A Project Report

on

Analysis of Twitter accounts for detecting bots and humans using Machine Learning

by

1. Tanish Sanghavi
2. Shubham Shete
3. Aman Sarawgi

under the guidance of

Prof. Anita A. Lahane

Department of Computer Engineering

University of Mumbai

April - 2020

MANJARA CHARITABLE TRUST

RAJIV GANDHI INSTITUTE OF TECHNOLOGY, MUMBAI

Juhu-Versova Link Road Versova, Andheri(W), Mumbai-53.

Certificate

Department of Computer Engineering

This is to certify that

1. Tanish Sanghavi
2. Shubham Shete
3. Aman Sarawgi

Have satisfactory completed this project entitled

Analysis of Twitter accounts for detecting bots and humans using Machine Learning

Towards the partial fulfillment of the
BACHELOR OF ENGINEERING
IN
(COMPUTER ENGINEERING)
as laid by University of Mumbai.

Guide **Head of Department**

Prof. Anita A. Lahane **Dr.Satish Y. Ket**

Principal
Dr.Sanjay Bokade

Project Report Approval for B. E.

This project report entitled **Analysis of Twitter accounts for detecting bots and humans using Machine Learning** by **Tanish Sanghavi,Shubham Shete,Aman Sarawgi** is approved for the degree of Computer Engineering.

Examiners

1. ———————————

2. ———————————

Date:

Place:

Declaration

We wish to state that the work embodied in this project titled "Analysis of Twitter accounts for detecting bots and humans using Machine Learning" forms our own contribution to the work carried out under the guidance of "Prof. Anita A. Lahane" at the Rajiv Gandhi Institute of Technology.

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. we also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. we understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Students Signatures)	**(Students Signatures)**	**(Students Signatures)**
Tanish Sanghavi (B815)	**Shubham Shete (B818)**	**Aman Sarawgi (B849)**

Abstract

Twitter is a web application playing dual roles of online social networking and microblogging. Users communicate with each other by publishing text-based posts. In Contrast, many examples exist of cases where fake accounts created by bots or computers have been detected successfully using machine learning models. Initially,the biggest problem was spam. However, as Twitter became an important tool for protest, political conversation, and mobilization, the possibilities for harm increased significantly.

Twitter popularity has fostered the emergence of a new spam marketplace. The services that this market provides include: the sale of fraudulent accounts, affiliate programs that facilitate distributing Twitter spam, as well as a cadre of spammers who execute large scale spam campaigns. In addition, twitter users have started to buy fake followers of their accounts. In this project we present machine learning algorithms we have used to detect fake followers in Twitter.We identified a number of characteristics that distinguish fake and genuine followers. We used these characteristics as attributes to machine learning algorithms to classify users as fake or genuine.

Contents

List of Figures

List of Tables

List of Algorithms

- **S**pearman Correlation
- **D**ecision Tree Classifier
- **R**andom forest
- **K** Nearest Neighbor
- **S**upport Vector Machine

List of Acronyms

- **R**OC-Reciever Operating Character
- **S**VM-Support Vector Machine
- **K**NN-K Nearest Neighbor

List of Symbols

- $+$: Addition
- $=$: Assigning a value
- $==$: Checking Equality
- **-** : Subtraction

Chapter 1

Introduction

1.1 Introduction

Twitter has become a popular media hub where people can share news, jokes and talk about their moods and discuss news events. In Twitter users can send Tweets instantly to his/her followers. Also, Tweets can be retrieved using Twitter's real time search engine. The ranking of tweets in this search engine depends on many factors, one of which is the user's number of followers. Twitter's popularity has made it an attractive place for spam and spammers of all types. Spammers have various goals: spreading advertising to generate sales, phishing or simply just compromising the system's reputation. Given that spammers are increasingly arriving on twitter, the success of real time search services and mining tools lies in the ability to distinguish valuable tweets from the spam storm .There are various ways to fight spam and spammers such as URL blacklists, passive social networking spam traps, manual classification to generate datasets used to train a classifier that later will be used to detect spam and spammers.

So what is Twitter spam? As Twitter describes it in their website , Twitter spam is "a varietyof prohibited behaviors that violate the Twitter Rules." Those rules include among other things the type of behavior Twitter considers as spamming, such as:

- **P**osting harmful links (including links to phishing or malware).
- **A**ggressive following behavior (mass following and mass unfollowing for attention), particularly by automated Means.
- **A**busing the @reply or @mention function to post unwanted messages to users.
- **H**aving a small number of followers compared to the number of people one is following.
- **P**osting repeatedly to trending topics to try to grab attention.

Twitter actually fights spammers by suspending their accounts. But in general OSN (Online Social Networking) sites do not detect and suspend suspicious user accounts quickly.

They are not willing to deploy automated methods to detect and remove spam accounts fearing that this will lead to a serious discontentment among users. Thus, they wait until a sufficient number of users report a specific account as a spam account to suspend it. However, legitimate users are unwilling to invest time to report spammers. Hence spammers are allowed more time to spread spam.

Automated accounts, called bots, are common in social media. Although all bots are not bad, bots are easy means to engage in unethical and illegal activities in social media. Examples of such activities include selling accounts, spamming inappropriate content, and participating in sponsored activities. Many social metrics are calculated based on social media data.The significant presence of bots in social media will make many of these metrics useless. The exact number of bots is dynamic and unknown. The range of the estimates is between 3 percent to 7 percent. Social media sites, such as Twitter, regularly suspend abusive bots. Yet, the number of bots is growing because of almost zero-cost in creating new bots. Existing bot detection methods are not capable of fighting such evolving set of bots. There are several reasons. Current methods are mostly non-adaptive, require supervised training, and consider accounts independently. Typical features used in some of the methods need a long duration of activities (e.g. weeks) which makes the detection process useless, as the bots can initiate a fair amount of harm before being detected. Moreover, bots are becoming smarter. They mimic humans to avoid being detected and suspended and increase throughput by creating many accounts. We take a novel unsupervised approach of cross-correlating account activities, that can detect such dynamic bots as soon as two hours after starting their activities.

1.2 Organization of Report

Describe every chapter

- **Ch.1 Introduction:** This chapter contains the introduction about the twitter, what is bot?, other information on bots and bot activities.
- **Ch.2 Review of Literature:** It gives the Survey Existing system, Limitation Existing system or research gap, Problem Statement, objectives and scope.It gives the details of Hardware and Software,Design Details and Methodology.
- **Ch.3 Proposed System:** The proposed System has the main task to detect the twitter account user is genuine or bot. And this task is carried out by different algorithms.Here we are using 3 to 4 different algorithms to compare the accuracy of the system.It gives the details of Hardware and Software,Design Details and Methodology.
- **Ch.4 Implementation Plan:** Explanation of different algorithms going to use. It gives the working information in the form of gantt chart.
- **Ch.5 Conclusion:** we used Machine Learning techniques to predict weather an account on Twitter is a Bot or a real user.

Chapter 2

Literature Review

The prevalence of fake accounts and/or bots is continuously evolving, and feature based machine-learning detection systems employing highly predictive behaviour provide unique opportunities to develop an understanding of how to discriminate between bots and humans, i.e. between real vs. fake accounts on social media.

2.1 Survey of Existing system

In (Arzum Caratas, Serap Sahin, 2017) the social bot detection techniques, it is seen that the higher social bot detection rates (over 80 percent) are obtained with the combination of the structure-based properties of OSN and unsupervised machine learning methods. It is useful to conduct research on some possible approaches to increase the detection rate. The approaches may be (i) use of autonomous-intelligent agent based and (ii) identification-based approaches as the future directions of researches.

In (ESTÉE VAN DER WALT, 2017), it is shown that the engineered features that were previously used to detect fake accounts generated by bots are not similarly successful in the detection of fake accounts generated by humans. This paper reports on a study that focused on detecting fake accounts created by humans, as opposed to those created by bots. We investigated whether the results from past studies to detect bot accounts could be applied successfully to detect fake human accounts. A corpus of human accounts was enriched with engineered features that had previously been used to successfully detect fake accounts created by bots.citeben.

In (Supraja Gujarala, Brian Hudson, 2016),we have presented a machine learning pipeline for detecting fake accounts in online social networks. Rather than making a prediction for each individual account, our system classifies clusters of fake accounts to determine whether they have been created by the same actor. Our evaluation on both in-sample and out-of-sample data showed strong performance, and we have used the system in production to find and restrict more than 250,000 accounts.In this work we evaluated our framework on clusters created by simple grouping on registration date and registration IP address. In future work we expect to run our model on clusters created by grouping on other features, such as ISP or company, and other time periods, such as week or month. [**?**].

In (Cao Xiao, Theodore Hwa,2015), We trained models using random forest, logistic regression, and support vector machine classifiers. We evaluated the classifiers' performance with 80-20 split in-sample testing and out-of-sample testing with a more recent data set. The latter test is a better approximation of real-life performance, since models are trained on data

from the past and run on data from the present.To measure the classifiers' performance, we computed AUC (area under the ROC curve) and recall at 95 percent precision. In practice the desired precision rates and thresholds for classification may be higher or lower depending on business needs and the relative cost of false positives and false negatives. We found that the random forest algorithm provided the best results for all metrics. On the held-out test set, the random forest model produced AUC 0.98 and recall 0.90 at 95 percent precision. When run on out-of-sample testing data the random forest model again performed best, with AUC 0.95 and recall 0.72 at 95 percent precision.

In (Z. Chu, S. Gianvecchio, 2012), In this section, we first evaluate the accuracy of our classification system based on the ground truth set that includes both the training set are inputted into the classifier. LDA generates a weight table to achieve the maximum accuracy. In other words, it includes as many users as possible, whose classified class matches actual class. The weights are then used by the decision maker to classify users.

Year Of Publication	Title Of Paper	Name Of The Authors	Findings	Drawbacks	Future Scope
2017	A review on social bot detection techniques	Arzum Karatas, Serap Sahin	The success of structure based Sybil detection	Detection of boots on OSN as a challenging issue	Use of intelligent based approaches to avoid issues
2017	Using machine learning to detect fake identities: Bots vs Humans	Estee Van Der Walt, Jan Eloff	To observe social media platforms to detect bot activities	The activity needs to take place before its detected	To use certain algorithms to detect bot activities sooner
2016	Profile characteristics of fake Twitter account	Supraja Gujarala, Brian Hudson	Ground truth data set should be obtained to certify authentic Twitter account	A large data set is required ie – in millions to get accurate results	To use cluster sets in order to increase data efficiency
2015	Detecting Clusters of Fake Accounts in Online Social Networks	Cao Xiao, Theodore Hwa	Making prediction in cluster to identify if made accounts made by the same operator	Huge framework on clustering is required which is time consuming	To reduce the consumption in time while analysing the clusters
2012	Detecting automation of Twitter accounts: Are you a human, bot or cyborg	Zi Chu, S Gianvecchio	To observe the problem of automation by bots and cyborgs	Bots passes through because of lack of feature detections	To improve decision making strategy

Figure 2.1: Literature Survey

2.2 Limitation Existing system or research gap

Account-based features are lightweight enough to be used detecting real-time spam which requires instant analysis. The number of lists the user is a member of can be considered a useful metric to detect spammers since it is an obvious sign of the user's impact on others but it is open to manipulation by creating fake lists and adding the fake accounts which are under the CC infrastructure into these lists. Account-based features are lightweight enough to be used detecting real-time spam which requires instant analysis but they can be easily manipulated by spammers.

2.3 Problem Statement and objectives

Account-based features are lightweight enough to be used detecting real-time spam which requires instant analysis. The number of lists the user is a member of can be considered a useful metric to detect spammers since it is an obvious sign of the user's impact on others but it is open to manipulation by creating fake lists and adding the fake accounts which are under the CC infrastructure into these lists. Account-based features are lightweight enough to be used detecting real-time spam which requires instant analysis but they can be easily manipulated by spammers.

2.4 Future Scope

Bots are not only issue of twitter or any social network but is has become one of the big issue of internet. In recent years, the Internet has enabled access to widespread remote services in the distributed computing environment; however, integrity of data transmission in the distributed computing platform is hindered by a number of security issues. The botnet phenomenon supports a wide range of criminal activities, including DDoS attacks, click fraud, phishing, malware distribution, spam emails, and building machines for illegitimate exchange of information/materials. We can implement the same for other social media networks to detect bots by making little changes in the program. The system can again be evolved to detect bots in real-time as well as its spam activities.

Chapter 3

Proposed System

The proposed System has the main task to detect the twitter account user is genuine or bot. And this task is carried out by different algorithms.Here we are using 3 to 4 different algorithms to compare the accuracy of the system. For this the data set is processed and divided into training data and testing data and relation between the attributes in the dataset is also found.

3.1 Details of Hardware and Software

3.1.1 Software Requirements

- Operating System: Windows 7 or above
- Coding Language: Python
- Microsoft Excel

3.1.2 Hardware Requirements

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

3.2 Design Details

- The data set was taken from Tweepy and divided it into training set and test set and then applying the different data analysis techniques to extract the features of the dataset.
- Then applying different machine learning algorithms to predict the specific twitter user account is bot or nonbot(human).

