Table of Contents

1. Introduction **2**

1.1 How fake news works........ 2

1.2 Steps being are taken 2

1.3 Types of detection techniques 3

1.4 Approach using machine learning........ 3

1.5 Deep learning approach 4

2. Objective **5**

3. Methodology........ **5**

3.1 Implementation 6

3.1.1 Data Collection & Feature Analysis Data Collection 6

3.1.2 Importing the Libraries 7

3.1.3 Data pre-processing 7

3.1.4 Algorithms Used in implementation 10

a. Decision tree 10

b. Random Forest 15

c. Logistic Regression 17

4. Result Analysis **19**

5. Conclusion........ **20**

6. References **20**

1. Introduction

A few decades ago, the term "Fake News" was less obscure and less common, but in the digital age of social media, it has emerged as a big monster. Growing issues in our society include fake news, knowledge bubbles, news manipulation, and a loss of faith in the media. However, a thorough grasp of fake news and its sources is necessary in order to begin tackling this issue. Only then can one consider the various methods and We might be able to combat this predicament with the use of machine learning (ML), natural language processing (NLP), and artificial intelligence (AI) sectors. Over the past six months, there have been many different uses and definitions of the term "fake news."

It may happen so rapidly that measuring fake news or even accurately describing it would no longer `be an objective statistic. In its purest form, fake news is entirely untrue and has been edited to look like reliable reporting in order to gain as much attention and, consequently, advertising income as possible. Despite all of these flaws, some organizations have made an effort to classify fake news in various ways.

1.1 How fake news works

Platforms for social media have enormous influence. A 500 million tweets, emails, Facebook news etc. per day estimate is provided by internet live environment. These systems are widely used. They are the preferred setting for exchanging ideas, emotions, viewpoints, and goals. This offers the best circumstances for disseminating news with the fewest limits and limitations.

Today, getting news from online sources like social media is commonplace. For readers, news is frequently arbitrary. We frequently decide to consume material that speaks to the various emotions we experience. In light of this, the news that receives the most exposure could not be reliable or truthful. Real news may also be misrepresented during transmission. The same news may appear in various forms for a reader.

1.2 Actions are being taken to stop bogus news

WHO is collaborating with businesses like Facebook, Twitter, Google, Pinterest, Tencent, YouTube, and others to stop the spread of falsehoods. They work to remove information that could be harmful to the general public's health. There are ways to help in this battle.

1.3 Types of News Detection

There are two methods used in the fake news detection -

1.3.1 Manual Fake News Detection

All the methods and approaches one might employ to verify the news are frequently used in manual false news identification. It might entail going to websites for fact-checking. Real news could be crowdsourced and compared to unreliable news. However, the volume of data generated online every day is staggering. Furthermore, given how quickly information circulates online, manual fact-checking is quickly rendered useless. The amount of data produced makes manual fact checking difficult to scale. underlining the motivation for the development of automatic fake news detection.

1.3.2 Automatic Fake News Detection

The advantage of automated detection systems is in its automation and scalability. Research on false news identification employs a variety of methods and methodologies. It is also important to note that, depending on the viewpoint, these techniques frequently overlap. From a personal standpoint, I'll simply talk about two strategies.

These two approaches focus on the methods used, as opposed to the content being analyzed. They may also both involve Natural Language Processing (NLP) in their methodology.

The two approaches to fake news detection are:

a. **Machine Learning approach**

b. **Deep Learning approach**

1.4 Approach using Machine Learning

Giving computers the capacity to learn without being explicitly programmed is referred to as machine learning. To identify false information, a machine learning strategy use machine learning algorithm. These algorithms include, for instance:

a. **Naive Bayes:** uses probabilistic approaches based on Bayes theorem. This algorithm is often used for text classification.

b. **Decision Tree:** a supervised learning algorithm that has a tree-like flow. It helps in decision making. A useful algorithm for both classification and regression tasks.

c. **Random forest:** simply a combination of decision trees.

d. **Support Vector Machine:** a supervised learning algorithm. It examines data for classification and regression analysis. It classifies data into two categories.

e. **Logistic Regression:** contrary to the name, it is a classification algorithm used to estimate discrete values.

f. **K-nearest-neighbor:** a simple algorithm that is used for both classification and regression tasks. Though it is more widely used for classification problems.

1.5 Deep Learning Approach

Similar to machine learning algorithms, deep learning algorithms work. But there is a crucial distinction. Different layers in deep learning algorithms perceive data in different ways. The network comprising these algorithms is referred to as artificial neural networks. There have been numerous investigations into pure deep learning perspectives on fake news detection. At the end of the essay, I've included links to some of these published pieces.

Building classifiers to determine the veracity of news based solely on news content may be a methodology. Recurrent neural network (RNN) models and long-short term memories can be used to do this (LSTM). Click on this article, which describes text generation with RNN + TensorFlow, for more detail on RNN.

An RNN is a neural network with loops that enable the network to store information. Future events in RNNs are influenced by past experiences. LSTM is responsible for information storage. Artificial recurrent neural networks that provide information persistence are referred to as LSTM. They serve as the foundation for RNN layers. The capacity to "recall" values over time is offered by LSTM units. This affects how words and their occurrences are related.

It is possible to use both machine learning and deep learning methods together. The two have been combined in numerous published works. In addition to identifying fake news, the goal is to do it with the greatest degree of accuracy.

2. Objective

This study aims to create and implement a machine learning system that accurately predicts if a particular piece of content would be regarded as fake news. The following are the contributions of this paper:

I. Describes false news and the various machine learning methods that can be used to create a model that can correctly identify whether a piece of news is real or fake.

II. Gives a general overview of the origins and effects of fake news.

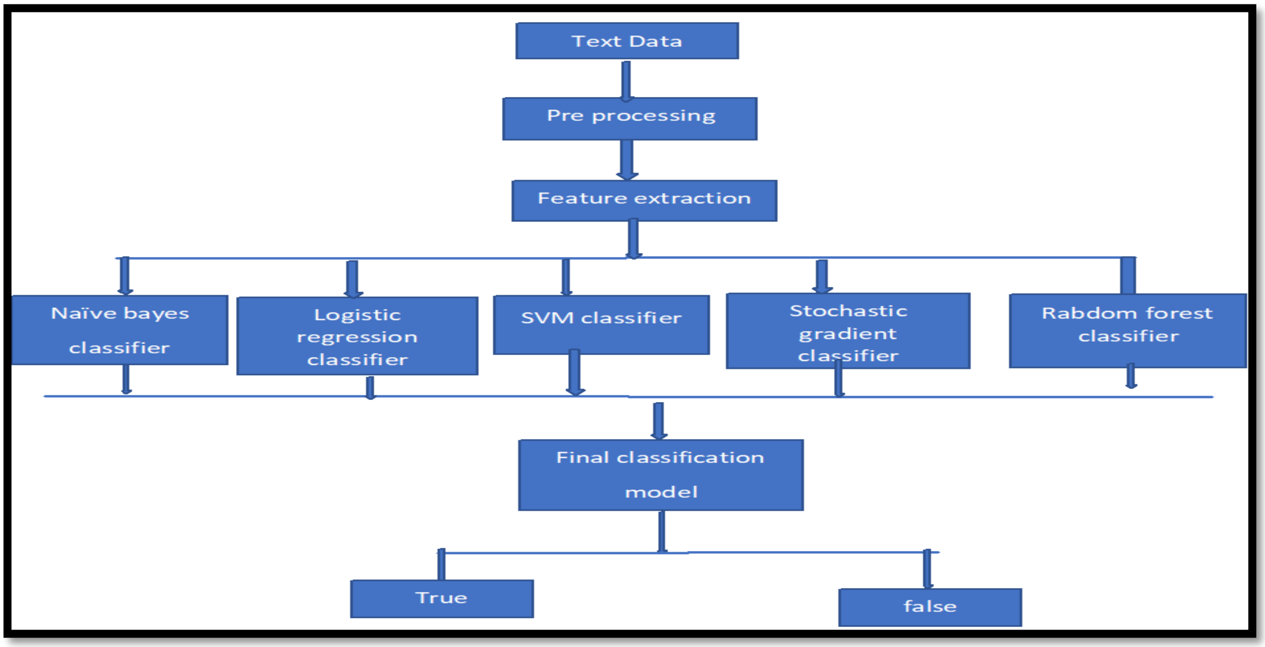
III. Both fake and legitimate news are covered by this cutting-edge Python framework for detecting bogus news. We will apply a Tfid Vectorizer to our dataset using SK-Learn tools.

