TABLE OF CONTENTS

DECLARATION

I, **, Aman Verma, Avinash Kumar Bharti, Himanshu Kumar, Kumari Priyanka, Singh Aman Prakash Bhai** hereby declare that this dissertation work entitled **"Twitter Bot Detection Using Machine Learning"**, has been carried out under the guidance of **Prof**. **Rani R Shetty**, Designation, ***Computer Science and Engineering department, S.D.M College of Engineering & Technology, Dharwad*****,** in partial fulfillment of the degree of ***Bachelor of Engineering in Computer Science and Engineering*** *from* ***Visvesvaraya Technological University, Belgaum***, during the academic year ***2018-19.***

I also declare that I have not submitted this dissertation to any other University for the award of any other degree.

Place: Dharwad.

Date:

NAME:

USN:

Signature of Student(s):

ACKNOWLEDGEMENT

I consider it a privilege to express my sincere gratitude and respect to all those who guided and inspired me throughout this project.

I express my heartfelt thanks to my guide **Prof. RANI R SHETTY,** Designation**,** Department of Computer Science and Engineering, S. D. M. College of Engineering and Technology, Dharwad, for his/her valuable guidance and suggestions during the course of this project. The successful completion of this project, owes to his/her coordination and streamlining of the project progress.

I extend my gratitude to our Project Coordinator(s) **Dr. U P Kulkarni, Prof. Ranganath G Yadawad, Prof. Nita G K and Prof. Indira R Umarji,** Department of Computer Science and Engineering, S. D. M. College of Engineering and Technology, Dharwad, for their valuable guidance and suggestions during the course of this project.

I am thankful to **Dr. S. B. Kulkarni,** Head of the Department, Department of Computer Science and Engineering, S. D. M. College of Engineering and Technology, Dharwad, for his encouragement and help throughout the BE course.

I acknowledge gratitude to **Dr. Shrikant B. Vanakudre,** Principal, SDM College of Engineering and Technology, Dharwad, for his timely help and inspiration during the tenure of the course.

I would like to thank all teaching and non-teaching staff of CSE department, SDMCET, for helping me to carry out my project successfully.

Last but not the least I am very much thankful to all my family members, classmates and friends for their valuable support during the period of my study.

Name	**USN**
Aman Verma	**2SD15CS009**
Avinash Kumar Bharti	**2SD15CS021**
Himanshu Kumar	**2SD15CS038**
Kumari Priyanka	**2SD15CS046**
Singh Aman Prakash Bhai	**2SD15CS125**

ABSTRACT

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.

LIST OF FIGURES

PROBLEM STATEMENT

The high-level goal of this project is to efficiently explore the evolving Twitter network and identify bots manifesting continually changing behavioral patterns. This is further subdivided into accessing the dataset, analyzing the dataset, implementing different machine learning algorithm, and finding the best model based on the trade-off. Further, training the dataset and testing the predictions on the Test set.

CHAPTER 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 variety of prohibited behaviors that violate the Twitter Rules." Those rules include among other things the type of behavior Twitter considers as spamming, such as:

- "Posting harmful links (including links to phishing or malware)
- Aggressive following behavior (mass following and mass unfollowing for attention), particularly by automated Means.
- Abusing the @reply or @mention function to post unwanted messages to users.
- Having a small number of followers compared to the number of people one is following
- Posting 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% to 7%. 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.

CHAPTER 2

LITERATURE SURVEY

There has been recent interest in the detection of malicious and/or fake users from both the online social networks and computer networking communities. For instance, Wang[4]looks at graph-based features to identify bots on Twitter, while Yang, Harkreader, and Gu [5] combine similar graph- based features with syntactic metrics to build their classifiers.Thomas et al. [6] use a similar set of features to provide a retrospective analysis of a large set of recently-suspended Twitter accounts.

Boshmaf et al. [7] instead create bots (rather than detecting them), claiming that 80% of bots are undetectable and that Facebook's Immune system [8] was unable to detect their bots. Lee, Caverlee, and Webb [9] create "honeypot" accounts to lure both humans and spammers into the open, then provide a statistical analysis of the malicious accounts they identified. In computer networks research, the detection of Sybil accounts in computer networks has been applied to social network data; these techniques tend to rely on the "fast mixing" property of a network—which may not exist in social networks [10]—and do not scale to the size of present-day social networks(e.g., SybilInfer [3] runs in time O(|V |2 log |V |), which is intractable for networks with millions users).

Most relevant is recent work by (Twitter employee and anti-spam engineer) Chu and colleagues [11], [12], which uses graph-theoretic, syntactic, and some semantic features to classify humans, bots, and cyborgs (human-assisted bots) in a Twitter dataset. Twitter has been widely used since 2006, and there are some related literature in twittering [12], [13], [14].

To better understand microblogging usage and communities, Java et al. [12] studied over 70,000 Twitter users and categorized their posts into four main groups: daily chatter (e.g., "going out for dinner"), conversations, sharing information or URLs, and reporting news. Their work also studied 1) the growth of Twitter, showing a linear growth rate; 2) its network properties, showing the evidence that the network is scale-free like other social networks [15]; and 3) the geographical distribution of its users, showing that most Twitter users are from the US, Europe, and Japan.

Krishnamurthy et al. [13] studied a group of over 100,000 Twitter users and classified their roles by follower-to- following ratios into three groups: 1) broadcasters, which have a large number of followers; 2) acquaintances, which have about the same number on either followers or following; and 3) miscreants and evangelists (e.g., spammers), which follow a large number of other users but have few followers.

Wu et al. [16] studied the information diffusion on Twitter, regarding the production, flow, and consumption of information.

Kwak et al. [17] conducted a thorough quantitative study on Twitter by crawling the entire Twittersphere. Their work analyzed the follower-following topology, and found nonpower-law follower distribution and low reciprocity, which all mark a deviation from known characteristics of human social networks.

Kim et al. [18] analyzed Twitter lists as a potential source for discovering latent characters and interests of users. A Twitter list consists of multiple users and their tweets. Their research indicated that words extracted from each list are representative of all the members in the list even if the words are not used by the members. It is useful for targeting users with specific interests.

In addition to network-related studies, several previous works focus on sociotechnological aspects of Twitter [7], [8], [19], [20], [21], such as its use in the workplace or during major disaster events. Twitter has attracted spammers to post spam content, due to its popularity and openness. Fighting against spam on Twitter has been investigated in recent works [14], [22], [23].

Yardi et al. [14] detected spam on Twitter. According to their observations, spammers send more messages than legitimate users, and are more likely to follow other spammers than legitimate users. Thus, a high follower-to- following ratio is a sign of spamming behavior.

