

Fake News Analysis

BE-CE Semester- VII

Prepared at



ISO 9001:2008

ISO 27001:2013

CMMI LEVEL-5

**Bhaskaracharya National Institute for Space Applications & Geo-informatics
Ministry of Electronics and Information Technology, Govt. of India.**

Gandhinagar

Prepared By

Shekh Mohammad Ayan

Krishna Zala

Dhyan Patel

ID No. 03

ID No. 03

ID No. 03

Guided By:

Prof. Hardik Joshi

Prof .Jenish Shah

Prof.Bhargav Suthar

Department of Computer Engineering

LJIET, Ahmedabad.

External Co- Guide:

Harsh Kiratsata

Project Scientist

BISAG- N, Gandhinagar.

SUBMITTED TO

GTU



L.J. Institute Of Engineering And Technology



CERTIFICATE

*This is to certify that the project report compiled by **Mr. Shekh Mohammad Ayan Moh. Faruk, Ms. Krishna Ashokbhai Zala and Mr. Dhyan Patel Sanjaybhai** students of 8th Semester **BE -CE** from **L.J. Institute of Engineering And Technology, GTU Ahmedabad** have completed their final Semester internship project satisfactorily. To the best of our knowledge this is an original and bonafide work done by them. They have worked on Machine Learning for **"Fake News Analysis"**, starting from January 3rd, 2022 to April 3rd, 2022.*

During their tenure at this Institute, they were found to be sincere and meticulous in their work. We appreciate their enthusiasm & dedication towards the work assigned to them.

We wish them every success.

Punit Lalwani

Project Scientist,

BISAG- N, Gandhinagar

T. P. Singh

Director General,

BISAG- N, Gandhinagar

Sample Certificate of College

CERTIFICATE

*This is to certify that the 8th Semester Internship Project entitled "**Fake News Analysis**" has been carried out by **Shekh Mohammad Ayan Krishna Zala & Patel Dhyan** under my guidance in fulfilment of the degree of Bachelor of Engineering in COMPUTER ENGINEERING (8th Semester) of L.J. Institute of Engineering and Technology –Ahmedabad, during the academic year 2022.*

Guide

Prof.Hardik Joshi

Prof.Jenish Shah

Prof.Bhargav Suthar

Head of the Department

Prof. Shweta Yagnik



About BISAG- N



ABOUT THE INSTITUTE

Modern day planning for inclusive development and growth calls for transparent, efficient, effective, responsive and low cost decision making systems involving multi-disciplinary information such that it not only encourages people's participation, ensuring equitable development but also takes into account the sustainability of natural resources. The applications of space technology and Geo-informatics have contributed significantly towards the socio-economic development. Taking cognizance of the need of geo-spatial information for developmental planning and management of resources, the department of Ministry of Electronics and Information Technology, Government of India, established "Bhaskaracharya National Institute for Space Applications and Geo-informatics" (BISAG- N). BISAG- N is an ISO 9001:2008, ISO 27001:2005 and CMMI: 5 certified institute. BISAG- N which was initially set up to carryout space technology applications, has evolved into a centre of excellence, where research and innovations are combined with the requirements of users and thus acts as a value added service provider, a technology developer and as a facilitator for providing direct benefits of space technologies to the grass root level functions/functionaries.

BISAG- N's Enduring Growth

Since its foundation, the Institute has experienced extensive growth in the sphere of Space technology and Geo-informatics. The objective with which BISAG- N was established is manifested in the extent of services it renders to almost all departments of the State. Year after year the institute has been endeavouring to increase its outreach to disseminate the use of geo-informatics up to grassroots level. In this span of nine years, BISAG- N has assumed multi-dimensional roles and achieved several milestones to become an integral part of the development process of the Gujarat State.

BISAG-N Timeline

2003-04



Gujarat
SATCOM
Network

2007-08



Centre for
Geo-informatics
Applications

2010-11



Academy of
Geo-informatics
for Sustainable
Development

2012-13

A full-fledged
Campus

Activities



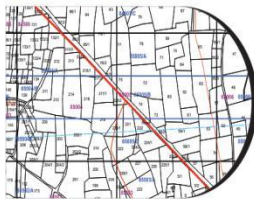
Satellite Communication..

for promotion and facilitation of the use of broadcast and teleconferencing networks for distant interactive training, education and extension.



Remote Sensing..

for Inventory, Mapping, Developmental planning and Monitoring of natural & man-made resources.



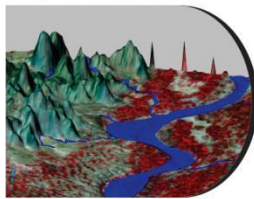
Geographic Information System..

for conceptualization, creation and organization of multi purpose common digital database for sectoral/integrated decision support systems.



Global Navigation Satellite System..

for Location based Services, Geo-referencing, Engineering Applications and Research.



Photogrammetry..

for Creation of Digital Elevation Model, Terrain Characteristic, Resource planning.



Cartography..

for thematic mapping, value added maps.



Software Development..

for wider usage of Geo-spatial applications, Decision Support Systems (desktop as well as web based), ERP solutions.



Education, Research and Training..

for providing Education, Research, Training & Technology Transfer to large number of students, end users & collaborators.

Applications of Geospatial Technology for Good Governance: Institutionalization

Through the geospatial technology, the actual situation on the ground can be accessed. The real life data collected through the technology forms the strong foundation for development of effective social welfare programs benefiting directly the grass root level people. The geospatial data collected by the space borne sensors along with powerful software support through Geographic Information System (GIS), the vital spatio-temporal maps, tables, and various statistics are being generated which feed into Decision Support System (DSS).

A multi-threaded approach is followed in the process of institutionalization of development of such applications. The 5 common threads which run through all the processes are: *Acceptability, Adaptability, Affordability, Availability and Assimilability*.

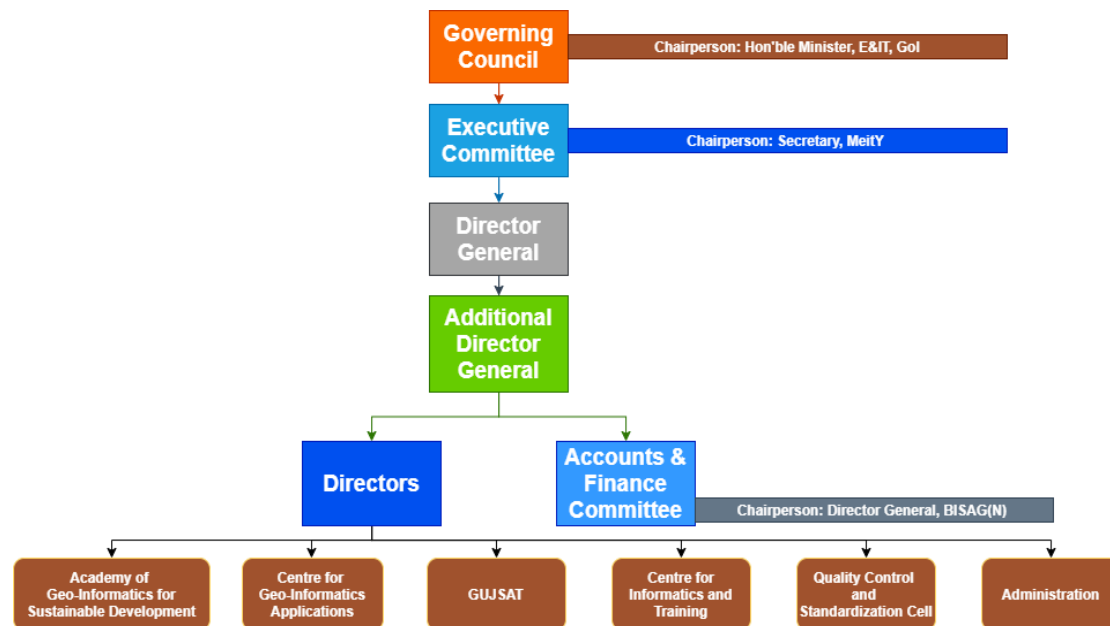
These are the “Watch Words” which any application developer has to meet. The “acceptability” addresses the issue that the application developed has met the wide acceptability among the users departments and the ultimate end beneficiary by way of providing all necessary data and statistics required. The “affordability” addresses the issue of the application product being cost effective. The “availability” aspect looks into aspect of easily accessible across any platform, anywhere and anytime. The applications should have inbuilt capability of easy adaptability to the changing spatio- and temporal resolutions of data, new aspects of requirements arising from time to time from users. The assimilability aspect ensures that the data from various sources / resolutions and technologies can be seamlessly integrated.

ACCEPTABILITY	<ul style="list-style-type: none"> ▪ Problem definition by users • Proof of Concept development without financial liability on users ▪ Execution through collaboration under user’s ownership
ADOPTABILITY	<ul style="list-style-type: none"> ▪ Applications as per present systems & database ▪ Maximum Automation ▪ Minimum capacity building requirement at the user end
AFFORDABILITY :	<ul style="list-style-type: none"> ▪ Multipurpose geo-spatial database, common, compatible, standardized (100s of layers) ▪ In house developed/open source software ▪ Full Utilization of available assets
AVAILABILITY:	<ul style="list-style-type: none"> ▪ Departmental /Integrated DSS ▪ Desired Product delivery anytime, anywhere in the country
ASSIMILABILITY	<ul style="list-style-type: none"> ▪ Integration of Various technologies like RS, GIS, GPS, Web MIS, Mobile etc.

Organizational Setup

The Institute is responsible for providing information and technical support to different Departments and Organizations. The Governing Body and the Empowered Executive Committee govern the functioning of BISAG- N. The Institute is registered under the Societies Registration Act 1860. Considering the scope and extent of activities of BISAG- N, its organizational structure has been charted out with defined functions.

Organizational Setup of BISAG- N



Governing Body

For smoother, easier and faster institutionalization of Remote Sensing and GIS technology, decision makers of the state were brought together to form the Governing Body. It is the supreme executive authority of the Institute. The Governing Body comprises of ex-officio members from various Government departments and Institutes.

