Social Media Profile Tagging – A novel machine learning approach for Twitter

Final Year Project

Session 2018-2022

4th Year Students

A project submitted in partial fulfilment of the

COMSATS University Degree

of

BS in Computer Science

Department of Computer Science

COMSATS University Islamabad, Lahore Campus

June 24, 2022

Project Detail

Type (Nature of project)	[] Development [] Research [✓] R&D
Area of specialization	Machine learning

Project Group Members

Sr.#	Reg. #	Student Name	Email ID	*Signature
(i)	FA18-BCS-087	Saaim Siddiqui	fa18-bcs-087@cuilahore.edu.pk	
(ii)	FA18-BCS-203	Abdul Aziz	fa18-bcs-203@cuilahore.edu.pk	
(iii)	FA18-BCS-035	Amber Riaz	fa18-bcs-035@cuilahore.edu.pk	

*The candidates confirm that the work submitted is their own and appropriate credit has been given where reference has been made to work of others

Plagiarism Free Certificate

This is to certify that, I am **Saaim Siddiqui** S/O **Nadeem Iqbal Siddiqui**, group leader of FYP under registration no **CUI/FA18-BCS-087/LHR** at Computer Science Department, COMSATS University Islamabad, Lahore Campus. I declare that my FYP Report-II is checked by my supervisor and the similarity index is ___11___% that is less than 20%, an acceptable limit by HEC. Report is attached herewith as Appendix A.

Date: _24-June-2022___ Name of Group Leader: ______Saaim Siddiqui_____ Signature: ___ ___

Name of Supervisor: __Yella Mehroze__ Co-Supervisor (if any): ____________________

Designation: ____Lecturer_______ Designation: ____________________

Signature: ______Yella____ Signature: ____________________

Abstract

The project utilizes the sheer number of Twitter profiles and classifies them based on profuse facets. Dominant aspects have been extracted from the users of Twitter that are discernible on a profile such as the number of followers, username, retweets count, and likes count, etc. The proposed system categorizes Twitter profiles into six categories: Political, Actor, Sports, Singer, Educational, and Content Creator. A machine learning approach has been used in the proposed model that works on these abundant features. Several heterogeneous models such as Bayesian networks, SVC, Random Forest, and CNN are used in the study to achieve desirable results. The preferred system is applicable to all Twitter profiles and helps label them through profile URLs. It also helps recommender systems and Twitter to analyze the profiles based on broader categories.

Table of Contents

List of figures

List of Tables

Chapter 1

Introduction

1 Chapter 1: Introduction

1.1 Introduction

Social media has become a very integral part of everyone's life. Whether it is for entertainment or communication or sharing your opinions all over the internet with people, we find like-minded people on the internet way more easily than in real life. The amount of these people keeps increasing as the internet becomes more accessible to the public. The number of people using social media in 2021 is over 4.48 billion worldwide. Currently, 56.8% of the world's total population is using social media [1]. The most dominant and popular social media platform is Facebook which has 2.85 billion monthly active users [2]. After Facebook, Twitter is the most engaging and interactive platform. People from various backgrounds use it. If talking of the second quarter of 2021, Twitter has 206 billion daily active users worldwide [3].

The short tweet length of 240 characters [4] and interactive design of this platform enable Twitter to be the most popular social network among celebrities. This allows a very concise and to-the-point message, to be delivered to the people. Communication with the public figures on Twitter is easier than on Facebook, as the public even gets responses from them [5]. So, there are many people all together in one place who share different kinds of interests. This number keeps on increasing gradually but there is no such mechanism that can automatically identify the category to which the profile belongs to. Since every user doesn't add the details of their work or profession on their profile it is hard to find these classifications.

The idea of this project is to detect the following types of profiles: Political, Actor, Sports, Singer, Educational, and Content Creator. There are millions of profiles on social media but there is no efficient way by which it can distinguish between the categories of profiles. There is a need for a system that can identify the type or category of profiles and tag them automatically. It can be helpful for Twitter because Twitter itself doesn't have a mechanism by which profiles can be tagged. Hence this system can also be used by Twitter to get insight into users. It can also be used as a recommendation system so any user will only get the profiles that belong to the desired category entered by the user.

A huge amount of data is generated over Twitter. The official Twitter API does not directly allow access to data older than 7 days. So, the data was scrapped by using code bots which uses the browser search to get data older than 7 days [6]. Web scraper was written in python in order to perform extraction of data from Twitter. Facets were extracted that are present in the profile such as the number of followers, following, creation_date, retweet_count, likes_count, and authentication from google. After the extraction of data, different preprocessing techniques such as "Bag of words" and "TF - IDF" was applied to data to extract meaningful information from the tweets and then reconfigure the latest tweets feature.

After applying preprocessing techniques, data was trained. Manual annotators were used for the training of data. Around 6000 profiles were annotated by three annotators. In case a conflict would exist, the decision was made based on a similarity score. The similarity score makes the resultant data authentic and genuine. The authentic data allows the model to get trained properly and reduces the percentage of error in the end model. After the annotation of the data, the models were trained on the annotated data. This trained model was then tested and validated while tuning parameters to get the optimal model. The model was attached to the backend of the website so that classification can be performed. The system will analyze the profile and then tag the profile into one of the six categories.

1.2 Objectives

- To categorize the social media profiles into specific categories like Political, Religious, Sports, Entertainment, Educational, Social Activist.
- To aid different advertisement agencies
- To facilitate researchers in distinguishing their targeted profiles
- To generate a novel corpus of Twitter profiles w.r.t their categories
- To enhance the Google Searching for a particular celebrity
- To facilitate recommender systems (that can be built based on results made by this study)
- To use deep learning models to get accurate stratification.
- To learn how to tune the parameters of a classifier to give the best results.
- To facilitate the users so they can search for all profiles of a particular domain.
- To filter out the fake profiles based on their profile information from Twitter
- To create a web application that would help categorize a profile from a URL

1.3 Problem Statement

A system that would tag the profiles into specifically defined categories out of million profiles present on Twitter by using machine learning algorithms. It can also be used as a recommender system. Users can find the profiles of only their interests, and they don’t have to go through all other types of profiles that are irrelevant to them.

1.4 Assumptions and constraints

There are some assumptions and constraints that must be considered before developing a system. This includes things such as devices and the environment in which our system will be used. Below are some assumptions and constraints which we are considering for this project:

- We assume that the user has a good PC or mobile phone on which he/she will use our system.
- We are assuming that the user has a good internet connection.
- Another assumption is that the user has a basic knowledge of operating a PC or mobile phone.
- We assume that the user has knowledge of twitter links which he will be asked by the system to enter as an input to predict the category of the profile.

1.5 Project Scope

- The key purpose of this project is to build a machine learning model that will classify Twitter users' profiles.
- To build a model we will collect data of almost 8000 Twitter profiles.
- A scrapper made by using python will be used to scrape the data from Twitter profiles.
- To collect the data from Twitter profiles, a scrapper written in python will be used.
- To annotate the extracted corpus manually into different classes.
- To extract the features and their selection based on profile characteristics.
- To train a good model on the basis of well-processed and balanced data.
- To train the model, Classical Machine Learning algorithms including Support Vector Classifier, Naive Bayes, and Random Forest will be used to achieve higher accuracy.
- To build a model based on Sci-kit learn and Natural Language Processing Toolkit (NLTK) in python.
- The Development Environment will be Jupyter Notebook and Google Colab.

Chapter 2

Requirement Analysis

2 Chapter 2: Requirement Analysis

2.1 Literature Review

2.1.1 Binary Profile Classification

An extensive study is conducted to support this study. In a study titled “The Machine Learning Approach to Twitter User Classification,” they used machine learning to classify the users based on their ethnicity and political orientation. The research uses gradient-based decision trees - GBDT to find the user profile [7]. The system works on encapsulating the main topics of interest in a user’s tweets. The system is a binary class classification problem and classifies the profiles based on political affiliation and ethnicity. The system had an accuracy of 88 percent. All the results have been achieved by using the following main features.

- **Profile Features:** “Who you are?”
- **Linguistic content:** “What you tweet”
- **Tweeting behavior:** “How you tweet?”

In this research, the scope of the system was limited as it only categorized the data into two classes. Defining users on the basis of only two factors i.e., political preferences and ethnicity, cannot provide accurate results.

2.1.2 Predicting Age and Gender

In another study conducted regarding Arabic Twitter User Profiling research, the system predicts the age and gender, based on tweets, and features to address the problem of fake or suspicious profiles [8].

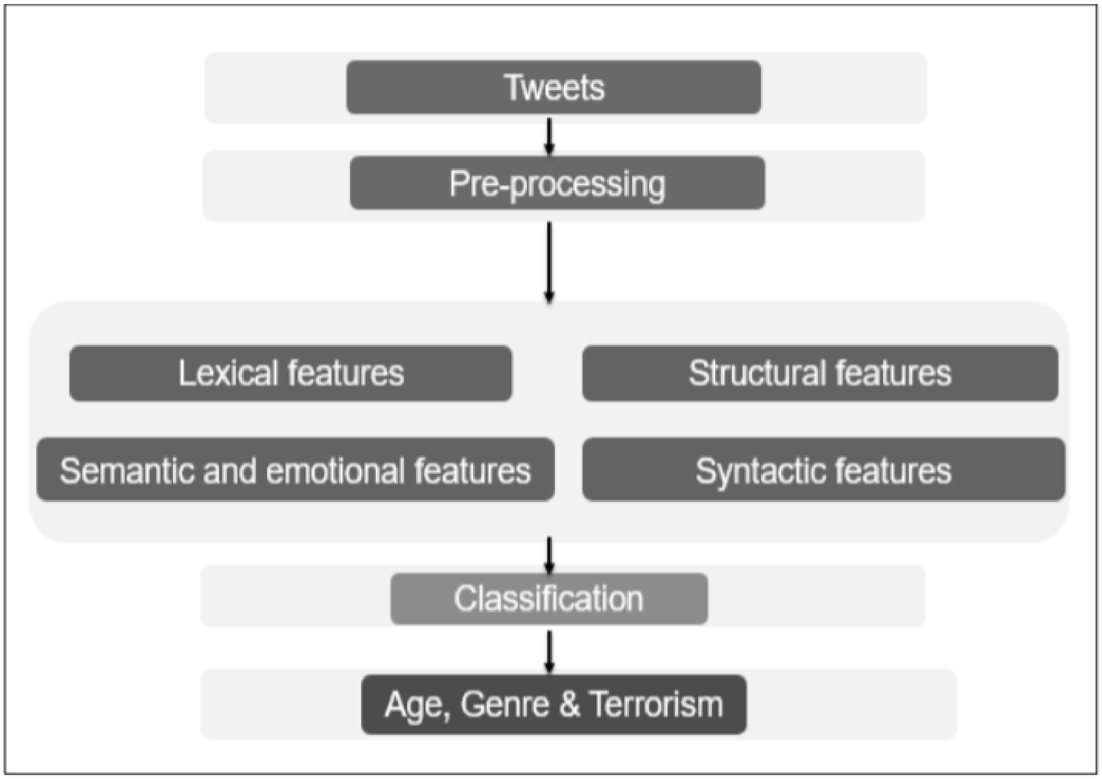


Figure 1: Architecture of our author profiling method.

The figure above shows the workflow of how the goals of the study have been achieved. In this study first of all tweets of different profiles have been collected and after that, they have been pre-processed. In pre-processing the aim was to remove noisy data by removing suffixes, prefixes, and URLs and converting plurals to singulars. This study focused on lexical features and syntactic features to build the model.

A total of 143 attributes were used which were then normalized. The model was built using the SVM algorithm which gave an accuracy of 73.49% for age, 83.7% for gender, and 88.7% for detecting dangerous profiles. The system uses several words based and character-based features to predict the age, gender, and fakeness of a profile.

This study uses gender and age classification with SVM to find suspicious profiles. The features of the model are not sufficient to find accurate classification. SVM is a classical machine learning algorithm the accuracy of the model can be improved by using deep learning systems.

2.1.3 Fake Profile Classification

The research of characteristics of fake profiles was encountered in a study that used pattern-matching algorithms and map-reducing techniques to detect fake profiles. The research deals with the fake accounts on Twitter. They mainly dealt with bot accounts detection on Twitter [9]. They obtained various attributes and then reduced the number of attributes to 9. The fake profiles were detected by comparing the creation time of two profiles and interval times between two consecutive profiles. The research found out that the fake profiles were being updated in a non-uniform distribution (figure b) where the true profiles were being updated on Sundays and Mondays largely (figure a).

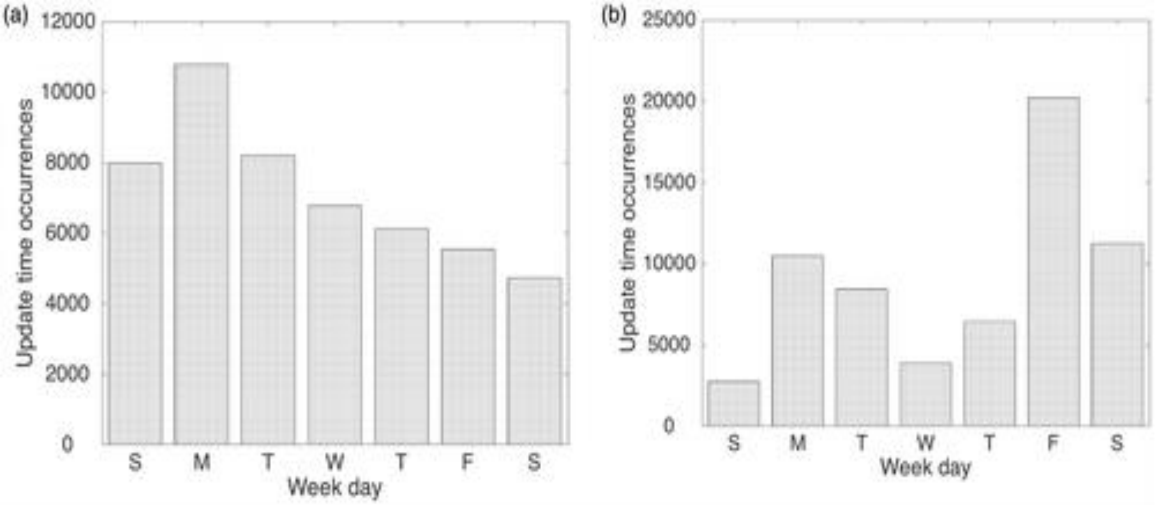


Figure 2: Update Time Occurrences

The research further investigated the creation time of both profile types and found out that the true or legitimate profiles were created uniformly throughout the week (figure a), whereas the fake profiles were mainly created in the later part of the week (figure b).

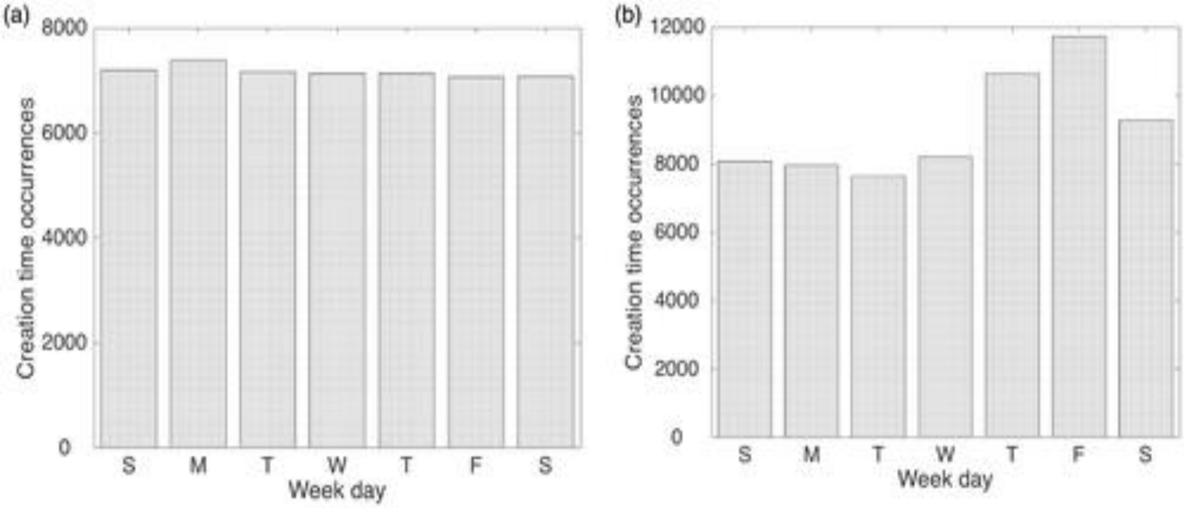


Figure 3: Creation Time Occurrences

The research observed that fake profiles have a larger number of friends and followers as compared to legitimate profiles. The friends to followers ratio of a profile was used to compare the fake profiles against legitimate profiles. True profiles had a friend-to-follower ratio of one whereas the fake profiles were seen to have a higher ratio.

