Predicting and Creating Popular News Articles to Achieve Social Impact and Generate Business Value

GitHub Link: https://github.com/woshiwilson/code15

Abstract

The proliferation in the number of digital news sources and content has and will continue to grow unabated in many years to come. How can the consumers and suppliers in this sector with such huge volume and velocity of incoming information select the most important news articles to read and publish respectively? In this study, we build a predictive model to find the articles that are the most popular, understand why they are so, and propose recommendations to all stakeholders within this sector. We discuss the tangible benefits of our solution in creating social impact and generating business value, as well as the limitations of our study.

1. Project Description

Worldwide digital media revenue and number of users have been increasing annually and these trends are forecasted to continue in years to come (see Figure 1).

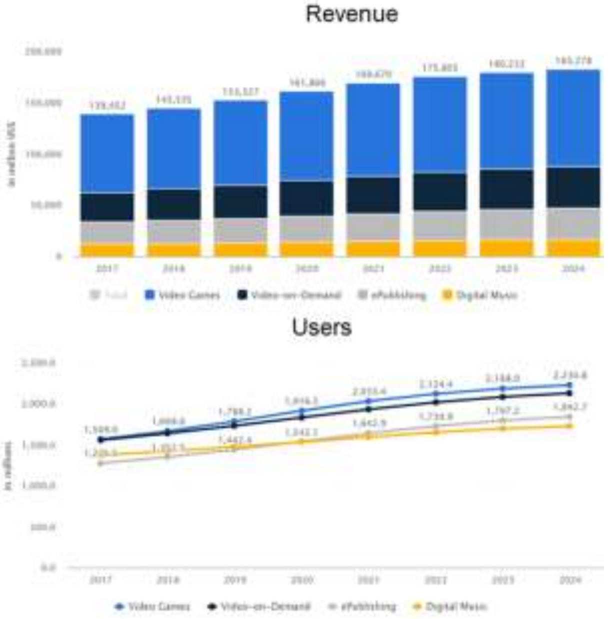

Figure 1. Top: Digital media revenue is set to increase from over US$150 billion in 2019 to over US$180 billion within 5 years. Bottom: The number of users is set to increase with a similar trend over the next 5 years.

Following this global digital transformation, an increasing number of news readers turn to digital means to consume information. Within the United States alone, digital news sources experience an unabated growing traffic of above 20 million unique visits annually (see Figure 2).

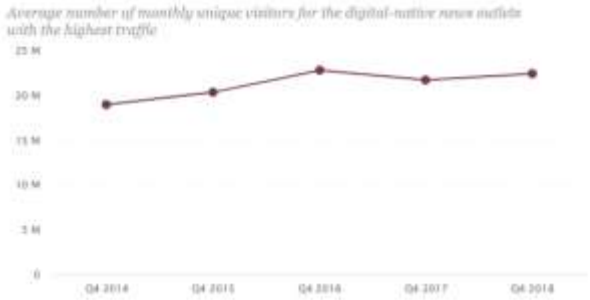

Figure 2. Average number of monthly unique visitors for the digital native news outlets with the highest traffic within the United States.

1.1 Problem Statement

The growing digital news market brings about an unprecedented abundance of content that is practically impossible for readers to consume in totality without cognitive overload and fatigue. However, this also presents several opportunities for exploration on how best to overcome the challenges and exploit the market potential to generate the greatest social impact and business value for all stakeholders. We propose to study one such area and our problem statement may be defined as such: To test and evaluate how popular a given news article will be and uncover the underlying factors driving that popularity for different groups of readers.

2. Study Objectives and Business Impact

From the above problem statement, our study aims to build a predictive model that will support the evaluation of how well an article will be received. It will empower users to only put out only the more impactful articles in a bid to ensure that the articles that do not garner much interest do not inundate readers and thus subsequently leading them to cognitive overload and fatigue. The predictors used in the model consists of the news articles' content and attributes. The predicted variable is the number of shares of the news articles, which acts as a measurement of the concept of "popularity". This variable is a good proxy to understanding whether a news article is gaining traction because it is reasonably assumed that readers will only share what they are interested in reading.

Two groups of users are expected to accrue benefits from our study: (1) The stakeholders in content creation and (2) the stakeholders who select the content to be put out.

