

COVID FAKE NEWS DETECTION

Submitted in partial fulfillment of the requirements
of the degree of

B.E. COMPUTER ENGINEERING

By

Aishwarya John	35	182044
Anashwara Kurien	42	182051
Sherwin Lobo	46	182055

Guide:

Mrs. Vincy Joseph

Designation: Assistant Professor

Department of Computer Science



**St. Francis Institute of Technology
(Engineering College)
University of Mumbai
2021-2022**

CERTIFICATE

This is to certify that the project entitled “Covid Fake News Detection” is a bonafide work of “ Aishwarya John(35), Anashwara Kurien(42) and Sherwin Lobo(46) ” submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of B.E. in Computer Engineering.

**Mrs. Vincy Joseph
Guide**

**Dr. Kavita Sonawane
Head of Department**

Project Report Approval for B.E.

This project report entitled (Covid Fake News Detection) by (Aishwarya John(35), Anashwara Kurien(42) and Sherwin Lobo(46)) is approved for the degree of B.E. in Computer Engineering.

Examiners

1. _____

2. _____

Date: 21-04-22

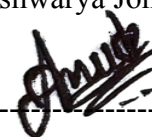
Place: Mumbai

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.



Aishwarya John 35



Anashwara Kurien 42



Sherwin Lobo 46

Date: 21-04-22

Abstract

The advent of the World Wide Web and the rapid adoption of social media platforms paved the way for information dissemination that has never been witnessed in human history before. With the current usage of social media platforms, consumers are creating and sharing more information than ever before, some of which are misleading with no relevance to reality. Automated classification of a text article as misinformation or disinformation is a challenging task. In this work, we propose to use natural language processing and machine learning ensemble approach for automated classification of news articles. Our study explores different textual properties that can be used to distinguish fake contents from real. By using those properties, we train a combination of different machine learning algorithms using various ensemble methods and evaluate their performance on real world datasets. Fake news and rumors are rampant on social media. Believing in rumors can cause significant harm. This is further exacerbated at the time of a pandemic. To tackle this, we curate and release a manually annotated dataset of many social media posts and articles of real and fake news on COVID-19. We benchmark the annotated dataset with three machine learning baselines - Logistic Regression, Ensemble and Support Vector Machine (SVM).

The need to stop the spreading of fake news, it's paramount and this paper proposes recognizing truthful information from false information during the pandemic COVID-19 through a guide learning method. This guide implies a model for distinguishing false messages in the online environment, such as Machine Learning algorithms, which can have an accuracy of over 90%.

Contents

Chapter		Contents	Page No.
1		INTRODUCTION	7-8
	1.1	Description	7
	1.2	Problem Formulation	7
	1.3	Motivation	7-8
	1.4	Proposed Solution	8
	1.5	Scope of the project	8
2		REVIEW OF LITERATURE	9
3		SYSTEM ANALYSIS	10-12
	3.1	Functional Requirements	10
	3.2	Non Functional Requirements	10
	3.3	Specific Requirements	10-11
	3.4	Use-Case Diagrams and description	11-12
4		ANALYSIS MODELING	13-14
	4.1	Activity Diagrams	13
	4.2	Functional Modeling	14
5		DESIGN	15-16
	5.1	Architectural Design	15
	5.2	User Interface Design	16
6		IMPLEMENTATION	17-23
	6.1	Algorithms / Methods Used	17-18
	6.2	Working of the project	18-23
7		CONCLUSIONS	24

References

Acknowledgements

List of Figures

Fig. No.	Figure Caption	Page No.
1	Use Case Diagram	12
2	Activity Diagram	13
3	DFD Diagram	14
4	Architectural Diagram	15
5	User Interface	16
6	Backend Screenshots	20-22
7	Frontend Screenshots	23

List of Abbreviations

Sr. No.	Abbreviation	Expanded form
1.	SVM	Support Vector Machine
2.	NLP	Natural Language Processing
3.	UML	Unified Modeling Language
4.	DFD	Data-flow Diagram
5.	BS4	Beautiful soup4
6.	TFIDF	Term Frequency Inverse Document Frequency

Chapter 1

Introduction

1.1 Description

The proposed work suggests a novel and amalgamated method combining some known and well researched methods merging the advantages of AI through the simplistic algorithm, Logistic Regression, SVM and Ensemble which is a simple, easy to understand, quick yet efficient algorithm. Along with that, use of NLP word morphology is done to help preprocess and remove error causing words to maximize classification of news as fake or genuine. Since a major drawback, for ML and AI algorithms, when it comes to processing information such as covid news articles would be the checking the accuracy of facts or news that have surfaced recently. We have used web scraping to scrape news from various news webpages.

