FAKE NEWS DETECTION USING NLP

BATCH MEMBER

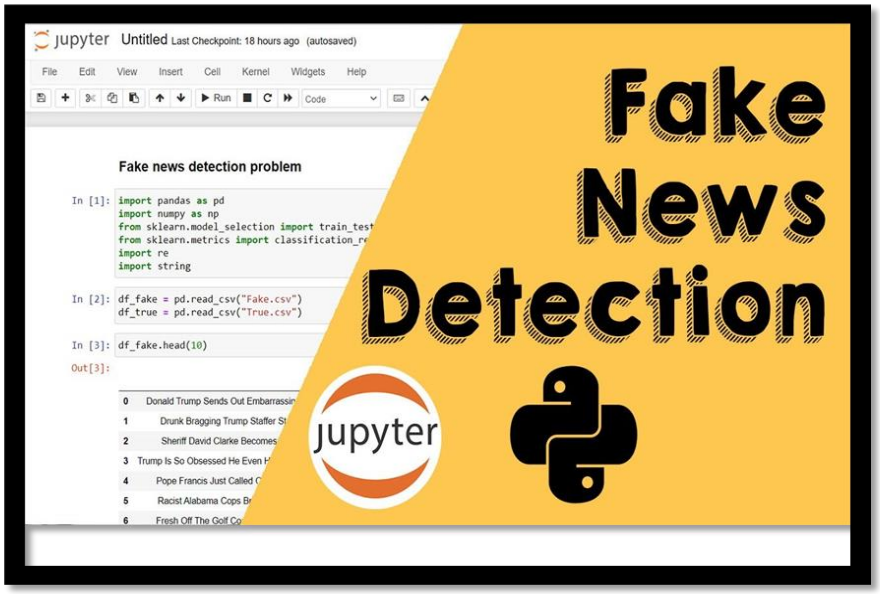
Phase 3 submission document

~~Phase 4 submission document~~

Project Tittle: Fake News Detection

Phase 4: ***Development Part*** **2**

Topic: *Begin building the fake news detection model by loading and pre-processing the dataset*



Fake news detection problem

```
In [1]: import pandas as pd
        import numpy as np
        from sklearn.model_selection import train_test
        from sklearn.metrics import classification_re
        import re
        import string

In [2]: df_fake = pd.read_csv("Fake.csv")
        df_true = pd.read_csv("True.csv")

In [3]: df_fake.head(10)
Out[3]:

0    Donald Trump Sends Out Embarrassin
1       Drunk Bragging Trump Staffer St
2       Sheriff David Clarke Becomes
3  Trump Is So Obsessed He Even H
4      Pope Francis Just Called C
5       Racist Alabama Cops Br
6      Fresh Off The Golf Co
```

Fake News Detection

Introduction:

- Fake news detection using Natural Language Processing (NLP) is a crucial and rapidly evolving area of research and technology designed to combat the spread of false or misleading information in the digital age.

- The proliferation of social media and online platforms, the creation and dissemination of fake news has become a significant problem, impacting public opinion, trust, and even political processes.

- NLP plays a vital role in addressing this challenge by leveraging techniques in machine learning and linguistics to automatically identify and filter out fake news from the vast amount of online content.

- Natural Language Processing is a branch of artificial intelligence that focuses on the interaction between humans and computers through natural language. It involves various tasks such as text classification, sentiment analysis, and language modeling, which can be harnessed for fake news detection.

Necessary step to follow:

1.Import Libraries:

Start by importing the necessary libraries:

Program:

```
import pandas as pd

import numpy as np

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import StandardScaler
```

2.Load the Dataset:

- Load your dataset into a Pandas DataFrame. You can typically find house price datasets in CSV format, but you can adapt this code to other formats as needed.

- Load the dataset into your preferred programming environment. Common tools for this task include Python and libraries like Pandas for handling tabular data or NLTK for natural language processing.

Program:

```
df = pd.read_csv(' E:\USA_Fakenews.csv ')
Pd.read()
```

3.Data Exploration:

After loading the dataset, it's important to perform some initial data exploration. This includes checking the dataset's structure, examining a few sample records, and understanding the distribution of labels (real vs. fake) to gain insights into the data.

Program:

```
# Display the first few rows of the dataset
 print(df.head())
 # Check the distribution of labels
 print(df['label'].value_counts())
```

4. Splitting the Data:

It's common practice to split the dataset into training and testing subsets. The training data is used to train your fake news detection model, while the testing data is used to evaluate the model's performance. Common splits are 70-30 or 80-20, with most of the data allocated for training.

