Chapter 3

Hierarchical Bayesian Anchors and Tail Gating for Long-Tail Estimation in Sparse Marketplace Data

Abstract. This chapter develops the methodological and algorithmic foundations behind a dedicated tail-classifier that isolates long-tail (“zombie”) listings from fast-selling (“sniper”) listings, and explains how hierarchical Bayesian anchor priors make the tail learnable in sparse, high-variance segments. The core contribution is a time-zero (t_0)-faithful pipeline that (i) constructs robust price and speed anchors through a backoff hierarchy with shrinkage, (ii) trains an Accelerated Failure Time (AFT) model under censoring to predict time-to-sale, and (iii) evaluates and tunes the model explicitly on classification performance at operational decision boundaries (72 hours and 21 days).

Chapter outline.

3.1 Motivation: tails, liquidity, and decision boundaries

Consumer-to-consumer marketplaces exhibit a characteristic high-frequency front—a large mass of listings that sell quickly—followed by a pronounced long tail of listings that remain live for weeks. Operationally, these regimes behave like two different markets: the front is governed by acute demand–supply matching and pricing proximity to the local clearing price, while the tail is dominated by mispricing, low seller responsiveness, listing quality issues, and segment-specific scarcity.

For a pricing or deal-selection system, the tail is not merely a statistical curiosity. It is where opportunity cost accumulates: capital and attention become trapped in illiquid inventory; monitoring costs rise; and the reliability of aggregate performance metrics degrades because a large fraction of outcomes are censored (unsold at the time of evaluation). In practice, decision-makers do not require a perfectly calibrated survival curve for every listing. They require robust answers at decision boundaries that map directly onto actions such as:

• “Will this sell in the next 72 hours?” (fast liquidity; opportunistic execution)
• “Is this likely to still be unsold after 21 days?” (zombie risk; avoid or demand discount)

This chapter therefore treats tail capture as a first-class modeling objective. The methodological move is to explicitly represent two boundary classifiers—Fast-72 and Slow-21—alongside a full survival model. The classifiers are not a concession to simplicity; they are a principled response to the fact that tails are disproportionately influenced by sparse segments and censoring, and therefore require specialized inductive bias and weighting.

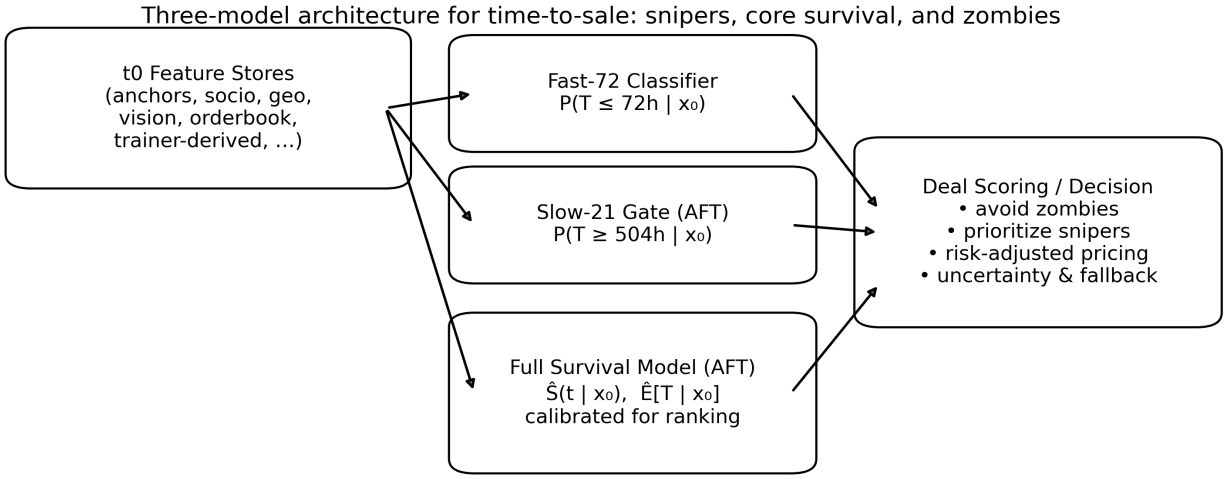

Figure 3.1: Three-model architecture for time-to-sale: Fast-72, core survival, and Slow-21.

3.2 Formal problem statement and notation

Let each marketplace listing i be represented at decision time t_0 by a feature vector $x_i(t_0)$ constructed strictly from information observable at t_0. Let the (random) time-to-sale be T_i measured in hours since t_0. Listings that do not sell within the observation window are right-censored by a censoring time C_i (e.g., last seen or evaluation cutoff). The observed outcome is:

$$y_i = \min(T_i, C_i), \quad \delta_i = \mathbf{1}[T_i \le C_i]$$

The modeling objective is to estimate either the survival function $S(t|x_i(t_0)) = P(T>t \mid x_i(t_0))$ or its induced boundary events. Two boundary labels are used operationally:

Fast-72 (sniper). $y^{fast72}_i = 1[T_i \le 72h]$ for sold listings. This supports short-horizon execution decisions.
Slow-21 (zombie). $y^{slow21}_i = 1[T_i \ge 504h]$ for sold listings. This supports avoidance or discounting of illiquid deals.

