International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 07 | July 2021 www.irjet.net p-ISSN: 2395-0072

Fake News Detection Using Machine Learning

Ajay Kale[1], Ashutosh Jagdale[2], Swapnil Gadam[3], Vedant Satpute[4]

[1-4]*Department of Information Technology, Pimpri Chinchwad College of Engineering, Pune – 411044, Maharashtra, India*

Abstract - *Fake news is one of the most common complaint heard in current generation. As, it spreads mis-information in society and minds, making them question even real news and facts. In many cases an innocent person is convicted of a crime on basis of fake news. And it's very costly to fact check individual news with limited human Resources. So, using a machine learning approach to tackle the problem is the most sensible solution. Any fake news is engineered in a way that it can fool someone into thinking it as real news. So, we have devised our method targeting these features that make fake news a fake news. We are performing Feature Engineering, in which new features are created on the basis of ML models (Spam detection, Clickbait Analysis, Classification, Sentiment analysis. Stance Detection) and these interns are used in a final ML model. Multiple ML models have been used to compare the result and find out the most accurate among them.*

Key Words: *Spam detection, Clickbait Analysis, Classification, Sentiment analysis, Stance Detection.*

I. INTRODUCTION

As the Technological Advancement is continuing, it is also getting cheaper and more accessible to masses. Technology like, Smartphone is bringing lot more power into the user's palm than ever before. And, with power comes responsibilities that most people cannot handle and hence misuse it. Result of this is fake news, that travels like wildfire, and if its tempting enough, people read and believe these atrocities and pass on them to others. The misleading information can cause someone to have a negative opinion of person, place, etc. Social media and messaging, sites and apps such as, Facebook, Instagram, Twitter, WhatsApp, etc. are the key contributors in spreading fake content. And, there is no commercially successful platform currently available that can verify if the news being forwarded is fake or real. This scenario inspired us to choose our topic, to prepare a Machine Learning Model that can successfully categorize if a news is fake or not.

Any fake news is engineered in a way that it can fool someone into thinking it as real news. So, we have devised our method targeting these features that make fake news a fake news. For a fake news to be tempting needs a fact to be repeated in multiple ways with reasons, in a way to "spam" users with information. To grab user's attention and to make them read the article, the title of article should be bold, surprising and one with claiming something, making it a "clickbait". Generally fake news doesn't talk about a single subject and jump from one topic to other hence one cannot "categorize" them to a single genre.

These are the few observations on which the system architecture is proposed.

II. LITERATURE REVIEW

A. Spam Detection

Houshmand Shirani-Mehr Studies on different Classification Models used for Spam detection System. And, how highest accuracy be achieved [1]. They used a database called UCI Machine Learning repository which consisted of 5574 text messages. Slitting the tokenized dataset into 70% training and 30% as testing. Will have 7,789 features in database after extraction of tokens for all the messages. Tokens with frequency less than 5 and more than 500 are removed. [2]

Applying different algorithms to the dataset using extracted features with different training set sizes.

Model	BH%	Accuracy%	SC%
Adaboost	0.51	98.59	92.17
SVM	0.31	98.86	92.99
k-nearest neighbours	0.40	97.47	82.60
Multinomial NB	0.51	98.88	94.47
Random Forest	0.29	98.57	90.62

From results it was suggested that, multinomial naive Bayes with Laplace smoothing and SVM with linear kernel are one of the best classification algorithms for SMS spam detection. [2]

Then did data Processing by removing stop words and using Lemmatization (the context of the sentence is kept as it is in lemmatization as compared to stemming). A word dictionary is created having words

and their respective frequency. Dictionary is created using 3000 most frequently used words from training dataset

Generated word count vector for each email of training set with vector size of 3000 features. Each vector has the frequency of 3000 words, where maximum number of them are zero.

Trained three models Naive Bayes, Support Vector Machines (SVM) and k-NN. Results indicate that Naive Bayes and SVM show better results than k-NN.

