

PROFESSIONAL TRAINING REPORT
At
SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Submitted in partial fulfillment of the requirements for the
award of Bachelor of Engineering Degree in

Information Technology

By

RITHWIK.E

Reg.No.39120096



DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING

**SATHYABAMA INSTITUTE OF SCIENCE
AND TECHNOLOGY**

(DEEMED TO BE UNIVERSITY)

(Established under Section 3 of UGC Act, 1956)

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai-600119

www.sathyabamauniversity.ac.in

APRIL 2022



SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade “A” by NAAC
(Established under Section 3 of UGC Act, 1956)
JEPPIAAR NAGAR, RAJIV GANDHI SALAI
CHENNAI-600119
www.sathyabama.ac.in

DEPARTMENT OF INFORMATION TECHNOLOGY

BONA FIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of RITHWIK (**Reg. No: 39120096**) who carried out the project entitled “**FAKE NEWS CLASSIFICATION USING NLP**” under my supervision from March 2022 to April 2022.

Internal Guide

Dr.S.GOWRI,M.E.,Ph.D

Head of the Department

Dr. R. SUBHASHINI, M.E., Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I, RITHWIK (Reg.No.39120076) here by declare that the Project report entitled "**FAKE NEWS CLASSIFICATION USING NLP**" done by me under the guidance of the **Dr.S.GOWRI** is submitted in partial fulfillment of the requirements for the award of Bachelor of degree in Information Technology.

DATE:

PLACE:



SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.SASIKALA, M.E., Ph.D., Dean, School of Computing** and **Dr.R.SUBHASHINI, M.E.,Ph.D., Head of the Department, Dept.of Information Technology** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.S.GOWRI,M.E.,Ph.D.,** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **DEPARTMENT OF INFORMATION TECHNOLOGY** who were helpful in many ways for the completion of the project.

TRAINING CERTIFICATE



ABSTRACT

The phenomenon of Fake news is experiencing a rapid and growing progress with the evolution of the means of communication and Social media. Fake news detection is an emerging research area which is gaining big interest. It faces however some challenges due to the limited resources such as datasets and processing and analysing techniques. In this work, we propose a system for Fake news detection that uses machine learning techniques. We used term frequency-inverse document frequency (TF-IDF) of bag of words and n-grams as feature extraction technique, and Support Vector Machine (SVM) as a classifier. We propose also a dataset of fake and true news to train the proposed system. Obtained results show the efficiency of the system. In this work, we propose a system for Fake news detection that uses machine learning techniques. We used term frequency-inverse document frequency (TF-IDF) of bag of words and n-grams as feature extraction technique, and Support Vector Machine (SVM) as a classifier. We propose also a dataset of fake and true news to train the proposed system. Obtained results show the efficiency of the system In our modern era where the internet is ubiquitous, everyone relies on various online resources for news. Along with the increase in the use of social media platforms like Facebook, Twitter, etc. news spread rapidly among millions of users within a very short span of time. The spread of fake news has far-reaching consequences like the creation of biased opinions to swaying election outcomes for the benefit of certain candidates. Moreover, spammers use appealing news headlines to generate revenue using advertisements via click-baits. In this paper, we aim to perform binary classification of various news articles available online with the help of concepts pertaining to Artificial Intelligence, Natural Language Processing and Machine Learning. We aim to provide the user with the ability to classify the news as fake or real and also check the authenticity of the website publishing the news.

TABLE OF CONTENTS

CHAPTER No.	TITLE	PAGE No
	ABSTRACT	06
	LIST OF FIGURES	09
	LIST OF ABBREVIATION	10
1	INTRODUCTION	11
	1.1 MACHINE LEARNING AND NLP	11
	1.1.1 MACHINE LEARNING	11
	1.1.2 NATURAL LANGUAGE PROCESSING	12
	1.1.2.1 STAGES OF NLP	12
	1.1.2.2 LEXICAL ANALYSIS	12
	1.1.2.3 SYNTACTIC ANALYSIS	12
	1.1.2.4 SEMANTIC ANALYSIS	13
	1.1.2.5 DISCOURSE INTEGRATION	13
	1.1.2.6 PRAGMATIC ANALYSIS	
2	AIM AND PROJECTSCOPE	14
	2.1 AIM	14
	2.2 PROJECT SCOPE	14
3	SYSTEM SPECIFICATION	15
	3.1 HARDWARE REQUIREMENTS	15
	3.2 SOFTWARE REQUIREMENTS	15
4	CONCEPTS INVOLVED	
	4.1 TOKENIZATION	16
	4.2 STEMMING	16
	4.3 STOPWORD REMOVAL	17
	4.4 VECTORIZATION	17
	4.5 CLASSIFICATION ALGORITHM	18
	4.6 LOGISTIC REASONING	19
5	MODULE IMPLEMENTATION	20
	5.1 STATIC SEARCH IMPLEMENTATION	20
	5.2 DYNAMIC SEARCH IMPLEMENTATION	20
	5.3 SYSTEM ARCHITECTURE	21
6	RESULT AND DISCUSSION	22
	6.1 RESULT AND DISCUSSION	22
	6.2 PERFORMANCE ANALYSIS	23

7	CONCLUSION	24
7.1	RERENCE	24
7.2	APPENDIX A	25
7.3	APPENDIX B	31

LIST OF FIGURES

FIGURE No.	TITLE	PAGE No.
1.1	NLP Representation	12
1.2	Tokenization	16
1.3	Stemming	17
1.4	vectorization	18
1.5	Logistic regression graph	19
1.6	System architecture	21

LISTOFABBREVIATIONS

NLP	-	natural language processing
NLTK	-	natural language tool kit
CV	-	count vectorizer
AI	-	artificialintelligence

1.INTRODUCTION

1.1 MACHINE LEARNING AND NLP

1.1.1 Machine learning

Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to perform the task. Machine learning is closely related to computational statistics, which focuses on making predictions using computers. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T , as measured by P , improves with experience E ." This is Alan Turing's definition of machine learning.

Deep learning is a class of machine learning algorithms that utilizes a hierarchical level of artificial neural networks to carry out the process of machine learning. The artificial neural networks are built like the human brain, with neuron nodes connected together like a web. While traditional programs build analysis with data in a linear way, the hierarchical function of deep learning systems enables machines to process data with a nonlinear approach.

The word "deep" in "deep learning" refers to the number of layers through which the data is transformed. More precisely, deep learning systems have a substantial credit assignment path (CAP) depth. The CAP is the chain of transformations from input to output. CAPs describe potentially causal connections between input and output. For a feedforward neural network, the depth of the CAPs is that of the network and is the number of hidden layers plus one (as the output layer is also parameterized). For recurrent neural networks, in which a signal may propagate through a layer more than once, the CAP depth is potentially unlimited.

Deep learning architectures such as deep neural networks, deep belief networks, recurrent neural networks and convolutional neural networks have been applied to fields including computer vision, speech recognition, natural language processing, audio recognition, social network filtering, machine translation, bioinformatics, drug design, medical image analysis, material inspection and board game programs, where they have produced results comparable to and in some cases superior to human experts.

