

**FAKE NEWS DETECTION USING NLP**

PHASE-5

Group Name:

V.Pravina

K.Sathya

G.Divyadharshini

P.Sathishkumar

**Problem statement:**

* *Problem*: The spread of fake news and misinformation on social media platforms and news websites is a significant problem that can have real-world consequences, including public panic, distrust, and even violence.
* *Objective*: To develop a robust NLP-based fake news detection system that can automatically classify news articles or social media posts as either "real" or "fake" based on their content.

**Solution:**

* Addressing the spread of fake news and misinformation requires a multi-faceted approach. Social media platforms and news websites must implement stringent fact-checking mechanisms and algorithms that prioritize accuracy over sensationalism. Media literacy education should be promoted to help individuals discern credible sources from unreliable ones. Governments and regulatory bodies can play a role in enforcing transparency and accountability in online content. Additionally, fostering a culture of critical thinking and responsible sharing among users is essential. Ultimately, collaboration between tech companies, educators, governments, and the public is necessary to combat this pressing issue and safeguard against the real-world consequences of misinformation.

**Project overview:**

* The project involves the following steps:

1. **Data Collection**: Gather a large dataset of labeled news articles or social media posts, with clear indications of their authenticity (real or fake).
2. **Data Preprocessing**: Clean and preprocess the text data, including tasks like text normalization, tokenization, and removal of stopwords and special characters.
3. **Feature Extraction**: Extract relevant features from the text, such as TF-IDF vectors, word embeddings (e.g., Word2Vec, GloVe), or BERT embeddings.
4. **Model Selection**: Choose an appropriate NLP model for fake news detection. Common choices include traditional machine learning models (e.g., Random Forest, Support Vector Machines) or deep learning models (e.g., LSTM, BERT).

**Architecture:**

* For this project, let's consider using a deep learning model like BERT (Bidirectional Encoder Representations from Transformers) for fake news detection:
* **Input**: Preprocessed text data (e.g., news articles or social media posts).
* **Model**: A pre-trained BERT model fine-tuned on the labeled dataset for binary classification (real or fake).
* **Output**: Binary classification output indicating whether the input text is real or fake news.

**Model training:**

1. **Data Splitting**: Split the dataset into training, validation, and test sets.
2. **Model Initialization**: Load a pre-trained BERT model.
3. **Fine-Tuning**: Fine-tune the BERT model on the training data using binary cross-entropy loss.
4. **Hyperparameter Tuning**: Optimize hyperparameters, including learning rate, batch size, and dropout rates, using the validation set.
5. **Model Evaluation**:
   1. Evaluate the model on the test set using metrics such as accuracy, precision, recall, F1-score, and ROC-AUC.
   2. Analyze confusion matrices to understand model performance.

**Development:**

* Develop Python scripts or Jupyter notebooks for data preprocessing, feature extraction, model training, and evaluation.
* Implement necessary functions for data loading, model building, and evaluation.
* Ensure code modularity and documentation for easy maintenance.

**User interface:**

* Create a user-friendly web-based interface for users to interact with the fake news detection system:
* Input: Users can enter text or upload news articles or social media posts.
* Output: Display the classification result (real or fake) along with confidence scores.
* Design an intuitive and responsive user interface using frameworks like Flask, Django, or React.

**Conclusion:**

* Summarize the project's goals and objectives.
* Highlight the significance of fake news detection in today's digital age.
* Discuss the chosen architecture and model.
* Present the model's performance metrics.
* Reflect on challenges encountered during development.
* Suggest potential improvements or future work, such as incorporating real-time data and improving model interpretability.

**DATA CLEANING:**

Input:

buzzfeed\_real**$**source <- **with**(buzzfeed\_real, **reorder**(source, source, **function**(x) **length**(x)))

**ggplot**(data = buzzfeed\_real) **+**

**ggtitle**("source count of real news in Buzzfeed") **+**

**geom\_bar**(**aes**(x= source),fill = "green") **+** **coord\_flip**()

Output:

df = df\_merge.drop(["title", "subject","date"], axis = 1)

df.isnull().sum()

text 0

status 0

dtype: int64

df = df.sample(frac = 1)

df.reset\_index(inplace = True)

df.drop(["index"], axis = 1, inplace = True)

df.columns

Index(['text', 'status'], dtype='object')

df.head()

|  | text | status |
| --- | --- | --- |
| 0 | (Reuters) - Maryland’s House of Delegates on M... | 1 |
| 1 | MOSCOW (Reuters) - Russian Deputy Foreign Mini... | 1 |
| 2 | 40% of Americans have given up looking for wor... | 0 |
| 3 | Whether Republican nominee Donald Trump actual... | 0 |
| 4 | Republicans are repulsive people. This is a kn... | 0 |

def wp(text):

text = text.lower()

text = re.sub('\[.\*?\]', '', text)

text = re.sub("**\\**W"," ",text)

text = re.sub('https?://\S+|www\.\S+', '', text)

text = re.sub('<.\*?>+', '', text)

text = re.sub('[**%s**]' % re.escape(string.punctuation), '', text)

text = re.sub('**\n**', '', text)

text = re.sub('\w\*\d\w\*', '', text)

return text

df["text"] = df["text"].apply(wp)

x = df["text"]

y = df["status"]

