In [1]:
```
import pandas
import numpy
```

In [2]:
```
import os
```

In [3]:
```
import strat
```

In [4]:
```
import constant
```

In [5]:
```
from autogluon import timeseries
```

In [6]:
```
import time_series
```

download data and save it

and also to a little of formatting

In [7]:
```
# sotcks_path, sp500_path = strat.saved_time_series_data_file_paths_from_url(constant.data_url)
# sotcks_path, sp500_path
```

In [8]:
```
# assert (constant.time_series_stocks_file_path, constant.time_series_sp500_file_path) == (sotcks_path, sp500_path)
```

load data from saved paths

- stocks , all stock options available ticker history from NYSE ~ New York Stock Exchange
  - you probably want to select some from it and then invest on
- sp500 or ~ Standard & Poor's ~ S&P a kinda (cu)rated selected group of 500 company merged into one indicator you can bet on

In [9]:
```
stocks_time_series_data_frame = timeseries.TimeSeriesDataFrame.from_path(constant.time_series_stocks_file_path)
```

In [10]:
```
def timestamps(time_series_data_frame):
```

```
    return time_series_data_frame.index.get_level_values('timestamp')
```

data splitting

we first split stocks data into nontest data set and test data set:

- stocks nontest data set will be used for training/validation of a price prediction model
- stocks test data set will be used to *test* the model on data it never saw (no dta leak)
  - we also keep only data that stocks test data set and the S&P data have in common (tmestamps);
    - because both will be used for testing and we will see if use that to make more money than simply betting of S&P

In [11]:
```
# Splitting the data into train and test sets based on date
nontest_time_series_data_frame, test_time_series_data_frame = stocks_time_series_data_frame.sort_index().split_by_t
    cutoff_time=constant.test_start_day_time_64,
)
nontest_time_series_data_frame = nontest_time_series_data_frame.dropna()
test_time_series_data_frame = test_time_series_data_frame.dropna()
```

In [12]:
```
sp500_time_series_data_frame = timeseries.TimeSeriesDataFrame.from_path(constant.time_series_sp500_file_path)
_, sp500_test_time_series_data_frame = sp500_time_series_data_frame.sort_index().split_by_time(
    cutoff_time=constant.test_start_day_time_64,
)
sp500_test_time_series_data_frame = sp500_test_time_series_data_frame[[
    timestamp in timestamps(test_time_series_data_frame)
    for timestamp in timestamps(sp500_test_time_series_data_frame)
]]
```

```
/home/wam/kood/sp500-strategies/envs/strategies_autogluon/lib/python3.11/site-packages/autogluon/timeseries/dataset/
ts_dataframe.py:222: UserWarning: Could not infer format, so each element will be parsed individually, falling back
to `dateutil`. To ensure parsing is consistent and as-expected, please specify a format.
  df[TIMESTAMP] = pd.to_datetime(df[TIMESTAMP])
```

In [13]:
```
to_iso = lambda timestamp: timestamp.isoformat(timespec='auto')[:10]
print()
print(f"earliest and latest date from stocks nontest data set:\n\t{to_iso(timestamps(nontest_time_series_data_frame
print()
print(f"earliest and latest date from stocks test data set:\n\t{to_iso(timestamps(test_time_series_data_frame)[0])}
```

```
print(f"earliest and latest date from S&P (test) data set:\n\t{to_iso(timestamps(sp500_test_time_series_data_frame)
```

```
earliest and latest date from stocks nontest data set:
        2013-02-08      2016-12-30

earliest and latest date from stocks test data set:
        2017-01-03      2018-02-07
earliest and latest date from S&P (test) data set:
        2017-01-03      2018-02-07
```

cross training/validation

the model(s) autogluon timeseries provide are sophisticated to to it automatically but because it is requested we gonna do it mannually
we take ten different chunks of the nontest data set and train a model on it
and then check how good it predicts the next **one** opened day prices we keep only the best one and voila

note: we also feed the volume, open price, etc. to the model as additional hinting data because it is available and why not

In [14]:
```
def best_cross_validated_predictor(
    nontest_time_series_data_frame,
    number_of_folds=3,
    train_time_limit=10,
):
    score_metric= 'MAPE'

    train_windows = []
    validate_windows = []

    train_scores = []
    validate_scores = []

    best_predictor = None
    best_validate_score_metric = -999999999999999999999

    nontest_time_series_data_frame.index.get_level_values('timestamp')

    splitter = timeseries.splitter.ExpandingWindowSplitter(prediction_length=1, num_val_windows=number_of_folds)
```

```python
for window_split_index, (train_split, validate_split) in enumerate(splitter.split(nontest_time_series_data_fram
    print(f"{window_split_index = }")
    # print(f"{train_split.index.get_level_values('timestamp')[0] = }")
    # print(f"{train_split.index.get_level_values('timestamp')[-1] = }")
    # print(f"{validate_split.index.get_level_values('timestamp')[0] = }")
    # print(f"{validate_split.index.get_level_values('timestamp')[-1] = }")
    train_windows.append(train_split.index.get_level_values('timestamp'))
    validate_windows.append(validate_split.index.get_level_values('timestamp'))
    predictor = timeseries.TimeSeriesPredictor(
        target='target',
        prediction_length=1,
        freq='B',
        # freq='D',
        eval_metric=score_metric,
        quantile_levels=[],
        path=os.path.join(constant.model_folder_path, "autogluon_model"),
        verbosity=2,
    )
    predictor.fit(
        train_data=train_split,
        tuning_data=validate_split,
        presets='fast_training',
        # presets='medium_quality',
        # 'WeightedEnsemble'
        # excluded_model_types=['SeasonalNaive','RecursiveTabular'],
        # excluded_model_types=['WeightedEnsemble'],
        # number_val_windows=5,
        # time_limit=60*15, # seconds !!!!!!!!!!
        time_limit=train_time_limit, # seconds !!!!!!!!!!
    )
    try:
        predictor.predict(
            data=nontest_time_series_data_frame,
            # known_covariates=,
            use_cache=False,
        )
    except Exception as error:
        print(f"model is broken we go to next fold {error}")
        continue
```

```
        # print(f"\nfitting summary:\n{predictor.fit_summary()['leaderboard']}")
        train_score = predictor.evaluate(train_split)
        validate_score = predictor.evaluate(validate_split)
        train_scores.append(train_score)
        validate_scores.append(validate_score)
        print(f"window split n*{window_split_index}; validation score: {validate_score[score_metric]}")
        print(f"best score: {best_validate_score_metric = }")
        print(f"score: {validate_score[score_metric] = }")
        if validate_score[score_metric] > best_validate_score_metric:
            best_validate_score_metric = validate_score[score_metric]
            best_predictor = predictor
            print(f"\tnew best model found ! {best_predictor =}")

    print(f"done {best_predictor =}")
    if best_predictor is None:
        1/0
    return (
        best_predictor,
        {
            "train_scores":train_scores,
            "validate_scores":validate_scores,
            "train_windows":train_windows,
            "validate_windows":validate_windows,
        },
    )
```

In [15]:
```
nontest_time_series_data_frame = nontest_time_series_data_frame.dropna()
```

