FAKE NEWS DETECTION

Phase 4:

Development Part 2

◆ In Phase 4 of the "Fake News Detection Using NLP" project, you will continue building the fake news detection model by applying Natural Language Processing (NLP) techniques, training a classification model, and preparing it for evaluation. This phase is essential for fine-tuning the model and ensuring that it can effectively distinguish between genuine and fake news articles. Here's a detailed outline of Phase 4:

1. NLP Techniques and Text Preprocessing:

Apply NLP techniques to the textual data to prepare it for classification. This involves techniques like:

□ **Text cleaning:** Removing any irrelevant characters, symbols, or HTML tags.

□ **Tokenization**: Splitting text into individual words or tokens.

□ **Stopword removal**: Eliminating common words that do not carry significant meaning.

□ **Lemmatization or stemming**: Reducing words to their base or root form.

□ **Handling imbalanced data (if applicable)**: Ensure that the dataset is balanced between genuine and fake news articles to avoid model bias.

2. Feature Extraction:

Transform text data into numerical features suitable for machine learning. Common techniques include:

- TF-IDF (Term Frequency-Inverse Document Frequency): Convert text into a numerical vector representing the importance of words in documents.
- Word embeddings: Use pre-trained word embeddings like Word2Vec or GloVe to capture semantic meaning.
- Document embeddings: Aggregate word embeddings to represent entire documents or titles.
- Consider the trade-off between computational complexity and feature representation quality when selecting these techniques.

3. Model Selection:

Choose an appropriate classification algorithm for the fake news detection task. Potential choices include:

- **Logistic Regression:** A simple yet effective linear model.
- **Random Forest:** An ensemble method that can handle non-linear relationships.
- Deep Learning Models (e.g., LSTM, BERT): Consider these for advanced performance (as mentioned in Phase 2).
- Experiment with various models and hyperparameters to optimize performance.

4. Model Training:

- Split the preprocessed data into training and testing sets for model evaluation.
- Train the selected classification model using the training data.
- Save the trained model and its associated parameters for future use.

5. Hyperparameter Tuning (if applicable):

Fine-tune the model's hyperparameters to achieve better performance. You can use techniques like grid search or random search.

6. Model Evaluation:

Evaluate the model's performance using various evaluation metrics such as:

- **Accuracy**: The proportion of correctly classified articles.
- **Precision**: The ability to correctly identify genuine news.
- **Recall**: The ability to correctly identify fake news.
- **F1-score**: A balanced measure of precision and recall.
- **ROC-AUC**: The area under the Receiver Operating Characteristic curve, which measures the model's ability to discriminate between classes.

7. Documentation:

- Document all the code and techniques used in text preprocessing, feature extraction, model selection, and evaluation.
- Explain the reasons behind the choices made during model selection and hyperparameter tuning.
- Present the model's evaluation results, highlighting its performance in detecting fake news.

Phase 4 is vital for model development and fine-tuning. It sets the stage for the final phase, where you will document the project and prepare it for submission

Program:

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

import nltk
nltk.download('punkt')

[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.

True

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from wordcloud import WordCloud, STOPWORDS
import nltk
import re
from nltk.corpus import stopwords
import seaborn as sns
import gensim
from gensim.utils import simple_preprocess
from gensim.parsing.preprocessing import STOPWORDS

import plotly.express as px
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import roc_auc_score
from sklearn.metrics import confusion_matrix

fake_data = pd.read_csv('//content//sample_data//Fake.csv')
print("fake_data",fake_data.shape)

true_data= pd.read_csv('//content//sample_data//True.csv')
print("true_data",true_data.shape)

fake_data (23481, 4)
true_data (21417, 4)

fake_data.head(5)

                                               title  \
0   Donald Trump Sends Out Embarrassing New Year’...
1   Drunk Bragging Trump Staffer Started Russian ...
```

```
2   Sheriff David Clarke Becomes An Internet Joke...
3   Trump Is So Obsessed He Even Has Obama’s Name...
4   Pope Francis Just Called Out Donald Trump Dur...

                                                text subject  \
0  Donald Trump just couldn t wish all Americans ...    News
1  House Intelligence Committee Chairman Devin Nu...    News
2  On Friday, it was revealed that former Milwauk...    News
3  On Christmas day, Donald Trump announced that ...    News
4  Pope Francis used his annual Christmas Day mes...    News

                date
0  December 31, 2017
1  December 31, 2017
2  December 30, 2017
3  December 29, 2017
4  December 25, 2017
```

```
true_data.head(5)
```

```
                                               title  \
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...

