

FAKE INSTAGRAM PROFILE DETECTION AND CLASSIFICATION USING MACHINE LEARNING

This project report submitted in partial fulfillment of the requirements for the

award of the degree of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

Submitted by

BATCH - [2019 - 2023]

Thurupu Tejasri (19K91A1247)

Nayanolla Nikitha (19K91A1233)

Mohammed Abdul Ahad Siddiqui (19K91A1231)

Under the guidance of

Mrs.B Shivani

Assistant professor



**TKR COLLEGE OF ENGINEERING AND TECHNOLOGY
AUTONOMOUS**

Approved By AICTE. Affiliated to JNTUH, Accredited by NBA & NAAC with 'A' Grade

Medbowli, Meerpet, Saroornagar, Hyderabad - 500097

[2022-2023]



TKR COLLEGE OF ENGINEERING AND TECHNOLOGY

(Sponsored by TKR Educational Society, Approved by AICTE, Affiliated by JNTUH)

Autonomous, Accredited by NAAC with 'A' Grade. Accredited by NBA

Medbowli, Meerpet, Saroornagar, Hyderabad - 500 097

Phone: 9100377790, e-mail: info@tkrcet.ac.in website: www.tkrct.ac.in



College Code : K9

CERTIFICATE

This is to Certify that the project report entitled as **“FAKE INSTAGRAM PROFILE DETECTION AND CLASSIFICATION USING MACHINE LEARNING”** is the bonafide work of **“Thurupu Tejasri(19K91A1247), Nayanolla Nikitha(19K91A1233), Mohammed Abdul Ahad Siddiqui(19K91A1231)”** who carried out the project work under my supervision in partial fulfillment for the award of the Degree of Bachelor of Technology in **Information Technology** to the **TKR College Of Engineering and Technology** affiliated to the Jawaharlal Nehru Technological University, Hyderabad.

Mrs.B.Shivani
Assistant Professor
SUPERVISOR

Dr.N.Satyanarayana
Professor and HOD

EXAMINER



TKR COLLEGE OF ENGINEERING AND TECHNOLOGY

(Sponsored by TKR Educational Society, Approved by AICTE, Affiliated by JNTUH)

Autonomous, Accredited by NAAC with 'A' Grade. Accredited by NBA

Medbowli, Meerpet, Saroornagar, Hyderabad - 500 097

Phone: 9100377790, e-mail: info@tkrcet.ac.in website: www.tkrct.ac.in



College Code : K9

PLAGARISM REPORT

This is to certify that the project report entitled “**FAKE INSTAGRAM PROFILE
DETECTION AND CLASSIFICATION USING MACHINE LEARNING**”

submitted by

THURUPU TEJASRI [19K91A1247]

NAYANOLLA NIKITHA [19K91A1233]

MOHAMMED ABDUL AHAD SIDDIQUI [19K91A1231]

Is Checked for plagiarism and similarity obtained 15%

Mrs.B.Shivani
Assistant Professor
SUPERVISOR

Dr.N.Satyanarayana
Head of the Department
Information Technology

PLAGARISM DOCUMENT



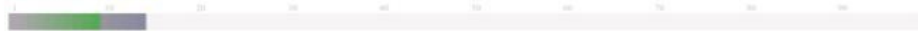
The Report is Generated by DrillBit Plagiarism Detection Software

Submission Information

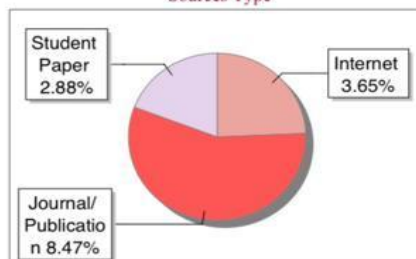
Author Name	19K91A1247,1233,1231
Title	Fake Instagram Profile Detection and Classifica..
Paper/Submission ID	725423
Submission Date	2023-04-21 15:42:50
Total Pages	43
Document type	Project Work

Result Information

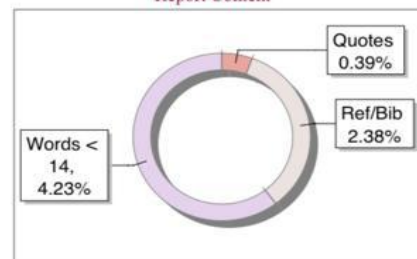
Similarity **15 %**



Sources Type



Report Content



Exclude Information

Quotes	Excluded
References/Bibliography	Excluded
Sources: Less than 14 Words Similarity	Excluded
Excluded Source	0 %
Excluded Phrases	Not Excluded

A Unique QR Code use to View/Download/Share Pdf File



ACKNOWLEDGMENT

Our sincere thanks and gratitude to our internal guide, **Mrs.B.SHIVANI** Assistant professor, Department of Information Technology, TKR College of Engineering and Technology for her constant guidance, encouragement and moral support throughout the project.

We are extremely thankful to **DR.N.SATYANARAYANA**, Head of the Department of Information Technology, TKR College of Engineering and Technology for the encouragement and support throughout the project.

We are also thankful and indebted to principal **DR.D.V.RAVI SHANKAR**, TKR College of Engineering and Technology for all the timely support and valuable suggestions during the period of our project.

Finally, we would also like to thank all the faculty and staff of IT Department who helped us directly or indirectly, and also parents and friends for their cooperation in completing the project work.

THURUPU TEJA SRI (19K91A1247)

NAYANOLLA NIKITHA (19K91A1233)

MOHAMMED ABDUL AHAD SIDDIQUI (19K91A1231)

DECLARATION

We hereby declare that this project report titled **FAKE INSTAGRAM PROFILE DETECTION AND CLASSIFICATION USING MACHINE LEARNING** submitted in partial fulfillment of the degree of **B.Tech** in **Information Technology** is a record of original work carried out by us under the supervision of **Mrs.B.Shivani**, Assistant Professor, Department of **Information Technology** and has not submitted to any other Institution or University for the award of any degree.

Date:

THURUPU TEJA SRI (19K91A1247)

NAYANOLLA NIKITHA (19K91A1233)

MOHAMMED ABDUL AHAD SIDDIQ (19K91A1231)

ABSTRACT

Instagram is today seen as a crucial medium for advertising, marketing, and social interaction as a result of the rise in Internet usage. Millions of people use it, but some of them tend to abuse it by creating phoney identities. Additionally, since followers are what determine a social media user's fame, individuals often utilise dishonest tactics to raise their profile's follower count. One of the major issues with Online Social Networks (OSNs) is fake interaction, which is used to artificially boost an account's popularity. Because phoney involvement causes businesses to lose money, inaccurate audience targeting in advertising, inaccurate product prediction systems, and an undesirable social network atmosphere, its detection is essential. This study focuses on the identification of automated and phoney Instagram profiles that generate phoney engagement. There was no publicly accessible dataset for automated and phoney accounts prior to our effort. Machine learning algorithms including Naive Bayes, Logistic Regression, Support Vector Machines, and Neural Networks are used to find these accounts. In order to address the dataset's unnatural bias, a cost-sensitive genetic algorithm is also suggested for the detection of automated accounts. Smote-nc technique is used to address the fake dataset's unevenness issue. The classification accuracy rates for the automated and bogus account detection datasets are 86% and 96%, respectively.

TABLE OF CONTENTS

Chapter No.	Name of the Content	Page No.
	Abstract	i
	List of Figures	
	List of Abbrevations	
1	INTRODUCTION	1
2	LITERATURE SURVEY	2
3	SYSTEM ANALYSIS	5
	3.1 System Requirement Analysis	5
	3.2 Existing System	6
	3.3 Proposed System	7
4	MODULES	8
	4.1 Gathering Data	9
	4.2 Data Preparation	9
	4.3 Data Wrangling	10
	4.4 Analyze Data	10
	4.5 Train the Model	11
	4.6 Testing the Model	11
	4.7 Deployment	11
5	SYSTEM DESIGN	12
	5.1. System Architecture	12
	5.2. Data Flow Diagrams	13
	5.3 UML Diagrams	14
	5.3.1 Use Case Diagram	15
	5.3.2 Sequence Diagram	16
	5.3.3 Class Diagram	17
	5.3.4 Activity Diagram	18
	5.3.5 Deployment Diagram	19

6	SYSTEM ENVIRONMENT	20
	6.1 Python	20
	6.2 Anaconda	20
7	IMPLEMENTATION	21
	7.1 Implementation	21
	7.2 Source Code	23
8	SCREENSHOTS and RESULTS	30
	8.1 Give the input to detect the profile is Fake or Real	30
	8.2 Given inputs to detect the profile is Fake or Real	31
	8.3 Real User	31
	8.4 Given inputs to detect the profile is Fake or Real	32
	8.5 Fake User	32
9	CONCLUSION	33
	REFERENCES	34

LIST OF FIGURES

Figure no.	Figure Name	Page no.
5.1	System Architecture	13
5.2	Data Flow Diagrams	14
5.4.1	Use Case Diagram	16
5.4.2	Sequence Diagram	17
5.4.3	Class Diagram	18
5.4.4	Activity Diagram	19
5.4.5	Deployment Diagram	20
8.1	Give the input to detect the profile is Real or Fake	30
8.2	Given inputs to detect the profile is Real or Fake	31
8.3	Real User	31
8.4	Given inputs to detect the profile is Real or Fake	32
8.5	Fake User	32

LIST OF ABBREVIATIONS

S.No	ABBREVIATION	DEFINITION
1	OSN	Online Social Networks
2	KNN	K-Nearest Neighbors Algorithm
3	SVM	Support vector Machine
4	UML	Unified Modeling Language
5	OMG	Object Management Group

1. INTRODUCTION

While many people have benefited from the full-size updates of records made available by the introduction of the Internet and social media, there has also been a full-size boom in the rise of cybercrimes, which are primarily directed at women. In the five years between 2011 and 2016, India saw a 457% increase in cybercrime, according to a 2019 report published inside the Economics Times. Most people think that's because of how social media, including Facebook, Instagram, and Twitter, has an impact on our daily life. While they merely aid in the development of a valid social network, the emergence of consumer debts on those websites typically only requires an email address. The ability of a real-life person to produce multiple bogus IDs makes it simple to produce impostors. With inside the digital international of social media, admission does no longer require this kind of checks, unlike the actual international state of affairs where multiple policies and guidelines are imposed to become aware of oneself in a completely unique manner (for example, at the same time as issuing one's passport or driver's licence). In this research, we investigate the particular ities of Instagram and attempt to utilise machine learning techniques, particularly Logistic Regression and Random Forest Algorithm, to determine whether an account is real or fraudulent.

2. LITERATURE SURVEY

Today, social media is expanding really quickly. These platforms are crucial for many people in society, especially for advertising campaigns and celebrities and politicians who try to market themselves on social media. Therefore, fictitious debts produced on behalf of people or organisations could be risky, undermine their reputations, and ultimately result in a decline in the number of people who actually like and support them. Additionally, a variety of phoney profiles are detrimental to social media's advantages for advertising and for businesses using it. Real customers also have a lot of concerns about their privacy in the online environment because of these phoney personas, which could be a method for cyberbullying.

