Prediction Of Fake Instagram Profile Using Machine Learning

Submitted in partial fulfillment of the requirements forthe award of

Bachelor of Technology degree in Information

Technologyby

Subhiksha C(37120074)

Saranya Shree S (37120063)

DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI
SALAI,CHENNAI - 600 119

MARCH – 2021

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with "A" Grade by NAAC
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai - 600 119.
Phone: 044 - 2450 3150 / 51 / 52 / 54 / 55 Fax: 044 - 2450 2344
www.sathyabama.ac.in

QUALITYSYSTEM CERTIFICATION DNV·GL ISO 9001:2015

SCHOOL OF COMPUTING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of Subhiksha C (37120074) and Saranya Shree (37120063) who carried out the project entitled "**Prediction Of Fake Instagram Profile Using Machine Learning**" under our supervision from November 2020 to March 2021.

Internal Guide

Dr. R. SUBHASHINI, M.E., Ph.D.,

Head of the Department

Dr. R. SUBHASHINI, M.E., Ph.D.,

Submitted for Viva voice Examination held on _____

Internal Examiner **External Examiner**

DECLARATION

We are Subhiksha C (37120074) and Saranya Shree (37120063)hereby declare that the project report entitled on "**Prediction Of Fake Instagram Profile Using Machine Learning**" done by me under the guidance Dr. R. SUBHASHINI, M.E., Ph.D.,,is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology degree in Information Technology.

DATE:

PLACE: Chennai **SIGNATURE OF THE CANDIDATES**

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D**, **Dean**, School of Computing , **Dr.SUBHASHINI M.E., Ph.D. ,** Head of the Department of Information and Technology for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Ms Dr. R. SUBHASHINI, M.E., Ph.D**.,for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Information Technology** who were helpful in many ways for the completion of the project.

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.

LIST OF FIGURES

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

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 Facebook or MySpace, to understanding dissemination-centric platforms reminiscent of twitter or Google Buzz, to Social interaction characteristic brought to present systems such as Flicker. 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. Easy monitoring of what customers share on-line would lead to catastrophic losses, let alone, if such bills had been hacked.

Profile information in online networks will also be static or dynamic. 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.

1.1. 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".

A Machine Learning system **learns from historical data, builds the prediction models, and whenever it receives new data, predicts the output for it**. The accuracy of predicted output depends upon the amount of data, as the huge amount of data helps to build a better model which predicts the output more accurately.

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:

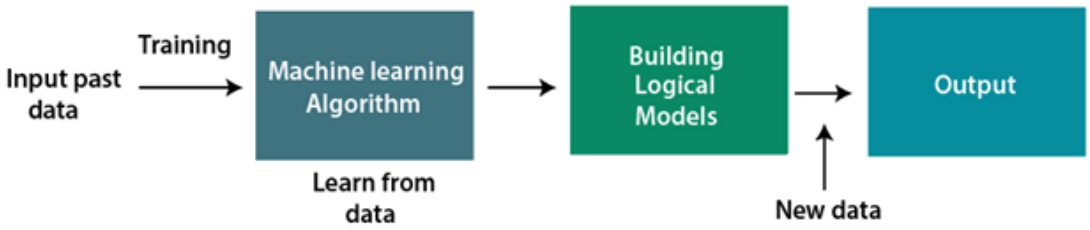


Fig1.1:Machine Learning Architecture

1.1.1. 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.*

1.1.2. 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

Supervised Learning

Supervised learning is a type of machine learning method in which we provide sample labeled 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 labeled 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. The supervised learning is based on supervision, and it is the same as when a student learns things in the supervision of the teacher. The example of supervised learning is **spam filtering**.

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

- **Classification**
- **Regression**

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 labeled, 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.

In unsupervised learning, we don't have a predetermined result. The machine tries to find useful insights from the huge amount of data.

It can be further classifieds into two categories of algorithms:

- **Clustering**
- **Association**

1.2. 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. If you ever diagrammed sentences in grade school, you've done these tasks manually before. 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.

1.3. 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. However, primarily, it is used for Classification problems in Machine Learning. 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 a hyperplane.

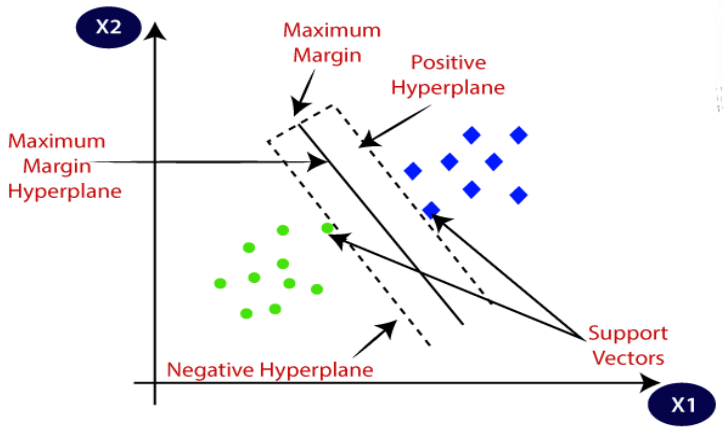


Fig 1.2:SVM

SVM chooses the extreme points/vectors that help in creating the hyperplane. 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 hyperplane:

Hyperplane: There can be multiple lines/decision boundaries to segregate the classes in n-dimensional space, but we need to find out the best decision boundary that helps to classify the data points. This best boundary is known as the hyperplane of SVM.