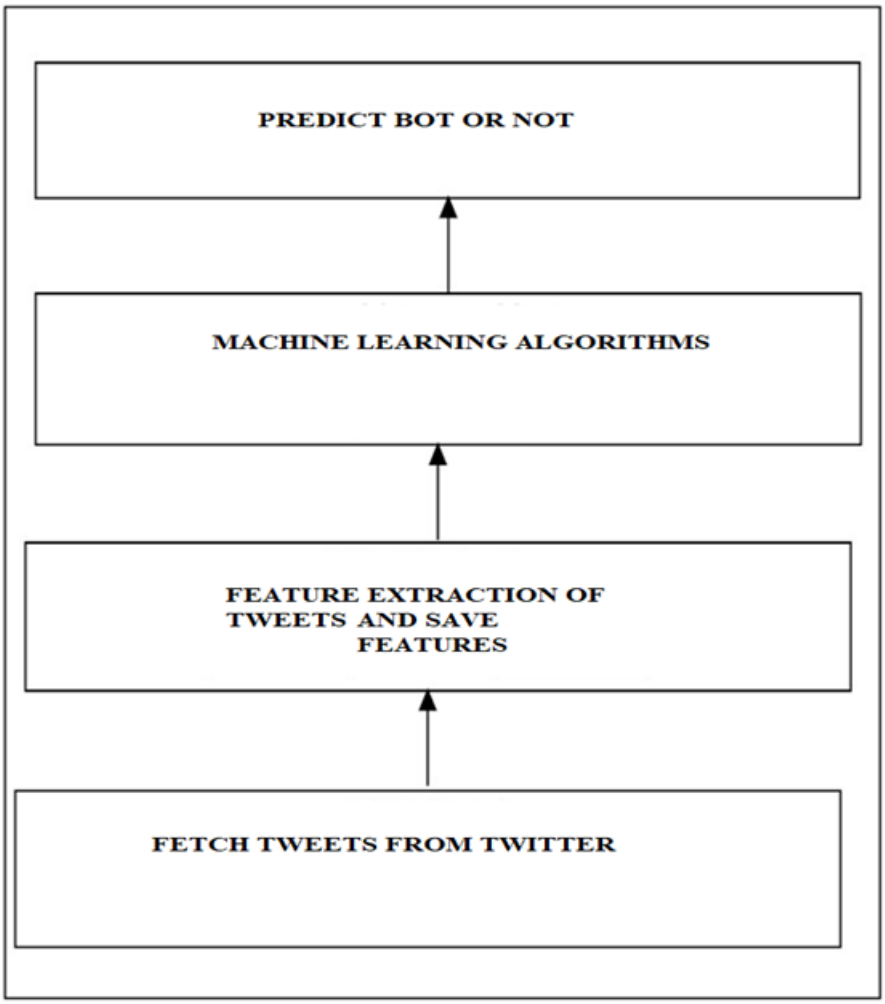

Figure 3.1: The Proposed System/System Architecture

3.2.1 Data flow diagram

- The data set was taken from Tweepy and divided it into training set and test set ,Using machine learning algorithm and training data we've trained our classifier model. Test set is used on classifier model for giving prediction according to the given scenario.
- In order to use machine learning to identify fake twitter accounts, we needed a labelled collection of users, pre classified as fake or genuine. The dataset is processed and divided into 70 percent(training set) and 30 percent(test set) on which data exploratory analysis has been done as well as it is also explored to feature extraction and feature engineering, Both training and test set are saved in .CSV format.

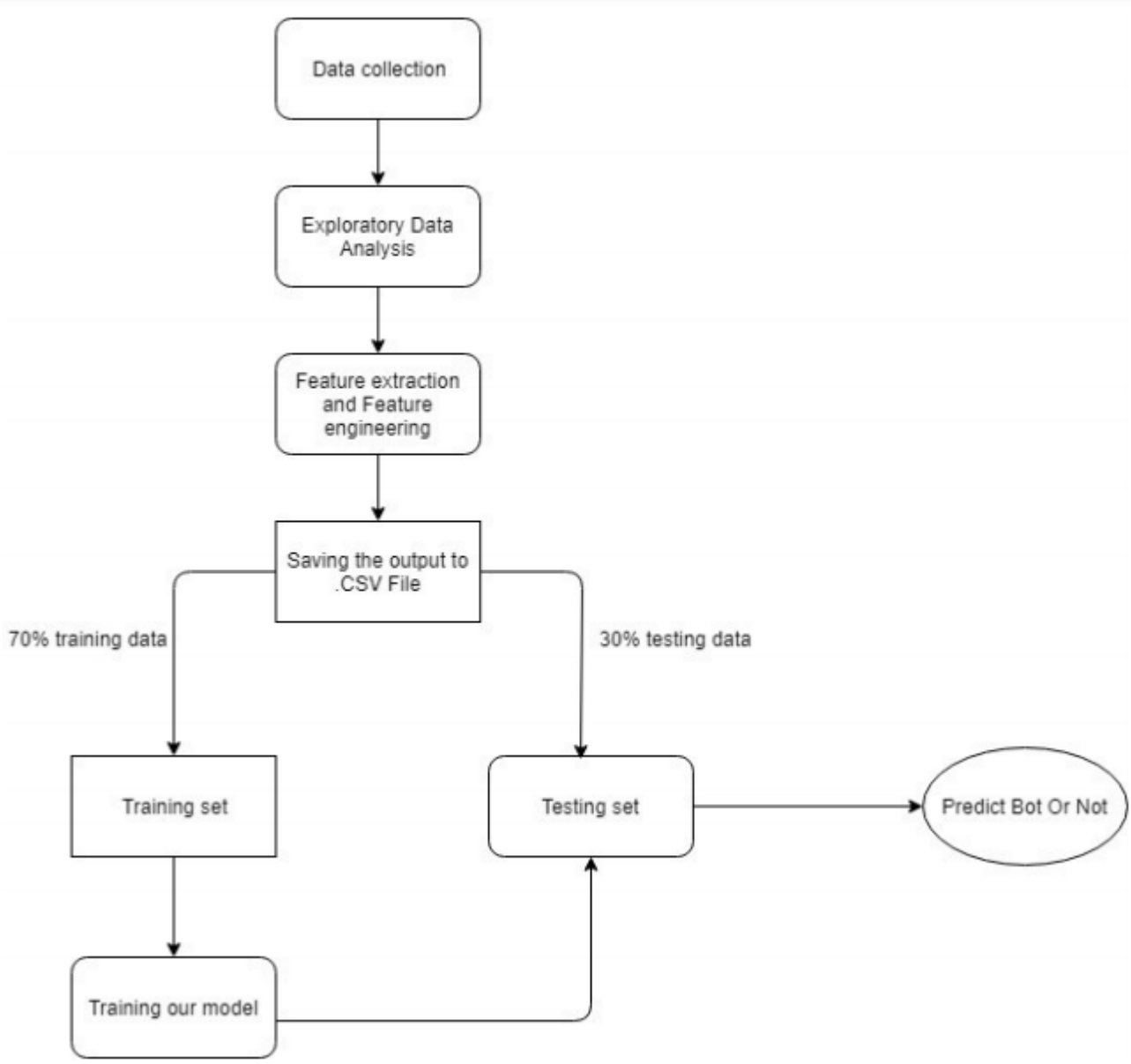

Figure 3.2: Data Flow Diagram

3.2.2 Sequence diagram

The main training data is given to different data analysis techniques and the result is store in feature.csv file format . This data is then used for further prediction using different machine learning algorithms. Then the data is store in final output file where the accounts are differentiate as bots and human(non-bots).

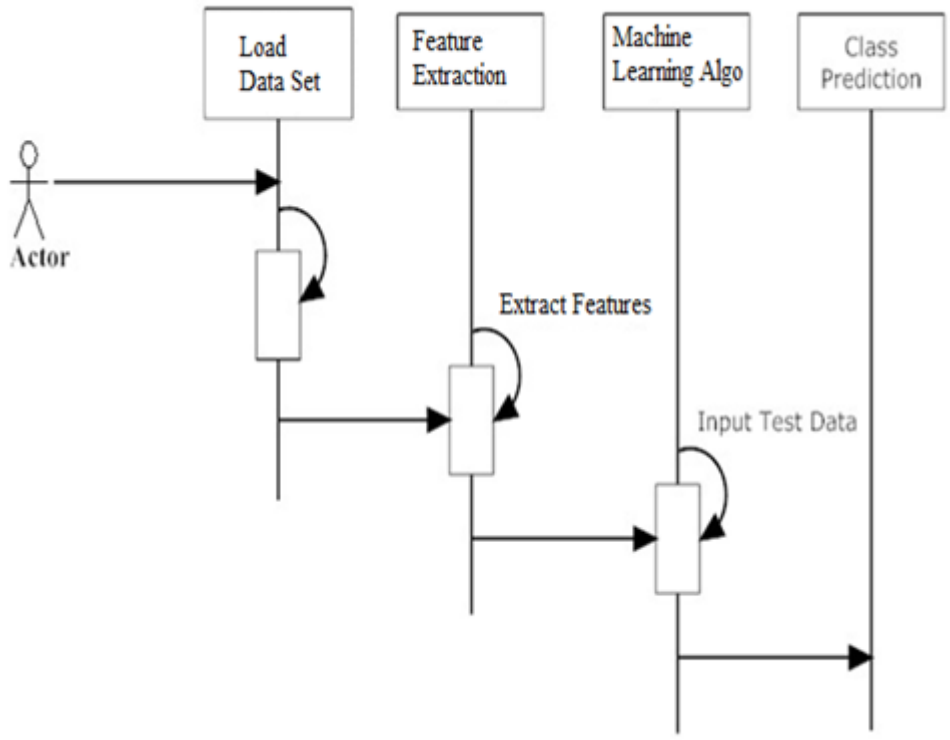

Figure 3.3: Sequence Diagram

3.3 Methodology/Procedures

3.3.1 Heat map

In this graph the null values are displayed with yellow color and the filled values with violet color. higher the number of null values, higher the chance of the account being a bot the contrast of the number of yellow patches show the difference between bot and humans this heat chart gives us a basic idea of how details in an account can help us identify if an specific account comes under being a bot or a human.

3.3.2 Friends to follower relation

In the relation of the followers count to the number of friends count. from the graph we see, bots have a huge number of followers in comparison to the number of friends the majority of points in the above graph are towards the beginning whereas, in the graph below, the points are more scattered and well spread.

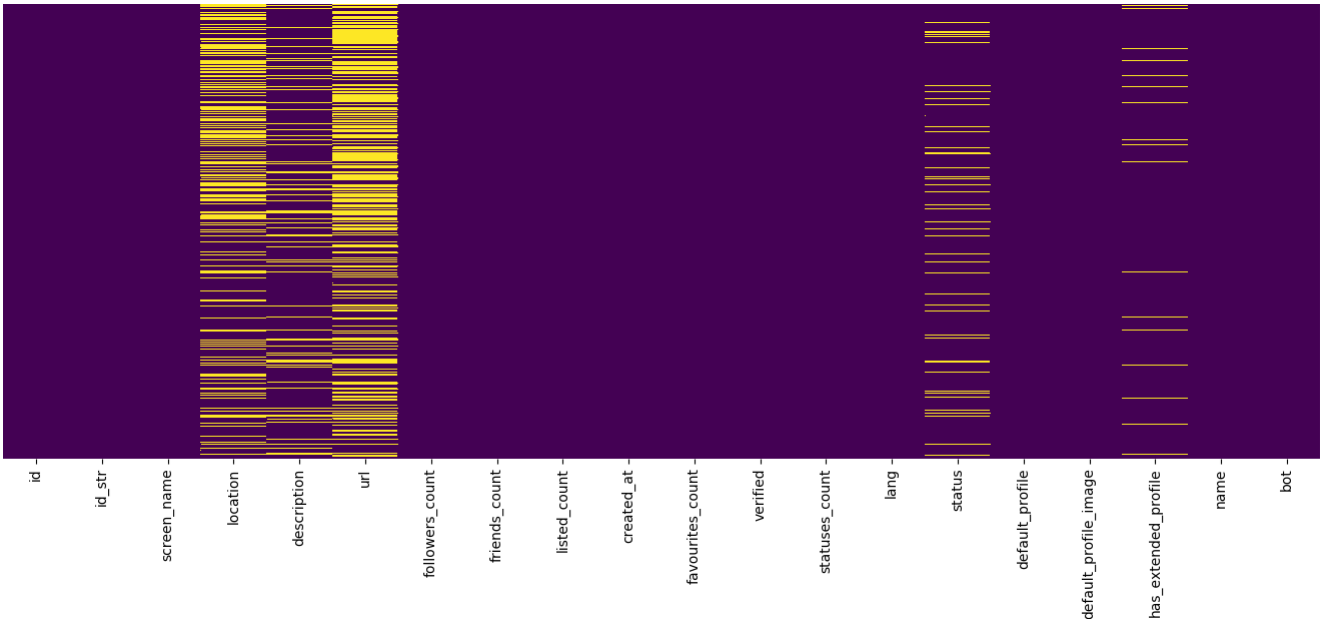

Figure 3.4: Heat map representation

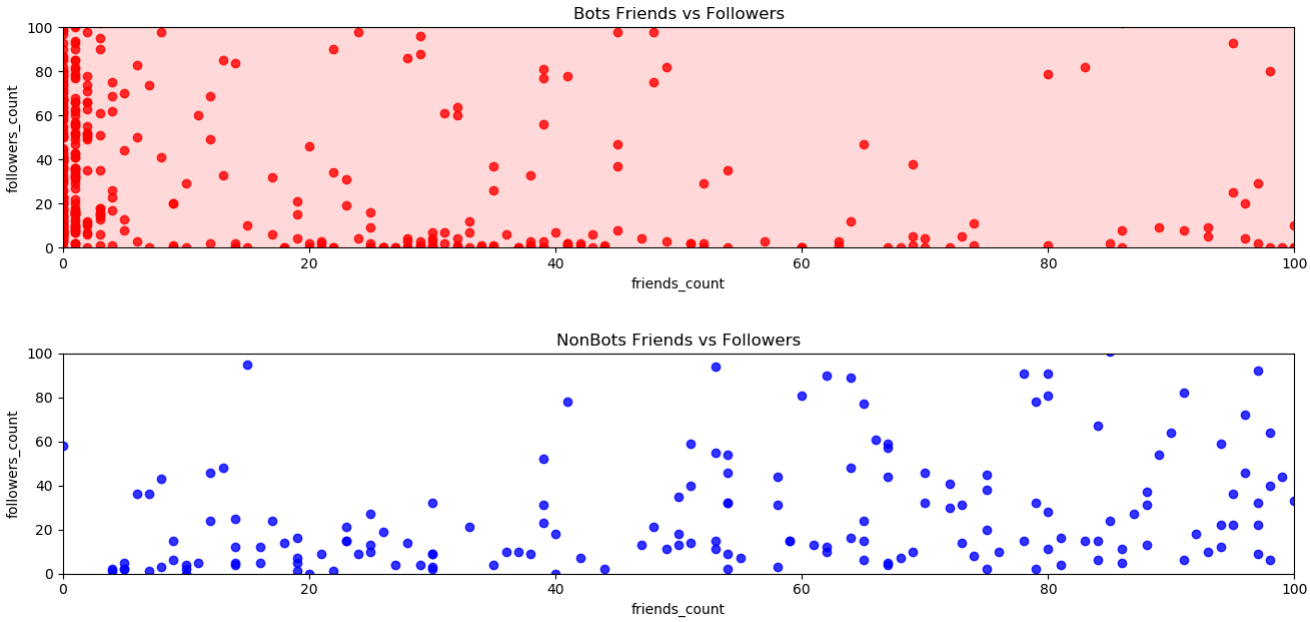

Figure 3.5: Friends to follower relation

3.3.3 Correlation

It shows the correlation between any two attributes higher the relation, more interdependent these attributes are to each other the strongly related attributes have more value and thus have a more important role in categorizing whether an account is a bot or a human the important attributes are favourite count, follower count and listed count. Correlation is an effect size and so we can verbally describe the strength of the correlation using the following guide for the absolute value of rs :

- 00-.19 “very weak”
- 20-.39 “weak”
- 40-.59 “moderate”
- 60-.79 “strong”
- 80-1.0 “very strong”

