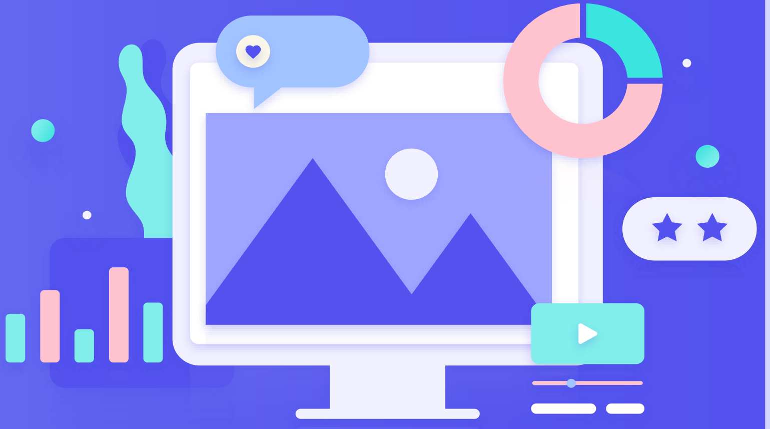
Online News Popularity dataset from the Mashable website

Python for Data Analysis A4 Project
Eva Padrino
Victor Quesnay

Table of contents

1. Introduction: What is our dataset about?

Our dataset summarizes a set of 61 features describing articles published on the Mashable wesbite. Mashable is an international entertainment, culture, technology, science, social digital media platform, news website, multi-platform media and entertainment company. The objective with these features is to predict the popularity of an article published on the Mashable website caracterized by the number of shares of an article. Every row of our dataset corresponds to an article and its features.



1. Introduction : Description of the features and the output.

Features

- 0. url: URL of the article (non-predictive)
- 1. timedelta: Days between the article publication and the dataset acquisition (non-predictive)
- 2. n_tokens_title: Number of words in the title
- 3. n_tokens_content: Number of words in the content
- 4. n_unique_tokens: Rate of unique words in the content
- 5. n_non_stop_words: Rate of non-stop words in the content
- 6. n_non_stop_unique_tokens: Rate of unique non-stop words in the content
- 7. num_hrefs: Number of links
- 8. num_self_hrefs: Number of links to other articles published by Mashable
- 9. num_imgs: Number of images
- 10. num_videos: Number of videos
- 11. average_token_length: Average length of the words in the content
- 12. num_keywords: Number of keywords in the metadata
- 13. data_channel_is_lifestyle: Is data channel 'Lifestyle'? (meaning the subject of the article)

1. Introduction : Description of the features and the output.

Features

- 14. data_channel_is_entertainment: Is data channel 'Entertainment'?
- 15. data_channel_is_bus: Is data channel 'Business'?
- 16. data_channel_is_socmed: Is data channel 'Social Media'?
- 17. data_channel_is_tech: Is data channel 'Tech'?
- 18. data_channel_is_world: Is data channel 'World'?
- 19. kw_min_min: Worst keyword (min. shares)
- 20. kw_max_min: Worst keyword (max. shares)
- 21. kw_avg_min: Worst keyword (avg. shares)
- 22. kw_min_max: Best keyword (min. shares)
- 23. kw_max_max: Best keyword (max. shares)
- 24. kw_avg_max: Best keyword (avg. shares)
- 25. kw_min_avg: Avg. keyword (min. shares)
- 26. kw_max_avg: Avg. keyword (max. shares)
- 27. kw_avg_avg: Avg. keyword (avg. shares)
- 28. self_reference_min_shares: Min. shares of referenced articles in Mashable, when the current article references other Mashable's articles we look at the shares of the articles, this feature is for the minimum found.

1. Introduction : Description of the features and the output.

Features

- 29. self_reference_max_shares: Max. shares of referenced articles in Mashable, when the current article references other Mashable's articles we look at the shares of the articles, this feature is for the maximum found.
- 30. self_reference_avg_sharess: Avg. shares of referenced articles in Mashable, when the current article references other Mashable's articles we look at the shares of the articles, this feature is for the average found.
- 31. weekday_is_monday: Was the article published on a Monday?
- 32. weekday_is_tuesday: Was the article published on a Tuesday?
- 33. weekday_is_wednesday: Was the article published on a Wednesday?
- 34. weekday_is_thursday: Was the article published on a Thursday?
- 35. weekday_is_friday: Was the article published on a Friday?
- 36. weekday_is_saturday: Was the article published on a Saturday?
- 37. weekday_is_sunday: Was the article published on a Sunday?
- 38. is_weekend: Was the article published on the weekend?

1. Introduction : Description of the features and the output.

Features

- 39. LDA_00: Closeness to LDA topic 0
- 40. LDA_01: Closeness to LDA topic 1
- 41. LDA_02: Closeness to LDA topic 2
- 42. LDA_03: Closeness to LDA topic 3
- 43. LDA_04: Closeness to LDA topic 4
- 44. global_subjectivity: Text subjectivity
- 45. global_sentiment_polarity: Text sentiment polarity
- 46. global_rate_positive_words: Rate of positive words in the content
- 47. global_rate_negative_words: Rate of negative words in the content
- 48. rate_positive_words: Rate of positive words among non-neutral tokens
- 49. rate_negative_words: Rate of negative words among non-neutral tokens
- 50. avg_positive_polarity: Avg. polarity of positive words
- 51. min_positive_polarity: Min. polarity of positive words
- 52. max_positive_polarity: Max. polarity of positive words
- 53. avg_negative_polarity: Avg. polarity of negative words

1. Introduction : Description of the features and the output.

Features

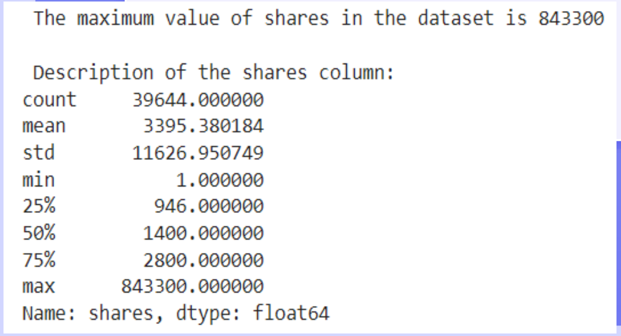
- 57. title_sentiment_polarity: Title polarity
- 58. abs_title_subjectivity: Absolute subjectivity level
- 59. abs_title_sentiment_polarity: Absolute polarity level

Output

- Shares : number of times the article was shared.

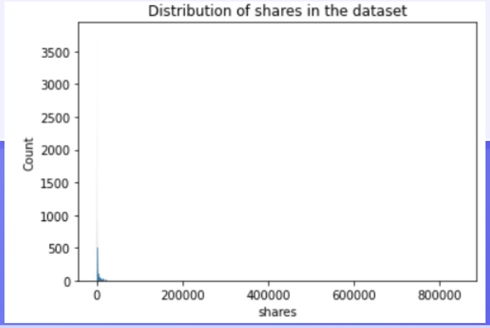
1. Introduction: What kind of problem are we confronted to?

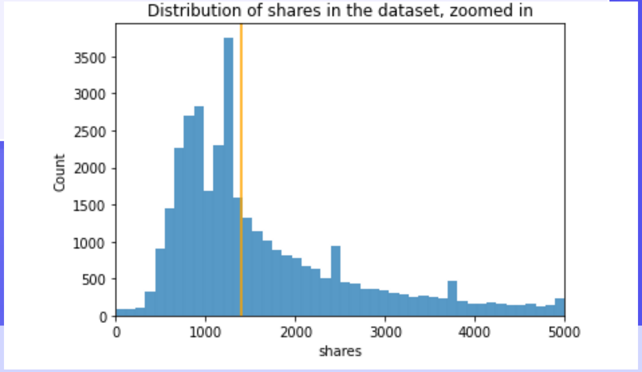
- The values of the column « shares » are numerical and represent the number of times an article was shared.
- The problem could be adressed as a regression problem keeping the numerical values but we decided to adress it as a classification problem by creating classes.
- How many classes did we decide to create? Since that in the dataset there were a few articles that had an extremely high number of shares compared to the others, we went with only 2 classes : « Popular » and « Not Popular ». How did we make the separation? We consider articles with a number of shares below the median (1400) « Not Popular » and the rest « Popular ».
- As we can see on the images below, most of articles shares of the dataset are below 2800 shares (3rd quantile), that is why we did not take the mean (or another measure) because it was very influenced by extreme values and was well above the 3rd quantile.



```
 The maximum value of shares in the dataset is 843300

 Description of the shares column:
count     39644.000000
mean       3395.380184
std       11626.950749
min           1.000000
25%         946.000000
50%        1400.000000
75%        2800.000000
max      843300.000000
Name: shares, dtype: float64
```





2. Cleaning and modifying the dataset

- Importing the dataset, we got rid of unwanted spaces.
- Some columns that were containing integers numbers were marked as 'float' type, we changed them to 'int' type.
- We got rid of the "url" and "time-delta" features since they were not predictive.
- We checked for duplicates but there were none.
- We looked for NaN values but there were none.
- We merged the columns 'weekday_is_Monday','weekday_is_Tuesday','weekday_is_wednesday','weekday_is_Thursday','weekday_is_Friday'weekday_is_Saturday,'weekday_is_Sunday' into one column called 'weekday' . We removed the column 'is_weekend' because we found that it was redundant to have columns for Saturday and Sunday and this one.   In the column 'weekday' we kept the words describing the days in a dataframe made for visualization ("dataVisu") and encoded the days in another dataframe ('dataExpl') for using machine learning models.
- We merged the columns 'data_channel_is_lifestyle','data_channel_is_entertainment','data_channel_is_bus' , 'data_channel_is_socmed','data_channel_is_tech' ,'data_channel_is_world' into one column called 'channel'. We kept the names of the channel in a dataframe that we were going to use for graphs and encoded them for using machine learning models.

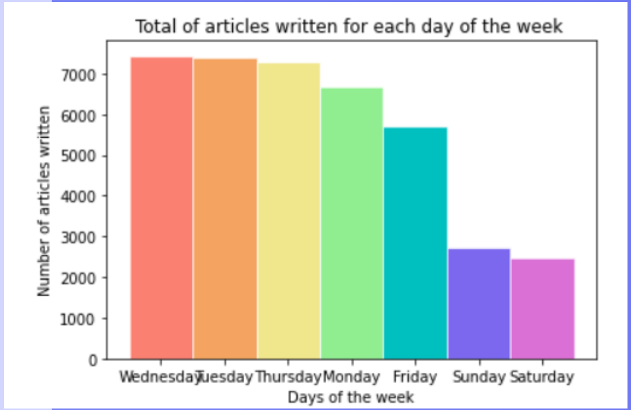
2. Cleaning and modifying the dataset

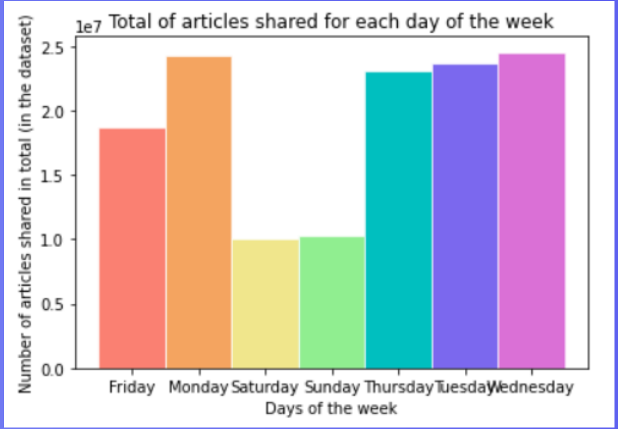
- Encoding variables and converting the columns to type ‘category’. At first we encoded categorical variables such as weekday, channel and shares with the cat.codes method and converted these columns to type ‘category’ with astype(‘category’). Our purpose was to reduce the time of algorithms to fit a model to our data. We also scaled all the features that were not categories. Unfortunately, when fitting models and plotting this new dataframe we encountered problems (because the of type category). We ended up, encoding them ourselves and making the columns as type ‘int’. We still scaled the ”non-categorical“ columns.

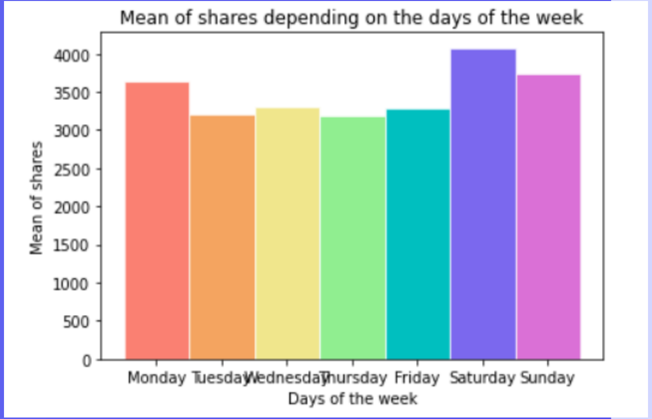
3. Data visualization

3. Data visualization : days of the week.

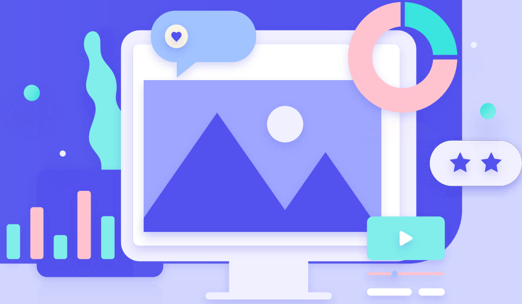
Visualizing how many articles are published in total for each day of the week and comparing how many shares articles are getting in total for each day of the week. Also visualizing the mean of shares for each day of the week.







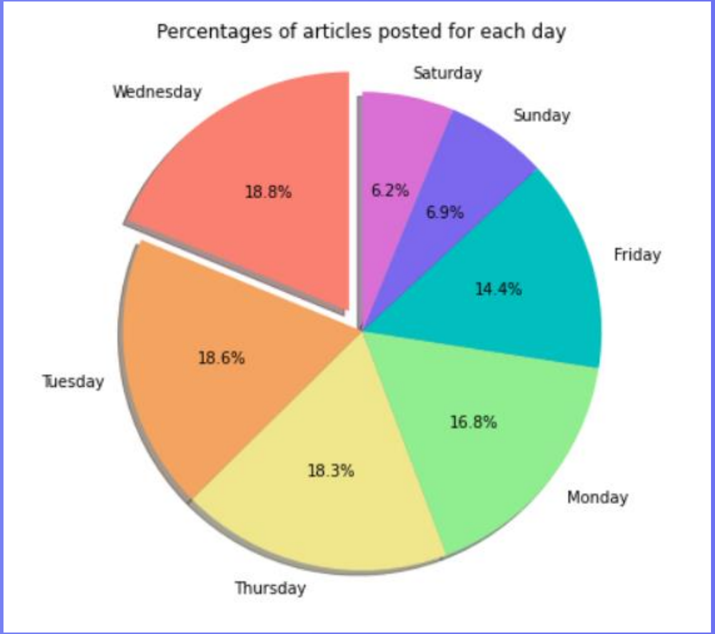
In our dataset, we saw that most articles were published during the week. The cumulation of shares for each day showed that there were in total more shares during the week but it is normal since there are more articles published during the week than the weekend. By plotting the mean, it showed us that indeed there were no big difference in shares between all the days (slightly better for the weekend).

