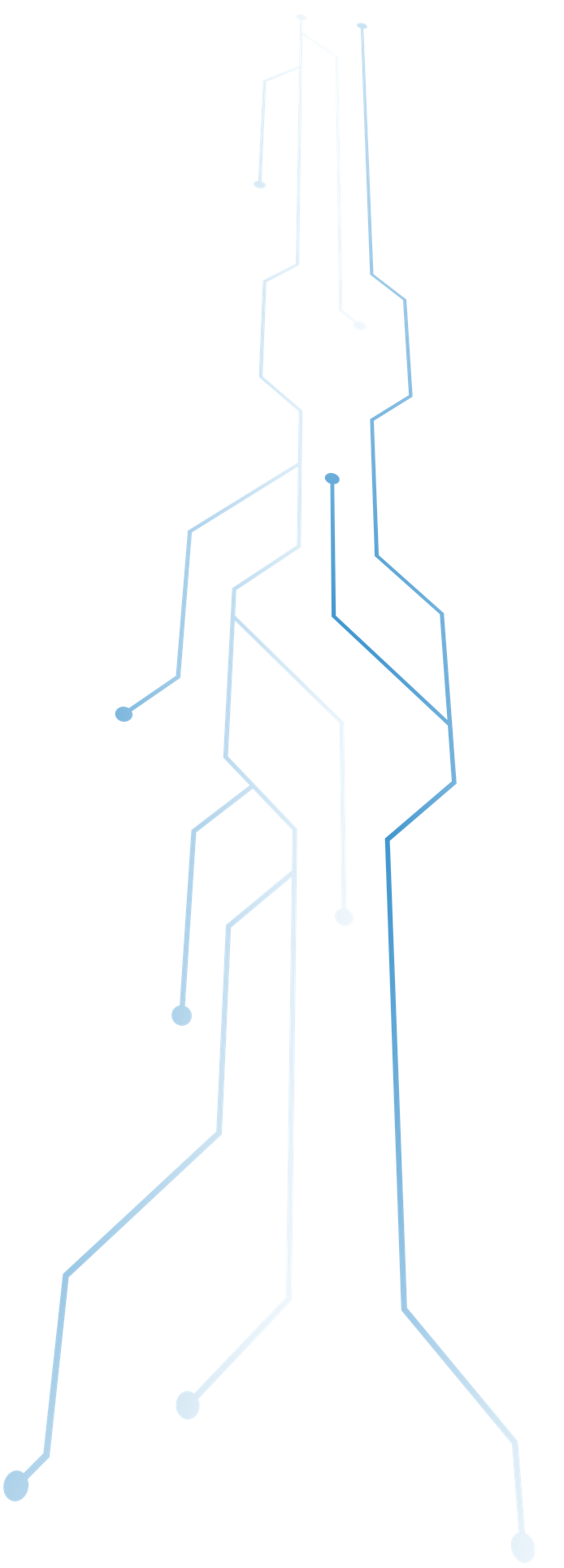
**PGPDSE-FT.G.NOV19 Batch**

|  |  |
| --- | --- |
| **Submitted By:** | **Mentored By:** |
| Sahil Pahuja | Mrs. Anjana Agrawal |
| Sanket Khadanga |  |
| Ankur Bisht |  |
| Ved Lad |  |
| Prem Shankar |  |
|  |  |



Capstone Project Report

Online News Popularity

|  |
| --- |
| **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 predict 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 count regression technique to Logistic Regression, Decision Tree and Random Forest. Their performances are recorded and compared. Feature selection methods are used to improve performance and reduce features. Random Forest turns out to be the best model for prediction, and it can achieve an auc score of 72.9924% with optimal parameters. Our work can help online news companies to predict news popularity before publication.

**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: 05-03-2020

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

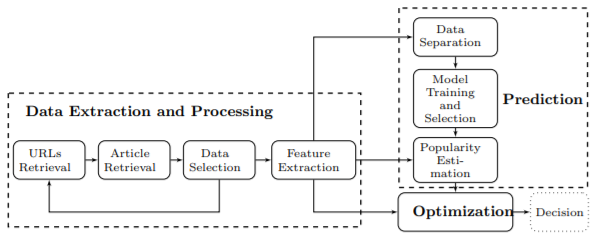


Fig: workflow architecture

|  |
| --- |
| **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.

