Project 4

**Biasing News using Prediction model with the help of Logistic Regression**

**OVERVIEW**

In the fast-growing world, online news articles have taken the spun and latest news is popped up each and every second all over the world. With so many news providers, bloggers and websites finding the authenticity of the news has been a big question. In alignment with this scenario in this project we will build a Supervised Model that will predict if the news is the fake or real by biasing it based on the training received using the Logistic Regression Algorithm. A very rare and famous approach of stemming is used in this project to fine tune the training process.

**Software Requirements**

1. Programming Language: Python

2. Environment: Jupyter Notebooks / Google Collab

3. Database: CSV (export type)

4. Operation System: Windows XP or above

5. Libraries Used: Pandas, Seaborn, NLTK, Sklearn, re

1. **Open a New Notebook and import the required libraires and read the csv file**

|  |  |
| --- | --- |
|  | import numpy as np  import pandas as pd  import re  from nltk.corpus import stopwords  from nltk.stem.porter import PorterStemmer  from sklearn.feature\_extraction.text import TfidfVectorizer  from sklearn.model\_selection import train\_test\_split  from sklearn.linear\_model import LogisticRegression  from sklearn.metrics import accuracy\_score  import nltk  nltk.download('stopwords')  # printing the stopwords in English  print(stopwords.words('english')) |

**Description:**

Importing libraries that are per our requirements. Getting all the stop words form nltk.corpus library. **Stopwords** are the English words which does not add much meaning to a sentence. They can safely be ignored without sacrificing the meaning of the sentence. ... Such words are already captured this in corpus named corpus.

1. **Data Pre-processing**

news\_dataset = pd.read\_csv('/content/data.csv')

news\_dataset.shape

news\_dataset.head()

**Output:**

|  | **id** | **title** | **author** | **text** | **label** |
| --- | --- | --- | --- | --- | --- |
| **0** | 0 | House Dem Aide: We Didn’t Even See Comey’s Let... | Darrell Lucus | House Dem Aide: We Didn’t Even See Comey’s Let... | 1 |
| **1** | 1 | FLYNN: Hillary Clinton, Big Woman on Campus - ... | Daniel J. Flynn | Ever get the feeling your life circles the rou... | 0 |
| **2** | 2 | Why the Truth Might Get You Fired | Consortiumnews.com | Why the Truth Might Get You Fired October 29, ... | 1 |
| **3** | 3 | 15 Civilians Killed In Single US Airstrike Hav... | Jessica Purkiss | Videos 15 Civilians Killed In Single US Airstr... | 1 |
| **4** | 4 | Iranian woman jailed for fictional unpublished... | Howard Portnoy | Print \nAn Iranian woman has been sentenced to... | 1 |

# counting the number of missing values in the dataset

news\_dataset.isnull().sum()

**Output :** id 0

title 558

author 1957

text 39

label 0

dtype: int64

news\_dataset = news\_dataset.fillna('')

news\_dataset['content'] = news\_dataset['author']+' '+news\_dataset['title']

print(news\_dataset['content'])

**Output:**   
0 Darrell Lucus House Dem Aide: We Didn’t Even S...

1 Daniel J. Flynn FLYNN: Hillary Clinton, Big Wo...

2 Consortiumnews.com Why the Truth Might Get You...

3 Jessica Purkiss 15 Civilians Killed In Single ...

4 Howard Portnoy Iranian woman jailed for fictio...

...

20795 Jerome Hudson Rapper T.I.: Trump a ’Poster Chi...

20796 Benjamin Hoffman N.F.L. Playoffs: Schedule, Ma...

20797 Michael J. de la Merced and Rachel Abrams Macy...

20798 Alex Ansary NATO, Russia To Hold Parallel Exer...

20799 David Swanson What Keeps the F-35 Alive

Name: content, Length: 20800, dtype: object

X = news\_dataset.drop(columns='label', axis=1)

Y = news\_dataset['label']

print(X)

print(Y)

**Output:**

id ... content

0 0 ... Darrell Lucus House Dem Aide: We Didn’t Even S...

1 1 ... Daniel J. Flynn FLYNN: Hillary Clinton, Big Wo...

