Predicting and Evaluating the Popularity of Online News Articles

William Jacob Sehnert
University of Rochester
wsehnert@u.rochester.edu

Raajan Jonnala
University of Rochester
rjonnala@u.rochester.edu

ABSTRACT

Given the rising popularity of the internet as a news source, it is imperative for companies to understand what makes an online news article popular. This paper examines this very predicament by working to describe the characteristics of popular online news articles using various popular machine learning classification techniques. Using a large data set from the UCI Machine Learning Repository, we predict the popularity of online news articles using four algorithms: **(I)** Logistic Regression, **(II)** Naïve Bayes, **(III)** Support Vector Machines and **(IV)** Random Forest. Before implementation, we analyzed the variance and dependency of various features on our target outcome to only include strongly explanatory features and improve accuracy. We found our feature selection techniques were minimally hepful in improving accuracy scores, largely because the data provided is extremely clean and well-prepared. This project was completed as part of an undergraduate course, CSC 240 – Data Mining taken at the University of Rochester during Spring 2019.

***Keywords* – Data Mining; Data; Feature Selection; Machine Learning; Feature Selection; Classification; Online News Popularity; Prediction**

ACM Reference Format

JONNALA, RAAJAN AND SEHNERT, WILLIAM JACOB. Predicting and Evaluating the Popularity of Online News Articles. Rochester, NY, USA, 7.

I. INTRODUCTION

In a study conducted by the Pew Research Center, 43% of, 43% of Americans reported getting their news online as of August 2017, an increase of five percentage points from 2016. Meanwhile, other previously popular news sources, television, radio and newspapers experienced a decrease in their share of providing Americans with news **[3]**. As the percentage of Americans who rely on the internet to obtain their news continues to increase, competition among companies to reach as many users as possible will become increasingly important.

An example of a company whose success relies directly on the popularity of its content Mashable, a "global, multi-platform media and entertainment company" which strives to be the "go-to source for tech, digital culture and entertainment content" for its users **[2]**. Mashable's success lies directly in its ability to reach a broad audience. In addition to its primary users, Mashable also wants to expand its breadth; one way to do this is for users to share their articles to other social media platforms. According to an updated Pew Research Center study, 20% of U.S. adults often get their news from social media, an increase of two percentage points from 2016 **[6]**. Thus, Mashable can increase its outreach significantly if its users frequently shares its content. With this information in mind, it would be very beneficial for companies like Mashable to predict the number of shares one of their articles will generate.

In this paper, we examine a dataset containing information on a vast number of online news articles from Mashable to determine **(I)** which machine learning techniques most effectively classify the data and **(II)** which attributes best characterize a popular online news article.

II. DATASET

A. Data Collection

The *Online News Popularity Data Set* is provided by the UCI Machine Learning Repository and contains information on 39,797 online news articles from the Mashable website **[4]**. Every attribute was considered with exceptions of the website URL for each article and the number of shares for each article, which serves as the target label. We summarize the attributes in **Table I.**

Every attribute, with the exception of the URL is continuous. From this point forward we will no longer consider the URL attribute. The data set required no cleaning, as K. Fernandes et. al removed all missing values and limited correlated information among features **[4]**. The majority of the attributes are intuitive to understand, but we describe a few that may be unfamiliar to the reader:

- *global_sentiment_polarity* – value between [-1,1] with 1 being extreme positivity and -1 being extreme negativity
- *global_subjectivity* – value between [0,1] where 0 represents objectivity and 1 represents pure subjectivity

- *LDA topics* – a neuro-linguistic programming schematic that extends beyond this paper. Pleaser reference the cited article to learn more **[1]**.

