Universidad Panamericana
Engineering Faculty

Machine Learning I class
Final project
December 5, 2023

Social Media Bot Detection Project

David Arturo Hernández Gómez
Adriana Leticia Martínez Estrada

1 Code execution instructions

This section provides a comprehensive guide for executing the project code developed as part of the *Social Media Bot Detection* project. The project is structured into two Jupyter notebooks hosted on Google Colab, each serving distinct purposes within the project framework.

1.1 Main Project Notebook

GitHub Repository: Main Project Notebook

Execution Environment: Google Colab

Prerequisites: Ensure that all the required libraries are installed. The list of dependencies can be found in the notebook or the accompanying `requirements.txt` file in the GitHub repository.

Execution Steps:

1. Open the notebook in Google Colab.
2. Run each cell sequentially. Cells include:
    - Importing necessary libraries and modules.
    - Data loading and preprocessing steps.
    - Model training, including both conventional approaches and those utilizing Grid Search.
    - Model evaluation, where models are tested and compared based on various metrics.
3. Review the output at each stage for understanding the data processing, model performance, and evaluation results.

1.2 Interactive Application Notebook

GitHub Repository: Interactive App Notebook

Execution Environment: Google Colab **Prerequisites**: Ensure that the necessary libraries, as listed in the notebook, are installed. This includes Gradio for the interactive web interface.

Execution Steps:

1. Open the notebook in Google Colab.
2. Install all required libraries using the provided pip command.
3. Run the notebook cells sequentially, which include:
    - Loading the trained model from the repository.
    - Defining the preprocessing function and the prediction function.
    - Setting up and launching the Gradio interface.
4. Utilize the Gradio web interface to input data and receive predictions from the trained model.

Note: Ensure a stable internet connection for seamless execution and interaction with the Colab notebooks.

2 Introduction

The emergence of social media platforms has revolutionized the way people communicate and disseminate information.(Hanson 2016) Alongside the rapid growth of these platforms, there has been an increase in the presence of automated accounts, commonly referred to as 'bots'. These bots can significantly impact the dynamics of social media, ranging from performing benign tasks like automated updates to more malicious activities such as spreading misinformation or spam. The *Social Media Bot Detection* project aims to address the growing concern of distinguishing these automated accounts from genuine human users.

In the digital era where social media plays a pivotal role in shaping public opinion and disseminating news, the ability to accurately identify bot accounts has become crucial. Bots are often employed to manipulate discussions, influence public opinion, or spread false information, posing threats to the authenticity of online discourse. Therefore, distinguishing between human and bot-operated accounts on social media platforms is not only a technical challenge but also a necessity for maintaining the integrity of digital communication. (Gilani et al. 2017; Haustein et al. 2016; Subrahmanian et al. 2016)

This project undertakes the challenge of developing a supervised machine learning model capable of accurately identifying bot accounts based on their social media behavior. The model is trained and tested on a dataset comprising various behavioral features of known bot and human-operated accounts. Our approach combines advanced data processing techniques with machine learning algorithms to analyze patterns and characteristics indicative of bot-like activities.

The primary objectives of this project are threefold:

- To analyze and understand the distinguishing features and behavior patterns of bots compared to human users on social media platforms.
- To develop a robust binary classification model that can effectively differentiate between bot and human-operated accounts.
- To evaluate the model's performance using various metrics, ensuring its reliability and accuracy in real-world applications.

By achieving these objectives, this project aims to contribute to the broader effort of ensuring the authenticity and reliability of social media as a platform for genuine human interaction and information exchange.

3 Dataset

The datasets used in this project provide a comprehensive overview of both human and bot-operated Twitter accounts (Yang et al. 2022). These datasets were obtained from the *Botometer* project datasets chosen for this study are from the year 2019, as they offer a rich dataset with diverse and detailed account information. It is important to note that with Twitter's recent changes in API access following its acquisition, the *Botometer* project's data extraction method has been deprecated. [1]

3.1 Dataset Description

The project uses two primary datasets:

1. **verified-2019:** This dataset consists of Twitter accounts verified as human. It includes labels and user objects, providing a reliable ground truth for human-operated accounts.
2. **botwiki-2019:** This dataset comprises self-identified bots from https://botwiki.org. Similar to the human dataset, it includes labels and user objects for bot accounts.

