FAKE NEWS CLASSIFICATION

SUBMITED BY
RUTUJA PATIL

ACKNOWLEDGMENT

I would like to express my gratitude for the opportunity and would like to thank Flip Robo Technologies for giving me an opportunity to work on this project of Fake News Classification. While going through this project I could somewhere find how text can be converted to vectors, and those vectors can be used for Fake News classification. More I knew that how these Fake News are sent to a large group of recipients without their prior consent and these consents gets spread like a fire, which actually happened at the time of COVID typically advertising for goods and services or business opportunities. In recent days the percentage of FAKE NEWS and SMS messages have gone very high. Even many frauds have been reported to be police and many got even highlighted in newspapers.

I am very grateful to DATA TRAINED team for providing me the adequate Trainings which actually helped me a lot to completion of this project. I took help from Mr. Mohd Kashif and the document links and study materials provided during project completion was very helpful.

During the completion of the project, I found various issues working with NLP and I overcome those problem with the help of Anaconda, Java T point, Kaggle and Medium.

INTRODUCTION

1. BUSINESS PROBLEM FRAMING

There are three main types of fake news, and all need to be tackled in different ways, by society and business alike. The first type, what we will call "deliberate fake news", is arguably the most "fake". This is the spreading of intentionally incorrect information (by bots or otherwise) with the goal of changing a societal outcome. This is the kind that has been disseminated on social media to change election outcomes. It isn't new, but has greater reach in a hyper-connected world. The second type is not so much 'fake' as it is misleading or serving an agenda. This is not new in any way, shape or form. Propaganda and biased news are the result of unscrupulous reporting, or politically-aligned publishers.
Third, we have unintentionally fake news. False news spreads faster than the truth, but it isn't necessarily due to nefarious intentions.

2. CONCEPTUAL BACKGROUND OF THE DOMAIN PROBLEM

Fake news's simple meaning is to incorporate information that leads people to the wrong path. Nowadays fake news spreading like water and people share this information without verifying it. This is often done to further or impose certain ideas and is often achieved with political agendas. For media outlets, the ability to attract viewers to their websites is necessary to generate online advertising revenue. So it is necessary to detect fake news.

3. REVIEW OF LITERATURE

This Data Set is downloaded from Kaggle and is being shared us by our SME, provided by Flip Robo Technologies. There are two datasets one is Fake and other is True News. In True news there are 21417 rows and 4 columns and False news are having 23481 rows and 4 columns. Label added of True and False as the news. Both were merged ignore index updated as True. Both data base label was updated. Merger shows an observation of 44898 rows and 5 columns.

4. MOTIVATION FOR THE PROBLEM UNDERTAKEN

Fake news has become one of the biggest problems of our age. It has serious impact on our online as well as offline discourse. One can even go as far as saying that, to date, fake news poses a clear and present danger to western democracy and stability of the society.

ANALYTICAL PROBLEM

1. MATHEMATICAL/ ANALYTICAL MODELLING OF THE PROBLEM

The set is firstly drawn with the help of panda's library which is in csv format. These data set having 44898 rows and 5 columns, size of data setstands 224490 and dimension is 2.

- The unique Values and number of Unique Values were checked ofthe data set.
- Columns were checked and info was drawn for the dataset which is maximum.
- Null values were checked along with the Percentage calculation wasdrawn from the Data Set in the form of a Data Frame, hence those columns were dropped.

- Duplicated values were checked, found 209 contents to be duplicatedand those rows were dropped.

- Finally, after Imputation the Visualization was created and Next stepwas taken towards data Analysis.

- Length of the Total characters, words and sentences were drawn tofind the relation between them.

- Various visualization was performed on the same.

2. DATA SOURCES AND THEIR FORMATS

This Data Set was provided by Flip Robo Technology to learn various aspects ofthe NLTK library. The data set contains

i. A collection of 44898 rows and 5 columns and was manually extracted from the Kaggle Web site and other websites.

ii. 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. You have to insert one label column zero for fake news and one for true news. We are combined both datasets using pandas concat method.

iii. This Data Sets contains 05 objects and Memory consumed is 1.7+MB

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 44898 entries, 0 to 44897
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   title         44898 non-null  object
 1   text          44898 non-null  object
 2   subject       44898 non-null  object
 3   date          44898 non-null  object
 4   Class_Labels  44898 non-null  object
dtypes: object(5)
memory usage: 1.7+ MB
```

3. DATA PRE-PROCESSING DONE

Data Pre-processing

- Loading Data set, Checking and imputing null values, Checking Duplicatedrows in columns and eliminating the rows those were duplicated.
- Using Encoding Methods.
- The Length of the Total Characters, words and sentences are captured and various measures were drawn in the form of a number.
- Descriptive analysis was performed.

- Correlation of the data set were checked and correlation were checked among independent variable were checked.

Text Pre- Processing

- Texts were drawn lower using string name. lower function.
- Tokenization was done (sentences were drawn into words)
- Stop Words were used (“English”)
- Texts Stemming were done as Stemming works better on Class Labels.
- Then corpus was joint in the form of a string join.

```
In [54]: def transform(text):
           text=text.lower() # make in lower case
           text=nltk.word_tokenize(text) # tokenized each word

           corpus=[]
           for i in text:
             if i.isalnum():# if text is alpha-numerical
               corpus.append(i) # append text

           new_corpus=corpus.copy() # making a copy
           corpus.clear() # clearing corpus

           for i in new_corpus: # from new corpus

         # using stop words of english language and all punctuations
             if i not in stopwords.words("english") and i not in string.punctuation:
               corpus.append(i) # appending

           new_corpus=corpus.copy() # making a copy
           corpus.clear() #clearing the copy

           ps=PorterStemmer() #importing Porter Stemmer

           for i in new_corpus:
             corpus.append(ps.stem(i)) # used stemming process

