Project Report

Predictive Analysis for Online News Popularity

Name NetId

Fangjing Wu (fw179)

Maharshi Rohith Donthi (md1838)

Shridhar Ajit Ashtikar (sa2245)

Tanishq Sharma (ts1266)

Terry Xu (tx55)

INDEX

1. ABSTRACT

In this project, we aim to build models that help understand the factors that contribute to the popularity of online news articles and identify patterns that may influence readership. With this project, we aim to uncover key drivers of article popularity and provide actionable recommendations for content creators and digital publishers. Various metrics like tokens, the day of the week the article was published, categories of news, polarities, number of positive and negative words, etc. were taken into consideration while implementing various models for regression namely linear regression, Poisson regression, random forest, Artificial Neural Networks, and XGBoost.

In this process, data preprocessing and cleaning were performed to check for any null and duplicate values based on which the outliers were spotted and removed respectively. Additionally, exploratory data analysis was performed to fathom the correlation of each independent variable with the dependent variable. Observing the data post-analysis, we were able to start implementing models on metrics like mean squared error, AIC, and BIC (for R-squared).

2. DATA PREPROCESSING

2.1 DATASET DESCRIPTION

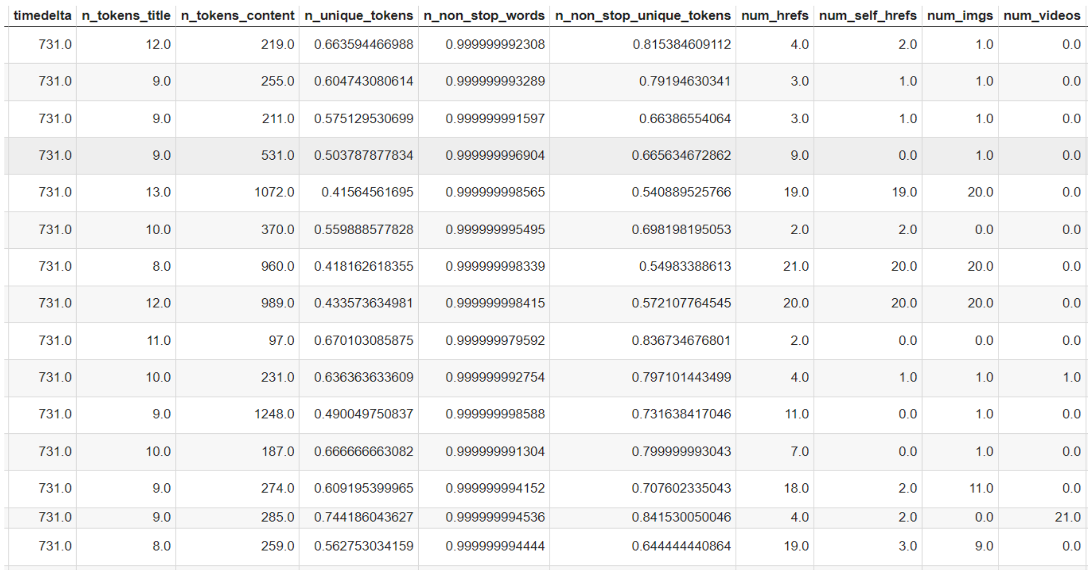
This dataset mainly comprises 61 variables, out of which 58 are predictive, 2 aren't and 1 is the target variable named "SHARES". The total number of attributes is 39797.

Attribute Information:

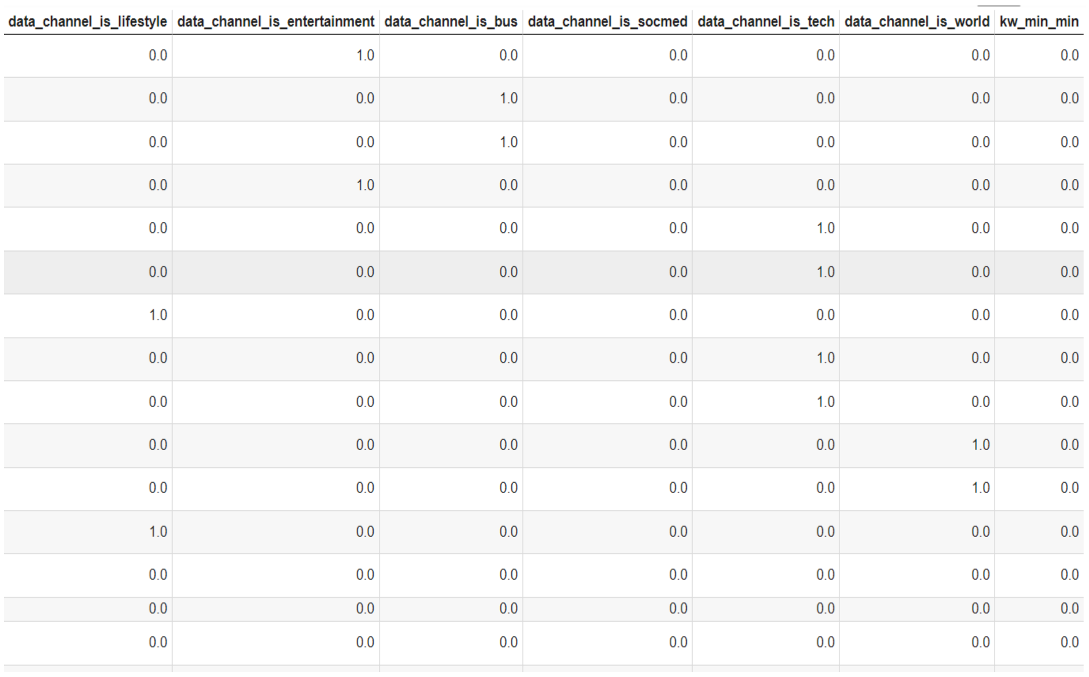
0. url:	URL of the article
1. timedelta:	Day between the article publication, the dataset acquired
2. n_tokens_title:	Number of words in the title
3. n_tokens_content:	Number of words in the content
4. n_unique_tokens:	Rate of unique words in the content
5. n_non_stop_words:	Rate of non-stop words in the content
6. n_non_stop_unique_tokens:	Rate of unique non-stop words in the content
7. num_hrefs:	Number of links
8. num_self_hrefs:	Number of links to other articles published by Mashable
9. num_imgs:	Number of images

10. num_videos:	Number of videos
11. average_token_length:	Average length of the words in the content
12. num_keywords:	Number of keywords in the metadata
13. data_channel_is_lifestyle:	Is the data channel 'Lifestyle'?
14. data_channel_is_entertainment:	Is data channel 'Entertainment'?
15. data_channel_is_bus:	Is the data channel 'Business'?
16. data_channel_is_socmed:	Is the data channel 'Social Media'?
17. data_channel_is_tech:	Is the data channel 'Tech'?
18. data_channel_is_world:	Is the data channel 'World'?
19. kw_min_min:	Worst keyword (min. shares)
20. kw_max_min:	Worst keyword (max. shares)
21. kw_avg_min:	Worst keyword (avg. shares)
22. kw_min_max:	Best keyword (min. shares)
23. kw_max_max:	Best keyword (max. shares)
24. kw_avg_max:	Best keyword (avg. shares)
25. kw_min_avg:	Avg. keyword (min. shares)
26. kw_max_avg:	Avg. keyword (max. shares)
27. kw_avg_avg:	Avg. keyword (avg. shares)
28. self_reference_min_shares:	Min. shares of referenced articles in Mashable
29. self_reference_max_shares:	Max. shares of referenced articles in Mashable
30. self_reference_avg_sharess:	Avg. shares of referenced articles in Mashable
31. weekday_is_monday:	Was the article published on a Monday?
32. weekday_is_tuesday:	Was the article published on a Tuesday?
33. weekday_is_wednesday:	Was the article published on a Wednesday?
34. weekday_is_thursday:	Was the article published on a Thursday?
35. weekday_is_friday:	Was the article published on a Friday?
36. weekday_is_saturday:	Was the article published on a Saturday?