Because the same features drive both survival time and boundary events, it is advantageous to use a survival model whose outputs can be queried at any horizon. The AFT family provides a natural parameterization for marketplace durations and supports censoring via interval-likelihood training.

$$\log T_i = f_\theta(x_i) + \sigma \varepsilon_i, \quad \varepsilon_i \sim \mathcal{D}$$

Here f(·) is a gradient-boosted tree ensemble, σ is a scale parameter, and is a choice of noise distribution (Normal, Logistic, or Extreme Value). In practice, the system trains with XGBoost's survival:aft objective and exposes the model both as a full duration predictor and as a boundary classifier by thresholding calibrated time predictions.

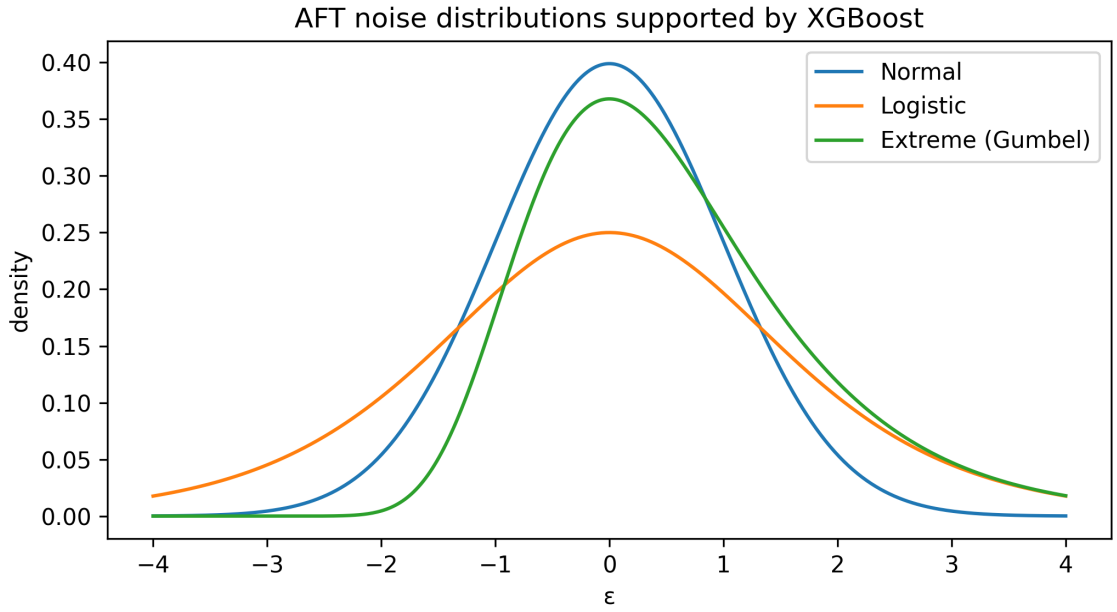

Figure 3.2: Noise distributions supported by the AFT objective (standardized forms).

3.3 Hierarchical Bayesian anchors for sparse segments

A central difficulty in tail modeling is that the most informative segments are often the sparsest. For example, "iPhone 14 Pro, 512GB, excellent condition, Oslo" may have only a handful of comparable sales in the last 30 days, yet its pricing and liquidity dynamics differ materially from the global population. A purely nonparametric model trained directly on raw features is forced to choose between two failure modes: (i) overfit to noise in small segments, or (ii) ignore segment structure and regress to global averages—both of which degrade tail discrimination.

The remedy is to inject anchor priors: statistically stable, leak-proof, time-zero estimates of the typical price and speed for a listing's segment. Anchors are not arbitrary baselines; they are engineered estimators with an explicit bias–variance trade-off and an explicit backoff hierarchy. In the system under study, anchors are produced by a dedicated certified store and exposed as features, not as ad-hoc joins.

Anchor types. The anchor-priors store emits a family of features that summarize both level (typical price) and tempo (typical time-to-sale) at t_0: • ptv_anchor_strict_t0: a strict segment median, computed over a hierarchy that preserves generation and avoids storage mixing. • ptv_anchor_smart: a stabilized anchor that blends strict estimates with higher-support backoffs and stability rules. • ptv_sold_30d: robust recent sold median for the relevant segment/window. • Support diagnostics such as anchor_n30_t0, anchor_n60_t0, and anchor_level_k indicating which hierarchy level fired.

The model subsequently consumes anchors primarily through relative features (e.g., price-to-anchor ratios, deltas vs sold medians, or speed residuals). Relative parameterizations are crucial: they allow the learner to exploit cross-segment invariants such as "overpricing by 15% relative to the local anchor increases zombie risk," even when absolute price levels differ across generations.

$$\hat{m}_s^{\text{post}} = \frac{n_s \hat{m}_s + \kappa \hat{m}_{\text{parent}(s)}}{n_s + \kappa}$$