           return " ".join(corpus)
```

4. DATA INPUTS- LOGIC- OUTPUT RELATIONSHIPS

- There is 04 independent variable and 01 dependent variable.
- I have used count-plot, cat-plot, bar-plot, pie chart for univariate analysis, bi-variate analysis.
- I have used count plot and cat plot to check the relation between independent and dependent variable.
- We used person correlation to check the relation between variables.
- We have totally used univariate, bivariate and multivariate graph to determine the relation among variables.

- We used Word Clouds to check the frequent used spam words and ham words to draw the relations among the independent variable and dependent variable.

5. STATE THE SET OF ASSUMPTIONS (IF ANY) RELATED TO THE PROBLEM UNDER CONSIDERATION

1. Naïve Bayes classifier

It is a supervised machine learning algorithm where words probabilities play the main rule here. If some words occur often in fake news but not in true news, then this incoming news is probably spam. Naïve bayes classifier technique has become a very popular method in mail filtering software.

2. Artificial Neural Networks classifier:

An artificial neural network (ANN), also called simply a "Neural Network" (NN), is a computational model based on biological neural networks. It consists of an interconnected collection of artificial neurons. An artificial neural network is an adaptive system that changes its structure based on information that flows through the artificial network during a learning phase.

6. HARDWARE AND SOFTWARE REQUIREMENTS AND TOOLS USED

Hardware:

SOFTWARE USED:

- Jupiter Note Book.
- Microsoft Office 2020
- Windows 11 OS

Library used: To run the program and to build the model we need some basic libraries as follows:

- NumPy
- Pandas
- Seaborn
- Matplotlib
- SciPy
- Sklearn
- Pickle
- Imbalance Learn
- NLTK

1) import pandas as pd:

Pandas are a popular Python-based data analysis toolkit which can be imported using import pandas as pd. It presents a diverse range of utilities, ranging from parsing multiple file formats to converting an entire data table into a NumPy matrix array. This makes pandas a trusted ally in data science and machine learning.

2) import NumPy as np:

NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.

3) import seaborn as sns:

Seaborn is a data visualization library built on top of matplotlib and closely integrated with pandas' data structures in Python. Visualization is the central part of Seaborn which helps in exploration and understanding of data.

4) Import matplotlib.pyplot as plt

matplotlib.pyplot is a collection of functions that make matplotlib work like MATLAB. Each pyplot function makes some change to a figure: e.g., creates a figure, creates a plotting area in a figure, plots some lines in a plotting area, decorates the plot with labels, etc.

5. Scipy:

SciPy is a free and open-source Python library used for scientific computing and technical computing. SciPy contains modules for optimization, linear algebra, integration, interpolation, special functions, FFT, signal and image processing, ODE solvers and other tasks common in science and engineering.

6. Sklearn

Scikit-learn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support-vector machines.

classification, regression and clustering algorithms including support-vector machines.

7. Pickle

"Pickling" is the process whereby a Python object hierarchy is converted into a byte stream, and "unpickling" is the inverse operation, whereby a byte stream (from a binary file or bytes-like object) is converted back into an object hierarchy.

8.NLTK

The Natural Language Toolkit, or more commonly NLTK, is a suite of libraries and programs for symbolic and statistical natural language processing for English written in the Python programming language.Scikit-learn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support-vector machines.

9. Pickle

“Pickling” is the process whereby a Python object hierarchy is converted into a byte stream, and “unpickling” is the inverse operation, whereby a byte stream (from a binary file or bytes-like object) is converted back into an object hierarchy.

10. NLTK

The Natural Language Toolkit, or more commonly NLTK, is a suite of libraries and programs for symbolic and statistical natural language processing for English written in the Python programming language.

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

    i. Length of the sentences of each line were drawn into a numeric value in separate column named Total Sentences.

    ii. Length of the words of each line is drawn into a numeric column named Total Words.

    iii. Length of the total characters were drawn into numeric values named Total Characters.

    iv. Most frequent words were checked using word clouds and collection library.

a. Testing of Identified Approaches (Algorithms)

Adaptive system that changes its structure based on information that flows through the artificial network during a learning phase.

1. We have used Naïve Bayes Models:
    - gnb=GaussianNB()
    - mnb=MultinomialNB()
    - bnb=BernoulliNB()

    - We have used Other Models:

    - lg=LogisticRegression()
    - rfc=RandomForestClassifier()
    - etc=ExtraTreesClassifier()
    - gbc=GradientBoostingClassifier()
    - sgd=SGDClassifier()
    - mlp=MLPClassifier()

- Run and evaluate selected models

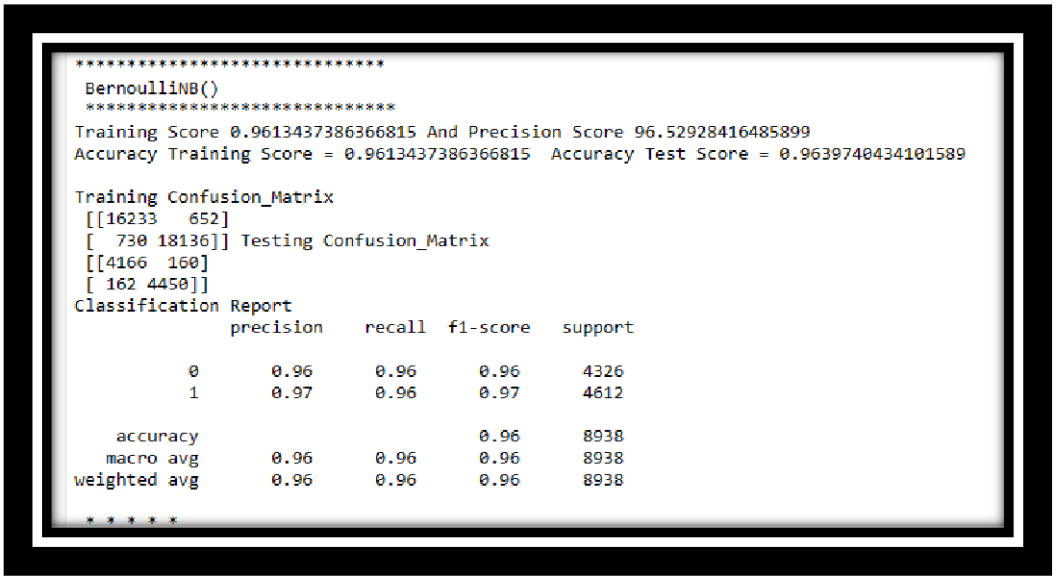
1.

```
*******************************
 GaussianNB()
 *******************************
Training Score 0.9066599535677323 And Precision Score 90.22540095361941
Accuracy Training Score = 0.9066599535677323  Accuracy Test Score = 0.8993063325128664

Training Confusion_Matrix
 [[15193  1692]
 [ 1645 17221]] Testing Confusion_Matrix
 [[3875  451]
 [ 449 4163]]
Classification Report
              precision    recall  f1-score   support

           0       0.90      0.90      0.90      4326
           1       0.90      0.90      0.90      4612

    accuracy                           0.90      8938
   macro avg       0.90      0.90      0.90      8938
weighted avg       0.90      0.90      0.90      8938

 * * * * *
```

2.

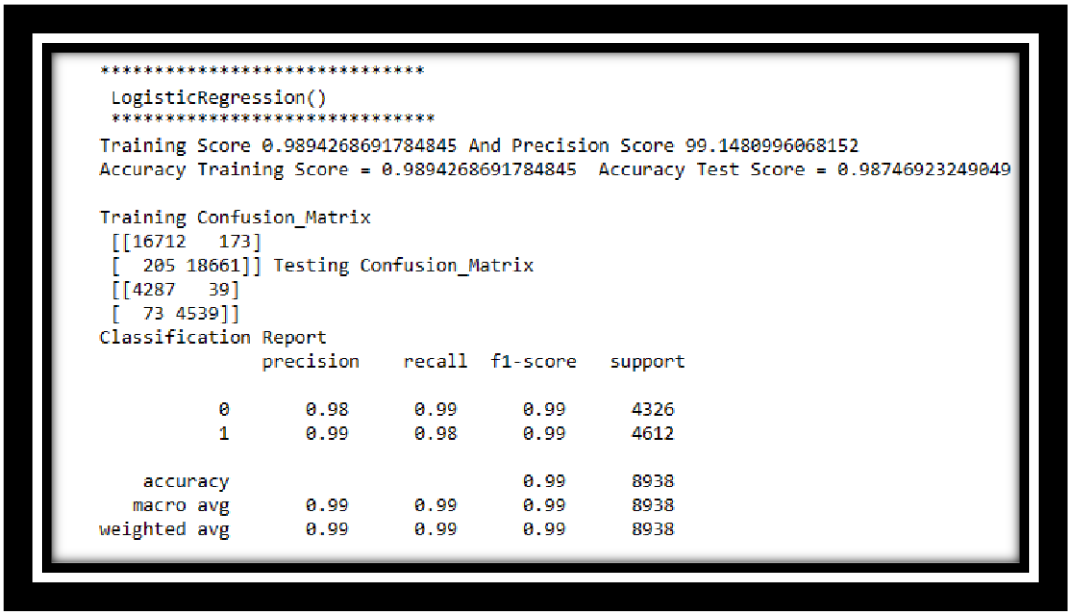
```
*******************************
 MultinomialNB()
 *******************************
Training Score 0.9249531481636877 And Precision Score 92.20252082888271
Accuracy Training Score = 0.9249531481636877  Accuracy Test Score = 0.9260460953233386

Training Confusion_Matrix
 [[15448  1437]
 [ 1246 17620]] Testing Confusion_Matrix
 [[3961  365]
 [ 296 4316]]
Classification Report
              precision    recall  f1-score   support

           0       0.93      0.92      0.92      4326
           1       0.92      0.94      0.93      4612

    accuracy                           0.93      8938
   macro avg       0.93      0.93      0.93      8938
weighted avg       0.93      0.93      0.93      8938