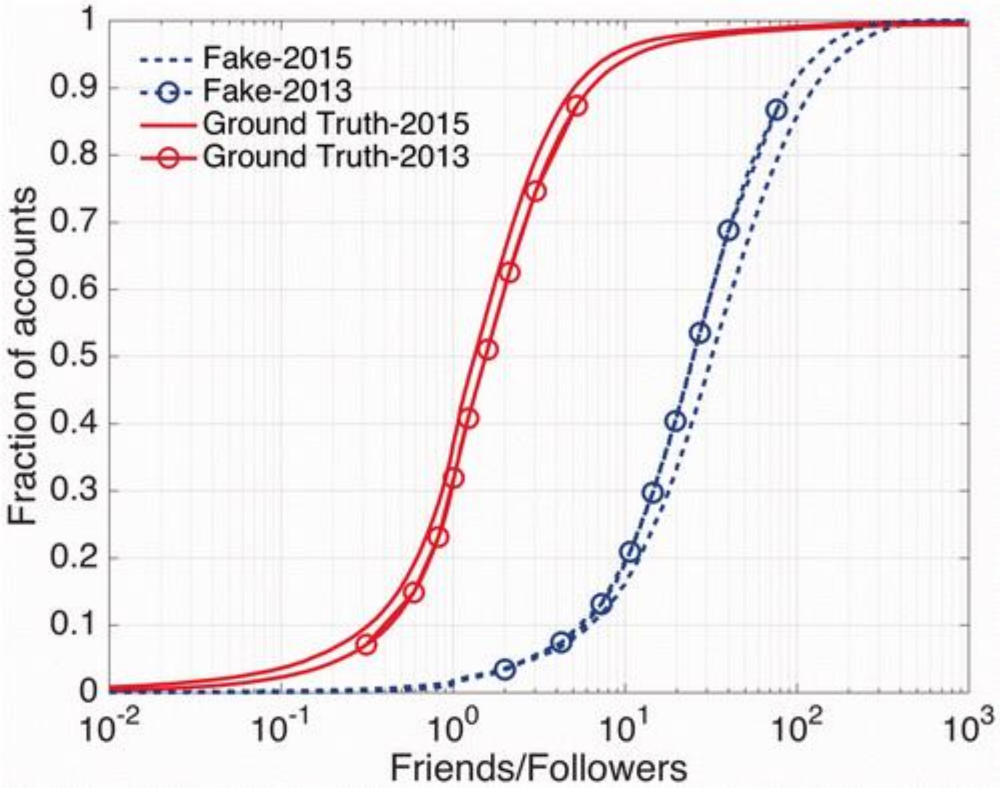


Figure 4: Friend to Follower Ratio of Fake and Real Profiles.

The scope of the project is only limited to accounts that are created autonomously that are also known as bot accounts. It does not classify other fake accounts which are possible identity theft. To widen the scope of the project and add more categories so that identity theft profiles can be caught as well.

2.1.4 Gender Classification on Twitter

Another research using Twitter classification titled “Empirical Evaluation of Profile Characteristics for Gender Classification on Twitter” was considered as literature for this research [10]. The study classified the gender of various twitter profiles based on dominant profile facets. The research explored the profile characteristics instead of tweets from a particular person to classify the gender of a profile. The aspects extracted from any profile are then given certain weights to determine their strength or importance in the classification of certain profiles. The study used a novel research technique to reduce the number of features from thousands to hundreds. The study converted names to phonemes to be used as a feature. They used the LOGIOS lexicon tool for this conversion.

Figure 5: Cloud tagging of phonemes of male users and female users

The above image shows the names converted into phonemes to use them as features, LOGIOS lexicon tool is used for this conversion.

The research considered a large dataset of 194,293 of which 104,535 were classified as male and 89,758 as females. Seven fields were collected from each profile. NB-tree classifiers were used which considered five features and obtained an accuracy of 82.5% with phenomes using the 3-gram features.

Table 1: Comparing Results with Phonemes and without Phonemes

	1-gram	2-gram	3-gram	4-gram	5-gram
Without phonemes (n-gram applied to characters of names)					
NB	NA	65.3	67.0	69.2	75.1
DT	NA	68.2	69.3	72.0	76.3
NB-TREE	NA	69.3	70.7	74.0	78.3
With phonemes (n-gram applied to set of phonemes)					
NB	65.2	65.3	66.0	NA	NA
DT	78.5	79.2	82.5	NA	NA

The study mainly considered the self-declared gender information on different social media websites by Twitter users. This can question the validity of the project. The study checked numerous profiles manually, but it was impractical hence only 5000 profiles could be manually checked.

2.1.5 Troll Profiles Detection

A study to detect troll profiles on Twitter was also considered as literature for this research. The abundance of fake profiles on Twitter has given birth to many cases of cyberbullying. The study dealt with the problem using supervised machine learning. Various twitter characteristics in context with text analysis were used to relate the fake profiles with the true user profile. Numerous profiles were studied to determine the authenticity of the profile [11]. The study used a java-based approach to collect data from numerous accounts due to the limitation caused by Twitter API to only select a few tweets in the same hour. Random Forrest and KNN machine learning models were used. The random forest model gave an accuracy of 66.48% and KNN gave an accuracy of 61.06%. A solution for the lower accuracy can be to use more NLP techniques.

The project lacked the advanced NLP and opinion mining techniques which is why low accuracy was encountered. The extracted features from Twitter profiles could not be properly related to troll Twitter profiles.

An extensive literature revealed that no or very little work has been done in this domain. There is a need to enhance the work relevant in this domain by considering more features in comparison with the features considered in previous studies. There is also a need to expand the classification scope by considering the problem as a multi-classification problem as compared to binary classification problem which was done in the previous studies. The literature discussed above considers various methods for feature derivation such as n-grams, parts of speech tags, and text-based profile characteristics. Using more dominant features of the profile along with their tweets will increase the accuracy. Based on the literature review made, this study will be providing solutions to the limitations in the form of better approaches for categorical classification.

2.2 Stakeholder’s list (Actors)

The stakeholders of the system can be defined as the end-users of the system. Stakeholders of a system are the people who stand to be merited by the system. Hence it becomes necessary to understand the difference between stakeholders and actors of the system. All stakeholders are the actors of a system, but all the actors cannot be considered as stakeholders of the system. The stakeholders of the system must stand to benefit by the system in any way, while the actors of the system can just be the person who stands to get the usability out of the system,

Table 2: Stakeholder List

User	Level of computer knowledge	Level of business knowledge	Frequency of use
User	Basic computer knowledge of how to search the web and insert links.	Good knowledge of the audience user wants to target so that he can select the particular kind of profile to contact.	Daily basis.

2.3 Requirements Elicitation

2.3.1 Functional Requirements

2.3.1.1 FR01: Introduction

Table 3: FR01: Introduction

FR01-01	The system shall be able to introduce itself
FR01-02	The system shall give the user instructions about the functionalities

2.3.1.2 FR02: Display

Table 4: FR02: Display

FR02-01	The system shall be able to display all defined categories
FR02-02	The system shall display the category of the searched profile by the user.

2.3.1.3 FR03: Search profile

Table 5: FR03: Search Profile

FR03-01	The System will allow the user to enter the URL into the text field.
FR03-02	The system shall ask the user to press the Search button to find the category of provided link.
FR03-03	The System will allow the user to view the Twitter account category associated with that particular link.

2.3.1.4 FR04: View history

Table 6: FR04: View history

FR04-01	The system will allow the user to view his previously searched links.
FR04-02	The System will allow the user to view the account type of his previously inserted link into the website.
FR04-03	The system will store the history of a session only.

2.3.2 Non-functional Requirements

2.3.2.1 NFR01: Performance

Table 7: NFR01: Performance

NFR01-01	The model should not predict instantly, there should be a few seconds delay between responses
NFR01-02	The system should give responses to the user 24/7.
NFR01-03	The average load time of the starting page of the system must be less than 5 seconds.

NFR01-04	The average processing time taken by the system to complete a request should be less than 15 seconds.
NFR01-05	System Mean Time to Failure should not be more than 50 seconds within one day of use.
NFR01-06	The average system response time should not be greater than 10 seconds.
NFR01-07	100 users should be able to simultaneously access the system with a response time not greater than 10 seconds.

2.3.2.2 NFR02: Reliability

Table 8: NFR02: Reliability

NFR02-01	The system should be 100 percent reliable for one year under normal usage conditions which means that there is a 100 percent chance that the system won’t experience critical failures and it should be tested.
NFR02-02	The system should have the reliability to serve a minimum of 100 users at a time without any critical failure.
NFR02-03	The average time between a failure should be less than a minute.

2.3.2.3 NFR03: Usability

Table 9: NFR03: Usability

NFR03-01	The system should be easy to use for users who do not have sufficient technical knowledge.
NFR03-02	The system interface should provide guidelines to the users who don’t have any knowledge on how to use it.
NFR03-03	The system should be 100 percent efficient so that the users can achieve

	their particular goals.
NFR03-04	Usability of the system should be easy so that there will be no chance of any errors.
NFR03-05	Both beginners and experienced users should be able to perform the task efficiently in a short amount of time.
NFR03-06	The chance of errors to perform the required task should be low.

2.3.2.4 NFR04: Portability

Table 10: NFR04: Portability

NFR04-01	The system shall run on multiple operating systems including Windows, Mac, Linux, and Android.
NFR04-02	The system should be able to run on any browser.

2.3.3 Requirement Traceability matrix

Table 11: Requirement traceability matrix

No	**Functional requirements**	**Use Case**	Priority	Test case
1.	FR-01	The system should introduce itself	low	03
2.	FR-03	The system shall display the classified search profile	high	01
3.	FR-04	The system shall display the history of a session	medium	02

2.4 Use Case Description

Table 12: Use case Search link

Use case ID: 001	**Use case Name: Search link**
Priority	**High**
Primary Actor: User	
Other Participating Actors: None	
Source:	Requirement – FR03-01 Requirement – FR03-02 Requirement – FR03-03
Use Case Summary	The search link option allows the user to add the URL link of a profile to predict the results.
Pre-condition:	Web page loaded
Normal Course of Events	**Alternate Path**
1. User will press the search bar	
2. User will enter the profile link	
3. User will hit enter	
Conclusion	This use case concludes when the user hits the enter button
Post Conditions	
The profile category will be displayed.	
Implementation constraints and specifications	
The system should give responses to the user 24/7.	
Use Case Cross References	
Includes	Twitter profile URL
Extends	None
Exceptions	
None	

Table 13: Use case search history

Use case ID: 002	**Use case Name: View Search History**
Priority **Medium**	
Primary Actor: User	
Other Participating Actors: None	
Source: Requirement – FR04-01 Requirement – FR04-02 Requirement – FR04-03	
Use Case Summary	Search history feature allows the user to view their previously searched profiles information
Pre-condition:	There should be some profiles searched previously.
Normal Course of Events	**Alternate Path**
1. The user will press the history button	
Conclusion	This use case concludes when the system displays the history.
Post Conditions	
The history will be displayed	
Implementation constraints and specifications	
If the user has not searched anything before in the current session, the history will not be displayed.	
Use Case Cross References	
Includes	None
Extends	None
Exceptions	
None	

2.5 Use Case Design

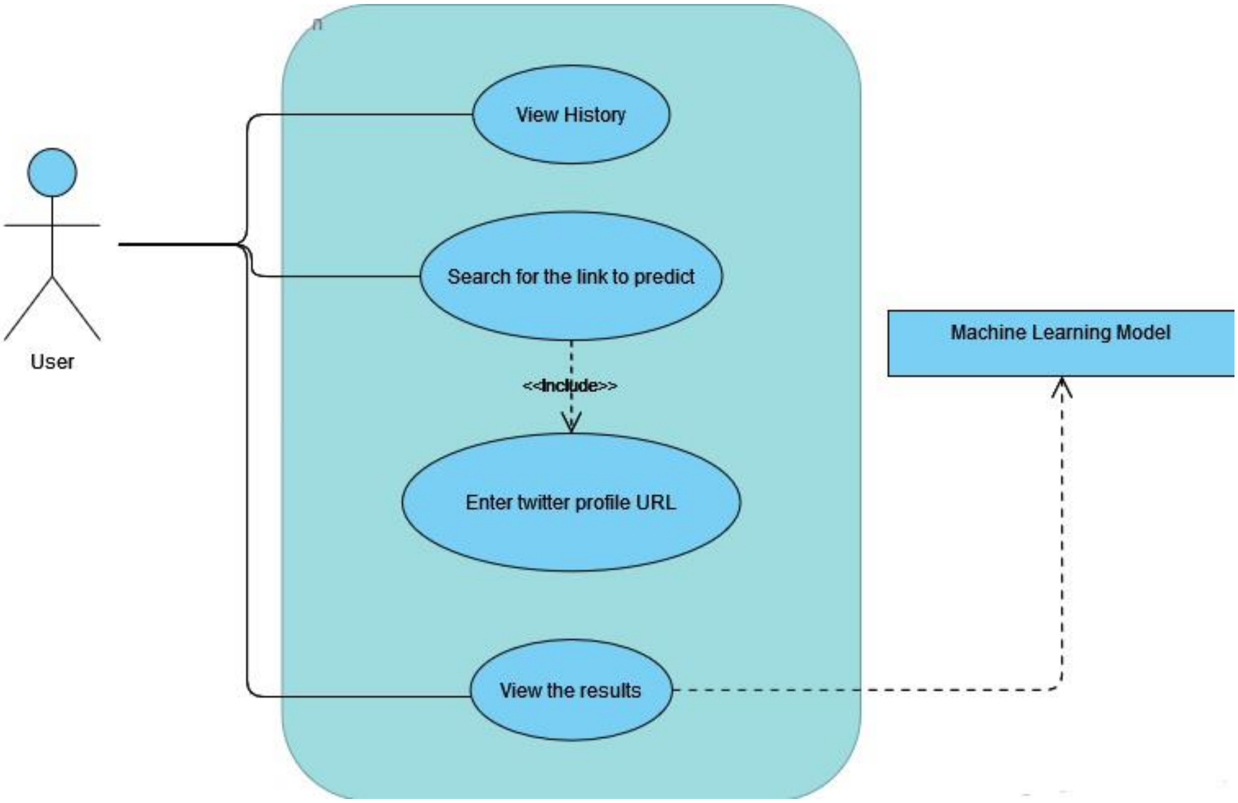


Figure 6: Use Case Design

2.6 Software development life cycle

The software development life cycle model is the way for the software development team to make their way through the challenging and complex processes that have to be carried out during the whole duration of development.