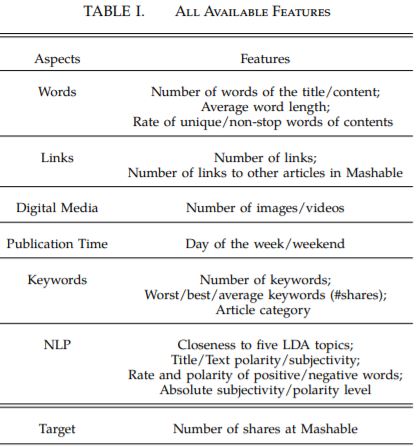
*Fig: Class Balance*

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:**

**Features:**

****

**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:**

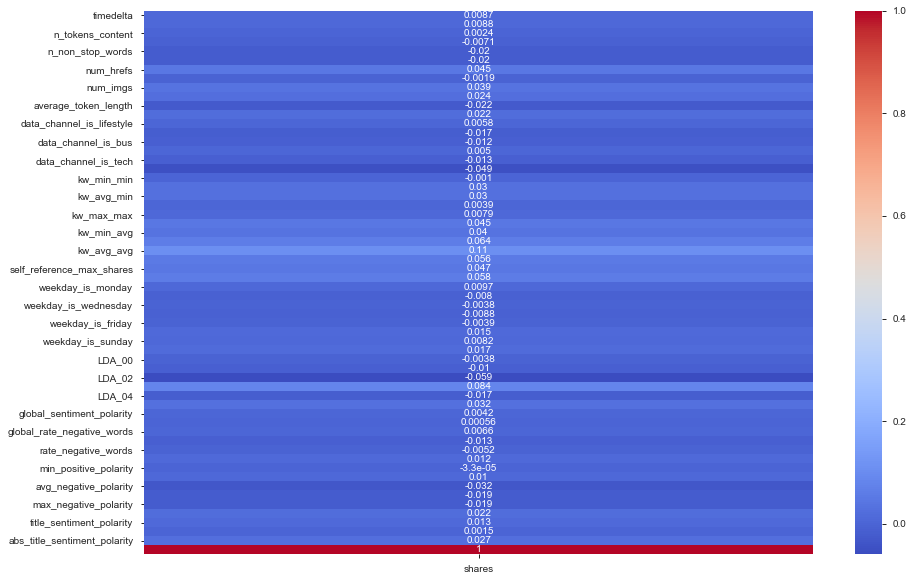
Most of the datasets on UCI are cleaned and same is with our dataset. There are no missing values.

|  |  |  |
| --- | --- | --- |
| 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.

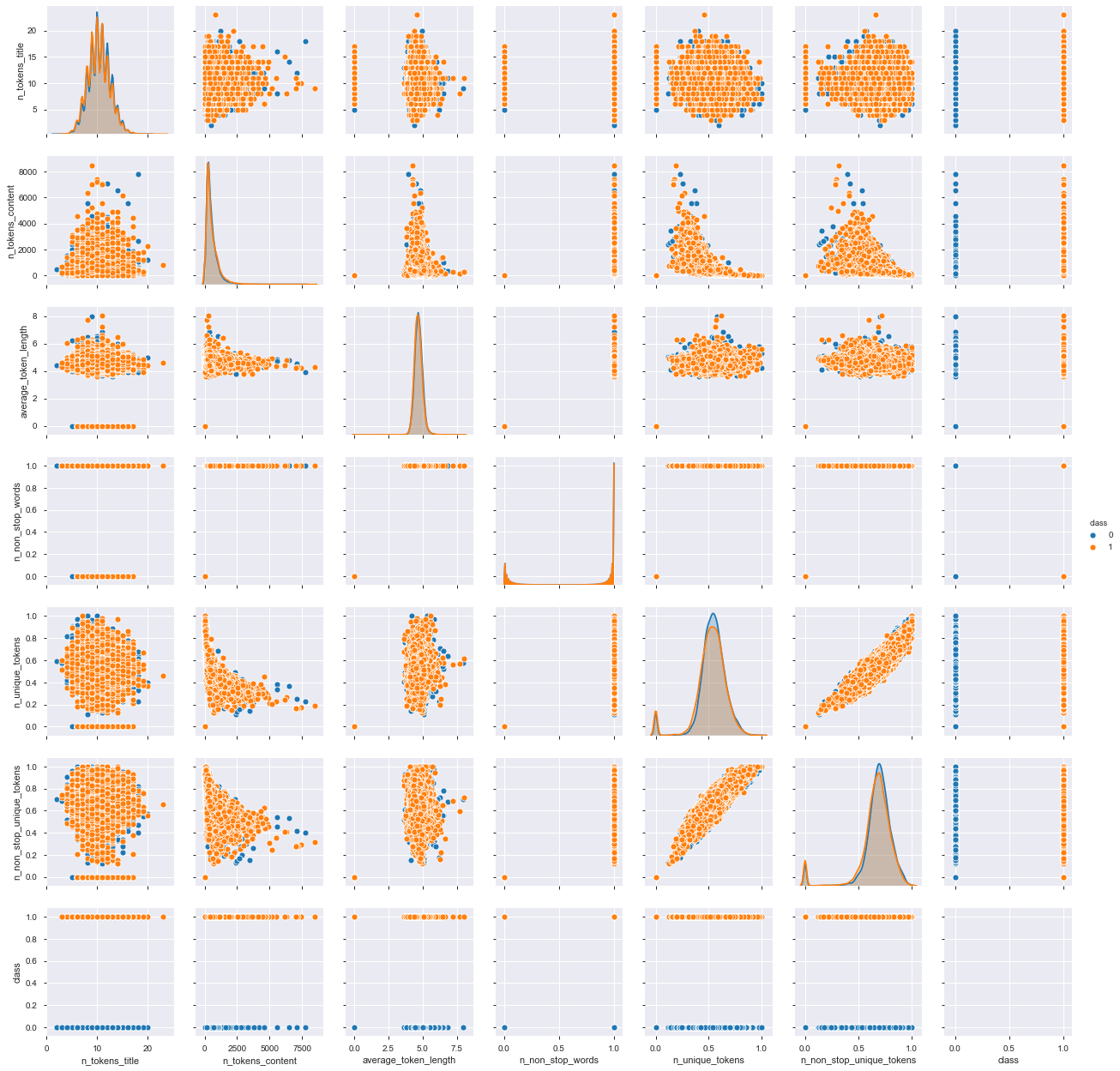
First, let’s look at correlation plot to get idea about how features are related to target



