

# Fake News Detection

## 1 DETECTING FAKE NEWS WITH PYTHON AND MACHINE LEARNING

### 1.1 What is Fake News?

The internet and social media have made it very easy for anyone to publish content on a website, blog or social media profile and potentially reach large audiences. With so many people now getting news from social media sites, many content creators/publishers have used this to their advantage. Lots of things we read online especially in your social media feeds may appear to be true, often is not.

**False news are stories or hoaxes created to deliberately misinform or deceive readers.**

## What if 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.

**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.

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

### 1.2 What is Passive-Aggressive Classifier ?

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. These are generally used for large-scale learning. In this the input data comes in sequential order and the machine learning model is updated step-by-step, as opposed to batch learning, where the entire training dataset is used at once.

### 1.3 Detecting Fake News with Python

To build a model using Passive-Aggressive Classifier to accurately classifies pieces of news as Real and Fake

## 1.4 The Fake News Dataset

The dataset used in this project has a shape of 7796×4. The first column identifies the news, the second and third are the title and text, and the fourth column has labels denoting whether the news is REAL or FAKE.

## 1.5 Steps for detecting fake news:-

### 1. Importing necessary libraries

```
[2]: # Importing Libraries to be used
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
```

### 2. Reading data into dataframe and getting shape of dataset , first 5 records and labels from dataframe

```
[3]: # Reading the inputs
filpath= "C:/Users/HP/Desktop/news.csv"
df=pd.read_csv(filpath)

# Print shape and head('First Five Examples')
print(df.shape)
print(df.head())

# Get the labels
labels=df.label
labels.head()
```

(6335, 4)

	Unnamed: 0	title \
0	8476	You Can Smell Hillary's Fear
1	10294	Watch The Exact Moment Paul Ryan Committed Pol...
2	3608	Kerry to go to Paris in gesture of sympathy
3	10142	Bernie supporters on Twitter erupt in anger ag...
4	875	The Battle of New York: Why This Primary Matters

	text	label
0	Daniel Greenfield, a Shillman Journalism Fello...	FAKE
1	Google Pinterest Digg Linkedin Reddit Stumbleu...	FAKE
2	U.S. Secretary of State John F. Kerry said Mon...	REAL
3	- Kaydee King (@KaydeeKing) November 9, 2016 T...	FAKE
4	It's primary day in New York and front-runners...	REAL

```
[3]: 0    FAKE
      1    FAKE
      2    REAL
      3    FAKE
      4    REAL
      Name: label, dtype: object
```

### 3. Splitting the dataset into training and testing sets

```
[4]: # Splitting the dataset
x_train,x_test,y_train,y_test=train_test_split(df['text'], labels, test_size=0.
→2, random_state=7)
x_train,x_test,y_train,y_test
```

```
[4]: (6237    The head of a leading survivalist group has ma...
      3722    < > Arnaldo Rodgers is a trained and educated ...
      5774    Patty Sanchez, 51, used to eat 13,000 calories...
      336    But Benjamin Netanyahu's reelection was regard...
      3622    John Kasich was killing it with these Iowa vot...
      ...
      5699
      2550    It's not that Americans won't elect wealthy pr...
      537    Anyone writing sentences like 'nevertheless fu...
      1220    More Catholics are in Congress than ever befor...
      4271    It was hosted by CNN, and the presentation was...
      Name: text, Length: 5068, dtype: object,
      3534    A day after the candidates squared off in a fi...
      6265    VIDEO : FBI SOURCES SAY INDICTMENT LIKELY FOR ...
      3123    It's debate season, where social media has bro...
      3940    Mitch McConnell has decided to wager the Repub...
      2856    Donald Trump, the actual Republican candidate ...
      ...
      4986    Washington (CNN) President Barack Obama announ...
      5789    The revival of middle-class jobs has been one ...
      4338    "I can guarantee that," Obama answered when as...
      5924    Videos 30 Civilians Die In US Airstrike Called...
      6030    The retired neurosurgeon lashed out Friday mor...
      Name: text, Length: 1267, dtype: object,
      6237    FAKE
      3722    FAKE
      5774    FAKE
      336    REAL
      3622    REAL
      ...
      5699    FAKE
      2550    REAL
      537    REAL
```

```

1220    REAL
4271    REAL
Name: label, Length: 5068, dtype: object,
3534    REAL
6265    FAKE
3123    REAL
3940    REAL
2856    REAL
...
4986    REAL
5789    REAL
4338    REAL
5924    FAKE
6030    REAL
Name: label, Length: 1267, dtype: object)

```

Initializing TfidfVectorizer with stop words from the English language and a maximum document frequency of 0.7 (terms with a higher document frequency will be discarded). Stop words are the most common words in a language that are to be filtered out before processing the natural language data. And a TfidfVectorizer turns a collection of raw documents into a matrix of TF-IDF features.

