FLIP ROBO

FAKE NEWS ASSIGNMENT

Submitted By:-

Rishabh Johri

Data Science-Intern

ACKNOWLEDGEMENT:-

It is a great pleasure to express my gratitude to Team Flip Robo, for giving me the opportunity to work on a interesting project, which helped me in improving my knowledge, coding skills and my analysation skills.

Team Flip Robo also gave me opportunity to build PowerPoint Presentation and Project Report, which will help me to share steps taken while building the entire model. It has helped me in deciding about the future prospects of various Data Science fields. Now, I will explain the understanding of the project through this report.

INTRODUCTION

- **Business Problem Framing :-**

  Fake News Filtering

  Fake news has become one of the biggest problems of our age. It has a 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.
  Fake news simple meaning is to incorporate information that leads people to the wrong path. Nowadays, fake news is 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 the fake news.

- Conceptual Background of the Domain Problem

  The main goal of the assignment is to show how you could design a Fake news filtering system from basics.

  In this project, we are using some machine learning and Natural language processing libraries like NLTK, re (Regular Expression) and Scikit Learn.

-Natural Language Processing :-

Machine learning data only works with numerical features so we have to convert text data into numerical columns. Here, we have to Preprocess the text and that is called natural language processing. In-Text Preprocessing, we are cleaning our text by stemming, lemmatization, removing stopwords, removing special symbols and numbers, etc. After cleaning the data, we have to feed this text data into a vectorizer which will convert this text data into numerical features.

● Review of Literature :-

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. I have inserted one label column 0 for Fake news and one for true news:

- Title: Headlines of the news.

- Text: Content of the news.

- Subject: Subject of the news.

- Date: Date of the news.

- Label: News is True(1)/False(0)

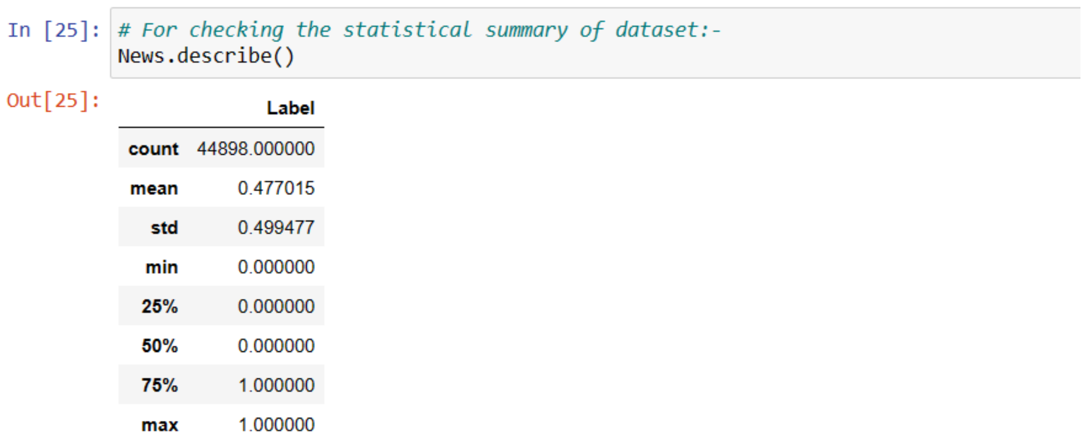
Motivation for the Problem Undertaken :-

The authenticity of Information has become a major 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 sensation of not-so-accurately 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.

Analytical Problem Framing :-

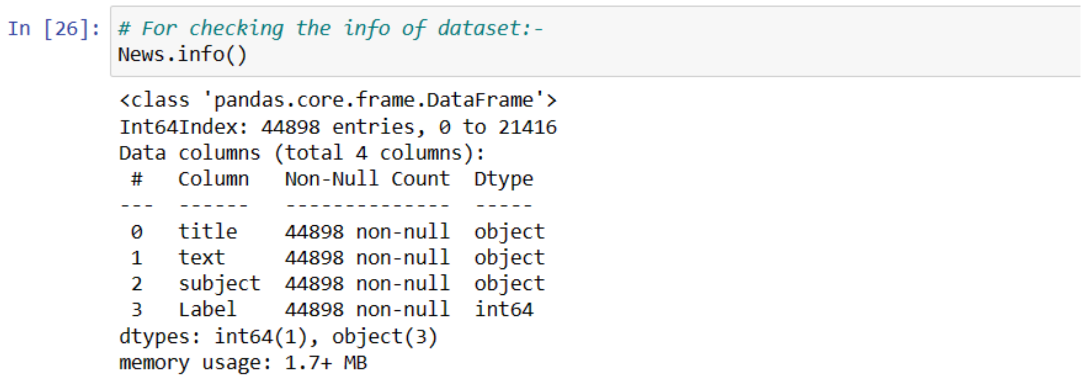
- Mathematical/ Analytical Modelling of the Problem :-
- For fetching the information and description of the dataset :-



```
In [25]: # For checking the statistical summary of dataset:-
         News.describe()
```