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

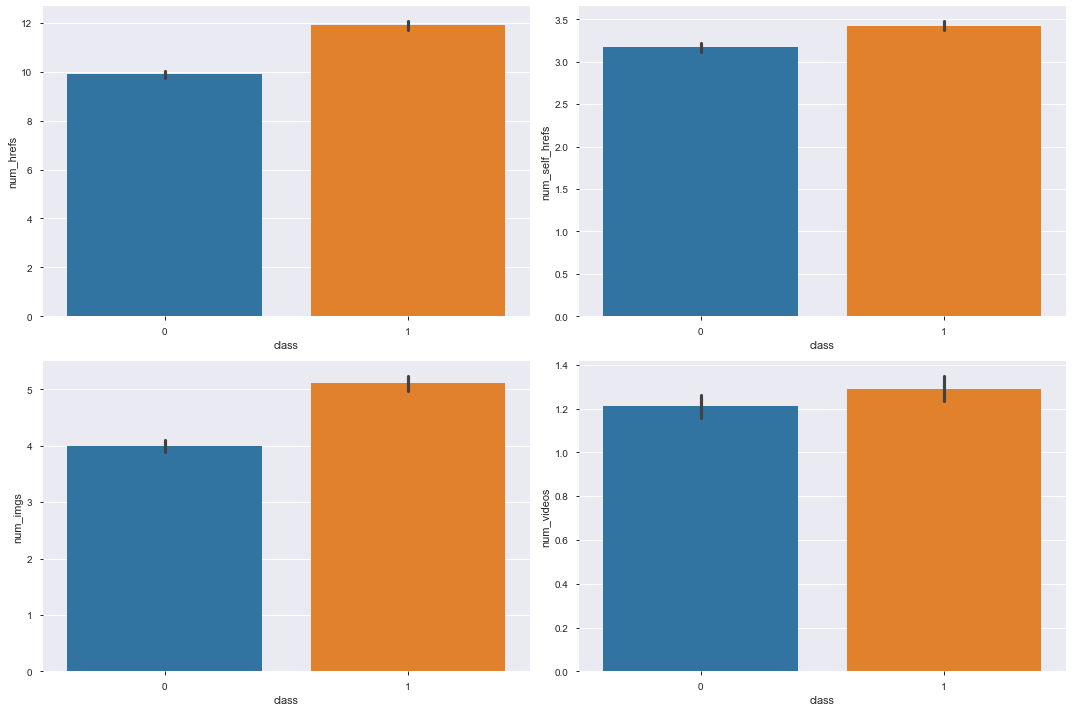
**Words**

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.

****

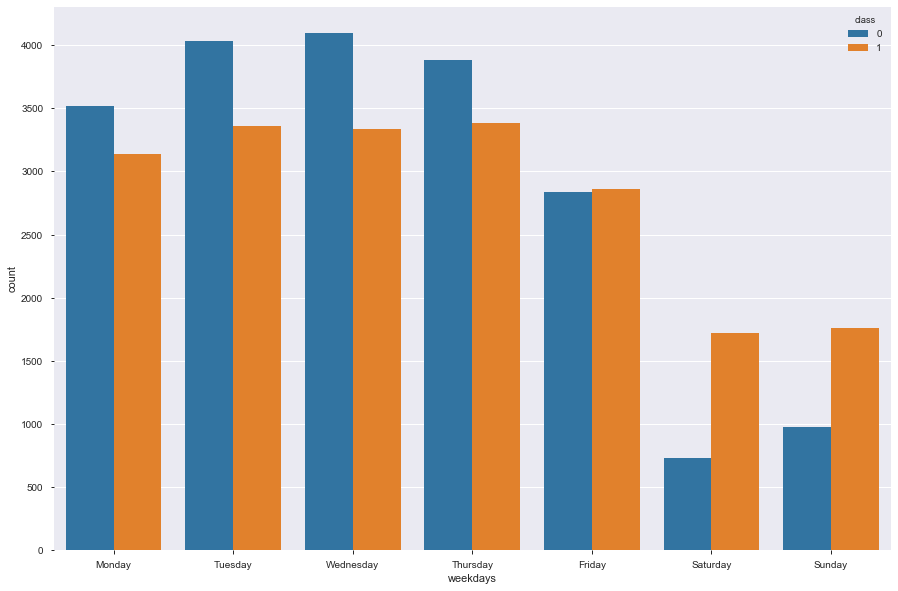
**Links & Digital Media:**

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**:

