Fake News Detection Using Logistic Regression, Sentiment Analysis and Web Scraping

Pranav Bharti[1], Mohak Bakshi[2], R.Annie Uthra* [3]
Department of CSE, SRM Institute of Science and Technology,
Kattankulatur,Chennai, India-603203
*[1]bpranav22@gmail.com, [2]mohakbaks@gmail.com,[3]annieuthra@gmail.com**

Abstract

In recent years information sharing through internet and social media have grown abundantly and it is hard to find the authenticity of the news. This paper aims to detect the fake news using machine learning and web scraping methodologies. The proposed methodology performs the fake news classification on the labeled dataset based on one of the Machine learning algorithms i.e. logistic regression. The proposed system also scrapes multiple news websites and checks for keywords mentioned in the input news article and calculate a percentage combining the results from the machine learning approach which uses the logistic regression algorithm and the web scraping approach. Also, a major factor to weigh in terms of fake news is the sentiment or the emotion behind the news article, the news article may be biased concerning a specific political propaganda, hence in favor of a side. But true news must be unbiased; it must be factual, not accusing or declarative. Thus, to inform the user about the bias in the news article, the proposed system performs sentimental analysis involving pre-processing, NLTK lemmatization, porter stemmer and uses naïve Bayesian algorithm and then presents the user with the sentiment behind the news article. The two above mentioned process will present the user with two outputs: the percentage of how accurate the news article combining the results from the logistic regression algorithm and web scraping approach and will output the sentiment behind the content through sentiment analysis. This will facilitate the user to take an informed decision about the nature of the news article given by the user.

Keywords:*Logistic regression, web scraping, sentiment analysis, NLTK lemmatization.*

1. Introduction

The increase in access to internet and the boom in social networking and media such as WhatsApp, Facebook, Instagram etc. made the access to the news information much easier, portable and faster. Often the general public with access to internet can now follow news, their articles of interest and much more, right on their fingertips, anywhere and anytime. News, or content, or media in general have a huge influence on the society, it is both capable to sway opinions, change mindsets as we say in the US elections in 2016 and thus, there arises a high chances that someone wishes to exploit this opportunity in their favor. Sometimes to achieve personal gains, mass-media may manipulate the information in different ways. This leads to producing of the news articles that are not completely true or even completely false. Hence, the core objective of fake news is to reshape or mold the public opinion on certain matters, mostly targeted at satisfying a personal agenda which often include political motives.

Furthermore, the feeling or sentiment behind the news article is a major factor to consider in terms of fake news; the news article may be skewed with respect to a specific political agenda, hence in favor of one hand. True news must be impartial; it must be factual, not accusatory or declarative.

The proposed system detects the fake news using machine learning algorithm, scraps multiple news websites and keyword checks mentioned in the input news article and calculates a percentage that combines both approach results. The proposed system also aims to perform sentimental analysis and thus present the user with the feeling behind the news article in order to inform the user about the bias in writing the news article. The above two processes present the user with two outputs: first, the percentage of how accurate the news article combining the results from the logistic regression and web scraping approach. Second, the sentiment behind the content through sentiment analysis. This will facilitate the user to take an informed decision about the nature of the news article given by the user.

2. Literature Survey

In an assortment of disciplines, including semantics and software and computer science engineering, counterfeit news has become a significant research subject. The authors clarify [1] how the problem is approached from the natural language processing perspective, with the objective of proposing a system to detect unauthentic information in news automatically. The fake news classifier [2] is constructed using logistic regression classifier, wherein the datasets were accumulated by BuzzFeed News for learning and testing of the system. A comprehensive tutorial-based approach is used for establishing the research and datasets were clearly listed, various detection strategies [3] were coalesced under an intensive framework for counterfeit news detection and state-of-the art patterns and models were employed. Fake News Tracker, a framework for counterfeit news comprehension and discovery [4] can naturally gather information for news pieces and social setting, which advantages further research of comprehension and anticipating counterfeit news with successful representation procedures. Content-based, source-based, and diffusion-based approaches [5] were presented. The work describes two opposite approaches and suggests an algorithmic solution synthesizing the main concerns. Also, raises awareness of the needs and opportunities of companies currently seeking to help automatically detect fake news through the provision of web services. The authors [6] provide a detailed analysis of the findings of the latest false news. This is characterized by the negative effect of online fake news and state-of - the-art detection methods. Many of these are focused on defining client, content, and background features that suggest misinformation. It has existing repositories that are used to classify fake news. Clickbait [7], draws in user and their interest with garish features or structures to click connects to expand income from promotions. The work breaks down the commonness of phony news given the advancement made conceivable by the rise of long-range informal communication locales in correspondence. The goal this work is to develop a solution that users can use to identify and remove pages that contain false and misleading information. The main objective of the work is to highlight frameworks [8], which models distributer news relations and client news connections at the same time for counterfeit news order, using a Tri-relationship Fake News (TriFN) installing structure. Auxiliary information using this framework is embedded for detection of fake news. The framework identifies three basic entities i.e. publishers, news pieces and social media users and the relationship between all of them brings out the additional information which is needed for detection of fake news.