3.2 Features

Both datasets contain the same set of features for each account, ensuring a consistent comparison between human and bot behaviors. The features are as follows:

- **Location:** Registered location of the user.
- **Followers Count:** Number of followers the account has.
- **Friends Count:** Number of other users the account is following.
- **Listed Count:** Number of lists the account is included in.
- **Favourites Count:** Number of tweets the account has liked.
- **Geo Enabled:** Whether the account has geolocation features enabled.
- **Verified:** Whether the account is verified by Twitter.
- **Statuses Count:** Total number of tweets and retweets by the account.
- **Profile Background Tile:** Whether the account uses a background tile image.
- **Profile Use Background Image:** Whether the account has a background image.
- **Has Extended Profile:** Whether the account has an extended profile.
- **Default Profile:** Whether the account uses the default Twitter profile settings.

3.3 Data Source and Selection

The decision to use data from 2019 was driven by the richness and diversity of the dataset available from that year. This period provides a significant snapshot of Twitter's user base, encompassing a wide variety of account types and activities, which is critical for developing a robust and accurate bot detection model.

[1] Botometer project: https://botometer.osome.iu.edu

4 Approach

This project adopts a comprehensive approach to classify Twitter accounts into humans and bots. We leverage advanced machine learning algorithms and data preprocessing techniques to analyze the patterns and characteristics that differentiate these two types of account behaviors.

4.1 Preprocessing

The preprocessing of the data is a critical step in ensuring the models receive clean, normalized, and relevant features for effective learning. The preprocessing steps include:

- **Data Cleaning:** Removing irrelevant features, handling missing values, and filtering out noise from the data.
- **Feature Selection:** Choosing the most informative features that contribute significantly to the classification task.
- **Data Transformation:** Normalizing and standardizing numerical data, and applying One-Hot Encoding to categorical data for a consistent scale across all features.

This preprocessing not only cleans and prepares the data but also ensures that the machine learning models can interpret the data correctly and efficiently.

4.2 Machine Lerning Algorithms

The project employs a variety of machine learning models, each with its unique strengths and applicability to the problem at hand:

1. **Ridge Classifier:** Used for its efficiency in handling multicollinearity and preventing overfitting through L2 regularization.
2. **Decision Tree Classifier:** Chosen for its interpretability and ability to model non-linear relationships without requiring feature scaling.
3. **Support Vector Machine (SVM):** Utilized for its effectiveness in high-dimensional spaces and versatility in kernel selection.
4. **Logistic Regression:** A robust baseline model known for its simplicity and efficiency in binary classification problems.

Each model offers distinct advantages. For instance, the Ridge Classifier mitigates overfitting, while the Decision Tree provides clear insights into feature importance. The SVM's flexibility in handling various types of data and Logistic Regression's efficiency make them valuable additions to our model ensemble.

4.3 Evaluation techniques

The performance of the models is evaluated using the following metrics:

- **Accuracy:** Measures the proportion of correctly classified instances.
- **Precision and Recall:** Evaluates the model's ability to correctly classify positive instances.

- **F1-Score:** The harmonic mean of precision and recall, providing a balance between the two.
- **AUC-ROC Curve:** Assesses the model's ability to discriminate between classes across different thresholds.

From a data science perspective, the combination of these models and evaluation techniques allows for a balanced and thorough analysis. Each model contributes uniquely to the ensemble, and together, they provide a diverse perspective on the data, enhancing the overall predictive power and reliability of our solution.

5 Experiments

In the course of our study, we conducted a series of experiments to evaluate the performance of various machine learning models. The models tested were Ridge Classifier, Decision Tree Classifier, Support Vector Machine (SVM), and Logistic Regression. Each model was assessed both in its conventional form and with hyperparameters optimized through Grid Search.

5.1 Model Parameters

- The Ridge Classifier was tested with its default regularization strength.
- The Decision Tree Classifier was evaluated with default settings and then optimized with a grid search over a range of maximum depths, minimum samples to split, and minimum samples per leaf.
- The SVM was run with its default parameters and subsequently with a grid search over various values for the penalty parameter C, the kernel coefficient γ, and different kernel types.
- The Logistic Regression was used with default hyperparameters.

5.2 Training Process

The training times varied significantly among the models. The Decision Tree Classifier with Grid Search required the most time to train due to the exhaustive search over the hyperparameter space. In contrast, conventional models were faster to train, with the Ridge Classifier being the quickest.

5.3 Evaluation Metrics

The models' performance was evaluated based on accuracy, f1-score, and recall. Moreover, the Receiver Operating Characteristic (ROC) curves and Confusion Matrices were generated to provide a visual representation of the models' capabilities in distinguishing between the classes.

