Tanzim Zaki SAKLAYEN (ID: 10520140)

Final Assignment - MAT6202 – Data Analysis and Visualization

The malware dataset was split into 30/70 for training and test models respectively. The algorithm as mentioned in the attached R-code file in Part 1 was executed in the training dataset to optimize and predict the test data to identify the malware and non-malware samples in the dataset.

The ultimate reason for values that were 65,535 or greater assigned to N/A is because the values create a wider standard deviation while predicting a case in a machine learning model. Hence, the decision to assign the values to N/A reduces the estimation of error in calculation.

N/A values are implemented in the program because machine learning models do not support N/A values. Therefore, the decision to remove N/A values is only reasonable as the proportion of the values represents a small percentage and would not have a major effect on the algorithm models.

The major reason for converting all values greater than 5 to be equal to 5 was to reduce overfitting of data and execute better prediction.

Therefore, the steps mentioned in the R-code file are necessary to clean the data as a clear indication of partition of samples and omitting unnecessary data to eliminate noise in the dataset.

The random seed of the student ID generated 3 models to deploy:

- **Binary Logistic Regression (BLR):** The machine-learning statistical model generates a binomial outcome with 1 or more descriptive variables. The regression model measures the relationship between 1 or more independent features with the dependent categorical variable (yes/no) by estimating possibilities utilizing cumulative logistic distribution.
  A fine example would be detecting malicious network traffic in Endpoint Detection Response Platform such as Palo Alto Engine.
- **Logistic Elastic-Net Regression (LENER):** The model can be defined as a mixture of ridge and lasso regression models. The model penalizes to logistic model for consisting of too many variables. The results in shrinking the coefficients of the less contributive variables toward zero.
  Therefore, the LENER model shrinks some coefficients toward 0 (ridge regression) and sets some coefficients to exactly 0 (lasso regression).
- **Classification Tree Analysis (CTA):** CTA is a machine learning algorithm used for analyzing structural mapping of binary decisions that generate categorical decisions such as complexity parameter with the assistance of cross-validation (CV).
  The model considered the training dataset and constructs a decision tree based on measured attributes on the test dataset.

Sensitivity can be referred to as the detection rate for actual malicious samples whereas specificity can be referred to the detection for actual non-malicious samples in the confusion matrix analysis.

The following terminology is crucial to understand to carry on tests on specificity, sensitivity, and accuracy:

- True Positive (TP): Correctly Identified as Malware
- False Positive (FP): Incorrectly Identified as Malware
- True Negative (TN): Correctly Identified as Clean
- False Negative (FN): Incorrectly Identified as Clean

Hence, the formulas are:

- Sensitivity: (TP)/(TP+FN)
- Specificity: (TN)/(TN+FP)
- Accuracy: (TP+TN)/(TP+FP+TN+FN).

Interpretation of Search Strategy, Hyperparameter Tuning, and Pruning

Binary Logistic Regression (BLR): A predicted probability code was executed to assign the allocated class of the model in line 94 and 95 respectively. Later, the calculation of the confusion matrix was summarized to further investigate the sensitivity, specificity, and accuracy of the malware dataset.

Logistic Elastic-Net Regression (LENER): The optimized hyperparameter values are defined in the R-code file for the LENER model. 10-fold CV was set in the models by initiating the number of repeats to 5 times for training the model. Moreover, lambda values were set to 50 to obtain an optimal solution.

The optimized lambda value of the model was 0.373 and the optimized alpha value was 0.1 to conduct the analysis. Therefore, the optimized lambda value suggests a mediocre low that can be referred to as the model was more complex and might execute the risk of overfitting the dataset. A further recommendation is to collect more raw data to generate an algorithm for better accuracy in the future.

The ‘glmnnet’ function was executed for penalizing the model whereas the ‘expand.grid’ function was utilized to tune the hyperparameter in the algorithm. As observed in the R-code file by initiating the coefficient analysis on line 135, the coefficient shrunk closer to 0 and has retained all the relevant variables. Therefore, the code assisted to evaluate further for an analysis of the confusion matrix, sensitivity, specificity, and accuracy of the malware dataset.

Classification Tree Analysis (CTA): The model predicted classes of actual malicious files on the test data set mentioned in lines 175 and 176 that generated a confusion matrix. Later, further investigation was carried out to discover the sensitivity, specificity, and accuracy of the malware dataset. Line 179-183 defines the dataset to be cleaned/tuned and correctly identify and label the malware in the dataset.