The authors discover what data can be utilized by using online life information and what impediments web-based life information has [9]. Likewise, the paper audits the different endeavors to beat these restrictions. At last, proposals have been made on the most proficient method to best use online life information in understanding general feeling during decisions. The paper induces Distant Supervision in Suggestion Mining through Part-of-Speech (POS) Embedding [10]. POS tags are input to neural network. The sentences have been extracted from the articles of Wiki How and Wiki suggestion. The paper [11] presents the primary profound learning way to deal with opinion extraction in feeling mining. Opinion extraction is a subtask of assumption examination which comprises of recognizing sentiment focuses in the supposition content, for example recognizing the particular parts of an item or administration is either lauded or grumbled about by the conclusion holder. A 7-layer profound neural system to label each word as either aspect or non-aspect word in opinionated sentences has been proposed in the work. The paper talks about current State of Text Sentiment Analysis [12] from Opinion to Emotion Mining. A comparative analysis has been made of different strategies like SVM, KNN etc. The paper presented best in class strategies and enhancements for content assumption investigation. The authors propose [13] Machine Learning (ML) based counterfeit news discovery technique which, by consolidating news substance and social context highlights, beats existing strategies in the writing, expanding their effectively high precision by up to 4.8%. Besides execution of this strategy inside a Facebook Messenger chatbot has been done and has been approved it with a real-life application, acquiring a fake news recognition exactness of 81.7%. The paper [14] proposes a model of various leveled proliferation to assess the believability of news on the Micro blog, identifying sub-occasions to depict its nitty gritty angles inside a news occasion. Hence, a three-layer believability system of occurrences, sub-occasions, and messages may reflect it from various scales for a news occasion and uncover indispensable data for surveying validity. To accomplish the last appraisal result, the notoriety nature of every substance is spread on this system subsequent to interfacing these elements with their semantic and social affiliations. A vector portrayal of records [15] is utilized in the related research. This portrayal of the vector is then given for additional processing to the algorithm. The work means to plan vectors that can deal with the highlights of phony news before further handling utilizing Indonesian language through language calculations. As per the vector space model, counterfeit news and unique news are spoken to. Vector model blend of recurrence term, opposite report recurrence and switched recurrence with 10-overlay cross approval utilizing calculation classifier bolster vector machine. The authors presented [16] how the current online social networks spread fake news. Discussing how existing social network technologies such as maximization, information dissemination, and epidemiological models contribute to the creation and dissemination of false news. Solutions are also checked to reduce the production and dissemination of fake news.

3. Proposed System

3.1.Problem Statement

The proposed work suggests a novel and amalgamated method combining some known and well researched methods merging the advantages of AI through the simplistic algorithm, Logistic Regression, which is a simple, easy to understand, quick yet efficient algorithm. Since a major drawback, for ML and AI algorithms, when it comes to processing information such as news articles would be the checking the accuracy of facts or news that have surfaced recently. Since the suggested algorithm for fake news detection only works on a predefined dataset, the training module might not work as efficiently for the same. Thus the proposed system boost the accuracy and bolsters the results through the integration of a web scraping module which is capable of scraping through various news websites, internationally recognized as accurate, for latest news articles and saving them into a text file for matching against the given input by the user.

Also, another important aspect that underlies the fake news in the modern times is the use of hate speech viz. text that appeals to the emotions of people, to move their opinion against some issue, usually satisfying a political agenda. Hence to propose a wholesome solution the proposed system counters the propagation of hate speech by curbing it from the source, by felicitating the social media user who perceives the fake news or hate speech article, usually aligning with his political or religious bias, as authentic and forwards it to other citizens. Thus, the proposed system also provides a sentiment analysis module which can recognize the underlying sentiment behind the input text and inform the user for any bias in the sentiment behind the text.

