**Dependencies**

* Numpy: for making numpy arrays
* Panda: for creating data frames and storing data in the data frame
* Re: Regular Expression use for searching the words in a text or paragraph
* Nltk.corpus, stopwords: nltk stands of natural language toolkit. corpus means the body of that particular text which is important, stopwords which doesn’t add much value to a paragraph vertex (eg. The, a)
* Nltk.stem.porter: we perform a function called as stemming so this stemming takes a word and removes the prefix and suffix of that word and returns root word of it
* Sklearn.feature\_extraction.text, TfidVectorizer: for converting text into feature vector. Feature Vectors are nothing but numbers
* Sklearn.model\_selection, train\_test\_split: for spiting our data set into training data and test data.
* Sklearn.linear\_model, LogisticRegression
* Sklearn\_matrics, accuracy\_score

**Preprocessing and Cleaning**

**Data Pre-processing**

Loading the dataset to a pandas Data Frame. Pandas Data Frame loads dataset into a more structured table. We create a new variable name as news\_dataset, where we store our dataset file which is in .csv (Comma Separated Value) filetype.



Let check the numbers of rows and column of the dataset



Output

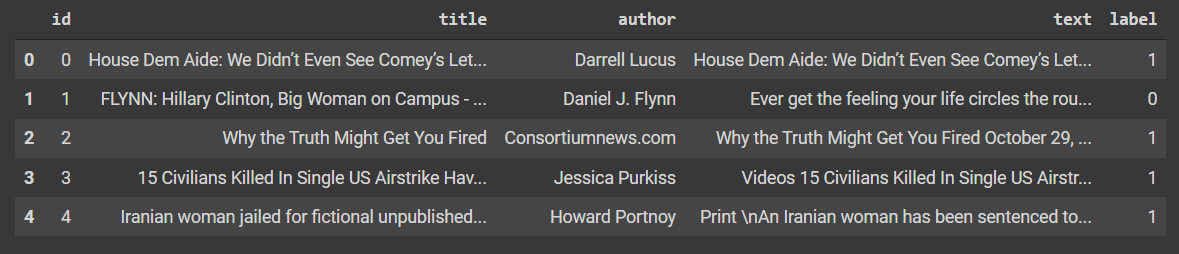


So, we have 20,800 rows and 5 columns which also shows we have 20,800 news articles and 5 features

We are checking first 5 rows of this data frame



Output:



In above dataset we have:

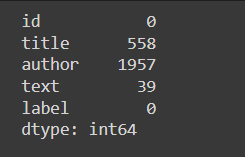
* Id: unique id for a news article
* Title: the title of a news article
* Author: author of the news article
* Text: the text of the article could be incomplete
* Label: a label that marks whether the news article is real or fake. (0: Real News, 1: Fake News)

**Cleaning**

Now we check the missing values in dataset and count total number in each column.



Output:

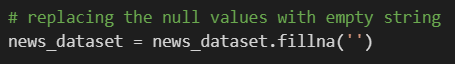


From above output we can see there are no missing values in column “id” and “label”. But in title, author and text columns we have 558, 1957, and 39 missing values respectively out of 20,800 news.

While we are preparing the dataset, we might not get the title of a news or author of a particular news or the text of the particular news. So, that is why we have missing value in our dataset.

We can drop these missing values or we can replace it with null string. If more values are missing, we use some methods like imputation (processing to replace those missing values with appropriate values)

For now, we are replacing the missing values with null string by doing,



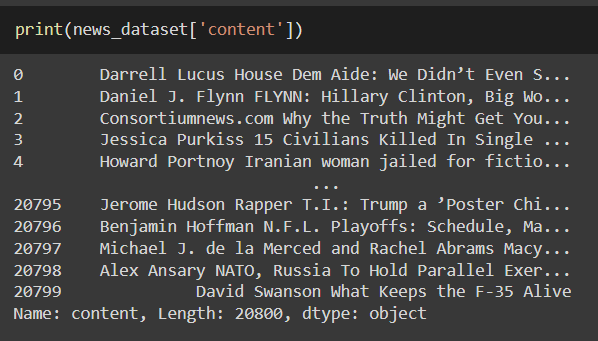
**Implementing Classification**

In our project we are going to include title and author for prediction. Because of that we are going to combine ‘title’ and ‘author’ column together. We are not going to use ‘text’ column because it can be huge paragraph and it takes a lot of time for processing.