In [16]:
```
best_predictor, info = best_cross_validated_predictor(
    nontest_time_series_data_frame=nontest_time_series_data_frame,
    number_of_folds=10,
    train_time_limit=10*60,
)
```

```
Warning: path already exists! This predictor may overwrite an existing predictor! path="../model/autogluon_model"
Beginning AutoGluon training... Time limit = 600s
AutoGluon will save models to '../model/autogluon_model'
window_split_index = 0
```

```
================== System Info ==================
AutoGluon Version:  1.1.1
Python Version:     3.11.7
Operating System:   Linux
Platform Machine:   x86_64
Platform Version:   #1 SMP PREEMPT_DYNAMIC Mon, 30 Sep 2024 23:49:50 +0000
CPU Count:          8
GPU Count:          0
Memory Avail:       1.55 GB / 7.60 GB (20.4%)
Disk Space Avail:   19.59 GB / 109.46 GB (17.9%)
=================================================
Setting presets to: fast_training

Fitting with arguments:
{'enable_ensemble': True,
 'eval_metric': MAPE,
 'freq': 'B',
 'hyperparameters': 'very_light',
 'known_covariates_names': [],
 'num_val_windows': 1,
 'prediction_length': 1,
 'quantile_levels': [],
 'random_seed': 123,
 'refit_every_n_windows': 1,
 'refit_full': False,
 'skip_model_selection': False,
 'target': 'target',
 'time_limit': 600,
 'verbosity': 2}

train_data with frequency 'None' has been resampled to frequency 'B'.
Provided train_data has 491053 rows (NaN fraction=3.3%), 499 time series. Median time series length is 1005 (min=10,
max=1005).
tuning_data with frequency 'None' has been resampled to frequency 'B'.
Provided tuning_data has 491053 rows (NaN fraction=3.3%), 499 time series. Median time series length is 1005 (min=1
0, max=1005).
        Setting num_val_windows = 0 (disabling backtesting on train_data) because tuning_data is provided.

Provided data contains following columns:
        target: 'target'
```

```
        past_covariates:
                categorical:        []
                continuous (float): ['past_covariate_open', 'past_covariate_high', 'past_covariate_low', 'past_covar
iate_volume']

To learn how to fix incorrectly inferred types, please see documentation for TimeSeriesPredictor.fit

AutoGluon will gauge predictive performance using evaluation metric: 'MAPE'
        This metric's sign has been flipped to adhere to being higher_is_better. The metric score can be multiplied
by -1 to get the metric value.
===================================================

Starting training. Start time is 2024-10-22 11:56:45
Models that will be trained: ['Naive', 'SeasonalNaive', 'RecursiveTabular', 'DirectTabular', 'ETS', 'Theta']
Training timeseries model Naive. Training for up to 84.3s of the 590.3s of remaining time.
        -0.0110       = Validation score (-MAPE)
        0.03    s     = Training runtime
        5.19    s     = Validation (prediction) runtime
Training timeseries model SeasonalNaive. Training for up to 97.5s of the 584.9s of remaining time.
        -0.0237       = Validation score (-MAPE)
        0.02    s     = Training runtime
        0.29    s     = Validation (prediction) runtime
Training timeseries model RecursiveTabular. Training for up to 116.9s of the 584.5s of remaining time.
        -0.0241       = Validation score (-MAPE)
        2.08    s     = Training runtime
        0.68    s     = Validation (prediction) runtime
Training timeseries model DirectTabular. Training for up to 145.4s of the 581.6s of remaining time.
        -0.0114       = Validation score (-MAPE)
        1.34    s     = Training runtime
        0.35    s     = Validation (prediction) runtime
Training timeseries model ETS. Training for up to 193.3s of the 579.8s of remaining time.
        -0.0111       = Validation score (-MAPE)
        0.07    s     = Training runtime
        102.35  s     = Validation (prediction) runtime
Training timeseries model Theta. Training for up to 238.6s of the 477.3s of remaining time.
        -0.0111       = Validation score (-MAPE)
        0.07    s     = Training runtime
        11.87   s     = Validation (prediction) runtime
Fitting simple weighted ensemble.
        Ensemble weights: {'DirectTabular': 0.4, 'Naive': 0.6}
        -0.0106       = Validation score (-MAPE)
```

```
        0.30    s     = Training runtime
        5.54    s     = Validation (prediction) runtime
Training complete. Models trained: ['Naive', 'SeasonalNaive', 'RecursiveTabular', 'DirectTabular', 'ETS', 'Theta',
'WeightedEnsemble']
Total runtime: 126.50 s
Best model: WeightedEnsemble
Best model score: -0.0106
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
Beginning AutoGluon training... Time limit = 600s
AutoGluon will save models to '../model/autogluon_model'
=================== System Info ===================
AutoGluon Version:  1.1.1
Python Version:     3.11.7
Operating System:   Linux
Platform Machine:   x86_64
Platform Version:   #1 SMP PREEMPT_DYNAMIC Mon, 30 Sep 2024 23:49:50 +0000
CPU Count:          8
GPU Count:          0
Memory Avail:       1.89 GB / 7.60 GB (24.8%)
Disk Space Avail:   19.59 GB / 109.46 GB (17.9%)
===================================================
Setting presets to: fast_training

Fitting with arguments:
{'enable_ensemble': True,
 'eval_metric': MAPE,
 'freq': 'B',
 'hyperparameters': 'very_light',
 'known_covariates_names': [],
 'num_val_windows': 1,
 'prediction_length': 1,
 'quantile_levels': [],
 'random_seed': 123,
 'refit_every_n_windows': 1,
 'refit_full': False,
 'skip_model_selection': False,
```

```
 'target': 'target',
 'time_limit': 600,
 'verbosity': 2}

train_data with frequency 'None' has been resampled to frequency 'B'.
window split n*0; validation score: -0.010559709114952026
best score: best_validate_score_metric = -9999999999999999999
score: validate_score[score_metric] = -0.010559709114952026
        new best model found ! best_predictor =<autogluon.timeseries.predictor.TimeSeriesPredictor object at 0x74a74
df33690>
window_split_index = 1
```

```
Provided train_data has 491552 rows (NaN fraction=3.3%), 499 time series. Median time series length is 1006 (min=11,
max=1006).
tuning_data with frequency 'None' has been resampled to frequency 'B'.
Provided tuning_data has 491552 rows (NaN fraction=3.3%), 499 time series. Median time series length is 1006 (min=1
1, max=1006).
        Setting num_val_windows = 0 (disabling backtesting on train_data) because tuning_data is provided.

Provided data contains following columns:
        target: 'target'
        past_covariates:
                categorical:        []
                continuous (float): ['past_covariate_open', 'past_covariate_high', 'past_covariate_low', 'past_covar
iate_volume']

To learn how to fix incorrectly inferred types, please see documentation for TimeSeriesPredictor.fit

AutoGluon will gauge predictive performance using evaluation metric: 'MAPE'
        This metric's sign has been flipped to adhere to being higher_is_better. The metric score can be multiplied
by -1 to get the metric value.
===================================================

Starting training. Start time is 2024-10-22 11:59:11
Models that will be trained: ['Naive', 'SeasonalNaive', 'RecursiveTabular', 'DirectTabular', 'ETS', 'Theta']
Training timeseries model Naive. Training for up to 85.1s of the 595.6s of remaining time.
        -0.0073       = Validation score (-MAPE)
        0.02    s     = Training runtime
        1.44    s     = Validation (prediction) runtime
Training timeseries model SeasonalNaive. Training for up to 99.0s of the 594.0s of remaining time.
        -0.0196       = Validation score (-MAPE)
        0.02    s     = Training runtime
        0.29    s     = Validation (prediction) runtime
Training timeseries model RecursiveTabular. Training for up to 118.7s of the 593.6s of remaining time.
        -0.0167       = Validation score (-MAPE)
        1.20    s     = Training runtime
        0.22    s     = Validation (prediction) runtime
Training timeseries model DirectTabular. Training for up to 148.0s of the 592.1s of remaining time.
        -0.0100       = Validation score (-MAPE)
        1.25    s     = Training runtime
        0.33    s     = Validation (prediction) runtime
Training timeseries model ETS. Training for up to 196.8s of the 590.4s of remaining time.
```

```
        -0.0073       = Validation score (-MAPE)
        0.07    s     = Training runtime
        101.08  s     = Validation (prediction) runtime
Training timeseries model Theta. Training for up to 244.6s of the 489.1s of remaining time.
        -0.0072       = Validation score (-MAPE)
        0.07    s     = Training runtime
        12.70   s     = Validation (prediction) runtime
Fitting simple weighted ensemble.
        Ensemble weights: {'Naive': 0.1, 'RecursiveTabular': 0.03, 'Theta': 0.87}
        -0.0072       = Validation score (-MAPE)
        0.30    s     = Training runtime
        14.36   s     = Validation (prediction) runtime
Training complete. Models trained: ['Naive', 'SeasonalNaive', 'RecursiveTabular', 'DirectTabular', 'ETS', 'Theta',
'WeightedEnsemble']
Total runtime: 119.80 s
Best model: WeightedEnsemble
Best model score: -0.0072
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
Beginning AutoGluon training... Time limit = 600s
AutoGluon will save models to '../model/autogluon_model'
=================== System Info ===================
AutoGluon Version:  1.1.1
Python Version:     3.11.7
Operating System:   Linux
Platform Machine:   x86_64
Platform Version:   #1 SMP PREEMPT_DYNAMIC Mon, 30 Sep 2024 23:49:50 +0000
CPU Count:          8
GPU Count:          0
Memory Avail:       1.43 GB / 7.60 GB (18.8%)
Disk Space Avail:   19.59 GB / 109.46 GB (17.9%)
===================================================
Setting presets to: fast_training

Fitting with arguments:
{'enable_ensemble': True,
 'eval_metric': MAPE,
```

```
 'freq': 'B',
 'hyperparameters': 'very_light',
 'known_covariates_names': [],
 'num_val_windows': 1,
 'prediction_length': 1,
 'quantile_levels': [],
 'random_seed': 123,
 'refit_every_n_windows': 1,
 'refit_full': False,
 'skip_model_selection': False,
 'target': 'target',
 'time_limit': 600,
 'verbosity': 2}

train_data with frequency 'None' has been resampled to frequency 'B'.
window split n*1; validation score: -0.007176118262151318
best score: best_validate_score_metric = -0.010559709114952026
score: validate_score[score_metric] = -0.007176118262151318
        new best model found ! best_predictor =<autogluon.timeseries.predictor.TimeSeriesPredictor object at 0x74a74
df2b450>
window_split_index = 2
```

```
Provided train_data has 492051 rows (NaN fraction=3.3%), 499 time series. Median time series length is 1007 (min=12,
max=1007).
tuning_data with frequency 'None' has been resampled to frequency 'B'.
Provided tuning_data has 492051 rows (NaN fraction=3.3%), 499 time series. Median time series length is 1007 (min=1
2, max=1007).
        Setting num_val_windows = 0 (disabling backtesting on train_data) because tuning_data is provided.

Provided data contains following columns:
        target: 'target'
        past_covariates:
                categorical:        []
                continuous (float): ['past_covariate_open', 'past_covariate_high', 'past_covariate_low', 'past_covar
iate_volume']

To learn how to fix incorrectly inferred types, please see documentation for TimeSeriesPredictor.fit

AutoGluon will gauge predictive performance using evaluation metric: 'MAPE'
        This metric's sign has been flipped to adhere to being higher_is_better. The metric score can be multiplied
by -1 to get the metric value.
===================================================

Starting training. Start time is 2024-10-22 12:02:28
Models that will be trained: ['Naive', 'SeasonalNaive', 'RecursiveTabular', 'DirectTabular', 'ETS', 'Theta']
Training timeseries model Naive. Training for up to 85.0s of the 595.2s of remaining time.
        -0.0079       = Validation score (-MAPE)
        0.02    s     = Training runtime
        1.42    s     = Validation (prediction) runtime
Training timeseries model SeasonalNaive. Training for up to 98.9s of the 593.7s of remaining time.
        -0.0193       = Validation score (-MAPE)
        0.02    s     = Training runtime
        0.34    s     = Validation (prediction) runtime
Training timeseries model RecursiveTabular. Training for up to 118.6s of the 593.2s of remaining time.
        -0.0184       = Validation score (-MAPE)
        1.07    s     = Training runtime
        0.23    s     = Validation (prediction) runtime
Training timeseries model DirectTabular. Training for up to 147.9s of the 591.8s of remaining time.
        -0.0101       = Validation score (-MAPE)
        1.28    s     = Training runtime
        0.32    s     = Validation (prediction) runtime
Training timeseries model ETS. Training for up to 196.7s of the 590.1s of remaining time.
```

```
        -0.0080       = Validation score (-MAPE)
        0.07    s     = Training runtime
        100.18  s     = Validation (prediction) runtime
Training timeseries model Theta. Training for up to 244.9s of the 489.7s of remaining time.
        -0.0078       = Validation score (-MAPE)
        0.07    s     = Training runtime
        12.56   s     = Validation (prediction) runtime
Fitting simple weighted ensemble.
        Ensemble weights: {'Naive': 0.5, 'RecursiveTabular': 0.06, 'Theta': 0.44}
        -0.0077       = Validation score (-MAPE)
        0.31    s     = Training runtime
        14.20   s     = Validation (prediction) runtime
Training complete. Models trained: ['Naive', 'SeasonalNaive', 'RecursiveTabular', 'DirectTabular', 'ETS', 'Theta',
'WeightedEnsemble']
Total runtime: 118.71 s
Best model: WeightedEnsemble
Best model score: -0.0077
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
Beginning AutoGluon training... Time limit = 600s
AutoGluon will save models to '../model/autogluon_model'
=================== System Info ===================
AutoGluon Version:  1.1.1
Python Version:     3.11.7
Operating System:   Linux
Platform Machine:   x86_64
Platform Version:   #1 SMP PREEMPT_DYNAMIC Mon, 30 Sep 2024 23:49:50 +0000
CPU Count:          8
GPU Count:          0
Memory Avail:       1.46 GB / 7.60 GB (19.2%)
Disk Space Avail:   19.59 GB / 109.46 GB (17.9%)
====================================================
Setting presets to: fast_training

Fitting with arguments:
{'enable_ensemble': True,
 'eval_metric': MAPE,
```

```
 'freq': 'B',
 'hyperparameters': 'very_light',
 'known_covariates_names': [],
 'num_val_windows': 1,
 'prediction_length': 1,
 'quantile_levels': [],
 'random_seed': 123,
 'refit_every_n_windows': 1,
 'refit_full': False,
 'skip_model_selection': False,
 'target': 'target',
 'time_limit': 600,
 'verbosity': 2}

train_data with frequency 'None' has been resampled to frequency 'B'.
window split n*2; validation score: -0.007717207582556126
best score: best_validate_score_metric = -0.007176118262151318
score: validate_score[score_metric] = -0.007717207582556126
window_split_index = 3
```

```
Provided train_data has 492550 rows (NaN fraction=3.3%), 499 time series. Median time series length is 1008 (min=13,
max=1008).
tuning_data with frequency 'None' has been resampled to frequency 'B'.
Provided tuning_data has 492550 rows (NaN fraction=3.3%), 499 time series. Median time series length is 1008 (min=1
3, max=1008).
        Setting num_val_windows = 0 (disabling backtesting on train_data) because tuning_data is provided.

Provided data contains following columns:
        target: 'target'
        past_covariates:
                categorical:        []
                continuous (float): ['past_covariate_open', 'past_covariate_high', 'past_covariate_low', 'past_covar
iate_volume']

To learn how to fix incorrectly inferred types, please see documentation for TimeSeriesPredictor.fit

AutoGluon will gauge predictive performance using evaluation metric: 'MAPE'
        This metric's sign has been flipped to adhere to being higher_is_better. The metric score can be multiplied
by -1 to get the metric value.
===================================================

Starting training. Start time is 2024-10-22 12:05:46
Models that will be trained: ['Naive', 'SeasonalNaive', 'RecursiveTabular', 'DirectTabular', 'ETS', 'Theta']
Training timeseries model Naive. Training for up to 85.0s of the 595.3s of remaining time.
        -0.0067       = Validation score (-MAPE)
        0.02    s     = Training runtime
        1.46    s     = Validation (prediction) runtime
Training timeseries model SeasonalNaive. Training for up to 98.9s of the 593.7s of remaining time.
        -0.0169       = Validation score (-MAPE)
        0.02    s     = Training runtime
        0.28    s     = Validation (prediction) runtime
Training timeseries model RecursiveTabular. Training for up to 118.7s of the 593.3s of remaining time.
        -0.0155       = Validation score (-MAPE)
        0.95    s     = Training runtime
        0.23    s     = Validation (prediction) runtime
Training timeseries model DirectTabular. Training for up to 148.0s of the 592.0s of remaining time.
        -0.0090       = Validation score (-MAPE)
        2.80    s     = Training runtime
        0.32    s     = Validation (prediction) runtime
Training timeseries model ETS. Training for up to 196.3s of the 588.8s of remaining time.
```

```
        -0.0068       = Validation score (-MAPE)
        0.07    s     = Training runtime
        99.46   s     = Validation (prediction) runtime
Training timeseries model Theta. Training for up to 244.6s of the 489.1s of remaining time.
        -0.0071       = Validation score (-MAPE)
        0.07    s     = Training runtime
        12.19   s     = Validation (prediction) runtime
Fitting simple weighted ensemble.
        Ensemble weights: {'Naive': 0.9, 'SeasonalNaive': 0.1}
        -0.0065       = Validation score (-MAPE)
        0.30    s     = Training runtime
        1.74    s     = Validation (prediction) runtime
Training complete. Models trained: ['Naive', 'SeasonalNaive', 'RecursiveTabular', 'DirectTabular', 'ETS', 'Theta',
'WeightedEnsemble']
Total runtime: 118.97 s
Best model: WeightedEnsemble
Best model score: -0.0065
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
Beginning AutoGluon training... Time limit = 600s
AutoGluon will save models to '../model/autogluon_model'
=================== System Info ===================
AutoGluon Version:  1.1.1
Python Version:     3.11.7
Operating System:   Linux
Platform Machine:   x86_64
Platform Version:   #1 SMP PREEMPT_DYNAMIC Mon, 30 Sep 2024 23:49:50 +0000
CPU Count:          8
GPU Count:          0
Memory Avail:       2.20 GB / 7.60 GB (28.9%)
Disk Space Avail:   19.59 GB / 109.46 GB (17.9%)
===================================================
Setting presets to: fast_training

Fitting with arguments:
{'enable_ensemble': True,
 'eval_metric': MAPE,
```

```
 'freq': 'B',
 'hyperparameters': 'very_light',
 'known_covariates_names': [],
 'num_val_windows': 1,
 'prediction_length': 1,
 'quantile_levels': [],
 'random_seed': 123,
 'refit_every_n_windows': 1,
 'refit_full': False,
 'skip_model_selection': False,
 'target': 'target',
 'time_limit': 600,
 'verbosity': 2}

train_data with frequency 'None' has been resampled to frequency 'B'.
window split n*3; validation score: -0.006502177090277285
best score: best_validate_score_metric = -0.007176118262151318
score: validate_score[score_metric] = -0.006502177090277285
        new best model found ! best_predictor =<autogluon.timeseries.predictor.TimeSeriesPredictor object at 0x74a74
df0c0d0>
window_split_index = 4
```

```
Provided train_data has 493049 rows (NaN fraction=3.3%), 499 time series. Median time series length is 1009 (min=14,
max=1009).
tuning_data with frequency 'None' has been resampled to frequency 'B'.
Provided tuning_data has 493049 rows (NaN fraction=3.3%), 499 time series. Median time series length is 1009 (min=1
4, max=1009).
        Setting num_val_windows = 0 (disabling backtesting on train_data) because tuning_data is provided.

Provided data contains following columns:
        target: 'target'
        past_covariates:
                categorical:        []
                continuous (float): ['past_covariate_open', 'past_covariate_high', 'past_covariate_low', 'past_covar
iate_volume']

To learn how to fix incorrectly inferred types, please see documentation for TimeSeriesPredictor.fit

AutoGluon will gauge predictive performance using evaluation metric: 'MAPE'
        This metric's sign has been flipped to adhere to being higher_is_better. The metric score can be multiplied
by -1 to get the metric value.
===================================================

Starting training. Start time is 2024-10-22 12:08:03
Models that will be trained: ['Naive', 'SeasonalNaive', 'RecursiveTabular', 'DirectTabular', 'ETS', 'Theta']
Training timeseries model Naive. Training for up to 85.1s of the 595.7s of remaining time.
        -0.0085       = Validation score (-MAPE)
        0.02    s     = Training runtime
        1.46    s     = Validation (prediction) runtime
Training timeseries model SeasonalNaive. Training for up to 99.0s of the 594.1s of remaining time.
        -0.0175       = Validation score (-MAPE)
        0.02    s     = Training runtime
        0.30    s     = Validation (prediction) runtime
Training timeseries model RecursiveTabular. Training for up to 118.7s of the 593.6s of remaining time.
        -0.0178       = Validation score (-MAPE)
        1.07    s     = Training runtime
        0.22    s     = Validation (prediction) runtime
Training timeseries model DirectTabular. Training for up to 148.1s of the 592.2s of remaining time.
        -0.0110       = Validation score (-MAPE)
        1.32    s     = Training runtime
        0.34    s     = Validation (prediction) runtime
Training timeseries model ETS. Training for up to 196.8s of the 590.4s of remaining time.
```

```
        -0.0086       = Validation score (-MAPE)
        0.07    s     = Training runtime
        101.56  s     = Validation (prediction) runtime
Training timeseries model Theta. Training for up to 244.4s of the 488.7s of remaining time.
        -0.0091       = Validation score (-MAPE)
        0.07    s     = Training runtime
        12.05   s     = Validation (prediction) runtime
Fitting simple weighted ensemble.
        Ensemble weights: {'Naive': 1.0}
        -0.0085       = Validation score (-MAPE)
        0.30    s     = Training runtime
        1.46    s     = Validation (prediction) runtime
Training complete. Models trained: ['Naive', 'SeasonalNaive', 'RecursiveTabular', 'DirectTabular', 'ETS', 'Theta',
'WeightedEnsemble']
Total runtime: 119.64 s
Best model: Naive
Best model score: -0.0085
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: Naive
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: Naive
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: Naive
Beginning AutoGluon training... Time limit = 600s
AutoGluon will save models to '../model/autogluon_model'
=================== System Info ===================
AutoGluon Version:  1.1.1
Python Version:     3.11.7
Operating System:   Linux
Platform Machine:   x86_64
Platform Version:   #1 SMP PREEMPT_DYNAMIC Mon, 30 Sep 2024 23:49:50 +0000
CPU Count:          8
GPU Count:          0
Memory Avail:       2.26 GB / 7.60 GB (29.8%)
Disk Space Avail:   19.59 GB / 109.46 GB (17.9%)
===================================================
Setting presets to: fast_training

Fitting with arguments:
{'enable_ensemble': True,
 'eval_metric': MAPE,
```

```
 'freq': 'B',
 'hyperparameters': 'very_light',
 'known_covariates_names': [],
 'num_val_windows': 1,
 'prediction_length': 1,
 'quantile_levels': [],
 'random_seed': 123,
 'refit_every_n_windows': 1,
 'refit_full': False,
 'skip_model_selection': False,
 'target': 'target',
 'time_limit': 600,
 'verbosity': 2}

train_data with frequency 'None' has been resampled to frequency 'B'.
window split n*4; validation score: -0.008534841162481415
best score: best_validate_score_metric = -0.006502177090277285
score: validate_score[score_metric] = -0.008534841162481415
window_split_index = 5
```

```
Provided train_data has 493548 rows (NaN fraction=3.3%), 499 time series. Median time series length is 1010 (min=15,
max=1010).
tuning_data with frequency 'None' has been resampled to frequency 'B'.
Provided tuning_data has 493548 rows (NaN fraction=3.3%), 499 time series. Median time series length is 1010 (min=1
5, max=1010).
        Setting num_val_windows = 0 (disabling backtesting on train_data) because tuning_data is provided.

Provided data contains following columns:
        target: 'target'
        past_covariates:
                categorical:        []
                continuous (float): ['past_covariate_open', 'past_covariate_high', 'past_covariate_low', 'past_covar
iate_volume']

To learn how to fix incorrectly inferred types, please see documentation for TimeSeriesPredictor.fit

AutoGluon will gauge predictive performance using evaluation metric: 'MAPE'
        This metric's sign has been flipped to adhere to being higher_is_better. The metric score can be multiplied
by -1 to get the metric value.
===================================================

Starting training. Start time is 2024-10-22 12:10:19
Models that will be trained: ['Naive', 'SeasonalNaive', 'RecursiveTabular', 'DirectTabular', 'ETS', 'Theta']
Training timeseries model Naive. Training for up to 85.1s of the 595.9s of remaining time.
        -0.0048       = Validation score (-MAPE)
        0.02    s     = Training runtime
        1.38    s     = Validation (prediction) runtime
Training timeseries model SeasonalNaive. Training for up to 99.1s of the 594.4s of remaining time.
        -0.0147       = Validation score (-MAPE)
        0.02    s     = Training runtime
        0.25    s     = Validation (prediction) runtime
Training timeseries model RecursiveTabular. Training for up to 118.8s of the 594.0s of remaining time.
        -0.0130       = Validation score (-MAPE)
        0.99    s     = Training runtime
        0.20    s     = Validation (prediction) runtime
Training timeseries model DirectTabular. Training for up to 148.2s of the 592.7s of remaining time.
        -0.0076       = Validation score (-MAPE)
        1.92    s     = Training runtime
        0.30    s     = Validation (prediction) runtime
Training timeseries model ETS. Training for up to 196.8s of the 590.4s of remaining time.
```

```
        -0.0048       = Validation score (-MAPE)
        0.07    s     = Training runtime
        93.60   s     = Validation (prediction) runtime
Training timeseries model Theta. Training for up to 248.3s of the 496.6s of remaining time.
        -0.0046       = Validation score (-MAPE)
        0.07    s     = Training runtime
        11.43   s     = Validation (prediction) runtime
Fitting simple weighted ensemble.
        Ensemble weights: {'Theta': 1.0}
        -0.0046       = Validation score (-MAPE)
        0.30    s     = Training runtime
        11.43   s     = Validation (prediction) runtime
Training complete. Models trained: ['Naive', 'SeasonalNaive', 'RecursiveTabular', 'DirectTabular', 'ETS', 'Theta',
'WeightedEnsemble']
Total runtime: 111.34 s
Best model: Theta
Best model score: -0.0046
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: Theta
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: Theta
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: Theta
Beginning AutoGluon training... Time limit = 600s
AutoGluon will save models to '../model/autogluon_model'
=================== System Info ===================
AutoGluon Version:  1.1.1
Python Version:     3.11.7
Operating System:   Linux
Platform Machine:   x86_64
Platform Version:   #1 SMP PREEMPT_DYNAMIC Mon, 30 Sep 2024 23:49:50 +0000
CPU Count:          8
GPU Count:          0
Memory Avail:       1.58 GB / 7.60 GB (20.8%)
Disk Space Avail:   19.59 GB / 109.46 GB (17.9%)
===================================================
Setting presets to: fast_training

Fitting with arguments:
{'enable_ensemble': True,
 'eval_metric': MAPE,
```

```
 'freq': 'B',
 'hyperparameters': 'very_light',
 'known_covariates_names': [],
 'num_val_windows': 1,
 'prediction_length': 1,
 'quantile_levels': [],
 'random_seed': 123,
 'refit_every_n_windows': 1,
 'refit_full': False,
 'skip_model_selection': False,
 'target': 'target',
 'time_limit': 600,
 'verbosity': 2}

train_data with frequency 'None' has been resampled to frequency 'B'.
window split n*5; validation score: -0.004643084011990639
best score: best_validate_score_metric = -0.006502177090277285
score: validate_score[score_metric] = -0.004643084011990639
        new best model found ! best_predictor =<autogluon.timeseries.predictor.TimeSeriesPredictor object at 0x74a7b
4d04290>
window_split_index = 6
```

```
Provided train_data has 494047 rows (NaN fraction=3.3%), 499 time series. Median time series length is 1011 (min=16,
max=1011).
tuning_data with frequency 'None' has been resampled to frequency 'B'.
Provided tuning_data has 494047 rows (NaN fraction=3.3%), 499 time series. Median time series length is 1011 (min=1
6, max=1011).
        Setting num_val_windows = 0 (disabling backtesting on train_data) because tuning_data is provided.

Provided data contains following columns:
        target: 'target'
        past_covariates:
                categorical:        []
                continuous (float): ['past_covariate_open', 'past_covariate_high', 'past_covariate_low', 'past_covar
iate_volume']

To learn how to fix incorrectly inferred types, please see documentation for TimeSeriesPredictor.fit

AutoGluon will gauge predictive performance using evaluation metric: 'MAPE'
        This metric's sign has been flipped to adhere to being higher_is_better. The metric score can be multiplied
by -1 to get the metric value.
===================================================

Starting training. Start time is 2024-10-22 12:13:22
Models that will be trained: ['Naive', 'SeasonalNaive', 'RecursiveTabular', 'DirectTabular', 'ETS', 'Theta']
Training timeseries model Naive. Training for up to 85.0s of the 595.3s of remaining time.
        -0.0045       = Validation score (-MAPE)
        0.02    s     = Training runtime
        1.41    s     = Validation (prediction) runtime
Training timeseries model SeasonalNaive. Training for up to 99.0s of the 593.7s of remaining time.
        -0.0127       = Validation score (-MAPE)
        0.02    s     = Training runtime
        0.26    s     = Validation (prediction) runtime
Training timeseries model RecursiveTabular. Training for up to 118.7s of the 593.3s of remaining time.
        -0.0127       = Validation score (-MAPE)
        1.04    s     = Training runtime
        0.21    s     = Validation (prediction) runtime
Training timeseries model DirectTabular. Training for up to 148.0s of the 592.0s of remaining time.
        -0.0062       = Validation score (-MAPE)
        1.35    s     = Training runtime
        0.34    s     = Validation (prediction) runtime
Training timeseries model ETS. Training for up to 196.7s of the 590.2s of remaining time.
```

```
        -0.0044       = Validation score (-MAPE)
        0.07    s     = Training runtime
        94.97   s     = Validation (prediction) runtime
Training timeseries model Theta. Training for up to 247.5s of the 495.0s of remaining time.
        -0.0041       = Validation score (-MAPE)
        0.07    s     = Training runtime
        11.69   s     = Validation (prediction) runtime
Fitting simple weighted ensemble.
        Ensemble weights: {'RecursiveTabular': 0.05, 'Theta': 0.95}
        -0.0041       = Validation score (-MAPE)
        0.30    s     = Training runtime
        11.90   s     = Validation (prediction) runtime
Training complete. Models trained: ['Naive', 'SeasonalNaive', 'RecursiveTabular', 'DirectTabular', 'ETS', 'Theta',
'WeightedEnsemble']
Total runtime: 112.57 s
Best model: WeightedEnsemble
Best model score: -0.0041
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
Beginning AutoGluon training... Time limit = 600s
AutoGluon will save models to '../model/autogluon_model'
=================== System Info ===================
AutoGluon Version:  1.1.1
Python Version:     3.11.7
Operating System:   Linux
Platform Machine:   x86_64
Platform Version:   #1 SMP PREEMPT_DYNAMIC Mon, 30 Sep 2024 23:49:50 +0000
CPU Count:          8
GPU Count:          0
Memory Avail:       1.69 GB / 7.60 GB (22.2%)
Disk Space Avail:   19.59 GB / 109.46 GB (17.9%)
===================================================
Setting presets to: fast_training

Fitting with arguments:
{'enable_ensemble': True,
 'eval_metric': MAPE,
```

```
 'freq': 'B',
 'hyperparameters': 'very_light',
 'known_covariates_names': [],
 'num_val_windows': 1,
 'prediction_length': 1,
 'quantile_levels': [],
 'random_seed': 123,
 'refit_every_n_windows': 1,
 'refit_full': False,
 'skip_model_selection': False,
 'target': 'target',
 'time_limit': 600,
 'verbosity': 2}

train_data with frequency 'None' has been resampled to frequency 'B'.
window split n*6; validation score: -0.004060298016537789
best score: best_validate_score_metric = -0.004643084011990639
score: validate_score[score_metric] = -0.004060298016537789
        new best model found ! best_predictor =<autogluon.timeseries.predictor.TimeSeriesPredictor object at 0x74a7b
5a5da50>
window_split_index = 7
```

```
Provided train_data has 495045 rows (NaN fraction=3.4%), 499 time series. Median time series length is 1013 (min=18,
max=1013).
tuning_data with frequency 'None' has been resampled to frequency 'B'.
Provided tuning_data has 495045 rows (NaN fraction=3.4%), 499 time series. Median time series length is 1013 (min=1
8, max=1013).
        Setting num_val_windows = 0 (disabling backtesting on train_data) because tuning_data is provided.

Provided data contains following columns:
        target: 'target'
        past_covariates:
                categorical:        []
                continuous (float): ['past_covariate_open', 'past_covariate_high', 'past_covariate_low', 'past_covar
iate_volume']

To learn how to fix incorrectly inferred types, please see documentation for TimeSeriesPredictor.fit

AutoGluon will gauge predictive performance using evaluation metric: 'MAPE'
        This metric's sign has been flipped to adhere to being higher_is_better. The metric score can be multiplied
by -1 to get the metric value.
===================================================

Starting training. Start time is 2024-10-22 12:16:28
Models that will be trained: ['Naive', 'SeasonalNaive', 'RecursiveTabular', 'DirectTabular', 'ETS', 'Theta']
Training timeseries model Naive. Training for up to 85.0s of the 595.3s of remaining time.
        -0.0108       = Validation score (-MAPE)
        0.02    s     = Training runtime
        1.52    s     = Validation (prediction) runtime
Training timeseries model SeasonalNaive. Training for up to 98.9s of the 593.7s of remaining time.
        -0.0138       = Validation score (-MAPE)
        0.02    s     = Training runtime
        0.34    s     = Validation (prediction) runtime
Training timeseries model RecursiveTabular. Training for up to 118.6s of the 593.2s of remaining time.
        -0.0148       = Validation score (-MAPE)
        1.03    s     = Training runtime
        0.20    s     = Validation (prediction) runtime
Training timeseries model DirectTabular. Training for up to 148.0s of the 591.9s of remaining time.
        -0.0121       = Validation score (-MAPE)
        2.04    s     = Training runtime
        0.34    s     = Validation (prediction) runtime
Training timeseries model ETS. Training for up to 196.5s of the 589.4s of remaining time.
```

```
        -0.0110       = Validation score (-MAPE)
        0.07    s     = Training runtime
        94.26   s     = Validation (prediction) runtime
Training timeseries model Theta. Training for up to 247.5s of the 494.9s of remaining time.
        -0.0119       = Validation score (-MAPE)
        0.07    s     = Training runtime
        11.57   s     = Validation (prediction) runtime
Fitting simple weighted ensemble.
        Ensemble weights: {'Naive': 0.81, 'SeasonalNaive': 0.19}
        -0.0106       = Validation score (-MAPE)
        0.30    s     = Training runtime
        1.87    s     = Validation (prediction) runtime
Training complete. Models trained: ['Naive', 'SeasonalNaive', 'RecursiveTabular', 'DirectTabular', 'ETS', 'Theta',
'WeightedEnsemble']
Total runtime: 112.60 s
Best model: WeightedEnsemble
Best model score: -0.0106
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
Beginning AutoGluon training... Time limit = 600s
AutoGluon will save models to '../model/autogluon_model'
=================== System Info ===================
AutoGluon Version:  1.1.1
Python Version:     3.11.7
Operating System:   Linux
Platform Machine:   x86_64
Platform Version:   #1 SMP PREEMPT_DYNAMIC Mon, 30 Sep 2024 23:49:50 +0000
CPU Count:          8
GPU Count:          0
Memory Avail:       2.47 GB / 7.60 GB (32.6%)
Disk Space Avail:   19.59 GB / 109.46 GB (17.9%)
====================================================
Setting presets to: fast_training

Fitting with arguments:
{'enable_ensemble': True,
 'eval_metric': MAPE,
```

```
 'freq': 'B',
 'hyperparameters': 'very_light',
 'known_covariates_names': [],
 'num_val_windows': 1,
 'prediction_length': 1,
 'quantile_levels': [],
 'random_seed': 123,
 'refit_every_n_windows': 1,
 'refit_full': False,
 'skip_model_selection': False,
 'target': 'target',
 'time_limit': 600,
 'verbosity': 2}

train_data with frequency 'None' has been resampled to frequency 'B'.
window split n*7; validation score: -0.010623202792195665
best score: best_validate_score_metric = -0.004060298016537789
score: validate_score[score_metric] = -0.010623202792195665
window_split_index = 8
```

```
Provided train_data has 495544 rows (NaN fraction=3.4%), 499 time series. Median time series length is 1014 (min=19,
max=1014).
tuning_data with frequency 'None' has been resampled to frequency 'B'.
Provided tuning_data has 495544 rows (NaN fraction=3.4%), 499 time series. Median time series length is 1014 (min=1
9, max=1014).
        Setting num_val_windows = 0 (disabling backtesting on train_data) because tuning_data is provided.

Provided data contains following columns:
        target: 'target'
        past_covariates:
                categorical:        []
                continuous (float): ['past_covariate_open', 'past_covariate_high', 'past_covariate_low', 'past_covar
iate_volume']

To learn how to fix incorrectly inferred types, please see documentation for TimeSeriesPredictor.fit

AutoGluon will gauge predictive performance using evaluation metric: 'MAPE'
        This metric's sign has been flipped to adhere to being higher_is_better. The metric score can be multiplied
by -1 to get the metric value.
===================================================

Starting training. Start time is 2024-10-22 12:18:38
Models that will be trained: ['Naive', 'SeasonalNaive', 'RecursiveTabular', 'DirectTabular', 'ETS', 'Theta']
Training timeseries model Naive. Training for up to 85.1s of the 596.0s of remaining time.
        -0.0055       = Validation score (-MAPE)
        0.02    s     = Training runtime
        1.43    s     = Validation (prediction) runtime
Training timeseries model SeasonalNaive. Training for up to 99.1s of the 594.4s of remaining time.
        -0.0103       = Validation score (-MAPE)
        0.02    s     = Training runtime
        0.22    s     = Validation (prediction) runtime
Training timeseries model RecursiveTabular. Training for up to 118.8s of the 594.0s of remaining time.
        -0.0119       = Validation score (-MAPE)
        0.96    s     = Training runtime
        0.19    s     = Validation (prediction) runtime
Training timeseries model DirectTabular. Training for up to 148.2s of the 592.8s of remaining time.
        -0.0078       = Validation score (-MAPE)
        1.13    s     = Training runtime
        0.31    s     = Validation (prediction) runtime
Training timeseries model ETS. Training for up to 197.1s of the 591.2s of remaining time.
```

