FLIP ROBO

FAKE NEWS PROJECT

Submitted by:
AKANKSHA PADHYE

ACKNOWLEDGMENT

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

I am highly indebted to Flip Robo Technologies, Bangalore for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

I want to thank my SME Khushboo Garg for providing the Dataset and helping us to solve the problem and addressing out our Query in right time.

I would like to express my gratitude towards my parents & members of Flip Robo for their kind co-operation and encouragement which help me in completion of this project.

I would like to express my special gratitude and thanks to our institute DataTrained & others seen unseen hands which have given us direct & indirect help in completion of this project. With help of their brilliant guidance and encouragement, I was able to complete my tasks properly and were up to the mark in all the tasks assigned. During the process, I got a chance to see the stronger side of my technical and non-technical aspects and also strengthen my concepts.

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.

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.

Conceptual Background of the Domain Problem

Fake news is defined as a made-up story with an intention to deceive or to mislead. The rate of production of fake news has increased exponentially. In the past news obtained from newspaper, radio or TV were considered as the best and authentic source of information about the real world and ongoing situations but now everything has changed. In the run of popularity and ill mind set the media houses and social media are spreading fake news. It’s becoming harder and harder to say whether a piece of news is real or fabricated.

The effect of fake news can be seen everywhere. The fake news leads to communal disturbance, character assassination, mental trauma, sometimes it is used as a weapon to achieve some illicit plans etc. these are like wild fire which spread too quickly and difficult to control. Which creates difficulty in differentiating between fake news and authentic news.

Review of Literature

Technologies such as Artificial Intelligence (AI) and Natural Language Processing (NLP) tools offer great promise for researchers to build systems which could automatically detect fake news. However, detecting fake news is a challenging task to accomplish as it requires models to summarize the news and compare it to the actual news in order to classify it as fake. Moreover, the task of comparing proposed news with the original news itself is a daunting task as it's highly subjective and opinionated.

Motivation for the Problem Undertaken

The goal is to build a prototype to classify the news as fake or not fakes in order to bring awareness and reduce unwanted chaos.

➔ To apply data pre-processing and preparation techniques in order to obtain clean data.
➔ Explore the effectiveness of multiple machine learning approaches and select the best for this problem.

ANALYTICAL PROBLEM FRAMING

Model Building Phase

You need to build a machine learning model. Before model building do all data pre-processing steps involving NLP. Try different models with different hyper parameters and select the best model. Follow the complete life cycle of data science. Include all the steps like-

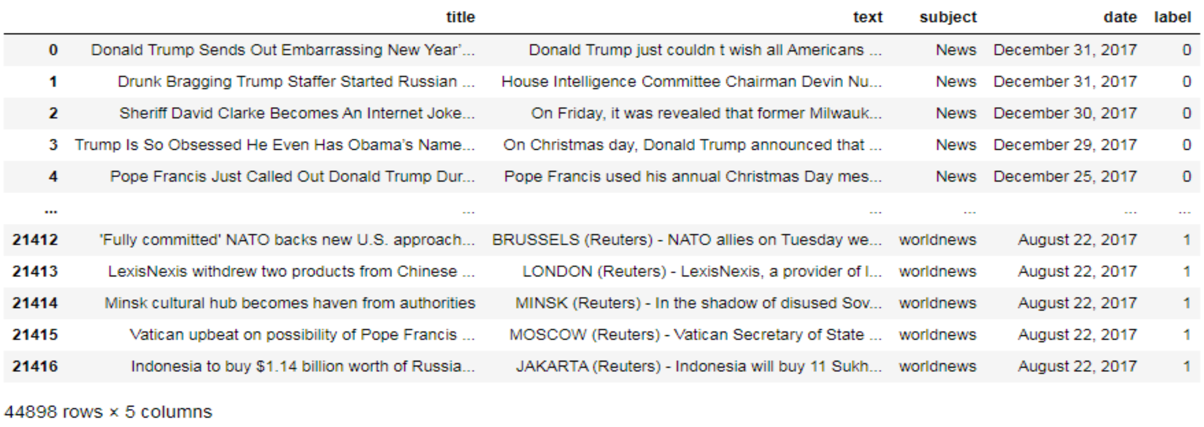
1. Data Cleaning
2. Exploratory Data Analysis
3. Data Pre-processing
4. Model Building
5. Model Evaluation
6. Selecting the best model

Data Sources and their formats

There are two datasets one for fake news and one for true news. In true news, there is 21417 news, and in fake news, there is 23481 news. The description of each ofthe column is given below:

1. **Title:** It is the title of the news.
2. **Text:** It contains the full text of the news article
3. **Subject:** It represents the category of the news
4. **Date:** It represents the date of the news article
5. **Label:** It tells whether the news is fake (1) or not fake (0).