Grier et al. [22] investigated spam on Twitter from the perspective of spam and click-through behaviors, and evaluated the effectiveness of using blacklists to prevent spam propagation. Their work found out that around 0.13 percent of spam tweets generate a visit, orders of magnitude higher than click-through rate of 0.003-0.006 percent reported for spam e-mail.

Exploiting the social trust among users, social spammers achieve a much higher success rate than traditional spam methods.

Thomas et al. [23] studied the behaviors of spammers on Twitter by analyzing the tweets originated from suspended users in retrospect. They found that the current marketplace for Twitter spam uses a diverse set of spamming techniques, including a variety of strategies for creating Twitter accounts, generating spam URLs, and distributing spam.

CHAPTER 3

DETAILED DESIGN

3.1 System 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.

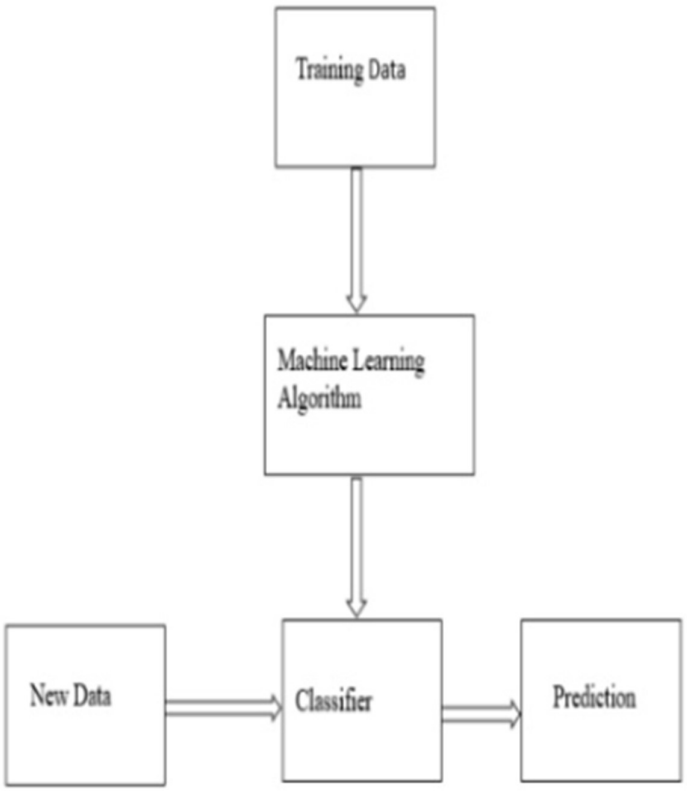


Fig 3.1 System Diagram

3.2 Data Flow Diagram

3.2.1 Level-0

Our Dataset contains 20 attributes out of which we have selected 8 attributes based on the spearman correlation**.** Some of the important attributes are Follower-Friend ratio, Username, URL ratio, Number of tweets, etc. which provide information about the users. These attributes are then used by our twitter bot detection system or Machine Learning Algorithms for predicting that whether a user is a human or a bot.

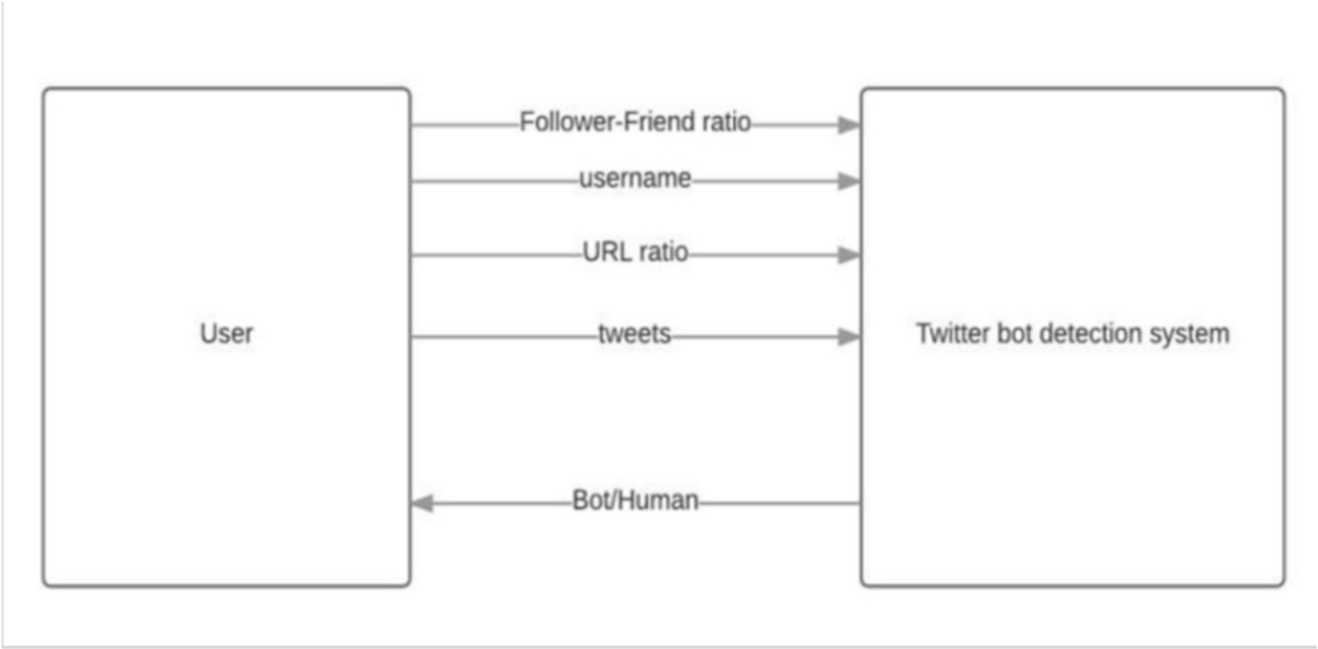


Fig 3.2.1 Data Flow Diagram(Level 0)

