**Online News Articles Popularity Prediction**

Manisha Therupalli, Dheeraj Makineedi

 2023-4-28

**1.Problem Definition:**

To predict the number of shares of a news article based on its textual content, category, publication date, and other attributes. How article features effects the popularity of the article.

**2.Motivation:**

In the online news industry, content providers or advertisers benefit greatly from the ability to accurately predict the popularity of content before it is published. Thus, it is interesting and meaningful to use machine learning techniques to predict the popularity of these articles. Popularity of news depends upon various features like usage of different keywords, relevance to a trending topic, perhaps even the day the article is published. It is necessary to know what makes one online news article more popular than another. As a preemptive approach, articles predicted to be unpopular can be enhanced to promote popularity by tweaking identified key features.

**3.Hypothesis:**

This problem falls under the domain of predictive analytics and can be tackled using machine learning algorithms such as regression, decision trees and Random Forest.

**4.Assumptions:**

1.The dataset is representative of news articles published by the Mashable website during the period from January 7, 2013, to January 2015.

2.The dataset contains no missing values, as it has already been preprocessed and cleaned.

3.The target variable "shares" is continuous and right skewed, with a few articles having a very high number of shares, while most articles have a lower number of shares.

4.The features in the dataset can be categorized into three groups: textual content features (such as number of words and average word length), keyword-based features (such as the number of keywords in the title), and metadata features (such as the article's category, author, and publication date).

5.There may be correlations between some of the features, which may need to be considered when performing further analysis or building predictive models.

**5.Approach:**

This problem falls under the domain of predictive analytics and can be tackled using machine learning algorithms such as regression, decision trees, SVM and Random Forest

**6.Why Above Approach:**

The approach of using machine learning algorithms such as regression, decision trees, SVM, and Random Forest is selected for the UCI Online News Popularity dataset as it is a predictive analytics problem where the goal is to predict the number of shares of a news article based on its features. Machine learning algorithms are particularly useful for this type of problem as they can learn patterns and relationships between the input features and the target variable from the data. Additionally, these algorithms can help to create a predictive model that can estimate the number of shares of an article and identify the most important features that drive the popularity of the article. This information can provide valuable insights to content creators and publishers for optimizing their content to increase engagement on social media.

**7.Project Background**

With the help of the Internet, online news can be instantly spread around the world. Social media, such as Twitter and Facebook, has made it easy for people to read and share news online. The popularity of an article is usually determined by its number of readers, likes, or shares. In our dataset, we use shares as the metric to measure popularity.

In the online news industry, content providers or advertisers benefit greatly from the ability to accurately predict the popularity of content before it is published. Thus, it is interesting and meaningful to use machine learning techniques to predict the popularity of these articles. The popularity of news depends upon various features like the usage of different keywords, relevance to a trending topic, and perhaps even the day the article is published. It is necessary to know what makes one online news article more popular than another. There are two main popularity prediction approaches, those that use features only known after publication and those that use features known before publication. The dataset we have used consists of known features before publication. Though the prediction performance is usually low in the latter case, it is of more value addition to authors and content creators to flag whether their produced article is going to be unpopular before publishing it. As a preemptive next step, articles predicted to be unpopular can be enhanced to promote popularity by tweaking identified key features.

**8.About the Data**

Machine Learning Repository of the University of California at Irvine provides a dataset with a heterogeneous set of features about articles published by Mashable, a popular blog site in the world, during a period of two years, January 7, 2013 – January 7, 2015.

A copy of this dataset is taken from [Kaggle](https://www.kaggle.com/datasets/thehapyone/uci-online-news-popularity-data-set) instead, due to difficulties in downloading from the original source. This dataset consists of 39,664 observations for 61 different variables, with the number of shares as our dependent variable. A Classification problem is developed from this continuous variable by first identifying the median value of all available shares as a threshold and engineering another feature that uses this threshold to mark articles as popular (1) or unpopular (0).

**8.1 SMART Questions**

Some of the SMART questions we initially framed were:

1. Does the day of the week an article is published have any correlation with the number of shares it receives? If yes, then which day receives the maximum shares and which day receives the minimum shares?
2. Which features of an article affect the prediction of shares an article receives, positively and negatively?
3. Which classification methodology produces the best results to predict whether an article is popular or not?
4. For the classification problem, is there a popularity imbalance in the dataset, and how to go around it? If yes, then what is the best metric to evaluate our model?
5. With what accuracy/precision/ recall can we predict, using the best model, whether an article is popular or not popular?