Here we create a new column called “content” and store “title” and “author” together.

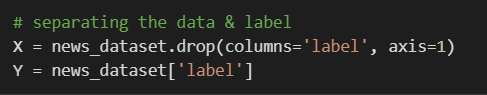


Output



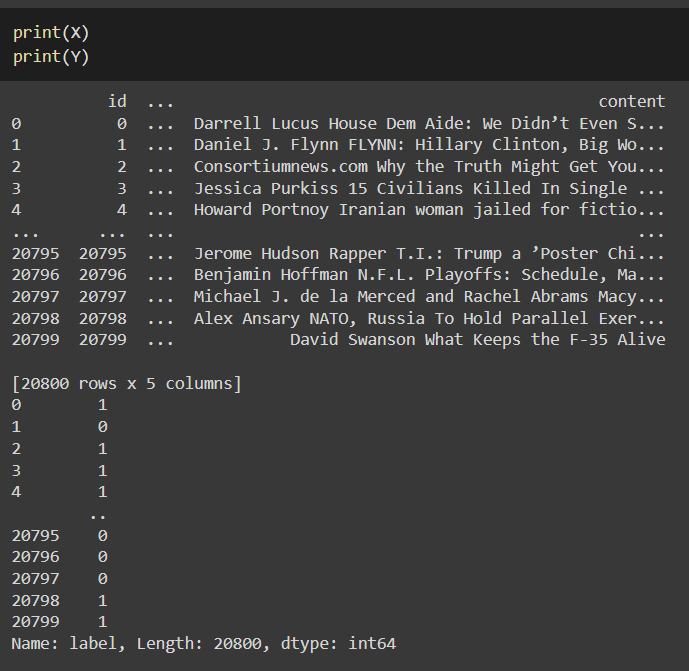
So now we are going to use this “content” column and “label” column to make predictions.

Now we are separating the “content” column and “label” column. In this case, the date is “content” column and label are “label” column.



Here, in variable “X” we have dataset without “label” column and in variable “Y” we have only the “label” column.

Output

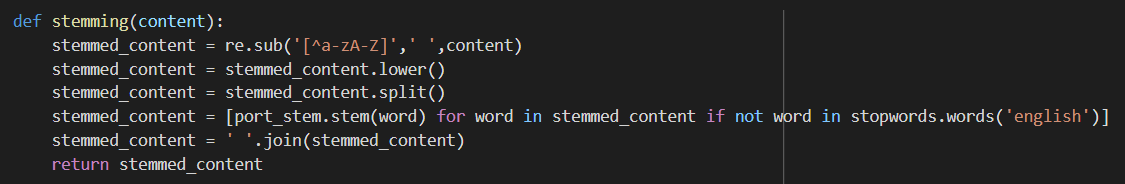


**Stemming**

Stemming is the process of reducing a word to its Root word. Example: actor, actress, acting --> act. This is very important step because we need to reduce words as much as possible to have better performance on our model.



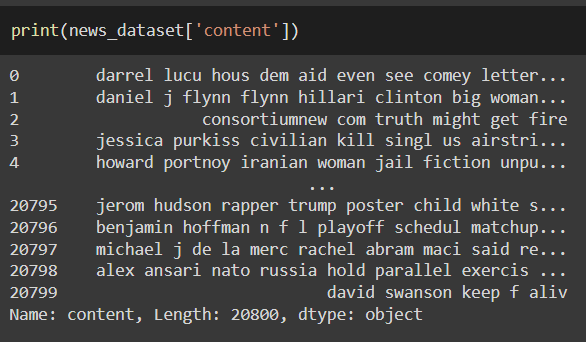
We create a function called as “stemming” which will search and store the alphabets and replace all numbers and special character with empty string (‘’) from our “context” column. Afterward, we convert all the words into lowercase and converted into List using split method. Then we take each word and convert them into there root word (stemming) and if found any stopwords we will remove it. Once we have done all above steps we will join all the words together.



