Public opinion on news

Text mining and sentiment analysis – MSc Data Science and Economics, University of Milan
Maria Pia Chiatante, 944032 – maria.chiatante1@studenti.unimi.it

Introduction

This report is going to present the analysis of public opinion on news, specifically on articles published by the New York Times, with the aim of studying the rate of success of comments made by readers and thus identifying the most controversial topics. The practical implementation has been carried out using the Python language on a Google Colab Notebook, available here: https://github.com/mariapiachiatante/DSE_SAOP_Project/blob/main/SAOP_Project_DSE_MP_Chiatante_.ipynb .

Opinion mining is generally defined as a sub-discipline of computational linguistics that focuses on extracting people's opinion from the web and it has several practical applications, from business and e-commerce industries to collaborative policy-making. In literature, tasks like this have been accomplished using different methodologies, such as supervised **classification** algorithms (e.g. Naïve Bayes, Support Vector Machines, Decision Trees), but also unsupervised techniques, like k-means **clustering**. [1]

In this project we have tried to solve the problem focusing only on supervised classification models, namely logistic regression and random forest.

Research question and methodology

The goal of this project is to analyse the **rate of success of comments** made by readers to articles of the New York Times, which can be interpreted as a measure of how controversial the commented article was (the higher the rate of success of a comment, the more controversial is the related article). The ultimate purpose of the project is that of studying which **topics** were most **controversial**.

Therefore, the research question here can be summarized as follows: "based on comments' rate of success, which are the most controversial topics associated to New York Times articles?".

The methodology used in order to answer to this question is made up of several steps. First of all, there is an **exploratory analysis** which aims at describing the dataset and the features involved.

Then, a **Latent Dirichlet Allocation** (LDA) model is performed on pre-processed data in order to extract latent topics from articles' text.

Finally, two different types of classification models are implemented in order to analyse and predict the success rate of comments: **logistic regression** and **random forest**. For this purpose, a new target variable is created, based on three variables already existing in the dataset. The performance of the different models is evaluated and compared to each other in order to select the best model and to identify the most controversial topics.

Experimental results

Exploratory analysis

The **New York Times Comments dataset** is a collection of articles and comments published on the New York Times in the periods between January - May 2017 and January - April 2018. It is made up of eighteen different .csv files, nine of which represent articles and the remaining nine the comments related to those articles, with a monthly time granularity.
For the purpose of this project, the available .csv files have been grouped into two big datasets: *articles* and *comments*.

The ***articles*** dataset contains information about 9,335 articles over 16 features, where each article is identified by a unique serial alphanumeric code. For each article we know the author, the type (article or blogpost), the headline, the most important keywords, the number of the page on which it was printed, the publication date, the snippet with the text, the source ("The New York Times" or the "International New York Times"), the type of the material (e.g. News, Editorial, Letter, Interview, etc.), the web URL to access the online version of the article and the total number of words.
Moreover, the features *newDesk* and *sectionName* represent the topics associated to the articles. In particular, *newDesk* identifies 44 different topics (e.g. Sports, Climate, Dining, etc.) while *sectionName* identifies 62 different topics (e.g. Politics, Family, Economy, etc.).

The ***comments*** dataset contains information about 2,176,364 comments over 34 features, where each comment is identified by a unique serial integer number. For each comment we know a lot of information but, for the sake of simplicity, we will report here only the most important. For instance, we know the article to which the comment is referred, the type (a comment itself, a user reply to another user's comment or the reporter reply to a user's comment), whether it was promoted by the NYT or not (binary feature *editorsSelection*), the number of the page on which it was printed, the number of recommendations it received, the number of replies that were made to that comment, whether it was shared or not, whether it was identified as trusted or not, the source ("The New York Times" or the "International New York Times"), the location of the user who made the comment and the type of the material (e.g. News, Editorial, Letter, Interview, etc.).

Moreover, again the features *newDesk* and *sectionName* represent the topics associated to the articles to which a comment is referred.