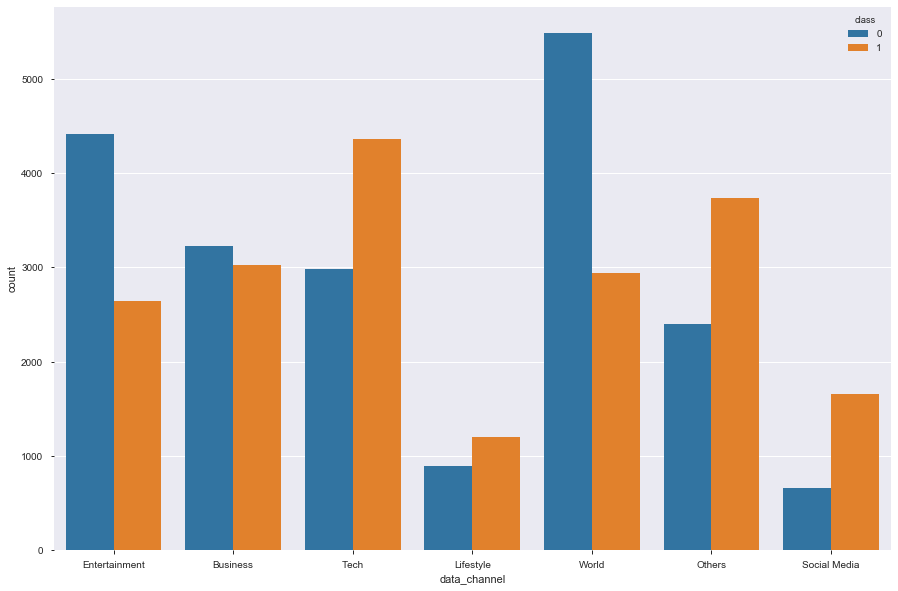
We can see here % of popular articles out of total number of articles published on a day is higher if article is published on Saturday or Sunday. There are high chances that article become popular if published on weekend.



**News Category**

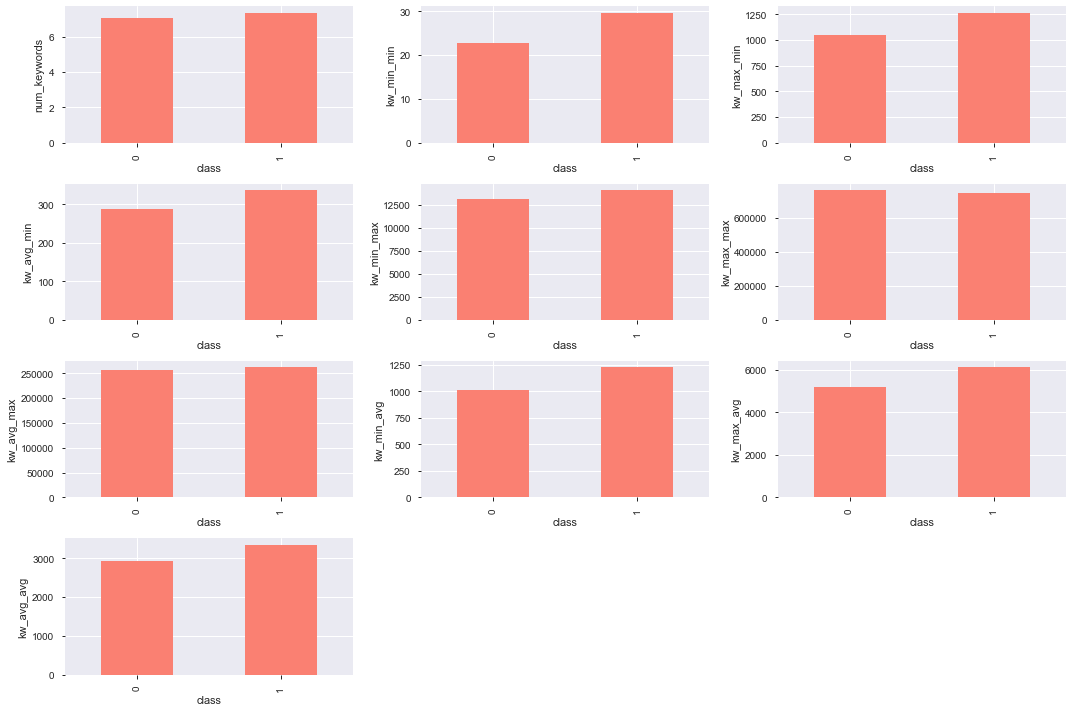
We can see here news article belonging to technology, social media and lifestyle has higher chance of becoming popular than other categories.