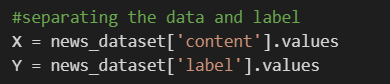
We apply above created ‘stemming’ function to our content column



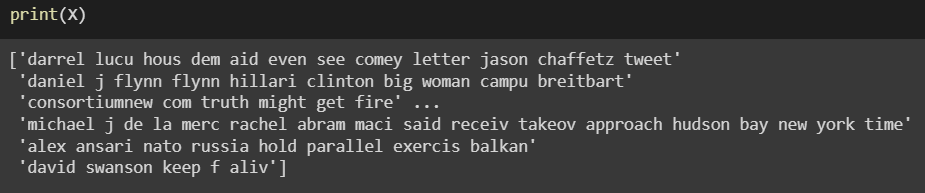
Output

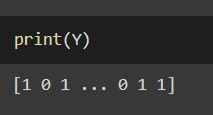


We are separating the “content” and “label” columns in X and Y respectively, which we are going to feed our machine learning model.



Output

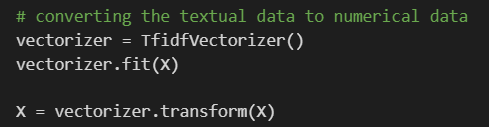




We can see from above output that our “content” values are still in textual form. But computer cannot understand text so that we need to convert all these text into meaningful numbers that computer understands.

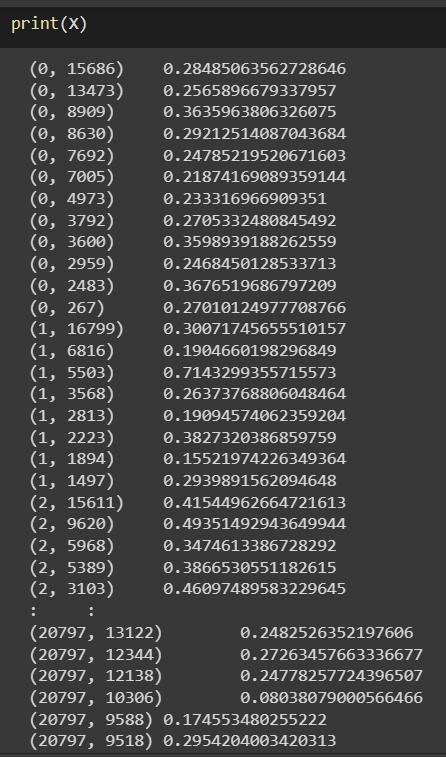
**Clustering**

So, for this we will be using vectorizer function. Which convert text into feature vector. Feature Vectors are nothing but numbers.



Here, we are using special vectorizer function called TfidfVectorizer, where “Tf” stands for Term Frequency and “idf” stands for inverse document frequency. “Tf” basically counts the number of times a particular word is repeating in a document, text, paragraph. Repetition tells the model that it is a very important word and it assign a particular numerical value to that word. “idf” finds those values or word which are repeating many times and it detects that those words are not significant and it reduce its importance value. So, we are fitting the ‘content’ column to TfidfVectorizer function.

Output



We can from above outputs that our data is converted in the numerical form from textual form.

**Splitting the dataset to training & test data**

Now we are splitting our data into two data which are “test” and “strain”. “train” represents training data and “test” represents testing data. We want 80 percentage of data to be training data and 20 percentage of the data to be testing data (test\_size = 0.2). These 20 percentages of our data will be store in X\_test and rest of 80 percentage will be store in X\_train. The label for X\_train data will be stored in Y\_train and label for X\_test data will be stored in Y\_test. Variable Y basically contains 0 and 1 which represents real and fake news respectively. By doing Stratify = Y, we segregated in equal proportion, there will be similar proportion as it was in the original dataset. Random\_state=2 is for reproduce a particular code in same manner for all. For example, while you are preparing this code if you mention 2 here that data will be splitted in a same way as it is splitting for me. It can be any integer value.



