

FAKE PROFILE IDENTIFICATION IN ONLINE SOCIAL NETWORK USING MACHINE LEARNING AND NLP

Submitted in partial fulfillment of the required for the award of

Bachelor of Science degree in Computer Science

By

VAISHNAVI.V

(REGISTER NO 40290106)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA

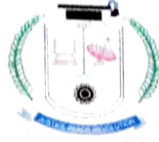
INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC|12B Status by UGC|Approved by AICTE

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI 600199

MAY 2023



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

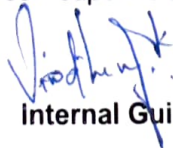
Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

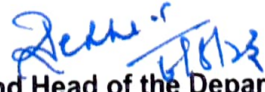
DEPARTMENT OF COMPUTER SCIENCE

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **VAISHNAVI V (40290106)** who carried out the project entitled "**FAKE PROFILE IDENTIFICATION IN ONLINE SOCIAL NETWORK USING MACHINE LEARNING AND NLP**" under our supervision from JANUARY 2023 TO MAY 2023.



Internal Guide

Ms. VINODHINI K, M.Sc., Assistant Professor


Dean and Head of the Department

Dr. REKHA CHAKRAVARTHI M.E., Ph.D.

Submitted for Viva Voce examination held on 09.05.2023


Internal Examiner


External Examiner

DECLARATION

I am, **VAISHNAVI V (40290106)** hereby declare that the Project Report entitled **"FAKE PROFILE IDENTIFICATION IN ONLINE SOCIAL NETWORK USING MACHINE LEARNING AND NLP"** done by us under the guidance of **Ms. VINODHINI K, M.Sc., Assistant Professor** Department of Computer Science at **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY,** Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai-600119 is submitted in partial fulfilment of the requirements for the award of Bachelor of Science degree in Computer Science.

DATE: 09.05.2023

PLACE: CHENNAI

VAISHNAVI V

SIGNATURE OF THE CANDIDATE

Vaishnavi

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE NO
	ABSTRACT	X
	LIST OF FIGURES	VII
	LIST OF TABLES	VIII
	LIST OF ABBREVIATIONS	IX
1	INTRODUCTION	1
	1.1 OVERVIEW OF THE PROJECT	1
	1.2 LITERATURE SURVEY	2
2	SYSTEM ANALYSIS	7
	2.1 EXISTING SYSTEM	7
	2.2 DISADVANTAGES OF EXISTING SYSTEM	7
	2.3 PROPOSED SYSTEM	8
	2.4 ADVANTAGES OF PROPOSED SYSTEM	8
	2.5 FEASIBILITY STUDY	9
	2.6 ECONOMIC STUDY	9
	2.7 TECHNICAL STUDY	9

CHAPTER	TITLE	PAGE NO
	2.8 SOCIAL FEASIBILITY	10
3	SYSTEM REQUIREMENTS	11
	3.1 SOFTWARE REQUIREMENTS	11
	3.2 HARDWARE REQUIREMENTS	11
	3.3 REQUIREMENT ANALYSIS	11
	3.4 PYTHON	11
	3.5 ANACONDA	12
	3.6 ANACONDA NAVIGATOR	13
	3.7 JUPYTER NOTEBOOK	14
4	SYSTEM ARCHITECTURE	15
	4.1 ARCHITECTURE DIAGRAM	15
	4.2 DATA FLOW DIAGRAM	16
	4.3 ENTITY RELATIONSHIP DIAGRAM	17
5	SYSTEM IMPLEMENTATION	18
	5.1 ARTIFICIAL INTELLIGENCE	18

CHAPTER	TITLE	PAGE NO
	5.2 MACHINE LEARNING	18
	5.3 NATURAL LANGUAGE PROCESSING (NLP)	20
	5.4 SUPPORT VECTOR MACHINE (SVM)	22
	5.5 NAÏVE BAYES	24
	5.6 LOGISTIC REGRESSION	25
	5.7 RANDOM FOREST	26
	5.8 CORRELATION HEATMAP	28
	5.9 CONFUSION MATRIX	29
6	CONCLUSION	31
7	APPENDIX	32
	7.1 SOURCE CODE	32
	7.2 OUTPUT	37
8	REFERENCE	38

LIST OF FIGURES

CHAPTER NO	FIGURE NAMES	PAGE NO
5	Fig 5.1 Comparing the models using Cross-Validation	28
	Fig 5.2 Correlation Heatmap Between Features	29
	Fig 5.3 Confusion Matrix (2 x 2)	30
	Fig 5.4 Confusion Matrix for Instagram Account	30
7	Fig 7.1 Final evaluation Code	37
	Fig 7.2 Final Output	37

LIST OF TABLES

CHAPTER NO	TABLE NAMES	PAGE NO
1	3.1 SOFTWARE REQUIREMENTS	11
2	3.2 HARDWARE REQUIREMENTS	11

LIST OF ABBREVIATIONS

1. NLP - Natural Language Processing
2. SVM - Support Vector Machine
3. OSN - Online Social Network
4. SN - Social Network
5. FIS - Facebook Immune System
6. AI - Artificial Intelligence
7. RF - Random Forest
8. DT - Decision Tree
9. NB - Naïve Bayes
10. GUI - Graphical User Interface
11. CLI - Command Line Interface
12. PCA - Principle Component Analysis
13. DF - Data Flow
14. ER - Entity Relationship
15. TPR - True Positive Rate
16. FPR - False Positive Rate
17. AUC - Area Under ROC Curve
18. ROC - Receiver Operating Characteristic

ABSTRACT

At present social network sites are part of the life for most of the people. Every day several people are creating their profiles on the social network platforms and they are interacting with others independent of the user's location and time. The social network sites not only providing advantages to the users and also provide security issues to the users as well their information. To analyze, who are encouraging threats in social network we need to classify the social networks profiles of the users. From the classification, we can get the genuine profiles and fake profiles on the social networks. Traditionally, we have different classification methods for detecting the fake profiles on the social networks. But we need to improve the accuracy rate of the fake profile detection in the social networks. In this paper we are proposing Machine learning and Natural language Processing (NLP) techniques to improve the accuracy rate of the fake profiles detection. We can use the Support Vector Machine (SVM) and Naïve Bayes algorithm.

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW OF THE PROJECT

Social networking has end up a well-known recreation within the web at present, attracting hundreds of thousands of users, spending billions of minutes on such services. Online Social network (OSN) services variety from social interactions-based platforms similar to Instagram or Facebook or MySpace, to understanding dissemination-centric platforms reminiscent of twitter or Google Buzz, to social interaction characteristic brought to present systems such as Flickr. The opposite hand, enhancing security concerns and protecting the OSN privateness still signify a most important bottleneck and viewed mission.

When making use of Social Network's (SN's), one of a kind men and women share one-of-a-kind quantities of their private understanding. Having our individual know-how entirely or in part uncovered to the general public, makes us excellent targets for unique types of assaults, the worst of which could be identification theft. Identity theft happens when any individual uses character's expertise for a private attain or purpose. During the earlier years, online identification theft has been a primary problem considering it affected millions of people's worldwide. Victims of identification theft may suffer unique types of penalties; for illustration, they would lose time/cash, get dispatched to reformatory, get their public image ruined, or have their relationships with associates and loved ones damaged. At present, the vast majority of SN's does no longer verifies ordinary users" debts and has very susceptible privateness and safety policies. In fact, most SN's applications default their settings to minimal privateness; and consequently, SN's became a best platform for fraud and abuse. Social Networking offerings have facilitated identity theft and Impersonation attacks for serious as good as naive attackers. To make things worse, users are required to furnish correct understanding to set up an account in Social Networking web sites.

