# Analysing the Popularity of Online News: A Machine Learning Approach

Machine Learning - Final Group Project

Radosław Dawidowski; Aleksandra Hryncyszyn; Giacomo Magnocavallo

## Introduction

In the digital age, online news consumption has become a crucial part of daily life, influencing public opinion and shaping societal discourse. The vast array of available content has led to an intensely competitive landscape where the popularity of news articles can significantly impact a media company's profitability. Understanding what drives readers to consume some articles over others is not only of academic interest but is also critical for publishers seeking to optimise their content strategy in the ever-evolving digital media space.

Thus, this project aims to unravel the mechanisms behind the popularity of online news articles. Specifically, the focus is on identifying and analysing various factors to determine their influence on an article's appeal to its audience. The factors taken into account range from the structural elements of the articles, such as word count and title length, to more subjective aspects like content sentiment and topic. By leveraging a dataset of 39,797 online news articles, this study seeks to classify articles based on their popularity and delve into the characteristics that distinguish popular articles from less popular ones.

The findings of this research could be valuable for multiple stakeholders. For content creators and journalists, it could provide insights into audience preferences, guiding them in crafting more engaging and impactful stories. For publishers and marketers, it aids in strategizing content dissemination and enhancing user engagement, which is key to maintaining visibility and relevance in the digital landscape. Furthermore, for the general public and academic researchers, this analysis offers a window into the dynamics of information consumption and dissemination in the digital era, contributing to a broader understanding of media influence and its implications for society.

## Methodology

The dataset used in this project is the "Online News Popularity" dataset, which comprises 39,797 news articles, each characterised by 61 attributes. These attributes include various metrics related to the content of the articles, such as the number of words in the title (n_tokens_title), the number of words in the content (n_tokens_content), as well as more specific features like the number of images (num_imgs) and videos (num_videos). Additionally, the dataset includes various metrics related to the sentiment and subjectivity of the content.

The initial exploration began with loading the dataset using the pandas library in Python, followed by an examination of its structure using head() and describe() functions. It was observed that most of the data is numerical, except for the 'url' variable that was discarded as irrelevant for the classification purposes. Moreover, for the n_unique_tokens, all of the variables had values below 1, except for one entry with value 701, it was thus discarded as an outlier. In reviewing the dataset of 39,797 news articles, we found an interesting trend in share counts. While most articles received shares between 1 and 1400, some stood out with exceptionally high shares, reaching up to 843,300. Initially considered outliers, we decided to retain these instances as they likely represent viral articles. Keeping them provides a comprehensive view of the wide range of shares articles can attract, including exceptional cases of widespread engagement.

As the aim of this study was to classify the popularity of the articles, a new binary variable was formed based on the values of 'shares' feature. As the describe() method indicated that the median for 'shares' was equal to 1400, it was used as a threshold for determining if an article can be considered as popular or not. The new, binary variable called 'popular_article' assumes values 1 for the entries with 'share' value larger or equal to 1400, whereas the rest of the entries assume value 0. Such an approach allows for more focused analysis of the factors underlying articles' popularity using classification.

Firstly, the dataset was divided into training and testing sets, with 80% of the data allocated for training. This split enabled to train models on a substantial data portion and then assess their performance and generalisation ability on the independent test set. Then, using all of the available features, four classification models were built - Logistic Regression, k-Nearest Neighbors algorithm (kNN), Linear Support Vector Classification (LinearSVC), and Random Forest. To determine optimal hyperparameters settings, GridSearchCV was used, a method that systematically works through multiple combinations of parameter tunes, cross-validating as it goes to determine which tune gives the best performance. For Logistic Regression and LinearSVC, the parameter 'C', which controls the strength of regularisation, was fine-tuned. Regularisation is critical in preventing overfitting, especially in datasets with many features. The kNN model's key parameter, 'n_neighbors' (which determines the number of neighbours to use for making predictions), was also optimised to ensure the model accurately captures the underlying trends in the data. Lastly, for the Random Forest model, parameters like 'max_depth,' 'max_features,' and 'n_estimators' were fine-tuned to balance the model's complexity and its ability to generalise. On top of that, GridSearchCV was also designed to include different data scaling methods (StandardScaler, MinMaxScaler, or none). Then, using the test set, performances with optimal hyperparameters were evaluated and compared.

