FAKE NEWS DETECTION

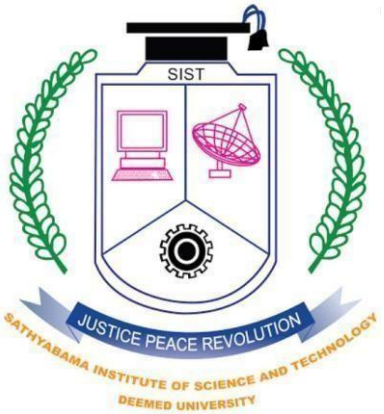
Submitted in partial fulfillment of the requirements for

the award of

Bachelor of Engineering degree in Computer Science and Engineering

by

LAKSHTHA SRI S (**37110401**)
MANJU SHAJAN (**37110431**)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,

CHENNAI – 600 119

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with "A" grade by NAAC
Jeppiaar Nagar, Rajiv Gandhi Salai,
Chennai – 600119
www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this project report is the bonafide work of **LAKSHTHA SRI S (37110401)** and **MANJU SHAJAN (37110431)** who carried out the project entitled "**FAKE NEWS DETECTION**" under my supervision from **August 2020** to **March 2021**.

Internal Guide
Dr.Mercy Paul Selvan M.E., Ph.D.,

Head of the Department

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

Dr. L. Lakshmanan, M.E., Ph.D.,

Submitted for Viva voce Examination held on_____

Internal Examiner **External Examiner**

DECLARATION

LAKSHTHA SRI S (37110401**) and MANJU SHAJAN (**37110431**)** hereby declare that the Project Report entitled "**FAKE NEWS DETECTION"** is done by me under the guidance of **Dr.Mercy Paul Selvan M.E., Ph.D.,** Department of Computer Science and Engineering at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering**.**

DATE:

PLACE: CHENNAI **SIGNATURE OF THE CANDIDATE**

ACKNOWLEDGEMENT

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

I convey my thanks to **Dr. T. Sasikala, M.E., Ph.D., Dean**, School of Computing, **Dr. S. Vigneswari, M.E., Ph.D., and Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department** of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

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

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

In recent years, due to the booming development of online social networks, fake news for various commercial and political purposes has been appearing in large numbers and widespread in the online world. With deceptive words, online social network users can get infected by these online fake news easily, which has brought about tremendous effects on the offline society already. An important goal in improving the trustworthiness of information in online social networks is to identify the fake news timely. This paper aims at investigating the principles, methodologies and algorithms for detecting fake news articles, creators and subjects from online social networks and evaluating the corresponding performance. Information preciseness on Internet, especially on social media, is an increasingly important concern, but web-scale data hampers, ability to identify, evaluate and correct such data, or so called "fake news," present in these platforms. In this paper, we propose a method for "fake news" detection and ways to apply it on Facebook, one of the most popular online social media platforms. This method uses Naive Bayes classification model to predict whether a post on Facebook will be labeled as real or fake. The results may be improved by applying several techniques that are discussed in the paper. Received results suggest, that fake news detection problem can be addressed with machine learning methods.

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	
	LIST OF FIGURES	
	LIST OF ABBERVATION	
1	**INTRODUCTION**	**9**
	Objective	10
	Significance of fake news	10
2	**LITERATURE SURVEY**	**11**
	Overview	11
	Literature Survey	11
3	**METHODOLOGY**	**17**
	System Architecture Diagram	17
	Modules Implementation	17
	Data collection	17
	Preprocessing data	19
	Feature extraction	20
	Training the classifier	21
	Technology Used	22
	Python	22
	Natural Language processing (NLP)	23
	Machine Learning-Algorithms	24
	Naïve Bayes Algorithm	24
	Logistic Regression	25

	K-Nearest Neighbor (KNN) Algorithm	26
	Support Vector Machine(SVM) Algorithm	28
	Implementation	30
	Data flow	31
4	**RESULT AND DISCUSSION**	**33**
5	**CONCLUSION**	**37**
	REFERENCE	**38**
	APPENDIX	**40**
	A.SAMPLE CODE B.OUTPUT SCREENSHOT C.PLAGIARISM REPORT	

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE NO.
3.1	System architecture	17
3.2.1	Data collection	18
3.2.2	Preprocessing data	20
3.3.3.2	Logistic function	26
3.3.3.4	Support vector machine (SVM)	29
3.3.3	Algorithm comparison	29
3.5	Data flow	33
4.1	Microsoft power shell	34
4.2	Train dataset	34
4.3	Webpage	34
4.4	Entering the article	35
4.5	Output of checked article	35
4.6	Output of checked article	36

CHAPTER-1

INTRODUCTION

These days' fake news is creating different issues from sarcastic articles to a fabricated news and plan government propaganda in some outlets. Fake news and lack of trust in the media are growing problems with huge ramifications in our society. Obviously, a purposely misleading story is "fake news" but lately blathering social media's discourse is changing its definition. Some of them now use the term to dismiss the facts counter to their preferred viewpoints.

The importance of disinformation within American political discourse was the subject of weighty attention, particularly following the American president election. The term 'fake news' became common parlance for the issue, particularly to describe factually incorrect and misleading articles published mostly for the purpose of making money through page views. In this paper, it is seeked to produce a model that can accurately predict the likelihood that a given article is fake news. Facebook has been at the epicenter of much critique following media attention. They have already implemented a feature to flag fake news on the site when a user see's it, they have also said publicly they are working on to distinguish these articles in an automated way. Certainly, it is not an easy task. A given algorithm must be politically unbiased

– since fake news exists on both ends of the spectrum – and also give equal balance to legitimate news sources on either end of the spectrum. In addition, the question of legitimacy is a difficult one. However, in order to solve this problem, it is necessary to have an understanding on what Fake News is. Later, it is needed to look into how the techniques in the fields of machine learning, natural language processing help

us to detect fake news.

OBJECTIVES

The objective of this project is to examine the problems and possible significance related with the spread of fake news. We will be working on different fake news data set in which we will apply different machine learning algorithms to train the data and test it to find which news is the real news or which one is the fake news. As the fake news is a problem that is heavily affecting society and our perception of not only the media but also facts and opinions themselves. By using the artificial intelligence and the machine learning, the problem can be solved as we will be able to mine the patterns from the data to maximize well defined objectives. So, our focus is to find which machine learning algorithm is best suitable for what kind of text data set. Also, which data set is better for finding the accuracies as the accuracies directly depends on the type of data and the amount of data. The more the data, more are your chances of getting correct accuracy as you can test and train more data to find out your results.

SIGNIFICANCE OF FAKE NEWS DETECTION

However, on the other hand, social media provides the ideal place for the creation and spread of fake news. Fake news can become extremely influential and has the ability to spread exceedingly fast. With the increase of people using social media, they are being exposed to new information and stories every day. Misinformation can be difficult to correct and may have lasting implications. There for we need to make sure the news we read are correct and real so, there should be be fake news detection in our applications so that we don’t fall for those fake news spreading around us .

CHAPTER – 2

LITERATURE SURVEY

OVERVIEW

The available literature has described many automatic detection techniques of fake news and deception posts. Since there are multidimensional aspects of fake news detection ranging from using chatbots for spread of misinformation to use of click baits for the rumor spreading. There are many click baits available in social media networks including Facebook which enhance sharing and liking Proceedings of posts which in turn spreads falsified information. Lot of work has been done to detect falsified information.

LITERATURE SURVEY

Niall J. Conroy et all in Automatic Deception Detection: Methods for Finding Fake News 2015, made this research surveys the current state-of-the-art technologies that are instrumental in the adoption and development of fake news detection. “Fake news detection" is defined as the task of categorizing news along a continuum of veracity, with an associated measure of certainty. Veracity is compromised by the occurrence of intentional deceptions. The nature of online news publication has changed, such that traditional fact checking and vetting from potential deception is impossible against the flood arising from content generators, as well as various formats and genres. The paper provides a typology of several varieties of veracity assessment methods emerging from two major categories – linguistic cue approaches (with machine learning), and network analysis approaches. We see promise in an innovative hybrid approach that combines linguistic cue and machine learning, with network-based behavioral data. Although designing a fake news detector is not a straightforward problem, we propose

operational guidelines for a feasible fake news detecting system.

Yimin Chen et all in Misleading online content: Recognizing clickbaits as false news 2015, Tabloid journalism is often criticized for its propensity for exaggeration, sensationalization, scare-mongering, and otherwise producing misleading and low quality news. As the news has moved online, a new form of tabloidization has emerged: 'clickbaiting.' 'Clickbait' refers to "content whose main purpose is to attract attention and encourage visitors to click on a link to a particular web page" ['clickbait,' n.d.] and has been implicated in the rapid spread of rumor and misinformation online. This paper examines potential methods for the automatic detection of clickbait as a form of deception. Methods for recognizing both textual and non-textual clickbaiting cues are surveyed, leading to the suggestion that a hybrid approach may yield best results.

Maryam M Najafabadi et all in Deep learning applications and challenges in big data analytics 2015, Big Data Analytics and Deep Learning are two high-focus of data science. Big Data has become important as many organizations both public and private have been collecting massive amounts of domain-specific information, which can contain useful information about problems such as national intelligence, cyber security, fraud detection, marketing, and medical informatics. Companies such as Google and Microsoft are analyzing large volumes of data for business analysis and decisions, impacting existing and future technology. Deep Learning algorithms extract high-level, complex abstractions as data representations through a hierarchical learning process. Complex abstractions are learnt at a given level based on relatively simpler abstractions formulated in the preceding level in the hierarchy. A key benefit of Deep Learning is the analysis and learning of massive amounts of unsupervised data, making it a valuable tool for Big Data Analytics where raw data is largely unlabeled and un-categorized. In the present study, we explore how Deep Learning can be utilized for addressing some important problems in Big Data Analytics, including

extracting complex patterns from massive volumes of data, semantic indexing, data tagging, fast information retrieval, and simplifying discriminative tasks. We also investigate some aspects of Deep Learning research

that need further exploration to incorporate specific challenges introduced by Big Data Analytics, including streaming data, high-dimensional data, scalability of models, and distributed computing. We conclude by presenting insights into relevant future works by posing some questions, including defining data sampling criteria, domain adaptation modeling, defining criteria for obtaining useful data abstractions, improving semantic indexing, semi-supervised learning, and active learning.

Eugenio Tacchini et all in Some like it hoax: Automated fake news detection in social networks 2017, In recent years, the reliability of information on the Internet has emerged as a crucial issue of modern society. Social network sites (SNSs) have revolutionized the way in which information is spread by allowing users to freely share content. As a consequence, SNSs are also increasingly used as vectors for the diffusion of misinformation and hoaxes. The amount of disseminated information and the rapidity of its diffusion make it practically impossible to assess reliability in a timely manner, highlighting the need for automatic hoax detection systems. As a contribution towards this objective, we show that Facebook posts can be classified with high accuracy as hoaxes or non-hoaxes on the basis of the users who "liked" them. We present two classification techniques, one based on logistic regression, the other on a novel adaptation of Boolean crowdsourcing algorithms. On a dataset consisting of 15,500 Facebook posts and 909,236 users, we obtain classification accuracies exceeding 99% even when the training set contains less than 1% of the posts. We further show that our techniques are robust: they work even when we restrict our attention to the users who like both hoax and non-hoax posts. These results suggest that mapping the diffusion pattern of information can be a useful component of automatic hoax detection systems.

