NUS National University of Singapore | Faculty of Science

DSA4263

Sense-making Case Analysis: Business and Commerce

Final Project

Twitter Bot Detection

By:

Amirtha D/O Anbalagan (A0226673Y)
Binali Manilka Pilapitiya De Alwis (A0225755Y)
Bryan Yeo (A0216419E)
Chang An Le Harry Jr (A0201825N)
Dongmen Runze (A0219951X)
Eldora Boo Puay Eng (A0221833M)

TABLE OF CONTENTS

1 Abstract

The proliferation of bots on social media platforms, particularly Twitter, has raised concerns regarding the authenticity and reliability of user interactions and content dissemination. This paper investigates the escalating trend of bot activities on Twitter, and proposes a robust methodology for their detection. Leveraging the comprehensive Cresci-2017 dataset, our study delves into both individual account behaviors and tweet characteristics to unveil bot presence.

In our research, we evaluated several machine learning models for user classification, ultimately selecting XGBoost, as the most effective after comparative analyses with Random Forest and Logistic Regression. For tweet analysis, Latent Dirichlet Allocation (LDA) emerged as the preferred choice following evaluations against BERT, XGBoost, and various NLP models.

Our findings underscore promising results, paving the way for a two-stage architecture for bot detection. Initially, we propose scrutinizing the tweets, followed by identifying bot users for bot-generated content. This sequential approach offers enhanced accuracy in bot detection. We believe our findings not only shed light on the escalating bot activities on Twitter but also offer practical insights for enhancing bot detection mechanisms, thereby fostering a more trustworthy online environment.

2 Introduction

Our dataset includes labeled genuine and bot user accounts and tweets, vital for Twitter bot detection research. The prevalence of bots on the platform severely affects user experience and information integrity. SparkToro and Followerwonk's analysis of 44,058 active Twitter accounts in May 2022 revealed a significant bot presence. Similarly, Similarweb reported bots generate 20%-29% of US Twitter content, raising concerns about authenticity and misinformation spread.

Regulatory scrutiny over social media platforms is increasing. In 2022, the Texas Attorney General investigated Twitter's bot accounts under the Texas Deceptive Trade Practices Act. This underscores the need for robust bot detection models to comply with legal standards and curb malicious activities.

Bot presence distorts engagement metrics, causing financial losses in advertising campaigns. Effective bot detection preserves advertising integrity by ensuring investments reach genuine users, maintaining marketing efficacy. Bots also influence political discourse, posing threats recognized by governments and international agencies. Social media platforms play a crucial role in democratic processes, necessitating the authenticity of interactions to safeguard national security and democratic integrity. User trust is paramount for platforms like Twitter. Recent events, such as Elon Musk's involvement, highlight the challenge of bot proliferation, risking user departure to competitors like Threads if not addressed effectively.

Our research aims to refine bot detection models to reduce fraud and enhance platform reliability and integrity. Improving these models supports a safer digital environment, fostering trust among users and stakeholders.

3 Dataset

Our primary datasets were obtained from the Bot Depositary, a central hub for annotated Twitter bot datasets. We chose this dataset primarily for its accessibility—it was freely available for download and did not require permission from the dataset owners, unlike others we considered. Our criteria for dataset selection included the number of data points, level of detail, pre-labeling, and labeling process. The chosen dataset best met these criteria, making it ideal for our analysis.

The dataset was categorized into genuine, social spam bot, or traditional spam bot groups by CrowdFlower contributors. Although this process may introduce some human bias, it was the most accurately labeled dataset of its size that we found. Many other labeled datasets were synthetic, posing limitations such as inability to use usernames and user IDs for data gathering, and challenges in building models based on nonsensical synthetic tweets. Thus, this dataset was deemed the most suitable for our purposes.

Volunteers reviewed information for each Twitter account, verifying account details and classifying them as "Spambot" or "Genuine". They were provided with a list of signs to identify potential spambot activity, including repetitive or automated behavior, solicitation for followers or retweets, and promotion of topics related to sex, pornography, or dubious job offers. The dataset comprised files on genuine accounts, social spam bots, and traditional spam bots, each further divided into datasets on user accounts and tweets. For our project, we merged and re-categorized these datasets to create comprehensive, labeled datasets for Tweets and Users separately.

3.1 Users Dataset

The initial users dataset had 42 feature columns. After merging the datasets of genuine, traditional spam bots, and social spam bots, we added a "Type" column to differentiate between bots and genuine accounts, resulting in a combined dataset of 11,017 entries. This dataset comprised 7,543 bot accounts and 3,474 genuine accounts, representing a split of 68.5% bots and 31.5% genuine accounts. While this imbalance of 68.5% bots and 31.5% genuine accounts deviates from the typical real-world split of around 20% bots and 80% real users, it presents valuable opportunities for bot detection model enhancement.

Several significant features in the Users dataset include:

Activity Metrics

- 'statuses_count': Number of tweets and retweets by the user.
- 'followers_count': Number of users following the account.
- 'friends_count': Number of users the account is following.

User Profile Settings

- 'lang': Language setting of the user profile.

- 'time_zone': Time zone setting of the user profile.
- 'location': Geographical location setting of the user profile.
- 'default_profile': Indicates if the profile uses the platform's default settings.
- 'geo_enabled': Indicates if the location feature is enabled on the profile.

For detailed information on the features, please refer to the Users_DataDictionary.txt.

3.2 Tweets Dataset

Initially, the tweets dataset comprised 25 feature columns. After merging the datasets of genuine, traditional spam bots, and social spam bots, we added a 26th column named "IsBot" to distinguish between tweets from bots and genuine users. We extracted a combined dataset of 60,000 entries with and even split between Bots and Genuine accounts.

Several significant features in the Tweets dataset include:

Core Tweet Information

- id: Serves as a unique identifier for each tweet.
- text: Contains the textual content of the tweet.

