The Localist Ranking Methodology

Version 2 — A transparent, locally calibrated algorithm for ranking businesses from public Google Business Profile data

Localist by Konigle
UEN: 201815303H,
#12-07, Suntec Tower One,
7 Temasek Boulevard,
Singapore 038987

White paper · May 2026

Contents

1. Executive summary

Localist publishes ranked directories of local businesses — one page per city and business vertical (for example, “Web development companies · New York”). Every ranking is computed from exactly three pieces of public information on a business’s Google Business Profile (GBP): its star rating, its review count, and the timestamps and ratings of its five most recent reviews.

This paper specifies the complete method, called the Localist v2 score. In one sentence: a business’s score is its local market’s average rating, moved toward its own rating in proportion to how much recent, verifiable review evidence it has.

$$v2 = C + F \cdot (v1 - C)$$

where *C* is the average rating of all businesses in the same city and vertical, *v1* is a credibility-weighted (Bayesian) average of the business’s own rating, and *F* is a freshness factor that discounts evidence when a business has stopped receiving reviews. Sections 4 and 5 define each term precisely; Section 7 walks through a full example market.

Three design commitments distinguish the method:

- **Local by construction.** Every parameter is calibrated per (vertical, city) pool. A business is measured strictly against its own local competitors — a New York web agency is never compared against restaurants, or against agencies in Austin.
- **Reader-recomputable.** Each directory page publishes the three pool parameters (C, m, g) alongside the ranking. Any visitor or business owner can verify their score with a calculator. Paid placements are visually separate and never affect organic positions.
- **Score, flags, and presentation are separate layers.** The score is deliberately simple. Integrity signals (review bursts, declining recent ratings, dormancy) appear as visible labels; they never silently adjust the number. What we show is what we computed.

2. Design principles

Five principles governed every design decision. They are worth stating because they explain why the method is simple where it could have been sophisticated.

- **P1 — Only public, Google-verified inputs.** The score uses nothing a business self-reports: no testimonials, no submitted case studies, no off-platform claims. If Google has not published it, it does not move the number.

- **P2 — Volume of evidence matters as much as the headline star.** A 5.0 average from one review tells you almost nothing; a 4.5 from 200 reviews tells you a lot. The score must encode this asymmetry.
- **P3 — Reputation is local.** Review behaviour differs enormously between verticals and cities. A restaurant with 25 reviews is brand new; an IT consultancy with 25 reviews is established. All calibration is therefore relative to the business's own peer pool.
- **P4 — Evidence expires.** A review from three years ago is weaker evidence about today's service than a review from last month. A business that stopped earning reviews should drift back toward "unknown", not keep a rank it earned in a different era.
- **P5 — Anyone can recompute it.** A ranking that cannot be audited is an assertion, not a measurement. The formula must be short enough to publish, and every input must be visible on the public GBP listing.

3. Why simpler approaches fail

3.1 Ranking by raw star rating

Sorting by the Google star average alone puts a 5.0-rated business with a single review above a 4.8-rated business with 900 reviews. It also produces mass ties: Google rounds displayed ratings to one decimal place, so a competitive page can contain a dozen businesses all showing 4.8. Raw stars violate principle P2 and give no way to break ties on merit.

3.2 The previous Localist formula (v1-2025)

Localist's first published method multiplied the rating by a saturating function of the review count:

$$\text{Score} = R \cdot (1 - e^{(-n / 25)})$$

This correctly kept one-review businesses out of the top spots, but production experience surfaced three structural failures:

- **It shrinks toward zero, not toward "average".** A business with few reviews is treated the same as a bad business — both sink to the bottom. Statistically, the honest estimate for a business we know little about is "probably near the local average", not "terrible".
- **Volume can beat quality.** The table below shows the inversion: a mediocre 3.2-star business with 500 reviews outscores an excellent 4.9-star business with 20 reviews. No visitor believes the first business is better; it is merely older.

- **One constant cannot fit every market.** With the constant fixed at 25 reviews, the formula saturates for nearly every restaurant (rankings collapse to raw, rounded stars) while review count dominates for low-volume verticals such as IT services. The same formula behaves as two different algorithms depending on the page. It also contains no notion of time: a business dormant for three years keeps its rank indefinitely.

