Fake News Detection From Pandemic News

Design project report submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology

in

Computer Science & Engineering

Submitted by

Sabarinath R

Sabir Ebrahim

Richard T S

FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)
ANGAMALY-683577

Affiliated to

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY
KERALA

DECEMBER 2020

FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)
Mookkannoor(P.O), Angamaly-683577

CERTIFICATE

This is to certify that project report entitled "**Fake News Detection From Pandemic News**" is a bonafide report of the Design Project(CS 341) presented during V^{th} semester by **Sabarinath R (FIT18CS111),Sabir Ebrahim (FIT18CS112), Richard Sebastian (FIT18CS107)**, in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology (B.Tech) in Computer Science & Engineering during the academic year 2020-2021.

Dr. Paul P Mathai
Project Guide

Dr. Prasad J C
Head of the Department

ABSTRACT

For some years, mostly since the rise of social media, fake news have become a society problem, in some occasion spreading more and faster than the true information. 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. Even an expert in a particular domain has to explore multiple aspects before giving a verdict on the truthfulness of an article. As fake news creates a serious problem in our society during this pandemic situation it is necessary to check the credibility of a news before sharing it to others and before believing its content. We, with the use various machine learning classifiers, Natural Language Processing and use of vectorizing techniques comes up with a system to check the credibility of a news. Along with the vectorized features of news content we have also extracted and added other textual features in order to increase the accuracy and efficiency of our model.

ACKNOWLEDGMENT

Apart from the efforts put in by us, the success of this project depends largely on the encouragement and guidelines of many others.

We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

Ms. ANITHA P, chairman, FISAT Governing Body, who provided us with vital facilities required by the project right from inception to completion.

We would like to express our heartfelt thanks to our Principal Dr.George Isaac and Dr. Prasad J C, Head of the Department of Computer Science and Engineering for their support.

I am very grateful to Mr. Pankaj Kumar G, Ms. Remya R, Ms. Chethana Joy, Ms. Roshna K I, Dr. Paul P Mathai for their expert guidance and continuous encouragement throughout to see that this project reaches its target.

Sabir Ebrahim

Richard T S

Sabarinath R

Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

1.1 Overview

The advent of the World Wide Web and the rapid adoption of social media platforms (such as Facebook and Twitter) paved the way for information dissemination that has never been witnessed in the human history before. Besides other use cases, news outlets benefitted from the widespread use of social media platforms by providing updated news in near real time to its subscribers. The news media evolved from newspapers, tabloids, and magazines to a digital form such as online news platforms, blogs, social media feeds, and other digital media formats. It became easier for consumers to acquire the latest news at their fingertips.
However, such platforms are also used with a negative perspective by certain entities commonly for monetary gain and in other cases for creating biased opinions, manipulating mindsets, and spreading satire or absurdity. The phenomenon is commonly known as fake news. Increase in the spread of such fake news is a serious issue when the news is about a global pandemic just like covid-19.
In this paper, we identified and extracted a set of features present in news(title and text) that allows the accurate prediction of reliability of a news based on various super-vised machine learning techniques.
In the first phase of our project , we applied four machine learning techniques namely Logistic Regression, Decision Tree and Passive Aggressive Classifier to classify News using a Covid-19 News dataset composed of about 1200 Covid-19 News data out of which nearly half of them is labelled as fake. The machine was training using features extracted from TF-IDF Vectorization techniques.
In the next phase, the research was extended using other machine learning algorithms like Support Vector Machine (Linear,Poly,RBF) and Random Forest for selecting the best one after comparing their accuracy on our dataset. Along with the features extracted using TF-IDF vectorization techniques , some other manually extracted features from the news were also used to train the model in order to increase the accuracy and efficiency of our Project. Model performance with the use of n-grams were also analysed.

1.2 Scope of project

Fake news has quickly become a society problem, being used to propagate false or rumour information in order to change people's behaviour. our world view is shaped on the basis of information we digest. There is increasing evidence that consumers have reacted absurdly to news that later proved to be fake . One recent case is the spread of novel corona virus, where fake reports spread over the Internet about the origin, nature, and behaviour of the virus . The situation worsened as more people read about the fake contents online. Identifying such news online is a daunting task. Our project is a small contribution to the society to help the society to check the credibility of a news. It also helps in preventing chaos in this pandemic situation . The scope of our project as of now if confined only on the prediction using the textual features of the given news .

1.3 Objectives

As information is spreading faster than ever before it is necessary to identify which is fake and which is real . So we develop a system which can predict whether a news is real or not by giving its title and text as input to a UI which then with the help of our trained model classify the news as fake or real. Thus it helps the user to check the credibility of the news and it could prevent the spread false information. We are building a package for fake news detection so that it will be easy for others to use our model for checking news reliability and it will be easy to develop/modify the system for advanced performance.

Chapter 2

Literature Survey

There has been a rapid increase in the spread of fake news in the last decade, most prominently observed in the 2016 US elections [1]. 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 such area affected by fake news is the medical and pandemic sector where a rumour can have disastrous consequences and may bring harm to society and economy.

The World Wide Web contains data in diverse formats such as documents, videos, and audios. News published online in an unstructured format (such as news, articles, videos, and audios) is relatively difficult to detect and classify as this strictly requires human expertise. However, computational techniques such as natural language processing (NLP) can be used to detect anomalies that separate a text article that is deceptive in nature from articles that are based on facts [2]. Other techniques involve the analysis of propagation of fake news in contrast with real news . More specifically, the approach analyzes how a fake news article propagates differently on a network relative to a true article. The response that an article gets can be differentiated at a theoretical level to classify the article as real or fake. A more hybrid approach can also be used to analyze the social response of an article along with exploring the textual features to examine whether an article is deceptive in nature or not.

A number of studies have primarily focused on detection and classification of fake news on social media platforms such as Facebook and Twitter [3,4]. At conceptual level, fake news has been classified into different types; the knowledge is then expanded to generalize machine learning (ML) models for multiple domains [5]. The study by Ahmed et al. [6] included extracting linguistic features such as n-grams from textual articles and training multiple ML models including K-nearest neighbor (KNN), support vector machine (SVM), logistic regression (LR), linear support vector machine (LSVM), decision tree (DT), and stochastic gradient descent (SGD), achieving the highest accuracy (92 percentage) with SVM and logistic regression. So we decided to use these machine learning algorithams and nlp techniques to create a new and improvised fake news detection system.

2.1 Background Study

Studies were conducted with regard to the existing systems and three distinct machine learning algorithms were selected in the initial phase based on the research and discussions.

Passive aggressive classifier:Passive aggressive classifiers are a family of Machine learning algorithms that are generally used for large-scale learning. It is one of the few online-learning algorithms. It categorize data into diffarent classes by learnig the relationship from a given sequence of input patterns We can simply say that an online-learning algorithm will get a training example, update the classifier, and then throw away the example[7]

Decision Tree: The Decision Tree algorithm is a supervised machine learning technique which can be used for both Classification and Regression type problems.In Decision Trees, for predicting a class label for a record we start from the root of the tree. We compare the values of the root attribute with the record's attribute. On the basis of comparison, we follow the branch corresponding to that value and jump to the next node .The tree splits into branches till it reaches the state where no further classification is possible hence the tree is build[8]

Logistic regression:Logistic regression is a classification algorithm used to find the probability of event success and event failure. It is used when the dependent variable is binary(0/1,True/False, Yes/No) in nature. It supports categorizing data into discrete classes by studying the relationship from a given set of labelled data. It learns a linear relationship from the given dataset and then introduces a non-linearity in the form of the Sigmoid function[9]

2.1.1 Logistic Regression

Logistic regression is a classification algorithm used to find the probability of event success and event failure. It is used when the dependent variable is binary(0/1, True/False, Yes/No) in nature.It supports categorizing data into discrete classes by studying the relationship from a given set of labelled data.It learns a linear relationship from the given dataset and then introduces a non-linearity in the form of the Sigmoid function.[1] The formula for the sigmoid function is the following:

$$sigmoid(x) = \frac{1}{1 + e^x}$$

By computing the sigmoid function of X (that is a weighted sum of the input features), we get a probability (between 0 and 1 obviously) of an observation belonging to one of the two categories. That is logistic regression can be simply written as prediction P(y) as a function of X[9,10]
We can classify news as FAKE and REAL and predict whether a given news is FAKE using logistic regression method.
Here we take title and body of the news data(text) as features.
Applying nlp techniques we splits the data into words and preprocess the data .
Applying tf-idf vectorizer will give the vectorized value of each words of each news with unique weightage for each. With the help of this vectorized value we trains the logistic regression model in which we will give labels fake or real as the target . It learns a linear relationship from the given dataset, and then introduces a non-linearity in the form of the Sigmoid function.
Then we introduce a new news¿ apply nlp techniques and tf idf vectorization¿Calculate probability for the target fake and for the target real.
*target class with high probability is what we get as the predicted target for the given news.