Program:

```
from sklearn.model_selection
import train_test_split X_train, X_test,
y_train, y_test = train_test_split(df['text'],
df['label'], tes_size=0.3,random_state=42)
```

5.Data Preprocessing and Model Building:

With the dataset loaded and split, you can proceed with data preprocessing and build a machine learning or deep learning model to detect fake news. This involves further steps such as text vectorization, model selection, training, and evaluation.

Remember that the quality of the dataset and the preprocessing steps you perform have a significant impact on the performance of your fake news detection model. It's crucial to ensure that your dataset is representative and balanced, and that your preprocessing steps are tailored to the nature of the data and the requirements of your chosen model.

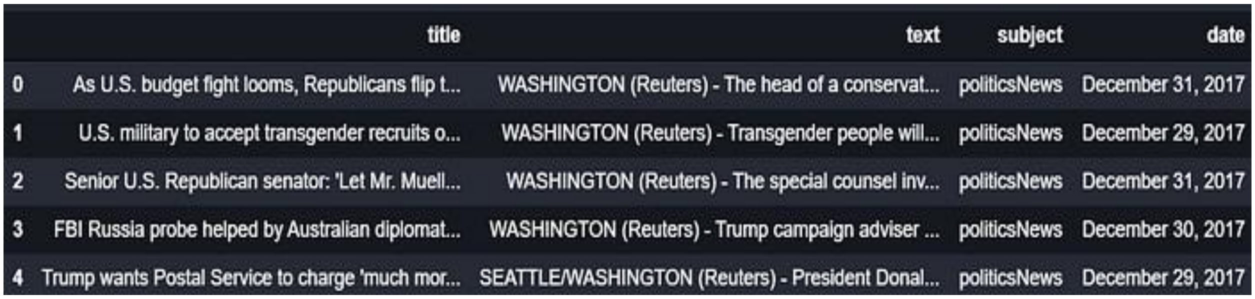
```
import pandas as pd

# Load dataset from a CSV file
df = pd.read_csv('fake_news_dataset.csv')
# Display the first few rows of the dataset to inspect its structure
print(df.head())
```

Using Database Access Libraries:

If your data is stored in a database, you'll establish a connection to the database and then fetch the data:

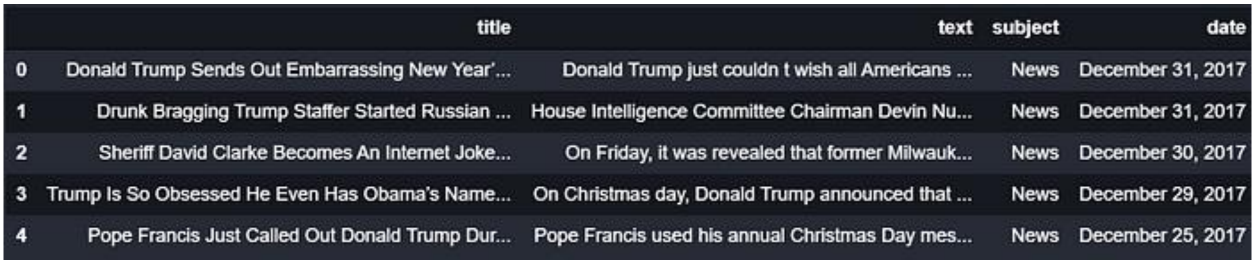
```
import psycopg2
# Establish a connection to the PostgreSQL
database conn = psycopg2.connect(database="your_db",
user="your_user", password="your_password",
host="your_host", port="your_port")
# Fetch data from the database
query = "SELECT * FROM your_table;" df =
pd.read_sql_query(query, conn)
# Close the database connection when done conn.close()
```

