MTH9899 Final Project

2021.5

Quanzhi Bi, Junhan Wang, Liang Shan

Outline

01 Project Structure

02 Data Preparation

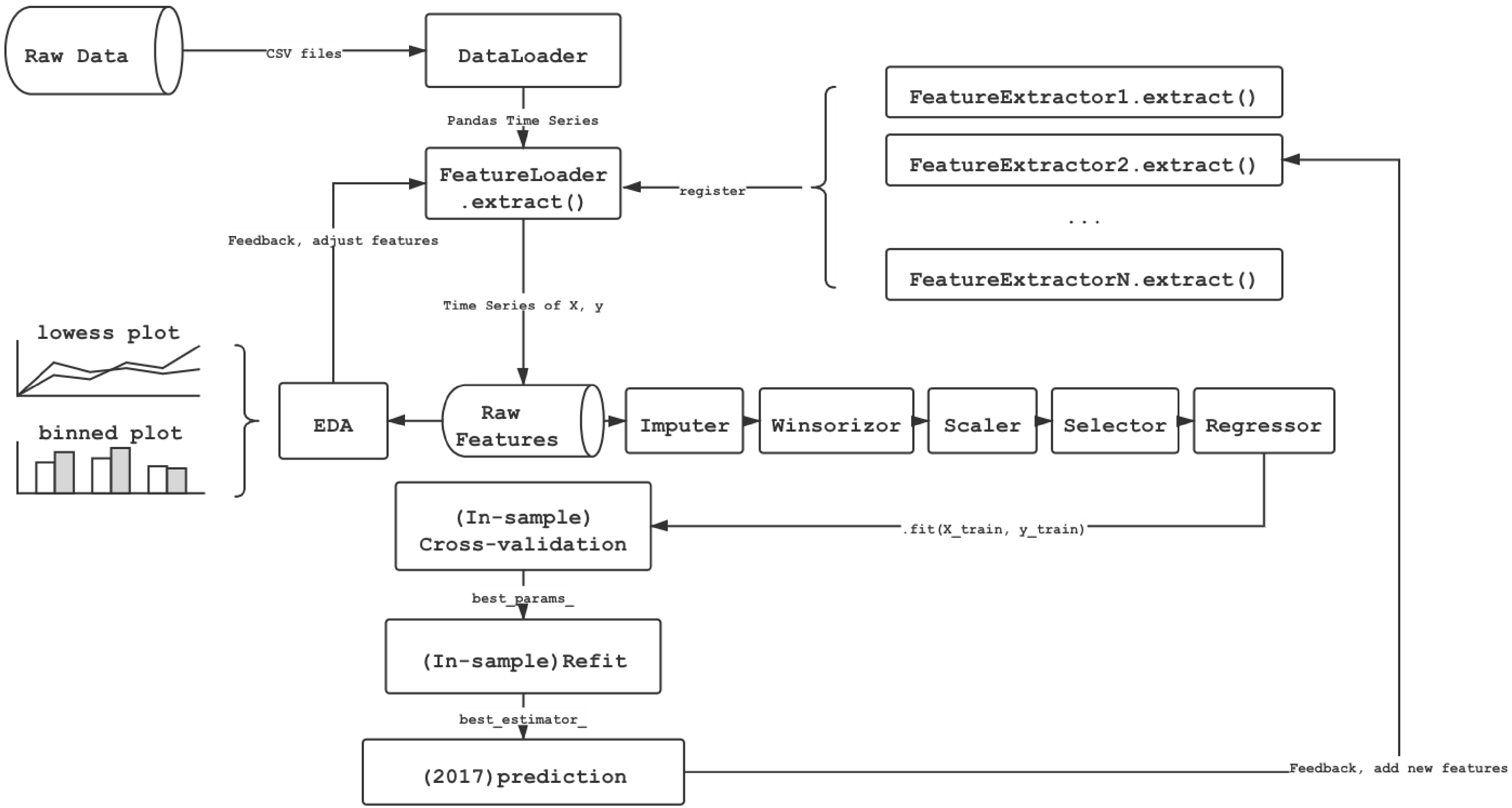
03 Feature Engineering

04 Model Tuning

05 Model Performance

01 Project Structure

The ML Pipeline



02 Data Preparation

Data preprocessing

Cleaning

- Impute the missing values by the **median** of its previous value, because
- (1) median is robust to the outliers.
- (2)will assign the lowest signal value to the model, avoid learning fake signal.

Clipping

- To avoid learning the outliers, we winsorized the data (both the features and the target) by quantile (0.01, 0.99).

