**Machine Learning Project Report**

**“Fake News Classifier”**

Report​

​

# Table Of Content

[1 Introduction 2](#_Toc55342487)

[2 DataSet 2](#_Toc55342495)

[3 Data Exploration](#_Toc55342499) 4

[4 Data Preprocessing](#_Toc55342499) 5

[5 Data Visualization](#_Toc55342499) 7

[6 Feature Engineering](#_Toc55342499) 9

[7 Training Model](#_Toc55342499) 12

[8 Conclusion 2](#_Toc55342499)3

[9 References 2](#_Toc55342499)4

# Introduction

Fake news is misinformation or manipulated news that is spread across the social media with an intention to damage a person, agency and organization. With the current usage of social media platforms, consumers are creating and sharing more information than ever before, some of which are misleading with no relevance to reality. Fake news has been gathering a lot of attention worldwide recently.

It is very easy for someone to write fake news on the web that would grab the people's attention. Automated classification of a text article as misinformation or disinformation is a challenging task. Even an expert in a particular domain has to explore multiple aspects before giving a verdict on the truthfulness of an article. As time flows, the amount of data, especially text data increases exponentially. Along with the data, our understanding of AI also increases and the computing power enables us to train very complex and large models faster. Because of the increasing spread of such unverified click-bait articles on every platform, detecting fake news is believed to be a complex yet important task in the recent years.

This project aims to classify fake news from a dataset of labelled news articles using the Natural Language Processing, Machine Learning algorithms and Deep Learning techniques. Use of bag-of-words, n-grams, TF-IDF has been made, and the data is trained on different classifiers to investigate which of them works well for this specific dataset of labelled news statements. The precision, recall and f1 scores help us determine which model works best.

# Data Set

In this project, Fake and Real News dataset from Kaggle is used for this Fake News Classification task. The link to the dataset is given below:

<https://www.kaggle.com/clmentbisaillon/fake-and-real-news-dataset>

Below files are given in this dataset:

* fake.csv​ – collection of fake news
* true.csv​ – collection of real news

The link to these files are as follows:

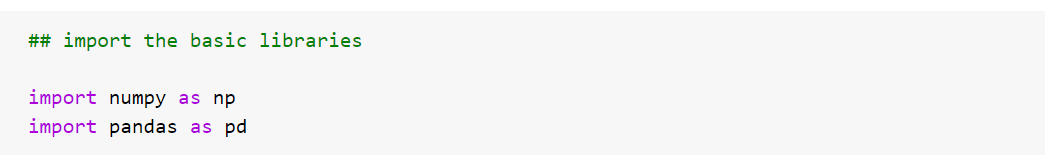
Fake.csv:

<https://drive.google.com/file/d/1V7KFJKOSvvqurJovOYWL0_36icCEWjOO/view?usp=sharing>

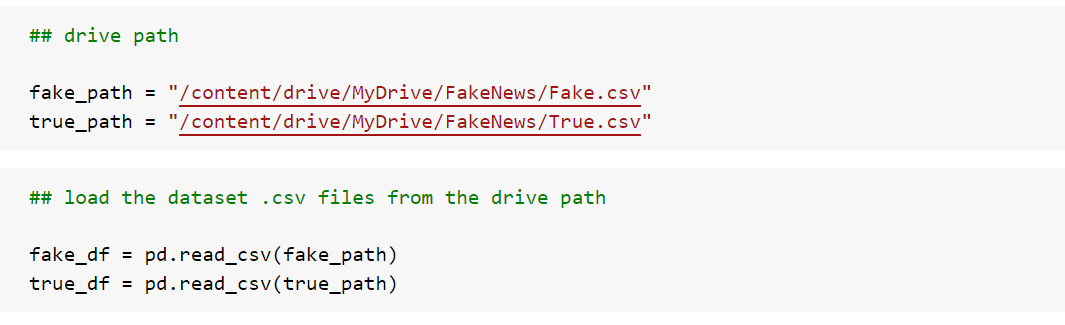
True.csv:

<https://drive.google.com/file/d/1AB_FpNFclUOOrklWxR0nsaqz96PbOaOK/view?usp=sharing>

We will import below libraries:

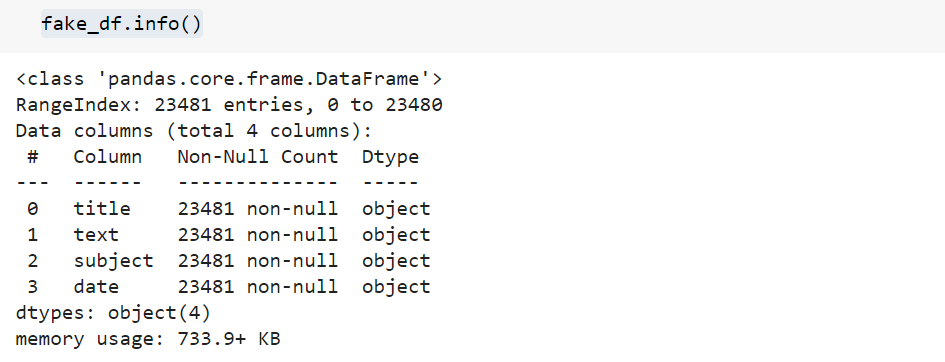


**Reading Dataset:**



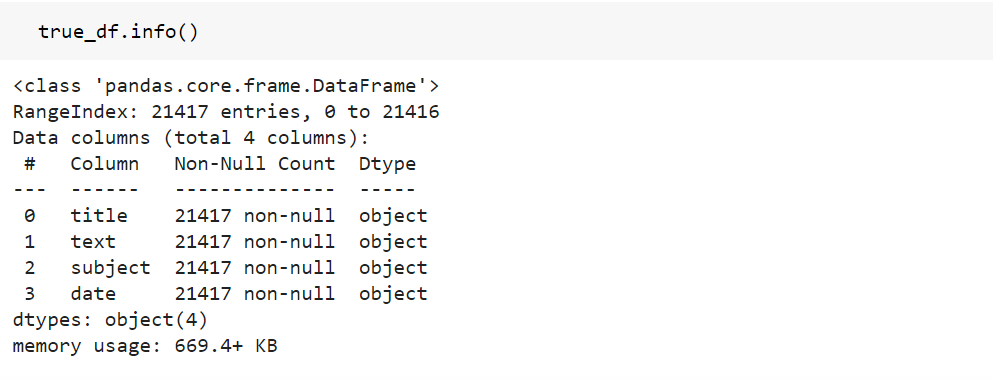
**fake.csv**

* **title** (object) : The title of the article.
* **text** (object) : The text of the article.
* **subject** (object) : The subject of the article.
* **date** (object) :The date at which the article was posted.



**true.csv**

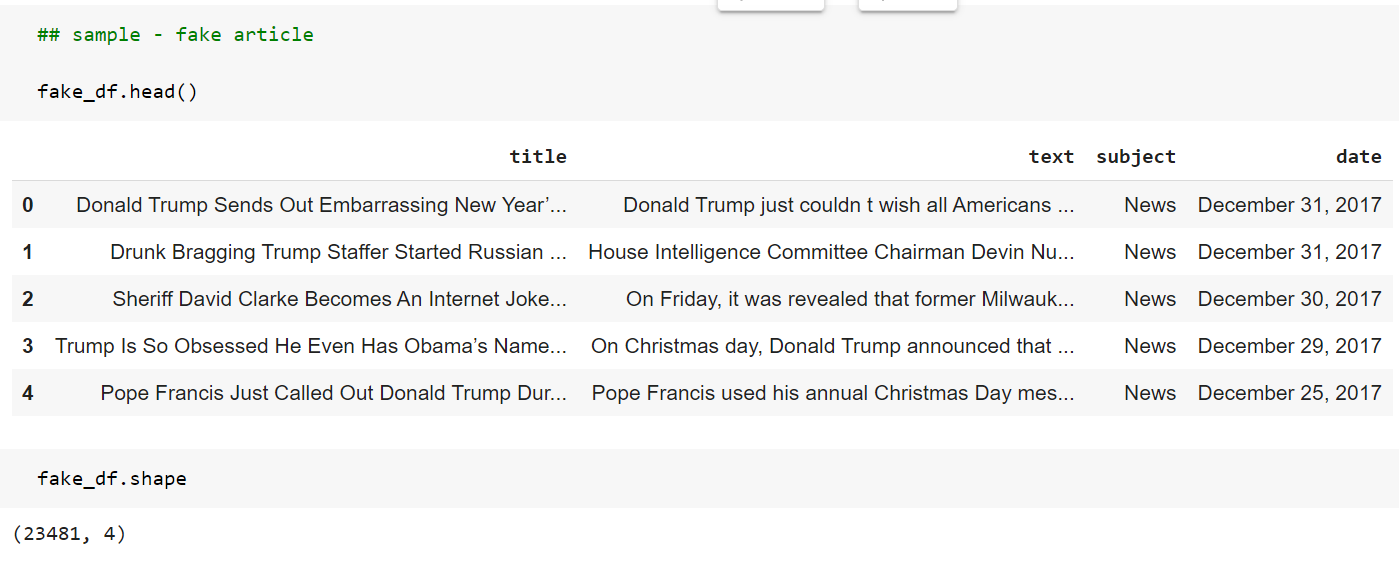
* **title** (object) : The title of the article.
* **text** (object) : The text of the article.
* **subject** (object) : The subject of the article.
* **date** (object) :The date at which the article was posted.



