

A Project Report

on

Fake News Detection Using Machine Learning

*Submitted in partial fulfilment of the
requirement for the award of the degree of*

Bachelor of Technology in Computer Science and
Engineering



(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

**Under The
Supervision of**

**Dr. Ajay Shanker
Singh
Associate Professor**

Submitted By

18SCSE1010232 – Devesh Kumar

18SCSE1010230 – Anuraj Koli

**SCHOOL OF COMPUTING SCIENCE AND ENGINEERING DEPARTMENT
OF COMPUTER SCIENCE AND ENGINEERING GALGOTIAS
UNIVERSITY, GREATER NOIDA**

DECEMBER-2021



**SCHOOL OF COMPUTING SCIENCE AND
ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA**

CANDIDATE'S DECLARATION

We hereby certify that the work which is being presented in the project, entitled **“Fake News Detection using Machine Learning”** in partial fulfillment of the requirements for the award of the Bachelor of Technology submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of September 2021 to December 2021, under the supervision of Dr. Ajay Shanker Singh (Associate Professor), School of Computing Science and Engineering, Galgotias University, Greater Noida

The matter presented in the project has not been submitted by me/us for the award of any other degree of this or any other places.

Devesh Kumar, 18SCSE1010232

Anuraj Koli, 18SCSE1010230

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Dr. Ajay Shanker Singh

Associate Professor

CERTIFICATE

The Final Project Viva-Voce examination of Devesh Kumar(18SCSE1010232), Anuraj Koli(18SCSE1010230) has been held on_____and his work is recommended for the award of B.Tech.

Signature of Examiner(s)

Signature of Supervisor(s)

Signature of Project Coordinator

Signature of Dean

Date: December, 2021

Place: Greater Noida

Abstract

The advent of the World Wide Web and the rapid adoption of social media platforms (such as Facebook and Twitter) have opened the way for the dissemination of information that has never been seen before in human history. With the current use of social media, consumers are creating and sharing more information than ever before, some misleading unrelated to reality. Automatic classification of text such as false information or disinformation is a challenging task. Even an expert in a particular field should consider many factors before making a decision on the validity of an article. In this work, we propose to use a combination of machine learning tools in the default categories of news topics. Our study examines various text structures that can be used to distinguish counterfeit content from reality. Using those structures, we train a combination of different machine learning processes using different methods of integrating and evaluating their performance in 4 real-world data sets. Assessment tests ensure the maximum effectiveness of our proposed student integration approach compared to individual students.

TABLE OF CONTENTS

INTRODUCTION

Introduction

Aim and Objectives

Motivation

Scope of Project

LITERATURE SURVEY

Introduction

Existing System

Need of New System

Problem Definition

DESIGN AND IMPLEMENTATION

Proposed System

Design

System Design Diagram

Methodology used

RELATED WORK

Spam Detection

Stance Detection

Benchmark Dataset

Datasets

Document Level

IMPORTING DEPENDENCIES

DATA PRE-PROCESSING

STEMMING

SPLITTING THE DATASET

TRAINING THE DATASET

MAKING A PREDICTIVE SYSTEM

CONCLUSION

Summary

Future Scope

REFERENCES

Reference

Introduction:

The rise of false news during the U.S. Presidential Election The 2016 edition not only highlights the dangers of false news outcomes but also the challenges posed by the attempt to distinguish untrue stories. Fake stories may be a new name in comparison but it is not something new. Counterfeit stories have been around at least since the emergence and popularity of individual newspapers, in the 19th century. However, advances in technology and the distribution of news by various media outlets have increased the prevalence of non-modern news. As such, the effects of misinformation have increased dramatically in the past and something must be done to prevent this from happening in the future.

We found three very common motives for fictional news and chose only one as a target for this project as a way to reduce search in a meaningful way. The first incentive for the writing of false stories, which goes back to the newspapers of the 19th-century one-party faction, is to influence public opinion. Second, requiring the latest technological advances, is the use of counterfeit themes like clickbait to make money. The third reason for writing non-fiction, equally prominent but not so dangerous, is comedy. While all three subsets of non-fiction, namely, (1) clickbait, (2), influential, and (3) satire, share the same thread of falsehood, their broad outcomes are very different. Therefore, this paper will focus mainly on false stories as defined, "intentional content that deliberately impersonates as the discovery of real-world news." This definition does not include sarcasm, which is intended to be ridiculous and not deceptive to readers. Therefore, our goal is to move beyond these achievements and use machine learning to classify, at least as well as humans, more difficult discrepancies between real and fake news. There are two ways machines can solve the problem of false news better than humans. The first is that machines are better at finding and tracking statistics than humans. In addition, machines can be very effective in testing the database for all relevant articles and responses based on a wide variety of sources. Any of these methods can be helpful in finding false information, but we have decided to focus on how the machine can solve the problem of non-existent news using supervised reading that removes the language and content element only from the source in question. With so many counterfeit information detection tactics, a "fake" article published by a trusted author from a reliable source will not be caught. This approach will combat these "false" sections of false news. In short, this work can be the same as a person's face when reading a portable copy of a newspaper article, without access to the internet or external information of the story (compared to reading something online where they can simply look for relevant sources). The machine, like someone in a coffee shop, will have access to the words in this article only and must use techniques that do not rely on the list of authors' names and sources. The current project involves the use of machine learning and natural language processing methods to create a model that can reveal documents, with great opportunities, false news articles.

Many of the current automated methods of this problem focus on the "blacklist" of authors and sources who are well-known fake news producers. However, what if the author is anonymous or if a false story is published by a reputable source? In these cases, it is necessary to easily rely on the content of the article to decide whether it is false or not. By collecting examples of both real and false stories and model training, it should be possible to distinguish non-fiction stories with a certain degree of accuracy.

The aim of this project is to determine the effectiveness and limitations of language-based techniques for obtaining non-language-based learning algorithms including but not limited to convolutional neural and recurrent neural networks. The outcome of this project should determine the extent to which this work can be achieved by analysing the patterns contained in this document and not seeing external information about the land.

This type of solution is not intended to be the ultimate solution for the separation of false issues. Like the "black" methods mentioned, there are situations where it fails and some succeed. Instead of being an end-to-end solution, this project is intended to be a single tool that can be used to help people who are trying to separate false stories. Alternatively, it could be a single tool used in future applications that cleverly combines multiple tools to create an end-to-end solution for making the wrong editing process.

Aim and Objectives:

The main objective behind the development and upgradation of existing projects are the following smart approaches:

- Be Aware of such article while forwarding to others
- Reveal True stories
- Prevent from false crisis events
- Be Informative

Motivation:

Machine learning (ML) is a type of artificial intelligence (AI) that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so. Machine learning algorithms use historical data as input to predict new output values. The extensive spread of faux news can have a significant negative impact on individuals and society. First, fake news can shatter the authenticity equilibrium of the news ecosystem for instance. Understanding the truth of new and message with news detection can create positive impact on the society.

