Industrial Engineering Journal
ISSN: 0970-2555
Volume: 52, Issue 4, April 2023

FAKE NEWS DETECTION

Mr.S.Sudheer Kumar[1], Ms.V.Gayathri[2] , Mr.P.Sai Kumar[3],Mr.G.Nithin[4], Dr.V.Sreerama Murthy[5]

1. BTECH, NADIMPALLI SATYANARAYANA RAJU INSTITUTE OF TECHNOLOGY, SONTYAM, VISAKHAPATNAM, ANDHRA PRADESH, INDIA - 531173

2. BTECH, NADIMPALLI SATYANARAYANA RAJU INSTITUTE OF TECHNOLOGY , SONTYAM, VISAKHAPATNAM, ANDHRA PRADESH, INDIA - 531173

3. BTECH, NADIMPALLI SATYANARAYANA RAJU INSTITUTE OF TECHNOLOGY , SONTYAM, VISAKHAPATNAM, ANDHRA PRADESH, INDIA - 531173

4. BTECH, NADIMPALLI SATYANARAYANA RAJU INSTITUTE OF TECHNOLOGY , SONTYAM, VISAKHAPATNAM, ANDHRA PRADESH, INDIA - 531173

5. PROFESSOR COMPUTER SCIENCE AND ENGINEERING, NADIMPALLI SATYANARAYANA RAJU INSTITUTE OF TECHNOLOGY, SONTYAM, VISAKHAPATNAM, ANDHRA PRADESH, INDIA-531173

ABSTRACT

Every day, the internet publishes more than millions of news articles. This number will rise significantly if we include the tweets from Twitter. The internet is now the primary medium for disseminating false information. It is necessary to have a system in place to recognise fake news that is posted online so that readers can be cautioned. By evaluating the news' text data using machine learning techniques, several researchers have developed ways to spot false news. Here, we'll also talk about machine learning methods for accurately identifying bogus news. To determine if a piece of news is true or false, we will train machine learning classifiers. We're going to train three well-known To forecast bogus news, classification techniques Logistics Regression, Support Vector Classifier, and Naive-Bayes were used. We shall determine which of these three algorithms performs the task the best after analysing the performance of all three. Before we can organise fake news detectors, we must first introduce the datasets that contain both actual and fake news. To categorise the datasets, we employ NLP, machine learning, and deep learning algorithms. By incorporating fake news categorization and current machine learning algorithms, we produce a thorough audit of detecting fake news. We attempt to determine whether the new News being entered by the end user in our site is a false one or a legitimate one using these training data sets as a reference.

1 INTRODUCTION

Every day, the internet publishes more than millions of news articles. This number will rise significantly if we incorporate the tweets from Twitter. The internet is now the primary medium for disseminating false information. It is necessary to have a system in place to recognise fake news that is posted online so that readers can be cautioned. By evaluating the news' text data using machine learning techniques, several researchers have developed ways to spot false news. Here, we'll also talk about machine learning methods for accurately identifying bogus news.

To determine if a piece of news is true or false, we will train machine learning classifiers. We're going to train three well-known To forecast bogus news, classification techniques Logistics Regression, Support Vector Classifier, and Naive-Bayes were used. We shall determine which of these three algorithms performs the task the best after analysing the performance of all three.

Before we can organise fake news detectors, we must first introduce the datasets that contain both actual and fake news. To categorise the datasets, we employ NLP, machine learning, and deep learning algorithms. By incorporating fake news categorization and current machine learning algorithms, we produce a thorough audit of detecting fake news.

We attempt to determine whether the new news being entered by the end user on our site is a false one or a legitimate one using these training data sets as a reference.