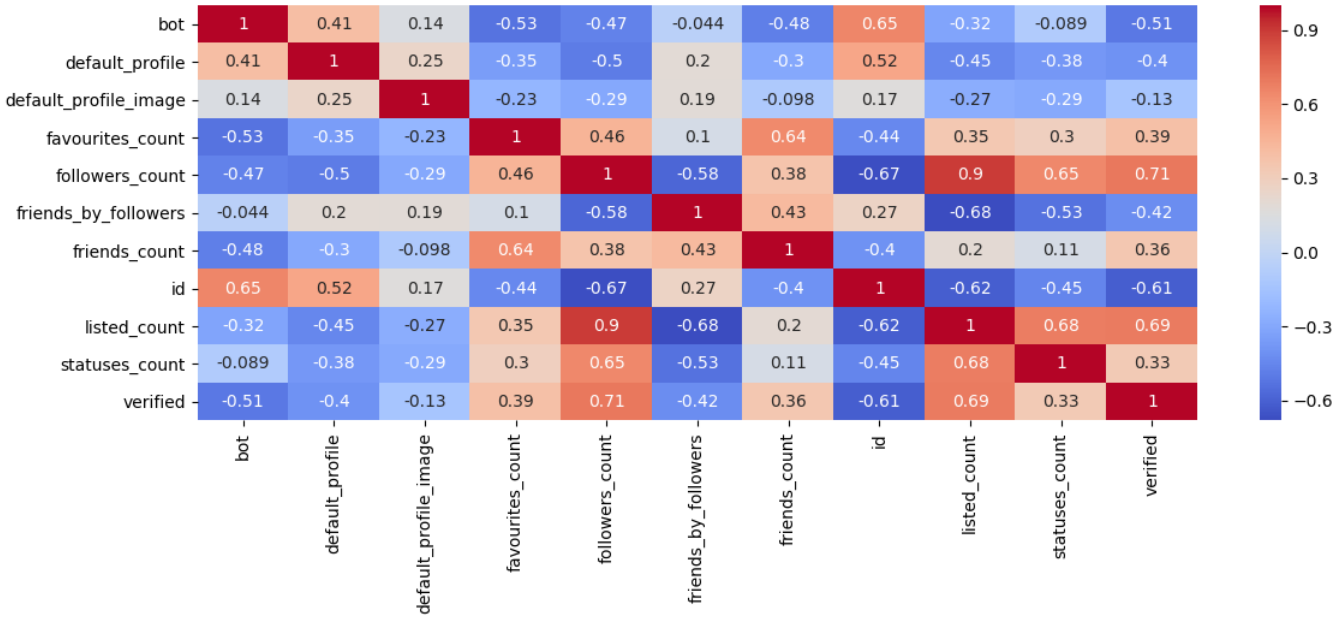

Figure 3.6: Correlation

3.3.4 Features of Dataset

Dataset contains 20 attributes out of which we have selected 8 attributes based on the spearman correlation.

- ID
- Friends count
- Follower count
- Listed count
- Favourite count
- Verified
- Statuses count
- Default profile
- Default profile image

3.3.5 Confusion Matrix

A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known. It allows the visualization of the performance of an algorithm. Each row of the matrix represents the instances in a predicted class while each column represents the instances in an actual class (or vice versa). The name stems from the fact that it makes it easy to see if the system is confusing two classes It allows easy identification of confusion between classes e.g. one class is commonly mislabeled as the other. Most performance measures are computed from the confusion matrix.

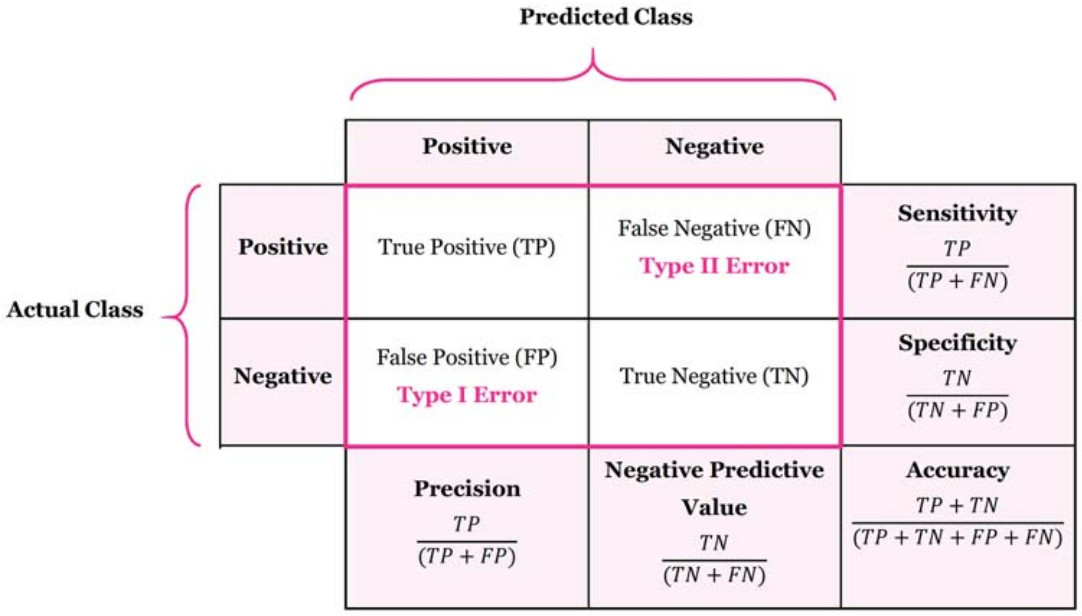

Figure 3.7: Confusion Matrix

3.3.6 ROC Curve

A receiver operating characteristic curve, or ROC curve, is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. The ROC curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. The true-positive rate is also known as sensitivity, recall or probability of detection in machine learning. The false-positive rate is also known as probability of false alarm and can be calculated as (1 specificity).The ROC curve is thus the sensitivity or recall as a function of fall-out. ROC analysis provides tools to select possibly optimal models and to discard suboptimal ones independently from (and prior to specifying) the cost context or the class distribution. ROC analysis is related in a direct and natural way to cost/benefit analysis of diagnostic decision makingThe ROC curve was first developed by electrical engineers and radar engineers during World War II for detecting enemy objects in battlefields and was soon introduced to psychology to account for perceptual detection of stimuli.

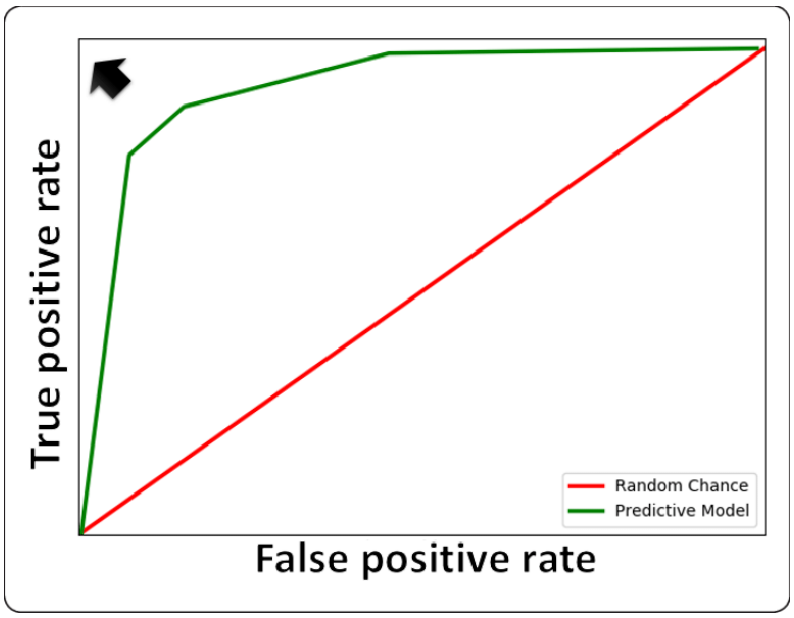

Figure 3.8: ROC Curve

3.3.7 Algorithms

For bot detection finding relationship between different attributes between in the dataset and applying different algorithms .And then the result of different algorithms will be compared and most accurate algorithm is found. The algorithms we are using are as follows:

- ***Decision Tree***

Decision tree is the most powerful and popular tool for classification and prediction. A Decision tree is a flowchart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.

A tree can be "learned" by splitting the source set into subsets based on an attribute value test. This process is repeated on each derived subset in a recursive manner called recursive partitioning. The recursion is completed when the subset at a node all has the same value of the target variable, or when splitting no longer adds value to the predictions. The construction of decision tree classifier does not require any domain knowledge or parameter setting, and therefore is appropriate for exploratory knowledge discovery. Decision trees can handle high dimensional data. In general decision tree classifier has good accuracy. Decision tree induction is a typical inductive approach to learn knowledge on classification.

- ***Random Forest Classifier***

Random Forest is a flexible, easy to use machine learning algorithm that produces, even without hyper-parameter tuning, a great result most of the time. It is also one of the most used algorithms, because it's simplicity and the fact that it can be used for both classification and regression tasks.

Random Forest is a supervised learning algorithm. Like you can already see from it's name, it creates a forest and makes it somehow random. The "forest" it builds, is an ensemble

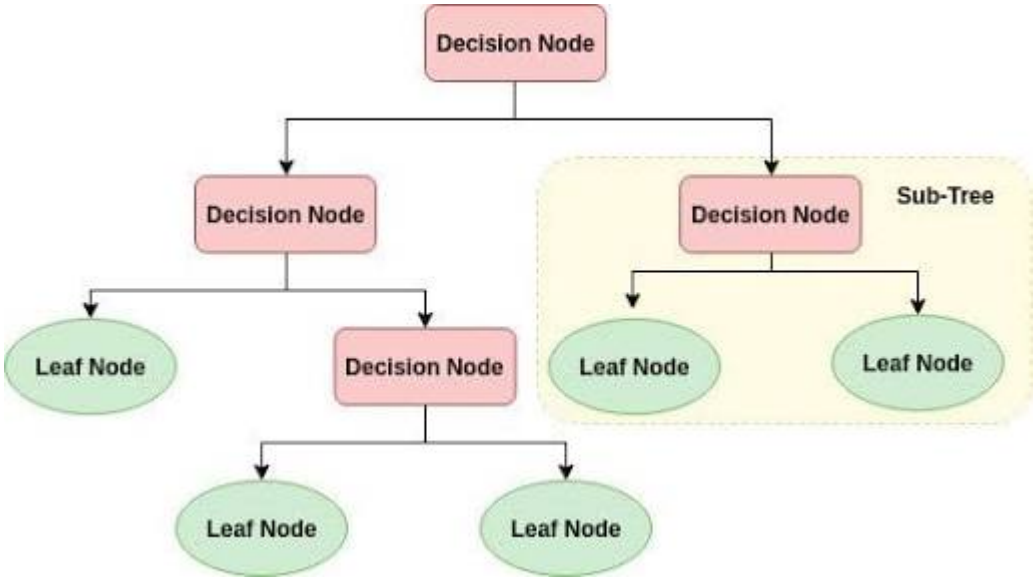

Figure 3.9: Decision Tree

of Decision Trees, most of the time trained with the "bagging" method. The general idea of the bagging method is that a combination of learning models increases the overall result. To say it in simple words: Random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction. One big advantage of random forest is, that it can be used for both classification and regression problems, which form the majority of current machine learning systems.

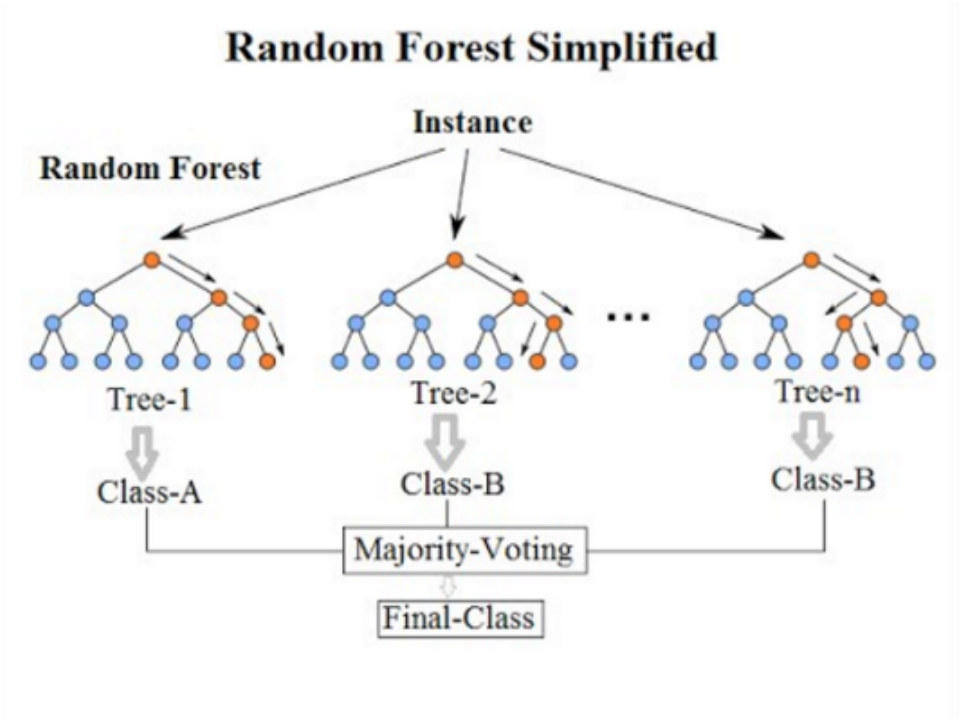

Figure 3.10: Random Forest

- ***K Nearest Neighbor***

K Nearest Neighbor(KNN) is a very simple, easy to understand, versatile and one of the topmost machine learning algorithms. KNN used in the variety of applications such as finance, healthcare, political science, handwriting detection, image recognition and video recognition. KNN algorithm used for both classification and regression problems. In K-NN classification, the output is a class membership. An object is classified by a plurality vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of that single nearest neighbor. In k-NN regression, the output is the property value for the object. This value is the average of the values of k nearest neighbors.

K-NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until function evaluation. Both for classification and regression, a useful technique can be to assign weights to the contributions of the neighbors, so that the nearer neighbors contribute more to the average than the more distant ones. For example, a common weighting scheme consists in giving each neighbor a weight of 1/d, where d is the distance to the neighbor.