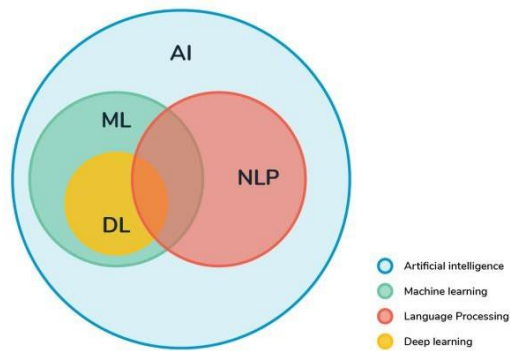


Fig. 1 : Graphical representation of relationship between various fields in artificial intelligence

1.1.2 Natural Language Processing

NLP is an area of computer science and artificial intelligence concerned with the interactions between computers and human (natural) languages, in particular how to program computers to fruitfully process large amounts of natural language data.

Natural language processing (NLP) is a subfield of linguistics, computer science, information engineering, and artificial intelligence concerned with the interactions between computers and human (natural) languages, in particular how to program computers to process and analyse large amounts of natural language data.

1.1.2.1 Stages in NLP

1.1.2.2 Lexical Analysis

Lexical Analysis involves identifying and the structure of words. Lexicon of a language means the collection of words and phrases in a language. Lexical analysis is dividing the whole chunk of txt into paragraphs, sentences, and words.

1.1.2.3 Syntactic Analysis

Syntactic Analysis involves analysis of words in the sentence for grammar and arranging words in a manner that shows the relationship among the words. The sentence such as “The school goes to boy” is rejected by English syntactic analyser

1.1.2.4 Semantic Analysis

Semantic Analysis draws the exact meaning or the dictionary meaning from the text. The text is checked for meaningfulness. It is done by mapping syntactic structures and objects in the task domain. The semantic analyse disregards sentence such as “hot ice-cream”

1.1.2.5 Discourse Integration

The meaning of any sentence depends upon the meaning of the sentence just before it. In addition, it also brings about the meaning of immediately succeeding sentence. So in Discourse Integration gives the meaning based on all the sentences given before it. Eg. Consider the sentence “Water is flowing on the bank of the river” But bank has two meanings One Financial Institute and Two River of the bank here System has to consider the second meaning.

1.1.2.6 Pragmatic Analysis

During this, what was said is re-interpreted on what it actually meant. It involves deriving those aspects of language which require real world knowledge

Pragmatic analysis in NLP can be defined as the set of logical and linguistic tools with the help of which analysts can develop systematic accounts of discursive political interactions. They attempt to classify the full variety of the inferences that any hearer or reader can make when encountering the locations of the author or speaker

Pragmatic Analysis is part of the process of extracting information from text. Specifically, it's the portion that focuses on taking structures set of text and figuring out what the actual meaning was. It actually comes from the field of linguistics (as a lot of NLP does), where the context is considered from the text.

2. AIM AND PROJECTSCOPE

2.1 AIM:

This project aims at building a model which classifies whether the news given is genuine news or false news by using Natural Language Processing

2.2 PROJECTSCOPE:

Scope The usage of this system greatly reduces the time required to search for a place leading to quicker decision making with respect to places to visit. Used to view the location view (the user can even zoom in and zoom out to get a better view) as well as 360 degree image embedded in the application. The System makes use of weather underground API for fetching the details of weather at accuracy

The user can also find the paths to follow to reach the final destination in map which gives a better view to the users. It becomes convenient for users to book their tour via website instead of visiting agency ultimately saves time and money.

3. SYSTEM SPECIFICATION

3.1 HARDWARE REQUIREMENT

- ☐ RAM: 4GB
- ☐ Processor Intel Core i5 and above

3.2 SOFTWARE REQUIREMENT

- ☐ Jupiter note
- ☐ Anaconda prompt
- ☐ Visual studio code

3.3 SYSTEM CONFIGURATION

This project can run on commodity hardware. We ran entire project on an Intel I5 processor with 8 GB Ram, 2 GB Nvidia Graphic Processor, It also has 2 cores which runs at 1.7 GHz, 2.1 GHz respectively. First part of the is training phase which takes 10-15 mins of time and the second part is testing part which only takes few seconds to make predictions and calculate accuracy.

4. CONCEPTS INVOLVED

4.1 TOKENIZATION

This process divides a large piece of continuous text into distinct units or tokens basically this process is often known as tokenization.

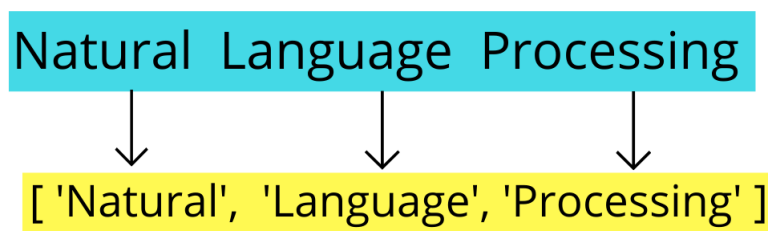
NLTK

provides a function called `word_tokenize()` for splitting strings into tokens (nominally words). It splits tokens based on white space and punctuation.

Requirement:

```
from nltk.tokenize import word_tokenize.
```

Tokenization



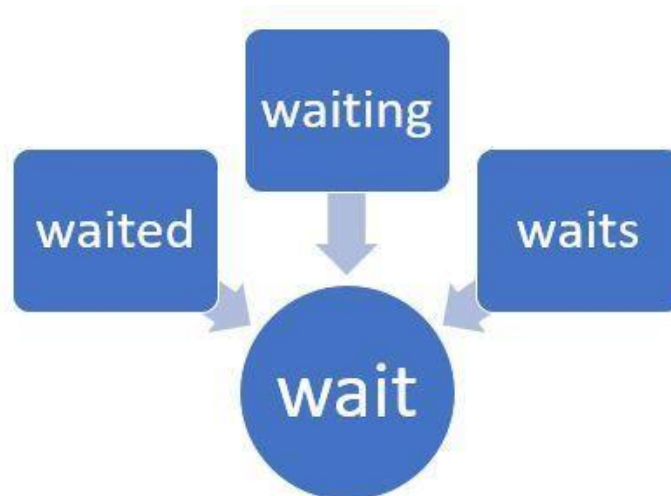
1.2 tokenization

4.2 STEMMING

This is the idea of removing the suffix of a word and reducing different forms of a word to a core root.

REQUIREMENTS :

- stemmer from `nltk.stem` package
There are different stemmers in this package
 - 1) Snowball
 - 2) Porter
 - 3) Lancaster



1.3 stemming

4.3 STOPWORDS REMOVAL

A stop word is a commonly used word (such as “the”, “a”, “an”, “in”) that a search engine has been programmed to ignore.

REQUIREMENTS :

- Import stopwords from
nltk.corpus
- ```
>>>from nltk.corpus import
stopwords
>>stopwords.words(„english“)
```

### 4.4 VECTORIZATION

The scikit-learn library offers easy-to-use tools to perform feature extraction of your text data that is Vectorization. The vectorization is a technique used to convert textual data to numerical format. Using vectorization, a matrix is created where each column represents a feature and each row represents an individual review.

