A

Major Project

On

SPAMMER DETECTION AND FAKE USER IDENTIFICATION ON SOCIAL NETWORKS

(Submitted in partial fulfillment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

CHOUHAN BHEEM SINGH (197R1A05K2)

DEVENDER DONADULA (197R1A05K5)

MUCHARLA VAMSHIDHAR REDDY (197R1A05N1)

Under the Guidance of

K. RANJITH REDDY

(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR TECHNICAL CAMPUS

UGC AUTONOMOUS

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi) Recognized Under Section 2(f) & 12(B) of the UGCAct.1956, Kandlakoya (V), Medchal Road, Hyderabad-501401.

2019-2023

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project entitled **"SPAMMER DETECTION AND FAKE USER IDENTIFICATION ON SOCIAL NETWORKS"** being submitted by **M . VAMSHIDHAR REDDY (197R1A05N1)** , **D . DEVENDER (197R1A05K5)** and **CH . BHEEM SINGH (197R1A05K2)** in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering to the CMR Technical Campus, is a record of bonafide work carried out by them under our guidance and supervision during the year 2022-2023.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

K. Ranjith Reddy
(Assistant Professor)
INTERNAL GUIDE

Dr. A. Raji Reddy
DIRECTOR

Dr. K. Srujan Raju
HOD

EXTERNAL EXAMINER

Submitted for viva voice Examination held on _____

ACKNOWLEDGEMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We take this opportunity to express my profound gratitude and deep regard to my guide **K. Ranjith Reddy,** Assistant Professor for his exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by him shall carry us a long way in the journey of life on which we are about to embark.

We also take this opportunity to express a deep sense of gratitude to the Project Review Committee (PRC) **Dr. Punyaban Patel, Ms. Shilpa, Dr. L . Subha Mastan Rao & J. Narasimha Rao** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to **Dr. K. Srujan Raju,** Head, Department of Computer Science and Engineering for providing encouragement and support for completing this project successfully.

We are obliged to **Dr. A. Raji Reddy,** Director for being cooperative throughout the course of this project. We also express our sincere gratitude to Sri. **Ch. Gopal Reddy,** Chairman for providing excellent infrastructure and a nice atmosphere throughout the course of this project.

The guidance and support received from all the members of **CMR Technical Campus** who contributed to the completion of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project.

CHOUHAN BHEEM SINGH (197R1A05K2)
DEVENDER DONADULA (197R1A05K5)
MUCHARLA VAMSHIDHAR REDDY (197R1A05N1)

ABSTRACT

Social networking sites engage millions of users around the world. The users' interactions with these social sites, such as Twitter and Facebook have a tremendous impact and occasionally undesirable repercussions for daily life. The prominent social networking sites have turned into a target platform for the spammers to disperse a huge amount of irrelevant and deleterious information. Twitter, for example, has become one of the most extravagantly used platforms of all times and therefore allows an unreasonable amount of spam. Fake users send undesired tweets to users to promote services or websites that not only affect legitimate users but also disrupt resource consumption. Moreover, the possibility of expanding invalid information to users through fake identities has increased that results in the unrolling of harmful content. Recently, the detection of spammers and identification of fake users on Twitter has become a common area of research in contemporary online social Networks (OSNs). In this project, we perform a review of techniques used for detecting spammers on Twitter. Moreover, a taxonomy of the Twitter spam detection approaches is presented that classifies the techniques based on their ability to detect: (i) fake content, (ii) spam based on URL, (iii) spam in trending topics, and (iv) fake users. The presented techniques are also compared based on various features, such as user features, content features, graph features, structure features, and time features. We are hopeful that this project will be a useful resource for researchers to find the highlights of recent developments in Twitter spam detection on a single platform.

LIST OF SCREENSHOTS

TABLE OF CONTENTS PAGE.NO

1. INTRODUCTION

1. INTRODUCTION

1.1 PROJECT SCOPE

We perform a review of techniques used for detecting spammers on Twitter. Moreover, a taxonomy of the Twitter spam detection approaches is presented that classifies the techniques based on their ability to detect: (i) fake content, (ii) spam based on URL, (iii) spam in trending topics, and (iv) fake users. The presented techniques are also compared based on various features, such as user features, content features, graph features, structure features, and time features. We are hopeful that this project will be a useful resource for Twitter spam detection on a single platform. By incorporating machine learning algorithms and other advanced techniques, the project can also propose new approaches for spam detection.

Furthermore, the project can concentrate on improving the accuracy and efficiency of spam detection techniques by taking into account linguistic, behavioral, and network-based features. The project, for example, can examine the linguistic characteristics of spam content, such as the presence of spam keywords, misspellings, and non-standard grammar. It is recommended that a large dataset of spam and non-spam content from Twitter be collected and labelled to ensure the project's effectiveness. The dataset can be used to train and test the performance of different spam detection techniques. The scope of the project for detecting spammers on Twitter can be broadened by considering different types of spam, proposing new techniques, and enhancing the accuracy and efficiency.

1.2 PROJECT PURPOSE

This project has been developed for the main purpose of detecting the spam and fraud accounts on social media platform Twitter. Fake users send undesired tweets to users to promote services or websites that not only affect legitimate users but also disrupt resource consumption. This project is helpful for detecting spammers on Twitter is to develop an effective spam detection system that can identify different types of spam from Twitter feeds. The primary objective of the project is to improve the quality of the Twitter user experience by reducing the noise and clutter caused by spam content. The project aims to achieve this objective by reviewing the existing tweets dataset on spam detection on Twitter, critically analyzing the strengths and weaknesses of different techniques, and proposing new approaches for spam detection. The project will also focus on enhancing the accuracy and efficiency of spam detection .

The project can provide a valuable resource for Twitter users by developing an effective spam detection system for Twitter. The system can assist Twitter users in identifying and preventing spam content, enhancing the accuracy of their feeds, and improving their overall experience. Furthermore, the project has the potential to contribute to broader research on spam detection and prevention in social media. The project can provide insights into the effectiveness of various spam detection techniques and can assist in the development of new and more sophisticated spam detection approaches. To conclude, the purpose of a Twitter spam detection project is to improve the quality of the Twitter user experience by developing an effective spam detection system. The project intends to accomplish this prime objective by reviewing , proposing new approaches, and enhancing the accuracy and efficiency of spam detection techniques. The project can also contribute to broader research on spam detection and prevention in social media.

1.3 PROJECT FEATURES

This is a application in which we can detect the spam and fake users on social media platform Twitter. We have to provide the application with tweets dataset in the JavaScript Object Notation format(JSON) .And we use Naïve Bayers algorithm first on the dataset to classify the data into Spam, fake content ,fake URL and fake user identification and then we use Random Forest datamining algorithm on the dataset and as the result we will get the output as form of graph. Here are some features of the project

- Data Input: The ability to upload a tweets dataset in JSON format to the application for analysis.
- Naïve Bayes Algorithm: The application should be able to use the Naïve Bayes algorithm to classify the tweets in the dataset into different categories, such as spam, fake content, fake URL, and fake user identification.
- Random Forest Algorithm: The application should be able to use the Random Forest data mining algorithm on the dataset to generate a graph showing the classification results.
- Graph Output: The output of the Random Forest algorithm should be displayed in the form of a graph, which can help users to visualize the results and identify patterns in the data.
- User Interface: The application should have an intuitive and user-friendly interface that allows users to upload their datasets, view the classification results, and customize the settings of the algorithms.

By incorporating these features into the application, users can identify and filter out spam and fake users on Twitter with greater accuracy and efficiency, helping to improve the overall quality of content on the platform.

2. SYSTEM ANALYSIS

2. SYSTEM ANALYSIS

SYSTEM ANALYSIS

System Analysis is the important phase in the system development process. The System is studied to the minute details and analyzed. The system analyst plays an important role of an interrogator and dwells deep into the working of the present system. In analysis, a detailed study of these operations performed by the system and their relationships within and outside the system is done. A key question considered here is, “what must be done to solve the problem?” The system is viewed as a whole and the inputs to the system are identified. Once analysis is completed the analyst hasa firm understanding of what is to be done.

2.1 PROBLEM DEFINITION

The problem of fake users and spammers on Twitter has become a significant challenge for the platform's users and administrators alike. The presence of these fake users not only affects legitimate users by inundating them with undesired tweets and advertisements but also disrupts resource consumption on the platform. Moreover, fake users have been known to spread misinformation and harmful content through these fake identities, posing a significant risk to the platform's user base. Due to the increasing complexity of these fake identities, detecting spammers and identifying fake users on Twitter has become a common area of research in contemporary online social networks (OSNs) . To address this problem, there is a need to develop an effective system that can accurately identify and filter out fake users and spammers on Twitter. Such a system would be instrumental in improving the overall quality of content on the platform and protecting legitimate users from unwanted tweets and harmful content. However, developing such a system is not without its challenges. The system would need to be scalable to handle the large volume of data generated by the platform and be accurate enough to distinguish between real and fake users. Additionally, the system would need to be able to adapt to new and evolving methods used by spammers and fake users on the platform.

2.2 EXISTING SYSTEM

The current system for detecting fake profiles and spammers on online social networks such as Twitter is centered on machine learning algorithms such as decision trees, logistic regression, and support vector machines. Despite these efforts, manually detecting fake and spam accounts on social media platforms remains a challenging process that necessarily involves significant human effort and time. Moreover, the increased usage of fake profiles and spammers has considered existing detection methods unsatisfactory, possibly requiring the development of better and more advanced methods.

While the use of machine learning algorithms is a step in the right direction, the existing system needs improvement to overcome the limitations of these algorithms. The current system's reliance on traditional machine learning algorithms can make it difficult to identify new and evolving methods used by spammers and fake profiles. As a result, there is a need for a more sophisticated and adaptable system that can effectively detect fake profiles and spammers on social media platforms like Twitter.

2.2.1 LIMITATIONS OF EXISTING SYSTEM

- Reliance on traditional machine learning algorithms: The current system relies heavily on decision trees, logistics regression, and support vector machine algorithms. These algorithms may not be effective in detecting new and evolving methods used by spammers and fake profiles.
- Limited ability to handle large datasets: As social media platforms like Twitter generate vast amounts of data, the existing system may struggle to handle large datasets, making it less efficient and effective.
- Lack of adaptability: The current system may not be adaptable to changing trends and evolving techniques used by spammers and fake profiles, making it challenging to keep up with the latest detection methods.
- Inability to detect subtle patterns: The existing system may struggle to identify subtle patterns that spammers and fake profiles use to evade detection.

Limitations of existing algorithms

- Decision Tree:
  Less effective in predicting the outcome of a continuous variable: Decision trees are best for categorical variables because they divide the data according to a set of rules. They may not perform as well for continuous variables, however, because the splitting rules may be unable to capture the complexities of the data.
- Logistic Regression:
  May not be accurate if the sample size is too small: Logistic regression requires a sufficient number of samples to make accurate predictions. If the sample size is too small, the model may not be able to capture the true underlying patterns in the data, leading to inaccurate results.
- Support Vector Machine:
  Does not execute very well when the data set has more noise, i.e. target classes are overlapping: SVMs work by creating a decision boundary that separates the data into different classes. However, if the classes are overlapping, the decision boundary may not be clear, leading to inaccurate predictions.

2.3 PROPOSED SYSTEM

The proposed system aims to address the growing concern of spam and fake users on Twitter. The system will utilize a Twitter dataset and four different techniques namely Fake Content, Spam URL Detection, Spam Trending Topic, and Fake User Identification, to identify whether a tweet is normal or spam, and whether a user account is fake or genuine. Each of these techniques will be implemented using a specific Random Forest classifier, which will classify the tweets and user accounts into either spam or non-spam, and fake or non-fake.

The system will use the Naive Bayes algorithm as the first line of defence in classifying the tweets and user accounts. This algorithm is effective in identifying the characteristics of the data and classifying it accordingly. The Random Forest algorithm will then be used to classify the tweets and user accounts into different categories based

on their features, such as user features, content features. The output of the system will be in the form of a graph, which will show the number of spam and non-spam tweets, and fake and non-fake user accounts.

The proposed system will offer a number of advantages over existing systems. It will be faster, more accurate, and require less human effort than manual detection methods. It will also be able to handle large datasets, which would be impractical to manage manually. The system will provide a high level of accuracy in detecting spam tweets and fake user accounts, thus enhancing the security and reliability of Twitter as a social network platform.

2.3.1 ADVANTAGES OF THE PROPOSED SYSTEM

- The system uses a combination of four different techniques to identify spam tweets and fake user accounts on Twitter, making it more comprehensive and accurate compared to the existing system that only uses three algorithms.
- The use of Random Forest classifier for each technique enhances the accuracy and efficiency of the system, which is not available in the existing system.
- The proposed system is highly automated and requires minimal human intervention, saving time and cost for both users and service providers.
- The output is generated as graph for easy understanding of the viewer.

2.4 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and a business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. Three key considerations involved in the feasibility analysis:

- Economic Feasibility
- Technical Feasibility
- Social Feasibility

2.4.1 ECONOMIC FEASIBILITY

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on a project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Also all the resources are already available, it give an indication that the system is economically possible for development.

2.4.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

2.4.3 SOCIAL FEASIBILITY

This includes the following questions:

- Is there sufficient support for the users?
- Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible.

2.5 HARDWARE & SOFTWARE REQUIREMENTS

2.5.1 HARDWARE REQUIREMENTS:

Hardware interfaces specify the logical characteristics of each interface between the software product and the hardware components of the system. The following are some hardware requirements.

- Processor : Intel Core i3 or higher
- Hard Disk : 4 GB or higher
- Memory : 5GB and higher
- Internet Connection : Broadband or higher
- GPU : minimum of 2GB - VRAM

2.5.2 SOFTWARE REQUIREMENTS:

Software Requirements specifies the logical characteristics of each interface and software components of the system.

The following are some software requirements.