Out[25]:

	Label
count	44898.000000
mean	0.477015
std	0.499477
min	0.000000
25%	0.000000
50%	0.000000
75%	1.000000
max	1.000000



```
In [26]: # For checking the info of dataset:-
         News.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 44898 entries, 0 to 21416
Data columns (total 4 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   Label    44898 non-null  int64 
dtypes: int64(1), object(3)
memory usage: 1.7+ MB
```

● 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.

● Data Preprocessing :-

In data pre-processing, I have done various steps to clean the dataset, as the dataset contains the comment that are in object datatype, which cannot be read by the model, so before giving the features to the model I had to convert that object datatype to meaningful data and that can be understand by the model, so for this I have used the NLP (Natural Processing Language).

“Natural language processing (NLP) refers to the branch of computer science and more specifically, the branch of artificial intelligence (AI) concerned with giving computers the ability to understand text and spoken words in the same way in which human beings can.”

● Data Inputs- Logic- Output Relationships :-

Here I have Used TF-IDF Vectorizer to encode the comments section.

TfidfVectorizer is the base building block of many NLP pipelines. It is a simple technique to vectorize text documents i.e. transform sentences into arrays of numbers and use them in subsequent tasks.

● Some Hardware and Software Requirements and Tools Used :

Anaconda-navigator

jupyter notebook

matplotlib-inline =0.1.6

numpy =1.23.2

Model/s Development and Evaluation :-

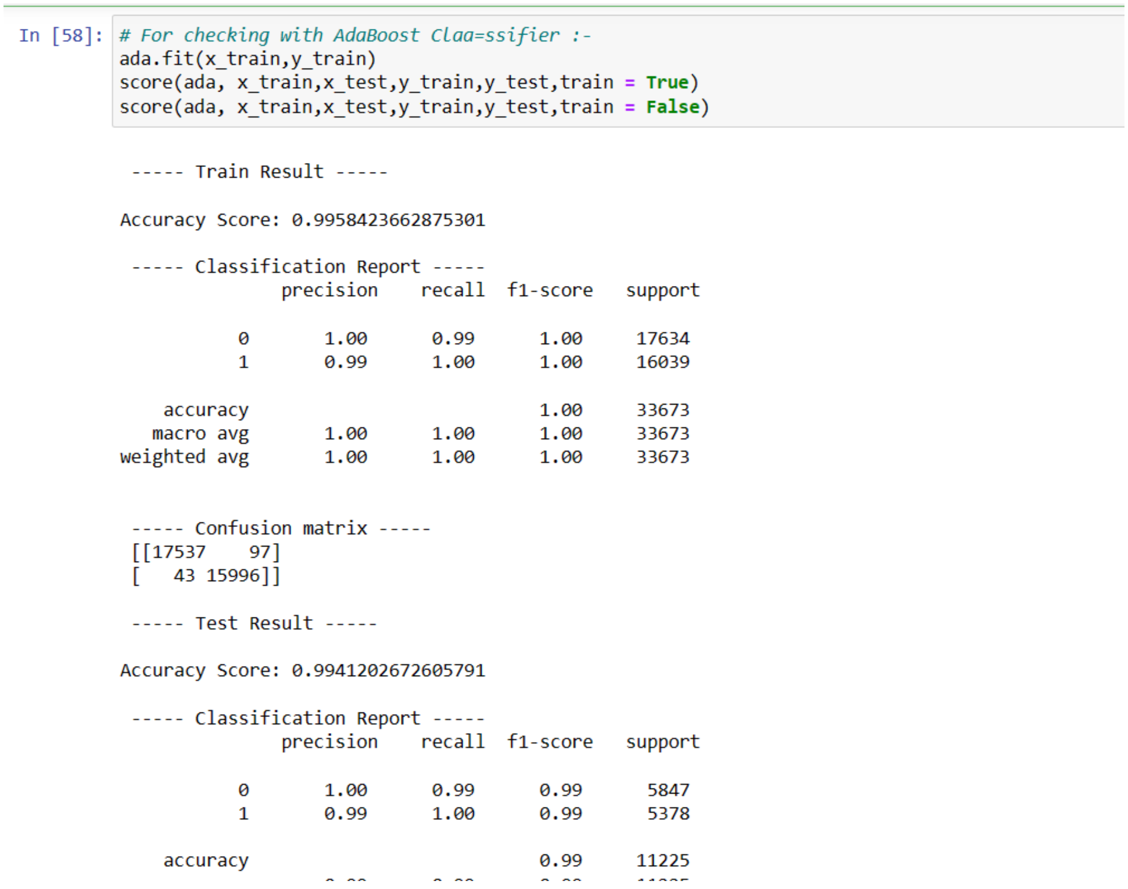
- Identification of possible problem-solving approaches (methods) :-
    - EDA
    - Description
    - Visualization
    - Data cleaning
    - Data Pre-processing (NLP)
    - Word Cloud
    - Encoding
    - Model Building
    - Select the best model
    - Cross-Validation.