Subsequently, to improve the performance of the models by decreasing the noise and their overfitting, three feature selection approaches were investigated. The first one relied on plotting the distributions of each variable and distinguishing the data coming from popular articles by red colour and those coming from the unpopular ones by blue colour. As can be observed on the picture below, the resulting purple colour indicates the areas where both histograms overlap. In general, most of the 58 plots were dominated by the purple shade, indicating that the differences between popular and unpopular articles aren't very distinct. Nevertheless, 18 features were selected where the red colour was the most distinguishable.

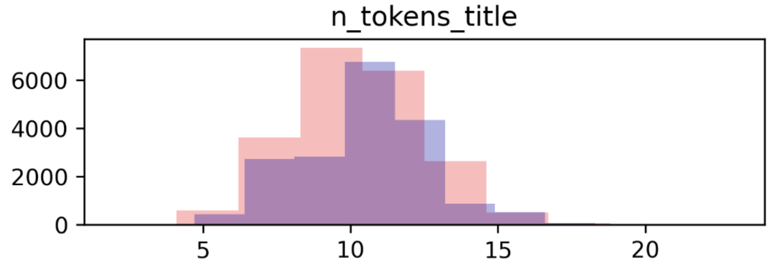
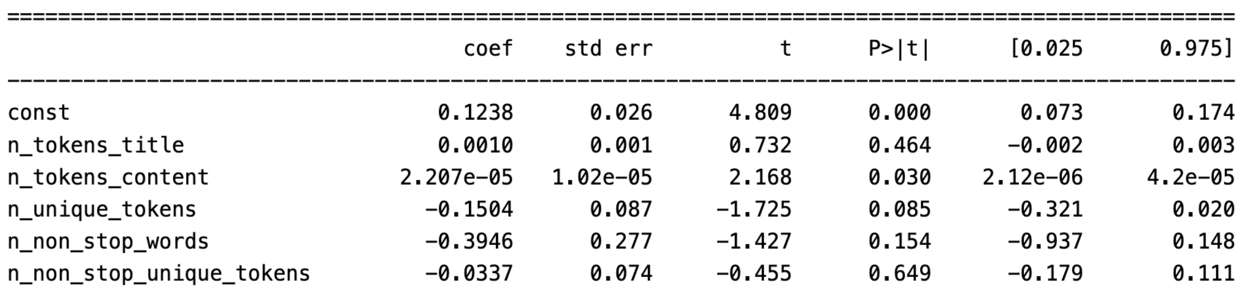


Figure 1 - distribution of n_tokens_title with popular (red) and unpopular articles (blue)

The second approach involved building a linear regression model predicting the values of 'shares' variable. Using the statsmodels library, statistical information was displayed for each of the variables used in the model (see: Figure 2). Based on the p-statistic, we selected 38 features that were statistically significant (p-value below 0.05).



```
==============================================================================================
                                  coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------------------
const                           0.1238      0.026      4.809      0.000       0.073       0.174
n_tokens_title                  0.0010      0.001      0.732      0.464      -0.002       0.003
n_tokens_content             2.207e-05   1.02e-05      2.168      0.030    2.12e-06     4.2e-05
n_unique_tokens                -0.1504      0.087     -1.725      0.085      -0.321       0.020
n_non_stop_words               -0.3946      0.277     -1.427      0.154      -0.937       0.148
n_non_stop_unique_tokens       -0.0337      0.074     -0.455      0.649      -0.179       0.111
```

Figure 2 - first 5 variables from OLS regression on Popularity Articles data set

Lastly, a Decision Tree Classifier was fitted on all of the available features along with the 'article_popularity' as a target. Firstly, the GridSearchCV was applied to determine the optimal combination of max_depth and max_features hyperparameter values to avoid overfitting of the model. Subsequently, utilising graphviz package, the importance of each feature used in the decision process was visualised. Resulting report included 30 variables, most of which overlapped with the previous feature selection methods. Finally, combining all of the features extracted in those three methods resulted in 48 selected variables. Then, the same classification models were fitted on this reduced dataset and the performance metrics were compared.

