Northeastern University

ASSIGNMENT FRONT SHEET

Course Name: **ALY6015 20904 Intermediate Analytics**

Professor Name: **ChuanLi Jiang**

Student Class: **Fall 2019 CPS** **Term: A. 2020**

Final Project Report

Completion Date: February 13th **Due Time:12:00am**

Statement of Authorship

I confirm that this work is my own. Additionally, I confirm that no part of this coursework, except where clearly quoted and referenced, has been copied from material belonging to any other person e.g. from a book, handout, another student. I am aware that it is a breach of Northeastern University's regulations to copy the work of another without clear acknowledgement and that attempting to do so renders me liable to disciplinary procedures. To this effect, I have uploaded my work onto Turnitin and have ensured that I have made any relevant corrections to my work prior to submission.

☒ **Tick here** to confirm that your paper version is identical to the version submitted through Turnitin

Final Project Report: Microsoft Malware Detections

Group Member & Student ID:

Jingyan Qiao, 001368423

Jiayi Wang, 001082307

Quoc Tuong Dong, 001089611

Ye Chen, 001300776

Introduction:

With the development of information industry and related technologies, computer and other machines become increasingly important in production of any industry, keeping machine in a safe condition and not getting infected by any type of malware is a top challenge for security department. The malware industry continues to be a well-organized, well-funded market dedicated to evading traditional security measures. Once a computer is infected by malware, criminals can hurt consumers and enterprises in many ways. To protect the enterprise and consumer customers' computer and reduce the risk of personal data leaking in advance, Microsoft takes this problem seriously and the factors of getting a machine infected was investigated in order to improve their security system. The dataset conducted from Microsoft contains various properties of each machine and the actual infection status of each machine generated by Microsoft's endpoint protection solution, Windows Defender, which was designed to meet certain business constraints, both in regards to user privacy as well as the time period during which the machine was running.

Logistic regression model is a classic machine learning model classification which makes it an appropriate regression analysis to conduct when the dependent variable is dichotomous (binary). Like all regression analyses, the logistic regression is a predictive analysis which is used to describe data and to explain the relationship between one dependent binary variable and one or more independent variables. The dependent variable (response) in the

given dataset is whether the malware was detected on the machine or not, therefore the logistic regression model is the fundamental model in this analysis. When pursuing the higher accuracy of the prediction for high-dimensional datasets, the trade-off between bias and variance appears all the time. The given dataset contains 83 columns and over 50k rows. Regularization regression is a method to avoid the risk of overfitting and reduce the variance of the models without substantial increase in bias when fitting a regression model on a high-dimensional dataset. Least Absolute Shrinkage and Selection Operator (LASSO) Regularization method is suitable for models with multiple highly correlated predictors to remove the redundant features in order to facilitate model interpretation to make decisions.

Decision tree is a model used to classify observations by splitting data based on the features, but it has weaknesses especially in big data when researchers are facing with various types of features or a large amount of data. The complexity of the decision trees severely reduces the efficiency and accuracy of the prediction model. Random Forest is a machine learning method in classification analysis which consists of a large number of individual decisions trees, functioning as an ensemble. Researchers harvest class predictions from each of the tree and decide which is the most important class base on the most votes method. To optimize predictions, researchers usually use bootstrapping methods. Another machine learning algorithm method well-suited for the classification field based on decision trees is Gradient Boosting Decision Trees (GBDT) model. It is an ensembled model using boosting technique, essentially combining weak classifiers to form a stronger one. Specifically in this case, the weak classifiers are different decision trees, therefore the result of GBDT model could be considered as a combination of these decision trees. The boosting process is sequential, which connects the subsequent trees to the previous ones with errors in the predictions generated in the past in order to reduce the prediction error gradually during the combination process. Besides, the GBDT model will not consider the result from one single tree as the final result. Thus, it can be used to solve the overfitting problem as well.

The goal of this study is to predict the probability of the machine being infected by malware, and determine the most significant factors affecting the infections for Microsoft Developers to take actions to improve their systems in the future. Also, it is an opportunity to gain a better understanding of the theory and performance of

various classification methods such as simple logistic regression, logistic regression model with Lasso Regularization, Random forest and Gradient Boosting Decision Trees method, and explore more on the relationship between large amount of predictors and categorical response variable, and further make prediction based on the findings.

Data & Methodology:

Data description:

The data was retrieved from Microsoft Competition in Kaggle and the data cleaning process was executed. The raw dataset contains 8,921,483 observations with 82 various features of the machine and one response variable, that is whether the machine was infected by the malware (Having Infections: coded as 1, Not Having Infections: coded as 0). After investigating the frequency of the missing values and the distribution of dimensions for each feature, those features with higher level of missing values (percentage of the missing values over 75%) or highly imbalanced dimensions were deleted. For instance, none of the machines appears to have a Beta software, therefore the feature *IsBeta* (whether the Beta software appears on each machine) was deleted and 50 variables were studied in this analysis.