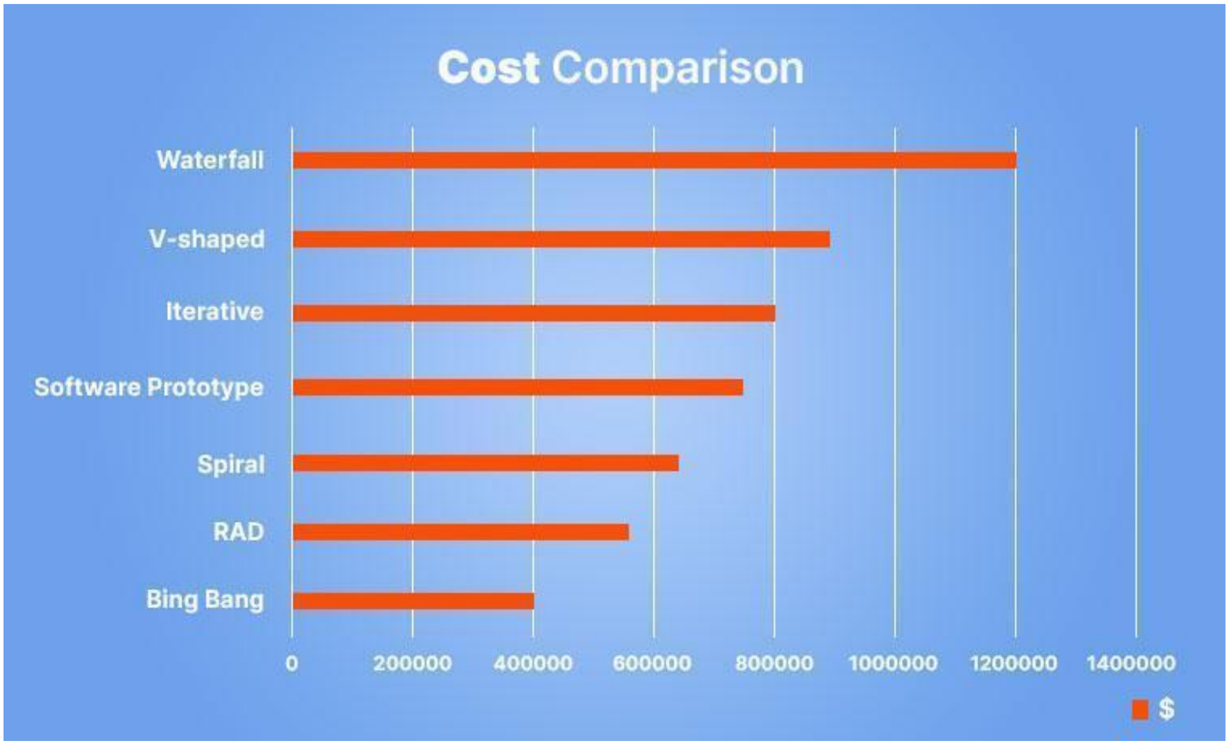


Figure 7: Cost-comparison of different SDLC models

Above statistics shows the cost comparison between different SDLC models and it shows that the waterfall model is the most expensive yet the most used model as well due to its very straightforward and traditional design.

2.6.1 Model to be used in our project:

The SDLC model most suitable for our project according to its nature is the Incremental model.

2.6.2 Incremental model

Incremental model is another version of the waterfall model which means that its design is based on the waterfall model, the difference here is that in this model the requirements are broken down into smaller components and then these components are processed iteratively and go through the basic stage of development. In each iteration there is an increment in the functionalities of the system, it keeps iterating until all required functionalities are achieved.

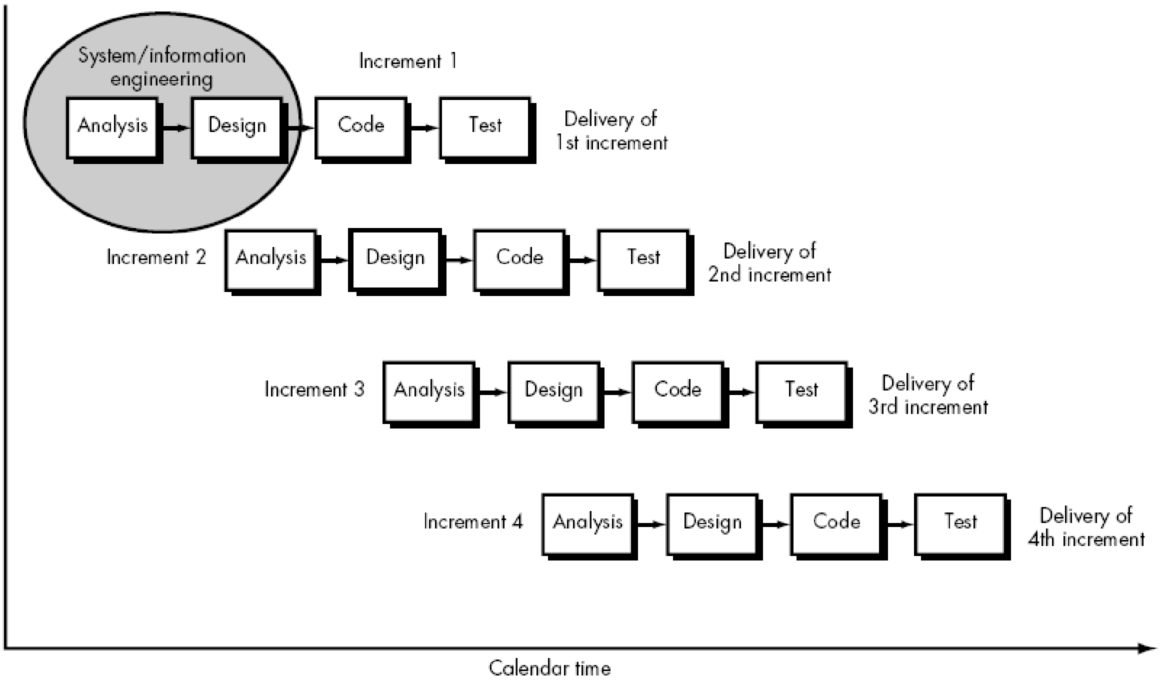


Figure 8: Standalone components are being developed separately.

The above image shows the working process of the incremental model, and how in each increment a process is completed, and in each increment, a component is developed and is added to the system. Once a component is developed all the focus is shifted to the next component, and the previously developed component is not touched during that time.

2.6.3 Why we are using this model

- The reason why this model is the most suitable for our system is that we have divided our system into smaller components and the development of our system will be carried out in iterations.

- The requirements of our system are clear.

- As a machine learning model requires parameter tunings several times in a project so this model offers us to do multiple iterations and revisions by which we can achieve the required accuracy.

- There can be changes in scope which would be easy to add because this system is being developed in components.

- Testing the system at several points in the system would be a much-needed factor in this system that this model offers.

Chapter 3

System design

3 Chapter 3: System design

3.1 Work breakdown structure

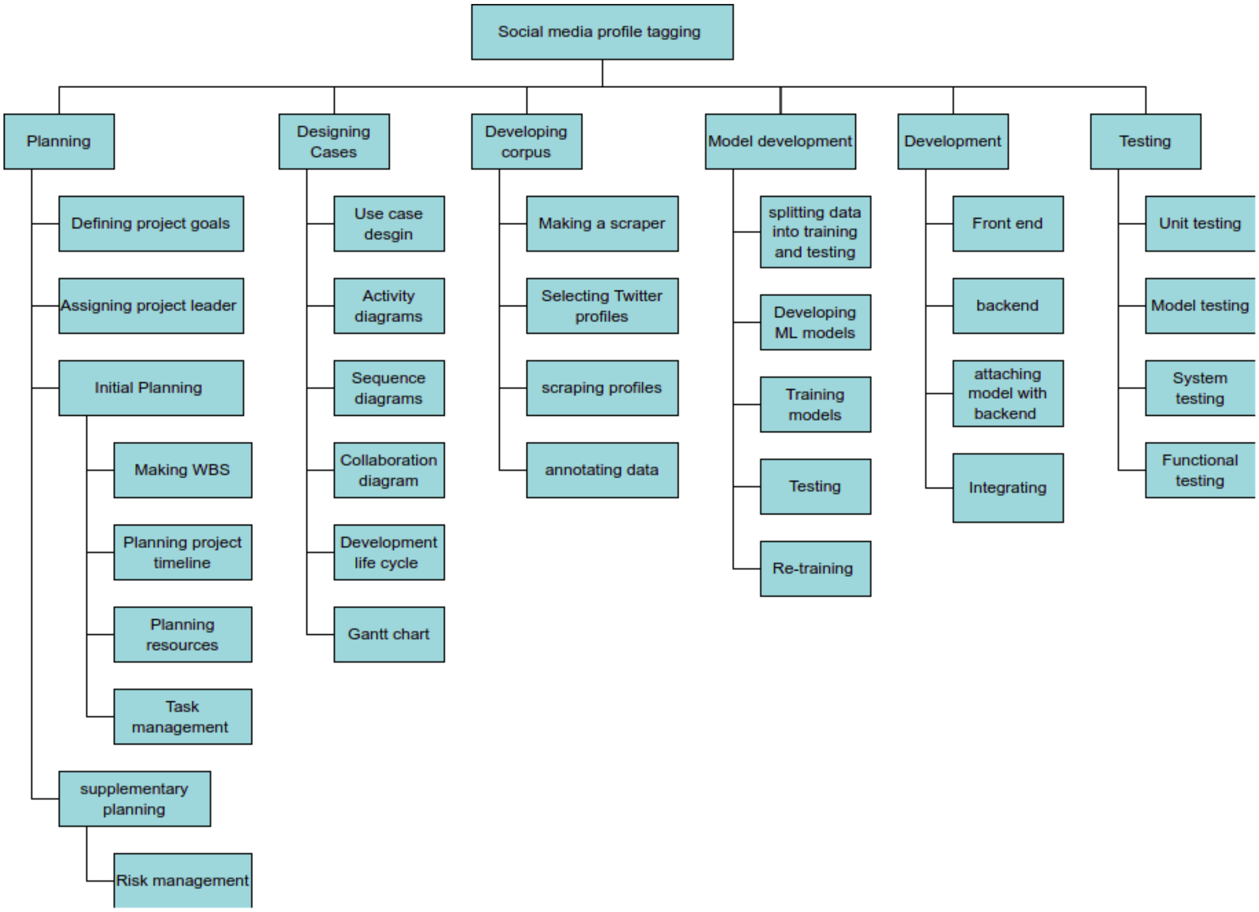


Figure 9: Work Breakdown Structure

3.2 Activity diagrams

3.2.1 AD01: Search Profile

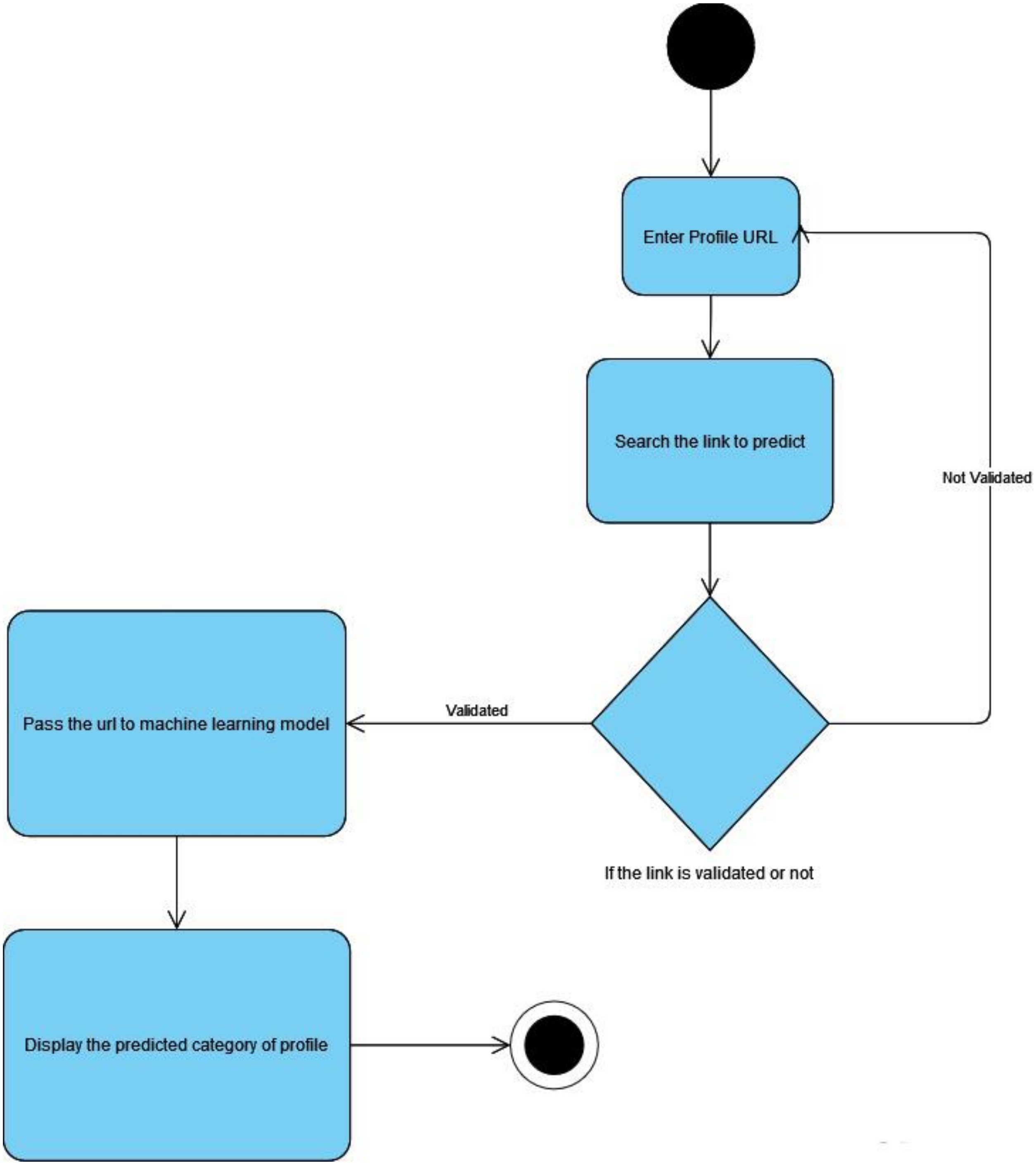

Enter Profile URL
Search the link to predict
Not Validated
Validated
Pass the url to machine learning model
If the link is validated or not
Display the predicted category of profile

