GREAT LAKES INSTITUTE OF MANAGEMENT

greatlearning

PGPDSE-FT.G.NOV19 Batch

Capstone Project Report

Online News Popularity

Submitted By:

Sahil Pahuja

Sanket Khadanga

Ankur Bisht

Ved Lad

Prem Shankar

Mentored By:

Mrs. Anjana Agrawal

ABSTRACT

With the expansion of the Internet, more and more people enjoy reading and sharing online news articles. The number of shares under a news article indicates how popular the news is. In this project, we intend to find the best model and set of features to increase the popularity of online news, using machine learning techniques. Our data comes from Mashable, a well-known online news website. We implemented different learning algorithms on the dataset, ranging from various classification technique like Logistic Regression, Decision Tree ,Random Forest & other Tree Boosted Models. Their performances are recorded and compared. Feature selection methods are used to improve performance and reduce features. LightGBM Forest turns out to be the best model for prediction, and it can achieve an auc score of 73.4% with optimal parameters. Our work can help online news companies to improve news popularity.

Keywords - Machine learning; Classification; Popularity prediction; Feature selection; Model selection

ACKNOWLEDGEMENT

At the outset, we are indebted to our Mentor Mrs. Anjana Agrawal for her time, valuable inputs and guidance. Her experience, support and structured thought process guided us to be on the right track towards completion of this project.

We are fortunate to have Ms. Akshita Sawhney as our TA –Academic Delivery. Her in-depth knowledge coupled with her passion in delivering the subjects in a lucid manner has helped us a lot. We are thankful to her for her guidance towards entire coursework

We also thank all the course faculty of the DSE program for providing us a strong foundation in various concepts of analytics & machine learning.

Date: 27-06-2020

Place: Gurugram

CERTIFICATE OF COMPLETION

This is to certify that the project titled **"CAPSTONE – PROJECT REPORT – ONLINE NEWS POPULARITY"** was undertaken and completed under the supervision of Mrs. Anjana Agrawal for the Post Graduate Program in Data Science and Engineering (PGPDSE).

Mentor: Mrs. Anjana Agrawal

EXECUTIVE SUMMARY

Background and Need:

Tons of news, stories and articles are published on website every day. The author or editor would like their articles get shared and referred around the world as many times as it can be. But even the most skilful journalists can't be completely sure that their news directly hit people's tastes, no matter how well organized or gorgeous it might be. There certainly exists a large amount of features contributing to an impressive online news or article. If one can know what kind of news people mostly like prior to the publication, creating an amazing article is just a matter of time and proper modification.

Scope and Objective:

Our objective is to present a novel methodology to model and predict the popularity of online news. We first introduce a new strategy and mathematical model to capture view patterns of online news. After a thorough analysis of such view patterns, we find the features or parameters which impact the popularity of an article significantly. Second, we turn to the prediction of future popularity, given continuous target variable (number of shares), making it a classification problem by making sub categories of target variable. Traditionally linear regression is used for the application under study, we show that the more expressive tree-based methods proves beneficial for predicting news popularity. We compare the results of both and show how beautifully the tree-based model improve the accuracy.

Approach and Methodology:

The dataset has been downloaded from the UCI Machine Learning Repository. The input to our algorithm is a large list of features of articles which were published in Mashable: popularity of referenced articles; natural language features (e.g. global subjectivity and polarity); popularity of articles used the same keyword; number of digital media (e.g. images and videos) and published time (e.g. day of the week). We use these features to predict the popularity of an article prior to its publication. We solve this problem in three ways. Firstly, we predicted the exact shares using different count based regression model like Poisson regression, negative binomial regression. We go for count based regression model as number of shares has to be a countable number not in fractions. Secondly, we divide the target columns into 2 sub categories by using median of target variable i.e. 1400 as threshold. If number of shares are more than 1400 then it is considered popular else it is not popular. Lastly, we try and see if number of shares can be further classified into more than 2 categories. We use clustering to find out optimal number of categories and check if problem can be converted into multi class classification problem.

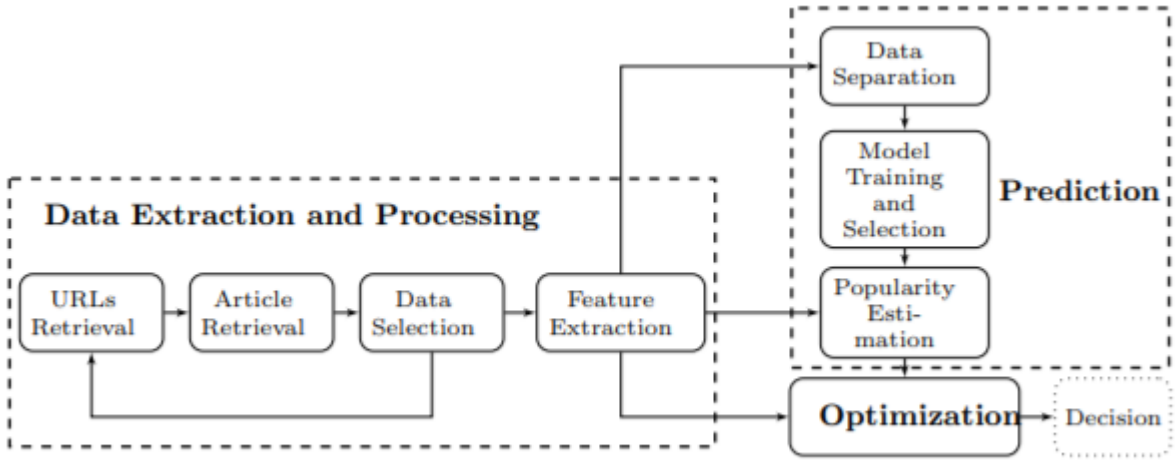


Fig: workflow architecture

PROJECT OVERVIEW

The Internet is an important tool for sharing messages. A recent survey has shown that around 50% of teenagers and adults in America choose to read online news in their daily life (Pew Research Centre, 2018) in 2016. This percentage has increased a lot in the past few years. For reasons that more and more people read online news and editors want their news to be popular, it would be meaningful to build a system to predict whether a news will be popular or not. Such system can not only help editors find how they could improve their news but also can bring significant commercial value. Here to predict the same machine learning models can play a vital role and hence, we selected this dataset for our project.

The original data contains 60 features including URL, rate of positive words, title sentiment polarity and so on. And 39645 samples are provided. All the features are numerical except URL. Considering that URL is not an important feature and there is no effective way to convert it from string to numerical, I deleted this feature directly. The original target is the number of shares. In order to do the classification task, I adopted the method used by Fernandes, Vinagre and Cortez (2015), treating 1400 shares as the threshold, which means if the number of shares of a news was larger than 1400, then it was a popular news, otherwise unpopular. And I used number 1 to represent popular and 0 to represent unpopular in the system. Moreover, to compare my system with theirs, I separated the dataset as they did, 70% for training and 30% for testing.

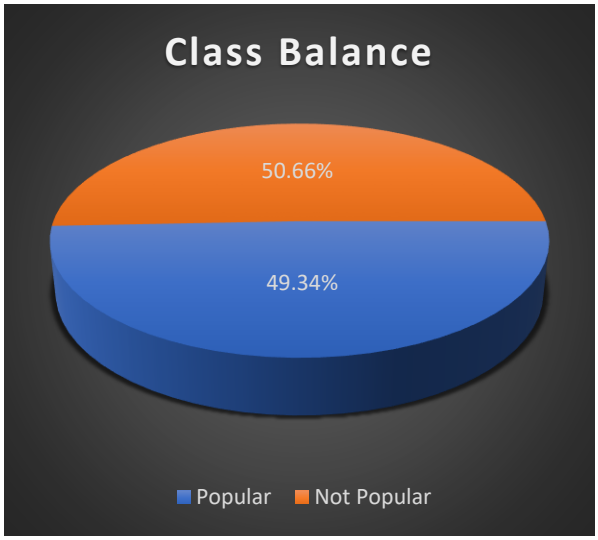


Fig: Class Balance

The dataset consists of feature belonging to 39645 articles. Of the 39645 articles in the dataset, 50.655% (20,082) were negative class samples that were not popular, and the 49.344% (19562) were positive class samples ending which became popular.

The dataset was formed so that each article would belong to a particular category whether popular or not popular. Use the dataset to build a Machine Learning Model to predict whether the article will be popular or not.

Data Description:

Aspects	Features
Words	▪ **Number of words of the title/content** ▪ **Average word length** ▪ **Rate of unique/non-stop words of content**
Links	▪ **Number of links** ▪ **Number of links to other articles in Mashable**
Digital Media	▪ **Number of images/videos**
Publication Time	▪ **Day of the week/weekend**
Keywords	▪ **Number of keywords** ▪ **Worst/best/average keywords** ▪ **Article category**
NLP	▪ **Closeness to five LDA topics** ▪ **Title/Text polarity/subjectivity** ▪ **Rate and polarity of positive/negative words** ▪ **Absolute subjectivity/polarity level**
Target	▪ **Number of shares of Mashable**

Words:
Features related to word basically tell different attribute related to words in each article

- n_tokens_title: Number of words in the title
- n_tokens_content: Number of words in the content
- n_unique_tokens: Rate of unique words in the content
- n_non_stop_words: Rate of non-stop words in the content
- n_non_stop_unique_tokens: Rate of unique non-stop words in the content
- average_token_length: Average length of the words in the content

Links and digital media
Features like number of links or link to other articles, count of images and videos

- num_hrefs: Number of links
- num_self_hrefs: Number of links to other articles published by Mashable
- num_imgs: Number of images
- num_videos: Number of videos

Publication Time:
The day article was published. Days have been already been converted into 7 dummies

News Category:
Whether article is related news about world, technology, social media, entertainment, lifestyle, business and rest were classified as Others.

Keywords:
Keywords here refer to words in article which were left out after text pre-processing and have been classified into three types: best, worst and avg. Depending on the usage of different keywords while sharing the articles, the count of shares has been recorded

- num_keywords: Number of keywords in the metadata
- kw_min_min: Worst keyword (min. shares)
- kw_max_min: Worst keyword (max. shares)
- kw_avg_min: Worst keyword (avg. shares)
- kw_min_max: Best keyword (min. shares)
- kw_max_max: Best keyword (max. shares)
- kw_avg_max: Best keyword (avg. shares)
- kw_min_avg: Avg. keyword (min. shares)
- kw_max_avg: Avg. keyword (max. shares)
- kw_avg_avg: Avg. keyword (avg. shares)

Self-Referencing links:
It refers to number of shares of articles/links mentioned in article itself.
For example, in news article, sometime links/articles are provided to give backdrop of the article or maybe to relate to a similar event there may be a link provided

- self_reference_min_shares: Min. shares of referenced articles in Mashable
- self_reference_max_shares: Max. shares of referenced articles in Mashable
- self_reference_avg_sharess: Avg. shares of referenced articles in Mashable

Natural Language Processing:
Latent dirichlet allocation algorithm has been used to classify all articles into 5 topics:
LDA_00, LDA_01, LDA_02, LDA_03, LDA_04.
Polarity and subjective for title and content has been calculated separately.