The sample data for the reference is as shown below:

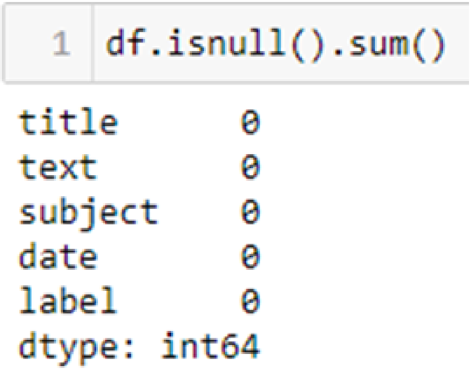


	title	text	subject	date	label
0	Donald Trump Sends Out Embarrassing New Year'...	Donald Trump just couldn t wish all Americans ...	News	December 31, 2017	0
1	Drunk Bragging Trump Staffer Started Russian ...	House Intelligence Committee Chairman Devin Nu...	News	December 31, 2017	0
2	Sheriff David Clarke 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
...	...	...	...	...	...
21412	'Fully committed' NATO backs new U.S. approach...	BRUSSELS (Reuters) - NATO allies on Tuesday we...	worldnews	August 22, 2017	1
21413	LexisNexis withdrew two products from Chinese ...	LONDON (Reuters) - LexisNexis, a provider of l...	worldnews	August 22, 2017	1
21414	Minsk cultural hub becomes haven from authorities	MINSK (Reuters) - In the shadow of disused Sov...	worldnews	August 22, 2017	1
21415	Vatican upbeat on possibility of Pope Francis ...	MOSCOW (Reuters) - Vatican Secretary of State ...	worldnews	August 22, 2017	1
21416	Indonesia to buy $1.14 billion worth of Russia...	JAKARTA (Reuters) - Indonesia will buy 11 Sukh...	worldnews	August 22, 2017	1

44898 rows × 5 columns

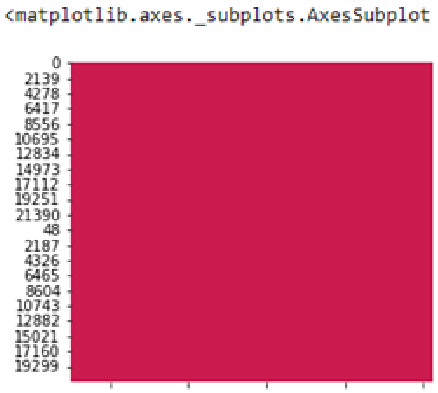
Checking the missing values

Moving ahead, cleaning the dataset will remove errors which in turn will increase productivity and render highest quality information in decision making. Here, in this dataset, the null values are checked using isnull().sum(), where we can see that there no null values present in the datset. The same has been visualized using heatmap.



```
df.isnull().sum()
```

```
title      0
text       0
subject    0
date       0
label      0
dtype: int64
```



```
<matplotlib.axes._subplots.AxesSubplot
```

Pre-processing using NLP

Data cleaning is the process of preparing data for analysis by removing or modifying data that is incorrect, incomplete, irrelevant, duplicated, or improperly formatted. This data is usually not necessary or helpful when it comes to analysing data because it may hinder the process or provide inaccurate results.

Before cleaning the data, a new column is created named ‘length_before_cleaning’ which shows the total length of the news respectively before cleaning the text.

The following steps were taken in order to clean the text:

- Transform the text into lower case.
- Replaced the email addresses with the text 'emailaddress'
- Replaced the URLs with the text 'webaddress'
- Removed the HTML tags
- Removed the numbers
- Removed extra newlines
- Removed the punctuations
- Removed the unwanted white spaces
- Removed the remaining tokens that are not alphabetic
- Removed the stop words

Tokenization

Word tokenization is the process of splitting a large sample of text into words. This is a requirement in natural language processing tasks where each word needs to be captured and subjected to further analysis.

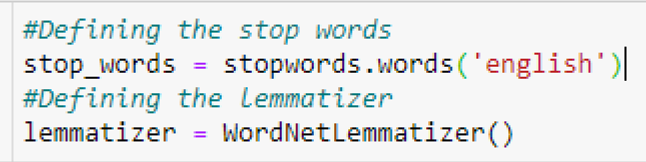
After cleaning the text, each comment i.e., the corpus is split into words. Thus, the text is tokenized into words using word_tokenize().