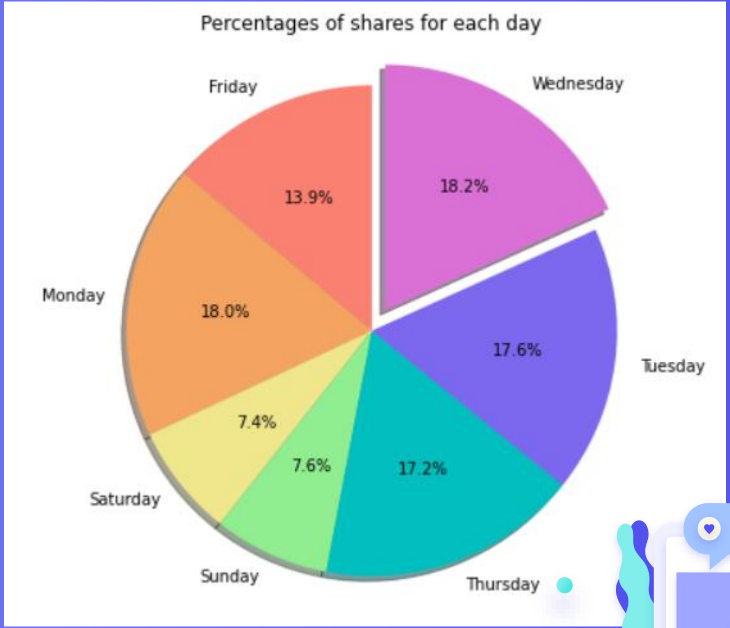


3. Data visualization : days of the week.

Visualizing how many articles are published in total for each day of the week and comparing how many shares articles are getting in total for each day of the week.

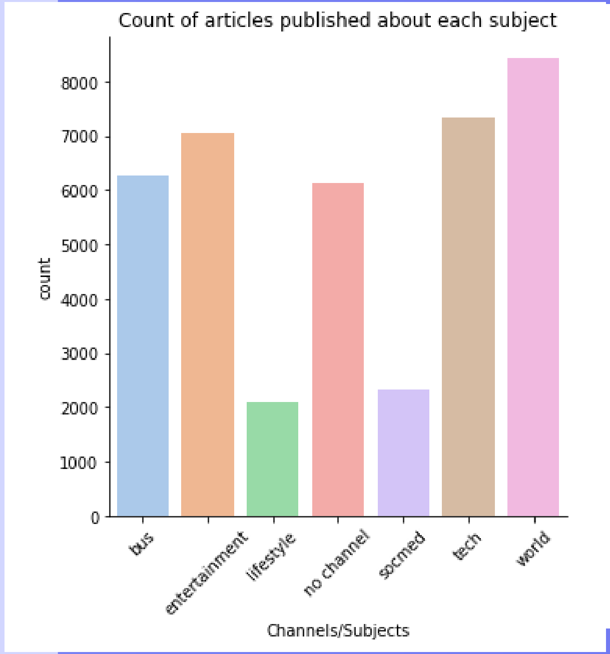
Same data but visualized with pie charts.

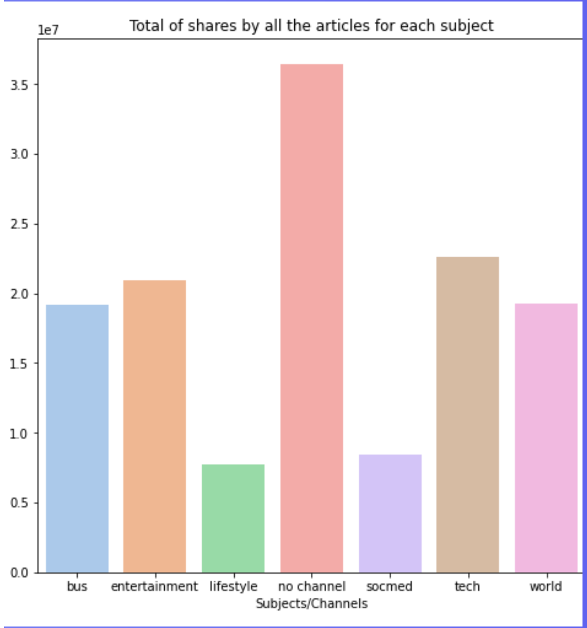


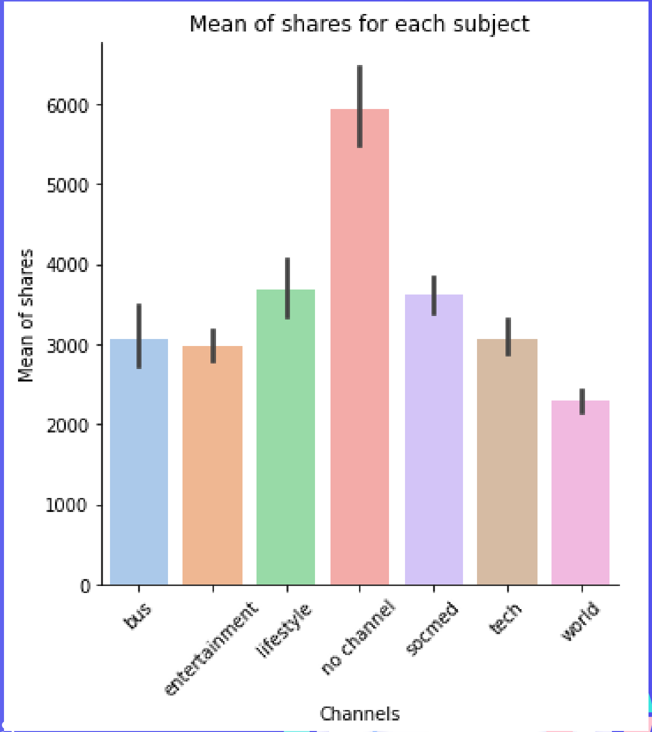


3. Data visualization : Subjects of the articles

Visualizing how many articles were published in total for each subject and what subjects got more shares in total. Also seeing the mean.





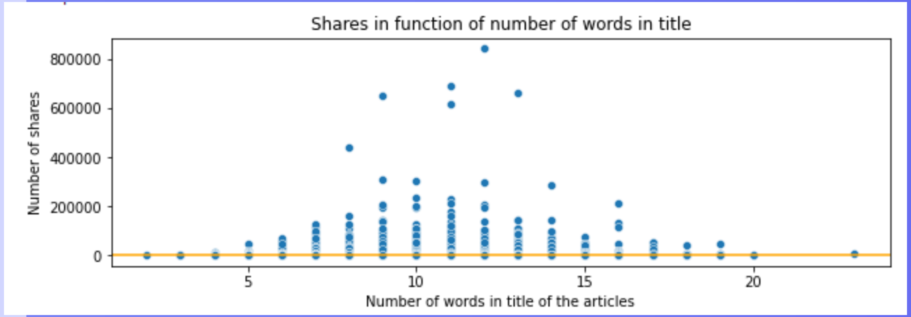


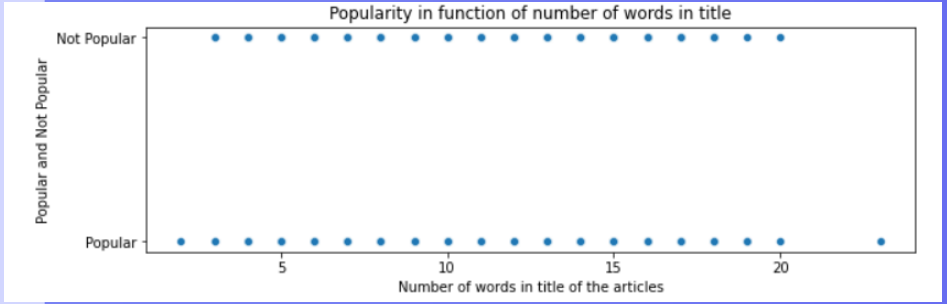
Most published articles are about the 'world' subject. When looking at the shares, articles with no channel specified have the most shares therefore we cannot know what these articles were about. When looking at the mean for all the articles that we know of the subject, we can see that the means are very close. In all cases, world is the most written about but not the most shared.

3. Data visualization : Content of the articles

Number of words in the title of the article

Visualizing the distribution of shares of articles depending on Popularity





At first glance, considering our problem a classification problem of 2 classes separated by the median value, the number of words in the title does not seem to make a difference in popularity. Popular and Not popular are distributed all along the lines. If we considered our problem a regression problem, articles with extreme number of shares have titles with a number of words between 7 and 15 and it would have an impact for these articles.

Since there are too much articles/rows in our dataset, we cannot just visualize this information with scatterplots.

Mean of number of words in title for Popular articles: 10.312 words

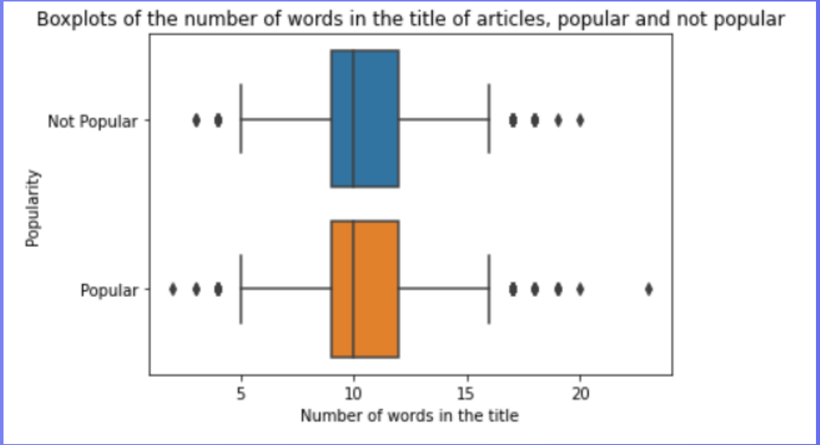
Mean of number of words in title for Not Popular articles: 10.5 words

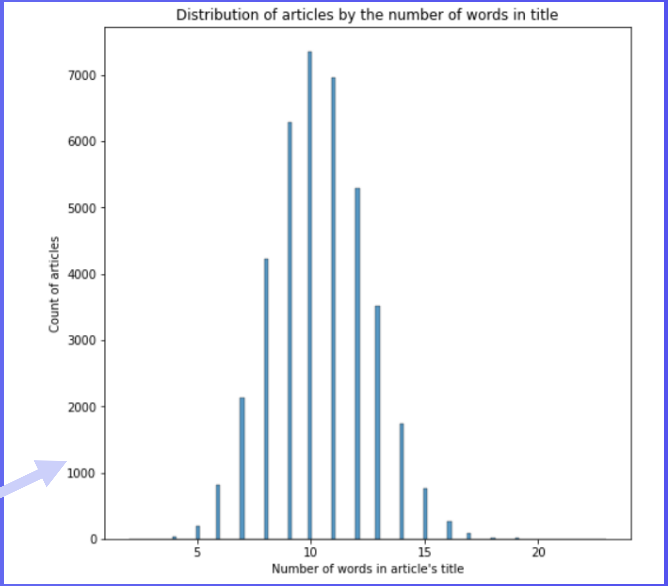
For Popular and not Popular articles the mean of number of words in the title are of 10 words basically (the same for each one).

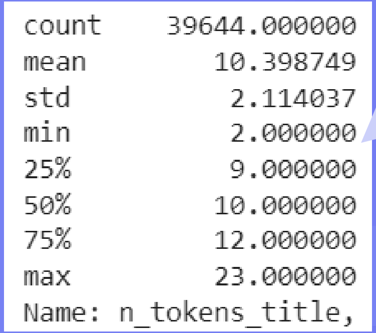
3. Data visualization : Content of the articles

Number of words in the title of the article

Here we can have a better understanding of the information than just with scatterplots.







```
count    39644.000000
mean        10.398749
std          2.114037
min          2.000000
25%          9.000000
50%         10.000000
75%         12.000000
max         23.000000
Name: n_tokens_title,
```

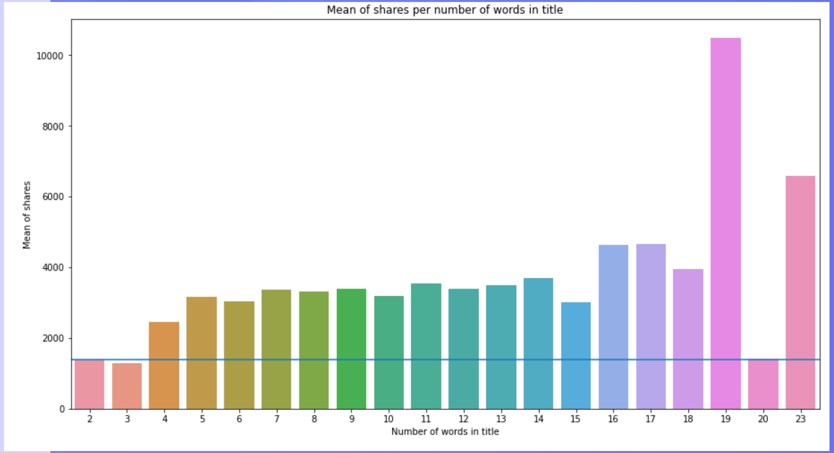
For all the dataset

Most articles, have a title of 10 words as we can see in the distribution

And in the boxplots graphs, we can confirm that with our 2 classes it does not make a difference.

