Proceedings of the Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV 2021).
IEEE Xplore Part Number: CFP21ONG-ART; 978-0-7381-1183-4

2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV) | 978-1-6654-1960-4/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICICV50876.2021.9388406

Combating Misinformation Dissemination through Verification and Content Driven Recommendation

[1]Sarah Hawa, [2]Lanita Lobo, [3]Unnati Dogra, [4]Prof Vijaya Kamble
[1,2,3,4,5]*Sardar Patel Institute of Technology, Bhartiya Vidya Bhavan's Campus, Andheri, West Mumbai, India*
[1]*sarah.hawa@spit.ac.in,* [2]*lanita.lobo@spit.ac.in,* [3]*unnati.dogra@spit.ac.in,* [4]*vijaya.kamble@spit.ac.in*

***Abstract*—The COVID 19 pandemic is a humanitarian emergency that poses an enormous threat to society and has impacted various social media platforms and journalism. News and social media has become an immensely popular platform for consumption of information. However, these platforms are also the bearer of fake news and information which causes negative effects and creates panic. Thus, this research work aim to tackle this problem by creating a unique hybrid model using Machine learning algorithms with Natural Language Processing (NLP) techniques to verify news. In order to make the proposed system foolproof, a superior content based recommendation system is developed which will encourage users to consume authenticated news and content from verified sources. Thus, such a system will provide a holistic approach as it not only verifies but also provides genuine and true recommendations for the same.**

***Index Terms*—Key Words: COVID-19; News; Machine Learning; Natural Language Processing (NLP); Content Based Recommendation.**

I. INTRODUCTION

SOCIAL MEDIA for content and news consumption is a sword with two edges and can have serious implications and consequence if not used ethically. Owing to its easily accessible nature, low cost and rapid relaying of information often lead people to pursue and consume news from social media. On the other hand, it permits the inescapable wide spread of fake news. Fake news may be spread intentionally or due to lack of awareness. The substantial spread of fake news has the capacity of having immensely negative influence on individuals and society. Fake news detection on social media has hence become a prevalent difficulty. Fake news is purposely written to mislead readers and to make them believe false information. Detection of origin of fake news is difficult since there is no one particular source. Anyone can be a peddler of false information on these gargantuan social media platforms. Earlier, forged news was just a reason of dissatisfaction but now has become an alarming issue. Misinformation is always a threat to humanity. Reception of fake news have caused several tragedies and have adversely affected people. A salient recent example of this is the COVID-19 pandemic, which leads to the circulation of false and unauthenticated news which is not only causing panic amongst citizens but also poses a threat to society. After reading fake news about Covid-19, many citizens have taken extreme actions which have affected their health too. The only way to eradicate the problem of fake news is to prevent oneself from consumption and further pass on of false information. This can be done by use of a fake news detection system that allows the user to verify whether the received information is hoax or not. According to the National Crime Records Bureau, 275 incidents of fake news were reported in India.

A. Literature Survey

The use of AI and machine learning has been widely used to tackle the problem of detection of misinformation. A number of approaches have been used in order to provide an optimum solution with the help of technology. In this section, we have reviewed different types of approaches, methods and related work with respect to detection and verification of fake information and news. A semi-supervised approach using K nearest neighbours to group articles based on similarities in words and then verifying the articles using neural network graphical techniques to capture textual dependencies [1]. However, such a model works accurately with a small multi-label dataset. Natural language processing (NLP) techniques such as text analytics, TD-IDF has been widely implemented and accepted as an approach for detection of fake news as it not only resolves ambiguity but also performs complex textual analysis for the computer. NLP techniques such as regular expressions, tokenization, stop words and normalisation plays an important role in pre-processing of data which makes it easier for the model to comprehend it. Regular expressions are used for feature extraction and string manipulations. Tokenization aids in the splitting of phrases and stop words help in removing words that don't add much value and therefore makes the phrase or sentence more concise. It has also been noticed that neural networks trained with N-gram which assigns probabil- ities to sentences and sequences of words performed better in comparison to sequence vectors [2] [3]. Furthermore, Models that are trained on a dataset consisting of news content rather than headlines tend to achieve higher accuracy and require more computational time. NLP when combined with recurrent neural network (RNN) and long short term memory networks (LSTM) further improves the efficiency of the model. It is also observed that classification machine learning models such as decision trees and Bayes-Net tend to give lower accuracy as compared to other models. In order to truly differentiate fake news from a real one, it is essential to analyse the linguistic characteristics of a news article to determine the true intent of the news. One such method was segregating the news article on the basis of a satire, hoaxes and propaganda and then comparing those to a set of real news articles [4].

