Fake News Detection

June 10, 2021

Problem Definition

Social media is a vast pool of content, and among all the content available for users to access, the news is an element that is accessed most frequently. This news can be posted by politicians, news channels, newspaper websites, or even common civilians. Our goal is to identify fake news from social media.

Our main focus from this project, more specifically from this dataset is to know whether the headline of a news article is important for detecting fake news. Our aim is to follow some supervised learning approach and by evaluating the result we can arrive at some conclusions that will be helpful to understand the importance of the headline. This project aims to provide a systematic evaluation of the machine learning algorithm for fake news detection.

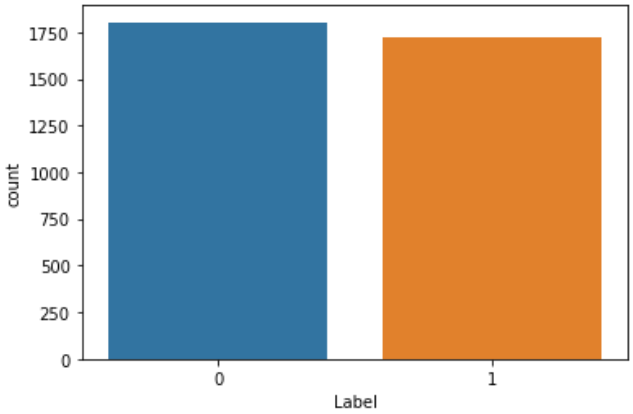
Goals

- Merge Headlines with Body. Implement Supervised Learning algorithms on both Body and Newstext.
- Then analyze the performance of the algorithms to come up with a hypothesis because of our approach.

Dataset Description

- The actual dataset, collected from **Kaggle,** had a dimension of (4009, 4).
- Number of attributes: 4 **[URL, Headline, Body, Label]**
- After pre-processing, the dimension remains (3522, 3).
- Label '**1**' represents the "**Real**" & '**0**' represents the "**Fake**"

- And the distribution of **Real** & **Fake** is **1719 : 1803**



- After split into Train-Test, number of instances: Train(2817) + Test(705)

Algorithms

- Our dataset is very small comparatively. But after vectorizing this dataset, the dimension of the dataset increases drastically. Because of this, we are using **Support Vector Machine** and **Multinomial Naive Bayes** which perform better on small but complex datasets.

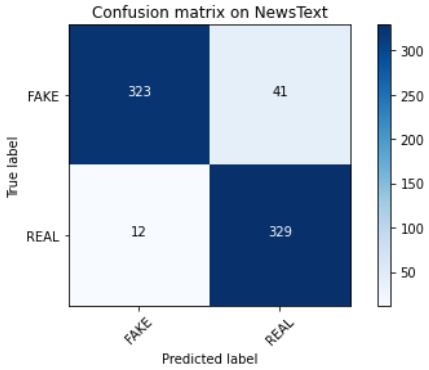
Our Proposed Approach

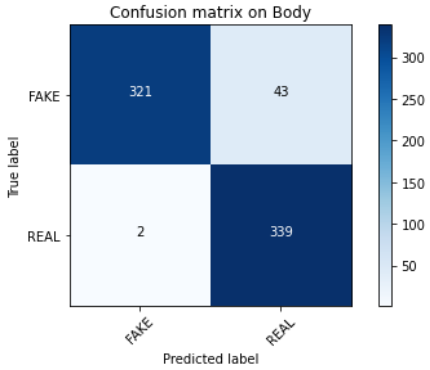
- The redundant attribute was dropped. [**URL Column**].
- 21 NULL values of [**Body**] have been removed from the Dataset.
- Duplicates (of **Headline** & **Body**) were removed from Dataset and the number of available instances remains 3522.
- A new column [**NewsText**] was generated combining two columns [**Headline + Body**].
- Then column [**Body** & **NewsText**] was converted into vectors using **TF-IDF** vectorizer.
- Then vectorized features were split into two parts (Train and Test).
    - The train set was used for training these models, which contains matrix dimensions of (2817, 47532) for **Body** and (2817, 47532) for **NewsText.**
    - The test set for evaluating the performance of these models, which contains matrix dimensions of (705, 47532) for **Body** and (705, 47532) for **NewsText.**
- Then the training set was fed to two classification algorithms to generate output.

Result/Comparison Metrics

- **Result analysis of MNB**

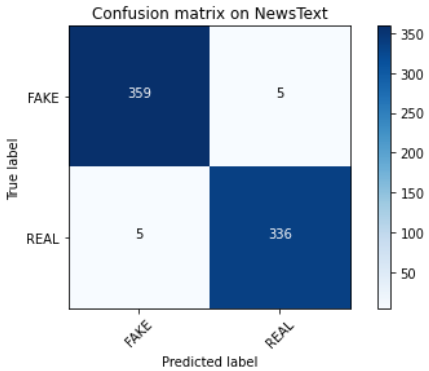
Score of Multinomial Naive Bayes		
	Accuracy	**F1 Score**
Body	93.61%	93.77%
NewsText	92.48%	92.54%

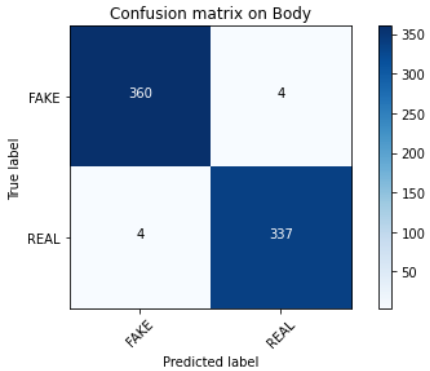




- **Result analysis of SVM**

Score of Support Vector Machine		
	Accuracy	**F1 Score**
Body	98.86%	98.88%
NewsText	98.58%	98.53%





Conclusion

We have implemented **SVM** and **MNB** on both **Body** and **NewsText** to verify which one is performing better. By analyzing the performance matrices, we can see that both algorithms are performing better on **Body** than **NewsText (**which is the combination of “Body” and “Headline” columns**).** So, the merging Headline with Body makes it worse than just Body. Now the question is why is this performing worse.

Our hypothesis is that The **Headline** contains the most significant words of News most of the time. So, when we merge the **Headline** with the **Body,** the frequency of those words increases. After applying the **TF-IDF** vectorizer, the weight of those significant words decreases because of higher frequency. This could be one of the reasons for performing badly on NewsText