The details which can be supplied with the aid of the person on the time of profile creation is known as static knowledge, the place as the small print that are recounted with the aid of the system within the network is called dynamic knowledge. Static knowledge includes demographic elements of a person and his/her interests and dynamic knowledge includes person runtime habits and locality in the network. The vast majority of current research depends on static and dynamic data. However, this isn't relevant to lots of the social networks, where handiest some of static profiles are seen and dynamic profiles usually are not obvious to the person network. More than a few procedures have been proposed by one of a kind researcher to realize the fake identities and malicious content material in online social networks. Each process had its own deserves and demerits.

The problems involving social networking like privacy, on-line bullying, misuse, and trolling and many others. Are many of the instances utilized by false profiles on social networking sites. False profiles are the profiles which are not specific i.e. They're the profiles of men and women with false credentials. The false Facebook profiles more commonly are indulged in malicious and undesirable activities, causing problems to the social community customers. Individuals create fake profiles for social engineering, online impersonation to defame a man or woman, promoting and campaigning for a character or a crowd of individuals. Facebook has its own security system to guard person credentials from spamming, phishing, and so on. And the equal is often called Facebook Immune system (FIS). The FIS has now not been ready to observe fake profiles created on Facebook via customers to a bigger extent.

1.2 LITERATURE SURVEY

[1] **Title:** Understanding User Profiles on Social Media for Fake News Detection

Authors: Kai Shu, Suhang Wang, Huan Liu – 2018

Description:

Consuming news from social media is becoming increasingly popular nowadays. Social media brings benefits to users due to the inherent nature of fast dissemination, cheap cost, and easy access. However, the quality of news is considered lower than traditional news outlets, resulting in large amounts of fake news. Detecting fake news becomes very important and is attracting increasing attention due to the detrimental effects on individuals and the society. The performance of detecting fake news only from content is generally not satisfactory, and it is suggested to incorporate user social engagements as auxiliary information to improve fake news detection. Thus, it necessitates an in-depth understanding of the correlation between user profiles on social media and fake news. In this paper, we construct real-world datasets measuring users trust level on fake news and select representative groups of both “experienced” users who are able to recognize fake news items as false and “naïve” users who are more likely to believe fake news. We perform a comparative analysis over explicit and implicit profile features between these user groups, which reveals their potential to differentiate fake news. The findings of this paper lay the foundation for future automatic fake news detection research.

[2] **Title:** Identifying Fake Profiles in LinkedIn

Authors: Shalinda Adikari, Kaushik Dutta – 2019

Description:

As organizations increasingly rely on professionally oriented networks such as LinkedIn (the largest such social network) for building business connections, there is increasing value in having one's profile noticed within the network. As this value increases, so does the temptation to misuse the network for unethical purposes.

Fake profiles have an adverse effect on the trustworthiness of the network as a whole, and can represent significant costs in time and effort in building a connection based on fake information. Unfortunately, fake profiles are difficult to identify. Approaches have been proposed for some social networks; however, these generally rely on data that are not publicly available for LinkedIn profiles. In this research, we identify the minimal set of profile data necessary for identifying fake profiles in LinkedIn, and propose an appropriate data mining approach for fake profile identification. We demonstrate that, even with limited profile data, our approach can identify fake profiles with 87% accuracy and 89% True Negative Rate, which is comparable to the results obtained based on larger data sets and more expansive profile information. Further, when compared to approaches using similar amounts and types of data, our method provides an improvement of approximately 14% accuracy.

[3] **Title:** A Feature Based Approach to Detect Fake Profiles in Twitter

Authors: Jyoti Kaubiyal, Ankit Kumar Jain - 2019

Description:

Social networking platforms, particularly sites like Twitter and Facebook have grown tremendously in the past decade and has solicited the interest of millions of users. They have become a preferred means of communication, due to which it has also attracted the interest of various malicious entities such as spammers. The growing number of users on social media has also created the problem of fake accounts. These false and fake identities are intensively involved in malicious activities such as spreading abuse, misinformation, spamming and artificially inflating the number of users in an application to promote and sway public opinion. Detecting these fake identities, thus becomes important to protect genuine users from malicious intents. To address this issue, we aim to use a feature-based approach to identify these fake profiles on social media platforms. We have used twenty-four features to identify fake accounts efficiently. To verify the classification results three classification

algorithms are used. Experimental results show that our model was able to reach 87.9% accuracy using the Random Forest algorithm. Hence, the proposed approach is efficient in detecting fake profiles.

[4] **Title:** Method for detecting spammers and fake profiles in social networks

Authors: Yuval Elovici, Michael FIRE, Gilad Katz - 2019

Description:

A method for protecting user privacy in an online social network, according to which negative examples of fake profiles and positive examples of legitimate profiles are chosen from the database of existing users of the social network. Then, a predetermined set of features is extracted for each chosen fake and legitimate profile, by dividing the friends or followers of the chosen examples to communities and analyzing the relationships of each node inside and between the communities. Classifiers that can detect other existing fake profiles according to their features are constructed and trained by using supervised learning.

[5] **Title:** Social Networks Fake Profiles Detection Using Machine Learning Algorithms

Authors: Yasyn Elyusufi, Zakaria Elyusufi – 2020

Description:

Fake profiles play an important role in advanced persisted threats and are also involved in other malicious activities. The present paper focuses on identifying fake profiles in social media. The approaches to identifying fake profiles in social media can be classified into the approaches aimed on analysing profiles data and individual accounts. Social networks fake profile creation is considered to cause more harm than any other form of cybercrime. This crime has to be detected even before the user is notified about the fake profile creation. Many algorithms and methods have been proposed for the detection of fake profiles in the literature.

This paper sheds light on the role of fake identities in advanced persistent threats and covers the mentioned approaches of detecting fake social media profiles. In order to make a relevant prediction of fake or genuine profiles, we will assess the impact of three supervised machine learning algorithms: Random Forest (RF), Decision Tree (DT-J48), and Naïve Bayes (NB).

CHAPTER 2

SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

There are lots of issues that make this procedure tough to implement and one of the biggest problems associated with fraud detection is the lack of both the literature providing experimental results and of real-world data for academic researchers to perform experiments on. The reason behind this is the sensitive financial data associated with the fraud that has to be kept confidential for the purpose of customer's privacy. Now, here we enumerate different properties a fraud detection system should have in order to generate proper results:

The system should be able to handle skewed distributions, since only a very small percentage of all credit card transactions is fraudulent.

There should be a proper means to handle the noise. Noise is the errors that is present in the data, for example, incorrect dates. Another problem related to this field is overlapping data. Many transactions may resemble fraudulent transactions when actually they are genuine transactions. The opposite also happens, when a fraudulent transaction appears to be genuine.

The systems should be able to adapt themselves to new kinds of fraud. Since after a while, successful fraud techniques decrease in efficiency due to the fact that they become well known because an efficient fraudster always find a new and inventive ways of performing his job. There is a need for good metrics to evaluate the classifier system. For example, the overall accuracy is not suited for evaluation on a skewed distribution, since even with a very high accuracy; almost all fraudulent transactions can be misclassified.

2.2 DISADVANTAGES OF EXISTING SYSTEM

- The most of existing methods has ignored the poor-quality data like noise or Feature handled complex.
- The problems involving social networking like privacy, on-line bullying, misuse, not accurate analysis and trolling and many others. There are many of the instances utilized by false profiles on social networking sites.

- False profiles are the profiles which are not specific i.e, They're the profiles of men and women with false credentials.

2.3 PROPOSED SYSTEM

A proper and thorough literature survey concludes that there are various methods that can be used to detect Fake profile detection. Some of these approaches are Machine Learning and NLP.

To analyze, who are encouraging threats in social network we need to classify the social networks profiles of the users. From the classification, we can get the genuine profiles and fake profiles on the social networks.