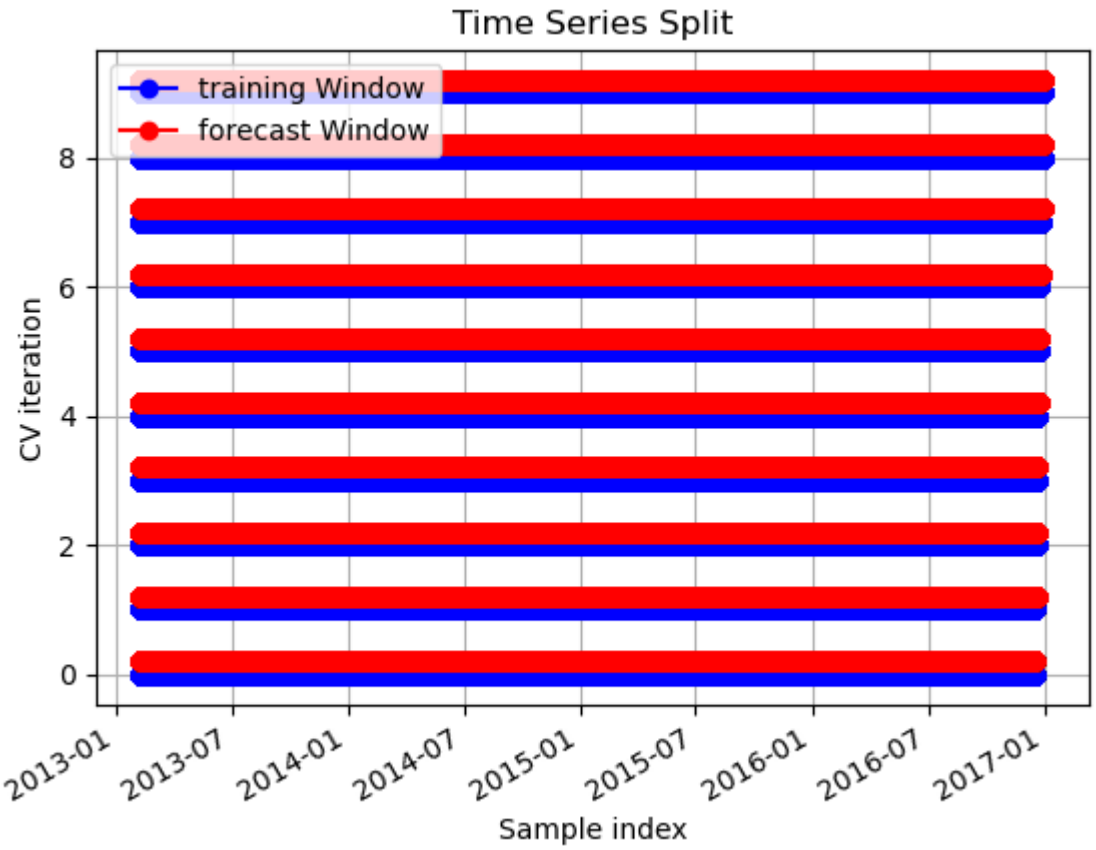
```
        -0.0055       = Validation score (-MAPE)
        0.07    s     = Training runtime
        98.04   s     = Validation (prediction) runtime
Training timeseries model Theta. Training for up to 246.4s of the 492.9s of remaining time.
        -0.0058       = Validation score (-MAPE)
        0.14    s     = Training runtime
        11.43   s     = Validation (prediction) runtime
Fitting simple weighted ensemble.
        Ensemble weights: {'DirectTabular': 0.12, 'Naive': 0.88}
        -0.0054       = Validation score (-MAPE)
        0.31    s     = Training runtime
        1.74    s     = Validation (prediction) runtime
Training complete. Models trained: ['Naive', 'SeasonalNaive', 'RecursiveTabular', 'DirectTabular', 'ETS', 'Theta',
'WeightedEnsemble']
Total runtime: 115.24 s
Best model: WeightedEnsemble
Best model score: -0.0054
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
Beginning AutoGluon training... Time limit = 600s
AutoGluon will save models to '../model/autogluon_model'
=================== System Info ===================
AutoGluon Version:  1.1.1
Python Version:     3.11.7
Operating System:   Linux
Platform Machine:   x86_64
Platform Version:   #1 SMP PREEMPT_DYNAMIC Mon, 30 Sep 2024 23:49:50 +0000
CPU Count:          8
GPU Count:          0
Memory Avail:       2.50 GB / 7.60 GB (32.9%)
Disk Space Avail:   19.59 GB / 109.46 GB (17.9%)
===================================================
Setting presets to: fast_training

Fitting with arguments:
{'enable_ensemble': True,
 'eval_metric': MAPE,
```

```
 'freq': 'B',
 'hyperparameters': 'very_light',
 'known_covariates_names': [],
 'num_val_windows': 1,
 'prediction_length': 1,
 'quantile_levels': [],
 'random_seed': 123,
 'refit_every_n_windows': 1,
 'refit_full': False,
 'skip_model_selection': False,
 'target': 'target',
 'time_limit': 600,
 'verbosity': 2}

train_data with frequency 'None' has been resampled to frequency 'B'.
window split n*8; validation score: -0.00540842185732615
best score: best_validate_score_metric = -0.004060298016537789
score: validate_score[score_metric] = -0.00540842185732615
window_split_index = 9
```