The second part of the analysis included three manifold learning algorithms - Principal Component Analysis (PCA), t-Distributed Stochastic Neighbour Embedding (t-SNA), and Uniform Manifold Approximation and Projection (UMAP). The aim was to reduce the dimensionality of our dataset for the sake of clearer and more informative data visualisation. A clustering algorithm was employed to visualise groups among popular and unpopular articles, aiming to identify influential variables affecting popularity trends. Upon thorough analysis, it was observed that while clusters emerged within the dataset based on certain variables, they did not align with our target variable. Consequently, it was determined that clustering is not an ideal method for addressing our research question.

## Results and Discussion

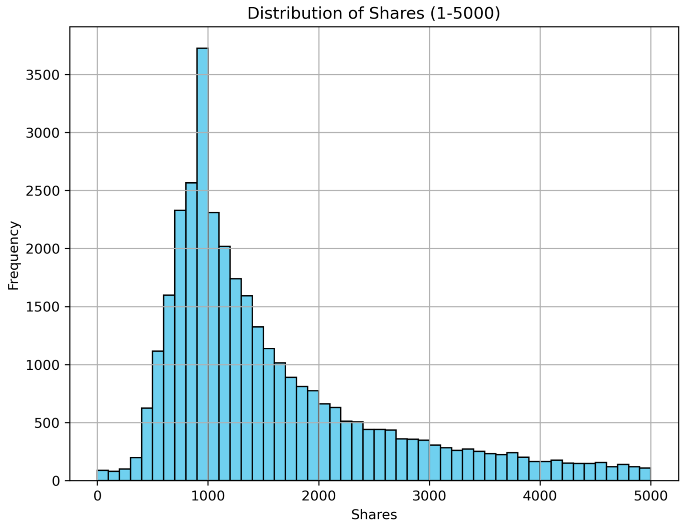


Figure 3 - distribution of shares from Popularity Articles

Firstly, the distribution of the variable 'shares', on which our target variable 'article_popularity' relies, was visualised using a boxplot (Figure X).

It can be observed that most of the articles had the number of shares around 1000. However, the median is equal to 1400 as there are many more records on the right side, up to the maximal value of 843 300 shares. However, the axis of this plot was limited for the sake of readability.

Then, GridSearchCV was used to determine the optimal hyperparameter values and preprocessing method on each of the classification algorithms.

Results are summarised in Table 1. One thing that can be readily noticed is that in each case the StandardScaler() method was preferred. Then, for both Logistic Regression and SVC it can be observed that the regularisation parameter C assumes low values, indicating strong regularisation. This makes sense, as our dataset is multidimensional, thus prone to overfitting. Then, quite a large n_neighbours value in kNN algorithm might imply that the dataset includes some noise, negatively influencing the results for low n_neighbours values. Finally, the optimal Random Forest model is quite a limited one, which is also in accordance with the expectation as this algorithm is the most prone to overfitting.

Table 1 - GridSearchCV - generated optimal parameters for four classification algorithms using all available features

| | LR | SVC | kNN | Random Forest |
|---|---|---|---|---|
| parameters | C = 0.119 | C = 0.00289 | n_neighbours = 91 | max_depth = 9<br>max_features = 21<br>n_estimators = 20 |
| preprocessing | StandardScaler() | StandardScaler() | StandardScaler() | StandardScaler() |

The same GridSearchCV algorithm resulted in different hyperparameter values for the dataset with limited features (see Table 2). Again, the StandardScaler() preprocessing method is preferred. This time, however, the regularisation parameter in LR and SCV is larger than before, indicating that less regularisation is required. Similarly, the optimal number of neighbours in the kNN algorithm has decreased, which might indicate less noise in the data. Finally, the Random Forest classifier was optimised to be even more limited than last time, using only 8 most meaningful features over the depth of 8 levels.