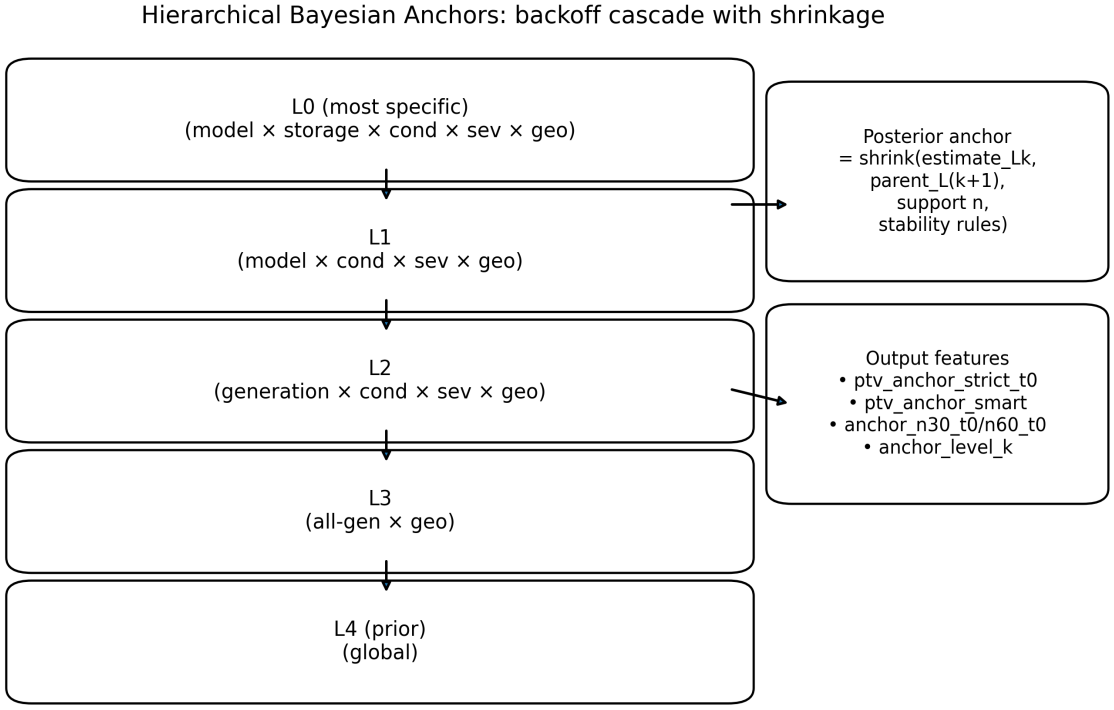

Figure 3.3: Backoff hierarchy for anchor estimation. Each level shrinks toward its parent using support-aware rules.

Statistical interpretation. Although anchors are implemented via robust SQL aggregations and deterministic fallback logic, they correspond closely to an empirical-Bayes estimator. Let s denote a segment node in the hierarchy, with an observed robust estimate $\hat{m}_s$ (e.g., a median sold price or median hours) computed

from n_s comparable outcomes inside a trailing window. When n_s is small, the estimator's variance is high; when n_s is large, the estimator is stable. A hierarchical construction yields a posterior mean of the shrinkage form shown above, where κ is an effective prior sample size. This has three desirable properties: (i) it is unbiased in the large-n limit, (ii) it shrinks aggressively in sparse segments, and (iii) it preserves locality whenever support exists.

Strict vs. smart anchors. The system distinguishes between a strict anchor (ptv_anchor_strict_t0) and a smart anchor (ptv_anchor_smart) to separate two objectives: interpretability and stability. The strict anchor answers a narrow question—"What is the robust median in the most specific valid cohort?"—and therefore retains clear semantic meaning. The smart anchor answers a broader operational question—"What anchor should a model trust today?"—and therefore includes additional stability checks, support thresholds, and (where necessary) backoff to reduce volatility.

Advanced anchor cascade. The smart anchor implements a fallback cascade over multiple cohorts and windows, combined with explicit clipping and stability rules. In pseudocode form:

```
Given listing features (model, generation, storage, geo, condition_score, severity) at t0:

1) Compute strict cohort medians over multiple windows (e.g., 30d and 60d) with support counts n30, n60.2)
Select the deepest hierarchy level Lk with n_eff ≥ n_min and stability_ok = True.
3) Shrink toward parent levels if n_eff is marginal:
      anchor = (n_eff * median_Lk + kappa * median_parent) / (n_eff + kappa)
4) Apply safety rails:
      - no storage mixing when unstable
      - preserve generation to avoid cross-gen drift
      - clip anchor within percentile band [p10, p90] from higher-support cohorts
5) Emit: ptv_anchor_smart, ptv_anchor_strict_t0, n_support, and anchor_level_k.
```

In effect, the anchor store externalizes the part of the learning problem that must be stable and leak-proof. This reduces the burden on the downstream tail classifier, which can now learn residual relationships with substantially improved sample efficiency.

3.4 Slow-21 gate: an AFT model tuned for tail discrimination

The Slow-21 gate is designed to answer a specific operational question: Is a listing likely to remain unsold for at least 21 days? This is not equivalent to predicting the exact duration for all listings. Instead, it is a boundary-focused model whose loss, weights, calibration, and evaluation are aligned with the 21-day decision.

Training data and censoring. The gate is trained on a combination of sold listings (events) and long-active listings (censored). Sold listings provide exact durations; censored listings provide lower bounds (the listing has survived at least that long). The AFT objective supports interval targets by specifying a lower bound l_i and upper bound u_i for each sample. For sold events, $l_i=u_i=T_i$. For right-censored samples, $u_i=+\infty$ and $l_i=C_i$.

$$\ell_i(\theta) = \log\left(F\left(\frac{\log u_i - \mu_i}{\sigma}\right) - F\left(\frac{\log l_i - \mu_i}{\sigma}\right)\right)$$

Here $\mu_i=f(x_i)$ is the model output in log-time space and F is the CDF of the chosen noise distribution. The interval likelihood reduces to a point likelihood for events and to a one-sided likelihood for right-censored observations.

