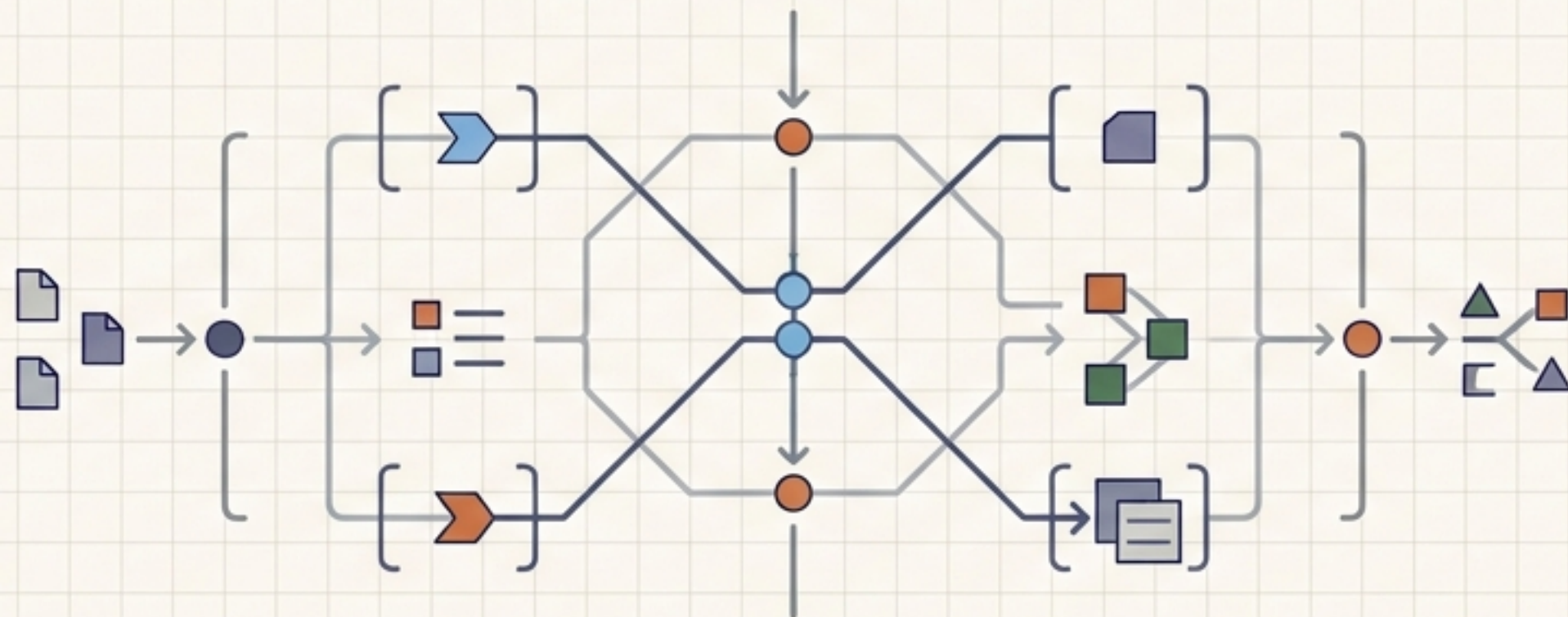


چگونه تحلیل‌گران و دانشمندان داده می‌توانند بیش از ۱۰ ساعت در هفته با هوش مصنوعی مصنوعی زمان ذخیره کنند؟

از کپی-پیست کردن‌های خسته‌کننده تا اتوماسیون یکپارچه در محیط کدنویسی.





