Dataminer: A novel Fake News detection model

Renato Cordeiro
Computer Engineering
San Jose State University
San Jose, USA
renato.silveiracordeiro@sjsu.edu

Abhinaya Yelipeddi
Computer Engineering
San Jose State University
San Jose, USA
abhi.yelipeddi@sjsu.edu

***Abstract*—Fake news is not something new but attracted world attention due to its possible impact in the last US presidential election. The use of social media as the main source of information for most of young people created a new paradigm where the flow of information is free and fast than ever. Ability to detect legit from fake articles is not something that everyone possess, creating a need for auxiliary tools. This paper proposes a novel fake news detection model based on 6 features and achieved a accuracy of 77.78%.**

Keywords—fake, news, veracity, text, content, headline, authors, trust, machine learning, model, detect, bias

I. Introduction

Fake news become an intensive area of research since its allegedly impact on the 2016 US presidential election. The rise of social media as one of the main source of news for people [1] had removed the fact check process that articles needed to go through when published by news companies. Without intermediaries, news have gained a capacity to spread faster than ever. Unfortunately most of people don't have the time or capacity to evaluate if an article is legit or not. Therefore, an automatic model that can help readers identify fake news from legit is much needed. In the next sections we present a novel approach where we apply a structured process to analyze fake news using an amalgamation of independent feature models. This paper is structured as follow: section II sets the definition of fake news used on this paper; section III presents the approach used and the features analyzed, section IV describes that datasets used; section V presents the model results; section VI proposes areas of improvement for future work and section VII describes the work division between the authors.

II. Fake News Definition

The first step in trying to detect fake news is to actually define what we understand as fake news. Although that term is now widely spread and everyone seems to understand it, its meaning can vary from person to person. For instance, an article that has claims but didn't show the proof or the basis for that claim is fake news? Or an article that is mostly true but has some false parts should be considered fake news? Therefore, a fake news definition is required in order to make sure that everyone is in the same page.

In this paper we use the same approach as [2] where the authors defined fake news as "*a news article that is intentionally and verifiably false*". But, unlike [2], we will consider satire articles as fake news.

III. Fake News Detection

Differentiate fake news from legit ones is not an easy task, even for humans. So many variables can contribute to the veracity of an article and therefore its important that such problem is tackled using a sound methodological approach. We propose a model that is based on intrinsic characteristics, known as features, of a text. It takes in account both the text content (headline, body) as well as its author. Fig. 1 below illustrates all features that we believe can provide useful information that can be extracted in order to assess the veracity of a text.

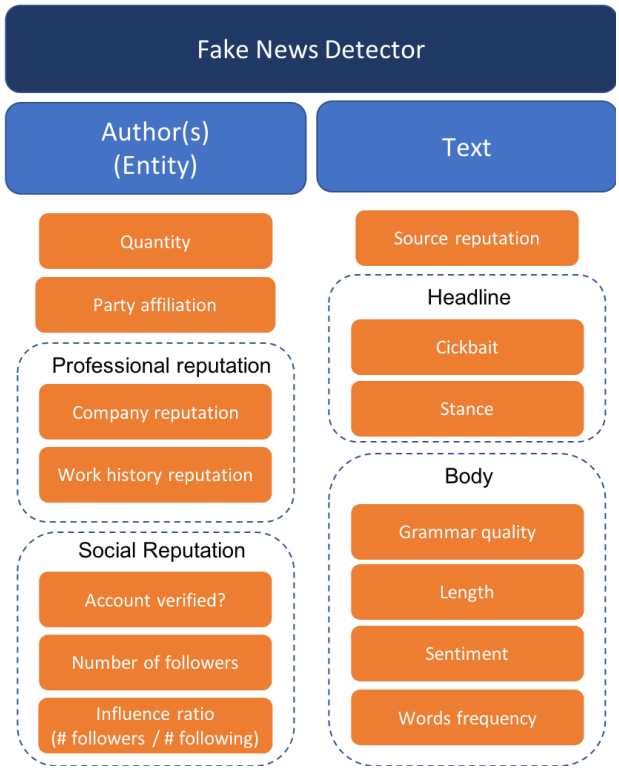

Fig. 1. Features of the Fake News Detector model

This paper analyses the features mentioned in Fig. 1 except the Professional and Social reputation ones since they would require specific datasets and third-party integration that were not available at the moment. For all the other features, we followed a 2-step process: first we did a high level analysis to identify possible correlation between the feature and the text veracity. In that step we used the entire dataset available. If a possible correlation was identified, we moved to the second step where we applied a categorization machine learning model in 80% of the dataset (training data) and tested its accuracy in the remaining 20% (test data).