Load Dataset: real.head



	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

Fake.head



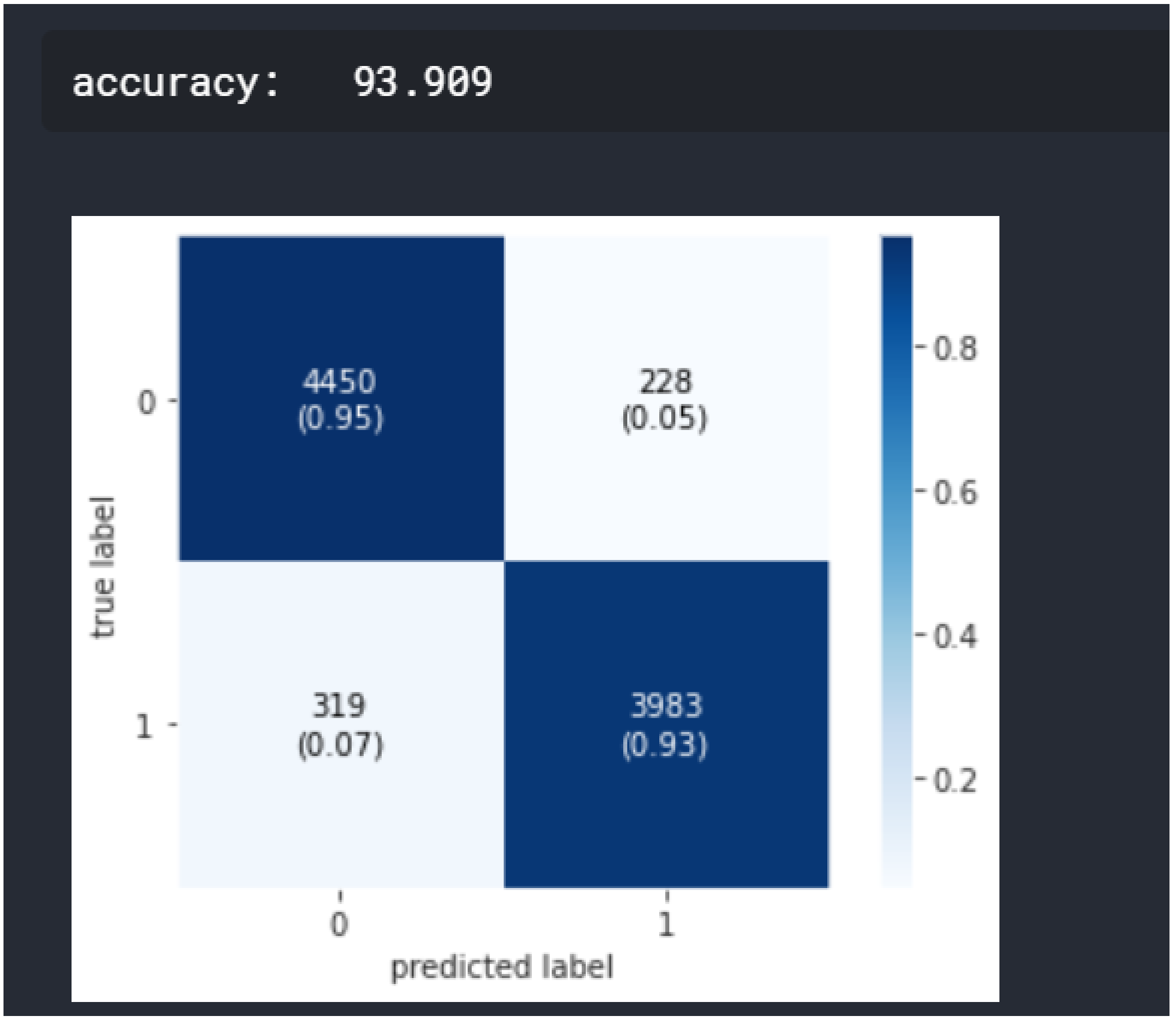
	title	text	subject	date
0	Donald Trump Sends Out Embarrassing New Year’...	Donald Trump just couldn t wish all Americans ...	News	December 31, 2017
1	Drunk Bragging Trump Staffer Started Russian ...	House Intelligence Committee Chairman Devin Nu...	News	December 31, 2017
2	Sheriff David Clarke Becomes 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

Program:

```
#Multinomial NB

from sklearn.feature_extraction.text
import TfidfTransformer
from sklearn.feature_extraction.text
import CountVectorizer
from sklearn.feature_extraction.text
import TfidfVectorizer
from sklearn.pipeline
import Pipeline
from sklearn.naive_bayes
import MultinomialNB
from sklearn.metrics
import accuracy_score
import sklearn.metrics as metrics
from mlxtend.plotting
import plot_confusion_matrix
from sklearn.metrics
import confusion_matrix

pipe = Pipeline([
    ('vect', CountVectorizer()),
    ('tfidf', TfidfTransformer()),
    ('clf', MultinomialNB())
])

model = pipe.fit(x_train, y_train)
prediction = model.predict(x_test)
```

```
score = metrics.accuracy_score(y_test, prediction)
print("accuracy:   %0.3f" % (score*100))
cm = metrics.confusion_matrix(y_test, prediction,
labels=[0,1])

fig, ax =
plot_confusion_matrix(conf_mat=confusion_matrix(y_test,
prediction),show_absolute=True,
                                 show_normed=True,
                                 colorbar=True)
plt.show()
```