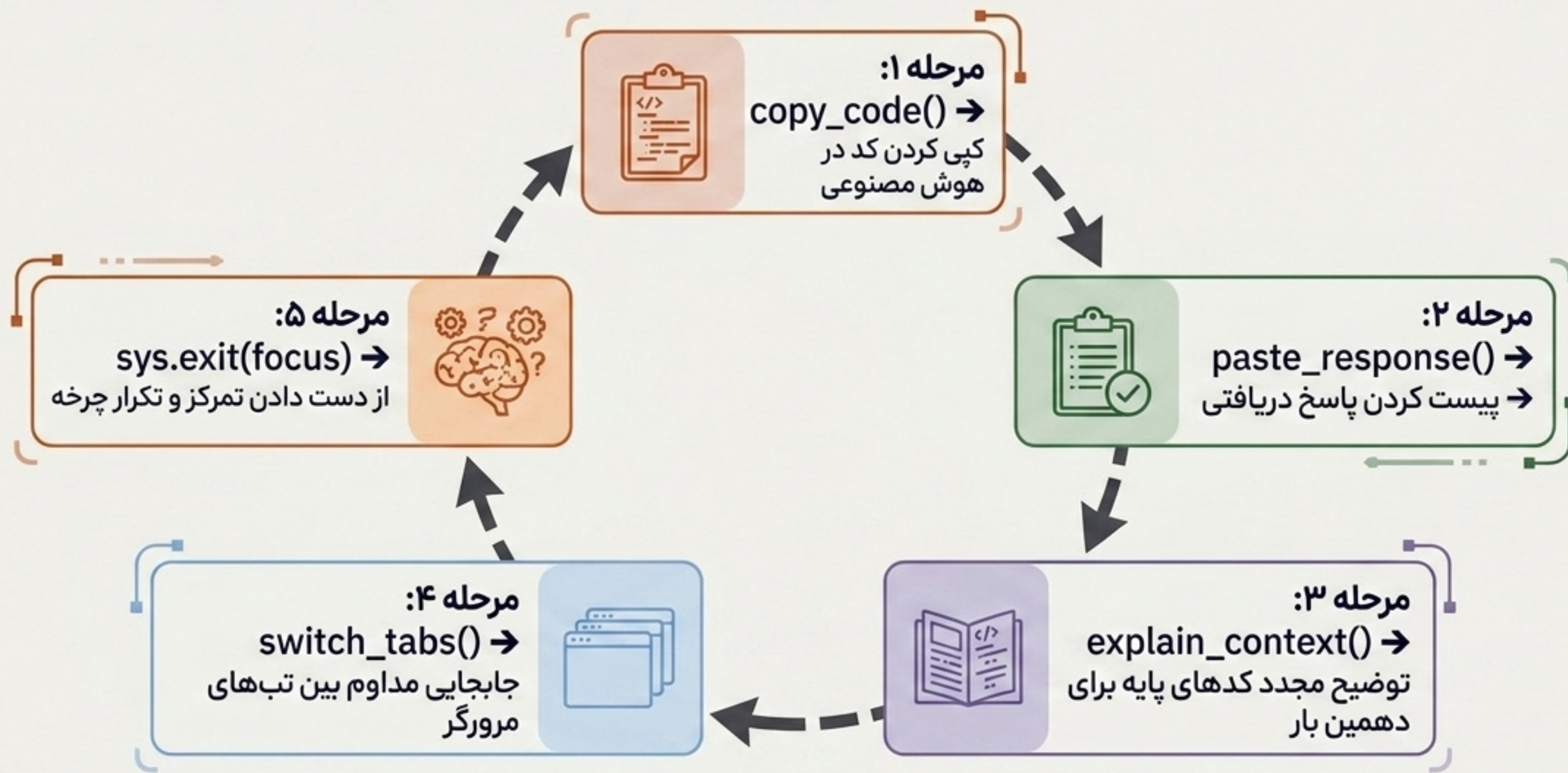
نادیده گرفتن هوش مصنوعی دیگر یک گزینه نیست!

اگر شما یک تحلیل‌گر داده (Data Analyst) یا دانشمند داده (Data Scientist) هستید، استفاده از هوش مصنوعی یک الزام است.

```
// ERROR_DETECTED
```

اما واقعیت تلخ این است: اکثر افراد در حال استفاده اشتباه از آن هستند.

چرخه معیوب کار با هوش مصنوعی (روش اشتباه)



جریمه از دست دادن بستر (The Context Penalty)



دقیقه اتلاف وقت پنهان در هر بار مراجعه

در حین کار روی یک پایپ‌لاین داده (Data Pipeline)، برای کمک گرفتن به هوش مصنوعی مراجعه می‌کنید... و سپس ۱۰ دقیقه کامل را صرف توضیح دادن کانتکست و فضایی می‌کنید که هوش مصنوعی قاعدتاً باید از قبل بداند.

