CAPSTON PROJECT

ONLINE NEWS POPULARITY

Batch: PGP DSE July B 2019

Submitted By

Kupa Shetty	-	MXPMC5NOFM
Shishira A V	-	ABYRJKFQYP
Tharshini P R	-	SK09N942HN
Vishal Ghosh	-	W76VC2MALE

Mentor: Mr. Vignesh Kumar

ABSTRACT AND KEYWORDS

ABSTRACT

News articles are an engaging type of online content that captures the attention of a significant amount of Internet users as it gives in short description of the topics. They are particularly enjoyed by mobile users and are massively spread through online social platforms. As a result, there is an increased interest in discovering the articles that will become popular among users. This objective falls under the broad scope of content popularity prediction and has direct implications in the development of new services for online advertisement and content distribution. Our aim to predict the popularity of the articles based on the how many times an article is been shared. There as both positive articles and negative articles which is been analyzed with text mining. Here, we are predicting the how popular is the article when it has image, videos, contents, on which days of the week the share is more and which main topic is getting shared to maximum.

Keywords

Text Mining, Topic Modeling, Sentiment Analysis, Machine Learning: Algorithms, Accuracy

ACKNOWLEDGEMENTS

We would like to express our sincerest gratitude to our Project Mentor Mr. Vignesh Kumar for providing her invaluable guidance, suggestions during the course of the project, his willingness to go out of the way and ensuring we were adequately prepared to meet the challenges this research project posed, without whom our project would have failed to come to fruition.

We're also thankful for the Institute and the resources they provided us and the DSE faculty that helped us understand the concepts of Machine Learning and Data Analytics.

Also, we're thankful for our families for extending the necessary support to complete this program.

CERTIFICATE OF COMPLETION

I hereby certify that the project titled "Online News Popularity" was undertaken and completed under my guidance and supervision by Krupa, Shishira, Tharshini, Vishal students of the July 2019 B batch of the Post Graduate Program in Data Science Engineering, Bangalore

Mr. Vignesh Kumar

Date:

TABLE OF CONTENTS:

INTRODUCTION

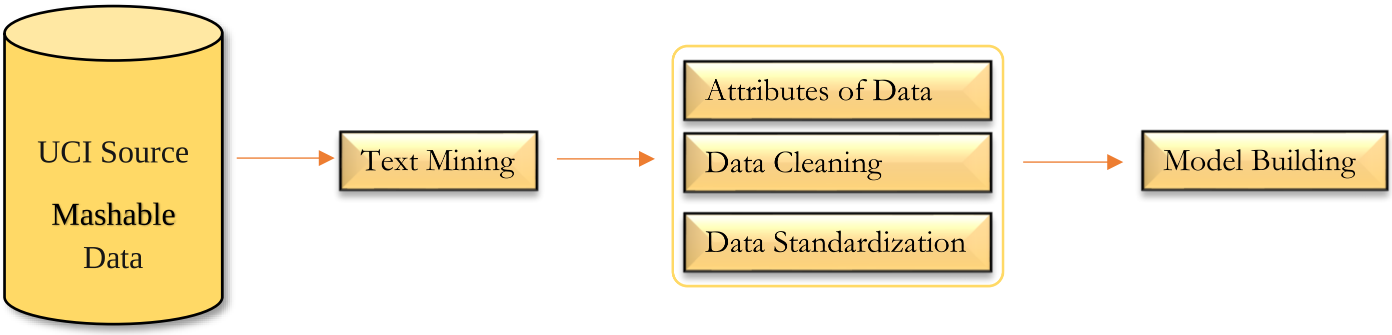
The widespread adoption of smartphone and access to internet social networking sites information on any corner of the world related topics like politics, economy, business, entertainment, technology, fashion has accelerated the competition of online news in the recent times. Mashable is a global, multi-platform media and entertainment company. Powered by its own proprietary technology, Mashable is the go-to source for technology, digital culture and entertainment content for its dedicated and influential audience around the globe.

In the era of reading and sharing information and news has gained the popularity as it has become the part of people’s entertainment lives. Hence, predicting accurately the popularity of news prior to its publication on how news which includes different topics about the world and individuals that can have both positive and negative influence plays a crucial role. A sample of 39000 observation from the articles are considered from Mashable website which is published between the years 2013 and 2015.

Different data channels have its popularity based on content, images, videos and other parameters. Words in the articles plays a significant role in getting popular. There were some extreme values which was driving down the accuracy of the models. Using IQR we removed those extreme values and normalized the data with min max scaler. PCA had no impact on improving the model accuracy. Gradient boost algorithm worked will though there was not much relation between the attributes.

PROJECT OBJECTIVE:

The dataset is from Mashable website which has shared the articles in the year 2015. We are predicting the number of shares of the content that the article will get once it is published. Research and developing methodology for predicting the number of shares and popularity. Since platforms like Mashable publish hundreds of articles in a day and these platforms earn revenue from number of shares of the articles which gains popularity. When does the number of share increase, which type of blog does consumer like? Whenever the number of shares is more those patterns will be analyzed. Depending on these patterns action could be taken on understanding how the article should be and thus understanding the customer. These articles also give a link to other articles and advertisements from which popularity increases and so is the revenue.



PROBLEM STATEMENT:

In this project, “Predicting number of shares that the article will get once it is published. Predicting the popularity of articles based on number of shares.” The problem utilizes list of article features for best machine learning model to accurately predict the number of shares and also predict how popular an article will be after publish.

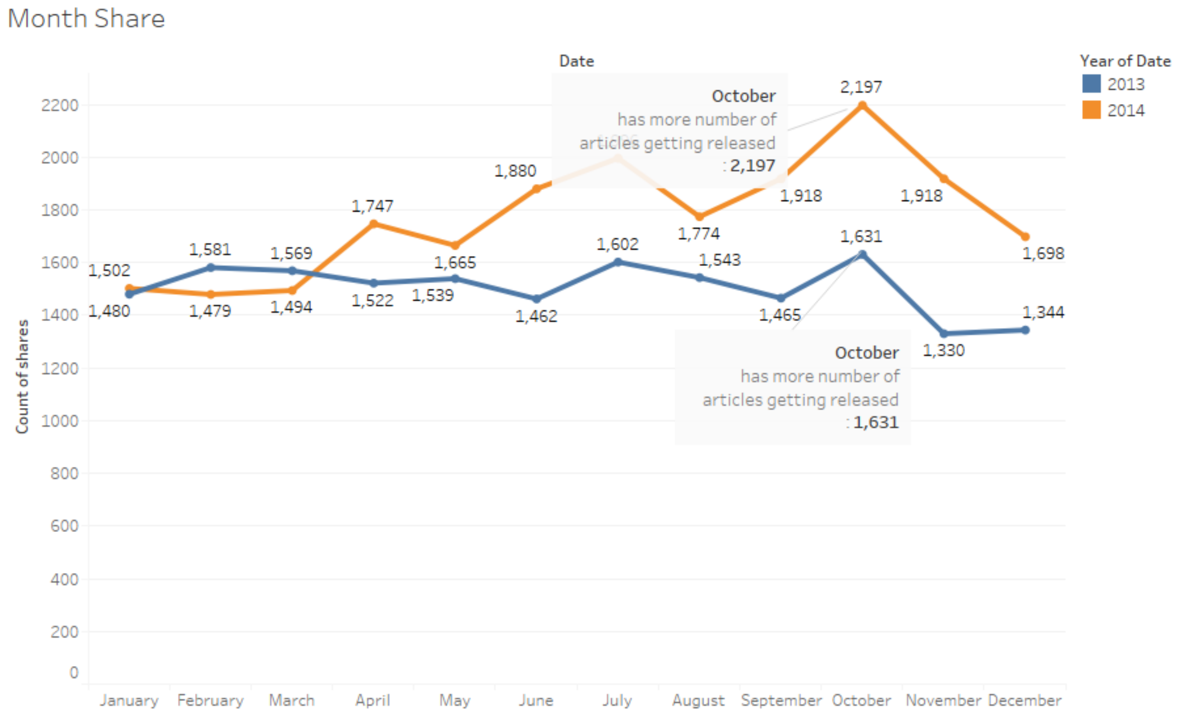
DOMAIN:

Multi-Media

DATA SOURCE:

https://archive.ics.uci.edu/ml/datasets/Online+News+Popularity

NEED FOR STUDY:



We can infer that 2014 has more articles getting publishing when compared with the 2013. Hence, we can say that more impactful news which is getting popularity as the years prolong. So, it is important to study how an article gains it popularity and attract the readers.

DATA CONSTRAINTS AND DATA INFORMATION:

The articles were published by Mashable (www.mashable.com) and their content as the rights to reproduce it belongs to them. Hence, this dataset gives some statistics associated with it. The original content be publicly accessed and retrieved using the provided URLs.

Feature Name	**Feature Description**	**Type of Data**
URL	URL of the article	Non-predictive
timedelta	Days between the article publication and the dataset acquisition	Non-predictive
Shares	Number of shares	Numerical

Tokenization: “Tokens” are usually individual words and “tokenization” is taking a text or set of text and breaking it up into its individual meaningful words i.e., converting sentences to words. Special characters and apostrophes are considered as separate words.