Scope of Project:

The usage of this system greatly reduces the time required to search for a place leading to quicker decision making with respect to places to visit. Used to view the location view (the user can even zoom in and zoom out to get a better view) as well as 360-degree image embedded in the application. The System makes use of weather underground API for fetching the details of weather at accuracy.

The user can also find the paths to follow to reach the final destination in map which gives a better view to the users. It becomes convenient for users to book their tour via website instead of visiting agency ultimately saves time and money.

LITERATURE SURVEY

Introduction

Our project is a web application which gives you the guidance of the day-to-day routine of fake news, spam message in daily news channel, Facebook, Twitter, Instagram and other social media. We have shown some data analysis from our dataset which have retrieve from many online social media and display the main source till now fake news and true news are engaged.

Our project is tangled with multiple models trained by our own and also some pretrained model extracted from Felipe Adachi. The accuracy of the model is around 95% for all the self-made model and 97% for this pretrained model. This model can detect all news and message which are related to covid-19, political news, geology, etc.

Existing System

We can get online news from different sources like social media websites, search engine, homepage of news agency websites or the factchecking websites. On the Internet, there are a few publicly available datasets for Fake news classification like BuzzFeed News, LIAR [15], BS Detector etc. These datasets have been widely used in different research papers for determining the veracity of news. In the following sections, I have discussed in brief about the sources of the dataset used in this work. This Existing system can help us to trained our model using machine learning technique.

Need of New System

Currently, many people are using the internet as a central platform to find the information about reality in world and need to be continue. Hence, we have mention above we will create fake news and message detection model which detect the reality of the news and message. Also, whose use our website can see the up to date about main source or keyword are getting most fake news and message and mapped up with chart. After and all everyone want to know how to prevent this hence we are giving some important tips to avoid this fake news of spreading rumour in the world.

Problems Definition

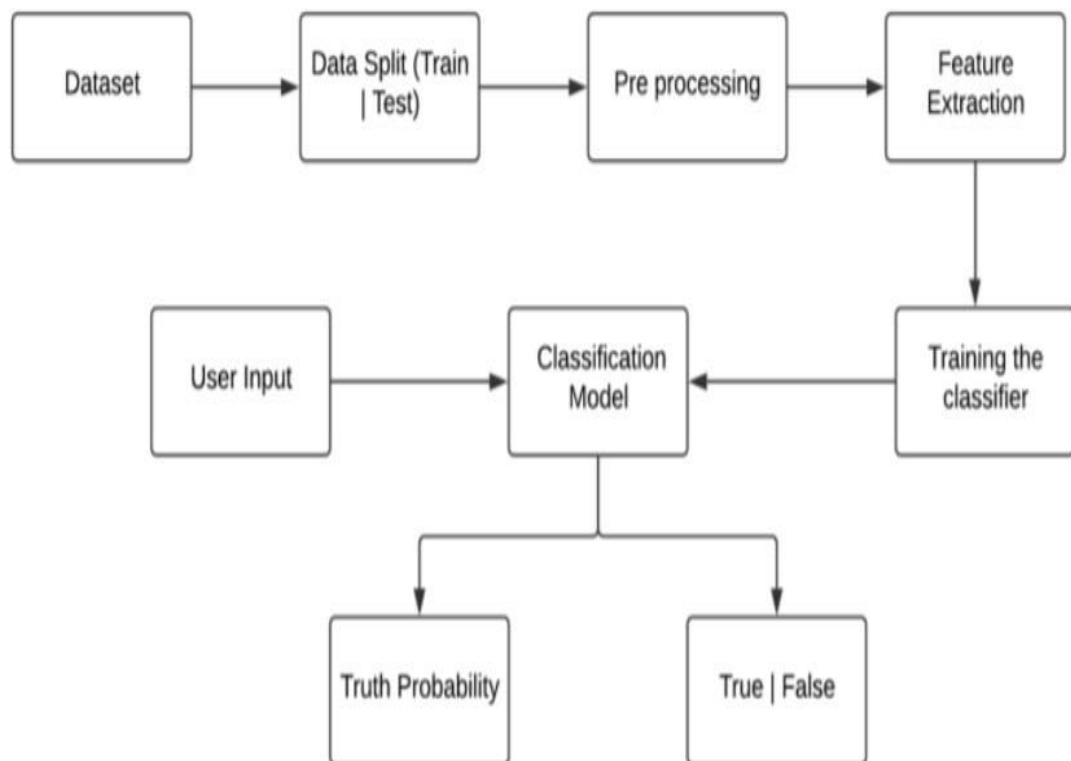
The system is a Web application which help user to detect the fake news. We have given the text box where the user has the option to paste the message or paste the url link of the news and other message link and after that it gives the reality of it. All the user gives data to detector may save for further use in order to update the statue of model, data analysis in future. We also help user by giving some guidance of how to prevent from such false event and how to stop with such event from spreading it.

DESIGN AND IMPLEMENTATION

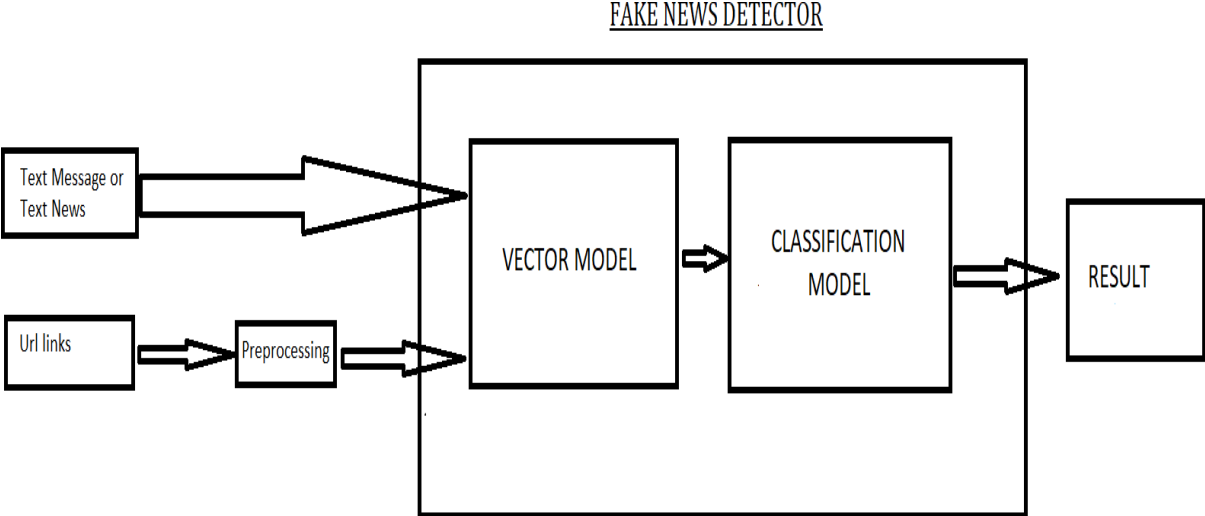
Proposed system

The system is a Web application which help user to detect the fake news. We have given the text box where the user has the option to paste the message or paste the URL link of the news and other message link and after that it gives the reality of it. All the user gives data to detector may save for further use in order to update the statue of model, data analysis in future. We also help user by giving some guidance of how to prevent from such false event and how to stop with such event from spreading it.