تغییر زاویه دید

“

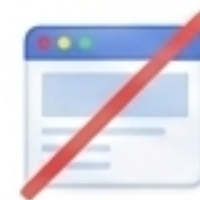
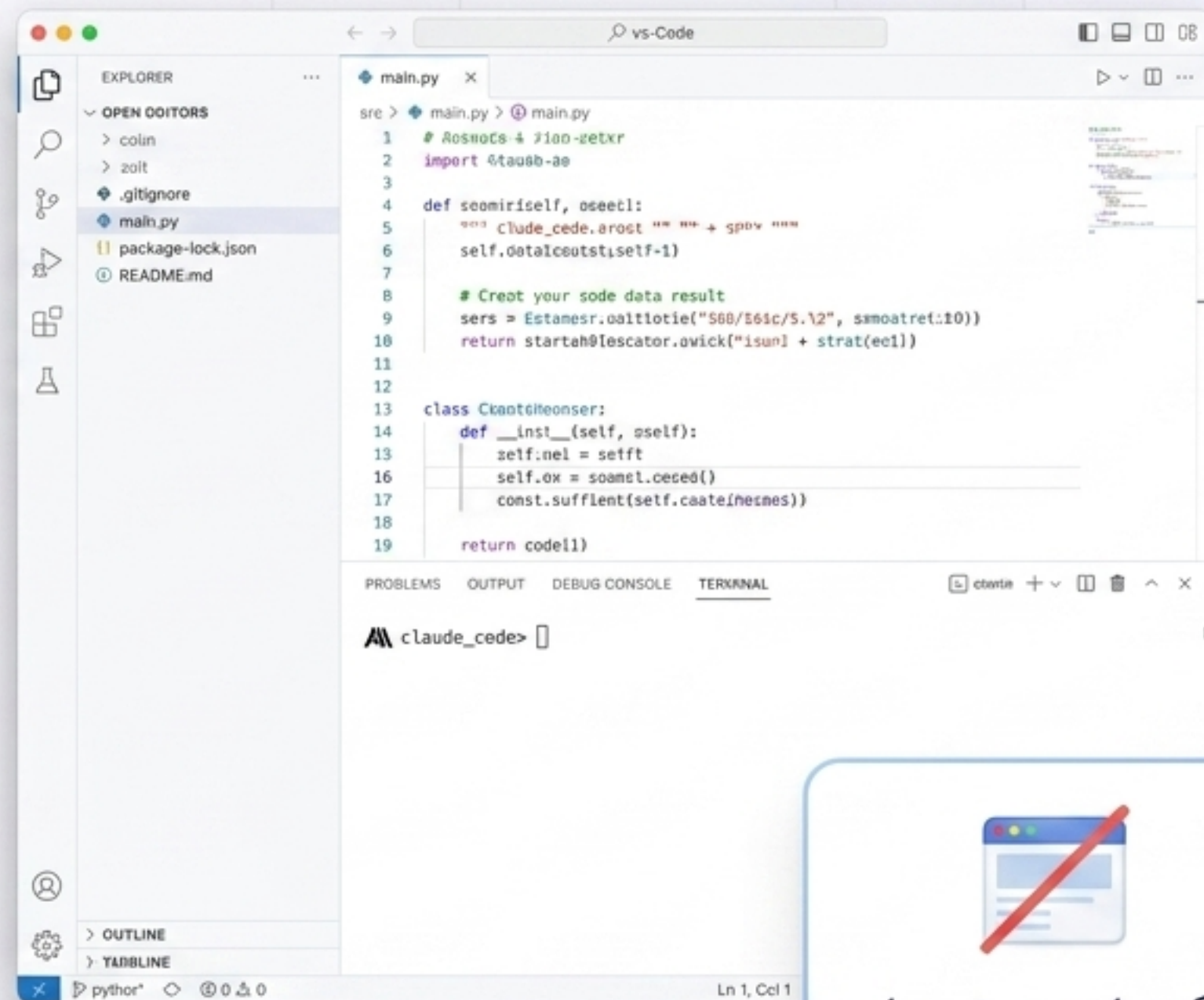
**مشکل از هوش مصنوعی نیست.
مشکل از جریان کاری (Workflow)
شماست.**

تا زمانی که هوش مصنوعی از محیط کاری شما جدا باشد، شما به جای حل مسئله، در حال مدیریت ابزارها هستید.

راه حل نهایی >_

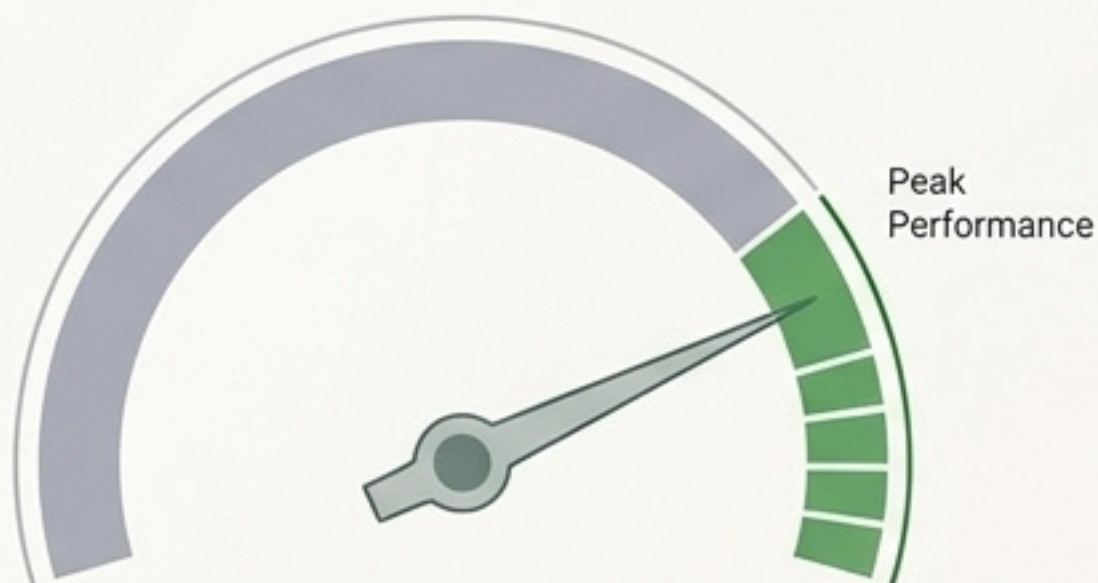
معرفی Claude Code

دستیار هوشمندی که مستقیماً در ترمینال (Terminal) و محیط توسعه (IDE) شما زندگی می‌کند.