2.1 Group 1: Journalists

The first group consists of people like journalists who work on creating news articles. Our model can help this group to gain a "preemptive" understanding of how their news articles will be perceived by the readers before they are even published. This is because our model analyzes the different aspects of the news article to generate actionable insights on how to structure a news article to be more engaging to the readers. Such insights serve as helpful guidelines with which they can adopt to achieve higher popularity of their news articles.

2.2 Group 2: News Agencies/Websites

The second group comprises news agencies/websites which need to handle numerous articles. With rapid velocity and volume of article submissions for publication, the screening process of selecting high-value articles that can reach the widest audience becomes arduous and subjective. The efficiency and efficacy of such screening is called into question and is difficult to assess how successful they have been. This is where our model comes into play. It can be useful for news agencies/websites to predict the popularity of the news articles to aid them in filtering out the articles which are not well-written and structured. Such automated processes can efficiently and effectively improve the quality of the articles posted on the news agencies/website so that they can achieve higher popularity among readers, capture market power from competitors, and generate higher sales and revenue.

3. Data Scraping and Dataset Exploration

The dataset which we obtained was sourced from the UCI Machine Learning Repository. The dataset consists of the label (i.e. the number of shares for each news article) as well as 60 features for over 30,000 individual news articles from 'Mashable', which is a digital media website. For conciseness, we grouped these 60 features into 14 groups (see Table 1).

Table 1. Groups of features within the UCI Machine Learning Repository.

S/N	Feature Group	Description
1	Article URL	URL of the article
2	Published date	Days since article publication
3	Word counts	Frequency and rate of unique words
4	References	Number of outlinks
5	Multi-media	Number of images and videos
6	Word attributes	Attributes of words
7	Article type	Article category
8	Keyword count	Frequency of keywords
9	Weekday/ Weekend	Day of article publication
10	LDA topics	Closeness of article to a topic
11	Global sentiment data	Positivity of global text
12	Positive/ Negative words	Frequency of positive and negative words; extent of sentiment orientation
13	Title Sentiment data	Extent of the article's title subjectivity and sentiment orientation
14	Shares	Number of shares (label)

It is from this collection of features that we selected what we deemed as relevant for our prediction task and we will describe the final list of features later. More importantly, this collection of features includes the URL for each individual article and this opens up opportunities for more feature engineering efforts. Hence, we scrapped the websites of 'Mashable' for the original corpus and created more features by applying natural language processing (NLP) techniques.

To scrape the data, we used the scrapy library and scraped it using a spider. This was done using css and xpath identification of the information we needed. They were subsequently exported into a .csv file. While the main chunk of the data needed to be scraped was the text content within the articles themselves, we also identified other important pieces of information that were not in the original dataset, such as the date and time that the article was published, the title of the article, and the tags that the article was classified under. These were all similarly scraped at the same time using the same spider.

4. Dataset Pre-Processing

There were two sets of unstructured text data which required data pre-processing and they are (1) the article content and (2) the article title, both of which were scrapped as abovementioned. The following steps were implemented to extract meaningful information after they were separately tokenised (see Table 2).

Table 2. Data pre-processing steps carried out on the article content and the article title.

STEP NAME	DESCRIPTION
BEAUTIFULSOUP EXTRACTION	EXTRACT MEANINGFUL INFORMATION FROM HTML FORMAT
REMOVE NON-ASCII CHARACTERS	REMOVE CHARACTERS THAT ARE NOT IN ASCII TO REDUCE NOISE
REMOVE PUNCTUATIONS	REDUCE IMPACT ON RESULTS DUE TO PUNCTUATIONS
REMOVE STOPWORDS	FOCUS ONLY ON MORE MEANINGFUL WORDS
STEM WORDS	STEM WORDS TO BASE TERM TO LET THE MODEL NOT BE AFFECTED BY THE SAME WORDS THAT ARE IN DIFFERENT FORMS

5. Feature Transformation: Unstructured Data

5.1 Bag-of-Words

In Bag-of-Words model, the text data is represented through the frequency of word occurrences within an article.

5.2 TF-IDF

In Term Frequency – Inverse Document Frequency (TF-IDF) model, the text data is represented through the importance of a word to its article in the collection of all articles.

5.3 Word Embedding

While Bag-of-Words and TF-IDF feature extraction methods are useful in many situations, they do not capture the semantics behind these words. Our dataset deals with news articles and readers are usually attracted to exciting and engaging titles and content. This prompted us to feature extract with word embeddings to ensure that the vector values will capture as much of the linguistic meaning of the words as possible.

