DATA SCIENCE

(Fake News Classifier)

Summer Internship Report Submitted in partial fulfilment

of the requirement for undergraduate degree of

Bachelor of Technology

In

Computer Science and Engineering

By

Hemanth Reddy Galigutta

221810302021

Under the Guidance of

Assistant Professor

Department of Electronics and Communication Engineering
GITAM School of Technology
GITAM (Deemed to be University)
Hyderabad-502329

July 2020

DECLARATION

I submit this industrial training work entitled **“DATA SCIENCE**” to GITAM (Deemed TO Be University), Hyderabad in partial fulfilment of the requirements for the award of the degree of “**Bachelor of Technology”** in “**Computer Science and Engineering”**. I declare that itwas carried out independently by me under the guidance of____________________, Asst. Professor, GITAM (Deemed to Be University), Hyderabad, India.

The results embodied in this report have not been submitted to any other University or Institute for the award of any degree or diploma.

Place: HYDERABAD

Student Name: Hemanth Reddy Galigutta

Date:

Student Roll No: 221810302021

GITAM (DEEMED TO BE UNIVERSITY)

Hyderabad-502329, India

Dated:

CERTIFICATE

This is to certify that the Industrial Training Report entitled **“DATA SCIENCE”** is being submitted by Hemanth Reddy Galigutta (221810302021) in partial fulfilment of the requirement for the award of **Bachelor of Technology in Computer Science and Engineering** at GITAM (Deemed to Be University), Hyderabad during the academic year 2020-21

ACKNOWLEDGEMENT

Apart from my effort, the success of this internship largely depends on the encouragement and guidance of many others. I take this opportunity to express my gratitude to the people who have helped me in the successful competition of this internship.

I would like to thank respected **Dr. N. Siva Prasad,** Pro Vice Chancellor, GITAM Hyderabad and **Dr. N Seetha Ramaiah,** Principal, GITAM Hyderabad

I would like to thank respected **S. Phani Kumar,** Head of the Department of Electronicsand Communication Engineering for giving me such a wonderful opportunity to expand my knowledge for my own branch and giving me guidelines to present a internship report. It helped me a lot to realize of what we study for.

I would like to thank the respected faculties who helped me to make this internship a successful accomplishment.

I would also like to thank my friends who helped me to make my work more organized and well-stacked till the end.

Student Name: Hemanth Reddy Galigutta

Student Pin no: 221810302021

INDEX:

LIST OF FIGURES:

CHAPTER 1

DATA SCIENCE

1.1 INTRODUCTION:

Data science is the study of the extraction of knowledge from data. It uses various techniques from many fields, including signal processing, mathematics, probability, machine learning, computer programming, statistics, data engineering, pattern matching, and data visualization, with the goal of extracting useful knowledge from the data. With computer systems able to handle more data, big data is an important aspect of data science

1.2 DATA SCIENCE DOMAINS:

Data science is a multidisciplinary area that uses scientific methods, procedures, tools and systems to extract knowledge and get insights into structured and unstructured data. Data science is related to data analytics, data mining and big data. It understands the phenomenon of the data. It employs techniques and theories drawn from many fields within the context of mathematics, statistics, computer science, and information science.

Statistics is one of the most important disciplines to provide tools and methods to find structure in and to give deeper insight into data, and the most important discipline to analyze and quantify uncertainty. Python provides various predefined modules to work on Data science projects.

Data science employs techniques and theories drawn from many fields within the context of mathematics, statistics, computer science, and information science.

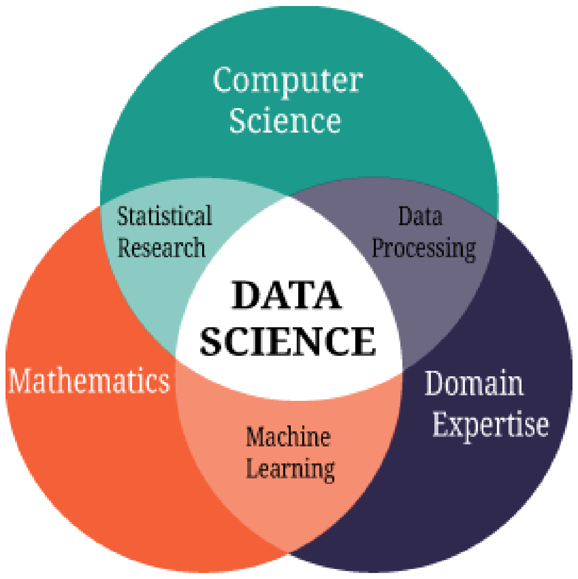


Figure 1.2.1: Data science Domains.

1.3 MACHINE LEARNING IN DATA SCIENCE:

The term **machine learning** refers to the automated detection of meaningful patterns in data. In the past couple of decades it has become a common tool in almost any task that requires information extraction from large data sets. We are surrounded by a machine learning based technology: search engines learn how to bring us the best results (while placing profitable ads), anti-spam software learns to filter our email messages, and credit card transactions are secured by software that learns how to detect frauds. Digital cameras learn to detect faces and intelligent personal assistance applications on smartphones learn to recognize voice commands.

Cars are equipped with accident prevention systems that are built using machine learning algorithms. Machine learning is also widely used in scientific applications such as bioinformatics, medicine, and astronomy.

One common feature of all of these applications is that, in contrast to more traditional uses of computers, in these cases, due to the complexity of the patterns that need to be detected, a human programmer cannot provide an explicit, fine detailed specification of how such tasks should be executed. Taking example from intelligent beings, many of our skills are acquired or refined through learning from our experience (rather than following explicit instructions given to us). Machine learning tools are concerned with endowing programs with the ability to "learn "and adapt.

Because machine learning is typically used to process large volumes of data, you may want to choose a powerful low-level language. However, if you're only just beginning to explore this field, it might be better to start with Python. Python is beginner-friendly, and can do the same thing that other coding languages can, but in fewer lines of code. If you are interested in exploring machine learning with Python, this paper will serve as your guide. This is paper gives overview of programming machine learning using Python.

1.4 USES OF MACHINE LEARNING:

To understand the concept of machine learning better, let's consider some more examples: web search results, real-time ads on web pages and mobile devices, email spam filtering, network intrusion detection, and pattern and image recognition. All these are by-products of applying machine learning to analyse huge volumes of data

Traditionally, data analysis was always being characterized by trial and error, an approach that becomes impossible when data sets are large and heterogeneous. Machine learning comes as the solution to all this chaos by proposing clever alternatives to analysing huge volumes of data.

By developing fast and efficient algorithms and data-driven models for real-time processing of data, machine learning can produce accurate results and analysis.

1.5 TYPES OF LEARNING ALGORITHMS:

The types of machine learning algorithms differ in their approach, the type of data they input and output, and the type of task or problem that they are intended to solve.

1.5.1 Supervised Learning:

When an algorithm learns from example data and associated target responses that can consist of numeric values or string labels, such as classes or tags, in order to later predict the correct response when posed with new examples comes under the category of supervised learning.

Supervised machine learning algorithms uncover insights, patterns, and relationships from a labelled training dataset – that is, a dataset that already contains a known value for the target variable for each record. Because you provide the machine learning algorithm with the correct answers for a problem during training, it is able to "learn" how the rest of the features relate to the target, enabling you to uncover insights and make predictions about future outcomes based on historical data.

Examples of Supervised Machine Learning Techniques are Regression, in which the algorithm returns a numerical target for each example, such as how much revenue will be generated from a new marketing campaign.

Classification, in which the algorithm attempts to label each example by choosing between two or more different classes. Choosing between two classes is called binary classification, such as determining whether or not someone will default on a loan. Choosing between more than two classes is referred to as multiclass classification.

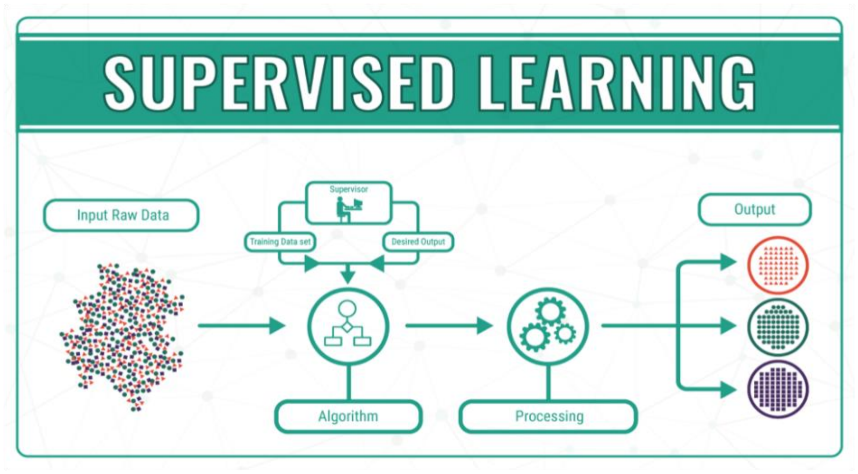


Figure 1.5.1.1: Supervised Learning

1.5.2 Unsupervised Learning:

When an algorithm learns from plain examples without any associated response, leaving to the algorithm to determine the data patterns on its own. This type of algorithm tends to restructure the data into something else, such as new features that may represent a class or a new series of uncorrelated values. They are quite useful in providing humans with insights into the meaning of data and new useful inputs to supervised machine learning algorithms.