◆ Hon'ble Minister of Electronics and Information Technology	H
◆ Hon'ble Minister of State Electronics and Information Technology	H
◆ Secretary of Government of India: Ministry of Electronics and Information Technology.....	S
◆ Chief Executive Officer, Niti Aayog	C
◆ Chairman, Indian Space Research Organization	C
◆ Secretary to Government of India: Department of Science and Technology	S
◆ Additional Secretary to Government of India: Ministry of Electronics and Technology	A
◆ Chief Secretary to Government of Gujarat	C
◆ Resident & Chief Executive Officer, National e-Governance Division, Ministry of Electronics and Information Technology.....	P
◆	a
◆	Member (Ex-Office)

- ◆ Financial Advisor to Government of India: Ministry of Electronics and Information TechnologyMember (Ex-Officio).....F
- ◆ Distinguished Professionals from the GIS field-Three (3) (To be nominated by the Chairperson).....D
- ◆ Director-General, Bhaskaracharya National Institute for Space Application and Geo-Informatics {BISAG(N)} Member Secretary (Ex-Officio).....D

Centre for Geo-informatics Applications

Introduction



The objective of this technology group is to provide decision support to the sectoral stakeholders through scientifically organized, comprehensive, multi-purpose, compatible and large scale (village level) geo-spatial databases and supporting analytical tools. These activities of this unit are executed by a well-trained team of multi-disciplinary scientists. The government has provided a modern infrastructure along with the state-of-the-art hardware and software. To study the land transformation and development over the years, a satellite digital data library of multiple sensors of last twenty years has been established and conventional data sets of departments have been co-registered with satellite data. The geo-spatial databases have been created using conventional maps, high resolution satellite 2D and 3D imagery and official datasets (attributes). The geo-spatial databases include terrain characteristics, natural and administrative systems, agriculture, water resources, city survey maps, village maps with survey numbers, water harvesting structures, water supply, irrigation, power, communications, ports, land utilization pattern, infrastructure, urbanization, environment data, forests, sanctuaries, mining areas, industries. They also include social infrastructure like the locations of schools, health centres, institutions, aganwadies, local government infrastructure etc. The geospatial database of nagar-palikas includes properties and amenities captured on city and town planning maps with 1000 GIS layers. Similar work for villages has been initiated as a pilot project.

The applications of space technology and geo-informatics have been operational in almost all the development sectors of the state. Remote sensing and GIS applications have provided impetus to planning and developmental activities at grass root level as well as monitoring and management in various disciplines.

The GIS based Applications Development

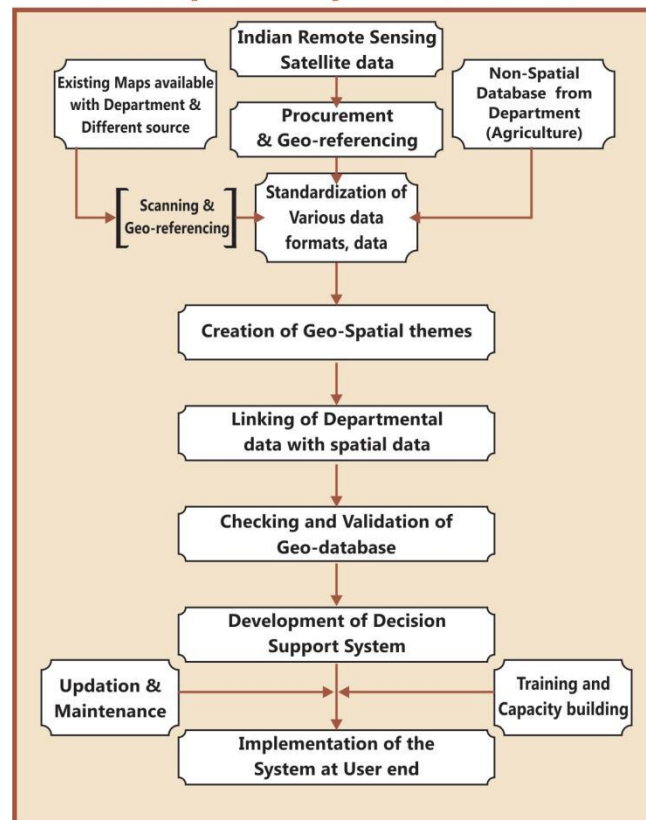
The GIS software is a powerful tool to handle, manipulate and integrate both the spatial and non-spatial data. The GIS system operates on the powerful backend data base and Sequential Query Language (SQL) to inquiry the data bases. It has the capability to handle large volume of data and process to yield values of parameters which can be input to very important

government activity as Decision Support System (DSS). Its mapping capabilities help the users and specialists in generating single and multi-theme wise maps.

The GIS based applications development has been institutionalized in BISAG- N. This process can be listed as (Refer Figure for Details)

- Making the users aware of the GIS capabilities through introductory training programme and by exposing to already developed projects as success stories.
- Helping the users in defining the GIS based projects.
- Digitizing the data available with the users and encouraging them to collect any additional data as may be required.
- Generating the appropriate data bases with the full involvement of the users following the data bases standards

Concept of Departmental GIS



Remote Sensing and GIS Sectoral Applications:

Geo-informatics based Irrigation Management and Monitoring System

- The Geo-spatial information system for Irrigation water Management and Monitoring system for command areas in Sardar Sarovar Narmada Nigam Limited (SSNL) has

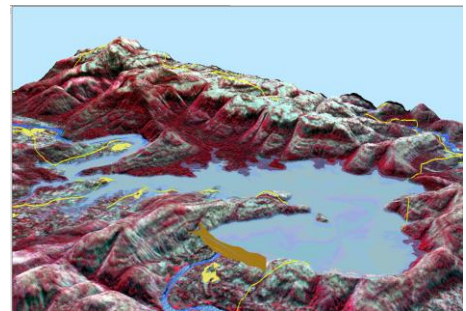


been developed. Satellite image-based Irrigation monitoring system has been developed in GIS. From the multi-spectral Satellite images of every month, the irrigated areas were extracted.

- The irrigated area were overlaid on the geo-referenced cadastral maps and the statistics of area irrigated has been estimated.
- The user friendly Customized Decision Support System (DSS) has been developed.

Preparation of DPR of Par–Tapi-Narmada Link using Geo-informatics for National Water development Agency (NWDA)

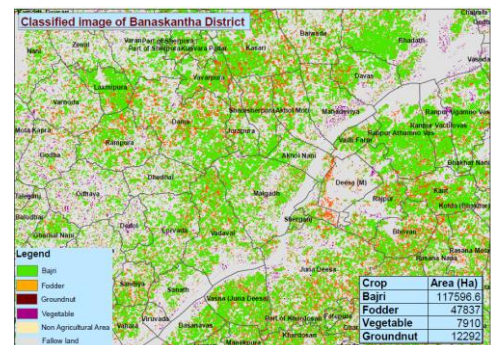
- The main objective of Par–Tapi-Narmada Link project is to divert surplus water available in west flowing rivers of south Gujarat and Maharashtra for utilization in the drought prone Saurashtra and Kachcha. On the request from NDWA, preparation of various maps for proposed DPR work was undertaken by the BISAG- N. Land use and submergence maps of proposed dams along with its statistics have been prepared by the BISAG- N. The detailed work consisted of generation of Digital Elevation Model (DEM), contour generation, Land use mapping, forest area generation of submergence extent at different levels etc.



Agriculture

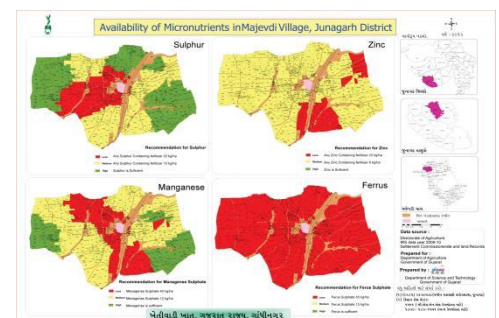
District and Village-level Crop Inventory

- Remote Sensing (RS) based Village-level Crop Acreage Estimation was taken up in two villages of Anand and Mehsana districts of Gujarat state. The major objective of this study was to attempt village-level crop inventory during two crop seasons of Kharif (monsoon season) and Rabi (winter season) using single-date Indian Remote Sensing (IRS) LISS-III and LISS-IV digital data of maximum vegetative growth stage of major crops during each season.
- District-level crop acreage estimation during three cropping seasons namely Kharif, Rabi and Zaid (summer) seasons was also carried out in all the 26-districts of Gujarat State. Summer crop acreage estimation Gujarat State was carried out during 2012.



Spatial Variability Mapping of Soil Micro-Nutrients

- The spatial variability of soil micro-nutrients like Fe, Mn, Zn and Cu in various villages of different districts, Gujarat state was mapped using geo-informatics technology. The major objectives of this study were i) to quantify the variability of Mn, Fe, Cu and Zn concentration in soil; ii) to map the pattern of micro-nutrient variability in cadastral maps, iii) suggest proper application of micro-nutrients based on status of deficiency for proper crop management and iv) preparation of village-level atlases showing spatial variability of micro-nutrients.



Geo-spatial Information System for Coastal Districts of Gujarat

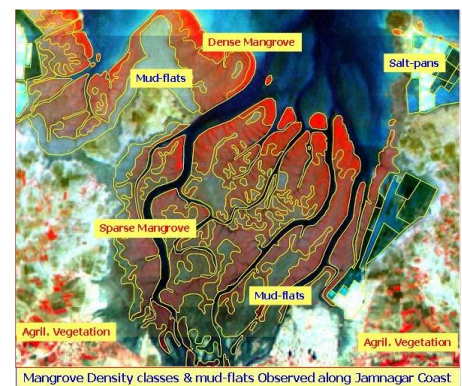
- The project on development of Village-level Geo-spatial Information System for Shrimp Farms in Coastal Districts of Gujarat, was taken with major objective of development of Village-level Geo-spatial Information System for Shrimp/Scampi areas using Remote Sensing (RS) and GIS. This project was sponsored by the Marine Products Export Development Authority (MPEDA), Ministry of Commerce & Industry, Government of India for scientific management of Scampi farms in the coastal districts which can help fishermen to better their livelihood and increase the economic condition on sustainable basis. The customized query shell was developed using the open source software for sharing the information amongst the officers from MPEDA and potential users. This has helped the farmers to plan their processing and marketing operations so as to achieve better remunerations.