The features assumed to be significant in this study were:

EngineVersion: the version of the engine.

AvSigVersion: the version of the anti-virus signature database.

AvProductStateIdentifier: the status of the anti-virus software used on the machine.

Wdft_IsGamer: whether the machine was a gamer machine.

CityIdentifier: where the machine was located in.

CountryIdentifier: the country the machine was located in.

OsVer: the version of the current operating system.

OsPlatform: the platform used for the operating system.

IsProtected: whether there is any anti-virus products on the machine.

Firewall: whether the firewall was built on the machine.

HasTpm: whether the machine has a trusted platform module.

SmartScreen: smart screen enabled on the machine.

Census_SystemVolumeTotalCapacity: the size of the system volume installed on the machine.

Census_OsBuildRevision: the latest version of the operating system on the machine.

Platform: the platform installed on the machine.

Census_TotalPhysicalRAM: the number of the Random-Access Memory.

Census_HasOpticalDiskDrive: whether the machine has an optical disk drive.

Census_OsEdition: the edition of the current operating system.

Census_IsTouchEnabled: whether the machine is a touch device.

Methodology:

- **Logistic Regression Model.**
  a. Summarize the logistic model and select the predictors with significant contribution based on the p value at a 95% confidence interval.
  b. Plot the predicted probability of the dependent variable.
- **LASSO Regularization Method: Logistic Regression Model.**
  a. Plot the cross validation curve with all selected predictors.
  b. Determine the value of Lambda that minimize the mean cross validation error and the value of lambda within one standard error of the minimal mean-squared error to identify the optimal logistic regression model with lasso regularization method.
- **Random Forest Model.**
  a. Encode the target features as factors.
  b. Split the dataset into 6 tree groups and split each group into training and testing sets.

c. Fit the random forest classification to the training set for 6 trees. X is the dependent variables, y is the results, *ntree* indicates the number of trees to grow resulting with 500 trees.

d. Visualize the results with prediction in binary format, generate the matrix and compare 6 different groups of data to determine the optimal decision tree.

- **Gradient Boosting Decision Trees (GBDT) Model.**

a. Labels Encoding: Label Encoding transform values to become numbers between 0 and *n-1*, where n is the number of different labels.

b. Frequency Encoding: Frequency Encoding is a special case of Label Encoding. Feature values are encoded based on the frequency. Transformed values are numbers between 0 and *m*, where *m* is the number of values with a frequency greater or equal than 2.

Analysis & Discussion:

Logistic Regression Analysis:

The predicted formula can be constructed at a 95% confidence interval by the logistic regression to multiple predictors as follows:

Predicted logit of (HasDetections) = 0.7084 *Platformwindows8 + 0.9071*SkuEditionEduation
+ 0.9809*SkuEditionEnterprise + 1.366*SkuEditionHome
+ 2.108*SkuEditionInvalid + 0.933*SkuEditionPro + 0.3086***IsProtected**
+ 0.1395***Wdft_IsGamer** + 0.7202*AppVersion. **(1)**

According to Equation 1, the log of the odds of the machine having infections was related to 9 features (p value < 0.05). All features have a coefficient greater than 0, which suggests that these features enhance the probability of the machine getting infected. The feature *IsProtected* (whether each machine has any active anti-virus products) and *Wdft_IsGamer* (whether the user of each machine is a gamer) have the lowest p value, therefore these two features are most critical factors increasing the probability of the machine having infections.

The goodness-of-fit statistics assess the fit of a logistic model against actual outcomes (i.e., whether a machine has infections). Adjusted R-squared value is 0.028, suggesting around 2.8% of the observations can be explained by the model. The large AIC value (80695) and Residuals (59748) also show the inefficacy of the model.

Table 1: Confusion Matrix (Logistic Regression Model).

Logistic		**Predicted**		
		0	**1**	**Total**
Actual	**0**	15144	14388	29532
	1	10688	19586	30274
Success Rate		**58.07%**		

**1: the machine that was infected by the malware.*
**0: the machine that was not infected by the malware.*

A confusion matrix was adopted to evaluate the performance of the logistic regression model. Table 1 suggests that the around half of the uninfected machine was predicted to be uninfected and about two-third of those infected was correctly classified by the logistic regression model. The overall success rate of the prediction was 58.07%. The Type I error (where the uninfected machines were falsely predicted) is relatively higher than Type II error (where the infected machines were falsely classified as uninfected) which defends the model in some ways, since it is riskier to classify an infected machine as uninfected than the other way around.