A. *Entity - Party Affiliation*

Political bias plays an important role in identifying fake news, especially when the dataset is more inclined to political views and election campaigns. Political party affiliations directly correlate to perception of fake news. According to Gallup/Knight Foundation survey, Americans believe that 39% of the news related to politics and elections is misinformation [3]. Hence identifying fake news based on political affiliation is the need of the hour.

To evaluate political affiliation on the dataset, we listed down the various political parties present in the dataset. From the list of political parties we chose the three most important political parties which cover 98% of the dataset. We pre-processed the dataset to include entries for the top three political parties. On the pre-processed data, we performed stemming and lemmatization using Porter Stemmer and WordNet Lemmatizer respectively. We used POS tagging to remove plurals. We also used Stop word removal and Rare word removal techniques. To extract feature vectors and not include numeric features, we used regular expression tokenizers. We also used word punctuation tokenizers and blank line tokenizers to remove punctuation marks and blank lines in the dataset for better feature extraction.

Our hypothesis was that speakers whose party affiliation is democrat, will speak mostly about the democrat agenda. Similarly speakers whose party affiliation is republican, will mostly speak about the republican agenda. Based on this hypotheses, using Count Vectorizer and TF-IDF Vectorizer, we extracted the mostly used words from democrats and republicans respectively. Then we compared whether the headlines of the speakers who are democrats fall into the democrat list of features. Similarly, we compared the same for republican speakers. If the headlines does not fall into the specific buckets, we conclude the news is fake.

We used Multinomial NB, Logistic Regression, Linear SVM and Random Forest classifiers to classify the test data set. Logistic Regression provided the best accuracy among them. The confusion matrix from the Logistical Regression classification on test dataset is show below (Fig. 2):

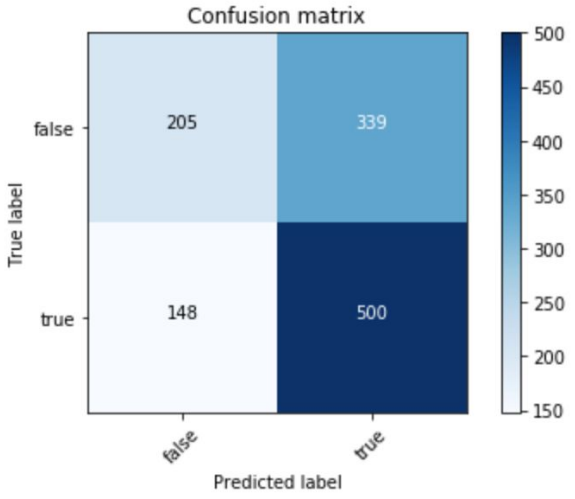

Fig. 2. Confusion matrix with Logistic Regression classifier

B. *Entity - Quantity*

Our hypothesis was that fake news articles, since it is created with the purpose to deceive the reader, would face more difficulties to be created as the number of people involved with it grew. After all, assuming that humans are generally good, it should be more difficult to get people together to do an unethical thing than the opposite. Therefore, using the Politifact dataset, we analysed the correlation between number of authors and article veracity. The Table I below illustrates the number of true/fake articles by the number of authors.

TABLE I. Number of Articles By Veracity And Number of Authors

Number of authors	***Article veracity***	
	True	***Fake***
0 authors	13	85
1 author	36	24
2 or more author(s)	71	11

The numbers pointed in the direction that our hypothesis was true and therefore we continue to create a classification model relating number of authors with the text veracity. We used a Logistic Classification model using liblinear solver and C parameter equal to 1 using the training dataset. The test dataset on that model provided an accuracy of 84.21%.

