DETECTING FAKE NEWS WITH PYTHON AND MACHINE LEARNING

BY NILAVO BORAL

M.SC. DATA SCIENCE

Introduction:

Do you trust all the news you hear from social media?

All news are not real, right?

How will you detect fake news?

The answer is Python. By practicing this advanced python project of detecting fake news, you will easily make a difference between real and fake news. Before moving ahead in this machine learning project, get aware of the terms related to it like fake news, tfidfvectorizer, PassiveAggressive Classifier.

What is Fake News?

A type of yellow journalism, fake news encapsulates pieces of news that may be hoaxes and is generally spread through social media and other online media. This is often done to further or impose certain ideas and is often achieved with political agendas. Such news items may contain false and/or exaggerated claims, and may end up being viralized by algorithms, and users may end up in a filter bubble.

What is a TfidfVectorizer?

TF (Term Frequency) : The number of times a word appears in a document is its Term Frequency. A higher value means a term appears more often than others, and so, the document is a good match when the term is part of the search terms.

- **TF = (No. of repetitions of words in sentence) / (No. of words in sentence)**

Sentence 1: good boy

Sentence 2: good girl

Sentence 3: boy girl good

TF	Sentence 1	Sentence 2	Sentence 3
good	1/2	1/2	1/3
boy	1/2	0	1/3
girl	0	1/2	1/3

IDF (Inverse Document Frequency) : Words that occur many times a document, but also occur many times in many others, may be irrelevant. IDF is a measure of how significant a term is in the entire corpus.

- **IDF = log ((No. of sentences) / (No. of sentences containing the word))**

Sentence 1: good boy

Sentence 2: good girl

Sentence 3: boy girl good

Words	IDF
good	log(3/3) = 0
boy	log(3/2)
girl	log(3/2)

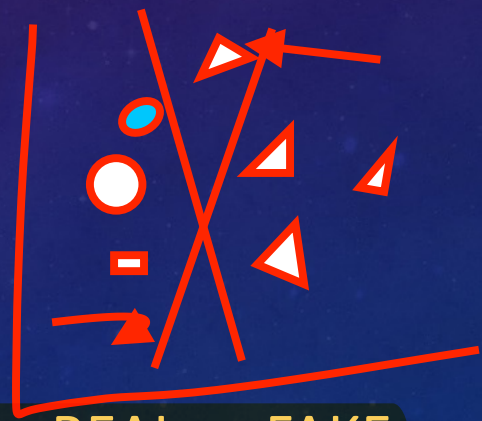
TF-IDA = TF * IDA

TF-IDA	Sentence 1	Sentence 2	Sentence 3
good	0	0	0
boy	(log(3/2)) / 2	0	(log(3/2)) / 3
girl	0	(log(3/2)) / 2	(log(3/2)) / 3

The TfidfVectorizer converts a collection of raw documents into a matrix of TF-IDF features.

What is a PassiveAggressiveClassifier?

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. Unlike most other algorithms, it does not converge. Its purpose is to make updates that correct the loss, causing very little change in the norm of the weight vector.



Detecting Fake News with Python :

To build a model to accurately classify a piece of news as REAL or FAKE.

About Detecting Fake News with Python :

This advanced python project of detecting fake news deals with fake and real news. Using sklearn, we build a TfidfVectorizer on our dataset. Then, we initialize a PassiveAggressive Classifier and fit the model. In the end, the accuracy score and the confusion matrix tell us how well our model fares.

Project :

1. Import libraries and dataset.

In [1]:
```
import numpy as np
import pandas as pd
import itertools
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import PassiveAggressiveClassifier
from sklearn.metrics import accuracy_score, confusion_matrix
```

In [2]:
```
df = pd.read_csv('/Users/nilavo/PPT/news.csv')
```

In [3]:
```
df.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

2. Split the dataset into training and testing sets.

```
In [11]:
#DataFlair - Split the dataset
x_train,x_test,y_train,y_test = train_test_split(df.text, df.label, test_size=0.2, random_state=7)
```

3. Now, fit and transform the vectorizer on the train set, and transform the vectorizer on the test set.

```
In [12]:
#DataFlair - Initialize a TfidfVectorizer
tfidf = TfidfVectorizer(stop_words='english', max_df=0.7)
```

```
In [13]:
#DataFlair - Fit and transform train set, transform test set
tfidf_train = tfidf.fit_transform(x_train)
tfidf_test = tfidf.transform(x_test)
```

```
In [14]:
tfidf_train
```

```
Out[14]: <5068x61651 sparse matrix of type '<class 'numpy.float64'>'
        with 1337098 stored elements in Compressed Sparse Row format>
```

```
In [15]:
tfidf_test
```

```
Out[15]: <1267x61651 sparse matrix of type '<class 'numpy.float64'>'
        with 322056 stored elements in Compressed Sparse Row format>
```

4. Next, we'll initialize a PassiveAggressiveClassifier. This is. We'll fit this on tfidf_train and y_train.
Then, we'll predict on the test set from the TfidfVectorizer and calculate the accuracy with accuracy_score() from sklearn.metrics.

In [16]:
```
#DataFlair - Initialize a PassiveAggressiveClassifier
pac = PassiveAggressiveClassifier(max_iter=50)
pac.fit(tfidf_train,y_train)

#DataFlair - Predict on the test set
y_pred = pac.predict(tfidf_test)
```

In [17]:
```
y_pred
```

Out[17]:
```
array(['REAL', 'FAKE', 'REAL', ..., 'REAL', 'FAKE', 'REAL'], dtype='<U4')
```

In [18]:
```
#DataFlair - Build confusion matrix
confusion_matrix(y_test,y_pred, labels=['FAKE','REAL'])
```

Out[18]:
```
array([[586,  52],
       [ 46, 583]])
```

In [19]:
```
#calculate accuracy
score = accuracy_score(y_test,y_pred)
print(f'Accuracy: {round(score*100,2)}%')
```

```
Accuracy: 92.27%
```

We got an accuracy of 92.27% with this model.

Thank You.