Load and Preprocess Dataset for Fake News Detection

Introduction:

- Provide a brief overview of the project, including the goals and objectives.
- Discuss the importance of fake news detection and the role that machine learning can play in this task.
- Introduce the dataset that will be used in the project.

Data Loading:

- Describe how the dataset was loaded into the Python environment.
- Explain any data cleaning steps that were performed.

Text Preprocessing:

- Describe how the text in each news article was preprocessed.
- Explain the rationale for each preprocessing step.

Vectorization:

- Describe how the text in each news article was converted to a vector of word frequencies.
- Explain the rationale for using the bag-of-words approach.
- Splitting the Dataset into Training and Testing Sets
- Describe how the dataset was split into training and testing sets.
- Explain the importance of using a separate testing set to evaluate the model.

```
import  numpy as np
import pandas as pd
import  itertools
from sklearn.model_selection import train_test_split
from sklearn.linear_model import PassiveAggressiveClassifier
from sklearn.metrics import accuracy_score, confusion_matrix
import string as st
import re
import nltk
from nltk import PorterStemmer, WordNetLemmatizer
import matplotlib.pyplot as plt

import os
for dirname, _, filenames in os.walk('input of fake data set'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

The notebook accomplishes these tasks -

- Text cleaning and preprocessing of fake_or_real_news dataset using NLTK and Regex library.
- Creating and transforming clean text into tf-idf vectors.
- Learning models like Passive Aggressive Classifier, XGBoost and LGBM to perform classification of fake and real news. (A few other algorithms were also tried but only best three out of those are chosen.)
- Evaluate each model's performance based on the accuracy scores and confusion matrices they produced.

The following text preprocessing steps are performed here -

- Remove punctuations
- Convert text to tokens
- Remove tokens of length less than or equal to 3
- Remove stopwords using NLTK corpus stopwords list to match
- Apply lemmatization

In [1]:

```
# This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python Docker image: https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import itertools
from sklearn.model_selection import train_test_split
from sklearn.linear_model import PassiveAggressiveClassifier
from sklearn.metrics import accuracy_score, confusion_matrix
import string as st
import re
import nltk
from nltk import PorterStemmer, WordNetLemmatizer
import matplotlib.pyplot as plt

# 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))

# You can write up to 5GB to the current directory (/kaggle/working/) that gets preserved as output when you create a version using "Save & Run All"
# You can also write temporary files to /kaggle/temp/, but they won't be saved outside of the current session
```

```
/kaggle/input/textdb3/fake_or_real_news.csv
```

```
In [2]:
data = pd.read_csv('../input/textdb3/fake_or_real_news.csv')
data.shape
```

```
Out[2]:
(6335, 4)
```

```
In [3]:
data.head()
```

Out[3]:

	Unnamed: 0	title	text	label
0	8476	You Can Smell Hillary's Fear	Daniel Greenfield, a Shillman Journalism Fello...	FAKE
1	10294	Watch The Exact Moment Paul Ryan Committed Pol...	Google Pinterest Digg Linkedin Reddit Stumbleu...	FAKE
2	3608	Kerry to go to Paris in gesture of sympathy	U.S. Secretary of State John F. Kerry said Mon...	REAL
3	10142	Bernie supporters on Twitter erupt in anger ag...	— Kaydee King (@KaydeeKing) November 9, 2016 T...	FAKE
4	875	The Battle of New York: Why This Primary Matters	It's primary day in New York and front-runners...	REAL

```
In [4]:
# Check how the labels are distributed
print(np.unique(data['label']))
print(np.unique(data['label'].value_counts()))
```

```
['FAKE' 'REAL']
[3164 3171]
```

Text cleaning and processing steps-

- Remove punctuations
- Convert text to tokens
- Remove tokens of length less than or equal to 3
- Remove stopwords using NLTK corpus stopwords list to match
- Apply lemmatization
- Convert words to feature vectors

In [5]:

```
# Remove all punctuations from the text

def remove_punct(text):
    return ("".join([ch for ch in text if ch not in st.punctuation]))
```

In [6]:

```
data['removed_punc'] = data['text'].apply(lambda x: remove_punct(x))
data.head()
```

Out[6]:

	Unnamed: 0	title	text	label	removed_punc
0	8476	You Can Smell Hillary's Fear	Daniel Greenfield, a Shillman Journalism Fello...	FAKE	Daniel Greenfield a Shillman Journalism Fellow...
1	10294	Watch The Exact Moment Paul Ryan Committed Pol...	Google Pinterest Digg Linkedin Reddit Stumbleu...	FAKE	Google Pinterest Digg Linkedin Reddit Stumbleu...
2	3608	Kerry to go to Paris in gesture of sympathy	U.S. Secretary of State John F. Kerry said Mon...	REAL	US Secretary of State John F Kerry said Monday...
3	10142	Bernie supporters on Twitter erupt in anger ag...	— Kaydee King (@KaydeeKing) November 9, 2016 T...	FAKE	— Kaydee King KaydeeKing November 9 2016 The l...
4	875	The Battle of New York: Why This Primary Matters	It's primary day in New York and front-runners...	REAL	Its primary day in New York and frontrunners H...

In [7]:

```
''' Convert text to lower case tokens. Here, split() is applied on white-spaces. But, it could be applied
    on special characters, tabs or any other string based on which text is to be seperated into tokens.
'''
def tokenize(text):
    text = re.split('\s+' ,text)
    return [x.lower() for x in text]
```

In [8]:

```
data['tokens'] = data['removed_punc'].apply(lambda msg : tokenize(msg))
data.head()
```

Out[8]:

	Unnamed: 0	title	text	label	removed_punc	tokens
0	8476	You Can Smell Hillary's Fear	Daniel Greenfield, a Shillman Journalism Fello...	FAKE	Daniel Greenfield a Shillman Journalism Fellow...	[daniel, greenfield, a, shillman, journalism, ...
1	10294	Watch The Exact Moment Paul Ryan Committed Pol...	Google Pinterest Digg Linkedin Reddit Stumbleu...	FAKE	Google Pinterest Digg Linkedin Reddit Stumbleu...	[google, pinterest, digg, linkedin, reddit, st...
2	3608	Kerry to go to Paris in gesture of sympathy	U.S. Secretary of State John F. Kerry said Mon...	REAL	US Secretary of State John F Kerry said Monday...	[us, secretary, of, state, john, f, kerry, sai...
3	10142	Bernie supporters on Twitter erupt in anger ag...	— Kaydee King (@KaydeeKing) November 9, 2016 T...	FAKE	— Kaydee King KaydeeKing November 9 2016 The l...	[—, kaydee, king, kaydeeking, november, 9, 201...
4	875	The Battle of New York: Why This Primary Matters	It's primary day in New York and front-runners...	REAL	Its primary day in New York and frontrunners H...	[its, primary, day, in, new, york, and, frontr...

In [9]:
```
# Remove tokens of length less than 3
def remove_small_words(text):
    return [x for x in text if len(x) > 3 ]
```

In [10]:
```
data['filtered_tokens'] = data['tokens'].apply(lambda x : remove_small_words(x))
data.head()
```

Out[10]:

	Unnamed: 0	title	text	label	removed_punc	tokens	filtered_tokens
0	8476	You Can Smell	Daniel Greenfield, a Shillman	FAKE	Daniel Greenfield a Shillman Journalism	[daniel, greenfield, a, shillman,	[daniel, greenfield, shillman,

	Unnamed: 0	title	text	label	removed_punc	tokens	filtered_tokens
		Hillary’s Fear	Journalism Fello...		Fellow...	journalism, ...	journalism, fel...
1	10294	Watch The Exact Moment Paul Ryan Committed Pol...	Google Pinterest Digg Linkedin Reddit Stumbleu...	FAKE	Google Pinterest Digg Linkedin Reddit Stumbleu...	[google, pinterest, digg, linkedin, reddit, st...	[google, pinterest, digg, linkedin, reddit, st...
2	3608	Kerry to go to Paris in gesture of sympathy	U.S. Secretary of State John F. Kerry said Mon...	REAL	US Secretary of State John F Kerry said Monday...	[us, secretary, of, state, john, f, kerry, sai...	[secretary, state, john, kerry, said, monday, ...
3	10142	Bernie supporters on Twitter erupt in anger ag...	— Kaydee King (@KaydeeKing) November 9, 2016 T...	FAKE	— Kaydee King KaydeeKing November 9 2016 The l...	[—, kaydee, king, kaydeeking, november, 9, 201...	[kaydee, king, kaydeeking, november, 2016, les...
4	875	The Battle of New York: Why This Primary Matters	It's primary day in New York and front-runners...	REAL	Its primary day in New York and frontrunners H...	[its, primary, day, in, new, york, and, frontr...	[primary, york, frontrunners, hillary, clinton...

In [11]:

```
''' Remove stopwords. Here, NLTK corpus list is used for a match. However, a customized user-defined
    list could be created and used to limit the matches in input text.
'''
def remove_stopwords(text):
    return [word for word in text if word not in
nltk.corpus.stopwords.words('english')]
```

In [12]:

```
data['clean_tokens'] = data['filtered_tokens'].apply(lambda x : remove_stopwords(x))
data.head()
```

Out[12]:

	Unnamed: 0	title	text	label	removed_punc	tokens	filtered_tokens	clean_tokens
0	8476	You Can Smell Hillary's Fear	Daniel Greenfield, a Shillman Journalism Fello...	FAKE	Daniel Greenfield a Shillman Journalism Fellow...	[daniel, greenfield, a, shillman, journalism, ...	[daniel, greenfield, shillman, journalism, fel...	[daniel, greenfield, shillman, journalism, fel...
1	10294	Watch The Exact Moment Paul Ryan Committed Pol...	Google Pinterest Digg Linkedin Reddit Stumbleu...	FAKE	Google Pinterest Digg Linkedin Reddit Stumbleu...	[google, pinterest, digg, linkedin, reddit, st...	[google, pinterest, digg, linkedin, reddit, st...	[google, pinterest, digg, linkedin, reddit, st...
2	3608	Kerry to go to Paris in gesture of sympathy	U.S. Secretary of State John F. Kerry said Mon...	REAL	US Secretary of State John F Kerry said Monday...	[us, secretary, of, state, john, f, kerry, sai...	[secretary, state, john, kerry, said, monday, ...	[secretary, state, john, kerry, said, monday, ...
3	10142	Bernie supporters on Twitter erupt in anger ag...	— Kaydee King (@KaydeeKing) November 9, 2016 T...	FAKE	— Kaydee King KaydeeKing November 9 2016 The l...	[—, kaydee, king, kaydeeking, november, 9, 201...	[kaydee, king, kaydeeking, november, 2016, les...	[kaydee, king, kaydeeking, november, 2016, les...
4	875	The Battle of New York: Why This Primary Matters	It's primary day in New York and front-runners...	REAL	Its primary day in New York and frontrunners H...	[its, primary, day, in, new, york, and, frontr...	[primary, york, frontrunners, hillary, clinton...	[primary, york, frontrunners, hillary, clinton...

In [13]:
```
# Apply lemmatization on tokens
def lemmatize(text):
    word_net = WordNetLemmatizer()
    return [word_net.lemmatize(word) for word in text]
```

In [14]:
```
data['lemma_words'] = data['clean_tokens'].apply(lambda x : lemmatize(x))
data.head()
```

Out[14]:

	Unnamed: 0	title	text	label	removed_punc	tokens	filtered_tokens	clean_tokens	lemma_words
0	8476	You Can Smell Hillary's Fear	Daniel Greenfield, a Shillman Journalism Fello...	FAKE	Daniel Greenfield a Shillman Journalism Fellow...	[daniel, greenfield, a, shillman, journalism, ...	[daniel, greenfield, shillman, journalism, fel...	[daniel, greenfield, shillman, journalism, fel...	[daniel, greenfield, shillman, journalism, fel...
1	10294	Watch The Exact Moment Paul Ryan Committed Pol...	Google Pinterest Digg Linkedin Reddit Stumbleu...	FAKE	Google Pinterest Digg Linkedin Reddit Stumbleu...	[google, pinterest, digg, linkedin, reddit, st...	[google, pinterest, digg, linkedin, reddit, st...	[google, pinterest, digg, linkedin, reddit, st...	[google, pinterest, digg, linkedin, reddit, st...
2	3608	Kerry to go to Paris in gesture of sympathy	U.S. Secretary of State John F. Kerry said Mon...	REAL	US Secretary of State John F Kerry said Monday...	[us, secretary, of, state, john, f, kerry, sai...	[secretary, state, john, kerry, said, monday, ...	[secretary, state, john, kerry, said, monday, ...	[secretary, state, john, kerry, said, monday, ...
3	10142	Bernie supporters on Twitter erupt in anger ag...	— Kaydee King (@KaydeeKing) November 9, 2016 T...	FAKE	— Kaydee King KaydeeKing November 9 2016 The l...	[—, kaydee, king, kaydeeking, november, 9, 201...	[kaydee, king, kaydeeking, november, 2016, les...	[kaydee, king, kaydeeking, november, 2016, les...	[kaydee, king, kaydeeking, november, 2016, les...
4	875	The Battle of New York: Why This Primary Matters	It's primary day in New York and front-runners...	REAL	Its primary day in New York and frontrunners H...	[its, primary, day, in, new, york, and, frontr...	[primary, york, frontrunners, hillary, clinton...	[primary, york, frontrunners, hillary, clinton...	[primary, york, frontrunners, hillary, clinton...

In [15]:

```
# Create sentences to get clean text as input for vectors

def return_sentences(tokens):
    return " ".join([word for word in tokens])
```

In [16]:

```
data['clean_text'] = data['lemma_words'].apply(lambda x : return_sentences(x))
data.head()
```

```
import pandas as pd
import numpy as np
import re
import nltk
df = pd.read_csv('fake_news_dataset.csv')
df.head()
df.shape
df.columns
df.dtypes
df['news'] = df['news'].apply(lambda x: re.sub(r'[^\w\s]', '', x))
df['news'] = df['news'].apply(lambda x: x.lower())
stopwords = nltk.corpus.stopwords.words('english')
df['news'] = df['news'].apply(lambda x: ' '.join([word for word in x.split() if
word not in stopwords]))
df['news'] = df['news'].apply(lambda x: ' '.join([word for word in x.split() if
len(word) > 2]))
lemmatizer = nltk.WordNetLemmatizer()
df['news'] = df['news'].apply(lambda x: ' '.join([lemmatizer.lemmatize(word) for
word in x.split()]))
df.to_csv('fake_news_preprocessed.csv', index=False)
```

Out[16]:

	Unnamed: 0	title	text	label	removed_punc	tokens	filtered_tokens	clean_tokens	lemma_words	clean_text
0	8476	You Can Smell Hillary's Fear	Daniel Greenfield, a Shillman Journalism Fello...	FAKE	Daniel Greenfield a Shillman Journalism Fellow...	[daniel, greenfield, a, shillman, journalism, ...	[daniel, greenfield, shillman, journalism, fel...	[daniel, greenfield, shillman, journalism, fel...	[daniel, greenfield, shillman, journalism, fel...	daniel greenfield shillman journalism fellow f...
1	10294	Watch The Exact Moment Paul Ryan Committed Pol...	Google Pinterest Digg Linkedin Reddit Stumbleu...	FAKE	Google Pinterest Digg Linkedin Reddit Stumbleu...	[google, pinterest, digg, linkedin, reddit, st...	[google, pinterest, digg, linkedin, reddit, st...	[google, pinterest, digg, linkedin, reddit, st...	[google, pinterest, digg, linkedin, reddit, st...	google pinterest digg linkedin reddit stumbleu...
2	3608	Kerry to go to Paris in gesture of sympathy	U.S. Secretary of State John F. Kerry said Mon...	REAL	US Secretary of State John F Kerry said Monday...	[us, secretary, of, state, john, f, kerry, sai...	[secretary, state, john, kerry, said, monday, ...	[secretary, state, john, kerry, said, monday, ...	[secretary, state, john, kerry, said, monday, ...	secretary state john kerry said monday stop pa...
3	10142	Bernie supporters on Twitter erupt in anger ag...	— Kaydee King (@KaydeeKing) November 9, 2016 T...	FAKE	— Kaydee King KaydeeKing November 9 2016 The l...	[—, kaydee, king, kaydeeking, novembe r, 9, 201...	[kaydee, king, kaydeeking, november, 2016, les...	[kaydee, king, kaydeeking, novembe r, 2016, les...	[kaydee, king, kaydeeking, november, 2016, les...	kaydee king kaydeeking novembe r 2016 lesson to...
4	875	The Battle of New York: Why This Primary	It's primary day in New York and front-runners...	REAL	Its primary day in New York and frontrunners H...	[its, primary, day, in, new, york, and,	[primary, york, frontrunners, hillary, clinton...	[primary, york, frontrunners, hillary, clinton...	[primary, york, frontrunners, hillary, clinton...	primary york frontrunners hillary clinton

	Unnamed: 0	title	text	label	removed_punc	tokens	filtered_tokens	clean_tokens	lemma_words	clean_text
		Matters				frontr...				dona...

In [17]:
```
# Generate a basic word cloud
from wordcloud import WordCloud, ImageColorGenerator