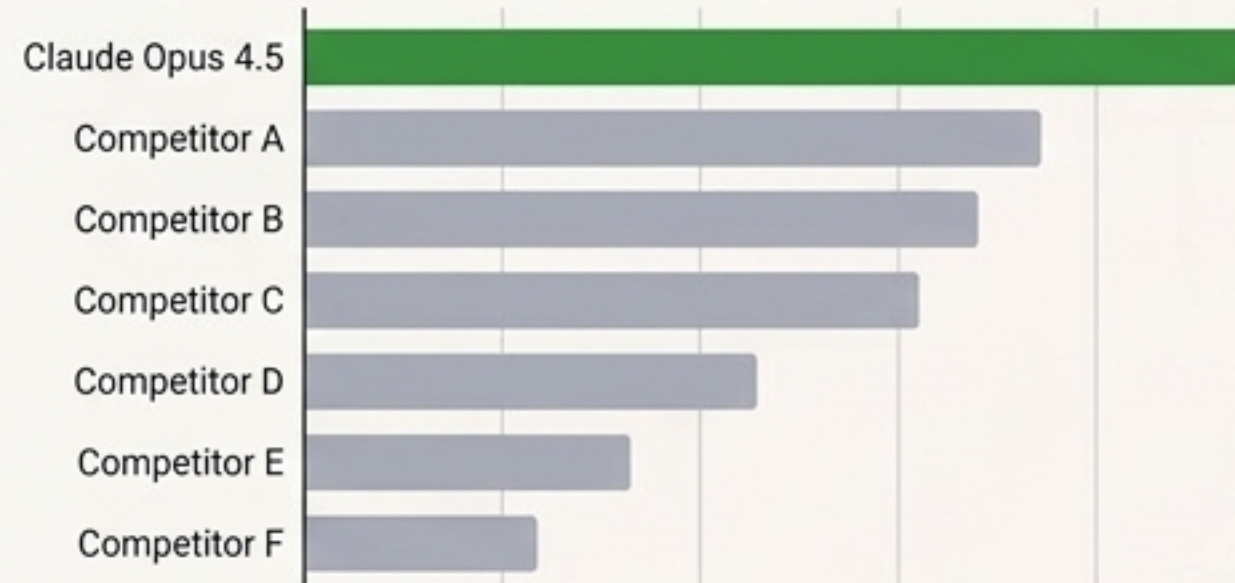


دیگر نیازی به تب‌های
مرورگر نیست.

موتور محرک: Claude Opus 4.5



قدرت گرفته از: مدل زبانی Claude Opus 4.5



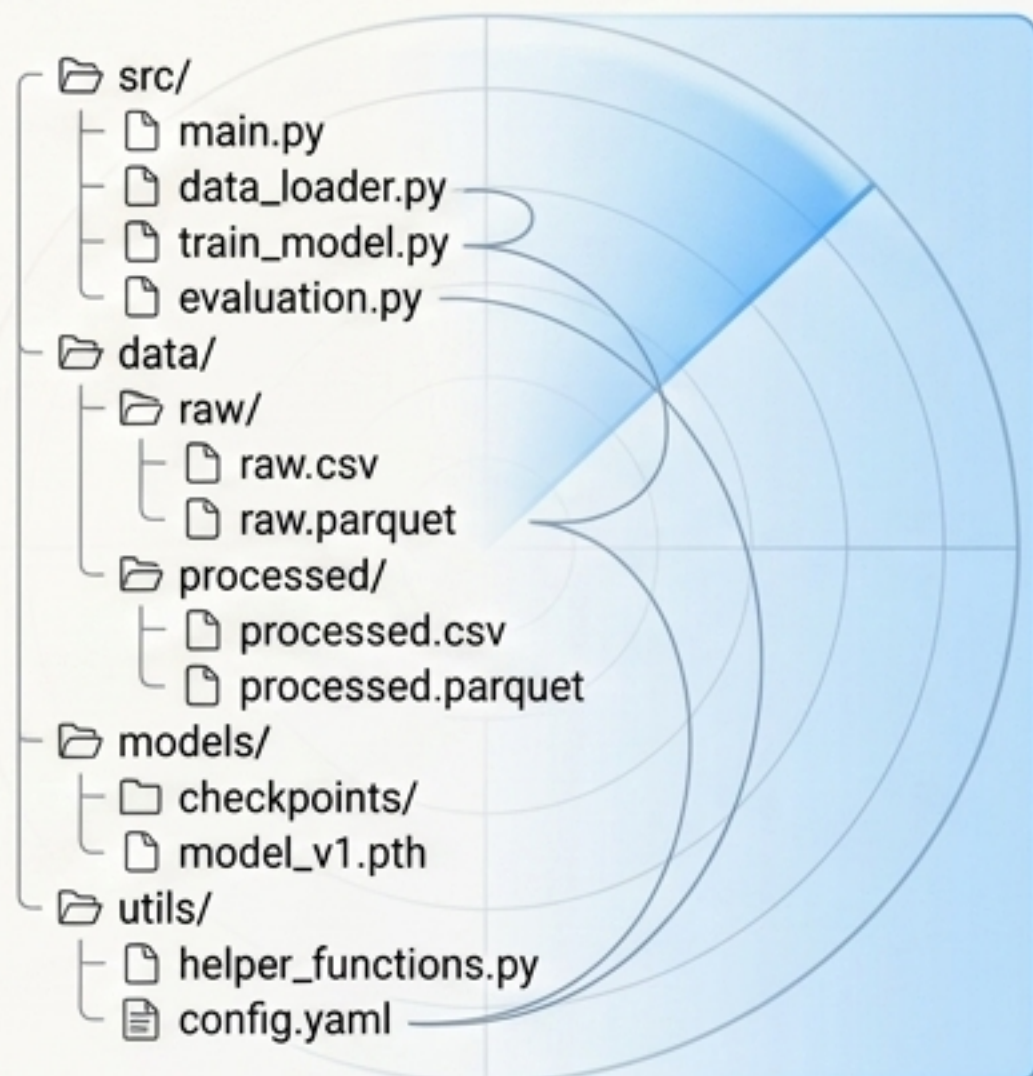
عملکرد: پیشرفته‌ترین حالت ممکن (State-of-the-art)
در بنچمارک SWE-bench.