Design



System Design Diagram



Related Work

Spam Detection

The problem of finding untrue sources of content based on content analysis is considered to be solved at least in the domain of spam detection, spam detection using reading calculator techniques to classify text as spam or legal. These methods include pre-text processing, feature extraction (e.g., word bag), and feature selection based on which factors lead to better performance in the test database. Once these features have been identified, they can be categorized using Nave Bayes, Support Vector Machines, TF-IDF, or neighbouring K filters. All of these separators are a feature of supervised machine learning, which means they need labelled data in order to learn the function.

The task of finding false news is the same and almost the same as the task of finding spam because both aim to distinguish official text examples from examples of illegal and malicious text. The question is, how can we use the same methods to obtain incorrect information? Instead of filtering as we do with spam, it may be helpful to be able to mark fake news articles so that readers are warned that what they are reading may be false news. The purpose of this project is not to determine if the text is inaccurate or not, but rather to let them know that they need to use additional scrutiny of other texts. Fake news detection, unlike spam detection, has many nuances that are not easily detected by text analysis. For example, a person actually needs to use their knowledge of a particular subject to determine if the stories are true or not. The "fakeness" of an article can be opened or closed by simply inserting someone else's name and inserting someone else's name. Therefore, the best we can do from a content-based perspective is to determine if it is something that needs to be tested. The idea may be for the reader to do a leg job of researching other articles on the topic to determine if the article is false or not, but "marking" would allow them to do so in appropriate circumstances.

Stance Detection

In December 2016, a group of volunteers from industry and education launched a competition called the Fake News Challenge. The aim of the competition was to promote the development of tools that can help humanities explorers to obtain unintentional information in the media through study equipment, natural language processing, and artificial intelligence. The editors have decided that the first step toward this great goal was to understand what other media organizations have to say about the topic in question. Therefore, they decided that one stage of their competition would be a competition to see the stand. Specifically, promoters create a database of news headlines and text bodies and challenge competitors to create well-designed dividers, which are related to a given topic, into one of four categories: "agree", "disagree", "Chat" or "unrelated." The top three teams achieved more than 80% accuracy in the test set for this task. The high-group model was based on a measured ratio between gradient-enhanced decision trees and a deep convolutional neural network.

Benchmark Dataset

This demonstrates previous work on fake news detection that is more directly related to our goal of using a text-only approach to make a classification. We did not only create a new benchmark dataset of statements, but also show that significant improvements can be made in fine-grained fake news detection by using meta-data (i.e., speaker, party, etc.) to augment the information provided by the text.

Datasets

The lack of stocks of false news data is certainly a stumbling block to advancing computer-based, text-based models covering a wide variety of topics. The database of false stories is not in line with our purpose because it contains world truth about the relationship between texts but not whether those texts are true or false statements or not. For our purpose, we need a collection of news articles divided directly into categories of genres (e.g., real vs. fake or real vs parody vs. clickbait vs. propaganda). With simple and standard NLP segmentation tasks, such as mood analysis, there are a number of labelled data from a variety of sources including Twitter, Amazon Review, and IMDb reviews. Unfortunately, the same is not true with labelled fiction stories. This poses a challenge for researchers and data scientists who want to explore the topic through supervised electronic learning methods. I researched the available data sets for sentence-level classification and methods for combining data sets to create complete sets with good and bad examples of document-level planning.

We produced a new benchmark dataset for fake news detection that includes 12,800 manually categorized quick statements on a selection of topics. these statements come from politifact.com, which gives heavy analysis of and links to the supply files for every of the statements. The labels for these records aren't genuine and false however as an alternative replicate the “sliding scale” of fake news and have 6 durations of labels. those labels, in order of ascending truthfulness, include 'pants-fire', 'fake', slightly real, 'half-actual', 'broadly speaking-actual', and proper. The creators of this database ran baselines together with Logistic Regression, aid Vector Machines, LSTM, CNN and an augmented CNN that used metadata. They reached 27% accuracy on this multiclass classification venture with the CNN that concerned metadata which include speaker and party related to the textual content.

Document Level

We've got accumulated information in such a way that we're more cautious that we've manipulate for greater bias in the sources and subjects. because the intention of our challenge became to locate patterns inside the language which are indicative of real or fake news, having source bias might be detrimental to our cause. consisting of any source bias in our dataset, i.e., patterns which can be unique to NYT, the guardian, or any of the fake information web sites, could allow the version to discover ways to companion assets with actual/fake news labels. getting to know to classify resources as faux or real information is an easy problem, but studying to categorize precise varieties of language and language patterns as faux or real information isn't always. As such, we have been very careful to put off as a lot of the supply-precise styles as possible to pressure our version to learn something more meaningful and generalizable.

We admit that there are certainly instances of fake news in the New York Times and probably instances of real news in the Kaggle dataset because it is based on a list of unreliable websites but, due to the fact these instances are the exception and now not the rule of thumb, we count on that the version will study from the majority of articles which might be consistent with the label of the source. additionally, we are not seeking to train a model to research data however as a substitute study delivery. To be clearer, the deliveries and reporting mechanisms found in faux news articles inside New York Times should still possess characteristics greater normally observed in real information, although they will comprise fictitious actual information.

Importing Dependencies

```
Desktop/Fake News Prediction/ x Fake News Prediction - Jupyter | x +
localhost:8888/notebooks/Desktop/Fake%20News%20Prediction/Fake%20News%20Prediction.ipynb
Jupyter Fake News Prediction Last Checkpoint: 27 minutes ago (unsaved changes) Python 3 (pykernel)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (pykernel)

In [40]: import numpy as np
import pandas as pd
import re
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score

In [41]: import nltk
nltk.download('stopwords')

[nltk_data] Downloading package stopwords to
[nltk_data] C:\Users\HP\AppData\Roaming\nltk_data...
[nltk_data] Package stopwords is already up-to-date!

Out[41]: True

In [42]: # printing the stopwords in English
print(stopwords.words('english'))

['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've", "you'll", "you'd", 'your', 'yours', 'y
ourself', 'yourselves', 'he', 'him', 'his', 'himself', 'she', "she's", 'her', 'hers', 'herself', 'it', "it's", 'its', 'itself',
'they', 'them', 'their', 'theirs', 'themselves', 'what', 'which', 'who', 'whom', 'this', 'that', "that'll", 'these', 'those',
'am', 'is', 'are', 'was', 'were', 'be', 'been', 'being', 'have', 'has', 'had', 'having', 'do', 'does', 'did', 'doing', 'a', 'a
n', 'the', 'and', 'but', 'if', 'or', 'because', 'as', 'until', 'while', 'of', 'at', 'by', 'for', 'with', 'about', 'against', 'b
etween', 'into', 'through', 'during', 'before', 'after', 'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out', 'on', 'of
f', 'over', 'under', 'again', 'further', 'then', 'once', 'here', 'there', 'when', 'where', 'why', 'how', 'all', 'any', 'both',
'each', 'few', 'more', 'most', 'other', 'some', 'such', 'no', 'nor', 'not', 'only', 'own', 'same', 'so', 'than', 'too', 'very',
's', 't', 'can', 'will', 'just', 'don', "don't", 'should', "should've", 'now', 'd', 'll', 'm', 'o', 're', 've', 'y', 'ain', 'ar
en', 'aren't', 'couldn', 'couldn't', 'didn', "didn't", 'doesn', "doesn't", 'hadn', "hadn't", 'hasn', "hasn't", 'haven', 'have
n't', 'isn', 'isn't', 'ma', 'mightn', "mightn't", 'mustn', "mustn't", 'needn', "needn't", 'shan', "shan't", 'shouldn', 'should
```