Past studies have shown that a hybrid of a machine learning model comprising linguistic characteristics will prove to be a helpful and efficient solution. Naive Bayes Classifier based on Bayes theorem and Support vector Machine (SVM) is considered as one of the most simple and popular machines learning algorithms when it comes to detection of fake news. The use of clustering models and the Mahalanobis distance shape the accuracy of an SVM model which in turn helps in identifying the dependency of the independent variables whereas Naive Bayes classifies the data on the degree of correlation between the variables [5] [6]. Another unique hybrid for news detection is a combination of Naive Bayes classifier, Support Vector Machines, and NLP which tends to perform better instead of using algorithms that cannot replicate and produce cognitive and linguistic functions. The three-part method is a combination of machine learning algorithms using supervised learning techniques, and natural language processing methods. As discussed earlier, the three methods can be used individually and successfully to serve the purpose. However, in order to gain accuracy, a hybrid model approach was selected for the detection fake news [7].

B. Objective

In this paper, we shall discuss how the problem of fake news can be tackled on an individual level. Fake news is now categorized under cyber crime since the level of damage caused by it is simply increasing day by day. Since fake news is a huge segment including information from various sectors, we shall focus on two types of prominent news section which are currently all around the globe. The system which will be talked about in this paper will be centered around COVID-19 and political news. Severe problems have happened due to misinformation about COVID-19 which has also claimed people's lives. In order to address the problem, we aim to build a system which will not only verify and authenticate the information or news provided by the user but will also allow him to access information which is published by trusted sources.

II. CONTENT AND URL VERIFICATION

In order to classify news accurately, we will be adopting a machine learning approach with natural language processing. This will help us classify and also authenticate whether a piece of information is false or correct thus aiding in its verification as seen in Figure. 1.

A. Dataset Preparation

Since we needed our data set to be compatible with different genres of news content and information, we chose news information from two significant events which we felt have been influenced the generation of fake news in the past. The dataset revolves around two genres: Political the COVID 19 pandemic. They are then analysed to extract features. The first Data set is based on Trump's US Presidential Campaign in the elections of 2016 [8] [9]. The term 'fake news' is said to have been actually coined during this period since the social media

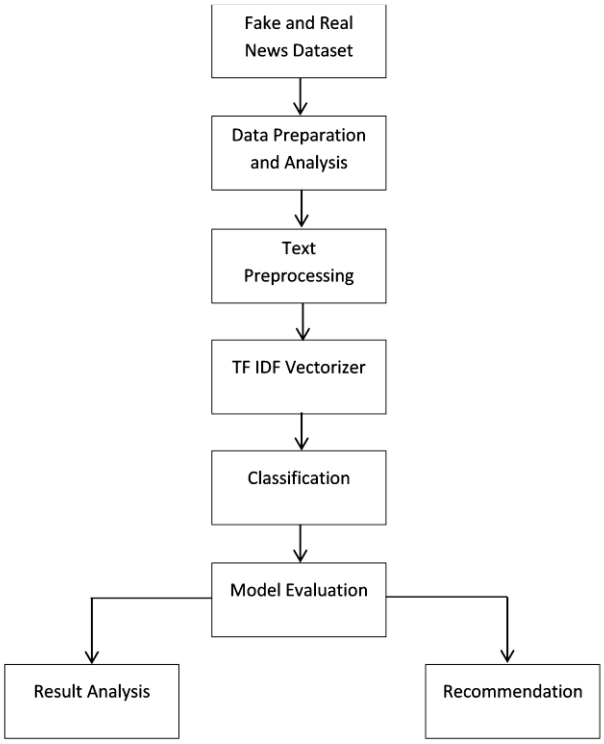

Fig. 1. Fake News Detection Block-Diagram

was flooded by false information and hoaxes about Trump's Campaign and actions. Second data set that has been used revolves around the COVID-19 Pandemic,that currently exists across the world. Today, half the information reports that are being shared across platforms about COVID-19 turn out to be fake. Fake news spread has seen a dramatic increase post the COVID-19 lockdown. This also has created a ruckus and has become a serious crisis. Three datasets were used:

1) ISOT Fake news dataset: This dataset contained articles obtained from legitimate sources as well as unreliable web-sites, all offered by politifact.com.

