A PROJECT REPORT ON

FAKE NEWS DETECTION USING MACHINE LEARNING

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE

BACHELOR OF ENGINEERING

In

COMPUTER ENGINEERING

Of

SAVITRIBAI PHULE PUNE UNIVERSITY

By

DHANASHREE GAIKWAD **Roll No.: 205B023**

TANMAY JAGTAP **Roll No.: 205B030**

SAVANI KHUTALE **Roll No.: 205B037**

KUMAR KOLHE **Roll No.: 205B039**

KUNAL PATIL **Roll No.: 205B040**

Under the guidance of

PROF S. S. PAWAR

DEPARTMENT OF COMPUTER ENGINEERING
SINHGAD COLLEGE OF ENGINEERING, PUNE-41
Accredited by NAAC

2020-21

Sinhgad Technical Education Society,
Sinhgad College of Engineering , Pune-41
Department of Computer Engineering

Sinhgad Institutes

Date:

CERTIFICATE

This is to certify that the project report entitled

"FAKE NEWS DETECTION USING MACHINE LEARNING"

Submitted by

Dhanashree Gaikwad	**Exam Seat No : S190234269**
Tanmay Jagtap	**Exam Seat No : S190234292**
Savani Khutale	**Exam Seat No : S190234319**
KUMAR KOLHE	**Exam Seat No : S190234321**
KUNAL PATIL	**Exam Seat No : S190234333**

is a bonafide work carried out by him/her under the supervision of Prof. S. S. Pawar and it is approved for the partial fulfillment of the requirements of Savitribai Phule Pune University , Pune for the award of the degree of Bachelor of Engineering (Computer Engineering) during the year 2020-21.

Prof. A. S. Kalaskar
Guide
Department of Computer Engineering

Prof. M. P. Wankhade
Head
Department of Computer Engineering

Dr. S. D. Lokhande
Principal
Sinhgad College of Engineering, Pune-41

Acknowledgement

We are profoundly grateful to Prof. S.S.Pawar for her expert guidance and continuous encouragement throughout to see that this project on “Fake News Detection Using Machine Learning” rights its target since its commencement to its completion.

We are also extremely grateful to our respected H.O.D. Prof. M. P. Wankhade for providing us with all the facilities and every help for smooth progress of our project.

We would also like to thank the committee staff of our group for timely help and inspiration for completion of this project.

At last we would like to thank all the unseen authors of various articles on the Internet, helping us to become aware of the research currently ongoing in this field and all other staff of DEPARTMENT OF COMPUTER ENGINEERING for providing help and support in our work.

Abstract

The easy access and exponential growth of the information available on social media networks has made it intricate to distinguish between false and true information. The easy dissemination of information by way of sharing has added to exponential growth of its falsification. The credibility of social media networks is also at stake where the spreading of fake information is prevalent. Thus, it has become a research challenge to automatically check the information viz a viz its source, content and publisher for categorizing it as false or true. Machine learning has played a vital role in classification of the information although with some limitations. This project reviews various Machine learning approaches in detection of fake and fabricated news. The limitation of such and approaches and improvisation by way of implementing deep learning is also reviewed.

List of Figures

List of Tables

Acronyms

ML: Machine Learning

UML : Unified Modelling Language

GUI : Graphical User Interface

SRS : System Requirement Specification

NLTK : Natural Language Tool Kit

RE : Regular Expression

TABLE OF CONTENTS

Chapter 1
Introduction

1.1 Background and Basics

Fake News contains misleading information that could be checked. This maintains lie about a certain statistic in a country or exaggerated cost of certain services for a country, which may arise unrest for some countries like in Arabic spring. There are organizations, like the House of Commons and the Crosscheck project, trying to deal with issues as confirming authors are accountable. However, their scope is so limited because they depend on human manual detection, in a globe with millions of articles either removed or being published every minute, this cannot be accountable or feasible manually. A solution could be, by the development of a system to provide a credible automated index scoring, or rating for credibility of different publishers, and news context. This project proposes a methodology to create a model that will detect if an article is authentic or fake based on its words, phrases, sources and titles, by applying supervised machine learning algorithms on an annotated (labeled) dataset, that are manually classified and guaranteed. Then, feature selection methods are applied to experiment and choose the best fit features to obtain the highest precision, according to confusion matrix results. We propose to create the model using different classification algorithms. The product model will test the unseen data, the results will be plotted, and accordingly, the product will be a model that detects and classifies fake articles and can be used and integrated with any system for future use.

1.2 Literature Survey

Table 1.1 Literature Survey

Sr. No.	Title of the paper	Authors	Publication	Year	Methods/ Techniques/ Algorithms	Finding /Limitation	Scope of Research
[1]	Fake News Detection Using ML approaches	Syed Ishfaq Manzoor; Jimmy Singla; Nikita	IEEE	2019	Classification Methods used	Determination Of algorithms used for fake news detection	Analysis of Various Machine Algorithms
[2]	Fake news Detection	M. Irfan Uddin	Hindawi Journal	2020	VM, KNN, Wang-CNN, and Wang-Bi-LSTM	Used to find the fake news	One direction for future work is to make these methods more.
[3]	Media-Rich Fake News Detection: A Survey	S. B. Parikh and P. K. Atrey	IEEE	2018	SVM	Survey of Algorithms done	Survey of ML Algorithms

1.3 Project Undertaken

1.3.1 Problem Definition

The Problem on social medias is widely spreading fake news across the world. Our aim is to find a solution to this problem.

