**FAKE NEWS DETECTION USING NLP**

**TEAM MEMBER**:

1.611721104054-M. MOURIYA

2.611721104058-D. PAPETHA

3.611721104073-S. SABITHRIJA

4.611721104701-S. ARTHI

5.611721104702-M.P. PREETHI

**Project Title**: Fake News Detection

**Phase 4**: Development Part 2

Topic: Continue building the fake news detection model by feature engineering, model training, and evaluation



**FAKE NEWS DETECTION:**

**Introduction:**

1. \*Data Collection\*: Gather a dataset of label news articles, where each article is label as either real or fake.

2. \*Data Preprocessing\*:

- Tokenization: Break down the text into individual words or tokens.

- Stop word Removal: Eliminate common words (e.g., "the", "is") that don't carry much information.

- Lemmatization/Stemming: Reduce words to their base or root form.

3. \*Feature Extraction\*:

- Convert the processed text into numerical form that can be fed into a neural network. This is usually done using techniques like TF-IDF (Term Frequency-Inverse Document Frequency) or word embeddings (e.g., Word2Vec, Glove).

4. \*Model Architecture\*:

- Design a deep learning architecture suitable for text classification. Common choices include:

- Recurrent Neural Networks (RNNs) or Long Short-Term Memory Networks (LSTMs) for sequential data.

- Convolutional Neural Networks (CNNs) with 1D convolutions for text.

5. \*Training\*:

- Split the dataset into training and validation sets.

- Feed the processed text data into the model and train it using appropriate loss functions (e.g., binary cross-entropy).

6. \*Validation and Testing\*:

- Evaluate the model on a validation set to tune hyperparameters.

- Finally, test the model on a separate test set to get an unbiased evaluation.

7. \*Model Evaluation\*:

- Metrics like accuracy, precision, recall, and F1-score are used to assess the performance of the model

Remember, the quality and diversity of your dataset, as well as the architecture and hyperparameters of your model, play crucial roles in the success of this task. Additionally, it's improved results

**Fake news detection using Natural Language Processing (NLP) involves using computational techniques to identify and classify misleading or false information in textual content. Here's a brief overview of the process:**

**1. \*Data Collection and Preprocessing\*:**

**- Gather a dataset containing both fake and real news articles.**

**- Preprocess the text by removing special characters, numbers, and stop words. Tokenize the text into words or phrases.**

**2. \*Feature Extraction\*:**

**- Extract relevant features from the preprocessed text. This may include word frequencies, n-grams, or more advanced techniques like word embeddings.**

**3. \*Model Training\*:**

**- Utilize machine learning or deep learning algorithms to build a classification model. Common techniques include:**

**- \*Traditional Machine Learning\*: Support Vector Machines (SVM), Logistic Regression, etc.**

**- \*Deep Learning\*: Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), and more recently, Transformer-based models like BERT.**

**4. \*Labeling and Validation\*:**

**- Label the data with appropriate classes (e.g., fake or real news).**

**- Split the dataset into training and testing sets to evaluate the model's performance.**

**5. \*Model Evaluation\*:**

**- Use metrics like accuracy, precision, recall, and F1-score to assess the model's effectiveness in distinguishing fake news from real news.**

**6. \*Fine-tuning and Optimization\*:**

**- Fine-tune hyperparameters, experiment with different algorithms, and possibly employ techniques like cross-validation to improve performance.**

**7. \*Deployment\*:**

**Once the model is trained and evaluated, it can be deployed in real-time applications to automatically flag potentially fake news articles.**

**8. \*Continuous Monitoring and Updating\*:**

**- Regularly update the model with new data to adapt to evolving patterns of fake news.**

**It's important to note that while NLP-based techniques are powerful, they are not foolproof and may have limitations. Staying vigilant and combining NLP with other strategies for fact-checking and verification is crucial in the fight against misinformation.**

**Natural Language Processing (NLP)**

**Natural Language Processing (NLP) is a field of artificial intelligence (AI) that focuses on enabling computers to understand, interpret, and generate human language in a way that is meaningful and contextually relevant. It involves the interaction between computers and natural language, allowing machines to process, analyze, and generate text or speech.**

**Key components and tasks in NLP include:**

**1. \*Tokenization\*:**

**- Breaking down a text into individual words or phrases, known as tokens. This is a fundamentalstep in text processing.**

**2. \*Part-of-Speech Tagging\*:**

**- Assigning a grammatical label (like noun, verb, adjective, etc.) to each word in a sentence. This helps in understanding the syntactic structure.**

**3. \*Parsing\*:**

**- Analyzing the grammatical structure of a sentence to determine how different words relate to each other. It helps in understanding the sentence's syntax and semantics.**

**4. \*\*Named Entity Recognition (NER)\*\*:**

**- Identifying and categorizing entities in a text, such as names of people, places, organizations, etc.**

**5. \*Sentiment Analysis\*:**

**- Determining the sentiment or emotional tone expressed in a piece of text, which could be positive, negative, or neutral.**

**6. \*Machine Translation\*:**

**- Automatically converting text from one language to another.**

**7. \*Topic Modeling\*:**

**- Identifying the main topics or themes in a collection of documents.**

**8. \*Text Summarization\*:**

**- Generating concise and coherent summaries of longer texts.**

**9. \*Question Answering\*:**

**- Providing answers to questions posed in natural language.**

**10. \*Chatbots and Conversational Agents\*:**

**- Creating systems that can engage in conversations with humans in a natural and meaningful way.**

**11. \*Text Generation\*:**

**- Creating coherent and contextually relevant text, which can range from simple responses to complex articles.**

**NLP is used in various applications like search engines, virtual assistants (like Siri or Alexa), sentiment analysis for social media monitoring, language translation services, and many more. It plays a crucial role in bridging the gap between human communication and computer understanding, enabling a wide range of applications in the digital world.**

Fake news detection is the process of identifying and distinguishing false or misleading information from legitimate and accurate news sources. It's a critical task in today's information age, where the spread of misinformation can have significant social, political, and economic consequences.

Here are some common techniques and strategies used in fake news detection:

1. \*Source Verification\*:

- Assess the credibility and reputation of the source. Established and reputable news organizations are generally more reliable.

2. \*Fact-Checking\*:

- Cross-verify information with trusted fact-checking organizations that specialize in debunking false claims.

3. \*Stance Detection\*:

- Determine the bias or political stance of a news article. This helps in understanding potential motivations behind the content.

4. \*Pattern Recognition\*:

- Identify common characteristics of fake news articles, such as sensational language, excessive use of capital letters or exclamation marks, and misleading headlines.

5. \*Content Analysis\*:

- Analyze the content for inconsistencies, logical fallacies, and unsupported claims.

6. \*Metadata Examination\*:

- Look at additional information about the article, such as publication date, authorship, and domain name, to spot potential red flags.

7. \*Social Media Analysis\*:

- Monitor social media platforms for trends and discussions around specific news items. Assess the credibility of the sources sharing the information.

8. \*\*Natural Language Processing (NLP)\*\*:

- Utilize NLP techniques to analyze the linguistic features of the text, including sentiment, syntax, and semantic structure, which can provide clues about the authenticity of the content.

9. \*Machine Learning Models\*:

- Train models on labeled datasets containing both real and fake news to automatically classify new articles.

**Given dataset:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | title | text | | subject | date | | ﬂag | |
| 0 | Donald Trump | Donald Trump just | | News | December 31, 2017 | | 0 | |
|  | Sends Out  Embarrass | couldn t  wish all | |  |  | |  | |
|  | ing New Year’... | Americans  ... | |  |  | |  | |
| 1 | Drunk Bragging | House Intelligenc | | News | December 31, 2017 | | 0 | |
|  | Trump  StaGer | e  Committe | |  |  | |  | |
|  |  | e | |  |  | |  | |
|  | Started  Russian | | Chairman  Devin Nu |  | |  | |  | |
| 2 | SheriG David Clarke Becomes An Internet Joke... | | On Friday, it was revealed that former Milwauk... | News | | December 30, 2017 | | 0 | |
| 3 | Trump Is So Obsessed He Even Has Obama’s Name... | | On Christmas day, Donald Trump announce d that ... | News | | December 29, 2017 | | 0 | |
| 4 | Pope Francis Just Called Out Donald Trump Dur... | | Pope Francis used his annual Christmas Day mes... | News | | December 25, 2017 | | 0 | |
| ... | ... | | ... | ... | | ... | | ... | |
| 23476 | McPain: John | | 21st Century | Middle- east | | January 16, 2016 | | 0 | |
|  | McCain  Furious | | Wire says  As 21WIRE |  | |  | |  | |
|  | That Iran Treated ... | | reported earl... |  | |  | |  | |
| 23477 | JUSTICE?  Yahoo | | 21st Century | Middle- east | | January 16, 2016 | | 0 | |
|  | Settles E-  mail | | Wire says  It s a |  | |  | |  | |
|  | Privacy Class-ac... | | familiar theme. ... |  | |  | |  | |
|  |  | |  |  | |  | |  | |

***100 rows\*6 columns***

**Overview of the process**:

Detecting fake news involves several steps:

1. \*Data Collection\*: Gathering a diverse set of news articles from various sources to create a comprehensive dataset.