2) FakeCovid: This data set is a multilingual cross-domain fact check news data set relating only to COVID-19. Thus on combining the datasets, we have significant amount of information from 50,000 plus articles from two different genres. This exhaustive dataset will help us analyze the linguistic characteristics and train the algorithm in a much more efficient and accurate manner

3) Data scraping: Web scraping is basically the process of extraction and collection of data from various web pages. Instead of collecting this data manually, we build a web scraping engine that can extract significant amounts of data in a very short span of time [10]. The web scraping engine consists of a crawler and a scraper.The web crawler browses the internet and tries to locate all the relevant content required. After which, the web scraper extracts that data. The data collected by web scraping complements the existing dataset thus helping helping the model in gaining better insights [11]. These insights will help improve the classification and decision

making of the model. The libraries used to create a web scraping engine were Beautiful Soup, Requests and spacy. In order to extract data from a web page, the Hypertext Markup Language (html) source code needs to be extracted, this is done by sending a Hypertext Transfer Protocol (HTTP) request to the web page using the requests module in python. The text() method of the requests module is used to extract the html text. An object of the Beautiful soup module is used to traverse through the html text and extracts the html tags of the titles, paragraph tags and header tags. Spacy module is used for named entity extraction. Extracting and labeling entities as either an organization, a person or a geopolitical entity helps in understanding what a particular text data is about. After extracting all the required data from a url, pandas module is used to organize the entire data in a data frame for training the model.

B. Dataset Visualization

For the purpose of Data Analysis we have used a sample of approaches in order to understand the characteristics and vivid properties of the data set acquired. Following are the visualizations carried out as seen in Figure 2.

1) Authenticity Check: Authenticity of the dataset is checked and news is distinguished as fake and real. The graph shows how much of the total percentage of information available is fake and real. This gives us an idea of the ratio of fake news existing in the given news genre. The graphs for Political News and COVID-19 news are shown below.

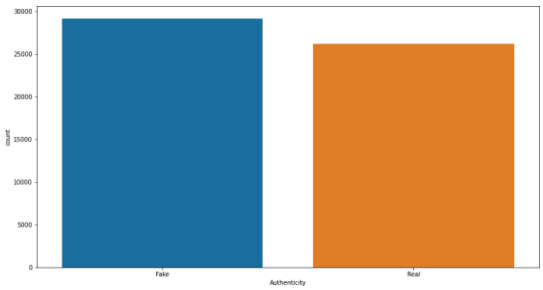

Fig. 2. Authenticity Check

2) Length of article: The lengths of news articles are analyzed. This can be used to check the difference between the size of real news content and that of the fake news. The below histogram bins both COVID-19 and Trump related news articles that possess similar word lengths and also where for every word length, the frequency distribution is displayed. The frequency distribution of length of the words for factual news articles is depicted by orange bars while for fake news articles, blue bars show the distribution. In Figure 3, some of the insights from this histogram can be seen.

- As seen, very few fake news articles have more than 30000 words. While as compared to that, there are higher number of true articles with word length of more than 30000 words.

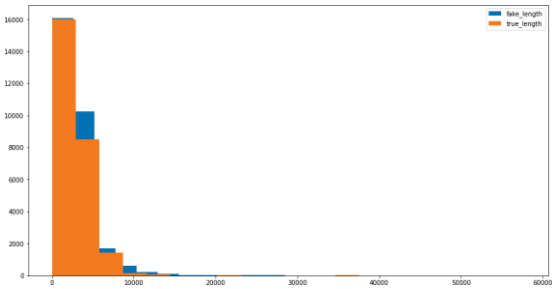

Fig. 3. Length of Articles

- There are upto 16,000 news articles that have text length of zero.