2.1.2 Passive Aggressive Classifier

Passive aggressive classifier creates a hyper plane to categorize data into two,and it creates a margin wich is proportional to the distance of hyperplane and this margin is used to correct the errors in the prediction. When ever a misclassification occurs the PA algorithm updates the classification model.The core concept is that the classifier adjusts it weight vector for each mis-classified training sample it receives, trying to get it correct. Let Xt and Y t be a sequence of input patterns for learning were Xt is the input data and Yt is the output data(TRUE or FALSE) for t number of times. At first we use a binary linear classifier Formula for linear classification is:

$$Y_t' = sign(W'^T + X_t)$$

learner first receives an incoming instance Xt at t th round; it then makes a prediction based on the current classification model,if the prediction is correct then Yt' =Yt,then no update is applied to the linear classifier if the prediction is not correct PA updates the classification algorithm,and from the next classification onwards the updated classification model is used[7,11]

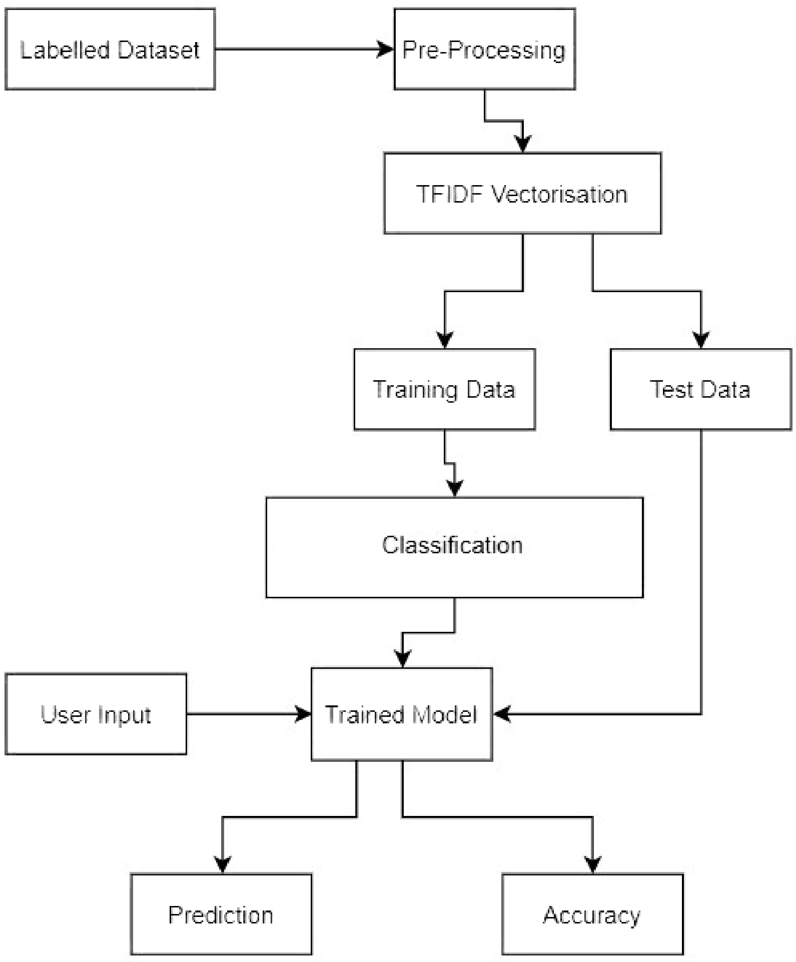
2.1.3 Decision Tree

. A Decision tree is a supervised learning algorithm , which can be used for classification and regression problems .In a Decision tree each node represents a feature, each link or branch represents a decision and each leaf represents an outcome. In decision trees, for predicting a class label for a record we start from the root of the tree. We compare the values of the root attribute with record's attribute. On the basis of comparison, we follow the branch corresponding to that value and jump to the next node. The comparison continues until the leaf node is encountered with the predicted class value. Initially the whole training data is considered as the root, Then according to the feature the at the root node the data gets split and this continues till no further classification is possible . The node is divided into sub-nodes using Gini impurity

$$Gini \quad impurity = 1 - \sum_{i=1}^{n} pi^2$$

Here for every split the gini impurity of each child node is calculated and compared, then the split with the lowest Gini impurity value is selected. This splitting continues till homogeneous nodes are encountered. Lower the Gini impurity higher is the homogeneity of the node.[8]

2.1.4 Block Diagram



Block diagram of Phase-1

2.1.5 Methods

Preprocessing

Tokenization: Tokenization is a common task in Natural Language Processing (NLP). It's a fundamental step in both traditional NLP methods like Count Vectorizer and Advanced Deep Learning-based architectures like Transformers.

Tokenization is a way of separating a piece of text into smaller units called tokens. Here, tokens can be either words, characters, or subwords. Hence, tokenization can be broadly classified into 3 types – word, character, and subword (n-gram characters) tokenization.[12]

Stopwords Removal: 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. Such words are already captured this in corpus named nltk corpus. We first download it to our python environment. Using the nltk library such words can be removed from out training data.[13]

Lemmatization: Lemmatization is the process of converting a word to its base form. Lemmatization considers the context and converts the word to its meaningful base form. Thus we can prevent words with similar meaning having different values after vectorization[14]. Here we use nltk wordnet lemmatizer for our project.

TF-IDF Vectorization

TF-IDF stands for Term Frequency - Inverse Document Frequency and is a statistic that aims to better define how important a word is for a document, while also taking into account the relation to other documents from the same corpus. This is performed by looking at how many times a word appears into a document while also paying attention to how many times the same word appears in other documents in the corpus. The rationale behind this is the following:

- a word that frequently appears in a document has more relevancy for that document, meaning that there is higher probability that the document is about or in relation to that specific word
- a word that frequently appears in more documents may prevent us from finding the right document in a collection; the word is relevant either for all documents or for none. Either way, it will not help us filter out a single document or a small subset of documents from the whole set.

So then TF-IDF is a score which is applied to every word in every document in our dataset. And for every word, the TF-IDF value increases with every appearance of the word in a document, but is gradually decreased with every appearance in other documents.[15]

Training and Testing

The prepared dataset is randomly split into training and testing data.Training the Logistic Regression model begins by calculating the tf-idf value of each news after preprocessing. According to the features extracted by tf-idf the dataset gets partitioned accordingly and the model is build. After training ,the accuracy of the model to perform correct predictions is determined by the process of testing. Testing involves the right prediction of each test data given to the model and predits the class for each testing data.The accuracy is the ratio of the correct predictions to the total number of samples during testing.
In this work,80for testing the trained model. The trained model is saved as a model.sav which can be reused later for making predictions for an unknown input.

Deployment of the trained model

The input retrieved from the user is the title and text of news to be checked. Relevant numerical features are extracted from the input using tf-idf to form a single feature vector. The trained model is deployed to these features values to make appropriate prediction. The prediction retrieved is a binary value, the label 1 corresponds to fake news and 0 corresponds to real news.

2.1.6 Algorithm

Logistic regression

Step 1 : START

Step 2 : Inputs dataset of fake and real news . with labelsfake and real(text data)

Step 3 : Split data using train test split method.

Step 4 : Preprocessing the test data using NLP techniquessuch as tokenization,lemmetization etc

Step 5 : Applying Tf-idf vectorization.

Step 6 :Applying logistic regression Classifier on thevectorized data along with the target label and save the trained model . (Learns a linearrelationship from the given dataset with the labelled class.)

Step 7 : Giving new user input

  Step 7.1 : Calculating the probability of being fake and real bystudying the relationship from the given set of labelled data.

Step 8 : Fake or Real Prediction

Passive Aggressive Classifier

Step 1 : START

Step 2 : Inputs dataset of fake and real news . with labelsfake and real(text data)

Step 3 : Split data using train test split method.