Chengcheng Shao et all in The spread of fake news by social bots 2017, The massive spread of fake news has been identified as a major global risk and has been alleged to influence elections and threaten democracies. Communication, cognitive, social, and computer scientists are engaged in efforts to study the complex causes for the viral diffusion of digital misinformation and to develop solutions, while search and social media platforms are beginning to deploy countermeasures. However, to date, these efforts have been mainly informed by anecdotal evidence rather than systematic data. Here we analyze 14 million messages spreading 400 thousand claims on Twitter during and following the 2016

U.S. presidential campaign and election. We find evidence that social bots play a key role in the spread of fake news. Accounts that actively spread misinformation are significantly more likely to be bots. Automated accounts are particularly active in the early spreading phases of viral claims, and tend to target influential users. Humans are vulnerable to this manipulation, retweeting bots who post false news. Successful sources of false and biased claims are heavily supported by social bots. These results suggests that curbing social bots may be an effective strategy for mitigating the spread of online misinformation.

Shivam B. Parikh et all in Media-Rich Fake News Detection: A Survey 2018, Fake News has been around for decades and with the advent of social media and modern day journalism at its peak, detection of media-rich fake news has been a popular topic in the research community. Given the challenges associated with detecting fake news research problem, researchers around the globe are trying to understand the basic characteristics of the problem statement. This paper aims to present an insight on characterization of news story in the modern diaspora combined with the differential content types of news story and its impact on readers. Subsequently, we dive into existing fake news detection approaches that are heavily based on text-based analysis, and also describe popular fake news data-sets. We conclude the paper by identifying 4 key open research challenges that can guide future research.

Stefan Helmstetter Heiko Paulheim in Weakly supervised learning for fake news detection on Twitter 2018, the problem of automatic detection of fake news in social media, e.g., on Twitter, has recently drawn some attention. Although, from a technical perspective, it can be regarded as a straight-forward, binary classification problem, and the major challenge is the collection of large enough training corpora, since manual annotation of tweets as fake or non-fake news is an expensive and tedious endeavor. In this paper, we discuss a weakly supervised approach, which automatically collects a large-scale, but very noisy training dataset comprising hundreds of thousands of tweets. During collection, we automatically label tweets by their source, i.e., trustworthy or untrustworthy source, and train a classifier on this dataset. We then use that classifier for a different classification target, i.e., the classification of fake and non-fake tweets. Although the labels are not accurate according to the new classification target (not all tweets by an untrustworthy source need to be fake news, and vice versa), we show that despite this unclean inaccurate dataset, it is possible to detect fake news with an F1 score of up to 0.9.

Kai Shu et all in Fake News Detection in Social Media 2018, Fake news and hoaxes have been there since before the advent of the Internet. The widely accepted definition of Internet fake news is: fictitious articles deliberately fabricated to deceive readers". Social media and news outlets publish fake news to increase readership or as part of psychological warfare. In general, the goal is profiting through clickbaits. Clickbaits lure users and entice curiosity with flashy headlines or designs to click links to increase advertisements revenues. This exposition analyzes the prevalence of fake news in light of the advances in communication made possible by the emergence of social networking sites. The purpose of the work is to come up with a solution that can be utilized by users to detect and filter out sites containing false and misleading information. We use simple and carefully selected features of the title and post to accurately identify fake posts. The

experimental results show a 99.4% accuracy using logistic classifier.

Della Vedova et all in Automatic Online Fake News Detection Combining Content and Social Signals 2018, The proliferation and rapid diffusion of fake news on the Internet highlight the need of automatic hoax detection systems. In the context of social networks, machine learning (ML) methods can be used for this purpose. Fake news detection strategies are traditionally either based on content analysis (i.e. analyzing the content of the news) or - more recently - on social context models, such as mapping the news' diffusion pattern. In this paper, we first propose a novel ML fake news detection method which, by combining news content and social context features, outperforms existing methods in the literature, increasing their already high accuracy by up to 4.8%. Second, we implement our method within a Facebook Messenger chatbot and validate it with a real-world application, obtaining a fake news detection accuracy of 81.7%.

Leonie Haiden et all in Definitional Challenges of Fake News 2018, This paper considers the question of how "fake news" and "disinformation" have been defined and how this, in turn, has impacted research that seeks to better understand and counter the spreading of false or misleading information online. The paper identifies a tendency in political discourse and policy making to focus on addressing disinformation, i.e. false information spread intentionally, and a neglect of the problem of misinformation (false information spread unintentionally). On the other hand, academic research struggles to distinguish between mis- and disinformation. So far it has only marginally addressed the issue of intent and, instead, categorized information based on a true/false dichotomy.2 The challenges created by this disconnect between academic research output and policymaking obstruct our ability to effectively counter the negative effects of mis- and disinformation.

CHAPTER - 3

METHODOLOGY

SYSTEM ARCHITECTURE DIGRAM

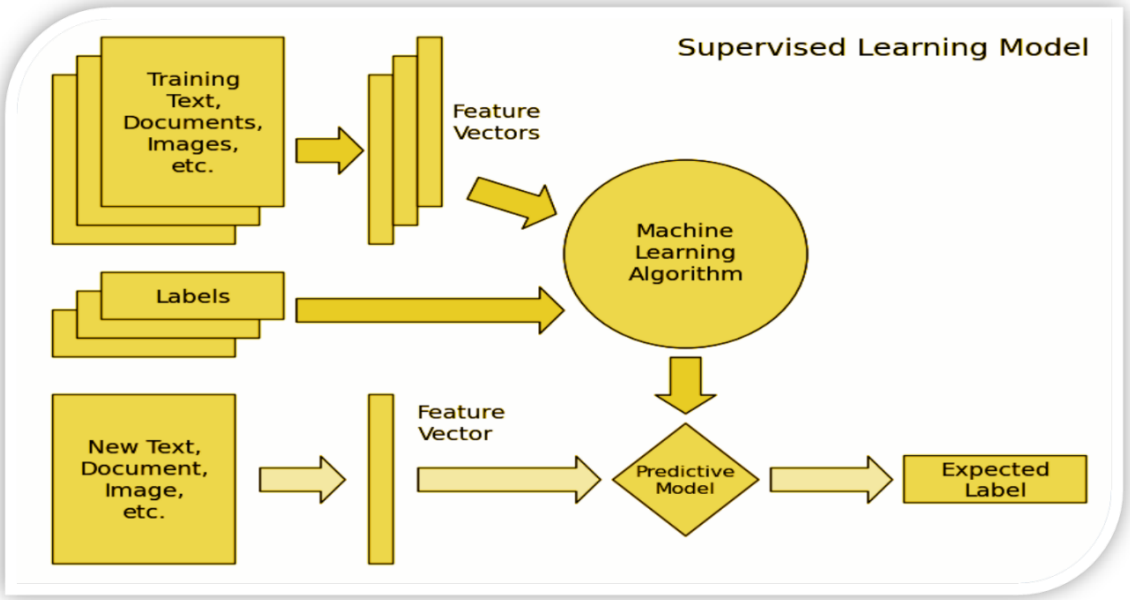


Fig 3.1 System architecture

MODULES IMPLEMENTATION

- Data collection.
- Preprocessing data.
- Feature extraction.
- Training the classifier.

DATA COLLECTION

Collecting data allows you to capture a record of past events so that we can use data analysis to find recurring patterns. From those patterns, you build predictive models using machine learning algorithms that look for trends and

predict future changes. Predictive models are only as good as the data from which they are

built, so good data collection practices are crucial to developing high-performing models. The data need to be error-free (garbage in, garbage out) and contain relevant information for the task at hand. For example, a loan default model would not benefit from tiger population sizes but could benefit from gas prices over time. Data collection is the process of gathering and measuring information from countless different sources. In order to use the data we collect to develop practical artificial intelligence (AI) and machine learning solutions, it must be collected and stored in a way that makes sense for the business problem at hand. So in this project we are using different packages and to load and read the data set we are using pandas. By using pandas, we can read the .csv file and then we can display the shape of the dataset with that we can also display the dataset in the correct form. We will be training and testing the data, when we use supervised learning it means we are labeling the data. By getting the testing and training data and labels we can perform different machine learning algorithms but before performing the predictions and accuracies, the data is need to be preprocessing i.e. the null values which are not readable are required to be removed from the data set and the data is required to be converted into vectors by normalizing and tokening the data so that it could be understood by the machine. Next step is by using this data, getting the visual reports, which we will get by using the Mat Plot Library of Python and Sickit Learn. This library helps us in getting the results in the form of histograms, pie charts or bar charts.

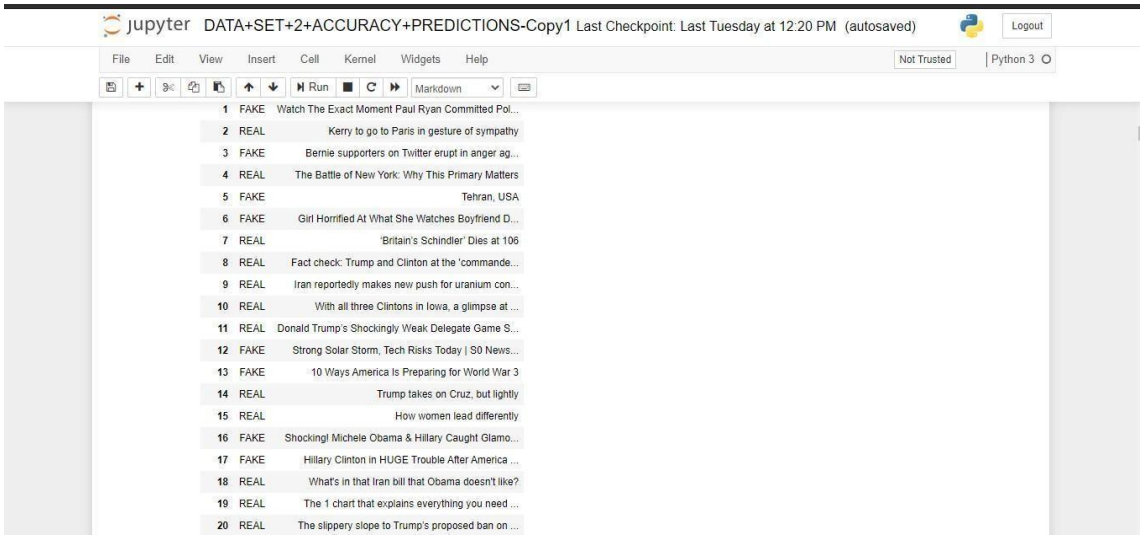


Fig 3.2.1 Data collection

PREPROCESSING DATA

Data preprocessing in Machine Learning is a crucial step that helps enhance the quality of data to promote the extraction of meaningful insights from the data. Data preprocessing in Machine Learning refers to the technique of preparing (cleaning and organizing) the raw data to make it suitable for a building and training Machine Learning models. Data preprocessing is an integral step in Machine Learning as the quality of data and the useful information that can be derived from it directly affects the ability of our model to learn; therefore, it is extremely important that we preprocess our data before feeding it into our model. In simple words, data preprocessing in Machine Learning is a data mining technique that transforms raw data into an understandable and readable format. When it comes to creating a Machine Learning model, data preprocessing is the first step marking the initiation of the process. Typically, real-world data is incomplete, inconsistent, inaccurate (contains errors or outliers), and often lacks specific attribute values/trends. This is where data preprocessing enters the scenario – it helps to clean, format, and organize the raw data, thereby making it ready-to-go for Machine Learning models. The data set used is split into a training set and a testing set containing in Dataset 1 -3256 training data and 814 testing data and in Dataset II- 1882 training data and 471 testing data respectively. Cleaning the data is always the first step. In this, those words are removed from the dataset. That helps in mining the useful information. Whenever we collect data online, it sometimes contains the undesirable characters like stop words, digits etc. which creates hindrance while spam detection. It helps in removing the texts which are language independent entities and integrate the logic which can improve the accuracy of the identification task.

