Part 1: General data preparation and cleaning

b.) Machine Learning algorithms cannot be performed on raw data. Thus, cleaning and preparation of data is part is a very critical process of Machine Learning. The outcome of machine learning can be misleading without the relevant data to train the model.

I. The value 65535 can be identified as an outlier in "How.Many.Times.File.Seen". This can further be verified following methods as well.
   - First, "How Many Times Files Seen" has a very high skewness of 14.1.
   - According to the boxplot that has been created also identifies that values 65535 is an outlier.
   - Furthermore, Grubbs's test, Rosner's test have also been used as statistical tests to prove that value 65535 is an outlier.

II. The numeric "Thread Started" column has been converted into a factor. A factor is a special case of a vector that is solely used for representing nominal variables in R. The advantage of using "Thread Started" as a factor is that it is more efficient than using numeric vectors. Because the category labels are stored only once. Therefore, factor uses computer memory more effectively than other methods. Furthermore, some machine learning algorithms utilize special methods to handle categorical variables. Thus, programming these categorical variables as a factor may also ensure that the ML algorithm treats data appropriately. (Lantz, 2013)

III. As the Appendix A illustrates, original data in "Characters in URL" does not follow normal data distribution (the bell curve). It has a skewness of 13.91 which makes the column in question invalid to statistical analysing. However, when natural log transformation is applied to "Characters in URL", it removes the skewness of the original "Characters in URL" to -0.244 as the Appendix B shows. Which makes it more valid and valuable for statistical analysis. (B, 2018)

IV. There are 15349 NA values can be found in the given MLDATASET_PartiallyCleaned.xlsx. Most of the machine learning algorithms do not handle NA values gracefully. Thus, these missing values can affect the statistical calculation.
    There are quite a few techniques that can be used to deal with these missing values in R. However, removing NA values will be the solution for this time. (B, 2018)

C.) The main idea behind splitting data into training and testing is to estimate the performance and the accuracy of the machine learning algorithms. Furthermore, the training dataset will be used to train machine learning algorithms so that they can make accurate predictions on the data. The test dataset will be used to test how well that trained machine learning algorithm can predict. (B, 2018)

NOTE: This given data consists of total number of 74604 malicious and 75175 non-malicious files details. Therefore, this given data set can be considered as well-balanced dataset since it has almost 50/50 of malicious and non-malicious files. The fact that this dataset is well-balanced may help to have a better prediction with better machine learning.

Part_2: Compare ML algorithms performances.

Prior to the start of the machine learning process, the 'sample ID' and 'initial statistical analysis' were excluded from the database as the machine learning process had no contribution from these features.

Binary Logistic Regression

Since this prediction expects a binary outcome either YES or NO. It is better to use Binary Logistic Regression modelling. There are two packages that have been used to work with binary logistic regression.

- Caret - has been used to classify and regression train the data.
- Glmnet - has been used to visualise correlation matrix.

Furthermore, the generalized linear models (gml()) function has been used to perform the Binary Logistic Regression with the argument of *family="binomial".* This makes sure to use logistic regression instead of linear regression. According to the outcome of the glm function,

- Deviance residuals can be considered as fairly fine since they are close to being centred on 0 and roughly symmetrical.

Min	**1Q**	**Median**	**3Q**	**Max**
-2.3536	**-0.9984**	**-0.1981**	**0.9982**	**2.8969**

