IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Fake News Detection Using Machine Learning

DARSHAN D B

Department of Computer Science and Engineerning

Presidency University, Bangalore-560064

BHUVAN GOWDA N

Department of Computer Science and Engineerning

Presidency University, Bangalore-560064

SACHITH R

Department of Computer Science and Engineerning

Presidency University, Bangalore-560064

DARSHITH K R

Department of Computer Science and Engineerning

Presidency University, Bangalore-560064

Mr. AFROJ ALAM

Assistant Professor

Department of Computer Science and Engineerning

Presidency University, Bangalore-560064

Abstract— Fake News and Scams started in the web time frame. The fake news design started basically to dupe per-clients, increase readership and is oftentimes used as a strategy for mental battling. Advances in the development and the spread of news through different sorts of media, without truly checking the real factors, have extended the spread of fake news today. The essential purpose behind this endeavor is to devised a classifier that can isolate fake news from certifiable news.

We propose in this research project, a fake news detection model that uses a Text Vectorizer and machine learning techniques to tackle the problem. Experimental evaluation yields the best performance using Term FrequencyInverted Document Frequency (TF-IDF) as feature extraction technique, and Passive-Aggressive Classifier as the classifier, with an accuracy of more than **97%**.

Keywords— Fake News, Real News, Machine Learning Techniques, Text classification, TF-IDF Vectorizer, Term Frequency (TF), Inverse Document Frequency (IDF), Passive Aggressive Classifier (PAC), Python.

I Introduction

The effects of fake news have increased exponentially in the recent past and something must be done to prevent this from continuing in the future. The dangerous effects of fake news, as previously defined, are made clear by events such as in which a man attacked a pizzeria due to a widespread fake news article. This story along with analysis provides evidence that humans are not very good at detecting fake news, possibly not better than chance. As such, the question remains whether machines can do a better job. A machine can solve the fake news problem using supervised learning that extracts features of the language and content only within the source in question, without utilizing any fact-checker or knowledge base.

Do you trust all the news you hear from social media? All news is not real, right? So how will you detect the fake news? – The answer is Python.

By practicing such an advanced python project of detecting fake news, we will easily make a difference between real and fake news. Before moving ahead in this advanced Python project, we have to be aware of related terms of fake news like the TF-IDF Vectorizer, and the Passive Aggressive Classifier.

This project will work you through the necessary steps and techniques used to implement such an analysis.

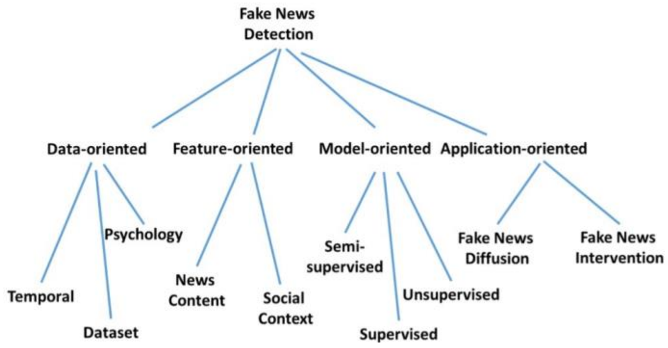


Figure 1: Future directions and open issues for fake news detection on social media

I.1 Problem Statement

This project proposes the question of whether it is possible to detect fake news through machine learning models. Specifically, the aim of this project is to determine the ideal model that is efficient in predicting fake news while also limiting the cost of memory and storage for computation. ”Fake news” has been a very recent and prevalent problem within recent years.

I.2 Detail of the Problem Application Area & Domain

Fake news spreads like a wildfire and this is a big issue in this era. We can learn how to distinguish fake news from real one. We will be using a supervised learning approach to implement the model.

As a consequence of the increase in cases of fake news in recent years, efforts have been made to crack down on the spread of misinformation throughout social media platforms. All popular social media platforms (Facebook, Twitter, Spotify, and YouTube) have permanently banned Alex Jones from using their networks following the events of ”Pizza Gate” in addition to multiple questionable accusations made by Jones, including an accusation made by Jones claiming that the Sandy Hook shooting was ”faked”.

