Twitter Bot Classification

Aishwarya Ajay Das
Computer Science
NY, USA
Aishwarya.Das@nyu.edu

Ajay Anil Thorve
Computer Science
NY, USA
ajaythorve@nyu.edu

Abstract

In today's world where twitter bots are a common phenomenon, it becomes extremely important to keep a tab on them in order to suppress the influence they have on the humans using twitter. These influences can be created using various methods like repeated spamming of same content, spreading rumours, creating fake followers to increase credibility of an account. In order to keep a tab on them, we need to identify them first. This paper focusses on the methods of identifying twitter bot accounts using various Machine Learning Techniques and testing it on the real world data to see how the different techniques perform against each other.

Keywords

Machine Learning, Twitter API, Twitter bots

I. Introduction

Twitterbot[7] according to wikipedia is a bot program used to produce automated posts on the Twitter microblogging service, or to automatically follow Twitter users. Recently a study[2] conducted in 2017 at University of Southern California suggests that 9 to 15% of Twitter accounts are bots controlled by softwares instead of humans. While some of these bots are definitely beneficial like dissemination of news in critical times, many of these can be used for malicious activities such as promoting terrorist propaganda and influencing the opinion of citizens in general. One of the examples of this is the recently concluded US elections where around 1 million[6] automated tweets were recorded between 1st and 2nd debate which were in favour of Contenders Hillary and Trump. Evidently enough, social network was a big part of how the US elections 2016 panned out eventually. Such malicious influences need to be controlled and the first step in achieving this is identifying if the accounts that are bots.

In this paper, we study the problem of identifying bots on Twitter. There are many factors involved in determining if an account is bot or not, like, if they are telling you they are a bot, or if they tweet the same thing to everybody or if their source is an API. We consider all these factors to identify bots, and we use different machine learning techniques to train and test our data. While doing so, we go through some previous work done in the similar domain in section 3. In section 4 and 5, we describe the dataset we used in this project, and the Machine Learning techniques we tried to classify the accounts.

We also compare the results of different techniques based on factors such as accuracy, precision, recall, time taken for training and testing. Based on these observations, we try and find the most suited technique for this particular problem. Finally, we look at some of the potential areas of application of this classification.

II. Motivation

Many of the Twitter bots serve a commercial purpose like bots designed to promote products, brands and political candidates. But some of them are influential bots[1]-realistic, automated identities that illicitly shape discussions on social media sites like Twitter and Facebook, posing a risk to freedom of expression. For example, the terrorist group ISIS has in the past used social media to spread radicalism by influencing youth to embrace their cause; an opinion piece[3] in Forbes asserted that Russia waged a social media disinformation campaign in the aftermath of Russian actions in the Ukraine;[4] and computer science students at the Technical University of Denmark built social bots that had a surprisingly large influence on the twitter world[5].

Recently the trends of creating and using bots to automate sharing tweets, following accounts, posting same content everywhere,etc. have been increasing. So we propose creating a way to identify such accounts accurately and efficiently. Later these accounts can used to identify if they are good bots or bad bots.

III. Related Work

[8]To classify bots, we need to understand the various factors that might help us identifying it. Some of the common signals of identifying Bots are: 1) They are telling you they are bot in their description , 2) Getting a direct response on your tweet,

Table 1: Classes of features employed by feature-based systems for social bot detection

Class	Description
Network	*Network features* captures various dimensions of information diffusion patterns. Statistical features can be extracted from networks based on *retweets, mentions, and hashing co-occurrence.* Examples include degree *distribution, clustering coefficient, and centrality measures.*
User	*User features* are based on Twitter meta-data related to an account, including language, geographic locations, and account creation time.
Friends	*Friend features* include descriptive statistics relative to an account's social contacts, such as median, moments, and entropy of the distribution of their numbers of followers, followees, and posts.
Timing	*Timing features* capture temporal patterns of content generation (tweets) and consumption(retweets); examples include the similarity to a Poisson process, or the average time between two consecutive posts.
Content	*Content features* are based on linguistic cues computed through natural language processing, especially part-of-speech tagging; examples include the frequency of verbs, nouns, and adverbs in tweets.
Sentiment	*Sentiment features* are built using general-purpose and Twitter-specific sentiment analysis algorithms, including happiness, arousal-dominance-valence, and emotion scores

3) Large amount of following, small amount of followers, 4) They tweet the same thing to everybody, 5) The follow/ unfollow game, 6) Duplicate profile pictures, 7) Coming from an API. Bots can be identified if observed carefully considering above mentioned factors.