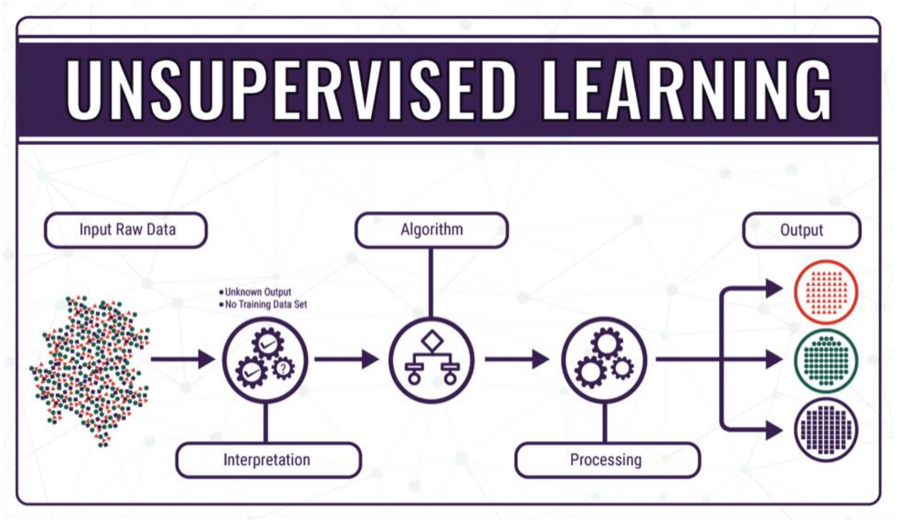


Figure 1.5.2.1: Unsupervised Learning

Popular techniques where unsupervised learning is used also include self-organizingmaps, nearest neighbour mapping, singular value decomposition, and k-means clustering. Basically, online recommendations, identification of data outliers, and segment text topics are all examples of unsupervised learning.

1.5.3 Semi Supervised Learning:

As the name suggests, semi-supervised learning is a bit of both supervised and unsupervised learning and uses both labelled and unlabelled data for training. In a typical scenario, the algorithm would use a small amount of labelled data with a large amount of unlabelled data.

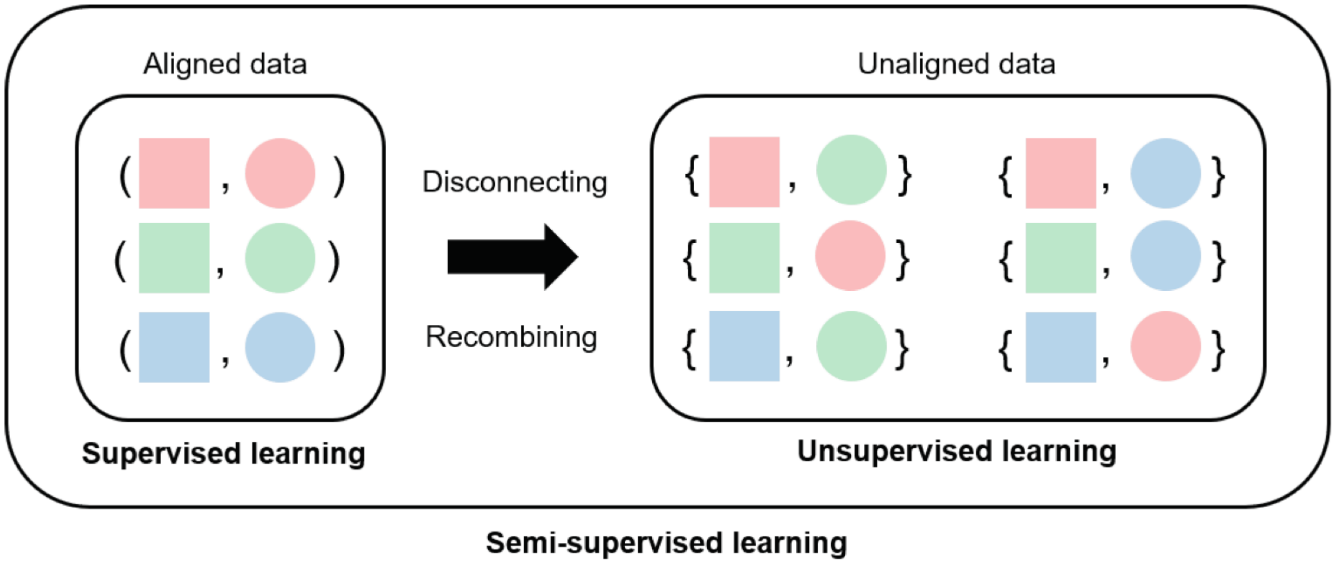


Figure 1.5.3.1: Semi Supervised Learning

CHAPTER 2

PYTHON

Basic programming language used for Data science learning is: PYTHON

2.1 INTRODUCTION TO PYHTON:

- Python is a high-level, interpreted, interactive and object-oriented scripting language.
- Python is a general-purpose programming language that is often applied in scripting roles
- Python is Interpreted: Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is like PERL and PHP.
- Python is Interactive: You can sit at a Python prompt and interact with the interpreter directly to write your programs.
- Python is Object-Oriented: Python supports the Object-Oriented style or technique of programming that encapsulates code within objects.

2.2 HISTORY OF PYTHON:

- Python was developed by GUIDO VAN ROSSUM in early 1990's
- Its latest version is 3.7, it is generally called as python3

2.3 FEATURES OF PYTHON:

- Easy-to-learn: Python has few keywords, simple structure, and a clearly defined syntax, This allows the student to pick up the language quickly.

- Easy-to-read: Python code is more clearly defined and visible to the eyes.
- Easy-to-maintain: Python's source code is fairly easy-to-maintaining.
- A broad standard library: Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- Portable: Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- Extendable: You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- Databases: Python provides interfaces to all major commercial databases.
- GUI Programming: Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.

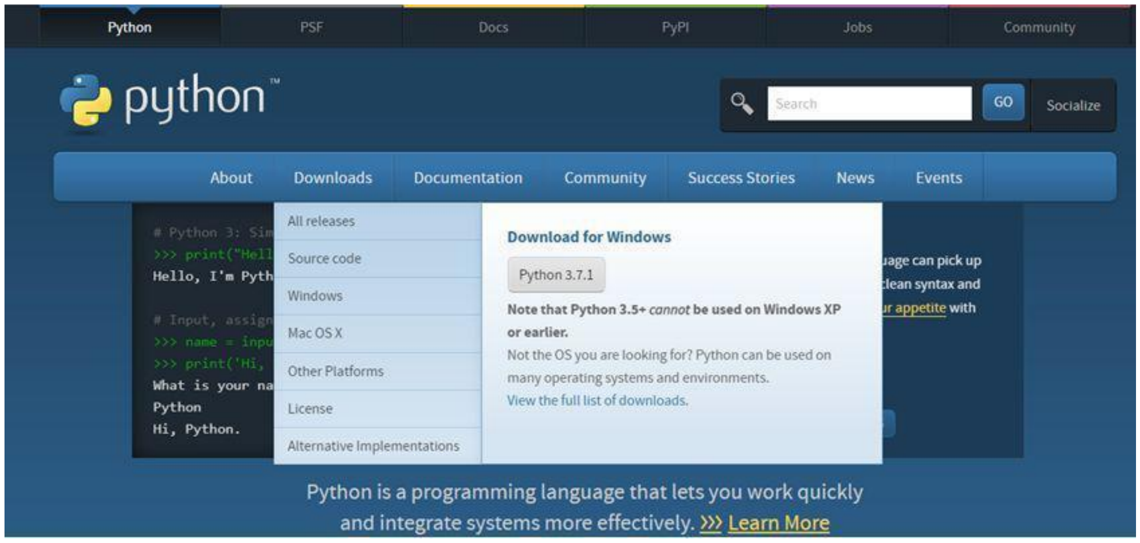
2.4 HOW TO SETUP PYTHON:

- Python is available on a wide variety of platforms including Linux and Mac OS X. Let's understand how to set up our Python environment.
- The most up-to-date and current source code, binaries, documentation, news, etc., is available on the official website of Python.

2.4.1 Installation (using python IDLE):

- Installing python is generally easy, and nowadays many Linux and Mac OS distributions include a recent python.

- Download python from www.python.org
- When the download is completed, double click the file and follow the instructions to install it.
- When python is installed, a program called IDLE is also installed along with it. It provides a graphical user interface to work with python.



```
# Python 3: Sim
>>> print("Hell
Hello, I'm Pyth

# Input, assign
>>> name = inpu
>>> print('Hi,
What is your na
Python
Hi, Python.
```

Note that Python 3.5+ *cannot* be used on Windows XP or earlier.

Not the OS you are looking for? Python can be used on many operating systems and environments.

View the full list of downloads.

uage can pick up
clean syntax and
ur appetite with

Python is a programming language that lets you work quickly and integrate systems more effectively. >>> Learn More

Figure 2.4.1.1: Python download

2.4.2 Installation (using Anaconda):

- Python programs are also executed using Anaconda.
- Anaconda is a free open source distribution of python for large scale data processing, predictive analytics and scientific computing.
- Conda is a package manager quickly installs and manages packages.

