**Abstract**

This research seeks to classify a headline as clickbait or non-clickbait using natural language processing and text categorization.

The term "clickbait" refers to an article title that uses tabloid style language to persuade readers to visit a specific website. The user's clicks often produce ad money, and the user's activity data is typically monetized. The clickbait content itself isn't written with any meaningful or useful purpose, but it's just a way to get people to click.

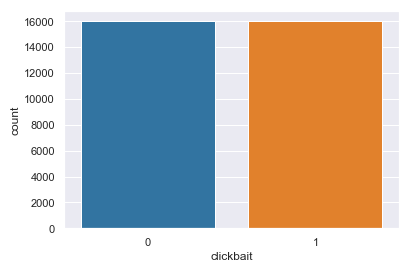
With the increasing popularity of social media and the number of people using the internet every day, we cannot talk about a shortage of information and there are always digital products fighting for our attention.

**Introduction**

For this assignment we have tried to classify the headlines as clickbait or not. We have used different NLP techniques (TFIDF, Word2vec, NLTK) on our texts, and 5 classification methods to see which would perform better. We have tried both with and without preprocessing the texts and we also have done some data visualization.

**Data**

We have used a dataset from Kaggle which has 32000 headlines, almost equally split in clickbait and not clickbait.

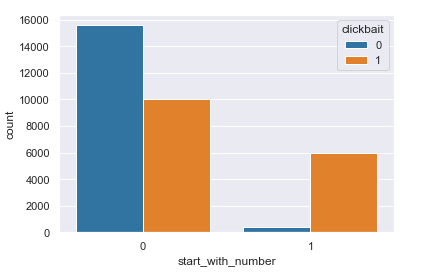
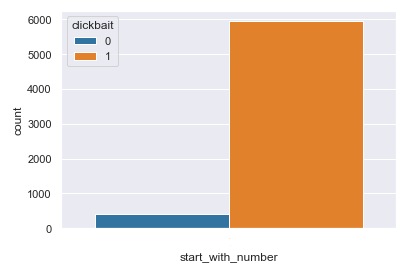


Where 0 represents not clickbait headlines and 1 represents clickbait headlines.

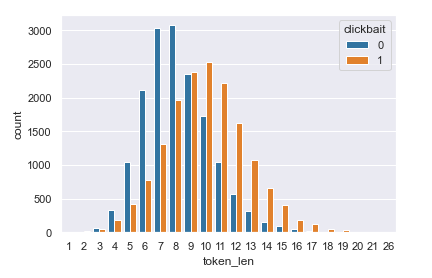
**Feature engineering & data visualization & text cleaning**

We have discovered that much more clickbait headlines start with a number such as: “17 Places All Music Lovers Should Go In London”

So we thought it would be a good idea to have a feature called “ start with number”.



We also added another feature named “token length” which measures how many words are in a headline.



We have tried running our models both with and without preprocessing, and we have found out that without preprocessing the models perform better.

But in terms of preprocessing we have tried:

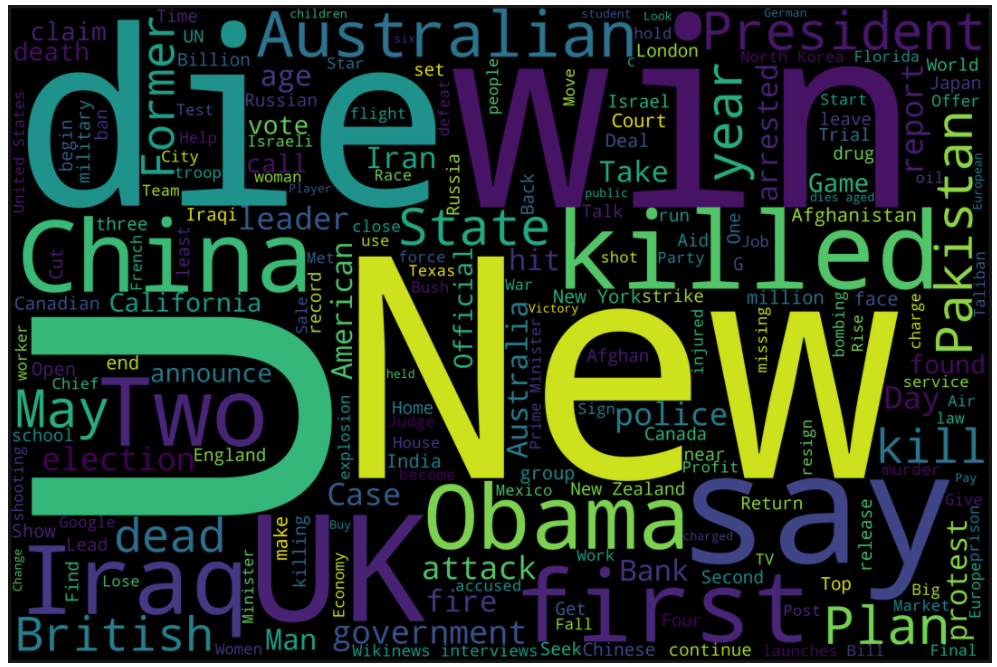
* getting rid of punctuation and additional spaces using RegEx library
* lowercase capitalized words
* removing stopwords and tokenize/lemmatize words using NLTK library

For data visualization purposes we have also used Wordcloud to see the most frequent words used in clickbait and not clickbait headlines:

Clickbait headlines:



Not clickbait headlines:



We have used TFidF with NLTK implementation and Word2vec to see which one would make our classification models perform better.

We have concatenated the TfidF features with the two features created by us (token length and start with number).

**Models**

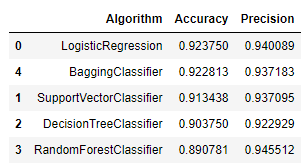
For the modelling part we have implemented the following 5 classification models:

* Logistic Regression
* SVM
* Decision Tree Classifier
* Random Forest Classifier
* Bagging Classifier

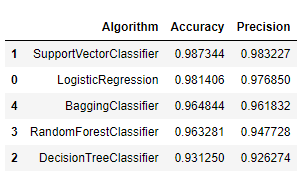
We have used parameter tuning for each model and we have computed the confusion matrixes and classification reports.

After that, we have sorted our models by accuracy to have a better perspective at which has performed better.

1. Results using TFidF



1. Word2Vec



**Coclusion**

The clickbait problem will not disappear very soon as for the moment this is a method of making money from ads. Even though some headlines can be clearly labeled as clickbait, some people still tend to click on those links out of curiosity. This is a big problem because clickbait can be strong related with fake news and disinformation. So a model to classify headlines as clickbait or not could definitely help some people not to believe everything they read on the internet, especially these days when each day we encounter loads and loads of information.