Capstone Project

Machine Learning Engineer Nanodegree

Predicting article retweets and likes based on the title

DINNE LIDIYA

February 23rd, 2019

1. Definition

This section defines problem statement as well as an overview of the predicting the number of likes and retweets an article receives based on the title. The metrics used for evaluating the model and the function set are also described below.

1.1 Project Overview

Social networks websites have become an important communication tool and source of information. The hours spent on average connected per day in the past years is up to 6 hours for adults and 9 for teenagers, while 30% of this time is on social networks . During a normal navigation on such platforms, users are exposed to several posts such as friends' statuses, images, news and more. With such amount of information and variety of content, the time for the user to decide to interact with the content is very small. we take around 50 milliseconds to make a good first impression and this has proved to be very powerful in a wide range of contexts.

Besides being a place for connecting with friends and sharing moments of the user's life, a survey has shown that social networks are also used as a source of news and information by 67% of the users. Part of these posts are articles that can be read on an external website. Typically such posts show the title of the article and sometimes a small part of its content and an image.

Considering the offer of content and competition with so many interesting posts, showing a proper title for the post affects the probability that a user will check the content. This measure has a strong impact on how many readers an article will have and how much of the content will be read. Furthermore, showing the user a content they prefer (to interact) increases the user satisfaction. It is thus important to accurately estimate the interaction rate of articles based on its title.

1.2 Related Work

In the literature is possible to find previous studies on the area of classifying the article focused on click-baits title detection . Click-bait headlines normally exploit the curiosity of the reader, proving enough information to make the reader curious, but not enough to fully satisfy the curiosity. In this way, the user is forced to click on the linked content to read the whole article. Some other studies also investigate this subject using deep learning on cross-domain sentiment analysis.

1.3 Problem Statement

When an author writes a text, it is expected that their words will influence and bring value to the readers. While writing, the title is one of the important details that needs to be taken into consideration, because this will normally be the first contact place of their work. Thus, to create a good first impression, to have more people read the article and interact with it, choosing a good title is very important.

Some of the most used platforms to spread ideas nowadays are Twitter and Medium. On the first one, articles are normally posted including external URLs and the title, where users can access and demonstrate satisfaction with like or retweet (share) of the original post. The second one shows the full text with tags to classify the article and claps (similar to Twitter's likes) to show how much the users appreciate the content. A correlation between these two networks can bring us more valuable information.

The problem to be solved is a classification task using supervised learning: Predict the number of likes and retweets an article receives based on the title

1.4 Evaluation Metrics

At least one evaluation metric is necessary to quantify the performance of the benchmarks and solution model. For this project, it will be used the accuracy, which is the number of correct predictions made as a ratio of all predictions made

Accuracy = Number of correct predictions / Total number of predictions made

This metric only works well if there are a similar number of samples belonging to each class. For this reason, we will divide the range of retweets and likes count in a way that respects this distribution.

2.Exploratory Visualization

This section will explore the data visualization of the existing dataset and analyze the possible metrics that will be used to understand the solution. We will identify the relationship between each one of the features with the overall performance of the article. Within the following steps we will discuss:

* General statistic overview of the dataset

* Distribution of retweets, claps and favorites

* Relation of the title length with the performance of the article

* Relation of the article's categories with the performance of the article

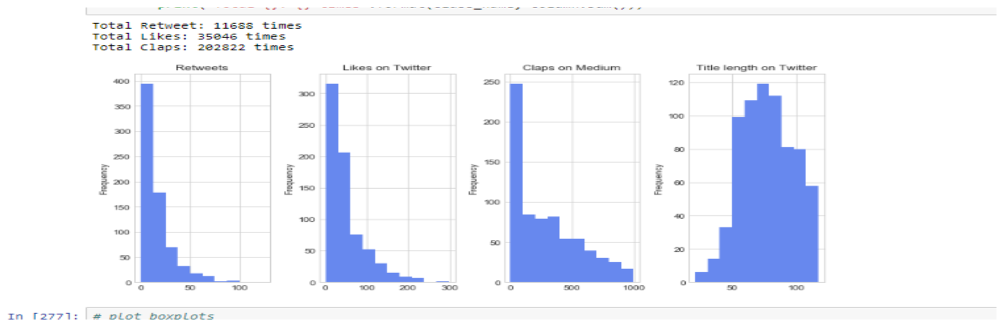
* Relation of the article's words with the performance of the article

* Try to identify the relation between the features

2. Histogram and Box plots

In this section we will check how the multiple features are distributed.