Scaling

- Some model make use of regularization techniques. So, we scale the feature to zero mean, unit variance.

Train-test split

Train Set

- Train Set: 2014, 2015, 2016. We do all of our exploratory data analysis, feature engineering and hyper-parameter tuning in this set.

Cross-validation

- Split the train set into K-folds. Each fold contains all the stocks' data available on the dates in that fold. We split the train set in this way to calculate the information coefficient (IC) as one of the references of feature selection.

Test set

- 2017. We only calculate the weighted R squared on this set as a performance metric of the model.

03 Feature Engineering

Feature selection criteria?

	Description	Characteristic
IC	daily IC = corr(feature, target)	cross-sectional stock-picking capability, not necessarily useful in a single model
IR	mean(IC) / std(IC)	
RankIC	daily RankIC = corr(rank(feature), rank(target))	
RankIR	mean(RankIC) / std(RankIC)	
weighted R squared	in-sample 1/estVol weighted R squared, using LR	works well in a single model

Feature categories and full list

- Trend features

Name	Description	In-sample R squared (bps)	Out-of-sample R squared (bps)
vol-weighted res_ret	ts_sum(res_ret*vol,20)/ts_sum(vol,20)	0.38	0.25
vol-weighted raw_ret	ts_sum(raw_ret*vol,20)/ts_sum(vol,20)	0.18	0.21
20-day res_ret	ts_sum(res_ret,20)	0.15	0.32
vol raw_ret divergence	corr(vol,raw_ret,20)	0.12	0.07
5-day res_ret	ts_sum(res_ret,5)	0.04	0.24
20-day intraday res_ret	ts_sum(res_ret_day,20)	(0.02)	0.06
5-day raw_ret	ts_sum(raw_ret,5)	(0.03)	0.07
RawReturn_LogLogSquared	log(-log(raw ret**2))	0.48	0.78
Reversal1D	1 Day's Reversal	0.19	(0.22)
VolWeightedReturn	Volatility weighted residual return	0.04	(0.11)
Reversal5D	5 Day's Reversal	0.01	0.05
RawReturn_close_EMA	Exponential moving average of raw return	(0.03)	0.04
FD1008	Fractional Differentiation	(0.03)	0.02
IntradayReturn	Intra-day return	(0.04)	(0.02)
BB	Bollinger Bands	(0.04)	(0.02)
RawNoWinsorCumReturn_close	Today's Raw Return	(0.05)	(0.05)

03 Feature Engineering

Feature categories and full list

- Volatility features

Name	Description	In-sample R squared (bps)	Out-of-sample R squared (bps)
20-day raw_ret-kurt	ts_kurt(raw_ret,20)	0.11	0.25
10-day res_ret sparsity	10-day {res_ret} 75 quantile – 25 quantile	0.04	0.14
20-day res_ret sparsity	20-day {res_ret} 75 quantile – 25 quantile	(0.01)	0.14
20-day raw_ret sparsity	20-day {raw_ret} 75 quantile – 25 quantile	(0.03)	0.05
20-day ranked-estVol-kurt	ts_kurt(rank(estVol),20)	(0.03)	0.08

- Volume features

20-day rank of vol-kurt	rank(ts_kurt(vol,20))	(0.04)	0.06
Liquidity_volume_close	ln(V_t + V_t-1 + ... + V_t-n+1), where V_t is trading volume in shares	(0.01)	0.00