2. \*Preprocessing\*: Cleaning and preparing the data for analysis. This involves tasks like removing special characters, stop words, and converting text to lowercase.

3. \*Feature Extraction\*: Transforming the text into a numerical format that can be used for machine learning. Techniques like TF-IDF (Term Frequency-Inverse Document Frequency) or word embeddings like Word2Vec are commonly used.

4. \*Labeling\*: Annotating the data with labels indicating whether each article is real or fake. This is typically done through human reviewers or using existing fact-checking databases.

5. \*Model Training\*: Using a machine learning algorithm (e.g., a classifier like Random Forest, Support Vector Machine, or a deep learning model like a neural network) to learn the patterns that differentiate real and fake news

.

6. \*Validation and Testing\*: Evaluating the model's performance on a separate set of data (validation set) that it hasn't seen during training. This helps assess if the model generalizes well.

7. \*Model Evaluation\*: Assessing the model's performance using metrics like accuracy, precision, recall, F1-score, etc.

8. \*\*Fine-tuning (Optional)\*\*: Adjusting model hyperparameters or trying different algorithms to improve performance.

9. \*Deployment\*: Integrating the model into a system or application where it can be used to analyze news articles in real-time.

10. \*Continuous Monitoring\*: Regularly updating the model with new data and retraining it to adapt to evolving forms of fake news.

**PROCEDDUER:**

The procedure for detecting fake news typically involves the following steps:

1. \*Data Collection\*: Gather a diverse dataset of news articles, containing both real and potentially fake news. Ensure the dataset spans various sources and topics.

2. \*Preprocessing\*:

- \*Text Cleaning\*: Remove any special characters, punctuation, and unnecessary whitespace.

- \*Lowercasing\*: Convert all text to lowercase for consistency.

- \*Tokenization\*: Break the text into individual words or tokens.

- \*Stopword Removal\*: Eliminate common words (e.g., "and," "the") that don't carry much information.

3. \*Feature Extraction\*:

- \*\*Bag of Words (BoW)\*\*: Represent the text as a matrix of word frequencies.

- \*\*TF-IDF (Term Frequency-Inverse Document Frequency)\*\*: Weigh the importance of words in a document relative to their frequency across the entire corpus.

- \*Word Embeddings\*: Transform words into numerical vectors using techniques like Word2Vec or GloVe.

4. \*Labeling\*:

- Assign labels indicating whether each article is real or fake. This could be done through manual fact-checking or using pre-existing labeled datasets.

5. \*Model Selection\*:

- Choose a suitable machine learning algorithm (e.g., Random Forest, Support Vector Machine) or deep learning approach (e.g., recurrent neural networks, transformers).

6. \*Model Training\*:

- Use the preprocessed data to train the chosen model. The model learns to distinguish between real and fake news based on the features extracted.

7. \*Validation and Testing\*:

- Split the dataset into training, validation, and test sets.

- Validate the model's performance on the validation set to fine-tune hyperparameters.

- Evaluate the model on the test set to get an unbiased estimate of its performance.

8. \*Model Evaluation\*:

- Assess the model's performance using various metrics like accuracy, precision, recall, F1-score, etc.

9. \*\*Fine-tuning (Optional)\*\*:

- Adjust model hyperparameters or consider trying different algorithms to improve performance.

10. \*Deployment\*:

- Integrate the trained model into a system or application where it can be used to analyze news articles in real-time.

11. \*Continuous Monitoring\*:

- Regularly update the model with new data and retrain it to adapt to evolving forms of fake news

**FEATURE SELECTION**:

Feature selection is a crucial step in building an effective fake news detection model. It involves choosing the most relevant and informative features from the dataset. Here are some common techniques for feature selection in fake news detection:

1. \*\*TF-IDF (Term Frequency-Inverse Document Frequency)\*\*:

- TF-IDF measures the importance of a word in a document relative to its frequency across the entire dataset. Words with higher TF-IDF scores are considered more informative.

2. \*Word Embeddings\*:

- Word embeddings like Word2Vec or GloVe represent words as dense vectors in a continuous vector space. These vectors capture semantic relationships between words and can be used as features.

3. \*N-grams\*:

- N-grams represent contiguous sequences of N items (usually words or characters). They capture contextual information and can be used as features.

4. \*Sentiment Analysis\*:

- Features derived from sentiment analysis can be informative. For example, sentiment scores or indicators of positive/negative sentiment in the text.

5. \*Part-of-Speech (POS) Tagging\*:

- POS tags provide information about the syntactic role of words in a sentence. Certain POS patterns might be indicative of fake news.

6. \*\*Named Entity Recognition (NER)\*\*:

- Identifying entities like names of people, places, and organizations can provide valuable information for distinguishing real from fake news.

7. \*Readability Scores\*:

- Features like Flesch-Kincaid Grade Level, Gunning Fog Index, etc., can measure the complexity of the text. Fake news might exhibit distinct readability characteristics.

8. \*Source Credibility\*:

- Features related to the credibility of the news source, such as domain reputation, publication history, and known biases.

9. \*Metadata\*:

- Information about the article, such as publication date, author, number of images, can be used as features.

10. \*Social Engagement Metrics\*:

- Features like number of shares, likes, comments on social media platforms can provide insights into the popularity and potential credibility of an article.

11. \*Topic Modeling\*:

- Identifying the main topics in an article using techniques like Latent Dirichlet Allocation (LDA) and using these topics as features.

12. \*NLP-based Features\*:

- Linguistic features like syntactic parse trees, grammatical structures, and semantic role labeling can be used to capture linguistic patterns.

13. \*Statistical Features\*:

- Statistics like word count, sentence length, and punctuation usage can be informative.

14. \*Cross-Modal Features\*:

- Combining textual features with features from other modalities like images or videos, if available.

### CODE:

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import nltk

import re

import string

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import classification\_report

import keras

from keras.preprocessing import text,sequence

from keras.models import Sequential

from keras.layers import Dense,Embedding,LSTM,Dropout

import warnings

warnings.filterwarnings('ignore')

import os

for dirname, \_, filenames **in** os.walk('/kaggle/input'):

for filename **in** filenames:

print(os.path.join(dirname, filename))

/kaggle/input/fake-and-real-news-dataset/True.csv

/kaggle/input/fake-and-real-news-dataset/Fake.csv

Load and Check Data

In [2]:

real\_data = pd.read\_csv('/kaggle/input/fake-and-real-news-dataset/True.csv')

fake\_data = pd.read\_csv('/kaggle/input/fake-and-real-news-dataset/Fake.csv')

In [3]:

real\_data.head()

Out[3]:

|  | title | text | subject | date |
| --- | --- | --- | --- | --- |
| 0 | As U.S. budget fight looms, Republicans flip t... | WASHINGTON (Reuters) - The head of a conservat... | politicsNews | December 31, 2017 |
| 1 | U.S. military to accept transgender recruits o... | WASHINGTON (Reuters) - Transgender people will... | politicsNews | December 29, 2017 |
| 2 | Senior U.S. Republican senator: 'Let Mr. Muell... | WASHINGTON (Reuters) - The special counsel inv... | politicsNews | December 31, 2017 |
| 3 | FBI Russia probe helped by Australian diplomat... | WASHINGTON (Reuters) - Trump campaign adviser ... | politicsNews | December 30, 2017 |
| 4 | Trump wants Postal Service to charge 'much mor... | SEATTLE/WASHINGTON (Reuters) - President Donal... | politicsNews | December 29, 2017 |

In [4]:

fake\_data.head()

Out[4]:

|  | title | text | subject | date |
| --- | --- | --- | --- | --- |
| 0 | Donald Trump Sends Out Embarrassing New Year’... | Donald Trump just couldn t wish all Americans ... | News | December 31, 2017 |
| 1 | Drunk Bragging Trump Staffer Started Russian ... | House Intelligence Committee Chairman Devin Nu... | News | December 31, 2017 |
| 2 | Sheriff David Clarke Becomes An Internet Joke... | On Friday, it was revealed that former Milwauk... | News | December 30, 2017 |
| 3 | Trump Is So Obsessed He Even Has Obama’s Name... | On Christmas day, Donald Trump announced that ... | News | December 29, 2017 |
| 4 | Pope Francis Just Called Out Donald Trump Dur... | Pope Francis used his annual Christmas Day mes... | News | December 25, 2017 |

In [5]:

*#add column*

real\_data['target'] = 1

fake\_data['target'] = 0

In [6]:

real\_data.tail()