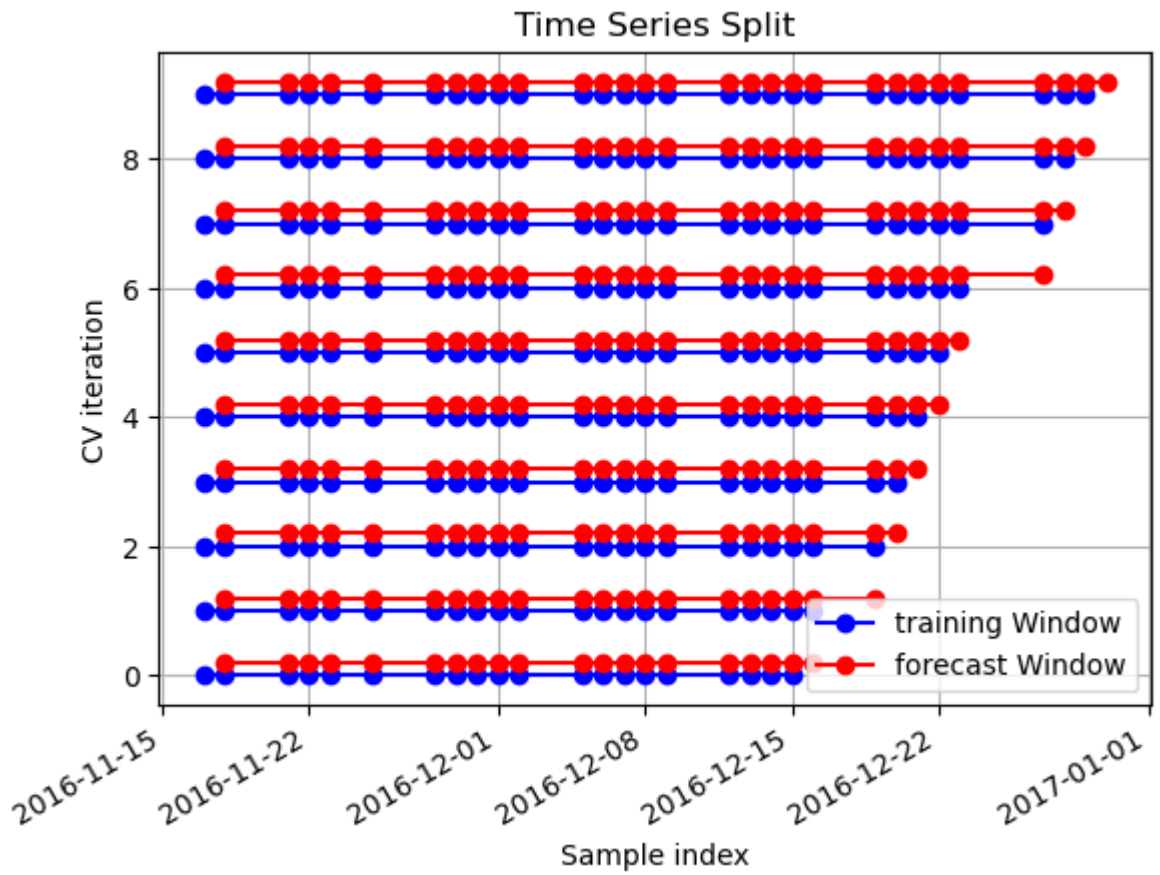
```
Provided train_data has 496043 rows (NaN fraction=3.4%), 499 time series. Median time series length is 1015 (min=20,
max=1015).
tuning_data with frequency 'None' has been resampled to frequency 'B'.
Provided tuning_data has 496043 rows (NaN fraction=3.4%), 499 time series. Median time series length is 1015 (min=2
0, max=1015).
        Setting num_val_windows = 0 (disabling backtesting on train_data) because tuning_data is provided.

Provided data contains following columns:
        target: 'target'
        past_covariates:
                categorical:        []
                continuous (float): ['past_covariate_open', 'past_covariate_high', 'past_covariate_low', 'past_covar
iate_volume']

To learn how to fix incorrectly inferred types, please see documentation for TimeSeriesPredictor.fit

AutoGluon will gauge predictive performance using evaluation metric: 'MAPE'
        This metric's sign has been flipped to adhere to being higher_is_better. The metric score can be multiplied
by -1 to get the metric value.
===================================================

Starting training. Start time is 2024-10-22 12:20:52
Models that will be trained: ['Naive', 'SeasonalNaive', 'RecursiveTabular', 'DirectTabular', 'ETS', 'Theta']
Training timeseries model Naive. Training for up to 85.1s of the 595.8s of remaining time.
        -0.0069       = Validation score (-MAPE)
        0.02    s     = Training runtime
        1.41    s     = Validation (prediction) runtime
Training timeseries model SeasonalNaive. Training for up to 99.0s of the 594.2s of remaining time.
        -0.0143       = Validation score (-MAPE)
        0.02    s     = Training runtime
        0.31    s     = Validation (prediction) runtime
Training timeseries model RecursiveTabular. Training for up to 118.8s of the 593.8s of remaining time.
        -0.0159       = Validation score (-MAPE)
        0.98    s     = Training runtime
        0.21    s     = Validation (prediction) runtime
Training timeseries model DirectTabular. Training for up to 148.1s of the 592.5s of remaining time.
        -0.0101       = Validation score (-MAPE)
        1.21    s     = Training runtime
        0.34    s     = Validation (prediction) runtime
Training timeseries model ETS. Training for up to 196.9s of the 590.8s of remaining time.
```

```
        -0.0070       = Validation score (-MAPE)
        0.07    s     = Training runtime
        94.26   s     = Validation (prediction) runtime
Training timeseries model Theta. Training for up to 248.2s of the 496.4s of remaining time.
        -0.0074       = Validation score (-MAPE)
        0.07    s     = Training runtime
        11.59   s     = Validation (prediction) runtime
Fitting simple weighted ensemble.
        Ensemble weights: {'DirectTabular': 0.01, 'Naive': 0.99}
        -0.0069       = Validation score (-MAPE)
        0.30    s     = Training runtime
        1.75    s     = Validation (prediction) runtime
Training complete. Models trained: ['Naive', 'SeasonalNaive', 'RecursiveTabular', 'DirectTabular', 'ETS', 'Theta',
'WeightedEnsemble']
Total runtime: 111.60 s
Best model: WeightedEnsemble
Best model score: -0.0069
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
window split n*9; validation score: -0.006895487288128591
best score: best_validate_score_metric = -0.004060298016537789
score: validate_score[score_metric] = -0.006895487288128591
done best_predictor =<autogluon.timeseries.predictor.TimeSeriesPredictor object at 0x74a7b5a5da50>
```