Since the suggested algorithm for fake news detection only works on a predefined dataset, the training module might not work as efficiently for the same. It helps the user with a checking mechanism, a system that could inform him about the various characteristics of the input article such as accuracy, etc.

1.2 Problem Formulation

The proposed system boosts the accuracy and bolsters the results through the integration of a web scraping module which is capable of scraping through various news websites, internationally recognized as accurate, for latest news articles in the regional language Malayalam and classifies the news into real and fake news.

1.3 Motivation

There has been a rapid increase in the spread of fake news in the last decade, most prominently observed in the 2016 US elections. Such proliferation of sharing articles online that do not conform to facts has led to many problems not just limited to politics but covering various other domains such as sports, health, and also science.

One recent case is the spread of novel corona virus, where fake reports spread over the Internet about the origin, nature, and behavior of the virus. The situation worsened as more people read

about the fake contents online. Identifying such news online is a daunting task. Therefore we are aiming at classifying such covid news on the internet as real and fake specifically on the regional language malayalam.

1.4 Proposed Solution

In this project a model is built that predicts if the news is fake or genuine news. Since this problem is a kind of text classification, Implementing a Ensemble classifier will be best as this is standard for text-based processing. The actual goal is in developing a model to test genuinity of news for the regional language Malayalam, which gives a higher accuracy for the different news sets in the dataset. We have also applied two more algorithms and found the accuracy of the dataset. With this technique we have classified the news into fake and true news and found out the accuracy of each algorithm applied.

1.5 Scope of the project

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

Chapter 2

Review of Literature

For detection of fake news, simple classification models are not enough to get accurate results. In paper [1], the authors have demonstrated the use of text-based preprocessing along with three classifiers i.e., Passive Aggressive, Naïve Bayes, and Support Vector Machine for fake news detection. Basic text preprocessing techniques like tokenization to tokenize words to base form, porter stemming and removal of stopwords to preprocess the data. Along with the classifiers other ML algos used were CountVectorizer and TfidfVectorizer to make a model based on count vectorization and TF-IDF. The trained model was tested on different datasets to achieve the highest accuracy of 93% with Passive Aggressive, 85% in naive bayes and 84% in SVM.

In paper [2], the authors have used machine learning as well as natural language processing techniques to detect fake news. First the headline was tokenized using the NLTK library. Using NLP, stopwords were removed and the headlines were preprocessed. Machine learning algorithm was applied and the dataset was split into training and testing data. The model was trained on three algorithms:- SVM, Logistic Regression and Naive Bayes Classification. The accuracy of Naive Bayes was the highest, giving a total accuracy of 0.83 followed by SVM having an accuracy of 0.81. The author has provided a comparison on the accuracy of all the three algorithms.

Chapter 3

System Analysis

3.1 Functional Requirements:

Dataset: The lack of manually labeled fake news datasets is certainly a bottleneck for advancing computationally intensive, text-based models that cover a wide array of topics. The dataset for the fake news challenge does not suit our purpose due to the fact that it contains the ground truth regarding the relationships between texts but not whether or not those texts are actually true or false statements. For our purpose, we need a set of news articles that is directly classified into categories of news types. We have web scrapped some of the true news from various web pages and true news dataset was taken from the web.

Libraries in python:

- Numpy
- Pandas
- Nltk
- ensemble
- DecisionTreeClassifier
- text, sequence
- model_selection, preprocessing, svm
- TfidfVectorizer, CountVectorizer
- text, sequence
- dump

3.2 Non-Functional Requirements:

These are basically the quality constraints that the system must satisfy according to the project contract. The priority or extent to which these factors are implemented varies from one project to other. They are also called non-behavioral requirements.

- The processing of each request should be done within 10 seconds.
- The site should load in 3 seconds when the number of simultaneous users are > 10000
- The system should provide better accuracy.
- User friendly.

3.3 Specific Requirements

The hardware environment consists of the following:

CPU	: Intel Pentium IV 600MHz or above
Mother Board	: Intel 810 or above
Hard disk space	: 20GB or more
Display	: Color Monitor
Memory	: 128 MB RAM
Other Devices	: Keyboard, mouse.

a) Server side

The web application will be hosted on a web server which is listening on the web standard port, port 80.

b) Client side

Monitor screen – the software shall display information to the user via the monitor screen

Mouse – the software shall interact with the movement of the mouse and the mouse buttons. The mouse shall activate areas for data input, command buttons and select options from menus.

Keyboard – the software shall interact with the keystrokes of the keyboard. The keyboard will input data into the active area of the database.