 * * * * *
```

3.



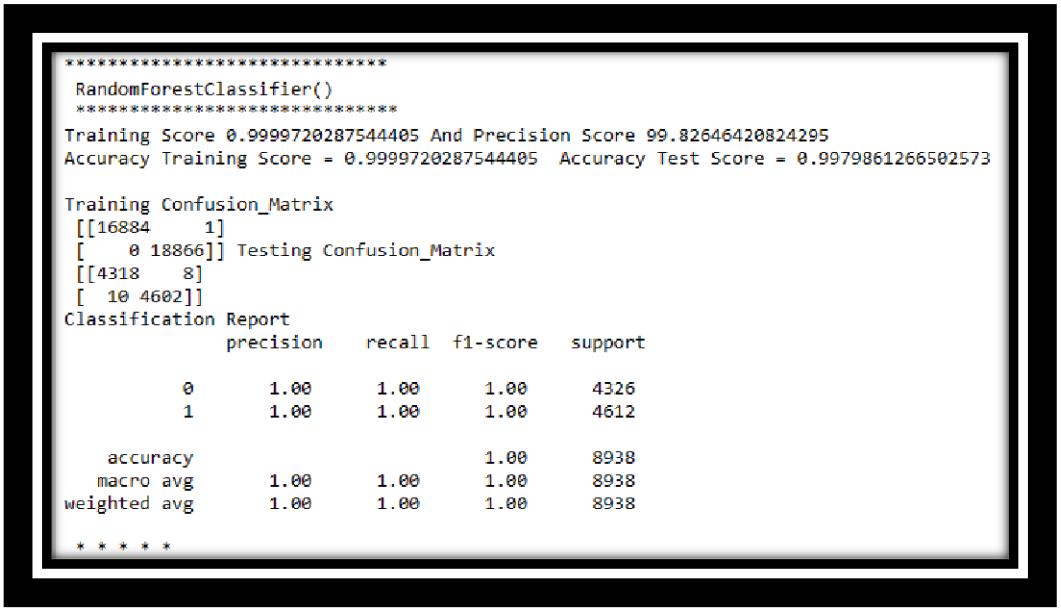
```
*******************************
 BernoulliNB()
 *******************************
Training Score 0.9613437386366815 And Precision Score 96.52928416485899
Accuracy Training Score = 0.9613437386366815  Accuracy Test Score = 0.9639740434101589

Training Confusion_Matrix
 [[16233   652]
 [  730 18136]] Testing Confusion_Matrix
 [[4166  160]
 [ 162 4450]]
Classification Report
              precision    recall  f1-score   support

           0       0.96      0.96      0.96      4326
           1       0.97      0.96      0.97      4612

    accuracy                           0.96      8938
   macro avg       0.96      0.96      0.96      8938
weighted avg       0.96      0.96      0.96      8938

 * * * * *
```

4.



```
*******************************
 LogisticRegression()
 *******************************
Training Score 0.9894268691784845 And Precision Score 99.1480996068152
Accuracy Training Score = 0.9894268691784845  Accuracy Test Score = 0.98746923249049

Training Confusion_Matrix
 [[16712   173]
 [  205 18661]] Testing Confusion_Matrix
 [[4287   39]
 [  73 4539]]
Classification Report
              precision    recall  f1-score   support

           0       0.98      0.99      0.99      4326
           1       0.99      0.98      0.99      4612

    accuracy                           0.99      8938
   macro avg       0.99      0.99      0.99      8938
weighted avg       0.99      0.99      0.99      8938
```

5.



```
*******************************
 RandomForestClassifier()
 *******************************
Training Score 0.9999720287544405 And Precision Score 99.82646420824295
Accuracy Training Score = 0.9999720287544405  Accuracy Test Score = 0.9979861266502573

Training Confusion_Matrix
 [[16884     1]
 [    0 18866]] Testing Confusion_Matrix
 [[4318    8]
 [  10 4602]]
Classification Report
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      4326
           1       1.00      1.00      1.00      4612

    accuracy                           1.00      8938
   macro avg       1.00      1.00      1.00      8938
weighted avg       1.00      1.00      1.00      8938

 * * * * *
```

6.

```
*******************************
 ExtraTreesClassifier()
 *******************************
Training Score 0.9999720287544405 And Precision Score 99.62857767096351
Accuracy Training Score = 0.9999720287544405  Accuracy Test Score = 0.9922801521593198

Training Confusion_Matrix
 [[16884     1]
 [    0 18866]] Testing Confusion_Matrix
 [[4309   17]
 [  52 4560]]
Classification Report
              precision    recall  f1-score   support

           0       0.99      1.00      0.99      4326
           1       1.00      0.99      0.99      4612

    accuracy                           0.99      8938
   macro avg       0.99      0.99      0.99      8938
weighted avg       0.99      0.99      0.99      8938

 * * * * *
```

7.

```
*******************************
 GradientBoostingClassifier()
 *******************************
Training Score 0.9968392492517691 And Precision Score 99.71671388101983
Accuracy Training Score = 0.9968392492517691  Accuracy Test Score = 0.9945177892145894

Training Confusion_Matrix
 [[16860    25]
 [   88 18778]] Testing Confusion_Matrix
 [[4313   13]
 [  36 4576]]
Classification Report
              precision    recall  f1-score   support

           0       0.99      1.00      0.99      4326
           1       1.00      0.99      0.99      4612

    accuracy                           0.99      8938
   macro avg       0.99      0.99      0.99      8938
weighted avg       0.99      0.99      0.99      8938

 * * * * *
```

8.

```
*******************************
 SGDClassifier()
 *******************************
Training Score 0.9905736902464267 And Precision Score 99.55956837700947
Accuracy Training Score = 0.9905736902464267  Accuracy Test Score = 0.9875811143432536

Training Confusion_Matrix
 [[16800    85]
 [  252 18614]] Testing Confusion_Matrix
 [[4306   20]
 [  91 4521]]
Classification Report
              precision    recall  f1-score   support

           0       0.98      1.00      0.99      4326
           1       1.00      0.98      0.99      4612

    accuracy                           0.99      8938
   macro avg       0.99      0.99      0.99      8938
weighted avg       0.99      0.99      0.99      8938

 * * * * *
```

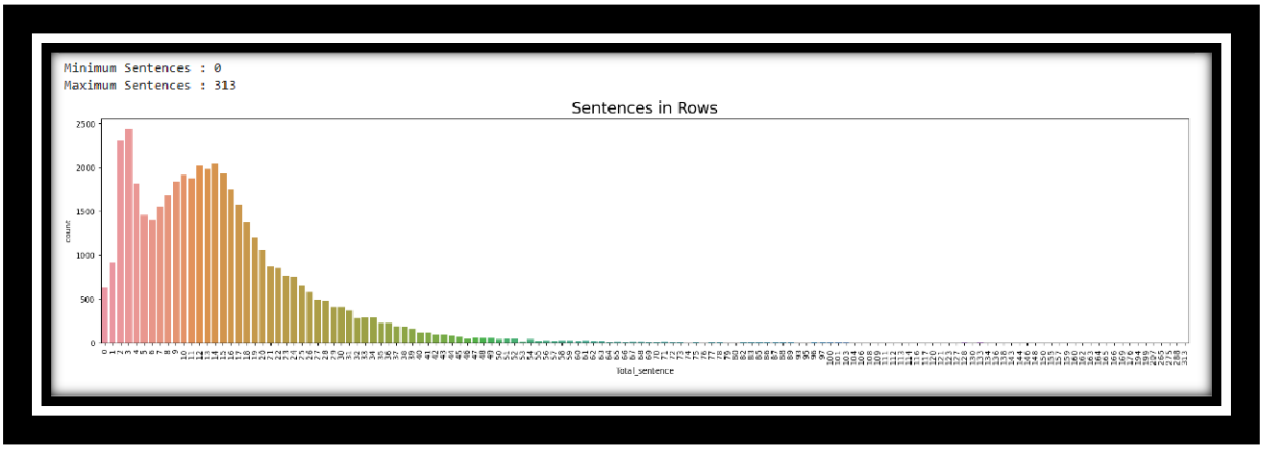
9.

```
*******************************
 MLPClassifier()
 *******************************
Training Score 0.9999720287544405 And Precision Score 98.91681109185441
Accuracy Training Score = 0.9999720287544405  Accuracy Test Score = 0.9892593421347058

Training Confusion_Matrix
 [[16884     1]
 [    0 18866]] Testing Confusion_Matrix
 [[4276   50]
 [  46 4566]]
Classification Report
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      4326
           1       0.99      0.99      0.99      4612

    accuracy                           0.99      8938
   macro avg       0.99      0.99      0.99      8938
weighted avg       0.99      0.99      0.99      8938

