Malware Prediction System

Background and Context

Cybersecurity stands out as a crucial challenge in today's digital age. With more personal and professional tasks moving online, there's an escalating risk of malware intrusions. Predicting and proactively guarding against such threats can redefine how we approach digital safety. Malware incidents might result in exposure of confidential data, monetary setbacks, and a drop in system efficiency, to list a few ramifications. This project aims to harness the power of machine learning to forecast the probability of a computer being targeted by malware.

Data Source Identification

Microsoft Malware Prediction | Kaggle

The data set we are using is Microsoft's Malware prediction competition dataset from Kaggle. The dataset has around 9 million rows and 83 columns, making it a very large, industrial-scale dataset. The test set consists of about 8 million predictions to be made, making it a challenging problem. Each row represents a unique machine with all its specifications, notably the version and software the machine operates on as well as various other specifications. The variable to be predicted is the Has_detections column, which indicates whether or not malware was detected on the machine. It should be noted that the data is roughly split by time, so it includes older versions of softwares that may no longer be widely used, and may need to be handled during analysis.

Proposed ML techniques

1. Logistic Regression
    a. This is a straightforward yet efficient method for binary classification tasks
    b. A good starting point as a baseline model for its simplicity and interpretability
2. Random Forest Classification.
    a. We have a very large amount of features to choose from, and RF's could be used to perform innate feature selection.
    b. Random Forests are usually very effective for classification problems involving large datasets.
    c. RF's are also very flexible, boasting various hyperparameter tuning options that we could use to finetune the model for different scoring parameters.
3. Gradient Boosting
    a. They can capture more complex patterns and hence result in better accuracy.
    b. Being more sensitive to noise, we would like to experiment with its efficacy in solving this problem.
4. Neural Networks
    a. Neural Networks can help us learn higher-order information from the data.
    b. In case our feature engineering approaches do not work best, Deep Learning can help us circumvent some of the issues.
    c. Having ample data in the dataset, deep learning is a worthy experiment to compare performance.