



HAL
open science

Raising a Model for Fake News Detection Using Machine Learning in Python

Gerardo Agudelo, Octavio Parra, Julio Barón Velandia

► **To cite this version:**

Gerardo Agudelo, Octavio Parra, Julio Barón Velandia. Raising a Model for Fake News Detection Using Machine Learning in Python. 17th Conference on e-Business, e-Services and e-Society (I3E), Oct 2018, Kuwait City, Kuwait. pp.596-604, 10.1007/978-3-030-02131-3_52 . hal-02274166

HAL Id: hal-02274166

<https://inria.hal.science/hal-02274166>

Submitted on 29 Aug 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Raising a Model for Fake News Detection Using Machine Learning in Python

Gerardo Ernesto Rolong Agudelo¹, Octavio José Salcedo Parra^{1,2} and Julio Barón Velandia¹

¹Universidad Distrital “Francisco José de Caldas”, Faculty of Engineering,
Bogotá D.C., Colombia
gerolonga@correo.udistrital.edu.co, {osalcedo,
jbaron}@udistrital.edu.co

²Universidad Nacional de Colombia, Department of Systems and Industrial Engineering, Faculty of Engineering, Bogotá D.C., Colombia
ojsalcedop@unal.edu.co

Abstract. Fake news has been spreading in greater numbers and has generated more and more misinformation, one of the clearest examples being the United States presidential elections of 2016, for which a lot of false information was circulated before the votes that improved the image of Donald Trump over Hillary's Clinton (Singh, Dasgupta, Sonagra, Raman, & Ghosh, n.d.). Because fake news is too much, it becomes necessary to use computational tools to detect them; this is why the use of algorithms of Machine Learning like “CountVectorizer”, “TfidfVectorizer”, a Naive Bayes Model and natural language processing for the identification of false news in public data sets is proposed.

Keywords: Fake News, Machine Learning, NLTK, Sklearn.

1 Introduction

With the arrival of the technological era and with its inventions such as radio, internet and television; information media such as the written newspapers were left aside and the world opened the doors to new ways to find out about the events, most of the news content on television and radio is reviewed and controlled but the content of the internet is hardly supervised and even more so when this does not violate any law (Gu, Kropotov, & Yarochkin, 2017), this is how fake news has become a place in today's society, from apparently harmless publications on social networks (Lazer et al., 2017) to pages web completely dedicated to the production of false information, but made in such a way that they manage to imitate with mastery some of the most recognized newspapers and news channels, besides the fact that it is not possible to make a formal definition of a false news (Mauri, Jonathan, Tommaso, & Michele, 2017).

2 Related work

Because the problem addressed is very relevant in this information age, several previous works have been carried out from different perspectives, focused in different ways and using different techniques, but ultimately all seek to combat misinformation, some of these Studies will be presented below.

In (Rubin, Conroy, Chen, & Cornwell, 2016) they make an approach to the detection of false news based on satire, for this they first made a conceptual description of the satirical humor about twelve satirical humor news and compared it with its real counterpart. For the classification process, they used an SMV (Support Vector Machine) algorithm with five characteristics to predict; absurdity, humor, grammar, negative sensations and punctuation, being the algorithm tested with 360 news achieving 90% accuracy to find satirical news using the combination of absurdity, grammar and punctuation.

In (Bourgonje, Schneider, & Rehm, 2017) they make a study of the news regarding its owner and the relevance with its content, focusing mainly on the detection of clickbait, that is, news in which its heading does not have any relation with its development. The methodology was applied on a public data set and achieved a success of 89.59. In the study they seek to provide a tool that helps to check the news coming from traditional and non-traditional media. In the study separate parts of the content of the news, analyzing them separately and determining their veracity, later if they find that some of them are false or contradict another part of the information, you can guess that it is a false news. This method can also be useful for detecting news with political bias that is, showing a political position favorable to a position.

On the other hand in (Shu, Wang, Sliva, Tang, & Liu, 2017) present a review of several existing methods for the detection of false news, on the one hand there are works focused on the processing of news content and its form, those based on knowledge use external sources to verify the information exposed in the news. Those based on style seek to find within the news signs of language that demonstrates subjectivity or disappointment.

The author of (Wang, n.d.) publishes a dataset called "LIAR" which brings together in total twelve thousand eight hundred fragments of declarations of the page "POLIFACT.COM" manually tagged; being one of the largest public datasets in this topic, with which you can do fact checking analysis and allow automation studies to detect false news. In addition, the study used neural networks to demonstrate that by combining meta-data with text, a great improvement in the detection of false news is achieved.

The study done in (Shao, Ciampaglia, Varol, Flammini, & Menczer, 2017) makes an analysis of how bots have been used to spread false news on social networks like twitter and facebook.

