Case Study: Support Ticket Triage with n8n and Gemini

By Olusegun Marcus · Built and tested on public data (BANKING77 banking support tickets) · Run date: 24 October 2026

Summary

I built a free, self-hosted n8n workflow that reads banking support tickets from a Google Sheet, sends each one to Gemini, and writes a category, a confidence and an urgency back to the same sheet. I then scored the category against the dataset's labels, so the figure below comes from a comparison with known answers.

Headline result: on 100 tickets, Gemini's category matched the dataset label on 96 of 100. All four misses are confusions between neighbouring categories. Confidence and urgency have no answer key, so I did not score them.

Problem

Support teams receive tickets in volume and have to route each one. I wanted to test how far a no-cost workflow could sort them, and how to measure whether its answers are accurate enough to trust.

Data

- Source: the public BANKING77 dataset of banking customer-service queries. Each ticket has a label naming its category.
- Sample: 100 tickets, 10 in each of 10 categories: card_arrival, lost_or_stolen_card, declined_card_payment, pending_transfer, transfer_not_received_by_recipient, cash_withdrawal_not_recognised, passcode_forgotten, verify_my_identity, request_refund and transaction_charged_twice.
- Labels were used exactly as published and were not edited after seeing results.

Workflow

Built in n8n running on my own computer (localhost), with Google Sheets and the Gemini API free tier:

- Manual trigger, then Google Sheets reads the 100 tickets.
- Edit Fields keeps only the ticket ID and the ticket text. The label column is dropped here, so the model never receives it.
- Loop Over Items, batch size 1: one ticket per request.
- Gemini (model gemini-3.1-flash-lite, JSON output) returns category, confidence and urgency. Retry on failure: up to 5 tries, 5 seconds apart.
- A short JavaScript step reads the JSON. If it cannot, it writes PARSE_ERROR so the failure is visible.
- Google Sheets updates the matching row by ticket ID, then a Wait node pauses 6 seconds to stay under the free rate limit.

The prompt lists the 10 category names with a one-line description of each, defines urgency (high if money is already lost or at risk, medium if a payment or transfer is stuck, low if it is a general question), and asks for JSON only. The full prompt is in the workflow file linked from my portfolio.

How I measured

- Category: a sheet formula marks a row "yes" if Gemini's category is identical to the dataset label, and "no" otherwise. This is deliberately strict.

- Confidence and urgency: not scored. They are Gemini's own labels, and the dataset has no answer for them.
- Checks: 100 rows filled, no PARSE_ERROR, and every answer one of the 10 allowed names.

Results

Measure	Result	How it was checked
Category matches the dataset label	96 / 100	Sheet formula against the dataset label
Answers outside the 10 categories	0	Checked on the results sheet
Confidence and urgency	Not scored	No answer key exists

Two runs. My first prompt contained a hint I wrote after reading ticket 95. That run scored 97 / 100 (misses: tickets 14, 25 and 26) and was interrupted at ticket 50 by a network error, so I resumed it from ticket 50. I then removed the hint, cleared the answers and re-ran all 100 tickets in one go. The 96 / 100 in this study is that second run.

The four misses

Ticket	Dataset label	Gemini said
25: "I am still waiting for a transaction to be completed"	transfer_not_received_by_recipient	pending_transfer
26: "I just lost my wallet and I see that they are already withdrawing money from my account."	cash_withdrawal_not_recognised	lost_or_stolen_card
55: "How long does it take for a transfer to get to a recipient?"	transfer_not_received_by_recipient	pending_transfer
95: "how long do money transfers take?"	transfer_not_received_by_recipient	pending_transfer

What I found

- 96 of 100 matched, with no unreadable or off-list answers.
- All four misses are between neighbouring categories, and in my reading each could be defended either way. That is my judgement, not a property of the dataset.
- Tickets 25, 55 and 95 ask how long a transfer takes or whether it has finished. The dataset labels them transfer_not_received_by_recipient and Gemini read them as pending_transfer.
- Removing the hint from the prompt cost one point (97 to 96), so the hint was helping on a few tickets. Reporting the neutral run keeps the result honest.

Limitations

- Small sample: 100 tickets and one scored run per prompt. I did not measure run-to-run variation.
- BANKING77 tickets are short single messages. 96 / 100 is not a general accuracy figure for support tickets.
- Confidence and urgency were not scored, and urgency follows a rule I wrote.
- The category descriptions in the prompt were written by me after I had seen a few of the tickets.
- Model, free quotas and limits are as they stood on 24 October 2026 and may change.
- Not production-ready. A person would need to review routing decisions.

Next steps

- Add a “how long does a transfer take” category or merge the two transfer categories, then re-test on tickets not used here. This is a hypothesis; I have not tested it.
- Run the same 100 tickets several times to measure run-to-run variation.
- Score urgency on a sample by hand, the way I did for the review project.
- Test on longer, messier tickets.

Method

I used an AI-assisted workflow design approach: I used Claude to plan the pipeline, debug errors and design the scoring checks, and I built, ran and verified every step myself against the dataset’s labels.