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.

****

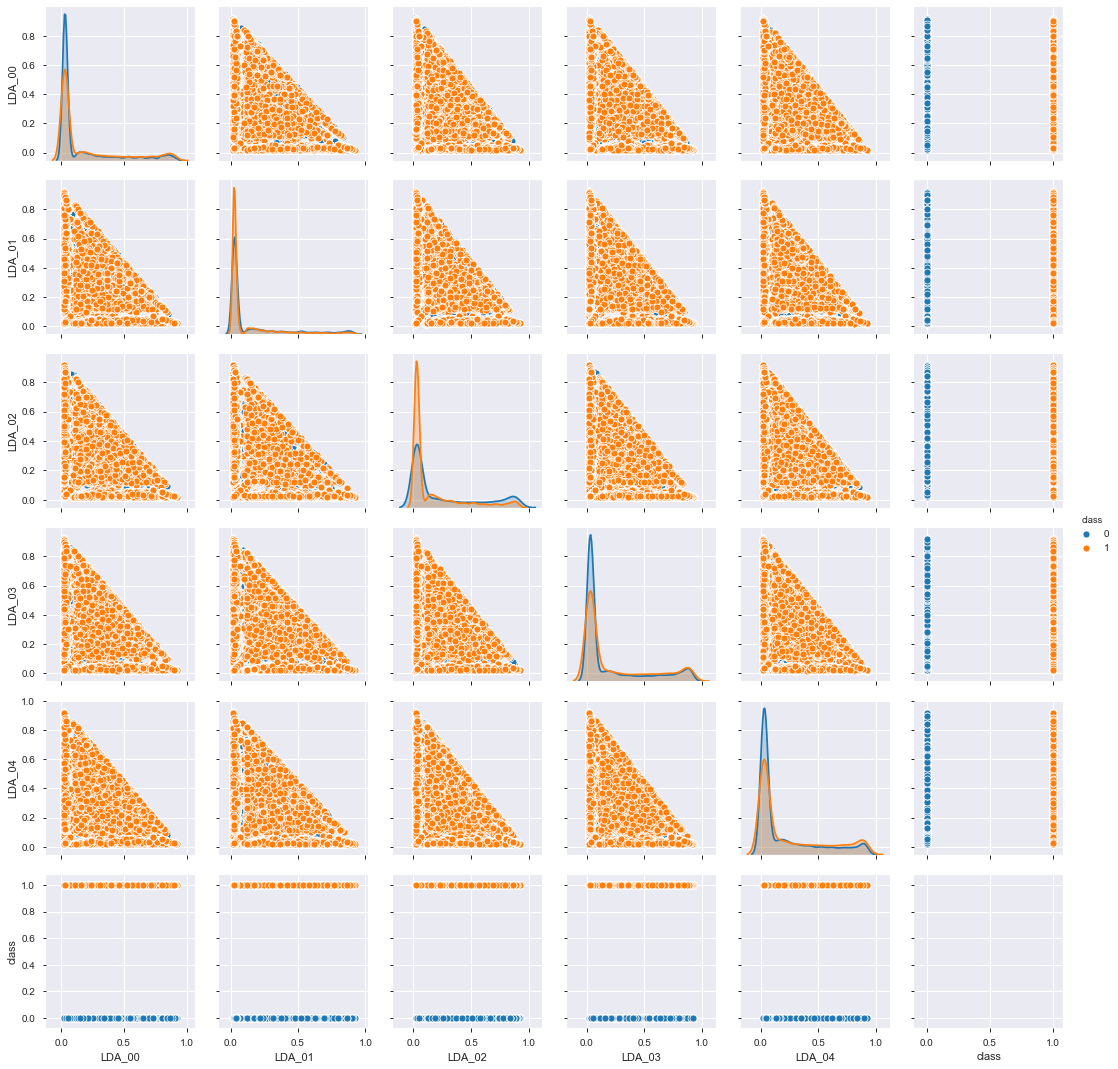
**Keywords**

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

****

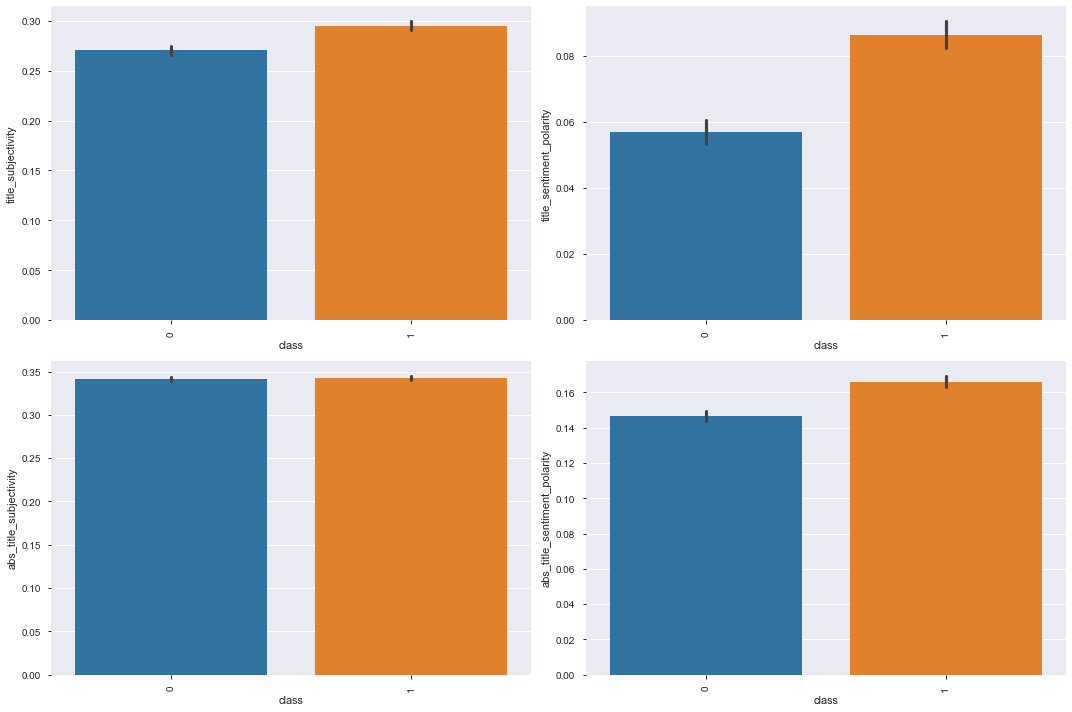
