FLIP ROBO

Fake News Detection Project

NLP Project

Submitted by:

Sai Kumar M

ACKNOWLEDGMENT

A Fake News Detection project using NLP. Data is provided by the organizer . Referred to kagal and GitHub for better understanding and to get advanced techniques. Referred Youtube for CNN understanding and took help from these Official websites like w3schools and others for base knowledge.

This Project comes up with the applications of NLP (Natural Language Processing) techniques for detecting the 'fake news', that is, misleading news stories that come from non-reputable sources. Only by building a model based on a count vectorizer (using word tallies) or a (Term Frequency Inverse Document Frequency) tfidf matrix, can only get you so far. But these models do not consider the important qualities like word ordering and context. It is very possible that two articles that are similar in their word count will be completely different in their meaning. The data science community has responded by taking actions against the problem. There is a Kaggle competition called the “Fake News Challenge” and Facebook is employing AI to filter fake news stories out of users’ feeds. Combatting the fake news is a classic text classification project with a straightforward proposition. Is it possible for you to build a model that can differentiate between “Real “news and “Fake” news? A proposed work on assembling a dataset of both fake and real news and employing a few ML models in order to create a model to classify an article into fake or real based on its words and phrases.

INTRODUCTION

● Business Problem Framing

The authenticity of Information has become a longstanding issue affecting businesses and society, both for printed and digital media. On social networks, the reach and effects of information spread occur at such a fast pace and so amplified that distorted, inaccurate, or false information acquires a tremendous potential to cause real-world impacts, within minutes, for millions of users.

Recently, several public concerns about this problem and some approaches to mitigate the problem were expressed.

● Conceptual Background of the Domain Problem

In the world of Artificial Intelligence and advancement in technologies. The authenticity of Information has become a longstanding issue affecting businesses and society, both for printed and digital media. On social networks, the reach and effects of information spread occur at such a fast pace and so amplified that distorted, inaccurate, or false information acquires a tremendous potential to cause real-world impacts, within minutes, for millions of users. Recently, several public concerns about this problem and some approaches to mitigate the problem were expressed.

● Motivation for the Problem Undertaken

This model will then be used by huge companies for their advancement and here it will be easy to find whether the news is fake or genuine which will be very helpful for companies for their development.

Analytical Problem Framing

- ## Data Sources and their formats

In this project, you are given a dataset in the fake-news_data.zip folder. The folder contains a CSV file train_news.csv and you have to use the train_news.csv data to build a model to predict whether a news is fake or not fake. You have to try out different models on the dataset, evaluate their performance, and finally report the best model you got on the data and its performance.

Data- Description:

There are 6 columns in the dataset provided to you.

The description of each of the column is given below:

“id”: Unique id of each news article

“headline”: It is the title of the news.

“news”: It contains the full text of the news article

“Unnamed:0”: It is a serial number

“written_by”: It represents the author of the news article

“label”: It tells whether the news is fake (1) or not fake (0).

- ## Data Preprocessing Done

In this process following are carried on

1. data cleaning
2. cleaning text
3. removing stop words
4. applying stemming

- ## Data Inputs- Logic- Output Relationships

The NLP model is best for Fake news detection, Here for this model we will provide news (text) as input and we prepare a model to recognize whether the news is fake or not.

- ## Hardware and Software Requirements and Tools Used

Hardware

1. RAM – 128 GB DDR4 2133 MHz
2. 2 TB Hard Disk (7200 RPM) + 512 GB SSD
3. GPU – NVidia TitanX Pascal (12 GB VRAM)
4. Intel Heatsink to keep temperature under control

