Unmasking Bot Activity: Identifying Automated Tweets in the Ukraine-Russia Conflict Discourse

Cebrail Durna - 30124142, Victor Han - 30112492, Callum Matheson - 30066858 and
Logan Perry-Din - 30070661
University of Calgary, SENG 550 - Scalable Data Analytics, Fall 2023
Dr. Diwakar Krishnamurthy

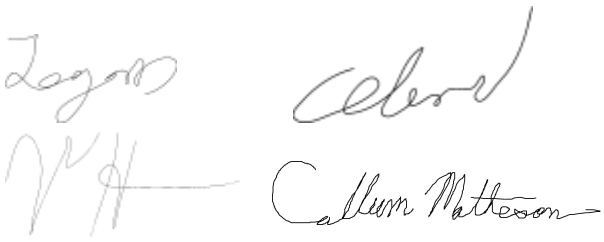
PREAMBLE

Contributions

All members contributed equally to this project and worked well together. Analysis and discussion of data was a key part of the development of this project. Distribution of work is 25% for all four members. Specific contributions from individual members include but are not limited to:

- Logan: Initial loading of data, cleaning, and visualization. Feature extraction.
- Callum: Development of heuristic evaluation and testing; feature extraction.
- Victor: Research and development of ML model. Feature extraction and normalizing data.
- Cebrail: Aggregation of features, development of ML model, feature extraction and normalizing data.

Signed Declaration



Source Code

The following repository contains all source code and data used for this project:
https://github.com/Logan-PD/SENG-550-Project

The following kaggle link directs to the original dataset, distributed with a creative commons CC0 free use license:
https://www.kaggle.com/datasets/bwandowando/ukraine-russian-crisis-twitter-dataset-1-2-m-rows

Abstract **- This paper examines the feasibility of classifying twitter accounts as bots or humans, without a label to train a model on, with a specific lens on the Russian-Ukraine conflict. Through exploratory data analysis, visualizations, research and discussion, a set of features that could indicate a bot account was generated. By processing the raw data using PySpark methods, features such as average sentiment, use of hashtags and volume of tweets are normalized into numeric features. These features have then been inputted into a naive weighted heuristic evaluation that outputs a numeric score that represents bot likelihood. These same features have also been used in a k-means machine learning model where accounts are clustered and grouped, revealing which accounts are likely bots, which are not, and which are uncertain. Due to a lack of labels in the dataset to classify accounts as bots or not, metrics were evaluated by inspecting accounts manually. Accounts such as "FuckPutinBot" who were clearly bots were used to compare scores with more uncertain accounts. The scores from the heuristic evaluation are finally compared with the clusters generated from the model.**

Index Terms - Big Data, Machine Learning, Twitter, PySpark

INTRODUCTION

What is the Problem You Selected?

Using automated bots to perform actions in mass has been a malicious yet effective strategy when proper security measures are not in place. These bots are used for spreading misinformation and creating irrelevant noise around sensitive content. This project will explore the possibility of predicting which tweets are from bots and which are not, in particular, tweets about the Ukraine - Russian war.

Why is it an Important Problem?

The United States Senate has confirmed that Russia used bot farms to influence the 2016 election. During the Russo-Ukraine war, Ukraine has already claimed to have taken down a Russian bot farm. Bot farms are becoming a popular and problematic issue in geopolitical events. Therefore the ability to accurately recognize bot activity becomes invaluable. Any company vulnerable to a bot farm would benefit from the ability to detect and remove bots from their platform. In our case, Twitter being able to accurately remove the influence of bot farms owned by state actors would minimize the effectiveness of malicious efforts. This project will explore the possibility of predicting which tweets are from bots and which are not.

What Have Others Done in this Space?

Given the significance of this problem that modern social media faces, many researchers and groups have put considerable effort into exploring bots and their behavior. Numerous papers have been published on how to detect bots on twitter[1][2][3], of which some have been referenced to justify choices made in this project. Prior to recent changes to X's API, websites such as 'Botometer' could allow individuals to probe the likelihood of an account being a bot on a scale from 1-10[4]. Recently, researchers from the University of Adelaide published a paper on the language used by bot accounts when interacting with human users which used the same data set as this project[5].

What are Some Existing Gaps that You Seek to Fill?