Data Pre-processing

```
Desktop/Fake News Prediction/ x Fake News Prediction - Jupyter | x +
localhost:8888/notebooks/Desktop/Fake%20News%20Prediction/Fake%20News%20Prediction.ipynb
Jupyter Fake News Prediction Last Checkpoint: 27 minutes ago (unsaved changes) Python 3 (pykernel)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (pykernel)

In [43]: # Loading the dataset to a pandas DataFrame
news_dataset = pd.read_csv("C:\Users\HP\Desktop\Fake News Prediction\train.csv")

In [44]: news_dataset.shape
Out[44]: (20800, 5)

In [45]: # print the first 5 rows of the dataframe
news_dataset.head()
Out[45]:
   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 Aistr...  1
4  4  Iranian woman jailed for fictional unpublished...  Howard Portnoy  Print 'nAn Iranian woman has been sentenced to...  1

In [46]: # counting the number of missing values in the dataset
news_dataset.isnull().sum()
Out[46]:
id          0
title      558
author     1957
text        39
label       0
dtype: int64

In [47]: # replacing the null values with empty string
news_dataset = news_dataset.fillna('')
```

```

Desktop/Fake News Prediction/ x Fake News Prediction - Jupyter | x +
localhost:8888/notebooks/Desktop/Fake%20News%20Prediction/Fake%20News%20Prediction.ipynb
Jupyter Fake News Prediction Last Checkpoint: 27 minutes ago (unsaved changes)
Python 3 (ipykernel)

In [47]: # replacing the null values with empty string
news_dataset = news_dataset.fillna("")

In [48]: # merging the author name and news title
news_dataset['content'] = news_dataset['author']+' '+news_dataset['title']

In [49]: 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

In [50]: # separating the data & label
X = news_dataset.drop(columns='label', axis=1)
Y = news_dataset['label']

In [51]: print(X)
print(Y)

      id      title \
0      0  House Dem Aide: We Didn't Even See Comey's Let...
1      1  FLYNN: Hillary Clinton, Big Woman on Campus - ...
2      2      Why the Truth Might Get You Fired
3      3  15 Civilians Killed In Single US Airstrike Hav...
4      4  Iranian woman jailed for fictional unpublished...
...
20795 20795  Rapper T.I.: Trump a 'Poster Child For White S...
20796 20796  N.F.L. Playoffs: Schedule, Matchups and Odds -...
20797 20797  Macy's Is Said to Receive Takeover Approach by...
20798 20798  NATO, Russia To Hold Parallel Exercises In Bal...
20799 20799  what Keeps the F-35 Alive

```

```

Desktop/Fake News Prediction/ x Fake News Prediction - Jupyter | x +
localhost:8888/notebooks/Desktop/Fake%20News%20Prediction/Fake%20News%20Prediction.ipynb
Jupyter Fake News Prediction Last Checkpoint: 27 minutes ago (unsaved changes)
Python 3 (ipykernel)

In [51]: print(X)
print(Y)

      id      title \
0      0  House Dem Aide: We Didn't Even See Comey's Let...
1      1  FLYNN: Hillary Clinton, Big Woman on Campus - ...
2      2      Why the Truth Might Get You Fired
3      3  15 Civilians Killed In Single US Airstrike Hav...
4      4  Iranian woman jailed for fictional unpublished...
...
20795 20795  Rapper T.I.: Trump a 'Poster Child For White S...
20796 20796  N.F.L. Playoffs: Schedule, Matchups and Odds -...
20797 20797  Macy's Is Said to Receive Takeover Approach by...
20798 20798  NATO, Russia To Hold Parallel Exercises In Bal...
20799 20799  what Keeps the F-35 Alive

      author \
0      Darrell Lucus
1      Daniel J. Flynn
2      Consortiumnews.com
3      Jessica Purkiss
4      Howard Portnoy
...
20795  Jerome Hudson
20796  Benjamin Hoffman
20797  Michael J. de la Merced and Rachel Abrams
20798  Alex Ansary
20799  David Swanson

      text \
0      House Dem Aide: We Didn't Even See Comey's Let...
1      Ever get the feeling your life circles the rou...
2      Why the Truth Might Get You Fired October 29, ...
3      Videos 15 civilians killed in single US airstr...
4      Print \nAn Iranian woman has been sentenced to...
...
20795  Rapper T. I. unloaded on black celebrities who...
20796  When the Green Bay Packers lost to the Washing...
20797  The Macy's of today grew from the union of sev...
20798  NATO, Russia To Hold Parallel Exercises To Bal...
20799  what Keeps the F-35 Alive

```

```
Desktop/Fake News Prediction/ x Fake News Prediction - Jupyter | x +
localhost:8888/notebooks/Desktop/Fake%20News%20Prediction/Fake%20News%20Prediction.ipynb
Jupyter Fake News Prediction Last Checkpoint: 28 minutes ago (autosaved) Logout
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)
20798 20798 NATO, Russia To Hold Parallel Exercises In Bal...
20799 20799 what Keeps the F-35 Alive
author \
0 Darrell Lucus
1 Daniel J. Flynn
2 Consortiumnews.com
3 Jessica Purkiss
4 Howard Portnoy
...
20795 Jerome Hudson
20796 Benjamin Hoffman
20797 Michael J. de la Merced and Rachel Abrams
20798 Alex Ansary
20799 David Swanson
text \
0 House Dem Aide: We Didn't Even See Comey's Let...
1 Ever get the feeling your life circles the rou...
2 why the Truth Might Get You Fired October 29, ...
3 Videos 15 Civilians Killed In Single US Aistr...
4 Print \n\n Iranian woman has been sentenced to...
...
20795 Rapper T. I. unloaded on black celebrities who...
20796 When the Green Bay Packers lost to the Washing...
20797 The Macy's of today grew from the union of sev...
20798 NATO, Russia To Hold Parallel Exercises In Bal...
20799 David Swanson is an author, activist, journa...
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...
```

```
Desktop/Fake News Prediction/ x Fake News Prediction - Jupyter | x +
localhost:8888/notebooks/Desktop/Fake%20News%20Prediction/Fake%20News%20Prediction.ipynb
Jupyter Fake News Prediction Last Checkpoint: 28 minutes ago (autosaved) Logout
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)
0 House Dem Aide: We Didn't Even See Comey's Let...
1 Ever get the feeling your life circles the rou...
2 Why the Truth Might Get You Fired October 29, ...
3 Videos 15 Civilians Killed In Single US Aistr...
4 Print \n\n Iranian woman has been sentenced to...
...
20795 Rapper T. I. unloaded on black celebrities who...
20796 When the Green Bay Packers lost to the Washing...
20797 The Macy's of today grew from the union of sev...
20798 NATO, Russia To Hold Parallel Exercises In Bal...
20799 David Swanson is an author, activist, journa...
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
[20800 rows x 5 columns]
0 1
1 0
2 1
3 1
4 1
..
20795 0
20796 0
20797 0
20798 1
20799 1
Name: label, Length: 20800, dtype: int64
Stemming:
```