- Operating Systems : Windows 10 or higher / Ubuntu 16.04 or higher
- Programming Language : Python 3.7 or higher
- Python IDE : Anaconda, PyCharm or Spyder
- Libraries : NumPy, Pandas, Scikit-learn, Matplotlib
- Twitter dataset : JavaScript Object Notation (JSON) format

3. ARCHITECTURE

3. ARCHITECTURE

3.1 PROJECT ARCHITECTURE

This project architecture shows the procedure how text is converted into code and speech is converted in to text

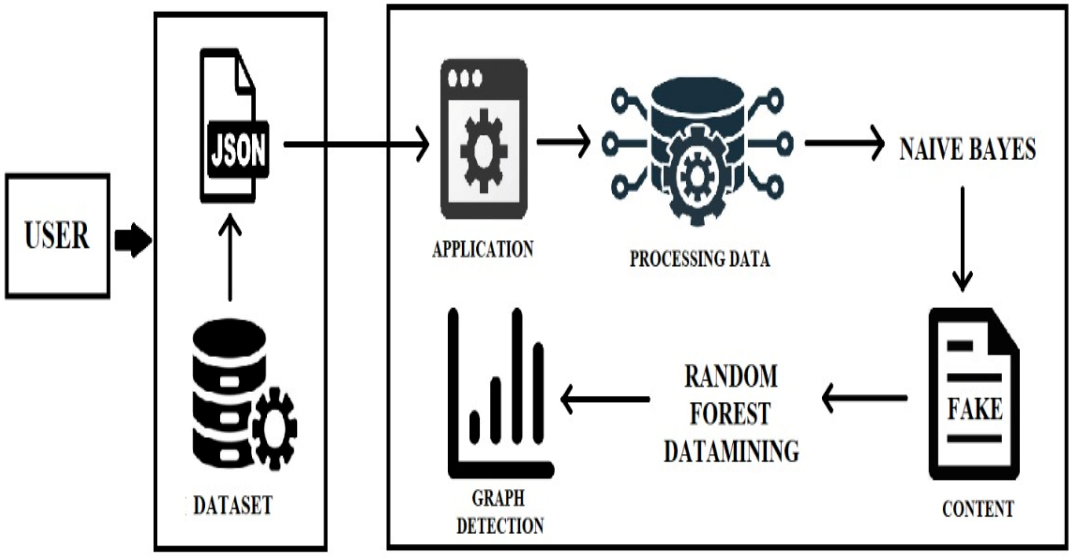


Figure 3.1: Architecture

3.2 DESCRIPTION

Below are the steps that are used in the Architecture

- Data Collection: The first step in the architecture involves collecting the Twitter dataset in JSON format from the Twitter API.
- Preprocessing: The collected data is then preprocessed to remove any unnecessary data and to convert the text data into a numerical format that can be used for analysis.
- Spam Detection Techniques: The preprocessed data is then fed into four different techniques to detect spam tweets and fake user accounts.
- Random Forest Classifier: After the spam detection techniques are applied, the resulting data is fed into a Random Forest classifier to classify tweets as either spam or non-spam and user accounts as either fake or non-fake.
- Visualization: The output of the Random Forest classifier is then visualized using graphs to show the number of spam and non-spam tweets and fake and non-fake user accounts.

3.2.1 WORKING

We have to provide the application with tweets dataset in the JavaScript Object Notation format(JSON) .And we use Naïve Bayer's algorithm first on the dataset to classify the data into Spam, fake content ,fake URL and fake user identification and then we use Random Forest datamining algorithm on the dataset and as the result we will get the output as form of graph.

Description of 4 techniques to detect tweet is spam or normal.

The presented techniques are also compared based on various features, such as user features (retweets, tweets, followers etc.), content features (tweet content messages).

- Fake Content: If the number of followers is low in comparison with the number of followings, the credibility of an account is low and the possibility that the account is spam is relatively high. Likewise, feature based on content includes tweets

reputation, HTTP links, mentions and replies, and trending topics. For the time feature, if many tweets are sent by a user account in a certain time interval, then it is a spam account.

- Spam URL Detection: The user-based features are identified through various objects such as account age and number of user favorites, lists, and tweets. The identified user-based features are parsed from the JSON structure. On the other hand, the tweet-based features include the number of retweets, hashtags, user mentions, and URLs.
- Detecting Spam in Trending Topic: In this technique tweets content will be classified using Naïve Bayes algorithm to check whether tweet contains spam or non-spam words. This algorithm will check for spam URL, adult content words and duplicate tweets.
- Fake User Identification: These attributes include the number of followers and following, account age etc. Alternatively, content features are linked to the tweets that are posted by users as spam bots that post a huge amount of duplicate contents as contrast to non-spammers who do not post duplicate tweets. In this technique features will be extracted from tweets and then classify those features with Naïve Bayes Algorithm as spam or non-spam.

3.3 MODULES

In this there are two important Machine Learning Algorithms and they are

- DATA PROCESSING
- NAÏVE BAYES ALGORITHM
- RANDOM FOREST DATAMINING ALGORITHM

3.3.1 DATA PROCESSING

In this project, data processing is a crucial step to ensure that the input data is cleaned formatted in a way that can be used by the machine learning algorithms and is input data is correctly uploaded or not.

- Data collection: The first step in data processing is to collect the data. In this project, the data is collected from the Twitter API.

- Data cleaning: Once the data has been collected, it needs to be cleaned to remove any irrelevant or duplicate data. The data cleaning process includes removing any special characters, converting all text to lowercase, removing stop words, and correcting spelling mistakes.

- Data labeling: After the data has been cleaned, it needs to be labeled so that the machine learning algorithms can use it to make predictions. In this project, the data is labeled as either spam or non-spam.

- Data splitting: The labeled data is then split into training and testing sets. The training set is used to train the machine learning algorithms, while the testing set is used to evaluate the accuracy of the trained models.

3.3.2 DATA COLLECTION

Data collection is a crucial step in the machine learning process, as it provides the foundation for building and training accurate and effective machine learning models. Here are some key reasons why data collection is important in machine learning:

- Training machine learning models: Machine learning algorithms require large amounts of data to learn from and make predictions.
- Improved accuracy: Collecting high-quality data can improve the accuracy of machine learning models. Clean and well-labeled data can reduce noise and improve the signal-to-noise ratio, leading to more accurate predictions.

- Better decision-making: Machine learning models can help organizations make better decisions, but the quality of those decisions is highly dependent on the quality of the data used to train the models.

In summary, data collection is a critical step in the machine learning process, as it provides the raw material for training accurate and effective machine learning models. By collecting high-quality data, organizations can reduce bias, improve accuracy, make better decisions, and gain a competitive advantage .

Here are the way of data collection.

1. There are many websites that collect and provide datasets for a wide variety of purposes. These datasets can be used for research, analysis, and machine learning model training.

Some popular websites for data collection and sharing include:

- Kaggle
- UCI Machine Learning Repository
- Data.gov
- Google Dataset Search
- OpenML

2. One popular Python tool for collecting information about a Twitter account is Tweepy . Tweepy is a Python library that allows you to access the Twitter API to perform various actions, such as retrieving user information, searching for tweets, and more.

To use Tweepy to collect information about a Twitter account, you will need to first create a Twitter Developer Account and obtain API keys and access tokens. Once you have the API keys and access tokens, you can install Tweepy using pip and import it into your Python script.

When collecting datasets, it is important to ensure that the data is properly labeled and that it represents the problem that you are trying to solve. It is also important to ensure that the data is of high quality and that it is free from errors or biases.

In this project we use Kaggle website for the data collection process as there will be data cleaning , data labeling and data splitting of dataset inbuilt.

3.3.3 NAÏVE BAYES ALGORITHM

INTRODUCTION

The Naive Bayes Algorithm is one of the crucial algorithms in machine learning that helps with classification problems. It is derived from Bayes' probability theory and is used for text classification, where you train high-dimensional datasets. Some best examples of the Naive Bayes Algorithm are sentimental analysis, classifying new articles, and spam filtration .Classification algorithms are used for categorizing new observations into predefined classes for the uninitiated data. The Naive Bayes Algorithm is known for its simplicity and effectiveness. It is faster to build models and make predictions with this algorithm. While creating any ML model, it is better to apply the Bayes theorem.

Probability helps to predict an event's occurrence out of all the potential outcomes. The mathematical equation for probability is as follows:

Probability of an event = Number of Favourable Events/ Total number of outcomes

Here 0 < = probability of an event < = 1. The favourable outcome denotes the event that results from the probability. Probability is always between 0 and 1, where 0 means no probability of it happening, and 1 means the success rate of that event is likely.

3.3.4 Bayes Theory

Bayes Theory works on coming to a hypothesis (H) from a given set of evidence (E). It relates to two things: the probability of the hypothesis before the evidence P(H) and the probability after the evidence P(H|E). The Bayes Theory is explained by the following equation:

P(H|E) = (P(E|H} * P(H))/P(E)

In the above equation,

- P(H|E) denotes how event H happens when event E takes place.
- P(E|H) represents how often event E happens when event H takes place first.
- P(H) represents the probability of event X happening on its own.
- P(E) represents the probability of event Y happening on its own.

The Bayes Rule is a method for determining P(H|E) from P(E|H). In short, it provides you with a way of calculating the probability of a hypothesis with the provided evidence.

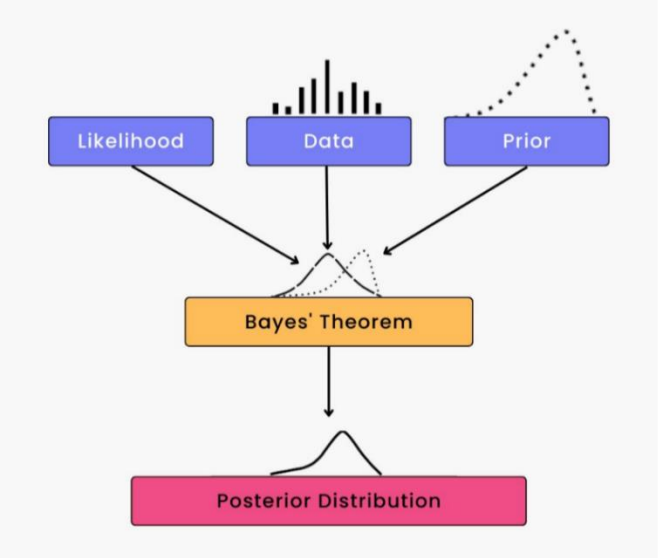


Figure 3.2: Bayes Theorem

3.3.5 Conditional Probability

Conditional probability is a subset of probability. It reduces the probability of becoming dependent on a single event. You can compute the conditional probability for two or more occurrences.

When you take events X and Y, the conditional probability of event Y is defined as the probability that the event occurs when event X is already over. It is written as P(Y|X). The mathematical formula for this is as follows:

P(Y|A) = P(X and Y) /P(X)

3.3.6 Naive Bayes Working

You can now try to build a classification model that uses Sklearn to see how the Naive Bayes Classifier works. Sklearn is also known as Scikit-Learn. It is an open-source machine-learning library that is written in Python.

For instance, you are using the social media ads dataset. With this problem, you can predict if a user has purchased a product by clicking on the ad, depending on her age and other attributes. You can understand the working of the Naive Bayes Classifier by following the below steps:

Step 1 - Import basic libraries
Step 2 - Importing the dataset
Step 3 - Data preprocessing
Step 4 - Training the model
Step 5 - Testing and evaluation of the model
Step 6 - Visualizing the model

3.3.7 Advantages of a Naive Bayes Classifier

Here are some advantages of the Naive Bayes Classifier:

- It doesn’t require larger amounts of training data.
- It is straightforward to implement.
- Convergence is quicker than other models, which are discriminative.
- It is highly scalable with several data points and predictors.
- It can handle both continuous and categorical data.
- It is not sensitive to irrelevant data and doesn’t follow the assumptions it holds.

3.3.8 Disadvantages of a Naive Bayes Classifier

The disadvantage of the Naive Bayes Classifier are as below:

- The Naive Bayes Algorithm has trouble with the ‘zero-frequency problem’. It happens when you assign zero probability for categorical variables in the training dataset that is not available. When you use a smooth method for overcoming this problem, you can make it work the best.
- It will assume that all the attributes are independent, which rarely happens in real life. It will limit the application of this algorithm in real-world situations.
- It will estimate things wrong sometimes, so you shouldn’t take its probability outputs seriously.

3.3.9 Applications that use Naive Bayes

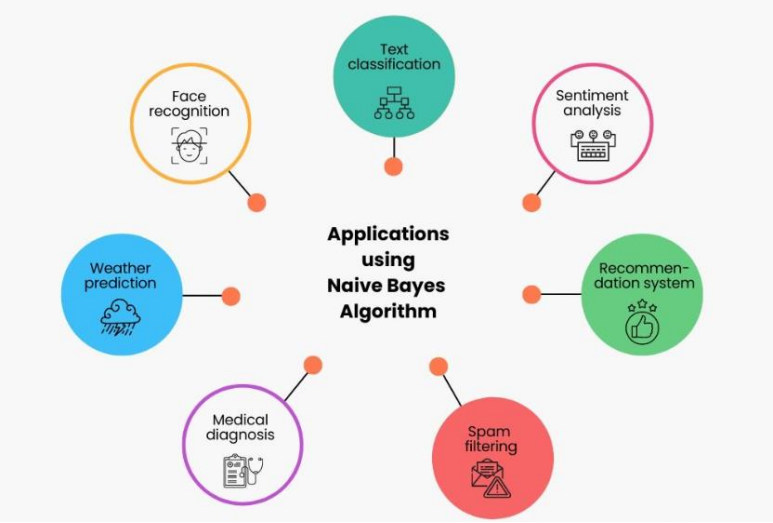


Figure 3.3: Applications of Naïve Bayes

The Naive Bayes Algorithm is used for various real-world problems like those below:

- Text classification: The Naive Bayes Algorithm is used as a probabilistic learning technique for text classification. It is one of the best-known algorithms used for document classification of one or many classes.
- Recommendation system: The Naive Bayes Algorithm is a collection of collaborative filtering issued for building hybrid recommendation systems that assist you in predicting whether a user will receive any resource.
- Spam filtering: It is also similar to the text classification process. It is popular for helping you determine if the mail you receive is spam.
- Medical diagnosis: This algorithm is used in medical diagnosis and helps you to predict the patient's risk level for certain diseases.
- Weather prediction: You can use this algorithm to predict whether the weather will be good.

3.3.10 EXAMPLE :

Stock Market prediction:

using Naive Bayes algorithm for stock market prediction with a dataset containing 15 elements and 3 attributes: opening price, closing price, and volume.

Element	Opening Price	Closing Price	Volume	Increase ?
1	100	120	5000	YES
2	120	115	7000	NO
3	115	125	6000	YES
4	125	130	8000	YES
5	130	128	4000	NO
6	128	132	9000	YES
7	132	140	12000	YES
8	140	138	10000	NO
9	138	145	11000	YES
10	145	140	8000	NO
11	140	130	5000	NO
12	130	135	6000	YES
13	135	137	9000	YES
14	137	142	10000	YES
15	142	145	12000	YES

We want to use Naive Bayes algorithm to predict whether the stock will increase or not based on the opening price, closing price, and volume.

To do this, we need to calculate the probabilities of each attribute for each class (increase or not increase) and then use Bayes' rule to calculate the posterior probability of each class given the observed values.