B. Clickbait analysis

The authors of referred paper collected both clickbait and non-clickbait categories. [3] They extracted 18,513 headlines from Wikinews articles which represent non-clickbait data, collected by Newsreader and for clickbait they collected articles from multiple sources as: 'BuzzFeed', 'Upworthy', 'Viral Nova', 'Scoop whoop', 'Viral Stories'. They carried out linguistic analysis on these articles using the Stanford Core NLP tool. [3]

They considered following features to determine clickbait:

Sentence Structure – Headline length, Length of words, Length of syntactic dependencies.

Word patterns - Stop words, Hyperbolic words, Internet slangs, Punctuation patterns and Common bait phrases.

Linguistic categories - Sentence subjects, Determiners, Possessive case.

N-gram feature - POS, Tags Word N-grams and Syntactic N-grams. [3]

These 14 features are used with 3 models, SVM with RBF kernel, Random Forests and Decision Trees. [3]

	Decision Tree	**Random Forest**	**SVM**
Accuracy	0.90	0.92	0.93
Roc-auc	0.90	0.97	0.97
Recall	0.89	0.91	0.90
Precision	0.91	0.94	0.95
F1-score	0.90	0.92	0.93

From the table it is clear that SVM achieved higher accuracy of 93%, precision of 0.95 and recall of 0.9. [3] Therefore, it is used to them to predict clickbait using two approaches, Topical similarity and Linguistic patterns.

In topical similarity approach they first extracted topics from a clickbait article and check similarity of between topics of new and previously extracted articles.

In linguistic patterns approach they recognized the linguistic styles within the articles. As linguistic pattern-based approach gives better results, hence it is primarily used.

C. Sentiment Analysis

Sheresh Zahoor and Rajesh Rohilla have worked on sentiment analysis of twitter database using the built-in libraries of python for analyzing the text. [4]

The lexical technique is primarily based totally on some of steps which are very essential to perform the sentiment from the sentence or word. The steps are:

1) Data collection - Data was collected from the twitter dataset and Rest API was used to collect one time collection of tweets based on the queries and it was stored in csv file. The csv file was having the following column such as date, text, retweet, hashtag, followers. [4]

2) Data pre-processing – Before the real sentiment evaluation a few pre-processing steps want to be carried out to make the tweet to be analyzed. The steps are Tokenization, N-grams extraction, Stemming and Lemmatization, Stop words removal

3) POS - This is the interaction of naturally labeling each word in the content as far as the grammatical feature it has a place with like thing, pronoun, qualifier and so on. [4]

4) Sentiment evaluation the use of in-constructed dictionary- In these studies in-constructed libraries were used to research the sentiment of unique occasions and the consequences acquired are as compared with every other. [4]

	Case-1		**Case-2**	
Sentiment	**Text Blob**	**Vader**	**Text Blob**	**Vader**
Positive	29.7%	44%	58.5%	62.3%
Negative	12.0%	17.6%	9.6%	17.2%
Neutral	58.3%	38.5%	32%	20.5%

On the basis of this results, by combining the score of both the libraries and substituting the values in the expression, the final sentiment score can be calculated that gives a good accuracy. [4]

D. News Classification

The goal is to create a model that takes properties such as message title, brief description as an input and these properties are combined into a single property as "text" which produces "category" of news to which it belongs. [5]

Methodology:

A preprocessing step is essential for converting the data

Feature / ML Techniques	TF IDF Feature		Binary Feature		Count Feature	
	Train	Dev	Train	Dev	Train	Dev
Kernel SVM	NA	NA	0.995	0.608	0.974	0.610
Random Forest	NA	NA	0.998	0.586	0.998	0.583
Logistic Regression	0.776	0.670	0.741	0.640	0.746	0.637
Naive Bayes	0.600	0.561	0.665	0.610	0.677	0.620

which is in the form of text from unstructured form to structured form. Following are the main steps that are involved in the process of text classification: [6]

1. News Collection

Accumulating news from various sources available as it has emerged as a major source for obtaining news.

2. News Pre-processing

Data now needs to be discriminated from unrelated words like punctuation marks, special characters etc. Data is made free from those words.

a. News Tokenization

It involves distributing the huge text into small tokens.

b. Stop Word Removal

It generally includes conjunctions, pronouns and prepositions and are removed eventually.

c. Word Stemming

This step reduces a word to its base word. The motive behind using stemming is to remove the suffixes so that the number of words would be brought down.