Machine learning techniques for fraud detection have been the subject of extensive research, with the majority of it concentrating on categorising online reviews and publicly accessible social media posts. The issue of identifying "fake news" has received a lot of attention in the literature, especially since late 2016 during the American Presidential election. A number of strategies are outlined by Conroy, Rubin, and Chen [9] with the purpose of accurately classifying the deceptive articles. They point out that shallow parts-of-speech (POS) tagging and simple content-related n-grams have frequently failed to account for crucial context information, making them insufficient for the classification task. These techniques have only been shown effective when used in conjunction with more sophisticated analytical techniques. Syntax Deep It has been demonstrated that analysis utilising Probabilistic Context Free Grammars (PCFG) is very useful when combined with n-gram techniques. Using online review corpora, Feng, Banerjee, and Choi [10] are able to classify cases of fraud with 85%-91% accuracy. On top of Feng's first deep syntax model, Feng and Hirst added a semantic analysis that checks 'object:descriptor' pairings for inconsistencies with the text for further advancement. With similar effectiveness, Rubin, Lukoianova, and Tatiana use a vector space model to examine rhetorical structure. A pre-existing knowledge base is necessary for the linguistic pattern similarity networks used by Ciampaglia et al.

2. LITERATURE SURVEY AND RELATED WORK

2.1 LITERATURE REVIEW

Some studies have looked into using the Passive Aggressive algorithm to identify bogus news. A approach that combines the Passive Aggressive algorithm and sentiment analysis for false news detection was proposed in a study named "Fake News Detection using Passive Aggressive Classifier with Sentiment Analysis" (S. M. Abdullah Al Mamun, et al., 2021). The study used a dataset of 3999 news articles and found that the suggested strategy had a 95.32% accuracy rate.

Passive Aggressive algorithm and bigram features were utilised in another paper titled "Fake news detection using passive aggressive algorithm and bigram features" (S. P. Sreeja and R. J. Megan a then, 2020) to identify bogus news. The accuracy of the proposed technique was 94.2% in the study, which used a dataset of 4025 news articles. A hybrid strategy that combines Passive Aggressive algorithm with multi-lingual approach is proposed in the study "Fake News Detection Using Hybrid Passive Aggressive Algorithm with Multi-Lingual Approach" (V. G. Nandhini and S. Sathish Kumar, 2021). The study's proposed strategy has a 95.7% accuracy rate when applied to a dataset of 2619 news articles in three different languages.

.

2.2 QUALITY OF ASSESSMENT OF CONTENT POSTED ON OSM

This section presents the research work done in the space of extracting and analyzing trustworthy and credible information from Twitter during real world events. One major challenge in consuming content from Twitter is that it is difficult to filter out good quality content from the large volume of content created. The quality of content on Twitter is polluted with the presence of phishing, spam, advertisements, fake images, rumours and inflammatory content. Media such as Twitter, which is a micro-blog is more suited for dissemination and sharing news based information, since it is mostly public, and gives a bigger range of audience for the content posted. Hence, majority of the work discussed in this survey, is centred around Twitter. Researchers have used various classical computational techniques such as classification, ranking, characterization and conducting user studies, to study the problem of trust on Twitter. Some of the researchers who applied various kinds of classifiers (Naive Bayes,
Decision Tree, SV) to identify spam, phishing and not credible content on Twitter, using message, user, network and topic based features on Twitter [1, 2, 3]. Ranking algorithms have been applied and fine tuned by researchers for questions pertaining to trust related problems such as credibility and spam [4, 5]. Each of the above mentioned work are discussed in detail, later in the chapter.

To research the issue of trust on Twitter, many methods, including classification, ranking, characterization, and user studies, have been used. Using message, user, network, and subject based data on Twitter, some researchers utilised different classifiers (Naive Bayes, Decision Tree, SV) to identify spam, phishing, and unreliable content on Twitter [1, 2, 3]. Researchers have used and improved ranking algorithms for queries relating to trust-related issues like credibility and spam [4, 5]. Later in the chapter, each of the aforementioned works is covered in detail.

2.3 EMERGENCE OF TWITTER AS A NEWS MEDIA

According to Kwak et al., 85% of topics discussed on Twitter are related to news [59], and the computer science research community has examined the usefulness of online social media, particularly Twitter, as a news disseminating agent. Their work highlighted the relationship between user specific parameters v/s the tweeting activity patterns, such as analysis of the number of followers and followers v/s the tweeting (re-tweeting) numbers. Twitter news event-related tweets can be mapped using the energy function [1,6]. The suggested methods function as novel event detection algorithms. The study examined 900 news stories from 2010 to 2011. Online social media activity related to news stories was analysed qualitatively and quantitatively by Castillo et al. [1,7,8]. They came to the conclusion that in-depth articles garnered fewer consistent social media reactions than news pieces detailing breaking news occurrences.