Many of the aforementioned publications and tools in this field made use of data only accessible from the old Twitter API. The two forms of bot detection algorithms presented in this project serve as a way to detect bots with data still available with the new Twitter API.

Furthermore, this project aims to look at the process as a whole of identifying bots from unlabeled data using a naive method, as well as detecting patterns using a k-means machine learning algorithm.

What are Your Data Analysis Questions?

This project aims to explore the feasibility of predicting whether a twitter account is likely a bot. Based on metrics about the account itself, and by analyzing the tweets of the account. Our research questions are:

- What are the largest factors in determining an automated account?
- Is it possible to simply use a heuristic to flag accounts?
- Can a clustering algorithm effectively find bots?

With these questions we hope to gain insight into the process of data science on raw data for a real world use.

What are you Proposing and what are your Main Findings?

This project is an investigation into the feasibility of using raw data to identify bots. Specifically, the feasibility of a naive linear combination of numeric features to output a score, as well as an unsupervised clustering algorithm, k-means, to classify twitter accounts as bots or not, or other categories of bot likelihood. We are comparing our results from our ML model with our heuristic evaluation to assess the accuracy of both. Our main finding is that while it does seem feasible to classify accounts as bots or not, more work is required to build a consistent classifier. While the classification itself requires more research, this paper shows how it is feasible to extract features from raw twitter data.

Background and related work

Review of Existing Work Pertinent to your Project

In 2015 the U.S. Government's Defense Advanced Research Projects Agency (DARPA) founded a bot detection program. This is seen by many as being the first instance of a nation recognizing the issue of social media bots sometimes referred to as Sybils. In 2016 DARPA hosted "The DARPA Twitter Bot Challenge" where six teams competed to find the most effective way of identifying bot accounts.

Since then there has been research showing that machine learning is an effective way to find bots on twitter. This study highlighted the weights and features used by their model to detect bots. Weights presented in this paper have greatly contributed to the weights used in the heuristic model in this project. Kantepe and Ganiz 2017[2].

Other research has shown that a wide variety of tweet related features can be used to find bots. Of note, the publication found that bots had a much higher tendency to negative sentiment where human users had a much higher usage of positive sentiment. Alarfaj et al. 2023[3].

Some outcomes-based inquiries have explored the language used by bots to control narratives with a specific lens on the ongoing Russo-Ukrainian conflict. Smart et al. 2022[5].

Methodology

This project was completed in three stages: 1) cleaning the data and extracting features, 2) running our heuristic evaluation and ML model, and 3) evaluating and comparing the results of our heuristic evaluation and model.

Experiment Setup

The bulk of the work for this project was analyzing and cleaning our dataset. Our dataset contains data about tweets concerning the Russian-Ukraine war. Each row is an individual tweet from a user with columns specifying the tweet text, time of tweet, as well as information for that user such as name, followers and following count, hashtags, and other various columns. Our general process was then to extract features from the raw data and aggregate the values such that each account has its own associated set of numeric, normalized features.

A big challenge for this project was the fact that our dataset did not contain any labels for bot accounts, we therefore had to be creative in our feature extraction and decide how to determine which accounts were bots. This made our project more realistic in terms of working with raw, uncleaned and uncurated data to solve a complex

problem. We decided to employ a heuristic evaluation of our features as a naive approach, using a weighted linear combination equation. Then we used an unsupervised k-means machine learning model to find patterns of accounts that could classify them as bots. Focusing on the heuristic evaluation and k-means model individually, we tuned them to get the best results before comparing the two.

Analyzing and exploring the raw data using databrick's data profiling functionality, we found certain characteristics from accounts such as "FuckPutinBot", could be used as factors for identifying bots. While "FuckPutinBot" is certainly a bot, it is not exactly the type of bot we were trying to identify. Our focus was more on bots pretending to be humans, while "FuckPuinBot" clearly states itself as a bot. However, we decided that some of the characteristics of accounts like "FuckPutinBot" could be used for identifying other, more malicious bots.

While analyzing the data, we decided on several factors that would influence the likelihood of a certain account being labeled a bot. By performing an exploratory data analysis on the raw data, creating visualizations and discussing what characteristics a bot account would likely have when compared to a human account, we created a set of features as inputs to our heuristic evaluation and k-means model.