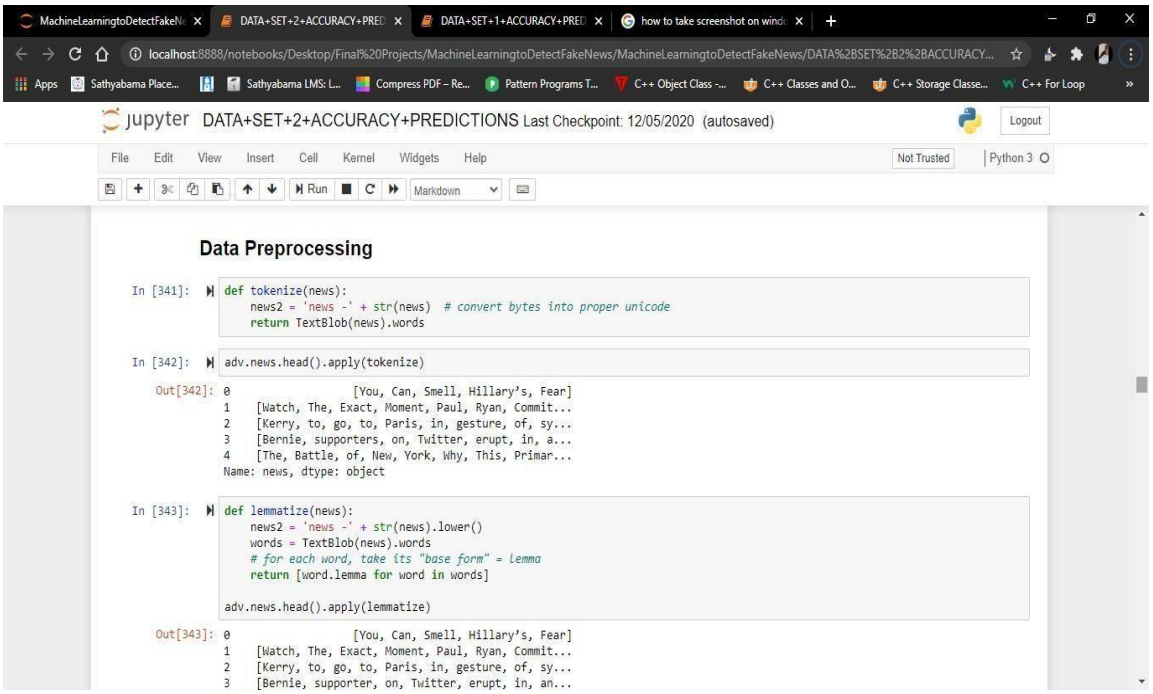


Fig 3.2.2 Preprocessing data

FEATURE EXTRACTION

Feature Extraction aims to reduce the number of features in a dataset by creating new features from the existing ones (and then discarding the original features). These new reduced set of features should then be able to summarize most of the information contained in the original set of features. In this way, a summarized version of the original features can be created from a combination of the original set. Feature extraction is the process of selecting a subset of relevant features for use in model construction. Feature extraction methods helps in to create an accurate predictive model. They help in selecting features that will give better accuracy. When the input data to an algorithm is too large to be handled and its supposed to be redundant then the input data will be transformed into a reduced illustration set of features also named feature vectors. Altering the input data to perform the desired task using this reduced representation instead of the full-size input. Feature extraction is performed on raw data prior to applying any machine learning algorithm, on the transformed data in feature space.

TRAINING THE CLASSIFIER

Classification is the process of predicting the class of given data points. Classes are sometimes called as targets/ labels or categories. Classification predictive modeling is the task of approximating a mapping function (f) from input variables (X) to discrete output variables (y). For example, spam detection in email service providers can be identified as a classification problem. This is s binary classification since there are only 2 classes as spam and not spam. A classifier utilizes some training data to understand how given input variables relate to the class. In this case, known spam and non-spam emails have to be used as the training data. When the classifier is trained accurately, it can be used to detect an unknown email. Classification belongs to the category of supervised learning where the targets also provided with the input data. There are many applications in classification in many domains such as in credit approval, medical diagnosis, target marketing etc. As In this project I am using Scikit-Learn Machine learning library for implementing the architecture. Scikit Learn is an open source python Machine Learning library which comes bundled in 3rd distribution anaconda. This just needs importing the packages and you can compile the command as soon as you write it. If the command doesn't run, we can get the error at the same time. I am using 4 different algorithms and I have trained these 4 models i.e. Naïve Bayes, Support Vector Machine, K Nearest Neighbors and Logistic Regression which are very popular methods for document classification problem. Once the classifiers are trained, we can check the performance of the models on test-set. We can extract the word count vector for each mail in test-set and predict it class with the trained models.

TECHNOLOGY USED

- PYTHON
- PYTHON LIBRARY

    PANDAS
    SCIKIT LEARN
    IPYTHON – JUPYTER(NOTEBOOK)
    MATPLOTLIB

- NATURAL LANGUAGE PROCESSING (NLP)
- MACHINE LEARNING – ALGORITHMS.

3.3.1 PYTHON

Python is an interpreter, high-level and general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object- oriented and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library. Guido van Rossum began working on Python in the late 1980's, as a successor to the ABC programming language, and first released it in 1991 as Python 0.9.1. Python 2.0 was released in 2000 and introduced new features, such as list comprehensions and a garbage collection system using reference counting and was discontinued with version
2.7.18 in 2020. Python 3.0 was released in 2008 and was a major revision of the language that is not completely backward-compatible and much Python 2 code does not run unmodified on Python 3.Python consistently ranks as one of the most popular programming languages.

NATURAL LANGUAGE PROCESSING (NLP)

Natural Language Processing, usually shortened as NLP, is a branch of artificial intelligence that deals with the interaction between computers and humans using the natural language. The ultimate objective of NLP is to read, decipher, understand, and make sense of the human languages in a manner that is valuable. Most NLP techniques rely on machine learning to derive meaning from human languages. NLP entails applying algorithms to identify and extract the natural language rules such that the unstructured language data is converted into a form that computers can understand. When the text has been provided, the computer will utilize algorithms to extract meaning associated with every sentence and collect the essential data from them. Sometimes, the computer may fail to understand the meaning of a sentence well, leading to obscure results.

Natural Language Processing is the driving force behind the following common applications:

- Language translation applications such as Google Translate.
- Word Processors such as Microsoft Word and Grammarly that employ NLP to check grammatical accuracy of texts.
- Interactive Voice Response (IVR) applications used in call centers to respond to certain users' requests.
- Personal assistant applications such as OK Google, Siri, Cortana, and Alexa.

MACHINE LEARNING – ALGORITHMS

..

NAV*I*E BAYES ALGORITHM

Naïve Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems. It is mainly used in text classification that includes a high-dimensional training dataset.

Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions. It is a probabilistic classifier, which means it predicts on the basis of the probability of an object. Some popular examples of Naïve Bayes Algorithm are spam filtration, Sentimental analysis, and classifying articles.

The Naïve Bayes algorithm is comprised of two words Naïve and Bayes, Which can be described as:

Naïve: It is called Naïve because it assumes that the occurrence of a certain feature is independent of the occurrence of other features. Such as if the fruit is identified on the bases of color, shape, and taste, then red, spherical, and sweet fruit is recognized as an apple. Hence each feature individually contributes to identify that it is an apple without depending on each other.

Bayes: It is called Bayes because it depends on the principle of Bayes' Theorem.

Bayes theorem: Bayes' theorem is also known as Bayes' Rule or Bayes' law, which is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability.

The formula for Bayes' theorem is given as: $P(A|B)=\frac{P(B|A)\,P(A)}{P(B)}$

Where, P(A|B) is Posterior probability: Probability of hypothesis A on the observed event B.

P(B|A) is Likelihood probability: Probability of the evidence given that the probability of a hypothesis is true.

P(A) is Prior Probability: Probability of hypothesis before observing the evidence.
P(B) is Marginal Probability: Probability of Evidence.

LOGISTIC REGRESSION

Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas Logistic regression is used for solving the classification problems.In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc.Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets. Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification. The below image is showing the logistic function:

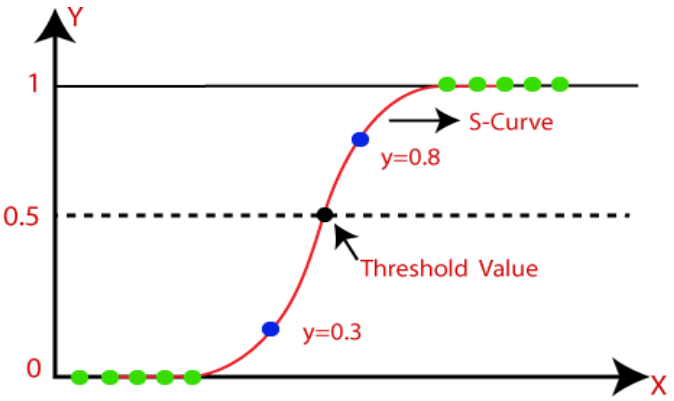


Fig 3.3.3.2 logistic function.

Logistic Function (Sigmoid Function) :

- The sigmoid function is a mathematical function used to map the predicted values to probabilities.
- It maps any real value into another value within a range of 0 and 1.
- The value of the logistic regression must be between 0 and 1, which cannot go beyond this limit, so it forms a curve like the "S" form. The S-form curve is called the Sigmoid function or the logistic function.
- In logistic regression, we use the concept of the threshold value, which defines the probability of either 0 or 1. Such as values above the threshold value tends to 1, and a value below the threshold values tends to 0.

K-Nearest Neighbor(KNN) ALGORITHM

In statistics, the k-nearest neighbors algorithm (k-NN) is a non-parametric classification method first developed by Evelyn Fix and Joseph Hodges in 1951,and later expanded by Thomas Cover. It is used for classification and regression. In both cases, the input consists of the k closest training examples in data set.