> System Analysis: هوش مصنوعی که نه تنها چت می‌کند، بلکه مانند یک مهندس ارشد داده کدهای شما را درک و اجرا می‌کند.

ویژگی اول: آگاهی کامل از کدهای پروژه (Full Codebase Awareness)

دستیار شما دیگر محدود به یک فایل متنی ساده نیست.

PROJECT ARCHITECTURE VIEW



KEY BENEFITS

✓ درک کامل معماری و ساختار پوشه‌های پروژه.

✓ فهم دقیق ارتباط بین فایل‌ها و ماژول‌های مختلف.

✓ حذف کامل نیاز به کپی‌پیست کردن توابع از فایل‌های دیگر به عنوان بستر (Context).

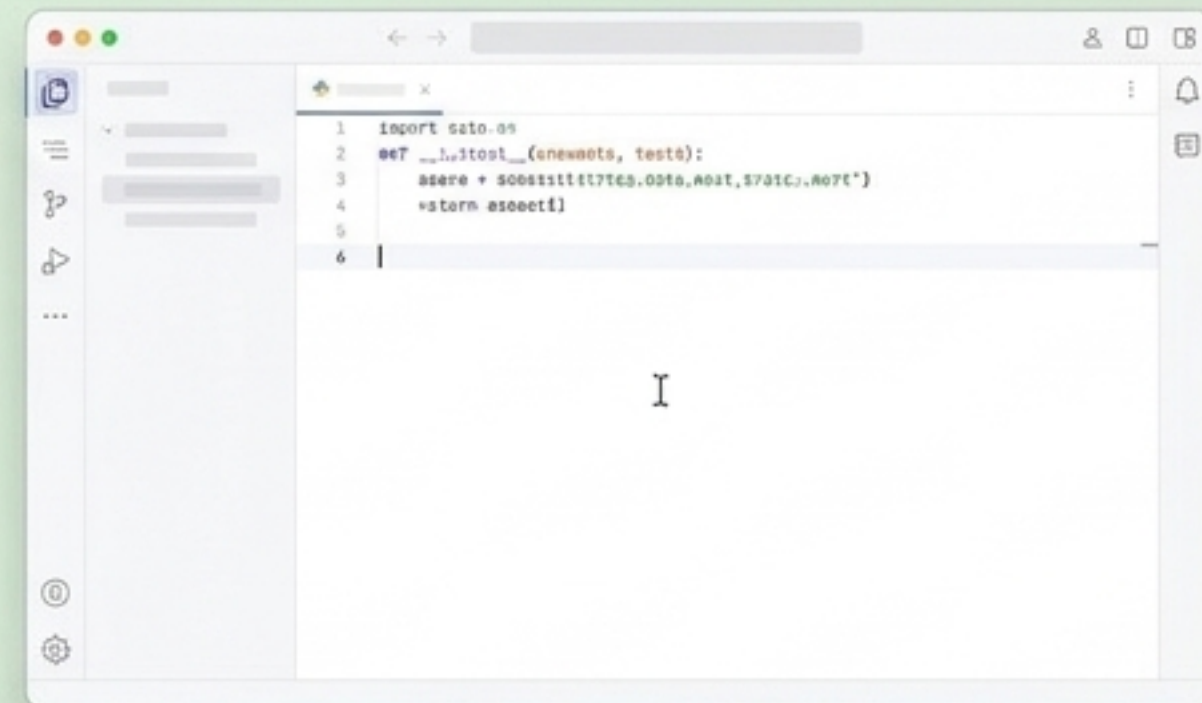
ویژگی دوم: بدون نیاز به جابجایی (Zero Context Switching)

ابزاری که دقیقاً همان جایی کار می‌کند که شما کار می‌کنید.