C. *Text - Headline - Clickbait*

Online advertisement is an 88-billion dollar industry [4] and thus attracts lot of players. Companies and individuals that show ads are usually remunerated based on the number of clicks that the ad receives. The use of that metric (revenue per click) went beyond the ad industry and arrived in content creators' world which includes news companies, bloggers and

others. Now it is common to see writers and reporters having part of their compensation based on the number of clicks that their article (headlines) received. This led to the development of headlines with catchy words or that tries to deceive the user to click on it, using almost any means necessary. In some extremes, the headline doesn't even reflect what the article says (known as stance, which is analyzed in detail in the next subsection). That kind of headline is known as clickbait headlines. We analyzed if the presence of such headlines can indicate any higher (or lower) probability of the article being fake.

Using the Clickbait dataset curated by Chakraborty et al. we first preprocessed the text by removing non-alphanumeric characters and replacing numbers with the actual 'number' word. Note that we didn't removed the stop words, which is a fairly common step during preprocessing. We didn't do that because we wanted to also assess the stop words contribution. Then we augmented the data by found some latent variables: part of speech, number of contractions, number of stop words, percentage of stop words and percentage of contractions. Since we created some latent features by using other existing features, it is important to check if we don't have features that are linear dependent on each other. To do that we analyzed the colinearity of all the features by using the heatmap shown below.

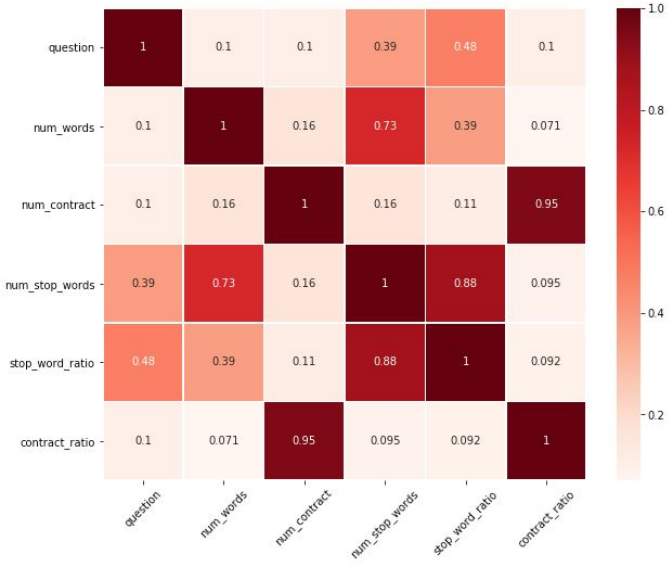

Fig. 3. Features collinearity heatmap

The features *contract_ratio* and *num_contract* showed a high collinearity. That was also observed for the pair (*stop_word_ratio*, *num_stop_words*) and (*num_stop_words*, *num_words*). Therefore, we decided to remove *num_stop_words* and *num_contract* from our dataset.

Logistic regression was applied to the remaining features, generating an accuracy of 97.68%.

D. Text - Headline - Stance

Stance detection attempts to identify if the headline of the text indeed reflects what the text body talks about. In order to evaluate that we decided to calculate the cosine distance between the text headline and the body from the Politifact dataset. We started preprocessing the text by tokenizing both the headline and body, removing non alphabetic characters and stemming (using Porter stemmer) the tokens. Then we calculated the TF-IDF of that collection of documents and calculated the cosine distance between the header the body. The result can be seen in Fig. 4 below. To better illustrate a possible (or not) relation between the cosine distance and text veracity, we performed linear regression in the data which resulted in a very low R2 score (0.002). Therefore we decided to not include that feature as a component of our final model.