1.4 Organization of Project Report

The project report is organized as follows:

Chapter 1

Chapter 1 is Introduction. It gives the background and basics of the project. It is followed by a detailed literature survey of similar works in the past. Problem statement and scope of the project are defined as well.

Chapter 2

Chapter 2 is Project Planning and Management. It has details of the system requirement specifications which include functional and non-functional requirements, system overview, deployment environment, external interface and other requirements. The

project process model applicable to this project is also mentioned. Cost estimate analysis and time line scheduling is done as well.

Chapter 3

Chapter 3 is Analysis and Design. It consists of idea matrix, mathematical model and feasibility analysis. All the analytical and design diagrams are also included in this chap- ter. These diagrams include use case diagram, activity diagram, architecture diagram, class diagram, ER diagram, sequence diagram, state transition diagram and deployment diagram.

Chapter 4

Chapter 4 focuses on Implementation and Coding . It describes the modules of the project and gives a rough idea about the coding part . It also includes database classes. This chapter covers the role of various subsystems/modules/classes along with implementation details listing of the code for the major functionalities

Chapter 5

Chapter 5 is Testing. In this chapter, test cases regarding different types of testing are given. The types of testing included are unit testing, integration testing and acceptance testing.

Chapter 6

Chapter 6 includes the GUI snapshots of final application and gives idea about the User Interface.

Chapter 7

Chapter 7 is the conclusion of Project.

Chapter 8

It has Future work if applicable related to the project.

Chapter 2
Project Planning and Management

2.1 Introduction

This chapter covers the project planning and management details. It also covers System Requirement specifications. SRS is considered as the base for the effort estimations and project scheduling.

2.2 System Requirement Specification (SRS)

2.2.1 Detail System Requirement Specifications(SRS)

2.2.1.1 System overview

Product Perspective

Analyzing and detecting fake news on the internet is one the hardest problem to be solved. The damage caused due to fake news on social media has increased due to the growth of the internet penetration in India, which has risen from 137 million internet users in 2012 to over 600 million in 2019.

In light of the recent incidents we also discover that fake news could have much more drastic effect even on country`s economy. So to minimize such news to create drastic effect, we have to verify fake news. Purpose of our project is to detect fake news.

Product Functions

1) Input title and body of the news article.
2) Check whether the text is manipulated to mislead the reader.
3) Notify whether news is true or false.

User Classes and Characteristics

1) Cyber Crime Cell

   It can be used to check whether an individual or organization is trying to mislead the public by publishing Fake News Articles.

2) General Public

   It is important for individuals to know whether the news they are consuming is trustworthy. As news plays an important role in lives of people involved in any profession nowadays.

Operating Environment

Operating environment for Fake News detector is as listed below:

1) Dataset : 21417 True and 23481 Fake News Articles
2) Operating System : Windows
3) Platform : Jupyter Notebook (Anaconda Navigator)
4) Libraries : Numpy, Pandas, Scikit-learn(CountVectorizer), Natural Language Toolkit (Stopwords, PorterStemmer), Re, Joblib, etc

Design and Implementation Constraints

1) Backend should be connected to database.
2) Processing time should be as low as possible.
3) Articles should not contain images or videos.
4) Articles should only be in English language.

User Documentation

- User manual

Assumptions and Dependencies

- We are assuming that the machine has the required resources (memory and processing power etc.) and capabilities to run the system.
- We are assuming that the system has the required packages and dependencies (Numpy, Pandas,Scikit-learn) to run the system.
- The system has a minimum RAM of 8GB to avoid timeout during model training.
- The user has updated system.

2.2.1.2 Functional Requirement

- Add title of the news article.
- Add text of the news article.

2.2.1.3 Non-Functional Requirements

Performance Requirements

Text uploading process should be as fast as possible. Also, the processing of data should be fast enough to get the results as soon as possible.

Safety Requirements

1. The backend could crash resulting in failure of the whole system.

2. Backup of electricity must be provided in case of Unintended power failure.

Security Requirements

1. The System could crash if the news article is too heavy.

Software Quality Attributes

Our software has many quality attributes as follow:

1) Availability: This software is freely available to all users. The accessibility of the software is easy for everyone.
2) Maintainability: After the deployment of the project if any error occurs then it can be easily maintained by the software developer.
3) User Friendly: Since, the software is a GUI application, the output generated is much user friendly in its behavior.
4) Integrity: Integrity refers to the extent to which access to software or data by unauthorized persons can be controlled.
5) Reliability: Our application is more reliable than previously used applications as it gives better accuracy.
6) Generalization: Our algorithm generalizes all the mentioned false news as one classifier which reduces creation of separate classifier.

2.2.1.4 Deployment Environment

1) Dataset : 9868 final features from the dataset
2) Operating System: Windows
3) Platform : Python 3.8.5
4) Libraries : Click 7.0, Flask 1.1.1, Gunicorn 19.9.0, Itsdangerous 1.1.0, Jinja2 2.10.1, Markupsafe 1.1.1, Werkzeug 0.15.6 etc

```
In [64]: final_features = []
         for i in cvfeatures:
             if i not in del_features:
                 final_features.append(i)
         len(final_features)

Out[64]: 9868

In [100]: joblib.dump(final_features,'final features') #Saving final_features

Out[100]: ['final features']
```

2.2.1.5 External Interface Requirements

User Interfaces

1. News Article Text Uploading page

   Button to upload the title and text.

2. Output showing whether News is true or false

   Bigger the body of the news articles, higher the accuracy.