IV. Following that, we fit the model with a Passive-Aggressive Classifier (PAC), which generates an accuracy score and a confusion matrix that indicate how well our model performed.

V. Outlines a potential resolution and lays the foundation for more research in this field.

3. Methodology

In this project, our aim is to propose a suitable solution that can result in a high-efficiency rate than the previously mentioned research works and implementations in methodology



‘ **Fig-1**

A classifier's performance may differ depending on the quantity and quality of the text data (or corpus), as well as the characteristics of the text vectors. Common noisy words, or "stopwords," are less significant when it comes to text feature extraction because they just add to the dimensionality of the feature set rather than the sentence's real meaning. They can be ignored for better performance.

This adds text context for feature extraction and reduces the size/dimensionality of the text corpus. Lemmatization is another technique used to reduce words to their essence, which results in the reduction of several words into a single discrete representation.

After this, a textbook exploratory data analysis for understanding the dataset and cleaning it has been conducted. Feature extraction and assortment of selection methods has been facilitated by the python’s “scikitlearn” library. For feature selection, the use of methods like simple bag-of-words and n-grams and then term frequency like tf-idf weighting has been done. (1) N-grams are permutations of word combinations. They help in providing context to the text by combining nearby words and making a single feature out of them

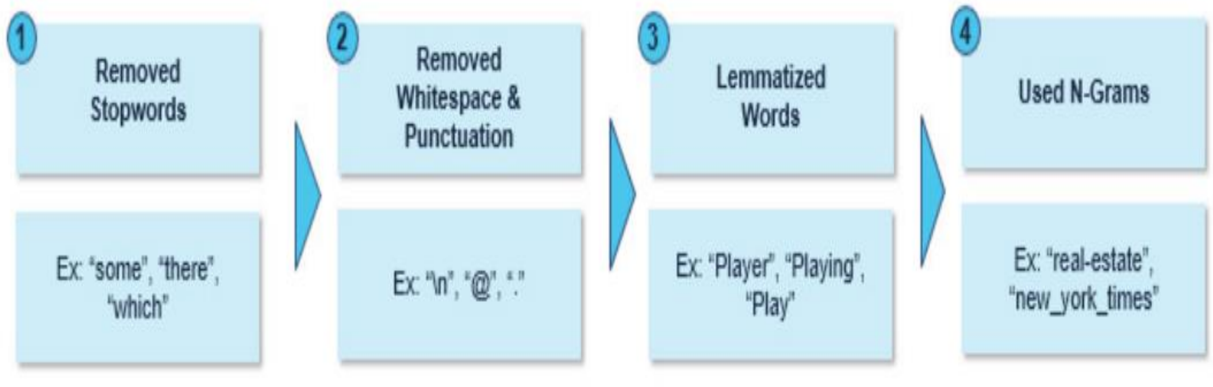

1
Removed Stopwords
Ex: “some”, “there”, “which”
2
Removed Whitespace & Punctuation
Ex: “\n”, “@”, “.”
3
Lemmatized Words
Ex: “Player”, “Playing”, “Play”
4
Used N-Grams
Ex: “real-estate”, “new_york_times”

Fig-2

3.1 Implementation

3.1.1 Data Collection & Feature Analysis Data Collection—

Guardian newspaper and Kaggle provide an API (Application Program Interface) which enables to populate the model with up-to date news. These samples data are shared among 03 files under the names:

- ’news.csv’ : which have a sample data of shape 4802 by 6 with 06 features (’Unnamed: 0’, ’title’, ’text’, ‘date,’location’,’label’) and contains a mixture of fake and real news.
- ’Fake.csv’ : which have a sample data of shape 23481 by 4 with 04 features (’title’, ’text’, ’subject’, ’date’) and contains only fake news.
- ’True.csv’ : which have a sample data of shape 21417 by 4 with 04 features (’title’, ’text’, ’subject’, ’date’) and contains only real news. Feature Analysis— From the compiled samples data obtained above, we formed our experimental dataset based on 03 features which are ’title’, ’text’, ’label’ with a shape of 51233 by 3 and contains a mixture of fake and real news.
- After merging the all data set a new dataset formed having the 6 columns
- ’article_title’: The first column contains the title of the news.
- ’article_text’ : The second column contains the plain-text news.
- ’label’ : The third column has labels denoting whether the news is REAL or FAKE.

3.1.2 Importing the Libraries

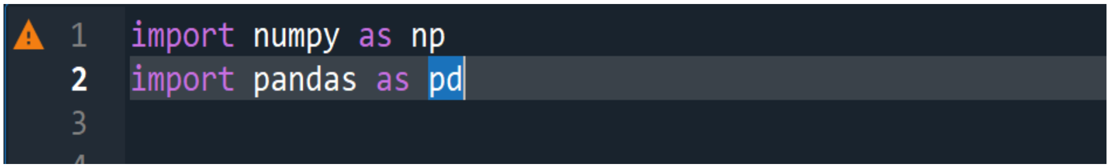
The most important libraries for data pre-processing of the data are-

a. Numpy:- The cornerstone Python module for scientific computing is called NumPy. A multidimensional array object, various derived objects (like masked arrays and matrices), and a variety of routines for quick operations on arrays are provided by this Python library. These operations include discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation, and much more.

b. Pandas:- Python's Pandas package is used to manipulate data sets. It offers tools for data exploration, cleaning, analysis, and manipulation.

3.1.3 Data pre-processing

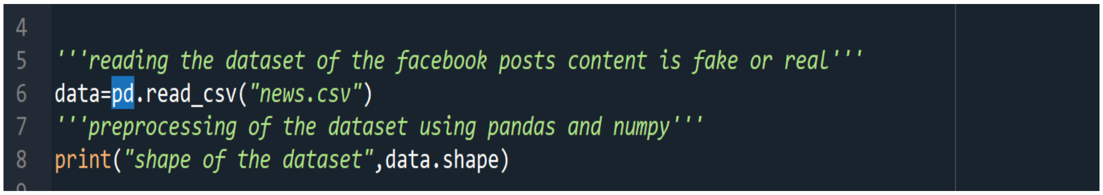
a. Any type of processing done on raw data to get it ready for another data processing operation is referred to as data preprocessing, which is a part of data preparation. It has historically been a crucial first stage in the data mining process.



```
import numpy as np
import pandas as pd
```

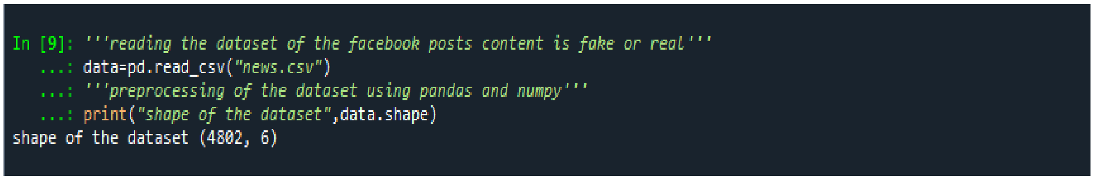
Fig-3

b. Loading the dataset through pandas library in the code for pre-preprocessing of the data and for knowing the shape of the data.



```

'''reading the dataset of the facebook posts content is fake or real'''
data=pd.read_csv("news.csv")
'''preprocessing of the dataset using pandas and numpy'''
print("shape of the dataset",data.shape)
```