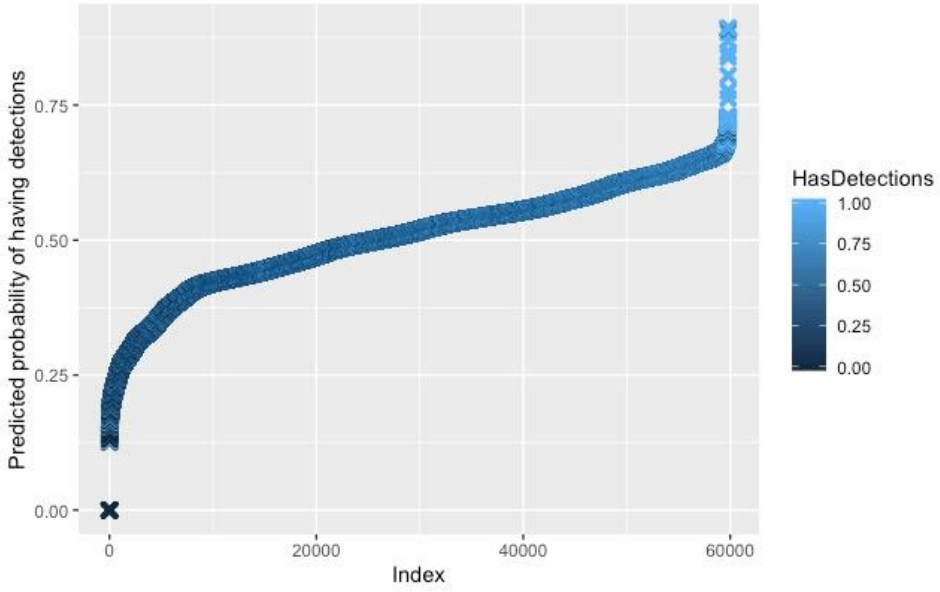


***Figure 1**: Predicted Probability of the Outcome variable, HasDetections (1 = Having Infections, 0 = Not Having Infections).*

The probability prediction plot shows the predicted probability of each machine having infected by malware along with the actual malware detection status. The Y-axis is the probability of having detections, and the X-axis is the probability of infection from the lowest to the highest value of 60,000 machines. The curve is relatively flat around the probability of 0.5, indicating that the prediction of the model is not efficient, which is consistent to the small R-squared value above.

Logistic Regression Analysis with LASSO Regularization Method:

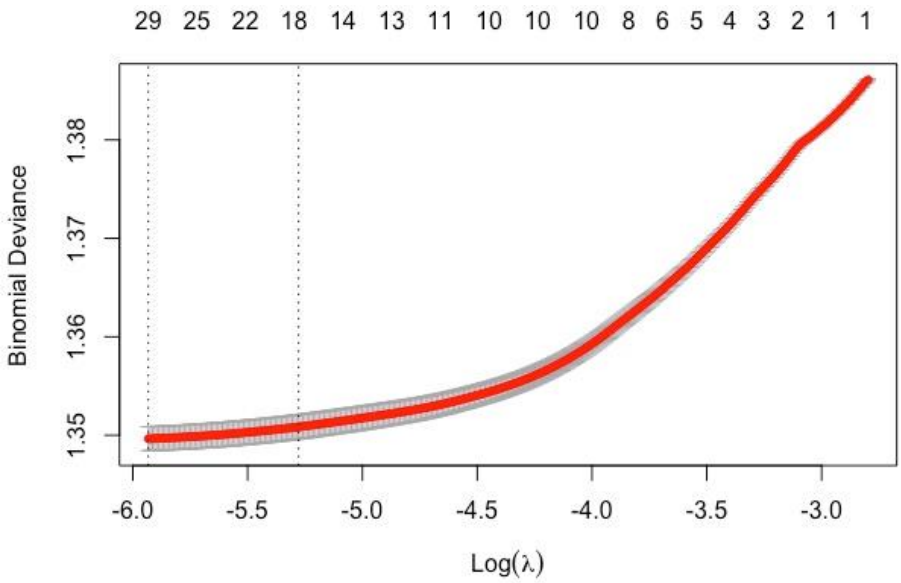


***Figure 2**: Mean Squared Error Versus the Logit of (lambda).*

From the Plot 3, the value of the mean squared error has increased as lambda value increases, and first dash lines chose two lambda values for the regularization method. The smaller lambda leads to the minimum mean squared error, and the lambda within one standard deviation that leads to the model with fewest features. The Lasso regularization process was applied with a lambda of 0.00265 to remove the redundant variables. The predicted formula was adjusted by adapting Lasso regularization method to the logistic regression model as follows:

Predicted logit of (HasDetections) = 0.1017 *Platformwindows8 + 0.0843*IsProtected

+ 0.0169*Wdft_IsGamer **(2)**

Several features were shrunken to zero from Equation 1 to Equation 2. It was noticed that only three features show significance after Lasso regularization: the *Platformwindows8* (the Windows 8 platform), *IsProtected* (whether any anti-virus products was on each machine) and *Wdft_IsGamer* (whether the user of each machine is a gamer). All these three features increase the risk of the machine getting infected by malware with a positive coefficient.