Traditionally, we have different classification methods for detecting the fake profiles on the social networks. But we need to improve the accuracy rate of the fake profile detection in the social networks.

On this paper we presented a machine learning and natural language processing system to observe the false profiles in online social networks. Moreover, we are adding the five algorithms such that model Support Vector Machine (SVM), Random Forest classifier, Gradient Boost classifier, Naïve Bayes, and Logistic Regression algorithm to increase the detection accuracy rate of the fake profiles. In final prediction we gain the values of accuracy, classification report and confusion matrix. This proposed system is used to evaluate the best model to increase the detection accuracy rate of the fake profiles.

2.4 ADVANTAGES OF PROPOSED SYSTEM

- Accuracy is high when compared to the existing system.
- Fully secure and easily detect the fake profile in social networks.
- Time consumption for detecting the fake profiles.

- More datasets are included.
- We can find the all types of profiles on different social media application also.

2.5 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis

- i. Economical Feasibility
- ii. Technical Feasibility
- iii. Social Feasibility

2.6 ECONOMIC FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

2.7 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources.

This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

2.8 SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

CHAPTER 3

SYSTEM REQUIREMENTS

3.1 SOFTWARE REQUIREMENTS

Operating System	Windows 7 or later
Tool	Anaconda(Jupyter notebook)
Documentation	Ms – Office
Software	Python

3.2 HARDWARE REQUIREMENTS

CPU type	I5
Ram size	4GB
Hard disk capacity	80 GB
Keyboard type	Internet keyboard
Monitor type	15 Inch colour monitor
CD -drive type	52xmax

3.3 REQUIREMENT ANALYSIS

Requirements are a feature of a system or description of something that the system is capable of doing in order to fulfil the system's purpose. It provides the appropriate mechanism for understanding what the customer wants, analyzing the needs assessing feasibility, negotiating a reasonable solution, specifying the solution unambiguously, validating the specification and managing the requirements as they are translated into an operational system.

3.4 PYTHON:

Python is a dynamic, high level, free open source and interpreted programming language. It supports object-oriented programming as well as procedural oriented

programming. In Python, we don't need to declare the type of variable because it is a dynamically typed language.

For example, `x=10`. Here, `x` can be anything such as String, int, etc.

Python is an interpreted, object-oriented programming language similar to PERL, that has gained popularity because of its clear syntax and readability. Python is said to be relatively easy to learn and portable, meaning its statements can be interpreted in a number of operating systems, including UNIX-based systems, Mac OS, MS-DOS, OS/2, and various versions of Microsoft Windows 98. Python was created by Guido van Rossum, a former resident of the Netherlands, whose favourite comedy group at the time was Monty Python's Flying Circus.

Features in Python

There are many features in Python, some of which are discussed below

- Easy to code
- Free and Open Source
- Object-Oriented Language
- GUI Programming Support
- High-Level Language
- Extensible feature
- Python is Portable language
- Python is Integrated language

3.5 ANACONDA

Anaconda distribution comes with over 250 packages automatically installed, and over 7,500 additional open-source packages can be installed from PyPI as well as the conda package and virtual environment manager. It also includes a GUI, Anaconda Navigator,^[12] as a graphical alternative to the Command Line Interface (CLI).

The big difference between conda and the pip package manager is in how package

dependencies are managed, which is a significant challenge for Python data science and the reason conda exists.

When pip installs a package, it automatically installs any dependent Python packages without checking if these conflict with previously installed packages. It will install a package and any of its dependencies regardless of the state of the existing installation. Because of this, a user with a working installation of, for example, Google Tensorflow, can find that it stops working having used pip to install a different package that requires a different version of the dependent numpy library than the one used by Tensorflow. In some cases, the package may appear to work but produce different results in detail.

Open source packages can be individually installed from the Anaconda repository, Anaconda Cloud (anaconda.org), or the user's own private repository or mirror, using the conda install command. Anaconda, Inc. compiles and builds the packages available in the Anaconda repository itself, and provides binaries for Windows 32/64 bit, Linux 64 bit and MacOS 64-bit. Anything available on PyPI may be installed into a conda environment using pip, and conda will keep track of what it has installed itself and what pip has installed.

Custom packages can be made using the conda build command, and can be shared with others by uploading them to Anaconda Cloud, PyPI or other repositories.

The default installation of Anaconda2 includes Python 2.7 and Anaconda3 includes Python 3.7. However, it is possible to create new environments that include any version of Python packaged with conda.

3.6 ANACONDA NAVIGATOR

Anaconda Navigator is a desktop Graphical User Interface (GUI) included in Anaconda distribution that allows users to launch applications and manage conda packages, environments and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository, install them in an environment, run the packages and update them. It is available for Windows, macOS and Linux.

The following applications are available by default in Navigator:

- JupyterLab
- Jupyter Notebook
- QtConsole
- Spyder
- Glue
- Orange
- RStudio
- Visual Studio Code

3.7 JUPYTER NOTEBOOK

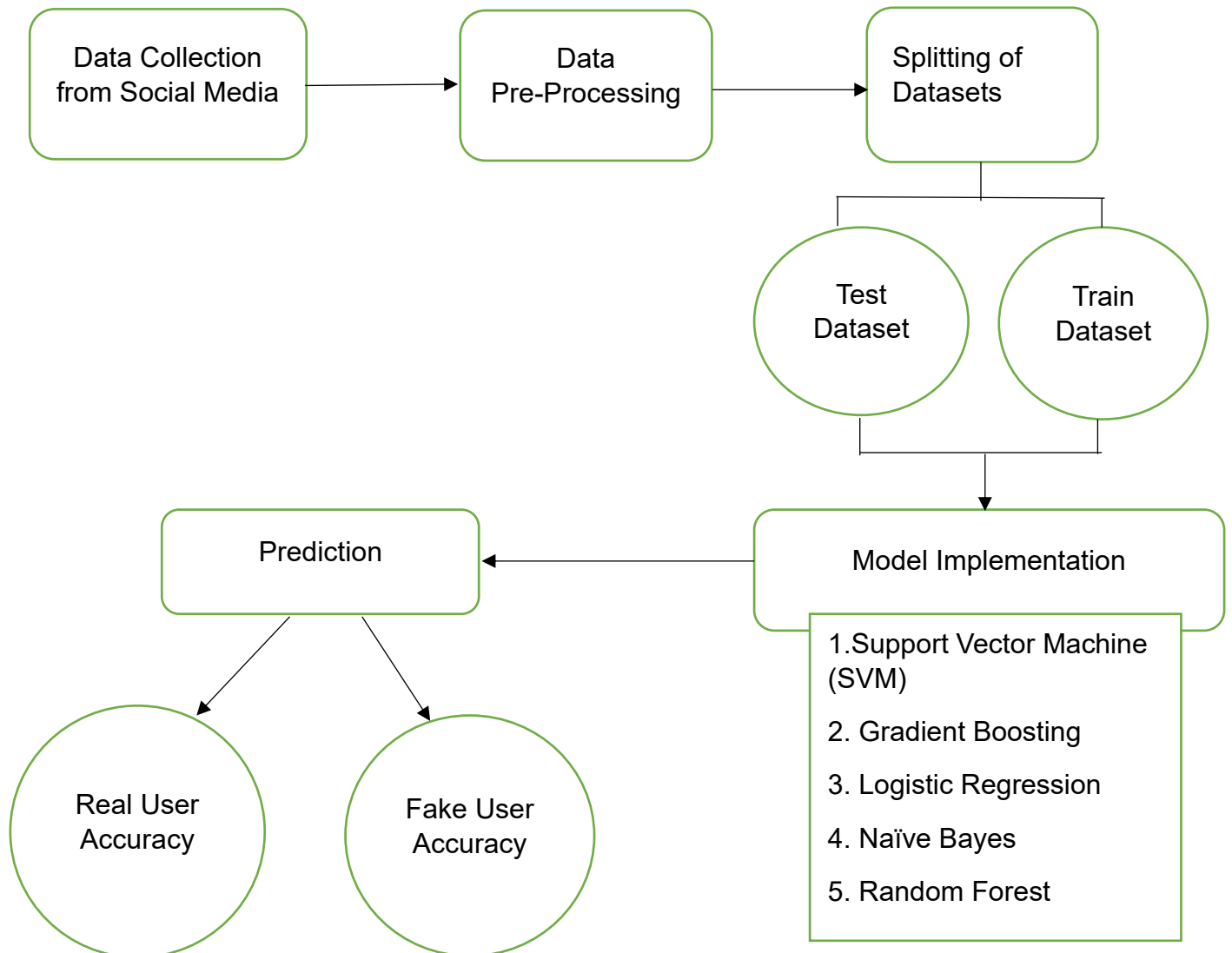
Jupyter Notebook (formerly IPython Notebooks) is a web-based interactive computational environment for creating Jupyter notebook documents. The "notebook" term can colloquially make reference to many different entities, mainly the Jupyter web application, Jupyter Python web server, or Jupyter document format depending on context. Usually ending with the ".ipynb" extension. Jupyter Notebook can connect to many kernels to allow programming in different languages. By default, Jupyter Notebook ships with the IPython kernel. As of the 2.3 release^{[11][12]} (October 2014), there are currently 49 Jupyter-compatible kernels for many programming languages, including Python, R, Julia and Haskell.