3) Word cloud : Word Clouds are typically used to highlight frequently seen terms depending on the frequency with which they are used and their prominence. A word cloud is an image that gives one a good idea about the nature of the content in just one look. Here, the words that have a higher frequency of occurrence in the text are highlighted as observed in Figure 4.The keywords from this word cloud can be displayed on the main website under the Important keywords section showing the news trend across the globe.

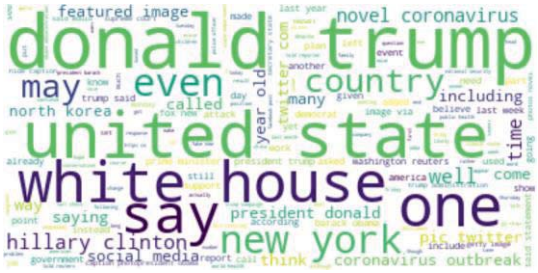

Fig. 4. Word Cloud

4) Lexical Distribution : Lexical Pattern shows words or chunks of test that occur in a data set with high frequency and the meaning of the parts of which are sometimes different than the meaning of the whole. The frequency of individual words alone is portrayed by this distribution as observed in Figure 5 and Figure 6

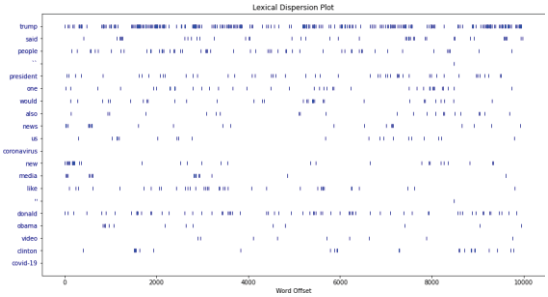

Fig. 5. Lexical plot-1

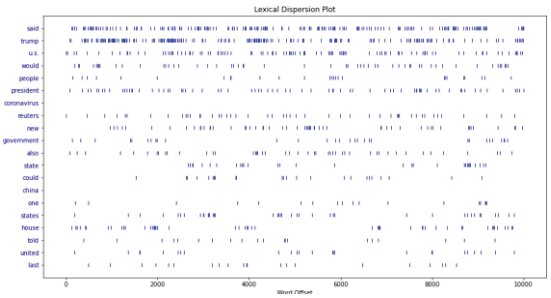

Fig. 6. Lexical plot-2

5) Text Pre-processing: The purpose of text processing is to transform our data into a numeric format which can be understood by the machine. This understanding of the data will help the algorithm make accurate classifications. However, before we transform this data into a numeric format, it is essential to cleanse and preprocess our data to get rid of any unwanted text, numbers, punctuation's etc that may not be very useful for our model. This step is one of the most important steps while working with machine learning and natural language processing problem statements since the data is unstructured. It's also important to note that text preprocessing techniques should be chosen in such a way that is useful and meaningful for our use case

6) Transforming sentences to lowercase: In text preprocessing, the first and the foremost step is to normalise all the text and words to lowercase. This step ensures that all the words are normalised and treated in the same way to avoid any duplicate copies and reception of the same word across the vocabulary.

7) Removal of punctuation: The second step is to get rid of punctuation as it's not significant for our problem statement. This will remove all the punctuations by standardising our text and thereby further reducing the word list. In order to do this, we will use string.punctuation available in the library.

8) Tokenization: After we normalise our text by removal of punctuations and transforming them into lowercase, we then proceed with tokenizing our text. This is an instrumental part of the text preparation process. Thus Tokenization aims to split the characters, words or sub-words etc into a smaller and shorter unit described as a token.

9) Stop words: Once we have converted our text into smaller tokens, we then proceed to remove insignificant words from the text that are generally used in sentence formation. However, they not only increase the word list but don't add any value to the text. The list of stopwords can be found by importing stopwords.

10) Stemming: After the text has been cleansed and normalised of stop words, punctuation's and irregular word sensitization, we then proceed to the last step which is termed as stemming. Steaming is defined as a process to translate alternative's of a given word to its root form by using linguistic methods such as affliction, suffixation etc. This aids in further reducing the word list by grouping words having the same variants.