The output depends on whether k-NN is used for classification or regression:

In k-NN classification, the output is a class membership. An object is classified by

a plurality vote of its neighbors, with the object being assigned to the class most

common among its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of that single nearest neighbor. In k-NN regression, the output is the property value for the object. This value is the average of the values of k nearest neighbors.

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K-NN algorithm.K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data.It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.Example: Suppose, we have an image of a creature that looks similar to cat and dog, but we want to know either it is a cat or dog. So for this identification, we can use the KNN algorithm, as it works on a similarity measure. Our KNN model will find the similar features of the new data set to the cats and dogs images and based on the most similar features it will put it in either cat or dog category.

Suppose there are two categories, i.e., Category A and Category B, and we have a new data point x1, so this data point will lie in which of these categories. To solve this type of problem, we need a K-NN algorithm. With the help of K-NN, we can easily identify the category or class of a particular dataset.

The K-NN working can be explained on the basis of the below algorithm:

Step-1: Select the number K of the neighbors

Step-2: Calculate the Euclidean distance of K number of neighbors

Step-3: Take the K nearest neighbors as per the calculated Euclidean distance.

Step-4: Among these k neighbors, count the number of the data points in each category.

Step-5: Assign the new data points to that category for which the number of the neighbor is maximum.

Step-6: Our model is ready.

SUPPORT VECTOR MACHINE (SVM) ALGORITHM

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane. In machine learning, support-vector machines (SVMs, also support- vector networks) are supervised learning models with associated learning algorithms that analyze data for classification and regression analysis. Developed at AT&T Bell Laboratories by Vladimir Vapnik with colleagues (Boser et al., 1992, Guyon et al., 1993, Vapnik et al., 1997), SVMs are one of the most robust prediction methods, being based on statistical learning frameworks or VC theory proposed by Vapnik and Chervonenkis (1974) and Vapnik (1982, 1995). Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting). An SVM maps training examples to points in space so as to maximise the width of the gap between the two categories. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two

different categories that are classified using a decision boundary or hyperplane:

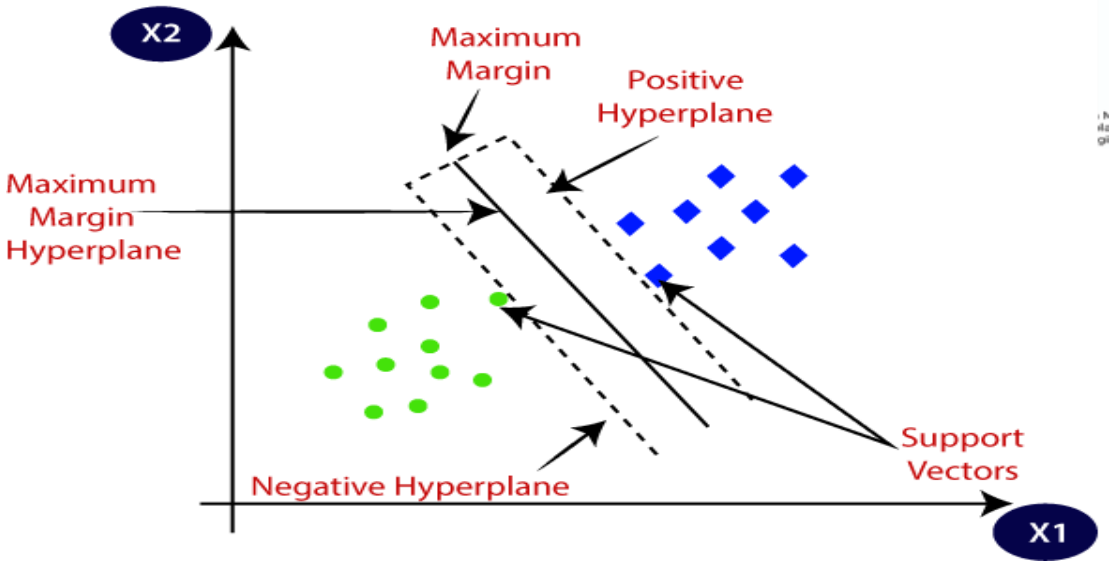


Fig 3.3.3.4 decision boundary

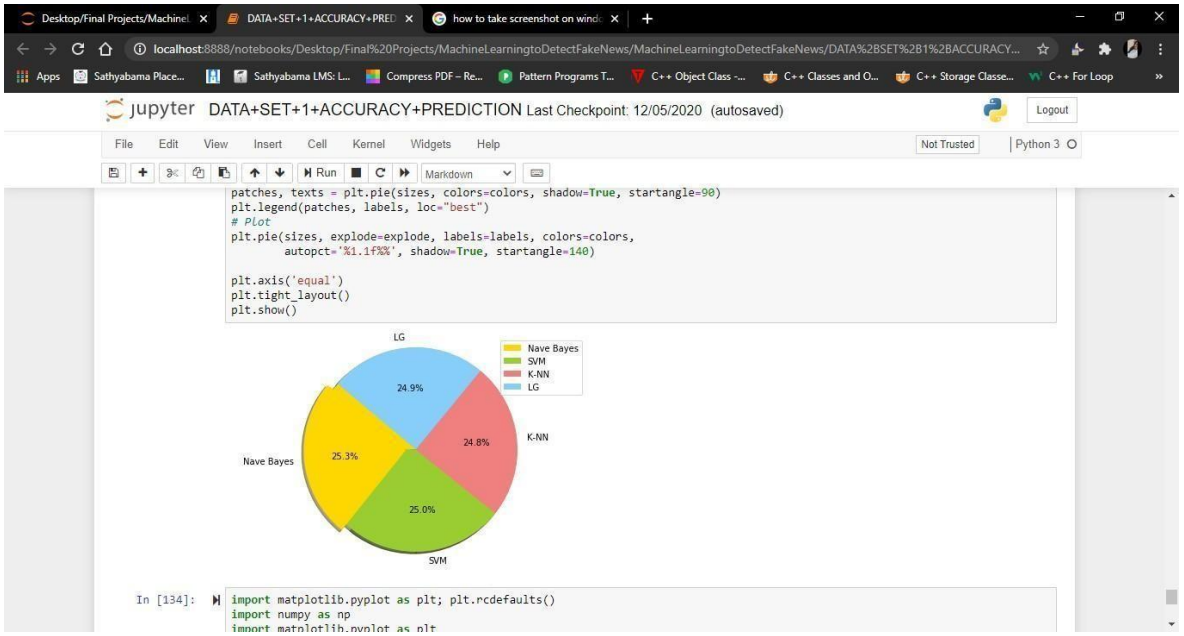
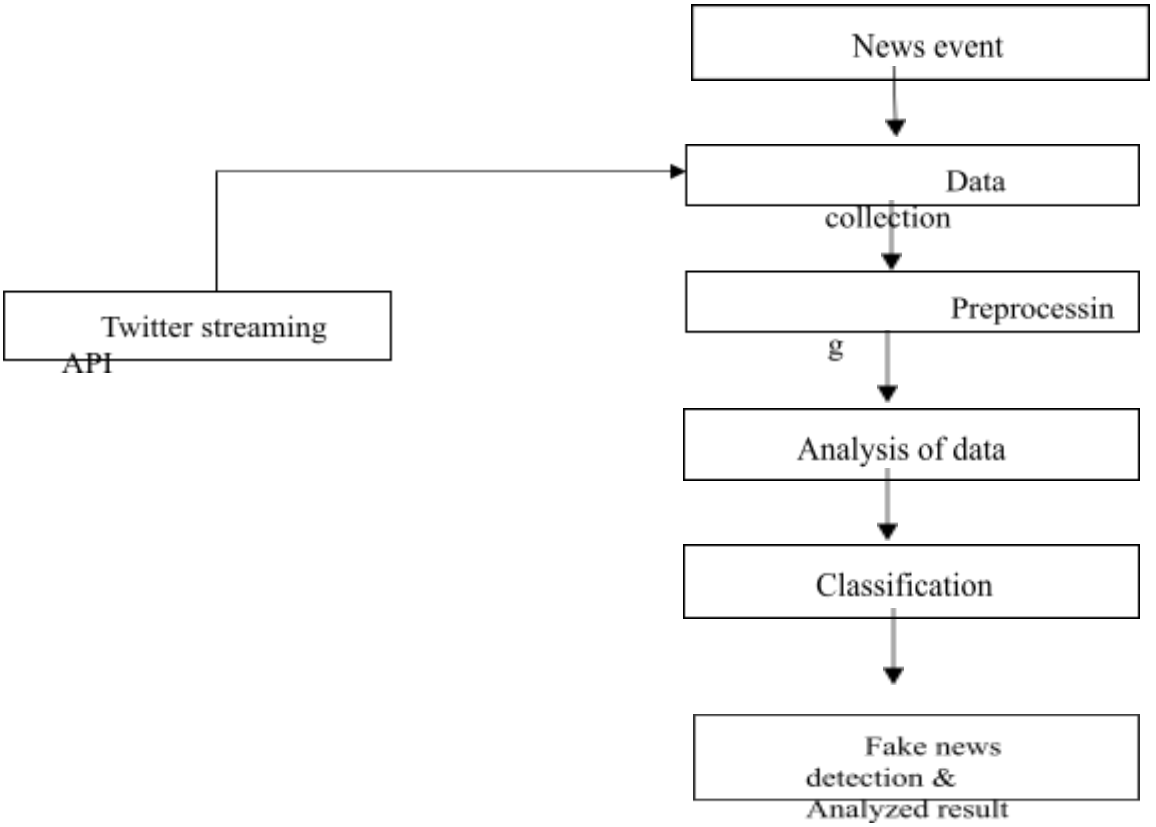