- LDA_00: Closeness to LDA topic 0
- LDA_01: Closeness to LDA topic 1
- LDA_02: Closeness to LDA topic 2
- LDA_03: Closeness to LDA topic 3
- LDA_04: Closeness to LDA topic 4
- title_subjectivity: Title subjectivity
- title_sentiment_polarity: Title polarity
- abs_title_subjectivity: Absolute subjectivity level
- abs_title_sentiment_polarity: Absolute polarity level
- global_subjectivity: Text subjectivity
- global_sentiment_polarity: Text sentiment polarity
- global_rate_positive_words: Rate of positive words in the content
- global_rate_negative_words: Rate of negative words in the content
- avg_positive_polarity: Avg. polarity of positive words
- min_positive_polarity: Min. polarity of positive words
- max_positive_polarity: Max. polarity of positive words
- avg_negative_polarity: Avg. polarity of negative words
- min_negative_polarity: Min. polarity of negative words
- max_negative_polarity: Max. polarity of negative words

Data Cleaning and Formatting:

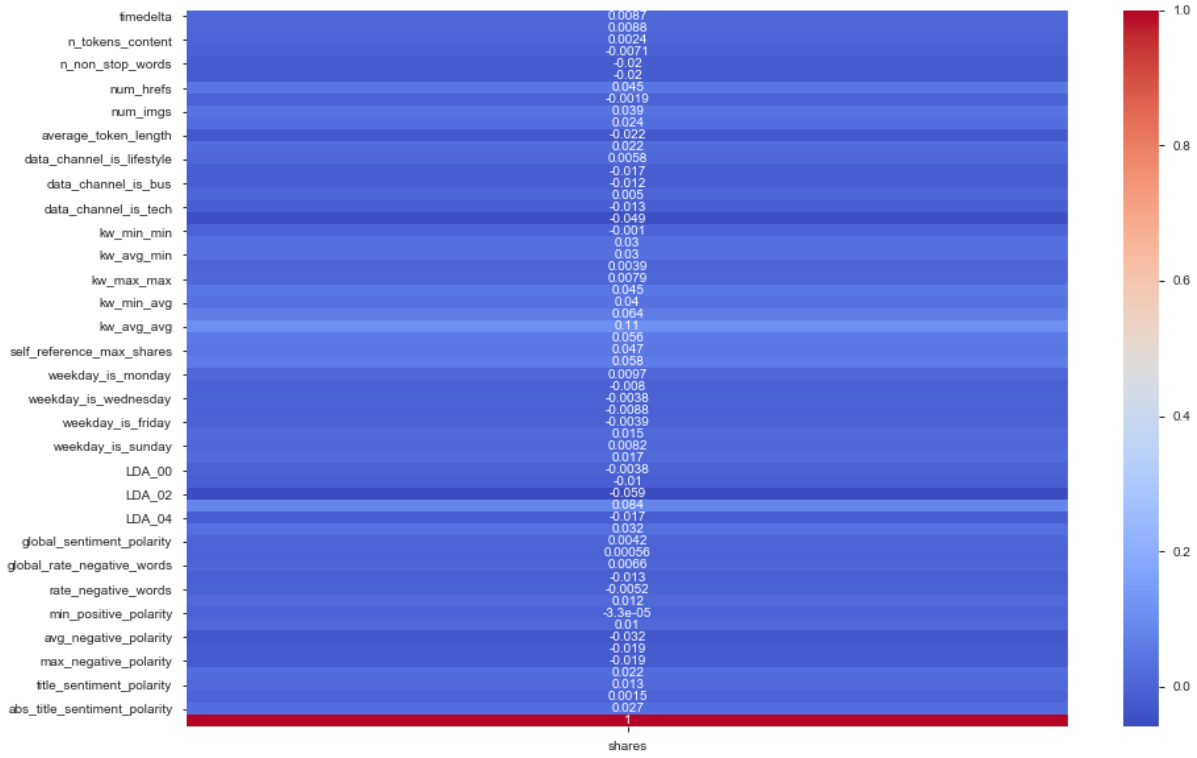
Most of the datasets on UCI are cleaned and same is with our dataset. There are no missing values.

Missing Values	There are no missing values.	
Duplicate Records	There are no duplicate records.	
Outlier Treatment	Distribution plots were made to check the skewness and outliers in the dataset.	Since, outliers are an important part of the dataset we retained them and transformed to map them to normal distribution using boxcox transformation
Encoding	Published Day, data channel of article	Both features are already converted into dummies
Sampling	There is no imbalance of target so no oversampling or undersampling technique is applied	

EXPLORATORY DATA ANALYSIS

The purpose of EDA is to find anomalies, patterns, trends, or relationships in the given dataset. For EDA, we will focus on a bivariate analysis between features and the Number of shares, because this is the target for our machine learning models. We will go by the categories defined above for features.

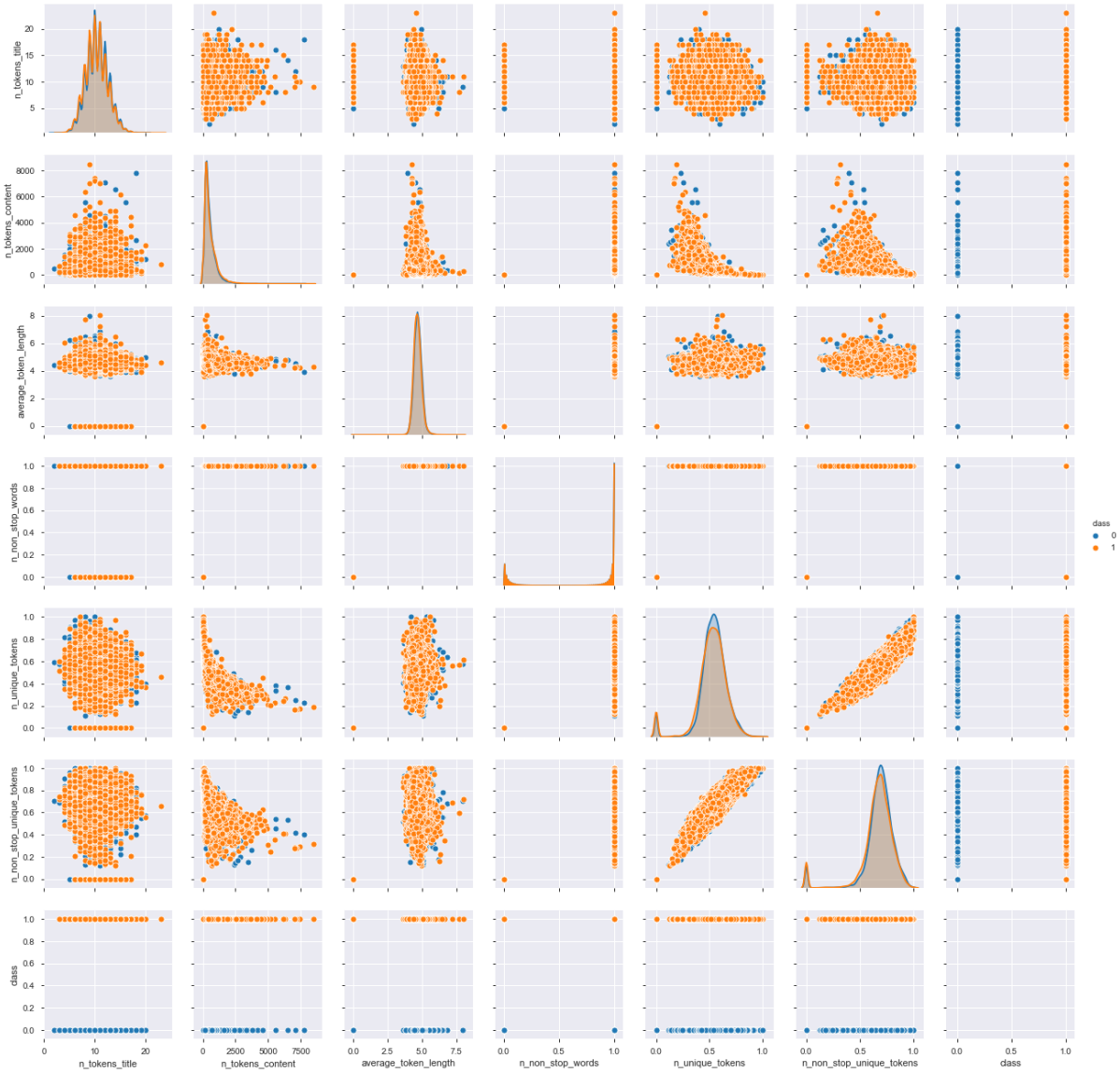
First, let's look at correlation plot to get idea about how features are related to target



We can see here there is very less correlation between features and target so we should not expect high accuracy for any model.

