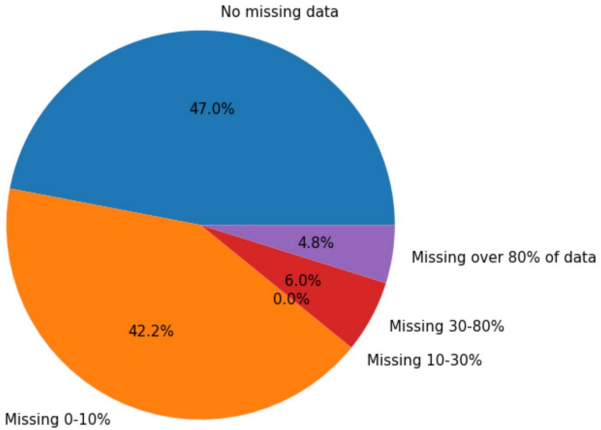
Malware Prediction System

Group 28
Akarsh Rastogi, Anqi Wang, Selim Shaalan,
Tseng-Han Yu, Ziqi Li

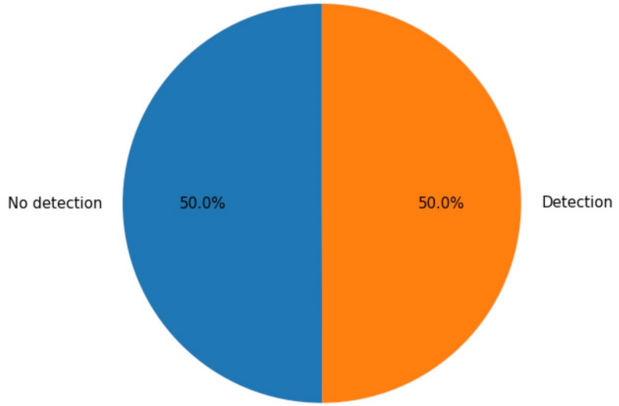
Missing Data Analysis and Target Variable Balance



In the dataset, there are :

- 39 columns without missing values
- 35 columns with less than 10% of missing values
- 0 columns with missing values between 10% and 30%
- 5 columns with between 30% and 80% of missing values
- 4 columns with more than 80% of missing values

We shall remove the 9 columns for which more than 30% data is missing.



The target variable, which is detections of Malware, is exceptionally well balanced.

There are 4462591 cases of no detections , and 4458892 cases of malware detections.

Data Cleaning

We'll start off by removing the 9 columns that have 30%+ missingness, which are mostly census data.

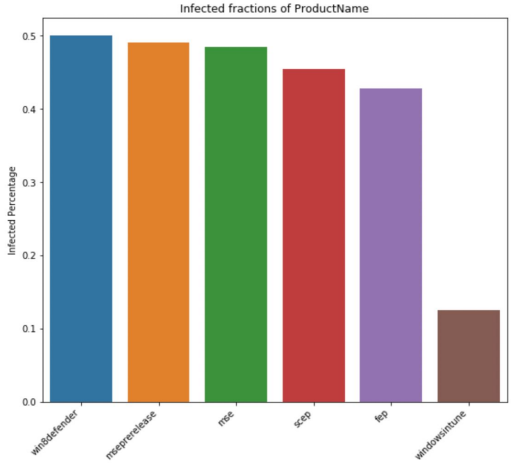
This leaves 34 other columns with missingness.

- For the categorical columns, we will use KNN imputation or mode based imputation
  - Since the data is in somewhat chronological order, we can impute using a subset of rows or columns to compute KNN (using the entire dataset is too computationally expensive)
  - We could just impute with the mode if the KNN proves too time consuming given the massive dataset.
- For the numerical columns, we will impute the missing values with the mean.

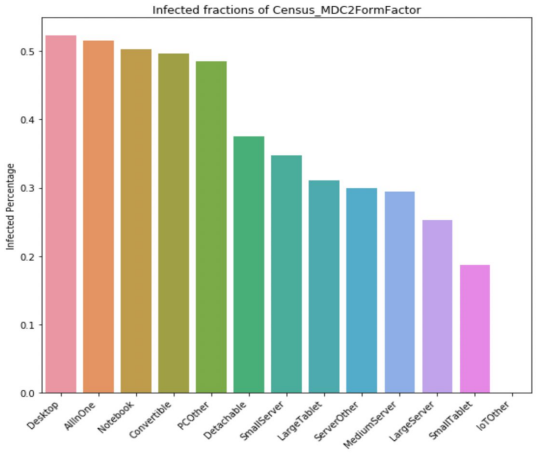
Binary variables and identifiers such as isBeta and countryIdentifier were stored as integers/floats, but we converted them all to categories and used One-hot-encoding to pre-process them.

All columns were checked for outliers and none were found.

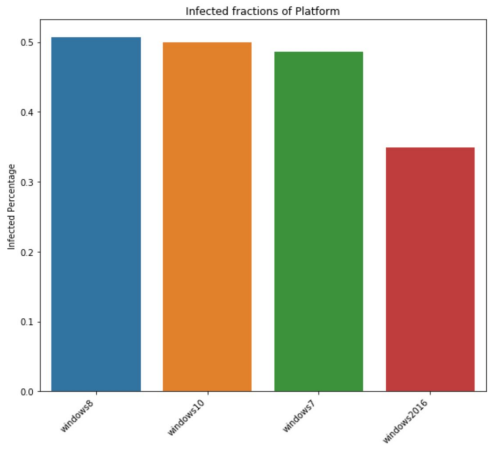
Malware Infections : Analysis of Software, Platform, and Form



There is a fairly distributed share of the Defense Software installed on the devices that were infected. Windowsintune was notably less affected by malware.