[9]In this paper the authors mention about the various algorithm used by them to create a social bot detection system. They divide it into three major classes 1) Graph-based, 2) Crowd-sourcing ,3) Feature-based. They after comparing the performance of all the three and also an hybrid consisting of the above three methodology, mention that the performance of the Feature-based was the best. They then classify the features according to the one mentioned in the table above.

[15]Analysing the twitter data to classify accounts as bots or not can be achieved by some careful observation of this data. This articles helps in identifying some very important patterns which helps distinguish bots from human accounts. Some of these patterns are number of followers and number of followings. If we consider these two features which can be obtained from the user metadata from twitter API, some obvious patterns can be observed. Similarly, lexical diversity can also be calculated of the accounts' tweet content and inferences can be made with previously conducted research which says that humans have a lexical diversity factor of 0.7. Given that most bots have a lexical diversity close to 1(unique set of vocabulary in each tweet), this can also be an important feature in classification. The author also provides some insights into techniques that are a workaround to avoid hitting twitter API maximum request limits while pulling data.

IV. Data

The dataset that we considered had a mix of a number of accounts, some human and some bots and consisted of the following attributes for each of them:

{id, id_str, screen_name, location, description, url, followers_count, friends_count, listed_count, created_at, favourites_count, verified, statuses_count, lang, default_profile, status, default_profile_image, has_extended_profile name, bot, desc_bot, Age}

A sample of this data can be found here: SampleData (https://github.com/AjayThorve/ML-project/blob/master/Stage%202/Data/Sample.csv)

All the columns are extracted from the Twitter API via the tweepy module in python, except the columns bot, desc_bot and Age.The column bot specifies whether the account is a bot or not. The column desc_bot is created at run time and is a boolean field which has a value of 1 (if the word "bot" exists in the description) or 0 (if the word "bot" doesn't exist in the description). The column Age is created at run time and is an integer field which has a value equal to the age of the account in days.

The Input provided to our system would be a list of Twitter Handles and if they are a bot or human. The system would make its prediction and return the accuracy of the results. To do this first the system will convert it into the SampleData Format as mentioned above. Then it will extract the features that are relevant to our models and put it in the model to finally display out the result.

The initial data that we took into consideration is approximately 2300 records which has been split in the ratio

of 70:30[12][13]. 70% of the data is used train the model and the rest 30% is used to test the model.

V. Algorithms Used

We started by selecting features that we thought gave some insights. We did some statistical analysis to zero down on the features. Some of the patterns are shown below.

The chart (Figure 1) below shows the the behaviour of Human and Bots in terms of the account that follow them. As we can see Human account have more followers on an average.

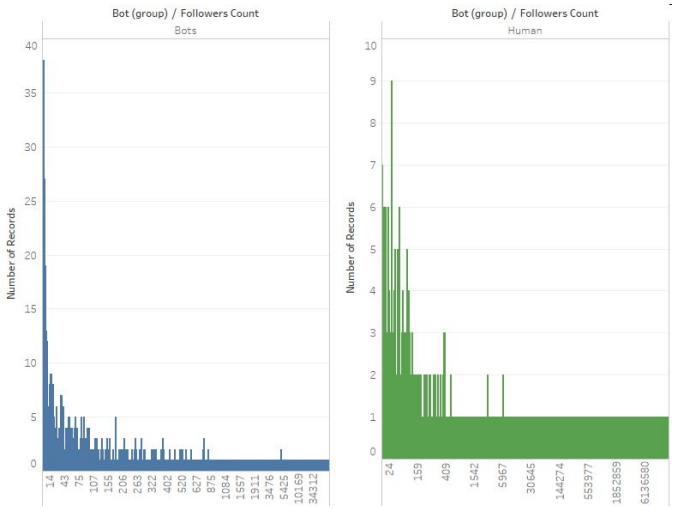

Figure 1: Followers count(Bots and Humans)

The chart (Figure 2) below shows the the behaviour of Human and Bots in terms of the account they follow. As we can see Humans follow other account more on an average.

