Unveiling Truth: A Deep Dive into Distinguishing Real News from Satirical News using Machine Learning

Blog by: Divya Shanbhag, Harshita Vyas and Jaanvi Das

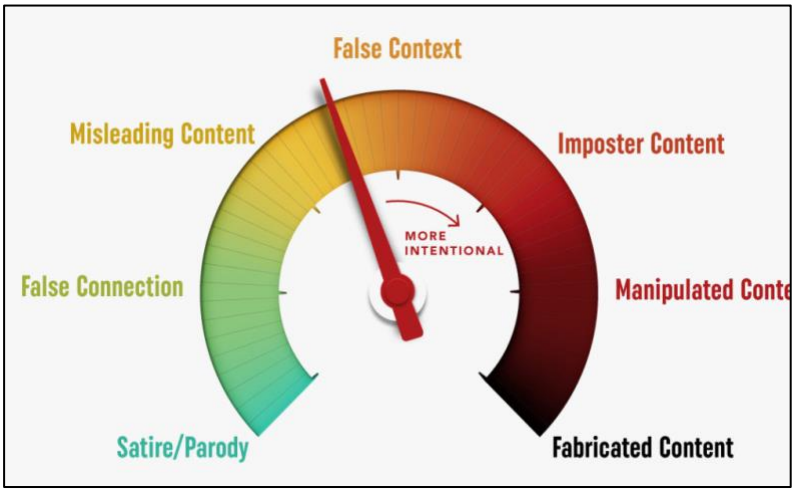
Introduction

In the digital age, the widespread dissemination of fake news has created an environment where distinguishing between genuine news and satirical content has become increasingly challenging. While satirical news serves as a source of both entertainment and social critique, its potential confusion with actual news has led to a decline in media literacy and trust in the news. This project aims to address this issue by creating a robust machine learning model that can accurately differentiate between real news and satirical content.

With the coexistence of satirical news and genuine information on various digital platforms, the task of distinguishing between the two has become complex. The similarities in presentation, writing style, and sometimes misleading content have contributed to the blurring of lines between fact and satire. This confusion has led to the unchecked spread of misinformation, undermining the credibility of news sources and the principles of responsible journalism. Thus, there is an urgent need for a reliable machine learning solution that can effectively identify and differentiate between satirical and real news articles.

Why Satirical News?

Focusing on satirical news in this project is motivated by various valid reasons. Firstly, the clarity of intent in satirical news, which is designed for entertainment and commentary with humor, contrasts sharply with the ambiguous nature of "fake news." Moreover, by distinguishing satire as a form of expression, the project can safeguard free speech, avoiding potential censorship concerns. Additionally, this approach respects cultural sensitivities, acknowledging the diverse nature of humor and satire across different societies. By honing in on the recognizable patterns of satirical content, the project can develop more effective detection models, enhancing media literacy education and fostering critical thinking among readers. Overall, this nuanced focus acknowledges the need to combat misinformation while respecting the nuances of expression and cultural diversity.



Dataset Formation

Data was acquired by web scrapping. We used one website for each label- Real and Satirical.

Data was added and collected regularly to the dataset over the course of 3 weeks until the number of articles reached approximately 500 for each label.

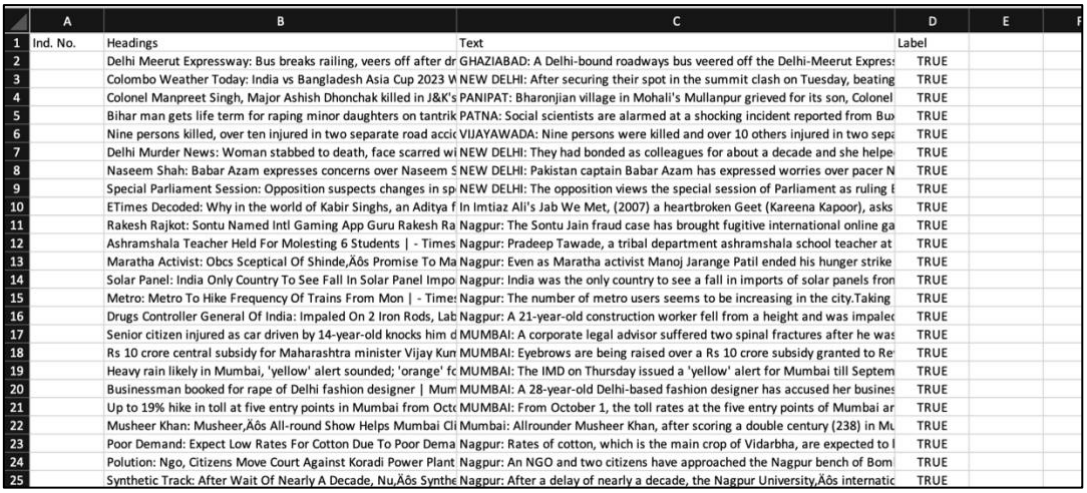
No. of datapoints for -

Satirical News: 508 Articles

Real News: 521 Articles

Dataset Description

Variable	Description
Ind. No	Index of the article
Headings	Heading of the article
Text	Content (or Body) of the article
Label	Describes if the article is Real or Satirical