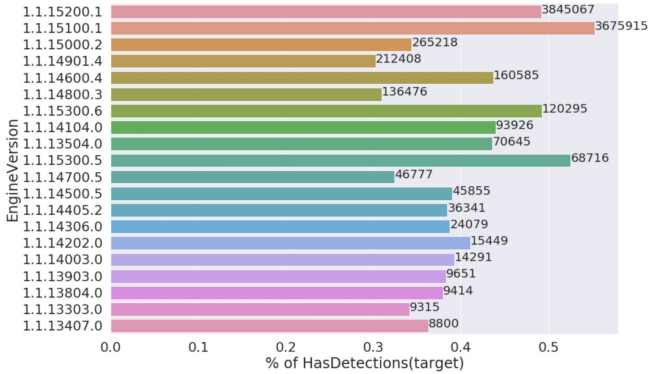
In terms of Form factor, Desktops were highly affected by malware, followed by Notebooks and Convertibles. Smaller devices and servers were relatively less affected.

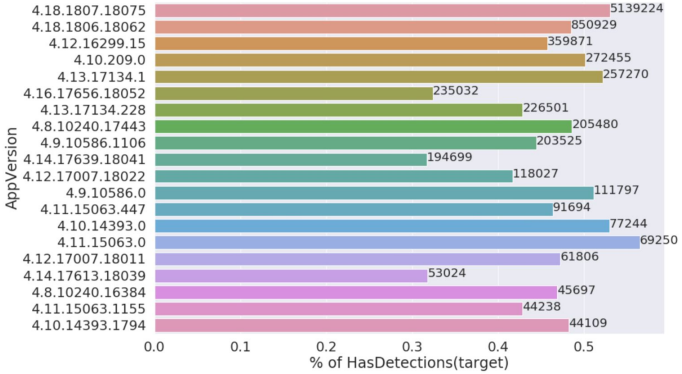


All platforms except Windows 2016 were affected almost equally.

Malware Infections : Analysis of Engine and App Version

As relationships between versions of Engine and App with target variable would have meaningful and straightforward interpretation, we decided to go forward and explore their relationships.

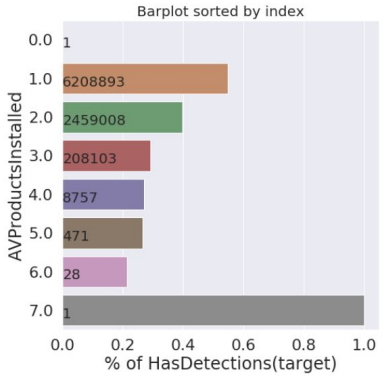




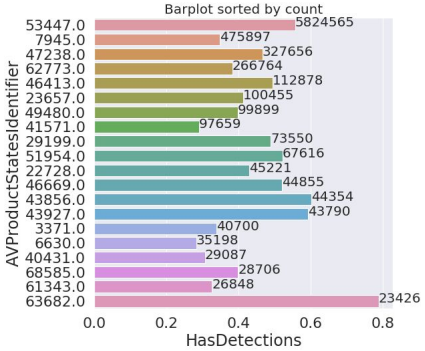
The vertical axis lists engine versions. The horizontal axis shows the percentage of malware detections, while numbers beside the bars represent the total count for each version. From the graph, we can observe newer versions seem to have a higher percentage of detections.

The vertical axis lists app versions. The horizontal axis shows the percentage of malware detections, while numbers beside the bars represent the total count for each version. From the graph, we can observe there is no obvious trend between app versions and percentage of detections.

Malware Infections : Analysis of AVProductsInstalled and AVProductStatesIdentifier



We also decided to explore relationships between AVProductsInstalled and AVProductStatesIdentifier with target variable as antivirus targets malware.The bar plot presents the proportion of positive malware detections across a range of different counts of antivirus products installed on machines. As the number of antivirus products installed increases from 1 to 3, there is a noticeable decrease in the proportion of detections. Notably, machines with a single antivirus product installed show the highest proportion of detections, which decreases significantly with two products and gradually with three, while systems with more than three products have very few detections.



The bar plot illustrates the proportion of positive malware detections (HasDetections) associated with various antivirus product identifiers. The bars are sorted by the count of occurrences for each ID associated with antivirus software, with ID '53447.0' having the most detections and the state '63682.0' having the least. The plot indicates that identifier '53447.0' has the highest overall count but does not necessarily correspond to the highest proportion of detections.

Machine Learning Techniques

Logistic Regression:

We decided to implement multivariate binary logistic regression as a benchmark model. It can provide a baseline to which we can compare more complex models.

Pros:

- Fast to train
- Easy to interpret

Cons:

- Assumption of linearity
- Prone to overfitting

Machine Learning Techniques

Random Forest:

Since there is a large amount of features, we believed RF classification could be used to perform innate feature selection. Moreover, we can tune the model to further improve its performance.

Pros:

- High performance
- Flexibility

Cons:

- Large model size
- Computationally intensive

Machine Learning Techniques

Gradient Boosting:

This technique may provide higher predictive accuracy than RF model, but it is more sensitive to noise. We would like to experiment with its efficacy in the future.

Neural Networks:

This helps us learn higher-order information from the data. Since we have ample data in the dataset, deep learning is a worthy experiment to compare performance.