

Where am I losing money right now, and can I trust Google's estimate of the fix? A brief for your AI assistant

Hand this whole file to your assistant along with the notebook from <https://www.stochasticstrata.com/blog/where-am-i-losing-money>. It tells the assistant what to do, what to refuse, and what to say before anyone raises a budget or a bid on the strength of a report or a simulator's number.

For the assistant: what you are being asked to do

The person you are helping runs paid search and wants two things: to know where their campaign is leaking (budget or bids) from the impression-share report they already have, and to know how far to trust the platform's own estimate of what a change would bring. You do the technical parts: get the export, map its columns, set the notebook's input cells, run it, and read the output back in plain words. Ask one question at a time. Do not paste code at them unless they ask.

Two things you must never do. Never price the rank leak from the report alone. The impressions a campaign is outbid in are reachable only with a higher bid, at a higher price than it pays now, and the report carries neither the count you would win nor the price. And never present the platform's projection as a forecast: it is a replay of the last seven days at a setting that was never run, with everyone else frozen. Both are measured below.

The two input files

A campaign export, one line per day. From the Google Ads interface: Campaigns page, add the competitive-metrics columns (Search impr. share, Search lost IS (budget), Search lost IS (rank)) next to impressions, clicks, cost and conversions, segment by day, filter to ONE campaign, download as CSV. The notebook reads that download as it comes (title lines, percentages, the Total row, a European download, an .xlsx) or a tidy file with the columns date, impressions, clicks, cost, conversions, search_impression_share, search_budget_lost_impression_share, search_rank_lost_impression_share and, optionally, daily_budget.

A projection log, one line per change the person acted on, kept by hand: change_date, what_changed, projected_clicks, projected_conversions, actual_clicks, actual_conversions over a window the same length as the simulator's, and optionally change_in_lookback (yes if something else changed in the week before the projection). Have them write the simulator's projected clicks and conversions down BEFORE they apply the change: the interface shows them in the simulators and they are gone afterwards.

From the API, the daily export is this query. Replace the campaign id. The notebook accepts the API's column names as they are. Running a query needs API access and a developer token, which a freelancer often lacks; the no-API route is the interface download above, or the Report Editor. Thirty days gives a steadier split; seven matches the simulator's own window.

```
-- One row per day for one campaign: the counts and the three impression-share metrics. Run it
-- through the Google Ads API (the Query Builder only writes the query; running it needs API access), save as CSV
-- with these column names. cost_micros is euros times a million; the notebook divides it.
SELECT
  segments.date,
  campaign.id,
  campaign.name,
  metrics.impressions,
  metrics.clicks,
  metrics.cost_micros,
  metrics.conversions,
  metrics.search_impression_share,
  metrics.search_budget_lost_impression_share,
  metrics.search_rank_lost_impression_share,
  campaign_budget.amount_micros
FROM campaign
WHERE segments.date DURING LAST_30_DAYS
  AND campaign.id = 1234567890 -- the campaign you are reading
  AND campaign.advertising_channel_type = 'SEARCH'
ORDER BY segments.date
```

Clamped values: the API returns 0.0999 for an impression share under ten per cent and 0.9001 for a lost share over ninety; the interface shows '< 10%' and '> 90%'. The notebook reads all four as what they are, a floor or a ceiling, and says so. Do not round them into numbers.

The arithmetic the report supports, and nothing more

With impressions I , impression share s , share lost to budget b and share lost to rank r ($s + b + r = 1$): the eligible pool is $E = I / s$; the impressions lost to budget are $E \times b$ and to rank $E \times r$; the win share when the campaign is entered into an auction is $w = s / (s + r)$. A bigger budget would have won $E \times b \times w$ impressions (the win share of the lost ones, not all of them) at about the usual click rate, conversion rate and price. Multiply by the click rate and the conversion rate for the conversions. The naive count, $E \times b \times \text{click rate} \times \text{conversion rate}$, is too big by one over w .

Measured: rebuilt from the formulas above alone, the split reproduces the notebook's pinned values on 8 of 8 quantities. On the worked campaign the eligible pool is 94,091, 29,234 impressions were lost to budget and 20,860 to rank, the win share is 67.8 per cent, the naive count says 50.5 conversions a week and the corrected count 34.3.

The budget rule, and its measured gap

The campaign was entered into $(1 - b)$ of its auctions. A budget of m times the current one enters it into $\min(1, m \times (1 - b))$ of them, so clicks, cost and conversions scale by that share divided by $(1 - b)$ until the budget stops binding at $m = 1 / (1 - b)$. Read the person's numbers into that rule for a quick answer, and tell them what it assumes: that the auctions the budget kept them out of look like the ones it let them into, and that last week's rates hold.

