FAKE NEWS DETECTION USING

NLP

**TEAM MEMBERS 1.611721104054:M.Mouriya**

**2.611721104058:D.Papetha**

**3.611721104073:S.Sabithrija 4.611721104701:S.Arthi**

**5.611721104702:M.P.Preethi**

### Phase 3 Submission Document

**PROJECT:FAKE NEWS DETECTION**



**Introduction:**

In today's digital age, the proliferation of fake news has become a pervasive and concerning issue. Misinformation, disinformation, and fake news stories can

spread like wildﬁre through various online platforms, inﬂuencing public opinion, and sometimes even cing real-world consequences.

To combat this challenge, the application of Natural Language Processing (NLP) techniques has emerged as a powerful tool for fake news detection.

NLP, a subﬁeld of artiﬁcial intelligence, focuses on the interaction between computers and human language. By harnessing the power of NLP, we can develop sophisticated algorithms and models that can analyze and scrutinize textual content to identify false or misleading information.

This research explores the pivotal role of NLP in fake news detection, its methodologies, and the potential beneﬁts it oGers in addressing the ever-growing problem of fake news in our digital society.

**Detecting fake news using Natural Language Processing (NLP) involves several steps, including loading and processing data. Here's a high-level overview of the process:**

### \*\*Data Collection\*\*:

* + Gather a dataset containing both real and fake news articles. You can ﬁnd datasets like LIAR-PLUS, FakeNewsNet, or create your own.

1. **\*\*Data Preprocessing\*\*:**
   * Remove irrelevant information like HTML tags, special characters, and punctuation.
   * Tokenize the text into words or subword units using techniques like word tokenization or subword tokenization (e.g., using spaCy or tokenizers like BERT's WordPiece).
   * Perform text normalization, which includes lowercasing, stemming, or lemmatization.
2. **\*\*Feature Extraction\*\*:**
   * Convert the text data into numerical vectors. Common methods include TF- IDF (Term Frequency-Inverse Document Frequency) or word embeddings like Word2Vec, GloVe, or BERT embeddings.
3. **\*\*Data Splitting\*\*:**

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

1. **\*\*Model Building\*\*:**
   * Choose an NLP model for fake news detection. Popular choices include LSTM, CNN, or transformer-based models like BERT or GPT.
2. **\*\*Training\*\*:**
   * Train the chosen model on the training data. You may use techniques like transfer learning to ﬁne-tune pre-trained models.
3. **\*\*Validation\*\*:**
   * Monitor the model's performance on the validation dataset to prevent overﬁtting.
4. **\*\*Testing and Evaluation\*\*:**
   * Evaluate the model's performance on the test dataset using metrics like accuracy, precision, recall, F1-score, and ROC AUC.
5. **\*\*Deployment\*\*:**
   * Once satisﬁed with the model's performance, deploy it in a real-world environment, such as a website or an application.
6. **\*\*Monitoring\*\*:**
   * Continuously monitor the model's performance, as the fake news landscape evolves over time. You might need to retrain the model periodically.

**FAKE NEWS NLP WITH MODELS:0**

1. Introduction

#### In this notebook, we will explore how Natural Language Processing (NLP) can be used to detect and

**classify fake news. We will look at diGerent techniques and approaches that can be used to identify and classify fake news and analyze their eGectiveness.**

1. Use cases

**Fake news is becoming increasingly prevalent, and it can have serious implications. NLP can be used to**

**detect and classify fake news in order to protect society from its damaging eGects. Additionally, NLP can be used to identify topics and trends in news articles, which can be beneﬁcial for research and news**

**outlets.**

In [1]:

import numpy as n p *# linear algebra*

import pandas as pd *# data processing, CSV ﬁle I/O (e.g. pd.read\_csv)*

*# Input data ﬁles are available in the read-only "../input/" directory*

*# For example, running this (by clicking run or pressing Shift+Enter) will list all ﬁles under the input directory*

import os

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

print(os.path.join(dirname, ﬁlename))

# Loading data

In [2]:

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

fake['ﬂag'] = 0

fake

Out[2]:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 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. ... |  |  |  |
|  |  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 23478 | Sunnistan:  US and | Patrick  Henningse | Middle-  east | January  15, 2016 | 0 |
|  | Allied  ‘Safe | n 21st  Century |  |  |  |
|  | Zone’ Plan  to T... | WireReme  mber ... |  |  |  |
| 23479 | How to Blow $700 | 21st Century | Middle- east | January 14, 2016 | 0 |
|  | Million: Al  Jazeera | Wire says  Al Jazeera |  |  |  |
|  | America F... | America will... |  |  |  |
| 23480 | 10 U.S.  Navy | 21st Century | Middle- east | January 12, 2016 | 0 |
|  | Sailors  Held by | Wire says  As 21WIRE |  |  |  |
|  | Iranian Military ... | predicted in ... |  |  |  |

