Creating a model to detect malware using supervised learning algorithms

Background

N00BIoT Email Sentry 2.0

N00BIoT's Email Sentry is a malware detection platform. The first version of Email Sentry wasn't particularly effective, so the N00BIoT commissioned you as an expert in machine learning. An early phase of the project used principal component analysis to determine if there were specific factors about emails that could help to identify malicious emails.

Based on this the N00BIoT software team tried to further refine their malware classification system. The results were still underwhelming.

The decision has been made to explore further supervised learning models to create a more effective malware classifier.

MalwareSamples Data

The programming team has again provided you with email data. Two sets of data are provided to help with your investigation of the accuracy of various supervised learning models.

The first data file **MalwareSamples10000.csv** is a curated dataset. The data are sampled from emails such that approximately 50% of the data contain malware samples, and 50% of the data are from legitimate emails. This data set may be used for training of your machine learning models.

EmailSample Data

This data set (**EmailSamples50000.csv**) contains a sample chosen randomly from all emails processed by N00BIoT (without any consideration for whether the sample was malicious or legitimate). This set provides a reasonable approximation of total monthly email activity in a busy N00BIoT client environment. You will note through a brief examination of the data, that there are far fewer malicious emails in the EmailSample data – it is important to note that an overly zealous classifier may result in many false positives.

SCENARIO

Following your initial consultation with N00BIoT, the software development team has extracted data sets based upon your recommendations.

N00BIoT intends to launch a new version of Email Sentry at the end of the year. It will be marketed as *N00BIoT ES2 (Powered By AI).*

The software team is scrambling to produce a reliable email detector and has turned to you to provide the machine learning expertise and analysis to deliver a product with the following goals:

- Very low false-positives on malware detection
- High level of sensitivity in detecting malware.

TASK

You are to apply supervised machine learning algorithms to the data provided. You will train your ML model using the MalwareSample set, and then test them against the EmailSamples data set.

All analyses are to be done using **R**. You will report on your findings.

Part 1 – Preparing your data for constructing a supervised learning model using MalwareSamples10000.csv

You will need to write the appropriate code to,

i. Import the dataset **MalwareSamples10000.csv** into R studio.
ii. Set the random seed using your student ID.
iii. Partition the data into training and test sets using an **80/20 split**.

The variable **isMalware** is the classification label and the **outcome** variable.

Part 2 – Evaluating your supervised learning models

a) 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 modelling approaches are given by **myModels**.

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

b) For each of your supervised learning approaches you will need to:
   i. Run the algorithm in R on the **training set.**
   ii. Optimise the hyperparameter(s) of the models (except for binary logistic regression model).
   iii. For the binary logistic regression model, perform recursive feature elimination (RFE) on the model to ensure the model is not overfitted. See Workshop 5 for an example, except in this instance, specify the argument **function=lrFuncs** in the *rfeControl(.)* command instead.
   iv. Evaluate the predictive performance of the models on the **test set**, and provide the confusion matrix for the estimates/predictions, along with the sensitivity, specificity and accuracy of the model.

c) For the binary logistic regression model, report on the RFE process (i.e, information on which *k*-fold CV was used, and the number of repeated CV if using **repeatedcv**) and the evaluation of the candidate (simplified) models, and the selection of the final logistic regression model.

d) For the other two models, report how they were tuned, including information on search range(s) for the tuning hyperparameter(s), 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 (where appropriate).

e) Report on the predictive performances of the three models and how they compare to each other.

Part 3 – "Real world" testing

a) Load new test data from the "real world" **EmailSamples50000.csv**.

b) For each of your models (with the optimised parameters which you have identified in part 2), run your classifier on the **EmailSamples50000.csv** test data.

c) For each optimised model, produce a confusion matrix and report the following:
   i. Sensitivity (the detection rate for actual malware samples)
   ii. Specificity (the detection rate for actual non-malware samples)
   iii. Overall Accuracy

d) A brief statement which includes a final recommendation on which model to use and why you chose that model over the others. Parsimony, accuracy, and to a lesser extent, interpretability should be taken into account.

What to Report

You **must** do all of your work in R.

1. Submit a single report containing:
    a. A description of your three selected supervised learning algorithms.
    b. For each algorithm:
        i. The optimised hyperparameters for the algorithm.
        ii. A confusion matrix on the test set of the MalwareSamples.csv data showing the accuracy of the algorithm with the optimised parameters.
        iii. A confusion matrix showing the accuracy of the algorithm for the 'real world' EmailSamples.csv data
        iv. A description of the accuracy, sensitivity and selectivity of the optimised algorithm when applied to the 'real world' data.
    c. A paragraph explaining your chosen algorithm and parameters and why this was chosen over the alternatives. Written in language appropriate for a **non-mathematical audience**.

    Note: At the end you will present your findings of 3 algorithms showing 2 confusion matrix tables for each (1 for the **MalwareSamples** dataset, and 1 for the **EmailSamples** dataset). You will also need present a description of, **describe and compare** the accuracy, sensitivity and **specificity** of each of the 3 algorithms.

2. If you use any external references in your analysis or discussion, you must cite your sources.

3. **Copy only the relevant tables and figures to the report. Screenshots of the tables are not encouraged. You should copy the values across to Excel and format them appropriately.**

Marking Criteria

Criterion	Contribution to assignment mark
Good explanation of three appropriate supervised learning algorithms that are selected for the task	10%
Accurate implementation of each supervised machine learning algorithm	30%
Evidence of optimisation of each algorithm	20%
Correct explanation and discussion of accuracy, sensitivity and specificity for each algorithm	20%
Good explanation and justification for recommended algorithm, and tuning parameters.	10%
Communications skills - report and analysis well-articulated and communicated using language appropriate for a non-mathematical audience.	10%

Submission Instructions:

Your submission must include the following:

- Your report (5 pages or less, **excluding cover/contents page**)
- A copy of your R code

The report must be submitted through **TURNITIN** and checked for originality. The R code are to be submitted separately via a Blackboard 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.

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.

Before applying for an extension, please check out the ECU Guidelines for Extensions which details circumstances that can and cannot be used to gain an extension. For example, normal work commitments, family commitments and extra-curricular activities are not accepted as grounds for granting you an extension of time because 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.