

Two Quantile-Based Quality Metrics

Francesco Balocco

February 17, 2026

1 Why quantile-based metrics?

When star ratings are *selectively observed* (some restaurants get visited and reviewed more often than others), raw stars mix two things:

1. **Observable-driven valuation:** what we can predict from observable characteristics (price level, cuisine tags, location proxies, etc.).
2. **Residual quality:** what remains after controlling for observables (idiosyncratic quality, luck, reviewer tastes, measurement noise).

After constructing a latent quality score \hat{z}_i (read: “our estimated score for restaurant i ”), we want a *rank-based* metric: “How good is restaurant i relative to others?” Quantiles do exactly this.

2 Key objects: CDF and quantile function

A quick “symbol cheat sheet” (plain language)

This guide uses a few symbols repeatedly. Here is what they mean, without requiring a math background:

- **Restaurant index:** i (restaurant i), j (restaurant j), etc.
- **Latent score variable:** Z is the score in general (think: “quality score as a random variable”).
- **A particular score level:** z is one possible value on the score axis.
- **Estimated score:** \hat{z}_i is the score we computed from data for restaurant i . The “hat” $\hat{}$ just means “estimated from data”.
- **Percentile:** p is a percentile written between 0 and 1. For example, $p = 0.80$ means the **80th percentile**.
- **Stars:** Y is the observed star rating (e.g., 3.5, 4.0, 4.5).

Intuition first: two functions that convert between “value” and “rank”

Two functions do all the work in both metrics:

1. The **CDF** turns a score value into a percentile (a rank):

$$F_Z(z) \approx \text{“fraction of restaurants with score } \leq z\text{”}.$$

2. The **quantile function** turns a percentile back into a value on some scale:

$$Q_Z(p) \approx \text{“the score value at percentile } p\text{”}.$$

If you remember only one phrase, use:

$$F(\text{value}) = \text{fraction at or below}, \quad Q(\text{fraction}) = \text{value at that fraction}.$$

Let Z be a (latent) score with cumulative distribution function (CDF)

$$F_Z(z) = \mathbb{P}(Z \leq z).$$

The (generalized) quantile function is the inverse CDF:

$$Q_Z(p) = \inf\{z : F_Z(z) \geq p\}, \quad p \in (0, 1).$$

These are linked by the identity (for continuous distributions)

$$Q_Z(F_Z(z)) = z$$

and by the **probability integral transform**: if $Z \sim F_Z$, then $U = F_Z(Z)$ is (approximately) uniform on $(0, 1)$.

Empirical versions (what we compute). With scores $\{\hat{z}_j\}_{j=1}^N$ in a reference set (typically *training data*), the empirical CDF is

$$\hat{F}_Z(z) = \frac{1}{N} \sum_{j=1}^N \mathbf{1}\{\hat{z}_j \leq z\}.$$

In words: count how many restaurants have score $\leq z$, then divide by N to get a fraction. Given $p \in (0, 1)$, an empirical quantile can be obtained from order statistics (often with interpolation):

$$\hat{Q}_Z(p) \approx \text{Quantile}(\{\hat{z}_j\}_{j=1}^N, p).$$

In words: sort the scores; the p -quantile is the value around “ p of the way through” the sorted list.

3 Construction A (basic): “unbiased” stars via a uniform target

3.1 Definition

Compute the percentile of restaurant i in the latent-score distribution:

$$p_i = \hat{F}_Z(\hat{z}_i).$$

In words: p_i is the share of restaurants that score below restaurant i . So $p_i = 0.90$ means restaurant i is better than about 90% of restaurants by the latent score. Then map percentiles linearly onto a 1–5 star scale:

$$s_i^{(\text{unif})} = 1 + 4p_i.$$

Interpretation: by construction, $s^{(\text{unif})}$ spreads restaurants evenly across the 1–5 scale (a “fair” ranking where every percentile is equally represented). **Why “unbiased” in this project:** it does not copy the (possibly biased) shape of the observed star distribution; it uses a neutral target where each percentile gets equal space on the 1–5 scale.