- In windows
  - Step 1: Open Anaconda.com/downloads in web browser.
  - Step 2: Download python 3.4 version for (32-bitgraphic installer/64 -bit graphic installer)
  - Step 3: select installation type(all users)
  - Step 4: Select path(i.e. add anaconda to path & register anaconda as default python 3.4) next click install and next click finish
  - Step 5: Open jupyter notebook (it opens in default browser)

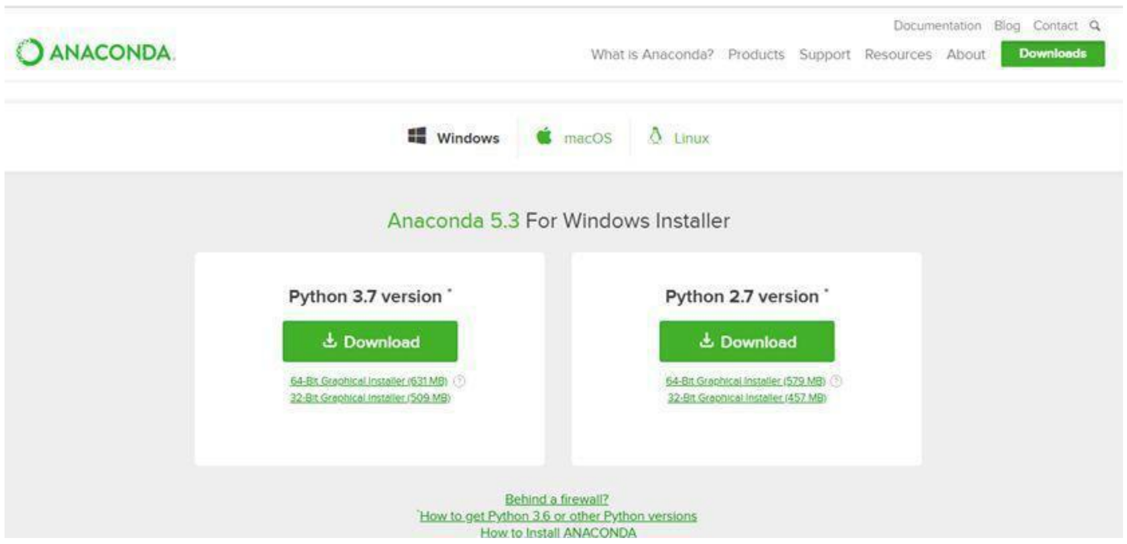

ANACONDA.
Documentation Blog Contact
What is Anaconda? Products Support Resources About Downloads
Windows macOS Linux
Anaconda 5.3 For Windows Installer
Python 3.7 version *
Download
64-Bit Graphical Installer (631 MB)
32-Bit Graphical Installer (509 MB)
Python 2.7 version *
Download
64-Bit Graphical Installer (579 MB)
32-Bit Graphical Installer (457 MB)
Behind a firewall?
How to get Python 3.6 or other Python versions
How to Install ANACONDA

Figure2.4.2.1: Anaconda download

Figure 2.4.2.2: Jupyter notebook

2.5 PYTHON VARIABLE TYPES:

- Variables are nothing but reserved memory locations to store values. This means that when you create a variable you reserve some space in memory.
- Variables are nothing but reserved memory locations to store values.
- Based on the data type of a variable, the interpreter allocates memory and decides what can be stored in the reserved memory.
- Python variables do not need explicit declaration to reserve memory space. The declaration happens automatically when you assign a value to a variable.
- Python has various standard data types that are used to define the operations possible on them and the storage method for each of them.
- Python has five standard data types –
    - Numbers
    - Strings
    - Lists

- Tuples
- Dictionary

2.5.1 Python Numbers:

- Number data types store numeric values. Number objects are created when you assign a value to them.
- Python supports four different numerical types − int (signed integers) long (long integers, they can also be represented in octal and hexadecimal) float (floating point real values) complex (complex numbers).

2.5.2 Python Strings:

- Strings in Python are identified as a contiguous set of characters represented in the quotation marks.
- Python allows for either pairs of single or double quotes.
- Subsets of strings can be taken using the slice operator ([ ] and [:]) with indexes starting at 0 in the beginning of the string and working their way from -1 at the end.
- The plus (+) sign is the string concatenation operator and the asterisk (*) is the repetition operator.

2.5.3 Python Lists:

- Lists are the most versatile of Python's compound data types.
- A list contains items separated by commas and enclosed within square brackets

([]).

- To some extent, lists are similar to arrays in C. One difference between them is that all the items belonging to a list can be of different data type.
- The values stored in a list can be accessed using the slice operator ([ ] and [:]) with indexes starting at 0 in the beginning of the list and working their way to end -1.
- The plus (+) sign is the list concatenation operator, and the asterisk (*) is the repetition operator.

2.5.4 Python Tuples:

- A tuple is another sequence data type that is similar to the list.
- A tuple consists of a number of values separated by commas. Unlike lists, however, tuples are enclosed within parentheses.
- The main differences between lists and tuples are: Lists are enclosed in brackets ([ ]) and their elements and size can be changed, while tuples are enclosed in parentheses (( )) and cannot be updated.
- Tuples can be thought of as read-only lists.
- For example − Tuples are fixed size in nature whereas lists are dynamic. In other words, a tuple is immutable whereas a list is mutable. You can't add elements to a tuple. Tuples have no append or extend method. You can't remove elements from a tuple. Tuples have no remove or pop method.

2.5.5 Python Dictionary:

- Python's dictionaries are kind of hash table type. They work like associative arrays or hashes found in Perl and consist of key-value pairs. A dictionary key can be almost any Python type, but are usually numbers or strings. Values, on the other hand, can be any arbitrary Python object.

- Dictionaries are enclosed by curly braces ({ }) and values can be assigned and accessed using square braces ([]).
- You can use numbers to "index" into a list, meaning you can use numbers to find out what's in lists. You should know this about lists by now, but make sure you understand that you can only use numbers to get items out of a list.
- What a dict does is let you use anything, not just numbers. Yes, a dict associates one thing to another, no matter what it is.

2.6 PYTHON FUNCTION:

2.6.1 Defining a Function:

You can define functions to provide the required functionality. Here are simple rules to define a function in Python. Function blocks begin with the keyword def followed by the function name and parentheses (i.e.()).

Any input parameters or arguments should be placed within these parentheses. You can also define parameters inside these parentheses

The code block within every function starts with a colon (:) and is indented. The statement returns [expression] exits a function, optionally passing back an expression to the caller. A return statement with no arguments is the same as return None.

2.6.2 Calling a Function:

Defining a function only gives it a name, specifies the parameters that are to be included in the function and structures the blocks of code. Once the basic structure of a function is finalized, you can execute it by calling it from another function or directly from the Python prompt.

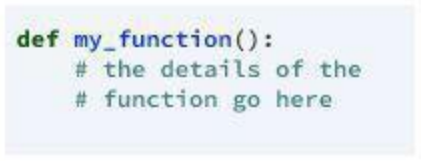
2.7 PYTHON USING OOP's CONCEPTS:

2.7.1 Class:

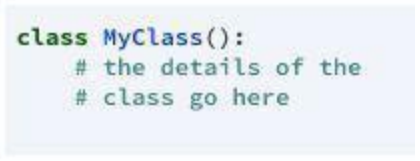
- Class: A user-defined prototype for an object that defines a set of attributes that characterize any object of the class. The attributes are data members (class variables and instance variables) and methods, accessed via dot notation.
- Class variable: A variable that is shared by all instances of a class. Class variables are defined within a class but outside any of the class's methods. Class variables are not used as frequently as instance variables are.
- Data member: A class variable or instance variable that holds data associated with a class and its objects.
- Instance variable: A variable that is defined inside a method and belongs only to the current instance of a class.

- Defining a Class:
    - We define a class in a very similar way how we define a function.
    - Just like a function, we use parentheses and a colon after the class name(i.e. ():) when we define a class. Similarly, the body of our class is

Indented like a functions body is.



```
def my_function():
    # the details of the
    # function go here
```



```
class MyClass():
    # the details of the
    # class go here
```

Figure 2.7.1.1: Defining a Class

2.7.2 __init__method in Class:

- The init method — also called a constructor — is a special method that runs when an instance is created so we can perform any tasks to set up the instance.
- The init method has a special name that starts and ends with two underscores:_init_().

CHAPTER 3

CASE STUDY

3.1 PROBLEM STATEMENT:

Fake news Classifier from the datasets as either as fake or real using Machine

Learning Algorithm:

1. LOGISTIC REGRESSION

3.2 DATA SET:

The given data set consists of the following parameters:

1. ID: unique id for a news article
2. TITLE: the title of a news article
3. AUTHOR: author of the news article
4. TEXT: the text of the article; could be incomplete
5. LABLE: a label that marks whether the news article is real or fake:

   1: Fake news

   0: real News

3.3 OBJECTIVE OF THE CASE STUDY:

To get a better understanding and chalking out a plan of action for solution of the client we used binomial Logistic regression, there can be only two possible types of the dependent variables, such as 0 or 1, Pass or Fail, Real or Fake. we have adapted the view point of looking at label categories and for further deep understanding of the problem, we have also removed column lable as it is just a number and here based on the column label we classified news as real or fake and taking input as author and title column. And output is real is fake or news is real. And displayed as if news is real it return to 1 else 0 if news is fake.

