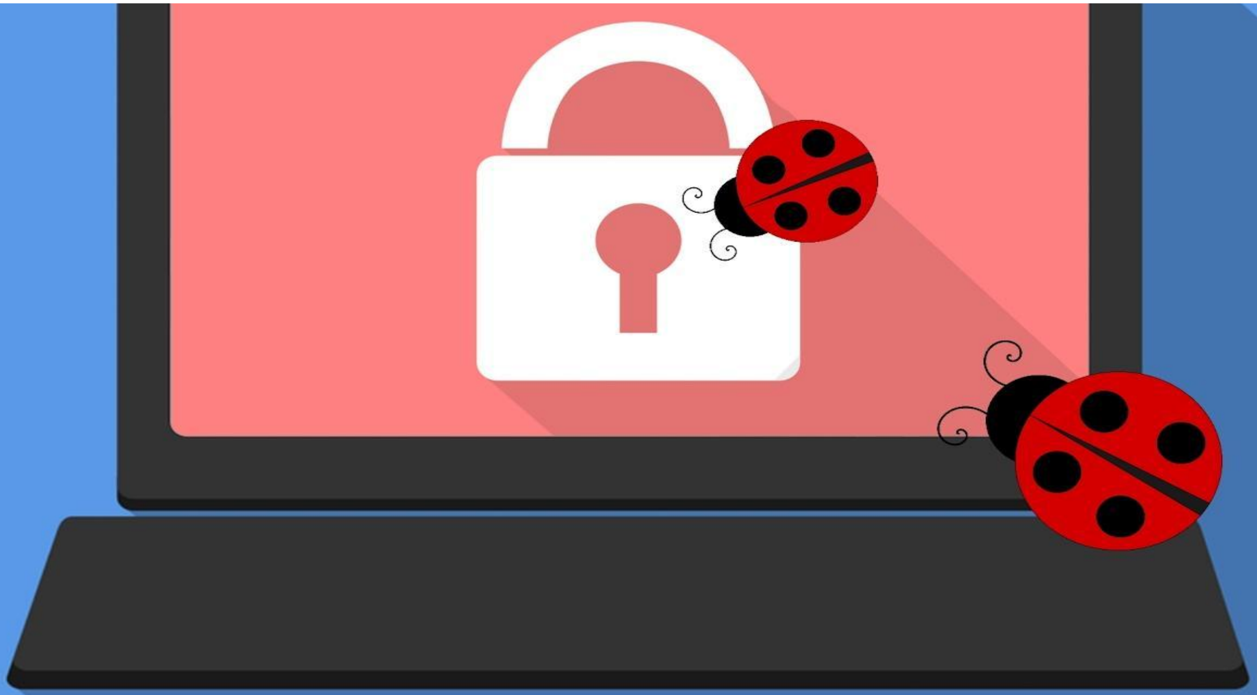
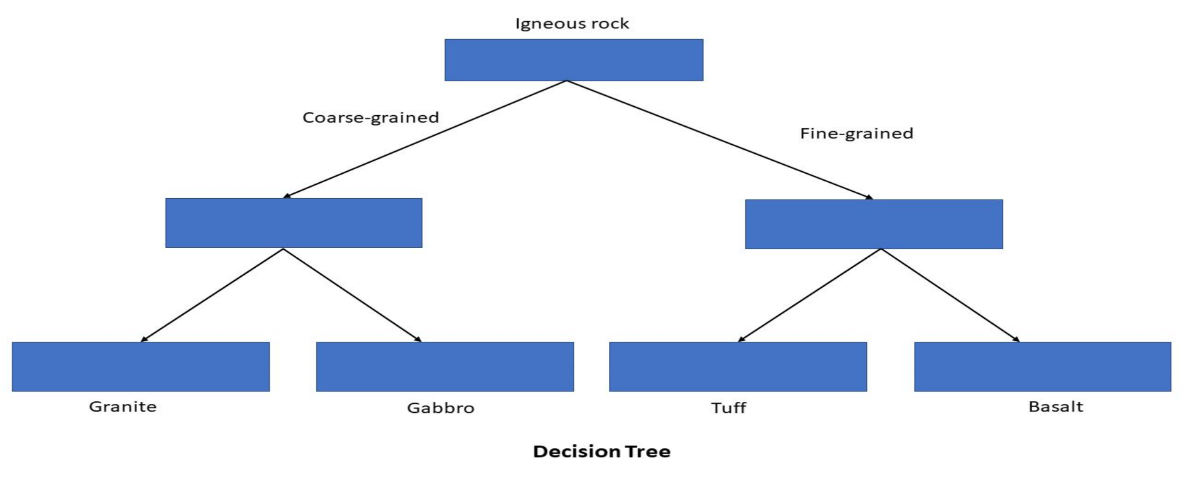
Mischief Managed: Malware Detection



By Anthony Brocco

Our Business Goals



Decision Tree

This project aims to develop a decision tree classifier, to identify and categorize malware on Android devices for businesses as a contracted business.

-Malware poses a significant threat to both individual users and organizations, making its timely detection crucial.

-Decision trees are chosen as the modeling technique due to their interpretability and ability to handle complex feature interactions.

Business Standpoint

Business Understanding:

- Why Detect Malware: A robust malware detection system is essential for protecting users' personal information
- Impact of False Positives/Negatives: False positives (flagging non-malware as malware) can lead to user frustration and unnecessary actions, while false negatives (missing actual malware) can have severe security implications. Therefore, the focus is on minimizing false negatives to enhance security.
- User Experience: Minimizing false positives is also critical to maintaining a positive user experience. Frequent false alarms can lead to user dissatisfaction and potential distrust in the security system.