روش قدیمی (The Old Way)



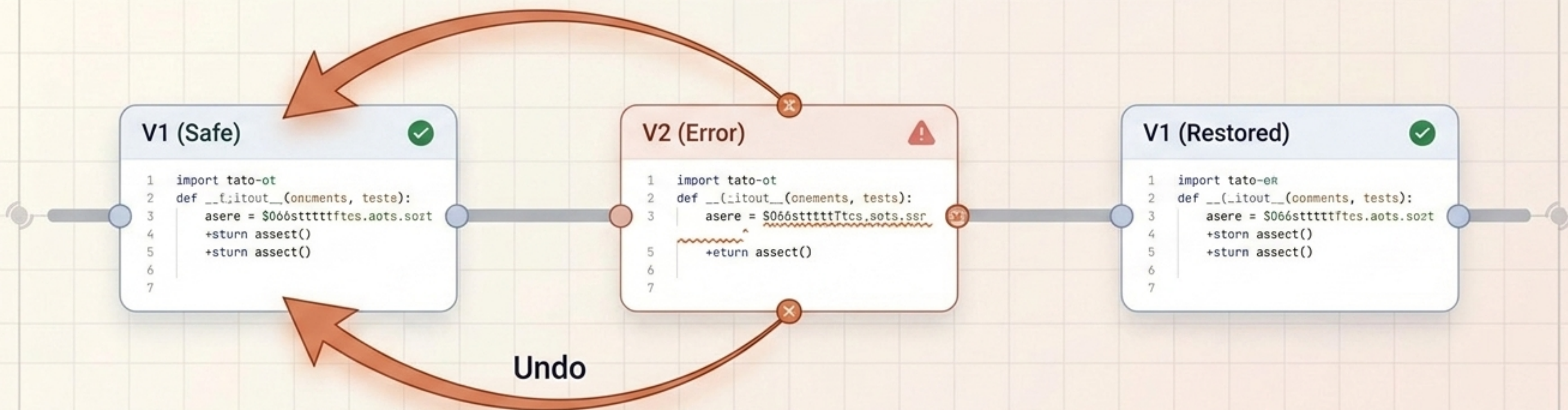
روش جدید (The New Way)



- ✓ بدون نیاز به باز کردن تب‌های مرورگر.
- ✓ حفظ ۱۰۰٪ تمرکز (Flow state) در حین توسعه و تحلیل داده.
- ✓ ارتباط مستقیم و زنده با فایل‌های محلی روی سیستم شما.

ویژگی سوم: چک‌پوینت‌های فوری (Instant Rollbacks)

با خیال راحت کد بزنید، تست کنید و از ریسک نترسید.



سیستم ایجاد نقطه ذخیره
(Checkpoints) کاملاً خودکار.

اگر چیزی خراب شد یا کد اشتباهی تولید
شد، به صورت فوری به حالت قبل
بازگردید.

ریسک اجرای کدهای تولید شده توسط
هوش مصنوعی در محیط محلی به
صفر می‌رسد.

۷ کاربرد طلایی Claude Code برای دانشمندان داده



کارهایی که ساعت‌ها طول می‌کشید، حالا در چند دقیقه انجام می‌شود.

۱. پاک‌سازی خودکار داده‌ها



[ACTION_REQUIRED]

- مدیریت و پر کردن مقادیر گم‌شده (Missing values).
- اصلاح و یکپارچه‌سازی انواع داده‌ها (Data types).
- حذف داده‌های تکراری (Duplicates) تنها در چند ثانیه.



۱ تا ۲ ساعت صرفه‌جویی

به ازای هر دیتاست (Dataset).

۲. نوشتن و دیباگ کوئری‌های SQL

```
SELECT
  table,
  table, name
JOIN sample_and_instr table
  FOR [key_conId] = new_table.mxId
JOIN sample_one_rate
  OVER (deba = [key_conId] * [conId,
    value, montly)
FROM table;

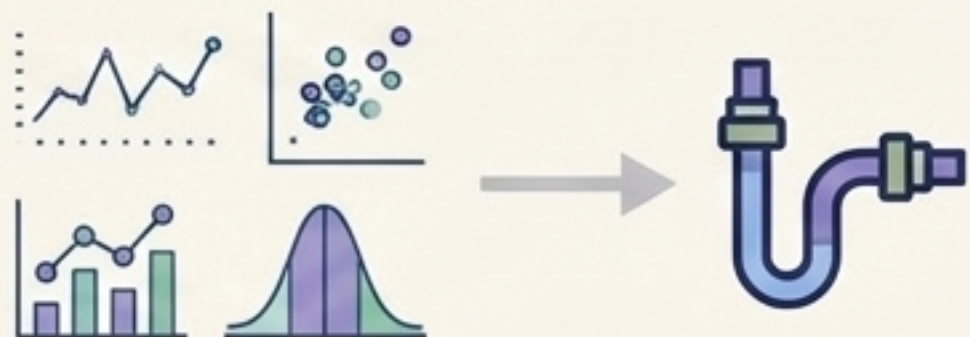
[ACTION_REQUIRED]
```

- مدیریت جوین‌های پیچیده (Complex joins).
- نوشتن عبارات جدول مشترک (CTEs).
- طراحی توابع پنجره‌ای (Window functions).
- توضیح کامل کوئری و بهینه‌سازی عملکرد (Optimization) آن.



