



FM

FLOWMATRIX.DEV

Premium Workflow

A practical guide – step by step

Customer Health Scoring

Combine product usage and support signals into a simple customer health score.

CUSTOMER SUCCESS

ADVANCED

ABOUT 3 HOURS SETUP

OVERVIEW

Customer Health Scoring

Combine product usage and support signals into a simple customer health score.

CATEGORY	DIFFICULTY	DURATION
Customer Success	Advanced	About 3 hours setup

What you use it for

Spot at-risk accounts before renewal conversations.

Tools in this workflow

PostHog Analytics API · OpenAI API · Supabase · Slack

How to implement it

1

Collect usage signals

Pull active users, feature usage and login frequency from PostHog.

WHAT YOU NEED TO DO

1. Create a PostHog read-only API key.
2. Collect weekly active users per account.
3. List key feature events per account.
4. Store the metrics in Supabase.

CHECK AT THE END

- ☐ Usage data covers all accounts
- ☐ Metrics are stored weekly

RESULT

A weekly usage dataset exists.

LINKS

PostHog API: <https://posthog.com/docs/api>

2

Add support signals

Add open tickets, response times and recent complaints to the score.

WHAT YOU NEED TO DO

1. Export open ticket counts per account.
2. Add average first response time.
3. Flag accounts with urgent complaints.
4. Join the data with usage metrics.

CHECK AT THE END

- ☐ Support data is matched by account
- ☐ Urgent complaints are flagged

RESULT

Usage and support signals are combined.

LINKS

PostHog Documentation: <https://posthog.com/docs>

3

Build the health score

Define a transparent score with weights for usage, support and payments.

WHAT YOU NEED TO DO

1. Assign a weight to each signal.
2. Normalise each metric from 0 to 100.
3. Calculate the weighted score.
4. Store the score and its inputs.

CHECK AT THE END

- ☐ Weights are documented

[] Every score is reproducible

RESULT

Each account has a transparent health score.

LINKS

Supabase Database Guide: <https://supabase.com/docs/guides/database>

4

Explain the score

Ask the model to write a short explanation for the customer success team.

WHAT YOU NEED TO DO

1. Send score drivers to the OpenAI API.
2. Ask for a two-sentence explanation.
3. Avoid recommendations without data.
4. Store the explanation with the score.

CHECK AT THE END

- ☐ Explanation cites score drivers
- ☐ No unsupported claims are added

RESULT

The team understands why an account is at risk.

LINKS

OpenAI Text Generation: <https://platform.openai.com/docs/guides/text-generation>

5

Alert the owner

Notify the account owner when an important account drops below the threshold.

WHAT YOU NEED TO DO

1. Define a threshold for at-risk accounts.
2. Post an alert in Slack for the owner.
3. Include the score and top drivers.
4. Create a follow-up task in Supabase.

CHECK AT THE END

- ☐ Alerts go to the right owner
- ☐ Follow-up task is created

RESULT

At-risk accounts get a timely follow-up.

LINKS

Slack chat.postMessage: <https://api.slack.com/methods/chat.postMessage>

RECOMMENDED TOOLS & RESOURCES

Recommended Tools & Resources

All tools in this workflow with their official website.

PostHog Analytics API

<https://posthog.com/>

OpenAI API

<https://platform.openai.com/?ref=flowmatrix>

Official URL

Supabase

<https://supabase.com/?ref=flowmatrix>

Official URL

Slack

<https://slack.com/>

Full version as Premium PDF

All steps, tools and resources as a PDF – €4.99.

[Buy now – €4.99](#)