The proposed system helps the user with checking mechanism, a system that could inform him about the various characteristics of the input article such as accuracy, hate speech etc. This will felicitate the users to make an informed decision that would further help to make them understand the authenticity of various texts and articles of social media to decide to curb the propagation of the fake news at the source.

3.2. Fake News Detection

The proposed work is being implemented through the integration of 3 modules, namely the logistic regression module and the web scraping module to detect fake news and the sentimental analysis as shown in Fig. 1.

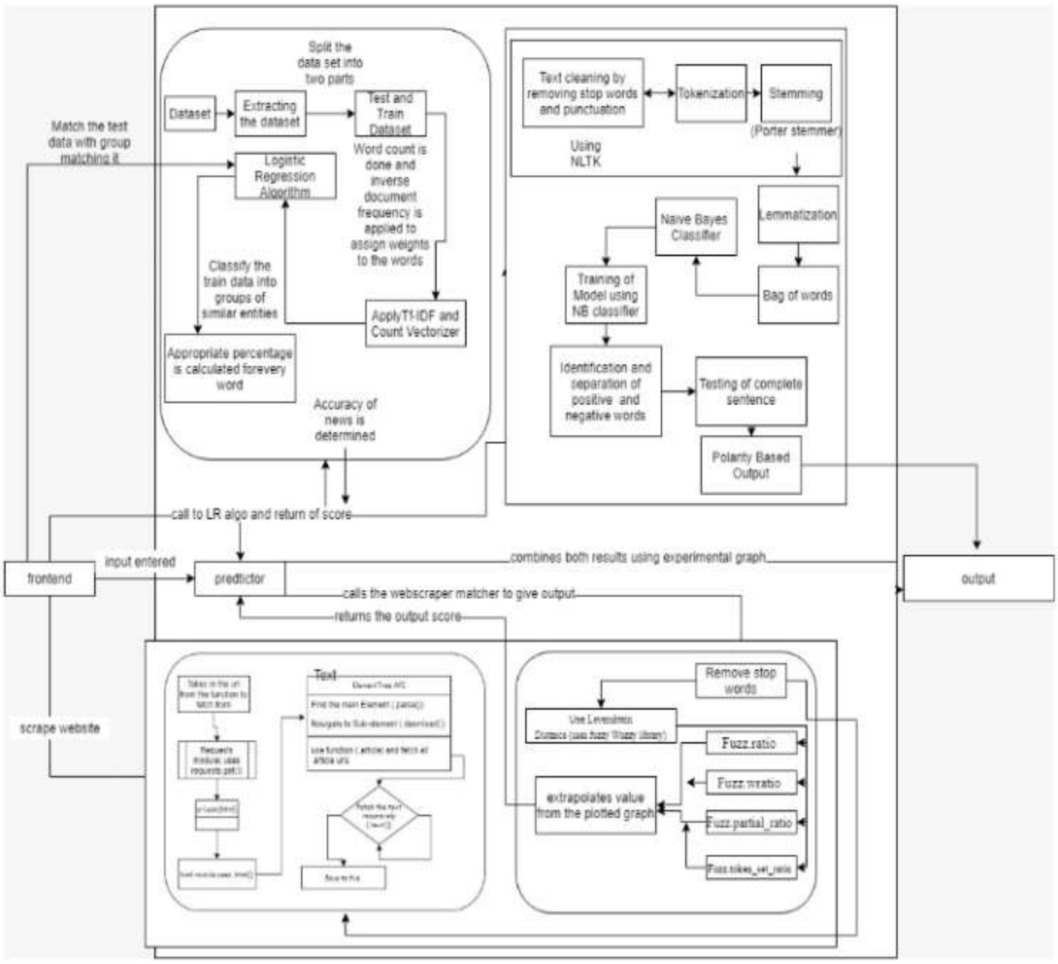

Split the data set into two parts
Dataset
Extracting the dataset
Test and Train Dataset
Match the test data with group matching it
Logistic Regression Algorithm
Word count is done and inverse document frequency is applied to assign weights to the words
Classify the train data into groups of similar entities
ApplyT1-IDF and Count Vectorizer
Appropriate percentage is calculated forevery word
Accuracy of news is determined
Text cleaning by removing stop words and punctuation
Tokenization
Stemming
(Porter stemmer)
Using NLTK
Naive Bayes Classifier
Lemmatization
Training of Model using NB classifier
Bag of words
Identification and separation of positive and negative words
Testing of complete sentence
Polarity Based Output
call to LR algo and return of score
frontend
input entered
predictor
combines both results using experimental graph
output
calls the webscraper matcher to give output
returns the output score
scrape website
Text
Remove stop words
Use Levenshtein Distance (uses fuzzy Wuzzy library)
Fuzz ratio
Fuzz wratio
extrapolates value from the plotted graph
Fuzz partial_ratio
Fuzz token_set_ratio

