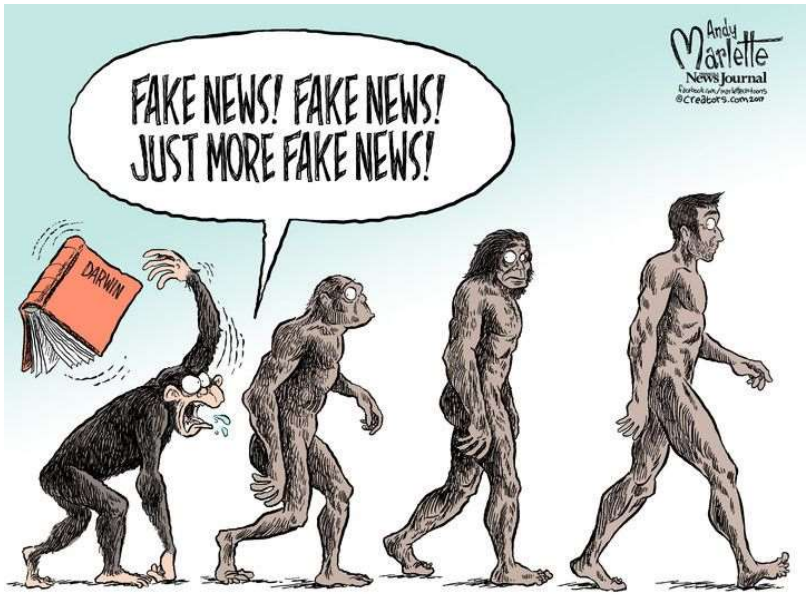
FLIP ROBO

Fake News Detection Project

(Using ML (NLP))



Submitted by:

ARUN KUMAR

ACKNOWLEDGMENT

This project includes the professional reference of much external research analysis done by various organisations and individuals. Such references are mentioned below:

1. Fake news classifier on US Election News | https://www.analyticsvidhya.com/blog/2020/12/fake-news-classifier-on-us-election-news%F0%9F%93%B0-lstm-%F0%9F%88%9A/

2. Detecting opinion spams and fake news using text classification | https://onlinelibrary.wiley.com/doi/epdf/10.1002/spy2.9

3. A Literature Review of NLP Approaches to Fake News Detection and TheirApplicability to Romanian-Language News Analysis | https://revistatransilvania.ro/wp-content/uploads/2020/11/2020.10.7-Busioc-et-al.pdf

4. Proceedings of the Fourteenth International AAAI Conference on Web and Social Media (ICWSM 2020) | Toward a Better Performance Evaluation Framework for Fake News ClassificationLia Bozarth, Ceren BudakUniversity of Michigan, School of Information105 S State StAnn Arbor, MI 48109{lbozarth, cbudak}@umich.edu | https://ojs.aaai.org/index.php/ICWSM/article/view/7279/7133

5. Improvement of Misleading and Fake News Classification for Flective Languages by Morphological Group Analysis by Jozef Kapusta [OrcID] and Juraj Obonya [OrcID]|Department of Informatics, Constantine the Philosopher University in Nitra, Nitra SK-94974, Slovakia|Author to whom correspondence should be addressed. Informatics 2020, 7(1), 4; https://doi.org/10.3390/informatics7010004
Received: 24 December 2019 / Revised: 24 January 2020 / Accepted: 3 February 2020 / Published: 5 February 2020| https://www.mdpi.com/2227-9709/7/1/4/htm

6. Trust or Suspect? An Empirical Ensemble Framework for Fake News Classification Travel's Practice at WSDM Cup 2019 Fake News Classification Challenge
Shuaipeng Liu, Shuo Liu, Lei Ren
Meituan-Dianping Group {liushuaipeng, liushuo19, renlei04}@meituan.com | https://shuaipengliu.com/resource/wsdm2019.pdf

7. Detecting Fake News for Reducing Misinformation Risks Using Analytics Approaches | https://www.researchgate.net/publication/333776916_Detecting_Fake_News_for_Reducing_Misinformation_Risks_Using_Analytics_Approaches

8. INTERNATIONAL CONFERENCE ON RECENT TRENDS IN ADVANCED COMPUTING 2019, ICRTAC 2019 | Analysis of Classifiers for Fake News Detection

TABLE OF CONTENTS

INTRODUCTION

1. What is Fake News Detection?

Against the changes the world is currently facing, an increasing number of daily tasks are **moving online**, including reading the news and being informed about relevant topics. The increase in the volume of information made available led, in turn, to **fake news dissemination** becoming a trending topic on the Internet. Recent events such as the pandemic have shown that "***Fake news exerts a significant negative influence on societies by highlighting stories that do not necessarily report on facts"***. Research communities expressed concern about this flood of misinformation and introduced automated fake news identification solutions based on Natural Language Processing (NLP) techniques. In this study, we reference existing datasets and related work in the analysis of English fake news, discuss potential detection techniques for Fake news, as well as establish future work plans for this research initiative.

2. Importance of Fake News Detection.

"***Fake news serves to misinform people and manipulate their opinions for several reasons"*.** News media has become a channel to pass on the information of what's happening in the world to the people living. Often people perceive whatever conveyed in the news to be true. There were circumstances where even the news channels acknowledged that their news is not true as they wrote. But some news has a significant impact not only on the people or government but also on the economy. One news can shift the curves up and down depending on the emotions of people and political situation.

It is ***important to identify the fake news from the real true news***. The problem has been taken over and resolved with the help of Natural Language Processing tools which help us identify fake or true news based on historical data.