2 2 ... Consortiumnews.com Why the Truth Might Get You...

3 3 ... Jessica Purkiss 15 Civilians Killed In Single ...

4 4 ... Howard Portnoy Iranian woman jailed for fictio...

... ... ... ...

20795 20795 ... Jerome Hudson Rapper T.I.: Trump a ’Poster Chi...

20796 20796 ... Benjamin Hoffman N.F.L. Playoffs: Schedule, Ma...

20797 20797 ... Michael J. de la Merced and Rachel Abrams Macy...

20798 20798 ... Alex Ansary NATO, Russia To Hold Parallel Exer...

20799 20799 ... David Swanson What Keeps the F-35 Alive

[20800 rows x 5 columns]

0 1

1 0

2 1

3 1

4 1

..

20795 0

20796 0

20797 0

20798 1

20799 1

Name: label, Length: 20800, dtype: int64

**Description:**

Loading the data to a pandas dataframe and making all the null values to NaN. Because will processing data we don’t need float values. Then dropping the label column in order to use it for testing at later stages.

1. **Using Stemming Process**

|  |  |
| --- | --- |
| port\_stem = PorterStemmer()  def stemming(content):      stemmed\_content = re.sub('[^a-zA-Z]',' ',content)      stemmed\_content = stemmed\_content.lower()      stemmed\_content = stemmed\_content.split()      stemmed\_content = [port\_stem.stem(word) for word in stemmed\_content if not word in stopwords.words('english')]      stemmed\_content = ' '.join(stemmed\_content)      return stemmed\_content  news\_dataset['content'] = news\_dataset['content'].apply(stemming)  print(news\_dataset['content'])  Output:  0 darrel lucu hous dem aid even see comey letter...  1 daniel j flynn flynn hillari clinton big woman...  2 consortiumnew com truth might get fire  3 jessica purkiss civilian kill singl us airstri...  4 howard portnoy iranian woman jail fiction unpu...  ...  20795 jerom hudson rapper trump poster child white s...  20796 benjamin hoffman n f l playoff schedul matchup...  20797 michael j de la merc rachel abram maci said re...  20798 alex ansari nato russia hold parallel exercis ...  20799 david swanson keep f aliv  Name: content, Length: 20800, dtype: object  #separating the data and label  X = news\_dataset['content'].values  Y = news\_dataset['label'].values  print(X)  Output :  ['darrel lucu hous dem aid even see comey letter jason chaffetz tweet'  'daniel j flynn flynn hillari clinton big woman campu breitbart'  'consortiumnew com truth might get fire' ...  'michael j de la merc rachel abram maci said receiv takeov approach hudson bay new york time'  'alex ansari nato russia hold parallel exercis balkan'  'david swanson keep f aliv']  print(Y)  [1 0 1 ... 0 1 1]  # converting the textual data to numerical data  vectorizer = TfidfVectorizer()  vectorizer.fit(X)  X = vectorizer.transform(X)  print(X)  Output:  (0, 15686) 0.28485063562728646  (0, 13473) 0.2565896679337957  (0, 8909) 0.3635963806326075  (0, 8630) 0.29212514087043684  (0, 7692) 0.24785219520671603  (0, 7005) 0.21874169089359144  (0, 4973) 0.233316966909351  (0, 3792) 0.2705332480845492  (0, 3600) 0.3598939188262559  (0, 2959) 0.2468450128533713  (0, 2483) 0.3676519686797209  (0, 267) 0.27010124977708766  (1, 16799) 0.30071745655510157  (1, 6816) 0.1904660198296849  (1, 5503) 0.7143299355715573  (1, 3568) 0.26373768806048464  (1, 2813) 0.19094574062359204  (1, 2223) 0.3827320386859759  (1, 1894) 0.15521974226349364  (1, 1497) 0.2939891562094648  (2, 15611) 0.41544962664721613  (2, 9620) 0.49351492943649944  (2, 5968) 0.3474613386728292  (2, 5389) 0.3866530551182615  (2, 3103) 0.46097489583229645  : :  (20797, 13122) 0.2482526352197606  (20797, 12344) 0.27263457663336677  (20797, 12138) 0.24778257724396507  (20797, 10306) 0.08038079000566466  (20797, 9588) 0.174553480255222  (20797, 9518) 0.2954204003420313  (20797, 8988) 0.36160868928090795  (20797, 8364) 0.22322585870464118  (20797, 7042) 0.21799048897828688  (20797, 3643) 0.21155500613623743  (20797, 1287) 0.33538056804139865  (20797, 699) 0.30685846079762347  (20797, 43) 0.29710241860700626  (20798, 13046) 