TF (Term Frequency)

Term Frequency is defined as how frequently the word appear in the document

$$TF(t, d) = \frac{\text{number of times } t \text{ appears in } d}{\text{total number of terms in } d}$$

$$IDF(t) = \log \frac{N}{1 + df}$$

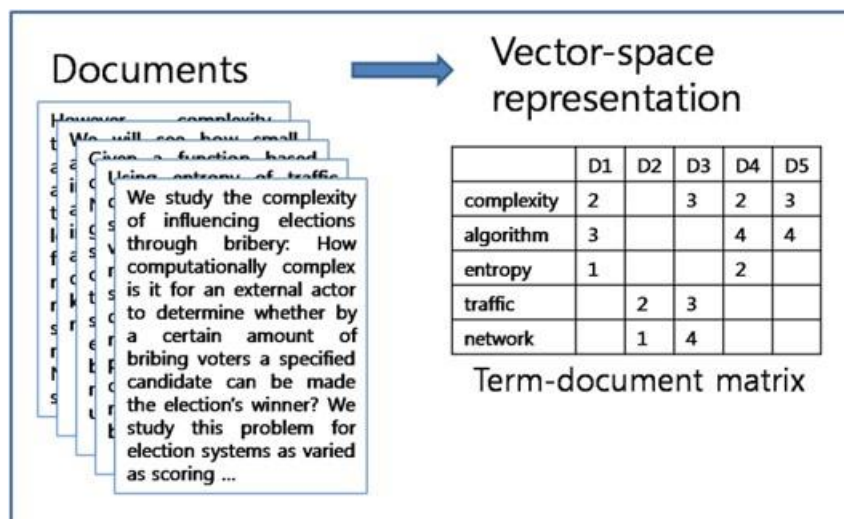
$$TF - IDF(t, d) = TF(t, d) * IDF(t)$$

## Term Frequency-Inverse Document Frequency(TF-IDF)

**TD-IDF** basically tells importance of the word in the corpus or dataset

- It is the combination of Term frequency and **Inverse Document Frequency** .
- Inverse Document frequency is another concept which is used for finding out importance of the word. It is based on the fact that less frequent words are more informative and important.

$IDF(t) = \log_e(\text{Total number of documents} / \text{Number of documents with term } t \text{ in it})$



1.4 vectorization

## 4.5 CLASIFICATION ALGORITHMS

Passive-aggressive classification is one of the available incremental learning algorithms and it is very simple to implement

Passive: If the prediction is correct, keep the model and do not make any changes. i.e., the data in the example is not enough to cause any changes in the model.

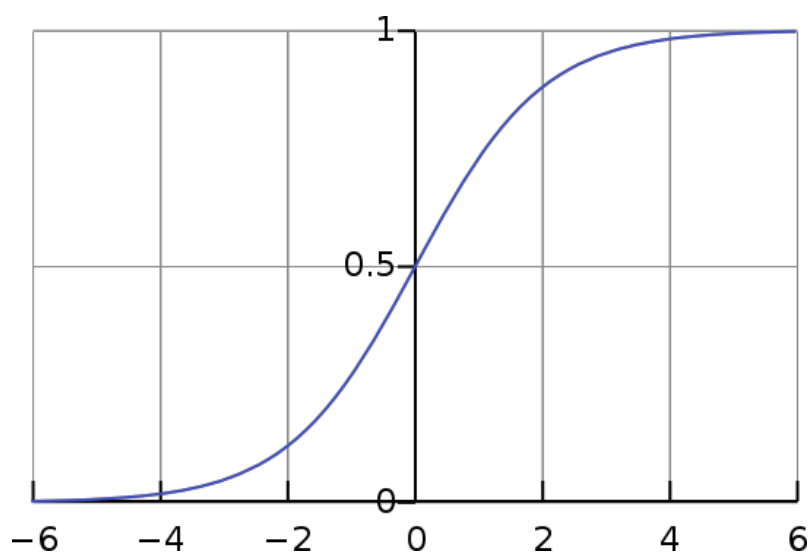
- Aggressive: If the prediction is incorrect, make changes to the model. i.e., some change to the model may correct it.

## 4.6 Logistic Regression

- Logistic Regression is a Machine Learning algorithm which is used for the classification problems, it is a predictive analysis algorithm and based on the concept of probability.
- logistic function  $1 / (1 + \exp(-x))$  normalizes everything to be between 0 and 1. Then we can interpret the number you get as a probability.

### REQUIREMENTS:

- Import Logistic regression from sklearn.linear\_model
- ```
>>> from sklearn.linear_model import  
LogisticRegression
```



1.5 logistic regression

5.

MODULE IMPLEMENTATION

5.1 Static Search Implementation-

In static part, we have trained and used 3 out of 4 algorithms for classification. They are Naïve Bayes, Random Forest and Logistic Regression

Step 1: In first step, we have extracted features from the already pre-processed dataset. These features are; Bag-of-words, Tf-Idf Features and N-grams.

Step 2: Here, we have built all the classifiers for predicting the fake news detection. The extracted features are fed into different classifiers. We have used Naive-bayes, Logistic Regression, and Random forest classifiers from sklearn. Each of the extracted features was used in all of the classifiers.

Step 3: Once fitting the model, we compared the f1 score and checked the confusion matrix.

Step 4: After fitting all the classifiers, 2 best performing models were selected as candidate models for fake news classification.

Step 5: We have performed parameter tuning by implementing GridSearchCV methods on these candidate models and chosen best performing parameters for these classifier.

Step 6: Finally selected model was used for fake news detection with the probability of truth.

Step 7: Our finally selected and best performing classifier was Logistic Regression which was then saved on disk. It will be used to classify the fake news

5.2 Dynamic Search Implementation

Our dynamic implementation contains 3 search fields which are-

- 1) Search by article content.
- 2) Search using key terms.
- 3) Search for website in database.

In the first search field we have used Natural Language

Processing for the first search field to come up with a proper solution for the problem, and hence we have attempted to create a model

which can classify fake news according to the terms used in the newspaper articles. Our application uses NLP techniques like

CountVectorization and TF-IDF Vectorization before passing it through a Passive Aggressive Classifier to output the authenticity as a percentage probability of an article.

The second search field of the site asks for specific keywords to be searched on the net upon which it provides a suitable output for the

percentage probability of that term actually being present in an article or a similar article with those keyword references in it.

The third search field of the site accepts a specific website domain name upon which the implementation looks for the site in our true sites

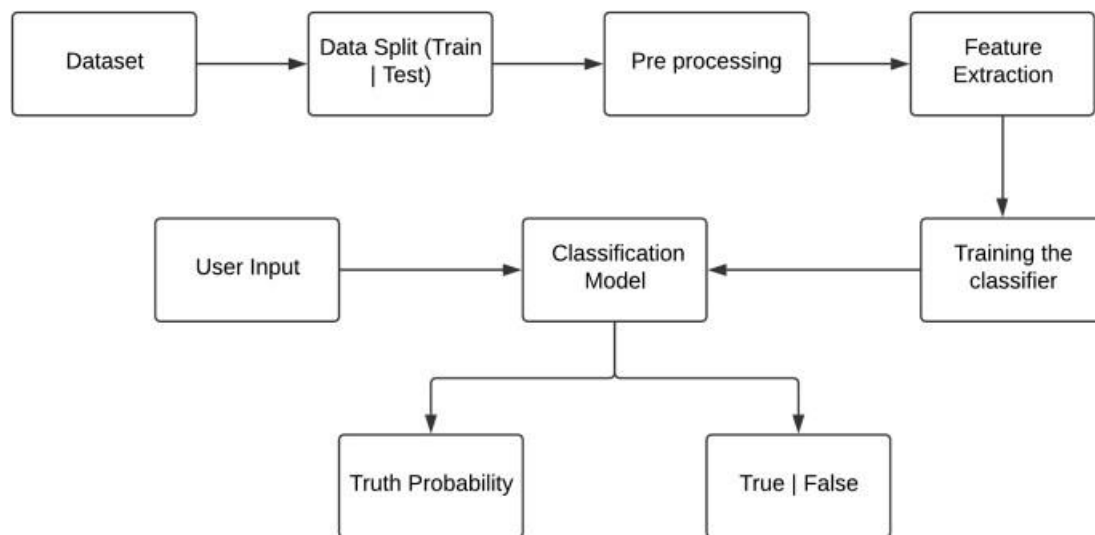
database or the blacklisted sites database. The true sites database holds the domain names which regularly provide proper and authentic

news and vice versa. If the site isn't found in either of the databases then the implementation

