FAKE NEWS DETECTION MODEL

Submitted by:

BHAVNA PIPLANI

ACKNOWLEDGMENT

This includes mentioning of all the references, research papers, data sources, professionals and other resources that helped and guided me in completion of the project.

INTRODUCTION

A. 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.

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

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

Data Cleaning, Data Visualization using different plotting methods like distplot, countplot, Data pre-processing using NLP, Tfidf vectorizer, and Model Training using different algorithms.

B. Analytical Problem Framing

a) Mathematical/ Analytical Modelling of the Problem

Statistical modelling is the process of applying statistical analysis to a dataset. A statistical

model is a mathematical representation (or mathematical model) of observed data.

When data analysts apply various statistical models to the data they are investigating, they are able to understand and interpret the information more strategically. Rather than sifting through the raw data, this practice allows them to identify relationships between variables, make predictions about future sets of data, and visualize that data so that non-analysts and stakeholders can consume and leverage it. Most common techniques will fall into the following two groups:

Supervised learning, including regression and classification models.

Unsupervised learning, including clustering algorithms and association rules.

Some of the most common classifiers models include **LogisticRegression(), MultinomialNB(), LinearSVC(),**

DecisionTreeClassifier(), RandomForestClassifier(),KNeighborsClassifier()

b) Data Sources and their formats

Data Source is by fetching data from different e-commerce websites for further Data Cleaning, Data pre-processing and Model Training. Columns for the same to be used are news and label.

C. Explanatory Data Analysis

Importing the basic libraries to start any machine learning model.

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.simplefilter("ignore")
```

Importing the dataset from the csv file provided.

```
# importing the data from csv file
train_data = pd.read_csv("fakenews.csv",sep=',',index_col=0)
train_data
```

	id	headline	written_by	news	label
0	9653	Ethics Questions Dogged Agriculture Nominee as...	Eric Lipton and Steve Eder	WASHINGTON — In Sonny Perdue's telling, Geo...	0
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	19113	Cotton to House: 'Do Not Walk the Plank and Vo...	Pam Key	Sunday on ABC's "This Week," while discussing ...	0
3	6868	Paul LePage, Besieged Maine Governor, Sends Co...	Jess Bidgood	AUGUSTA, Me. — The beleaguered Republican g...	0
4	7596	A Digital 9/11 If Trump Wins	Finian Cunningham	Finian Cunningham has written extensively on...	1

After importing the csv file to our dataframe. We will check for no. of rows and columns.

```
train_data.shape  # checking the rows and cols count
```

```
(20800, 5)
```

The dataset has 20800 rows and 5 columns after removing unnamed 0 column. Because that same value we are getting with index also.

After checking the no. of rows and columns count, we will check the null values if any, column count, datatypes of columns.

```
1 train_data.isnull().sum() # null  values column wise counts
```

```
id              0
headline      558
written_by   1957
news           39
label           0
dtype: int64
```

The dataset is not clean. There are few missing or null values in the dataset.

Now, we will check the information about the dataframe which shows null values in few columns.

```
1 # checking the information about the not-null, datatypes, rows and cols
2 train_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 20800 entries, 0 to 20799
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   id          20800 non-null  int64 
 1   headline    20242 non-null  object
 2   written_by  18843 non-null  object
 3   news        20761 non-null  object
 4   label       20800 non-null  int64 
dtypes: int64(2), object(3)
memory usage: 975.0+ KB
```

Now, it can be observed that it has Nan values. For that, further data cleaning will be required. We

dont need columns headline and written_by to detect the news if it is fake or not. we only need to preprocess news and predict the label. So we will drop these two columns and will drop rows having null values in news columns

```
1 train_data=train_data.drop(columns=['headline', 'written_by'])
2 train_data
```

	id	news	label
0	9653	WASHINGTON — In Sonny Perdue's telling, Geo...	0
1	10041	HOUSTON — Venezuela had a plan. It was a ta...	0
2	19113	Sunday on ABC's "This Week," while discussing ...	0
3	6868	AUGUSTA, Me. — The beleaguered Republican g...	0
4	7596	Finian Cunningham has written extensively on...	1
...	...	...	...
20795	5671	No, you'll be a dog licking of the vomit of yo...	1
20796	14831	By Rixon Stewart on November 5, 2016 Rixon Ste...	1
20797	18142	posted by Eddie You know the Dakota Access Pip...	1
20798	12139	It's officially summer, and the Society Boutiq...	0
20799	15660	Emory University in Atlanta, Georgia, has anno...	0

There are 39 rows where news has null values for that we can't detect if it is fake or not. So, we will remove the specific rows.