Table 2 - GridSearchCV - generated optimal parameters for four classification algorithms using selected features

| | LR | SVC | kNN | Random Forest |
|---|---|---|---|---|
| parameters | C = 0.345 | C = 1.0 | n_neighbours = 59 | max_depth = 8<br>max_features = 8<br>n_estimators = 20 |
| preprocessing | StandardScaler() | StandardScaler() | StandardScaler() | StandardScaler() |

Then, all of the models were fit on the data and evaluated using the test set. For the models using all available features, score values ranged from 0.6457 (SVC) to 0.6567 (Random Forest), indicating that all of the models exhibited quite mediocre performance. However, the test set results have similar values, which implies that there's no strong overfitting issue. Alternatively, it might be the case that the dataset is not representative of all of the factors influencing the popularity of the articles, or that the collected data is not reliable. This might be the case especially as some of the variables rely on subjective impressions.

Table 3 - Test set and train set scores for four classifiers using all of the available features

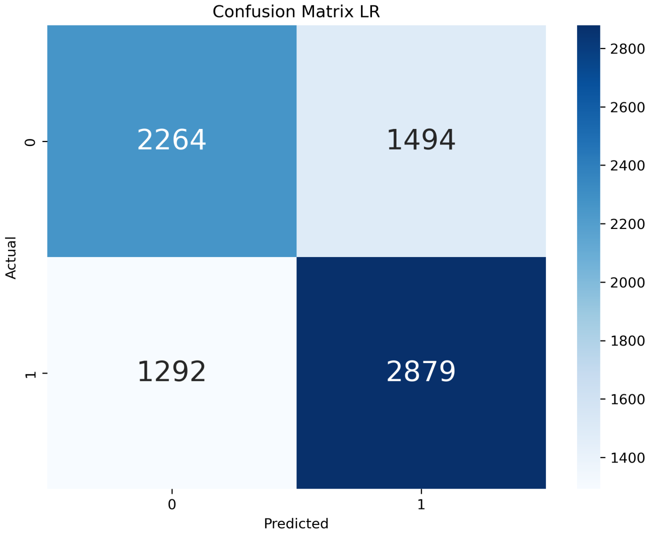
| | Logreg | SVC | kNN | Random Forest |
|---|---|---|---|---|
| Test set score | 0.6471 | 0.6457 | 0.6563 | 0.6567 |
| Train set score | 0.6562 | 0.6563 | 0.6365 | 0.7403 |

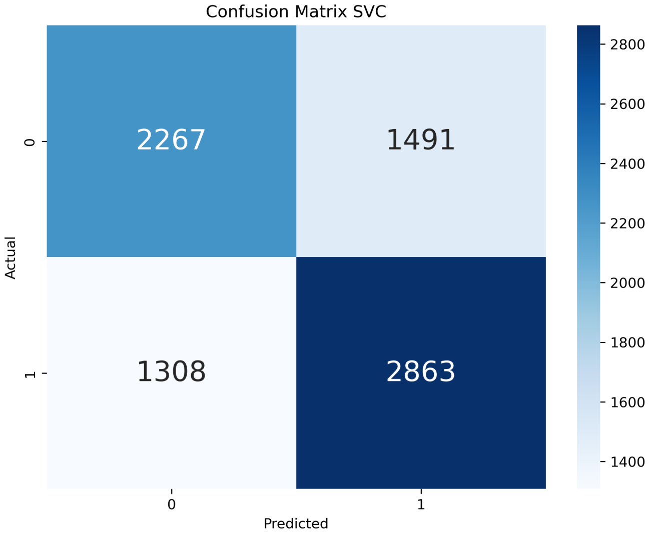
Similar case was for the scores from the models that used selected features. It turned out that the worst test score (0.6387, kNN) was actually slightly worse than the worst score from models on all available features (0.6457, SVC). However, for the rest of the models, the score has improved. Most importantly, the best score (0.6592, RF) is slightly larger than the best score from the previous models (0.6567, RF).