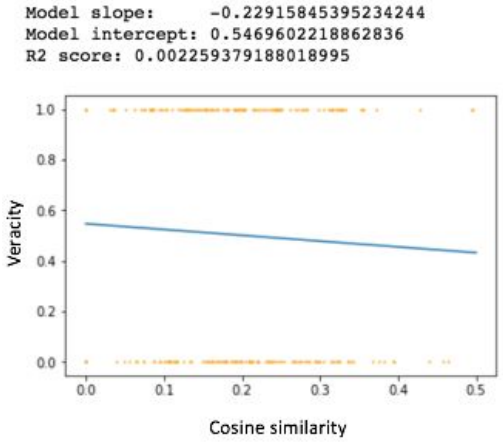

Fig. 4. Cosine similarity between the text headline and body

E. Text - Body - Grammar Quality

From day to day experience of seeing fake products and news, one can note that those not rarely presents some gross visual and grammar errors. Therefore, we decided to analyze if spelling errors could indeed be an useful feature to assess a text veracity. As always, we started by performing a high level analysis to see if we see any possible relation between the number of grammar errors and the text veracity. We used the whole Politifact dataset, passing both headline and body text in a grammar checker (Python Language Check) and plotting the result against the text veracity. The result can be seen in Fig. 5 below.

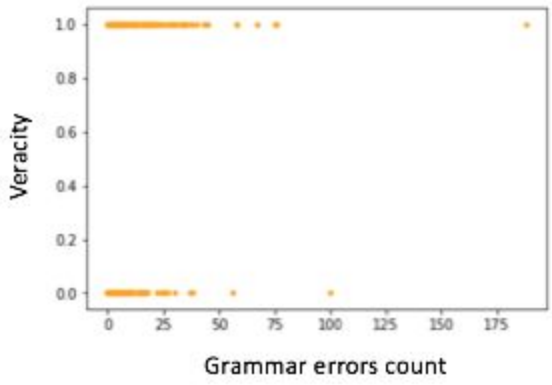

Fig. 5. Grammar errors and text veracity

We can clearly see that, except for a few outliers, both true and fake articles have a some grammar errors and none of them really tends to have more (or less) than the other one. Therefore, we decided to not have grammar errors in the final model.

F. *Text - Body - Length*

A simple but interesting feature that we decided to analyze was the length of the article. The reasoning was that it could be more difficult to create long fake news that can still pass to the reader as a true article. Since one of the objective of fake news is to get spread as far and fast as possible, having text that can indicate that the text is not legit could jeopardize that. To investigate such hypothesis we used the Politifact dataset and calculated the length of each article. To get an overall idea of possible correlation we decided to plot the length of the article against its veracity. The result can be seen below.

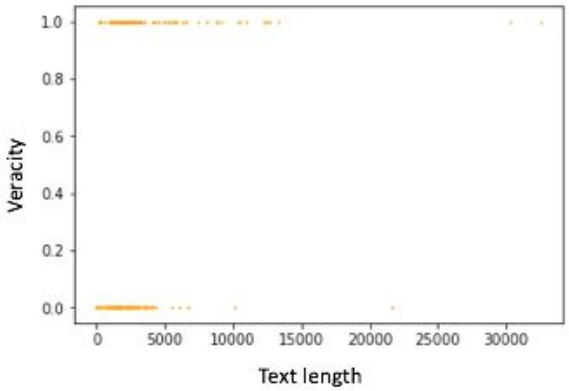

Fig. 6. Cosine similarity between the text headline and body

Articles up to around 5,000 characters appears to be indifferent regarding veracity, but above that threshold, legit articles have a predominance. Therefore we decided to apply Logistic Regression, which gave us an accuracy of 71.05%.

G. *Text - Body - Sentiment*

Sentiment is usually one of the first analysis done on NLP projects and we expected it to indicate that negative articles would have more probability of being fake. The idea was that negative text would be more viral and thus pursuit by fake news writers. We used the Politifact dataset and applied Vader Sentiment Analyzer in the articles' text. We then plotted the result which showed almost no correlation between sentiment and veracity. Fig. 7 below illustrates that.