Table 1: Attribute Description

Category	Features
Links	Days between publication and acquisition; Number of links in article; Number of Mashable links in article
Topic	Channels from Mashable website: Lifestyle, Entertainment, Business, Social Media, Tech, World;
Date/Time	Indicator variable for each day of week in addition to weekend as a whole
Media	Number of images and videos
Words	Number of words in title and in article; Average word length; Rate of both unique and non-stop words
Keywords	Number of keywords; Best, Worst and Average Keyword by # shares
Neuro-Linguistic Programming	Closeness to 5 LDA* topics; Text and title polarity and subjectivity
Share	Number of shares (target); Number of shares of linked articles

*Latent Dirichlet Algorithm **[1]**

In order to concretely define whether or not an article is popular, we analyzed the distribution of the *Shares* target feature. We wanted to use a measure of central tendency, either the mean or the median. In order to decide which is optimal, we analyzed the distribution of the range of shares. Initially, we found its range to be very large at [1, 843,330]. Furthermore, when creating a summary table, we noticed, however, that the 75% of the number of shares were less than or equal to 2,800. Thus, with such a significant discrepancy between the 75th quartile and maximum value, we created a histogram to visualize the spread.

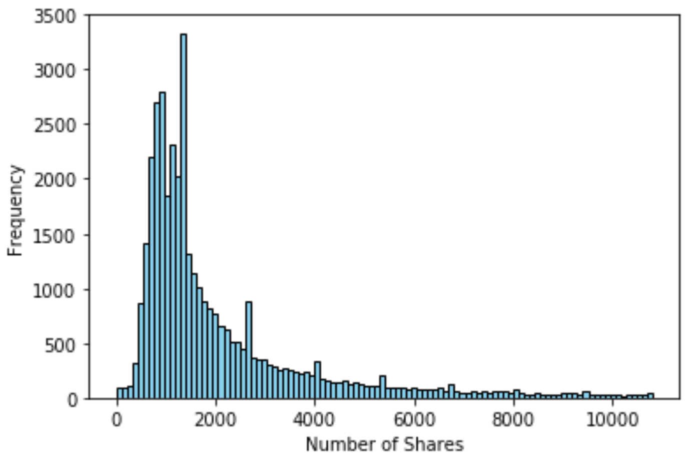

Fig. 1 Histogram of Shares

We eliminated the last 5% of the data to enhance the quality of the histogram. The number of shares at the 95th percentile is 10,800, which indicates we are not underrepresenting larger values since the difference between the 95th and 75th percentile is much larger than the interquartile range. Thus, we will use the median, 1,400 shares, to be the cutoff for a successful article.

III. Filter Methods

Now that we adequately described and understood the data, we proceeded to analyze the features and their importance to characterizing a popular article.

A. Variance Threshold

The first filter method we applied was the variance filter. We first use the ***train_test_split*** method found in the ***sklearn.model_selection*** library to separate the data into a test set and training set. We included 30% of the data in the testing set and 70% of the data in the training set in every machine learning approach. Then, we computed the variance for each of the attributes in the training set and applied a filter using a threshold variance to eliminate features that are constant.

We initially used a variance threshold of 0.0, but we found that all features had a non-constant variance, so we failed to remove any from further analyses at this step. We then tried using a threshold of 0.1, which filtered out five quasi-constant features:

- *global_sentiment_polarity*
- *global_rate_positive_words*
- *global_rate_negative_words*
- *min_positive_polarity*
- *max_negative_polarity*

We removed each of the features listed above from our further analyses, leaving us with 54 features.

B. Mutual Information

The next filter method we computed was the mutual information of each feature and label using the ***mutual_info_classif*** found in the ***sklearn.feature_selection*** library which uses the equation below:

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} P(x,y) log \frac{P(x,y)}{P(x)P(y)}$$

The mutual information score of a given feature on the label is a representation of dependency, with 0 indicating the variables are entirely independent. The equation can also be written as

$$I(X;Y) = H(X) - H(X|Y).$$

where H(X) represents the entropy of X. The entropy of X defines its level of randomness, so higher values indicate higher levels of uncertainty. In this equation, the features are represented by Y and the number of shares by Y. Thus, the H(X|Y) term represents the entropy of X given Y, and, if it is less than H(X), then Y reduces the uncertainty of X. Thus, mutual information can also be looked at as a reduction in uncertainty.