Software

1. Jupyter Notebook

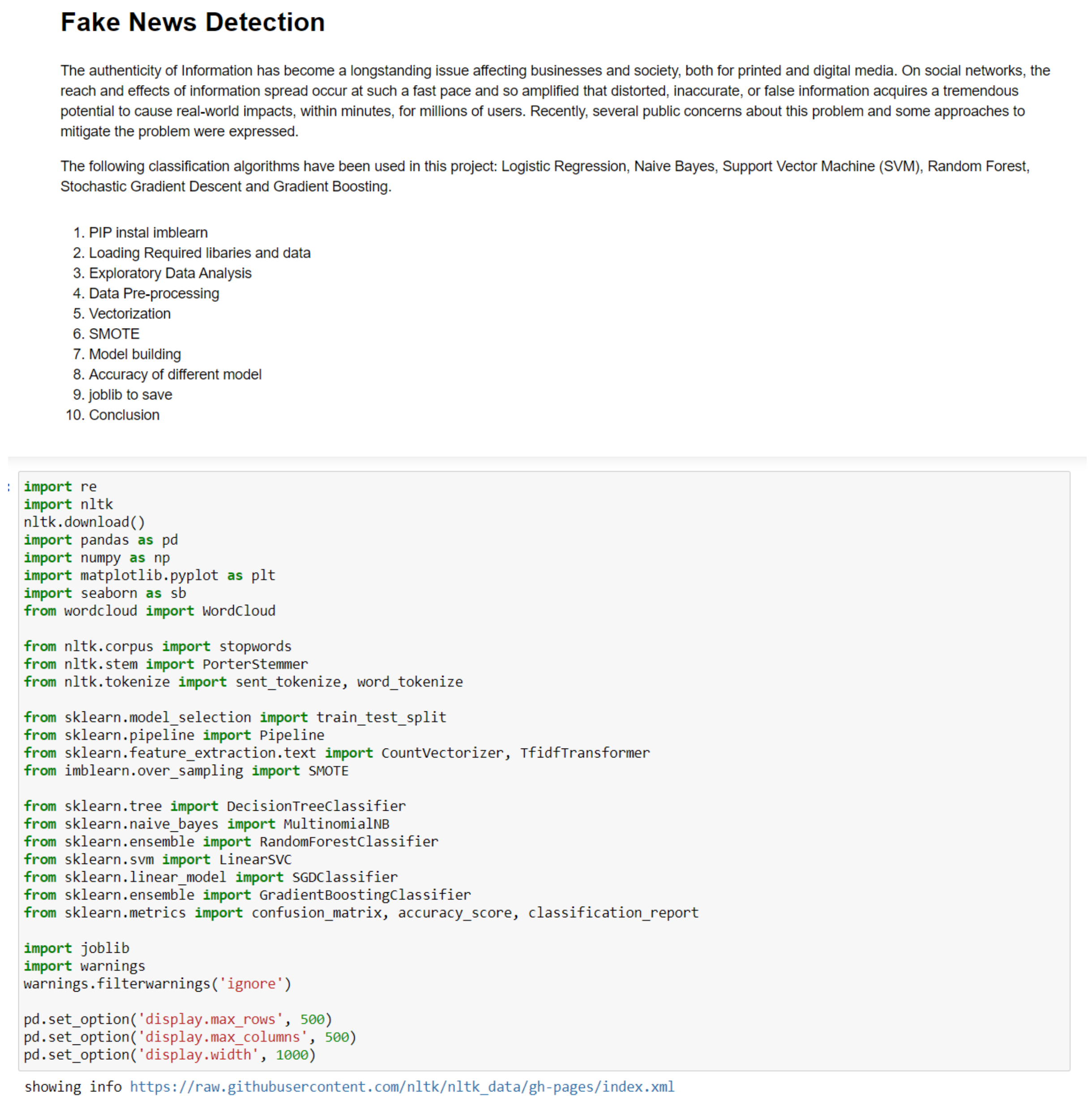
Model/s Development and Evaluation

- Testing of Identified Approaches (Algorithms)

  Algorithms used for the training and testing

  X_train, X_test, y_train, y_test = train_test_split(X_vec, y, test_size=0.23, random_state = 0)

- Run and Evaluate selected models, Key Metrics for success in solving problem under consideration and Visualizations of the complete model with snapshot.



Fake News Detection

The authenticity of Information has become a longstanding issue affecting businesses and society, both for printed and digital media. On social networks, the reach and effects of information spread occur at such a fast pace and so amplified that distorted, inaccurate, or false information acquires a tremendous potential to cause real-world impacts, within minutes, for millions of users. Recently, several public concerns about this problem and some approaches to mitigate the problem were expressed.

The following classification algorithms have been used in this project: Logistic Regression, Naive Bayes, Support Vector Machine (SVM), Random Forest, Stochastic Gradient Descent and Gradient Boosting.

