

**Fake News Detection**

Using machine learning techniques

**CSE-4709** | Machine Learning | 31-Aug-2020

|  |  |
| --- | --- |
| **GROUP MEMBERS** | |
| 154446 | FOPA YUFFON  AMADOU OLABI |
| 160041082 | MOHAMED MOCTAR |
| 160041083 | TANI BARKAT |
| 160041085 | MIKAYILOU NAMBA |
| 160041087 | ALY ABDEL  KADER GELANY |

Contents

[**INTRODUCTION** 1](#_Toc49868274)

[**What is Fake News?** 1](#_Toc49868275)

[**What is TfidfVectorizer?** 2](#_Toc49868276)

[**Remarks** 2](#_Toc49868277)

[**What is a passive-aggressive Classifier?** 2](#_Toc49868278)

[**Objective** 3](#_Toc49868279)

[**The Dataset** 3](#_Toc49868280)

[**Project Workflow** 4](#_Toc49868281)

[**Schema** 4](#_Toc49868282)

[**WorKflow** 4](#_Toc49868283)

# **INTRODUCTION**

Fake news spreads like a wildfire and this is a big issue in this era. We can learn how to distinguish fake news from a real one. We will be using supervised learning to implement the model.

Do you trust all the news you hear from social media? All news is not real, right?

So how will you detect the fake news? The answer is Python.

By practicing such an advanced python project of detecting fake news, we will easily make a difference between real and fake news. Before moving ahead in this advanced Python project, we have to be aware of related terms of fake news like the **TF-IDF vectorizer**, and the **PassiveAggressive** Classifier.

This project will work you through the necessary steps and techniques used to implement such an analysis.

# **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 viral by algorithms, and users may end up in a filter bubble.

# **What is TfidfVectorizer?**

**TfidfVectorizer** - Transforms text to feature vectors that can be used as input to an estimator.

A **vocabulary** is a dictionary that converts each token (word) to a feature index in the matrix, each unique token gets a feature index.

Example: What is? (e.g.: u'me': 8 )

It tells you that the token 'me' is represented as feature number 8 in the output matrix.

## **Remarks**

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

[**TF (Term Frequency):**](https://en.wikipedia.org/wiki/Tf%E2%80%93idf) 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):**](https://en.wikipedia.org/wiki/Tf%E2%80%93idf) Words that occur many times a document, but also occur many times in many others, maybe irrelevant. IDF is a measure of how significant a term is in the entire corpus.

Example: is this a matrix or just a vector?

Each sentence is a vector, the sentences you've entered are a matrix with 3 vectors. In each vector, the numbers (weights) represent features TF-IDF score.

For example: 'Julie': 4 --> Tells you that in each sentence 'Julie' appears you will have non-zero (TF-IDF) weight.

As you can see in the 2'nd vector:

[ 0. 0.68091856 0. 0. 0.51785612 0.51785612 0. 0. 0. 0. 0. ]

The 5'th element scored 0.51785612 - the TF-IDF score for 'Julie'.

# **What is a 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.

# **Objective**

* To build a model to accurately classify a piece of news as REAL or FAKE.
* This advanced python project of detecting fake news deals with fake and real news. Using SK-Learn, 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.

# **The Dataset**

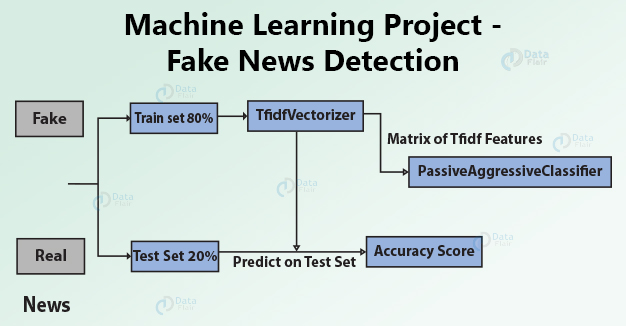
The dataset we’ll use for this python project- we’ll call it news.csv. This dataset has a shape of 7796×4. The first column identifies the news, the second and third are the title and text, and the fourth column has labels denoting whether the news is REAL or FAKE.

NB: This dataset is taken from the Kaggle website where it can be found under the title [Fake and Real news dataset](https://www.kaggle.com/clmentbisaillon/fake-and-real-news-dataset).

This dataset is based on political news feeds in the USA.

# **Project Workflow**

## **Schema**



## **WorKflow**

* Collecting Datasets on Fake News Analysis.
* The building of a dataset from a set of known and well-done datasets.
* Loading and Analyzing dataset.
* Splitting the dataset into training and testing sets.
* Preprocessing of the text.
* Choose a Learning Model, Methodology, or Schema for training the dataset.
* Fitting the Model with proper parameters and Predicting a feasible outcome. Check the Model Accuracy.
* Report and Visualization of the predicted outcomes.
* If the results are not that convincing, then Tuning and Optimizing Model with necessary algorithms is needed.
* Testing the Optimized Model and Reporting its whereabouts and results.
* After the Previous Step, if the results obtained are not still that convincing then "Repeat Previous->Previous Step" with a more efficient technique.
* Summary Report on the Model