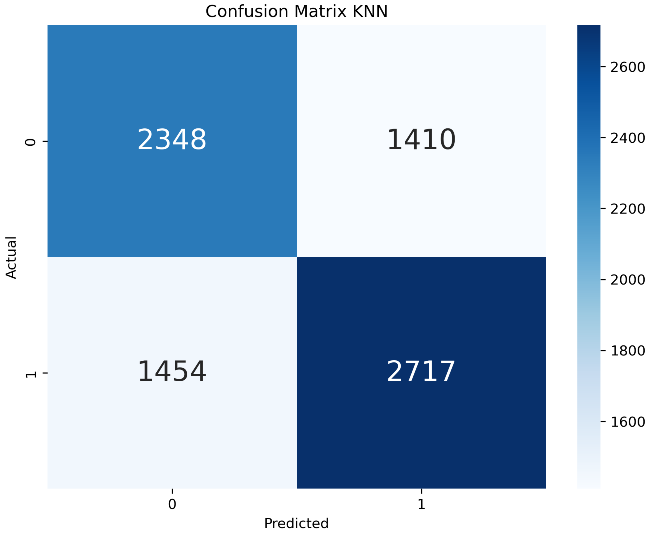
Table 4 - Test set and train set scores for four classifiers using selected features

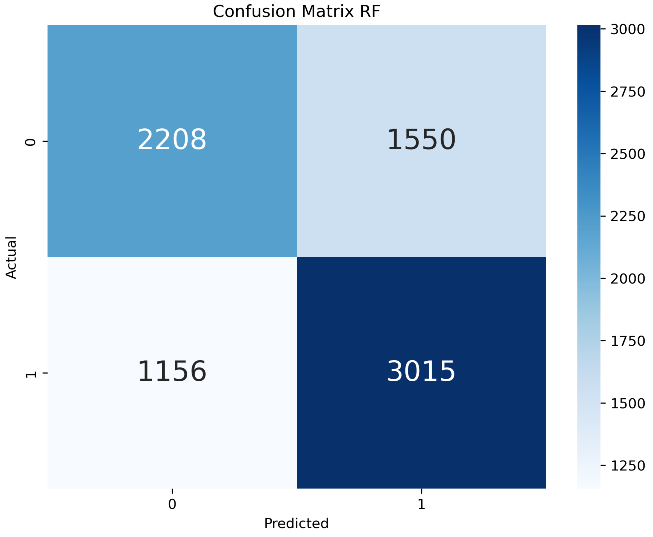
| | Logreg | SVC | kNN | Random Forest |
|---|---|---|---|---|
| Test set score | 0.6486 | 0.6469 | 0.6387 | 0.6592 |
| Train set score | 0.6555 | 0.6557 | 0.6606 | 0.7053 |

To further delve into the performance of models using selected features, confusion matrices were plotted for each of them.