37. weekday_is_sunday:	Was the article published on a Sunday?
38. is_weekend:	Was the article published on the weekend?
39. LDA_00:	Closeness to LDA topic 0
40. LDA_01:	Closeness to LDA topic 1
41. LDA_02:	Closeness to LDA topic 2
42. LDA_03:	Closeness to LDA topic 3
43. LDA_04:	Closeness to LDA topic 4
44. global_subjectivity:	Text subjectivity
45. global_sentiment_polarity:	Text sentiment polarity
46. global_rate_positive_words:	Rate of positive words in the content
47. global_rate_negative_words:	Rate of negative words in the content
48. rate_positive_words:	Rate of positive words among non-neutral tokens
49. rate_negative_words:	Rate of negative words among non-neutral tokens
50. avg_positive_polarity:	Avg. polarity of positive words
51. min_positive_polarity:	Min. polarity of positive words
52. max_positive_polarity:	Max. polarity of positive words
53. avg_negative_polarity:	Avg. polarity of negative words
54. min_negative_polarity:	Min. polarity of negative words
55. max_negative_polarity:	Max. polarity of negative words
56. title_subjectivity:	Title subjectivity
57. title_sentiment_polarity:	Title polarity
58. abs_title_subjectivity:	Absolute subjectivity level
59. abs_title_sentiment_polarity:	Absolute polarity level
60. shares:	Number of shares (target)



timedelta	n_tokens_title	n_tokens_content	n_unique_tokens	n_non_stop_words	n_non_stop_unique_tokens	num_hrefs	num_self_hrefs	num_imgs	num_videos
731.0	12.0	219.0	0.663594466988	0.999999992308	0.815384609112	4.0	2.0	1.0	0.0
731.0	9.0	255.0	0.604743080614	0.999999993289	0.79194630341	3.0	1.0	1.0	0.0
731.0	9.0	211.0	0.575129530699	0.999999991597	0.66386554064	3.0	1.0	1.0	0.0
731.0	9.0	531.0	0.503787877834	0.999999996904	0.665634672862	9.0	0.0	1.0	0.0
731.0	13.0	1072.0	0.41564561695	0.999999998565	0.540889525766	19.0	19.0	20.0	0.0
731.0	10.0	370.0	0.559888577828	0.999999995495	0.698198195053	2.0	2.0	0.0	0.0
731.0	8.0	960.0	0.418162618355	0.999999998339	0.54983388613	21.0	20.0	20.0	0.0
731.0	12.0	989.0	0.433573634981	0.999999998415	0.572107764545	20.0	20.0	20.0	0.0
731.0	11.0	97.0	0.670103085875	0.999999979592	0.836734676801	2.0	0.0	0.0	0.0
731.0	10.0	231.0	0.636363633609	0.999999992754	0.797101443499	4.0	1.0	1.0	1.0
731.0	9.0	1248.0	0.490049750837	0.999999998588	0.731638417046	11.0	0.0	1.0	0.0
731.0	10.0	187.0	0.666666663082	0.999999991304	0.799999993043	7.0	0.0	1.0	0.0
731.0	9.0	274.0	0.609195399965	0.999999994152	0.707602335043	18.0	2.0	11.0	0.0
731.0	9.0	285.0	0.744186043627	0.999999994536	0.841530050046	4.0	2.0	0.0	21.0
731.0	8.0	259.0	0.562753034159	0.999999994444	0.644444440864	19.0	3.0	9.0	0.0



data_channel_is_lifestyle	data_channel_is_entertainment	data_channel_is_bus	data_channel_is_socmed	data_channel_is_tech	data_channel_is_world	kw_min_min
0.0	1.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	1.0	0.0	0.0	0.0	0.0
0.0	0.0	1.0	0.0	0.0	0.0	0.0
0.0	1.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	1.0	0.0	0.0
0.0	0.0	0.0	0.0	1.0	0.0	0.0
1.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	1.0	0.0	0.0
0.0	0.0	0.0	0.0	1.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	1.0	0.0
0.0	0.0	0.0	0.0	0.0	1.0	0.0
1.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0

Fig 1.

2.2 DATA CLEANING

Initially, while performing data preprocessing and cleaning, certain observations were made and resolved. Some of them include:

a) There were variables named weekday_is_monday till weekday_is_sunday. All the data in those columns were binary, i.e., 0 and 1. We tried grouping the column entities into a single column but removed binary values and numbered them based on the day of the week. For instance, Monday was numbered 1, Tuesday was 2, and so on. But by doing so, we could not just eliminate 7 columns, but also improve the analysis as each day had a unique value representing it. The new column is now named – day_count

b) There are categories of news namely: data_channel_is_entertainment, data_channel_is_bus, data_channel_is_socmed, data_channel_is_news, etc. Overall, there were 6 different variables. These variables again had binary distribution which was again labeled as per order. So, overall, they are indexed from numbered 1 to 6. This helped us easily identify which category of news we were referring to. This new column is now called merged_data_channel.

With this cleaning process being performed, we were able to reduce the number of columns to 48. As for reducing the number of attributes, we tried figuring out the number of outliers in each case by figuring out the interquartile range (difference of quartile 3 and quartile 1), taking the target variable as SHARES, which helped us bring the number of attributes to be analyzed to 32784.

Next, all the attributes were plotted to take out their respective distributions for further observation. In this process, we deep-dived into the target variable distribution wherein the use of histogram, box plot, violin plot, and kde-plot was plotted. This gave us a fair idea about the target variable we are dealing with + a tap on the outliers (which weren't present).

In this process, a correlation heatmap was plotted to understand each independent variable's relation with the dependent variable. This gave us a clear picture of which independent variable to opt for during the remaining part of the analysis wherein various machine learning models were put to test.

