FAKE NEWS DETECTION USING NLP

BATCH MEMBER

620121243050 : SOWMIYA. R

Phase 3 submission document

Project Title: Fake news detection

Phase 3: Development part 1

Topic: Start building the fake news detection model by loading and pre-processing the dataset.

Fake news detection

Introduction:

- Fake news detection using Natural Language Processing (NLP) is a critical application in the field of data science and information security. NLP techniques can be employed to automatically identify and classify fake or misleading information in textual content. Here's a brief introduction to the process

- **Data collection:** The first step is to gather a diverse dataset of news articles, encompassing both genuine and fake news, preferably labeled or annotated.

- **Text processing:** cleanand preprocess the text data by removing stopwords, punctuation, and other irrelevant elements. Tokenization and stemming/lemmatization can also be applied.

- **Future Extraction:** Convert the textual data into numerical features that machine learning algorithms can work with. Common techniques include TF-IDF (Term Frequency-Inverse Document Frequency) and word embeddings like Word2Vec or GloVe..

- **Training:** Train the selected model on the labeled dataset. This involves feeding the model both the text data and their corresponding labels (fake or genuine).

- **Validation and Testing:** Assess the model's performance using validation data to fine-tune hyperparameters. Then, evaluate the model's accuracy, precision, recall, and F1-score on a test dataset.

- **Ensemble Methods:** Combining multiple models or using ensemble techniques can often improve detection accuracy.

- **Post-processing:** Apply post-processing techniques to refine the model's output, such as setting a confidence threshold for classifying news as fake.

- **Continuous Learning:** Fake news evolves, so the model should be updated regularly to adapt to new forms of disinformation.

	A	B	C	D
1	title	text	subject	date
2	Donald Trum	Donald Trum	News	31-Dec-17
3	Drunk Bragg	House Intelli	News	31-Dec-17
4	Sheriff Davic	On Friday, it v	News	30-Dec-17
5	Trump Is So	On Christmas	News	29-Dec-17
6	Pope Franci	Pope Francis	News	25-Dec-17
7	Racist Alaba	The number	News	25-Dec-17
8	Fresh Off Th	Donald Trum	News	23-Dec-17
9	Trump Said	In the wake c	News	23-Dec-17
10	Former CIA I	Many people	News	22-Dec-17
11	WATCH: Bra	Just when yo	News	21-Dec-17
12	Papa John's	A centerpiec	News	21-Dec-17
13	WATCH: Pau	Republicans	News	21-Dec-17
14	Bad News F	Republicans	News	21-Dec-17
15	WATCH: Lin	The media ha	News	20-Dec-17
16	Heiress To D	Abigail Disne	News	20-Dec-17
17	Tone Deaf T	Donald Trum	News	20-Dec-17
18	The Internet	A new anima	News	19-Dec-17
19	Mueller Spo	Trump suppc	News	17-Dec-17
20	SNL Hilariou	Right now, th	News	17-Dec-17
21	Republican S	Senate Majo	News	16-Dec-17
22	In A Heartles	It almost see	News	16-Dec-17
23	KY GOP Stat	In this #MET	News	13-Dec-17
24	Meghan Mc	As a Democr	News	12-Dec-17
25	CNN CALLS	Alabama is a	News	12-Dec-17
26	White House	A backlash e	News	12-Dec-17

Given data set:

Input:1

```
fake = pd.read_csv('../input/fake-and-real-news-dataset/Fake.csv')
fake['flag'] = 0
fake
```

Output:1

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

Input:2

```
true = pd.read_csv('../input/fake-and-real-news-dataset/True.csv')
true['flag'] = 1
true
```

Output:2

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

Input:3

```
df = pd.DataFrame()
df = true.append(fake)
```

Input:4

```
df.info()
```

Input:5

```
df = df.drop_duplicates()
df = df.reset_index(drop=True)
```

Input:6

```
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 ')
```

Input:7

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

Input:8

```
for i, val in enumerate(df['date']):
    df['date'].iloc[i] = pd.to_datetime(df['date'].iloc[i], format='%B%d,%
Y', errors='coerce')
```

Input:9

```
df['date'] = df['date'].astype('datetime64[ns]')
```

Input:10

```
df.info()
```

Input:11

```
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'].astype(str)
```

Input:12

```
sub = df[['month', 'flag']]
sub = sub.dropna()
sub = sub.groupby(['month'])['flag'].sum()
```