Latent Dirichlet Allocation model

The Latent Dirichlet Allocation (LDA) is a statistical model that allows sets of observations to be explained by unobserved groups, such that similar data are associated to the same group. In this specific case, since observations are words collected into articles, it implies that each article is a mixture of a small number of **topics** and that each word's presence is attributable to one of the articles' topics.

Before performing the LDA model, it is necessary to apply some pre-processing techniques to the *articles* dataset, namely removal of punctuation and stop words and **lemmatization**.

The LDA model is implemented using the Python open-source library "gensim", which is pretty popular for topic modelling tasks. Since the actual number of topics identified by the feature *sectionName* is 62 and those identified by the feature *newDesk* are 44, we set the parameter *num_topics* to 10 in order to extract a small number of more general **latent topics**.

```
[(4,
  '0.099*"International" + 0.064*"Relations" + 0.046*"Media" + 0.040*"United" '
  '+ 0.039*"World" + 0.038*"States" + 0.037*"Rights" + 0.037*"Labor" + '
  '0.037*"Trade" + 0.037*"B"'),
 (2,
  '0.107*"Travel" + 0.097*"Vacations" + 0.095*"Education" + 0.081*"K" + '
  '0.080*"China" + 0.060*"12" + 0.057*"University" + 0.052*"Colleges" + '
  '0.051*"Universities" + 0.045*"State"'),
 (8,
  '0.117*"United" + 0.117*"States" + 0.100*"J" + 0.098*"Trump" + '
  '0.096*"Politics" + 0.094*"Government" + 0.094*"Donald" + 0.025*"US" + '
  '0.022*"Defense" + 0.015*"2016"'),
 (7,
  '0.090*"NY" + 0.070*"Manhattan" + 0.064*"National" + 0.063*"D" + '
  '0.057*"Department" + 0.055*"Times" + 0.048*"Paul" + 0.038*"Restaurant" + '
  '0.037*"NYC" + 0.033*"Soccer"'),
 (0,
  '0.150*"The" + 0.122*"Program" + 0.119*"TV" + 0.104*"Television" + '
  '0.051*"Court" + 0.050*"Supreme" + 0.047*"Office" + 0.046*"Neil" + '
  '0.045*"Stephen" + 0.036*"Interest"')]
```

In the picture above we report the top 5 **most significant topics**, together with the 10 most significant words associated to each topic, and we try to give them an interpretation. For instance, we can infer that the second topic in order of significance (topic 2) can be represented by the general topic "Graduate Education", since it includes words such as "education", "university" and "colleges". Analogously, the third most significant topic (topic 8) can be represented by the general topic "USA Politics", because the terms "United", "States" and "US" appear among the most significant words, together with "Trump" and "Donald".

This library also provides two important evaluation metrics that help us in measuring the performance of an LDA model. The first metric is the so-called **perplexity**, which captures how surprised a model is of new data it has not seen before and is measured as the normalized log-likelihood of a held-out test set. Therefore, the lower the perplexity value, the better the model is: in our case, the perplexity of the LDA model is around to -510.01.
The other important evaluation metric is the **coherence score**, which measures the degree of semantic similarity between high scoring words in a topic, in a range between 0 and 1. These measurements help distinguish between topics that are semantically interpretable and topics that are artifacts of statistical inference. In particular, we chose to compute the "c_v" coherence score, which is based on normalized pointwise mutual information (NPMI) and cosine similarity.
A value of the coherence score closer to 1 identifies a good model, since this means that the relative distance between words within a topic is very low. However, in our case we obtained a value of 0.4118, which is quite low, thus further adjustments could be done in order to improve the LDA model, but this is out of the scope of this project.

