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 libraries 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  Description :  Importing all the required libraries. Here‘re’ library is regular expression library and from nltk (natural language tool kit) which has corpus library containing stop words are imported. Stemming is a process where producing morphological variants of a root/base word and this is performed by importing PorterStemmer library present in nltk. Vectorization is performed by  TfidfVectorizer from sklearn library. train\_test\_split splits arrays or matrices into random train and test subsets**.** [**Logistic Regression**](https://en.wikipedia.org/wiki/Logistic_regression) is a Machine Learning classification algorithm that is used to predict the probability of a categorical dependent variable. accuracy\_score is used to calculate accuracy of the model designed.    nltk.download('stopwords')  # printing the stopwords in English  print(stopwords.words('english')) |

Output:

['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've", "you'll", "you'd", 'your', 'yours', 'yourself', 'yourselves', 'he', 'him', 'his', 'himself', 'she', "she's", 'her', 'hers', 'herself', 'it', "it's", 'its', 'itself', 'they', 'them', 'their', 'theirs', 'themselves', 'what', 'which', 'who', 'whom', 'this', 'that', "that'll", 'these', 'those', 'am', 'is', 'are', 'was', 'were', 'be', 'been', 'being', 'have', 'has', 'had', 'having', 'do', 'does', 'did', 'doing', 'a', 'an', 'the', 'and', 'but', 'if', 'or', 'because', 'as', 'until', 'while', 'of', 'at', 'by', 'for', 'with', 'about', 'against', 'between', 'into', 'through', 'during', 'before', 'after', 'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out', 'on', 'off', 'over', 'under', 'again', 'further', 'then', 'once', 'here', 'there', 'when', 'where', 'why', 'how', 'all', 'any', 'both', 'each', 'few', 'more', 'most', 'other', 'some', 'such', 'no', 'nor', 'not', 'only', 'own', 'same', 'so', 'than', 'too', 'very', 's', 't', 'can', 'will', 'just', 'don', "don't", 'should', "should've", 'now', 'd', 'll', 'm', 'o', 're', 've', 'y', 'ain', 'aren', "aren't", 'couldn', "couldn't", 'didn', "didn't", 'doesn', "doesn't", 'hadn', "hadn't", 'hasn', "hasn't", 'haven', "haven't", 'isn', "isn't", 'ma', 'mightn', "mightn't", 'mustn', "mustn't", 'needn', "needn't", 'shan', "shan't", 'shouldn', "shouldn't", 'wasn', "wasn't", 'weren', "weren't", 'won', "won't", 'wouldn', "wouldn't"]

Description:

All the stop words present in English language are printed. In [computing](https://en.wikipedia.org/wiki/Computing), stop words are words which are filtered out before or after [processing of natural language](https://en.wikipedia.org/wiki/Natural_language_processing) data (text).

1. **Data Pre-processing**

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

news\_dataset.shape

**Output:**

(20800, 5)

Description:

It gives the size of the data frame that is the number of rows and columns present in it. The input data file which is in .csv format is read and stored in a data frame called news\_dataset

news\_dataset.head()

**Output:**

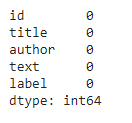


Description:

The top five rows of the data set are printed using .head( ) method. In the input data set we have id, title, author, text and label whose value is either 0 or 1. If the news is true its value is 1 and if its 0 it indicates the news is false.

news\_dataset.isnull().sum()

**Output:**



Description:

It gives number of null values present in each column along with column name and data type of the elements present.

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

Description:

All the null or missing values if present in the data set are replaced by space. Since by default empty value is by default float (0.0) and processes such as vectorization and stemming cannot be performed on float values. Title and author columns are merged and stored in a new column named content separated by a space.

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

We now differentiate the label and data present. The column with name label is dropped and the remaining columns are stored in a new data frame called X and mentioning the axis as 1 if the axis is one the data is dropped column wise if its 0 data is dropped row wise. The column with name label is stored in Y. Since vectorization and stemming cannot be performed on integers.