Fig. 1: Proposed Fake News Detection Architecture

3.2.1 Logistic regression

The logistic regression module performs the simple task of taking the dataset, splitting it into two parts, viz. test and train set. The train dataset which is the BuzzFeed dataset is used to train the regression model for the user input which is the news to be tested in this case. It is a classification algorithm used for machine learning that predicts the likelihood of a categorical dependent variable, where it will be either fake or authentic henceforth logistic regression will help to describe a relationship between a set of independent variables and categorical dependent variables. The dependent variable in logistic regression is a binary variable that includes data encoded as 1 (the user given news is fake) or 0 (the news is authentic), hence these are the only two classes. The model gives a authenticity value between

0 and 1 later on converted into percentage and hence can be easily categorized as how much the news is authentic or fake.

In plain terms it forecasts the possibility of incidence of fakeness in the news set by fitting the data to the logit function which has already been trained by the BuzzFeed dataset which the proposed system used to train the model. Modelling the probability of unreliability in relation to our predictor variables is based on logit transformation and maximum likelihood estimation. We will continue with the basic linear regression equation with dependent variable included in a relation function to proceed with logistic regression:

$$h(y) = \beta o + \beta(\text{Fake}) \quad (1)$$

In logistic regression our interest lies in the likelihood of outcome dependent variable (success or failure in relation to news being fake or authentic). As described above, h () is the link function. This function is defined using two things: Success(p) likelihood and Loss Probability(1-p). p would satisfy the conditions

Since probability must be positive so the exponential form should be used. Exponent of this equation can never be negative for any value of slope and dependent variable.

$$f = \exp(\beta o + \beta(\text{Fake})) = e\hat{}\,(\beta o + \beta(\text{Fake})) \quad (2)$$

f must be divided by a number greater than f to allow the likelihood less than 1, It can be achieved easily by:

$$f = \exp(\beta o + \beta(\text{Fake})) / \exp(\beta o + \beta(\text{Fake})) + 1 = e\hat{}(\beta o + \beta(\text{Fake})) / e\hat{}(\beta o + \beta(\text{Fake})) + 1 \quad (3)$$

Using (1), (2) and (3),it can be rewritten as:

$$f = e\hat{}y/ 1 + e\hat{}y \quad (4)$$

where f is the probability of success which means the news is fake. If p is the probability of success, 1-p will be the probability of failure which means that news is authentic which can be written as:

$$q = 1 - f = 1 - (e\hat{}y/ 1 + e\hat{}y) \quad (5)$$

where q is the probability of failure.

On dividing, Equation (4) and (5), we get,

$$f/(1-f) = e\hat{}y \quad (6)$$

After taking log on both sides, we get, y=log(f/1-f) is the link function. This function ultimately has the purpose to take a linear combination and convert those values to the scale of probability between fake and authentic (1 and 0 respectively).

To train and construct the classifier model, a training dataset that has the defined discriminants is used. The classification model is then used for the evaluation of unauthenticated news data and responds with a percentage to decide how much the news data is false.

The labelled news dataset in the study serves as preliminary data used as basis to extract the values for needed in the model. The dataset is secondary in nature which was retrieved from an online dataset repository consisting of 10,000 news articles gathered from different English sources from of which 5,000 are mark as fake news, and 5,000 are mark as legitimate news. Each row in the dataset contains, the actual content of news and a classification on whether the content is false or true.

3.2.2 Web Scraping

However, the regression module alone is not enough to test against facts which form a major percentage of a news. Hence, to produce more accurate results and to check against newly surfaced news articles, the web scraping module is coupled with this model. The web scraping uses an inbuilt module called 'newspaper' which in turn combines two basic inbuilt modules of python viz. 'Requests' and 'lxml'. The requests module is used to send all kinds of HTTP requests. It is a pretty straightforward module which is imported in python and the necessary news website is requested using the Requests. Get(URL)

The usage of requests module has been majorly implemented for its simplicity. The other module as mentioned above as used by the newspaper module is the 'lxml' module. After the page has been requested by the Requests module, the lxml file is used to handle the XML and HTML files. The whole HTML page can be seen in the form of an XML tree, having Elements and Sub-Elements. The text of the article is usually wrapped into some Sub-Element, hence following a tree like hierarchy. The newspaper module uses the Element Tree API viz. etree which parses through the XML Tree as shown in the Fig.2 and helps to fetch the text from the appropriate tag.

and helps to fetch the text from the appropriate tag.