Despite efforts of many social media websites and governments cracking down on fake news, many young people today generally are not able to tell the difference between fake news and real news. According to a Stanford study, it found that many students have a very strong inability in discerning between fake news. In the study, high school students were given two posts announcing the candidacy of Donald Trump’s presidential campaign. One post was given by an actual Fox News account another one posted by an account that ”looked” like it was from Fox News. 25% of the could not tell the difference between real and fake news sources. With over 30% of students favoring that the fake news account was more trustworthy.

Indeed, some politically charged or bogus articles that would be esteemed false frequently have more perspectives and offers via web-based media destinations than real news stories towards the most recent three months of the political race. As per an examination by Buzz-channel, posts, and stories composed from the best twenty most noteworthy performing trick locales and hyper-hardliners had over 8.7 million offers, ”responses” and remarks contrasted with the main twenty most noteworthy performing significant news associations had about 7.4 million offers, ”responses” and remarks via online media destinations.

The research problem was initially defined through the following use cases: In light of a single event/story, the framework would decide whether certain sources or articles are regarded to be fake news dependent on a given likelihood. Through the sources analyzed the machine learning agent would assign a level of bias and factuality of these articles by comparing them to each other and assign scores of the bias and factuality of the source.

A sort of sensationalist reporting, counterfeit news exemplifies bits of news that might be scams and is commonly spread through web-based media and other online media. This is regularly done to further or force certain thoughts and is frequently accomplished with political plans. Such news things may contain bogus and additionally overstated cases and may wind up being viral by calculations, and clients may wind up in a channel bubble.

I.3 Challenges and Motivation

Motivation— The motivation for research on this topic was that this is a relatively new area of research with many opinions but not many concrete solutions. Many implementations focus primarily on the host of the article, but even articles hosted on otherwise trustworthy websites can be classified as fake news. The primary motivation of this project was to bring awareness, propose a solution, and work towards minimizing the effects of fake news.

Challenges— Throughout the project and its analysis study, we faced some challenges such as:

- **Data Collection**: While collecting the data, we faced an issue with the relation of topics among news feeds. This is due to the variety of categorized topics covered by most informative platforms.
- **Data Analysis and Interpretation**: After collecting these data, we spend a lot of time analyzing from a feature-based perspective.
- **Data Preprocessing Methodology**: After proper analysis and interpretation, we need to prepare and sanitize the data for it to be suitable for more insights into the learning process.
- **Learning Model Selection** (Which model suits well the given problem.)

- **Update the Model for better Accuracy**: Here we try to tune the parameters/hyperparameter to obtain a better accuracy from the selected model.

I.4 Objectives

The purpose of this paper is to design and implement a machine learning implementation that correctly predicts if a given article would be considered fake news. The contributions of this paper are as follows

- Introduces the topic of fake news and the various machine learning algorithms to build a model to accurately classify a piece of news as REAL or FAKE.
- Provides an overview of the history and implications of fake news.
- This advanced python project of detecting fake news deals with fake and real news. Using SK-Learn tools, we will a **TfidfVectorizer** on our dataset.
- Then, we initialize a **Passive-Aggressive** Classifier (PAC) to fit the model, which will result in an accuracy score and a confusion matrix that tell us how well our model fares.
- Presents a possible solution and lays some groundwork in further study in this area.

II Background Study

Several groups and organizations have also worked on similar ideas in their own implementations. These works highlight some of the challenges of fake news detection. One implementation by *Katharine Jarmul*, founder of data analysis company *Kjamistan*, uses a Passive-Aggressive Classifier to detect fake news [9].The implementation is a tutorial on using different Bayesian models posted on DataCamp[9], which offers courses on a variety of data science topics including R and Python.

One paper titled *' Exploiting Network Structure to Detect Fake News'*, written by three Stanford University students, also implements a Neural Network for classifying fake news[12]. Their implementation also takes into account the social context in addition to article-specific features such as the title and content in an article in an attempt to improve prediction accuracy. This is one of the few possible ways to improve prediction accuracy without improving natural language processing.

Another paper titled *'Fake News Detection: Deep Learning Approach'* implemented three different neural network models to compare with the only difference between them being how they took in the article content and title[14]. This indicates that the way one goes about processing text in an article makes a huge difference in the performance of a model. This makes sense considering that an article's content is generally the only thing that can be analyzed to truly determine its authenticity.