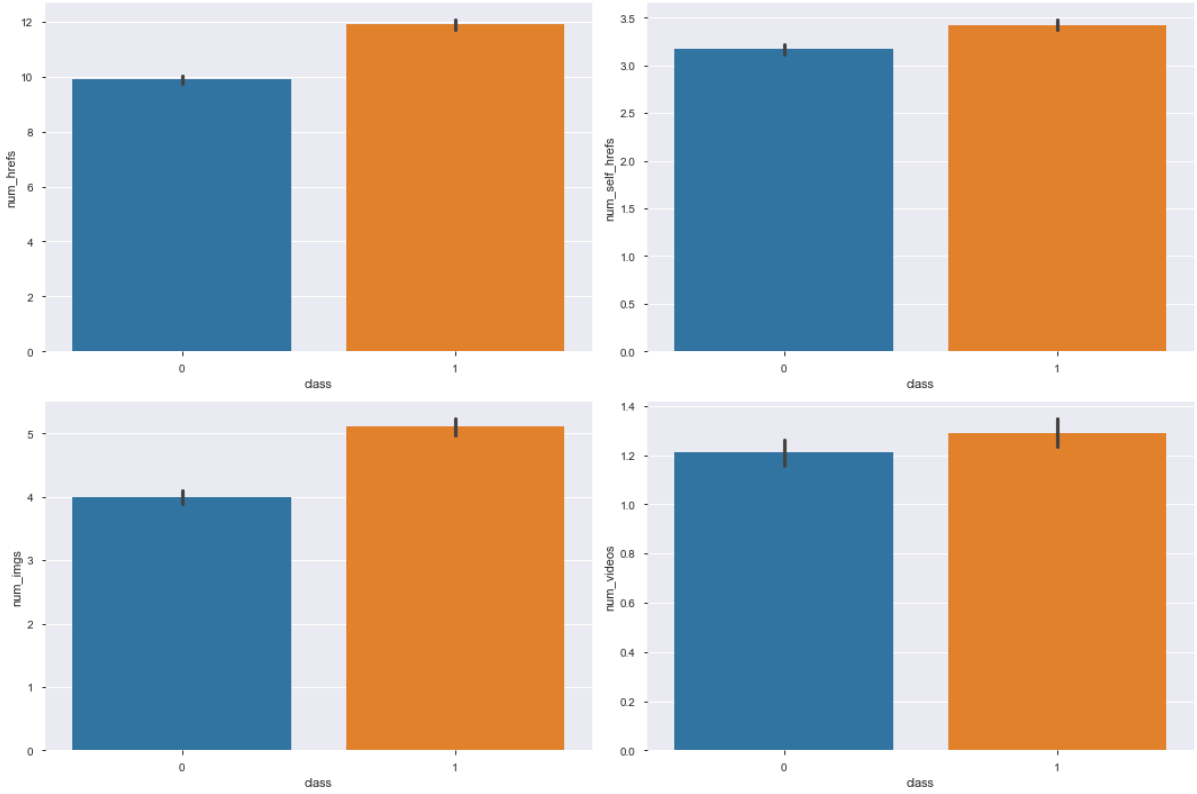
Words

Here we have pairplot for all words related features. We have use hue='target' so we can see if there is separation between the two classes of target corresponding to each feature. The more separate are the distribution of classes on diagonal for the feature, more significant is the feature. Only 'n_unique_tokens' and 'n_non_stop_unique_tokens' are significant to some extent as per this pairplot.



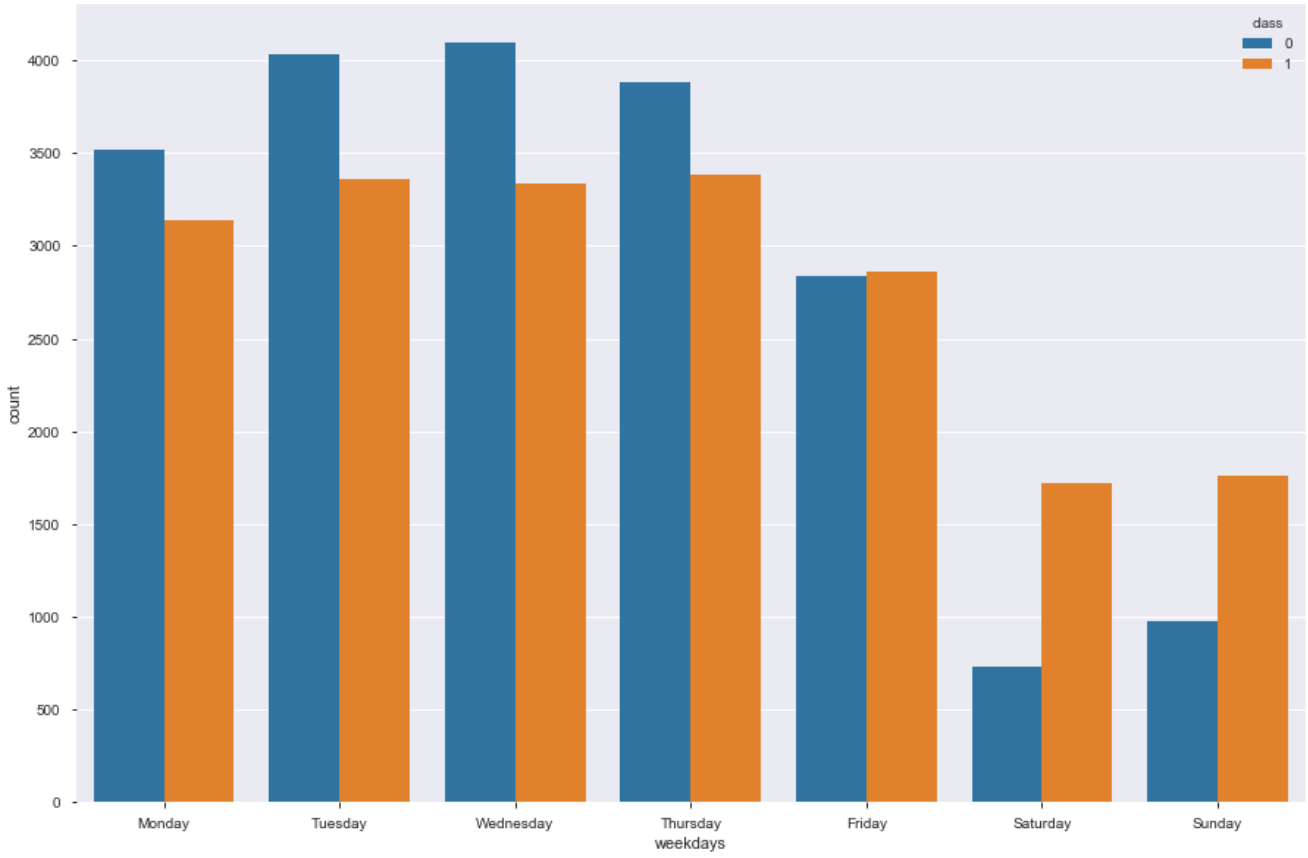
Links & Digital Media:

Mean number of videos are same for both the classes so it doesn't seem like a significant feature where number of hyperlinks may be significant in comparison to other features.



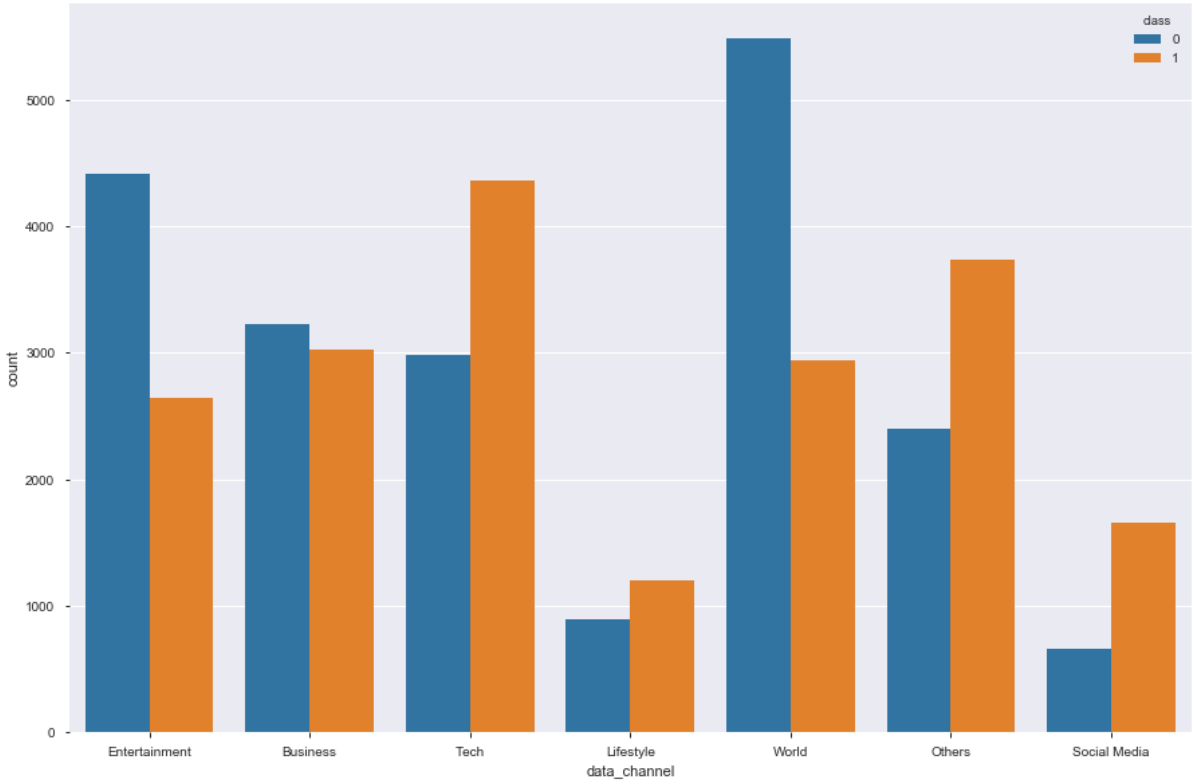
Publication Time:

We can see here % of popular articles out of total number of articles published on a day is higher if article is published on Saturday or Sunday. There are high chances that article become popular if published on weekend.