3.2.2 Level-1

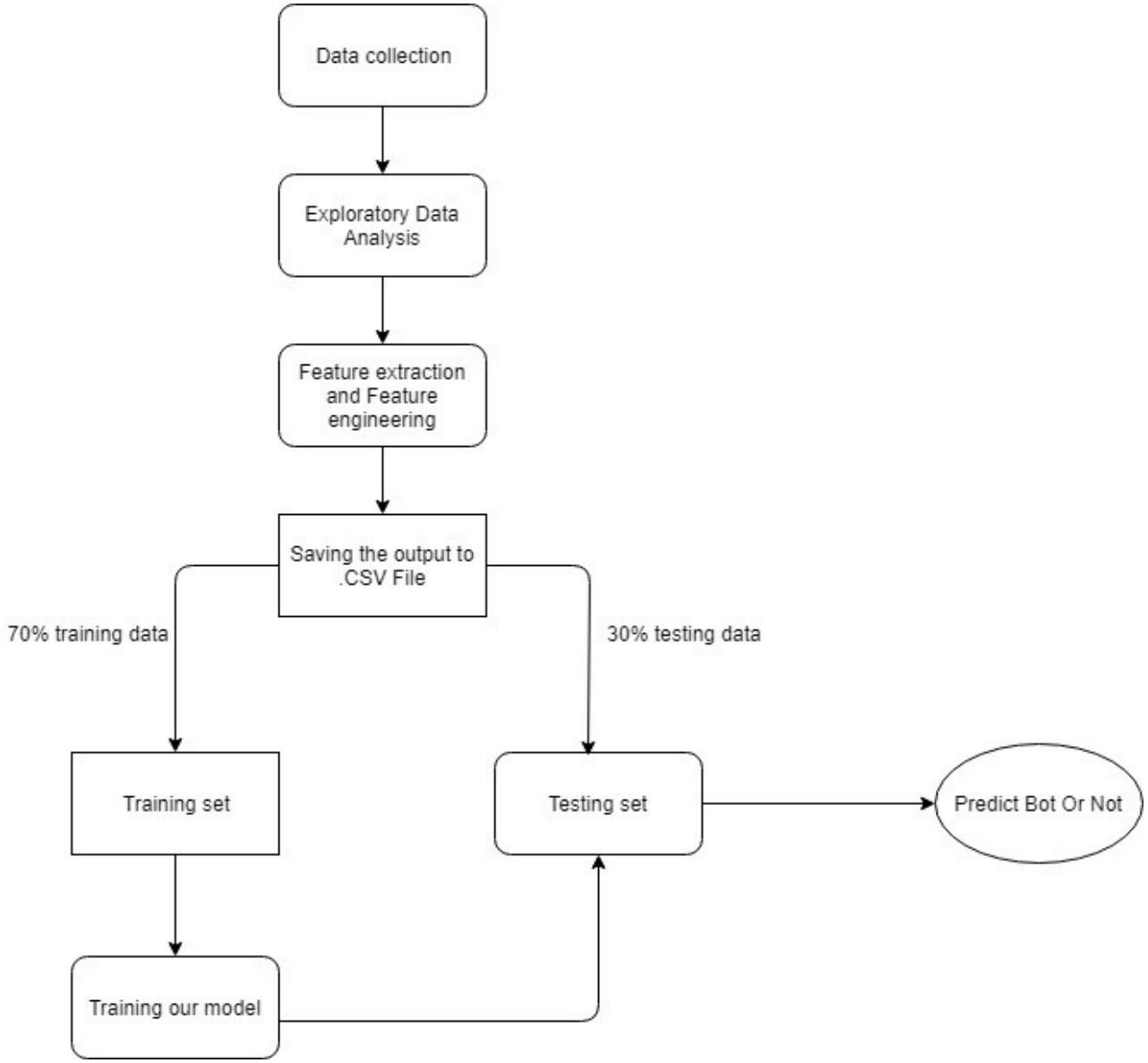


Fig 3.2.2 Data Flow Diagram (Level 1)

In order to use machine learning to identify fake twitter accounts, we needed a labelled collection of users, pre classified as fake or genuine. We get the real time data (dynamic data) from tweepy API which consist of 2798 training set and 578 test set. The dataset is divided into 70%(training set) and 30%(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.

Now next step is to train the model using training set, Attribute considered here are important which we've considered using spearman correlation, Some of the important attributes are

- Listed_count
- verified account
- Friends to follower ratio

Now test set is used on our already trained model for prediction whether the user is bot or not.

3.3 Activity Flow Diagram

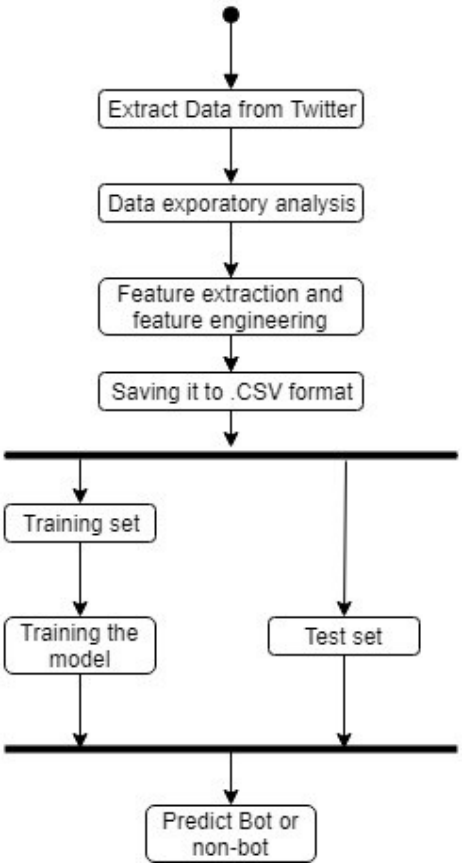


Fig 3.3 Activity Flow Diagram

CHAPTER 4

PROJECT SPECIFIC REQUIREMENTS

This project is developed using Python as development tool. Python version 3.6 is used for this purpose. The project dataset is downloaded from Kaggle and kept in the working system. For development environment , 'Spyder' is installed. Spyder is a powerful scientific environment written in Python, for Python, and designed by and for scientists, engineers and data analysts. It offers a unique combination of the advanced editing, analysis, debugging, and profiling functionality of a comprehensive development tool with the data exploration, interactive execution, deep inspection, and beautiful visualization capabilities of a scientific package. All the required python libraries such as tweepy, pandas, numpy, matplotlib, seaborn are installed so as to provide the functionalities of python in-built functions.

4.1 System Configuration

Hardware Requirements:

System: Intel i5 2.4 GHz

RAM: 4 GB

Hard Disk: 1 TB

Software Requirements:

Operating System: Windows 7 or above

Coding Language: Python

Software: Anaconda

4.2 Python 3

Python is one of the most common language used for machine learning because it is easy to learn, enormous packages it supports like scikit learn that contains the implementation of common machine learning algorithms, built-in library supports, pre-processed models, support for larger networks and huge toolset.

4.2.1 Benefits of Python

- Scikit learn- It features various classification, regression and clustering algorithms including support vector machines, random forest, gradient boosting, K-means and DBSCAN, and is designed to interoperate with the python numerical and scientific libraries NumPy and SciPy.
- Data frame object for data manipulation with integrated indexing
- Tools for reading and writing data between in memory data structures and different file formats.
- Data alignment and integrated handling of missing data.
- Reshaping and pivoting of datasets.
- Label based slicing, fancy indexing and sub setting of large data set.
- Data structure column insertion and deletion.
- Group by engine allowing split-apply-combine operations on dataset.
- Data set merging and joining
- Hierarchical axis indexing to work with high-dimensional data in a lower-dimensional data structure.