Software Requirements:

Development Tools: VS code, Google Colab, Anaconda IDE,Pip Package

Front End : Anvil

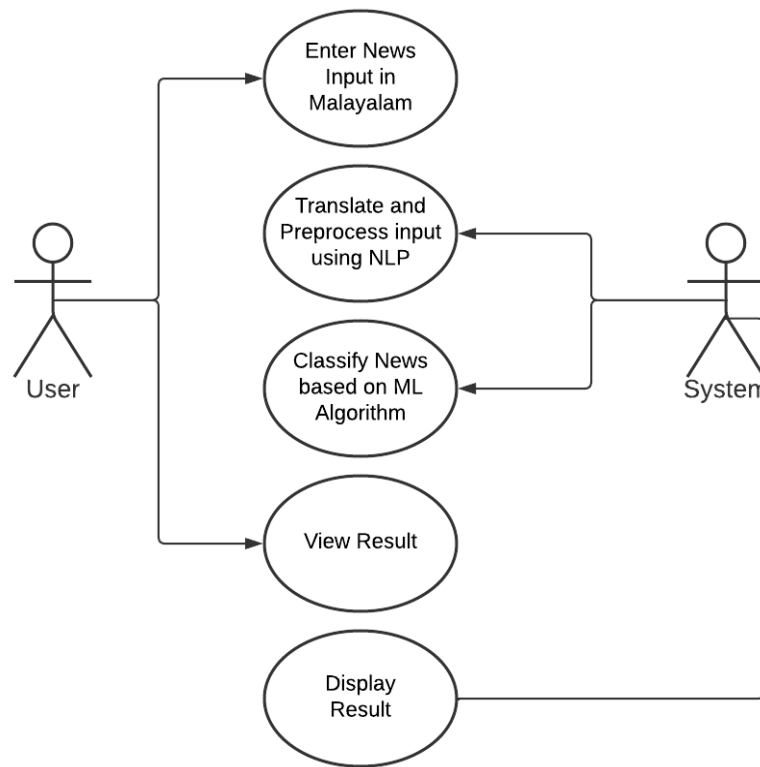
Back End : Python

Connection : Localhost

Operating System : Windows 10

The actual program that will perform the operations is written in Python.

3.4 Use-Case Diagrams and description:



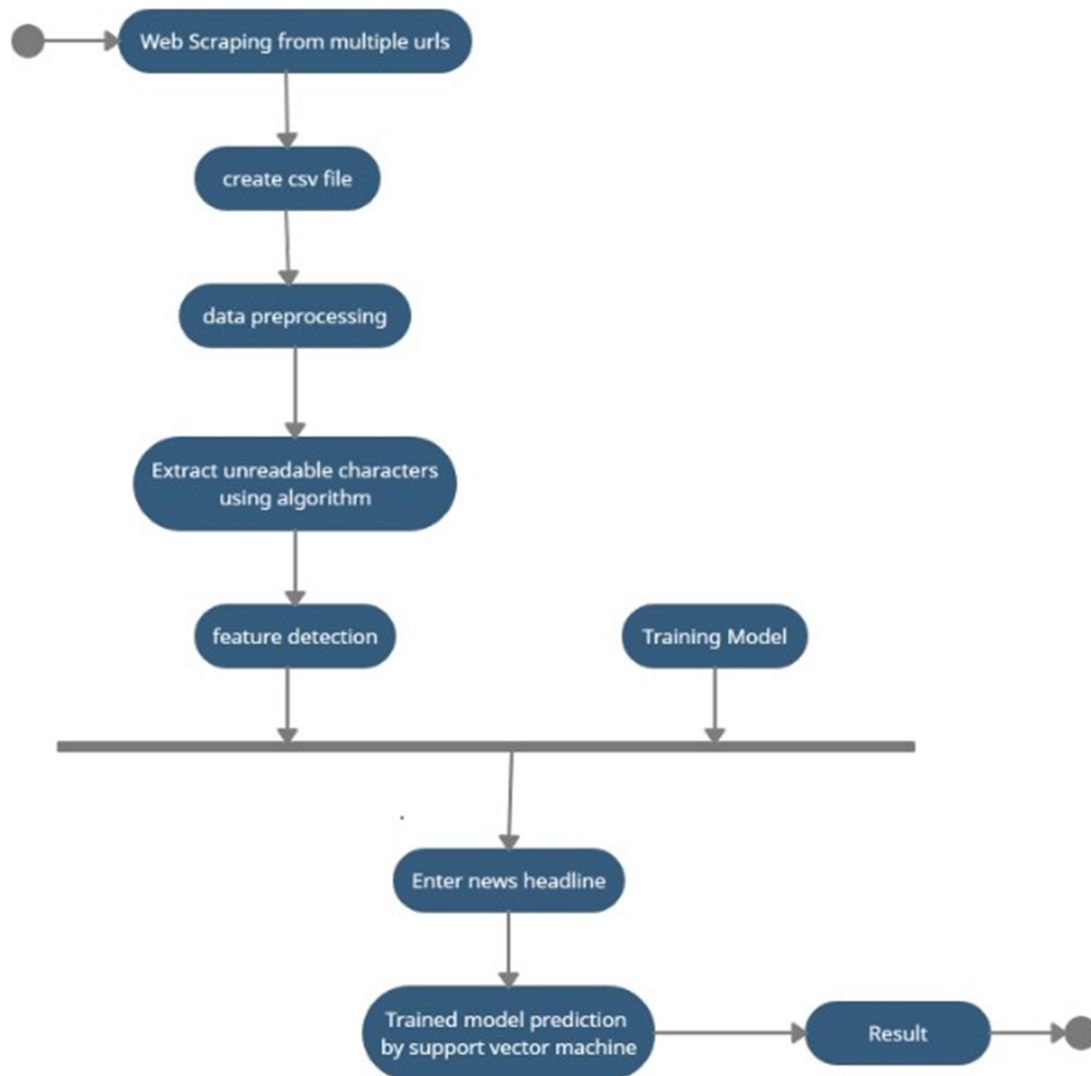
(Fig1: UML Diagram)

