

How to Build Your Own Instagram AI Analytics System

A step-by-step guide for creators who want to know their numbers

What You'll Have at the End

Tool 1: Analytics Dashboard

Your Instagram post metrics auto-synced to Google Sheets every morning at 8am — likes, reach, saves, plays, and more.

Tool 2: AI Reel Analyzer

An AI agent (powered by Gemini 2.5 Flash) that watches your Reels and gives you hook scores, transcripts, and specific recommendations.

Why This Matters

Most creators post and hope for the best. They have a gut feeling about what works but no real data to back it up. The result? Guessing. Repeating content that doesn't land. Missing what actually resonates.

When you know your numbers — which posts got the most saves, which Reels had the strongest hooks, which CTAs drove comments — you stop guessing and start making decisions. You create more of what works. You grow faster with less effort.

This guide builds you that system. No ongoing subscription costs. No black-box tools. Just your own data, in your own spreadsheet, analyzed by one of the best AI models in the world.

What You'll Need Before You Start

- **Instagram Creator or Business account** (personal accounts can't access the analytics API)
- **A Facebook Page connected to your Instagram** — this is required; more on this below
- A Mac (the scheduling step uses a Mac-specific tool; Windows users can skip to manual runs)
- A Google account
- **For Part 2 only:** a Dust account (dust.tt), a Google AI Studio account with billing enabled, and a Railway account (railway.app)

⚠ Critical: Instagram must be linked to a Facebook PAGE — not your personal profile

This is the #1 gotcha. Instagram can be linked to Facebook two ways: your personal profile (for sharing posts) or a Facebook Page (for API access). You need the Page connection. In

Instagram → Settings → Creator Tools → Connect to Facebook Page. If you don't have a Page, create a blank one at facebook.com/pages/create — it takes 2 minutes and doesn't need any content.

PART 1

Instagram Analytics → Google Sheets

Automatic daily sync of all your post metrics

Step 1 — Get Instagram API Access

Think of this like giving yourself a special key to read your own Instagram data. Meta runs a developer platform where you create a small app that acts as your key.

1a — Create a Meta Developer App

1. Go to developers.facebook.com and sign in with your Facebook account
2. Click My Apps → Create App → select Business as the type
3. Give it any name (e.g. "My Instagram Analytics") and click Create
4. On the next screen, find Instagram Graph API in the list and click Add Product

1b — Generate your API Token

5. Inside your app, go to Instagram Graph API → User Token Generator
6. Connect your Instagram Creator account when prompted
7. Make sure these 4 permissions are checked: **instagram_basic**, **instagram_manage_insights**, **pages_read_engagement**, **pages_show_list**
8. Click Generate Token and copy it — this is your short-lived token (valid 1 hour)

1c — Exchange for a Long-Lived Token (valid 60 days)

Run this command in your Mac terminal (replace the placeholders with your real values):

```
curl
"https://graph.facebook.com/v21.0/oauth/access_token?grant_type=fb_exchange_token&c
lient_id=YOUR_APP_ID&client_secret=YOUR_APP_SECRET&fb_exchange_token=YOUR_SHORT_TOK
EN"
```

Copy the `access_token` from the response. That's your long-lived token — save it somewhere safe.

1d — Find your Instagram Business Account ID

Your Instagram account has a special ID different from your username. Here's how to find it:

9. Go to facebook.com/pages, click your Page → About → find the Page ID (a long number at the bottom)
10. Run this in terminal:

```
curl  
"https://graph.facebook.com/v21.0/YOUR_PAGE_ID?fields=instagram_business_account&access_token=YOUR_TOKEN"
```

11. Copy the id value from the result — that's your Instagram Business Account ID


 **Tip**

Keep both your long-lived token and Instagram Business Account ID in a notes app. You'll need them again in Part 2 and every time you refresh your token.

Step 2 — Set Up Google Sheets API

You'll give a small robot (called a "service account") permission to write data into your spreadsheet.

