☑ Data Pipeline To-Do List for Copilot Agent

Goal: Implement a full pipeline for **data cleaning**, **preprocessing**, **feature engineering**, and **model selection** for the LOE/generic erosion forecasting project.

---

1. Data Cleaning & Preprocessing

1.1 Basic Sanity Checks

- [ ] Remove exact duplicate records based on:
  - [ ] `country`
  - [ ] `brand_name`
  - [ ] `months_postgx`
- [ ] For each brand, verify:
  - [ ] No multiple rows exist for the same `months_postgx`.
  - [ ] `months_postgx` spans a reasonable range (e.g. -24 to +23).
- [ ] Decide how to handle brands with very short history:
  - [ ] Keep them and rely on global models that share information across brands.

1.2 Handle Missing Values

- [ ] For `time_to_50pct` (train only):
  - [ ] Create `reached_50pct` = 1 if `time_to_50pct` is not null, else 0.
  - [ ] Create `time_to_50pct_imputed`:
    - [ ] Use real `time_to_50pct` when available.
    - [ ] Use `24` (or last observed month) when not reached.
- [ ] For `avg_vol` (`avg_j`):
  - [ ] Recompute as mean volume over months -12..-1 where data exists.
  - [ ] If less than 12 months of pre-LOE data exist:
    - [ ] Impute using median `avg_vol` by therapeutic area **or**
    - [ ] Fit a simple regression of log(avg_vol) on `country`, `therapeutic_area`, `hospital_rate`, etc.
- [ ] For `n_gxs` (number of generics):
  - [ ] If missing, set `n_gxs` to 0 (after verifying this is consistent with the structure).
  - [ ] Check NA patterns and confirm assumption is safe.

1.3 Outlier Treatment

- [ ] For `n_gxs`:
  - [ ] Create `n_gxs_capped = min(n_gxs, 15)`.
  - [ ] Optionally create `log1p(n_gxs_capped)`.
- [ ] For `volume` and `vol_norm`:
  - [ ] Keep `vol_norm` as main signal for modeling.
  - [ ] Use `volume` mainly for revenue/ROI calculations.

- [ ] Do **not** remove rows with `vol_norm > 1`.
- [ ] Create `vol_norm_gt1` = 1 if `vol_norm > 1` else 0.

1.4 Scaling (Where Needed)

- [ ] For models needing scaling (e.g. neural/linear):
  - [ ] Leave `months_postgx` raw or standardize (mean 0, std 1).
  - [ ] Apply `log1p` + standardization to `n_gxs_capped`.
  - [ ] Keep `hospital_rate` as-is or use MinMaxScaler.
  - [ ] Apply `log1p` + standardization to `avg_vol`.
  - [ ] Keep `vol_norm` unscaled for interpretability.

1.5 Categorical Encoding (Leakage-Safe)

- [ ] Identify key categorical variables:
  - [ ] `country`
  - [ ] `therapeutic_area`
  - [ ] `main_package`
- [ ] Implement target encoding for these variables:
  - [ ] Encode using mean `vol_norm` **or** mean bucket probability.
  - [ ] Compute encodings **within** CV folds (train only, apply to val).
- [ ] Create additional categorical-driven features:
  - [ ] `ther_area_erosion_rank` = rank of therapeutic area by average erosion.

---

2. Feature Engineering

2.1 Time-Based Features

- [ ] Create base time features:
  - [ ] `months_postgx`
  - [ ] `months_postgx_sq = months_postgx ** 2`
- [ ] Create period flags:
  - [ ] `is_early_period` = 1 if 0 ≤ `months_postgx` ≤ 5 else 0.
  - [ ] `is_mid_period` = 1 if 6 ≤ `months_postgx` ≤ 11 else 0.
  - [ ] `is_late_period` = 1 if 12 ≤ `months_postgx` ≤ 23 else 0.
- [ ] (Optional) Create pre-LOE buckets for pre-entry analysis.

2.2 Competition Features (`n_gxs`)

- [ ] Create competition features:
  - [ ] `n_gxs_capped` = min(`n_gxs` , 15).
  - [ ] `log_n_gxs` = log1p(`n_gxs_capped`).
  - [ ] `has_competition` = 1 if `n_gxs` > 0 else 0.
  - [ ] `high_competition` = 1 if `n_gxs` ≥ 5 else 0.
  - [ ] `competition_intensity` = `n_gxs / (months_postgx + 1)` for post-LOE rows.