4.3 Anaconda Software

Anaconda is a python distribution, with installation and package management tools. It provides large selection of packages and commercial support. It is an environment manager, which provides the facility to create different python environments, each with their own settings.

Anaconda can help with:

- Python can be installed over the multiple platforms
- Different environments can be supported separately
- Distributing with not having correct privileges and
- Support and running with specific packages and libraries

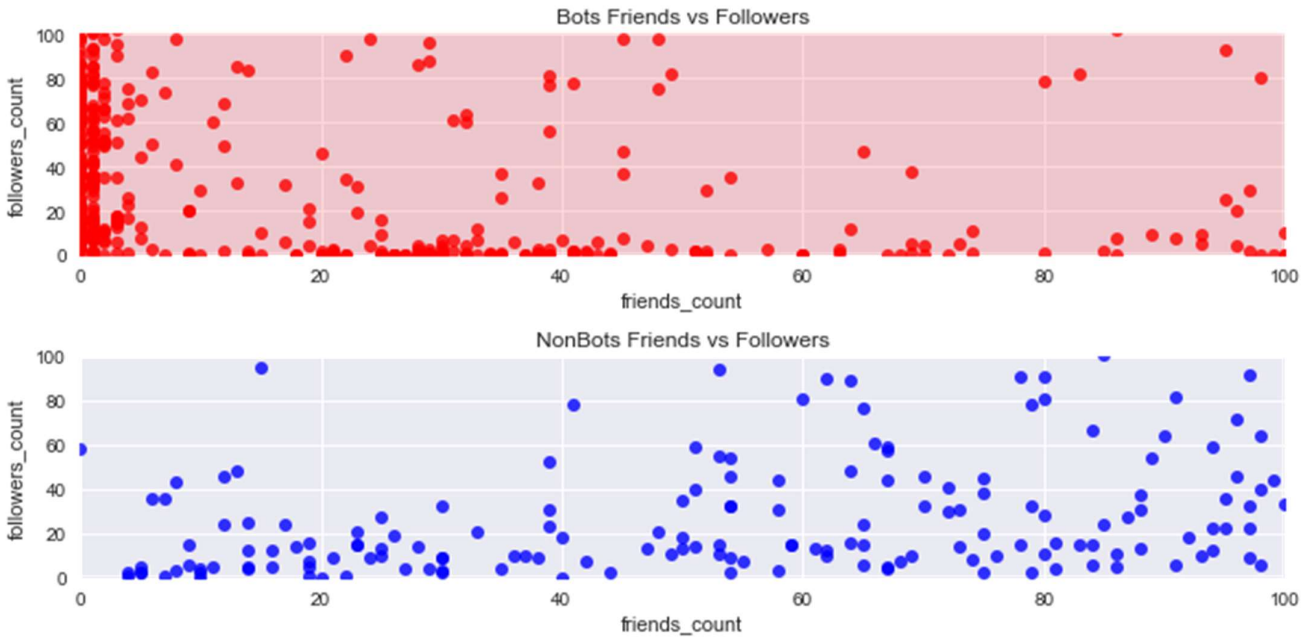
CHAPTER 5

IMPLEMENTATION

5.1 Data Analysis

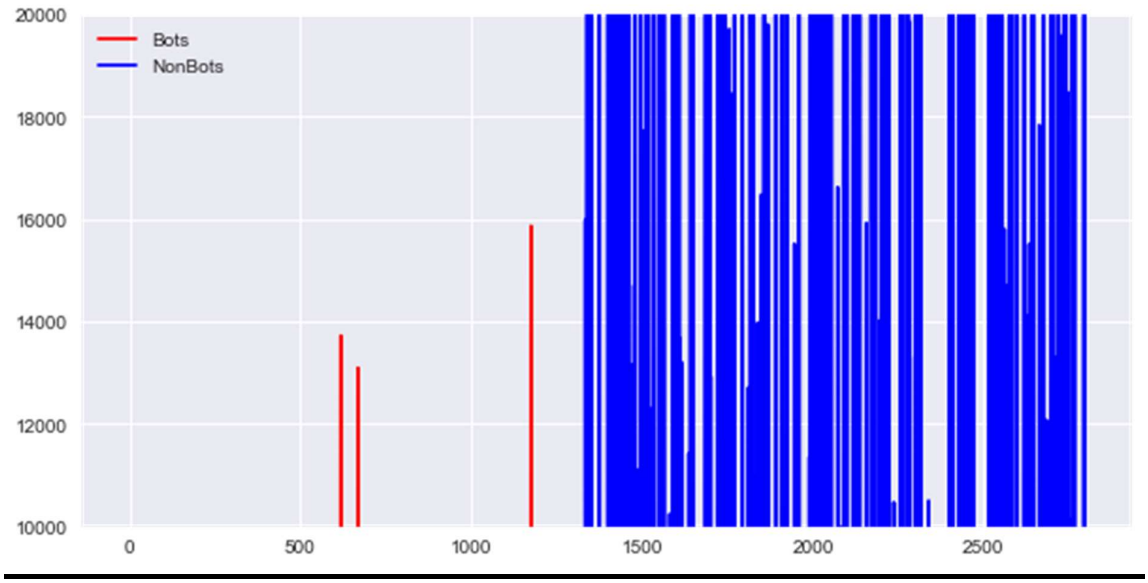
5.1.1 Identifying Missingness in the data

We did Exploratory Data Analysis. In which we found out all the *NULL* values in the dataset. This is the heatmap of all *NULL* values.It is evident from the heatmap that most user doesn't have their location,description or URL mentioned in their profile.



Above are two plots for 'Bots friends vs Followers' and 'Non-Bots Friends vs followers', They depict the general trend on the characteristics of bots i.e They possess more followers count compared to friends. On the other hand, Non-Bots have generally an equal number of both friends and followers with some slight variance.

5.1.2 Identifying Imbalance in the data



Whenever the listed count is between 10,000 to 20,000 there are 5% of the Bots and 95% Non-Bots which are real users.