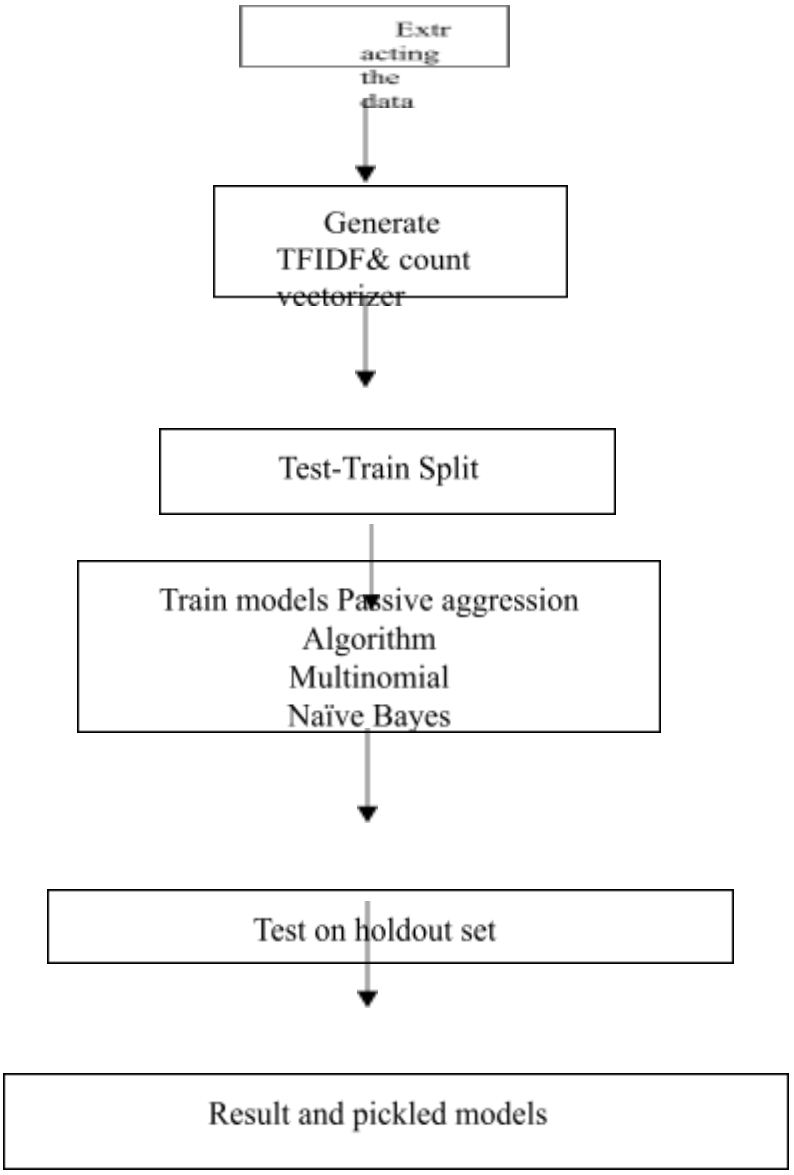
Fig 3.3.3 algorithm comparison

IMPLEMENTATION

- Importing packages such as pandas, numpy, csv, and importing algorithms from sklearn as we are using the Python Library.
- Reading the CSV file and displaying the data set.
- Remove the erroneous null and nan values.
- Calculating the length of news so as to check the length of fake and real news.
- Data Preprocessing - NLTK tokenization
  - Tokenizing and normalizing the text
  - Lemmatization and Vectorization
- Feature Extraction
- Training a model to detect fake
- Build and evaluate
- Evaluate Accuracies

DATA FLOW


News event
Data collection
Preprocessing
Analysis of data
Classification
Fake news detection & Analyzed result
Twitter streaming API


Extr
acting
the
data
Generate
TFIDF& count
vectorizer
Test-Train Split
Train models Passive aggression
Algorithm
Multinomial
Naïve Bayes
Test on holdout set
Result and pickled models

CHAPTER – 4
RESULT AND DISCUSSION

4.1 RESULT AND DISCUSSION

Algorithm's accuracy depends on the type and size of your dataset. More the data, more chances of getting correct accuracy. Machine learning depends on the variations and relations .Understanding what is predictable is as important as trying to predict it.While making algorithm choice, speed should be a consideration factor. The project highlights the results of the Application and the snapshots for each of the activities are shown along with the discussion of each activity describing it's working. Each snapshots describes every single step of making Fake news detection

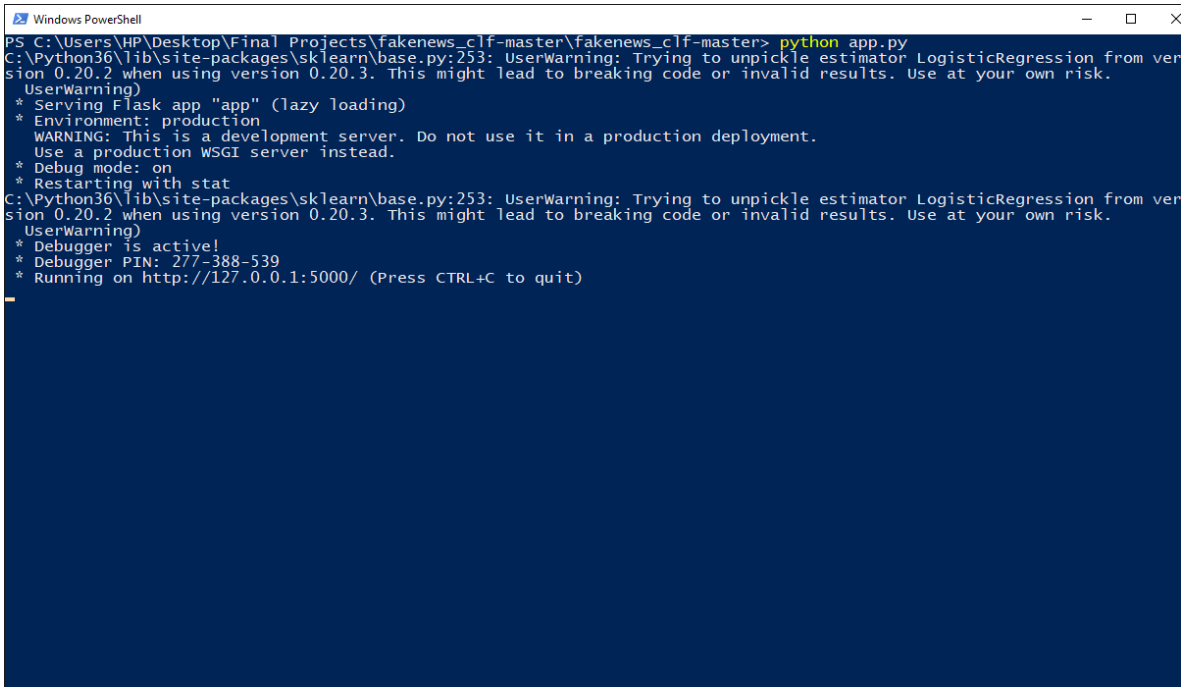


Fig 4.1: Microsoft power shell

Figure 4.1 Here we open the Microsoft power shell in order to obtain the link for

the page where we check each article is fake or real news.

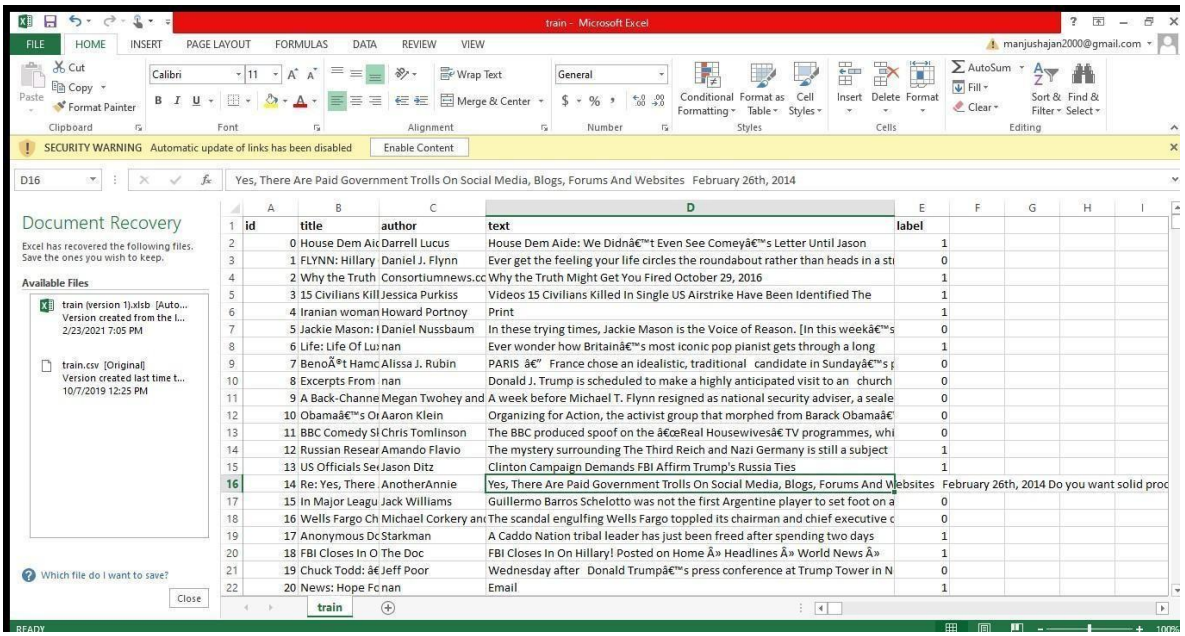


Fig 4.2: train dataset

Figure 4.2 This is the trained dataset where both the fake news data and real news data are stored and trained, here the fake news comes under 1 in the label whereas the real news as 0.

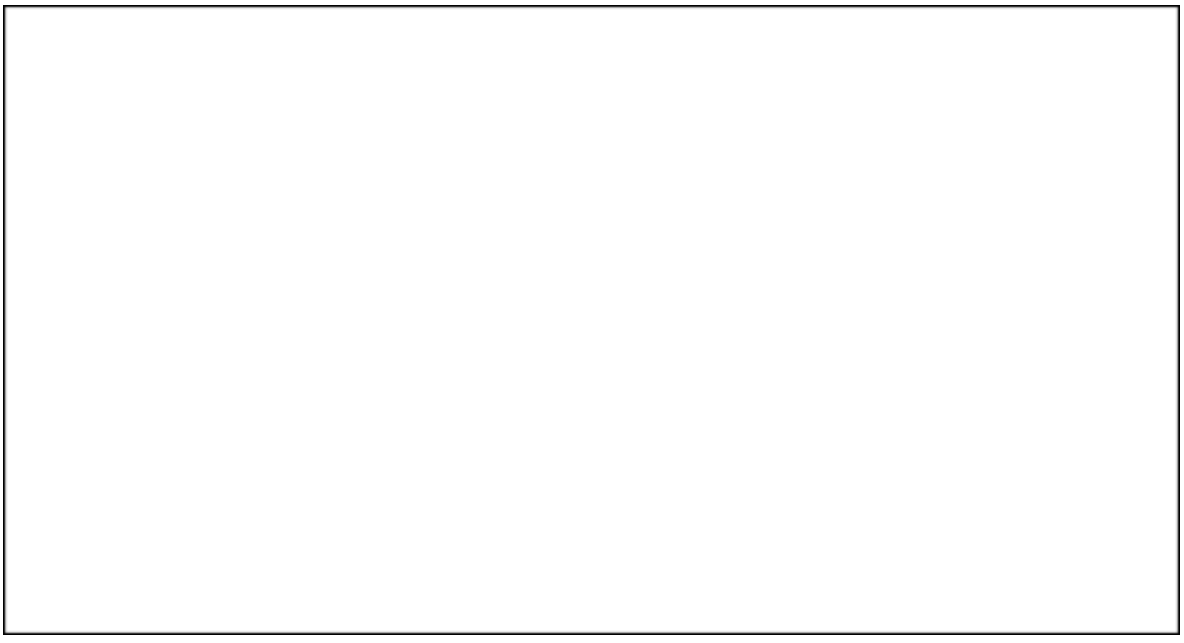


Fig 4.3: Webpage

Fig 4.3 this is the webpage where we enter the article to check the accuracies.