In our study, we adopted the use of Word2Vec to vectorize the titles and content corpus that the neural networks can understand. Word2Vec essentially detects word similarities mathematically and groups vectors of similar words together in a vector space. This allows the extracted vector to encompass some of the semantics, which is crucial for the identification of a successful and widely shared news article.

6. Feature Transformation: Create New Features

We also created new features we believe are relevant predictors of news article popularity and we describe them next.

6.1 Latent and Temporal Interest Features

Latent interest refers to the intrinsic, underlying interest in the topics that the readers of the news agency/website, which is ‘Mashable’ in this case, already have. ‘Mashable’ may have a niche in certain topics, which in turn draws these readers who are already interested in these topics to its website. Hence, we want to determine how similar the topics of a particular news article are relative to this latent interest. We believe that this similarity correlates positively with news article popularity and can act as a predictor in our model.

In contrast to the more pervading and long-term nature of latent interest, temporal interest refers to the short-lived interest for certain topics. Such interests behave like a fad with intense enthusiasm that dies off after a short passage of time. The high relevance for the topics of the day influences user’ sharing behaviour.. We believe that this short-term interest correlates positively with news article popularity and can also act as another predictor.

For latent interest, we took the tags of all the news articles and performed a frequency comparison, where each tag was assigned its frequency of appearance over all news articles. Then, we took the average number of appearances of the tag for each article and tagged that number to the article. The higher this number was, the more common the tags were on the website. This was meant to create a distinction between articles that had many similar tags over time and articles that were somewhat novel to the platform.

For temporal interest, we attempted to use Google’s search trend data for the week before the article was published and took the average of the Google search index using the tags as search keywords. However, while this worked well, it was limited by the throttling of the Google results as the api used (pytrends) was only able to obtain 100 lines of results at one go before returning a timeout for the rest of the results. Unable to get around this, we then left this portion out in the interest of time and for this study. However, if a workaround could be found for this constraint, it is believed that this feature would be one of the more important features.

6.2 Readability Features

Readability refers to how easy an article is to read, understand, and respond to. We believe that readability correlates positively with news article popularity. It entails the use of simple language, active verbs, and short sentences. To operationalise the concept of "readability", the following features were created from the article content to measure it:

- Number of verbs
- Proportion of tokens that are verbs
- Average sentence length
- Average word length
- Average number of word syllabus

It should be noted that in the process of creating these features, the data pre-processing carried out on them was somewhat different in the earlier section of this report. This is because the previous data pre-processing process described removed the information necessary to derive these features. Here, only the steps of removing HTML tags and converting all characters to lowercase were carried out after tokenisation.

The number and proportion of verbs were computed after parts-of-speech tagging was conducted using the nltk package. The average sentence length was computed by taking '.', '?', and '!' as terminals of the sentences. The average word length is the average number of characters in each word. Finally, a heuristic based on the English language was used to determine the average number of word syllabus.

6.3 Time of Article Publication Features

The number of shares is also affected by the temporal aspects of the publication. For example, an article published on a weekend night may fetch higher viewership and shares than one published during office hours on a weekday. To control for this, we created temporal features and incorporated them in our model:

- Hour of day
- Day of week
- Week of year

6.4 Summary of Final Features for Modeling

In total, our model incorporated the vectorized title, vectorized content, and 14 other groups of features (i.e. total of 16 features), consisting of features processed from scrapped data, from created new features, and from the original dataset:

- Vectorized content
- Vectorized title
- Time of publication
- Readability of content
- Latent interest
- Number of links in article
- Number of images
- Number of videos
- Closeness of LDA topic
- Content subjectivity
- Content sentiment polarity
- Rate of positive/negative words in content
- Average polarity of positive/negative words
- Title subjectivity
- Title polarity

7. Machine Learning Models

Several supervised machine learning models were selected for the regression task of predicting the number of article shares. They were selected on the basis of their varied algorithmic approaches (linear, non-linear, neural network) in attempt to find one that is most suitable. They are:

- Linear Regression
- Stochastic Gradient Descent
- Light GBM
- Support Vector Machine
- Neural Network

Each machine learning model was then paired with a feature extraction method (see Table 3).

Table 3. Pairing feature extraction methods and machine learning models.