The Notebook interface was added to IPython in the 0.12 release^[14] (December 2011), renamed to Jupyter notebook in 2015 (IPython 4.0 – Jupyter 1.0). Jupyter Notebook is similar to the notebook interface of other programs such as Maple, Mathematica, and SageMath, a computational interface style that originated with Mathematica in the 1980s. According to *The Atlantic*, Jupyter interest overtook the popularity of the Mathematica notebook interface in early 2018.

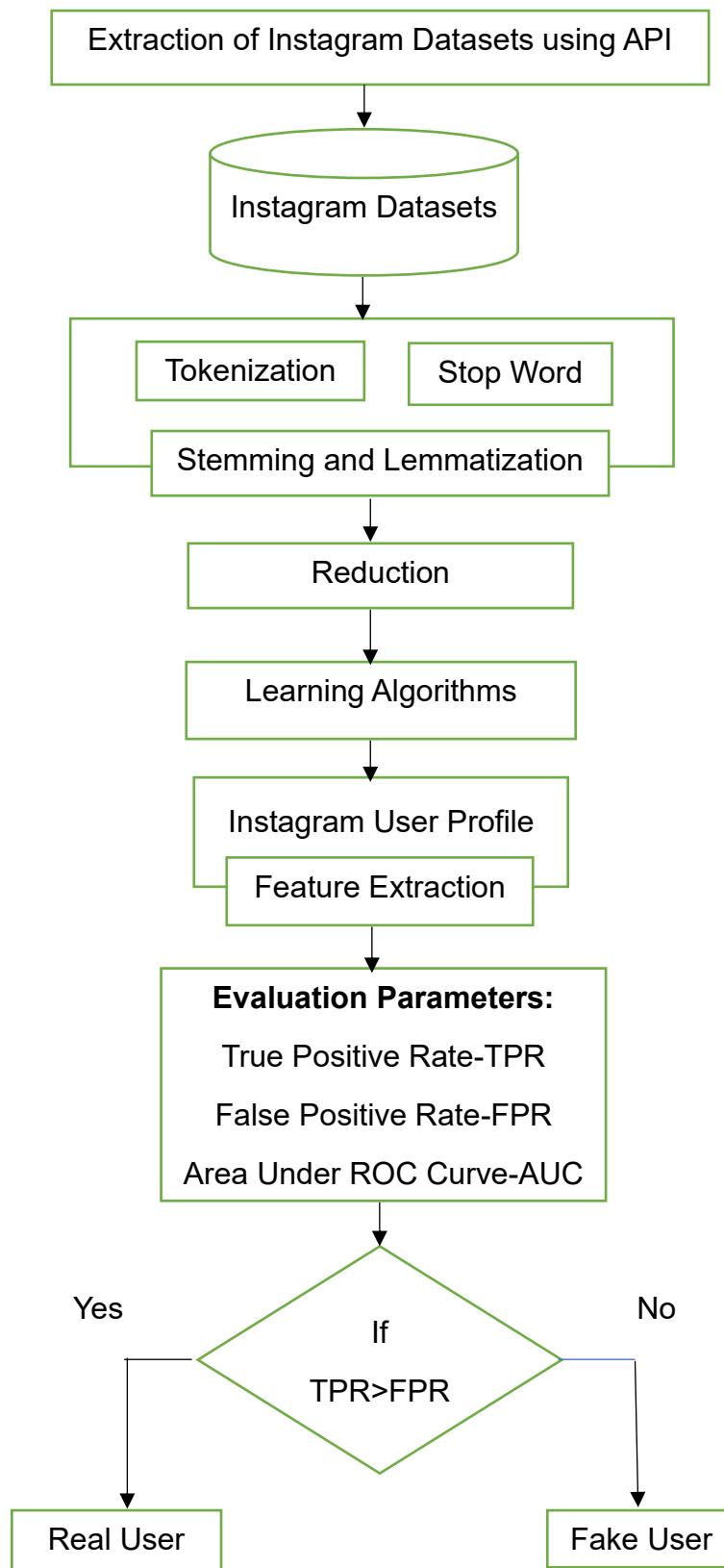
CHAPTER 4

SYSTEM ARCHITECTURE

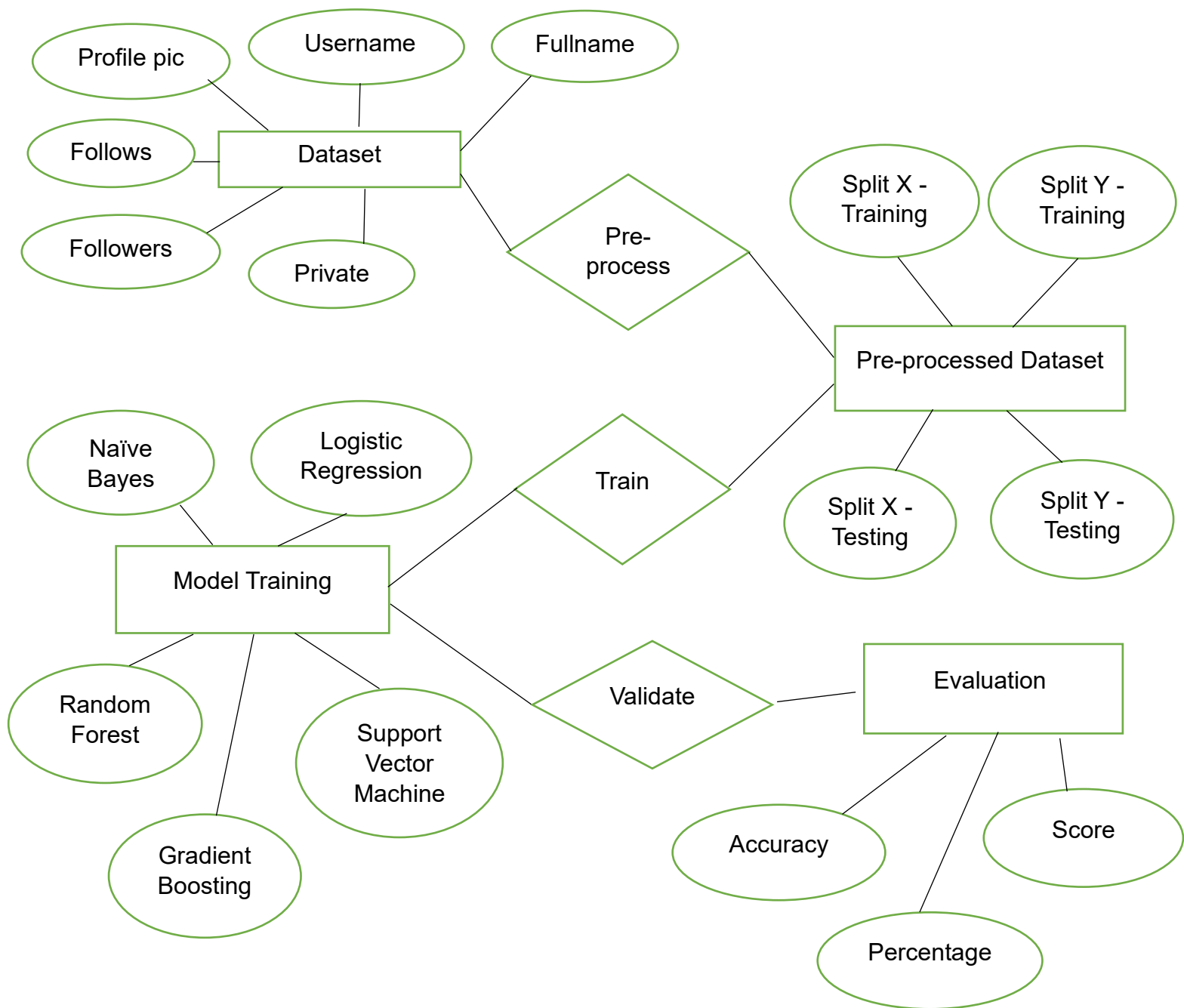
4.1 ARCHITECTURE DIAGRAM



4.2 DATA FLOW (DF) DIAGRAM



4.3 ENTITY RELATIONSHIP (ER) DIAGRAM



CHAPTER 5

SYSTEM IMPLEMENTATION

5.1 ARTIFICIAL INTELLIGENCE:

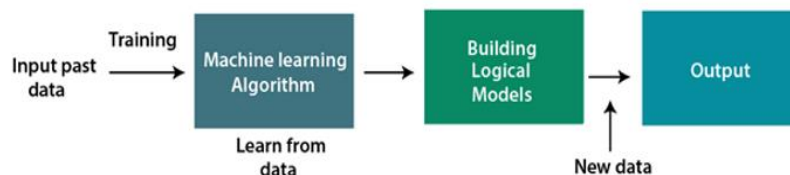
Artificial intelligence (AI) is the ability of a computer program or a machine to think and learn. It is also a field of study which tries to make computers "smart". As machines become increasingly capable, mental facilities once thought to require intelligence are removed from the definition. AI is an area of computer sciences that emphasizes the creation of intelligent machines that work and reacts like humans. Some of the activities computers with artificial intelligence are designed for include: Face recognition, Learning, Planning, Decision making etc.,

Artificial intelligence is the use of computer science programming to imitate human thought and action by analysing data and surroundings, solving or anticipating problems and learning or self-teaching to adapt to a variety of tasks.

5.2 MACHINE LEARNING

Machine learning is a growing technology which enables computers to learn automatically from past data. Machine learning uses various algorithms for **building** mathematical models and making predictions using historical data or information. Currently, it is being used for various tasks such as image recognition, speech recognition, email filtering, Facebook auto-tagging, recommender system, and many more.

Machine Learning is said as a subset of artificial intelligence that is mainly concerned with the development of algorithms which allow a computer to learn from the data and past experiences on their own. The term machine learning was first introduced by Arthur Samuel in 1959.



We can define it in a summarized way as: “Machine learning enables a machine to automatically learn from data, improve performance from experiences, and predict things without being explicitly programmed”. Suppose we have a complex problem, where we need to perform some predictions, so instead of writing a code for it, we just need to feed the data to generic algorithms, and with the help of these algorithms, machine builds the logic as per the data and predict the output. Machine learning has changed our way of thinking about the problem. The below block diagram explains the working of Machine Learning algorithm:

5.2.1 FEATURES OF MACHINE LEARNING

- Machine learning uses data to detect various patterns in a given dataset.
- It can learn from past data and improve automatically.
- It is a data-driven technology.
- Machine learning is much similar to data mining as it also deals with the huge amount of the data.

5.2.2 CLASSIFICATION OF MACHINE LEARNING

At a broad level, machine learning can be classified into three types:

1. Supervised learning
2. Unsupervised learning
3. Reinforcement learning

1) SUPERVISED LEARNING

Supervised learning is a type of machine learning method in which we provide sample labelled data to the machine learning system in order to train it, and on that basis, it predicts the output. The system creates a model using labelled data to understand the datasets and learn about each data, once the training and processing are done then we test the model by providing a sample data to check whether it is predicting the exact output or not. The goal of supervised learning is to map input data with the output data.