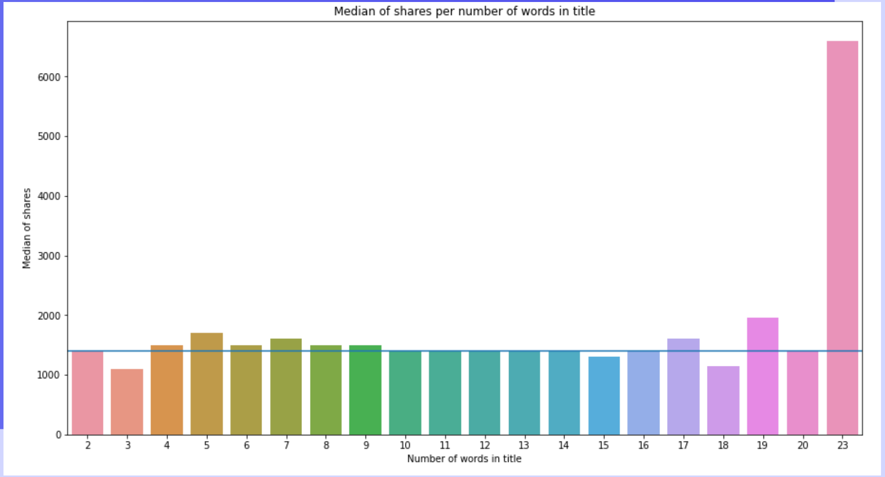
3. Data visualization : Content of the articles

Number of words in the title of the article



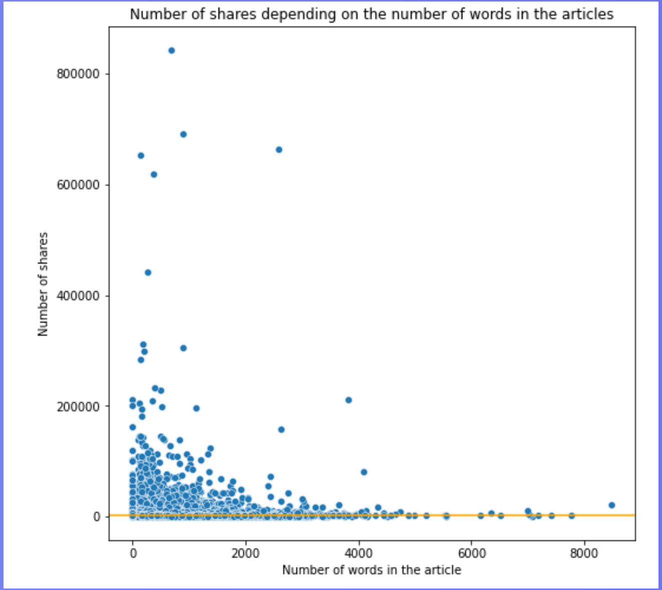
These 2 graphs confirm what we have said earlier (with the median horizontal line) : for a classification problem where we have 2 different categories, the categories englobe such a large amount of lines and the number of words in the title do not really have an influence on them but have an influence on the real number of shares. The reality is that by taking the mean, articles with a large number of words in the title have a high number of shares

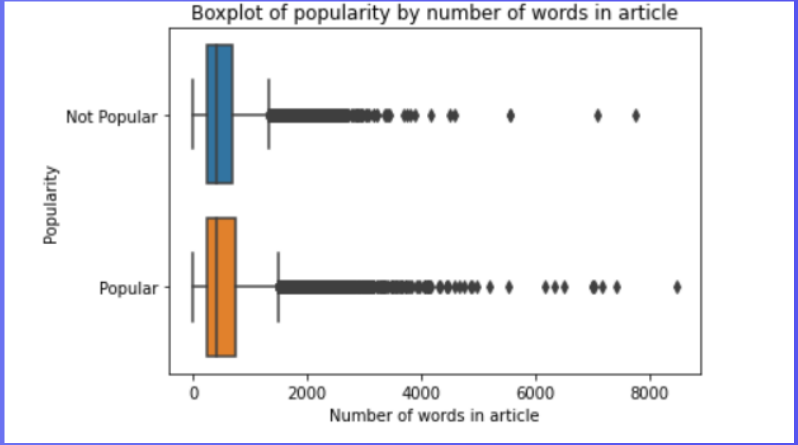
We wanted to see how many articles had 23 words in their title and we only found 1. Meaning that even for a regression problem, that article having such an extreme number of shares is not representative because we cannot tell without having other examples.

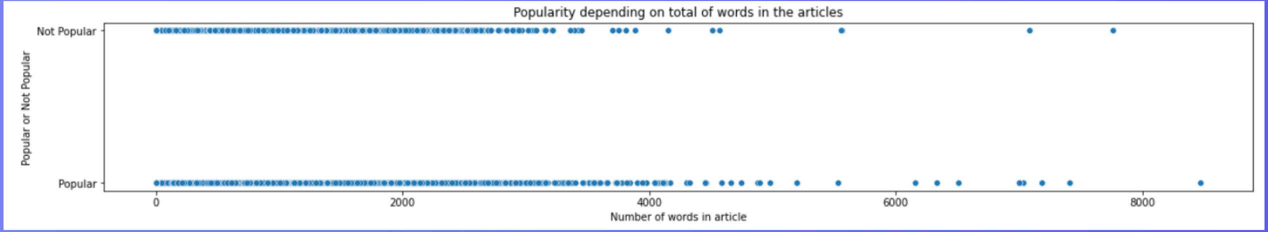


3. Data visualization : Content of the articles

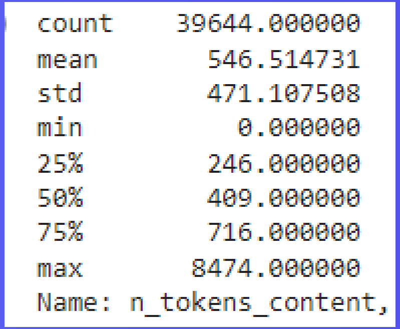
Number of words in the article







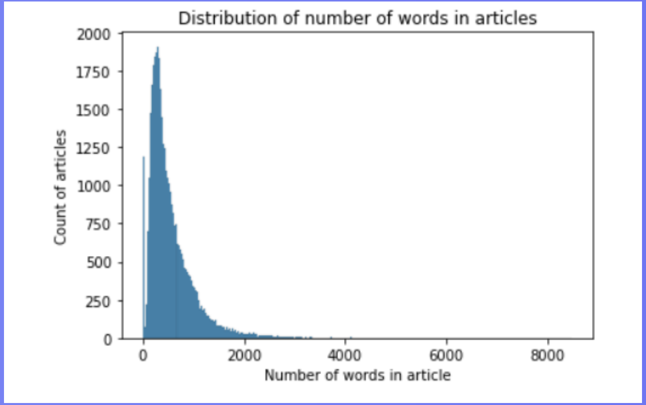
All articles

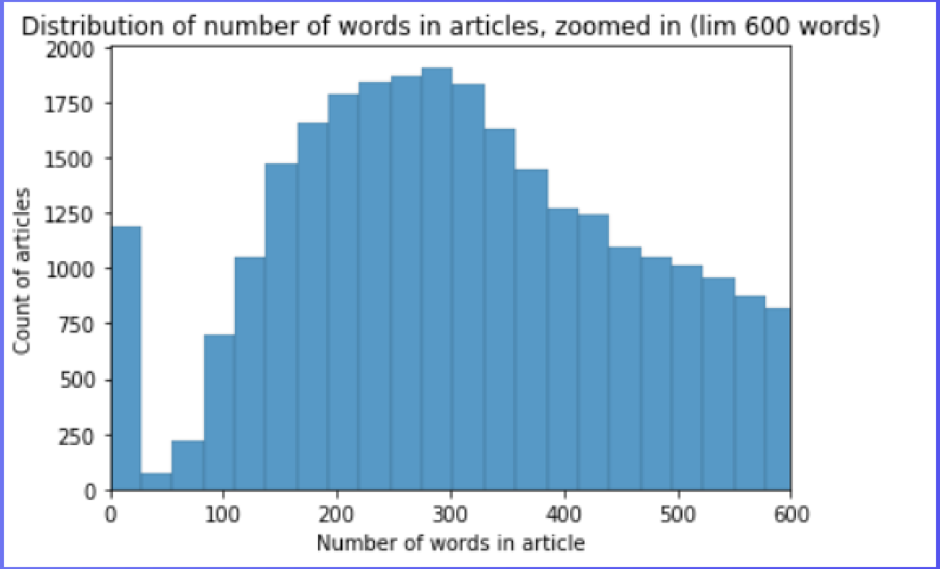


```
count     39644.000000
mean        546.514731
std         471.107508
min           0.000000
25%         246.000000
50%         409.000000
75%         716.000000
max        8474.000000
Name: n_tokens_content,
```

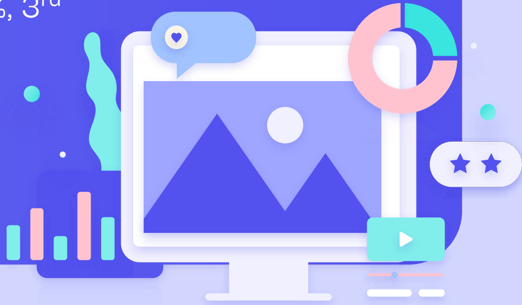
3. Data visualization : Content of the articles

Number of words in the article



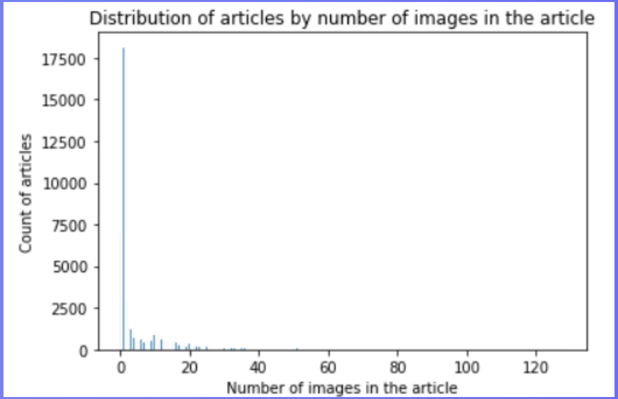


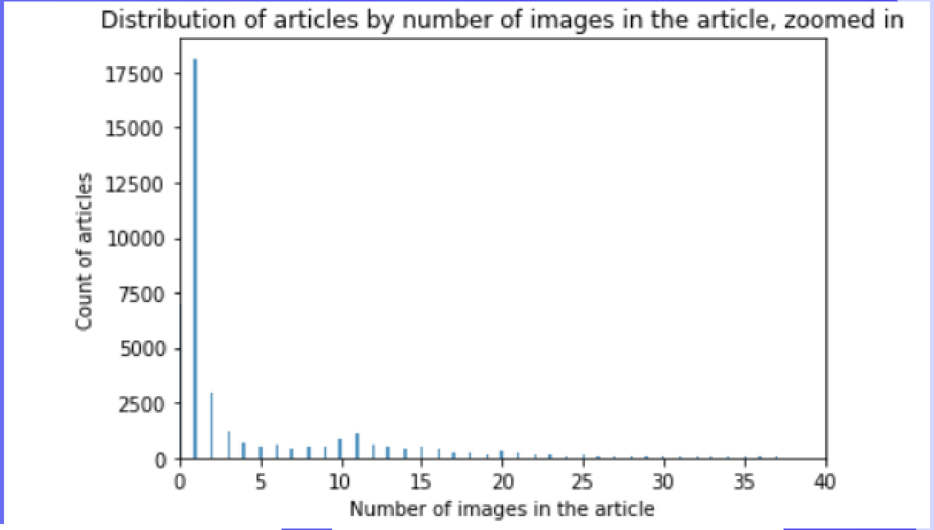
With the graphs here and in the previous slide, we can say that articles of all popularity (75%, 3rd quantile) do no exceed 716 words.

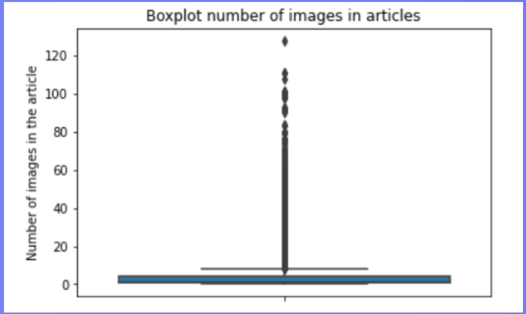


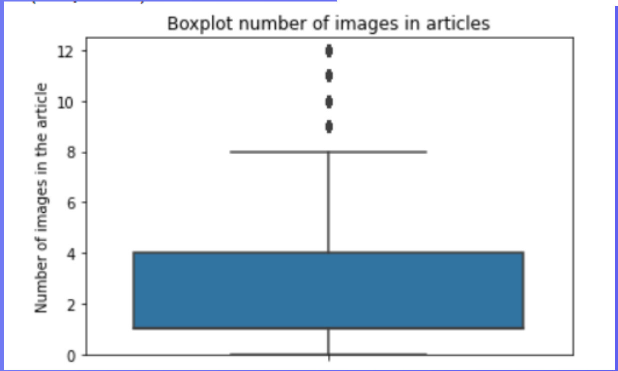
3. Data visualization : Content of the articles

Number of images in the article







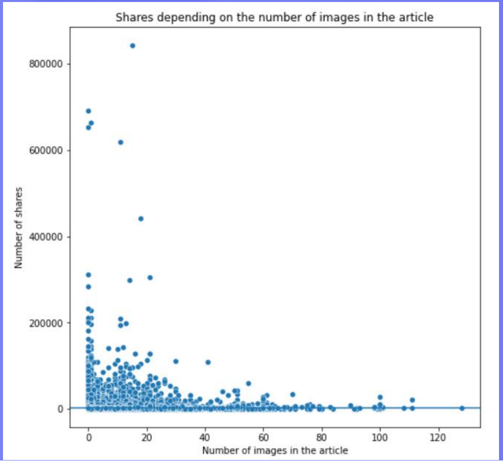


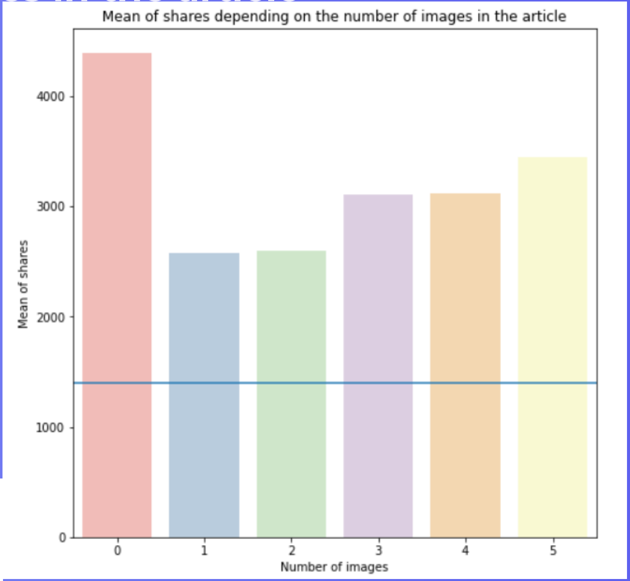
```
count     39644.000000
mean          4.544143
std           8.309434
min           0.000000
25%           1.000000
50%           1.000000
75%           4.000000
max         128.000000
Name: num_imgs, dtype:
```

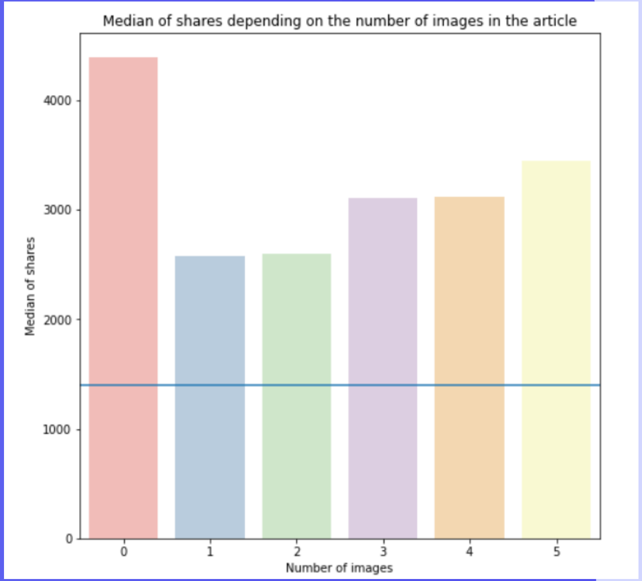
Interpretation:

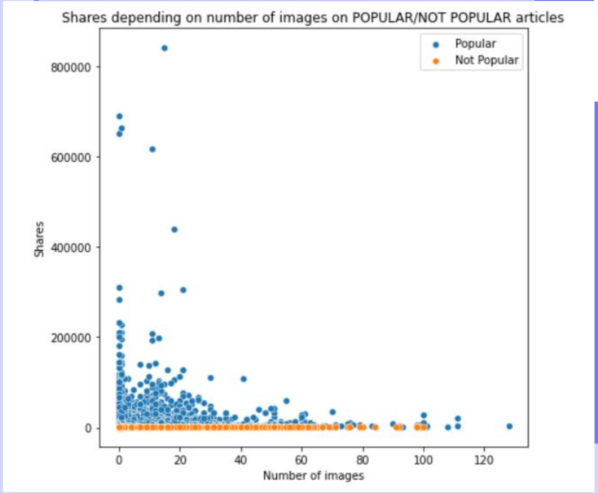
3. Data visualization : Content of the articles