Line 195 and 196 suggested a pruning in the classification tree where 10-fold CV was set with a tuned length of Complexity Parameter (CP) values equal to 15. At last, a plot diagram was executed, and Cross-Validation (CV) results were set for various CP values (Line 194-198). The plot interprets that further pruning was not necessary since the accuracy of the CTA decreases monotonically from the 8th CP value.

Comparison of Machine Learning Models

Binary Logistic Regression (BLR): Table 1 below presents an analysis of the Confusion Matrix of the algorithm consisting of specificity, sensitivity, and accuracy of the model. Please refer to line 107-111 in the R-code file for the execution formula.

		Actual ↓	Actual ↓
		Yes ↓	No ↓
Prediction →	Yes →	35,40 (67.8%)	15,939 (30.3%)
Prediction →	No →	16,820 (32.2%)	36,683 (69.7%)
	Total files →	52,222 (100%)	552,622 (100%)
	Accuracy: 68.75%	Sensitivity: 69.71%	Specificity: 67.79%

Table 1: Confusion Matrix of BLR

Table 1 demonstrates that there was equality in predicting the appropriate cases and proportion of malware and non-malware files from various sources in the LENER model. However, the accuracy calculation slightly error-free that is 68.75% as further investigation is required to discover the validation of the malware dataset.

Logistic Elastic-Net Regression (LENER): Table 2 below presents an analysis of the Confusion Matrix of the algorithm consisting of specificity, sensitivity, and accuracy of the model. Please refer to line 154-158 in the R-code file for the execution formula.

		Actual ↓	Actual ↓
		Yes ↓	No ↓
Prediction →	Yes →	34,354 (65.8%)	13,843 (26.3%)
Prediction →	No →	17,868 (34.2%)	38,779 (73.7%)
	Total files →	53,222 (100%)	52,622 (100%)
	Accuracy: 69.75%	Sensitivity: 73.69%	Specificity: 65.78%

Table 2: Confusion Matrix of LENER

Apart from the slightly improved calculation by an approximate 4% in sensitivity and deficit of 2% in specificity compared to the BLR model, the accuracy calculation slightly error-free that is 68.75% as further investigation is required to discover the validation of the malware dataset such as CTA is required to discover the validation of the malware dataset.

Classification Tree Analysis (CTA): Table 3 below presents an analysis of the Confusion Matrix of the algorithm consisting of specificity, sensitivity, and accuracy of the model (cleaned/tuned dataset). Please refer to line 188-192 in the R-code file for the execution formula.

		Actual ↓	Actual ↓
		Yes ↓	No ↓
Prediction →	Yes →	35,586 (66.86%)	7, 795 (14.81%)
Prediction →	No →	18,636 (35.02%)	44,827 (85.19%)
	Total files →	53,222 (100%)	52,622 (100%)
	Accuracy: 81.34%	Sensitivity: 78.92%	Specificity: 83.79%

Table 3: Confusion Matrix of CTA (cleaned/tuned version)

Table 3 clearly defines that the calculation of error-free estimation was drastically improved to identify the proportion of malware and non-malware files from various sources in the CTA model. The accuracy rate is highly error-free with a rate of 81.34% supporting the rate of sensitivity with an increase by approximately 5% and the rate of sensitivity with an increase by almost 18% compared to the LENER model.

Therefore, CTA proved to be the better performing prefixing cases precisely with a smaller number of misclassifications of malware and non-malware files/samples.

Evaluation of Machine Learning Models

Table 4 below presents an overview of the specificity, sensitivity, and accuracy of all 3 models.

	Sensitivity ↓	Specificity ↓	Accuracy ↓
BLR →	69.71%	67.79%	68.75%
LENER →	73.69%	65.78%	69.75%
CTA →	78.92%	83.79%	81.34%

Table 4: Sensitivity, Specificity, and Accuracy measures of all 3 models

Initially, line 173 in the R-code generated a confusion matrix of the uncleaned dataset before pruning the test set. The results of the unpruned matrix of CTA were unambiguous and similar to BLR and LENER as the accuracy, sensitivity, specificity generated a lower proportion rate of malware with error-free estimation to the dataset.

The evaluation of the prediction test set on the dataset can be justified that BLR and LENER performed equally with no major difference to estimate the proportion of malware in the total dataset. In contrast, CTA performed well enough as the calculation consumes more duration of analyzing error-free estimations while executing the search values during the tuning process of the model.

Moreover, CTA would be a wise choice to consider in terms of parsimony with the performance of the dataset as it can handle a larger dataset and perform better on predicting positive and negative cases of malware.

To conclude, one can recommend that further test analyses be carried out such as Bagging Tree and Random Forest while tuning more datasets for improved estimation of discovering the proportion of actual malware in the dataset.