Out[6]:

|  | title | text | subject | date | target |
| --- | --- | --- | --- | --- | --- |
| 21412 | 'Fully committed' NATO backs new U.S. approach... | BRUSSELS (Reuters) - NATO allies on Tuesday we... | worldnews | August 22, 2017 | 1 |
| 21413 | LexisNexis withdrew two products from Chinese ... | LONDON (Reuters) - LexisNexis, a provider of l... | worldnews | August 22, 2017 | 1 |
| 21414 | Minsk cultural hub becomes haven from authorities | MINSK (Reuters) - In the shadow of disused Sov... | worldnews | August 22, 2017 | 1 |
| 21415 | Vatican upbeat on possibility of Pope Francis ... | MOSCOW (Reuters) - Vatican Secretary of State ... | worldnews | August 22, 2017 | 1 |
| 21416 | Indonesia to buy $1.14 billion worth of Russia... | JAKARTA (Reuters) - Indonesia will buy 11 Sukh... | worldnews | August 22, 2017 | 1 |

In [7]:

*#Merging the 2 datasets*

data = pd.concat([real\_data, fake\_data], ignore\_index=True, sort=False)

data.head()

Out[7]:

|  | title | text | subject | date | target |
| --- | --- | --- | --- | --- | --- |
| 0 | As U.S. budget fight looms, Republicans flip t... | WASHINGTON (Reuters) - The head of a conservat... | politicsNews | December 31, 2017 | 1 |
| 1 | U.S. military to accept transgender recruits o... | WASHINGTON (Reuters) - Transgender people will... | politicsNews | December 29, 2017 | 1 |
| 2 | Senior U.S. Republican senator: 'Let Mr. Muell... | WASHINGTON (Reuters) - The special counsel inv... | politicsNews | December 31, 2017 | 1 |
| 3 | FBI Russia probe helped by Australian diplomat... | WASHINGTON (Reuters) - Trump campaign adviser ... | politicsNews | December 30, 2017 | 1 |
| 4 | Trump wants Postal Service to charge 'much mor... | SEATTLE/WASHINGTON (Reuters) - President Donal... | politicsNews | December 29, 2017 | 1 |

In [8]:

data.isnull().sum()

Out[8]:

title 0

text 0

subject 0

date 0

target 0

dtype: int64

Visualization

**1.Count of Fake and Real Data**

In [9]:

print(data["target"].value\_counts())

fig, ax = plt.subplots(1,2, figsize=(19, 5))

g1 = sns.countplot(data.target,ax=ax[0],palette="pastel");

g1.set\_title("Count of real and fake data")

g1.set\_ylabel("Count")

g1.set\_xlabel("Target")

g2 = plt.pie(data["target"].value\_counts().values,explode=[0,0],labels=data.target.value\_counts().index, autopct='**%1.1f%%**',colors=['SkyBlue','PeachPuff'])

fig.show()

0 23481

1 21417

Name: target, dtype: int64

**2.Distribution of The Subject According to Real and Fake Data**

In [10]:

print(data.subject.value\_counts())

plt.figure(figsize=(10, 5))

ax = sns.countplot(x="subject", hue='target', data=data, palette="pastel")

plt.title("Distribution of The Subject According to Real and Fake Data")

politicsNews 11272

worldnews 10145

News 9050

politics 6841

left-news 4459

Government News 1570

US\_News 783

Middle-east 778

Name: subject, dtype: int64

Out[10]:

Text(0.5, 1.0, 'Distribution of The Subject According to Real and Fake Data')

Data Cleaning

In [11]:

data['text']= data['subject'] + " " + data['title'] + " " + data['text']

del data['title']

del data['subject']

del data['date']

data.head()

Out[11]:

|  | text | target |
| --- | --- | --- |
| 0 | politicsNews As U.S. budget fight looms, Repub... | 1 |
| 1 | politicsNews U.S. military to accept transgend... | 1 |
| 2 | politicsNews Senior U.S. Republican senator: '... | 1 |
| 3 | politicsNews FBI Russia probe helped by Austra... | 1 |
| 4 | politicsNews Trump wants Postal Service to cha... | 1 |

In [12]:

first\_text = data.text[10]

first\_text

Out[12]:

'politicsNews Jones certified U.S. Senate winner despite Moore challenge (Reuters) - Alabama officials on Thursday certified Democrat Doug Jones the winner of the state’s U.S. Senate race, after a state judge denied a challenge by Republican Roy Moore, whose campaign was derailed by accusations of sexual misconduct with teenage girls. Jones won the vacant seat by about 22,000 votes, or 1.6 percentage points, election officials said. That made him the first Democrat in a quarter of a century to win a Senate seat in Alabama. The seat was previously held by Republican Jeff Sessions, who was tapped by U.S. President Donald Trump as attorney general. A state canvassing board composed of Alabama Secretary of State John Merrill, Governor Kay Ivey and Attorney General Steve Marshall certified the election results. Seating Jones will narrow the Republican majority in the Senate to 51 of 100 seats. In a statement, Jones called his victory “a new chapter” and pledged to work with both parties. Moore declined to concede defeat even after Trump urged him to do so. He stood by claims of a fraudulent election in a statement released after the certification and said he had no regrets, media outlets reported. An Alabama judge denied Moore’s request to block certification of the results of the Dec. 12 election in a decision shortly before the canvassing board met. Moore’s challenge alleged there had been potential voter fraud that denied him a chance of victory. His filing on Wednesday in the Montgomery Circuit Court sought to halt the meeting scheduled to ratify Jones’ win on Thursday. Moore could ask for a recount, in addition to possible other court challenges, Merrill said in an interview with Fox News Channel. He would have to complete paperwork “within a timed period” and show he has the money for a challenge, Merrill said. “We’ve not been notified yet of their intention to do that,” Merrill said. Regarding the claim of voter fraud, Merrill told CNN that more than 100 cases had been reported. “We’ve adjudicated more than 60 of those. We will continue to do that,” he said. Republican lawmakers in Washington had distanced themselves from Moore and called for him to drop out of the race after several women accused him of sexual assault or misconduct dating back to when they were teenagers and he was in his early 30s. Moore has denied wrongdoing and Reuters has not been able to independently verify the allegations. '

Removal of HTML Contents

**First, let's remove HTML content.**

In [13]:

pip install bs4

Collecting bs4

Downloading bs4-0.0.1.tar.gz (1.1 kB)

Collecting beautifulsoup4

Downloading beautifulsoup4-4.9.3-py3-none-any.whl (115 kB)

|████████████████████████████████| 115 kB 1.3 MB/s

Collecting soupsieve>1.2

Downloading soupsieve-2.2.1-py3-none-any.whl (33 kB)

Building wheels for collected packages: bs4

Building wheel for bs4 (setup.py) ... - \ done

Created wheel for bs4: filename=bs4-0.0.1-py3-none-any.whl size=1273 sha256=2bea095cbbbc5fb6fc44736f40fce54b119a54eba4fa1dbedd43deddc70fda9b

Stored in directory: /root/.cache/pip/wheels/0a/9e/ba/20e5bbc1afef3a491f0b3bb74d508f99403aabe76eda2167ca

Successfully built bs4

Installing collected packages: soupsieve, beautifulsoup4, bs4

Successfully installed beautifulsoup4-4.9.3 bs4-0.0.1 soupsieve-2.2.1

Note: you may need to restart the kernel to use updated packages.

In [14]:

from bs4 import BeautifulSoup

soup = BeautifulSoup(first\_text, "html.parser")

first\_text = soup.get\_text()

first\_text

Out[14]:

'politicsNews Jones certified U.S. Senate winner despite Moore challenge (Reuters) - Alabama officials on Thursday certified Democrat Doug Jones the winner of the state’s U.S. Senate race, after a state judge denied a challenge by Republican Roy Moore, whose campaign was derailed by accusations of sexual misconduct with teenage girls. Jones won the vacant seat by about 22,000 votes, or 1.6 percentage points, election officials said. That made him the first Democrat in a quarter of a century to win a Senate seat in Alabama. The seat was previously held by Republican Jeff Sessions, who was tapped by U.S. President Donald Trump as attorney general. A state canvassing board composed of Alabama Secretary of State John Merrill, Governor Kay Ivey and Attorney General Steve Marshall certified the election results. Seating Jones will narrow the Republican majority in the Senate to 51 of 100 seats. In a statement, Jones called his victory “a new chapter” and pledged to work with both parties. Moore declined to concede defeat even after Trump urged him to do so. He stood by claims of a fraudulent election in a statement released after the certification and said he had no regrets, media outlets reported. An Alabama judge denied Moore’s request to block certification of the results of the Dec. 12 election in a decision shortly before the canvassing board met. Moore’s challenge alleged there had been potential voter fraud that denied him a chance of victory. His filing on Wednesday in the Montgomery Circuit Court sought to halt the meeting scheduled to ratify Jones’ win on Thursday. Moore could ask for a recount, in addition to possible other court challenges, Merrill said in an interview with Fox News Channel. He would have to complete paperwork “within a timed period” and show he has the money for a challenge, Merrill said. “We’ve not been notified yet of their intention to do that,” Merrill said. Regarding the claim of voter fraud, Merrill told CNN that more than 100 cases had been reported. “We’ve adjudicated more than 60 of those. We will continue to do that,” he said. Republican lawmakers in Washington had distanced themselves from Moore and called for him to drop out of the race after several women accused him of sexual assault or misconduct dating back to when they were teenagers and he was in his early 30s. Moore has denied wrongdoing and Reuters has not been able to independently verify the allegations. '