- Coefficient is the most interesting and important part of this model. According to the coefficients,
    - When the outcome is a binary variable R chooses the first level as the reference level by default. For instance, in this dataset the binary outcome should be YES or NO in the Actually.Malicious. Since the No come first alphabetically, NO will be the reference level in this case. Thus, the BLR model will compare the odds of malicious over the odds of non-malicious. Likewise, Exposed Corporate Intranet Pages is the reference level in Download Source. .com is the reference level in TLD. Evidence.of.Code.Obfuscation NO is the reference level and 1 is the reference level in Threads.Started.
    - Download.Speed, How.Many.Times.File.Seen, Evidence.of.Code.Obfuscation, Mean.World.Length.of.Extracted.Strings and Evidence.of.Code.ObfuscationYES are significant predictors when it is predicting the maliciousness.
        - As 'Mean Word Length of Extracted String' increases by 1, the log of the odds of the file maliciousness goes up 1.17. In other words, the odd of a file is being malicious increase 3.2% as 'Mean Word Length of Extracted String' increases by 1.
        - However, on the other hand, as the times that the file is seen increases by 1, the log of the odds of the file maliciousness decreases 0.0031. if it is said in another way, the odd of a file is being malicious decreases approximately 1% as 'How Many Times File Seen' increases by 1.
    - In addition to that, above-mentioned predators also have p-values which are well below 0.05. Therefore, these log of the odds and log of the odds ratios are statistically significant factors to the maliciousness.

- Basically, 'Evidence.of.Code.ObfuscationYES' and 'Download.SpeedGreater than 10MB/s' inversely corelated, while 'Download Speed' 1MB/s to 10MB/s, Mean.Word.Length.of.Extracted.Strings and 'Evidence.of.Code.Obfuscation YES' are positively corelated to the 'Actually Malicious'.
- Other variables like Files.Size.Bytes, Similarity.Score and Ping.Time.To.Server have negative correlation to the maliciousness. But they are not significant predictors of the maliciousness.

Therefore, this model's log of odds (log(odds)) is -4.214 and it also has 27 of log of the odds ratios. Moreover, this logit model will be given by

```
logit(p) = (-4.214) + (-0.5734) x Download.SourceGit Repository + (-5.205e-01) x Download.SourceLinked Directly to Google Search Result +
... + (-5.482e-02) x TLD.edu + .... + (1.274e+00) x Download.Speed 1MB/s to 10MB/s ... + (-5.987e-05) x Ping.Time.To.Server + (1.174e+00) x
Mean.Word.Length.of.Extracted.Strings...
```

Moreover, Binary Logistic Regression can be plotted as showing in Appendix C. This model pseudo-R^2 (R squared) values is 0.11 which is the overall effect size. After that the R script will calculate the probability of Malicious. According to the ROC calculations the best threshold to classify maliciousness of a file can be set to 0.5. Which means that the probability of maliciousness is greater than 0.5, it classifies as malicious and vice versa.

According to the Confusion matrix:

	YES	NO
YES	24030	8231
NO	28192	44391

24030 Actually Malicious files out of 52222 were correctly identified. With 67.4% of true positives.
44391 out of 52622 Non-Malicious files were correctly identified by the BLR model. With 69.9% of true negatives.

- However, this model also has 54% of False negatives as well as 15.6% of False positives.
- Furthermore, this BLR model True positives rate also known as Sensitivity is 46%. While the True Negative Rate or Specificity is around 84.4%.
- Finally, the overall accuracy is around 65.2%. Nonetheless, the accuracy of this model is driven by specificity alone.

Logistic Elastic-Net Regression

Basically, Logistic Elastic-Net Regression is a combination of both ridge regression and lasso regression. The train function from caret has been used to perform elastic-net regression. Actually.Malicious has been given to the function as the formular, Since the defined formula is a factor, the function treats model logistic regression modelling. Furthermore, the script has used a different set of values α and λ as tuning hyperparameters which allows to choose the best model among the set of models. In this case, a sequence of 100 values will be used for λ and among all these values the optimal result is given by the best α value with 0.1 and the best λ value with 0.3053856.

Best α	Best λ
0.1	0.3053856

According to the Coefficients:

- The model has been simplified by removing quite a few values from the mode. Since they no longer contribute to the model.
  - There are seven Download.Source features have been removed from the model compared to Exposed Corporate Intranet Pages.
  - And 5 TLD features have been removed from the model compared to .com

- Ping.Time.To.Server, File.Size.Bytes and Similarity.score and Characters.in.URL have also been removed from the mode.