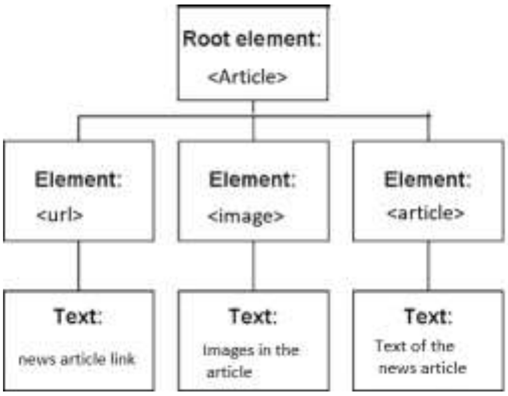


Fig.2 Understanding XML Tree

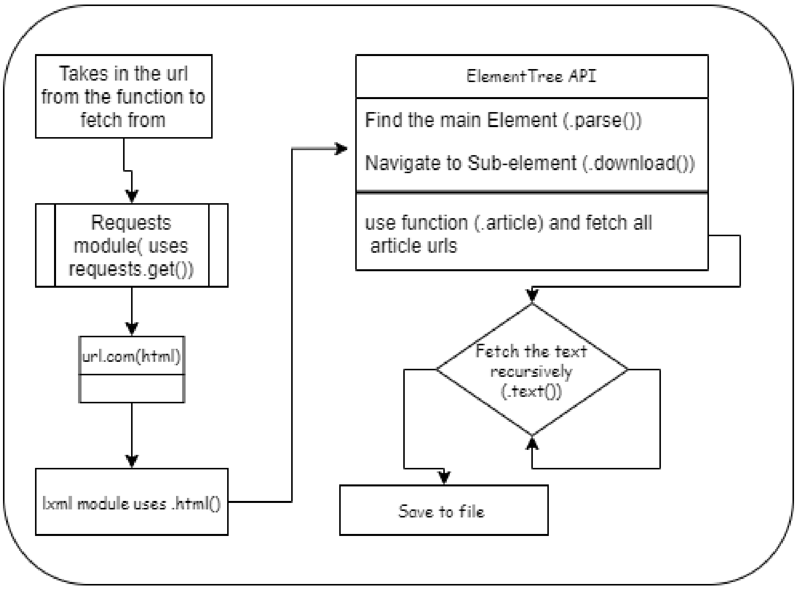

Takes in the url from the function to fetch from
Requests module(uses requests.get())
url.com(html)
lxml module uses .html()
ElementTree API
Find the main Element (.parse())
Navigate to Sub-element (.download())
use function (.article) and fetch all article urls
Fetch the text recursively (.text())
Save to file

Fig.3 Scraping the Website

The process can be better understood diagrammatically through the Fig.3 as shown.
After the website has been scraped and the news articles have been saved to a file, a matcher script is run which matches the input string by the user with the scraped content to yield the results. For matching the strings, the file containing the scraped articles is removed of any stop words, matches the strings based on Levenshtein Distance (used by FuzzyWuzzy module in python), which can be described as the difference between the two given strings based upon the minimum no of deletions, insertions and modification required to convert one string to another. The given library provides 4 different ratios that have been used in the proposed model:

a) Fuzz.ratio – which gives the ratio by comparing the strings exactly. Viz. cases are matched, and punctuations are matched and a number depicting the match score out of 100 is given.
b) Fuzz.WRatio –it is similar to exact ratio except the fact that it handles lower and upper case and unnecessary punctuations.
c) Fuzz.partialratio – It works in such a way that it calculates the length of the two strings and checks whether the shorter one is a part of longer one or not.
d) Fuzz.token_set_ratio -It works in this manner:

    [sorted_intersection] + [sorted_rest_of_strings_in_str1]
    [sorted_intersection] + [sorted_rest_of_strings_in_str2]

    and then each one is compared using simple ratio. Here, the sorted intersection means common tokens between the two strings sorted in the alphabetical order. Sorted rest of the strings refer to remaining of the tokens in the string.

The 4 ratios are calculated, a score out of 5 is assigned to different scores given by the ratios. After an experimental analysis a graph is plotted as shown in Fig.4. The graph is then used to further extrapolate the score based upon the score given by the ratio.