2.3 EXPLORATORY DATA ANALYSIS

We tried to visualize the target variable (shares) to fathom its distribution. Following are the visuals:

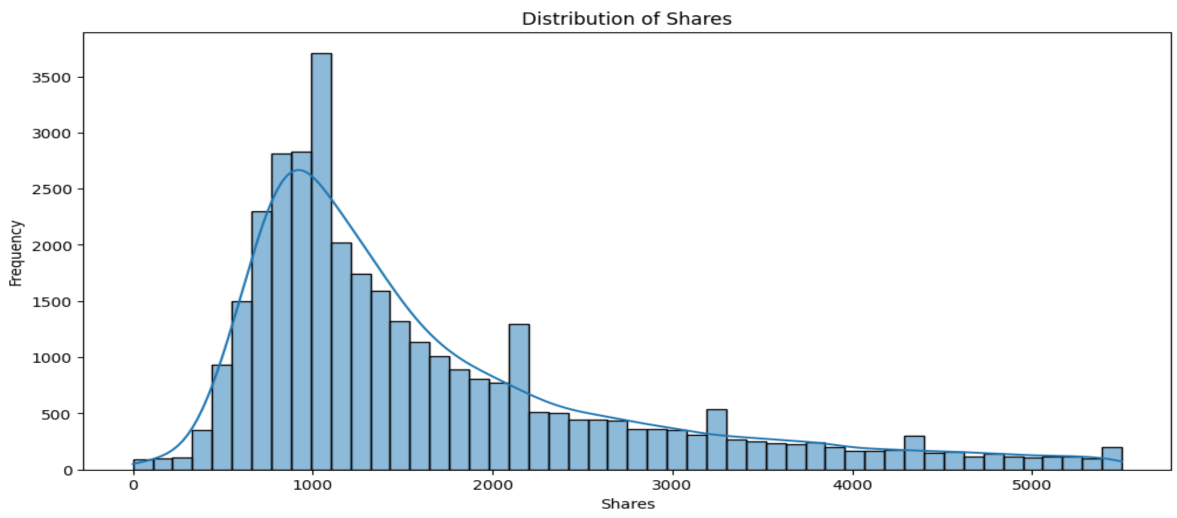


Fig 2.

The graph depicts a Poisson distribution, characterized by its discrete bars representing the probability of event occurrences. The peak suggests the most probable number of events, aligning with the Poisson's mean equals variance property.

Additionally, we figured out the correlation of independent variables with that of the dependent variables through a correlation heatmap, whose visual we have below:

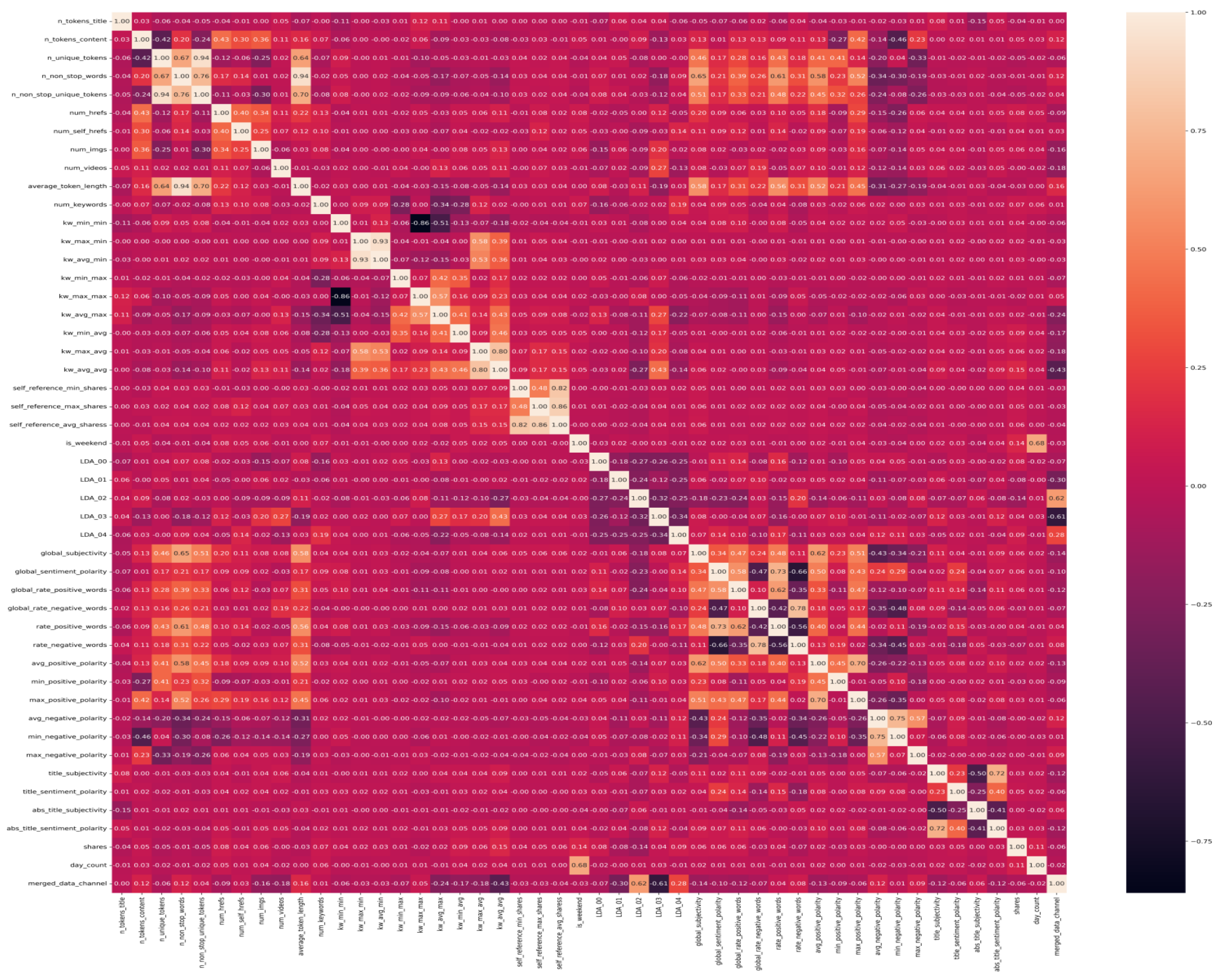


Fig 3.

We also visualized and understood the features involving this dataset. Following are the visuals and the findings:

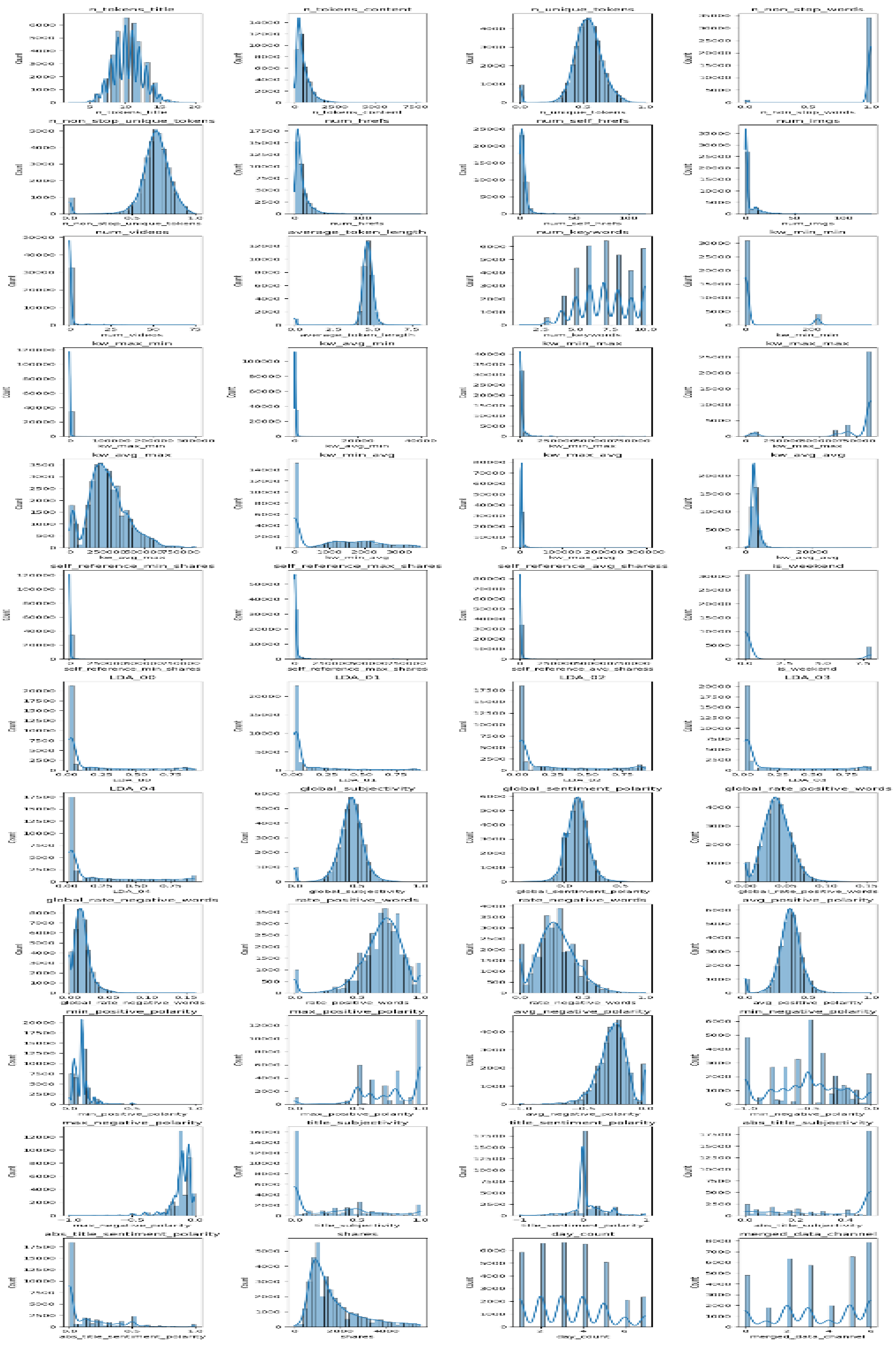

n_tokens_title
n_tokens_content
n_unique_tokens
n_non_stop_words
n_non_stop_unique_tokens
num_hrefs
num_self_hrefs
num_imgs
num_videos
average_token_length
num_keywords
kw_min_min
kw_max_min
kw_avg_min
kw_min_max
kw_max_max
kw_avg_max
kw_min_avg
kw_max_avg
kw_avg_avg
self_reference_min_shares
self_reference_max_shares
self_reference_avg_sharess
is_weekend
LDA_00
LDA_01
LDA_02
LDA_03
LDA_04
global_subjectivity
global_sentiment_polarity
global_rate_positive_words
global_rate_negative_words
rate_positive_words
rate_negative_words
avg_positive_polarity
min_positive_polarity
max_positive_polarity
avg_negative_polarity
min_negative_polarity
max_negative_polarity
title_subjectivity
title_sentiment_polarity
abs_title_subjectivity
abs_title_sentiment_polarity
shares
day_count
merged_data_channel
Count

Fig 4.

- N_tokens_title: The graph ranges from 0 to over 20 tokens per title, with most values clustered around 10 tokens.
- N_tokens_content: The graph ranges from 0 to 7500 tokens per content, with most values skewed towards the lower end, under 1000 tokens.
- N_unique_tokens: The graph ranges from 0 to 1 for the proportion of unique tokens, with most values densely concentrated around 0.3 to 0.6.
- N_non_stop_words: The graph ranges from 0 to 1 for the proportion of non-stop words, with the majority of values concentrated very close to 1.
- N_non_stop_unique_tokens: The graph ranges from 0 to 1 for the proportion of non-stop unique tokens, with most values peaking sharply around 0.5.
- Num_hrefs: The graph ranges from 0 to over 100 hyperlinks (num_hrefs) with most values skewed towards the lower end, particularly concentrated below 25 links.
- Num_self_hrefs: The graph ranges from 0 to over 100 self-references (num_self_hrefs) with a majority of the values concentrated very close to 0.
- Num_imgs: The graph ranges from 0 to over 100 images (num_imgs), with the majority of values skewed towards 0 to 10 images.
- Num_videos: The graph ranges from 0 to over 75 videos (num_videos), with the vast majority of values concentrated at 0.
- Average_token_length: The graph ranges from 0 to about 8 in average token length, with most values sharply peaking around 5.
- Num_keywords: The range of the graph is from 2,000 to 10,000, and most of the values are inclined to the lower end of the graph, with the count of the value 2,000 being the highest.
- kw_min_min: The range of the graph is from 0 to 200, and most of the values are inclined to 0.
- kw_max_min: The range of the graph is from 0 to 120,000, and most of the values are inclined to 10,000.
- kw_avg_min: The range of the graph is from 0 to 40,000, and most of the values are inclined to 0.
- kw_min_max: The range of the graph is from 0 to 250,000, and most of the values are inclined to 250,000.
- Kw_max_max: The range of the graph is from 0 to 250,000, and most of the values are inclined to 250,000.
- kw_avg_max: The range of the graph is 0-3,500, and most of the values are inclined to 2,500.
- kw_min_avg: The range of the graph is 2000 - 10000 and the respective value most of the values are inclined to is 0

- kw_max_avg: The range of the graph is from 0 to 300,000, most of the values are inclined to 10,000.
- Kw_avg_avg: The range of the graph is from 0 to 20,000, and most of the values are inclined to 0.
- self_reference_min_shares: The range of the graph in the image is from 0 to 50000, and most of the values are inclined to 0.
- self_reference_max_shares: The range of the graph is from 0 to 120,000, and the most common value (mode) is 10,000.
- Self_reference_avg_sharess: The range of the variable in the graph is from 0 to 750000, and the mode of this observation (most frequent value) is 10,000.
- Is_weekend: The range of the variable in the graph is from 0 to 7.5, and the mode of this observation is 0.
- LDA_00: The range of the variable in the graph is 0-1, and the mode of this observation (most frequent value) is 20,000 with 0.00.
- LDA_01: The range of the variable 'LDA_01' in the graph appears to be from 0 to approximately 0.3, and the mode of the observation—the value that occurs most frequently—is at 0.
- LDA_02: The range of the variable 'LDA_02' in the graph is from 0 to approximately 1, and the mode of the observation is at 0.
- LDA_03: The range of the variable 'LDA_03' in the graph is from 0 to approximately 1, and the mode of the observation is at 0.
- LDA_04: The range of the variable 'LDA_04' in the graph is from 0 to just above 0.75, and most of the values are inclined towards 0.
- global_subjectivity: The range of the variable 'global_subjectivity' in the graph is from 0 to 1, and most of the values are inclined towards a peak at approximately 0.45.
- global_sentiment_polarity: The range of the variable 'global_sentiment_polarity' in the graph is from 0 to 0.5, and most of the values are inclined towards a peak at approximately 0.1.
- Global_rate_positive_words: The range of the variable 'global_rate_positive_words' in the graph is from 0 to approximately 0.16, and most of the values are inclined towards a peak at about 0.03.
- global_rate_negative_words: The range of the variable 'global_rate_negative_words' in the graph is from 0 to approximately 0.1, and most of the values are inclined towards a peak at about 0.02.
- rate_positive_words: The range of the variable 'rate_positive_words' in the graph is from 0 to 1, and most of the values are inclined towards a peak at approximately 0.05.
- rate_negative_words: The range of the variable 'rate_negative_words' in the graph is from 0 to 1, and most of the values are inclined towards a peak at approximately 0.2.
- Avg_positive_polarity: The range of the variable 'avg_positive_polarity' in the graph is from 0 to 1, and most of the values are inclined towards a peak at approximately 0.35.