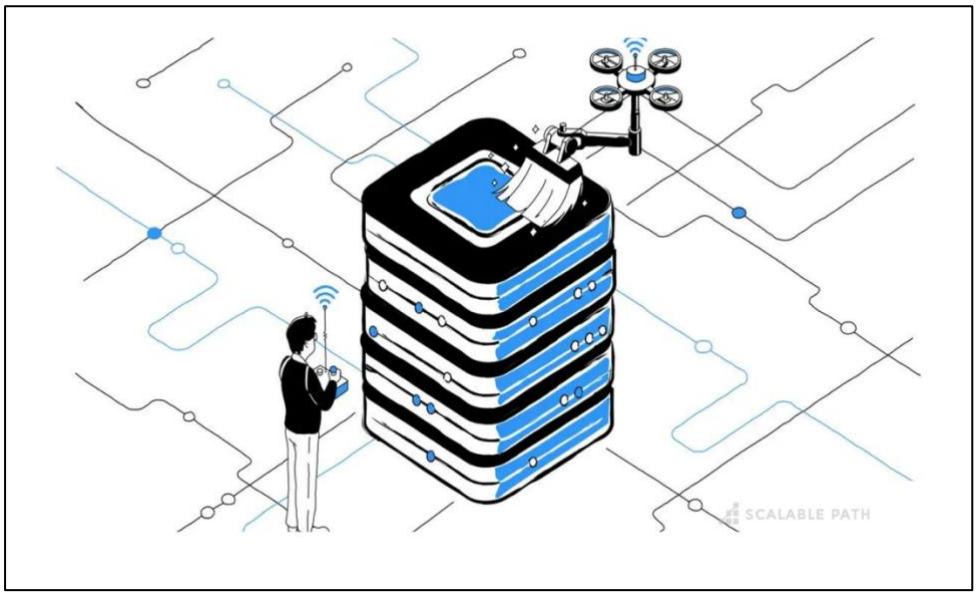


	A	B	C	D	E	F
1	Ind. No.	Headings	Text	Label		
2		Delhi Meerut Expressway: Bus breaks railing, veers off after dr	GHAZIABAD: A Delhi-bound roadways bus veered off the Delhi-Meerut Expres	TRUE		
3		Colombo Weather Today: India vs Bangladesh Asia Cup 2023 M	NEW DELHI: After securing their spot in the summit clash on Tuesday, beating	TRUE		
4		Colonel Manpreet Singh, Major Ashish Dhonchak killed in J&K's	PANIPAT: Bharonjian village in Mohali's Mullanpur grieved for its son, Colonel	TRUE		
5		Bihar man gets life term for raping minor daughters on tantrik	PATNA: Social scientists are alarmed at a shocking incident reported from Bu	TRUE		
6		Nine persons killed, over ten injured in two separate road accic	VIJAYAWADA: Nine persons were killed and over 10 others injured in two sepa	TRUE		
7		Delhi Murder News: Woman stabbed to death, face scarred wi	NEW DELHI: They had bonded as colleagues for about a decade and she helpe	TRUE		
8		Naseem Shah: Babar Azam expresses concerns over Naseem S	NEW DELHI: Pakistan captain Babar Azam has expressed worries over pacer N	TRUE		
9		Special Parliament Session: Opposition suspects changes in sp	NEW DELHI: The opposition views the special session of Parliament as ruling E	TRUE		
10		ETimes Decoded: Why in the world of Kabir Singhs, an Aditya f	In Imtiaz Ali's Jab We Met, (2007) a heartbroken Geet (Kareena Kapoor), asks	TRUE		
11		Rakesh Rajkot: Sontu Named Intl Gaming App Guru Rakesh Ra	Nagpur: The Sontu Jain fraud case has brought fugitive international online ga	TRUE		
12		Ashramshala Teacher Held For Molesting 6 Students | - Times	Nagpur: Pradeep Tawade, a tribal department ashramshala school teacher at	TRUE		
13		Maratha Activist: Obcs Sceptical Of Shinde,Äôs Promise To Ma	Nagpur: Even as Maratha activist Manoj Jarange Patil ended his hunger strike	TRUE		
14		Solar Panel: India Only Country To See Fall In Solar Panel Impo	Nagpur: India was the only country to see a fall in imports of solar panels fron	TRUE		
15		Metro: Metro To Hike Frequency Of Trains From Mon | - Time	Nagpur: The number of metro users seems to be increasing in the city.Taking	TRUE		
16		Drugs Controller General Of India: Impaled On 2 Iron Rods, Lab	Nagpur: A 21-year-old construction worker fell from a height and was impaled	TRUE		
17		Senior citizen injured as car driven by 14-year-old knocks him d	MUMBAI: A corporate legal advisor suffered two spinal fractures after he was	TRUE		
18		Rs 10 crore central subsidy for Maharashtra minister Vijay Kun	MUMBAI: Eyebrows are being raised over a Rs 10 crore subsidy granted to Re	TRUE		
19		Heavy rain likely in Mumbai, 'yellow' alert sounded; 'orange' fc	MUMBAI: The IMD on Thursday issued a 'yellow' alert for Mumbai till Septem	TRUE		
20		Businessman booked for rape of Delhi fashion designer | Mum	MUMBAI: A 28-year-old Delhi-based fashion designer has accused her busines	TRUE		
21		Up to 19% hike in toll at five entry points in Mumbai from Oct	MUMBAI: From October 1, the toll rates at the five entry points of Mumbai ar	TRUE		
22		Musheer Khan: Musheer,Äôs All-round Show Helps Mumbai Cli	Mumbai: Allrounder Musheer Khan, after scoring a double century (238) in Mu	TRUE		
23		Poor Demand: Expect Low Rates For Cotton Due To Poor Dema	Nagpur: Rates of cotton, which is the main crop of Vidarbha, are expected to l	TRUE		
24		Polution: Ngo, Citizens Move Court Against Koradi Power Plant	Nagpur: An NGO and two citizens have approached the Nagpur bench of Bom	TRUE		
25		Synthetic Track: After Wait Of Nearly A Decade, Nu,Äôs Synthe	Nagpur: After a delay of nearly a decade, the Nagpur University,Äôs internatic	TRUE		

Raw Web Scraped Data (Real News)

Preprocessing the Data

Preprocessing text data is a critical step in natural language processing (NLP) tasks that involves cleaning and transforming raw text into a format that is suitable for analysis or model training. Text preprocessing is a critical step in natural language processing, ensuring that raw text data is refined and cleaned before analysis. It addresses the challenges posed by the complexity of human language, including variations in word forms, sentence structures, and the presence of noise. By standardizing the data, text preprocessing enhances the accuracy of NLP models, enabling them to better understand context and extract meaningful insights. It also helps in reducing the computational burden by simplifying the data, making it more manageable for analysis. Overall, text preprocessing acts as a bridge between human language and computational algorithms, facilitating the effective understanding and interpretation of textual information.



