Research Task - Successful Request Spike

This exercise focuses on reading and processing application access logs, and building a detection model to trigger **Successful Request Spike** alerts (described below). It is supposed to take a few hours and requires experience in data analytics, and REST APIs.

Notes

- There is no single solution to this problem; it is an open task and you can take it wherever you would like to
- Strive for clarity and keep it as simple as possible
- Whenever you are unsure about anything, make some assumptions, document them, and proceed accordingly
- Document your work flow along with the algorithm
- Prefer Jupyter Notebook for documentation and implementation

Input Dataset

Attached is a dataset we have created containing requests generated by different IPs.

Initial Data Analysis

Try to understand the following:

- What is the meaning of each column in the access log? How can you use it in the model?
- How do user actions look in the logs?
- How can we detect abnormal behavior in this dataset?
- How should we display the abnormal behaviour so that an analyst can understand the detected event?

Please add the answers to these questions to your documentation.

Detection Model Requirements

Your detection model should trigger alerts on **Successful Request Spikes.**

A **Successful Request Spike** occurs when a single IP address is generating an extremely large number of successful requests to a single endpoint over 10 minutes, when compared with that generated by other IP addresses.

Notice that for each endpoint the extreme amount (threshold) can be different. Also, try triggering a reasonably low number of alerts.

Implementation

Design and implement an algorithm that triggers **Successful Requests Spike** alerts in the given dataset. The output of the algorithm should be a CSV file with **"alert"** rows containing the following columns:

- timestamp (when did the behavior take place?)
- caller_ip (which IP was misbehaving?)
- endpoint (which endpoint was seeing the spike in successful requests?)
- cnt_requests (How many requests performed in the 10 minutes interval)

Furthermore, provide figures answering the following questions the detection algorithm you've implemented -

- Display an alert of your algorithm in a graphical display showing the abnormal behaviour so that an analyst can understand the detected event
- Decide on simple measurements and KPIs to monitor your algorithm, compute them on the data and present your findings.

Test your algorithm

- How many alerts did you trigger?
- How do you avoid triggering too many?
- Do you use the same threshold for each endpoint?
- How do you calculate the threshold?

Please add the answers to these questions to your documentation.
When finished, reply to the email with your documentation, the algorithm and your output (alert.csv)

Good Luck!