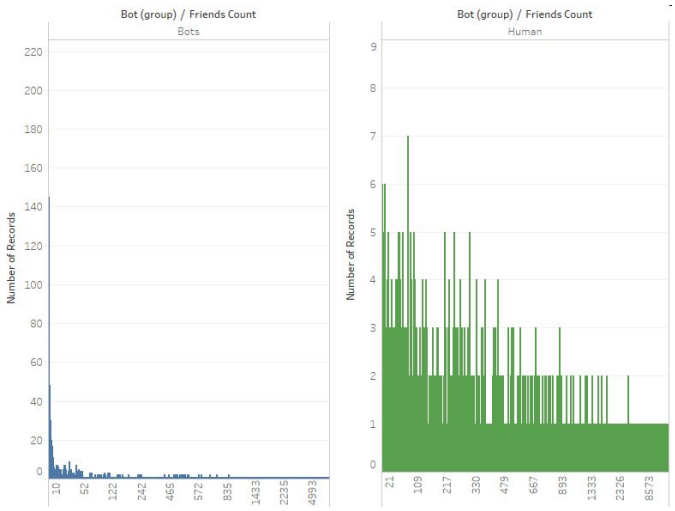

Figure 2: Friends count(Bots and Humans)

The chart (Figure 3) shows the the behaviour of Human and Bots in terms of if they are verified or not. As we can see bots are not verified but human accounts are verified.

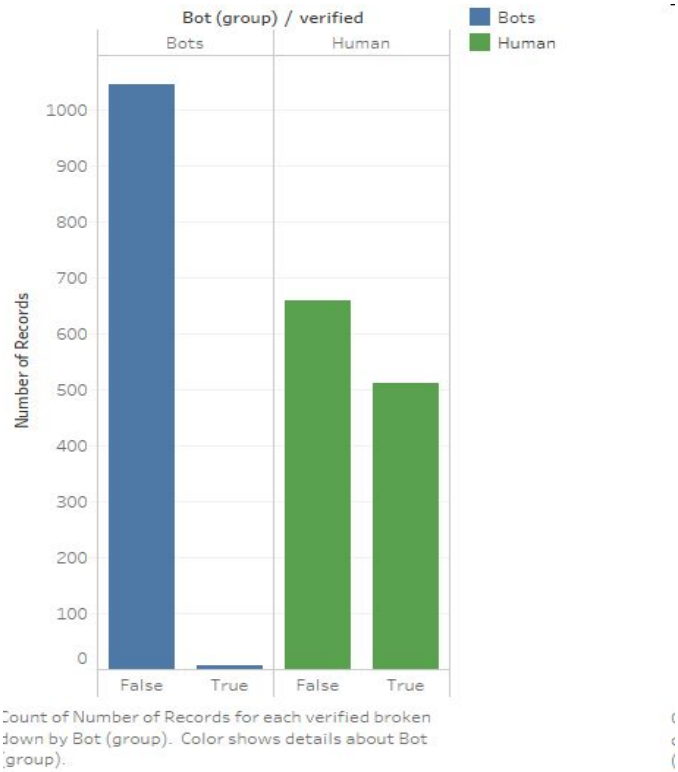

Figure 3: Verified (Bots and Humans)

The chart (Figure 4) shows the the behaviour of Human and Bots in terms of if they have default profile pic. As we can see Human tend to have different profile pic as people like to customize their pic.

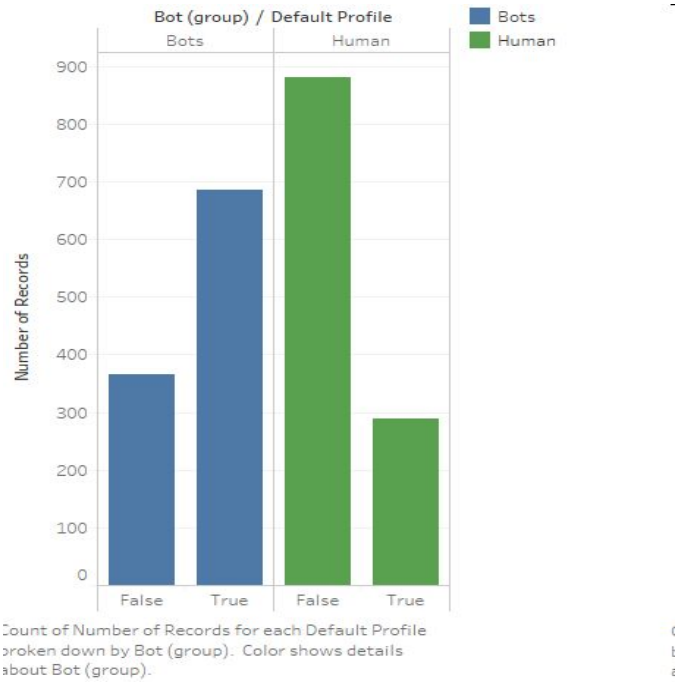

Figure 4: Default profile (Bots and Humans)

The chart (Figure 5) shows the the behaviour of Human and Bots in terms of if they have the word bot in their description. As we can see Human don't tend to have the word bot in their description.