Business	Rating	Reviews	Old formula	v2 core (v1)*
Excellent, newer	4.9	20	2.70	4.57
Mediocre, high volume	3.2	500	3.20	3.25

**Bayesian average of Section 4.3 with illustrative parameters C = 4.3, m = 25. The old formula ranks the mediocre business first; the v2 core restores the order a reasonable visitor expects.*

4. The Localist v2 score

4.1 Notation

Symbol	Meaning	Source
R	Star rating of the business (1.0–5.0)	Public GBP listing
n	Total review count of the business	Public GBP listing
a	Age, in months, of the min(n, 5)-th most recent review — i.e. the oldest of the last five (0 if n = 0)	Last-5 review timestamps
C	Prior rating: mean of R across the (vertical, city) pool	Computed per page
m	Prior weight: clamp(median of n across the pool, 10, 50)	Computed per page
g	Grace period: clamp(median of a across pool businesses with n ≥ 5, 3, 12), in months	Computed per page
v1	Evidence-weighted score (Section 4.3)	Computed
F	Freshness factor in (0, 1] (Section 4.4)	Computed
v2	Final Localist score (Section 4.5)	Computed & published

4.2 Local calibration: C, m and g

All three calibration parameters are derived from the pool of businesses on the same directory page, and republished on that page:

- **C (prior rating)** is the arithmetic mean rating of the pool. It answers: “what does an average business in this market look like?” A business about which we know nothing is scored at C — neither promoted nor punished.
- **m (prior weight)** is the pool’s median review count, clamped to the range [10, 50]. It sets how many reviews a business needs before its own rating outweighs the prior. Deriving m from the pool implements principle P3: in review-rich markets (restaurants) the bar is higher; in review-sparse markets (B2B services) it is lower. The clamp keeps extreme pools from making the score either gullible or unreachable.
- **g (grace period)** is the pool’s median value of a, computed only over businesses with at least five reviews, clamped to [3, 12] months. A business whose last five reviews arrived within the typical local window is “fresh” and pays no penalty. Restricting the median to established businesses prevents a crowd of new one-review listings from dragging the grace period down.

Thin pools: if a page has fewer than 10 businesses, C and m fall back to the vertical-wide values (all cities pooled); if no business in the pool has 5 or more reviews, g defaults to 6 months.

4.3 Step 1 — evidence-weighted score (Bayesian shrinkage)

$$v1 = (n \cdot R + m \cdot C) / (n + m)$$

This is the standard Bayesian (credibility-weighted) average used by large rating platforms. The interpretation is direct: every business starts with *m* phantom reviews at the market-average rating *C*, and each real review pulls the score toward the business’s own rating *R*. With few reviews, v1 stays close to C; as n grows past m, the business’s own record dominates. v1 always lies between C and R.

4.4 Step 2 — freshness factor

$$F = 0.5 \text{ ^ } (\max(0, a - g) / 24)$$

F equals 1 while the business’s recent-review age a is within the local grace period g, then decays with a 24-month half-life: evidence that is two years staler than the local norm counts half; four years staler counts a quarter.

Why decay the deviation rather than the review count? Discounting n directly looks natural but fails exactly where it matters: for a business with 400 reviews, halving n to 200 leaves the score essentially unchanged, because 200 still dwarfs the prior weight m. High-volume businesses would be immune to staleness. Applying F to the deviation (v1 − C), as Step 3 does, makes

dormancy bite proportionally at every volume: a fully dormant business returns to the market average no matter how many reviews it accumulated in a previous era.

4.5 Step 3 — final score

$$v2 = C + F \cdot (v1 - C)$$

Read it as: “start from the local average; move toward the business’s own evidence-weighted rating; move less if the evidence is stale.” The complete procedure is given as Algorithm 1.

```
Algorithm 1 — Localist v2 scoring for one directory page (vertical V, city L)

Input:  set B of businesses in (V, L); for each b in B:
        R_b  (Google star rating, 1.0–5.0)
        n_b  (Google review count)
        t_b  (timestamps of the min(n_b, 5) most recent reviews)
Output: score v2_b and rank for each eligible b

 1:  C  <- mean of R_b over B                          // local prior rating
 2:  m  <- clamp( median of n_b over B, 10, 50 )       // prior weight
 3:  E  <- { b in B : n_b >= 5 }                       // established subset
 4:  for each b in B:
 5:      a_b <- months elapsed since the oldest timestamp in t_b
         (a_b <- 0 if n_b = 0)
 6:  g  <- clamp( median of a_b over E, 3, 12 )       // grace period, months
         (g <- 6 if E is empty)
 7:  for each b in B:
 8:      v1_b <- ( n_b * R_b  +  m * C ) / ( n_b + m )
 9:      F_b  <- 0.5 ^ ( max(0, a_b - g) / 24 )
10:      v2_b <- C  +  F_b * ( v1_b - C )
11:  Eligible <- E   // businesses with n_b >= 5
12:  sort Eligible by ( v2_b desc, n_b desc, a_b asc )
13:  Merit list   <- { b in Eligible : v2_b >= C }
14:  Also notable <- remaining businesses, same sort order
15:  publish v2_b (2 d.p.), C, m, g, and the crawl date on the page
```

4.6 Properties

- **Bounded and well-behaved.** v1 always lies between C and R; v2 always lies between C and v1. No business can score above 5.0 or ride the prior above its own rating.