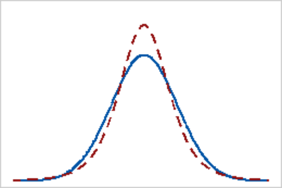
03 Feature Engineering

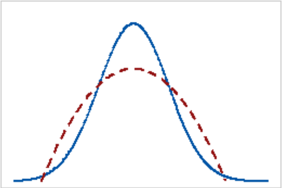
Stories behind the features

- **ts_kurt(raw_ret,20) ~ "volatility"**

  20-day kurtosis of {raw_ret} series

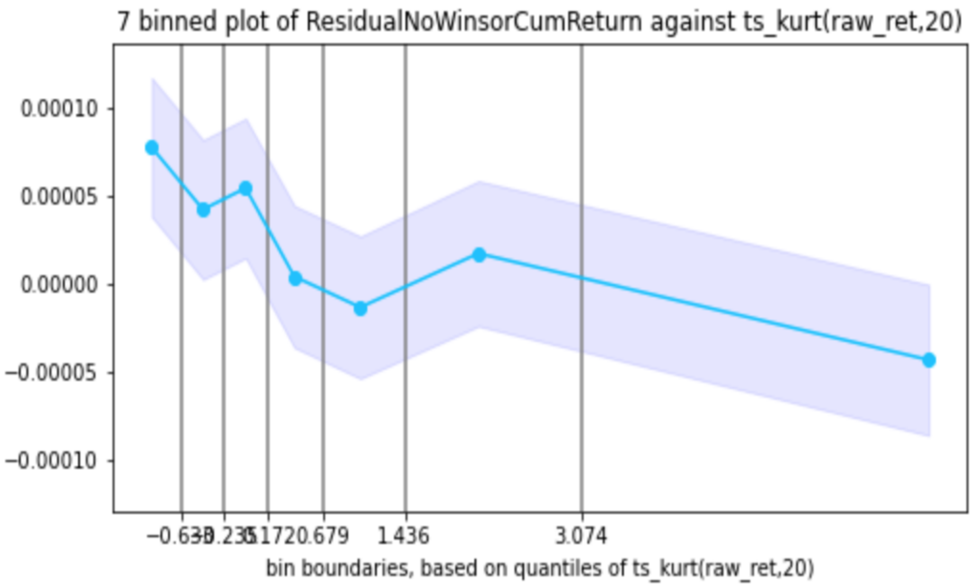
$$\mathrm{Kurt}[X] = \mathrm{E}\left[\left(\frac{X-\mu}{\sigma}\right)^4\right] = \frac{\mathrm{E}[(X-\mu)^4]}{\left(\mathrm{E}[(X-\mu)^2]\right)^2}$$

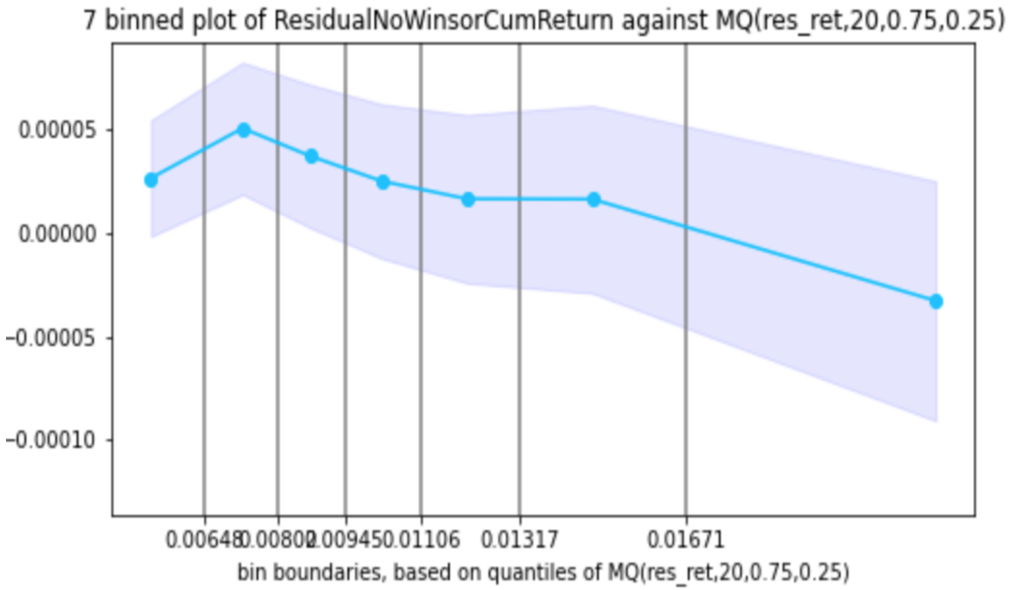




- **MQ(res_ret,20,0.75,0.25) ~ "volatility"**

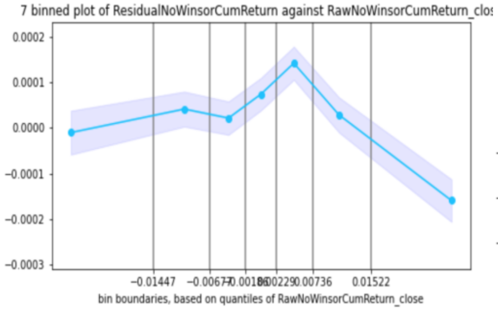
  20-day {res_ret} series,
  75-quantile – 25-quantile