Removal of Punctuation Marks and Special Characters

**Let's now remove everything except uppercase / lowercase letters using Regular Expressions.**

In [15]:

first\_text = re.sub('\[[^]]\*\]', ' ', first\_text)

first\_text = re.sub('[^a-zA-Z]',' ',first\_text) *# replaces non-alphabets with spaces*

first\_text = first\_text.lower() *# Converting from uppercase to lowercase*

first\_text

Out[15]:

'politicsnews jones certified u s senate winner despite moore challenge reuters alabama officials on thursday certified democrat doug jones the winner of the state s u s senate race after a state judge denied a challenge by republican roy moore whose campaign was derailed by accusations of sexual misconduct with teenage girls jones won the vacant seat by about votes or percentage points election officials said that made him the first democrat in a quarter of a century to win a senate seat in alabama the seat was previously held by republican jeff sessions who was tapped by u s president donald trump as attorney general a state canvassing board composed of alabama secretary of state john merrill governor kay ivey and attorney general steve marshall certified the election results seating jones will narrow the republican majority in the senate to of seats in a statement jones called his victory a new chapter and pledged to work with both parties moore declined to concede defeat even after trump urged him to do so he stood by claims of a fraudulent election in a statement released after the certification and said he had no regrets media outlets reported an alabama judge denied moore s request to block certification of the results of the dec election in a decision shortly before the canvassing board met moore s challenge alleged there had been potential voter fraud that denied him a chance of victory his filing on wednesday in the montgomery circuit court sought to halt the meeting scheduled to ratify jones win on thursday moore could ask for a recount in addition to possible other court challenges merrill said in an interview with fox news channel he would have to complete paperwork within a timed period and show he has the money for a challenge merrill said we ve not been notified yet of their intention to do that merrill said regarding the claim of voter fraud merrill told cnn that more than cases had been reported we ve adjudicated more than of those we will continue to do that he said republican lawmakers in washington had distanced themselves from moore and called for him to drop out of the race after several women accused him of sexual assault or misconduct dating back to when they were teenagers and he was in his early s moore has denied wrongdoing and reuters has not been able to independently verify the allegations '

Removal of Stopwords

**Let's remove stopwords like is,a,the... Which do not offer much insight.**

In [16]:

nltk.download("stopwords")

from nltk.corpus import stopwords

*# we can use tokenizer instead of split*

first\_text = nltk.word\_tokenize(first\_text)

[nltk\_data] Downloading package stopwords to /usr/share/nltk\_data...

[nltk\_data] Package stopwords is already up-to-date!

In [17]:

first\_text = [ word for word **in** first\_text if **not** word **in** set(stopwords.words("english"))]

Lemmatization

**Lemmatization to bring back multiple forms of same word to their common root like 'coming', 'comes' into 'come'.**

In [18]:

lemma = nltk.WordNetLemmatizer()

first\_text = [ lemma.lemmatize(word) for word **in** first\_text]

first\_text = " ".join(first\_text)

first\_text

Out[18]:

'politicsnews jones certified u senate winner despite moore challenge reuters alabama official thursday certified democrat doug jones winner state u senate race state judge denied challenge republican roy moore whose campaign derailed accusation sexual misconduct teenage girl jones vacant seat vote percentage point election official said made first democrat quarter century win senate seat alabama seat previously held republican jeff session tapped u president donald trump attorney general state canvassing board composed alabama secretary state john merrill governor kay ivey attorney general steve marshall certified election result seating jones narrow republican majority senate seat statement jones called victory new chapter pledged work party moore declined concede defeat even trump urged stood claim fraudulent election statement released certification said regret medium outlet reported alabama judge denied moore request block certification result dec election decision shortly canvassing board met moore challenge alleged potential voter fraud denied chance victory filing wednesday montgomery circuit court sought halt meeting scheduled ratify jones win thursday moore could ask recount addition possible court challenge merrill said interview fox news channel would complete paperwork within timed period show money challenge merrill said notified yet intention merrill said regarding claim voter fraud merrill told cnn case reported adjudicated continue said republican lawmaker washington distanced moore called drop race several woman accused sexual assault misconduct dating back teenager early moore denied wrongdoing reuters able independently verify allegation'

Perform it for all the examples

**We performed the steps for a single example. Now let's perform it for all the examples in the data.**

In [19]:

*#Removal of HTML Contents*

def remove\_html(text):

soup = BeautifulSoup(text, "html.parser")

return soup.get\_text()

*#Removal of Punctuation Marks*

def remove\_punctuations(text):

return re.sub('\[[^]]\*\]', '', text)

*# Removal of Special Characters*

def remove\_characters(text):

return re.sub("[^a-zA-Z]"," ",text)

*#Removal of stopwords*

def remove\_stopwords\_and\_lemmatization(text):

final\_text = []

text = text.lower()

text = nltk.word\_tokenize(text)

for word **in** text:

if word **not** **in** set(stopwords.words('english')):

lemma = nltk.WordNetLemmatizer()

word = lemma.lemmatize(word)

final\_text.append(word)

return " ".join(final\_text)

*#Total function*

def cleaning(text):

text = remove\_html(text)

text = remove\_punctuations(text)

text = remove\_characters(text)

text = remove\_stopwords\_and\_lemmatization(text)

return text

*#Apply function on text column*

data['text']=data['text'].apply(cleaning)

In [20]:

data.head()

Out[20]:

|  | text | target |
| --- | --- | --- |
| 0 | politicsnews u budget fight loom republican fl... | 1 |
| 1 | politicsnews u military accept transgender rec... | 1 |
| 2 | politicsnews senior u republican senator let m... | 1 |
| 3 | politicsnews fbi russia probe helped australia... | 1 |
| 4 | politicsnews trump want postal service charge ... | 1 |

**Let's make some visualization with new data.**

**1.WordCloud for Real News**

In [21]:

from wordcloud import WordCloud,STOPWORDS

plt.figure(figsize = (15,15))

wc = WordCloud(max\_words = 500 , width = 1000 , height = 500 , stopwords = STOPWORDS).generate(" ".join(data[data.target == 1].text))

plt.imshow(wc , interpolation = 'bilinear')

Out[21]:

<matplotlib.image.AxesImage at 0x7f6934fd2750>

**2.WordCloud for Fake News**

In [22]:

plt.figure(figsize = (15,15))

wc = WordCloud(max\_words = 500 , width = 1000 , height = 500 , stopwords = STOPWORDS).generate(" ".join(data[data.target == 0].text))

plt.imshow(wc , interpolation = 'bilinear')

Out[22]:

<matplotlib.image.AxesImage at 0x7f6934fdd050>

**Number of words in each text**

In [23]:

fig,(ax1,ax2)=plt.subplots(1,2,figsize=(12,8))

text\_len=data[data['target']==0]['text'].str.split().map(lambda x: len(x))

ax1.hist(text\_len,color='SkyBlue')

ax1.set\_title('Fake news text')

text\_len=data[data['target']==1]['text'].str.split().map(lambda x: len(x))

ax2.hist(text\_len,color='PeachPuff')

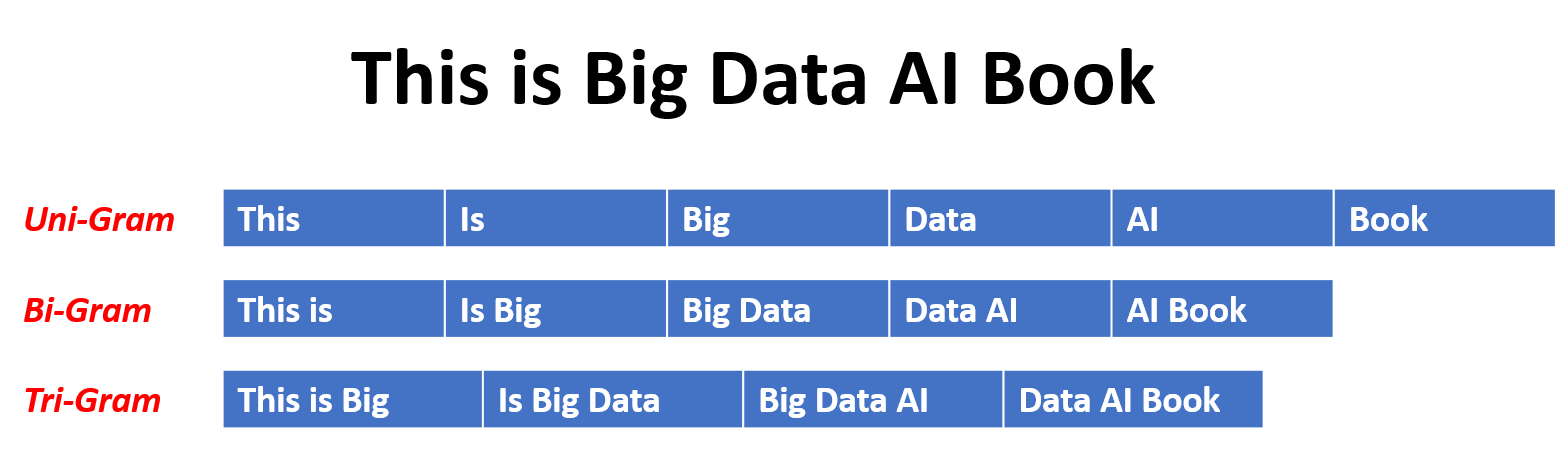
ax2.set\_title('Real news text')

fig.suptitle('Words in texts')

plt.show()