**Association Algorithms**

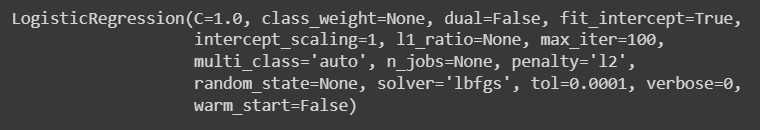
**Training the Model: Logistic Regression**



We now train our model on the basics of obtaining data (X\_train and Y\_train) and it will plot sigmoid function curve using Logistic Regression.



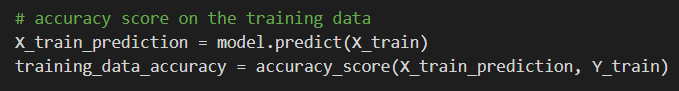
Output



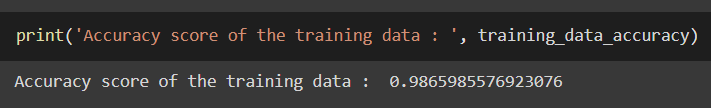
**Data visualization / Evaluation**

As it is trained, we need to find its accuracy and other scores. Here, model will be asked to predict values and model’s prediction will be compared to the original label values.

Let’s check the accuracy score on the training data by making prediction on X\_train data and all the prediction will be stored in X\_train\_prediction variable. For accuracy score, we use predicted data verses original labels.

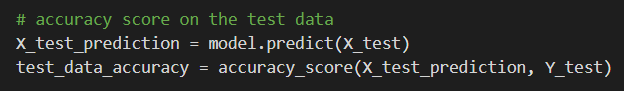


Output

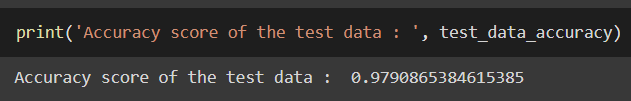


So, our accuracy score is 0.9865985576923076 meaning it is 98 percent accurate. This means that our training data is really good.

Now, let’s check the accuracy of test data which is the real deal.



Output



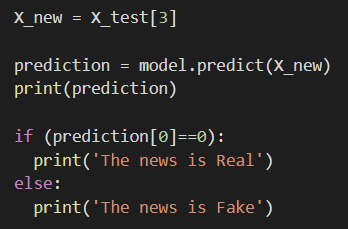
So, our accuracy score is 0.9790865384615385 meaning it is almost 98 percent accurate. This means our model has not overtrained with training data and it is performing very well.

We have successfully train our model and we have also evaluated our model with high accuracy score.

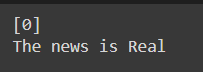
**Making a Predictive System**

We build a new Predictive System where if you give a new news data to our model it should predict whether the news is real or fake.

Here, X\_new is variable were store new news data. Then we will predict whether news is real or fake using our trained model and show us a result.

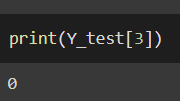


Output



Our model predicts that the new news we input is Real.

While checking the respective column in original label for accuracy I found



Which is also true. So, this means that my model is working fine.

**Conclusion**

As mentioned earlier, Media trust worldwide has dropped by 8% between 2020 and 2021. In 2020, only 29% of US adults said they mostly trust news media. 52% of Americans say they regularly encounter fake news online. 67% of US adults say they’ve come across false information on social media. To reduce that different approach are can be made and of the approach with Machine Learning with Logistic Regression model is the one we choose. This project would make a very persuasive argument for the implementation of a full-scale data mining and analysis.

**References**

1. https://letter.ly/fake-news-statistics/
2. Data Mining: Concepts and Techniques, 3rd ed. Jiawei Han, Micheline Kamber, and Jian Pei. Morgan Kaufmann Series in Data Management Systems Morgan Kaufmann Publishers, July 2011.
3. Introduction to Data Mining, 2nd ed. Pang-Ning Tan, Michael Steinbach, Anuj Karpatne, Vipin Kumar. Pearson Publisher, 2019.
4. Mining of Massive Datasets by Jure Leskovec, Anand Rajaraman, Jeff rey D. Ullman, 2014.
5. https://www.youtube.com/watch?v=nacLBdyG6jE