3.2 CDF \rightarrow quantile view

This is a CDF-to-quantile mapping:

$$\hat{z} \xrightarrow{\hat{F}_Z} p \xrightarrow{Q_{\text{Uniform}(1,5)}} s^{(\text{unif})}, \quad \text{where} \quad Q_{\text{Uniform}(1,5)}(p) = 1 + 4p.$$

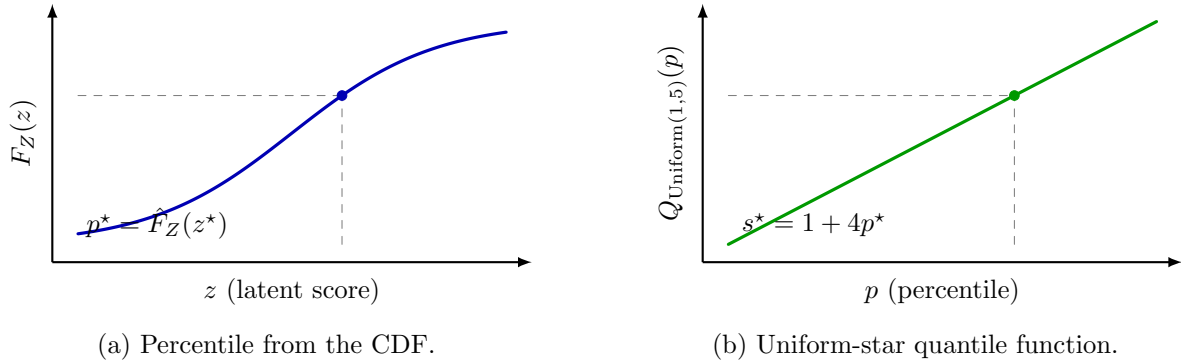


Figure 1: Basic construction: use the latent-score CDF to get a percentile, then apply the quantile function of a *uniform* 1–5 star distribution.

4 A quick picture of “percentile” (area under the curve)

Students often find it helpful to remember that the CDF is an *accumulated area* under a density curve. In a schematic latent-score distribution, the shaded area up to z^* equals the percentile $p^* = F_Z(z^*)$.

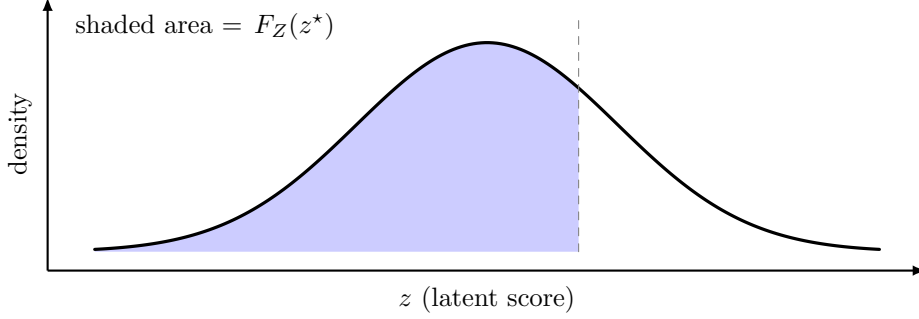


Figure 2: Percentile intuition: $F_Z(z^*)$ is the area under the density to the left of z^* .

5 Construction B (quantile-match): match the *observed* star distribution

5.1 Definition

Again compute percentiles in latent scores:

$$p_i = \hat{F}_Z(\hat{z}_i).$$

But instead of forcing a uniform 1–5 distribution, map p_i through the *empirical quantile function of observed stars*:

$$s_i^{(\text{match})} = \hat{Q}_Y(p_i),$$

where Y is the observed star rating (in a reference set, typically rated restaurants in the training sample).

Interpretation: this produces a star-like number that preserves *the same marginal star distribution* as the original stars (e.g., if stars are mostly around 4.0–4.3, the mapped score will also be mostly around 4.0–4.3). **Plain-language interpretation:** “if my restaurant is at the 80th percentile by latent score, I give it the star value that the 80th percentile restaurant had in the original ratings.”

5.2 CDF \rightarrow quantile view

This is the same CDF-to-quantile pipeline, but with a different target quantile function:

$$\hat{z} \xrightarrow{\hat{F}_Z} p \xrightarrow{\hat{Q}_Y} s^{(\text{match})}.$$

In practice, \hat{Q}_Y is often a step function because stars are discrete/rounded.

6 Quantile matching as a “percentile-preserving” map (QQ intuition)

Quantile matching is easiest to visualize as “keep the percentile the same, but change the scale”. That is exactly what

$$s^{(\text{match})} = \hat{Q}_Y(\hat{F}_Z(\hat{z}))$$

does.