By using Statistical Modeling and Machine learning methods, we will attempt to predict if an article is popular or not, based on features that are available prior to the publication of the article.

Now we will begin our analysis.

**9. Initial Data Analysis**

Glossary of all the columns in the dataset:

* Url: URL of the article (non-predictive)
* timedelta: Days between the article publication and the dataset acquisition (non-predictive)
* ntokenstitle: Number of words in the title
* ntokenscontent: Number of words in the content
* nuniquetokens: Rate of unique words in the content
* nnonstop\_words: Rate of non-stop words in the content
* nnonstopuniquetokens: Rate of unique non-stop words in the content
* num\_hrefs: Number of links
* numselfhrefs: Number of links to other articles published by Mashable
* num\_imgs: Number of images
* num\_videos: Number of videos
* averagetokenlength: Average length of the words in the content
* numkeywords: Number of keywords in the metadata
* datachannelislifestyle: Is data channel 'Lifestyle'?
* datachannelis\_entertainment: Is data channel 'Entertainment'?
* datachannelis\_bus: Is data channel 'Business'?
* datachannelis\_socmed: Is data channel 'Social Media'?
* datachannelis\_tech: Is data channel 'Tech'?
* datachannelis\_world: Is data channel 'World'?
* kwminmin: Worst keyword (min. shares)
* kwmaxmin: Worst keyword (max. shares)
* kwavgmin: Worst keyword (avg. shares)
* kwminmax: Best keyword (min. shares)
* kwmaxmax: Best keyword (max. shares)
* kwavgmax: Best keyword (avg. shares)
* kwminavg: Avg. keyword (min. shares)
* kwmaxavg: Avg. keyword (max. shares)
* kwavgavg: Avg. keyword (avg. shares)
* selfreferencemin\_shares: Min. shares of referenced articles in Mashable
* selfreferencemax\_shares: Max. shares of referenced articles in Mashable
* selfreferenceavg\_sharess: Avg. shares of referenced articles in Mashable
* weekdayismonday: Was the article published on a Monday?
* weekdayistuesday: Was the article published on a Tuesday?
* weekdayiswednesday: Was the article published on a Wednesday?
* weekdayisthursday: Was the article published on a Thursday?
* weekdayisfriday: Was the article published on a Friday?
* weekdayissaturday: Was the article published on a Saturday?
* weekdayissunday: Was the article published on a Sunday?
* is\_weekend: Was the article published on the weekend?
* LDA\_00: Closeness to LDA topic 0
* LDA\_01: Closeness to LDA topic 1
* LDA\_02: Closeness to LDA topic 2
* LDA\_03: Closeness to LDA topic 3
* LDA\_04: Closeness to LDA topic 4
* global\_subjectivity: Text subjectivity
* globalsentimentpolarity: Text sentiment polarity
* globalratepositive\_words: Rate of positive words in the content
* globalratenegative\_words: Rate of negative words in the content
* ratepositivewords: Rate of positive words among non-neutral tokens
* ratenegativewords: Rate of negative words among non-neutral tokens
* avgpositivepolarity: Avg. polarity of positive words
* minpositivepolarity: Min. polarity of positive words
* maxpositivepolarity: Max. polarity of positive words
* avgnegativepolarity: Avg. polarity of negative words
* minnegativepolarity: Min. polarity of negative words
* maxnegativepolarity: Max. polarity of negative words
* title\_subjectivity: Title subjectivity
* titlesentimentpolarity: Title polarity
* abstitlesubjectivity: Absolute subjectivity level
* abstitlesentiment\_polarity: Absolute polarity level
* shares: Number of shares (target)

*Source: [kaggle](https://www.kaggle.com/datasets/thehapyone/uci-online-news-popularity-data-set)*

Understanding the data at hand was extremely important. The dataset had been specifically curated to facilitate a research paper that is referenced toward the end [1]. It was then donated to the Machine Learning Repository of UCI after extensive text feature extractions.