12. Go to console.cloud.google.com → create a new project (any name is fine)
13. Go to APIs & Services → Library → search for and enable both Google Sheets API and Google Drive API
14. Go to APIs & Services → Credentials → Create Credentials → Service Account
15. Give it any name → click through to finish → open the service account you just created
16. Click the Keys tab → Add Key → JSON → download the file → rename it to `service_account.json`
17. Open your Google Sheet → click Share → paste the service account email from the JSON file (looks like `something@your-project.iam.gserviceaccount.com`) → give it Editor access
18. Copy your Sheet ID from the URL:
`docs.google.com/spreadsheets/d/THIS_PART_HERE/edit`

 **Don't skip the Share step**

If you forget to share the sheet with the service account email, the script will fail with a permissions error even though everything else is correct.

Step 3 — Set Up and Run the Python Script

3a — Create your project folder

```
mkdir ~/instagram-analyticscd ~/instagram-analytics
```

Create a file called `.env` in that folder:

```
INSTAGRAM_ACCESS_TOKEN=your_long_lived_token_hereINSTAGRAM_USER_ID=your_instagram_b  
usiness_account_idGOOGLE_SHEET_ID=your_sheet_idGOOGLE_SERVICE_ACCOUNT_FILE=service_  
account.json
```

Move your `service_account.json` into the `instagram-analytics` folder.

3b — Install dependencies and run

```
pip3 install requests gspread google-auth python-dotenvpython3 main.py
```

On the first run you should see: Done — 16 inserted, 0 updated. Check your Google Sheet — your posts are there!

Step 4 — Schedule Daily Runs (Mac)

This makes the script run automatically every morning at 8am without you touching anything.

19. Save the launchd plist file (provided with the scripts) to `~/Library/LaunchAgents/`
20. Run in Terminal:

```
launchctl load ~/Library/LaunchAgents/com.yourname.instagram-analytics.plist
```

21. Verify it loaded: `launchctl list | grep instagram`

Token reminder

Your Instagram token expires every 60 days. When it does, generate a new one using steps 1b–1c above and paste it into your `.env` file. Set a calendar reminder now!

What Lands in Your Google Sheet

Every post gets its own row with these columns:

post_id	timestamp	media_type	permalink
likes	comments	shares	saves
reach	impressions	video_views	plays
total_interactions	caption	last_updated	

Re-running the script updates existing rows rather than creating duplicates — so your data stays clean no matter how many times you run it.

PART 2

AI Reel Analyzer Agent

Gemini 2.5 Flash watches your Reels and tells you exactly what to improve

This tool goes beyond numbers. It watches the actual video of your Reel and gives you a structured breakdown: hook quality, transcript, visual pacing, CTA strength, and three specific things to improve next time.

Accounts you'll need:

- dust.tt — the AI agent platform
- aistudio.google.com — for your Gemini API key (billing required — more below)
- railway.app — free hosting for the analyzer server

Step 1 — Get Your Gemini API Key

22. Go to aistudio.google.com/apikey
23. Click Create API Key — copy and save the key (it starts with Alza...)
24. Enable billing on the linked Google Cloud project: console.cloud.google.com/billing → link a billing account to your project

⚠ Billing is required — but costs are tiny

Google AI Studio's free tier has a quota of 0 for Gemini 2.5 models used via API. You must link a billing account to your Google Cloud project. The good news: analyzing a Reel with Gemini 2.5 Flash costs roughly \$0.01–0.03. For personal use, your monthly bill will likely be under \$1.

Step 2 — Deploy the MCP Server to Railway

The MCP server is the bridge between your Instagram account and the Dust AI agent. We deploy it to Railway so it runs 24/7 in the cloud.

2a — Install Railway CLI and log in

```
brew install railway/railway/railway railway login
```

2b — Deploy from your project folder

```
railway init railway up
```

Set your environment variables:

```
railway variables set INSTAGRAM_ACCESS_TOKEN="your_token"railway variables set  
INSTAGRAM_USER_ID="your_ig_business_id"railway variables set  
GEMINI_API_KEY="your_gemini_key"railway variables set  
MCP_SECRET="any_password_you_choose"railway variables set PORT="8080"
```

Get your permanent public URL:

```
railway domain
```

You'll get something like <https://your-project.up.railway.app> — copy this URL.