In (Bajaj, n.d.) they present a study of Deep Learning using natural language processing for the detection of false news; thus, different models are presented, and an assessment is made of which may be the best option to obtain adequate results.

In (Farajtabar et al., N.d.) present a framework for the detection of false news combining learning and a model of network activities providing the possibility of doing a real-time analysis in social networks such as twitter.

3 Methodology

In the study carried out natural language processing (PLN) is used as a Python computational tool; This programming language uses different libraries and platforms, among them its PANDAS natural language processing library (Python Data Analysis Library) which is an open source library with BSD license that provides data structures and data analysis tools . Additionally, NLTK was used, which is a set of libraries and programs oriented to natural language processing and Scikit-learn which is a specialized machine learning library for classification, regression and clustering. The three libraries mentioned above have been designed to operate in conjunction with the other Numpy and Scipy libraries which were also included in the program.

To obtain news for the study, a public data set located in a github repository was used https://github.com/GeorgeMcIntire/fake_real_news_dataset compiled in equal parts for ten thousand five hundred and fifty-eight (10558) news items collected in total between the years 2015 and 2017 written in English with their title, full text and false or true label which were taken from different media, making scrapping processes in news web portals for half of real news and news from a published dataset in Kaggle conformed only by false news.

So once having the dataset, the methodology consisted of three fundamental stages; the pre-processing that involved transforming the dataset from a .csv file to a Python object belonging to Pandas; a data frame to be able to deal with it efficiently. Subsequently, for processing, the data was changed so that the first half of the data with false label and the second half with a true label were not simply what would cause impartiality when applying the machine learning methods. Once this is done, groups of data are taken to make training and test sets with which tokenisation algorithms are executed so that the result is processed by the Multinomial Naive Bayes algorithm of the Scikit-Learn package and finally an array was made in analysis. of confusion to make analysis of the results obtained.

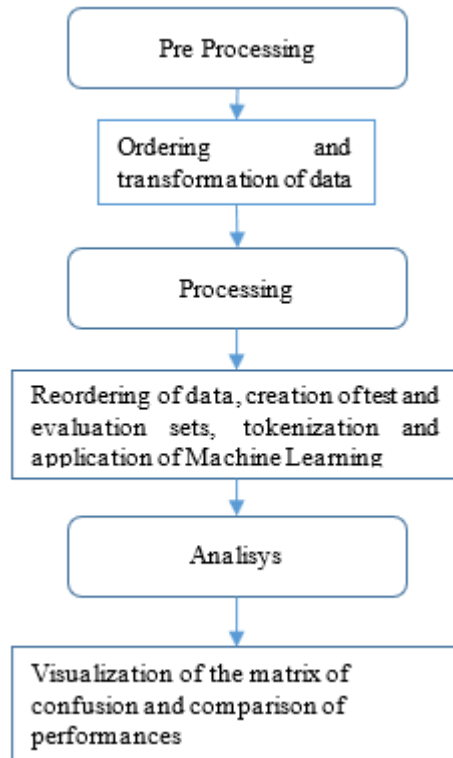


Fig. 1. Source: Own design.

4 Design

To begin with the processing of the data, it was necessary to use the `read_csv()` function of the Pandas library, passing the path of the file in which the .csv file is located, which converts to the Data Frame format. For the creation of the test and training sets, the `train_test_split()` function of the sklearn library was used, which takes as parameters the column with which the learning will be done, the type of classification that must be determined, the size with which will be the test set and a random to scramble the data.

Subsequently, sets "bags" of features are gathered, which are words or subsets of words with which you can extract the frequencies that have the word within the paragraphs belonging to the news texts with two different functions `CountVectorizer()`, but first it is necessary to do a new cleaning, since at the time of applying machine learning one looks for to see a relation between the veracity of the news and the words that more frequently appear in this one; as is logical there will be many occurrences of "stop words" is words like "that, in, on" these words that serve as connectors

and to give structure to the sentences but semantically does not have a great meaning, so it becomes necessary to get rid of those "stop words" and then if you can proceed to build structures made up of all the words that are part of the news.

Having the sets of words conformed, the NaiveBayes () function is sent as arguments so that it makes the process of determining from the word bag and the training sets if the news should be classified as false or true and subsequently an open source function is used to graph a confusion matrix in which the main diagonal shows the quantity of correctly classified news and the ones that are not seen outside the diagonal.

5 Implementation

The first thing that was done so that the program functions correctly is to import the necessary libraries so that all the functions used are recognized by the interpreter:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
import pickle
import nltk
import numpy as np
import matplotlib.pyplot as plt
import itertools
```

The file is then imported into a DataFrame from the Pandas library and formatted to make it easy to manipulate the data using the following commands.