Lemmatization

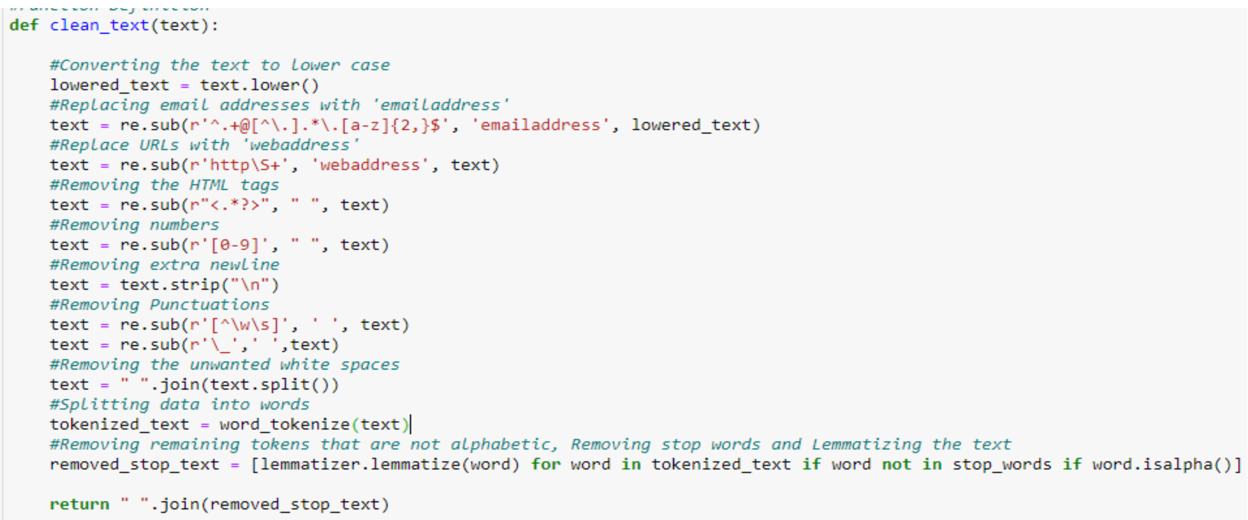
Lemmatization in NLTK refers to the morphological analysis of words, which aims to remove inflectional endings. It helps in returning the base or dictionary form of a word known as the lemma. The NLTK Lemmatization method is based on WorldNet's built-in morph function.Thus, the words are lemmatized using WordNetLemmatizer() after importing the necessary library to perform the same and then creating the instance for it.

All the text cleaning or the above steps are performed by defining a function and applying the same using apply() to the ‘News’ column of the dataset. Below is the code shown:

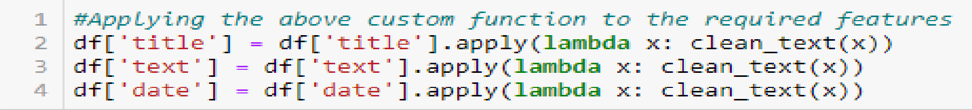
```
#Defining the stop words
stop_words = stopwords.words('english')

#Defining the lemmatizer
lemmatizer = WordNetLemmatizer()
```



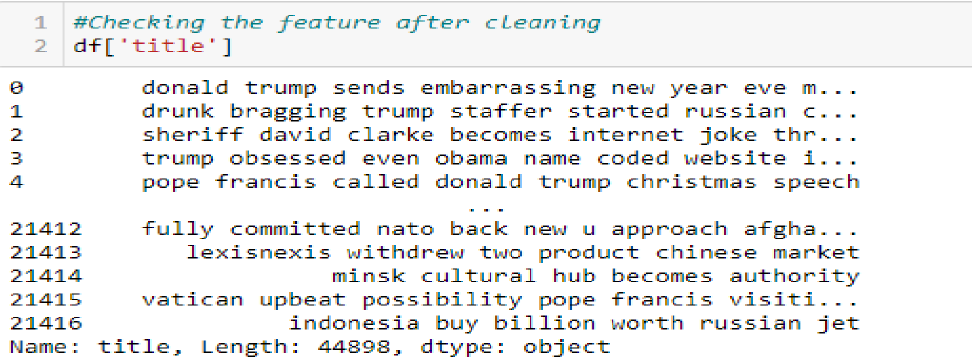
```
#Defining the stop words
stop_words = stopwords.words('english')
#Defining the lemmatizer
lemmatizer = WordNetLemmatizer()
```



```
def clean_text(text):

    #Converting the text to lower case
    lowered_text = text.lower()
    #Replacing email addresses with 'emailaddress'
    text = re.sub(r'^.+@[^\.].*\.[a-z]{2,}$', 'emailaddress', lowered_text)
    #Replace URLs with 'webaddress'
    text = re.sub(r'http\S+', 'webaddress', text)
    #Removing the HTML tags
    text = re.sub(r"<.*?>", " ", text)
    #Removing numbers
    text = re.sub(r'[0-9]', " ", text)
    #Removing extra newline
    text = text.strip("\n")
    #Removing Punctuations
    text = re.sub(r'[^\w\s]', ' ', text)
    text = re.sub(r'\_',' ',text)
    #Removing the unwanted white spaces
    text = " ".join(text.split())
    #Splitting data into words
    tokenized_text = word_tokenize(text)
    #Removing remaining tokens that are not alphabetic, Removing stop words and Lemmatizing the text
    removed_stop_text = [lemmatizer.lemmatize(word) for word in tokenized_text if word not in stop_words if word.isalpha()]

    return " ".join(removed_stop_text)
```