Number of images in the article





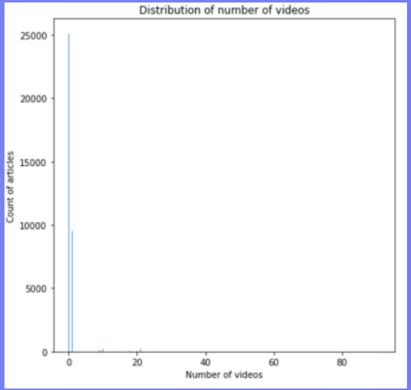


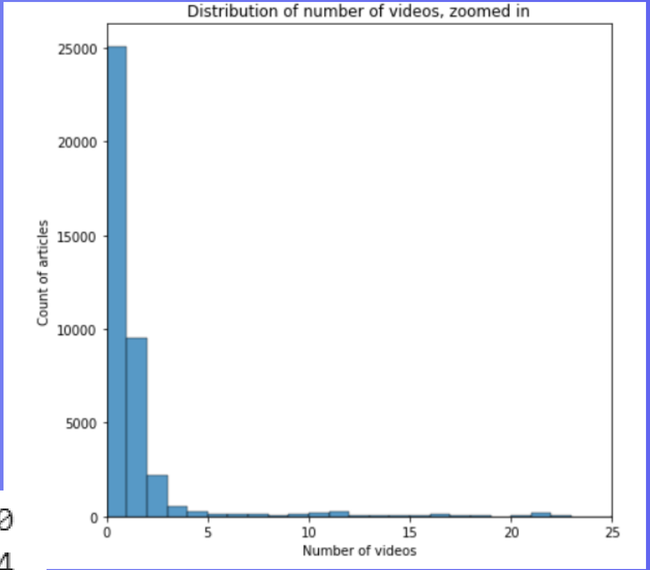


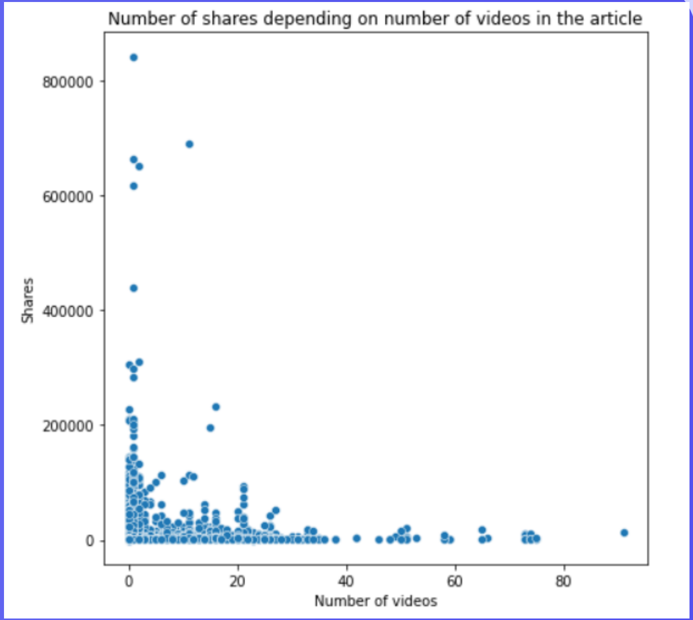
With the graphs here and in the previous slide, we can see that popular articles have a tendency to have less images but also there are less articles in the dataset with a lot of images.

3. Data visualization : Content of the articles

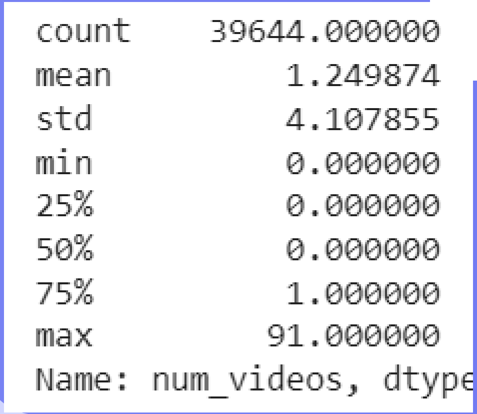
Number of videos in the article







All articles

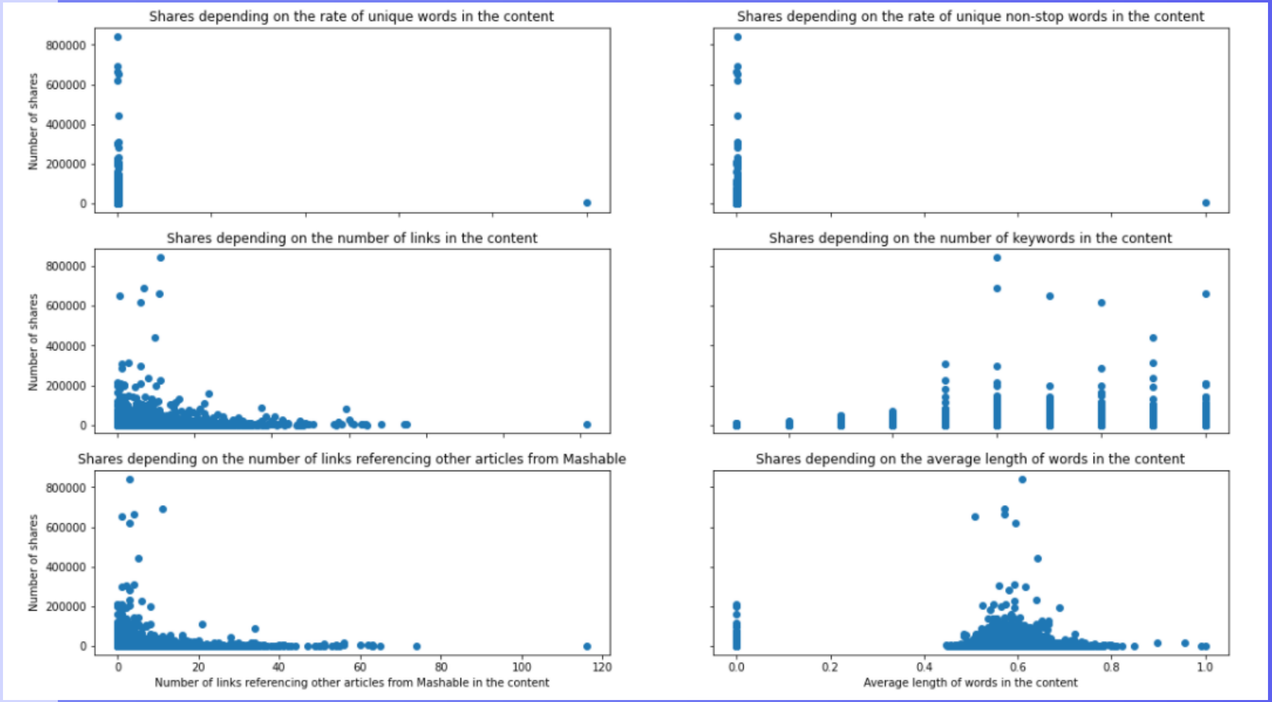


```
count     39644.000000
mean          1.249874
std           4.107855
min           0.000000
25%           0.000000
50%           0.000000
75%           1.000000
max          91.000000
Name: num_videos, dtype
```

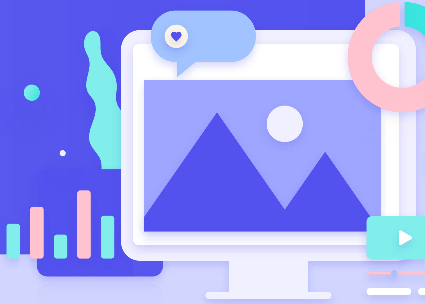
Half of the dataset, does not have a video in its content and the articles with high number of videos do not have more shares, therefore it is hard to predict shares only based on this feature.

3. Data visualization : Content of the articles

Other variables about content.

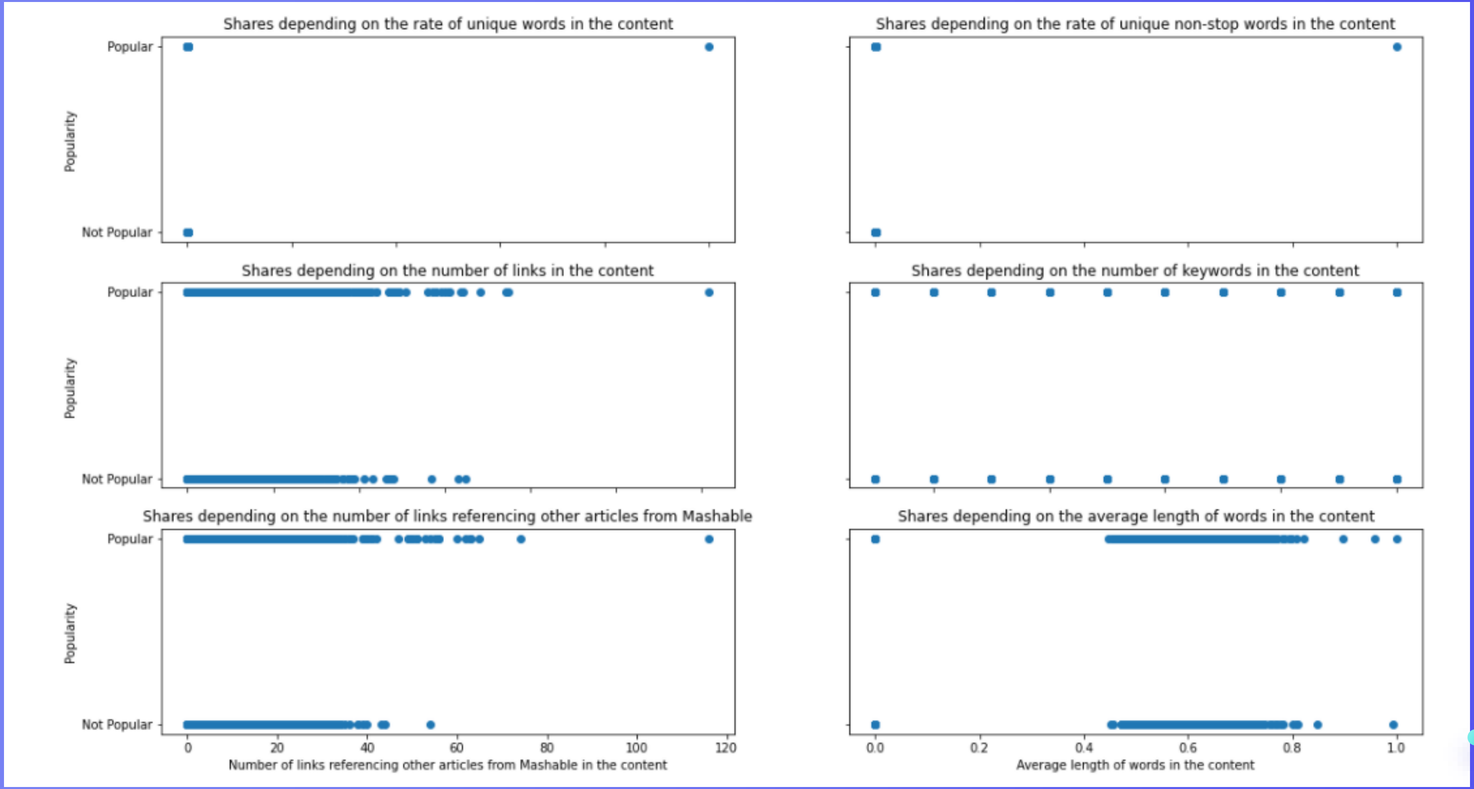


This is quite the same case as the previous slides about content, for the links most articles have a few, for the length of words most articles contain the approximatively words with the same length.



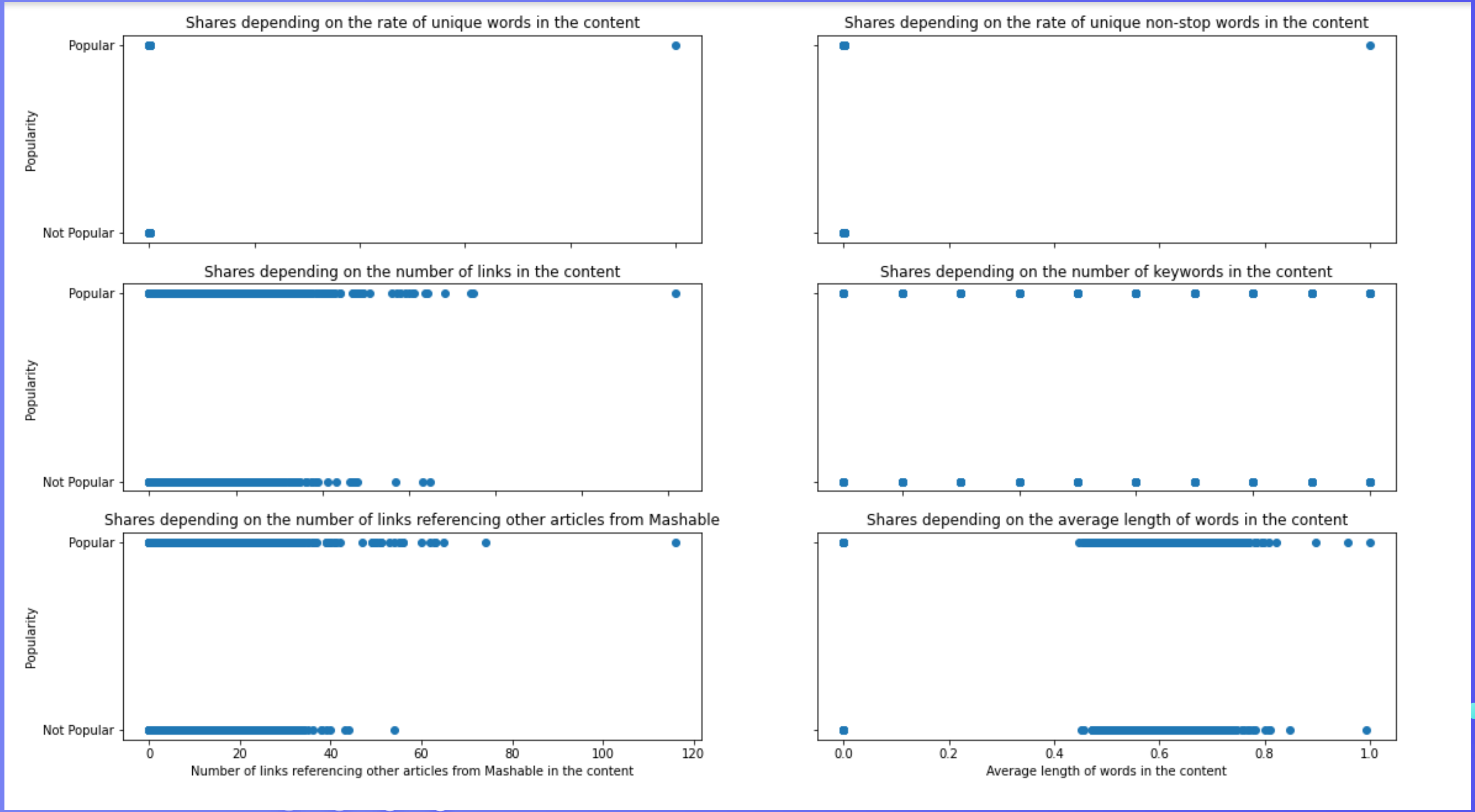
3. Data visualization : Content of the articles