Feature Name	Feature Description	Type of Data
N_tokens_title	Number of words in the title	Discrete
n_tokens_content	Number of words in the content	Numerical
average_token_length	Average length of the words in the content. It is the sum of length of each word in the content divided by total number of words in the content.	Numerical

Bag of Words (BOW): In text processing, words of the text represent discrete, categorical features. The mapping from textual data to real valued vectors is called feature extraction. One of the simplest techniques to numerically represent text is Bag of Words. We make the list of unique words in the text corpus called vocabulary. Then we can represent each sentence or document as a vector with each word. Another representation can be counting the number of times each word appears in a document. The most popular approach is using the **TF-IDF** technique.

***Note**: Documents are rows which is collection of terms. Terms are the columns which is collection of words*

Term Frequency-Inverse Document Frequency (TF-IDF) technique is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. It is often used as a weighting factor in searches of information retrieval. The TF-IDF value increases proportionally to the number of times a word appears in the document and is offset by the number of documents in the corpus that contain the word, which helps to adjust for the fact that some words appear more frequently in general.

- **Term Frequency (TF)** = (Number of times term t appears in a document)/ (Total number of terms in the document)
- **Inverse Document Frequency (IDF)** = log(N/n), where, N is the number of documents and n is the number of documents a term t has appeared in. The IDF of a rare word is high, whereas the IDF of a frequent word is likely to be low. Thus, having the effect of highlighting words that are distinct. Inverse document frequency factor is incorporated which diminishes the weight of terms that occur very frequently in the document set and increases the weight of terms that occur rarely.

Feature Name	Feature Description	Type of Data
N_unique_tokens	Rate of unique words in the content	Numerical
n_non_stop_words	Rate of non-stop words in the content	Numerical
n_non_stop_unique_tokens	Rate of unique non-stop words in the content	Numerical

Note: Stop words *— frequent words such as "the, is", etc. that do not have specific semantic. In total there are 179 stop words in English.*

Metadata: They are the brief notes of the data. They are the data that contains information about other data. Metadata of a subject can be an image, year, events, which gives the brief description of the main content of the data. Used for discovery and identification.

Topic Modeling: A Topic Model can be defined as an unsupervised technique to discover topics across various text documents. It is a process to automatically identify topics present in a text object and to derive hidden patterns exhibited by a text corpus. These topics are abstract in nature, i.e., words which are related to each other form a topic. It represents each and every term and document as a vector.

Note: *Topics are cluster of words (similar and related words), documents are cluster of topics, topic is a frequency of words*

Keywords: These are words which helps to connect to the main data. These are used based on most frequently used words for quick search by the users.

Feature Name	Feature Description	Type of Data
Num_keywords	Number of keywords in the metadata	Discrete
kw_min_min	Worst keyword (min. shares)	Numerical
kw_max_min	Worst keywords (max. shares)	Numerical
kw_avg_min	Worst keyword (avg. shares)	Numerical
kw_min_max	Best keyword (min. shares)	Numerical
kw_max_max	Best keyword (max. shares)	Numerical
kw_avg_max	Best keyword (avg. shares)	Numerical
Kw_min_avg	Avg. keyword (min. shares)	Numerical
kw_max_avg	Avg. keyword (max. shares)	Numerical
kw_avg_avg	Avg. keyword (avg. shares)	Numerical

Latent Dirichlet Allocation (LDA): LDA assumes documents are produced from a mixture of topics. Those topics then generate words based on their probability distribution. Given a dataset of documents, LDA backtracks and tries to figure out what topics would create those documents in the first place. LDA is a matrix factorization technique. In vector space, any corpus collection of documents can be represented as a document-term matrix. In our case, the topic model is split into 4 topics. It determines the percentage of probability of each topic being present in the contents. We can observe that percentage probability of each document shares with all the 5 LDA's.

Feature Name	Feature Description	Type of Data
LDA_00	Closeness to LDA topic 0	Numerical
LDA_01	Closeness to LDA topic 1	Numerical
LDA_02	Closeness to LDA topic 2	Numerical
LDA_03	Closeness to LDA topic 3	Numerical
LDA_04	Closeness to LDA topic 4	Numerical

Feature Name	Feature Description	Type of Data
num_hrefs	It gives the count of number of links in the news link	Numerical
num_self_hrefs	It gives the number of links to other articles published by Mashable	Numerical
num_imgs	It is the number of images related to the article	Numerical
num_videos	It is the number of videos related to the article	Numerical
self_reference_min_shares	Min. shares of referenced articles in Mashable	Numerical
self_reference_max_shares	Max. shares of referenced articles in Mashable	Numerical
self_reference_avg_shares	Avg. shares of referenced articles in Mashable	Numerical
weekday_is_Monday	Was the article published on a Monday?	Categorical
weekday_is_Tuesday	Was the article published on a Tuesday?	Categorical

weekday_is_Wednesday	Was the article published on a Wednesday?	Categorical
weekday_is_Thursday	Was the article published on a Thursday?	Categorical
weekday_is_Friday	Was the article published on a Friday?	Categorical
weekday_is_Saturday	Was the article published on a Saturday?	Categorical
weekday_is_Sunday	Was the article published on a Sunday?	Categorical
is_weekday	Was the article published on the weekday?	Categorical
is_weekend	Was the article published on the weekend?	Categorical

Data Channel: It is the column categorizes the data based on such as lifestyle, entertainment, business, social media, technology and world (economics, politics etc.)

Feature Name	**Feature Description**	**Type of Data**
data_channel_is_lifestyle	Is data channel ‘Lifestyle’?	Categorical
data_channel_is_entertainment	Is data channel ‘Entertainment’?	Categorical
data_channel_is_bus	Is data channel ‘Business’?	Categorical
data_channel_is_social media	Is data channel ‘Social Media’?	Categorical
data_channel_is_tech	Is data channel ‘Tech’?	Categorical
data_channel_is_world	Is data channel ‘World’?	Categorical

Subjectivity: Language can contain expressions that are objective or subjective. Objective expressions are facts. **Subjective** expressions are opinions that describe people’s feelings towards a specific subject or topic.

Ex: *This is a phone is good. (is subjective as it expresses an opinion towards the taste of the phone.)*

Feature Name	**Feature Description**	**Type of Data**
global_Subjectivity	Text subjectivity	Numerical
global_rate_positive_words	Rate of positive words in the content	Numerical
global_rate_negative_words	Rate of negative words in the content	Numerical
rate_positive_words	Rate of positive words among non-neutral tokens	Numerical

rate_negative_words	Rate of negative words among non-neutral tokens	Numerical
title_subjectivity	Title subjectivity	Numerical
abs_title_subjectivity	Absolute subjectivity level	Numerical

Sentiment Analysis: To determine, from a text corpus, whether the sentiment towards any topic or product etc. is positive, negative, or neutral. Sentiment analysis has compound score considers only positive and negative polarity score and it ignores neutral score.

Polarity: to identifying sentiment orientation (positive, neutral, and negative) in written or spoken language. It ranges between -1 and +1. They are emotions which is expressed on tangible items.

Positive Score = $\frac{No\ of\ times\ occurrence\ of\ positive\ word \times Sentiment\ Values\ of\ Positive\ Word}{Overall\ words}$

Negative Score = $\frac{No\ of\ times\ occurrence\ of\ negative\ word \times Sentiment\ Values\ of\ negative\ Word}{Overall\ words}$

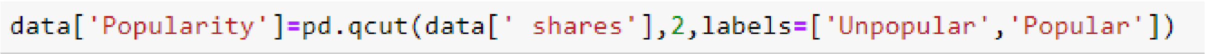
Sentence Polarity = Positive score – Negative score

Compound Score: $\frac{Sentence\_Polarity}{\sqrt{(Sentence\_Polarity^2)+\alpha}}$

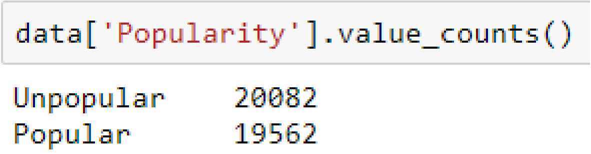
Feature Name	Feature Description	Type of Data
global_sentiment_polarity	Text sentiment polarity	Numerical
avg_positive_polarity	Avg. polarity of positive words	Numerical
min_positive_polarity	Min. polarity of positive words	Numerical
max_positive_polarity	Max. polarity of positive words	Numerical
avg_negative_polarity	Avg. polarity of negative words	Numerical
min_negative_polarity	Min. polarity of negative words	Numerical
max_negative_polarity	Max. polarity of negative words	Numerical
title_sentiment_polarity	Title polarity	Numerical
abs_title_sentiment_polarity	Absolute polarity level	Numerical

CREATING TARGET COLUMN:

There are 39644 observations/articles which are collected from Mashable website. There are 61 attributes and no null values. Total number of shares are predicted by the company based in the content, impact of days and data channels. Using ‘qcut’ from pandas library we have categorized number of shares into ‘Popular’ and ‘Unpopular’.