**Closeness to LDA topic:**

Distribution is pretty much similar for all features related to LDA\_topics.

****

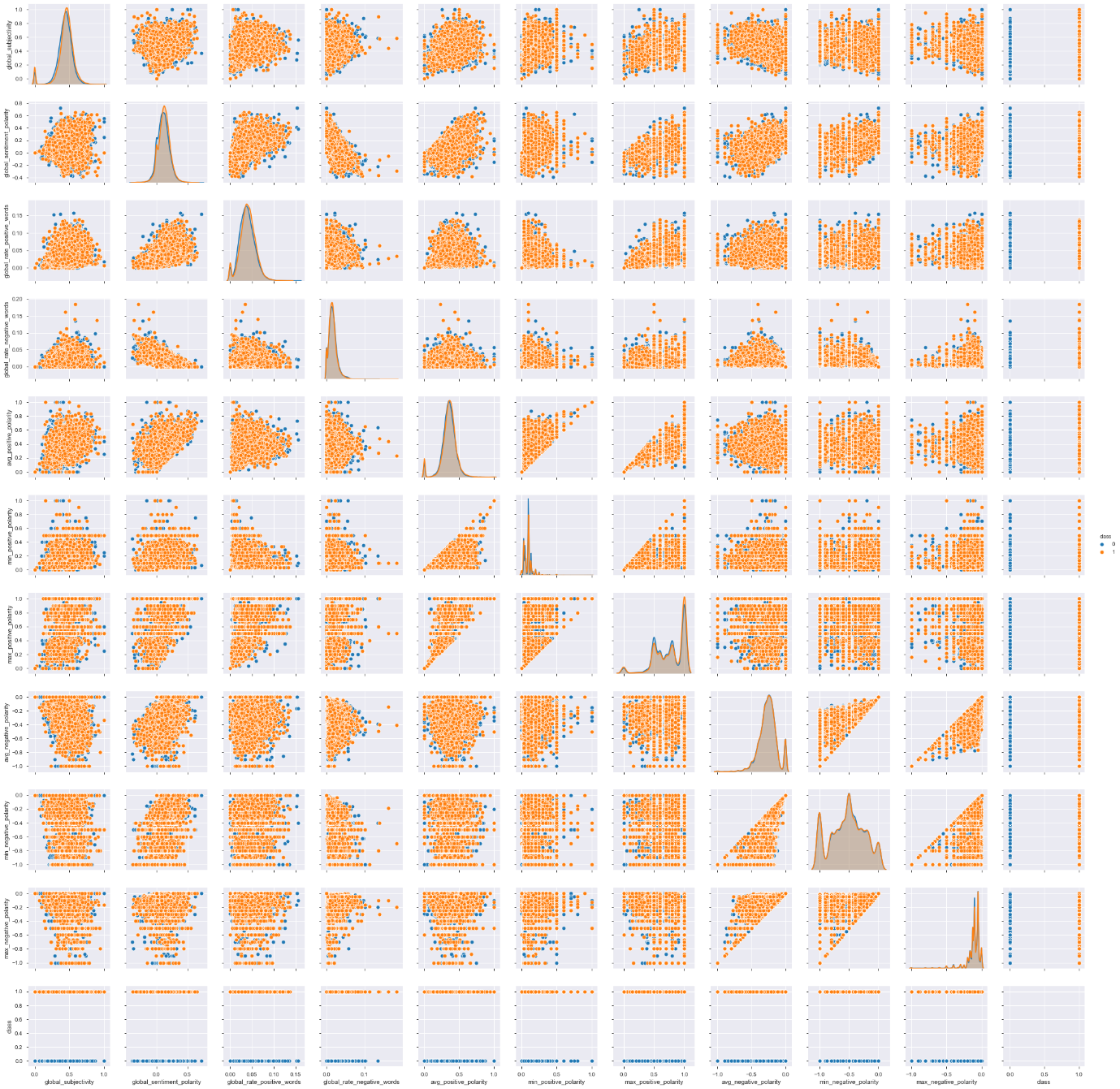
**Title Related features**:

Difference in Mean of title\_sentiment\_polarity for the 2 classes is high and moderate for abs\_title\_sentiment\_polarity



**Polarity and Subjectivity**

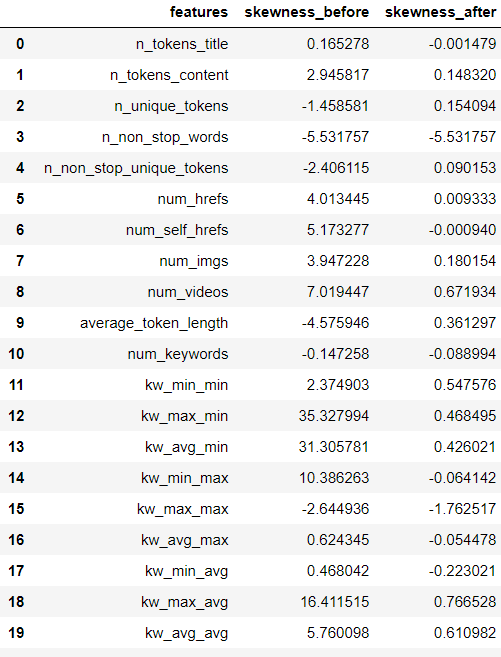
KDE plot is overlapping for both the classes in all the features so none of them seems significant for prediction.

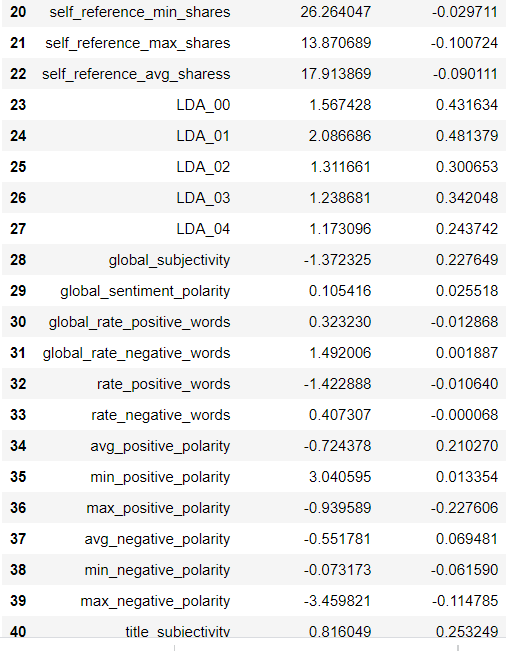
****

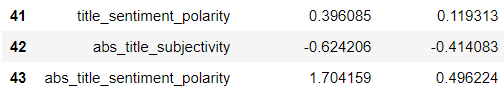
**Skewness Treatment**

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.







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 3 target variables as follows:

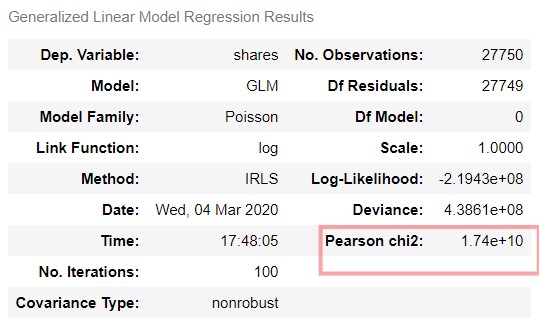
* Target variable is numerical discrete and perform count based regression
* Target variable is categorical with two classes and perform binary classification
* Target variable is categorical with multiple classes and perform multi class classification

**Count Based Regression:**

**Poisson Regression:**

Poisson regression is a form of regression analysis used to model discrete data. It is appropriate when the conditional distributions of Y (count data) given the observed counts are expected to be Poisson distributions.