Stemming

```
Stemming:

Stemming is the process of reducing a word to its Root word

example:

actor, actress, acting -> act

In [52]: port_stem = PorterStemmer()

In [53]: def stemming(content):
stemmed_content = re.sub('[^a-zA-Z]', ' ', content)
stemmed_content = stemmed_content.lower()
stemmed_content = stemmed_content.split()
stemmed_content = [port_stem.stem(word) for word in stemmed_content if not word in stopwords.words('english')]
stemmed_content = ' '.join(stemmed_content)
return stemmed_content

In [54]: news_dataset['content'] = news_dataset['content'].apply(stemming)

In [55]: 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
```

```
In [55]: 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

In [56]: #separating the data and label
X = news_dataset['content'].values
Y = news_dataset['label'].values

In [57]: 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 [58]: print(Y)
[1 0 1 ... 0 1 1]

In [59]: Y.shape
Out[59]: (20800,)

In [60]: # converting the textual data to numerical data
vectorizer = TfidfVectorizer()
```

Desktop/Fake News Prediction/ Fake News Prediction - Jupyter | localhost:8888/notebooks/Desktop/Fake%20News%20Prediction/Fake%20News%20Prediction.ipynb

Jupyter Fake News Prediction Last Checkpoint: 29 minutes ago (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (pykernel)

```
In [59]: Y.shape
Out[59]: (20800,)
```

```
In [60]: # converting the textual data to numerical data
vectorizer = TfidfVectorizer()
vectorizer.fit(X)
X = vectorizer.transform(X)
```

```
In [61]: 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)	0.270532480845492
(0, 3600)	0.3598939188262559
(0, 2959)	0.2468450128533713
(0, 2483)	0.3676519686797209
(0, 267)	0.2701012497708706
(1, 16799)	0.3007174565510157
(1, 6816)	0.1904660198296849
(1, 5503)	0.7143299355715573
(1, 3568)	0.26373768806048464
(1, 2813)	0.19094574062359204
(1, 2223)	0.3827320386859759
(1, 1894)	0.15521974226349364
(1, 1497)	0.2939891562094648
(2, 15611)	0.41544962664721613
(2, 9620)	0.49351492943649944
(2, 5968)	0.3474613386728292
(2, 5389)	0.3866530551182615
(2, 3103)	0.46097489583229645
:	:

Desktop/Fake News Prediction/ Fake News Prediction - Jupyter | localhost:8888/notebooks/Desktop/Fake%20News%20Prediction/Fake%20News%20Prediction.ipynb

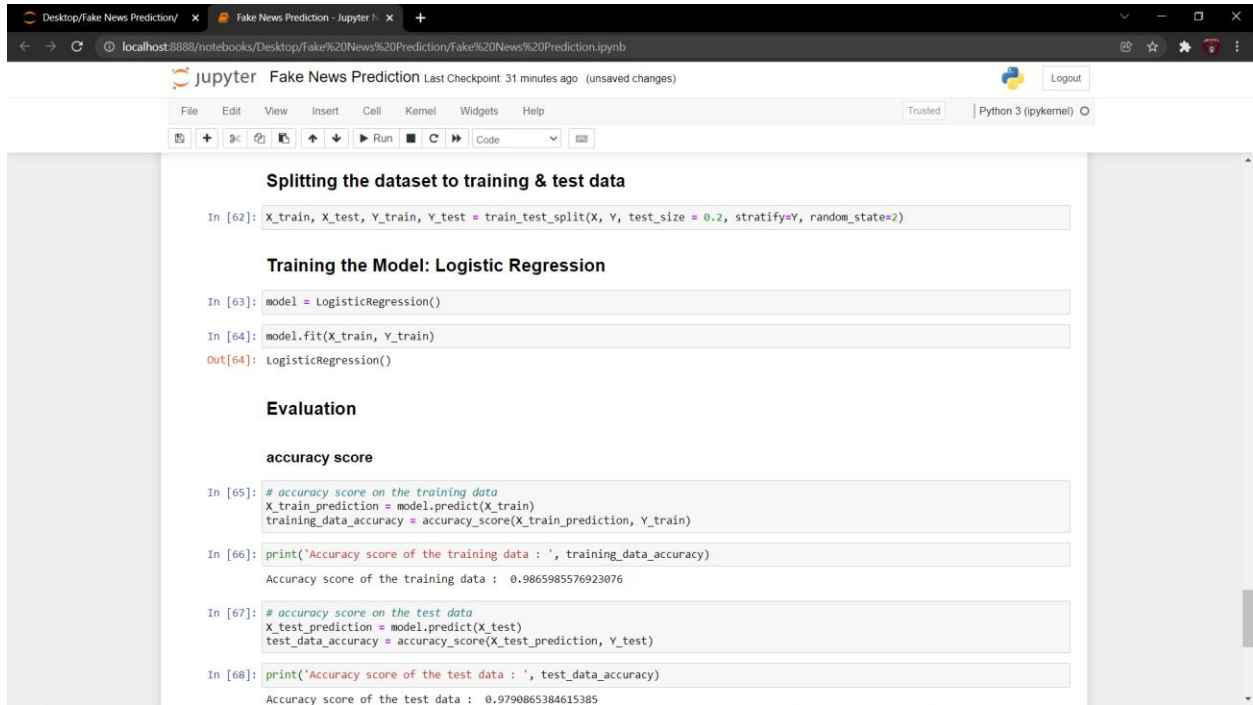
Jupyter Fake News Prediction Last Checkpoint: 30 minutes ago (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (pykernel)