1. PIP instal imblearn
2. Loading Required libaries and data
3. Exploratory Data Analysis
4. Data Pre-processing
5. Vectorization
6. SMOTE
7. Model building
8. Accuracy of different model
9. joblib to save
10. Conclusion

```python
import re
import nltk
nltk.download()
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sb
from wordcloud import WordCloud

from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from nltk.tokenize import sent_tokenize, word_tokenize

from sklearn.model_selection import train_test_split
from sklearn.pipeline import Pipeline
from sklearn.feature_extraction.text import CountVectorizer, TfidfTransformer
from imblearn.over_sampling import SMOTE

from sklearn.tree import DecisionTreeClassifier
from sklearn.naive_bayes import MultinomialNB
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import LinearSVC
from sklearn.linear_model import SGDClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.metrics import confusion_matrix, accuracy_score, classification_report

import joblib
import warnings
warnings.filterwarnings('ignore')

pd.set_option('display.max_rows', 500)
pd.set_option('display.max_columns', 500)
pd.set_option('display.width', 1000)
```

```
showing info https://raw.githubusercontent.com/nltk/nltk_data/gh-pages/index.xml
```

```python
# getting csv formate data

data=pd.read_csv('train_news.csv')
data
```

	Unnamed: 0	id	headline	written_by	news	label
0	0	9653	Ethics Questions Dogged Agriculture Nominee as...	Eric Lipton and Steve Eder	WASHINGTON — In Sonny Perdue's telling, Geo...	0
1	1	10041	U.S. Must Dig Deep to Stop Argentina's Lionel ...	David Waldstein	HOUSTON — Venezuela had a plan. It was a ta...	0
2	2	19113	Cotton to House: 'Do Not Walk the Plank and Vo...	Pam Key	Sunday on ABC's "This Week," while discussing ...	0
3	3	6868	Paul LePage, Besieged Maine Governor, Sends Co...	Jess Bidgood	AUGUSTA, Me. — The beleaguered Republican g...	0
4	4	7596	A Digital 9/11 If Trump Wins	Finian Cunningham	Finian Cunningham has written extensively on...	1
...	...	...	...	...	...	...
20795	20795	5671	NaN	NeverSurrender	No, you'll be a dog licking of the vomit of yo...	1
20796	20796	14831	Albert Pike and the European Migrant Crisis	Rixon Stewart	By Rixon Stewart on November 5, 2016 Rixon Ste...	1
20797	20797	18142	Dakota Access Caught Infiltrating Protests to ...	Eddy Lavine	posted by Eddie You know the Dakota Access Pip...	1
20798	20798	12139	How to Stretch the Summer Solstice - The New Y...	Alison S. Cohn	It's officially summer, and the Society Boutiq...	0
20799	20799	15660	Emory University to Pay for '100 Percent' of U...	Tom Ciccotta	Emory University in Atlanta, Georgia, has anno...	0

20800 rows × 6 columns

```python
# getting csv formate data

data=pd.read_csv('train_news.csv')
data
```

```python
# getting the unique values in a column, total number of unique values in a column

data.nunique()
```

```python
# finding null values, each column gives out total number of null values of that column

print(data.isnull().sum())
print()

# finding percentage of missing values in each column

print(data.isnull().sum()/len(data)*100 )
```

```python
# getting information about each column which gives null value, count and data type

data.info()
```

```python
#used to view some basic statistical details like percentile, mean, std and so on

data['label'].describe()
```