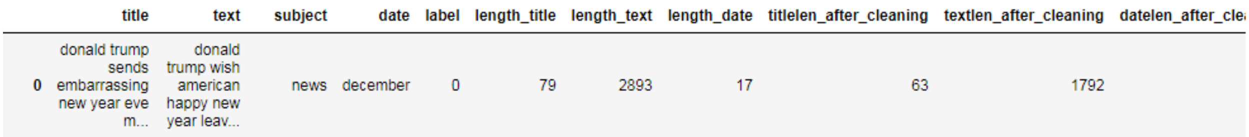
```
#Applying the above custom function to the required features
df['title'] = df['title'].apply(lambda x: clean_text(x))
df['text'] = df['text'].apply(lambda x: clean_text(x))
df['date'] = df['date'].apply(lambda x: clean_text(x))
```



```
#Checking the feature after cleaning
df['title']
```

```
0        donald trump sends embarrassing new year eve m...
1        drunk bragging trump staffer started russian c...
2        sheriff david clarke becomes internet joke thr...
3        trump obsessed even obama name coded website i...
4        pope francis called donald trump christmas speech
                               ...
21412    fully committed nato back new u approach afgha...
21413       lexisnexis withdrew two product chinese market
21414                    minsk cultural hub becomes authority
21415    vatican upbeat possibility pope francis visiti...
21416                indonesia buy billion worth russian jet
Name: title, Length: 44898, dtype: object
```

We also created new features for comparing the original length before cleaning and the new length after cleaning.

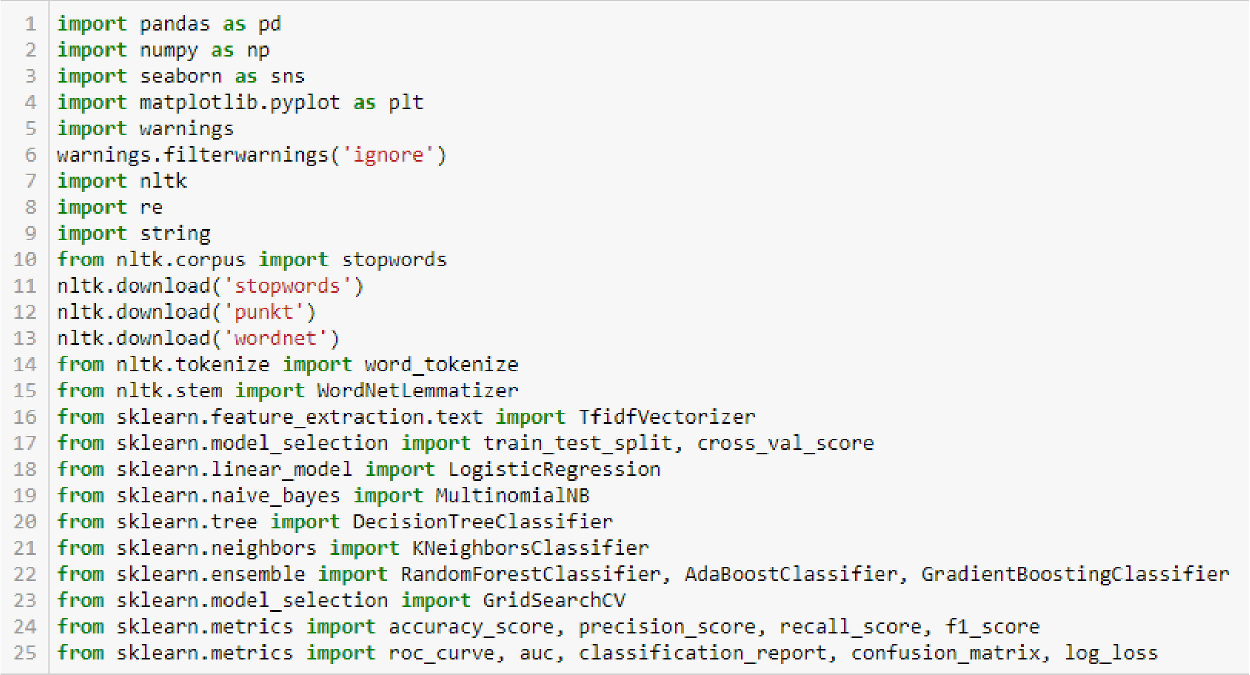


	title	text	subject	date	label	length_title	length_text	length_date	titlelen_after_cleaning	textlen_after_cleaning	datelen_after_clea
0	donald trump sends embarrassing new year eve m...	donald trump wish american happy new year leav...	news	december	0	79	2893	17	63	1792	

HARDWARE AND SOFTWARE REQUIREMENTS AND TOOLS USED

For doing this project, the hardware used is a laptop with high end specification and a stable internet connection. While coming to software part, I had used anaconda navigator and in that I have used Jupyter notebook to do my python programming and analysis.