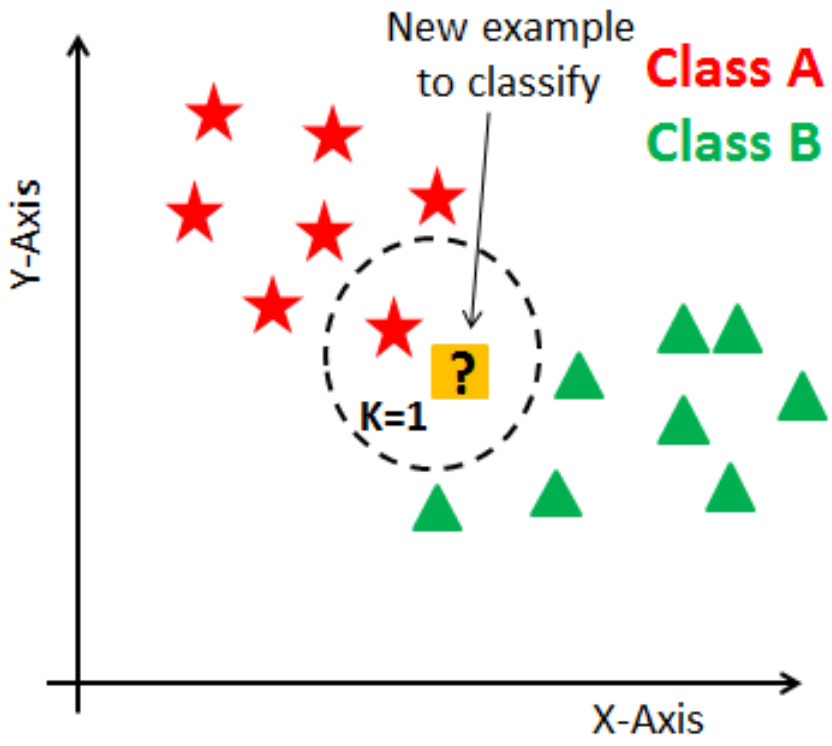

Figure 3.11: K-Nearest Neighbor

- ***Support Vector Machine***

Support-Vector Machines(SVM) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of

training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on the side of the gap on which they fall.

In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces. When data are unlabelled, supervised learning is not possible, and an unsupervised learning approach is required, which attempts to find natural clustering of the data to groups, and then map new data to these formed groups.

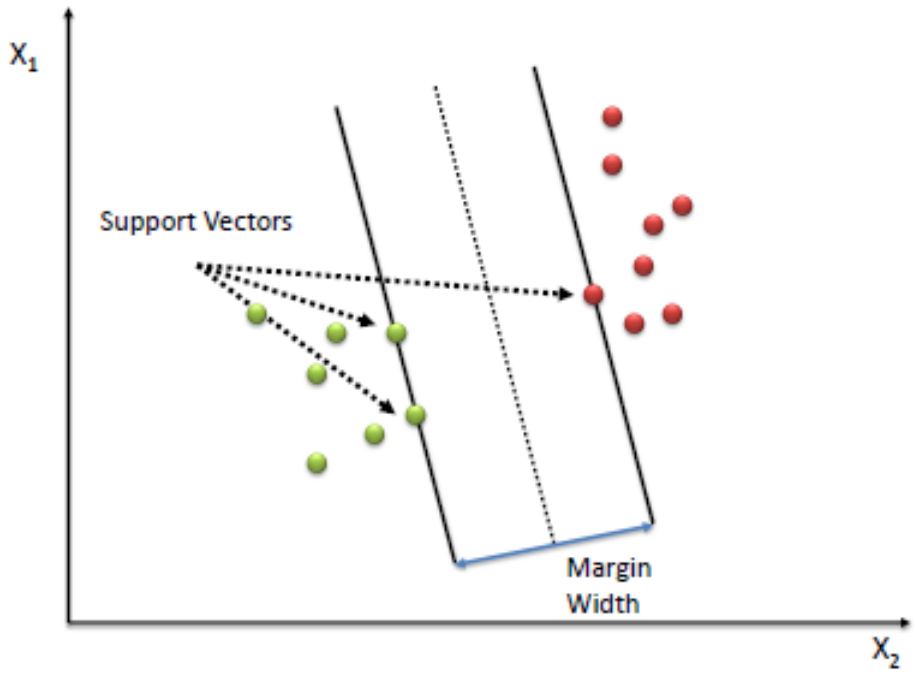

Figure 3.12: Support Vector Machine

Chapter 4

Results & Discussions

4.1 Implemented Algorithm's Psudo-code

- ***Decision Tree***

```
def decisionTreeLearning(examples, attributes,parent_examples):
if  len(examples) == 0:
return pluralityValue(parent_examples)
elif  len(attributes) == 0:
return pluralityValue(examples)
elif  (all examples classify the same):
return their classification

A = max(attributes, key(a)=importance(a, examples)
tree = new Tree(root=A)
for value in A.values():
exs = examples[e.A == value]
subtree = decisionTreeLearning(exs, attributes.remove(A), examples)
tree.addSubtreeAsBranch(subtree, label=(A, value)]]

return tree
```

- ***Random Forest***

```
function RandomForest(S , F)
H ←
for i  1, . . . , B
S(i) ← A bootstrap sample from S
hi ← RandomizedTreeLearn(S(i), F)
H ← H  {hi}
end for
return H
end function
function RandomizedTreeLearn(S , F)
At each node:
```

```
f ← very small subset of F
Split on best feature in f
return The learned tree
end function
```

- ***K-Nearest Neighbor***

```
Classify(X,Y,x)//X:Training Data,Y:Class Labels of X,x:Unknown Sample
for i=1 to m do
Compute distance d(X,x)
end for
Compute set I containing indices for the k smallest distances d(X,x)
return majority label for{Y,where iI}
```

- ***Support Vector Machine***

```
import pylab as pl
feautures_train,labels_train,features_test,label_test=DogData()
from sklearn.svm import SVC
clf= SVC(kernel="linear")
clf.fit(features_train,labels_train)
pred= clf.predict(features_test)
from sklearn.metrics import accuracy_score
acc= accuracy_score(pred,labels_test)
```

