquiry. An additional step could even examine layout, and media such as images and video also included within the page for further exploration of how information supplementary to the article reading experience might play a role in popularity.

Another area of interest for further pursuit would be in comparing the predictability of different metrics and different ways of defining popularity and engagement. As sharing an article is a relatively involved interaction compared to the simple act of saving/liking/disliking/commenting, it could be interesting to examine how different it is from those other engagement metrics. Especially in regards to articles of positive vs negative sentiment, the different engagement metrics can represent different intentions and not align with each other. For example, the number of comments can rack up quite quickly under a controversial news article but the number of likes might remain low due to its controversial nature. Comparing the predictability of such metrics and then combining the predictions could make for an interesting project in a more nuanced prediction of an article's reception.

# 6 References

[1]https://www.forbes.com/sites/petersuciu/2019/10/11/more-americans-are- getting-their-news-from-social-media/?sh=1704a9f93e17

[2]https://archive.ics.uci.edu/ml/datasets/Online+News+Popularity

[3] https://scikit-learn.org/stable/modules/generated/sklearn.metrics. ConfusionMatrixDisplay.html?highlight=confusion%20matrix#sklearn.metrics. ConfusionMatrixDisplay