Hardware Interfaces

Hardware interfaces for this application will run on windows and other operating systems as long as hardware requirements are met. For a good performance a system with standard configuration(Processor-Intel i5 10^{th} generation or above) along with a minimum RAM of 8GB will work fine, but for large news articles more RAM is required.

Software Interfaces

1. Operating System – This software does not require a particular Operating System(Windows, Mac, Linux, etc.) to run but only a web browser and internet connection because it is a web application.
2. Backend - Flask is used for backend.
3. Frontend – HTML and CSS for user interface.
4. Libraries – Matplotlib, Seaborn, wordcloud(WordCloud,STOPWORDS)(For Data Visualization).

Communication Interface

1) Web application runs on Web browser.
2) Button interface for uploading the text.
3) Notifying whether article is true or false.

2.2.1.6 Other Requirements

All the other Requirements are stated in the document.

2.3 Project Process Modeling

Iterative waterfall model is the best suited for this project. Iterative waterfall model can be thought of as incorporating the necessary changes to the classical waterfall model to make it usable in practical software development projects. It is almost same

as the classical waterfall model except some changes are made to increase the efficiency of the software development. The iterative waterfall model provides feedback paths from every phase to its preceding phases, which is the main difference from the classical waterfall model.

In our project, every phase is well defined and to be executed one after the other sequentially, like the waterfall model. But if need be, there is space for going back to previous stages making changes. Hence, iterative waterfall is to be used for this project.

2.4 Cost and Efforts Estimates

- Time estimates
  - The time estimate of this project is approximate 3 month.

2.5 Project Scheduling

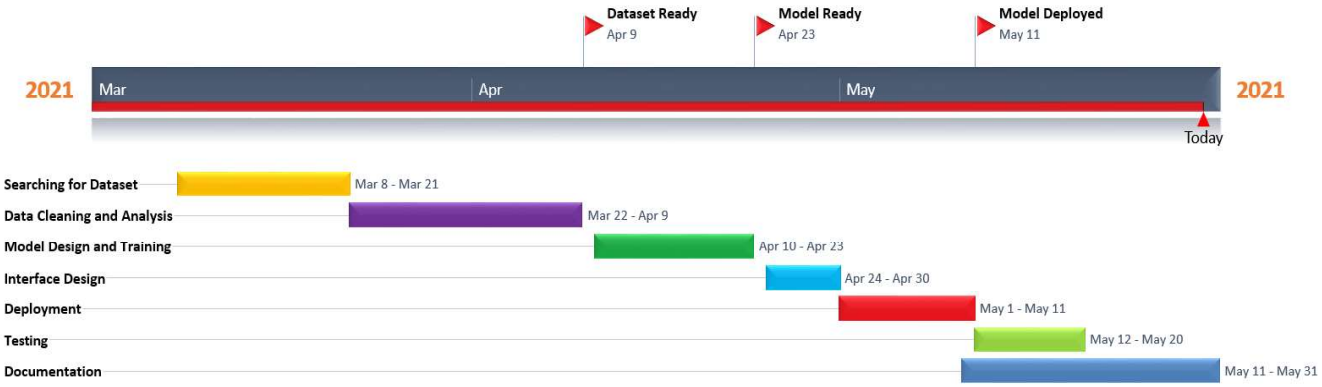


Fig. (2.1) Time Line Chart

Chapter 3
Analysis and Design

3.1 Introduction
This chapter covers the analysis and design of the considered system.

3.2 IDEA Matrix

Table 3.1 IDEA Matrix

Increase	• Efficiency in detection of Fake and misleading texts. • Recognition of number of false connects.	• Performance • Efficiency • Scalability • Extensibility
Improve	• Method of False information detection • Working of algorithm on larger datasets • Working of algorithm on text with number of special symbols	• Interoperability • Security • Crime detection
Ignored	• Storage required for the larger dataset	• Storage
Invent	• Algorithm optimized for many news subjects	• Algorithms
Deliver	• Deliverables : application detecting real-world fake news in press or media	• Flexible fake news detection application

Decrease	• Waste of time • Software maintenance can be done simultaneously	
Educate	• Educate project members	• Project Member
Evaluate	• Tight evaluation of news publishing houses for security purpose	• Throughput
Eliminate	• Any hardware up gradation	• Hardware up gradation
Accelerate	• Use of application on web by common people.	• Access to individual device • Innovation
Associate	• Use of open source helps to detect hoaxes very quickly • On-demand focus on particular news agency	• Open source
Avoid	• Continuous source checking by individuals	• Cost and manpower centric approach

3.3 Mathematical Model

Since the hypothesis function for logistic regression is sigmoid in nature hence, The First important step is finding the gradient of the sigmoid function. We can see from the derivation below that gradient of the sigmoid function follows a certain pattern.
Hypothesis Function-

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \frac{1}{1 + e^{-\boldsymbol{\theta}^T \boldsymbol{x}}}$$

3.4 Feasibility Analysis (NP Completeness Analysis)

The problem is to detect fake news articles. The main objective of our project is to provide a real-time solution for any possible textual piece to be classified as true or fake.

In the proposed system, we are planning to use logistic regression for detecting fake news. For Logistic Regression -

Training Complexity: O((f+1)csE)
Prediction Complexity: O((f+1)cs)
In these, f is the number of features, c is the number of classes(possible outputs), s is the number of samples in our dataset ,E is the number of epochs you are willing to run the gradient descent(whole passes through dataset),
where,
f=9868
c=1(for binary classification)
s=44898
E=1

Hence O((f+1)cs) runs in polynomial time complexity. The algorithm falls in P.

Hence, the given problem is NP.