2.4 SPAM AND PHISHING DETECTION

For the accuracy of the information on Twitter, spam, malware, phishing assaults, and compromised accounts are a serious worry. Researchers have investigated and put forth numerous practical strategies to get rid of these problems. It has been discovered that URL shortener services like bit.ly are used to disguise the origin of phishing URLs, making it challenging to identify them. Genuine consumers lose millions of dollars each year as a result of phishing scams on social networking sites like Twitter and Facebook. Researchers have created tools to detect phishing tweets on Twitter by utilising elements including tweet content and a Chrome Extension for real-time phishing detection. Most of the accounts used to transmit spam on Twitter have been identified, and spam distributed via URLs has been described. were legitimate user accounts that had been compromised. A rating system was suggested to lessen the power of spammers on the network, and link farming, in which Twitter accounts link to one another to promote their content, has also been investigated. A Criminal account Inference Algorithm (CIA) has been presented to find new cybercriminals on Twitter after an analysis of the ecosystem of these individuals and their supporters on Twitter. Using tools like URL searches, username pattern matching, and phrase recognition, machine learning approaches have also been used to locate spammers on Twitter. These studies offer crucial insights into the problems relating to the information quality on Twitter and suggest practical remedies to address them.

2.5 TRUST AND CREDIBILITY ASSESSMENT

This section focuses on research that has been conducted to evaluate and analyze trust and credibility of content on online social media platforms (OSM). One such study is Truthy, which was developed by Ratkiewicz et al. to examine the spread of information on Twitter and calculate a trustworthiness score for a public stream of micro-blogging updates related to an event to identify political smears, astroturfing, misinformation, and other forms of social pollution. The researchers presented instances of abusive behavior by Twitter users in their work, and Truthy is a live web service built upon this research. Machine learning techniques have also been employed by researchers to identify credible and non-credible content on OSM. For instance, [1] Castillo et al. demonstrated that automated classification algorithms can be utilized to detect news topics from conversational topics and assess their credibility based on various Twitter features, achieving a precision and recall of 70-80% with the J48 decision tree classification algorithm. Their study included various types of features, such as message, user, topic, and propagation-based features, and they observed that tweets without URLs tended to be related to non-credible news, while tweets that included negative sentiment terms were linked to credible news. The researchers evaluated their results against human-annotated ground truth data.

2.6 USER MODELING ON ONLINE SOCIAL MEDIA

We showcase some of their research on user modelling approaches used to study user behaviour on social networks. Two parameters were used by Yin et al. to describe user behaviour: themes relevant to users' intrinsic interests and topics linked to the temporal context [17]. They created the Dynamic Temporal Context-Aware Mixture model, a latent class statistical mixture model (DTCAM). On the basis of four sizable social media datasets, they assessed their system. The authors showed how user modelling techniques might be successfully applied to enhance the functionality of social network recommender systems. To incorporate several aspects and model users' posting behaviour on Twitter, Xu et al. suggested a mixed latent topic model[16]. The authors made the assumption that the models' treatment of the data would have an impact on a user's behaviour.

Complexity of withheld content and created latent topic quality. Using more than 2 million tweets, Abel et al. created a user modelling framework for news suggestions on Twitter [15]. By using semantic enrichment and temporal factors, the authors suggested various methods for developing user profiles that are based on hashtags, entities, or topics. Their findings demonstrated that taking into account temporal profile patterns can enhance the quality of recommendations.

were legitimate user accounts that had been compromised. A rating system was suggested to lessen the power of spammers on the network, and link farming, in which Twitter accounts link to one another to promote their content, has also been investigated. An algorithm called the Criminal account Inference Algorithm (CIA) has been devised to identify the ecosystem of cybercriminals and their followers on Twitter. new Twitter criminals. Using tools like URL searches, username pattern matching, and phrase recognition, machine learning approaches have also been used to locate spammers on Twitter. These studies offer crucial insights into the problems relating to the information quality on Twitter and suggest practical remedies to address them.