- [ ] Brand-level competition features (train / analysis):
  - [ ] `max_n_gxs_post` = max `n_gxs` post-LOE.

2.3 Lag & Rolling Features

- [ ] For each `(country, brand_name)`:
  - [ ] Add lags of `vol_norm`:
    - [ ] `vol_norm_lag1`
    - [ ] `vol_norm_lag3`
    - [ ] `vol_norm_lag6`
  - [ ] Add differences and changes:
    - [ ] `vol_norm_diff1` = `vol_norm - vol_norm_lag1`
    - [ ] `vol_norm_diff3` = `vol_norm - vol_norm_lag3`
    - [ ] `vol_norm_pct_change` = `(vol_norm - vol_norm_lag1) / (vol_norm_lag1 + ε)`
  - [ ] Add rolling statistics:
    - [ ] `vol_norm_roll_mean_3`
    - [ ] `vol_norm_roll_mean_6`
    - [ ] `vol_norm_roll_std_3`
    - [ ] `erosion_rate_3m` = `vol_norm_lag3 - vol_norm`
- [ ] For Scenario 1:
  - [ ] Ensure lag/rolling features for test brands are computable from **pre-LOE** data only.
- [ ] For Scenario 2:
  - [ ] Include lag/rolling features based on months 0–5 at forecast origin (month 6).

2.4 Brand-Level Static Features

- [ ] From pre-LOE history, compute:
  - [ ] `avg_vol` / `avg_j` (recomputed baseline volume).
  - [ ] Pre-LOE trend: slope of `vol_norm` vs time for months -12..-1.
  - [ ] Pre-LOE volatility: std of `vol_norm` in months -12..-1.
  - [ ] `pre_loe_growth_flag` = 1 if trend slope > 0 else 0.
- [ ] From EDA-derived metrics (train only):
  - [ ] `time_to_50pct_imputed`
  - [ ] `reached_50pct`
- [ ] From categories:
  - [ ] `ther_area_erosion_rank`
  - [ ] `is_high_erosion_area`
  - [ ] `hospital_rate_bucket` ∈ {0–25, 25–50, 50–75, 75–100}.

---

3. Model Selection & Training

3.1 Use Global Models (Not Per-Brand ARIMA)

- [ ] Choose global models that learn across all brands:
  - [ ] LightGBM / XGBoost / CatBoost for tabular modeling.

    - [ ] (Optional) Add LSTM / Temporal Fusion Transformer for time-series deep learning.
- [ ] Avoid per-brand ARIMA due to scale and complexity unless specifically needed for a hybrid baseline.

3.2 Scenario-Specific Pipelines

Scenario 1 (No Post-Entry Data)

- [ ] Define features available at t = 0 (pre-LOE only).
- [ ] Implement one of the following strategies:
    - [ ] **Horizon-as-row**:
        - [ ] Build a dataset where each row corresponds to (brand, horizon `h` in [0–23]).
        - [ ] Include:
            - [ ] Brand-level static features.
            - [ ] Pre-LOE time-series summary features.
            - [ ] Horizon `h` as a feature.
        - [ ] Target: `vol_norm_h`.
    - [ ] **Separate models per horizon**:
        - [ ] Train 24 models, one for each forecast month (0–23).
        - [ ] Each model uses the same set of origin features but a different target.
- [ ] Use LightGBM (or similar) with sample weights reflecting bucket importance and time-window importance.

Scenario 2 (First 6 Months Known)

- [ ] At forecast origin (month 6), define features including:
    - [ ] All brand-level and static features.
    - [ ] Early post-LOE behavior (months 0–5):
        - [ ] Mean `vol_norm` 0–5.
        - [ ] Trend (slope) 0–5.
        - [ ] Last observed `vol_norm` at month 5.
        - [ ] Rolling stats over 0–5.
- [ ] Use the same modeling strategy as Scenario 1 (horizon-as-row or per-horizon model), but applied to horizons 6–23.