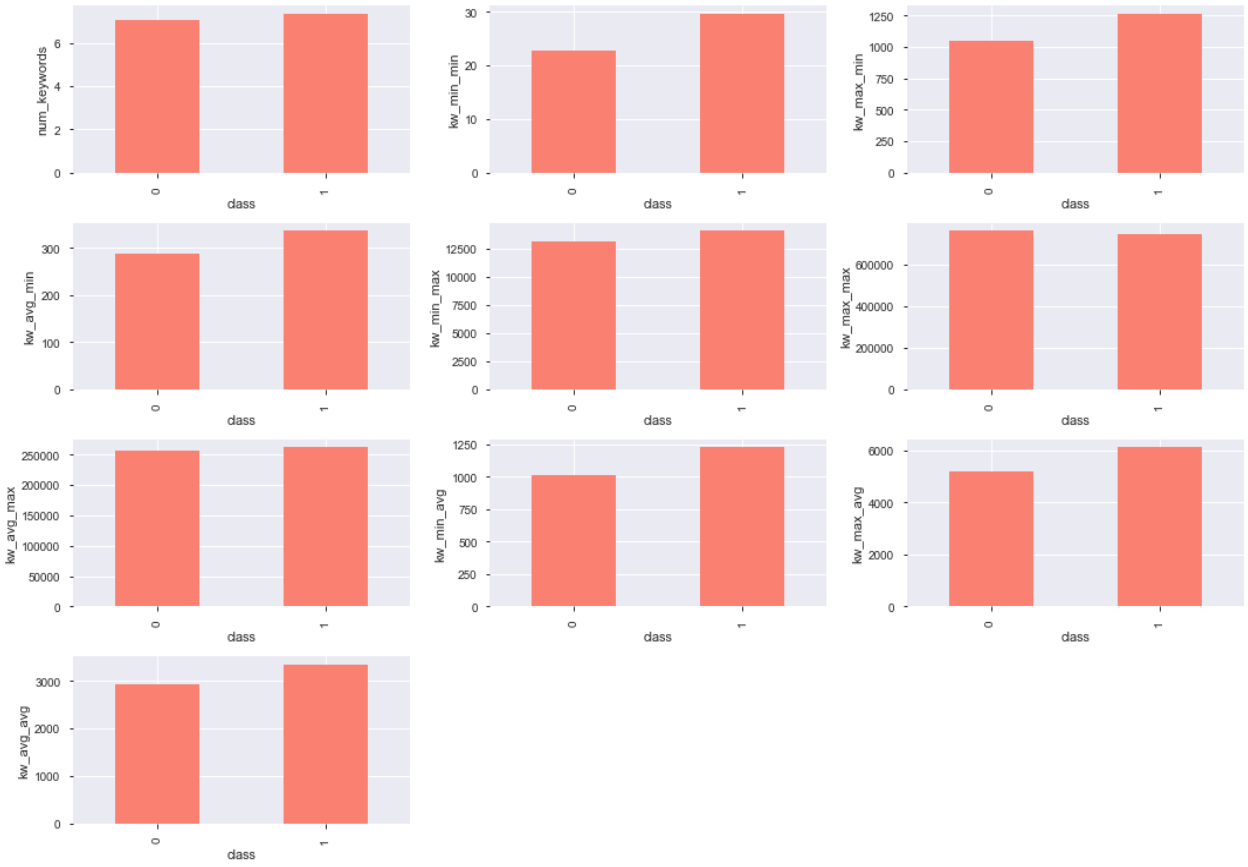
News Category

We can see here news article belonging to technology, social media and lifestyle has higher chance of becoming popular than other categories.
We can also validate the data here as we can see the count of world news is highest and it makes sense as well because during the period 2013-2015, major events happened that impacted the whole world like outbreak of Ebola in Africa, Malaysian airlines plane MH370 went missing.



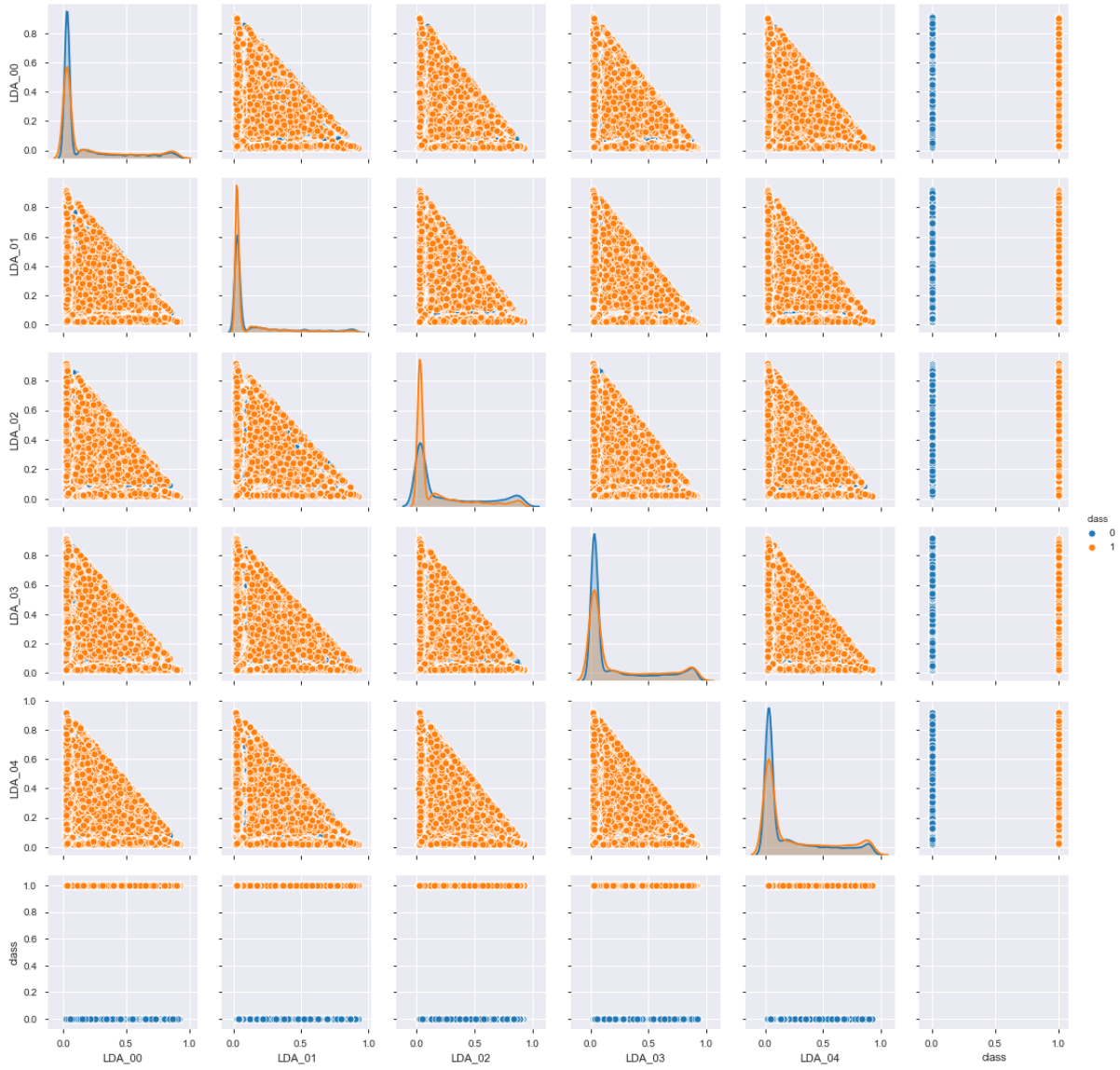
Keywords

kw_max_avg, kw_min_avg and kw_avg_avg have significant difference in their mean for classes popular and not popular while rest have similar mean for both the classes



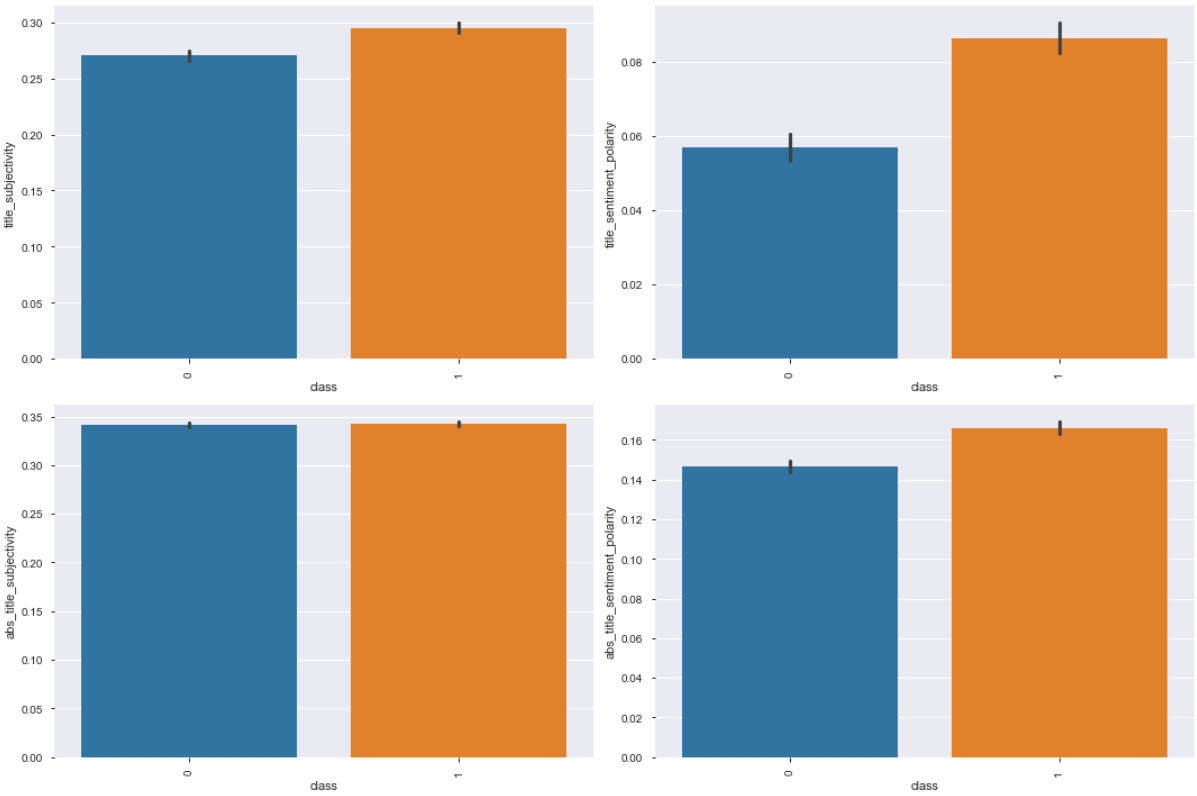
Closeness to LDA topic:

Distribution is pretty much similar for all features related to LDA_topics.



