Malware Prediction:

Predictive Modeling of Systems Vulnerable to Malware

DSC 630 – Predictive Analytics

Elena Adame & Alan Donahue

13 August 2022

In the last few years, various forms of malicious software, or malware, have been released on the public domain to wreak havoc and harvest credentials that can be used to pilfer money or steal sensitive information. The most notable attack was the Solarwinds hack beginning in December 2020, which compromised the supply chain of the software developed by Solarwinds. One customer of Solarwinds software, Microsoft, was able to confirm that attackers were able to take advantage of Microsoft program configuration weaknesses to probe deeper into the network. There is no surprise that malware would be used to move deeper into the target space. PCMag reported that of all malware attacks that occurred in the first few months of 2020, Microsoft Windows computers were the primary targets, being hit 83% of the time. (Cohen, 2020).

While Microsoft Defender works to catch all types of emerging threats, it works to catch threats that have only made it into the network. What Microsoft Defender does not do is provide an assessment of the vulnerability of the system based on inherent factors or traits. This research group has identified a dataset to identify the likelihood of machine or system being targeted or compromised by malware based on the various traits or properties of the associated machine or system. The output of this research could be incorporated into the already existing infrastructure of Microsoft Windows Defender to provide a more holistic understanding of the network and to work to proactively secure identified vulnerable systems.

The data for this project was collected by Microsoft in 2019 from 8.9 million various Microsoft systems. The dataset contains specific machine information such as the version and build of the current Operating System (OS) as well as other Device Census data. These attributes are critical to developing a model that can predict malware presence as these are the attributes targeted for exploitation from malicious attackers. Microsoft has also provided the ‘HasDetections’ variable. This variable indicates whether Microsoft Defender detected the

presence of malware on the associated system or not. A ‘1’ indicates positive confirmation of malware detected and a ‘0’ indicates no malware has been detected. In looking at this problem statement, we used the ‘HasDetections’ variable as our target variable. We aimed to develop a model that, based on the ‘HasDetections’ variable, would identify systems with attributes that led to malware infecting the system. As there are only two states that the system can exist in, with malware or without, we marked this issue as a binary classification problem. Initially, we proposed three models based on this binary classification: Logistic Regression, Decision Tree, and K-Nearest Neighbor (KNN). What we discovered, however, was that we could not use KNN for this problem statement. The dataset, even when stripped down to 6 out of 83 features, still contained over 1 million unique datapoints; utilizing KNN as our model for this project was not feasible due to the limited processing power of our equipment.

For measuring the performance of our model, we settled on running a Classification Report of each of the models. We chose to focus primarily on the Recall value of our models. Keeping in mind that fixing and securing vulnerable systems is costly and time-consuming, we needed to produce a model that would quickly and accurately predict and isolate the correct systems. A model that produced many False Negatives would be considered a failure, as the model would incorrectly identify systems as not vulnerable when they were in fact vulnerable. This type of model would also leave a company wide open to exploitation as it would not know which systems were truly at risk of being targeted or infected with malware. The Recall metric directly relies on the number of False Negatives as it is the ratio of True Positives to the sum of True Positives and False Negatives. The lower the number of False Negatives, the better our Recall and the better our model.

From the initial Exploratory Data Analysis (EDA) we performed on the dataset, we were able to confirm that the dataset was not biased toward the presence of malware on a system or the lack of

it. Figure 1 below shows that the dataset is near evenly split between systems with malware and systems without malware.

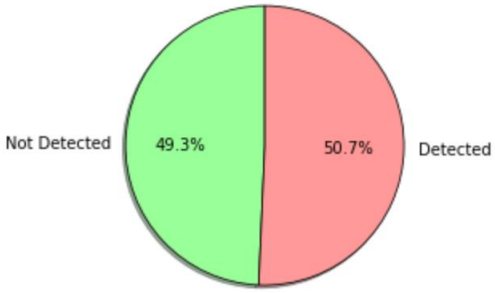

Figure 1: Percent of machines detected with malware. Data in chart is from the 'HasDetections' column within the dataset.

