Kaggle Microsoft Malware Prediction

Preston Pozderac
GitHub Link: www.github.com/Preston-Pozderac/BattelleInterviewChallengeCode

February 7, 2022

1 Introduction

As our society continues to rely more and more on technology, it is imperative that we protect our devices from malware infection. We combat this mainly through the use of anitvirus (AV) software and various implementations within a machine's software and hardware. In order to improve our current defenses, I will analyze a set of data collected from the reports of Microsoft's endpoint protection solution, Windows Defender. The full description of the data can be found at www.kaggle.com/c/microsoft-malware-prediction/data. The goal of my analysis is to use the dataset to predict whether the machine has detected malware on its system (from the variable 'HasDetections') and identify trends in the data that lead to malware infection.

2 Data Processing

Prior to generating my models, I first need to understand the data. The original data set is comprised of 83 variables (one is an indexing variable) with 8,921,483 data points. A quick inspection finds that nine of these variables are missing for over 90% of the data. These variables are immediately removed and all data points with additional missing entries are trimmed as well. This leaves 74 variables and 7,180,589 data points, roughly 80% of the original entries. Initial exploratory data analysis shows that a large proportion of the variables were categorical or binary data. I use one hot encoding to transform the data prior to applying my machine learning models. I also see that 'HasDetections', the prediction variable, has a roughly even 50/50 distribution between true and false. This means I do not have to apply any sampling methods and performance can be measured from the receiver operating characteristic (ROC) curve instead of a precision recall curve.

These variables can be split into three main categories: Antivirus or protection variables, computer variables, and location variables. The AV or protection variables cover the information behind the AV software, such as status and version numbers, and other protection options for computers, like a firewall, secure boot mode, or trusted platform module (TPM). These will be the primary focus of my investigation as they are naturally the first line of defense against malware. The second category of computer variables can be further broken down into software and hardware variables. A large subset of these refer to the operating system (OS) of the device as well as the various hardware components: processor, hard drive, and RAM. Finally, a few variables refer to the geographical or company location of the device.

For my models, I develop three subsets of the training data. First, I keep the majority of the data and remove a few variables with high cardinality that prevent my model from fitting in a timely manner. Physical insight is also used when removing these variables, i.e. the properties of the monitor are removed since technical knowledge would suggest that it has no impact on virus protection. The second data subset includes only the AV or protection variables. Finally, the second subset still contains high cardinality variables that prevent one of my models from fitting. I then aggregate the data by keeping all unique entries, within a variable, that represented 99% of the data, and put all the remaining entries under one new category. For example, the 'AVProductStateIdentifier' originally has $23,107$ unique state values in the data. However $1,320$ of those states represent 99% of the total data. The remaining states are all replaced by a new state -1 to clearly define the aggregated 1% of the data. These AV aggregated variables are my final subset.

3 Machine Learning Models

I apply three models to the data set: a baseline model, a decision tree classifier (DTC), and logistic regression. My baseline model is to always predict the true case for 'HasDetections' and serves as my benchmark for my other models. The training accuracy and area under the ROC curve (AROC) are 0.5069 and 0.5 respectively. I then train the data on a DTC with a max depth of ten and perform five fold cross validation. This model is able to be applied on all three of our data subsets. From the AV aggregate data, I find a training accuracy score of 0.5995 and AROC score of 0.5975. Notable improvement from the baseline, but it is likely these scores could be increased with additional optimization. The logistic regression is only able to be trained on the AV aggregate data. The training accuracy of this model is 0.6026 and had an AROC of 0.6006. A marginal improvement from the DTC, but also requires the longest computing time and least amount of variables.

While these models are incredibly powerful for predicting 'HasDetections', the main purpose I am using them for is feature importance. The DTC produces an importance score for each feature that measures which variables produced the largest reduction in the gini impurity (a measure of incorrectness in model predictions). When trained on the AV data (aggregated or not), the two main variables of importance are 'AVProductStateIdentifier', and 'EngineVersion'. These variables also have high significance when the DTC is trained on the majority of the data. This is further supported by the logistic regression model. The variables with the largest coefficients, both positive and negative, are those associated with the 'AVProductStateIdentifier'. Armed with this knowledge, I will dive deeper into this variable to look for more information and patterns in the data.

4 AV Product State

My models indicate a strong reliance on 'AVProductStateIdentifier' for predicting malware. The variable is defined as the "ID for the specific configuration of a user's antivirus software." Based on this definition, I thought that this variable can be used to determine high risk states and find trends in the other variables that lead to these high risk states. This means instead of having to look at the complicated relationship between 'HasDetections' and several variables simultaneously, I use the AV product state as an anchor point to simplify things.

I begin by looking for trends between the 'HasDetections' and AV product state, accounting for the frequency of each state. I see that over 66% of the data lies in one specific state, 53447, which is the most important in our DTC model. However it is unlikely that there is any numerical significance to changing states. Even though the state is represented by a number, the frequency of malware detections from state 100 to 101 could increase or decrease dramatically with no correlation. Therefore I start looking for trends in the other categorical variables. However the discrete nature of the data means linear trends are difficult to find. A clustering approach may be more appropriate to determine the values that produce the highest malware infection. Finally I look at specific state values to determine how they deviate from the entire data set. For example, AV product state 53447 has an above average detection of malware and a below average number of machines with secure boot enabled. This secure boot enabled could be a direct cause of the increased malware. More work needs to be done to analyze the differences in the variables between each state.

5 Future Investigations

My analysis is in its early stages and requires further developments in both the models and the exploratory data analysis. To improve the models, I would begin with applying ensemble learning to our DTC. First, I can perform a random forest classifier which can still output the importance score. The second would be an adaptive boosting method to train a series of weak learner decision trees. The exploratory data analysis must dive deeper into the computer software variables. These could have an impact with built in protections or inherent flaws that malware knows how to exploit. An extension of my work in AV product state comparison could be applied to the data more generally. Specifically, comparing the averages or modes of variables between high-risk ('HasDetections'=True $> 60\%$) and low-risk cases could demonstrate which variables correlate to increased malware detection.