According to the confusion metrics:

	YES	NO
YES	35175	15849
NO	17047	36773

There are 34501 Actually Malicious files have been correctly identified by the Elastic-Net out of 52222. Thus, this mode has around 66.1% true positives.

- Moreover, 38647 Non-Malicious files have been correctly identified by this model out of 52622. Thus, this model's true negatives are around 73.4%
- Nonetheless, the model holds 33.9% of false negatives and 26.6% of false positives.
- This Elastic-Net model True positives rate also known as Sensitivity is 67.36%. While the False Positive Rate or Specificity is around 69.9%.
- Furthermore, this BLR model True positives rate also known as Sensitivity is 66.07%. While the False Positive Rate or Specificity is around 73.4%.
- Finally, the overall accuracy is around 66.07%. Nonetheless, the accuracy of this model is evenly driven by specificity.

Random Forest.

The random Forests is a combination of the simplicity of decision trees with flexibility resulting in tremendous upgrades in accuracy. In this scenario, the elements that was supposed to predict has been omitted entirely, Therefore, it is going to be an unsupervised Machine Learning Algorithm.

There are three packages that have been used for Random Forest. Furthermore, the ranger function has been used to train the Random Forest model. First, the R script utilizes the default parameters to train the Random Forest model. According to these default values, confusion matrix can be defined as bellow:

	YES	NO
YES	47302	8226
NO	4920	44396

This Random Forest model was able to correctly predict 47302 of actually malicious files correctly. Furthermore, it was also able to correctly predict 44396 of actually non-malicious files out of 52,622. This model has only 12.85 % of OOB error.

However, later it will be checked if these 500 trees, and default variables that are considered at each internal node are enough for optimal classification. Therefore, next time, R code has been used with slightly sophisticated code to train the model.

Prior to training the Random Forest on the dataset the hyperparameters are tuned simultaneously with use of the expand.grid function. However, for the sake of simplicity each of these hyperparameters include only 3 values. trained Hyperparameters are 'mtry' which specifies the number of splitting features in any node, 'min.node.size' which specifies the minimum number of nodes and sample.fraction specifies the presentation of the sample from the original data. However, for the sake of simplicity each of these hyperparameters include only 3 values.

Considering the top five results of the Random Forest model to the given dataset, the best combination has been given by 400 number of trees with 4 number of mtry (best number of splitting features for any given tree), minimum node size of 10. Furthermore, the best number of sample fractions from the original data set is 70% which is performed with replacement.

However, this best combination still has around 12.5% Out of Bag misclassification error. Which is a bit better than the model with default parameters.

Moreover, this model has 87.5% of accuracy which is driven mainly by the sensitivity. This best Random Forest model has a sensitivity of about 90.4% and a specificity of about 84.6%. Nonetheless, according to the table in the Appendix D, the third best Random Forest has also around 87.5% accuracy with 90.4% of sensitivity and 84.6% of specificity which is also most like the best model. But it contains only 300 number of trees, which makes this model a bit faster than the best model. Therefore, this is another interesting factor that needs to be taken into consideration, before selecting the best model.

d.) Recommended Model.

The ultimate intention of this investigation is to develop the best malware detection algorithms based on the file's behaviors and some other parameters. There are few points that need to be considered to select the best out of these three.

In this scenario, the best algorithm should have a higher level of true positive rate since the main goal is to detect malware files. Therefore, the best Machine learning algorithm should have higher sensitivity. According to the following table, BLR and Elastic-Net Regression both have almost similar accuracy which is around 65% to 69%. However, BLR has low Sensitivity (~46%) and low accuracy. Thus, BLR can be dumped.