Supervised learning can be grouped further in two categories of algorithms:

- Classification
- Regression

2) UNSUPERVISED LEARNING

Unsupervised learning is a learning method in which a machine learns without any supervision. The training is provided to the machine with the set of data that has not been labelled, classified, or categorized, and the algorithm needs to act on that data without any supervision. The goal of unsupervised learning is to restructure the input data into new features or a group of objects with similar patterns.

It can be further classified into two categories of algorithms:

- Clustering
- Association

3) REINFORCEMENT LEARNING

Reinforcement learning is a machine learning training method based on rewarding desired behaviours and/or punishing undesired ones. In general, a reinforcement learning agent is able to perceive and interpret its environment, take actions and learn through trial and error.

5.3 NATURAL LANGUAGE PROCESSING (NLP)

Natural language Processing (NLP) is a branch of artificial intelligence that helps computers understand, interpret and manipulate human language. NLP draws from many disciplines, including computer science and computational linguistics, in its pursuit to fill the gap between human communication and computer understanding. While natural language processing isn't a new science, the technology is rapidly advancing thanks to an increased interest in human-to-machine communications, plus an availability of big data, powerful computing and enhanced algorithms. Natural language processing includes many different techniques for interpreting human language, ranging from statistical and machine learning methods to rules-based and algorithmic approaches. We need a broad array of approaches because the text- and voice-based data varies widely, as do the practical applications.

Basic NLP tasks include tokenization and parsing, lemmatization/stemming, part-of-speech tagging, language detection and identification of semantic relationships. In general terms, NLP tasks break down language into shorter, elemental pieces, try to understand relationships between the pieces and explore how the pieces work together to create meaning.

These underlying tasks are often used in higher-level NLP capabilities, such as:

- **Content categorization.** A linguistic-based document summary, including search and indexing, content alerts and duplication detection.
- **Topic discovery and modelling.** Accurately capture the meaning and themes in text collections, and apply advanced analytics to text, like optimization and forecasting.
- **Contextual extraction.** Automatically pull structured information from text-based sources.
- **Sentiment analysis.** Identifying the mood or subjective opinions within large amounts of text, including average sentiment and opinion mining.
- **Speech-to-text and text-to-speech conversion.** Transforming voice commands into written text, and vice versa.
- **Document summarization.** Automatically generating synopses of large bodies of text.
- **Machine translation.** Automatic translation of text or speech from one language to another.

In all these cases, the overarching goal is to take raw language input and use linguistics and algorithms to transform or enrich the text in such a way that it delivers greater value.

5.3.1 NLP PRE-PROCESSING

Text preprocessing involves transforming text into a clean and consistent format that can then be fed into a model for further analysis and learning.

Text preprocessing techniques may be general so that they are applicable to many types of applications, or they can be specialized for a specific task.