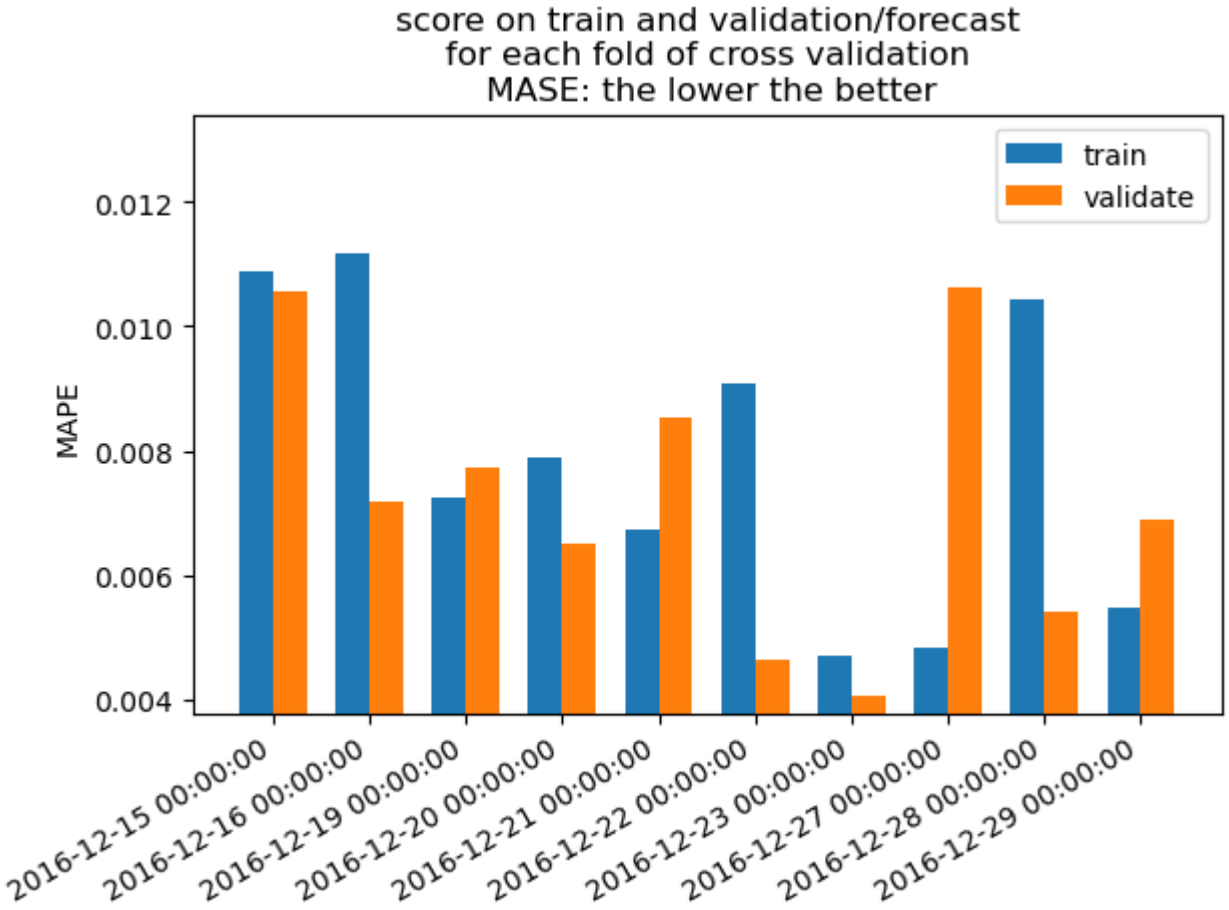
In [17]:
```python
# strat.plot_windows(
strat.plot_windows(
    train_windows=info['train_windows'],
    forecast_windows=info['validate_windows'],
    folder_path=constant.graph_folder_path,
    file_name='autogluon_cross_validation_time_windows.png',
)
```



In [18]:
```
# last 20 points for each fold
strat.plot_windows(
    train_windows=[window[-20-i:] for i,window in enumerate(info['train_windows'])],
    forecast_windows=[window[-20-i:] for i,window in enumerate(info['validate_windows'])],
    folder_path=constant.graph_folder_path,
    file_name='cross_validation_time_windows_tails.png',
)
```



In [19]:
```
chart = strat.create_grouped_bar_chart(
    group_labels=[window[-1] for window in info['train_windows']],
    group1_data=[-d['MAPE'] for d in info['train_scores']],
    group2_data=[-d['MAPE'] for d in info['validate_scores']],
    group1_label='train',
    group2_label='validate',
    y_axis_label=list((info['train_scores'])[0].keys())[0],
    chart_title=f"score on train and validation/forecast\nfor each fold of cross validation\n MASE: the lower the b
    file_name='metric_train.png',
)
```