                                                text       subject  \
0  WASHINGTON (Reuters) - The head of a conservat...  politicsNews
1  WASHINGTON (Reuters) - Transgender people will...  politicsNews
2  WASHINGTON (Reuters) - The special counsel inv...  politicsNews
3  WASHINGTON (Reuters) - Trump campaign adviser ...  politicsNews
4  SEATTLE/WASHINGTON (Reuters) - President Donal...  politicsNews

                 date
0  December 31, 2017
1  December 29, 2017
2  December 31, 2017
3  December 30, 2017
4  December 29, 2017
```

```
#adding additonal column to seperate betwee true & fake data
# true =1, fake =0
true_data['target'] = 1
fake_data['target'] = 0
df = pd.concat([true_data, fake_data]).reset_index(drop = True)
df['original'] = df['title'] + ' ' + df['text']
df.head()
```

```
                                               title  \
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...

                                                text       subject  \
0  WASHINGTON (Reuters) - The head of a conservat...  politicsNews
1  WASHINGTON (Reuters) - Transgender people will...  politicsNews
2  WASHINGTON (Reuters) - The special counsel inv...  politicsNews
3  WASHINGTON (Reuters) - Trump campaign adviser ...  politicsNews
4  SEATTLE/WASHINGTON (Reuters) - President Donal...  politicsNews

                 date  target  \
0  December 31, 2017       1
1  December 29, 2017       1
2  December 31, 2017       1
3  December 30, 2017       1
4  December 29, 2017       1

                                            original
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...
```

```
df.isnull().sum()
```

```
title       0
text        0
subject     0
date        0
target      0
original    0
dtype: int64
```

```
import nltk
nltk.download('stopwords')
stop_words = stopwords.words('english')
stop_words.extend(['from', 'subject', 're', 'edu', 'use'])
def preprocess(text):
    result = []
    for token in gensim.utils.simple_preprocess(text):
        if token not in gensim.parsing.preprocessing.STOPWORDS and len(token)
> 2 and token not in stop_words:
            result.append(token)

    return result
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
```

```
df.subject=df.subject.replace({'politics':'PoliticsNews','politicsNews':'Poli
ticsNews'})
```

```
sub_tf_df=df.groupby('target').apply(lambda
x:x['title'].count()).reset_index(name='Counts')
sub_tf_df.target.replace({0:'False',1:'True'},inplace=True)
fig = px.bar(sub_tf_df, x="target", y="Counts",
             color='Counts', barmode='group',
             height=350)
fig.show()
```

```
sub_check=df.groupby('subject').apply(lambda
x:x['title'].count()).reset_index(name='Counts')
fig=px.bar(sub_check,x='subject',y='Counts',color='Counts',title='Count of
News Articles by Subject')
fig.show()
```

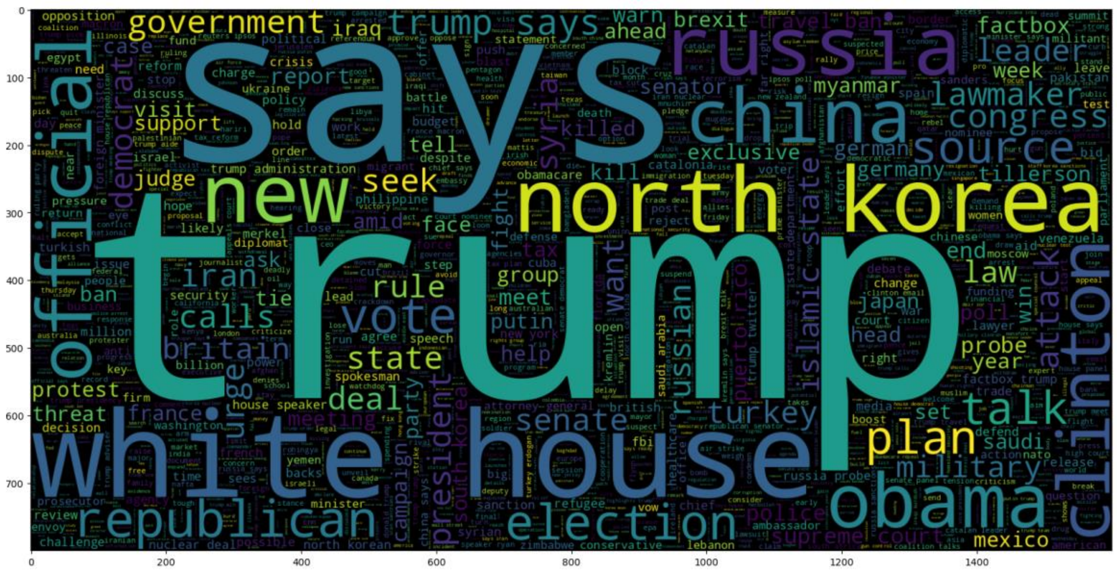
```
df['clean_title'] = df['title'].apply(preprocess)
df['clean_title'][0]
```