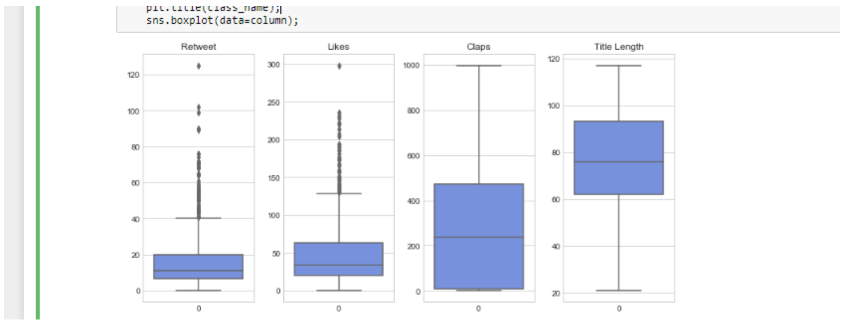
Histogram:



```
Total Retweet: 11688 times
Total Likes: 35046 times
Total Claps: 202822 times
```

```
In [277]: # plot boxplots
```

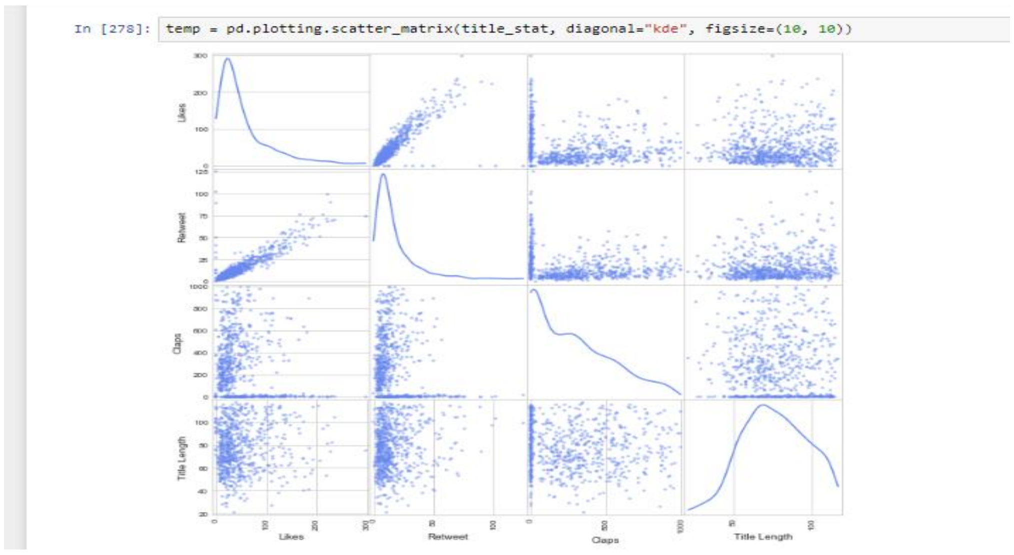
Boxplot:



```
    plt.title(class_name);
    sns.boxplot(data=column);
```

3. Scatter Matrix :

Here we try to find a relationship between the multiple features that we gathered from Twitter and Medium. We can notice for the image 3, we can notice a clear relationship between the number of retweets and likes. They are directed connected, it means, the more retweets, the more likes the article will receive and vice versa.



```
In [278]: temp = pd.plotting.scatter_matrix(title_stat, diagonal="kde", figsize=(10, 10))
```

2.1 Algorithms and Techniques

Classification is a common task of machine learning (ML), which involves predicting a target variable taking into consideration the previous data. To reach such classification, it is necessary to create a model with the previous training data, and then use it to predict the value of the test data. This process is called supervised learning, since the data processing phase is guided toward the class variable while building the model.

Predicting the number of retweets and likes of an article can be treated as a classification problem, because the output will be discrete values (range of numbers). As input, the title of the articles with each word as a token (t1, t2, t3, … tn), the title length and the number of words in the title.