Logistic Regression model

As explained in a previous section of this report, the main goal of this project is to study the rate of success of comments in order to identify the most controversial topics associated to New York Times articles. This can be seen as a **classification** task, since comments classified as successful can be interpreted as more controversial and, therefore, classification models can be implemented in order to achieve this goal.
However, the first thing we need to do is to define how we measure whether a comment is considered successful or not. For this purpose, we have created a new **target variable** (called "y") starting from three variables already existing in the dataset, namely *editorsSelection*, *recommendations* and *replyCount*. More specifically, we define a comment as successful if three conditions are satisfied at the same time:

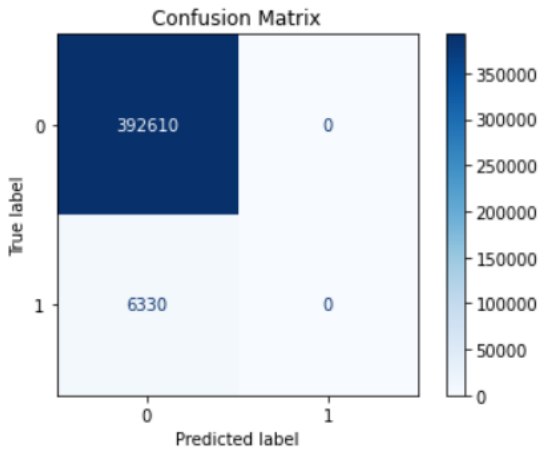
1. It has been promoted by the editor (*editorsSelection* = 1);
2. The number of recommendations received by other users is larger than the average number of recommendations (*recommendations* > 19);
3. It has received at least one reply from other users or reporters (*replyCount* > 0).

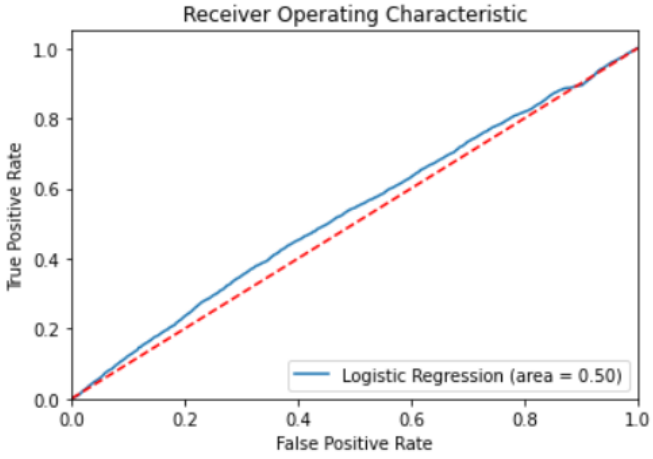
Moreover, we filter only for users' comments excluding other users or reporters' replies, since we are only interested in comments themselves; this allows to reduce the dimension of the data to be analysed and, as a consequence, also computational costs.

After that, we split the data into **training and test sets** and then we are ready to fit a logistic regression classifier on the training data, making a prediction for the test set. The features used as **explanatory variables** of the logistic regression model are the following ones:

- *articleWordCount*, because the total number of words in an article can be related to its importance and, as a consequence, it might represent a more (or less) controversial topic;
- *printPage*, since we believe that the position of an article on the newspaper depends on its relevance and, again, this might have an impact on the success of a comment;
- *sharing*, since we hypothesize that a comment that has been shared is more likely to be successful;
- *trusted*, since we hypothesize that a comment that has been identified as trusted is more likely to be successful;