Engagement Metrics

- reply_count: Measures the number of replies a tweet has received.
- favorite_count: Counts the number of times a tweet has been favorited.

For detailed information on the features, please refer to the Tweets_DataDictionary.txt.

3.3 Rationale

We're analyzing Users and Tweets datasets separately for a nuanced understanding of user profiles and online behaviors. This method effectively pinpoints different bot types: Spam Bots and Malicious Link Sharing via Tweets, and Fake Follower Bots and Botnets via Users. We've avoided merging the datasets to maintain computational efficiency and ensure scalability. Separation allows for manageable processing and flexible expansion as data volumes change, recognizing that tweet activity and user growth may not always align. Additionally, unique dataset features cater to specific analysis requirements, justifying the decision to keep them distinct.

3.4 Scraped Dataset

We scraped recent tweets from active accounts using Python, Selenium, and BeautifulSoup to update our dataset beyond the older cresci-2017 data. This new dataset includes the latest tweets and associated metadata such as reply counts, likes, and user details.

3.5 Synthetic Dataset

We used the Faker library to create synthetic data but encountered issues with the incoherence of generated tweets and the authenticity of user IDs, limiting analysis and verification. Therefore, we focused on authentic datasets for reliable analysis.

4 Exploratory Data Analysis

4.1 Users Dataset

We examined four distinct hypotheses pertaining to various features in our Users dataset. Among them, the most notable findings were:

Hypothesis 1: Bots have a disproportionate ratio of followers to friends, indicating non-reciprocal relationships.

This hypothesis was supported by our analysis, although we had to adjust our statistical approach due to the distribution of the data. Initially, we considered using a t-test, which assumes normality in the data distribution for both groups being compared. However, significant deviations from normality in our data made the t-test unsuitable. Instead, we employed the Mann-Whitney U test, a non-parametric test that does not require the normality assumption and is effective for comparing differences in medians between two independent groups. In Fig 1, our findings revealed that the median followers-to-friends ratio for 'Bots' is lower than for 'Not Bots,' suggesting that bots tend to follow more accounts relative to how many follow them back. Additionally, the data for 'Not Bots' displayed a larger interquartile range (IQR), indicating a greater variance in how genuine accounts engage with others. This group also showed a skew towards higher followers-to-friends ratios, which is characteristic of popular genuine accounts or influencers who typically gather many followers without reciprocally following back.

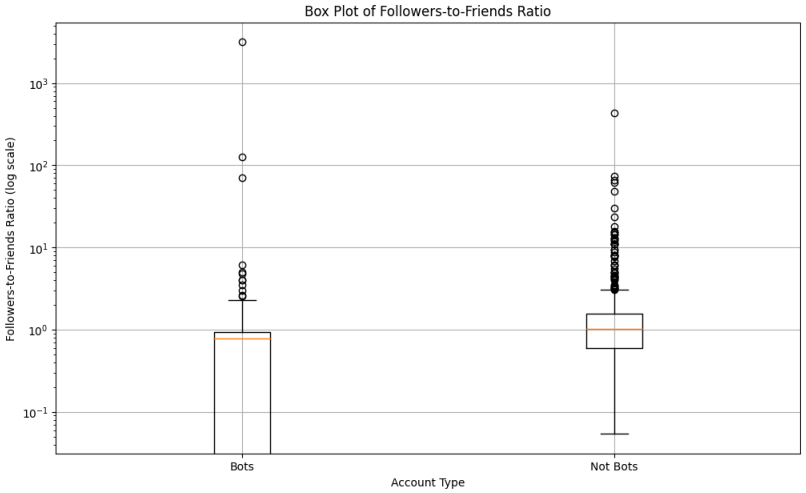

Fig 1. Box plot of follower-to-friends ratio for Bots vs Genuine accounts

Hypothesis 2: Bots are more likely to have a default or generic profile setup.

Our findings partially supported this hypothesis. Although we didn't find a significant use of default images, bots tended to reuse the same images more frequently than genuine users. This analysis was based on a smaller dataset of 1,000 records because we needed to convert image links to actual images before we could analyze them. Ultimately, the results weren't strong enough to include in our final models, mainly because the computational and storage costs were too high. In future iterations of our work, we plan to delve deeper into this

aspect, potentially using more advanced image analysis techniques to improve efficiency and effectiveness.

Hypothesis 3: Bots engage less with content in terms of likes and retweets but may post content that is heavily retweeted within bot networks.

The analysis confirmed this behavior, showing clear distinctions in engagement levels. Boxplots indicated that genuine accounts are more active across key metrics: 'listed_count', 'statuses_count', and 'favourites_count'. To further test this, we utilized a simple linear regression model based on these engagement metrics, which achieved an impressive accuracy of 89.65% as seen in Fig 2. The classification report from this model highlighted its effectiveness, with high precision and recall rates for identifying both bot and genuine accounts. Specifically, the model had a precision of 0.88 for bots and 0.96 for genuine accounts, with recall rates of 0.99 for bots and 0.69 for genuine accounts, respectively. The confusion matrix in Fig 3 also confirmed the model's strong performance, displaying minimal false positives and false negatives. Based on these results, we decided to incorporate these engagement metrics into our subsequent models to enhance their predictive accuracy and reliability.

```
Accuracy: 0.896551724137931

Classification Report:
              precision    recall  f1-score   support

         Bot       0.88      0.99      0.93      1528
     Genuine       0.96      0.69      0.80       676

    accuracy                           0.90      2204
   macro avg       0.92      0.84      0.87      2204
weighted avg       0.90      0.90      0.89      2204
```