**EDT :**

Input:

df.info()

df = df.drop\_duplicates()

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

*# Correcting some data*

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

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

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

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

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

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

df['date'] = df['date'].replace(['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]:

linkcode

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)*

Output:

[<matplotlib.lines.Line2D at 0x7f16df024f10>]

<BarContainer object of 8 artists>

<matplotlib.image.AxesImage at 0x7f16b19d6490>

**PREPROCESSING :**

Input:

*#Data Pre-processing*

import numpy as np

import pandas as pd

*#Data Visualisation*

import seaborn as sns

import matplotlib.pyplot as plt

import plotly.express as px

import plotly.graph\_objects as go

*#Handling Warnings*

import warnings

warnings.filterwarnings('ignore')

*#Text pre-processing*

import string

string.punctuation

import re

import nltk

from nltk.stem.porter

import PorterStemmer

from nltk.stem import WordNetLemmatizer

*#Machine Learning*

from sklearn.model\_selection import train\_test\_split,GridSearchCV,RandomizedSearchCV

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.feature\_selection import SelectKBest,chi2,f\_classif

from sklearn.ensemble import RandomForestClassifier,VotingClassifier,AdaBoostClassifier,GradientBoostingClassifier,BaggingClassifier

from sklearn.naive\_bayes import GaussianNB

from sklearn.naive\_bayes import BernoulliNB

from sklearn.naive\_bayes import MultinomialNB

from sklearn.neighbors import KNeighborsClassifier

from sklearn.tree import DecisionTreeClassifier

import xgboost as xgb

from sklearn import tree

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

from xgboost import XGBClassifier

import os

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

for filename **in** filenames:

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

Output:

    "                                                   text label  \n",  
       "0     Daniel Greenfield, a Shillman Journalism Fello...  FAKE  \n",  
       "1     Google Pinterest Digg Linkedin Reddit Stumbleu...  FAKE  \n",  
       "2     U.S. Secretary of State John F. Kerry said Mon...  REAL  \n",  
       "3     — Kaydee King (@KaydeeKing) November 9, 2016 T...  FAKE  \n",  
       "4     It's primary day in New York and front-runners...  REAL  \n",  
       "...                                                 ...   ...  \n",  
       "6330  The State Department told the Republican Natio...  REAL  \n",  
       "6331  The ‘P’ in PBS Should Stand for ‘Plutocratic’ ...  FAKE  \n",  
       "6332   Anti-Trump Protesters Are Tools of the Oligar...  FAKE  \n",  
       "6333  ADDIS ABABA, Ethiopia —President Obama convene...  REAL  \n",  
       "6334  Jeb Bush Is Suddenly Attacking Trump. Here's W...  REAL  \n",  
       "\n",

**Model Training:**

Input:

def output\_lable(n):

if n == 0:

return "Fake News"

elif n == 1:

return "Not A Fake News"

def manual\_testing(news):

testing\_news = {"text":[news]}

new\_def\_test = pd.DataFrame(testing\_news)

new\_def\_test["text"] = new\_def\_test["text"].apply(wordopt)

new\_x\_test = new\_def\_test["text"]

new\_xv\_test = vectorization.transform(new\_x\_test)

pred\_LR = LR.predict(new\_xv\_test)

pred\_DT = DT.predict(new\_xv\_test)

pred\_GBC = GBC.predict(new\_xv\_test)

pred\_RFC = RFC.predict(new\_xv\_test)

return print("**\n\n**LR Prediction: **{}** **\n**DT Prediction: **{}** **\n**GBC Prediction: **{}** **\n**RFC Prediction: **{}**".format(output\_lable(pred\_LR[0]), output\_lable(pred\_DT[0]),

output\_lable(pred\_GBC[0]),

output\_lable(pred\_RFC[0])))

Output:

3488 The White House on Wednesday disclosed a group...

4358 Neil Gorsuch, President Donald Trump’s appoint...

4465 WASHINGTON The clock began running out this we...

5784 Federal appeals court judge Neil Gorsuch, the ...

6660 Republican members of Congress are complaining...

6823 Over the course of the U.S. presidential campa...

7922 After going through a week reminiscent of Napo...

8194 The following timeline charts the origin and s...

8195 Global health officials are racing to better u...

8247 U.S. President Barack Obama visited a street m...

8465 ALGONAC, MICH.—Parker Fox drifted out of the D...

8481 Global health officials are racing to better u...

8482 The following timeline charts the origin and s...

8505 Global health officials are racing to better u...

8506 The following timeline charts the origin and s...

8771 In a speech weighted with America’s complicate...

8970

9008 The following timeline charts the origin and s...

9009 Global health officials are racing to better u...

9307 It’s the near future, and North Korea’s regime...

9618 GOP leaders have unleashed a stunning level of...

9737 Caitlyn Jenner posted a video on Wednesday (Ap...

10479 The Democratic and Republican nominees for the...

**Model Building:**

Input:

model = Sequential()

model.add(Embedding(input\_dim=vocabSize,output\_dim=150,input\_length=300))

model.add(Dropout(0.2))

model.add(LSTM(128))

model.add(Dropout(0.2))

model.add(Dense(64,activation='sigmoid'))

model.add(Dropout(0.2))

model.add(Dense(4,activation='softmax')

news = str(input())

manual\_testing(news)

Output:

[LGBMClassifier(boosting\_type='gbdt', class\_weight=None, colsample\_bytree=1.0,

importance\_type='split', learning\_rate=0.1, max\_depth=-1,

min\_child\_samples=20, min\_child\_weight=0.001, min\_split\_gain=0.0,

n\_estimators=100, n\_jobs=-1, num\_leaves=31, objective=None,

random\_state=2728, reg\_alpha=0.0, reg\_lambda=0.0, silent='warn',

subsample=1.0, subsample\_for\_bin=200000, subsample\_freq=0),

<catboost.core.CatBoostClassifier at 0x7f1c47b93790>, GradientBoostingClassifier(ccp\_alpha=0.0, criterion='friedman\_mse', init=None,

learning\_rate=0.1, loss='deviance', max\_depth=3,

max\_features=None, max\_leaf\_nodes=None,

min\_impurity\_decrease=0.0, min\_impurity\_split=None

min\_samples\_leaf=1, min\_samples\_split=2,

min\_weight\_fraction\_leaf=0.0, n\_estimators=100,

n\_iter\_no\_change=None, presort='deprecated',

random\_state=2728, subsample=1.0, tol=0.0001,

validation\_fraction=0.1, verbose=0,

warm\_start=False),

XGBClassifier(base\_score=0.5, booster='gbtree', colsample\_bylevel=1,

colsample\_bynode=1, colsample\_bytree=1, enable\_categorical=False,

gamma=0, gpu\_id=-1, importance\_type=None,

interaction\_constraints='', learning\_rate=0.300000012,

max\_delta\_step=0, max\_depth=6, min\_child\_weight=1, missing=nan,

monotone\_constraints='()', n\_estimators=100, n\_jobs=-1,

num\_parallel\_tree=1, objective='binary:logistic',

predictor='auto', random\_state=2728, reg\_alpha=0, reg\_lambda=1,

scale\_pos\_weight=1, subsample=1, tree\_method='auto',

use\_label\_encoder=True, validate\_parameters=1, verbosity=0),

RandomForestClassifier(bootstrap=True, ccp\_alpha=0.0, class\_weight=None,

criterion='gini', max\_depth=None, max\_features='auto',

max\_leaf\_nodes=None, max\_samples=None,

min\_impurity\_decrease=0.0, min\_impurity\_split=None,

min\_samples\_leaf=1, min\_samples\_split=2,

min\_weight\_fraction\_leaf=0.0, n\_estimators=100,

n\_jobs=-1, oob\_score=False, random\_state=2728, verbose=0,

warm\_start=False)]

**Evaluation:**

Input:

linkcode

cv\_report = classification\_report(y\_test,pred,target\_names = ['0','1'])

print(cv\_report)

Output:

0 as u.s. budget fight looms, republicans flip t...

1 u.s. military to accept transgender recruits o...

2 senior u.s. republican senator: 'let mr. muell...

3 fbi russia probe helped by australian diplomat...

4 trump wants postal service to charge 'much mor...

0 as us budget fight looms republicans flip thei...

1 us military to accept transgender recruits on ...

2 senior us republican senator let mr mueller do...

3 fbi russia probe helped by australian diplomat...

4 trump wants postal service to charge much more...

**Display:**

Input:

# Import libraries

**library**(tm) # for NLP

**library**(plyr) # for pre-processing

**library**(tidyverse) # for pre-processing and visualisation

**library**(reshape2) # for melt function

**library**(e1071) # for Naive Bayes classifier

**library**(glmnet) # for Logistic Regression classifier

**library**(randomForest) # for Random Forest classifier

Output:

'from nltk.stem import WordNetLemmatizer \n\nlemmatizer = WordNetLemmatizer()\n\ndef lemmatize\_text(text):\n return [lemmatizer.lemmatize(w) for w in df["article"]]\n\ndf[\'article\'] = df["article"].apply(lemmatize\_text)'