3.5 UML Diagrams

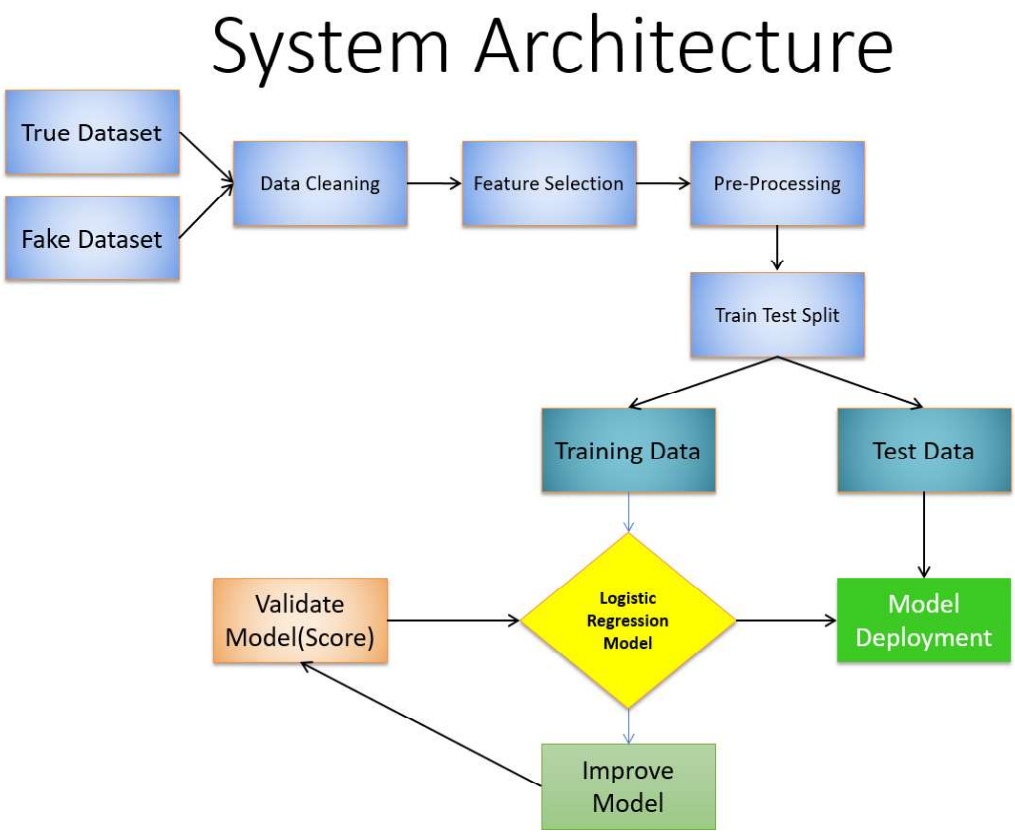


Fig. (3.1) Architecture Diagram

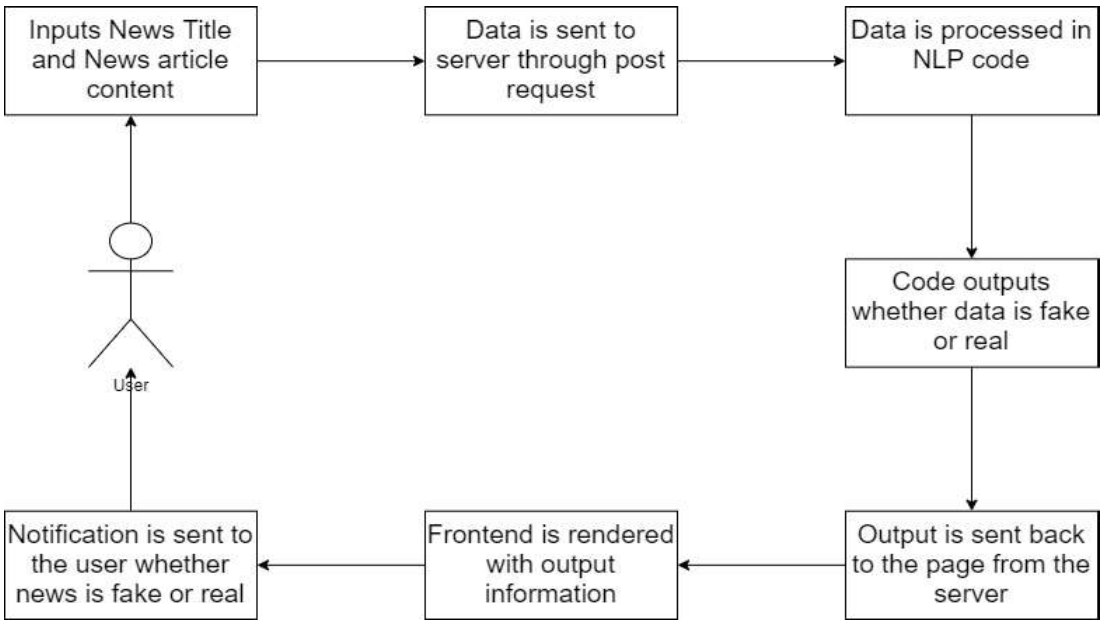


Fig. (3.2) Deployment Diagram

Chapter 4
Implementation and Coding

4.1 Introduction

This chapter covers the role of various subsystems/modules/classes along with implementation details listing of the code for the major functionalities.

4.2 Database Schema

It consists of fake news and real news of various domains related to politics, sports, environment, news across the world, education, etc. These fake news are selected because they have a significant impact on public information.

