Fake News Detection

Aviv Farag

College of Computing & Informatics, Drexel University, Philadelphia, PA 19104, USA

Abstract - Fake news is article that contains misleading information aiming to change other's opinion, thus gaining power (political, business, etc.). In this study, I propose a machine learning model based on Naive Bayes and implemented in PySpark for classifying document into two groups of news: reliable and fake. Data cleaning, stop words removing, and counting terms frequency were all implemented to generate the training and test datasets. Results of the ML model were compared to the baseline using confusion matrix, and revealed a great improvement in accuracy and F1 score.

Keywords - Fake News, Classification, PySpark, Naive Bayes, NLTK

GitHub - https://github.com/avivfaraj/DSCI632-Project

---

1 Introduction

Fake news is news articles that contain false facts with the aim of manipulating people's perception on a given subject. Another definition is "low quality news with intentionally false information" [1]. Fake news can be spread easily across social media because of their low production cost that contribute to profitability, and the format of the news which is small pieces of information [2].

Moreover, bots are being employed on social media and have a great impact on spreading fake news since they play an important role in amplifying fake news in the very early moments of a post, and they also target influential users using replies and mentions [3]. This strategy is often used by groups of interest in order to affect a country's election [4]. It can also be utilized by other groups or individuals in the business section in order to affect the reputation of others [4].

Finally, everyone can post, reply and share on social media and combined with the idea that "everyone now has their own truth, which is based on their personal knowledge and experience and not much else" [4] it is another reason for the wide spread of fake news across social media. This study aims to utilize PySpark machine learning libraries in order to classify instances into either reliable or not.

2 Dataset

The news dataset from Kaggle[5] contains three attributes and the target column as shown in table 1.

There are 20,800 rows, each of which describes one instance of an article. Also, there are 4194 unique authors in this dataset.

Column	Description
author	The writer of the article
title	The title of the article
text	Content of the article
label	Target: 0 - reliable, 1 - fake

Table 1: Dataset description

A sample of the dataset is shown in figure 1. Several fields shows three dots (...) which means that the value is much larger than the width of the column. Moreover, several rows contain the title within the text such as in the first instance (index 0). Finally, several instances are missing some attributes. For example, author is missing in both rows 6,8.

index	title	author	text	label
0	House Dem Aide: W...	Darrell Lucus	House Dem Aide: W...	1
1	FLYNN: Hillary Cl...	Daniel J. Flynn	Ever get the feel...	0
2	Why the Truth Mig...	Consortiumnews.com	Why the Truth Mig...	1
3	15 Civilians Kill...	Jessica Purkiss	Videos 15 Civilia...	1
4	Iranian woman jai...	Howard Portnoy	Print \nAn Irania...	1
5	Jackie Mason: Hol...	Daniel Nussbaum	In these trying t...	0
6	Life: Life Of Lux...	NaN	Ever wonder how B...	1
7	Benoît Hamon Wins...	Alissa J. Rubin	PARIS — France...	0
8	Excerpts From a D...	NaN	Donald J. Trump i...	0
9	A Back-Channel Pl...	Megan Twohey and ...	A week before Mic...	0

Figure 1: Data Samples

3 Exploratory Data Analysis

Understanding the data is a crucial step in building a machine learning model. In this section we are going to explore the data set in order to get some insights about it. We will first check whether the target column (label) is balanced, then we will explore which attributes are missing and their correlation with the target column.

3.1 Class Balance

A data set that contains significantly more instances of one class than other classes is considered to be imbalanced. In such case, a machine learning algorithm might tend toward the major class. Therefore, our first step is examining class balance as shown below:

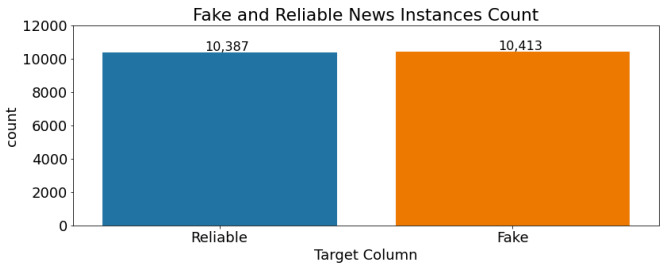

Figure 2: Target column distribution

The figure above shows the distribution of the target column. There are 10,387 (49.93%) reliable articles and 10,413 (50.07%) unreliable articles. Therefore, our data set is balanced.