- **Testing of Identified Approaches (Algorithms) :-**

  Algorithms used for the training and testing:
    - AdaBoost Classifier
    - GradientBoosting Classifier
    - RandomForest Classifier.
    - Decision Tree Classifier.

Model Building and Performance :-

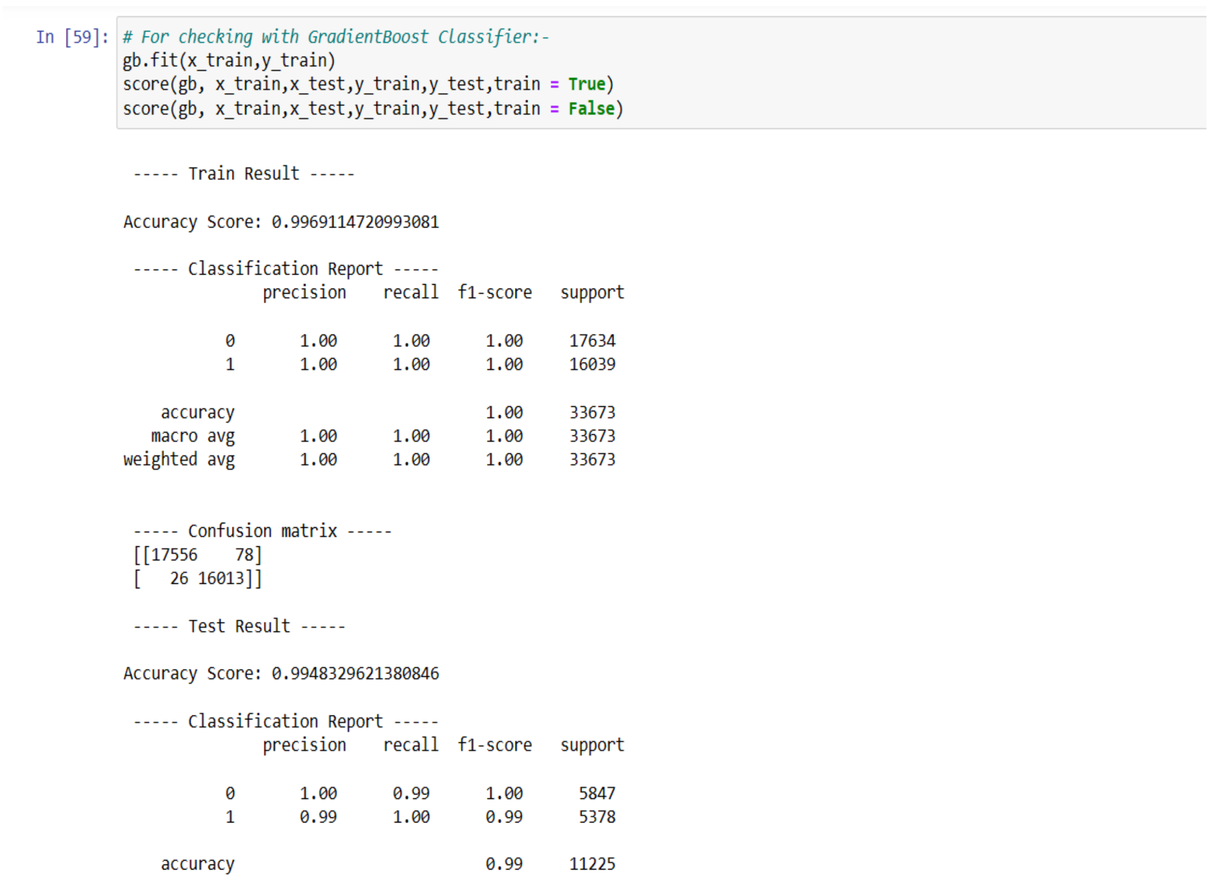
AdaBoost Classifier :-



```
In [58]: # For checking with AdaBoost Claa=ssifier :-
         ada.fit(x_train,y_train)
         score(ada, x_train,x_test,y_train,y_test,train = True)
         score(ada, x_train,x_test,y_train,y_test,train = False)
```

```
 ----- Train Result -----

Accuracy Score: 0.9958423662875301

 ----- Classification Report -----
              precision    recall  f1-score   support

           0       1.00      0.99      1.00     17634
           1       0.99      1.00      1.00     16039

    accuracy                           1.00     33673
   macro avg       1.00      1.00      1.00     33673
weighted avg       1.00      1.00      1.00     33673


 ----- Confusion matrix -----
 [[17537    97]
 [   43 15996]]

 ----- Test Result -----

Accuracy Score: 0.9941202672605791

 ----- Classification Report -----
              precision    recall  f1-score   support

           0       1.00      0.99      0.99      5847
           1       0.99      1.00      0.99      5378

    accuracy                           0.99     11225
```