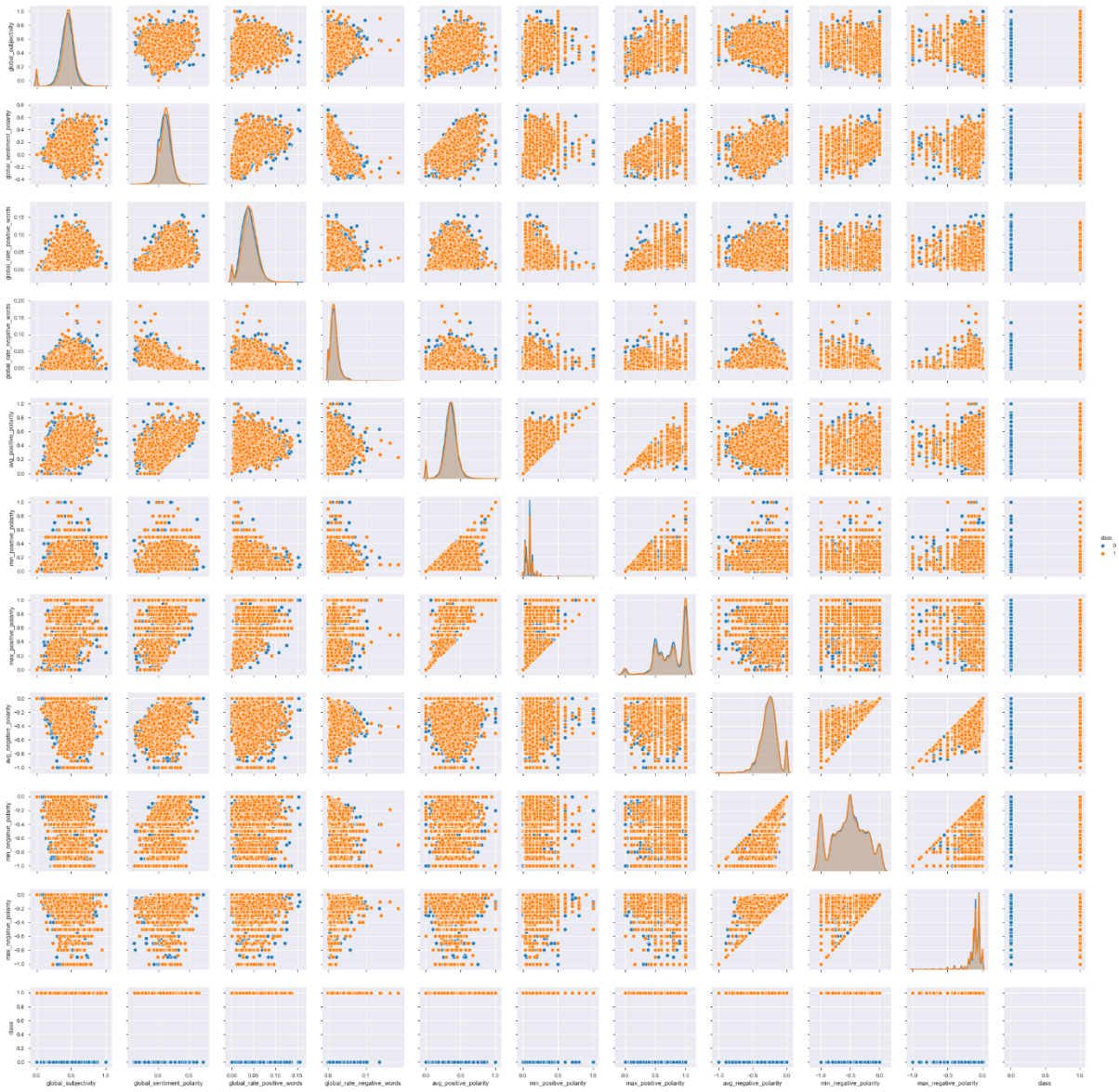
Title Related features:

Difference in Mean of title_sentiment_polarity for the 2 classes is high and moderate for abs_title_sentiment_polarity



Polarity and Subjectivity

KDE plot is overlapping for both the classes in all the features so none of them seems significant for prediction.

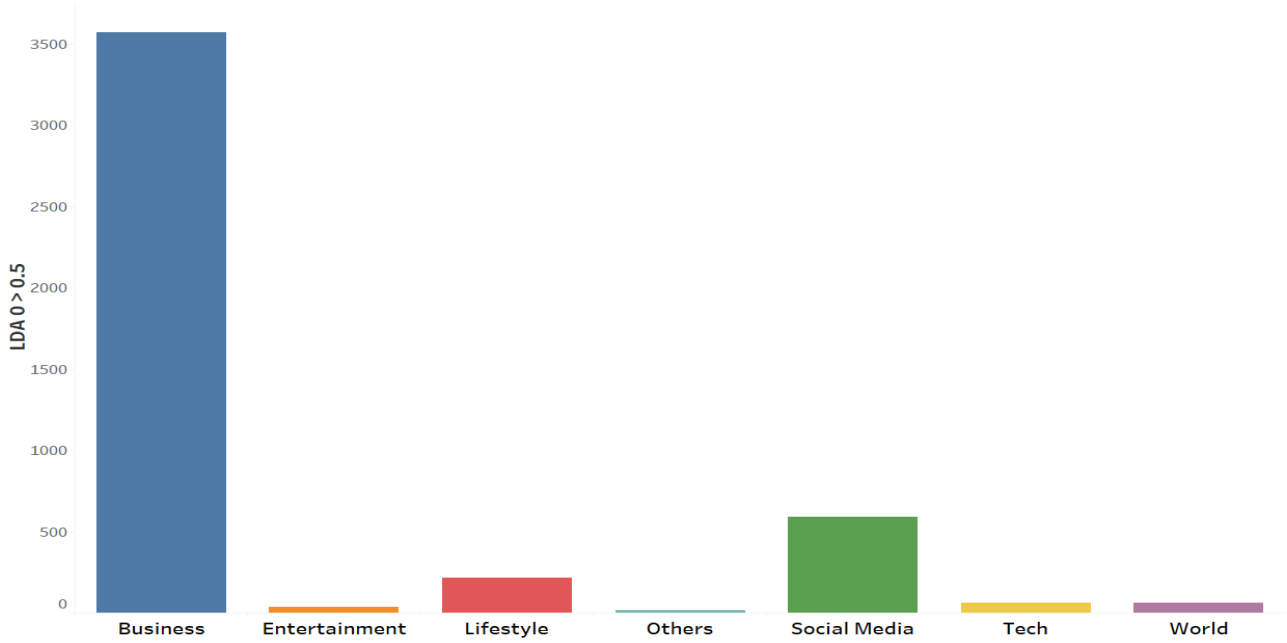


Top Insights

LDA(Latent Dirichlet allocation) is a topic modelling technique where the task is to identify topics that best describes a set of documents. So in our dataset we had 5 LDA columns which means our articles are divided into 5 topics. So, we know LDA provides probability so if a news article is closer to one of the LDA generated topics then it will have value at least greater than .5. Here we have our dataset by each of the LDA topic with value >.5 and we were able to identify the LDA topic by checking the distribution of channel (Business, World etc) in the filtered data

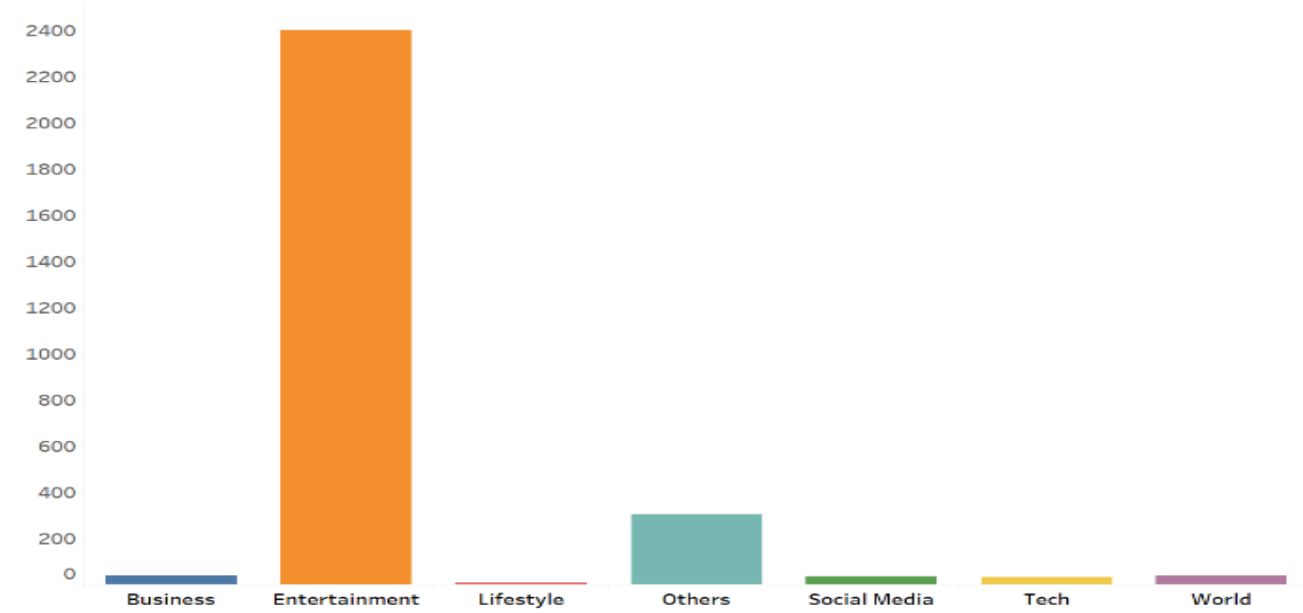
LDA 00 >.5

LDA 00 represents Business news as Business articles have highest count in filtered dataset.



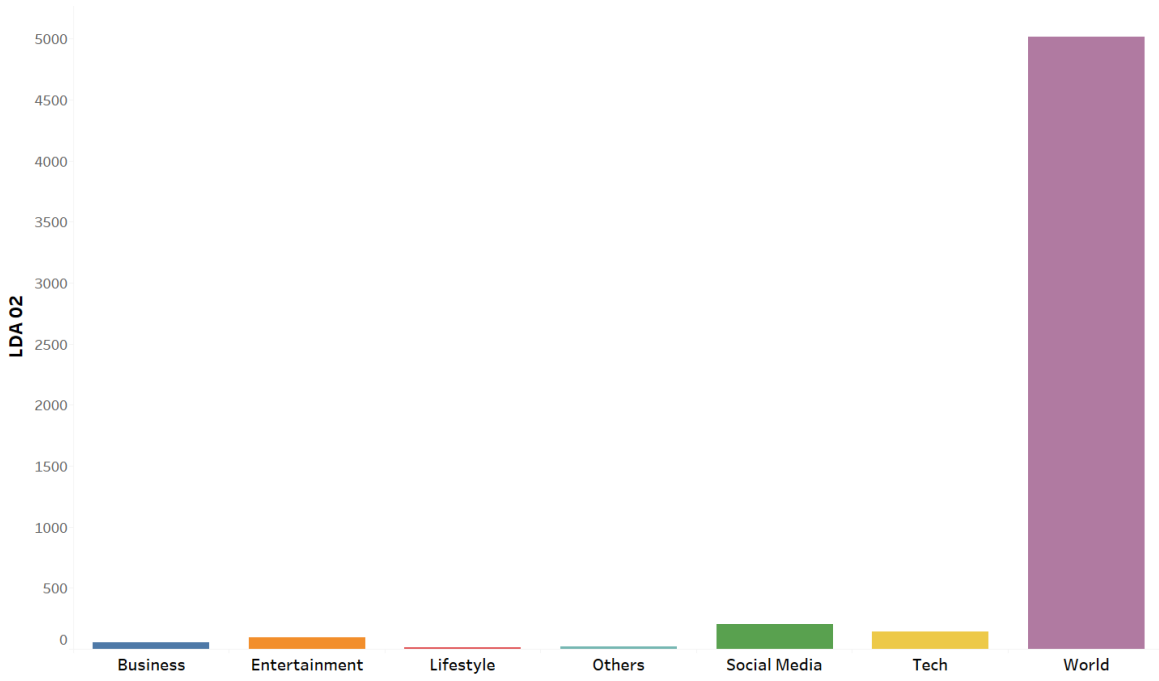
LDA 01>.5

LDA 01 represents Entertainment news as entertainment articles have highest count in filtered dataset.