Environment and Forest

Mapping and Monitoring of Mangroves in the Coastal Districts of Gujarat State

- Gujarat Ecology Commission, with technical inputs from the Bhaskaracharya National Institute for Space Applications and Geo-informatics - N (BISAG- N) made an attempt to publish Mangrove Atlas of the Gujarat state. Mangrove atlas for 13-coastal districts with 35-coastal talukas in Gujarat, have been prepared using Indian Remote sensing satellite images. The comparison of mangrove area estimates carried out by BISAG- N and Forest Survey of India (FSI) indicates a net increase in the area under mangrove cover. The present assessment by BISAG- N, has recorded 996.3 sq. km under mangrove cover, showing a steep rise to the tune of 88.03 sq. km. In addition to the existing Mangrove cover, the present assessment also gives the availability of potential area of 1153 sq. km, where mangrove regeneration program can be taken up.



Academy of Geo-informatics



for Sustainable Development

Introduction

- Considering the requirement of high end research and development in the areas having relevance of geo-informatics technology for sustainable development, a separate infrastructure has been established. In collaboration with different institutes in the state as well as in the country, R&D activities are being carried out in the areas of climate change, environment, disaster management, natural resources management, infrastructure development, resources planning, coastal hazard and coastal zone management studies, etc. under the guidance of eminent scientists.
- Various innovative methodologies/models developed in this academy through the research process have helped in development of various applications. There are plans to enhance R&D activities manifold during coming years.
- This unit also provides training to more than 600 students every year in the field of Geo-informatics to the students from various backgrounds like water resources, urban planning, computer Engineering, IT, Agriculture in the areas of Remote sensing, GIS and their applications.
- This Academy has been established as a separate infrastructure for advanced research and development through following schools:
 - School of Geo-informatics
 - School of Climate & Environment
 - School of Integrated Coastal Zone Management
 - School of Sustainable Development Studies
 - School of Natural Resources and Bio-diversity
 - School of Information Management of Disasters
 - School of Communication and Society



During XIIth Five year Plan advance applied research through above schools shall be the main thrust area. Already M. Tech and Ph.D. students of other Universities/ Institutes are doing research in this academy in applied sciences under various collaborative programmes.

M. Tech. Students' Research Programme

The academy started M. Tech. students' research programme in a systematic way. It admitted 11 students from various colleges and universities in Gujarat, Rajasthan and Madhya Pradesh for period of 10 months from August 2011 to May 2012. All the students were paid stipend of Rs. 6000 per month during the tenure. The research covered the following areas:

- Cloud computing techniques
- Mobile communication
- Design of embedded systems
- Aquifer modelling
- Agricultural and Soils Remote Sensing
- Digital Image processing Techniques (Data Fusion and Image Classification).

The research resulted in various dissertations and publications in national and international journals.

- Now nine students, one from IIT, Kharagpur, three from GTU, one from M. S University, Vadodara and four from GU, are undergoing their Ph. D programme. Out of nine, two thesis have been submitted. Two students are from abroad. One each from Vietnam and Yemen. Since then (after approval of research programme from the Governing Body), 200+ papers have been published by the Academy.

CANDIDATE'S DECLARATION

We declare that 8th semester internship project report entitled “ **Fake News Analysis** ” is our own work conducted under the supervision of the external guide **Punit Lalwani** from BISAG-N (**Bhaskaracharya National Institute for Space Applications & Geo-informatics**). We further declare that to the best of our knowledge the report for this project does not contain any part of the work which has been submitted previously for such project either in this or any other institutions without proper citation.

Candidate 1's Signature

Shekh Mohammad Ayan

Student ID: O3

Candidate 2's Signature

Krishna Zala

Student ID: O3

Candidate 3's Signature

Patel Dhyan

Student ID: O3

Submitted To:

L.J. Institute of Engineering And Technology –Ahmedabad.

ACKNOWLEDGMENT

We are grateful to **T.P. Singh**, Director General (BISAG-N) for giving us this opportunity to work the guidance of renowned people of the field of MIS Based Portal also providing us with the required resources in the company.

We would like to express our endless thanks to our external guide **Mr. Punit Lalwani**, And Admin Department **Mr. Sidhharth Patel** at Bhaskaracharya National Institute of Space Application and Geo-informatics for their sincere and dedicated guidance throughout the project development.

Also, our hearty gratitude to our Head of Department, **Prof. Shewta Yagnik** and our internal guide **Prof. Hardik Joshi, Prof. Jenish Shah, Prof. Bhargav Suthar** for giving us encouragement and technical support on the project.

Shekh Mohammad Ayan

Student ID: O3

Krishna Zala

Student ID: O3

Patel Dhyan

Student ID: O3

Index

1. Introduction.....	2
✓ Project Profile	2
✓ Project Purpose	2
✓ Project Scope	3
✓ Project Abstract	4
✓ Literature Review.....	5
2. Details of Library Used.....	6
✓ Introduction to Pandas	6
✓ Introduction to Sklearn	10
✓ Introduction to NLTK.....	14
3. System Analysis.....	19
✓ Existing System	19
✓ Problem Identification	19
✓ System Requirement	20
4. System Design	21
✓ Flow Diagram.....	21
✓ Architecture Diagram.....	29
5. Screenshots	43
6. Future Enhancement	51
7. Conclusion.....	52
8. References.....	53

1. Introduction

- **Project Profile**

Our project is an 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 model trained by our own. The accuracy of the model is around 95% for all the selfmade model. This model can detect all Real and Fake news and message which are related to covid-19, political news, geology ,etc.

- **Project Purpose**

Internet is one of the important inventions and a large number of persons are its users. These persons use this for different purposes. There are different social media platforms that are accessible to these users. Any user can make a post or spread the news through these online platforms. These platforms do not verify the users or their posts. So some of the users try to spread fake news through these platforms. These fake news can be a propaganda against an individual, society, organization or political party. A human being is unable to detect all these fake news. So there is a need for machine learning classifiers that can detect these fake news automatically. Use of machine learning classifiers for detecting the fake news is described in this systematic literature review.

Different researchers are working for the detection of fake news. The use of Machine learning is proving helpful in this regard. Researchers are using different algorithms to detect the false news. Fake news detection is big challenge. We have used the machine learning for detecting fake news. Fake news are increasing with the passage of time. That is why there is a need to detect fake news. The algorithms of machine learning are trained to fulfill this purpose. Machine learning algorithms will detect the fake news automatically once they have trained.

- **Project Scope**

Due to increasing use of internet, it is now easy to spread fake news. A huge number of persons are regularly connected with internet and social media platforms. There is no any restriction while posting any news on these platforms. So some of the people takes the advantage of these platforms and start spreading fake news against the individuals or organizations. This can destroy the reput of an individual or can affect a business. Through fake news, the opinions of the people can also be changed for a political party. There is a need for a way to detect these fake news. Machine learning classifiers are using for different purposes and these can also be used for detecting the fake news. The classifiers are first trained with a data set called training data set. After that, these classifiers can automatically detect fake news

The Project is all about detection of fake news using machine learning algorithms which helps many people to distinguish between which news is real and which is actually fake.

The scope of project is very vast and in future there are many chances that such fake news detection will be used by most of the people around the globe as data as escalating day by day which gives the opportunity to the fakesters to spread more and more fake news and earn by doing that.

- **Project Abstract**

In our modern era where the internet is ubiquitous, everyone relies on various online resources for news. Along with the increase in the use of social media platforms like Facebook, Twitter, etc. news spread rapidly among millions of users within a very short span of time. The spread of fake news has far-reaching consequences like the creation of biased opinions to swaying election outcomes for the benefit of certain candidates. Moreover, spammers use appealing news headlines to generate revenue using advertisements via click-baits. In this project, we aim to perform binary classification of various news articles available online with the help of concepts pertaining to Artificial Intelligence, Natural Language Processing and Machine Learning. We aim to provide the user with the ability to classify the news as fake or real.

● **Literature Review**

The system is an 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,after that it gives the reality of it.

When user paste the news our machine learning trained model works in the background and gives the output.

On clicking on the prediction button after entering the news user can check whether the news is real or fake.

2. Details of libraries

- **Introduction to Pandas**

Pandas library must be used in the life cycle of python. It is very popular and widely used along with numpy and matplotlib. It is used for data analysis and cleaning. Pandas provide fast and flexible datastructure such as Dataframe, which are designed to work with structure data very easily and intuitively.

Pandas is open source library built on top of numpy.

- It allows fast data cleaning, preparation and analysis.
- It excels in performance and productivity.
- It also has built-in visualization features.
- It can work with the data from wide variety of source.

We can see the application of python in many fields such as -Economics, Recommendation system – spotify, Netflix and Amazon , stock Prediction, Neuro science , Statistics, Advertising, Analytics , Natural Language processing.

Data can be analysed in pandas in two ways –

1. Data frames - In this , data is two dimensional and consist of multiple series. Data is always represented in rectangular table.
 2. Series - In this , Data is one dimensional and consist of single list with index.
- To import pandas we need to install it first using ‘pip install pandas’.
 - To use all the methods and features of pandas import it using ‘ import pandas as pd’, for easy use name of pandas is given as pd.
 - Main application of pandas is read and analysis files.

This all are main methods of pandas library :

1. For input output purpose, `pandas.read_pickle.` `pandas.read_table.`
`pandas.read_csv.``pandas.read_fwf.``pandas.read_clipboard.`
`pandas.read_excel.`
2. For creating series and manipulating them, `pandas.Series.` `pandas.Series.T.`
`pandas.Series.array.` `pandas.Series.at.` `pandas.Series.attrs.`
3. Some of general functions , `pandas.melt.` `pandas.pivot.` `pandas.pivot_table.`
`pandas.crosstab.` `pandas.cut.` `pandas.qcut.`

Here, `df.user id` means `userid` is column of dataset and is want only user id column columns in result.

If want `userid` and `rating` columns only then use `df[['userid', 'rating']]` .

`df.language=='python'` it means only language column and in language column only the columns having python value it is return as true and otherwise false. •

`df[df.language=='python']['rating']` means display only rating column but that records only in which value of language column is python.