بیش از ۳۰ دقیقه صرفه‌جویی
به ازای هر کوئری پیچیده.

۳. ساخت فوری پایپ‌لاین‌های EDA



[ACTION_REQUIRED]

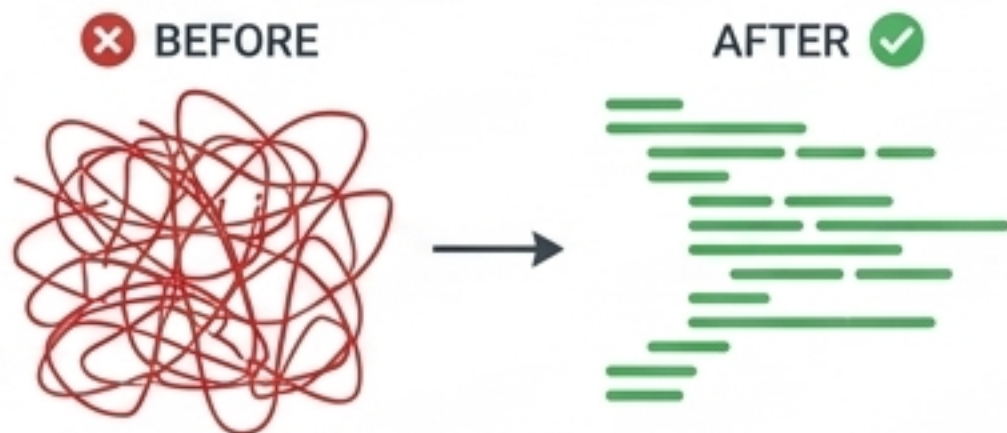
- رسم توزیع‌ها (Distributions).
- محاسبه و نمایش همبستگی‌ها (Correlations).
- تشخیص داده‌های پرت و حرید (Outlier detection).
- اجرای تمام این موارد به صورت یکجا و همزمان (All in one go).



۱ تا ۲ ساعت صرفه‌جویی

در کارهای اکتشافی (Exploratory work).

۴. توضیح و بازنویسی کدهای قدیمی (Legacy Code)



[ACTION_REQUIRED]

- درک سریع پایپ‌لاین‌های شلوغ و نامرتب نوشته شده توسط دیگران.
- پاک‌سازی (Clean up) و بازنویسی کدها بر اساس استانداردهای روز.
- حذف نیاز به خواندن خط به خط کدهای مستند نشده.

ROI



ساعات صرفه‌جویی

برای فرآیند مهندسی معکوس
(Reverse-engineering).

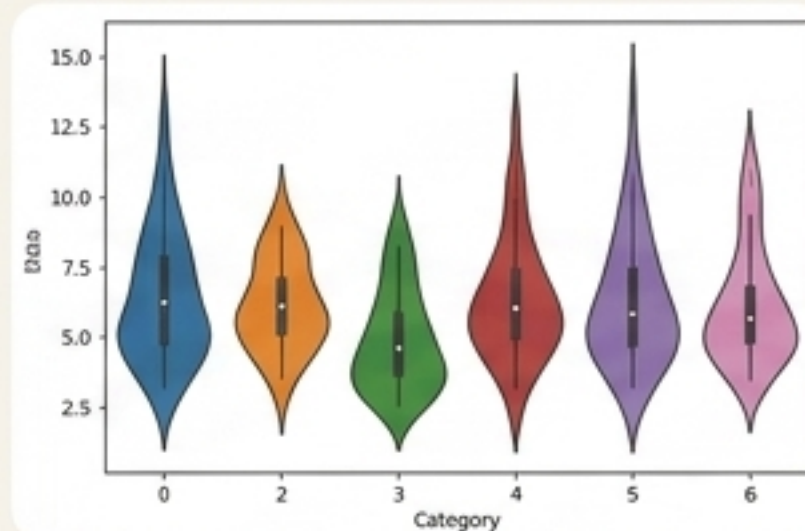
۵. تولید سریع مصورسازی‌ها

Matplotlib —

Seaborn —

Plotly —

[ACTION_REQUIRED]



- کدنویسی سریع برای کتابخانه‌های Matplotlib, Seaborn, Plotly.
- امکان تکرار و تغییر (Iterate) سریع ظاهر نمودارها با دستورات متنی.
- حذف کامل نیاز به جستجو در گوگل برای پیدا کردن کردن سینتکس‌های فراموش شده.