The dimensions of the hyperplane depend on the features present in the dataset, which means if there are 2 features (as shown in image), then hyperplane will be a straight line. And if there are 3 features, then hyperplane will be a 2-dimension plane.

We always create a hyperplane that has a maximum margin, which means the maximum distance between the data points.

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

1.4. 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:

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

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

1.5 Problem Statement

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. This noise in actual data limits the accuracy of generalization that can be achieved, irrespective of how extensive the training set is.

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.

DISADVANTAGES:

- 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.

CHAPTER 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 94% 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 97.9% 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 3
DEVELOPMENT PROCESS

3.1 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.

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. The source code is freely available and open for modification and reuse. Python has a significant number of users.

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
- Interpreted Language

3.2 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 Anaconda 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 Anaconda and the pip package manager is in how package dependencies are managed, which is a significant challenge for Python data science and the reason Anaconda 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.

In contrast Anaconda analyses the current environment including everything currently installed, and, together with any version limitations specified (e.g. the user may wish to have Tensorflow version 2,0 or higher), works out how to install a compatible set of dependencies, and shows a warning if this cannot be done.

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 Anaconda 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 Anaconda environment using pip, and Anaconda will keep track of what it has installed itself and what pip has installed.

Custom packages can be made using the Anaconda 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 Anaconda.

3.2.1 Anaconda Navigator

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows users to launch applications and manage Anaconda 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:[16]

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

3.2.2 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. A Jupyter Notebook document is a JSON document, following a versioned schema, containing an ordered list of input/output cells which can contain code, text (using Markdown), mathematics, plots and rich media, 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.

3.3 RESOURCE REQUIREMENTS:

SOFTWARE REQUIREMENTS:

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

HARDWARE REQUIREMENTS:

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

3.4 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 ++orks.On this paper we presented a machine learning & natural language processing system to observe the unreliable users in on-line social networks. Moreover, we are adding the SVM classifier algorithm to increase the detection accuracy rate of the fake profiles.In our research paper, as stated earlier, we will be emphasizing on the SVM algorithm and how it is used in fake news detection systems.

ADVANTAGES

- High accuracy is obtained and 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.

3.5 SYSTEM ARCHITECTURE

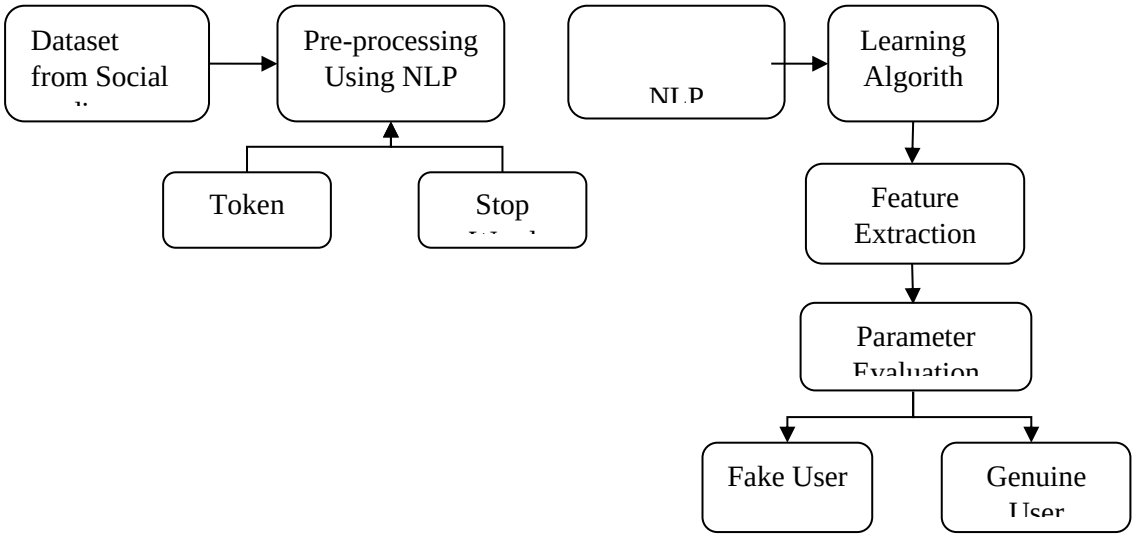


Fig 3.1:System Architecture

The presented process used Facebook profile to notice false profiles. The working method of the proposed procedure includes three principal phases;

1. NLP Pre-processing
2. Principal Component Analysis (PCA)
3. Learning Algorithm

3.6 MODULE DESCRIPTION

- MODULE 1: Dataset
- MODULE 2: Data preprocessing
- MODULE 3: Principle component Analysis (PCA)
- MODULE 4: Training the Model using algorithm
- MODULE 5: Evaluation