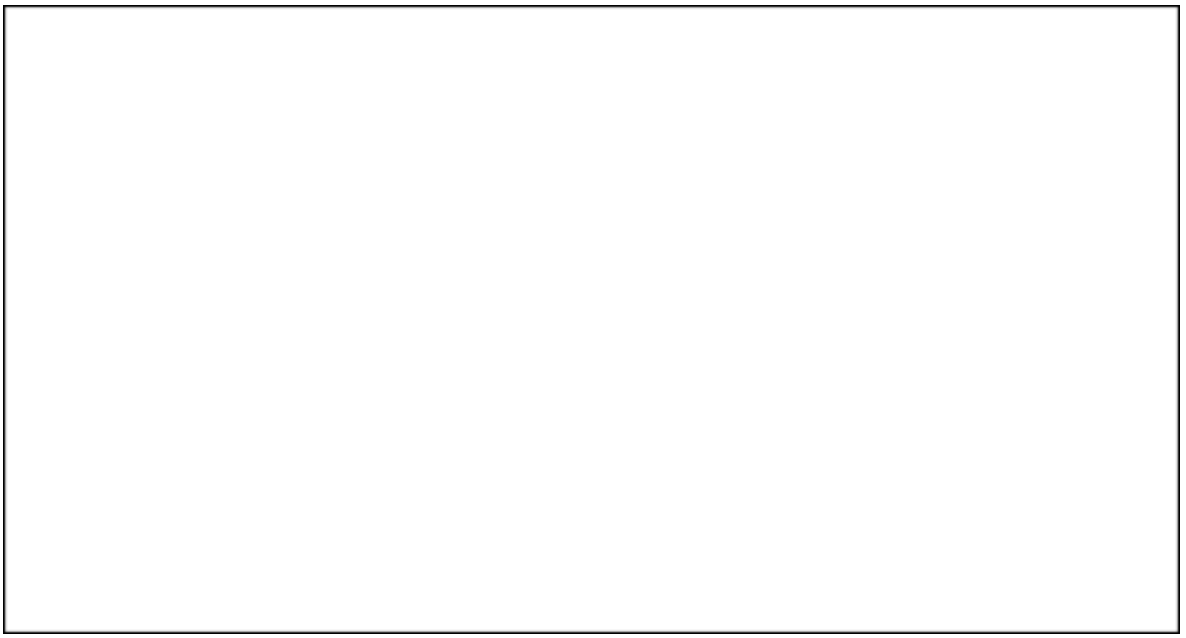


Fig 4.4 entering the article

Figure 4.4 here we enter the article to check whether it is fake or real news with accuracies.

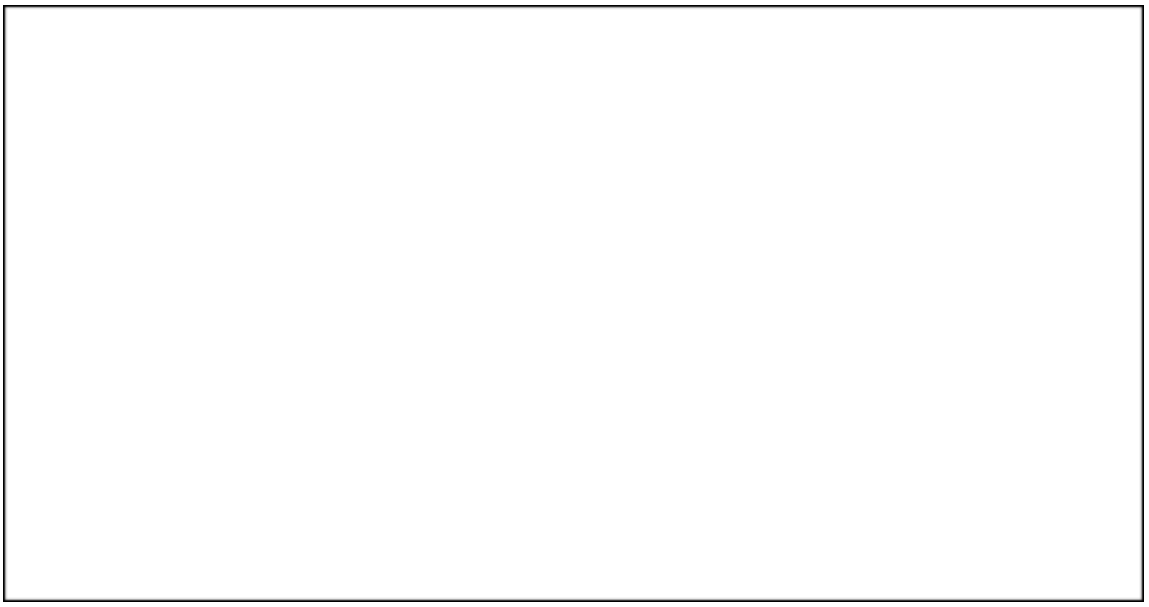


Fig 4.5 output of checked article

Figure 4.5 here the output of an article is shown, showing that the given article is 100% truthful.

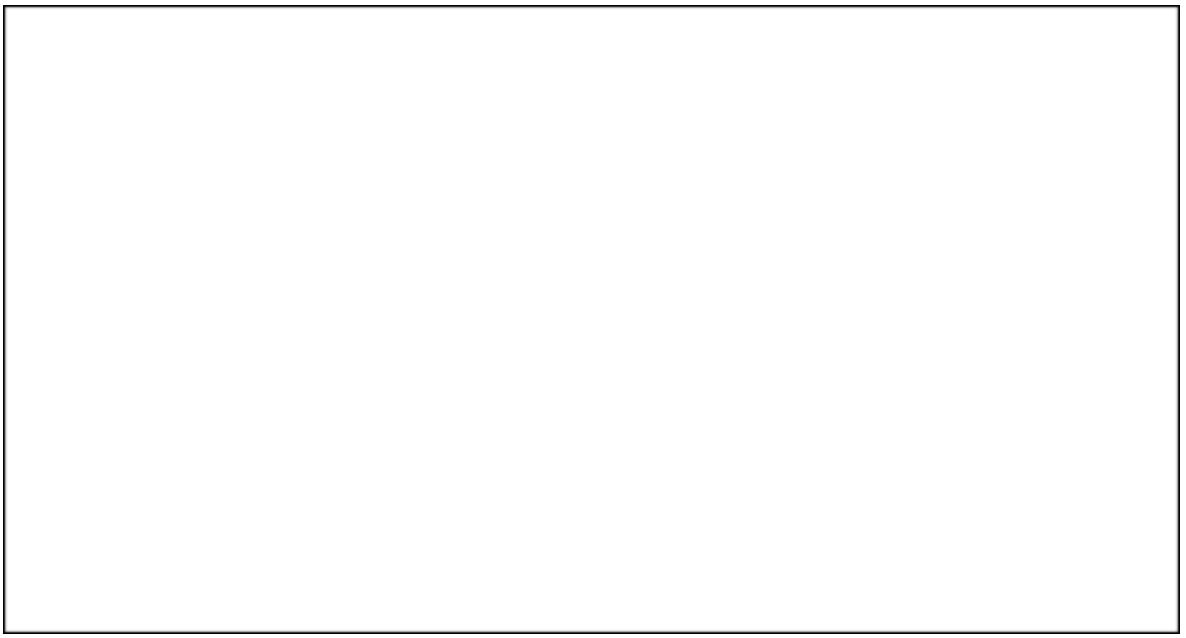


Fig 4.6: output of checked article

Figure 4.6 here the output of an article is shown, showing that the given article is 98.93% fake news or an opinion piece.

CHAPTER- 5
CONCLUSION

CONCLUSION

Many people consume news from social media instead of traditional news media. However, social media has also been used to spread fake news, which has negative impacts on individual people and society. In this paper, an innovative model for fake news detection using machine learning algorithms has been presented. This model takes news events as an input and based on twitter reviews and classification algorithms it predicts the percentage of news being fake or real.

REFERENCES

[1]. Chen, Y., Conroy, N. J., & Rubin, V. L. (2015, November). Misleading online content: Recognizing clickbait as false news. In Proceedings of the 2015 ACM on Workshop on Multimodal Deception Detection (pp. 15-19). ACM.

[2]. Conroy, N. J., Rubin, V. L., & Chen, Y. (2015, November). Automatic deception detection: Methods for finding fake news. In Proceedings of the 78th ASIS&T Annual Meeting: Information Science with Impact: Research in and for the Community (p. 82). American Society for Information Science.

[3]. Della Vedova, M. L., Tacchini, E., Moret, S., Ballarin, G., DiPierro, M., & de Alfaro, L. (2018, May). Automatic Online Fake News Detection Combining Content and Social Signals. In 2018 22nd Conference of Open Innovations Association (FRUCT) (pp. 272-279). IEEE.

[4]. Haiden, L., & Althuis, J. (2018). The Definitional Challenges of Fake News.

[5]. Helmstetter, S., & Paulheim, H. (2018, August). Weakly supervised learning for fake news detection on Twitter. In 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM) (pp. 274-277). IEEE.

[6]. Najafabadi, M. M., Villanustre, F., Khoshgoftaar, T. M., Seliya, N., Wald, R., & Muharemagic, E. (2015). Deep learning applications and challenges in big data analytics. Journal of Big Data, 2(1), 1.

[7]. Parikh, S. B., & Atrey, P. K. (2018, April). Media-Rich Fake News Detection: A Survey. In 2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR) (pp. 436-441). IEEE.

[8]. Shao, C., Ciampaglia, G. L., Varol, O., Flammini, A., & Menczer, F. (2017). The spread of fake news by social bots. arXiv preprint arXiv:1707.07592, 96-104.

[9]. Stahl, K. (2018). Fake News Detection in Social Media.