Same data as the previous slide but with categories



3. Data visualization : Content of the articles

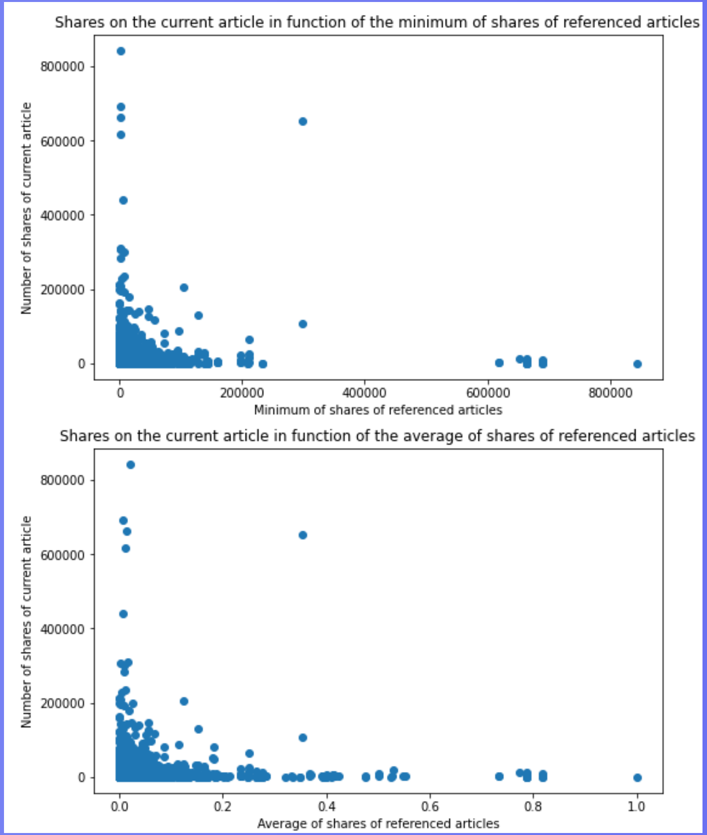
Other variables about content: Do the shares of referenced articles affect the shares?

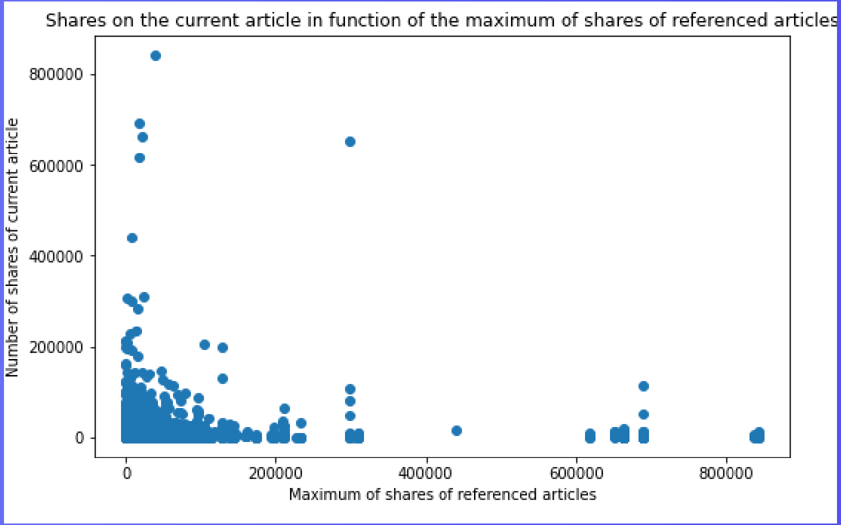


The shares do not affect the shares of the current article (points distributed along the lines)

3. Data visualization : Content of the articles

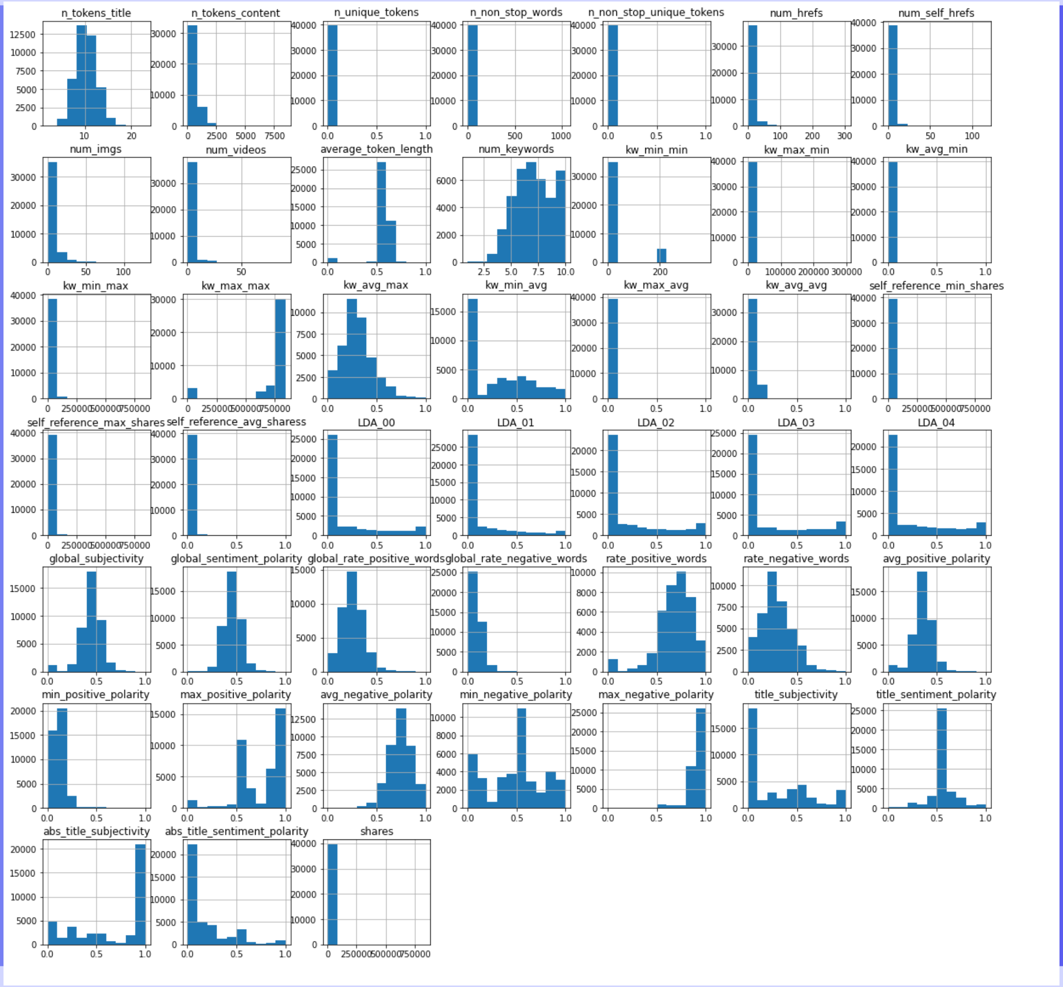
Other variables about content: Do the shares of referenced articles affect the shares?

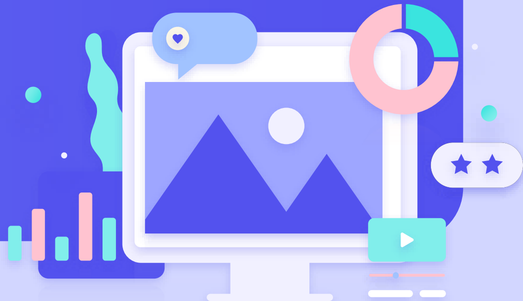




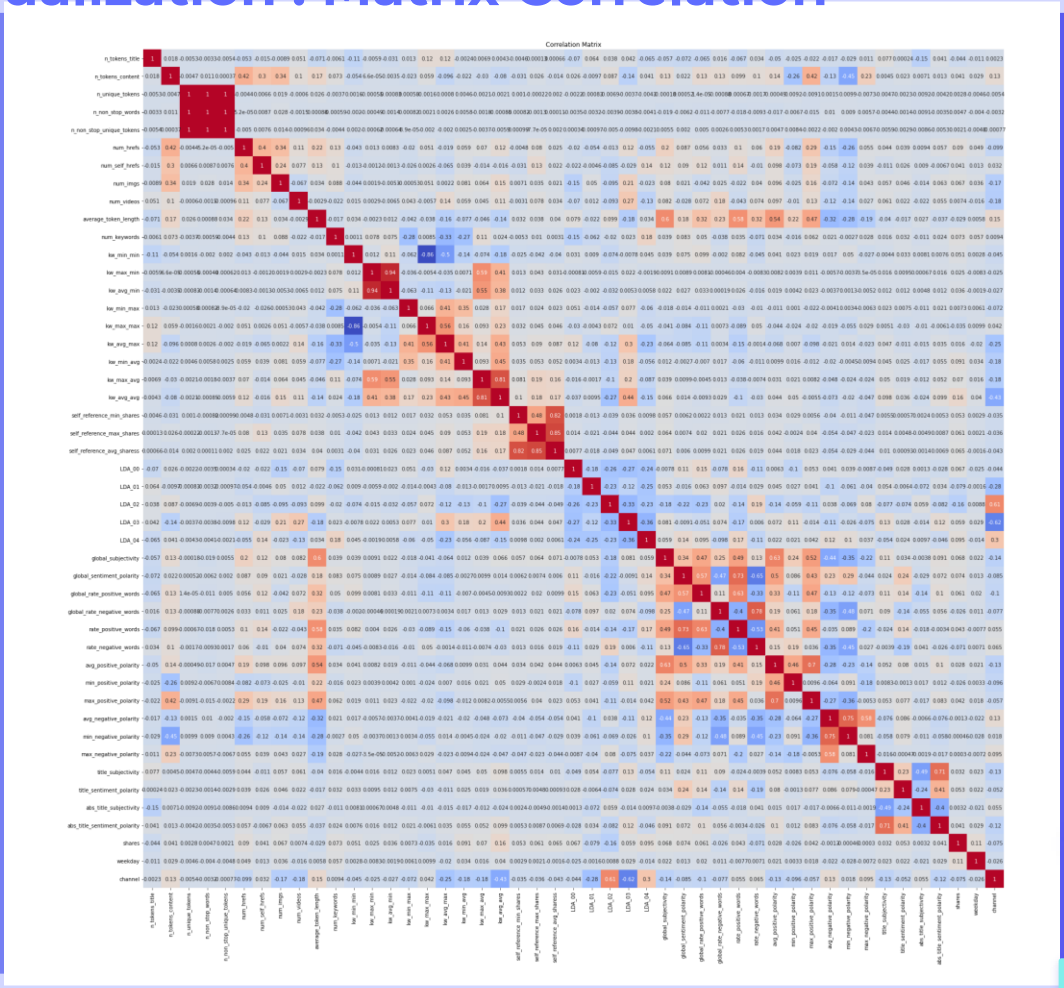
We plotted other graphs in the jupyter file with no particular explanation and did not put them in the report

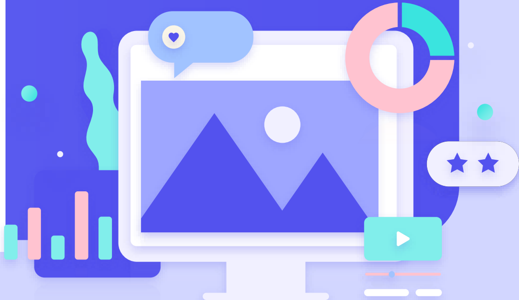
3. Data visualization : Distributions of all the features



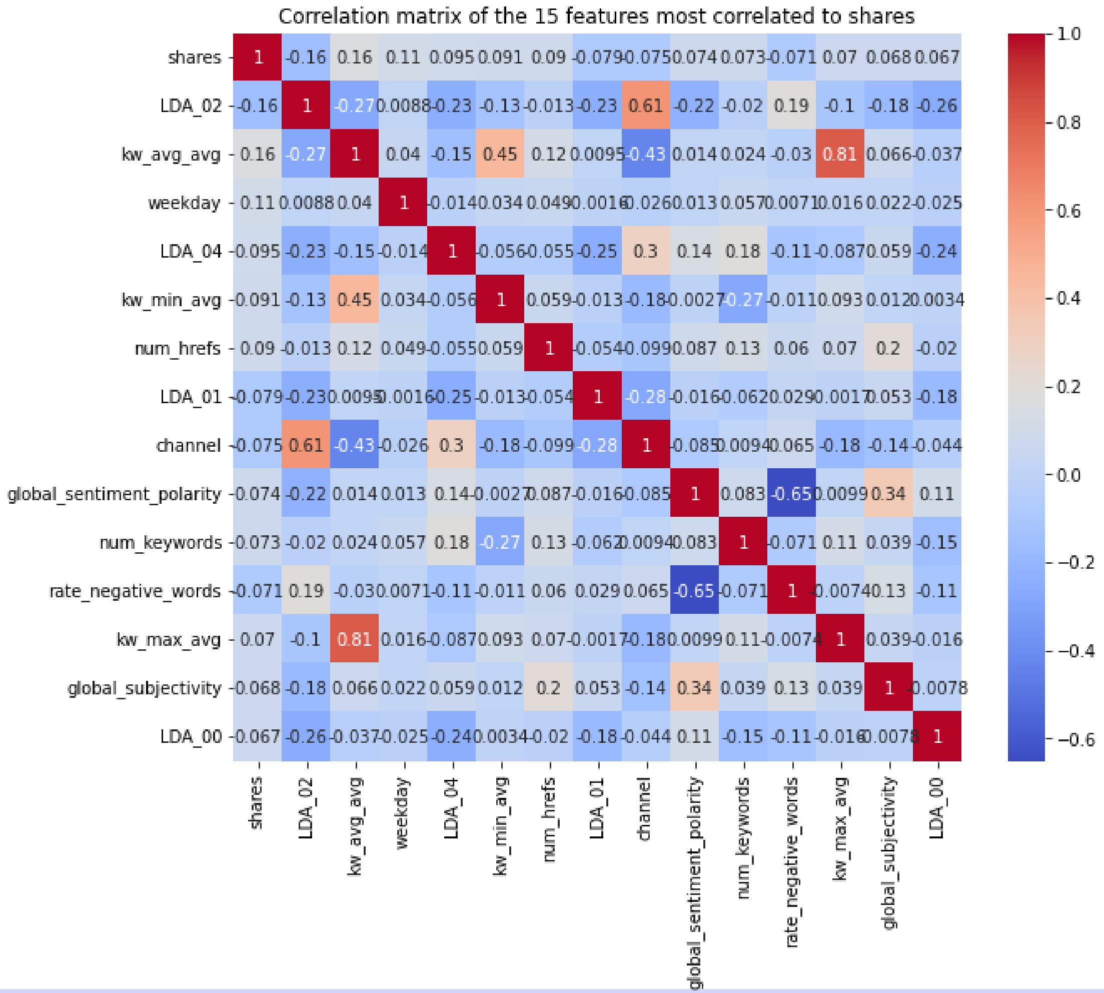


3. Data visualization : Matrix Correlation





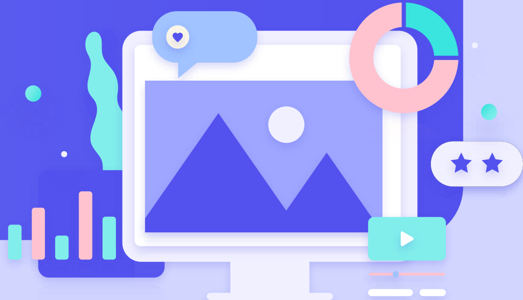
3. Data visualization : Matrix Correlation of 15 most correlated features to shares