`read-` is used to read file using pandas.

`Head-` used to display no. Of rows of data

`Shape-`used to display dimensions of data

`Info-` if want to know some more information about our dataset.

`Describe-` we can get various information about the numerical columns in our dataframe.

Pandas Series is a one-dimensional labeled array capable of holding data of any type (integer, string, float, python objects, etc.). The axis labels are collectively called index. Pandas Series is nothing but a column in an excel sheet.

create a series by calling `pandas.Series()`. A list, numpy array, dict can be turned into a pandas series. You should use the simplest data structure that meets your needs.

Pandas DataFrame is two-dimensional size-mutable, potentially heterogeneous tabular data structure with labeled axes (rows and columns). A Data frame is a two-dimensional data structure, i.e., data is aligned in a tabular fashion in rows and columns.

DataFrame make the tabular format rows and column using dictionary object.

```
In [1]: import pandas as pd
df=pd.read_csv('F:\language_data.csv')
df.shape
```

```
Out[1]: (20, 3)
```

```
In [19]: df.userid
```

```
Out[19]: 0      1.0
1      2.0
2      3.0
3      4.0
4      5.0
5      NaN
6      7.0
7      8.0
8      9.0
9     10.0
10     NaN
11     12.0
12     13.0
13     NaN
14     15.0
15     16.0
16     17.0
17     18.0
18     19.0
19     20.0
Name: userid, dtype: float64
```

Pandas library-1

```
In [6]: df[df.language=='python']['rating']
```

```
Out[6]: 0      4
        5      3
        10     3
        12     2
        15     3
        19     3
        Name: rating, dtype: int64
```

```
In [6]: import pandas as pd
        a=[1,2,3,4,5,6]
        b=[5,6,8,0,8,9]
        a1=pd.Series(a)
        b1=pd.Series(b)
        df=pd.DataFrame(a,b)
        df
```

```
Out[6]: 0
        5  1
        6  2
        8  3
        0  4
        8  5
        9  6
```

Pandas library-2

```
In [3]: df[['userid', 'rating']]
```

```
Out[3]:
```

	userid	rating
0	1	4
1	2	3
2	3	2
3	4	1
4	5	5
5	6	3

Pandas library-3

Features of Pandas Library:

1. Eloquent syntax and rich functionalities that gives you the freedom to deal with missing data
 2. Enables you to create your own function and run it across a series of data
- High-level abstraction
3. Contains high-level data structures and manipulation tools

- **Introduction to Sklearn**

Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib.

Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python. This library, which is largely written in Python, is built upon **NumPy**, **SciPy** and **Matplotlib**

Features

Rather than focusing on loading, manipulating and summarising data, Scikit-learn library is focused on modeling the data. Some of the most popular groups of models provided by Sklearn are as follows –

Supervised Learning algorithms – Almost all the popular supervised learning algorithms, like Linear Regression, Support Vector Machine (SVM), Decision Tree etc., are the part of scikit-learn.

Unsupervised Learning algorithms – On the other hand, it also has all the popular unsupervised learning algorithms from clustering, factor analysis, PCA (Principal Component Analysis) to unsupervised neural networks.

Clustering – This model is used for grouping unlabeled data.

Cross Validation – It is used to check the accuracy of supervised models on unseen data.

Dimensionality Reduction – It is used for reducing the number of attributes in data which can be further used for summarisation, visualisation and feature selection.

Ensemble methods – As name suggest, it is used for combining the predictions of multiple supervised models.

Feature extraction – It is used to extract the features from data to define the attributes in image and text data.

Feature selection – It is used to identify useful attributes to create supervised models.

Open Source – It is open source library and also commercially usable under BSD license.

In Machine Learning, vectorization is **a step in feature extraction**. The idea is to get some distinct features out of the text for the model to train on, by converting text to numerical vectors

1. Count Vectorizer
2. TF-IDF Vectorizer

Countvectorizer is a method to convert text to numerical data. To show you how it works let's take an example:

```
text = ['Hello my name is james, this is my python notebook']
```

The text is transformed to a sparse matrix as shown below.

	hello	is	james	my	name	notebook	python	this
0	1	2	1	2	1	1	1	1

Natural Language Processing (NLP) is a sub-field of artificial intelligence that deals understanding and processing human language. In light of new advancements in machine learning, many organizations have begun applying natural language processing for translation, chatbots and candidate filtering.

TF-IDF is an abbreviation for Term Frequency Inverse Document Frequency. This is very common algorithm to transform text into a meaningful representation of numbers which is used to fit machine algorithm for prediction. Let's take sample example and explore two different spicy sparse matrix before go into deep explanation . It gives overall view what i am trying to explain below .Simple basic example data :

Train Document Set:

d1: The sky is blue.

d2: The sun is bright.

Test Document Set:

d3: The sun **in** the sky is bright.

d4: We can see the shining sun, the bright sun.

Python Code:

```
from sklearn.feature_extraction.text import
TfidfVectorizer,CountVectorizer
import pandas as pd

# set of documents

train = ['The sky is blue.','The sun is bright.']
test = ['The sun in the sky is bright', 'We can see the shining sun,
the bright sun.']

# instantiate the vectorizer object

countvectorizer = CountVectorizer(analyzer= 'word',
stop_words='english')
tfidfvectorizer = TfidfVectorizer(analyzer='word',stop_words=
'english')

# convert th documents into a matrix

count_wm = countvectorizer.fit_transform(train)
tfidf_wm = tfidfvectorizer.fit_transform(train)

#retrieve the terms found in the corpora
# if we take same parameters on both Classes(CountVectorizer and
TfidfVectorizer) , it will give same output of get_feature_names()
methods)
```

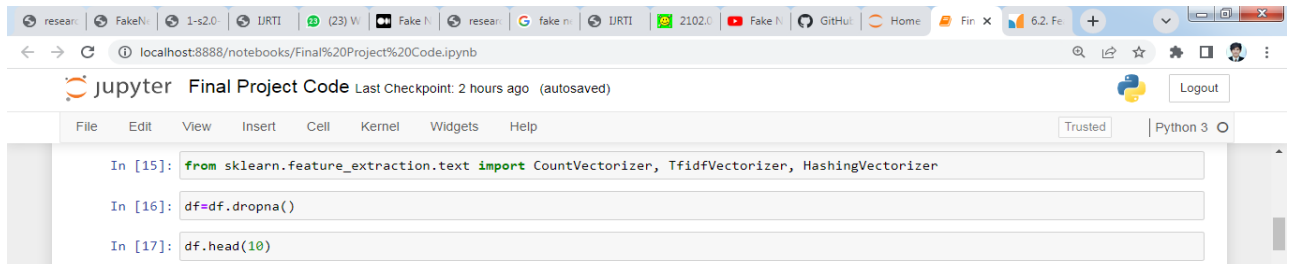
Output:

Count Vectorizer

	blue	bright	sky	sun
Doc1	1	0	1	0
Doc2	0	1	0	1

TD-IDF Vectorizer

	blue	bright	sky	sun
Doc1	0.707107	0.000000	0.707107	0.000000
Doc2	0.000000	0.707107	0.000000	0.707107



```
In [15]: from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer, HashingVectorizer

In [16]: df=df.dropna()

In [17]: df.head(10)
```

TF-IDF is better than Count Vectorizers because it not only focuses on the frequency of words present in the corpus but also provides the importance of the words. We can then remove the words that are less important for analysis, hence making the model building less complex by reducing the input dimensions

- **Introduction to NLTK**

NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active discussion forum.

Thanks to a hands-on guide introducing programming fundamentals alongside topics in computational linguistics, plus comprehensive API documentation, NLTK is suitable for linguists, engineers, students, educators, researchers, and industry users alike. NLTK is available for Windows, Mac OS X, and Linux. Best of all, NLTK is a free, open source, community-driven project.

NLTK has been called “a wonderful tool for teaching, and working in, computational linguistics using Python,” and “an amazing library to play with natural language.”

Natural Language Processing with Python provides a practical introduction to programming for language processing. Written by the creators of NLTK, it guides the reader through the fundamentals of writing Python programs, working with corpora, categorizing text, analyzing linguistic structure, and more. The online version of the book has been updated for Python 3 and NLTK 3.

Text Preprocessing

For all the below functions, you can use libraries like ‘nltk, re, etc.’

Tokenization: First we create tokens of the text data.

Removing Stopwords: Remove the stopwords like “a, an, the, then, etc.”

Because we know that these words have no or very less impact on whether a news is fake or not.

Stemming: Converting the word into its base form, for example: converting ‘worked’ to its base form which is ‘work’.

Tokenizing

By **tokenizing**, you can conveniently split up text by word or by sentence. This will allow you to work with smaller pieces of text that are still relatively coherent and meaningful even outside of the context of the rest of the text. It’s

your first step in turning unstructured data into structured data, which is easier to analyze.

When you're analyzing text, you'll be tokenizing by word and tokenizing by sentence. Here's what both types of tokenization bring to the table:

Tokenizing by word: Words are like the atoms of natural language. They're the smallest unit of meaning that still makes sense on its own. Tokenizing your text by word allows you to identify words that come up particularly often. For example, if you were analyzing a group of job ads, then you might find that the word "Python" comes up often.

```
word_tokenize(example_string)
```

```
["Muad'Dib", 'learned', 'rapidly', 'because', 'his', 'first', 'training', 'was', 'in', 'how',  
'to', 'learn', '.', 'And', 'the', 'first', 'lesson', 'of', 'all', 'was', 'the', 'basic', 'trust', 'that',  
'he', 'could', 'learn', '.', 'It', '"s", 'shocking', 'to', 'find', 'how', 'many', 'people', 'do',  
'not', 'believe', 'they', 'can', 'learn', ',', 'and', 'how', 'many', 'more', 'believe',  
'learning', 'to', 'be', 'difficult', '.']
```

Filtering Stop Words

Stopwords are **the English words which does not add much meaning to a sentence**. They can safely be ignored without sacrificing the meaning of the

sentence. For example, the words like the, he, have etc. Such words are already captured this in corpus named corpus. We first download it to our python environment

Stop words are words that you want to ignore, so you filter them out of your text when you're processing it. Very common words like 'in', 'is', and 'an' are often used as stop words since they don't add a lot of meaning to a text in and of themselves.