 * * * * *
```

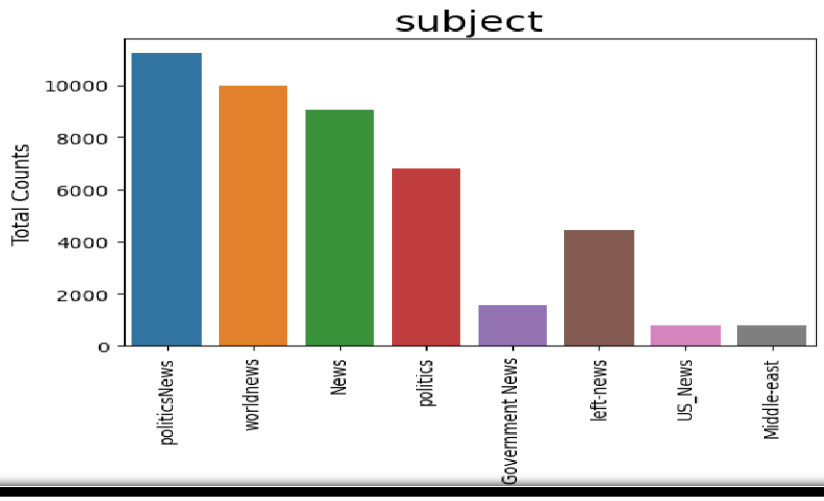
- Key Metrics for success in solving problem under consideration

  - **Precision Score**
  - **Model Training Score**
  - **Accuracy Score**
  - **Confusion Metrics**
  - **Classification Report**

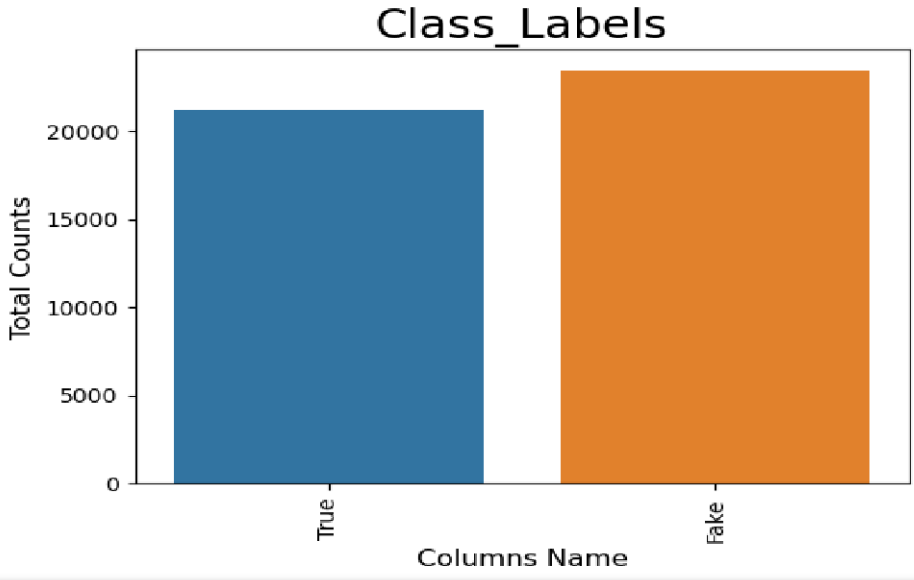
- Visualizations:

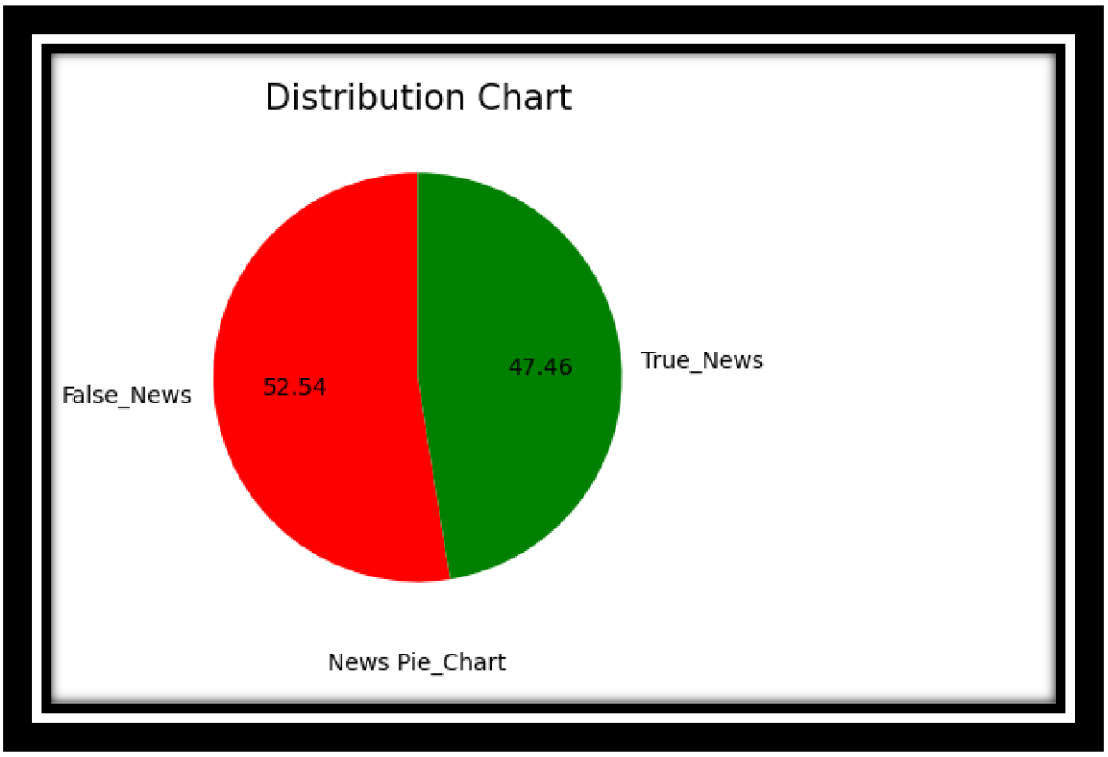


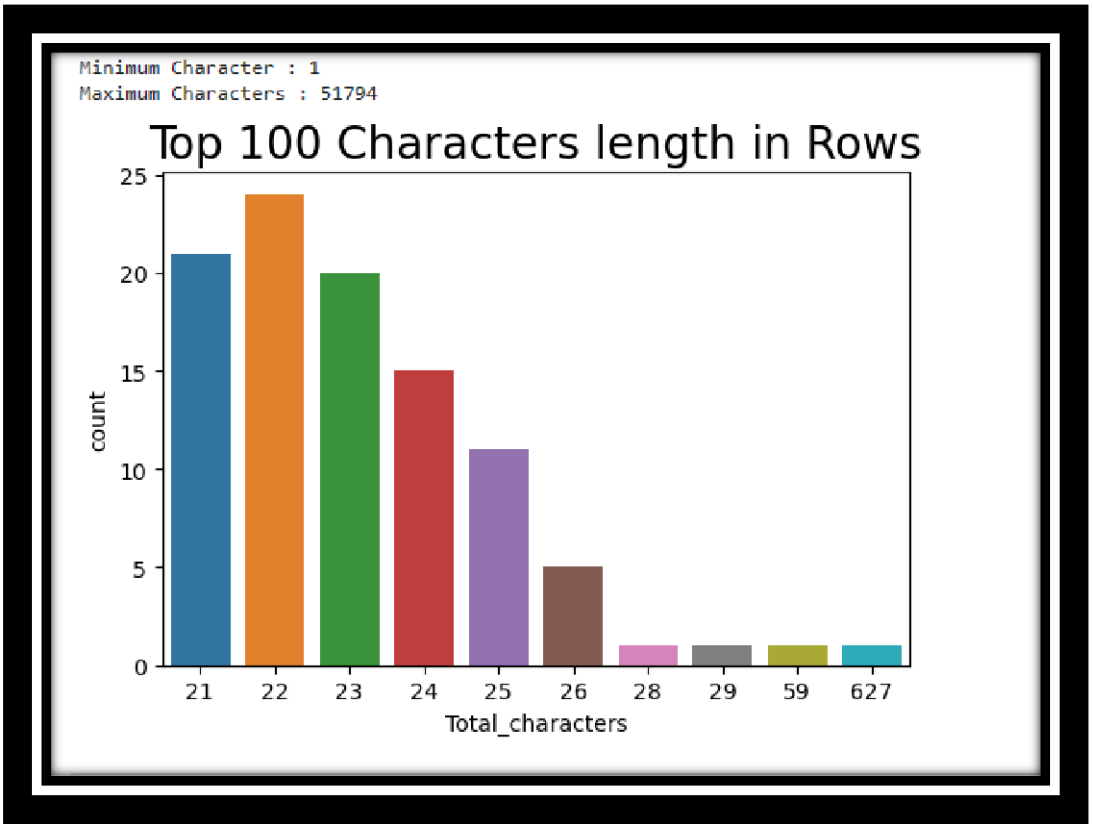
```
politicsNews    11220
worldnews        9991
News             9050
politics         6838
left-news        4459
Name: subject, dtype: int64
```

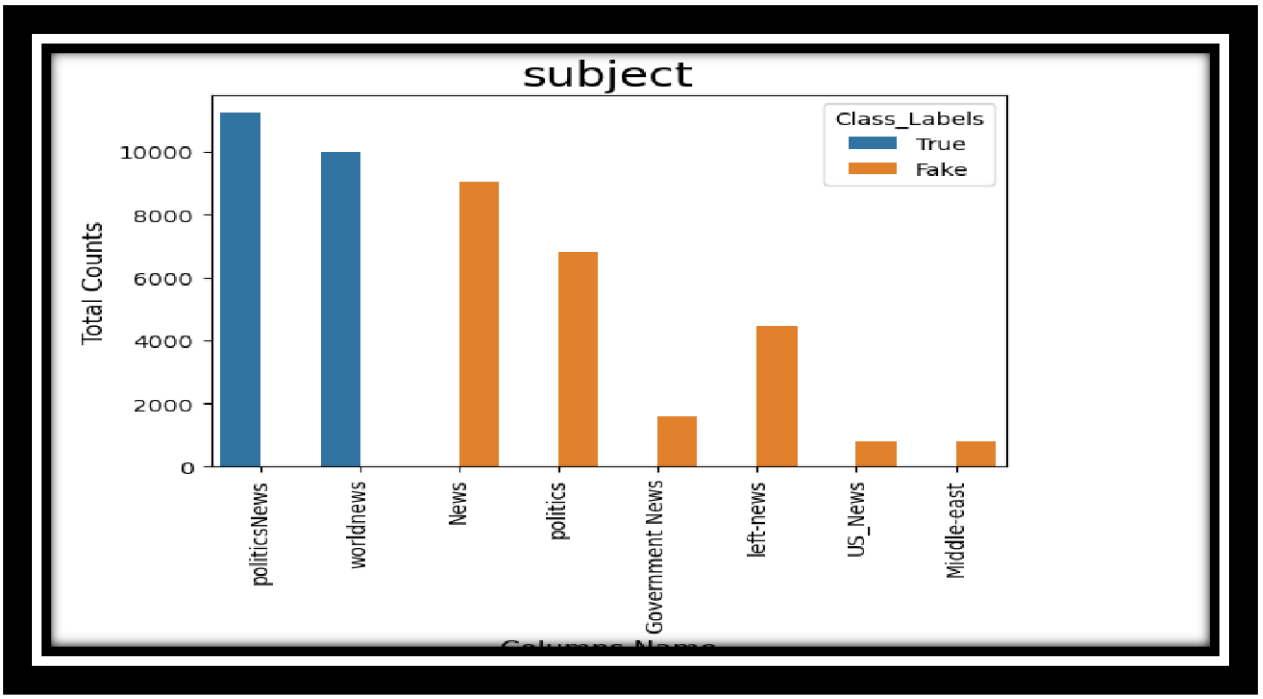


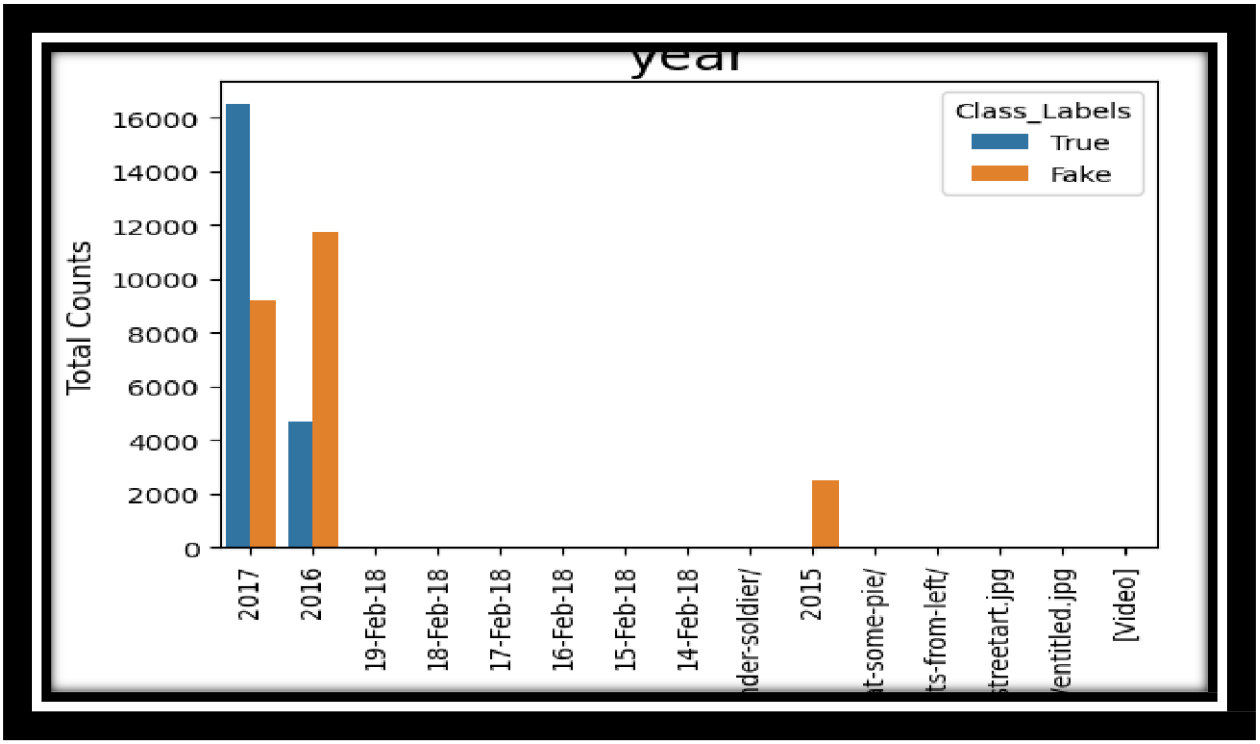