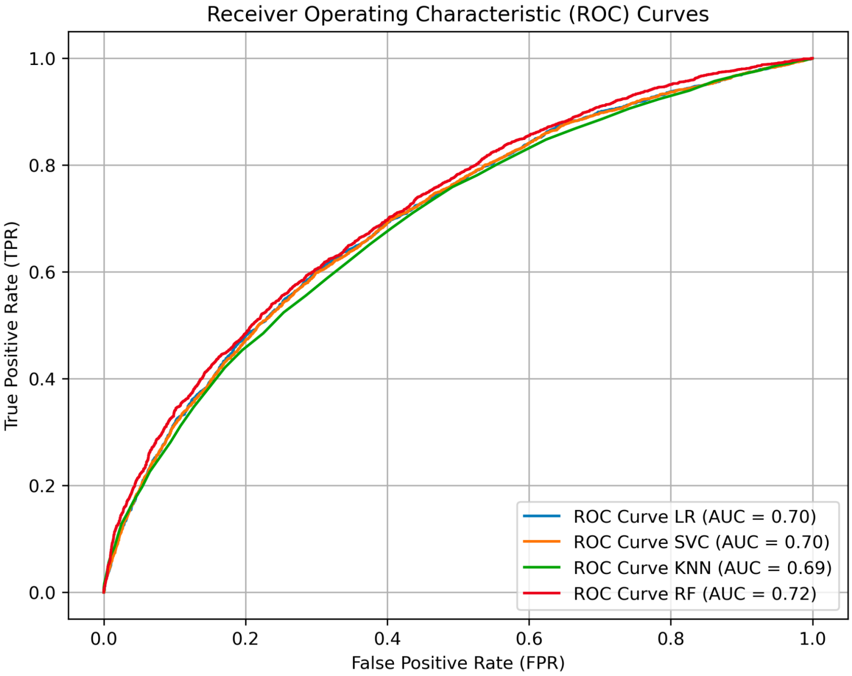

Confusion Matrix LR


Confusion Matrix SVC


Confusion Matrix KNN

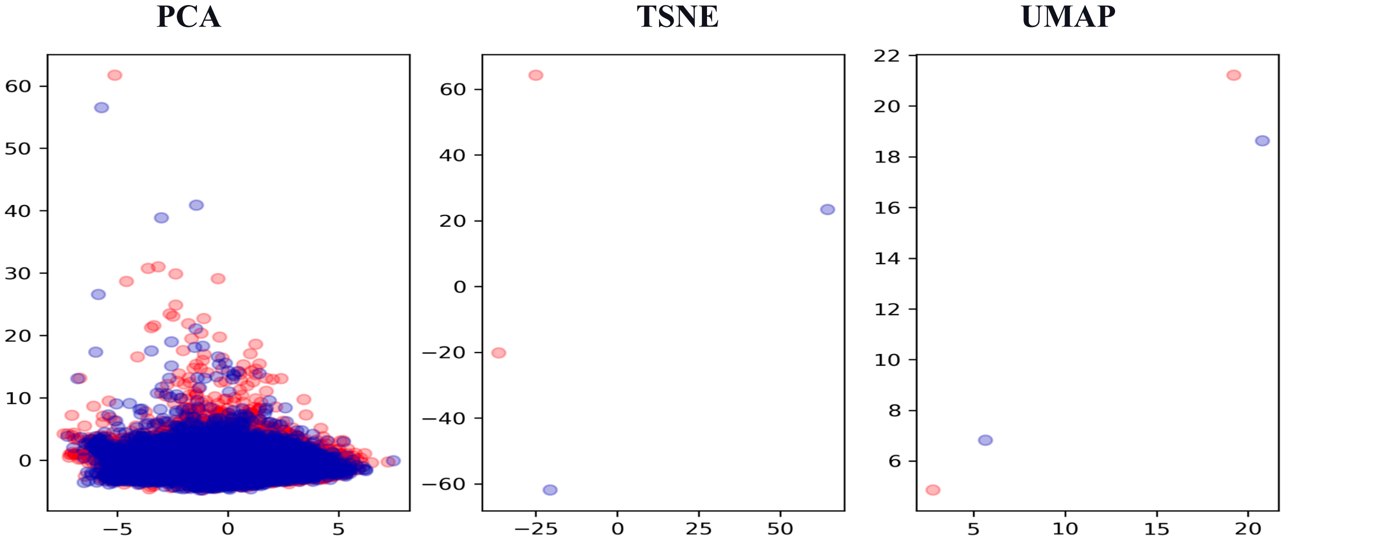

Confusion Matrix RF

It can be observed that even though the Random Forest algorithm has the best overall performance, it actually results in the highest number of False Positive predictions. For situations where the minimization of False Positive errors is important, actually the kNN algorithm would be the best choice.
Finally, to visualise the performance of the models, also ROC curves were plotted and the area under them (AUC) calculated.



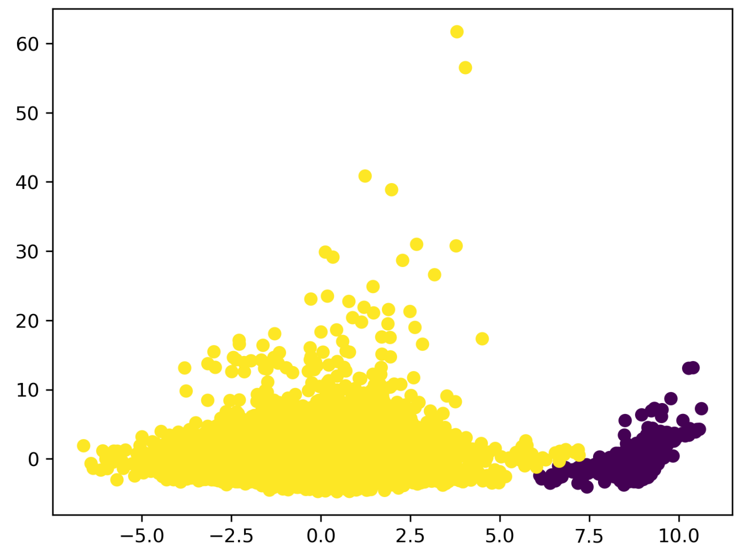
The ROC curves serve as visual representations of how effectively each model distinguishes popular and less popular articles, showcasing the trade-off between correctly identifying popular articles (True Positives) and minimising misclassifications (False Positives) across various classification thresholds. Logistic Regression and Support Vector Machine (SVM) display a moderate discriminatory ability, while the k-Nearest Neighbors (kNN) model shows slightly weaker discrimination. Conversely, the Random Forest model stands out with the highest AUC of 0.72, indicating its superior capability in correctly classifying article popularity levels by achieving higher true positive rates while minimising false positives, thus emphasising a favourable trade-off between sensitivity and specificity.