# Data Exploration

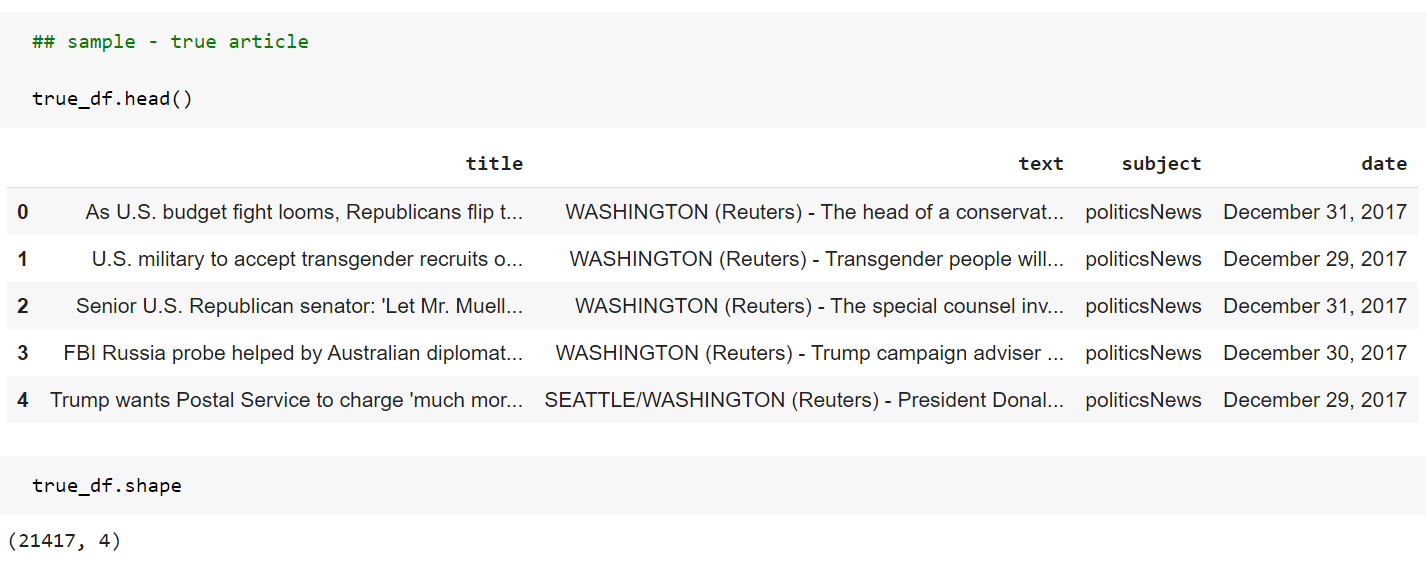
1. **View fake.csv dataset:**

The fake.csv file contains 23481 fake news articles. The dataset is created for this file is **‘fake\_df’.**



1. **View true.csv dataset:**

The true.csv file contains 21417 true news articles. The dataset is created for this file is ‘**true\_df’.**

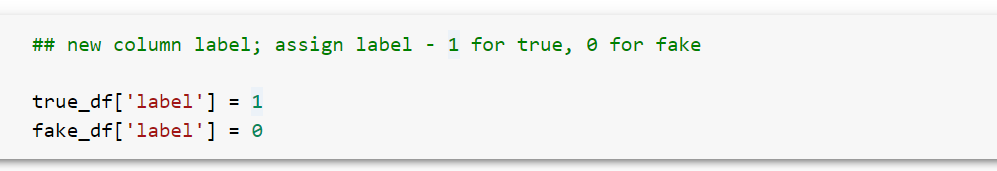


# Data Preprocessing

The fake.csv file contains 23481 fake news articles and the true.csv file contains 21417 true news articles.

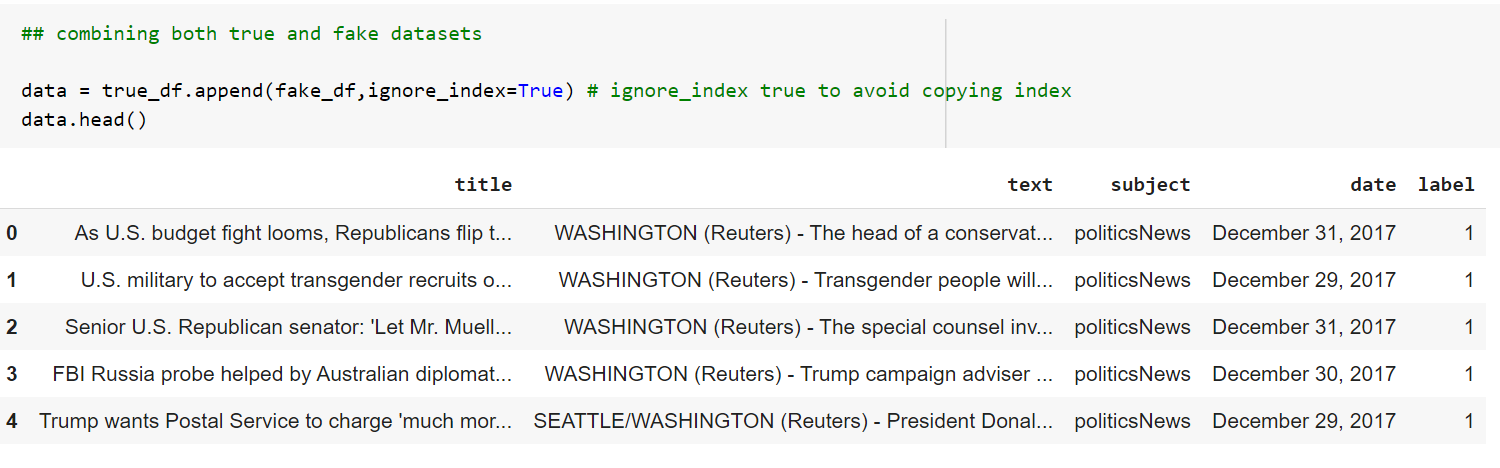
1. **Create a new column ‘label’ :**

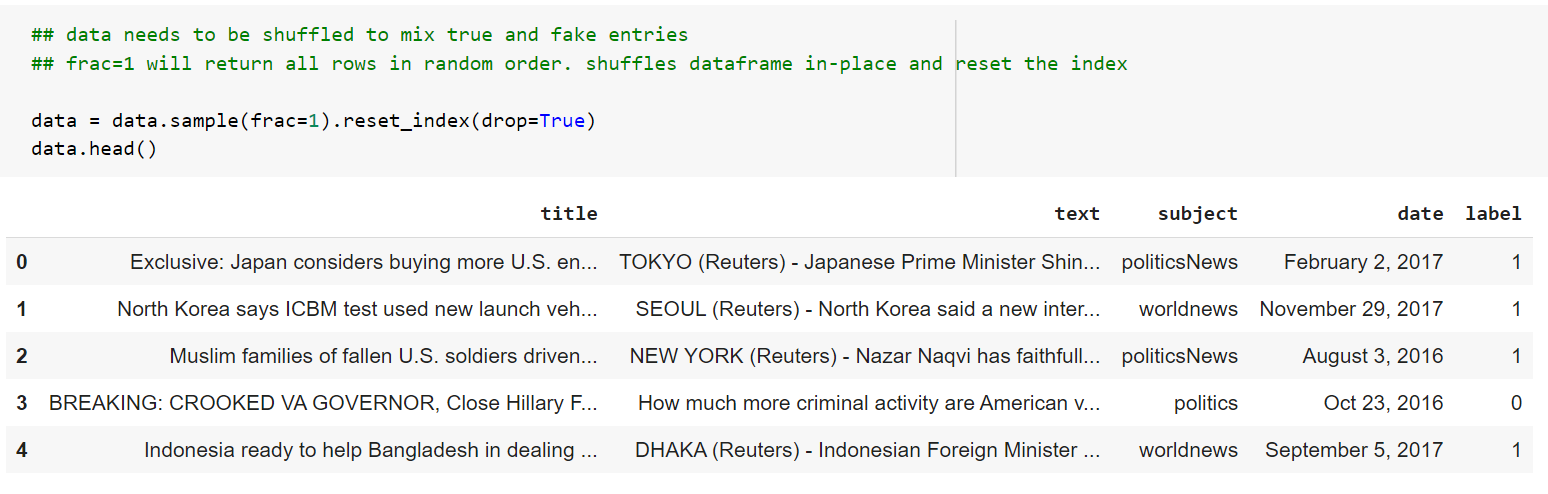
A new column **‘label’** is created in both the true dataset: **true\_df** ; and fake dataset: **fake\_df** which denotes the class of the news - whether the article is fake or true. The ‘label’ column is assigned – ‘1’ for true news article, ‘0’ for fake news article.



1. **Combine both the true\_df and fake\_df datasets:**

For the purpose of classification task, both the fake and true datasets are combined and shuffled to create the main dataset that would be used for training the model.

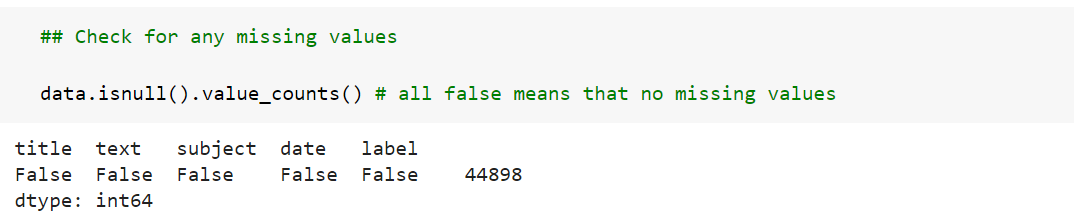




The combined dataset is named **‘data’** and has **44898 records**. The resulting dataset will be balanced because it contains approximately equal number of fake and true news. A dataset that has a balanced number of records for both the classes will help to create a model with better accuracy.