Input:13

```
sub = sub.drop('NaT')
```

Input:14

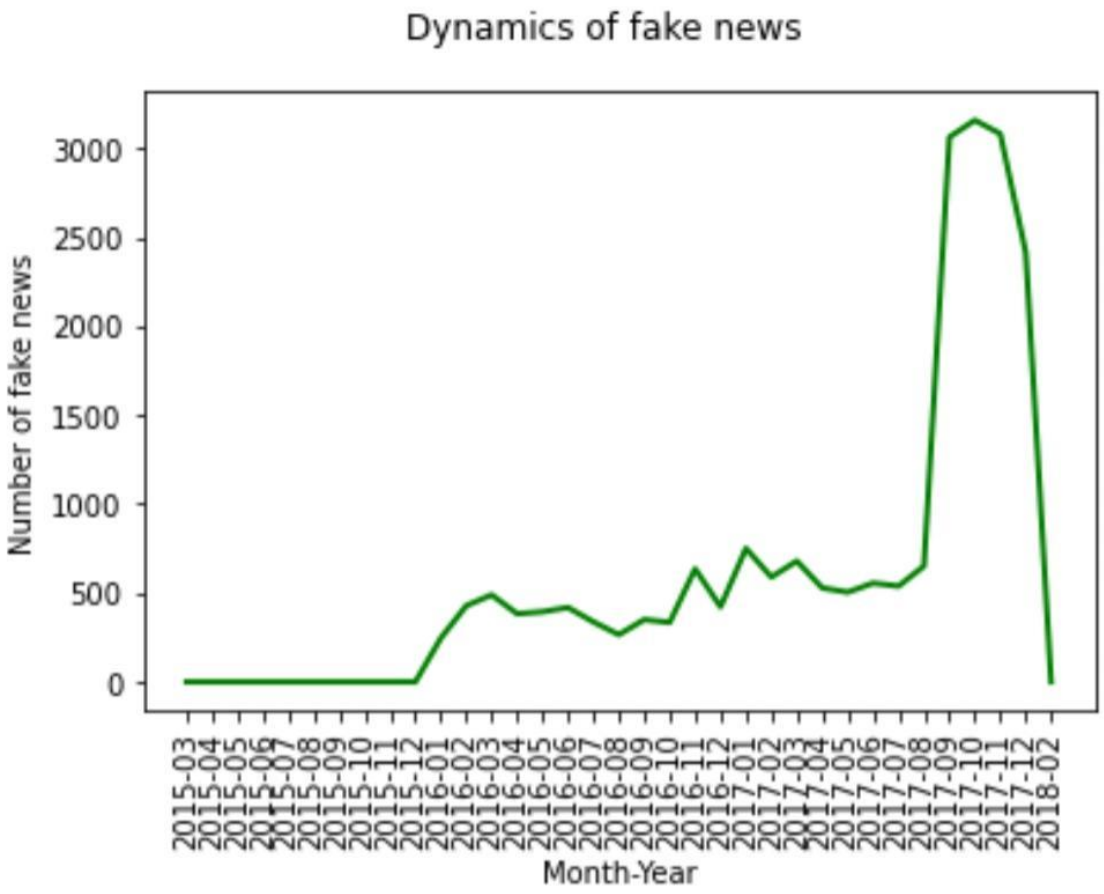
```
import matplotlib.pyplot as plt

plt.suptitle('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')
```