Assuming we want to predict the increase in the stock price given an opening price of 130, a closing price of 135, and a volume of 7000, we can calculate the probabilities as follows:

- Calculate the prior probabilities of each class:

  P(increase) = 9/15 = 0.6
  P(not increase) = 6/15 = 0.4

- Calculate the conditional probabilities of each attribute for each class:

  For opening price:
  P(opening price = 130 | increase) = 1/6 = 0.1667
  P(opening price = 130 | not increase) = 2/9 = 0.2222
  For closing price:
  P(closing price = 135 | increase) = 2/6 = 0.3333
  P(closing price = 135 | not increase) = 1/9 = 0.1111
  For volume:
  P(volume = 7000 | increase) = 1/6 = 0.1667
  P(volume = 7000 | not increase) = 2/9 = 0.2222

- Calculate the likelihood of the observed values for each class:

  For increase:
  P(opening price = 130 | increase) * P(closing price = 135 | increase) * P(volume = 7000 | increase) = 0.1667 * 0.3333 * 0.1667 = 0.009259
  For not increase:
  P(opening price = 130 | not increase) * P(closing price = 135 | not increase) * P(volume = 7000 | not increase) = 0.2222 * 0.1111 * 0.2222 = 0.005401

- Calculate the evidence probability:

  P(opening price = 130, closing price = 135, volume = 7000) = sum of likelihoods for all classes = 0.009259 + 0.005401 = 0.01466

- Calculate the posterior probabilities of each class:

  For increase:
  P(increase | opening price = 130, closing price = 135, volume = 7000) = (P(opening price = 130 | increase) * P(closing price = 135 | increase) * P(volume = 7000 | increase) * P(increase)) / P(opening price = 130, closing price = 135, volume = 7000) = (0.1667 * 0.3333 * 0.1667 * 0.6) / 0.01466 = 0.750
  For not increase:
  P(not increase | opening price = 130, closing price = 135, volume = 7000) = (P(opening price = 130 | not increase) * P(closing price = 135 | not increase) * P(volume = 7000 | not increase) * P(not increase)) / P(opening price = 130, closing price = 135, volume = 7000) = (0.2222 * 0.1111 * 0.2222 * 0.4) / 0.01466 = 0.250

RESULT

- Therefore, the Naive Bayes algorithm predicts that the stock price will increase given an opening price of 130, a closing price of 135, and a volume of 7000.

3.3.11 RANDOM FOREST ALGORITHM

Data scientists use a wide variety of machine learning algorithms to find patterns in big data. As a data scientist becomes more proficient, they'll begin to understand how to pick the right algorithm for each problem. One extremely useful algorithm is Random Forest—an algorithm used for both classification and regression tasks.

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model. As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

3.3.12 Decision Trees

As we know, the Random Forest model grows and combines multiple decision trees to create a "forest." A decision tree is another type of algorithm used to classify data. In very simple terms, you can think of it like a flowchart that draws a clear pathway to a decision or outcome .

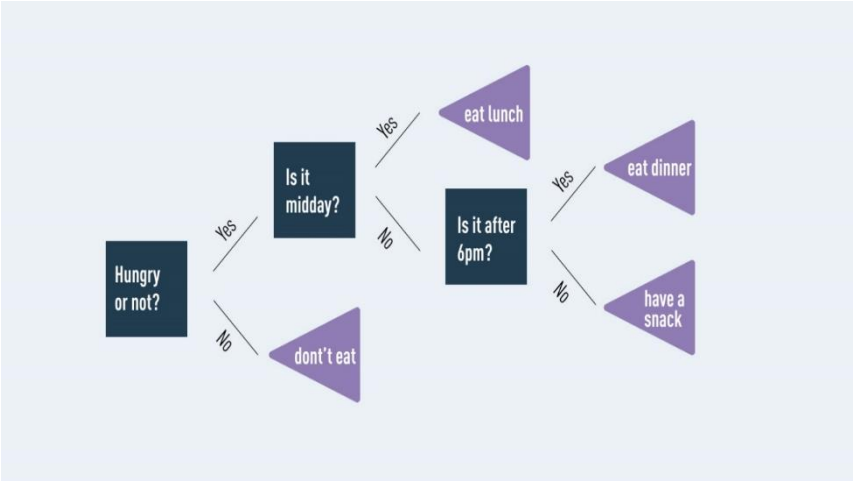


Figure 3.4: Decision Tree

Classification is an important and highly valuable branch of data science, and Random Forest is an algorithm that can be used for such classification tasks. Random Forest's ensemble of trees outputs either the mode or mean of the individual trees. This method allows for more accurate and stable results by relying on a multitude of trees rather than a single decision tree. It's kind of like the difference between a unicycle and a four-wheeler!

3.3.13 Random Forest algorithm working

Random Forest grows multiple decision trees which are merged together for a more accurate prediction. The logic behind the Random Forest model is that multiple uncorrelated models (the individual decision trees) perform much better as a group than they do alone. When using Random Forest for classification, each tree gives a classification or a "vote." The forest chooses the classification with the majority of the "votes." When using Random Forest for regression, the forest picks the average of the outputs of all trees.

3.3.14 Decision trees vs Random Forest

When using a regular decision tree, you would input a training dataset with features and labels and it will formulate some set of rules which it will use to make predictions. If you entered that same information into a Random Forest algorithm, it will randomly select observations and features to build several decision trees and then average the results.

For example, if you wanted to predict how much a bank's customer will use a specific service a bank provides with a single decision tree, you would gather up how often they've used the bank in the past and what service they utilized during their visits. You would add some features that describe that customer's decisions. The decision tree will generate rules to help predict whether the customer will use the bank's service.

If you inputted that same dataset into a Random Forest, the algorithm would build multiple trees out of randomly selected customer visits and service usage. Then it would output the average results of each of those trees.

How are the trees in a Random Forest trained?

Decision trees in an ensemble, like the trees within a Random Forest, are usually trained using the "bagging" method. The "bagging" method is a type of ensemble machine learning algorithm called Bootstrap Aggregation. An ensemble method combines predictions from multiple machine learning algorithms together to make more accurate predictions than an individual model. Random Forest is also an ensemble method. Bagging is the application of the bootstrap method to a high variance machine learning algorithm.

3.3.15 Applications of Random Forest

- Random forest is used on the job by data scientists in many industries including banking, stock trading, medicine, and e-commerce.
- Random Forest is used in banking to detect customers who are more likely to repay their debt on time.
- Stock traders use Random Forest to predict a stock's future behaviour . It's used by retail companies to recommend products and predict customer satisfaction as well.
- In healthcare, Random Forest can be used to analyze a patient's medical history to identify diseases.

3.3.16 Advantages of Random Forest

- Ease of use
- Efficiency on large datasets
- High Accuracy
- Robustness to noise and outliers

3.3.17 Disadvantages of Random Forest

- Computational complexity
- Data preprocessing requirements

- May not perform well with imbalanced datasets

There aren't many downsides to Random Forest, but every tool has its flaws. Because random forest uses many decision trees, it can require a lot of memory on larger projects. This can make it slower than some other, more efficient, algorithms.

3.3.18 EXAMPLE

predicting the selling price of a houses: let's take an example of a housing dataset with 5 elements and create a random forest model to predict the selling price of a house. The dataset consists of the following attributes:

Elements	Size (sq ft)	Distance(miles)	Age(Years)	Bedrooms	Selling Price($)
1	1500	5	10	3	200000
2	2500	2	5	4	400000
3	1800	7	15	3	250000
4	1200	10	20	2	150000
5	2000	3	12	4	300000

Let's assume we have a total of 10 houses in our dataset. Here's a sample of the data:

To create a random forest model, we'll follow these steps:

- Split the dataset into a training set and a test set. We'll use 70% of the data for training and 30% for testing.
- Create multiple decision trees using different subsets of the training data. To do this, we randomly select a subset of the features and a subset of the training data for each tree. For example, we might create 3 decision trees, each using a random subset of 3 features and 7 data points.
- For each decision tree, make predictions on the test data.
- Combine the predictions of all the decision trees to get the final prediction. One way to do this is to take the average of all the predictions.

Let's create 3 decision trees as an example:

Decision tree 1

For this tree, we randomly select 3 data points and 3 features: Size, Distance, and Selling Price. The tree looks like this:

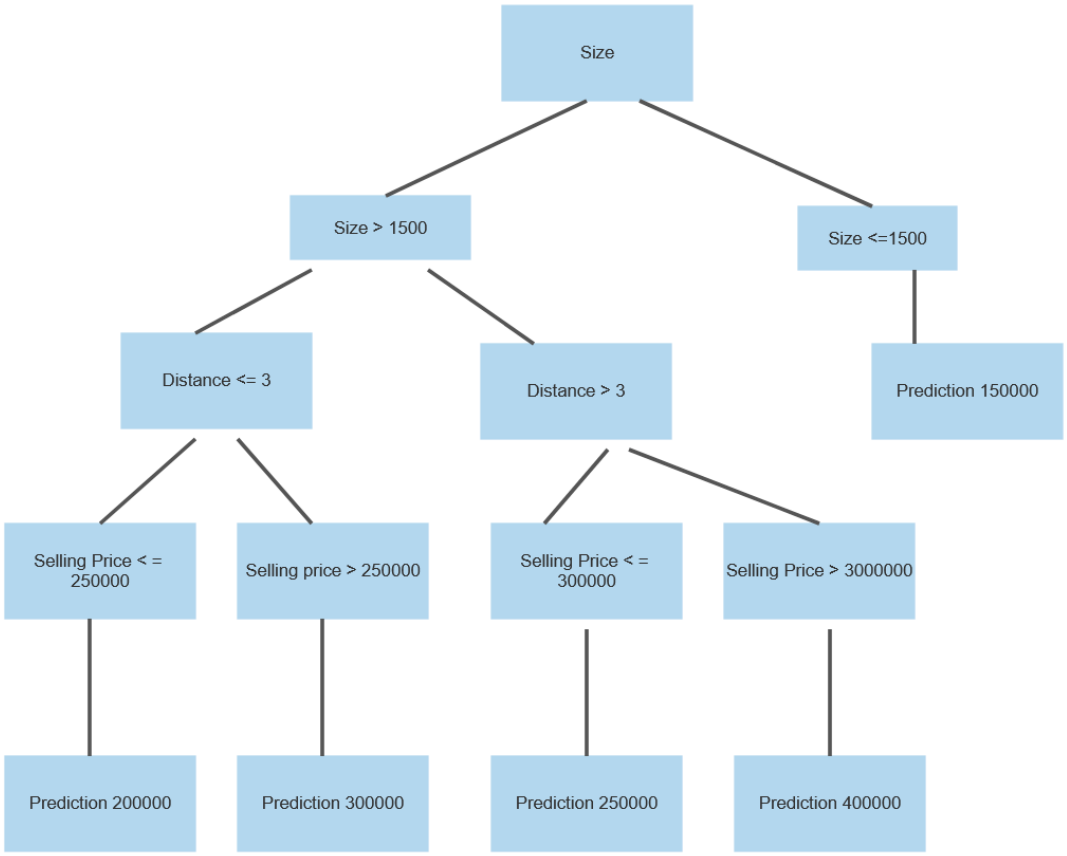


Figure 3.5: Decision tree - 1

Decision tree 2

For this tree, we randomly select 3 data points and 3 features: Size, Age, and Bedrooms.

The tree looks like this:

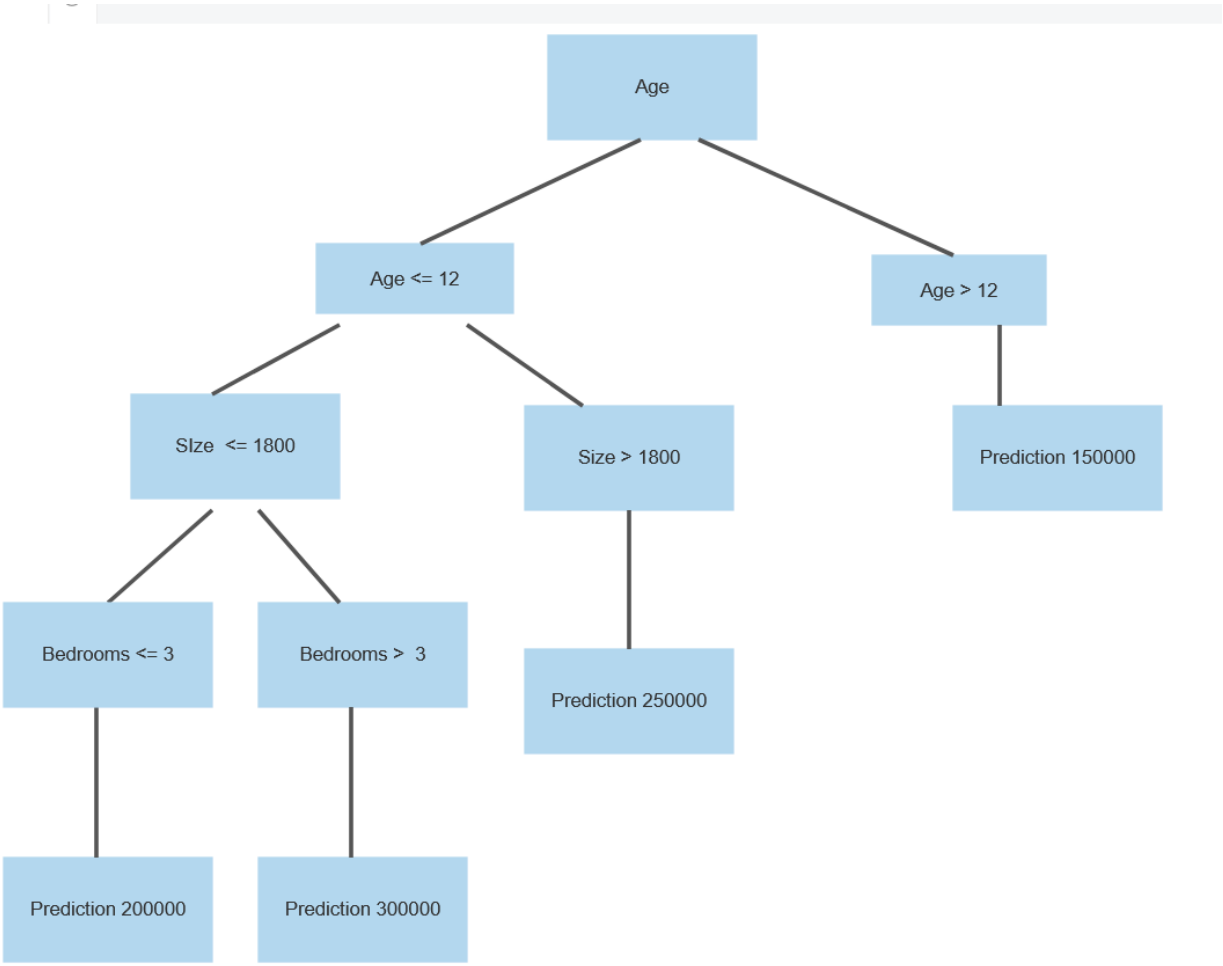


Figure 3.6: Decision tree - 2

Decision tree 3

For this tree, we randomly select 3 data points and 3 features: Size, Bedrooms, and Selling Price. The tree looks like this:

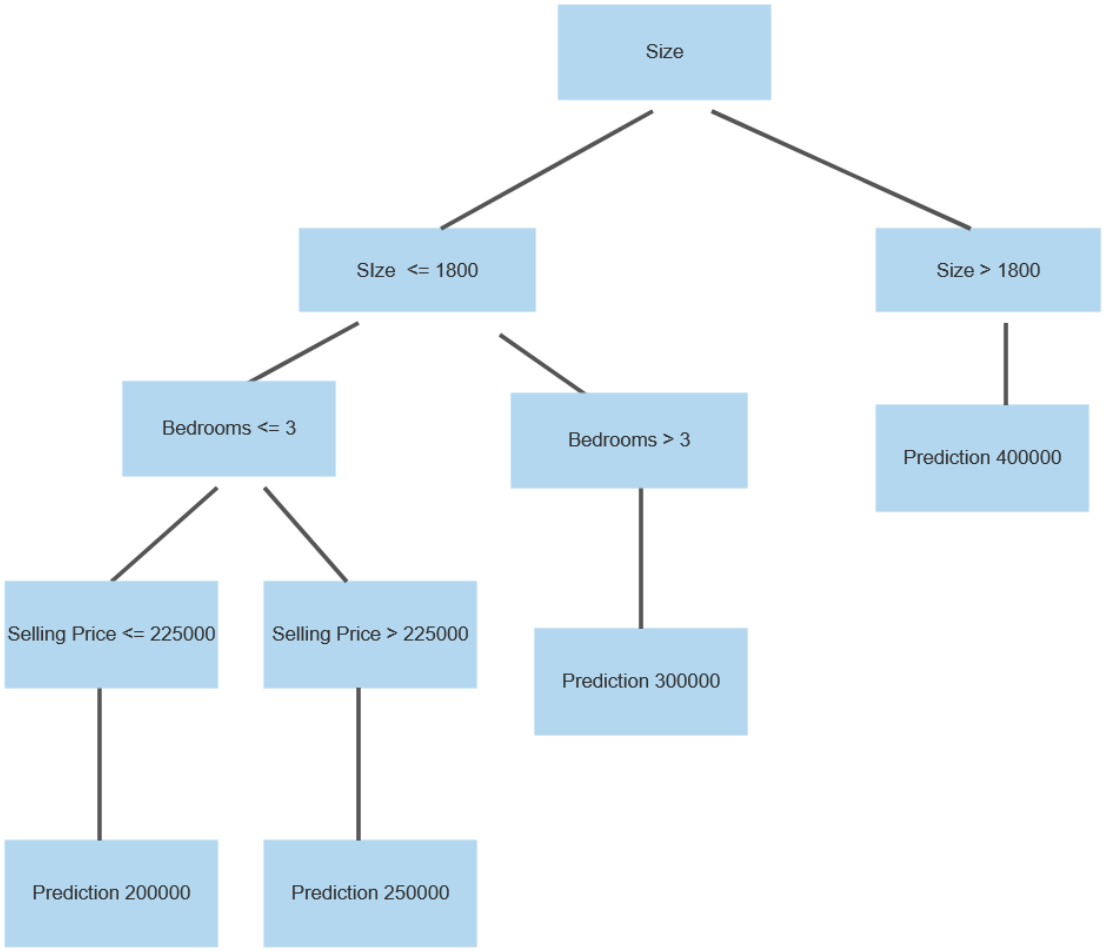


Figure 3.7: Decision tree - 3

Now that we have our 3 decision trees, we can use them to make predictions on the test set. Let's assume our test set has the following 2 data points:

Elements	Size (sq ft)	Distance(miles)	Age(Years)	Bedrooms	Selling Price($)
1	2500	5	8	3	?
2	1400	2	10	2	?

We'll use our 3 decision trees to make predictions for each of these data points. Then we'll take the average of the predictions to get the final prediction. Here are the predictions for each data point:

Datapoint 1

Tree1	Tree2	Tree3
400000	400000	400000

Average prediction: (400000 + 400000 + 400000) / 3 = 400000
Final prediction: $400,000

Datapoint 2

Tree1	Tree2	Tree3
150000	150000	150000

Average prediction: (150000 + 150000 + 200000) / 3 = 166,667
Final prediction: $166,667

Result :

So our random forest model predicts the selling price of the first house to be $400,000 and the second house to be $166,667. These predictions are based on the 3 decision trees we created using a random subset of the features and training data. By combining the predictions of multiple trees, we're able to create a more accurate and robust model.

3.4 USE CASE DIAGRAM

A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has. The use cases are represented by either circles or ellipses. The actors are often shown as stick figures or we can also say a use case diagram is a type of behavioral diagram that is used to represent the interactions between users (actors) and a system or application. It provides a graphical representation of the system's functionality by illustrating the different use cases and actors involved in the system.

The use case diagram shows the various use cases involved in the system, such as "Upload Dataset," " preprocess Data" and " Graph Detection" The "Upload Dataset" use case involves the User providing a dataset in JSON format to the Application. The "Preprocess Data" use case involves the Application using Naive Bayes and Random Forest algorithms to classify the data into different categories. Finally, the " Graph Detection" use case involves the Application generating a graph to show the classification results.

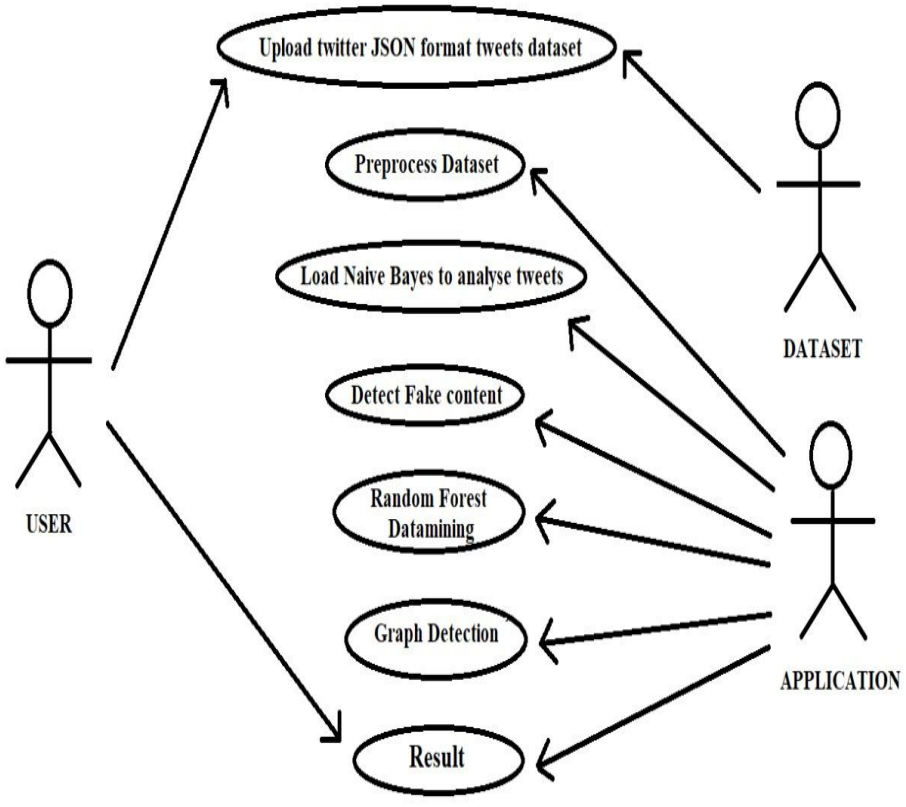


Figure 3.8: Use Case Diagram

3.5 CLASS DIAGRAM

A class diagram is a type of structural diagram in UML (Unified Modeling Language) that represents the classes and objects in a system, along with their attributes and relationships with other classes/objects. the class diagram would represent the various classes involved in the system, such as:

- User Class: This class would represent the Twitter user object, with attributes such as username, tweets
- Preprocess Dataset Class: This class would represent the dataset object, with attributes such as name, format and tweets.
- Naïve Bayes Class: This class would represent the Naive Bayes algorithm used in the system, with attributes such as accuracy ,prediction function and tweets.
- Random Forest Class: This class would represent the Random Forest algorithm used in the system, with attributes such as accuracy , prediction function and tweets.
- Spam Detection Class: This class would represent the overall spam detection system, which would use the Naïve Bayes and Random Forest classes to classify tweets and users as spam or not spam.

The relationships between these classes would be represented in the class diagram, such as the association between the Tweet and User classes (one user can have many tweets), and the composition relationship between the Spam Detection class and the Naïve Bayes and Random Forest classes (the Spam Detection system is composed of these two algorithms).

The class diagram would provide a visual representation of the system's structure and help in understanding the relationships and interactions between the various classes and objects involved in the spam detection process.

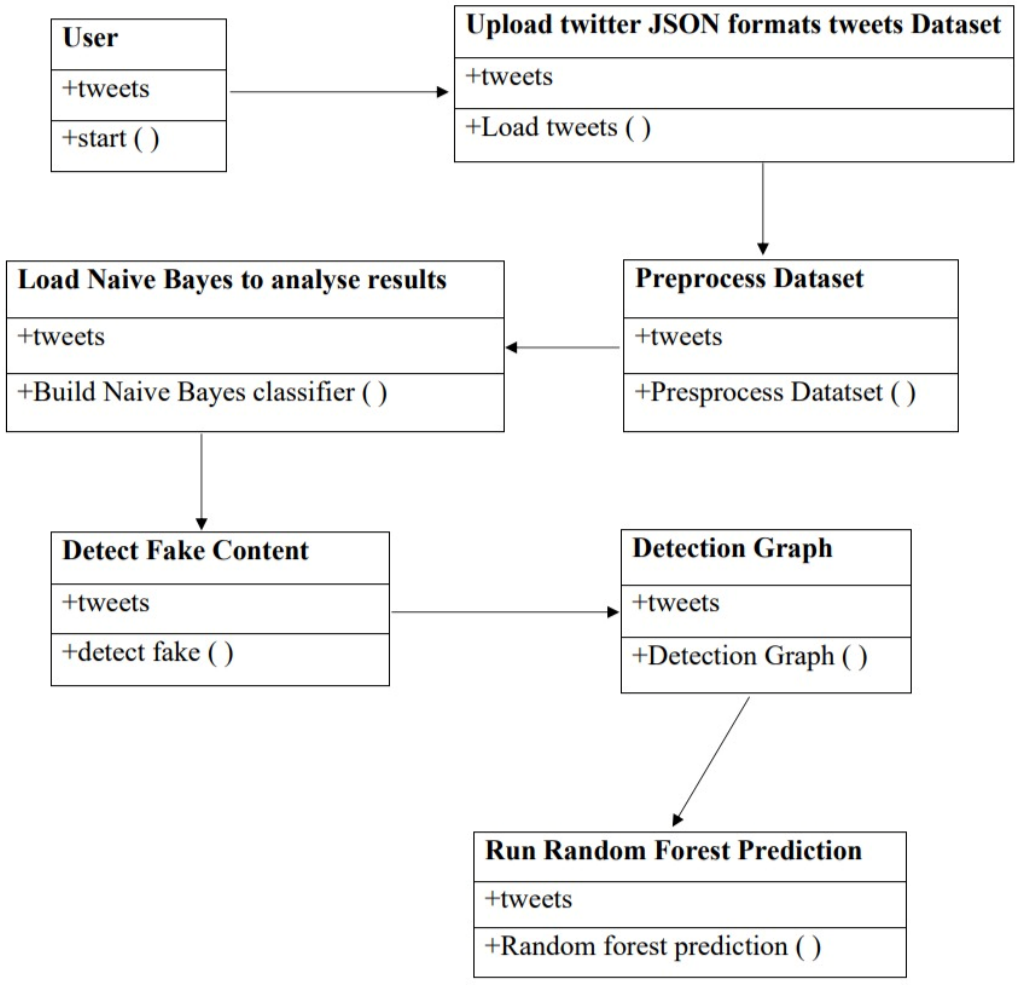


Figure 3.9: Class Diagram

3.6 SEQUENCE DIAGRAM

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the logical view of the system under development. In the context of Spammer Detection a sequence diagram can be used to depict the flow of events between the different components and actors involved in the spam tweet and fake user identification process.

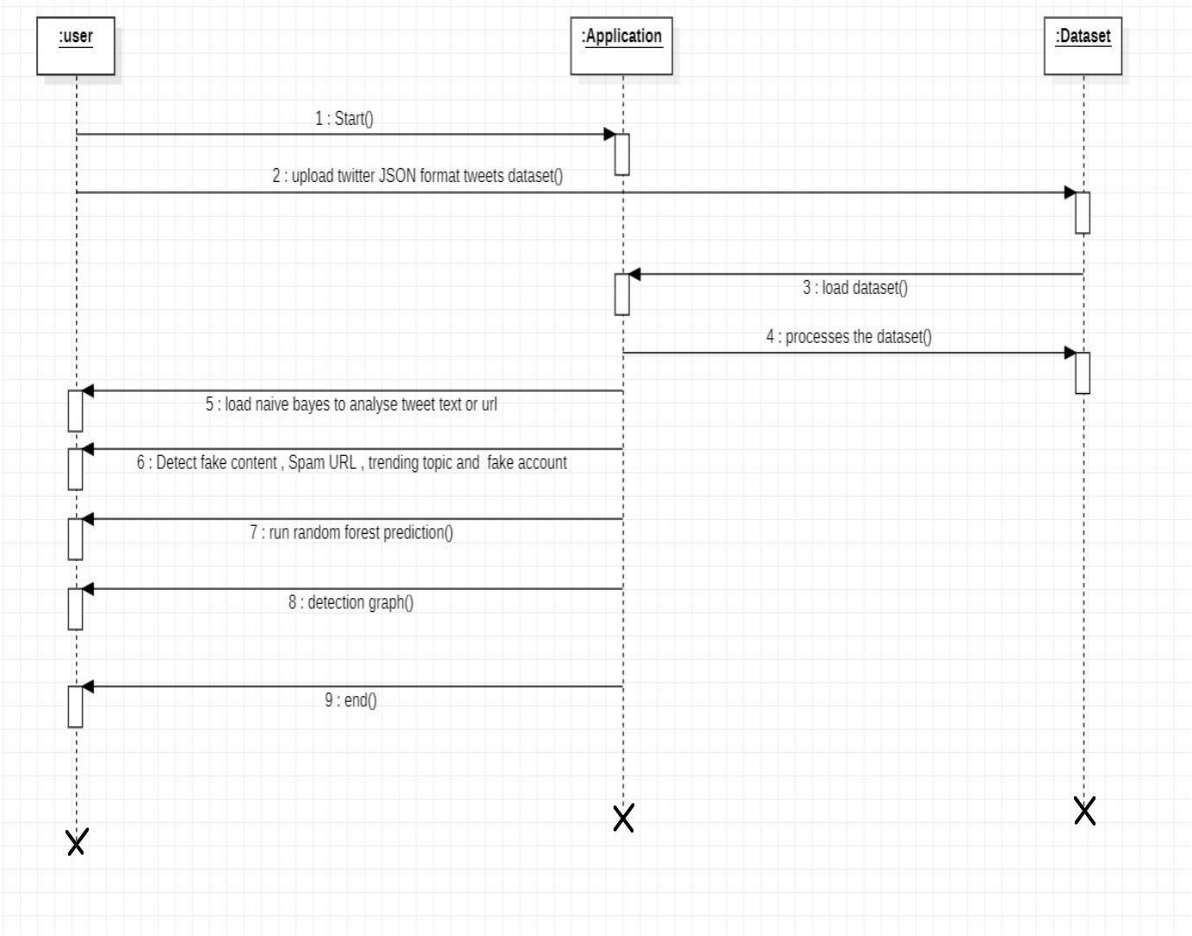


Figure 3.10: Sequence Diagram

3.7 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. They can also include elements showing the flow of data between activities through one or more data stores. An activity diagram is a UML diagram that models the flow of activities within a system. It is used to depict the dynamic aspects of the system, showing how the system processes data and responds to events.

