|  |  |
| --- | --- |
| A picture containing text  Description automatically generated | Can a machine learning program identify when an article might be fake?  April Meyer  DSC630-T302 Predictive Analytics (2213-1) |

# **Executive Summary**

There is a substantial amount of information at our fingertips that allows us to be quickly informed. However, with that information comes false information that requires validation. The goal of this project is to develop a machine learning algorithm to detect that fake information, so that misinformation does not continue to be spread and we are accurately informed.

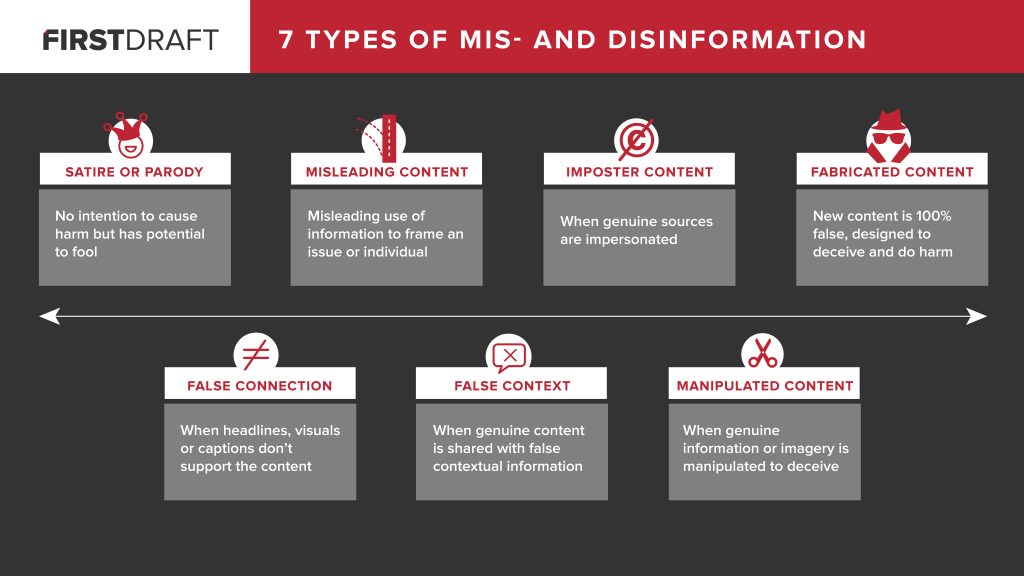
The model that was used was a Passive Aggressive Classifier. They are commonly used for large-scale learning (3). It is considered an ‘online-learning algorithm ‘, because occurs in a consecutive order and the model is updated step-by-step. The data that was used to train this model was obtained from Kaggle.

Accuracy scores and confusion matrix were used to gauge the model. A confusion matrix was created to show if there is an imbalance in classes causing the accuracy score to be high. The confusion matrix was used to calculated precision and recall. The PassiveAggressiveClassifier performed well in predicting if an article was fake or real. It had an accuracy of 96.44% with a precision 96.17% and recall of 96.81.

# **Preliminary Analysis**

## **Background of the problem**

Information is at our fingertips with the advancements of technology, but with that substantial amount of data and information comes false information. What is “False News”? It is a story that is manufactured with no verifiable data, sources, or references (1). It may be intended to mislead the reader or to received economic incentives from have higher readers. Social media has only made it easier and faster to spread fake news. Below is a visual created by Claire Wardle to show the types of misinformation (1).



## **Problem Statement**

The purpose of this project is to develop a machine learning algorithm to detect fake news, so that readers may identify worthwhile sites or links.

