



Government College Of Engineering, Chandrapur

Twibot-Spotter

Session: 2022- 23

Computer Science and Engineering

Guide

Prof. Shailendra

Shende

- Pawankumar Tummawar -69

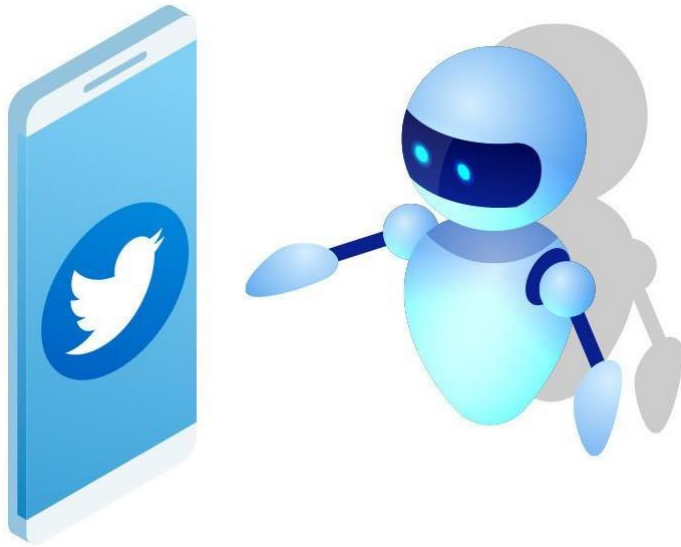
- Ritika Telang -65

- Vishal Dhandare -73

Content

- 1) Introduction
- 2) Problem definition
- 3) Abstract
- 4) Objectives
- 5) Literature survey
- 6) Plan of action
- 7) The role of ML
- 8) Why not KNN?
- 9) Why not Decision tree?
- 10) Why we used Random forest?
- 11) How Random forest works?
- 12) Software used
- 13) References

Problem Definition



- In today's date, approximately 80% of twitter accounts are bots.
- These bot accounts are responsible for
 - a. Spreading fake news.
 - b. Spamming
 - c. Increasing number of fake followers
 - d. Re – tweets

Abstract

With the proliferation of social media platforms, detecting and mitigating the impact of bots on user behavior has become a crucial issue. Twitter, being one of the most popular platforms, has been heavily affected by bots that can spread misinformation, manipulate opinions, and create a toxic environment. In this presentation, we will explore various techniques and tools for detecting Twitter bots. We will discuss how machine learning algorithms can be used to **distinguish between genuine and fake accounts** based on their **behavior patterns**, such as **posting frequency, content, and network characteristics**. Finally, we will discuss how these techniques can be applied in practice to mitigate the impact of Twitter bots and enhance the user experience.