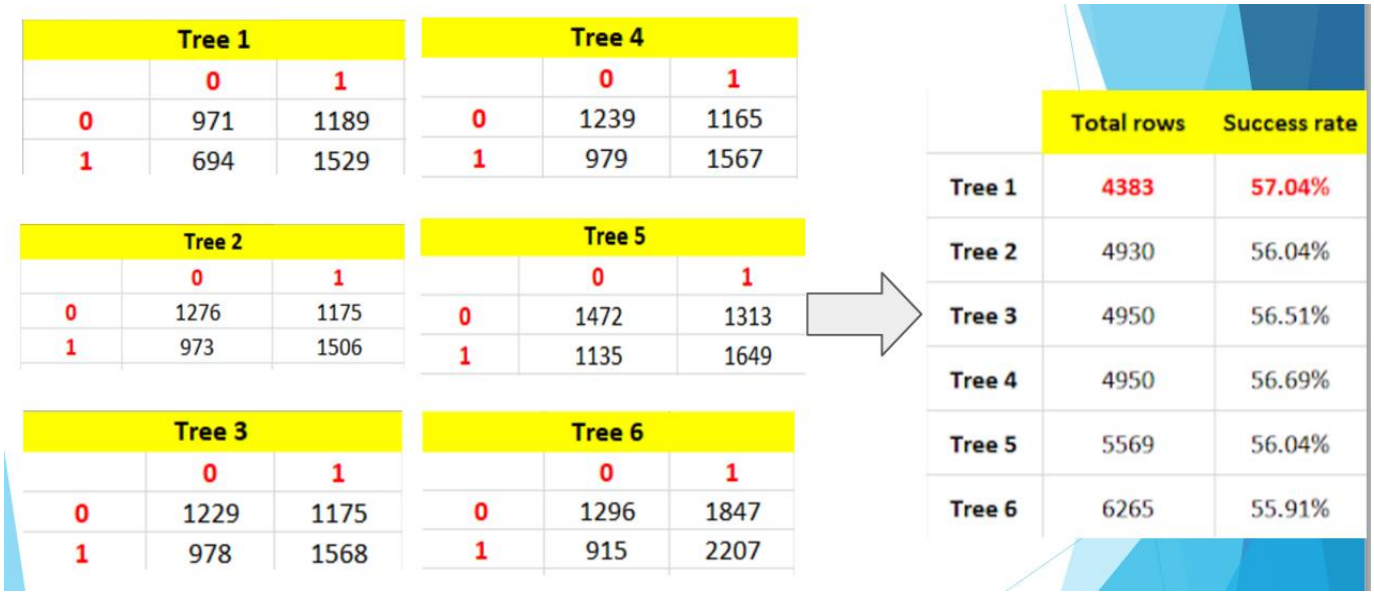
Table 2: Confusion Matrix (Logistic Regression Model with Lasso Regularization method).

LASSO		**Predicted**		
		0	**1**	**Total**
Actual	**0**	4411	4446	8857
	1	3082	6061	9143
Success Rate		**58.18%**		

**1: the machine that was infected by the malware.*
**0: the machine that was not infected by the malware.*

After performing the Lasso regularization method, the accuracy rate was improved by around 0.1% compared with the result from the simple logistic regression model shown in table 1. Whereas, the maximal R-squared value was generated to be around 0.039 which indicates the inefficacy of the model.

Random Forest Model:



Tree 1		
	0	1
0	971	1189
1	694	1529

Tree 4		
	0	1
0	1239	1165
1	979	1567

Tree 2		
	0	1
0	1276	1175
1	973	1506

Tree 5		
	0	1
0	1472	1313
1	1135	1649

Tree 3		
	0	1
0	1229	1175
1	978	1568

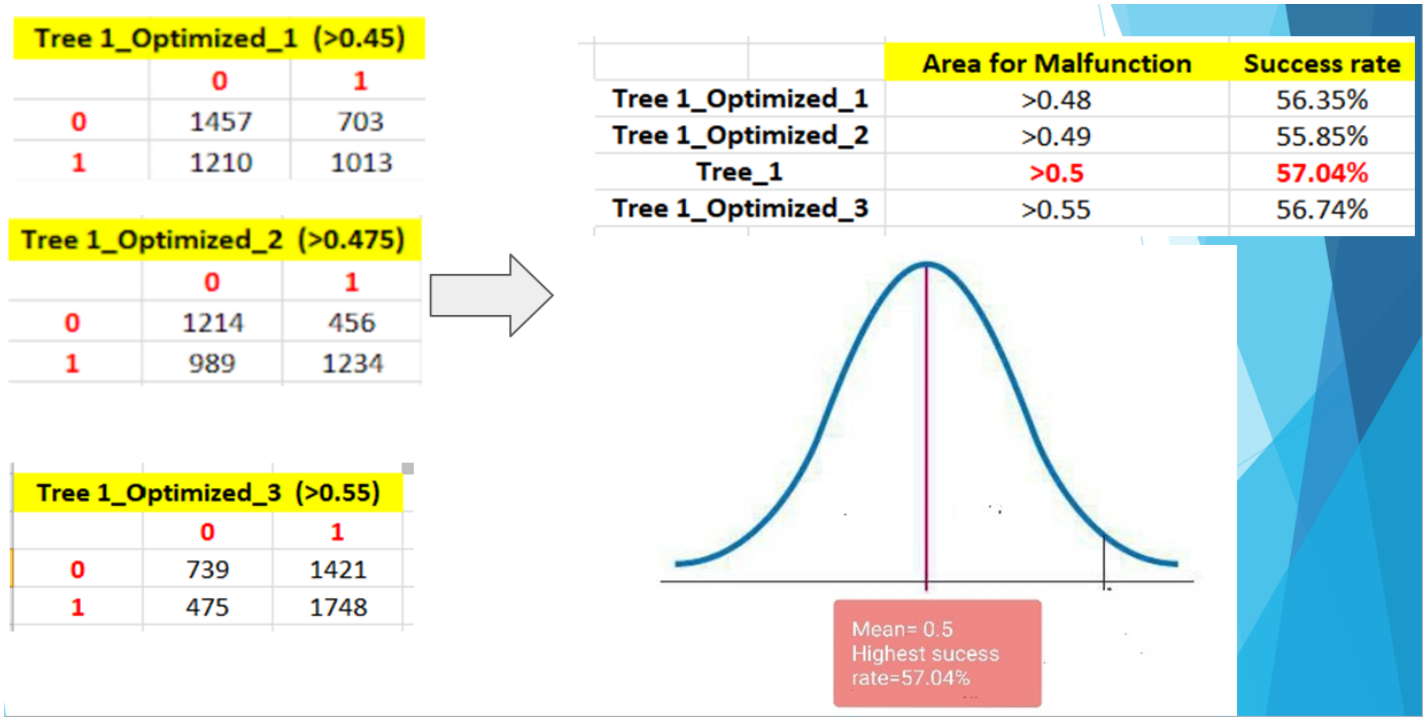
Tree 6		
	0	1
0	1296	1847
1	915	2207

	Total rows	Success rate
Tree 1	4383	57.04%
Tree 2	4930	56.04%
Tree 3	4950	56.51%
Tree 4	4950	56.69%
Tree 5	5569	56.04%
Tree 6	6265	55.91%

Note: 0, 1 on the left columns of the tables in Figure 3 represent the actual detection status in the test set and the upper ones represent the predicted result.

***Figure 3**: Confusion Matrices (Random Forest Model).*

In the Random Forest, the dataset was split up into 6 trees with slightly different number of inputs (500-1000 rows) in each tree to test whether size of the inputs actually affects the prediction rate. From Figure 3, it can be observed that Tree 1 with the lowest number of rows (4883 inputs) has the highest success rates at 57.04%. The higher the number of observations is, the lower the success rate is with Tree 6 (6265 inputs) which can only predicted 55.91% of the test set successfully. As a result, the number of Type I error cases (wrongly classified the machine as infected one when it is actually not) are greater than Type II error (wrongly classified the infected machine) cases across all the trees, which reduces the severity of wrong predictions.



Note: 0, 1 on the left columns of the tables in Figure 4 represent the actual detection status in the test set and the upper ones represent the predicted result.

***Figure 4:** Confusion Matrices with Various Thresholds.*

The confidence level was set lower to 0.45 and 0.475 and also it was set to be 0.55 in order to compare the result of the success rate. Using Tree 1 as the base for the optimization, success rates fluctuates as the Confidence

level of area for malfunction changes. Nevertheless, the most optimized success rate (highest) is still the original Tree 1 with 57.04% success rate. Any confidence level above or below 0.5 decreases the success rate of the Random Forest prediction model. Thus, the confidence level was normally distributed between 0 and 1 with a mean of 0.5. However, the Type II error was reduced from the original Tree 1 (15.83%) to the Optimized 3 of Tree 1 (10.84%) with a threshold of 0.55. Therefore, the Optimized 3 case with confidence level at 0.55 would be considered as a more useful decision tree with a 0.3% success rate reduction.

