BOX A.4: SELECTION OF FEATURES AND PRUNING METHODS

Modelling features in this project are intentionally rich: target-specific signals, peer information, macro and earnings releases, and their event-aware transforms (gates/decays), typically present in both raw and standardized variants. Left unfiltered, this breadth risks variance inflation – weak, redundant, or degenerate columns that add noise and might hinder performance rather than add predictive power. To compress and optimize the feature space, a robust, multi-level pruning process was applied using complementary methods:

1. Deterministic prefiltering: within each path**:** **(a)** standardized twins were preferred (if a raw level and a standardized/de-meaned version coexist, the raw is dropped and the standardized variant is kept); while **(b)** a lightweight correlation screen prunes features whose association with the target is negligible: columns with absolute Pearson (and, when invoked, absolute Spearman) below **0.10** or undefined due to missing overlap are discarded.. In practice, this step *eliminates families of near-random event gates and miscellaneous low-signal artifacts* while leaving intact standardized, economically grounded signals.

2. Leakage-safe model-based pruning: two complementary selectors follow pre-filtering:

(A) Permutation Importance (PI): *wi*thin each outer fold, the training window is split chronologically using `*val_frac = 0.25*` into (i) a *sub-train slice*—the earlier 75% used to fit a lightweight RidgeCV scoring model—and (ii) an *inner validation slice*—the later 25% used only for evaluation and entirely preceding the fold's test window. On the inner validation slice, each feature is shuffled in isolation; predictions are recomputed, and the resulting drop-in hit rate is recorded. The shuffle is repeated several times per feature and averaged to yield a per-fold PI score. PI is expressed in percentage points of hit (e.g., PI = 0.01 implies a 1pp average reduction in validation hit when the feature is destroyed). PI scores are then averaged across outer folds, and features with mean Δhit at or below the chosen cutoff (e.g., ≤ 1pp) are removed. Unlike correlation, which reflects pairwise association, PI quantifies incremental contribution given the rest of the feature set; a column can appear correlated yet contribute zero—or negative—Δhit once competing predictors are present. This method systematically removes features that harm validation performance, yielding a cleaner, more parsimonious set before any model is evaluated on the fold's test window.

(B) Stability selection with Ridge. For each outer fold, a RidgeCV model is fitted on the fold's training window and features are ranked by the absolute value of their coefficients (larger |coef| ⇒ stronger contribution within that fold). The top 40 features from each fold are recorded. After processing all folds, a feature is kept only if it appears in at least 60% of the folds. This rule rewards predictors that recur across time and filter out one-off, regime-specific spikes. Because the ranking is computed using only the training window in each fold, the selection remains leakage safe.

The two filters are combined conservatively: first PI-neutral/harmful columns are dropped, then intersect with the stability keepers. If this intersection becomes too thin to learn from, a safety floor restores a "drop-harmful-only" set so the path remains trainable.

3. Top K feature selection for XGB (hybrid step): after PI and stability selection, the remaining features form a compact, time-robust pool. The pipeline then fixes **K per path** so that XGB trains only on the strongest slice of this pool. This stage uses the cached per-fold Ridge rankings *(one ordered list of features per fold, produced on that fold's training window by RidgeCV and saved* *for reuse—features are sorted by absolute coefficient, highest first)*, which order the surviving features. Next, a small grid of K values (e.g., 20, 30, 40, 60) is tried (using the Fast K-sweep). For each fold and each K, only the top-K in that fold's cached order are passed to a lightweight, early stopping XGB that is tuned on the time-ordered inner validation slice of the training window and then scored on the fold's test window. Metrics are recorded in hit rate (primary) and Spearman rank correlation (secondary). Because rankings are cached, the sweep is computationally cheap and remains leakage safe. The K that yields the highest average hit rate across folds is chosen; ties are broken by higher average Spearman, and if still tied, by the smaller K for parsimony.

Contribution to modelling. The selection-and-pruning pipeline *reduces variance without amputating core signal*. Standardization ensures comparability; the correlation screen trims obvious dead weight; PI removes features that fail to move validation hit; stability selection enforces time-robustness; and the hybrid Ridge/XGB interface—by training XGB only on a fold-appropriate feature set—focuses non-linear capacity on a curated subset. All filtering steps live inside a walk-forward protocol with an embargo equal to the prediction horizon, ensuring realistic out-of-sample timing (no training label depends on outcomes from the test window). *The result is a smaller, cleaner, and more stable feature set for each path*, accompanied by supporting documentation: PI tables, lists of removed "harmful/neutral" columns, stability counts, and the final feature inventories used for training.

Referenced Functions (by name): *build_model_frame*, *prefer_standardized*, *drop_unrelated_features*, *audit_corr_nan*, *audit_lowcorr*, *apply_model_based_filters*, *perm_importance_by_fold_val*, *ridge_stability_select*, *cache_ridge_rankings*, *sweep_K_with_cached_rankings*, *pick_best_K*, *run_path*