- **Fresh businesses are untouched.** If a ≤ g then F = 1 and v2 = v1 exactly. The freshness machinery is invisible to any business receiving reviews at the normal local pace.
- **Dormancy is forgetting, not punishment.** As a grows, v2 converges to C — the market average — not to the bottom of the list. We stop claiming to know the business is excellent; we do not start claiming it is bad. This is symmetric: a dormant low-rated business also drifts toward C, because old bad evidence expires too.
- **More evidence, more credit.** For businesses rated above the local average, v2 increases strictly with review count at any fixed rating and freshness. Two businesses with identical ratings are separated by volume and recency of proof.
- **No cross-market leakage.** Scores are functions of the pool. The same business may hold different scores on different pages — correctly, because it holds a different competitive position in each market (see Section 9).

5. Edge cases and safeguards

Case	Handling	Rationale
n = 0 (no reviews)	v2 = C by construction; F set to 1. Not eligible for ranking (Section 6). Sorted last by tie-break.	No evidence, no claim.
1 ≤ n ≤ 4	a is measured from the oldest available review. Scored normally but not eligible for the ranked list.	v1 already sits close to C; five reviews is the minimum for a public quality claim.
Pool smaller than 10 businesses	C and m fall back to vertical-wide values.	A tiny pool makes the local average too noisy to serve as a prior.
No business with n ≥ 5 in pool	g defaults to 6 months.	The local review pace cannot be estimated.
Tied scores	Ties broken by review count (descending), then by recency of the last review.	Google’s 0.1-star rounding makes ties common in review-rich markets.

A display rule accompanies the last two rows of the scoring pipeline: for businesses with fewer than five reviews, directory pages show the raw stars and review count with an “insufficient data” label instead of a Localist score. The computed score would be nearly identical to C for all of them — correct for ordering, but misleading if printed next to a one-star badge.

6. From scores to a published ranking

Scores become a directory page through four published rules:

- **Eligibility gate.** Only businesses with at least 5 reviews are ranked. Others appear unranked with an “insufficient data” label.
- **Sort order.** v2 descending; ties broken by review count, then by most recent review.
- **Merit list.** Businesses with v2 ≥ C — those demonstrably at or above their local average — form the headline “Merit Top 10”. Remaining eligible businesses fill an “Also notable” block. Because the cutoff is the pool’s own average, it self-adjusts to every market and reads naturally: “ranked above the local average.”
- **Publication cycle.** GBP data is re-crawled and rankings are recomputed monthly. Every page carries the stamp “Rankings computed: <month year>” and publishes C, m and g. Between cycles the ranking does not move — visitors see a stable, dated list, not one that reshuffles on every visit.

Featured or paid placements, where offered, appear in a visually separate block and never alter organic positions. This is a hard guarantee of the platform, unchanged from v1.

7. Worked example: one full market

Consider a fictional pool of eight web-development businesses in one city. Column a is the age in months of each business’s fifth-most-recent review.

Business	R	n	a (mo)	v1	F	v2
Apex Web Studio	4.9	87	1	4.75	1.00	4.75
BrightPixel	4.7	210	2	4.66	1.00	4.66
Citywide Digital	4.5	150	4	4.50	0.97	4.50
DevCraft	4.9	20	2	4.61	1.00	4.61
Eastside Code Co.	4.8	400	36	4.77	0.39	4.60
FastSite Labs	3.2	500	3	3.32	1.00	3.32
Greenline Tech	5.0	3	5	4.53	0.94	4.53
Harbor Apps	4.0	12	8	4.40	0.87	4.42

7.1 Pool calibration

- C = mean rating = (4.9 + 4.7 + 4.5 + 4.9 + 4.8 + 3.2 + 5.0 + 4.0) / 8 = 4.50
- m = median review count = 118.5 → clamped to the [10, 50] range → m = 50
- g = median a over the seven businesses with n ≥ 5 = median(1, 2, 4, 2, 36, 3, 8) = 3 months (within the [3, 12] clamp; Greenline Tech, with n = 3, is excluded from this median)