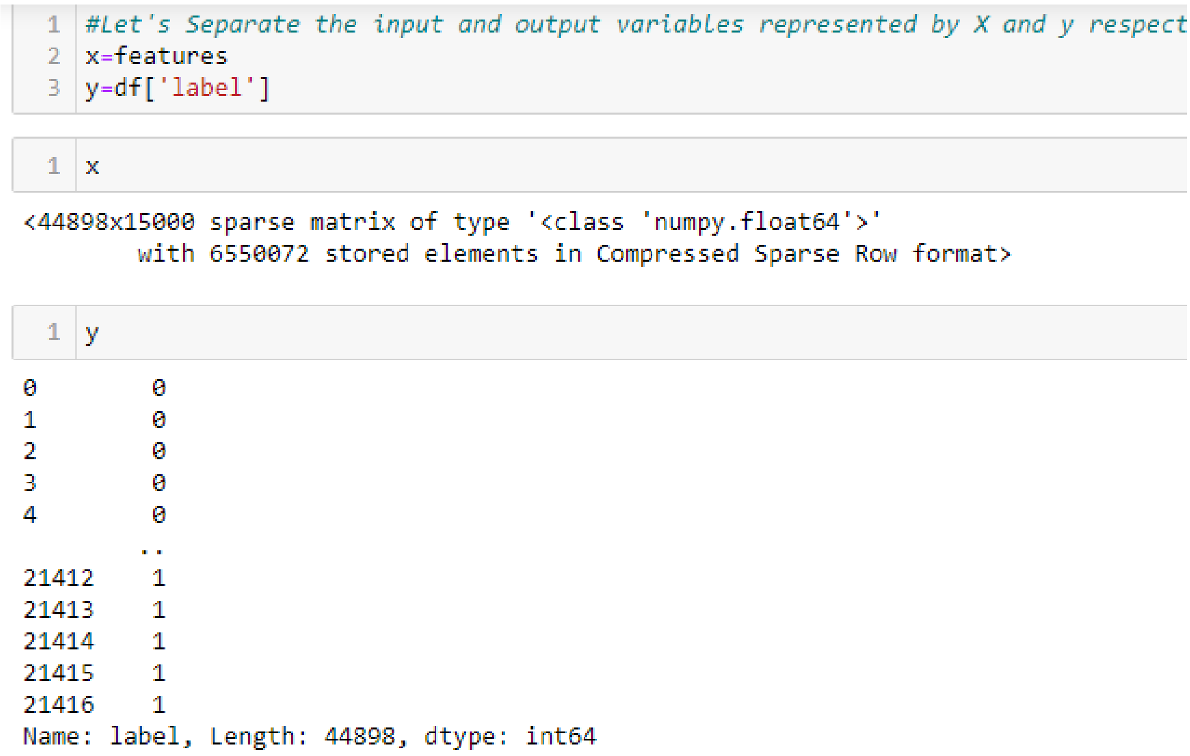
For using an CSV file, Microsoft excel is needed. In Jupyter notebook, I had used lots of python libraries to carry out this project and I have mentioned below with proper justification:



```python
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
import nltk
import re
import string
from nltk.corpus import stopwords
nltk.download('stopwords')
nltk.download('punkt')
nltk.download('wordnet')
from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import MultinomialNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier, GradientBoostingClassifier
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
from sklearn.metrics import roc_curve, auc, classification_report, confusion_matrix, log_loss
```

MODELS DEVELOPMENT AND EVALUATION

Separating the input and output variables



```
#Let's Separate the input and output variables represented by X and y respect
x=features
y=df['label']
```

```
x
```

```
<44898x15000 sparse matrix of type '<class 'numpy.float64'>'
	with 6550072 stored elements in Compressed Sparse Row format>
```

```
y
```