Looking at other columns in the dataset, we made the decision to remove columns that were missing more than 30% of their values. In total, 9 columns were removed from the dataset. After examination, these columns were determined to not be pertinent to the model creation. For missing values that remained after dropping the 9 columns, we removed any rows from the dataset that included missing values.

We severely underestimated the necessary computing power needed for this project. This dataset contained over 7 million unique system samples. Each sample had 74 variables associated with each sample. Additionally, a majority of the 74 variables were categorical variables, requiring us to create dummy variables to perform the logistic regression model. We received a Memory Error when we attempted to do this. We scaled the sample size down to 10,000 samples and then down to 5,000 samples. Even to create dummy variables for those 5,000 samples required more memory than we possessed, 74 TB to be specific.

To get around this memory error, we made the decision to severely scale back our features to 6 features we felt could provide us with the best prediction. We chose the following features: Engine Version, Platform, OS Version, SMode, Firewall, and Census – Portable Operating System. These features were chosen based on this research group's current understanding of cybersecurity and the features most likely to be targeted when malware looks for implantation. We particularly chose to include SMode as this is a feature Microsoft allows that restricts the system to only downloading and using Microsoft licensed applications. This would restrict the computer from downloading third-party applications from the internet which could potentially contain malware. OS Version was chosen as malicious cyber attackers can pinpoint vulnerabilities within a computer's OS version to exploit and deliver their malware.

When beginning with model creation, we created and deployed a baseline classification model. This was done to take an initial look at how a model could potentially perform and what aspects we would need to fix. The output of the model showed that 25.3% of the data was identified as False Negatives. However, the model did correctly predict 50% of the data.

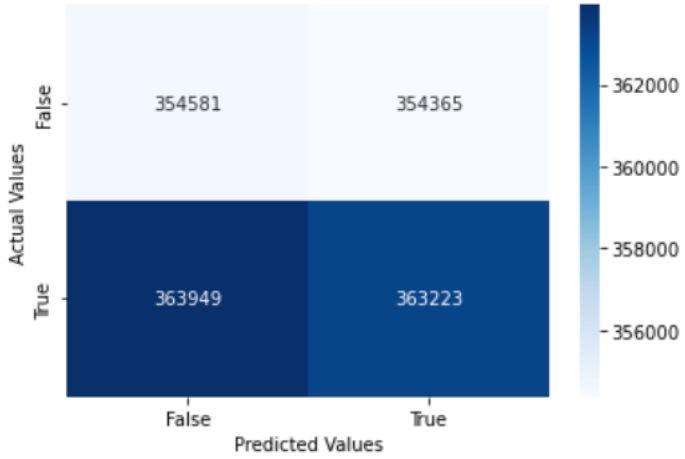

Figure 2: Baseline Classification Confusion Matrix

With this in mind, we moved on to creating a Decision Tree model, hoping that we could correctly label a larger percentage of the data than the baseline. However, the Decision Tree model also performed poorly. Our target F-1 Score was 0.70. This model produced an F-1 Score of 0.53 and 0.56 for systems with and without malware, respectively. Additionally, the confusion matrix for this model was very similar to that of the baseline.