From survival prediction to a gate. After training, the model produces a predicted time-to-sale $\hat{T}$ (optionally calibrated). The Slow-21 gate is then the induced classifier:

```
ŷ_slow21 = 1[  T^_calibrated  ≥  504h ].
```

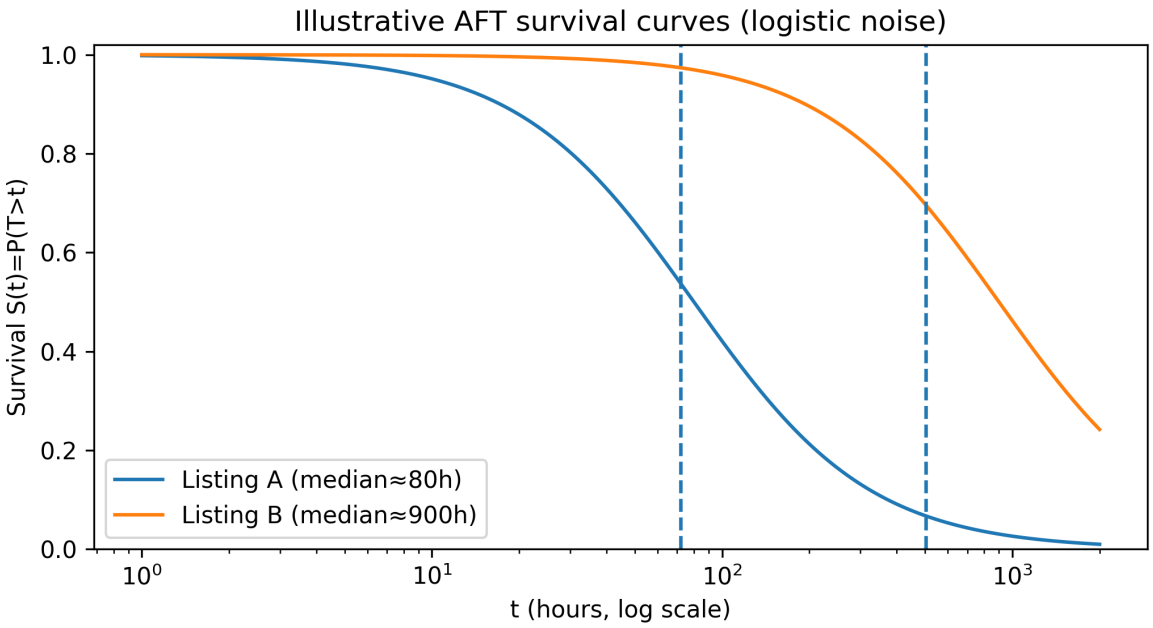

Figure 3.4: Illustrative survival curves and the operational boundaries at 72h and 504h.

3.4.1 Boundary-aligned loss shaping

A naive survival model will allocate most of its capacity to the dense front of the distribution—hours to a few days—because those regions dominate both sample counts and gradient mass. However, for zombie avoidance the relevant errors occur near and beyond 21 days. The Slow-21 gate therefore shapes the effective loss landscape through three complementary mechanisms:

(i) Recency weighting. Marketplace conditions drift (seasonality, cohort effects, supply shocks). The model uses exponential time-decay with half-life h days, normalized to keep the average weight at 1:

$$w_i^{\text{time}} = \frac{2^{-a_i/h}}{\mathbb{E}[2^{-a/h}]}$$

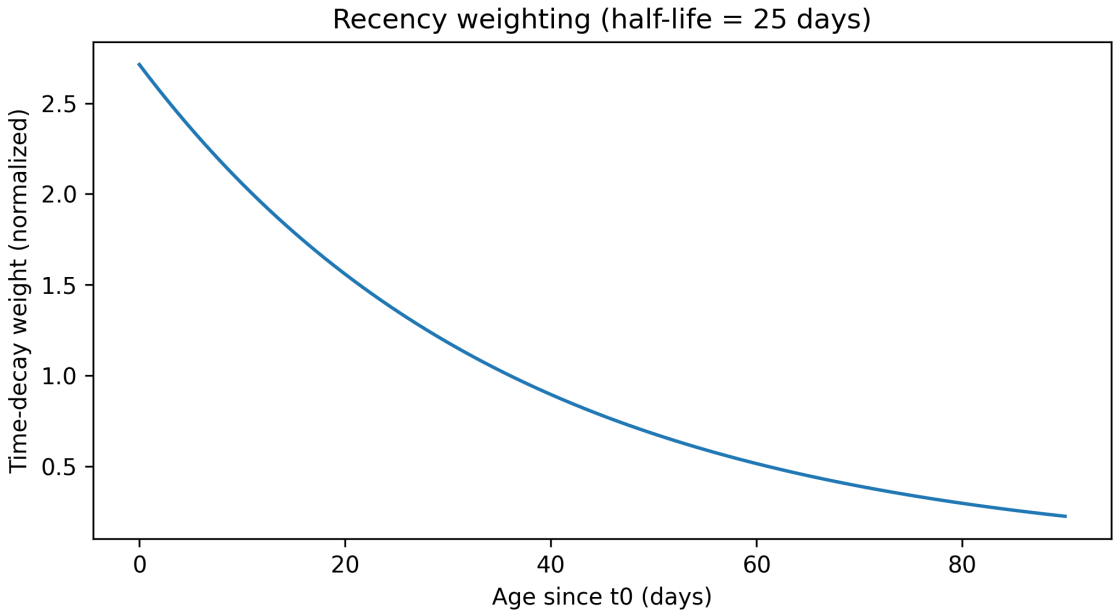

Figure 3.5: Recency weighting with a 25-day half-life (normalized).

(ii) Tail emphasis. To ensure the model allocates sufficient capacity to the slow regime, event samples beyond 7 days (168h) and beyond 21 days (504h) receive multiplicative up-weighting. With parameters (w_{slow}, $w_{\text{very-slow}}$), the multiplier is:

```
w_tail(y) = 1·1[y≤168h] + w_slow·1[168h<y≤504h] + (w_slow·w_very_slow)·1[y>504h].
```

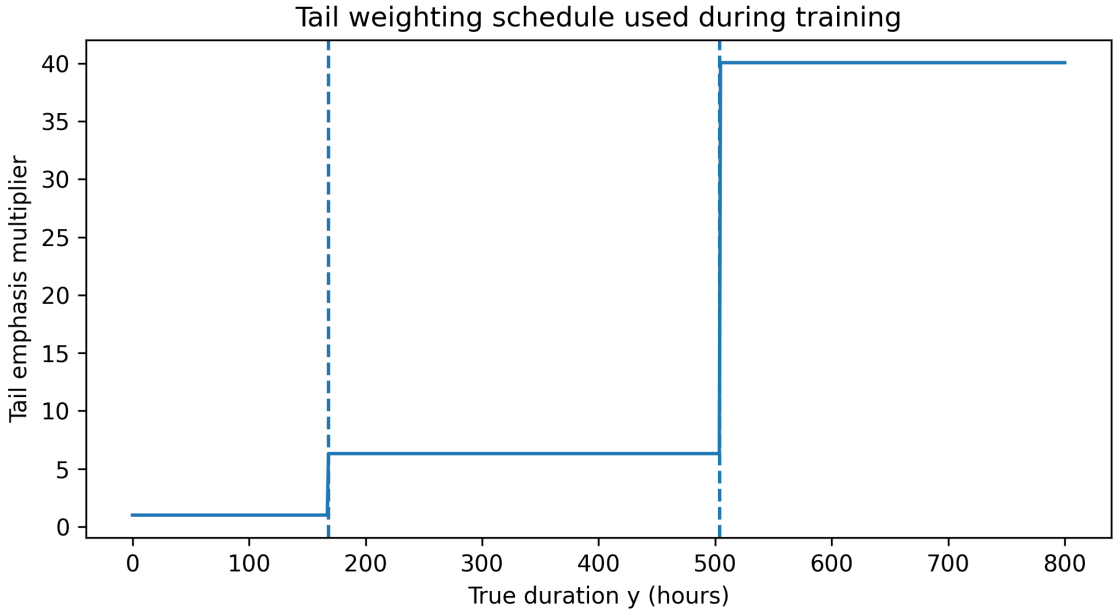

Figure 3.6: Tail emphasis schedule (illustrative parameters from the best trial).

(iii) Boundary focus. Finally, the system boosts gradient signal in a narrow neighborhood around the operational boundary τ=504h. This is implemented as a smooth bump function centered at τ and normalized to mean 1:

$$w_i^{\text{bnd}} = \frac{1 + k\exp\left(-\left|\frac{y_i - \tau}{\sigma_b}\right|^2\right)}{\mathbb{E}\left[1 + k\exp\left(-\left|\frac{y - \tau}{\sigma_b}\right|^2\right)\right]}$$