In [52]:
```
try:
    _ = best_predictor.predict(nontest_time_series_data_frame)
except:
    best_fold_index = None
    best_validate_score = -9999999999
    for i, score in enumerate(info['validate_scores']):
        if -score['MAPE'] > best_validate_score:
            best_fold_index = i
            best_validate_score = -score['MAPE']
    train_split_dates = info["train_windows"][best_fold_index]
    train_split = nontest_time_series_data_frame[
        (timestamps(nontest_time_series_data_frame) >= numpy.min(train_split_dates) ) &
```

```
    (timestamps(nontest_time_series_data_frame) <= numpy.max(train_split_dates) )
]
validate_split_dates = info["validate_windows"][best_fold_index]
validate_split = nontest_time_series_data_frame[
    (timestamps(nontest_time_series_data_frame) >= numpy.min(validate_split_dates) ) &
    (timestamps(nontest_time_series_data_frame) <= numpy.max(validate_split_dates) )
]
best_predictor = timeseries.TimeSeriesPredictor(
    target='target',
    prediction_length=1,
    freq='B',
    eval_metric='MASE',
    quantile_levels=[],
    path=os.path.join(constant.model_folder_path, "try_again_autogluon_model"),
    verbosity=2,
)

best_predictor.fit(
    train_data=train_split,
    tuning_data=validate_split,
    # presets='fast_training',
    presets='medium_quality',
    # 'WeightedEnsemble'
    # excluded_model_types=['SeasonalNaive','RecursiveTabular'],
    # excluded_model_types=['WeightedEnsemble'],
    # number_val_windows=5,
    # time_limit=60*40, # seconds !!!!!!!!!!
    # time_limit=train_time_limit, # seconds !!!!!!!!!!
)
```

```
Beginning AutoGluon training...
AutoGluon will save models to '../model/try_again_autogluon_model'
=================== System Info ===================
AutoGluon Version:  1.1.1
Python Version:     3.11.7
Operating System:   Linux
Platform Machine:   x86_64
Platform Version:   #1 SMP PREEMPT_DYNAMIC Mon, 30 Sep 2024 23:49:50 +0000
CPU Count:          8
GPU Count:          0
Memory Avail:       2.26 GB / 7.60 GB (29.7%)
Disk Space Avail:   19.60 GB / 109.46 GB (17.9%)
===================================================
Setting presets to: medium_quality

Fitting with arguments:
{'enable_ensemble': True,
 'eval_metric': MASE,
 'freq': 'B',
 'hyperparameters': 'light',
 'known_covariates_names': [],
 'num_val_windows': 1,
 'prediction_length': 1,
 'quantile_levels': [],
 'random_seed': 123,
 'refit_every_n_windows': 1,
 'refit_full': False,
 'skip_model_selection': False,
 'target': 'target',
 'verbosity': 2}

train_data with frequency 'None' has been resampled to frequency 'B'.
DatetimeIndex(['2013-02-08', '2013-02-11', '2013-02-12', '2013-02-13',
               '2013-02-14', '2013-02-15', '2013-02-19', '2013-02-20',
               '2013-02-21', '2013-02-22',
               ...
               '2016-12-13', '2016-12-14', '2016-12-15', '2016-12-16',
               '2016-12-19', '2016-12-20', '2016-12-21', '2016-12-22',
               '2016-12-23', '2016-12-27'],
              dtype='datetime64[ns]', name='timestamp', length=478371, freq=None)
```

```
Provided train_data has 495045 rows (NaN fraction=3.4%), 499 time series. Median time series length is 1013 (min=18,
max=1013).
tuning_data with frequency 'None' has been resampled to frequency 'B'.
Provided tuning_data has 495045 rows (NaN fraction=3.4%), 499 time series. Median time series length is 1013 (min=1
8, max=1013).
        Setting num_val_windows = 0 (disabling backtesting on train_data) because tuning_data is provided.

Provided data contains following columns:
        target: 'target'
        past_covariates:
                categorical:        []
                continuous (float): ['past_covariate_open', 'past_covariate_high', 'past_covariate_low', 'past_covar
iate_volume']

To learn how to fix incorrectly inferred types, please see documentation for TimeSeriesPredictor.fit

AutoGluon will gauge predictive performance using evaluation metric: 'MASE'
        This metric's sign has been flipped to adhere to being higher_is_better. The metric score can be multiplied
by -1 to get the metric value.
===================================================

Starting training. Start time is 2024-10-22 12:59:43
Models that will be trained: ['Naive', 'SeasonalNaive', 'RecursiveTabular', 'DirectTabular', 'ETS', 'Theta', 'Tempor
alFusionTransformer']
Training timeseries model Naive.
        -0.5121       = Validation score (-MASE)
        0.02    s     = Training runtime
        1.36    s     = Validation (prediction) runtime
Training timeseries model SeasonalNaive.
        -0.6136       = Validation score (-MASE)
        0.02    s     = Training runtime
        0.25    s     = Validation (prediction) runtime
Training timeseries model RecursiveTabular.
        -0.4469       = Validation score (-MASE)
        8.86    s     = Training runtime
        0.32    s     = Validation (prediction) runtime
Training timeseries model DirectTabular.
        -0.6269       = Validation score (-MASE)
        6.10    s     = Training runtime
        0.58    s     = Validation (prediction) runtime
```

```
Training timeseries model ETS.
        -0.5206       = Validation score (-MASE)
        0.07    s     = Training runtime
        98.16   s     = Validation (prediction) runtime
Training timeseries model Theta.
        -0.5579       = Validation score (-MASE)
        0.07    s     = Training runtime
        11.95   s     = Validation (prediction) runtime
Training timeseries model TemporalFusionTransformer.
        -0.2983       = Validation score (-MASE)
        1937.94 s     = Training runtime
        0.93    s     = Validation (prediction) runtime
Fitting simple weighted ensemble.
        Ensemble weights: {'RecursiveTabular': 0.18, 'TemporalFusionTransformer': 0.82}
        -0.2880       = Validation score (-MASE)
        0.17    s     = Training runtime
        1.25    s     = Validation (prediction) runtime
Training complete. Models trained: ['Naive', 'SeasonalNaive', 'RecursiveTabular', 'DirectTabular', 'ETS', 'Theta', '
TemporalFusionTransformer', 'WeightedEnsemble']
Total runtime: 2067.90 s
Best model: WeightedEnsemble
Best model score: -0.2880
```

In [53]:
```
_ = best_predictor.predict(nontest_time_series_data_frame)
```

```
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
```

In [54]:
```
# # we retraine the model without tme limit on all training data (the cross validateion is done internally)
# # cause it crash on predict when trained splitted time window during manual cross valdiation
# new_predictor = timeseries.TimeSeriesPredictor(
#     target='target',
#     prediction_length=1,
#     freq='B',
#     # freq='D',
#     eval_metric='MAPE',
#     quantile_levels=[],
#     path=os.path.join(constant.model_folder_path, "retrained_autogluon_model"),
#     verbosity=2,
# )
# new_predictor.fit(
```

```
#     train_data=nontest_time_series_data_frame,
#     presets='fast_training',
#     time_limit=30*60,
# )
new_predictor = best_predictor
```

In [55]:
```
print(f"prices (target) just before {constant.test_start_date}:")
nontest_time_series_data_frame[
    timestamps(nontest_time_series_data_frame) == timestamps(nontest_time_series_data_frame)[-1]
]
```

```
prices (target) just before 2017-01-01:
```

Out[55]:

item_id	timestamp	past_covariate_open	past_covariate_high	past_covariate_low	target	past_covariate_volume
A	**2016-12-30**	45.760	45.82	45.375	45.56	1216100
AAL	**2016-12-30**	47.420	47.66	46.470	46.69	4495016
AAP	**2016-12-30**	171.320	172.00	168.600	169.12	513003
AAPL	**2016-12-30**	116.650	117.20	115.430	115.82	30586265
ABBV	**2016-12-30**	62.729	62.93	62.410	62.62	5999195
...	**...**	...	...	...	...	...
XYL	**2016-12-30**	49.980	50.00	49.360	49.52	646428
YUM	**2016-12-30**	63.930	63.94	63.160	63.33	1887055
ZBH	**2016-12-30**	103.310	103.93	102.850	103.20	973822
ZION	**2016-12-30**	43.070	43.31	42.690	43.04	1938930
ZTS	**2016-12-30**	53.640	53.74	53.270	53.53	1701204

499 rows × 5 columns

In [56]:
```
print(f"example of predicted prices (target) at {constant.test_start_date}:")
```

```
new_predictor.predict(nontest_time_series_data_frame)
```

```
data with frequency 'None' has been resampled to frequency 'B'.
example of predicted prices (target) at 2017-01-01:
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
```

Out[56]:

		mean
item_id	**timestamp**	
A	**2017-01-02**	45.508049
AAL	**2017-01-02**	46.682302
AAP	**2017-01-02**	169.303435
AAPL	**2017-01-02**	115.876835
ABBV	**2017-01-02**	62.729101
...	**...**	...
XYL	**2017-01-02**	49.494562
YUM	**2017-01-02**	62.881351
ZBH	**2017-01-02**	102.833213
ZION	**2017-01-02**	43.060032
ZTS	**2017-01-02**	53.513590

499 rows × 1 columns

In [57]:
```
# actual first date of the test data_set
print(f"actual prices (targets) at the date at {constant.test_start_date}:")
test_time_series_data_frame[
    timestamps(test_time_series_data_frame) == timestamps(test_time_series_data_frame)[0]
]
```

```
actual prices (targets) at the date at 2017-01-01:
```

Out[57]:

item_id	timestamp	past_covariate_open	past_covariate_high	past_covariate_low	target	past_covariate_volume	actual_future_retu
A	**2017-01-03**	45.93	46.7500	45.740	46.49	1739726	0.0131
AAL	**2017-01-03**	47.28	47.3400	46.135	46.30	6737752	0.0086
AAP	**2017-01-03**	170.78	171.3600	169.310	170.60	691526	0.0082
AAPL	**2017-01-03**	115.80	116.3300	114.760	116.15	28781865	-0.0011
ABBV	**2017-01-03**	62.92	63.0300	61.935	62.41	9328198	0.0141
...	**...**	...	...	...	...	...	
XYL	**2017-01-03**	49.95	50.2500	49.210	49.65	1596877	0.0149
YUM	**2017-01-03**	63.56	63.7899	62.820	63.21	4793381	0.0036
ZBH	**2017-01-03**	103.87	103.9000	102.590	103.33	1381230	0.0091
ZION	**2017-01-03**	43.74	44.3300	42.720	43.18	2896301	0.0143
ZTS	**2017-01-03**	53.88	54.3800	53.335	53.59	3580369	0.0097

499 rows × 6 columns

backtesting

we take the model trained on the nontest data set at make it predict "the next day's price"
at numerous random dates of the test data set:
then when we have all the actual returns for the given dates we average it

- for each date; we then give the model/predicto all history up to the price to be predicted
- for all those predicted prices we calculate the future return
  - ~ how much in % that price would worth the next day

- we pick the best ticker (with the highest **predicted** return) and return the actual return
  - remember that we are predicting next day values of a **past** data set so we can compare prediction with reality
- this strategy would be:

> *for each day I predict the company whose stock will*
> *increase the most and go all on that one and sell it the next day*

we take the same dates from sp500 (standard and poors)
~ a preselected averaged group of stock options (no picking to do here) and calculate what would have been the overall return
we comapre both return and check what strategy would have bring more cash

note: we take n random dates but a real serious way would be predicted for all dates of the testing period but the prediction is kinda slow on my laptop so I assumed that is good enough. if ti was real money I would simply buy/rent the needed machine

In [58]:
```
def back_testing_stocks(time_series_data_frame, predictor, backtesting_dates):
    n_best_predicted_returns = []
    n_actual_returns = []
    time_series_data_frame['actual_future_return'] = time_series.calculate_future_return_series(time_series_data_fr
    for backtesting_date in backtesting_dates:
        up_to_backtesting_date = time_series.from_first_up_to_before_last_date_series(time_series_data_frame,backte
        next_rows_as_prediction = time_series.append_predicted_next_row(up_to_backtesting_date, predictor)
        two_latest_rows_only = time_series.keep_only_two_latest_rows(next_rows_as_prediction)
        two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_row
        best_predicted_future_return_row = two_latest_rows_only[
            two_latest_rows_only['predicted_future_return'] == two_latest_rows_only['predicted_future_return'].max(
        ]
        n_actual_returns.append(best_predicted_future_return_row['actual_future_return'].values[0])
        n_best_predicted_returns.append(best_predicted_future_return_row['predicted_future_return'].values[0])
    return n_actual_returns, n_best_predicted_returns
```