MODULE 1: Data collection

To collect the Dataset from Facebook profiles. Collecting data for training the ML model is the basic step in the machine learning pipeline. The predictions made by ML systems can only be as good as the data on which they have been trained. Following are some of the problems that can arise in data collection:

- Inaccurate data. The collected data could be unrelated to the problem statement.
- Missing data. Sub-data could be missing. That could take the form of empty values in columns or missing images for some class of prediction.
- Data imbalance. Some classes or categories in the data may have a disproportionately high or low number of corresponding samples. As a result, they risk being under-represented in the model.
- Data bias. Depending on how the data, subjects and labels themselves are chosen, the model could propagate inherent biases on gender, politics, age or region, for example. Data bias is difficult to detect and remove.

MODULE 2: PRE-PROCESSING

Once the data is extracted from the twitter source as the datasets, this information has to be passed to the classifier. The classifier cleans the dataset by removing redundant data like stop words, emoticons in order to make sure that non textual content is identified and removed before the analysis.

Text pre-processing is an essential a part of any NLP method and the significance of the NLP pre-processing are

- To minimize indexing (or knowledge) records dimension of the textual content records
    1. Stop words bills 20-30% of total phrase counts in a special textual content record
    2. Stemming may just diminish indexing size as much as forty- 50%
- To make stronger the efficiency and effectiveness of the IR method
    1. Stop words aren't valuable for shopping or textual content mining
    2. Stemming used for matching the similar words in a text record

Tokenization:

Tokenization is the process of breaking a circulate of textual content into phrases, phrases, symbols, or different significant factors called tokens. The aim of the tokenization is the exploration of the phrases in a sentence. The list of tokens turns into input for further processing akin to parsing or textual content mining. Tokenization is valuable both in linguistics (where it's a form of textual content segmentation), and in laptop science, the place it forms a part of lexical analysis. Textual knowledge is simplest a block of characters at the starting.

All strategies in know-how retrieval require the words of the data set. For that reason, the requirement for a parser is a tokenization of records. This might be sound trivial because the text is already saved in computing device-readable codecs. However, some problems are nonetheless left, like the removing of punctuation marks. Different characters like brackets, hyphens, and so on require processing as well.

Stop word Removal:

Stop phrases are very more often than not used fashioned phrases like ‘and’, ‘are’, ‘this’ etc. They don't seem to be useful in classification of records. So, they must be removed. However, the development of such stop phrases record is problematic and inconsistent between textual sources. This process also reduces the text knowledge and improves the approach performance. Each textual content report offers with these phrases which are not vital for text mining applications.

Stemming and Lemmatization:

The aim of both stemming as well as lemmatization is to scale down inflectional types & mostly derivationally associated varieties of a phrase to a fashioned base kind.

Stemming usually refers to a crude heuristic process that chops off the ends of words in the hope of accomplishing this goal accurately more often than not, and quite often involves the removal of derivational affixes.

Lemmatization often refers to doing matters competently with the usage of a vocabulary and morphological analysis of phrases, in most cases aiming to eliminate inflectional endings only and to come back the base or dictionary type of a word, which is often called the lemma.

MODULE 3: PCA

Principal Component Analysis purpose is to extract the fundamental understanding from the table, to symbolize it as a suite of new orthogonal variables known as major accessories, and to show the sample of similarity of the observations and of the variables as elements in maps.

MODULE 4: Train the Model using Algorithm

In this proposed system we are using two machine learning algorithms named as Support Vector Machine (SVM) and naïve Bayes algorithms.

Support Vector Machine (SVM):

An SVM classifies information by means of finding the exceptional hyperplane that separates all information facets of 1 type from those of the other classification.

The best hyperplane for an SVM method that the one with the biggest line between the two classes. An SVM classifies data through discovering the exceptional hyperplane that separates all knowledge facets of one category from those of the other class. The help vectors are the info aspects which are closest to the keeping apart hyperplane.

Naive Bayes:

Naive Bayes algorithm is the algorithm that learns the chance of an object with designated features belonging to a unique crew/category. In brief, it's a probabilistic classifier.

The Naive Bayes algorithm is called "naive" on account that it makes the belief that the occurrence of a distinct feature is independent of the prevalence of other aspects. For illustration, if we're looking to determine false profiles based on its time, date of publication or posts, language and geo-position. Even if these points depend upon each and every different or on the presence of the other facets, all of these properties in my view contribute to the probability that the false profile.

MODULE 5: EVALUATION

- 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.
- To classify the fake profile or genuine profiles in Facebook

CHAPTER 4
TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub – assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

4.1. TYPES OF TESTS

4.1.1. UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.

- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

4.1.2. INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

4.1.3. FUNCTIONAL TEST

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised

Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

4.1.4. SYSTEM TEST

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

4.1.5. WHITE BOX TESTING

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

4.1.6. BLACK BOX TESTING

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

CHAPTER 5
CONCLUSION

In this paper, 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 paper we took the Facebook 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 paper.