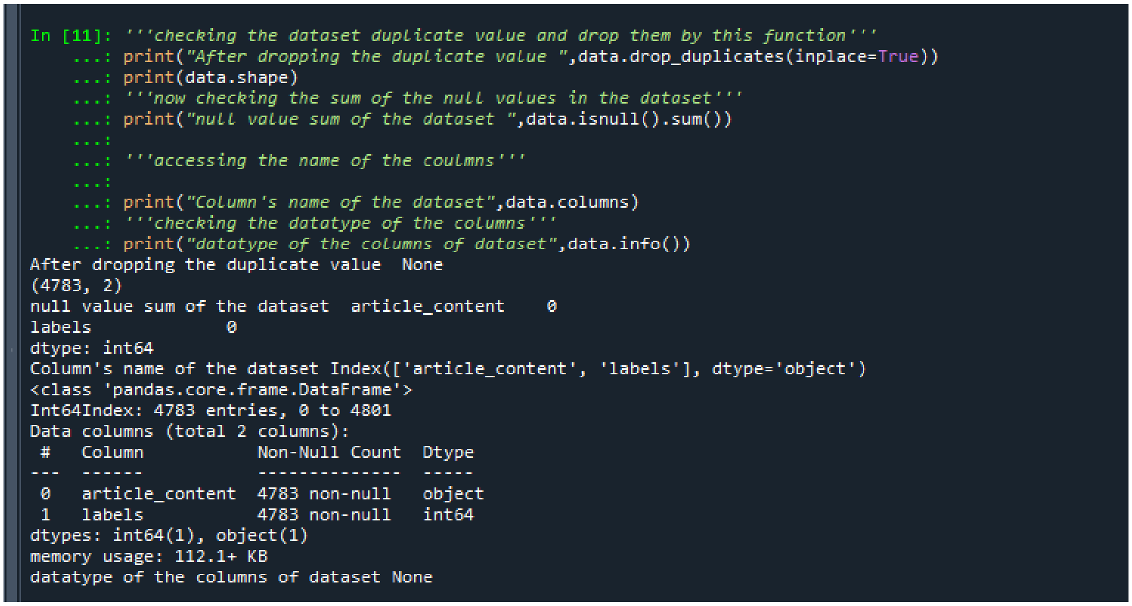
Fig-4



```
In [9]: '''reading the dataset of the facebook posts content is fake or real'''
   ...: data=pd.read_csv("news.csv")
   ...: '''preprocessing of the dataset using pandas and numpy'''
   ...: print("shape of the dataset",data.shape)
shape of the dataset (4802, 6)
```

Fig-5

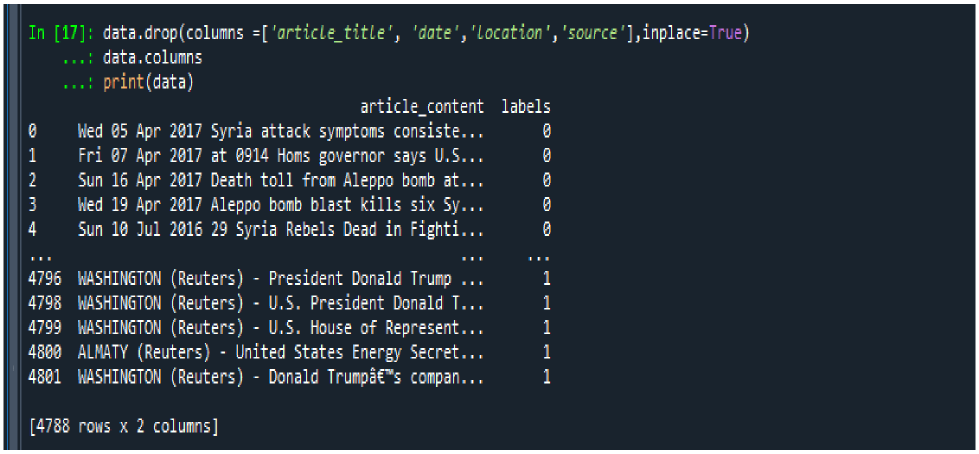
c. After loading the dataset. Now checking the duplicates values , sum of all Null values , unique value and dropping the duplicate values.



```
In [11]: '''checking the dataset duplicate value and drop them by this function'''
    ...: print("After dropping the duplicate value ",data.drop_duplicates(inplace=True))
    ...: print(data.shape)
    ...: '''now checking the sum of the null values in the dataset'''
    ...: print("null value sum of the dataset ",data.isnull().sum())
    ...:
    ...: '''accessing the name of the coulmns'''
    ...:
    ...: print("Column's name of the dataset",data.columns)
    ...: '''checking the datatype of the columns'''
    ...: print("datatype of the columns of dataset",data.info())
After dropping the duplicate value  None
(4783, 2)
null value sum of the dataset  article_content    0
labels             0
dtype: int64
Column's name of the dataset Index(['article_content', 'labels'], dtype='object')
<class 'pandas.core.frame.DataFrame'>
Int64Index: 4783 entries, 0 to 4801
Data columns (total 2 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   article_content  4783 non-null   object
 1   labels           4783 non-null   int64
dtypes: int64(1), object(1)
memory usage: 112.1+ KB
datatype of the columns of dataset None
```

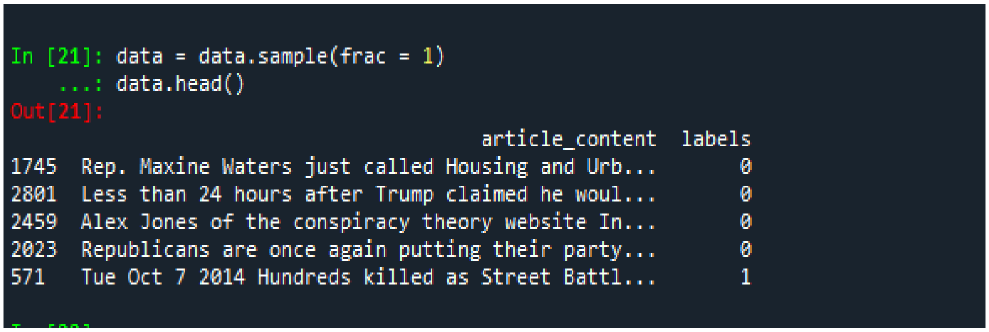
Fig-6

d . After Dropping the columns which is not required in the dataset. Shuffling also done in this dataset. After shuffling the indexing also changes. So, index is also reset.



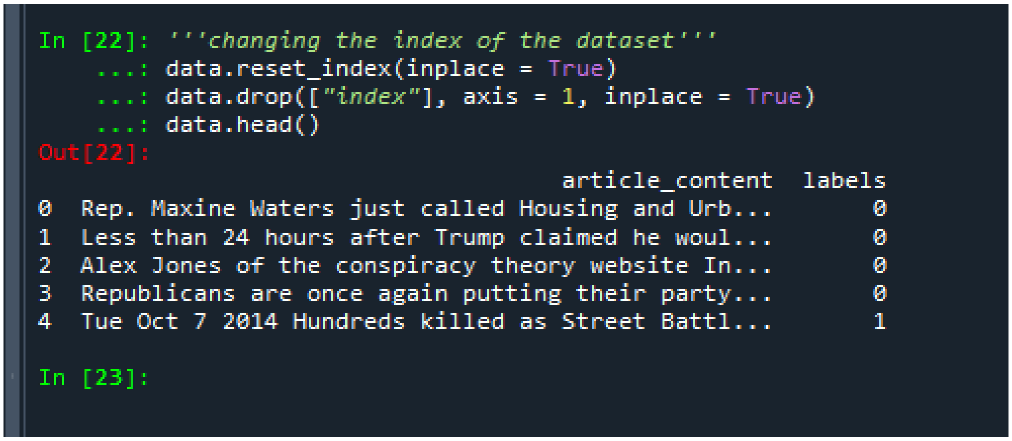
```
In [17]: data.drop(columns =['article_title', 'date','location','source'],inplace=True)
    ...: data.columns
    ...: print(data)
                                         article_content  labels
0     Wed 05 Apr 2017 Syria attack symptoms consiste...       0
1     Fri 07 Apr 2017 at 0914 Homs governor says U.S...       0
2     Sun 16 Apr 2017 Death toll from Aleppo bomb at...       0
3     Wed 19 Apr 2017 Aleppo bomb blast kills six Sy...       0
4     Sun 10 Jul 2016 29 Syria Rebels Dead in Fighti...       0
...                                                  ...     ...
4796  WASHINGTON (Reuters) - President Donald Trump ...       1
4798  WASHINGTON (Reuters) - U.S. President Donald T...       1
4799  WASHINGTON (Reuters) - U.S. House of Represent...       1
4800  ALMATY (Reuters) - United States Energy Secret...       1
4801  WASHINGTON (Reuters) - Donald Trumpâ€™s compan...       1

[4788 rows x 2 columns]
```

Fig-7



```
In [21]: data = data.sample(frac = 1)
    ...: data.head()
Out[21]:
                                        article_content  labels
1745  Rep. Maxine Waters just called Housing and Urb...       0
2801  Less than 24 hours after Trump claimed he woul...       0
2459  Alex Jones of the conspiracy theory website In...       0
2023  Republicans are once again putting their party...       0
571   Tue Oct 7 2014 Hundreds killed as Street Battl...       1
```

Fig-8



```
In [22]: '''changing the index of the dataset'''
    ...: data.reset_index(inplace = True)
    ...: data.drop(["index"], axis = 1, inplace = True)
    ...: data.head()
Out[22]:
                                     article_content  labels
0  Rep. Maxine Waters just called Housing and Urb...       0
1  Less than 24 hours after Trump claimed he woul...       0
2  Alex Jones of the conspiracy theory website In...       0
3  Republicans are once again putting their party...       0
4  Tue Oct 7 2014 Hundreds killed as Street Battl...       1

In [23]:
```

Fig-9

Data pre-processing completed of the dataset. After pre-processing we have to apply algorithms to predict the accuracy score of the dataset and plot the required maps for the fake or true news. After preprocessing the dataset saved to new file named as “filter.csv” for further used in the algorithms.