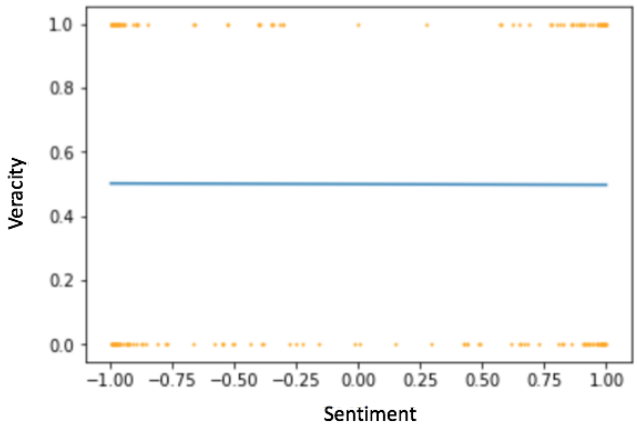

Fig. 7. Veracity versus sentiment of Politifact dataset

That result indeed surprised us and therefore we decided to perform the same analysis on a different dataset (Liar Liar). The outcome was almost the same as can be seen below.

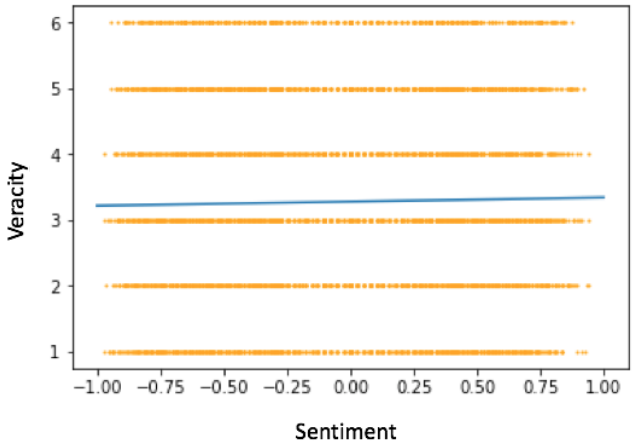

Fig. 8. Veracity versus sentiment of Politifact dataset

H. *Text - Body - Words Frequency*

There are a set of words that are frequently used in fake news. Identifying those words and their frequency will help in differentiating factual from fake. From the dataset, we used the statement of the speakers and pre-processed the data using tools such as stop-words removal, punctuation removal,

stemming, lemmatization and null value removal. Later, we used CountVectorizer and TF-IDF Vectorizer to identify features from the speaker's statement.

We used Multinomial NB, Logistic Regression, Linear SVM and Random Forest classifiers to classify the test data set. Multinomial NB with TF-IDF Vectorizer provided the best accuracy among them. The confusion matrix from the Multinomial NB classification on test dataset is show below (Fig. 9):

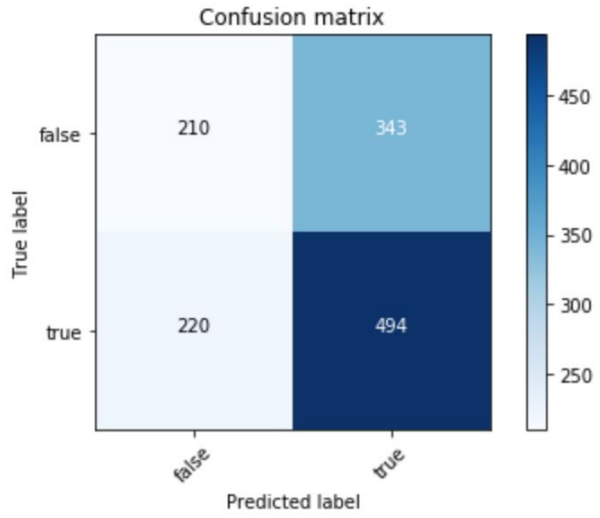

Fig. 9. Confusion matrix of word frequency

I. *Text - Body - Source reputation*

Probably one of the most important feature to identify fake news is the reputation of its source. We obtained a Politifact curated list of website domains known to host fake news. We then created a penalization score for article hosted on those domains. This is the only feature that we used a simple rule instead of a machine learning categorization model.

J. *Model consolidation*

After analyzing 9 features, we discovered that 3 (stance, grammar quality and sentiment) didn't show correlation with an article veracity. The remaining 6 features though are important components and we performed amalgamation on them in order to have a final consolidated model. For each of the prediction model output we applied a weight and the combined all them together as shown in the Formula 1 below:

$$F(A) = \sum w_i * p_i \quad (1)$$

where: *F(A)* is the Fake news probability of an article A; p_i is probability the fake news model of feature *i* and w_i is the weight applied to reflect the importance of that feature in the overall model. The weight applied to each feature model was the normalized accuracy of the respective model. We did that so it could reflects how accurate that component result really is.