Step 4 : Preprocessing the test data using NLP techniquessuch as tokenization,lemmetization etc

Step 5 : Applying Tf-idf vectorization.

Step 6 : Applying Passive Aggresive Classficattion to thevectorized data along with the target label.

  Step 6.1 :It will linearly classify the given datasets andcreates a relationship

Step 7 : Giving new user input

  Step 7.1 : It will Predict the probability of being fake or real bycomparing with the relationship from the trained dataset

  Step 7.2 : If there is any misclassification occured the classifierwill update the classifcation model itself by adjusting theweight vector for the next prediciton

Step 8 : Fake or Real(Prediction according to the classifier)

Decision Tree

Step 1 : Start

Step 2 : Read data set

```
Step 3 : Split data set into training and test data
Step 4 : create n number of bootstrapped data sets from the training set
Step 5 : Select random subspace of features from the data set
Step 6 : Build the decision trees associated with the selected data points
    Step 6.1 : compute the entropy for the data set Entropy(s)
    Step 6.2 : For every attribute of the data set
        Step 6.2.1 : calculate entropy for all other values Entropy(A)o
        Step 6.2.2 : Take average information entropy for the currentat tributeo
        Step 6.2.3 : Calculate gain for the current attribute
    Step 6.3 : pick the highest gain attribute
    Step 6.4 : repeat until we get the tree we deserve
Step 7 : Evaluate prediction of the decision tree
Step 8 : Measure accuracy of prediction
Step 9 : stop
```

2.1.7 Comparison between different methods

Comparisan between 3 methods used by our each team members

No	Classification	Percentage
1	Logistic Regression	91.8%
2	Passsive Aggressive Classifier	92.7%
3	Decision Tree	84.5%

Table 2.1: Comparison Table

Of all 3 classifications methods used above logistic regression proved to be the fastest without compormising the accuracy of classification,so logistic regression was taken.

Chapter 3

DESIGN

The goal of this thesis is to implement the fake news detection library "FakeNews-Detection" Which employs a textual feature-based machine learning model and is easy to use, modify, and extend.

In the second phase of the project, we implement an effcient fake news prediction system that is a feature based classifier using Logistic Regression Model which yielded us the best accuracy on comparison with Decision Tree, Random Forest ,SVM etc.
Source and text content are the mainly used components for fake news identification. Hence both these elements can be used for identifying the features for the preparation of the dataset that serves the machine learning model. However, the system proposed in this project is a prediction model that feeds on the text-based features of a news rather than its source, since there are many rumours and news without any source provided.

3.1 Problem Statement

A web application to check whether a given news is fake or real using NLP and machine learning techniques. The purpose of this project is to utilize efficient machine learning algorithms for the classification of fake and real news based on the features extracted from text and title of the news with the help of Natural Language Processing Techniques.
At the times of pandemics like covid-19 ,the spread of fake news is creating great havoc and misunderstandings in our society. This work concentrates on predicting the reliability of a news content from any media platform using NLP and machine learning algorithms. We are also developing a web UI and a streamlit ml app that predicts the nature of a news using title and text as input .
A streamlit data analysis interface also have been developed in order to analyse and study the performance of difference machine learning algorithms on our dataset.

3.2 Framework Overview

A broad range of multiple features extracted from the dataset consisting of Fake/Real News serves the classifer model.A Logistic Regression model,which is a supervised machine learning model.To make more effcient and accurate predictions for any user input new text features like unwanted capitalisation,number of word count in title and text and number presence in the text

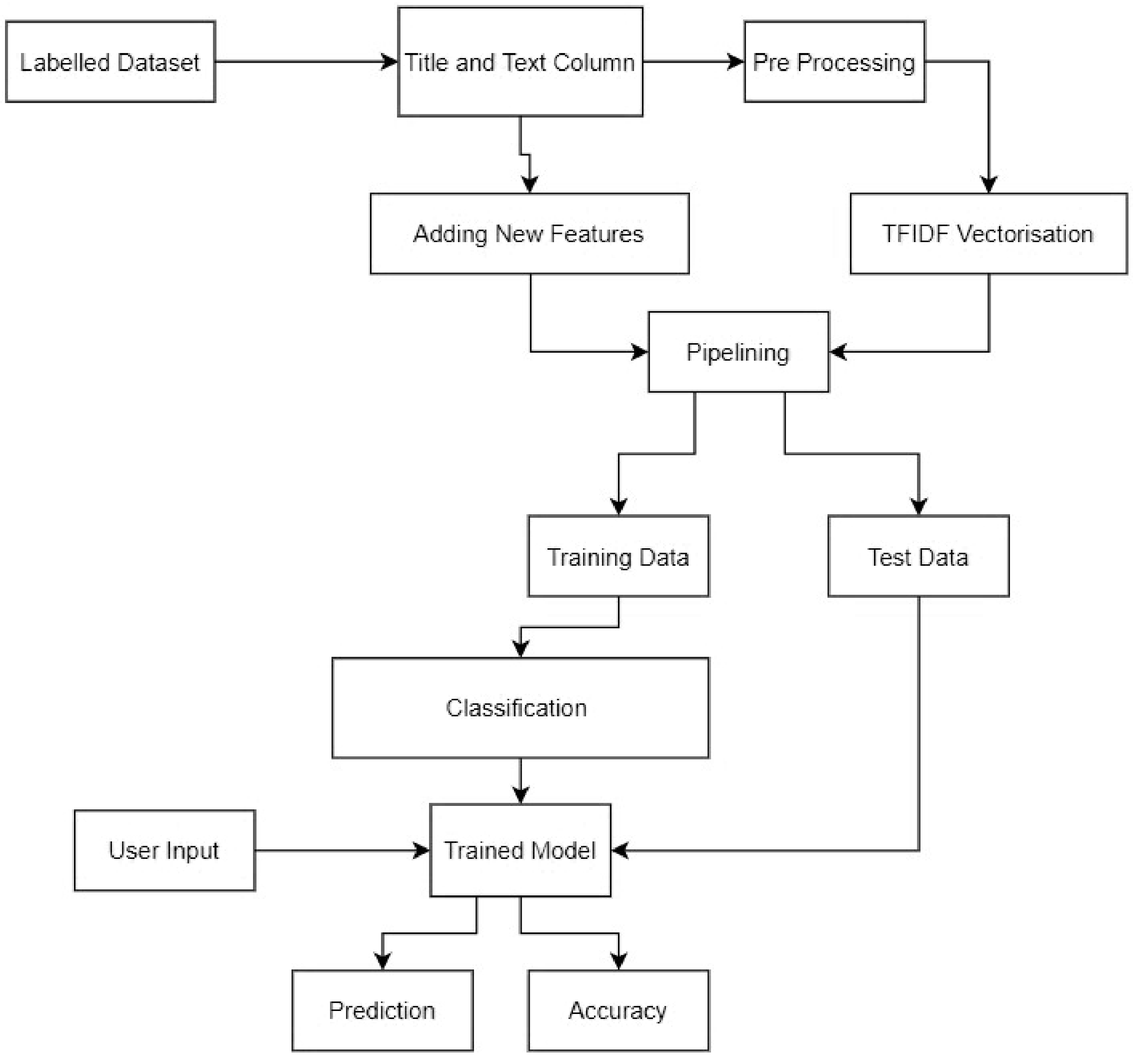


Figure 3.1: Block diagram of Framework

3.3 Algorithm

Algorithm for training

Input : A covid news dataset consisting of list of true as well as fake labelled news
Output : Trained model,accuracy

```
Step 1  : START
Step 2  : Inputs dataset of fake and real news . with labels fake and real(text data).
Step 3  : Select title,text and label columns from the csv file.
    Step 3.1 : Extract new features from the text and title of news.
Step 4  : Preprocessing title and text using NLP techniques such as tokenization,lemmetization
          etc
Step 5  : Applying Tf-idf vectorization on title and text.
Step 6  : Using pipeline feature union, join vectorized features along with newly
          extracted features to create new dataset for training.
Step 7  : Split testing and training data
Step 8  :Applying logistic regression Classifier on the dataset along with the target
          label . ( Learns a linearrelationship from the given dataset with the labelled
          class.)
Step 9  : Save the trained model for accuracy analysis and future use.
Step 10 : STOP
```