3. **Business Problem – Fake News influencing social and political environments.**

   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 sensationalism of not-so-accurate eye-catching and intriguing headlines aimed at retaining the attention of audiences to sell information has persisted all throughout the history of all kinds of information broadcast. On social networking websites, the reach and effects of information spread are however significantly amplified and occur at such a fast pace, that distorted, inaccurate, or false information acquires a tremendous potential to cause real impacts, within minutes, for millions of users.

4. **Review of Literature.**

   In this section, we discuss frequently employed methods for compiling fake news corpora, alongside the machine learning solutions developed to analyze data and build classifiers capable of distinguishing between fake and true news.

5. **Future of Fake News Detection.**

   Fighting fake news is a difficult and challenging task. With an increasing impact on the social and political environment, fake news exerts an unprecedently dramatic influence on people's lives. In response to this phenomenon, initiatives addressing automated fake news detection have gained popularity, generating widespread research interest. However, most approaches targeting English and low-resource languages

experience problems when devising such solutions. This study focuses on the progress of such investigations, while highlighting existing solutions, challenges, and observations shared by various research groups. In addition, given the limited amount of automated analyses performed on fake news, we inspect the applicability of the available approaches, while identifying future research paths.

We assessed the available solutions, potential obstacles, the results obtained by using various NLP techniques, and the data recent experiments relied on. Drawing on the existing international literature, and in places on our own related research findings, we also outlined potential approaches to the automated detection of fake news.

ANALYTICS OF THE BUSINESS PROBLEM

1. What is Analytical problem framing?

Analytic problem framing involves translating the business problem into terms that can be addressed analytically via data and modelling. It's at this stage that you work backwards from the results / outputs you want to the data/inputs you're going to need, where you identify potential drivers and hypotheses to test, and where you nail down your assumptions. Analytic problem framing is the antithesis of merely working with the ready-to-hand data and seeing what comes of it, hoping for something insightful. Typically, the process moves on from here to data collection, cleansing and transformation, Methodology selection and model building, never to return. But if you're willing to borrow and use a concept from complex adaptive systems – maps and models – you can make repeat use of this stage to improve your overall outcome.

2. **Analytics of the business problem**

   As discussed above we need to build a ML Model to predict the news from the dataset as fake or true news. For that we need some analytical data's. So we have to find the previous works done by others in this same topic to eliminate the repetition of work. Also we need to collect data with proper amalgamation of true and fake news. The attributes should contain the news title, actual news, written by, date of the news, published by,etc.,

3. **Hardware Requirements**

   A mid level computer that runs on Intel i3/i5/i7 or A10/A11/M1 or ryzen 3/5 or any other equivalent chipset and a suitable processor.

4. **Software Requirements**

   Windows / Linux /Mac OS

5. **Tools, Libraries and Packages used**

   Tool: 1.Anaconda Navigator
   2. Jupyter Notebook

   Libraries and Packages:

   1. Numpy
   2. Pandas
   3. Matplotlib
   4. Seaborn
   5. NLTK and its derivatives
   6. Regex expressions
   7. String
   8. Classification ML models.

6. Data sources

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).

The shape of the data set is **20800 x 6**, that is data set consist of **20800 rows** and **6 columns.**

7. Pre-Assumptions

As we look into the data –set, primarily we could infer that the data-set is Pretty much well informed.

1. It has all the required data's such

#	Column	Non-Null Count	Dtype
0	Unnamed: 0	20800 non-null	int64
1	id	20800 non-null	int64
2	headline	20242 non-null	object
3	written by	18843 non-null	object
4	news	20761 non-null	object
5	label	20800 non-null	int64

Dtype: int64 (3), object (3)

2. The numbers of entries in the Label column is divided into 2, fake (1) and Not Fake (0). It is distributed as represented as below:
#values and count of the target variable,
1 – 10413 (Fake News)
0 – 10387 (Not a Fake News)

8. Data Pre- Processing

Here we will try to make the given data-set efficient for the ML model. For that we will carry out a series of process which will make the data-set perfect for building a ML model. This series of process includes,

1. **Collection of basic statistical data.**
2. **Exploratory data analysis.**
3. **Text Data Pre-Processing.**
4. **Text Data Vectorization.**

By following these processes we can achieve a more efficient data-set. We will use **Python** through **Jupyter notebook** for data processing. Also we will use Libraries such as **Pandas, Numpy for Analysis** and **Matplotlib, seaborn for visualization**.

Collection of basic statistical data

Now we will import the required libraries in to the Jupyter Notebook using Python codes.

Importing Libraries

```
##importing libraries

#data manipulation
import pandas as pd
import numpy as np
import re
import string

##Machine learning and text processing libraries
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer
from nltk.stem import WordNetLemmatizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
from sklearn.naive_bayes import MultinomialNB
from sklearn.linear_model import LogisticRegression
from sklearn.svm import LinearSVC
from wordcloud import WordCloud


#libraries used for visualizations
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

We now have to load the data set into the Notebook.

Importing the CSV file

```
#importing the csv file
pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)
dftrain=pd.read_csv(r"train_news")
dftrain.head(10)
```

	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
5	5	3196	Whatever the Outcome on November 8th the US Wi...	NaN	Taming the corporate media beast Whatever the ...	1
6	6	5134	Rapid Evolution Saved This Fish From Pollution...	JoAnna Klein	The State of New Jersey says you can't eat the...	0
7	7	1504	Alabama Prison Officials Retaliate Against Pri...	Brian Sonenstein	Advocates say prison officials at the Kilby Co...	1
8	8	13559	NaN	steventexas	People have made up their minds on president.\...	1
9	9	4203	Can We Live in a Constant State of Love?	Gillian	Leave a reply \nToni Emerson – When we fall in...	1

