Scriptspråk

Kurs_NÄTD24LIN_SCSP25

Workshop 2: Incident Analysis - CSV Data Processing

https://github.com/muhad308/Incident-Analysis---CSV-Data-Processing-Part-2

1. Problem Solving and Challenges:-

What were the biggest challenges you encountered in this workshop part?

How did you solve them?

- One of the biggest challenges was cleaning and standardizing data from the CSV file.
  - Cost values used commas instead of dots or contained spaces.
  - I solved this with a helper function parse_cost() that cleaned and converted the data safely to floats.
- Another issue was handling missing or invalid numbers in columns like resolution_minutes and affected_users.
  - I used conditional checks to convert valid values and default invalid ones to zero or None.
  - Example: -

```
row["affected_users"] = int(row["affected_users"]) if row["affected_users"].isdigit() else 0
```

- Managing grouped data (like incidents per site or severity) was tricky.
  - I used defaultdict and Counter from the collections module to simplify aggregation and avoid key errors.

2. Learning and Development

What have you learned that you couldn't do before?

- I learned how to use defaultdict and Counter from collections module to efficiently aggregate and count data without manual initialization.
- I gained hands-on experience with key Python modules like csv, statistics, and collections.
- I also learned how to generate multiple report files (.csv and .txt) automatically from one data source.

Which concept(s) was/are the most difficult to understand and why?

- The hardest concept was managing nested dictionary structures in device_statistics where each device stores multiple lists (costs, users, severity scores) for later averaging.
  - At first it was confusing how values could be lists or dictionaries inside dictionaries, but once I understood it, it became a powerful way to structure data.

3. Professional Relevance

How can you use these skills in your future role as a network engineer?

- These skills are directly useful in a network engineering environment where large amounts of incident data are generated every week.
- Automating reports like this helps identify:
    - Sites with recurring outages
    - Devices with high failure rates
    - Weekly costs and impact trends

Examples of real-world situations where this type of automation would be valuable:

- Real-world uses include:
    - Generating automatic weekly summaries for network operations teams
    - Prioritizing hardware replacements based on incident frequency and cost
    - Feeding analysis results into dashboards or ticketing systems for proactive monitoring
- This kind of automation can save time, reduce manual errors and improve decision-making in network maintenance and planning.

4. Code Quality and Improvements

If you were to do the workshop part again, what would you do differently?

- I would split the code into separate functions for reading, processing, and writing data.
- Add logging to capture data issues and runtime errors more transparently.
- Introduce visualizations (e.g., graphs for weekly costs or severity distribution) for easier interpretation.

Which parts of your solution are you most satisfied with and why?

- I'm most satisfied with the site and device summaries which give clear insight into problem areas and costs.
- The automation of generating multiple output files everything runs in one go.
- The way the script handles messy data without crashing, which is realistic for network logs.