<u>ouput:</u>



```
accuracy:   93.909
```

Preprocessing Data:

- Preprocessing data in the context of fake news detection refers to the series of data cleaning and transformation steps performed on the text data to prepare it for analysis or machine learning.

- These preprocessing steps are essential to ensure that the text data is in a suitable format for training and evaluating fake news detection models.

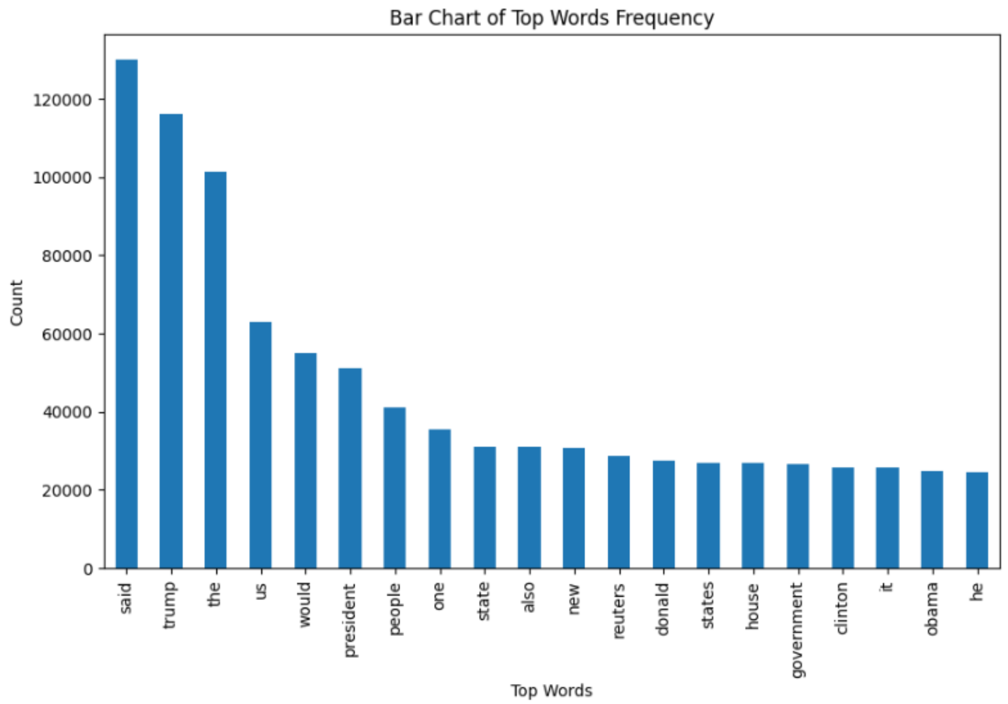
- Text Cleaning: Removing any noise, special characters, punctuation, or symbols that do not contribute to the analysis. This step helps ensure that the text data is as clean as possible.
- Tokenization: Splitting the text into individual words or tokens. Tokenization is a fundamental step for analyzing text data on a word-level basis.
- Lowercasing: Converting all text to lowercase to ensure consistency and prevent the model from treating the same word in different cases as different words.
- Stopword Removal: Eliminating common words (stopwords) such as "the," "and," "in," which are frequently used in text but typically do not contain much information for fake news detection.

