QRT Data Challenge (2021) – Submission Overview

Reconstruction of Liquid Asset Performance

Mohamed Mohamed El Bechir
ENS Challenge Data username: `mohamed_elbechir`

Task. For each row (day `ID_DAY`, target liquid asset `ID_TARGET`), predict the **sign** of the liquid return `RET_TARGET` using 100 illiquid returns `RET_*`. The official metric is the magnitude-weighted directional accuracy:

$$\text{Score} = \frac{\sum_i |y_i| \, \mathbf{1}[\text{sign}(\hat{y}_i) = \text{sign}(y_i)]}{\sum_i |y_i|}.$$

Validation protocol : I use **GroupKFold** cross-validation with groups=`ID_DAY` (day-wise split) All *label-dependent* steps (proxy selection and threshold calibration) are computed **within training folds only** or using **out-of-fold (OOF)** predictions.

Feature engineering. I train one model per target asset. The feature set is the same across assets, but the contextual features are computed separately for each asset and fold.

The feature set is constructed as follows:

- **Raw illiquid returns:** the 100 illiquid return columns `RET_*`.
- **Market state:** cross-sectional volatility `RET_MKT_STD`, defined as the row-wise standard deviation of illiquid returns.
- **Correlation proxy (training only):** for each target asset, the illiquid return `RET_k` with the highest absolute correlation with `RET_TARGET` on the training split. This proxy is used solely to prevent information leakage in contextual averages.
- **Proxy-excluded contextual averages:** `RET_MKT_MEAN_EXCL_PROXY` (market average excluding the proxy), and `RET_SEC1_MEAN_EXCL` (sector average excluding the proxy, based on `CLASS_LEVEL_1` when available).
- **Relative returns:** asset-level deviations from contextual averages, defined as `RET_i_REL_MKT = RET_i - RET_MKT_MEAN_EXCL_PROXY` and `RET_i_REL_SEC = RET_i - RET_SEC1_MEAN_EXCL`.

This yields **303 features per model** in the final feature space.

Model (implemented). For each `ID_TARGET`, I train a weighted **VotingRegressor** combining:

- **Ridge** pipeline: median imputation, standardization, and Ridge($\alpha = 3.0$),
- **XGBoost regressor** (boosted trees reducing bias, with native missing-value handling),
- **ExtraTrees regressor** (bagged randomized trees reducing variance, with native missing-value handling).

Training uses **target clipping** at $\tau = 0.07$:

$$\tilde{y} = \text{clip}(y, -0.07, 0.07),$$

while evaluation is performed using the original sign-based metric.

Post-hoc decision threshold : Because the metric depends only on the sign of predictions, I calibrate an additive decision threshold on OOF predictions:

$$\widehat{s} = \text{sign}(\hat{y} - \theta).$$

A grid search on OOF predictions selects a *robust* optimum $\theta =$ **-0.000078** (based on a smoothed score curve), and the final submission applies $\text{sign}(\hat{y} - \theta)$.

Key results :

- GroupKFold CV (by `ID_DAY`): **74.7872%** weighted accuracy, fold standard deviation **0.6467%**.
- OOF threshold calibration: robust optimum $\boldsymbol{\theta = -0.000078}$.
- Public leaderboard: **Top 3** position *at the time of submission.*

Metric	Score	Comment
Public Leaderboard	0.7508	Top 3 (at submission time)
Local CV (GroupKFold)	74.7872	Split by day
Gap	small	CV aligned with LB

Interpretability : Feature importance analysis is used as a qualitative diagnostic, suggesting that predictive contributions for a given target asset concentrate on a limited subset of illiquid assets and contextual features.

Provided material:

Jupyter notebook (full pipeline and diagnostics) and this overview.