The final result is a number between 0 and 1, indicating the probability of the article in being fake. We decided to use a cut-off of 0.5, so any article that receives a score above that will be classified as fake. Fig. 10 below shows the Python code used for the model consolidation.

```
def isFakeNews(text, headline="", numAuthors = 0, source = "", party =""):
    accur = [0.84, 0.56, 0.98, 0.71, 0.6, 1] # using the (normalized) accuracy as weigths
    w = [float(i)/sum(accur) for i in accur]
    sumW = 0
    prob = []
    prob.append(w[0] * DATAMINERS_getAuthorScore(numAuthors))
    sumW += w[0]
    if ( (headline != "") & (party != "")):
        prob.append(w[1] * DATAMINERS_getPartyAffiliationScore(headline, party))
        sumW += w[1]
    if (headline != ""):
        prob.append(w[2] * DATAMINERS_getClickbaitScore(headline))
        sumW += w[2]
    prob.append(w[3] * DATAMINERS_getBodyLengthScore(len(text)))
    sumW += w[3]
    prob.append(w[4] * DATAMINERS_getWordFrequencyScore(text))
    sumW += w[4]
    if (party != ""):
        prob.append(w[5] * DATAMINERS_getSourceReputationScore(source))
        sumW += w[5]

    probTotal = sum(prob[0:len(prob)]) / sumW
    return probTotal
```

Fig. 10. Consolidated Fake News model code

IV. DATASETS

This project used 3 different fake news datasets in order to perform analysis on all the features mentioned in section III above. The datasets are described below.

A. *Liar Liar*

This dataset has more than 10,000 articles already labeled for its veracity. They used a range of veracity which included 'true', 'mostly-true', 'half-true', 'barely-true', 'false', 'pants-fire'. For high level analysis we used those discrete values but for classification purposes we considered 'barely-true', 'false', 'pants-fire' as fake news and the others as true news. The downside of the Liar Liar dataset is that the articles don't have headlines. Each article has the following details:

- label, statement, subject, speaker, speaker_job_title, state_info, party_affiliation, barely_true_counts, false_counts, half_true-counts, mostly_true_counts, pants_on_fire_counts, context

B. *Politifact*

Different from Liar Liar, this dataset has articles with headlines which allows us to work on features that were not possible with the previous dataset alone. It has 120 true articles and 120 fake articles, each one having the following information:

- authors, canonical_link, images, source, text, title, url

C. *Clickbait*

This dataset was compiled especifically for Clickbait analysis by Chakraborty et al. and provides a list of 327

domains known to host fake articles and others. Each domain in the list is classified in:

- fake news, impostor site, parody site, some fake stories

V. Results

The consolidated Fake News model was tested using the Politifact dataset and presented an accuracy of 56.67%. The respective confusion matrix is shown in Fig. 11 below.

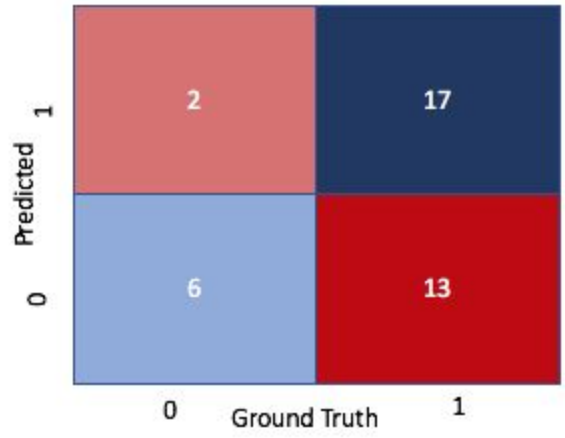

Fig. 11. Fake News model confusion matrix

A close look shows that False Negatives (articles that are fake but the model wrongly identified as legit) are the main responsible for the low accuracy. Since the model result a range between 0 and 1 and we used a cut-off value of 0.5, we decided to rerun the test dataset but now using a margin of error of 10 p.p. for the cut-off. So, if a prediction falls between 0.4 and 0.6 the model result would be ignored. The use of such margin of error increased the model accuracy to 77.78% and resulted in the confusion matrix can be seen in Fig. 12.

