Coincent

Artificial Intelligence Project

Title- News Classification using Natural Language Processing

Name – Fanindra Saini
Domain – Artificial Intelligence with Python
Email – fanin.s.pbl@gmail.com

Abstract

This Project Classifies News based on Natural Language Processing using Supervised Learning. This NLP program identifies wheather the news is genuine or not (True or False) by identifying frequency of words found in the news text.

Objective

The Project aims at building a model which classfies wheather the news given is genuine news or false news by using NLP.

Methodologies and Concepts used

We have a dataset that contains true.csv and fake.csv files.
The Project uses following concepts-

- **Supervised Learning -** It is the Process of machine Learning that trains under supervision or we can say that it is trained by providing both the x value and y or target values.
  The goal is to approximate the mapping funtion so well that when we have new data(x) the model will be able to predict the target value(y) accurately.

- **Text Preprocessing -** It is the process of transforming text into a clean and consistent format that can then be fed into a model for further analysis and learning.
  It involves three Stages -
  1. Tokenization - The tokenization stage involves converting a sentence into a stream of words, also called “tokens.” we are using “punkt” tokenizer for our system.
  2. Stemming - the process of converting all words to their base form, or stem
  3. Stopword Removal – A Stopword is a commonly used word like is,the,a,an,etc that are needed to be ignored.

- **Vectorization -**
  it is a technique used to convert textual data to numerical format. Using vectorization, a matrix is created where each column represents a feature and each row represents an indevidual review.

- **Classification Algorithms-**
  we will use 2 classification algorithms.
  1. Logistic Regresion – It is a Predictive analysis algorithm based on the concept of probability.
  2. Passive Agressive Classification – It is the Incremental learning Algorithm. Passive means if the prediction is correct do not make any changes to the model and Agressive means if the prediction is incorrect, make changes to the model

NLTK (Natural language toolkit)-
it is a powerfull tool to preprocess text data for further analysis with model for instances. It supports tasks such as classification, Stemming, taging, parsing, Semantic Reasoning, and tokenization in python.

Code

```
from sklearn.datasets import fetch_openml
import nltk
import pandas as pd

#nltk.download("punkt")
```

#Importing Dataset

```
fake_df=pd.read_csv('Dataset/Fake.csv')
true_df=pd.read_csv('Dataset/True.csv')
print(fake_df.columns)
print(true_df.columns)
```

#Assigning target values

```
fake_df["genuineness"]=0
true_df["genuineness"]=1

data=pd.concat([fake_df,true_df],axis=0)
data=data.reset_index(drop=True)
data=data.drop(['title','subject','date'],axis=1)
```

#Tokeniztion

```
from nltk.tokenize import word_tokenize
data['text']=data['text'].apply(word_tokenize)
```

#Stemming

```
from nltk.stem.snowball import SnowballStemmer
sb=SnowballStemmer("english",ignore_stopwords=False)
def stem_it(text):
    return [sb.stem(word) for word in text]
data['text']=data['text'].apply(stem_it)
```

#Spliting Data

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(data['text'],
data['genuineness'],test_size=0.25)
```

#Feature Extraction

```
from sklearn.feature_extraction.text import TfidfVectorizer
tfidf=TfidfVectorizer(max_df=0.7)
tfidf_train=tfidf.fit_transform(x_train)
tfidf_test=tfidf.fit_transform(x_test)

from sklearn.metrics import accuracy_score
```

#Logistic Regression

```
from sklearn.linear_model import LogisticRegression
model1=LogisticRegression(max_iter=900)
model1.fit(tfidf_train,y_train)
```

#Passive Aggressive Classification Algorithm

```
from sklearn.linear_model import PassiveAggressiveClassifier
model2=PassiveAggressiveClassifier(max_iter=100)
model2.fit(tfidf_train,y_train)
pred2=model2.predict(tfidf_test)
```

#Prediction Result

```
cr1=accuracy_score(y_test,pred1)
cr2=accuracy_score(y_test,pred2)
print("using LogisticRegression : ",cr1)
print("using PassiveAgressiveClassifier :",cr2)
```

ScreenShots

In [1]:
```
import nltk
```

In [2]:
```
#nltk.download('punkt')
```

In [3]:
```
import pandas as pd
```

In [4]:
```
fake=pd.read_csv("Dataset/Fake.csv")
true=pd.read_csv("Dataset/True.csv")
```

In [5]:
```
display(fake.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 23481 entries, 0 to 23480
Data columns (total 4 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   title    23481 non-null  object
 1   text     23481 non-null  object
 2   subject  23481 non-null  object
 3   date     23481 non-null  object
dtypes: object(4)
memory usage: 733.9+ KB

None
```

In [6]:
```
display(true.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21417 entries, 0 to 21416
Data columns (total 4 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   title    21417 non-null  object
 1   text     21417 non-null  object
 2   subject  21417 non-null  object
 3   date     21417 non-null  object
dtypes: object(4)
memory usage: 669.4+ KB

None
```

In [7]:
```
display(fake.subject.value_counts())
```

```
News               9050
politics           6841
left-news          4459
Government News    1570
US_News             783
Middle-east         778
```