```
Fake    23478
True    21211
Name: Class_Labels, dtype: int64
```




Distribution Chart
False_News
52.54
47.46
True_News
News Pie_Chart

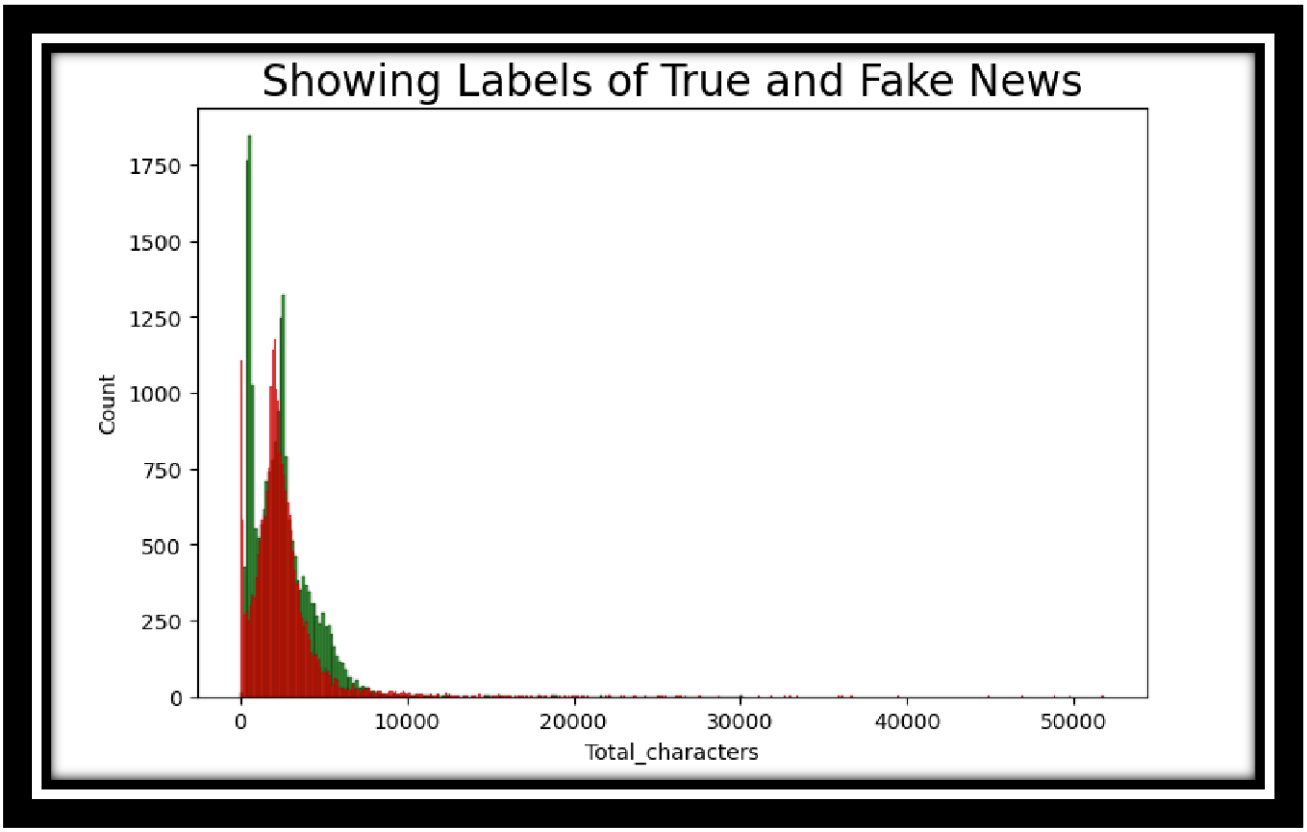

Minimum Character : 1
Maximum Characters : 51794
Top 100 Characters length in Rows
count
25
20
15
10
5
0
21
22
23
24
25
26
28
29
59
627
Total_characters


subject
Class_Labels
True
Fake
Total Counts
10000
8000
6000
4000
2000
0
politicsNews
worldnews
News
politics
Government News
left-news
US_News
Middle-east

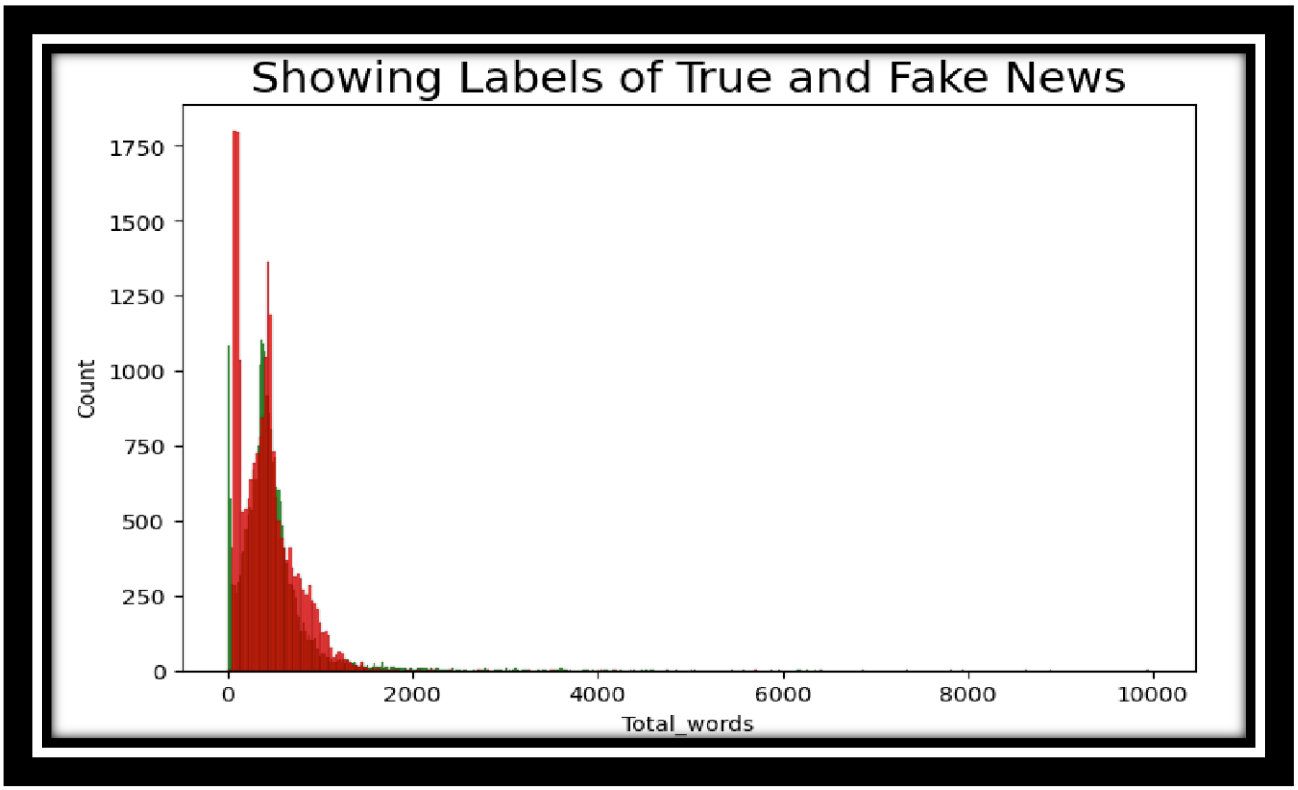

year
Class_Labels
True
Fake
Total Counts
16000
14000
12000
10000
8000
6000
4000
2000
0
2017
2016
19-Feb-18
18-Feb-18
17-Feb-18
16-Feb-18
15-Feb-18
14-Feb-18
2015
[Video]

Checking Fake News and True News

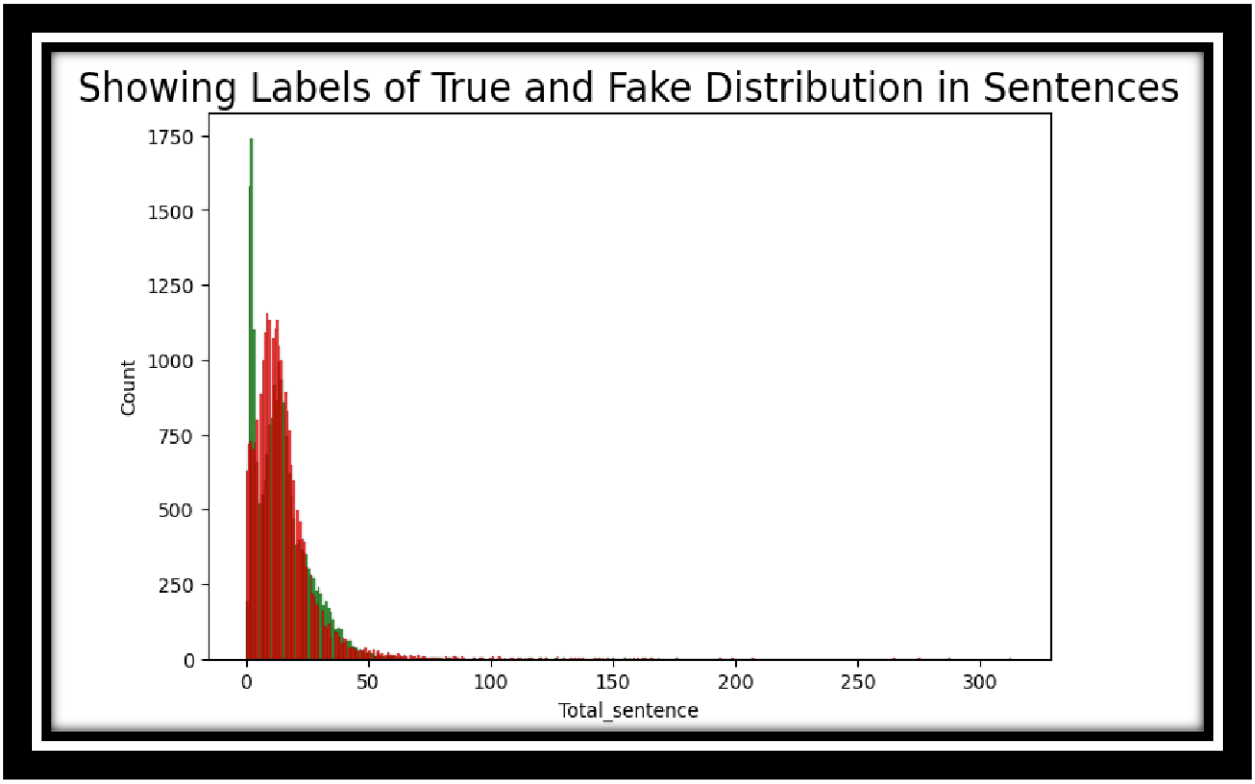
SPAM False News and True News Classification based on Total characters



SPAM False News and True News Classification based on Total Words



SPAM False News and True News Classification based on Total Sentences



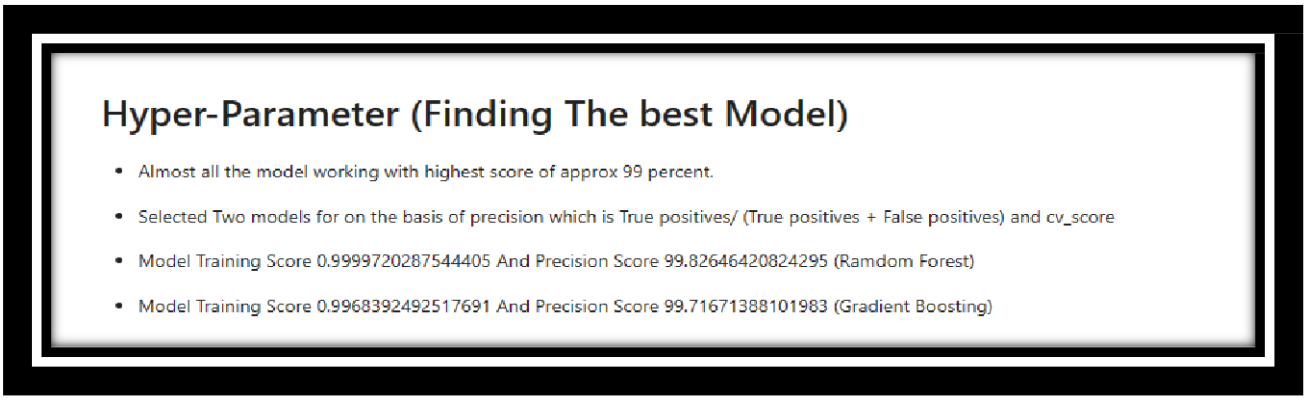
➢ Interpretation of the Results

- This dataset was very special as it had a separate dataset of True News and False News and result was drawn from object based independent variable.