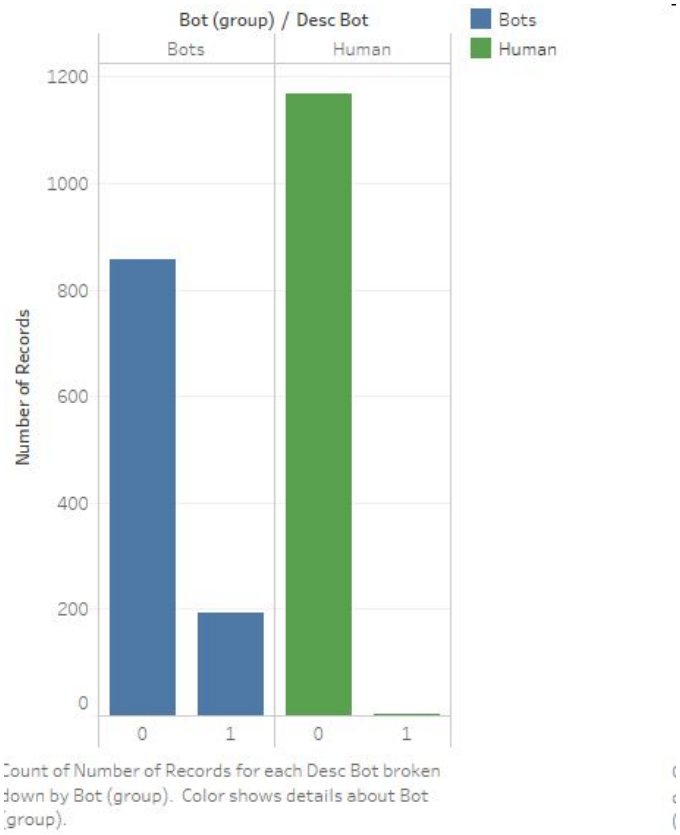

Figure 5: Description_ bot (Bots and Humans)

After narrowing down the features we went on to decide on which features are important to our analysis[10][11][14]. We implemented the recursive feature extraction technique and the extra tree classification to understand the relative importance of each attribute.

After we got the relative importance of the feature we tested various different algorithm like Naive Bayes, Support Vector Machines, Decision Trees, Random Forests and AdaBoost. The performance of each of these algorithms is mentioned in the result section.

We plan on adding more techniques to this list as and when we come across any relevant ones throughout the course of this project. Also we will try and include more features in our initial feature analysis and see if there are more important feature that are not yet discovered by us.

VI. Result

[11]After implementing Recursive Feature Elimination technique and Feature Importance property functions in Sklearn, we get a clear picture how each feature contributes to our classifier. This can be seen in Figure 6, where we can conclude that Age, Verified, friends_count, statuses_count, favourites_count, listed_count, followers_count are the features with most possible impact on our classifier.

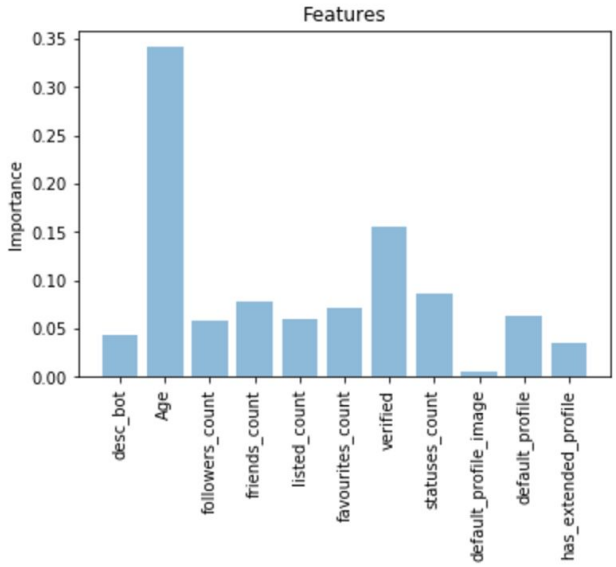

Figure 6: Feature Importances

After implementing the algorithms mentioned in previous section, we found out parameters like Accuracy, Precision, Recall, f1_score, AUC score for each of the algorithms. Table 2 provides a comparison of all of those algorithms based on the 5 measuring parameters. Decision Trees, Random Forest and Adaboost techniques gave the best results.