6 Results

The results from our experiments are presented below, detailing the performance of each model in terms of accuracy, f1-score, and recall. Additionally, we include the training time to give an insight into the computational efficiency of each model.

6.1 Model Performance

Model Set	Model Name	Accuracy	F1-Score	Recall
GridSearch	RidgeClassifier	0.740	0.629	0.740
GridSearch	DecisionTreeClassifier	0.975	0.975	0.975
GridSearch	SVC	0.965	0.965	0.965
GridSearch	LogisticRegression	0.940	0.941	0.940
Conventional	RidgeClassifier	0.740	0.629	0.740
Conventional	DecisionTreeClassifier	0.971	0.971	0.971
Conventional	SVC	0.922	0.925	0.922
Conventional	LogisticRegression	0.940	0.941	0.940

Table 1: Summary of Model Performance

The Decision Tree Classifier with Grid Search emerged as the most accurate model, as reflected by the metrics shown in Table 1. This model not only achieved the highest accuracy but also scored equally high in terms of f1-score and recall.

6.2 Visual Representation of Results

For a more intuitive understanding of model performance, we provide visual representations in the form of ROC curves and Confusion Matrices.

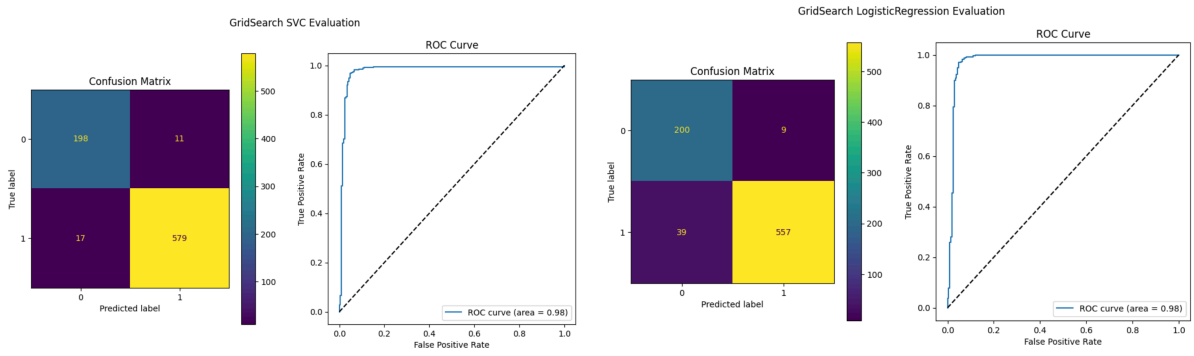

Figure 1: ROC Curve and Confusion Matrix for GridSearch SVC and Logistic Regression models.

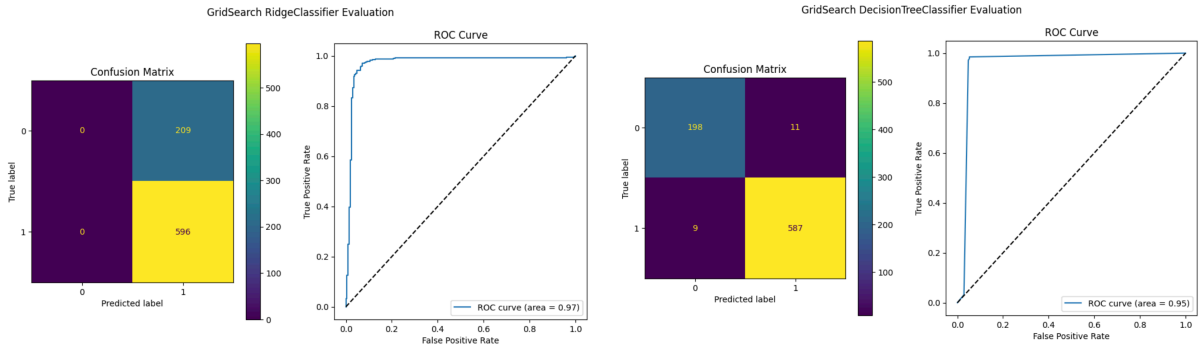

Figure 2: ROC Curve and Confusion Matrix for GridSearch Ridge Classifier and Decision Tree Classifier models.