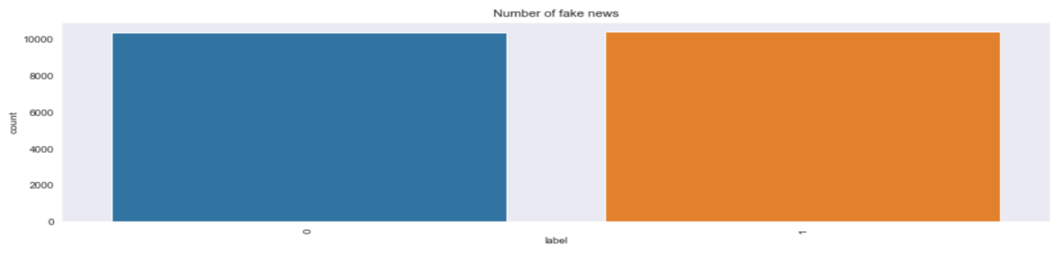
```python
data['news'][0]
```

Exploratory Data Analysis

Exploratory Data Analysis

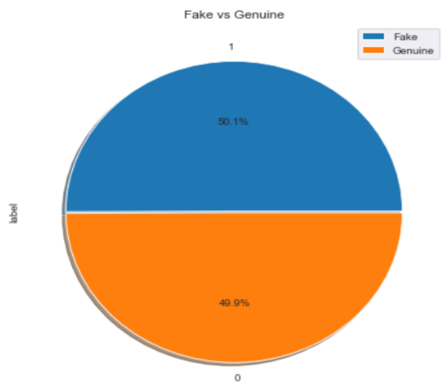
```
plt.figure(figsize=(16,4))
plt.xticks(rotation=90)
sb.set_style("dark")
print(data['label'].value_counts()/len(data)*100)
sb.countplot(x='label',data=data)
plt.xlabel('label')
plt.title('Number of fake news');
```

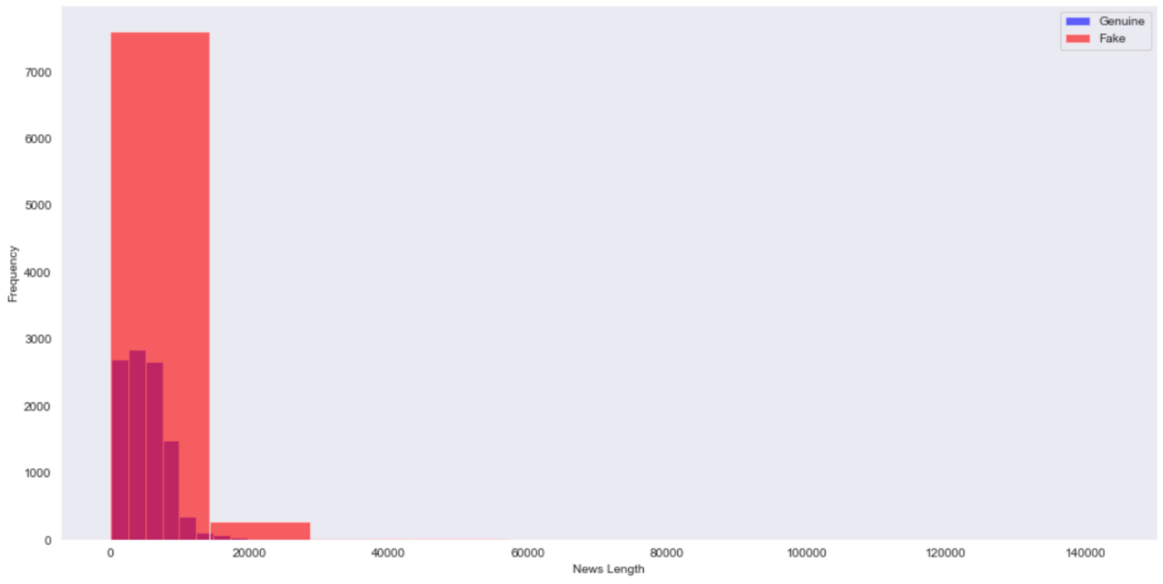
```
1    50.0625
0    49.9375
Name: label, dtype: float64
```