Output:14

```
[<matplotlib.lines.Line2D at 0x7f16df0
24f10>]
```



Input:15

```
sub2 = df[['subject', 'flag']]
sub2 = sub2.dropna()
```

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

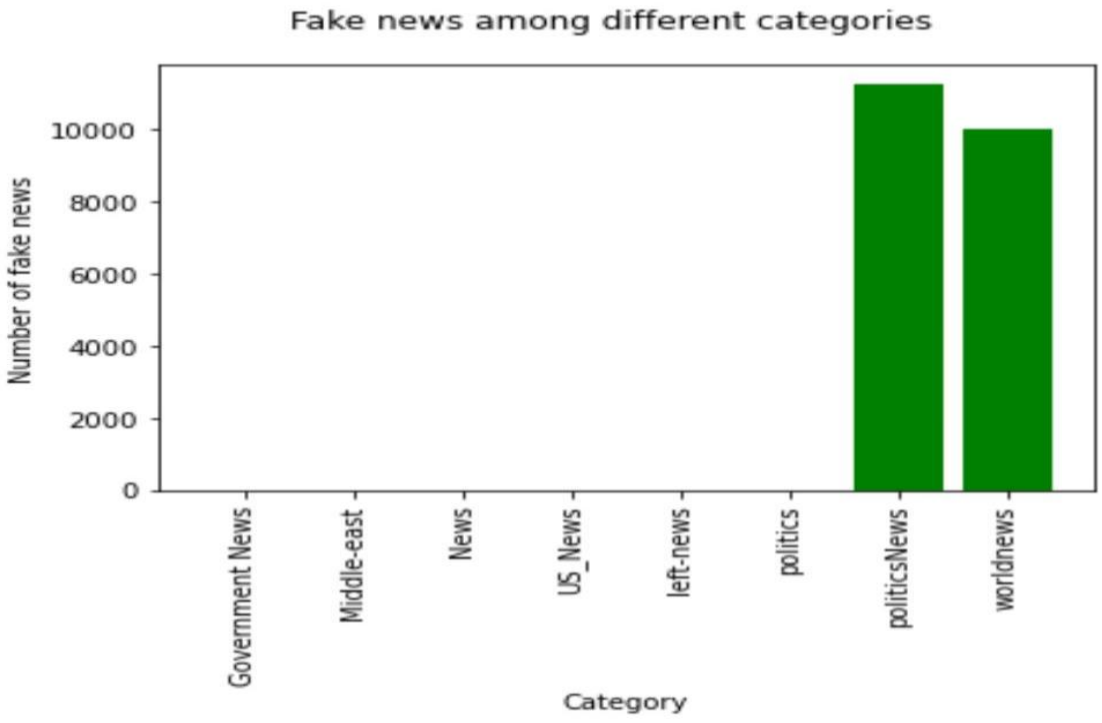
Input:16

```
plt.suptitle('Fake news among different categories')
plt.xticks(rotation=90)
plt.ylabel('Number of fake news')
plt.xlabel('Category')

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

Output:16

```
<BarContainer object of 8 artists>
```



Input:17

```
nlp = df
```

Input:18

```
from sklearn.feature_extraction.text import TfidfVectorizer

corpus = nlp[nlp['flag'] == 1]['title'].iloc[0:500]
tfidf1 = TfidfVectorizer()
vecs = tfidf1.fit_transform(corpus)

feature_names = tfidf1.get_feature_names()
dense = vecs.todense()
list_words = dense.tolist()
df_words = pd.DataFrame(list_words, columns=feature_names)
```

Input:19

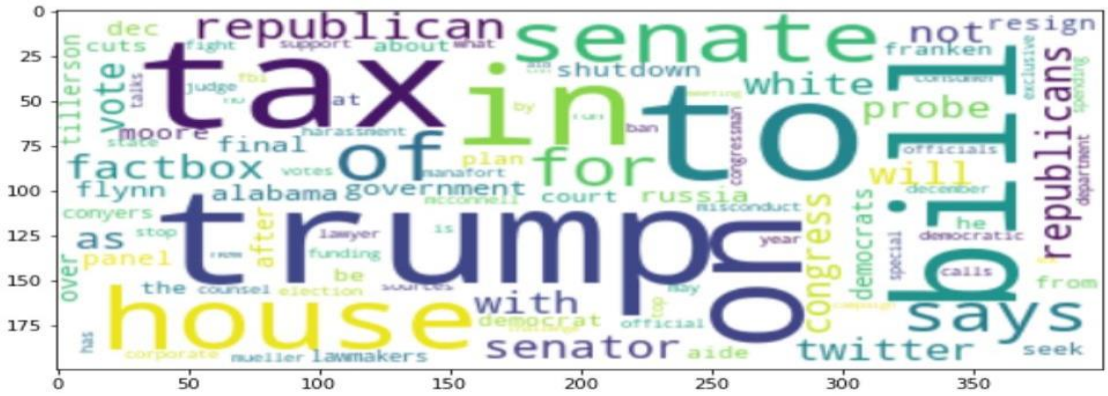
```
from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator
df_words.T.sum(axis=1)
Cloud = WordCloud(background_color="white", max_words=100).generate_from_f
requencies(df_words.T.sum(axis=1))
```

Input:20

```
import matplotlib.pyplot as plt
plt.figure(figsize=(12,5))
plt.imshow(Cloud, interpolation='bilinear')
```

Output:20

