RFECV + LightGBM

1. Task Objective

Target variable: Tc_target (whether a fracture will occur in the next 30 days, binary classification).

Data source: feature_selection.preprocess_training_table

Goal: Perform wrapper feature selection; select the most useful features for prediction without temporal leakage, and present the performance (PR-AUC, F1, Accuracy) on an independent test set.

2. Data Preprocessing and Feature Candidate Space

2.1 Aggregation and Cleaning

Group by (Tc_BaseCode, Tc_BaseCode_Mapped, Tc_SectionBreakStartKM, Tc_r_date) and perform mean aggregation on numeric columns to avoid meaningless aggregation of string/date columns.

Exclude irrelevant columns (Wagon_ICWVehicle).

After averaging, if an entire group is missing, resulting in NULL, fill it with 0.

Only retain the numeric Wagon_* features + w_row_count + Tng_Tonnage as candidate features. Do not consider ID/geolocation fields as model features to reduce the risk of leakage/overfitting.

2.2 Time Series Sorting and Partitioning

Sort by Tc_r_date (pdf["_order"] = to_datetime(Tc_r_date); np.argsort followed by rearrangement).

Time Split:

Train/Test: Split the dataset 80/20 by time, ensuring that the test set completely follows the training set (to prevent future information leakage).

Cross-Validation: Use TimeSeriesSplit (n_splits=5) for time-aware CV.

3. Wrapper Feature Selection Design

3.1 Base Learner (Estimator)

Use LightGBM (LGBMClassifier) as the estimator

3.2 RFECV (Recursive Feature Elimination + CV)

Tool: sklearn.feature_selection.RFECV

Result: A set of selected features.

```
Selected features: ['Wagon_Twist14m', 'Wagon_BounceFrt', 'Wagon_BounceRr', 'Wa
gon_LP1', 'Wagon_LP2', 'Wagon_Speed', 'Wagon_BrakeCylinder', 'Wagon_IntrainFor
ce', 'Wagon_Acc2', 'Wagon_Acc4', 'Wagon_Twist2m', 'Wagon_Acc1_RMS', 'Wagon_Acc
3_RMS', 'Wagon_Rail_Pro_L', 'Wagon_Rail_Pro_R', 'Wagon_SND', 'Wagon_VACC_R',
'Wagon_Curvature', 'Wagon_SND_L', 'Wagon_SND_R', 'Tng_Tonnage']
```

4. Threshold Tuning (F1)

On the validation fold of the TimeSeriesSplit of the training set, use precision_recall_curve to scan different thresholds and calculate F1 = 2 * P * R / (P + R).

Find the threshold with the highest F1 for each fold and take the median as the global optimal threshold, best_thr.

This approach not only considers the precision-recall trade-off under imbalanced conditions, but also reduces reliance on chance in a single fold.

5. Training and Evaluation Process (Avoiding Leakage)

RFECV is performed on the training set X_tr, y_tr , selected_feats is obtained;Use selected_feats to adjust the threshold in the training set using temporal cross-validation (to obtain best_thr); The final LightGBM is trained using the entire training set data plus selected_feats; Evaluate once on the independent test set X_te, y_te (using only the final threshold; no further parameter tuning is performed).

6. Test Result

```
=== TEST  ===
PR-AUC: 0.2335444999786159
F1    : 0.19374068554396423
ACC   : 0.7632644130839077
```

7. Future Improvement Plans

Stronger time series features: rolling window/lag, statistics (mean/variance/slope, last non-zero value, rate of change, etc.); seasonal coding.

Grouped Time CV: If the same road segment appears repeatedly, consider using Group-TimeSeriesSplit (group=BaseCode, Tc_SectionBreakStartKM) to prevent hidden leakage of "cross-folds of the same road segment."

Model Transformation Method: SFS+Transform