

A Project Report on

A Web Framework to Predict Fake News Using ML

Submitted in partial fulfillment of the requirements for the award
of the degree of

Bachelor of Engineering

in

Information Technology

by

Jaagrut Shah(19204002)

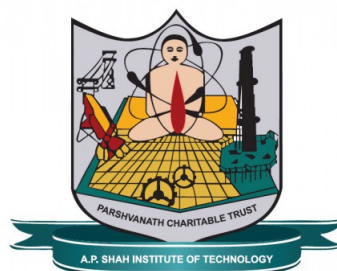
Jigar Desai(19204003)

Yash Jain(19204013)

Under the Guidance of

Prof. Apeksha Mohite

Prof. Geetanjali Kalme



Department of Information Technology

NBA Accredited

A.P. Shah Institute of Technology

G.B.Road,Kasarvadavli, Thane(W), Mumbai-400615

UNIVERSITY OF MUMBAI

Academic Year 2021-2022

Approval Sheet

This Project Report entitled "*A Web Framework to Predict Fake News Using ML*" Submitted by "*Jaagrut Shah*"(19204002), "*Jigar Desai*"(19204003), "*Yash Jain*"(19204013) is approved for the partial fulfillment of the requirement for the award of the degree of *Bachelor of Engineering* in *Information Technology* from *University of Mumbai*.

Prof. Geetanjali Kalme
Co-Guide

Prof. Apeksha Mohite
Guide

Prof. Kiran Deshpande
Head Department of Information Technology

Place:A.P.Shah Institute of Technology, Thane

Date:

CERTIFICATE

This is to certify that the project entitled "***A Web Framework to Predict Fake News Using ML***" submitted by "***Jaagrut Shah***"(19204002), "***Jigar Desai***"(19204003), "***Yash Jain***"(19204013) for the partial fulfillment of the requirement for award of a degree ***Bachelor of Engineering in Information Technology***, to the University of Mumbai, is a bonafide work carried out during academic year 2021-2022.

Prof. Geetanjali Kalme
Co-Guide

Prof. Apeksha Mohite
Guide

Prof. Kiran Deshpande
Head Department of Information Technology

Dr. Uttam D.Kolekar
Principal

External Examiner(s)

1.

2.

Place: A.P. Shah Institute of Technology, Thane

Date:

Acknowledgement

We have great pleasure in presenting the report on **A Web Framework to Predict Fake News Using ML**. We take this opportunity to express our sincere thanks towards our guide **Prof. Apeksha Mohite** & Co-Guide **Prof. Geetanjali Kalme** Department of IT, APSIT thane for providing the technical guidelines and suggestions regarding line of work. We would like to express our gratitude towards his constant encouragement, support and guidance through the development of project.

We thank **Prof. Kiran B. Deshpande** Head of Department, IT, APSIT for his encouragement during progress meeting and providing guidelines to write this report.

We thank **Prof. Vishal S. Badgujar** BE project co-ordinator, Department of IT, APSIT for being encouraging throughout the course and for guidance.

We also thank the entire staff of APSIT for their invaluable help rendered during the course of this work. We wish to express our deep gratitude towards all our colleagues of APSIT for their encouragement.

Student Name1: Jaagrut Shah
Student ID1: 19204002

Student Name2: Jigar Desai
Student ID2: 19204003

Student Name3: Yash Jain
Student ID3: 19204013

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, We have adequately cited and referenced the original sources. We also declare that We have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We 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.

(Jaagrut Shah 19204002)

(Jigar Desai 19204003)

(Yash Jain 19204013)

Date:

Abstract

Fake news has been a drag ever since the web boomed. The news or social media networks that allow us to gather information about incidents happening over this world can be contaminated with fake news to run a particular political or personal agenda. Combating this fake news is vital because the world's view is formed by information. People not only make important decisions supported by information but also form their own opinions. If this information is fake, it can have devastating consequences. Verifying each news one by one a person is unfeasible. This study proposes a system that will attempt to consistently categorize and forecast whether a news item is false or not, to speed up the process of identifying fake news. On data-sets obtained from diverse sources such as Kaggle, machine learning techniques such as Naive Bayes, Logistic Regression, Random Forest, and SVM Classifier have been attempted and tested. The analysis and findings of various models are also included in the publication. The difficult process of detecting fake news is often made simple by combining the right models with the right instruments. In recent years, the World Wide Web (WWW) has grown into a massive repository of user-generated material and opinionated data. Users can easily share their thoughts and sentiments by using social media platforms such as Twitter, Facebook, and Whats-App. Twitter, Facebook, and other social media sites are examples of this. where millions of people express their opinions and sentiments regarding various topics in their everyday interactions. These ever-increasing subjective data are, without a doubt, an especially rich source of data for any careful decision-making process. Sentiment Analysis has emerged as a way to automate the analysis of such data. Its goal is to find opinionated material on the Internet and classify it according to its polarity, whether it has a positive or bad connotation, or if it tends to be a neutral statement in this world of opinions. Although sentiment analysis is a text-based problem, various problems make it more complex than typical text-based analysis. This implies that any attempt to solve these challenges is required, and it has opened up various avenues for future research into negations, concealed feelings identification, slang, and polysemy. However, the expanding scope of knowledge necessitates the use of automated data analysis techniques such as Natural Language Processing to analyze text. An in-depth study of several methodologies used in Fake News Prediction & Sentiment Analysis is conducted in this research to determine the scope of work.

Contents

1	Introduction	1
2	Literature Review	3
3	Objectives	6
4	Project Design	7
4.1	UML Diagrams	8
4.2	Motivation	10
4.3	Model Used	11
4.4	MODEL ANALYSIS TERMS	12
4.5	Existing System Architecture	14
4.6	Data:	15
4.7	Model:	15
5	Project Implementation	17
6	Testing	21
6.1	Unit Testing	21
6.2	Integration Testing	21
7	Result	22
8	Conclusions and Future Scope	26
	Bibliography	26
	Appendix	29
	Publication	30

List of Figures

4.1	Flow Diagram	8
4.2	Use-case Diagram	9
4.3	Block Diagram	9
4.4	Activity Diagram	10
4.5	Multinomial NB count confusion matrix	13
4.6	Confusion Matrix for Logistic Regression	13
4.7	Confusion matrix for SVC	14
4.8	operation of how the existing model works	16
5.1	Index.html	17
5.2	App.py	18
5.3	Result.html	19
5.4	TwitterApi.py	20
7.1	Getting News from online sources	22
7.2	Predicted Results	23
7.3	Updated Data set after Each News Added	24
7.4	Twitter sentiment analysis	25

List of Abbreviations

ML: Machine Learning
PDF: Portable Document Format
DF: Data Frame
RE: Regular Expression
TF: Term Frequency
IDF: Inverse Document Frequency
LR: Logistic Regression
NB: Naïve Bayes
RF: Random Forest
SVC: Support Vector Classifier
CSV: Comma Separated Value file

Chapter 1

Introduction

The term "fake news" was relatively unknown and unpopular a few decades ago. However, in the digital age of social media, it has evolved into a gigantic monster. Fake news, information bubbles, news manipulation, and, consequently, a lack of faith in the media are all major issues in our society. To begin tackling this issue, however, it's critical to have a solid understanding of fake news and the sources behind it, only then can one consider the different Machine Learning (ML), Natural Language Processing (NLP), as well as AI (Artificial Intelligence) approaches and domains that might be effective in tackling this challenge. Within the last six months, the term "fake news" has been used in a variety of ways, with numerous definitions presented. The New York Times, for example, defines it as "a fabricated fiction intended to deceive." Measuring or even precisely classifying false news can probably turn into a subjective rather than objective process. Fake news, in its most basic form, is entirely made up and modified to look like legitimate journalism to gain the greatest exposure and, with it, advertising revenue. Despite these flaws, several organizations have attempted to define fake news in various ways.

Sentiment analysis, also identified as a point of view mining, is a technique for automatically detecting sentiments expressed in text that is becoming a challenge in many research areas, particularly in the data mining field for social media, with applications ranging from product ratings and feedback analysis to customer decision making. In recent years, social networking sites to gather and share public opinion have become even more important. With the rapid growth of Web 2.0, a growing number of individuals prefer to express themselves online., opinions, and approaches over the Internet, resulting in a massive supply of user-generated content and opinionated data., opinions, and approaches over the Internet, resulting in a massive supply of user-generated content and opinionated data. today we are living in the world which is surrounded by 99 percentage of data. There are different micro-blogging sites where users express their views about different products these views are nothing but opinions of people and it will go waste if it is not used in proper way so there is a need to use opinions of people in improving productivity, usefulness, functionality of particular product or application or technique or any entertainment resource. Hence, there is a need to develop a product which can analyse opinions of people. This product will be useful in increasing market value of industries as well as satisfy needs of customers.

The process of computationally identifying and categorizing opinions expressed in a piece of text, especially in order to determine the writer's attitude towards a particular topic or product. Sentiment Analysis is the process of detecting the contextual polarity of text. In other words, it

determines whether a piece of writing is positive, negative or neutral.

Twitter, one of the largest social media site receives tweets in millions every day in the range of Zettabyte per year. This huge amount of raw data can be used for industrial or business purpose by organizing according to our requirement and processing.