For our fake news detection method, fake news and real news datasets are required for training.

4.3 Operational Details

The project has two stages / Modules : Training the model from datasets and testing the news for checking if it is actually detecting whether the news is fake or real based on training.

Major Classes in the project are as follows:

1: Importing Required Libraries

Libraries used are Natural Language Toolkit, PorterStemmer, stopwords, sklearn, CountVectorizer, WordCloud, Joblib, seaborn and matplotlib .

Description of libraries used:

PorterStemmer – It is used for stemming. Stemming is the process of removing the suffixes from the words. It is desirable because sometimes the words mean the same so we can remove the suffixes from the word.

Stopwords - Stopwords are the English words which does not add much meaning to a sentence. They can safely be ignored without sacrificing the meaning of the sentence. For example, the words like the, he, have etc.

Seaborn - **Seaborn** is a **Python** data visualization **library** based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

Matplotlib - **Matplotlib** is a cross-platform, data visualization and graphical plotting library for **Python** and its numerical extension NumPy.

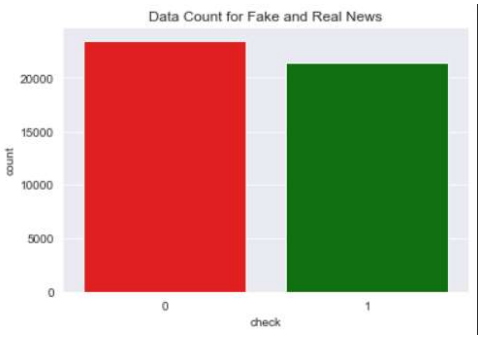
2: Importing Data Set

As mentioned earlier, we have list of fake and real news in our database. So in this class, we are loading the batch of both type of news and each one is randomly selected. The names of the datasets are ‘true.csv’ and ‘fake.csv’.

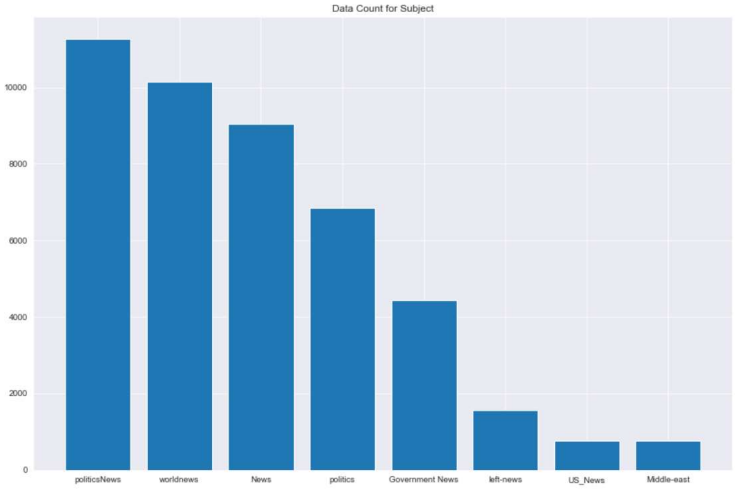
3: Data cleaning and Visualization

The data set consists of columns like title, text, subject, date. We have checked for null values. We have introduced new column in the data set to check if news is fake or real. True is assigned a value 1 and False is assigned a value 0.

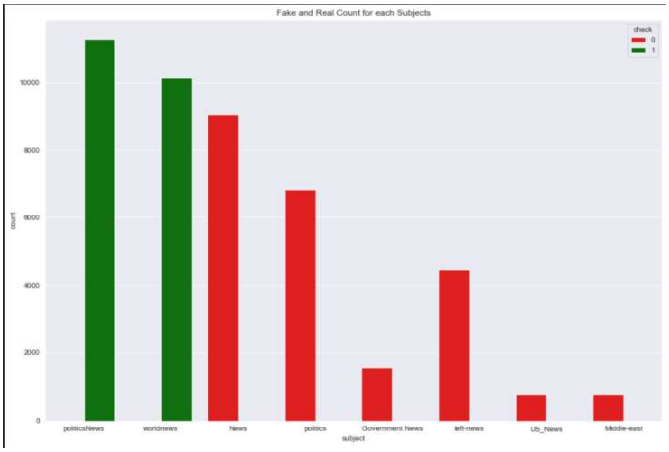
The count of fake and real news is equal hence the dataset is balanced.



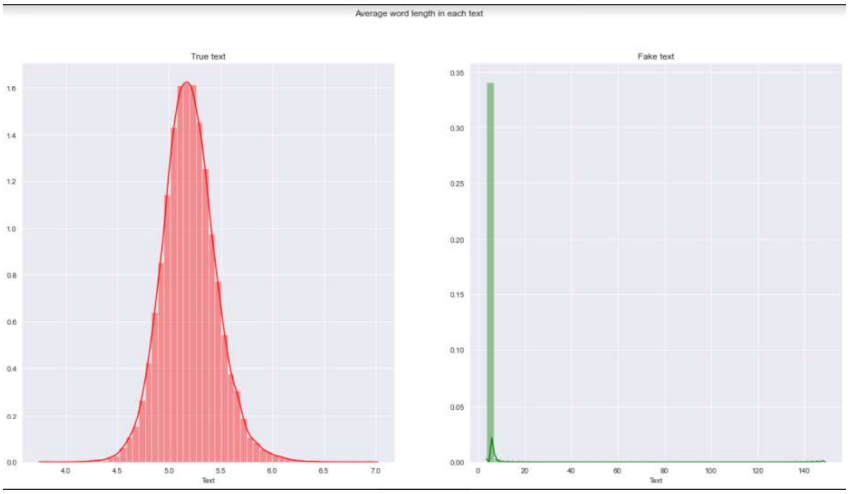
The data set consists of various news like political news, World news, News, Left-News, Government News, US News, etc.