	shares	LDA_02	kw_avg_avg	weekday	LDA_04	kw_min_avg	num_hrefs	LDA_01	channel	global_sentiment_polarity	num_keywords	rate_negative_words	kw_max_avg	global_subjectivity	LDA_00
shares	1	-0.16	0.16	0.11	0.095	0.091	0.09	-0.079	-0.075	0.074	0.073	-0.071	0.07	0.068	0.067
LDA_02	-0.16	1	-0.27	0.0088	-0.23	-0.13	-0.013	-0.23	0.61	-0.22	-0.02	0.19	-0.1	-0.18	-0.26
kw_avg_avg	0.16	-0.27	1	0.04	-0.15	0.45	0.12	0.0095	-0.43	0.014	0.024	-0.03	0.81	0.066	-0.037
weekday	0.11	0.0088	0.04	1	-0.014	0.034	0.049	0.0016	-0.026	0.013	0.057	0.0071	0.016	0.022	-0.025
LDA_04	0.095	-0.23	-0.15	-0.014	1	-0.056	-0.055	-0.25	0.3	0.14	0.18	-0.11	-0.087	0.059	-0.24
kw_min_avg	0.091	-0.13	0.45	0.034	-0.056	1	0.059	-0.013	-0.18	-0.0027	-0.27	-0.011	0.093	0.012	0.0034
num_hrefs	0.09	-0.013	0.12	0.049	-0.055	0.059	1	-0.054	-0.099	0.087	0.13	0.06	0.07	0.2	-0.02
LDA_01	-0.079	-0.23	0.0095	0.0016	-0.25	-0.013	-0.054	1	-0.28	-0.016	-0.062	0.029	-0.0017	0.053	-0.18
channel	-0.075	0.61	-0.43	-0.026	0.3	-0.18	-0.099	-0.28	1	-0.085	0.0094	0.065	-0.18	-0.14	-0.044
global_sentiment_polarity	0.074	-0.22	0.014	0.013	0.14	-0.0027	0.087	-0.016	-0.085	1	0.083	-0.65	0.0099	0.34	0.11
num_keywords	0.073	-0.02	0.024	0.057	0.18	-0.27	0.13	-0.062	0.0094	0.083	1	-0.071	0.11	0.039	-0.15
rate_negative_words	-0.071	0.19	-0.03	0.0071	-0.11	-0.011	0.06	0.029	0.065	-0.65	-0.071	1	-0.0074	0.13	-0.11
kw_max_avg	0.07	-0.1	0.81	0.016	-0.087	0.093	0.07	-0.0017	-0.18	0.0099	0.11	-0.0074	1	0.039	-0.016
global_subjectivity	0.068	-0.18	0.066	0.022	0.059	0.012	0.2	0.053	-0.14	0.34	0.039	0.13	0.039	1	-0.0078
LDA_00	0.067	-0.26	-0.037	-0.025	-0.24	0.0034	-0.02	-0.18	-0.044	0.11	-0.15	-0.11	-0.016	-0.0078	1

These are 15 most correlated features to ‘shares’ the target, between them we can see high correlations between:

- channel and LDA_02 which is normal because LDA_02 is Closeness to LDA topic 2.
- Kw_avg_avg and kw_max_avg: which is normal because it is about average keywords.
- Rate_negative_words and global_sentiment_negative correlated negatively which is also normal, because if the rate is bigger the sentiment will be more negative.

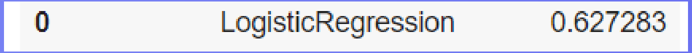


4. Data prediction

4. Data prediction : Logistic Regression

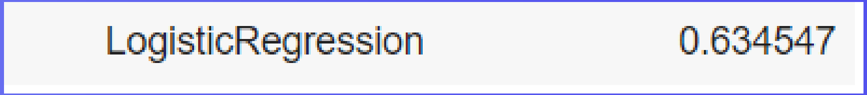
We splitted the dataset into a training set(75%) and a testing set(25%) . We also shuffled the original dataframe.
Logistic Regression is a Machine Learning classification algorithm that is used to predict the probability of a categorical dependent variable.

Cross-validation, score obtained :

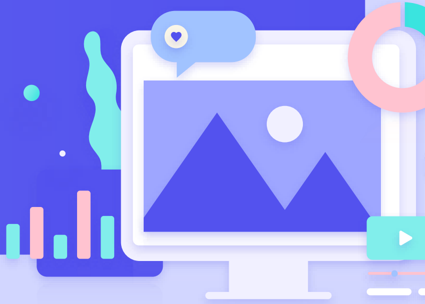


```
0          LogisticRegression      0.627283
```

Predict, score obtained :



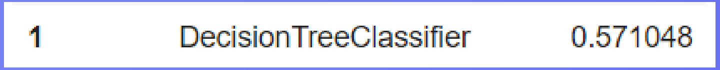
```
     LogisticRegression            0.634547
```



4. Data prediction : Decision Tree Classifier

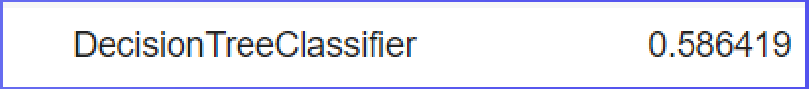
Decision Trees are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.

Cross-validation :

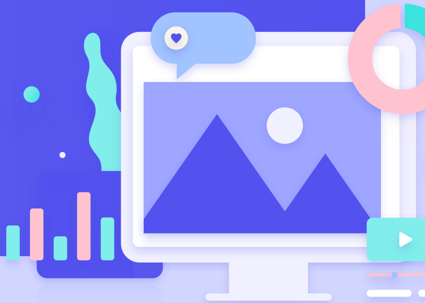


1	DecisionTreeClassifier	0.571048

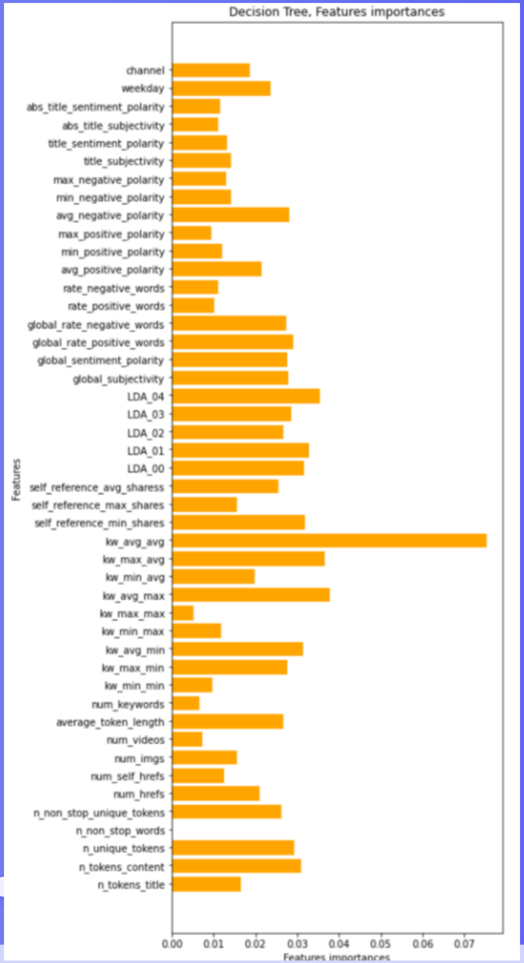
Predict :



DecisionTreeClassifier	0.586419



4. Data prediction : Decision Tree Classifier : Features importances

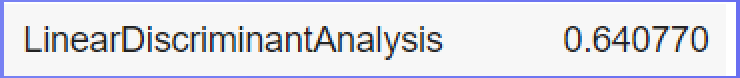


We can observe on the graph that kw_avg_avg is the most importance feature on that model

4. Data prediction : Linear Discriminant Analysis

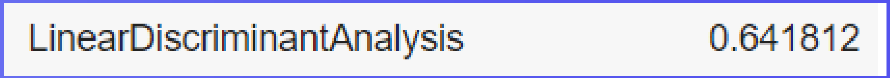
Linear Discriminant Analysis is a classifier with a linear decision boundary, generated by fitting class conditional densities to the data and using Bayes' rule. The model fits a Gaussian density to each class, assuming that all classes share the same covariance matrix.

Cross-validation :

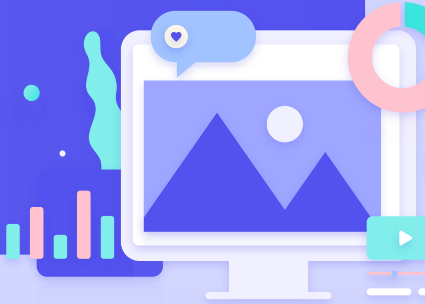


```
LinearDiscriminantAnalysis        0.640770
```

Predict :



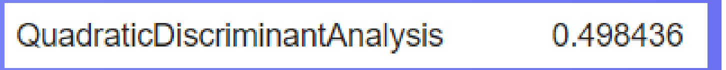
```
LinearDiscriminantAnalysis        0.641812
```



4. Data prediction : Quadratic Discriminant Analysis

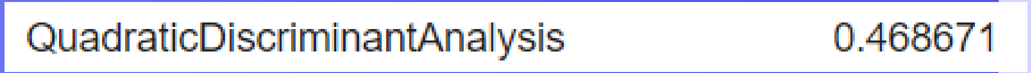
Quadratic Discriminant Analysis. A classifier with a quadratic decision boundary, generated by fitting class conditional densities to the data and using Bayes' rule. The model fits a Gaussian density to each class.

Cross-validation :

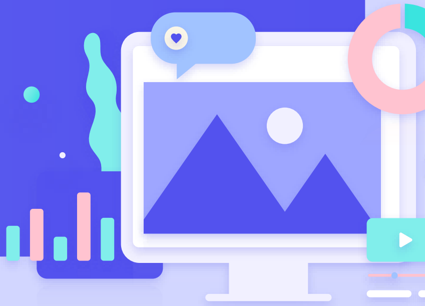


QuadraticDiscriminantAnalysis	0.498436

Predict :

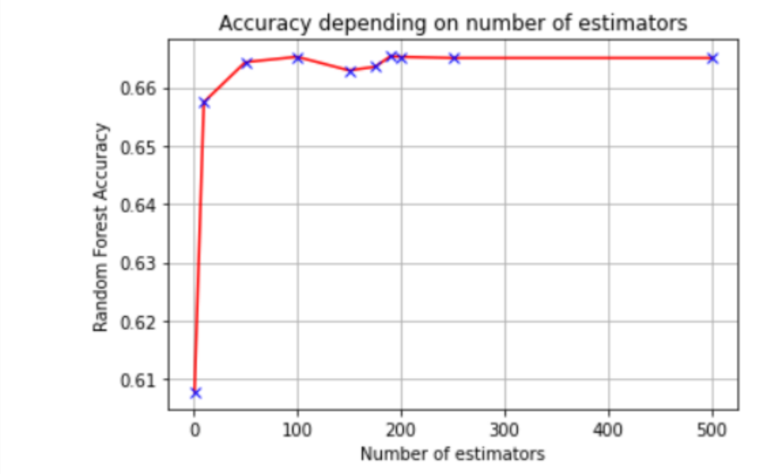


QuadraticDiscriminantAnalysis	0.468671

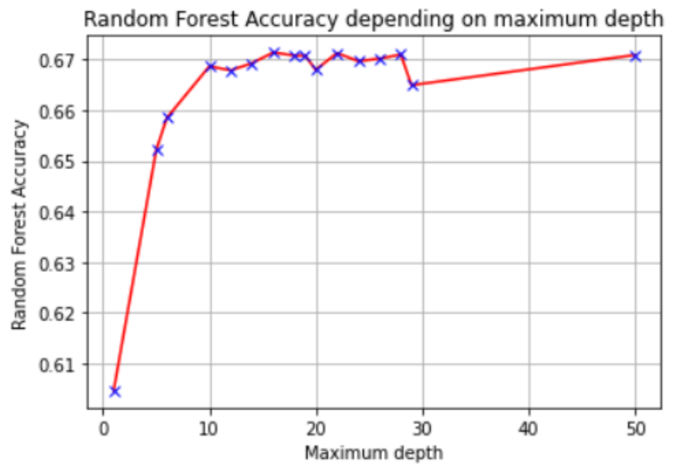


4. Data prediction : Random Forest Classifier

To avoid getting a very bad model, we are going to get some best parameters for the Random Forest Classifier (number of estimators and depth).