1. **Check for missing values:**

The dataset **‘data’** is checked for any missing or NaN values.

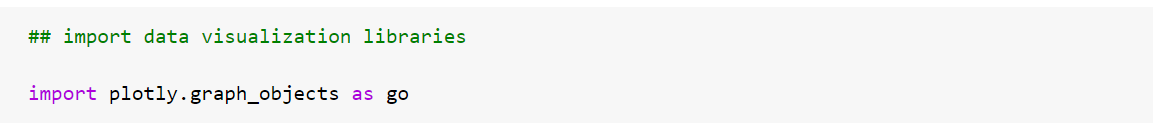


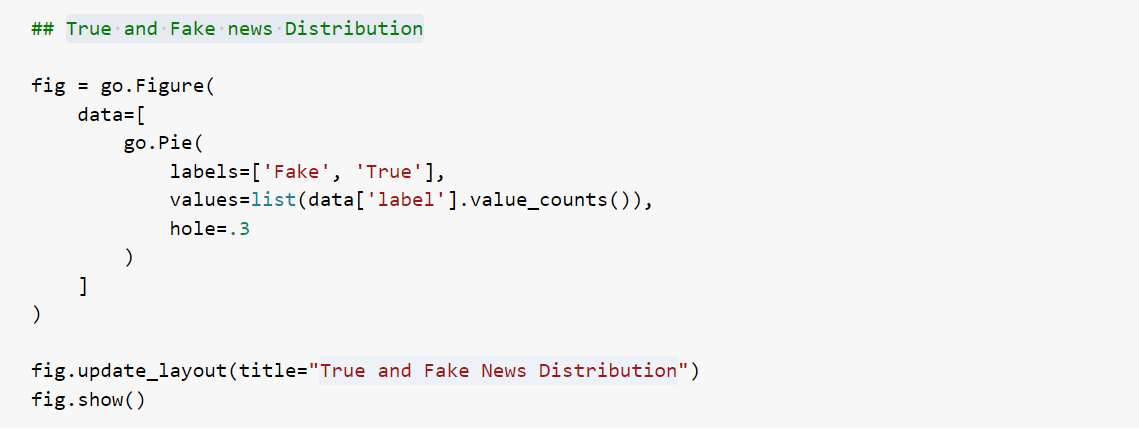
There are no missing values in the dataset.

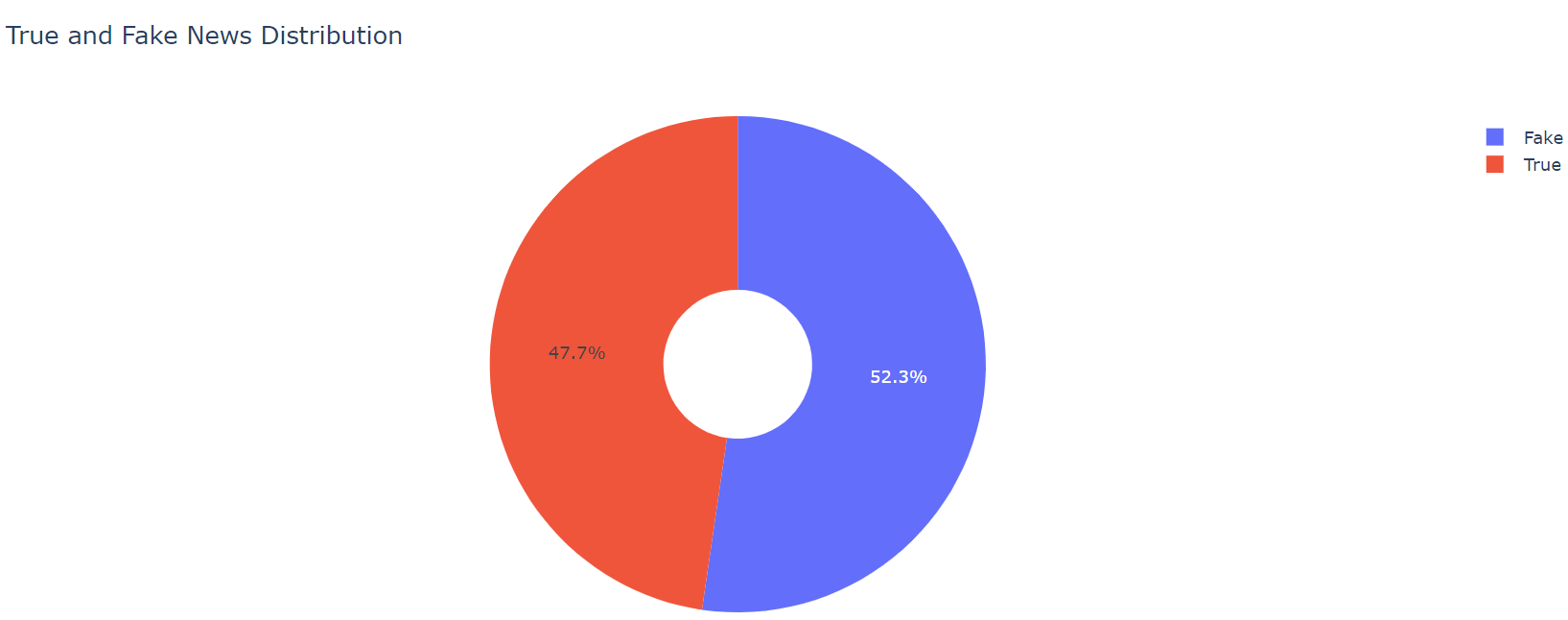
# Data Visualization

1. **True and Fake news Distribution:​**

The distribution of True and Fake news articles is depicted using Pie charts to show the percentage of both the news in the dataset. To generate the charts, plotly python library is used. 47.7% of total news is true articles and 52.3% is fake articles.

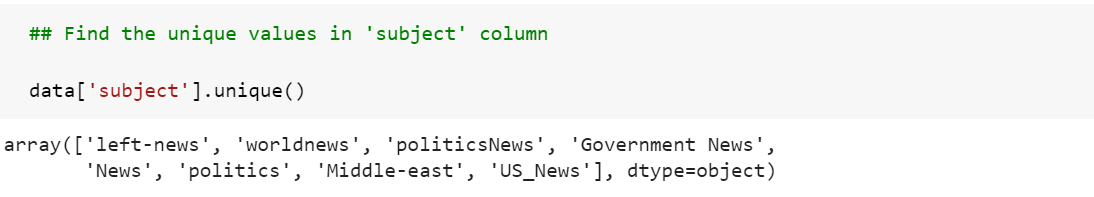






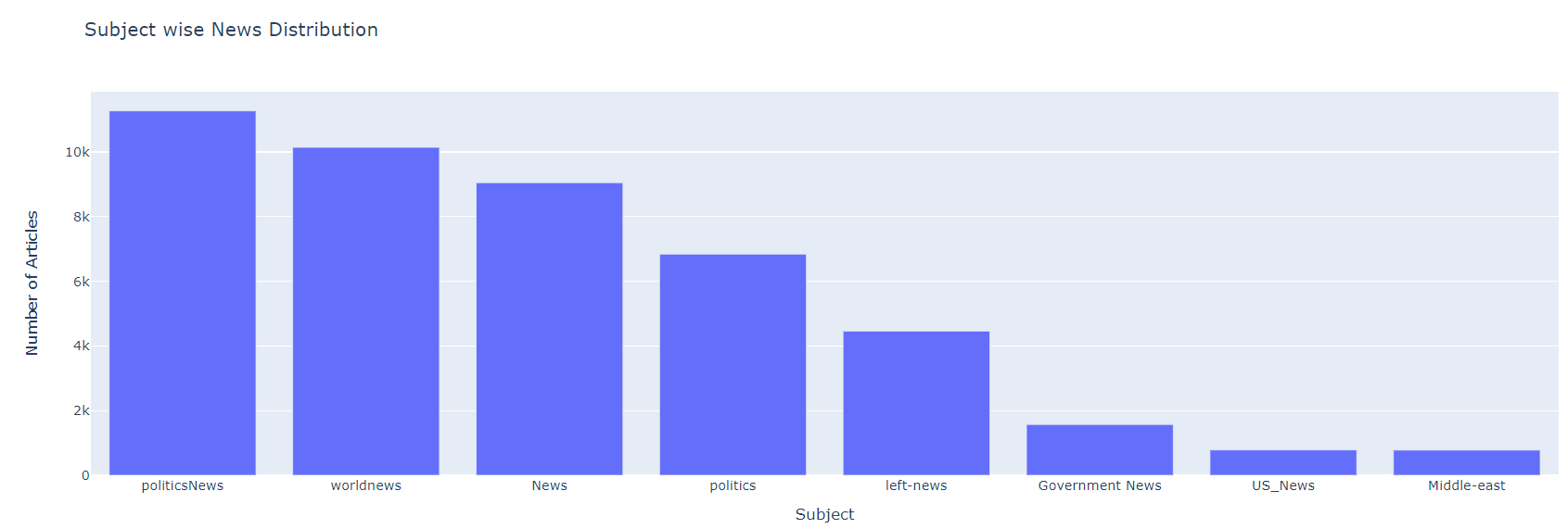
1. **Subject wise news Distribution:​**

The news articles in the dataset belong to different subject categories. The subject categories are:



A bar-chart can be generated to view the number of articles belonging to each category as below:





# Feature Engineering

1. **Define dependent and independent features:​**

For the fake news classification task, the **‘text’** column will be used to train the model and predict the **‘label’** to which it belongs. Therefore, the **‘text’** column values will be used to extract independent features and the **‘label’** column becomes the dependent feature**.**



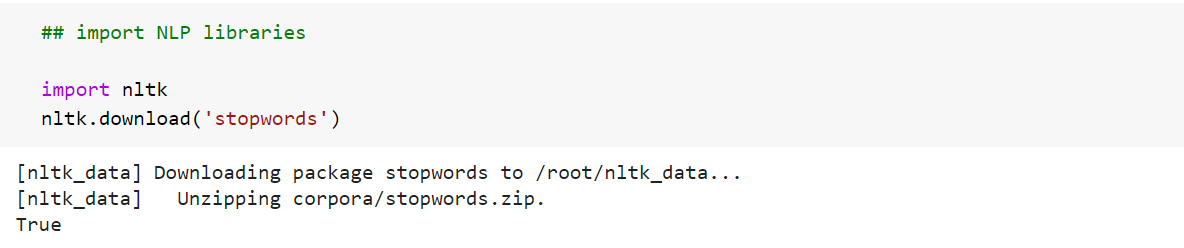
1. **Use NLTK to clean text:​**

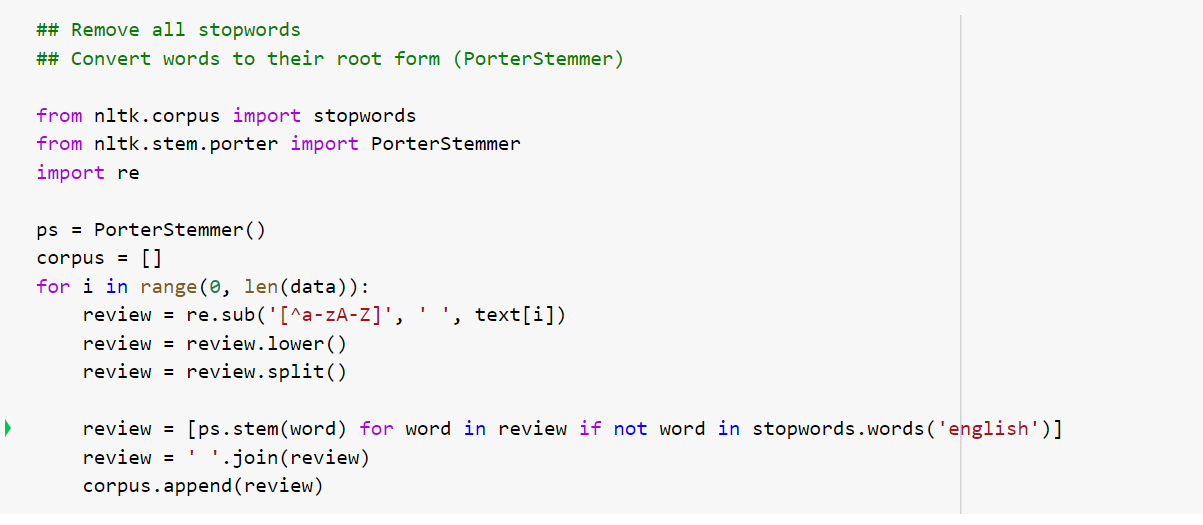
NLTK library from python is very powerful and useful for any Natural Language Processing Task. With the help of NLTK stopwords and PorterStemmer, the news article text data can be cleaned in order to reduce the text length and get more meaningful words. Both these techniques are applied on the **‘text’** column.

**Stopwords:** Stopwords are the most common words in any natural language. For the purpose of analyzing text data and building NLP models, these stopwords might not add much value to the meaning of the document. Generally, the most common words used in a text are “the”, “is”, “in”, “for”, “where”, “when”, “to”, “at” etc.

For text classification, where the text is to be classified into different categories, stopwords are removed or excluded from the given text so that more focus can be given to those words which define the meaning of the text.

**PorterStemmer:** Stemming is normalization of words, which means reducing a word to its root form. Stemming cuts off the end or beginning of a word by taking into account a list of common prefixes or suffixes that could be found in that word. In most natural languages, a root word can have many variants. For example, the word ‘play’ can be used as ‘playing’, ‘played’, ‘plays’, etc.





1. **Use TFIDF Vectorizer:​**

The TfidfVectorizer class from the sklearn.feature\_extraction.text module can be used to create feature vectors containing TF-IDF values. TF-IDF is a product of two terms: TF(Term Frequency) and IDF(Inverse Document Frequency). TF-IDF which measures how important a particular word is with respect to a document and the entire corpus. Words which are rare in a document will have a high score in the TF-IDF vector.

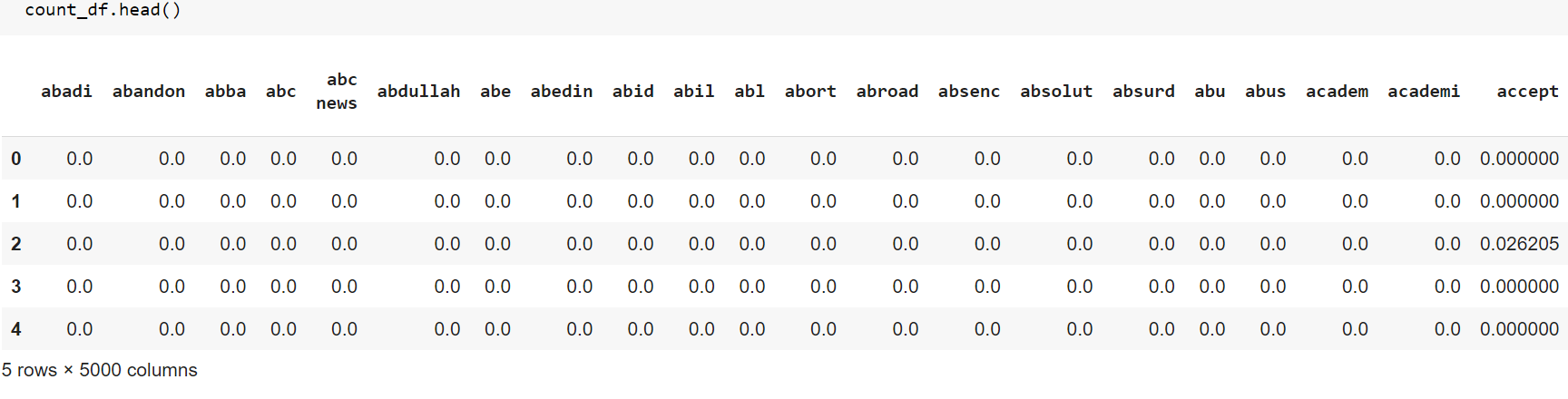
An N-Gram is a sequence of N-words in a sentence. Here, N is an integer which stands for the number of words in the sequence. For example, N=1, then it is referred to as a uni-gram. If N=2, then it is a bi-gram. Similarly if N=3, then it is a tri-gram.



Here, 5000 features are created using TFIDF Vectorizer with N-gram range (1,3). These 5000 most relevant features in the corpus are selected to be used for training the model.

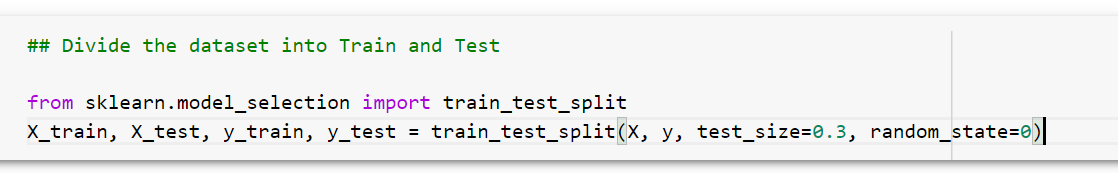


All the news articles in the dataset will be now represented by these 5000 features and each feature will have a value depending upon its occurrence in the article.



1. **Split Train and Test data:​**

The dataset is divided into Train and Test data. 70% of the data will be used to train the model and 30% of the data will be used to predict the output and test the model for accuracy.



# Training Models

1. **Multinomial Naïve Bayes Algorithm:​**

The Multinomial Naive Bayes algorithm is based on the Bayes theorem. It calculates the probability of an event occurring based on the prior knowledge of conditions related to an event, based on the following formula:

P(A|B) = P(A) \* P(B|A)/P(B)

The probability of event A is calculated when probability of event B is already provided.

