NEXCREST · SMS LAND ACQUISITION · ANALYSIS

Dashboard Rebuild and Campaign ROI Analysis

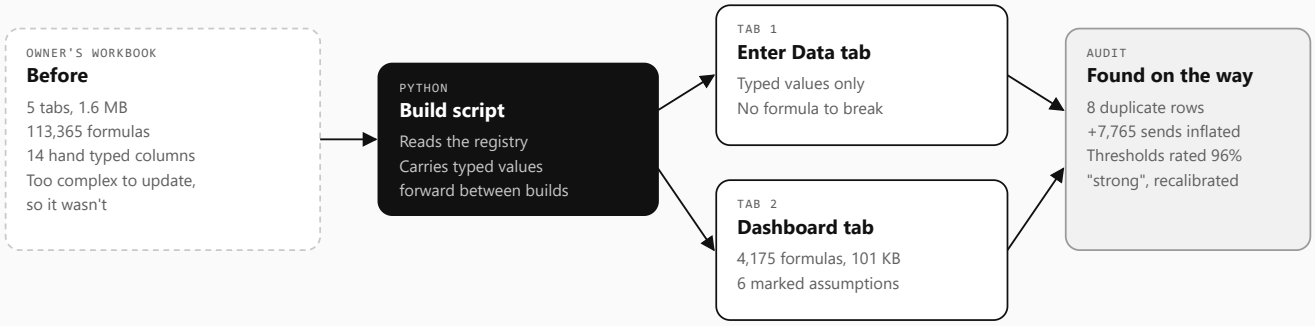
A workbook the owner can actually keep updated, and the analysis that showed cheap leads were the wrong target.

Built with Claude Code · Excel, Python, openpyxl, statistics

113k → 4k	1.6 MB → 101 KB	8	2.8x
FORMULAS	FILE SIZE	DUPLICATE ROWS CAUGHT	BETTER COST PER CONTRACT

HOW IT WORKS

Dashboard rebuild: 113,365 formulas down to 4,175



Excel · Python · openpyxl · Data audit · Data analysis · Statistics · Dashboard design · Claude Code

01 The workbook problem

The owner's SMS campaign dashboard had 5 tabs, 113,365 formulas and 14 columns typed by hand. It was 1.6 MB and so easy to break that it stopped being updated.

02 The rebuild

Enter Data tab
Only typed values. There is no formula on this tab, so there is nothing to overwrite by accident.

Dashboard tab
Every formula lives here, plus six blue assumption cells that are clearly marked as the only things to change.

Generated by script
The workbook is rebuilt from the campaign database with one command, carrying forward the hand entered columns that exist nowhere else.

Safe to share
No tables, macros, charts or dynamic array functions, so it imports into Google Sheets cleanly.

Smaller and faster
4,175 formulas and 101 KB instead of 113,365 formulas and 1.6 MB.

Refuses to lose data
The build stops if it cannot carry forward the hand entered values, unless forced.

03 What the rebuild found

Duplicate campaigns
Eight campaigns were entered twice under two spellings, once dated to the 1st and once to the true date, inflating totals by 7,765 sends.

Thresholds that judged nothing
The verdict thresholds rated 96% of campaigns strong and none weak, so the verdict column said Scale for almost everything. I recalibrated them from the real distribution.

04 Campaign ROI analysis

An interactive dashboard of cost per lead, cost per contract, ROAS, ROI and net, grouped by campaign, state or cohort, with a volume floor so tiny sends cannot top the ranking.

> The state with the cheapest leads converted only 4% of leads to contract, against 25% in another state.
>
> The pricier state was 2.8 times better on cost per contract, the number that actually pays.
>
> Budget decisions moved from cost per lead to cost per contract.

I also listed the four fields the data still needed for true ROI (send date, closed date with a Closed stage, gross profit per deal and SMS cost per campaign), most of which the registry and web app later supplied.

05 Market selection research

Counties were being picked by demand metrics such as sell through rate, on the assumption that hot markets produce more motivated sellers. I joined market data to the team's own campaign results and tested it.

> Demand correlated negatively with qualified leads per thousand texts.
>
> Recommendation: use demand as an exit screen to drop dead markets, not as a targeting score. The owner adopted it.

06 Tech stack

Build	Python and openpyxl, with Excel automation for recalculation
Analysis	Python, SQL on the campaign registry, an HTML and JavaScript dashboard
How I built it	Claude Code, with every figure traced back to a source row

---

Charles Lorenz Almenanza · Land Data Analyst and Claude Code builder · lorenzalmenanza@gmail.com · charles-almenanza-portfolio.pages.dev