CHAPTER 4

MODEL BUILDING

4.1 DATA PREPROCESSING:

Pre-processing of the data actually involves the following steps:

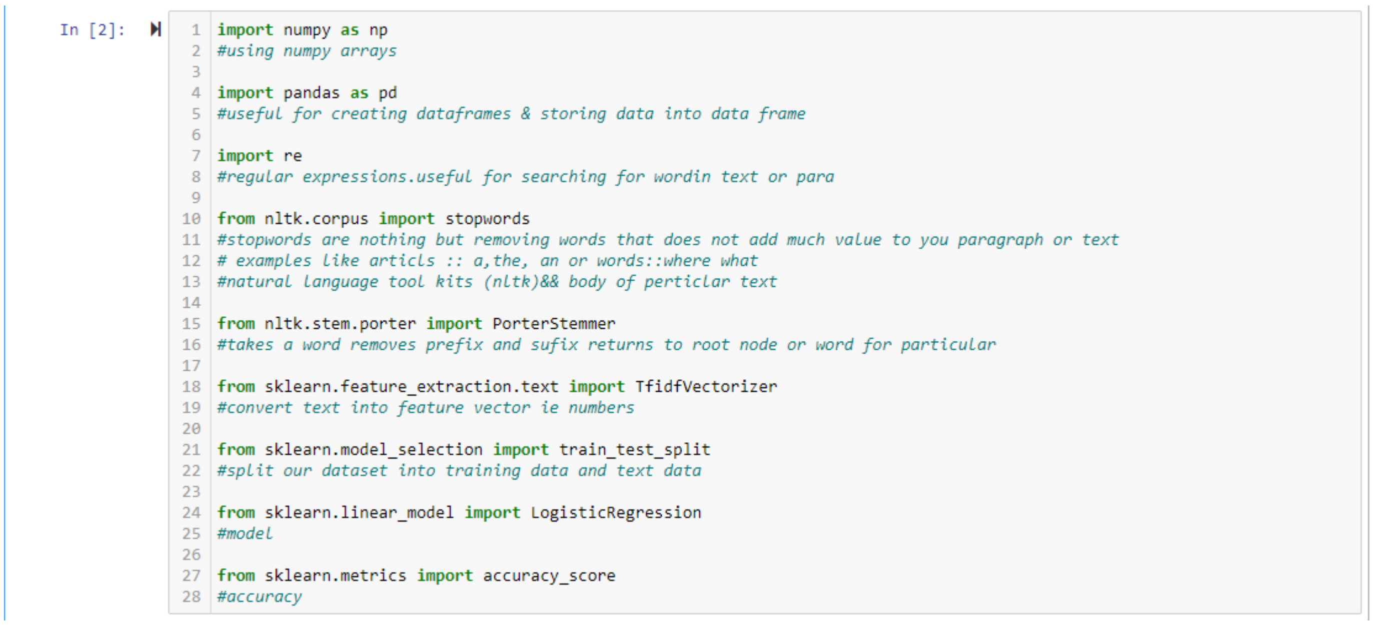
4.1.1 GETTING THE DATASET:

We can get the data set from database or we can get the data from client.

(Data Set used in the project is been downloaded from Kaggle website)

4.1.2 IMPORTING THE LIBRARIES:

We have to import the libraries as per the requirement of the algorithm.

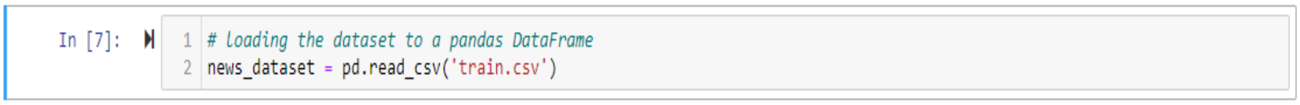


```
In [2]:
import numpy as np
#using numpy arrays

import pandas as pd
#useful for creating dataframes & storing data into data frame

import re
#regular expressions.useful for searching for wordin text or para

from nltk.corpus import stopwords
#stopwords are nothing but removing words that does not add much value to you paragraph or text
# examples like articls :: a,the, an or words::where what
#natural language tool kits (nltk)&& body of perticlar text

from nltk.stem.porter import PorterStemmer
#takes a word removes prefix and sufix returns to root node or word for particular

from sklearn.feature_extraction.text import TfidfVectorizer
#convert text into feature vector ie numbers

from sklearn.model_selection import train_test_split
#split our dataset into training data and text data

from sklearn.linear_model import LogisticRegression
#model

from sklearn.metrics import accuracy_score
#accuracy
```

Figure 4.1.2.1: Importing Libraries Dependencies.

4.1.3 IMPORTING THE DATA-SET:

Pandas in python provide an interesting method read_csv(). The read_csv function reads the entire dataset from a comma separated values file and we can assign it to a Data Frame to which all the operations can be performed. It helps us to access each and every row as well as columns and each and every value can be access using the data frame. Any missing value or NaNvalues have to be cleaned.

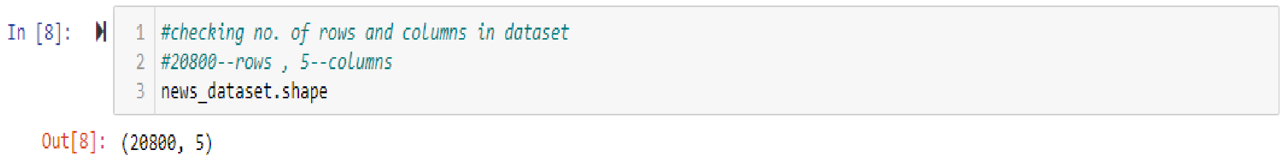


```
In [7]:  # loading the dataset to a pandas DataFrame
         news_dataset = pd.read_csv('train.csv')
```

Figure 4.1.3.1: Reading the Dataset

4.1.4 DATA SHAPE:

We use data shape to know how many rows and columns are present in our dataset

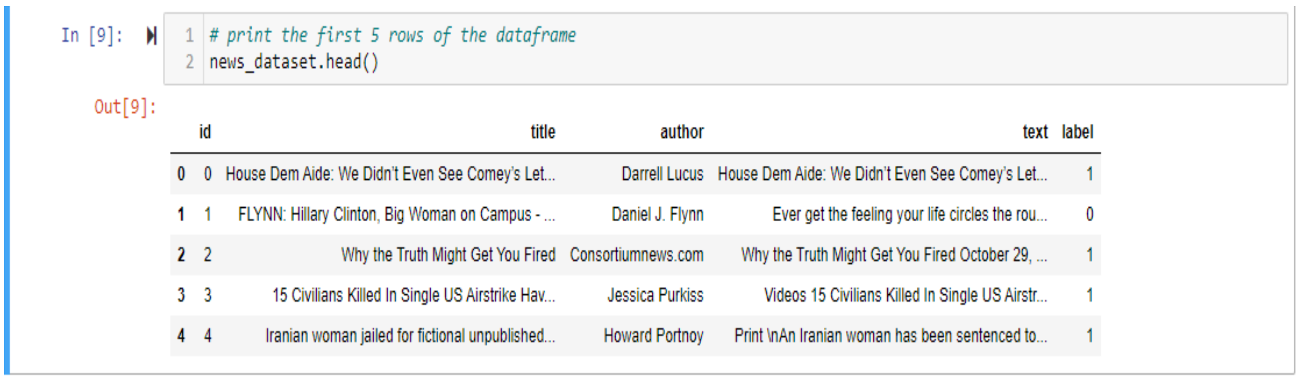


```
In [8]:  #checking no. of rows and columns in dataset
         #20800--rows , 5--columns
         news_dataset.shape

Out[8]:  (20800, 5)
```

Figure 4.1.4.1:Data shape

4.1.5 DATA SET:

Displaying 5 rows of dataset to make sure headers.



```
In [9]:  # print the first 5 rows of the dataframe
         news_dataset.head()
```

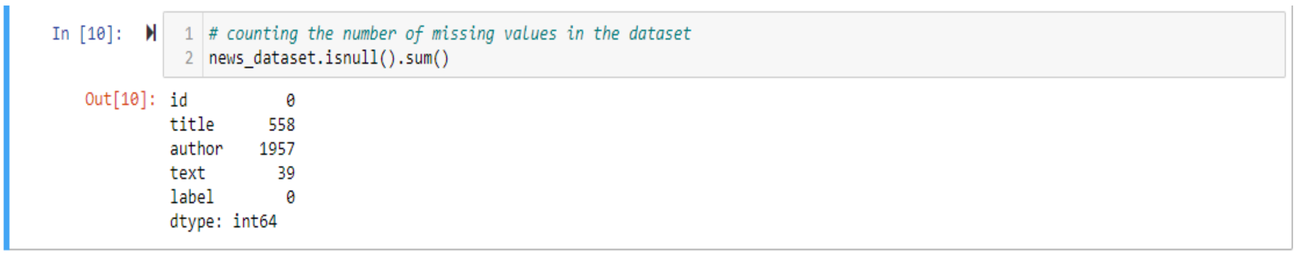
Out[9]:

	id	title	author	text	label
0	0	House Dem Aide: We Didn't Even See Comey's Let...	Darrell Lucus	House Dem Aide: We Didn't Even See Comey's Let...	1
1	1	FLYNN: Hillary Clinton, Big Woman on Campus - ...	Daniel J. Flynn	Ever get the feeling your life circles the rou...	0
2	2	Why the Truth Might Get You Fired	Consortiumnews.com	Why the Truth Might Get You Fired October 29, ...	1
3	3	15 Civilians Killed In Single US Airstrike Hav...	Jessica Purkiss	Videos 15 Civilians Killed In Single US Airstr...	1
4	4	Iranian woman jailed for fictional unpublished...	Howard Portnoy	Print \nAn Iranian woman has been sentenced to...	1