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.

APPENDIX

(A)SAMPLE SOURCE CODE:


```
 "cells":[
 {
   "cell_type": "markdown",

"metadata":{},
  "source":[
 "#AnInstagramAccountisFakeorGenuine"
  ]
 },
 {
   "cell_type": "code",

 "execution_count":111,
  "metadata":{},

 "outputs":[],
  "source":[
 "#ImportPackages\n",
 "importnumpyasnp\n",
  "importpandasaspd\n",
 "importmatplotlib.pyplotasplt\n",
    "import seaborn as sns"

  ]
 },
 {
  "cell_type":"code",
  "execution_count":112,
  "metadata":{},
  "outputs":[],
  "source":[
 "defload_train_data():\n", \n",
 "train_data=pd.read_csv('train.csv',header=0)\n",
 " \n",
 " X_train=train_data.drop(columns='fake')\n",
 " y_train=train_data['fake']\n",
 " \n",
 " returnX_train,y_train"
  ]
 },
  {
```

```
"cell_type":"code",
 "execution_count":113,
 "metadata":{},
 "outputs":[
{
 "data":{
"text/html":[
 "<div>\n",  "<stylescoped>\n",
 " .dataframetbodytrth:only-of-type{\n",
 "vertical-align:middle;\n",
 " }\n",
 "\n",
 " .dataframetbodytrth{\n",vertical-align:top;\n",
" }\n", "\n",
" .dataframetheadth{\n",
"text-align:right;\n",
 " }\n",
 "</style>\n",
 "<tableborder=\"1\"class=\"dataframe\">\n",
 " <thead>\n",
 " <trstyle=\"text-align:right;\">\n",
 " <th></th>\n",
"<th>profilepic</th>\n",
" <th>nums/lengthusername</th>\n",
 " <th>fullnamewords</th>\n",
" <th>nums/lengthfullname</th>\n",
 " <th>name==username</th>\n",
 " <th>descriptionlength</th>\n",
 " <th>externalURL</th>\n",
 "<th>private</th>\n",
 " <th>#posts</th>\n",
 " <th>#followers</th>\n",
 " <th>#follows</th>\n",
"  </tr>\n",

" </thead>\n",
 " <tbody>\n",
 " <tr>\n",
 " <th>190</th>\n",
 " <td>1</td>\n",
 " <td>0.18</td>\n",
 " <td>1</td>\n",
 " <td>0.0</td>\n",
```

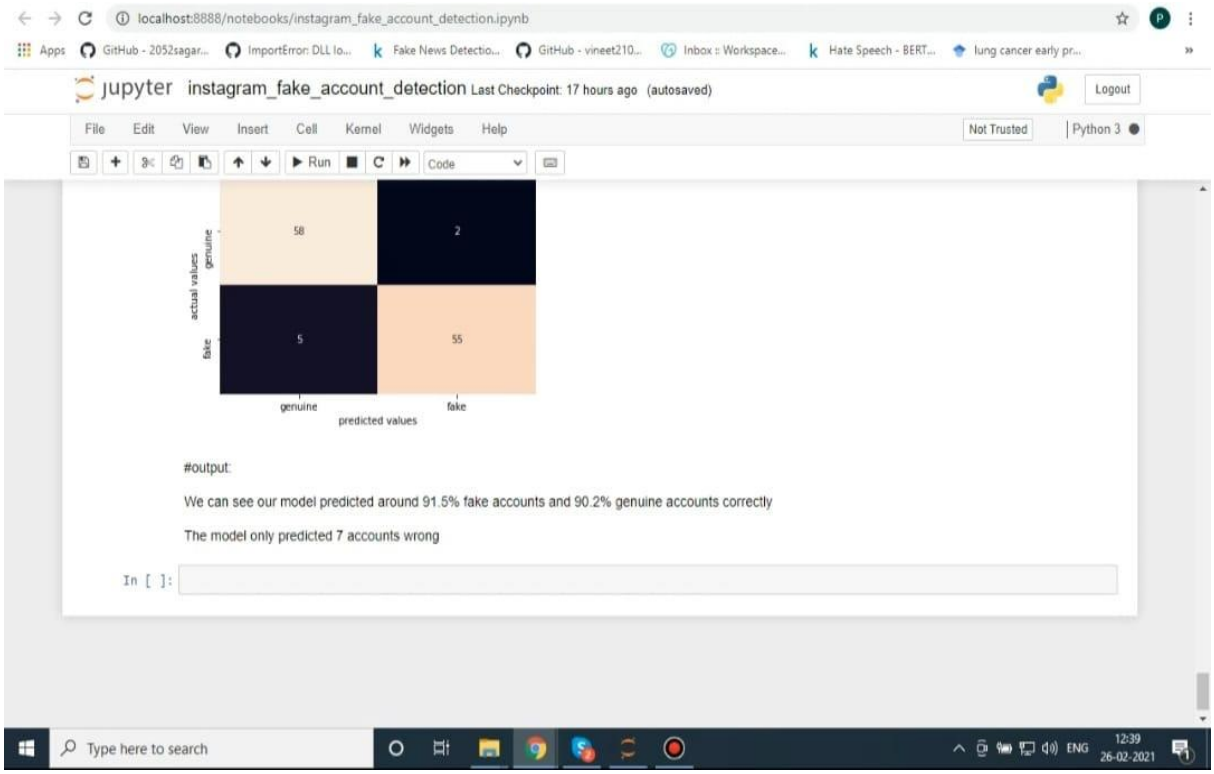
```
 " <td>0</td>\n",
"  <td>70</td>\n",
" <td>0</td>\n",
 " <td>1</td>\n",
 " <td>93</td>\n",
" <td>67</td>\n",
" <td>149</td>\n",
" </tr>\n",
 " <tr>\n",
 " <th>77</th>\n",
 " <td>1</td>\n",
 " <td>0.00</td>\n",
 " <td>2</td>\n",
 " <td>0.0</td>\n",
 " <td>0</td>\n",
" <td>18</td>\n",
" <td>0</td>\n",
"<td>1</td>\n"
" <td>14</td>\n",
" <td>178</td>\n",
" <td>245</td>\n",
" </tr>\n",
 " <tr>\n",
" <th>302</th>\n",
 " <td>0</td>\n",
 "<td>0.12</td>\n",
 "<td>1</td>\n",
 "<td>0.0</td>\n",
 "<td>0</td>\n",
 "<td>0</td>\n",
 "<td>0</td>\n",
"<td>1</td>\n",
 "<td>0</td>\n",
" <td>31</td>\n",
"<td>213</td>\n",
" </tr>\n",
" <tr>\n",
" <th>330</th>\n",
"<td>0</td>\n",
" <td>0.36</td>\n",
"<td>1</td>\n",
"<td>0.0</td>\n",
```