3.1.4 Algorithms Used For Model Training

a. **Decision tree**- The decision tree is a crucial tool that operates using a framework similar to a flow chart and is primarily used for categorization issues. Every internal node in the decision tree gives a condition or "test" on an attribute, and the branching is based on the results of the test. After computing all characteristics, a class label is finally applied to the leaf node. The classification rule is represented by the distance from the root to the leaf. The fact that it can be used with a category and a dependent variable is wonderful. They do an excellent job at pointing out the most crucial variables and accurately illustrating how the variables are related. They are important in developing new variables and characteristics that are helpful for data exploration and accurately forecast the target variable.

There are various benefits of using decision trees in machine learning:

- With each extra data point, the cost of using the tree to forecast data reduces.
- Works with both numerical and category data.
- Can multiple-output problems be modelled
- employs the white box model (making results easy to explain)
- The dependability of a tree can be measured and tested.

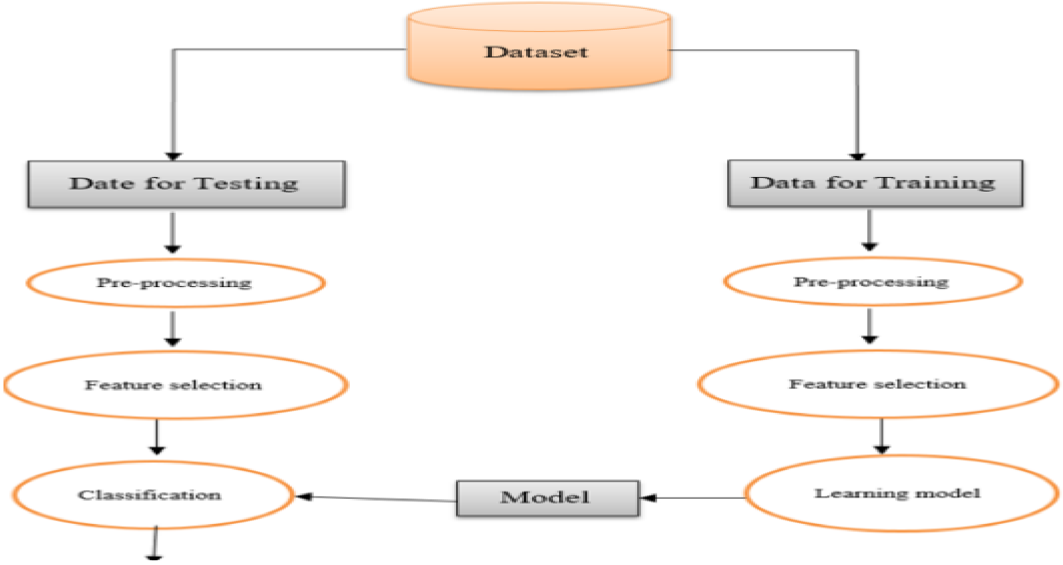
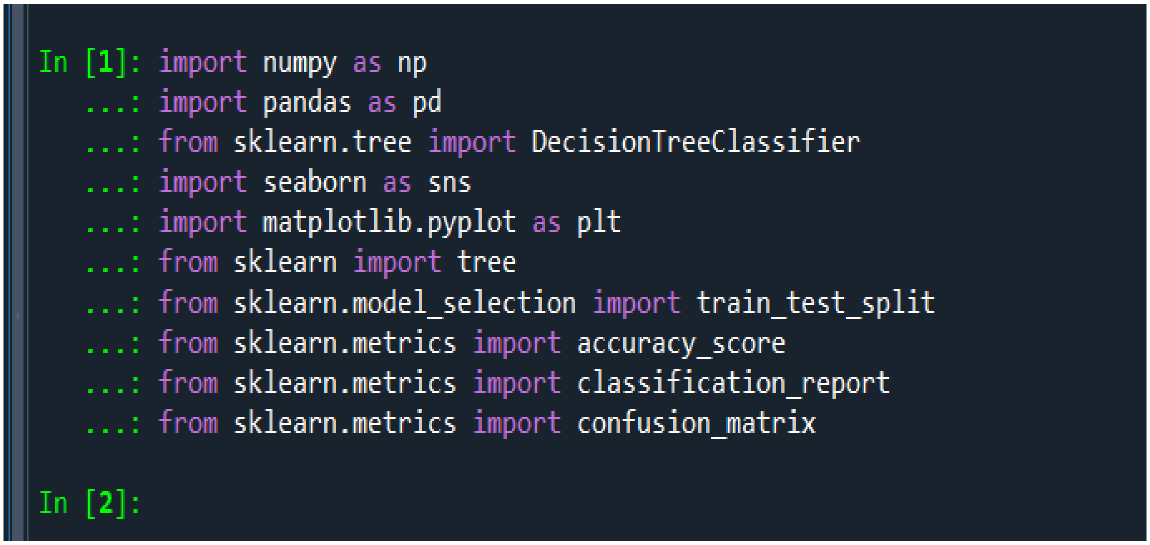


Fig-10

The main objective is to employ a variety of classification techniques to create a classification model that can be used as a scanner for fake news by identifying specifics in the news and embedded in a Python application to find fake news data. Additionally, the Python code has undergone the necessary refactoring's to create an optimized code.

k-Nearest Neighbors (k-NN), Linear Regression, XG Boost, Naive Bayes, Decision Tree, Random Forests, and Support Vector Machine are the classification techniques used in this model (SVM). All of these algorithms strive for maximum accuracy. Whenever possible, combine their averages and compare the results.

(i). Importing the important libraries which is used to trained the model.



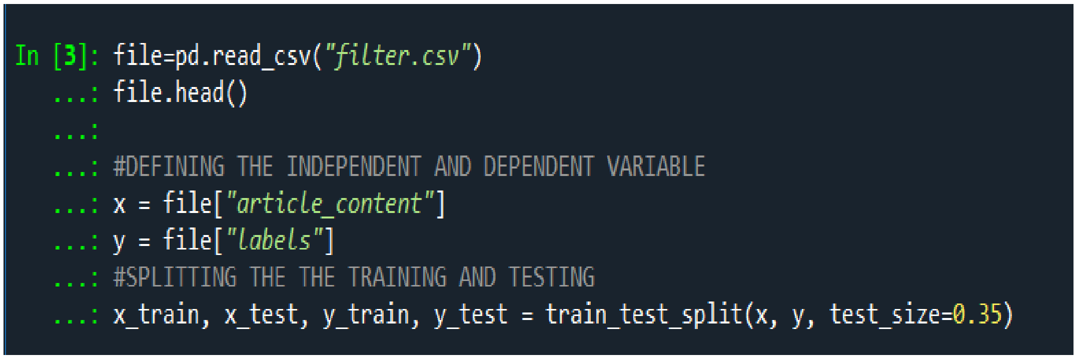
```
In [1]: import numpy as np
   ...: import pandas as pd
   ...: from sklearn.tree import DecisionTreeClassifier
   ...: import seaborn as sns
   ...: import matplotlib.pyplot as plt
   ...: from sklearn import tree
   ...: from sklearn.model_selection import train_test_split
   ...: from sklearn.metrics import accuracy_score
   ...: from sklearn.metrics import classification_report
   ...: from sklearn.metrics import confusion_matrix

In [2]:
```

Fig-11

a. **Matplotlib**-Matplotlib is a low level graph plotting library in python that serves as a visualization utility.

b. **Sklearn-** Scikit-learn (Sklearn) is the most useful and robust library for machine lerning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python.