Finally, another paper entitled *'Online Passive-Aggressive Active learning'* implemented a PassiveAggressive Active (PAA) learning algorithms by adapting the Passive-Aggressive algorithms in online active learning settings[7]. Unlike conventional Perceptron-based approaches that employ only the misclassified instances for updating the model, this proposed PAA learning algorithms not only use the misclassified instances to update the classifier but also exploit correctly classified examples with low prediction confidence.
Specifically, they propose several variants of PAA algorithms to tackle three types of online learning tasks: binary classification, multi-class classification, and costsensitive classification.

III Feasibility Study

A feasibility analysis evaluates the project's potential for success; therefore, perceived objectivity is an essential factor in the credibility of the study for potential investors and lending institutions. In the case of this research project, the feasibility analysis will try to outline the How's and Why's of this implementation and its requirements. Therefore the feasibility study will examine separately this study area which would result as follow.

IV Proposed Methodology

In this section, our aim is to propose a suitable solution that can result in a high-efficiency rate than the previously mentioned research works and implementations in section II.

IV.1 Overview of the Project and Related Terms

This project is about determining if a given news is *Fake* or *Real* through a set of mechanism illustrated in figure 2

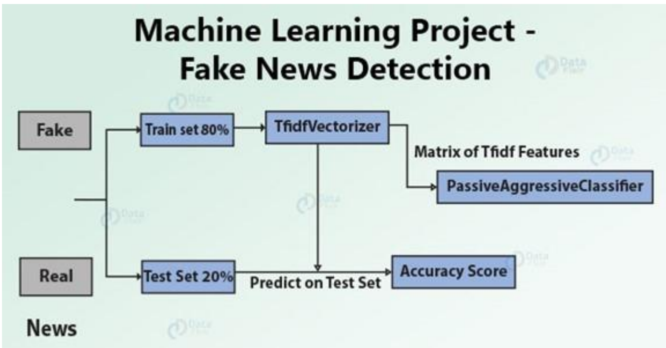


Figure 2: Project Overview Mechanism

IV.2 Structured Design Methodology:

Analysis Perspective

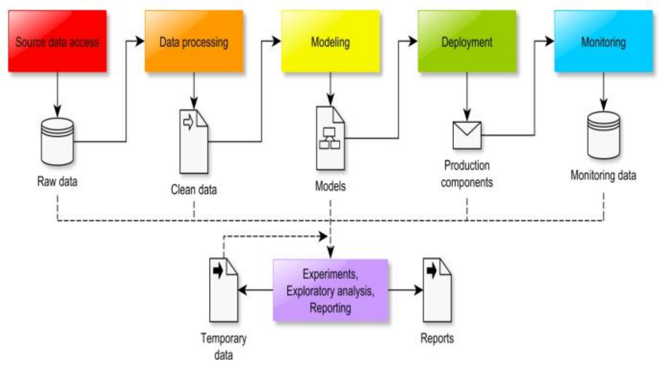


Figure 3: Design Workflow

IV.3 Data Collection & Feature Analysis

Data Collection— Guardian newspaper and Kaggle provide an API(Application Program Interface) which enables to populate the model with up-to date news. These samples data are shared among 03 files under the names:

- *'news.csv'* : which have a sample data of shape 6335 by 4 with 04 features ('Unnamed: 0', 'title', 'text', 'label') and contains a mixture of fake and real news.
- *'Fake.csv'* : which have a sample data of shape 23481 by 4 with 04 features ('title', 'text', 'subject', 'date') and contains only fake news.
- *'True.csv'* : which have a sample data of shape 21417 by 4 with 04 features ('title', 'text', 'subject', 'date') and contains only real news.

Feature Analysis— From the compiled samples data obtained above, we formed our experimental dataset based on 03 features which are *'title', 'text', 'label'* with a shape of 51233 by 3 and contains a mixture of fake and real news.

- **'title'**: The first column contains the title of the news.
- **'text'** : The second column contains the plain-text news.
- **'label'** : The third column has labels denoting whether the news is REAL or FAKE.