In this project, we are going to implement a system in Hadoop which analyses twitter data where cluster of nodes will be formed. Twitter data is in the form of comments which are nothing but sentiments that is opinions, feelings of people. This data will be collected by using Twitter API. By analysing this data, our system will give output in the form of positive, negative and neutral tweets. In this case, it makes the use of data dictionary for classifying the data. This data can be used further according to particular application. And this analysed data can be represented in the form of pie-charts.

Chapter 2

Literature Review

Many strategies for automatically detecting fake news and deception posts have been reported in the literature. There are multidimensional aspects identification of bogus news ranging from using chatbots for the spread of misinformation to the use of clickbait for the spreading. There are many clickbait's available in social media networks including Facebook which enhance sharing and liking of posts which in turn spreads falsified information. There has been a lot of effort put towards detecting fake data.

In this section, we'll talk about some of the papers that used machine learning to identify and classify fake news. Title: "A robust technique of fake news detection using Ensemble Voting Classifier and comparison with other classifiers", Springer nature 2019. Author: Atik Mahabub Findings: Atik[1] uses a distinctive technique in detecting fake news by creating an 'Ensemble Voting classifier'. It uses many well-known Machine-Learning classifiers such as Naïve Bayes, KNN, SVM, and many more to verify the news. Further, cross-validation was used and the top three machine learning algorithms with the best accuracy were used in the Ensemble Voting Classifier. This model proposed a recognition structure that can productively predict the output and find the important highlights of the news. This allowed for a result ranging from the early to late 90s. Title: "Text mining-based Fake News Detection Using Ensemble Methods", International Journal of Automation and Computing. Author: Harita Reddy, Namratha Raj, Manali Gala, Annappa Basava. Findings: Text-mining-based methods for the False news detection have been evaluated by Harita Reddy et al [2]. This paper provided a hybrid approach that combines word vector representations and stylometric features using ensemble methods like bagging, boosting, and voting. After the selection of important features, Random Forest, Nave Bayes, SVM, and other algorithms were used after the key features were chosen. This resulted in accuracies up to 95.49 percentage. "Fake News Detection Using Machine Learning and Natural Language Processing," ISSN: 2277-3878, Volume-7, Issue-6, International Journal of Recent Technology and Engineering (IJRTE). Author: Kushal Agarwalla, Shubham Nandan, Varun Anil Nair, D. Deva Hema Findings: The Natural Language Processing technique was exploited by Kushal Agarwalla et al [3]. to verify the news. NLTK from Python was used with various models including Logistic Regression, SVM, and Naïve Bayes with Lid stone Smoothing. Naive Bayes with 3 Lid stone smoothing performed admirably and gave a result of 83 percentage. Using only vector-based methods to extract certain features and train classifiers may not be the most accurate answer because these are specific to the dataset. 2017 IEEE First Ukraine Conference on Electrical and Computer Engineering, titled: "Fake News Detection Using Nave Bayes Classifier." Author: Mykhailo Granik, Volodymyr Mesyura Findings: Mykhailo Granik et al [4].

implemented a basic approach using For the detection of bogus news, the Naive Bayes classifier is used. The model was developed as a software system and put to the test. using Facebook news posts. It compares spam communications and false news items and concludes that same methodologies may be used for both fake news detection and spam filtering, yielding a 75 percent accuracy result. 2018 IEEE International Students' Conference on Electrical, Electronics, and Computer Sciences, "Fake News Detection." Authors: Akshay Jain, Amey Kasbe Findings: Akshay Jain et al [5]. proposed a model with two variants It uses the Naive Bayes classifier to predict if a Facebook post will be tagged as FAKE or REAL The first model used the title as their source for vocabulary building, using a count vectorizer. And the second model used text as its source. The AUC scores of the two models were compared, and the second model was found to be superior, with scores of 0.93 and 0.912 with and without n grams, respectively. SMU Data Science Review, Vol. 1: No. 3, Article 10, "Fake News Detection: A Deep Learning Approach." Aswini Thota, Priyanka Tilak, Simrat Ahluwalia, and Nibrat Lohia are the authors (2018) Findings: To detect bogus news, Aswini Thota et al [6] used Deep Learning architectures. TfIDF, GloVe, and Word2Vec were used along with the DNN model to precisely predict the stance between the article body and given pair of the headline. This paper achieved an overall accuracy rate of 94.31 percent. Stanford University CS 224N - Winter 2017. Title: "The Pope Has a New Baby! Fake News Detection Using Deep Learning". 4 Authors: Samir Bajaj Findings: Samir [7] explored different models ranging from Logistic Regression to CNN, RNN, and GRU. This work is mainly concentrated on using pure NLP perspective to identify the presence of fake news by utilizing linguistic features. The highest precision of 0.97 was obtained using CNN with Max Pooling and Attention. As fake news becomes more adept at duplicating real news, this technique may become obsolete. Volume 8, Issue 10, August 2019 of the International Journal of Innovative Technology and Exploring Engineering. "An Efficient Fake News Detection System Using Machine Learning," according to the title. A. Lakshmanarao, Y. Swathi, and T. Srinivasa Ravi Kiran are the authors." A. Lakshmanarao and colleagues [8]. SVM, Knn, Decision tree, and Random Forest were used to create four models, which were then compared. It was observed that Random Forest Classification gave the highest score of 90.7 percentage while least was provided by Support Vector Machines at 75.5 percentage. "Evaluating Machine Learning Algorithms for Fake News Detection," IEEE Student Conference on Research and Development, IEEE Student Conference on Research and Development, IEEE Shlok Gilda et al [9] are the authors, and Shlok Gilda et al are the findings. To identify bogus news, the team relied solely on Natural Language Processing. Probabilistic context-free grammar (PCFG) and Term frequency inverse document frequency (TF-IDF) of bigrams were applied with various models like gradient boosting and stochastic gradient descent. Among other models, TFIDF of bi-grams with stochastic gradient descent identified fake news with higher accuracy. Title: Association for Computational Linguistics, August 2018. "Automatic Detection of Fake News." Veronica Perez-Rosas, Bennett Kleinberg, Alexandra Lefevre, Rada Mihalcea, Veronica Perez-Rosas, Bennett Kleinberg, Alexandra Lefevre, Rada Mihalcea, Veronica Perez-Rosas, Bennett Klein Findings: While the previous publications used machine learning to detect bogus news, Veronica [10] adds a new dimension to the equation by incorporating human testing. This is the only study that has tallied and compared human and machine performance. This study used two data-sets: Fake News AMT, which comprises general news, and Celebrity News, which, as the name implies, contains celebrity news. The paper utilised two annotators to determine if the news was real or false, and they had a 70 percentage agreement rate. Annotators have been found to outperform the automated system.

Quick technological advancement have authorized news papers and journalism to be distributed over the web and the rise of Twitter, YouTube, Instagram, Facebook and some other social networking sites. Networking Sites have become a noteworthy method to speak for people with each other and offer schemes and thoughts. Critical components of a person these networking sites is quick sharing of information. Specifically in this situation, exactness of the news or information distributed is critical. Fake news spreading on different networking sites has become the most concerning issue. Fake news has majorly influenced everyday lives and the social requests of many individuals and caused some negative impacts. Here, the most thorough electronic databases have been broken down to take a greater look at articles about identification of news that is fake on networking sites using an efficient practice of literature review. The fundamental point to study this is revealing the advantages that AI uses for the knowledge about fake news and its victory in one application or the other. Accordingly, assumptions were made that the victory of computerized reasoning gadgets is more than 90 percentage. This is accepted to be a manual for anyone related to this field(researchers and individuals).

Chapter 3

Objectives

- To show the news relevancy and analysis to attain accuracy in anticipating real and dependable news.
- To work on this issue, before trying a prediction, a layered model is provided, each time the data is analyzed, it gets better and better at giving you information that helps you understand it.
- To demonstrate demonstrable effectiveness in predicting false news and social media postings using several Machine Learning methods.
- To halt the propagation of false material on social media platforms that may cause user confusion.
- To be able to give more and more accurate news on the screen.

Chapter 4

Project Design

The model to be made for predicting the relevance of a News can be approached through the proposed steps to be followed. They are listed below as:

- Collecting Data from data-sets
- Data cleaning and Pre-processing
- Feature Extraction using TFIDF
- Providing data to Models
- Result Analysis

The Model is decided based on the best accuracy provided and the user-supplied data is fed into the model, which predicts the outcome. Also, the Sentiments are similarly Neutral, Positive, and Negative, are the three categories. The analysis of sentiments depends upon the:

Methods Of NLP:

Like the library called notch, is used for this process to make it gives a great amount of accuracy.