Algorithm for prediction

Input : Text and title of a new news.
Output : Predicted label(fake or real) of the fresh news input

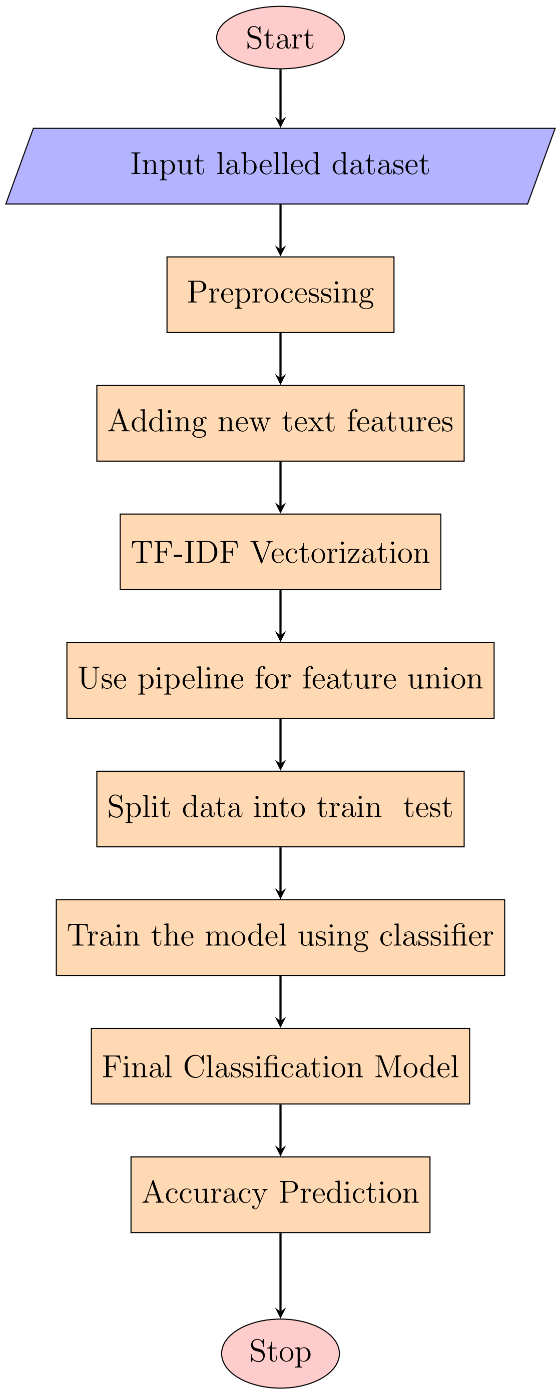
```
Step 1  : START
Step 2  : Input text and title of the news to be tested.
    Step 2.1 : Extract new features from the text and title of news.
Step 3  : Preprocessing title and text using NLP techniques such as tokenization,lemmetization
          etc
Step 4  : Applying Tf-idf vectorization on title and text.
Step 5  : Using pipeline feature union, join vectorized feature along with newly
          extracted features.
Step 6  : Classification using trained model.
```

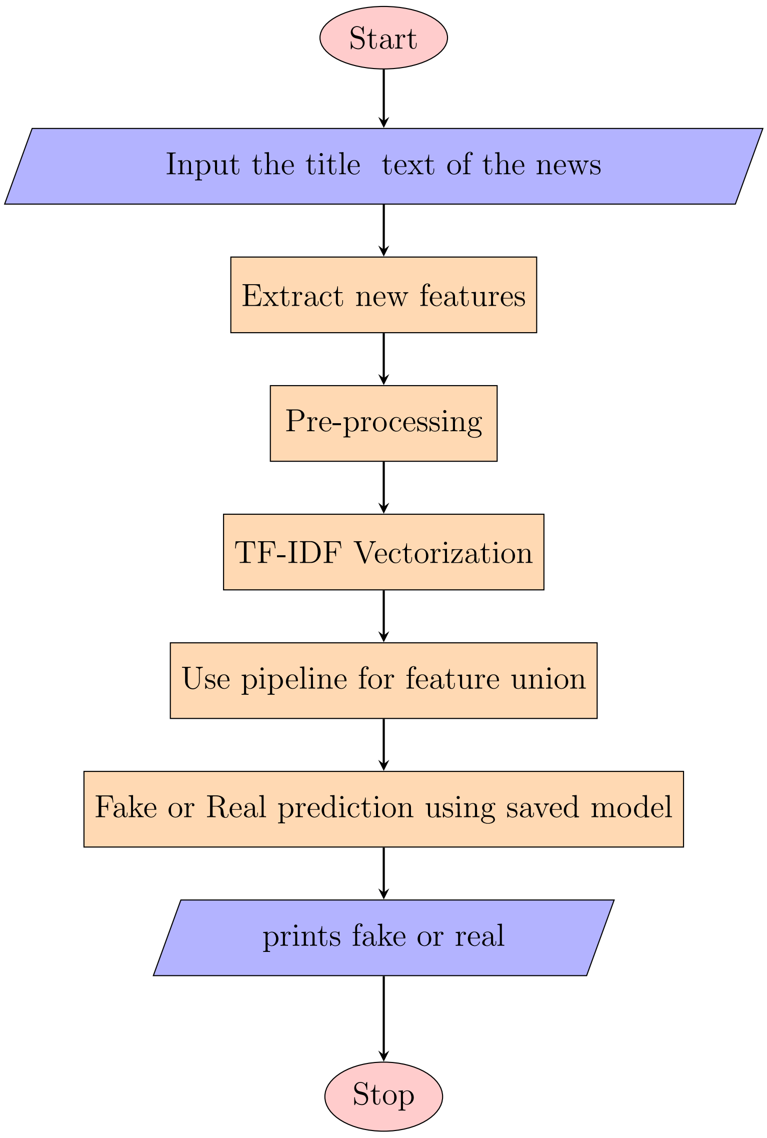
Step 6.1 : Calculating the probability of being fake and real by studying the relationship from the given set of labelled data in the trained model.

Step 7 : Fake or Real Prediction.

Step 8 : STOP

3.4 Flowchart


Start
Input labelled dataset
Preprocessing
Adding new text features
TF-IDF Vectorization
Use pipeline for feature union
Split data into train test
Train the model using classifier
Final Classification Model
Accuracy Prediction
Stop


Start
Input the title text of the news
Extract new features
Pre-processing
TF-IDF Vectorization
Use pipeline for feature union
Fake or Real prediction using saved model
prints fake or real
Stop

3.5 System Requirements

3.5.1 Hardware Requirements

- 512MB Ram
- intel Pentium Processor
- internet connection

3.5.2 Software Requirements

- Python 3.7(3.8/3.9 also compatible)
  - sklearn
  - pandas
  - nltk
  - pickle
  - streamlit

  - matpotlib
  - Flask Web framework

- HTML5
- CSS

3.6 Design Methodology

In Python, we created a library “FakeNewsDetection” to detect the reliability of a news. First we need to load and operate the dataset(csv file), and in order to do that we used the module pandas.
And we used many dependency libraries like NLTK (for applying NLP techniques on text data for cleaning and preprocessing), picke(for saving and loading trained mode), sklearn (has many machine learning models like logistic regression which is used here for training and prediction. It also provides vectorization and pipelining functions prebuild) etc. Streamlit web framework tool was used in our project to create a fake news prediction ml app and to create a data streamlit data analysis interface analyse and study the performance of difference machine learning algorithms on our dataset.
Here we have also used flask library for the implementation of the prediction Webui.
Matpotlib library was used to deploy a graphical comparison of different models in the streamlit web ui.
• **Scikit-learn**: It provides a selection of effcient tools for machine learning and statistical modeling including classiffcation, regression, clustering and dimensionality reduction via a consistence interface in Python.

• **Pandas**:It is a fast, powerful, exible and easy to use open source data analy- sis and manipulation tool, built on top of the Python programming language

* **nltk**: NLTK stands for Natural Language Toolkit. This toolkit is one of the most powerful NLP libraries which contains packages to make machines understand human language and reply to it with an appropriate response. Tokenization, Stemming, Lemmatization, Punctuation, Character count, word count are some of these packages

* **pickle** : Python pickle module is used for serializing and de-serializing python object structures. The process to converts any kind of python objects (list, dict, etc.) into byte streams (0s and 1s) is called pickling or serialization or flattening or marshalling. We can converts the byte stream (generated through pickling) back into python objects by a process called as unpickling.