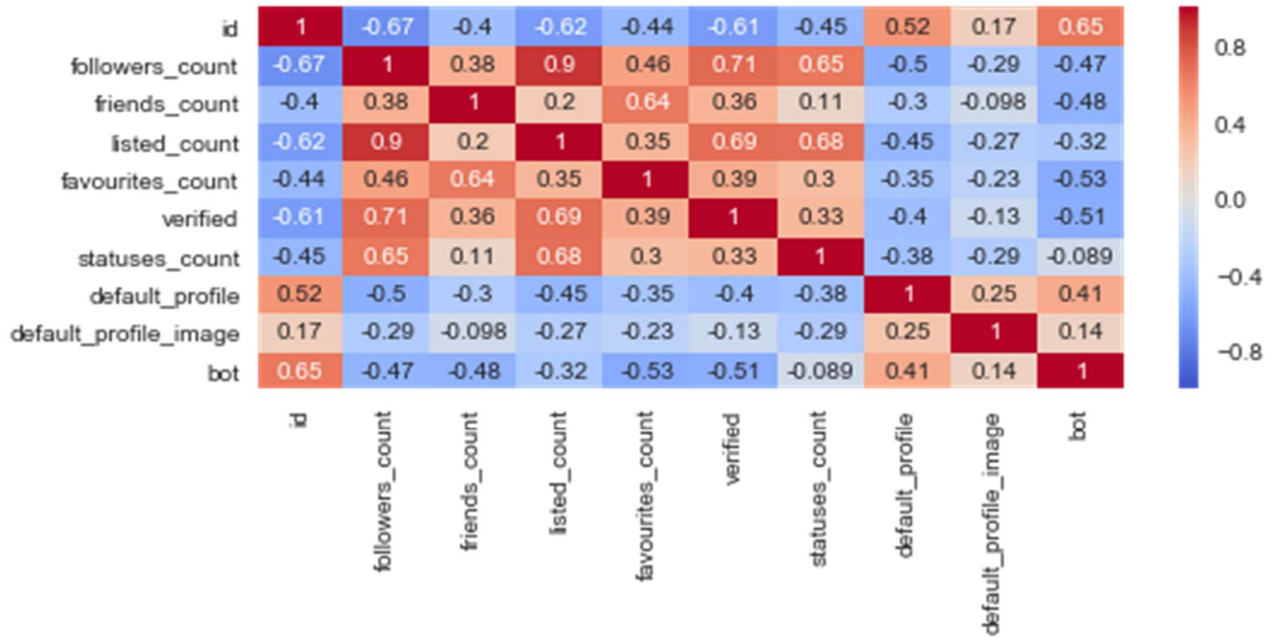
5.2 Feature Independence using Spearman correlation

Spearman's correlation coefficient is a statistical measure of the strength of a monotonic relationship between paired data. In a sample, it is denoted by *'rs'* and is by design constrained as *-1<=rs<=1* and its interpretation is similar to that of Pearsons, e.g. the closer is to the stronger the monotonic relationship. 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"

Out[29]:

	id	followers_count	friends_count	listed_count	favourites_count	verified	statuses_cou
id	1.000000	-0.672925	-0.402346	-0.615005	-0.439430	-0.611899	-0.451945
followers_count	-0.672925	1.000000	0.375522	0.896126	0.457363	0.709732	0.649117
friends_count	-0.402346	0.375522	1.000000	0.204403	0.641529	0.356452	0.111118
listed_count	-0.615005	0.896126	0.204403	1.000000	0.349059	0.694340	0.684976
favourites_count	-0.439430	0.457363	0.641529	0.349059	1.000000	0.394227	0.295108
verified	-0.611899	0.709732	0.356452	0.694340	0.394227	1.000000	0.333278
statuses_count	-0.451945	0.649117	0.111118	0.684976	0.295108	0.333278	1.000000
default_profile	0.522990	-0.496899	-0.296358	-0.447376	-0.348043	-0.404650	-0.375918
default_profile_image	0.166601	-0.293838	-0.097607	-0.269035	-0.226956	-0.132298	-0.289999
bot	0.652131	-0.468430	-0.483105	-0.318445	-0.526228	-0.508555	-0.089018



Spearman Correlation Result

- There is no correlation between id, statuses_count, default_profile, default_profile_image and the target variable.
- There is a strong correlation between verified, listed_count, friends_count, followers_count and the target variable.
- We cannot perform correlation for categorical attributes. So we will take screen_name, name, description, status into feature engineering. While use verified, listed_count for feature extraction.

5.3 Implementing Different Models

We implement different machine learning model and hence find their accuracy on test and training set. We will also plot the ROC curve which is a graphical plot created by plotting the true positive rate against the false positive rate at the various threshold which depicts the performance of models.

5.3.1 Decision Tree Classifier

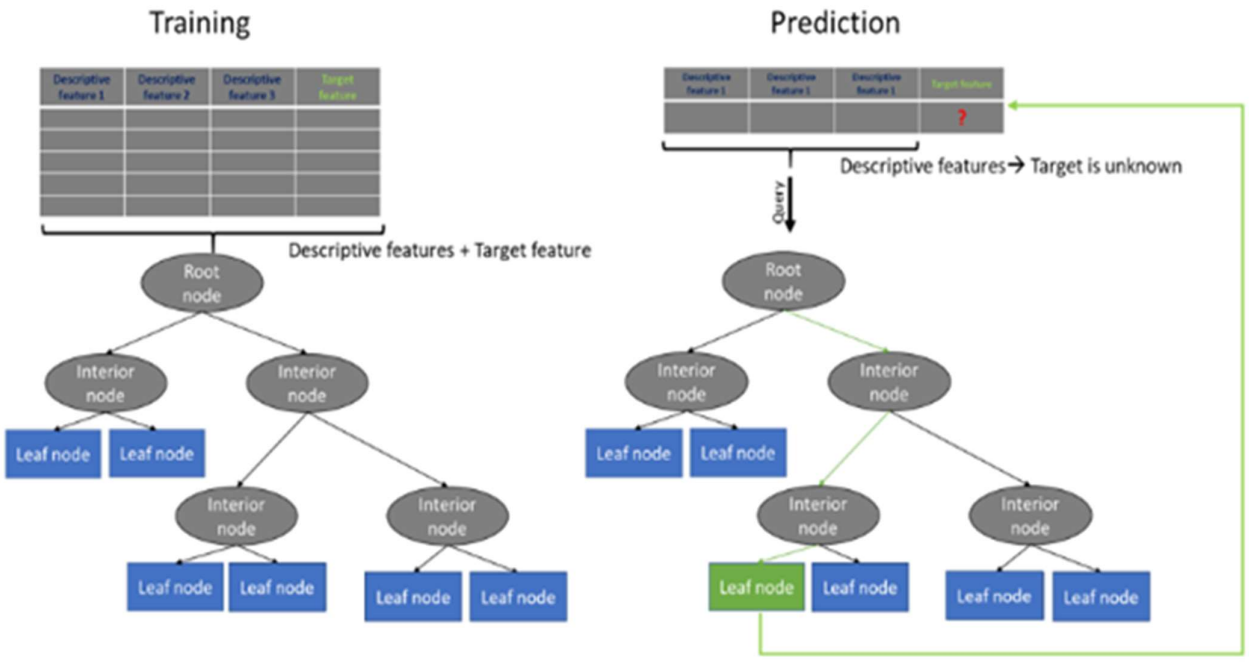
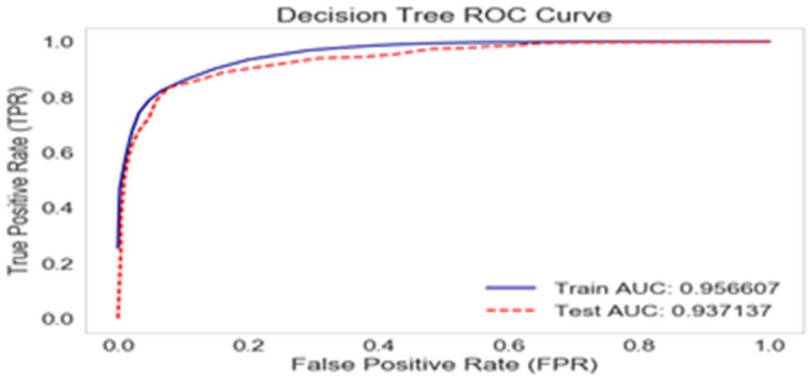