C. Vectorizer

Once our text has been preprocessed, it's essential to translate the text obtained into integers or floating-point values which are very essential in predictive modelling. For this very purpose, we used vectorizers. We will be using the Term Frequency-Inverse Document Frequency for the same.

1) Term frequency-inverse document frequency TFIDF: This statistical tool is used for evaluating the relevancy of a particular word in a group of documents. This is essential for automated analysis of text which helps in scoring the relevance and significance of words. The statistical measure is calculated by multiplying the term frequency along with the inverse document frequency.

D. Classification

1) Na¨ive Bayes: This is based on Bayes Theorem wherein all the pairs of features that are classified are independent. This involves preprocessing the text by using TF-IDF and then implementing the multinomial naive Bayes on these preprocessed outputs which allows the algorithm to work on the most prominent words within a document.

2) Linear Support Vector Classifier (SVC): The objective of this Classifier is to fit to the provided data, returning a best fit hyper plane that categorizes the data. Post getting the hyper plane features can be added to the classifier to see what exactly the predicted class is.

3) Logistic regression: Logistic regression is a statistics and analytical model which uses a function(logistic) to portray a dependent variable which is binary in nature.

4) Random forest: Random forests are methods typically learning methods deployed for classification. These are also used for regression and operations which need multitude of decision trees that are constructed at the time of training.This gives us an output that is a class which is the mode of the all the classes that are present.

5) K-nearest Neighbours: This is an algorithm which is used to store various cases which are available to the model and later categorise new cases by means of a measure of similarity.

E. Model Evaluation

Classification algorithms aid us in understanding and differentiating a fake piece of information from a real one. Once the news article has been normalised by performing textual processes such as stemming, tokenization and vectorization, the next essential step is to test and evaluate the model for different classifiers on the basis of Accuracy, F1 score, Recall and Precision to validate the performance. These performance parameters will aid us in identifying and selecting the appropriate model for detection of misinformation using content-based information.

III. RECOMMENDATION

The recommendation system is a very important part of the project. The purpose of a news recommendation system is to provide the users with correct and trustworthy information and

sources in response to the URL or news article provided by the users. A recommendation system algorithm can be either classified as content-based recommender algorithm or collaborative recommender algorithm. Content-based recommender algorithm creates a profile for a particular user by using the features associated with the content provided by the user. It uses keywords from the user content and tries to recommend the most similar article or item with the help of those keywords [12]. The collaborative recommender algorithm is a machine-independent algorithm. It uses ratings and tries to build an inter-user connection with users having similar rating patterns. Furthermore, with the help of this data, it tries to predict a particular user's interest and future behaviour [13]. This project uses a content-based recommendation system to recommend verified news articles to the users. As the purpose of this project is to provide a user with the most accurate information on a particular topic, content-based recommender serves to be the right choice for our problem statement.

A. Methodology

Content-based filtering mechanism uses concepts like Term Frequency (TF), Inverse term frequency (IDF) and cosine similarity to determine the relative importance of a document or a news article as demonstrated in Figure III-A

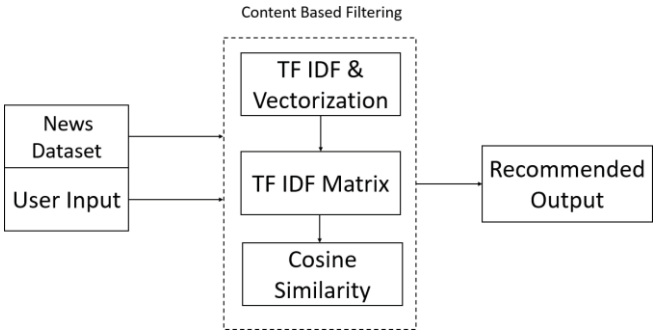

Fig. 7. Recommendation System Block Diagram

1) TF-IDF vectorization: Similar to the news verification model the recommendation machine uses TF-IDF vectorization to find the most relevant keywords that describe the news article in the best possible way and to dampen unnecessary high frequency words that can be neglected.

2) TF-IDF matrix: The scikit-learn library in python provides an in-built TF-IDF vectorizer that is used to calculate the TF-IDF score for each word in a document. The TF-IDF matrix that is generated contains all the words with their respective TF-IDF scores.