The best number of estimators is 190

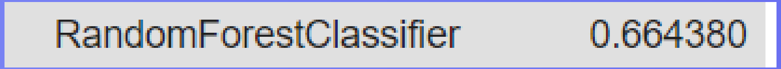


The best number of depth is 28

4. Data prediction : Random Forest Classifier

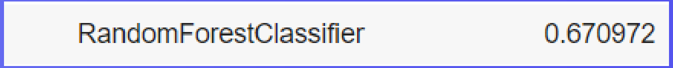
A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

Cross-validation :



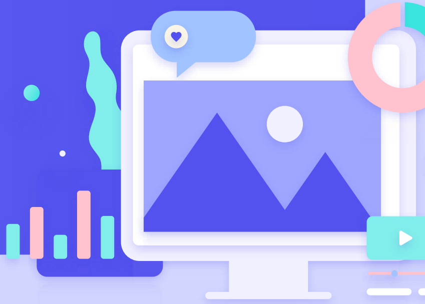
```
RandomForestClassifier      0.664380
```

Predict :

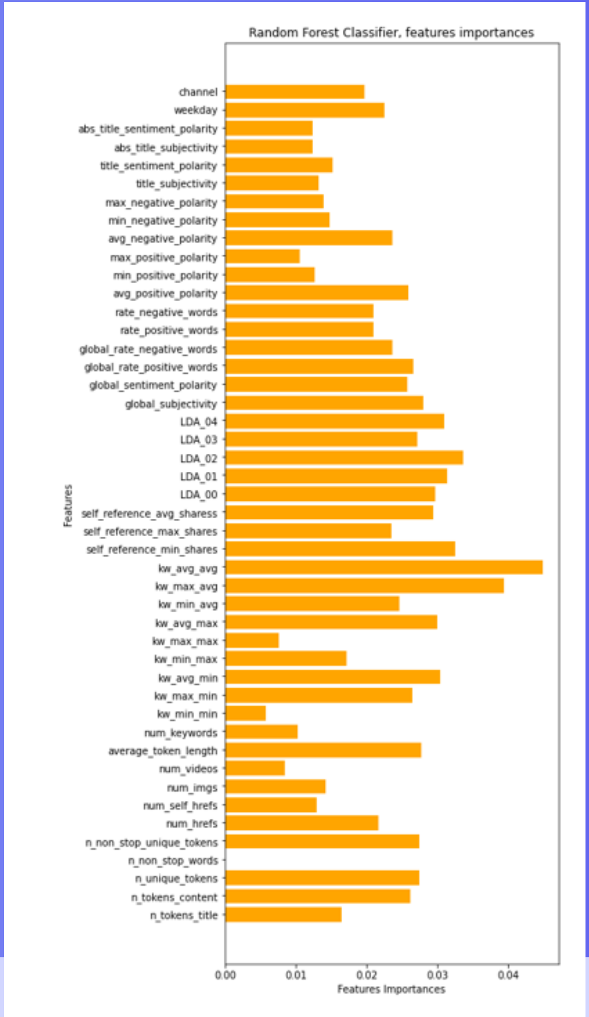


```
RandomForestClassifier      0.670972
```

<u>Interpretation:</u>



4. Data prediction : Random Forest Classifier, Features Importances

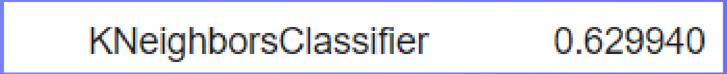


We can observe on the graph that kw_avg_avg is the most importance feature on that model

4. Data prediction : K-Neighbors Classifier

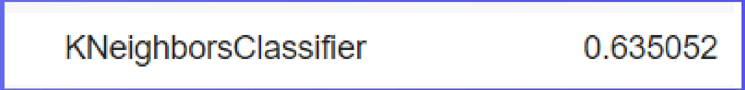
The principle behind nearest neighbor methods is to find a predefined number of training samples closest in distance to the new point, and predict the label from these.

Cross-validation :

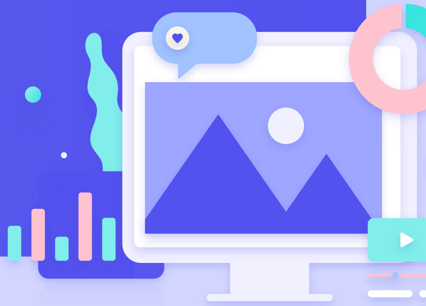


```
KNeighborsClassifier      0.629940
```

Predict :

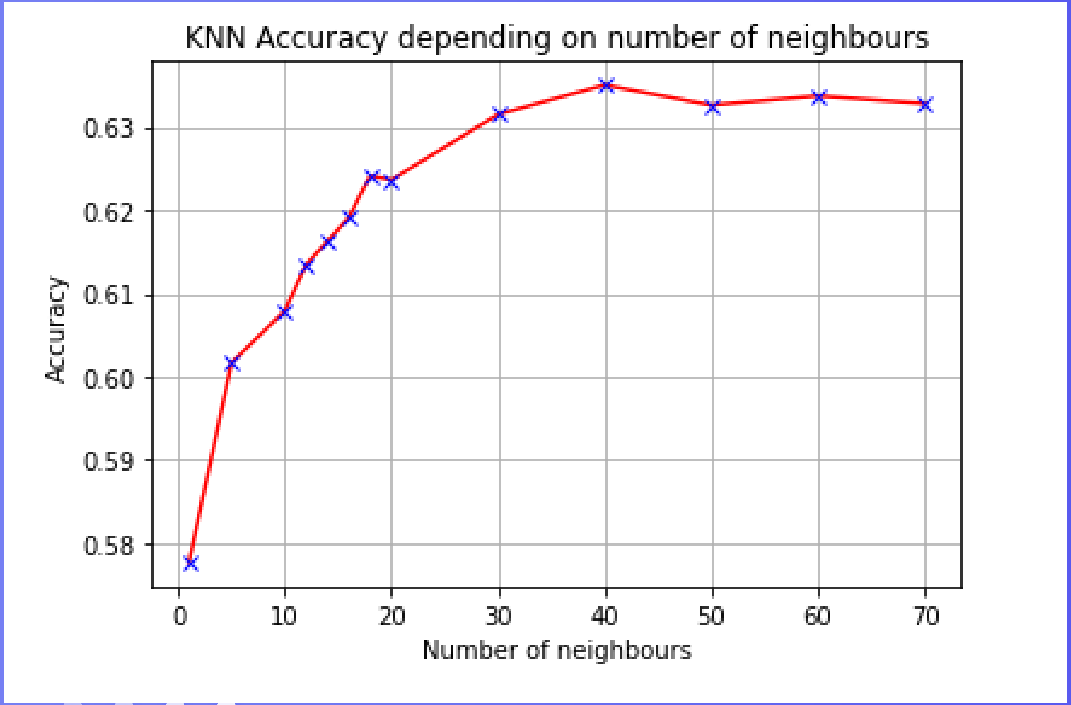


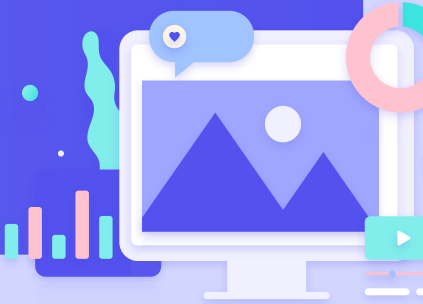
```
KNeighborsClassifier            0.635052
```



4. Data prediction : K-Neighbors Classifier

To avoid to get a very bad model, we are going to get some best parameters for the K-Neighbors Classifier (number of neighbors).

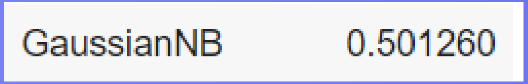




4. Data prediction : Gaussian Naive-Bayes

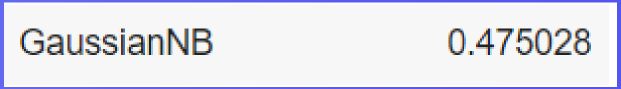
Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes' theorem with the "naive" assumption of conditional independence between every pair of features given the value of the class variable.

Cross-validation :

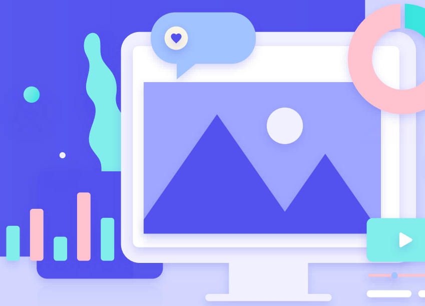


```
GaussianNB      0.501260
```

Predict :



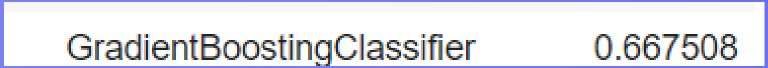
```
GaussianNB      0.475028
```



4. Data prediction : Gradient Boosting

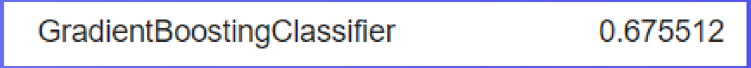
Gradient boosting classifiers are a group of machine learning algorithms that combine many weak learning models together to create a strong predictive model.

Cross-validation :

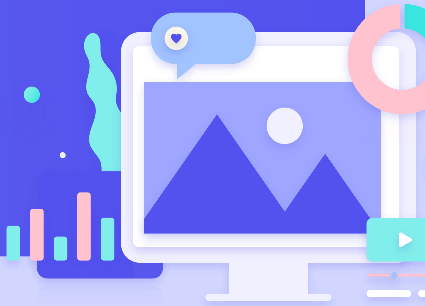


GradientBoostingClassifier	0.667508

Predict :

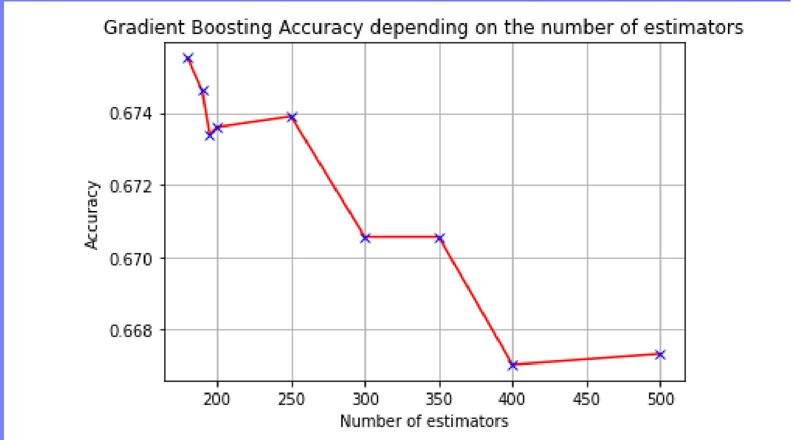


GradientBoostingClassifier	0.675512

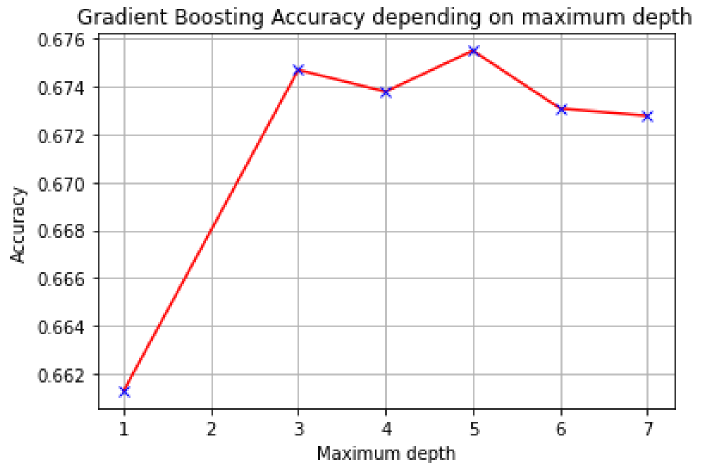


4. Data prediction : Gradient Boosting

To avoid to get a very bad model, we are going to get some best parameters for the Gradient Boosting Classifier (number of estimators and depth).

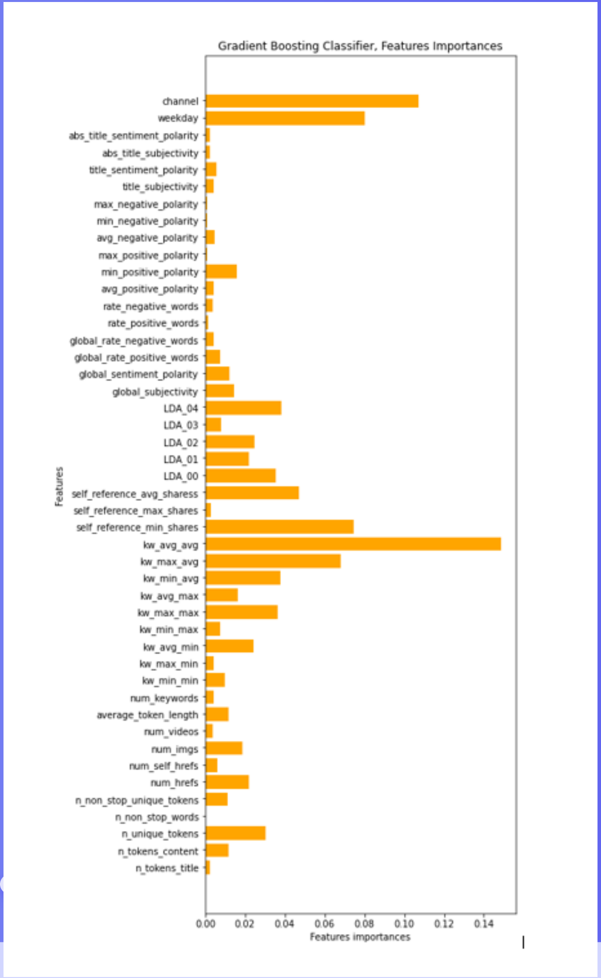


The best number of estimators is 180



The best depth is 5

4. Data prediction : Gradient Boosting, Features Importances

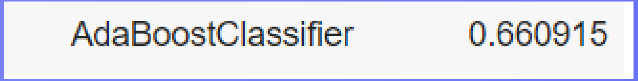


We can observe on the graph that kw_avg_avg is the most importance feature on that model

4. Data prediction : Ada Boost Classifier

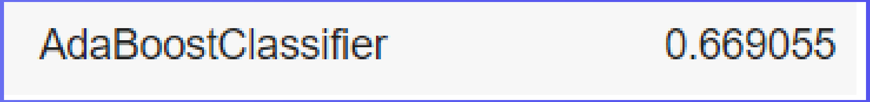
An AdaBoost classifier is a meta-estimator that begins by fitting a classifier on the original dataset and then fits additional copies of the classifier on the same dataset but where the weights of incorrectly classified instances are adjusted such that subsequent classifiers focus more on difficult cases.

Cross-validation :

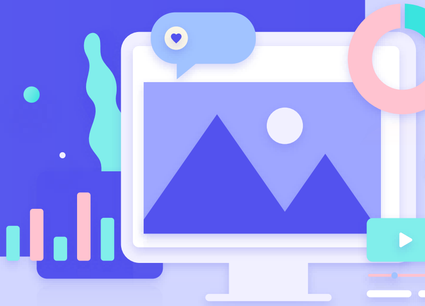


AdaBoostClassifier	0.660915

Predict :

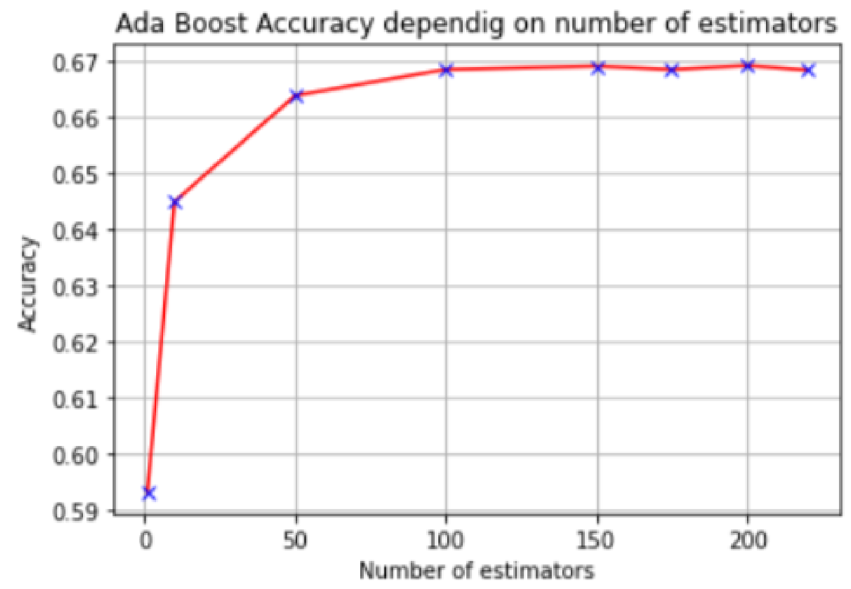


AdaBoostClassifier	0.669055



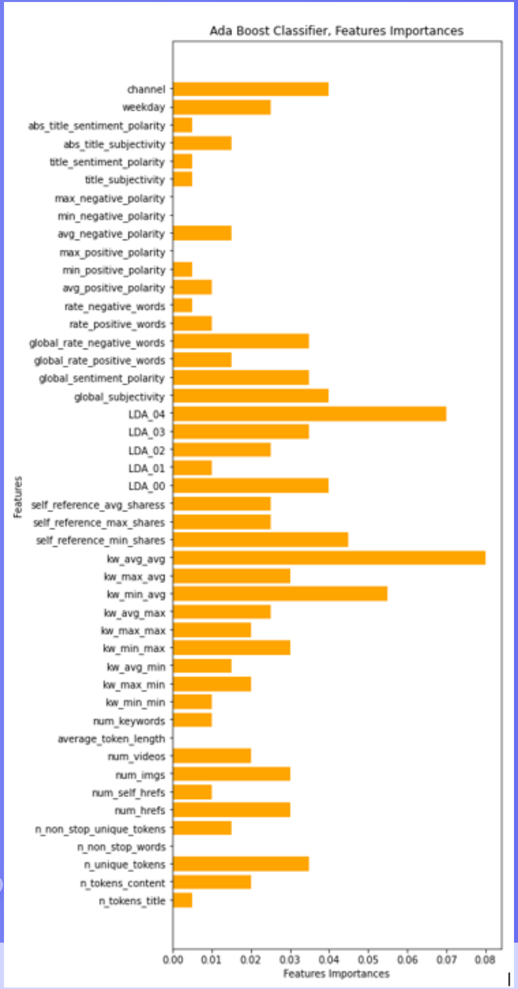
4. Data prediction : Ada Boost Classifier

To avoid to get a very bad model, we are going to get some best parameters for the Ada Boost Classifier (number of estimators).