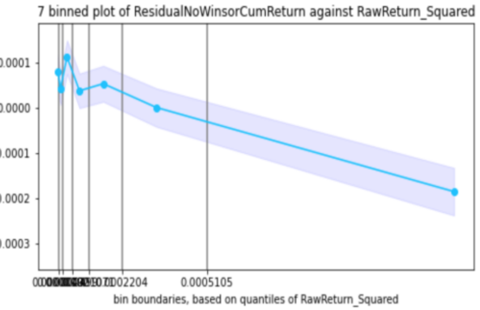


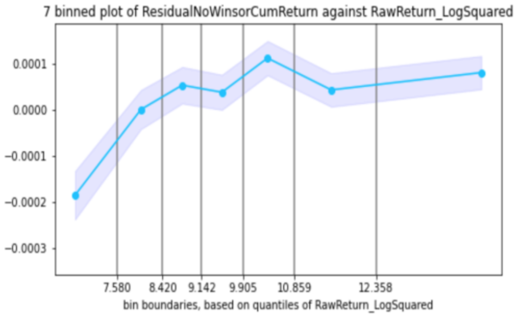


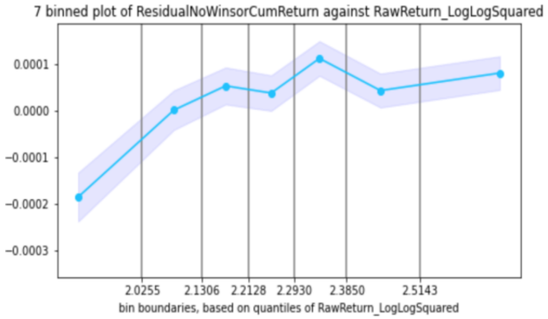
Stories behind the features

- **RawReturn_LogLogSquared**



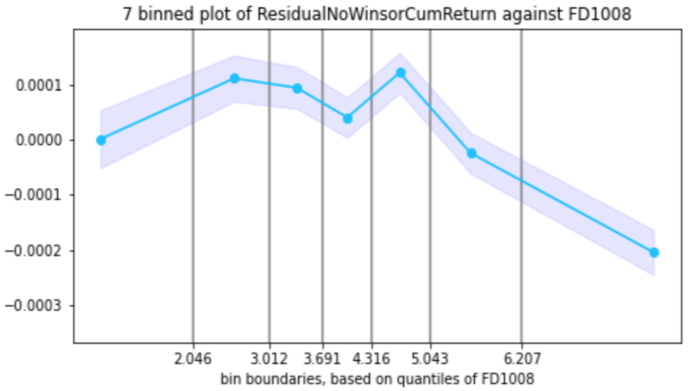






Raw Return → Squared → -Log(Squared) → Log(-Log(Squared))