As a result, numerous researchers have looked into the challenge of identifying spammers and bad actors using technological learning techniques in social media during the past few years. There are, however, just a small number of research articles discussing how to identify fraudulent fans or money owing. In this section, we shed light on recent additions of false debt responses and spammers.

Ferrara et al. added a technique to stumble on bot customers on Twitter primarily based totally at the fantastically shared capabilities that distinguish them from valid customers. To classify debts into the valid or bot class, they have employed a device to learn about the approaches and behavioural patterns between valid and bot debt.

Cresci et al. have created and used a baseline dataset of verified human and fake fans on Twitter. They used the baseline dataset in their research to train a collection of machine learning classifiers that were built using reviewed media-use-related policies and capabilities. Their suggested method is effective at identifying phoney debts; results obtained using their method show that it can correctly classify more than 95% of the debts from the real education set.

In a barely extraordinary approach, **Zhang and Lu.** Introduced a unique approach for the detection of fake money owed in Weibo. Their proposed answer has extraordinary aspects. At first, that they'd this premise why such money owed exist with inside the first place. In the second, they investigated the overlap among fans listing of the clients of fake fans, and that they located a excessive overlap among their follower lists. Their research located 395 nearduplicates, which caused 11.90 million fake money owed that despatched 1,000,000 hyperlinks with inside the network.

Thomas et al. made a group of 1.8 million tweets despatched via way of means of 32.9 million Twitter money owed. In their research, they located Twitter suspended approximately 1.1 million of these money owed. They have decided on randomly a hundred of these money owed to research their tweets and confirm they had been spamming money owed.

They made a similarly evaluation on that a hundred decided on money owed, and that they locate 93 of the chosen money owed had been suspended for posting junk mail and the unsolicited commercial of numerous products. Three different money owed had been suspended for re-tweeting content material of extraordinary information money owed, and the alternative four remained money owed had been suspended for reproduction and competitive advertising posts.

Gao et al. have used a fixed of capabilities for efficiently reconstructing junk mail tweets into campaigns in place of studying them separately. The end result suggests their proposed answer received. However, the disadvantage in their approach is its low detection accuracy.

Benevenuto et al. proposed a option to stumble on spammers from non spammers. In their approach, they used an SVM classifier, that is a supervised gadget gaining knowledge of algorithm. They have used 23 conduct and 39 content material capabilities to distinguish spammers from nonspammers, and that they accomplished experiments via way of means of 5-fold cross-validations. The experiments display they had been nearly a success in figuring out spammers from non-spammers.

Bala Anand et al. advanced a new gadget to stumble on fake customers at the Twitter platform the use of a graph-primarily based totally semi-supervised gaining knowledge of algorithm (EGSLA) and examine and amassing behavioral and person-generated content material (UGC) information. The version first gathered customers information, analyzed them to extract beneficial capabilities, and then accomplished type on those capabilities and made decisions. The experimental results show that the EGSLA method achieved high performance and became more beneficial than other algorithms, including decision trees, KNN, SVM, and game theory-based techniques, in terms of type correctness.

Sahoo et al. provided a hybrid version to stumble on malicious profiles on social media specializing in Twitter. The proposed hybrid version consists of modules; first, they analyzed and extracted capabilities the use of Petri internet structure, then they used those capabilities because the classifiers enter to categorize profiles as malicious and legitimates classes. The experimental outcomes display that the proposed method effectively distinguished extraordinary twitter money owed and received a excessive detection fee in phrases of type accuracy. Therefore, in step with literature, many researchers have been the use of gadget gaining knowledge of strategies to conquer security issues in social networks. Surveyed research typically focused on junk mail detection on microblogging social media. They have investigated many answers to resolve the hassle of junk mail and Fake money owed on Twitter and different microblogging social media. However, to date, there's no complete answer to fake money owed at the Instagram platform, that is one in every of the motivations at the back of this study. In order to effectively classify unusual Instagram person debt, we have presented a green method for identifying phoney debt on the Instagram platform .

3. SYSTEM ANALYSIS

3.1 Software Requirement Analysis

3.1.1 Functional Requirements

These are the specifications that the end user has asked the system to meet as a minimum set of features. Each of these features shall be a part of the System to the extent provided in the contract. These are shown or articulated in terms of the input to be provided to the system, the action taken, and the results anticipated. They are essentially the user-stated criteria that, in contrast to non-functional requirements, are visible immediately in the finished product.

- Data Collection
- Handling Missing Values
- Remove Duplicate rows
- Data Cleaning
- Data Labeling
- Data Scaling
- Finding Feature Importance
- Data Visualization
- Model Creating
- Model Training
- Hyper parameter tuning
- Model boosting and bagging
- Model Evaluation
- Model Deployment

3.1.2 Non-Functional Requirements

In essence, they are the requirements for quality that the system must meet in accordance with the project contract. Depending on the project, these criteria may be prioritised differently or applied to a different degree. They are also called non-behavioral requirements.

They basically deal with issues like:

- Portability
- Security
- Maintainability
- Reliability
- Scalability
- Performance
- Reusability
- Flexibility

3.1.3 Software Requirements

- Programming Language : Python
- IDE : PyCharm
- UML Design : Start UML
- Tools : PIP

3.1.4 Hardware Requirements

- Processor : Intel i3 and above
- RAM : 4GB and Higher
- Hard Disk : 500GB: Minimum

3.2 EXISTING SYSTEM

To determine if an account is authentic or fake, only few factors are taken into consideration. These factors play prime role in decision making. The accuracy of the model degrades when the number of factors is low. The existing systems lack in such a case, as the parameters considered by them seems to be outdated. Fake account creation has greatly improved due to improvements in the tools accessible for fake involvement. The apps that are currently being utilised to identify bogus accounts frequently miss this. Hence, the existing methods have turned outdated. The most commonly used algorithm by existing fake account detection tools is the Random forest algorithm. When accurate inputs are available and there are no missing inputs, it works perfectly.

The algorithm has few drawbacks. In case some of the inputs are missing, it gets so tedious for random forest algorithm to give the results.

DISADVANTAGES OF EXISTING SYSTEM

The algorithm has few drawbacks. In case some of the inputs are missing, it gets so tedious for random forest algorithm to give the results. The algorithms used in this existing system, if some of the inputs are not appropriate, the algorithm could not produce accurate results. Also, When there is an increase in the number of trees, the algorithm's time efficiency takes a hit. So, in this study we applied the gradient boosting approach.

3.3 PROPOSED SYSTEM

The existing system uses random forest algorithm to identify the fake account. In proposed system we are using Gradient Boosting Algorithm. It has been suggested that phoney Instagram profiles be automatically identified in order to protect Instagram users' social lives. supervised learning machine algorithms make it easier to predict phoney Instagram profiles. Fake profile IDs are classified and saved in a data dictionary so that the relevant authorities can take the appropriate action against fraudulent social media profiles. The classification techniques that were utilised to train the dataset have been compared through experimentation. The factors used by the existing systems to detect the fake accounts are very less. The prediction becomes accurate when the number of parameters used are more efficient. In previously used algorithms, if some of the inputs are not appropriate, the algorithm could not produce accurate results. As a result, we employed the gradient boosting approach in this study. Decision trees are one of the main factors in it. We made use of several parameters. These parameters are further considered as inputs which are used to form the decision trees in order to apply gradient boosting algorithm.

ADVANTAGES OF PROPOSED SYSTEM

This algorithm gives us an output even if some inputs are missing. This is the major reason for choosing this algorithm. Due to use of this algorithm we were able to get highly accurate results.

4.MODULES

- Gathering Data
- Data preparation
- Data Wrangling
- Analyze Data
- Train the model
- Test the model
- Deployment

4.1 Gathering Data

The first stage of the machine learning life cycle is data gathering. This step's objective is to identify and collect all data-related issues.

In this step, we need to identify the different data sources, as data can be collected from kaggle such as csv files. It is one of the most crucial phases of the life cycle. The quantity and quality of the collected data will determine the efficiency of the output. The more data there is, the more accurate the prediction will be.

This step includes the below tasks:

- Identify various data sources
- Collect data
- Integrate the data obtained from different sources

By performing the previous task, we obtain a cohesive set of data, also known as a dataset. It will be applied in following actions.

4.2 Data preparation

We must prepare the data for further steps after gathering it. Data preparation is the process of organising and preparing our data for use in machine learning training.

In this stage, we initially group all the data together before randomly arranging them.

This method can be separated into two different steps:

- Data exploration:

It is used to understand the nature of data that we have to work with. We must know the qualities, formats, and characteristics of the data.

A more accurate known of the data leads to successful results. We discover correlations, broad trends, and outliers in this.

- Data pre-processing:

The pre-processing of data for its analysis is the next stage.

4.3 Data Wrangling

Cleaning and transforming raw data into a usable format is the process of "data wrangling". It is the process of cleaning the data, selecting the variable to use, and transforming the data in a proper format to make it more suitable for analysis in the next step. It is one of the most crucial steps in the entire procedure. In order to address the quality issues, data cleaning is necessary.

The information we have gathered may not always be beneficial to us; some of it may not even be. The challenges that acquired data may have in real-world applications include:

- Missing Values
- Duplicate data
- Invalid data
- Noise

As a result, we clean the data using a variety of filtering methods.

Because the above problems could compromise the effectiveness of the conclusion, they must be found and fixed.

4.4 Data Analysis

The data has now moved on to the analysis stage after being cleaned and prepared. This step involves:

- Selection of analytical techniques
- Building models
- Review the result

The aim of this step is to build a machine learning model to analyze the data using various analytical techniques and review the outcome. The first step is to categorise the difficulties, after which we choose machine learning approaches like classification. Using the prepared data, create the model, then assess it.

Hence, in this step, we take the data and use machine learning algorithms to build the model.

4.5 Train Model

The model must now be trained in order to be improved for a better solution to the problem.

We use datasets to train the model with different machine learning algorithms. A model must be trained in order for it to comprehend the different patterns, laws, and features.

4.6 Test Model

We test the machine learning model once it has been trained on a specific dataset. In this phase, we give our model a test dataset to see if it is accurate.

According to the needs of the project or challenge, testing the model determines its accuracy %.