These factors/features are:

- Positivity and Negativity of Tweets
- Average Tweets per Day and Total Tweets
- Account Age
- Followers and Following Count
- Ratio of Tweets about Conflict vs Not
- Frequency of Common Hashtags

Our reasoning for associating these features with bot likelihood are as follows.

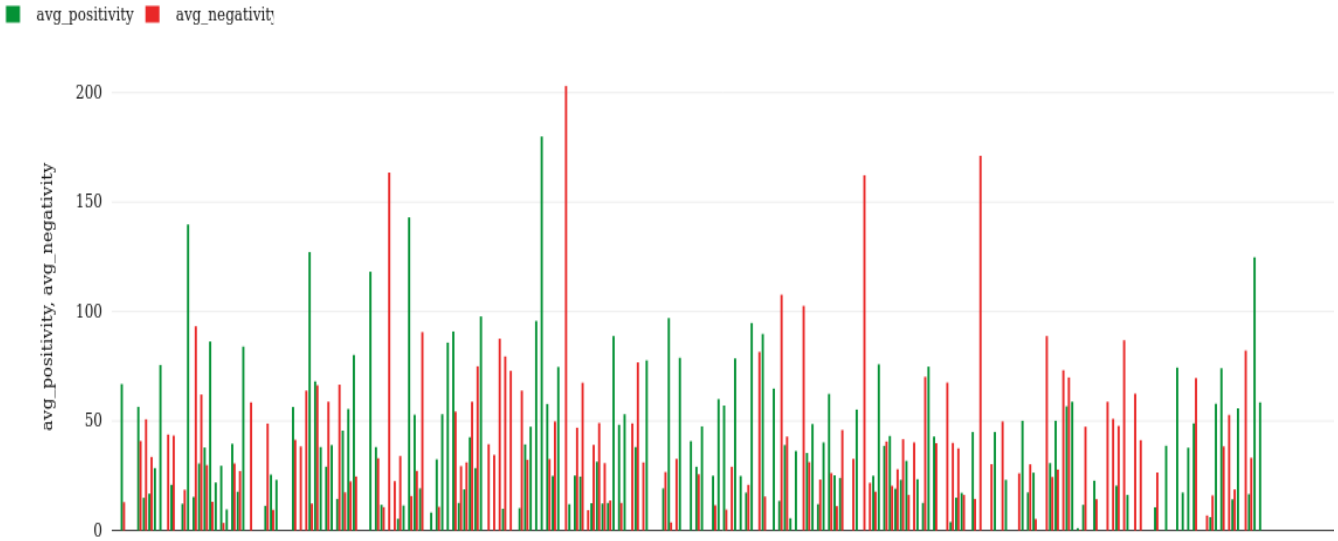
I. *Positivity and Negativity*

To attack other cultures, governments might use tactics such as automated tweets to spread negative messages about those cultures. Therefore, we consider the negativity of tweets as well as their positivity a crucial factor in identifying potential bot accounts. To determine the average negativity of an account's tweets, we used library VADER (Valence Aware Dictionary and sEntiment Reasoner) to produce numerical scores based on the text of tweets.

Figure I shows the normalized average negativity and positivity score for a subset of accounts. Each bar on the x-axis is a different user account. From this we can see how some accounts have a much higher positive and/or negative sentiment than average.

Figure I

AVERAGE NEGATIVITY AND POSITIVITY FOR SUBSET OF USER ACCOUNTS



II. *Tweets per Day*

A high volume of tweets over a short period raises red flags. A consistent tweet pattern, such as one tweet per hour, strongly suggests a bot account - it is extremely unlikely any human would tweet in this manner. The timing of tweets may also tell us more, but that introduces the challenge of assuming the time zone and typical work routines of individuals from the area.

Figure II represents these abnormal number of tweets from a user. The x-axis bars each represent a user.

FIGURE II
COUNT OF TWEETS FOR SUBSET OF USER ACCOUNTS

III. *Account Age*

The Age of an account is also a great indicator of a genuine account. An account that is over a decade old is extremely unlikely to be a bot created to spread misinformation and sentiment for the Ukraine-Russian war.

IV. *Followers and Following*

This includes the number of followers and people following. An account with practically no followers or following but many tweets would raise suspicion. On the other hand, a high number of both may indicate a real account. We should be able to pick up on patterns related to this.