Here's how to import the relevant parts of NLTK in order to filter out stop words:

```
>>> nltk.download("stopwords">>>> from nltk.corpus import stopwords>>>
from nltk.tokenize import word_tokenize
```

```
In [37]: import nltk
from nltk.corpus import stopwords
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
n't", 'wasn', "wasn't", 'weren', "weren't", 'won', "won't", 'wouldn', "wouldn't"]
```

Stemming is the process of producing morphological variants of a root/base word. Stemming programs are commonly referred to as stemming algorithms or stemmers. A stemming algorithm reduces the words “chocolates”, “chocolatey”, “choco” to the root word, “chocolate” and “retrieval”, “retrieved”, “retrieves” reduce to the stem “retrieve”.

Some more example of stemming for root word "like" include:

-> "likes"
-> "liked"
-> "likely"

Applications of stemming are:

- Stemming is used in information retrieval systems like search engines.
- It is used to determine domain vocabularies in domain analysis.

```
# import these modules  
from nltk.stem import PorterStemmer  
from nltk.tokenize import word_tokenize
```

```
ps = PorterStemmer()
```

```
# choose some words to be stemmed  
words = ["program", "programs", "programmer", "programming",  
"programmers"]
```

```
for w in words:  
    print(w, " : ", ps.stem(w))
```

Output:

```
program : program  
programs : program  
programmer : program  
programming : program  
programmers : program
```

3. System Analysis

- **Existing System**

1. Our project of Fake news analysis can work in any computer with minimum specification.
2. The analysis process takes less than a moment and this is very beneficial for the people.
3. The first thing to start with is Fake news. With the help of Fake news analysis, it will be easy to identify the news and check whether the news is fake or real.
4. This project is made to reach each and everyone in the society who are getting wrong influence by the fakesters and specially for the person who blindly believe any news without knowing that whether it is fake or real.

- **Problem Identification**

Internet is one of the important inventions and a large number of persons are its users. These persons use this for different purposes. There are different social media platforms that are accessible to these users. Any user can make a post or spread the news through these online platforms. These platforms do not verify the users or their posts. So some of the users try to spread fake news through these platforms. These fake news can be a propaganda against an individual,

society, organization or political party. A human being is unable to detect all these fake news. So there is a need for machine learning classifiers that can detect these fake news automatically. Use of machine learning classifiers for detecting the fake news is described in this project.

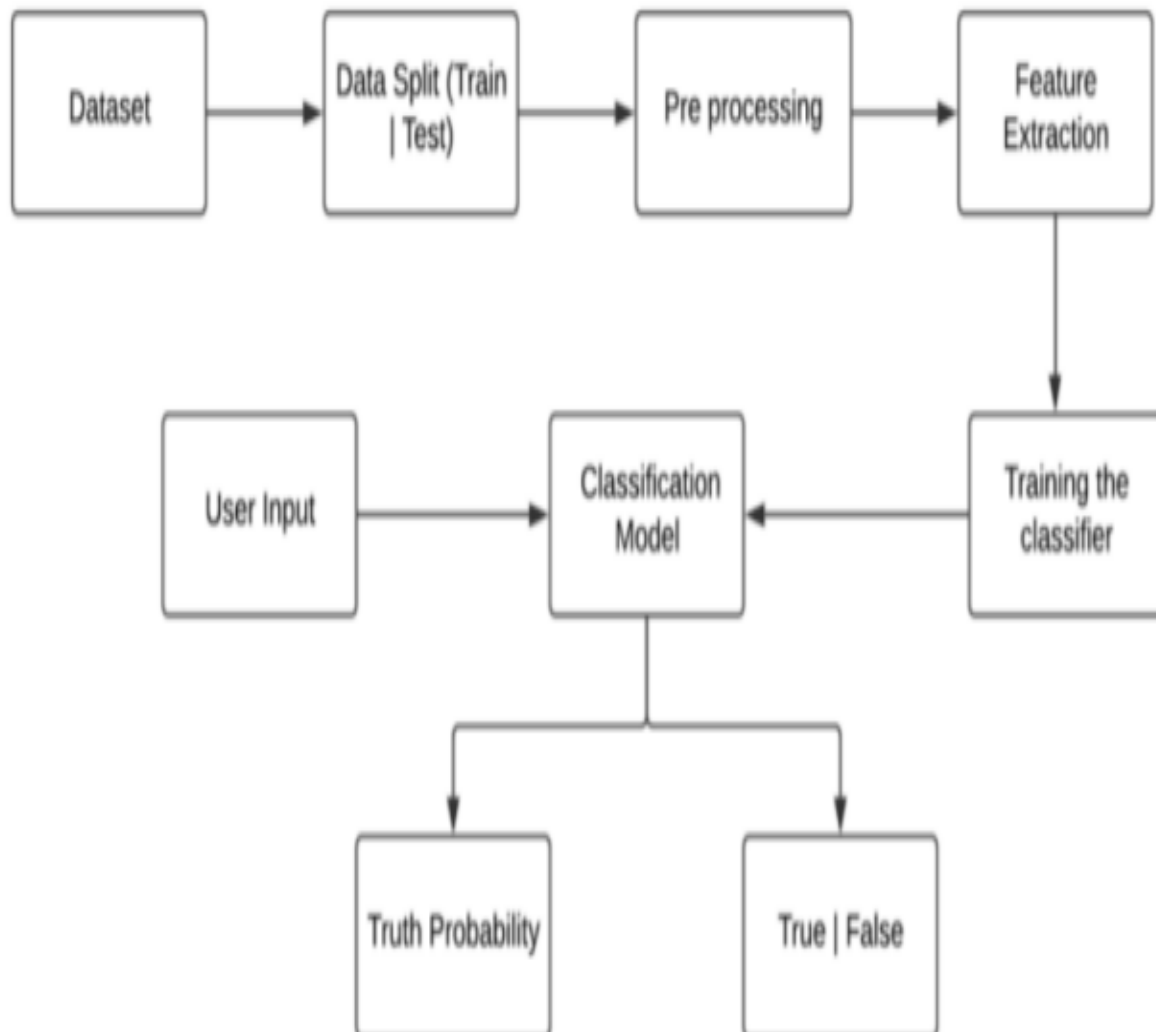
- **System Requirement**

Hardware and Software Requirement

- Modern Operating System:
- Windows 7 or 10
- Mac OS X 10.11 or higher, 64-bit
- Linux: RHEL 6/7, 64-bit (almost all libraries also work in Ubuntu)
- x86 64-bit CPU (Intel / AMD architecture)
- 4 GB RAM
- 5 GB free disk space

4. System Design

- **Flow Diagram**



- The way this Fake News Analysis works is displayed in this flow chart.
- The Process is discussed below in detail.

1. Dataset.

To create a machine learning model, the first thing we required is a dataset as a machine learning model completely works on data. The collected data for a particular problem in a proper format is known as the **dataset**.

Dataset may be of different formats for different purposes, such as, if we want to create a machine learning model for business purpose, then dataset will be different with the dataset required for a liver patient. So each dataset is different from another dataset. To use the dataset in our code, we usually put it into a CSV **file**. However, sometimes, we may also need to use an HTML or xlsx file.

The rapid increase in fake news, which causes significant damage to society, triggers many fake news related studies, including the development of fake news detection.. The resources for these are mainly available as public datasets taken from kaggle. We surveyed 4-5 datasets related to fake news getting the proper and reliable dataset is very important to create any machine learning model. There are many datasets available on internet on well known platforms such as kaggle. we studied 4-5 datasets in detail however the number of rows found in that dataset was rarely 5 to 6 thousands due to that when model is trained with such dataset it gives the accuracy of 1 which for any machine learning model is rarely possible. so dataset played an important role in building this model.

2. Train-Test Split

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.

The train-test split is a technique for evaluating the performance of a machine learning algorithm.

It can be used for classification or regression problems and can be used for any supervised learning algorithm.

The procedure involves taking a dataset and dividing it into two subsets. The first subset is used to fit the model and is referred to as the training dataset. The second subset is not used to train the model; instead, the input element of the dataset is provided to the model, then predictions are made and compared to the expected values. This second dataset is referred to as the test dataset.

➤ **Train Dataset:** Used to fit the machine learning model.

➤ **Test Dataset:** Used to evaluate the fit machine learning model.

The objective is to estimate the performance of the machine learning model on new data: data not used to train the model.

This is how we expect to use the model in practice. Namely, to fit it on available data with known inputs and outputs, then make predictions on new examples in the future where we do not have the expected output or target values.

The train-test procedure is appropriate when there is a sufficiently large dataset available.

3. Data preprocessing

Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model.

When creating a machine learning project, it is not always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way. So for this, we use data preprocessing task.

A real-world data generally contains noises, missing values, and maybe in an unusable format which cannot be directly used for machine learning models. Data preprocessing is required tasks for cleaning the data and making it suitable

for a machine learning model which also increases the accuracy and efficiency of a machine learning model.

It involves below steps:

- **Getting the dataset**
- **Importing libraries**
- **Importing datasets**
- **Finding Missing Data**
- **Encoding Categorical Data**
- **Splitting dataset into training and test set**
- **Feature scaling**

4. Feature extraction

Feature extraction refers to the process of transforming raw data into numerical features that can be processed while preserving the information in the original data set. It yields better results than applying machine learning directly to the raw data.

Feature extraction can be accomplished manually or automatically:

Manual feature extraction requires identifying and describing the features that are relevant for a given problem and implementing a way to extract those features. In many situations, having a good understanding of the background or domain can help make informed decisions as to which features could be useful. Over decades of research, engineers and scientists have developed feature

extraction methods for images, signals, and text. An example of a simple feature is the mean of a window in a signal.

Automated feature extraction uses specialized algorithms or deep networks to extract features automatically from signals or images without the need for human intervention. This technique can be very useful when you want to move quickly from raw data to developing machine learning algorithms. Wavelet scattering is an example of automated feature extraction.