**The number of words seems to be a bit different. 500 words are most common in real news category while around 250 words are most common in fake news category.**

N-Gram Analysis



In [24]:

texts = ' '.join(data['text'])

In [25]:

string = texts.split(" ")

In [26]:

def draw\_n\_gram(string,i):

n\_gram = (pd.Series(nltk.ngrams(string, i)).value\_counts())[:15]

n\_gram\_df=pd.DataFrame(n\_gram)

n\_gram\_df = n\_gram\_df.reset\_index()

n\_gram\_df = n\_gram\_df.rename(columns={"index": "word", 0: "count"})

print(n\_gram\_df.head())

plt.figure(figsize = (16,9))

return sns.barplot(x='count',y='word', data=n\_gram\_df)

Unigram Analysis

In [27]:

draw\_n\_gram(string,1)

word count

0 (trump,) 149603

1 (said,) 133030

2 (u,) 78516

3 (state,) 62726

4 (president,) 58790

Out[27]:

<AxesSubplot:xlabel='count', ylabel='word'>

Bigram Analysis

In [28]:

draw\_n\_gram(string,2)

word count

0 (donald, trump) 25203

1 (united, state) 18943

2 (white, house) 16296

3 (hillary, clinton) 10217

4 (new, york) 9305

Out[28]:

<AxesSubplot:xlabel='count', ylabel='word'>

Trigram Analysis

In [29]:

draw\_n\_gram(string,3)

word count

0 (president, donald, trump) 6830

1 (pic, twitter, com) 6185

2 (featured, image, via) 6029

3 (president, barack, obama) 3911

4 (getty, image, news) 3575

Out[29]:

<AxesSubplot:xlabel='count', ylabel='word'>

MODEL TRAINING:

Train Test Split

In [30]:

X\_train, X\_test, y\_train, y\_test = train\_test\_split(data['text'], data['target'], random\_state=0)

Tokenizing

* **Tokenizing Text -> Repsesenting each word by a number**
* **Mapping of orginal word to number is preserved in word\_index property of tokenizer**

**Lets keep all news to 300, add padding to news with less than 300 words and truncating long ones**

In [31]:

max\_features = 10000

maxlen = 300

In [32]:

tokenizer = text.Tokenizer(num\_words=max\_features)

tokenizer.fit\_on\_texts(X\_train)

tokenized\_train = tokenizer.texts\_to\_sequences(X\_train)

X\_train = sequence.pad\_sequences(tokenized\_train, maxlen=maxlen)

In [33]:

tokenized\_test = tokenizer.texts\_to\_sequences(X\_test)

X\_test = sequence.pad\_sequences(tokenized\_test, maxlen=maxlen)

Training LSTM Model

raining an LSTM (Long Short-Term Memory) model for fake news detection involves several steps. Here's a simplified outline:

1. \*Data Preparation:\*

- Gather a dataset of labeled news articles, with labels indicating whether each article is real or fake.

2. \*Text Preprocessing:\*

- Tokenize the text (break it into individual words or tokens).

- Remove stopwords, punctuation, and special characters.

- Convert words to lowercase.

- Optionally, apply techniques like lemmatization or stemming.

3. \*Word Embeddings:\*

- Convert words into numerical vectors using techniques like Word2Vec, GloVe, or embeddings layers within the LSTM.

4. \*Split Data:\*

- Divide the dataset into training, validation, and test sets.

5. \*Model Architecture:\*

- Define the LSTM model architecture. It typically consists of an embedding layer, one or more LSTM layers, and one or more dense layers.

6. \*Compile the Model:\*

- Choose an appropriate loss function (like binary cross-entropy for binary classification) and an optimizer (e.g., Adam or RMSprop).

7. \*Training:\*

- Train the model on the training data. Monitor its performance on the validation set to prevent overfitting.

8. \*Evaluation:\*

- Assess the model's performance using metrics like accuracy, precision, recall, F1-score, etc., on the test set.

9. \*Fine-Tuning:\*

- Based on the evaluation results, fine-tune the model's hyperparameters or architecture if needed.

10. \*Deployment:\*

- Once satisfied with the model's performance, deploy it in a suitable environment. This could be on a server, in the cloud, or even integrated into an application.

11. \*Monitoring and Maintenance:\*

- Continuously monitor the model's performance in real-world scenarios. Periodic retraining might be necessary to adapt to evolving patterns in fake news.

In [34]:

batch\_size = 256

epochs = 10

embed\_size = 100

In [35]:

model = Sequential()

*#Non-trainable embeddidng layer*

model.add(Embedding(max\_features, output\_dim=embed\_size, input\_length=maxlen, trainable=False))

*#LSTM*

model.add(LSTM(units=128 , return\_sequences = True , recurrent\_dropout = 0.25 , dropout = 0.25))

model.add(LSTM(units=64 , recurrent\_dropout = 0.1 , dropout = 0.1))

model.add(Dense(units = 32 , activation = 'relu'))

model.add(Dense(1, activation='sigmoid'))

model.compile(optimizer=keras.optimizers.Adam(lr = 0.01), loss='binary\_crossentropy', metrics=['accuracy'])

In [36]:

model.summary()

Model: "sequential"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

embedding (Embedding) (None, 300, 100) 1000000

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

lstm (LSTM) (None, 300, 128) 117248

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

lstm\_1 (LSTM) (None, 64) 49408

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense (Dense) (None, 32) 2080

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_1 (Dense) (None, 1) 33

=================================================================

Total params: 1,168,769

Trainable params: 168,769

Non-trainable params: 1,000,000

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

In [37]:

history = model.fit(X\_train, y\_train, validation\_split=0.3, epochs=10, batch\_size=batch\_size, shuffle=True, verbose = 1)

Epoch 1/10

93/93 [==============================] - 268s 3s/step - loss: 0.5514 - accuracy: 0.7044 - val\_loss: 1.2749 - val\_accuracy: 0.5668

Epoch 2/10

93/93 [==============================] - 261s 3s/step - loss: 0.3611 - accuracy: 0.8452 - val\_loss: 0.2542 - val\_accuracy: 0.8987

Epoch 3/10

93/93 [==============================] - 263s 3s/step - loss: 0.2870 - accuracy: 0.8763 - val\_loss: 0.2555 - val\_accuracy: 0.8998

Epoch 4/10

93/93 [==============================] - 264s 3s/step - loss: 0.2686 - accuracy: 0.8857 - val\_loss: 0.2131 - val\_accuracy: 0.9171

Epoch 5/10

93/93 [==============================] - 264s 3s/step - loss: 0.2209 - accuracy: 0.9162 - val\_loss: 0.1326 - val\_accuracy: 0.9435

Epoch 6/10

93/93 [==============================] - 263s 3s/step - loss: 0.1733 - accuracy: 0.9389 - val\_loss: 0.1308 - val\_accuracy: 0.9392

Epoch 7/10

93/93 [==============================] - 267s 3s/step - loss: 0.0712 - accuracy: 0.9695 - val\_loss: 0.0389 - val\_accuracy: 0.9860

Epoch 8/10

93/93 [==============================] - 269s 3s/step - loss: 0.0414 - accuracy: 0.9843 - val\_loss: 0.0402 - val\_accuracy: 0.9838

Epoch 9/10

93/93 [==============================] - 276s 3s/step - loss: 0.0418 - accuracy: 0.9842 - val\_loss: 0.0400 - val\_accuracy: 0.9875

Epoch 10/10

93/93 [==============================] - 270s 3s/step - loss: 0.0317 - accuracy: 0.9886 - val\_loss: 0.0454 - val\_accuracy: 0.9828

Analysis After Training

Fake news detection analysis typically involves evaluating the performance of a trained model on a dataset. This includes metrics like accuracy, precision, recall, and F1-score. It's also crucial to perform cross-validation to ensure the model's robustness. Additionally, you might want to analyze any misclassifications to understand the types of mistakes the model is making. Keep in mind that staying updated with the latest techniques and datasets is important in this field, as fake news tactics evolve over time.