Gradient Boosting Decision Trees (GBDT) Model:

The dataset was randomly split into train set (70%) and test set (30%), with each set containing approximately equal number of infected machines and uninfected machines. The train set was used to build the model so that the probability of test machine getting infected can be predicted (1 = Having Infections, 0 = Not Having Infections). A probability larger than 50% means the machine was predicted to be infected, and not infected with a probability lower than 50%.

Table 3: Confusion matrix (GBDT Model with 60,000 observations).

GBDT		**Predicted**		
		0	**1**	**Total**
Actual	**0**	4555	4286	8841
	1	3609	5356	8965
Success Rate		**55.66%**		

**1: the machine that was infected by the malware.*
**0: the machine that was not infected by the malware.*

Table 4: Precision, Recall Rate and F1 Score of GBDT Model (with 60,000 observations).

	Precision	**Recall**	**F1**
0	51.52%	55.79%	53.57%
1	59.73%	55.55%	57.56%

**1: the machine that was infected by the malware.*
**0: the machine that was not infected by the malware.*

According to Table 3, 4555 out of 8841uninfected machine were correctly classified and, 3609 out of 8965 infected machine were correctly classified, the predict accuracy was 55.66%. The precision and recall rate measure the performance of the prediction model with higher proportion representing better performance and higher accuracy rate. Theoretically, the modification of the model aims to increase both the precision and recall rate, but increasing the precision rate always leads to a lower recall rate, and vice versa. Thus, the combination of precision and recall rate, F1 score, was embraced to give a better evaluation on the model. From Table 4, the F1 score is less than 60% in each group of infected or uninfected machines, which indicates the model poorly explained the observations.

Table 5: Confusion Matrix (GBDT Model with 100,000 observations).

GBDT		**Predicted**		
		0	**1**	**Total**
Actual	**0**	8685	6290	14975
	1	6508	8517	15025
Success Rate		**57.34%**		

**1: the machine that was infected by the malware.*
**0: the machine that was not infected by the malware.*

Table 6: Precision, Recall Rate and F1 Score of GBDT Model (with 100,000 observations).

	Precision	**Recall**	**F1**
0	58.00%	57.16%	57.58%
1	56.69%	57.52%	57.10%

**1: the machine that was infected by the malware.*
**0: the machine that was not infected by the malware.*

The GBDT model was adopted on a larger dataset with 100,000 observations and the accuracy rate was improved by approximately **2.68%** shown in table 5. Also, F1 Score was higher than 55% for both infected and uninfected cases which confirms the better efficacy of the model with larger amount of data.