#### 4. Now let's fit and transform the vectorizer on the train set, and transform the vectorizer on the test set.

```

[5]: # Create Tfidf Vector
tfidf_vectorizer=TfidfVectorizer(stop_words='english', max_df=0.7)

# Transforming train and test set to Tfidf vectors
Tfidf_train = tfidf_vectorizer.fit_transform(x_train)
Tfidf_test = tfidf_vectorizer.transform(x_test)
Tfidf_train[0:2],Tfidf_test[0:2]

```

```

[5]: (<2x61651 sparse matrix of type '<class 'numpy.float64'>'
      with 271 stored elements in Compressed Sparse Row format>,
      <2x61651 sparse matrix of type '<class 'numpy.float64'>'
      with 247 stored elements in Compressed Sparse Row format>)

```

#### 5. Next, we'll initialize a PassiveAggressiveClassifier. 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.

```

[6]: # Creating passive aggressive classifier
model = PassiveAggressiveClassifier(max_iter=50)
model.fit(Tfidf_train,y_train)

# Predicting on Test set and calculating accuracy

```

```
y_pred=model.predict(TfIdf_test)
score=accuracy_score(y_test,y_pred)
print("Accuracy of model:-",score)
```

Accuracy of model:- 0.9234411996842936

Now we will do hyperparameter tuning for regularization parameter and number of iterations, by varying one hyperparameter and keeping the other constant and measuring for which value of hyperparameter the accuracy is maximum.

## 6. Tuning of regularization parameter

```
[7]: # Tuning regularization parameter

C = [0.1,0.2,0.5,0.9,1]
accuracy = []
for reg in C:
    model = PassiveAggressiveClassifier(max_iter=50 , C=reg)
    model.fit(TfIdf_train,y_train)

    y_pred=model.predict(TfIdf_test)
    score=accuracy_score(y_test,y_pred)
    accuracy.append(score)

max_accuracy = max(accuracy)
C_max = C[accuracy.index(max_accuracy)]

print("Accuracy is max for regularization:-" ,C_max,"And it is:-",max_accuracy)
```

Accuracy is max for regularization:- 0.9 And it is:- 0.930544593528019

We got accuracy maximum for regularization parameter 0.9 with accuracy 0.930544593528019 .

## 7. Tuning of max iterations hyperparameter

```
[9]: # Tuning max_iter parameter

Iteration = [10,50,100,200,500,1000]
accuracy = []
for epoch in Iteration:
    model = PassiveAggressiveClassifier(max_iter=epoch , C=C_max)
    model.fit(TfIdf_train,y_train)

    y_pred=model.predict(TfIdf_test)
    score=accuracy_score(y_test,y_pred)
    accuracy.append(score)

max_accuracy = max(accuracy)
iter_max = Iteration[accuracy.index(max_accuracy)]
```

```
accuracy
print("Accuracy is max for regularisation:-" ,iter_max,"and it is:
    ↳-",max_accuracy)
```

C:\Users\HP\anaconda3\envs\tensorflow-sessions\lib\site-packages\sklearn\linear\_model\\_stochastic\_gradient.py:577: ConvergenceWarning: Maximum number of iteration reached before convergence. Consider increasing max\_iter to improve the fit.  
ConvergenceWarning)

Accuracy is max for regularisation:- 500 and it is:- 0.9297553275453828

In this we got accuracy maximum for 500 iterations with accuracy of 0.9297553275453828 .

## 2 Final Model

**8. Now we will use the hyperpaameters value obtained during tuning and create the final model**

```
[10]: model_final = PassiveAggressiveClassifier(max_iter=iter_max , C=C_max)
model_final.fit(TfIdf_train,y_train)

y_pred=model_final.predict(TfIdf_test)
score_final=accuracy_score(y_test,y_pred)
score_final
```

```
[10]: 0.9297553275453828
```

**We got an accuracy of 0.929% with this model**

**9. Finally, we will print out a confusion matrix to gain insight into the number of false and true negatives and positives.**

```
[11]: # Building Confusion Matrix
confusion_matrix(y_test,y_pred, labels=['FAKE', 'REAL'])
```

```
[11]: array([[592,  46],
          [ 43, 586]], dtype=int64)
```

So with this model we have 592 true positives, 586 true negatives, 43 false positives, and 46 false negatives.

### 2.1 Summary

With this project we learned to detect fake news with Python. We took a political dataset, implemented a TfidfVectorizer, initialized a PassiveAggressiveClassifier, and fit our model. We ended up obtaining an accuracy of 92.97% in magnitude.

The reason for selecting this model can be understood from the problem we are dealing. We needed to detect whether the news is Real or False on a large dataset. For example in case of

'Twitter' where there are millions of comments or posts per hour, it is computationally expensive to use a batch algorithm because of the sheer size of the data. That's why we used Passive-Agressive classifier which is an online-learning algorithm where the algorithm will get a training example, update the classifier, and then throw away the example.