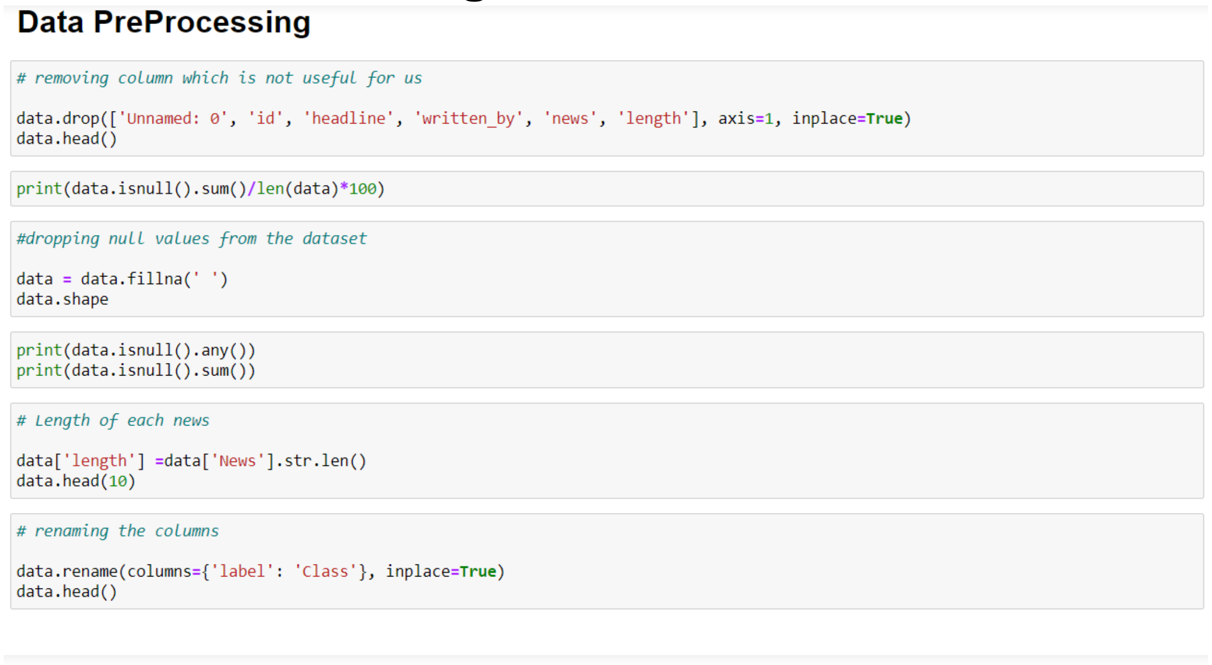
Representing of spam with pie chart

```
data["label"].value_counts().plot(kind = 'pie',
                                  explode=[0, 0.01],
                                  figsize=(8, 7),
                                  autopct='%1.1f%%',
                                  shadow=True)
plt.title("Fake vs Genuine")
plt.legend(["Fake", "Genuine"])
plt.show()
```





- ## Data PreProcessing



Data PreProcessing

```python
# removing column which is not useful for us

data.drop(['Unnamed: 0', 'id', 'headline', 'written_by', 'news', 'length'], axis=1, inplace=True)
data.head()
```

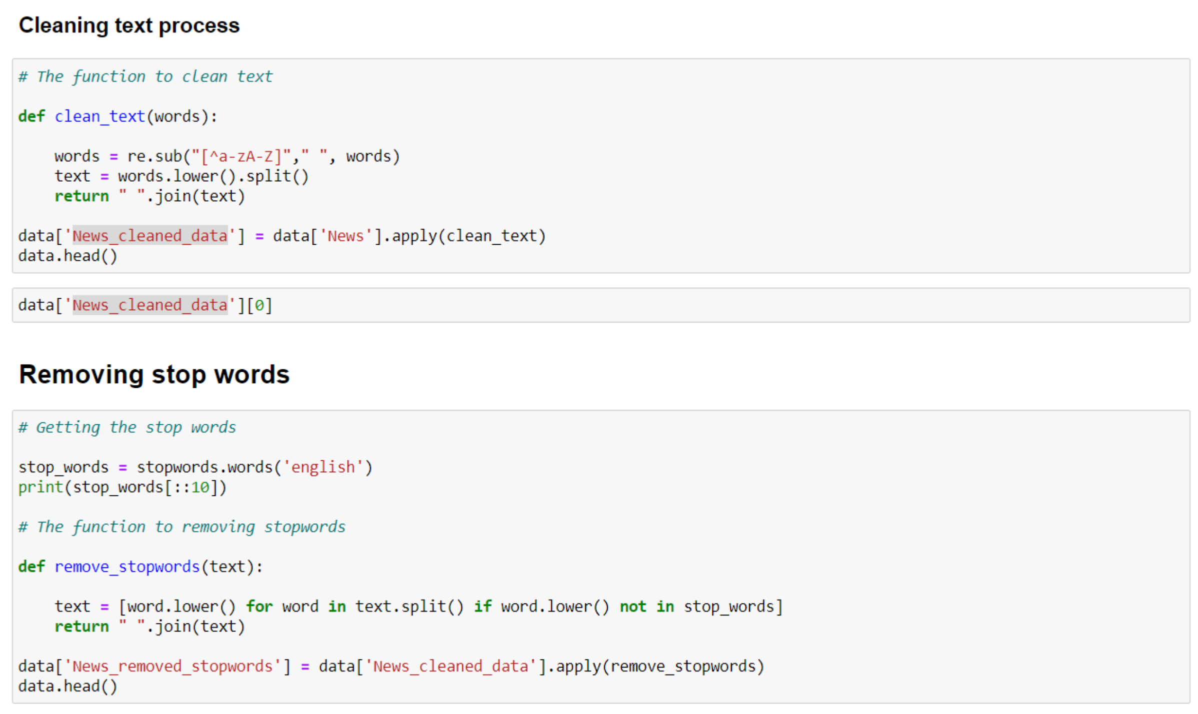
```python
print(data.isnull().sum()/len(data)*100)
```

```python
#dropping null values from the dataset

data = data.fillna(' ')
data.shape
```

```python
print(data.isnull().any())
print(data.isnull().sum())
```

```python
# Length of each news

data['length'] =data['News'].str.len()
data.head(10)
```

```python
# renaming the columns

data.rename(columns={'label': 'Class'}, inplace=True)
data.head()
```