Fig 2. Results for a simple linear regression model using activity metrics

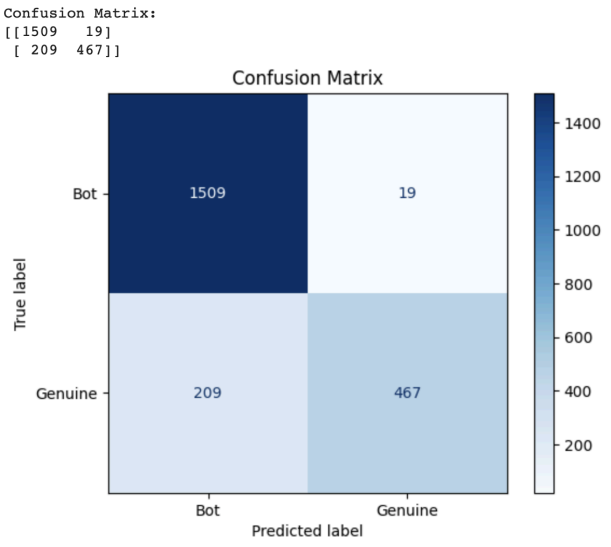

Fig 3. Confusion matrix for a simple linear regression model using activity metrics

Hypothesis 4: Bots are less likely to be geographically consistent or provide accurate location data.

We observed that a higher percentage of genuine users have location services enabled as seen in Fig 4. Despite this, both bots and genuine accounts frequently used descriptions of random or fictitious places. Notably, bots reused the same location descriptions more often compared to genuine users. For example, bots had 887 unique locations listed, while genuine users had 1796 unique locations, despite the overall dataset containing significantly more bot entries. This significant discrepancy highlights that bots are less likely to offer diverse or accurate geographical information. The nearly equal number of location data entries between bots and genuine accounts further underscores this point, indicating a strategic but limited use of location data by bots. Given these findings, we decided to further explore the nature of location data in our models, as this could provide deeper insights into bot behavior and enhance our detection capabilities.

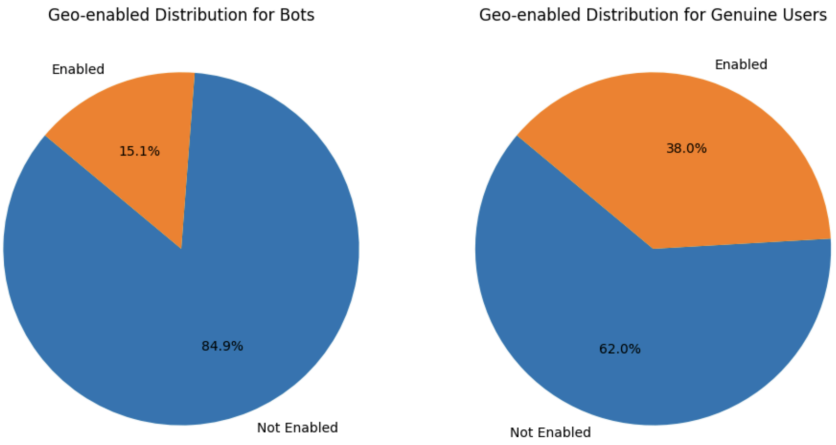

Fig 4. Proportion of accounts that have enabled geo location for Bots vs. Genuine accounts

4.2 Tweets Dataset

We examined two distinct hypotheses pertaining to various features in our Users dataset. Among them, the most notable findings were:

Hypothesis 1: Bots generate tweets with more hashtags and links compared to genuine accounts.

This hypothesis was found to be true, albeit with less significance than initially anticipated. On average, bot tweets contained approximately 0.32 hashtags and 0.42 URLs, while genuine tweets had approximately 0.19 hashtags and 0.25 URLs. Although bot tweets do exhibit a higher average count of hashtags and URLs, the difference is not as pronounced as expected. However, further analysis revealed that 19.74% of bot tweets contained hashtags compared to 14.43% of genuine tweets, and 41.62% of bot tweets contained URLs compared to 24.23% of genuine tweets as shown in Fig 5. Notably, bot tweets displayed almost double the percentage of URLs, a significant observation considering that bots often utilize fraudulent links to perpetrate fraud. Given these findings, we opted to incorporate the presence of hashtags and URLs as features in our models, recognizing their potential significance in distinguishing between bot and genuine tweets.

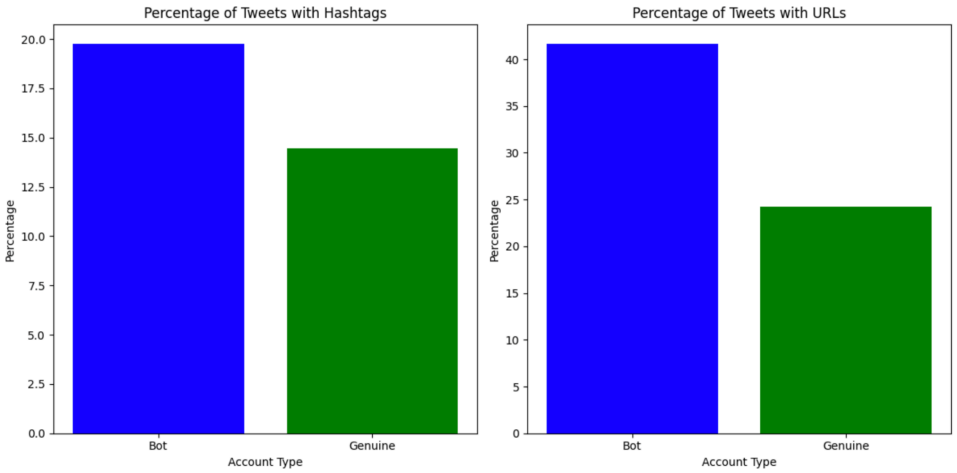

Fig 5. Percentage of Tweets with Hashtags & URLs for Bots vs Genuine accounts

Hypothesis 2: Bots may exhibit different temporal dynamics in terms of when they are active compared to genuine users. They may also post at certain times or post a lot of tweets at a stretch.

The data supported Hypothesis 2. Bot tweets show regular 15-minute intervals of activity in Fig 6, indicative of automated scheduling, unlike the sporadic timing seen in human tweeting patterns in Fig 7. Genuine tweets peak outside typical work hours, reflecting the natural rhythm of human activity. These trends confirm that bots follow a systematic schedule, whereas genuine users' Twitter activity aligns with their personal schedules.