```
#basic info
dftrain.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20800 entries, 0 to 20799
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   Unnamed: 0  20800 non-null  int64 
 1   id          20800 non-null  int64 
 2   headline    20242 non-null  object
 3   written_by  18843 non-null  object
 4   news        20761 non-null  object
 5   label       20800 non-null  int64 
dtypes: int64(3), object(3)
memory usage: 975.1+ KB
```

```
#shape of the data-set
dftrain.shape
```

```
(20800, 6)
```

```
#Basic statistical Data
dftrain.describe().T
```

	count	mean	std	min	25%	50%	75%	max
Unnamed: 0	20800.0	10399.500000	6004.587135	0.0	5199.75	10399.5	15599.25	20799.0
id	20800.0	10399.500000	6004.587135	0.0	5199.75	10399.5	15599.25	20799.0
label	20800.0	0.500625	0.500012	0.0	0.00	1.0	1.00	1.0

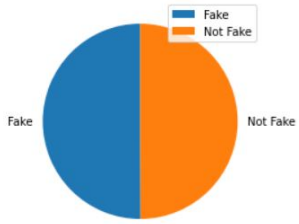
```
dftrain.describe(include='object').T
```

	count	unique	top	freq
headline	20242	19803	The Dark Agenda Behind Globalism And Open Borders	5
written_by	18843	4201	Pam Key	243
news	20761	20386		75

```
#values and count of the target variable
dftrain.label.value_counts()
```

```
1    10413
0    10387
Name: label, dtype: int64
```

```
# checking the data distribution among the data-set
y=dftrain.label.value_counts()
mylabels = ["Fake", "Not Fake"]
plt.pie(y, labels = mylabels, startangle = 90)
plt.legend()
plt.show()
```



Observations:

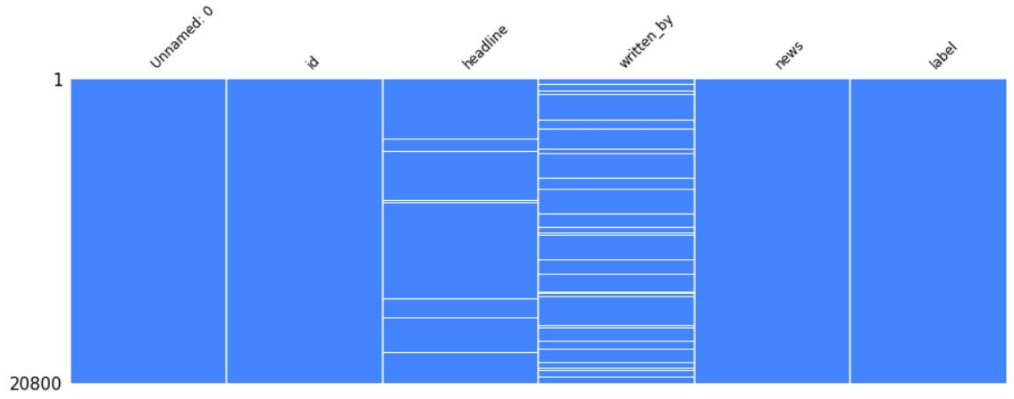
- The Dataset contains 6 columns, namely "Unnamed: 0", "id", "headline", "written by", "news", "label". The shape of the data set is 20800 x 6.
- It appears that the "Label" column is our Target Variable. Whereas "news" is our main text resource. We will use this to build a ML model.
- The target value seems to be equally distributed.

Handling Null Values:

Now we will check the null values in our data set.

```
# Program to visualize missing values in dataset
# Importing the libraries
import missingno as msno
# Visualize missing values as a matrix
msno.matrix(dftrain,labels=True, sparkline=False, figsize=(15,5), fontsize=12, color=(0.27, 0.52, 1.0))
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x225ac103d30>
```

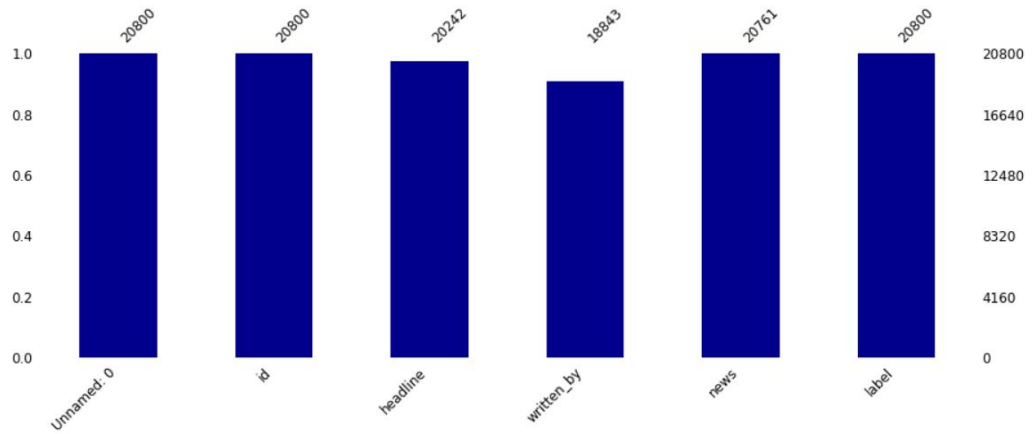


Observations:

1. Title: Unnamed: 0
   NaN val: 0
2. Title: id
   NaN val: 0
3. Title: headline
   NaN val: 558
4. Title: written_by
   NaN val: 1957
5. Title: news
   NaN val: 39
6. Title: label
   NaN val: 0