Figure 10: AD01: Search profile

3.2.2 AD02: Viewing Search History

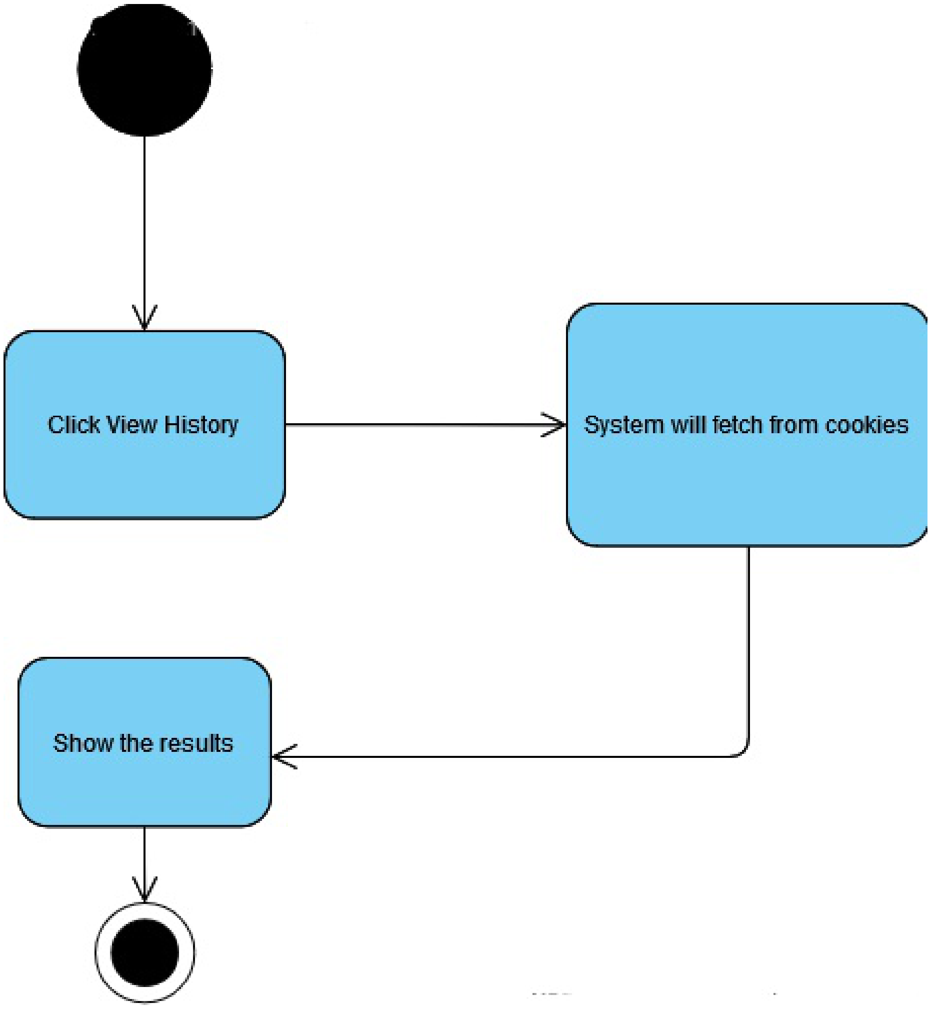


Figure 11: AD02: View Search History

3.3 Sequence diagrams

3.3.1 SD01: Search profile

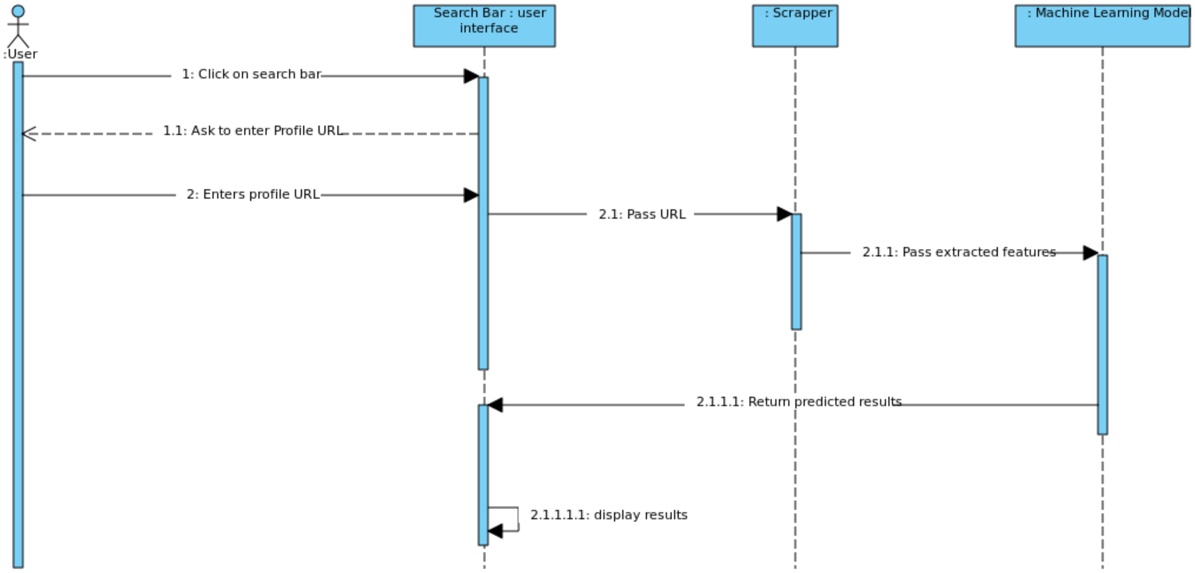

:User
Search Bar : user interface
: Scrapper
: Machine Learning Model
1: Click on search bar
1.1: Ask to enter Profile URL
2: Enters profile URL
2.1: Pass URL
2.1.1: Pass extracted features
2.1.1.1: Return predicted results
2.1.1.1.1: display results

Figure 12: SD01: Search profile

3.3.2 SD02: Viewing Search History

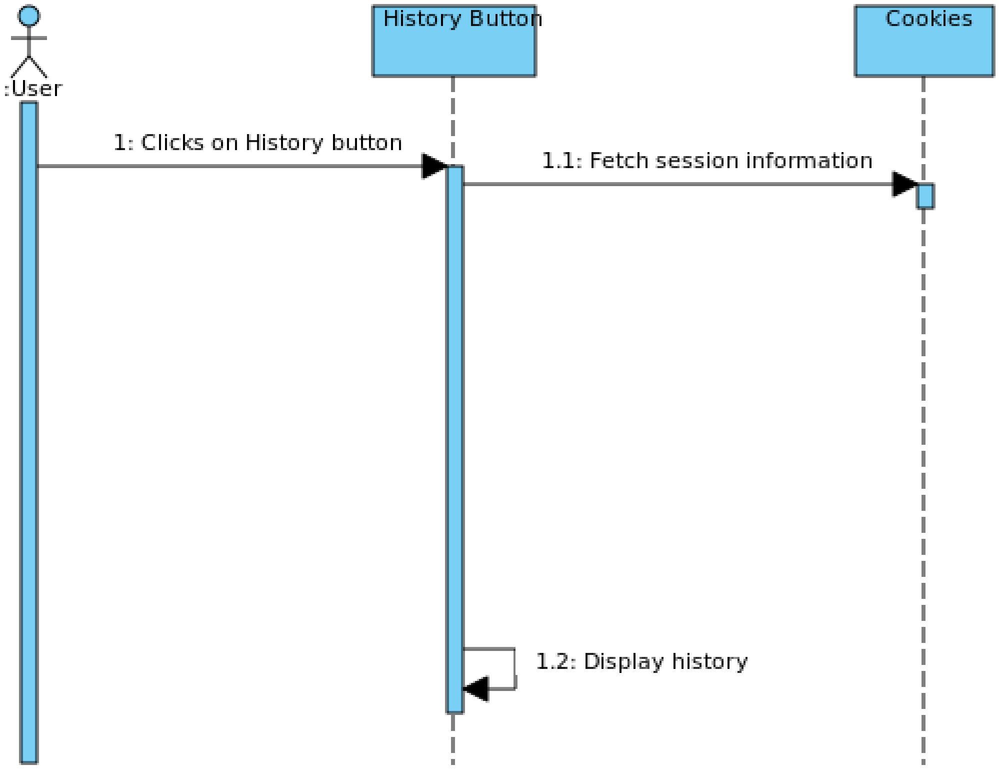


Figure 13: SD02: View search History

3.4 Class Diagram

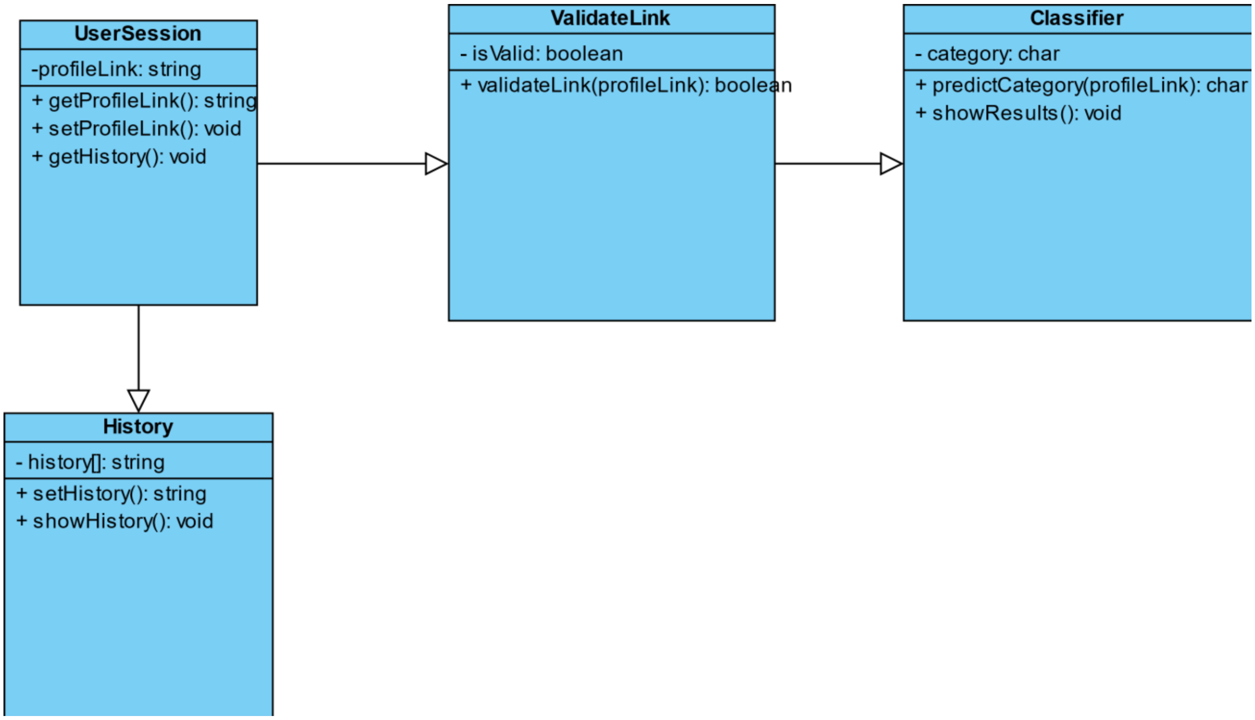


Figure 14: Class Diagram

3.5 Gantt chart

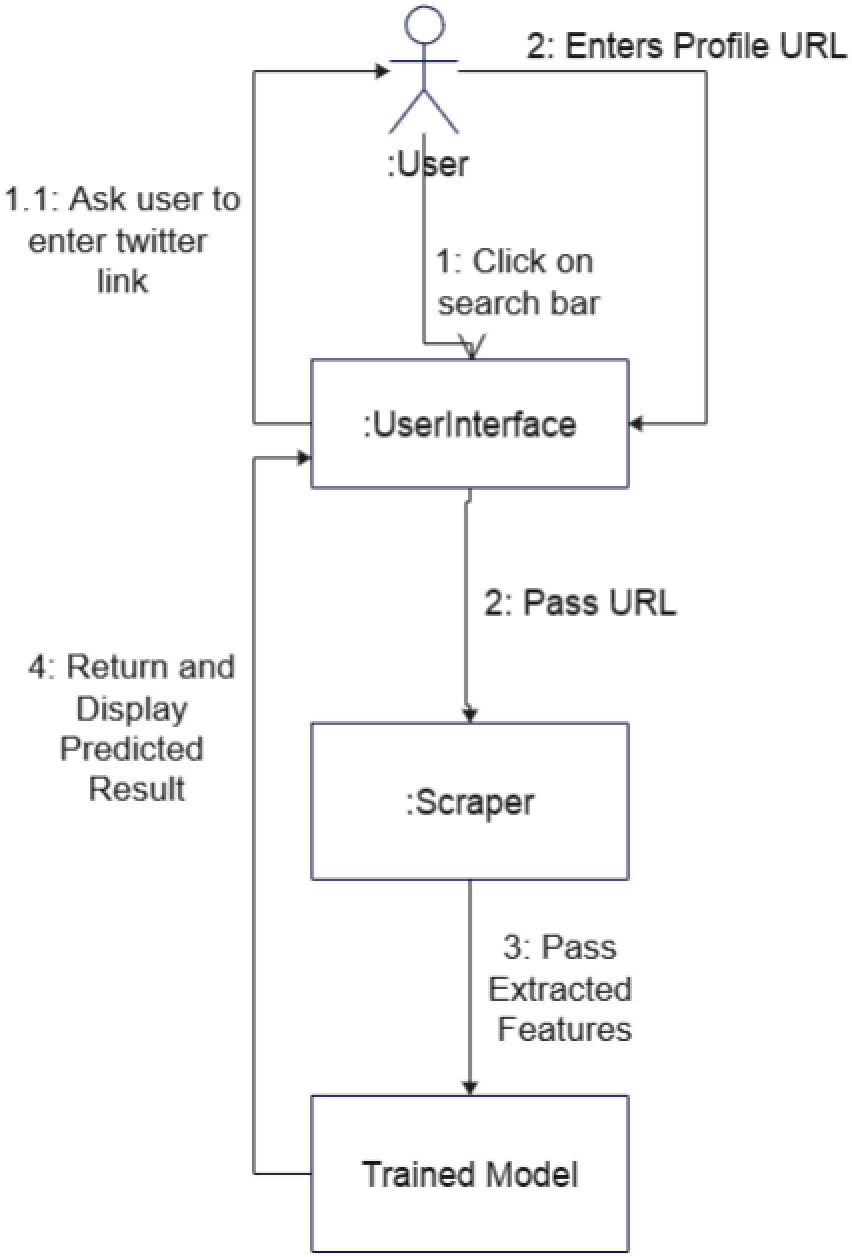
Table 14: Gantt Chart

Tasks	Sep 2021	Oct 2021	Nov 2021	Dec 2021	Jan 2022	Feb 2022	Mar 2022	Apr 2022	May 2022	Jun 2022
Planning	▨									
Building a scraper		▨								
Data Collection/ Extraction			▨							
Data Annotation				▨						
Data Preprocessing				▨	▨					
Model Development						▨	▨	▨		
Model training						▨	▨	▨		
Testing and Validation						▨	▨	▨		

Parameter Tuning						▨	▨	▨		
Analyzing Results									▨	▨
Web Interface									▨	▨
Deployment										▨

3.6 Collaboration diagram

3.6.1 CD01: Search profile



:User
2: Enters Profile URL
1.1: Ask user to enter twitter link
1: Click on search bar
:UserInterface
2: Pass URL
4: Return and Display Predicted Result
:Scraper
3: Pass Extracted Features
Trained Model

Figure 15: CD01: Search profile

3.6.2 CD02: Viewing Search History

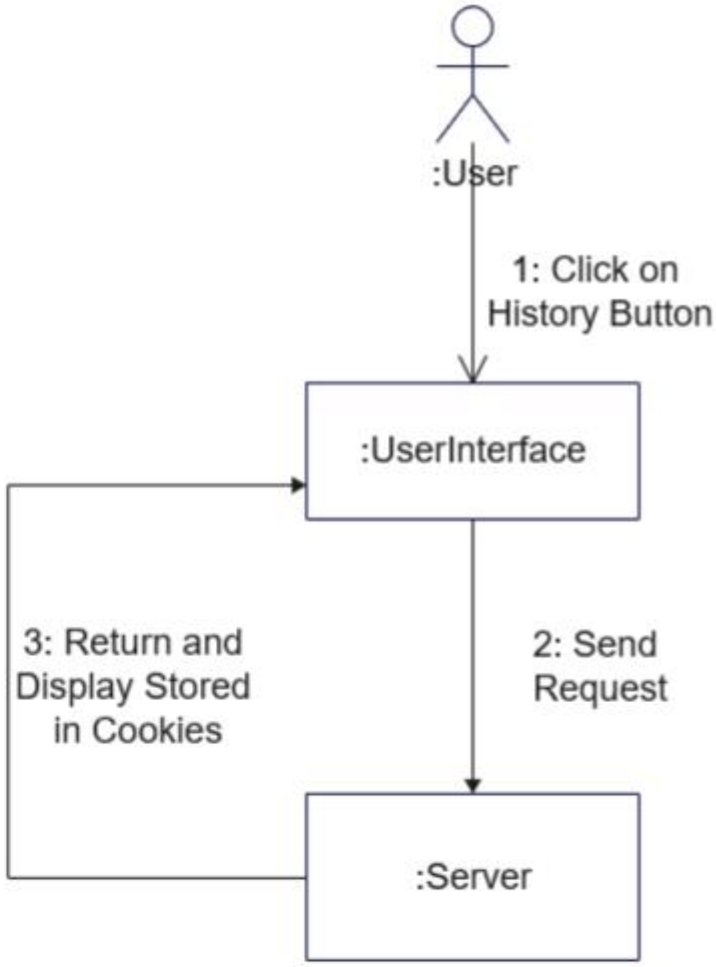
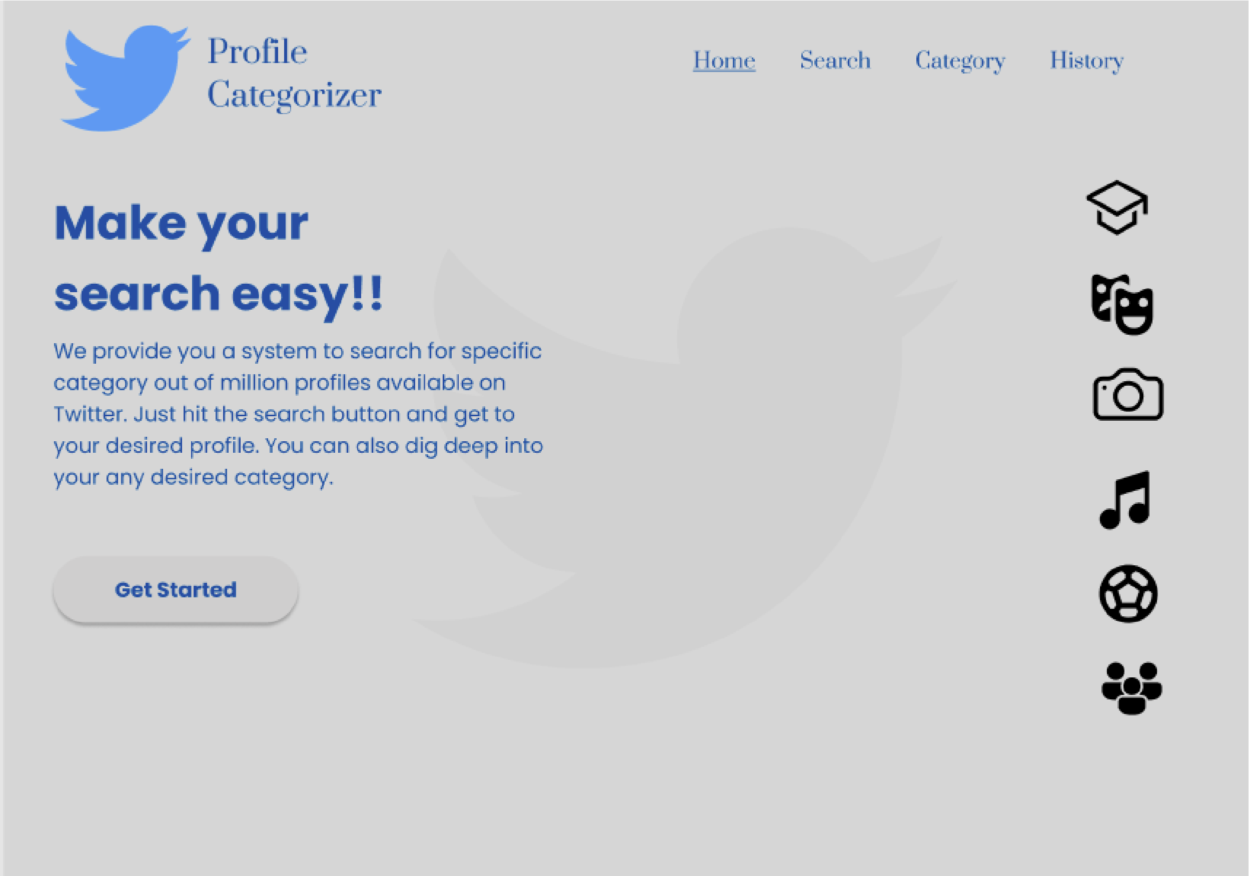

:User
1: Click on History Button
:UserInterface
3: Return and Display Stored in Cookies
2: Send Request
:Server

Figure 16: CD02: Viewing Search History

3.7 Prototypes

3.7.1 Home Page



Make your search easy!!