We counted the count of fake and real datasets subjects and found that subject topics for fake and real news are totally different. So we dropped the subject column.



We merged the ‘title’ and ‘text’ column under one ‘column’ called ‘text’. Made a word cloud for real and fake text. We counted the average word length in each text.

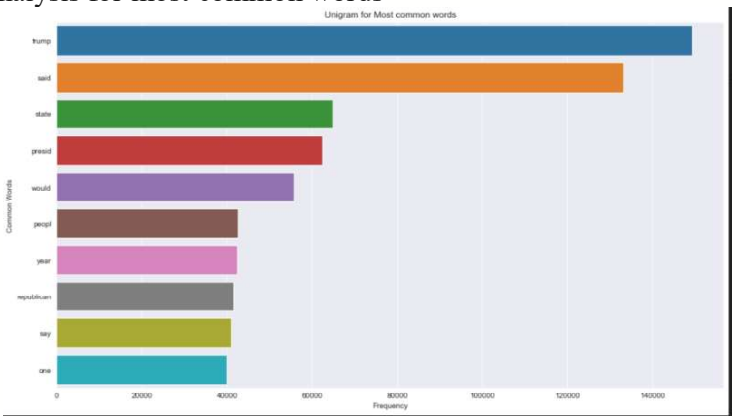


The distribution of words in true and fake news seems to be a bit different. 2500 characters are mainly present in original text category while around 5000 characters are mainly present in fake text category.

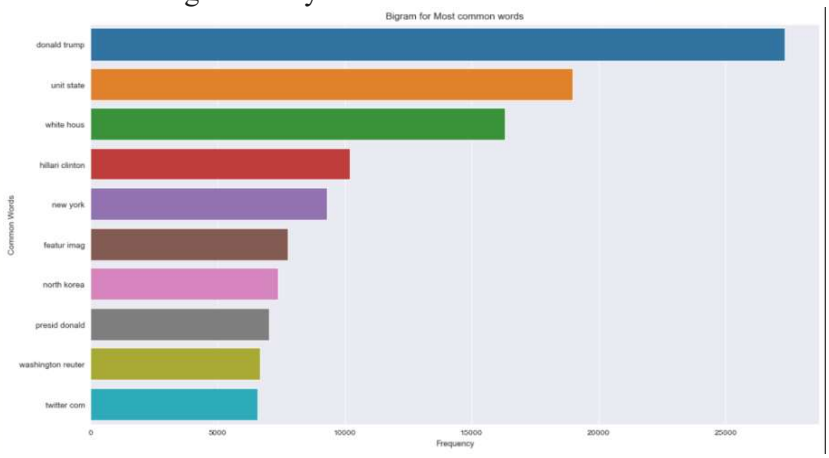
4 : Preprocessing of Data

Using PorterStemmer, We have removed all the characters like (, [] () . /) other than Numbers and alphabets. Using lower() function we have lowered all the characters. Then we have removed the words which do not affect the sentence meaning eg. Pronouns, articles, etc. After that we have appended those words to corpus[] list. We have transformed each and every text in corpus to a vector using CountVectorizer. Here we have considered 10000 most common words with ngram ranging 1-3 in corpus. Then we did unigram, biagram and trigram analysis for Most Common words.

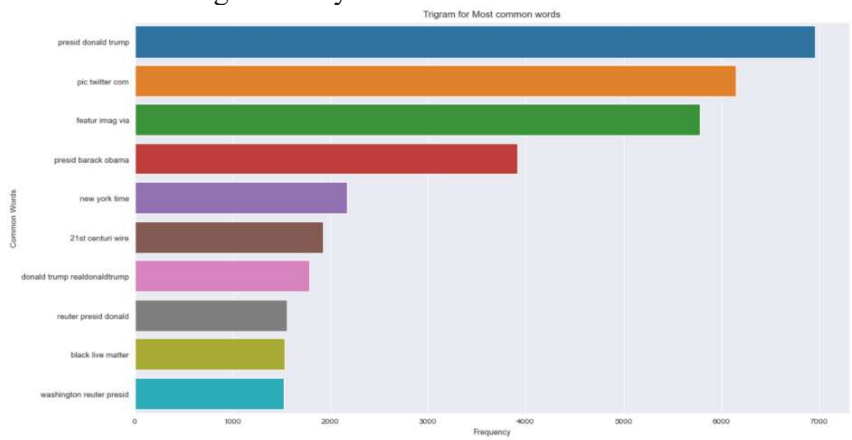
Unigram Analysis for most common words



Bigram analysis for most common words



Trigram analysis for most common words



From the cvfeatures there are few countable numerical values which do not have much significance for model training
Hence we have removed such values. Finally we have saved the final features.

5: Logistic Regression

Logistic regression is a supervised learning classification algorithm used to predict the probability of a target variable. The nature of target or dependent variable is dichotomous, which means there would be only two possible classes. Logistic regression can be divided into following types –

- Binary or Binomial
- Multinomial
- Ordinal

Using Logistic Regression we got 99.67% accuracy.

We also tried another algorithm called Multinomial.

Multinomial - In such a kind of classification, dependent variable can have 3 or more possible ***unordered*** types or the types having no quantitative significance. For example, these variables may represent “Type A” or “Type B” or “Type C”.