Similarly, the prediction of Fake or Not depends upon the features extracted (i.e., no. of words along with their inverse frequency-converted to numeric using vectorization) using TFIDF from the combined text total obtained by either user of Twitter API. The classification here depends on:

- Features present in Text
- Probability calculation

4.1 UML Diagrams

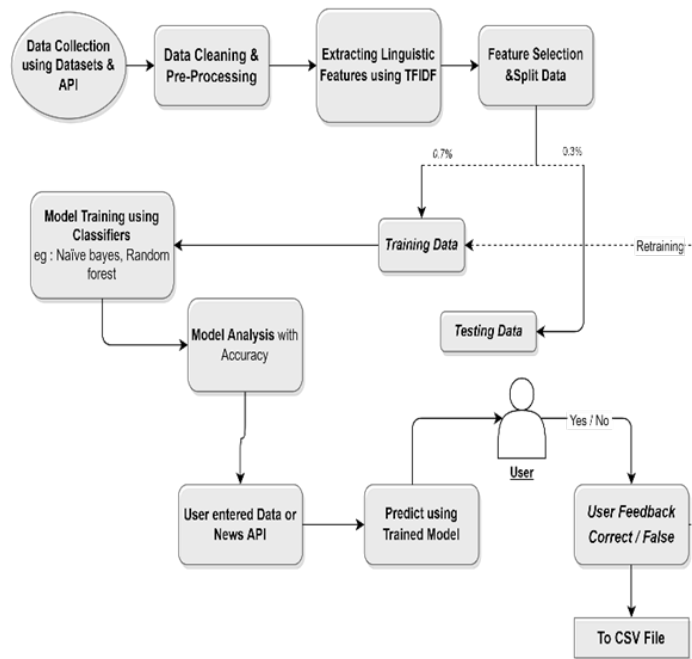


Figure 4.1: Flow Diagram

Figure 4.1 is our flow diagram that shows how the processes in our system work and how the whole output process is passed through various processes in order to get the desired results. It starts with the first step of data collection. This is the basic step, as without it it is not possible to do anything. We first take out data from kaggle, where we find a data-set of truly predicted and falsely predicted news. We gather the data-set together and perform cleaning, as it is mentioned there, data preprocessing. After data preprocessing, the textual new is converted into numeric ones by adding weights to the words, letting the model know the importance of a word in a text. Then it is followed by training and testing, which is a required phase of machine learning model building and evaluation. After this, the model we developed, the Multinomial Naive Bayes, is used to predict the text coming from the form on the web-page or the text obtained from the Twitter API. This is then shown, or said, displayed to the user via a web-page. And after waiting for the user's feedback, the model is retrained and the data is also saved to a CSV file. The retraining of the model is possible because we have used an online training concept where the already trained model can be retrained by adding new text and label in each iteration.

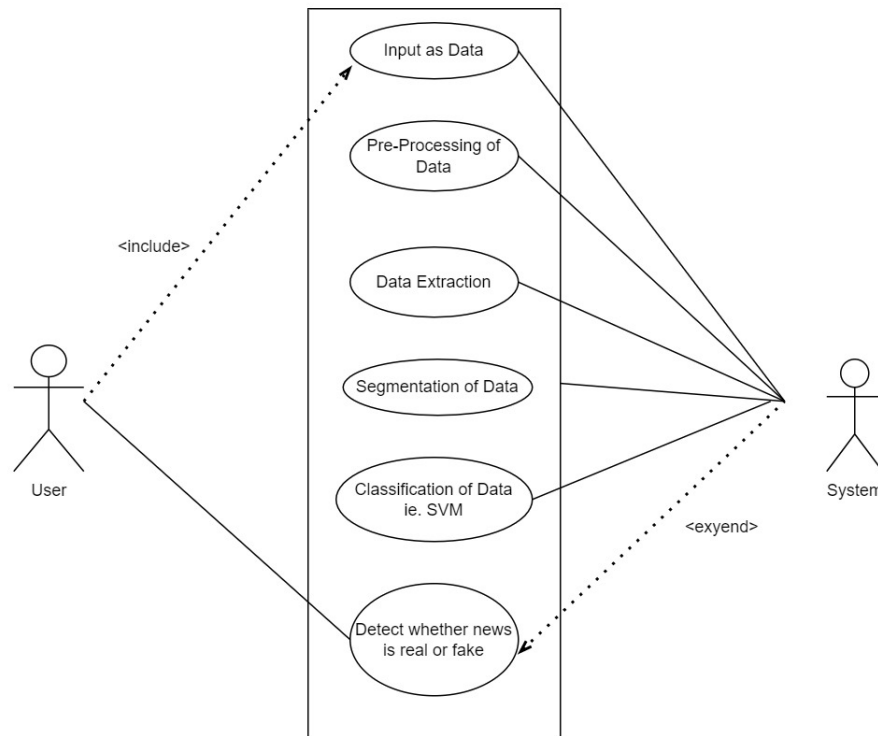


Figure 4.2: Use-case Diagram

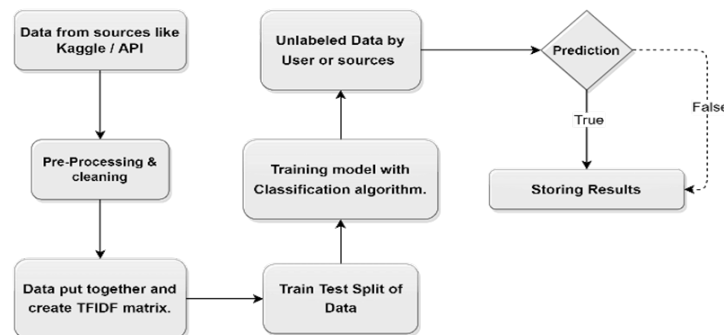


Figure 4.3: Block Diagram

Figure 4.3 Block diagram shows the representation of the project in summarized manner where one or two method are contained in one block rather than including each and every step. The first block represents the data collection process which includes data-set from kaggle for training and API after model is ready. The next block shows pre-processing phase which is data cleaning for making it ready for machine to know it more accurately. Then the data is summed up and a TFIDF matrix is generated and then train test split is preformed. In train test split the data-set is splitted into two. Where the training part has more data and testing has less. The Model is provided the training data from which it learns and gets ready to predict. After getting trained the model needs to be tested to provide accuracy, there comes in the testing data-set. This will provide use results by comparing the predicted data vs the actually present data in dataset. Then next block here shows the input data from user which is then given to the

model for prediction and further store the results.

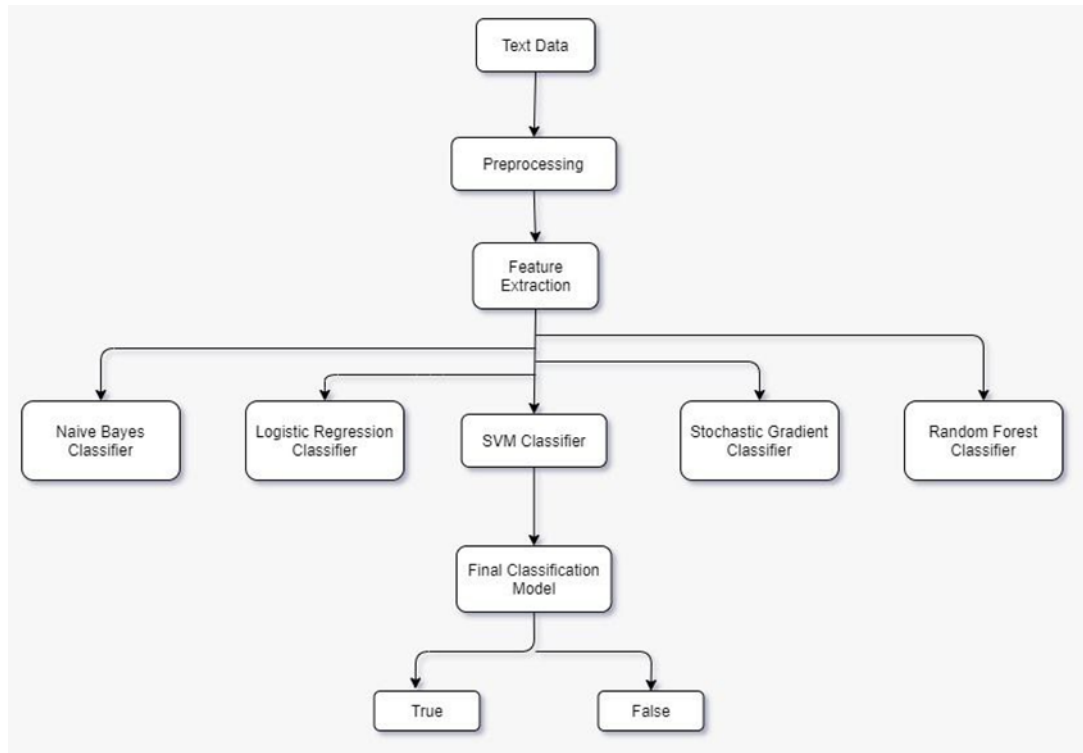


Figure 4.4: Activity Diagram

Initially, we collected a labeled dataset from Kaggle for training our model. After collecting the dataset, it needs to be cleaned to make it more efficient for the model to understand the pattern. Once cleaning is done, we now extract features from our linguistic data (data in text form) using the TFIDF vectorization technique. After feature selection, the data is sliced into trains and tested in the ratio specified. The classification models are trained, and their accuracy is measured. Model selection is now completed, so the trained model is now ready to be used for prediction purposes. The user now enters the data for prediction purposes. Through the back-end, the data travels, and then, after preprocessing, it is passed to the model as input, and an output is provided. The user then has the choice to support the prediction or not. If the prediction is positively supported, its prediction remains the same; else it is changed and fed to the model again for training as well as added to the CSV file for storing purposes.