***streamlit**: Streamlit is a popular open-source framework used for model deployment by machine learning and data science teams. Streamlit is an open-source Python library that makes it easy to create and share beautiful, custom web apps for machine learning and data science. Streamlit allows you to convert your normal python code into beautiful UI which is quite easy and very interactive.

***flask**:Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries

***Matplotlib**:Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python

3.7 Dataflow Diagram

3.7.1 Level 0

Level 0 data flow diagram shows an abstraction view of the system where input is a news(title and text) from the user side. The input from user is fed to the trained news classification system from where output can be retrieved using the technique of class prediction. The figure below shows the Level 0 dataflow diagram.

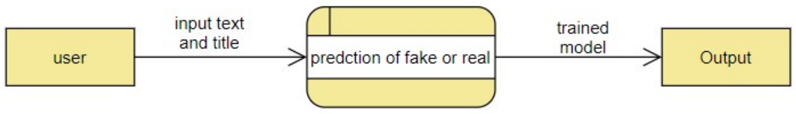


Figure 3.2: Level 0

3.7.2 Level 1

Level 1 dataflow diagram is shown above. We give a labelled covid news dataset as input for training. After cleaning and pre-processing we train the model using different classifiers to predict whether a given news is fake or not.

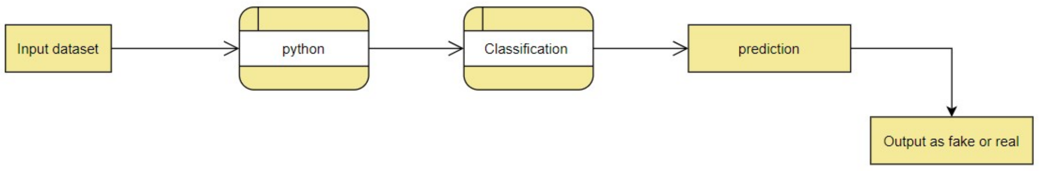


Figure 3.3: Level 1

3.7.3 Level 2

In this level what we are doing is that we pre-process the input dataset(Covid-news dataset) using Tokenization, Lemmatization , and stop-words removal. Then we vectorize the text contents of news in the dataset using TFIDF vectorisation. Along with the vectorized value we are adding additional text features like word count in the title, word count of the whole text , counting of unwanted capitalisation in the tittle and number presence. Then with the use of different classifying algorithms (logistic regression, decision tree , svm etc.) we compare the accuracies of the prediction. As we get the highest accuracy for Logistic Regression we create model for prediction using Logistic Regression . Using the prediction model the user can predict whether the new news is real or not .

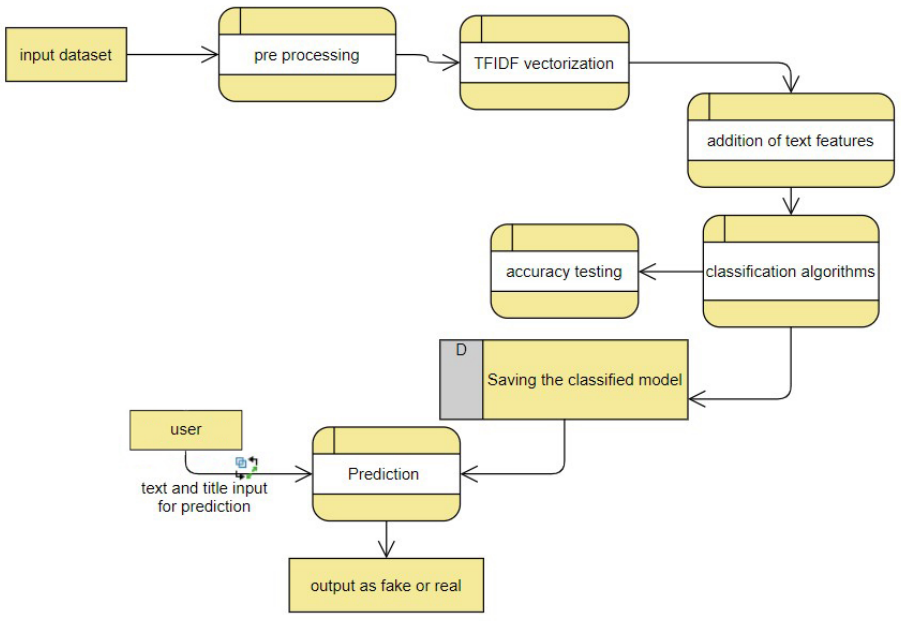

input dataset
pre processing
TFIDF vectorization
addition of text features
classification algorithms
accuracy testing
D
Saving the classified model
user
text and title input for prediction
Prediction
output as fake or real

Figure 3.4: Level 2

3.8 Data set /Resource development/design

The initial step of implementation is to collect data of fake and real news related to pandemic. The second step is to identify relevant features from these news that label it as fake or real. These features are extracted and then used for training and testing along with vectorized features of news on the selected machine learning algorithm.

3.8.1 Dataset

A Covid-19 News dataset composed of about 1200 Covid-19 News data out of which nearly half of them is labelled as fake . It was found from the internet[16]. The dataset is given here :
`https://drive.google.com/drive/folders/1cp62MA3ZwafUWgMukrFushdJ9lXkQ6e5?usp=sharing`

3.8.2 Features

For the identification of the features each news was analysed from fake and True class. These identified features are programmatically extracted from each news for the preparation of the final dataset. For this work we have chosen four distinct features

- Title word count : It is the length of news title . Fake news usually have long title to attract readers. So we chose this feature.
- Text word count : It is the length of news content . Fake news generally have less content than proper sourced real news. Unwanted capitalization: It is the count of whole capital words in title of a news . Fake news creators use more capitalization in title to attract readers.
- Number Presence : It counts the occurrence of numerical values in a news title/text.

With the help of feature union in python pipeline these features are appended along with TF-IDF vectorized values and used for training.

Chapter 4

IMPLEMENTATION AND TESTING

4.1 Implementation of FakeNewsDetection

Installation
Since we are building the library FakeNewsDetection , so it can not be used alone but only by incorporating it into other programs. To compile the program some dependency libraries like nltk , sklearn ,pandas, pickle etc are needed. These libraries can be installed during compile time using "pip install lib-name " command .
FakeNewsDetection is a free and open source library which can be easily integrated. Listing 4.1 shows the command for installing the library to respective systems. By installing this packages several other modules also installed with this. Our package is programmed in python3. It implements the idea for checking the reliability of a news.

Listing 4.1: Installing FakeNewsDetection

gitclone :`https://gitlab.com/DesignProjectLab49/2020-CSB14-FAKEDATADETECTIONFROMPANDeMICDATA.git`

python m pip i n s t a l l 2020-CSB14-FAKEDATADETECTIONFROMPANDEMICDATA

Implementation

• **train()**:Used in main.py to load the dataset, load the training model , after training trained model is saved.
Keyword arguments :
1. model = "LR" : model argument determines the classifier model to be loaded and used for training.

• **get-train-data()**: Load the dataset and split into train and test data.
Arguments :
1. dataset: type dataframe ,data to be used for train and test

• **predict()**: used in prediction.py . Accepts title and text as input to predict the nature of news.
Arguments:
1. title,text= data type string. New user input to check a news.

•**compare()**: load different classifier models and pass their accuracy,confusion matrix and classification report into a dictionary.

Arguments:
1. force: Boolean data type argument . If tree it trains all models from the beginning. Else it will load the saved model.

•**getAccuracy ()**: return accuracy of trained models
Arguments : None
• **getConfmatrix**: return confusion matrix of trained models
Arguments : None
• **getReport()**: return classification report of models.
Arguments : None

4.2 Testing

1. Unit Test

A unit test is the micro level of testing,that checks if every single component operates in the right way. A unit test helps you to isolate what is broken function in your application and fix the bugs found in them.They form atomic segments of the code,which provides quick results and only if the unit tests are cleared the code needs to be merged.Unit test are used to predict the bugs early in the program under development.Unit testing helps in verifying internal design internal logic and internal parts of a program,it also helps in error handling.
-In our project unit tests have been done on many user defined functions like feature extraction and preprocessing.
Unit testing was done onthese functions with a sample data which were extracted from the actual dataset to perform testing on the created functions.
1. System Testing

System testing is testing of a complete integrated software system .System testing takes,all of the integrated components that have passed integration testing as its input.The system testing includes the testing of fully integrated application in order to check how the components interact with each other when working as a system and verification of every input to check for the desired output.