```
# dropping rows having null values
train_data=train_data.dropna()
train_data
```

	id	news	label
0	9653	WASHINGTON — In Sonny Perdue's telling, Geo...	0
1	10041	HOUSTON — Venezuela had a plan. It was a ta...	0
2	19113	Sunday on ABC's "This Week," while discussing ...	0
3	6868	AUGUSTA, Me. — The beleaguered Republican g...	0
4	7596	Finian Cunningham has written extensively on...	1
...	...	...	...
20795	5671	No, you'll be a dog licking of the vomit of yo...	1
20796	14831	By Rixon Stewart on November 5, 2016 Rixon Ste...	1
20797	18142	posted by Eddie You know the Dakota Access Pip...	1
20798	12139	It's officially summer, and the Society Boutiq...	0
20799	15660	Emory University in Atlanta, Georgia, has anno...	0

20761 rows × 3 columns

After removing null values,we will check for the data if it is balanced or not.

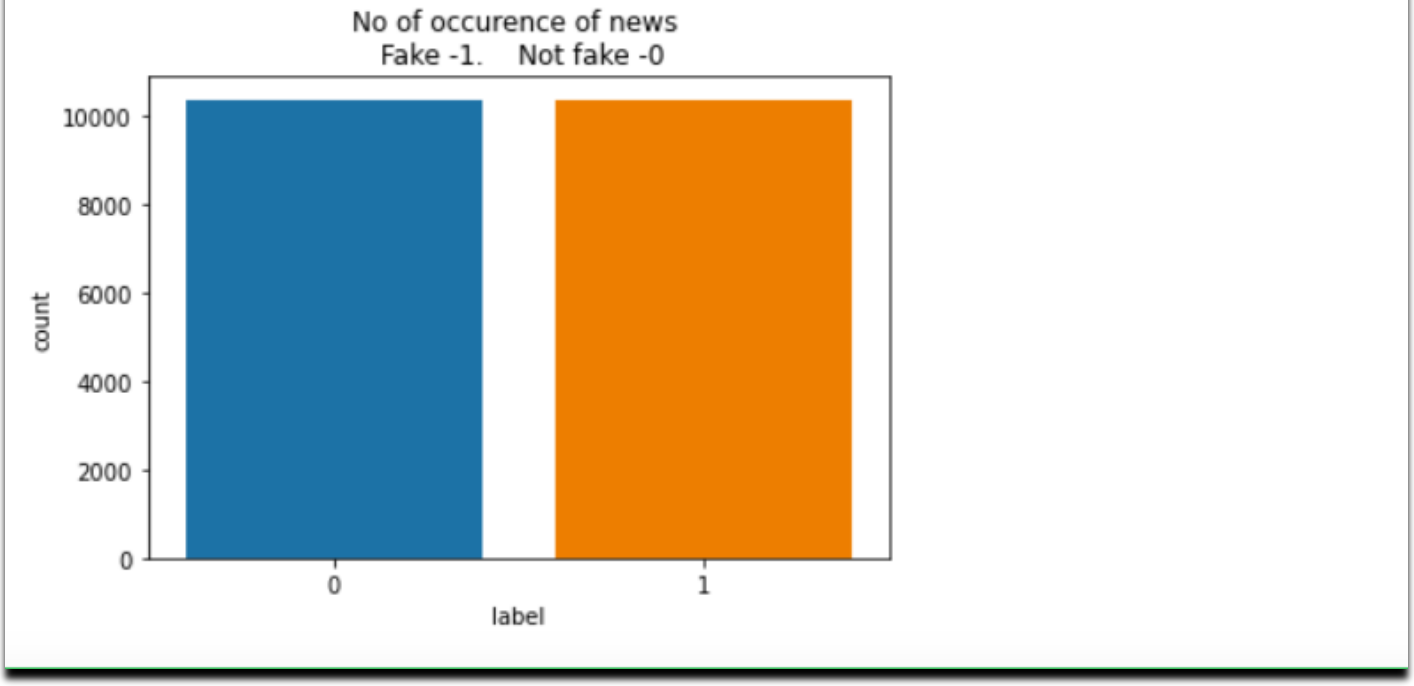
```
train_data.label.value_counts()
```