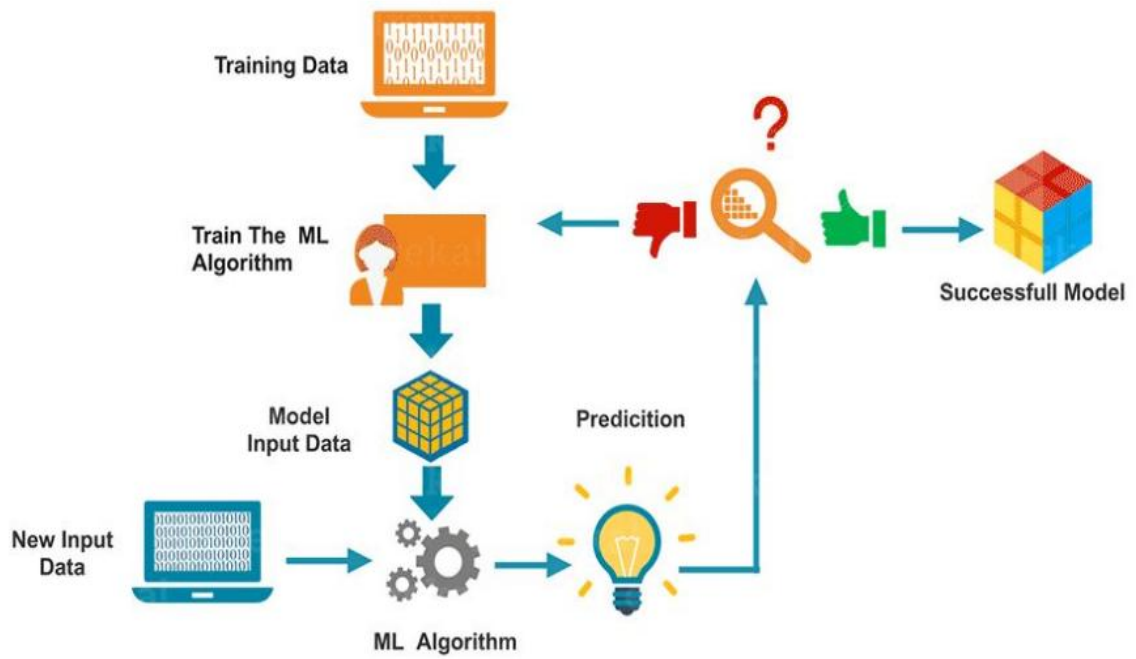
4.7 Deployment

Deployment, the final stage of the machine learning life cycle, involves implementing the model in a practical system.

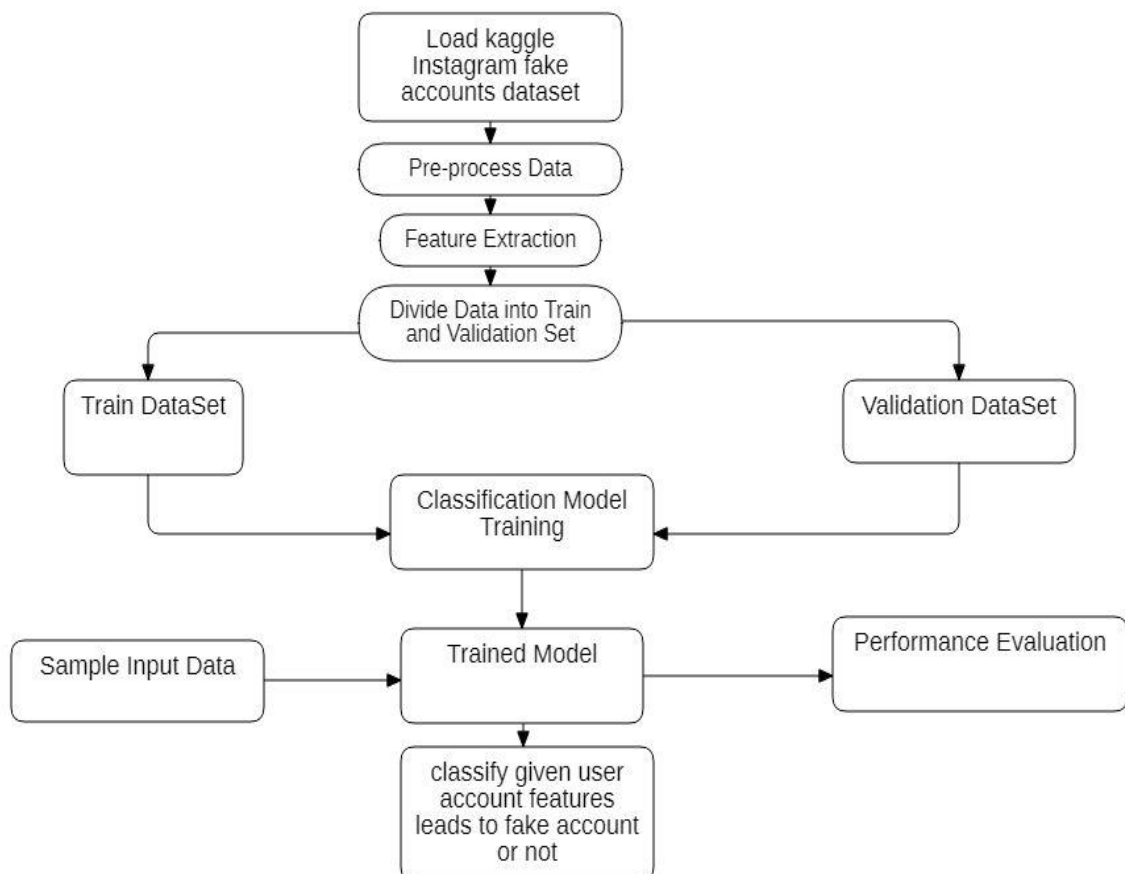
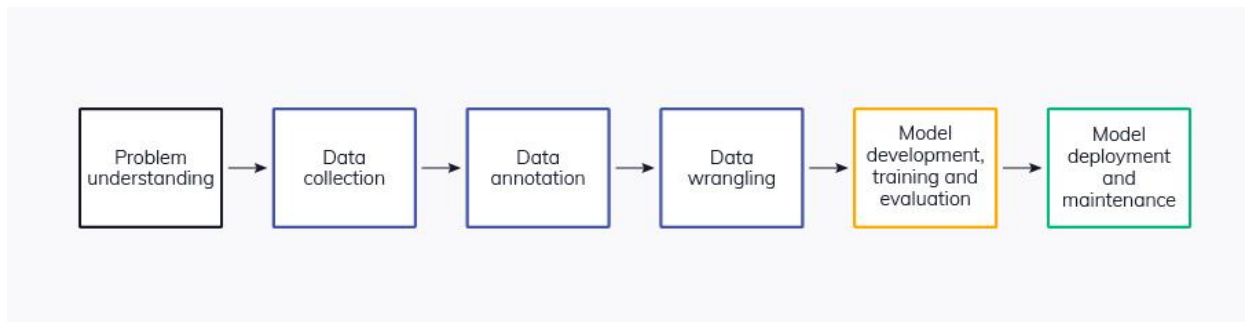
We implement the model in the actual system if it delivers an accurate output that meets our requirements quickly and as planned. However, we will first determine whether the project is using the given data to improve performance before deploying it. The project's final report is made during the deployment phase.

5.SYSTEM DESIGN

5.1 SYSTEM ARCHITECTURE



5.2 DATA FLOW DIAGRAMS



Activate Win

5.3 UML DIAGRAMS

Unified Modeling Language (UML) is a modeling language. The main purpose of UML is to visualize the way a system has been designed. It is a visual language to sketch the behavior and structure of the system. This was adopted by Object Management Group (OMG) as a standard in 1997.

Need Of UML:

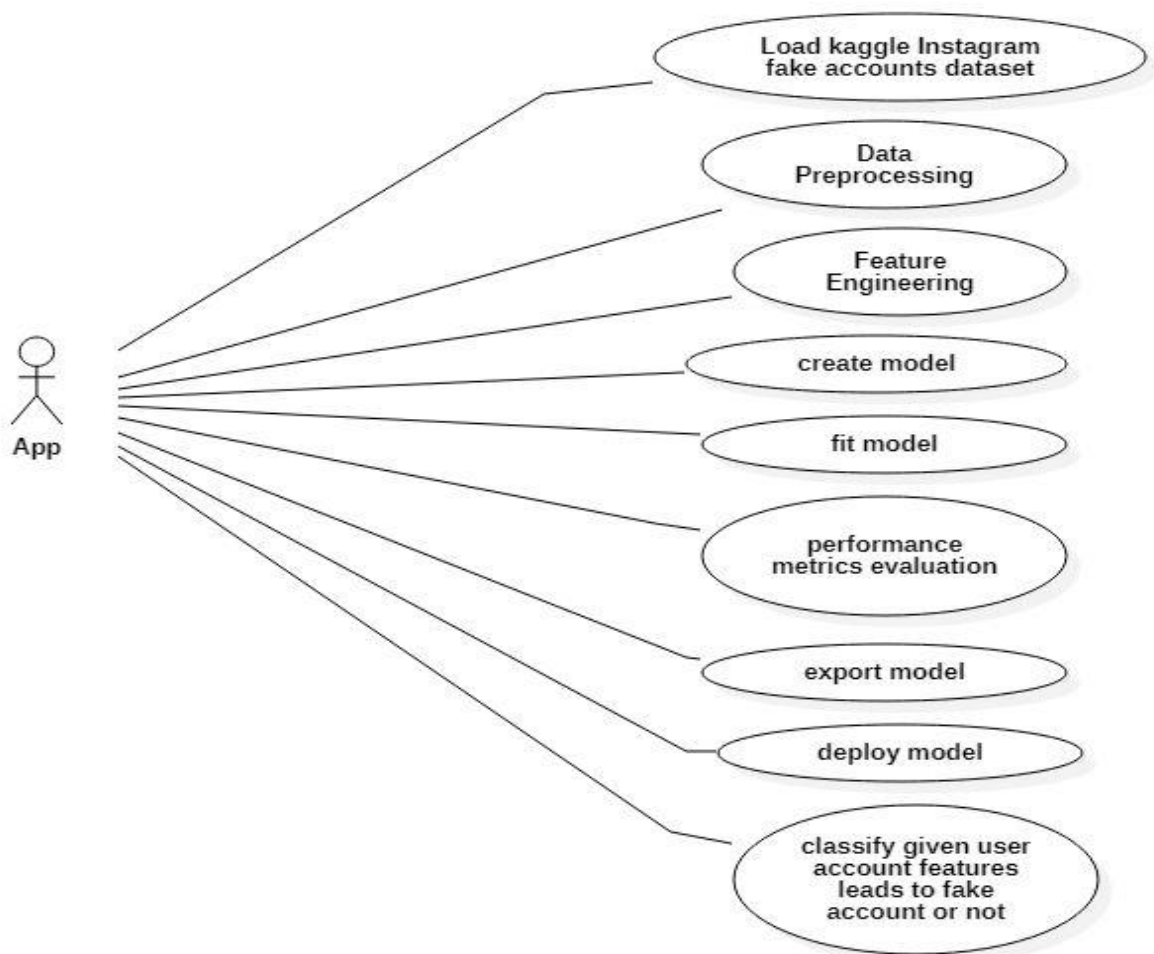
- Complex applications necessitate collaboration and planning from multiple teams, as well as clear and direct communication between them.
- When it comes to code, businessmen are dumb. As a result, UML becomes crucial for non-programmers to comprehend the main requirements, features, and processes of the system.

UML Characteristics:

- It's a model language that's been broadened.
- It varies from other programming languages in a number of ways.
- It is tied to object-oriented analysis and design.