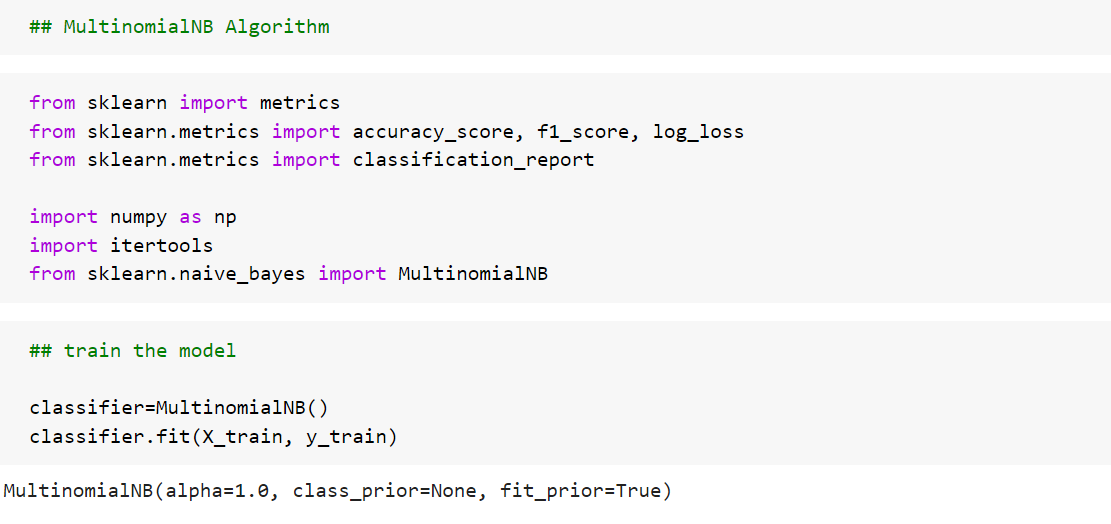
P(B) = prior probability of B

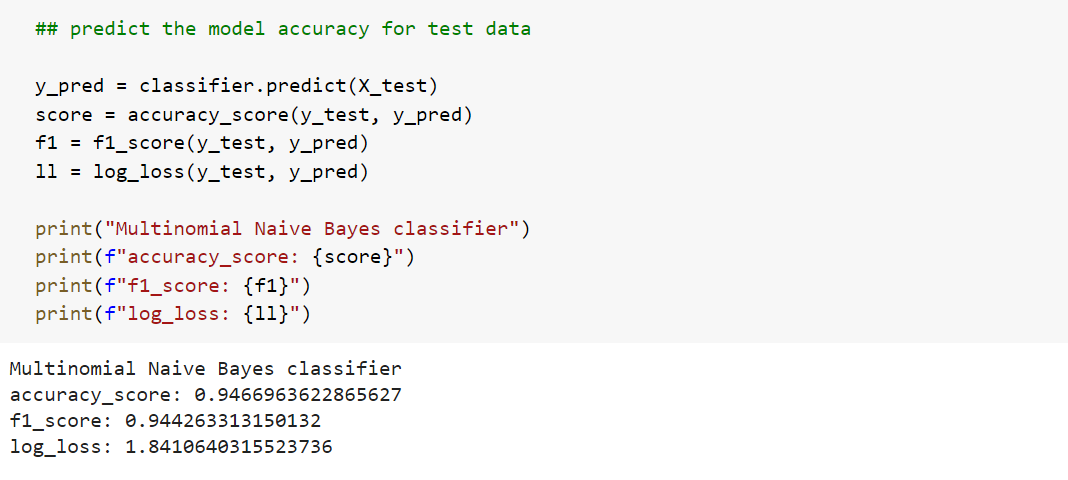
P(A) = prior probability of A

P(B|A) = occurrence of B given probability of A

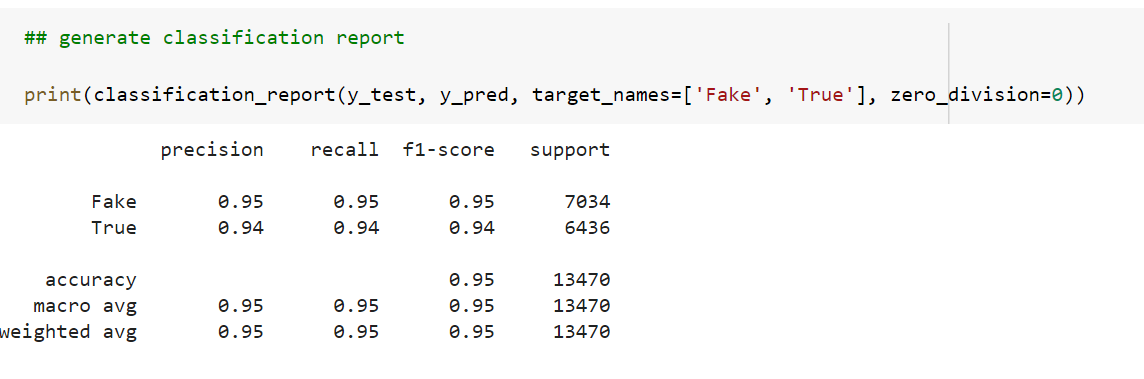
The above formula calculates the probability of each tag for a given sample and then gives the tag with the highest probability as output.

A Naive Bayes classifier is a probabilistic machine learning model that is used for classification task. It is fast and easy to implement and one of the most popular and simple machine learning classification algorithms.





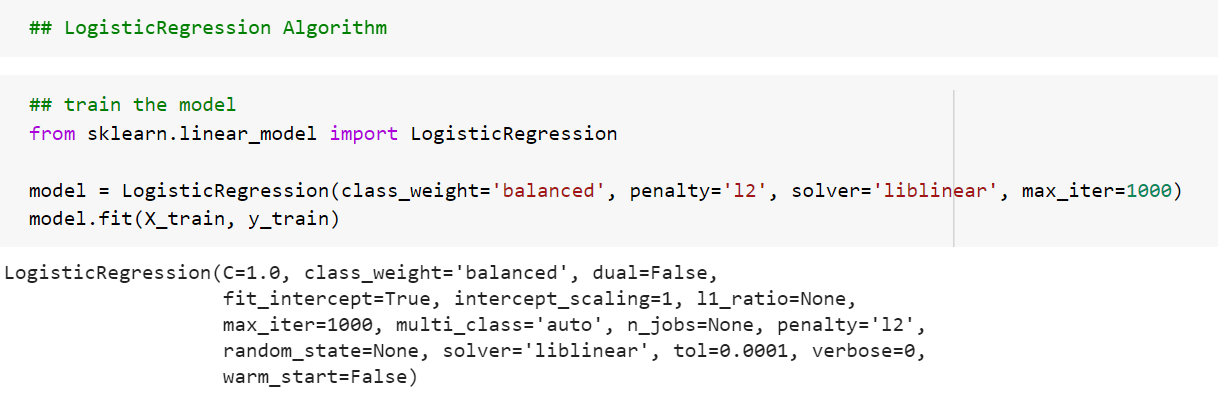




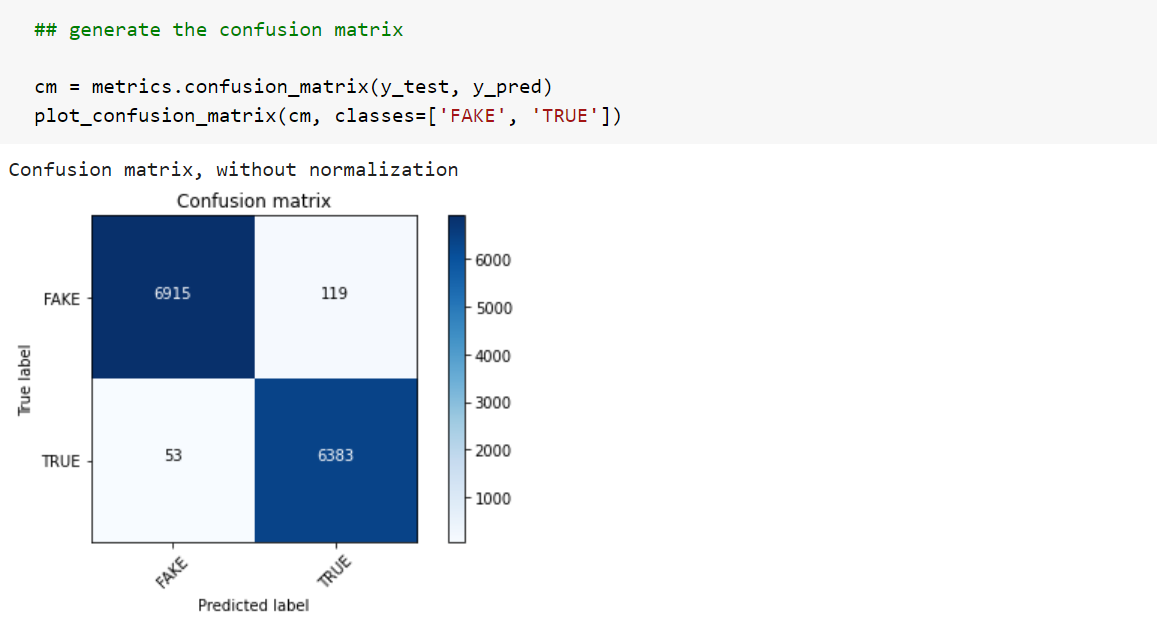
1. **Logistic Regression Algorithm:​**

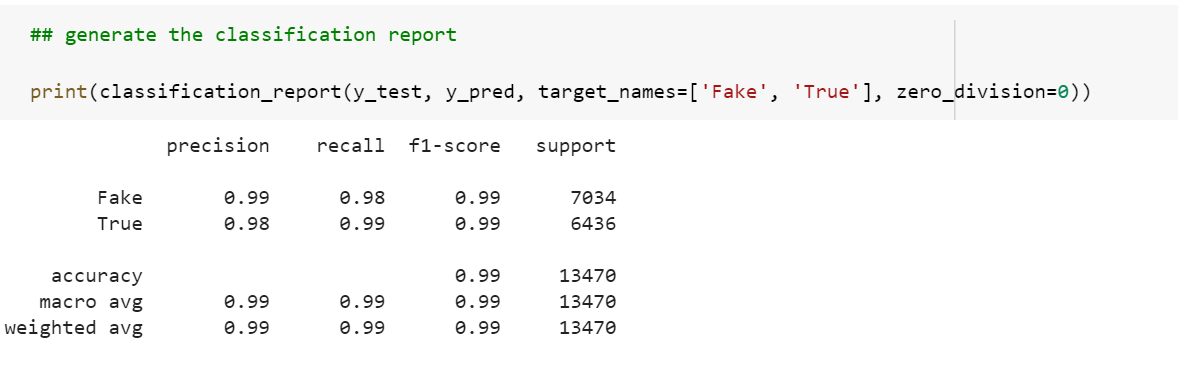
Logistic regression is a supervised classification algorithm, used when the value of the target variable is categorical in nature. The target variable (or output), y, can take only discrete values for given set of features (or inputs), X. The model builds a regression model to predict the probability that a given data entry belongs to the category numbered as “1”. It is a powerful machine learning algorithm that utilizes a sigmoid function and works best on both binary and multi-class classification problems.