Objective

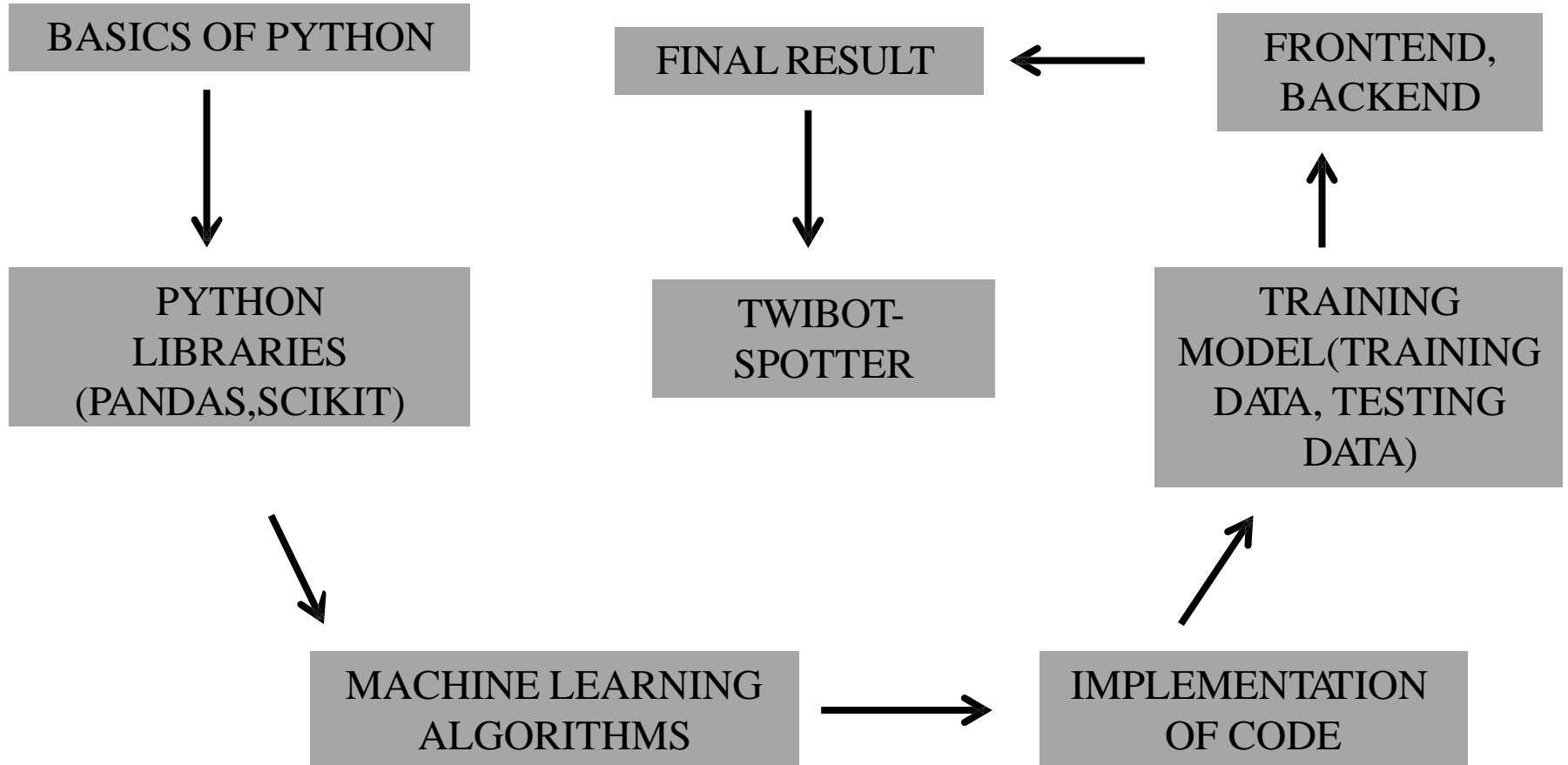
Classify a twitter user as a human or a bot by observing the difference among them based on parameters such as:-

- Followers Count
- Following Count
- Verified Accounts
- Following to Followers Ratio
- Account details

Literature survey

Sr. no	Title of the paper	Author	Major observations/ findings
1.	Twitter Bot Detection with Reduced Feature Set	Jefferson Viana Fonseca Abreu, Celia Ghedini Ralha	Random Forest algo was found to be the Best Algorithm
2.	A Comprehensive Twitter Bot Detection Benchmark	Shangbin Feng, Herun Wan, Ningnan Wang Jundong Li, Minnan Luo	The RF algorithm was found to be giving the best Result
3.	Detecting malicious activity in Twitter using deep learning techniques	Loukas Llias, Loanna Roussaki	Random Forest overtook the other algo's such as (knn , Decision Tree) in terms of accuracy

Plan of action



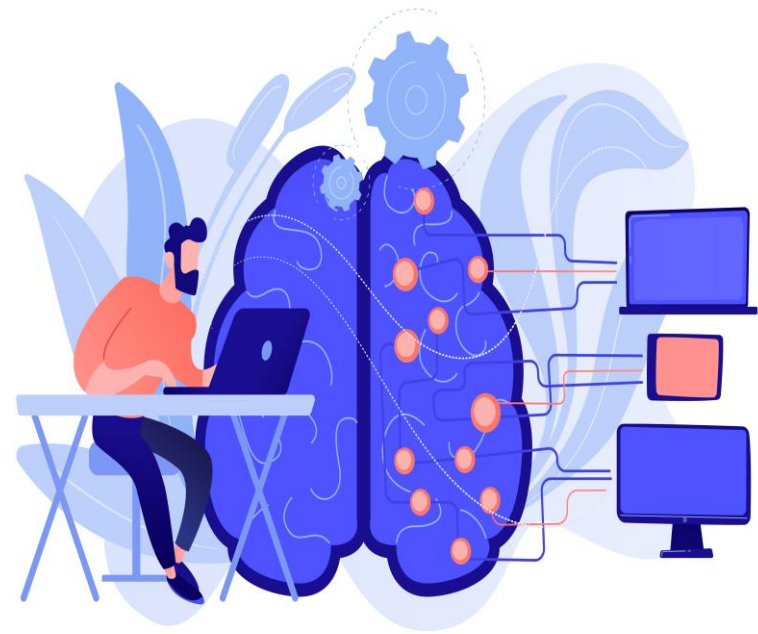
The role of ML



- ML can detect Twitter bots by analyzing behavior patterns such as tweet frequency, timing, and content similarity.
- Suspicious account activity, like high-volume tweeting or interactions with known bot accounts, can also be identified.
- ML helps develop models to distinguish real users from bots and reduce harmful content on Twitter.