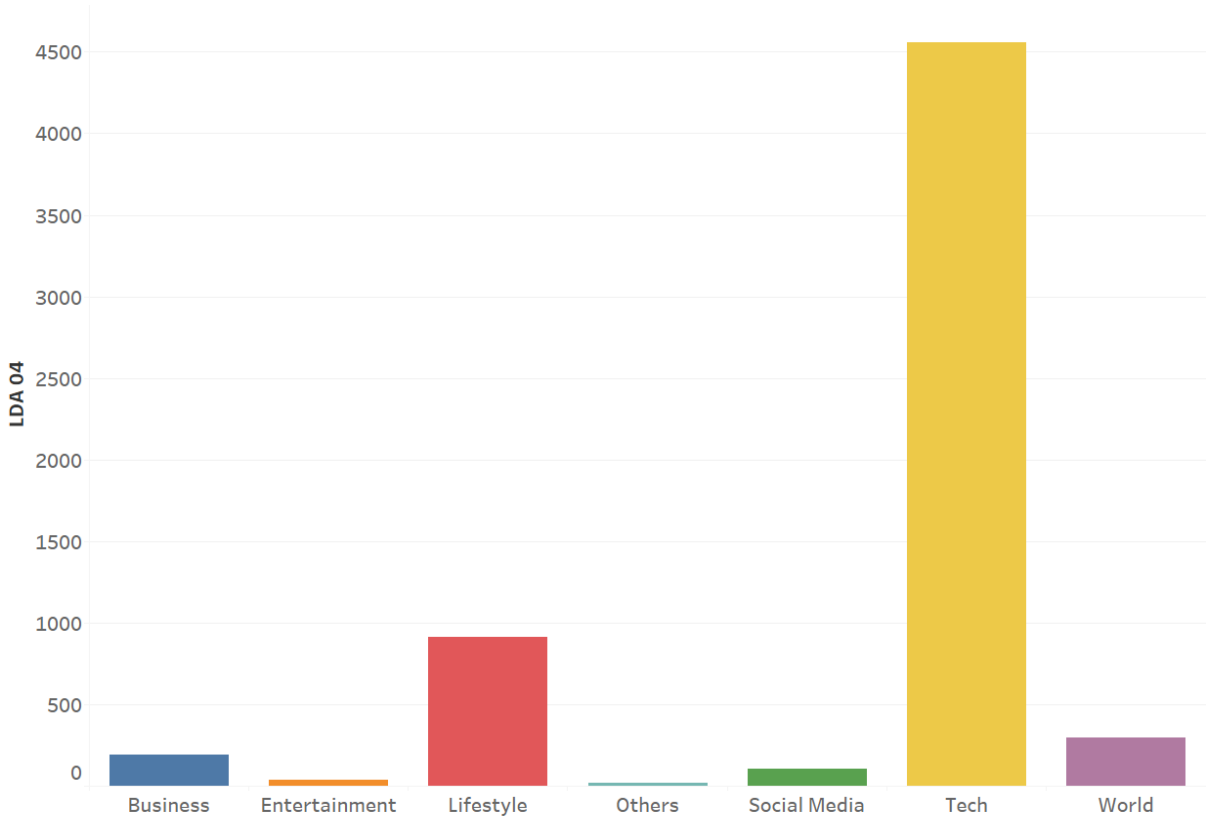
LDA 02 >.5

LDA 02 represents World news as world articles have highest count in filtered dataset.



LDA 04>.5

LDA 04 represents Tech news mainly as Tech articles have highest count in filtered dataset.



STATISTICAL ANALYSIS

We tried to observe the relationship between all the features and the target statistically. We check whether the feature distribution is following the conditions for parametric test ie distribution should be normal ,variance of the 2 distribution should be equal and it should be highly randomised. We check the normality condition through Anderson-Darling test as the number of samples >5000 and variance condition through levene test. Thereafter, we went with corresponding parametric or non parametric test. We used 2 sample independent t-test if the feature was numerical (& target is categorical) and Mannwhitneyu if the conditions were not followed. We went with Chi Square test if the feature was categorical.
We have done all the tests on **significance level .05**. Below are summarized results :

Feature	Test Used	(Test Statistic,P value)	Decision
n_tokens_title	Mannwhitneyu	(186294064.0, 1.330869103162258e-19)	Reject Null Hypothesis
n_tokens_content	Mannwhitneyu	(193221302.0, 0.002480246263434255)	Reject Null Hypothesis
average_token_length	Mannwhitneyu	(185584157.0, 9.224955818124774e-22)	Reject Null Hypothesis
n_non_stop_words	Mannwhitneyu	(193678332.0, 0.007625170962600819)	Reject Null Hypothesis
n_unique_tokens	Mannwhitneyu	(186145700.5, 9.361730835468883e-20)	Reject Null Hypothesis
n_non_stop_unique_tokens	Mannwhitneyu	(181606124.5, 5.7070320970527415e-39)	Reject Null Hypothesis
num_hrefs	Mannwhitneyu	(178317313.0, 2.3809339046487327e-57)	Reject Null Hypothesis
num_self_hrefs	Mannwhitneyu	(190209380.5, 1.769294177245526e-08)	Reject Null Hypothesis
num_imgs	Mannwhitneyu	(180922611.0, 5.128635789257361e-47)	Reject Null Hypothesis
num_videos	Mannwhitneyu	(190736532.0, 2.8925753163749233e-09)	Reject Null Hypothesis
num_keywords	Mannwhitneyu	(181009742.0, 5.7523294821106615e-43)	Reject Null Hypothesis
kw_min_min	Mannwhitneyu	(191696195.5, 1.2541396495957891e-06)	Reject Null Hypothesis
kw_max_min	Mannwhitneyu	(177947168.5, 1.8842106101942255e-59)	Reject Null Hypothesis
kw_avg_min	Mannwhitneyu	(175627109.5, 9.712112594152472e-75)	Reject Null Hypothesis
kw_min_max	Mannwhitneyu	(187400211.5, 7.405847852256953e-17)	Reject Null Hypothesis
kw_max_max	Mannwhitneyu	(188154339.5, 1.1426421541485793e-22)	Reject Null Hypothesis
kw_avg_max	Mannwhitneyu	(193648434.5, 0.0074537125120966346)	Reject Null Hypothesis
kw_min_avg	Mannwhitneyu	(178568183.0, 2.502303988938753e-60)	Reject Null Hypothesis
kw_max_avg	Mannwhitneyu	(155839376.5, 3.1187200168383085e-278)	Reject Null Hypothesis
kw_avg_avg	Mannwhitneyu	(149637815.5,0.0)	Reject Null Hypothesis

self_reference_min_shares	Mannwhitneyu	(162076202.0, 3.3586523549066074e-201)	Reject Null Hypothesis
self_reference_max_shares	Mannwhitneyu	(166036706.5, 5.8482179181631545e-158)	Reject Null Hypothesis
self_reference_avg_sharess	Mannwhitneyu	(161194116.5, 1.7029746407077358e-211)	Reject Null Hypothesis
LDA_00	Mannwhitneyu	(189980869.0, 7.84052379963435e-09)	Reject Null Hypothesis
LDA_01	Mannwhitneyu	(179728778.0, 6.441606242246763e-49)	Reject Null Hypothesis
LDA_02	Mannwhitneyu	(165174341.5, 6.246218165765207e-166)	Reject Null Hypothesis
LDA_03	Mannwhitneyu	(186580322.0, 2.8384133763441035e-18)	Reject Null Hypothesis
LDA_04	Mannwhitneyu	(183604607.0, 1.1461499212427687e-29)	Reject Null Hypothesis
title_subjectivity	Mannwhitneyu	(188888112.0, 1.832196683175706e-12)	Reject Null Hypothesis
title_sentiment_polarity	Mannwhitneyu	(184481429.0, 1.6671727772752878e-29)	Reject Null Hypothesis
abs_title_subjectivity	Mannwhitneyu	(196198036.0, 0.4160587029920939)	Reject Null Hypothesis
abs_title_sentiment_polarity	Mannwhitneyu	(190005634.0, 8.257102785681583e-10)	Reject Null Hypothesis
global_subjectivity	Mannwhitneyu	(174479398.5, 5.696907687225355e-83)	Reject Null Hypothesis
global_sentiment_polarity	Mannwhitneyu	(178170553.5, 4.5560704003495906e-58)	Reject Null Hypothesis
global_rate_positive_words	Mannwhitneyu	(180381205.0, 2.5008729273842684e-45)	Reject Null Hypothesis
global_rate_negative_words	Mannwhitneyu	(190856535.5, 5.146618040070318e-07)	Reject Null Hypothesis
avg_positive_polarity	Mannwhitneyu	(186597412.0, 3.2327045124640064e-18)	Reject Null Hypothesis
min_positive_polarity	Mannwhitneyu	(186561888.5, 1.519476707788335e-19)	Reject Null Hypothesis
max_positive_polarity	Mannwhitneyu	(184405232.5, 6.58120253902758e-28)	Reject Null Hypothesis
avg_negative_polarity	Mannwhitneyu	(195451118.5, 0.19700276223304003)	Fail to reject null hypothesis
min_negative_polarity	Mannwhitneyu	(195414848.0, 0.18699501934224405)	Fail to reject null hypothesis
max_negative_polarity	Mannwhitneyu	196238533.5, 0.43504209970509766)	Fail to reject null hypothesis
Data_channel	Chi Square	(578.4531685111237, 1.0349991174202705e-121)	Reject Null Hypothesis
Days	Chi Square	(809.4454581750938, 1.4014686112275905e-171)	Reject Null Hypothesis

SKEWNESS TREATMENT

All the numerical features are highly skewed. To measure skewness, we have made a skewness table for all the features and applied boxcox transformation to treat it. The Skewness was checked after applying boxcox as well.

	features	skewness_before	skewness_after
0	n_tokens_title	0.165278	-0.001479
1	n_tokens_content	2.945817	0.148320
2	n_unique_tokens	-1.458581	0.154094
3	n_non_stop_words	-5.531757	-5.531757
4	n_non_stop_unique_tokens	-2.406115	0.090153
5	num_hrefs	4.013445	0.009333
6	num_self_hrefs	5.173277	-0.000940
7	num_imgs	3.947228	0.180154
8	num_videos	7.019447	0.671934
9	average_token_length	-4.575946	0.361297
10	num_keywords	-0.147258	-0.088994
11	kw_min_min	2.374903	0.547576
12	kw_max_min	35.327994	0.468495
13	kw_avg_min	31.305781	0.426021
14	kw_min_max	10.386263	-0.064142
15	kw_max_max	-2.644936	-1.762517
16	kw_avg_max	0.624345	-0.054478
17	kw_min_avg	0.468042	-0.223021
18	kw_max_avg	16.411515	0.766528
19	kw_avg_avg	5.760098	0.610982

20	self_reference_min_shares	26.264047	-0.029711
21	self_reference_max_shares	13.870689	-0.100724
22	self_reference_avg_sharess	17.913869	-0.090111
23	LDA_00	1.567428	0.431634
24	LDA_01	2.086686	0.481379
25	LDA_02	1.311661	0.300653
26	LDA_03	1.238681	0.342048
27	LDA_04	1.173096	0.243742
28	global_subjectivity	-1.372325	0.227649
29	global_sentiment_polarity	0.105416	0.025518
30	global_rate_positive_words	0.323230	-0.012868
31	global_rate_negative_words	1.492006	0.001887
32	rate_positive_words	-1.422888	-0.010640
33	rate_negative_words	0.407307	-0.000068
34	avg_positive_polarity	-0.724378	0.210270
35	min_positive_polarity	3.040595	0.013354
36	max_positive_polarity	-0.939589	-0.227606
37	avg_negative_polarity	-0.551781	0.069481
38	min_negative_polarity	-0.073173	-0.061590
39	max_negative_polarity	-3.459821	-0.114785
40	title_subjectivity	0.816049	0.253249
41	title_sentiment_polarity	0.396085	0.119313
42	abs_title_subjectivity	-0.624206	-0.414083
43	abs_title_sentiment_polarity	1.704159	0.496224

There is highly noticeable difference in skewness before and after boxcox transformation.