- min_positive_polarity: The range of the variable 'min_positive_polarity' in the graph is from 0 to 1, and most of the values are inclined towards a peak at approximately 0.05.
- max_positive_polarity: The range of the variable 'max_positive_polarity' in the graph is from 0 to 1, and most of the values are inclined towards the maximum value of 1.
- avg_negative_polarity: The range of the variable 'avg_negative_polarity' in the graph is from -1 to 0, and most of the values are inclined towards a peak at approximately -0.3.
- min_negative_polarity: The range of the variable 'min_negative_polarity' in the graph is from -1 to 0, and most of the values are inclined towards a peak at approximately -0.5.
- max_negative_polarity: The range of the variable 'max_negative_polarity' in the graph is from -1 to 0, and most of the values are inclined towards a peak at approximately -0.05.
- title_subjectivity: The range of the variable 'title_subjectivity' in the graph is from 0 to 1, and most of the values are inclined towards a peak at 0.
- title_sentiment_polarity: The range of the variable 'title_sentiment_polarity' in the graph is from -1 to 1, and most of the values are inclined towards a peak at 0.
- abs_title_subjectivity: The range of the variable 'abs_title_subjectivity' in the graph is from 0 to 0.5, and most of the values are inclined towards a peak at 0.5.
- abs_title_sentiment_polarity: The range of the variable 'abs_title_sentiment_polarity' in the graph is from 0 to 1, and most of the values are inclined towards a peak at 0.
- shares: The range of the variable 'shares' in the graph is from 0 to above 5000, and most of the values are inclined towards a peak at approximately 0 to 100.
- Day_count:The range of the variable 'day_count' in the graph is from 0 to 7, and the distribution shows peaks at values corresponding to whole numbers from 1 to 7, suggesting these represent the days of the week.
- Merged_data_channel: The range of the variable 'merged_data_channel' in the graph is from 0 to above 6, and the distribution shows significant peaks at values that correspond to the categories represented by whole numbers, likely indicating different channels or categories within the merged data set.

3. Model Selection

To construct a robust predictive model, we employed various regression techniques, each tailored to address specific challenges inherent in the dataset.

To ascertain the reliability and generalizability of our models, we executed k-fold cross-validation. This technique is pivotal in gauging the model's performance across diverse data subsets, thereby reducing the risk of overfitting to any specific dataset.

3.1 Linear Models Approach

3.1.1 Linear Regression Model:

- Standard linear regression model. We used it as a baseline approach.
- Lagging behind the regularized models, Linear Regression reveals higher AIC (469,969.8) and BIC (470,406.679) values. The associated MSE of 1,086,202 suggests that its simplicity may come at the expense of predictive accuracy, especially in the face of complex relationships.

3.1.2 Lasso Regression Model:

- The Lasso Regression performs L1 regularization to introduce sparsity and reduce overfitting. It was chosen because there were many correlated variables.
- It displayed the lowest AIC and BIC values (0.48 and 6,429.312), coupled with a commendable MSE of 1,086,199. Its ability to introduce sparsity and select pivotal features positions it as a strong contender.

3.1.3 Ridge Regression Model:

- Ridge Regression performs L2 regularization to reduce overfitting. It handles correlated variables better than standard linear regression.
- While Ridge Regression shows slightly elevated AIC and BIC values (16.29 and 11,424.290), its reasonable MSE of 1,086,401 highlights its efficacy in managing multicollinearity, rendering it suitable for specific scenarios.

3.1.4. Elastic Net Regression Model:

- The Elastic Net Regression is a hybrid of both Lasso and Ridge regression. It provides both variable selection from Lasso and grouped selection from Ridge to handle correlated variables well.
- Exhibiting competitive AIC and BIC values (0.89 and 6,453.663), Elastic Net Regression effectively balances the strengths of Lasso and Ridge regression. The closely aligned MSE of 1,086,225 underscores its proficiency in handling correlated variables.

3.1.5. Poisson Regression Model:

- Generalized linear model using a Poisson distribution. It is well-suited for count data by assuming a Poisson distribution. It models the relationship between dependent and independent variables in the context of discrete counts. This makes it an appropriate choice when working with datasets. Also, the dependent variable follows a Poisson distribution.

- Noteworthy for its significantly elevated AIC and BIC values (15.8 million), the Poisson Regression model raises concerns about potential misfit or issues with the Poisson assumption. The comparatively lower MSE of 1,123,880 suggests reasonable predictive accuracy but prompts scrutiny regarding its overall suitability for the dataset.

3.1.6 Building the Full and Baseline Models:

- We developed a comprehensive linear model (full.model) using all predictors to gauge their combined influence on the article shares. This holistic approach was intended to capture the multifaceted nature of the factors affecting article popularity.
- The baseline model (start.model) was fundamental in our analysis. It served as a minimalist model containing only the intercept, allowing us to measure the incremental value added by each predictor in the full model.

3.1.7 Stepwise Model Selection:

- Our strategy here involved refining the model to its most effective form. We employed the stepAIC function for an iterative process of adding and removing variables based on their AIC values. This method helped us strike a balance between model complexity and explanatory power.
- We utilized forward, backward, and both-direction stepwise approaches to ensure a comprehensive exploration of possible models. This multi-pronged approach was crucial for identifying the most parsimonious model.

3.1.8 BIC-based Model Selection:

- To complement our AIC-based selection, we also employed the BIC criterion, which tends to favor simpler models by penalizing complexity more heavily. This provided a useful counterbalance to the AIC approach, ensuring our model wasn't overfitting.

3.1.9 Model Summaries and AIC Comparison:

- We meticulously analyzed the summaries of each model iteration. This involved scrutinizing the significance of individual predictors, the overall model fit, and the variance explained by the model.

3.2 Extension to Other Models

3.2.1 Poisson and Negative Binomial Regression:

- Our investigation expanded to include Poisson and Negative Binomial regressions, acknowledging the count nature of our dependent variable. We were particularly cautious about the potential overdispersion in the Poisson model, which led us to consider the Negative Binomial model as a viable alternative.

3.2.2 Quasi-Poisson Model:

- The Quasi-Poisson model was introduced as an intermediary between the Poisson and Negative Binomial models. This model was particularly beneficial in handling overdispersion by relaxing the strict variance-mean relationship assumed in the standard Poisson regression.

3.2.3 Stepwise Selection for GLMs:

- Similar to our linear models, we applied stepwise selection based on AIC for our GLMs, at least the Poisson, and Negative Binomial models.

3.3 Evaluation Metrics

(i) AIC (Akaike Information Criterion): AIC is designed to balance the goodness of fit of a model with its complexity or the number of parameters. It penalizes overly complex models to avoid overfitting.

(ii) BIC (Bayesian Information Criterion): BIC, similar to AIC, penalizes model complexity, but it does so more strongly. BIC puts a higher penalty on additional parameters.

(iii) MSE (Mean Squared Error): The Mean Squared Error (MSE) assesses the performance of regression models by measuring the average of the squared differences between predicted and actual values providing an estimation of the model's training accuracy.

(iv) R-squared: It is a statistical measure that represents the proportion of the variance in the dependent variable explained by the independent variables in a regression model. It ranges from 0 to 1, with higher values indicating a better fit.

