Assignment 2
Machine Learning Modelling

Scenario

WA Cyber Command – WACY-COM has acquired aggregate data about 200,000 identified cyber-attacks and scans. The data are sourced from a Honey-pot project which places fake servers across the globe and records attacker activity and techniques. As Honeypots are simulated networks and devices, they allow researchers to safely monitor malicious traffic without endangering real computers or networks.

When analysing cyber-attacks, the level of sophistication of attackers can range in from low-level scammers, right up to Advanced Persistent Threats (APTs) which are often associated with state-sponsored cyber-attacks. The attacker tools and techniques generally vary depending on the sophistication of the attacker.

A research project has been undertaken by WACY-COM to determine what patterns exist in state-sponsored APT attacks.

Typically, a complex attack can involve multiple attacking computers (with different source-IP addresses) and different payloads and targets. By coordinating attacks from multiple devices, the attacks can become more difficult to detect and stop.

Note: The scenario and data are loosely based on real-world cyber threats and attacks. However, this data set has been curated entirely to help you understand the types of data, correlations and issues that you may experience when handling real-world cyber security data.

Data description

The aggregated data available to WACY-COM are described by the following features (with data types given in square brackets):

[Categorical] Port – The port or service that was being attacked on the honey-pot network. Well known ports include 80/443 (Web traffic), 25 (Email reception), 993 (Email collection)
[Categorical] Protocol – The Internet Protocol in use to conduct the attack
[Numeric] Hits – How many 'hits' the attacker made against the network
[Numeric] Average Request Size (Bytes) – Average 'payload' sent by the attacker
[Numeric] Attack Window (Seconds) – Duration of the attack
[Numeric] Average Attacker Payload Entropy (Bits) – An attempt to qualify whether payload data were encrypted (higher Shannon entropy may indicate random data, data obfuscation or encryption)
[Categorical] Target Honeypot Server OS – The Operating System of the simulated server
[Numeric] Attack Source IP Address Count – How many unique IP addresses were used in the attack

[Numeric] Average ping to attacking IP (milliseconds) – Used to detect 'distance' to the attacker. The average ping time 'back' to the attacker's IP addresses were calculated.
[Numeric] Average ping variability (st.dev) – High variability pings can indicate a saturated or unreliable link.
[Numeric] Individual URLs requested – How many different URLs were probed or attacked (Only relevant for Web Server ports)
[Categorical] Source OS (Detected) – The detected operating system of the attacking IP address. Acquired by scanning and fingerprinting the IP address of the attacking server
[Categorical] Source Port Range – What range of source ports were used by the attacker. Typically, 'low' ports are reserved for system services. Higher ports are used by end-user applications.
[Categorical] Source IP Type (Detected) – Whether the IP of the attacker can be linked to known proxies/VPNs or TOR (technologies that can be used to hide the real source of the attack), or Likely ISP traffic (which may indicate the attacker is leveraging compromised end-user computers)
[Numeric] IP Range Trust Score – A trust score generated by an existing WACY-COM system. This system integrates with open-source intelligence (OS-Int) databases to identify potentially compromised on malicious IP addresses
[Binary] APT – Was the attack conducted by a known Advanced Persistent Threat actor (APT).

The raw data for the above variables are contained in the **ML_dataset2.csv** file.

Initially the research team believed they would be able to gain insight from various statistical analyses of the dataset. Their initial attempts to classify data lacked sensitivity and had many false positives. The results of WACY-COM's analysis have been included in the **Initial.Modelling.Result** column – the results of this analysis are unacceptable.

Objectives

You have been brought on as part of a data analysis team to determine if APT activity can be inferred from other attack parameters.

Task

You are to train your selected supervised machine learning algorithms using the master dataset provided, and compare their performance to each other and to WACY-COM's initial attempt to classify the samples.

Part 1 – General data preparation and cleaning.

a) Import the ***ML_dataset2.csv*** into R Studio. This version is the same as Assignment 1, but with an addition column at the end.

b) Write the appropriate code in R Studio to prepare and clean the ***ML_dataset2*** dataset as follows:

   i. Clean the whole dataset based on what you have suggested / feedback received for Assignment 1.
   ii. For the feature **Source.OS.Detected**, merge its categories **Windows 10** and **Windows Server 2008** together to form a new category, say **Windows_All**. Similarly for **Target.Honeypot.Server.OS**, merge its categories **Windows (Desktops)** and **Windows (Servers)** to form the new category named **Windows_DeskServ**. Further, combine **Linux** and **MacOS (All)** to form the category **MacOS_Linus**. Hint: use the *forcats:: fct_collapse(.)* function.
   iii. Log-transform **Average.ping.variability** using the *log(.)* function, and remove the original **Average.ping.variability** column from the dataset (unless you have overwritten it with the log-transformed data). Similarly, transform the following features using the square root, i.e. *sqrt(.)*, function instead.
      1. **Hits**;
      2. **Attack.Source.IP.Address.Count**;
      3. **Average.ping.to.attacking.IP.milliseconds;**
      4. **Individual.URLs.requested**.
   iv. Select only the complete cases using the *na.omit(.)* function, and name the dataset ***ML_dataset_cleaned.***

Briefly outline the preparation and cleaning process in your report and why you believe the above steps were necessary.

c) Write the appropriate code in R Studio to partition the data into training and test sets using an **30/70 split**. Be sure to set the randomisation seed using your **student ID**. Export both the training and test datasets as csv files, and these will need to be submitted along with your code.