In this use case diagram the user has to enter the news from the website in malayalam. That input is then translated using google translate library to english. The translated sentence is thus given to the algorithm applied to the machine to give an output whether the news extracted from the web browser is fake or true news. It is the system that applies the valid algorithm and checks the accuracy of the news on the basis of training given to the model and displays the result of the news given as input. After the result is computed by the system it is shown to the user where the news is classified as true or fake news.

Chapter 4

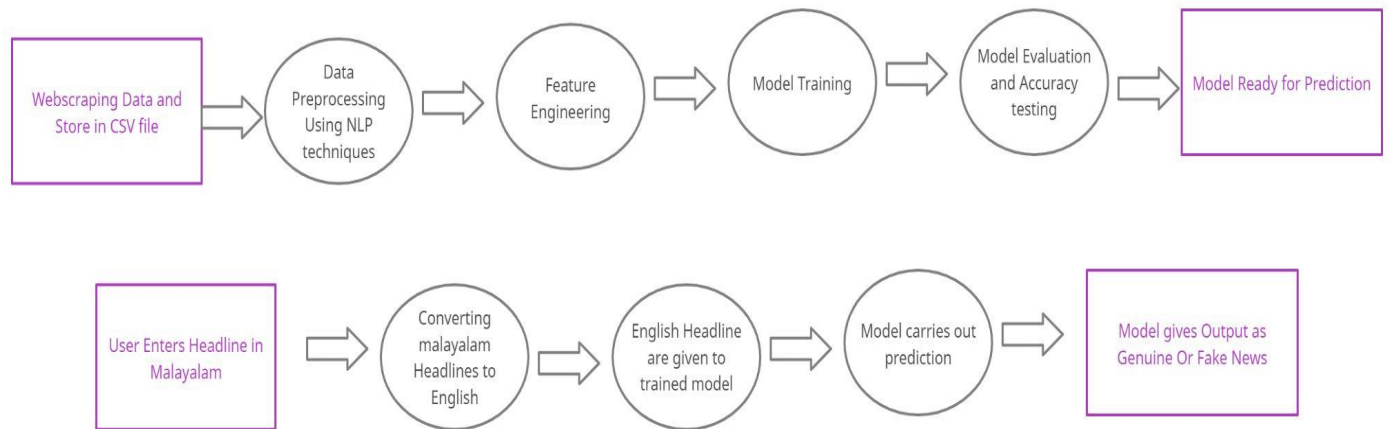
Analysis Modeling

4.1 Activity Diagrams / Class Diagram



(Fig2)