4.2 Motivation

This model was created to give us a taste of the world of Machine Learning and Data Analytics. The model is selected based on which one provides the highest level of accuracy, and the data provided by the user is entered into the model, which predicts the outcome. Rather than using a database, we prefer to use CSV files as a method for storing data. The first component that aids in this analysis is crucial to the rest of the process. Data Pre-Processing is the name of the game. After considering all of the classes, we can assess whether a news report is true or false. A new story's authenticity is also determined. The outputs are then classified using classification techniques.

4.3 Model Used

TF (Term Frequency):

This model was created to give us a taste of the world of Machine Learning and Data Analytics. The model is selected based on which one provides the highest level of accuracy, and the data provided by the user is entered into the model, which predicts the outcome. Rather than using a database, we prefer to use CSV files as a method for storing data. The first component that aids in this analysis is crucial to the rest of the process. Data Pre-Processing is the name of the game. After considering all of the classes, we can assess whether a news report is true or false. A new story's authenticity is also determined. The outputs are then classified using classification techniques.

IDF (Inverse Document Frequency)

The number one record has been broken. You can figure out how many papers there are when you divide how many documents there are by how many sheets there are, which is how many documents divided by how many sheets there are. The IDF is how many publications have the word win them (Inverse Document Frequency). If the term 'isn't' is often used in many documents in a corpus, it has less weight than if it is used a lot. Words that appear on multiple occasions in a document, as well as in other papers, may not be regarded as relevant. The IDF assesses a term's relevance over the entire corpus. The TFIDF Vectorizer converts into a raw document TF-IDF features matrix.

Logistic Regression:

The term "logistic regression" refers to a mathematical technique for predicting the outcome. Supervised learning model is used for the classification of results based on analyzing them by plotting points and making a curve to differentiate both types of points. Because the output is constantly dependent Logistic regression is a generalized linear model based on the sum of the inputs and parameters. In this type rather than using a line, we fit an S-shaped logistic function that predicts the values as binary (0, 1).

Multinomial NB (Naïve Bayes):

Multinomial NB is a sort of classifier that is suitable for classification with discrete features. Counting words for text classification is a discrete characteristic example. It is commonly used to solve document categorization problems, and it predicts the outcomes based on the frequency of words.

SVM:

There are many ways to classify things, and one of them is the SVM (Support Vector Machine). SVM's main goal is to make a boundary that divides n-dimensional space into classes so that we can easily cross it and categorize data by plotting or fitting it into a specific category, which we do when we look at the data. The problem with SVM is that it only considers the points nearest

to the hyperplane which makes other points useless. We can use SVM with a non-linear method to get the best accuracy.

To summarize, we are currently employing Naive Bayes, but we intend to change this in the future by upgrading it. This was chosen because it delivered the highest level of precision in the online training strategy using RIVER and was average in the normal Scikit Learn approach. Once we've fulfilled our model's goal, we can employ hyper-parameter tuning to improve our model selection.

4.4 MODEL ANALYSIS TERMS

Confusion Matrix:

A confusion matrix, also called an error matrix, a confusion matrix is employed to specifically solve the matter of applied mathematics classification within the machine learning field. Essentially, a confusion matrix is employed to allow the outline of however well the classification model or classifier has performed on a data-set that we all know the verity values. A confusion matrix features a tabular structure. The performance of the classifier is unbelievable. The confusion between totally different categories is well known. For example, mislabeling one category as another. The performance measures taken square measure calculated exploitation of this matrix. A confusion matrix is principally employed to represent the variety of outcomes of predictions on a classification model. The quantity of incorrect and proper predictions with values of count is summarised by employing a confusion matrix. This is often the key to the confusion matrix. The classification model is confused in an exceedingly different range of ways once it makes the predictions. This is often depicted by a confusion matrix. The errors that the classifier makes and also the styles of errors that are created are shown to exploit the confusion matrix.

Accuracy Score:

The most generally used metric for classification is accuracy, which provides information on a fraction of samples that are properly predicted. The Sklearn library is utilized. to predict the accuracy score, which takes the input as data-sets and displays the percentage of the model's precision using data-set labels and projected data-set labels.

Precision:

It is another often-used performance statistic for classification, which indicates the proportion of relevant outcomes, i.e., it identifies relevant data points.

Recall:

Another performance measure used in classification modeling is the proportion of meaningful outcomes, i.e., accurately categorized by the model.

Pipeline:

We cannot run the model just in the Jupyter notebook so we need to create a pipeline which is a '.sav' file that comprises the method to be followed once a data is passed through it.

We discussed here the Confusion matrix for tried & tested models. 1: True & 0: False.

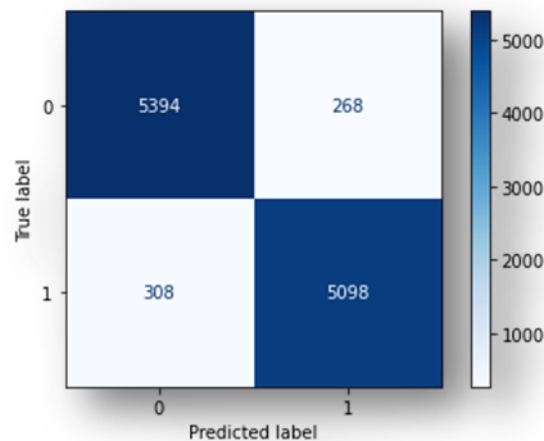


Figure 4.5: Multinomial NB count confusion matrix

In Figure 4.5, the number of true positives is 5394, and the number of false positives is 268. The false negatives are 308, and the true negatives are 5098. This means that the true negatives are correct 95 percent of the time.

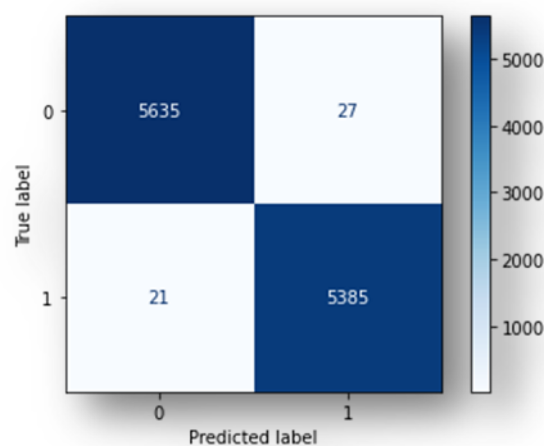


Figure 4.6: Confusion Matrix for Logistic Regression

Here, in figure 4.6 True Positives are 5635,
False Positives are 27,
False Negatives are 21, and
True Negatives are 5385.

Accuracy score would be (True Positives + True Negative) divide (True Positives + False Positives + False Negatives + True Negatives) equals (5635+5385) divide (True Positives + False Positives + False Negatives + True Negatives) equals (5635+5385) divide (5635+27+21+5385) equals 98 percent.

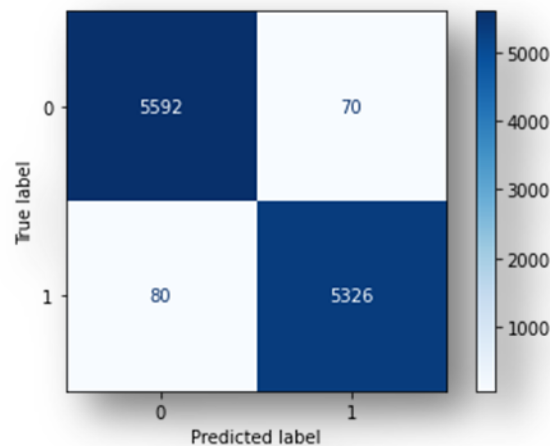


Figure 4.7: Confusion matrix for SVC

Here,in figure 4.7 True Positives equals 5592 False Positives equals 70
False Negatives equals 80
True Negatives equals 5326

Accuracy score equals (True Positives + True Negative) divide (True Positives + False Positives + False Negatives + True Negatives) equals (5592+5326)/ (5592+70+80+5326) equals 98 percent
Precision equals True Positives divides (True Positives + False Negatives) =954/ (954+54) equals 94.6 percent

Recall equals True Positives divides (True Positives + False Negatives) equals 954/ (954+79) equals 92.3 percent.

4.5 Existing System Architecture

A tutorial in an article showed how to "capture," score, integrate, and build a recurrent neural network model that can tell when news stories are based on fake news. They used a recurrent neural network (RNN), which looks for temporal patterns of user activity around a particular article or text and then looks for source properties in that behavior. Based on the information collected, an algorithm has been developed to identify and classify fake news [3]. According to the research, even a basic network model may outperform more complicated ones. The

model's complexity is not the ideal strategy here, and careful parameter and data selection are essential. Others have looked at linguistically infused neural networks and utilized convolutional neural networks. Another research, for example, explains how to identify tweets using a linguistically-infused neural network model that employs Long Short-Term Memory (LSTM) and Convolutional Neural Networks to recognize the words in the tweet (CNN). The Glove library of pre-trained vectors was used to introduce the linguistic component [11]. As a result of the multiple tries, everything is muddled and scattered. This area of study offers a great deal of space for research and improvement, particularly since news announcements include various variants, such as sarcasm, truncation, and metaphors. If you want to stop the spread of fake news, you need to act more quickly. However, efforts have been made to build a high-quality dataset from reliable and large data. This model used one of these benchmark data-sets, and it worked well.