When it is compared with Elastic-Net Regression and Radom Forest. The Random Forest has the highest sensitivity rate as well as specificity and accuracy than Elastic-Net Regression. Furthermore, The Random Forest algorithm can be explained easily. Therefore, Random Forest has higher parsimony. In addition to that, Random Forest can be implemented in practice easily without any feature scaling processes and its interpretability is also higher compared to other selected algorithms. Which means the process of random forest can be understood by a non-technical personal. Most importantly, it also has the capability to handle missing value well. Since the outcome of this algorithm is binary, the random forest fits best. However, as the disadvantages of random forest, it is very sensitive to the training data. Even during this analysis only 30% of original data has been used to train the algorithm. There is a possibility of errors in these values with different datasets. Nonetheless, this disadvantage is minor compared to its advantages.

models	**Specificity**	**Sensitivity**	**Accuracy**
Binary Logistic Regression	84.49%	45.96%	65.26%
Elastic-Net Regression	73.44%	66.07%	69.77%
Random Forest.	84.59%	90.45%	87.51%

e.) Compare initial attempt to Random Forest.

During first analysis that was conducted by TOBORROM, there are 33728 actually malicious file have been correctly identified. And 44817 actually non-malicious files that have been correctly identified. However, it also has 7805 of false positives and 19494 of false negatives.

Furthermore, the initial attempts specification can be calculated as follows.

	YES	NO
YES	33728	7805
NO	18494	44817

- Specificity = (44817/ (44817+7805)) *100 = 85.17%
- Sensitivity = (33728/ (33728+18494)) *100 = 64.59%
- Accuracy = (44817+33728)/ (44817+7805) + (33728+18494) = 74.91%

Therefore, TOBORROM's initial attempt to classify samples has 85.17% of True Negative rate (specificity). And it has 64.59% of True Positive rate (Sensitivity). Furthermore, the initial attempt has 74.59% of accuracy. However, this accuracy is largely driven by the specificity.

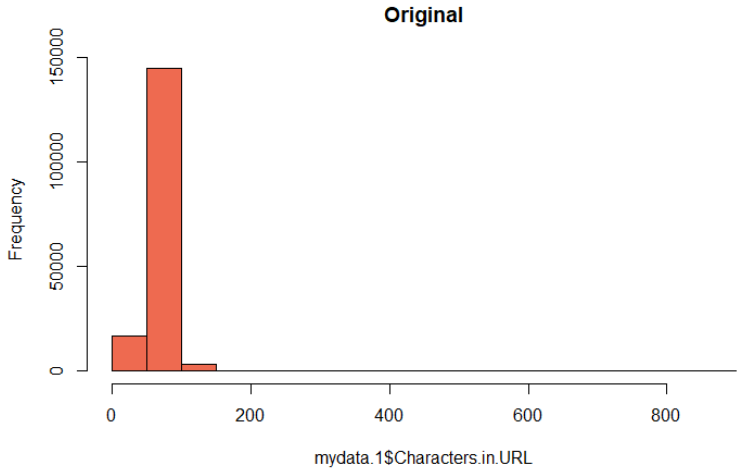
As this report mentioned above, the optimal algorithm should have a higher specificity since this analysis deals with malware detection. The Random Forest has higher specificity and accuracy than the initial ROBORROM attempt. That difference can be observed clearly from the plot in the Appendix F. Therefore, it is still recommended to use Random Forest tree as the optimal algorithm.

models	**Specificity**	**Sensitivity**	**Accuracy**
Initial Statical Analysis	85.17%	64.59%	74.91%
Random Forest.	84.59%	90.45%	87.51%