[10] Tacchini, E., Ballarin, G., Della Vedova, M. L., Moret, S., & de Alfaro, L. (2017). Some like it hoax: Automated fake news detection in social networks. arXiv preprint arXiv:1704.07506.

https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm
https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning

APPENDIX

A.SAMPLE CODE

```
import pandas as pd
import numpy as np
import csv
import sys
import scipy
from sklearn import naive_bayes
from sklearn import svm
from sklearn.feature_extraction.text import CountVectorizer, TfidfTransformer
from sklearn.naive_bayes import MultinomialNB
from sklearn.svm import SVC
from sklearn.metrics import classification_report, f1_score, accuracy_score, confusion_matrix
from sklearn.pipeline import Pipeline
from sklearn.grid_search import GridSearchCV
from sklearn.cross_validation import StratifiedKFold, cross_val_score, train_test_split
from sklearn.learning_curve import learning_curve
from pandas.tools.plotting import scatter_matrix
import matplotlib.pyplot as plt
%matplotlib inline
from sklearn import model_selection
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
from textblob import TextBlob
from textblob import TextBlob, Word, Blobber
from textblob.classifiers import NaiveBayesClassifier
```

```
from textblob.taggers import NLTKTagger
```

```
adv=pd.read_csv('fakerealnews.csv')
adv.news.value_counts()
adv.describe()
adv.groupby('label').describe()
adv[adv.news.notnull()]
adv[adv.label.notnull()]
adv=adv[pd.notnull(adv['news'])]
adv=adv[pd.notnull(adv['label'])]
adv.isnull()
adv['length']=adv['news'].map(lambda text: len(text))
adv.head(30)
adv.length.plot( bins=20, kind='hist')
adv.hist(column='length', by='label', bins=50)
def tokenize(news):
    news2 = 'news -' + str(news) # convert bytes into proper unicode
    return TextBlob(news).words
adv.news.head().apply(tokenize)
def lemmatize(news):
    news2 = 'news -' + str(news).lower()
    words = TextBlob(news).words
    # for each word, take its "base form" = lemma
    return [word.lemma for word in words]
TextBlob("Strong Solar Storm, Tech Risks Today").tags TextBlob("What's
in that Iran bill that Obama doesn't like?").tags
bow_transformer=CountVectorizer(analyzer=lemmatize).fit(adv['news'])
len(bow_transformer.vocabulary_)
adv.news.head().apply(lemmatize)
news4=adv['news'][160]
news4
bow4 = bow_transformer.transform([news4])
bow4
bow4.shape
```

```
bow_transformer.get_feature_names()[665]
news_bow = bow_transformer.transform(adv['news'])
```

```
'sparse matrix shape:', news_bow.shape
'number of non-zeros:', news_bow.nnz
tfidf_transformer = TfidfTransformer().fit(news_bow)
tfidf4 = tfidf_transformer.transform(bow4)
tfidf4
tfidf_transformer.idf_[bow_transformer.vocabulary_['u']]
news_tfidf = tfidf_transformer.transform(news_bow)
news_tfidf.shape
%time spam_detector = MultinomialNB().fit(news_tfidf, adv['label'])
spam_detector=MultinomialNB().fit(news_tfidf, adv['label'])
spam_detector
'predicted:', spam_detector.predict(tfidf4)[0]
'expected:', adv.label[55]
all_predictions = spam_detector.predict(news_tfidf)
all_predictions
'accuracy', accuracy_score(adv['label'],  all_predictions)
'confusion matrix\n', confusion_matrix(adv['label'], all_predictions)

'(row=expected, col=predicted)'
plt.matshow(confusion_matrix(adv['label'],    all_predictions),    cmap=plt.cm.binary,
interpolation='nearest')
plt.title('confusion matrix')
plt.colorbar()
plt.ylabel('expected label')
plt.xlabel('predicted label')
print (classification_report(adv['label'], all_predictions))
```