## **Methods**

A Passive Aggressive Classifier will be the model used. They are commonly used for large-scale learning (3). It is a ‘online-learning algorithm ‘. For these types for of algorithms the input data occurs in consecutive order and the model is updated step-by-step, instead of batch learning (entire training dataset is used at once) (3). If the predication is correct the model does not change (passive) and if the predication is incorrect the model will update (aggressive). Below are the parameters for this model:

* C = regularization parameter. It signifies the correction the model will make on an inaccurate prediction (3).
* max\_iter = maximum number of iterations to be completed over the training data. (3).
* tol = stopping criterion.
  + If it is set to None, the model will stop when loss > previous\_loss – tol
  + Default is set to 1e-3

I will also need to have TfidfVectorizer in place to fit the Passive Aggressive Classifier Model. The TF-IDF stands for Term Frequency Inverse Document Frequency (2). The term frequency is the number of times a word appears in a document while the inverse document frequency is how important a term is (2).

Accuracy scores and confusion matrix will be used to evaluate the model.

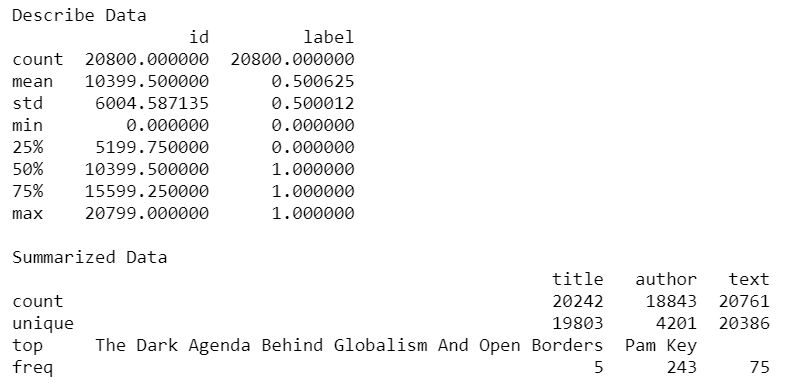
## **Data Sources**

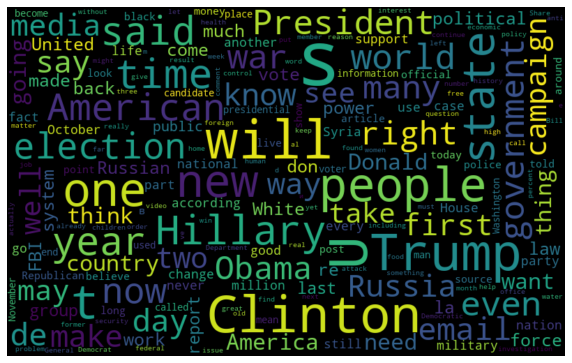
The data is from Kaggle, but I would like to use multiple sets to see the different accuracy scores. Link to data, <https://www.kaggle.com/c/fake-news/data>.

## **Data Exploration**

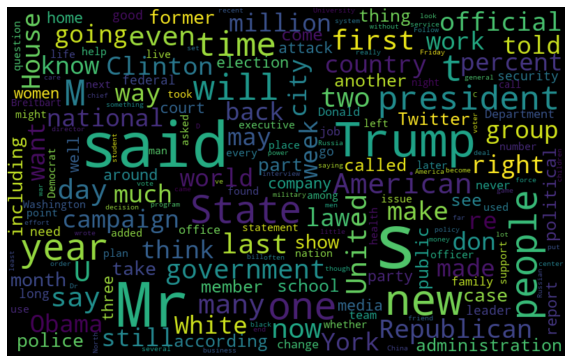
The initial analysis of the attributes for the dataset are below:

* id: unique id for a news article
* title: the title of a news article
* text: the text of the article (might be incomplete)
* label: a label that marks the article as potentially unreliable
  + 1: unreliable
  + 0: reliable



**WorldCloud for Fake Features:**

**WorldCloud for Real Features:**



## **Data Preparation**

The only data preparation required was a TfidfVectorizer. I used English for the stop words. The maximum document frequency was set to 0.7. English stop words that were used most often were removed. The TfidfVectorizer converted the documents into a matrix of TF-IDF features.