```
(1, 16799) 0.3007174565510157
(1, 6816) 0.1904660198296849
(1, 5503) 0.7143299355715573
(1, 3568) 0.26373768806048464
(1, 2813) 0.19094574062359204
(1, 2223) 0.3827320386859759
(1, 1894) 0.15521974226349364
(1, 1497) 0.2939891562094648
(2, 15611) 0.41544962664721613
(2, 9620) 0.49351492943649944
(2, 5968) 0.3474613386728292
(2, 5389) 0.3866530551182615
(2, 3103) 0.46097489583229645
:
:
```

(20797, 13122)	0.2482526352197606
(20797, 12344)	0.27263457663336677
(20797, 12138)	0.2478257724396507
(20797, 10306)	0.08038079000566466
(20797, 9588)	0.174553480255222
(20797, 9518)	0.2954204003420313
(20797, 8988)	0.36160868928000795
(20797, 8364)	0.22322585870464118
(20797, 7042)	0.21799048897828688
(20797, 3643)	0.21155500613623743
(20797, 1287)	0.33538056804139865
(20797, 699)	0.30685846079762347
(20797, 43)	0.29710241860790626
(20798, 13046)	0.22363267488270608
(20798, 11052)	0.4460515589182236
(20798, 10177)	0.3192496370187028
(20798, 6889)	0.32496285694299426
(20798, 5032)	0.4083701450239529
(20798, 1125)	0.4460515589182236
(20798, 588)	0.3112141524638974
(20798, 350)	0.28446937819072576
(20799, 14852)	0.567757267055112
(20799, 8036)	0.45983893273780013
(20799, 3623)	0.37927626273066584
(20799, 377)	0.567757267055112

Splitting the Dataset



The screenshot shows a Jupyter Notebook interface with the following content:

```
In [62]: X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.2, stratify=Y, random_state=2)
```

Training the Model: Logistic Regression

```
In [63]: model = LogisticRegression()
In [64]: model.fit(X_train, Y_train)
Out[64]: LogisticRegression()
```

Evaluation

accuracy score

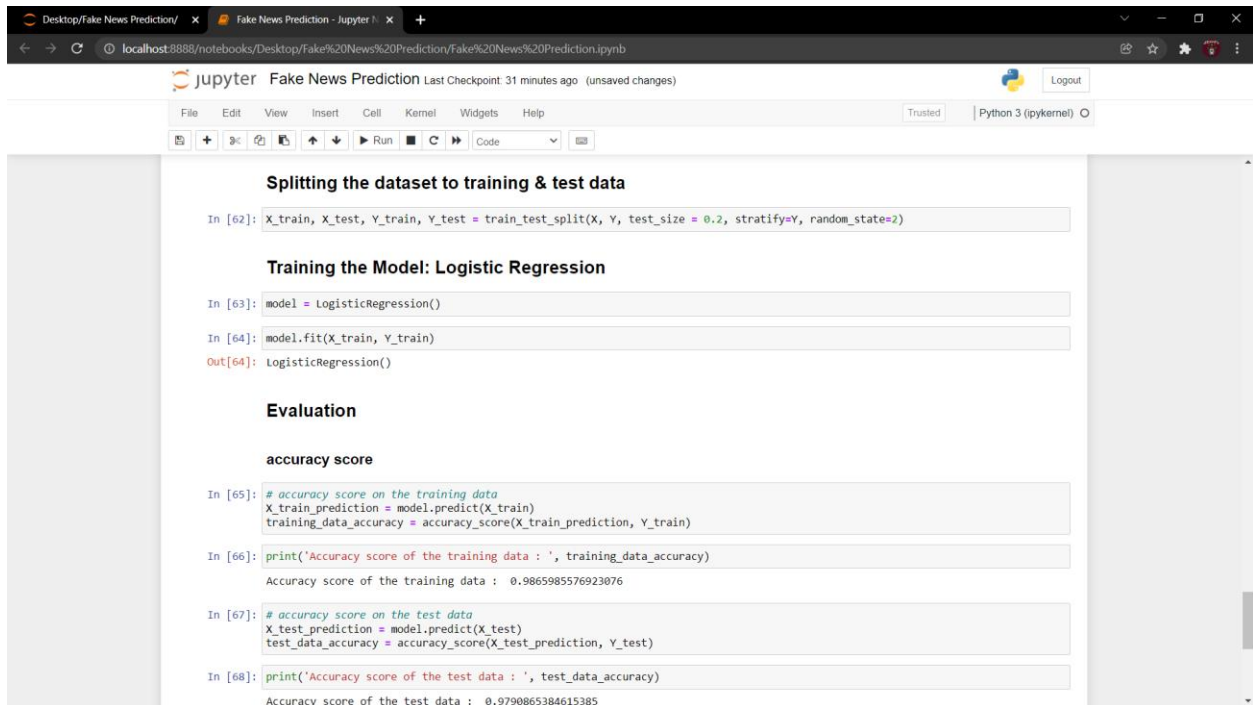
```
In [65]: # accuracy score on the training data
X_train_prediction = model.predict(X_train)
training_data_accuracy = accuracy_score(X_train_prediction, Y_train)

In [66]: print('Accuracy score of the training data : ', training_data_accuracy)
Accuracy score of the training data : 0.9865985576923076

In [67]: # accuracy score on the test data
X_test_prediction = model.predict(X_test)
test_data_accuracy = accuracy_score(X_test_prediction, Y_test)

In [68]: print('Accuracy score of the test data : ', test_data_accuracy)
Accuracy score of the test data : 0.9790865384615385
```

Training the Dataset



The screenshot shows a Jupyter Notebook interface with the following content:

```
In [62]: X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.2, stratify=Y, random_state=2)
```

Training the Model: Logistic Regression

```
In [63]: model = LogisticRegression()
In [64]: model.fit(X_train, Y_train)
Out[64]: LogisticRegression()
```

Evaluation

accuracy score

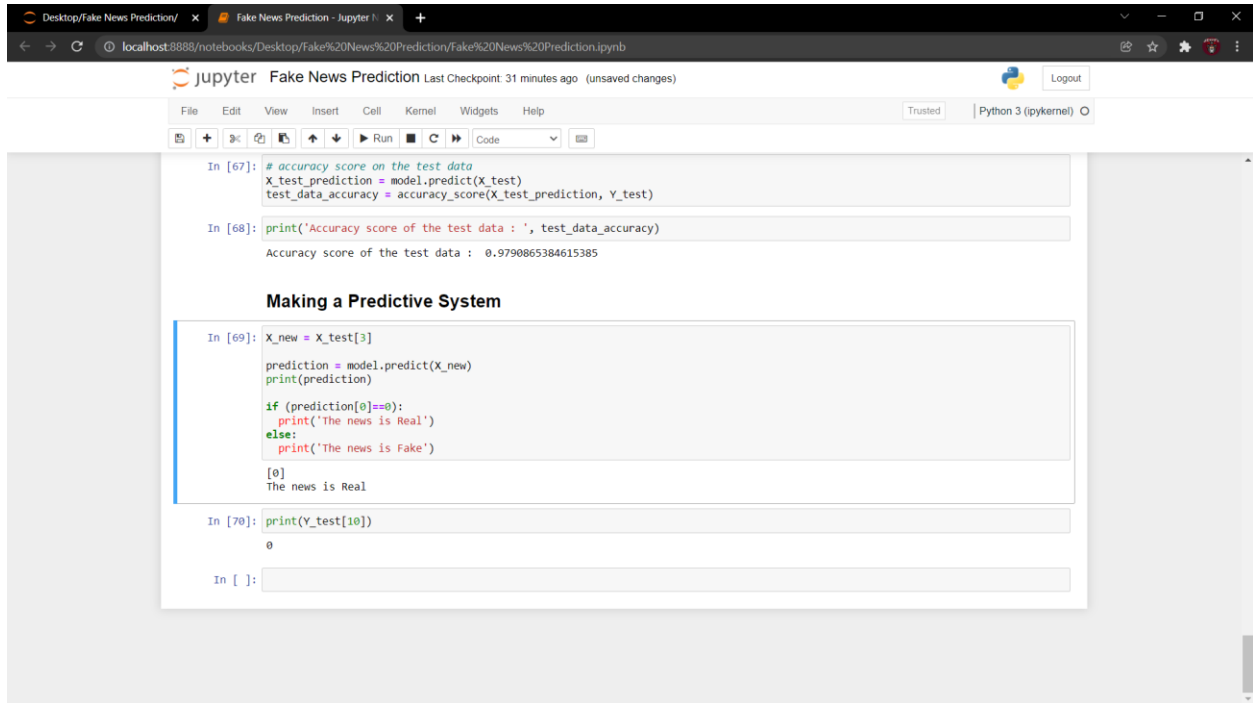
```
In [65]: # accuracy score on the training data
X_train_prediction = model.predict(X_train)
training_data_accuracy = accuracy_score(X_train_prediction, Y_train)