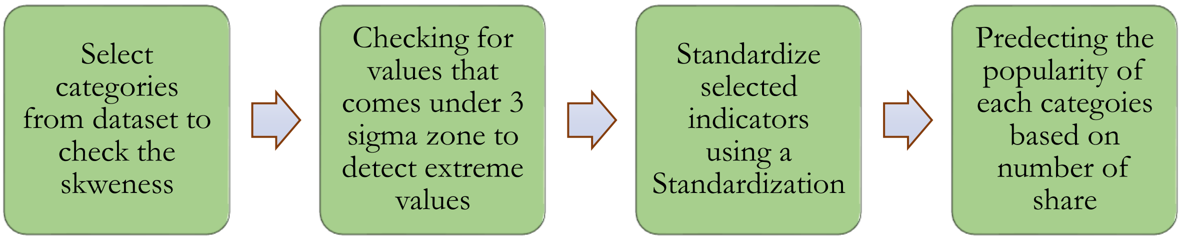
```
data['Popularity']=pd.qcut(data[' shares'],2,labels=['Unpopular','Popular'])
```



```
data['Popularity'].value_counts()
```

```
Unpopular    20082
Popular      19562
```

DATA PREPROCESSING:

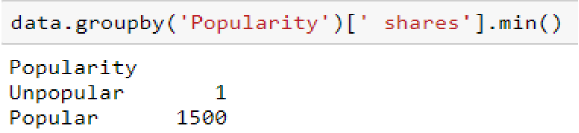


Select categories from dataset to check the skweness

Checking for values that comes under 3 sigma zone to detect extreme values

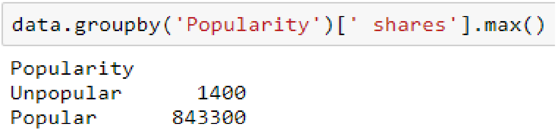
Standardize selected indicators using a Standardization

Predecting the popularity of each categoies based on number of share



```
data.groupby('Popularity')[' shares'].min()
```

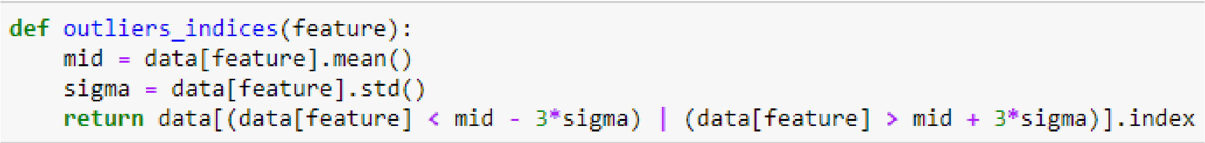
```
Popularity
Unpopular       1
Popular      1500
```



```
data.groupby('Popularity')[' shares'].max()
```

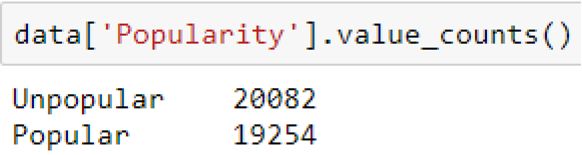
```
Popularity
Unpopular      1400
Popular      843300
```

As we can observe that under ‘Popular’ category there are come extreme values that will right skew that data. Hence, to get good accuracy score and to reduce the error we have to treat those extreme values to get better result. For detecting the extreme values, we used 3 sigma IQR.



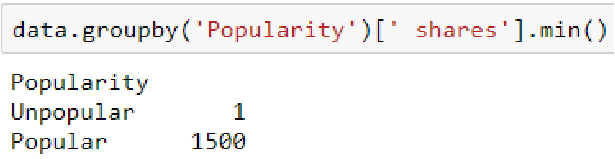
```
def outliers_indices(feature):
    mid = data[feature].mean()
    sigma = data[feature].std()
    return data[(data[feature] < mid - 3*sigma) | (data[feature] > mid + 3*sigma)].index
```

3 sigma zone will remove the outliers whose values is not in between ±3. Since number of shares has outliers, it has been treated using IQR method. The number of observations reduced from 39644 to 39336. As a result, the extreme values got eliminated.



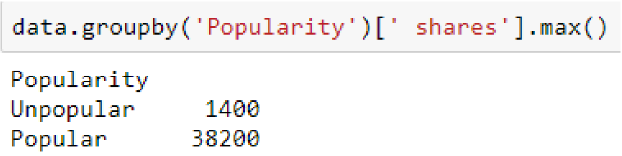
```
data['Popularity'].value_counts()
```

```
Unpopular    20082
Popular      19254
```



```
data.groupby('Popularity')[' shares'].min()
```

```
Popularity
Unpopular       1
Popular      1500
```

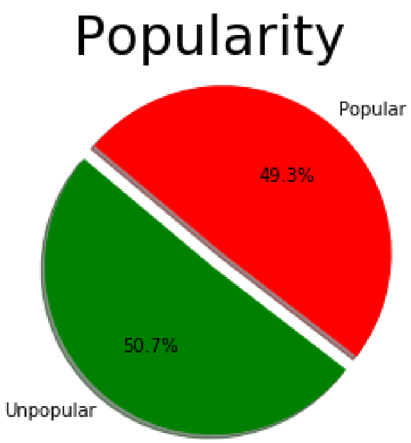


```
data.groupby('Popularity')[' shares'].max()
```

```
Popularity
Unpopular     1400
Popular      38200
```

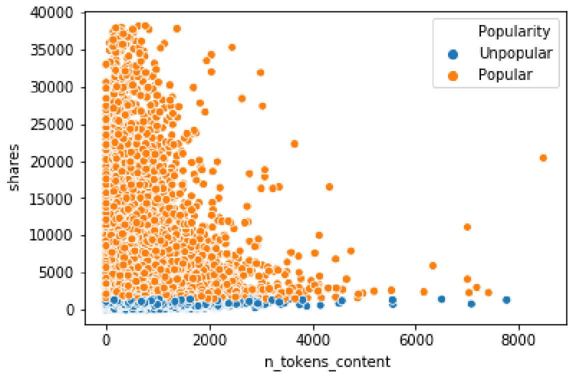
EXPLORATORY DATA ANALYSIS:

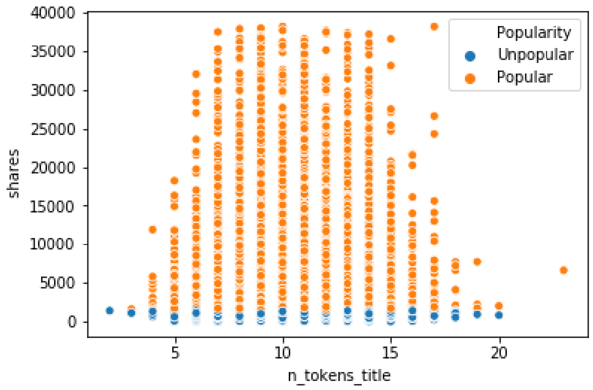
EDA is process in which visualization is done on each valuable to check how they are inter-related and what is the inference we get from the plot which will helpful to bring insights that will help to drive the business. In our data set, more the number of shares more is the revenue. Hence, it is important to study all the attributes which corresponds or which are related to dependent variable. Since majority of the data is derived from text mining it is important to study how each derived attribute from text mining corresponds to increase in number of shares and popularity.

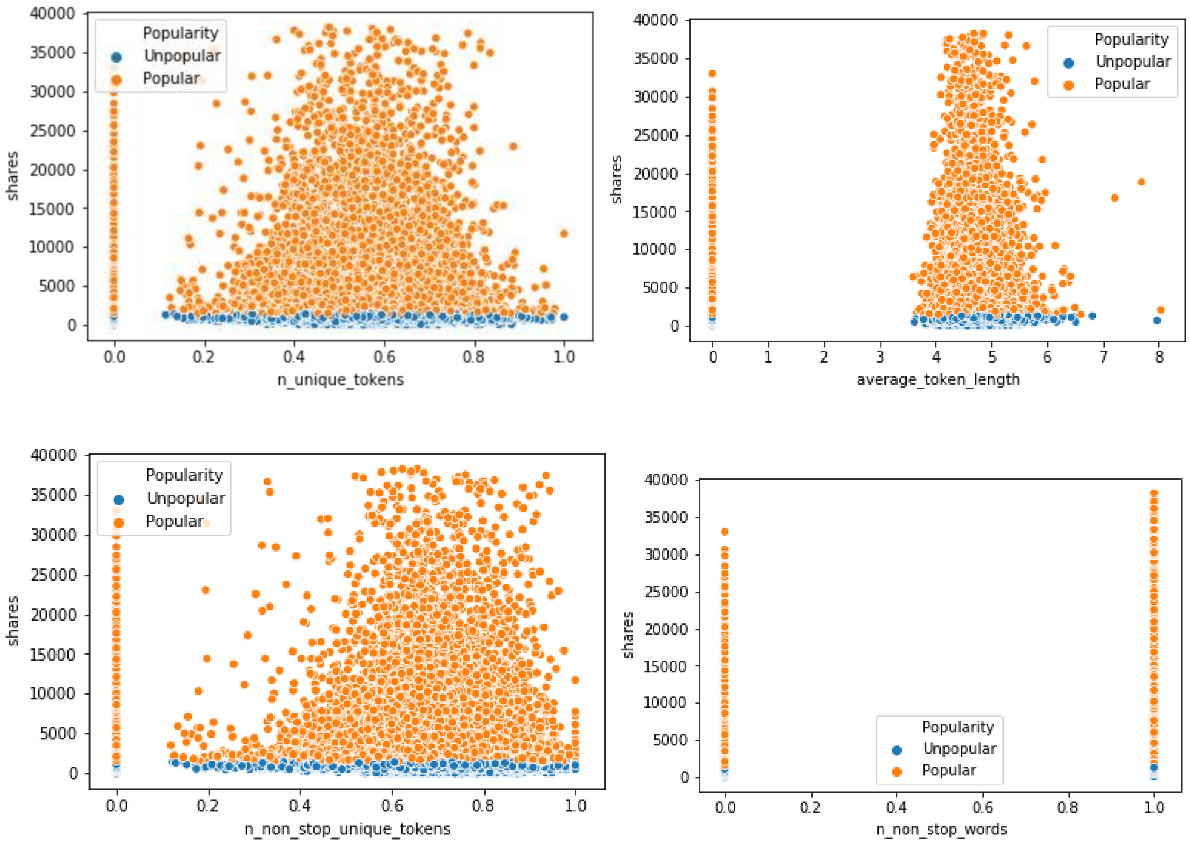


We can observe that of the total percentage of shares 50.7% of shares are unpopular and 49.3% are popular.