2.7 SUMMARY

This chapter summarises the research that has been done so far on monitoring and interpreting reliable news sources, particularly during actual occurrences. It can be difficult to separate reliable information from spam, phishing, bogus photos, rumours, and inflammatory content on Twitter due to the volume of content that is posted there. The review of the literature finds that academics have used a range of computational tools to investigate the issue of trust on Twitter, including categorization, ranking, and user studies. To recognise spam, phishing, and unverified content, they have employed classifiers like Naive Bayes and Passive Aggressive algorithm. Scholars have also looked into how Twitter became a news source and contrasted the news subjects it covered with those from more conventional news outlets. to block phishing, malware, and spam. Also, they have created tools like Truthy. spam, malware, and phishing. Additionally, they have developed tools like Truthy to analyse the dissemination of information on Twitter

Industrial Engineering Journal
ISSN: 0970-2555
Volume: 52, Issue 4, April 2023

and determine the credibility of a public stream of microblogging updates on an event. It has also been researched how users behave and engage with social media websites through user modelling. Although there has been progress, there are still research gaps that must be filled in order to assess reliable information from Twitter during actual occurrences.

3 PROPOSED WORK AND ALGORITHM

A matrix (i.e., word tallies related to how often they are used in other articles in your dataset) can help. In this paper, a model is built based on the count vectorizer. Using a passive aggressive classifier, which is typical for text-based processing, will work well since this challenge involves text classification. The actual objective is to create a model for text transformation using the tfidf vectorizer, which involves selecting the type of raw document that will be transformed into the tfidf vectorizer matrix using an n-number of the most frequently used words and/or phrases, whether or not they are in lower case.
those terms in a given text sample that appear at least a certain number of times.

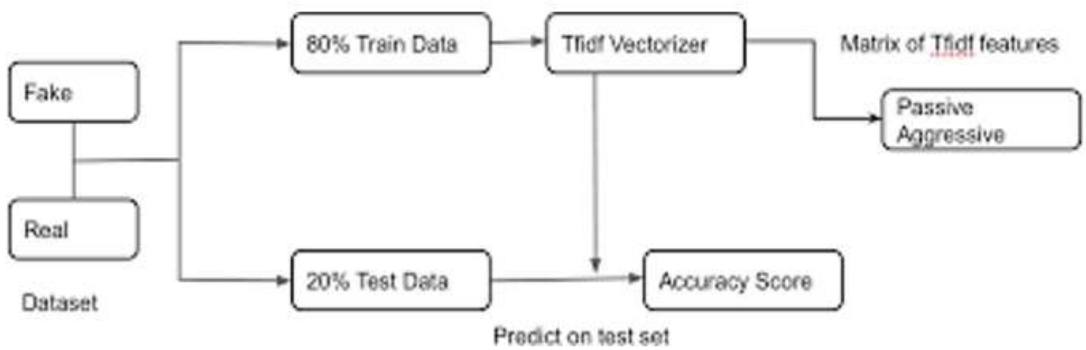

Fig : Architecture diagram

Structure of the System

The whole architecture of the software system is depicted in the architecture displayed below.

A. Data Splitting

We aim to divide the data sets into test data and training data after first collecting the data sets, reading them, and then dividing them again.
B. Observational Data Analysis

Once that is finished, we attempt to perform exploratory data analysis before attempting to apply preprocessing approaches to extract features from the datasets while also removing the null values.

We then begin extracting the features using certain NLP approaches once it is finished.
C. Data Preprocessing

The majority of social media data is informal communication with typos, slang, poor grammar, etc. Social media data is highly unstructured.

It is essential to boost performance and reliability as a result of the Create methods for utilising resources to help you make educated decisions. Before using the data for predictive modelling, the data must be cleaned in order to produce better insights. Basic pre-processing was applied to the News training data for this reason.
D. Cleansing of Data

We get data in either a structured or unstructured format while reading data. Unstructured data lacks a proper framework while a structured format has a clearly established pattern. We have a semi-structured format that sits between the two structures and is comparable to better structured than unstructured format. To draw attention to characteristics that we want our machine learning system to recognise, the text input must be cleaned.

