Team 2: Data Mining

[FA23\_DATS6103](https://github.com/ning-rui/FA23_DATS6103/tree/main) : Summary Report

Professor Ning Rui

Dec 4, 2023

**Microsoft Malware Prediction**

**Introduction**

**Our proposed project aims to predict the probability of Windows machines becoming infected by malware by analysing telemetry data, including various machine properties and infection records, from Microsoft's endpoint protection solution, Windows Defender.** The goal of this competition is to predict a Windows machine’s probability of getting infected by various families of malware, based on different properties of that machine. The telemetry data containing these properties and the machine infections was generated by combining heartbeat and threat reports collected by Microsoft's endpoint protection solution, Windows Defender. The notebook is structured to process and analyze data for malware classification, specifically in the context of Microsoft Malware Prediction. It includes steps for data pre-processing, feature extraction, and initial exploratory data analysis. The focus is on understanding the characteristics of the malware samples through feature analysis and class distribution, which are critical steps in building a predictive model for malware detection.

The methodology employed for data collection in this study was tailored to align with specific operational limitations, including the safeguarding of user data privacy and the operational duration of the equipment involved. The challenge of malware detection, a task that fundamentally relies on chronological data, is further heightened due to factors such as the introduction of new devices, fluctuating device connectivity, application of updates, and installation of different operating systems. The dataset presented here is divided based on timeframes, but due to the aforementioned complexities and data collection constraints, there might be inconsistencies observed in the alignment of your cross-validation, public, and private scores. It's also important to note that this dataset does not accurately reflect the typical distribution of malware on Microsoft customers' devices; it intentionally contains a higher ratio of malware-infected machines for study purposes.

Where we can predict a high-level overview based on the initial cells of the notebook, each cell's code and output must be analysed for a more comprehensive report.

**SMART Questions**

With the intention of focusing the project to relevant aspects of our cyber threats, we focused on recent data for properties in Malware detection in Microsoft. In particular, we inquired into to two aspects of byte files and asm files and tested various aspects ration with them:

1. What are the key features in the dataset that can be used to predict malware infections?
2. How does this project's outcome contribute to enhancing cybersecurity efforts?
3. Are asm files and. bytes file enough for malware detection?
4. Dataset has imbalance, can we handle it through hyper-parameter tuning, which model will stand out?

**Literature Review**

From the available information, it's known that the Microsoft Malware Classification Challenge, related to this Kaggle competition, was launched in 2015 with a significant dataset of nearly 0.5 terabytes, including disassembly and bytecode of over 20,000 malware samples. This dataset has become a standard benchmark in malware behaviour modelling research and has been cited in more than 50 research papers. These studies likely explore various aspects of malware detection, machine learning models for prediction, and the challenges in classifying and predicting malware threats in diverse computing environments [18†source].

**Description Of Data**

The dataset is comprised of a multitude of rows, each representing a unique machine, identified by its "Machine Identifier." The "Has Detections" column provides the ground truth, indicating whether malware was detected on the machine. The data is sourced from both heartbeat and threat reports collected by Windows Defender. The nature of this data offers a rich source of information for the development of our predictive model. There are two key data files, `asmoutputfile.csv` and `byteoutputfile.csv`, are loaded. These files likely contain features extracted from malware samples.

**Asm file:** The file contains 10,868 entries and 52 columns It includes an 'ID' column followed by various feature columns representing counts or metrics extracted from assembly language files of the software. These features include counts of different sections (.text, .data, etc.), assembly instructions (like jmp, mov, retf), and register usages (eax, ebx, ecx, etc.). The first few rows show varying counts across different features, indicating diverse characteristics in the assembly files of different software.

**Byte file:** This file also has a wide format with 258 columns, including an 'ID' column and 257 feature columns. The features are likely byte code frequencies or byte-level components extracted from the software files. Each column, named with hexadecimal values (0, 1, 2, ..., ff), represents a different byte code. The data in the first few rows show a wide range of values across other byte codes, suggesting significant variability in the byte-level characteristics of the software samples

**Analysis**

Based on the analysis of the Jupyter Notebook focused on Microsoft Malware Prediction, The notebook starts with importing necessary libraries and suppressing warnings. This setup is crucial for data handling and visualization.

Both files provide a rich set of features, capturing different aspects of the software's structure and content at the assembly and byte code levels. The variability in feature values across different samples indicates a diverse dataset, which is beneficial for training robust malware prediction models. The high dimensionality of the data, especially in `byteoutputfile.csv,` might pose challenges regarding computational resources and model complexity.

These datasets are a comprehensive basis for building and training machine learning models for malware prediction. The variety and depth of the features are likely to contribute significantly to the accuracy of such models. However, careful feature selection and pre-processing might be required to manage the complexity and optimize the performance.

The notebook prints the count of unigram byte features (257) and unigram asm features (51). This indicates a focus on feature extraction from assembly and byte files, typical in malware analysis. A visualization is generated to show the distribution of these features. This is important for understanding the dataset's structure and the diversity of malware samples.

**Training of Data**

We reviewed the contents of pre-processing, where The notebook appears to be focused on preprocessing a dataset, which is a common step in data analysis and machine learning projects. The goal is to clean and transform raw data into a format suitable for analysis or modeling. It begins with loading a dataset. This could be from a file (like CSV, Excel), a database, or an API. The specific source in your notebook was not mentioned. The notebook includes steps for data cleaning. This typically involves handling missing values, removing or imputing nulls, and correcting any inconsistencies in the data. There's evidence of feature engineering, which is the process of creating new features or modifying existing ones to improve the performance of machine learning models. The notebook demonstrates data transformation techniques. This might include normalization, standardization, or encoding categorical variables into a format that algorithms can work with. Some sections are dedicated to EDA, where the data is analyzed using various statistical methods and visualization techniques to uncover patterns, spot anomalies, test hypotheses, or check assumptions.

We have interspersed with comments and documentation, which explain the rationale behind each step and make the notebook easier to understand for others. The notebook likely ends with a transformed dataset, ready for further analysis or as input to a machine learning model.

The analysis serves as a comprehensive guide for preparing a dataset for analysis or modeling. It covers essential steps from data cleaning to feature engineering, ensuring the data is in an optimal state for subsequent processes. The inclusion of EDA also suggests a thorough approach to understanding the dataset before any advanced analysis or model building.