3. Feature Selection

Generation of word dictionary was been done with the aid of news description that we had already processed.

4. News Classification

Its aim is to classify the unseen news to their respective categories using methods such as Naive Bayes, Artificial Neural Networks, Decision Trees, SVM, etc. [6]

Conclusion:

We have constructed various models that use traditional Machine Learning techniques to predict news category through brief description and headlines.

Best achievement in the development set has an accuracy rate of 88.72%, as far as top 3 labels are concerned and accuracy rate of 68.85%, considering top 1 label as the model speculated.

III. SYSTEM ARCHITECTURE

Datasets:

We started with Fake News data given by Kaggle.com. During our initial walkthrough we realized that the data set had a very limited set of features. We also felt that the dataset can be further enriched by adding *domain ranking, spam detection, Clickbait analysis, sentiment analysis and news classification* features to identify whether the news is fake or not. Data was only having fake news so we balance it by adding real news from different data source.

Data Enrichment:

1. Acquired real news data and updated it to merge in fake news data.
2. Used Amazon Alexa API to acquire domain ranking data for individual URLs in the Data.
3. Also, weekly data is scrapped from other sites.

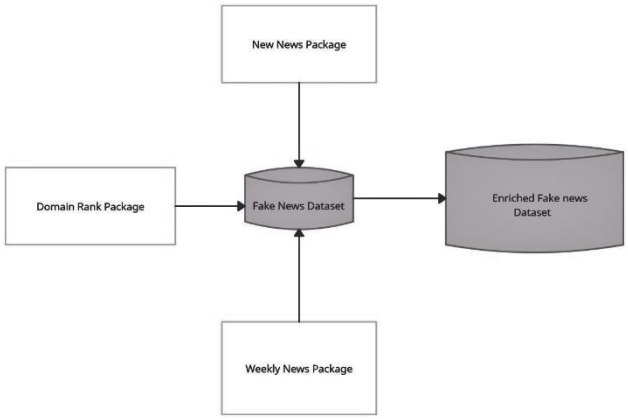

Fig 3.1.1 - Diagram depicts the data enrichment process for fake news.

Data Pre-processing:

As Data set was not even readable format so performed various data pre-processing such as dropping unwanted columns, handling the fields in dataset which may have blank entries, only keeping the essential characters and removing any special characters, used Stemming Library to stem words into their root format. Transformed Real News data from file level to original fake news format.

Distillation:

Performed several distillations, and after Merging, we came up with our final master dataset. This dataset went through some pre-processing such as, hot encoding of columns.

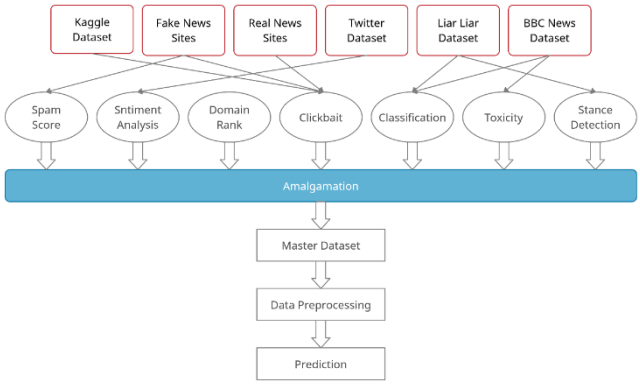

Fig 3.1.2 - Distillation and Merging

As the data is Fully ready, we Started to work on data using various plotting libraries available such as mathplotlib, seaborn etc. to analyze the data and extract out set of features that can help to distinguish real news and fake news.

Analysis:

The target distribution Study Concluded that about 53.8% data was insincere news which includes bs, conspiracy, satire, bias, fake.

Used word cloud and n-gram approach for further analysis. The Observations Suggests that the Fake news are Targeting Specific people, country, race or age. Now to Fish out the Differences between the two types of news we created some Meta Features such as - No. of words, No. of unique word, No. of characters, No. of stop words, No. of punctuations, no of upper-case words, no of title case words, Average length of the words. Features Analysis showed that the fake news has more words and characters compared to real news.