3.3 Bucket 1 vs Bucket 2 Handling

- [ ] Implement a Bucket-aware strategy:
    - [ ] **Option A – Sample weights only**:
        - [ ] For all training rows from Bucket 1 brands, set sample weight = 2.
        - [ ] For all training rows from Bucket 2 brands, set sample weight = 1.

    - [ ] **Option B – Separate regressors** (optional enhancement):
        - [ ] Train `Model_B1` on Bucket 1 brands (with oversampling or weights).
        - [ ] Train `Model_B2` on Bucket 2 brands.
        - [ ] (Optional) Train a classifier to predict bucket and blend outputs at prediction time.
- [ ] Start with Option A for simplicity.

3.4 Align Loss with Competition Metric via Sample Weights

- [ ] For each training row (brand, horizon `h`):
    - [ ] Determine which time window `h` belongs to:
        - [ ] Scenario 1: 0–5, 6–11, 12–23.
        - [ ] Scenario 2: 6–11, 12–23.
    - [ ] Assign a **time-window weight** approximating the metric's emphasis:
        - [ ] Higher weights for 0–5 (Scenario 1) and 6–11 (both scenarios).
        - [ ] Lower weights for later months.
    - [ ] Multiply by bucket weight (2 for Bucket 1, 1 for Bucket 2).
- [ ] Pass final weights as `sample_weight` to LightGBM/XGBoost.

3.5 Hybrid / Ensemble Strategy

- [ ] Implement a simple **physics baseline**:
    - [ ] For each brand, fit exponential decay for post-LOE:
        - [ ] `vol_norm(t) ≈ a * exp(-b * t) + c`.
    - [ ] Use **normalized volume** as the modeled quantity.
- [ ] Implement a **global ML model** (LightGBM):
    - [ ] Either predict `vol_norm` directly or predict residuals:
        - [ ] `residual = true_vol_norm - physics_baseline(t)`.
- [ ] (Optional) Add ARHOW or ARIMA+Holt-Winters forecasts:
    - [ ] Use them as extra features or as a third ensemble component.
- [ ] Combine ensemble components:
    - [ ] Define:
        - [ ] `y_phys` = physics baseline prediction.
        - [ ] `y_ml` = ML prediction.
        - [ ] (Optional) `y_ts` = ARHOW/TS prediction.
    - [ ] Fit blending weights on validation data:
        - [ ] `y_pred = w_phys * y_phys + w_ml * y_ml (+ w_ts * y_ts)`.

---

4. Cross-Validation Design

- [ ] Use **GroupKFold** or **StratifiedGroupKFold** for CV:

    - [ ] Group by `brand_name`.
    - [ ] If possible, stratify by `bucket` to balance Bucket 1 and 2 across folds.
- [ ] Ensure:
    - [ ] All months for each brand appear in only one fold (train or validation, not both).
    - [ ] Each fold has a realistic distribution of high- and low-erosion brands.
- [ ] In each fold, simulate Scenario 1 and Scenario 2 forecast origins and compute metric approximations.

---

5. High-ROI Minimal Stack (Implementation Priority)

- [ ] **Cleaning**:
    - [ ] Cap `n_gxs`, create `log_n_gxs`.
    - [ ] Recompute `avg_vol` from -12..-1.
    - [ ] Encode `therapeutic_area` into an erosion-based rank.
- [ ] **Features**:
    - [ ] Time indicators: `months_postgx`, `is_early_period`, `is_mid_period`, `is_late_period`.
    - [ ] Competition features: `log_n_gxs`, `has_competition`, `high_competition`.
    - [ ] Lags: `vol_norm_lag1`, `vol_norm_lag3`, `vol_norm_diff1`, `vol_norm_roll_mean_3`.
    - [ ] Brand static: `log_avg_vol`, `hospital_rate_bucket`, `is_high_erosion_area`.
- [ ] **Models**:
    - [ ] Scenario 1:
        - [ ] Use horizon-as-row LightGBM with time-window + bucket sample weights.
    - [ ] Scenario 2:
        - [ ] Same structure, with the addition of early post-LOE summary features (0–5).
- [ ] **Ensemble**:
    - [ ] Add an exponential-decay baseline and blend with LightGBM predictions.

This To-Do list provides a concrete set of tasks for Copilot (or any agent) to turn the conceptual plan into a working pipeline.