Index(['title', 'text', 'label'], dtype='object') (51233, 3)

	title	text	label
51228	LOL! TRUMP Responds To RACE-OBSESSED Congressm...	Donald Trump has been hit with celebrity backl...	FAKE
51229	Supreme Court faces 4-4 split in Obamacare con...	WASHINGTON (Reuters) - The Supreme Court on We...	TRUE
51230	UK aid minister to resign rather than be sacke...	LONDON (Reuters) - British aid minister Priti ...	TRUE
51231	After Kim Davis is jailed, marriage license is...	(CNN) With the clerk who had refused them in j...	REAL
51232	YOU WON'T BELIEVE THIS! CALIFORNIA GOVERNOR'S ...	I wrote AB 2466 because I want to send a mess...	FAKE

Figure 4: Compiled Dataset

NB: These samples data were taken from Guardian newspaper and Kaggle website where it can be found under the title *Fake and Real news dataset*[4]. Kaggle provide an API(Application Program Interface) which enables to populate the dataset with up-to date news. The training and testing data were collected using the API.

NB: This dataset is based on the topic of political news feeds in the USA and other countries with political instabilities.
In the next step, preprocessing of the dataset like removing stop words, punctuation marks, missing fields was done to sanitize the samples data.

IV.4 Machine Learning Model

Passive Aggressive algorithms— are online learning algorithms. Such an algorithm remains passive for a correct classification outcome, and turns aggressive in the event of a miscalculation, updating and adjusting. Unlike most other algorithms, it does not converge. Its purpose is to make updates that correct the loss, causing very little change in the norm of the weight vector.

IV.4.1 Classification Model used

For this research project, we decided to use a Passive Aggressive Classification(PAC) methodology as a model.
Passive Aggressive Algorithms are a family of online learning algorithms (for both classification and regression) proposed by Crammer at al. [3]. The idea is very simple and their performance has been proofed to be superior to many other alternative methods.
The Passive-Aggressive algorithms are a family of Machine learning algorithms that are not very well known by beginners and even intermediate Machine Learning enthusiasts. However, they can be very useful and efficient for certain applications.[1]

Note: This is a high-level overview of the algorithm explaining how it works and when to use it. It does not go deep into the mathematics of how it works.

V Implementation

Fake News Detection—— is based on a Passive-Aggressive Classification Algorithm. It was developed in Python using Google Colab Online IDE. The full implemented source code can be found under the GitHub Repository named *ML-Project*[8]. In this section, we will be outlining the algorithms and pseudo-codes used through well-elaborated flowcharts and explanations of the above mentioned Proposed Methodology given in section IV.

V.1 Overview of the experiment

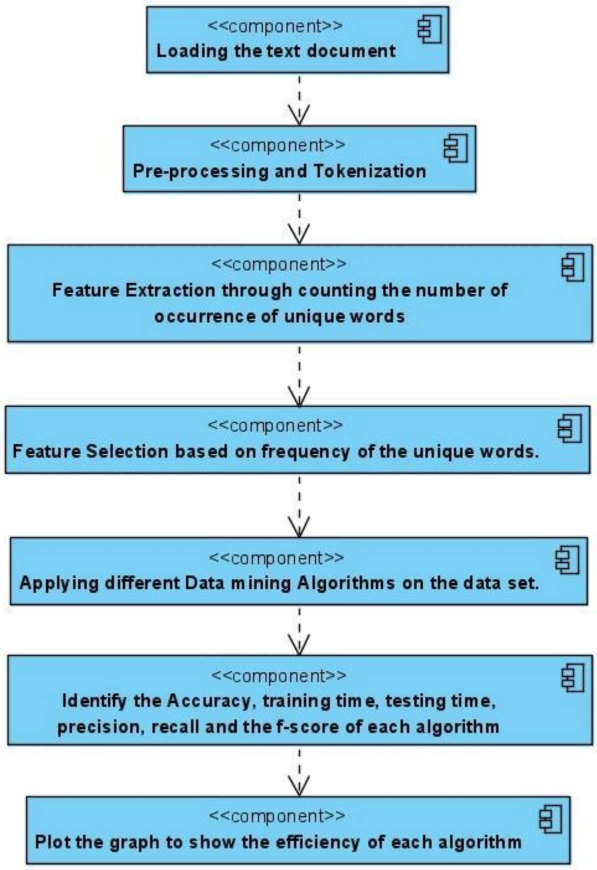


Figure 8: Workflow diagram of the proposed framework.