E. Development of Features

Text data can be used to create a variety of features, including word count, the number of big words, the number of unique words, the frequency of n-grams, etc. We can provide computers the ability to comprehend text and do tasks like Clustering and Classification by developing representations of words that capture their meanings, semantic links, and many sorts of context they are used in.

4 METHODOLOGIES

Description of the dataset

A dataset is a group of data that has been arranged and shown in a structured manner. It can be applied to a variety of tasks, including machine learning, research, and analysis.

There are 7797 news stories total, both real and fraudulent, in the news.csv dataset that is accessible on Kaggle. Each row in the data represents a news article and has three values: title, text, and label. The data is organised in a CSV format. 3898 of the 7797 items have the designation "genuine news," whereas 3898 have the designation "false news." One can enter a news article's title or content into a fake news detection algorithm and compare the output with the relevant label in the dataset to determine if the article is authentic or fake.

A A dataset typically consists of a collection of observations, or separately acquired pieces of data. It is common to refer to each observation as a "data point." Depending on the kind of data being gathered, these data points can be organised in various ways.

A dataset may also contain metadata, which tells us things like the format, source, and collection method of the data.

It is crucial to take into account a dataset's size, quality, and the kinds of analyses that may be done on it when working with it. In order to ensure that the data is appropriate for analysis, it is also crucial to correctly clean and preprocess the data. Datasets come in a wide variety of formats, including structured data (such unstructured data (such as text data), semi-structured data (such as data in a database) (such as XML or JSON data).

In machine learning, one of the most challenging problems to solve has nothing to do with intricate computations. Getting the right datasets in the right organisation is the problem.

Pre-processing:

Stopword Elimination: Stop words, such as "the," "a," "an," "in," "on," etc., are often used terms in a language that don't have much significance. As they don't significantly add to the text's meaning, these words can be eliminated from the text data. The TfidfVectorizer is initialised in the code with the stop words argument set to "english," which eliminates stop words from the English language.

Inverted Document Frequency Term Frequency (TF-IDF) the vectorization The text input is transformed using TfidfVectorizer into a numerical format that may be utilised to train a machine learning model. By multiplying the term frequency—the number of times a word appears in a document—by the inverse of the number of documents in the dataset that contain the word, the TF-IDF formula determines the significance of each word in a document (inverse document frequency). This provides terms that frequently appear in one document but infrequently in another document more weight, indicating that they are more crucial for differentiating between papers. The code fits the TfidfVectorizer to the training data (tfidf.fit transform(x train)), and then transforms the training and testing data (tfidf train=tfidf.fit transform(x train) and tfidf test=tfidf.transform(x test)).

Max Document Frequency: Moreover The TfidfVectorizer eliminates words from the dataset that appear too frequently in addition to stop words. Using the max df option, which defines the maximum document frequency as a percentage of all the documents in the dataset, is how this is done. Words that appear in more than 70% of the publications will be deleted from the text data because the code specifies max df=0.7.

Overall, by reducing noise and complexity in the text input, these preprocessing processes make it simpler for a machine learning model to identify patterns and generate reliable predictions.

Separating the train and test:

1. Train-Test Split: The dataset is split into training and testing sets using the train test split function from sklearn.model

Industrial Engineering Journal
ISSN: 0970-2555
Volume: 52, Issue 4, April 2023

selection. This is carried out to assess the the machine learning model's performance with unknown data. The data is divided into training and testing sets using the function train test split(df['text'], lbl, test size=0.2, random state=42). The random state argument is used to guarantee that the same random split is produced each time the code is executed, and the test size argument provides the percentage of the dataset that should be utilised for testing (20%).

Prediction and Fitting: The classification model is fitted to the training data using the Passive Aggressive Classifier algorithm from sklearn.linear model. This algorithm is a linear classifier that modifies its parameters in reaction to instances that are incorrectly classified, making it especially beneficial for streaming data and online learning. The model is fitted using pac.fit(tfidf train,y train) in the code. data used in training. Using y pred=pac.predict(tfidf test), the model may be used to forecast the testing data once it has been trained.