TOKENIZATION

Tokenization is used in natural language processing to split paragraphs and sentences into smaller units that can be more easily assigned meaning. The first step of the NLP process is gathering the data (a sentence) and breaking it into understandable parts (words).

STOP WORD

Stop word removal is one of the most commonly used preprocessing steps across different NLP applications. The idea is simply removing the words that occur commonly across all the documents in the corpus. Typically, articles and pronouns are generally classified as stop words.

STEMMING AND LEMMATIZATION

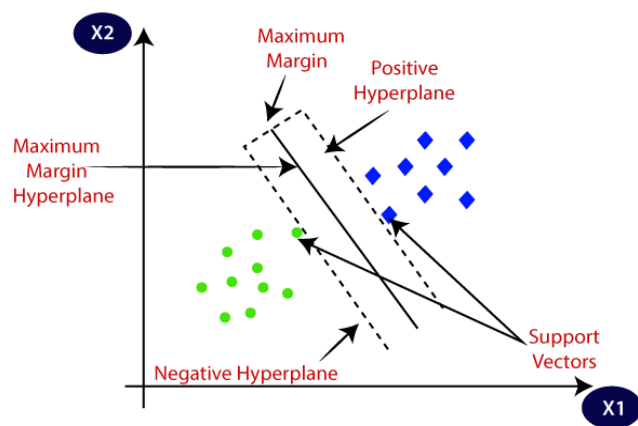
Stemming is basically removing the suffix from a word and reduce it to its root word. For example: “Flying” is a word and its suffix is “ing”, if we remove “ing” from “Flying” then we will get base word or root word which is “Fly”. We uses these suffix to create a new word from original stem word.

Lemmatization considers the context and converts the word to its meaningful base form, which is called Lemma. For instance, stemming the word 'Caring' would return 'Car'. For instance, lemmatizing the word 'Caring' would return 'Care'.

5.4 SUPPORT VECTOR MACHINE(SVM)

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data

point in the correct category in the future. This best decision boundary is called hyper plane.



SVM chooses the extreme points/vectors that help in creating the hyper plane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyper plane:

HYPER PLANE

There can be multiple lines/decision boundaries to segregate the classes in n -dimensional space. This best boundary is known as the hyper plane of SVM. The dimensions of the hyper plane depend on the features present in the dataset, which means if there are 2 features (as shown in image), then hyper plane will be a straight line. And if there are 3 features, then hyper plane will be a 2-dimension plane. We always create a hyper plane that has a maximum margin.

SUPPORT VECTORS

The data points or vectors that are the closest to the hyper plane and which affect the position of the hyper plane are termed as Support Vector. Since these vectors support the hyper plane, hence called a Support vector.

5.5 NAIVE BAYES

1. Naïve Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems.
2. It is mainly used in *text classification* that includes a high-dimensional training dataset.
3. Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.
4. It is a probabilistic classifier, which means it predicts on the basis of the probability of an object.
5. Some popular examples of Naïve Bayes Algorithm are spam filtration, Sentimental analysis, and classifying articles.

The Naïve Bayes algorithm is comprised of two words Naïve and Bayes, which can be described as:

- **NAÏVE:** It is called Naïve because it assumes that the occurrence of a certain feature is independent of the occurrence of other features. Such as if the fruit is identified on the bases of colour, shape, and taste, then red, spherical, and sweet fruit is recognized as an apple. Hence each feature individually contributes to identify that it is an apple without depending on each other.
- **BAYES:** It is called Bayes because it depends on the principle of Bayes' Theorem.

BAYES' THEOREM

- Bayes' theorem is also known as **Bayes' Rule** or **Bayes' law**, which is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability.

- The formula for Bayes' theorem is given as:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Where, $P(A|B)$ is Posterior probability: Probability of hypothesis A on the observed event B.

5.6 LOGISTIC REGRESSION

Logistic regression is a supervised learning classification algorithm used to predict the probability of a target variable. The nature of target or dependent variable is dichotomous, which means there would be only two possible classes.

In simple words, the dependent variable is binary in nature having data coded as either 1 (stands for success/yes) or 0 (stands for failure/no).

Mathematically, a logistic regression model predicts $P(Y=1)$ as a function of X . It is one of the simplest ML algorithms that can be used for various classification problems such as spam detection, Diabetes prediction, cancer detection etc.

TYPES OF LOGISTIC REGRESSION

Generally, logistic regression means binary logistic regression having binary target variables, but there can be two more categories of target variables that can be predicted by it. Based on those number of categories, Logistic regression can be divided into following types –

BINARY OR BINOMIAL

In such a kind of classification, a dependent variable will have only two possible types either 1 and 0. For example, these variables may represent success or failure, yes or no, win or loss etc.

MULTINOMIAL

In such a kind of classification, dependent variable can have 3 or more possible unordered types or the types having no quantitative significance.

ORDINAL

In such a kind of classification, dependent variable can have 3 or more possible ordered types or the types having a quantitative significance. For example, these variables may represent “poor” or “good”, “very good”,

“Excellent” and each category can have the scores like 0,1,2,3.

LOGISTIC REGRESSION ASSUMPTIONS

Before diving into the implementation of logistic regression, we must be aware of the following assumptions about the same –

In case of binary logistic regression, the target variables must be binary always and the desired outcome is represented by the factor level 1. There should not be any multi-collinearity in the model, which means the independent variables must be independent of each other. We must include meaningful variables in our model. We should choose a large sample size for logistic regression.

5.7 RANDOM FOREST

Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.

WORKING OF RANDOM FOREST ALGORITHM

Before understanding the working of the random forest, we must look into the ensemble technique. *Ensemble* simply means combining multiple models. Thus, a collection of models is used to make predictions rather than an individual model.

Ensemble uses two types of methods:

1. **BAGGING**– It creates a different training subset from sample training data with replacement & the final output is based on majority voting. For example, Random Forest.
2. **BOOSTING**– It combines weak learners into strong learners by creating sequential models such that the final model has the highest accuracy.

Steps involved in random forest algorithm:

Step 1: In Random Forest n number of random records is taken from the data set having k number of records.

Step 2: Individual decision trees are constructed for each sample.

Step 3: Each decision tree will generate an output.

Step 4: Final output is considered based on Majority Voting or Averaging for Classification and regression respectively.

IMPORTANT FEATURES OF RANDOM FOREST

1. **Diversity**- Not all attributes/variables/features are considered while making an individual tree, each tree is different.
2. **Immune to the curse of dimensionality**- Since each tree does not consider all the features, the feature space is reduced.
3. **Parallelization**-Each tree is created independently out of different data and attributes. This means that we can make full use of the CPU to build random forests.
4. **Train-Test split**- In a random forest we don't have to segregate the data for train and test as there will always be 30% of the data which is not seen by the decision tree.
5. **Stability**- Stability arises because the result is based on majority voting/averaging.

Models Comparison Using Cross-Validation

```
In [17]: from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from sklearn.svm import SVC
from sklearn.naive_bayes import GaussianNB

model_list = [LogisticRegression(max_iter=600),
               SVC(),
               GaussianNB(),
               RandomForestClassifier(random_state=55),
               GradientBoostingClassifier(random_state=56)]

train_scores = []
val_scores = []
```

Fig 5.1 Comparing the models using Cross-Validation

5.8 CORRELATION HEATMAP

Correlation heatmaps are a type of plot that visualize the strength of relationships between numerical variables. Correlation plots are used to understand which variables are related to each other and the strength of this relationship. A correlation plot typically contains a number of numerical variables, with each variable represented by a column. The rows represent the relationship between each pair of variables. The values in the cells indicate the strength of the relationship, with positive values indicating a positive relationship and negative values indicating a negative relationship.