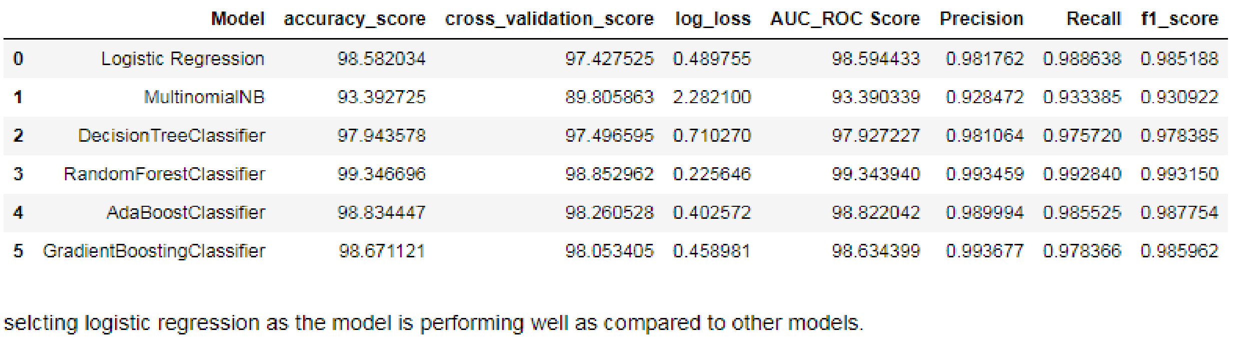
```
0        0
1        0
2        0
3        0
4        0
        ..
21412    1
21413    1
21414    1
21415    1
21416    1
Name: label, Length: 44898, dtype: int64
```

Splitting the train and test data

Training and testing the models minimize the effects of data discrepancies and better understand the characteristics of the model. The **training** data is used to make the machine recognize patterns in the data and the **test** data is used to see how well the machine can predict new answers based on its training. ‘X’ and ‘y’ were split for training and testing using train_test_split in a ratio of 70:30 respectively.

Building the model

After running various algorithms, the final results are as follows:



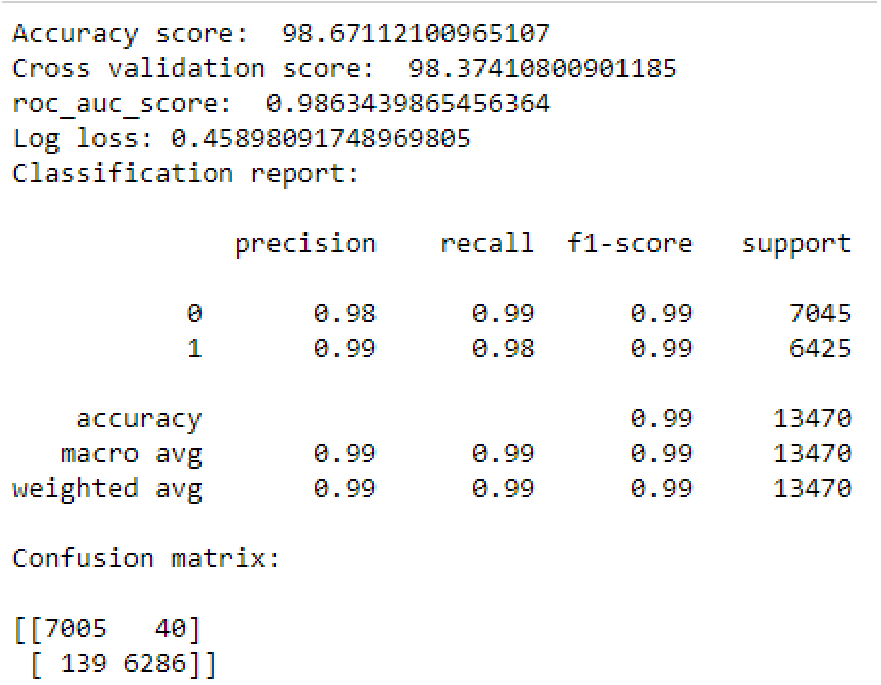
	Model	accuracy_score	cross_validation_score	log_loss	AUC_ROC Score	Precision	Recall	f1_score
0	Logistic Regression	98.582034	97.427525	0.489755	98.594433	0.981762	0.988638	0.985188
1	MultinomialNB	93.392725	89.805863	2.282100	93.390339	0.928472	0.933385	0.930922
2	DecisionTreeClassifier	97.943578	97.496595	0.710270	97.927227	0.981064	0.975720	0.978385
3	RandomForestClassifier	99.346696	98.852962	0.225646	99.343940	0.993459	0.992840	0.993150
4	AdaBoostClassifier	98.834447	98.260528	0.402572	98.822042	0.989994	0.985525	0.987754
5	GradientBoostingClassifier	98.671121	98.053405	0.458981	98.634399	0.993677	0.978366	0.985962

selcting logistic regression as the model is performing well as compared to other models.

After running the algorithms and according to the scores of performance metrics and other scores, we can see that Logistic Regression algorithms is performing well. Now, we will perform Hyperparameter Tuning to find out the best parameters and try to increase the scores.