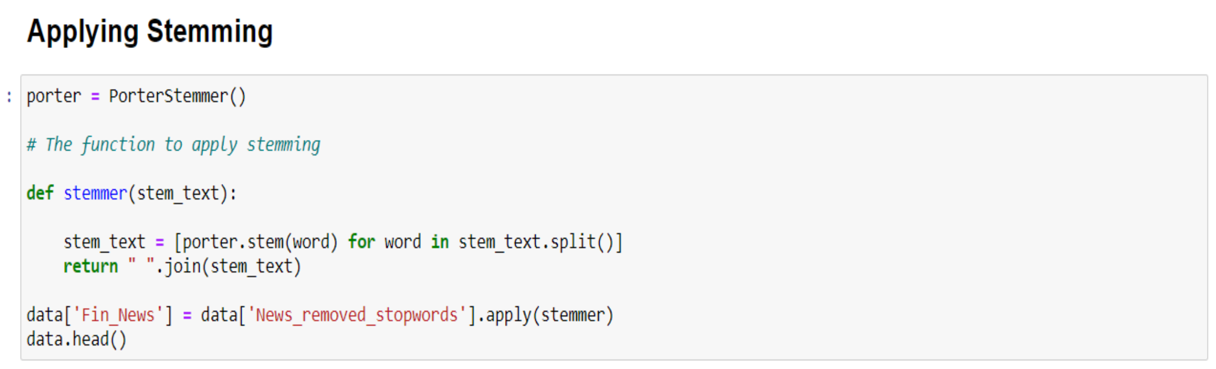
Cleaning text process

```python
# The function to clean text

def clean_text(words):

    words = re.sub("[^a-zA-Z]"," ", words)
    text = words.lower().split()
    return " ".join(text)

data['News_cleaned_data'] = data['News'].apply(clean_text)
data.head()
```

```python
data['News_cleaned_data'][0]
```

Removing stop words

```python
# Getting the stop words

stop_words = stopwords.words('english')
print(stop_words[::10])

# The function to removing stopwords

def remove_stopwords(text):

    text = [word.lower() for word in text.split() if word.lower() not in stop_words]
    return " ".join(text)

data['News_removed_stopwords'] = data['News_cleaned_data'].apply(remove_stopwords)
data.head()
```



Applying Stemming

```python
porter = PorterStemmer()

# The function to apply stemming

def stemmer(stem_text):

    stem_text = [porter.stem(word) for word in stem_text.split()]
    return " ".join(stem_text)

data['Fin_News'] = data['News_removed_stopwords'].apply(stemmer)
data.head()
```

SMOTE

The target class variable is imbalanced, The simplest way to improve imbalanced dataset is balancing them by oversampling instances of the minority class or undersampling instances of the majority class.

We will try to balancing classes by using one of the advanced techniques like the SMOTE method

(Synthetic Minority Over-sampling Technique).