```
# Visualize missing values as a matrix
msno.bar(df = dftrain, figsize=(15, 5), fontsize=12, color="darkblue")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x225ac17e940>
```



Observation:

1. The above plot and data shows that, there is some data missing from the news column in the given data-set.
2. we will now drop the Null values in our data-set.

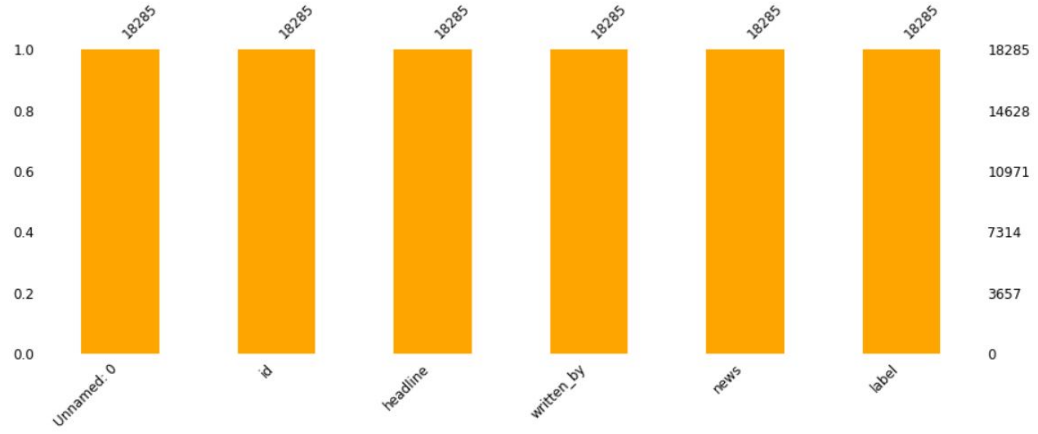
Dropping the Null-Values.

```
# dropping the null-values
dftrain.dropna(inplace= True)
print(dftrain.shape)
print(dftrain.isnull().sum())
```

```
(18285, 6)
Unnamed: 0    0
id            0
headline      0
written_by    0
news          0
label         0
dtype: int64
```

```
#visualizing the change
msno.bar(df = dftrain, figsize=(15,5), fontsize=12, color="orange")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x225ac2c5fa0>
```



Observations:

- We can see that the present data-set contains no null values, and it is equally distributed.

Dropping the less important columns:

1. We can see that the columns

- "headline",
- "written_by",
- "id",
- "Unnamed: 0"

are not providing any important insights as of now. So dropping them would be a wise choice for now.

```
dftrain = dftrain.drop(["headline", "written_by", "id","Unnamed: 0"], axis=1)
dftrain.head(5)
```

	news	label
0	WASHINGTON — In Sonny Perdue's telling, Geo...	0
1	HOUSTON — Venezuela had a plan. It was a ta...	0
2	Sunday on ABC's "This Week," while discussing ...	0
3	AUGUSTA, Me. — The beleaguered Republican g...	0
4	Finian Cunningham has written extensively on...	1

Further, we can move on to the text preprocessing.

Text Data Pre-Processing:

Here, we will

- Transform the text into lower case.
- Replace the email addresses.
- Replace the URLs.
- Remove the HTML tags.
- Remove the numbers.
- Remove extra newlines.
- Remove the punctuations.
- Remove the unwanted white spaces.
- Remove the stop words.

Defining the function

```
# cleaning the text data for vectorization
# defining the function
def clean_txt(text):
    text = text.lower() #Converting the text to lower case
    text = re.sub('\[.*?\]','',text) #Replacing email addresses
    text = re.sub('\\W', ' ', text) #Removing Punctuations
    text = re.sub('https?://\S+|www\.S+', '', text)  #Replace URLs with 'webaddress'
    text = re.sub('<.*?>+', '', text) #Removing the HTML tags
    text = re.sub('[%s]' % re.escape(string.punctuation), '', text)#Removing Punctuations
    text = re.sub('\n', '', text) #Removing new lines
    text = re.sub('\w*\d\w*', '', text)
    tokenized_text = word_tokenize(text) #word_tokenization
    stop_words = set(stopwords.words('english') + ['u','ur','im','doin', 'ü', 'â', 'e', 'ur', 'doin', 'ure','READ MORE']) #decla
ring stop Stop_Words
    WL = WordNetLemmatizer() #declaring lemmatizer
    text = [WL.lemmatize(word) for word in tokenized_text if word not in stop_words if word.isalpha()] # lemmatization and remov
al of stop_words
    return " " .join(text)
```

Applying the defined function to the News column.