V. *Ratio of Tweets About Conflict vs. Tweets About Other Topics*

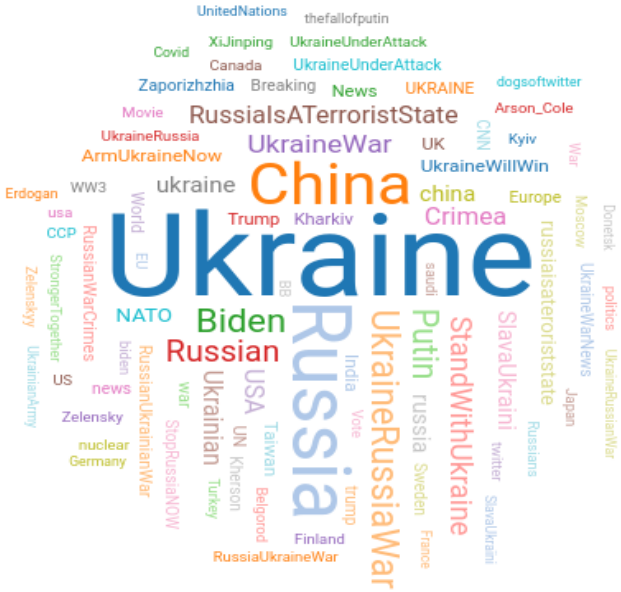
Assuming tweets are correctly pulled in this dataset, the ratio of total tweets vs tweets about the conflict from a user should indicate how often the user has tweeted about the Russo-Ukraine war. For instance, if the numTweets column is 5000 in one row, and 5050 in another row, that means between the time of extraction, there have been 50 tweets from that user. If we only have 5 rows from this user, that means that 5 out of 50 tweets from this user have been about the war.

VI. *Frequency of Common Hashtags*

During our research, a common metric that came up was the frequency of a hashtag. We are computing the count of the most used hashtag. For instance if an account uses the hashtag "FuckPutin" hundreds of times in our dataset, they would receive a high score in this metric.

Figure III helps understand this metric and was generated from a data structure containing all the hashtags of our tweets. By filtering by the most occurring hashtags per user, we generated our hashtag metric.

FIGURE III
WORD CLOUD OF MOST COMMONLY OCCURRING HASHTAGS



With these features decided upon, we worked on turning the data into numeric values.

Data Processing

To turn our raw data into numerical features, a large amount of data-processing and cleaning was required. We chose to use Pyspark dataframes to process our data. This Pyspark data type is a modern data type that wraps RDDs and makes scalable computing easier, working based off of a row-column data representation.

We tried standardizing followers, following, and tweets using standard deviation alone. We realized outliers skewed our standard deviation to be higher than our mean. We opted to take a logarithm of these unbounded values (we experimented with different bases depending on the scale of the parameter) before we measured standard deviation. This significantly lowered the standard deviation, however the outliers still caused a skew. Our standard deviation still exceeded half of our mean, which meant users with a score of 0 followers for example, were still only 1.5 standard deviations off of the mean which made normalization difficult. We opted to apply a filter to the outliers to prevent skewing. We deemed this valid, as users with an abnormally large number of followers, or tweets for instance, should be uncommon and detectable without intricate algorithms.

Experimentation Factors

With our numerical features normalized, we inputted them into our heuristic evaluation and k-means model.

Ⅰ. *Heuristic Evaluation*

Our first algorithm to determine if an account is likely a bot was our heuristic evaluation. A weighted linear combination of our features which outputs a score. By adding factors that would increase the likelihood and subtracting factors that would decrease the likelihood, all with their own weight, our heuristic evaluation creates a numeric score from 0-1 that represents the likelihood of an account being a bot. The equation is:

$$Hscore = Sigmoid([w_n * n - w_p * p] * [w_{tpd} * tpd] + [w_{fr} * fr + w_r * r + w_h * h - w_a * a] * [w_c * c]) \quad (1)$$

where $hscore$ refers to the heuristic score of an account and each variable refers to one of our features. Each feature has an associated weight, denoted by w, multiplied by that feature to adjust its impact in determining the final score. E.g. for negativity, n, its associated weight is w_n. The following table denotes what feature each variable represents:

TABLE I
HSCORE FEATURES AND WEIGHTS

Feature	Symbol	Feature Coefficient	Feature Weight
Negativity	*n*	+	1.5
Positivity	*p*	-	2.0
Tweets per Day	*tpd*	+	2.2
Following/Follower Ratio	*fr*	+	2.0
Tweet Ratio	*r*	+	1.0
Max Hashtag	*h*	+	2.5
Account Age	*a*	-	1.3
Tweet Count	*c*	+	1.6

With this equation, each account received a score that determined the likelihood of it being a bot. These scores and their evaluation are discussed in the *Results* section of this report.