Project Detailed Workflow: Steps——

- Collecting Sample data related to Fake News Analysis.
- Building of a dataset from a set of known and welldone datasets.
- Loading and Analyzing dataset.
- Splitting the dataset into training and testing sets.
- Preprocessing of the dataset.
- Choose a Learning Model, Methodology, or Schema for training the dataset.
- Fitting the Model with proper parameters and Predicting a feasible outcome(likelihood).
- Determining the Model Accuracy Score.
- Report and Visualization of the predicted outcomes.
- If the results are not that convincing, then Tuning and Optimizing Model with necessary algorithms, is needed.
- Testing the Optimized Model and Reporting its whereabouts and results.
- After Previous.Step, if the obtained results are not still that convincing then "Repeat Previous→Previous.Step" with a more efficient technique.
- Summary Report on the Models' Metrics and Visualizations.
- Testing the Model with a random news feed from any source to verify its performance and how well it can guess correctly.

V.2 Feature engineering

Feature engineering—— is the process of using domain knowledge to extract features from raw data via data mining techniques. These features can be used to improve the performance of machine learning algorithms. Feature engineering can be considered as applied machine learning itself.

Feature Extraction—— It uses data reduction which allows the elimination of less important features. The collection of words obtained through tokenization is sorted to find the unique words. The unique words and the count will be stored separately for further processing.
For this research project, we selected the features *'text'* and *'label'* from our compiled dataset shown in figure 9 below.

Feature Selection—— We used the feature *'text'* as **input data** or data to be analyzed and the feature *'label'* as **target data** or result/output from the input data.

```
Index(['title', 'text', 'label'], dtype='object') (51233, 3)
```

	title	text	label
51228	LOL! TRUMP Responds To RACE-OBSESSED Congressm...	Donald Trump has been hit with celebrity backl...	FAKE
51229	Supreme Court faces 4-4 split in Obamacare con...	WASHINGTON (Reuters) - The Supreme Court on We...	TRUE
51230	UK aid minister to resign rather than be sacke...	LONDON (Reuters) - British aid minister Priti ...	TRUE
51231	After Kim Davis is jailed, marriage license is...	(CNN) With the clerk who had refused them in j...	REAL
51232	YOU WON'T BELIEVE THIS! CALIFORNIA GOVERNOR'S ...	I wrote AB 2466 because I want to send a mess...	FAKE

Figure 9: Compiled Dataset with Selected features.

Data Preprocessing—— To preprocess the data, we use a **TF-IDF Vectorizer** to transform the text to feature vectors that can be used as input to an estimator.

<u>What is a TF-IDF Vectorizer?</u>

TF-IDF is an abbreviation for Term Frequency Inverse Document Frequency. This is very common algorithm to transform text into a meaningful representation of numbers which is used to fit machine algorithm for prediction. Let's take sample example and explore two different spicy sparse matrix before go into deep explanation.

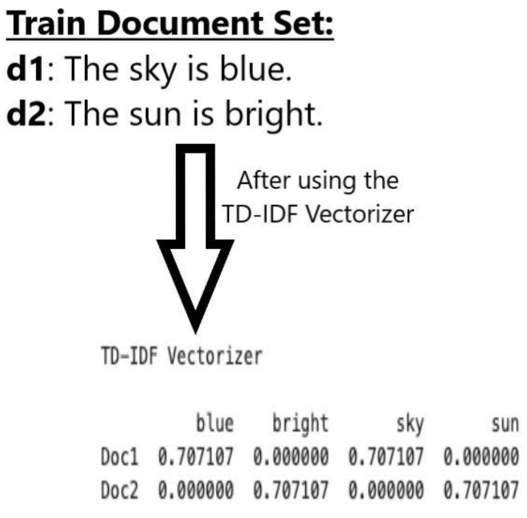


Figure 10: Text Document converted to Spicy Sparse Matrix of TF-IDF Vectorizer.

Here, we can see clearly that TF-IDF Vectorizer consider overall documents of weight of words. A vocabulary is a dictionary that converts each token (word) to a feature index in the matrix, each unique token gets a feature index. The TF-IDF Vectorizer converts a collection of raw documents into a matrix of TF-IDF features.

- **TF (Term Frequency)**: The number of times a word appears in a document is its Term Frequency. A higher value means a term appears more often than others, and so, the document is a good match when the term is part of the search terms.
- **IDF (Inverse Document Frequency)**: Words that occur many times in a document, but also occur many times in many others, maybe irrelevant. IDF is a measure of how significant a term is in the entire corpus.