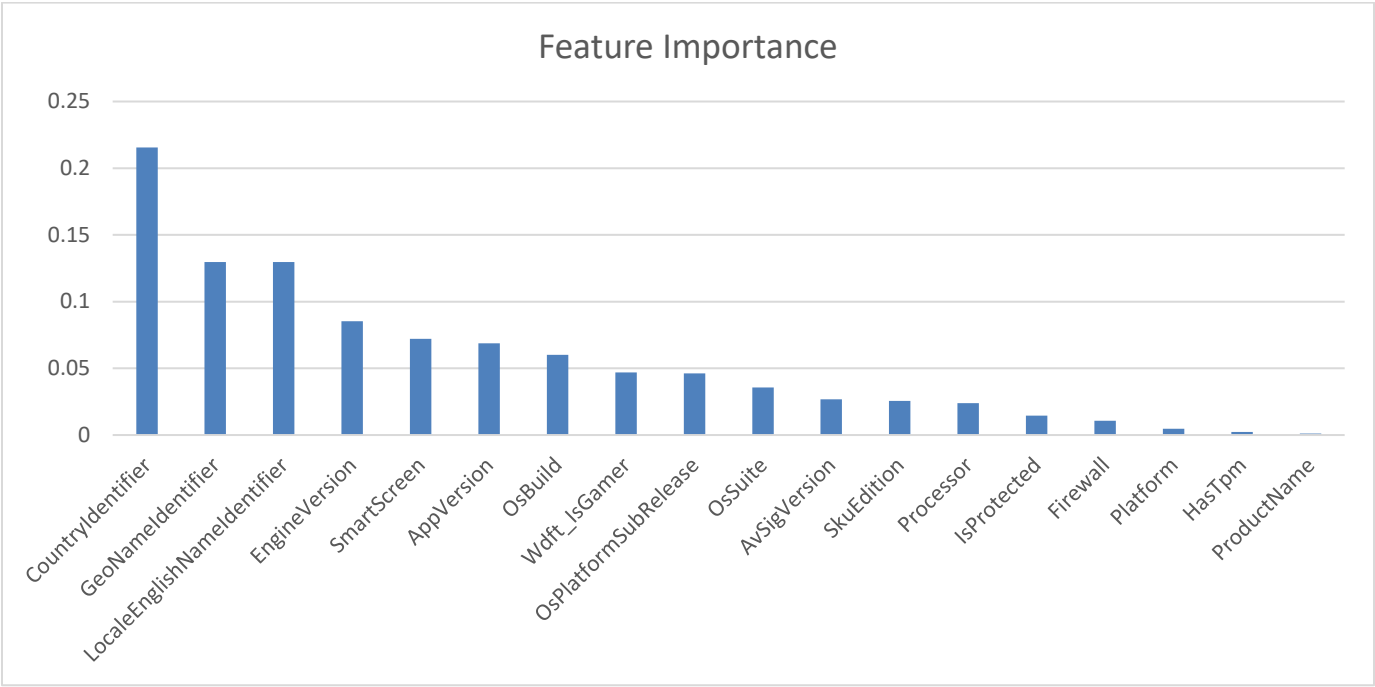


***Figure 5**: The importance of the features shown significant contribution (with 60,000 observations).*

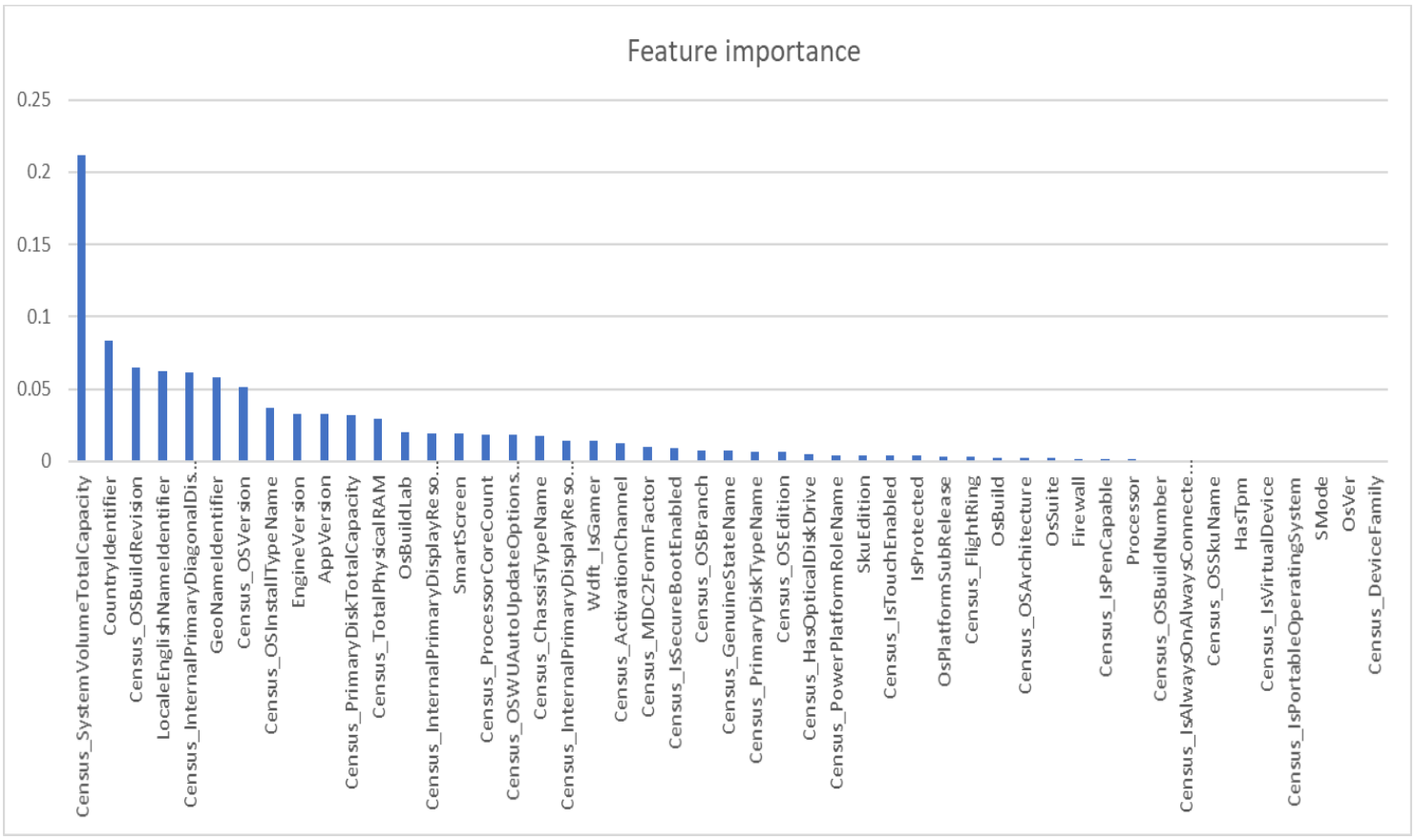


***Figure 6**: The importance of the features shown significant contribution (with 100,000 observations).*

The Gradient Boosting Decision Trees (GBDT) model generated the frequency of each feature used in the model fitting process and a feature with a higher frequency was considered to have higher importance. The bar charts were drawn for two GBDT models to visualize the impacts of those features on the probability of the machine getting infected. The top three features with highest importance from Figure 5 were *CountryIdentifier* (the country each machine was located in), *LocaleEnglishNameIdentifier* (the language code of the machine defined by Microsoft, i.e. English = 1033.) and *GeoNameIdentifier* (the geographic region the machine was located in) which all related to the location of the machine. *Census_SystemVolumeTotalCapacity* (the size of the system volume installed on the machine), *CountryIdentifier* (the country each machine was located in) and *Census_OsBuildRevision* (the latest version of the operating system on the machine, i.e. the latest version of Windows 10 =1909) appear to be the top three features with higher contribution. The importance of each feature has changed from Figure 5 to Figure 6 with different amount of observations. Whereas, there is no enough evidence to claim that those top three features in Figure 6 would give a precise prediction with a relatively low success rate.

Conclusion & Recommendations:

From the results discussed above, it can be concluded that the Lasso logistic regression model is the best model of prediction for this dataset with the highest success rate of 58.18% while the random forest which believed to be one of the best models for classification machine learning, has the lowest success rate among all three models both before and after optimization. Also, the Gradient Boosting Decision Trees (GBDT) model was not a proper approach to predict the probability of each machine getting infected with a smaller accuracy rate (57.34%). R programming language was found to have limited performance handling with variables containing over 1000 unique values. In addition, the same process was done in Python for the GBDT model with more high-dimensional features which results in a higher accuracy rate of 62.02%.

Error probability during extraction. The observations used in this study was only around 6.7% of the whole dataset due to the limitation of the computers and R programming language, which makes the results hardly