Figure 4.1.5.1:Data set

4.1.6 COUNT MISSING VALUES IN DATA SET:

We count number of missing values in data set so that no data can be miss placed in Dataset.



```
In [10]:  # counting the number of missing values in the dataset
          news_dataset.isnull().sum()

Out[10]:  id           0
          title      558
          author    1957
          text        39
          label        0
          dtype: int64
```

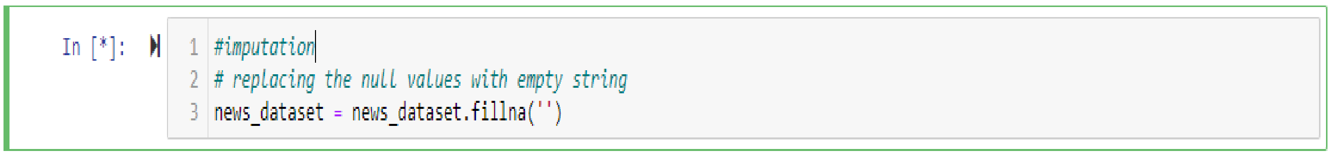
Figure 4.1.6.1: Count Missing values.

4.1.7 DATA IMPUTATION AND MERGING:

Imputation is another approach to resolve the problem of missing data.

The missing column values are substituted by another computed value. There might be scenarios where the dataset is small or where each row of the dataset represents a critical value.

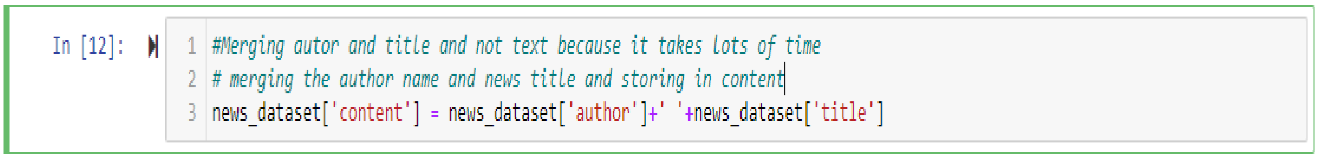
In those cases, we cannot remove the row from the dataset. The missing values can be imputed.



```
In [*]:  #imputation
         # replacing the null values with empty string
         news_dataset = news_dataset.fillna('')
```

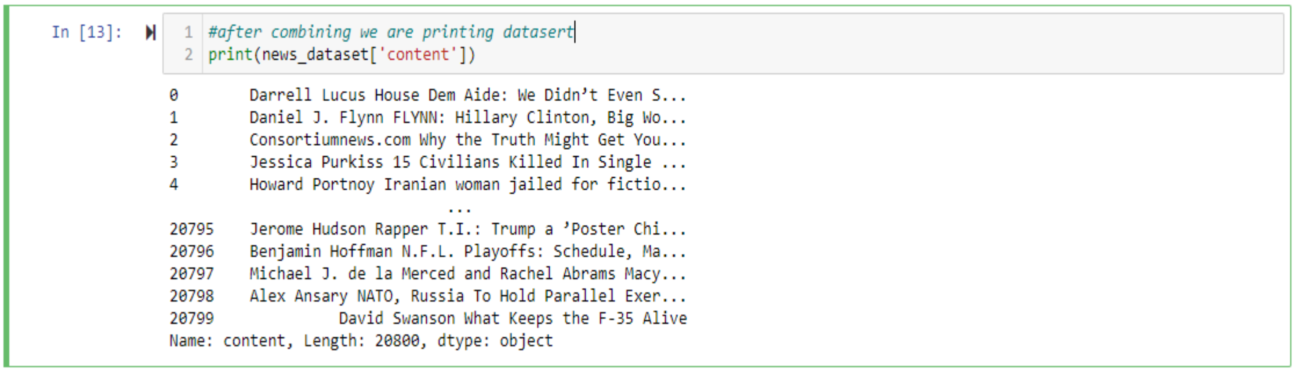
Figure 4.1.7.1: Replacing null values.

Merging the dataset of two column that is author and title and storing in content of new_dataset and not including text of dataset because it takes lot of time to predict news which is fake or real.



```
In [12]:  #Merging autor and title and not text because it takes lots of time
          # merging the author name and news title and storing in content
          news_dataset['content'] = news_dataset['author']+' '+news_dataset['title']
```

Figure 4.1.7.2: Merging dataset of two columns.



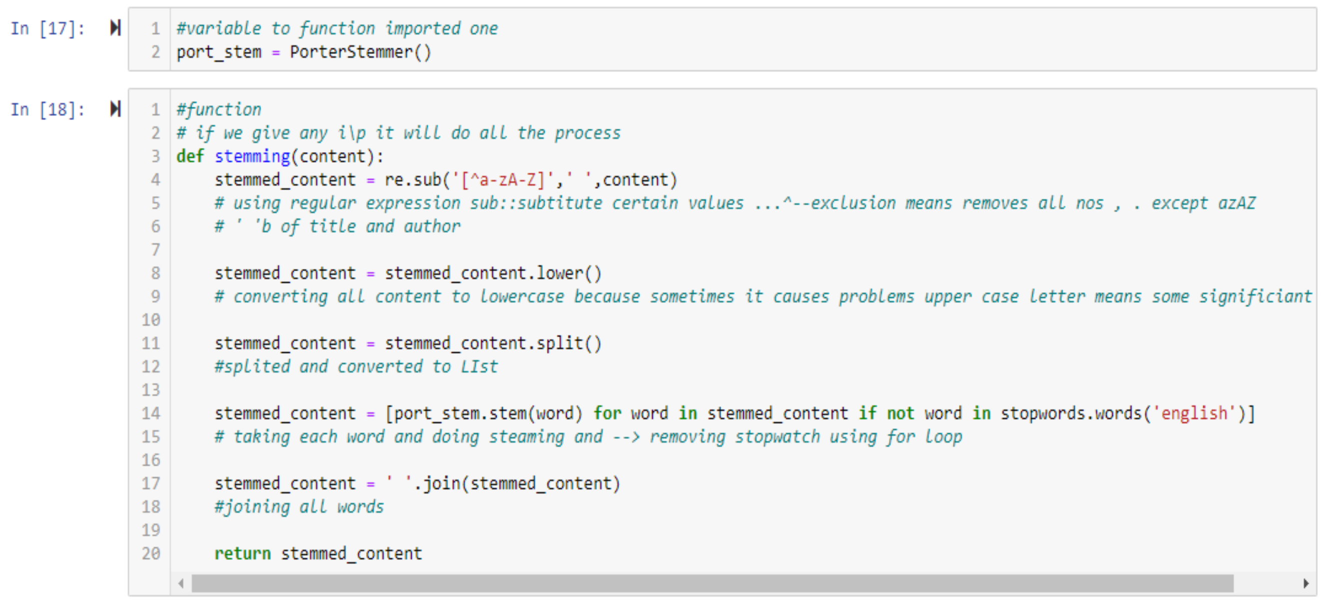
```
In [13]:  #after combining we are printing datasert
          print(news_dataset['content'])

0        Darrell Lucus House Dem Aide: We Didn’t Even S...
1        Daniel J. Flynn FLYNN: Hillary Clinton, Big Wo...
2        Consortiumnews.com Why the Truth Might Get You...
3        Jessica Purkiss 15 Civilians Killed In Single ...
4        Howard Portnoy Iranian woman jailed for fictio...
                               ...
20795    Jerome Hudson Rapper T.I.: Trump a ’Poster Chi...
20796    Benjamin Hoffman N.F.L. Playoffs: Schedule, Ma...
20797    Michael J. de la Merced and Rachel Abrams Macy...
20798    Alex Ansary NATO, Russia To Hold Parallel Exer...
20799              David Swanson What Keeps the F-35 Alive
Name: content, Length: 20800, dtype: object
```

Figure 4.1.7.3: Printing dataset after merging.

4.1.8 STEMMING:

Stemming is definitely the simpler of the two approaches. With stemming, words are reduced to their word stems. A word stem need not be the same root as a dictionary-based morphological root, it just is an equal to or smaller form of the word.

Stemming algorithms are typically rule-based. You can view them as heuristic process that sort-of lops off the ends of words. A word is looked at and run through a series of conditionals that determine how to cut it down.
For example, we may have a suffix rule that, based on a list of known suffixes, cuts them off. In the English language, we have suffixes like “-ed” and “-ing” which may be useful to cut off in order to map the words “cook,” “cooking,” and “cooked” all to the same stem of “cook.”

