# **FAKE NEWS DETECTION USING NLP**

912421104031:S.Priyadharshini

Phase 4 development part 2

Project: **Continue building the fake news detection model by applying NLP techniques and training a classification model.**

* **Text Preprocessing and Feature Extraction**
* **Model training and evaluation**



INTRODUCTION:

Fake news on different platforms is spreading widely and is a matter of serious concern, as it causes social wars and permanent breakage of the bonds established among people. A lot of research is already going on focused on the classification of fake news.

Here we will try to solve this issue with the help of machine learning in [Python](https://www.geeksforgeeks.org/python-programming-language/).

Before starting the code, download the dataset by clicking the [link](https://drive.google.com/file/d/1q5jpI5M1EA9x3YPrLupmiu3gffkmGlHj/view?usp=sharing).

**Steps to be followed**

1. Importing Libraries and Datasets
2. Data Preprocessing
3. Preprocessing and analysis of News column
4. Converting text into Vectors
5. Model training, Evaluation, and Prediction

**DATA SOURCE: Dataset Link:**[**https://www.kaggle.com/datasets/clmentbisaillon/fake-and-real-news-dataset**](https://www.kaggle.com/datasets/clmentbisaillon/fake-and-real-news-dataset)

Creating a fake news detection project can be a valuable and socially responsible endeavor. Fake news has become a significant issue in the age of the internet, and developing a system to identify and combat it can help promote accurate information dissemination. Here's a general outline of the steps to create a fake news detection project:

1. **Define the Scope and Objectives:** Clearly outline the goals of your project. Are you building a web application, a browser extension, or an API for detecting fake news? Who is your target audience, and what kind of fake news are you focusing on (e.g., political, health, entertainment)?
2. **Data Collection:** You'll need a reliable dataset of both real and fake news articles. Datasets like "Fake News Challenge" or "LIAR" can be a good starting point. Scraping news articles from various sources can also help to create a diverse dataset.
3. **Preprocessing:** Clean and preprocess the text data. This typically includes text lowercasing, tokenization, removing stop words, stemming, and lemmatization.
4. **Feature Extraction:** Convert text data into numerical features. Common techniques include TF-IDF (Term Frequency-Inverse Document Frequency) and Word Embeddings (Word2Vec, GloVe).
5. **Model Selection:** Choose a machine learning model for fake news detection. Common choices include:
   * Naive Bayes
   * Support Vector Machines
   * Recurrent Neural Networks (RNN)
   * Convolutional Neural Networks (CNN)
   * Transformers (e.g., BERT, GPT-3)
6. **Training and Evaluation:** Split your dataset into training and testing sets. Train your chosen model and evaluate its performance using metrics like accuracy, precision, recall, and F1-score.
7. **Fine-tuning:** Experiment with hyperparameter tuning and different model architectures to improve your model's accuracy.
8. **Cross-validation:** Implement cross-validation to ensure the model's generalizability.
9. **Deployment:** If you're building a web application or API, deploy your model to a server or a cloud platform. Popular choices for deployment include AWS, Azure, and Google Cloud.
10. **User Interface:** Create a user-friendly interface where users can input news articles or URLs for analysis.
11. **Testing and Validation:** Test your system extensively with different types of news articles. Ensure that the false-positive and false-negative rates are within acceptable limits.
12. **Feedback Mechanism:** Implement a feedback system for users to report false positives/negatives, which can help improve the model over time.
13. **Ethical Considerations:** Be aware of potential biases in your data and model. Make sure your system doesn't discriminate based on political, social, or other factors.
14. **Continuous Improvement:** Regularly update your model to adapt to changing trends in fake news and improve its accuracy.
15. **Education and Awareness:** Consider adding educational components to your project to help users understand how fake news works and how to identify it themselves.

Remember that fake news detection is a challenging problem, and no model is perfect. It's essential to continuously work on improving your model and stay up-to-date with the latest research in natural language processing and machine learning. Additionally, ethics, user privacy, and transparency should be at the forefront of your project.