3.2 Missing Values

Missing values could lead to several errors that might lead to either termination of the program, or unreliable results. Therefore, it is required to identify rows that are missing attributes as well as determine a proper way to deal with them.

There are 1957 rows that are missing the author attribute, 1931 are also labeled as fake. Title is missing in 558 rows, all of them have both author and text and are labeled as fake. Finally, text is missing in 39 instances, each of which is also missing the author attribute, and is labeled as fake.

To sum up, more than 98% of the rows that are missing one or more attribute are labeled as fake. This will be helpful in creating a good baseline for the machine learning model. Additionally, all 39 rows that are missing both the text and the author will be deleted because there is no content to process in those instances.

4 Methodology

4.1 Baseline

A baseline is a guess that could be done easily by anyone, and which the machine learning model is trying to improve. Based on the EDA, one can label all rows that have at least one missing attribute as fake. Otherwise, it is labeled as reliable. In this case, precision is great since reliable was guessed for the majority of rows in our dataset. However, the recall is low (0.56), so the accuracy score is 0.61 and F1 score is 0.77 as can be seen in figure 3. Our goal is to design a pipeline and utilize a machine learning algorithm to classify news into either reliable or unreliable, and achieve results that are better than the baseline.

```
Accuracy: 0.618
Recall: 0.567
Precision: 0.997
F1 Score: 0.723
+----------------+-----+----+
|label\prediction|  0.0| 1.0|
+----------------+-----+----+
|             0.0|10361|  26|
|             1.0| 7924|2489|
+----------------+-----+----+
```

Figure 3: Baseline

4.2 Data Pre-processing

Target class is balanced, but there are several rows which are missing at least one attribute. Therefore, we must clean the data before developing pipeline. The process is described in figure 4 shown below.

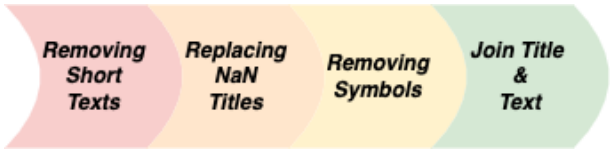

Figure 4: Cleaning dataset

First stage is deleting all rows in which the text is shorter than 60 characters. The goal is to classify news articles, so instances with little to none content are not part of this study.

Next stage deal with rows that do not have a title. In this case, "NaN" string [1] is replaced with white-space.

[1] The dataset was read with Pandas package and was converted to PySpark because PySpark modules didn't read it correctly. For that reason, null values were marked as NaN in Pandas dataframe and were converted to NaN strings in PySpark dataframe.

As long as there is some content, it is valuable for the analysis.

Third, symbols (e.g. $,%,#) were removed from every title and text in each row. Later on, text will be converted to numerical values, so it is an important step. It prevents those characters from expanding the dimension of the variables, thus affecting machine learning model.

Finally, a new column is created, and is composed of both the title and the text. There are some rows in which the text already includes the title. In such cases, the text was taken.

After cleaning the data, there are 9,825 unreliable and 10,385 reliable articles, so the target class remains balanced (48.7% and 51.3% respectively).

4.3 Pipeline

Document classification requires converting sentences to words, and then to numerical values. The process contains 4 stages as described in figure 5. They are being executed one after the other in order to represent each word as a number (frequency).

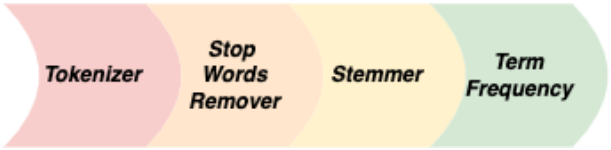

Figure 5: Pipeline Stages

Tokenizer is the first stage in which the text is being split into a list of words. It uses white-space as the splitter, and also convert upper case letters to lower case.