The best number of estimator is 200

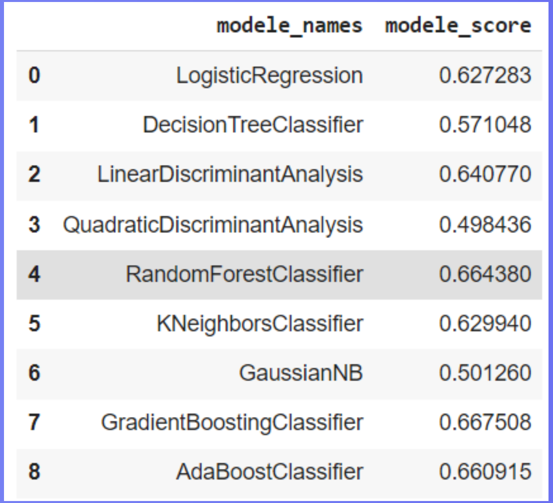
4. Data prediction : Ada Boost Classifier: Features Importances



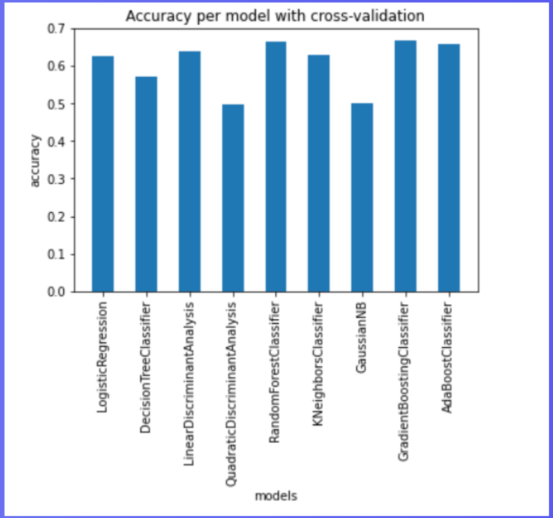
We can observe on the graph that kw_avg_avg is the most importance feature on that model

4. Data prediction : cross-validation method

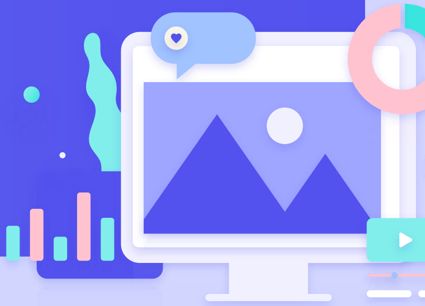
Here, we have the summary of all the models accuracies for the cross-validation method.



	modele_names	modele_score
0	LogisticRegression	0.627283
1	DecisionTreeClassifier	0.571048
2	LinearDiscriminantAnalysis	0.640770
3	QuadraticDiscriminantAnalysis	0.498436
4	RandomForestClassifier	0.664380
5	KNeighborsClassifier	0.629940
6	GaussianNB	0.501260
7	GradientBoostingClassifier	0.667508
8	AdaBoostClassifier	0.660915

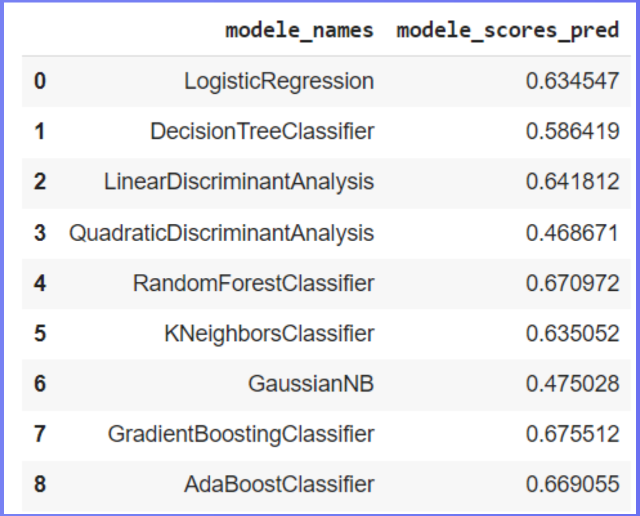


We can see that the Gradient Boosting Classifier has the best accuracy.

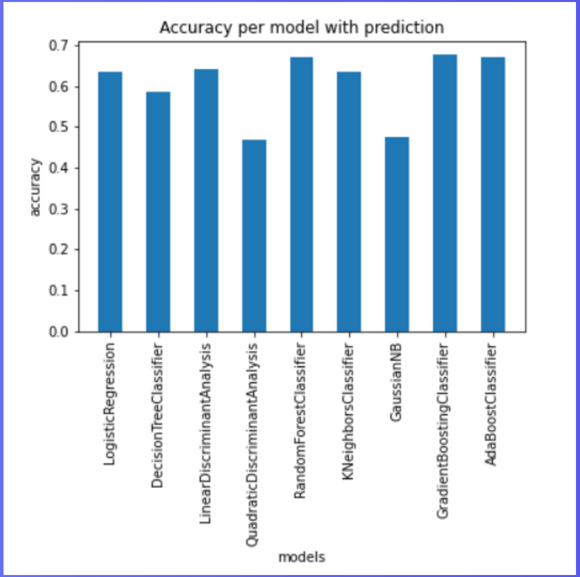


4. Data prediction : prediction method

Here, we have the summary of all the models accuracies for the prediction method.



	modele_names	modele_scores_pred
0	LogisticRegression	0.634547
1	DecisionTreeClassifier	0.586419
2	LinearDiscriminantAnalysis	0.641812
3	QuadraticDiscriminantAnalysis	0.468671
4	RandomForestClassifier	0.670972
5	KNeighborsClassifier	0.635052
6	GaussianNB	0.475028
7	GradientBoostingClassifier	0.675512
8	AdaBoostClassifier	0.669055

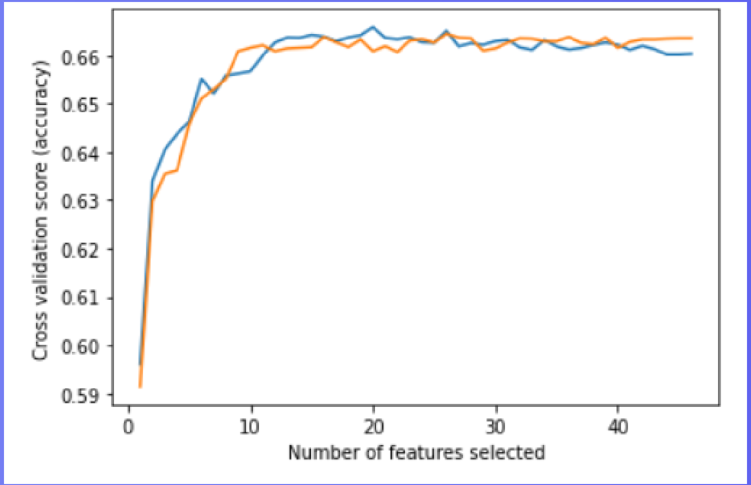


We can see once again that the Gradient Boosting Classifier has the best accuracy score.

4. Data prediction : Tuning of Gradient Boost model

We previously saw that the best model was the Gradient Boosting model. However, this model can be improved again thanks to the tuning.

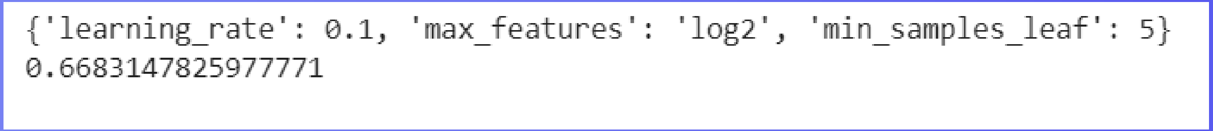
First, we are going to determine the optimum number of features to get the minimum time of running for the best accuracy. For that, we will use the method RFEVC from sklearn.



Thanks to the method RFE from sklearn, we will be able to get the best 26 features from the dataset.

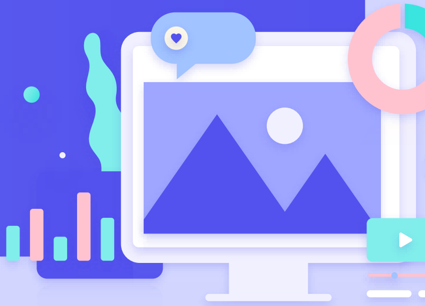
4. Data prediction : Tuning of Gradient Boost model

Next we are going to find the best parameters for the Gradient Boosting Classifier. For this part, the only parameters we used are max_features, learning_rate and min_sample_leaf. Others parameters are existing for this model but it would take too long to run and are less interesting.



```
{'learning_rate': 0.1, 'max_features': 'log2', 'min_samples_leaf': 5}
0.6683147825977771
```

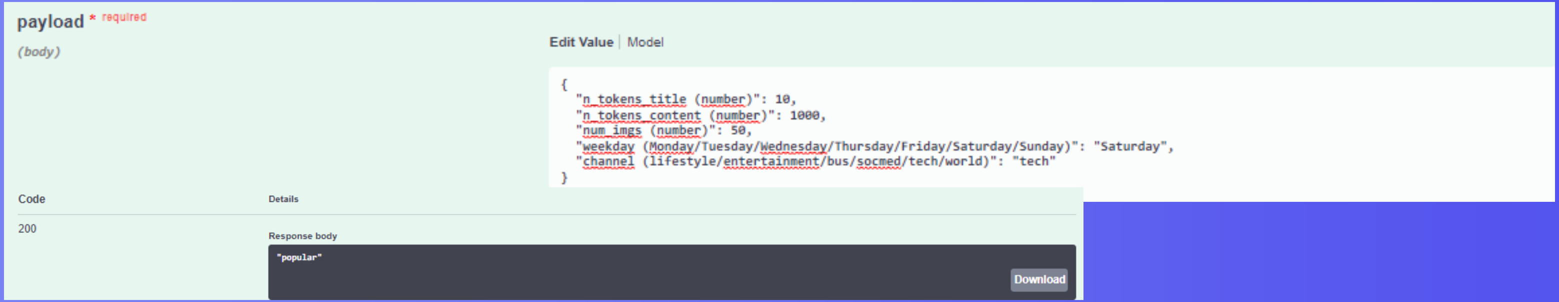
Thanks to the method GridSearchCV from sklearn, we can get the best combination of parameters for the model which is max_features=log2 learning_rate=0.1 max_depth=5 min_samples_leaf=5 and n_estimators=180.



5. Flask API

5.Flask API, explanation on how to use it in the READme file.

Now that we have our best models with the best parameters, let's use it to create an API with flask which takes several features which are understandable by the user (we will not use the 26 features recomanded by RFECV model) and then answer will be given if the article is popular or not thanks to the model.



payload * required

(body)

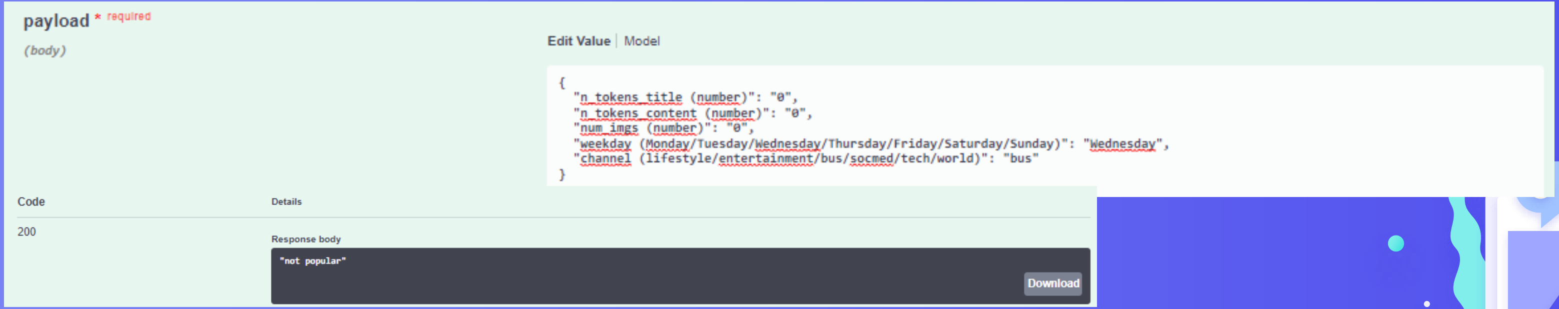
Edit Value | Model

```
{
  "n_tokens_title (number)": 10,
  "n_tokens_content (number)": 1000,
  "num_imgs (number)": 50,
  "weekday (Monday/Tuesday/Wednesday/Thursday/Friday/Saturday/Sunday)": "Saturday",
  "channel (lifestyle/entertainment/bus/socmed/tech/world)": "tech"
}
```

Code	Details
200	Response body

```
"popular"
```

With 10 tokens in the title, 1000 tokens in the content, 50 images, posted a Saturday and with a tech subject, the API answers us that the article should be popular



payload * required

(body)

Edit Value | Model

```
{
  "n_tokens_title (number)": "0",
  "n_tokens_content (number)": "0",
  "num_imgs (number)": "0",
  "weekday (Monday/Tuesday/Wednesday/Thursday/Friday/Saturday/Sunday)": "Wednesday",
  "channel (lifestyle/entertainment/bus/socmed/tech/world)": "bus"
}
```

Code	Details
200	Response body

```
"not popular"
```

With 0 tokens in the title, 0 tokens in the content, 0 images, posted a Wednesday and with a bus subject, the API answers us that the article should not be popular