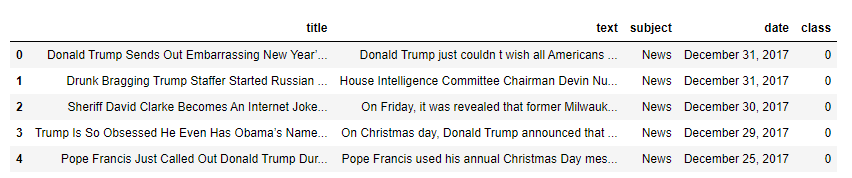
**PROGRAM:**

|  |
| --- |
| **import** pandas as pd  **import** seaborn as sns  **import** matplotlib.pyplot as plt |

Let’s import the downloaded dataset.

|  |
| --- |
| data **=** pd.read\_csv('News.csv',index\_col**=**0)  data.head() |

**Output :**



**Data preprocessing**

The shape of the dataset can be found by the below code.

|  |
| --- |
| data.shape |

**Output:**

(44919, 5)

As the title, subject and date column will not going to be helpful in identification of the news. So, we can drop these column.

|  |
| --- |
| data **=** data.drop(["title", "subject","date"], axis **=** 1) |

Now, we have to check if there is any null value (we will drop those rows)

|  |
| --- |
| data.isnull().sum() |

**Output:**

text 0

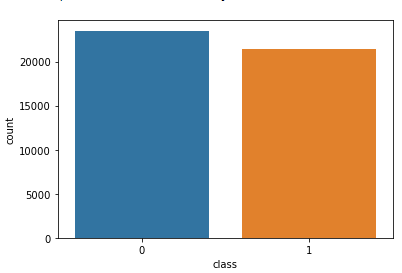
class 0

|  |
| --- |
| # Shuffling  data **=** data.sample(frac**=**1)  data.reset\_index(inplace**=**True)  data.drop(["index"], axis**=**1, inplace**=**True) |

Now Let’s explore the unique values in the each category using below code.

|  |
| --- |
| sns.countplot(data**=**data,                x**=**'class',                order**=**data['class'].value\_counts().index) |

**Output:**



**Preprocessing and analysis of News column**

Firstly we will remove all the stopwords, punctuations and any irrelevant spaces from the text. For that [NLTK](https://www.geeksforgeeks.org/tokenize-text-using-nltk-python/) Library is required and some of it’s module need to be downloaded. So, for that run the below code.

|  |
| --- |
| **from** tqdm **import** tqdm  **import** re  **import** nltk  nltk.download('punkt')  nltk.download('stopwords')  **from** nltk.corpus **import** stopwords  **from** nltk.tokenize **import** word\_tokenize  **from** nltk.stem.porter **import** PorterStemmer  **from** wordcloud **import** WordCloud |

Once we have all the required modules, we can create a function name preprocess text. This function will preprocess all the data given as input.

|  |
| --- |
| **def** preprocess\_text(text\_data):      preprocessed\_text **=** []    **for** sentence **in** tqdm(text\_data):          sentence **=** re.sub(r'[^\w\s]', '', sentence)          preprocessed\_text.append(' '.join(token.lower()  **for** token **in** str(sentence).split()  **if** token **not** **in** stopwords.words('english')))    **return** preprocessed\_text |

 To implement the function in all the news in the text column, run the below command.

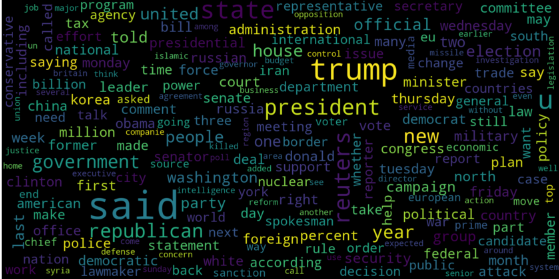
|  |
| --- |
| preprocessed\_review **=** preprocess\_text(data['text'].values)  data['text'] **=** preprocessed\_review |

*This command  will take some time (as the dataset taken is very large).*

Let’s visualize the WordCloud for fake and real news separately.