7.2 Two businesses in full

DevCraft (4.9 stars, 20 reviews, fifth-latest review 2 months old):

$$\mathrm{v1} = (20 \cdot 4.9 + 50 \cdot 4.50) \,/\, (20 + 50) = (98 + 225) \,/\, 70 = 4.614$$

$$\mathrm{a} = 2 \le \mathrm{g} = 3 \;\Rightarrow\; \mathrm{F} = 1 \;\Rightarrow\; \mathrm{v2} = 4.61$$

Eastside Code Co. (4.8 stars, 400 reviews, but no review in three years — fifth-latest is 36 months old):

$$\mathrm{v1} = (400 \cdot 4.8 + 50 \cdot 4.50) \,/\, (400 + 50) = 2145 \,/\, 450 = 4.767$$

$$\mathrm{F} = 0.5\text{^}((36 - 3) \,/\, 24) = 0.5\text{^}1.375 = 0.386$$

$$\mathrm{v2} = 4.50 + 0.386 \cdot (4.767 - 4.50) = 4.60$$

7.3 The published page

Rank	Business	Score	Stars	Reviews	Placement
1	Apex Web Studio	4.75	4.9	87	Merit list
2	BrightPixel	4.66	4.7	210	Merit list
3	DevCraft	4.61	4.9	20	Merit list
4	Eastside Code Co.	4.60	4.8	400	Merit list · dormancy note
5	Citywide Digital	4.50	4.5	150	Merit list (v2 = C)
6	Harbor Apps	4.42	4.0	12	Also notable
7	FastSite Labs	3.32	3.2	500	Also notable
—	Greenline Tech	—	5.0	3	Insufficient data (n < 5)

Three outcomes illustrate the method's character. First, DevCraft — twenty reviews, all recent, at 4.9 stars — overtakes Eastside's four hundred stale ones; under v1 arithmetic alone Eastside would have ranked first. Second, FastSite Labs' 500 reviews cannot lift a 3.2-star record above

businesses the market actually rates higher: volume is credibility, not quality. Third, Greenline Tech's perfect 5.0 is neither rewarded nor punished — it is simply not yet a claim the data can support, and the page says so.

8. Integrity flags: signals that inform but never adjust

The last five reviews carry more than timestamps: per-review star ratings and full text enable integrity checks. These are deliberately kept out of the score. A hidden fraud weight inside the formula would break the recomputability promise (P5); a published list of flags does not. Each flag either annotates a listing visibly or, in severe cases, gates it out of the ranking pending manual review.

Flag	Trigger	Effect
Trending lower	Mean of the last 5 review stars is at least 1.5 below the lifetime rating, and $n \geq 50$	Visible badge on the listing
Dormancy note	$F < 0.5$ (recent-review age more than one half-life beyond the local grace period)	Visible badge; score already reflects the decay
Review burst	All 5 recent reviews within a short window (e.g. 14 days) immediately following 12+ months of silence	Listing held for manual review; freshness reset from burst reviews is ignored
Duplicate text	Near-duplicate wording across the recent reviews	Listing held for manual review

Example of the first flag: a business showing 4.8 stars across 320 lifetime reviews whose last five ratings are 2, 3, 1, 2 and 3 (mean 2.2) is displaying a lifetime average that no longer describes it. Its score — anchored to the slow-moving lifetime figures Google publishes — will react only gradually; the badge tells visitors what the number cannot yet.

The trend threshold is deliberately conservative. Five reviews is a small sample: a mean of five can swing a full star on two unhappy customers. Requiring both a 1.5-star gap and at least 50 lifetime reviews keeps the false-positive rate low enough for a public accusation of decline.

9. How to read a Localist page

- **The score is local.** "4.61 in New York web development" is a statement about a business's standing among New York web-development companies. The same business

may score differently on another city's or vertical's page because it faces different competition there. Scores are never presented as a global badge.

- **Percentile beats decimals.** Alongside the score, pages display standing relative to the pool — "above the local average of 4.50" or "top 15% in this market" — because a raw 4.61 carries little meaning on its own.
- **The stamp matters.** Every page states when its data was crawled and when rankings were computed. A comparison table exported or sent from a page carries the same stamp and a link back to the live page, because a ranking is a dated measurement, not a permanent fact.
- **The parameters are printed.** C, m and g appear on every page. With those three numbers, the formulas in Section 4 and a business's public GBP listing, any reader can reproduce any score to two decimal places.

10. Limitations and gaming resistance

No method built on public review data is beyond manipulation or error; honesty about the boundaries is part of the methodology.