How Machine learns?

- Machine learning uses algorithms to analyze data.
- The algorithms look for patterns to distinguish between human users and bots.
- The model is trained on labeled data and tested on new data to improve accuracy over time.
- The model can then be tested on new, unlabeled data to evaluate its accuracy and refine the algorithm.



Why not KNN?

```
training data on KNeighborsClassifier model:
```

	precision	recall	f1-score	support
0	0.67	0.02	0.04	290
1	0.48	0.99	0.65	270
accuracy			0.49	560
macro avg	0.58	0.50	0.35	560
weighted avg	0.58	0.49	0.33	560

```
Average Accuracy: 47.69%
```

```
PS C:\Users\Pawan\Desktop\real\sample> █
```

Why Not Decision Tree?

training data on Decision Tree model:

	precision	recall	f1-score	support
0	0.67	0.69	0.68	290
1	0.66	0.64	0.65	270
accuracy			0.66	560
macro avg	0.66	0.66	0.66	560
weighted avg	0.66	0.66	0.66	560

Average Accuracy: 68.43%

PS C:\Users\Pawan\Desktop\real\sample>

Why we used Random Forest?

```
training data on Random Forest model:
```

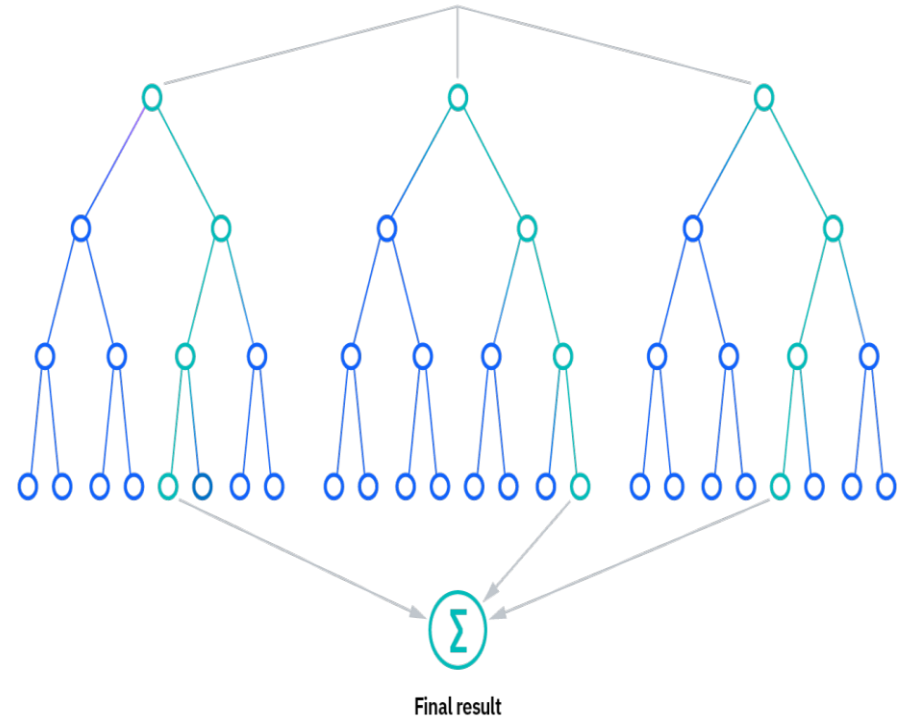
	precision	recall	f1-score	support
0	0.70	0.72	0.71	290
1	0.69	0.67	0.68	270
accuracy			0.70	560
macro avg	0.70	0.70	0.70	560
weighted avg	0.70	0.70	0.70	560

Average Accuracy: 72.86%

PS C:\Users\Pawan\Desktop\real\sample> █

How Random forest works?

- Random Forest is a machine learning algorithm that uses decision trees to classify data.
- It creates multiple decision trees by randomly selecting features and data samples for each tree.
- Each decision tree independently classifies the data, and the final classification is determined by combining the results of all the trees.



Software Used:-

- Python
- Scikit
- Python libraries like pandas
- ML algorithms(knn,Decision Tree,Random Forest)
- HTML, CSS, Javascript (frontend)
- Flask (backend)

References

- Custom classification algorithm to sense the bots vs human on social media space like <https://github.com/topics/twitter-bot-detection>
- <https://github.com/jubins/MachineLearning-Detecting-Twitter-Bots>
- <https://www.ijsr.net/archive/v8i7/ART20199245.pdf> (International Journal of Science and Research (IJSR))
- <https://www.kaggle.com/>