Fake News Detection Using Machine Learning Algorithm

D.Keerthi(S180532), G.Lakshmi Neeraja(S180746)

Department of Computer Science Engineering

Ragiv Gandhi University of Knowledge and Technologies

Srikakulam,Andhra Pradesh

PROBLEM STATEMENT : CLASSIFYING FAKE NEWS

Forest fires and fake news spread at asimilar speed.Sometimes fake newses can be threatening to life and cause emergency situations.Thus it is important to prevent the misinformation spread across the platforms.Fake news can be identified on the following characterstics.

**Language:**News in the form of text and string.

Since the data is enormous it is not possible to analyse the every single text statement by a person.A technique based on natural language processing is applied to identify the news in realtime.

**Abstract:** In our modern era where the internet is ubiquitous, everyone relies on various online resources for news. Along with the increase in the use of social media platforms like Facebook, Twitter, etc. news spread rapidly among millions of users within a very short span of time. The spread of fake news has far-reaching consequences like the creation of biased opinions to swaying election outcomes for the benefit of certain candidates. Moreover, spammers use appealing news headlines to generate revenue using advertisements via click-baits. In this paper, we aim to perform binary classification of various news articles available online with the help of concepts pertaining to Artificial Intelligence, Natural Language Processing and Machine Learning. We aim to provide the user with the ability to classify the news as fake or real and also check the authenticity of the website publishing the news

Natural Language Processing: Natural language processing (NLP) is the ability of a computer program to understand human language as it is spoken and written -- referred to as natural language. It is a component of artificial intelligence.

**Project:**

To get the accurately classified collection of news as real or fake we have to build a machine learning model.

To deals with the detection of fake or real news, we will develop the project in python with the help of ‘sklearn’, we will use ‘CountVectorizer’ in our news data which we will gather from online media.

After the first step is done, we will initialize the classifier, transform and fit the model. In the end, we will calculate the performance of the model using the appropriate performance matrix/matrices. Once will calculate the performance matrices we will be able to see how well our model performs.

1.Data Analysis:

Here I will explain the dataset.In this python project, we have used the CSV dataset. The dataset contains 20800 rows and 4 columns.

This dataset has four columns

1. **title**: this represents the title of the news.
2. **author**: this represents the name of the author who has written the news.
3. **text**: this column has the news itself.
4. **label**: this is a binary column representing if the news is fake (1) or real (0).

2.Import Libraries:

The very basic data science libraries are sklearn, pandas, NumPy e.t.c and some specific libraries such as transformers.

import pandas as pd

import numpy as np

import seaborn as sns

from sklearn.naive\_bayes import MultinomialNB

3 Read dataset from CSV File

data=pd.read\_csv("train.csv")

data.head()

data.shape

Before proceeding, we need to check whether a null value is present in our dataset or not.

data.isna().sum()

If you have null values present in your dataset then you can drop it. In the code given below is used to remove the null values.

data.dropna(axis=0,inplace=True)

### **4 Data Preprocessing**

### In data processing, we will focus on the text column on this data which actually contains the news part. We will modify this text column to extract more information to make the model more predictable. To extract information from the text column, we will use a library, which we know by the name of ‘nltk’.

Here we will use functionalities ofthe **‘nltk**‘ library named Removing Stopwords, Tokenization, and Lemmatization. So we will see these functionalities one by one with these three examples. Hope you will have a better understanding of extracting information from the text column after this.

#### 4.1 Removing Stopwords:-

These are the words that are used in any language used to connect words or used to declare the tense of sentences. This means that if we use these words in any sentence they do not add much meaning to the context of the sentence so even after removing the stopwords we can understand the context.

!pip install nltk

import nltk

nltk.download('stopwords')

#removing the stopwords from the

from nltk.corpus import stopwords

from nltk.stem.porter import PorterStemmer

#### 4.2 Tokenization:-

Tokenization is the process of breaking text into smaller pieces which we know as tokens.  
Each word, special character, or number in a sentence can be depicted as a token in NLP.