```
# applying the clean_txt function to the "news" column
dftrain['news'] = dftrain['news'].apply(clean_txt)
dftrain.head(5)
```

	news	label
0	washington sonny perdue telling georgian growi...	0
1	houston venezuela plan tactical approach desig...	0
2	sunday abc week discussing republican plan rep...	0
3	augusta beleaguered republican governor maine ...	0
4	finian cunningham written extensively internat...	1

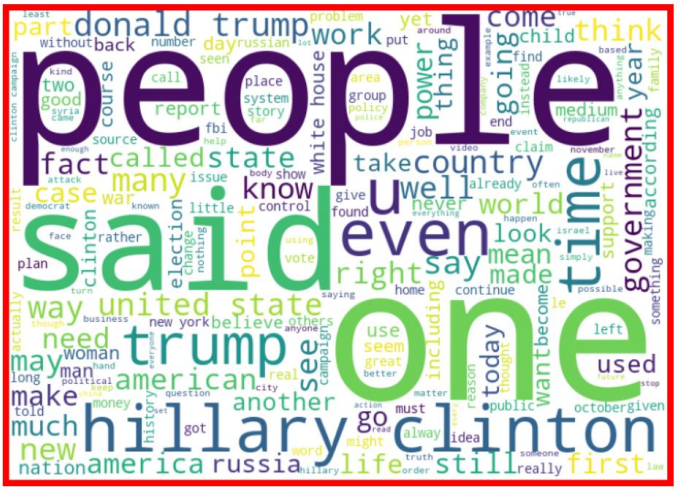
Observations:

1. All the functions have been applied and we got a cleaned text output. This text is further vectorized to build a Machine Learning model.

Word Cloud:

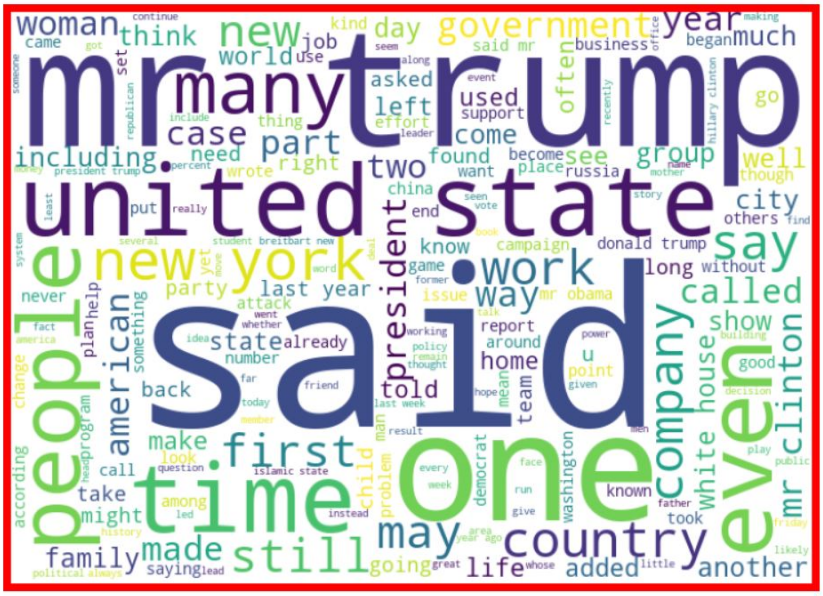
Getting sense of words in the Fake News.

```
#Getting sense of words Fake news
plot = dftrain['news'][dftrain['label']==1]
plot_cloud = WordCloud(width=700,height=500,background_color='white',max_words=200).generate(' '.join(plot))
plt.figure(figsize=(10,8),facecolor='r')
plt.imshow(plot_cloud)
plt.axis('off')
plt.tight_layout(pad=0)
plt.show()
```



Getting sense of words in the Not a Fakes News attribute.

```
#Getting sense of words Not_Fake news
plot = dftrain['news'][dftrain['label']==0]
plot_cloud = WordCloud(width=700,height=500,background_color='white',max_words=200).generate(' '.join(plot))
plt.figure(figsize=(10,8),facecolor='r')
plt.imshow(plot_cloud)
plt.axis('off')
plt.tight_layout(pad=0)
plt.show()
```



Text Data Vectorization:

```
# vectorizing the text data using TfidfVectorizer
tfidf = TfidfVectorizer(max_features = 15000)
x = tfidf.fit_transform(dftrain['news'])
y = dftrain['label']
```

ML Model Development and Evaluation

1. Problem Identification

The given objective for our project is to predict the instance that whether the News is Fake or not. That instance is labeled in the Target variable as 1's and 0's where 1's are Fake News and 0's are Not a Fake News. This is a classic case of "**Classification**". So we will use different type of classifying machine learning algorithms to build a prediction model.

2. Listing of ML Models

For this given Objective we will use the following Machine Learning Models.

1. Logistic Regression.
2. RandomForestClassifier.
3. MultinomialDB.
4. Gradient Boosting Classifier.
5. DecisionTreeClassifier.

1. Logistic Regression.

   Logistic Regression is a supervised learning algorithm that is used when the target variable is categorical. Hypothetical function h(x) of linear regression predicts unbounded values. But in the case of Logistic Regression, where the target variable is categorical we have to strict the range of predicted values. Consider a classification problem, where we need to classify whether an email is a spam or not. So, the hypothetical function of linear regression could not be used here to predict as it predicts unbound values, but we have to predict either 0 or 1 in the process.

2. Random Forest Classifier.

   The Random forest classifier creates a set of decision trees from a randomly selected subset of the training set. It is basically a set of decision trees (DT) from a randomly selected subset of the training set and then It collects the votes from different decision trees to decide the final prediction.

3. MultinomialDB.

   MultinomialDB is among one of the very simple and powerful algorithms for classification based on Bayes Theorem with an assumption of independence among the predictors. The Naive Bayes classifier assumes that the presence of a feature in a class is not related to any other feature. Naive Bayes is a classification algorithm for binary and multi-class classification problems.

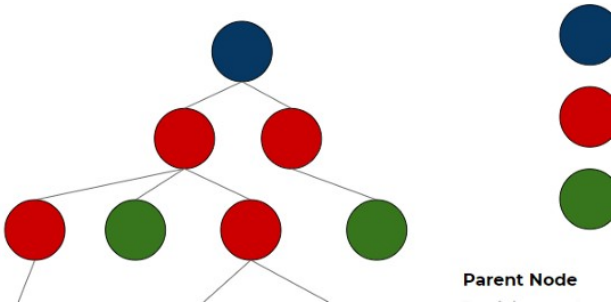
4. Gradient Boosting Classifier.

   Gradient Boosting is a popular boosting algorithm. In gradient boosting, each predictor corrects its predecessor's error. In contrast to Adaboost, the weights of the training instances are not tweaked,

instead, each predictor is trained using the residual errors of predecessor as labels.
There is a technique called the Gradient Boosted Trees whose base learner is CART (Classification and Regression Trees).

5. DecisionTreeClassifier.

   A classification tree is used when the dependent variable is categorical. The value obtained by leaf nodes in the training data is the mode response of observation falling in that region it follows a top-down greedy approach.
   The general idea behind the Decision Tree is to find the splits that can separate the data into targeted groups.
   Terminologies associated with decision tree



Processing the Data-set for Training and Testing.

Now we will use the train_test _split to split and train the data-set for testing the data-set. This is a crucial phase of building the ML model.

1. Training set is a subset of the dataset used to build predictive models.

2. Test set, or unseen data, is a subset of the dataset used to assess the likely future performance of a model. If a model fits to the training set much better than it fits the test set, over fitting is probably the cause.

Creating train_test_split