4.2 Functional Modeling

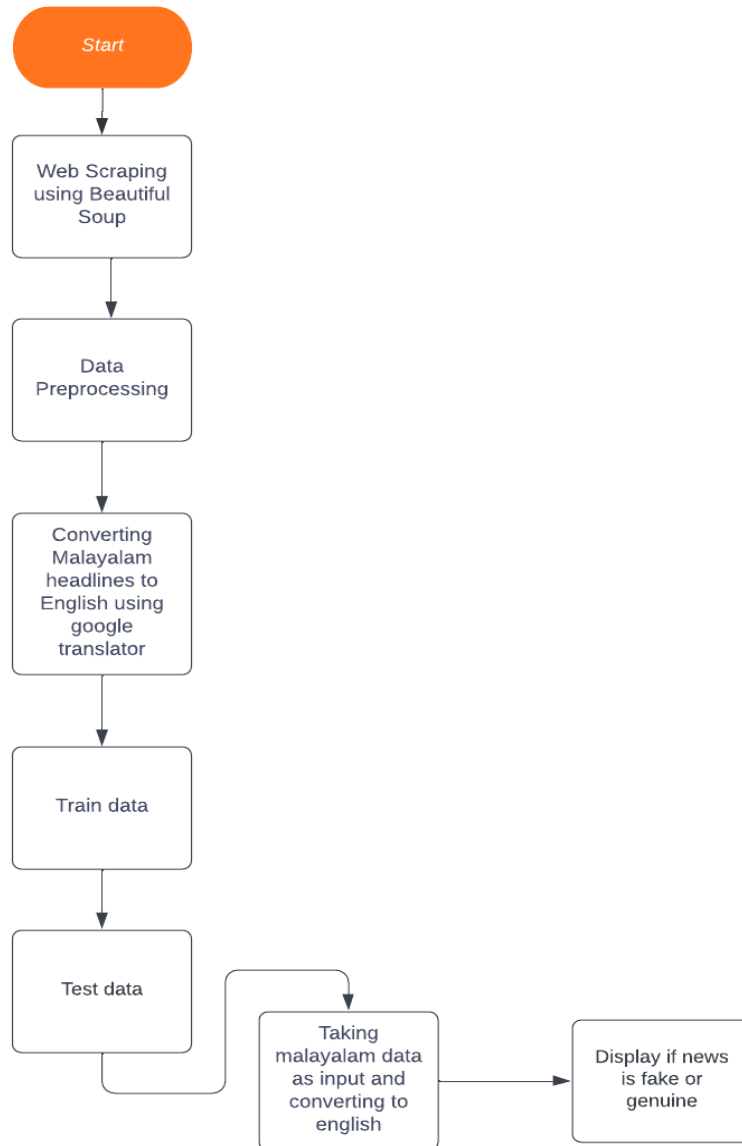


(Fig3)

Chapter 5

Design

5.1 Architecture Design:



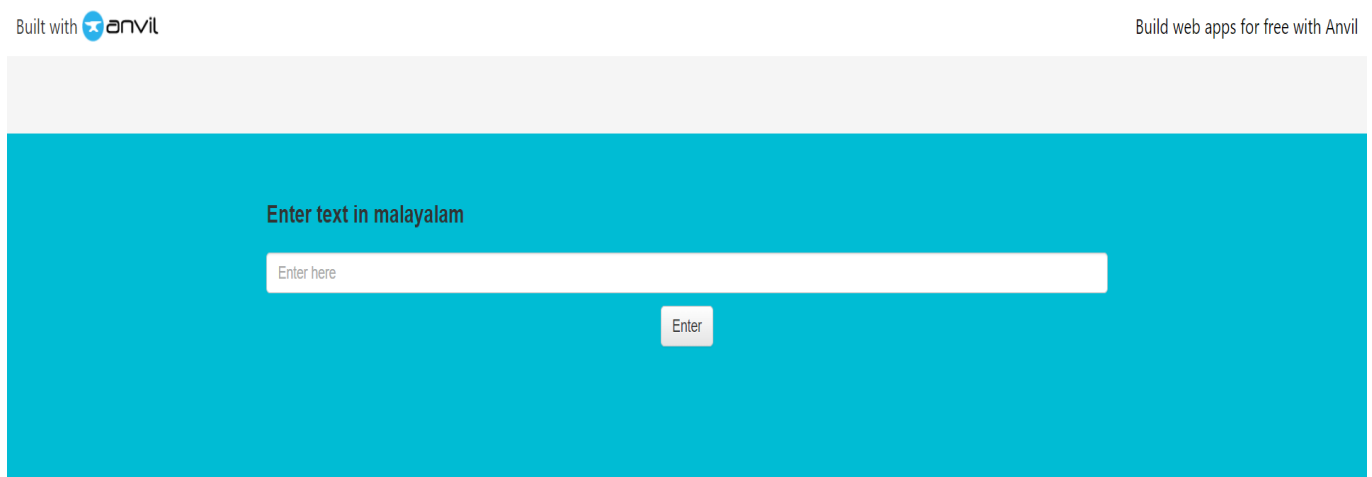
The project flow starts with data scraping that is scraping news headlines from various news sources by using beautiful soup4. The data that is collected is then merged into a csv file by using algorithms. This csv file is then cleaned and the unreadable characters are removed from the dataset. Thus we have a clean dataset which can be used for our project system. The model then

undergoes feature selection which is desirable to reduce the number of input variables to both reduce the computational cost of modeling and, in some cases, to improve the performance of the model. Thus we trained the model by giving various inputs into it.

Then the data is trained upon algorithms and we checked the accuracy of each algorithm that was applied on the dataset. The best algorithm is chosen and we have made a front end for our project using anvil. Here the user gives an input that is a news headline. The piece of news undergoes the algorithm and hence we get the output whether the news entered by the user is true or fake.