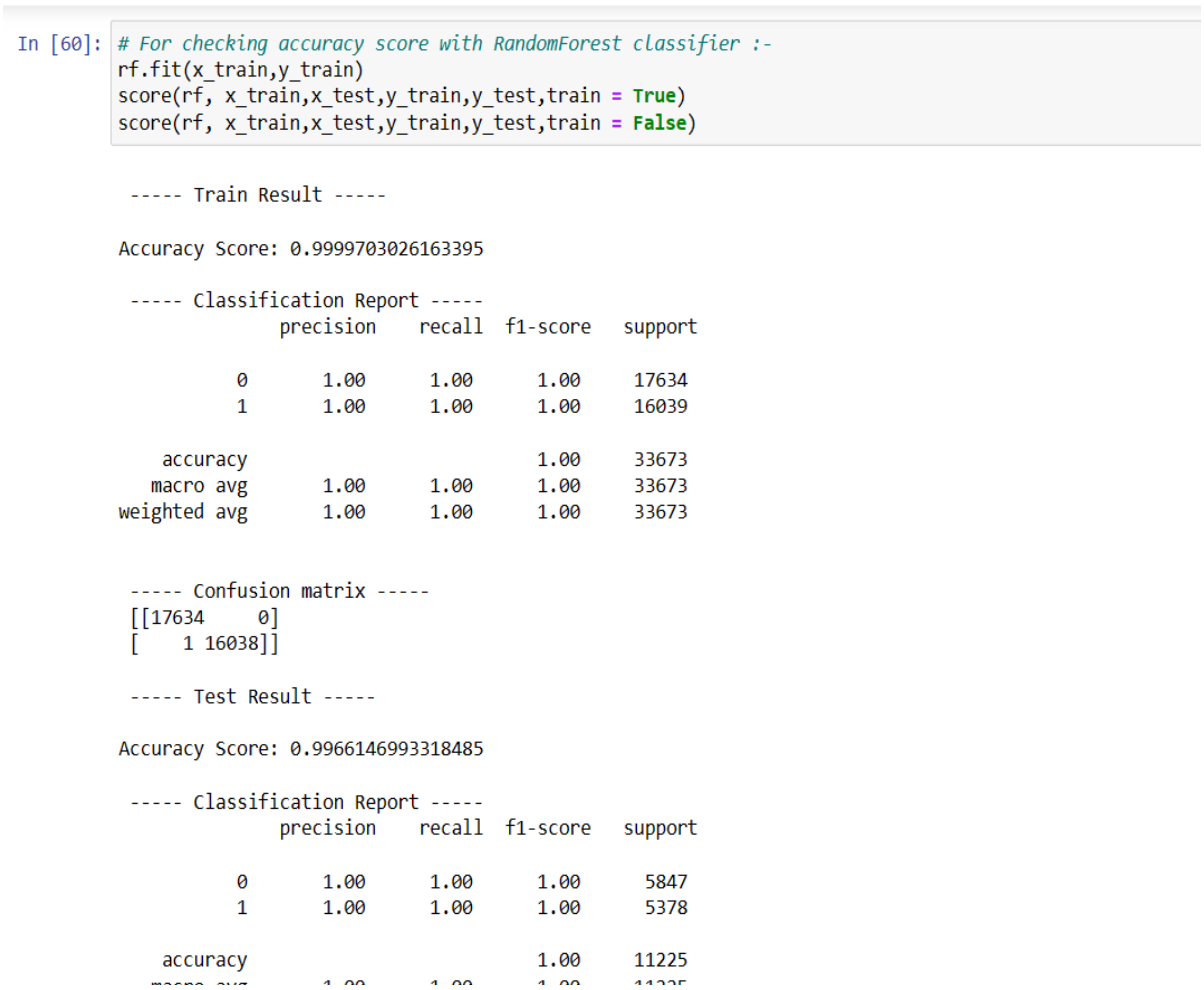
GradientBoosting Classifier :-



```
In [59]: # For checking with GradientBoost Classifier:-
         gb.fit(x_train,y_train)
         score(gb, x_train,x_test,y_train,y_test,train = True)
         score(gb, x_train,x_test,y_train,y_test,train = False)
```

```
 ----- Train Result -----

Accuracy Score: 0.9969114720993081

 ----- Classification Report -----
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     17634
           1       1.00      1.00      1.00     16039

    accuracy                           1.00     33673
   macro avg       1.00      1.00      1.00     33673
weighted avg       1.00      1.00      1.00     33673


 ----- Confusion matrix -----
 [[17556    78]
 [   26 16013]]

 ----- Test Result -----

Accuracy Score: 0.9948329621380846

 ----- Classification Report -----
              precision    recall  f1-score   support

           0       1.00      0.99      1.00      5847
           1       0.99      1.00      0.99      5378

    accuracy                           0.99     11225
```