```
#Creating train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,random_state=42,test_size=.25,stratify=y)
```

```
# find the shape of x and y
x.shape, y.shape
```

```
((18285, 15000), (18285,))
```

```
# find the shape of x_train and y_train
x_train.shape, y_train.shape
```

```
((13713, 15000), (13713,))
```

Evaluation of ML Models.

Here will now build the ML models with the help of the given Data-set. We will also use accuracy score, confusion matrix, F1 score, and Roc_Auc score for finding the efficiency of the built Machine Learning Model. Model evaluation aims to estimate the generalization accuracy of a model on future (unseen/out-of-sample) data.

Classification Accuracy

Accuracy is a common evaluation metric for classification problems. It's the number of correct predictions made as a ratio of all predictions made. We use sklearn module to compute the accuracy of a classification task.

Confusion Matrix

A confusion matrix provides a more detailed breakdown of correct and incorrect classifications for each class.

Area under Curve (AUC)

Area under ROC Curve is a performance metric for measuring the ability of a binary classifier to discriminate between positive and negative classes.

F-Measure

F-measure (also F-score) is a measure of a test's accuracy that considers both the precision and the recall of the test to compute the score. Precision is the number of correct positive results divided by the total predicted positive observations. Recall, on the other hand, is the number of correct

positive results divided by the number of all relevant samples (total actual positives).

Importing machine learning models

```
# importing machine learning models
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.naive_bayes import MultinomialNB

from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import GradientBoostingClassifier

#Importing error metrics
from sklearn.metrics import classification_report,confusion_matrix,accuracy_score,roc_curve,auc
from sklearn.model_selection import GridSearchCV,cross_val_score
```

Initializing the model:

```
# Intializing the model
RF = RandomForestClassifier()
LR = LogisticRegression()
DT = DecisionTreeClassifier()
GBC = GradientBoostingClassifier()
MNB = MultinomialNB()

# appending models with their respective declarations
models= []
models.append(('RandomForestClassifier', RF))
models.append(('LogisticRegression',LR))
models.append(('MultinomialNB()',MNB))
models.append(('DecisionTreeClassifier',DT))
models.append(('MultinomialNB', MNB))

# Creating empty list
Model=[]
score=[]
cvs=[]
roc_auc_score=[]
Precision=[]
```

Creating a loop to run the data through the models

```
# creating a loop to run the data through the models
for name,model in models:

    # model fitting
    Model.append(name)
    model.fit(x_train,y_train)
    print(model)
    pre=model.predict(x_test)
    print('\n')
    # accuracy score
    aucc_score=accuracy_score(y_test,pre)
    print('accuracy_score: ',aucc_score)
    score.append(aucc_score*100)
    print('\n')
    # cross-validation score
    cv_score=cross_val_score(model,x,y,cv=10,scoring='roc_auc').mean()
    print('Cross Val Score : ', cv_score)
    cvs.append(cv_score*100)
    print('\n')
    # classification report
    print('classification_report\n',classification_report(y_test,pre))
    # roc_auc
    false_positive_rate,true_positive_rate, thresholds=roc_curve(y_test,pre)
    roc_auc=auc(false_positive_rate, true_positive_rate)
    print('roc auc score : ', roc_auc)
    roc_auc_score.append(roc_auc*100)
    print('\n')
    # confusion matrix
    print('Confusion Matrix:\n',confusion_matrix(y_test,pre))
    print('\n')
    print("..........................................................")
    print('\n')
```

1. Random Forest Classifier.

```
RandomForestClassifier()


accuracy_score:  0.9311023622047244


Cross Val Score :  0.9879663059644376


classification_report
              precision    recall  f1-score   support

           0       0.91      0.97      0.94      2591
           1       0.96      0.88      0.92      1981

    accuracy                           0.93      4572
   macro avg       0.94      0.92      0.93      4572
weighted avg       0.93      0.93      0.93      4572

roc auc score :  0.9248919345904969


Confusion Matrix:
 [[2517   74]
 [ 241 1740]]
```

Here in Random Forest Classifier we have acquired a *Accuracy score of 93.1%.* The *Cross Val Score of 98.7%.*

2. Logistic Regression.

```
LogisticRegression()


accuracy_score:  0.9426946631671042


Cross Val Score :  0.9890544210427284


classification_report
               precision    recall  f1-score   support

           0       0.95      0.95      0.95      2591
           1       0.93      0.93      0.93      1981

    accuracy                           0.94      4572
   macro avg       0.94      0.94      0.94      4572
weighted avg       0.94      0.94      0.94      4572

roc auc score :  0.9415372320331454


Confusion Matrix:
 [[2462  129]
 [ 133 1848]]
```

Here in Logistic Regression we have acquired a *Accuracy score of 94.2%.* The *Cross Val Score of 98.9%.*

3. MultinomialDB.

```
MultinomialNB()

accuracy_score:  0.8937007874015748

Cross Val Score :  0.9738574032571139

classification_report
              precision    recall  f1-score   support

           0       0.87      0.96      0.91      2591
           1       0.94      0.80      0.87      1981

    accuracy                           0.89      4572
   macro avg       0.90      0.88      0.89      4572
weighted avg       0.90      0.89      0.89      4572

roc auc score :  0.8831580446507356

Confusion Matrix:
 [[2493   98]
 [ 388 1593]]
```

Here in MultinomialDB we have acquired a *Accuracy score of 89.3%.*

The *Cross Val Score of 97.3%.*

4. Decision Tree Classifier.