5.2 User Interface Design

Anvil is an open source platform for building and hosting full-stack web apps written entirely in Python. Drag & drop your UI, then write Python on the front-end and back-end to make it all work. It is a full-stack python UI builder with built-in database and user authentication. It supports on-site installation and team collaboration. Anvil has simple integration with existing services and code. It also has built-in support for all Python packages which allows the implementation and deployment of full stack dashboards quickly.



(Fig5)

Chapter 6

Implementation

6.1 Algorithms / Methods Used:

1. Logistic Algorithm:

We have created a simple logistic regression model to classify COVID news to either true or fake, using the data we have scrapped from various web sources. The process is simple and easy. We have cleaned and pre-processed the text data, performed feature extraction using NLTK library, built and deployed a logistic regression classifier and evaluated the model's accuracy at the end.

2. Support Vector Machine:

A support vector machine is a discriminative classifier. It uses the concept of a hyperplane to separate the two classes. The algorithm aims to establish an optimal hyperplane, which in two dimensions is a line dividing a plane in two parts with the corresponding classes on either side of the hyperplane.

3. Ensemble:

Ensembles can give you a boost in accuracy on your dataset. Ensemble learners tend to have higher accuracies, as more than one model is trained using a particular technique to reduce the overall error rate and improve the performance of the model.

4. Beautiful soup:

Beautiful Soup is a Python library for pulling data out of HTML and XML files. It works with your favorite parser to provide idiomatic ways of navigating, searching, and modifying the parse tree. It commonly saves programmers hours or days of work. These instructions illustrate all major features of Beautiful Soup 4, with examples. It will show what the library is good for, how it works, how to use it, how to make it do what you want, and what to do when it violates your expectations.

5. TFIDF Vectorization:

TF-IDF is an abbreviation for Term Frequency Inverse Document Frequency. This is a very common algorithm to transform text into a meaningful representation of numbers which is used to fit a machine algorithm for prediction.

In TfidfVectorizer we consider the 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.

6. Preprocessing:

Train Test Split is one of the important steps in Machine Learning. It is very important because your model needs to be evaluated before it has been deployed. And that evaluation needs to be done on unseen data because when it is deployed, all incoming data is unseen. The main idea behind the train test split is to convert the original data set into 2 parts train test where train consists of training data and training labels and test consists of testing data and testing labels.

6.2 Working of the project

Code:

Beautiful soup:

```
csv_writer = csv.writer(csv_file)

csv_writer.writerow(['new'])

for url in urls:

    response = requests.get(url)

    soup = BeautifulSoup(response.content, "html.parser")

for article in soup.find_all('div',class_=' _2ndvO _3MrhL'):

    try:

        date = article.find('div',class_='xOe7d').text

        news = article.find('div', class_=' _2VPRN').p.text

        new=(news +' on '+ date)

        print(new)

        csv_writer.writerow([new])
```

Function used:

```
def train_model(classifier, feature_vector_train, label, feature_vector_valid, is_neural_net=False):

    # fit the training dataset on the classifier

    classifier.fit(feature_vector_train, label)
```

```

# predict the labels on validation dataset

predictions = classifier.predict(feature_vector_valid)

if is_neural_net:

    predictions = predictions.argmax(axis=-1)

return metrics.accuracy_score(predictions, valid_y)

```

Logistic Regression:

```

accuracy = train_model(LogisticRegression(), xtrain_tfidf, train_y, xvalid_tfidf)

print("Logistic Regression", accuracy)

```

Support Vector Machine:

```

accuracy = train_model(svm.SVC(), xtrain_tfidf, train_y, xvalid_tfidf)

print("SVM: ", accuracy)

```

Ensemble:

```

accuracy = train_model(DecisionTreeClassifier(), xtrain_tfidf, train_y, xvalid_tfidf)

print("ENSEMBLE ", accuracy)

```

TFIDF Vectorization:

```

tfidf_vect = TfidfVectorizer(analyzer='word', token_pattern=r'\w{1,}', max_features=5000)

tfidf_vect.fit(dataframe['News'])

xtrain_tfidf = tfidf_vect.transform(train_x)

xvalid_tfidf = tfidf_vect.transform(valid_x)

dump(tfidf_vect, 'tfidf.pkl')

```

Preprocessing:

```
encoder = preprocessing.LabelEncoder()
train_y = encoder.fit_transform(train_y)
valid_y = encoder.fit_transform(valid_y)
```

load the dataset

```
[8] dataframe=pd.read_csv('/content/drive/MyDrive/Malayalam_News.csv')
#dataframe=pd.read_csv('/content/drive/MyDrive/MalayalamNews.csv') csv path - aishwarya
```

```
[9] dataframe.head()
```