Ⅱ. *K-Means Model*

Since our dataset is not labeled, an unsupervised machine learning algorithm is called for. We chose K-means clustering, as it is the most straightforward and has built in support with Pyspark. Pyspark distributes the data points across its worker nodes, and computes the centroid averages which are aggregated by the driver. The driver can update the centroids and ask the workers to recompute their calculations, making the iterative process incredibly easy to parallelize.

K-means clustering will allow us to detect patterns effortlessly and evaluate each cluster. We assume that our algorithm will be able to return distinguishable clusters that we can label ourselves. We will likely end up with several clusters unrelated to bots, a cluster that is mostly bots, and potentially a cluster or two that are unclear.

We assumed that certain parameters are more important in determining bot likelihood. We were able to manipulate the normalization ranges, for instance we could have “fraction of negative tweets” range from 0-150 and “average tweets per day” range from 0-70 so that the model is more sensitive to the former parameter.

Experiment Process

With our features created from the raw data, our experiment process then involved adjusting and tuning both our heuristic evaluation and basic k-means model.

Referencing weights used in earlier experiments such as Kantepe and Ganiz 2017[2] and Alarfaj et al. 2023[3] we were able to skip most of the guess and check work when assigning weights to our non-machine-learned algorithm. For instance it has been determined that ‘Max Hashtag’ is a strong indicator of bots[2]. Thus, this feature is most highly weighted in our algorithm. Further, it has been observed that bots have a much higher tendency to use negative sentiment compared to human users[3]. Reflecting

that, Hscore applies a high negative coefficient to a user's average positive sentiment. Thus, our heuristic evaluation scored users with higher usage of positive sentiment lower, and therefore as being less likely to be a bot.

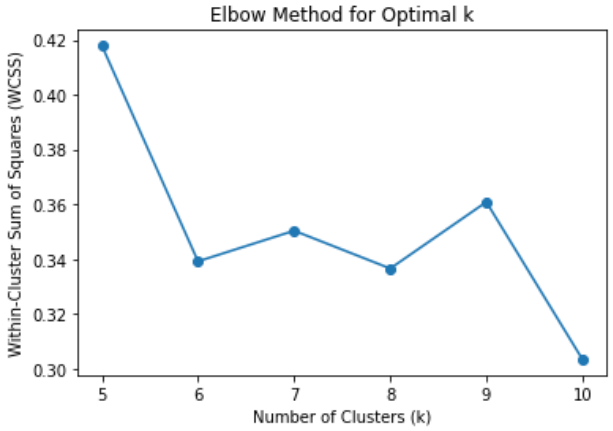
With weights set, the equation is split into four categories: tweet scores, tweet score modifiers, account scores, and account score modifiers. Tweet scores and account scores measure the suspiciousness of a user's tweets and of their account respectively. The score modifiers account for the magnitude of a user's presence. That is, a user who is slightly negative over 1000 tweets would be labeled more likely to be a bot than someone who is extremely negative in one tweet.

Next we proceeded with our K means clustering algorithm. After removing outliers, scaling, and normalizing our data, we found that our model was able to cluster our data effectively with our centroids being far apart.

The elbow method indicated that 6 was the optimal number of clusters. However, we decided to try each number to measure hypothetical results.

FIGURE IV

ELBOW METHOD GRAPH GENERATE FOR OPTIMAL K



We found that anything under 7 clusters gave us practically no insight. For instance, with 4 clusters we had a group which we deemed half automated half genuine, which would leave us doing most of the labeling. At 6 clusters this centroid moved further from the crowd, but to our judgment it still contained a notable amount of genuine accounts. 8 clusters appeared to be a sweet spot, however a few automated accounts seemed to slip into other centroids. We found that with 7 clusters, we were able to keep most of the targeted bot accounts inside our 'bot' cluster, and separate users whose activity is inconsistent with normal twitter users but who are clearly not bots. These suspicious, but not bot accounts might be political commentators, individuals who talk about very few topics, perhaps even negatively, but are still most likely human accounts.