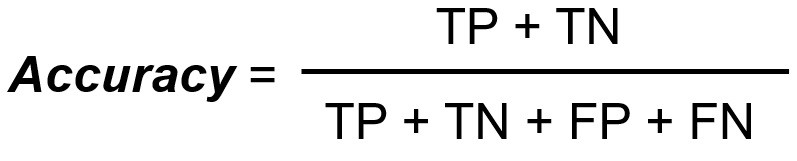
## **Modeling**

To start the data was split using train\_test\_split. It is a function from Sklearn for splitting data arrays into two subsets: for training data and for testing data. test\_size was set to 0.2 which specifies the size of the testing dataset.

Next the vectorizer was fit and transformed on the train set and transformed set. Once this was completed the PassiveAggressiveClassifier was set and fit on the tfidf\_train and y\_train.

## **Results**

Accuracy scores and confusion matrix was used to evaluate the model. Classification Accuracy was calculated with the below:



The result was an accuracy of 96.44%.

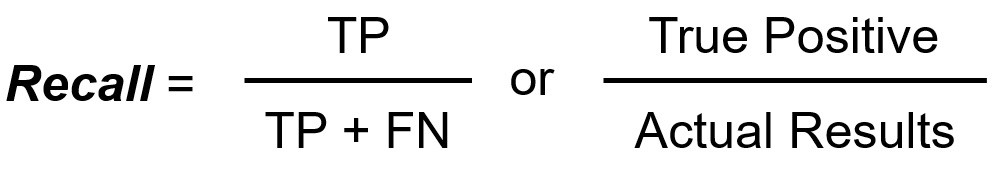
Next a confusion matrix was created to show if there is an imbalance in classes causing the accuracy score to be high. Using the confusion matrix, precision and recall can be calculated.

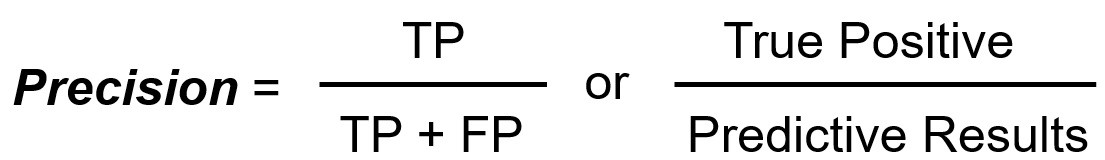
The confusion matrix resulted in 2033 true positives, 1979 true negatives, 81 false positives, and 67 false negatives.



Precision is the True Positive divided by the True Positive plus the False Positive and is 96.17%. Precision shows how many are actual positive out of the predicted positive.

Recall is the True Positive divided by the True Positive plus the False Negative and is 96.81%. Recall will show how many of the Actual Positives the model captured by labeling it as True Positive.





## **Discussion/Conclusion**

Overall, the PassiveAggressiveClassifier performed well in predicting if an article was fake or real. It had an accuracy of 96.44% with a precision 96.17% and recall of 96.81. Future applications of this model could be used by social media website/apps to filter out the fake news or miss-information that shows up on customer’s feeds. I would also like to investigate seeing if using the ‘title’ feature to predict the article’s likelihood of being Fake.

## **Acknowledgements**

I would like to acknowledge the Kaggle Repository and UTK Machine Learning Club for providing this dataset I used for the project. I would also like to acknowledge Filippos Dounis. His code gave me a reference point and guided on using a modeling to determine fake news.

# **References:**

# <https://guides.lib.umich.edu/fakenews>

# <https://medium.com/@cmukesh8688/tf-idf-vectorizer-scikit-learn-dbc0244a911a>

# <https://medium.com/swlh/detecting-fake-news-with-python-and-machine-learning-f78421d29a06>

# <https://www.geeksforgeeks.org/passive-aggressive-classifiers/>