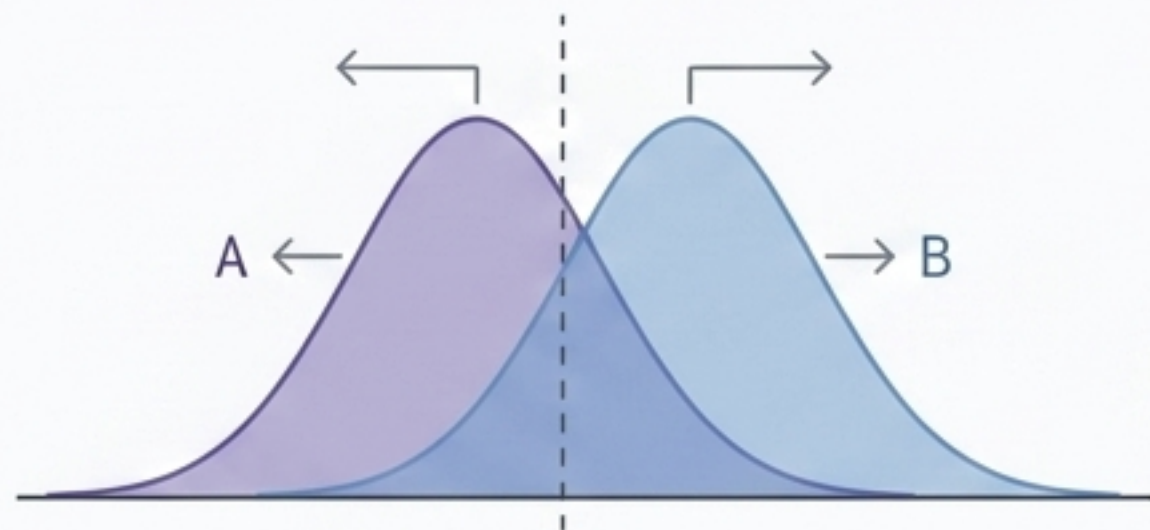
ROI



**بیش از ۴۵ دقیقه
صرفه‌جویی**

برای طراحی هر نمودار
حرفه‌ای.

۶. طراحی فریم‌ورک‌های تست A/B



[ACTION_REQUIRED]

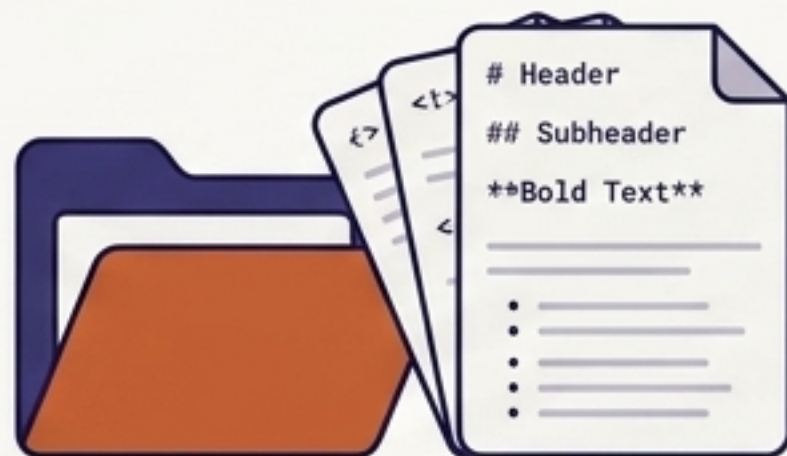
- تعیین دقیق اندازه نمونه (Sample size).
- محاسبه مدت زمان مورد نیاز برای تست (Duration).
- تعریف معیارهای موفقیت (Success metrics).
- تضمین اعتبار آماری (Statistically sound) تمامی مراحل.

ROI



ساعت‌ها صرفه‌جویی
در مرحله برنامه‌ریزی و طراحی تست.

۷. مستندسازی خودکار کارها



[ACTION_REQUIRED]

- تولید خودکار داک سترینگ‌ها (Docstrings) برای توابع.
- نوشتن فایل‌های README جامع برای پروژه‌ها.
- درج کامنت‌های درون خطی (Inline comments) به صورت هوشمند و بدون دخالت دست.



صرفه‌جویی در انرژی روانی
انجام وظیفه‌ای که همیشه از آن
فرار می‌کنید!

سنتز نهایی: مقایسه جریان‌های کاری

ویژگی	روش سنتی (تب مرورگر)	روش نوین (Claude Code)
آگاهی از پروژه	✗ محدود به متن کپی شده	✓ دسترسی به کل کد (Codebase)
تمرکز (Flow)	✗ پرش مداوم بین پنجره‌ها	✓ یکپارچه در IDE و ترمینال
امنیت تغییرات	✗ کپی دستی و ریسک بالای خطا	✓ چک‌پوینت و Rollback فوری
سرعت اجرای تسک	✗ ساعت‌ها درگیری ذهنی	✓ تنها چند دقیقه زمان اجرا

از ساعت‌ها به دقیقه‌ها...

هر یک از این ۷ وظیفه پیش از این
ساعت‌ها زمان می‌برد.
اکنون تنها در چند دقیقه انجام می‌شود.

با اصلاح جریان کاری و انتقال هوش مصنوعی به محیط توسعه (IDE)، بیش از
۱۰ ساعت در هفته زمان ارزشمند خود را پس بگیرید.

`sudo >` تبدیل ترمینال به قوی‌ترین دستیار داده.