3) Cosine Similarity: After calculating the word importance the calculation of relevance and similarity of one document with respect to other documents is calculated using the cosine similarity. Cosine similarity is the calculation of angle difference between two vectors under consideration. Less angle difference indicates that the two documents are more similar. As the cosine value of a 0-degree angle is 1, any document with a cosine similarity value closest to the value 1 is considered as the closest fit for recommendation to the user.

4) Recommendations: The training method of the recommendation engine will perform TF-IDF vectorization across the entire input document and calculates the cosine similarity for every document present in the dataset. The predict method of the engine will finally return the best possible recommendation for the given input.

```
In [71]: results('As U.S. budget fight looms, Republicans flip their fiscal script')

Out[71]: [{'id': 244,
           'Name': 'u.s. tax revamp still incomplete as republicans eye social program cuts'},
          {'id': 6,
           'Name': 'white house, congress prepare for talks on spending, immigration'},
          {'id': 687,
           'Name': 'deficit worries complicate path for republican tax cuts'},
          {'id': 417, 'Name': "senate tax bill stalls on deficit-focused 'trigger'"},
          {'id': 446, 'Name': "senate tax bill stalls on deficit-focused 'trigger'"},
```

Fig. 8. Recommendation Results

IV. RESULTS

In order to truly understand and select a method which would help us in the detection of misinformation, we selected two genres of news to make our model compatible with any type of news genre. This will help us understand if the model works accurately for different genres of news and will further aid in correctly predicting whether it's fake or real. Next, we carried out text processing techniques by using natural language processing to normalize the text which aided in reducing the vocabulary. We selected 5 classifiers to understand which algorithm worked best with our problem statement. The classification algorithm was selected on the basis of F1 score, Recall, Accuracy and Precision. From the above results in Table I, it is seen that SVC recorded an accuracy of 99 per cent. It is also observed that SVC had a clearly high recall and precision values as compared to the other 4 algorithms which indicate that the model identified the class of news accurately and recorded a higher number of true positives and true negatives for the dataset. After SVC, we also see that Naive Bayes, logistic regression and Random forest record accuracy of 94, 96 and 97 per cent respectively. The precision and recall values for these classifiers were also quite high which indicates a good number of true positives and true negatives as given in TableII. K nearest neighbours, however, recorded a very low accuracy of 69 per cent and had a high number of false positives and negatives, making it unfit for adoption in case of our problem statement. Thus by using these metrics, we determined that linear SVC worked best as a classifier for our problem statement.

TABLE I
TF-IDF VECTORIZER RESULTS

	precision	recall	f1 score	accuracy
Naive Bayes	0.94	0.94	0.95	0.94
Support Vector Machine	0.99	0.99	0.98	0.99
Logistic Regression	0.98	0.97	0.98	0.98
Random Forest	0.96	0.97	0.97	0.97
K Neighbours	0.78	0.69	0.66	0.69

TABLE II
CONFUSION MATRIX INTERPRETATION

	TP	FP	TN	PN
Naive Bayes	5572	316	4883	304
Support Vector Classifier	5893	44	5107	31
Logistic Regression	5859	103	5048	65
Random Forest	5828	274	4877	96
K-Neighbours	5818	3288	1863	106

A. Model Deployment

Model deployment is a very essential and integral part of our project where we combine our news verification model with our recommendation engine and deploy the model using flask. By doing so, we will make the model available for users to use which help further help in tackling the problem of news verification. The website will not only verify but will also recommend true information about similar topics to the user. This end to end solution will help us solve the problem and provide an effective solution in combating the spread of misinformation.