With the ascent of deep learning, feature extraction has been largely replaced by the first layers of deep networks – but mostly for image data. For signal and time-series applications, feature extraction remains the first challenge that requires significant expertise before one can build effective predictive models.

5. Model Training

Model training is the primary step in machine learning, resulting in a working model that can then be validated, tested and deployed. The model's performance during training will eventually determine how well it will work when it is eventually put into an application for the end-users.

Both the quality of the training data and the choice of the algorithm are central to the model training phase. In most cases, training data is split into two sets for training and then validation and testing.

The selection of the algorithm is primarily determined by the end-use case. However, there are always additional factors that need to be considered, such as algorithm-model complexity, performance, interpretability, computer resource requirements, and speed. Balancing out these various requirements can make selecting algorithms an involved and complicated process.

To train an ML model, you need to specify the following:

Input training datasource

Name of the data attribute that contains the target to be predicted

Required data transformation instructions

Training parameters to control the learning algorithm

6. Classification Model

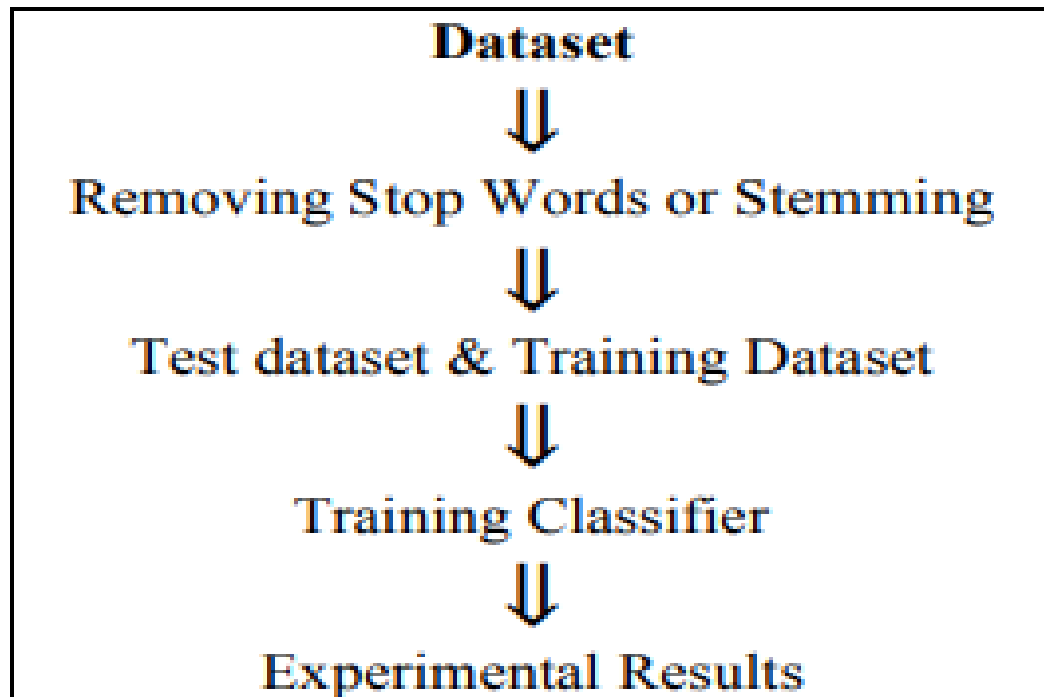
#	Model	Data pre-processing		Impact from		Highlights
		Normalization	Scaling	Collinearity	Outliers	
1	Logistic Regression	Yes	No	Yes	Yes	<ul style="list-style-type: none"> • Highly descriptive with good accuracy • Reasonable computational requirements
2	Artificial Neural Networks	No	Yes	Yes	Yes	<ul style="list-style-type: none"> • High prediction accuracy • Self-extracts features • Heavy computational requirements for large datasets
3	Random Forest	No	No	No	Yes	<ul style="list-style-type: none"> • High prediction accuracy • Provides limited explainability • Works well with both continuous & categorical predictors
4	Naïve Bayes	NA	NA	Yes	Yes	<ul style="list-style-type: none"> • Applicable to categorical predictors only • Suitable for small train data
5	KNN	Yes	Yes	Yes	Yes	<ul style="list-style-type: none"> • Performs local approximation, no prediction formula • Heavy computational requirement

7. Model Evalution

After training your model, you have to check to see how it's performing. This is done by testing the performance of the model on previously unseen data. The unseen data used is the testing set that you split our data into earlier. If testing was done on the same data which is used for training, you will not get an accurate measure, as the model is already used to the data, and finds the same patterns in it, as it previously did. This will give you disproportionately high accuracy.

In the end, you can use your model on unseen data to make predictions accurately.

Architecture Diagram



Project Steps:

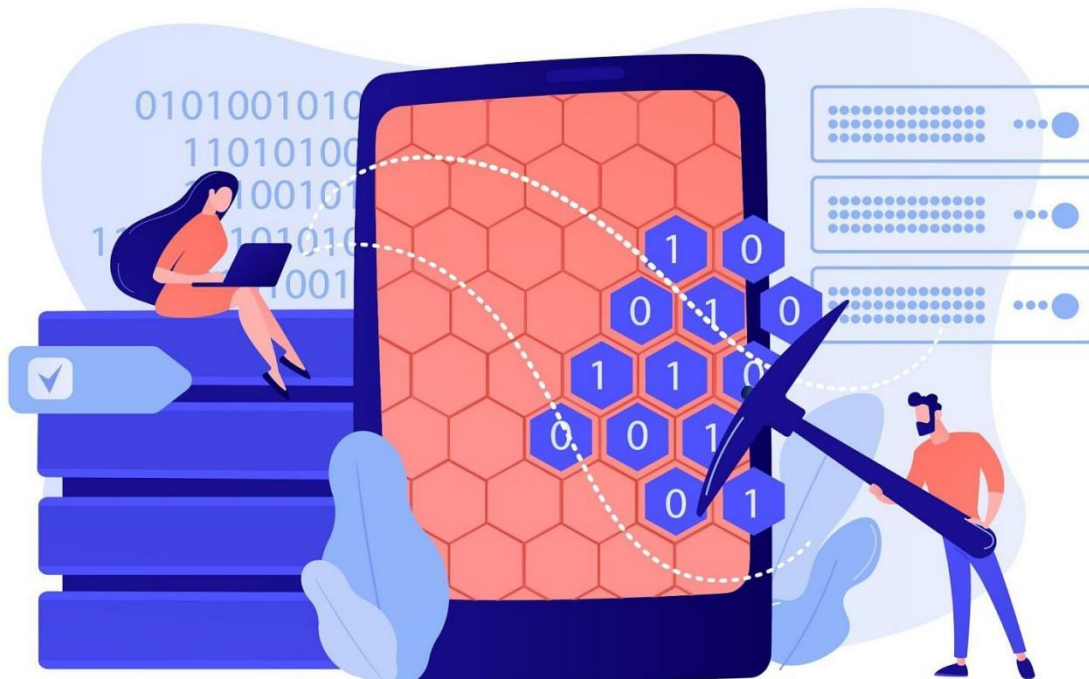
Step 1: Getting The Dataset :

To create a machine learning model, the first thing we required is a dataset as a machine learning model completely works on data. The collected data for a particular problem in a proper format is known as the dataset

We search out for many datasets that contain both real and fake news.however the best dataset we got it from kaggle.com.

This dataset consists of many rows and columns.

Mainly columns includes News,News title,Text and Its classification that the news is fake or real.



Step 2 : Importing Libraries.

Libraries are sets of routines and functions that are written in a given language. **A robust set of libraries can make it easier for developers to perform complex tasks without rewriting many lines of code.**

It is very useful for fundamental scientific computations in Machine Learning. It is particularly useful for linear algebra, Fourier transform, and random number capabilities.

Libraries that are imported during creation of this model are

- Pandas
- Numpy
- Matplotlib
- Nltk
- Re
- Sklearn
- Iter Tools



Step 3 : Importing Dataset.

Dataset which was downloaded is imported using and read with the help of python library called pandas.

Steps to import any dataset in python are:

Step 1: Capture the File Path. Firstly, capture the full path where your CSV file is stored.

Step 2: Apply the Python code. Type/copy the following code into Python, while making the necessary changes to your path.

Step 3: Run the Code.

Step 4: Finding Missing Data.

Ways to handle missing data:

There are mainly two ways to handle missing data, which are:

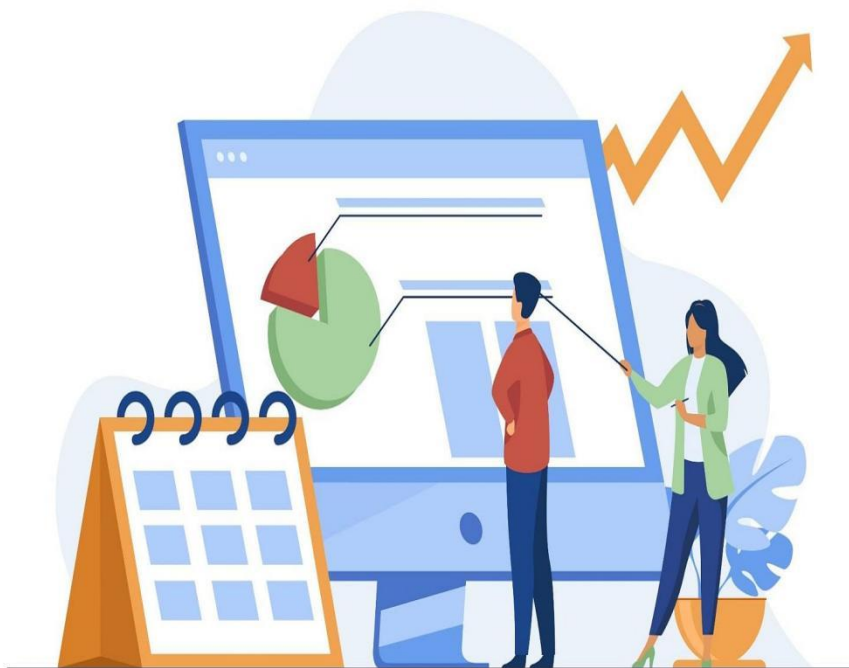
By deleting the particular row:

The first way is used to commonly deal with null values. In this way, we just delete the specific row or column which consists of null values. But this way is not so efficient and removing data may lead to loss of information which will not give the accurate output.