	News	Outcome	mal_text
0	Pakistan Prime Minister Khan criticised over m...	1	കേസുകളായി യോഗത്തിൽ യോഗത്തിൽ യോഗത്തിൽ വിമർശിച്ചു.
1	Chhattisgarh reports 2,665 fresh Covid-19 case...	1	കഴിഞ്ഞ 24 മണിക്കൂറിനുള്ളിൽ 2,665 പുതിയ കോണിഡ് ...
2	Read: WHO urges countries to donate 10 million...	1	വായിക്കുക: 10 ദശലക്ഷം വാക്സിൻ ഡോസുകൾ സംഭാവന ചെ...
3	WHO in talks with India about vaccine exports ...	1	വാക്സിൻ കയറ്റുമതിയെക്കുറിച്ച് ആരാണു് ഇന്ത്യയുമാ...
4	Read - Covid-19: Maharashtra CM announces nigh...	1	വായിക്കുക - കോത്ത് -1: മാർച്ച് 28 മുതൽ സംസ്ഥാന...

(Fig 6)

```
[10] cedilla2latin = [[u'Á', u'A'], [u'á', u'a'], [u'Č', u'C'], [u'č', u'c'], [u'Š', u'S'], [u'š', u's']]
tr = dict([(a[0], a[1]) for (a) in cedilla2latin])
```

```
[11] nltk.download('stopwords')
stops = set(stopwords.words("english"))
def cleantext(string):
    text = string.lower().split()
    text = " ".join(text)
    text = re.sub(r"http(\S)+", ' ',text)
    text = re.sub(r"www(\S)+", ' ',text)
    text = re.sub(r"&", ' and ',text)
    tx = text.replace('&',' ')
    text = re.sub(r"^[0-9a-zA-Z]+", ' ',text)
    text = text.split()
    text = [w for w in text if not w in stops]
    text = " ".join(text)
    return text
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data] Unzipping corpora/stopwords.zip.
```

```
[12] for line in dataframe:
    line = cleantext(line)
```

(Fig 7)

Function to train Classifier and calculate Accuracy

```
[22] def train_model(classifier, feature_vector_train, label, feature_vector_valid, is_neural_net=False):  
    # fit the training dataset on the classifier  
    classifier.fit(feature_vector_train, label)  
  
    # predict the labels on validation dataset  
    predictions = classifier.predict(feature_vector_valid)  
  
    import pickle  
    pickle.dump(classifier, open('kuy.pkl', 'wb'))  
  
    if is_neural_net:  
        predictions = predictions.argmax(axis=-1)  
  
    return metrics.accuracy_score(predictions, valid_y)
```

(Fig 8)

SVM

A support vector machine (SVM) is a type of deep learning algorithm that performs supervised learning for classification or regression of data groups. In AI and machine learning, supervised learning systems provide both input and desired output data, which are labeled for classification.

```
[23] def SVM():  
    accuracy = train_model(svm.SVC(), xtrain_tfidf, train_y, xvalid_tfidf)  
    print("SVM: ", accuracy)  
    SVM()
```

SVM: 0.9476594176188721

(Fig 9)

```
tr = Translator()  
inp = input("Enter a statement in malayalam >>>> ")  
eng = tr.translate(inp).text  
eng = eng.lower().replace('\W+', " ")  
  
removed_stopword = []  
for word in eng.split():  
    if word not in stops:  
        removed_stopword.append(word)  
  
eng = np.array([" ".join(removed_stopword)])  
eng
```

Enter a statement in malayalam >>>> കേസുകളായി യോഗത്തിൽ യോഗത്തിൽ യോഗത്തിൽ വിമർശിച്ചു.
array(['cases criticized meeting meeting.'], dtype='<U33')

(Fig 10)

```
[27] import joblib
import time
import pandas as pd

import pickle
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.feature_extraction.text import CountVectorizer
import numpy as np

vectorizer = open("/content/tfidf.pkl", "rb")
cv = joblib.load(vectorizer)

# load Model For News Prediction
svm_model = open("/content/kuy.pkl", "rb")
clf = joblib.load(svm_model)
```