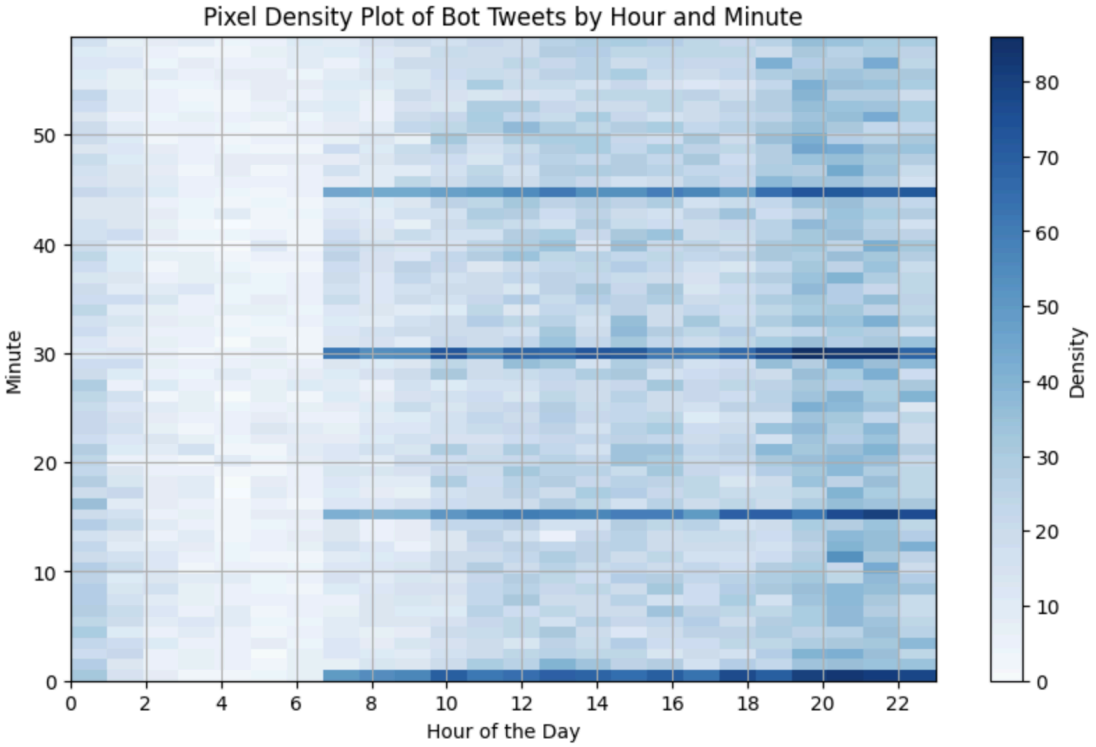

Fig 6. Pixel density plot of Bot Tweets by Hour and Minute of the day

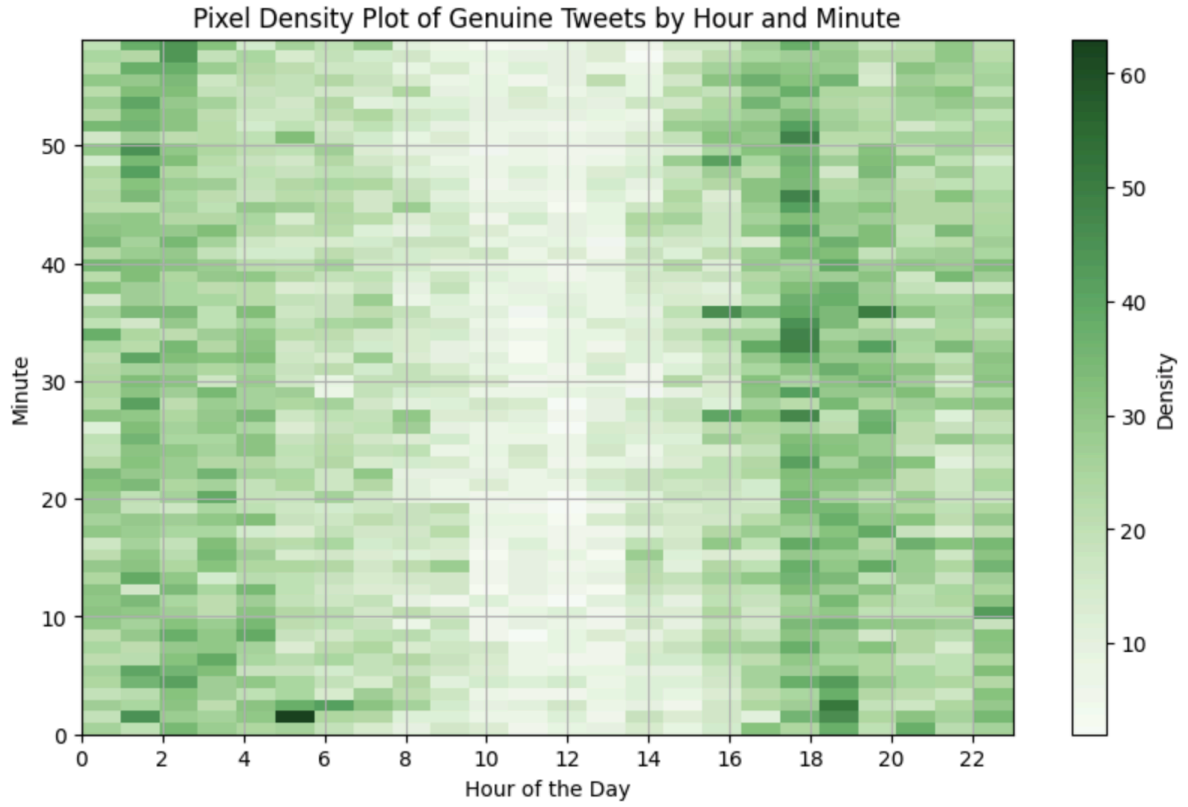

Fig 7. Pixel density plot of Genuine Tweets by Hour and Minute of the day

5 Methods used

5.1 Users Dataset

Feature Identification and Initial Data Pre-processing

Removing entirely null elements and columns with over 50% missing rows, such as 'following,' 'contributors_enabled,' 'notifications,' 'follow_request_sent,' 'url,' 'default_profile_image,' 'profile_background_tile,' 'is_translator,' 'protected,' and 'verified,' ensured dataset integrity and relevance.