Fig 5.3.1 Decision Tree Classifier

Accuracy

Training Accuracy: 0.88247 Test Accuracy: 0.87857



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.

Construction of Decision Tree

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.

Decision Tree Representation

Decision trees classify instances by sorting them down the tree from the root to some leaf node, which provides the classification of the instance. An instance is classified by starting at the root node of the tree, testing the attribute specified by this node, then moving down the tree branch corresponding to the value of the attribute as shown in the above figure. This process is then repeated for the subtree rooted at the new node.

5.3.2 Random Forest Classifier

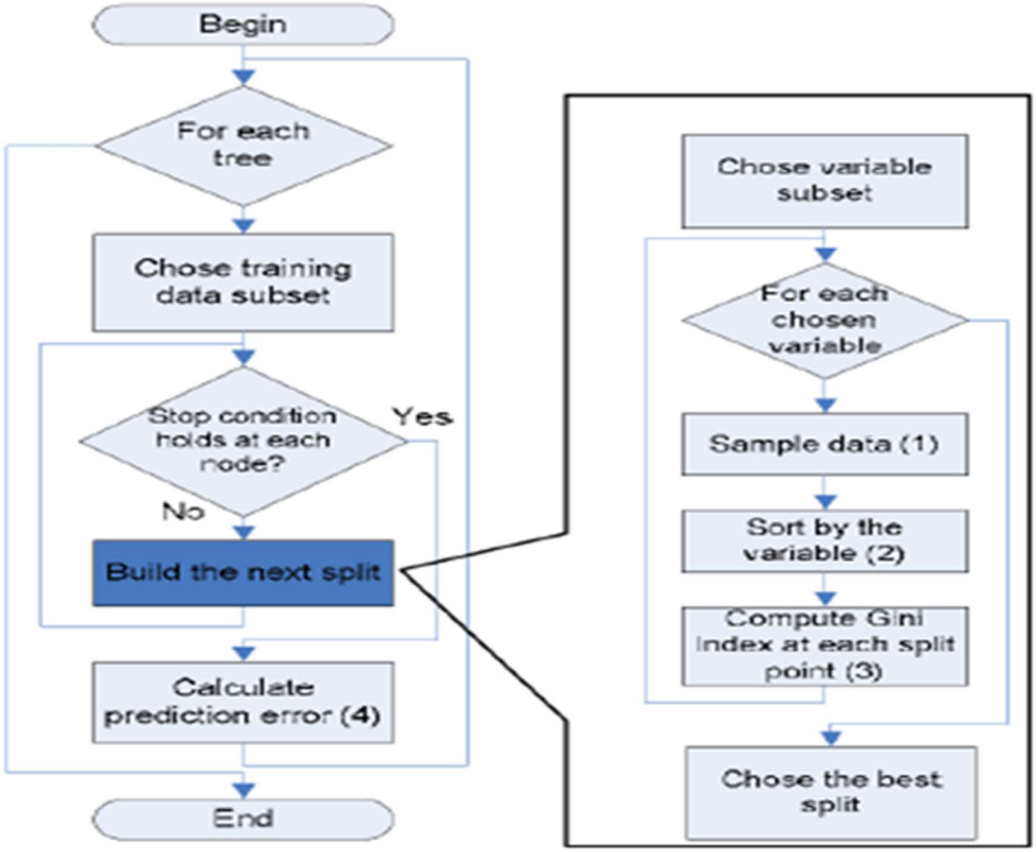
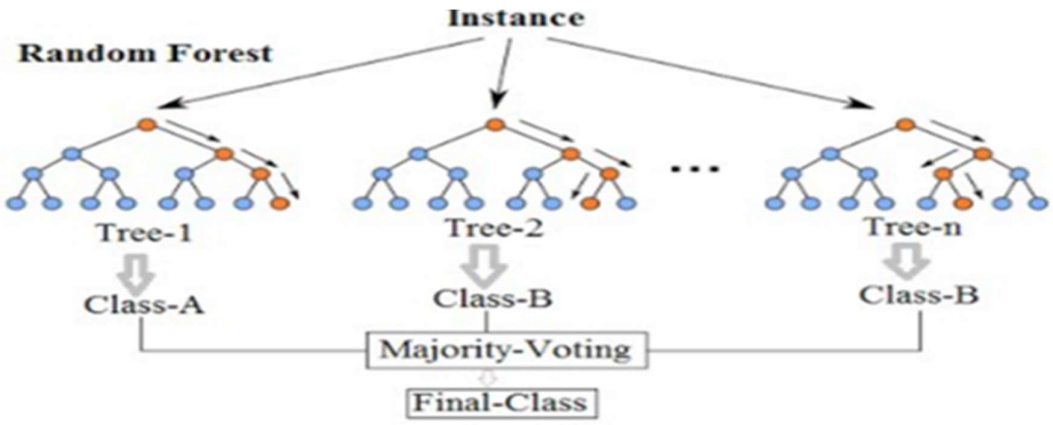