```
News               9050
politics           6841
left-news          4459
Government News    1570
US_News             783
Middle-east         778
Name: subject, dtype: int64
```

In [8]:
```
fake['target']=0
true['target']=1
```

In [9]:
```
display(fake.head())
```

	title	text	subject	date	target
0	Donald Trump Sends Out Embarrassing New Year'...	Donald Trump just couldn t wish all Americans ...	News	December 31, 2017	0
1	Drunk Bragging Trump Staffer Started Russian ...	House Intelligence Committee Chairman Devin Nu...	News	December 31, 2017	0
2	Sheriff David Clarke Becomes An Internet Joke...	On Friday, it was revealed that former Milwauk...	News	December 30, 2017	0
3	Trump Is So Obsessed He Even Has Obama's Name...	On Christmas day, Donald Trump announced that ...	News	December 29, 2017	0
4	Pope Francis Just Called Out Donald Trump Dur...	Pope Francis used his annual Christmas Day mes...	News	December 25, 2017	0

In [10]:
```
data=pd.concat([fake,true],axis=0)
date=data.reset_index(drop=True)
```

In [11]:
```
data=data.drop(['subject','date','title'],axis=1)
```

In [12]:
```
print(data.columns)
```

```
Index(['text', 'target'], dtype='object')
```

In [13]:
```
from nltk.tokenize import word_tokenize
```

In [14]:
```
data['text']=data['text'].apply(word_tokenize)
```

Stemming

In [15]:
```
print(data.head(10))
```

```
                                                text  target
0  [Donald, Trump, just, couldn, t, wish, all, Am...       0
1  [House, Intelligence, Committee, Chairman, Dev...       0
2  [On, Friday, ,, it, was, revealed, that, forme...       0
```

```
8  [Many, people, have, raised, the, alarm, regar...       0
9  [Just, when, you, might, have, thought, we, d,...       0
```

```
In [16]: from nltk.stem.snowball import SnowballStemmer
         porter=SnowballStemmer('english')
```

```
In [17]: def stem_it(text):
             return [porter.stem(word) for word in text]
```

```
In [18]: data['text']=data['text'].apply(stem_it)
```

```
In [19]: print(data.head(10))
```

```
                                                text  target
0  [donald, trump, just, couldn, t, wish, all, am...       0
1  [hous, intellig, committe, chairman, devin, nu...       0
2  [on, friday, ,, it, was, reveal, that, former,...       0
3  [on, christma, day, ,, donald, trump, announc,...       0
4  [pope, franci, use, his, annual, christma, day...       0
5  [the, number, of, case, of, cop, brutal, and, ...       0
6  [donald, trump, spent, a, good, portion, of, h...       0
7  [in, the, wake, of, yet, anoth, court, decis, ...       0
8  [mani, peopl, have, rais, the, alarm, regard, ...       0
9  [just, when, you, might, have, thought, we, d,...       0
```

```
In [ ]:
```

```
In [20]: def stop_it(t):
             dt=[word for word in t if len(word)>2]
             return dt
```

```
In [21]: data['text']=data['text'].apply(stop_it)
```

```
In [22]: print(data.head(10))
```

```
                                                text  target
0  [donald, trump, just, couldn, wish, all, ameri...       0
1  [hous, intellig, committe, chairman, devin, nu...       0
2  [friday, was, reveal, that, former, milwauke, ...       0
3  [christma, day, donald, trump, announc, that, ...       0
4  [pope, franci, use, his, annual, christma, day...       0
5  [the, number, case, cop, brutal, and, kill, pe...       0
6  [donald, trump, spent, good, portion, his, day...       0
7  [the, wake, yet, anoth, court, decis, that, de...       0
8  [mani, peopl, have, rais, the, alarm, regard, ...       0
9  [just, when, you, might, have, thought, get, b...       0
```

```
(33672, 88704)        0.02029318699659072
(33672, 88679)        0.02149354353063991
(33672, 56184)        0.06313350632068794
(33672, 15758)        0.03184930276229931
(33672, 7380) 0.03824636792195158
(33672, 46835)        0.05830560243910609
(33672, 57605)        0.056846212468271706
(33672, 88636)        0.04422215177168904
(33672, 35811)        0.025389108517263295
(33672, 89025)        0.03913888579638787
(33672, 83265)        0.23557877667757735
(33672, 26807)        0.06531815063617692
(33672, 65485)        0.01887814444135543
```

```
In [ ]:
```

LogisticRegression

```
In [29]: from sklearn.linear_model import LogisticRegression
         from sklearn.tree import DecisionTreeClassifier
         from sklearn.metrics import accuracy_score
```

```
In [30]: model_1=LogisticRegression(max_iter=900)
         model_1.fit(tfidf_train,y_train)
         pred_1=model_1.predict(tfidf_test)
         cr1=accuracy_score(y_test,pred_1)
         print(cr1*100)
```

```
98.86859688195992
```

PassiveAggressiveClassifier

```
In [31]: from sklearn.linear_model import PassiveAggressiveClassifier
         model=PassiveAggressiveClassifier(max_iter=50)
         model.fit(tfidf_train,y_train)
```

```
Out[31]: PassiveAggressiveClassifier(max_iter=50)
```

```
In [32]: y_pred=model.predict(tfidf_test)
         accscore=accuracy_score(y_test,y_pred)
         print('The accuracy of prediction is ',accscore*100)
```

```
The accuracy of prediction is  99.59020044543429
```

Conclusion

The project is able to calcuate results with 98.86859688195992% accuracy using Logistic Regression and 99.59020044543429% accuracy with Passive Aggressive classification.

Hence, we have successfully developed a basic news classifier with good accuracy.