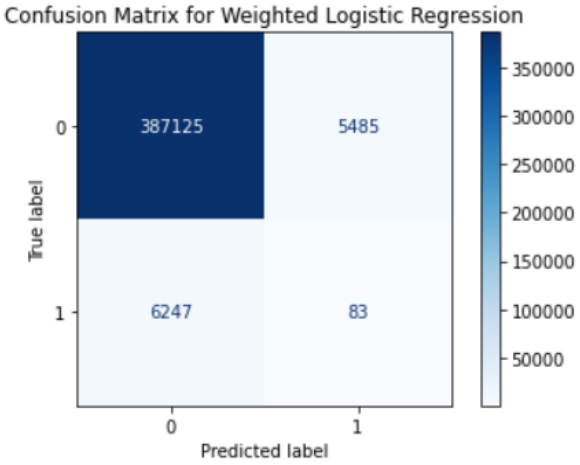
This model produces an accuracy score of 98% which seems to be promising. However, if we take a look at the **confusion matrix**, we observe that our model classifies all the comments as non-successful, which is not good at all since our main goal is to identify successful comments. The **ROC Curve** confirms this evidence because, ideally, we would like to obtain a curve skewed to the left as much as possible, since this would mean that our model is able to minimize the false positive rate and maximize the true positive rate.

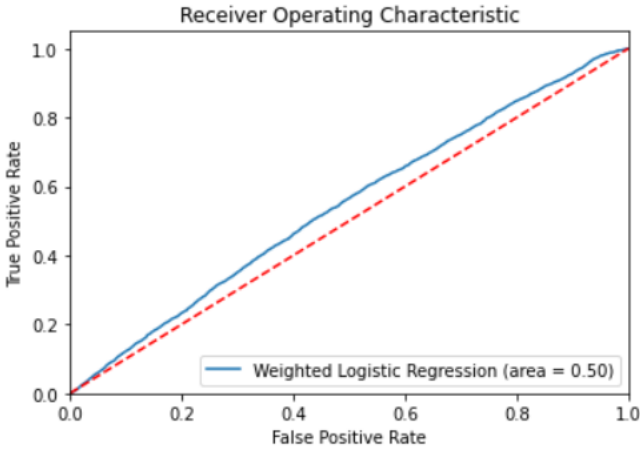




A situation like this usually happens when the **distribution** of the target variable is **highly unbalanced**: most of the observations belong to a majority class but a standard classifier is not able to capture this phenomenon because it assumes an equal distribution of each label. In such cases, the accuracy score can be a misleading evaluation metric, while the ROC Curve and the AUC score are more appropriate since they are able to measure how good a model is at distinguishing between classes.

One possible way of handling this challenge is that of adjusting the **weights** of the classifier according to the real distribution of the data. In this specific case, we see that the target variable has a distribution of majority-to-minority class of 98:2, which means that 98% of the comments in the dataset is non-successful while only the 2% of comments is considered successful. Therefore, we implement a new logistic regression model with weights exactly equal to 98 and 2, using the same explanatory variables as before.

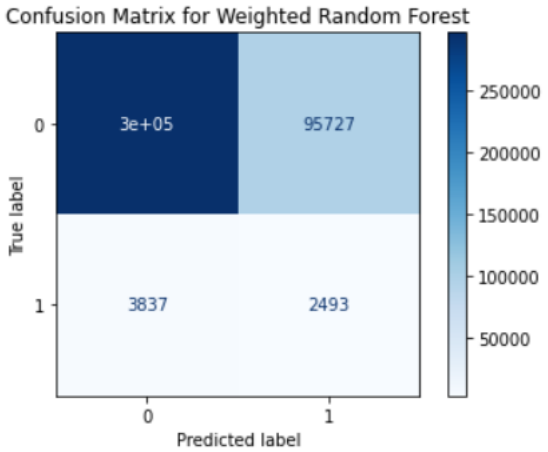


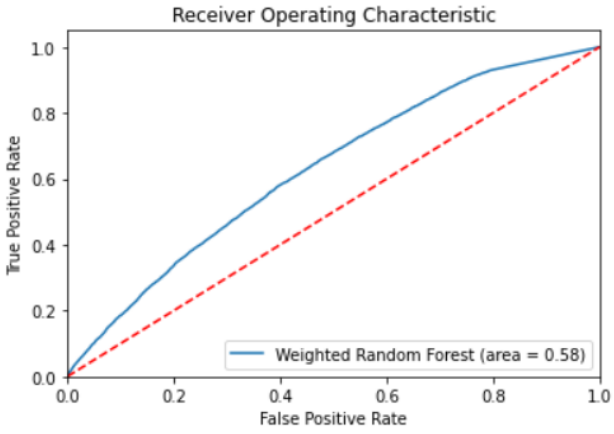


This model performs slightly better than the previous one since now it is able to predict some successful comments, but the true positive rate is only 1% ($\frac{83}{83+6247}$) and the AUC index is still 0.5.