DATA CLEANING FOR TEXT MINING:

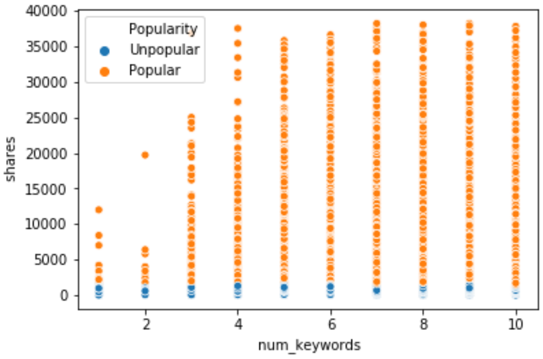


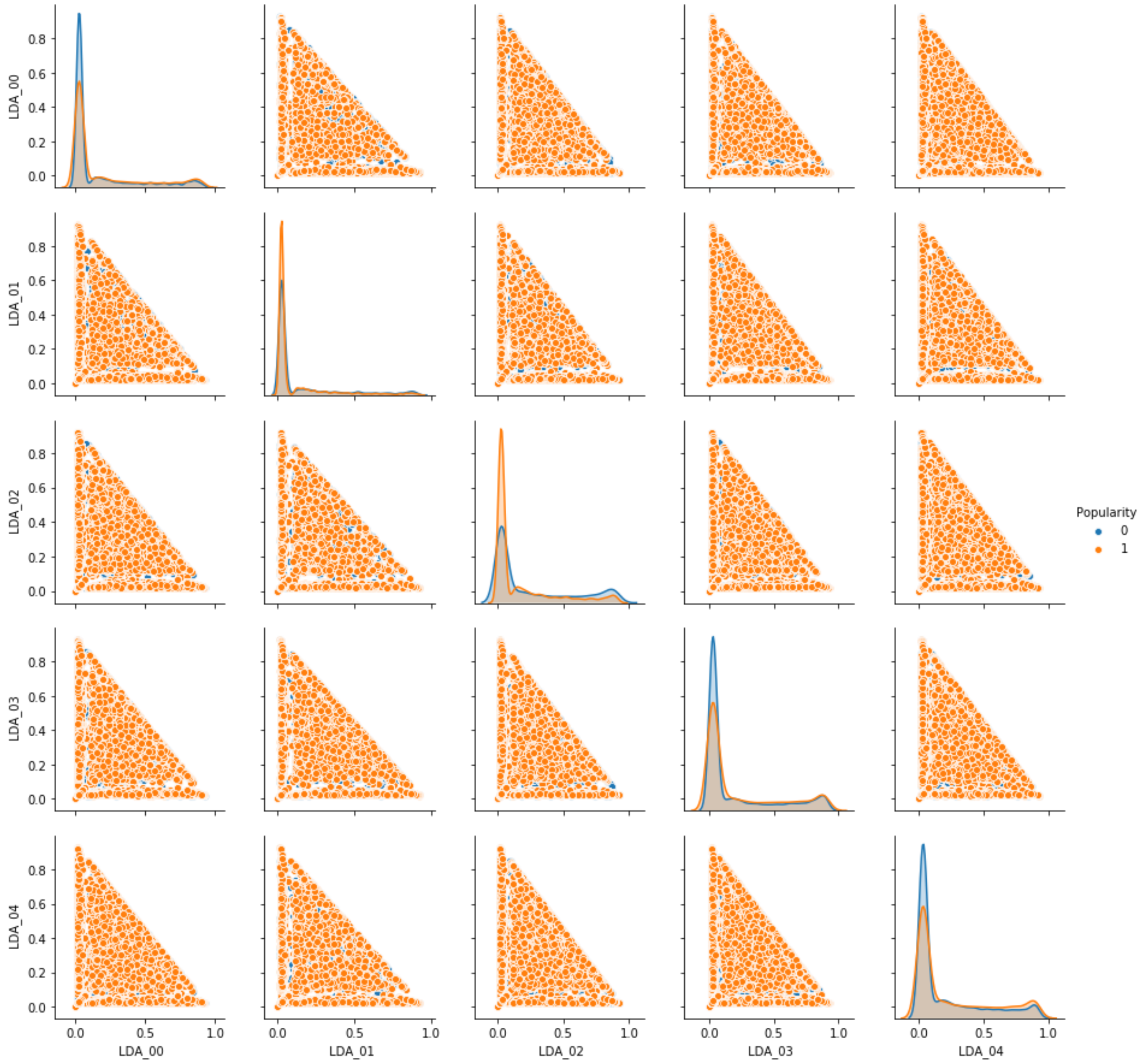




To make a blog powerful we need have some commanding words which will appeal the readers. Maximum if such words in an article will make them more intensive. We can observe from the above scatter plots that the rate of unique token in content and title much be more and, in our case, it should be in the range of average token length should be between 4 to 7. When we take non-stop words the rate percentage should be more to make an article popular and, in our case, it ranges between 50% to 90%.

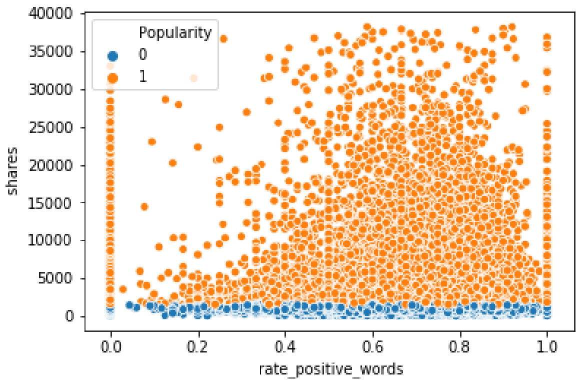
TOPIC MODELING USING LDA:

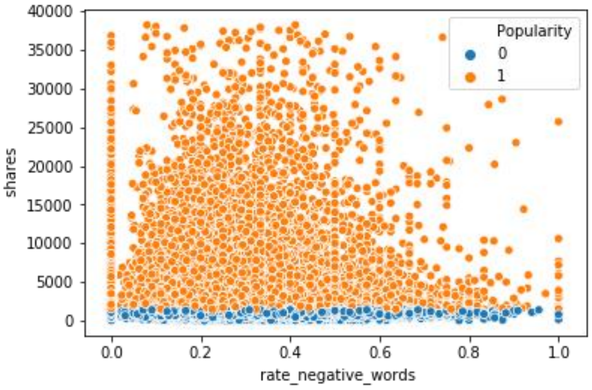




We can observe from the share vs number of keywords plot that more the number of key words more an article will be shares. In our case the topic keywords range between 4 to 10 which has the maximum number of shares. LDA gives the percentage of probability of each topic of an article being present when n-gram is given as 5.

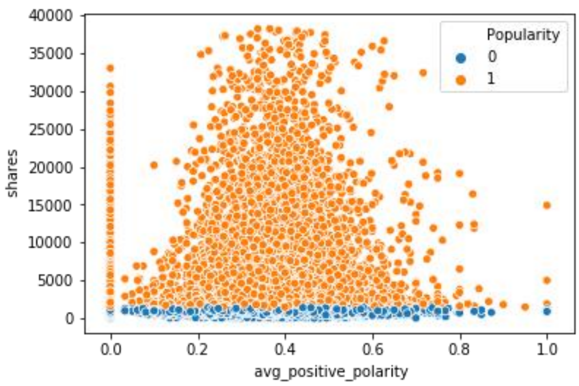
SUBJECTIVITY:

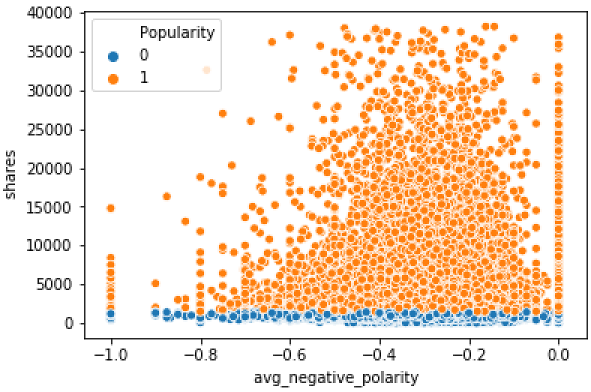




As the subjectivity expressions are opinions that describe people's feelings towards a specific subject or topic. It represents individual sentences to determine whether a sentence expresses an opinion or not. Hence, we can see that the rate positive words range between 50% to 90% which gives a positive feeling on the articles. On the contrary is the rate of negative words which range between 10% to 50% which gives the negative feelings.