```
features =
pd.read_csv("fake_or_real_news.csv",usecols=['text', 'label'])
```

Thus, the parts that we will be using for this study of the news, the text and its actual classification have been stored in features; First, mixing them is done to avoid that the classification is affected by the order of the news. These two columns are separated and used to create the training and test sets.

```
features.sample(frac=1)
```

6

```
trainig_set = features.text[:1900]
label_train = features.label[:1900]
test_set = features.text[1900:]
label_test = features.label[1900:]
```

The following commands make the word arrays to be generated do not contain stop words.

```
count_vectorizer = CountVectorizer(stop_words='english')
tfidf_vectorizer = TfidfVectorizer(stop_words='english',
max_df=0.7)
```

Then other counVectorizer functions are used to do a tokenization and frequency count of the tokens and the result is put into matrices made up of the tokens of the test set and the evaluation set.

```
count_train = count_vectorizer.fit_transform(trainig_set)
count_test = count_vectorizer.transform(test_set)
```

To contrast, another way of counting the frequency of the tokens is used and again applied to the test and training sets.

```
tfidf_train = tfidf_vectorizer.fit_transform(trainig_set)
tfidf_test = tfidf_vectorizer.transform(test_set)
```

The algorithm chosen for the automatic learning process was NaiveBayes which works with conditional probability to determine the relationship of one of the tokens with the truth of the news. Thus, the algorithm was run using both forms of tokenization previously named

```
clf = MultinomialNB()
clf.fit(tfidf_train, label_train)
pred = clf.predict(tfidf_test)
score = accuracy_score(label_test, pred)
```

Once the score is set, which is the percentage value of the degree of certainty that the algorithm had when making the classification. It was displayed using the following command:

```
print("certeza: %0.3f" % score)
```

Later an open source function was used to graph a confusion matrix and to visualize the results of the process.