The second part of the analysis attempted to use feature reduction to visualise the dataset and apply clustering algorithms.



When visualised using PCA, our data doesn't exhibit clear clusters, suggesting that popular articles bear strong similarity to unpopular ones. On the other hand, TSNE and UMAP offer a clearer distinction between these classes, showcasing a significant separation between popular and unpopular articles. However, owing to the data's lack of robustness and its peculiar appearance, any conclusions drawn from these insights should be approached without attaching significance.

**K-means cluster**



Upon applying K-means clustering to our dataset, it delineated the data into 2 classes. However, this clustering doesn't correspond with our earlier visualisations, indicating that the clustering might have occurred due to factors unrelated to an article's popularity—perhaps related to topics, sentiment, or sensitivity. Notably, these variables do not decisively dictate an article's popularity, as articles with similarities in these aspects can vary widely in their popularity.

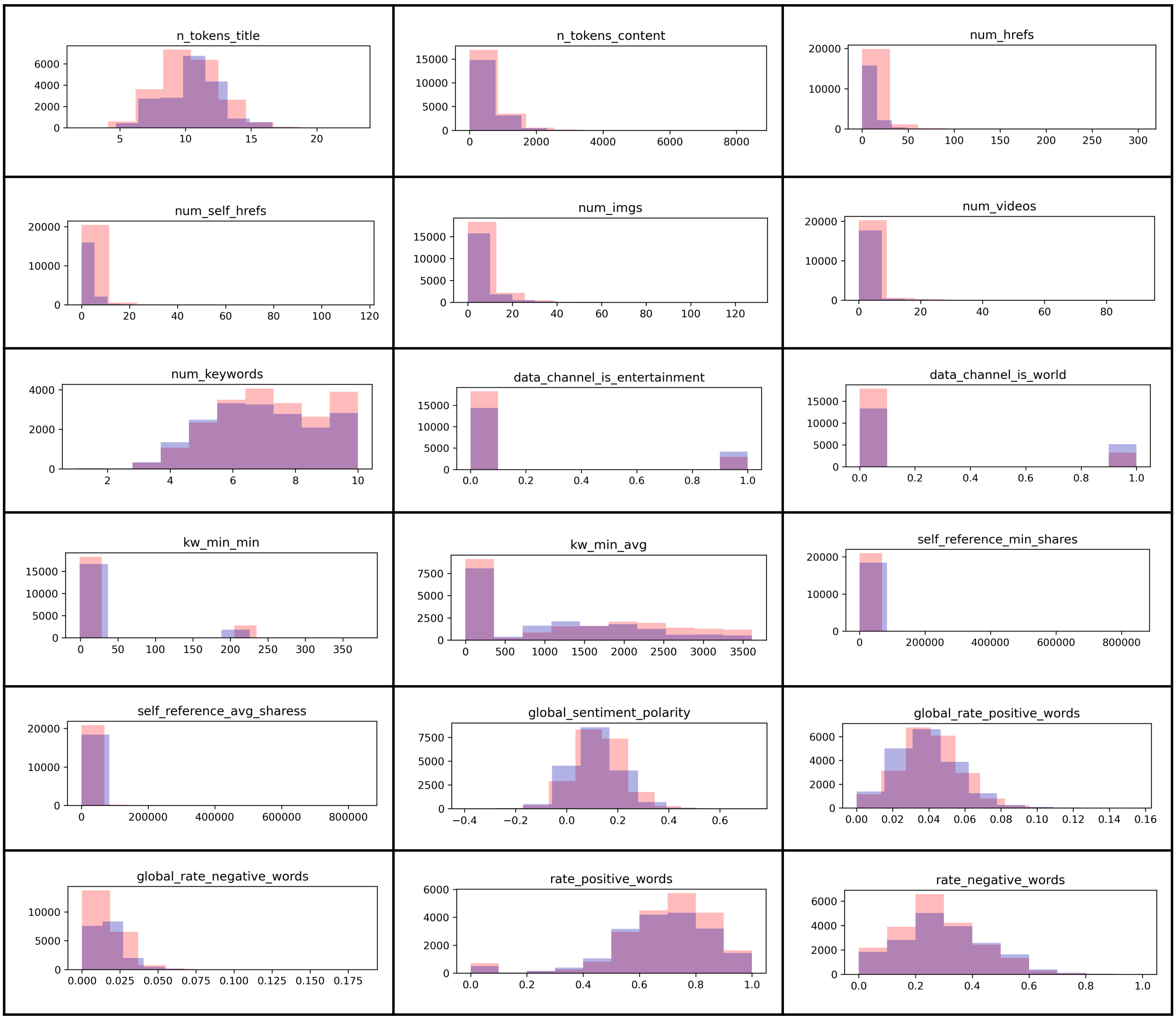
## Conclusions

Based on the comprehensive analysis of factors influencing online news article popularity, this study uncovered essential insights into the intricate dynamics of article engagement. Through meticulous exploration and classification models, it was evident that while distinguishing popular from less popular articles remains a challenging task, certain features contribute more significantly to this differentiation. Feature selection techniques aided in identifying crucial attributes, refining models to enhance predictive accuracy. Notably, the Random Forest model exhibited superior performance in classifying article popularity, showcasing promising potential for refining content strategies. However, it's crucial to note that the classification was far from a perfect one, implying room for improvement within data collection and the choice of features.