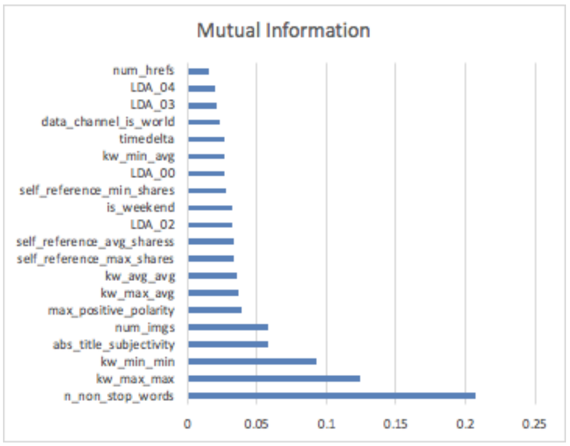

Fig. 2 Mutual Information Scores of top 20 features

C. F-Value

We then compute the ANOVA F-Value between the features and label using the ***f_classif***, which is also found in the ***sklearn.feature_selection*** library, and uses the following equation

$$F = \frac{\left(\frac{\text{RSS}_1 - \text{RSS}_2}{p_2 - p_1}\right)}{\left(\frac{\text{RSS}_2}{n - p_2}\right)}$$

The formula itself is a bit complicated, but it can be simplified by thinking of it in terms of explained variance divided by unexplained variance. Thus, a higher F-value indicates the given feature explains a greater proportion of the variance in the number of shares.

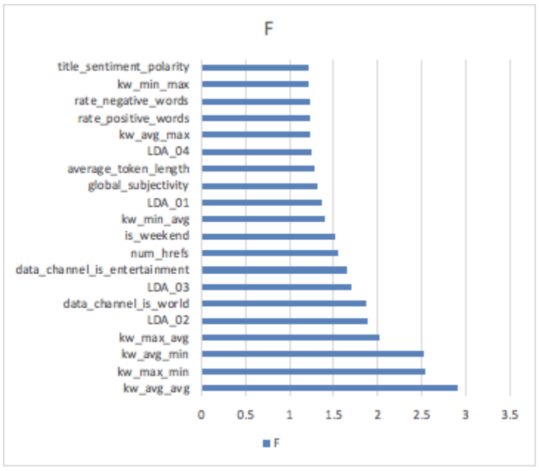

Fig. 3 F-Value of top 20 features

D. COMBINING RESULTS

Given the results of parts *A.* and *B.*, we wanted to find a creative mechanism that has not been previously used. Ren and Yang use cross-validation techniques to complete their feature selection using Fisher Scores and logistic regression **[5]**. Since we computed two scores that measure mutual dependency, we decided instead to determine the set of features to be used in our machine learning approaches by averaging the mutual information and F values for each feature. Since each value uses a different scale, we first normalized values so that each fell in the range [0,1] using the following equation:

$$z_i = \frac{x_i - \min(x)}{\max(x) - \min(x)}$$

We show the top 20 remaining features following this computation in the **Table 2.**

Table 2

n_non_stop_words	*LDA_02*
kw_max_max	*is_weekend*
kw_min_min	*self_reference_min_shares*
abs_title_subjectivity	*LDA_00*
num_imgs	*kw_min_avg*
mx_positive_polarity	*timedelta*
kw_max_avg	*data_channel_is_world*
kw_avg_avg	*LDA_03*
self_reference_max_shares	*LDA_04*
self_reference_avg_shares	*num_hrefs*

Top 20 Features According to Normalized MI and F-Value Average

IV. Machine Learning Approaches

After eliminating our data of quasi-constant features and selecting the top 20 features that the label, number of shares, is more dependent on, we proceeded to apply some common machine learning algorithms to best classify a popular article. In this section, we will apply *A. Logistic Regression, B. Support-Vector Machine*, *C. Naïve Bayesian Classifier* and *D. Random Forest* to determine the characteristics of a popular online news article.