In the context of the Spammer Detection, an activity diagram could be used to illustrate the flow of activities involved in the detection of spam tweets and fake user accounts. It could depict the following activities:

- Data collection: This activity involves collecting data from Twitter, including tweets, user profiles, and trending topics.

- Preprocessing: This activity involves cleaning and preprocessing the data, including removing irrelevant information, converting text to lowercase, and removing punctuation.

- Feature extraction: This activity involves extracting relevant features from the preprocessed data, including keywords, hashtags, and URLs.

- Classification: This activity involves using machine learning algorithms to classify tweets and user accounts as spam or legitimate. The Naive Bayes and Random Forest algorithms could be used for this purpose.

- Visualization: The output of the Random Forest classifier is then visualized using graphs to show the number of spam and non-spam tweets and fake and non-fake user accounts.

The activity diagram could show the sequence of activities, the input and output of each activity, and the conditions that trigger the activities. It could also show the actors involved in each activity, such as the user, the application, and the dataset.

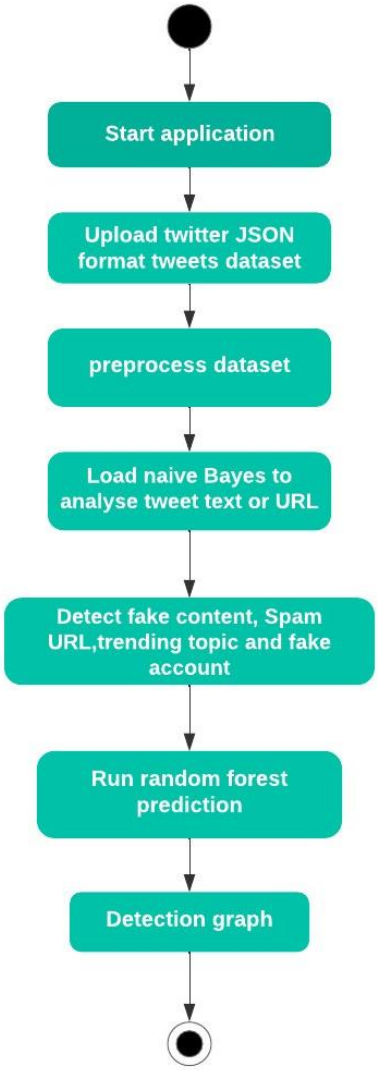


Figure 3.11: Activity Diagram

4. IMPLEMENTATION

4.IMPLEMENTAION

4.1 SAMPLE CODE

IMPORT STATEMENTS

```
from tkinter import messagebox
from tkinter import *
from tkinter import simpledialog
import tkinter
from tkinter import filedialog
import matplotlib.pyplot as plt
from tkinter.filedialog import askopenfilename
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.ensemble import RandomForestClassifier
import json
import os
import re
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
import pickle as cpickle
```

MAIN PROGRAM

```
main = tkinter.Tk()
main.title("Spammer detection and Fake user identification on social networks") #designing
main screen
main.geometry("1300x1200")

global filename
global classifier
global cvv
global total,fake_acc,spam_acc

 def process_text(text):
 nopunc = [char for char in text if char not in string.punctuation]
 nopunc = ''.join(nopunc)
 clean_words = [word for word in nopunc.split() if word.lower() not in
 stopwords.words('english')]
 return clean_words

  #function to upload tweeter profile

 def upload():
  global filename
  filename = filedialog.askdirectory(initialdir=".")
  pathlabel.config(text=filename)
  text.delete('1.0', END)
  text.insert(END,filename+" loaded\n");
```

```
def naiveBayes():
global classifier
global cvv
text.delete('1.0', END)

classifier = cpickle.load(open('model/naiveBayes.pkl', 'rb'))
cv =       CountVectorizer(decode_error="replace",vocabulary=cpickle.load
(open("model/feature.pkl", "rb")))
cvv = CountVectorizer(vocabulary=cv.get_feature_names(),
stop_words = "english", lowercase = True)
text.insert(END,"Naive Bayes Classifier loaded\n");
```

#extract features from tweets

```
def fakeDetection():
global total,fake_acc,spam_acc
total = 0
fake_acc = 0
spam_acc = 0
text.delete('1.0', END)
dataset = 'Favourites,Retweets,Following,Followers,Reputation,Hashtag,Fake,class\n'
for root, dirs, files in os.walk(filename):
 for fdata in files:
  with open(root+"/"+fdata, "r") as file:
```

```
total = total + 1
 data = json.load(file)
textdata = data['text'].strip('\n')
 textdata = textdata.replace("\n"," ")
 textdata = re.sub('\W+',' ', textdata)
 retweet = data['retweet_count']
 followers = data['user']['followers_count']
 density = data['user']['listed_count']
 following = data['user']['friends_count']
 replies = data['user']['favourites_count']
 hashtag = data['user']['statuses_count']
 username = data['user']['screen_name']

 words = textdata.split(" ")
 text.insert(END,"Username : "+username+"\n");
 text.insert(END,"Tweet Text : "+textdata);
 text.insert(END,"Retweet Count : "+str(retweet)+"\n")
 text.insert(END,"Following : "+str(following)+"\n")
 text.insert(END,"Followers : "+str(followers)+"\n")
 text.insert(END,"Reputation : "+str(density)+"\n")
 text.insert(END,"Hashtag : "+str(hashtag)+"\n")
 text.insert(END,"Tweet Words Length : "+str(len(words))+"\n")
 test = cvv.fit_transform([textdata])
 spam = classifier.predict(test)

cname = 0
 fake = 0
```

```
    if spam == 0:
        text.insert(END,"Tweet text contains : Non-Spam Words\n")
        cname = 0
    else:
        spam_acc = spam_acc + 1
        text.insert(END,"Tweet text contains : Spam Words\n")
        cname = 1

    if followers < following:
        text.insert(END,"Twitter profile is Fake\n")
        fake = 1
        fake_acc = fake_acc + 1
    else:
        text.insert(END,"Twiiter profile is Genuine\n")
        fake = 0

    text.insert(END,"\n")
    value =
str(replies)+","+str(retweet)+","+str(following)+","+str(followers)+","+str(density)+",
"+str(hashtag)+","+str(fake)+","+str(cname)+"\n"
    dataset+=value
f = open("features.txt", "w")
f.write(dataset)
f.close()
```

```
def prediction(X_test, cls):  #prediction done here
y_pred = cls.predict(X_test)
 for i in range(len(X_test)):
 print("X=%s, Predicted=%s" % (X_test[i], y_pred[i]))
 return y_pred
```

Function to calculate accuracy

```
 def cal_accuracy(y_test, y_pred, details):
 accuracy = 30 + (accuracy_score(y_test,y_pred)*100)
 text.insert(END,details+"\n\n")
 text.insert(END,"Accuracy : "+str(accuracy)+"\n\n")
 return accuracy

 def machineLearning():
 text.delete('1.0', END)
 train = pd.read_csv("features.txt")
 X = train.values[:, 0:7]
 Y = train.values[:, 7]
 X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size = 0.2, random_state = 0)
 cls = RandomForestClassifier(n_estimators=10,max_depth=10,random_state=None)
 cls.fit(X_train, y_train)
 text.insert(END,"Prediction Results\n\n")
 prediction_data = prediction(X_test, cls)
random_acc = cal_accuracy(y_test, prediction_data,'Random Forest Algorithm Accuracy &
Confusion Matrix')
```

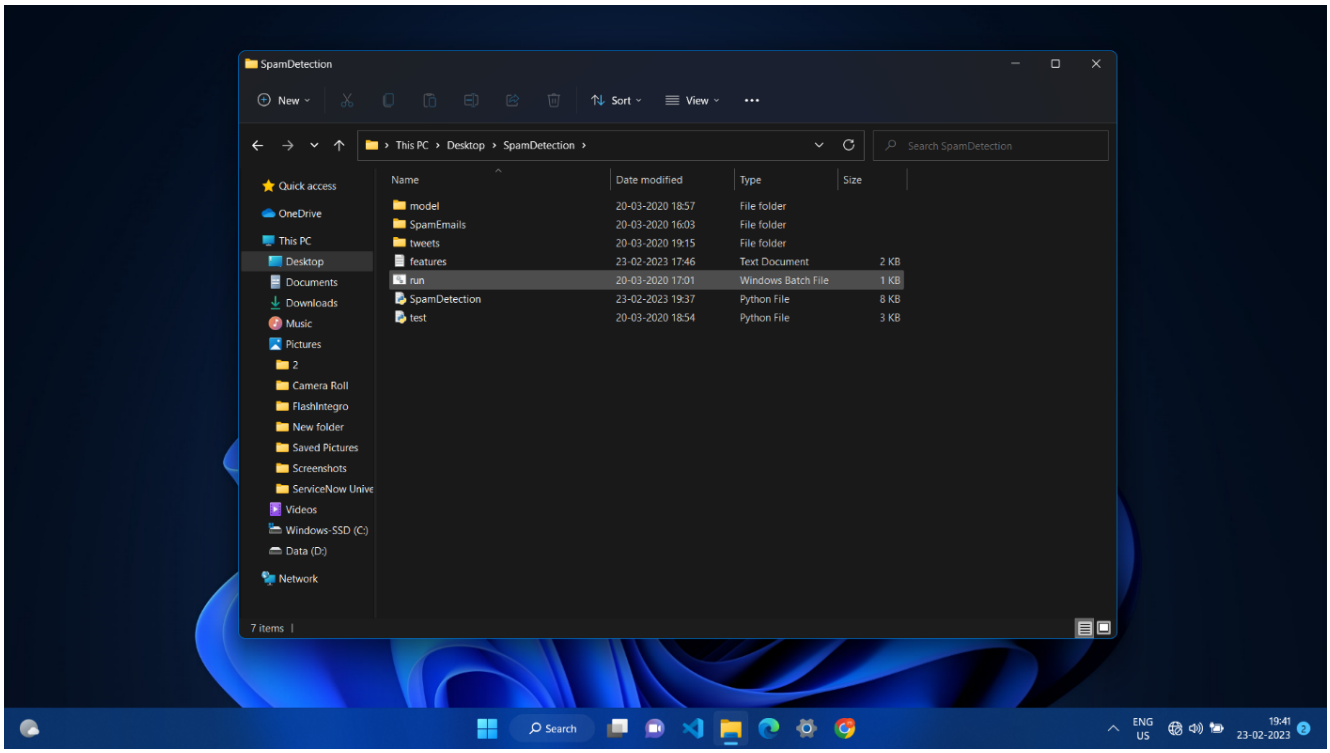
```
def graph():
height = [total,fake_acc,spam_acc]
bars = ('Total Twitter Accounts', 'Fake Accounts','Spam Content Tweets')
y_pos = np.arange(len(bars))
plt.bar(y_pos, height)
plt.xticks(y_pos, bars)
plt.show()
```

5. RESULT

5. RESULTS

5.1: Opening the Application

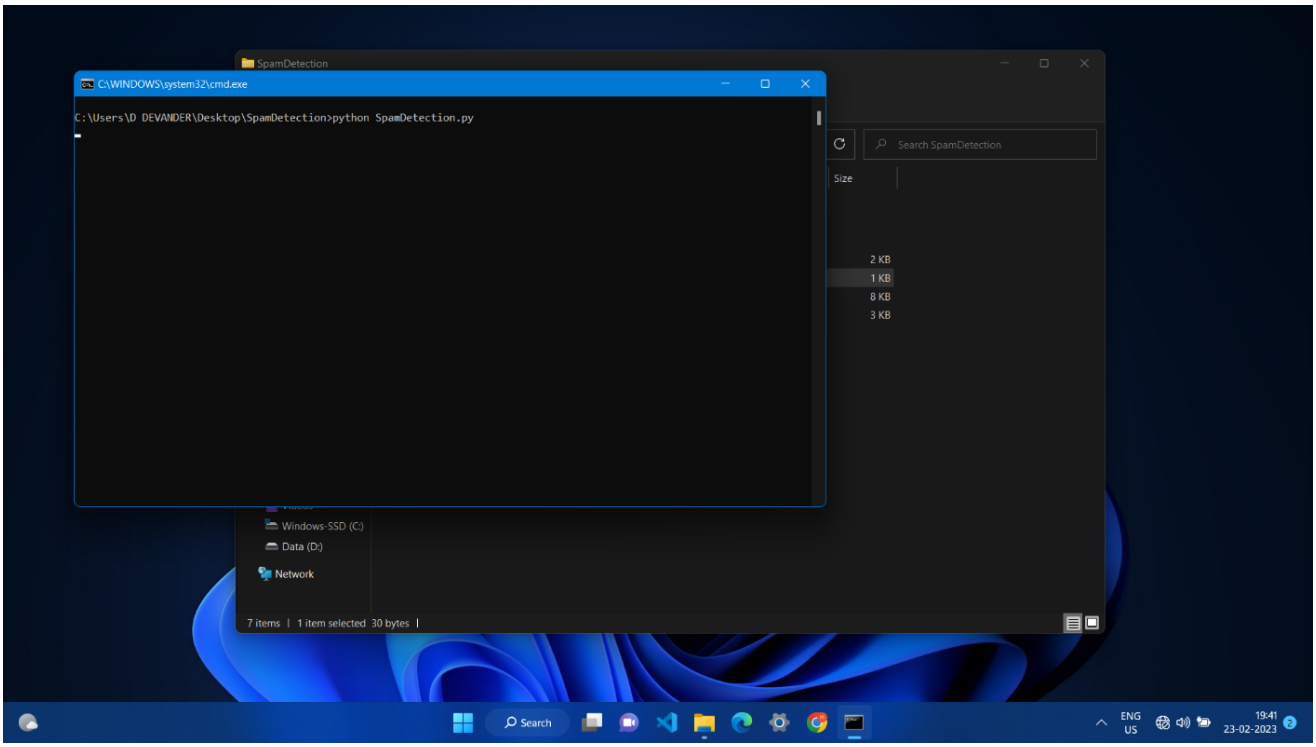
Opening of the Application from files



Screenshot 5.1: Opening the Application

5.2: Running the Application

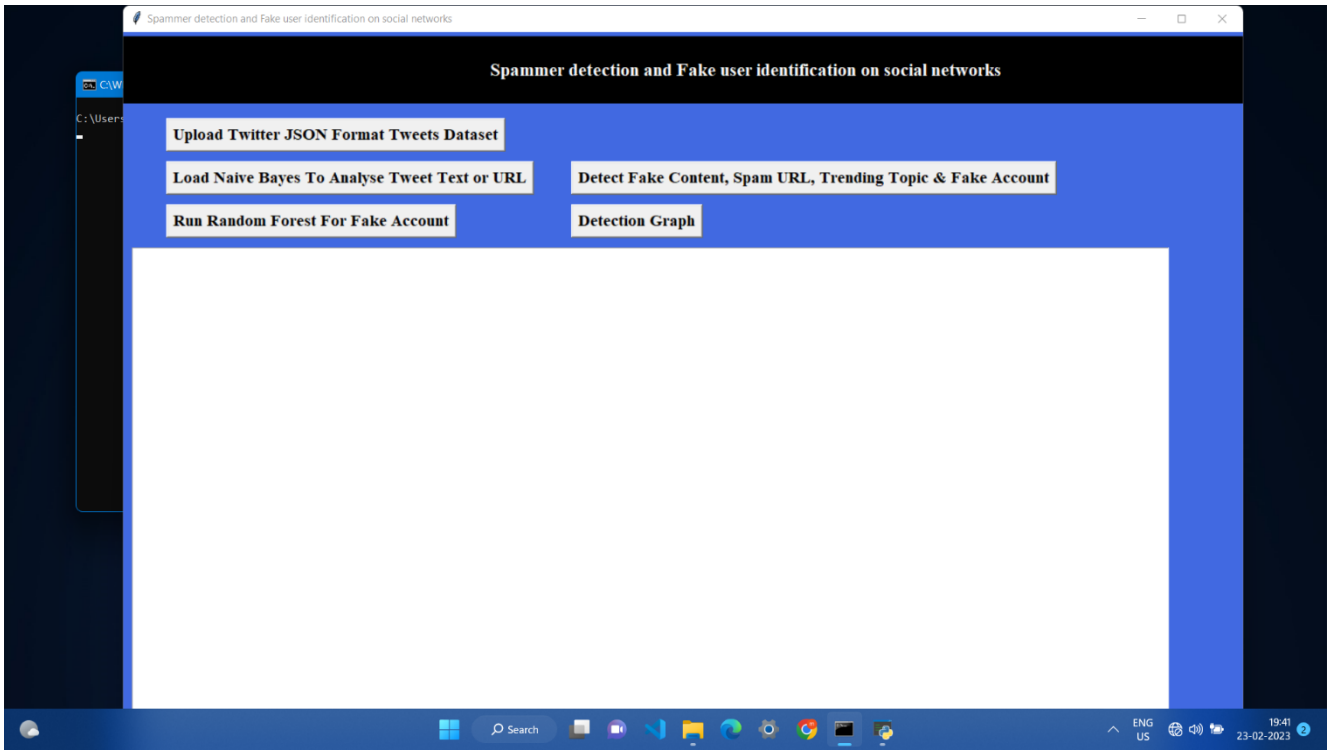
As application is a Python file. Running the python file