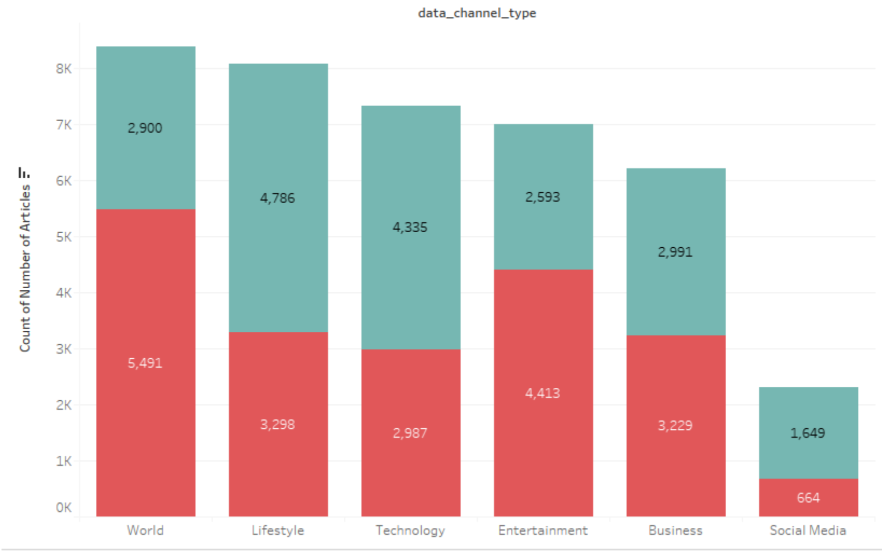
SENTIMENT POLARITY:



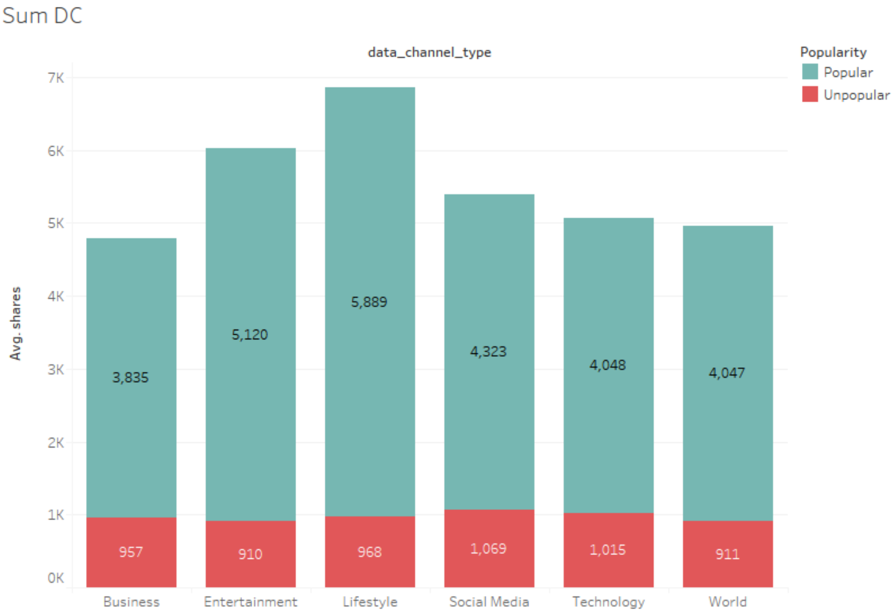


Polaity stresses on emotions on the tangile items. They are the generak sentiment of of what article is about. It is expressed in precentage. In our case, we can obsserve that for positive polarity the range should be between 0.3 to 0.5 and negative sentiment polarity should be between -0.2 to to -0.5. Hence we can conclude by saying an article which has only postive or negative polarity will have minimum number of shares. Hence, we should have bith postive and negative polarity. This cannot always hold good.

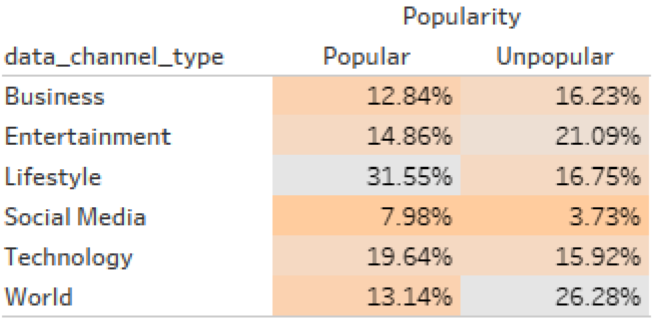
DATA CHANNEL:



This is the graph which represents types of data channels vs number of articles in each data types. World has highest number of articles getting published. But maximum count can bee that they are not popular. Lifestyle had a greater number of popular articles. Social media has minimal number of articles. Hence, we can say that articles under lifestyle, world and technology has more articles

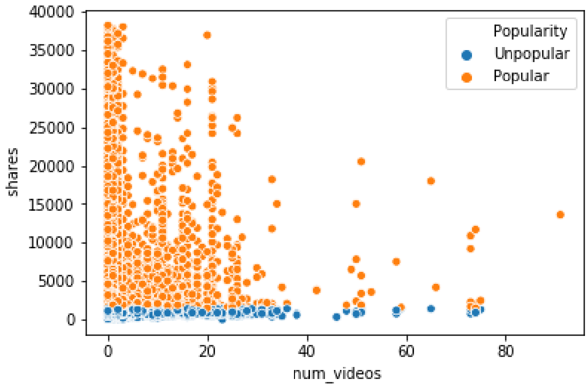


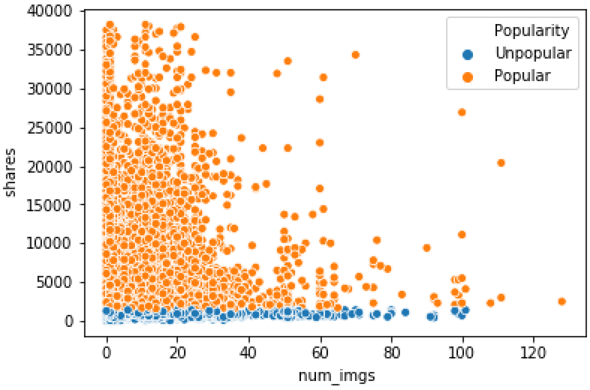
The graph represents types of data channels vs average shares of the articles in each data channel. We can see that maximum average share is with articles related to lifestyle though world has a greater number of articles getting published. Second highest shares of articles are on entertainment. Least is on business.



data_channel_type	Popularity Popular	 Unpopular
Business	12.84%	16.23%
Entertainment	14.86%	21.09%
Lifestyle	31.55%	16.75%
Social Media	7.98%	3.73%
Technology	19.64%	15.92%
World	13.14%	26.28%

IMAGES AND VIDEOS:



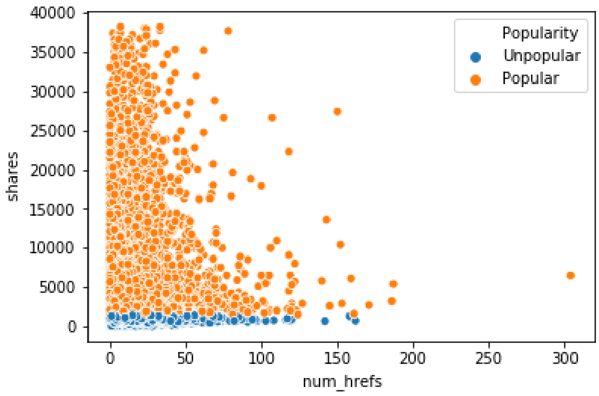


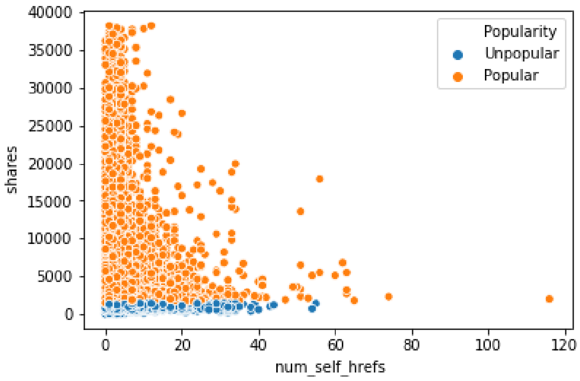
- Image count range between 1-25
- Lesser the image sharing is high

Video count range between 1-4

Lesser the videos sharing is high

NUMBER OF LINKS IN ARTICLES AND LINKS TO OTHER ARTICLES:



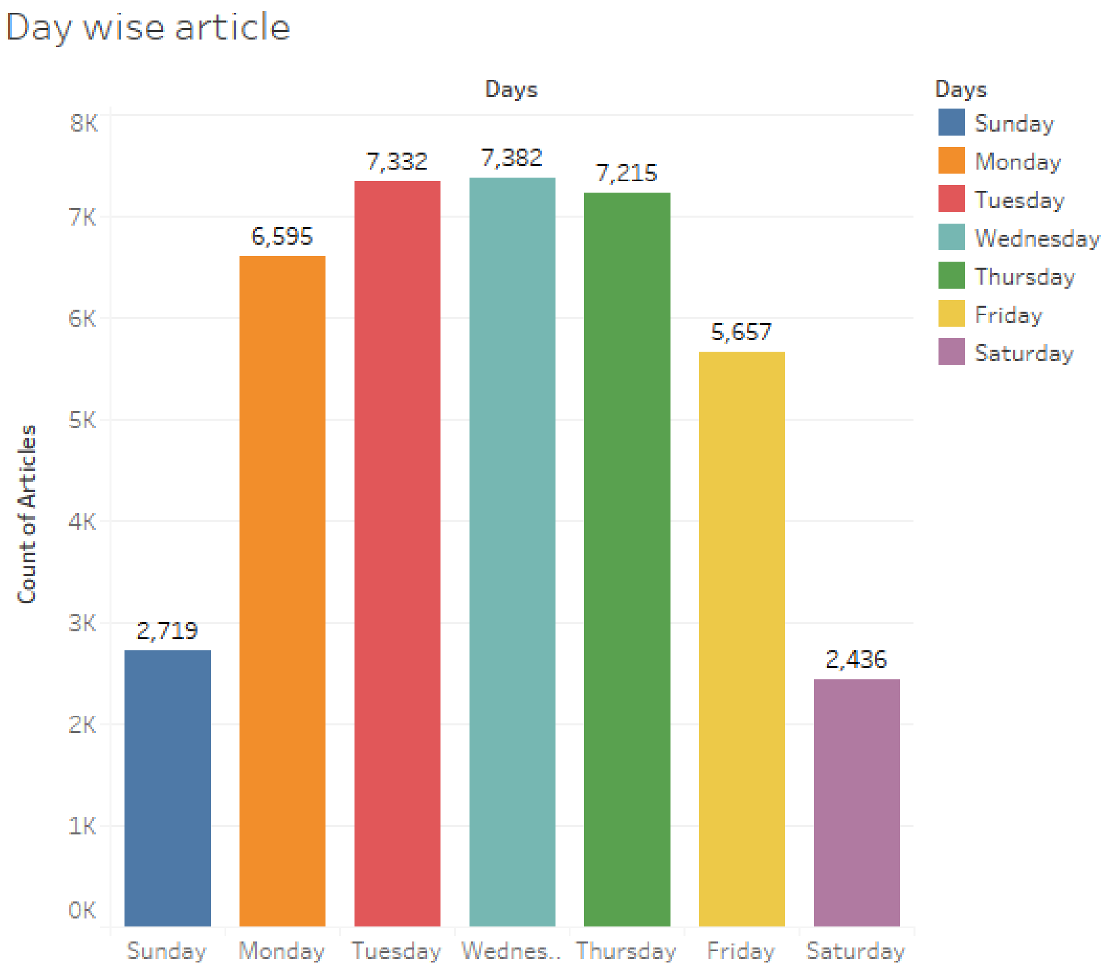


- Link count range between 0-40
- Lesser the links sharing is high

Link for other articles count range between 1-4

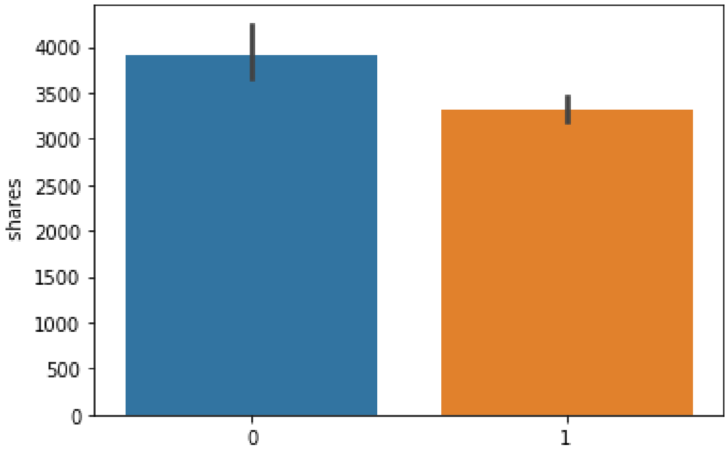
From this we can infer that more the number of links in an article reader will tend to read those articles which will affect on the original article as the reader may then to ignore or forget to read the articles.

DAY WISE SHARE:



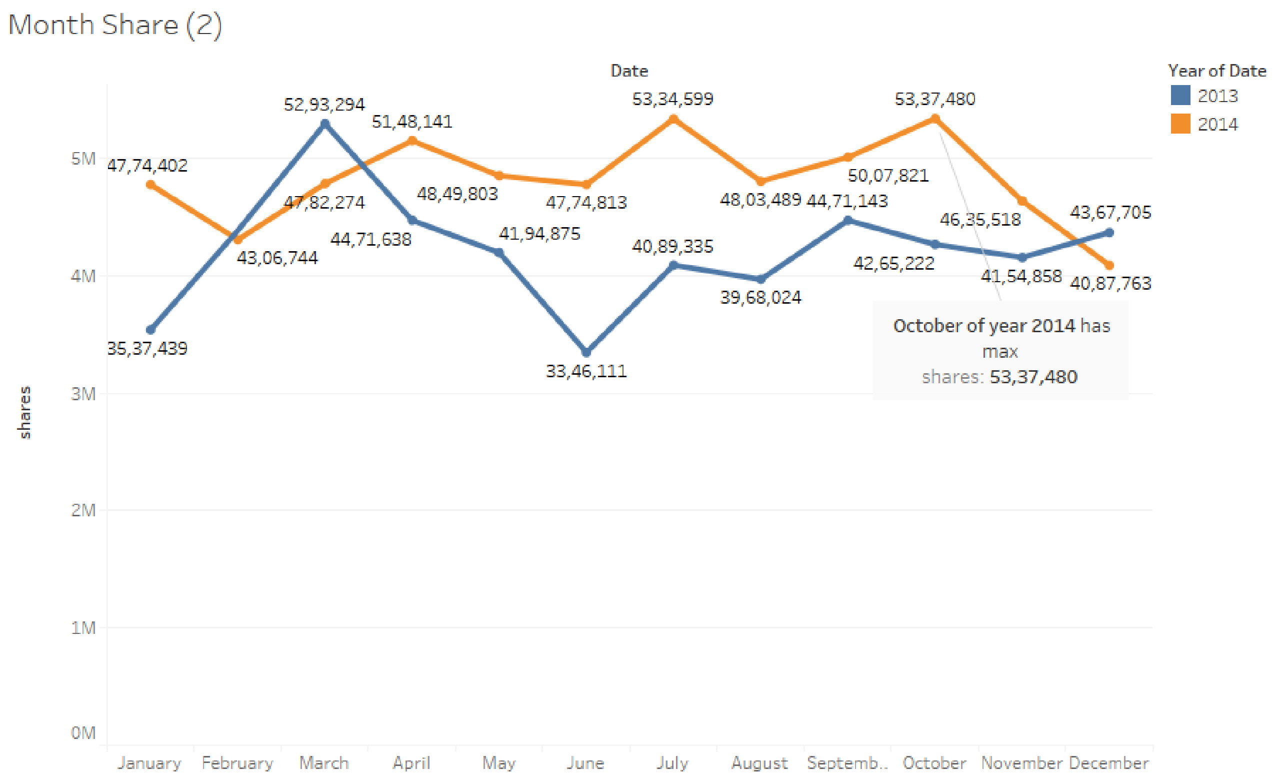
Weekday (0): More share

Weekend (1): Less Share



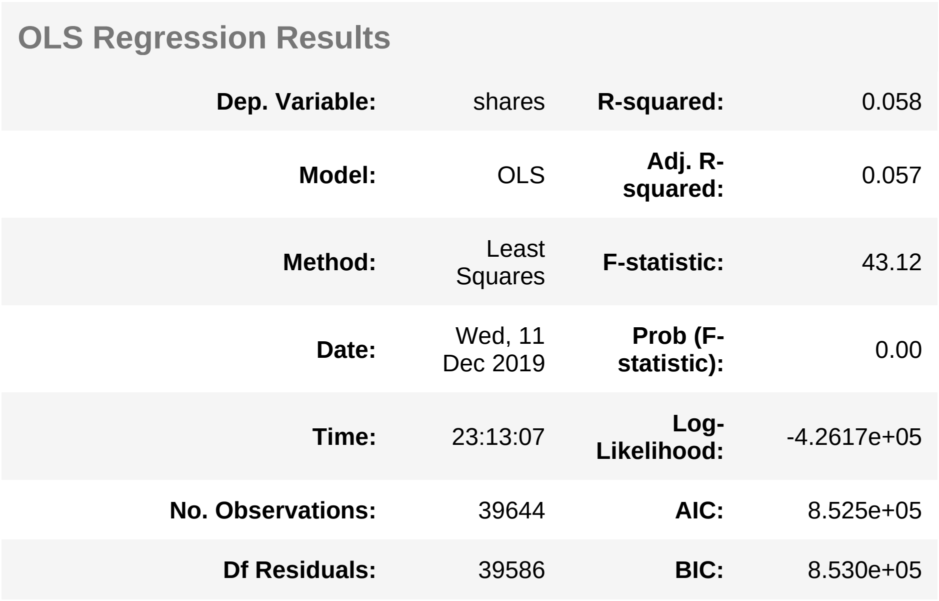
The number of articles getting published is more on Wednesdays and Tuesday. Less number of articles are published on Saturday and Sunday. Hence, we can say that the number of shares of each article is more on weekdays when compared to weekend.