MODELLING TECHNIQUE

It's important to establish a naive baseline before we begin making machine learning models. If the models we build cannot outperform a naive guess then we might have to admit that machine learning is not suited for this problem. This could be because we are not using the right models, because we need more data, or because there is a simpler solution that does not require machine learning. Establishing a baseline is crucial so we do not end up building a machine learning model only to realize we can't actually solve the problem.

For any machine learning task, a good naive baseline is to run our model on simple cleaned data (i.e., original data with no null values), categorical features encoded and outlier treated. If after feature engineering our models or using ensemble models cannot do better than baseline then we need to rethink our approach.

As we stated in the beginning, we are going to take 2 target variables as follows:

- Target variable is categorical with two classes and perform binary classification
- Target variable is categorical with multiple classes and perform multi class classification

Binary Classification:

Next, we converted target into 2 classes using median of number of targets i.e. 1400 as threshold. We take 1 as popular and 0 as not popular.

Metric: AUC Score

There are a number of metrics used in machine learning tasks and it can be difficult to know which one to choose. Most of the time it will depend on the particular problem and if you have a specific goal to optimize for. Rather than calculating multiple metrics and trying to determine how important each one is, we should use a single metric.

Since, our classification problem is to predict the popularity of the article i.e., whether the article will be popular or not, we will be using 'AUC score as our evaluation metric'.

- AUC is **scale-invariant**. It measures how well predictions are ranked, rather than their absolute values.
- AUC is **classification-threshold-invariant**. It measures the quality of the model's predictions irrespective of what classification threshold is chosen.

Since our data is balanced so we don't need to go for F1_score as it is used for imbalanced dataset and we can't use accuracy because it might give us misleading results(if TP is very high but TN is 0 then also accuracy will be good).AUC Score would be best metric for our dataset i.e. area under the curve and it will be a reliable metric.

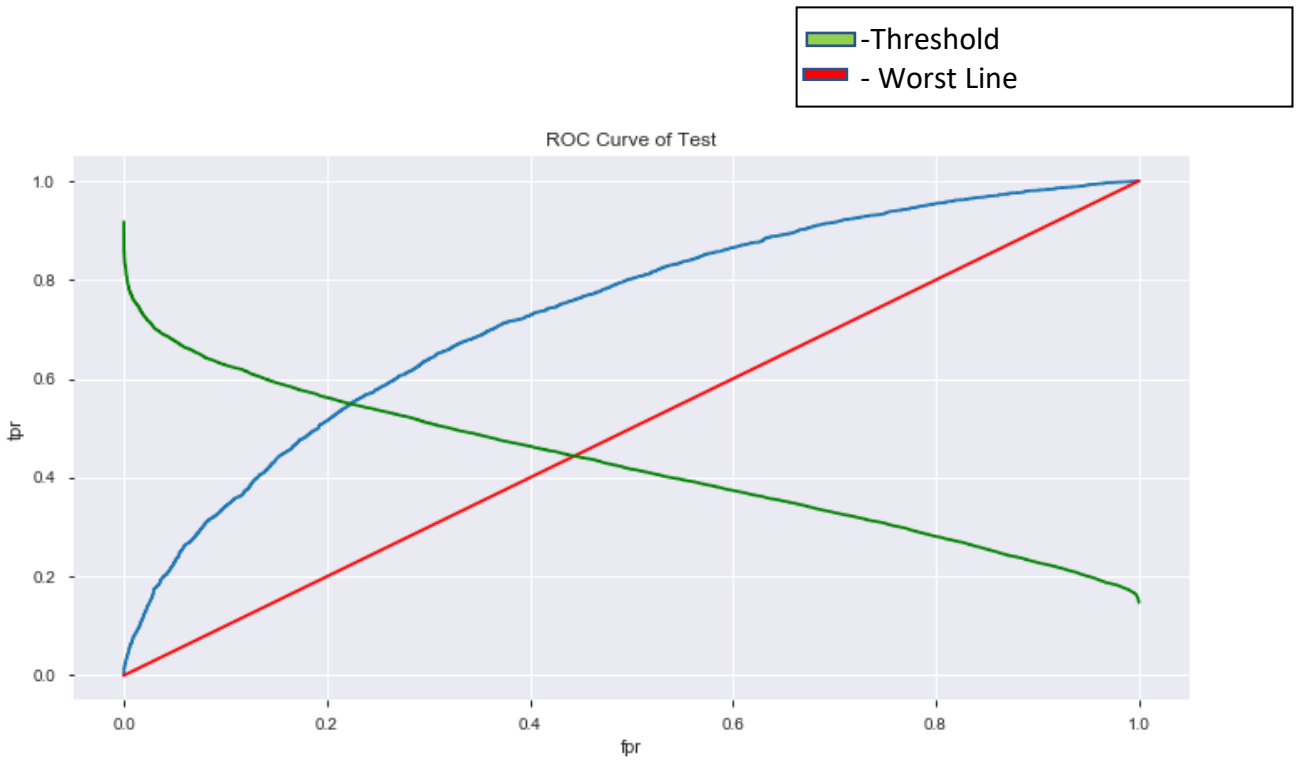
Base Model:

We performed the Logistic Regression, Decision Tree and Random Forest Classification on our dataset. For logistic regression we did not tune any hyperparameters and gradient descent was able to converge. For Decision Tree and Random Forest, we had to tune our hyperparameters because of limited computing power as fully grown decision tree without any hyperparameter tuning would lead to memory error otherwise hyperparameter tuning is not required for base model.

Below is the comparison of scores for the above three models

	Model	AUC_Score	Accuracy	Precision	Recall
0	Decison Tree Base	0.681615	0.634942	0.635892	0.621914
1	Random Forest Base	0.729924	0.669665	0.670328	0.660467
2	Logistic Regression Base	0.517912	0.502775	0	0

We can see clearly how roc curve is showing tpr and fpr for all thresholds and we can check the tpr and fpr for any threshold visually.



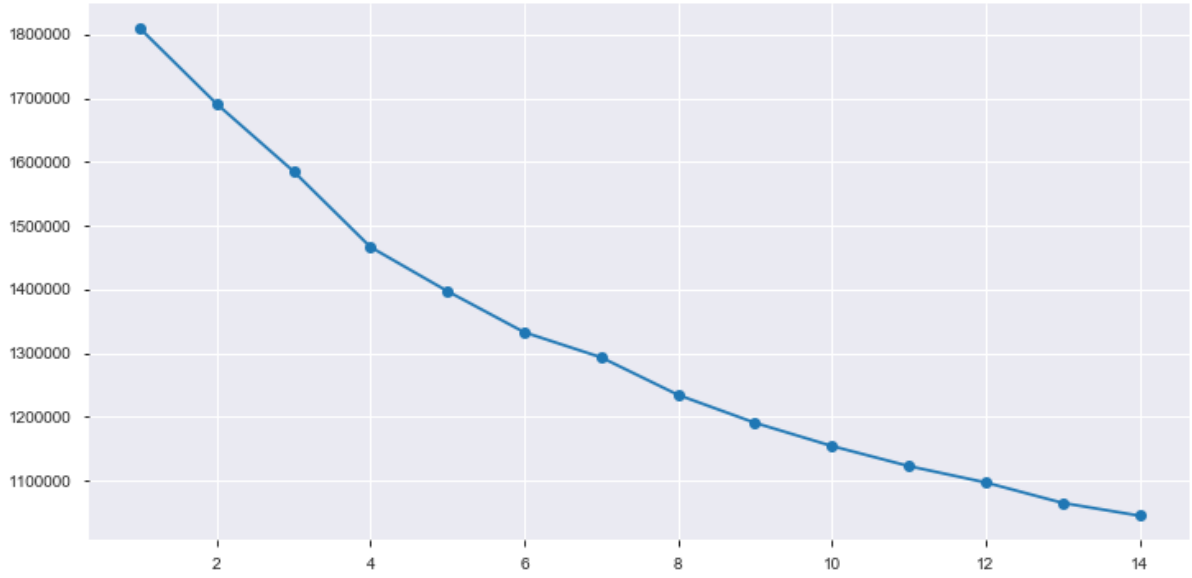
ROC curve for Random Forest

Multi Class Classification:

We checked the possibility of more than 2 classes and checked if we could get better results. We used clustering to find optimal number of clusters. Below are the results:

Elbow Method:

Firstly, we checked with elbow method for range 1-14, the number of clusters that could be formed. But as we can see, we are not getting concrete results here although dip is maximum from 3 to 4 clusters if checked numerically but no clear formation of elbow shape.
So, to we went ahead and did silhouette score test to find optimal number of clusters.