For this task, we evaluated the following algorithms:

***Support vector machine (SVM)*:** SVM contructs a hyperplane (or a set) that can separate the points in the defined labels. The distance between the closest data points and the hyperplan is named margin. An ideal separation is defined by a hyperplan that has the largest distance to the closest points of any class, so the challenge is to find the coefficients that maximize this margin. Only these closest data points are relevant to identify (or to support the definition of) the hyperplane, and they are named vectors. SVM performs linear classifications, but also can efficiently perform non-linear, for this is necessary use a *kernel trick*, mapping their inputs into a high-dimensional feature spaces.

This model was chosen, because it works well then big quantity of features and relatively small quantity of data and to deal well with linear and non-linear datasets. And due the fact we have more samples than number of features, it can generate a good prediction.

Decision trees: This model uses a decision tree to classifies the dataset into smaller subsets, and to define a conclusion about a target value. The tree consists of leaves, where the intermediate ones are the decision nodes and the ones from the extremes are the final outcomes.

This model was chosen, because it can be easily interpreted, visualized and explained. Also due the fact that this model implicitly perform variable screening or feature selection.

***Gaussian naive Bayes (GaussianNB)*:** This model is a classification technique based on the Bayes' Theorem. It assumes the independence among the involved features. Nevertheless, this approach performers well even on data that are dependent between them. This algorithm was created by Bayes to prove the existence of God. It relies on the probability of an event, based on prior knowledge of conditions that might be related to the event. This model was chosen, because this family of algorithms can predict well with small set of data and when there is a large number of features comparatively.

K-nearest neighbors (KNN): This algorithm takes in consideration the k closest points (neighbors) around the target and use them learn how to classify the desired point.

This model was chosen, because its simple to implement, no assumption about the data is necessary and the non-parametric nature of KNN gives an advantage in certain settings where the data may be highly unusual.

***Logistic regression*:** This model is named after the core statistical function that it is based on, the logistic function. The Logistic regression estimates the parameters of this function (coefficients), and as result it predicts the probability of presence of the characteristic of interest.This model was chosen, because provides probabilities for outcomes and a convenient probability scores for observations.

***Naive Bayes classifier for multinomial models (MultinomialNB)*:** This model is similar to the Gaussian naive Bayer, but the difference is that it was a multinomial distribution of the dataset, instead of a gaussian one.This is model was chosen, because it works well for data which can easily be turned into counts, such as word counts in text. However, in practice, fractional counts such as TF-IDF may also work.

In the end, it was selected those with the best accuracy. To estimate it, it was used a 5-fold cross validation that splits the dataset in 5 parts, 4 of training and 1 of testing.

2.2 Benchmark

This project run the same testing and training data for multiple algorithms, the comparison between them was used to evaluate the overall performance. The overall benchmark was made comparing our data with the logistic regression results.

3 Methodology

3.1 Data Preprocessing

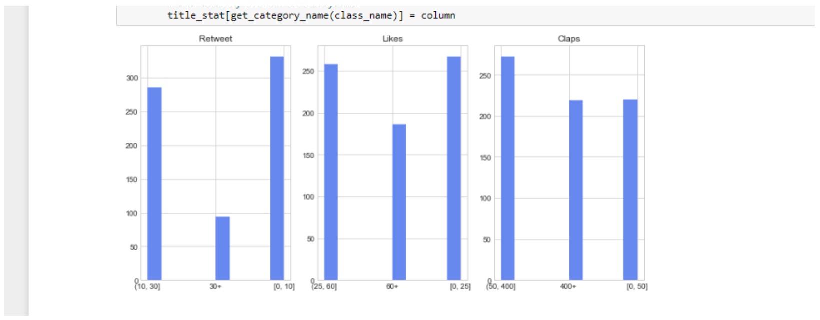
Data cleaning:

The first part of the data processing was to clean the dataset. After downloading the tweets, we removed the ones that didn't have any URL (that points to the Medium article) or title. Data points with values of likes, claps or retweets that were not positive numbers or zero were also excluded.

Words that were Twitter users were replaced by the character '@' (that could be used on the statistics) and words that were wrong non ASCII characters were also removed.

Some of the data points have the same URL, it means, that they shared more than once on the account of Twitter. After analyzing each one of the duplicates, we noticed that there were two types of retweets: same URL and same title; and same URL and a different title. We removed the ones of the first type. For the second type, we left, because the titles were completely rewritten and it can be considered as one different data point.