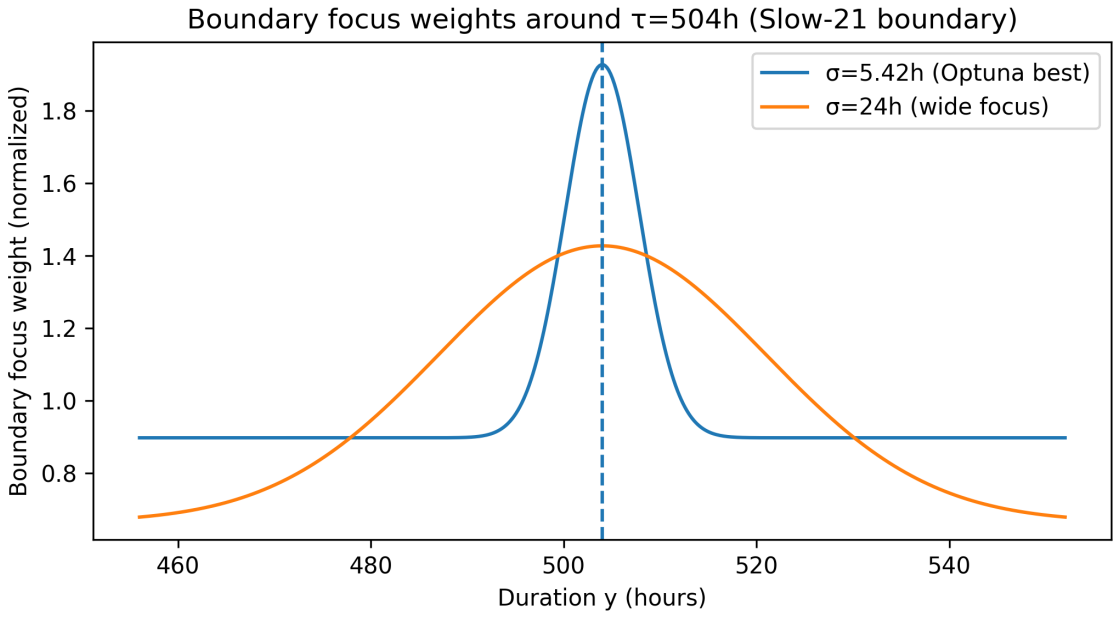

Figure 3.7: Boundary focus weights around τ=504h. Smaller σ concentrates learning near the boundary.

3.4.2 Hyperparameter tuning as boundary risk minimization

Hyperparameters are tuned using Bayesian optimization (Optuna) with an objective aligned to the business decision: maximize Slow-21 F1 on a held-out evaluation slice consisting of recently sold listings. Formally, with predicted gate labels $\hat{y}$ and true gate labels y, the objective is:

```
F1_slow21 = 2·(precision·recall)/(precision+recall),  where
precision = TP/(TP+FP),  recall = TP/(TP+FN).
```

The tuning loop trains an AFT model for each candidate hyperparameter set, applies monotone calibration (Section 3.4.3), computes the induced gate classification at τ=504h, and returns F1. Critically, the objective is computed on an evaluation slice anchored to the most recent sold outcomes, which improves robustness under drift and aligns the gate with the current market regime.

Best trial (example run). An illustrative run reached Slow-21 F1 = 0.8462 on the held-out sold-last-7-days slice. The tuned parameters emphasize stable learning (small num_leaves, moderate min_data_in_leaf), strong regularization ($\lambda_2 \approx 9.9$), and explicit tail/boundary emphasis:

```
         learning_rate : 0.012325905936212151
            num_leaves : 22
       min_data_in_leaf : 27
       feature_fraction : 0.5743839511722094
       bagging_fraction : 0.9493826345890998
              lambda_l2 : 9.91609810506581
              aft_scale : 1.4707316349984765
       slow_tail_weight : 6.309081750766779
  very_slow_tail_weight : 6.34946339903477
       boundary_focus_k : 1.1493500554722806
   boundary_focus_sigma : 5.420730276951488
```

3.4.3 Boundary-aware isotonic calibration of predicted time

Tree-based survival models often exhibit systematic calibration error in the tails: extreme predictions regress toward the mean, and the mapping from raw model score to realized time is not perfectly linear. To reduce boundary error at τ=504h, the pipeline fits a monotone calibration map g(·) using isotonic regression on a recent sold slice. The calibrator is trained on pairs ($\hat{T}$, T) with optional sample weights that up-weight durations near τ, and the final predicted time is $\hat{T}$_cal = g($\hat{T}$).

```
Inputs:
  - pred_cal: raw model predictions (hours) on a recent SOLD slice
  - y_cal:   observed durations (hours) on the same slice
  - cal_w:   optional boundary-focus weights around τ=504h

Fit:
  g = IsotonicRegression(out_of_bounds="clip")
  g.fit(pred_cal, y_cal, sample_weight=cal_w)

Apply:
  T^_cal = g.predict(T^_raw)
```

Isotonic regression is particularly well-suited here because it enforces monotonicity (higher predicted time should not map to lower calibrated time) while remaining flexible enough to correct nonlinear distortions. The “out_of_bounds=clip” setting ensures stable extrapolation for rare extreme predictions.

3.4.4 Evaluation: accuracy where it matters and the cost of false zombies

A tail gate must balance two competing failure modes: (i) false zombies (FP): labeling a healthy listing as slow, causing missed opportunity; and (ii) missed zombies (FN): labeling a slow listing as fast, trapping capital in illiquid inventory. Standard precision/recall/F1 at τ=504h captures the high-level trade-off, but operational deployment requires an additional cost-sensitive lens: the system explicitly measures how often the gate mistakenly rejects very good deals.

Sacrifice metrics. Define LT10 = 240h (10 days) and MID = [240h, 504h). Two sacrifice rates are computed:

```
SAC_LT10 = P( ŷ_slow21 = 1  |  T < 240h )
SAC_MID  = P( ŷ_slow21 = 1  |  240h ≤ T < 504h )
```

SAC_LT10 is the critical guardrail: it quantifies the share of very fast sales that are wrongly rejected as zombies. SAC_MID is less costly and can be tuned more aggressively to reduce zombie exposure.