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  Description:  The port stemmer function in the library is called and stored in a variable called port\_stem. The function named stemming( )is defined whose input parameter is content column has been passed The first parameter to the sub function is the regular expression that finds the pattern to substitute. The second parameter is the new text that you want as a replacement for the old text and the third parameter is the text string on which the substitute operation will be performed separate by a comma(‘,’). Then the content is converted into lower case and split each word and is split with respect to space and next step is to stem each word provided the word is not a stop word and store it in stemmed\_content and finally all the words in stemmed\_content are joined using space and returning stemmed\_content.  Now this function is applied to content column of data set and stored back in the same content column of the data set news\_dataset.  X = news\_dataset['content'].values  Y = news\_dataset['label'].values  print(X)  print(Y)  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']  [1 0 1 ... 0 1 1]  Description:  Only the values of contents column are stored in X variable and values of label in Y variable. Since for vectorization only characters are allowed, id column is not considered.  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:  With vectorization the uniqueness of each word in the whole text is calculated and the textual data is replaced with numerical data. Tfidvectorizer( ) function is called and stored in the variable called vectorizer and the variable containing content is passed.   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)  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 :  The data set is divided into two parts for training and testing which is in the ratio of 8 : 2 with a random state of 2 and the model is trained using logistic regression using .fit( ) method.   1. **Evaluating the Model and Testing the Accuracy**   X\_train\_prediction = model.predict(X\_train)  training\_data\_accuracy = accuracy\_score(X\_train\_prediction, Y\_tra in)  print('Accuracy score of the training data : ', training\_data\_accuracy)    Output:  Accuracy score of the training data : 0.9873798076923077  Description :  The accuracy of model in predicting training data is calculated using accuracy\_score( ) method whose input parameters are training data predicted and the actual data in the training data set. The accuracy of the model in predicting training data is 98.7%  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:

The accuracy of model in predicting testing data is calculated using accuracy\_score( ) method whose input parameters are testing data predicted and the actual data in the testing data set. The accuracy of the model in predicting testing data is 97.9%

1. **Creating a Prediction System**

X\_new = X\_test[5]

prediction = model.predict(X\_new)

print(prediction)

if (prediction[0]==0):

print('The news is Real')

else:

print('The news is Fake')

Output:

[1]

The news is Fake

Description:

We are storing the fifth value of testing data set in X\_new variable and predicting the output using the model which has been trained and if the predicted value is 0 the news is real else it is a fake news.

**7. Creating a Confusion Matrix**

from sklearn.metrics import confusion\_matrix

confusion\_matrix(Y\_test, X\_test\_prediction)

Output:

array([[1952, 79],

[ 15, 2114]])

Description:

A Confusion matrix is an N x N matrix used for evaluating the performance of a classification model, where N is the number of target classes. The matrix compares the actual target values with those predicted by the machine learning model. This gives us a holistic view of how well our classification model is performing and what kinds of errors it is making

**8. Determining Accuracy from Confusion Matrix**

def accuracy(confusion\_matrix):

    diagonal\_sum = confusion\_matrix.trace()

    sum\_of\_all\_elements = confusion\_matrix.sum()

    return (diagonal\_sum / sum\_of\_all\_elements \* 100)

print( 'Accuracy is '+ str(accuracy(confusion\_matrix(Y\_test, X\_test\_prediction)))+'%')

Output:

Accuracy is 97.74038461538461%

Description:

The diagonal elements represent the number of points for which the predicted label is equal to the true label, while off-diagonal elements are those that are mislabelled by the classifier. The higher the diagonal values of the confusion matrix the better, indicating many correct predictions.

|  |
| --- |
|  |

**Conclusion**

**The values have been predicted by the model trained using Logistic regression algorithm, which is one of the most efficient algorithms and the values predicted by the model are not 100% accurate but approximately around 97%.**

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