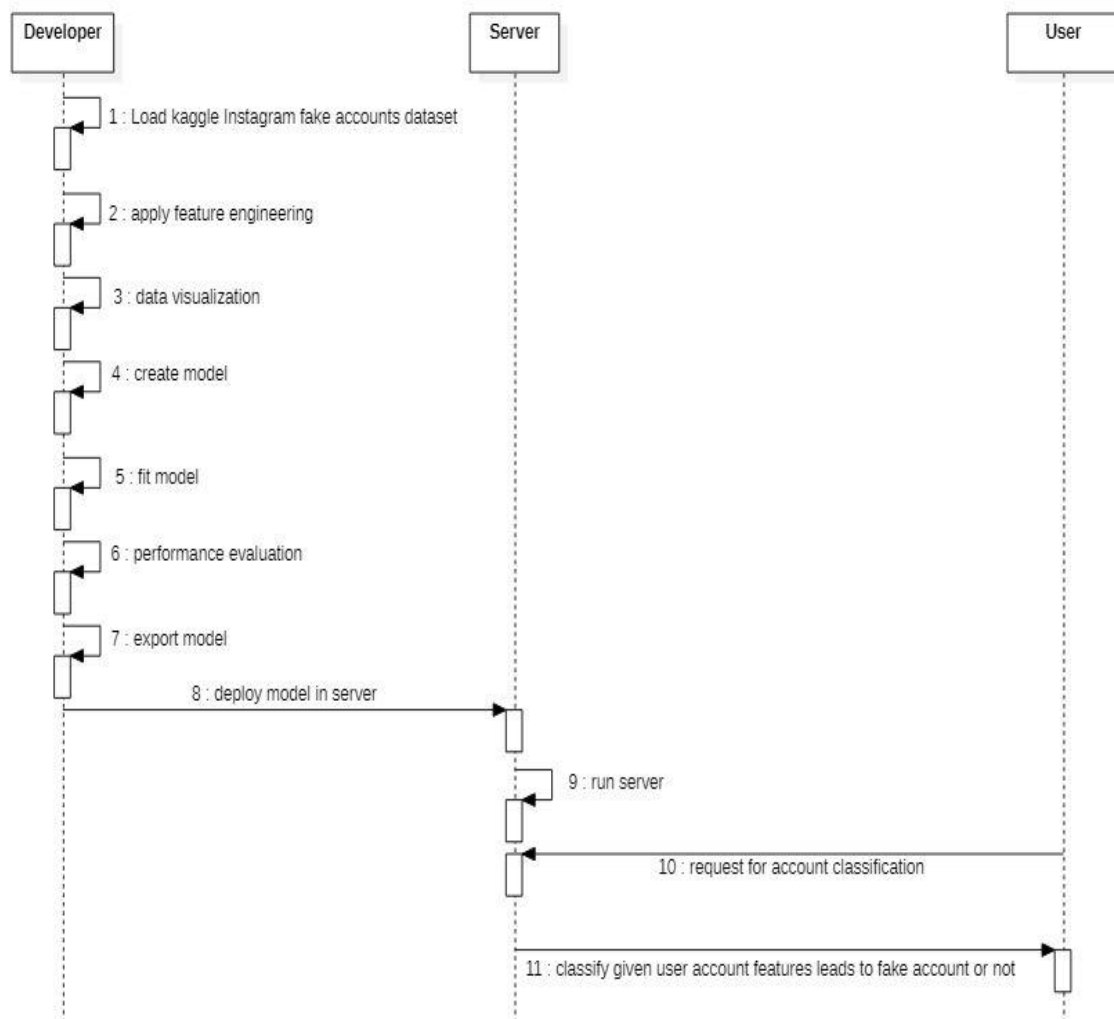
5.3.1 Use case Diagram

- The purpose of use case diagram is to capture the dynamic aspect of a system. This is used to gather a system's needs, taking into consideration both internal and external factors.
- A use case diagram's principal objective is to identify which system operations are carried out for which actor. One can illustrate the parts played by the system's actors.
- Creating objects-oriented software and the software development process both benefit greatly from the use of the UML. The design of software projects is expressed mostly through graphical notations in the UML.



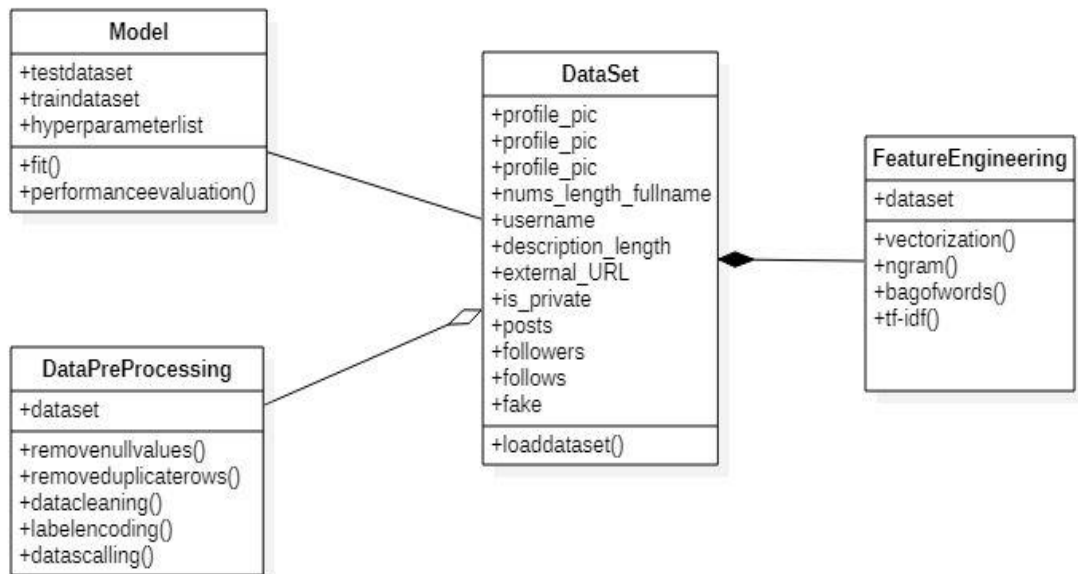
5.3.2 Sequence Diagram

- A sequence diagram shows how things interact in a sequential manner, or the order in which these interactions occur.
- These diagrams sometimes known as event diagrams or event scenarios. This helps in understanding how the objects and component interacts to execute the process.
- This has two dimensions which represents time (Vertical) and different objects (Horizontal).



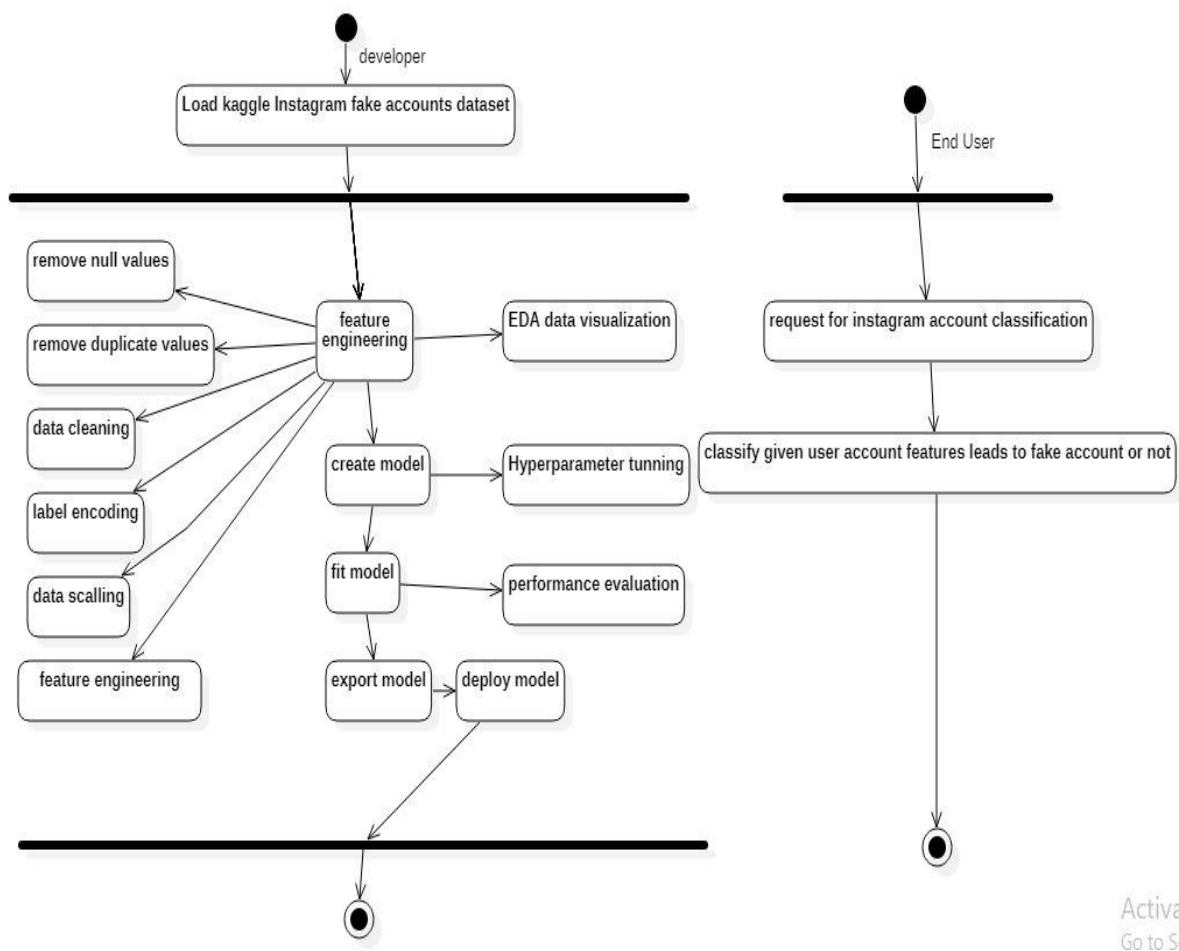
5.3.3 Class Diagram

- The class diagram describes the structure of a system by showing the system's classes, their attributes, operations, and the relationships among the classes.
- It explains which class contains information and also describes responsibilities of the system. This is also known as structural diagram.