Mathematical Understanding of TF-IDF——

TF-IDF is a measure of originality of a word by comparing the number of times a word appears in a document with the number of documents the word appears in. **TF-IDF** = *TF*(*t*,*d*) ⍰ *IDF*(*t*)—— where *TF*(*t*,*d*) is the number of times term *t* appears in a document *d* as shown:

$$TF(t,d) = \sum_{x \in d} fr(x,t) \tag{6}$$

where *fr*(*x*,*t*) is a simple function defined as:

$$fr(x,t) = \begin{cases} 1, & if \rightarrow x = t \\ 0, & otherwise \end{cases} \tag{7}$$

and **IDF**(*t*) is the Inverse Document Frequency which is computed as follow:

$$IDF(t) = \log\left(\frac{1+n}{1+DF(d,t)}\right) + 1 \tag{8}$$

where *n* is the number of documents and **DF**(*d*,*t*) is the Document Frequency of the term *t*.

Remark—— In **TfidfVectorizer** we consider **overall document weightage** of a word. It helps us in dealing with the most frequent words. Using it we can penalize them. TfidfVectorizer weights the word counts by a measure of how often they appear in the documents.

V.3 Training and Test Set Generation

The train-test split procedure is used to estimate the performance of machine learning algorithms when they are used to make predictions on data not used to train the model.

It is a fast and easy procedure to perform, the results of which allow you to compare the performance of machine learning algorithms for your predictive modeling problem. Although simple to use and interpret, there are times when the procedure should not be used, such as when you have a small dataset and situations where the additional configuration is required, such as when it is used for classification and the dataset is not balanced.[10]

Remark—— The train-test split procedure is appropriate when you have a very large dataset, a costly model to train, or require a good estimate of model performance quickly.

In this research project, we used a split percentage of Train: 80%, Test: 20% which is also called a **8:2 split ratio** on the compiled dataset (figure 4) with a shape of 51233 by 3. We split the dataset with the selected features shown in figure 9.

After splitting the dataset, we got:

- **X-Train**: with length *{40986}* which represents 80% of the dataset.
- **X-Test**: with length *{10247}* which represents 20% of the dataset.
- **Y-Train** and **Y-Test**: which are the target data for **X-Train** and **X-Test** respectively with the same split ratio.

V.4 Running the Classifier

Pre-Execution—— Before executing the classifier, we use **TF-IDF Vectorizer** to sanitize, transform and preprocess both **X-Train** and **X-Test** as mentioned in the above sections. Thus, to obtain both **TF IDF-Train** and **TF IDF-Test** respectively (with more features) which are **vectorized versions** of X-Train and X-Test.

Execution—— We use the obtained vectorized versions **TF _IDF-Train** and **TF IDF-Test** to train and test the Passive Aggressive Classifier with their respective target samples. While setting the PAC model, we configured some of its parameters with some values as shown below:

```
PassiveAggressiveClassifier(C=1.0, average=False, class_weight=None,
                            early_stopping=False, fit_intercept=True,
                            loss='hinge', max_iter=50, n_iter_no_change=5,
                            n_jobs=None, random_state=None, shuffle=True,
                            tol=0.001, validation_fraction=0.1, verbose=0,
                            warm_start=False)
```

Figure 11: Model Configuration while running.

VI Result Analysis

Evaluation Metrics—— To evaluate the performance of algorithms for fake news detection problem, various evaluation metrics have been used. In this subsection, we review the most widely used metrics for fake news detection. Most existing approaches consider the fake news problem as a classification problem that predicts whether a news article is fake or not:

- True Positive (**TP**): when predicted fake news pieces are actually annotated as fake news;
- True Negative (**TN**): when predicted true news pieces are actually annotated as true news;
- False Negative (**FN**): when predicted true news pieces are actually annotated as fake news;
- False Positive (**FP**): when predicted fake news pieces are actually annotated as true news.