4.2 Results

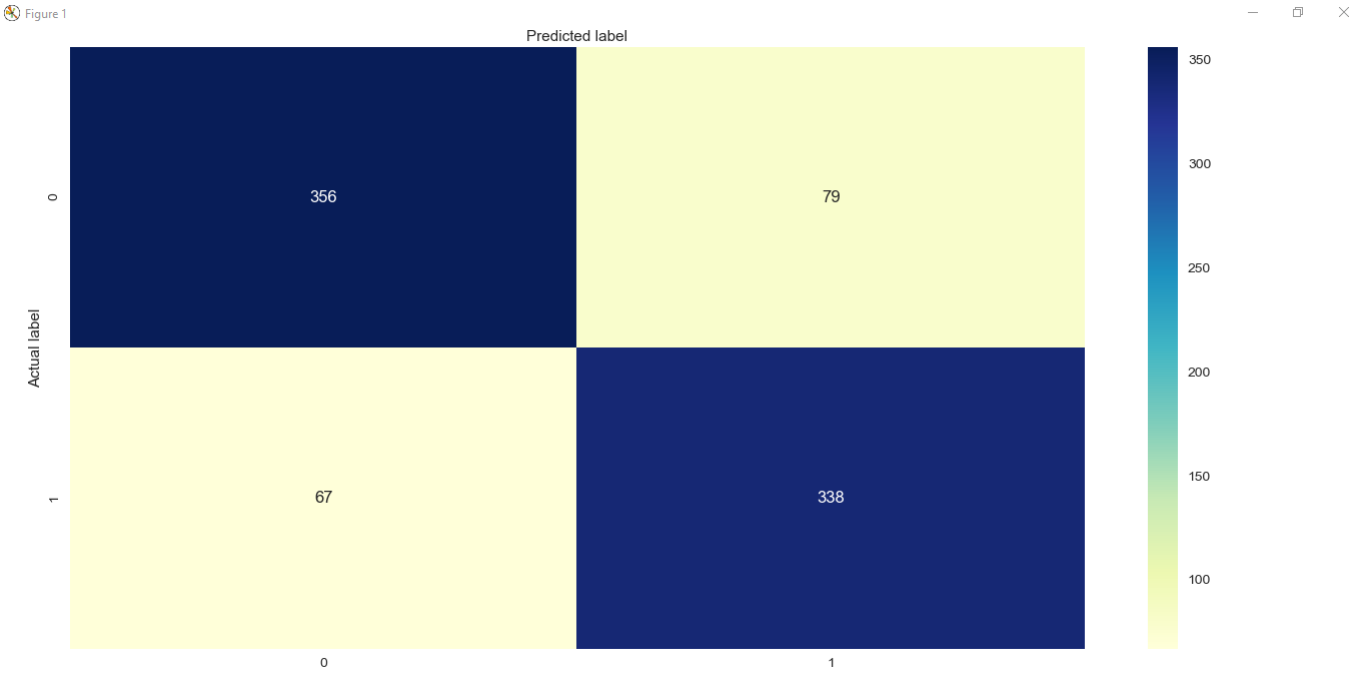

Figure 4.1: Confusion Matrix of Decision Tree

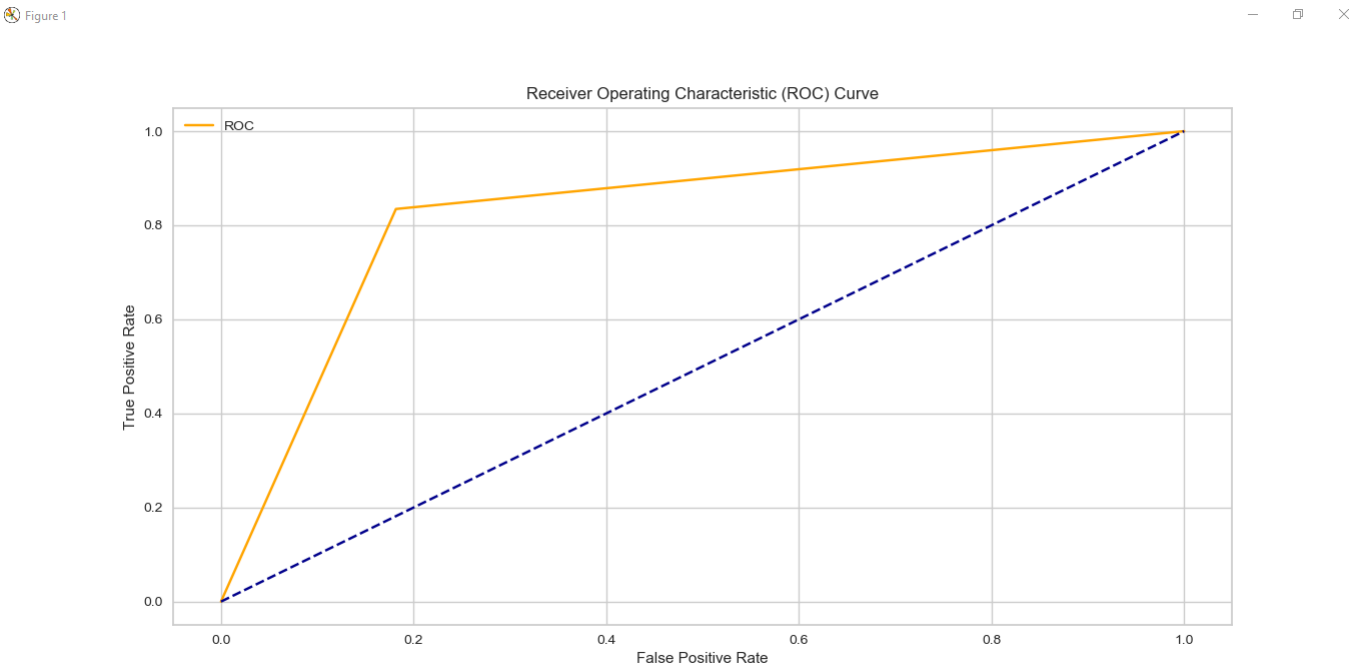

Figure 4.2: ROC curve of Decision Tree

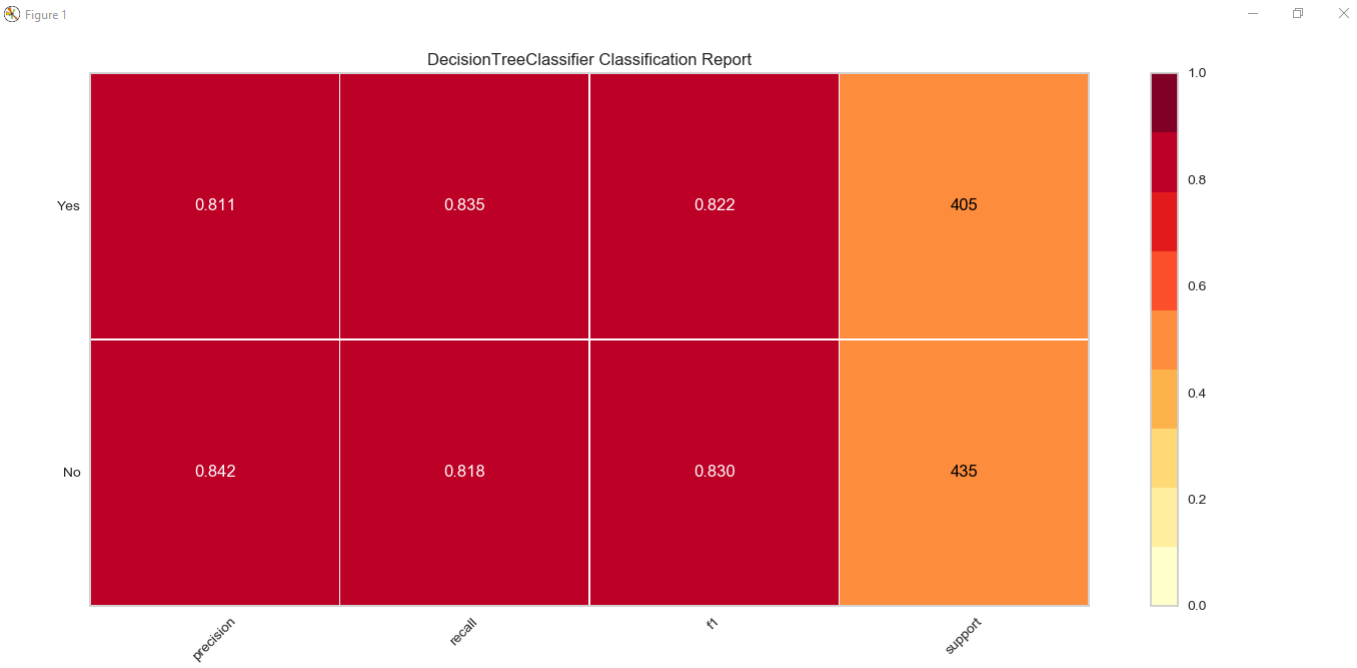

Figure 4.3: Classification Report of Decision Tree

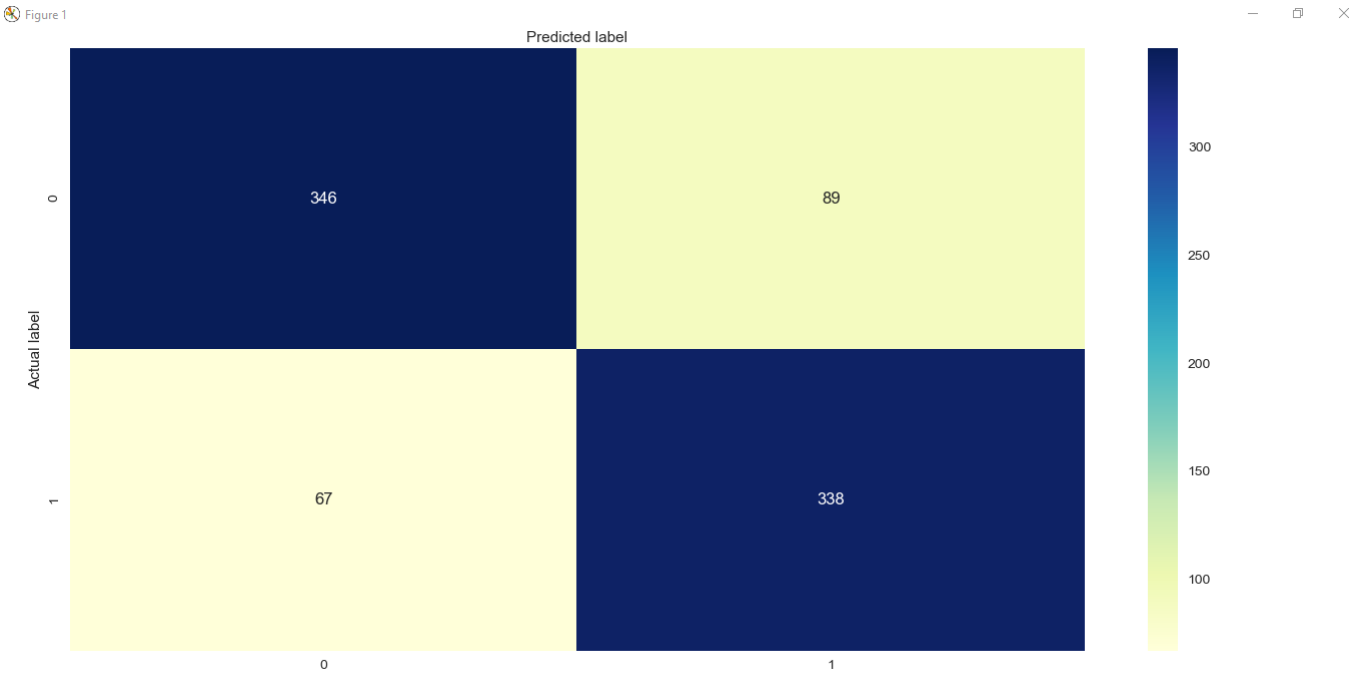

Figure 4.4: Confusion Matrix of KNN

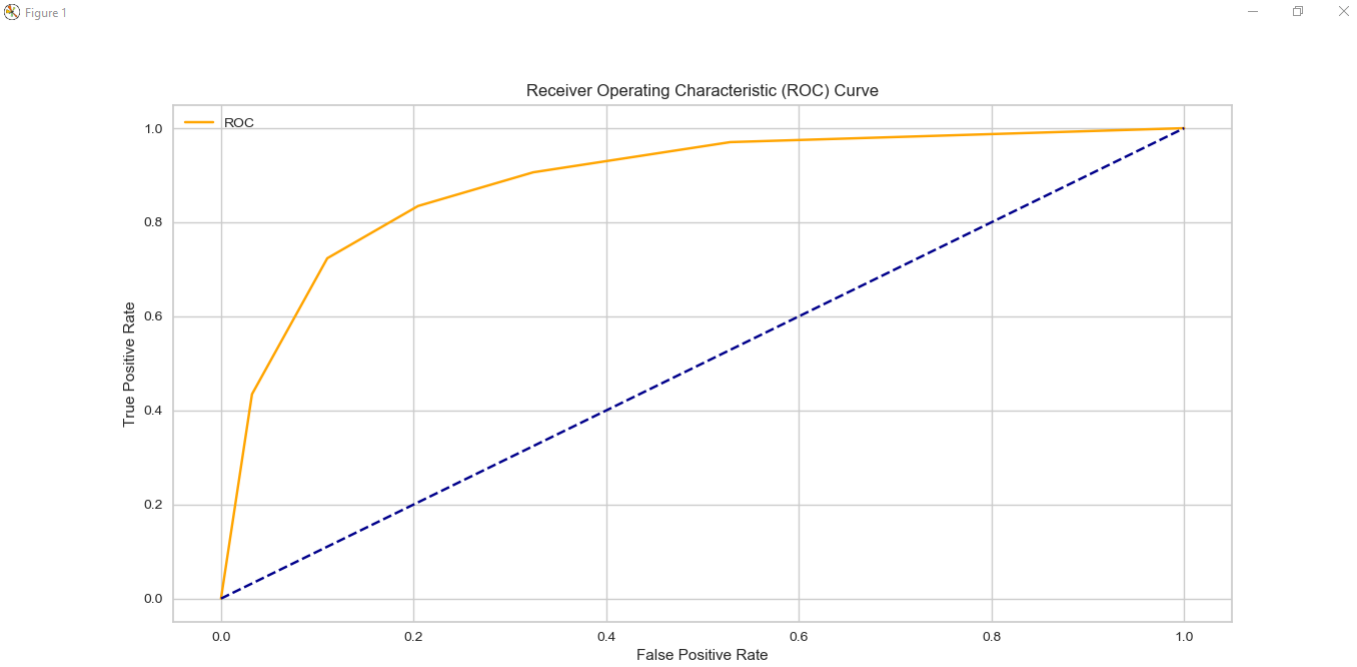

Figure 4.5: ROC curve of KNN

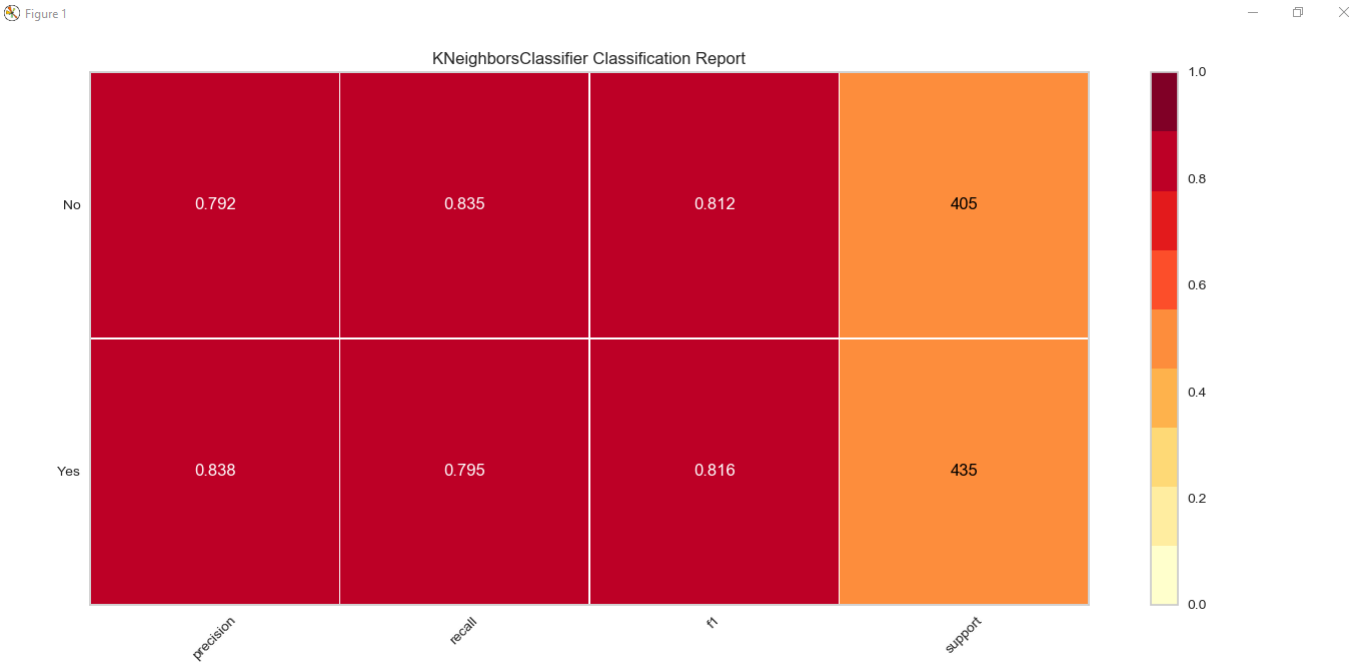

Figure 4.6: Classification Report of KNN

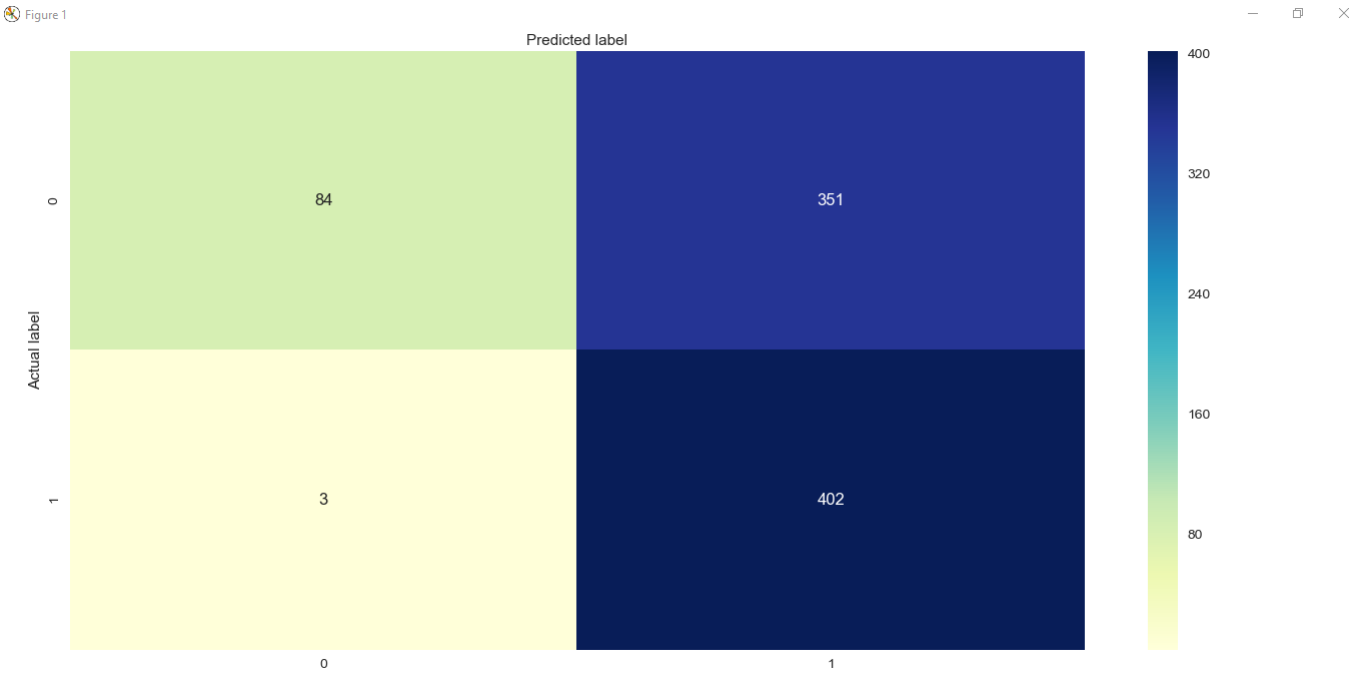

Figure 4.7: Confusion Matrix of SVM

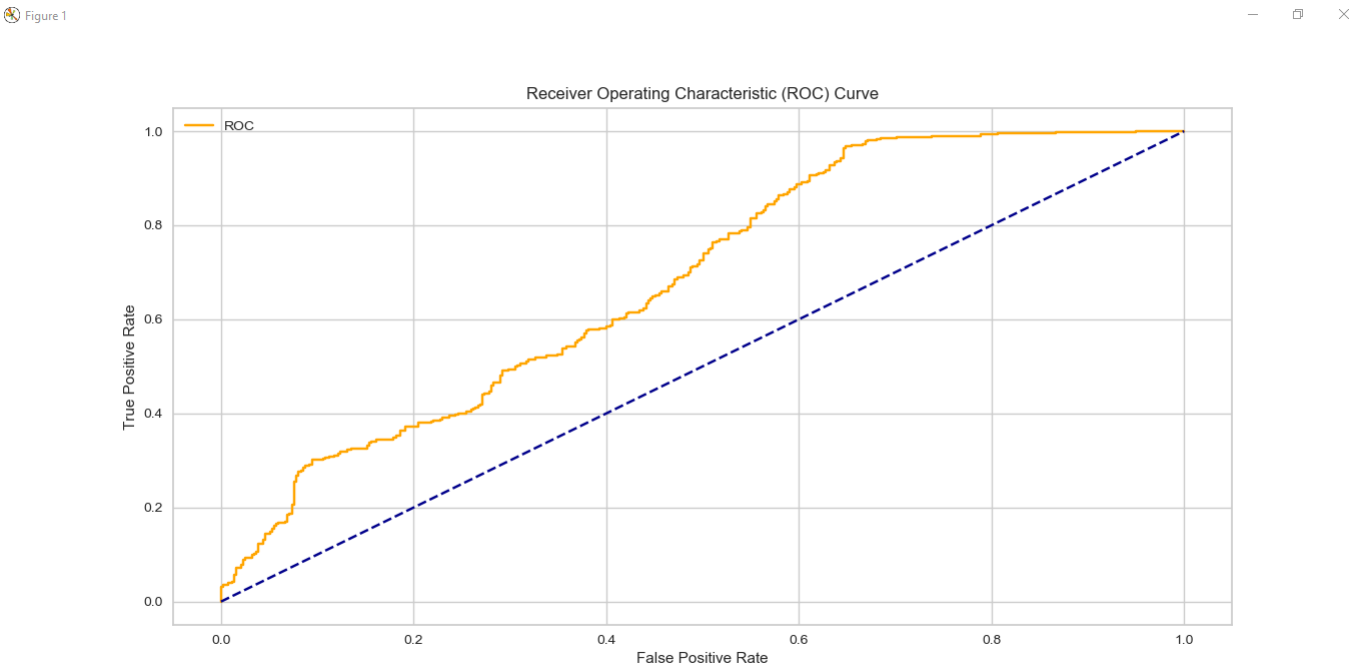

Figure 4.8: ROC curve of SVM

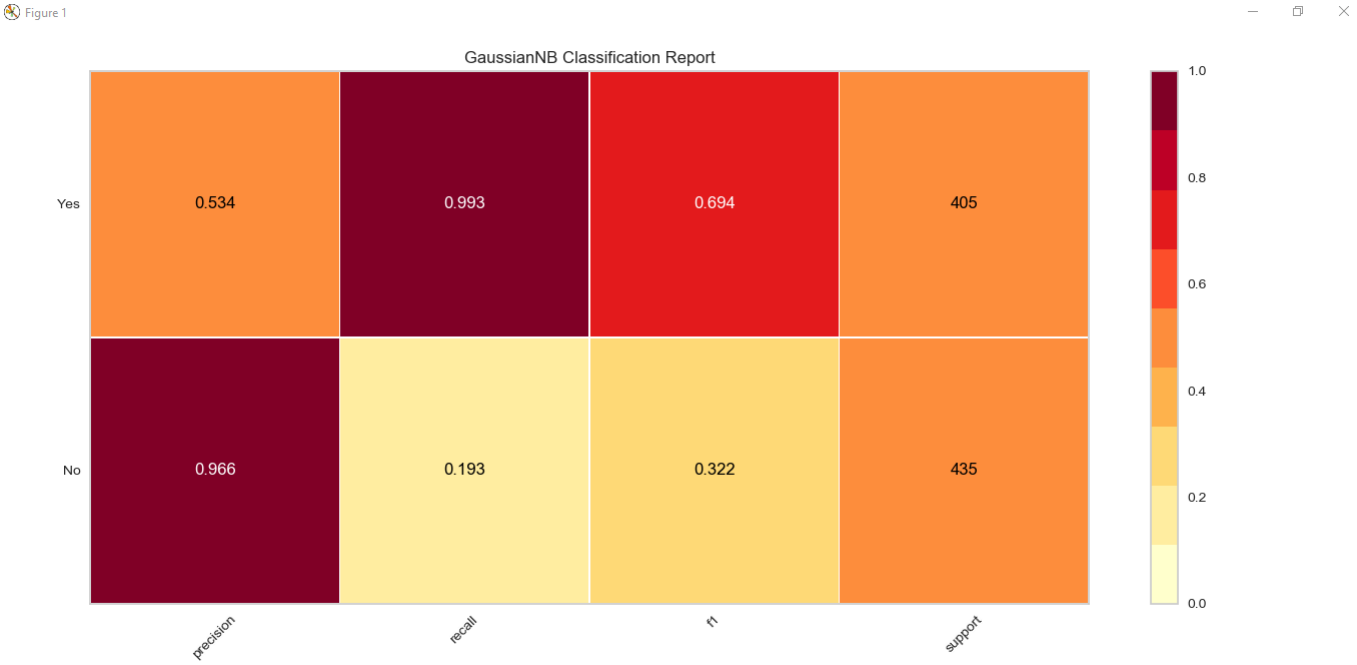

Figure 4.9: Classification Report of SVM

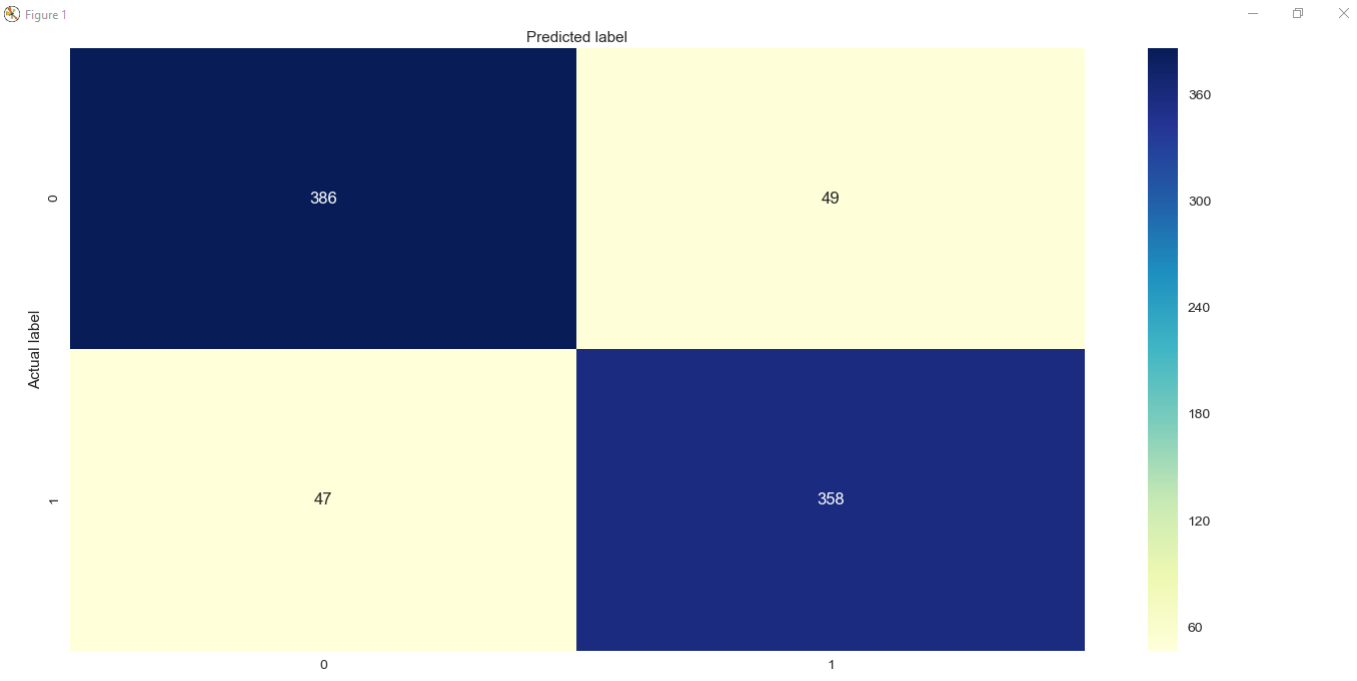

Figure 4.10: Confusion Matrix of Random Forest

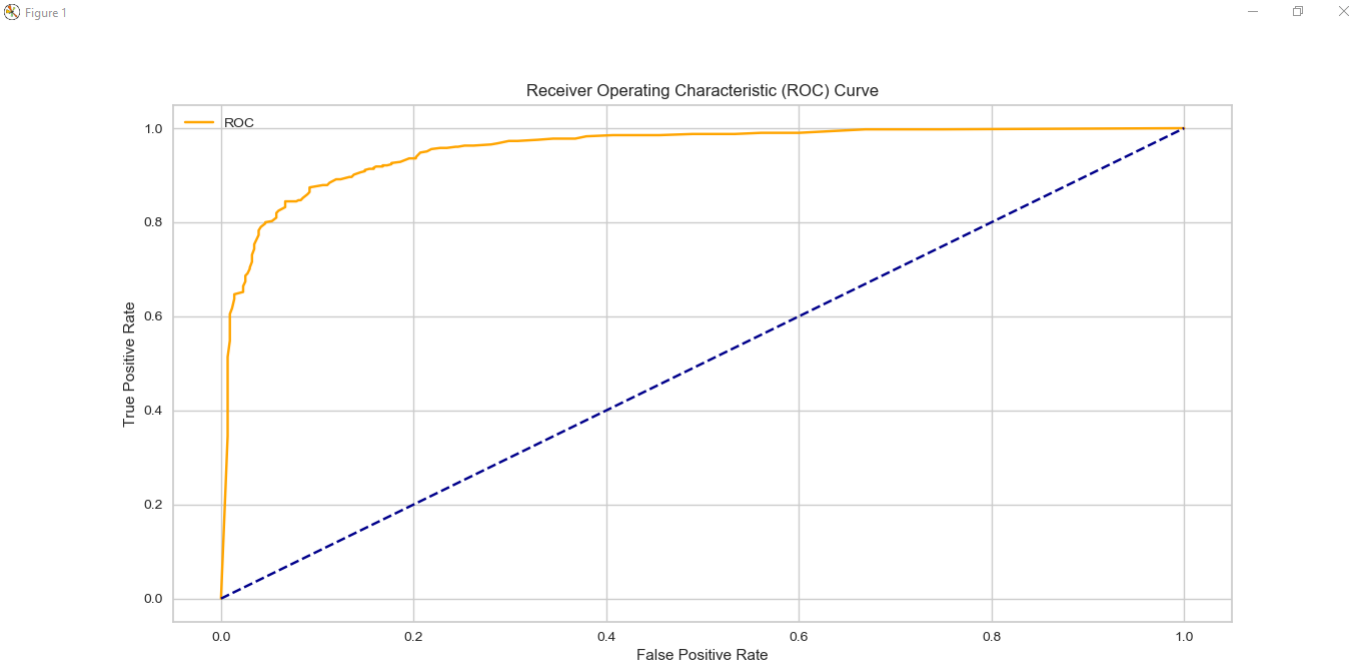

Figure 4.11: ROC curve of Random Forest

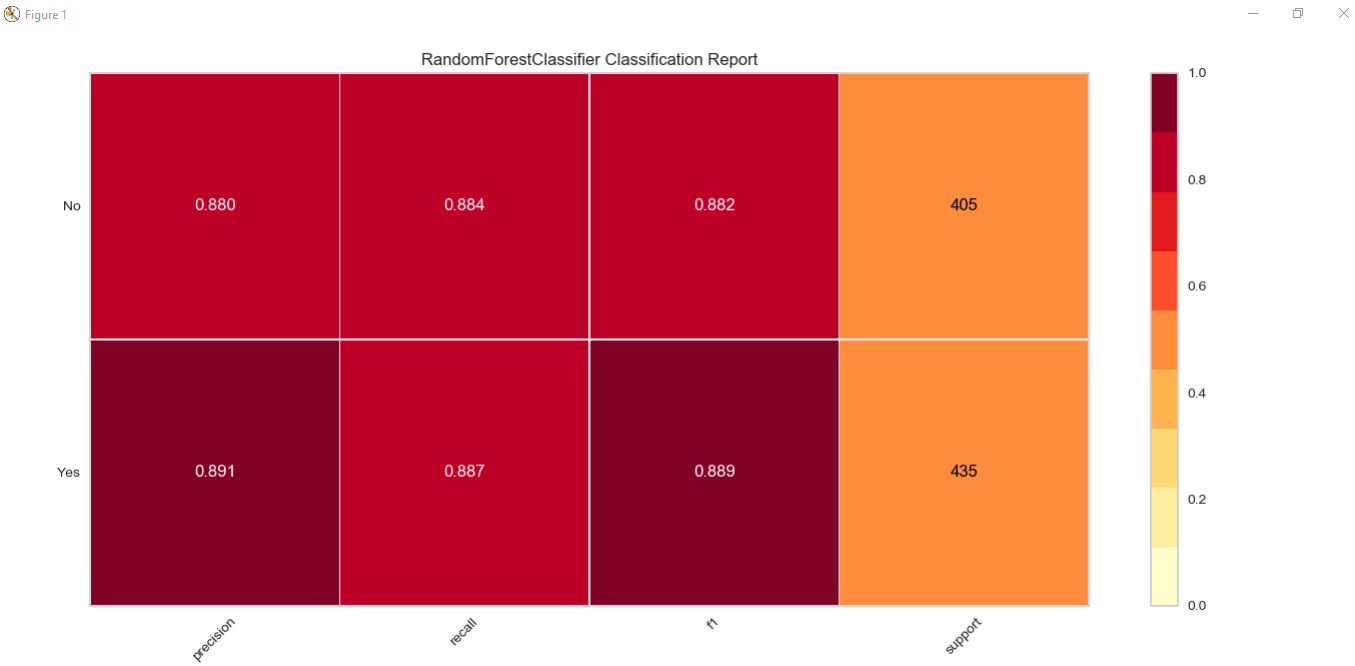

Figure 4.12: Classification Report of Random Forest

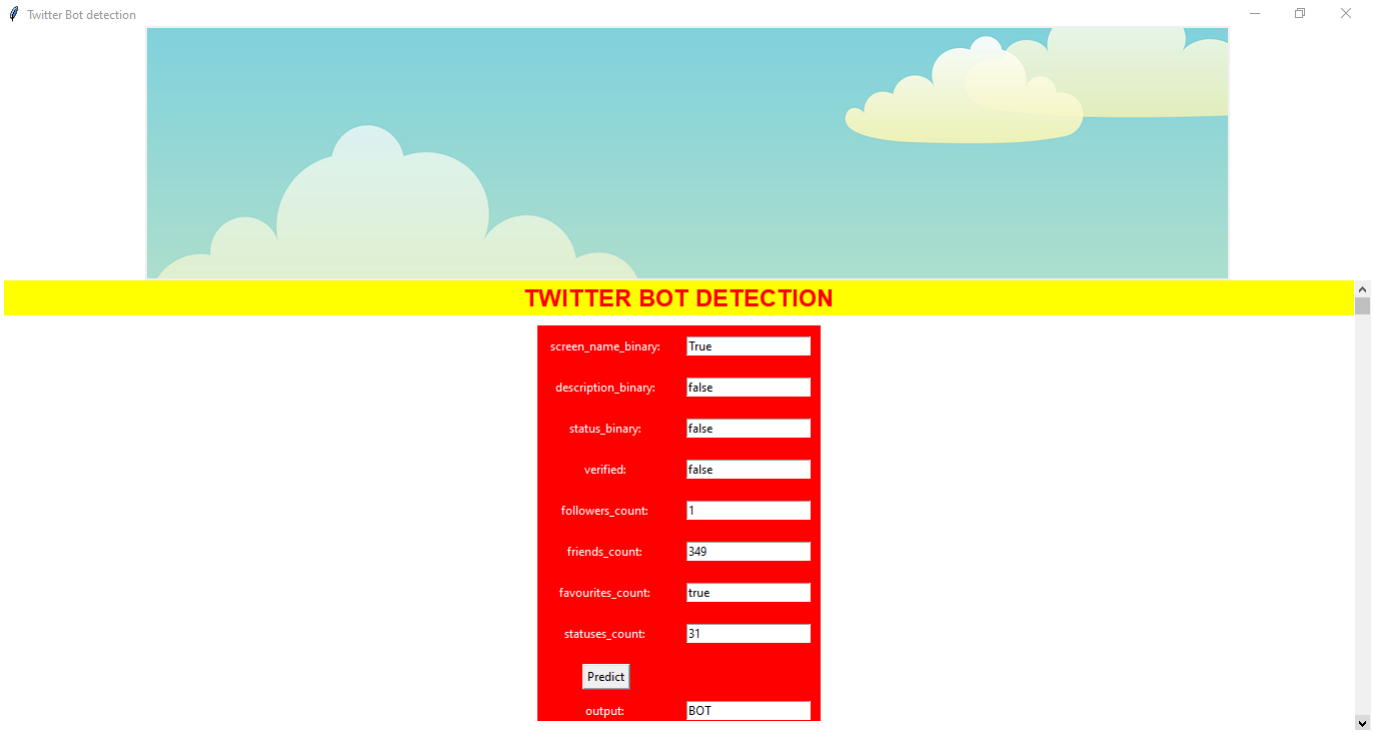

Figure 4.13: Output of Bot Account

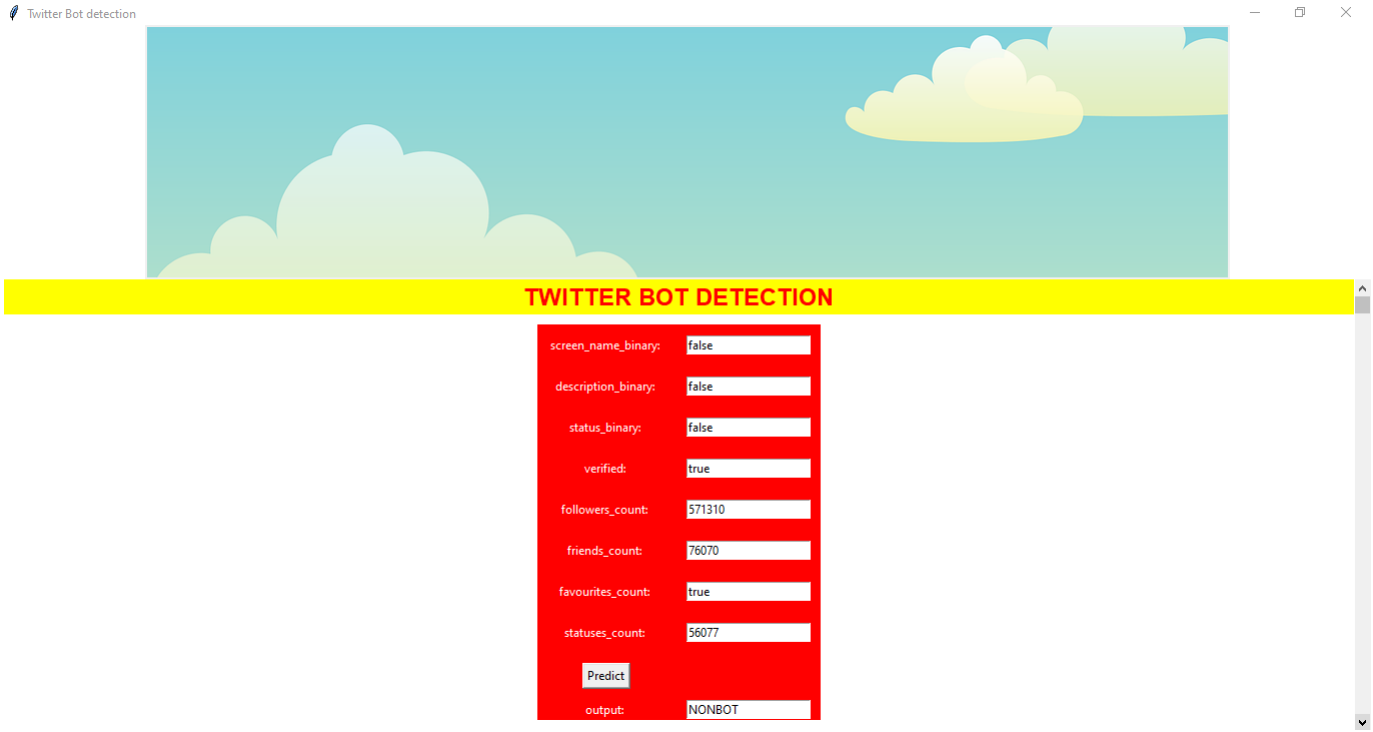

Figure 4.14: Output of Non-Bot Account

4.3 Discussion-Comparative study/Analysis

Table 4.1: Comparison of Different Algorithms

Algorithm	Training Accuracy	Testing Accuracy
Decision Tree	99.94%	82.14%
Random Forest	99.94%	88.45%
K-NEarest Neighbor	87.07%	81.42%
Support Vector Machine	58.81%	57.85%

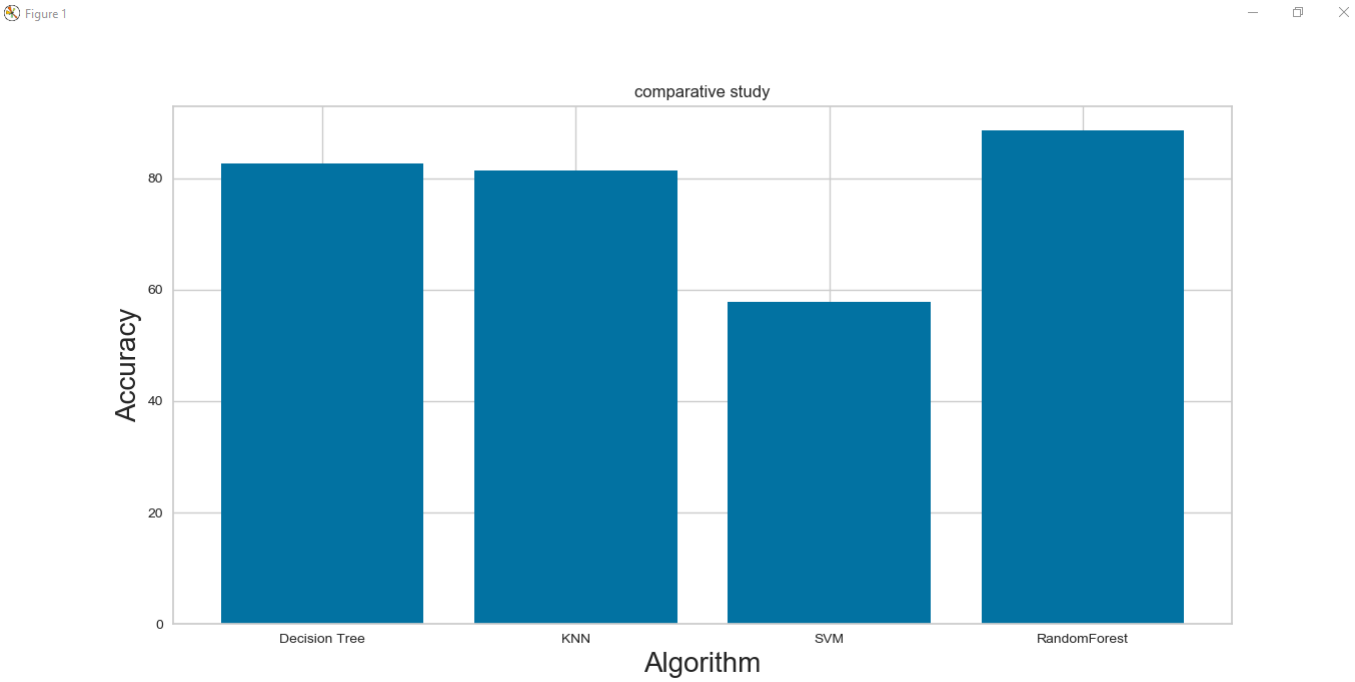

Figure 4.15: Comparison of the Algorithms

Chapter 5

Conclusions

Twitter bot is a program used to produce automated posts, follow Twitter users or serve as spam to entice clicks on the Twitter microblogging service. In this project, we used Machine Learning techniques to predict weather an account on Twitter is a Bot or a real user. We have performed significant amount of feature engineering, along with feature extraction - selected features out of 20 helped us to identify whether an account is bot or non bot. Our framework will be able to identify whether a twitter user is a bot or a human. We can extend our work to other social media platform like facebook. Our work will safeguard oneself and an organization from false information, malicious content and ensure their brand value. Our project can also be utilized to identify human online traffic from bot activity.

Appendix

Appendix I

During the process of detecting the fake identities humans and bots have same behavior. These are applied to many supervised machine learning models. Many engineered features are existing but are not much successful in implementing to detect the malicious accounts. Existing system use only two parameters.

"Friend-to-followers ratio, Friend count Less prediction accuracy, not an real time analysis and existing system not used for an long dataset.

Create a social media tweets, hashtags, social media posts, feeds, comments. Create non relational databases. Using data set preparation, cleaning .Then create a dataset. Applying the Ml supervised machine learning algorithms. Finally evaluate and visualize the results. Its gives accuracy more than 85 percent. Its an real time data analytics. The existing system detect fake identities to 50 percent of accuracy. Three types of machine learning algorithm are used to detect the fake identities. The model is dependent on features (name, location, profile image).