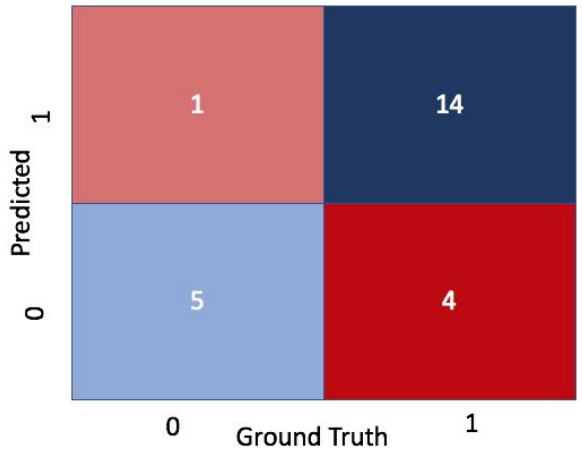

Fig. 12. Fake News model confusion matrix with margin of error of 10 p.p.

VI. Future Work

This study analysed 9 features although in section III we mentioned 14 possible factors. Therefore, a immediate future research could include the remaining 5 features not analyzed in this paper, which includes the professional and social reputation. Additional components could also be added. Another area of improvement is the optimization of the weights applied to the model of each feature. Although we are comfortable with our choice of using each model accuracy as its weight, we don't discard the possibility of better values. Lastly, the use of a bigger dataset, especially for testing, could help improve this model further.

VII. Division of Work

This project was developed by Renato Cordeiro and Abhinaya Yelipeddi, with each one being responsible for specific tasks as illustrated in the Table II below.

TABLE II. Tasks by Team Member

Task	***Member***	***Task***	***Member***
Paper structure, Sections I, II	Renato	Text - Body - Words Frequency	Abhinaya
Entity - Party Affiliation	Abhinaya	Text - Body - Source reputation	Renato
Entity - Quantity	Renato	Model consolidation	Renato
Text - Headline - Clickbait	Renato	Section IV	Renato
Text - Headline - Stance	Renato	Section V	Renato
Text - Body - Grammar Quality	Renato	Section VI	Renato
Text - Body - Length	Renato	Section VII	Renato
Text - Body - Sentiment	Renato	Acknowledgment & References	Abhinaya, Renato

Acknowledgment

The authors would like to acknowledge the strong contribution made by Dr. Ali Arsanjani in providing deep knowledge in Machine Learning and NLP as well as guidance in dataset finding and analysis approach, all of them fundamental to the development of this paper.

References

[1] "News Use Across Social Media Platforms 2017", Pew Research Center's Journalism Project, 2018. [Online]. Available: http://www.journalism.org/2017/09/07/news-use-across-social-media-platforms-2017. [Accessed: 08- Dec- 2018].

[2] K. Shu, A. Sliva, S. Wang, J. Tang and H. Liu, "Fake News Detection on Social Media", ACM SIGKDD Explorations Newsletter, vol. 19, no. 1, pp. 22-36, 2017.

[3] K. Foundation, "Americans' views of misinformation in the news and how to counteract it - Knight Foundation", Knight Foundation, 2018. [Online]. Available: https://www.knightfoundation.org/reports/americans-views-of-misinformation-in-the-news-and-how-to-counteract-it. [Accessed: 13- Dec- 2018].

[4] "Amazon Sets Its Sights on the $88 Billion Online Ad Market", Nytimes.com, 2018. [Online]. Available: https://www.nytimes.com/2018/09/03/business/media/amazon-digital-ads.html. [Accessed: 13- Dec- 2018].

[5] Bonzanini, M. (2016). *Mastering social media mining with Python:Acquire and analyze data from all corners of the social web with Python*. Birmingham: Packt Publishing.

[6] A. Patankar and J. Bose, "Bias Discovery in News Articles Using Word Vectors", 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), 2017.

[7] M. Bonzanini, Mastering social media mining with Python. Birmingham, UK: Packt Publishing, 2016.