(v) Adjusted R-squared modifies the R-squared value by considering the number of independent variables in the model. It penalizes the inclusion of irrelevant variables, providing a more accurate assessment of the model's explanatory power while accounting for complexity.

3.4 Linear Model Results

Model	**AIC**	**BIC**	**R-Squared**	**Adjusted R-Squared**	**MSE**
Linear Regression	469969.8	470406.67	0.111	0.111	1086202
Lasso Regression	0.48	6429.31			1086199
Ridge Regression	16.29	11424.29			1086401
Elastic Net Regression	0.89	6453.66			1086225
Poisson Regression	15834850	15835286.30			1123880
Full	587451.1	578900.1	0.1111	0.1098	1081848
AICBoth	587431.4	587761.4	0.1109	0.1099	1082087
AICForward	587430.2	587751.9	0.1108	0.1099	1082087
AICbackwards	587431.2	587761.4	0.1109	0.1099	1082087
BIC	587481.2	587709.8	0.109	0.1084	1084371
Intercept	5891482.1	591499.1			
GLM Poisson*	19806754	19807195	0.1211908		1090200
Negative Binomial*	571084	571532.3	0.00838994		1135412
NB Intercept	575813	575829.5			

Table 1.

3.5 Tree-Based Ensemble Model

Tree-based ensemble learning models are a type of machine learning algorithm that focuses on building multiple simple individual decision trees to form a "decision forest." These models can be broadly categorized into two main types: one based on random sampling, which independently constructs multiple decision trees by randomly selecting subsets of data and sets of features; and the other based on sequential construction, where each tree is sequentially built to correct the predictive errors of its predecessor. For tree-based ensemble approaches, three commonly used models are random forest, GBM, and XGBoost. Compared to standard decision tree models, these ensemble methods significantly enhance overall prediction accuracy and reduce the risk of overfitting by integrating the predictions of multiple models. The advantage of these ensemble approaches lies in their ability to capture complex structures in the data while maintaining the model's ability to generalize.

3.5.1 Extreme Gradient Boosting

Decision tree based ensemble learning models are a type of machine learning algorithm that focuses on building multiple simple individual decision trees to form a "decision forest." These models can be broadly categorized into two main types: one based on random sampling, which independently constructs multiple decision trees by randomly selecting subsets of data and sets of features; and the other based on sequential construction, where each tree is sequentially built to correct the predictive errors of its predecessor. For tree-based ensemble approaches, three commonly used models are random forest, gradient boosting method (GBM), and Extreme Gradient Boosting (XGBoost). Compared to standard decision tree models, these ensemble methods significantly enhance overall prediction accuracy and reduce the risk of overfitting by integrating the predictions of multiple models. The advantage of these ensemble approaches lies in their ability to capture complex structures in the data while maintaining the model's ability to generalize.

3.5.2 Data and method

The dataset included 35103 samples, and 53 variables with 52 features and 1 label. 52 features were scaled using min-max scaling approach and the label variable was scaled using logarithmic transformation. The Dataset was split to train and test subsets with the ratio 7:3. The train dataset was used to do hyperparameters searching and fit the final model for prediction; the test dataset was used to calculate the RMSEs by comparing the predictions and the true values from the test dataset. Auto-training function from the R package h2o was used to select the models initially based on the complete dataset. Random search approach was used to search for optimal hyperparameters for the selected XGBoost model.

3.5.3 Hypothesis testing

Bootstrapping method was used to conduct hypothesis tests. Two group of specific variables were chosen to do two hypothesis testings. Group 1: data_channel_is_lifestyle, data_channel_is_entertainment,data_channel_is_bus,data_channel_is_socmed, data_channel_is_tech, data_channel_is_world; Group 2: is_weekend, day_count. To verify if the feature(s) were significantly important to the model, the null hypothesis was none of the features in either group was important to the mode; and the alternative hypothesis was at least one of the features was significant. For each bootstrapping iteration, a new dataset was randomly resampled from the original dataset. Then the new dataset was split into train and test subsets. By doing the bootstrapping 1000 times each for the null and alternative hypothesis, two sequences of RMSEs were acquired for one hypothesis test. To compare the means of the two tests, the non-parametric Kruskal Wallis H test was implemented to calculate the significance of the variable groups.

3.5.4 Results

The final XGBoost model is configured with a subsample rate of 0.7, which implies that during each iteration, 70% of the data samples are randomly chosen to grow the trees. This helps in making the model more robust and prevents overfitting. The model consists of 500 estimators, indicating that it integrates 500 trees for making predictions. The stepsize or learning rate to train the final model was 0.05 with a maximum depth of 3 for each tree. Lastly, the 'base_score' was set to 7.423 which was determined by the root mean of the label variable. It is used as the initial prediction score of all instances before the model starts boosting iterations.

We have acquired the 10 most important variables to the model, which were LDA_00, kw_avg_avg, kw_min_avg, num_hrefs, kw_max_avg, kw_max_max, n_unique_tokens, kw_max_min, self_reference_min_shares, LDA_01. After bootstrapping iterations, the average RMSEs were 0.5572 and 0.5538 for group 1, and 0.5605 and 0.5538 for group 2 for the null hypothesis and the alternative hypothesis separatetly. The H statistics calculated by the Kruskal Wallis H test were 11.9198 and 40.9189 for the two tests. Both tests were significant with p-values equal to 0.005 for group 1 and lower than 0.001 for group 2.

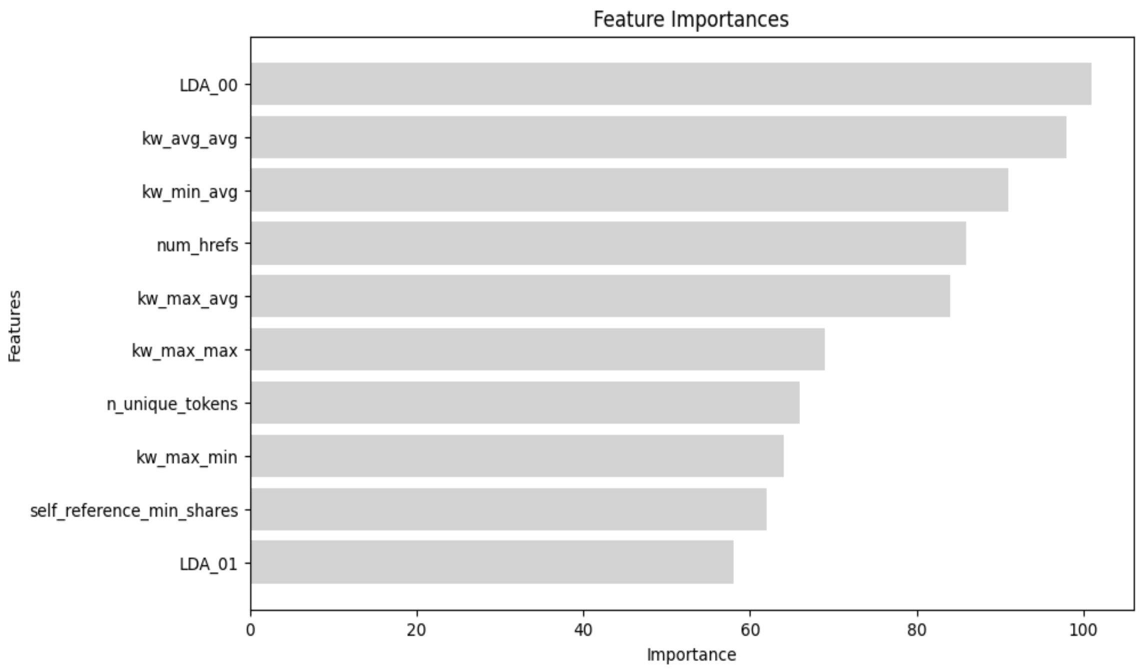


Fig 5. XGBoost model feature importances

Variables	H statistics	p-value
data_channel_is_lifestyle data_channel_is_entertainment data_channel_is_bus data_channel_is_socmed data_channel_is_tech data_channel_is_world	11.9198	0.00005
is_weekend day_count	40.9189	<0.001

