IDS Miniproject technical report

Emma Kuparinen, Sriharsha Ayyagary, Heljä Räisänen

Definition of clickbait

We defined clickbait as a headline, which attempts to get the user's attention, and encourages the user to click on it by creating often exaggerated promises. This definition included articles which, for example, used a lot of superlatives or phrases along the lines of "You wouldn't believe this."

Clickbait also does not elaborate on these in the headline, and the user would have to open the article to see what the main points of the article are. For example, clickbait might promise "This amazing method works really well!", but users would have to open the article to see what the method is.

Typical properties of clickbait headlines were, for example, usage of superlatives, talking directly to the reader (using the word "you"), or "Top X" type of lists.

Data collection

We used both a readily available dataset published on Kaggle.com and a dataset we collected ourselves.

The ready dataset was "News Clickbait Dataset" [1], which contained two CSV files with 20 000-30 000 news headlines from unknown sources. The files additionally contained a binary label for each headline indicating whether they are clickbait or not. The author of the dataset didn't publish any information about the basis on which this classification was made, but we manually analyzed some of the headlines and decided that the classification was similar enough to our own definition of clickbait that we decided to use them for machine learning (ML) algorithm training.

In addition to this data, we used web scraping to collect more news headlines from some websites. Our goal was to examine some of the most read English-language dedicated news websites. We had an original goal of trying to examine multiple news sources from multiple countries to contrast the levels of clickbait in places such as the United States, United Kingdom, and India. However, because of the time constraints and difficulties with

trying to build a web scraper in Python, we ended up narrowing down the list of news websites we examined to four. These were CNN, BBC, AP News, and NPR. In addition, we included BuzzFeed as a control, as it is seen as a rather outstanding example of clickbait.

These four news websites in particular were chosen because of the relative ease in trying to scrape the headlines through a relatively consistent Python script. We found a guide online on how to build a web scraper. It called for using the library BeautifulSoup, and it would look through the user-provided URL for a specific characteristic to scrape.

In our case, as we were looking for headlines, we would examine the webpage with the inspect element tool to find out how the certain website would program its implementation of headlines. The four news websites we ended up going with had a rather straight-forward implementation, with headlines each having a certain class ID that could be put into the Python script. Other news websites had more complex implementations, with each headline having its own separate link in the code. Fox News was one of such websites. Therefore, it would make scraping all of the headlines on a certain page substantially more difficult. Because of this factor, those websites were not included.

Additionally, we sought to avoid websites that were locked behind paywalls, in other words, having to subscribe to properly access the website. Websites such as NYTimes.com, the online version of The New York Times, have this issue. The four websites we used in this case did not have any issues in this regard.

To analyze a wide sample set of data, we decided to examine each website's main page over the span of two months. We did this with the use of The Wayback Machine. All four of the news websites we analyzed, plus BuzzFeed, allowed us to get access to this similar time range of data. While the number of headlines produced by each site for each day would differ, having a consistent time frame is one way to control this variable.

The four news sites we ended up using did present a scope of different types of news sources, across the United States and United Kingdom. NPR (formerly) and BBC are both partly funded by the state, AP News is run by the non-profit Associated Press, and CNN is owned by Warner Bros. Discovery, which is a for-profit corporation.

Exploratory data analysis

We manually went through some of the labelled headlines and thought about what set the two classes apart. In addition, we performed term frequency – inverse document frequency (TF-IDF) analysis with two documents: all clickbait headlines and all non-clickbait (called "news") headlines.

Here are the highest ranked, stemmed words from the TF-IDF Analysis for each class.

Non-clickbait (news) set		Clickbait set	
word	Tf-idf score	word	Tf-idf score
new	0.27	you	0.74
kill	0.27	your	0.34
us	0.22	thing	0.16
die	0.15	peopl	0.12
win	0.14	we	0.12
say	0.12	make	0.12
dead	0.12	know	0.11
uk	0.10	time	0.11
presid	0.10	2015	0.10
bomb	0.10	17	0.10
crash	0.10	21	0.09
year	0.10	actual	0.08
australian	0.10	base	0.08
elect	0.10	19	0.08
feature	Correlation with news	feature	Correlation with clicbait
'dead'	-2.79	‘you'	4.06
'kill'	-2.78	'charact'	3.89
'china'	-2.74	'hilari'	3.88
'iraq'	-2.60	'everyon'	3.68
'plan'	-2.32	'2015'	3.55
'obama'	-2.32	'love'	3.10
'launch'	-2.28	'actual'	3.01
'die'	-2.23	'everi'	2.82
'south'	-2.12	'your'	2.70
'court'	-2.11	'thing'	2.74

Note that negative correlation means that the feature is associated more with news.