(ii). Loading the dataset and defining the dependent and independent variable in the code and splitting the train and test set.

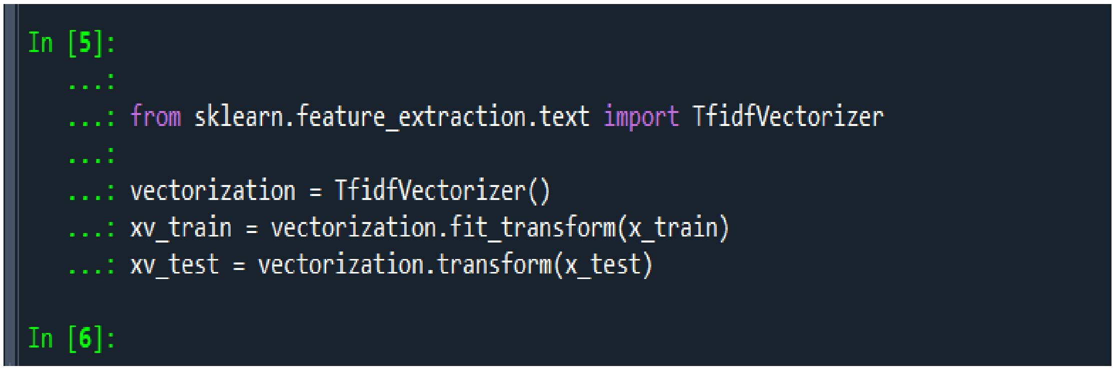


```
In [3]: file=pd.read_csv("filter.csv")
   ...: file.head()
   ...:
   ...: #DEFINING THE INDEPENDENT AND DEPENDENT VARIABLE
   ...: x = file["article_content"]
   ...: y = file["labels"]
   ...: #SPLITTING THE THE TRAINING AND TESTING
   ...: x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.35)
```

Fig-12

The train-test split procedure is used to estimate the performance of machine learning algorithms when they are used to make predictions on data not used to train the model. It is a fast and easy procedure to perform, the results of which allow you to compare the performance of machine learning algorithms for your predictive modeling problem. Although simple to use and interpret, there are times when the procedure should not be used, such as when you have a small dataset and situations where the additional configuration is required, such as when it is used for classification and the dataset is not balanced.

(iii). Now we have to convert the text into vectors for processing the news.



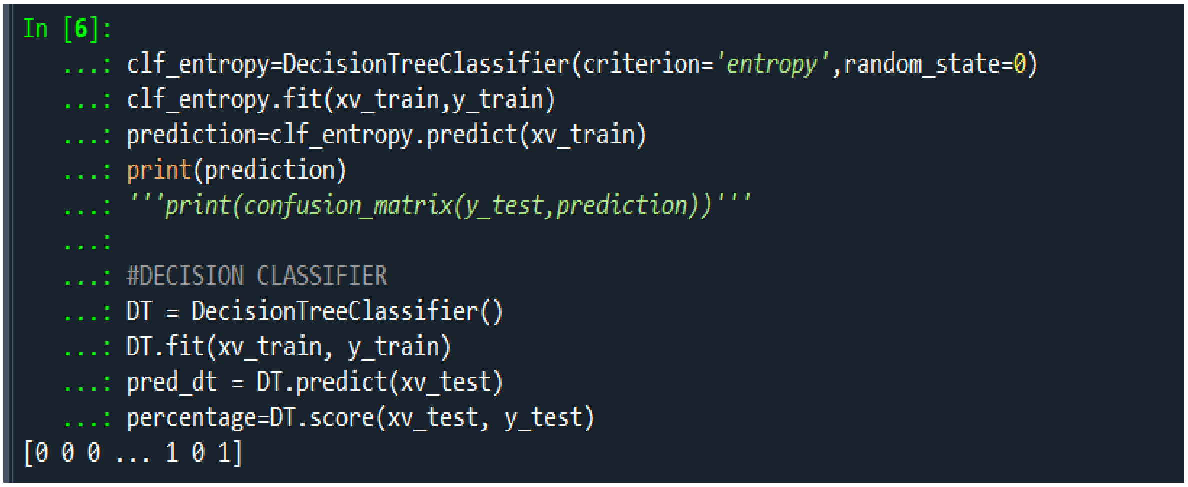
```
In [5]:
   ...:
   ...: from sklearn.feature_extraction.text import TfidfVectorizer
   ...:
   ...: vectorization = TfidfVectorizer()
   ...: xv_train = vectorization.fit_transform(x_train)
   ...: xv_test = vectorization.transform(x_test)

In [6]:
```

Fig-13

what is a TF-IDF Vectorizer?

TF-IDF is an abbreviation for Term Frequency Inverse Document Frequency. This is very common algorithm to transform text into a meaningful representation of numbers which is used to fit machine algorithm for prediction.

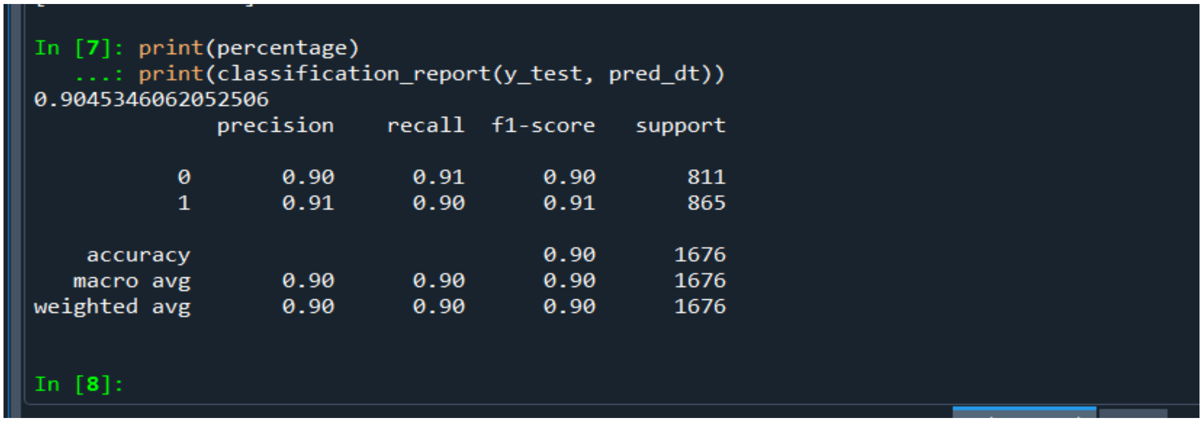
(iv). After splitting the dataset defining the entropy of the labels. Also running the classifier of the decision tree for predicting the accuracy score of the model.



```
In [6]:
   ...: clf_entropy=DecisionTreeClassifier(criterion='entropy',random_state=0)
   ...: clf_entropy.fit(xv_train,y_train)
   ...: prediction=clf_entropy.predict(xv_train)
   ...: print(prediction)
   ...: '''print(confusion_matrix(y_test,prediction))'''
   ...:
   ...: #DECISION CLASSIFIER
   ...: DT = DecisionTreeClassifier()
   ...: DT.fit(xv_train, y_train)
   ...: pred_dt = DT.predict(xv_test)
   ...: percentage=DT.score(xv_test, y_test)
[0 0 0 ... 1 0 1]
```