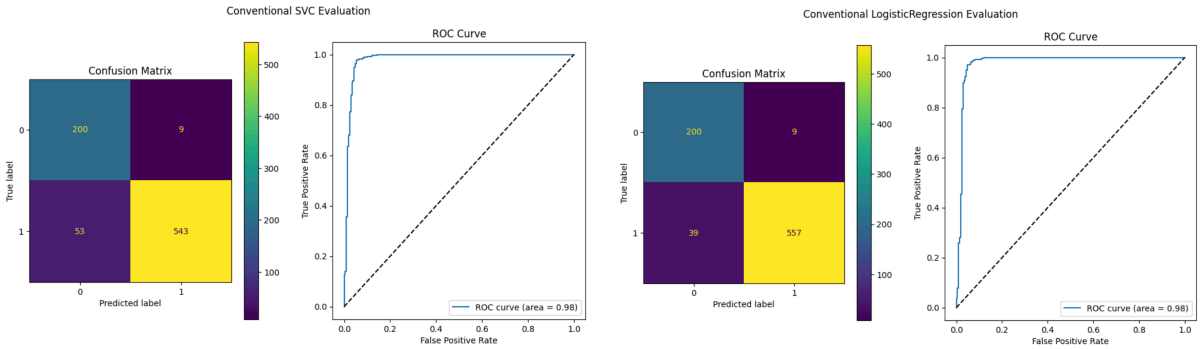

Figure 3: ROC Curve and Confusion Matrix for Conventional SVC and Logistic Regression models.

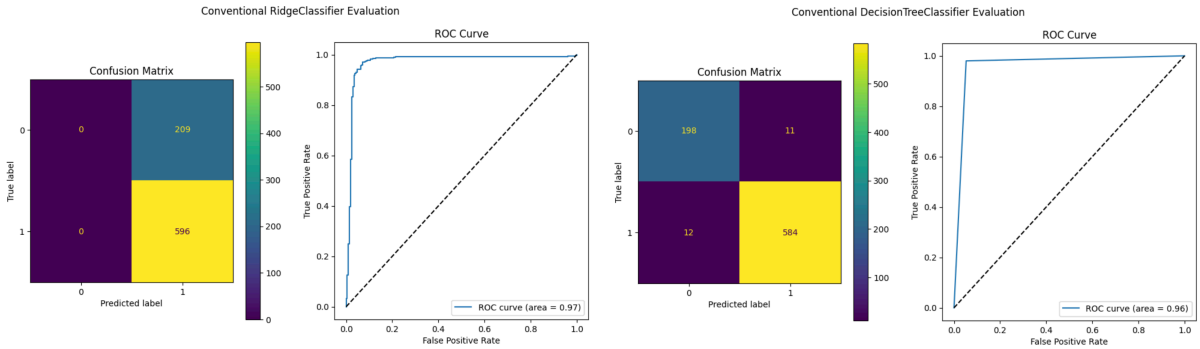

Figure 4: ROC Curve and Confusion Matrix for Conventional Ridge Classifier and Decision Tree Classifier models.

The ROC curves in Figures 1, 2, 3, and 4 illustrate the trade-off between true positive rate and false positive rate for each model. The closer the curve follows the left-hand border and then the top border of the ROC space, the more accurate the test.

6.3 Comparative Analysis

Upon comparing the GridSearch models with their Conventional counterparts, it is evident that the optimization of hyperparameters through GridSearch has generally led to better performance metrics, particularly in the case

7 Conclusions

7.1 Adriana Martinez

Through the course of this project, we have successfully developed and evaluated a set of machine learning models to distinguish between bot and human-operated social media accounts. Our experiments have led to the identification of a highly effective model, with the Decision Tree Classifier optimized via GridSearch showcasing superior performance metrics.

7.1.1 Project Achievements

The project's primary accomplishment lies in the model's ability to generalize and accurately classify accounts with a remarkable accuracy rate. The chosen features and modeling approach have proven to be effective in capturing the underlying patterns that differentiate bots from human users.

7.1.2 Recommendations for Future Work

Refactoring and Modularity: Future efforts should focus on code refinement for enhanced clarity and reusability. Functions ought to be succinct, serving single responsibilities and facilitating easy maintenance and scalability.

Exception Handling: Robust exception handling strategies must be implemented to ensure stability and reliability, particularly in data loading and model training phases.

Parameter Optimization: More efficient hyperparameter optimization methods, such as RandomizedSearchCV, should be considered over GridSearchCV to enhance the model tuning process.

Documentation and Comments: Comprehensive documentation and in-line comments are critical for long-term code sustainability and should be improved to facilitate better understanding.