SMOTE technique is one of the most commonly used oversampling methods to solve the imbalance problem. It goal is to balance class distribution by randomly increasing minority class examples by replicating them.

```
# We will use imbalanced-learn library to apply SMOTE method

smote = SMOTE()
X_train_sm,y_train_sm = smote.fit_resample(X_train,y_train)
```

```
# shape of training and testing after SMOTE

print(X_train_sm.shape)
print(y_train_sm.shape)
```

```
(16062, 272293)
(16062,)
```

Model Building

We use the following classification models:

1. Decision tree Classifier
2. Naive Bayes Classifier
3. Random Forest Classifier
4. Gradient Boosting
5. SVM (Support Vector Machine)
6. Stochastic Gradient Descent

To make the vectorizer > transformer > classifier easier to work with, we will use Pipeline class in SK-Learn.

Decision tree Classifier

```
model_dtc = Pipeline([('tfidf', TfidfTransformer()),
                      ('model',DecisionTreeClassifier()),
                     ])

model_dtc.fit(X_train_sm,y_train_sm)

ytest = np.array(y_test)
pred_0 = model_dtc.predict(X_test)

print('accuracy %s' % accuracy_score(pred_0, y_test))
print(classification_report(ytest, pred_0))
```

```
accuracy 0.9782608695652174
              precision    recall  f1-score   support

           0       0.98      0.98      0.98      2402
           1       0.98      0.98      0.98      2382

    accuracy                           0.98      4784
   macro avg       0.98      0.98      0.98      4784
weighted avg       0.98      0.98      0.98      4784
```

Naive Bayes Classifier

```
model_nbc = Pipeline([('tfidf', TfidfTransformer()),
                      ('model',MultinomialNB()),
                      ])

model_nbc.fit(X_train_sm,y_train_sm)

ytest = np.array(y_test)
pred_1 = model_nbc.predict(X_test)

print('accuracy %s' % accuracy_score(pred_1, y_test))
print(classification_report(ytest, pred_1))
```

```
accuracy 0.7508361204013378
              precision    recall  f1-score   support

           0       0.67      1.00      0.80      2402
           1       1.00      0.50      0.67      2382

    accuracy                           0.75      4784
   macro avg       0.83      0.75      0.73      4784
weighted avg       0.83      0.75      0.73      4784
```

Random Forest Classifier

```
model_rfc = Pipeline([('tfidf', TfidfTransformer()),
                      ('model',RandomForestClassifier(n_estimators=50)),
                      ])

model_rfc.fit(X_train_sm,y_train_sm)

ytest = np.array(y_test)
pred_2 = model_rfc.predict(X_test)

print('accuracy %s' % accuracy_score(pred_2, y_test))
print(classification_report(ytest, pred_2))
```

```
accuracy 0.9303929765886287
```

Accuracy of different model

We tested six different models and now we will be finding the best model with accuracy.

```
dec_acc  = accuracy_score(pred_0, y_test)
nb_acc   = accuracy_score(pred_1, y_test)
rf_acc   = accuracy_score(pred_2, y_test)
gb_acc   = accuracy_score(pred_3, y_test)
svm_acc  = accuracy_score(pred_4, y_test)
sg_acc   = accuracy_score(pred_5, y_test)
```

```
# Geeting the best accuracy from these 6 models

models = pd.DataFrame({
                      'Model': ['Decision Tree', 'Naive Bayes', 'Random Forest', 'Gradient Boosting', 'SVM', 'SGD'],
                      'Score': [dec_acc, nb_acc, rf_acc, gb_acc, svm_acc, sg_acc]})

models.sort_values(by='Score', ascending=False)
```

	Model	Score
3	Gradient Boosting	0.982651
0	Decision Tree	0.978261
4	SVM	0.977634
5	SGD	0.974080
2	Random Forest	0.930393
1	Naive Bayes	0.750836

joblib to save

```
# saving the best model based on accuracy

joblib.dump(model_gbc,'news_detection.pkl')
```

```
['news_detection.pkl']
```

CONCLUSION

- ## Key Findings and Conclusions of the Study

  With the provided dataset we were able to build a model without any hurdles. Data provided by organizations is pretty huge. In Data Science the weightage of the data plays a huge role in the building model.

  Finally we are able to gain an accuracy of 98% in this fake news detection moel

- ## Learning Outcomes of the Study in respect of Data Science

  Learning Outcomes of the Study in respect of Data Science are we got practical experience in Natural Language Processing. We were able to use new functions which helps us in simplifying our project and saves our time , and able to solve minor problems with new techniques which came in the way of modelling.