Table 2: Algorithms used and its comparisons

	Accuracy	Precision	Recall	f1_score	AUC
Naive Bayes	0.684685	0.804321	0.98503	0.758065	0.68378
SVM	0.503003	0.75299	0.008982	0.017804	0.504491
Decision Tree	0.891892	0.92107	0.88024	0.890909	0.891927
Random Forest	0.900901	0.933205	0.862275	0.897196	0.901017
Adaboost	0.912913	0.934462	0.916168	0.913433	0.912903

We also measured the Receiver Operating Characteristic (ROC) curve area to compare the ML algorithms Naive Bayes, Decision Tree, Random Forest and Adaboost to determine the best classifier. The following figures show the ROC curve plotter using Python libraries, Matplotlib and sklearn.

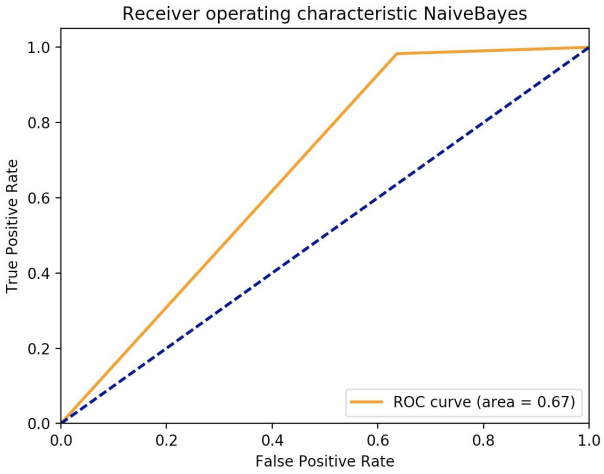

Figure 7: ROC Curve- Naive bayes

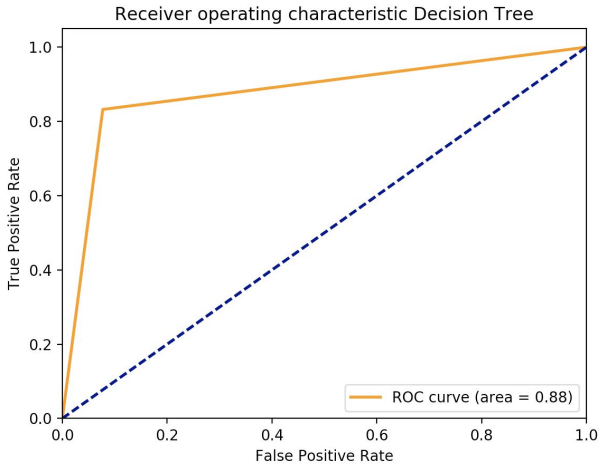

Figure 8: ROC Curve- Decision Tree

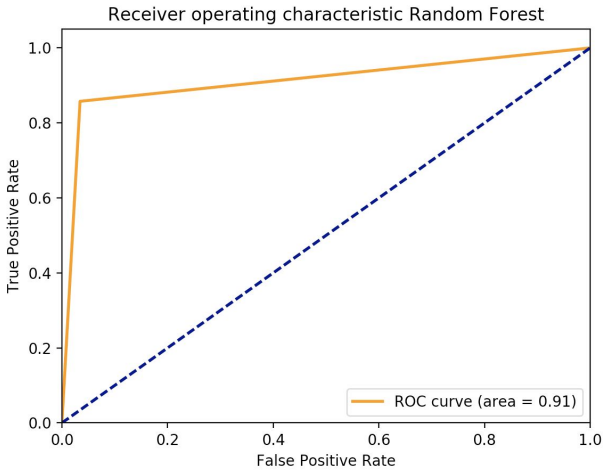

Figure 9: ROC Curve- Random Forest

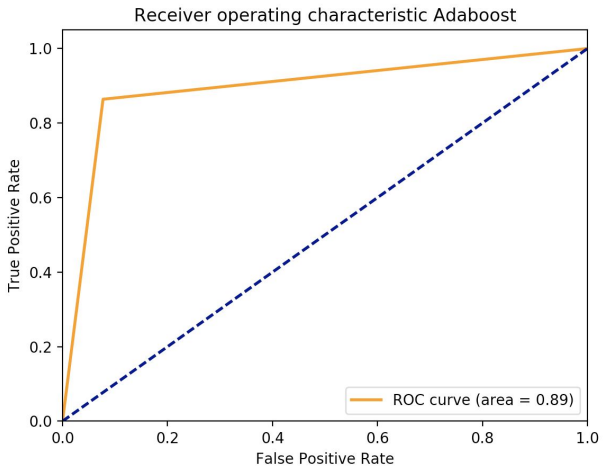

Figure 10: ROC Curve- Adaboost

Looking at the comparisons, it's safe to say that Random Forest and Adaboost work the best given the current features. However, we plan to use more algorithms through the course of this semester, like Artificial Neural Network and also work on features generated using tweets of the users by techniques such as sentiment analysis and lexical diversity of the tweet content.

VII. Code

Here is the link to our code (**https://github.com/AjayThorve/ML-project/tree/master/Stage%202**). The main file which will be invoked at run time is index.py. Then some prepossessing is done to add columns relevant to our analysis. Then we calculate the relative importance of each of them is calculated and the most important feature are used for all the different algorithms as mentioned above.

VIII. Video Link

(to be filled in the future)

IX. Evaluation

(to be filled in the future)

X. Conclusion

(to be filled in the future)

Acknowledgment

(to be filled in the future)

References

[1] V.S. Subrahmanian, Amos Azaria, Skylar Durst, Vadim Kagan, Aram Galstyan, Kristina Lerman, Linhong Zhu, Emilio Ferrara, Alessandro Flammini, Filippo Menczer,IEEE The DARPA Twitter Bot Challenge