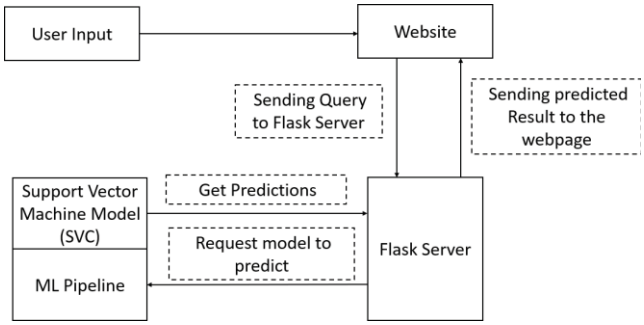

Fig. 9. Model Deployment Block Diagram

1) Front end: Hyper-text is the text that gives links forother associated documents and Markup is the process of giving tags to parts of the available content. For the project we have used HTML 5 version. The HTML code is responsible for giving and defining the structure of content on the website and also annotating the content. In the HTML file, in order to accomplish the navigation bar for content and URL verification, we linked three html files each for home, content and url by using the link href tag. After that we used an input form since the user will be required to enter the data that needs to be verified. Along with an input bar we defined a button to validate the prediction search and give the data from the user to the machine code. Eventually prediction text is defined wherein the final result is displayed onto the website that is "The news is fake/real". CSS is used to give the colors, styles, layout, gradients and other styling aesthetics to the website. The CSS file is linked into the HTML files. The main challenge here was to style the navigation bar properly. First the unlisted list items were turned into one row text then separated and aligned to the center. Then boxes of relevant dimensions were created for each and sections were created for each html file viz home, content and url. Text was styled and backgrounds were changed. A special hover feature was added in order to highlight the items that the mouse pointer hovers above in the navigation pane. Finally a gradient was added to give the website a soothing touch. The website is shown below. After designing the HTML website, we build the pipeline for our project. In order to simplify the working, we create a pipeline for our model using our functions for text processing using NLP, SVC and TFIDF. We save this pipeline with the help of the dump function which is available in the jolib library.

2) Connection of the web-page with the model using Flask: After implementing the font end, we work on connecting the front end with our back-end using a web application server called Flask. This is one of the most integral and crucial parts of deploying a machine learning model. The first and foremost step would to load the saved pipeline model after which we define a function called request_content and request URL which will aid in obtaining the input query from the user. To do so, we create an object of the flask and defines the route. The route function will help the flask application in identifying the route to the given URL path, after which, flask will then call for the home function for rendering of the HTML files. Once flask receives a query, it detects a POST method and upon detection, it will call the GET function to receive the user input. One the query has been requested and obtained by the model, the model will feed it into the pipeline to get the result along with the recommendations, and return the results to the user. This will be initiated with the help of the REQUEST results function which will send the results back to the web-page.

3) Local Host: For the last step, we will be using the python local-host server to deploy our model. It is a relatively quick web server that has the ability to process HTTP request without having to install any other software's. Python comes with its own built-in server that aids in running static HTML web-pages. To do so, we proceed on to calling the run function which initiates the starting of the flask server. We are then provided with an address http://127.0.0.1:5000/. This will direct us to our web-page where the flask server wound have rendered the HTML template.

V. CONCLUSION

In this paper, we introduced the concept of training the model with different types of news genres. This combined genre will aid us in improving the decision-making and classification of news articles irrespective of the genre or content. Furthermore, in order to truly understand the linguistic features and characteristics of news and misinformation, various text and data analysis such as comparison of the two types of information categories on that basis of their length and number of articles. Theses insights were further visualised through word clouds and lexical plots to understand the most frequent words and vocabulary that is generally used in fake and real article's. These visualizations and analysis helped in determining which features from the dataset should be considered before