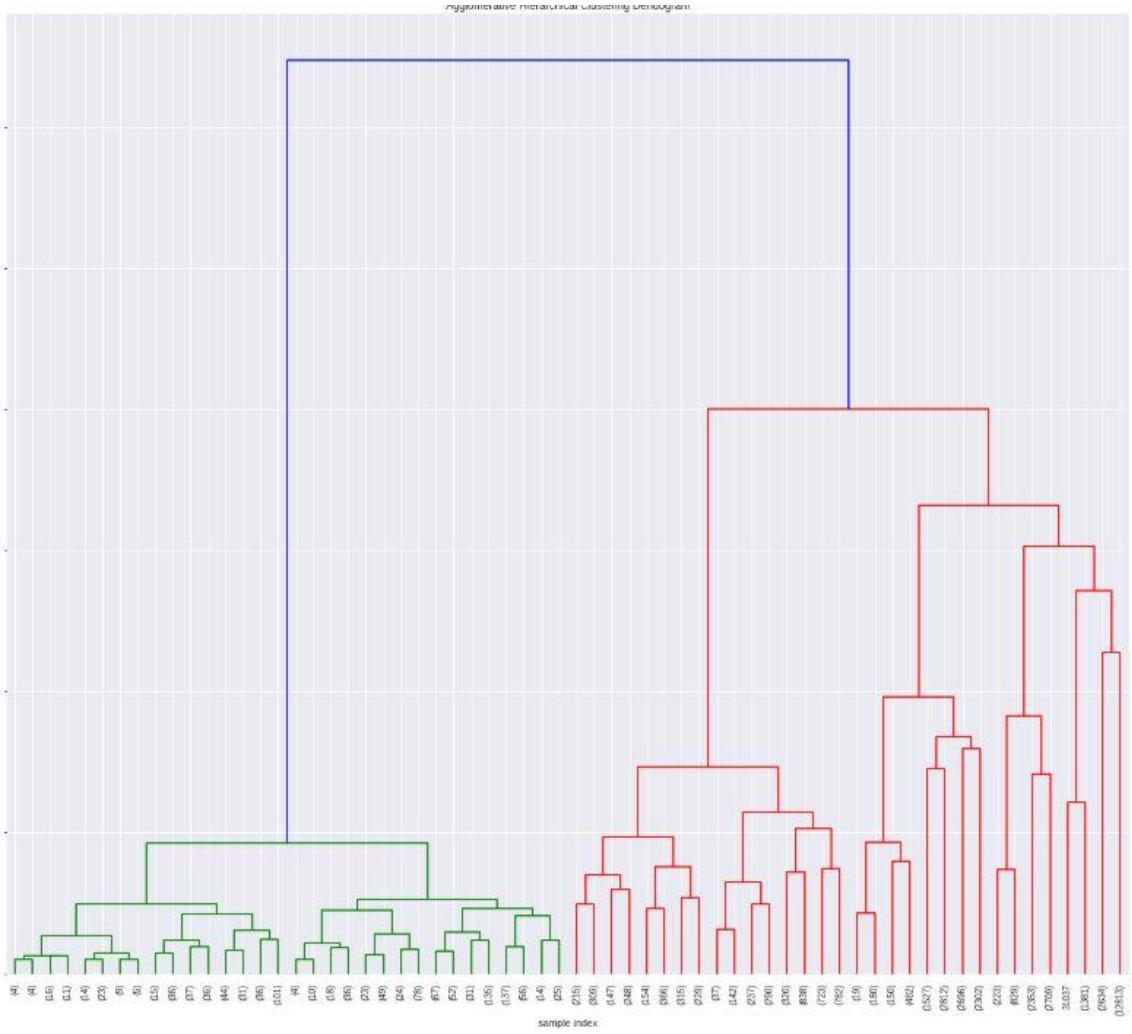
Silhouette score:

Using silhouette score, we got a much clearer picture that optimal number of clusters is 2 as there is significant dip in silhouette score for 3 clusters.

NUMBER OF CLUSTERS	SILHOUETTE SCORE
2	0.36962
3	0.08496
4	0.08481
5	0.06014
6	0.06830
7	0.07516
8	0.06960
9	0.07117
10	0.07537
11	0.08621
12	0.08978
13	0.09045
14	0.09122

Dendogram:

We also checked optimal number of clusters through hierarchical clustering approach where we built a dendogram using ward linkage. In dendogram also we got 2 as optimal number of clusters.



Feature Selection:

Feature selection is the process of selecting a subset of relevant attributes to be used in making the model in machine learning. Effective feature selection eliminates redundant variables and keeps only the best subset of predictors in the model which also gives shorter training times. Besides this, it avoids the curse of dimensionality and enhance generalization by reducing overfitting

Following Feature selection methods have been tried:

RFE:

- RFE method works by recursively removing attributes and building a model on those attributes that remain.
- It uses accuracy metric (here we use auc score) to rank the feature according to their importance.
- The RFE method takes the model to be used and the number of required features as input.
- It then gives the ranking of all the variables, 1 being most important.
- It also gives its support, True being relevant feature and False being irrelevant feature.

On applying RFE using RF as estimator, we get 47 features

Features:

Words	'n_tokens_content','n_unique_tokens', 'n_non_stop_unique_tokens', 'average_token_length'
Links and digital media	'num_hrefs', 'num_self_hrefs', 'num_imgs', 'self_reference_min_shares', 'self_reference_max_shares', 'self_reference_avg_sharess'
News Category	'data_channel_is_entertainment', 'data_channel_is_bus', 'data_channel_is_socmed', 'data_channel_is_tech', 'data_channel_is_world'
Publication Date	'weekday_is_saturday', 'weekday_is_sunday', 'is_weekend'
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'
NLP	, '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', 'title_subjectivity', 'title_sentiment_polarity', 'abs_title_sentiment_polarity'

Backward Elimination:
In backward elimination, we start with all the features and removes the least significant feature at each iteration which improves the performance of the model. We repeat this until no improvement is observed on removal of features

Applying Backward elimination, we got 39 features

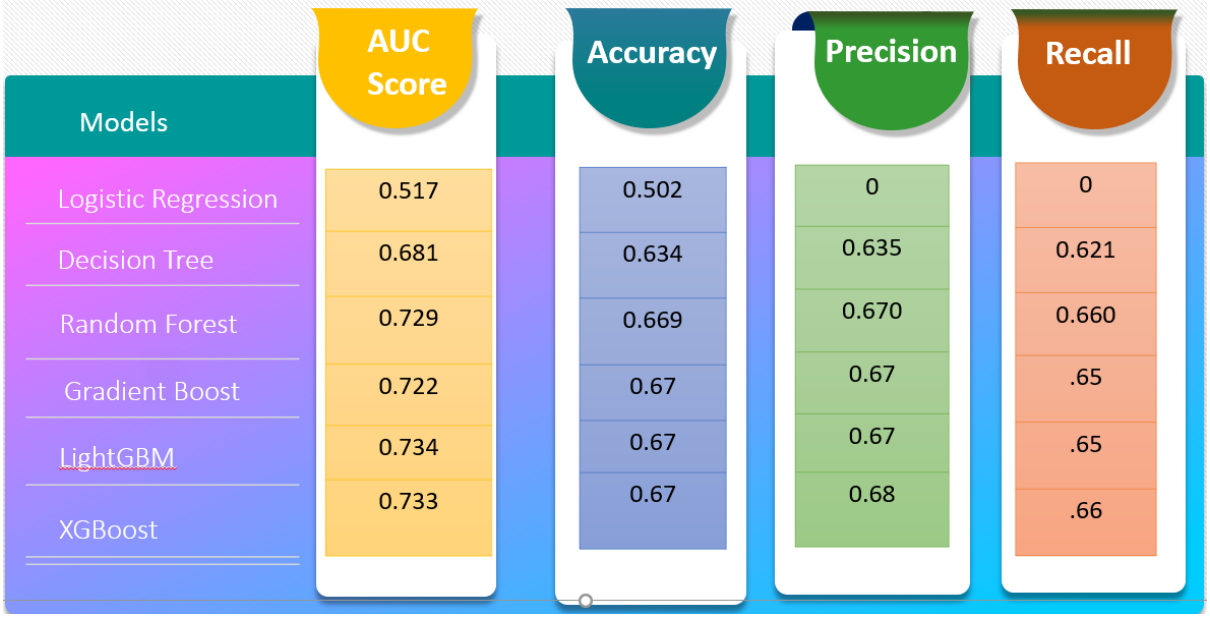
Features:

Words	'n_tokens_content'
Links and digital media	'num_hrefs', 'num_self_hrefs', 'num_imgs', 'num_videos', 'self_reference_max_shares', 'self_reference_avg_sharess',
News Category	'data_channel_is_entertainment', 'data_channel_is_bus', 'data_channel_is_socmed', 'data_channel_is_tech', 'data_channel_is_world
Publication Date	'weekday_is_monday', 'weekday_is_tuesday', 'weekday_is_wednesday', 'weekday_is_thursday', 'weekday_is_saturday', 'is_weekend'
Keywords	'kw_min_min', 'kw_max_min', 'kw_avg_min', 'kw_min_max', 'kw_max_max', 'kw_avg_max', 'kw_max_avg', 'kw_avg_avg',
NLP	'LDA_00', 'LDA_01', 'LDA_04', 'global_subjectivity', 'global_rate_positive_words', 'global_rate_negative_words', 'rate_positive_words', 'rate_negative_words', 'min_positive_polarity', 'title_subjectivity', 'title_sentiment_polarity', 'abs_title_subjectivity'

We went with backward elimination as RFE was taking all features whereas Backward elimination reduced feature count to 39 and the difference between score generated with RFE features and Backward elimination technique features was very less so in order to reduce complexity of the model we went with Backward elimination.

Final Model Building

We went ahead with Boosted Tree based models to improve our AUC score, precision & recall. After feature engineering ,LightGBM & XGBoost did the job for us as we can see below. The scores generated by the two models were close and variance was almost same with XGBoost giving slightly less variance error.



RECOMMENDATIONS

Below are some of the recommendations, publisher or author of article can use to increase the popularity of his/her article before publishing it. Tweaking the article as per the recommendation can help publishers categorise the articles to be published or not.

- Restrict article word length to 2500 words
- Number of embedded links should be in range 0-25
- Number of images should be in range 0-10. Lesser the better
- Number of videos should be between 0-5. Lesser the better
- References to older articles which have high popularity
- Publish articles on weekend
- Focusing more on Entertainment or Lifestyle articles can get you popularity

REFERENCES & BIBLIOGRAPHY

- A Proactive Intelligent Decision Support System for Predicting the Popularity of Online News Kelwin Fernandes1 , Pedro Vinagre2 , and Paulo Cortez2 1 INESC TEC Porto/Universidade do Porto, Portugal 2 ALGORITMI Research Centre, Universidade do Minho, Portugal
- K. Fernandes, P. Vinagre and P. Cortez. A Proactive Intelligent Decision Support System for Predicting the Popularity of Online News. Proceedings of the 17th EPIA 2015 - Portuguese Conference on Artificial Intelligence, September, Coimbra, Portugal.
- Is Your Story Going to Spread Like a Virus? Machine Learning Methods for News Popularity Prediction Xuandong Lei Xiaoti Hu Hongsheng Fang xuandong@stanford.edu xiaotihu@stanford.edu hsfang@stanford.edu