3. Evaluation: Using the accuracy score and confusion matrix functions from sklearn.metrics, the machine learning model's performance is assessed. The confusion matrix function computes the number of true positive, true negative, false positive, and false negative predictions made by the model, whereas the accuracy score function computes the percentage of correct predictions generated by the model. The accuracy score is calculated in the code as score=accuracy score(y test,y pred), and the confusion matrix is calculated as confusion matrix(y test,y pred, labels=['FAKE','REAL']).

Overall, these testing and splitting stages aid in assessing how well a machine learning model performs on previously unexplored data and guarantee that the model is generalizable to new data.

Classification: Passive Aggressive Classifier for Detecting Fake News:

A form of machine learning algorithm called a passive-aggressive classifier is frequently employed for binary classification problems, where the objective is to determine which of two categories a given input belongs to. "Real news" and "fake news" could be the two categories in the context of fake news identification.

Iteratively modifying a weight vector based on each training example it sees is how the passive-aggressive classifier operates. For each case, the classifier predicts something and then determines whether or not that prediction was accurate. The weight vector stays the same if the forecast is accurate. The weight vector is updated in a way that attempts to minimise the loss between the models if the prediction is wrong, though. both the example's actual label and the projected label. This update is carried out in a "passive-aggressive" fashion, which means that it is aggressive enough to fix the error but passive enough to avoid overfitting to the training data.

We would first need to train the classifier on a sizable dataset of labelled news articles, where each piece is classified as either "real news" or "false news," in order to employ a passive-aggressive classifier for fake news identification. Based on the characteristics of the articles, such as the language used, the tone of the article, and the sources referenced, the classifier would subsequently learn to differentiate between the two categories.

We may use the classifier to forecast the label of a fresh news item once it has been trained. To extract the article's features, such as the words used and the tone of the writing, we would first need to preprocess it. The classifier would then process these data and produce a prediction of whether the article is "genuine news" or "fake news," depending on the features. Overall, the passive-aggressive classifier can be a useful tool for spotting fake news because it can accurately anticipate future articles and learn to tell the difference between authentic and fraudulent news articles based on their properties. Yet it's vital to remember that no classifier is flawless, and there might be times when it predicts things incorrectly. As a result, it's crucial to combine the classifier with other techniques for identifying fake news. such as the verification of sources and fact-checking.

5.RESULTS AND DISCUSSION SCREENSHOTS

In [2]:
```
df=pd.read_csv('news.csv')
df.shape
df.head()
```

Out[2]:

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

Fig : We reading a dataset by using pd.read_csv

Industrial Engineering Journal
ISSN: 0970-2555
Volume: 52, Issue 4, April 2023

Here we are reading news.csv file by using a pd.read_csv function. pd.read_csv('news.csv') reads the CSV file news.csv and assigns the resulting data to a pandas DataFrame called dataframe

Fig : Same amount Fake and real news

Here, We have taken the amount of ratio real and fakes and by taking this datasets, Our model objecr will predict whether the news is fake or real.

```
In [12]: #prediction
         y_pred=pac.predict(tfidf_test)
         score=accuracy_score(y_test,y_pred)
         print(f'Accuracy: {round(score*100,2)}%')

         Accuracy: 93.13%
```