Model No.	Feature Extraction Method	Machine Learning Model
1	Bag-of-words	Linear Regression
2	TF-IDF	Linear Regression
3	Bag-of-words	Stochastic Gradient Descent
4	TF-IDF	Stochastic Gradient Descent

5	BAG-OF-WORDS	LIGHT GBM
6	TF-IDF	LIGHT GBM
7	BAG-OF-WORDS	SUPPORT VECTOR MACHINE
8	TF-IDF	SUPPORT VECTOR MACHINE
9	WORD EMBEDDING	NEURAL NETWORK

To facilitate the joining of the matrices resulting from the vectorisation of article content and article title with both the features of the original dataset as well as newly created features, a pipeline was built. The pipeline passed the data (80-20 split into train and test sets) through a 3-fold cross-validation sequence in randomised hyperparameter tuning, where applicable. This is shown in the diagram below, where the (1) article content was retrieved and vectorised, (2) article title was retrieved and vectorised, (3) original features from the dataset, and (4) newly created features were joined into the final matrix.

To prevent data leakage during the cross validation process, we created a pipeline to pass the raw features to the model and calculate the validation score. This will allow the vectorization to occur within each fold of cross-validation to prevent data leakage.

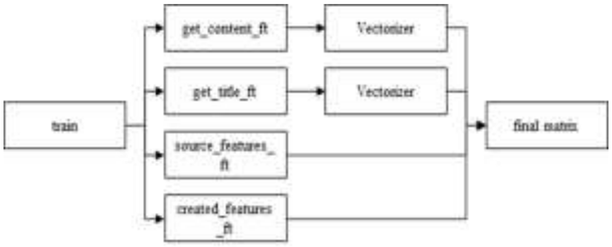

Figure 3. Diagram of the pipeline built on how various features were joined.

We also built a neural network model with 2 hidden layers using Long Short Term Memory (LSTM) and dense layers. The corpus of pre-processed title and content were vectorized through word embedding technique (Word2Vec). 14 other features were also put through 2 dense hidden layers before concatenating them for a joint output.

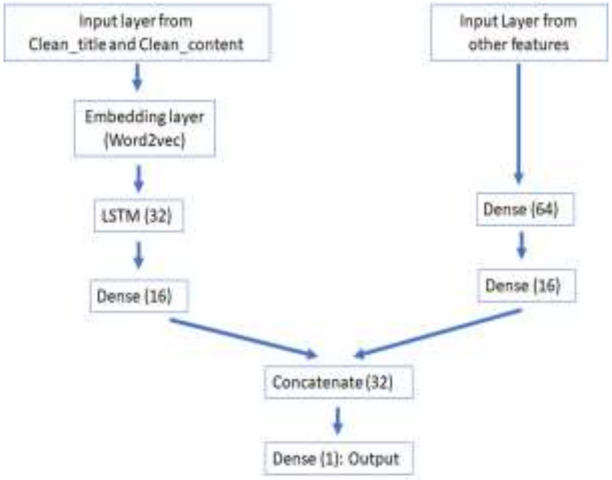
Input layer from Clean_title and Clean_content
Input Layer from other features
Embedding layer (Word2vec)
LSTM (32)
Dense (64)
Dense (16)
Dense (16)
Concatenate (32)
Dense (1): Output

Figure 4. Illustration of the Neural Network Layers (2 hidden layers + word embedding layer).

8. Models' Results

8.1 Model Quality

The metrics for model evaluation are mean absolute error (MAE) and root mean squared error (RMSE). MAE measures the magnitude of error (i.e. difference between the predicted and true number of shares) and RMSE measures the square root of the average of squared errors. RMSE punishes large deviations more and hence may be taken as a more conservative measure, but is susceptible to outliers, which itself can be challenging to define. In general, we sought alignment between these two metrics in assessing model quality. The results are shown below (see Table 4).

Table 4. Evaluation metrics and results of the prediction models tested.

MODEL NO.	MEAN ABSOLUTE ERROR (MAE)	ROOT MEAN SQUARED ERROR (RMSE)
1	6833.4	13243.4
2	6730.4	13028.0
3	2277.0	10518.9
4	2269.1	10526.3
5	3050.6	10401.5
6	3036.5	10373.3
7	2338.6	10468.2
8	2339.0	10468.0
9	**3382.9**	**7036.6**

Linear regression models (models 1 and 2) are unable to capture the complexity and non-linearity of a text analysis and performed the worst in both MAE and RMSE. Other models (models 3-8) performed similarly on RMSE but scored differently on MAE (2269.1 to 3050.6).