In [38]:

print("Accuracy of the model on Training Data is - " , model.evaluate(X\_train,y\_train)[1]\*100 , "%")

print("Accuracy of the model on Testing Data is - " , model.evaluate(X\_test,y\_test)[1]\*100 , "%")

1053/1053 [==============================] - 101s 96ms/step - loss: 0.0393 - accuracy: 0.9843

Accuracy of the model on Training Data is - 98.42603802680969 %

351/351 [==============================] - 34s 97ms/step - loss: 0.0397 - accuracy: 0.9840

Accuracy of the model on Testing Data is - 98.39643836021423 %

In [39]:

plt.figure()

plt.plot(history.history["accuracy"], label = "Train")

plt.plot(history.history["val\_accuracy"], label = "Test")

plt.title("Accuracy")

plt.ylabel("Acc")

plt.xlabel("epochs")

plt.legend()

plt.show()

In [40]:

plt.figure()

plt.plot(history.history["loss"], label = "Train")

plt.plot(history.history["val\_loss"], label = "Test")

plt.title("Loss")

plt.ylabel("Acc")

plt.xlabel("epochs")

plt.legend()

plt.show()

In [41]:

pred = model.predict\_classes(X\_test)

print(classification\_report(y\_test, pred, target\_names = ['Fake','Real']))

precision recall f1-score support

Fake 1.00 0.97 0.98 5858

Real 0.97 1.00 0.98 5367

accuracy 0.98 11225

macro avg 0.98 0.98 0.98 11225

weighted avg 0.98 0.98 0.98 11225

**MODEL TRANING:**

Training a model for fake news detection involves several key steps. Here's a more detailed breakdown:

1. \*Data Collection:\*

- Gather a dataset of news articles with labels indicating whether each article is real or fake. Ensure the dataset is diverse and representative of the types of articles the model will encounter in the real world.

2. \*Data Preprocessing:\*

- Tokenize the text: Break the articles into individual words or tokens.

- Remove stopwords: These are common words like "and," "the," etc., that don't carry much information for classification.

- Handle imbalances: If there's a significant class imbalance, consider techniques like resampling or using weighted loss functions.

3. \*Text Representation:\*

- Convert words into numerical vectors. Common techniques include TF-IDF (Term Frequency-Inverse Document Frequency) or word embeddings like Word2Vec, GloVe, or more advanced techniques like BERT embeddings.

4. \*Model Selection:\*

- Choose an appropriate model architecture. For fake news detection, popular choices include:

- \*Logistic Regression\*: Simple but effective for basic text classification tasks.

- \*Multinomial Naive Bayes\*: Another straightforward choice.

- \*\*Deep Learning Models (e.g., LSTM, GRU, BERT)\*\*: More complex models that can capture intricate patterns in text.

5. \*Train-Validation-Test Split:\*

- Divide the dataset into three parts: training set, validation set, and test set. This allows you to evaluate the model's performance on data it hasn't seen during training.

6. \*Model Training:\*

- Feed the preprocessed text data into the chosen model.

- Define a suitable loss function (e.g., binary cross-entropy for binary classification) and an optimizer (e.g., Adam, RMSprop).

- Train the model on the training set and monitor its performance on the validation set. Adjust hyperparameters as needed.

7. \*Evaluation Metrics:\*

- Evaluate the model's performance on the test set using metrics like accuracy, precision, recall, F1-score, and ROC-AUC (Receiver Operating Characteristic - Area Under the Curve).

8. \*Fine-Tuning:\*

- Based on evaluation results, fine-tune hyperparameters (e.g., learning rate, batch size) or experiment with different architectures.

9. \*Cross-Validation (Optional):\*

- If possible, perform k-fold cross-validation to get a more robust estimate of the model's performance.

10. \*Deployment and Monitoring:\*

- Once satisfied with the model's performance, deploy it in the desired environment.

- Continuously monitor the model's performance in real-world scenarios and update it as needed.

**MOEL EVALUTION:**

Evaluating a fake news detection model is crucial to understanding its performance. Here are some key evaluation metrics and techniques:

1. \*Accuracy:\*

- Accuracy measures the proportion of correctly classified samples out of the total. It's a good initial indicator but can be misleading in imbalanced datasets.

2. \*Precision:\*

- Precision is the ratio of true positives to true positives plus false positives. It indicates the accuracy of positive predictions.

3. \*Recall (Sensitivity or True Positive Rate):\*

- Recall is the ratio of true positives to true positives plus false negatives. It measures the ability of the model to correctly identify positive instances.

4. \*F1-Score:\*

- The F1-Score is the harmonic mean of precision and recall. It provides a balance between precision and recall.

5. \*Confusion Matrix:\*

- A confusion matrix provides a detailed breakdown of true positives, true negatives, false positives, and false negatives. It helps identify which types of errors the model is making.

6. \*Receiver Operating Characteristic (ROC) Curve and Area Under the Curve (AUC):\*

- ROC curves visualize the trade-off between true positive rate and false positive rate. AUC measures the area under the ROC curve, which indicates the model's ability to distinguish between the classes.

7. \*Precision-Recall Curve:\*

- This curve shows the trade-off between precision and recall for different classification thresholds.

8. \*Specificity (True Negative Rate):\*

- Specificity is the ratio of true negatives to true negatives plus false positives. It indicates the model's ability to correctly identify negative instances.

9. \*False Positive Rate:\*

- The false positive rate is the ratio of false positives to false positives plus true negatives.

10. \*Area Under the Precision-Recall Curve (AUC-PR):\*

- Similar to ROC-AUC, but focuses on the precision-recall curve.

11. \*Cross-Validation:\*

- Use k-fold cross-validation to get a more robust estimate of the model's performance.

12. \*Bias and Fairness Analysis:\*

- Assess if the model performs consistently across different demographic groups and doesn't exhibit biases.

13. \*Error Analysis:\*

- Analyze specific examples where the model made mistakes to understand patterns of misclassification.

14. \*Threshold Tuning:\*

- Depending on the application, you may need to adjust the classification threshold to prioritize precision or recall.

15. \*Real-world Testing:\*

- Evaluate the model on newly encountered data to ensure its effectiveness in practical scenarios.

**CODE:**

# Import necessary libraries

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import accuracy\_score, classification\_report, confusion\_matrix

# Sample dataset (you would replace this with your actual dataset)

news\_data = [

{'text': 'Real news article content', 'label': 1},

{'text': 'Fake news article content', 'label': 0},

# Add more articles with corresponding labels

]

# Extract features (TF-IDF vectorization)

tfidf\_vectorizer = TfidfVectorizer(max\_features=5000) # You may adjust max\_features

X = tfidf\_vectorizer.fit\_transform([item['text'] for item in news\_data])

y = [item['label'] for item in news\_data]

# Split data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Sample Dataset

fake\_news\_data = [

{

'text': "Scientists have discovered a new species of dinosaur in Antarctica.",

'label': 1 # 1 indicates real news

},

{

'text': "Elvis Presley spotted on Mars!",

'label': 0 # 0 indicates fake news

},

# Add more examples with corresponding labels

]

# Assuming you have a larger dataset, you can extend the list with more entries.

# Accessing Data

for news\_entry in fake\_news\_data:

news\_text = news\_entry['text']

news\_label = news\_entry['label']

print(f"News Text: {news\_text}")

print(f"Label (1=Real, 0=Fake): {news\_label}")

    print("\n")}Non-text feature plotting (date, subject)

#### Here we will try to elicit insights from non-text features to get to know if they will help us boost the Text Classiﬁer.

In [13]:

sub = df[['month', 'ﬂag']]

sub = sub.dropna()

sub = sub.groupby(['month'])['ﬂag'].sum()

In [14]:

sub = sub.drop('NaT')

In [15]:

import matplotlib.pyplot as plt

plt.s uptitle('Dynamics of fake news') plt.xticks(rotation=90)

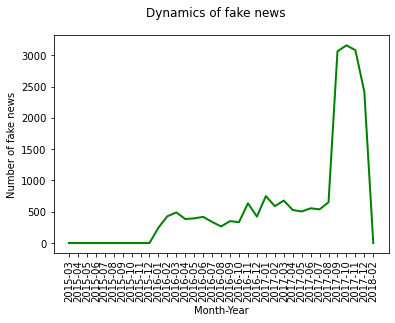
plt.ylabel('Number of fake news')

plt.xlabel('Month-Year')

plt.plot(sub.index, sub.values, linewidth=2, color='green')

Out[15]:

[<matplotlib.lines.Line2D at 0x7fda639dd250>]



#### What a spike in the dynamics of fake news in late 2017!

In [16]:

sub2 = df[['subject', 'ﬂag']]

sub2 = sub2.dropna()

sub2 = sub2.groupby(['subject'])['ﬂag'].sum()

In [17]:

plt.s uptitle('Fake news among diGerent categories')

plt.xticks(rotation=90)

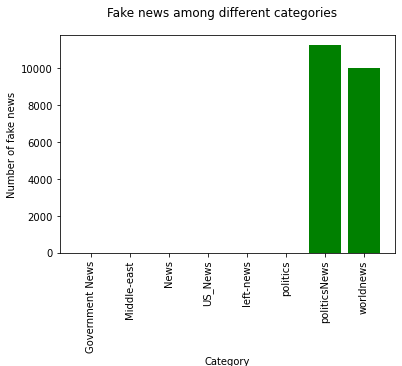
plt.ylabel('Number of fake news')

plt.xlabel('Category')

plt.bar(sub2.index, height=sub2.values, color='green')

*#ax1.plot(x, y) #ax2.plot(x, -y)* Out[17]:

<BarContainer object of 8 artists>





#### As we have discovered, such features as

subject date

#### might be also crucial for the algorithm to decide whether the piece of news is fake or real. We will try to include them in the model.