Accuracy achieved by the model is approx 93%

Fig : Accuracy score

Here, after predicting the model object we are getting a 93% Accuracy Score

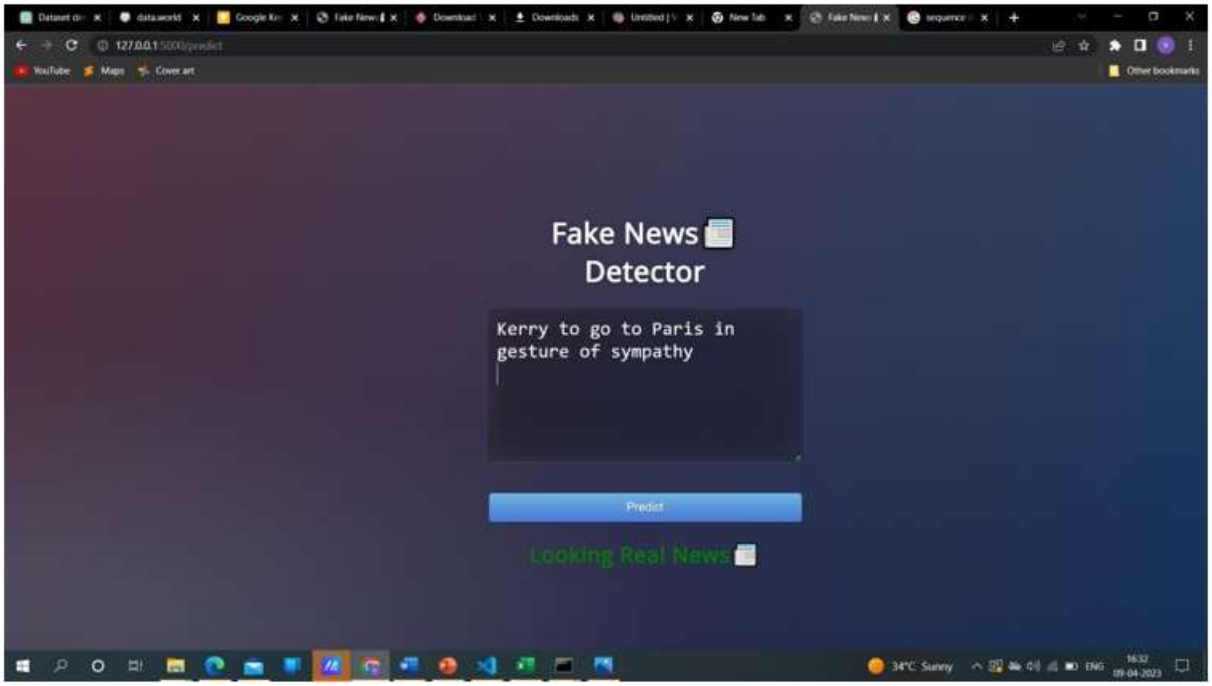

Fig : The news looking Real News

The above news looking real after clicking to the predict by the user

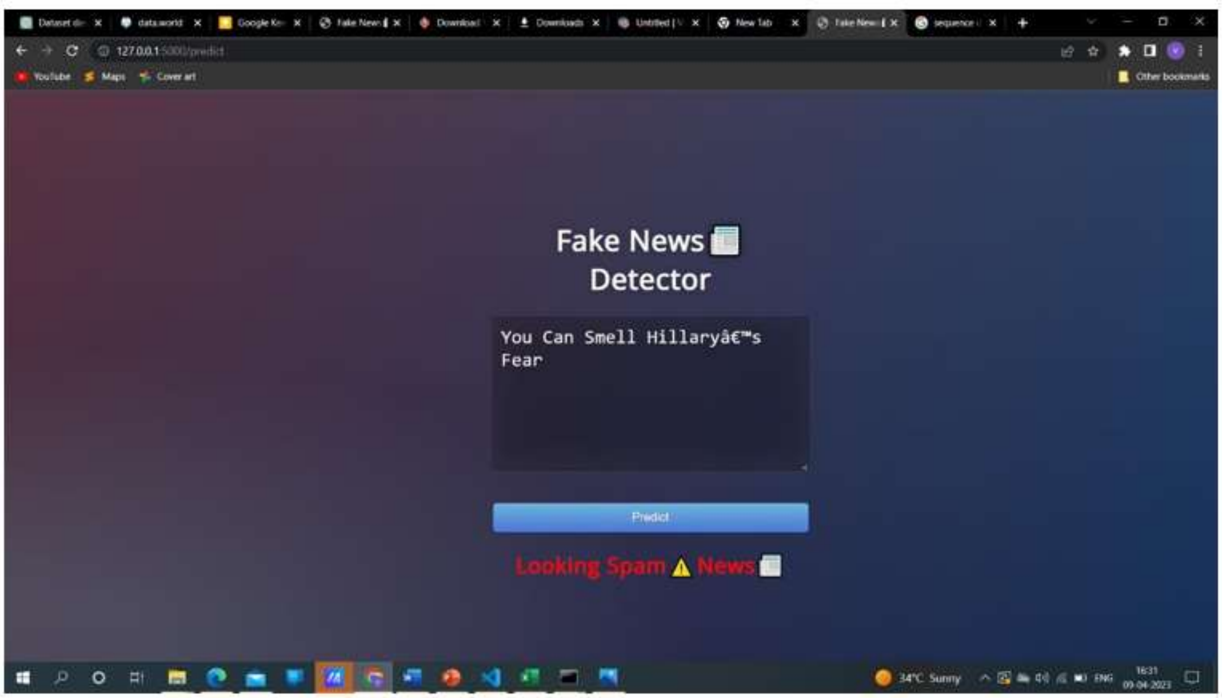

Fig : The news looking fake News

The above news looking fake after clicking to the predict by the user

6.CONCLUSION AND FUTURE WORK

CONCLUSION

In conclusion, utilising a pre-trained PassiveAggressiveClassifier model and a TfidfVectorizer for feature extraction, the provided code constructs a straightforward Flask web application for identifying fake news. Users of the software can enter news stories to get a prediction of whether they are authentic or phoney.

The dataset used to train the model is loaded from a CSV file, and the model is loaded from a stored pickle file. The train test split function from sklearn.model selection divides the data into training and testing sets. To convert the input data and determine whether the news is phoney or real, use the fake news det function.

The web application renders the HTML and builds the routes for the home page and prediction page using the Flask framework. templates to display the prediction results and the input form. The predict() function evaluates user input and returns the prediction, while the home() function generates the index.html template.

This code serves as a foundation for creating more intricate fake news detection programmes, and it may be improved by adding new features, increasing the model's precision, and refining the user interface.

Future Aims

The present solution employs TF-IDF vectorization to extract features from the text in order to add more features. The accuracy of the model can be increased by incorporating further characteristics like sentiment analysis, entity recognition, or topic modelling. The current implementation merely distinguishes between false and real news articles. To provide more functionality, such as summarization, translation, or fact-checking greater benefits for the users.