On balance, the neural network model (model 9) with word embedding feature extraction performed relatively better than the other models. While the MAE for some other models (i.e. models 3-8) outperforms the LSTM recurrent neural networks model slightly, their RMSE are significantly worse off. This is congruent to the popular belief that LSTM recurrent neural networks are suitable for text analysis and explains the reason why we chose to explore this model for the analysis of a large online text corpus.

Aside from the model chosen, word embedding feature extraction allowed the neural network model to retain the semantics of the articles in the features. This facilitated the model to more accurately pick out the articles that have popular content that will be widely shared.

8.2 Important Predictors

For better interpretability, we also ran further models to understand the importance variables. This time, we ran the number of news article shares against the 14 numerical features excluding the vectorised content and the vectorised title.

The first model used was a simple linear regression and the features that were statistically significant (p < .05) and marginally statistically significant (p < .1) were compared (see Table 5).

Table 5. Summary of the selected important predictors from a linear regression model.

Selected Feature	Coefficient	Statistical Significance
number of outlinks	-35.1	^
Number of images	+36.3	*
Global subjectivity	+4387.4	*
Max. negative polarity	-2152.9	^
Title subjectivity	+728.6	^
title sentiment polarity	+1170.6	*
Day of week (transformed 1)	-198.2	^
Day of week (transformed 2)	+162.2	^
Week of Year (Transformed 1)	+174.0	^
Week of Year (Transformed 2)	+259.6	*

^ $p < .1$, * $p < .05$.

The second model used was a Light GBM model and the feature importance plot was generated (see Figure 5).

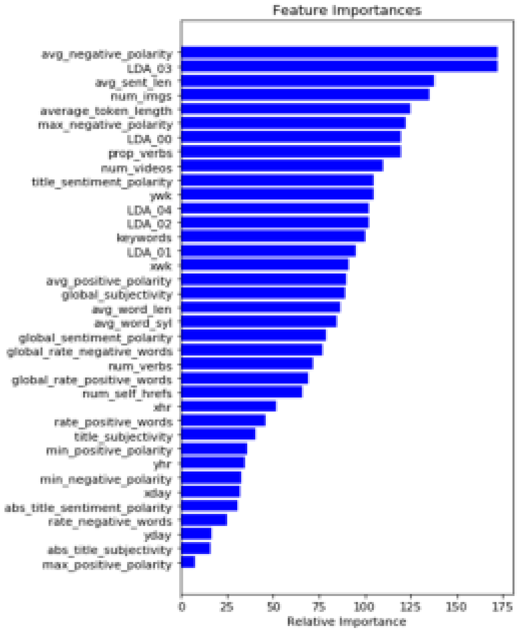

Figure 5. Important features from a Light GBM model.

Taken together, the relevant findings are that opinion content are more popular than factual content (‘global subjectivity’) and readers are especially influenced by sentiment polarity (‘abs title sentiment polarity’, ‘avg negative polarity’, ‘max negative polarity’). We also found that the week in which the article was published (‘xwk’, ‘ywk’) and latent interest (‘keywords’) were moderately important, while readability features (‘avg sent len’, ‘average token length’, ‘prop verbs’) stood out as among the most important features. The number of images and the number of videos were important too, and it can be argued that they play the same role as the concept of “readability” in that they assist in the efficient absorption of information by the readers.

9. Key Takeaways / Business Value

9.1 Tips to Popularise Online Articles

With the results, some key takeaways of how to make an online news article popular and widely shared are as follows:

9.1.1 VISUALS AND READABILITY

Audiences appreciate visuals and readability. It is recommended that the number of images and videos to be increased for stronger visual impact. Shorter sentences and words also improve readability of readers.

9.1.2 NEGATIVE OPINIONS

Articles that convey opinions with a negative slant generate more buzz. Titles that are negative and subjective are more popular among readers. Articles that carry negative sentiments are generally more widely shared than positive ones.

9.1.3 UNIQUE OPINIONS

Unique articles attract attention. Unique articles (associated with a lower latent interest index) are more eye-catching and shared. While readers may not associate a platform with a certain topic, this novelty does not prevent them from sharing it.

9.1.4 DISTRACTIONS

Distractions such as outlinks affect the likelihood of shares. Reducing the number of outlinks to other articles will increase the likelihood of the current article getting shared.