Moreover, visualisations like PCA, t-SNE, and UMAP highlighted potential separations between popular and unpopular articles, though their robustness in predicting article engagement warrants cautious interpretation.

Overall, this study provides valuable insights for content creators, publishers, and researchers, shedding light on the multifaceted nature of online news consumption, albeit with a need for further exploration and refinement in modelling approaches.

# Appendix 1


n_tokens_title
0 2000 4000 6000
5 10 15 20
n_tokens_content
0 5000 10000 15000
0 2000 4000 6000 8000
num_hrefs
0 10000 20000
0 50 100 150 200 250 300
num_self_hrefs
0 10000 20000
0 20 40 60 80 100 120
num_imgs
0 5000 10000 15000
0 20 40 60 80 100 120
num_videos
0 10000 20000
0 20 40 60 80
num_keywords
0 2000 4000
2 4 6 8 10
data_channel_is_entertainment
0 5000 10000 15000
0.0 0.2 0.4 0.6 0.8 1.0
data_channel_is_world
0 5000 10000 15000
0.0 0.2 0.4 0.6 0.8 1.0
kw_min_min
0 5000 10000 15000
0 50 100 150 200 250 300 350
kw_min_avg
0 2500 5000 7500
0 500 1000 1500 2000 2500 3000 3500
self_reference_min_shares
0 10000 20000
0 200000 400000 600000 800000
self_reference_avg_sharess
0 10000 20000
0 200000 400000 600000 800000
global_sentiment_polarity
0 2500 5000 7500
−0.4 −0.2 0.0 0.2 0.4 0.6
global_rate_positive_words
0 2000 4000 6000
0.00 0.02 0.04 0.06 0.08 0.10 0.12 0.14 0.16
global_rate_negative_words
0 5000 10000
0.000 0.025 0.050 0.075 0.100 0.125 0.150 0.175
rate_positive_words
0 2000 4000 6000
0.0 0.2 0.4 0.6 0.8 1.0
rate_negative_words
0 2000 4000 6000
0.0 0.2 0.4 0.6 0.8 1.0