We performed Poisson regression by taking independent variable (Number of shares) as numerical discrete. Below is the summary of result:



Since the very assumption of Poisson Regression that mean of fitted model should be same variance of fitted model is not followed here ,if we check Pearson Chi2 coefficient , it should be close to 1 indicating mean is same as variance which is not the case here so we cannot apply Poisson Regression.

Hence, we go for negative binomial regression.

**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’.

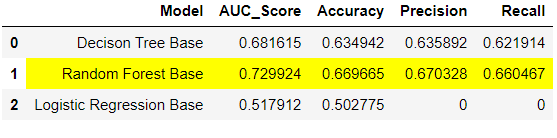
* AUC is **scale-invariant**. It measures how well predictions are ranked, rather than their absolute values.
* AUC is **classification-threshold-invariant**. It measures the quality of the model's predictions irrespective of what classification threshold is chosen.

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.

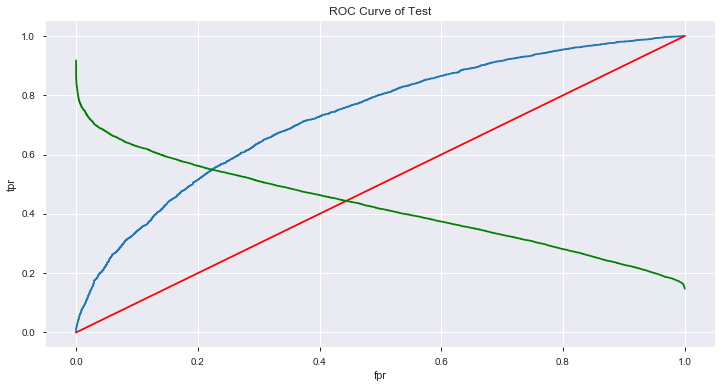
Below is the comparison of scores for the above three models



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.

-Threshold

- Worst Line

****

ROC curve for Random Forest

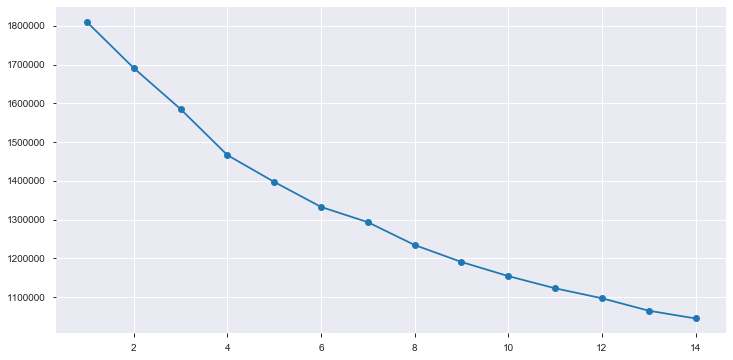
**Multi Class Classification:**

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.



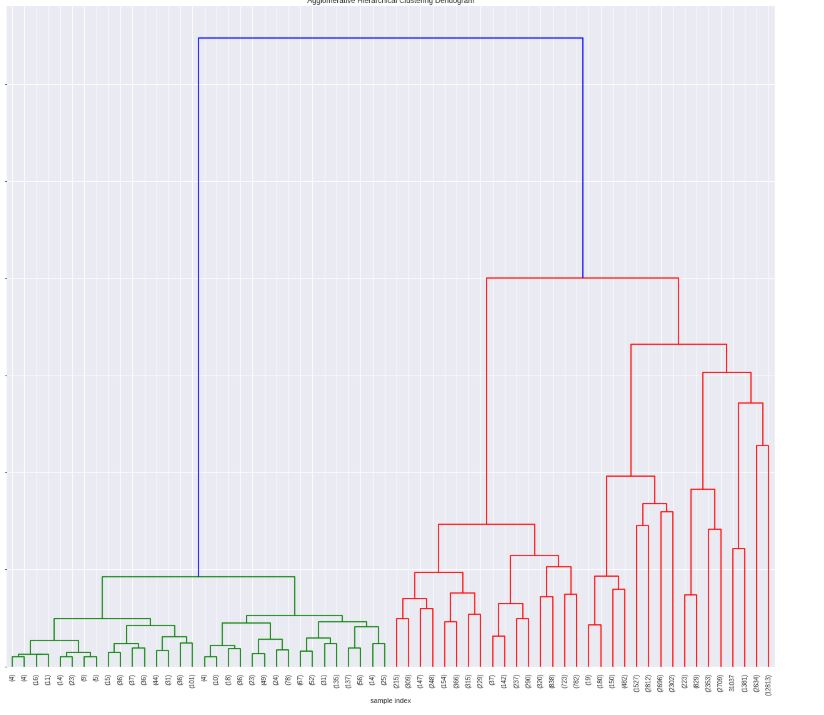
**Silhouette score:**

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' |

|  |
| --- |
| **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](mailto:hsfang@stanford.edu)