The problem can be broken down into 3 statements-

- 1) Use NLP to check the authenticity of a news article.
 - 2) If the user has a query about the authenticity of a search query then we he/she can directly search on our platform and using our custom algorithm we output a confidence score.
 - 3) Check the authenticity of a news source.
- These sections have been produced as search fields to take inputs in 3 different forms in our implementation of the problem statement.

5.3 SYSTEM ARCHITECTURE



1.6 system architecture

6.RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS

6.1 RESULT AND DISCUSSION

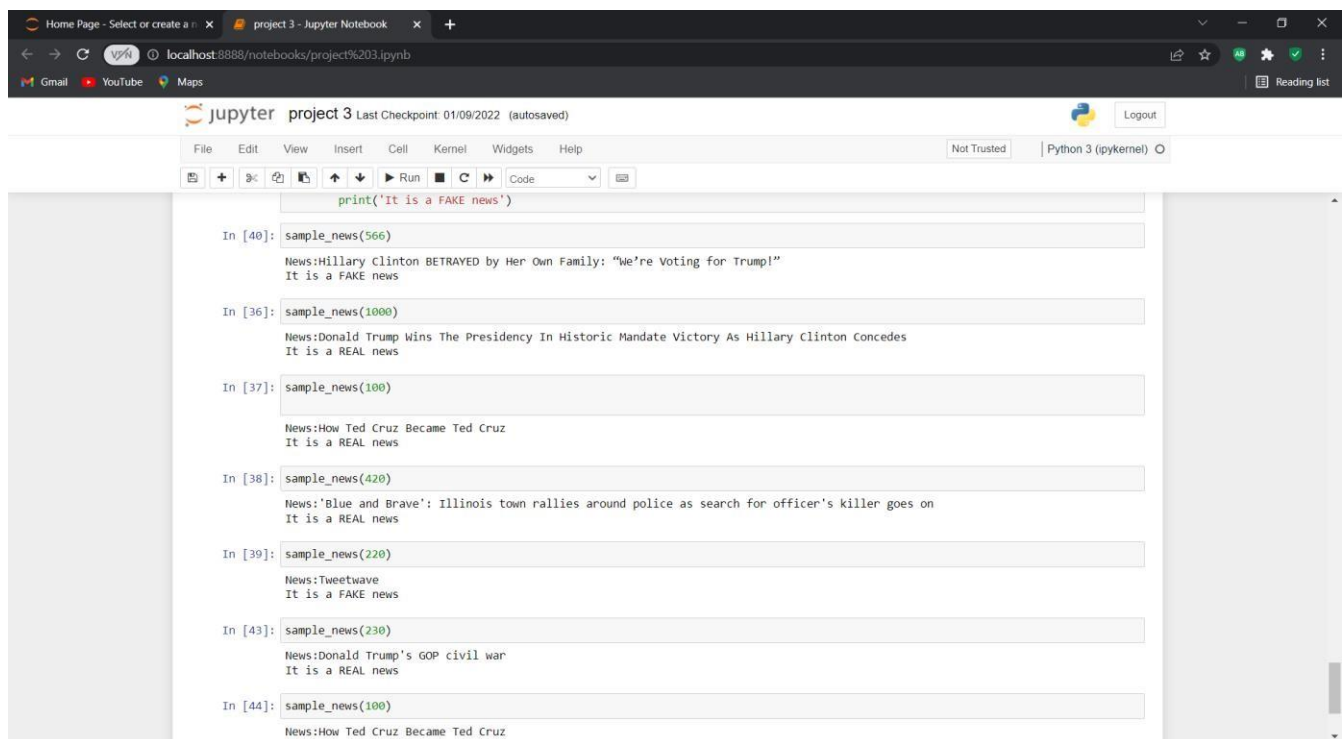
Implementation was done using the above algorithms with Vector features- Count Vectors and Tf-Idf vectors at Word level and Ngram level. Accuracy was noted for all models. We used K-fold cross validation technique to improve the effectiveness of the models.

A. Dataset split using K-fold cross validation

This cross-validation technique was used for splitting the dataset randomly into k-folds. (k-1) folds were used for building the model while kth fold was used to check the effectiveness of the model. This was repeated until each of the k-folds served as the test set. I used 3-fold cross validation for this experiment where 67% of the data is used for training the model and remaining 33% for testing.

B. Confusion Matrices for Static System

After applying various extracted features (Bag-of-words, Tf-Idf, N-grams) on three different classifiers (Naïve bayes, Logistic Regression and Random Forest), their confusion matrix showing actual set and predicted sets are mentioned below:



```
print('It is a FAKE news')
```

```
In [40]: sample_news(566)
News:Hillary Clinton BETRAYED by Her Own Family: "We're Voting for Trump!"
It is a FAKE news
```

```
In [36]: sample_news(1000)
News:Donald Trump Wins The Presidency In Historic Mandate Victory As Hillary Clinton Concedes
It is a REAL news
```

```
In [37]: sample_news(100)
News:How Ted Cruz Became Ted Cruz
It is a REAL news
```

```
In [38]: sample_news(420)
News:'Blue and Brave': Illinois town rallies around police as search for officer's killer goes on
It is a REAL news
```

```
In [39]: sample_news(220)
News:Tweetwave
It is a FAKE news
```

```
In [43]: sample_news(230)
News:Donald Trump's GOP civil war
It is a REAL news
```

```
In [44]: sample_news(100)
News:How Ted Cruz Became Ted Cruz
```

6.2 PERFORMANCE ANALYSIS

ACCURACY

Whenever the accuracy metric is used, we aim to learn the closeness of a measured value to a known value. It's therefore typically used in instances where the output variable is categorical or discrete — Namely a classification task.

PRECISION

In instances where we are concerned with how exact the model's predictions are we would use Precision. The precision metric would inform us of the number of labels that are actually labeled as positive in correspondence to the instances that the classifier labeled as positive.