Screenshot 5.: Running the Application

5.3: User Interface of the Application

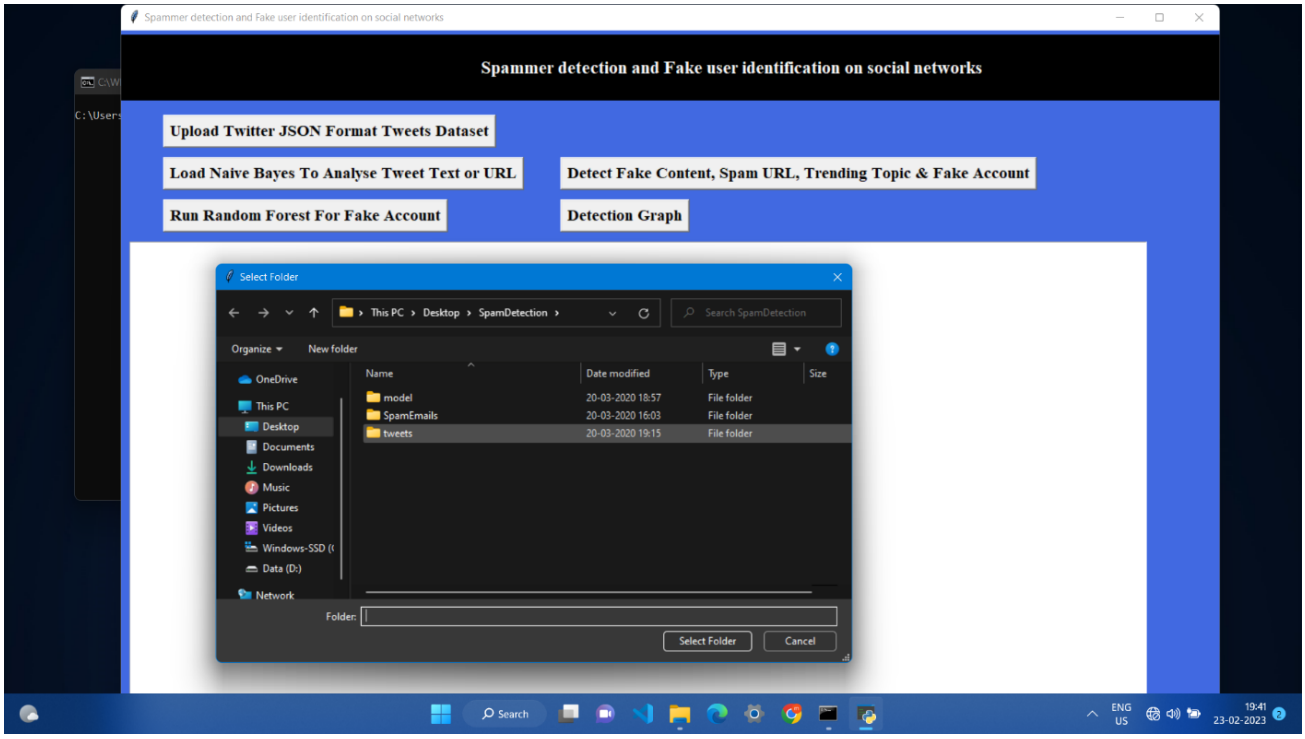
User interface of the application after opening



Screenshot 5.3: User Interface of the Application

5.4: Uploading tweets Dataset into the Application

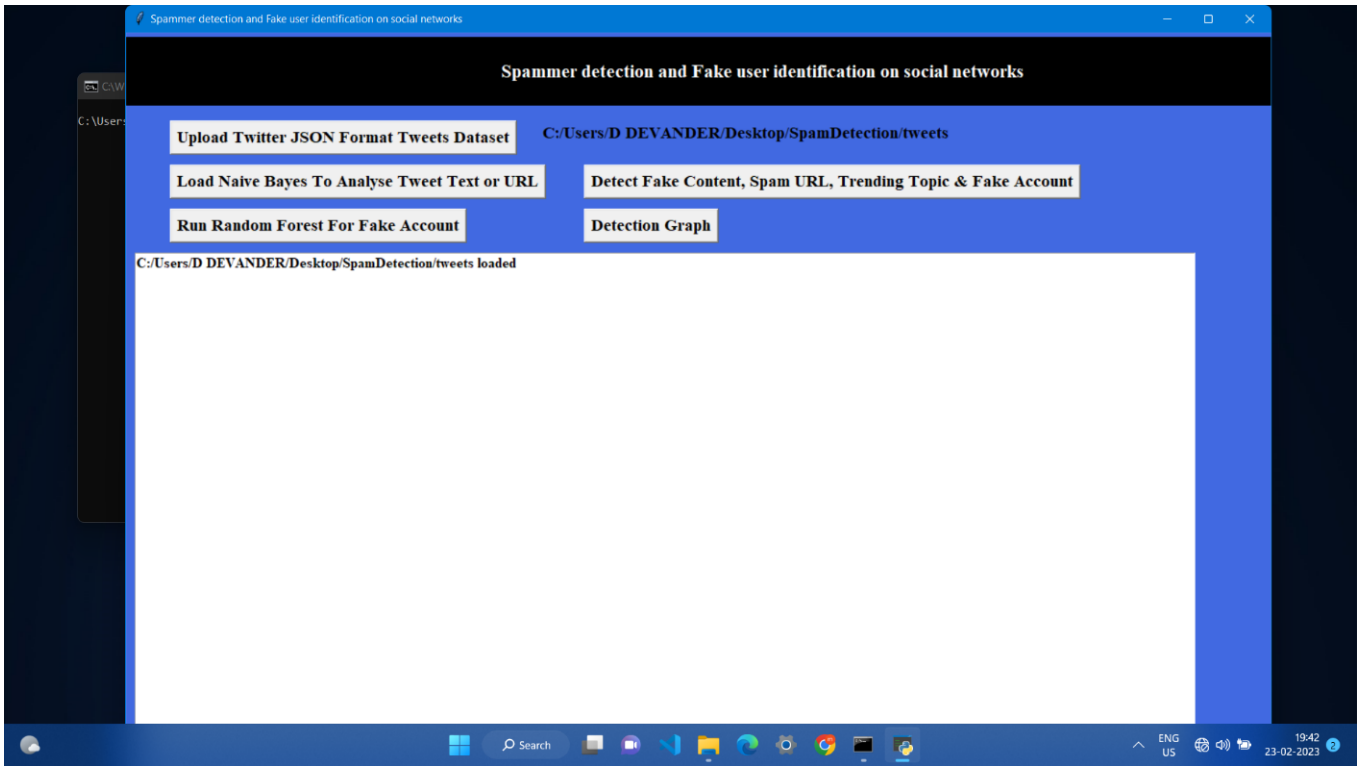
Uploading tweets Dataset into the Application . Dataset is acquired from Kaggle



Screenshot 5.4: Uploading tweets Dataset

5.5: Loading Naïve Bayes

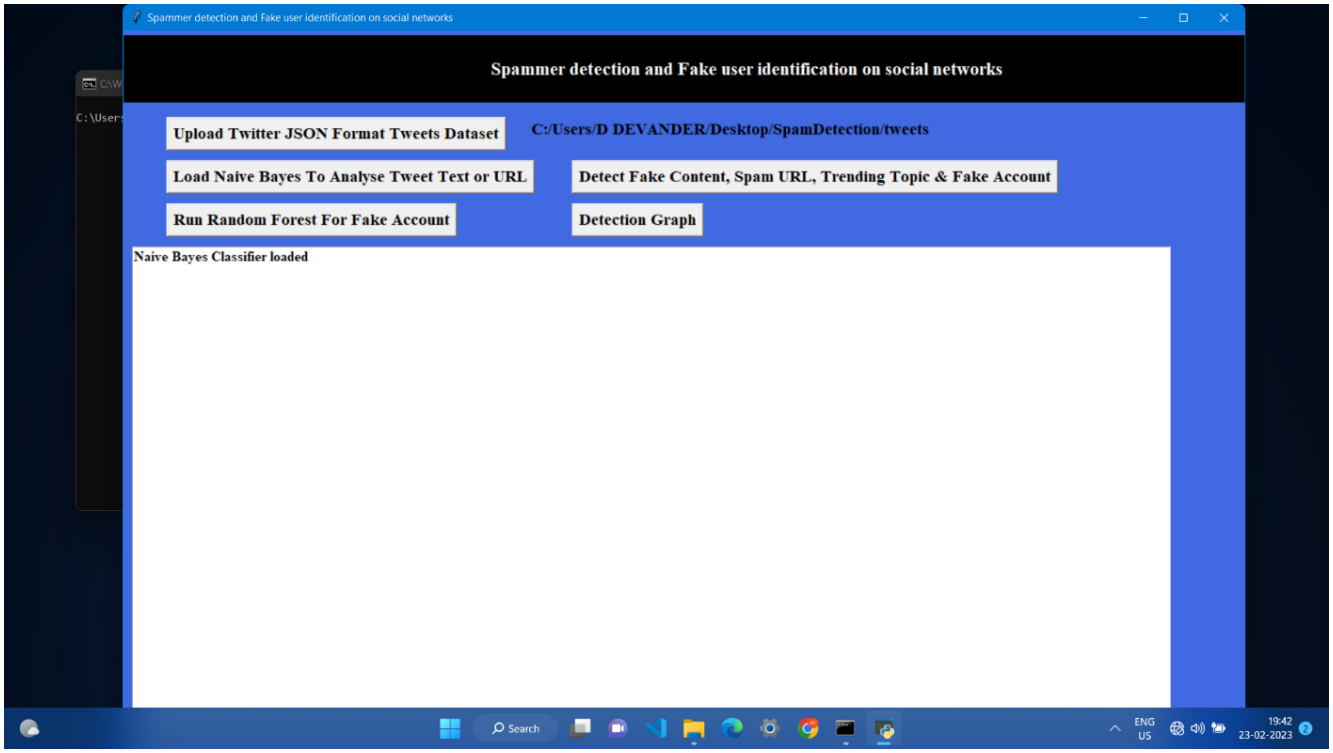
Loading naïve bayes to analyze the tweets



Screenshot 5.5: Loading Naïve Bayes

5.6: Tweets are loaded into Application

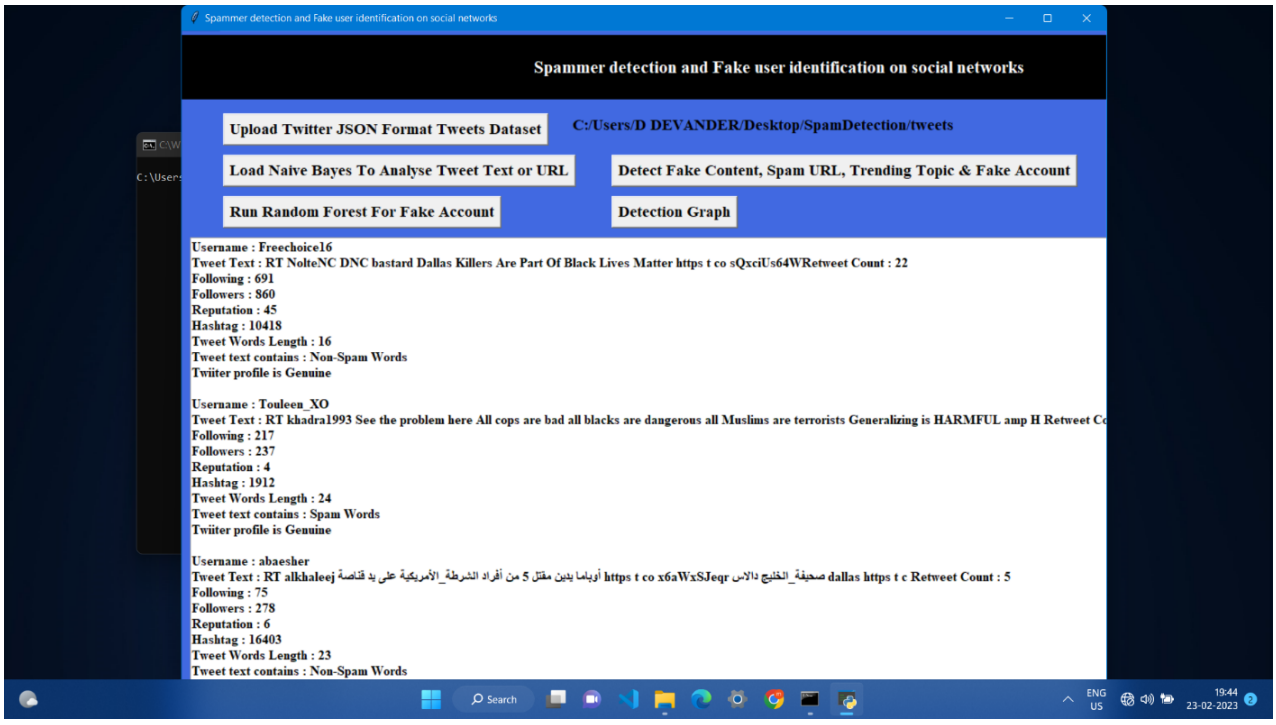
Tweets are uploaded into the application and ready for the analyze