In [59]:
```
def back_testing_sp500(time_series_data_frame, backtesting_dates):
    n_actual_returns = []
    all_future_return_series = time_series.calculate_future_return_series(time_series_data_frame['target'])
    time_series_data_frame['actual_future_return'] = time_series.calculate_future_return_series(time_series_data_fr

    for backtesting_date in backtesting_dates:
        future_return_at_the_given_date = sp500_test_time_series_data_frame[
```

```
            timestamps(sp500_test_time_series_data_frame) == backtesting_date
        ]['actual_future_return']
        n_actual_returns.append(future_return_at_the_given_date)
    return n_actual_returns
```

In [60]:
```
# n_random_dates = time_series.pick_backtesting_dates_from_time_series_data_frame(test_time_series_data_frame, (num
# backtesting_dates = numpy.sort(n_random_dates)
backtesting_dates = numpy.sort(numpy.unique(timestamps(test_time_series_data_frame)))
```

In [61]:
```
number_of_dates = len(backtesting_dates)
```

In [62]:
```
n_picked_ticker_actual_returns, n_picked_ticker_best_predicted_returns = back_testing_stocks(
    time_series_data_frame=test_time_series_data_frame,
    predictor=new_predictor,
    backtesting_dates=backtesting_dates,
)
n_sp500_actual_returns = back_testing_sp500(
    time_series_data_frame=sp500_test_time_series_data_frame,
    backtesting_dates=backtesting_dates,
)
```

```
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
Warning: 499 time series (100.0%) are shorter than 5 and cannot be predicted by RecursiveTabular. Fallback model Sea
sonalNaive is used for these time series.
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
Warning: 500 time series (100.0%) are shorter than 5 and cannot be predicted by RecursiveTabular. Fallback model Sea
sonalNaive is used for these time series.
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'D' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
Warning: 500 time series (100.0%) are shorter than 5 and cannot be predicted by RecursiveTabular. Fallback model Sea
sonalNaive is used for these time series.
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'D' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
Warning: 500 time series (100.0%) are shorter than 5 and cannot be predicted by RecursiveTabular. Fallback model Sea
sonalNaive is used for these time series.
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
```

```
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
Warning: 500 time series (100.0%) are shorter than 5 and cannot be predicted by RecursiveTabular. Fallback model Sea
sonalNaive is used for these time series.
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
Warning: 500 time series (100.0%) are shorter than 5 and cannot be predicted by RecursiveTabular. Fallback model Sea
sonalNaive is used for these time series.
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
Warning: 1 time series (0.2%) are shorter than 5 and cannot be predicted by RecursiveTabular. Fallback model Seasona
lNaive is used for these time series.
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
```

```
['target'])
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
```

```
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
```

```
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
```

```
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
```

```
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
```

```
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
```

```
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
```

```
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
```

```
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
```

```
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
```

```
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
```

```
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
Warning: 1 time series (0.2%) are shorter than 5 and cannot be predicted by RecursiveTabular. Fallback model Seasona
lNaive is used for these time series.
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
Warning: 1 time series (0.2%) are shorter than 5 and cannot be predicted by RecursiveTabular. Fallback model Seasona
lNaive is used for these time series.
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
Warning: 1 time series (0.2%) are shorter than 5 and cannot be predicted by RecursiveTabular. Fallback model Seasona
lNaive is used for these time series.
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
```

```
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
Warning: 1 time series (0.2%) are shorter than 5 and cannot be predicted by RecursiveTabular. Fallback model Seasona
lNaive is used for these time series.
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
Warning: 1 time series (0.2%) are shorter than 5 and cannot be predicted by RecursiveTabular. Fallback model Seasona
lNaive is used for these time series.
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
Warning: 1 time series (0.2%) are shorter than 5 and cannot be predicted by RecursiveTabular. Fallback model Seasona
lNaive is used for these time series.
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
```

```
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
```

```
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
```

```
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
```

```
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
```

```
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
```

```
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
```

```
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
```

```
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
```

```
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
```

```
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
Warning: 1 time series (0.2%) are shorter than 5 and cannot be predicted by RecursiveTabular. Fallback model Seasona
lNaive is used for these time series.
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
Warning: 1 time series (0.2%) are shorter than 5 and cannot be predicted by RecursiveTabular. Fallback model Seasona
lNaive is used for these time series.
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
Warning: 1 time series (0.2%) are shorter than 5 and cannot be predicted by RecursiveTabular. Fallback model Seasona
lNaive is used for these time series.
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
Warning: 1 time series (0.2%) are shorter than 5 and cannot be predicted by RecursiveTabular. Fallback model Seasona
lNaive is used for these time series.
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
```

```
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
Warning: 1 time series (0.2%) are shorter than 5 and cannot be predicted by RecursiveTabular. Fallback model Seasona
lNaive is used for these time series.
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
```

```
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
Warning: 1 time series (0.2%) are shorter than 5 and cannot be predicted by RecursiveTabular. Fallback model Seasona
lNaive is used for these time series.
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
Warning: 1 time series (0.2%) are shorter than 5 and cannot be predicted by RecursiveTabular. Fallback model Seasona
lNaive is used for these time series.
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
Warning: 1 time series (0.2%) are shorter than 5 and cannot be predicted by RecursiveTabular. Fallback model Seasona
lNaive is used for these time series.
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
```

```
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
Warning: 1 time series (0.2%) are shorter than 5 and cannot be predicted by RecursiveTabular. Fallback model Seasona
lNaive is used for these time series.
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
Warning: 1 time series (0.2%) are shorter than 5 and cannot be predicted by RecursiveTabular. Fallback model Seasona
lNaive is used for these time series.
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
Warning: 1 time series (0.2%) are shorter than 5 and cannot be predicted by RecursiveTabular. Fallback model Seasona
lNaive is used for these time series.
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
```

```
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
```

```
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
```

```
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
```

```
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
```

```
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
```

```
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
Warning: 1 time series (0.2%) are shorter than 5 and cannot be predicted by RecursiveTabular. Fallback model Seasona
lNaive is used for these time series.
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
```

```
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
Warning: 1 time series (0.2%) are shorter than 5 and cannot be predicted by RecursiveTabular. Fallback model Seasona
lNaive is used for these time series.
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
Warning: 1 time series (0.2%) are shorter than 5 and cannot be predicted by RecursiveTabular. Fallback model Seasona
lNaive is used for these time series.
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
Warning: 1 time series (0.2%) are shorter than 5 and cannot be predicted by RecursiveTabular. Fallback model Seasona
lNaive is used for these time series.
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
```

```
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
Warning: 1 time series (0.2%) are shorter than 5 and cannot be predicted by RecursiveTabular. Fallback model Seasona
lNaive is used for these time series.
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
```

```
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
```

```
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
```

```
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
```

```
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
```

```
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
```

```
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
```

```
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
```

```
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
```

```
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
```

```
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
```

```
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
```

```
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
```

```
Warning: 1 time series (0.2%) are shorter than 5 and cannot be predicted by RecursiveTabular. Fallback model Seasona
lNaive is used for these time series.
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
Warning: 1 time series (0.2%) are shorter than 5 and cannot be predicted by RecursiveTabular. Fallback model Seasona
lNaive is used for these time series.
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
Warning: 1 time series (0.2%) are shorter than 5 and cannot be predicted by RecursiveTabular. Fallback model Seasona
lNaive is used for these time series.
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
Warning: 1 time series (0.2%) are shorter than 5 and cannot be predicted by RecursiveTabular. Fallback model Seasona
lNaive is used for these time series.
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
Warning: 1 time series (0.2%) are shorter than 5 and cannot be predicted by RecursiveTabular. Fallback model Seasona
lNaive is used for these time series.
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
Warning: 1 time series (0.2%) are shorter than 5 and cannot be predicted by RecursiveTabular. Fallback model Seasona
lNaive is used for these time series.
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
```

```
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
```

```
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
```

```
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
```

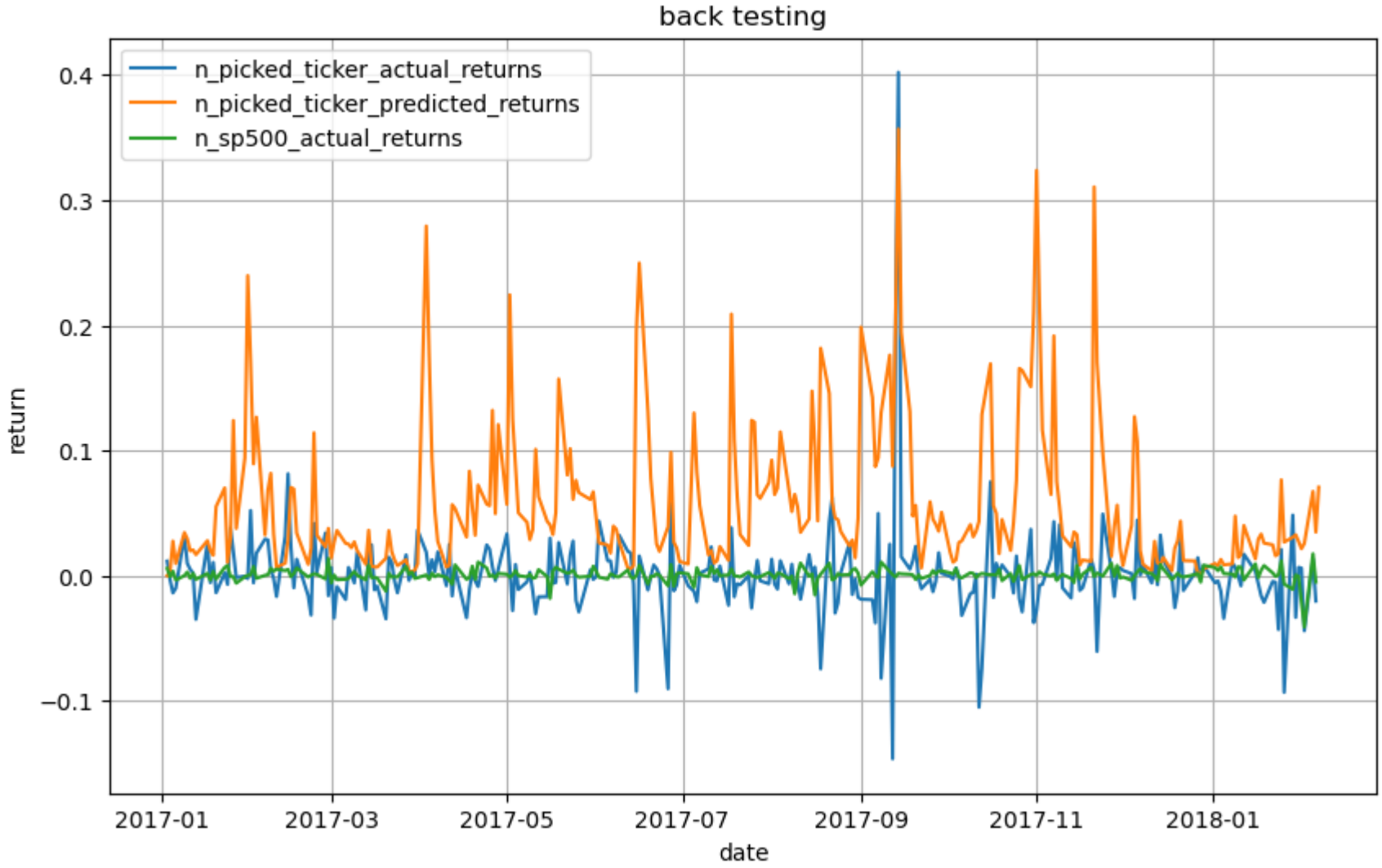
```
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
```

```
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
```

```
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
```

```
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
data with frequency 'None' has been resampled to frequency 'B'.
Model not specified in predict, will default to the model with the best validation score: WeightedEnsemble
/tmp/ipykernel_385478/92941470.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy
  two_latest_rows_only['predicted_future_return'] = time_series.calculate_future_return_series(two_latest_rows_only
['target'])
```