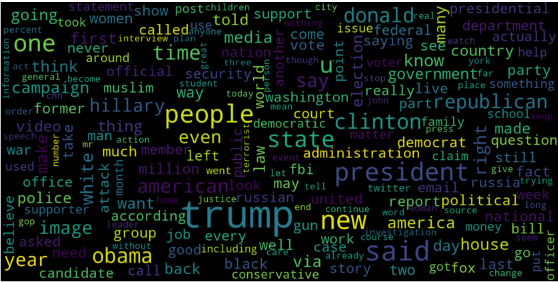
|  |
| --- |
| # Real  consolidated **=** ' '.join(      word **for** word **in** data['text'][data['class'] **==** 1].astype(str))  wordCloud **=** WordCloud(width**=**1600,                        height**=**800,                        random\_state**=**21,                        max\_font\_size**=**110,                        collocations**=**False)  plt.figure(figsize**=**(15, 10))  plt.imshow(wordCloud.generate(consolidated), interpolation**=**'bilinear')  plt.axis('off')  plt.show() |

**Output :**



|  |
| --- |
| # Fake  consolidated **=** ' '.join(      word **for** word **in** data['text'][data['class'] **==** 0].astype(str))  wordCloud **=** WordCloud(width**=**1600,                        height**=**800,                        random\_state**=**21,                        max\_font\_size**=**110,                        collocations**=**False)  plt.figure(figsize**=**(15, 10))  plt.imshow(wordCloud.generate(consolidated), interpolation**=**'bilinear')  plt.axis('off')  plt.show() |

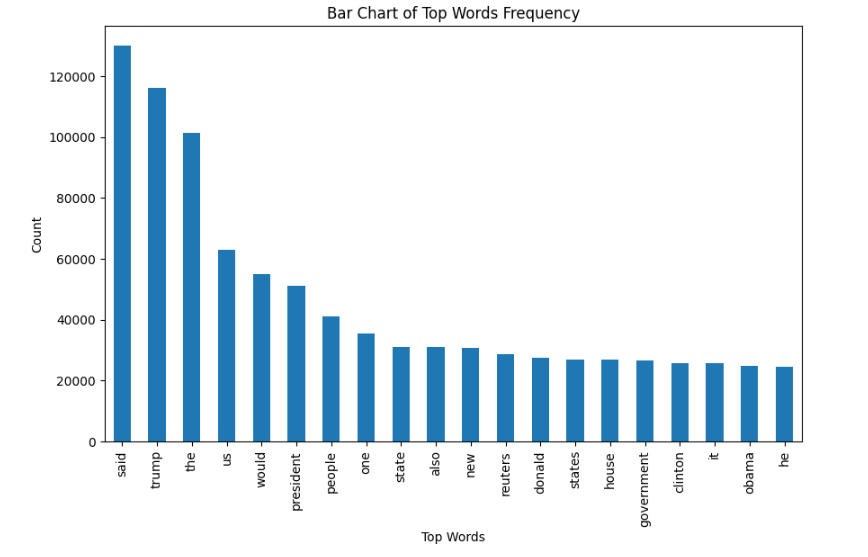
**Output :**



Now, Let’s plot the bargraph of the top 20 most frequent words.

|  |
| --- |
| **from** sklearn.feature\_extraction.text **import** CountVectorizer      **def** get\_top\_n\_words(corpus, n**=**None):      vec **=** CountVectorizer().fit(corpus)      bag\_of\_words **=** vec.transform(corpus)      sum\_words **=** bag\_of\_words.sum(axis**=**0)      words\_freq **=** [(word, sum\_words[0, idx])  **for** word, idx **in** vec.vocabulary\_.items()]      words\_freq **=** sorted(words\_freq, key**=lambda** x: x[1],                          reverse**=**True)  **return** words\_freq[:n]      common\_words **=** get\_top\_n\_words(data['text'], 20)  df1 **=** pd.DataFrame(common\_words, columns**=**['Review', 'count'])    df1.groupby('Review').sum()['count'].sort\_values(ascending**=**False).plot(      kind**=**'bar',      figsize**=**(10, 6),      xlabel**=**"Top Words",      ylabel**=**"Count",      title**=**"Bar Chart of Top Words Frequency"  ) |

**Output :**



**Converting text into Vectors**

Before converting the data into vectors, split it into train and test.

|  |
| --- |
| **from** sklearn.model\_selection **import** train\_test\_split  **from** sklearn.metrics **import** accuracy\_score  **from** sklearn.linear\_model **import** LogisticRegression    x\_train, x\_test, y\_train, y\_test **=** train\_test\_split(data['text'],                                                      data['class'],                                                      test\_size**=**0.25) |

Now we can convert the training data into vectors using [TfidfVectorizer](https://www.geeksforgeeks.org/understanding-tf-idf-term-frequency-inverse-document-frequency/).

|  |
| --- |
| **from** sklearn.feature\_extraction.text **import** TfidfVectorizer    vectorization **=** TfidfVectorizer()  x\_train **=** vectorization.fit\_transform(x\_train)  x\_test **=** vectorization.transform(x\_test) |

**Model training, Evaluation, and Prediction**

Now, the dataset is ready to train the model.