- **FD1008**



Fractional Differentiation

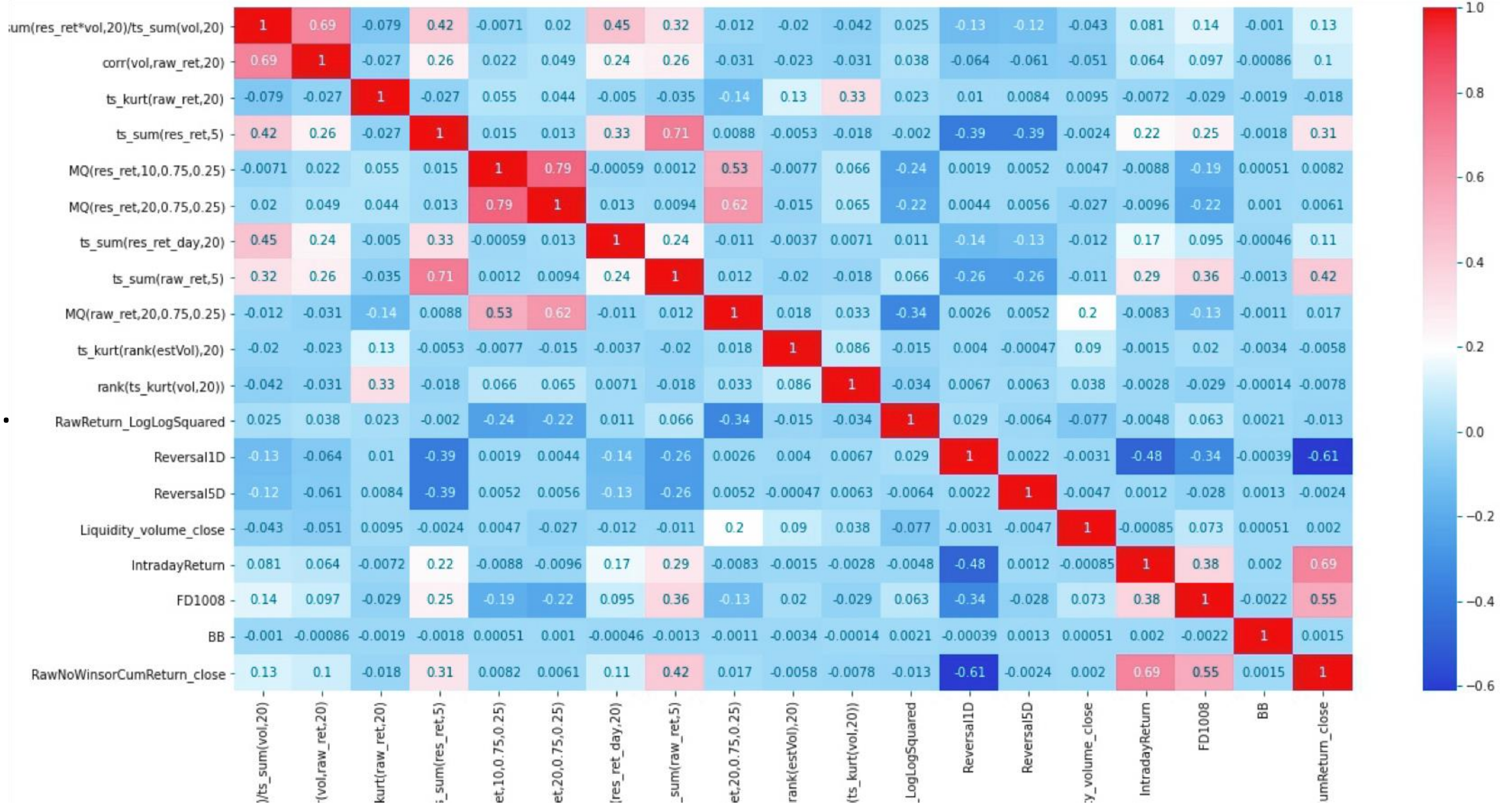
Differentiation of price with a non-integer factor d:

$$(1-B)^d = 1 - dB + \frac{d(d-1)}{2!}B^2 - \frac{d(d-1)(d-2)}{3!}B^3 + \ldots,$$ where B is a backshift operator.

Here we let d=0.8.

Feature correlation

In-sample correlation matrix of all features, excluding abs(corr) ≥ 0.8.



04 Model Tuning

The models we used

Linear Regression, Ridge Regression, XGBoost, ExtraTrees

Hyper-parameter Tuning for XGBoost

We used grid search on the training set to select the best combination of the parameters.
The parameters can be divided into two categories:

- **To fit the training set:**
  - `n_estimators: 1000`
    - the number of trees in the boosting algorithm.
  - `max_depth: [3, 5, 7, None]`
    - the maximized depth of each CART in the boosting algorithm.
- **To avoid over-fitting the training set:**
  - `earling_stopping_rounds: 5`
    - validation metric needs to improve at least once in every `early_stopping_rounds` round(s) to continue training.
  - `learning_rate: [0.01, 0.05, 0.1]`
    - boosting learning rate.
  - `reg_alpha: [0.1, 1]`
    - L1 regularization term on weights.
  - `subsample: [0.5, 1]`
    - Subsample ratio of the training instance.
  - `colsample_bytree: [0.5, 1]`
    - Subsample ratio of columns when constructing each tree.