For the remaining data points, we removed the ones that are considered outliers. We reached the numbers: retweet and likes have 711 items (658 without outliers) each; claps has the same number with or without outliers, 711.



```
    title_stat[get_category_name(class_name)] = column
```

Assigning classes to the dataset

For this project, we decided to classify the number of retweets and likes in ranges. We wanted to make use of the properties of the Classification family of the Supervised Learning algorithms.

To avoid the Class Imbalance Problem, we divided the dataset into similar groups, as shown in image bellow.

Bag of words

To be possible to analyze the title in each data point, we need to map each word into a number. This is necessary because machine learning models normally don't process raw text, but numerical values. To reach this, we used a bag of words model . In this model, it is taken into consideration the presence and often the frequency of words, but the order or position is ignored.

For the calculation of the bag of words, we will use a measure called Term Frequency, Inverse Document Frequency (TF-IDF) . The goal is to limit the impact of tokens (words) that occur very frequently.

At this step, we processed the collection of documents and built a vocabulary with the known words. We reached a vocabulary of 1356 words for retweets, 1399 words for likes and 1430 words for claps.

3.2 Implementation

Training and Testing Data Split

Before starting the training and the evaluation of the models, we split the dataset into test and training sets. Retweets and likes have a total of 658 data points each, with 526 (80% approximately) as training and 132 as testing points. Claps has 711 data points, with 568 (80% approximately) as training and 143 as testing points.

Training and Evaluating Models

For evaluating the model and perform the prediction, it was chosen to start with the models: support vector machines (SVM), decision trees, gaussian naive Bayes (GaussianNB), k-nearest neighbors and logistic regression.

During the implementation tests, we noticed that these models were not satisfying what we expected, so we also trained and predicted using a naive Bayes classifier for multinomial models (MultinomialNB) and gradient boosting for classification.

The classification process followed the steps:

* Load the data (title, likes, retweets and claps count)

* Clean the title removing not desired words

* Filter the outliers from the dataset

* Classify the features in ranges

* Divide the dataset in training and test

* Create a bag of words using TF-IDF for the titles

* Train the model and calculate the accuracy

Model Performance Metrics:

We separated the dataset into learning and validation set. A validation set is important to reduce the risk of over-fitting of the chosen model. To avoid discarding relevant data points, we used a cross-validation strategy.

Cross-validation splits the training dataset in k folds, being k - 1 folders used to train the model and the last one to test it. This strategy will repeat multiple times and the overall performance is the average of the computed values.

To estimate the model's accuracy, we used a 5-fold cross validation that split the dataset into 5 parts, 4 of training and 1 of testing.

Refinement

During the models' implementation a lot of steps were tested and some of them needed to be modified to reach better performance. For choosing a better parameter, we interacted over the options and decided on the one that optimized the accuracy.

Ranges of likes and retweets: We tried ranges with different number of elements and also different values for the ranges. Some of them were underperforming, while others reached close values. The chosen one divided the ranges with a similar number of data points and also offers a good overview of the feature analyzed.

New models: We increased the number of models to be tested in three. The classifiers previous chosen were not reaching the desired accuracy, we decided to add new models to try to make better predictions. The new models have a worse performance than the first ones.

Outliers: We made some tests to discover if we should keep the outliers for the training or remove them. During the tests, we discover if we keep the outliers, the accuracy was always worse.

Bag of words: To create the bag of Words, we had the option of choosing the CountVectorizer or TfidfVectorizer. During the simulation we got better results with the last one, TfidfVectorizer.

