Duke University
Everything Data - CS 216
Team 23: Jess Beering, Molly Carmody, Bella Hutchins, Ryan Nicholson, Catalina Sanchez-Carrion
20 April 2018

Predicting Popularity in New Media

The Problem

Social media today is one of the most prominent ways people find important information about our world regarding politics, the economy, social issues, and more. The conflation of social media and news reporting is an especially contentious issue since the 2016 U.S. election, given the impact that social media posts are reported to have had on the election results. News and media outlets aim for their posts to reach the largest number of people as possible, and thus have a high popularity among social media users. There are various factors that the popularity of an article depends on, including the day and time it is published, the subject matter, multimedia within the piece, and more. Therefore, we are interested in studying the impacts that factors like these have on the popularity of articles in order to predict how many shares an article will get, and thus, how popular it will be.

Along with deciding if an article will be popular overall, determining the most important features of a popular article is valuable. Knowing which features correlate to an article's number of shares would provide insight to social media platforms as how to craft a popular article.

Related Work

Considering the high usage and dominance of social media in society today, many similar studies have been conducted to determine the perceived probability of a post. For example, HP Labs analyzed news articles to create a statistical model to rate the popularity of a tweet. This study considered article features such as source, category, emotional sentiment of words, and name dropping. Their conclusion was that the source of the news mattered most for a high number of tweets.

Another group of data scientists from the Washington Post looked at the effects of metadata, contextual, temporal, and social features on the popularity of Washington Post articles. They performed regression models and a 10-fold cross validation on collected feature data.

In our project we decided to focus on specific features including usage of visual aides, the day of the week in which the article was posted, and article category. Additionally, we wanted to analyze the actual content of each article.

Our Approach

Our approach was to use different classifier models to evaluate which article features most affected the number of shares. We used three classifiers: Naive Bayes, Regression and Random Forest. The classifiers were trained using a subset of the original data where the number of shares and attributes were known.

Our Goals

We established 3 main goals:

1. To explore which attributes would best predict the popularity/number of shares an article receives.
2. To explore the accuracy and strength of three different predictive models: Naive Bayes, Regression and Random Forest.
3. To assess the importance/relevance of the number of Google Trending Topics that an article contains and the correlation to number of shares.

Data Sources

We used data from the following sources:

1. UCI Online News Popularity Data Set: a dataset that contains over 60 article attributes for over 39,000 Mashable News articles
2. Google Trending Topics (per month for 2013 and 2014): list of top Google searches for given month and year of various categories

Factors

Below is a list of factors we included in our classifier:

- Week day posted
- Number of visual aides
- Category (lifestyle, entertainment, business, social media, technology, world)
- Trending Words Count
- Automated Readability Index (ARI)

Process

Tools

Below is a list of Python libraries we worked with:

- Python Scikit-learn
- Python Pandas
- Python Numpy
- Python csv
- Python matplotlib

Data Cleaning

Cleaning original csv file: The original UCI dataset was relatively clean and contained over 60 article features. We focused on a subset of these features that we deemed important to popularity. Therefore we created a more concise csv file that only contained the features of interest; this was utilized for the Naive Bayes prediction model.

Cleaning to get trending word counts: The majority of our data cleaning was adding our own additional features to the csv file for further analysis. We wanted to perform our own analysis on the article content. In order to accomplish this, we had to upload the content of each article into the csv file. This was the most significant task because we had to use cloud computing and MapReduce since the dataset contained over 39,000 articles. After acquiring the article content, we wanted to evaluate the number of trending search words that each article contained to see if there was a correlation between trending words and popularity. We used Google Trends Top Charts as the source of the trending words. Google Trends provides the top searches for specific categories for each month; we uploaded the top searches for ten specific categories for each month of 2013 and 2014 as this is the time frame of the articles in the dataset. Using the top trending words we created features for the count of trending words in the title of each article and the actual content.