We provide you a system to search for specific category out of million profiles available on Twitter. Just hit the search button and get to your desired profile. You can also dig deep into your any desired category.

Figure 17: Home Page

3.7.2 Search Page

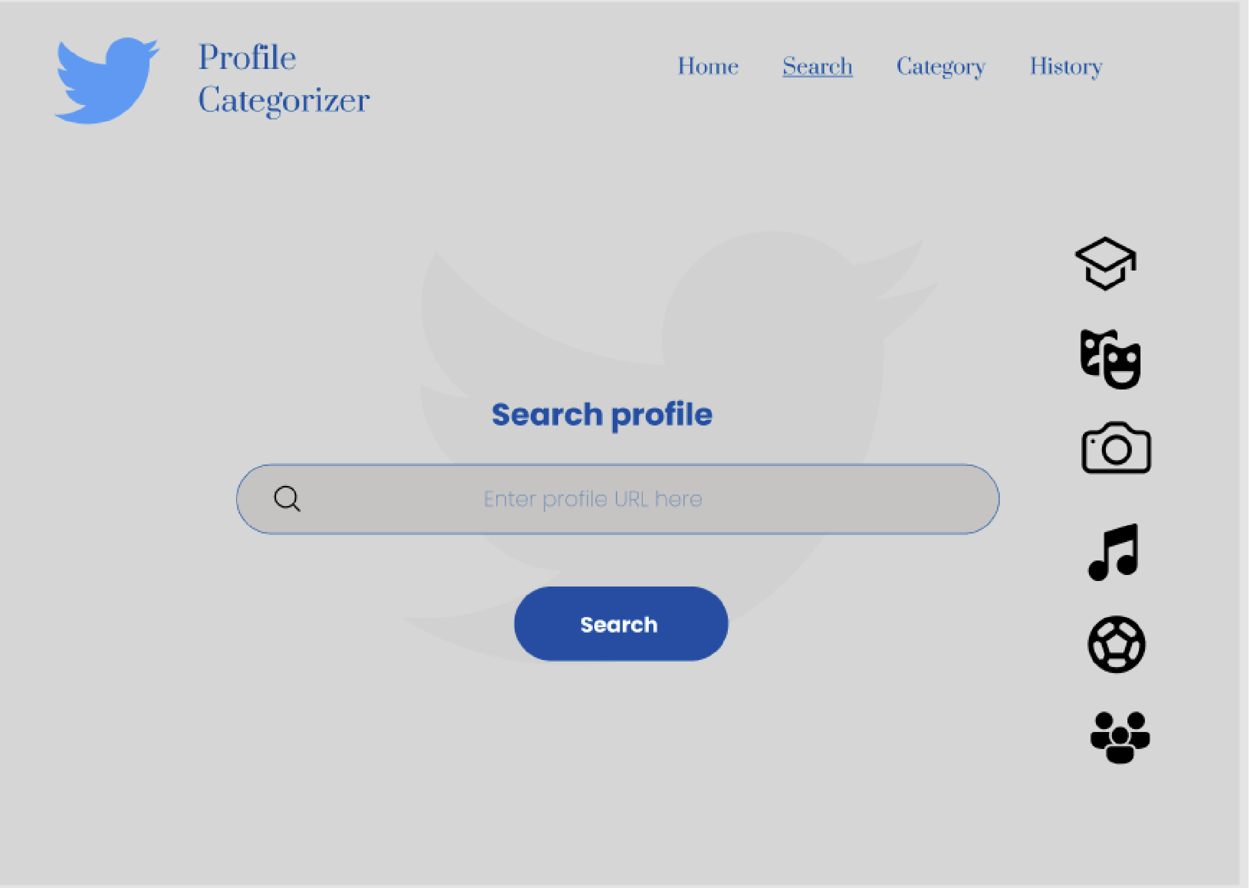


Figure 18: Search Page

3.7.3 Results Page

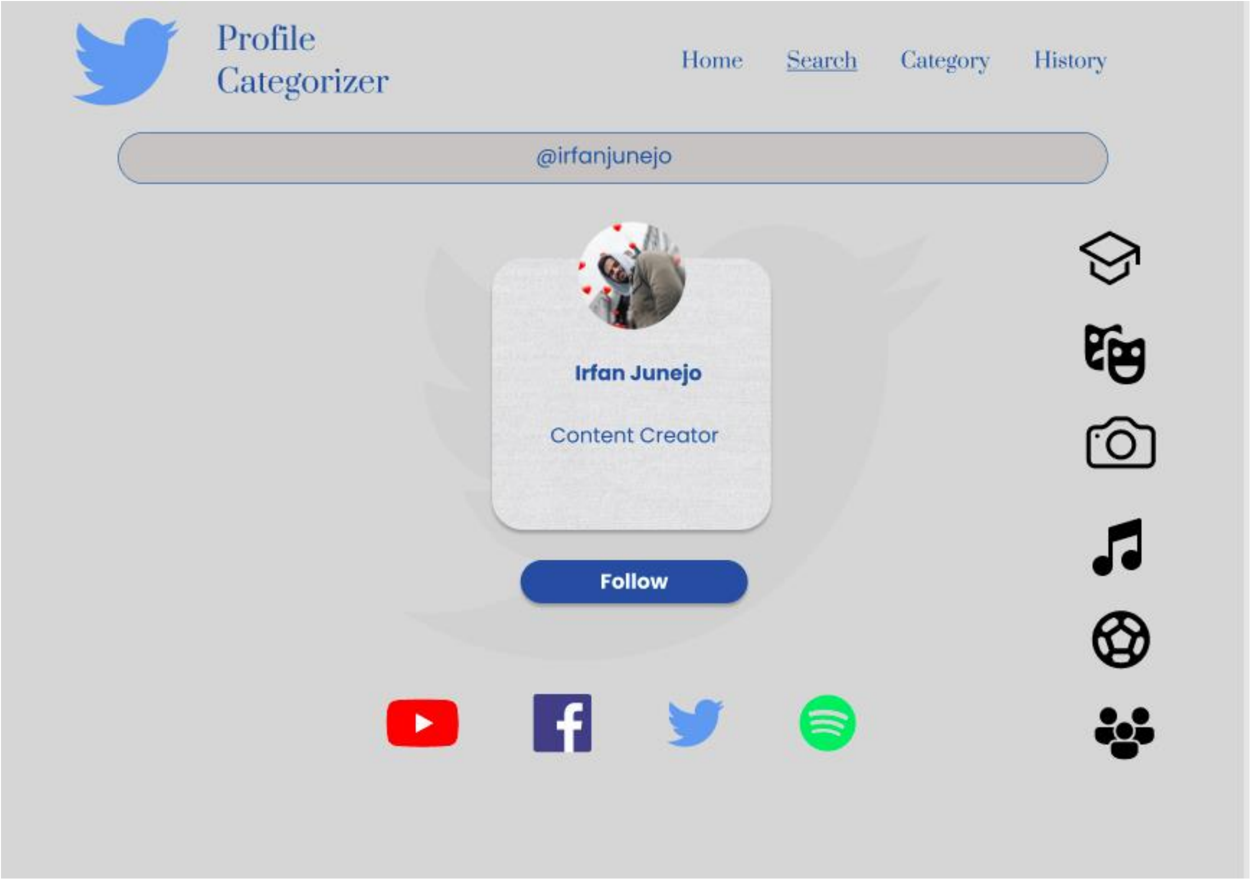


Figure 19: Result Page

Chapter 4

Machine Learning Approaches

4 Chapter 4: Model Architecture

4.1 Introduction

Twitter is a popular blogging platform that has been used in different kinds of analysis research. The tweets can be from all kinds of people. The main motive of this system is to classify the Twitter profiles or tag the Twitter profiles. The system proposed in this report grouped the profiles on Twitter into one of the six categories by using machine learning models. These six categories of profiles are given below:

- Political
- Actor
- Sports
- Singer
- Educational
- Content Creator

4.2 Building a scraper

We used a scraper to first scrape the data off Twitter to build this corpus. First, the scraper was built on selenium to fetch the twitter handlers of profiles which belonged to a particular category on a page. A scraping bot was created in python using the GitHub library sncrape, which was used to scrape a required set of information from Twitter accounts. Two scrapers were built to extract all the information required to train the models. Second scraper collected all the information related to that particular profile. The demo code for the scraper is given below which finds the profile of one category social activist.

```
from selenium import webdriver
from selenium.webdriver.chrome.options import Options

options = Options()
options.add_argument("start-maximized")

browser = webdriver.Chrome(chrome_options=options,
executable_path=r'/mnt/sda6/Univerisity/Semester 7/Final Year
project/Scraper/chromedriver_linux64/chromedriver')

browser.get('https://www.twitter.com/login')
username = browser.find_element_by_xpath('//input[@name="text"]')
username.send_keys('SuicidalPastaa')
username.send_keys(Keys.RETURN)
```

```
   my_password = getpass()
password = browser.find_element_by_xpath('//input[@name="password"]')
password.send_keys(my_password)

search = browser.find_element_by_xpath('//input[@aria-label="Search query"]')
category = 'social activist'
search.send_keys('{} "filter:verified"'.format(category))
search.send_keys(Keys.RETURN)

browser.find_element_by_link_text('People').click()

twitter_handles = []
i = 0
while i<10:
    users = browser.find_elements_by_xpath('//div[@data-testid="UserCell"]')
    for user in users:
        userHandle = user.find_element_by_xpath('.//span[contains(text(),
"@")]').text
        twitter_handles.append(userHandle)
    browser.execute_script('window.scrollTo(0, document.body.scrollHeight);')
    sleep(1)
    i = i + 1

import pandas as pd
dict = {'social activist': twitter_handles}
df = pd.DataFrame(dict)

df.to_csv("social.csv")
```

The profiles were authenticated through google information on the personality as well. We used twitter verified profiles to check only real profiles. This information extracted from the accounts before the pre-processing of the dataset includes

- Account ID
- Username
- Profile link
- Number of followers
- Number of following
- 10 Latest tweets
- Retweets count
- Likes count

- Location

4.3 Preprocessing

Once the data has been scrapped it was preprocessed to see if there are any null values for example, if an account was deleted or if there was an exception while scraping. This extracted data was raw and needed to be converted into features to make the data meaningful for the machine learning models. The preprocessing includes deleting the extra features that were extracted from the profiles which are not needed for the training. After deleting the unnecessary features, we were left with:

- Account URL
- 10 latest tweets
- Description
- Location
- Verified status

4.4 Annotation

After converting the raw data into meaningful features, the corpus was annotated manually. The entire corpus was annotated by three annotators among which the majority rule was followed. This means if annotators conflict on an entry, the category on which the majority of annotators agree was used. The target was to create a dataset of around 6000 profiles which will help train the machine learning models. This data was stored in a csv file.

4.5 Machine learning Models

Then the system was built by using various machine learning models which are explained below. Classical machine learning algorithms were used on the extracted features to see if the required accuracy is achieved and then the project was moved on to deep learning models where these models were applied to the tweets of various profiles to extract meaningful information from them. Both Classical and deep learning models are being used because sometimes Classical machine learning outperforms some of the complex decision trees such as given in the research by Jason Brownlee where classical algorithms outperformed Multilayer perceptron and Long Short-Term memory. (LSTM) [11]. Hence, we studied both algorithm types to see which gives the more accurate results.

4.6 Classical learning models

Classical machine learning algorithms do not use multilayer architectures and they work better on smaller ideas. Classical machine learning algorithms use direct feature engineering which makes them easy to interpret. Following are the classical models

- Naive Bayes algorithm/ Bayesian networks
- Support vector classifiers
- Random Forest classification.

4.6.1 Naive Bayes algorithm/ Bayesian networks

A Naive Bayes classifier works on the probabilities of various events occurring. The model works best for the text classification tasks. The model has less training time and requires a small amount to be trained efficiently.

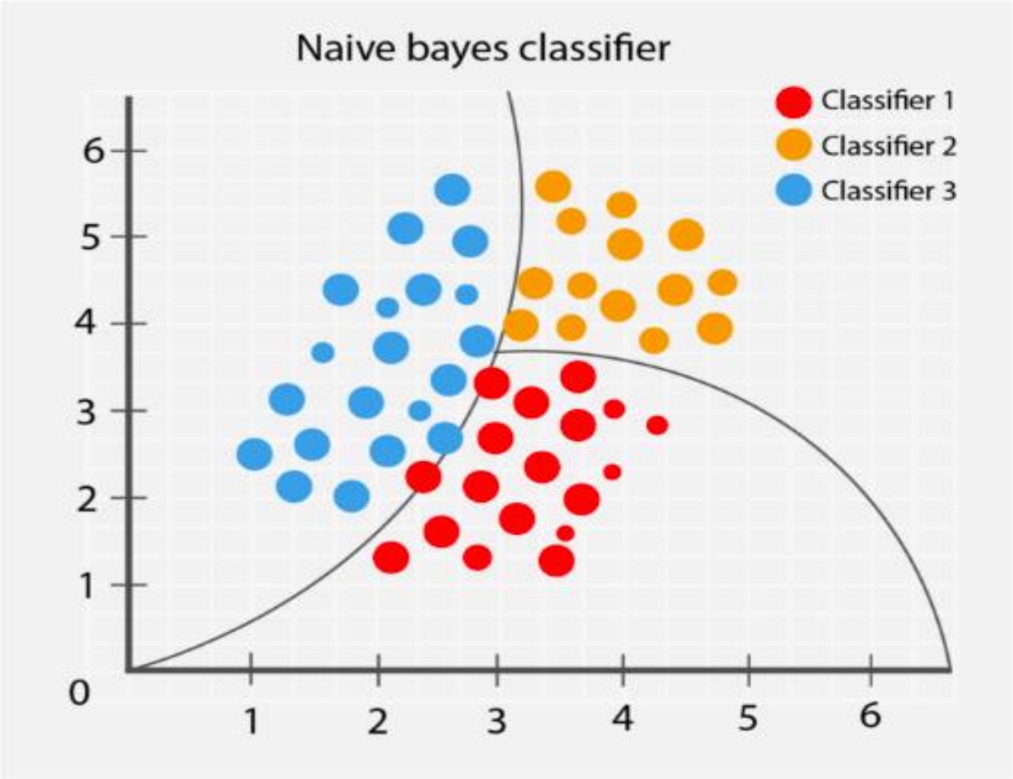


Figure 20: Naive Bayes Classifier

The above image shows the Naive Bayes classifier which is the group of simple probabilistic classifiers based on Bayes's theorem with assumptions between the features.