Fig. 10. Front-end for URL verification

Fig. 11. Front-end for Content verification

we train the model. From the data analysis and visualisations, it was clear that the text length wasn't of significant use to us. Other factors such as date, subject were discarded. The title was merged with the news content and a corpus was made using the two. This corpus was then preprocessed using natural language processing where the text was normalised and standardised by converting it into lowercase, by removing stop words, numbers, punctuation's. After which, the obtained text was tokenised and reduced to it's root form by stemming. These techniques helped in reducing the word size and list and helped us in obtaining the useful part of the corpus while getting rid of unnecessary and insignificant words and vocabulary. After which, we trained the model by using 5 classifiers out of which linear SVC outperformed all the other classifiers on the basis of accuracy, precision, F1 and recall. Furthermore, after building our classification model, we built a content based recommendation engine which would suggest similar articles using the combination of TF-IDF and cosine similarity. The recommendation system was built and integrated in the system with an aim to provide factual and authenticate articles once the user has sent in a request to verify a certain piece of information or URL. Thus by doing so we aimed to create an end to end solution that would combat the spread of misinformation through not only verification but also authenticated recommendations. In order to combine this entire project and provide this end to end solution to the user, we connected the model with the HTML web page using FLASK and deployed it on pythons local host server. The Front-end was designed keeping in mind the simplicity and user experience. Such a holistic platform will not only help in spreading awareness about misleading information but will also provide true information. This will keep the user informed and will urge them to verify every piece of content they read on the internet. Thus keeping themselves and their loved ones safe from the perils of misinformation. For future work, neural networks and advanced classifiers could be used to further train the model and increase the complexity. Another future extension of the project could include verification of images and videos along with content based information. Such features would aid in increasing general awareness about consuming false content along with a solution to verify the same. This would help users be more informed, alert and safe

REFERENCES

[1]C. K. Hiramath and G. C. Deshpande, "Fake News Detection Using Deep Learning Techniques," 2019 1st International Conference on Advances in Information Technology (ICAIT), Chikmagalur, India, 2019, pp. 411-415, doi: 10.1109/ICAIT47043.2019.8987258.

[2]S. H. Kong, L. M. Tan, K. H. Gan and N. H. Samsudin, "Fake News Detection using Deep Learning," 2020 IEEE 10th Symposium on Computer Applications Industrial Electronics (ISCAIE), Malaysia, 2020, pp. 102-107, doi: 10.1109/ISCAIE47305.2020.9108841.

[3]H. Telang, S. More, Y. Modi and L. Kurup, "Anempirical Analysis of Classification Models for Detection of Fake News Articles," 2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), Coimbatore, India, 2019, pp. 1-7, doi: 10.1109/ICECCT.2019.8869504.

[4]H. Rashkin, E. Choi, J. Y. Jang, S. Volkova, and Y. Choi. "Truth of varying shades: Analyzing language in fake news and political fact-checking". In Proceedings of EMNLP, pages 2921–2927, 2017.

[5] Conroy, N.K., Rubin, V.L. and Chen, Y. (2015), Automatic deception detection: Methods for finding fake news. Proc. Assoc. Info. Sci. Tech., 52: 1-4. doi:10.1002/pra2.2015.145052010082.

[6] Mykhailo Granik, Volodymyr Mesyura," Fake News Detection using naive Bayes classifier", 2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON), Kiev, Ukraine,doi:10.1109/UKRCON.2017.8100379

[7]B. Al Asaad and M. Erascu, "A Tool for Fake News Detection," 2018 20th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), Timisoara, Romania, 2018, pp. 379-386, doi: 10.1109/SYNASC.2018.00064.

[8] Ahmed H, Traore I, Saad S. "Detecting opinion spams and fake news using text classification",Journal of Security and Privacy, Volume 1, Issue 1, Wiley, January/February 2018.

[9] Ahmed H, Traore I, Saad S. (2017) "Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques. In: Traore I., Woungang I., Awad A. (eds) Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments. ISDDC 2017. Lecture Notes in Computer Science, vol 10618. Springer, Cham(pp. 127-138).

[10]R. Diouf, E. N. Sarr, O. Sall, B. Birregah, M. Bousso and S. N. Mbaye, "Web Scraping: State-of-the-Art and Areas of Application," 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 2019, pp. 6040-6042, doi: 10.1109/BigData47090.2019.9005594.

[11]D. M. Thomas and S. Mathur, "Data Analysis by Web Scraping using Python," 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 2019, pp. 450-454, doi: 10.1109/ICECA.2019.8822022.

[12]S. N. Ferdous and M. M. Ali, "A semantic content based recommendation system for cross-lingual news," 2017 IEEE International Conference on Imaging, Vision Pattern Recognition (icIVPR), Dhaka, 2017, pp. 1-6, doi: 10.1109/ICIVPR.2017.7890880.

[13]Y. Dong, S. Liu and J. Chai, "Research of hybrid collaborative filtering algorithm based on news recommendation," 2016 9th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), Datong, 2016, pp. 898-902, doi: 10.1109/CISP-BMEI.2016.7852838.