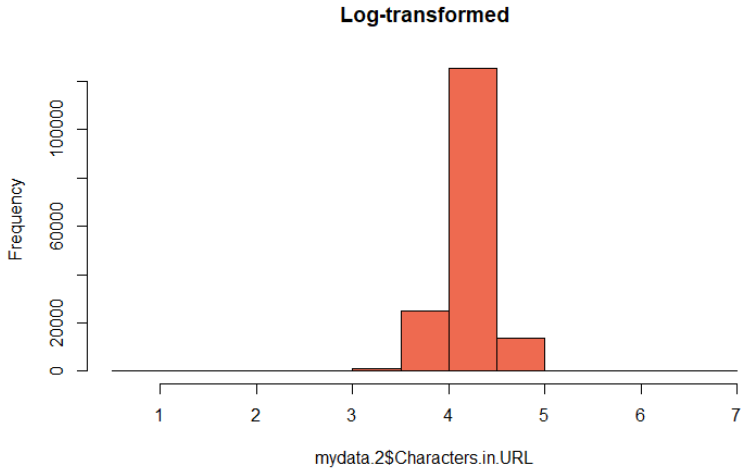
References:

1. B. (2018a). Cryptocurrency Mining Craze Going for Data Centers. Bitdefender. https://cdn2.hubspot.net/hubfs/341979/Cryptojacking/Bitdefender-Whitepaper-Cryptocurrency-Mining-Craze-Going-for-Data-Centers-2018.pdf (B, 2018)
2. Soetewey, A. (2020, August 11). Outliers detection in R. Stats and R. https://statsandr.com/blog/outliers-detection-in-r/#:%7E:text=of%20the%20plot).-,Boxplot,useful%20to%20detect%20potential%20outliers.&text=Observations%20considered%20as%20potential%20ou (Soetewey, 2020)
3. Lantz, B. (2013). Machine learning with R. Packet publishing ltd. (Lantz, 2013)
4. Kassambara, A. (2018). Machine learning essentials: Practical guide in R. Sthda.
5. Lantz, B. (2019). Machine learning with R: expert techniques for predictive modeling. Packt Publishing Ltd.

Appendix A: the histogram of original “Characters in URL

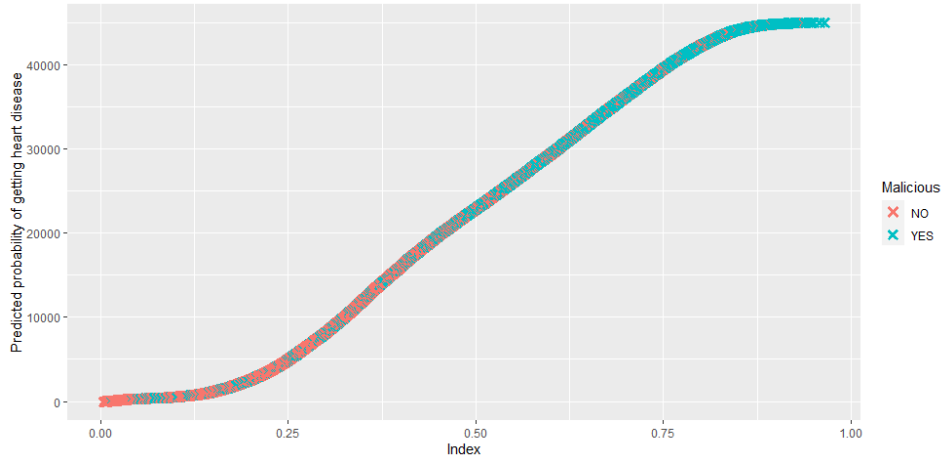


Appendix B: the histogram of Log-Transformed “Characters in URL”



Click here to return to where you were reading

Appendix C: Binary Logistic Regression Plot.