Since Bayesian networks work on probabilities the formula for the network is given as:

$$\underset{\text{Posterior Probability}}{P(c \mid x)} = \frac{\overset{\text{Likelihood}}{P(x \mid c)}\,\overset{\text{Class Prior Probability}}{P(c)}}{\underset{\text{Predictor Prior Probability}}{P(x)}}$$

$$P(c \mid \mathbf{X}) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

Figure 21: Bayesian Network Formula

Posterior probability can be calculated from the Bayes theorem from the formula mentioned above. The terms included in this formula are explained below:

- $P(c|x)$ = posterior of target class
- P(c) = prior probability of the target class
- P(x) = prior probability of the predictor
- $P(x|c)$ = likelihood of the given class

4.6.2 Support Vector Classifiers

Support Vector Machines (SVM) fall into the category of supervised machine learning algorithms. It can be utilized for both classification and regression problems but mostly it is used for classification problems. In the SVC algorithm, we plot each data item with the value of each feature which becomes the value of a particular coordinate in N-dimensional space where N is the number of features. After plotting this we find the hyper-plane that differentiates the different classes to perform classification [12]. For example, if we have two classes then a graphical representation of SVC based on some rules can be shown in the figure below:

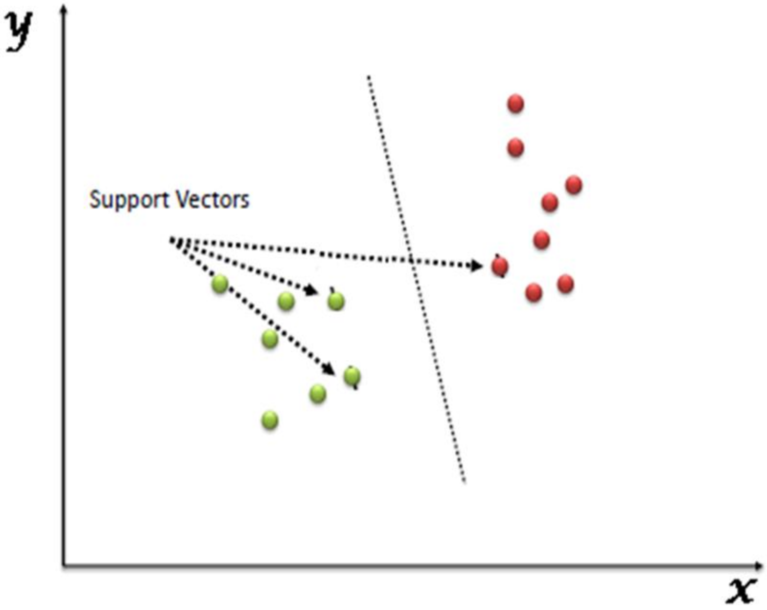


Figure 22: Graphical Representation of SVC

So, from the above discussion and figure above we can see that the dots are Support Vectors which represent coordinates of individual observation and the hyper-plane which is best segregating the two classes is the Support Vector Machine.

In many cases we will get more than one hyper-plane then a question can arise on how to identify which hyper-plane is best? So, the answer to this is we will follow a thumb rule which is “Select that hyper-plane which is best at separating the classes [12].**”** Maybe in some scenarios we find all the hyper-planes best for segregation as shown in the **figure below** then we will calculate the margin between the hyper-plane and nearest data-point and we select the hyper-plane which gives a greater margin [12]. Likewise, there can be more scenarios of selecting the best hyper-plane for classification.

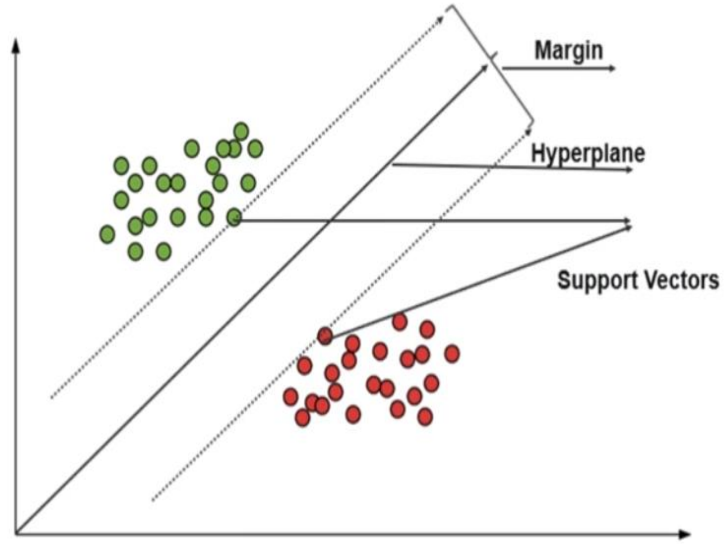


Figure 23: Margin between hyperplane and nearest support vectors

Representation of training examples in SVC is attribute-value pairs which means the model will be trained based on particular attributes (features) with their values annotated with a particular class. These attribute values along with the labeled class in the training example will be first label encoded (with numeric value) and then will be passed to SVM () to train the model.

Support Vector Classifier uses an incremental method to train the model as it uses one training example from Training Data at a time and modifies the current Hypothesis (the possible label which best fits the set of Training Examples).

Support Vector Classifier benefits in a number of ways such as:

- When we have well clear margins of separation between classes it performs really well
- It also performs effectively when the number of samples is greater than the number of dimensions.

Along with these pros it also has some drawbacks which are:

- When we have a large number of datasets it does not perform well because of the expensive training time

In Support Vector Machine probability estimates are calculated using an expensive five-fold cross-validation rather than calculating it directly. These calculations are included in the related SVC method of the Python sci-kit learn library [12].

4.6.3 Random Forest classification

Random Forest Classifier also belongs to supervised machine learning algorithms and is used for Regression and Classification problems same as SVC. It uses a technique called ensemble learning in which many classifiers are combined to provide solutions to complex problems.

Predictions from Random Forest Classifier comes from the predictions of decision trees. It classifies by taking the mean or average of the predictions made by decision trees. This means that to increase the accuracy we simply need to increase the number of trees. Random Forest overcomes the problem of overfitting and increases the precision which is the problem in many other supervised machine learning algorithms.

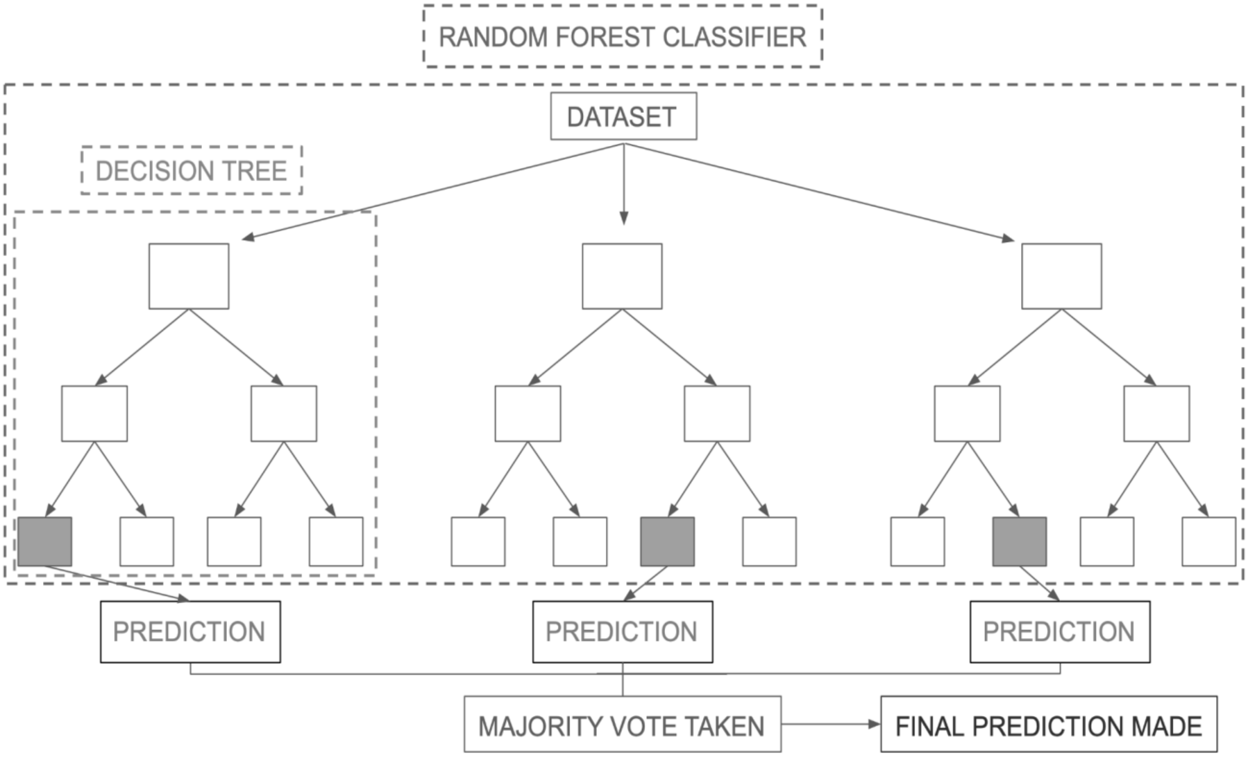


Figure 24: Working of Random Forest Classifier

In Random Forest every decision tree has leaf nodes, decision nodes, and a root node. The lead node of every tree is considered as the final prediction produced by that particular tree. After that, the final prediction is made by taking the majority votes of all trees. This means that the class which is predicted by the majority of the trees will be the final output.

4.7 Deep learning models

These are the neural network architectures that can be single-layered as well as multiple layered. These models learn features directly from the data without any need for manual feature extraction. The deep learning algorithms will be used on the latest tweet feature to extract features from the tweets automatically.

- CNN - Convolutional Neural Network

4.7.1 CNN - Convolutional Neural Network

A Convolutional neural network is a deep learning algorithm that can work on multiple layers and architecture. A Convolutional neural network is mainly used for image processing but it can be used for other applications as well. The application that we are using the CNN for is mainly composed of NLP and classification based on weighted features.

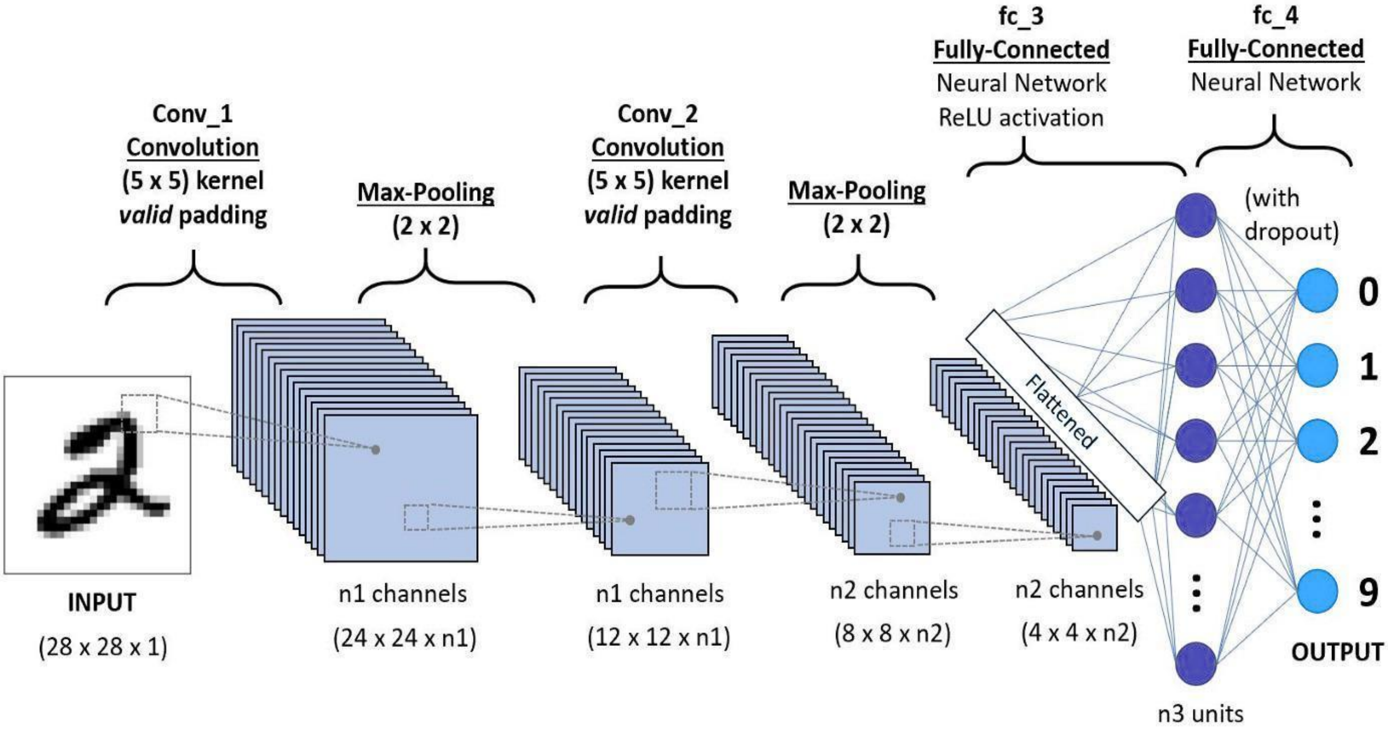


Figure 25: Structure of CNN Model

The model works on the weightage system which means it assigns weights to different features in the targeted dataset and then makes a feature vector that classifies the given models accordingly. CNN has advantages because it requires much less preprocessing on the data than other classical methods. It is a feed-forward neural network that is based on artificial intelligence. While CNN is mainly popular for its usage in image data it is also popular for text classification. The text classification in CNN is a little variant from the image classification in CNN. During a text classification using CNN the model, a result is obtained only when a pattern is detected among the textual data [13]. The current project uses CNN for textual classification. The dataset is obtained from Twitter and the tweets are used for the classification of the profession.

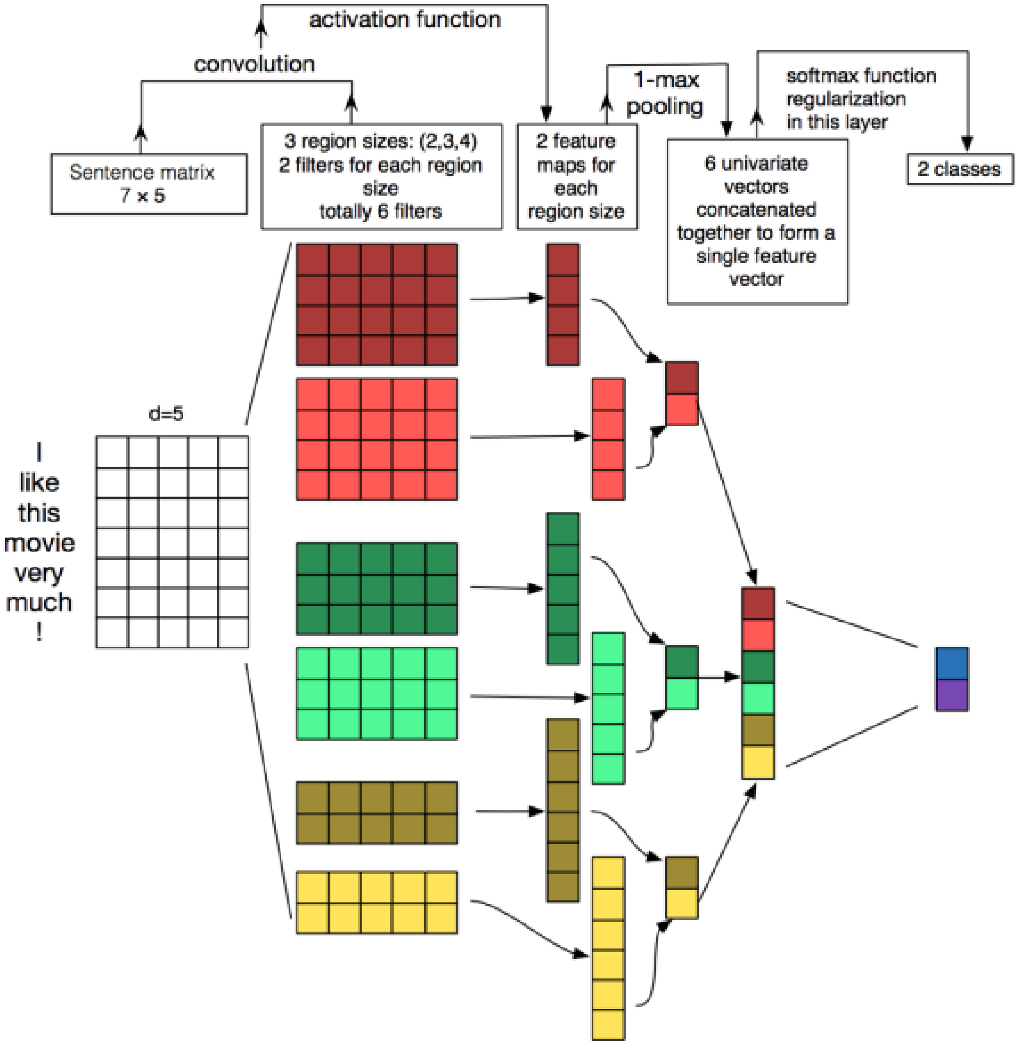

activation function
convolution
1-max pooling
softmax function regularization in this layer
Sentence matrix 7 x 5
3 region sizes: (2,3,4) 2 filters for each region size totally 6 filters
2 feature maps for each region size
6 univariate vectors concatenated together to form a single feature vector
2 classes
d=5
I
like
this
movie
very
much
!