Cross-Validation: Employing cross-validation methods would provide a more robust assessment of model performance as opposed to a simple train-test split.

Model Evaluation: Expanding evaluation metrics to include precision, F1-score, and the precision-recall curve would offer a more nuanced view of model performance.

Data Visualization: Incorporating advanced data visualization techniques would aid in a deeper understanding of both the data characteristics and model outcomes.

Error Analysis: Post-evaluation, an in-depth error analysis should be conducted to uncover the model's weaknesses and direct future improvements.

7.2 David Arturo

The endeavor to classify Twitter accounts using machine learning models has yielded promising results. The diverse data samples have enabled the model to effectively generalize across different classes.

7.2.1 Project Insights

The chosen model exhibits robustness, as evidenced by high accuracy and F1-score on the test data. This affirms the model's correct adjustment to the data and its readiness for deployment in a real-world setting.

7.2.2 Outlook and Future Adaptations

With the evolution of social platforms, it is crucial to continually update the dataset to maintain model relevance. Feature extrapolation to newer platforms and the integration of updated bot and human account data will be vital for ongoing model accuracy and applicability.

Data Update and Expansion: For future iterations, an updated dataset reflecting the current social media landscape is necessary. The quest for new bot and human account samples will enhance the model's learning capability.

Feature Engineering: Advanced feature engineering techniques can be explored to further refine the model's predictive power.

Adaptability to New Platforms: Given the dynamic nature of social media, the model's flexibility to adapt to new platforms, such as the successor of Twitter, will be key to its sustained success.

8 Graphical User Interface (GUI)

The development of a user-friendly graphical interface is an essential aspect of our project, facilitating the practical application of our predictive model. Housed within the 'APP.ipynb' notebook, the interface serves as a conduit between raw user data input and the model's prediction output, discerning bot from human-operated Twitter accounts.

8.1 Preprocessing and Model Integration

A crucial component of the GUI is the preprocessing pipeline, implemented within the same notebook. This pipeline mirrors the data processing steps utilized during model training, including scaling and One-Hot Encoding (OHE), thus obviating the need for users to enter pre-scaled or encoded data. The seamless integration ensures that the model receives data in the correct format for accurate predictions.

8.2 Model Importation and Application Deployment

The predictive model is dynamically retrieved from a GitHub repository, ensuring the application uses the most updated version. The retrieval process is executed via HTTP requests, which fetch the serialized model file for use within the application.

8.3 Results and User Interaction

Upon successful importation of the model, the Gradio framework is utilized to launch the application. Gradio's intuitive design enables the end-user to input raw data in a comprehensible format. The application's backend processes this input through the established preprocessing pipeline before presenting it to the model. The model then evaluates the data and renders a prediction, which

is displayed to the user. This prediction indicates whether the account in question is likely to be a bot or a genuine human user.

8.4 Conclusion

The GUI represents a significant stride towards democratizing the use of our machine learning model. It embodies our commitment to making sophisticated technology accessible and user-friendly, allowing users with varying levels of technical expertise to leverage our model's capabilities for their analyses.

References

Gilani, Zafar, Reza Farahbakhsh, and Jon Crowcroft (2017). "Do bots impact Twitter activity?" In: *Proceedings of the 26th International Conference on World Wide Web Companion*, pp. 781–782.

Hanson, J. (2016). *The Social Media Revolution: An Economic Encyclopedia of Friending, Following, Texting, and Connecting*. Bloomsbury Publishing. URL: https://books.google.com.mx/books?id=bhbHEAAAQBAJ.

Haustein, Stefanie, Timothy D Bowman, Kim Holmberg, Andrew Tsou, Cassidy R Sugimoto, and Vincent Larivière (2016). "Tweets as impact indicators: Examining the implications of automated "bot" accounts on T witter." In: *Journal of the Association for Information Science and Technology* 67.1, pp. 232–238.

Subrahmanian, Venkatramanan S, Amos Azaria, Skylar Durst, Vadim Kagan, Aram Galstyan, Kristina Lerman, Linhong Zhu, Emilio Ferrara, Alessandro Flammini, and Filippo Menczer (2016). "The DARPA Twitter bot challenge." In: *Computer* 49.6, pp. 38–46.

Yang, Kai-Cheng, Emilio Ferrara, and Filippo Menczer (2022). "Botometer 101: Social bot practicum for computational social scientists." In: *Journal of Computational Social Science* 5.2, pp. 1511–1528.