We identified several key features to predict the 'Type'—our target variable for distinguishing bots within the user dataset. These features include 'followers_to_friends_ratio', 'listed_count', 'statuses_count', 'favourites_count', 'geo_enabled', and 'location'. To effectively process these features, we initially applied one-hot encoding to the "location" feature, transforming this categorical variable into a format suitable for our machine learning algorithms. This process converts each unique location into its own binary column, ensuring accurate category interpretation by the model without assuming any ordinal relationship between locations.

Pipeline Integration and Data Transformation

To efficiently prepare our data for modeling, we utilized the Pipeline module from the scikit-learn library. The pipeline was crucial for maintaining the integrity and efficiency of our data processing. By structuring our data transformation and preparation steps into a pipeline, we ensured that each stage is executed sequentially and isolated from one another, which is key in preventing data leakage during training and validation phases. The pipeline includes stages for imputation, where missing values are replaced with the mean of each column, and standardization, which normalizes the data by removing the mean and scaling to unit variance.

Feature Engineering

Our feature engineering efforts were further detailed by creating the 'followers_to_friends_ratio' by dividing 'followers_count' by 'friends_count'. This feature helps in distinguishing unusual social patterns often exhibited by bots. Additional features like 'account_age_days' and 'has_description' were specifically engineered to enhance our models' ability to identify bots based on account maturity and the presence of a user description. Additional transformations included converting the 'created_at' timestamp into 'account_age_days' to capture the account's age, which is instrumental in identifying potentially fraudulent activity associated with newer accounts. The 'description' field was also transformed into a binary indicator of whether an account description exists.

Validation Approach

For model validation, we employed a standard train-test split, allocating 80% of our data for training and 20% for testing, with the split conducted using a random seed for reproducibility.

Parameter Tuning and Dimensionality Reduction

The significant increase in dataset dimensionality post-one-hot encoding, expanding it to a total of 2,241 columns, necessitated careful feature selection and parameter tuning. Dimensionality reduction using PCA was employed to manage this high dimensionality while preserving 95% of the variance in the original features. This approach not only helped in managing the complexity but also in enhancing the models' performance by focusing on the most informative features.

Model Selection

In addressing the challenge of categorizing bots, we selected logistic regression, random forest, and XGBoost as our primary models. Logistic regression was chosen as the baseline model due to its simplicity and efficiency in handling binary classification problems. It provides a clear probabilistic interpretation of model predictions, making it easy to implement and fast to train, which is particularly beneficial for preliminary analysis and rapid iterations. However, logistic regression can struggle with non-linear relationships and complex interactions between features.

To address these limitations, we incorporated Random Forest, a robust ensemble technique that builds multiple decision trees and aggregates their predictions. This model excels in managing overfitting, a common pitfall of logistic regression when dealing with high-dimensional data, and captures nonlinear interactions more effectively. Random forest also provides feature importance scores, aiding in interpretability regarding which predictors most influence the classification of bots.

Finally, XGBoost was added to our modeling suite for its advanced capacity to handle large datasets with a gradient boosting framework that systematically refines models through successive training rounds, focusing on correcting previous prediction errors. XGBoost is particularly advantageous for its execution speed and model performance, often outperforming many other algorithms on structured data. It is especially useful when dealing with imbalanced datasets, a common scenario in bot detection and our project as well, by leveraging its built-in capabilities to weigh classes differently.

5.2 Tweets Dataset

Feature Identification and Initial Data Pre-processing

We began by curating a dataset comprising 60,000 rows, evenly divided between bot-generated tweets and those from genuine users, from the original pool of over 1.6 million

tweets. This selection aimed to strike a balance between computational efficiency, storage capacity, and dataset size.

During the curation process, several features such as 'truncated', 'geo', 'contributors', 'favorited', and 'retweeted' lacked sufficient data and were excluded from the dataset. Additionally, 'possibly_sensitive' and 'place' contained minimal non-null values and were removed. However, to ensure dataset completeness and coherence, all remaining columns, except 'text' and 'in_reply_to_screen_name', had their values propagated.

In this initial data pre-processing phase, our focus was on narrowing down the dimensions to essential components, including tweets ('text'), all float-type attributes, and the removal of columns with missing values. Subsequently, we conducted straightforward processing to identify float-type attributes with the highest single prediction accuracy rate.

Training & Test Data

The data was partitioned into training and testing subsets, comprising 80% and 20% of the data respectively. This ratio ensures ample data for training while retaining a significant portion for testing to assess the model's generalization capabilities.

Model Selection

Baseline Model

As a benchmark, XGBoost was selected due to its fast performance and ease of interpretation. Initially, our analysis encompassed a dataset abundant in features, where we implemented a dual approach: evaluating quantitative metrics such as retweet and favorite counts, alongside textual analysis through Natural Language Processing (NLP). Our assessment covered a range of classification models including KNN, Logistic Regression, Random Forest, and AdaBoost, all known for their operational efficiency and user-friendliness, allowing for rapid iterations and comprehensive evaluations.