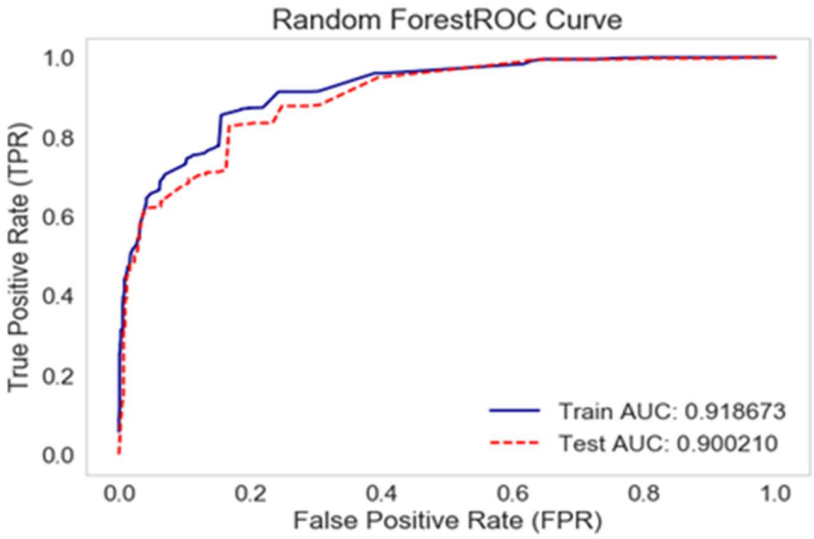
Fig 5.3.2 Random Forest Classifier



Accuracy

Training Accuracy: 0.82524 Test Accuracy: 0.79167

ROC CURVE



Random Forest:

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.

How it works

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 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.

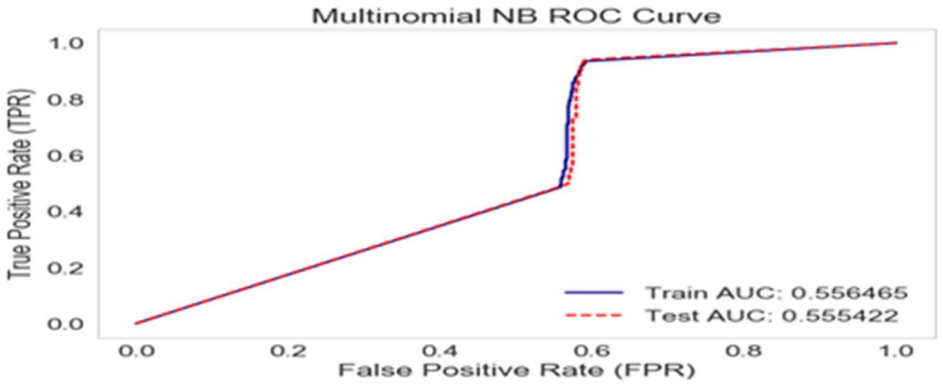
5.3.3 Multinomial Naive Bayes Classifier

Accuracy

Training Accuracy: 0.54216

Test Accuracy: 0.56310

ROC CURVE



The simplest solutions are usually the most powerful ones, and Naïve Bayes is a good proof of that. In spite of the great advances of the Machine Learning in the last years, it has proven to not only be simple but also fast, accurate and reliable. It has been successfully used for many purposes, but it works particularly well with natural language processing (NLP) problems.

Naive Bayes is a family of probabilistic algorithms that take advantage of probability theory and Bayes' Theorem to predict the category of a sample (like a piece of news or a customer review). They are probabilistic, which means that they calculate the probability of each category for a given sample, and then output the category with the highest one. The way they get these probabilities is by using Bayes' Theorem, which describes the probability of a feature, based on prior knowledge of conditions that might be related to that feature.

Bayes' Theorem

Now we need to transform the probability we want to calculate into something that can be calculated using word frequencies. For this, we will use some basic properties of probabilities, and Bayes' Theorem.

Bayes' Theorem is useful when working with conditional probabilities because it provides us with a way to reverse them:

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)}$$

.

5.4 Our Proposed Classifier

Step 1:Creating copy of dataframe in train_set

Step 2:Converting id to int

Step 3:Replacing Null values with 0 in Friends_count column

Step 4:Replacing Null values with 0 in Followers_count column

Step 5:Preparing bag of words for bot accounts

Step 6:Converting verified account into vectors (True->1 False->0)

Step 7:If the name,description,screen_name,status columns contains bot ,

then Store the data set in predicted_set_1

7.1:Assign bot column as 1 (BOT)

7.2:Store the rest data set in verified_set for next step

Step 8:For all verified account, Store the data set in predicted_set_2

8.1:Assign bot column as 0 (NON_BOT)

8.2:Store the rest data set in followers_following_set for next step

8.3:predicted_set_1=:concatenate(predicted_set_1,predicted_set_2)

Step 9:If followers_count is less than 50 and statuses_count greater than 1000 , then then Store the data set in predicted_set_2

9.1:Assign bot column as 1 (BOT)

9.2:Store the rest data set in followers_retweet_set for next step

9.3:predicted_set_1=:concatenate(predicted_set_1,predicted_set_2)

Step 10:If followers_count is less than 150 and statuses_count greater than 10000 then then Store the data set in predicted_set_2

10.1:Assign bot column as 1 (BOT)

10.2:predicted_set_1=:concatenate(predicted_set_1,predicted_set_2)

Step 11:Store the rest data set in predicted_set_2 and assign bot column as 0 (NON_BOT)

11.1:predicted_set_1=:concatenate(predicted_set_1,predicted_set_2)

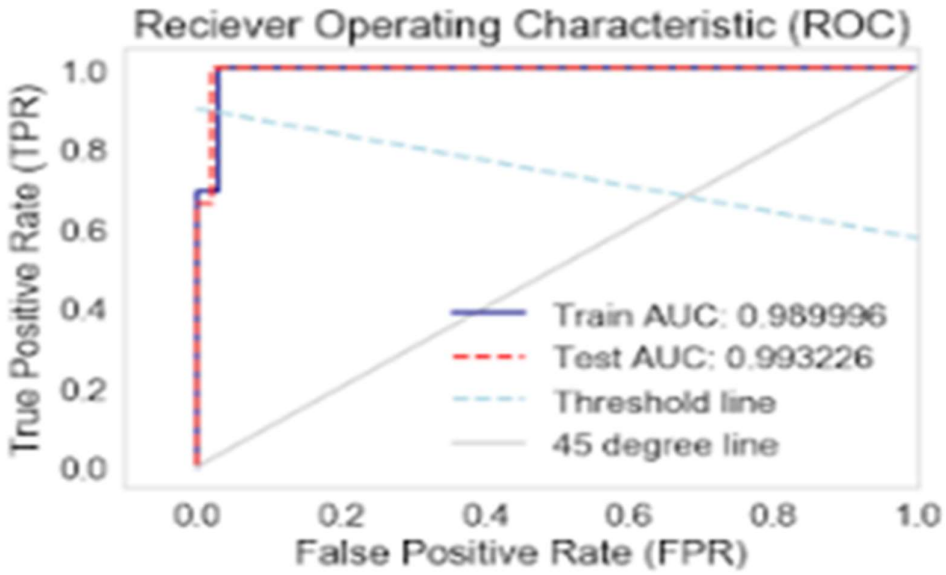
Step 12:Return predicted_set_1

Accuracy

Train Accuracy: 0.964627151052

Test Accuracy: 0.938596491228

Predicted results saved to submission.csv. File shape: (575, 2)



CHAPTER 6

TESTING

6.1 Dataset

The Dataset was taken from the twitter(Using Tweepy).