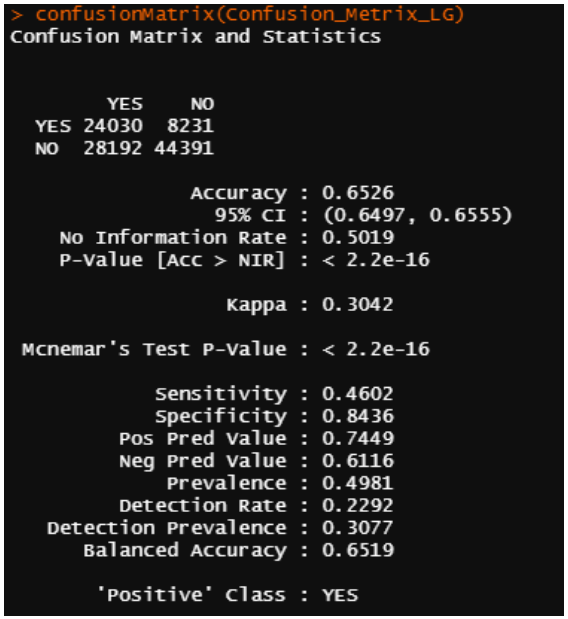
Click here to return to where you were reading.

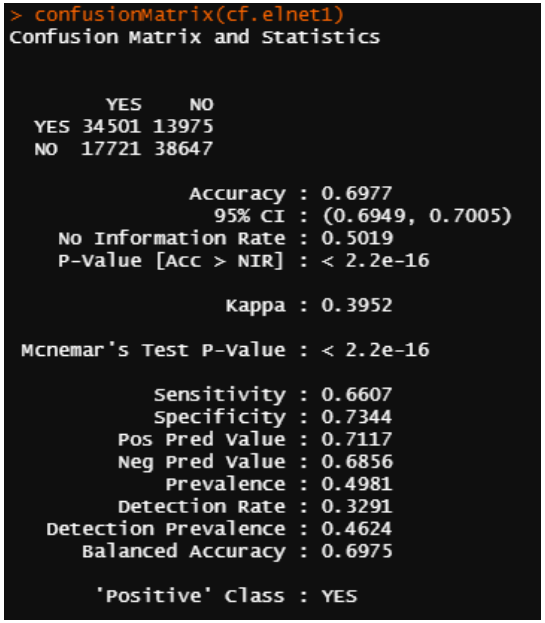
Appendix D: Random Forest outcome with tuned hyperparameters.

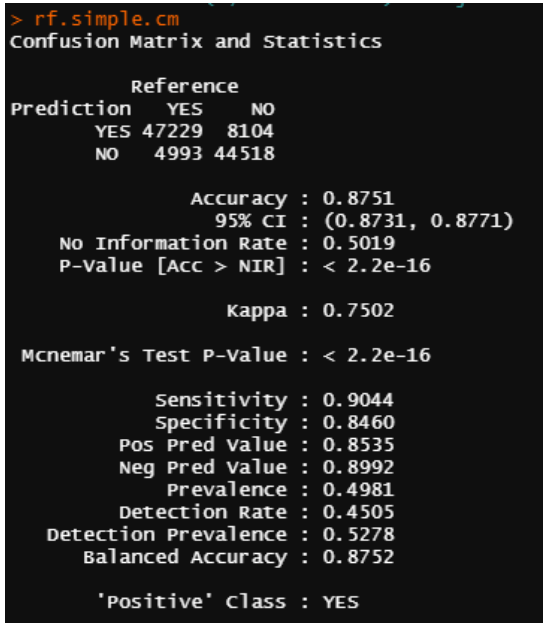
num.trees	mtry	min.node.size	replace	sample.fraction	sample.fraction	test.sen	test.spe	test.acc
400	4	10	TRUE	0.7	12.574	90.439	84.600	87.508
500	4	10	TRUE	0.7	12.574	90.462	84.603	87.521
300	4	10	TRUE	0.7	12.618	90.433	84.636	87.523
500	4	6	FALSE	0.5	12.629	90.290	84.712	97.490
500	4	10	TRUE	0.5	12.638	90.699	84.463	87.569

Click here to return to where you were reading

Appendix E: Screenshot of the Binary Logistic Regression, Elastic-Net Confusion Metrix, Random Forest models.







Appendix E: Random Forest vs Initial Statical Analysis.

Click here to return where you were reading.