23481 rows × 5 columns

In [3]:

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

true['ﬂag'] = 1

true

Out[3]:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | title | text | subject | date | ﬂag |
| 0 | As U.S.  budget ﬁght looms, Republica ns ﬂip t... | WASHING TON  (Reuters) - The head of a conservat.  .. | politicsNe ws | December 31, 2017 | 1 |
| 1 | U.S.  military to | WASHING TON | politicsNe ws | December 29, 2017 | 1 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | accept transgend er recruits  o... | (Reuters) - Transgend er people  will... |  |  |  |
| 2 | Senior U.S. Republica n senator: 'Let Mr.  Muell... | WASHING TON  (Reuters) - The special counsel inv... | politicsNe ws | December 31, 2017 | 1 |
| 3 | FBI Russia probe helped by Australian diplomat... | WASHING TON  (Reuters) - Trump campaign adviser ... | politicsNe ws | December 30, 2017 | 1 |
| 4 | Trump wants Postal Service to charge 'much mor... | SEATTLE/ WASHING TON  (Reuters) - President Donal... | politicsNe ws | December 29, 2017 | 1 |
| ... | ... | ... | ... | ... | ... |
| 21412 | 'Fully committe d' NATO  backs new U.S.  approach..  . | BRUSSELS  (Reuters) - NATO  allies on Tuesday we... | worldnews | August 22,  2017 | 1 |
| 21413 | LexisNexis withdrew two products | LONDON  (Reuters) - LexisNexis, | worldnews | August 22,  2017 | 1 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | from | a provider |  |  |  |
| 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 |

21417 rows × 5 columns

In [4]:

df = pd.DataFrame()

df = true.append(fake)

# EDA + Data cleaning

#### Let's check the datatypes.

In [5]:

df.info()

<class 'pandas.core.frame.DataFrame'> Int64Index: 44898 entries, 0 to 23480 Data columns (total 5 columns):

# Column Non-Null Count Dtype

1. title 44898 non-null object
2. text 44898 non-null object
3. subject 44898 non-null object
4. date 44898 non-null object
5. ﬂag 44898 non-null int64 dtypes: int64(1), object(4)

memory usage: 2.1+ MB

#### Removing the duplicates and preventing problems with indexing.

In [6]:

df = df.drop\_duplicates()

df = df.reset\_index(drop=True)

#### We can see that the date format is not the one we need. I will apply the appropriate date format for future purposes.

In [7]:

*# Correcting some data*

df['date'] = df['date'].r eplace(['19-Feb-18'],'February 19, 2018')

df['date'] = df['date'].r eplace(['18-Feb-18'],'February 18, 2018')

df['date'] = df['date'].r eplace(['17-Feb-18'],'February 17, 2018')

df['date'] = df['date'].r eplace(['16-Feb-18'],'February 16, 2018')

df['date'] = df['date'].r eplace(['15-Feb-18'],'February 15, 2018')

df['date'] = df['date'].r eplace(['14-Feb-18'],'February 14, 2018')

df['date'] = df['date'].r eplace(['13-Feb-18'],'February 13, 2018')

df['date'] = df['date'].str.replace('Dec ', 'December ')

df['date'] = df['date'].str.replace('Nov ', 'November ')

df['date'] = df['date'].str.replace('Oct ', 'October ')

df['date'] = df['date'].str.replace('Sep ', 'September ')

df['date'] = df['date'].str.replace('Aug ', 'August ')

df['date'] = df['date'].str.replace('Jul ', 'July ')

df['date'] = df['date'].str.replace('Jun ', 'June ')

df['date'] = df['date'].str.replace('Apr ', 'April ')

df['date'] = df['date'].str.replace('Mar ', 'March ')

df['date'] = df['date'].str.replace('Feb ', 'February ')

df['date'] = df['date'].str.replace('Jan ', 'January ')

In [8]:

df['date'] = df['date'].str.replace(' ', '')

In [9]:

for i, val **in** enumerate(df['date']):

df['date'].iloc[i] = pd.to\_datetime(df['date'].iloc[i], format='%B**%d**,%Y', errors='coerce') *# by setting the parameter to "coerce", we will set unappropriate values to NaT (null)*

/opt/conda/lib/python3.7/site-packages/pandas/core/indexing.py:1636: SettingWithCopyWarning: A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas- docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy

self.\_setitem\_single\_block(indexer, value, name)

In [10]:

df['date'] = df['date'].a stype('datetime64[ns]')

In [11]:

df.info()

<class 'pandas.core.frame.DataFrame'> RangeIndex: 44689 entries, 0 to 44688 Data columns (total 5 columns):

# Column Non-Null Count Dtype

1. title 44689 non-null object
2. text 44689 non-null object
3. subject 44689 non-null object
4. date 44679 non-null datetime64[ns]
5. ﬂag 44689 non-null int64

dtypes: datetime64[ns](1), int64(1), object(3) memory usage: 1.7+ MB

In [12]:

import datetime as dt

df['year'] = pd.to\_datetime(df['date']).dt.to\_period('Y')

df['month'] = pd.to\_datetime(df['date']).dt.to\_period('M')

df['month'] = df['month'].a stype( str)