Logistic regression becomes a classification technique only when a decision threshold is set. This setting is dependent on the classification problem itself. The decision for the value of the threshold value is majorly affected by the values of precision and recall. Ideally, we want both precision and recall to be 1, but this doesn’t happen practically.









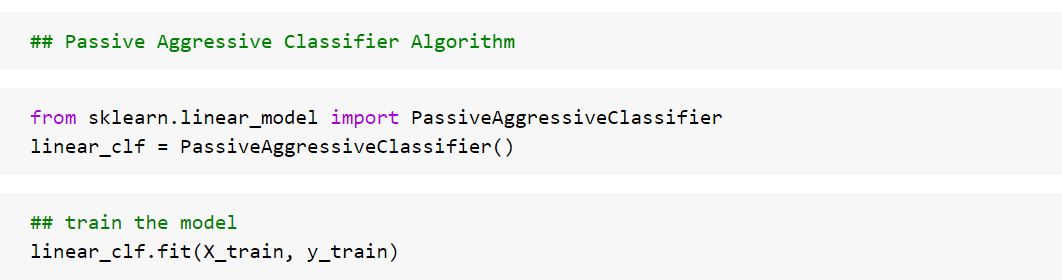
1. **Passive Aggressive Classifier Algorithm:​**

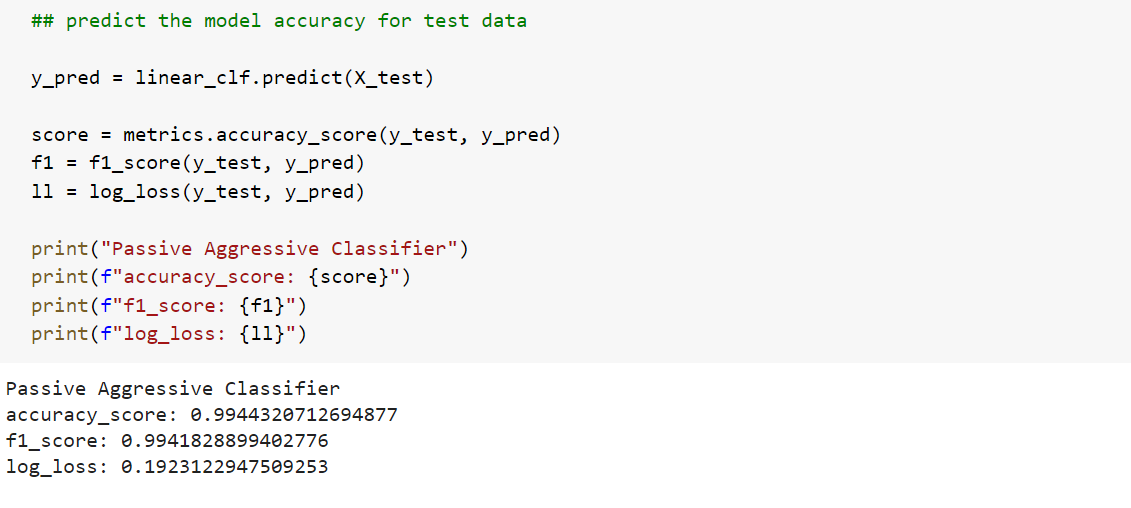
The Passive Aggressive Classifier Algorithm is an “online learning algorithm”. These algorithms are generally used for large-scale learning. In online machine learning algorithms, the input data comes in sequential order and the machine learning model is updated step-by-step, as opposed to batch learning, where the entire training dataset is used at once. This is very useful in situations where there is a huge amount of data and it is computationally infeasible to train the entire dataset because of the sheer size of the data.

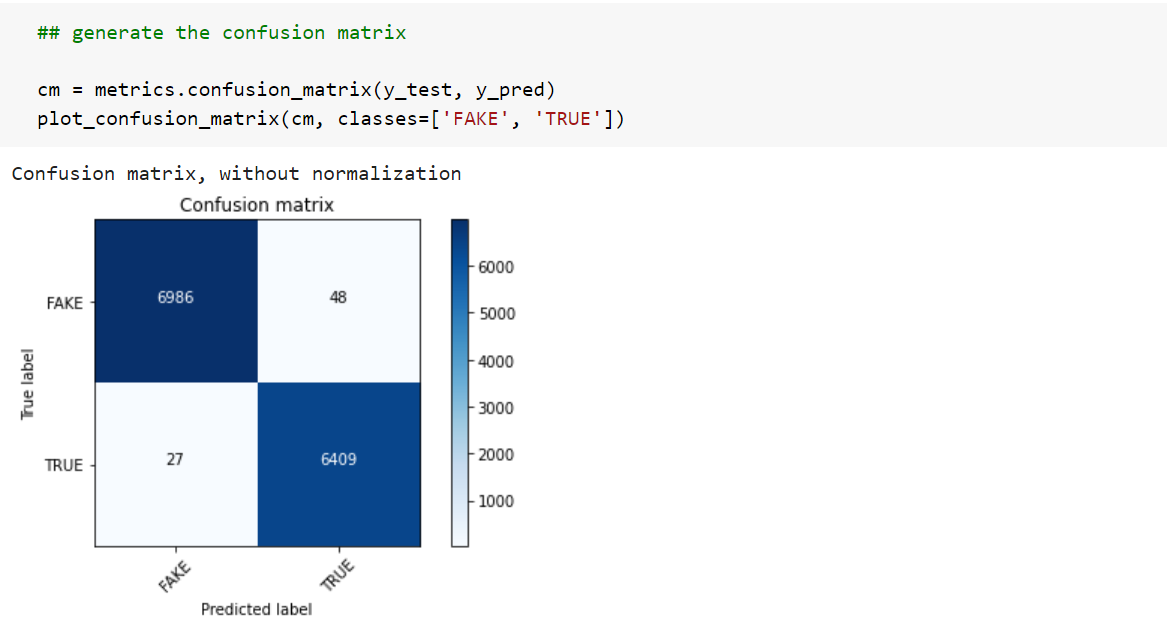
Passive-Aggressive algorithms do not require a learning rate. However, they do include a regularization parameter. Passive-Aggressive algorithms are called so because:

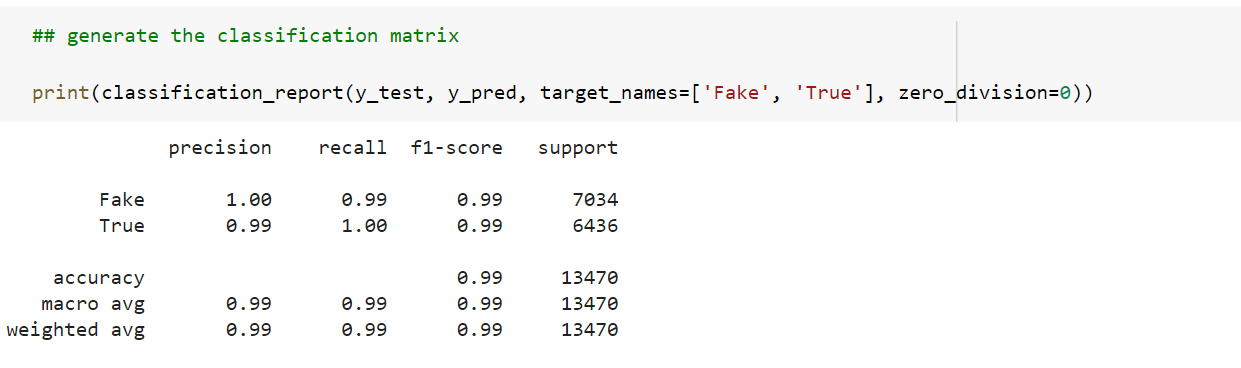
Passive: If the prediction is correct, keep the model and do not make any changes. i.e., the data in the example is not enough to cause any changes in the model.

Aggressive: If the prediction is incorrect, make changes to the model. i.e., some change to the model may correct it.