```
['budget', 'fight', 'looms', 'republicans', 'flip', 'fiscal', 'script']
```

```
df['clean_joined_title']=df['clean_title'].apply(lambda x:" ".join(x))
```

```
plt.figure(figsize = (20,20))
wc = WordCloud(max_words = 2000 , width = 1600 , height = 800 , stopwords =
stop_words).generate(" ".join(df[df.target == 1].clean_joined_title))
plt.imshow(wc, interpolation = 'bilinear')
```

```
<matplotlib.image.AxesImage at 0x7faeceeece50>
```



```
fig = px.histogram(x = [len(nltk.word_tokenize(x)) for x in
df.clean_joined_title], nbins = 50)
fig.show()
```

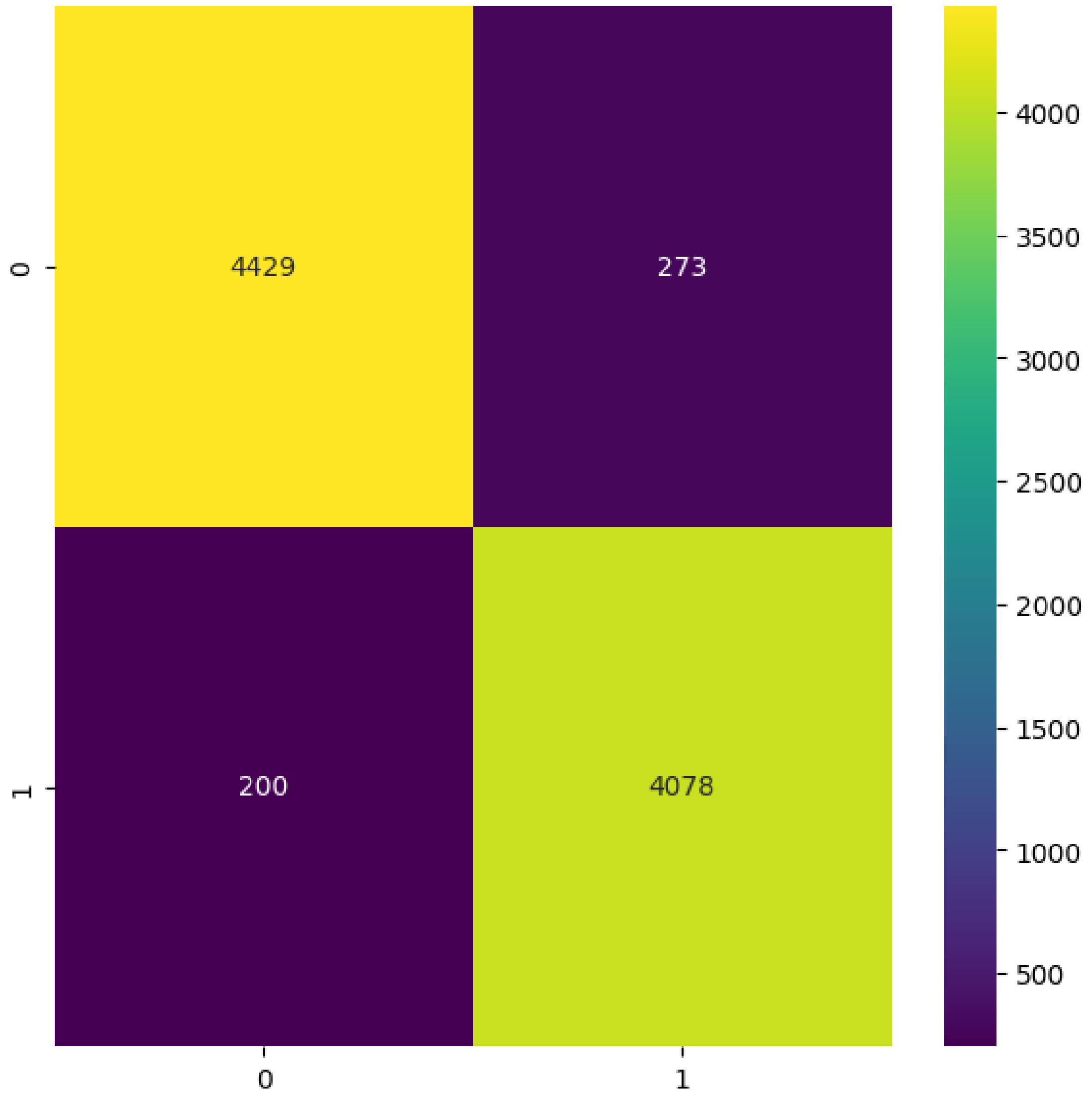
```
The maximum number of words in a title is = 34
```