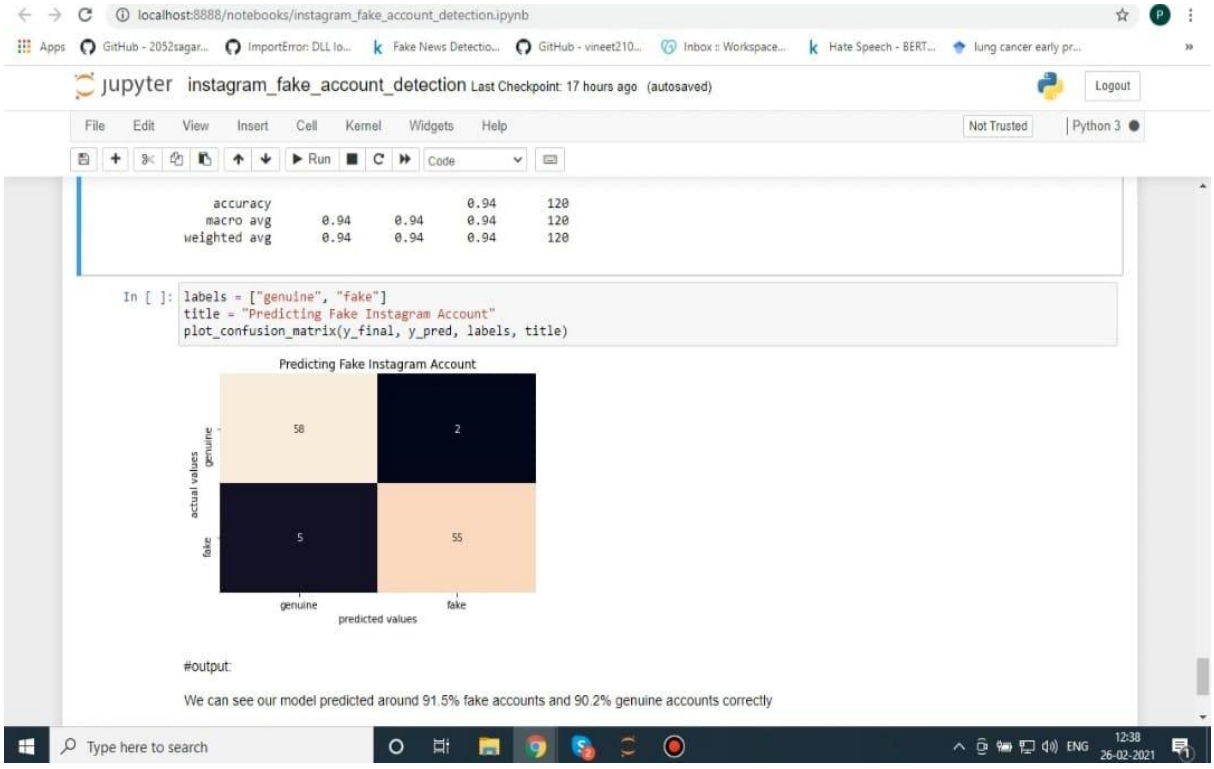
"<td>0</td>\n",
"<td>0</td>\n",
"<td>0</td>\n",
"<td>0</td>\n",
"<td>0</td>\n",
"<td>21</td>\n",
"<td>44</td>\n",
"</tr>\n",
"<tr>\n",
" <th>462</th>\n",
"<td>1</td>\n",
"<td>0.83</td>\n",
"<td>1</td>\n",
"<td>0.0</td>\n",
" <td>0</td>\n",
" <td>32</td>\n",
"<td>0</td>\n",
"<td>0</td>\n",
"<td>4</td>\n",
"<td>61</td>\n",
"<td>76</td>\n",
"</tr>\n",
"<tr>\n",
"<th>...</th>\n",
"<td>...</td>\n",
"<td>...</td>\n",
"<td>...</td>\n",
"<td>...</td>\n",
"<td>...</td>\n",
"<td>...</td>\n",
"<td>...</td>\n",
"<td>...</td>\n",
"<td>...</td>\n",
" <td>...</td>\n",
" <td>...</td>\n",
" <td>...</td>\n",
"</tr>\n",
"<tr>\n",
"<th>70</th>\n",
"<td>1</td>\n",
"<td>0.33</td>\n",
"<td>2</td>\n",
"<td>0.0</td>\n",
"<td>0</td>\n",