As can be seen, the news set is defined by words about serious news topics, while the clickbait set has very common words and numbers. We used the top 50 highest ranked words from each class as features in our machine learning algorithm.

Data preprocessing

The scraping process initially produced HTML files/objects of websites. Using Python, we extracted the headlines from this data and saved them in CSV files.

To do textual analysis on the headlines, we used Python and the NLTK library to tokenize the headlines and stem the words. Stopwords were removed from the headlines but excluding personal pronouns. Our intuition was that these would be an important feature considering our definition of clickbait. Features were extracted, including significant word counts, some patterns that we found are common in clickbait, and some features that are commonly used in NLP like sentence and word length and the occurrences of different word classes in a sentence. This way, the headlines were encoded into feature vectors that would be used in the machine learning step.

The complete list of features we used on top of the significant words is: number of words written in all caps, instances of the phrase “how to”, average word length, whether the first word is “what”, sentence length, and the number of 2-digit numbers, dates, nouns, adjectives, pronouns, superlatives and stopwords.

Machine Learning

We selected four possible algorithms to use in our task: Naive Bayes classifier, Logistic Regression, Support vector machines classification, and K-nearest neighbours classification. To pick the best one, we trained and tested each one with the same partition of the labelled clickbait corpus and performed a Wald test on the accuracies. We used the hypothesis that the difference in accuracy of each algorithm pair was not significant, or in other words that the differences in the sample accuracies we calculated didn’t indicate a difference in the actual accuracies of the algorithms on this task. The results indicated that there was evidence that Logistic Regression performed the best with the training data, and we selected it to be used to label the data we had gathered.

Model analysis

Due to limited time, we did not have much time to analyse our model or do diagnostics. The most highly correlated features included mostly topic words. These can be found in the Python script. The accuracy of the Logistic Regression model on the test data was ~90,6%, which is probably good considering the task: we are only interested in the estimated amount of clickbait per news site, and ~10% error probably wouldn’t affect this severely or will affect all sites in the same proportion. Of the incorrectly classified headlines, 35% were news headlines that were classified as clickbait, and 65% were clickbait headlines that were classified as news. This shows that the model was biased towards the news label. It would have been interesting to analyse how the labeled falsely

as clickbait vs. as news error rates affect the calculated proportions of clickbait per news site, which we will present in the next section.

Results

Our data analysis provided the results as CSV files for each analysed website. These files included the names of the headlines, and whether that headline was classified as a news article or clickbait article. From this, we calculated the percentage of articles that weren't classified as clickbait articles. The percentage of clickbait articles was the highest for Buzzfeed and lowest for BBC News.

We ordered the sites based on this percentage and plotted the results as a bar chart, showing results for each site individually. This chart can be found in the associated blog post.

Deliverable

Our deliverable is a blog post we made on GitHub Pages. This blog post explains what clickbait is and why it can become a problem in the news sites. We are also presenting our solution, the algorithm that analyses web sites and provides results in terms of the percentage of clickbait in all news on the analysed sites.

The blog post also shows the results we have found so far, including visualisations. We have added a bar chart showing the results for all analysed sites in the blog post.

Possible future steps

In the future, an analysis of more websites could be done to provide more overall information about the state of the news sites. It could be possible to analyse the news for multiple different countries as well and doing that might provide information about the sites with the least clickbait in the given country.

It is also a consideration to expand this program into a browser extension, which automatically removes clickbait articles. This would provide clear value to the user who prefers to find news with no clickbait headlines.

References

[1] Singh, Vikas. *News Clickbait Dataset*. Kaggle.com. https://www.kaggle.com/datasets/vikassingh1996/news-clickbait-dataset Accessed 15.10.2025.