XGBoost distinguished itself in accuracy, demonstrating a range between 70% and 74%. Using the feature importance tool within XGBoost, we determined 'retweet_count' to be the most predictive element for distinguishing bots, followed by 'favorite_count,' 'num_hashtags,' 'num_mentions,' and 'num_urls.' This discovery, in conjunction with XGBoost's computational speed and clarity, affirmed its status as our benchmark model. We refined the model further by adjusting its learning rate and tree depth to enhance both precision and the impact of salient features.

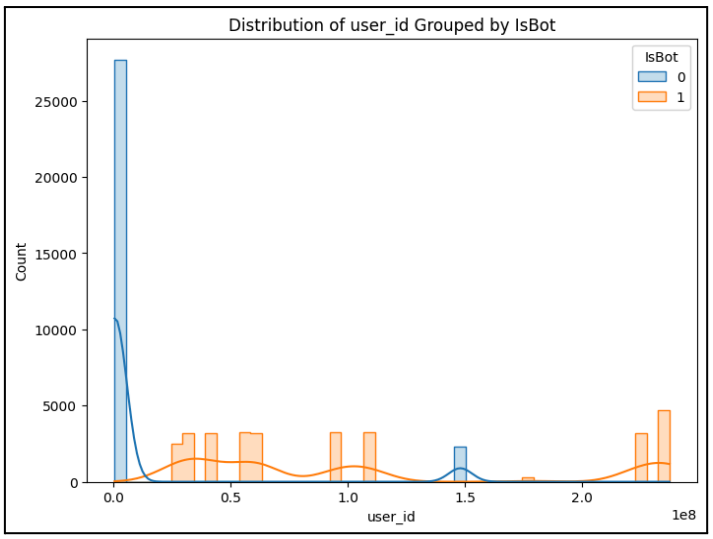

Fig 8. Distribution of user_id for Bots vs. Genuine Accounts

Referencing Figure 8, the analysis of feature importance highlighted unique 'user_id' distributions differentiating bots from genuine users, indicating batch creation of bots within certain periods. However, 'user_id' was found to be impractical for ongoing application due to its non-generalizable nature, potential obsolescence over time as bot strategies evolve, and its specificity to the dataset's timeframe. This led to a strategic shift towards more reliable and behavior-oriented metrics such as 'retweet_count,' 'favorite_count,' 'num_hashtags,' 'num_urls,' and 'num_mentions.' These features, indicative of user engagement, are in line with our hypothesis and provide a more stable foundation for bot detection.

BERT Model

BERT was selected as a primary challenger due to its cutting-edge performance in numerous natural language processing tasks. BERT's architecture is specifically designed to understand the contextual nuances of language, making it exceedingly effective for tasks that require a deep understanding of textual data. For this analysis, BERT was employed not only to process and analyze raw textual data but also to integrate and analyze additional numeric features, such as 'Retweet Count', 'Mention Count', and 'Follower Count'. These features were standardized and combined with a binary 'Verified' status to enrich the model's input, providing a broader context that extends beyond simple text analysis. The inclusion of these user engagement metrics was hypothesized to enhance the model's ability to distinguish between bots and human users effectively.

LDA (Latent Dirichlet Allocation)

Latent Dirichlet Allocation (LDA) was utilized as another challenger model, focusing on unsupervised topic modeling to extract thematic patterns from the text data. LDA identifies latent topics by grouping commonly co-occurring words, which can provide insightful features for subsequent classification models. In this project, LDA was instrumental in generating features that represent the underlying topics within the tweets, which were

presumed to vary significantly between bots and humans. These topic distributions were then used as input features for a RandomForest classifier, exploring whether thematic content could serve as a reliable indicator of automated behavior.

NLP (Natural Language Processing) Model

In the development of our natural language processing (NLP) model, we faced significant challenges due to the complexity and multilingual nature of the dataset, which included tweets in languages like Korean, Chinese, and French. The need to identify and remove stop words for each language greatly complicated the text cleaning and model optimization process. Ultimately, the diversity of languages led to difficulties in running the model effectively. For future research, isolating English-language texts may allow for focused model development, but this would limit the model's applicability to English and necessitate separate models for each language to ensure accuracy. Thus, while an attempt was made, the NLP model did not produce meaningful results due to these language barriers and the dataset's limitations.

6 Results & Discussion

5.1 Results from Users Dataset

In our evaluation of models for binary classification on our user dataset, we analyzed the performance based on several key metrics. The results are summarized below:

Model	Accuracy	Precision	Recall	F1-Score	AUC-ROC
Logistic Regression	91.08%	92.37%	94.13%	93.24%	95.30%
Random Forest	92.62%	**96.71%**	91.84%	94.21%	98.04%
XGBoost	**93.77%**	94.59%	**95.96%**	**95.27%**	**98.75%**

XGBoost excels across most performance metrics, particularly in recall, F1-score, and AUC-ROC, indicating its strong ability to correctly identify bots while minimizing false negatives. The model's high AUC-ROC also underscores its exceptional capability to differentiate between classes. The effectiveness of XGBoost can be attributed to its method of sequentially building trees, where each new tree corrects errors from the previous ones, combined with regularization to prevent overfitting.

Random Forest demonstrates the highest precision, signaling strong accuracy when it labels users as bots. This characteristic minimizes false positives, crucial in avoiding disruptive false alarms. However, its recall is lower than XGBoost, likely due to its methodology of averaging results across many decision trees without adjusting for errors in individual trees progressively, which can cause some bots to be missed.

Logistic Regression, being a linear model, struggles with the complex nonlinear patterns within the dataset, reflected in its lower scores across precision, accuracy, F1-score, and AUC-ROC. Its decent recall indicates some sensitivity, but its overall effectiveness is limited by the inability to capture intricate interactions between features.

XGBoost is the preferred model for bot detection in our Users dataset, balancing high recall and precision for accurate identification. Its superior AUC-ROC demonstrates its strong class differentiation, essential for a reliable bot detection system. This precision is crucial in fraud detection, where it's important to catch fraudulent bots (reducing false negatives) without mislabeling genuine users (limiting false positives), thereby maintaining system integrity and user trust. XGBoost's effective handling of imbalanced data, due to its advanced features, makes it particularly apt for complex fraud detection tasks.