Figure 26: CNN for NLP

The main layers which are involved in the CNN are Conv2D which is the convolutional layer, max pooling layer which selects the maximum form each feature map, dense layer which selects the neurons from all previous layers.

4.8 Training and parameter tuning

These models have been trained while constantly being tuned to produce results with the best accuracy. The models have been made by using the Sci-kit learn library on python. The python library for natural language processing which is Natural Language Processing Toolkit (NLTK) was also used for statistical analysis of English language tweets. Various techniques such as "bag of words" and "TF-IDF" were used on the latest tweets feature to extract meaningful information from the tweets and then reconfigure the latest tweets feature. The trained model was tested and validated to make sure that the accuracy of the model is valid for real-world processes as well. The model then was stored and used as a backend for the website.

The target is to create a machine learning model with decent accuracy which can tell the category of a particular Twitter user. A web application was developed at the end of the project using HTML, CSS, bootstrap, flask, and python. The web application contains an input field that allows the user to insert a profile URL by which the category to which the profile belongs, will be given.

4.9 Evaluation measures

Evaluation of the machine learning model is carried out using

Table 15: Evaluation Measures

Evaluation Measures	Definition	Formula
Accuracy	Accuracy is defined as the proportion of correctly classified test instances	$Accuracy = Correctly\ Classified\ Test\ Instances\ /\ Total\ Number\ of\ Test\ Instances$
Precision	Precision (P) is the proportion of the predicted Positive cases that were correct	$Precision = Correctly\ classified\ positive\ instances\ /\ Total\ Number\ of\ clasiified\ positive\ Instances$
Recall	Recall the proportion of Positive cases that were correctly classified	$Recall = Correctly\ Classified\ positive\ Instances\ /\ Total\ Number\ of\ positive\ Instances$
F1 Score	When we assign the same weights to Precision and Recall i.e., β = 1, the F-measure becomes F1 -measure	$F1\ score = 2 * precision * Recall\ /\ precision + Recall$

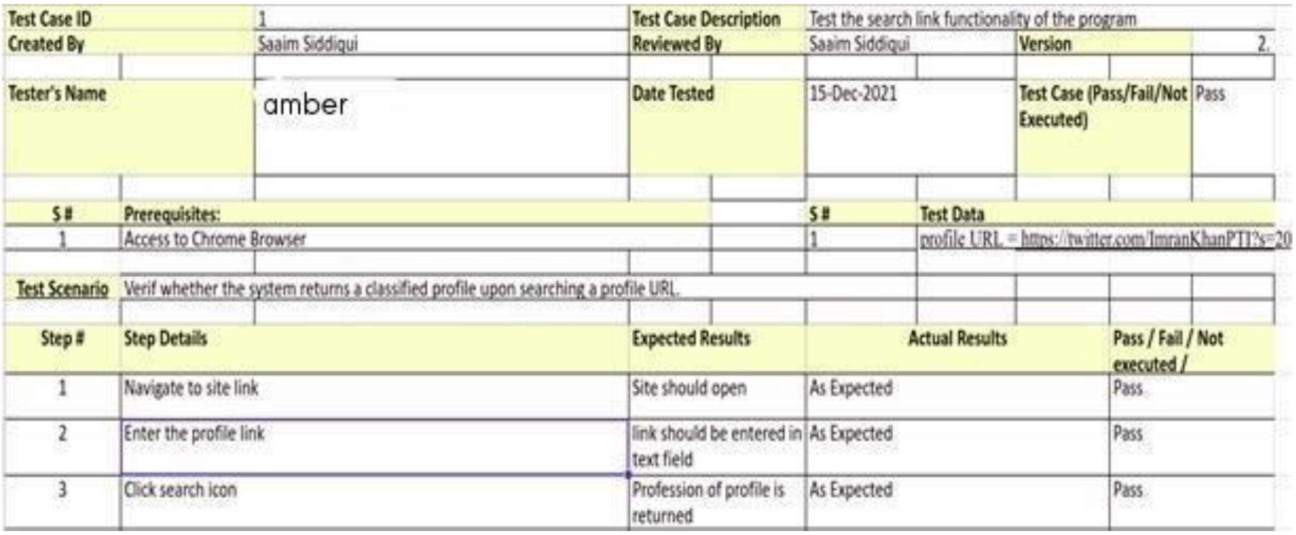
Chapter 5

System testing

5 Chapter 5: System testing

5.1 Test cases

5.1.1 Test Case 1

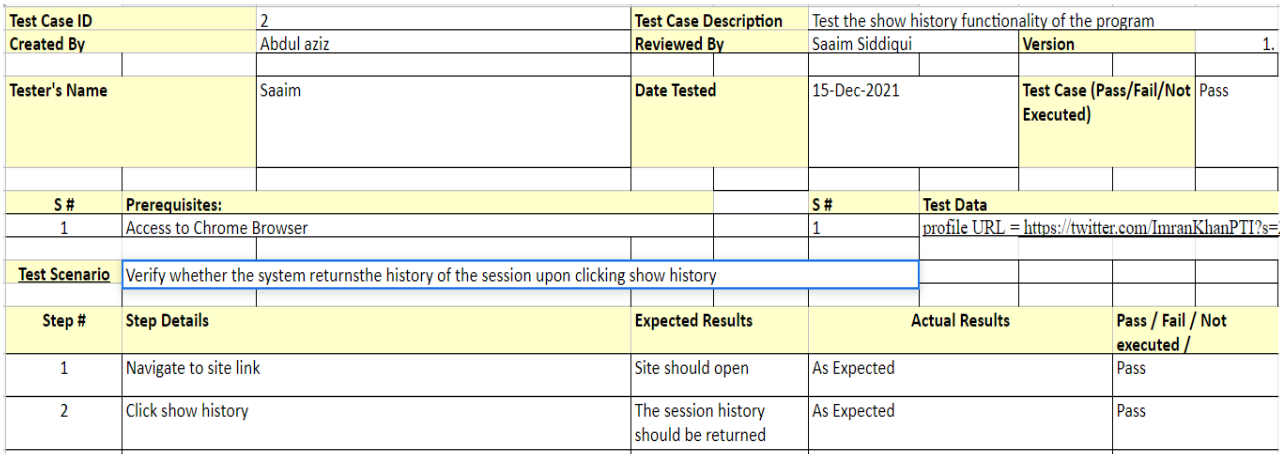


Test Case ID	1	Test Case Description	Test the search link functionality of the program		
Created By	Saaim Siddiqui	Reviewed By	Saaim Siddiqui	Version	2.
Tester's Name	amber	Date Tested	15-Dec-2021	Test Case (Pass/Fail/Not Executed)	Pass
S #	Prerequisites:		S #	Test Data	
1	Access to Chrome Browser		1	profile URL = https://twitter.com/ImranKhanPTI?s=20	
Test Scenario	Verif whether the system returns a classified profile upon searching a profile URL.				

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed /
1	Navigate to site link	Site should open	As Expected	Pass
2	Enter the profile link	link should be entered in text field	As Expected	Pass
3	Click search icon	Profession of profile is returned	As Expected	Pass

Figure 27: Test Case 01

5.1.2 Test Case 2

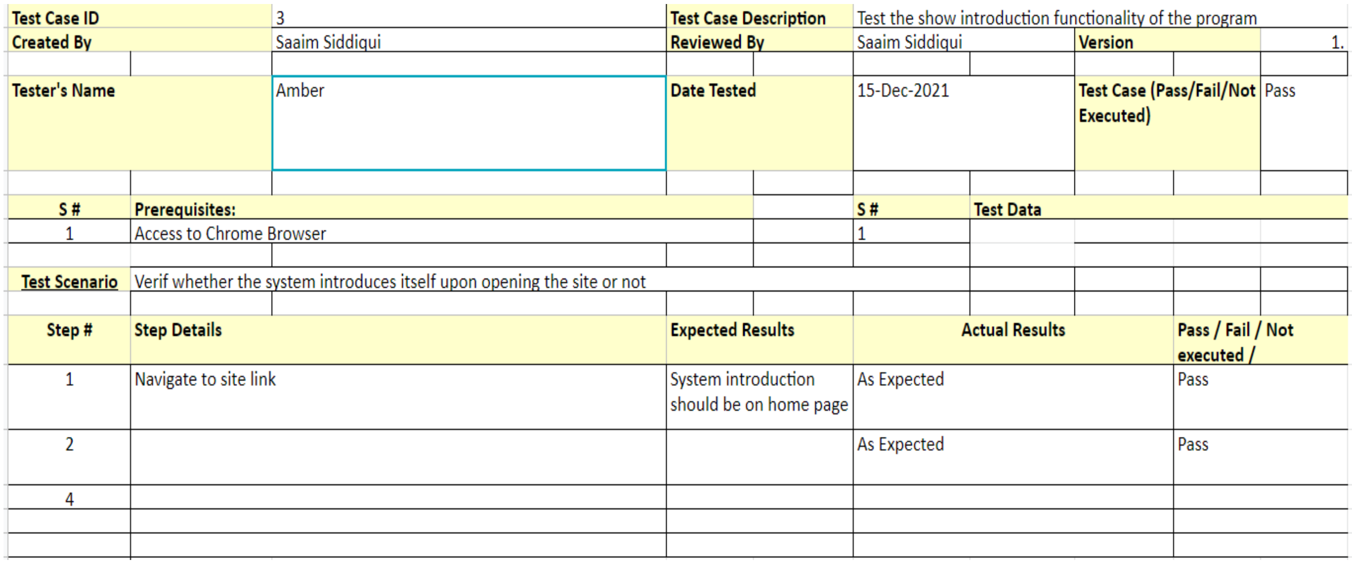


Test Case ID	2	Test Case Description	Test the show history functionality of the program		
Created By	Abdul aziz	Reviewed By	Saaim Siddiqui	Version	1.
Tester's Name	Saaim	Date Tested	15-Dec-2021	Test Case (Pass/Fail/Not Executed)	Pass
S #	Prerequisites:		S #	Test Data	
1	Access to Chrome Browser		1	profile URL = https://twitter.com/ImranKhanPTI?s=	
Test Scenario	Verify whether the system returnsthe history of the session upon clicking show history				

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed /
1	Navigate to site link	Site should open	As Expected	Pass
2	Click show history	The session history should be returned	As Expected	Pass

Figure 28: Test Case 02

5.1.3 Test Case 3



Test Case ID	3	Test Case Description	Test the show introduction functionality of the program		
Created By	Saaim Siddiqui	Reviewed By	Saaim Siddiqui	Version	1.
Tester's Name	Amber	Date Tested	15-Dec-2021	Test Case (Pass/Fail/Not Executed)	Pass

S #	Prerequisites:	S #	Test Data
1	Access to Chrome Browser	1	

Test Scenario	Verif whether the system introduces itself upon opening the site or not

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed /
1	Navigate to site link	System introduction should be on home page	As Expected	Pass
2			As Expected	Pass
4				

Figure 29: Test case 03

Chapter 6

Results and Analysis

6 Chapter 6: Results and Analysis

6.1 Corpus details

The corpus used in the training of models contains a dataset of 6000 entries out of which every category contains roughly 1000 entries. The dataset is made sure that it is balanced so that the training of the model can be achieved properly. The proper figure of each category is given below.

Table 16: corpus details

Category	**Instances**
Politician	985
Actor	989
Singer	980
Sports	1027
Education	979
Content Creator	980

6.2 Features and techniques

Two techniques were used to extract the features from a preprocessed dataset. “Bag of words” and “TF- IDF”. The techniques were applied to the sentences to get stemmed words and then the frequency was fed to the machine learning models to train these models. The number of max features was kept at 50 and 1000 to measure the difference between the accuracy of the models which turned out to be the almost same in both cases.

6.3 Features extracted through bag of words

First, the models were trained on using the features extracted through bag of words. The macro accuracy and various scores of the models are given in the table below.

Table 17: Evaluations Measures of Classical Models Using BOW Features

Model	**Accuracy**	**Precision**	**Recall**	**F1score**
SVM	0.815	0.818	0.814	0.814
Naive Bayes	0.557	0.55	0.553	0.544
Logistic Regression	0.858	0.858	0.857	0.857
Random Forest	0.882	0.881	0.881	0.881

6.4 “TF - IDF” followed by bag of words

After this “TF - IDF” followed by bag of words was applied on the dataset to find stemmed words and then model training was done. This increased the overall accuracy of the models. The accuracy after using this approach was:

Table 18: Evaluation Measures of Classical Models using TF-IDF followed by BOW Features

Model	Accuracy	Precision	Recall	F1score
SVM	0.897	0.898	0.897	0.897
Naive Bayes	0.557	0.55	0.553	0.544
Logistic Regression	0.899	0.898	0.898	0.898
Random Forest	0.868	0.866	0.866	0.865

Deep learning model CNN was also used to compare the accuracy of classical and deep learning models. A CNN model with 2 dense layers and a sigmoid activation function was used. The model was trained on 100 epochs with a batch size of 10. So far the accuracy attained form the model was 17%.

Chapter 7

Conclusion

7 Chapter 7: Conclusion

7.1 Problems faced and lessons learned

So far, the biggest problem faced through this project is the collection of data. Due to the update in twitter terms and conditions, their source code was also updated. This made all the scraping bots inefficient as the classes' names from the source code changed. Twitter's own library was only accessible by people who had proper authentication from Twitter developers. The process of finding a proper library that can help extract the data this project requires was painfully long due to which a lot of project deadlines were shifted forward. In the end, a proper library was found and the scraping of data began. The data needs to be annotated manually for better training of the models. The lesson was learned in this to always test out the libraries and API's before submitting the proposal of a subject so that last-minute problems are not encountered.

The other problems faced with the project were the training of models. The deep learning models tend to take a greater number of resources than the classical models hence proper hardware should be available for the training of the deep learning models. The lesson learned in this case is to always list out the resources that a project might take so that if one does not have resources the expensive models can be ruled out.

7.2 Project summary

Social media has connected people the way that they never were, people make new friends, new connections, and a whole virtual social life. Twitter is among the most popular social platforms which allow people to interact with others having the same line of interest and make a virtual community. There are thousands of people interacting on the same platform belonging to different fields such as business, technology, health, arts, literature, etc. But the problem arises when all these users' profiles are not sorted into some categories and they all are just present in one space without belonging to one specific class. Twitter doesn't provide any additional information that can distinguish between the profiles of the people belonging to some specific class of interests.

The idea of working on this project is to develop an automatic Twitter profile tagging system that will categorize the profiles into the class they belong to. The profiles have been categorized into six categories like Politician, Actor, Sports, Content Creator, Educational, and Singer. For this purpose, a dataset has been collected containing user twitter handles belonging to different lines of interest using a bot that has been created on a Python library called Selenium. After the collection of Twitter handles, a bot was created on python using the GitHub library snscrape which we used to scrape the attributes such as user tweets, description, number of followers etc. After that, some features have been extracted from the collected dataset of Twitter profiles like the number of followings, the number of followers, retweets, tweets, likes count, etc. After collecting all the required information, data processing techniques have been applied like "TF - IDF" and "Bag of Words" by which the meaningful information from the tweets has been extracted.