Measured against the notebook's replay on the worked campaign: the rule's knee is 1.4507 times the budget (the replay's 1.4508), and its extra conversions land -4.4 per cent from the replay's at the knee, 4.4 per cent at worst across the ladder. The difference is last week's luck in conversions, which the rule inherits. The replay found 35.9 extra conversions for 631 euros more spend, at 17.60 each, the same as the average before the lift.

The rank leak: do not price it from the report

The naive count of the rank leak (the rank-lost impressions at the usual rates) is 36.0 conversions 'at the usual cost' of 18.75 euros on the worked campaign. Measured against the replay with the budget lifted: at half again the bid the campaign gains 42.2 conversions (the naive count is -14.6 per cent off) at 30.63 euros each, 1.63 times the naive price (last week's reported cost per conversion), and 1.74 times the campaign's expected average of 17.60 euros; at twice the bid 54.5 conversions (-33.9 per cent) at 33.30 each, 1.78 times. The count is within reach at a high enough bid; the price is wrong at every bid, because the auctions gained are the expensive ones by definition.

And the order matters: with the budget left where it is, raising the bid half again LOWERS conversions from 79.5 to 66.4 a week on the worked campaign: the same budget buys fewer, dearer clicks. Resolve the budget leak first.

The platform's projection: what it is, and how to read it

Google's simulators, in Google's words, use data from the last seven days to estimate how the ads might have performed at a different bid or budget. That is a replay: last week re-run at the new setting with everyone else (demand, competitors, the person's own other changes) frozen at what they were. Read it as a description of last week, not a forecast of next week.

Measured on five hundred synthetic campaigns with a bid rise of thirty per cent: when the world holds still, the projection's clicks land within twenty per cent of the following week's on 89 per cent of campaigns and its conversions on 60 per cent. When demand rises a quarter the projection runs -17.6 per cent on clicks; when competition rises a fifth, +23.9; when a change sits inside the lookback, +13.2. Before reading a projection, check the person's change history for anything inside the simulator's dates. The API's simulation rows carry a

start_date and an end_date.

The conversion figure is the weakest number the simulator prints, and Google says so: conversions are few, so the rate learned from one week is noisy. On campaigns under fifty conversions a week, read the conversion projection as a range at least a fifth wide either way: the middle miss there was 20.0 per cent, and it was off by a third or more on 29 per cent of weeks. Score its clicks first.

The projection log, and its floors

Every time the person acts on a projection, log what it promised and what happened. The notebook scores the log, and refuses a trust reading below 8 logged changes for clicks and 30 for conversions. Below those, measured on the synthetic population's own errors, a typical error is mostly luck. It shows the per-change errors instead. On the shipped demo log of 12 changes it reads the click error and withholds the conversion one.

Honest caveats: say these before anyone acts

- The eligible pool is Google's estimate of the auctions the ad could compete in near its current bid; the rank-lost share is the auctions within reach, and the whole split moves when the bid does.
- While the budget binds, the rank-lost share understates how often the campaign is outbid. Lift the budget, let the shares settle, then read the rank leak.
- A clamped share is a floor or a ceiling, not a number. The headroom behind '> 90%' is at least what is shown, and the true share can sit anywhere above it.
- The corrected budget count assumes an even spread of the budget across the week. A budget that runs out at a fixed hour misses that hour's auctions, and the correction is approximate.
- A projection describes last week with everyone else frozen. Demand, competition and the person's own changes are outside it.
- Bids far from the one that ran are the weakest region of any replay; the logged data holds few auctions that speak for them.
- None of this says whether the ads work. It says where the campaign was kept out of the auction and what re-entering it would cost.

How to run the notebook

Local, private: install Python 3.11 or later with numpy, pandas, matplotlib and jupyter; open notebook.ipynb; set YOUR_CAMPAIN_EXPORT and YOUR_PROJECTION_LOG to the file paths, and YOUR_VALUE_PER_CONVERSION if the person knows it; run all cells. Nothing leaves the machine.

Colab, convenient: upload the notebook and the files to Google Colab and run all cells. Say plainly that this puts the person's export on Google's servers; for a campaign export that is usually acceptable, but it is their call.

Read the last section's output back to them in plain words: the split, the binding constraint, what a bigger budget would have won and at about what cost, the warnings, and the trust reading on their log if it clears the floors.

What to refuse

- Fewer than seven days with an impression share: the shares swing with the weekday and the platform updates them a day or two late.
- Any count or price for the rank leak from the report alone.
- A trust reading from a projection log below the floors.
- Reading the projection as a forecast, or reading it at all without checking the lookback for changes.

Synthetic worked campaign; the person's data will look different. The method transfers; the numbers do not.