7. CONCLUSION

In the 21st century, the majority of the tasks are done online. Newspapers that were earlier preferred as hard-copies are now being substituted by applications like Facebook, Twitter, and news articles to be read online. Whatsapp's forwards are also a major source. The growing problem of fake news only makes things more complicated and tries to change or hamper the opinion and attitude of people towards use of digital technology. When a person is deceived by the real news two possible things happen- People start believing that their perceptions about a particular topic are true as assumed. Thus, in order to curb the phenomenon, we have developed our Fake news Detection system that takes input from the user and classify it to be true or fake. To implement this, various NLP and Machine Learning Techniques have to be used. The model is trained using an appropriate dataset and performance evaluation is also done using various performance measures. The best model, i.e. the model with highest accuracy is used to classify the news headlines or articles. As evident above for static search, our best model came out to be Logistic Regression with an accuracy of 65%. Hence we then used grid search parameter optimization to increase the performance of logistic regression which then gave us the accuracy of 75%. Hence we can say that if a user feed a particular news article or its headline in our model, there are 75% chances that it will be classified to its true nature. The user can check the news article or keywords online; he can also check the authenticity of the website. The accuracy for dynamic system is 93% and it increases with every iteration. We intend to build our own dataset which will be kept up to date according to the latest news. All the live news and latest data will be kept in a database using Web Crawler and online database.

7.1 REFERENCE

- [1] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu, "Fake News Detection on Social Media: A Data Mining Perspective" arXiv:1708.01967v3 [cs.SI], 3 Sep 2017
- [3] M. Granik and V. Mesyura, "Fake news detection using naive Bayes classifier," 2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON), Kiev, 2017, pp. 900-903.
 - [4] Fake news websites. (n.d.) Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Fake_news_website. Accessed Feb. 6, 2017
 - [5] Cade Metz. (2016, Dec. 16). The bittersweet sweepstakes to build an AI that destroys fake news.
 - [6] Conroy, N., Rubin, V. and Chen, Y. (2015). "Automatic deception detection: Methods for finding fake news" at Proceedings of the Association for Information Science and Technology, 52(1), pp.1-4.

7.2 APPENDIX A

SNAP SHOT

The screenshot shows the Visual Studio Code interface with a Jupyter Notebook open. The Explorer sidebar on the left shows 'NO FOLDER OPENED'. The main editor area contains the following code cells:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

```
df = pd.read_csv('fake_or_real_news.csv')
```

```
df.head()
```

	Unnamed: 0	title	text	label
0	8476	You Can Smell Hillary's Fear	Daniel Greenfield, a Shillman Journalism Fello...	FAKE
1	10294	Watch The Exact Moment Paul Ryan Committed Pol...	Google Pinterest Digg LinkedIn Reddit Stumbleu...	FAKE
2	3608	Kerry to go to Paris in gesture of sympathy	U.S. Secretary of State John F. Kerry said Mon...	REAL
3	10142	Bernie supporters on Twitter erupt in anger ag...	— Kaydee King (@KaydeeKing) November 9, 2016 T...	FAKE
4	875	The Battle of New York: Why This Primary Matters	It's primary day in New York and front-runners...	REAL

```
df.shape
```

```
(6335, 4)
```

```
df.describe().T
```

The screenshot shows the Visual Studio Code interface with a Jupyter Notebook open. The Explorer sidebar on the left shows 'NO FOLDER OPENED'. The main editor area contains the following code cells:

```
df.isnull().sum()
```

```
df.drop(df.filter(regex="Unnamed"),axis=1, inplace=True)
```

```
df
```

	title	text	label
0	You Can Smell Hillary's Fear	Daniel Greenfield, a Shillman Journalism Fello...	FAKE
1	Watch The Exact Moment Paul Ryan Committed Pol...	Google Pinterest Digg LinkedIn Reddit Stumbleu...	FAKE
2	Kerry to go to Paris in gesture of sympathy	U.S. Secretary of State John F. Kerry said Mon...	REAL
3	Bernie supporters on Twitter erupt in anger ag...	— Kaydee King (@KaydeeKing) November 9, 2016 T...	FAKE
4	The Battle of New York: Why This Primary Matters	It's primary day in New York and front-runners...	REAL
...	...	...	...
6330	State Department says it can't find emails fro...	The State Department told the Republican Natio...	REAL
6331	The 'P' in PBS Should Stand for 'Plutocratic' ...	The 'P' in PBS Should Stand for 'Plutocratic' ...	FAKE
6332	Anti-Trump Protesters Are Tools of the Oligarc...	Anti-Trump Protesters Are Tools of the Oligarc...	FAKE
6333	In Ethiopia, Obama seeks progress on peace, se...	ADDIS ABABA, Ethiopia — President Obama convene...	REAL
6334	Jeb Bush Is Suddenly Attacking Trump. Here's W...	Jeb Bush Is Suddenly Attacking Trump. Here's W...	REAL

6335 rows x 3 columns

Visual Studio Code interface showing a Jupyter Notebook for project 3.ipynb. The Explorer sidebar on the left indicates "NO FOLDER OPENED" and provides options like "Open Folder", "Clone Repository", and "Create Java Project".

The Jupyter Notebook contains the following code cells:

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
df['label'] = le.fit_transform(df['label'])
```

The output of the first cell is a DataFrame with 6335 rows and 3 columns:

	title	text	label
0	You Can Smell Hillary's Fear	Daniel Greenfield, a Shillman Journalism Fello...	0
1	Watch The Exact Moment Paul Ryan Committed Pol...	Google Pinterest Digg LinkedIn Reddit Stumble...	0
2	Kerry to go to Paris in gesture of sympathy	U.S. Secretary of State John F. Kerry said Mon...	1
3	Bernie supporters on Twitter erupt in anger ag...	— Kaydee King (@KaydeeKing) November 9, 2016 T...	0
4	The Battle of New York Why This Primary Matters	It's primary day in New York and front-runners...	1
...	...	...	...
6330	State Department says it can't find emails fro...	The State Department told the Republican Natio...	1
6331	The 'P' in PBS Should Stand for 'Plutocratic' ...	The 'P' in PBS Should Stand for 'Plutocratic' ...	0
6332	Anti-Trump Protesters Are Tools of the Oligarc...	Anti-Trump Protesters Are Tools of the Oligarc...	0
6333	In Ethiopia, Obama seeks progress on peace, se...	ADDIS ABABA, Ethiopia —President Obama convene...	1
6334	Jeb Bush Is Suddenly Attacking Trump. Here's W...	Jeb Bush Is Suddenly Attacking Trump. Here's W...	1

The second cell contains the following code:

```
import nltk
import re
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
```

Visual Studio Code interface showing the continuation of the Jupyter Notebook for project 3.ipynb. The Explorer sidebar remains the same.

The Jupyter Notebook contains the following code cells:

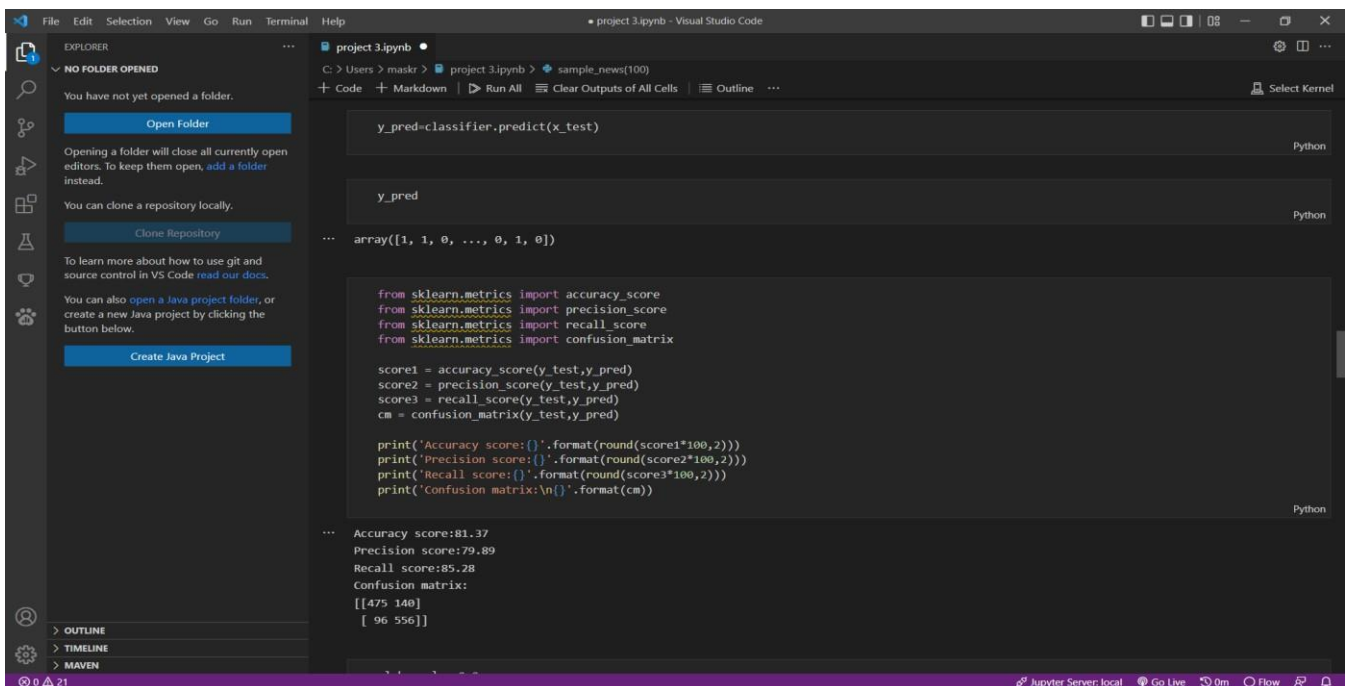
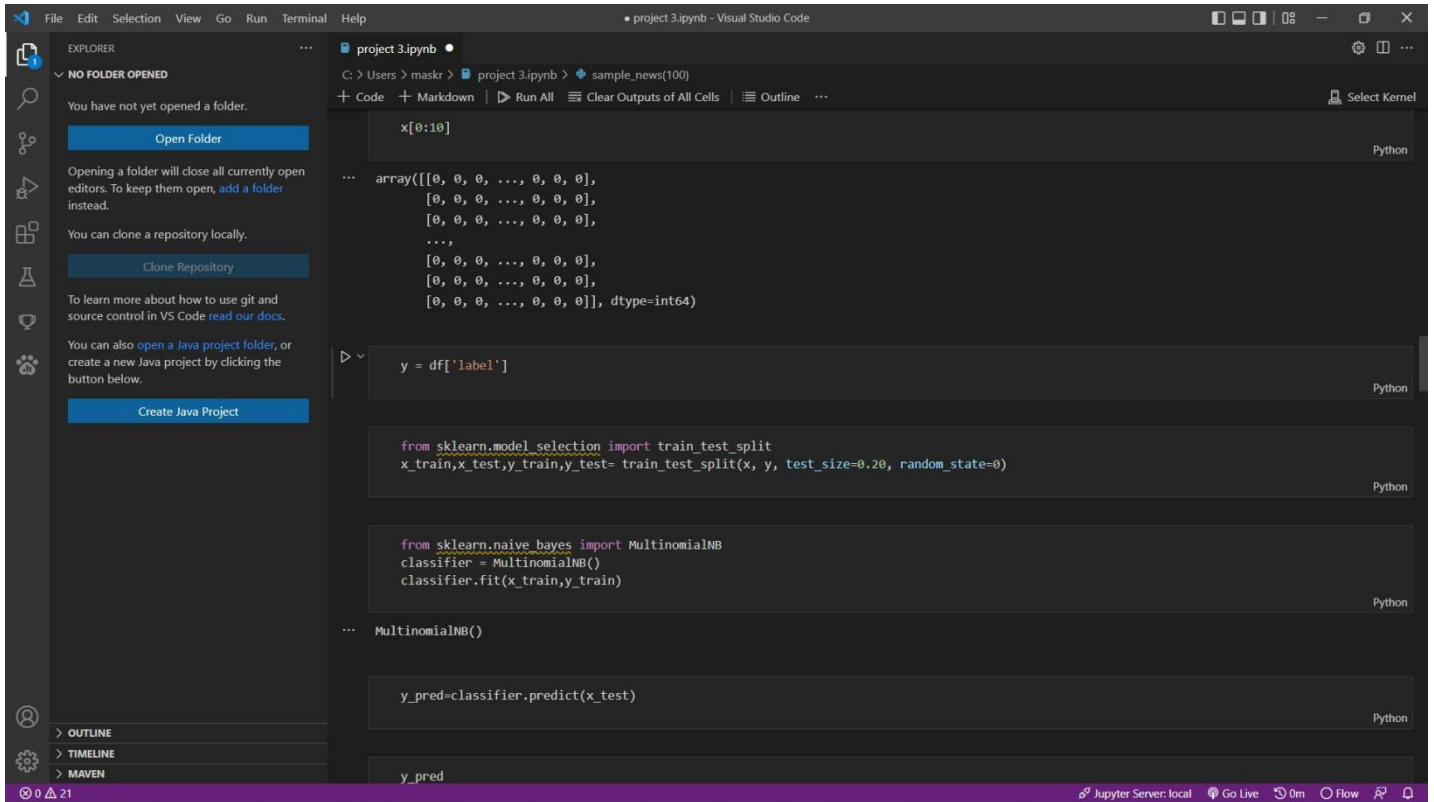
```
corpus = []
for i in range(0, df.shape[0]):
    title = re.sub('[^a-zA-Z]', ' ', str(df.title[i]))
    title = title.lower()
    words = title.split()
    words = [word for word in words if word not in set(stopwords.words('english'))]
    ps = PorterStemmer()
    words = [ps.stem(word) for word in words]
    title = ' '.join(words)
    corpus.append(title)
```

The output of the second cell is a list of stemmed titles:

```
corpus[0:10]
['smell hillari fear',
'watch exact moment paul ryan commit politic suicid trump ralli video',
'kerri go pari gestur sympathi',
'berni support twitter erupt anger dnc tri warn',
'battl new york primari matter',
'tehran usa',
'girl horrifi watch boyfriend left facetim',
'britain schindler die',
'fact check trump clinton command chief forum',
'iran reportedli make new push uranium concess nuclear talk']
```

The third cell contains the following code:

```
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer()
x = cv.fit_transform(corpus).toarray()
```



```

alpha_val = 0.0
best_accuracy = 0.0
for i in np.arange(0.1,1.1,0.1):
    temp_classifier = MultinomialNB(alpha=i)
    temp_classifier.fit(x_train,y_train)
    temp_y_pred=temp_classifier.predict(x_test)
    score = accuracy_score(y_test,temp_y_pred)
    print('Accuracy score for {} is {}'.format(round(i,1),round(score*100,2)))

    if score > best_accuracy:
        best_accuracy = score
        alpha_val = i

print('The best accuracy is {}% with alpha value as {}'.format(round(best_accuracy*100, 2), round(alpha_val,1)))

```