```
cm = confusion_matrix(label_test, pred, labels=['FAKE',
'REAL'])
```

```
plot_confusion_matrix(cm, classes=['Falas', 'Reales'])
```

6 Discussion and results analysis

After running the Naive Bayes algorithm with the two forms of tokenization CountVectorizer and TfidfVectorizer the following percentages of certainty were obtained:

- Count Vectorizer: certainty: 0.881
- TfidfVectorizer: certainty 0.848

Likewise, the confusion matrices were plotted with the results of both classifications, producing the following results:

TfidfVectorizer:

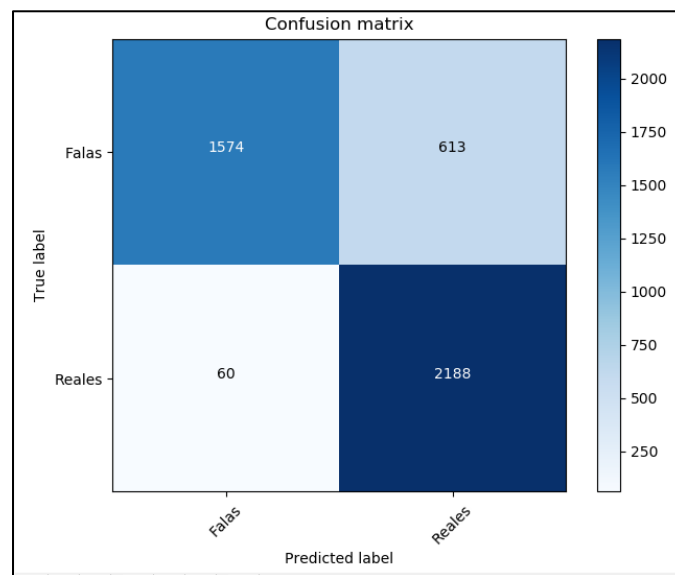


Fig. 2. Source: Own design.

Count Vectorizer:

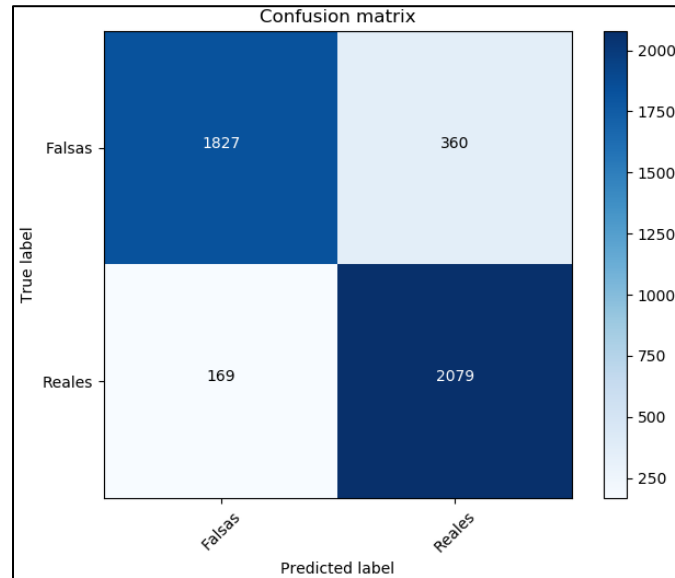


Fig. 3. Source: Own design.

This shows that it was more effective to use CountVectorizer as a classification method, since it successfully classified 89.3% of the news correctly classified as false 1827 and as true 2079.

The model used, however, proves not to have the same effectiveness as others, since in (Chiu, Gokcen, Wang, & Yan, nd) for example they use a model based on Support Vector Machines and achieve an average success rate of 95% and in (Chaudhry, Baker, & Thun-Hohenstein, nd) make an approximation using deep neural networks and achieve a certainty of up to 97.3% in the classification process.

7 Conclusions

Addressing an objective such as the classification of news is a complex task even using a standard procedure of classification of texts, since the news has a large number of characteristics that can be evaluated and to achieve a certainty greater than 95% it is necessary to consider them.

References

1. Allcott, H., & Gentzkow, M. (2017). Social Media and Fake News in the 2016 Election. *Journal of Economic Perspectives—Volume, 31(2—Spring), 211–236.* <http://doi.org/10.1257/jep.31.2.211>

2. Bajaj, S. (n.d.). The Pope Has a New Baby! Fake News Detection Using Deep Learning. Retrieved from <https://web.stanford.edu/class/cs224n/reports/2710385.pdf>
3. Bourgonje, P., Schneider, J. M., & Rehm, G. (2017). From Clickbait to Fake News Detection: An Approach based on Detecting the Stance of Headlines to Articles, 84–89. Retrieved from <http://www.aclweb.org/anthology/W/W17/W17-4215.pdf>
4. Chaudhry, A. K., Baker, D., & Thun-Hohenstein, P. (n.d.). Stance Detection for the Fake News Challenge: Identifying Textual Relationships with Deep Neural Nets. Retrieved from <https://web.stanford.edu/class/cs224n/reports/2760230.pdf>
5. Chiu, J., Gokcen, A., Wang, W., & Yan, X. (n.d.). Classification of Fake and Real Articles Based on Support Vector Machines. Retrieved from <https://pdfs.semanticscholar.org/05be/61a7726916644e49e578378d4faa7aa5f4ce.pdf>
6. Farajtabar, M., Yang, J., Ye, X., Xu, H., Trivedi, R., Khalil, E., ... Zha, H. (n.d.). Fake News Mitigation via Point Process Based Intervention. Retrieved from <http://proceedings.mlr.press/v70/farajtabar17a/farajtabar17a.pdf>
7. Gu, L., Kropotov, V., & Yarochkin, F. (2017). How Propagandists Abuse the Internet and Manipulate the Public.
8. Lazer, D., Baum, M., Grinberg, N., Friedland, L., Joseph, K., Hobbs, W., & Mattsson, C. (2017). Combating Fake News : An Agenda for Research and Action Final report written by, (May).
9. Mauri, M., Jonathan, G., Tommaso, V., & Michele, M. (2017). A Fiel Guide To Fake News.
10. Rubin, V. L., Conroy, N. J., Chen, Y., & Cornwell, S. (2016). Fake News or Truth? Using Satirical Cues to Detect Potentially Misleading News. Retrieved from https://www.researchgate.net/profile/Victoria_Rubin/publication/301650504_Fake_News_or_Truth_Using_Satirical_Cues_to_Detect_Potentially_Misleading_News/links/571fed4c08aeaced788acd8e/Fake-News-or-Truth-Using-Satirical-Cues-to-Detect-Potentially-Misleading
11. Shao, C., Ciampaglia, G. L., Varol, O., Flammini, A., & Menczer, F. (2017). The spread of fake news by social bots, (July). <http://doi.org/1707.07592>
12. Shu, K., Wang, S., Sliva, A., Tang, J., & Liu, H. (2017). Fake News Detection on Social Media: A Data Mining Perspective. ACM SIGKDD Explorations Newsletter, 19.
13. Singh, V., Dasgupta, R., Sonagra, D., Raman, K., & Ghosh, I. (n.d.). Automated Fake News Detection Using Linguistic Analysis and Machine Learning. Retrieved from http://sbp-brims.org/2017/proceedings/papers/challenge_papers/AutomatedFakeNewsDetection.pdf
14. Wang, W. Y. (n.d.). Liar, Liar Pants on Fire : A New Benchmark Dataset for Fake News Detection. Retrieved from <https://www.cs.ucsb.edu/~william/papers/acl2017.pdf>