(Fig 11)

```
 @anvil.server.callable
def predict_fake(inp):
    tr = Translator()
    eng = tr.translate(inp).text
    eng = eng.lower().replace('\W+', " ")

    removed_stopword = []
    for word in eng.split():
        if word not in stops:
            removed_stopword.append(word)

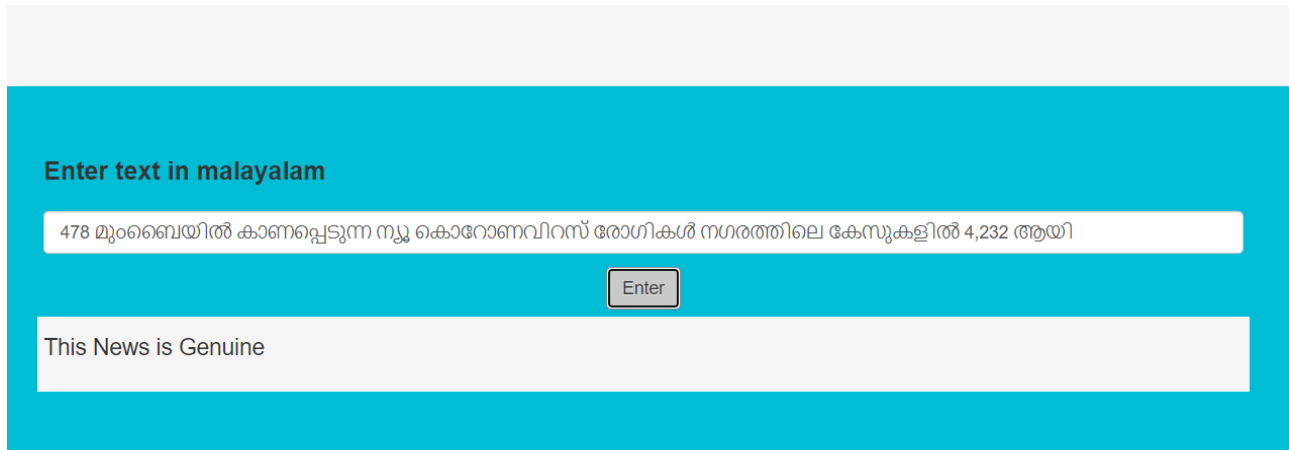
    eng = np.array([" ".join(removed_stopword)])
    tfidf_vect = TfidfVectorizer(analyzer='word', token_pattern=r'\w{1,}', max_features=5000)
    tfidf_vect.fit(dataframe['News'])
    X=tfidf_vect.transform(eng)

    result = clf.predict(X)
    if (result >= 0.5):
        return("Genuine")

    else:
        return("Fake")
```

(Fig 12)

Screenshots of UI:



Enter text in malayalam

478 മുറിയ്ക്കലിൽ കാണപ്പെടുന്ന ന്യൂ കൊറോണാവിറസ് രോഗികൾ നഗരത്തിലെ കേന്ദ്രങ്ങളിൽ 4,232 ആയി

Enter

This News is Genuine

(Fig 13)



Enter text in malayalam

ഓക്സിജൻ അളവ് അളക്കുന്നതിനായി ഫോൺ ക്യാമറ ഉപയോഗിക്കുന്ന അപ്ലിക്കേഷൻ

Enter

This News is Fake

(Fig 14)

Chapter 7

Conclusion

It can be seen that machine learning algorithms can be applied for detection of fake news. We see that SVM algorithm gives the highest accuracy for the news dataset. This project differentiates whether the news entered in the input is true or fake news. Fake news and hoaxes have been there since before the advent of the Internet. The widely accepted definition of Internet fake news is: fictitious articles deliberately fabricated to deceive readers”. 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.

Clickbaits lure users and entice curiosity with flashy headlines or designs to click links to increase advertisements revenues. This exposition analyzes the prevalence of fake news in light of the advances in communication made possible by the emergence of social networking sites. Along with COVID-19 pandemic we are also fighting an 'infodemic'. Fake news and rumors are rampant on social media. Believing in rumors can cause significant harm. This is further exacerbated at the time of a pandemic. The purpose of the work is to come up with a solution that can be utilized by users to detect and filter out false and misleading information. We used Word Level TF-IDF for feature selection on a dataset of 10,000 news headlines of real and fake news on COVID-19. Then we trained the classifier using three machine learning algorithms - ensemble , Logistic Regression and Support Vector Machine (SVM). We obtained the best performance of SVM: 0.9487652045705861. The Gui was prepared using Anvil. The purpose of the work was to come up with a solution that can be utilized by users to detect and filter out sites containing false and misleading information and we successfully implemented the same.

References

- Sa, Ahmed & hinkelmann, knut & Corradini, Flavio. *“Development of Fake News Model using Machine Learning through Natural Language Processing”*, 2020.
- K. Agarwalla, S. Nandan, V. A. Nair, D. D. Hema, “Fake News Detection using Machine Learning and Natural Language Processing”, International Journal of Recent Technology and Engineering (IJRTE),ISSN: 2277-3878, Volume-7, Issue-6, March 2019

Acknowledgments

We express our deep sense of gratitude to our project guide Mrs. Vincy Joseph for encouraging us and guiding us throughout this project. We were able to successfully complete this project with the help of her deep insights into the subject and constant help.

We are very much thankful to Dr. .Kavita Sonawane, HOD of the COMPS department at St. Francis Institute of Technology for providing us with the opportunity of undertaking this project which has led to us learning so much in the domain of Natural Language Processing.

Last but not the least we would like to thank all our peers who greatly contributed to the completion of this project with their constant support and help.