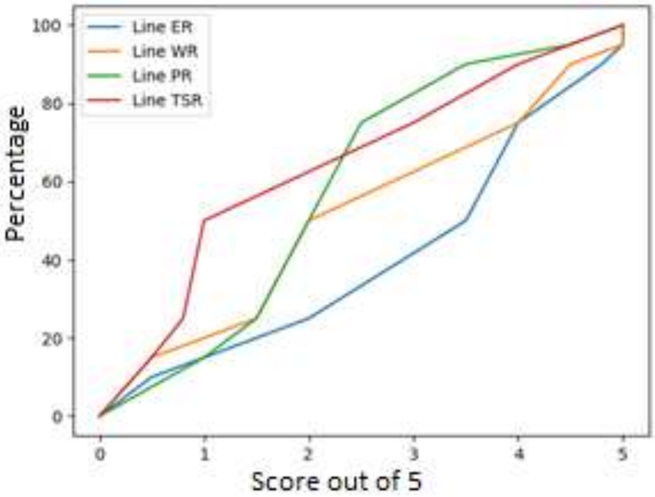


Fig.4: Graph of 4 Ratios on Experimental Basis

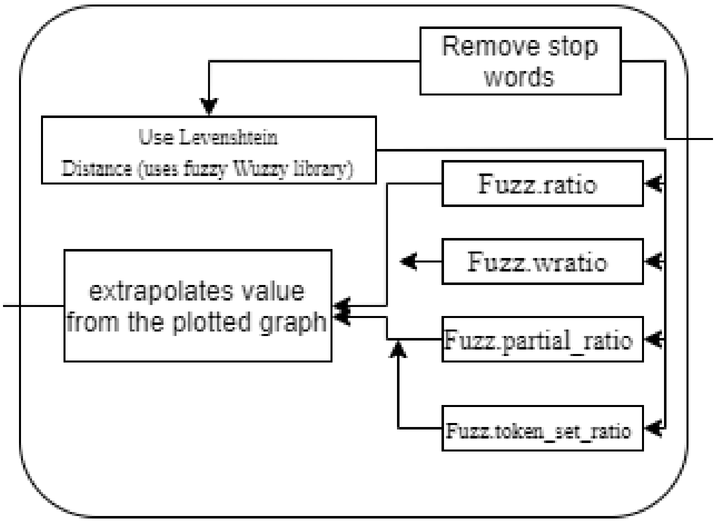

Remove stop words
Use Levenshtein Distance (uses fuzzy Wuzzy library)
Fuzz.ratio
Fuzz.wratio
Fuzz.partial_ratio
Fuzz.token_set_ratio
extrapolates value from the plotted graph

Fig.5: Matching with the Dataset Prepared from Scraping

Here ER represents exact ratio, WR represents WRatio, PR represents partial ratio and TSR represents token set ratio. The score of 5 represents highest score and the score 0 represents lowest score. The score from all the above ratios is combined and the average is given as the percentage accuracy match with the input string and the scraped news the whole process is explained as shown in Fig.5.

3.2.3 Combining the results from the logistic regression module and the scraping module

For more accuracy and better results, the results from both the logistic regression module and the scraping module are combined using a graph plotted appropriately using the values on experimental basis for the scraping module and the logistic regression module. The graph is further extrapolated for further results generated by the modules and combined by taking the mean of the results obtained from the graph and the output is generated for the web scraping module in terms of accuracy percent.

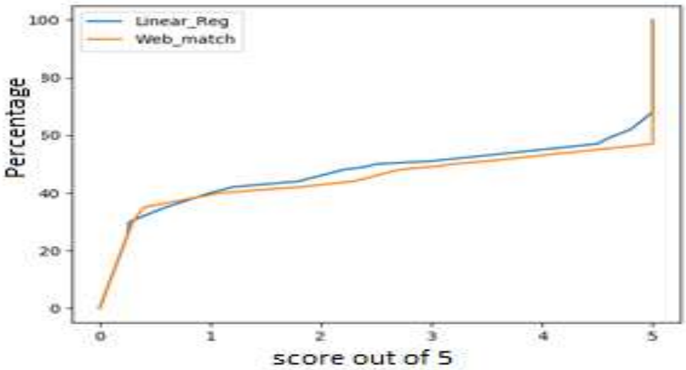


Fig.6: Combining Results from Web Scraping and Logistic Regression through Experimental Analysis

Score out of 5 in percentage is given in Fig.6 for linear regression and web scraping where the score of 5 is the highest and the score of 0 being the lowest.