text = " ".join([x for x in data['clean_text']])
# Create and generate a word cloud image:
wordcloud = WordCloud(max_font_size=30, max_words=1000).generate(text)

# Display the generated image:
plt.figure(figsize= [20,10])
plt.imshow(wordcloud)
plt.axis("off")
plt.show()
```

In [18]:
```
# Prepare data for the model. Convert label in to binary

data['label'] = [1 if x == 'FAKE' else 0 for x in data['label']]
data.head()
```

Out[18]:

	Unnamed: 0	title	text	label	removed_punc	tokens	filtered_tokens	clean_tokens	lemma_words	clean_text
0	8476	You Can Smell Hillary's Fear	Daniel Greenfield, a Shillman Journalism Fello...	1	Daniel Greenfield a Shillman Journalism Fellow...	[daniel, greenfield, a, shillman, journalism, ...	[daniel, greenfield, shillman, journalism, fel...	[daniel, greenfield, shillman, journalism, fel...	[daniel, greenfield, shillman, journalism, fel...	daniel greenfield shillman journalism fellow f...
1	10294	Watch The Exact Moment Paul Ryan Committed	Google Pinterest Digg Linkedin Reddit Stumbleu...	1	Google Pinterest Digg Linkedin Reddit Stumbleu...	[google, pinterest, digg, linkedin, reddit, st...	[google, pinterest, digg, linkedin, reddit, st...	[google, pinterest, digg, linkedin, reddit, st...	[google, pinterest, digg, linkedin, reddit, st...	google pinterest digg linkedin reddit stumbleu...

	Unnamed: 0	title	text	label	removed_punc	tokens	filtered_tokens	clean_tokens	lemma_words	clean_text
		Pol...								
2	3608	Kerry to go to Paris in gesture of sympathy	U.S. Secretary of State John F. Kerry said Mon...	0	US Secretary of State John F Kerry said Monday...	[us, secretary, of, state, john, f, kerry, sai...	[secretary, state, john, kerry, said, monday, ...	[secretary, state, john, kerry, said, monday, ...	[secretary, state, john, kerry, said, monday, ...	secretary state john kerry said monday stop pa...
3	10142	Bernie supporters on Twitter erupt in anger ag...	— Kaydee King (@KaydeeKing) November 9, 2016 T...	1	— Kaydee King KaydeeKing November 9 2016 The l...	[—, kaydee, king, kaydeeking, novembe r, 9, 201...	[kaydee, king, kaydeeking, november, 2016, les...	[kaydee, king, kaydeeking, november, 2016, les...	[kaydee, king, kaydeeking, november, 2016, les...	kaydee king kaydeeking novembe r 2016 lesson to...
4	875	The Battle of New York: Why This Primary Matters	It's primary day in New York and front-runners...	0	Its primary day in New York and frontrunners H...	[its, primary, day, in, new, york, and, frontr...	[primary, york, frontrunners, hillary, clinton...	[primary, york, frontrunners, hillary, clinton...	[primary, york, frontrunners, hillary, clinton...	primary york frontrunners hillary clinton dona...

In [19]:
```
# Split the dataset

