

Our project tackles the broad task of analyzing news article topics and determining whether the topic of the article is in a specific category (Sports, Music, Entertainment, etc). Due to time, we categorized on news that were on the topic of big tech companies such as Facebook, Apple, Microsoft, etc. The scope of our project deals with analyzing news articles under the Facebook “Trending” news column. From that analysis, we can extrapolate information to classify other news articles as articles about certain tech companies via the usage of a machine learner. We find this task to be important because of how relevant tech-related discussion is in recent times and because of how widespread the usage of Facebook is. Since Facebook has over 1 billion users, it is an easy source to distribute important news to many different demographics. In fact, according to a study by Pew Research Center, 67% of Americans get their news from social media. Additionally, Facebook has recently been a topic of hot discussion regarding fake news and algorithms that present biased posts. Therefore, we believe it is crucial for social media news platforms to give relevant news topics, and due to the prevalence of the tech scene, we decided to pursue those avenues of media. Ultimately our project aims to allow easy distribution of important tech-related news articles relevant to certain users on Facebook.

To begin, we needed to gather data by collecting information from many Facebook articles that were featured. We wrote scripts to handle retrieving this data from Facebook. The scraping scripts were written in Python 3.6.4 and Selenium, a user interface (UI) automation tool that automates UI testing. Selenium was selected because it can simulate a real human user and trick the web browsers. It can distribute and scale scripts over many different environments and can create robust, regression automation tests. Regression automation test’s purpose is to catch bugs that were accidentally introduced and make sure previous bugs stay dead. Initially, we collected data using Mozilla Firefox because Selenium IDE, a Firefox add-on, was only available on Firefox and we had experience with Selenium IDE. Since we did not use Selenium IDE for this experiment but instead we used Selenium WebDriver which is available on multiple browsers, we were not restricted to one browser. After running into issues with Firefox when running the scripts on the AWS cloud server, we changed from Firefox to Google Chrome. Another reason we changed from Firefox to Chrome was because Chrome is the most popular web browser with roughly 78% of browser usage verses 11% for Firefox. It was important to collect data from the most commonly used browser since there was a greater chance most Facebook users use Chrome to access their Facebook accounts. Our raw data consisted of roughly 50,000 data each with the 7 attributes listed below:

- Type: The topic (top trends, politics, science and technology, sports, or entertainment)
- Title: Title of news trend
- Description: The short description located under the title
- Trend Link: The link that redirects the user when the trend is clicked. The link redirects the user to a compilation of news on a Facebook page.
- Rank: Where the trend is ranked in the trending list
- Scrape ID: An integer to keep track of which round of scraping the data is collected from
- Timestamp: The exact time and day the data was collected (YY-MM-DD HH-MM-SS)

Type was used to distinguish the different topics during the analysis. The analysis investigated the top trends for all five topics on the Facebook news platform in addition to the top trends for each of the topics. Title and description were collected for detail for each news trend. The trend link was used to uniquely identify each trend. Rank was collected to discover

where in the list certain new trends were placed and whether they moved up or down the list. Scrape ID was used with trend link to uniquely identify trends for the whole dataset. Timestamp was used to keep track of the time and date the data was collected, which was critical when analyzing the trend behavior at different days of the week.

We preprocessed our data to address the incompatibility between our raw CSV and what Weka accepted. We further preprocessed our data and removed duplicate news articles by filtering by description and went in by hand to add a classification attribute based on whether the article is related to our chosen topic or not. This resulted in a tremendous decrease in our dataset size, from approximately 50,000 to 1,213 data points, with 25% held out for testing, and there was a very small percentage of articles that were relevant to our topic. This was expected as there are many repeated news articles on Facebook. After this preprocessing, we used Weka to experiment with implementing different classifiers and recording accuracy and precision and recall for the “yes” classification.

Classifier	Accuracy	Precision (Yes)	Recall (Yes)
ZeroR	95.2747	N/A	0
Naive Bayes	93.956	40.9	62.8
Logistic	96.4835	64.9	55.8
IBk	96.2637	65.5	44.2
J48	95.2747	N/A	0
SimpleLogistic	96.7033	81.0	39.5
BayesNet	90.7692	31.9	83.7

Figure 1a: Training Data

Classifier	Accuracy	Precision (Yes)	Recall (Yes)
ZeroR	79	N/A	0
Naive Bayes	87	90	42.9
Logistic	90	86.7	61.9
IBk	91	100	57.1
J48	79	N/A	0
SimpleLogistic	86	100	33.3

BayesNet	87	90	42.9
----------	----	----	------

Figure 1b: Test Data

Figure 1a above depicts the results that various machine learners from Weka achieved on our training data. We recorded accuracy, precision, and recall. Figure 2b depicts the results of machine learners on our test data. While some of the results might be sporadic, we focused on precision. We believed precision and recall are more important than accuracy for our project because a large percentage of our collected data was not related to tech companies. Thus, there are many examples in our dataset that are not relevant to tech companies and will be classified as a clear “no”. As a result, the accuracy will be high, even if the classifier was not optimal. Precision was given more importance than recall, as precision showed how many of our classifications were correct classifications, whereas recall only shows how many within a category we were able to classify correctly as that category. In other words, precision is more important because we are placing more importance on showing relevant news articles, even if it might lead to fewer news articles.

Our solution, in terms of accuracy, performed well across the board with different machine learners. This however, is to be expected, as our data shows that most news articles will not be about tech companies because there is such a large variety of topics that the media covers. As a result, machine learners will all get similar accuracies by classifying most articles as not relevant to tech companies. Additionally, the machine learners with the highest precision were Simple Logistic and IBk, which we ran on Weka using our testing data.

The unpruned decision tree showed us that title was the most important attribute for our classifiers. This makes intuitive sense as title holds concise and reliable information about any article. As a key finding, we observed that the classifications primarily work with key words in the article titles, therefore news articles regarding large tech companies often have the name of the companies in the title.

For future experiments, we recommend conducting data collection over a longer period. If more data is collected, there should be more articles about a specific topic or in our case, articles about tech companies. This would improve the experiment since there would be more tech related articles to train on. As a result, the accuracy produced with more data would be more reliable and critical.

The members of the group that worked on this project are Julie Kim, Joshua Koo, and Shin Lee. Shin wrote, debugged, and ran the scripts to scrape data from Facebook. Shin and Julie worked on preprocessing the raw data collected. Julie and Joshua worked on the data preprocessing and testing. Data analysis and report was worked on collectively as a group.