Twitter is a popular social network where users share messages called tweets. Twitter allows us to mine the data of any user using Twitter API or Tweepy. The data will be tweets extracted from the user. The first thing to do is get the consumer key, consumer secret, access key and access secret from twitter developer available easily for each user. These keys will help the API for authentication.

Steps to obtain keys:

- Login to twitter developer section
- Go to “Create an App”
- Fill the details of the application.
- Click on Create your Twitter Application
- Details of your new app will be shown along with consumer key and consumer secret.
- For access token, click ” Create my access token”. The page will refresh and generate access token.

6.2 Features of Dataset

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

1. ID: These IDs are unique 64-bit unsigned integers, which are based on time, instead of being sequential. The full ID is composed of a timestamp, a worker number, and a sequence number. When consuming the API using JSON, it is important to always use the field id_str instead of id.
2. Friends_count: It depicts that the friends_count should be in proper ratio with the follower_count.
3. Follower_count: As the friends_count is dependent on follower count,these two attributes are strongly correlated.
4. Listed_count: A Twitter list is really a list of people on Twitter that are somehow connected. They might belong to a certain category (i.e. news organizations), or they might be connected through their content (i.e. related to gardening), or they might even be connected through an event that they're all attending (i.e. the Oscars).
5. Favourite_count: It's the number of tweets that given user has marked as favorite.
6. Verified: An account may be verified if it is determined to be an account of public interest.
7. Statuses_count: Returns the most recent Tweets authored by the authenticating user that have been retweeted by others.
8. Default_profile: Profile that who has not given enough information on the profile are most likely belongs to bot or spammers.
9. Default_profile_image: spammers and people who harass others often use accounts without a profile image.

CHAPTER 7

RESULTS

The final output from the classifier will be of the form as shown below-The class label 'bot' represent whether a given user is bot or a real user. The entry 1 represent a bot and 0 represent a real user i.e. non-bot.

Index	id	bot
572	218833868	1
574	3119554528	1
0	2281292622	0
2	765871267	0
9	88856792	0
11	1566746503	0
12	90420314	0
13	184910040	0
14	157690631	0
15	42420346	0
20	31348594	0
21	122085859	0
22	23573083	0
23	43152482	0
24	188857501	0
26	35094637	0
27	146252766	0
29	55117855	0
31	234837526	0
36	19058681	0
39	515652246	0

CHAPTER 8

CONCLUSION AND FUTURE SCOPE

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. We implemented various algorithm but finally implemented our own which gave us AUC>95%.

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.

CHAPTER 9

REFERENCES

[1] “Top Trending Twitter Topics for 2011 from What the Trend,” http://blog.hootsuite.com/top-twitter-trends-2011/, Dec. 2011.

[2] “Twitter Blog: Your World, More Connected,” http://blog. twitter.com/2011/08/your-world-more-connected.html, Aug. 2011.

[3] Alexa, “The Top 500 Sites on the Web by Alexa,” http:// www.alexa.com/topsites, Dec. 2011.

[4] “Amazon Comes to Twitter ,”http://www.readwriteweb.com/amazon_comes_to_twitter.

[5] “Best Buy Goes All Twitter Crazy with @Twelpforce,”http://twitter.com/in_social_media/

[6] “Barack Obama Uses Twitter in 2008 Presidential Campaign,”http://twitter.com/Obama/, Dec. 2009.

[7] J. Sutton, L. Palen, and I. Shlovski, “Back-Channels on the Front Lines: Emerging Use of Social Media in the 2007 Southern California Wildfires,” Proc. Int’l ISCRAM Conf., May 2008.

[8] A.L. Hughes and L. Palen, “Twitter Adoption and Use in Mass Convergence and Emergency Events,” Proc. Sixth Int’l ISCRAM Conf., May 2009.

[9] S. Gianvecchio, M. Xie, Z. Wu, and H. Wang, “Measurement and Classification of Humans and Bots in Internet Chat,” Proc. 17th USENIX Security Symp., 2008.

[10] B. Stone-Gross, M. Cova, L. Cavallaro, B. Gilbert, M. Szydlowski, R. Kemmerer, C. Kruegel, and G. Vigna, “Your Botnet Is My Botnet: Analysis of a Botnet Takeover,” Proc. 16th ACM Conf. Computer and Comm. Security, 2009.

[11] S. Gianvecchio, Z. Wu, M. Xie, and H. Wang, “Battle of Botcraft: Fighting Bots in Online Games with Human Observational Proofs,” Proc. 16th ACM Conf. Computer and Comm. Security, 2009.

[12] A. Java, X. Song, T. Finin, and B. Tseng, “Why We Twitter: Understanding Microblogging Usage and Communities,” Proc. Ninth WebKDD and First SNA-KDD Workshop Web Mining and Social Network Analysis, 2007.

[13] B. Krishnamurthy, P. Gill, and M. Arlitt, “A Few Chirps about Twitter,” Proc. First Workshop Online Social Networks, 2008.

[14] S. Yardi, D. Romero, G. Schoenebeck, and D. Boyd, “Detecting Spam in a Twitter Network,” First Monday, vol. 15, no. 1, Jan. 2010.

[15] A. Mislove, M. Marcon, K.P. Gummadi, P. Druschel, and B. Bhattacharjee, “Measurement and Analysis of Online Social Networks,” Proc. Seventh ACM SIGCOMM Conf. Internet Measurement, 2007.

[16] S. Wu, J.M. Hofman, W.A. Mason, and D.J. Watts, “Who Says What to Whom on Twitter,” Proc. 20th Int’l Conf. World Wide Web, pp. 705-714, 2011.

[17] H. Kwak, C. Lee, H. Park, and S. Moon, “What Is Twitter, a Social Network or a News Media?” Proc. 19th Int’l Conf. World Wide Web, pp. 591-600, 2010.

[18] I.-C.M. Dongwoo Kim, Y. Jo, and A. Oh, “Analysis of Twitter Lists as a Potential Source for Discovering Latent Characteristics of Users,” Proc. CHI Workshop Microblogging: What and How Can We Learn From It?, 2010.

[19] D. Zhao and M.B. Rosson, “How and Why People Twitter: The Role Micro-Blogging Plays in Informal Communication at Work,” Proc. ACM Int’l Conf. Supporting Group Work, 2009.