💡 Why Railway instead of ngrok?

Railway gives you a permanent, always-on URL. ngrok (the alternative) gives you a temporary URL that changes every time you restart — meaning you'd have to update Dust constantly. Railway solves that permanently.

Step 3 — Connect to Dust

25. Log into dust.tt → Settings → Tools → + Add Tool → Remote MCP Server
26. Fill in — URL: your Railway URL | Authorization: **Bearer YOUR_MCP_SECRET**
27. Click Save — Dust automatically detects all 4 tools

The 4 tools Dust will see:

<code>list_reels</code>	Fetches your most recent Reels with ID, caption, and permalink
<code>download_reel</code>	Downloads a Reel and uploads it to Gemini for analysis
<code>analyze_reel</code>	Runs Gemini 2.5 Flash multimodal analysis on the Reel video
<code>get_reel_insights</code>	Fetches real engagement data: plays, reach, likes, shares, saves

Step 4 — Configure Your Dust Agent

In Dust, create or edit an agent and paste this as the system prompt:

```
You are an Instagram Reel performance analyst. When asked to analyze Reels: 1. Call list_reels → show the user the list 2. For each Reel to analyze: a. download_reel(media_id) → get gemini_file_uri b. analyze_reel(gemini_file_uri) → structured analysis c. get_reel_insights(media_id) → real engagement data 3. Present the analysis and metrics together. Highlight the top 3 improvements the creator should make.
```

What the agent returns for each Reel:

- **Hook score (1-10)** with a specific suggestion to strengthen the opening 3 seconds
- **Full transcript** of everything spoken in the video
- **Visual style breakdown** — lighting, colour palette, editing pace, B-roll usage
- **Audio analysis** — voiceover quality, background music, overall audio quality
- **CTA strength score (1-10)** with the exact CTA you used
- **3 actionable recommendations** specific to that Reel
- **Real metrics** — plays, reach, likes, shares, saves pulled live from Instagram

Keeping Everything Running

Refreshing Your Token (Every 60 Days)

Your Instagram token expires after 60 days. When it does, both tools stop working. Here's the 2-minute fix:

28. Go to developers.facebook.com → your app → Instagram Graph API → User Token Generator → generate a new token
29. Exchange for a long-lived token using the curl command from Step 1c
30. Update your `instagram-analytics/.env` file with the new token
31. Update Railway:

```
railway variables set INSTAGRAM_ACCESS_TOKEN="YOUR_NEW_TOKEN"
```

Set a calendar reminder

Put a recurring event every 55 days: "Refresh Instagram API token." It takes 2 minutes and saves you from a broken setup.

Troubleshooting Common Issues

Problem	Fix
HTTP 400 error	Your token has expired. Refresh it using the steps above.
me/accounts returns empty	Instagram is linked to your personal Facebook profile, not a Page. Go to Instagram → Creator Tools → Connect to Facebook Page.
Google Sheets permission error	You forgot to share the sheet with the service account email. Open the sheet → Share → add the service account email as Editor.
Dust agent can't reach server	Your Railway deployment may have stopped. Run railway up again from your project folder.
Model not found (404)	The Gemini preview model in your <code>server.py</code> has been retired. Open <code>server.py</code> , find the model name, and change it to <code>gemini-2.5-flash</code> . Then redeploy with railway up.
429 quota exceeded / limit: 0	Billing is not enabled on your Google Cloud project. Go to console.cloud.google.com/billing and link a billing account. Costs are typically under \$1/month for personal use.

Gemini file URI not found	Gemini File API files expire after 48 hours. Re-run <code>download_reel</code> to upload the video again and get a fresh URI, then re-run <code>analyze_reel</code> .
No media_url for Reel	Make sure your token has <code>instagram_basic</code> and <code>instagram_manage_insights</code> permissions enabled in the Meta Developer Console.

You built this. Now use it.

You now have an analytics dashboard that runs itself every day and an AI analyst that can review any Reel on demand. Most creators will never have this. You do.

The next step is simple: look at your data, run your last 5 Reels through the analyzer, and find the one pattern that shows up across your top performers. Then make three more of that.

— **Content Kuba**