- Firstly, the datasets were checked for Null values. Hence no chance of imputation. Those were dropped.
- The data set had 209 duplicated rows. Checked and dropped.
- I found maximum object-based columns and no integer columns.
- I used maximum Count Plot and Cat Plot followed by Pie chart, Pearsoncorrelation plot to find the relationship with target variable.
- I notice a huge number of outliers and high skewness in the data.
- We used lower function, string. Punction function, stemming for text pre-processing.
- We used feature extraction TfidVectorizer.
- We used Word Clouds to detect the frequent used words in Fake News and True News.

- We have to use multiple models Mainly based on Naïve Based Classifier,NLP classifiers and other Classifiers.
- Random Forest Classification and Linear Regression has best precision score, and accuracy score above 99 percent and close to 100 percentage.

- Finally selected Random Forest Classification Model and saved the model and finally printed the score and compared the data with the test data with the original data.

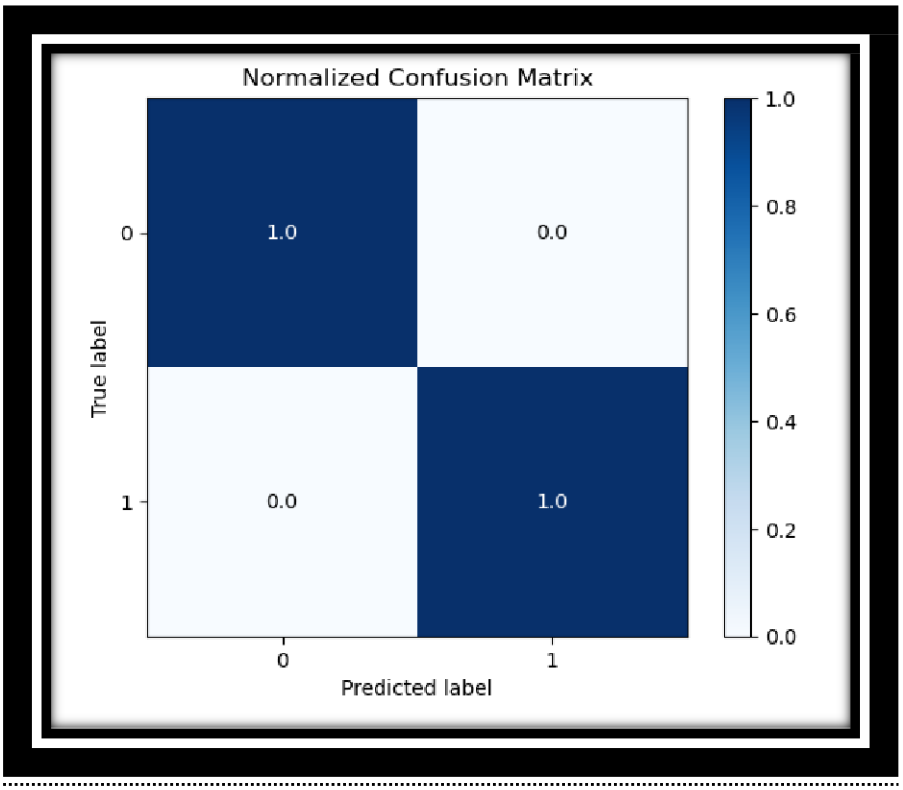
Hyper Parameter



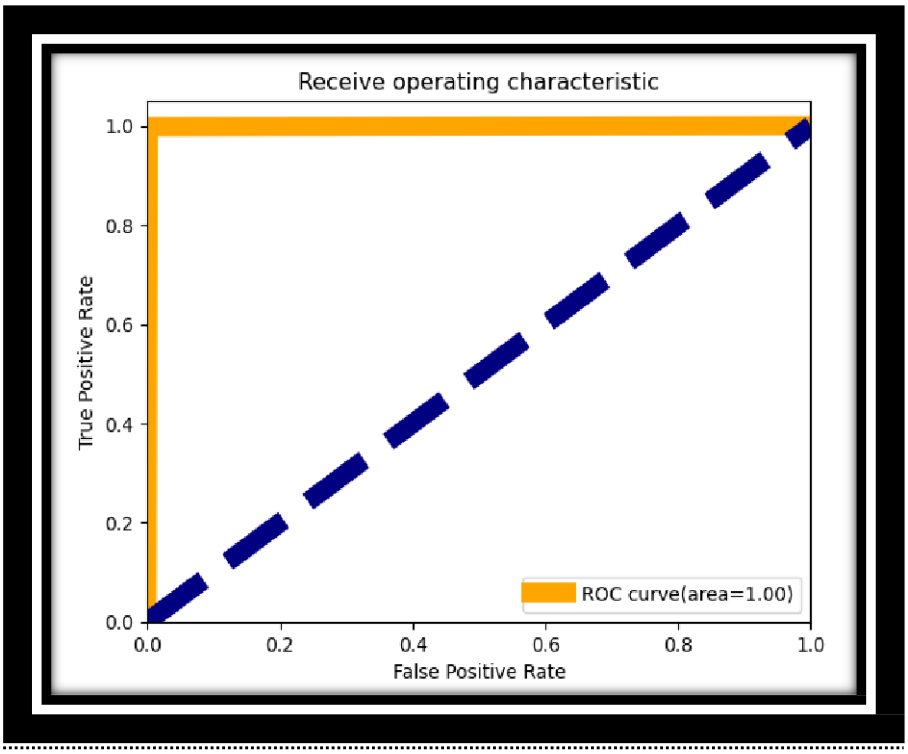
Using Random Forest Classifier and as trail method we will use Logistic Regression.