Data collection is the first activity. It is collected from various social media networks (twitter, kaggle, data.gov) etc. Then create non-relational databases. Then cleaning process is started after that the data is stored in relational databases. Then train the dataset using supervised machine learning algorithms (Linear regression, Navies Bayes). Finally the results are visualised and evaluated.

Bibliography

[1] Arzum Karatas, Serap Sahin, “A review on Social Bot Detection Techniques and Research Directions” ACSAC, Oct 2017, pp 1-3.

[2] Zafar Gilani, Ekaterina Kochmar, John Crowcraft, “ Classification of Twitter Accounts into Automated Agents and Human Users”, ACM/KDD, 2017, pp 489-496

[3] Estee Van Der Walt, Jan Eloff, ”Using machine learning to detect fake identities: bots vs humans”,USENIX Security Sym, 2018, pp 321-330

[4] ”Estee Van Der Walt, Jan Eloff,” Using machine learning to detect fake identities: bots vs humans”,2018, pp. 25-32.

[5] Ashraf Khalil, Hassan Hajjdiab, and Nabeel Al-Qirim,” Detecting Fake Followers in Twitter: A Machine Learning Approach”,Artif. Intell. Secur, Dec 2017, pp 143-152.

[6] Arzum Karatas, Serap Sahin”,A Review on Social Bot Detection Techniques and Research Directions”,24-2 Oct 2017,pp. 1 - 3.

[7] S. Gurajala, J. S. White, B. Hudson, B. R. Voter, J. N. Matthews, ”Profile characteristics of fake Twitter accounts”, Big Data Soc., vol. 3, 2016, pp. 205-211.

[8] C. Xiao, D. M. Freeman, T. Hwa, ”Detecting clusters of fake accounts in online social networks”, Proc. 8th ACM Workshop Artif. Intell. Secur., 2015, pp. 91-101.

[9] Zi Chu, Steven Gianvecchio, Haining Wang, “ Detecting automation of twitter accounts: are you a human, bot or cyborg”, ACSAC, 2012, pp. 811-824.

Publication by Students

Paper entitled "Analysis of Twitter accounts for detecting bots and humans using Machine Learning" is presented at *International Conference on Circuits, Systems, Information and Communication Technology Applications (CSCITA - 2014)*, In collaboration with IEEE Bombay Section,Mumbai with the approval of IEEE, USA.

Annexure

Analysis of Twitter Accounts For Detecting Bots Using Machine Learning

=210mm =297mm

Prof. Anita A. Lahane
Computer Engineering
Rajiv Gandhi Institute of Technology
Mumbai,India
anita.lahane@mctrgit.ac.in

Tanish Sanghavi
Computer Engineering
Rajiv Gandhi Institute Of Technology
Mumbai, India
tanssang@gmail.com

Shubham Shete
Computer Engineering
Rajiv Gandhi Institute Of Technology
Mumbai, India
shubhamshete2506@gnail.com

Aman Sarawgi
Computer Engineering
Rajiv Gandhi Institute Of Technology
Mumbai, India
amannsarawgi@gmail.com

***Abstract*— Twitter is an online social media platform to share news, messages and micro blogging where people communicate in short messages called tweets. In 2019 there was 330 million active user , out of which research says that about 9% to 15% of the total accounts are bots. In today's date, Twitter has become an important medium for political conversation, protest and for giving personal views so the possibilities for harm increased exponentially. Twitter popularity is giving rise to new spam marketplace. In addition, twitter users have started to buy fake followers for the only purpose of showoff. In this project we are using machine learning algorithms to detect a twitter account is human or bot. We identified a number of characteristics that distinguish fake and genuine account. We used these characteristics as attributes to machine learning algorithms to classify users as human or bot.**

Keywords— **Twitter, Bot, Automatic identification, social networks, social media**