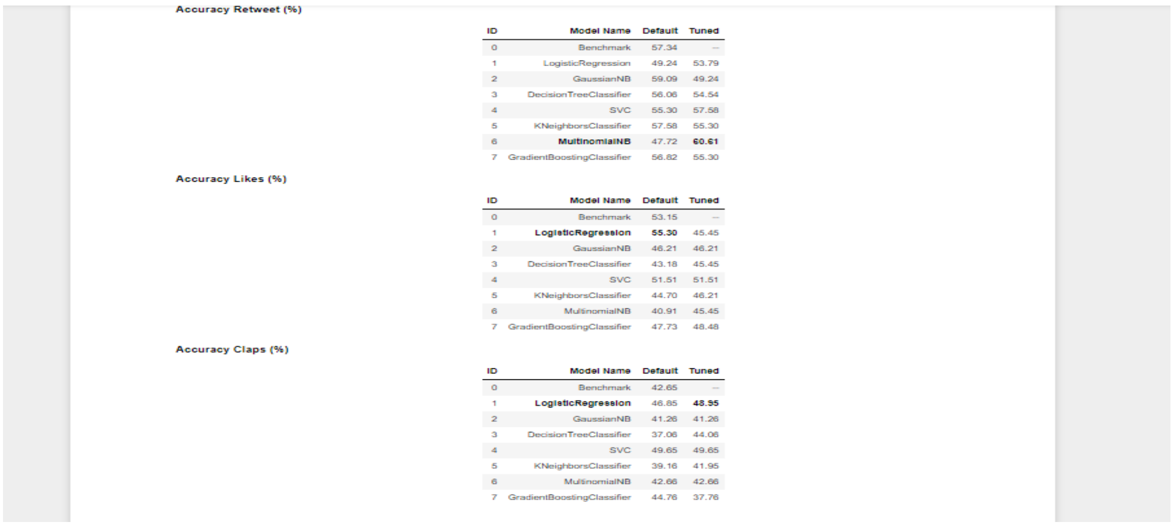
Clean word: Another step during implementation was to decide if we should keep the title in the original way or remove the undesired words. The tested to remove the name of the Twitter users that appeared in some titles, some wrong characters that appeared on our dataset during the crawling process. After checking the results, we decided to clean up the data.

Model's parameters: For each model tested, we calculated the accuracy for the default model (without any parameter, just the default ones) and also we tried to come with better parameters to evolve the accuracy. To test the combination of the new parameters, fine tune the model, we used grid search (GridSearchCV).

Results:

Model Evaluation and Validation

The tables bellow describe the accuracy values we reached with the proposed model. The final accuracy for each of the features are: likes is 55.3%, retweets is 60.6% and claps is 49%.


Accuracy Retweet (%)

ID	Model Name	Default	Tuned
0	Benchmark	57.34	—
1	LogisticRegression	49.24	53.79
2	GaussianNB	59.09	49.24
3	DecisionTreeClassifier	56.06	54.54
4	SVC	55.30	57.58
5	KNeighborsClassifier	57.58	55.30
6	**MultinomialNB**	47.72	**60.61**
7	GradientBoostingClassifier	56.82	55.30

Accuracy Likes (%)

ID	Model Name	Default	Tuned
0	Benchmark	53.15	—
1	**LogisticRegression**	**55.30**	45.45
2	GaussianNB	46.21	46.21
3	DecisionTreeClassifier	43.18	45.45
4	SVC	51.51	51.51
5	KNeighborsClassifier	44.70	46.21
6	MultinomialNB	40.91	45.45
7	GradientBoostingClassifier	47.73	48.48

Accuracy Claps (%)

ID	Model Name	Default	Tuned
0	Benchmark	42.65	—
1	**LogisticRegression**	46.85	**48.95**
2	GaussianNB	41.26	41.26
3	DecisionTreeClassifier	37.06	44.06
4	SVC	49.65	49.65
5	KNeighborsClassifier	39.16	41.95
6	MultinomialNB	42.66	42.66
7	GradientBoostingClassifier	44.76	37.76

Justification

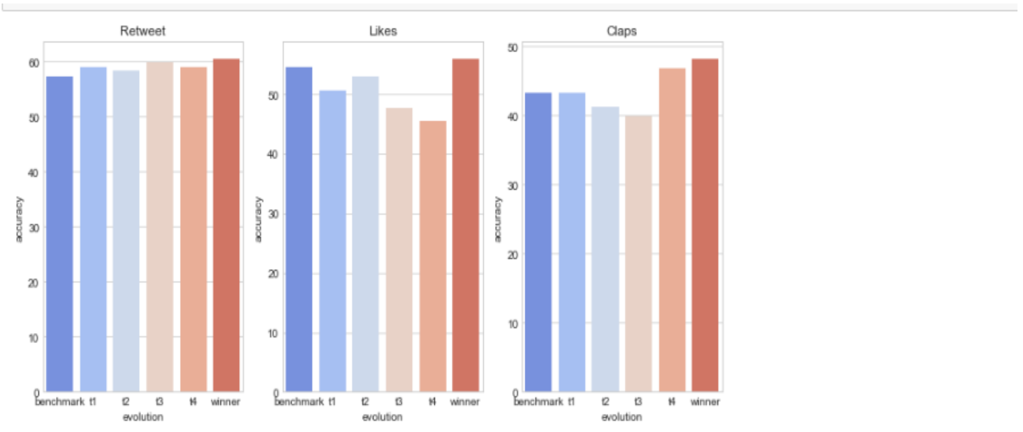
The benchmark used on this project was Logistic Regression model, with ID 0 and named Benchmark on the tables Accuracy Likes, Accuracy Retweets and Accuracy Claps above.