```
0    10387
1    10374
Name: label, dtype: int64
```

Data is quite balanced . Rows count for all the label are almost same. So , no data balancing technique is required.

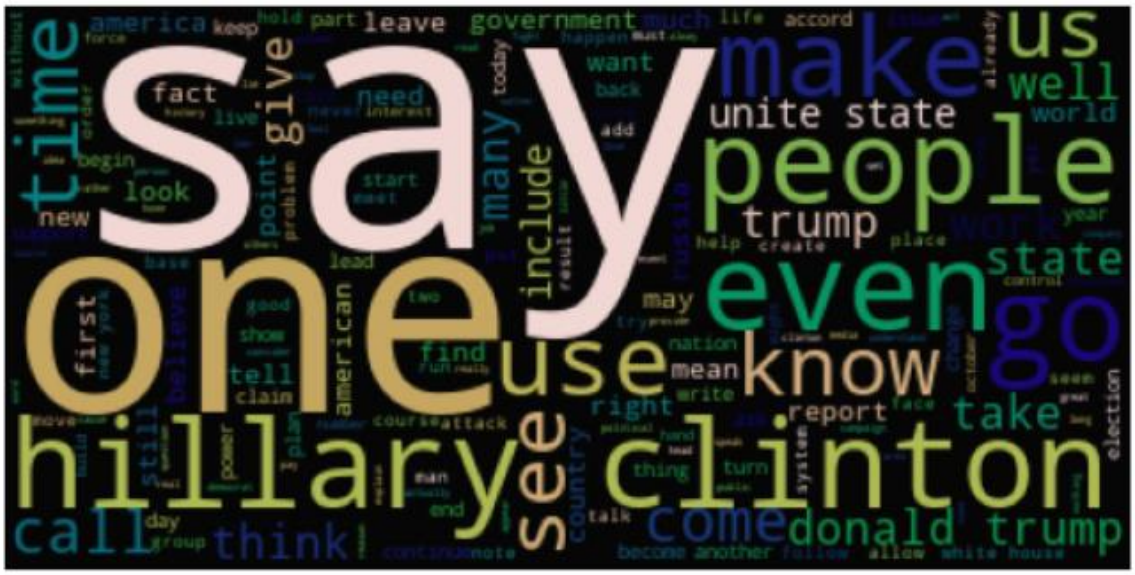
D. Data Visualization

```
plt.figure(figsize=(6,4))
sns.countplot(train_data['label'])
plt.title("No of occurence of news \n Fake -1.    Not fake -0")
plt.show()
```



Above countplot shows the number of occurrence of news in both values i.e for 0 and 1

```
1 wordcloud(train_data,"label")
```



Above wordcloud displays few words that are used in news feature.

E. Pre-Processing Pipeline

The Natural Language ToolKit is one of the best-known and most-used NLP libraries in the Python ecosystem, useful for all sorts of tasks from tokenization, to stemming, to part of speech tagging, and beyond.

Tokenization is a step which splits longer strings of text into smaller pieces, or tokens. Larger

chunks of text can be tokenized into sentences, sentences can be tokenized into words, etc. Further processing is generally performed after a piece of text has been appropriately tokenized.

For our task, we will tokenize our sample text into a list of words. This is done using NTLK's word_tokenize() function.

```
import re
import nltk
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
```

```
stop_words=set(stopwords.words('english'))
lemma=WordNetLemmatizer()
```

Using this below function we will clean the data like removing everything from the text other than alphabets , tokenization , lemmatization .

```
def clean_news(news_text):
    news_text=re.sub(r'http\$+','',news_text) # removing the url
    news_text=re.sub('[^a-zA-Z]',' ',news_text) #removing Numbers and punctuation
    news_text=str(news_text).lower().replace('\\','').replace('_',' ') #converting all to l
    news_text=word_tokenize(news_text) #tokenization
    news_text=[item for item in news_text if item not in stop_words] # removing stop words
    news_text=[lemma.lemmatize(word=w,pos='v') for w in news_text] #lemmatization
    news_text=[i for i in news_text if len(i)>=2] # removing the words having length <2
    return news_text
```

Below is the representation of cleaned dataset for model training.

```
train_data['news']=[" ".join(news_text) for news_text in train_data['news'].values] # conve
train_data
```

	id	news	label
0	9653	washington sonny perdue tell georgians grow we...	0
1	10041	houston venezuela plan tactical approach desig...	0
2	19113	sunday abc week discuss republican plan repeal...	0
3	6868	augusta beleaguer republican governor maine se...	0
4	7596	finian cunningham write extensively internatio...	1
...	...	...	...
20795	5671	dog lick vomit chinese overlords	1
20796	14831	rixon stewart november rixon stewart nov migra...	1
20797	18142	post eddie know dakota access pipeline protest...	1
20798	12139	officially summer society boutique society mem...	0
20799	15660	emory university atlanta georgia announce fund...	0

20761 rows × 3 columns

Encoding text into vectors for further model training

```
from sklearn.feature_extraction.text import TfidfVectorizer
tfid=TfidfVectorizer(smooth_idf=False,max_features=20000,ngram_range=(1,3),analyzer='word')
X=tfid.fit_transform(train_data["news"])
y=train_data["label"]
```

After all the data cleaning and data pre processing, X and y variables are processed for training the model.