Performance Metrics

Without labels, or any way to validate bots, measuring the performance of a human made algorithm becomes quite arbitrary. However, we would expect to see self-declared bots such as 'FuckPutinBot' and 'GasInfoBot' to have a large Hscore. While the weightings of the features used can be justified and the result produced can be manually checked albeit painstakingly, the best way to evaluate the Hscore result may be to compare the results found by the ML algorithm.

We were unable to use supervised learning metrics due to our unlabeled dataset. We decided against the elbow method as it only gives us insight into general clustering to determine the optimal K value, not insight into accuracy of cluster's qualitative results. Instead we found that two groups were similar but needed to be differentiated. One was an automated(bot) group, and the other was a 'Political Enthusiast' group. We evaluated the number of political enthusiasts that slipped into the bot group, and vice versa. We deemed our model accurate as long as the model correctly distinguished these types of accounts.

RESULTS

Key Findings

The main finding from this project was discovering the difficulty in applying machine learning to raw data. Our K-means clustering algorithm is only as good as our data processing. All of our parameters and metrics we deemed important are anecdotal, they are not a perfect representation of metrics related to bot account detection. However, based on our exploratory data analysis, the features we chose to determine if an account is a bot are more likely to indicate non-human behavior than other data points in the original dataset. The majority of the work in this project involved examining the raw data and transforming it into usable features, showing the difficulty and work required to use large, raw twitter data for complex machine learning analysis.

Diagnosing our ML model, as well as our heuristic evaluation, involved manually inspecting accounts that were deemed likely bots. Some accounts such as "FuckPutinBot" were clearly bots, and thus when evaluated by both our heuristic and ML model as likely a bot, we deemed our analysis to have at least some degree of accuracy.

We initially believed that the patterns on bot behavior would be harder to find. For instance, we believed some parameters may not be positively or negatively correlated with bot behavior, but rather a neutral value. We also believed that there may be instances where two parameters would be correlated, which our linear equation heuristic would not be able to account for. However, these

scenarios were not observed, which allowed both approaches to perform better than we expected.

For our heuristic, we looked at the score produced for accounts that were self-admitted bots such as “FuckPutinBot” and used it as a benchmark score to compare with other account scores. From a range of 0-1, where 1 is most likely a bot and 0 is least likely, ‘FuckPutinBot’ scored 1, showing how our heuristic reflected our features and predictions well. Other accounts with low scores such as ‘Kendalmint2’ who is likely not a bot scored 0.15, indicating our heuristic worked well for non-bots too. However, some accounts such as ‘danger_gamer75’ scored 0.99. Upon inspecting their account, it was clear that they were not a bot. Therefore, while some accounts received an accurate score, others did not. This was expected for our naive implementation.

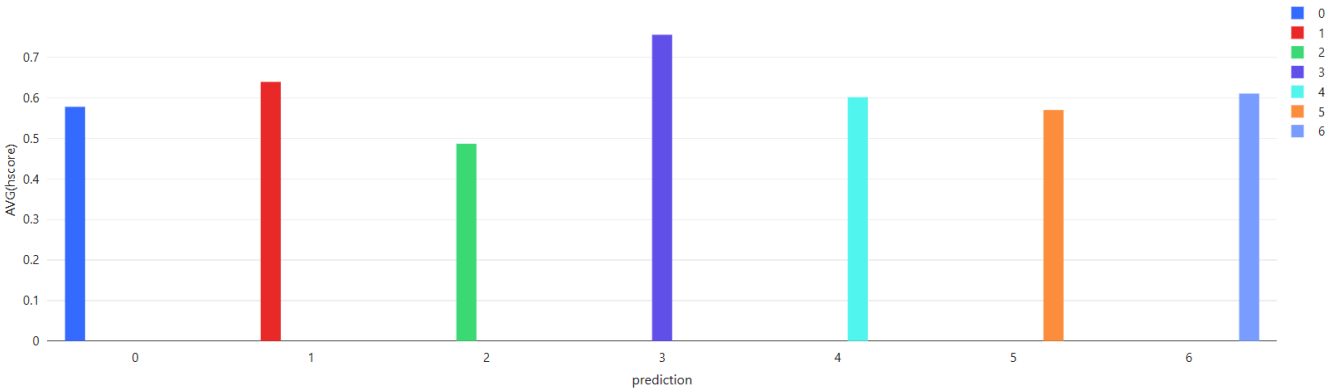
Our K-means model was able to cluster our usernames into several key groups. Our final model had seven clusters. As we predicted, one cluster was the most effective at classifying accounts as a bot, and one was most effective at classifying accounts as human. However, the other clusters were not as easy to classify. By looking at some of the accounts in each cluster, we determined that there are many accounts that were probably not bots, but still displayed behaviour similar to bots. These accounts would have large amounts of political tweets, with negative sentiment, low followers and following count, yet we felt that they were most likely not bots. For these accounts, our model struggled to place them in the same cluster as most likely human accounts.