**9.1 Understanding the Data**

Some of the attributes in the dataset have already been encoded for machine learning. However, we will decode it into a single column for visualization purposes. Such columns include:

# 1. Data\_Channel: Type of article (Entertainment, Lifestyle, Media, Technology, World, etc.)

# 2. Publish\_Day: Day the article was published (Monday, Tuesday, etc.)

Some of the features are dependent on the particularities of the Mashable service (whose articles have been used as data source): articles often reference other articles published in the same service; and articles have meta-data, such as keywords, data channel type, and the total number of shares (when considering Facebook, Twitter, Google+, LinkedIn, Stumble-Upon and Pinterest). The minimum, average and maximum number of shares was determined for all Mashable links cited in the article and were extracted to prepare the data. Similarly, the rank of all article keyword average shares was determined, to get the worst, average, and best keywords. For each of these keywords, the minimum, average, and maximum number of shares was extracted as a feature. [1]

Several features are extracted by performing natural language processing on the original articles. The Latent Dirichlet Allocation (LDA) algorithm was applied to all Mashable articles in order to first identify the five top relevant topics and then measure the closeness of the current article to such topics. To compute the subjectivity and polarity sentiment analysis, the pattern web mining module was used. Subjectivity measures how subjective the article is – in terms of personal opinions vs factual information. It ranges from 0-1 with a higher score indicating more personal opinions and a lower score indicating factual inclination. Polarity sentiment analysis indicates the sentiment – negative, neutral, and positive. It ranges from –1 to 1.

An interesting point was raised by Prof Edwin Lo in his feedback on the project proposal. Is the number of shares a moving target? Is it a function of the number of days that have passed since when an article was published that determines the number of shares it receives? After researching online, one paper [1] suggested that there is a convergence of shares reached and the number of days it took that to happen was roughly 21.

There is a feature in the dataset: timedelta; which gives us the No. of days between the article being published and this dataset being collected. To reinforce the above assumption, it was important to see whether there was any relation between the timedelta and the number of shares the article received within our dataset.

We plotted this using a scatter plot:

A picture containing chart

Description automatically generated

Text

Description automatically generated

Two Observations here:

1. It seems like there is no concrete relationship between when the article was published and the number of shares it receives if data is acquired after 3 weeks, that is, there appears to be some convergence in the number of shares an article receives a definite number of days (21) after it has been published.

2. In the dataset at hand, the majority of the articles (~87%) have shares in the range of 0-5000. Whereas the maximum number of shares received is as high as 843300.

In addition, to eliminate the problem of class imbalance in our target variable, we used the median value of shares, instead of the mean to engineer our target variable.

**9.2 Exploratory Data Analysis**

While looking at the dataset, we had a few curious questions. we will help answer them with EDA.

Q1. What is the effect of digital content on the popularity of an article (hence the number of shares it receives)?

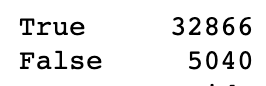
There are two primary digital content features – number of images and number of videos in an article. We’ll look at their distribution first.

Number of Videos:

Chart

Description automatically generated

Majority of the number of videos is distributed between 0 and 1. Looking at their value counts:



Here, True represents if the count is 0 or 1 and False represents any other number of videos. We see again that ~87% of the data is in the former category.

So, we decided to change our analysis a bit. Now we wanted to see if having 0 number of videos is more beneficial than having 1 or vice versa. We changed the scope of analysis to between just these two groups. To see this, we plotted boxplots:

Chart, box and whisker chart

Description automatically generated

From the above plot, we see that the average of num\_video = 0 and 1 is about the same. We can assume that having a greater number of videos does not have a direct effect on the number of shares.

Next, we plotted the relationship between Number of images and shares:

Chart, scatter chart

Description automatically generated

Looking at this plot, we see that there isn’t any concrete relationship between number of images to the number of shares. Although, articles that tend to have a number of images in the range of 0-20 seem to have a higher number of shares, crudely speaking.

Q2. Is there a relationship between the number of words in the content and the number of words in the title in the article’s popularity?

We first decided to see the distribution of the Number of words in the title:

Chart, histogram

Description automatically generated

Wow, it appears to be a nice normally distributed graph with an average of 10 words!

Next, we saw the distribution of the number of words in the content:

Chart, histogram

Description automatically generated

Upon checking the value counts of this feature, we identified an anomaly. Some articles appeared to have 0 words in their content! As the next step, we subset these mystery articles from the dataset and decided to have a look at the articles through their links directly.