```
1 X.shape
```

```
(20761, 20000)
```

```
1 y.shape
```

```
(20761,)
```

Libraries and packages used for model training are listed below

```
1 #importing the model training libraries
2 from sklearn.model_selection import train_test_split
3 from sklearn.naive_bayes import MultinomialNB
4 from sklearn.svm import LinearSVC
5 from sklearn.tree import DecisionTreeClassifier
6 from sklearn.ensemble import RandomForestClassifier
7 from sklearn.neighbors import KNeighborsClassifier
```

```
1 from sklearn.metrics import accuracy_score,confusion_matrix,classification_report
2 import warnings
3 warnings.filterwarnings('ignore')
```

F. Model Development and Evaluation

Identification of possible problem-solving approaches (methods). Most common techniques will fall into the following two groups:

Supervised learning, including regression and classification models.

Unsupervised learning, including clustering algorithms and association rules.Testing of Identified Approaches (Algorithms)

Here, In this project I will be using **LogisticRegression(),MultinomialNB(),LinearSVC(),DecisionTreeClassifier(),RandomForestClassifier(),KNeighborsClassifier()** algorithms

First of all, Lets train the model. Here,I am using test_size=.22 that means 22% of data will go for testing purpose.

```
1 x_train,x_test,y_train,y_test=train_test_split(X,y,test_size=.22,random_state=85)
```

Using different algorithms, we will try to find the best model.

```
#using algorithms in for loops
model=[LogisticRegression(),MultinomialNB(),LinearSVC(),DecisionTreeClassifier(),RandomFore
for m in model:
    m.fit(x_train,y_train)
    y_pred=m.predict(x_test)
    print("Accuracy score of " , m , "is " , accuracy_score(y_test,y_pred))
    print("confusion matrix of " , m , "is \n",confusion_matrix(y_test,y_pred))
    print("classification report of " , m, "is \n",classification_report(y_test,y_pred))
    print("**********************************************************************\n")
```

```
Accuracy score of  LogisticRegression() is  0.9516199649737302
confusion matrix of  LogisticRegression() is
 [[2165  126]
 [  95 2182]]
classification report of  LogisticRegression() is
              precision    recall  f1-score   support

           0       0.96      0.95      0.95      2291
           1       0.95      0.96      0.95      2277

    accuracy                           0.95      4568
   macro avg       0.95      0.95      0.95      4568
weighted avg       0.95      0.95      0.95      4568

**********************************************************************
```

The best performing model among all being tested was LinearSVC with more than 96% of accuracy. Also, precision and F1 score and recall for all the label was pretty good.

Doing a **GridSearchCV** is a great way to do **hyperparameters** tuning.

Hyperparameter tuning using GridSearchCv

```
# LinearSVC is best performing model so finding its best parameter
from sklearn.model_selection import GridSearchCV
```

```
params = {
            'C': [0.1, 0.5, 1.0, 10.0],
         }

linear_svc_grid = GridSearchCV(LinearSVC(random_state=1, max_iter=1000000), param_grid=para
linear_svc_grid.fit(x_train,y_train)

print('Train Accuracy : %.3f'%linear_svc_grid.best_estimator_.score(x_train, y_train))
print('Test Accuracy : %.3f'%linear_svc_grid.best_estimator_.score(x_test, y_test))
print('Best Accuracy Through Grid Search : %.3f'%linear_svc_grid.best_score_)
print('Best Parameters : ',linear_svc_grid.best_params_)
```

Now, Let's check cross validate the LinearSVC model.

```
# cross validating LinearSVC
from sklearn.model_selection import cross_val_score

score=cross_val_score(linear_svc_grid,X,y,cv=5,scoring='accuracy')
print("Cross Validation Score : ", score,"\n")
print("Mean" , score.mean())
print("Standard Deviation" , score.std())
```

```
Cross Validation Score :  [0.96580785 0.96387283 0.96989403 0.96531792 0.
96579961]

Mean 0.9661384485621509
Standard Deviation 0.002006726617404872
```

Accuracy for validation was also proved good. Mean was around .966. And Standard Deviation was very less.

Now, as the model is performing good with the score of 96% , we will save the predicted_model .

Saving the model- Serialization

```
1 print(pred,'\t',y_pred)
```

```
[1 1 0 ... 0 1 0]        [1 1 1 ... 1 1 1]
```

```
1 # saving the prediction model
2
3 import pickle
4 filename="fakenews.pkl"
5 pickle.dump(linear_svc_grid,open(filename,'wb'))
```

```
1 ds_pred=pd.DataFrame(data=pred,columns=['label'])
2 ds_pred
```

	label
0	1
1	1
2	0
3	1
4	0
...	...
4563	1
4564	0
4565	0
4566	1
4567	0

4568 rows × 1 columns

CONCLUSION

Key Findings and Conclusions of the Study:

Data received from csv file was quite clean other than few null values were seen.

Data Preprocessing using NLP was done including tokenization, lemmatization, data cleaning by removing every symbol, special characters etc other than alphabets.

Data was balanced as the fake news and not fake news count was almost equal.

LinearSVC was the best performing and fit model.