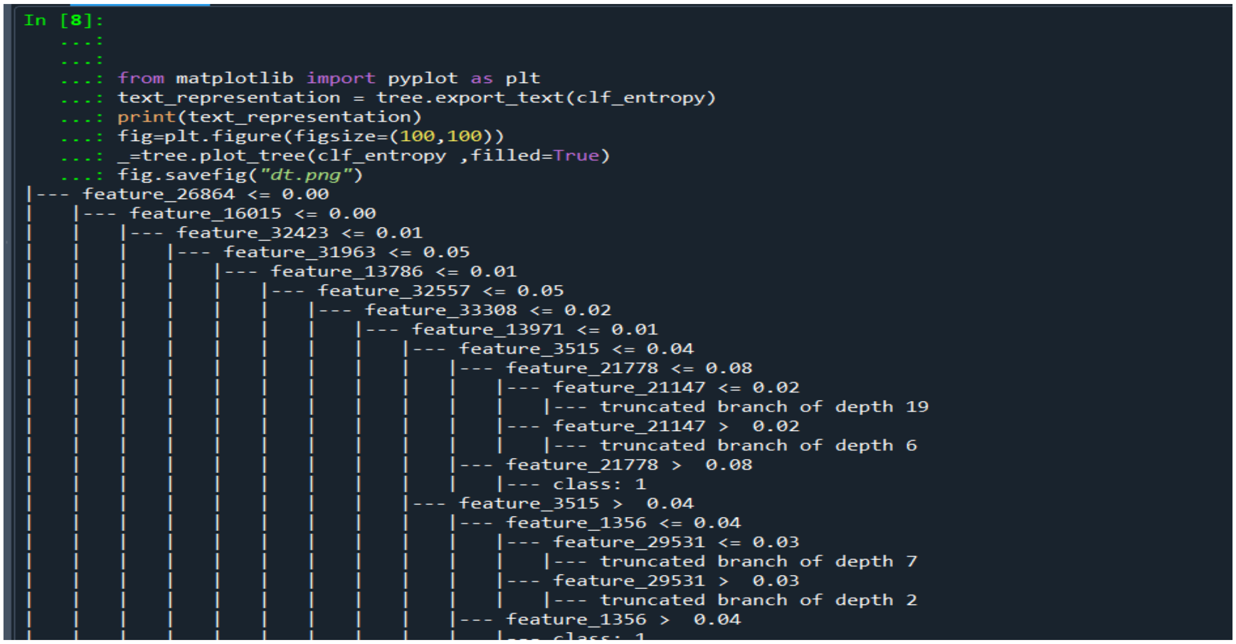
Fig-14

(iv). After prediction the accuracy score of the model plotting it to the graph.



```
In [7]: print(percentage)
   ...: print(classification_report(y_test, pred_dt))
0.9045346062052506
              precision    recall  f1-score   support

           0       0.90      0.91      0.90       811
           1       0.91      0.90      0.91       865

    accuracy                           0.90      1676
   macro avg       0.90      0.90      0.90      1676
weighted avg       0.90      0.90      0.90      1676


In [8]:
```

Fig-15(classification report and accuracy score)



```
In [8]:
   ...:
   ...:
   ...: from matplotlib import pyplot as plt
   ...: text_representation = tree.export_text(clf_entropy)
   ...: print(text_representation)
   ...: fig=plt.figure(figsize=(100,100))
   ...: _=tree.plot_tree(clf_entropy ,filled=True)
   ...: fig.savefig("dt.png")
|--- feature_26864 <= 0.00
|   |--- feature_16015 <= 0.00
|   |   |--- feature_32423 <= 0.01
|   |   |   |--- feature_31963 <= 0.05
|   |   |   |   |--- feature_13786 <= 0.01
|   |   |   |   |   |--- feature_32557 <= 0.05
|   |   |   |   |   |   |--- feature_33308 <= 0.02
|   |   |   |   |   |   |   |--- feature_13971 <= 0.01
|   |   |   |   |   |   |   |   |--- feature_3515 <= 0.04
|   |   |   |   |   |   |   |   |   |--- feature_21778 <= 0.08
|   |   |   |   |   |   |   |   |   |   |--- feature_21147 <= 0.02
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 19
|   |   |   |   |   |   |   |   |   |   |--- feature_21147 >  0.02
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 6
|   |   |   |   |   |   |   |   |   |--- feature_21778 >  0.08
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- feature_3515 >  0.04
|   |   |   |   |   |   |   |   |   |--- feature_1356 <= 0.04
|   |   |   |   |   |   |   |   |   |   |--- feature_29531 <= 0.03
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 7
|   |   |   |   |   |   |   |   |   |   |--- feature_29531 >  0.03
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 2
|   |   |   |   |   |   |   |   |   |--- feature_1356 >  0.04
|   |   |   |   |   |   |   |   |   |   |--- class: 1
```

Fig-16

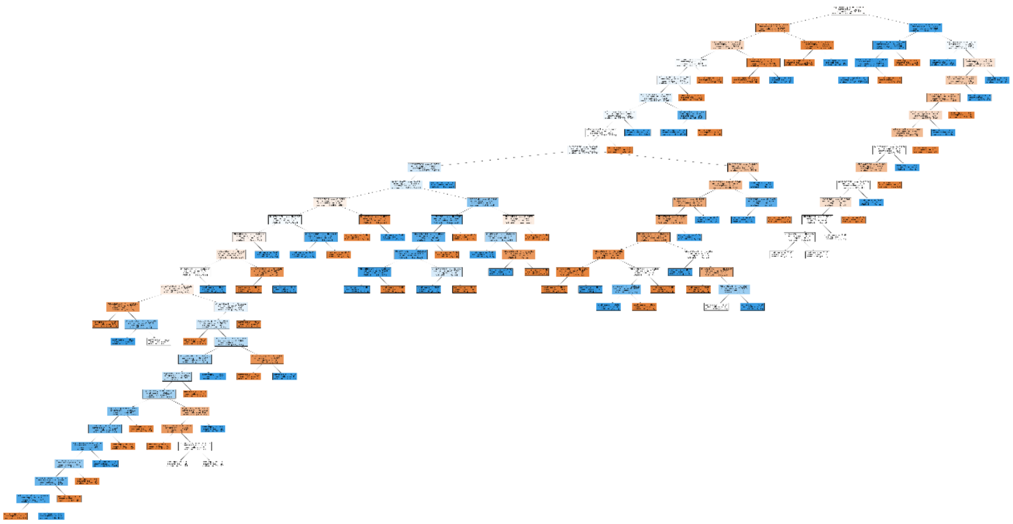
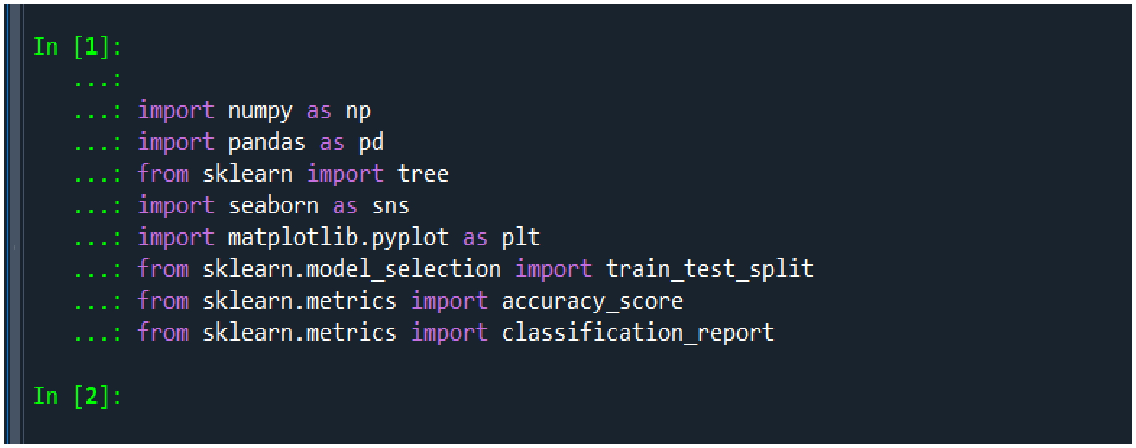


Fig-17

b. **Random forest-**Random Forest are built on the concept of building many decision tree algorithms, after which the decision trees get a separate result. The results, which are predicted by large number of decision tree,are taken up by the random forest. To ensure a variation of the decision trees, the random forest randomly selects a subcategory of properties from each group The applicability of Random forest is best when used on uncorrelated decision trees. If applied on similar trees, the overall result will be more or less similar to a single decision tree. Uncorrelated decision trees can be obtained by bootstrapping and feature randomness.

(i). Importing the important libraries which is used to trained the model.