- **Monthly latency.** A sudden reputation collapse appears in rankings only at the next monthly recompute. The page stamp makes this latency explicit.
- **The five-review window.** Freshness is estimated from the five most recent reviews, so a dormant business could reset its freshness with five quick purchased reviews. The review-burst flag (Section 8) targets exactly this pattern, and month-over-month review-count deltas are logged to strengthen velocity estimation over time.
- **Google's rounding.** GBP publishes ratings rounded to 0.1 stars, which caps the precision of any downstream score and makes tie-breaks unavoidable in dense markets.
- **Review quality is not judged.** The score counts verified reviews; it does not assess whether a review is thoughtful or thin. Text-based signals are used only for the integrity flags, never as score inputs, because subjective quality weights cannot be recomputed by readers.
- **What we will not do.** Self-reported testimonials, paid reviews, off-platform ratings and "featured" payments never enter the score. Businesses with strong off-platform evidence (industry directories, case studies) may attach it to a listing request for manual staff review — it can fast-track inclusion, never a score.

11. Frequently asked questions from business owners

Why is my Localist score lower than my Google star rating?

The score blends your rating with the local average in proportion to your review count, and discounts stale evidence. A 5.0 rating with eight reviews genuinely does not yet prove more than a 4.6 with two hundred — the score says so. As verified recent reviews accumulate, your score approaches your rating.

What is the fastest legitimate way to improve my rank?

Ask genuinely satisfied customers to leave a Google review, steadily rather than in bursts. Recent reviews raise both your evidence weight and your freshness. Ten thoughtful new reviews can move a young business from “insufficient data” into the ranked list; sustained reviewing keeps an established business from decaying toward the market average.

Why does my score differ between two cities I operate in?

Each page calibrates to its own market: different competitor averages (C), different review-volume norms (m), different review pace (g). Your standing is measured against the businesses a visitor would actually compare you with on that page.

Can I pay to improve my position?

No. Featured placements exist on some pages but are visually separate and never change organic positions. There is no mechanism — internal or paid — that adjusts a computed score.

My last five reviews were all 5 stars. Why do I see a “trending lower” or dormancy note?

Dormancy notes come from timing, not sentiment: if the fifth-most-recent review is old relative to the local pace, the note appears even if every review is glowing. It disappears automatically once review activity resumes.

Appendix A — Parameter summary

Parameter	Definition	Range / default	Published?
C	Mean rating of the (vertical, city) pool	1.0–5.0	Yes, per page
m	clamp(median pool review count, 10, 50)	[10, 50]	Yes, per page
g	clamp(median recent-review age over pool businesses with n ≥ 5, 3, 12)	[3, 12] mo; 6 if undefined	Yes, per page
Half-life	Decay rate of stale evidence	24 months (fixed)	Yes, global
Eligibility	Minimum reviews to be ranked	n ≥ 5 (fixed)	Yes, global
Merit cutoff	Threshold for the headline list	v2 ≥ C	Yes, global
Recompute cycle	Crawl and ranking refresh	Monthly	Yes, per page stamp

Fixed constants (the 24-month half-life, the [10, 50] and [3, 12] clamps, the 5-review gate) were chosen against live listing data to balance responsiveness against manipulation resistance, and are held constant across all markets so that only the three published pool parameters vary.

Appendix B — Reference implementation

The complete scoring pipeline, as deployed, in standard Python (pandas). Runtime is a few seconds for tens of thousands of businesses; the method adds no meaningful computational cost at any realistic catalog size.

```
import pandas as pd

# df columns: business_id, vertical, city, rating, review_count,
#             months_since_kth_review   (k = min(review_count, 5))

def localist_v2(df: pd.DataFrame) -> pd.DataFrame:
    pool = df.groupby(['vertical', 'city'])
    df['C'] = pool['rating'].transform('mean')
    df['m'] = pool['review_count'].transform('median').clip(10, 50)

    est = df[df.review_count >= 5]
    g = (est.groupby(['vertical', 'city'])['months_since_kth_review']
            .median().clip(3, 12).rename('g'))
    df = df.join(g, on=['vertical', 'city'])
    df['g'] = df['g'].fillna(6)
```

```
    df['v1'] = ((df.review_count * df.rating + df.m * df.C)
                / (df.review_count + df.m))
    excess = (df.months_since_kth_review.fillna(0) - df.g).clip(lower=0)
    df['F'] = 0.5 ** (excess / 24)
    df['v2'] = df.C + df.F * (df.v1 - df.C)

    df['eligible'] = df.review_count >= 5
    df = df.sort_values(
        ['vertical', 'city', 'eligible', 'v2', 'review_count'],
        ascending=[True, True, False, False, False])
    df['merit'] = df.eligible & (df.v2 >= df.C)
    return df
```

Questions, corrections or disputes: hello@konigle.com. We would rather a reader recompute a score and challenge it than trust it blindly — that is what this document is for.