4.6 Data:

Classifying a news item as "fake news" can be a difficult and time-consuming process. As a result, an existing dataset has been made use of, which has already collected and identified phoney news.. The data source used for this 7 project is the LIAR dataset. Given below is a brief description of the data files used for this study. The dataset has been cited in the paper [12] "Liar, Liar Pants on Fire": A New Benchmark Data-set for Fake News Detection, to be published in the Proceedings of the Association for Computational Linguistics' 55th Annual Meeting (ACL 2017), brief paper, Vancouver, BC, Canada, July 30-August 4, ACL [13]. The original dataset contained 13 variables/columns for train, test, and validation sets. Only two variables from the original dataset were used for this classification job for the purpose of simplicity. The other factors can be used in a subsequent study to get a more thorough picture. The two columns utilised are 'Statement,' which contains the actual news statement, and 'Label,' which indicates whether the statement is true or incorrect. The procedure used for reducing the number of classes in the dataset: True, mostly true, half-true become 'True'. Barely true, false, pants-fire becomes 'False'. It's worth noting that this dataset includes three separate TSV files for train, validation, and test. As a result, there is no need to manually divide the data into test, train, and validation.

4.7 Model:

Classifiers may not be able to do well if they have many text data or if the test vectors are not the right shape or size. When extracting text characteristics, for example, frequent noisy words, also known as stop words,' are less significant keywords that do not contribute to the genuine meaning but rather increase the dimensionality of the feature set and may be removed to enhance efficiency. It contributes to reducing the size and complexity of the given text and the provision of text context for feature extraction. Lemmatization may also condense many words into a single discrete form [14].

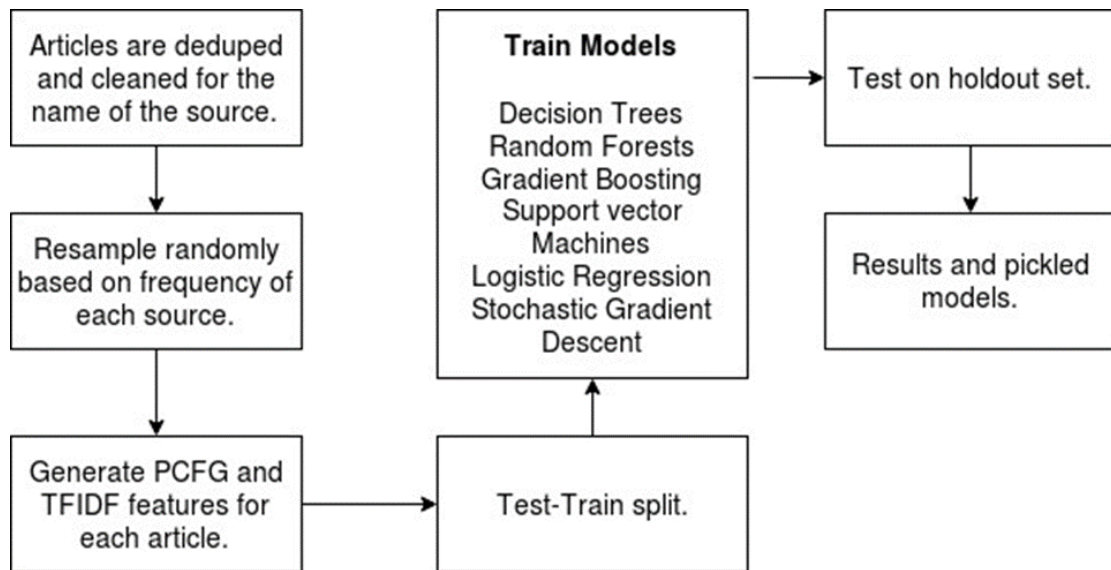
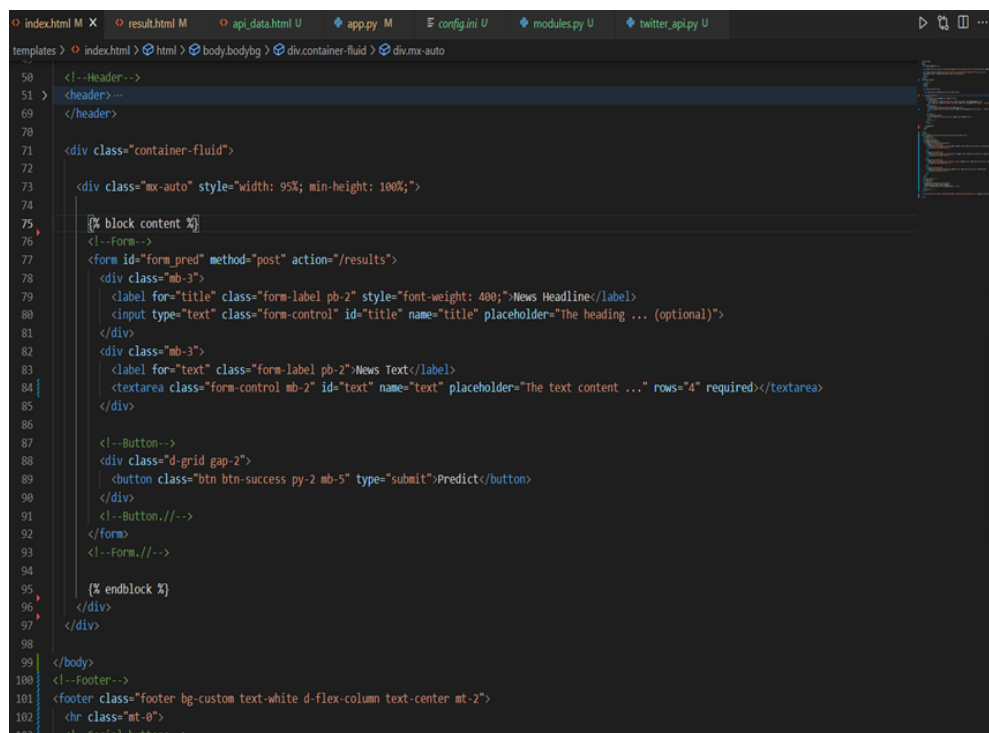


Figure 4.8: operation of how the existing model works

Chapter 5

Project Implementation

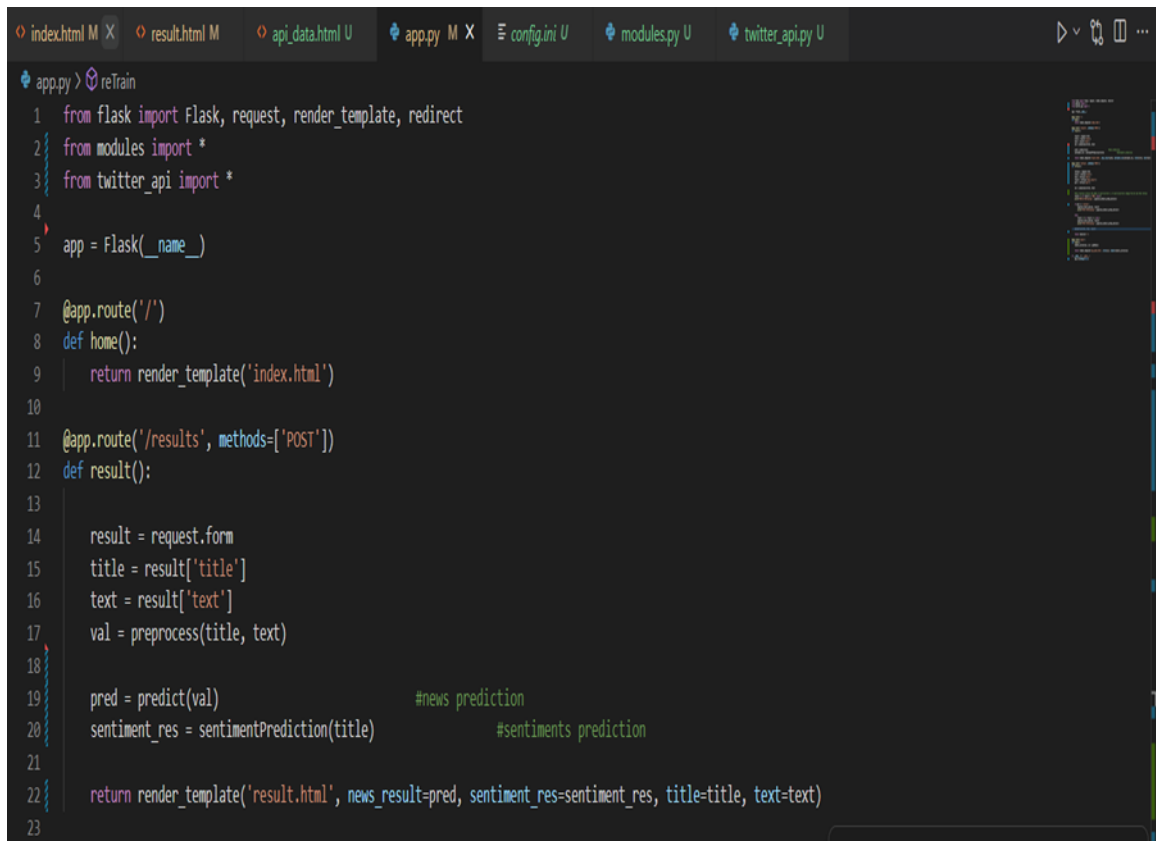
This chapter contains the snapshots of the code written to implement different modules of the system. It starts with creating the main flask app through which the execution starts and then creating a main function that would handle the flow by calling the next functions as needed. Below is the screenshot of our first interface code which is the front-end code from which the execution begins and the user sees the page.



```
50 <!--Header-->
51 <header-->
69 </header>
70
71 <div class="container-fluid">
72
73   <div class="mx-auto" style="width: 95%; min-height: 100%;">
74
75     {% block content %}
76     <!--Form-->
77     <form id="form_pred" method="post" action="/results">
78       <div class="mb-3">
79         <label for="title" class="form-label pb-2" style="font-weight: 400;">News Headline</label>
80         <input type="text" class="form-control" id="title" name="title" placeholder="The heading ... (optional)">
81       </div>
82       <div class="mb-3">
83         <label for="text" class="form-label pb-2">News Text</label>
84         <textarea class="form-control mb-2" id="text" name="text" placeholder="The text content ..." rows="4" required></textarea>
85       </div>
86       <!--Button-->
87       <div class="d-grid gap-2">
88         <button class="btn btn-success py-2 mb-5" type="submit">Predict</button>
89       </div>
90     </form-->
91   </div>
92 </div>
93 </div>
94
95 {% endblock %}
96 </div>
97 </div>
98
99 </body>
100 <!--Footer-->
101 <footer class="footer bg-custom text-white d-flex-column text-center mt-2">
102   <hr class="mt-0">
103   <!--Social buttons-->
```

