Malware analysis

Homework 1 - Machine Learning
Engineering in Computer Science
"La Sapienza" University of Rome

Costa Marco 1691388

18 novembre 2018

Indice

1 Introduction

The purpose of this homework is to apply machine learning algorithms to malware analysis. This report discusses some methods to detect a malware and which family it belongs to. The DREBIN dataset was used for the experiment.

2 DREBIN dataset

DREBIN dataset contains 5.560 malwares and 123.453 bening applications. For each application, it contains a text file that describes all the features of the application. Each feature belongs to one of 8 categories (S1 to S8). Moreover, the dataset contains also a dictionary file in csv format in which there are the SHA1 Hash of the malware in the dataset and the family they belongs to.

3 Experiments

For both problems three different classification algorithms were used:

- Logistic Regression (LR):
  it was chosen to test because of its simplicity and fast computation.
- Support Vector Machine (SVM):
  it was also tested and able to outperform Logistic regression.
- Random Forest (RF):
  it is also a robust algorithm and usually perform very well in practice.

3.1 Malware detecion

This problem is to classify whether an application is malware or not. It can be tackled using the feature vectors.
Good metrics for this problem are Accuracy score and F1 score, since it combines precision and recall.
To solve the task I considered the following things:

1. Choice of dataset:
   The Drebin dataset contains 5560 positive examples and more than 100.000 negative examples, causing a problem of class imbalance. So experiment was performed both on the entire dataset and on a partial dataset.

2. Feature used:
   As suggested in the delivery, the experiment was carried out using both all the features and only a part of them.

So several tests have been carried out and these are the results:

- Using the entire dataset:
  - Using all the features:

Algorithm	**Accuracy**	**F1 score**	**Time**
Logistic Regression	0.960728	0.325643	0.883 s
Support Vector Machine	0.970158	0.549004	43.237 s
Random Forest	0.988761	0.861244	5.137 s

  - Using only few features (Permission, api_call and url):

Algorithm	**Accuracy**	**F1 score**	**Time**
Logistic Regression	0.957808	0.236559	0.32 s
Support Vector Machine	0.962433	0.298263	210.489 s
Random Forest	0.971734	0.626111	3.038 s

- Using a partial dataset (5560 malwares and 5560 bening applications):
  - Using all the features:

Algorithm	**Accuracy**	**F1 score**	**Time**
Logistic Regression	0.820743	0.814861	0.146 s
Support Vector Machine	0.863609	0.864462	1.319 s
Random Forest	0.932254	0.932898	0.553 s

  - Using only few features (Permission, api_call and url):

Algorithm	**Accuracy**	**F1 score**	**Time**
Logistic Regression	0.790168	0.777778	0.028 s
Support Vector Machine	0.828537	0.833721	1.486 s
Random Forest	0.864508	0.865636	0.382 s

3.2 Malware classification

Each of the 5560 malwares belongs to one of 179 malware families. The goal of this task is the follow:
Given a particular malware, determine which family it belongs to.
A good metric for this problem is Accuracy score to measure the performance of the algorithms (given a set of malwares, how many malware the algorithm can classify the correct family).
For this experiment, only malicious applications was used and the dictionary in the dataset was used as ground truth.
These are the results of this experiment:

Algorithm	**Accuracy**	**Time**
Logistic Regression	0.638489	4.31 s
Support Vector Machine	0.780576	1.663 s
Random Forest	0.892086	0.591 s

4 Code

Python programming language was used for this task.
In addition, the following libraries have been used:

- Numpy: a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays;
- Scikit-learn: a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms.

5 Results

Three different algorithms were tested on two different problems. The analysis of the results is similar between these two problems:
Random forest performed better than support vector machine, although they both outperformed logistic regression. However, logistic regression performed faster than the others.

6 Conclusions

After analyzing the results of the various experiments, it can be concluded that:

- Random forest performs better than support vector machine and logistic regression in both the problems;
- Logistic regression is faster than the other two, but is the worst in the classification;

- The use of some features instead of all causes a decrease in the performance of all algorithms;
- The use of a partial dataset allows to obtain a greater f1 score, but a decrease in accuracy score;