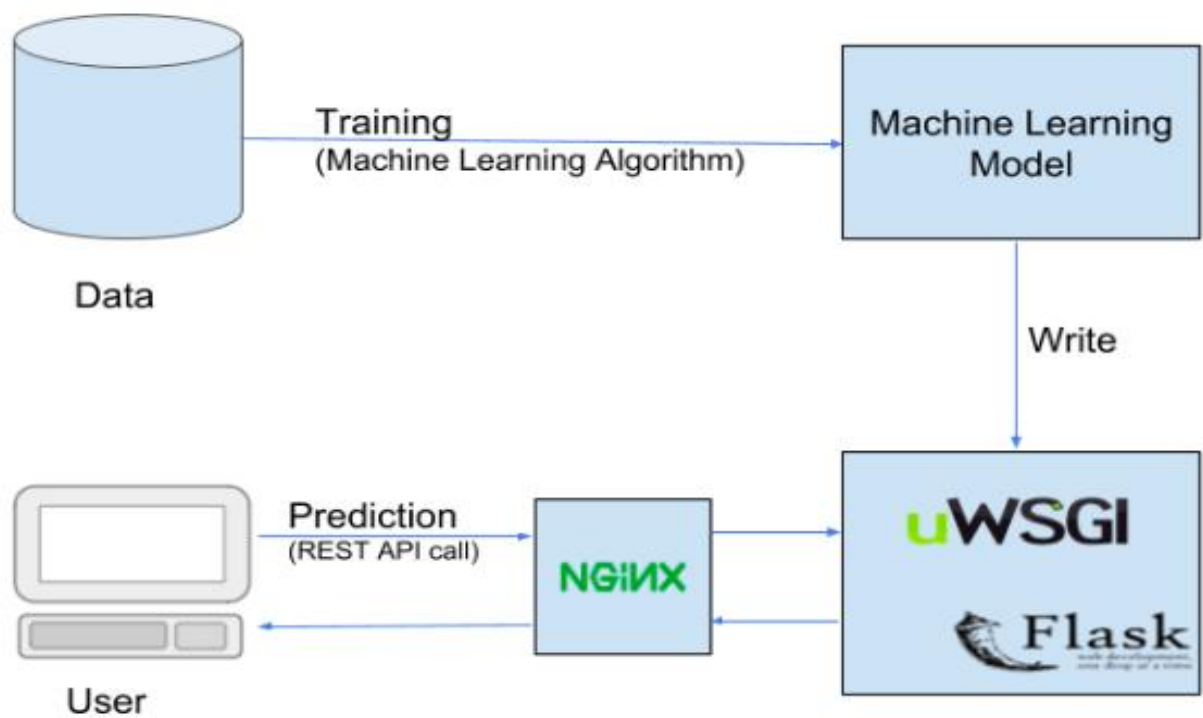
5.3.4 Activity Diagram

- It is behavioral diagram which reveals the behavior of a system. It sketches the control flow from an activity's start point to its completion point, illuminating the various decision options that might be taken along the way.
- This doesn't show any message flow from one activity to another, it is sometimes treated as the flowchart. Despite they look like a flowchart, they are not.
- The business and operational step-by-step processes of system components can be described using activity diagrams in the Unified Modelling Language.



Activat
Go to Set

5.3.5 Deployment Diagram



6.SYSTEM ENVIRONMENT

6.1 Python

Python is a dynamically semantic, interpreter-based, object-oriented, high-level programming language. It is particularly desirable for Rapid Application Development as well as for usage as a scripting or glue language to tie existing components together due to its high-level built-in data structures, dynamic typing, and dynamic binding. Python's straightforward syntax emphasises readability and makes it simple to learn, which lowers the cost of programme maintenance.

For all popular platforms, the Python interpreter and the comprehensive standard library are freely distributable and available in source or binary form. Python programmes are simple to debug because a segmentation failure is never caused by a bug or incorrect input. Instead, the interpreter raises an exception when it finds a mistake. The interpreter prints a stack trace if the programme doesn't catch the exception.

With a source level debugger, you may step through the code one line at a time, create breakpoints, check local and global variables, evaluate arbitrary expressions, and more. Python's ability to be introspective is demonstrated by the fact that the debugger is developed in Python. However, adding a few print statements to the source code is frequently the simplest way to debug a programme. The quick edit-test-debug cycle makes this straightforward method quite effective.

6.2 Anaconda

Data scientists, IT specialists, and future company executives can all use Anaconda as their data science platform. It is a compilation of Python, R, etc. It develops become one of the greatest platforms for any project with more than 300 data science packages. With the help of various machine learning and AI algorithms, Anaconda's tools make it simple to gather data from several sources. By pushing a single button, any project may be deployed in an environment that is simple to administer.

The Anaconda Platform's major goal is to make it simple for those who are passionate about these disciplines. It comes with a large number of pre-installed libraries and packages and only requires a single installation procedure. This platform is simple to use and suitable for beginners.

7.IMPLEMENTATION

7.1 IMPLEMENTATION

Data pre-processing: The dataset features are presented in two types.

Categorical Features: Categorical Features has various categories of data for e. g languages, different tweets, profile colors.

Numerical Features:It contains data of the numerical type.

Feature Extraction: In feature reduction phase has extracted the different features and reduce the dimensionality of features. Feature reduction has used two different techniques to reduce the dimensionality of features.

1. Principal Component Analysis (PCA)
- 2.Spearman's Rank-Order Correlation

1.Principal Component Analysis (PCA): PCA is a dimension reduction technique this is used to reduce feature vector dimensions. It finds the top number of features that best describe the data and covers as much variance of it, unnecessary features by assigning a lower weight so they did not impact on the data mining process.

2. Spearman's Rank-Order Correlation: Spearman's Rank-Order Correlation is a type of feature selection filtering method. It measures the strength and direction of the monotonic relationship between two variables P and Q.

```
x = extract_features(x)
print(x.columns)
```

Train and Test data:

Train and Test is a method to measure the accuracy of your model. We will split the data set into two sets: a training set and a testing set.80% for training and 20% for testing.You train the model using the training set and test the model using the testing set.

splitting train and test data

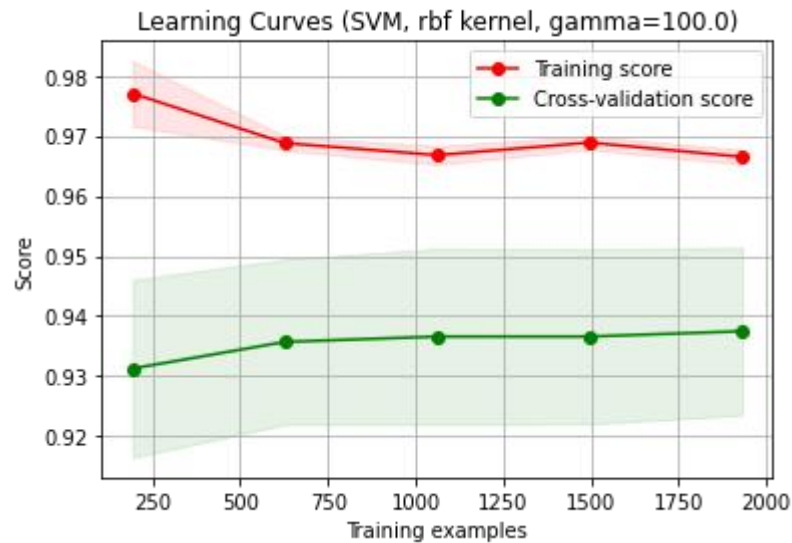
```
X_train,X_test,y_train,y_test = train_test_split(x, y, test_size=0.20, random_state=44)
```



```
y_test,y_pred = train(X_train,y_train,X_test,5, 7)
```

Mean Training Score: 0.2495509215763637

Mean Test Score: 0.9374662630509122



Random Forest Algorithm:

Random forest algorithm is a supervised classification algorithm. A forest of many trees is created by this algorithm. In general, the more trees in the forest the more robust the forest looks like. In the same way in the random forest classifier, the higher the number of trees in the forest gives the high accuracy results.

Gradient Boosting Algorithm:

Gradient boosting algorithm is like random forest algorithm which uses decision trees as its main component. We also changed the way we find the fake accounts i.e., we introduced new methods to find the account. Spam comments, engagement rate, and false activity are the techniques employed. These inputs are used to form decision trees that are used in the gradient boosting algorithm. This algorithm gives us an output even if some inputs are missing. This is the major reason for choosing this algorithm. Due to the use of this algorithm we were able to get highly accurate results.

Result:

The dataset is divided into 80% for training and 20% for testing. Cross validation should be done to get good accuracy. The account will be detected whether it is fake or real by comparing each feature using algorithm.

7.2 SOURCE CODE

```
!pip install gender_guesser
```

```
import sys
```

```
import csv
```

```
import datetime
```

```
import numpy as np
```

```
import pandas as pd
```

```
import matplotlib.pyplot as plt
```

```
from datetime import datetime
```

```
import gender_guesser.detector as gender
```

```
from sklearn.impute import SimpleImputer
```

```
from sklearn.model_selection import cross_val_score
```

```
from sklearn.model_selection import cross_validate
```

```
from sklearn import metrics
```

```
from sklearn import preprocessing
```

```
from sklearn.linear_model import LinearRegression
```

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

```
from sklearn.model_selection import StratifiedKFold, train_test_split
```

```
from sklearn.model_selection import GridSearchCV
```

```
from sklearn.metrics import accuracy_score, mean_absolute_error
```

```
from sklearn.model_selection import learning_curve
```

```
from sklearn.metrics import classification_report
```

```
from sklearn.metrics import confusion_matrix
```

```
from sklearn.ensemble import RandomForestClassifier
```

```

def read_datasets():
    """ Reads users profile from csv files """
    real_users = pd.read_csv("data/users.csv")
    fake_users = pd.read_csv("data/fusers.csv")

    x = pd.concat([real_users, fake_users])
    y = len(fake_users)*[0] + len(real_users)*[1]

    return x,y


def predict_sex(name):
    d = gender.Detector(case_sensitive=False)
    first_name= str(name).split(' ')[0]
    sex = d.get_gender(u"{}".format(first_name))

    sex_code_dict = {'female': -2, 'mostly_female': -1, 'unknown': 0, 'andy': 0, 'mostly_male': 1,
'male': 2}
    code = sex_code_dict[sex]

    return code


def extract_features(x):
    lang_list = list(enumerate(np.unique(x['lang'])))

    lang_dict = { name : i for i, name in lang_list }

    x.loc[:, 'lang_code'] = x['lang'].map( lambda x: lang_dict[x]).astype(int)
    x.loc[:, 'sex_code'] = predict_sex(x['name'])

    feature_columns_to_use =
['statuses_count', 'followers_count', 'friends_count', 'favourites_count', 'listed_count', 'sex_code', 'l
ang_code']

    x = x.loc[:, feature_columns_to_use]

```

```

    return x

def plot_learning_curve(estimator, title, X, y, ylim=None, cv=None, n_jobs=1,
train_sizes=np.linspace(.1, 1.0, 5)):

    plt.figure()
    plt.title(title)
    if ylim is not None:
        plt.ylim(*ylim)
    plt.xlabel("Training examples")
    plt.ylabel("Score")

    train_sizes, train_scores, test_scores = learning_curve(estimator, X, y, cv=cv,
n_jobs=n_jobs, train_sizes=train_sizes)

    train_scores_mean = np.mean(train_scores, axis=1)
    train_scores_std = np.std(train_scores, axis=1)
    test_scores_mean = np.mean(test_scores, axis=1)
    test_scores_std = np.std(test_scores, axis=1)
    plt.grid()

    plt.fill_between(train_sizes, train_scores_mean - train_scores_std,
                     train_scores_mean + train_scores_std, alpha=0.1,
                     color="r")
    plt.fill_between(train_sizes, test_scores_mean - test_scores_std,
                     test_scores_mean + test_scores_std, alpha=0.1, color="g")
    plt.plot(train_sizes, train_scores_mean, 'o-', color="r",
             label="Training score")
    plt.plot(train_sizes, test_scores_mean, 'o-', color="g",
             label="Cross-validation score")

    plt.legend(loc="best")

    return plt

def train(X_train,y_train,X_test, nSplits, CV):