Random Forest

We tried to implement a different classification model, a weighted random forest, in order to see whether there is an improvement compared to the logistic regression classifier. The weights and the explanatory variables considered are the same that we used for the last logistic regression model.

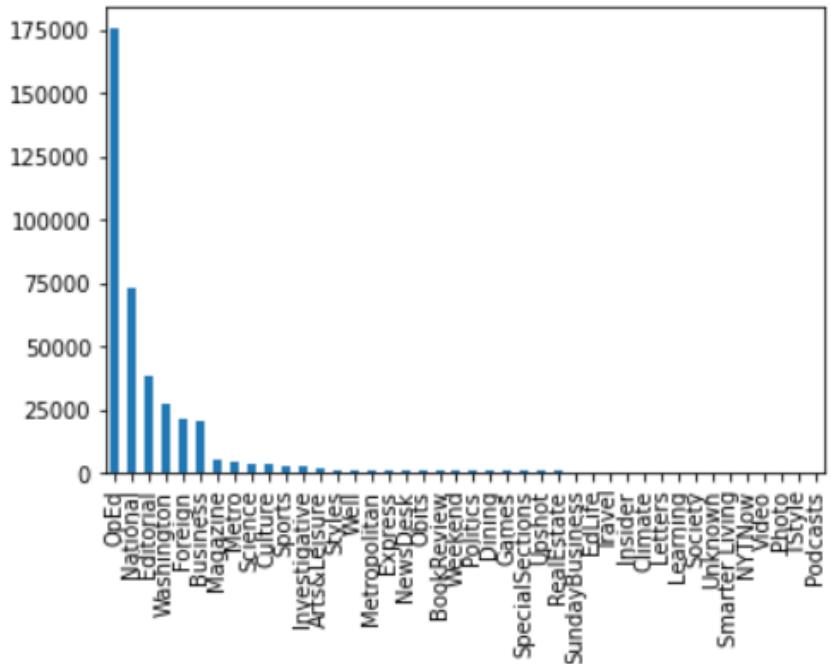




Even if the false positive rate generated by this model has increased a lot, at least now it is able to predict way more successful comments than before, the ROC Curve is more skewed to the left and

the AUC score has changed from 0.5 to 0.58. Therefore, we can conclude that the weighted random forest classifier is performing better than any logistic regression classifier.

At this point, we make a prediction for the whole dataset using the weighted random forest classifier and we filter only for the comments that our model predicted as successful. Then, we look at the topics (represented by the feature *newDesk*) occurring more frequently for those comments and, thus, we are able to somehow identify the **most controversial topics**: OpEd, National, Editorial, Washington, Foreign and Business.



The vertical axis represents the number of successful comments associated to each topic

Concluding remarks

In conclusion, we can say that the random forest model has a better performance on this task compared to the logistic regression model and that, in general, it is useful to adjust the weights when the target variable's distribution is highly unbalanced.

Moreover, the Latent Dirichlet Allocation model, that follows a completely different approach, has allowed us to identify some latent topics that were not captured by the other models.

If interested, the reader can find below some **ideas for future work**:

- changing the way we define whether a comment is successful or not, namely the way we build the target variable, since it can have an influence on its distribution;

- trying to implement other types of classification models, in order to compare their performance and pick the best one;
- including in the model new explanatory variables that were not considered previously, since it can help in making a better prediction.

References

[1] Shelke, N. M., Deshpande, S., & Thakre, V. (2012). Survey of techniques for opinion mining. *International Journal of Computer Applications*, 57(13), 0975-8887.