```
X_train, X_test, y_train, y_test = train_test_split(df.clean_joined_title,
df.target, test_size = 0.2,random_state=2)
vec_train = CountVectorizer().fit(X_train)
X_vec_train = vec_train.transform(X_train)
X_vec_test = vec_train.transform(X_test)

#model
model = LogisticRegression(C=2)

#fit the model
model.fit(X_vec_train, y_train)
predicted_value = model.predict(X_vec_test)

#accuracy & predicted value
accuracy_value = roc_auc_score(y_test, predicted_value)
print(accuracy_value)
```

```
0.9475943910154114

/usr/local/lib/python3.10/dist-
packages/sklearn/linear_model/_logistic.py:458: ConvergenceWarning:

lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
```

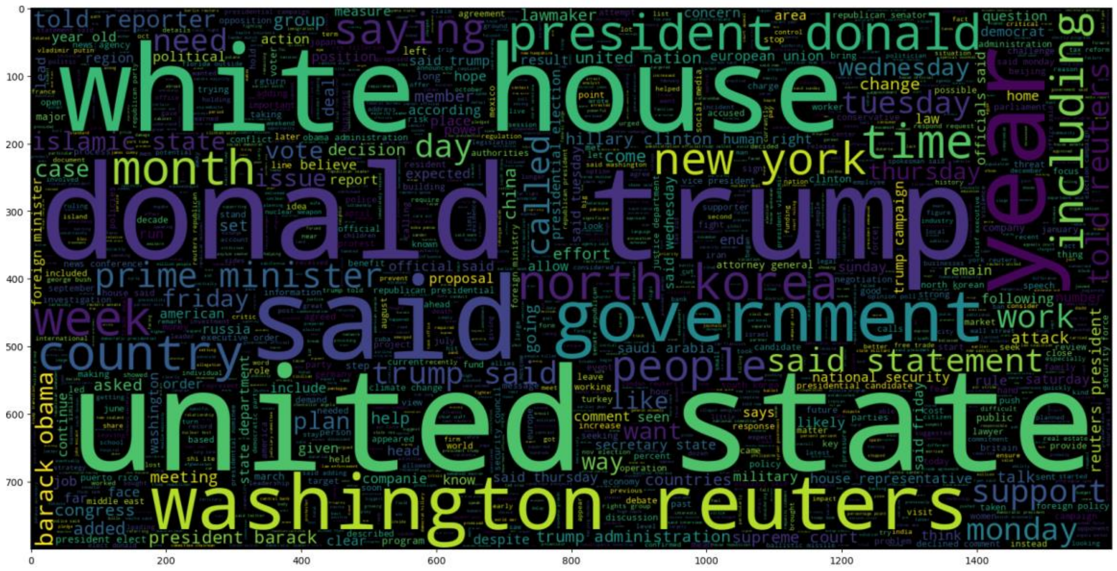
```
cm = confusion_matrix(list(y_test), predicted_value)
plt.figure(figsize = (7, 7))
sns.heatmap(cm, annot = True,fmt='g',cmap='viridis')
```

```
<Axes: >
```



```
df['clean_text'] = df['text'].apply(preprocess)
df['clean_joined_text']=df['clean_text'].apply(lambda x:" ".join(x))

plt.figure(figsize = (20,20))
wc = WordCloud(max_words = 2000 , width = 1600 , height = 800 , stopwords =
stop_words).generate(" ".join(df[df.target == 1].clean_joined_text))
plt.imshow(wc, interpolation = 'bilinear')
```

```
<matplotlib.image.AxesImage at 0x7faecf1c4130>
```



```
maxlen = -1
for doc in df.clean_joined_text:
    tokens = nltk.word_tokenize(doc)
    if(maxlen<len(tokens)):
        maxlen = len(tokens)
print("The maximum number of words in a News Content is =", maxlen)
fig = px.histogram(x = [len(nltk.word_tokenize(x)) for x in
df.clean_joined_text], nbins = 50)
fig.show()
```