X_train,X_test,y_train,y_test = train_test_split(data['clean_text'], data['label'], test_size=0.2, random_state = 5)

print(X_train.shape)
print(X_test.shape)
```

```
(5068,)
(1267,)
```

TF-IDF : Term Frequency - Inverse Document Frequency

The term frequency is the number of times a term occurs in a document. Inverse document frequency is an inverse function of the number of documents in which that a given word occurs.

The product of these two terms gives tf-idf weight for a word in the corpus. The higher the frequency of occurrence of a word, lower is it's weight and vice-versa. This gives more weightage to rare terms in the corpus and penalizes more commonly occuring terms.

Other widely used vectorizer is Count vectorizer which only considers the frequency of occurrence of a word across the corpus.

```
In [20]:
from sklearn.feature_extraction.text import TfidfVectorizer

tfidf = TfidfVectorizer()
tfidf_train = tfidf.fit_transform(X_train)
tfidf_test = tfidf.transform(X_test)

print(tfidf_train.toarray())
print(tfidf_train.shape)
print(tfidf_test.toarray())
print(tfidf_test.shape)
```

```
[[0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 ...
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]]
(5068, 68134)
[[0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 ...
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]]
(1267, 68134)
```

Passive Aggressive Classifiers

- Passive Aggressive algorithms are online learning algorithms. Such an algorithm remains passive for a correct classification outcome, and turns aggressive in the event of a miscalculation, updating and adjusting.
- Passive: If the prediction is correct, keep the model and do not make any changes. i.e., the data in the example is not enough to cause any changes in the model.
- Aggressive: If the prediction is incorrect, make changes to the model. i.e., some change to the model may correct it.

These are typically used for large datasets where batch learning is not possible due to huge volumes of frequently incoming data.

- Some important parameters -
- C : This is the regularization parameter, and denotes the penalization the model will make on an incorrect prediction
- max_iter : The maximum number of iterations the model makes over the training data.
- tol : The stopping criterion.

In [21]:
```
# Passive Aggresive Classifier
pac = PassiveAggressiveClassifier(max_iter=50)
pac.fit(tfidf_train,y_train)

pred = pac.predict(tfidf_test)
print("Accuracy score : {}".format(accuracy_score(y_test, pred)))
print("Confusion matrix : \n {}".format(confusion_matrix(y_test, pred)))
```

```
Accuracy score : 0.936069455406472
Confusion matrix :
 [[592  38]
 [ 43 594]]
```

In [22]:
```
# Logistic Regression model
from sklearn.linear_model import LogisticRegression

lr = LogisticRegression(max_iter = 500)
lr.fit(tfidf_train, y_train)
print('Logistic Regression model fitted..')

pred = lr.predict(tfidf_test)
print("Accuracy score : {}".format(accuracy_score(y_test, pred)))
print("Confusion matrix : \n {}".format(confusion_matrix(y_test, pred)))
```

```
Logistic Regression model fitted..
Accuracy score : 0.9179163378058406
Confusion matrix :
 [[565  65]
 [ 39 598]]
```

Logistic Regression could not outperform XGBoost and LGBM but its performance is considerably close to them and it is much less complex.

In [23]:
```
import xgboost
from xgboost import XGBClassifier

xgb = XGBClassifier()
xgb.fit(tfidf_train, y_train)

print('XGBoost Classifier model fitted..')
pred = xgb.predict(tfidf_test)
print("Accuracy score : {}".format(accuracy_score(y_test, pred)))
print("Confusion matrix : \n {}".format(confusion_matrix(y_test, pred)))
```