3.3 Sentimental analysis

3.3.1 Text-Cleaning

Tools like NLTK (Natural Language Toolkit) are used. It helps to convert raw text into a list of words. We split the text into words, choosing alphanumeric character strings (A-Z, 0-9, a-z and '_'). We remove all punctuations like commas, quotes etc. along with the whitespaces.

3.3.2 TF-IDF Vectorizer

Tokenization of the information was completed, and a corpus was made. TF-IDF, term recurrence reverse report recurrence vectorizer from the scikit-learn library is utilized for creating highlights right now. The TF-IDF vectorizer utilizes the corpus created utilizing tokenization. TF-IDF is an estimation plot allotting appraisal or loads dependent on its term recurrence (tf) and the reverse variable recurrence (IDF) for each word in a report. The weight relegated by TF-IDF vectorizer is utilized as a parameter to pass judgment on the pertinence of a word in the document. The words containing higher weight esteems are regarded progressively significant. The worth expands relatively to how often a term shows up in the content yet is remunerated by the event of the term in the corpus. TF-IDF is represented by:

$$\text{TF-IDF} = \text{TF}(g, h) \times \text{IDF}(g, h) \quad (7)$$

Here, Term Frequency signified as TF and is determined from the tally (c), term (t) in report (d) and spoke to as TF (t,d). The recurrence of event of words to a double element is changed over by utilizing 1 (present in record) and 0 (not present in report). The frequencies can be standardized utilizing normal and logarithms. The reverse archive recurrence (IDF) for a word 'w' in report content (t) as processed by:

$$\text{IDF}(g, h) = 1 + \log T\,(1 + df(g)) \quad (8)$$

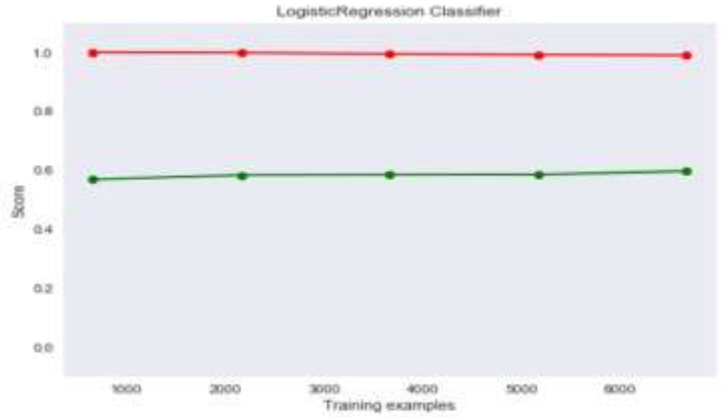
Here, T speaks to the complete archive include in our corpus and df (t) speaks to the check of the quantity of records where the term t is available. The result of two estimates will assist with registering TF-IDF. Euclidian's normalized form is used to calculate the final TF-IDF:

$$\underline{Tfidf} = \frac{tfidf}{||tf\ idf||} \quad (9)$$

Here ||tf idf|| is the Euclidean norm

4. Results and Discussions

The proposed system utilizes three diverse assessment metrics: Precision score, Recall score and F1 score. The F1 score can be deciphered as Precision and Recall weighted normal.

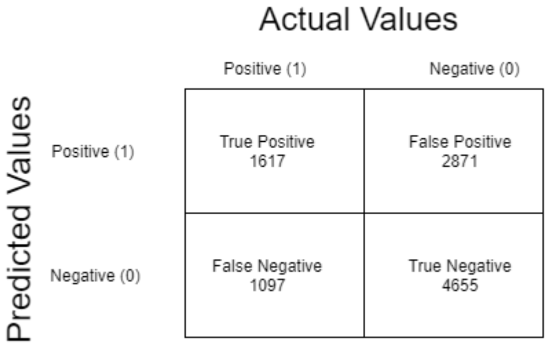


5.

6.

Fig. 7: The Results of Logistic Regression

The logistic regression had an f1 score in the range of 70's as shown in Fig. 7. It is due to fewer data points which we used for training purposes and our model's simplicity. We might add a few more feature selection approaches for potential implementations, such as POS tagging, word2vec and topic modeling.



Actual Values

Predicted Values	Positive (1)	Negative (0)
Positive (1)	True Positive 1617	False Positive 2871
Negative (0)	False Negative 1097	True Negative 4655

Fig. 8: Predicted Values v/s Actual Values

The confusion matrix results are given above which looks highly promising as the results of True Positive (TP) and True Negative (TN) are high.