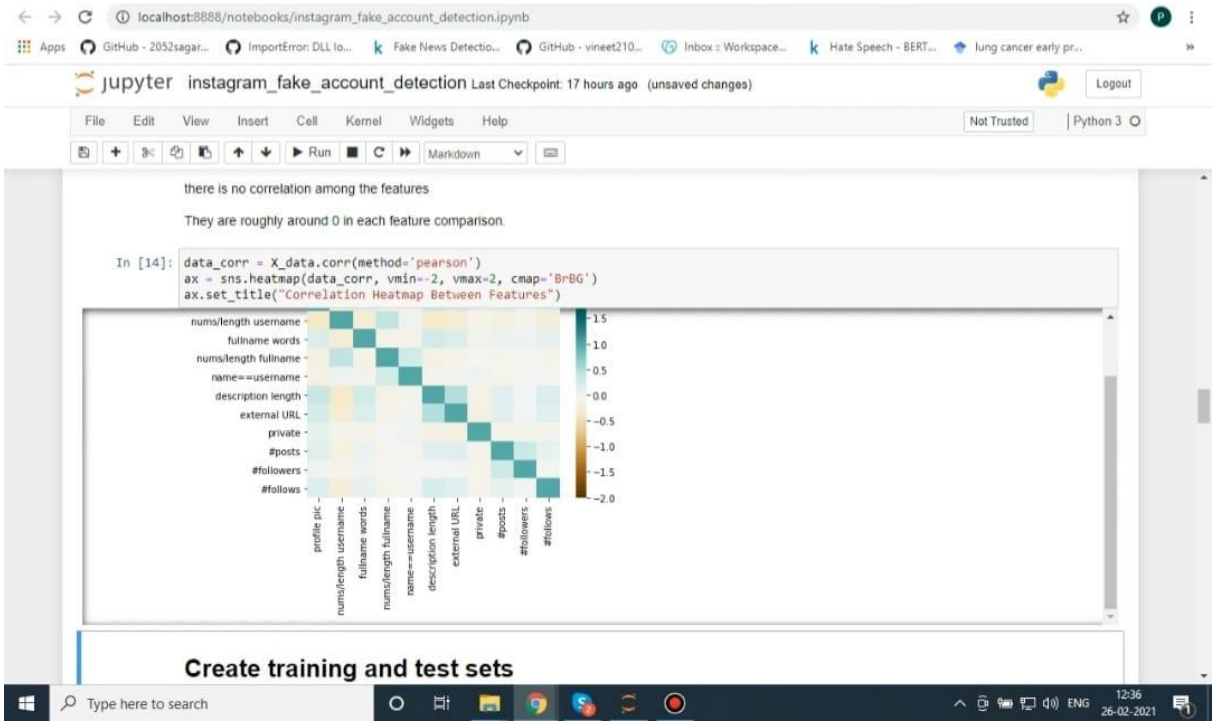
"<td>70</td>\n",
"<td>0</td>\n",
"<td>0</td>\n",
"<td>74</td>\n",
"<td>399</td>\n",
"<td>452</td>\n",
"</tr>\n",
"<tr>\n",
"<th>132</th>\n",
"<td>1</td>\n",
"<td>0.00</td>\n",
"<td>2</td>\n",
"<td>0.0</td>\n",
"<td>0</td>\n",
"<td>149</td>\n",
"<td>0</td>\n",
"<td>1</td>\n",
"<td>92</td>\n",
"<td>484</td>\n",
"<td>3296</td>\n",
"</tr>\n",
"<tr>\n",