```

```

""" Trains and predicts dataset with a SVM classifier """

# Scaling features
X_train = preprocessing.scale(X_train)
X_test = preprocessing.scale(X_test)

Cs = 10.0 ** np.arange(-2,3,.5)
gammas = 10.0 ** np.arange(-2,3,.5)
param = [{'gamma': gammas, 'C': Cs}]

cvk = StratifiedKFold(n_splits= nSplits)

classifier = SVC()

clf = GridSearchCV(classifier, param_grid=param, cv=cvk)
clf.fit(X_train,y_train)

print("The best classifier is: ", clf.best_estimator_)
clf.best_estimator_.fit(X_train, y_train)

print()

# Estimate score
scores = cross_validate(clf.best_estimator_, X_train,y_train, cv=CV)

for k in [*scores]:
    print(k + ": ", scores[k])

print()

print("Mean Training Score: {}".format(scores['fit_time'].mean()))
print("Mean Test Score: {}".format(scores['test_score'].mean()))

```

```

        title = 'Learning Curves (SVM, rbf kernel,
gamma={})'.format(clf.best_estimator_.gamma)

    plot_learning_curve(clf.best_estimator_, title, X_train, y_train, cv=CV)

    plt.show()

    # Predict class
    y_pred = clf.best_estimator_.predict(X_test)

    return y_test,y_pred

x,y = read_datasets()
print("dataset read complete")

x = extract_features(x)
print(x.columns)

x.head(10)

print(y[:-1])

# splitting train and test data
X_train,X_test,y_train,y_test = train_test_split(x, y, test_size=0.20, random_state=44)

y_test,y_pred = train(X_train,y_train,X_test,5, 7)

confusionMatrix = confusion_matrix(y_test, y_pred)
print('Confusion matrix, without normalization')
print(confusionMatrix)
plot_confusion_matrix(confusionMatrix)
print(classification_report(y_test, y_pred, target_names=['Fake','Genuine']))

```

```

# RANDOM FOREST
rf_classifier = RandomForestClassifier(n_estimators=100, max_depth=2, random_state=0)
rf_classifier.fit(X_train, y_train)
train_predictions = rf_classifier.predict(X_train)
prediction = rf_classifier.predict(X_test)
print(X_test[:5])
print(prediction)

err_training = mean_absolute_error(train_predictions, y_train)
err_test = mean_absolute_error(prediction, y_test)

print("Train Accuracy is : {}".format(100 - (100*err_training)))
print("Test Accuracy is : {}".format(100 - (100*err_test)))

prediction = rf_classifier.predict([[9950,658,701,18,11,0,5]])
print(X_test.head())
print(prediction)

def plot_roc_curve(y_test, y_pred):
    false_positive_rate, true_positive_rate, thresholds = roc_curve(y_test, y_pred)

    print ("False Positive rate: ",false_positive_rate)
    print ("True Positive rate: ",true_positive_rate)

    roc_auc = auc(false_positive_rate, true_positive_rate)

    plt.title('Receiver Operating Characteristic')
    plt.plot(false_positive_rate, true_positive_rate, 'b',
             label='AUC = %0.2f%% roc_auc)
    plt.legend(loc='lower right')
    plt.plot([0,1],[0,1], 'r--')

```

```

plt.xlim([-0.1,1.2])
plt.ylim([-0.1,1.2])
plt.ylabel('True Positive Rate')
plt.xlabel('False Positive Rate')
plt.show()

from sklearn.metrics import roc_curve, auc
plot_roc_curve(y_test, y_pred)

import pickle
pickle.dump(rf_classifier, open('rf_classifier.pkl', 'wb'))

#GRADIENT BOOSTING
from sklearn.ensemble import GradientBoostingClassifier

clf = GradientBoostingClassifier(n_estimators=100, learning_rate=1.0,max_depth=1,
random_state=0)
clf.fit(X_train, y_train)
train_predictions = clf.predict(X_train)
prediction = clf.predict(X_test)

err_training = mean_absolute_error(train_predictions, y_train)
err_test = mean_absolute_error(prediction, y_test)

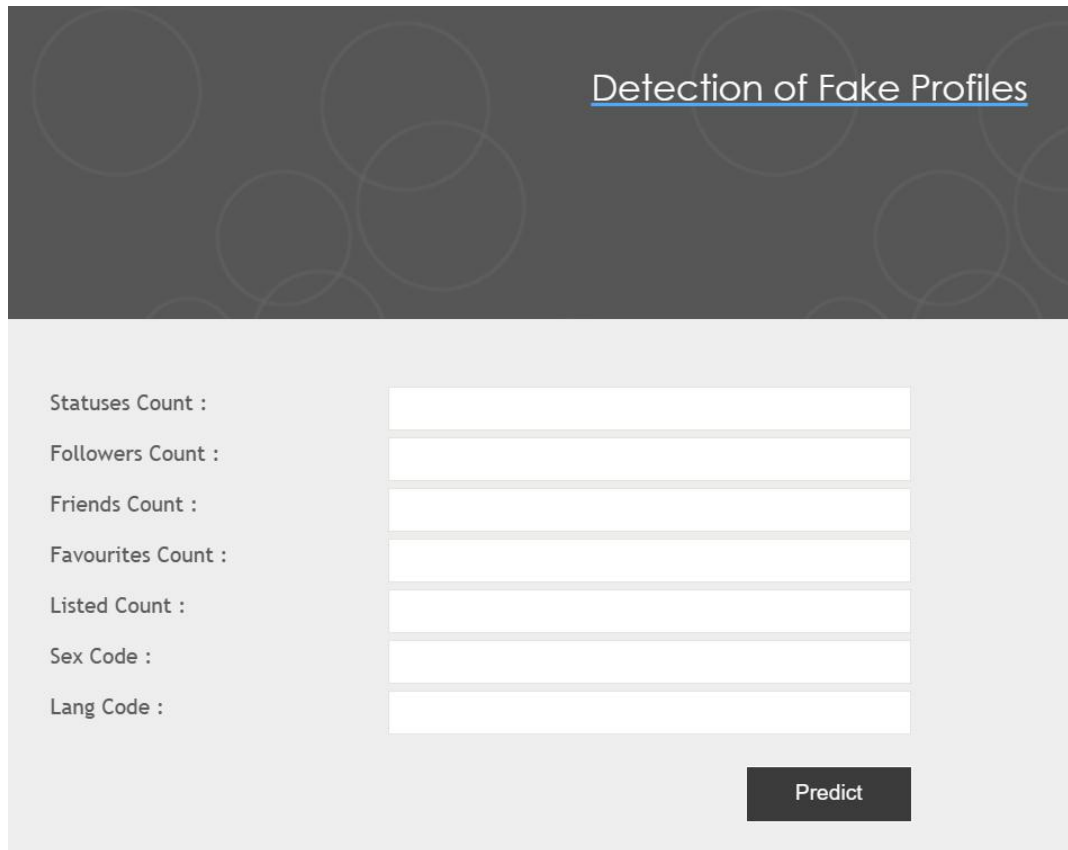
print("Train Accuracy is : {}".format(100 - (100*err_training)))
print("Test Accuracy is : {}".format(100 - (100*err_test)))

prediction = clf.predict([[9950,658,701,18,11,0,5]])
print(X_test.head())
print(prediction)

from sklearn.metrics import roc_curve, auc
plot_roc_curve(y_test, y_pred)

```


8.SCREENSHOTS AND RESULTS



The screenshot displays a web interface for detecting fake profiles. The title 'Detection of Fake Profiles' is underlined in the top right corner. Below the title, there are seven input fields arranged vertically, each preceded by a label: 'Statuses Count :', 'Followers Count :', 'Friends Count :', 'Favourites Count :', 'Listed Count :', 'Sex Code :', and 'Lang Code :'. To the right of these labels is a single vertical input field that spans the height of all seven labels. At the bottom right of the form area is a dark button labeled 'Predict'.

Fig.8.1 Give the input to detect the profile is real or fake

Detection of Fake Profiles

Statuses Count :	<input type="text" value="20"/>
Followers Count :	<input type="text" value="3500"/>
Friends Count :	<input type="text" value="2000"/>
Favourites Count :	<input type="text" value="250"/>
Listed Count :	<input type="text" value="28"/>
Sex Code :	<input type="text" value="1"/>
Lang Code :	<input type="text" value="1"/>

Predict

Fig.8.2 Given inputs to detect profile is real or fake

Detection of Fake Profiles

Real User

Statuses Count :	<input type="text"/>
Followers Count :	<input type="text"/>
Friends Count :	<input type="text"/>
Favourites Count :	<input type="text"/>
Listed Count :	<input type="text"/>
Sex Code :	<input type="text"/>
Lang Code :	<input type="text"/>

Predict

Fig.8.3 Real user

Detection of Fake Profiles

Statuses Count :	<input type="text" value="20"/>
Followers Count :	<input type="text" value="3500"/>
Friends Count :	<input type="text" value="1000"/>
Favourites Count :	<input type="text" value="0"/>
Listed Count :	<input type="text" value="80"/>
Sex Code :	<input type="text" value="1"/>
Lang Code :	<input type="text" value="1"/>

Fig.8.4 Given inputs to detect the profile is real or fake

Detection of Fake Profiles

Fake User

Statuses Count :	<input type="text"/>
Followers Count :	<input type="text"/>
Friends Count :	<input type="text"/>
Favourites Count :	<input type="text"/>
Listed Count :	<input type="text"/>
Sex Code :	<input type="text"/>
Lang Code :	<input type="text"/>

Fig.8.5 Fake user

8.CONCLUSION

A new classification algorithm was proposed to improve detecting fake accounts on social networks, where the Random Forest trained model decision values were used to train a neural network model. To reach our goal we used dataset run it into the preprocessing phase where different feature reduction techniques were used to reduce the feature vector. In the classification phase, random forest learning algorithms were used. The results of the analyses showed that random forest has archived better accuracy results with all feature sets comparing with other classifiers, with classification accuracy around 98%. It was noticed that the Neural Network algorithm has the lowest classification accuracy compared with random forest.

In case some of the inputs are missing, it gets so tedious for random forest algorithm to give the results. So, gradient boosting algorithm is used to detect the account is fake or real even some inputs are missing. The Accuracy of detecting fake accounts is found to be higher using Gradient Boosting Algorithm followed by Random Forest Algorithm for a given dataset. So, here we are using gradient boosting algorithm to detect accurate output.