By formulating this as a classification problem, we can define the following metrics,

$$Precision = \frac{|TP|}{|TP| + |FP|} \tag{9}$$

$$Recall = \frac{|TP|}{|TP| + |FN|} \tag{10}$$

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{11}$$

$$Accuracy = \frac{|TP| + |TN|}{|TP| + |TN| + |FP| + |FN|} \tag{12}$$

These metrics are commonly used in the machine learning community and enable us to evaluate the performance of a classifier from different perspectives. Specifically, accuracy measures the similarity between predicted fake news and real fake news. Precision measures the fraction of all detected fake news that are annotated as fake news, addressing the important problem of identifying which news is fake. However, because fake news datasets are often skewed, high precision can be easily achieved by making fewer positive predictions. Thus, recall is used to measure the sensitivity or the fraction of annotated fake news articles that are predicted to be fake news. F1 is used to combine precision and recall, which can provide an overall prediction performance for fake news detection. Note that for Precision, Recall, F1, and Accuracy, the higher the value, the better the performance.

The Receiver Operating Characteristics (**ROC**) curve provides a way of comparing the performance of classifiers by looking at the trade-off in the False Positive Rate (**FPR**) and the True Positive Rate (**TPR**). To draw the ROC curve, we plot the FPR on the x-axis and TPR along the y-axis. The ROC curve compares the performance of different classifiers by changing class distributions via a threshold. *TPR* and *FPR* are defined as follows (note that TPR is the same as recall defined above):

$$TPR = \frac{|TP|}{|TP| + |FN|} \tag{13}$$

$$FPR = \frac{|FP|}{|FP| + |TN|} \tag{14}$$

Based on the ROC curve, we can compute the Area Under the Curve (AUC) value, which measures the overall performance of how likely the classifier is to rank the fake news higher than any true news. Based on [15], AUC is defined as below.

$$AUC = \frac{\sum (n_0 + n_1 + 1 + r_i) - n_0(n_0 + 1)/2}{n_0 \times n_1} \tag{15}$$

where r_i is the rank of i^{th} fake news piece and $n_0(n_1)$ is the number of fake (true) news pieces. It is worth mentioning that *AUC* is more statistically consistent and more discriminating than accuracy [16], and it is usually applied in an imbalanced classification problem, such as fake news classification, where the number of ground truth fake news articles and true news articles have a very imbalanced distribution.

VI.1 Overview

In training, the model had an accuracy score of more than 97%. The model was able to predict whether an article was real or fake news 80% of time using the *Fake News Net*, *PolitiFact* and *Kaggle* dataset. This dataset was chosen because it was formatted differently than the Fake News Net dataset. A benefit of the *PolitiFact* portion of the dataset is that it had an almost equal amount of real vs fake news samples, This ensured that the model was not biased towards real or fake news in testing.

VI.2 Generation of Confusion-Matrix & Heat-Map

After training and testing the model, a confusion-matrix was generated and visualized in figure 12 as shown below:

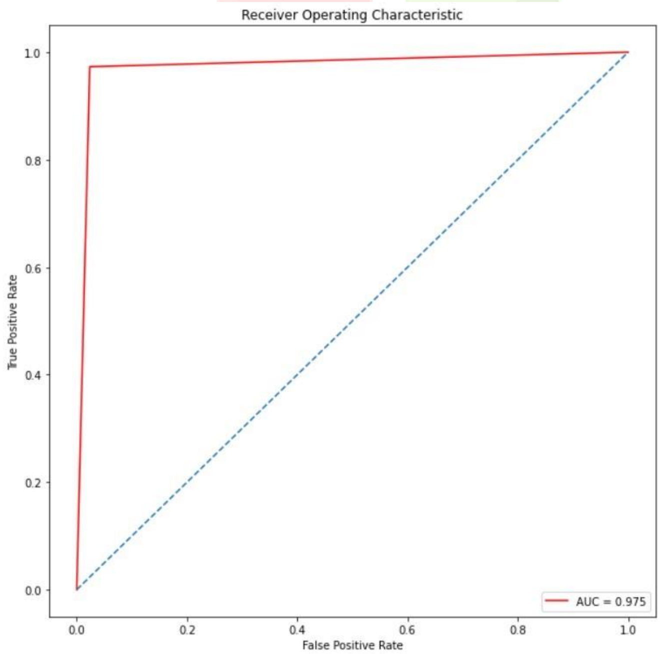
$$Confusion\ Matrix = \begin{matrix} 5168 & 125 \\ 134 & 4820 \end{matrix} \tag{16}$$

Figure 12: Heat-Map Visualization of PAC.