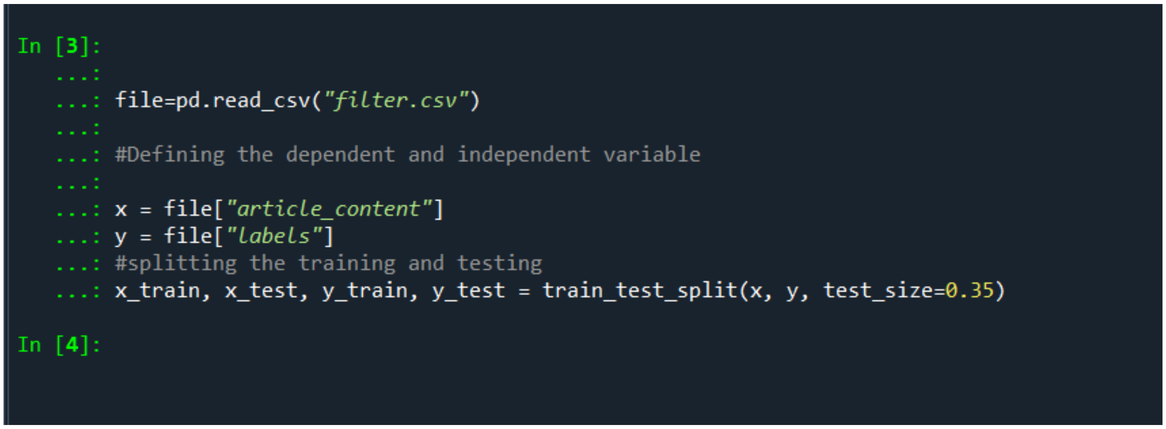


```
In [1]:
   ...:
   ...: import numpy as np
   ...: import pandas as pd
   ...: from sklearn import tree
   ...: import seaborn as sns
   ...: import matplotlib.pyplot as plt
   ...: from sklearn.model_selection import train_test_split
   ...: from sklearn.metrics import accuracy_score
   ...: from sklearn.metrics import classification_report

In [2]:
```

Fig-18

- Accuracy score is used to measure the model performance in terms of measuring the ratio of sum of true positive and true negatives out of all the predictions made.
- The model precision score measures the proportion of positively predicted labels that are actually correct. Precision is also known as the positive predictive value. Precision is used in conjunction with the recall to trade-off false positives and false negatives.

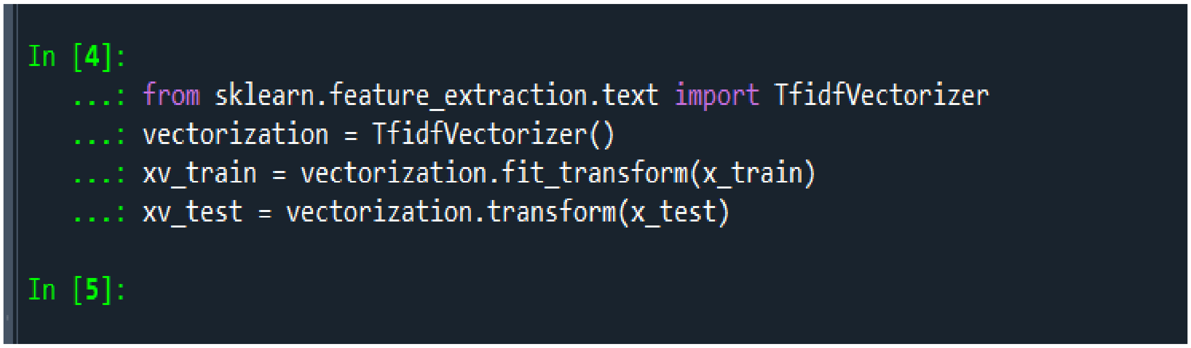
(ii). Loading the dataset and defining the dependent and independent variable in the code and splitting the train and test set.



```
In [3]:
   ...:
   ...: file=pd.read_csv("filter.csv")
   ...:
   ...: #Defining the dependent and independent variable
   ...:
   ...: x = file["article_content"]
   ...: y = file["labels"]
   ...: #splitting the training and testing
   ...: x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.35)

In [4]:
```

Fig-19

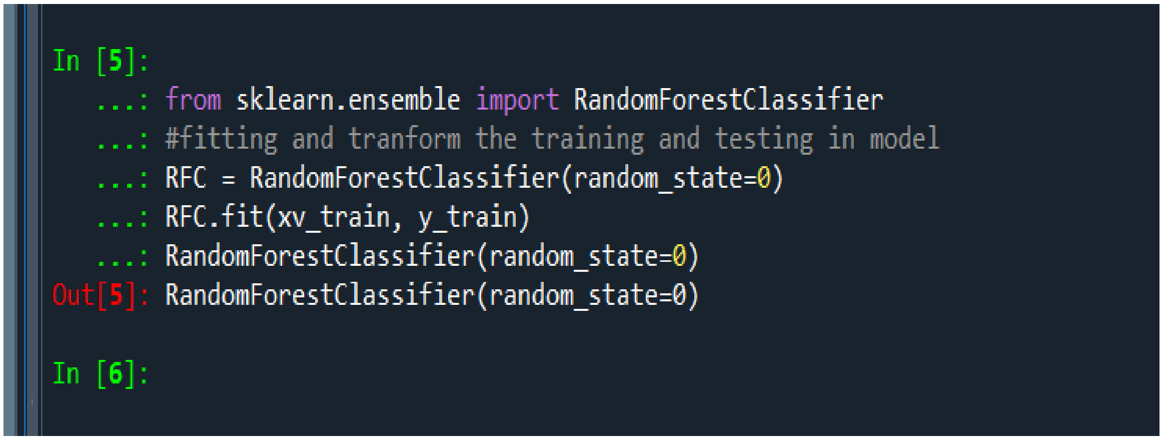
(iii). Now we have to convert the text into vectors for processing the news.



```
In [4]:
   ...: from sklearn.feature_extraction.text import TfidfVectorizer
   ...: vectorization = TfidfVectorizer()
   ...: xv_train = vectorization.fit_transform(x_train)
   ...: xv_test = vectorization.transform(x_test)

In [5]:
```

Fig-20

(iv). After splitting the dataset defining the entropy of the labels. Also running the classifier of the decision tree for predicting the accuracy score of the model.

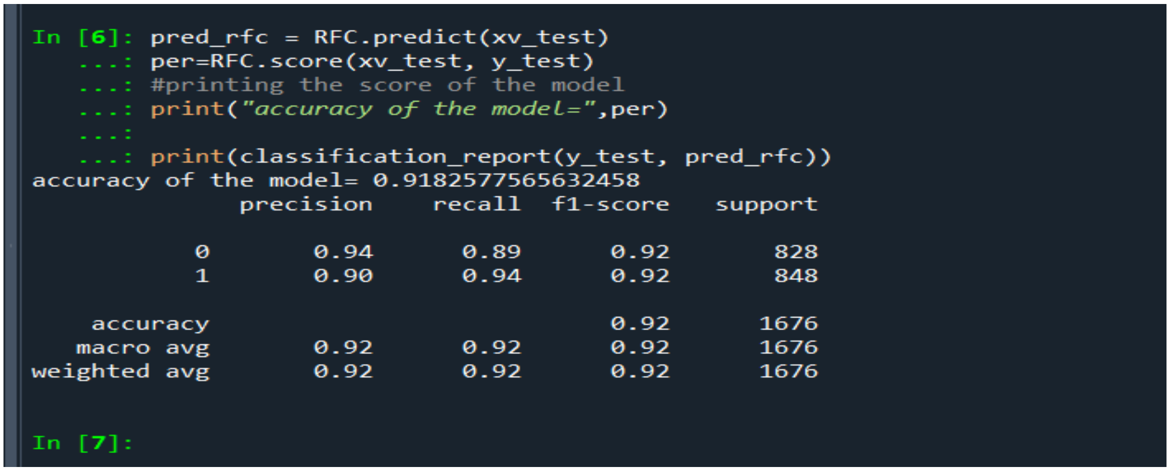


```
In [5]:
   ...: from sklearn.ensemble import RandomForestClassifier
   ...: #fitting and tranform the training and testing in model
   ...: RFC = RandomForestClassifier(random_state=0)
   ...: RFC.fit(xv_train, y_train)
   ...: RandomForestClassifier(random_state=0)
Out[5]: RandomForestClassifier(random_state=0)

In [6]:
```

Fig-21

Running the Classifier

Pre-Execution—— Before executing the classifier, we use TF-IDF Vectorizer to sanitize, transform and preprocess both X-Train and X-Test as mentioned in the above.Thus, to obtain both TF IDF-Train and TF IDF-Test respectively (with more features) which are vectorized versions of X-Train and X-Test. Execution.We use the obtained vectorized versions TF IDF-Train and TF IDF-Test to train and test.



```
In [6]: pred_rfc = RFC.predict(xv_test)
   ...: per=RFC.score(xv_test, y_test)
   ...: #printing the score of the model
   ...: print("accuracy of the model=",per)
   ...:
   ...: print(classification_report(y_test, pred_rfc))
accuracy of the model= 0.9182577565632458
              precision    recall  f1-score   support

           0       0.94      0.89      0.92       828
           1       0.90      0.94      0.92       848

    accuracy                           0.92      1676
   macro avg       0.92      0.92      0.92      1676
weighted avg       0.92      0.92      0.92      1676

In [7]:
```

Fig-22

c. **Logistic Regression-** One of the most often used Machine Learning algorithms, within the category of Supervised Learning, is logistic regression. Using a predetermined set of independent factors, it is used to predict the categorical dependent variable. In a categorical dependent variable, the output is predicted via logistic regression**.**

- Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets.

- Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification.
- Logistic regression uses the concept of predictive modeling as regression; therefore, it is called logistic regression, but is used to classify samples; Therefore, it falls under the classification algorithm.

Workflow-

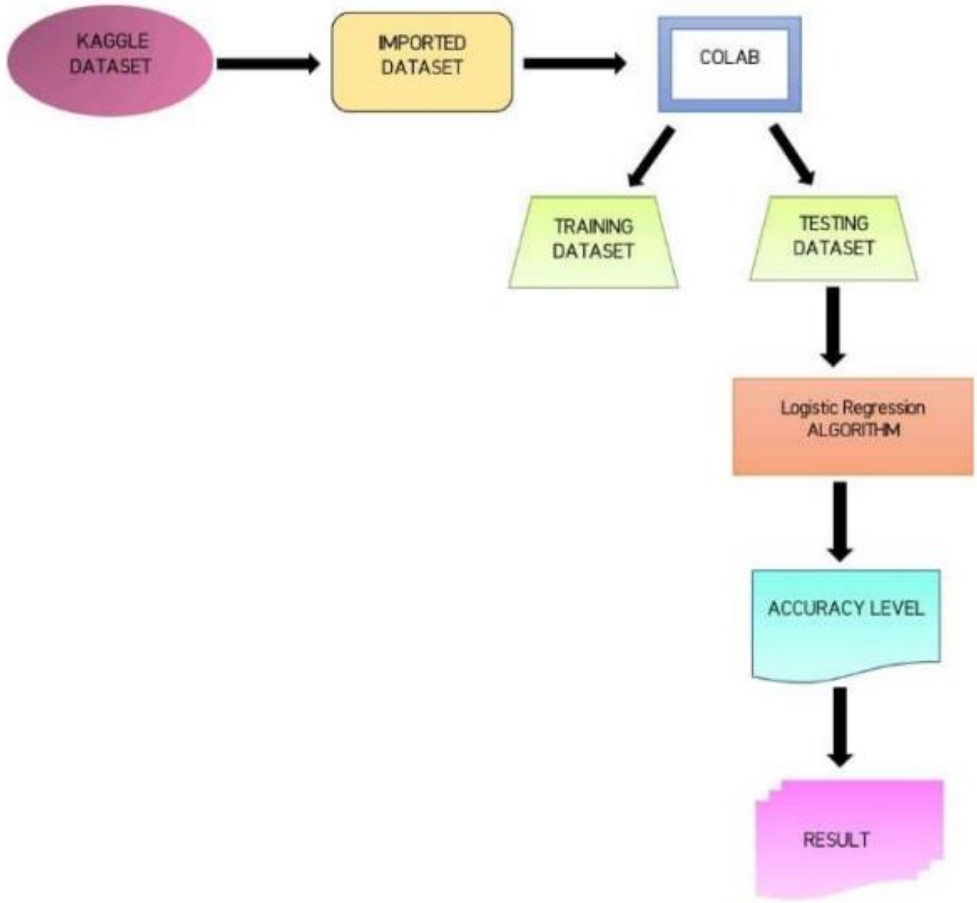
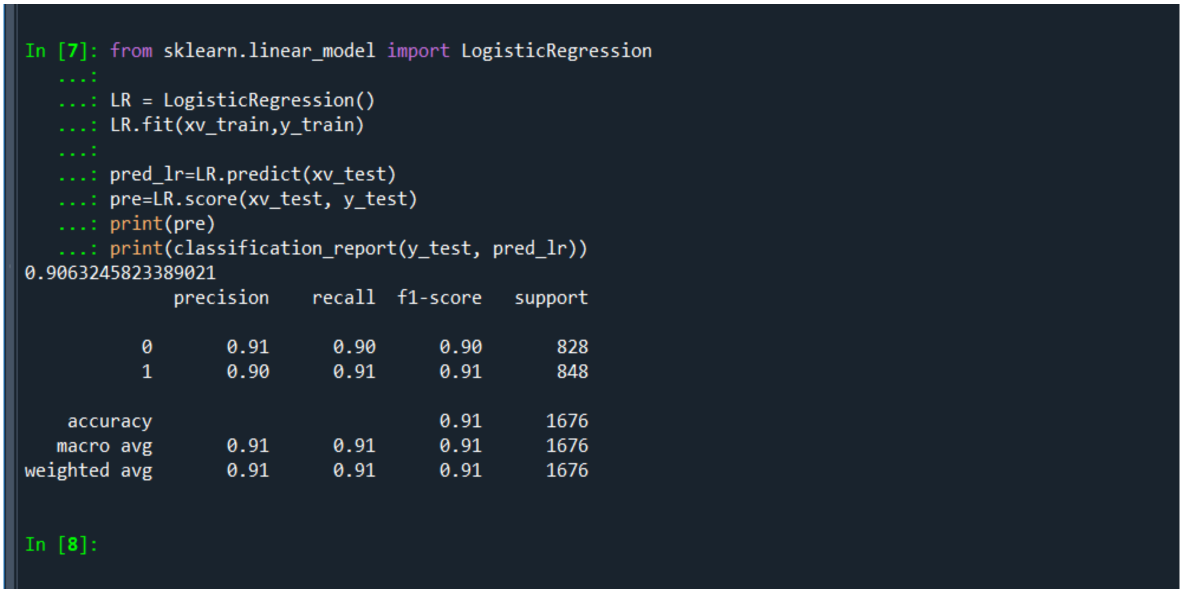


Fig-23

In this all the steps are same as above algorithms. so now the accuracy score of the model is shown in the below figure-



```
In [7]: from sklearn.linear_model import LogisticRegression
   ...:
   ...: LR = LogisticRegression()
   ...: LR.fit(xv_train,y_train)
   ...:
   ...: pred_lr=LR.predict(xv_test)
   ...: pre=LR.score(xv_test, y_test)
   ...: print(pre)
   ...: print(classification_report(y_test, pred_lr))
0.9063245823389021
              precision    recall  f1-score   support

           0       0.91      0.90      0.90       828
           1       0.90      0.91      0.91       848

    accuracy                           0.91      1676
   macro avg       0.91      0.91      0.91      1676
weighted avg       0.91      0.91      0.91      1676


In [8]:
```

Fig-24

The accuracy of predictions made by a classification algorithm is evaluated using a classification report. how many of the forecasts came true and how many didn't. More specifically, the metrics of a classification report are predicted using True Positives, False Positives, True Negatives, and False Negatives, as illustrated above.

4. Result Analysis

1. The study's findings demonstrate that the Decision tree algorithm is 90 percent accurate at identifying bogus news sources.
2. The study's findings demonstrate that the Random Forest algorithm is 98 percent accurate at identifying bogus news sources.
3. The study's findings demonstrate that the Logistic regression algorithm is 90 percent accurate at identifying bogus news sources.

4. Conclusion

Fake news research has never been more important than it is now. Especially during a time when the world is fighting a pandemic. The approaches explored in this article only scratch the surface. There are so many more approaches and criteria for fake news detection. Datasets also impact the accuracy of fake news detection tasks.

Their quality and quantity are impactful. It is also worth noting that, as much as our focus is on automated approaches, the human element is key to this fight. A combination of human and automated approaches gives rise to a hybrid approach. I hope this article challenges you to join the fight against fake news by creating better and greater solutions.

5. References

I. kaggle kernels output therealsampat/fake-news-detection -p /path/to/dest.

II. An_Empirical_Analysis_of_Nave_Bayes_SVM_Logistic_Regression_and_Random_Forest_to_Spot_False_Information_in_Real-World_Networks.pdf

III. A_System_for_Fake_News_Detection_by_using_Supervised_Learning_Model_for_Social_Media_Contents.pdf

IV. Fake_News_Detection_in_Social_Networks_Using_Machine_Learning_and_Deep_Learning_Performance_Evaluation.pdf