Note that the training set is typically larger than the test set in practice. However, given the size of this dataset, you are asked to use 30% of the data only to train your ML models to save time.

Part 2 – Compare the performances of different ML algorithms

a) Randomly select **THREE** supervised learning modelling algorithms to test against one another by running the following code. Make sure you enter your student ID into the command *set.seed(.)*. Your 3 ML approaches are given by **myModels**.

```
library(tidyverse)
set.seed(Enter your student ID)
models.list1 <- c("Logistic Ridge Regression",
```

```r
                  "Logistic LASSO Regression",
                  "Logistic Elastic-Net Regression")
models.list2 <- c("Classification Tree",
                  "Bagging Tree",
                  "Random Forest")
myModels <- c("Binary Logistic Regression",
              sample(models.list1,size=1),
              sample(models.list2,size=1))
myModels %>% data.frame
```

For each of your three ML modelling approaches, you will need to:

b) Run the ML algorithm in R on the **training set** with **APT** as the outcome variable. **Exclude** **Sample.ID** and **Initial.Modelling.Result** from the modelling process.

c) Perform hyperparameter tuning to optimise the model (except for the Binary Logistic Regression model):

- Outline your hyperparameter tuning/searching strategy for each of the ML modelling approaches. Report on the search range(s) for hyperparameter tuning, which k-fold CV was used, and the number of repeated CVs (if applicable), and the final optimal tuning parameter values and relevant CV statistics (i.e. CV results, tables and plots), where appropriate.
- If your selected tree model is **Bagging**, you must tune the **nbagg**, **cp** and **minsplit** hyperparameters, with **at least 3 values** for each.
- If your selected tree model is **Random Forest**, you must tune the **num.trees**, **mtry**, **min.node.size**, and **sample.fraction** hyperparameters, with **at least 3 values** for each.
- Be sure to set the randomisation seed using your **student ID**.

d) Evaluate the performance of each ML models on the **test** set. Provide the confusion matrices and report and describe the following measures in the context of the project:

- Sensitivity (the detection rate for APT)
- Specificity (the detection rate non-APT)
- Overall Accuracy

e) Provide a brief statement on your final recommended model and why you chose that model over the others. Parsimony, and to a lesser extent, interpretability maybe taken into account if the decision is close. *You may outline your model coefficients for your logistic or penalised model if it helps your argument.*

f) Create a confusion matrix for the variable **Initial.Modelling.Result** in the **test set**. Recall the data in this column correspond to WACY-COM's initial attempt to classify the samples. Compare and comment on the performance of your optimal ML model in part e) to the initial modelling results by the WACY-COM research team.

What to submit

Gather your findings into a report (maximum of 5 pages) and citing sources, if necessary.

Present how and why the data was manipulated, how the ML models were tuned and finally how they performed to each other and to the initial analysis by WACY-COM. You may use graphs, tables and images where appropriate to help your reader understand your findings.

Make a final recommendation on which ML modelling approach is the best for this task.

Your final report should look professional, include appropriate headings and subheadings, should cite facts and reference source materials in APA-7th format.

Your submission must include the following:

- Your report (5 pages or less, excluding cover/contents page)
- A copy of your R code, and two csv files corresponding to your training and test datasets.

The report must be submitted through **TURNITIN** and checked for originality. The R code and data sets are to be submitted separately via another submission link.

Note that no marks will be given if the results you have provided cannot be confirmed by your code. Furthermore, all pages exceeding the 5-page limit will not be read or examined.

Marking Criteria

Criterion	**Contribution to assignment mark**
Accurate implementation data cleaning and of each supervised machine learning algorithm in R.	20%
Explanation of data cleaning and preparation.	10%
An outline of the selected modelling approaches, the hyperparameter tuning and search strategy, the corresponding performance evaluation in the **training set** (i.e. CV results, tables and plots), and the optimal tuning hyperparameter values.	20%
Presentation, interpretation and comparison of the performance measures (i.e. confusion matrices) among the selected ML algorithms. Justification of the recommended modelling approach and how it compares against the initial modelling results in the **test set.**	30%
Report structure and presentation (including tables and figures, and where appropriate, proper citations and referencing in APA-	20%

7th style). Report should be clear and logical, well structured, mostly free from communication, spelling and grammatical errors.	

Academic Misconduct

Edith Cowan University regards academic misconduct of any form as unacceptable. Academic misconduct, which includes but is not limited to, plagiarism; unauthorised collaboration; cheating in examinations; theft of other student's work; collusion; inadequate and incorrect referencing; will be dealt with in accordance with the ECU Rule 40 Academic Misconduct (including Plagiarism) Policy. Ensure that you are familiar with the Academic Misconduct Rules.

Assignment Extensions

Applications for extensions must be completed using the ECU Application for Extension form, which can be accessed online.

Normal work commitments, family commitments and extra-curricular activities are not accepted as grounds for granting you an extension as you are expected to plan ahead for your assessment due dates.

Please submit applications for extensions via email to both your tutor and the Unit Coordinator.

Where the assignment is submitted no more than 7 days late, the penalty shall, for each day that it is late, be 5% of the maximum assessment available for the assignment. Where the assignment is more than 7 days late, a mark of zero shall be awarded.