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)

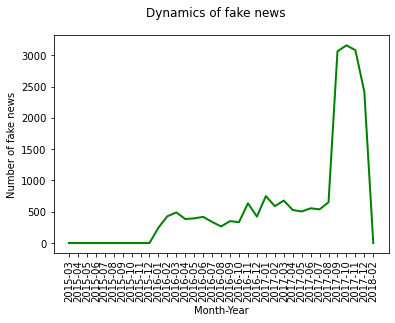
plt.ylabel('Number of fake news')

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)

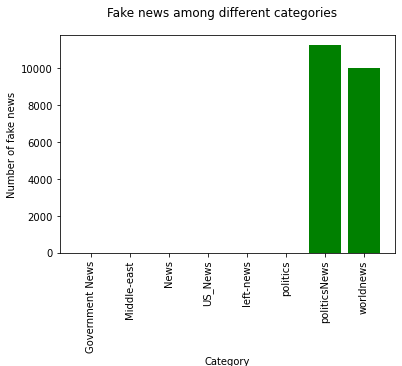
plt.ylabel('Number of fake news')

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()

df\_words = pd.DataFrame(list\_words, columns=feature\_names)

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]:

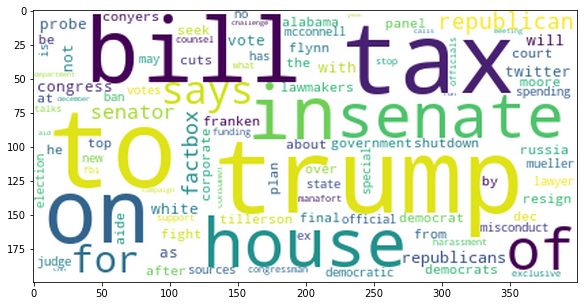
import matplotlib.pyplot as plt

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.

#### And ﬁnally TﬁdfVectorizing. You can also take CountVectorizer, but I prefer Tﬁdf as it has masses of advantages.

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)

# Problem Statment

#### 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. Model building

**I will use several approaches to solve the classiﬁcation task, such as:**

1. Traditional (which are known as eficient for text classiﬁcation):
   1. SVM
   2. Naive Bayes
   3. XGBoost
2. Not-very-traditional (Experimental): PyCaret NLP toolkit (I will apply unsupervised model to generate features which I will in turn pass on to the supervised model)
   1. Linear SVC

In [29]:

scores = {}

In [30]:

from sklearn.svm import LinearSVC

from sklearn.model\_selection import cross\_val\_score from sklearn.metrics import a ccuracy\_score

clf = LinearSVC(max\_iter=100, C=1.0)

clf.ﬁt(X\_train, y\_train)

y\_pred\_SVM = clf.predict(X\_test)

print(cross\_val\_score(clf, X\_text, n lp['ﬂag'], cv=3)) print(a ccuracy\_score(y\_pred\_SVM, y\_test))

scores['LinearSVC'] = a ccuracy\_score(y\_pred\_SVM, y\_test) [0.91105592 0.93031686 0.92696026]

0.958706265256306

#### This looks suspiciously good, but lets try another algorithm.

* 1. Naive Bayes

In [31]:

from sklearn.naive\_bayes import MultinomialNB

clf2 = MultinomialNB()

clf2.ﬁt( X\_train, y\_train)

y\_pred\_MNB = clf2.predict(X\_test)

print(cross\_val\_score(clf2, X\_text, n lp['ﬂag'], cv=3))

print(a ccuracy\_score(y\_pred\_MNB, y\_test))

scores['MultinomialNB'] = a ccuracy\_score(y\_pred\_MNB, y\_test) [0.88957508 0.89406552 0.92883996]

0.939924057499322

#### Okay, this model performs a little worse, but still very good.

* 1. XGBoost

In [32]:

from xgboost import XGBClassiﬁer

clf3 = XGBClassiﬁer(eval\_metric='rmse', use\_label\_encoder=False)

clf3.ﬁt(X\_train, y\_train)

y\_pred\_XGB = clf3.predict(X\_test)

print(cross\_val\_score(clf3, X\_text, n lp['ﬂag'], cv=3)) print(a ccuracy\_score(y\_pred\_XGB, y\_test))

scores['XGB'] = a ccuracy\_score(y\_pred\_XGB, y\_test) [0.88615157 0.92353652 0.90695489]

0.9374830485489558

# 6. Conclusion

#### NLP can be used to detect and classify fake news, and can be a powerful tool in ﬁghting its damaging eGects. By exploring diGerent techniques and approaches, we can gain an understanding of how NLP can be used to combat fake news.