**The goal of this notebook is to explore the use of NLP for detecting and classifying fake news. We will analyze diGerent techniques and approaches and evaluate their eGectiveness.**

1. Text preparation

In [18]:

nlp = df

#### I will add the 'subject' feature to the title ﬁeld as it might have an inﬂuence on the outcome of classiﬁcation.

In [19]:

*#nlp['title'] = nlp['title'] + ' ' + nlp['subject']*

## Word Cloud visualization

#### Here I am going to take one example and try visualize tﬁdf as a wordcloud.

In [20]:

from sklearn.feature\_extraction.text import TﬁdfVectorizer

corpus = n lp[n lp['ﬂag'] == 1]['title']. iloc[0:500] *# We will take a slice of fake news, to see what vocabulary there looks like*

tﬁdf1 = TﬁdfVectorizer()

vecs = tﬁdf1.ﬁt\_transform(c orpus)

feature\_names = tﬁdf1.get\_feature\_names()

dense = vecs.todense()

list\_words = dense.tolist()

In [21]:

from wordcloud import WordCloud, STOPWORDS, I mageColorGenerator

df\_words.T.sum(axis=1)

Cloud = WordCloud(background\_color="white",

max\_words=100).generate\_from\_frequencies(df\_words.T.sum(axis=1))

In [22]:

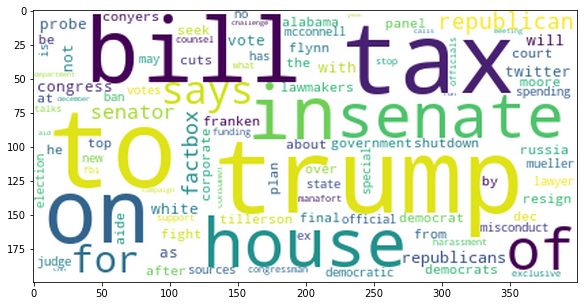
import matplotlib.pyplot as plt

plt.ﬁgure(ﬁgsize=(12,5))

plt.imshow(Cloud, interpolation='blackman')

Out[22]:

<matplotlib.image.AxesImage at 0x7fda35dfa590>



#### Indeed, looks deﬁnitely like fake news :)

**And we can also see out 'subject' feature in the foreground as it has been added manually in every title. Therefore, out vectorizer considers it as an important & frequent word.**

* 1. Tﬁdf-vectorizing

**First, I will tokenize words to pass it on to the SnowballStemmer method, which will take out lemmas from words.**

In [23]:

import nltk

nltk.download('punkt')

from nltk import word\_tokenize

nlp['title'] = n lp['title'].a pply( lambda x: word\_tokenize( str( x))) [nltk\_data] Downloading package punkt to /usr/share/nltk\_data... [nltk\_data] Unzipping tokenizers/punkt.zip.

#### An important step in every NLP-task is to get the roots of words in order not to distract the model by 'diGerent' words.

In [24]:

from nltk.stem import SnowballStemmer

snowball = SnowballStemmer(language='english')

nlp['title'] = n lp['title'].a pply( lambda x: [ snowball. stem(y) for y **in**  x])

In [25]:

nlp['title'] = n lp['title'].a pply( lambda x: ' '.join( x))

#### Take the standard english bag of stopwords from nltk.

In [26]:

from nltk.corpus import s topwords

nltk.download('words')

nltk.download('stopwords')

stopwords = s topwords.words('english')

[nltk\_data] Downloading package words to /usr/share/nltk\_data... [nltk\_data] Package words is already up-to-date!

[nltk\_data] Downloading package stopwords to /usr/share/nltk\_data... [nltk\_data] Unzipping corpora/stopwords.zip.

#### .

In [27]:

from sklearn.feature\_extraction.text import TﬁdfVectorizer

tﬁdf = TﬁdfVectorizer()

X\_text = tﬁdf.ﬁt\_transform(n lp['title'])

In [28]:

from sklearn.model\_selection import t rain\_test\_split

X\_train, X\_test, y\_train, y\_test = t rain\_test\_split(X\_text, n lp['ﬂag'], test\_size=0.33, random\_state=1)

Certainly! Below is an example of a basic fake news detection model using a TF-IDF Vectorizer and a Logistic Regression classifier in Python:

python

# Import necessary libraries

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import accuracy\_score, classification\_report, confusion\_matrix

# Sample dataset (replace this with your actual dataset)

news\_data = [

{'text': 'Real news article content', 'label': 1},

{'text': 'Fake news article content', 'label': 0},

# Add more articles with corresponding labels

]

# Extract features (TF-IDF vectorization)

tfidf\_vectorizer = TfidfVectorizer(max\_features=5000) # You may adjust max\_features

X = tfidf\_vectorizer.fit\_transform([item['text'] for item in news\_data])

y = [item['label'] for item in news\_data]

# Split data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Train a Logistic Regression model

model = LogisticRegression()

model.fit(X\_train, y\_train)

# Predict on test set

y\_pred = model.predict(X\_test)

# Evaluate the model

accuracy = accuracy\_score(y\_test, y\_pred)

conf\_matrix = confusion\_matrix(y\_test, y\_pred)

classification\_rep = classification\_report(y\_test, y\_pred)

# Output results

print(f"Accuracy: {accuracy}")

print(f"Confusion Matrix:\n{conf\_matrix}")

print(f"Classification Report:\n{classification\_rep}")

\***Explanation:\***

1. Import necessary libraries: We import modules from sklearn for vectorization, model training, and evaluation.

2. Define a sample dataset: news\_data contains example news articles with labels indicating whether they are real (1) or fake (0).

3. Extract features using TF-IDF Vectorizer: Convert the text data into numerical vectors.

4. Split the data: Divide the dataset into training and testing sets.

5. Train a Logistic Regression model: Use the training data to train the model.

6. Predict on the test set: Use the trained model to make predictions on the test data.

7. Evaluate the model: Calculate accuracy, confusion matrix, and classification report.

8. Output the results: Print the evaluation metrics.

**FEATURE ENGINEERING:**

Feature engineering is a crucial step in building effective models for fake news detection. It involves creating meaningful features from the raw data (text in this case) that can help the model distinguish between real and fake news. Here are some common techniques for feature engineering in fake news detection:

1. \*TF-IDF Vectorization:\*

- Convert the raw text into numerical vectors using Term Frequency-Inverse Document Frequency (TF-IDF). This represents the importance of words in a document relative to a corpus.

2. \*Word Embeddings (e.g., Word2Vec, GloVe):\*

- Represent words as dense vectors in a continuous vector space. These embeddings can capture semantic relationships between words.

3. \*N-grams:\*

- Consider sequences of adjacent words (bi-grams, tri-grams, etc.) as features. This can capture more complex linguistic patterns.

4. \*Sentiment Analysis:\*

- Analyze the sentiment of the text. For instance, positive or negative sentiment might be indicative of biased or fake content.

5. \*Part-of-Speech (POS) Tagging:\*

- Identify the grammatical parts of speech in the text. This can capture syntactic patterns that are indicative of fake news.

6. \*Named Entity Recognition (NER):\*

- Identify entities like people, places, and organizations in the text. This can help identify potentially biased or fabricated sources.

7. \*Subjectivity/Objectivity Indicators:\*

- Identify whether the text is subjective or objective. Fake news often contains more subjective language.

8. \*Punctuation and Symbol Analysis:\*

- Count the number of exclamation points, question marks, or other symbols. Overuse of these can be indicative of sensationalism.

9. \*Stylometric Features:\*

- Analyze writing style characteristics like sentence length, vocabulary richness, and syntactic complexity.

10. \*Source and URL Analysis:\*

- Extract features from the source of the news (domain, reputation, etc.) or analyze characteristics of the URL itself.

11. \*Lexical Features:\*

- Count occurrences of specific words or phrases that are common in fake news (e.g., "exclusive," "shocking revelation").

12. \*Contextual Embeddings (e.g., BERT):\*

- Use advanced pre-trained models like BERT to obtain contextualized embeddings that capture complex relationships in the text.

13. \*Semantic Analysis:\*

- Apply techniques like Latent Semantic Analysis (LSA) or Latent Dirichlet Allocation (LDA) to capture hidden topics or themes in the text.