Tokenization is the process of breaking down a piece of code into smaller units called tokens.

ps=PorterStemmer()

review=[ps.stem(word) for word in list if word not in

set(stopwords.words('english')

' '.join(review)

corpus=[]

for i in range(0,len(data)):

  review=re.sub('[^a-zA-Z]',' ',data['title'][i])

  review=review.lower()

  list=review.split()

  review=[ps.stem(word) for word in list if not word in set(stopwords.words('english'))]

  corpus.append(' '.join(review))

### **5. VECTORIZATION**

Vectorization is a methodology in NLP to map words or phrases from vocabulary to a corresponding vector of real numbers which is used to find word predictions, word similarities/semantics.To make documents’ corpora more relatable for computers, they must first be converted into some numerical structure. There are few techniques that are used to achieve this such as ‘Bag of Words’.Here, we are using vectorizer objects provided by Scikit-Learn which are quite reliable right out of the box.

 #bag of words model

from sklearn.feature\_extraction.text import CountVectorizer

cv=CountVectorizer()

x=cv.fit\_transform(corpus).toarray()

x.shape

x[0]

y=data["label"]

y.shape

Count Vectorizer: The most straightforward one, it counts the number of times a token shows up in the document and uses this value as its weight.

### 6. MODELING:

After Vectorization, we split the data into test and train data.

#training set and test set spliting

from sklearn.model\_selection import train\_test\_split

x\_train,x\_test,y\_train,y\_test=train\_test\_split(x,y,test\_size=0.25,random\_state=0)

I fit Naïve-Bayes ML model to the data,

#### Naive-Baye:

from sklearn.naive\_bayes import MultinomialNB

classifier=MultinomialNB()

classifier.fit(x\_train,y\_train)

y\_pred=classifier.predict(x\_test)

y\_pred

accuracy\_score(y\_test,y\_pred)

Output: Accuracy-90.2%

After that, predicted on the test set from the CountVectorizer and calculated the accuracy with accuracy\_score() from sklearn. metrics.

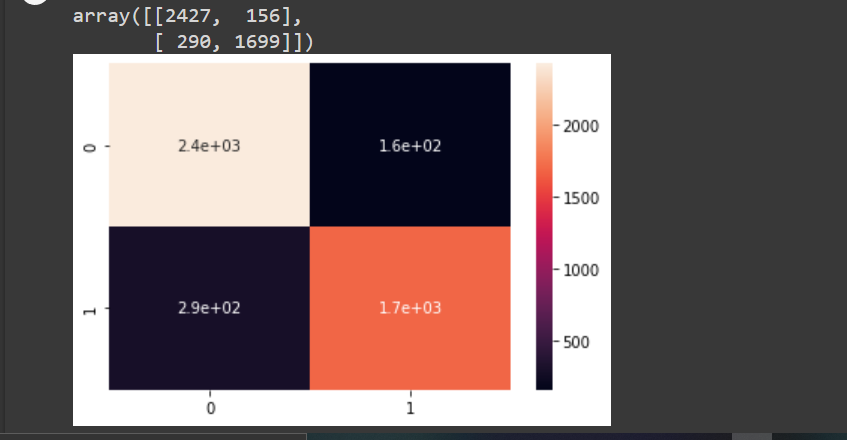
from sklearn.metrics import confusion\_matrix,accuracy\_score

cm=confusion\_matrix(y\_test,y\_pred)

sns.heatmap(cm,annot=True)

cm

Output:

  
Conclusion:

The Naïve-Bayes classifier performed the best here and gave an accuracy of 90.2%.

We print a confusion matrix to gain insight into the number of false and true negatives and positives

Fake news detection techniques can be divided into those based on style and those based on content, or fact-checking. Too often it is assumed that bad style (bad spelling, bad punctuation, limited vocabulary, using terms of abuse, ungrammaticality, etc.) is a safe indicator of fake news.

More than ever, this is a case where the machine’s opinion must be backed up by clear and fully verifiable indications for the basis of its decision, in terms of the facts checked and the authority by which the truth of each fact was determined.

|  |
| --- |
| Project By  G.Lakshmi Neeraja(S180746)  D.Keerthi (S180532) |