We got 96.46% accuracy using MultinomialNB hence, Using Logistic Regression is our final model.

4.4 Screenshots

1: Importing Required Libraries

Importing the Required Libraries

```
[ ]  import pandas as pd
     import re
     from nltk.corpus import stopwords
     from nltk.stem.porter import PorterStemmer
     from sklearn.feature_extraction.text import CountVectorizer
     from wordcloud import WordCloud, STOPWORDS
     import joblib
     import matplotlib.pyplot as plt
     import seaborn as sns
```

2: Importing Data Set

Importing the Dataset

```
[ ]  true = pd.read_csv('true.csv')
     fake = pd.read_csv('fake.csv')
```

3: Data Cleaning and Visualization

	title	text	subject	date
0	As U.S. budget fight looms, Republicans flip t...	WASHINGTON (Reuters) - The head of a conservat...	politicsNews	December 31, 2017
1	U.S. military to accept transgender recruits o...	WASHINGTON (Reuters) - Transgender people will...	politicsNews	December 29, 2017
2	Senior U.S. Republican senator: 'Let Mr. Muell...	WASHINGTON (Reuters) - The special counsel inv...	politicsNews	December 31, 2017
3	FBI Russia probe helped by Australian diplomat...	WASHINGTON (Reuters) - Trump campaign adviser ...	politicsNews	December 30, 2017
4	Trump wants Postal Service to charge 'much mor...	SEATTLE/WASHINGTON (Reuters) - President Donal...	politicsNews	December 29, 2017

	title	text	subject	date	check
0	Irish border row thwarts May bid to clinch Bre...	BRUSSELS (Reuters) - Prime Minister Theresa Ma...	worldnews	December 3, 2017	1
1	TICKING TIME BOMB: Why More Young Muslims In T...	These are statistics are shocking and very tel...	politics	Mar 23, 2016	0
2	As Syria war tightens, U.S. and Russia militar...	AL UDEID AIR BASE, Qatar (Reuters) - Even as t...	worldnews	August 24, 2017	1
3	CHRISTIAN HIGH SCHOOL Told By State They Are N...	The drip drip drip of communism Leftists are s...	politics	Dec 7, 2015	0
4	California voters turn down drug pricing initi...	LOS ANGELES (Reuters) - California voters turn...	politicsNews	November 9, 2016	1

	check	Text
0	1	Irish border row thwarts May bid to clinch Bre...
1	0	TICKING TIME BOMB: Why More Young Muslims In T...
2	1	As Syria war tightens, U.S. and Russia militar...
3	0	CHRISTIAN HIGH SCHOOL Told By State They Are N...
4	1	California voters turn down drug pricing initi...

Balanced data Set

4: Preprocessing of Data

Preprocessing of Data

```
[1] ps = PorterStemmer()
    corpus = []
    for i in range(len(df)):
        review = re.sub('[^a-zA-z_0-9]',' ',df['Text'][i]) #Removing all charecters(,[]{}./ etc) other than numbers and alphabets
        review = review.lower() #lowering the charecters
        review = review.split()

        #Removing words which does not affect the sentence meaning. eg. pronouns, articles, etc
        review = [ps.stem(word) for word in review if not word in stopwords.words('english')]
        review = ' '.join(review)
        corpus.append(review)
```

```
[ ] corpus[1]
```

Transforming each and every text in corpus to vector using CountVectorizer
Here we have considered 10000 most common words with ngram ranging 1-3 in corpus

```
cv = CountVectorizer(max_features = 10000, ngram_range = (1,3))
textv = cv.fit_transform(corpus)
```

Top 10 most common words

```
[ ] sum_words = textv.sum(axis=0)
```

```
[ ] words_freq = [(word, sum_words[0, idx]) for word, idx in cv.vocabulary_.items()]
    words_freq =sorted(words_freq, key = lambda x: x[1], reverse=True)
```

```
[ ] words_freq[:10]
```

```
[ ] def get_top_text_ngrams(corpus, n, g):
        vec = CountVectorizer(ngram_range=(g, g)).fit(corpus)
        bag_of_words = vec.transform(corpus)
        sum_words = bag_of_words.sum(axis=0)
        words_freq = [(word, sum_words[0, idx]) for word, idx in vec.vocabulary_.items()]
        words_freq =sorted(words_freq, key = lambda x: x[1], reverse=True)
        return words_freq[:n]
```

5: Logistics Regression

Training model using Logistic Regression Algorithm

```
[ ] from sklearn.linear_model import LogisticRegression
```

```
[ ] model = LogisticRegression()
    model.fit(X_train,y_train)
```

```
C:\Users\jagta\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:762: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
LogisticRegression()
```

```
[ ] model.score(X_test,y_test)*100
```

```
99.67037861915368
```

Training model using another algorithm - MultinomialNB Algorithm

```
[ ] from sklearn.naive_bayes import MultinomialNB
    model1 = MultinomialNB()
    model1.fit(X_train,y_train)
```

```
MultinomialNB()
```

```
[ ] model1.score(X_test,y_test)*100
```

```
96.46325167037863
```

Here we got the 96.46% accuracy

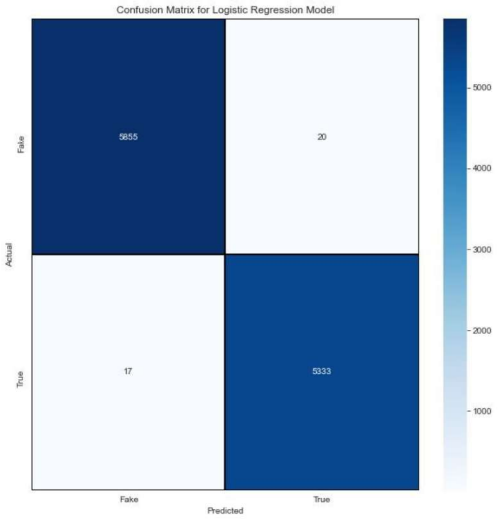
Chapter 5
Testing

5.1 Introduction

This chapter covers the testing approach used and the test cases.
We divided the dataset for training and testing.
75% was used for training while 25% for testing.