A. Logistic Regression

Description

Logistic regression is a linear model that helps predict categorical variables, so in this case, predicting the number of shares where a successful article has greater than or equal to 1,400 shares. The model is useful because it allows us to isolate causal effects of explanatory variables on a predicted outcome. The goal of this regression is to assign weights to features in a manner that maximizes the likelihood of our probability and use that to categorize the response variable. (number of shares).

Components

We first must implement the link function, which, given a score, computes the following equation:

1 / (1 + np.exp(-scores))

Next, in order to maximize the likelihood, we used a log-likelihood equation to help with binary classification. By using the equation below, the weights of each feature are not affected by the following equation.

$$ll = \sum_{i=1}^{N} y_i\beta^T x_i - log(1 + e^{\beta^T x_i})$$

In the equation above, y represents the target class, x represents each point and B represents the weight assigned to each point.

Finally, we needed an equation to calculate the gradient of the log-likelihood. We decided to use gradient ascent as the choice because it will maximize the function.

$$\bigtriangledown ll = X^T(Y - Predictions)$$

Next, we calculated the weights of the features, which are used to help classify the online articles and determine the accuracy. The calculated weights were -18321 and -17653, which were relatively accurate compared to the library implementation of the algorithm.

Implementation

We implemented the model using a Jupyter notebook, where we hard-coded the mathematical equations to find the weights and then the accuracy. No libraries were implemented to do the coding for this classifier. When we run the classifier algorithm, we obtained about 64% accuracy.

Feature Selection Analysis

Before feature selection, the accuracy level was 65.5% and went to about 64 % after feature selection. The decrease is likely a result of the restricted number of features used in the analysis, since including more explanatory variables in a regression will generally increase its explanatory power.

B. Naïve Bayesian Classifier

Description

The second model used is the Naïve Bayes classifier, which is an algorithm based off of Bayes Theorem. This classifier is fast and accurate, so it is ideal when working with bigger datasets. It is important to note that the effect of an attribute value on a given class is independent of the other attribute values, which is important since cross-variation can reduce the accuracy of a classifier.

Components

This classifier can be broken up into five different steps:

1. Create a set of attributes and class labels.
2. Use the classifier to predict if the attribute belongs to either the popular or unpopular class by calculating the highest posterior probability using the formula below.
$$P(C_i|\boldsymbol{X}) = \frac{P(\boldsymbol{X}|C_i)P(C_i)}{P(\boldsymbol{X})}.$$
3. Maximize the numerator of the fraction, $P(X|C_i)P(C_i)$, because P(X) is the same for all classes.

4. Next, to reduce computation in evaluating P(X|C$_i$), the naïve assumption of class-conditional independence is made. This infers that the attributes' values are conditionally independent of one another, given the class label of the tuple. The formula used is presented below.

$$P(\boldsymbol{X}|C_i) = \prod_{k=1}^{n} P(x_k|C_i) = P(x_1|C_i) \times P(x_2|C_i) \times \cdots \times P(x_n|C_i). \quad (8.1$$

5. Lastly, To predict the class label of popular or nonpolar articles, P(X|C$_i$)P(C$_i$) is evaluated for each class C$_i$ and for each attribute as well.

Implementation

We implemented this classifier using the pandas' library.

Feature Selection

Feature selection did not improve accuracy, as it hovered around 64% for every set of features used.

C. Support-Vector Machine

Description

We then implemented the Support-Vector-Machine Algorithm. This was our most accurate classifier compared to K. Fernandes research paper **[4]**. The main goal for this classifier is to decide an imaginary hyperplane boundary to separate the classes. It uses linear or non-linear mapping to create the hyperplane which is made from supporting vectors and margins.