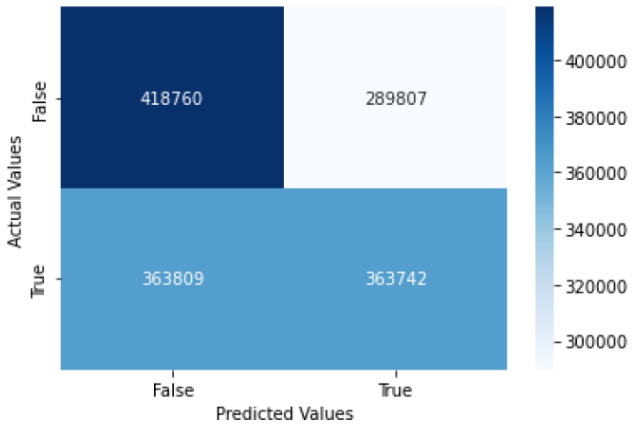

Figure 3: Confusion matrix for Decision Tree

Keeping the same features, we then utilized a Grid Search on the Decision Tree model in an effort to perform hyperparameter tuning of the model. We found that the best hyperparameters were a Criterion of Entropy, Max Depth of 20, and Minimum Sample Leaf of 50. With these parameters in place, we adjusted the Decision Tree model and ran it again. These changes did not increase the scores of Recall or F-1. The confusion matrix (Figure 4) did show a marginal increase in the number of False Negatives produced (+ 264). We determined that adjusting the hyperparameters did not truly help the model.

Next, we built out a Logistic Regression model to determine if this would produce better results. We found that this model was similar to the original Decision Tree model. The model, while producing similar numbers for the confusion matrix, produced the same F-1 and Recall scores.

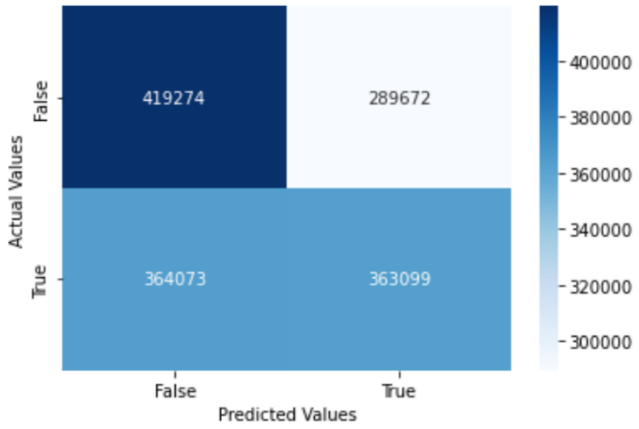

Figure 4: Confusion Matrix for Decision Tree with adjusted Hyperparameters

After identifying that adjusting the hyperparameters of the Decision Tree model produced marginal differences with regard to model output, we next adjusted what features we were using. We swapped out the Census Data for Portable Operating Systems to the Census Data for Device Family. This variable reports the type of device that the OS is intended for (Windows Desktop, Windows Mobile, etc.). However, while adjusting the features in this manner did not change the F-1 or Recall Scores, it did produce the worst confusion matrix out of all the models. The model was only able to correctly predict 50% (versus the previous 55%) of the data. Additionally, this model produced the largest number of False Negatives, with 27.7% of the data being incorrectly labeled as such.

As we were limited by computing power and the time it took for each of the models to run, we had no choice but to leave the models with a Recall score of 0.59. While this is not ideal, it is 0.11 points away from our target Recall score of 0.70. We feel that increasing the total number of features included in the model would increase the Recall score. However, we cannot do this model creation with our current equipment and resources. The original error we saw when attempting to build our model weeks ago with all features showed that 74 TB of memory would be needed. We feel that a company would need to invest in cloud computing to be able to perform the calculations

required for the model. Additionally, we acknowledge that Microsoft collected this data from it's endpoint protection solution, Windows Defender. Windows Defender automatically prepares reports with system information and sends the report back to Microsoft. To turn this feature off, a user must manually disable the automatic reports sent to Microsoft. As such, we are aware that the data came from an unstandardized set of users. We believe that this model would produce better outputs if it were to be applied to a standardized dataset from a single company's systems. Companies can control their system settings by enforcing what OS is running, what version of that OS is allowed, and other variables related to the system. A more focused dataset would allow us to better understand the companies needs and tailor a model to their data.

References

Cohen, J. (2020, August 28). *Windows Computers Were Targets of 83% of All Malware Attacks in Q1 2020*. Retrieved from PCMag: https://www.pcmag.com/news/windows-computers-account-for-83-of-all-malware-attacks-in-q1-2020