Program:

```
from sklearn.feature_extraction.text import CountVectorizer


def get_top_n_words(corpus, n=None):
    vec = CountVectorizer().fit(corpus)
    bag_of_words = vec.transform(corpus)
    sum_words = bag_of_words.sum(axis=0)
    words_freq = [(word, sum_words[0, idx])
                  for word, idx in vec.vocabulary_.items()]
    words_freq = sorted(words_freq, key=lambda x: x[1],
                        reverse=True)
    return words_freq[:n]
```

```
common_words = get_top_n_words(data['text'], 20)
df1 = pd.DataFrame(common_words, columns=['Review',
'count'])

df1.groupby('Review').sum()['count'].sort_values(ascending=F
alse).plot(
    kind='bar',
    figsize=(10, 6),
    xlabel="Top Words",
    ylabel="Count",
    title="Bar Chart of Top Words Frequency"
)
```

Output:



Code:

```
from sklearn.feature_extraction.text import TfidfVectorizer

vectorization = TfidfVectorizer()
x_train = vectorization.fit_transform(x_train)
```

```
x_test = vectorization.transform(x_test)
```

Model training, Evaluation, and Prediction:

Now, the dataset is ready to train the model.

For training we will use Logistic Regression and evaluate the prediction accuracy using accuracy_score.

Program:

```
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
model.fit(x_train, y_train)
# testing the model
print(accuracy_score(y_train,
model.predict(x_train)))
print(accuracy_score(y_test, model.predict(x_test)))
```

Output :

```
0.993766511324171
0.9893143365983972
```

Let's train with Decision Tree Classifier.

Program:

```
from sklearn.tree import DecisionTreeClassifier
model = DecisionTreeClassifier()
model.fit(x_train, y_train)
```

```
# testing the model
print(accuracy_score(y_train,
model.predict(x_train)))
print(accuracy_score(y_test, model.predict(x_test)))
```

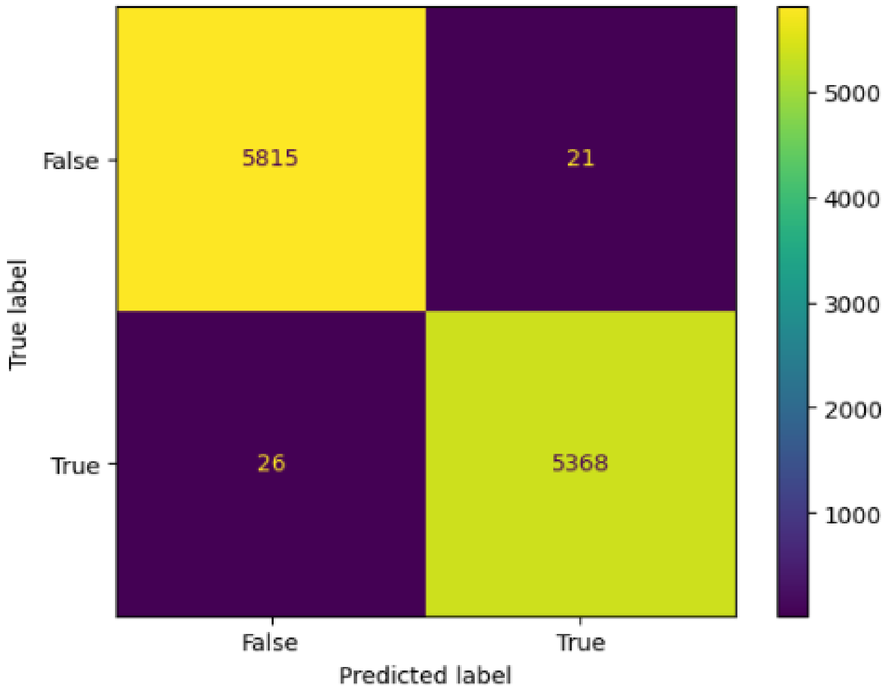
Output :

```
0.9999703167205913
0.9951914514692787
```

The confusion matrix for Decision Tree Classifier can be implemented with the code below.

```
# Confusion matrix of Results from Decision Tree
classification

from sklearn import metrics
cm = metrics.confusion_matrix(y_test,
model.predict(x_test))
cm_display =
metrics.ConfusionMatrixDisplay(confusion_matrix=cm,
display_labels=[False, True])
cm_display.plot()
plt.show()
```

Output:



Conclusion:

- *Data preprocessing is a crucial step in fake news detection, as it significantly impacts the quality and effectiveness of the model.*
- *Preprocessing prepares the text data for analysis and machine learning by cleaning, standardizing, and transforming it.*
- *Lemmatization or Stemming: Reducing words to their base or root form helps standardize words, reduce dimensionality, and group similar words together.*

.

- *Numerical Value Removal: Removing numerical values, which are often irrelevant to the task of fake news detection.*
- *Handling Missing Data: Addressing missing data issues, ensuring that the dataset is complete and reliable.*