We found that the main parameters in determining bot likelihood are hashtag repetition and number of total tweets, as classified by our bot cluster.

Comparing our heuristic evaluation and k-means model, we found that both were generally good at finding accounts that are actually bots. However, many accounts that are likely not bots, but had similar characteristics as bot accounts, were given a high bot score. We found that both our heuristic evaluation and our k-means model struggled with these types of accounts. However, the k-means model was better at differentiating these unclear accounts from definitively bot accounts, as well as from human accounts.

Figure V compares the heuristic evaluation to our k-means model by showing the average h-score of each cluster. From this graph we can see how certain clusters had varying scores, yet many had very similar scores, showing the difficulty in classifying certain accounts.

FIGURE V
AVERAGE H-SCORE OF EACH CLUSTER



Conclusions and Future Work

We analyzed several features that have been identified by other research as being powerful at determining bots, notably topic variation of tweets, and longest session[2]. Taking the cosine difference of a TF-IDF matrix belonging to an individual user we could measure the variation of a user’s corpus of tweets. While we were able to do this for single users at a time, we were unable to scale this feature to the size of our dataset. We were able to find a cheap analog for this feature by measuring the number of tweets in our dataset compared to the number of tweets a user had at their first entry in our data set compared to the last. That is, if a user had 50 tweets at their first entry and 60 tweets in their last entry and 5 tweets in our dataset we can tell that 50 percent of the user’s most recent tweets are related to one topic. Further, longest streak, a feature used by Kantepe and Ganiz 2017[2], would measure a user’s longest streak of tweets without a 4 hour break. This feature while unrealized by our team would provide a way to undoubtedly classify bots if the streak were longer than say 16 hours.

While timelines did not afford the opportunity, the team had wished to investigate other topics. For example, insights into what bots were talking about, classifying distinct families of bots, or identifying trending topics originating from bot accounts. Despite these questions

remaining unexplored, the final dataframes produced by the project may be a good starting point for further research.

We also did not examine accounts as a whole. We only used the tweets in our dataset, which were pulled in from the twitter API. Our heuristic and especially our K-means would have likely been impacted if our dataset contained tweets outside the Russo-Ukraine conflict. Given a larger scope we would likely need to look into more parameters for each approach, as well as adjust the K value in the clustering algorithm.

References

[1] D Stukal, et al. (2017) Detecting bots on Russian political Twitter. Big Data 5:4, 310–324, DOI: 10.1089/big.2017.0038.

[2] M. Kantepe and M. C. Ganiz, "Preprocessing framework for Twitter bot detection," 2017 International Conference on Computer Science and Engineering (UBMK), Antalya, Turkey, 2017, pp. 630-634, doi: 10.1109/UBMK.2017.8093483.

[3] F. K. Alarfaj, et al. "Twitter Bot Detection Using Diverse Content Features and Applying Machine Learning Algorithms". Sustainability 2023, 15, 6662. https://doi.org/10.3390/su15086662

[4] Botometer, formerly BotOrNot, "open source bot detection algorithm by OSoMe", Indiana University

[5] B. Smart, et al. "Malicious Activity in Online Social Networks; How Bots are Driving Discussion around the Russia/Ukraine war", 2022 on arXiv, Cornell University, https://arxiv.org/pdf/2208.07038.pdf