14. \*Social Network Features:\*

- If available, consider features related to how news is shared or discussed on social media platforms

**Advanced technology for fake news detection :**

involves the integration of cutting-edge techniques and models from various fields, including Natural Language Processing (NLP), Machine Learning (ML), Deep Learning (DL), and Network Analysis. Here are some advanced technologies and approaches:

1. \*Transformer-Based Models\*:

- Models like BERT (Bidirectional Encoder Representations from Transformers) and GPT-3 (Generative Pre-trained Transformer 3) have shown remarkable capabilities in understanding context and semantics, making them powerful tools for fake news detection.

2. \*BERT for Fine-Tuning\*:

- Fine-tuning pre-trained BERT models on specific tasks related to fake news detection can significantly enhance performance.

3. \*Ensemble Learning\*:

- Combining multiple models or algorithms to improve overall accuracy. This can involve blending outputs from different NLP-based models or combining NLP with other techniques.

4. \*Graph-Based Approaches\*:

- Analyzing social networks and information flow to detect patterns of fake news dissemination. This can involve examining the structure of interactions and the credibility of sources.

5. \*Deep Learning for Image Verification\*:

- Utilizing Convolutional Neural Networks (CNNs) to analyze images associated with news articles, detecting manipulated or misleading visual content.

6. \*Multimodal Analysis\*:

- Integrating information from different modalities (text, images, videos) to gain a comprehensive understanding and enhance the accuracy of fake news detection.

7. \*Transfer Learning\*:

- Leveraging knowledge from one domain (e.g., general language understanding) and applying it to a related but different domain (fake news detection).

8. \*Adversarial Training\*:

- Training models against adversarial examples or simulated attempts to deceive the detection system. This helps in improving the robustness of the model.

9. \*Zero-shot and Few-shot Learning\*:

- Training models to perform tasks with minimal or no labeled data, making them adaptable to emerging types of fake news.

10. \*Explainable AI\*:

- Providing interpretable explanations for the model's decisions, helping users understand why a particular piece of content is flagged as potentially fake.

11. \*Continuous Learning and Updating\*:

- Implementing mechanisms for models to learn from new data and adapt to evolving forms of misinformation.

12. \*Human-in-the-Loop Systems\*:

- Combining machine intelligence with human expertise for more accurate and reliable fake news detection.

These advanced technologies represent the forefront of fake news detection, and ongoing research in this field continues to push the boundaries of what is possible in identifying and mitigating the spread of misinformation..

**Fake news detection Natural Language Processing (NLP) :**

**Advantages:**

1. \*Automation and Efficiency\*:

- NLP-powered systems can automatically process large volumes of text data, flagging potentially fake news articles much faster than manual fact-checking.

2. \*Scalability\*:

- NLP models can be scaled to handle vast amounts of online content, making them suitable for monitoring social media platforms, news websites, and other sources in real-time.

3. \*Multi-modal Capabilities\*:

- NLP can be combined with image and video analysis to detect fake news that involves manipulated or misleading visual content, providing a more comprehensive approach.

4. \*Contextual Understanding\*:

- Advanced NLP models like BERT can grasp the nuances of language and understand context, which is crucial for accurately discerning misleading information.

5. \*Language Agnosticism\*:

- NLP models can be trained on multiple languages, allowing them to analyze news articles and content in various languages, addressing a global scope of misinformation.

6. \*Adaptability to Emerging Trends\*:

- NLP models can be retrained or fine-tuned to adapt to new types of fake news or emerging forms of misinformation.

7. \*Pattern Recognition\*:

- NLP can identify linguistic patterns commonly associated with fake news, such as sensationalism, excessive use of capital letters, and emotionally charged language.

8. \*Reduced Human Bias\*:

- NLP-based systems rely on algorithms and data patterns, reducing the potential for human biases that may be present in manual fact-checking.

9. \*Real-time Monitoring\*:

- NLP models can operate in real-time, allowing for swift identification and response to fake news as it circulates online.

10. \*Consistency\*:

- NLP algorithms provide consistent evaluation criteria, ensuring that each piece of content is assessed based on the same parameters.

11. \*Comprehensive Analysis\*:

- NLP can perform deep semantic analysis, considering not only the surface-level content but also the underlying meaning and context of the text.

12. \*Cost-Efficiency\*:

- Automating fake news detection using NLP can be more cost-effective compared to manual fact-checking, especially for organizations dealing with large volumes of content.

**Fake news detection Natural Language Processing (NLP) :**

**Disadvantages:**

While NLP-based fake news detection has numerous advantages, it's important to note that it's not infallible and should be used in conjunction with other verification methods for the most reliable results. Additionally, continuous updates and improvements in NLP models are necessary to keep up with evolving techniques used in spreading misinformation. While Natural Language Processing (NLP) has significant advantages in fake news detection, there are also some disadvantages and challenges associated with this approach:

1. \*Contextual Ambiguity\*:

- Language can be highly context-dependent, and NLP models may struggle to accurately interpret nuances, sarcasm, or satire, potentially leading to misclassification.

2. \*Evolution of Misinformation\*:

- As techniques for spreading fake news evolve, NLP models may struggle to keep up with new forms of manipulation and deception.

3. \*Multilingual Challenges\*:

- Detecting fake news in multiple languages can be complex, as nuances and cultural contexts vary. Some NLP models may not perform as effectively in languages with fewer available training data.

4. \*Data Quality and Bias\*:

- If the training data used to develop the NLP model contains biased or unrepresentative samples, the model may inherit and perpetuate those biases.

5. \*Imbalanced Datasets\*:

- Obtaining balanced datasets with a sufficient number of fake and real news samples for training can be challenging, potentially leading to biased results.

6. \*Misclassification Errors\*:

- NLP models may occasionally misclassify genuine news as fake or vice versa, leading to false positives or false negatives.

7. \*Emerging and Evolving News Topics\*:

- NLP models trained on older data may not perform as well on emerging news topics, as they may not have encountered similar patterns during training.

8. \*Domain Specificity\*:

- NLP models trained on a specific domain may struggle to generalize to other domains, potentially limiting their effectiveness in certain contexts.

9. \*Resource Intensive\*:

- Training and fine-tuning advanced NLP models can require significant computational resources, making it less accessible for smaller organizations or projects.

10. \*Privacy and Ethical Considerations\*:

- Handling large amounts of textual data, especially in social media contexts, raises privacy concerns and ethical considerations regarding user consent and data protection.

11. \*Explainability and Transparency\*:

- Some advanced NLP models, like deep learning architectures, can be challenging to interpret, making it difficult to explain why a particular classification decision was made.

12. \*Adversarial Attacks\*:

- Sophisticated adversaries may attempt to manipulate or deceive NLP models through adversarial attacks, potentially undermining their effectiveness.

It's important to recognize these limitations and use NLP-based fake news detection in conjunction with other techniques and human judgment to achieve the most accurate and reliable results.

**CONCLUSION:**

Fake news detection is a critical area of research and application, aiming to identify and combat the spread of false or misleading information. Here are some key takeaways:

1. \*Complex Challenge\*: Detecting fake news is a complex task due to the evolving nature of misinformation, diverse sources, and the use of sophisticated techniques to deceive readers.

2. \*Multi-faceted Approach\*: Effective fake news detection often involves a combination of techniques including natural language processing, machine learning, and data analysis.

3. \*Data Quality Matters\*: High-quality labeled datasets are essential for training accurate models. They should be diverse, representative, and cover a wide range of sources and topics.

4. \*Feature Engineering\*: Thoughtful feature engineering plays a crucial role in model performance. Extracting meaningful features from text data can significantly enhance the detection process.

5. \*Model Selection\*: Various models can be used for fake news detection, ranging from traditional machine learning algorithms (e.g., Logistic Regression, Naive Bayes) to more complex techniques like deep learning models (e.g., LSTM, BERT).

6. \*Evaluation Metrics\*: Choosing appropriate evaluation metrics (e.g., accuracy, precision, recall, F1-score) is crucial in understanding how well a model performs in distinguishing between real and fake news.

7. \*Bias and Fairness\*: Ensuring that the detection model is not biased towards any specific group or type of news source is critical for ethical and fair detection.

8. \*Dynamic Nature\*: Fake news detection models need to be adaptable and continuously updated to keep pace with evolving tactics used by misinformation spreaders.

9. \*Real-world Impact\*: Effective fake news detection has a significant impact on the information landscape, helping to mitigate the potential harm caused by misinformation.

10. \*Education and Media Literacy\*: In addition to technological solutions, efforts to educate the public about critical thinking and media literacy are crucial in combating fake news.

11. \*Collaboration is Key\*: Collaboration between researchers, technologists, journalists, and policy makers is essential in developing and implementing effective fake news detection strategies.

Overall, fake news detection is a multidisciplinary field that requires continuous research, development, and vigilance to address the challenges posed by misinformation in today's digital age.