Correlation heatmaps can be used to find potential relationships between variables and to understand the strength of these relationships. In addition, correlation plots can be used to identify outliers and to detect linear and nonlinear relationships. The color-coding of the cells makes it easy to identify relationships between variables at a glance. Correlation heatmaps can be used to find both linear and nonlinear relationships between variables.

```
Out[14]: Text(0.5, 1.0, 'Correlation Heatmap Between Features')
```

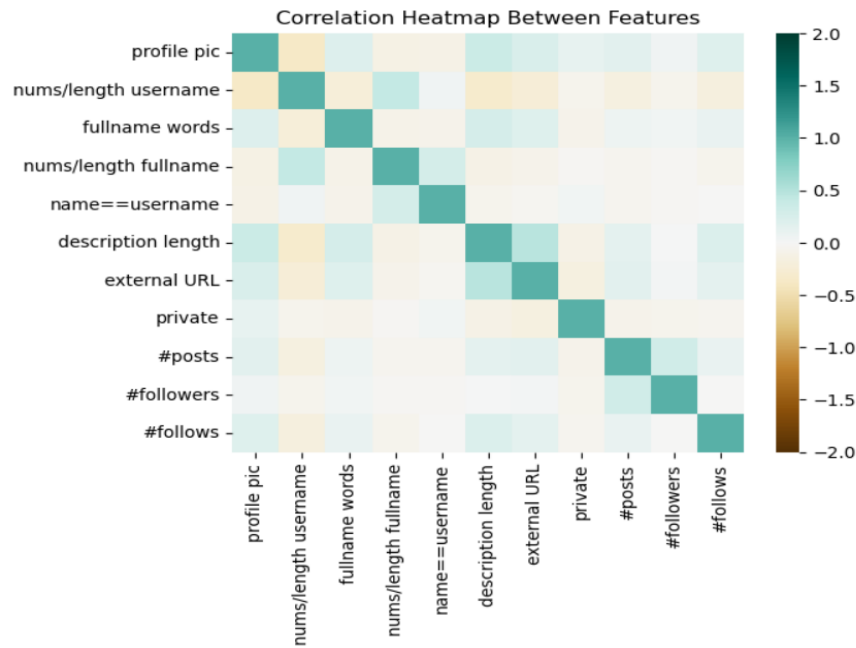


Fig 5.2 Correlation Heatmap Between Features

5.9 CONFUSION MATRIX

A confusion matrix is used to measure the performance of a classifier in depth. In this simple guide to Confusion Matrix, we will get to understand and learn confusion matrices better.

A confusion matrix presents a table layout of the different outcomes of the prediction and results of a classification problem and helps visualize its outcomes.

FEATURES OF CONFUSION MATRIX

- For the 2 prediction classes of classifiers, the matrix is of 2*2 table, for 3 classes, it is 3*3 table, and so on.
- The matrix is divided into two dimensions, that are **predicted values** and **actual values** along with the total number of predictions.
- Predicted values are those values, which are predicted by the model, and actual values are the true values for the given observations.

- It looks like the below table:

n = total predictions	Actual: No	Actual: Yes
Predicted: No	True Negative	False Positive
Predicted: Yes	False Negative	True Positive

Fig 5.3 Confusion Matrix (2 X 2)

The above table has the following cases:

- **True Negative:** Model has given prediction No, and the real or actual value was also No.
- **True Positive:** The model has predicted yes, and the actual value was also true.
- **False Negative:** The model has predicted no, but the actual value was Yes, it is also called as **Type-II error**.
- **False Positive:** The model has predicted Yes, but the actual value was No. It is also called a **Type-I error**.

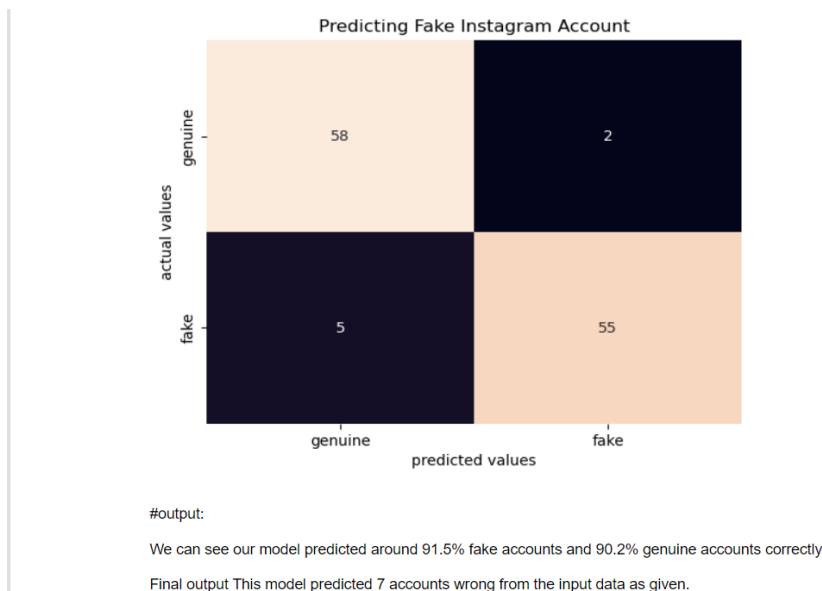


Fig 5.4 Confusion Matrix for Instagram Account

CHAPTER 6

CONCLUSION

In this project, we proposed machine learning algorithms along with natural language processing techniques. By using these techniques, we can easily detect the fake profiles from the social network sites. In this project we took the Instagram dataset to identify the fake profiles. The NLP pre-processing techniques are used to analyze the dataset and machine learning algorithm such as SVM and Naïve Bayes are used to classify the profiles. These learning algorithms are improved the detection accuracy rate in this project

CHAPTER 7

APPENDIX

7.1 SOURCE CODE

```
#Import Packages
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

def load_train_data():
    train_data = pd.read_csv('train.csv', header = 0)
    X_train = train_data.drop(columns='fake')
    y_train = train_data['fake']
    return X_train, y_train

from sklearn.datasets import load_files

def load_test_data():
    test_data = pd.read_csv('test.csv', header = 0)
    X_test = test_data.drop(columns='fake')
    y_test = test_data['fake']
    return X_test, y_test

from sklearn.model_selection import cross_validate

def get_classifier_cv_score(model, X, y, scoring='accuracy', cv=7):
    scores = cross_validate(model, X, y, cv=cv, scoring=scoring,
return_train_score=True)
    train_scores = scores['train_score']
    val_scores = scores['test_score']
    train_mean = np.mean(train_scores)
    val_mean = np.mean(val_scores)
    return train_mean, val_mean

def print_grid_search_result(grid_search):
```

```

print(grid_search.best_params_)

best_train =
grid_search.cv_results_["mean_train_score"][grid_search.best_index_]

print("best mean_train_score: {:.3f}".format(best_train))

best_test =
grid_search.cv_results_["mean_test_score"][grid_search.best_index_]

print("best mean_test_score: {:.3f}".format(best_test))

from sklearn.metrics import confusion_matrix

def plot_confusion_matrix(y_actual, y_pred, labels, title=""):

    data = confusion_matrix(y_actual, y_pred)

    ax = sns.heatmap(data,

                      annot=True,

                      cbar=False,

                      fmt='d',

                      xticklabels = labels,

                      yticklabels = labels)

    ax.set_title(title)

    ax.set_xlabel("predicted values")

    ax.set_ylabel("actual values")

#data loading
X_data, y_data = load_train_data()
X_data.info()
X_data.head()
X_data.tail()
X_data.shape
y_data.shape

# Finding Missing Values
X_data.isnull().sum()

# Check if Imbalance in Labels

```

#labels is about 1:1 which means there is no imbalance in the labels.