Training:

We only ran models which helped us to drive the importance of the factors and weigh for prediction equation. We visualized equation at the end as per model driven weight. We believed that model driven Weight will be more accurate

As features such as domain rank and number of words are very valuable but can also lead to overfitting of models. So, we opted to make two models to get Uniform representation of all features.

Model Verification models that we have used are *ROC* and *Precision-Recall*. Also, we have used *K-fold cross Validation* to further increase training quality

Have applied two machine learning models namely *Naïve Bayes* and *Logistic Regression.*

Final Accuracy After applying k-fold cross validation

- Logistic regression – 85%
- Naïve Bayes – 88%

IV. EXPERIMENTATION RESULTS AND DISCUSSION

Our hypothesis about polynomial equation can be shown as follows. Ideally, models internally construct equation and predict accordingly.

y = x1(domain score) + x2(sentiment analysis) + x3(spam detection) + x4(news classification) + x5(toxicity detection) + x6(clickbait detection) + x7(word count)

Our accuracies for the models are as follows.

- Logistic Regression - 85%
- Naive Bayes - 88%
- Further models are to be applied

Final model probability(y) can be mapped as:

>90% = True

75-90% = Mostly True

50-75% = Half True

40-50% = Mostly False

25-40% = False

<25% = Pants on Fire

V. CONCLUSION

In this paper we have discussed all the Components of our proposed method, how they work together to provide the final output, how data is enriched to provide all round news coverage, what type of analysis was done to arrive at the conclusion, how the final architecture came about, and the partial outputs that we have acquired where we can see that Naive Bayes has higher accuracy than Logistic Regression. In future Studies we will do the final model analysis with more types of models, to suggest the model with highest accuracy. Also, we will be exploring the possibility of applying re-enforcement learning in our Method. And also, we will be working on adding more features in the master dataset.

VI. ACKNOWLEDGEMENT

We thank our institute Pimpri Chinchwad College of Engineering, Pune for the infrastructure provided to us. We are deeply indebted to our guide and Head of Department who have inspired us in undertaking this task. It is difficult to express our gratitude in words, as it is, they who lead us marching towards this goal.

REFRENCES

[1] "SMS Spam Collection Data Set from UCI Machine Learning Repository," [Online]. Available: http://archive.ics.uci.edu/ml/datasets/SMS+Spam+Collection.

[2] Shirani-Mehr and Houshmand, "SMS spam detection using machine learning approach," in Standford, 2013.

[3] A. Chakraborty, B. Paranjape and S. K. N. Ganguly, "Stop Clickbait: Detecting and preventing clickbaits in online news media," in IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM),, 2016.

[4] S. Zahoor and R. Rohilla, "Twitter Sentiment Analysis Using Lexical or Rule Based Approach: A Case Study," in 8th International Conference on Reliability, Infocom Technologies and Optimization, Noida,India, 2020.

[5] "Kaggle News Category Dataset.news-category-dataset," 2018. [Online]. Available: https://scikit-learn.org/stable/modules/svm.html.

[6] A. Patro, M. Patel, R. Shukla and D. J. Save, "Real Time News Classification Using Machine Learning," in International Journal of Advanced Science and Technology, 2020.

[7] Lai and Chih-Chin, "An empirical study of three machine learning methods for spam filtering," in Knowledge-Based Systems, 2007.

[8] Manzoor, S. Ishfaq and J. Singla, "Fake news detection using machine learning approaches: A systematic review.," in 3rd International Conference on Trends in Electronics and Informatics (ICOEI). IEEE, 2019.

[9] Merryton, A. Rajasenah and G. Augasta, "A Survey on Recent Advances in Machine Learning Techniques for Fake News Detection," in Test Eng. Manag83 , 2020.

[10] M. Gupta and e. al, "A comparative study of spam SMS detection using machine learning classifiers.," in Eleventh International Conference on Contemporary Computing (IC3). IEEE, 2018.

[11] R. Oshikawa, J. Qian and W. Y. Wang, "A survey on natural language processing for fake news detection," in arXiv preprint arXiv:1811.00770, 2018.