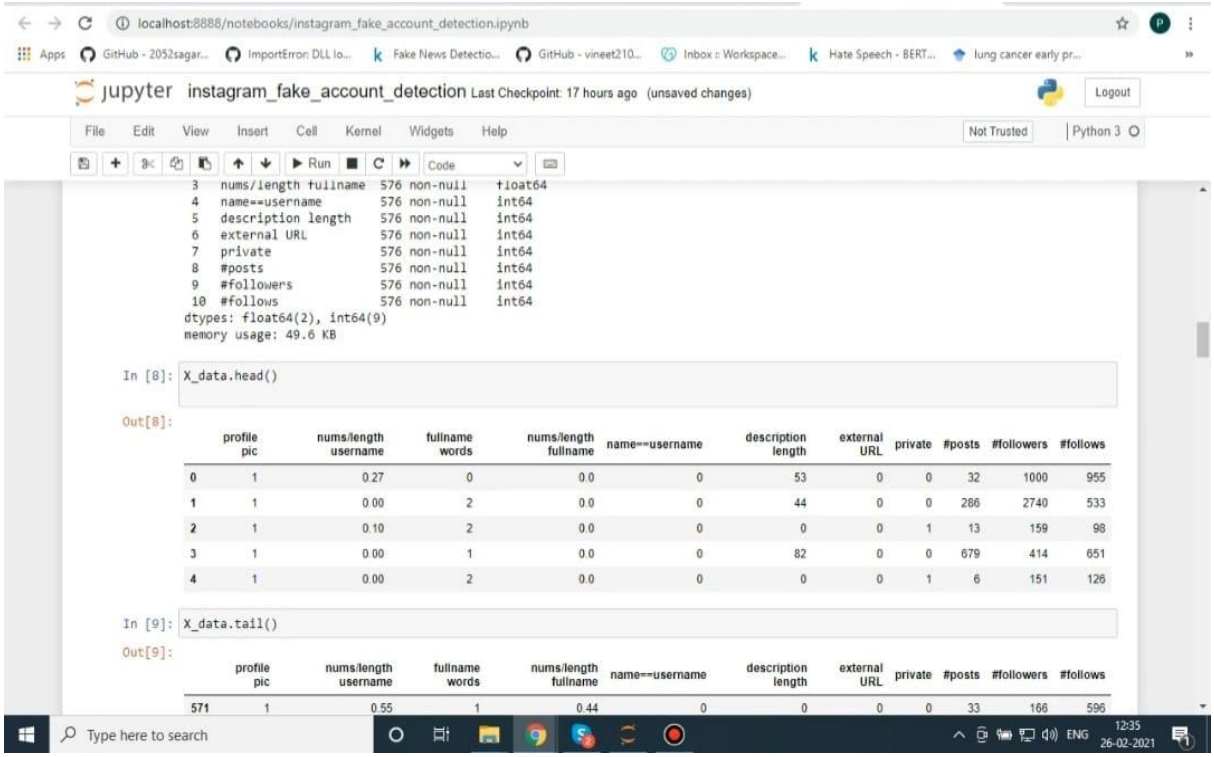
" <th>289</th>\n",
"<td>0</td>\n",
"<td>0.38</td>\n",
"<td>1</td>\n",
"<td>0.0</td>\n",
"<td>0</td>\n",
"<td>0</td>\n",
"<td>0</td>\n",
"<td>0</td>\n",
"<td>0</td>\n",
"<td>60</td>\n",
"<td>31</td>\n",
"</tr>\n",
"<tr>\n",
"<th>109</th>\n",
"<td>1</td>\n",
"<td>0.00</td>\n",
"<td>2</td>\n",
"<td>0.0</td>\n",
"<td>0</td>\n",

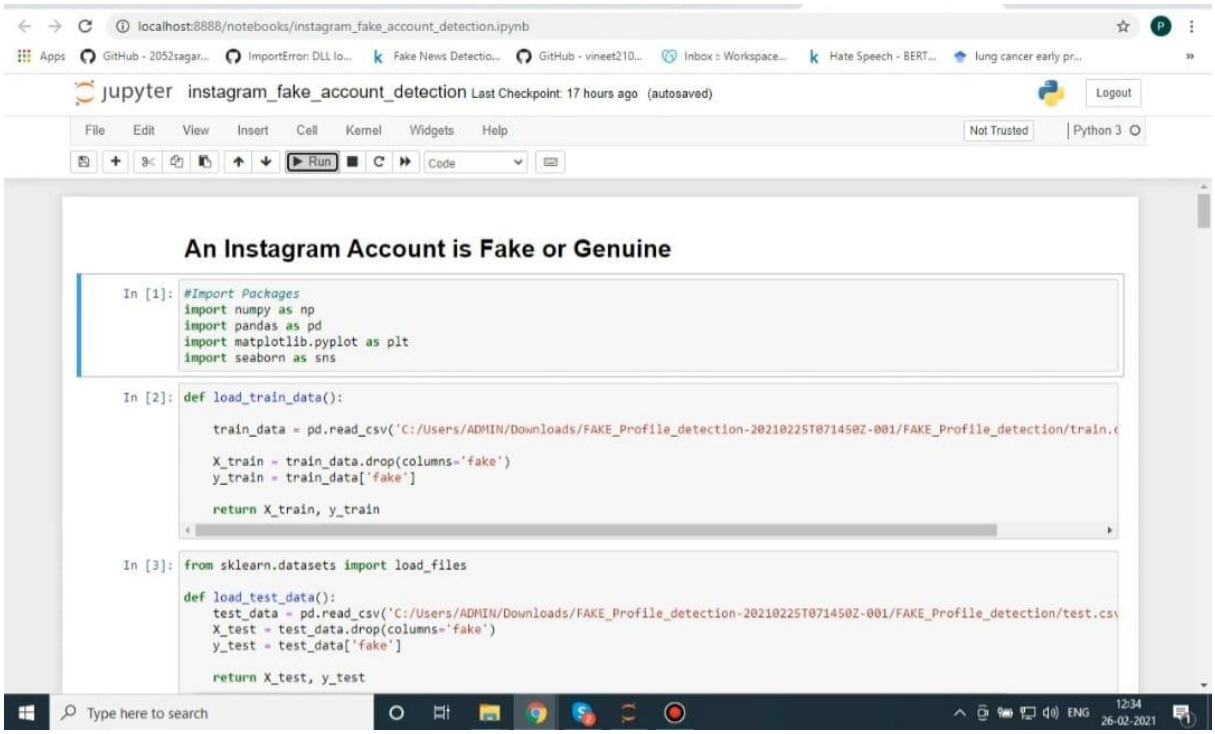
B)SNAPSHOTS:


localhost:8888/notebooks/instagram_fake_account_detection.ipynb
jupyter instagram_fake_account_detection Last Checkpoint: 17 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Code
58
2
5
55
actual values
genuine
fake
genuine
fake
predicted values
#output:
We can see our model predicted around 91.5% fake accounts and 90.2% genuine accounts correctly
The model only predicted 7 accounts wrong
In [ ]:
Type here to search
ENG
12:39
26-02-2021


localhost:8888/notebooks/instagram_fake_account_detection.ipynb
jupyter instagram_fake_account_detection Last Checkpoint: 17 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Code
accuracy 0.94 120
macro avg 0.94 0.94 0.94 120
weighted avg 0.94 0.94 0.94 120
In [ ]:
labels = ["genuine", "fake"]
title = "Predicting Fake Instagram Account"
plot_confusion_matrix(y_final, y_pred, labels, title)
Predicting Fake Instagram Account
58
2
5
55
actual values
genuine
fake
genuine
fake
predicted values
#output:
We can see our model predicted around 91.5% fake accounts and 90.2% genuine accounts correctly
Type here to search
ENG
12:38
26-02-2021


localhost:8888/notebooks/instagram_fake_account_detection.ipynb
jupyter instagram_fake_account_detection Last Checkpoint: 17 hours ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Markdown
there is no correlation among the features
They are roughly around 0 in each feature comparison.
In [14]:
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")
Correlation Heatmap Between Features
nums/length username
fullname words
nums/length fullname
name==username
description length
external URL
private
#posts
#followers
#follows
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
Create training and test sets
Type here to search
ENG
12:36
26-02-2021


localhost:8888/notebooks/instagram_fake_account_detection.ipynb
jupyter instagram_fake_account_detection Last Checkpoint: 17 hours ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Code
3 nums/length fullname 576 non-null float64
4 name==username 576 non-null int64
5 description length 576 non-null int64
6 external URL 576 non-null int64
7 private 576 non-null int64
8 #posts 576 non-null int64
9 #followers 576 non-null int64
10 #follows 576 non-null int64
dtypes: float64(2), int64(9)
memory usage: 49.6 KB
In [8]:
X_data.head()
Out[8]:
profile pic nums/length username fullname words nums/length fullname name==username description length external URL private #posts #followers #follows
0 1 0.27 0 0.0 0 53 0 0 32 1000 955
1 1 0.00 2 0.0 0 44 0 0 286 2740 533
2 1 0.10 2 0.0 0 0 0 1 13 159 98
3 1 0.00 1 0.0 0 82 0 0 679 414 651
4 1 0.00 2 0.0 0 0 0 1 6 151 126
In [9]:
X_data.tail()
Out[9]:
profile pic nums/length username fullname words nums/length fullname name==username description length external URL private #posts #followers #follows
571 1 0.55 1 0.44 0 0 0 0 33 166 596
Type here to search
ENG
12:35
26-02-2021



(C)PLAGARISM REPORT:

journal paper (1)

ORIGINALITY REPORT

10%	5%	6%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Universiti Teknologi MARA Student Paper	2%
2	"International Conference on Intelligent Data Communication Technologies and Internet of Things (ICICI) 2018", Springer Science and Business Media LLC, 2019 Publication	1%
3	www.ijarse.com Internet Source	1%
4	"Advances in Big Data and Cloud Computing", Springer Science and Business Media LLC, 2019 Publication	1%
5	zenodo.org Internet Source	1%
6	Submitted to Indian Institute of Technology, Madras Student Paper	1%
7	Submitted to Aston University Student Paper	1%