By calculating the mean:

In this way, we will calculate the mean of that column or row which contains any missing value and will put it on the place of missing value. This strategy is useful for the features which have numeric data such as age, salary, year, etc. Here, we will use this approach.

To handle missing values, we will use Scikit-learn library in our code, which contains various libraries for building machine learning models. Here we will use Imputer class of sklearn.preprocessing library.



Since machine learning model completely works on mathematics and numbers, but if our dataset would have a categorical variable, then it may create trouble while building the model. So it is necessary to encode these categorical variables into numbers.

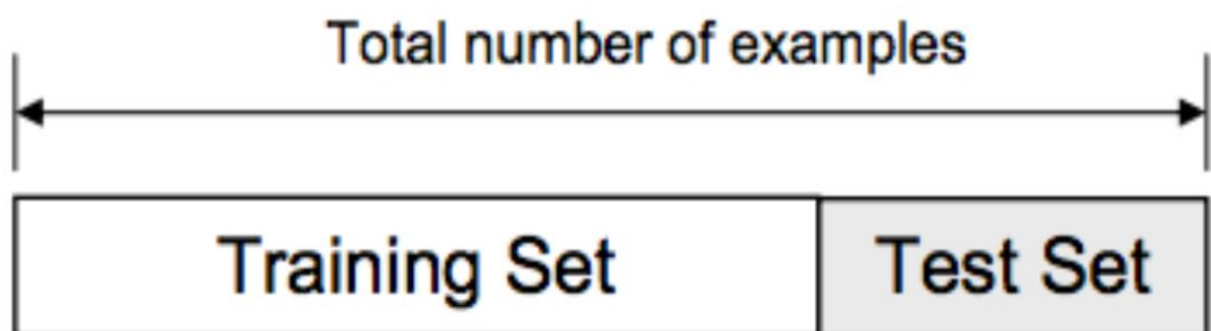
In our model we have categorical variables such as label they are fake and real. In order to get good accuracy we have to convert them into 0 and 1.

This is nothing but dealing with categorical data.

Step 5: Splitting dataset into training and test set

`train_test_split` is a function in **Sklearn model selection** for splitting data arrays into **two subsets**: for training data and for testing data. With this function, you don't need to divide the dataset manually.

By default, Sklearn **train_test_split** will make random partitions for the two subsets. However, you can also specify a random state for the operation.



Step 6 : Feature scaling

Feature scaling is a method used to normalize the range of independent variables or features of data. In data processing, it is also known as data normalization and is generally performed during the data preprocessing step. Just to give you an example — if you have multiple independent variables like age, salary, and height; With their range as (18–100 Years), (25,000–75,000 Euros), and (1–2 Meters) respectively, feature scaling would help them all to be in the same range

Now, since you have an idea of what is feature scaling. Let us explore what methods are available for doing feature scaling. Of all the methods available, the most common ones are:

Normalization

Also known as min-max scaling or min-max normalization, it is the simplest method and consists of rescaling the range of features to scale the range in [0, 1]. The general formula for normalization is given as:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

Here, $\max(x)$ and $\min(x)$ are the maximum and the minimum values of the feature respectively.

We can also do a normalization over different intervals, e.g. choosing to have the variable laying in any $[a, b]$ interval, a and b being real numbers. To rescale a range between an arbitrary set of values $[a, b]$, the formula becomes:

$$x' = a + \frac{(x - \min(x))(b - a)}{\max(x) - \min(x)}$$

Standardization

Feature standardization makes the values of each feature in the data have zero mean and unit variance. The general method of calculation is to determine the distribution mean and standard deviation for each feature and calculate the new data point by the following formula:

$$x' = \frac{x - \bar{x}}{\sigma}$$

Here, σ is the standard deviation of the feature vector, and \bar{x} is the average of the feature vector.

Scaling to unit length

The aim of this method is to scale the components of a feature vector such that the complete vector has length one. This usually means dividing each component by the Euclidean length of the vector:

$$x' = \frac{x}{\|x\|}$$

$\|x\|$ Is the Euclidean length of the Feature Vector.

In addition to the above 3 widely-used methods, there are some other methods to scale the features viz. Power Transformer, Quantile Transformer, Robust Scaler, etc. For the scope of this discussion, we are deliberately not diving into the details of these techniques.

Step 7 : Algorithm Selection

This step includes the selection of algorithm

Algorithms that are studied during the internship period includes :

Decision Tree Algorithm

Linear Regression Algorithm

Classification Algorithm

KNN Algorithm

Naive baiyes Algorithm

Multinomial Algorithm

Support vector machine

Passive regressive Classifier

Among all the algorithms we decided to choose **PASSIVE REGRESSIVE ALGORITHM**

Which worked best on our model and gave the accuracy of 95%.

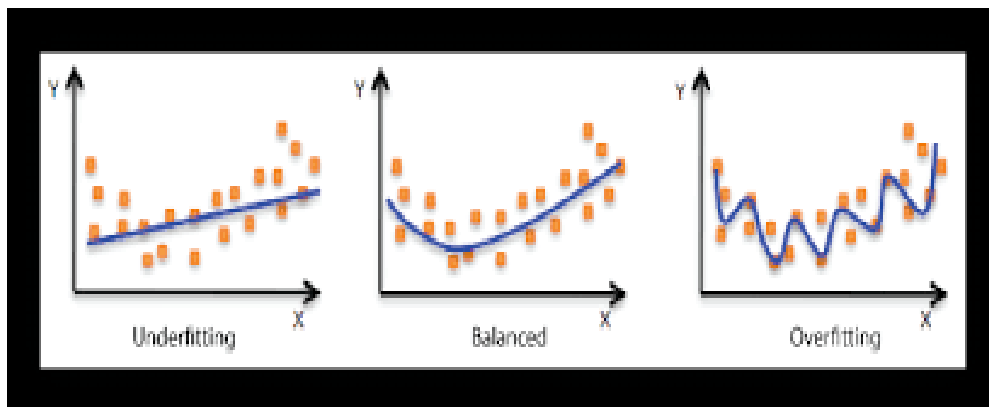
Step 8: Model Fit

Model fitting is **a measure of how well a machine learning model generalizes to similar data to that on which it was trained**. A model that is well-fitted produces more accurate outcomes

Ideally, **the case when the model makes the predictions with 0 error**, is said to have a good fit on the data. This situation is achievable at a spot between overfitting and underfitting.

Three types of fitting are there :

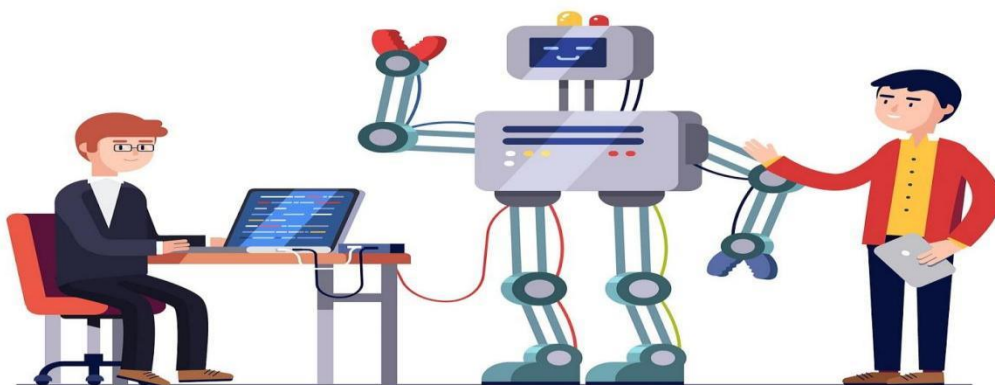
1. under fitting
2. Balanced fitting
3. Over fitting



Step 9 : Evaluating the Model:

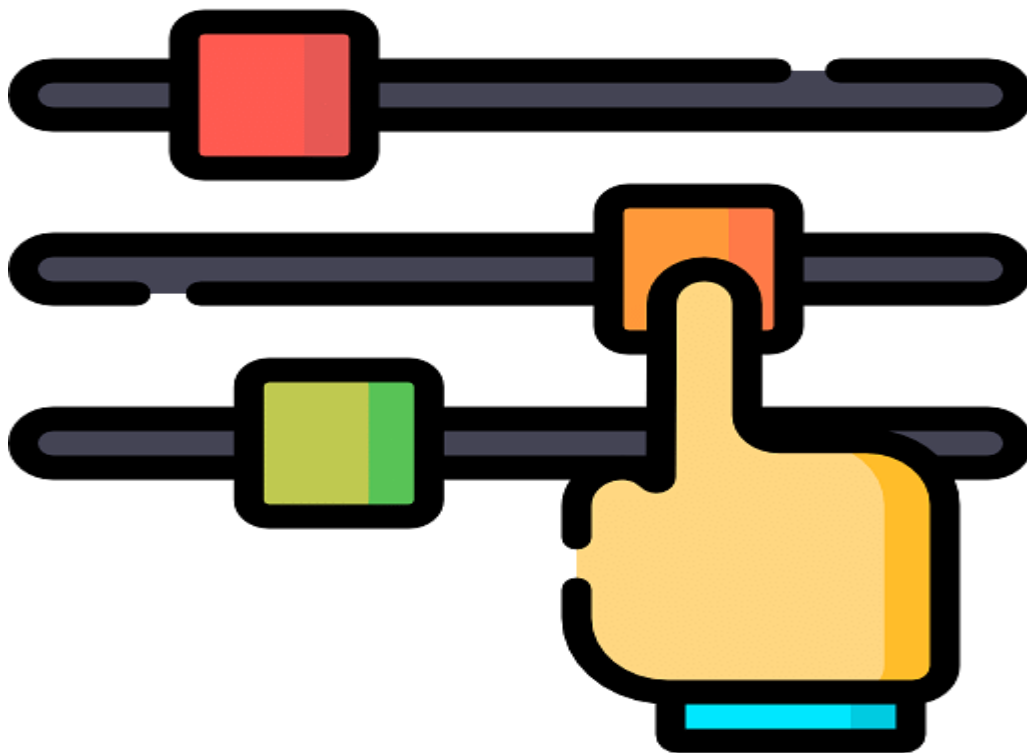
After training your model, you have to check to see how it's performing. This is done by testing the performance of the model on previously unseen data. The unseen data used is the testing set that you split our data into earlier. If testing was done on the same data which is used for training, you will not get an accurate measure, as the model is already used to the data, and finds the same patterns in it, as it previously did. This will give you disproportionately high accuracy.