4.3 Logging

Logging file is saved as app.log. We have used 3 levels of logging:

- INFO: Confirmation that things are working as expected
  - A logger info is provided at the end of every major user defined function to understand it is functioning properly.
- WARNING(): An indication that something unexpected happened, or indicative of some problem in the near future. The software is still working as expected.
  - used in the code segment where pre trained ml model is loaded. Returns a warning and trains the model from the beginning.

- ERROR(): Due to a more serious problem, the software has not been able to perform some function.
  - - logger error is provided at the code segment where dataset is loaded ,returns error if the csv file is not present in the directory.
  - - train the model before prediction: when trying to predict nature of a news before training the model.

Chapter 5

RESULTS & ANALYSIS

The aim of performance evaluation is to study and analyse the performance of the classifier in correctly classifying the instances by using the evaluation metrics such as accuracy matrix. In this project, we have used a total of seven different machine learning algorithm in order to know which algorithm is more successful in predicting whether the news is fake or not.
In the initial phase, 3 machine learning algorithms namely logistic regression, passive aggressive classifier, and decision tree were assigned with the problem statement. All the algorithms performed well with the given dataset and however logistic regression had the highest accuracy among them.
In the later phase 4 more classifiers were used (Linear svm, poly svm, random forest, rbf svm) and then also the accuracy of logistic regression was comparatively high.

5.1 Comparison With Existing System

A table comparing the details of our sytem and the currently existing system is given below

Existing System	Our package :"Fake News Detection"
No package implementation - not easy to install ,use and modify	Package implementation of training and prediction - Easy to install ,use amd modify - Web and app implementation made easy
Vectorized features are only used for training	Manually extracted text features are added with vectorized features to increase accuracy of the model
Normal lemmatization and preprocessing	Pos tag Lemmatization is used to increase the efficiency of preprocessing and vectorization -POS tagging is a supervised learning solution that uses features like the previous word, next word, is first letter capitalized etc. NLTK has a function to get pos tags and it works after tokenization process
Only few classification algorithms applied	Compared different classification algorithms like Logistic Regression, PAC, Decision Tree ,Random Forest ,Linear SVM, RBF SVM, Poly SVM and chose the one with best accuracy for training

Table 5.1: Comparison With New Existing

Classifier	Existing	New
LR	91.8	**93.6**
DT	84.5	64.5
PAC	92.0	89.1
LIN SVM	90.0	90.0
POLY SVM	63.6	59.09
RBF SVM	91.8	91.8
Random Forest	87.3	87.3

Table 5.2: Comparison Table Phase 2

Chapter 6

CONCLUSION

In this project, seven distinct machine learning algorithms were implemented in total as a part of research and studies to predict fake news from real news. The model performs a prediction on the input text and title of a news to predict whether the given news is fake one or a real one. The experiments were conducted effectively on the test dataset, and the comparison was performed on existing methods. The model was also implemented as a web UI and also as a Package. The package implementation allows any user to use our model for the prediction of a news. Based on the results, it can be concluded that the proposed scheme is efficient in utilising the data provided to make accurate and desired predictions to a greater extend

Pros and Cons

With the addition of new text features and using pos tag lemmatisation instead of normal one we were able to achieve a fair accuracy of 93.6 . The model that has been trained and tested can be saved as a model(.mdl) file to use it anytime later that ensures reusability of the system. The design of the project is user-friendly as it provides a better user interface to perform prediction. The user can experience the quality of the system from the front end without caring much about its functionality. The use of UI for comparing different models also enables anyone who does not know much about coding understand about the difference in accuracies of each models. The package implementation of the model enables any user to import our project and use it according to their need.
Our proposed system only predicts using the textual features of the news, as there many other features which we are not considering in a news like pictures etc which makes it harder to differentiate between fake one and real one. Our training model largely depends on the dataset we use to train.

Future Scope of the Project

The scope of this project can be extended by identifying more features and adding them to the currently created dataset. The research can be extended towards exploring features other than just textual features like source and propagation style of news .

The experiment can be repeated by using other machine learning algorithms as well as Neural networks . Parameter tuning and ensamble methods can be utilized to increase the efficiency of our model.
Since there are many media contents like audio, video and images associated with the propagation of fake news they can also be considered at the time of fact

checking.
Since we have implemented our project as a package we hope that it will be easy to make modifications and improvements in the future.

Bibliography

[1] A. D. Holan, 2016 Lie of the Year: Fake News, Politifact, Washington, DC, USA, 2016.

[2] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, “Fake news detection on social media,” ACM SIGKDD Explorations Newsletter, vol. 19, no. 1, pp. 22–36, 2017.
https://dl.acm.org/doi/10.1145/3137597.3137600 .

[3] S. Vosoughi, D. Roy, and S. Aral, “The spread of true and false news online,” Science, vol. 359, no. 6380, pp. 1146–1151, 2018.
https://science.sciencemag.org/content/359/6380/1146 .

[4] H. Allcott and M. Gentzkow, “Social media and fake news in the 2016 election,” Journal of Economic Perspectives, vol. 31, no. 2, pp. 211–236, 2017.
https://www.aeaweb.org/articles?id=10.1257/jep.31.2.211 .

[5] BY DR. VAIBHAV KUMAR 22/06/2020 , Hands-On Guide to Predict Fake News
https://analyticsindiamag.com/hands-on-guide-to-predictfake-news-using-logistic-regression-svm-and-naive-bayesmethods/.

[6] H. Ahmed, I. Traore, and S. Saad, “Detection of online fake news using n-gram analysis and machine learning techniques,” in Proceedings of the International Conference on Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments, pp. 127–138, Springer, Vancouver, Canada, 2017.
https://link.springer.com/chapter/10.10072F978-3-319-69155-8_9

[7] ML Algorithms addendum: Passive Aggressive Algorithms
https://www.bonaccorso.eu/2017/10/06/ml-algorithms-addendum-passive-aggressive-algorithms/ .

[8] Feb 9 Prince Yadav , Decision Tree in Machine Learning, Towardsdatascience .
https://towardsdatascience.com/decision-tree-in-machinelearning- .

[9] Jaime Zornoza, Sigmoid function ,towardsdatascience, Feb 9 2020.
https://towardsdatascience.com/logistic-regressionexplained-9ee73cede081 .

[10] Saimadhu Polamuri, Dataaspirant, HOW THE LOGISTICREGRESSION MODEL WORKS; March 2, 2017
https://dataaspirant.com/how-logistic-regression-modelworks .

[11] Passive Aggressive Classifiers 17-07-2020
https://www.geeksforgeeks.org/passive-aggressive-classifiers/ .

[12] SHUBHAM SINGH, JULY 18, 2019 What is Tokenization in NLP
https://www.analyticsvidhya.com/blog/2019/07/how-get-started-nlp-6-unique-ways-perform-tokenization/ .

[13] Python - Remove Stopwords,
https://www.tutorialspoint.com/python$_{t}ext_{p}rocessing/python_{r}emove_{s}topwords.htm.$

[14] Prateek Sawhney Sep 3, Introduction to Stemming and Lemmatization (NLP)
https://medium.com/swlh/introduction-to-stemming-vs-lemmatization-nlp-8c69eb43ecfe .

[15] Cory Maklin, May 5, 2019 , TF IDF — TFIDF Python Example,
https://towardsdatascience.com/natural-language-processing-feature-engineering-using-tf-idf-e8b9d00e7e76 .

[16] Dataset : Susan Li ,Jul 22
https://towardsdatascience.com/covid-fake-news-detection-with-a-very-simple-logistic-regression-34c63502e33b .