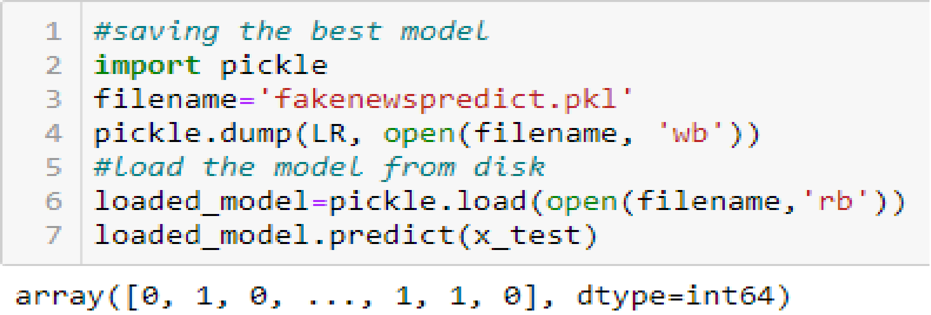
Hyperparameter Tuning

After tuning the model, we got logistic regression as the best performing model and the code is given below:



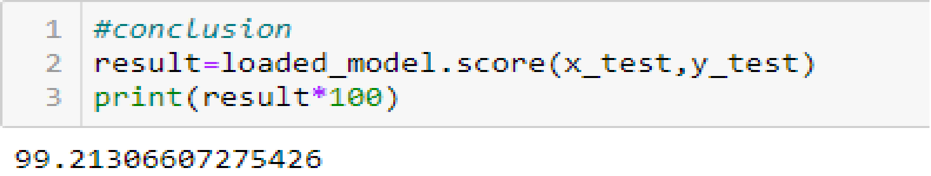
```
Accuracy score:  98.67112100965107
Cross validation score:  98.37410800901185
roc_auc_score:  0.9863439865456364
Log loss: 0.45898091748969805
Classification report:

              precision    recall  f1-score   support

           0       0.98      0.99      0.99      7045
           1       0.99      0.98      0.99      6425

    accuracy                           0.99     13470
   macro avg       0.99      0.99      0.99     13470
weighted avg       0.99      0.99      0.99     13470

Confusion matrix:

[[7005   40]
 [ 139 6286]]
```

Finalizing the model



```
#saving the best model
import pickle
filename='fakenewspredict.pkl'
pickle.dump(LR, open(filename, 'wb'))
#load the model from disk
loaded_model=pickle.load(open(filename,'rb'))
loaded_model.predict(x_test)
```

```
array([0, 1, 0, ..., 1, 1, 0], dtype=int64)
```



```
#conclusion
result=loaded_model.score(x_test,y_test)
print(result*100)
```

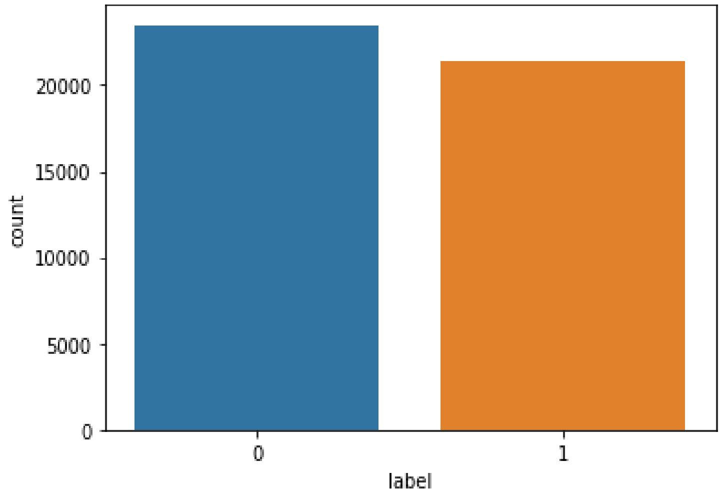
```
99.21306607275426
```

After predicting using test data, we will store the results in a csv file. The final loaded model is having the accurarcy of 99% which is the best for any model.

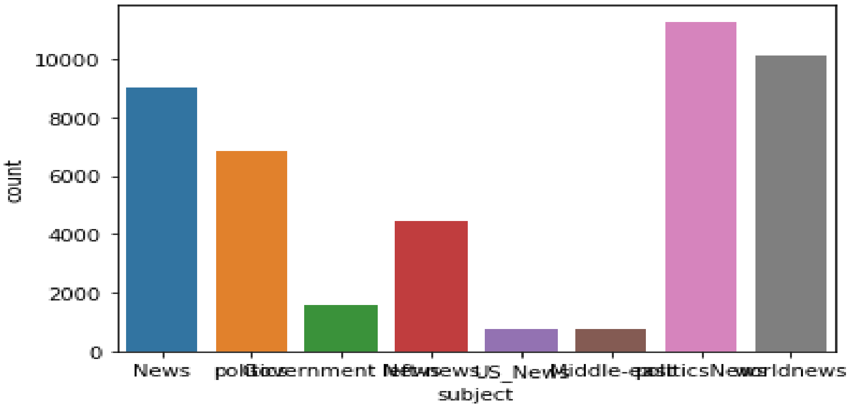
DATA VISUALIZATION

Plotting countplot for label

```
0    23481
1    21417
Name: label, dtype: int64
```