REFERENCES

- [1]Ramalingam, D., Chinnaiah, V. (2018). Fake profile detection techniques in large scale online social networks.
- [2]Ferrara, E., Varol, O., Davis, C., Menczer, F., Flammini,A. (2016). The rise of social bots. *Communications of the ACM*, 59(7): 96-104.
- [3]Cresci, S., Di Pietro, R., Petrocchi, M., Spognardi, A.,Tesconi, M. (2015). Fame for sale: Efficient detection of fake Twitter followers. *Decision Support Systems*.
- [4]Zhang, Y., Lu, J. (2016). Discover millions of fake followers in Weibo. *Social Network Analysis and Mining*.
- [5]Thomas, K., Grier, C., Song, D., Paxson, V. (2011).Suspended accounts in retrospect: An analysis of twitter spam. In *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*, pp.243-258.
- [6]Benevenuto, F., Magno, G., Rodrigues, T., Almeida, V.(2010). Detecting spammers on twitter. In *Collaboration,Electronic Messaging, Anti-abuse and Spam Conference (CEAS)*, p. 12.

Fake Instagram Profile Detection and Classification Using Machine Learning

¹Mrs.B. SHIVANI, ²T. TEJA SRI, ³N. NIKHITHA, ⁴M.A. AHAD SIDDIQUI

¹Assistant Professor, Dept.of IT, TKR College of Engineering & Technology, Meerpet, Hyderabad,
bhutam.shivani@gmail.com

²BTech student, Dept.of IT, TKR College of Engineering & Technology, Meerpet, Hyderabad,
tejasrithurupu@gmail.com

³BTech student, Dept.of IT, TKR College of Engineering & Technology, Meerpet, Hyderabad,
nnikhitha900@gmail.com

⁴BTech student, Dept.of IT, TKR College of Engineering & Technology, Meerpet, Hyderabad,
ahadsiddiqui042@gmail.com

Abstract: *With the increase in Internet usage, Instagram is now considered a very important platform for advertising marketing and social interaction. It is used by millions of users but, some users tend to misuse the platform by creating false identities. Moreover, the popularity of social media users is determined by followers and hence users resort to different wrong means to promote increased profile followers. Fake engagement is one of the significant problems in Online Social Networks (OSNs) which is used to increase the popularity of an account in an inorganic manner. The detection of fake engagement is crucial because it leads to loss of money for businesses, wrong audience targeting in advertising, wrong product predictions systems, and unhealthy social network environment. This study is related with the detection of fake and automated accounts which leads to fake engagement on Instagram. Prior to this work, there were no publicly available dataset for fake and automated accounts. For the detection of these accounts, machine learning algorithms like Naive Bayes, Logistic Regression, Support Vector Machines and Neural Networks are applied. Additionally, for the detection of automated accounts, cost sensitive genetic algorithm is proposed to handle the unnatural bias in the dataset. To deal with the unevenness problem in the fake dataset, Smote-nc algorithm is implemented. For the automated and fake account detection datasets, 86% and 96% classification accuracies are obtained, respectively.*

Keywords: *Online Social Networks, Machine learning, Instagram, fake profile identification.*

I. INTRODUCTION

With the arrival of the Internet and social media, at the same time as masses of humans have benefitted from the full-size re assets of records available, there was a full-size boom with inside the upward push of cyber-crimes, mainly targeted closer to women. According to a 2019 file with inside the Economics Times, India has witnessed a 457% upward push in cybercrime with inside the 5 years span among 2011 and 2016. Most speculate that that is because of effect of social media inclusive of Facebook, Instagram and Twitter on our day by day lives. While those simply assist in growing a legitimate social network, advent of consumer debts in those websites normally desires simply an email-id. A actual lifestyles man or woman can create more than one fake IDs and for this reason impostors can effortlessly be made. Unlike the actual international state of affairs in which more than one policies and guidelines are imposed to become aware of oneself in a completely unique manner (as an instance at the same time as issuing one's passport or driver's license), with inside the digital international of social media, admission does now no longer require this kind of checks. In this paper, we study the one-of-a-kind debts of Instagram, specifically and try and verify an account as fake or actual the use of Machine Learning strategies

specifically Logistic Regression and Random Forest Algorithm.

Instagram is an online photo and video sharing social networking platform that has been available on both Android and iOS since 2012. As of May 2019, there are over a billion users registered on Instagram. In the recent years, Instagram has been found to be using third party apps, called bots. While these can definitely impersonate a user and tarnish their reputation leading to 'identity theft', there has also been greater instances of malicious ways of promoting the brand image of a company known as "influencer marketing". These days a number of businesses are using social media to heed to their customers' needs which has led to yet another malpractice called Angler phishing. All these malpractices have made it vital to implement strong fraud detection techniques and hence we propose our solution.

II. LITERATURE SURVEY

Today, social networks are developing at an incredible speed. These services are vital to the human masses in society, especially to advertising campaigns, celebrities, and politicians trying to market themselves using fans and fanatics on social media. Therefore, fake money owed

created in the name of human beings and corporations can be dangerous, reputational damage to human and organization, and in the long run hastened the downgrading of their true likes and fans. Moreover, all kinds of fake profiles have a detrimental effect on the social media benefits of advertising, marketing and advertising organizations. These fake profiles can be a form of cyberbullying; Real customers also have great concerns about their privacy in the online environment with those fake profiles.

Therefore, in recent years, many researchers have investigated the problem of detecting malicious sports and spammers on social media using system scanning strategies. However, there is a limited pattern of research articles on detecting phantom debt or fake fanatics. In this section, we highlight every spammer and fake debt answer that has been added these days.

Ferrara et al. It provided a way to hit bot users on Twitter based entirely on the somewhat common powers that set them apart from valid users. In their proposed approach, they used a system that defines the method and behavior patterns between valid debt and bot debt to classify the money owed on the bot or valid bill.

Cresci et al. have created and used a baseline dataset of verified human and fake fans on Twitter. In their work, they exploited the baseline dataset to educate a fixed of gadget gaining knowledge of classifiers constructed primarily based totally on reviewed policies and capabilities set the use of the media. Their proposed approach is green in detecting fake money owed; the consequences accomplished via way of means of their approach display it can classify greater than 95% of the money owed successfully from the authentic schooling set.

In a barely extraordinary approach, Zhang and Lu. Introduced a unique approach for the detection of fake money owed in Weibo. Their proposed answer has extraordinary aspects. At first, that they'd this premises why such money owed exist with inside the first place. In the second, they investigated the overlap among fans listing of the clients of fake fans, and that they located an excessive overlap among their follower lists. Their research located 395 nearduplicates, which caused 11.90 million fake monies owed that despatched 1,000,000 hyperlinks with inside the network.

Thomas et al. made a group of 1.8 million tweets despatched via way of means of 32.9 Twitter money owed. In their research,

they located Twitter suspended approximately 1.1 million of these money owed. They have decided on randomly a hundred of these money owed to research their tweets and confirm they had been spamming money owed.

They made a similarly evaluation on that a hundred decided on money owed, and that they locate 93 of the chosen money owed had been suspended for posting junk mail and the unsolicited commercial of numerous products. Three different monies owed had been suspended for re-tweeting content material of extraordinary information money owed, and the alternative four remained money owed had been suspended for reproduction and competitive advertising posts.

Gao et al. have used a fixed of capabilities for efficiently reconstructing junk mail tweets into campaigns in place of studying them separately. The end result suggests their proposed answer received. However, the disadvantage in their approach is its low detection accuracy.

Benevenuto et al. proposed a option to stumble on spammers from non spammers. In their approach, they used an SVM classifier, that is a supervised gadget gaining knowledge of algorithm. They have used 23 conduct and 39 content

material capabilities to distinguish spammers from nonspammers, and that they accomplished experiments via way of means of 5-fold cross-validations. The experiments display they had been nearly a success in figuring out spammers from non-spammers.

Bala Anand et al. advanced a new gadget to stumble on fake customers at the Twitter platform the use of a graph-primarily based totally semi-supervised gaining knowledge of algorithm (EGSLA) and examine and amassing behavioral and person-generated content material (UGC) information. The version first gathered customers information, analyzed them to extract beneficial capabilities, and then accomplished type on those capabilities and made decisions. The experimental consequences display that the EGSLA algorithm accomplished excessive overall performance and turned into greater useful than different algorithms inclusive of choice tree, KNN, SVM, and game theory-primarily based totally techniques in phrases of type accuracy.

Sahoo et al. Provided a hybrid model for finding malicious social media profiles by targeting Twitter. The proposed hybrid version consists of modules; First, they analyzed and extracted the efficiencies with the internet form of Petrie, and then

used the capabilities of those efficiencies as the classifiers come to categorize the profiles as malicious and legitimate instructions. Experimental results show that the proposed method effectively outperformed Twitter's money owed and obtained a high detection rate in type-accuracy words. Therefore, according to the literature, many researchers have used the devices to gain knowledge about techniques to overcome security problems in social networks. The research surveyed generally focused on spotting spam on microblog social networks. They have researched many solutions to solve the problem of spam and fake money owed on Twitter and the amazing microblogging social network. However, as of now, there is no complete solution to phantom debt on the Instagram platform, and this is one of the motivations at the end of this study. Therefore, in this paper, we propose an inexperienced approach to detect fake debts on the Instagram platform, which can correctly classify the large money owed by Instagram people.

III. PROPOSED WORK

In this paper, the automatic detection of fake profiles has been proposed to identify fake Instagram profiles so that the social life of Instagram users is secure. The prediction of fake Instagram profiles is

facilitated using supervised learning machine algorithms. Upon classification, fake profile IDs are stored in a data dictionary to further help the concerned authorities to take necessary actions against fraudulent social media profiles. Experimentation has been done to compare the classification algorithms used to train the dataset. The factors used by the existing systems to detect the fake accounts are very less. The prediction becomes accurate when the number of parameters used are more efficient. In previously used algorithms, if some of the inputs are not appropriate, the algorithm could not produce accurate results. Hence, in this research, we made use of gradient boosting algorithm. It uses decision trees as a prime factor. We made use of several parameters. These parameters are further considered as inputs which are used to form the decision trees in order to apply gradient boosting algorithm.

SYSTEM ARCHITECTURE

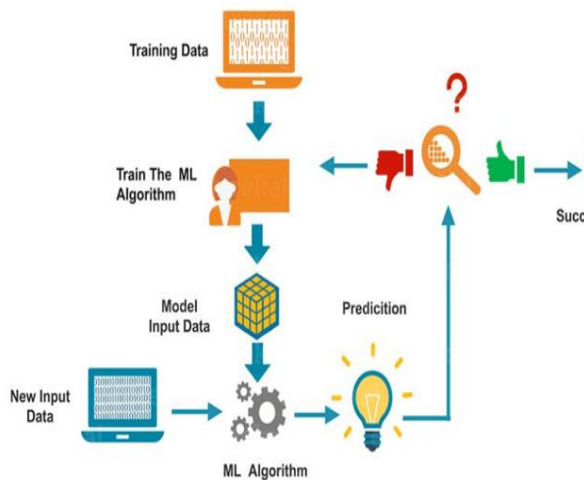


Fig.1 System architecture

GATHERING DATA

Data Gathering is the first step of the machine learning life cycle. The goal of this step is to identify and obtain all data-related problems.

In this step, we need to identify the different data sources, as data can be collected from Kaggle such as csv files. It is one of the most important steps of the life cycle. The quantity and quality of the collected data will determine the efficiency of the output. The more will be the data, the more accurate will be the prediction.

