

Detecting Fake News with Python and Machine Learning

Developed by: Nimisha Davis

Linkedin : <https://www.linkedin.com/in/nimishadavis/>

What is Fake News?

Fake news refers to fabricated or exaggerated information, often spread through social media and other online platforms. It is a form of yellow journalism designed to mislead or manipulate readers, frequently serving political or ideological purposes. These news items often contain false or distorted claims, and their viral nature is amplified by algorithms, creating filter bubbles where users are exposed only to similar content.

What is TfidfVectorizer?

The TfidfVectorizer is a tool used to convert textual data into numerical features based on Term Frequency (TF) and Inverse Document Frequency (IDF):

TF (Term Frequency): Measures how often a word appears in a document. Words appearing frequently in a document indicate relevance to the context. IDF (Inverse Document Frequency): Reduces the weight of words that are common across many documents, focusing on terms that are unique and significant to the dataset. This vectorizer transforms raw text into a matrix of TF-IDF features, making it suitable for machine learning models.

What is a PassiveAggressiveClassifier?

The Passive Aggressive Classifier is an online learning algorithm that updates its model dynamically:

Passive: If the prediction is correct, no changes are made. Aggressive: If there's a misclassification, the model updates itself to minimize error. This algorithm is particularly efficient for large-scale learning and does not converge like traditional algorithms. Its goal is to make precise updates while minimally altering the weight vector's norm.

Detecting Fake News with Python

This project focuses on building a machine learning model to classify news articles as either REAL or FAKE. Using Python and scikit-learn, the workflow includes:

Preprocessing the dataset and applying the TfidfVectorizer to extract features. Training a PassiveAggressiveClassifier to distinguish fake news from real news. Evaluating the model's performance using the accuracy score and confusion matrix.

This approach combines advanced Python techniques to address the growing issue of fake news dissemination, offering a practical and scalable solution.

```
In [126... import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
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
```

EDA

```
In [128... #Read the data
df=pd.read_csv('news.csv')
df.head()
```

| Out[128... | 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 [129... `df.shape`

Out[129... (6335, 4)

In [130... `df.size`

Out[130... 25340

In [131... `#Get the labels`
`labels=df.label`
`labels.head()`

Out[131... 0 FAKE
1 FAKE
2 REAL
3 FAKE
4 REAL
Name: label, dtype: object

In [132... `df.isna().sum()`

Out[132... Unnamed: 0 0
title 0
text 0
label 0
dtype: int64

```
In [133... df.duplicated().sum()
```

```
Out[133... 0
```

```
In [134... df.dtypes
```

```
Out[134... Unnamed: 0    int64  
title         object  
text          object  
label         object  
dtype: object
```

Split the dataset

```
In [136... x=df['text']  
y=labels  
x_train,x_test,y_train,y_test=train_test_split(x,y, test_size=0.2, random_state=7)
```

TfidfVectorizer

```
In [138... #Initialize a TfidfVectorizer  
tfidf_vectorizer=TfidfVectorizer(stop_words='english', max_df=0.7)  
  
#Fit and transform train set, transform test set  
tfidf_train=tfidf_vectorizer.fit_transform(x_train)  
tfidf_test=tfidf_vectorizer.transform(x_test)
```

PassiveAggressiveClassifier

```
In [140... #Initialize a PassiveAggressiveClassifier  
pac=PassiveAggressiveClassifier(max_iter=50)  
pac.fit(tfidf_train,y_train)  
  
#Predict on the test set and calculate accuracy  
y_pred=pac.predict(tfidf_test)
```

```
score=accuracy_score(y_test,y_pred)
print(f'Accuracy: {round(score*100,2)}%')
```

Accuracy: 93.21%

```
In [141... #Build confusion matrix
confusion_matrix(y_test,y_pred, labels=['FAKE', 'REAL'])
```

```
Out[141... array([[593,  45],
       [ 41, 588]], dtype=int64)
```

So with this model, we have 589 true positives, 587 true negatives, 42 false positives, and 49 false negatives.

Key Metrics :

True Positives (TP = 589)

These are instances where the model correctly predicted "Fake News" when the actual label was "Fake News."

True Negatives (TN = 587)

These are instances where the model correctly predicted "Real News" when the actual label was "Real News."

False Positives (FP = 42)

These are instances where the model incorrectly predicted "Fake News," but the actual label was "Real News." Type I Error.

False Negatives (FN = 49)

These are instances where the model incorrectly predicted "Real News," but the actual label was "Fake News." Type II Error.

```
In [144... #END
```