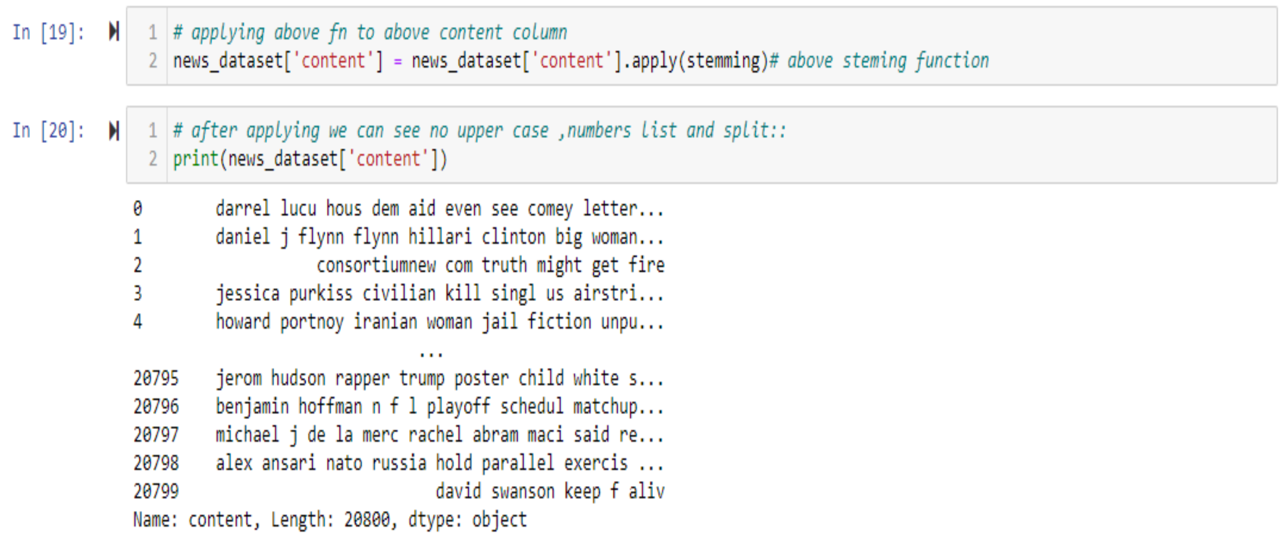


```
In [17]:
#variable to function imported one
port_stem = PorterStemmer()
```

```
In [18]:
#function
# if we give any i\p it will do all the process
def stemming(content):
    stemmed_content = re.sub('[^a-zA-Z]',' ',content)
    # using regular expression sub::subtitute certain values ...^--exclusion means removes all nos , . except azAZ
    # ' 'b of title and author

    stemmed_content = stemmed_content.lower()
    # converting all content to lowercase because sometimes it causes problems upper case letter means some significiant

    stemmed_content = stemmed_content.split()
    #splited and converted to LIst

    stemmed_content = [port_stem.stem(word) for word in stemmed_content if not word in stopwords.words('english')]
    # taking each word and doing steaming and --> removing stopwatch using for loop

    stemmed_content = ' '.join(stemmed_content)
    #joining all words

    return stemmed_content
```

Figure 4.1.8.1: Stemming Process.

4.1.9 AFTER STEMMING PRINTING DATASET:



```
In [19]:
# applying above fn to above content column
news_dataset['content'] = news_dataset['content'].apply(stemming)# above steming function
```

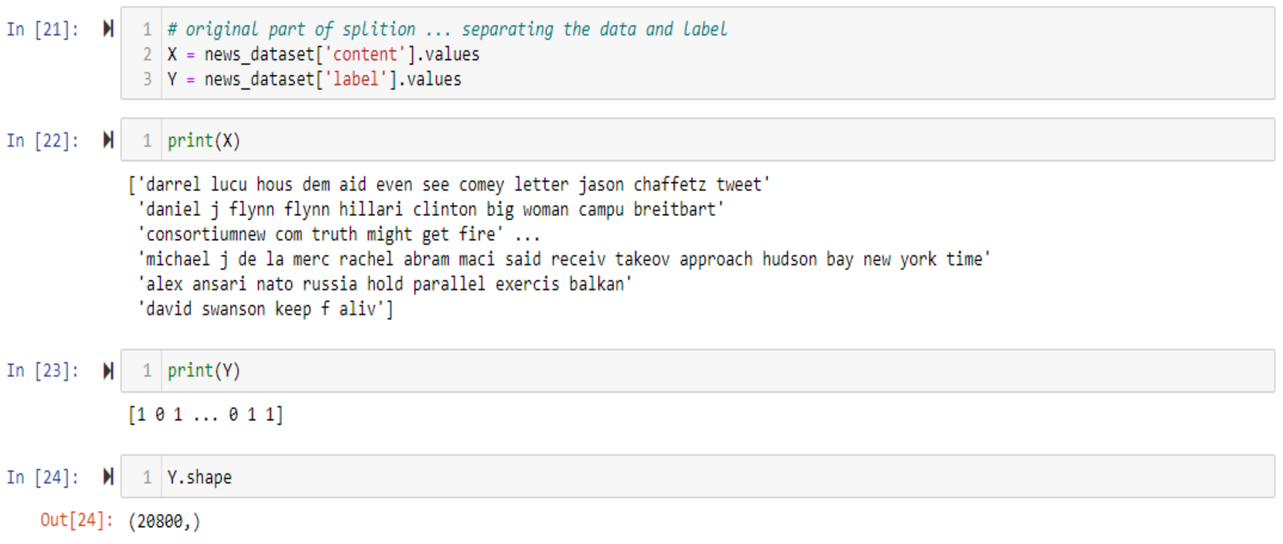
```
In [20]:
# after applying we can see no upper case ,numbers list and split::
print(news_dataset['content'])
```

```
0        darrel lucu hous dem aid even see comey letter...
1        daniel j flynn flynn hillari clinton big woman...
2                   consortiumnew com truth might get fire
3        jessica purkiss civilian kill singl us airstri...
4        howard portnoy iranian woman jail fiction unpu...
                               ...
20795    jerom hudson rapper trump poster child white s...
20796    benjamin hoffman n f l playoff schedul matchup...
20797    michael j de la merc rachel abram maci said re...
20798    alex ansari nato russia hold parallel exercis ...
20799                             david swanson keep f aliv
Name: content, Length: 20800, dtype: object
```

Figure 4.1.9.1: Printing Dataset after stemming.

4.1.10 SPLITTING THE DATA OF COLUMNS:

Separating author, title and lable storing in different variable that is X contains “contents” and Y contains “lables”.Printing the values of X and Y as follows:



```
In [21]:
# original part of splition ... separating the data and label
X = news_dataset['content'].values
Y = news_dataset['label'].values
```

```
In [22]:
print(X)
```

```
['darrel lucu hous dem aid even see comey letter jason chaffetz tweet'
 'daniel j flynn flynn hillari clinton big woman campu breitbart'
 'consortiumnew com truth might get fire' ...
 'michael j de la merc rachel abram maci said receiv takeov approach hudson bay new york time'
 'alex ansari nato russia hold parallel exercis balkan'
 'david swanson keep f aliv']
```

```
In [23]:
print(Y)
```

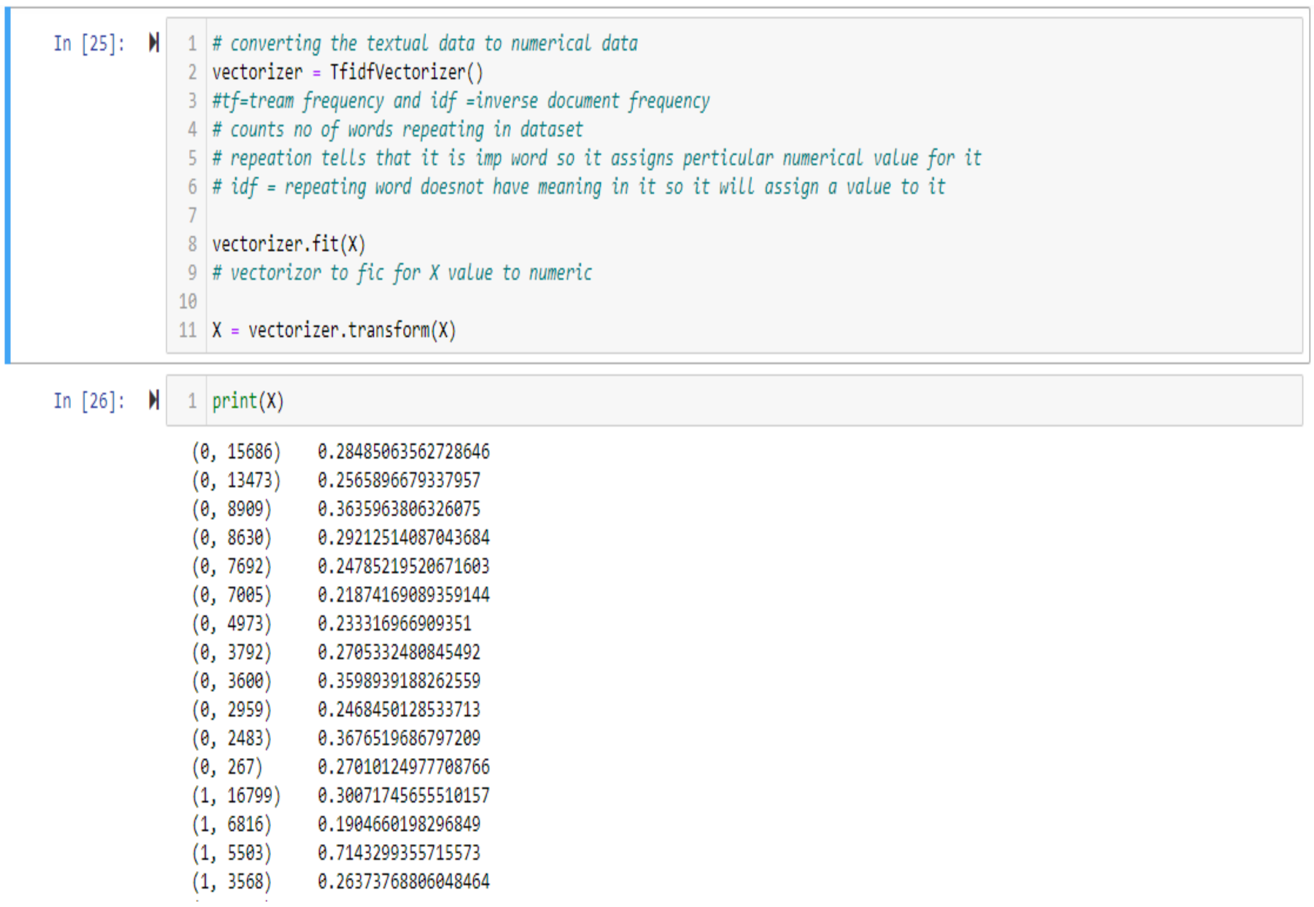
```
[1 0 1 ... 0 1 1]
```

```
In [24]:
Y.shape
Out[24]: (20800,)
```