0.22363267488270608  (20798, 11052) 0.4460515589182236  (20798, 10177) 0.3192496370187028  (20798, 6889) 0.32496285694299426  (20798, 5032) 0.4083701450239529  (20798, 1125) 0.4460515589182236  (20798, 588) 0.3112141524638974  (20798, 350) 0.28446937819072576  (20799, 14852) 0.5677577267055112  (20799, 8036) 0.45983893273780013  (20799, 3623) 0.37927626273066584  (20799, 377) 0.5677577267055112  **Description:**  **Stemming** usually refers to a crude heuristic process that chops off the ends of words in the hope of achieving this **goal** correctly most of the time, and often includes the removal of derivational affixes.  Here in the above function we use regular expression but removing unwanted data other than alphabetical words and thereby removing stopwords and collecting the main keywords and storing them in to a variable.  We perform vectorization another process of extraction by getting Tfid.  **TF-IDF** is a statistical measure that evaluates how relevant a word is to a document in a collection of documents. This is done by multiplying two metrics: how many times a word appears in a document, and the inverse document frequency of the word across a set of documents. Then we use vectorizer.transform it returns number of times a term appeared in the document represented by the row of the matrix   1. **Splitting the dataset to training & test data**   X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, test\_size = 0.2, stratify=Y, random\_state=2)  Training the Model: Logistic Regression  model = LogisticRegression()  model.fit(X\_train, Y\_train)  **Output:**  LogisticRegression(C=1.0, class\_weight=None, dual=False, fit\_intercept=True,  intercept\_scaling=1, l1\_ratio=None, max\_iter=100,  multi\_class='auto', n\_jobs=None, penalty='l2',  random\_state=None, solver='lbfgs', tol=0.0001, verbose=0,  warm\_start=False)  **Description:**  Splitting the data into training dataset and testing data which helps us to find the exact need. By perform logistic regression we do this project.   1. **Evaluating the Model and Testing the Accuracy**   # accuracy score on the training data  X\_train\_prediction = model.predict(X\_train)  training\_data\_accuracy = accuracy\_score(X\_train\_prediction, Y\_train)  print('Accuracy score of the training data : ', training\_data\_accuracy)  **Output :**  Accuracy score of the training data : 0.9865985576923076  # accuracy score on the test data  X\_test\_prediction = model.predict(X\_test)  test\_data\_accuracy = accuracy\_score(X\_test\_prediction, Y\_test)  print('Accuracy score of the test data : ', test\_data\_accuracy)  **Output :**  Accuracy score of the test data : 0.9790865384615385 |  |

**Description:**

Using predict() function enables us to protect the labels of the data values on the basis of the trained model. It returns the labels of the data passed as arguments based upon the trained data obtained from the model. We calculate accuracy of each set using accuracy\_score () function.

1. **Creating a Prediction System**

X\_new = X\_test[3]

prediction = model.predict(X\_new)

print(prediction)

if (prediction[0]==0):

  print('The news is Real')

else:

  print('The news is Fake')

Output :

[0]

The news is Real

**Description:**

By collecting values from the X\_test we predict the values. As per given data set if the prediction is 0 then the news is Real else the new is fake.

We can cross check the prediction value with Y\_test.

|  |
| --- |
|  |

**Conclusion**

Here in this project, we are going to know whether the news is real or fake.

Using stemming approach to get accurate data instead of unwanted data make our work easier. Mainly by using regular expression anywhere in the competitive programming will provide you needed information from the data. Then by using Logistic Regression we will find whether the data given is fake or real by building a proper predictive system.

|  |  |
| --- | --- |
|  |  |