The steps we have taken to preprocess the data are:

1) Text Lowercasing: Converting all text to lowercase helps ensure that words with the same characters but different cases are treated as identical, improving uniformity in the dataset.

```
[ ]  headings = df['Headings'].str.lower()
     text = df['Text'].str.lower()
```

lowercasing the text

2) Data Cleaning: This involves the removal of any irrelevant or unnecessary data, such as special characters or symbols to ensure the data is consistent and accurate.

```
def remove_pun(text):
  return re.sub(r'[!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~]', '', text)

headings = headings.apply(remove_pun)
text = text.apply(remove_pun)

print(headings)
print(text)
```

```
import unicodedata

def remove_special_characters(text):
    # Remove non-ASCII characters
    text = unicodedata.normalize('NFKD', text).encode('ascii', 'ignore').decode('utf-8', 'ignore')
    # Remove remaining special characters
    text = re.sub(r'[^\w\s]', '', text)
    return text

headings = headings.apply(remove_special_characters)
text = text.apply(remove_special_characters)
```

Removing the $pec!@1 Characters

3) Tokenization: This process involves breaking down the text into individual words or tokens, making it easier to analyse and process each element separately.

```
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize

nltk.download('stopwords')
nltk.download('punkt')

words_headings = []
words_text = []

for items1 in headings:
  # Tokenize the text
  tokens1 = word_tokenize(items1)
  words_headings.append(tokens1)


for items2 in text:
  # Tokenize the text
  tokens2 = word_tokenize(items2)
  words_text.append(tokens2)
```

‘Code’, ‘to’, ‘tokenise’, ‘the’, ‘data’, ‘using’, ‘NLTK’, ‘library’

4) Stop Word Removal: Stop words are commonly occurring words like "the," "is," and "in" that do not carry significant meaning. Removing these words helps to reduce noise and focus on the more relevant content.

```
[ ]  filtered_words_text = []

     for tokens in words_text:
       filtered_tokens2 = [word for word in tokens if word.lower() not in stop_words]
       filtered_words_text.append(filtered_tokens2)


     print("After stop words removal: ",filtered_words_text)
```

Stop Words ~~are~~ Removed

5) Lemmatizing: Lemmatization is a text preprocessing technique used in natural language processing (NLP) that involves reducing words to their base or root form, known as the lemma.

```
lemmatized_words_headings = [[lemmatizer.lemmatize(word) for word in tokens]
    for tokens in filtered_words_headings
]
```

Words are Lemmatised -> Word be Lemma

6) Text Normalization: Normalizing tokens in data preprocessing, refers to the process of refining the text data to ensure consistency and uniformity in the words. While lemmatization involves reducing words to their base or root forms, normalizing tokens can involve various additional steps to standardize the text further.

```
[ ]  def text_normalization(tokens):
         normalized_tokens = []
         for word in tokens:
             if 'exmple' in word:
                 normalized_tokens.append(re.sub(r'exmple', 'example', word))
             else:
                 normalized_tokens.append(word)
         return normalized_tokens
```

Contractions, for example, ~~aren't~~ are not present after this process

7) Vectorization: Vectorization is the process of converting text data into numerical representations that can be processed by machine learning algorithms, enabling the analysis of text through mathematical computations.

```
[67] from sklearn.feature_extraction.text import TfidfVectorizer

     vectorizer = TfidfVectorizer()
     X_headings_train_tfidf = vectorizer.fit_transform(X_headings_train)
     X_headings_val_tfidf = vectorizer.transform(X_headings_val)
     X_headings_test_tfidf = vectorizer.transform(X_headings_test)
     text_tfidf = vectorizer.transform([text])
```

sparse matrix of type '<class 'numpy.float64'>'

(Data is no longer in text format)

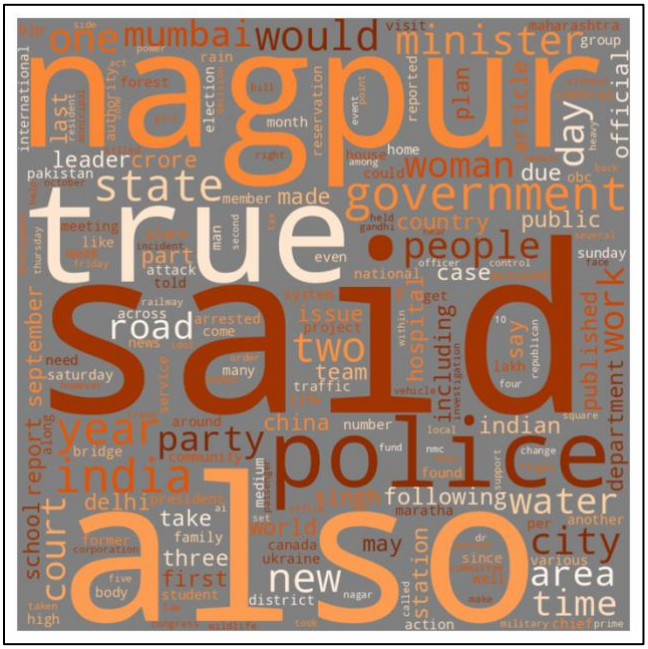
Exploratory Data Analysis

The main purpose of EDA is to help look at data before making any assumptions. It can help identify obvious errors, as well as better understand patterns within the data, detect outliers or anomalous events, find interesting relations among the variables.