Figure 4.1.10.1: Splitting data.

4.1.11 VECTORIZATION:

Vectorization is used to change text to numerical data. Using a function instead can help in minimizing the running time and execution time of code efficiently. Various operations are being performed over vector instead of arrays such as dot product of vectors which is also known as scalar product as it produces single output, outer products which results in square matrix of dimension equal to of the vectors, Element wise multiplication which products the element of same indexes and dimension of the matrix remain unchanged.



```
In [25]:
# converting the textual data to numerical data
vectorizer = TfidfVectorizer()
#tf=tream frequency and idf =inverse document frequency
# counts no of words repeating in dataset
# repeation tells that it is imp word so it assigns perticular numerical value for it
# idf = repeating word doesnot have meaning in it so it will assign a value to it

vectorizer.fit(X)
# vectorizor to fic for X value to numeric

X = vectorizer.transform(X)
```

```
In [26]:
print(X)
```

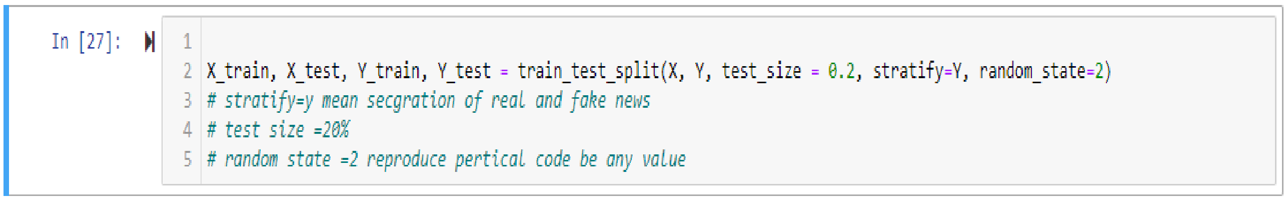
```
  (0, 15686)	0.28485063562728646
  (0, 13473)	0.2565896679337957
  (0, 8909)	0.3635963806326075
  (0, 8630)	0.29212514087043684
  (0, 7692)	0.24785219520671603
  (0, 7005)	0.21874169089359144
  (0, 4973)	0.233316966909351
  (0, 3792)	0.2705332480845492
  (0, 3600)	0.3598939188262559
  (0, 2959)	0.2468450128533713
  (0, 2483)	0.3676519686797209
  (0, 267)	0.27010124977708766
  (1, 16799)	0.30071745655510157
  (1, 6816)	0.1904660198296849
  (1, 5503)	0.7143299355715573
  (1, 3568)	0.26373768806048464
```

Figure 4.1.11.1: Converting text to numerical dataset.

4.1.12 SPLITTING THE DATASET TO TRAINING & TEST DATA:

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 additional configuration is required, such as when it is used for classification and the dataset is not balanced.



```
In [27]:

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.2, stratify=Y, random_state=2)
# stratify=y mean secgration of real and fake news
# test size =20%
# random state =2 reproduce pertical code be any value
```

Figure 4.1.12.1: Training and Test data.

4.2 ALGORITHMS USED:

The technique used is Logistic regression to solve the problem statement.

4.2.2 LOGISTIC REGRESSION:

- Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.
- Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, **it gives the probabilistic values which lie between 0 and 1**.
- Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas **Logistic regression is used for solving the classification problems**.
- In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).
- The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc.
- 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. The below image is showing the logistic function:

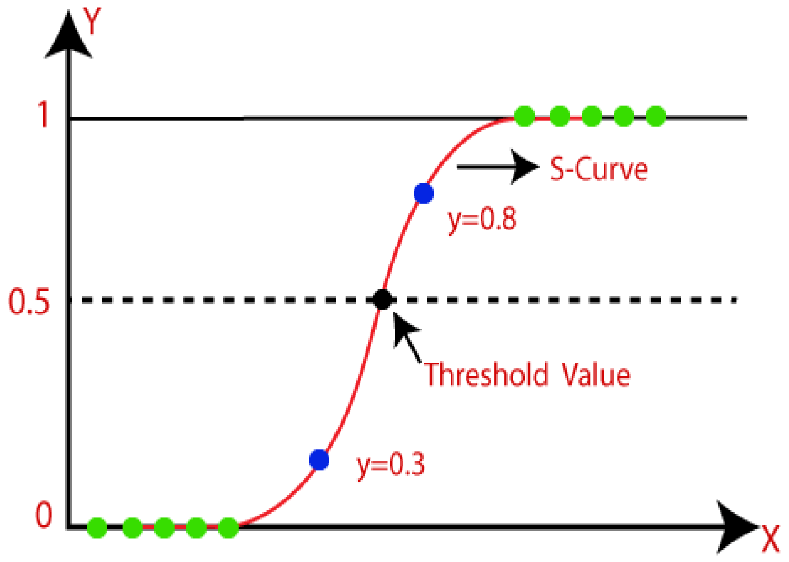


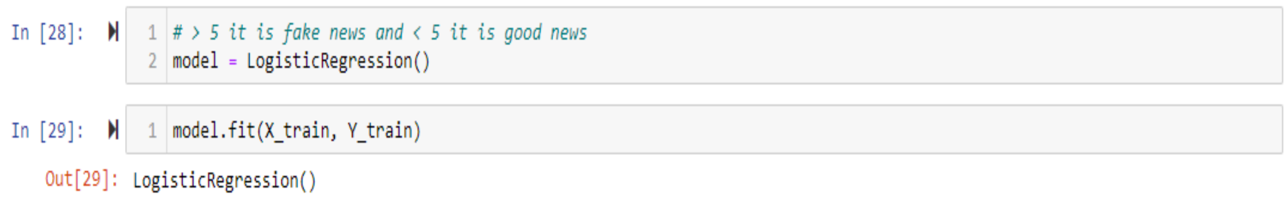
Figure 4.3.2.2: Logistic Regression Graph.

4.3 LOGISTIC REGRESSION ALGORITHM:

4.3.1 Logistic Function :

- The sigmoid function is a mathematical function used to map the predicted values to probabilities.
- It maps any real value into another value within a range of 0 and 1.
- The value of the logistic regression must be between 0 and 1, which cannot go beyond this limit, so it forms a curve like the "S" form. The S-form curve is called the Sigmoid function or the logistic function.
- In logistic regression, we use the concept of the threshold value, which defines the probability of either 0 or 1. Such as values above the threshold value tends to 1, and a value below the threshold values tends to 0.