Understanding Our Data

Data Source: Our dataset comprises diverse Android application features, including permissions, system calls, and API calls. These features serve as inputs for our decision tree

Target Variable: We're dealing with a binary target variable, where 1 signifies malware, and 0 denotes goodware (non-malicious apps).

Feature Importance: Identifying the most influential features is critical. We'll rank the features to pinpoint what aspects of Android apps have the most impact on distinguishing malware from goodware.

Data Splitting: To gauge model performance, we'll divide the dataset into training and testing sets. Our decision tree classifier will learn from the training data and be evaluated on unseen data, considering metrics like accuracy, precision, and recall.

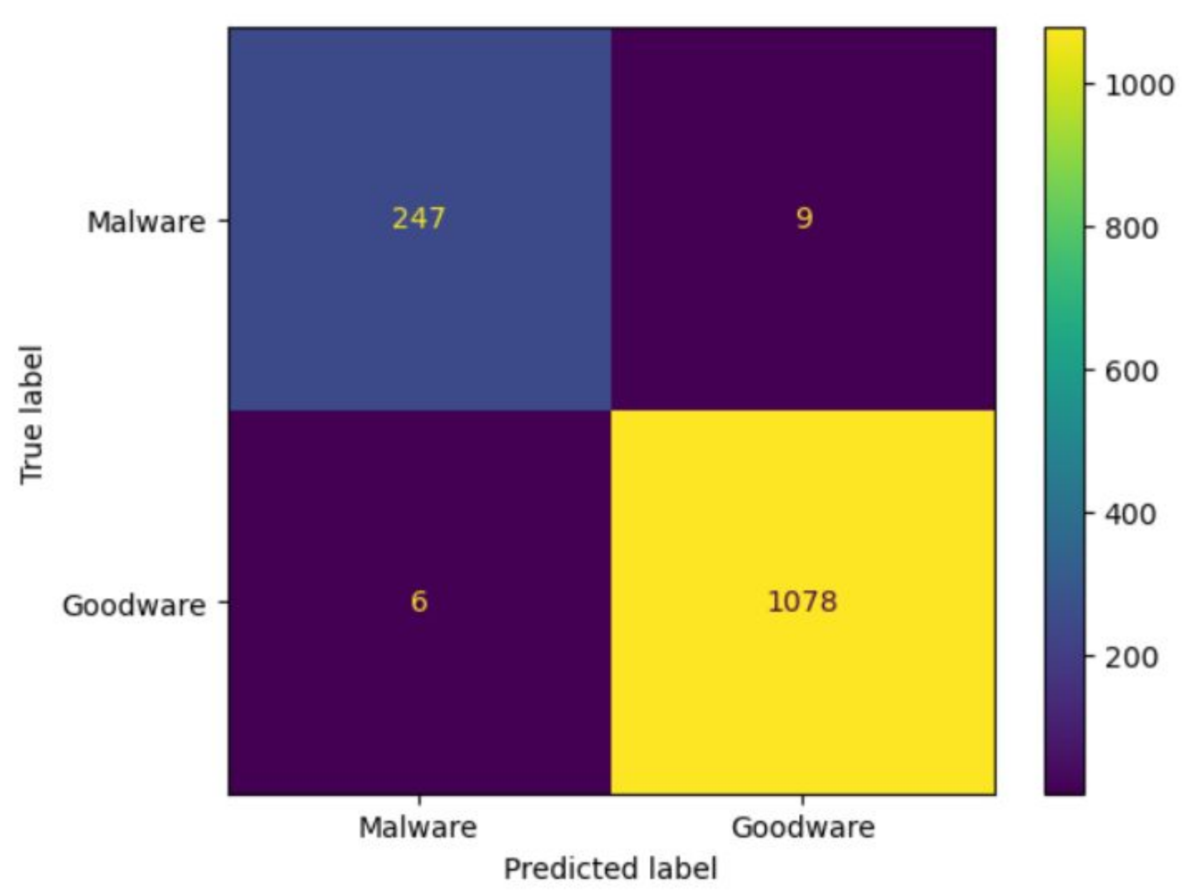
Why Decision Trees?

- Both before and after Refining a list of most important features, the decision tree steadily outperformed its logistic regression counterpart. Most importantly in the recall metric, which is important for reducing false negatives
- Due to its interpretability and ability to capture complex feature interactions, which are important for detecting malware patterns in Android applications.

How do we evaluate our models?

- Accuracy, Precision, Recall: Key metrics for assessing model performance, with a focus on minimizing false negatives to avoid missing malware.
- Confusion Matrix: Provides a detailed breakdown of model predictions, aiding in understanding false positives and false negatives.
- Feature Importance Analysis: Helps identify which aspects of Android applications contribute most to distinguishing malware from goodware.
- Learning Curve Analysis: Assesses how the model's performance scales with increasing data, aiding in data collection decisions.

Confusion Matrix of our final Decision Tree model



Our Future

Recommendations:

- Continue gathering more diverse and up-to-date data to improve model generalization.
- Collaborate with cybersecurity experts to gain deeper insights into malware behaviors and features.

Next Steps:

- Implement the model in a real-world Android security application for continuous monitoring.
- Develop an intuitive user interface for non-technical users to benefit from the model's insights.
- Stay updated with emerging Android malware trends and adapt the model accordingly.

Thank you for your time!

Feel free to message me on linkedin or github, as well as email meif you have any further questions.

https://github.com/brocc12

https://www.linkedin.com/in/anthony-brocco-480b0818a/

anthonybrocco98@gmail.com