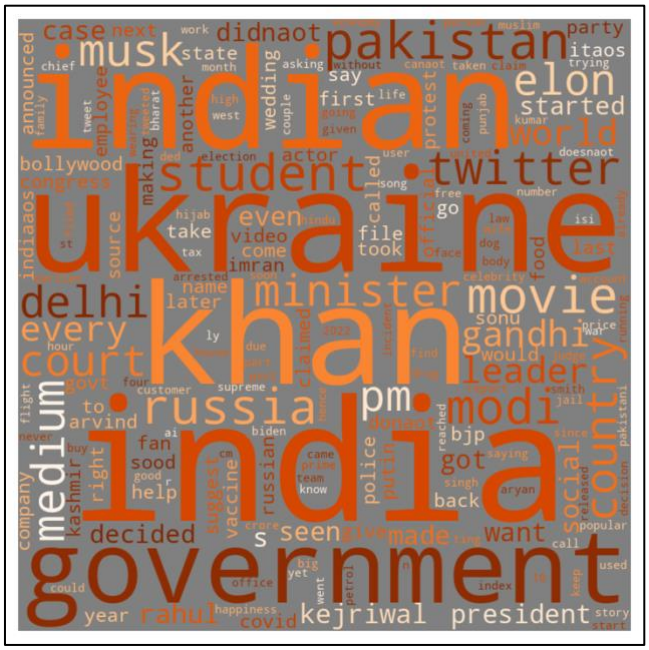
Exploratory Data Analysis, or EDA, is essential because it enables you to discover patterns, evaluate the quality of the data, and generate hypotheses from your dataset. This ensures that any subsequent analysis or decision-making is based on a solid understanding of the characteristics of the data as well as any potential difficulties.

The visualisations for this project and the analysis for each of them are as follows:

1) Word clouds: Word clouds or tag clouds are graphical representations of word frequency that give greater prominence to words that appear more frequently in a source text. For our project, we created two-word clouds, one that visualised words from the Real News Dataset, while the other visualised words from the Satirical News Dataset.



WordCloud for Real News



WordCloud for Satirical News

It is found that both the word clouds feature very similar words prominently, the most common being:

i) India

ii) Ukraine

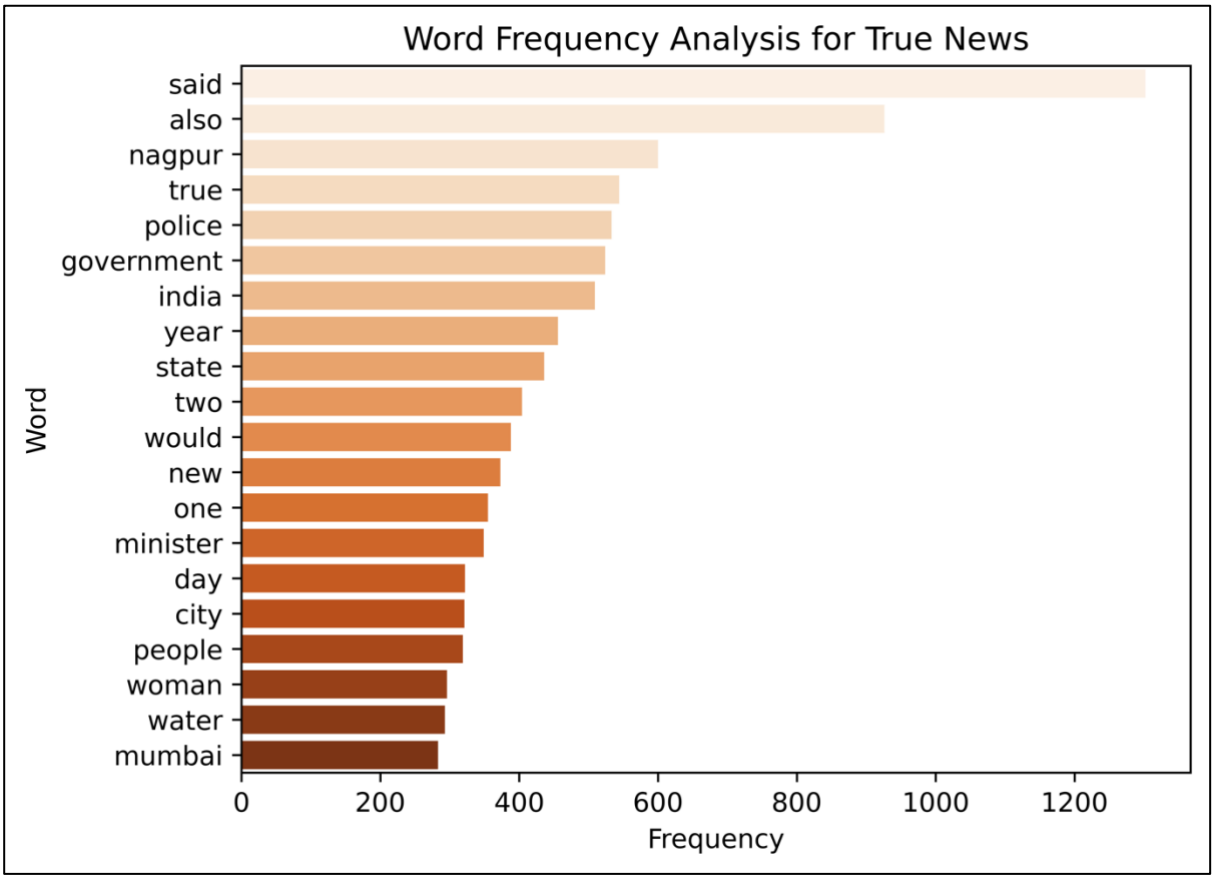
iii) People

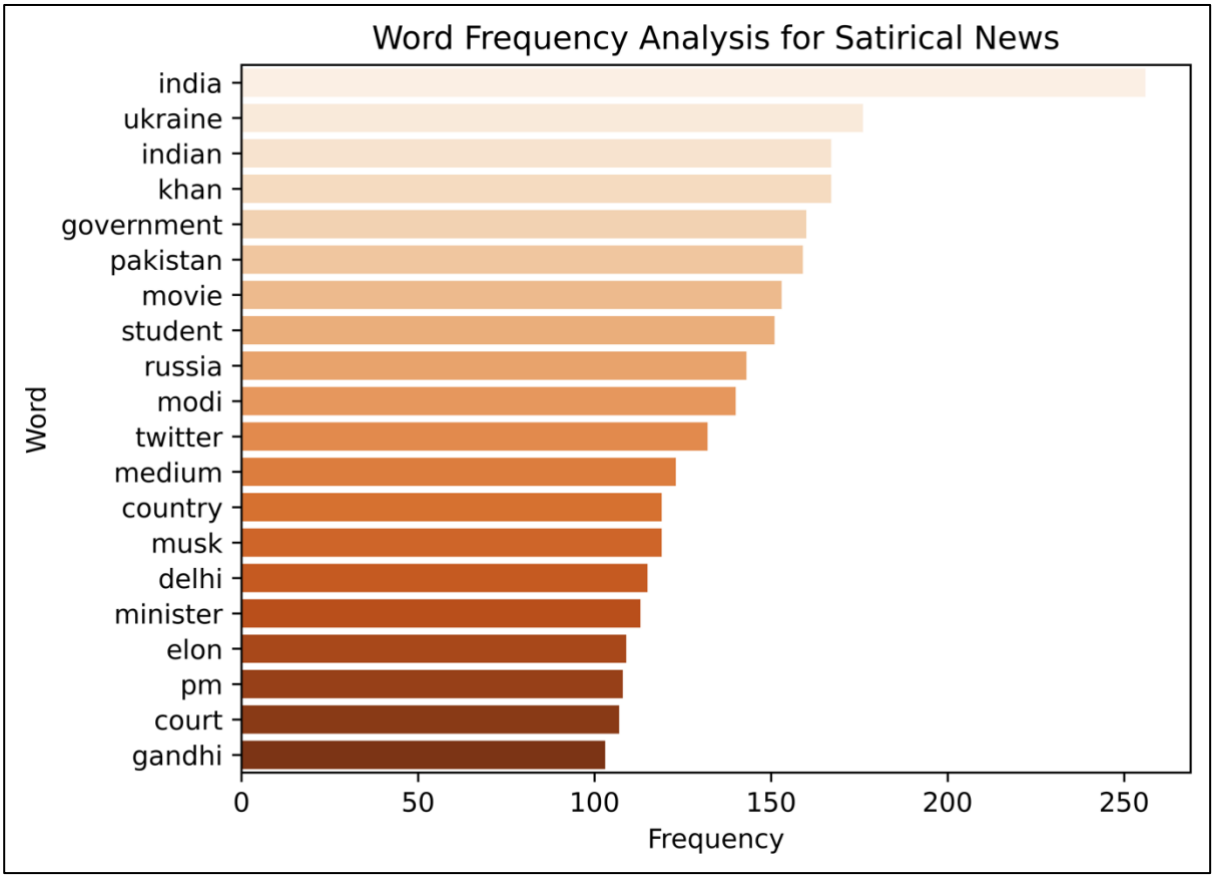
iv) Minister

v) Government

The similarity between these word clouds is attributed to the fact that both these 'news' outlets are of the same nationality, and were collected over the same timeframe. Since satire news articles parody real current events, the same topics – and hence the same words – were bound to be seen.

2) Word Frequency Analysis:





Word Frequency Analysis is a method used to determine the occurrence of words within a given text or dataset. It involves counting the number of times each word appears and then ranking the words based on their frequency. After removing stop words, we could see how often certain words appeared in the text. These words, as already mentioned in the word cloud provided insights into the actual content the data was talking about. By getting numerical values, it was easier to compare between the most popular words. By seeing exactly how much the frequency of words differed by, it provided further context as to what the main focus areas of our dataset were. For example, while 'Ukraine' was only one position behind 'India' as the most frequent word, their frequencies differed by 80 occurrences. So, while yes, 'Ukraine' was a heavy topic of discussion, being the second most frequent, it is no doubt 'India' that should be given the majority of our attention. In this case, while it may seem obvious that 'India' is the most frequent word by a mile for two Indian news outlets, other revelations were made by utilising the exact frequencies of words, like 'Khan' and 'Government' having equal occurrences. Hence, word frequency analysis is a significant step while carrying out Exploratory Data Analysis.

3) N-grams: N-grams are contiguous sequences of n items from a given sample of text or speech. By trial-and-error, we found that our data provided the most relevant insights with bi-grams (meaning two-word sequences).

	Ngrams	Count
898	(elon, musk)	102
619	(social, media)	89
1822	(rahul, gandhi)	81
1681	(arvind, kejriwal)	73
2039	(imran, khan)	66
7008	(sonu, sood)	54
1685	(kashmir, files)	52
142	(supreme, court)	43
2594	(pm, modi)	37
1285	(prime, minister)	33
8072	(aryan, khan)	32
723	(happiness, index)	31
3397	(bharat, jodo)	27
2038	(pm, imran)	27
3398	(jodo, yatra)	26

Bi-grams for Satirical News

	Ngrams	Count
197	(also, published)	183
198	(published, following)	183
199	(following, articles)	183
12	(nagpur, nagpur)	120
2298	(prime, minister)	103
971	(new, delhi)	85
821	(municipal, corporation)	68
723	(high, court)	66
10270	(state, government)	60
1047	(chief, minister)	58
2933	(last, year)	54
2138	(world, cup)	53
5293	(united, states)	51
8000	(mumbai, mumbai)	49
2231	(police, station)	48

Bi-grams for Real News

While individual words provide insight on specific topics of discussion, utilizing N-grams provides context to these points. It can also bring out words that did not place well in the aforementioned data analysis techniques, because they were used frequently in a phrase. A notable instance of this we found is how the word ‘happiness’ wasn’t a word that stood out in either the Word Cloud or during the Word Frequency Analysis, the bi-gram ‘happiness-index’ was visible among the most frequent ones. Since individual words are many but phrases do not occur quite as much, N-grams can provide deeper insights into our data.