representative to the entire dataset. A larger sample was selected to optimize the GBDT model and it did improve the accuracy rate by 2.68%, which confirms the significance of the sample size. Although we extracted the dataset randomly with equal amounts of the infected and uninfected machines, there might still be other factors might explain the final results, for instance, some properties of those machine might be recorded in a specific order.

Lack of transparency, confidentiality and unexplainable features. Out of 83 original columns, only around half of them are chosen for this analysis. It is due to the fact that a lot of features are not clearly explained in detail, encoded or even missing for all the observations. One of the examples is the Geographic features. All of the countries are encoded in a confidential way and the probability of the machine getting infected by the malware in each country cannot be studied as a result. Those Geographic features were excluded from the Random Forest prediction because they have more than 200 unique values for each variable and *RandomForest* function can only assign 53.

For further study, researchers should execute these models using Python on computers with generous memory to utilize the whole dataset and focus on the features appearing to be significant in this analysis. It could be an alternative way to study those machines located in the same place or fixed one of other features with gargantuan number of categories to gain a more precision prediction result and different sets of observations could be selected to verify the significance of each property of the machines.

References

1. Bronshtein, A. (2019, February 27). Train/Test Split and Cross Validation in Python. Retrieved from https://towardsdatascience.com/train-test-split-and-cross-validation-in-python-80b61beca4b6

2. Computing Classification Evaluation Metrics in R. (n.d.). Retrieved from https://blog.revolutionanalytics.com/2016/03/com_class_eval_metrics_r.html

3. Microsoft Malware Prediction. (n.d.). Retrieved from https://www.kaggle.com/c/microsoft-malware-prediction/data

4. Microsoft Malware Prediction. (n.d.). Retrieved from https://www.kaggle.com/c/microsoft-malware-prediction/discussion/84065

5. Openspecs-Office. (n.d.). [MS-OE376]: Part 4 Section 7.6.2.39, LCID (Locale ID). Retrieved from https://docs.microsoft.com/en-us/openspecs/office_standards/ms-oe376/6c085406-a698-4e12-9d4d-c3b0ee3dbc4a

6. Person. (2019, January 8). Rstudio, is it useable for large data sets (9gb)? Retrieved from https://community.rstudio.com/t/rstudio-is-it-useable-for-large-data-sets-9gb/21138/7

7. Yurtoğlu, N. (2018). http://www.historystudies.net/dergi//birinci-dunya-savasinda-bir-asayis-sorunu-sebinkarahisar-ermeni-isyani20181092a4a8f.pdf. History Studies International Journal of History, 10(7), 241–264. doi: 10.9737/hist.2018.658