```
DecisionTreeClassifier()

accuracy_score:  0.8801399825021873

Cross Val Score :  0.882396279327029

classification_report
              precision    recall  f1-score   support

           0       0.90      0.89      0.89      2591
           1       0.86      0.86      0.86      1981

    accuracy                           0.88      4572
   macro avg       0.88      0.88      0.88      4572
weighted avg       0.88      0.88      0.88      4572

roc auc score :  0.8783242034370908

Confusion Matrix:
 [[2311  280]
 [ 268 1713]]
```

Here in Decision Tree Classifier we have acquired a *Accuracy score of 88.0%.* The *Cross Val Score of 88.2%.*

5. Gradient Boosting Classifier

```
GradientBoostingClassifier()


accuracy_score:  0.9192913385826772


Cross Val Score :  0.9813808203679828


classification_report
              precision    recall  f1-score   support

           0       0.94      0.92      0.93      2591
           1       0.90      0.92      0.91      1981

    accuracy                           0.92      4572
   macro avg       0.92      0.92      0.92      4572
weighted avg       0.92      0.92      0.92      4572

roc auc score :  0.9189873267285839


Confusion Matrix:
 [[2387  204]
 [ 165 1816]]
```

Here in Gradient Boosting Classifier we have acquired a *Accuracy score of 91.9%.* The *Cross Val Score of 98.1%.*

Final Results:

	Model	Accuracy Score	Cross Val Score	Roc_Auc_curve
0	RandomForestClassifier	93.307087	98.797579	92.698525
1	LogisticRegression	94.269466	98.905442	94.153723
2	GradientBoostingClassifier	91.929134	98.138082	91.898733
3	DecisionTreeClassifier	88.079615	88.270743	87.914082
4	MultinomialNB	89.370079	97.385740	88.315804

Observation:

1. We could visibly see that the "Logistic Regression" is performing well and good, when compared to all other ML Models.

Thus we will consider "Logistic Regression" as our final ML Model.

2. We will now use "GridSearchCV" to hyper tune the model for better efficiency.

Hypertunning the best model using GridSearchCV:

Finding the best parameters for GridSearchCV

```
# finding the best parameters for GridSearchCV
from sklearn.model_selection import GridSearchCV
def grid_cv(mod,parameters,scoring):
    clf = GridSearchCV(mod,parameters,scoring, cv=10)
    clf.fit(x,y)
    print(clf.best_params_)
# GridSearchCV
lr=LogisticRegression()
parameters={'penalty': ['l1', 'l2'],'C':[0.001,.009,0.01,.09,1,5,10,25]}
grid_cv(lr,parameters,'accuracy')
```

```
{'C': 25, 'penalty': 'l2'}
```

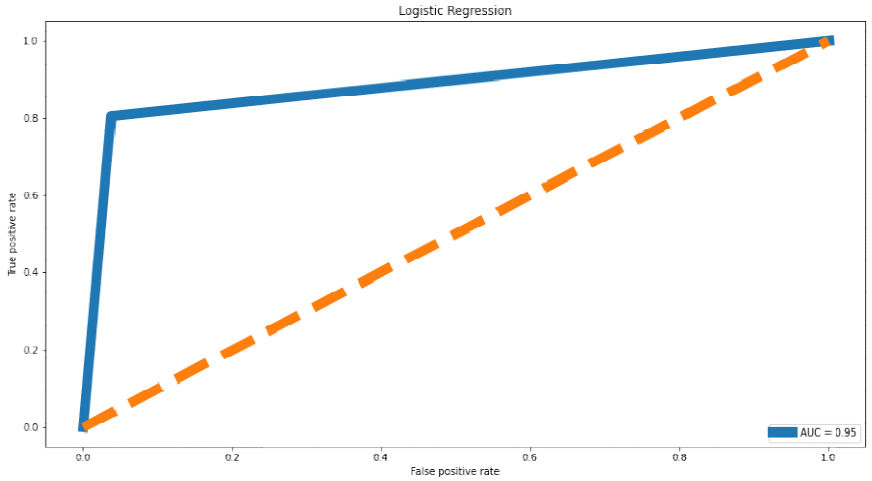
Using the GridSearchCV:

```
# Using the GridSearchCV
clf_lr = LogisticRegression(C=25,penalty='l2')
clf_lr.fit(x_train,y_train)
LR_pred = clf_lr.predict(x_test)
print("Accuracy score: ",accuracy_score(y_test,LR_pred)*100)
print('Cross validation score: ',cross_val_score(clf_lr,x,y,cv=3,scoring='accuracy').mean()*100)
print('Classification report: \n')
print(classification_report(y_test,LR_pred))
print('Confusion matrix: \n')
print(confusion_matrix(y_test,LR_pred))
```

```
Accuracy score:  95.42869641294838
Cross validation score:  95.58654634946677
Classification report:

              precision    recall  f1-score   support

           0       0.96      0.96      0.96      2591
           1       0.94      0.95      0.95      1981

    accuracy                           0.95      4572
   macro avg       0.95      0.95      0.95      4572
weighted avg       0.95      0.95      0.95      4572

Confusion matrix:

[[2479  112]
 [  97 1884]]
```

Plotting AUC_ROC curve

```
#Plotting AUC_ROC curve
fpr,tpr,thresholds=roc_curve(LR_pred,y_test)
roc_auc=auc(fpr,tpr)
plt.figure(figsize=(15,8))
print('AUC_ROC curve:\n')
plt.title("Logistic Regression")
plt.plot(false_positive_rate,true_positive_rate, lw=10, label='AUC = %0.2f'% roc_auc)
plt.plot([0,1],[0,1],lw=10,linestyle='--')
plt.legend(loc='lower right')
plt.xlabel('False positive rate')
plt.ylabel('True positive rate')
plt.show()
```



Observations:

1. With this improved Accuracy and precision we will finalize "Logistic Regression" as our final model.

2. Now following this we will try to predict then news as fake or not using the ML model.

3. The Auc_Roc curve shows the model is performing well.

4. AUC = 0.96

Predictions using the built ML model:

```
prediction = pd.DataFrame(data = y_test,)
prediction['Predicted values'] = LR_pred
prediction.to_csv('Fake_news_Predict.csv')
prediction.tail(10)
```

	label	Predicted values
555	0	0
12760	1	1
17547	0	0
8506	0	0
7126	0	0
1280	1	1
15383	0	0
20284	0	0
8210	1	1
16114	1	1

Saving the job file:

```
# Creating Pickle File
import joblib
joblib.dump(clf_lr,'Fake_news_Prediction.pkl')
```

```
['Fake_news_Prediction.pkl']
```

Manually Testing the Model:

```
def output_label(n):
    if n == 1:
        return "Fake News"
    else:
        return "Not a Fake News"
```

```
def manual_testing(news):
    testing_news = {"text" : [news]}
    new_def_test = pd.DataFrame(testing_news)
    new_def_test["text"] = new_def_test["text"].apply(clean_txt)
    new_x_test = new_def_test["text"]
    new_xv_test = tfidf.transform(new_x_test)
    pred_LR = clf_lr.predict(new_xv_test)
    return print("\n Logistic_Regression_prediction : {}".format(output_label(pred_LR)))
```

Test for Not Fake News:

```
news = str(input())
manual_testing(news)
```

```
A few weeks after the election of Donald J. Trump, pundits with their eyes glued to Twitter believed theyâ€™d finally deciphere
d the master plan behind the  â€™s tweeting. Every time he detonated a  bomb on Twitter, they suspected, it was a sly bid to d
ivert the public eye from more serious news about his impending administration. So when Mr. Trump reignited the dormant debate
over flag burning one morning  â€”  tweeting that those who set fire to the flag should suffer â€œloss of citizenship or year
in jail!â€  â€”  New York magazineâ€™s Jonathan Chait quickly produced a column decoding the message. He called it a â€œstran
ge fightâ€ and a â€œclassic  nationalist distractionâ€ that proves Mr. Trumpâ€™s â€œdangerous and authoritarian politics is
calculated, and not merely crazy. â€ But soon a competing theory emerged: Minutes before Mr. Trumpâ€™s tweet, â€œFox  Friends,
â€ one of the  â€™s favorite shows, ran a segment referencing an American flag burned on a college campus to protest his victo
ry. He wasnâ€™t carrying out some strategy. He was  his TV. Twitter is an impulsive medium. Log in and youâ€™re greeted immedi
ately with a text box asking â€œWhatâ€™s happening?â€  â€”  as in, right this second. Mr. Trump, who is set to become the nat
ionâ€™s 45th president on Friday, is a master of Twitter, but also in its thrall. Theories of some grand Trump Twitter plot for
get that impetuous and aggressive tweeting has been a habit of Mr. Trumpâ€™s for years. He famously avoids alcohol Twitter is h
is vice. And after eight years of  posts from @BarackObama, equally  journalists are transfixed by Mr. Trumpâ€™s timeline,
```

```
Logistic_Regression_prediction : Not a Fake News
```

Test for Fake News:

```
news = str(input())
manual_testing(news)
```

"Home / Foreign Affairs / The Fix is In â€" Russia Kicked Off UN Human Rights Council, While Terrorist Saudi Arabia Re-Elected The Fix is In â€" Russia Kicked Off UN Human Rights Council, While Terrorist Saudi Arabia Re-Elected Jay Syrmopoulos October 29, 2016 1 Comment New York, NY â€" Russia lost an election to the UN Human Rights Council (UNHRC) for the first time since the councilâ€™s inception in 2006 â€" narrowly being beaten out by Croatia â€" as arguably the biggest supporter of terrorism in the world, Saudi Arabia, was re-elected to the council in spite of strong condemnation from global human rights organizations. The 47 council seats are elected for a three-year term and regionally distributed, with staggered elections for one third of the seats every year. Russia had just completed a three-year term running against both Hungary and Croatia for the two available seats from Eastern Europe. After the U.S. lobbied heavily against the Russians, Hungary finished substantially ahead in the voting, with Croatia receiving 114 votes and Russia garnering only 112 votes of the 193 member countries. â€œIt was a very close vote and very good countries competing, Croatia, Hungary. They are fortunate because of their size, they are not exposed to the winds of international diplomacy. Russia is very exposed. Weâ€™ve been in the UNHRC for several years, and I am sure next time we will stand and get back in,â€ Russiaâ€™s UN envoy Vitaly Churkin said. Russia will be eligible to run for a seat on the UN HRC again next year. Staunch U.S. ally, and renowned global terror exporter, Saudi Arabia, easily made it onto the council with 152 votes on the Asian ballot, and will represent the region alongside Iraq, Japan and China for the next three years. Accor

```
Logistic_Regression_prediction : Fake News
```

CONCLUSION:

- **The "*Logistic Regression*" works well with the given dataset.**
- **The manual testing gives a very positive result.**

Limitations:

- **Lack of more information leads to minimized usage of the ML models potential.**
- **More feature engineering is need for the data-set.**
- **With the help of more supporting information the model can be used efficiently for all news sectors.**

End