5.2 Testing using confusion matrix

Logistic regression is used to predict how many were true and how many were false.
For that we used confusion matrix to compare predicted output with actual output.
Following were the results:



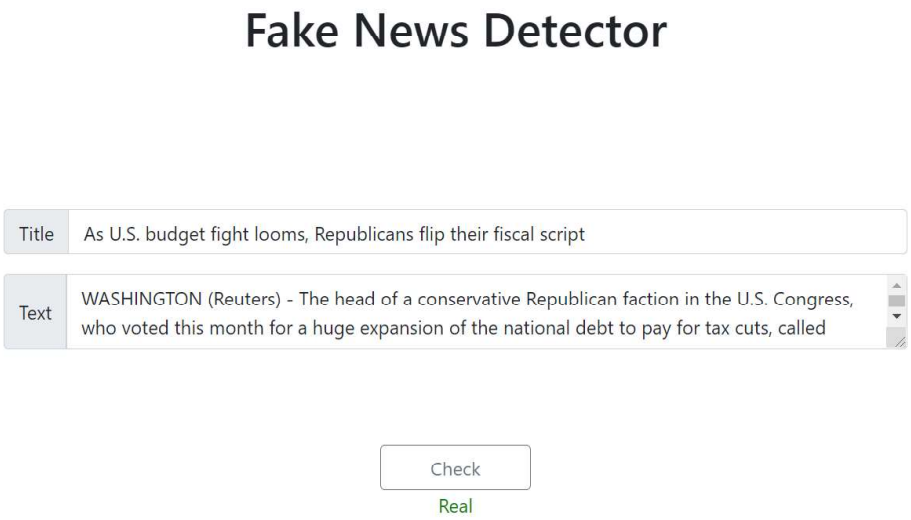
Out of the total, about 5855 which were predicted as fake outputs matched with the actual outputs, i.e. detected correctly that the news was fake. Only 17 results did not match.
So, using logistic regression we got an accuracy of 99.67%.

Chapter 6
Results and Discussions

6.1 Main GUI snapshots

6.1.1 GUI Page 1

6.1.2 GUI Page 2



6.1.3 GUI Page 3

6.2 Discussions

Recent political events have lead to an increase in the popularity and spread of fake news. As demonstrated by the widespread effects of the large onset of fake news, humans are inconsistent if not outright poor detectors of fake news. With this, efforts have been made to automate the process of fake news detection. The most popular of such attempts include “blacklists” of sources and authors that are unreliable. While these tools are useful, in order to create a more complete end to end solution, we need to account for more difficult cases where reliable sources and authors release fake news. As such, the goal of this project was to create a tool for detecting the language patterns that characterize fake and real news through the use of machine learning and natural language processing techniques. The results of this project demonstrate the ability for machine learning to be useful in this task. We have built a model that catches many intuitive indications of real and fake news as well as an application that aids in the visualization of the classification decision.

CONCLUSION

We proposed a fake news detection model using NLP(Natural language processing) which will help people to not be prey to false information. Social media and news outlets publish fake news to increase readership or as part of psychological warfare. In general, the goal is profiting through clickbaits.

The data we used in our work is collected from the World Wide Web and contains news articles from various domains to cover most of the news rather than specifically classifying political news. The primary aim of the research is to identify patterns in text that differentiate fake articles from true news.

We took a Fake and True News dataset, implemented a data cleaning function, split the data to train and test. Using Count Vectorizer for preprocessing, logistic regression for confusion matrix to test the model , we ended up obtaining an accuracy of 96.46%.

With many people trying to inject more false news/information on the internet, very few are using advanced technology to fight such scams. Thus using this model we can prevent spread of false news by helping people to detect if it is true or not.

FUTURE WORK

In the project, we have successfully detected the fake news articles using logistic regression. Previously, approaches like state of art were being used for this work. Earlier methods were so time consuming and also was not so efficient. In this project, we have achieved significant accuracy for fake news detection. In future, technologist can work on improving the accuracy and on the real time detection of fake news.

REFERENCES

[1] Yash Shukla, Nalini Yadav, Akshaya Hari, "An Unique Approach For Detection of Fake News using Machine Learning", International Journal for Research in Applied Science & Engineering Technology, DOI: 10.22214/ijraset.2019.6087, June 2019.

[2] Swetha subrahmanian, Sruthy , Biji , "Basic Introduction to Fake News Detection", International Journal for Research in Applied Science & Engineering Technology, DOI :10.22214/ijraset.2021.33360 ,March 2021.

[3]Dataset Used was taken from- https://www.kaggle.com/clmentbisaillon/fake-and-real-news-dataset

[4] https://github.com/nishitpatel01/Fake_News_Detection

[5] https://en.wikipedia.org/wiki/Fake_news

[6] https://en.wikipedia.org/wiki/Machine_learning

[7] https://blog.paperspace.com/fake-news-detection/

[8] https://www.analyticsvidya.com

[9] https://github.com/likeaj6/FakeBananas