This step includes the below tasks:

- Identify various data sources
- Collect data
- Integrate the data obtained from different sources

By performing the above task, we get a coherent set of data, also called as a dataset. It will be used in further steps.

DATA PREPARATION

After collecting the data, we need to prepare it for further steps. Data preparation is a step where we put our data into a suitable place and prepare it to use in our machine learning training.

In this step, first, we put all data together, and then randomize the ordering of data.

This step can be further divided into two processes:

- Data exploration:

It is used to understand the nature of data that we have to work with. We need to understand the characteristics, format, and quality of data.

A better understanding of data leads to an effective outcome. In this, we find Correlations, general trends, and outliers.

- Data pre-processing:

Now the next step is pre-processing of data for its analysis.

DATA WRANGLING

Data wrangling is the process of cleaning and converting raw data into a useable format. It is the process of cleaning the

data, selecting the variable to use, and transforming the data in a proper format to make it more suitable for analysis in the next step. It is one of the most important steps of the complete process. Cleaning of data is required to address the quality issues.

It is not necessary that data we have collected is always of our use as some of the data may not be useful. In real-world applications, collected data may have various issues, including:

- Missing Values
- Duplicate data
- Invalid data
- Noise

So, we use various filtering techniques to clean the data.

It is mandatory to detect and remove the above issues because it can negatively affect the quality of the outcome.

DATA ANALYSIS

Now the cleaned and prepared data is passed on to the analysis step. This step involves:

- Selection of analytical techniques
- Building models
- Review the result

The aim of this step is to build a machine learning model to analyze the data using various analytical techniques and review the outcome. It starts with the determination of the type of the problems, where we select the machine learning techniques such as Classification. then build the model using prepared data, and evaluate the model.

Hence, in this step, we take the data and use machine learning algorithms to build the model.

TRAIN MODEL

Now the next step is to train the model, in this step we train our model to improve its performance for better outcome of the problem.

We use datasets to train the model using various machine learning algorithms. Training a model is required so that it can understand the various patterns, rules, and, features.

TEST MODEL

Once our machine learning model has been trained on a given dataset, then we test the model. In this step, we check for the accuracy of our model by providing a test dataset to it. Testing the model determines the percentage accuracy of the model as per the requirement of project or problem.

DEPLOYMENT

The last step of machine learning life cycle is deployment, where we deploy the model in the real-world system.

If the above-prepared model is producing an accurate result as per our requirement

with acceptable speed, then we deploy the model in the real system. But before deploying the project, we will check whether it is improving its performance using available data or not. The deployment phase is similar to making the final report for a project.

IV. RESULTS

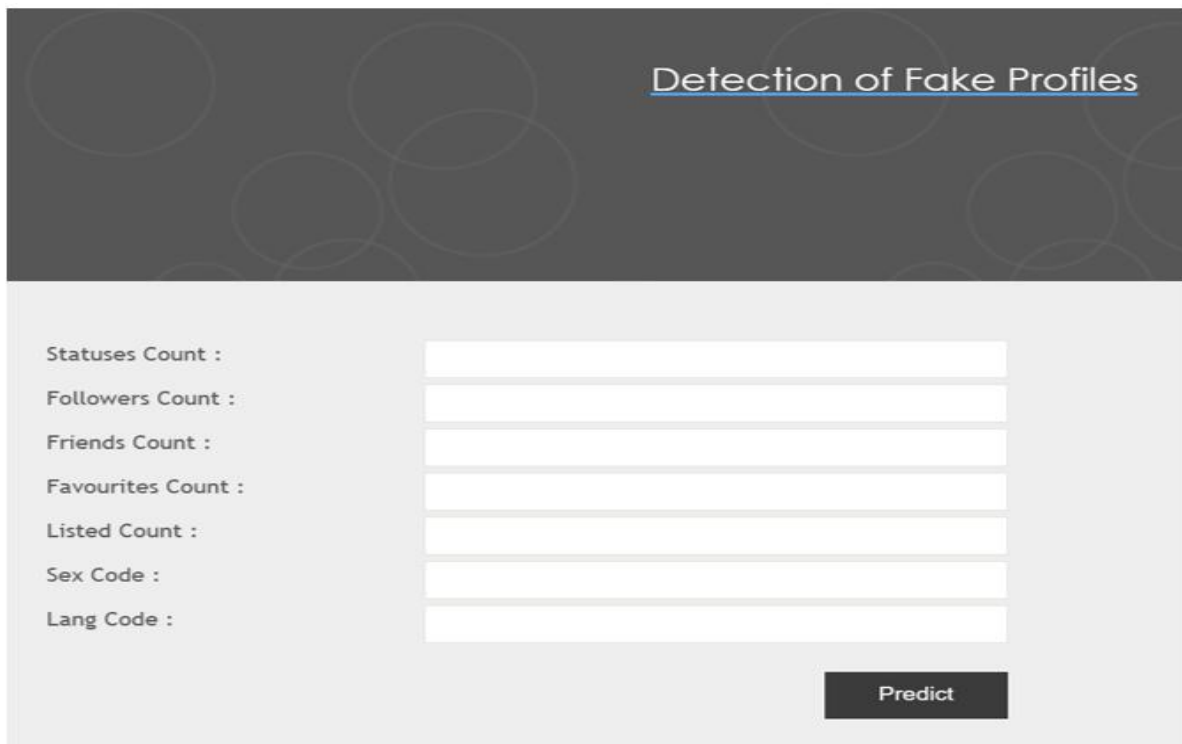


Fig.2 Give the input to detect the profile is fake or real

Detection of Fake Profiles

Statuses Count :	<input style="width: 60%;" type="text" value="20"/>
Followers Count :	<input style="width: 60%;" type="text" value="3500"/>
Friends Count :	<input style="width: 60%;" type="text" value="2000"/>
Favourites Count :	<input style="width: 60%;" type="text" value="250"/>
Listed Count :	<input style="width: 60%;" type="text" value="28"/>
Sex Code :	<input style="width: 60%;" type="text" value="1"/>
Lang Code :	<input style="width: 60%;" type="text" value="1"/>

Fig.3 Given inputs to detect the profile is fake or real

Detection of Fake Profiles

Real User

Statuses Count :	<input style="width: 60%;" type="text"/>
Followers Count :	<input style="width: 60%;" type="text"/>
Friends Count :	<input style="width: 60%;" type="text"/>
Favourites Count :	<input style="width: 60%;" type="text"/>
Listed Count :	<input style="width: 60%;" type="text"/>
Sex Code :	<input style="width: 60%;" type="text"/>
Lang Code :	<input style="width: 60%;" type="text"/>

Fig.4 Real User

Detection of Fake Profiles

Statuses Count :	<input style="width: 60%;" type="text" value="20"/>
Followers Count :	<input style="width: 60%;" type="text" value="3500"/>
Friends Count :	<input style="width: 60%;" type="text" value="1000"/>
Favourites Count :	<input style="width: 60%;" type="text" value="0"/>
Listed Count :	<input style="width: 60%;" type="text" value="80"/>
Sex Code :	<input style="width: 60%;" type="text" value="1"/>
Lang Code :	<input style="width: 60%;" type="text" value="1"/>

Fig.5 Given inputs to detect the profile is fake or real

Detection of Fake Profiles

Fake User

Statuses Count :	<input style="width: 60%;" type="text"/>
Followers Count :	<input style="width: 60%;" type="text"/>
Friends Count :	<input style="width: 60%;" type="text"/>
Favourites Count :	<input style="width: 60%;" type="text"/>
Listed Count :	<input style="width: 60%;" type="text"/>
Sex Code :	<input style="width: 60%;" type="text"/>
Lang Code :	<input style="width: 60%;" type="text"/>

Fig.6 Fake User

V. CONCLUSION

A new classification algorithm was proposed to improve detecting fake accounts on social networks, where the RANDOM FOREST trained model decision values were used to train a Neural Network model. To reach our goal we used dataset run it into the pre-processing phase where different feature reduction techniques were used to reduce the feature vector. In the classification phase, random forest learning algorithms were used. The results of the analyses showed that random forest has archived better accuracy results with all feature sets comparing with other classifiers, with classification accuracy around 98%. It was noticed that the Neural Network algorithm has the lowest classification accuracy compared with random forest.

In case some of the inputs are missing, it gets so tedious for random forest algorithm to give the results. So, gradient boosting algorithm is used to detect the account is fake or real even some inputs are missing. The Accuracy of detecting fake accounts is found to be higher using Gradient Boosting Algorithm followed by Random Forest Algorithm for a given dataset. So,

here we are using gradient boosting algorithm to detect accurate output.

REFERENCES

- [1] Ramalingam, D., Chinnaiah, V. (2018). Fake profile detection techniques in large scale online social networks.
- [2] Ferrara, E., Varol, O., Davis, C., Menczer, F., Flammini, A. (2016). The rise of social bots. *Communications of the ACM*, 59(7): 96-104.
- [3] Cresci, S., Di Pietro, R., Petrocchi, M., Spognardi, A., Tesconi, M. (2015). Fame for sale: Efficient detection of fake Twitter followers. *Decision Support Systems*.
- [4] Zhang, Y., Lu, J. (2016). Discover millions of fake followers in Weibo. *Social Network Analysis and Mining*.
- [5] Thomas, K., Grier, C., Song, D., Paxson, V. (2011). Suspended accounts in retrospect: An analysis of twitter spam. In *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*, pp.243-258.
- [6] Benevenuto, F., Magno, G., Rodrigues, T., Almeida, V. (2010). Detecting spammers on twitter. In *Collaboration, Electronic Messaging, Anti-abuse and Spam Conference (CEAS)*, p. 12

- [7] Schoonjans, F. (2019). ROC curve analysis with MedCalc. [Online] MedCalc. Available at: <https://www.medcalc.org/manual/roc-curves.php> [Accessed 10 Jun. 2019].
- [8] Kietzmann, J.H., Hermkens, K., McCarthy, I.P., Silvestre, B.S., 2011. Social media? Get serious! Understanding the functional building blocks of social media. *Bus.Horiz., SPECIAL ISSUE: SOCIAL MEDIA* 54, 241251. doi: 10.1016/j.bushor.2011.01.005.
- [9] Krombholz, K., Hobel, H., Huber, M., Weippl, E., 2015. Advanced Social Engineering Attacks. *J Inf SecurAppl* 22, 113–122. doi: 10.1016/j.jisa.2014.09.005

CERTIFICATION OF PUBLICATION

This is to certify that the paper entitled

“Fake Instagram Profile Detection and Classification Using Machine Learning”

Authored by:

Mrs.B. SHIVANI, T. TEJA SRI, N. NIKHITHA, M.A. AHAD SIDDIQUI

From

TKR College of Engineering & Technology, Meerpet, Hyderabad, has been published in

ZKG INTERNATIONAL JOURNAL, VOLUME VIII, ISSUE I, APRIL, 2023



IMPACT FACTOR

Editor-In-Chief ZKG INTERNATIONAL