RandomForest Classifier :-



```
In [60]: # For checking accuracy score with RandomForest classifier :-
         rf.fit(x_train,y_train)
         score(rf, x_train,x_test,y_train,y_test,train = True)
         score(rf, x_train,x_test,y_train,y_test,train = False)
```

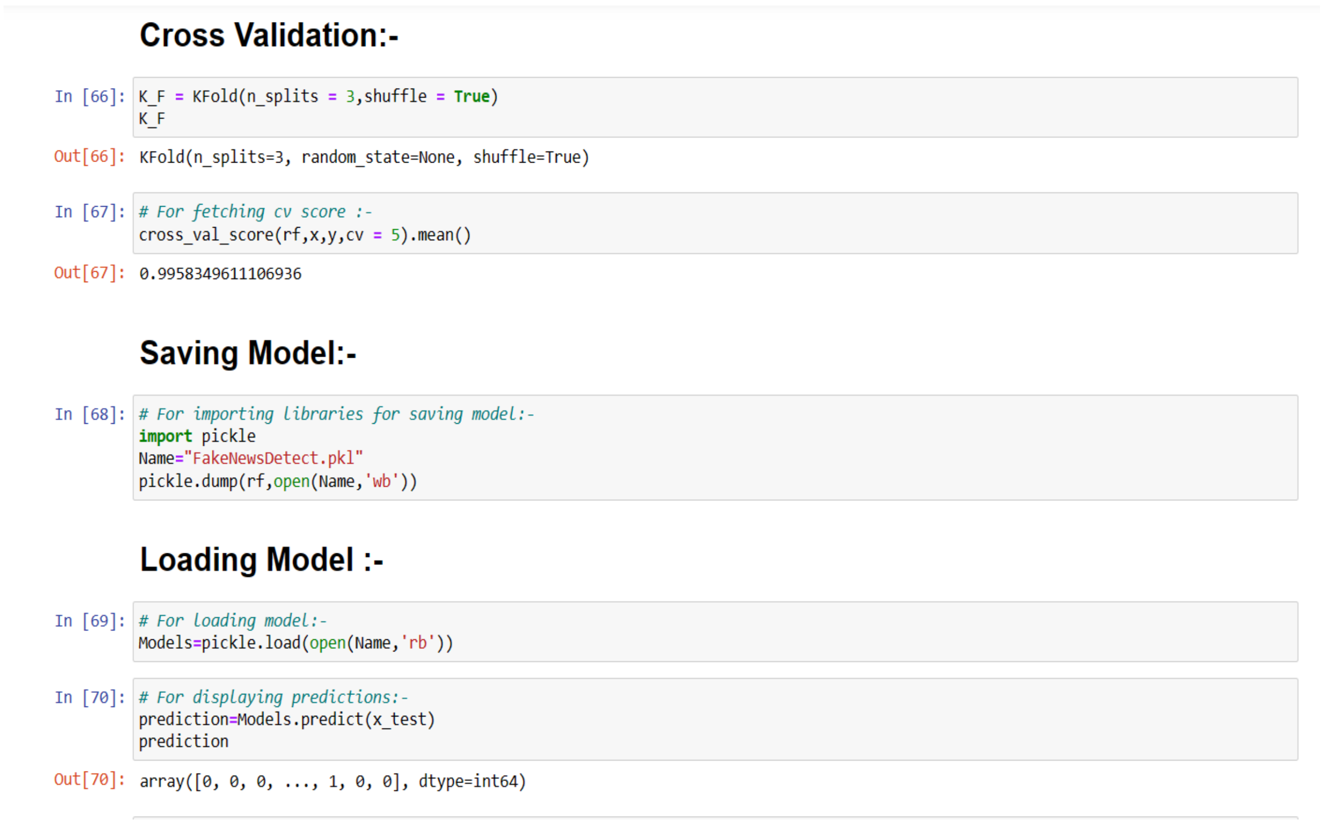
```
 ----- Train Result -----

Accuracy Score: 0.9999703026163395

 ----- Classification Report -----
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     17634
           1       1.00      1.00      1.00     16039

    accuracy                           1.00     33673
   macro avg       1.00      1.00      1.00     33673
weighted avg       1.00      1.00      1.00     33673


 ----- Confusion matrix -----
 [[17634     0]
 [    1 16038]]

 ----- Test Result -----

Accuracy Score: 0.9966146993318485

 ----- Classification Report -----
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      5847
           1       1.00      1.00      1.00      5378

    accuracy                           1.00     11225
```

DecisionTree Classifier :-

```
In [62]: # For checking accuracy score with DecisionTree classifier :-
         dt.fit(x_train,y_train)
         score(dt, x_train,x_test,y_train,y_test,train = True)
         score(dt, x_train,x_test,y_train,y_test,train = False)
```

```
 ----- Train Result -----

Accuracy Score: 0.9999703026163395

 ----- Classification Report -----
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     17634
           1       1.00      1.00      1.00     16039

    accuracy                           1.00     33673
   macro avg       1.00      1.00      1.00     33673
weighted avg       1.00      1.00      1.00     33673