Cleaning to evaluate the readability index: We also created a feature for the automated readability index (ARI) for the content of each article. The Automated Readability Index is a universal benchmark test for English text. The test yields a score that indicates the estimated reading level of the text. For example, a score of roughly 5 shows that an article is fit for fourth or fifth grade readers, whereas an article with a score of 14 is fit for college-level readers. We assigned each article a score and used this as a feature to see if there was a correlation to an article having a high number of shares.

Results and Evaluation

Naive Bayes

We conducted Naive Bayes on our dataset using three features with multiple subcategories to predict whether an article is unpopular, normal, or popular. We denoted an unpopular article having less than 957 shares with a 0, a normal article having between 957 and 2800 shares with a 1 and popular having over 2801 shares with a 2. These separation ranges were determined using the percentile function within Python's statistic library. The range of shares for the bottom 25th percentile was [1, 957], for the middle 25th to 75th percentile [957, 2800], and for the top 25th percentile [2800, 843300].

Our features were:

- Number of visual aids (images, videos),
- Article category (lifestyle, entertainment, business, social media, tech, world), and
- Day of the week posted.

Below is a chart comparing target popularity values with the prediction popularity values from the classifier.

Target Popularity Values	Predicted Popularity Values
[0 1 1 0 1 0 2 2 2 1 1 1 2 1 2 1 2 1 0 2 1 1 2 2 0 1 1 0 1 1 2 1 0 1 2 2 2 1 2 0 2 1 1 2 2 0 2 1 1 2]	[0 0 0 0 2 0 1 0 1 0 0 2 2 2 0 1 2 2 0 0 1 0 0 0 0 1 0 0 0 0 2 0 0 1 1 2 0 0 2 0 0 0 0 1 0 0 0 0 2 0]

Accuracy of Naive Bayes Classifier Using All Features: 38%

This accuracy is very low, which signifies that Naive Bayes is not the best prediction model to use for our data. However, we will still need to add two features into the classifier, the trending word count and readability score and this could improve the accuracy score.

Furthermore, we performed Naive Bayes on each separate feature category (visual aides, article category, and day of week) to see if the accuracy score differed among them. The chart below shows that the accuracy was highest when only predicting based off article category, which means the type of article has a higher relation to the number of shares an article will receive.

Features	Accuracy
Visual Aides	49.42%
Article Category	56.27%
Day of the week	52.43%

Random Forest

Another predictive model that we wrote for our project is Random Forest, imported from sklearn. In Random Forest, the "training" consists of giving a group of decision trees a list of features and the shares that accompany those features. As the model takes in information, it creates decision trees that are made up of many small "questions" leading to the "target" answer (the number of shares). This "teaches" the trees how certain features impact the total number of shares of an article.

In order to predict the number of shares of articles, the Random Forest model takes in features and uses the decision trees that it formed during the "training" process to come to a conclusion about how many shares an article has.

We used the same features in our Random Forest model as we did in our Naive Bayes model: number of visual aids, article category, and day of the week posted. Before our presentation, we also plan to add a feature for trending words during time of publication, to see if there is a correlation between trending words at the time and the popularity of an article that contains those words.

Our code also extracts information about the individual impact that each feature has on the number of shares as an article, which helps us understand how important one feature is versus another. The code produces a visual representation of this so as to most easily compare the importance of these features.

We are still in the process of running the Random Trees model on our large amount of data, and will include accuracy as well as the impacts of each feature on the number of shares, in our final report.

Regression

We are currently doing exploratory data analysis on our dataset. Our dependent variable is number of shares and our independent variables are day of week, number of visual aides, article category, automated readability index, and number of trending topics in the article. Our regression model will help us understand which features influence the popularity of an article the most. When we finish cleaning the data (removing outliers, adding/removing variables, variable transformations, etc.), the coefficient correlations will be more accurate and we will be able to distinguish what really influences an article's popularity.