Model Performance

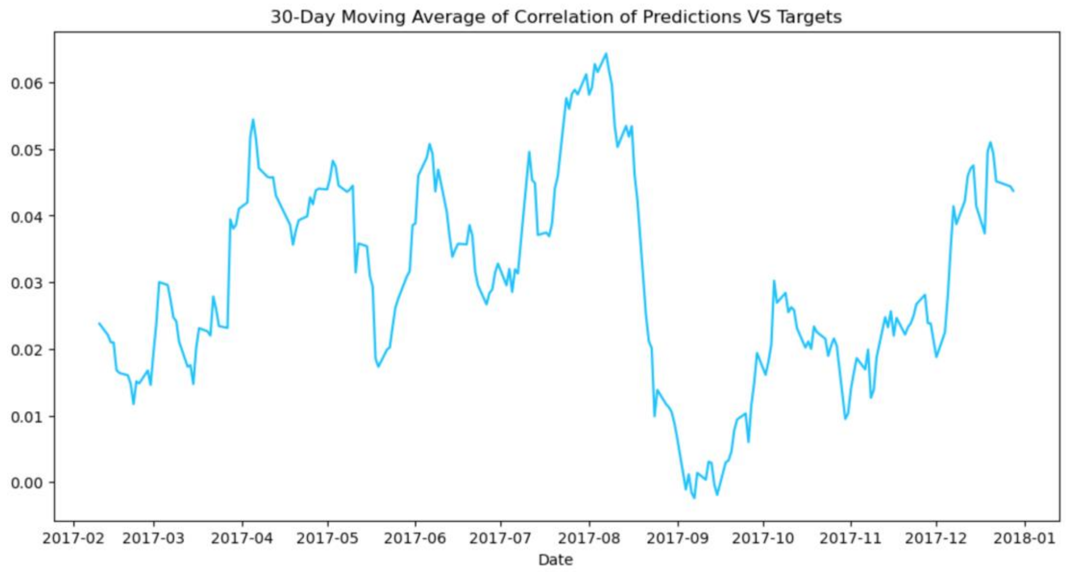
Model Name	CV R squared (bps)	In-sample (14-16) R squared (bps)	Out-of-sample (17) R squared (bps)
Linear Regression (7 features)		1.34	1.30
Linear Regression (19 features)		1.83	2.18
Ridge Regression (7 features)	0.94	1.33	1.28
Ridge Regression (19 features)	1.01	1.83	2.11
XGBoost (7 features)	0.56	8.31	1.46
XGBoost (19 features)	1.12	16.76	**5.65**
ExtraTrees (7 features)	3.67	18.41	3.98
ExtraTrees (19 features)	4.30	30.70	3.24

Model Name	CV R squared (bps)	14-15 (bps)	16 (bps)	17 (bps)
XGBoost (19 features)*	2.32	24.83	4.83	3.08
ExtraTrees (19 features)*	3.93	37.98	4.51	2.32

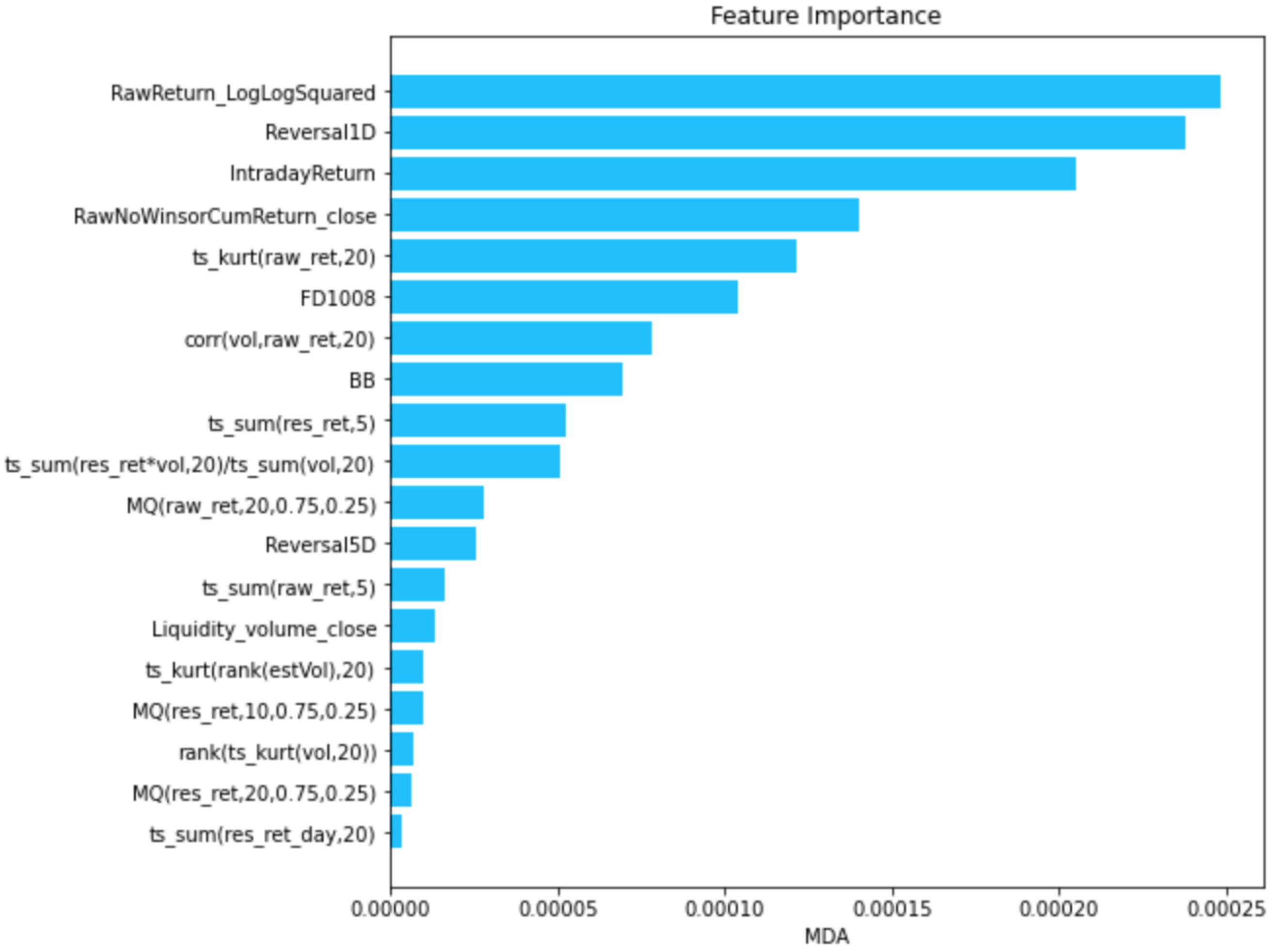
Remark: * means using 14-15 to train, 16 to validate, 17 to test

