Report: Identifying Human and Bot Accounts on Twitter Using Machine Learning

Group 7: Shivank Agrawal, Ruoheng Du, Yiwen Ge, Yandong Xiang

Introduction

The rise of automated accounts (bots) on social media platforms, such as Twitter, has raised concerns over the dissemination of misinformation and malicious content. Bots, which are automated accounts spreading content without human oversight, can manipulate discussions, propagate fake news, and skew public opinion. To address this issue, our project aims to classify Twitter accounts into human-operated and bot-operated accounts using machine learning techniques. This binary classification task utilizes a Kaggle dataset containing information on 37,438 Twitter accounts described by 19 features, including metadata (e.g., account age, followers count), behavioral insights (e.g., tweets per day, favorites count), and descriptive attributes (e.g., profile description, verified status). The primary objectives of this project are:

1. To preprocess and analyze the dataset to gain insights into its characteristics.
2. To experiment with and optimize a range of machine learning models for the classification task.
3. To evaluate model performance using metrics such as AUC, F1-score, precision, and recall to select the most effective approach.

Exploratory Data Analysis

The initial data cleaning process involved removing non-informative or redundant features. Features such as “created_at”, “profile_background_image_url”, and “profile_image_url” were dropped because they provide repetitive information as “account_age_days”, “default_profile”, “default_profile_image” provide. Next, missing values were addressed systematically. The “location” feature had only 4 missing rows, which were removed. Missing values in “lang” (7,957 rows) were replaced with “Unknown”, as accounts without language data showed distinct patterns. For the “description” feature, a new boolean feature, “default_description”, was created to indicate the presence of a description, leveraging its strong predictive signal.

Exploration of numerical features revealed significant skewness in variables: “followers_count”, “favourites_count”, “friends_count”, “statuses_count”, and “average_tweets_per_day”. Log transformations were applied to these features to normalize their distributions. Correlation analysis identified multicollinearity between “statuses_count” and “average_tweets_per_day”, leading to the removal of “statuses_count”. Boolean features were converted to numeric values (0/1), and chi-square tests confirmed statistically significant relationships with the target variable (“account_type”). For categorical features, “location” and “lang”, rare categories in “location”, which appeared less than 5 times were grouped into “Other” to reduce noise and prevent overfitting.

Finally, numerical features were standardized using “StandardScaler”, and target encoding was applied to “lang” and “location”, ensuring consistency across all splits. These preprocessing steps provided a clean, balanced, and well-prepared dataset for modeling.

Methodology

The dataset was split into training (60%), validation (20%), and test (20%) sets using stratified sampling to maintain class balance. Multiple machine learning models were implemented to compare their effectiveness. Each model was tuned using cross-validation on the training set and validated on the validation set. The models were tuned with important hyperparameters including class balancing weight. Final evaluation was conducted on the test set. The following models were implemented and optimized:

1. Logistic Regression: A baseline linear model.
2. Decision Tree: A simple, interpretable tree-structured model capturing non-linear relationships.
3. Support Vector Machine (SVM): A kernel-based approach for non-linear classification.
4. Random Forest: An ensemble of decision trees for robust performance.
5. AdaBoost: A boosting algorithm combining weak learners.
6. XGBoost: An optimized gradient boosting method known for its speed and accuracy.
7. Neural Networks: A deep learning approach using a fully connected feedforward architecture.

Results

Model	**AUC**	**Precision**	**Recall**	**F1-Score**
Logistic Regression	0.8500	0.6452	0.8004	0.7144
Decision Tree	0.9037	0.8257	0.7264	0.7729
Support Vector Machine	0.9106	0.7457	0.8213	0.7817
Random Forest	0.9325	0.8509	0.7602	0.8030
AdaBoost	0.9042	0.7124	0.8173	0.7612
XGBoost	0.9341	0.7867	0.8254	0.8056
Neural Networks	0.9247	0.7303	0.8523	0.7866

The test set performance of all models highlighted distinct strengths and trade-offs:

1. Logistic Regression: The model achieved a relative low AUC of 0.8500, the model's simplicity limited its ability to capture complex patterns, leading to a relatively lower F1-score (0.7144). Precision (0.6452) and recall (0.8004) for bots (class 1) indicates that it was effective in detecting bots but struggled with the relatively high false positive rate despite applying class weighting.
2. Decision Tree: The Decision Tree provided significant improvement in precision for bots (0.8257) compared to the baseline model Logistic Regression and achieved an AUC of 0.9037. However, its recall dropped to 0.7264, indicating that the model likely prioritized precision over recall. The single-tree structure limited its ability to make predictions with higher precision.
3. Support Vector Machine (SVM): SVM performed well with an AUC of 0.9106 and balanced precision-recall for bots (F1-score of 0.7817). Its kernel-based approach helped

in capturing non-linear relationships effectively, increasing both precision and recall compared to the baseline model, but came at a high computational cost.

4. Random Forest: This ensemble method achieved the highest precision for bots (0.8509) and an high AUC of 0.9325 indicates its strong predictive power and robustness. It balanced the trade-off between recall and precision effectively (F1-score of 0.8030). However, the recall score dropped slightly compared to the baseline model, indicating that it traded off recall to achieve excellence in other scores.
5. AdaBoost: While AdaBoost showed higher AUC, precision, recall, and F1-score compared to the baseline model, it also showed competitive recall for bots (0.8173) compared to other non-linear models. However, its overall AUC (0.9042) were lower compared to Random Forest and XGBoost, highlighting its limitations in handling the complexity of the dataset. The sequential learning process might make it more sensitive to noise in the data.
6. XGBoost: XGBoost emerged as the best-performing model with the highest AUC of 0.9341, striking a balance between precision (0.7867) and recall (0.8254) for bots. Its ability to handle class imbalance and its gradient boosting framework enabled it to capture complex relationships, making it the most robust option.
7. Neural Networks: Despite achieving a high AUC of 0.9247 and the best recall for bots (0.8523), when compared to the other non-linear models such as Random Forest and XGBoost, Neural Networks slightly lagged in precision (0.7303) and F1-score (0.7866). This indicated potential overfitting or sensitivity to hyperparameter tuning.

Discussion

The results indicate that the ensemble method XGBoost is most effective for this task, likely due to its ability to capture complex, non-linear relationships in the data. Logistic Regression and Decision Tree, while interpretable, struggle to match the performance of these ensemble methods. Neural Networks show promise but require further tuning to justify their higher computational cost.

Given the nature of the dataset and the problem, XGBoost is the recommended model due to its superior performance across all metrics and its scalability for larger datasets. Its robust performance across metrics makes it ideal for detecting bots while minimizing false positives and negatives. Besides, its feature importance analysis could also provide actionable insights into the characteristics of bot accounts.

Conclusion

This project successfully applied machine learning techniques to classify Twitter accounts as human or bot. The comprehensive exploratory data analysis process ensured data quality and interpretability, while the implementation and optimization of multiple models provided robust benchmarks for performance comparison. Future work could explore incorporating interpretability methods, such as SHAP or LIME, to better understand the features influencing the model's decisions.

By leveraging XGBoost, the proposed solution offers a practical and efficient approach for detecting bots, contributing to the broader effort to mitigate misinformation on social media platforms.