We found that these were errors in the dataset. These articles did in fact have words in their content. This was an important catch because we’ll have to remove these erroneous records from the dataset before we train our model.

Before we proceeded to see the relationship of these two features with the target, out of curiosity, we wanted to see how they were related to each other. We plotted this relationship through a scatter plot:

Chart, scatter chart

Description automatically generated

It seems like, except for a few outliers, number of words in the content peak when the Number of words in the title is between 8-14 and they fall off gradually as it increases or decreases from this range. This is interesting.

Plotting relationship between number of words in title to popularity:

Chart

Description automatically generated

The unpopular line is not of that much interest as the number of shares for unpopular articles remains constant. But with popular articles, we see a constant increase in shares when the title increases and there is a sharp rise between 15-20. But this is also followed by a sharp decrease once ~18 or 19 words are crossed. Something to keep in mind for content creators.

We then plotted the relationship between number of words in the content and number of shares:

Chart, scatter chart

Description automatically generated

Here, we don’t see a significant relationship, but it does appear as though keeping the content small (0-2000 words), the number of shares could be higher. This is probably because readers would prefer reading a small catchy article over a long boring one and are more likely to share it.

Q3. Evaluating whether Day of Week has any effect on the popularity, in the number of shares?

To analyze this effectively, we created count plots on the Days of the Week, and we kept the hue as to whether the article was popular or not. This is what the plot looked like:

Chart, bar chart

Description automatically generated

This plot gives us highly useful insight. An article published over the weekend is more likely to be popular as opposed to an article that is published over the weekday. This makes intuitive sense since people have more time to read articles over the weekend as opposed to weekdays. To showcase this, we’ll plot the percentage chance of popularity on all weekdays according to our data:

Chart, bar chart

Description automatically generated

We observe a trend here. Near the weekends, the percentage of popular articles increases, and it peaks on Saturday, however, it gradually decreases then and is the lowest during mid-week on Wednesday.

Q4. Evaluating whether Data Channel has any effect on the popularity.

We took a similar approach to visualize this as above:

Chart, bar chart

Description automatically generated

We can observe that in the Data Channel of technology and social media, the number of popular news is much larger than unpopular ones, and in the Data Channel of world and entertainment, the number of unpopular news is larger than popular ones. This reflects that the readers of "Mashable.com" prefer the channel of technology and social media over the channel of world and entertainment.

To reinforce this further, we’ll plot the percentages like before:

Chart, bar chart

Description automatically generated

Q5. Title sentiment vs the number of shares.

This feature tells us whether the article expresses a negative sentiment, neutral or positive sentiment. We plotted a simple relation with the number of shares using a scatter plot:

Chart, scatter chart

Description automatically generated

Mostly the articles have titles that are not too positive or negative. It lies within the range of -0.5 to 0.5. However, the highest concentration can be seen in the 0 axes i.e., high no. of articles is neutral in nature with the higher number of shares. We also observe two small peaks at -1 and 1 in the number of shares.

Q6. Global subjectivity’s effect on the number of shares.

This feature gives us information on the subjectivity of the article. A high rating indicates more personal opinions whereas, a low rating indicates more factual information. Again, we plotted this using a simple scatter plot:

Chart, scatter chart

Description automatically generated

Maximum shares are received by articles that have values for Global Subjectivity between 0.3 to 0.7. Hence, we conclude that most of the articles with neutral Global Subjectivity receive maximum shares, that is, the articles contain a good blend of personal opinions and factual information.

**10. Pre-processing and Modeling**

**10.1 Pre-processing**

The following pre-processing steps were performed:

* Subset –
  + We first subset our dataset to remove all articles that had a timedelta of less than 21.
  + We also subset the dataset to remove those articles that had 0 words in their content, as identified during the EDA.
* Dropping irrelevant columns – Further, irrelevant columns like URL and timedelta were removed from the dataset.
* Correlation heatmap – We created a correlation heatmap between all the features of our dataset to eliminate features with high correlation. We chose our threshold value as 0.7. This was done to avoid the problem of any multicollinearity in modeling. This heatmap was a 45 x 45 matrix and therefore, we haven’t included it here. It can be found in the technical analysis.
* Outlier treatment – Since most of our columns were numerical, we decided to treat the outliers with the following rule:
  + Outliers clipped at Q1 – 1.5 IQR and Q3 + 1.5 IQR
  + Any value below or above this range was clipped at this boundary level, not removed from the dataset.
* Lastly, to deal with the imbalanced scale/range of features, we decided to employ StandardScaling as opposed to Log transformations. This was done because few features contained negative values and we also wanted to preserve the distribution of our dataset.