05 Model Performance

Correlation of Predictions & Targets

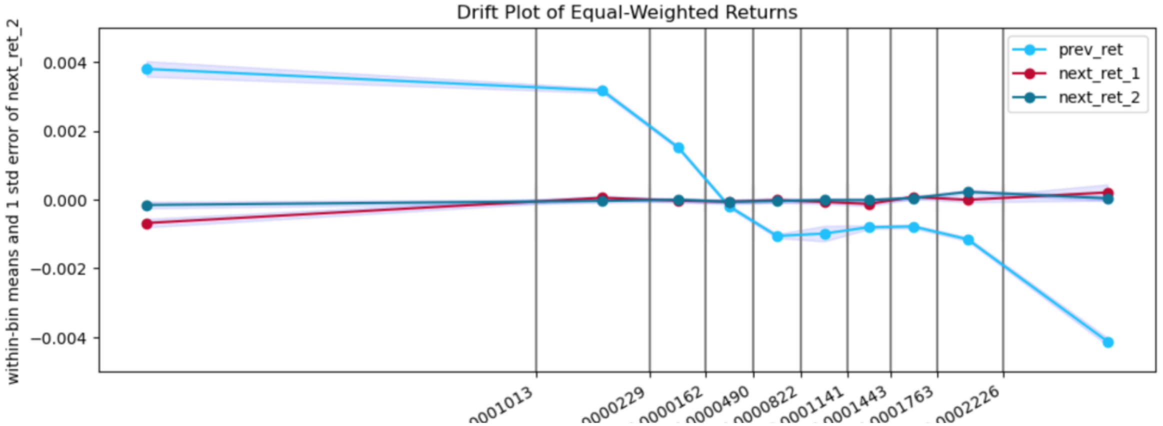


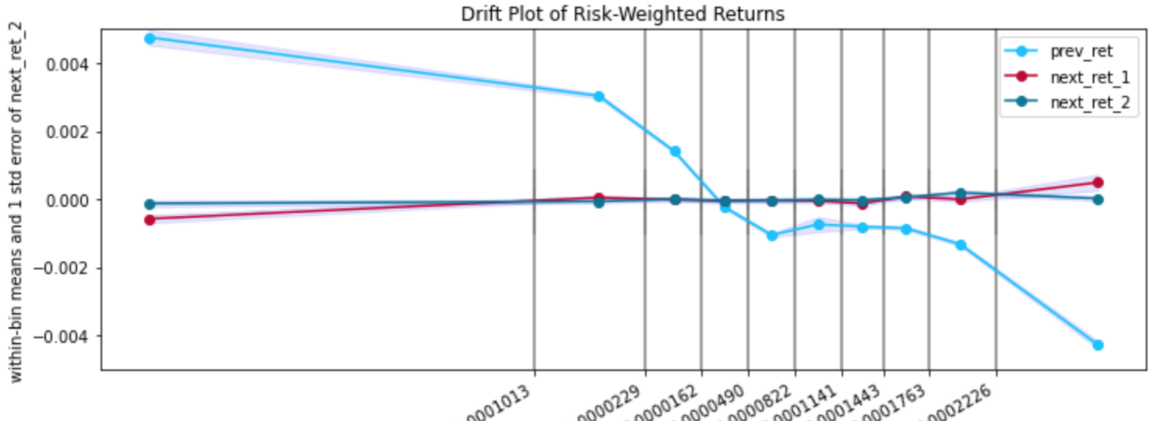