Table 2. Hypothesis testings statistics and p-values

3.6 Artificial Neural Network

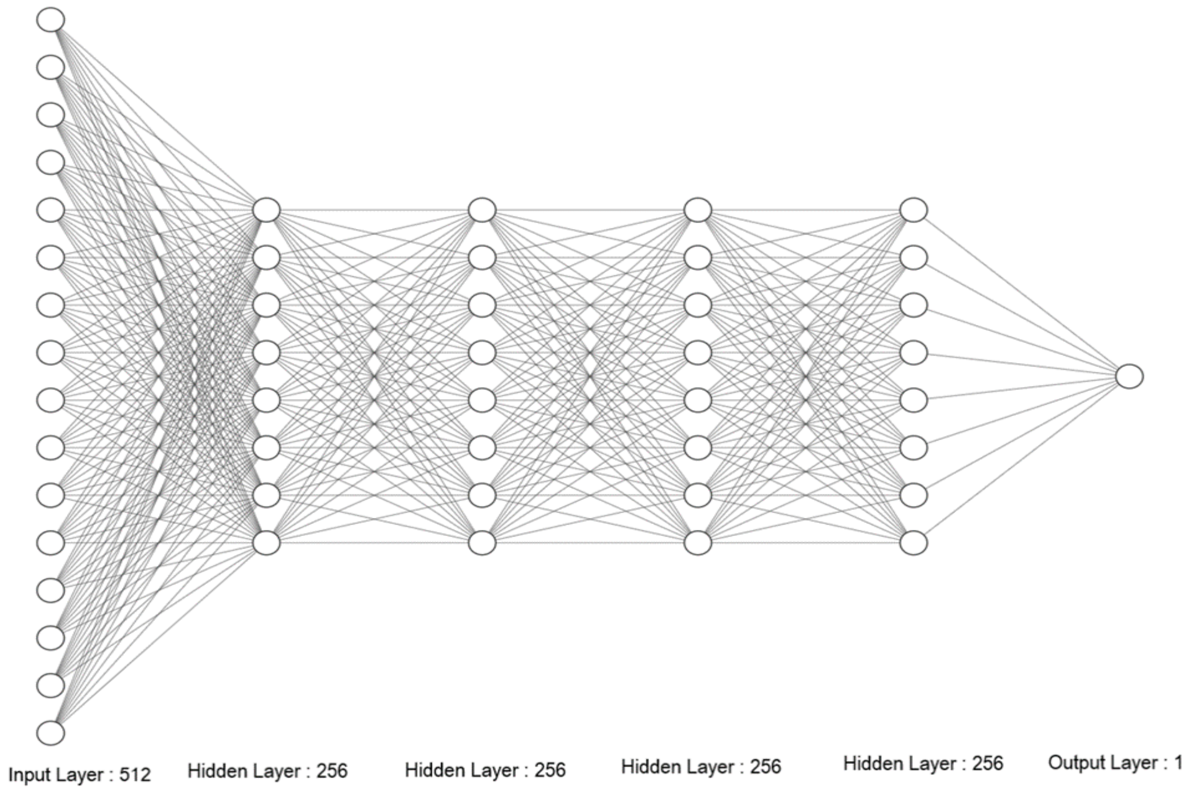
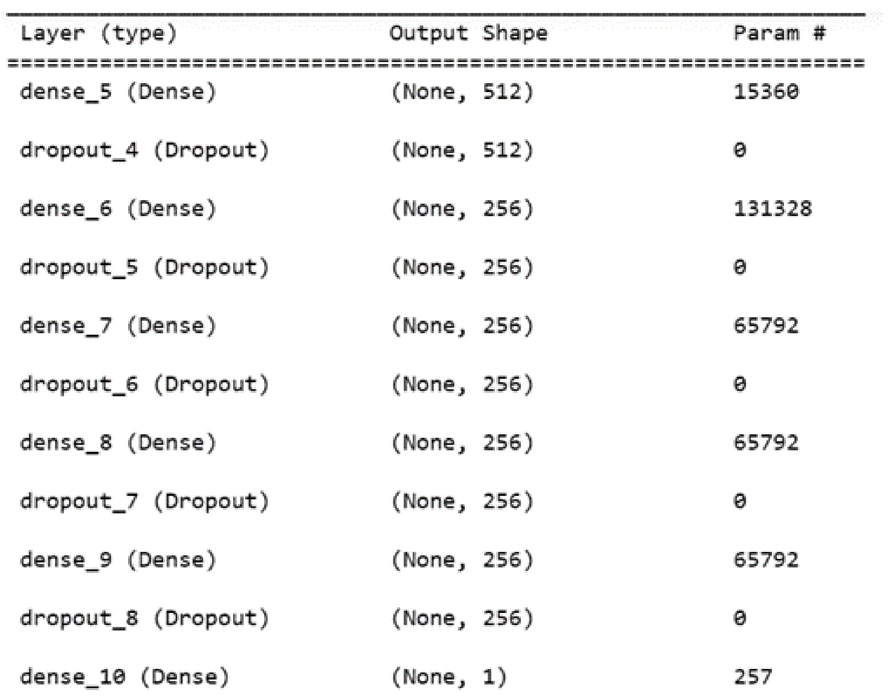


Fig 6. Neural Network Architecture (Scaled down by 2^5)

3.6.1 Architecture:

The next model used is a feed-forward deep neural network model with an input layer with 512 neurons, followed by four hidden layers with 256 neurons each and an output layer. All the layers are dense. All the layers except the output layer have RELU activation function. The output layer has an exponential activation function. Additionally, there are dropout layers between all the layers to avoid overfitting the model. The neural network architecture can be seen in the figure above wherein for visualization purposes, the neurons are scaled down by 2^5 (i.e. for input, 16 neurons are shown in the figure, whereas the actual neurons are 512). The total number of parameters in the model was 344,321.



```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_5 (Dense)             (None, 512)               15360

 dropout_4 (Dropout)         (None, 512)               0

 dense_6 (Dense)             (None, 256)               131328

 dropout_5 (Dropout)         (None, 256)               0

 dense_7 (Dense)             (None, 256)               65792

 dropout_6 (Dropout)         (None, 256)               0

 dense_8 (Dense)             (None, 256)               65792

 dropout_7 (Dropout)         (None, 256)               0

 dense_9 (Dense)             (None, 256)               65792

 dropout_8 (Dropout)         (None, 256)               0

 dense_10 (Dense)            (None, 1)                 257
```

Table 3. Neural Network Model Architecture

3.6.2 Preprocessing:

Before training the model, the correlation matrix of all the features was studied. Based on that, a total of 30 features were selected for training the model. The rest were removed so as to avoid multicollinearity. Only the features with correlation in range (-0.7,0.7) were used to train the model. The next step was to scale the features. To scale the features, Min-Max scaling was used. The data was then divided into training, validation, and testing sets in the ratio 70:20:10.