Slice	Prec (slow)	Rec (slow)	F1 (slow)	SAC_LT10	SAC_MID	TP	FP	FN	TN
Eval sold last 7d	0.8314	0.8614	0.8462	0.0059	0.2688	143	29	23	746
Cal (recent sold)	0.6532	0.9474	0.7733	0.0143	0.6341	162	86	9	596

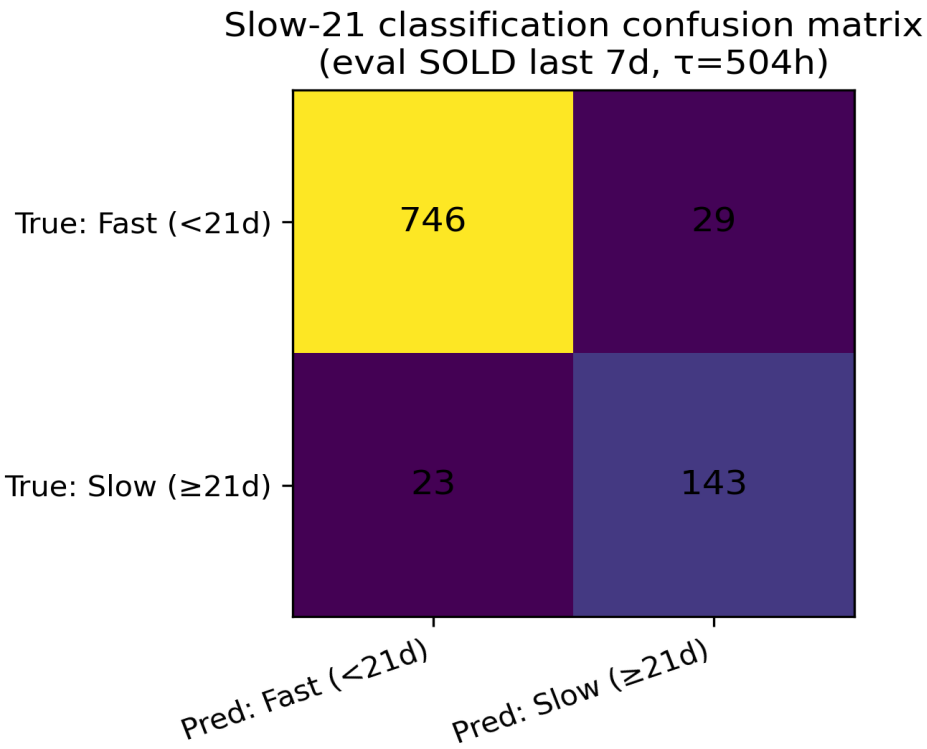

Figure 3.8: Confusion matrix for the Slow-21 gate on the evaluation slice.

3.4.5 Why anchors dominate: interpretability and sample efficiency

An important diagnostic for a tail gate is whether it learns interpretable, stable drivers of liquidity rather than transient noise. In the evaluated run, the top features by gain importance are anchor-centric, indicating that the model's dominant signal is the relative position of a listing against its segment's certified market baseline.

Feature	Gain importance (% of total)
ptv_anchor_strict_t0	28.153%
ptv_anchor_smart	25.452%
ptv_sold_30d	4.146%
member_since_year__missing	2.349%
ai_lqs_textonly_f	0.902%
review_count__missing	0.837%
aff_decile_in_seg	0.715%
allgen_30d_post_count	0.686%
image_count__missing	0.671%
anchor_n60_t0	0.657%
speed_median_hours_ptv	0.622%
battery_pct_effective	0.605%

The top three anchor features alone account for approximately 57.8% of gain importance in this run (ptv_anchor_strict_t0, ptv_anchor_smart, ptv_sold_30d). This is a desirable outcome: it means that the model's primary mechanism for tail discrimination is grounded in stable market priors rather than brittle micro-signals.

The remaining features serve mainly to adjust the anchor-based baseline: seller history (member_since_year), text-quality flags, socio-economic context, supply flow indicators (allgen_30d_post_count), and media/vision completeness signals. Such a decomposition is precisely what one seeks in long-tail modeling: the anchor encodes "what should happen," and the model learns "why this case deviates."

3.5 Fast-72 sniper classifier: complementary short-horizon signal

The slow tail is only half the liquidity story. Many operational decisions—especially deal selection—benefit from an explicit model of very fast sales. The Fast-72 classifier provides this complementary signal by predicting whether a listing will sell within 72 hours.

Label and training set. The binary label is defined on sold events as $y^{fast72}=1[T\leq 72h]$. Unlike Slow-21, Fast-72 does not require explicit censoring machinery because the

label is naturally observed for sold items. However, the training set still benefits from recency weighting to track market drift.

Model. The classifier is trained with XGBoost's binary:logistic objective, using tree regularization, subsampling, and optional monotone constraints. A typical training routine:

```
1) Load t0 feature matrix X and sold labels (duration_h, sold_event).
2) Define y_fast72 = 1[sold_event=1 ∧ duration_h ≤ 72].
3) Apply time-decay weights w_time based on listing age.
4) Train XGBoost binary:logistic with early stopping on an eval slice.
5) Select the decision threshold that maximizes F1_72 on eval:
       thr* = argmax_thr F1(y_eval, p_eval ≥ thr)
6) Persist model + threshold for downstream scoring.
```

In practice, Fast-72 captures short-horizon liquidity driven by immediate underpricing relative to the anchor, high listing quality, and strong buyer demand. When used jointly with Slow-21, it supports an explicit "snipers vs. zombies" decomposition instead of forcing the core survival model to span the entire time range with equal fidelity.