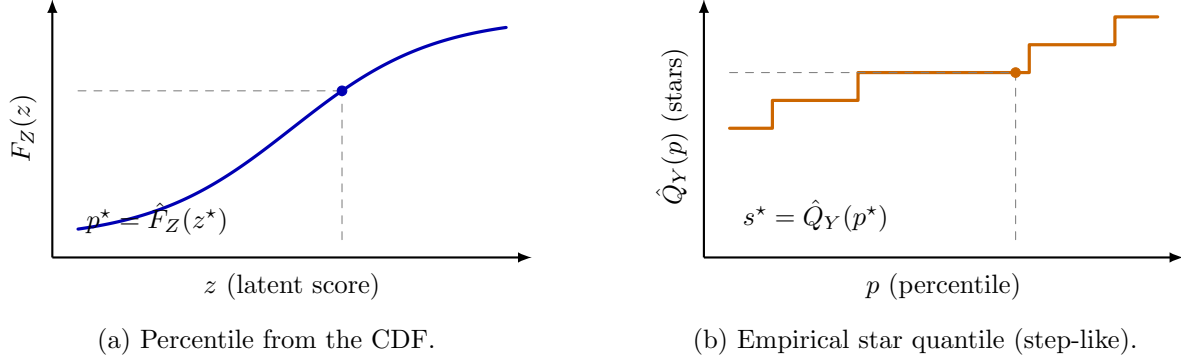


Figure 3: Quantile-match construction: use the latent-score CDF to get a percentile, then apply the *empirical* quantile function of observed stars. This matches the marginal star distribution by construction.

7 When to use which? (quick guidance)

- Use **basic uniform mapping** ($s^{(\text{unif})}$) when you want a *pure ranking* with a maximally spread 1–5 scale.
- Use **quantile-match** ($s^{(\text{match})}$) when you want numbers that “look like stars” and have the *same marginal distribution* as observed stars.
- Quantile-match is *not* a magic accuracy improvement: it enforces a histogram shape; it does not make individual predictions closer to ground truth.

8 Implementation checklist (train/validation/holdout)

Key practical rule (avoid “peeking” at the test set)

Build the mapping using **training data only** (the training CDF, and the training star quantiles). Then apply it to validation/test restaurants. This keeps evaluation honest: you are not using the test set to define what “top 10%” means.

1. Fit your latent score model using training data and compute training scores \hat{z}^{train} .
2. Define *training* empirical CDF \hat{F}_Z from \hat{z}^{train} .
3. For any restaurant (train/validation/test), compute $p_i = \hat{F}_Z(\hat{z}_i)$ using the *training* CDF.
4. Basic mapping: $s_i^{(\text{unif})} = 1 + 4p_i$.
5. Quantile-match: define \hat{Q}_Y from *training observed stars* Y^{train} , then set $s_i^{(\text{match})} = \hat{Q}_Y(p_i)$.

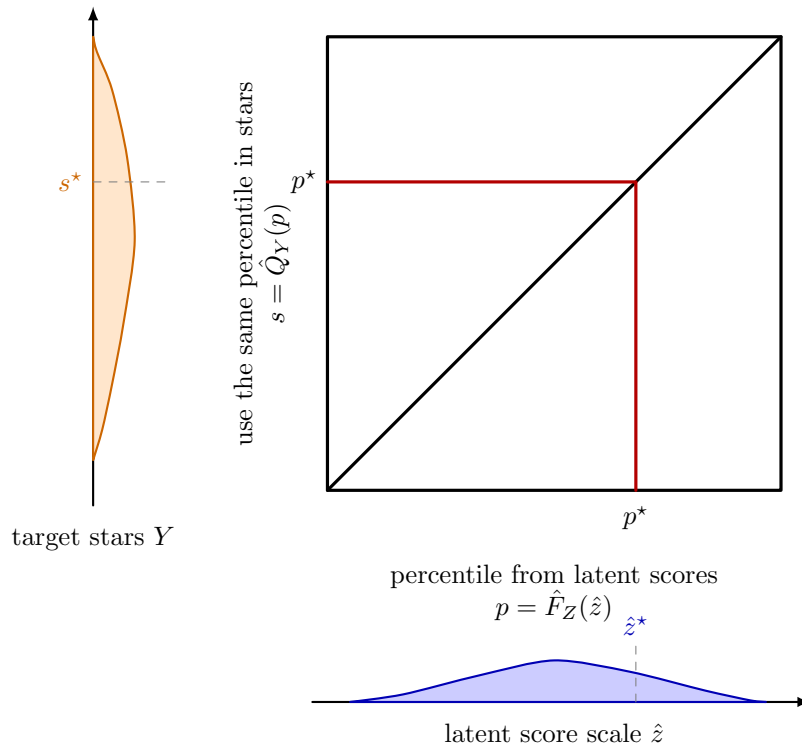


Figure 4: Quantile matching keeps the percentile p^* fixed (red path): start from a latent score \hat{z}^* , compute $p^* = \hat{F}_Z(\hat{z}^*)$, then report $s^* = \hat{Q}_Y(p^*)$.