#but here the ratio would be more 2:1.

```
unique, freq = np.unique(y_data, return_counts = True)
```

```
for i, j in zip(unique, freq):
```

```
    print("Label: ", i, ", Frequency: ", j)
```

```
data_corr = X_data.corr(method='pearson')
```

```
ax = sns.heatmap(data_corr, vmin=-2, vmax=2, cmap='BrBG')
```

```
ax.set_title("Correlation Heatmap Between Features")
```

Create Training And Test Sets

```
from sklearn.model_selection import train_test_split
```

```
X_train, X_test, y_train, y_test = train_test_split(X_data, y_data, test_size=0.2,  
random_state=50)
```

```
print(X_train.shape)
```

```
print(y_train.shape)
```

Models Comparision Using Cross-Validation

```
from sklearn.linear_model import LogisticRegression
```

```
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
```

```
from sklearn.svm import SVC
```

```
from sklearn.naive_bayes import GaussianNB
```

```
model_list = [LogisticRegression(max_iter=600),
```

```
              SVC(),
```

```
              GaussianNB(),
```

```
              RandomForestClassifier(random_state=55),
```

```
              GradientBoostingClassifier(random_state=56)]
```

```
train_scores = []
```

```
val_scores = []
```

```
for model in model_list:
```

```
    train, val = get_classifier_cv_score(model, X_train, y_train, 'average_precision')
```

```

train_scores.append(train)
val_scores.append(val)

models_score = sorted(list(zip(val_scores, train_scores, model_list)),
reverse=True)

print("-----")
for val, train, model in models_score:
    print("\nModel: {}".format(model.__class__.__name__))
    print("\ntrain_score: {:.3f}".format(train))
    print("\nvalidation_score: {:.3f}".format(val))
    print("-----")

```

Hyperparameter Tuning Using Grid Search

```

#Grid Search for RandomForestClassifier
from sklearn.model_selection import GridSearchCV
import os

model = RandomForestClassifier(random_state=55)

parameters = {'n_estimators': [300, 500, 700, 1000],
              'max_depth': [7, 9, 11, 13]}

grid1 = GridSearchCV(model, parameters, cv=7, scoring='average_precision',return_train_score=True)
grid1.fit(X_train, y_train)
print_grid_search_result(grid1)

```

Grid Search for Gradient Boosting Classifier

```

model = GradientBoostingClassifier(max_depth=5, random_state=56)

parameters = {'n_estimators': [50, 100, 200],
              'learning_rate': [0.001, 0.01, 0.1, 1.0, 10.0]}
grid2 = GridSearchCV(model, parameters, cv=7, scoring='average_precision', return_train_score=True)

```

```
grid2.fit(X_train, y_train)
print_grid_search_result(grid2)
```

Pipeline

```
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler

pipeline = Pipeline([('preprocessing', StandardScaler()), ('classifier', grid1.best_estimator_)])
pipeline.fit(X_train, y_train)
print("Test score: {:.3f}".format(pipeline.score(X_test, y_test)))
```

Final Evaluation

```
X_final, y_final = load_test_data()
print("Test score: {:.3f}".format(pipeline.score(X_final, y_final)))

from sklearn.metrics import classification_report
y_pred = pipeline.predict(X_final)
print(classification_report(y_final, y_pred, target_names=["genuine", "fake"]))
labels = ["genuine", "fake"]
title = "Predicting Fake Instagram Account"
plot_confusion_matrix(y_final, y_pred, labels, title)
```

7.2 OUTPUT

Final Evaluation

```
In [27]: X_final, y_final = load_test_data()

In [28]: print("Test score: {:.3f}".format(pipeline.score(X_final, y_final)))

Test score: 0.942

In [29]: from sklearn.metrics import classification_report
y_pred = pipeline.predict(X_final)
print(classification_report(y_final, y_pred, target_names=["genuine", "fake"]))

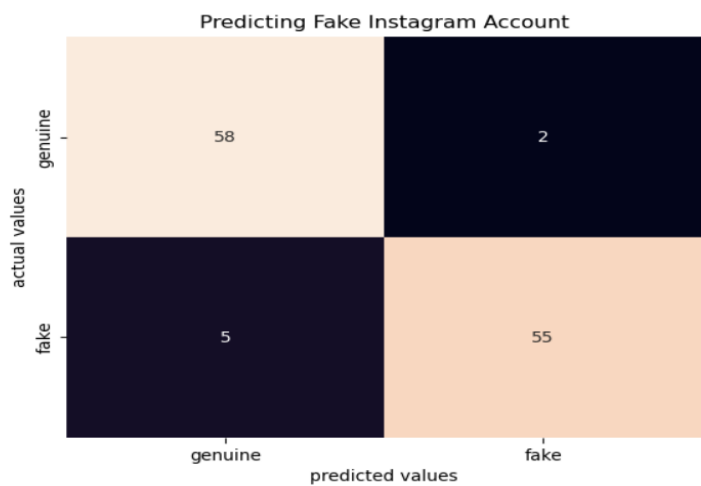
      precision    recall  f1-score   support

   genuine       0.92      0.97      0.94         60
    fake        0.96      0.92      0.94         60

 accuracy              0.94         120
  macro avg           0.94      0.94      0.94         120
 weighted avg           0.94      0.94      0.94         120

In [30]: labels = ["genuine", "fake"]
title = "Predicting Fake Instagram Account"
plot_confusion_matrix(y_final, y_pred, labels, title)
```

Fig 7.1 Final Evaluation Code



#output:

We can see our model predicted around 91.5% fake accounts and 90.2% genuine accounts correctly

Final output This model predicted 7 accounts wrong from the input data as given.

Fig 7.2 Final Output

CHAPTER 8

REFERENCES

1. Romanov, Aleksei, Alexander Semenov, Oleksiy Mazhelis, and Jari Veijalainen. "Detection of fake profiles in social media-Literature review." In *International Conference on Web Information Systems and Technologies*, vol. 2, pp. 363-369. SCITEPRESS, 2018.
2. Adikari, Shalinda, and Kaushik Dutta. "Identifying fake profiles in linkedin." *arXiv preprint arXiv:2006.01381* (2020).
3. Kaubiyal, Jyoti, and Ankit Kumar Jain. "A feature based approach to detect fake profiles in Twitter." In *Proceedings of the 3rd International Conference on Big Data and Internet of Things*, pp. 135-139. 2019.
4. Elovici, Yuval, F. I. R. E. Michael, and Gilad Katz. "Method for detecting spammers and fake profiles in social networks." U.S. Patent 9,659,185, issued May 23, 2019
5. Elyusufi, Y. and Elyusufi, Z., 2019, October. Social networks fake profiles detection using machine learning algorithms. In *The Proceedings of the Third International Conference on Smart City Applications* (pp. 30-40). Springer, Cham.
6. Ozbay, F.A. and Alatas, B., 2020. Fake news detection within online social media using supervised artificial intelligence algorithms. *Physica A: Statistical Mechanics and its Applications*, 540, p.123174.
7. Gurajala, S., White, J.S., Hudson, B. and Matthews, J.N., 2015, July. Fake Twitter accounts: profile characteristics obtained using an activity-based pattern detection approach. In *Proceedings of the 2015 International Conference on Social Media & Society* (pp. 1-7).
8. Ramalingam, D. and Chinnaiah, V., 2018. Fake profile detection techniques in large-scale online social networks: A comprehensive review. *Computers & Electrical Engineering*, 65, pp.165-177.

9. Ojo, Adebola K. "Improved model for detecting fake profiles in online social network: A case study of twitter." *Journal of Advances in Mathematics and Computer Science* (2019): 1-17.
10. Meel, Priyanka, and Dinesh Kumar Vishwakarma. "Fake news, rumor, information pollution in social media and web: A contemporary survey of state-of-the-arts, challenges and opportunities." *Expert Systems with Applications* (2019): 112986.