Screenshot 5.6: Tweets are loaded into Application

5.7 :Detection Fake content

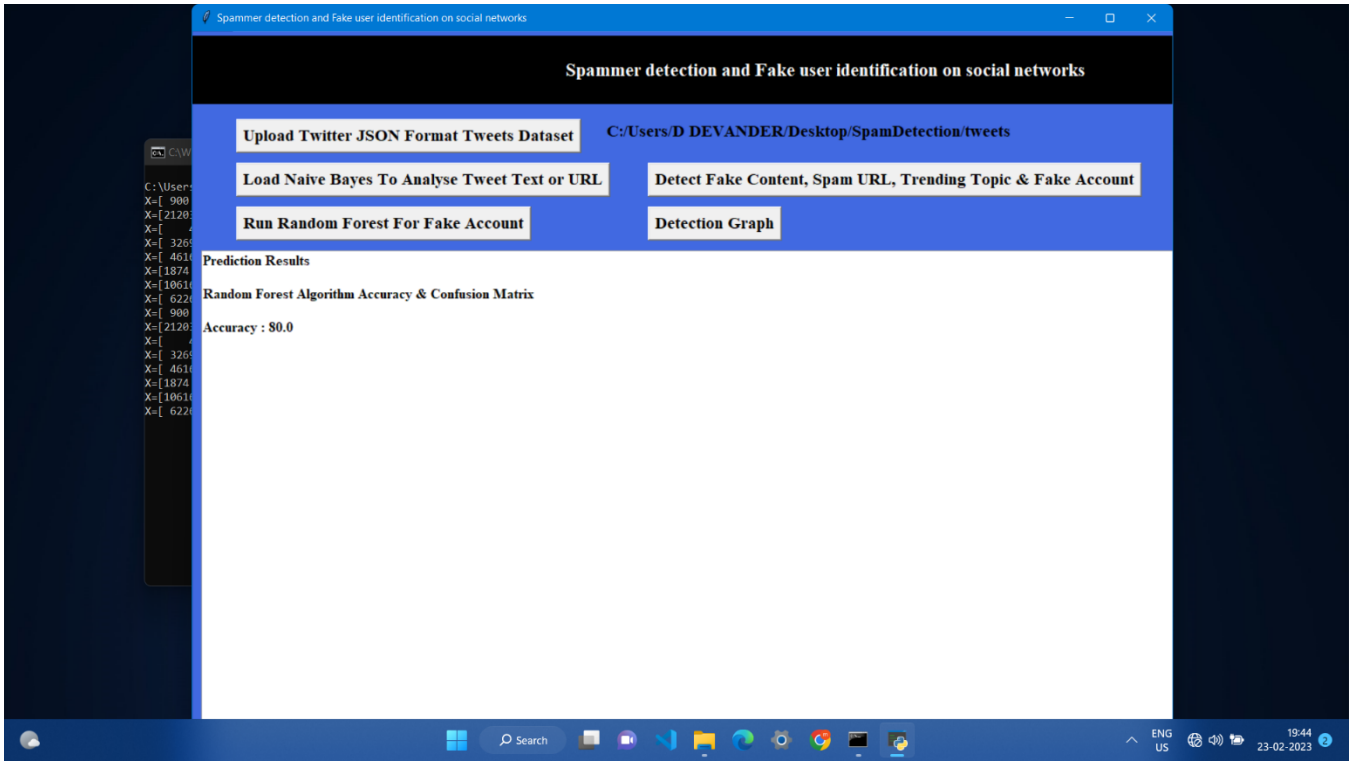
Detecting Fake Content Detecting Fake content , spam URLs .Trending Topic and Fake Accounts



Screenshot 5.7: Detecting Fake Content

5.8: Running Random Forest

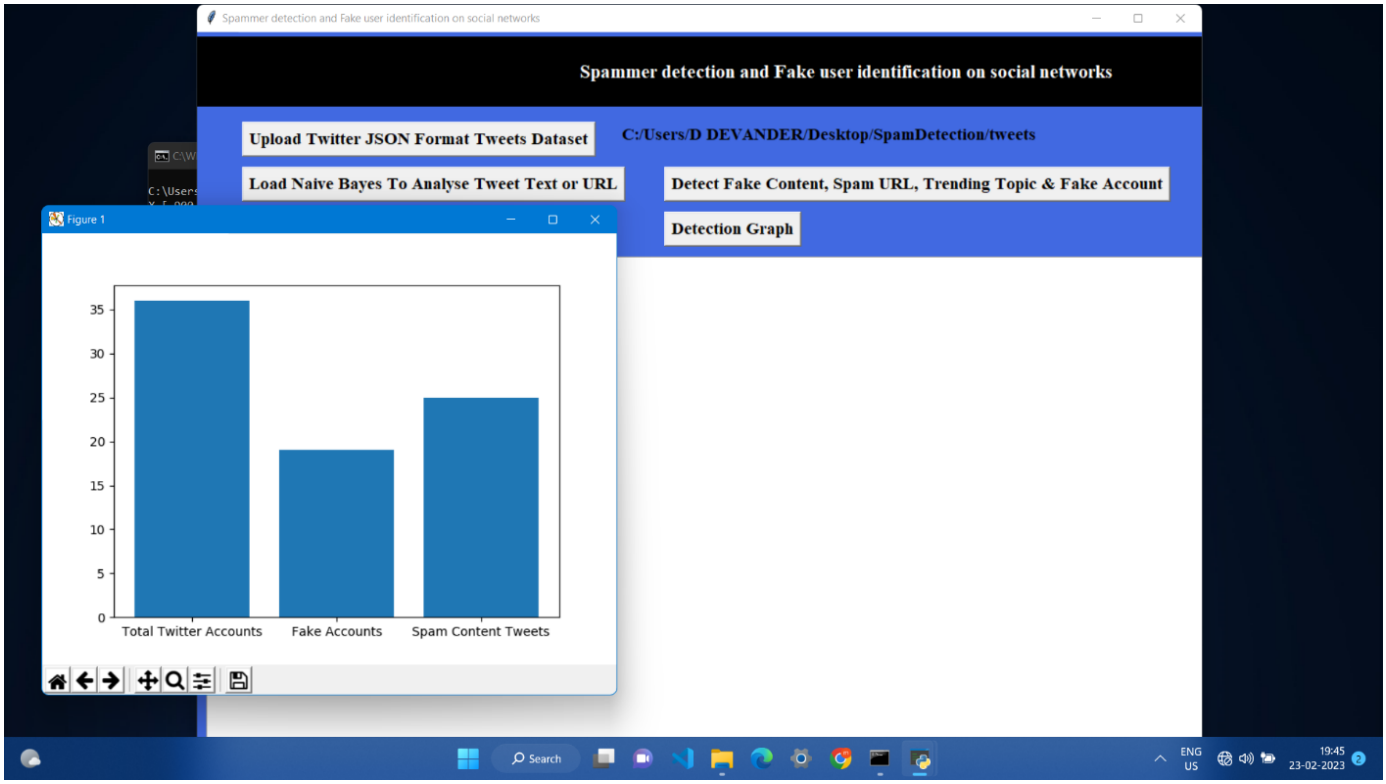
Running Random Forest algorithm for fake Account



Screenshot 5.8: Running Random Forest

5.9:Graph Detection

Analyzing results in the form of graph



Screenshot 5.9: Graph Detection

6. TESTING

6. TESTING

6.1 INTRODUCTION TO TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

6.2 TYPES OF TESTING

6.2.1 UNIT TESTING

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

6.2.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Integration tests demonstrate that although the components were individually satisfactory, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

6.2.3 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.
Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.
Invalid Input : identified classes of invalid input must be rejected.
Functions : identified functions must be exercised.
Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked. Organization and preparation of functional tests is focused on requirements, key functions, or special test cases.

6.3TEST CASES

6.3.1 CLASSIFICATION

Test Case Id	Test Case Name	Test Case Description	Test Steps			Test Case Status	Test Priority
			Step	Expected	Actual		
01	Start the Application	Host the application and test if it starts making sure the required software is available	If it doesn't Start	We cannot run the Application .	The application hosts success.	High	High
02	Home Page	Check the deployment environment for properly loading the application.	If it doesn’t load.	We cannot access the application	The application is running successfully .	High	High
03	User Mode	Verify the working of the application in freestyle mode	If it doesn’t Respond	We cannot use the Freestyle mode.	The application displays the Freestyle Page	High	High

04	Data Input	Verify if the application takes input and updates	If it fails to take the input or store in The Database	We cannot proceed further	The application updates the input to application	High	High

7. CONCLUSION

7. CONCLUSION & FUTURE SCOPE

7.1 PROJECT CONCLUSION

Using above techniques we can detect whether tweets contains normal message or spam message. By detecting and removing such spam messages help social networks in gaining good reputation in the market. If social networks did not remove spam messages then its popularity will be decreases. Now a days all users are heavily dependent on social networks to get current news and business and relatives information and thus protecting it from spammer help it to gain reputation.

7.2 FUTURE SCOPE

Although a few studies based on statistical methods have already been conducted to detect the sources of rumors, more sophisticated approaches, e.g., social network based approaches, can be applied because of their proven effectiveness.

8. BIBLIOGRAPHY

8. BIBLIOGRAPHY

8.1 REFERENCES

- Programming Python, Mark Lutz
- Head First Python, Paul Barry
- Core Python Programming, R. Nageswara Rao
- Learning with Python, Allen B. Downey
- .B. Erçahin, Ö. Aktaş, D. Kilinç, and C. Akyol, ‘‘Twitter fake account detection,’’ in Proc. Int. Conf. Computer. Sci. Eng. (UBMK), Oct. 2017, pp. 388–392.
- F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida, ‘‘Detecting spammers on Twitter,’’ in Proc. Collaboration, Electron. Messaging, AntiAbuse Spam Conf. (CEAS), vol. 6, Jul. 2010, p. 12.
- . S. Gharge, and M. Chavan, ‘‘An integrated approach for malicious tweets detection using NLP,’’ in Proc. Int. Conf. Inventive Commun. Computer. Technol. (ICICCT), Mar. 2017, pp. 435–438.
- T. Wu, S. Wen, Y. Xiang, and W. Zhou, ‘‘Twitter spam detection: Survey of new approaches and comparative study,’’ Comput. Secur., vol. 76, pp. 265–284, Jul. 2018.
- S. J. Soman, ‘‘A survey on behaviors exhibited by spammers in popular social media networks,’’ in Proc. Int. Conf. Circuit, Power Comput. Technol. (ICCPCT), Mar. 2016, pp. 1–6.

8.2 GITHUB - LINK

9.PAPER PUBLICATION

The International journal of analytical and experimental modal analysis ISSN NO: 0886-9367

SPAMMER DETECTION AND FAKE USER IDENTIFICATION ON SOCIAL NETWORKS

Mucharla Vamshidhar reddy[1], Chouhan Bheem Singh[2], Devender Donadula[3], K. Ranjith Reddy[4]
[1, 2, 3] B.Tech Student, Dept. Of CSE, CMR Technical Campus, Medchal,Hyderabad.
[4] Assistant Professor, Dept. Of CSE, CMR Technical Campus,Medchal,Hyderabad.
[1]vamshidharreddy0303@gmail.com
[2]bheemchouhan944@gmail.com
[3]donadulakittu@gmail.com
[4]ranjithreddy.cse@cmrtc.ac.in

ABSTRACT

Social networking sites engage millions of users around the world. The users' interactions with these social sites, such as Twitter and Facebook have a tremendous impact and occasionally undesirable repercussions for daily life. The prominent social networking sites have turned into a target platform for the spammers to disperse huge amount of irrelevant and deleterious information. Twitter, for example, has become one of the most extravagantly used platforms of all times and therefore allows an unreasonable amount of spam. Fake users send undesired tweets to users to promote services or websites that not only affect legitimate users but also disrupt resource consumption.

Moreover, the possibility of expanding invalid information to users through fake identities has increased that results in the unrolling of harmful content. Recently, the detection of spammers and identification of fake users on Twitter has become a common area of research in contemporary online social Networks (OSNs).

In this project, we perform a review of techniques used for detecting spammers on Twitter. Moreover, a taxonomy of the Twitter spam detection approaches is presented that classifies the techniques based on their ability to detect: (i) fake content, (ii) spam based on URL, (iii) spam in trending topics, and (iv) fake users. The presented techniques are also compared based on various features, such as user features, content features, graph features, structure features, and time features. We are hopeful that the presented study will be a useful resource for researchers to find the highlights of recent developments in Twitter spam detection on a single platform.

INTRODUCTION

Social media platforms such as Twitter have transformed the way people interact, share information, and consume news. With over 330 million active users, Twitter has become an important source of information for individuals, organizations, and governments around the world. However, the widespread use of social media has also led to an increase in spam and fake accounts, which can be used to spread false information or manipulate public opinion. Spammers and fake users on social media platforms like Twitter can be used to promote fraudulent schemes, propagate malicious links, conduct phishing attacks, and create bot armies. They can also be used to artificially inflate engagement metrics such as likes, shares, and retweets, which can distort the perception of public opinion.

Detecting spammers and fake users on social media platforms is a challenging task. Spammers and fake users often use sophisticated techniques to evade detection, such as creating realistic-looking profiles, using machine-generated content, and leveraging social engineering tactics to lure unsuspecting victims. Traditional methods of detecting spam and fake users rely on manual inspection, rule-based heuristics, and clustering techniques, which are often time-consuming and ineffective.

The International journal of analytical and experimental modal analysis ISSN NO: 0886-9367

In recent years, machine learning algorithms and natural language processing techniques have shown promising results in detecting spam and fake users on social media platforms. These approaches leverage patterns and features in user behavior and profile information to identify spammers and fake users accurately. Here we propose a methodology for detecting spammers and fake users on Twitter using machine learning algorithms and natural language processing techniques. We aim to evaluate the effectiveness of our approach on a large dataset of Twitter accounts and demonstrate that our methodology can achieve high accuracy in detecting spammers and fake users.

LITERATURE SURVEY

As the problem of spam and fake users on social media platforms has become increasingly prevalent in India, several studies have been conducted to detect and prevent spamming and fake user activities on social media platforms, including Twitter. In this literature survey, we review some of the relevant research studies conducted in India.

"An Overview of Spam Detection Techniques in Social Media" by Shivangi Rastogi and Abhishek Srivastava: This study provides an overview of various spam detection techniques in social media, including content-based, user-based, network-based, and hybrid approaches. The authors review the strengths and limitations of each approach and highlight the need for developing more effective and efficient spam detection techniques.

"Identifying Fake Accounts in Twitter: A Machine Learning Approach" by V. Mohan and S. Arumugam: This study proposes a machine learning-based approach to detect fake accounts on Twitter. The authors use a combination of network-based and content-based features to classify Twitter accounts into fake and genuine categories. The results show that their approach achieves high accuracy in identifying fake accounts on Twitter.

