Ticket Prioritization and Service Level Agreement (SLA) Dashboard

Fernando Djingga

10 July 2025

Introduction

I created a lightweight ticket prioritization and Service Level Agreement (SLA) dashboard that simulates how IT/Application Support teams handle incoming issues. My program processes raw support tickets containing severity, SLA deadlines, and timestamps, and then applies a set of prioritization rules to determine which tickets should be addressed first.

My project produces two outputs per run:

1. tickets_prioritized.csv – a clean, machine-readable CSV file for spreadsheets, ticketing and BI systems or tools
2. dashboard.xlsx – an Excel dashboard sorted by urgency for human triage and assignment

Using just pandas and openpyxl, my Python program automates what is typically a slow and error-prone manual triage process. This delivers practical value by ensuring SLA deadlines are respected and high-severity issues are escalated quickly.

High-Level Implementation

My program follows a simple and straightforward pipeline. Firstly, I will load tickets in a file containing all the information of the ticket. Each ticket is then assigned a numerical rank value for both severity and SLA. I set the weight of the rank value to be a 70 to 30 ratio between severity and SLA in the following order. For severity, the order I set is Critical > High > Medium > Low. For SLA, the order I set is 1h > 4h > 1d > 3d > 7d (as the SLA deadline). Then, my program sort the tickets by the combined score and then use creation timestamp for tie-breaking. Finally, my program records the final results as both a CSV file and a Excel file. The idea is to simulate an effective method to help support teams triage tickets under a deadline based on a reasonable prioritization algorithm. My workflow is load → prioritize/rank → save.

Coding Implementation

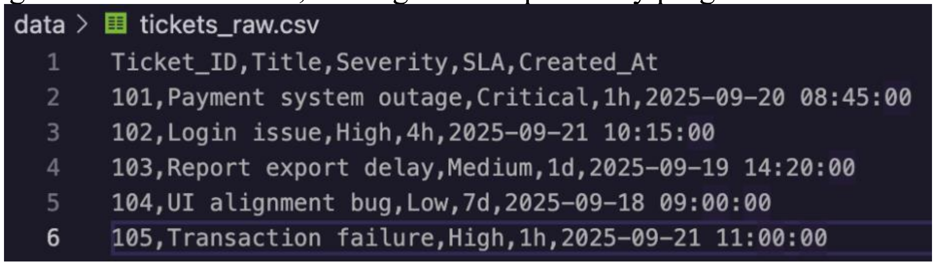
My code is structured in a single file ‘dashboard.py’, where ‘load_tickets(filepath)’ reads raw CSV into a pandas ‘DataFrame’ and handles the missing files accordingly. I implemented ‘apply_priority_rules(df)’ to map the severity and SLA values into numeric ranks using dictionaries (SEVERITY_MAP, and SLA_MAP). A priority score is also calculated as a weight average. The ‘DataFrame’ is then sorted and assigned the order by ‘Priority_Order’. Finally, ‘save_prioritized_data(df)’ saves the results to both CSV ‘ticket_prioritized.csv’ and an Excel dashboard ‘dashboard.xlsx’ using ‘openpyxl’.

The structure of my program is scalable and modular, meaning new severity or SLA categories may be added by updating the dictionaries.

Project Results

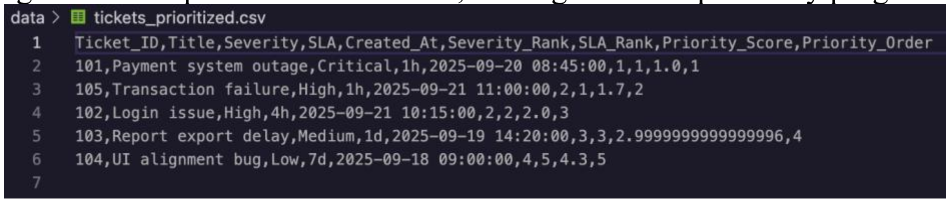
The program was tested on a sample dataset of 5 sample tickets containing mixed severities and SLA deadlines.

The following is the raw ticket data, serving as the input of my program.



```
data > tickets_raw.csv
Ticket_ID,Title,Severity,SLA,Created_At
101,Payment system outage,Critical,1h,2025-09-20 08:45:00
102,Login issue,High,4h,2025-09-21 10:15:00
103,Report export delay,Medium,1d,2025-09-19 14:20:00
104,UI alignment bug,Low,7d,2025-09-18 09:00:00
105,Transaction failure,High,1h,2025-09-21 11:00:00
```

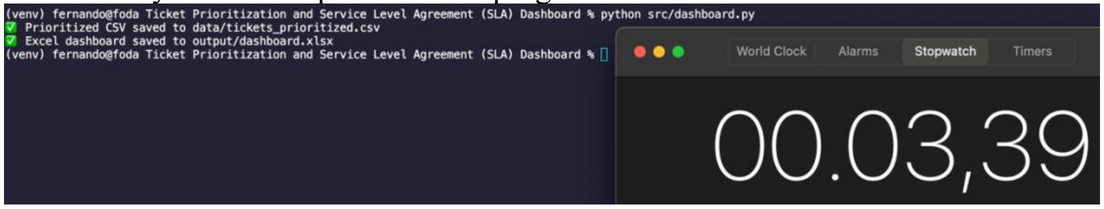
The following is the final prioritized ticket data, serving as the output of my program.



```
data > tickets_prioritized.csv
Ticket_ID,Title,Severity,SLA,Created_At,Severity_Rank,SLA_Rank,Priority_Score,Priority_Order
101,Payment system outage,Critical,1h,2025-09-20 08:45:00,1,1,1.0,1
105,Transaction failure,High,1h,2025-09-21 11:00:00,2,1,1.7,2
102,Login issue,High,4h,2025-09-21 10:15:00,2,2,2.0,3
103,Report export delay,Medium,1d,2025-09-19 14:20:00,3,3,2.9999999999999996,4
104,UI alignment bug,Low,7d,2025-09-18 09:00:00,4,5,4.3,5
```

A typical manual prioritization time for 5 tickets would take about 2 minutes (120 seconds).

The runtime of my automated prioritization program took 3.39 seconds.



```
(venv) fernando@foda Ticket Prioritization and Service Level Agreement (SLA) Dashboard % python src/dashboard.py
✅ Prioritized CSV saved to data/tickets_prioritized.csv
✅ Excel dashboard saved to output/dashboard.xlsx
(venv) fernando@foda Ticket Prioritization and Service Level Agreement (SLA) Dashboard % 
```

$$120 - 3.39 = 116.61$$
$$116.61 \div 120 \approx 97\%$$

My program increased the time taken by at least 97%. Practically, it would be much more difficult to prioritize a much larger amount of tickets than just 5, so the efficiency would probably be significantly greater at a larger scale. Manual triages is also subject to error, whereas an automated triage maintains 100% correctness.

Conclusion and Project Significance

In conclusion, my ticket prioritization and SLA dashboard project successfully automated a critical support function, which is to identify the most urgent tickets. My program reduced manual triage time from 120 seconds to 3.39 seconds, representing about a 97% improvement in efficiency. It also eliminated inconsistency, which ensured that severity and SLA rules were applied uniformly across all tickets. For IT, application support or finance teams, my tool saves time, improves accuracy, and allows for scalability, and documentation.

As a fresh graduate, I may not yet have practical workplace experience, but my automated project provides an attempt to improve operations by eliminating prioritization errors in the many ticket backlogs that support teams face. It reduces human workload, and improves SLA compliance.