Then, stop words were removed from each list of words. Stop words are words that appear many times in a document, and have no significance for the analysis. The list of stop words was obtained by using the NLTK[2] package in Python.

The following stage is the Stemmer that converts every word to its stem. This is a custom transformer[6] that utilizes PorterStemmer instance from NLTK package. This step reduces the dimension of the features[3] column. For instance, the words "Playing", "Plays", "Played", and "Play" are all converted to "play".

Final stage is the Term Frequency (TF) in which the program count the frequency of every term in a document. The result of this stage is our features column for the machine learning model.

5 ML Model

The dataset was split 70% training and 30% testing. After splitting the data, the designed pipeline was fitted on the train data, and transformed both training and test sets in order to compute the features column.

In this study, Naive Bayes Classifier was utilized in order to classify news. There are three types of Naive Bayes: Multinomial, Bernoulli, and Gaussian.

The first one can handle finite discrete data, the second can only handle binary (0,1) vectors, and the last one can handle continuous data [4]. The features column contains numbers corresponding to the frequency of every term in the dictionary, and therefore Multinomial Naive Bayes will be utilized.

6 Results

Testing results as well as the confusion matrix are shown in the figure below:

```
Accuracy: 0.921
Recall: 0.885
Precision: 0.973
F1 Score: 0.927
+----------------+----+----+
|label\prediction| 0.0| 1.0|
+----------------+----+----+
|             0.0|2985|  84|
|             1.0| 387|2540|
+----------------+----+----+
```

Figure 6: Testing Naive Bayes Classifier

Precision was down by 0.2 compare to the baseline. However, recall was significantly improved from 0.56 to 0.885, and therefore both accuracy and F1 score are much better than in the baseline. To be more specific, Accuracy jumped from 0.618 to 0.921 and F1 score changed from 0.723 to 0.927. F1 score can be viewed as the harmonic mean of precision and recall, thus proving this model's reliability.

[2] For more information: https://www.nltk.org

[3] Features column in PySpark is the column that contains all attributes (excluding target) for machine learning.

[4] For more information: PySpark - Naive Bayes Classifier

7 Conclusions

Recently, fake news has become ubiquitous in the media, leading to increasing levels of distrust amongst consumers. To help mitigate the spread of fake news, this paper presents a pipeline to classify news articles. This program was developed in PySpark, the optimal platform for processing big data. Implementation of this program will greatly decrease the prevalence of fake news, thus improving news quality and trust amongst consumers.

References

[1] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, Huan Liu, *Fake News Detection on Social Media: A Data Mining Perspective*. ACM SIGKDD Explorations News letter, Volume 19, Issue 1 June 2017, pp 22–36. https://doi.org/10.1145/3137597.3137600

[2] Allcott, Hunt, and Matthew Gentzkow. 2017. *Social Media and Fake News in the 2016 Election*. Journal of Economic Perspectives, 31: 211-36. https://doi.org/10.1257/jep.31.2.211

[3] Shao, Chengcheng & Ciampaglia, Giovanni & Varol, Onur & Flammini, Alessandro & Menczer, Filippo. (2017). *The spread of fake news by social bots*.

[4] Del Vicario, Michela and Bessi, Alessandro and Zollo, Fabiana and Petroni, Fabio and Scala, Antonio and Caldarelli, Guido and Stanley, H. Eugene and Quattrociocchi, Walter, *The spreading of misinformation online*. National Academy of Sciences, Volume 113, Number 3, Year 2016, Pages 554-559. https://doi.org/10.1073/pnas.1517441113

[5] Community Prediction Competition, *Fake News*, Kaggle.

[6] Clare S. Y. Huang, *Custom Transformer that can be fitted into Pipeline* https://csyhuang.github.io/2020/08/01/custom-transformer/