VI.3 Result analysis

```
----------------=====Model Report=====----------------

              precision    recall  f1-score   support

        FAKE       0.97      0.98      0.98      5293
        REAL       0.97      0.97      0.97      4954

    accuracy                           0.97     10247
   macro avg       0.97      0.97      0.97     10247
weighted avg       0.97      0.97      0.97     10247

------------------------------------------------------

::Confusion-Matrix::
 [[5168  125]
 [ 134 4820]]

------------------------------------------------------

Precision:      0.97

Recall:         0.97

Density:        0.605463

Dimensionality: 129377

------------------------------------------------------

Train Time:     0.536s

Test Time:      0.005s

Data-Preprocessing Time: 12.507s

------------------------------------------------------

Accuracy Score: 97.47%

------------------------------------------------------
```

Figure 13: Resulting Metrics from PAC Model.



VII Conclusion

The problem of fake news has gained attention in 2016, especially in the aftermath of the last US presidential elections. Recent statistics and research show that 62% of US adults get news on social media [6][5]. With the increasing popularity of social media, more and more people consume news from social media instead of traditional news media. However, social media has also been used to spread fake news, which has strong negative impacts on individual users and broader society.

It is certainly possible to classify news content into two types: fake news and real news, however, there will always be an inherent bias to this classification based on the researcher's own personal beliefs. Even though this is true, with tools like this research project it could be possible to at least cut down on the amount of objectively fake news that exists in the world today. With a preliminary result of 97%, this project could potentially contribute to accurately finding fake news and publicizing it, without the need for humans to have to do that work themselves. The Efficiency can be improved using about five classifier models like Support Vector Machines, logistic Regression, Logistic Regression CV, which can perform better classification and can give better accuracy. Using these classifiers, if the targets of the sample data are (REAL, REAL, FAKE, FAKE, REAL), then the output would be REAL as it is the majority. Apart from the classifier, we can also build a fact detector and a stance detector. A combination of all these tools would be the best way to classify the news accurately. Fake news detection is an emerging research area with few public datasets. We run our model on an existing dataset, showing that our model outperforms the original approach published by the authors of the dataset.

VIII Future Work

Our future work will be mostly focused on Sentiment and Emotion Analysis using some of the techniques outlined in this project. In our future work, we will run our model on a few other publicly available datasets, such as the LIAR and others which were released only recently, after we completed the current phase of our research.

References

[1] Alakesh Bora aka @alokesh985. *Passive Aggressive Classifiers*. GeeksForGeeks, 17-July-2020.

[2] Giuseppe Bonaccorso aka @giuseppebonaccorso. *ML Algorithms addendum: Passive Aggressive Algorithms*. www.bonaccorso.eu, 10 June 2017.

[3] Koby Crammer & al. *Online Passive-Aggressive Algorithms*. Journal of Machine Learning Research, March 2006.

[4] Cl´ement Bisaillon. *Fake and real news dataset*. Kaggle, March 2020.

[5] J. & al. Gottfried. *The 2016 presidential campaign–a news event that's hard to miss*. Pew Res Cent. 4, 2016.

[6] J. & al. Gottfried. *News use across social media platforms*. Pew Res Cent. 26, 2016.

[7] Jing Lu & Peilin Zhao & Steven C. H. Hoi. *Online Passive-Aggressive Active learning*. Springer Link, 22 March 2016.

[8] @IUT-Thesis-Group-Cmr & @IUT-Imperial-237. Machine Learning Project: Fake News Detector.

[9] Katharine Jarmul. *Detecting fake news with scikitlearn*. DataCamp Community, 2017.

[10] PhD Jason Brownlee. Train-Test Split for Evaluating Machine Learning Algorithms.

[11] PolitiFact. *"Why alex jones was banned from apple, facebook, youtube."*. PolitiFact.com, 07 August 2018.

[12] M. Rao. *"Exploiting network structure to detect fake news"*. Stanford University, School of Computer Science, 2018.

[13] M. Haag & M. Salam. *"Gunman in 'pizzagate' shooting is sentenced to 4 years in prison,"*. NyTimes, June 2017.

[14] A. Thota. *"Fake News Detection: A deep learning approach,"*. SMU, Data Science Review, 2018.

[15] David J Hand & Robert J Till. *A simple generalisation of the area under the roc curve for multiple class classification problems*. Machine learning, 2001.

[16] Charles X Ling & Jin Huang & Harry Zhang. *Auc: a statistically consistent and more discriminating measure than accuracy*.