Appendices

Appendix A

Sample Code

Main.py

```
import pickle as pk
import pandas
from .detect import Detect
from .prediction import Prediction
import logging
class fake_news_detection:
    def _init_(self,filename):
        logging.basicConfig(level=logging.DEBUG, filename='app.log', file
                            format='%(name)s␣-␣%(levelname)s␣-␣%(message
        self.filename = filename
        self.data = self.loadModel()
        self.tr = Detect(self.data)

    def loadModel(self):
        try :
            data = pandas.read_csv(self.filename)
            logging.info("csv␣file␣found")
        except :
            logging.error("file␣not␣found")
            exit(0)
        return  data
    def train(self, model = "LR"):
        self.tr.train(model)
    def predict(self,text,title):
        return Prediction().predict(text , title)
    def compare(self,force =False):
        if(not force):
            try :
                data = pk.load(open("compare_model.sav" , "rb"))
            except IOError:
                logging.warning("file␣not␣found,␣traning␣all␣model")
               m = self.tr.compare()
                data = {'accuracy': [self.tr.getAccuracy(model) for _, m
                        'cmatrix': [self.tr.getConfmatrix(model) for _,
                        'creport': [self.tr.getReport(model) for _, mode

                pk.dump(data, open("compare_model.sav", 'wb'))
        else:
                m= self.tr.compare()
                data= {'accuracy':[self.tr.getAccuracy(model) for _ , mo
                    'cmatrix':[self.tr.getConfmatrix(model) for _ , mode
                    'creport':[self.tr.getReport(model) for _, model in r

                pk.dump(data, open("compare_model.sav", 'wb'))
        return data
```

Test.py

```
from FakeNewsDetection import fake_news_detection
if _name_ == '_main_':
    fk = fake_news_detection("data/file1.csv")
    data = fk.compare()
    a=data['accuracy']
    b = data['cmatrix']
    c =data['creport']
    print("Accuracy␣of␣LR␣:",a[0])
    print("Confusion␣Matrix␣of␣LR␣:\n",b[0])
    print("classification␣report␣LR␣:\n",c[0])
    print("Accuracy␣of␣PAC␣:", a[1])
    print("Confusion␣Matrix␣of␣PAC␣:\n", b[1])
    print("classification␣report␣PAC␣:\n", c[1])
    print("Accuracy␣of␣SVM␣:", a[2])
    print("Confusion␣Matrix␣of␣SVM␣:\n", b[2])
    print("classification␣report␣SVM␣:\n", c[2])
    print("Accuracy␣of␣Poly_SVM␣:", a[3])
    print("Confusion␣Matrix␣of␣Poly_SVM␣:\n", b[3])
    print("classification␣report␣Poly_SVM␣:\n", c[3])
    print("Accuracy␣of␣RBF_SVM␣:", a[4])
    print("Confusion␣Matrix␣of␣RBF_SVM␣:\n", b[4])
    print("classification␣report␣RBF_SVM␣:\n", c[4])
    print("Accuracy␣of␣RF␣:", a[5])
    print("Confusion␣Matrix␣of␣RF␣:\n", b[5])
    print("classification␣report␣RF␣:\n", c[5])
    print("Accuracy␣of␣Decision␣Tree␣:", a[6])
    print("Confusion␣Matrix␣of␣Decision␣Tree␣:\n", b[6])
    print("classification␣report␣Decision␣Tree␣:\n", c[6])
```

Prediction.py

```
import pickle
from .detect import Detect
import pandas
class Prediction:
    def _init_(self):
       pass
    def predict(self,text , title):
        df = pandas.DataFrame({
```

```
                                              'text' :[text],
                                              'title' : [title]})
        log = self.load_file('fake_news_model.sav')
        result=log.predict(df)
        return result[0]
    def load_file(self,filename):
        fake_news_model = open(filename, 'rb')
        p = pickle.load(fake_news_model)
        return p
```

newinput.py

```
from FakeNewsDetection import fake_news_detection
if __name__ == '__main__':
    fk = fake_news_detection("corona_fake_news.csv")
    title = input("Enter title : ")
    text = input("Enter text : ")
    print(fk.predict(text, title))
```

Appendix B

Screenshots

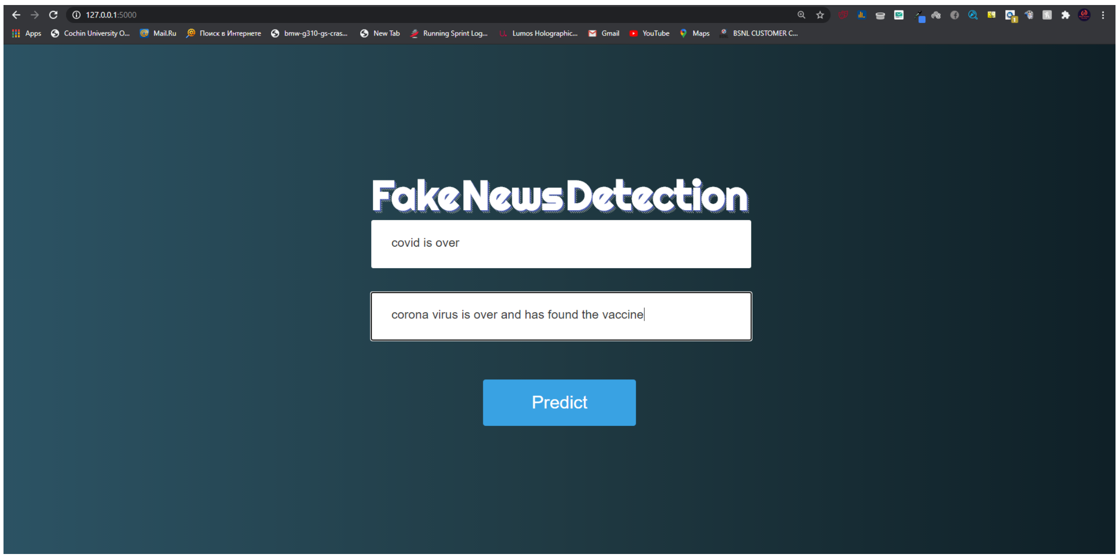


Figure B.1: Web ui

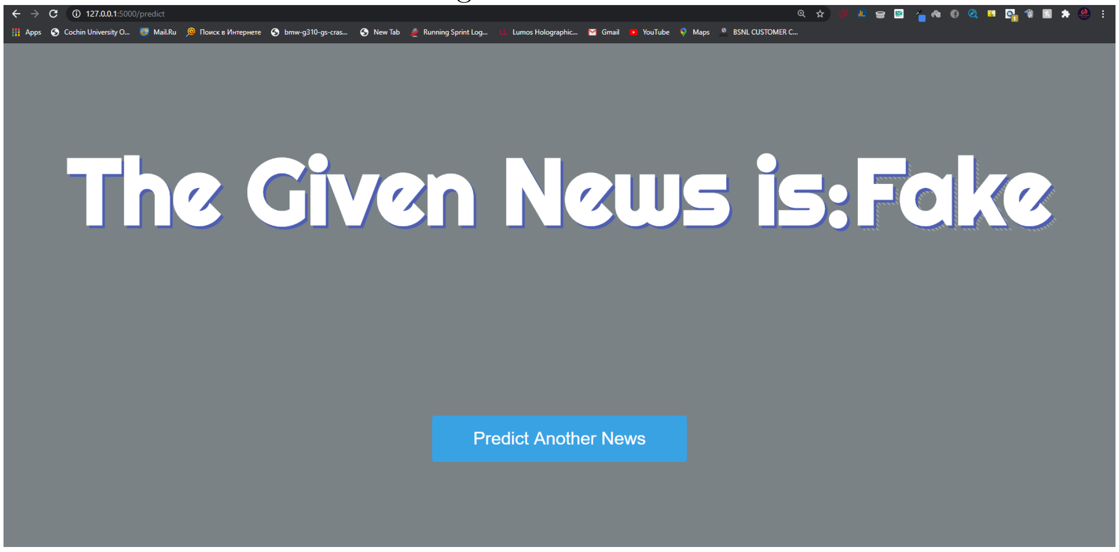
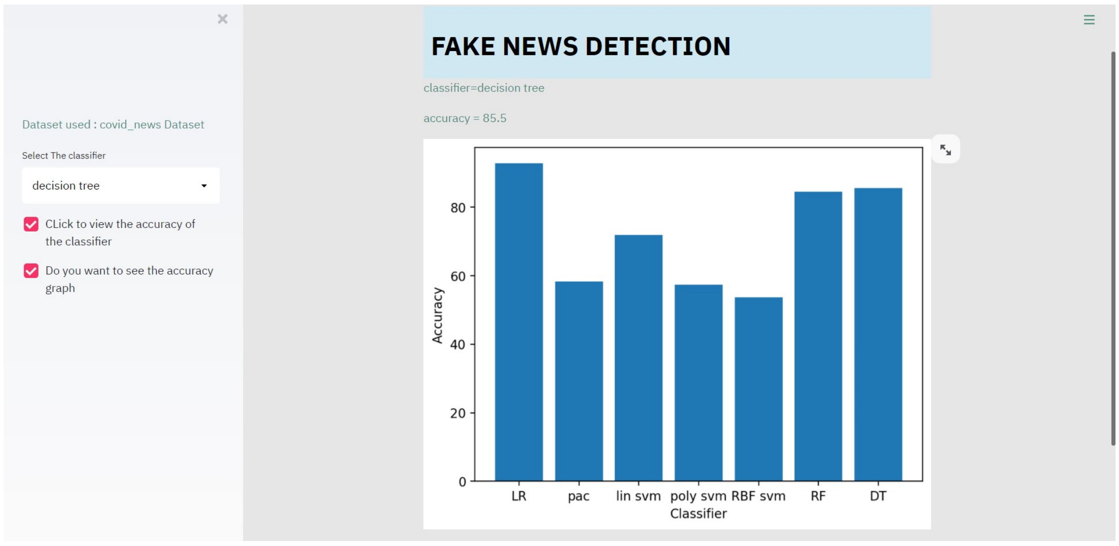