```
The maximum number of words in a News Content is = 4573
```

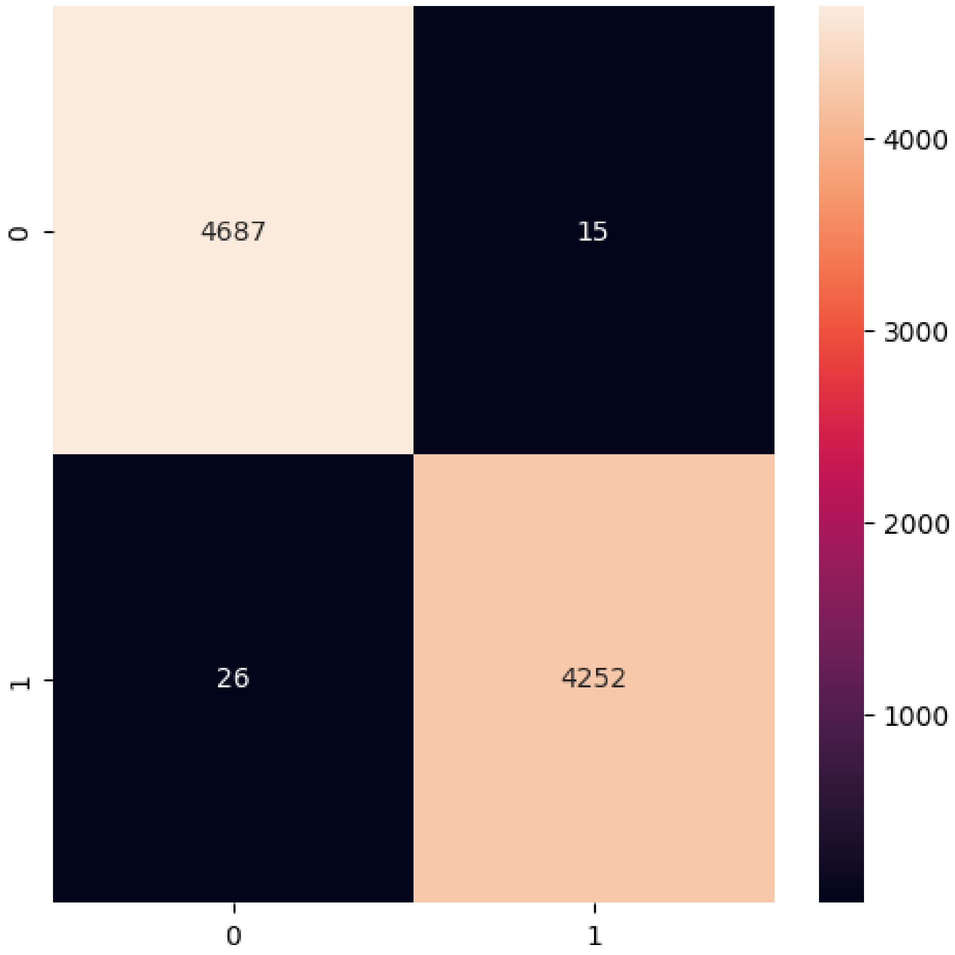
```
X_train, X_test, y_train, y_test = train_test_split(df.clean_joined_text,
df.target, test_size = 0.2,random_state=2)
vec_train = CountVectorizer().fit(X_train)
X_vec_train = vec_train.transform(X_train)
X_vec_test = vec_train.transform(X_test)
model = LogisticRegression(C=2.5)
model.fit(X_vec_train, y_train)
predicted_value = model.predict(X_vec_test)
accuracy_value = roc_auc_score(y_test, predicted_value)
print(accuracy_value)
```

```
0.9953661308915527

/usr/local/lib/python3.10/dist-
packages/sklearn/linear_model/_logistic.py:458: ConvergenceWarning:

lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
```

```
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
```

```
prediction = []
for i in range(len(predicted_value)):
    if predicted_value[i].item() > 0.5:
        prediction.append(1)
    else:
        prediction.append(0)
cm = confusion_matrix(list(y_test), prediction)
plt.figure(figsize = (6, 6))
sns.heatmap(cm, annot = True,fmt='g')
```

```
<Axes: >
```


4687
15
26
4252
0
1
0
1
4000
3000
2000
1000