4.3.2 Applying the algorithm:



```
In [28]:  # > 5 it is fake news and < 5 it is good news
          model = LogisticRegression()

In [29]:  model.fit(X_train, Y_train)

Out[29]: LogisticRegression()
```

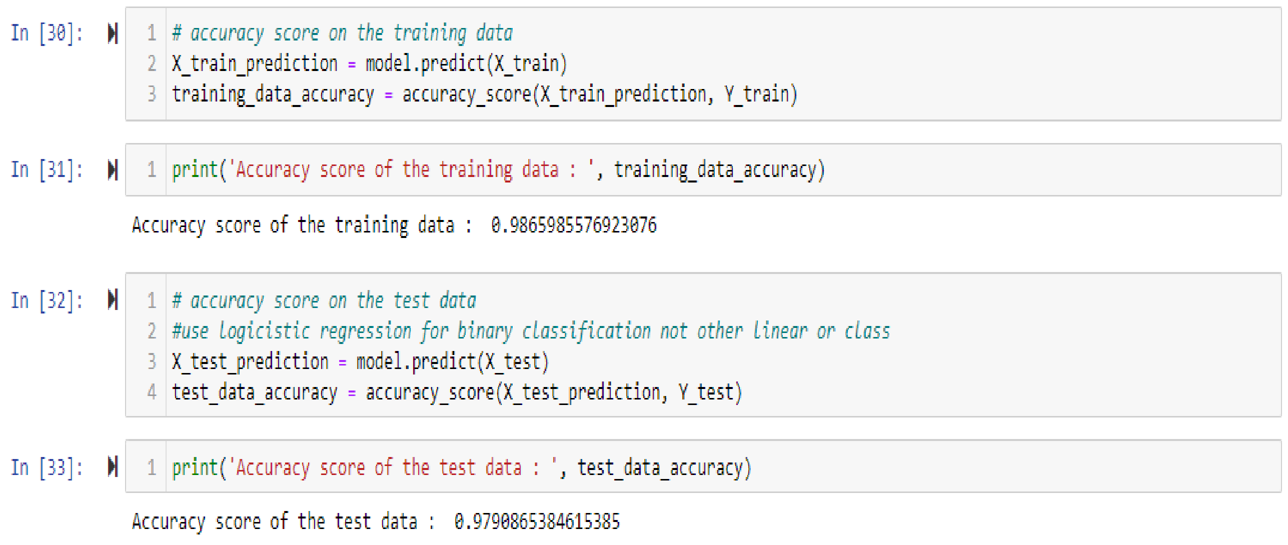
Figure 4.3.2.1: Applying the algorithm.

4.3.3 Accuracy:

Classification Accuracy is what we usually mean, when we use the term accuracy. It is the ratio of number of correct predictions to the total number of input samples.

$$Accuracy = \frac{Number\ of\ Correct\ predictions}{Total\ number\ of\ predictions\ made}$$

It works well only if there are equal number of samples belonging to each class.

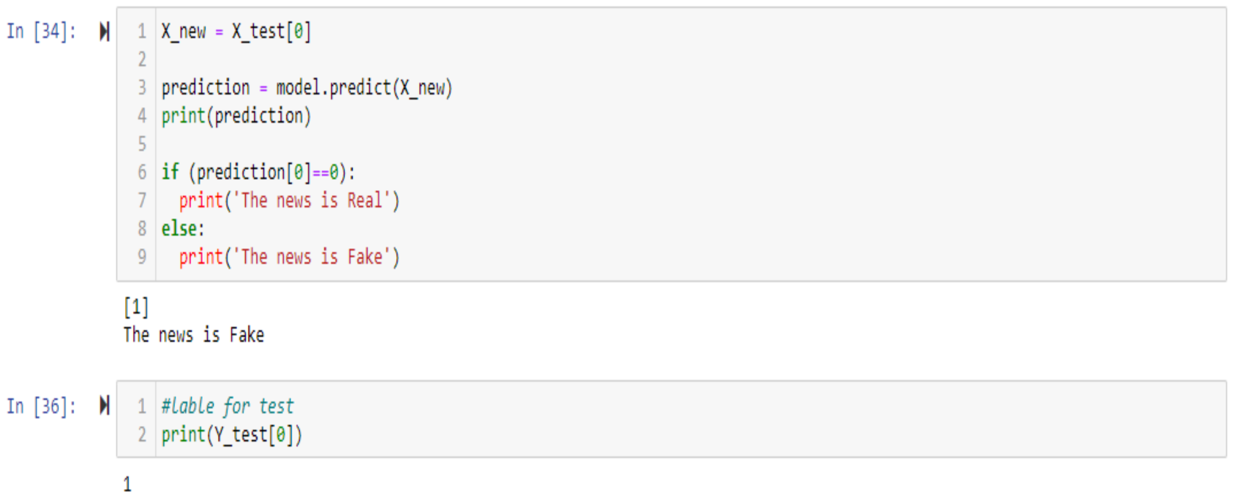


```
In [30]:  # accuracy score on the training data
          X_train_prediction = model.predict(X_train)
          training_data_accuracy = accuracy_score(X_train_prediction, Y_train)

In [31]:  print('Accuracy score of the training data : ', training_data_accuracy)

          Accuracy score of the training data :  0.9865985576923076

In [32]:  # accuracy score on the test data
          #use logicistic regression for binary classification not other linear or class
          X_test_prediction = model.predict(X_test)
          test_data_accuracy = accuracy_score(X_test_prediction, Y_test)

In [33]:  print('Accuracy score of the test data : ', test_data_accuracy)

          Accuracy score of the test data :  0.9790865384615385
```

Figure 4.3.3.1: Accuracy Score.

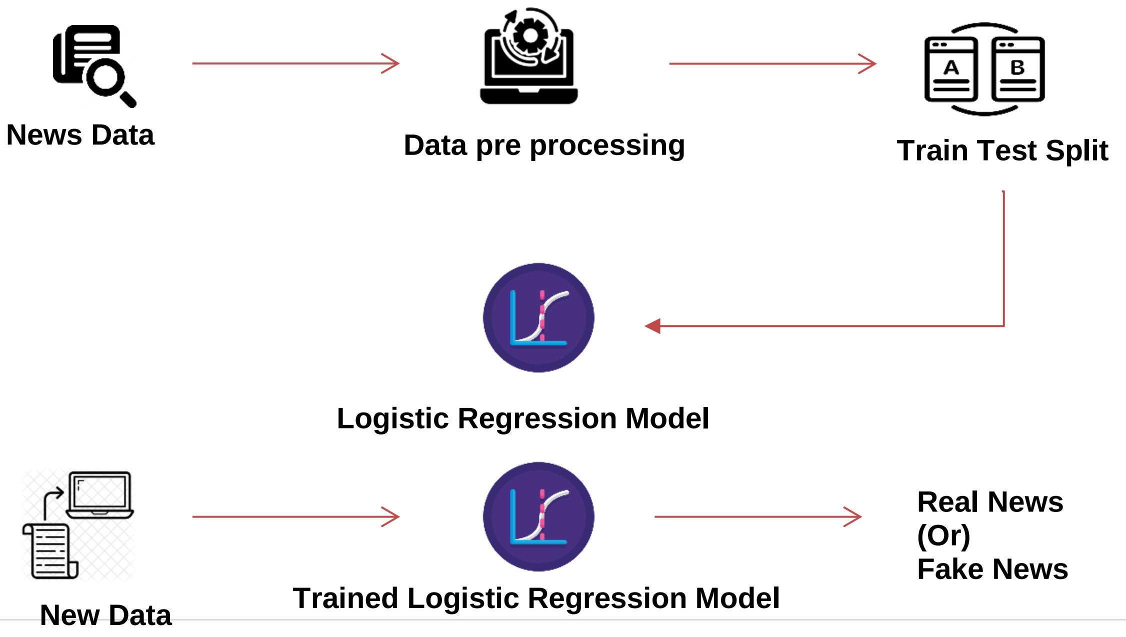
4.3.4 Predictive System:

As the name suggests, 'Predictive' means to predict something, so predictive analytics is the analysis done to predict the future event using the previous data. It is the process of extracting information from existing sets of data to find useful information, trends and forecast future events. Predictive analytics does not tell the exact thing that will happen in the future. It predicts what might happen in the future.



```
In [34]:  X_new = X_test[0]

          prediction = model.predict(X_new)
          print(prediction)

          if (prediction[0]==0):
            print('The news is Real')
          else:
            print('The news is Fake')

          [1]
          The news is Fake

In [36]:  #lable for test
          print(Y_test[0])

          1
```

Figure 4.3.4.1: Final Execution.



CONCLUSION:

The task of classifying news manually requires in-depth knowledge of the domain and expertise to identify anomalies in the text. In this research, we discussed the problem of classifying fake news articles using machine learning models and ensemble techniques. The data we used in our work is collected from the KAGGLE and contains news articles from various domains to cover most of the news rather than specifically classifying political news. The primary aim of the research is to identify patterns in text that differentiate fake articles from true news.

Fake news detection has many open issues that require attention of researchers. For instance, in order to reduce the spread of fake news, identifying key elements involved in the spread of news is an important step. Real time fake news identification in videos can be another possible future direction.

Finally, this application is only one that would be necessary in a larger toolbox that could function as a highly accurate fake news classifier. Other tools that would need to be built may include a fact detector and a stance detector. In order to combine all of these "routines," there would need to be some type of model that combines all of the tools and learns how to weight each of them in its final decision

REFERENCES:

[1] https://www.kaggle.com/c/fake-news/data?select=train.csvhttps://open-platform.theguardian.com/

[2] https://developer.nytimes.com/

[3] www.codingninjas.com

[4] www.youtube.com/semicolon

[5] RESEARCH PAPER 1-Automatic detection of fake news- Ver´onica P´erezRosas1, Bennett Kleinberg2, Alexandra Lefevre1 Rada Mihalcea1

[6] RESEARCH PAPER 2- GloVe: Global Vectors for Word Representation Jeffrey Pennington, Richard Socher, Christopher D. Manning Computer Science Department, Stanford University, Stanford, CA 94305. This paper was about GloVe . It is a type of word embedding which is very helpful in text classification problems. It was recently developed by Stanford university