9.2 Business Value

We have previously identified two different groups of stakeholders who will benefit from our study. This benefit is tangible and not just knowledge for curiosity's sake.

9.2.1 NEWS/MEDIA PLATFORMS

In a day and age where information and articles are abundant, there are news fatigue and media fatigue. Both are the consequence of a reader or a user being bombarded with too much information and this leads to a sharp loss of interest in the topic matter or platform. Essentially, this means that an essential strategy for news and media sites would be to put out one great article that gains traction, rather than peppering users with many similar articles as that would result in that one good article being obscured by many other similar ones. One definite use case of our predictive model would be as the first round article screening. In other words, our model can be used to identify news articles with the highest chance of achieving high popularity and the highest potential for viral sharing. This is extremely important for companies in this industry as views and shares directly translate to sales and revenue of the platform.

The business use case of this varies from platform to platform. For example, for content sites that rely on user generated submissions and uploads, this first-cut creates an automatic filtering, using past data to create a predictive forecast on whether the article will become popular. This reduces the workload of editors and reduces expenses of the company because they will be able to efficiently and swiftly narrow down to only the articles with the highest potential of achieving popularity. This also puts an objective function on such potential, which is not easy to do in an industry where there can be much subjectivity. This objective function on subjectivity also reduces the reliance on talent when it comes to human resources because the variance and thus margin of error after a standardized filter such as this model will be reduced. This means that there is less risk in hiring an inexperienced editor as the chances of him or her disproportionately and negatively affecting the popularity of news articles will decrease with our model in place to reduce subjective judgments.

For other platforms that rely on a group of inhouse writers rather than relying on user-generated submissions, such as newspaper opinion pieces, our model could help in the optimization of what has already been submitted. For example, as there are various features such as the number of words and even the tags attached to the article, minor tweaks could be made to already edited articles right before publication, and run through this model to validate if the tweaks would be able to optimize what was already written. This allows the company to do more with what they already have.

9.2.2 WRITERS

The other group of people who stand to benefit are the writers themselves. Our model gives them an opportunity to obtain feedback or compare potential articles even before the articles go 'live' on the website. This allows them to make tweaks and to optimize their articles for improved audience reception, thus stretching the potential value of the news article. This is especially useful as it not only gives writers an objective way to view their usually subjective pieces, but this also gives them a chance to do a "what-if" as this has not been previously possible.

In the past, if writers aimed to isolate the best way to write articles by tweaking their style of writing or various elements, the results that they would have seen would not be measurable as each writing piece is so varied. While the writer believes that he or she has written something similar and changed one or two features between piece to piece to see what works and what does not, it is inevitable that the nature of the article or even the time of article upload will be different, thus the basis of comparison would be largely flawed. Our model allows for the control of many of these variables as two of the exact same piece except for intended edits can be tested and compared to find out what the impact is on the piece's popularity.

10. Limitations

We have also identified a few limitations with our study.

Firstly, the dataset used (with articles from 'Mashable') may have already drawn a certain crowd of followers and readers, and thus our model runs the risk of only being generalizable to ‘Mashable’ articles. However, we believe that these features are definitely still relevant even in other articles. There are two ways to cope with this. One way is to ensure that this model stays relevant even for other platforms would be for us to rerun our model with a larger and more inclusive dataset to obtain a more generalized model. The other way is to use the same method to obtain a separate model for a dataset comprising only articles coming from that particular platform that we are looking to explore.

Secondly, while we used follow-up models in a bid to interpret the factors and their effects on the number of news article shares, this is more of a proxy explanation. This is because the complexity of our models implies that we are unable to actually isolate each effect in the original predictive model because the computations are largely ‘a black box’. Hence, the proxy explanations should be taken with care because they do not completely explain the model and the factor interactions.

Lastly, we acknowledge that our model may be more suitable for use for opinion pieces or pieces meant to generate discussion. If the purpose of journalism is to be an impartial source of information, the actions taken to amend the news article for the sake of the number of shares, such as deliberate sensationalisation of the news, would be an inadvertent negative consequence from the use of our model.

References

K. Fernandes, P. Vinagre, and P. Cortez. (2015, May 31). *Online News Popularity Data Set*. UCI Machine Learning Repository. Retrieved from http://archive.ics.uci.edu/ml/datasets/Online+News+Popularity