```

...
Accuracy score for 0.1 is 80.9
Accuracy score for 0.2 is 81.53
Accuracy score for 0.3 is 81.77
Accuracy score for 0.4 is 81.53
Accuracy score for 0.5 is 81.29
Accuracy score for 0.6 is 81.37
Accuracy score for 0.7 is 81.37
Accuracy score for 0.8 is 81.53
Accuracy score for 0.9 is 81.37
Accuracy score for 1.0 is 81.37
The best accuracy is 81.77% with alpha value as 0.3

```

```

classifier = MultinomialNB(alpha=0.3)
classifier.fit(x_train,y_train)
y_pred=classifier.predict(x_test)

```

```

score1 = accuracy_score(y_test,y_pred)
score2 = precision_score(y_test,y_pred)
score3 = recall_score(y_test,y_pred)
cm = confusion_matrix(y_test,y_pred)

print('Accuracy score:{}'.format(round(score1*100,2)))
print('Precision score:{}'.format(round(score2*100,2)))
print('Recall score:{}'.format(round(score3*100,2)))
print('Confusion matrix:\n{}'.format(cm))

```

```

...
Accuracy score:81.77
Precision score:80.91
Recall score:84.51
Confusion matrix:
[[485 130]
 [101 551]]

```

```

from sklearn.linear_model import LogisticRegression
lr = LogisticRegression(random_state=0)
lr.fit(x_train,y_train)

...
LogisticRegression(random_state=0)

y_pred = lr.predict(x_test)

y_pred

...
array([[1, 1, 0, ..., 0, 1, 0]])

```

Visual Studio Code interface showing a Jupyter Notebook cell. The Explorer sidebar on the left indicates 'NO FOLDER OPENED'. The main editor displays a Python code cell with the following content:

```

from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
from sklearn.metrics import confusion_matrix

score1 = accuracy_score(y_test,y_pred)
score2 = precision_score(y_test,y_pred)
score3 = recall_score(y_test,y_pred)
cm = confusion_matrix(y_test,y_pred)

print('Accuracy score:{}'.format(round(score1*100,2)))
print('Precision score:{}'.format(round(score2*100,2)))
print('Recall score:{}'.format(round(score3*100,2)))
print('Confusion matrix:\n{}'.format(cm))

```

The output of the cell is displayed below the code:

```

Accuracy score:82.24
Precision score:83.0
Recall score:82.36
Confusion matrix:
[[505 110]
 [115 537]]

```

The bottom status bar shows 'Python 3.8.8 64-bit' and various extension icons.

Visual Studio Code interface showing a Jupyter Notebook cell. The Explorer sidebar on the left indicates 'NO FOLDER OPENED'. The main editor displays a Python code cell with the following content:

```

from sklearn.linear_model import LogisticRegression
lr = LogisticRegression(C=0.7,random_state=0)
lr.fit(x_train,y_train)

LogisticRegression(C=0.7, random_state=0)

y_pred = lr.predict(x_test)

score1 = accuracy_score(y_test,y_pred)
score2 = precision_score(y_test,y_pred)
score3 = recall_score(y_test,y_pred)
cm = confusion_matrix(y_test,y_pred)

print('Accuracy score:{}'.format(round(score1*100,2)))
print('Precision score:{}'.format(round(score2*100,2)))
print('Recall score:{}'.format(round(score3*100,2)))
print('Confusion matrix:\n{}'.format(cm))

```

The output of the cell is displayed below the code:

```

Accuracy score:82.4
Precision score:83.46
Recall score:82.06
Confusion matrix:
[[509 106]
 [117 535]]

```

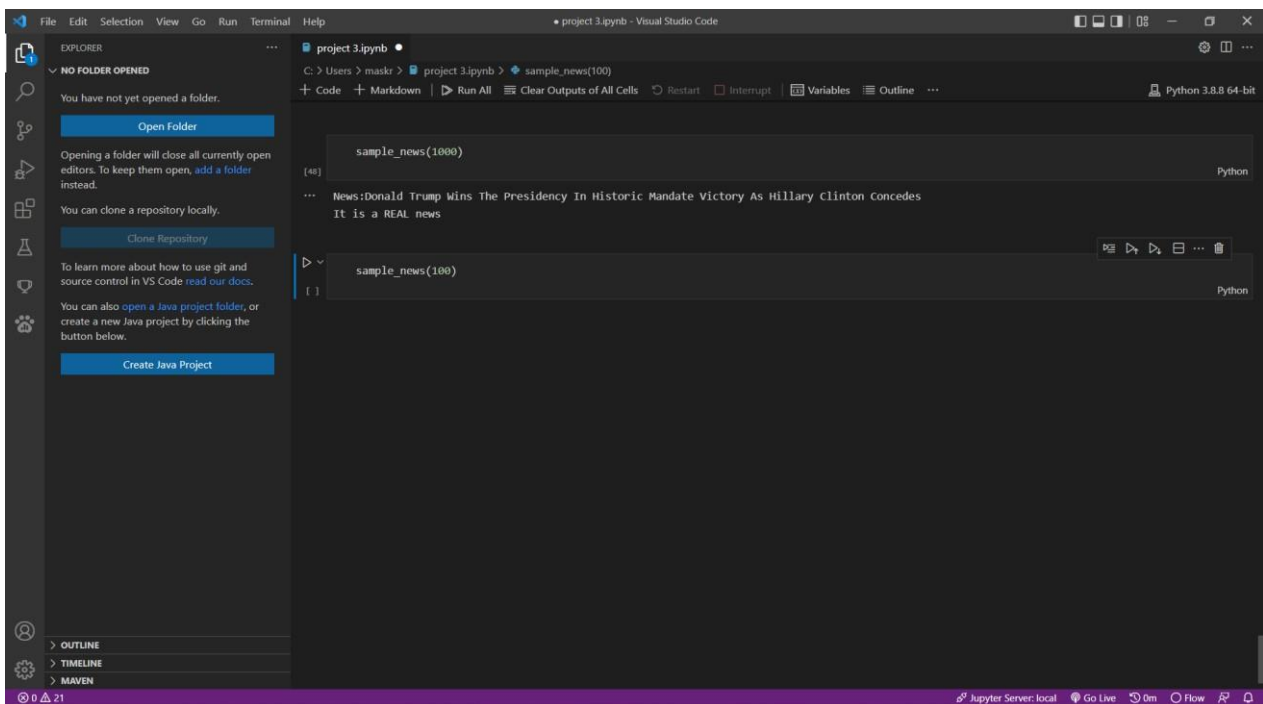
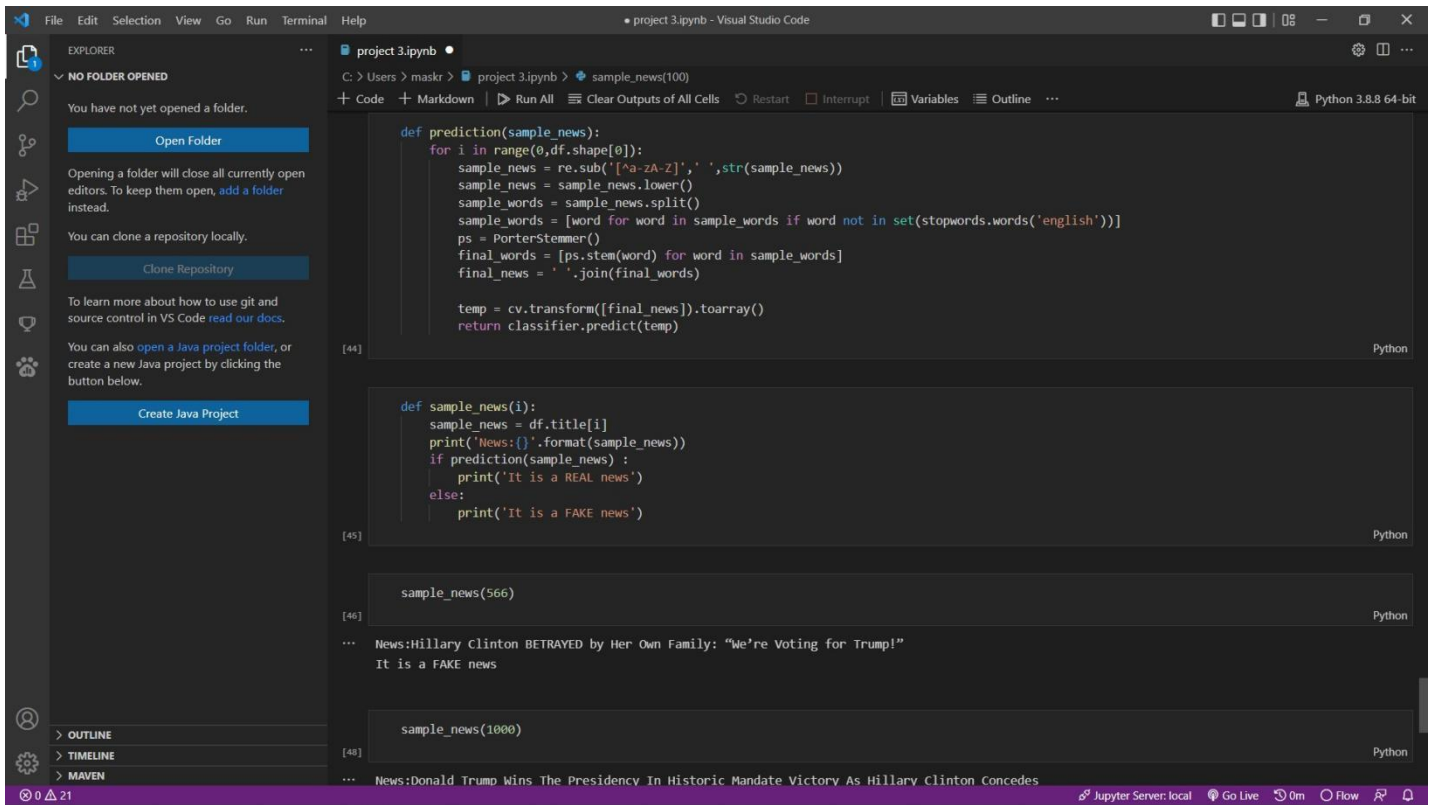
Below the output, a function definition is visible:

```

def prediction(sample_news):
    for i in range(0,df.shape[0]):
        sample_news = re.sub('[^a-zA-Z]', ' ',str(sample_news))

```

The bottom status bar shows 'Python 3.8.8 64-bit' and various extension icons.



APPENDIX B

Source code

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
import re
import string

df_fake = pd.read_csv("Fake.csv")
df_true = pd.read_csv("True.csv")

df_fake.head(5)

df_fake["class"] = 0
df_true["class"] = 1

df_fake.shape, df_true.shape
df_fake_manual_testing = df_fake.tail(10)
for i in range(23480,23470,-1):
    df_fake.drop([i], axis = 0, inplace = True)
df_true_manual_testing = df_true.tail(10)
for i in range(21416,21406,-1):
    df_true.drop([i], axis = 0, inplace = True)

df_fake.shape, df_true.shape

df_fake_manual_testing["class"] = 0
df_true_manual_testing["class"] = 1

df_fake_manual_testing.head(10)

df_true_manual_testing.head(10)

df_manual_testing = pd.concat([df_fake_manual_testing,df_true_manual_testing], axis = 0)
df_manual_testing.to_csv("manual_testing.csv")
```

```

df_marge = pd.concat([df_fake, df_true], axis =0 )
df_marge.head(10)

df_marge.columns

df = df_marge.drop(["title", "subject","date"], axis = 1)

df.isnull().sum()

df = df.sample(frac = 1)

df.head()

df.reset_index(inplace = True)
df.drop(["index"], axis = 1, inplace = True)
df.reset_index(inplace = True)
df.drop(["index"], axis = 1, inplace = True)

df.columns

df.head()

def wordopt(text):
    text = text.lower()
    text = re.sub('[\.\*\?\\]', '', text)
    text = re.sub("\\W", " ",text)
    text = re.sub('https?://\S+|www\.\S+', '', text)
    text = re.sub('<.*?>+', '', text)
    text = re.sub('[%s]' % re.escape(string.punctuation), '', text)
    text = re.sub('\n', '', text)
    text = re.sub('\w*\d\w*', '', text)
    return text

df["text"] = df["text"].apply(wordopt)

x = df["text"]
y = df["class"]

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.25)

from sklearn.feature_extraction.text import TfidfVectorizer

vectorization = TfidfVectorizer()
xv_train = vectorization.fit_transform(x_train)
xv_test = vectorization.transform(x_test)

```

```

from sklearn.linear_model import LogisticRegression

LR = LogisticRegression()
LR.fit(xv_train,y_train)

pred_lr=LR.predict(xv_test)

LR.score(xv_test, y_test)

print(classification_report(y_test, pred_lr))

from sklearn.tree import DecisionTreeClassifier

DT = DecisionTreeClassifier()
DT.fit(xv_train, y_train)

pred_dt = DT.predict(xv_test)

DT.score(xv_test, y_test)

print(classification_report(y_test, pred_dt))

from sklearn.ensemble import GradientBoostingClassifier

GBC = GradientBoostingClassifier(random_state=0)
GBC.fit(xv_train, y_train)

pred_gbc = GBC.predict(xv_test)

GBC.score(xv_test, y_test)

print(classification_report(y_test, pred_gbc))

from sklearn.ensemble import RandomForestClassifier

RFC = RandomForestClassifier(random_state=0)
RFC.fit(xv_train, y_train)

pred_rfc = RFC.predict(xv_test)

RFC.score(xv_test, y_test)

print(classification_report(y_test, pred_rfc))

```

```

def manual_testing(news):
    testing_news = {"text":[news]}
    new_def_test = pd.DataFrame(testing_news)
    new_def_test["text"] = new_def_test["text"].apply(wordopt)
    new_x_test = new_def_test["text"]
    new_xv_test = vectorization.transform(new_x_test)
    pred_LR = LR.predict(new_xv_test)
    pred_DT = DT.predict(new_xv_test)
    pred_GBC = GBC.predict(new_xv_test)
    pred_RFC = RFC.predict(new_xv_test)

    return print("\n\nLR Prediction: {} \nDT Prediction: {} \nGBC Prediction: {} \nRFC Prediction:
    {}".format(output_label(pred_LR[0]),
    e(pred_DT[0]),
    e(pred_GBC[0]),
    e(pred_RFC[0])))

```

output_label

output_label

output_label