Fraud detection challenge

Abstract

In this challenge you will face real use-case data in which you will have to complete the entire process of detecting masqueraders (משתמשים מתחזים). You will have to: (1) examine the data, (2) preprocess it, (3) analyze it, (4) extract features and (5) apply data mining methods that you acquired during the course. Your final task will be to distinguish masqueraders from benign users, from the history of bash commands using data mining methods.

The Dataset

The directory **FraudedRawData** contains 40 users' history bash commands segments. For every UserX (X=1..40), you have a file that contains 15,000 bash commands such that each 100 commands are defined as a segment. The first 5,000 entries (=50 segments) in each file are training entries, i.e., they are guaranteed to be UserX's commands.

The other 10,000 entries (=100 segments) contain both the segments of UserX and of masqueraders. 90 of the segments are genuine (i.e., benign) and 10 segments are entered by a masquerader (randomly sorted). Your task is to differentiate the 10 sessions from the other 90.

The file **challengeToFill.csv** provides the key for the rest of the task. It is a 40x150 matrix. Each row represents a user index and each column represents whether the segment of 100 commands has been entered by the user (labeled by 0) or by a masquerader (1).
For example, with our counting starting at 0, the commands 6400 - 6500 in file User0 have actually been entered by a masquerader (represented as "0" in cell BN2 in Excel or 65th in the row). Commands 1200 – 1300 (represented as "1" by N2 also in Excel or the 13th cell in the row) is guaranteed to have been entered by User0.

For 10 out of the 40 users (User0…User9) - Full classification is given in the **challengeToFill.csv** for you to evaluate your algorithm before you apply it on the other 30 users.

Please ask all your relevant questions in the course forum to make it available for other students. (On moodle2)

Guiding questions

- What kind of problem is this (i.e., classification, clustering, regression – could be all)?
- What defines an instance?
- What type of features could we extract?
- What types of commands are in there?
- Is there a meaning to the command length?
- Is there a meaning to several commands in a row?
- What kind of statistical information can I produce?
- What type of criteria can help me evaluate my methods? (Accuracy will probably not help here).

In the **challengeToFill.csv** you were already supplied with 10 users' train and test set full labels. First, try your methods on them and then apply them among the rest.

Grading

For each correct prediction of a benign segment you will receive **1 point**. For each correct prediction of a masqueraded segment you will receive **9 points**. Consider the fact that in the test set you have 30 users, and each one them has 10 masqueraded segments and 90 benign segments. The maximum score is 30x(90x1+10x9)=5400. Notice that labeling all the instances with a single label (0 or 1) will produce a score of 2700.
Your scores will be calculated according to the following formula:

$$FinalGrade = 0.7 * min\left(100, \left(\frac{ClassificationScore}{4575}\right) * 0.95\right) + 0.3 * ReportGrade$$

Besides the final grade score of the classification, the work will be also evaluated according to the creativity of the feature engineering and the use of better fitting algorithms.
The two highest scoring students will receive extra credit (1 point to the final course grade).

Submission

Fill the blank spaces in the **challengeToFill.csv** file with 1/0 according to your solution. Fill '1' for masquerader segment and '0' for the genuine user commands segment. Make sure you fill all the blank spaces in the matrix only with 1/0, and the rows' order is User0… User39.

Intermediate assessment

In order for you to assess your progress (compared to your previous attempts as well as those of your peers), intermediate tests will be held during the period until the submission deadline. As part of these tests we will evaluate and publish your results on the other 30 users (user10…user39). The intermediate evaluation will be provided once a week. You can submit up to 15 intermediate files (in the entire period until the deadline) challengeToFill.csv under the name: 123456789_987654321_x.csv where the numbers are your id's and x is the iteration counter starting from 1. The intermediate submission will be made via email.

It is your responsibility to keep up with the course's message board for details about:

1. The final submission deadline (mandatory) – set to May 31st 2023.
2. The intermediate tests (optional)

You are encouraged to submit in pairs (although you don't have to).

Final Submission: the final submission should contain three (3) files:

- The last, most successful result file of challengeToFill.csv
- A description of your data mining methods and the feature engineering process. Highlight things that you think are creative and out-of-the-box thinking. The file should be up to 1 page, use the format of this document for sizing and font. Name it **description.doc/docx**.
- Also, add your source code to the submission in a directory called **code.** You do not have to be organized with your code, it will only be checked in case of the suspicion of cheating. You are encouraged to use other APIs and open source libraries, but submit only code that you wrote, even if it does not compile.
-

Submit all the deliverables in a single zip file entitled id1Number_id2Number.zip (for example 123456789_987654321.zip) through one of the group members' Moodle account.
Good Luck!