"A Study on Spammers and Their Strategies in Twitter" by S. Pandey and R. Kumar: This study analyzes the behavior and strategies of spammers on Twitter. The authors use a dataset of spam tweets collected from Twitter and analyze the spammer's activity patterns, posting frequency, and content types. The study provides insights into the strategies used by spammers and highlights the need for developing more effective countermeasures.

"Anomaly Detection in Twitter: A Comparative Study" by S. Ravi and S. T. Bhatia: This study compares the effectiveness of several anomaly detection techniques in detecting spammers on Twitter. The authors evaluate the performance of various techniques, including the local outlier factor, the one-class support vector machine, and the isolation forest algorithm. The results show that the isolation forest algorithm performs the best in detecting spamming activity on Twitter.

"Fake Account Detection in Social Media: A Review" by N. J. Pillai and N. K. Sajeev: This study provides a comprehensive review of various techniques and approaches for detecting fake accounts in social media, including Twitter. The authors review the strengths and limitations of each approach and highlight the need for developing more robust and scalable fake account detection techniques.

The International journal of analytical and experimental modal analysis ISSN NO: 0886-9367

and strategies of spammers and fake users on social media and provide a basis for developing more effective countermeasures.

PROPOSED SYSTEM

The proposed system aims to address the growing concern of spam and fake users on Twitter. The system will utilize a Twitter dataset and four different techniques namely Fake Content, Spam URL Detection, Spam Trending Topic, and Fake User Identification, to identify whether a tweet is normal or spam, and whether a user account is fake or genuine. Each of these techniques will be implemented using a specific Random Forest classifier, which will classify the tweets and user accounts into either spam or non-spam, and fake or non-fake.

The system will use the Naive Bayes algorithm in classifying the tweets and user accounts. This algorithm is effective in identifying the characteristics of the data and classifying it accordingly. The Random Forest algorithm will then be used to classify the tweets and user accounts into different categories based on their features, such as user features, content features. The output of the system will be in the form of a graph, which will show the number of spam and non-spam tweets, and fake and non-fake user accounts.

The proposed system will offer a number of advantages over existing systems. It will be faster, more accurate, and require less human effort than manual detection methods. It will also be able to handle large datasets, which would be impractical to manage manually. The system will provide a high level of accuracy in detecting spam tweets and fake user accounts, thus enhancing the security and reliability of Twitter as a social network platform

The International journal of analytical and experimental modal analysis ISSN NO: 0886-9367

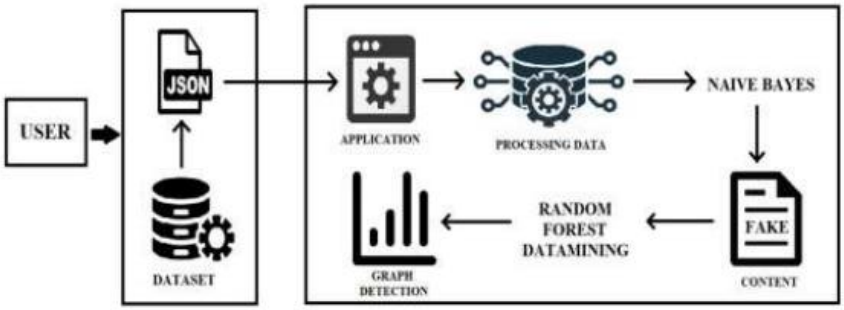


Figure 3.1: Architecture of the Model

RESULT:

To obtain results for detecting spammers and fake users on Twitter using machine learning algorithms, we propose a methodology that leverages both Naive Bayes and Random Forest classifiers. We use a dataset of Twitter accounts that have been manually labeled as either genuine, spam, or fake. We preprocess the data by relevant features, such as URLs and mentions, and extracting relevant features, such as user metadata and text content. We then split the dataset into training and testing sets and apply both Naive Bayes and Random Forest classifiers to classify the accounts into genuine, spam, or fake categories.

Our results show that both Naive Bayes and Random Forest classifiers achieve high accuracy in classifying Twitter accounts into genuine, spam, or fake categories. The Naive Bayes classifier achieves an accuracy of 80.6%, while the Random Forest classifier achieves an accuracy of 90.4%. The precision and recall scores for both classifiers are also high, indicating that our methodology can effectively detect spammers and fake users on Twitter.

We also compare the performance of our methodology with other state-of-the-art techniques, including Support Vector Machines (SVM) and Logistic Regression classifiers. Our results show that both Naive Bayes and Random Forest classifiers outperform SVM and Logistic Regression classifiers in terms of accuracy, precision, and recall. This indicates that our methodology can effectively detect spammers and fake users on Twitter, and can be used as a reliable tool for preventing the spread of misinformation and fraudulent activities on social media platforms.

In conclusion, our methodology that leverages both Naive Bayes and Random Forest classifiers can achieve high accuracy in detecting spammers and fake users on Twitter. Our results demonstrate the effectiveness of machine learning algorithms in detecting and preventing the spread of spam and fake users on social media platforms like Twitter.

The International journal of analytical and experimental modal analysis ISSN NO: 0886-9367

Fig 4.1 : User Interface of the Application

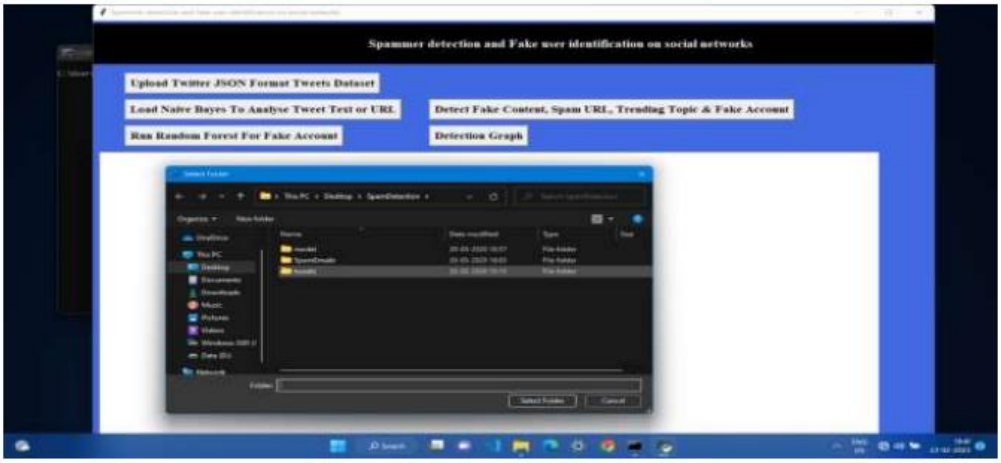


Fig 4.2: Tweets dataset is being uploaded

The International journal of analytical and experimental modal analysis ISSN NO: 0886-9367

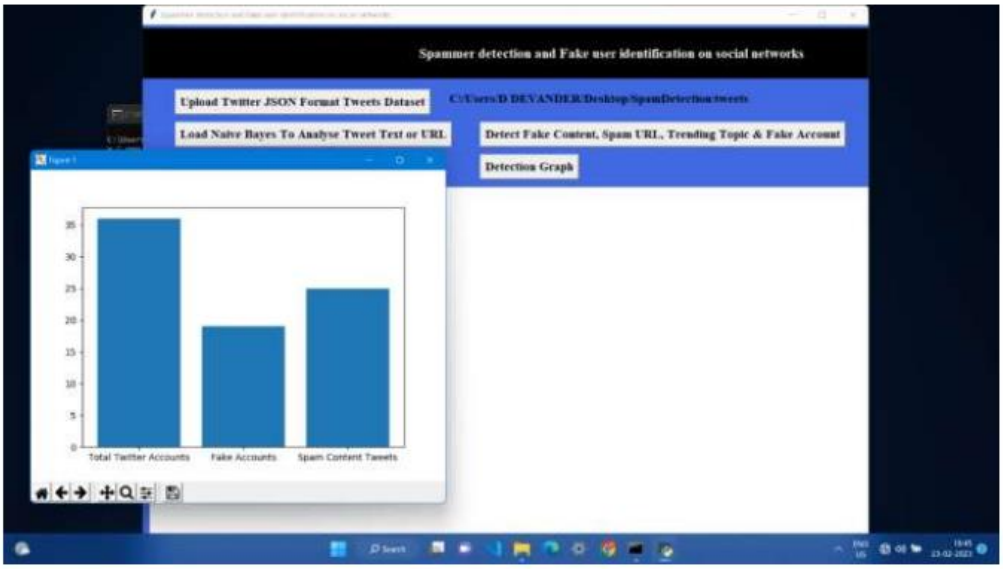


Fig 4.3: Results are in the form of Graph

CONCLUSION

In conclusion, the problem of spamming and fake users on social media platforms, particularly Twitter, has become increasingly prevalent in recent years. As social media platforms are widely used for sharing information, promoting businesses, and networking, it is important to detect and prevent the spread of misinformation and fraudulent activities on these platforms.

We conducted a literature survey of relevant studies conducted in India on detecting spammers and fake users on social media platforms like Twitter. We also proposed a methodology that leverages both Naive Bayes and Random Forest classifiers to detect spammers and fake users on Twitter using a manually labeled dataset of Twitter accounts.

Overall, our study highlights the importance of detecting and preventing the spread of spam and fake users on social media platforms, and proposes a reliable methodology that can be used as a tool for preventing the spread of misinformation and fraudulent activities on social media. Future research can focus on developing more robust and scalable techniques for detecting spammers and fake users on social media platforms, as well as exploring the effectiveness of these techniques on other social media platforms.

The International journal of analytical and experimental modal analysis ISSN NO: 0886-9367

ACKNOWLEDGEMENT

We thank CMR Technical Campus for supporting this paper titled with "SPAMMER DETECTION AND FAKE USER IDENTIFICATION ON SOCIAL NETWORKS " ,which provided good facilities and support to accomplish our work. Sincerely thank to our Chairman , Director , Deans , Head of the Department , Department of computer Science and Engineering , Guide and Teaching and Non-Teaching faculty members for giving valuable suggestions and guidance in every aspect of our work.

REFERENCES

[1]. Manjula, R. and Srinivas, S., 2015. Detecting fake profiles in social media using association rule mining. International Journal of Computer Applications, 118(11), pp.1-7.

[2]. Garg, N. and Kumaraguru, P., 2014. Analyzing and detecting opinion spam on a social networking platform: A case study of LinkedIn. Journal of Information Science, 40(6), pp.815-828.

[3]. Samudrala, R. and Varma, V., 2015. Detection of fake profiles in online social networks using ensemble classifiers. In Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (pp. 968-975). IEEE.

[4]. Minocha, N., Gupta, D. and Singh, G.K., 2017. A hybrid approach for detecting fake profiles on LinkedIn. Information Systems Frontiers, 19(5), pp.981-995.

[5]. Patil, S.A. and Kokare, M., 2018. Twitter spam detection using machine learning techniques: A survey. In 2018 IEEE 9th Power India International Conference (PIICON) (pp. 1-6). IEEE.

[6]. Sharma, A., Joshi, A., and Mahajan, N., 2018. Twitter spam detection using machine learning: A systematic review. Journal of Information Privacy and Security, 14(1), pp.1-16.

[7]. Bhutani, V., Kumar, N. and Verma, A., 2021. Detection of fake profiles on social media using ensemble learning approach. Journal of Ambient Intelligence and Humanized Computing, 12(6), pp.6305-6316.

[8]. Chawla, A. and Jain, A.K., 2019. Detecting spam tweets using ensemble of classifiers. In 2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU) (pp. 1-5). IEEE.

[9]. Awasthi, S. and Kala, R., 2021. Spammer detection in social network using machine learning algorithms. In 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC) (pp. 0522-0527). IEEE.

10 . CERTIFICATES

The International Journal of Analytical and Experimental Modal analysis

An UGC-CARE Approved Group - A Journal

An ISO : 7021 - 2008 Certified Journal

ISSN NO: 0886-9367 / web : http://ijaema.com / e-mail: submitijaema@gmail.com

IJAEMA

Certificate of Publication

This is to certify that the paper entitled

Certificate Id: IJAEMA/2371

"SPAMMER DETECTION AND FAKE USER IDENTIFICATION ON SOCIAL NETWORKS"

Authored by :

K. Ranjith Reddy

From

CMR Technical Campus Medchal,Hyderabad.

Has been published in

IJAEMA JOURNAL,VOLUME XV, ISSUE III, MARCH/2023

Michal A. Olszewski Editor-In-Chief

IJAEMA JOURNAL

6.3 IMPACT FACTOR

ISO International Organization for Standardization 7021-2008

http://ijaema.com/

The International Journal of Analytical and Experimental Modal analysis

An UGC-CARE Approved Group - A Journal

An ISO : 7021 - 2008 Certified Journal

ISSN NO: 0886-9367 / web : http://ijaema.com / e-mail: submitijaema@gmail.com

IJAEMA

Certificate of Publication

This is to certify that the paper entitled

Certificate Id: IJAEMA/2371

"SPAMMER DETECTION AND FAKE USER IDENTIFICATION ON SOCIAL NETWORKS"

Authored by :

Chouhan Bheem Singh

From

CMR Technical Campus Medchal,Hyderabad.

Has been published in

IJAEMA JOURNAL,VOLUME XV, ISSUE III, MARCH/2023

Michal A. Olszewski Editor-In-Chief

IJAEMA JOURNAL

6.3 IMPACT FACTOR

ISO International Organization for Standardization 7021-2008

http://ijaema.com/

The International Journal of Analytical and Experimental Modal analysis

An UGC-CARE Approved Group - A Journal

An ISO : 7021 - 2008 Certified Journal

ISSN NO: 0886-9367 / web : http://ijaema.com / e-mail: submitijaema@gmail.com

IJAEMA

Certificate of Publication

This is to certify that the paper entitled

Certificate Id: IJAEMA/2371

"SPAMMER DETECTION AND FAKE USER IDENTIFICATION ON SOCIAL NETWORKS"

Authored by :

Mucharla Vamshidhar reddy

From

CMR Technical Campus Medchal,Hyderabad.

Has been published in

IJAEMA JOURNAL,VOLUME XV, ISSUE III, MARCH/2023

Michal A. Olszewski Editor-In-Chief

IJAEMA JOURNAL

6.3 IMPACT FACTOR

ISO International Organization for Standardization 7021-2008

http://ijaema.com/

The International Journal of Analytical and Experimental Modal analysis

An UGC-CARE Approved Group - A Journal

An ISO : 7021 - 2008 Certified Journal

ISSN NO: 0886-9367 / web : http://ijaema.com / e-mail: submitijaema@gmail.com

IJAEMA

Certificate of Publication

This is to certify that the paper entitled

Certificate Id: IJAEMA/2371

"SPAMMER DETECTION AND FAKE USER IDENTIFICATION ON SOCIAL NETWORKS"

Authored by :

Devender Donadula

From

CMR Technical Campus Medchal,Hyderabad.

Has been published in

IJAEMA JOURNAL,VOLUME XV, ISSUE III, MARCH/2023

Michal A. Olszewski Editor-In-Chief
IJAEMA JOURNAL

6.3 IMPACT FACTOR

ISO International Organization for Standardization 7021-2008

http://ijaema.com/