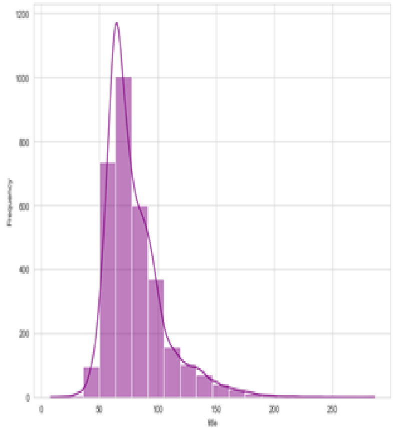


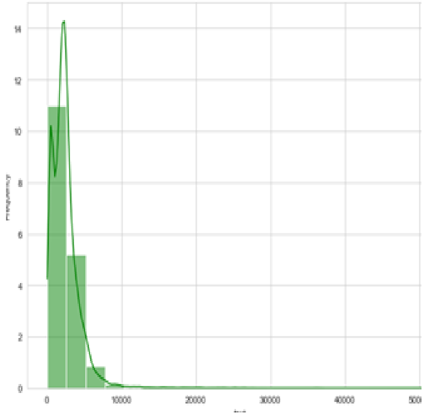
```
politicsNews       11272
worldnews          10145
News                9050
politics            6841
left-news           4459
Government News     1570
US_News              783
Middle-east          778
Name: subject, dtype: int64
```

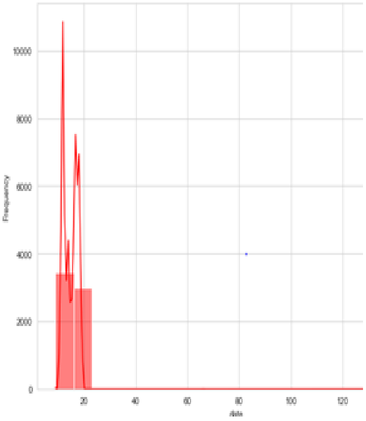


- Targeted variable has two values. 0 stands for fake news and 1 stands for true news. Fake news counts are 23481 which is more than true news with count 21417.
- Maximum news subject is on politicsnews followed by worldnews and the least news is from subject USnews and middle-east.

Histplots for checking the distribution







- In title column majority comments are between 65-75, where maximum length is 1000
- In text column maximum text is between 0-2000 and the maximum frequency is 11
- In date column most of the date is between 5-10, where frequency is 3000

CONCLUSION

The finding of the study is that when the news's are being published on a bogus name, the author names not available that news are end up being Fake, and also, we can understand this fake news's are desperately being spread among the public to create a fake image of an individual, or to get profit out of it or to destroy the good deeds of the target person.

Learning Outcomes of the Study in respect of Data Science

The universe of "fake news" is much larger than simply false news stories. Some stories may have a nugget of truth, but lack any contextualizing details. They may not include any verifiable facts or sources. Some stories may include basic verifiable facts, but are written using language that is deliberately inflammatory, leaves out pertinent details or only presents one viewpoint. "Fake news" exists within a larger ecosystem of mis- and disinformation.

Misinformation is false or inaccurate information that is mistakenly or inadvertently created or spread; the intent is not to deceive. Disinformation is false information that is deliberately created and spread "in order to influence public opinion or obscure the truth"

-(https://www.merriam-webster.com/dictionary/disinformation).

As per our evaluation, we found that lesser number of Authors or bogus names or authors unknown have released fake news. We trained 20800 observations for five context categories using a Random Forest algorithm for context detection. Then, the system classifies the fake news in one of the trained contexts in the text conversation. In our testbed, we observed 48.41% of records have fake news but if we search for the authors names in fake news only 10% of the authors spread almost all the fake news. Hence, our proposed approach can identify the Fake news and the authors who spread fake news, as discussed usually on a no source news or on a bogus name these fake news's are spread.

Limitations of this work and Scope for Future Work

The limitation of the study is that this data was taken in a shorter time frame on a current trend which might help us in a prediction for a shorted period of time. So, if the prediction of fake news was done with very old data with our model there are chances that the prediction won't be accurate. Same applies for not immediate future data. So, in such case if we have analysis the trend of the news, and if we split the news category as politics, sports arts, general, local, international then we might get some accurate prediction.

Problems faced while working in this project:

- More computational power was required.
- More missing data were present in the dataset.
- Loss was more for some algorithms.