B.OUTPUT SCREEN SHOTS

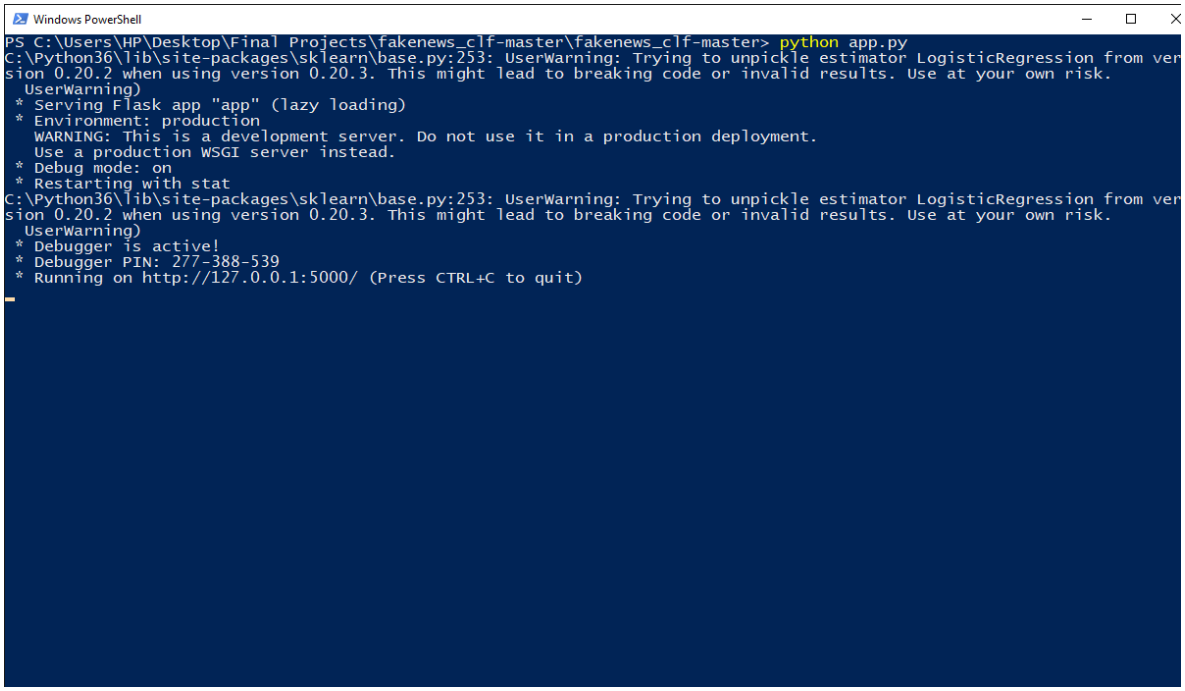


Fig no B.1: Microsoft power shell

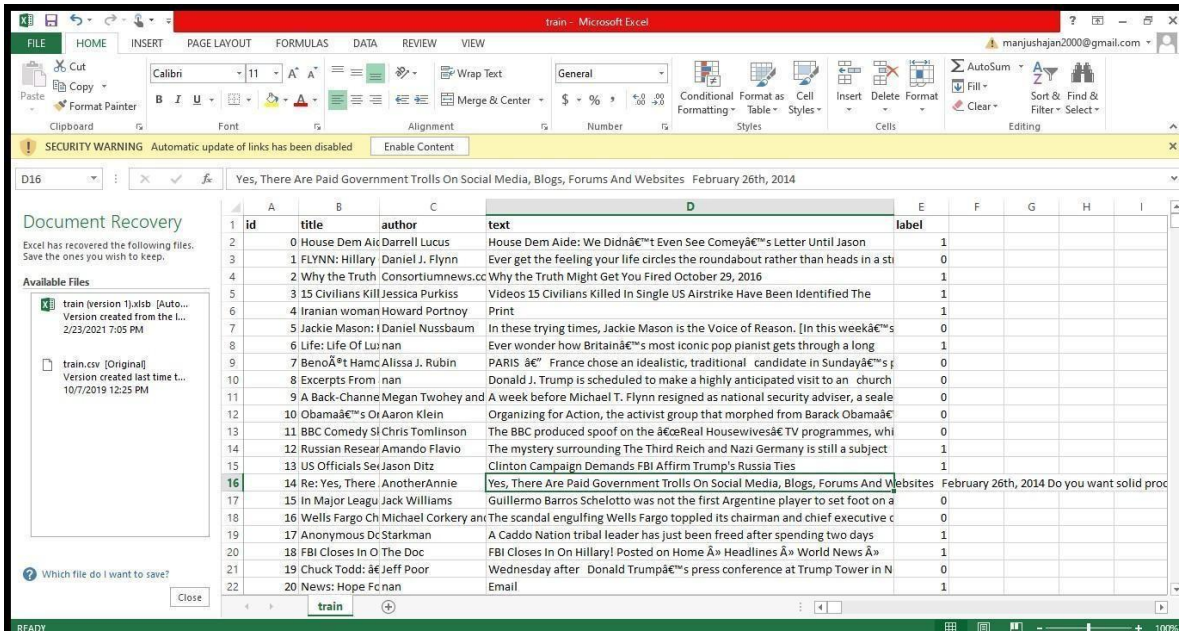


Fig no B.2 : train dataset

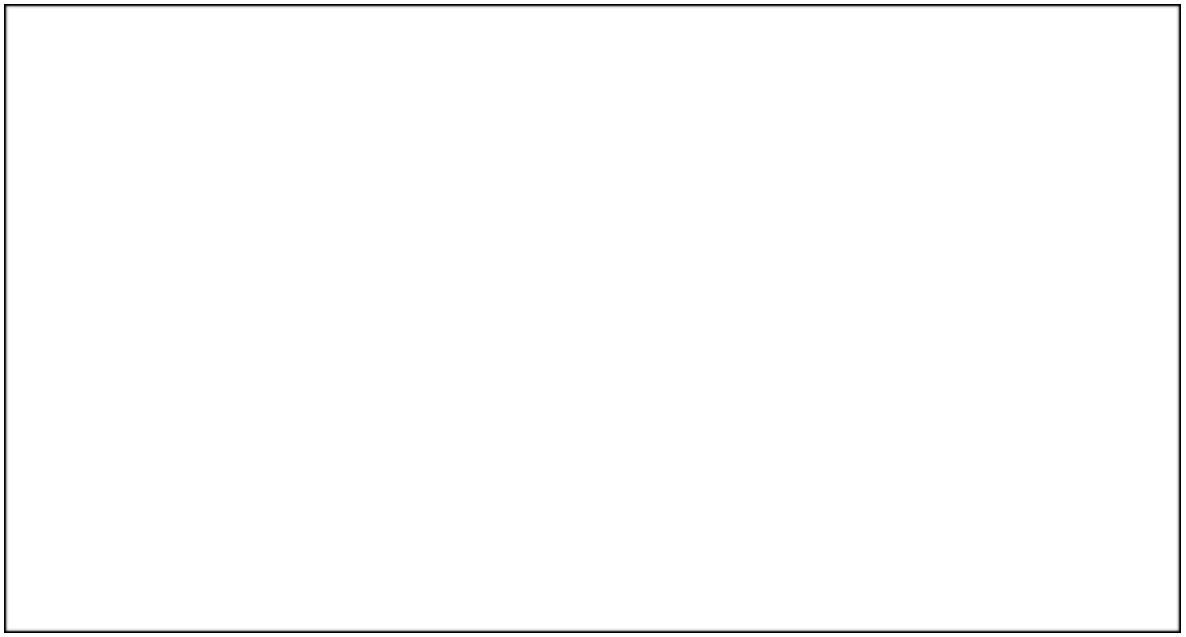


Fig no B.3: Webpage

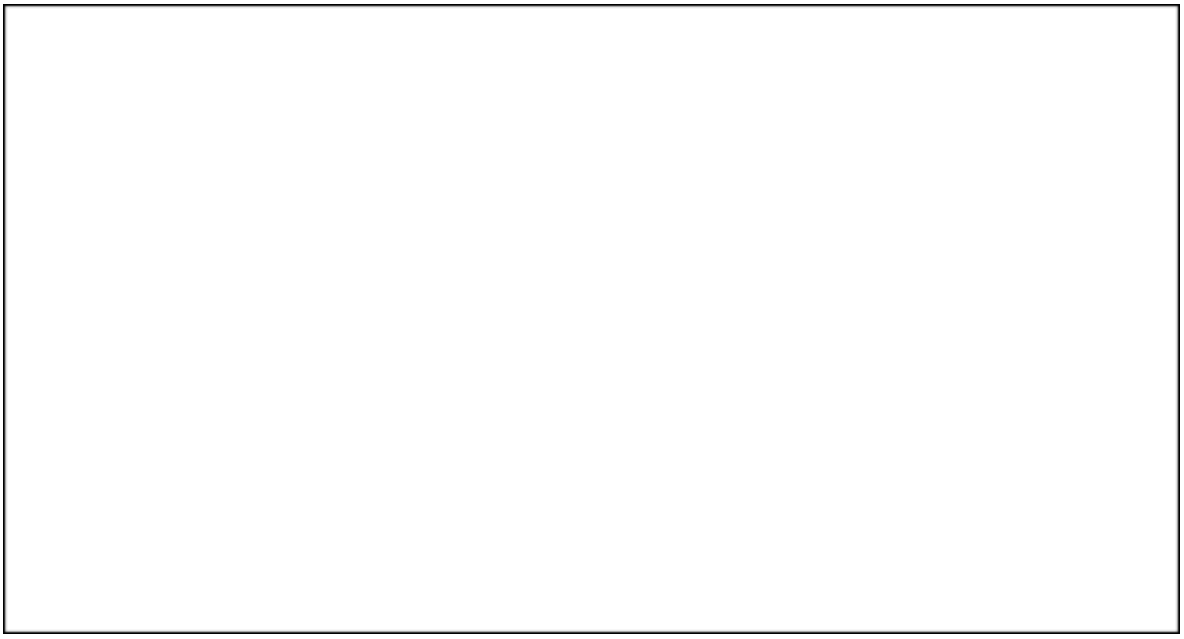


Fig no B.4: entering the article

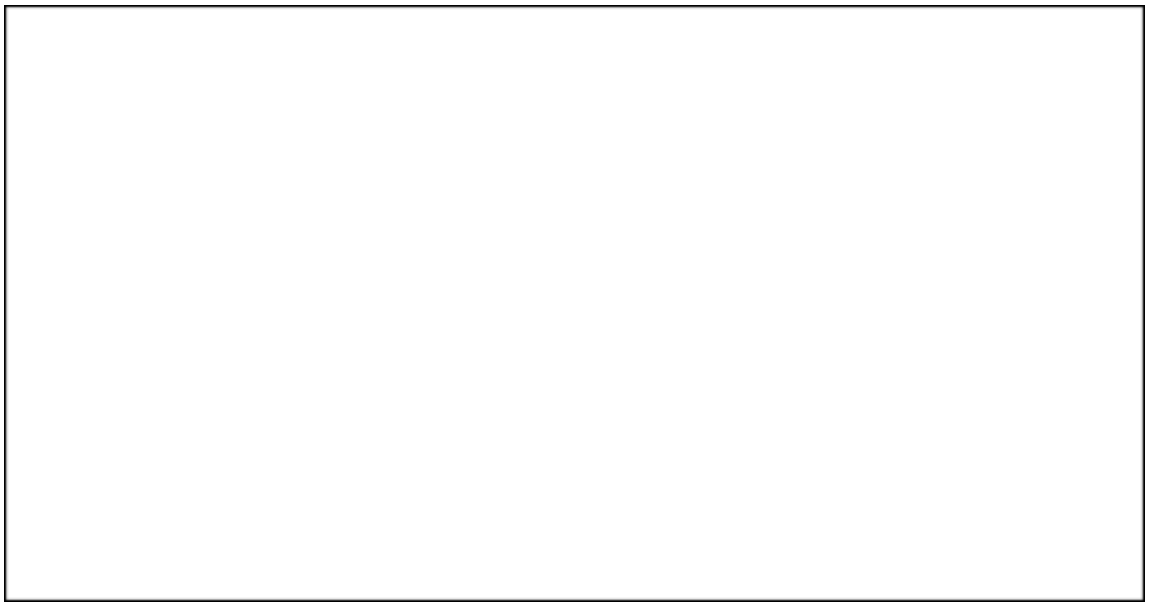


Fig B.5 output of checked article

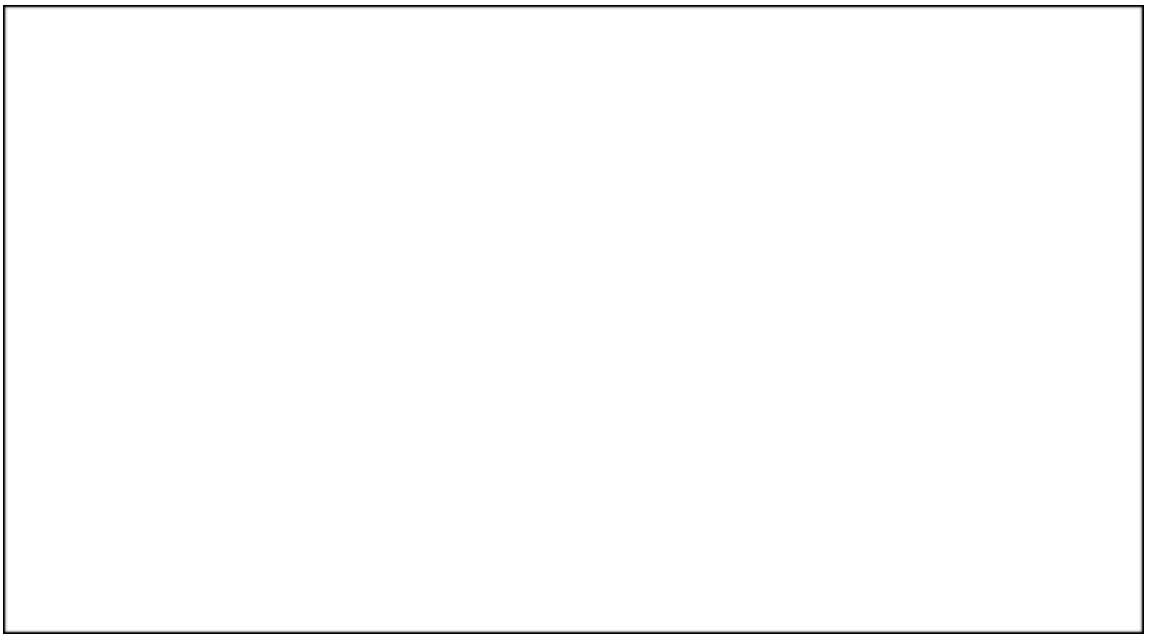
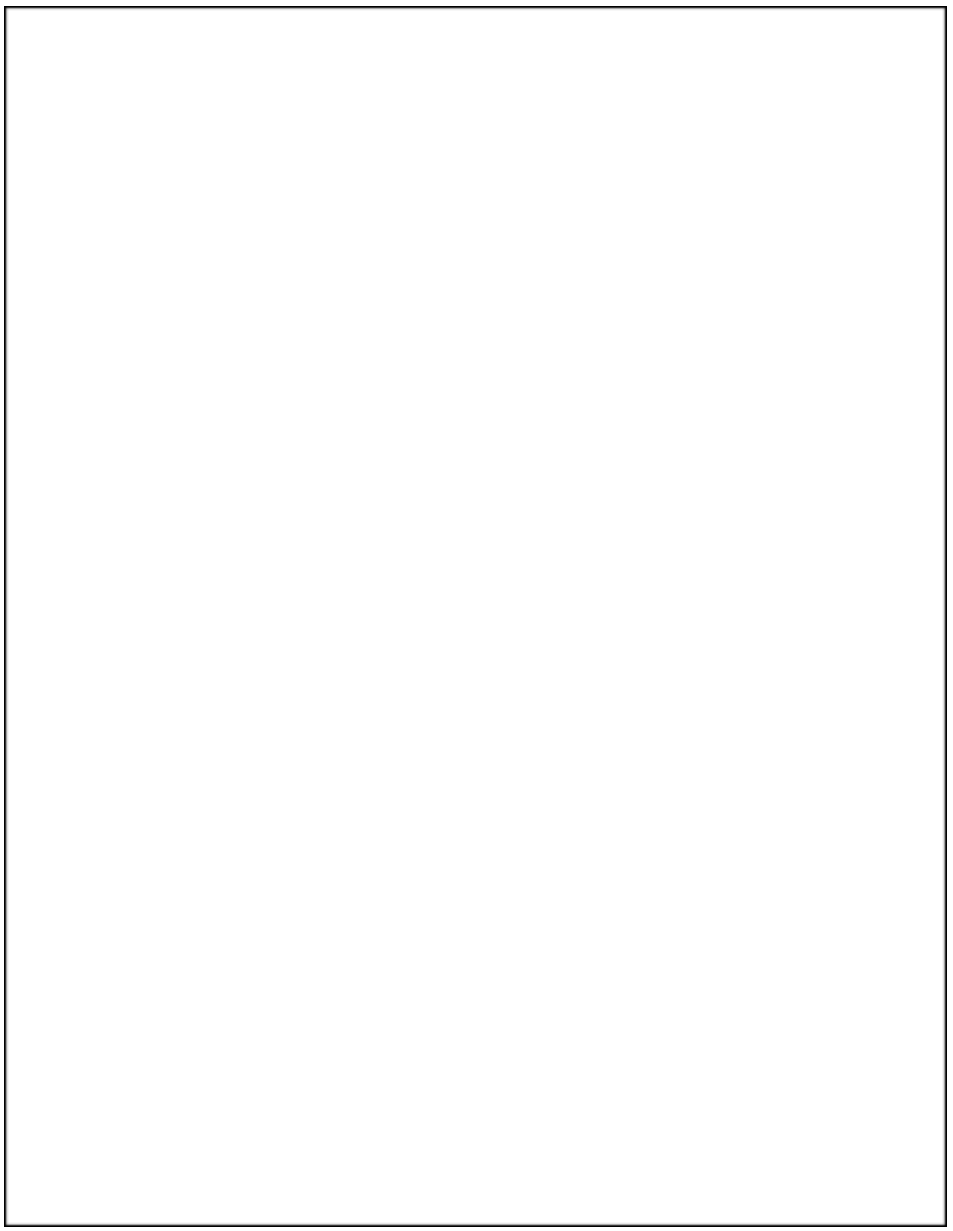


Fig B.6 no: output of checked article

C.PALGIRISM REPORT



FAKE NEWS DETECTION

ORIGINALITY REPORT

5%	1%	5%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Shivam B. Parikh, Pradeep K. Atrey. "Media-Rich Fake News Detection: A Survey", 2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), 2018 Publication	3%
2	"Machine Learning Algorithms based Skin Disease Detection", International Journal of Innovative Technology and Exploring Engineering, 2019 Publication	1%
3	Submitted to Indian Institute of Technology, Madras Student Paper	1%
4	Syed Ishfaq Manzoor, Jimmy Singla, Nikita. "Fake News Detection Using Machine Learning approaches: A systematic Review", 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), 2019 Publication	<1%
5	"10th International Conference on Theory and	<1%

	Application of Soft Computing, Computing with Words and Perceptions - ICSCCW-2019", Springer Science and Business Media LLC, 2020 Publication	
6	"Conference Proceedings of ICDLAIR2019", Springer Science and Business Media LLC, 2021 Publication	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		