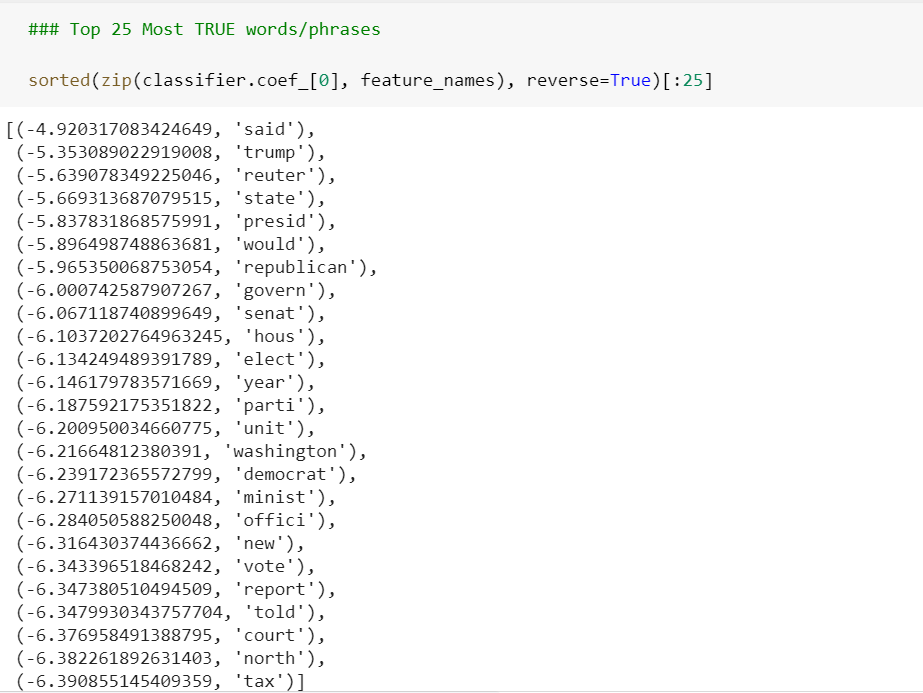


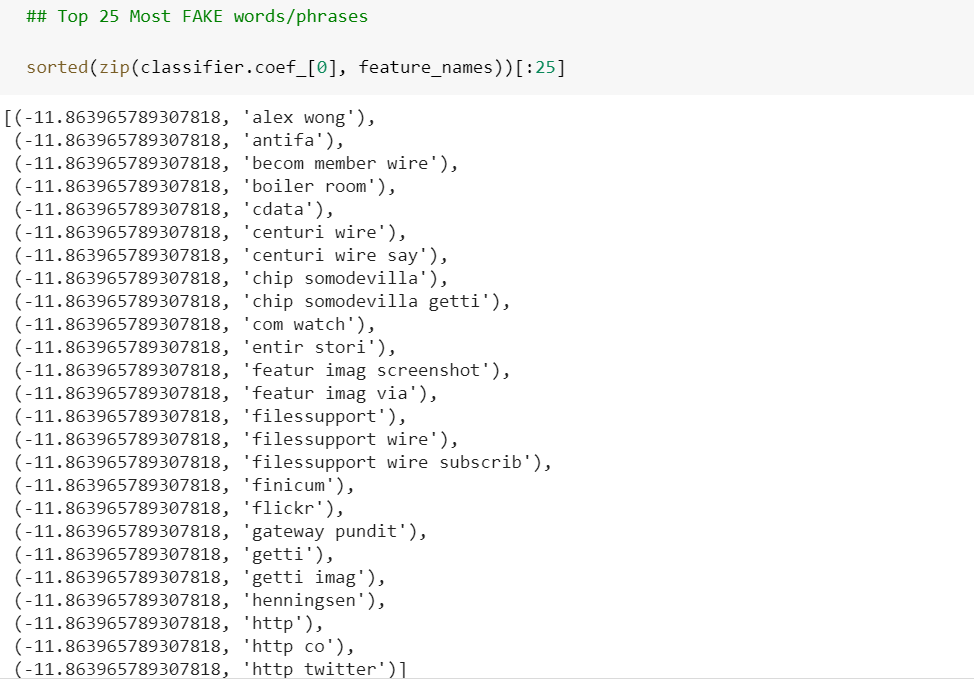






In this fake news classification task, Passive-Aggressive Classifier gives the best performance. It gives both accuracy and F1-score of 99%, the classifier can be used to predict the most relevant features for this dataset.



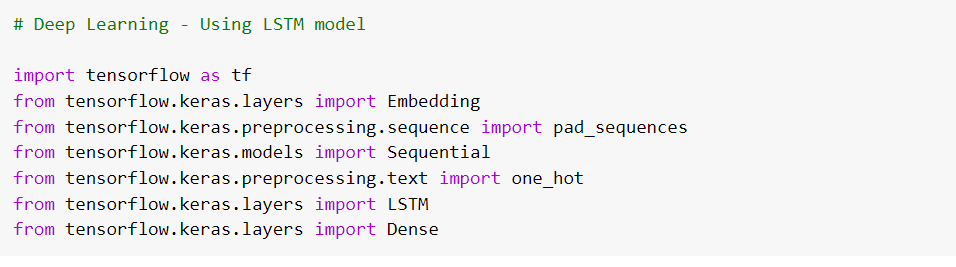


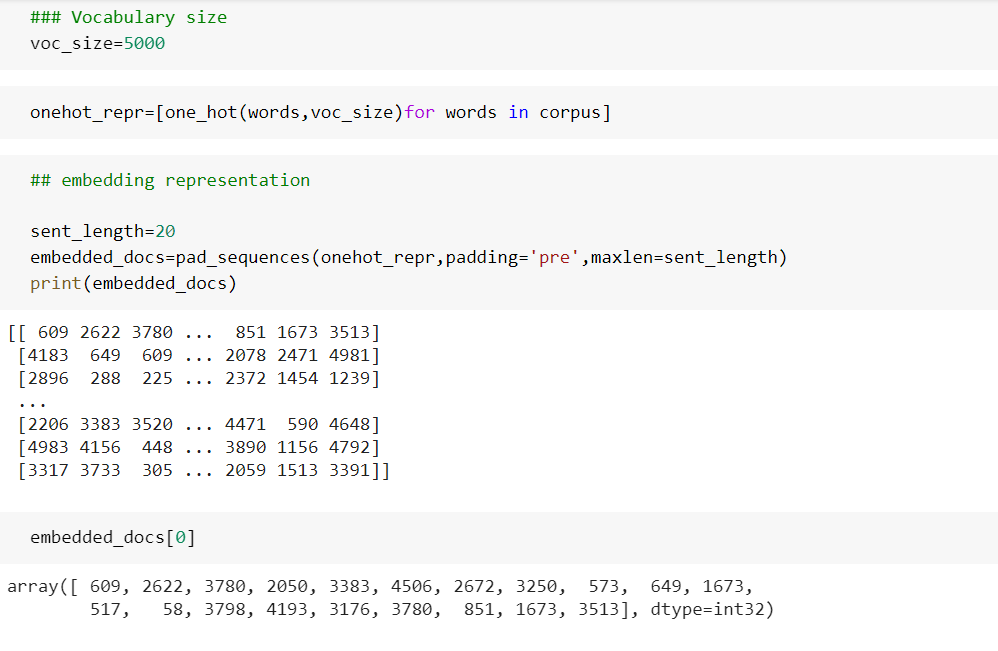
1. **LSTM (Long short-term memory) Algorithm:​**

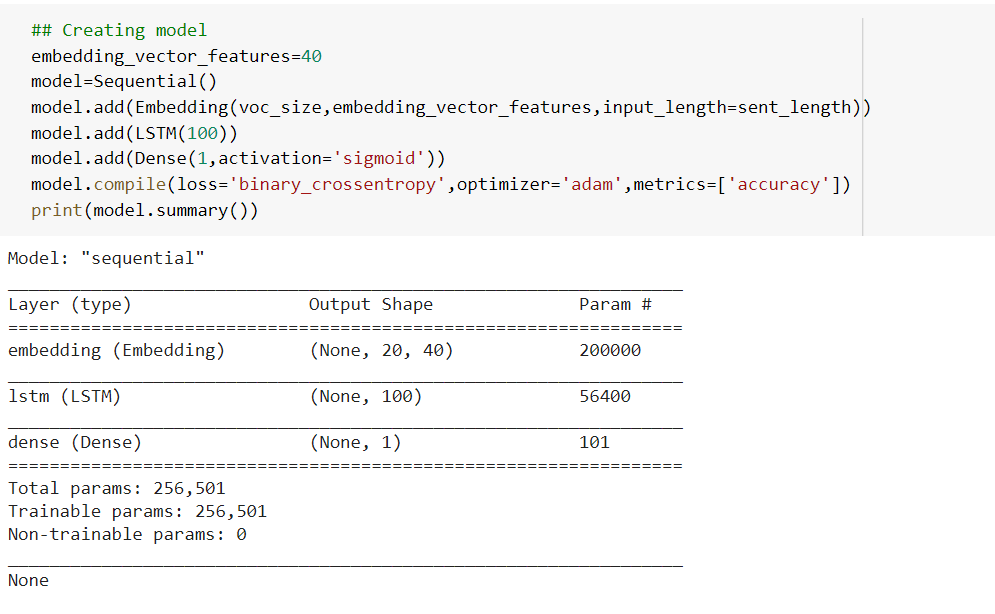
The Long short-term memory (LSTM model) is an artificial recurrent neural network architecture used in the field of deep learning. Unlike standard feed-forward neural networks, LSTM has feedback connections. It can not only process single data points, but also entire sequences of data.