[2] Onur Varol, Emilio Ferrara, Clayton A. Davis, Filippo Menczer, Alessandro Flammini, Online Human-Bot Interactions: Detection, Estimation, and Characterization

[3] S. Shane and B. Hubbard, "ISIS Displaying a Deft Command of Varied Media," The New York Times, 20 Aug. 2014; www.nytimes.com/2014/08/31/world/middleeast/isis-displaying-a-deft-command-of-varied-media.html.

[4] P.R. Gregory, "Inside Putin's Campaign of Social Media Trolling and Faked Ukrainian Crimes," Forbes, 11 May 2014; www.forbes.com/sites/paulroderickgregory/2014/05/11/inside-putins-campaign-of-social-media-trolling-and-faked-ukrainian-crimes.

[5] S. Lehmann and P. Sapieżyński, "You're Here Because of a Robot,"blog,4Dec.2013; http://sunelehmann.com/2013/12/04/youre-here-because-of-a-robot.

[6] D. Guilbeault and S. Woolley, "How Twitter Bots Are Shaping the Election", 1Nov.2016; https://www.theatlantic.com/technology/archive/2016/11/election-bots/506072/

[7] https://en.wikipedia.org/wiki/Twitterbot

[8] Bas van den beld, "How to recognize Twitter bots: 7 signals to look out for ", 20Aug.2012; http://www.stateofdigital.com/how-to-recognize-twitter-bots-6-signals-to-look-out-for/

[9] Emilio Ferrara, Onur Varol, Clayton Davis, Filippo Menczer, Alessandro Flammini "The Rise of Social Bots", 6Mar.2017; https://arxiv.org/pdf/1407.5225.pdf

[10] Mark A. Hall, Lloyd A. Smith, "Feature Selection for Machine Learning: Comparing a Correlation-based Filter Approach to the Wrapper"

[11] Jason Brownlee, "Feature Selection in Python with Scikit Python";http://machinelearningmastery.com/feature-selection-in-python-with-scikit-learn/

[12] https://www.researchgate.net/post/Is_there_an_ideal_ratio_between_a_training_set_and_validation_set_Which_trade-off_would_you_suggest

[13] http://information-gain.blogspot.com/2012/07/why-split-data-in-ratio-7030.html

[14] http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.ExtraTreesClassifier.html#sklearn.ensemble.ExtraTreesClassifier

[15] Erin Shellman, "Bot or Not: an end-to-end data analysis in python"; 17Aug.2015; http://www.erinshellman.com/bot-or-not/