In [66]: print('Accuracy score of the training data : ', training_data_accuracy)
Accuracy score of the training data : 0.9865985576923076

In [67]: # accuracy score on the test data
X_test_prediction = model.predict(X_test)
test_data_accuracy = accuracy_score(X_test_prediction, Y_test)

In [68]: print('Accuracy score of the test data : ', test_data_accuracy)
Accuracy score of the test data : 0.9790865384615385
```

Making a Predictive System



The screenshot shows a Jupyter Notebook interface with the following content:

```
In [67]: # accuracy score on the test data
X_test_prediction = model.predict(X_test)
test_data_accuracy = accuracy_score(X_test_prediction, Y_test)

In [68]: print('Accuracy score of the test data : ', test_data_accuracy)
Accuracy score of the test data : 0.9790865384615385
```

Making a Predictive System

```
In [69]: X_new = X_test[3]
prediction = model.predict(X_new)
print(prediction)

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

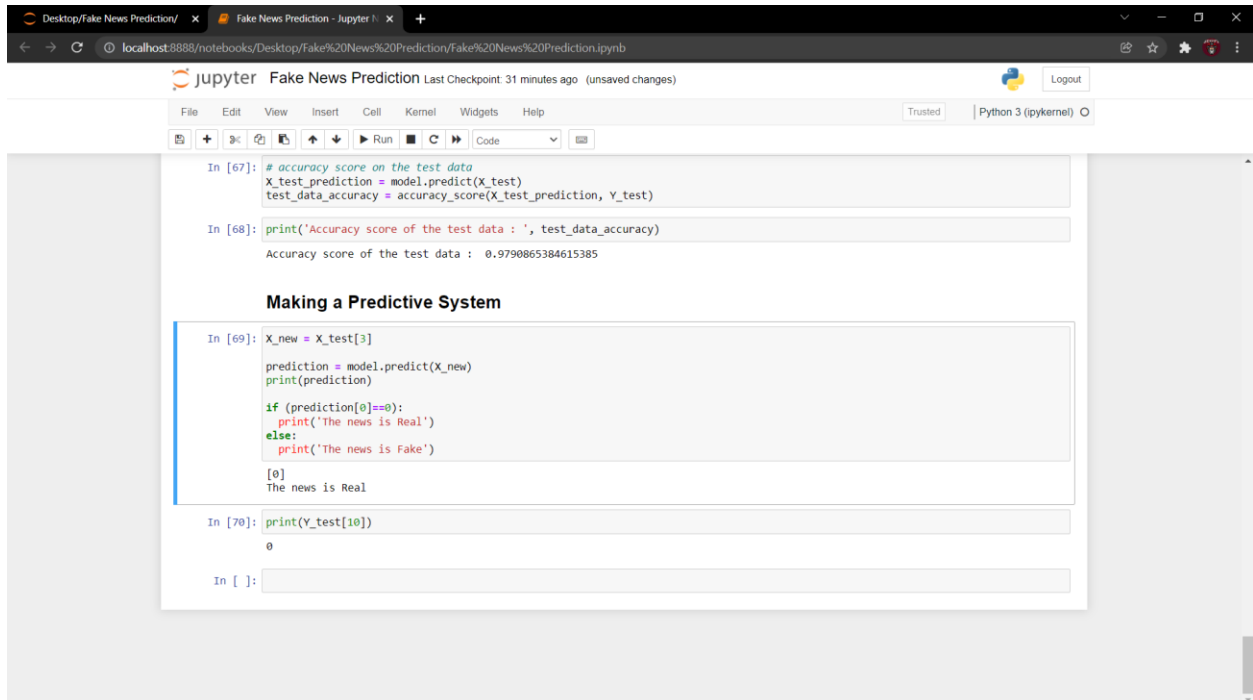
[0]
The news is Real

In [70]: print(Y_test[10])
0

In [ ]:
```

Output –

Attaching 5 different outputs



This screenshot is identical to the one above, showing the same Jupyter Notebook code and output. It displays the calculation of test data accuracy and the implementation of a simple predictive system that classifies news as 'Real' or 'Fake' based on a model's prediction.

Desktop/Fake News Prediction/ x Fake News Prediction - Jupyter | x +

localhost:8888/notebooks/Desktop/Fake%20News%20Prediction/Fake%20News%20Prediction.ipynb

Jupyter Fake News Prediction Last Checkpoint: 32 minutes ago (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (pykernel)

Accuracy score of the training data : 0.9865985576923076

```
In [67]: # accuracy score on the test data
X_test_prediction = model.predict(X_test)
test_data_accuracy = accuracy_score(X_test_prediction, Y_test)

In [68]: print('Accuracy score of the test data : ', test_data_accuracy)
Accuracy score of the test data : 0.9790865384615385
```

Making a Predictive System

```
In [69]: X_new = X_test[3]
prediction = model.predict(X_new)
print(prediction)

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

[0]
The news is Real

In [71]: print(Y_test[12])
1

In [ ]:
```

Desktop/Fake News Prediction/ x Fake News Prediction - Jupyter | x +

localhost:8888/notebooks/Desktop/Fake%20News%20Prediction/Fake%20News%20Prediction.ipynb

Jupyter Fake News Prediction Last Checkpoint: 32 minutes ago (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (pykernel)

Accuracy score of the training data : 0.9865985576923076

```
In [67]: # accuracy score on the test data
X_test_prediction = model.predict(X_test)
test_data_accuracy = accuracy_score(X_test_prediction, Y_test)

In [68]: print('Accuracy score of the test data : ', test_data_accuracy)
Accuracy score of the test data : 0.9790865384615385
```

Making a Predictive System

```
In [69]: X_new = X_test[3]
prediction = model.predict(X_new)
print(prediction)