Model Building

Moving on to the part where we create the model, the heart of the project. Because this is a binary classification project, the two main models we decided to try out were

1) kNN
2) Naïve Bayes

We also decided to apply these models to our data in three different ways:

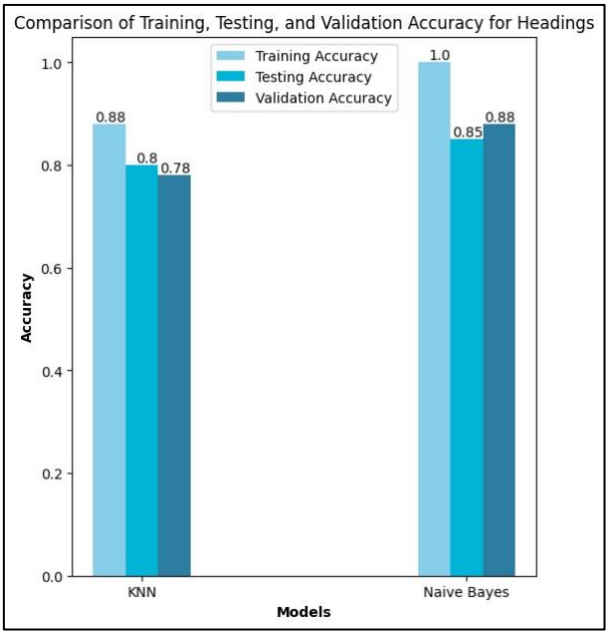
1) Headings of the Article only
2) Body of the Article only
3) Entire Combined Article (Heading and Body)

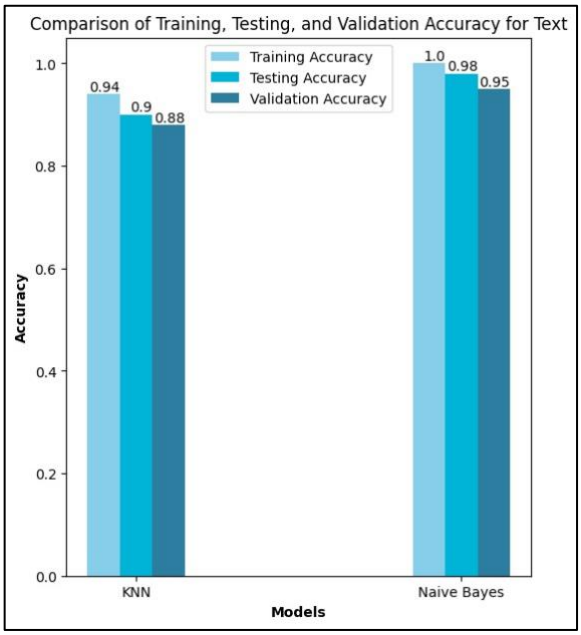
The reason to do this is to not only test out the accuracies for these models in three different ways, but having these models also makes the experience very user-centric.

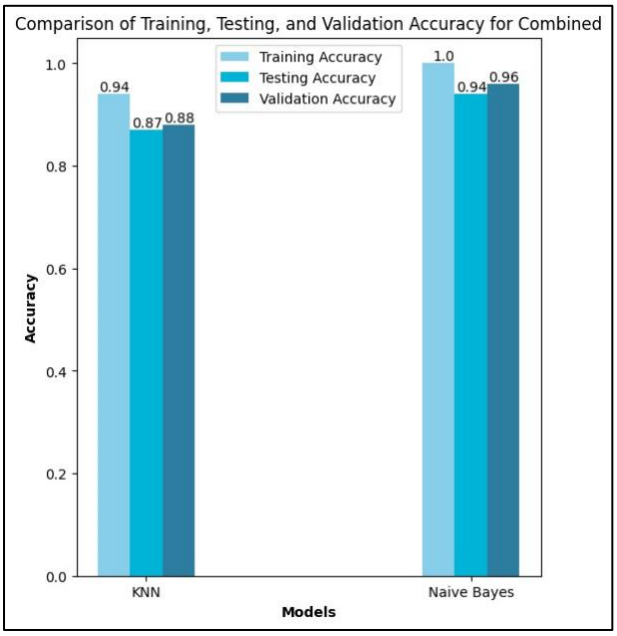
```
[ ] # Initialize classifiers
    knn_classifier_headings = KNeighborsClassifier(n_neighbors=7)
    naive_bayes_classifier_headings = MultinomialNB()

    # Training the classifiers
    knn_classifier_headings.fit(X_headings_train_tfidf, y_train1)
    naive_bayes_classifier_headings.fit(X_headings_train_tfidf, y_train1)

    # Predict on the test set
    knn_predictions_headings = knn_classifier_headings.predict(X_headings_test_tfidf)
    naive_bayes_predictions_headings = naive_bayes_classifier_headings.predict(X_headings_test_tfidf)
```

Training and Predicting Data on the Models

The accuracies for these models for the three different categories are as follows:

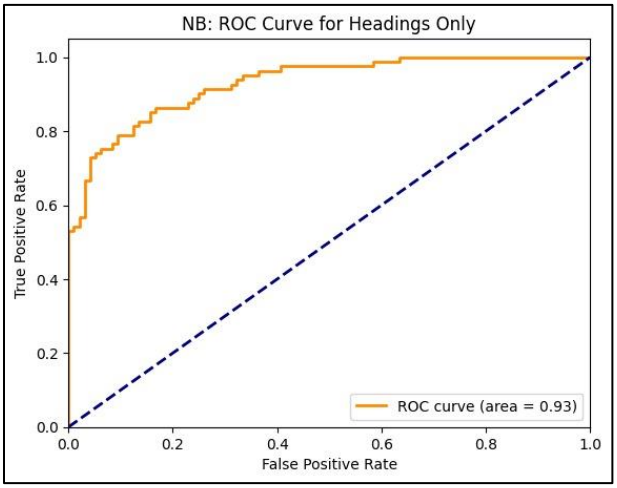


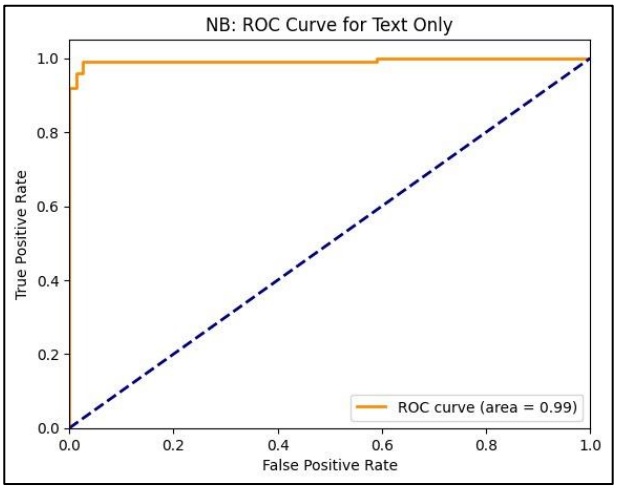


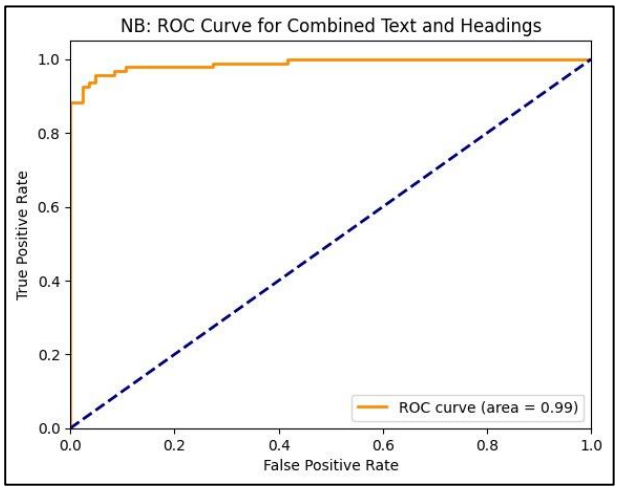


As is evident from the graphs, Naïve Bayes has performed better in every single category, even after altering parameters in kNN, like the number of Nearest Neighbors, or altering the split in the dataset, it was always outperformed by Naïve Bayes.

Moving forward, we will be focusing on the Naïve Bayes Model.







These are the AOC-ROC curves for the three categories. The model performs quite well, as is evident from the graphs, as well as the accuracy scores.

Now that the models have been built, they are then made into pickle files, so that they can be deployed.

Model Deployment

The model has been deployed using StreamLit. It is a user-friendly platform. As mentioned earlier, users have three options on how they want to input the articles- Headings only, Content only, and both Combined. Other than this, these three options can also be accessed by uploading a screenshot of the article. The user can screenshot only the article, only the content or the entire article. The model will extract the data from it using the Tesseract library, and input it into the model automatically.

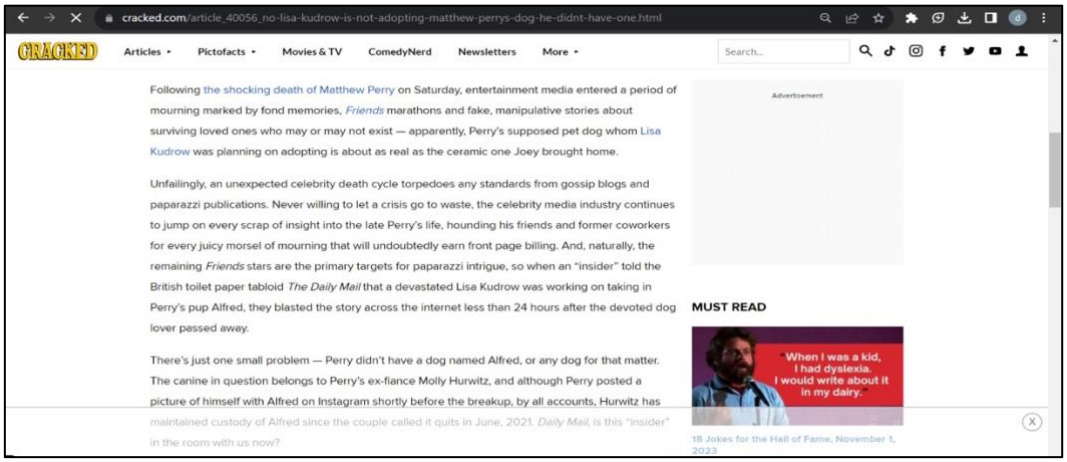
Front Page of Deployment

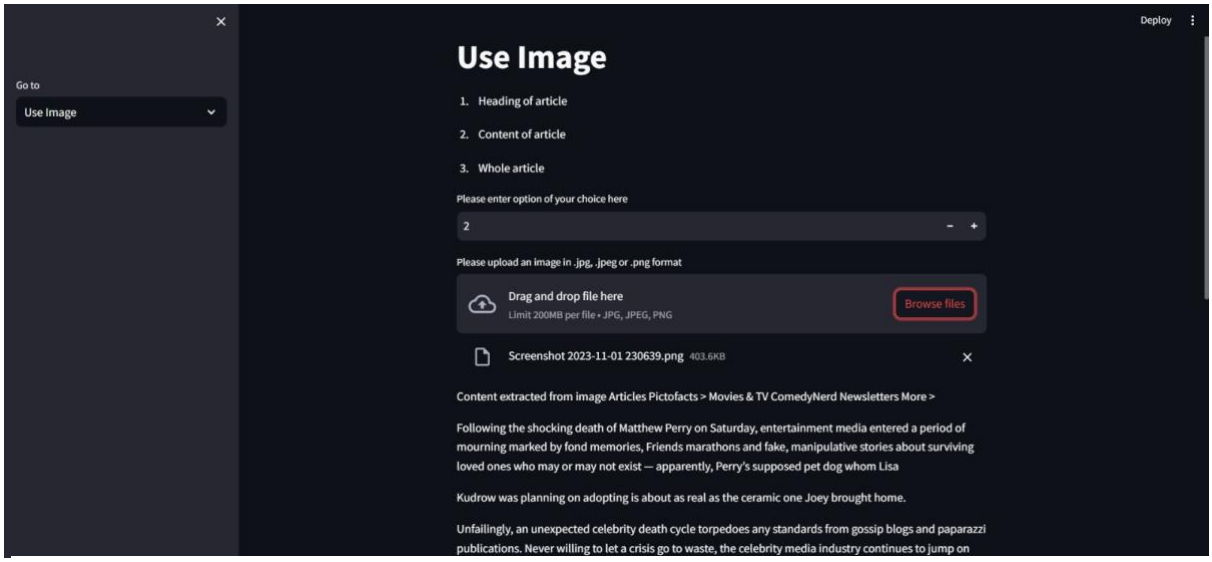
Using Text Data as input:

Jet Airways' Properties Worth ₹ 538 Crore Seized In Money Laundering Case

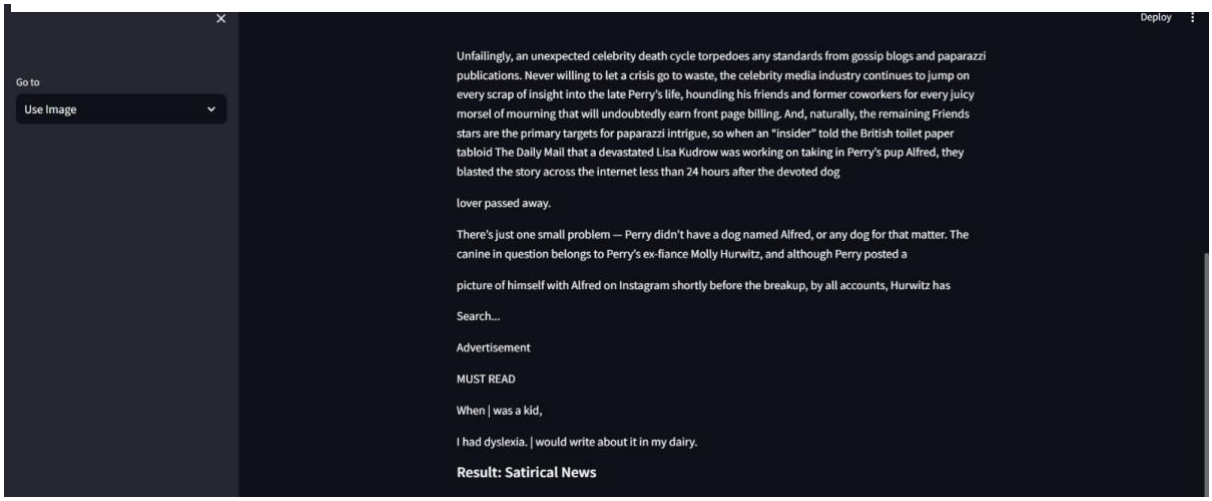
Output after inputting Real News Headline as Text Input

Using Screenshot of an Article as input:





Using Image as Input and how text is extracted from it



Getting Result 'Satirical News' from a Screenshot of Body of a Satirical News Article

Future Scope

1. Integration with Social Media Platforms: Integrating the project with popular social media platforms can enable real-time monitoring and fact-checking of news articles and posts shared on these platforms. This could involve developing browser extensions or mobile applications that provide immediate feedback on the authenticity of news content shared on social media.

2. Internationalization and Multilingual Support: Expanding the project's capabilities to support multiple languages and cater to diverse cultural contexts can broaden its impact on a global scale. Adapting the algorithms and methodologies to different languages and cultural nuances can significantly enhance the project's usability and effectiveness in various regions around the world.

3. Collaboration with News Organizations and Fact-Checking Initiatives: Establishing partnerships and collaborations with reputable news organizations and fact-checking initiatives can further validate and strengthen this project's credibility. This collaboration can foster a more comprehensive approach to combatting misinformation and disinformation while promoting media literacy and responsible journalism practices.

4. Educational Initiatives and Outreach Programs: Launching educational initiatives and outreach programs aimed at promoting media literacy and critical thinking can empower individuals to become more discerning consumers of news and information. This could involve organizing workshops, seminars, and online courses to educate the public about the nuances of distinguishing between real and satirical news.

5. Integration with Browser and News Aggregator Platforms: Integrating the project's functionalities directly into web browsers or news aggregator platforms can provide users with real-time feedback on the credibility of the news articles they come across online. This seamless integration can facilitate informed decision-making and contribute to a more informed and responsible online community.

By exploring these future scopes, the project can continue to make a substantial contribution to the ongoing efforts to mitigate the challenges posed by satirical news and promote a more informed and responsible media landscape.

Conclusion

In the end, the project's achievements have helped to foster transparency, accountability, and critical thinking in how people share and consume news. By promoting a more careful and discerning approach to media consumption, the project has played a role in creating a more knowledgeable and responsible society that can confidently navigate today's complex media landscape.

Python Notebook Link:

https://colab.research.google.com/drive/1J0HV7_kAa5cCv5yE-3hDvtVObCl506Bu?usp=sharing/

References

1) Nirav Shah M, Ganatra A. A systematic literature review and existing challenges toward fake news detection models. Soc Netw Anal Min. 2022;12(1):168. doi: 10.1007/s13278-022-00995-5. Epub 2022 Nov 14. PMID: 36407554; PMCID: PMC9663194.
2) Balshetwar, S.V., RS, A. & R, D.J. Fake news detection in social media based on sentiment analysis using classifier techniques. Multimed Tools Appl 82, 35781–35811 (2023). https://doi.org/10.1007/s11042-023-14883-3