**10.2. Logistic Regression using Stats(Feature Engineering)**

We first decided to build a Logistic Regression model using the Statsmodel library with all the features available to us. This was the result:

Table

Description automatically generated

We then decided to rank all the features based on their p-values and build another model without these insignificant features. This was the result:

Table

Description automatically generated

No improvement in R-squared or adjusted R-squared value. Since variables were removed, we didn't expect to see this improvement either. Although, this tells us that we can build our models using a lesser number of features as well, and we'll explain the same variability in the response.

However, with p-values, we can only fail to reject the null hypothesis. This tells us that we don't have enough evidence in our dataset to reject whether the coefficients for these features are 0 or not.

For the rest of the models, we decided to include all the features that are available to us after removing the correlated once as part of pre-processing. We have used classification algorithms like KNN, Logistic Regression using Sklearn, Decision Trees, Random Forest, and SVC. Forall these models, train-test split was first done in the ratio 7:3. After which we performed standard scaling on all our features then.

**10.3. KNN ALGORITHM**

The k-nearest neighbor’s method, generally known as KNN or k-NN, is a non-parametric, supervised learning classifier that utilizes proximity to classify or predict the grouping of a single data point.

As first step, we plotted the error rate vs accuracy plot, and based on this we identified that the error rate was low when the k value is at 17. So, by looking at that plot we found that the optimum value of the k is 17.

**Error rate vs k value**

Chart, line chart

Description automatically generated

So, for the KNN algorithm, we choose the number of neighbors as 17 and the accuracy of the model with this K value was 63%. The confusion matrix and the classification report for the KNN model are shown below:

Chart, treemap chart

Description automatically generated

From the confusion matrix, we can interpret that 2557 are classified as false negative and 1662 are classified as false positive.

Table

Description automatically generated

We then plotted the ROC for our KNN model and found out the Area under the curve to be 0.69.

Chart, line chart

Description automatically generated

**10.4. Logistic Regression using the Sklearn library**

When the dependent variable is dichotomous, logistic regression is the proper regression strategy to use (binary). While implementing the logistic regression model, the accuracy of the model is 64%. The confusion matrix and the classification report are shown below.

Chart, treemap chart

Description automatically generated

From the confusion matrix, we get 2316 as a false negative and 2105 as a false positive.

Table

Description automatically generated

The AUC for the logistic model is 0.7, from the AUC we can say that there is a 70% chance that the model will be able to distinguish between positive class and negative class, and the logistic model is said to be a considerable model.

Chart, line chart

Description automatically generated

**10.5. Decision Tree**

We have implemented the decision tree for the given data set with a maximum depth of 5 first, we considered the decision tree with the default parameters. The accuracy of the model is 62%. So, then we adjusted some of the parameters in the decision tree function and tried different depths. We decided that the decision tree with maximum depth at 5 is considered optimum with an accuracy of 64%. The classification and confusion matrix for the decision tree algorithm is shown below-

Table

Description automatically generated

Chart, treemap chart

Description automatically generated

From the confusion matrix, we can analyze that 2107 rows are classified as false negative and 2159 are classified as false positive.

From the Decision tree, we also retrieved the feature ranking to help answer one of our SMART questions regarding which feature has the highest effect on popularity. This list of feature importance is below:

Text

Description automatically generated

The best-rated keyword’s average share is the feature that determines popularity the best.

We also plotted the ROC for the decision tree and found the Area under the curve to be 0.67 for the most optimum model.

Chart, line chart

Description automatically generated

**10.6. Random Forest**

During training, random forests (also known as random choice forests) generate a huge number of decision trees to use as an ensemble learning approach for classification, regression, and other problems. The output of a random forest is the class selected by the vast majority of trees, which is useful for solving classification issues. When a regression task is given, the average prediction of the individual trees is given back. Decision trees may overfit their training data, although random decision forests mitigate this problem. Random forests are more effective than decision trees in most cases. So, for the random forest, we just used the default parameters, and the accuracy of the model is 66%. Moreover, we tried random forests with different parameters, but the accuracy of the model has not increased. The confusion matrix and classification report for the random forest is shown below.