Components

We decided to try both a linear and non-linear plan but decided to use a linear for our dataset. This is because of how the data was organized and we realized it would be best to use a linear approach. The approach we used first was to find the hyperplane by using the MMH (Maximum marginal hyperplane). Using this approach allows to give the greatest separation between the two classes. In order to find the hyperplane, we must use the appropriate mathematical equations which are listed below.

$$\boldsymbol{W} \cdot \boldsymbol{X} + b = 0,$$

The equation above shows how separating a hyperplane can be written, where W stands for vector weight and b is the scalar. Next, we specifically used two linear equations that will give us the boundaries in how to separate the classes. They are listed below,

$$H_1 : w_0 + w_1x_1 + w_2x_2 \geq 1 \quad \text{for } y_i = +1,$$
$$H_2 : w_0 + w_1x_1 + w_2x_2 \leq -1 \quad \text{for } y_i = -1.$$

Utilizing these two equations are needed to create the needed classes to categorize the popularity of the online articles. Lastly, once we trained some of the test data, in order to classify the new popularity articles, we rearranged one of the previously talked about formulas to actually classify the data. It is shown below,

$$d(\boldsymbol{X}^T) = \sum_{i=1}^{l} y_i \alpha_i \boldsymbol{X}_i \boldsymbol{X}^T + b_0,$$

We also tried the non-linear case, however, we will not go into details about it since it was not our main focus.

Implementation

We implemented this classifier by using a pandas library in python and we could find the accuracy level of this algorithm. We found the accuracy level to be around 65.5% which is 10% higher than previous work done by it.

4.3.3 Feature Selection (Before and After)

Feature selection were both around 66% before and after feature selection.

D. Random Forest

Components

The last classifier we used was the random forest classifier in which we used about 500 decision trees to make the forest. Each of the individual decision trees we created were generated using a random amount of attributes. From here, each tree is dependent on a random vector that is sampled independently. After this, we classified the trees by each tree decides on its own which type of article is the best and from there it determines which type of article is the most popular.

Implementation

We implemented this classifier by using the specified libraries and directly deciding how many trees to classify it by. We used 500 decision trees in our case.

Feature Selection

We found that the accuracy levels were around 66% which we found to be highly accurate.

V. Results

The results of applying the four machine learning algorithms above to all features and the top 20 features according to mutual information score, F-value and a combination of the two can be summarized in the table below. The highest accuracy levels we could obtain for all classifiers is 70% because of how the dataset was organized. A summary table of the results can be found on the last page of this paper.

ACKNOWLEDGEMENTS

We first and foremost would like to thank Professor Ted Pawlicki for the time and effort he has put in to teaching this class in addition to providing us with the opportunity to not only learn about, but implement many common data mining algorithms. From acquiring the data, to selecting appropriate features, to applying the algorithms and finally analyzing our results, we have both learned how to successfully complete a data mining task. We would lastly like to thank the University of Rochester for the time and resources we have been provided while completing this project.

REFERENCES

[1] 1367994986571440. (2018, June 24). LDA Topic Modeling. Retrieved May 1, 2019 from https://towardsdatascience.com/lda-topic-modeling-an-explanation-e184c90aadcd

[2] About - About. (n.d.). Retrieved May 1, 2019, from https://mashable.com/about/

[3] Gottfried, J., Shearer, E., Gottfried, J., & Shearer, E. (2017, September 07). Internet closes in on TV as a source of news in U.S. Retrieved May 1, 2019, from https://www.pewresearch.org/fact-tank/2017/09/07/americans-online-news-use-vs-tv-news-use/

[4] 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

[5] Ren, He and Quan Yang. “Predicting and Evaluating the Popularity of Online News.” (2015).

[6] Shearer, E., & Shearer, E. (2018, December 10). Social media outpaces print newspapers in the U.S. as a news source. Retrieved May 1, 2019, from tank/201/12/10 /social-media-outpaces-printnewspapers-in-the-u-s-as-a-news-source/

Before Feature Selection	Accuracy	Mutual Information Accuracy	F-Value Accuracy	Combined Mutual Information and F-Value Accuracy
Logistic Regression	65.5%	65.5%	64.4%	64.2%
Naïve Bayes	63.9%	63.9%	64.0%	64.0%
SVM	65.5%	65.5%	*66.0%	*66.0%
Random Forest	*66.3%	*66.3%	65.0%	65.0%

Table 3 Summary of our Results