```
Training Score RandomForestClassifier(min_samples_split=4) And Precision Score 99.80494148244473
Accuracy Training Score = 0.9999720287544405  Accuracy Test Score = 0.9982098903557843

Training Confusion_Matrix
 [[16884     1]
 [    0 18866]] Testing Confusion_Matrix
 [[4317    9]
 [   7 4605]]
Classification Report
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      4326
           1       1.00      1.00      1.00      4612

    accuracy                           1.00      8938
   macro avg       1.00      1.00      1.00      8938
weighted avg       1.00      1.00      1.00      8938
```

Model Confusion Matrix



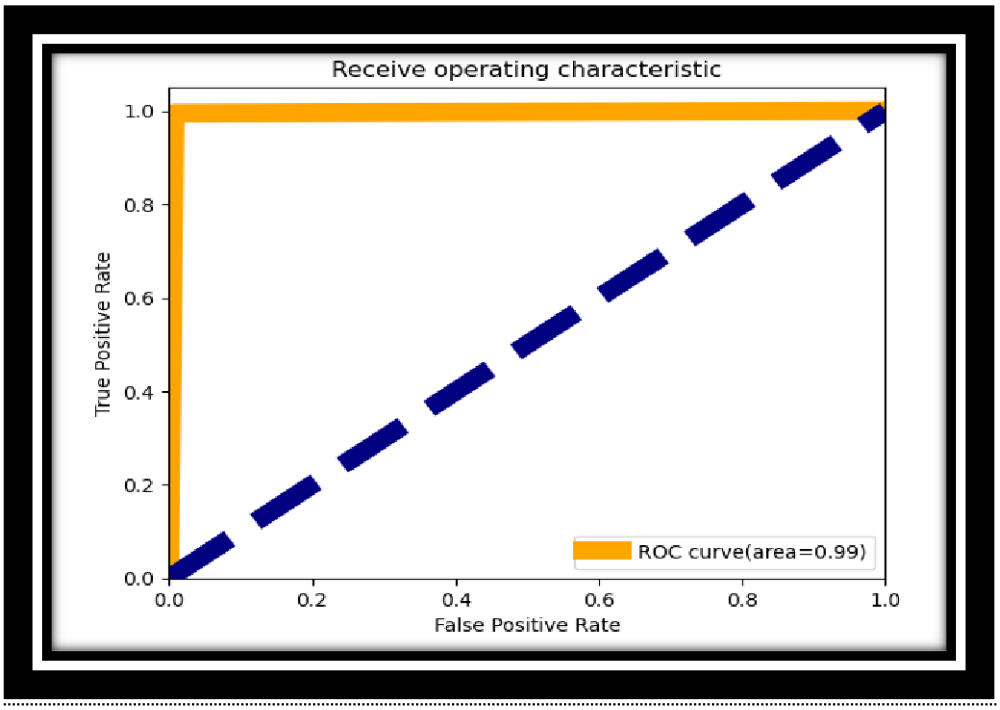
Graph ROC_AUC GRAPH



LOGISTIC CLASSIFIER:

```
Training Score LogisticRegression(penalty='l1', solver='liblinear') And Precision Score 99.54357748315583
Accuracy Training Score = 0.9945176358703253  Accuracy Test Score = 0.9940702618035354

Training Confusion_Matrix
 [[16823    62]
 [  134 18732]] Testing Confusion_Matrix
 [[4305   21]
 [  32 4580]]
Classification Report
              precision    recall  f1-score   support

           0       0.99      1.00      0.99      4326
           1       1.00      0.99      0.99      4612

    accuracy                           0.99      8938
   macro avg       0.99      0.99      0.99      8938
weighted avg       0.99      0.99      0.99      8938
```

AUC ROC GRAPH



There was an increment in logistic model but Random Forest Classifier works with a approx. 100 percent precision and accuracy. We will save the Random Forest Classification model for future needs.

SAVING MODEL

```
In [109…
# loading pickle

import pickle
filename='fakenews_classification.pkl'
pickle.dump(rfc,open(filename,"wb"))
```

Loading Model

```
In [110…
# loading pack file
pickled_model= pickle.load(open(filename,'rb'))
result=pickled_model.score(x_test,y_test)
print("Score Obtained",result*100)

Score Obtained 99.82098903557844
```

```
In [112…
array=np.array(y_test)

conclude=pd.DataFrame([pickled_model.predict(x_test)[:],y_test[:]],index=['Predicted','Original'])
conclude
```

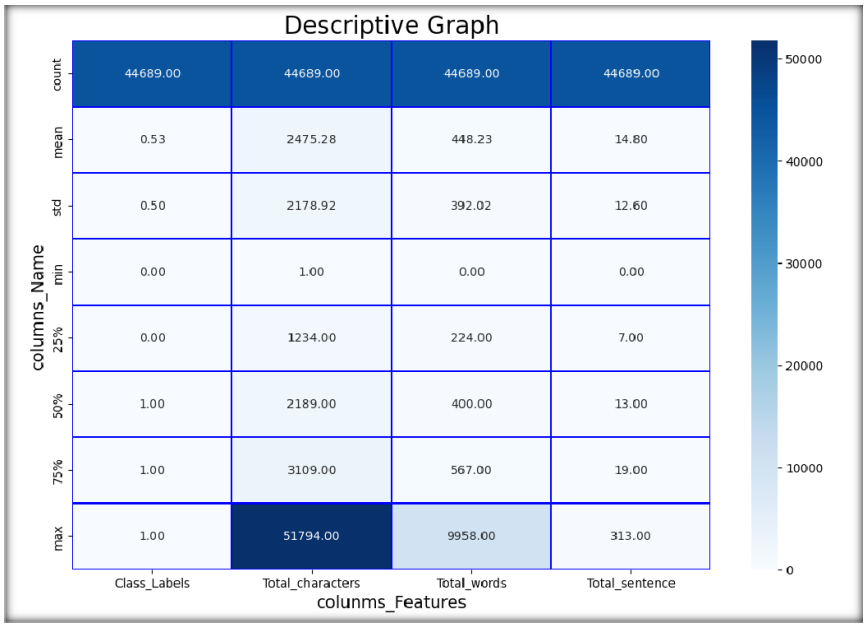
Out[112…

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41
Predicted	1	0	1	1	0	1	0	1	1	1	1	0	1	0	1	0	0	1	0	0	0	1	1	1	1	1	0	0	1	0	0	0	1	0	0	0	0	1	1	0	0	1
Original	1	0	1	1	0	1	0	1	1	1	1	0	1	0	1	0	0	1	0	0	0	1	1	1	1	1	0	0	1	0	0	0	1	0	0	0	0	1	1	0	0	1

In [ ]:

7. OTHERS

Descriptive Graph:



➢ Key Findings and Conclusions of the Study

By, the help of Text columns in Data set I have made a FAKE NEWS CLASSIFICATION Model. We have done EDA, data cleansing and Visualized Data on the independent variable and dependent variable of Data Set. In this data set we found few rows as duplicated rows. Those were dropped after analysis. While cleaning the data Set, we analysed that few columns are not useful, those were dropped after EDA, Null Values were checked and we found no null values. So I dropped that column. After that we have done prediction on basis of Data Pre-processing, Checked Descriptive Analysis along with that I Checked Correlation, removed I removed the Punctations, stop words, extra space, leaning and trailing white space, converted text into vectors using count Vectorizer.

Finally performed Train and Test, on the extracted variable in that Data Set. We worked with various classifiers while using these given models and finally selected best model, On a trial method we used Logistic Regression, For Model Selection we used the model with best accuracy, and Precession Score followed by F1 and CV score.

Logistic Regression and Random Forest Classification model were selected and sent for Hyper Tunning was used with Grid Search CV, And Best was Saved on the basis of Accuracy score and Precision Score.

Finally, I used that model to compare with predicted and Actual test data.

Thus, our project Stands completed as filename:'fakenews_classification.pkl'

➢ Learning Outcomes of the Study in respect of Data Science

I found that the dataset was quite interesting to handle as it contains all objective type types of data in it. Improvement in computing technology has made it possible to examine social information that cannot previously be captured, processed and analysed. New analytical techniques of machine learning can be used in FAKE NEWS Classification.

The power of visualization has helped us in understanding the data by graphical representation it has made me to understand what data is trying to say. Data cleaning is one of the most important steps to remove missing value and to replace null value and zero values with their respective mean, median or mode.

This study is an EXPLORATORY DATA ANALYSIS has attempted to use machine learning algorithms in finding the probability of FAKE NEWS AND TRUE NEWS in each model. To conclude the application of Machine Learning in prediction is still at an early stage. I hope this study has moved a small step ahead in providing solution to the companies.

The Challenges I faced was when I was Thinking to take title column as well as text column both for x test, its was showing lab to collapse, when I faced the same issue with google Collab. I tried pulling both the columns for NLP and I took the help for Md Kashif Sir and used various sources like Kaggle and Medium and NLTK library where I faced problem. Even The Algorithm was working huge time to complete the test and CV_Score generating was again time taking in each of the cases.

Finally, I had to run the test twice and thrice with different standardization process and models. I was actually thinking to get best out of those and come out of something that's very new.

However, I finally achieved a good model out of this. WITH PRECISION of approx. 100 % and Model Accuracy of 98 Percentage.

Limitations of this work and Scope for Future Work

This model doesn't predict Future probability. As the future will be always be unpredictable at all times due to this, the risk in FAKE NEWS AND TRUE NEWS classification remains an import factor. My Model can predict for a time period and needs to be updated as per need or on Regular basis to work perfectly. Machine can classify the NEWS but the intensity of the user remains unpredictable and Language used in typing can be another factor. So, the best way to be future ready to get the model updated once or twice as per the market standard and Requirements.