Feature Importance by MDA

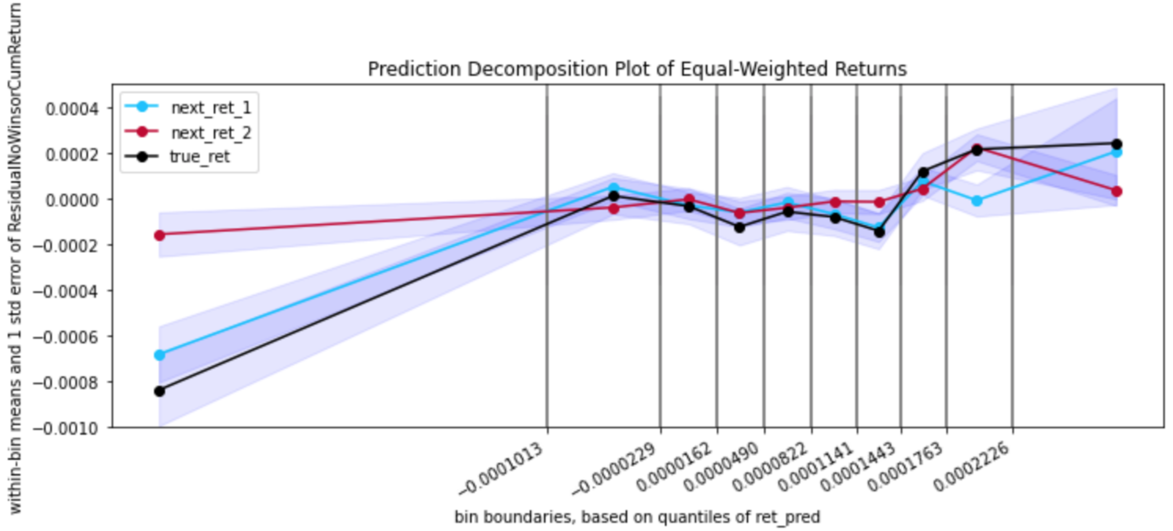


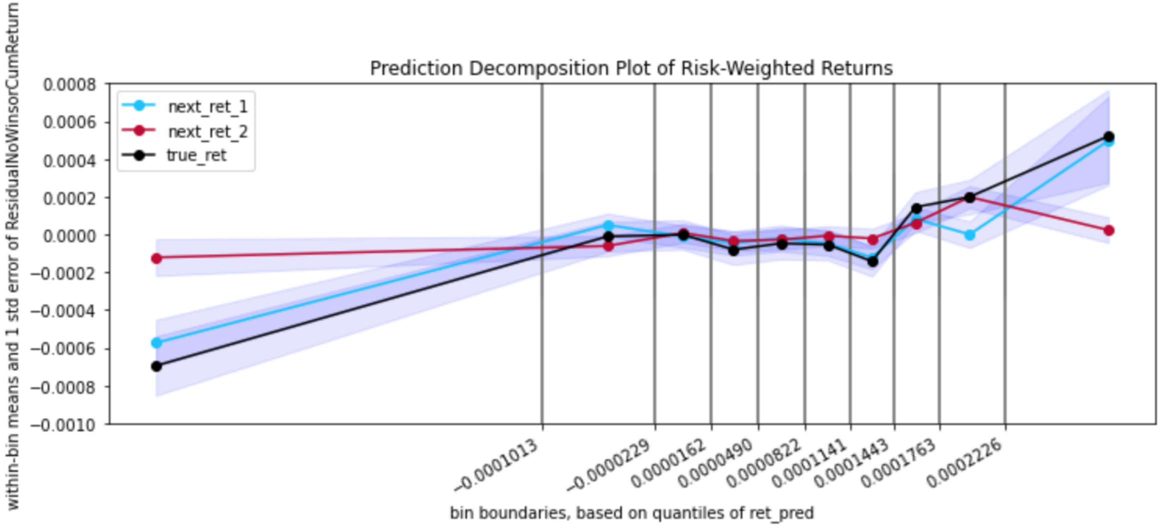
05 Model Performance

Drift Plot









Thank you!