When used on testing data, you get an accurate measure of how your model will perform and its speed.



Parameter Tuning:

Once you have created and evaluated your model, see if its accuracy can be improved in any way. This is done by tuning the parameters present in your model. Parameters are the variables in the model that the programmer generally decides. At a particular value of your parameter, the accuracy will be the maximum. Parameter tuning refers to finding these values.



7. Making Predictions

In the end, you can use your model on unseen data to make predictions accurately.



5. Snapshots

```
import pandas as pd
```

```
df=pd.read_csv("data.csv")  
df.head(3)
```

Unnamed: 0		title	text	subject	date	label
0	25591	U.S. State Department email restored after glo...	WASHINGTON (Reuters) - The U.S. State Departme...	politicsNews	18-Aug-17	Real
1	20371	OBAMA?S MUSLIM DHS Advisor REFUSED To Videotap...	When a Muslim man with dual citizenship in Egy...	left-news	26-Jun-16	Fake
2	7788	WATCH: Leonardo DiCaprio Gave an Oscar Accept...	He s one of the most recognizable actors of hi...	News	29-Feb-16	Fake

```
X = df.drop('label', axis=1)
y = df['label']
# Delete missing data
df = df.dropna()
df2 = df.copy()
df2.reset_index(inplace=True)
```

```
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
import re
import nltk
nltk.download('stopwords')
ps = PorterStemmer()
corpus = []
for i in range(0, len(df2)):
    review = re.sub('[^a-zA-Z]', ' ', df2['text'][i])
    review = review.lower()
    review = review.split()
```

```
review = [ps.stem(word) for word in review if not  
review = ' '.join(review)  
corpus.append(review)
```

```
from sklearn.feature_extraction.text import TfidfVectorizer  
tfidf_v = TfidfVectorizer(max_features=5000, ngram_range=(1,3))  
X = tfidf_v.fit_transform(corpus).toarray()  
y = df2['label']
```

```
from sklearn.model_selection import train_test_split  
X_train, X_test, y_train, y_test = train_test_split(X, y,  
                                                    test_size=0.2,  
                                                    random_state=0)
```

```

from sklearn.linear_model import PassiveAggressiveClassifier
from sklearn import metrics
import numpy as np
import itertools
classifier = PassiveAggressiveClassifier(max_iter=1000)
classifier.fit(X_train, y_train)
pred = classifier.predict(X_test)
score = metrics.accuracy_score(y_test, pred)
print("accuracy:  %0.3f" % score)

```

```

import matplotlib.pyplot as plt

def plot_confusion_matrix(cm, classes,
                          normalize=False,
                          title='Confusion matrix',
                          cmap=plt.cm.Blues):

    plt.imshow(cm, interpolation='nearest', cmap=cmap)
    plt.title(title)
    plt.colorbar()
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes, rotation=45)
    plt.yticks(tick_marks, classes)

```

```

if normalize:
    cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
    print("Normalized confusion matrix")
else:
    print('Confusion matrix, without normalization')

thresh = cm.max() / 2.
for i, j in itertools.product(range(cm.shape[0]),
                              range(cm.shape[1])):
    plt.text(j, i, cm[i, j],
             horizontalalignment="center",
             color="white" if cm[i, j] > thresh else "black")

```

```

plt.tight_layout()
plt.ylabel('True label')
plt.xlabel('Predicted label')
cm = metrics.confusion_matrix(y_test, pred)
plot_confusion_matrix(cm, classes=['FAKE', 'REAL'])

```

```

# Tokenization
review = re.sub('[^a-zA-Z]', ' ', df['text'][300])
review = review.lower()
review = review.split()
review = [ps.stem(word) for word in review if not
          word in stopwords.words('english')]
review = ' '.join(review)
# Vectorization
val = tfidf_v.transform([review]).toarray()
# Predict
classifier.predict(val)

```

```

import pickle
pickle.dump(classifier, open('model2.pkl', 'wb'))
pickle.dump(tfidf_v, open('tfidfvect2.pkl', 'wb'))

```



```

# Load model and vectorizer
joblib_model = pickle.load(open('model2.pkl', 'rb'))
joblib_vect = pickle.load(open('tfidfvect2.pkl', 'rb'))
val_pkl = joblib_vect.transform([review]).toarray()
joblib_model.predict(val_pkl)

```

```

from flask import Flask, render_template, request, jsonify
import nltk
import pickle
from nltk.corpus import stopwords
import re
from nltk.stem.porter import PorterStemmer
app = Flask(__name__)
ps = PorterStemmer()
# Load model and vectorizer
model = pickle.load(open('model2.pkl', 'rb'))
tfidfvect = pickle.load(open('tfidfvect2.pkl', 'rb'))
# Build functionalities
@app.route('/', methods=['GET'])
def home():

```

```

    return render_template('index.html')
def predict(text):
    review = re.sub('[^a-zA-Z]', ' ', text)
    review = review.lower()
    review = review.split()
    review = [ps.stem(word) for word in review if not word in stopwords]
    review = ' '.join(review)
    review_vect = tfidfvect.transform([review]).toarray()
    prediction = 'FAKE' if model.predict(review_vect) == 0 else 'RE
    return prediction

```

```

@app.route('/', methods=['POST'])
def webapp():
    text = request.form['text']
    prediction = predict(text)
    return render_template('index.html', text=text,
                           result=prediction)
@app.route('/predict/', methods=['GET', 'POST'])
def api():
    text = request.args.get("text")
    prediction = predict(text)
    return jsonify(prediction=prediction)
if __name__ == "__main__":
    app.run()

```

FAKE NEWS PREDICTION

[API](#) [Blog](#) [NoteBook](#) [Code Source](#)

A fake news prediction web application using Machine Learning algorithms, deployed using Django and Heroku.

Enter your text to try it.

Write your text here...

Predict

FAKE NEWS PREDICTION

[API](#) [Blog](#) [NoteBook](#) [Code Source](#)

A fake news prediction web application using Machine Learning algorithms, deployed using Django and Heroku.

Enter your text to try it.

'OBAMA?S MUSLIM DHS Advisor REFUSED To Videotape Fellow Muslim During Investigation?But Wants Every American Gun Owner To Be FORCED To Do THIS'

Predict

A fake news prediction web application using Machine Learning algorithms, deployed using Django and Heroku.

Enter your text to try it.

'OBAMA?S MUSLIM DHS Advisor REFUSED To Videotape Fellow Muslim During Investigation?But Wants Every American Gun Owner To Be FORCED To Do THIS'

Predict

Prediction : FAKE

A fake news prediction web application using Machine Learning algorithms, deployed using Django and Heroku.

Enter your text to try it.

Romania's upper house approves judiciary bill critics say is too political

Predict

Prediction : REAL

6. Future Enhancement

We're living in the age of fake news. **Fake news** consist of deliberate misinformation under the guise of being authentic news, spread via some communication channel, and produced with a particular objective like generating revenue, promoting or discrediting a public figure, a political movement, an organization, etc.

During the 2018 national elections in **Brazil**, WhatsApp was used to spread alarming amounts of misinformation, rumors and false news favoring Jair Bolsonaro. Using this technology, it was possible to exploit encrypted personal conversations and chat groups involving up to 256 people, making these chat groups much harder to spot compared to the Facebook News Feed or Google's search results.

Last year, the two main Indian political parties took these tactics to a new scale by trying to influence India's 900 million eligible voters through creating content in Facebook and spreading it on WhatsApp (both parties have been accused of spreading false or misleading information, or misrepresentation online). **India** is WhatsApp's largest market (more than 200 million Indians users), and a place where users forward more content than anywhere else in the world.

How can fake news have such an impact? The answer is in the way humans process information.

Massive amounts of data gave birth to AI systems that are already producing human-like synthetic texts, powering a new scale of disinformation operation. Based on Natural Language Processing (NLP) techniques, several lifelike text-generating systems have proliferated and they are becoming smarter every day. .

7. Conclusion

Due to increasing use of internet, it is now easy to spread fake news. A huge number of persons are regularly connected with internet and social media platforms. There is no any restriction while posting any news on these platforms. So some of the people takes the advantage of these platforms and start spreading fake news against the individuals or organizations.

This can destroy the reput of an individual or can affect a business. Through fake news, the opinions of the people can also be changed for a political party. There is a need for a way to detect these fake news. Machine learning classifiers are using for different purposes and these can also be used for detecting the fake news.

The classifiers are first trained with a data set called training data set. After that, these classifiers can automatically detect fake news. In this systematic literature review, the supervised machine learning classifiers are discussed that requires the labeled data for training.

Labeled data is not easily available that can be used for training the classifiers for detecting the fake news. In future a research can be on the use of the unsupervised machine learning classifiers for the detection of fake news

8.References

Libraries :

Pandas : <https://pandas.pydata.org/pandas-docs/stable/>

Numpy : <https://numpy.org/doc/>

Matplotlib : <https://matplotlib.org/stable/index.html>

Nltk : <https://www.nltk.org/>

Pickle : <https://docs.python.org/3/library/pickle.html>

Youtube : <https://www.youtube.com/>

Documentations :

<https://ieeexplore.ieee.org/document/8843612>

https://link.springer.com/chapter/10.1007/978-3-319-69155-8_9

<https://onlinelibrary.wiley.com/doi/10.1002/spy2.9>

<https://doi.org/10.1109/ECTICon.2018.8620051>



Report Verification Procedure

Date:25/04/2022

Project Name: Fake News Analysis

Student Name & ID: 1. Shekh Mohammad Ayan : o3

2. Zala Krishna : o3

3. Patel Dhyan : o3

Soft Copy

Hard Copy

Report Format:

Project Index:

Sign by Training Coordinator

Sign by Project Guide