For training we will use [Logistic Regression](https://www.geeksforgeeks.org/understanding-logistic-regression/) and evaluate the prediction accuracy using accuracy\_score.

|  |
| --- |
| **from** sklearn.linear\_model **import** LogisticRegression    model **=** LogisticRegression()  model.fit(x\_train, y\_train)    # testing the model  print(accuracy\_score(y\_train, model.predict(x\_train)))  print(accuracy\_score(y\_test, model.predict(x\_test))) |

**Output :**

0.993766511324171

0.9893143365983972

Let’s train with [Decision Tree](https://www.geeksforgeeks.org/decision-tree/) Classifier.

|  |
| --- |
| **from** sklearn.tree **import** DecisionTreeClassifier    model **=** DecisionTreeClassifier()  model.fit(x\_train, y\_train)    # testing the model  print(accuracy\_score(y\_train, model.predict(x\_train)))  print(accuracy\_score(y\_test, model.predict(x\_test))) |

**Output :**

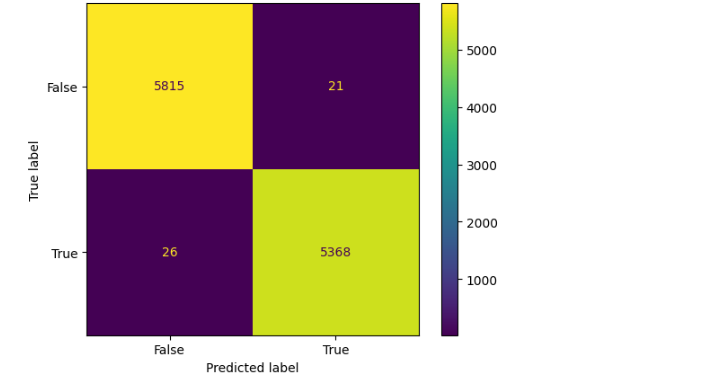
0.9999703167205913

0.9951914514692787

The confusion matrix for Decision Tree Classifier can be implemented with the code below.

|  |
| --- |
| # Confusion matrix of Results from Decision Tree classification  **from** sklearn **import** metrics  cm **=** metrics.confusion\_matrix(y\_test, model.predict(x\_test))    cm\_display **=** metrics.ConfusionMatrixDisplay(confusion\_matrix**=**cm,                                              display\_labels**=**[False, True])    cm\_display.plot()  plt.show() |

**Output :**



**Conclusion**

Decision Tree Classifier and Logistic regression are performing well.

**Advantages :**

Fake news detection offers several advantages that contribute to a more informed and responsible information environment:

1. **Promotes Information Accuracy:** Fake news detection helps maintain the integrity of information dissemination by reducing the spread of false or misleading information.
2. **Enhances Trust:** It helps rebuild and maintain trust in journalism and reliable news sources, as it allows users to distinguish between credible and non-credible sources.
3. **Informed Decision-Making:** It enables individuals to make more informed decisions in various aspects of life, such as politics, health, finance, and personal safety.
4. **Social Stability:** By curbing the influence of fake news, society becomes less vulnerable to manipulation and polarization, leading to more stable and cohesive communities.
5. **Economic Impact:** Reducing the influence of fake news can have a positive impact on the economy by preventing market volatility triggered by false financial information.
6. **Health and Safety:** In healthcare, it can prevent the spread of dangerous misinformation, while in emergencies, it can provide accurate information to ensure public safety.
7. **Reputation Protection:** For individuals and organizations, fake news detection can protect reputations by identifying and responding to false claims or defamatory content.
8. **Journalistic Integrity:** It supports the work of professional journalists and publications, reinforcing the importance of quality reporting.
9. **Prevents Manipulation:** Reducing the impact of fake news prevents malicious actors from manipulating public opinion, elections, or events for their benefit.
10. **Educational Tool:** Fake news detection can be used as an educational tool to teach critical thinking and media literacy, empowering people to be discerning consumers of news.
11. **Filtering Content:** It helps platforms and social media networks filter out or label fake news, limiting its visibility and impact.
12. **Legal Action:** Identifying and addressing fake news can lead to legal actions against those who intentionally spread false information with malicious intent.
13. **Improved User Experience:** It enhances the user experience on online platforms by reducing the presence of deceptive and harmful content.
14. **Transparency:** The process of fake news detection can be transparent, showing users the criteria and methods used to assess the credibility of news, which promotes trust.
15. **Research Opportunities:** Fake news detection research can lead to advancements in natural language processing, machine learning, and data analysis techniques.
16. **International Relations:** Reducing the spread of fake news can help in international relations by preventing the escalation of conflicts based on false information.