Figure 5.1: Index.html

The Figure 5.1 code snippet shows the HTML code for the web page, which the user looks at the start which contains some CSS with the help of the responsiveness of Bootstrap5 which makes it more attractive and convenient. The next step in the user's process of execution is to interact with the web-page by filling out the form which asks for the title and text of the news. Here the title is optional but the text is mandatory. After getting the necessary inputs the user can click on the Predict button which will take him to the next process where it is redirected to the results page.



```
app.py > reTrain
1 from flask import Flask, request, render_template, redirect
2 from modules import *
3 from twitter_api import *
4
5 app = Flask(__name__)
6
7 @app.route('/')
8 def home():
9     return render_template('index.html')
10
11 @app.route('/results', methods=['POST'])
12 def result():
13
14     result = request.form
15     title = result['title']
16     text = result['text']
17     val = preprocess(title, text)
18
19     pred = predict(val)           #news prediction
20     sentiment_res = sentimentPrediction(title)   #sentiments prediction
21
22     return render_template('result.html', news_result=pred, sentiment_res=sentiment_res, title=title, text=text)
23
```

Figure 5.2: App.py

Figure 5.2 shows the stage where the text and title are extracted from the form and then passed to the pre-process function which in turn returns a single string containing the pre-process data and then we use our predict function to get the results of the text depicting as “Real” or “Not Real”. This is displayed on the web on the results page which shows the result along with the input which was provided. It also provides a button that holds the value of correctly predicted or not. This button works as feedback to the prediction by the machine. It was added to make the model more accurate with the help of the newly obtained data and user feedback – most important for any organization.

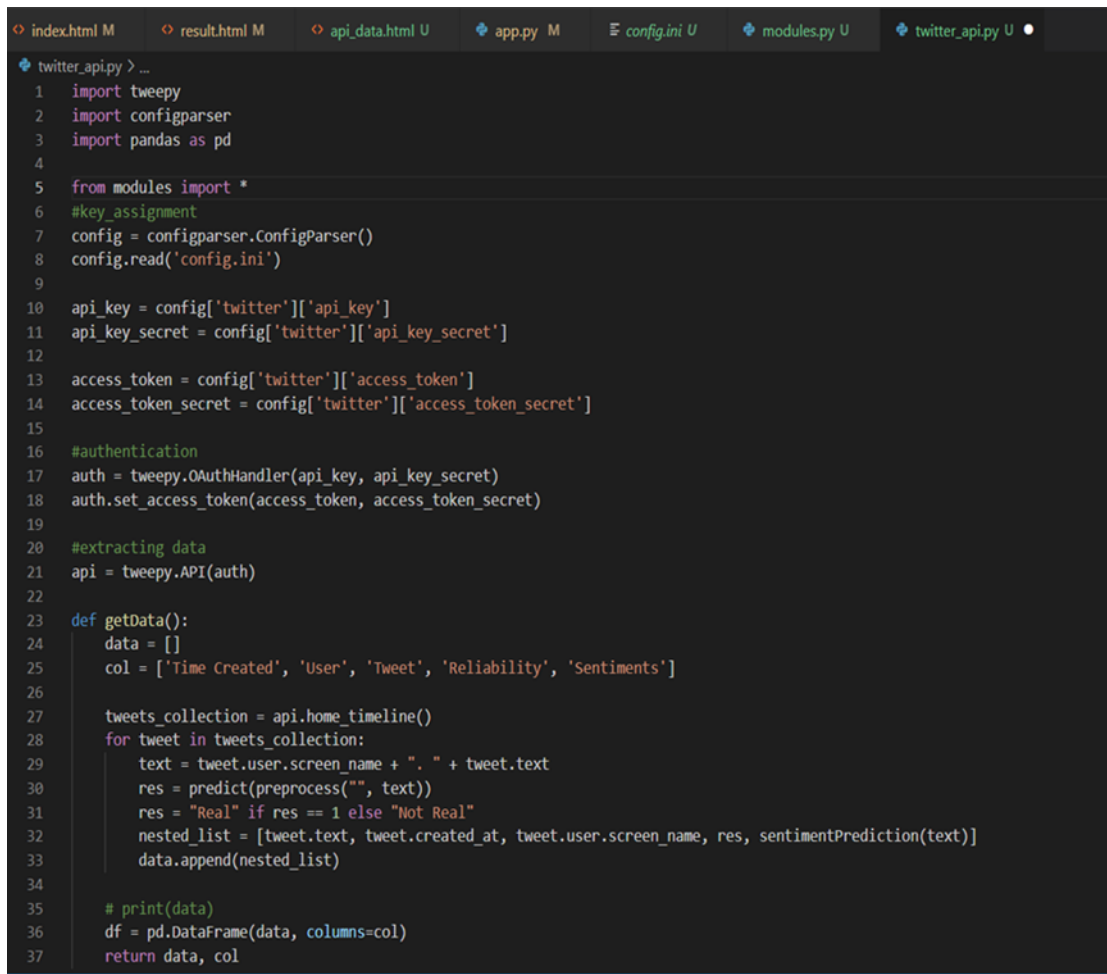
```

index.html M  result.html M X  api_data.html U  app.py M  config.ini U  modules.py U  twitter_api.py U
templates > result.html > div > div.text-center > form > div.row.text-center.mb-5 > input#hide
1  {% extends 'index.html' %}
2  {% block content %}
3  <style>
4  #hide{display: none;}
5  </style>
6  <div>
7
8      <div class="text-center" style="width: 100%;">
9
10         <form action="/retrain" method="post">
11             <input type="text" name="title" id="hide" value="{{title}}"/>
12             <h4> {{title}} </h4> <hr/>
13             <textarea class="form-control" name="text" rows="12" value="{{text}} </textarea> <br/>
14             <div class="row text-center mb-5">
15                 <input type="text" name="news_result" id="hide" value="{{news_result}}"/>
16                 <div class="col-md-6 py-2"> <h1> {{news_result}} </h1> {{sentiment_res}} </div>
17                 <div class="col-md-3 py-2 d-grid">
18                     <button type="submit" class="btn btn-success" name="poll" value="correct"> Correctly Predicted</button>
19                 </div>
20                 <div class="col-md-3 py-2 d-grid">
21                     <button type="submit" class="btn btn-danger" name="poll" value="incorrect"> Incorrectly Predicted</button>
22                 </div>
23             </div>
24         </form>
25
26     </div>
27
28 </div>
29 <!--container.-->
30
31 <script>
32     function cvalue(){ document.getElementById("cvalue").style.display = "block"; }
33 </script>
34 </div>
35
36 {% endblock %}