 ----- Confusion matrix -----
 [[17634     0]
 [    1 16038]]

 ----- Test Result -----

Accuracy Score: 0.9951893095768374

 ----- Classification Report -----
              precision    recall  f1-score   support

           0       0.99      1.00      1.00      5847
           1       1.00      0.99      0.99      5378

    accuracy                           1.00     11225
   macro avg       1.00      1.00      1.00     11225
```

Cross Validation along with Saving and Loading Model:-



Cross Validation:-

```
In [66]: K_F = KFold(n_splits = 3,shuffle = True)
         K_F

Out[66]: KFold(n_splits=3, random_state=None, shuffle=True)

In [67]: # For fetching cv score :-
         cross_val_score(rf,x,y,cv = 5).mean()

Out[67]: 0.9958349611106936
```

Saving Model:-

```
In [68]: # For importing libraries for saving model:-
         import pickle
         Name="FakeNewsDetect.pkl"
         pickle.dump(rf,open(Name,'wb'))
```

Loading Model :-

```
In [69]: # For loading model:-
         Models=pickle.load(open(Name,'rb'))

In [70]: # For displaying predictions:-
         prediction=Models.predict(x_test)
         prediction

Out[70]: array([0, 0, 0, ..., 1, 0, 0], dtype=int64)
```

- **Interpretation of the Results :-**

  RandomForest Classifier is giving the best result as compared to others.

CONCLUSION :-

Apply computing theory, languages, and algorithms, as well as mathematical and statistical models, and the principles of optimization to appropriately formulate and use data analyses. Formulate and use appropriate models of data analysis to solve hidden solutions for business-related challenges And Perform well in a group.