3.6 Integration: three-model architecture for pricing and deal selection

A key design decision is to treat Fast-72, core survival, and Slow-21 as a coordinated system rather than independent predictors. All three share the same leak-proof t0 feature interface, which enables consistent interpretation and stable deployment.

Why not a single model? A single global survival model is theoretically sufficient, but in practice it is suboptimal for three reasons: (i) the loss is dominated by the dense front, starving the tail of gradient signal; (ii) operational actions are discrete and boundary-driven; and (iii) error asymmetry is extreme—false zombies are costly but missed zombies are also costly. Dedicated boundary models let the system allocate capacity and calibration exactly where decisions occur.

Mixture-of-experts view. Let $p_{fast}=P(T\le 72h|x_0)$ from Fast-72, and $p_{slow}=P(T\ge 504h|x_0)$ induced by the AFT gate (or by mapping calibrated $\hat{T}$ to a probability). A simple and effective integration pattern is:

```
Score(listing) =  α · log(price / anchor)  +  β · p_fast  −  γ · p_slow  −  λ · uncertainty,
```

where $\alpha,\beta,\gamma,\lambda$ are policy parameters (or learned coefficients) that encode the institution's appetite for speed versus value versus risk. Because anchors provide a stable baseline, the score operates mainly on relative mispricing. The gates then supply explicit liquidity risk.

Interaction with the full survival model. The full AFT survival model remains essential: it provides a continuous survival curve $S(t|x_0)$, median and tail quantiles, and expected time-to-sale. In production, the gates can be used as: (i) additional features to the survival model, (ii) post-model adjustments to enforce guardrails (e.g., reject if p_slow

exceeds a threshold), or (iii) routing signals that select different downstream strategies (aggressive pricing for snipers, conservative for zombies).

3.7 Empirical results and diagnostics

This section summarizes representative outcomes from an end-to-end training run of the Slow-21 gate under the certified feature-store stack. The emphasis is on tail discrimination quality, stability diagnostics, and interpretability.

Dataset composition (representative run). Base dataset size was 35,073 rows. Of these, 21,758 had observed sold outcomes, with 20,816 sold samples used for training and 942 recent sold samples reserved for evaluation ("sold last 7d"). In addition, 4,666 listings were treated as right-censored at ≥21 days (unsold beyond the censoring minimum).

Tail discrimination. On the evaluation slice, the gate achieved precision 0.8314 and recall 0.8614 for the slow class, corresponding to F1 = 0.8462. Recall at this level is critical: it means that the majority of true zombies are identified before capital is committed.

Opportunity cost control. The sacrifice rate SAC_LT10 = 0.0059 indicates that fewer than 1% of listings that would sell within 10 days are misclassified as slow. This is a strong operational guarantee: the model can be used as an aggressive zombie filter without significantly suppressing high-velocity deals.

Anchor dominance and stability. As shown in Section 3.4.5, anchor features dominate gain importance. This is consistent with the hypothesis that long-tail outcomes are primarily driven by mispricing relative to the local market baseline. When anchors are leak-proof and stable, the model's behavior remains stable under drift because the dominant inputs track the evolving market.

3.8 Practical guidance and extensions

The Slow-21 gate and hierarchical anchors are designed to be production-robust: they expose stable, interpretable controls (support thresholds, backoff levels, half-life, tail weights) that can be tuned without redesigning the learning system. The following practices have proven essential:

• Treat decision boundaries as first-class: tune and validate explicitly at 72h and 504h rather than relying on generic global metrics.
• Anchor everything: prefer relative-to-anchor features (ratios, deltas) over absolute prices to reduce segment drift sensitivity.
• Separate stability from learning: put stability rules, backoff, and certification in the feature store; keep the model focused on residual structure.
• Use loss shaping deliberately: recency weights track drift, tail weights ensure capacity allocation, boundary focus aligns gradients with the decision.• Monitor SAC_LT10 continuously: it is the principal early-warning signal that the zombie filter is harming good deals.

Extensions. Two natural extensions follow directly from the architecture. First, incorporate explicit inventory/stock signals (concurrent listings in the same segment at t0) to capture short-term supply pressure; this is expected to improve tail separation in high-inventory cohorts. Second, elevate the mixture-of-experts view into a formal probabilistic mixture model where Fast-72 and Slow-21 gates parameterize mixture weights, yielding a calibrated full distribution with sharper tail behavior.

Summary. The core thesis of this chapter is that tail capture in marketplace survival modeling is achieved not by a single monolithic learner, but by a coordinated system:

hierarchical Bayesian anchors provide stable, leak-proof priors; boundary-focused gates align learning with decisions; and survival models provide continuous estimates for downstream planning. Together, these components transform the long tail from a source of irreducible uncertainty into a tractable, measurable, and controllable risk.

References

[1] D. R. Cox and D. Oakes. Analysis of Survival Data. Chapman & Hall, 1984.
[2] J. P. Klein and M. L. Moeschberger. Survival Analysis: Techniques for Censored and Truncated Data. Springer, 2003.[3] T. Hastie, R. Tibshirani, and J. Friedman. The Elements of Statistical Learning. Springer, 2009.
[4] XGBoost Documentation: Accelerated Failure Time (AFT) Survival Objective (survival:aft).