I. INTRODUCTION

Twitter is a popular social media medium where people share their ideas, opinions and express their feeling freely. In twitter, users can follow the people they find interesting and get followed back by people with similar ideologies.

A particular user can send tweets which can be viewed by everyone and its a global platform for people to discuss about important topics. Tweets about a particular topic can be retrieved using the twitter's real time search engine.

Tweets have ranking, which is based of many individual attributes, the most important attribute is the number of followers the user has. These tweet appear at the top and get the most views, thus it is a very influential tweet.

With the increase in the popularity of twitter and the number of users, it has become an attraction for spam and the spammers of all types. Spammers have various goals: Advertising false news, phising attack or just compromising a users authenticity. As the number of spammers increase, it is increasingly becoming difficult for the search engine to distinguish between the valuable tweet and false rumors.

The number of twitter bots have also been on a rise, this reduces the authenticity of the messages on twitter. Twitter accounts controlled by bots have the ability to perform actions like tweeting, re-tweeting, liking, following, unfollowing or even direct messaging a user.

The bots can spam a user with unnecessary messages or can even spread wrong information which other users might read and fall for, this can be very dangerous and thus result in the loss of activity of users on twitter. Social media sites, such as Twitter, regularly suspend abusive bots. Yet, the number of bots is growing because of almost zero-cost in creating new bots. Existing bot detection methods are not capable of fighting such evolving set of bots. They copy or mimic humans to avoid being detected and suspended and increase throughput by creating many accounts.

II. RELATED WORK

In the paper written by Estee Van Der Val, it is shown that the engineered features that were previously used to detect fake accounts generated by bots are not similarly successful in the detection of fake accounts generated by humans.

In the paper written by Arzum Caratas, Serap Sahin, the social bot detection techniques, it is seen that the higher social bot detection rates (over 80 percent) are obtained with the combination of the structure-based properties of OSN and unsupervised machine learning methods.

In the paper written by Supraja Gujarala, Brian Hudson, they have presented a machine learning pipeline for detecting fake accounts in online social networks. Rather than making a prediction for each individual account, our system classes clusters of fake accounts to determine whether they have been created by the same actor.

In the paper written by Cao Xiao, Theodore Hwa, they have trained models using random forest, logistic regression, and support vector machine classifiers. They also evaluated

the classifiers' performance with 80-20 split in-sample testing and out-of-sample testing with a more recent data set.