5.2 Results from Tweets Dataset

In our evaluation of models for binary classification on our Tweets dataset, we analyzed the performance based on several key metrics. The results are summarized below:

Model	Accuracy	Precision	Recall	F1-Score	AUC-ROC
BERT	50.37%	50.24%	50.16%	50.25%	53.23%
Latent Dirichlet Allocation (LDA)	**89.30%**	**89.37%**	**89.33%**	**89.29%**	**95.12%**
XGBoost	73.01%	68.80%	82.35%	75.03%	78.62%

BERT is underwhelming, with all metrics closely aligned with the base chance level, suggesting it's not well-tuned to the task of bot detection in this instance. An accuracy and F1-score just above 50% suggest that the model is struggling to capitalize on its architectural complexity in this particular task. An accuracy and F1-score marginally over 50% point to a struggle in leveraging its advanced architecture, and an AUC-ROC score of 53.23% indicates poor discrimination between classes. This may signal the need for more tailored preprocessing, feature selection, or hyperparameter optimization to exploit BERT's capabilities.

Conversely, **Latent Dirichlet Allocation (LDA)** shows excellent performance, with all principal metrics approaching 90%. The high AUC-ROC score of 95.12% denotes a strong capability to differentiate between bots and humans, likely because LDA effectively captures thematic patterns in the data, which seem to be significant markers for bot detection. The high scores across the board for LDA imply that the topics it discerns are potent predictors of bot activity.

XGBoost demonstrates moderate effectiveness, with an accuracy of 73.01%, precision at 68.80%, and an impressive recall of 82.35%, which suggests it is fairly good at identifying most bots but at the expense of misclassifying some legitimate users. The F1-score of 75.03% and AUC-ROC at 78.62% are reasonable but not outstanding. XGBoost's average performance could stem from overfitting, suboptimal parameter tuning, or noisy data. It may benefit from a larger and cleaner dataset, more refined feature engineering, and rigorous cross-validation to improve its performance.

7 Conclusion

7.1 Summary of Users Dataset Results

The evaluation of models on the Users dataset identified XGBoost as the most effective tool for bot detection. Its success is likely due to its sophisticated approach to model building, which allows it to capture complex patterns and relationships in the data that simpler models might miss.

The insights gathered suggest that in the realm of bot detection, models that can handle complex and non-linear relationships in data tend to perform better. For practical application, this means employing models that not only detect obvious bot-like activity but can also uncover subtler, more sophisticated patterns of behavior that may not be immediately apparent. This nuanced detection is crucial, as bot operators continually evolve their strategies to mimic genuine user behavior.

7.2 Summary of Tweets Dataset Results

For bot detection within the Tweets dataset, LDA emerged as the most adept model. LDA's strength lies in its capacity to distill complex textual features into a smaller, more interpretable set of topics, which simplifies the classification process and scales effectively with larger datasets. This attribute was instrumental in its superior performance, demonstrating the model’s suitability for extensive text mining applications.

However, despite LDA's impressive results, it is important to recognize its limitations, such as intensive computational demands during inference processes like Gibbs sampling or variational inference, particularly as the dataset expands. Moreover, LDA's simplification of text data into a 'bag of words' overlooks the sequence of words, potentially omitting vital contextual information. This limitation becomes apparent when considering the subtleties of content intention, as in the case of tweet fraud detection. For example, LDA might not discern the difference in intent between two tweets containing similar words but with entirely different meanings, which could lead to misclassification:

- "Barcelona just won the world cup! I have won a lot of money betting on them!"
- "Win a lot of money by betting on Barcelona in this World Cup! <link>"

Although both tweets might be flagged as potentially fraudulent by LDA due to similar content, their contexts are markedly different, which could lead to incorrect classifications. Therefore, while LDA performs well with the given dataset, its effectiveness is heavily dependent on the nature of the data and requires careful tuning of parameters to optimize results. On the other hand, BERT and XGBoost remind us that there's no one-size-fits-all solution in model selection, and achieving high performance in bot detection relies on carefully tailoring the model to the data's unique properties.