In all the scenarios our models predicted better than the benchmark, for this reason, we can assume the decisions made on the step Metodology - Refinement led us to a better model.

Conclusion

Free-Form Visualization

The image bellow shows the evolution of the development of this project. We started with the benchmark. From there we started adding and testing features and treating the dataset. The steps are described by the table \ref{table:accuracy-evolution}.



During the evolution of the metrics, some variables deprecated the accuracy value, but in further steps, it made it grow. The evolution happened in the following steps:

* Benchmark

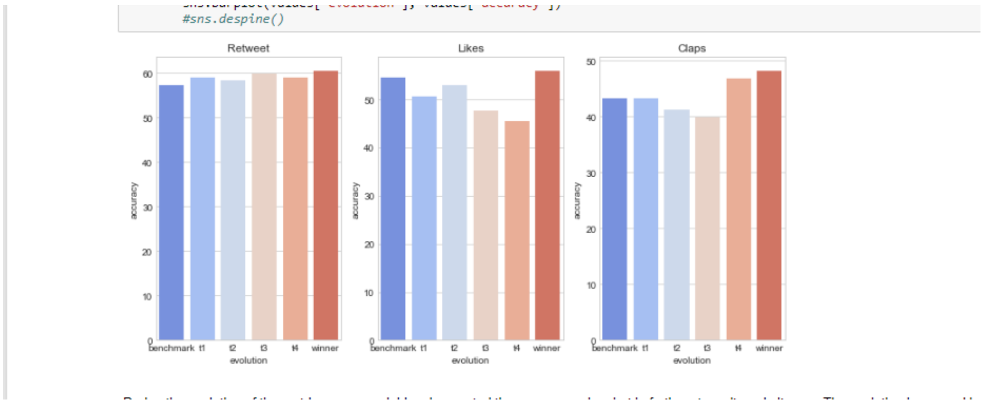
* t1: Removed outliers

* t2: t1 + Cleaned the words

* t3: t2 + Added TF-IDF

* t4: t3 + Used stopwords

* Winner: t4 + Parameters from the model tuned



```
#sns.despine()
```

Reflection

In this project, we developed classifiers to understand how many times an article will receive interaction like retweets and likes (both on Twitter) and claps (on Medium). We also presented a list of words that have a high change to impact positively with the readers, when used on the title or the article. We classified and extracted information about the Categories used on Medium that are commonly presented on our top performers. The number of words and length of the title were also discussed and presented an optimal number to increase the success numbers.

Besides the mathematical analysis used to extract important characteristics of the dataset, we also developed and trained models to predict the how an article would perform. To achieve this machine learning project, some features and characteristics were used:

* Bag of words to tokenize the words of the title

* Term Frequency, Inverse Document Frequency (TF-IDF) to translate the frequency of words in the dataset

* Clean the dataset and each title before training the model (remove Twiter users and invalid characters)

* Grid search to search for the best model parameters

* Remove outliers before processing the data

* Test the dataset to discover a good relation between train and test data points

* Test and split the number of like, retweet and claps in ranges

* Use stopwords to remove common terms of the language

Following these steps listed here, this methodology and framework can be used to classify any kind of article and subjects that are created on Medium and shared on Twitter. This solution is not limited by the context either the subject of the articles and can be easily reproduced to other datasets.

The hard part of the project was to reach a higher accuracy than the one found with simple models, it was necessary multiple reiterations and several modifications on the initial assumption. Reaching the 61%, 55% and 49% is not the ideal solution, but it can clearly lead to the creation of a good title.

Improvement

For future work we can think about some additional improvements: Adding more features to the original dataset making possible to relate more information to the success of the article. For example, we can correlate the words of the title, with trendy words of the month; Bring more data points to train our model, would also increase the accuracy of the solution; and try to use the position of the word on the title to classify its importance.