Overall, fake news detection is a vital component of maintaining a well-informed and stable society in the digital age. It has far-reaching positive effects on individual well-being, community cohesion, and the broader information ecosystem.

**DIS-ADVANTAGES:**

While fake news detection is crucial for maintaining the integrity of information dissemination and making informed decisions, there are also some potential disadvantages and challenges associated with these efforts:

1. **False Positives:** Fake news detection algorithms may incorrectly label legitimate news articles as fake, leading to a loss of credibility for accurate sources.
2. **Censorship Concerns:** Aggressive fake news detection measures could inadvertently censor or stifle free speech, limiting the ability to express diverse opinions and ideas.
3. **Bias in Algorithms:** Machine learning models used for fake news detection can inherit biases present in training data, potentially leading to discrimination against certain groups or viewpoints.
4. **Adversarial Manipulation:** Those who spread fake news are constantly evolving their tactics, making it difficult for detection systems to keep up.
5. **Privacy Concerns:** Collecting and analyzing large volumes of data to identify fake news can raise privacy concerns, especially when it comes to user data and online activity.
6. **Resource-Intensive:** Implementing and maintaining effective fake news detection systems can be resource-intensive in terms of computing power and human oversight.
7. **Complexity and Overhead:** The development and maintenance of fake news detection systems can be complex and may require continuous adjustments to remain effective.
8. **Legitimacy Challenges:** Deciding what constitutes fake news can be a contentious issue, as different people and organizations have varying definitions of credibility.
9. **Cat-and-Mouse Game:** Fake news creators often adapt their strategies, making it challenging to stay ahead in the battle against misinformation.
10. **Limitations in Multimodal Content:** Detecting fake news in images and video is more challenging than text and requires advanced technology.
11. **User Privacy Concerns:** Analyzing user behavior and interactions for fake news detection may raise concerns about the surveillance of individuals.
12. **Legal and Ethical Issues:** Determining liability and the legal consequences for spreading fake news can be complex, as it varies by jurisdiction.
13. **Ineffectiveness for Non-Text Content:** Fake news detection systems primarily designed for textual content may struggle to identify misinformation in non-text formats, such as memes or audio.
14. **Lack of Universal Standards:** There is no universally accepted standard for fake news detection, making it challenging to create a consistent and effective approach.
15. **Impact on Small Publishers:** Smaller news outlets may be disproportionately affected if they are falsely identified as spreading fake news, potentially impacting their viability.
16. **Overreliance on Technology:** Relying solely on technology for fake news detection can lead to complacency among users and a reduced emphasis on media literacy and critical thinking.

Balancing the advantages of fake news detection with these disadvantages and challenges is crucial to create a robust and ethical approach to combating misinformation in the digital age.

**CONCLUSION FOR PHASE 4:**

In conclusion, fake news detection is a critical endeavor with both significant advantages and notable challenges. In an era where information spreads rapidly and widely, distinguishing between credible and misleading content is vital for the well-being of individuals, communities, and society at large. Fake news detection provides several benefits, including promoting information accuracy, enhancing trust in journalism, empowering informed decision-making, and safeguarding social stability.

However, the journey to effective fake news detection is not without its pitfalls. Issues like false positives, censorship concerns, biases in algorithms, and the constant evolution of adversarial tactics can present significant challenges. Additionally, privacy, resource requirements, and complex legal and ethical dilemmas must be addressed in the development and deployment of fake news detection systems.