The results for the web scraping highly depend upon the user if he scrapes the latest news, however the accuracy of the web scraping module alone comes around to be 45%. But when combined with the logistic regression module it comes around to 80%

It was seen that the F1 Score of Sentiment Analysis is 74%, the precision score cam out to be 65% and recall score was near to 84%.

5. Conclusion

With social media increasingly prevalent, more and more people are receiving news from social media rather than traditional news media. Online networking has since been used to disseminate misleading news, which has had significant adverse effects on individual consumers and broader community.

In this paper, we discussed the problem of fake news by combining three separate approaches for greater accuracy in identifying false news. Based on the findings discussed above, the results of this analysis indicate that a Fake News Classifier can detect false news with 80% accuracy. This is also suggested, however, that new discriminants need to be incorporated in future research to boost the precision of fake news classifier to at least 90% and above.

References

[1] Torabi Asr, Fatemeh, and Maite Taboada. "Big Data and quality data for fake news and misinformation detection." Big Data & Society 6, no. 1 (2019): 2053951719843310.

[2] Granik, Mykhailo, and Volodymyr Mesyura. "Fake news detection using naive Bayes classifier." In 2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON), pp. 900-903. IEEE, 2017.

[3] Zhou, Xinyi, Reza Zafarani, Kai Shu, and Huan Liu. "Fakenews: Fundamental theories, detection strategies and challenges." In Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, pp. 836-837. ACM, 2019.

[4] Shu, Kai, Deepak Mahudeswaran, and Huan Liu. "FakeNewsTracker: a tool for fake news collection, detection, and visualization." Computational and Mathematical Organization Theory 25, no. 1 (2019): 60-71.

[5] Figueira, Á., & Oliveira, L. (2017). The current state of fake news: challenges and opportunities. Procedia Computer Science, 121, 817–825. doi: 10.1016/j.procs.2017.11.106

[6] Zhang, Xichen, and Ali A. Ghorbani. "An overview of online fake news: Characterization, detection, and discussion." Information Processing & Management (2019).

[7] Aldwairi, Monther, and Ali Alwahedi. "Detecting fake news in social media networks." Procedia Computer Science 141 (2018): 215-222.

[8] Shu, Kai, Suhang Wang, and Huan Liu. "Beyond news contents: The role of social context for fake news detection." In Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, pp. 312-320. ACM, 2019.

[9] Kwak, Jin-ah, and Sung Kyum Cho. "Analyzing Public Opinion with Social Media Data during Election Periods: A Selective Literature Review." Asian Journal for Public Opinion Research 5, no. 4 (2018): 285-301.

[10] Negi, Sapna, and Paul Buitelaar. "Inducing distant supervision in suggestion mining through part-of-speech embeddings." arXiv preprint arXiv:1709.07403 (2017).

[11] Poria, Soujanya, Erik Cambria, and Alexander Gelbukh. "Aspect extraction for opinion mining with a deep convolutional neural network." Knowledge-Based Systems 108 (2016): 42-49.

[12] Yadollahi, Ali, Ameneh Gholipour Shahraki, and Osmar R. Zaiane. "Current state of text sentiment analysis from opinion to emotion mining." ACM Computing Surveys (CSUR) 50, no. 2 (2017): 25.

[13] Della Vedova, Marco L., Eugenio Tacchini, Stefano Moret, Gabriele Ballarin, Massimo DiPierro, and Luca de Alfaro. "Automatic online fake news detection combining content and social signals." In 2018 22nd Conference of Open Innovations Association (FRUCT), pp. 272-279. IEEE, 2018.

[14] Jin, Zhiwei, Juan Cao, Yu-Gang Jiang, and Yongdong Zhang. "News credibility evaluation on microblog with a hierarchical propagation model." In 2014 IEEE International Conference on Data Mining, pp. 230-239. IEEE, 2014.

[15] Al-Ash, Herley Shaori, and Wahyu Catur Wibowo. "Fake News Identification Characteristics Using Named Entity Recognition and Phrase Detection." In 2018 10th International Conference on Information Technology and Electrical Engineering (ICITEE), pp. 12-17. IEEE, 2018.

[16] Campan, Alina, Alfredo Cuzzocrea, and Traian Marius Truta. "Fighting fake news spread in online social networks: Actual trends and future research directions." In 2017 IEEE International Conference on Big Data (Big Data), pp. 4453-4457. IEEE, 2017.