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

[0]
The news is Real

In [72]: print(Y_test[2])
1

In [ ]:
```


Desktop/Fake News Prediction/ x Fake News Prediction - Jupyter | x +

localhost:8888/notebooks/Desktop/Fake%20News%20Prediction/Fake%20News%20Prediction.ipynb

Jupyter Fake News Prediction Last Checkpoint: 32 minutes ago (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Accuracy score of the training data : 0.9865985576923076

```
In [67]: # accuracy score on the test data
X_test_prediction = model.predict(X_test)
test_data_accuracy = accuracy_score(X_test_prediction, Y_test)

In [68]: print('Accuracy score of the test data : ', test_data_accuracy)
Accuracy score of the test data : 0.9790865384615385
```

Making a Predictive System

```
In [69]: X_new = X_test[3]
prediction = model.predict(X_new)
print(prediction)

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

[0]
The news is Real

In [74]: print(Y_test[23])
0

In [ ]:
```

Desktop/Fake News Prediction/ x Fake News Prediction - Jupyter | x +

localhost:8888/notebooks/Desktop/Fake%20News%20Prediction/Fake%20News%20Prediction.ipynb

Jupyter Fake News Prediction Last Checkpoint: 33 minutes ago (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Accuracy score of the training data : 0.9865985576923076

```
In [67]: # accuracy score on the test data
X_test_prediction = model.predict(X_test)
test_data_accuracy = accuracy_score(X_test_prediction, Y_test)

In [68]: print('Accuracy score of the test data : ', test_data_accuracy)
Accuracy score of the test data : 0.9790865384615385
```

Making a Predictive System

```
In [69]: X_new = X_test[3]
prediction = model.predict(X_new)
print(prediction)

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

[0]
The news is Real

In [75]: print(Y_test[43])
1

In [ ]:
```

Localhost:8888/notebooks/Desktop/Fake%20News%20Prediction/Fake%20News%20Prediction.ipynb

Jupyter Fake News Prediction Last Checkpoint: 3 hours ago (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (pykernel)

```
In [28]: # accuracy score on the test data
X_test_prediction = model.predict(X_test)
test_data_accuracy = accuracy_score(X_test_prediction, Y_test)

In [29]: print('Accuracy score of the test data : ', test_data_accuracy)
Accuracy score of the test data : 0.9790865384615385
```

Making a Predictive System

```
In [30]: X_new = X_test[3]

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

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

[0]
The news is Real

In [32]: print(Y_test[63])
0

In [ ]:
```

Localhost:8888/notebooks/Desktop/Fake%20News%20Prediction/Fake%20News%20Prediction.ipynb

Jupyter Fake News Prediction Last Checkpoint: 3 hours ago (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (pykernel)

```
Accuracy score of the training data : 0.9865985576923076

In [28]: # accuracy score on the test data
X_test_prediction = model.predict(X_test)
test_data_accuracy = accuracy_score(X_test_prediction, Y_test)

In [29]: print('Accuracy score of the test data : ', test_data_accuracy)
Accuracy score of the test data : 0.9790865384615385
```

Making a Predictive System

```
In [30]: X_new = X_test[3]

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

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

[0]
The news is Real

In [33]: print(Y_test[67])
1

In [ ]:
```

Localhost:8888/notebooks/Desktop/Fake%20News%20Prediction/Fake%20News%20Prediction.ipynb

Jupyter Fake News Prediction Last Checkpoint: 3 hours ago (unsaved changes)

Python 3 (pykernel)

```
Accuracy score of the training data : 0.9865985576923076
```

```
In [28]: # accuracy score on the test data
X_test_prediction = model.predict(X_test)
test_data_accuracy = accuracy_score(X_test_prediction, Y_test)

In [29]: print('Accuracy score of the test data : ', test_data_accuracy)
Accuracy score of the test data : 0.9790865384615385
```

Making a Predictive System

```
In [30]: X_new = X_test[3]
prediction = model.predict(X_new)
print(prediction)

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

[0]
The news is Real
```

```
In [34]: print(Y_test[17])
1
```

```
In [ ]:
```

Localhost:8888/notebooks/Desktop/Fake%20News%20Prediction/Fake%20News%20Prediction.ipynb

Jupyter Fake News Prediction Last Checkpoint: 3 hours ago (autosaved)

Notebook saved Trusted Python 3 (pykernel)

```
Accuracy score of the training data : 0.9865985576923076
```

```
In [28]: # accuracy score on the test data
X_test_prediction = model.predict(X_test)
test_data_accuracy = accuracy_score(X_test_prediction, Y_test)

In [29]: print('Accuracy score of the test data : ', test_data_accuracy)
Accuracy score of the test data : 0.9790865384615385
```

Making a Predictive System

```
In [30]: X_new = X_test[3]
prediction = model.predict(X_new)
print(prediction)

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

[0]
The news is Real
```

```
In [35]: print(Y_test[87])
0
```

```
In [ ]:
```

Browser tabs: New Tab, Desktop/Fake News Prediction/, Fake News Prediction - Jupyter | x +

Address bar: localhost:8888/notebooks/Desktop/Fake%20News%20Prediction/Fake%20News%20Prediction.ipynb

Jupyter Fake News Prediction Last Checkpoint: 3 hours ago (unsaved changes) Python 3 (ipykernel)

File Edit View Insert Cell Kernel Widgets Help Trusted

```
Accuracy score of the training data : 0.9865985576923076
```

```
In [28]: # accuracy score on the test data
X_test_prediction = model.predict(X_test)
test_data_accuracy = accuracy_score(X_test_prediction, Y_test)
```

```
In [29]: print('Accuracy score of the test data : ', test_data_accuracy)
Accuracy score of the test data : 0.9790865384615385
```

Making a Predictive System

```
In [30]: X_new = X_test[3]
prediction = model.predict(X_new)
print(prediction)

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

```
[0]
The news is Real
```

```
In [36]: print(Y_test[59])
1
```

```
In [ ]:
```

CONCLUSION

Summary

With the help of Machine Learning we have created 5 prediction model which gives the accuracy above 90% and it cover all latest political covid 19 news. Also, with some pretrained model we have cover news related to history and sport.

We intent to build our own dataset which will be kept up to data according to the latest news in future.

Future Scope

This project can be further enhanced to provide greater flexibility and performance with certain modification whenever necessary.

References

- Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu, "Fake News Detection on Social Media: A Data Mining Perspective" arXiv:1708.01967v3 [cs.SI], 3Sep 2017
- M. Granik and V. Mesyura, "Fake news detection using naive Bayes classifier," 2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON), Kiev, 2017, pp. 900-903.
- Fake news websites. (n.d.) Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Fake_news_website. Accessed Feb. 6, 2017
- Cade Metz. (2016, Dec. 16). The bittersweet sweepstakes to build an AI that destroys fake news.
- Conroy, N., Rubin, V. and Chen, Y. (2015). "Automatic deception detection: Methods for finding fake news" at Proceedings of the Association for Information Science and Technology, 52(1), pp.1-4.