```
<matplotlib.image.AxesImage at 0x7f16b
19d6490>
```



Input:21

```
import nltk
nltk.download('punkt')
from nltk import word_tokenize

nlp['title'] = nlp['title'].apply(lambda x: word_tokenize(str(x)))
```

Input:22

```
from nltk.stem import SnowballStemmer

snowball = SnowballStemmer(language='english')
nlp['title'] = nlp['title'].apply(lambda x: [snowball.stem(y) for y in x])
```

Input:23

```
nlp['title'] = nlp['title'].apply(lambda x: ' '.join(x))
```

Input:24

```
from nltk.corpus import stopwords

nltk.download('words')
nltk.download('stopwords')
```

```
stopwords = stopwords.words('english')
```

Input:25

```
from sklearn.feature_extraction.text import TfidfVectorizer

tfidf = TfidfVectorizer()
X_text = tfidf.fit_transform(nlp['title'])
```

Input:26

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X_text, nlp['flag'], t
est_size=0.33, random_state=1)
```

Input:27

```
scores = {}
```

Input:28

```
from sklearn.svm import LinearSVC
from sklearn.model_selection import cross_val_score
from sklearn.metrics import accuracy_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, nlp['flag'], cv=3))
print(accuracy_score(y_pred_SVM, y_test))

scores['LinearSVC'] = accuracy_score(y_pred_SVM, y_test)
```

Output:28

```
[0.91105592 0.93031686 0.92696026]
0.958706265256306
```

Input:29

```
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, nlp['flag'], cv=3))
print(accuracy_score(y_pred_MNB, y_test))

scores['MultinomialNB'] = accuracy_score(y_pred_MNB, y_test)
```

Output:29

```
[0.88957508 0.89406552 0.92883996]
0.939924057499322
```

Input:30

```
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, nlp['flag'], cv=3))
print(accuracy_score(y_pred_XGB, y_test))

scores['XGB'] = accuracy_score(y_pred_XGB, y_test)
```

Output:30

```
[0.88615157 0.92353652 0.90695489]
0.9374830485489558
```

Input:31

```
pip install pycaret
```

Input:32

```
from pycaret.nlp import *

caret_nlp = setup(data=nlp, target='title', session_id=1)
```

Output:32

Description	Value
session_id	1
Documents	44689
Vocab Size	7568
Custom Stopwords	False

Input:33

```
lda = create_model('lda')
```

Input:34

```
lda_data = assign_model(lda)
```

Input:35

```
lda_data
```

Output:35

	title	text	subject
0	budget fight loom flip fiscal script	WASHINGTON (Reuters) - The head of a conservat...	politicsNe
1	transgend	WASHINGTON (Reuters) - Transgender people will...	politicsNe
2	senior let job	WASHINGTON (Reuters) - The special counsel inv...	politicsNe
3	diplomat	WASHINGTON (Reuters) - Trump campaign adviser ...	politicsNe
4	trump want postal charg much shipment	SEATTLE/WASHINGTON (Reuters) - President Donal...	politicsNe
...	...	...	...
44684	furious treat sailor well	21st Century Wire says As 21WIRE reported earl...	Middle-ea
44685	settl lawyer user	21st Century Wire says It s a familiar theme. ...	Middle-ea

Input:36

```
from catboost import CatBoostClassifier
```

Input:37

```
input_cat = lda_data.drop(['text','date','Perc_Dominant_Topic','flag','yea
r'], axis=1)
input_cat['month'] = input_cat['month'].astype(str)
target_cat = lda_data['flag']
```

Input:38

```
from sklearn.model_selection import train_test_split
X_train_cat, X_test_cat, y_train_cat, y_test_cat = train_test_split(input_
cat, target_cat, test_size=0.33, random_state=1)
```

Input:39

```
clf4 = CatBoostClassifier(iterations=1000,
                          cat_features=['title','subject','Dominant_Topic'
,'month']
                         )
```

Input:40

```
clf4.fit(X_train_cat, y_train_cat, early_stopping_rounds=10)
```

Output:40

```
<catboost.core.CatBoostClassifier at 0x7f167ddb8a50>
```

Input:41

```
scores['CatBoost'] = clf4.score(X_test_cat, y_test_cat)
```

Input:42

```
scores['CatBoost'] = clf4.score(X_test_cat, y_test_cat)
```

Output:42

```
{'LinearSVC': 0.958706265256306,
 'MultinomialNB': 0.939924057499322,
 'XGB': 0.9374830485489558,
 'CatBoost': 1.0}
```

Input:43

```
plt.bar(scores.keys(), scores.values())
```

Output:43

```
<BarContainer object of 4 artists>
```