In the paper written by Z. Chu, S. Gianvecchio, they have evaluated the accuracy of our classification system based on the ground truth set that includes both the training set are Input into the classifier. LDA generates a weight table to achieve the maximum accuracy.

Features that are based on accounts are lightweight enough to be used detecting real-time spam which requires instant analysis. The number of lists the user is a member of can be considered a useful metric to detect spammers as it is an obvious sign of the user's impact on others but its open to manipulation by creating fake lists and adding fake accounts which are under the CC infrastructure into these lists. Account-based features are lightweight enough to be used detecting real-time spam which requires instant analysis but they can be easily manipulated by spammers.

III. ARCHITECTURE

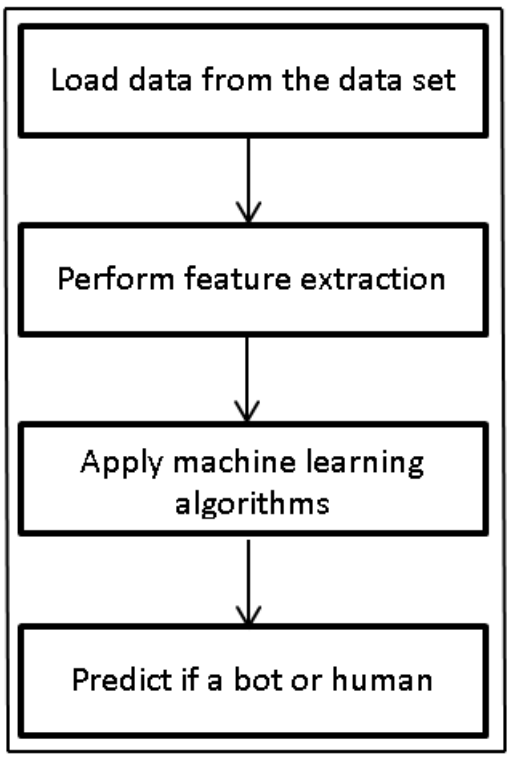

Fig 3.1: Architecture

Step 1: Firstly we need to load the data of the Twitter accounts from the data sets.
Step 2: Extract features of tweet and save the extracted features as csv file so that we can use it to train the classifiers.
Step 3: Apply different machine learning algorithms to predict whether the account is a bot.
Step 4: Test the classifiers using the user inputs and confirm whether it is a bot or not.

IV. IMPLEMENTATION

[1] Feature extraction and data analysis:-

(1) Heat Map:
In this graph the null values are displayed with yellow color and the number of null values with violet color. higher the number of null values, higher the chance of the account being a bot the contrast of the number of yellow patches show the difference between bot and humans this heat chart gives us a basic idea of how details in an account can help us identify if an specific account comes under being a bot or a human.

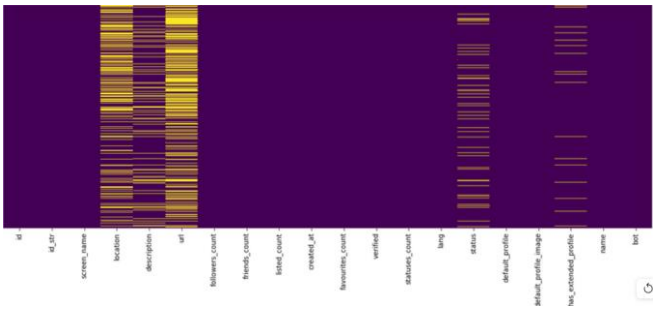

Fig 4.1: Heat Map

(2) Friends to Follower relation:
In the relation of the followers count to the number of friends count. from the graph we see, bots have a huge number of followers in comparison to the number of friends the majority of points in the above graph are towards the beginning whereas, in the graph below, the points are more scattered and well spread.

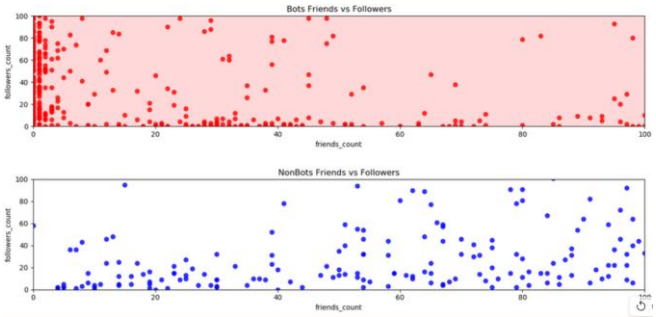

Fig 4.2: Friends Vs Followers Graph

(3) Correlation:
It shows the correlation between any two attributes higher the relation, more interdependent these attributes are to each other the strongly related attributes have more value and thus have a more important role in categorizing whether an account is a bot or a human the important attributes are favourite count, follower count and listed count. Correlation is an effect size and so we can sdescribe the strength of the correlation using the following
guide for the absolute values :

- 00-.19 \very weak"
- 20-.39 \weak"
- 40-.59 \moderate"
- 60-.79 \strong"

• 80-1.0 \very strong"

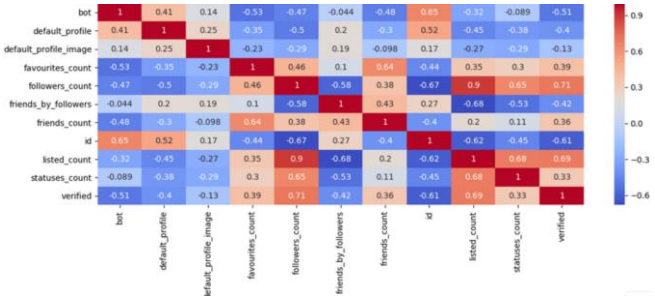

Fig 4.3: Correlation Graph

(4) Receiver operating characteristic curve (ROC):

A receiver operating characteristic curve or also known as ROC curve, is a graphical plot which illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied.

The receiver operating characteristic curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. The TPR is also known as sensitivity, recall or probability of detection in machine learning. The FPR is also known as probability of false alarm and can be calculated as (1 – specificity).

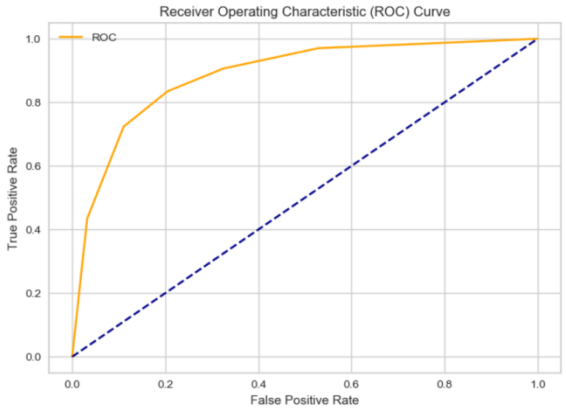

Fig 4.4: Receiver Operating Curve

[2] Algorithms:-

(1) Confusion Matrix:
A confusion matrix is a table that's used to describe the performance of a classifier on a set of test data for which the true values are known.

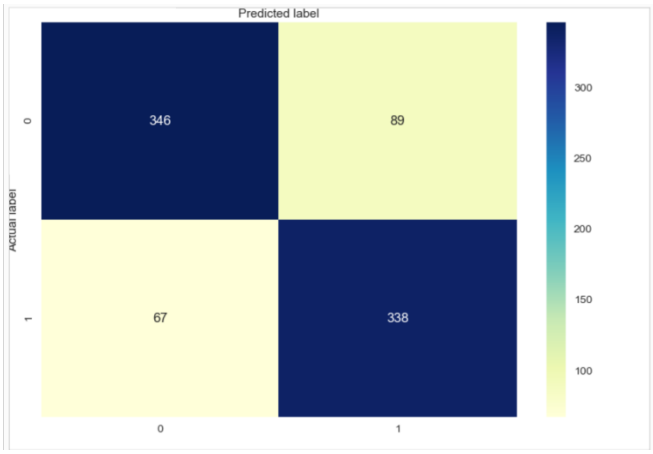

Fig 4.5: Confusion Matrix

True Negatives (TN): These are cases in which we predicted as humans, and they were bots.
False Positives (FP): These are cases in which we predicted as humans , and they were bots.
False Negatives (FN):These are cases in which we predicted as bots , and they were humans.
True Positives (TP): These are cases in which we predicted as bots , and they were bots.

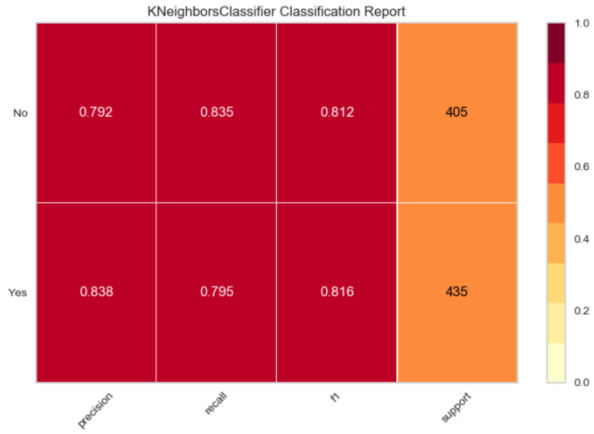

Fig 4.6: K Neighbour Classifier

Performance of the classifier is calculated as:

$$Recall\ (No) = \frac{\text{TN}}{\text{Actual (No)}}$$

$$Recall\ (Yes) = \frac{\text{TP}}{\text{Actual (Yes)}}$$

$$Precision\ (No) = \frac{\text{TN}}{\text{Predicted (No)}}$$

$$Precision(Yes) = \frac{TP}{\text{Predicted (Yes)}}$$

$$F1\ measure\ (No) = \frac{2 * \text{Precision (No)} * Recall\ (No)}{\text{Precision (No)} + Recall\ (No)}$$

$$F1\ measure\ (Yes) = \frac{2 * \text{Precision (Yes)} * Recall\ (Yes)}{\text{Precision (Yes)} + Recall\ (Yes)}$$

$$Accuracy = \frac{\text{TP} + \text{TN}}{\text{Total}}$$

(2) K – Nearest Neighbour:

The k-nearest neighbors (KNN) algorithm is a supervised machine learning algorithm that is used to solve both classification and regression problems. A supervised machine learning algorithm is one the algorithm which relies on labeled input data to learn a function which produces an appropriate output after given new unlabeled data.

It is very popular algorithm for classification of data. This algorithm is used for categorising dataset samples based on nearest training samples. To classify the test twitter account data, KNN algorithm identifies, k closest samples that are similar to test sample. The k nearest neighbour are identified by similarity of sample data set. The data sample similarities are computed with some set of the similarity measures. Euclidean distance is one of similarity computing approach. The distance between two data samples can be found using Euclidean distance. The performance of classification model is improved using cross validation technique. The cross validation technique is used to validate the classification model performance and accuracy. After k nearest neighbours is found, various different ways are used to predict the class label of the test twitter account data. A fixed k value is used for all classes.

The KNN Algorithm :

1. Load the twitter accounts data set.

2. Initialize value of K to the chosen number of neighbors.

3. For each example in the data set.
   3.1 Calculate distance between the query example and the current example from the data.
   3.2 Add the distance and index of the example.

4. Sort the ordered collection of the distances and indices from smallest to largest by the distances.

5. Select the first K entries from the sorted collection.

6. Get the labels of the selected K entries.

7. If there is regression, then return the mean of the K labels.

8. If there is classification, then return the mode of the K labels.

V. RESULT AND CONCLUSION

Twitter bots are turning into a very serious issue. To analyze the specific account is operated by human or a bot we are used different data analysis techniques such as analyzing missing data, heat map, friends to follower ratio, ROC curve, and machine learning algorithms are spearman's correlation, confusion matrix and KNN algorithm .

Performance of the classifier is calculated as:

	Recall	Precision	f-measure
No	0.795	0.836	0.813
Yes	0.835	0.795	0.816

Table 5.1: Accuracy Measure

The accuracy of confusion matrix is 81.43 % and accuracy of the K-Nearest Neighbors Algorithm (KNN algorithm) is 87.07 %.

VI. REFERENCES

[1] Van Der Walt, Estée, and Jan Eloff. "Using machine learning to detect fake identities: bots vs humans." IEEE Access 6 (2018): 6540-6549.

[2] Efthimion, Phillip George; Payne, Scott; and Proferes, Nicholas (2018) "Supervised Machine Learning Bot Detection Techniques to Identify Social Twitter Bots," *SMU Data Science Review*: Vol. 1 : No. 2 , Article 5.

[3] Arzum Karatas, Serap Sahin, "A review on Social Bot Detection Techniques and Research Directions", ACSAC, Oct 2017, pp 1-3.

[4] Zafar Gilani, Ekaterina Kochmar, John Crowcraft, "Classification of Twitter Accounts into Automated Agents and Human Users", ACM/KDD, 2017, pp 489-496.

[5] S. Gurajala, J. S. White, B. Hudson, B. R. Voter, J. N. Matthews, "Profile characteristics of fake Twitter accounts", Big Data Soc., vol. 3, 2016, pp. 205-211.

[6] Nikan Chavoshi, Hossein Hamooni, Abdullah Mueen, "Identifying correlated bots in Twitter", SocInfo, 2016.

[7] R. J. Oentaryo, A. Murdopo, P. K. Prasetyo, and E.-P. Lim, ''On profiling bots in social media,'' in Proc. Int. Conf. Social Inform., 2016, pp. 92–109.

[8] C. Xiao, D. M. Freeman, T. Hwa, "Detecting clusters of fake accounts in online social networks", Proc. 8th ACM Workshop Artif. Intell. Secur., 2015, pp. 91-101.

[9] Zi Chu, Steven Gianvecchio, Haining Wang, " Detecting automation of twitter accounts: are you a human, bot or cyborg", ACSAC, 2012, pp. 811-824.

[10] K. Thomas, C. Grier, D. Song, and V. Paxson, ''Suspended accounts in retrospect: An analysis of Twitter spam,'' in Proc. ACM SIGCOMM Conf.Internet Meas. Conf., 2011, pp. 243–258.

Acknowledgement

We wish to express our sincere gratitude to **Dr. Sanjay U. Bokade, Principal** and **Dr. Satish. Y. Ket , H.O.D.** of Department Computer Engineering of Rajiv Gandhi Institute of Technology for providing us an opportunity to do our project work on **"Analysis of Twitter accounts for detecting bots and humans using Machine Learning "**.

This project bears on imprint of many peoples. We sincerely thank our project guide **Prof. Anita A. Lahane** for her guidance and encouragement in carrying out this synopsis work.

Finally, we would like to thank our colleagues and friends who helped us in completing project work successfully

1. Tanish Sanghavi
2. Shubham Shete
3. Aman Sarawgi