7.3 Implementation within Twitter

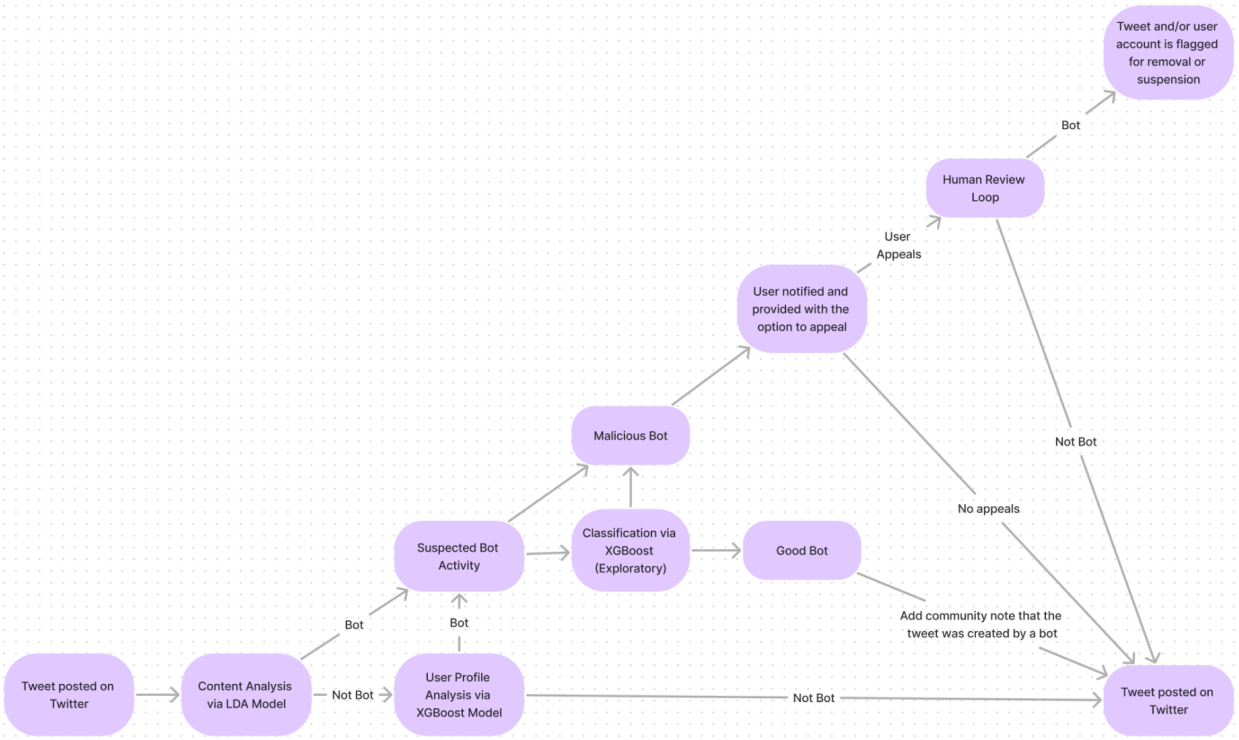

Fig 9. Twitter Architecture Diagram

The illustrated data pipeline delineates the sequential process for detecting and handling potential bot activities on Twitter (Fig 9). When a tweet is posted, it first undergoes content analysis by the LDA model to identify any textual patterns indicative of bot-generated content.

Tweets not immediately flagged by the LDA pass through to the next stage, where the XGBoost model analyzes the user profile for atypical behavior patterns that may suggest bot activity. If the content is flagged as suspect, it is then classified as either a 'Malicious Bot' or 'Good Bot' based on predefined criteria.

Following the analysis stages, tweets or users identified as 'Malicious Bots' trigger a notification to the user and an entry into the human review loop. This process allows for a more nuanced evaluation and potential override of the automated system's decisions. In the event of a user appeal, additional evidence is considered, which can either confirm the initial flagging or lead to a retraction.

Throughout the process, tweets and/or user accounts that are flagged as 'Malicious Bots' are earmarked for potential sanctions. In contrast, those determined to be 'Good Bots' or not bots proceed without action. All interactions, analysis outcomes, and appeal results are potentially tracked and displayed on an internal dashboard for continuous monitoring and refinement of the bot detection system. Data from this process can be utilized to further train and improve the LDA and XGBoost models, creating a feedback loop that enhances the system's accuracy and adaptability over time.

7.4 Applications to Real-World Fraud Detection

The models identified through our analysis are instrumental in identifying various fraud patterns. For instance, spam bots can be detected by their high posting frequency, repetitive content, and the presence of phishing links, which are significant markers for such fraudulent activities. Additionally, fake follower schemes can be identified by models like XGBoost, which can spot anomalies such as abnormal follower-to-following ratios, scant personal tweets, and coordinated following activities across multiple accounts. Influence campaigns, where bots manipulate public opinion or amplify specific topics, can also be flagged by detecting synchronized activities, unusually high tweet rates during specific events, or excessive retweets of particular accounts.

Enhancing these detection capabilities involves integrating additional tools. Anomaly detection systems can pinpoint outliers in tweet frequencies or account interactions that often indicate fraud. Real-time monitoring and alert systems can provide immediate notifications about suspicious activities, allowing for prompt action. Moreover, using account reports can help scrutinize accounts with frequent reports, aiding moderators in filtering out obvious bots and maintaining platform integrity. While LDA offers substantial benefits in thematic analysis for content-based bot detection, its effectiveness can be further improved with continuous adjustments to adapt to the dynamic nature of bot tactics and language use.

Word Count: 4878

8 Appendix

8.1 Data Dictionary

Users Dataset: Users_DataDictionary.txt.

Tweets Dataset: Tweets_DataDictionary.txt.

8.2 Bibliography

1. Cresci, S., Di Pietro, R., Petrocchi, M., Spognardi, A., & Tesconi, M. (2017b). The Paradigm-Shift of Social Spambots: Evidence, theories, and Tools for the arms race. Technical University of Denmark, DTU Orbit (Technical University of Denmark,DTU),963–972. https://orbit.dtu.dk/en/publications/0b309b1c-caea-4161-a775-853add742e60

2. Davis, C. A., Varol, O., Ferrara, E., Flammini, A., & Menczer, F. (2016). BotOrNot. ResearchGate. https://doi.org/10.1145/2872518.2889302

3. Editor Preview of Task — Tasks by CrowdFlower. (n.d.). https://ci.iit.cnr.it/fake/fake/crowdflower/instructions/

4. Martini, F., Samula, P., Keller, T., & Klinger, U. (2021). Bot, or not? Comparing three methods for detecting social bots in five political discourses. Big Data & Society, 8(2), 205395172110335. https://doi.org/10.1177/20539517211033566

5. Taylor, J. (2023, September 9). Bots on X worse than ever according to analysis of 1m tweets during the first Republican primary debate. The Guardian. https://www.theguardian.com/technology/2023/sep/09/x-twitter-bots-republican-primary-debate-tweets-increase

6. Thavasimani, K., & Srinath, N. K. (2021). A custom classifier to detect spambots on CRESCI-2017 dataset. In Lecture notes in electrical engineering (pp. 181–191). https://doi.org/10.1007/978-981-16-1338-8_16

7. Varanasi, L. (2022, September 9). Twitter bots appear to be be in line with the company's estimate of below 5% — but you wouldn't know it from how much they tweet, researchers say. Business Insider. https://www.businessinsider.com/twitter-bots-comprise-less-than-5-but-tweet-more-2022-9#:~:text=Similarweb%20reported%20that%2020%25%2D,some%20can%20pose%20serious%20threats.