```

Figure 5.3: Result.html

Figure 5.3 shows the results page where the Html code is used to take the input from the new form created for taking feedback over the prediction received from the user. And now as soon as the feedback is completed and submitted it is taken as input and passed further to the retrain function which then helps in categorizing it into correct or not and retrains the model and also passes it to the function where it will be written to a CSV file. In this way, we also retrain the model along with saving the newly obtained data in the dataset.



```

1 import tweepy
2 import configparser
3 import pandas as pd
4
5 from modules import *
6 #key_assignment
7 config = configparser.ConfigParser()
8 config.read('config.ini')
9
10 api_key = config['twitter']['api_key']
11 api_key_secret = config['twitter']['api_key_secret']
12
13 access_token = config['twitter']['access_token']
14 access_token_secret = config['twitter']['access_token_secret']
15
16 #authentication
17 auth = tweepy.OAuthHandler(api_key, api_key_secret)
18 auth.set_access_token(access_token, access_token_secret)
19
20 #extracting data
21 api = tweepy.API(auth)
22
23 def getData():
24     data = []
25     col = ['Time Created', 'User', 'Tweet', 'Reliability', 'Sentiments']
26
27     tweets_collection = api.home_timeline()
28     for tweet in tweets_collection:
29         text = tweet.user.screen_name + ". " + tweet.text
30         res = predict(preprocess("", text))
31         res = "Real" if res == 1 else "Not Real"
32         nested_list = [tweet.text, tweet.created_at, tweet.user.screen_name, res, sentimentPrediction(text)]
33         data.append(nested_list)
34
35     # print(data)
36     df = pd.DataFrame(data, columns=col)
37     return data, col

```

Figure 5.4: TwitterApi.py

Figure 5.4 is the code written for integrating the Twitter API for the work on Twitter data, we needed API so we are using the famous python library tweepy for making the connection between python and Twitter and using the predefined methods which help us take the data and make it go through the process with our functions and then make a list which would help us display it over the web. Further, the feedback mechanism goes hand in hand with the earlier processing and predicting. All these follow quite similar but different approaches to handling the data and showing the user what meaning news could depict.

Chapter 6

Testing

6.1 Unit Testing

Out of the different standard testing methods available, unit testing has been used as our main testing method. It is a testing process that tests individual components of the application in terms of its functionality and working. In our online fake news detection and sentiment analysis web application, we have a few such units in place that perform their respective action based on the user's input. These small units include getting data from the user through a form, cleaning the data and passing it to the model, updating results on the next page, give feedback form to take input from the user's end. From the user's end, it includes providing feedback on the prediction which was done by the machine. And then the next component is retraining based on the feedback received from the user. For all the above-mentioned components we found testing them individually has best suited here. Unit testing has also been used because it helps in finding any issues at an earlier stage and in the individual component rather than finding it later in the entire application.

6.2 Integration Testing

After each unit is tested individually, it is integrated with other units to create a functioning module that can perform specific tasks. These are then tested as a combined group through integration testing to ensure the whole segment of the application behaves as expected. So, in this system, once the individual components like twitter analysis and text analysis of news are completed it is combined and developed together on a single web page which is then tested as a whole.

Chapter 7

Result

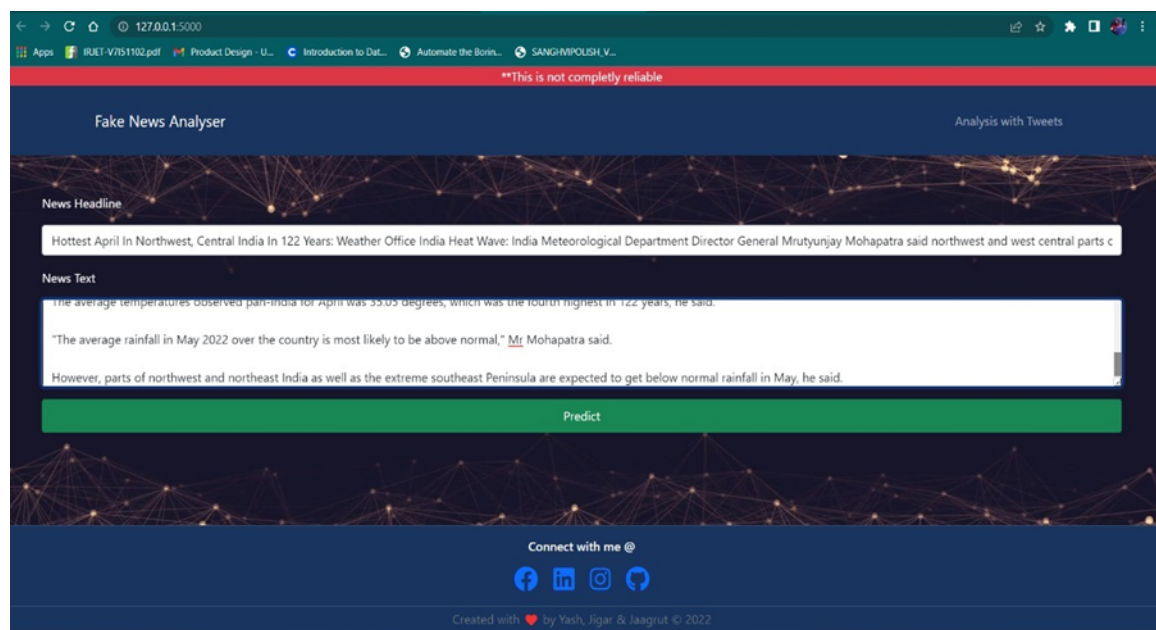


Figure 7.1: Getting News from online sources

Figure 7.1 shows news headline text-bar is where a user will enter the title of the news, and the news section is where a user will enter the news that he wants to forecast. The news title is not required because not every story has one. We will acquire the necessary result from machine learning and will be able to provide feedback to the user on whether the news is phoney or authentic.

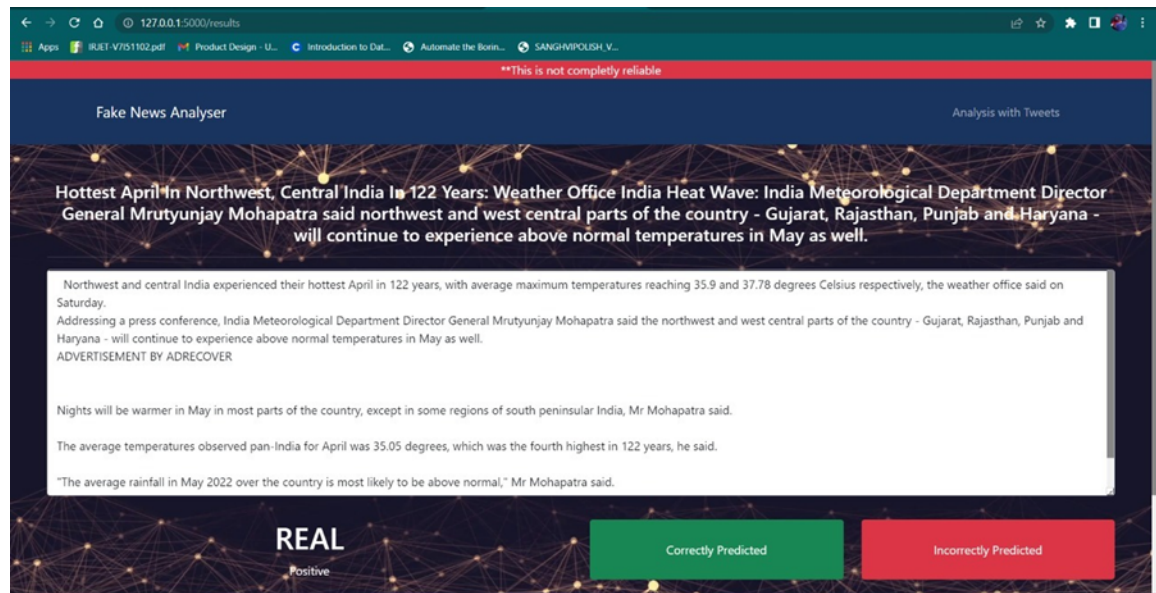


Figure 7.2: Predicted Results

Figure 7.2 depicts how our website gives us with the prediction as well as the model's projected attitudes, as well as a comment box to assist us categorise the news more properly, from which we will retrain our model and acquire fresh data. This is the results page where the user gets redirected as soon as he hits the predict button. This is the main page where the main value is added to the data-set and also the page where the users get answers to their questions.