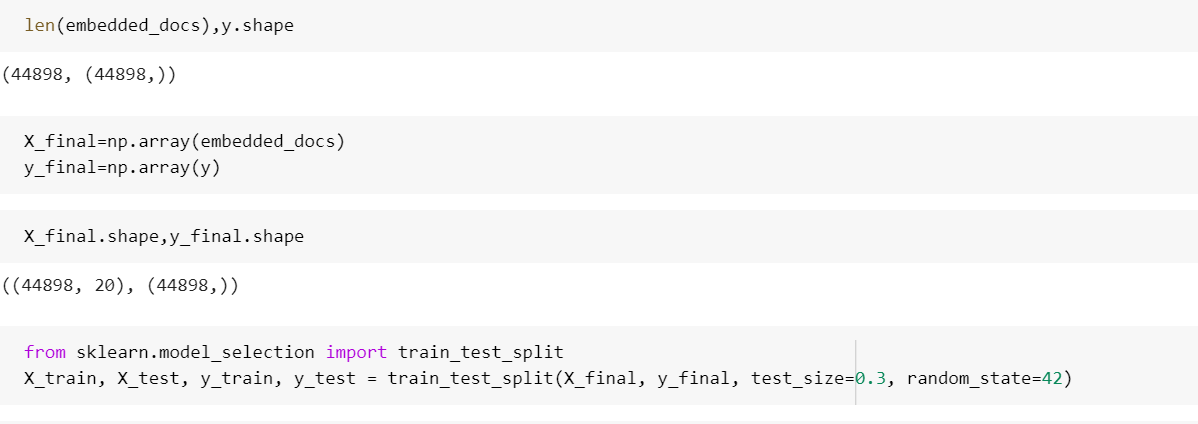
The reason to use LSTM is that it is effective in memorizing important information. In non-neural network classification techniques, models are trained on multiple word as separate inputs that are just word having no actual meaning as a sentence, and while predicting the class it will give the output according to statistics and not according to meaning. That means, every single word is classified into one of the categories.

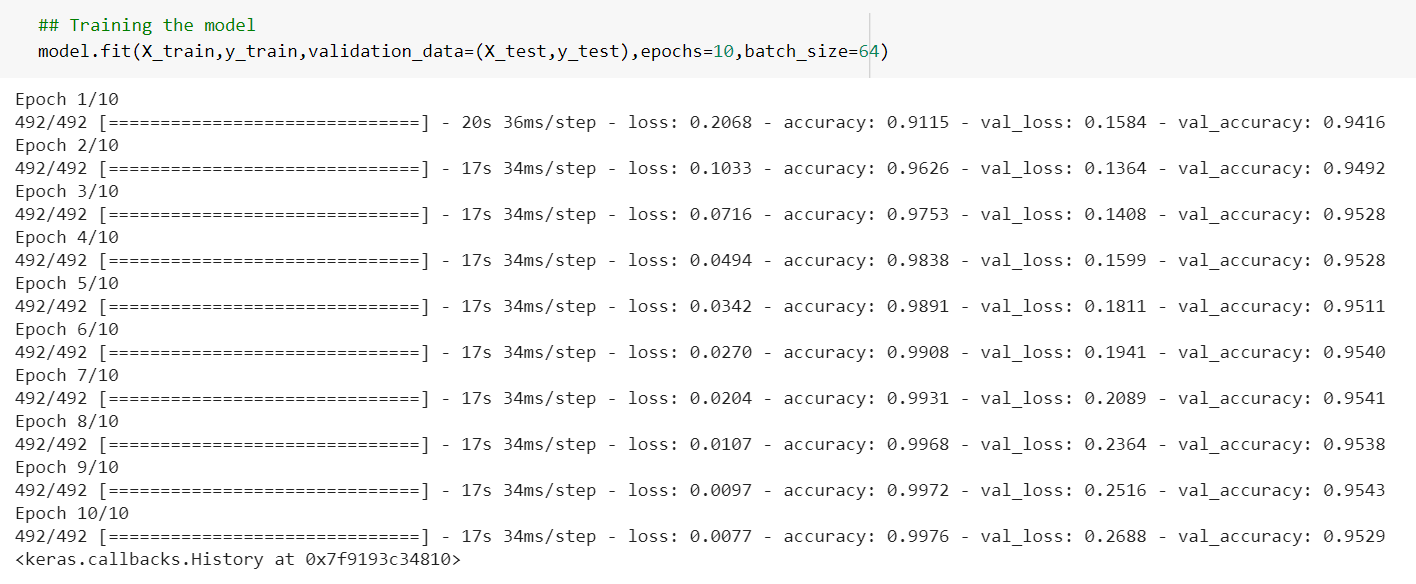
However, LSTM uses a multiple word string to find out the class to which it belongs. This is very helpful while working with Natural language processing. Using appropriate layers of embedding and encoding in LSTM, the model will be able to find out the actual meaning in input string and will give the most accurate output class.

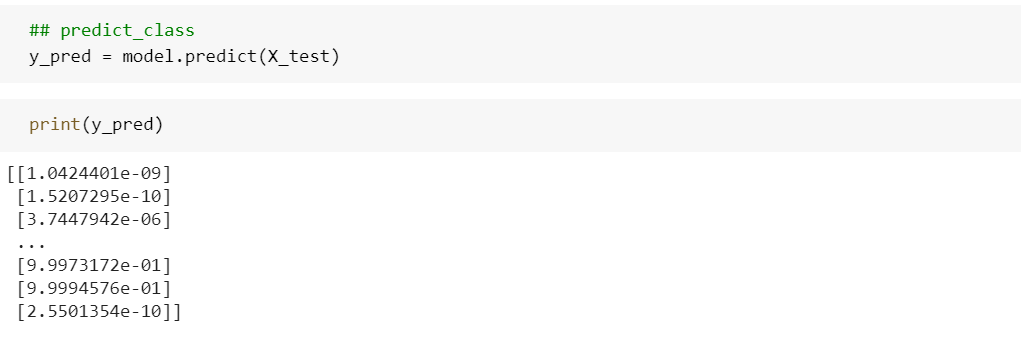


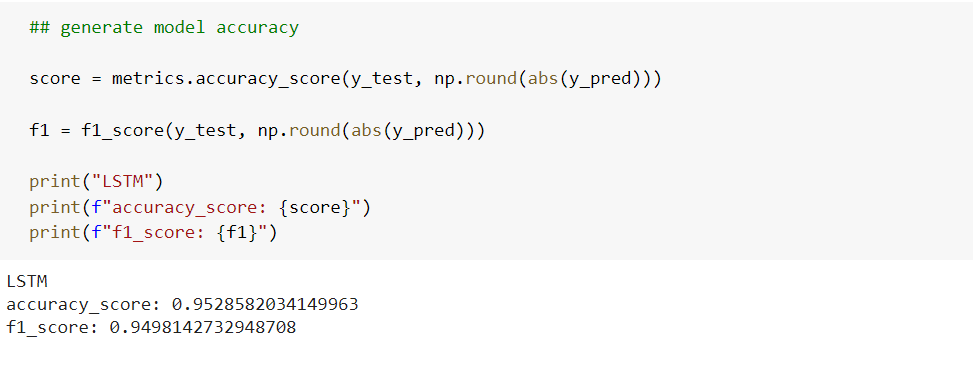


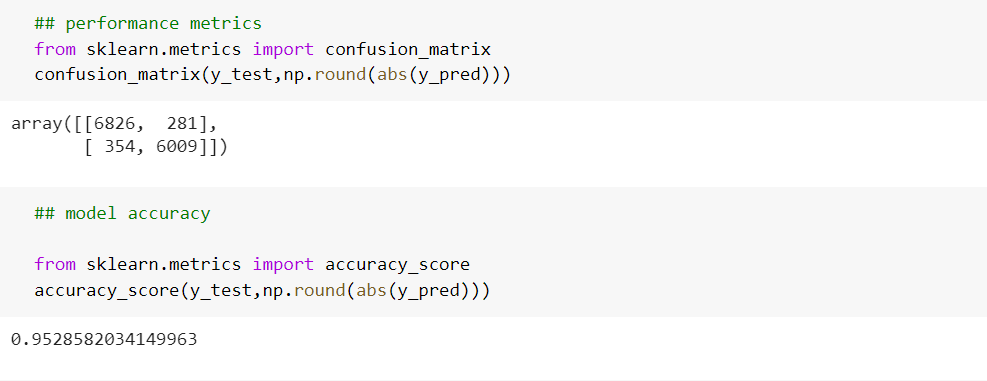












# Conclusion

In this fake news classification task, Passive-Aggressive Classifier gives the best performance. It gives both accuracy and F1-score of 99%.

The fake news challenge is perilous and is spreading rapidly like a wildfire as it becomes easier for information to reach the mass in various flavors. With the help of artificial intelligence, we can control and limit the spread of such misinformation more quickly and efficiently as compared to manual efforts.

Though there has been a big research progress in FID, it is still at the early stage and there are numerous open research issues and promising research directions to be studied, such as model adaptivity/generality to new events, embracing novel machine learning models, explanatory detection models, and so on.

Future planned research efforts involve combing attribution feature extraction with other factors that emerge from the research to produce tools that not only identify potential false content, but influence based content designed to compel a reader or target audience to make inaccurate or altered decisions.

For future improvements, concepts like POS tagging, word2vec and topic modelling can be utilized. These will give the model a lot more depth in terms of feature extraction and fine-tuned classification

# References

1. R. K. Kaliyar, "Fake News Detection Using A Deep Neural Network," 2018 4th International Conference on Computing Communication and Automation (ICCCA), 2018, pp. 1-7, doi: 10.1109/CCAA.2018.8777343.
2. T. Traylor, J. Straub, Gurmeet and N. Snell, "Classifying Fake News Articles Using Natural Language Processing to Identify In-Article Attribution as a Supervised Learning Estimator," 2019 IEEE 13th International Conference on Semantic Computing (ICSC), 2019, pp. 445-449, doi: 10.1109/ICOSC.2019.8665593.
3. <https://www.geeksforgeeks.org/passive-aggressive-classifiers/>
4. <https://www.geeksforgeeks.org/understanding-logistic-regression/>
5. <https://www.analyticsvidhya.com/blog/2021/06/lstm-for-text-classification/>