7.REFERENCES

[1] Castillo, C., Mendoza, M., & Poblete, B. (2011). Information credibility on Twitter. Proceedings of the 20th international conference on World wide web, 675-684.

[2] Gupta, A., Kumaraguru, P., Castillo, C., & Meier, P. (2013). TweetCred: Real-time credibility assessment of content on Twitter. Proceedings of the 22nd international conference on World Wide Web companion, 1001-1004.

[3] Stringhini, G., Kruegel, C., & Vigna, G. (2010). Detecting spammers on social networks. Proceedings of the 26th Annual Computer Security Applications Conference, 1-9.

[4] Zubiaga, A., Ji, H., & Knight, K. (2014). Tweet, but verify: epistemic study of information verification on Twitter. Proceedings of the 25th ACM conference on Hypertext and social media, 186-195.

[5] Zhou, X., Zafarani, R., & Shafiq, M. Z. (2015). A survey of fake news: Fundamental theories, detection methods, and opportunities. arXiv preprint arXiv:1811.00770.

[6] Kwak, H., Lee, C., Park, H., & Moon, S. (2010). What is Twitter, a social network or a news media?. Proceedings of the 19th international conference on World Wide Web, 591-600.

[7] Lu, Y., Wang, D., & Chau, M. (2011). Tweets and events: a study on the detection and evolution of news events using Twitter. Proceedings of the fifth international AAAI conference on weblogs and social media, 72-79.

[8] Zhao, W. X., Jiang, J., Weng, J., He, J., Lim, E. P., Yan, H., & Li, X. (2011). Comparing Twitter and traditional media using topic models. Proceedings of the 33rd European conference on Advances in Information Retrieval, 338-349.

[9] Conroy, N. J., Rubin, V. L., & Chen, Y. (2015). Automatic deception detection: Methods for finding fake news. Proceedings of the Association for Information Science and Technology, 52(1), 1-4. Rubin, V. L., Lukoianova, T., & Tatiana, L. (2015). Deception detection using rhetorical structure theory and vector space models. Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 1326-1336.