3.6.3 Training and testing:

The best hyperparameters we found and the model was trained. The neural network training was a feed-forward type as we are carrying out regression-based tasks. The loss function used for the model is the Poisson loss function. Since, our target variable is count type i.e. its distribution is Poisson, our output layer has an exponential activation function. It is also the reason why Poisson loss is being used here to train the model rather than using MSE or MAE which are typically used for regression-based models. Along with it, MSE was calculated in order to compare the NN model with other models.

The Poisson loss function is derived from the Poisson probability distribution, which models the number of events occurring within a fixed interval of time or space. For the Poisson loss function, we use the negative log-likelihood, which is the negative logarithm of the likelihood function. The Poisson loss is calculated for each observation, and the goal during training is to minimize the average Poisson loss across all observations. The mean of all the losses is displayed as Poisson loss during each epoch. Another metric calculated during the training was MAPE (Mean average percentage error).

During training, the model was trained for 200 epochs and an early stopping function based on the validation loss was applied in order to prevent the model from overfitting. The batch size was 32 and there were dropout layers added between the layers of the model to avoid overfitting.

3.6.4 Results:

The Poisson loss was found to be -7.08 on the training set, -7.069 on the validation set, and -6.79 on the test data. The MSE was found to be 0.3200 and for the validation set, it was found to be 0.3226 for the validation data and 0.3578 for the test data. The MAPE was found to be 6.1727 on the training data, 6.2328 on the validation data, and 7.3249 on the test data.

Data	**Poisson Loss**	**MSE**	**MAPE**
Training set	-7.080	0.3200	6.1727
Validation set	-7.069	0.3226	6.2328
Testing set	-6.790	0.3578	7.3249

Table 4.

3.6.5 Hypothesis Testing:

For models such as neural networks, to test hypotheses, we use bootstrap hypothesis testing. Bootstrap hypothesis testing is a resampling technique used to estimate the distribution of a statistic by resampling from the observed data with replacement. This method can be applied to various statistical analyses, including hypothesis testing. In the context of neural networks, bootstrap hypothesis testing can be used to assess the uncertainty and variability in model performance metrics.

The reason we use bootstrapping for hypothesis testing in models like neural networks is that it is a non-parametric approach. Bootstrapping is non-parametric, making minimal assumptions about the underlying distribution. This is advantageous in scenarios where distributional assumptions are unclear. In the case of neural networks, the model is more like a black box model. Thus, using bootstrap hypothesis testing is preferred here.

To conduct bootstrap hypothesis testing, we resampled the data with a replacement 1000 times. We then eliminated certain features from the dataset based on the hypothesis. Then trained a neural network model for each resampled data using the same architecture of the original model and noted the validation MSE for each model. So, in total for each hypothesis, 1000 different neural network models were trained. Based on the law of large numbers, we assume that the distribution of the MSEs obtained is normal. We then calculated the p-value based on the MSEs obtained and the MSE of the original model. We conducted the hypothesis testing at a significance level of 0.05 which is a standard practice.

The following hypotheses were tested:

Hypothesis test 1:

Null Hypothesis (H0): The features related to the category of the article (tech, business, social media, lifestyle, entertainment, and world) are not significant.

Alternative Hypothesis (H1): The features having information about the category of the article (tech, business, social media, lifestyle, entertainment, and world) are significant.

Result: We reject the null hypothesis(H0) as the p-value obtained from the test (0.0432) is less than the significance level of 0.05 (alpha). Thus, The features having information about the category of the article (tech, business, social media, lifestyle, entertainment, and world) are significant.

Hypothesis test 2:

Null Hypothesis (H0): The features related to the day the article was posted (Monday-Sunday or Weekend) are not significant.

Alternative Hypothesis (H1): The features related to the day the article was posted (Monday-Sunday or Weekend) are significant.

Result: We reject the null hypothesis(H0) as the p-value obtained from the test (0.00034) is less than the significance level of 0.05(alpha). Thus, The features related to the day the article was posted (Monday-Sunday or Weekend) are significant.

4. Results

4.1 Linear Models

Model	MSE
Linear Regression	1086202
Lasso Regression	1086199
Ridge Regression	1086401
Elastic Net Regression	1086225
Poisson Regression	1123880
Full	1081848
AICBoth	1082087
AICForward	1082087
AICbackwards	1082087
BIC	1084371
GLM Poisson*	1090200
Negative Binomial*	1135412

Table 5.

Ultimately, based on R-squared, AIC, and MSE, we decided the AIC stepwise both ways model was the best linear model. Surprisingly, despite being count data, the Poisson GLM and the Negative Binomial models were still worse. It had Multiple Very Statistically Significant variables, but combined with the low R-squareds, AICs, and MSEs, we decided on trying more complex models for a better fit.

4.2 Non - Linear Models

Model	**MSE**
XGBoost	0.5527
Neural Network	0.3578

Table 6.

In the case of non- linear models, the model which performed best in terms of MSE is the neural network model. We can see that the non-linear models perform better as we have seen that the data is non-linear. The XGBoost model performs good as well with a decent MSE of 0.5527.

5. Conclusion

The features relating to the category of the article are significant, borderline. Judgment call level. The features relating to the day of the week an article is posted is significant, not borderline. Very likely. The linear models did not perform well as there was no significant correlation between features and the target variable. The model which considered non-linearity performed better. The XGBoost gave an MSE of 0.5527 and the Neural Network model gave an MSE of 0.3578 which is significantly less than the linear models, which gave the MSE in millions. Since, the data was non linear, the target variable was count type i.e the data distribution was poisson, we conclude that the models which take non- linearity in account work better on this dataset.

6. References

1. Fernandes, K., Vinagre, P., Cortez, P., & Sernadela, P. (n.d.). *Online news popularity*. UCI Machine Learning Repository. http://archive.ics.uci.edu/dataset/332/online+news+popularity

2. Montesinos-Lopez, O. A., Montesinos-Lopez, J. C., Salazar, E., Barron, J. A., Montesinos-Lopez, A., Buenrostro-Mariscal, R., & Crossa, J. (2021). Application of a Poisson deep neural network model for the prediction of count data in genome-based prediction. *The Plant Genome*, *14*(3), e20118. https://doi.org/10.1002/tpg2.20118

3. Jabeen, H. (2023, April 9). Learn to use poisson regression in R. Dataquest. https://www.dataquest.io/blog/tutorial-poisson-regression-in-r/

4. UCLA. (n.d.). POISSON REGRESSION | R DATA ANALYSIS EXAMPLES. OARC Stats. https://stats.oarc.ucla.edu/r/dae/poisson-regression/

5. H2O.ai (2023). h2o: R Interface for the 'H2O' Scalable Machine Learning Platform. R package version 3.42.0.2. URL: https://CRAN.R-project.org/package=h2o

6. James Andrew Godwin (Apr 2, 2021), Towards Data Science, Ridge, LASSO, and ElasticNet Regression.
URL:https://towardsdatascience.com/ridge-lasso-and-elasticnet-regression-b1f9c00ea3a3

7. Dataset retrieved from https://archive.ics.uci.edu/dataset/332/online+news+popularity