MONTH WISE SHARES:



As the year prolongs, number of articles in each data channels are getting increased with the advent of technology. We can observe that, in the greater number of articles are getting released in the month of October least in December in the year 2014.

FEATURE SELECTION:



OLS Regression Results

Dep. Variable:	shares	**R-squared:**	0.058
Model:	OLS	**Adj. R-squared:**	0.057
Method:	Least Squares	**F-statistic:**	43.12
Date:	Wed, 11 Dec 2019	**Prob (F-statistic):**	0.00
Time:	23:13:07	**Log-Likelihood:**	-4.2617e+05
No. Observations:	39644	**AIC:**	8.525e+05
Df Residuals:	39586	**BIC:**	8.530e+05

Df Model:	57
Covariance Type:	nonrobust

	coef	std err	t	P>|t|	[0.025	0.975]
const	-8.506e+05	5.12e+06	-0.166	0.868	-1.09e+07	9.18e+06
n_tokens_title	88.2324	28.148	3.135	0.002	33.061	143.403
n_tokens_content	0.4597	0.219	2.095	0.036	0.030	0.890
n_unique_tokens	3973.8776	1883.545	2.110	0.035	282.084	7665.671
n_non_stop_words	-1123.8373	5803.035	-0.194	0.846	-1.25e+04	1.03e+04
n_non_stop_unique_tokens	-943.4342	1599.824	-0.590	0.555	-4079.128	2192.260
num_hrefs	18.5773	6.587	2.820	0.005	5.667	31.488
num_self_hrefs	-34.3386	17.503	-1.962	0.050	-68.645	-0.032
num_imgs	9.9232	8.779	1.130	0.258	-7.284	27.130
num_videos	3.8519	15.463	0.249	0.803	-26.456	34.160
average_token_length	-465.2246	238.444	-1.951	0.051	-932.581	2.132
data_channel_is_lifestyle	-866.8313	387.477	-2.237	0.025	-1626.296	-107.367
data_channel_is_entertainment	-779.3576	250.753	-3.108	0.002	-1270.840	-287.875
data_channel_is_bus	-461.3687	375.834	-1.228	0.220	-1198.013	275.276
data_channel_is_socmed	-1321.4819	366.055	-3.610	0.000	-2038.959	-604.005

data_channel_is_tech	-992.7337	364.826	-2.721	0.007	-1707.802	-277.665
data_channel_is_world	-398.9130	369.500	-1.080	0.280	-1123.142	325.316
num_keywords	4.5243	36.477	0.124	0.901	-66.972	76.021
kw_min_min	0.4199	1.594	0.263	0.792	-2.705	3.545
kw_max_min	0.0448	0.049	0.910	0.363	-0.052	0.141
kw_avg_min	-0.0612	0.302	-0.202	0.840	-0.654	0.531
kw_min_max	-0.0014	0.001	-1.207	0.227	-0.004	0.001
kw_max_max	-0.0002	0.001	-0.283	0.777	-0.001	0.001
kw_avg_max	-0.0002	0.001	-0.198	0.843	-0.002	0.001
kw_min_avg	-0.2937	0.074	-3.952	0.000	-0.439	-0.148
kw_max_avg	-0.1133	0.025	-4.541	0.000	-0.162	-0.064
kw_avg_avg	0.9789	0.142	6.878	0.000	0.700	1.258
self_reference_min_shares	0.0250	0.007	3.387	0.001	0.011	0.040
self_reference_max_shares	0.0054	0.004	1.341	0.180	-0.002	0.013
self_reference_avg_sharess	-0.0078	0.010	-0.763	0.445	-0.028	0.012
weekday_is_monday	-1.497e+05	9.03e+05	-0.166	0.868	-1.92e+06	1.62e+06
weekday_is_tuesday	-1.501e+05	9.03e+05	-0.166	0.868	-1.92e+06	1.62e+06
weekday_is_wednesday	-1.499e+05	9.03e+05	-0.166	0.868	-1.92e+06	1.62e+06

weekday_is_thursday	-1.502e+05	9.03e+05	-0.166	0.868	-1.92e+06	1.62e+06
weekday_is_friday	-1.503e+05	9.03e+05	-0.167	0.868	-1.92e+06	1.62e+06
weekday_is_saturday	-5.009e+04	3.01e+05	-0.166	0.868	-6.4e+05	5.4e+05
weekday_is_sunday	-5.028e+04	3.01e+05	-0.167	0.867	-6.4e+05	5.39e+05
is_weekend	-1.004e+05	6.02e+05	-0.167	0.868	-1.28e+06	1.08e+06
LDA_00	9.989e+05	6.02e+06	0.166	0.868	-1.08e+07	1.28e+07
LDA_01	9.991e+05	6.02e+06	0.166	0.868	-1.08e+07	1.28e+07
LDA_02	9.989e+05	6.02e+06	0.166	0.868	-1.08e+07	1.28e+07
LDA_03	9.995e+05	6.02e+06	0.166	0.868	-1.08e+07	1.28e+07
LDA_04	9.991e+05	6.02e+06	0.166	0.868	-1.08e+07	1.28e+07
global_subjectivity	1426.4018	835.471	1.707	0.088	-211.142	3063.946
global_rate_positive_words	-1.026e+04	7035.003	-1.458	0.145	-2.4e+04	3531.478
global_rate_negative_words	-1321.3326	1.34e+04	-0.098	0.922	-2.76e+04	2.5e+04
rate_positive_words	1023.2121	5671.050	0.180	0.857	-1.01e+04	1.21e+04
rate_negative_words	1369.5059	5715.911	0.240	0.811	-9833.817	1.26e+04
global_sentiment_polarity	686.7758	1637.146	0.419	0.675	-2522.070	3895.621

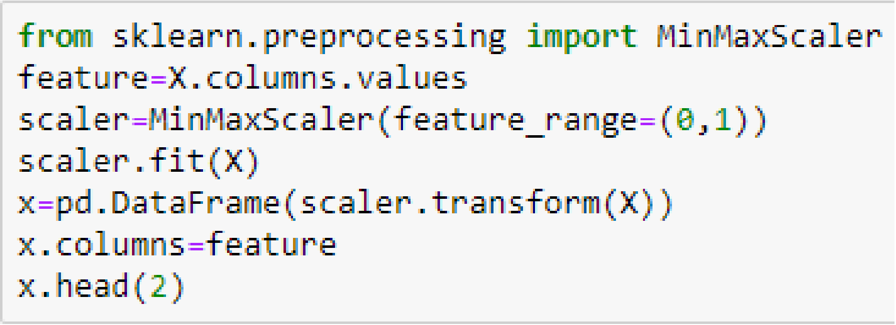
avg_positive_polarity	-1256.7904	1341.715	-0.937	0.349	-3886.584	1373.003
min_positive_polarity	-1533.6508	1123.394	-1.365	0.172	-3735.530	668.228
max_positive_polarity	318.3881	423.194	0.752	0.452	-511.082	1147.858
avg_negative_polarity	-1601.4158	1235.626	-1.296	0.195	-4023.272	820.440
min_negative_polarity	86.1288	450.543	0.191	0.848	-796.947	969.204
max_negative_polarity	-254.7444	1027.482	-0.248	0.804	-2268.633	1759.144
title_subjectivity	-269.1278	269.204	-1.000	0.317	-796.775	258.519
title_sentiment_polarity	-14.1856	245.944	-0.058	0.954	-496.241	467.870
abs_title_subjectivity	338.6001	357.508	0.947	0.344	-362.125	1039.325
abs_title_sentiment_polarity	650.4900	388.529	1.674	0.094	-111.035	1412.015
Popularity	4665.0486	120.992	38.557	0.000	4427.902	4902.195

From OLS summary we can observe that most of the variable are not significant except [' n_tokens_title', 'n_tokens_content', 'n_unique_tokens', 'n_non_stop_words', 'num_hrefs', '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_avg', 'kw_max_avg', 'kw_avg_avg', 'self_reference_min_shares', 'weekday_is_friday', 'is_weekend', 'LDA_03', 'avg_negative_polarity', 'Popularity'].

Since most of the attributes are derived from text mining all the attributes are interlinked. Hence, while building the model it is important to consider all the attributes.

MODEL BUILDING:

In this project, feature selection techniques are applied to improve the classification performance and/or scalability of the system. Thus, we aim to investigate if better or similar classification performance can be achieved with a smaller number of features. We have considered all the attributes except ‘shares’ as we have derived binary class ‘Popular’ and ‘Unpopular’ from number of shares. Without removing the outlier base model logistic regression gave accuracy of 36%. Hence, we treated outlier using IQR and scaling the accuracy of the algorithm started increasing.



```
from sklearn.preprocessing import MinMaxScaler
feature=X.columns.values
scaler=MinMaxScaler(feature_range=(0,1))
scaler.fit(X)
x=pd.DataFrame(scaler.transform(X))
x.columns=feature
x.head(2)
```

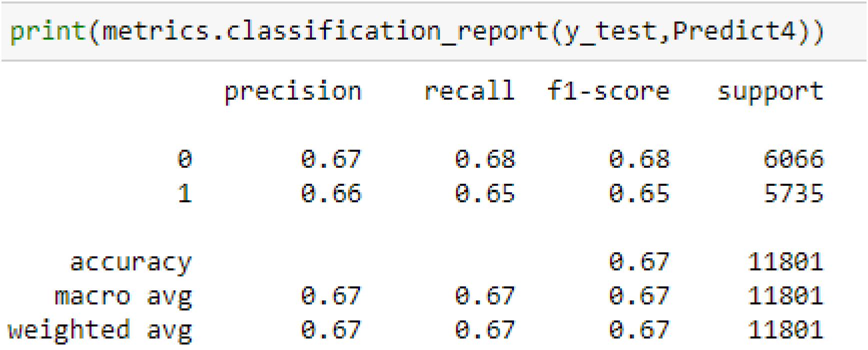
We will use the following model performance measures to check the model accuracy.