Table

Description automatically generated

Chart, treemap chart

Description automatically generated

From the confusion matrix, we can identify that 2166 are classified as false negative and 1793 are classified as false positive.

**Chart, line chart

Description automatically generated**

**10.7. Support Vector Machines (SVM):**

Lastly, we employed the SVM methodology. A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane that categorizes new examples.

An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification, implicitly mapping their inputs into high-dimensional feature spaces. By using this model, we obtained an accuracy of 65% by using some default parameters. We did not play around with SVM too much.

The classification and confusion reports are shown below.

**Table

Description automatically generated**

**Chart, treemap chart

Description automatically generated**

From the confusion matrix, we can identify that 2058 are classified as false negative and 1897 are classified as false positive.

Chart, line chart

Description automatically generated

**11. Feature Engineering**

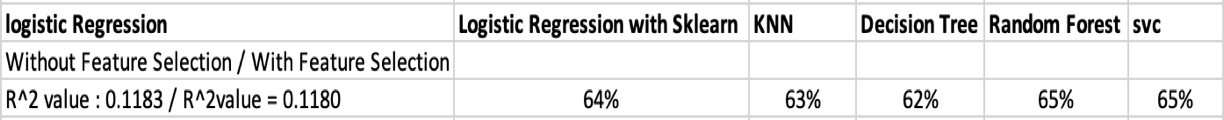
* Logistic Regression model is built initially using all available features, and then feature selection is performed based on p-values to build another model without insignificant features. While there is no improvement in the R-squared or adjusted R-squared value after removing the insignificant features, the process demonstrates that the same variability in the response can be explained using fewer features. Furthermore, the pre-processing step includes removing correlated features, standard scaling, and using several classification algorithms with a 7:3 train-test split ratio.
* Datatype of the time\_delta column is changed to DateTime datatype
* By using all data channel related columns in dataset new column Data\_channel is created
* By using all weekdays related columns in dataset new column publish\_DOW is created
* The values which have more than 0.7 correlation are deleted.

**12. Summary and Conclusions**

The key results that we identified from performing exploratory data analysis are:

* Publishing articles on Weekends may result in higher shares.
* Publishing articles on topics such as social media and technology may result in higher shares.
* A good article is one written with a good balance of personal opinions and factual information. This results in an increased number of shares.
* Remaining neutral towards content is important. Although, in some fringe examples, having extremely negative or positive opinions results in a higher share but it’s a hit-or-miss situation.
* Number of Words in the title to be between 10 – 18, not too long; not too short to catch the attention of the reader.
* Number of Words in the content should be in the range of 0-2000. Making the article short and catchy results in higher shares.
* The metric we choose for evaluation is AUC-

For Modeling, the summary of the performance of all the Models used can be seen below:



* The metric we choose for evaluation is AUC-ROC curve.
* Random Forest and SVC gave the best results out of all the models in terms of accuracy
* The AUC for Random Forest and SVC was also the highest – 0.71, 21% greater than random chance.

**13.Future directions:**

1.Exploring deep learning models: Deep learning models, such as convolutional neural networks (CNNs) or recurrent neural networks (RNNs), can be used to extract more complex features from the data and capture the nonlinear relationships between the features and the target variable.

2.Incorporating external data sources: External data sources, such as social media or sentiment analysis data, can be incorporated into the analysis to improve the predictive accuracy of the models and capture additional contextual information

**14. References**

1. K. Fernandes, P. Vinagre and P. Cortez. A Proactive Intelligent Decision Support System for Predicting the Popularity of Online News. Proceedings of the 17th EPIA 2015 - Portuguese Conference on Artificial Intelligence, September, Coimbra, Portugal.
2. UCI Online News Popularity Data Set, Machine Learning Repository of University of California at Irvine (2015). Retrieved November 6th, 2022, from [www.kaggle.com/datasets/thehapyone/uci-online-news-popularity-data-set](https://www.kaggle.com/datasets/thehapyone/uci-online-news-popularity-data-set)