Figure B.2: Web ui



Dataset used : covid_news Dataset

Select The classifier

decision tree

☑ CLick to view the accuracy of the classifier

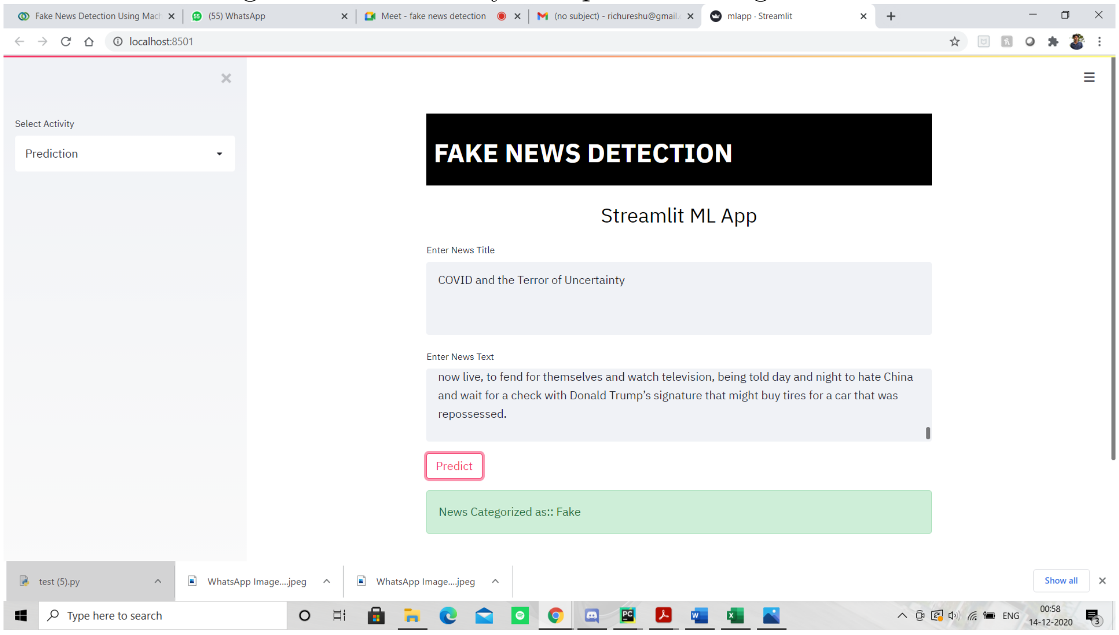
☑ Do you want to see the accuracy graph

FAKE NEWS DETECTION

classifier=decision tree

accuracy = 85.5

Figure B.3: Accuracy Comparison Using Streamlit



Fake News Detection Using Mac
(55) WhatsApp
Meet - fake news detection
(no subject) - richureshu@gmail.
mlapp · Streamlit
localhost:8501

Select Activity

Prediction

FAKE NEWS DETECTION

Streamlit ML App

Enter News Title

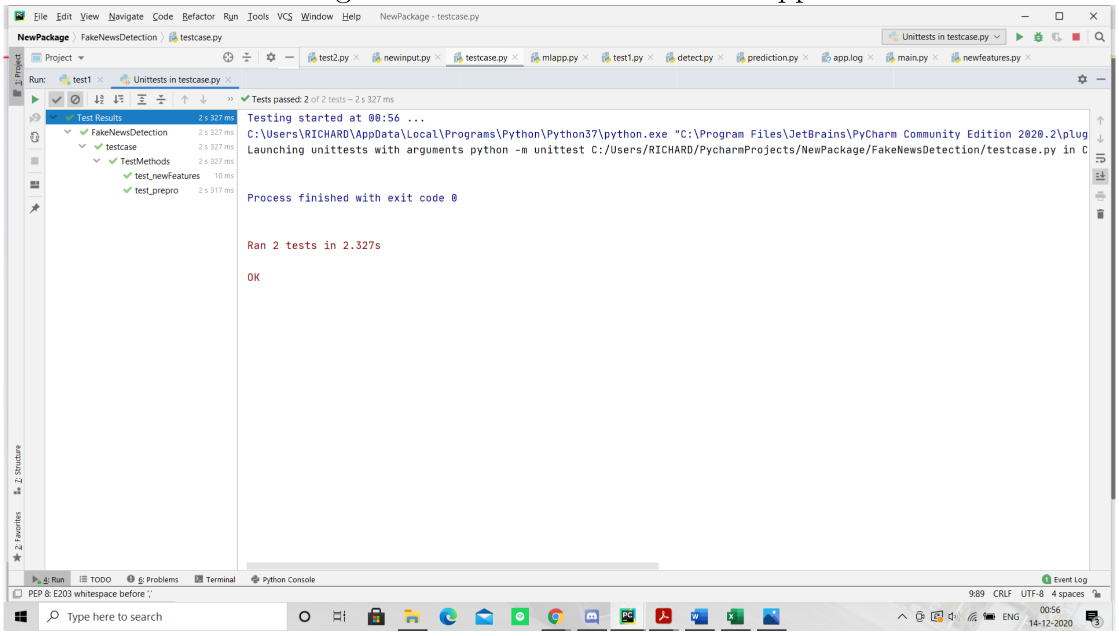
COVID and the Terror of Uncertainty

Enter News Text

now live, to fend for themselves and watch television, being told day and night to hate China and wait for a check with Donald Trump's signature that might buy tires for a car that was repossessed.

News Categorized as:: Fake

Figure B.4: Streamlit Prediction app



File Edit View Navigate Code Refactor Run Tools VCS Window Help NewPackage - testcase.py
NewPackage FakeNewsDetection testcase.py
Unittests in testcase.py
Project
test2.py
newinput.py
testcase.py
mlapp.py
test1.py
detect.py
prediction.py
app.log
main.py
newfeatures.py
Run: test1 Unittests in testcase.py

Tests passed: 2 of 2 tests – 2 s 327 ms

```
Test Results          2 s 327 ms
  FakeNewsDetection   2 s 327 ms
    testcase          2 s 327 ms
      TestMethods     2 s 327 ms
        test_newFeatures   10 ms
        test_prepro   2 s 317 ms
```

```
Testing started at 00:56 ...
C:\Users\RICHARD\AppData\Local\Programs\Python\Python37\python.exe "C:\Program Files\JetBrains\PyCharm Community Edition 2020.2\plug
Launching unittests with arguments python -m unittest C:/Users/RICHARD/PycharmProjects/NewPackage/FakeNewsDetection/testcase.py in C


Process finished with exit code 0


Ran 2 tests in 2.327s

OK
```

Figure B.5: Unit Testing

Appendix C

Datasets

A Covid-19 News dataset composed of about 1200 Covid-19 News data out of which nearly half of them is labelled as fake . It was found from the internet[8]. The dataset is given here :
`https://drive.google.com/drive/folders/1cp62MA3ZwafUWgMukrFushdJ9lXkQ6e5?usp=sharing`