Accuracy: Accuracy is the number of correct predictions made by the model by the total number of records. The higher the accuracy the better the model

Sensitivity or recall: Sensitivity (Recall or True positive rate) is calculated as the number of correct positive predictions divided by the total number of positives. It is also called recall or true positive rate (TPR). For a model sensitivity or recall should be more.

Specificity: Specificity (true negative rate) is calculated as the number of correct negative predictions divided by the total number of negatives.

Precision: Precision (Positive predictive value) is calculated as the number of correct positive predictions divided by the total number of positive predictions. In the above case how many correct popular models were predicted as popular.

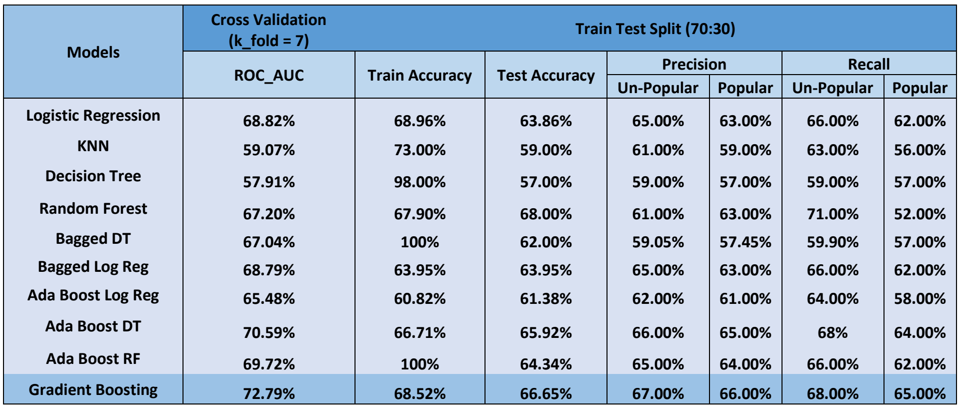


```
print(metrics.classification_report(y_test,Predict4))

              precision    recall  f1-score   support

           0       0.67      0.68      0.68      6066
           1       0.66      0.65      0.65      5735

    accuracy                           0.67     11801
   macro avg       0.67      0.67      0.67     11801
weighted avg       0.67      0.67      0.67     11801
```

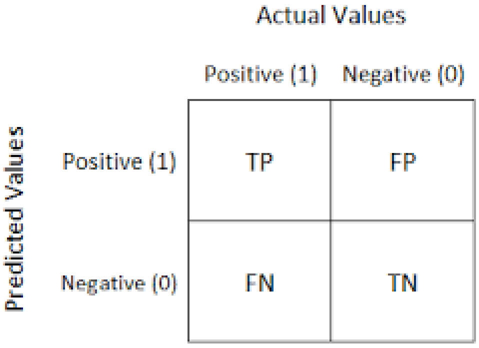


Models	Cross Validation (k_fold = 7)	Train Test Split (70:30)						
				Precision		Recall		
	ROC_AUC	Train Accuracy	Test Accuracy	Un-Popular	Popular	Un-Popular	Popular	
Logistic Regression	68.82%	68.96%	63.86%	65.00%	63.00%	66.00%	62.00%	
KNN	59.07%	73.00%	59.00%	61.00%	59.00%	63.00%	56.00%	
Decision Tree	57.91%	98.00%	57.00%	59.00%	57.00%	59.00%	57.00%	
Random Forest	67.20%	67.90%	68.00%	61.00%	63.00%	71.00%	52.00%	
Bagged DT	67.04%	100%	62.00%	59.05%	57.45%	59.90%	57.00%	
Bagged Log Reg	68.79%	63.95%	63.95%	65.00%	63.00%	66.00%	62.00%	
Ada Boost Log Reg	65.48%	60.82%	61.38%	62.00%	61.00%	64.00%	58.00%	
Ada Boost DT	70.59%	66.71%	65.92%	66.00%	65.00%	68%	64.00%	
Ada Boost RF	69.72%	100%	64.34%	65.00%	64.00%	66.00%	62.00%	
Gradient Boosting	72.79%	68.52%	66.65%	67.00%	66.00%	68.00%	65.00%	

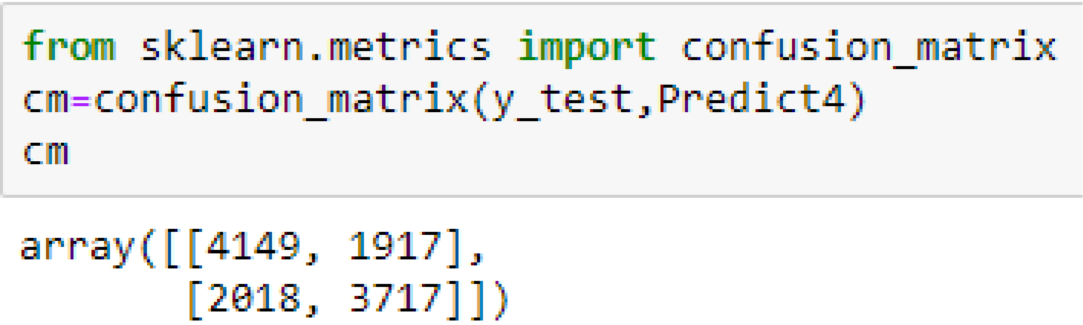
Gradient boost algorithm has an accuracy of 72.79% with training accuracy of 68.52% and testing accuracy of 66.65%. Precision score of popular and unpopular are 67% and 66% respectively. Recall score is 68% and 65%. Hence, gradient boost algorithm is the better fit model for the data set.

We also observed that even with out scaling there was not much change in the accuracy score.

Confusion Matrix:

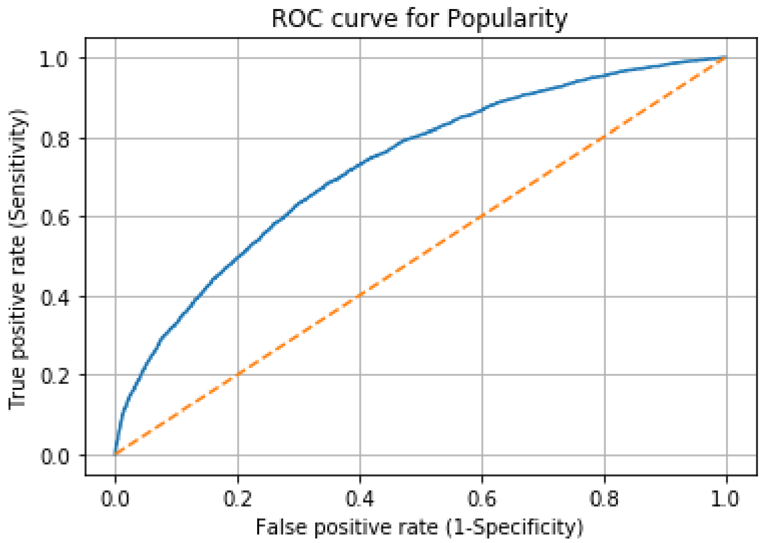


	Positive (1)	Negative (0)
Positive (1)	TP	FP
Negative (0)	FN	TN



```
from sklearn.metrics import confusion_matrix
cm=confusion_matrix(y_test,Predict4)
cm
```

```
array([[4149, 1917],
       [2018, 3717]])
```



COLCLUSION & RECOMMNATIONS:

In this project, we are working on retail analytics, using a dataset on Mashable a media company which publishes blogs and articles. The main objective is to predict whether the article will be popular or not depending on various features like the number of words the article consists or the number of images, videos or links were shared.

While working on visualising the features we found that the there were not much correlation between the dependent and independent feature but when each column were checked against shares, we got some insights like when the title had neither less nor more number of words in the title, that particular article got more number of shares. Another observation was when the words in content were in the range 6-15 the articles were popular. There were some extreme values which was driving down the accuracy of the models. Using IQR we removed those extreme values and normalized the data with min max scaler. PCA had no impact on improving the model accuracy. Gradient boost algorithm worked will though there was not much relation between the attributes.

- The number of keywords in the metadata influences the shares to a margin. The higher the value the better the shares chances. A value upward of 5 is recommended.
- The content should be less than 1500 words. The lesser the better.
- Title should be between 6 - 17 words.
- Unique words should be between 0.3 - 0.8%
- No. of links between 1 and 40 is preferred.
- Images - 0 to 3
- Videos – 0 to 25
- Minimal images and videos will make an article more interesting
- More articles are getting published on World data channel.
- Lifestyle and entertainment-based articles are preferred more by people.
- Best popular articles are usually posted on Mondays and Wednesday (and a bit of Tuesdays).
- Sundays and Saturdays (Weekends generally) are the worsts days to publish an article.
- Articles that talks about current trending are better for shares