```
XGBoost Classifier model fitted..
Accuracy score : 0.9289660615627466
Confusion matrix :
 [[587  43]
 [ 47 590]]
```

In [24]:
```
import lightgbm
from lightgbm import LGBMClassifier

lgbm = LGBMClassifier()
```

```
lgbm.fit(tfidf_train, y_train)

print('LightGBM Classifier model fitted..')
pred = lgbm.predict(tfidf_test)
print("Accuracy score : {}".format(accuracy_score(y_test, pred)))
print("Confusion matrix : \n {}".format(confusion_matrix(y_test, pred)))
```

```
LightGBM Classifier model fitted..
Accuracy score : 0.9289660615627466
Confusion matrix :
 [[581  49]
 [ 41 596]]
```

```
import pandas as pd

df = pd.read_csv('fake_or_real_news.csv')
```

Text Cleaning:

The text in each news article was cleaned by removing punctuation, stop words, and other irrelevant characters. Punctuation was removed using the string. punctuation module from the Python standard library. Stop words were removed using the nltk .corpus . stop words .words() function from the Natural Language Toolkit (NLTK) library.

```
import string

import nltk

text = text.translate(str.maketrans('', '', string.punctuation))

stopwords = set(nltk.corpus.stopwords.words('english'))

text = ' '.join([word for word in text.split() if word not in stopwords])
```

Converting Text to Vectors:

The text in each news article was converted to a vector of word frequencies using the bag-of-words approach.

The bag-of-words approach is a simple but effective way to represent text data for machine learning tasks. It works by converting each text document into a vector of word frequencies, where each element in the vector represents the number of times a particular word appears in the document.

```
 import CountVectorizer

vectorizer = CountVectorizer()

X = vectorizer.fit_transform(df['text'])
```

The X variable is a NumPy matrix containing the word frequencies for each news article.

Splitting the Dataset into Training and Testing Sets

The dataset was split into training and testing sets using the train_test_split function from the scikit-learn library.

The train_test_split function takes two arguments: the data to be split and the desired proportion of training and testing data.

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, df['label'], test_size=0.25)
```

The X_train and X_test variables contain the word frequencies for the training and testing sets, respectively. The y_train and y_test variables contain the labels for the training and testing sets, respectively.

```
import pandas as pd

import numpy as np

import re

import nltk

df = pd.read_csv('fake_news_dataset.csv')

df.head()

df.shape

df.columns

df.dtypes

df['news'] = df['news'].apply(lambda x: re.sub(r'[^\w\s]', '', x))

df['news'] = df['news'].apply(lambda x: x.lower())

stopwords = nltk.corpus.stopwords.words('english')

df['news'] = df['news'].apply(lambda x: ' '.join([word for word in x.split() if word not in stopwords]))

df['news'] = df['news'].apply(lambda x: ' '.join([word for word in x.split() if len(word) > 2]))

lemmatizer = nltk.WordNetLemmatizer()

df['news'] = df['news'].apply(lambda x: ' '.join([lemmatizer.lemmatize(word) for word in x.split()]))

df.to_csv('fake_news_preprocessed.csv', index=False)
```

Conclusion:

Summarize the main steps involved in loading and preprocessing the dataset.

Discuss any challenges that were faced and how they were overcome.

Suggest possible directions for future work.

This project report has described the process of loading and preprocessing a dataset for fake news detection. The dataset used in this project is the "Fake and real news dataset" available on which contains 50,000 news articles labeled as either real or fake.

In addition to the above sections, you may also want to include the following in your project report:

A brief literature review of related work on fake news detection.

A detailed description of the machine learning model that you plan to use for fake news detection.

A description of the evaluation metrics that you will use to assess the performance of the model.

Once you have written your project report, be sure to proofread it carefully and have someone else review it as well.