After the whole process of collection of data, we used this data to train our Classical Machine Learning and Deep Learning models such as Naive Bayes, Support Vector Classifier, Random Forest, Logistic Regression, and CNN. This system will now tag the profiles into broader categories by taking the Twitter Handles of the Twitter profiles. A web page has been made to make sure easier access to the classification system is available to the user.

7.3 Future work

The project can be further expanded by using sentences instead of just words to identify the profession of the user. The current project uses techniques like “bag of words” or “TF-IDF” to identify the profession of a user through their tweets and other profile information. The use of deep learning models such as LSTM can also make the project as the model considers sentences instead of just words before classifying. A user’s overall social network which means the people he is closely related to or his frequently talked to on social media can also reveal what kind of profession a user has. The project can be expanded by constantly retraining the model whenever a user searches a profile URL this way the model keeps on learning and the accuracy keeps on getting better. The classes in this project are kept small so that the project can be achieved in a particular time frame. The classes of the project can be further expanded to make a model that can classify from over 200 professions. The model would require precise training and execution plans to come to reality.

8 References

[1] [Online]. Available: https://backlinko.com/social-media-users.

[2] [Online]. Available: https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/.

[3] [Online]. Available: https://www.statista.com/statistics/242606/number-of-active-twitter-users-in-selected-countries/.

[4] "Counting characters | Docs | Twitter Developer Platform," [Online]. Available: https://developer.twitter.com/en/docs/counting-characters.

[5] M. Cross, Bloggerati, Twitterati: How Blogs and Twitter are Transforming Popular Culture: How Blogs and Twitter are Transforming Popular Culture, ABC-CLIO, 2011.

[6] "Twitter Developer Platform," [Online]. Available: https://developer.twitter.com/en/docs/twitter-api/v1/tweets/search/api-reference/get-search-tweets.

[7] M. Pennacchiotti and A.-M. Popescu, "A machine learning approach to twitter user classification," in *Fifth international AAAI conference on weblogs and social media*, 2011.

[8] R. Basti, S. Jamoussi, A. Charfi and A. B. Hamadou, "Arabic Twitter User Profiling: Application to Cyber-security.," in *WEBIST*, 2019, pp. 110--117.

[9] S. Gurajala, J. S. White, B. Hudson, B. R. Voter and J. N. Matthews, "Profile characteristics of fake Twitter accounts," *Big Data \& Society*, vol. 3.

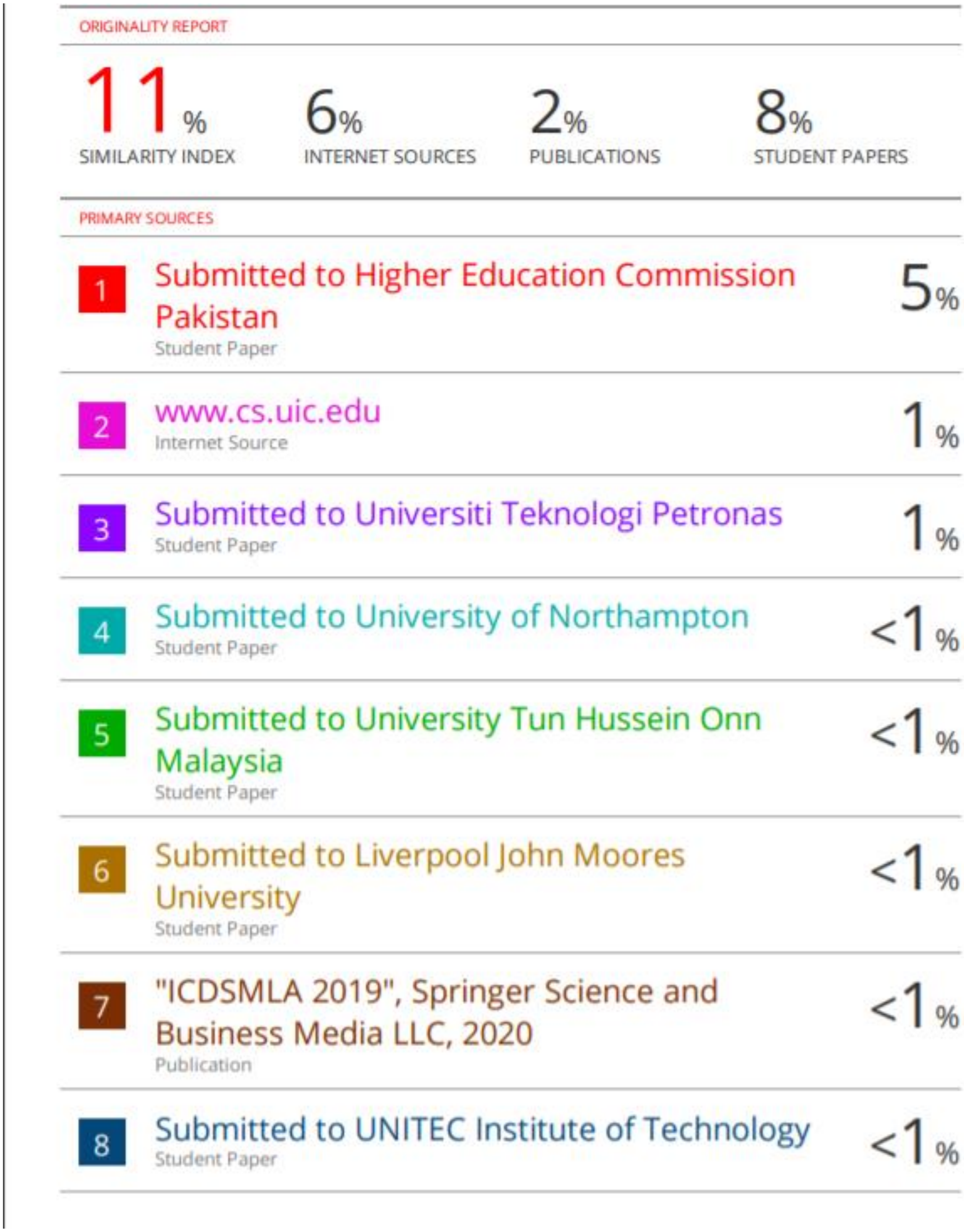
[10] J. S. Alowibdi, U. A. Buy and P. Yu, "Empirical Evaluation of Profile Characteristics for Gender Classification on Twitter," in *2013 12th International Conference on Machine Learning and Applications*, vol. 1, 2013, pp. 365-369.

[11] P. Galán-García, J. G. d. l. Puerta, C. L. Gómez, I. Santos and P. G. Bringas, "Supervised machine learning for the detection of troll profiles in twitter social network: application to a real case of cyberbullying," *Logic Journal of the IGPL*, vol. 24, pp. 42-53, 10 2015.

[12] S. Ray, S. Bansal, A. Gupta, D. Gupta and F. Shaikh, "Understanding Support Vector Machine algorithm from examples (along with code)," *Analytics Vidhya*, vol. 13, p. 19, 2017.

[13] Britz and Denny, "Understanding convolutional neural networks for NLP," Vols. http://www. wildml. com/2015/11/understanding-convolutional-neuralnetworks-for-nlp, 2015.

9 Appendix A

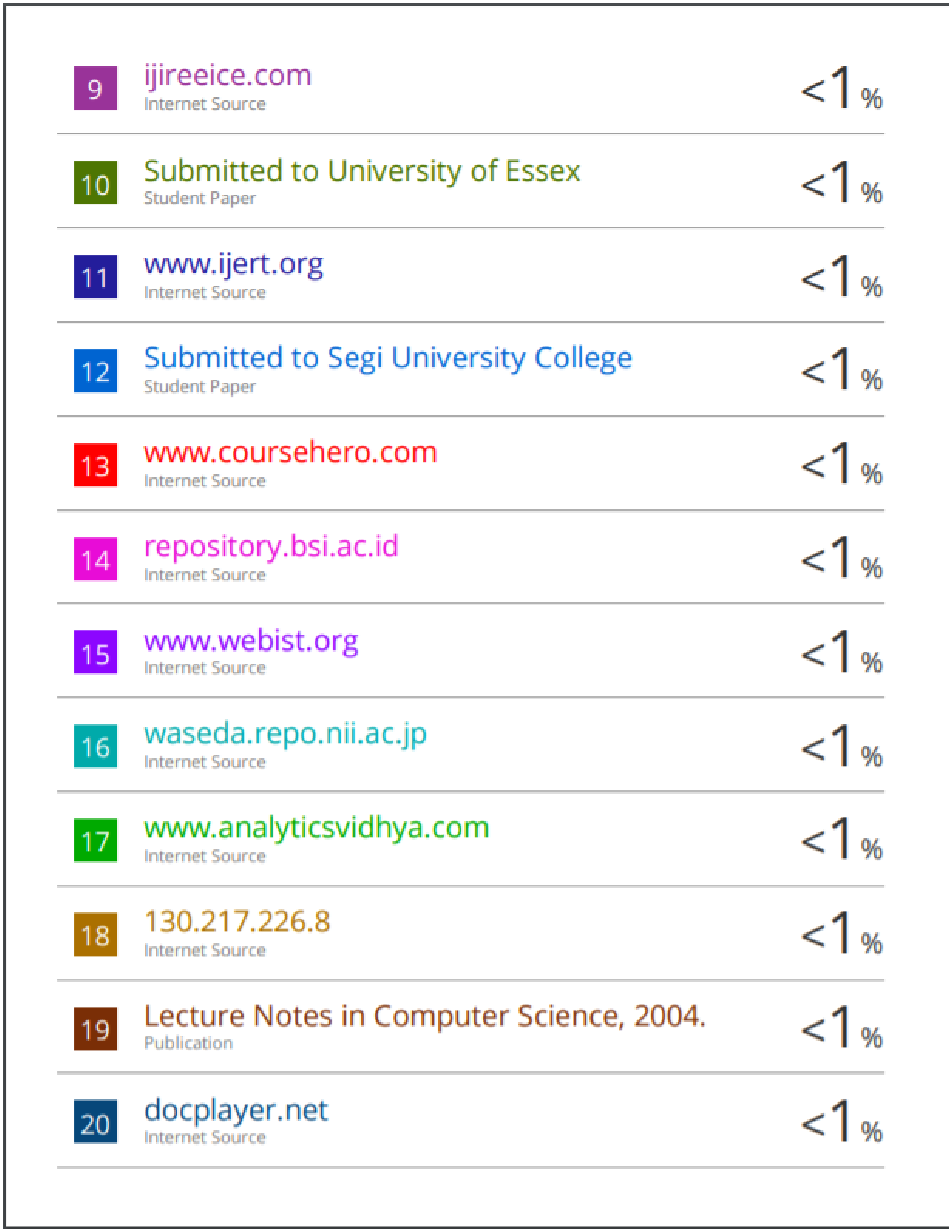


ORIGINALITY REPORT

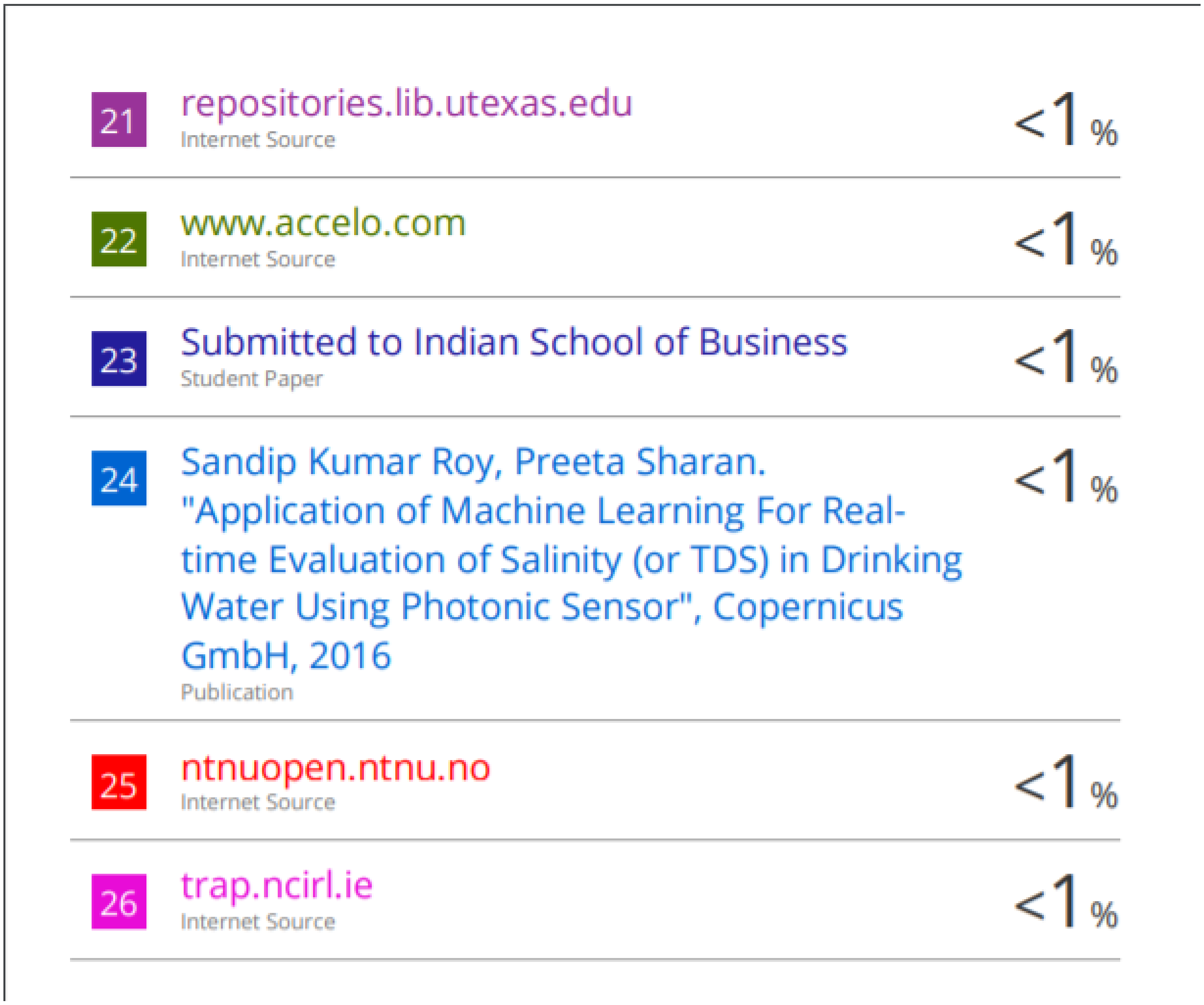
11%	6%	2%	8%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Higher Education Commission Pakistan Student Paper	5%
2	www.cs.uic.edu Internet Source	1%
3	Submitted to Universiti Teknologi Petronas Student Paper	1%
4	Submitted to University of Northampton Student Paper	<1%
5	Submitted to University Tun Hussein Onn Malaysia Student Paper	<1%
6	Submitted to Liverpool John Moores University Student Paper	<1%
7	"ICDSMLA 2019", Springer Science and Business Media LLC, 2020 Publication	<1%
8	Submitted to UNITEC Institute of Technology Student Paper	<1%



9	ijireeice.com Internet Source	<1 %
10	Submitted to University of Essex Student Paper	<1 %
11	www.ijert.org Internet Source	<1 %
12	Submitted to Segi University College Student Paper	<1 %
13	www.coursehero.com Internet Source	<1 %
14	repository.bsi.ac.id Internet Source	<1 %
15	www.webist.org Internet Source	<1 %
16	waseda.repo.nii.ac.jp Internet Source	<1 %
17	www.analyticsvidhya.com Internet Source	<1 %
18	130.217.226.8 Internet Source	<1 %
19	Lecture Notes in Computer Science, 2004. Publication	<1 %
20	docplayer.net Internet Source	<1 %



21	repositories.lib.utexas.edu Internet Source	<1 %
22	www.accelo.com Internet Source	<1 %
23	Submitted to Indian School of Business Student Paper	<1 %
24	Sandip Kumar Roy, Preeta Sharan. "Application of Machine Learning For Real-time Evaluation of Salinity (or TDS) in Drinking Water Using Photonic Sensor", Copernicus GmbH, 2016 Publication	<1 %
25	ntnuopen.ntnu.no Internet Source	<1 %
26	trap.ncirl.ie Internet Source	<1 %