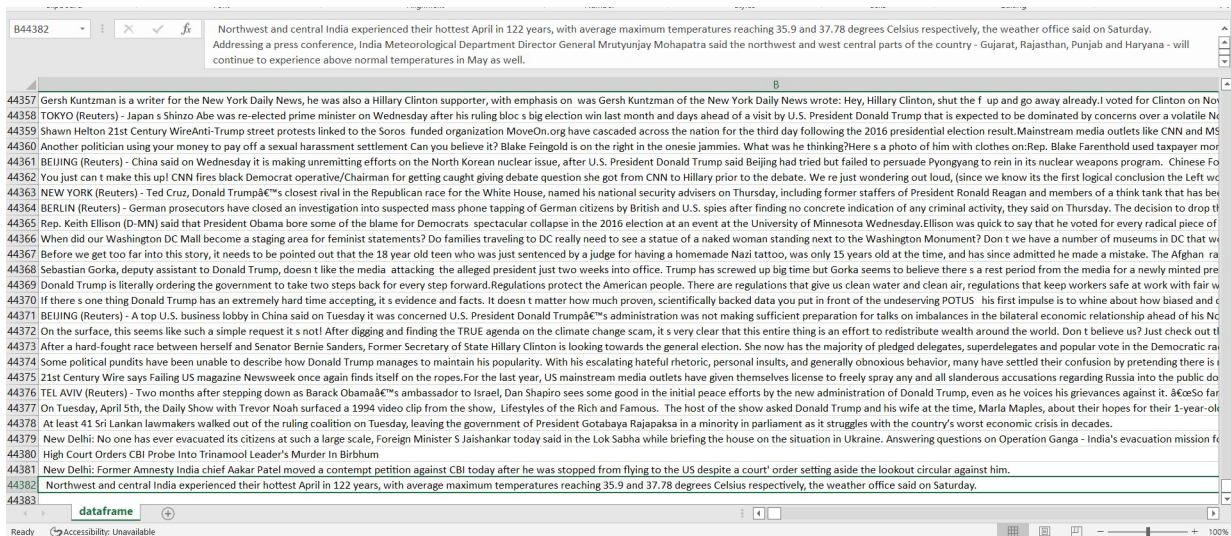


Figure 7.3: Updated Data set after Each News Added

Figure 7.3 is the screen shot of the csv file, which was the dataset used for training our model. Here is this screenshot we are showing that the newly processed and obtained data was also updated in the dataset by us. It is very beneficial and helpful for building new model as it would have a great combination of old dataset along with the new dataset. The model created in future with this dataset would give better accuracy than today's model.

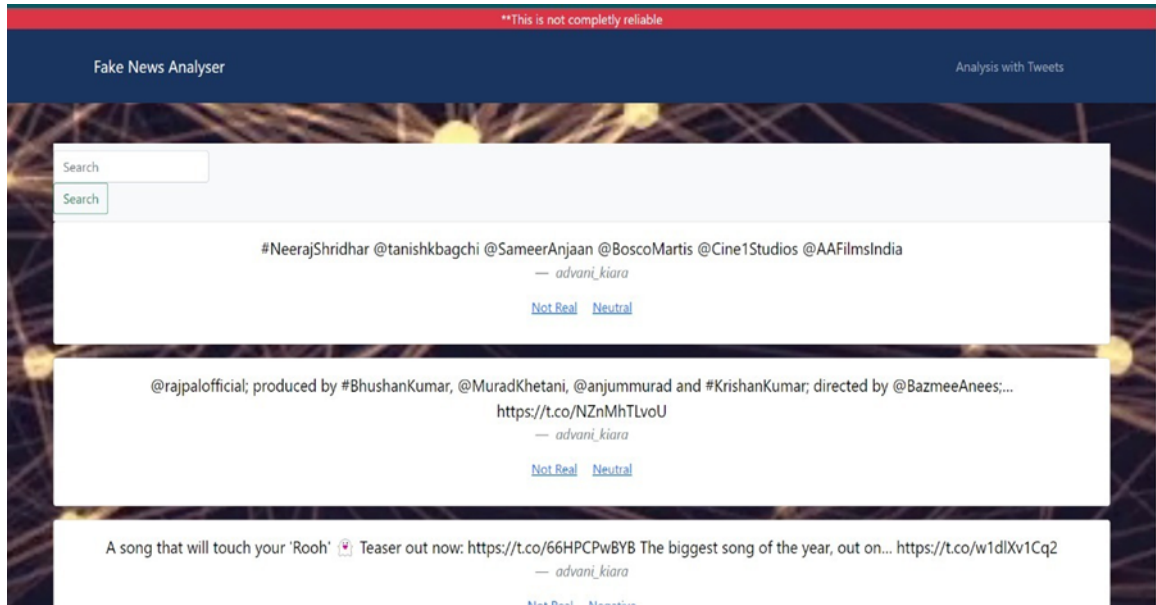


Figure 7.4: Twitter sentiment analysis

Figure 7.4 is the screenshot of the page which currently shows a Twitter dashboard view of tweets along with the results we got by putting the tweets as input. We will further improve the length of the tweets and the dashboard view for the user to feel better to get their findings of particular news from us. This would help us in making the user experience better and more reliable for users visiting the site for text-based as well as real-time news data.

Chapter 8

Conclusions and Future Scope

Fake news is categorized as any kind of made-up story to deceive or mislead. In this paper, we are trying to present the solution to the fake news detection task by using machine learning techniques. Machine Learning has opened a new front in the warfare against fake news. One must take advantage of this front and exploit it thoroughly. This paper has shown that the front is viable. The usage of machine learning in the identification of fake news is still in its infancy. Every model developed or system proposed moves us closer to a fake news-free internet. With this, endeavors are being made to automate the task of fake news detection. The most mainstream of such actions include blacklisting of sources and authors that are unreliable. Even though these tools are useful, to produce a progressive, end-to-end solution, we are required to represent tougher cases where reliable sources and authors are responsible for releasing fake news. The outcomes of this project show the capability of ML to be fruitful in this task. We have tried to build a model that helps in catching many intuitive indications of real and fake news as well as in the visualization of the classification decision. While we have only used basic algorithms, there is a high potential for the creation of better models.

Bibliography

- [1] A. Mahabub, "A robust technique of fake news detection using ensemble voting classifier and comparison with other classifiers," *SN Applied Sciences*, vol. 2, no. 4, pp. 1–9, 2020.
- [2] H. Reddy, N. Raj, M. Gala, and A. Basava, "Text-mining-based fake news detection using ensemble methods," *International Journal of Automation and Computing*, vol. 17, no. 2, pp. 210–221, 2020.
- [3] M. D. Ibrishimova and K. F. Li, "A machine learning approach to fake news detection using knowledge verification and natural language processing," in *International Conference on Intelligent Networking and Collaborative Systems*, pp. 223–234, Springer, 2019.
- [4] M. Granik and V. Mesyura, "Fake news detection using naive bayes classifier," in *2017 IEEE first Ukraine conference on electrical and computer engineering (UKRCON)*, pp. 900–903, IEEE, 2017.
- [5] A. Jain and A. Kasbe, "Fake news detection," in *2018 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS)*, pp. 1–5, IEEE, 2018.
- [6] A. Thota, P. Tilak, S. Ahluwalia, and N. Lohia, "Fake news detection: a deep learning approach," *SMU Data Science Review*, vol. 1, no. 3, p. 10, 2018.
- [7] S. Bajaj, "The pope has a new baby! fake news detection using deep learning," *CS 224N*, pp. 1–8, 2017.
- [8] S. Kaur, P. Kumar, and P. Kumaraguru, "Automating fake news detection system using multi-level voting model," *Soft Computing*, vol. 24, no. 12, pp. 9049–9069, 2020.
- [9] C. Conforti, M. T. Pilehvar, and N. Collier, "Towards automatic fake news detection: cross-level stance detection in news articles," in *Proceedings of the first workshop on fact extraction and VERification (FEVER)*, pp. 40–49, 2018.
- [10] A. Habib, M. Z. Asghar, A. Khan, A. Habib, and A. Khan, "False information detection in on-line content and its role in decision making: a systematic literature review," *Social Network Analysis and Mining*, vol. 9, no. 1, pp. 1–20, 2019.
- [11] S. Volkova, K. Shaffer, J. Y. Jang, and N. Hodas, "Separating facts from fiction: Linguistic models to classify suspicious and trusted news posts on twitter," in *Proceedings of the 55th annual meeting of the association for computational linguistics (volume 2: Short papers)*, pp. 647–653, 2017.
- [12] W. Y. Wang, "'liar, liar pants on fire': A new benchmark dataset for fake news detection," *arXiv preprint arXiv:1705.00648*, 2017.

- [13] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, “Fake news detection on social media: A data mining perspective,” *ACM SIGKDD explorations newsletter*, vol. 19, no. 1, pp. 22–36, 2017.
- [14] J. C. Reis, A. Correia, F. Murai, A. Veloso, and F. Benevenuto, “Supervised learning for fake news detection,” *IEEE Intelligent Systems*, vol. 34, no. 2, pp. 76–81, 2019.

Appendix

Installation and Running of the project

- Step1: For Python 3:
Create and name a virtual environment in python 3 with:
`py -3 -m venv <name of environment>`
For Python 2: create a virtual environment with the virtualenv module:
`py -2 -m virtualenv <name of environment>`
list the folder structure using the dir command:
`dir *<project name>*`
- Step2
Activate the Environment: Activate the virtual environment before installing Flask. The name of the activated environment shows up in the CLI after activation. For windows, activate the virtual environment with:
`<name of environment>|Scripts|activate`
- Step3 Flask: Install Flask within the activated environment using pip: `pip install Flask` (Flask is installed with all the dependencies.)
- Step4
Numpy
`pip install Numpy`
- Step5
Pandas
`pip install pandas`
- Step6
Nltk
`pip install Nltk`
- Step7
Jupyter Notebook
`pip install notebook`

Publication

Paper Title: A Web Framework to Predict Fake News Using Machine Learning

Authors: Jaagrut Shah, Jigar Desai, Yash Jain, Prof. Apeksha Mohite and Prof. Geetanjali Kalme

Conference: Third International Conference on Intelligent Computing, Instrumentation and Control Technologies, ICICICT 2022.

Conference Date: 11TH AND 12TH AUGUST 2022

Paper Status: Paper Accepted