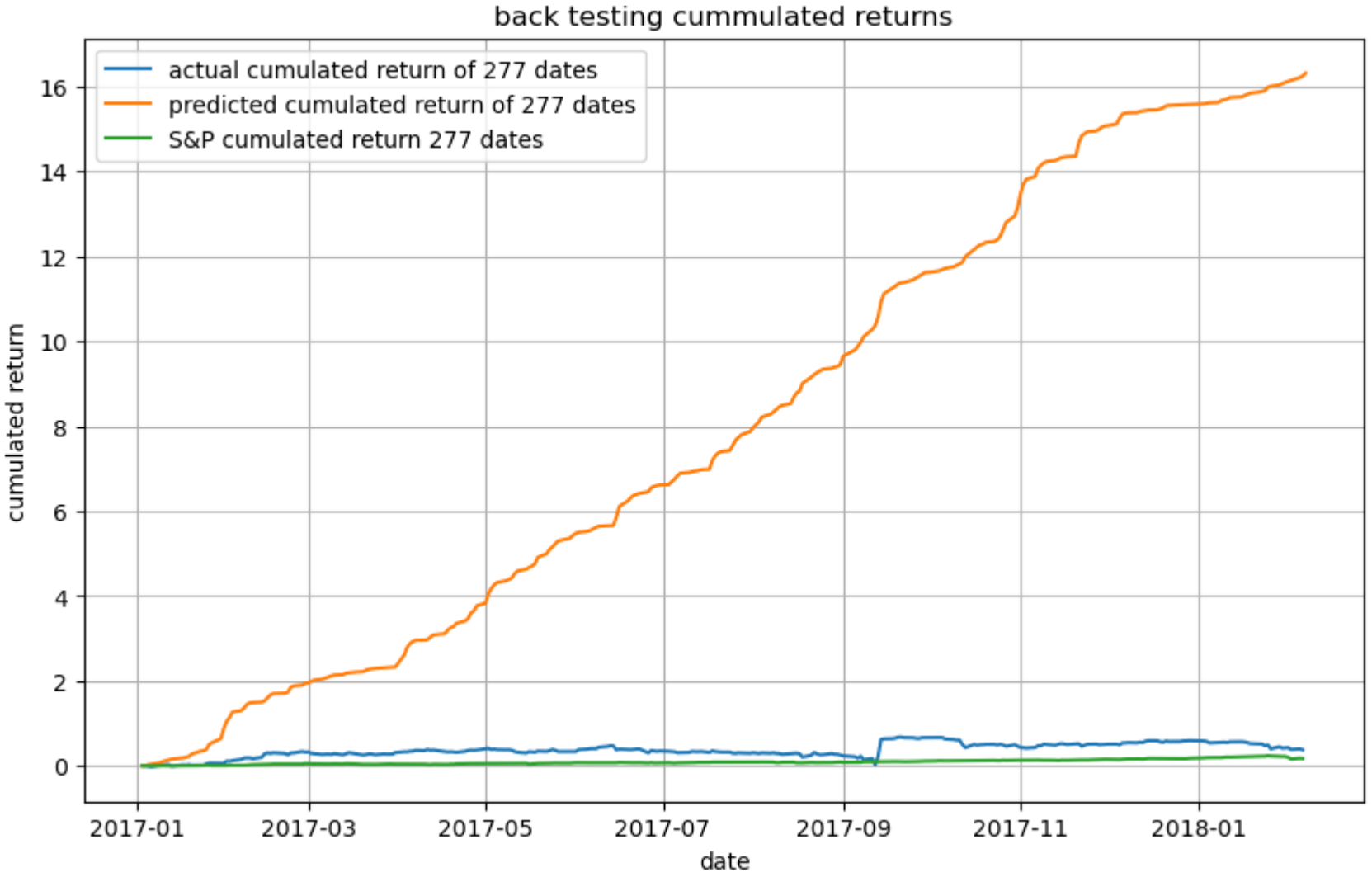
In [63]:
```
strat.plot_multiple_series(
    x=backtesting_dates,
    y_series_list=[
        n_picked_ticker_actual_returns,
        n_picked_ticker_best_predicted_returns,
        n_sp500_actual_returns,
    ],
    labels=[
        'n_picked_ticker_actual_returns',
        'n_picked_ticker_predicted_returns',
        'n_sp500_actual_returns',
    ],
    title='back testing',
    xlabel='date',
    ylabel='return',
    folder_path=constant.graph_folder_path,
    file_name='back_testing_return.png',
)
```




```
<Figure size 640x480 with 0 Axes>
```

In [64]:
```
strat.plot_multiple_series(
    x=backtesting_dates,
    y_series_list=[
        numpy.cumsum(n_picked_ticker_actual_returns),
        numpy.cumsum(n_picked_ticker_best_predicted_returns),
        numpy.cumsum(n_sp500_actual_returns),
```

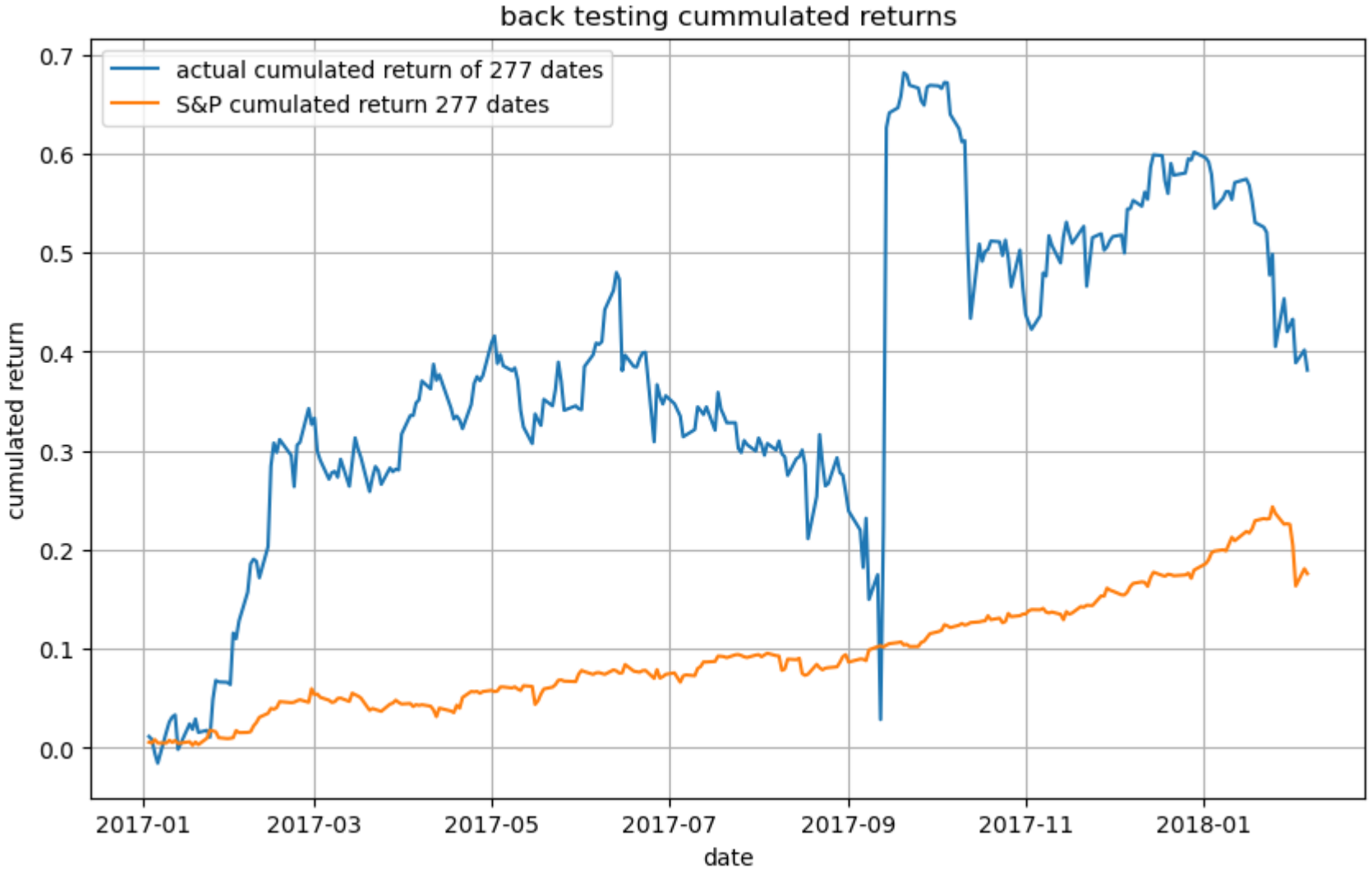
```
    ],
    labels=[
        f"actual cumulated return of {number_of_dates} dates",
        f"predicted cumulated return of {number_of_dates} dates",
        f"S&P cumulated return {number_of_dates} dates",
    ],
    title='back testing cummulated returns',
    xlabel='date',
    ylabel='cumulated return',
    folder_path=constant.graph_folder_path,
    file_name='back_testing_cumulated_return.png',
)
```




```
<Figure size 640x480 with 0 Axes>
```

In [69]:
```
strat.plot_multiple_series(
    x=backtesting_dates,
    y_series_list=[
        numpy.cumsum(n_picked_ticker_actual_returns),
        numpy.cumsum(n_sp500_actual_returns),
    ],
```

```
    labels=[
        f"actual cumulated return of {number_of_dates} dates",
        f"S&P cumulated return {number_of_dates} dates",
    ],
    title='back testing cummulated returns',
    xlabel='date',
    ylabel='cumulated return',
    folder_path=constant.graph_folder_path,
    file_name='back_testing_cumulated_return.png',
)
```




```
<Figure size 640x480 with 0 Axes>
```

In [65]:
```
print(f"how much we could earn (average return ) by using the model:\n\t{int(10000*numpy.mean(n_picked_ticker_actua
print(f"compared to simply betting on S&P:\n\t{int(10000*numpy.mean(n_sp500_actual_returns[:-1]))/100}%")
print(f"weither you check the cumulated sum or the averag or the sum or cumulated product depend on your investment
print(f"\t reinvesting all you earned/lost the next time or a fixed amount of money")
```

```
how much we could earn (average return ) by using the model:
        0.13%
compared to simply betting on S&P:
        0.06%
weither you check the cumulated sum or the averag or the sum or cumulated product depend on your investment plan:
         reinvesting all you earned/lost the next time or a fixed amount of money
```

note

in practice, it seems that you would not train over something that is x years old. you would more likely have your model continuously retrained every x days thanks to new daily data for example:

- the training window would be the all you can, up to last week
- and every week you have a routine that train again the model based on new fresh data
- and if your investement based on the prediction start to be below a defined expectation; you would either retrain the model automatically until it meets expectations again or request human intervention

report

the prediction model is an complex aggregation of different models
the logs below have the information, some more exhaustive data are in the repertory '../model/autogluon...'

In [66]:
```
new_predictor.fit_summary()
```

```
****************** Summary of fit() ******************
Estimated performance of each model:
                      model  score_val  pred_time_val  fit_time_marginal  \
0          WeightedEnsemble  -0.287953       1.252297           0.167924
1  TemporalFusionTransformer  -0.298310       0.932168        1937.939438
2          RecursiveTabular  -0.446931       0.320129           8.860688
3                     Naive  -0.512100       1.360595           0.019259
4                       ETS  -0.520564      98.163364           0.067728
5                     Theta  -0.557930      11.947530           0.067403
6             SeasonalNaive  -0.613604       0.253551           0.019232
7             DirectTabular  -0.626914       0.579308           6.096982

   fit_order
0          8
1          7
2          3
3          1
4          5
5          6
6          2
7          4
Number of models trained: 8
Types of models trained:
{'ETSModel', 'TimeSeriesGreedyEnsemble', 'DirectTabularModel', 'RecursiveTabularModel', 'TemporalFusionTransformerMo
del', 'SeasonalNaiveModel', 'ThetaModel', 'NaiveModel'}
****************** End of fit() summary ******************
```

Out[66]:

```
{'model_types': {'Naive': 'NaiveModel',
  'SeasonalNaive': 'SeasonalNaiveModel',
  'RecursiveTabular': 'RecursiveTabularModel',
  'DirectTabular': 'DirectTabularModel',
  'ETS': 'ETSModel',
  'Theta': 'ThetaModel',
  'TemporalFusionTransformer': 'TemporalFusionTransformerModel',
  'WeightedEnsemble': 'TimeSeriesGreedyEnsemble'},
 'model_performance': {'Naive': -0.5120998961978354,
  'SeasonalNaive': -0.6136036021837181,
  'RecursiveTabular': -0.44693111726921936,
  'DirectTabular': -0.6269135710515658,
  'ETS': -0.5205641999909835,
  'Theta': -0.5579296688077651,
  'TemporalFusionTransformer': -0.2983095911174175,
  'WeightedEnsemble': -0.2879526147193745},
 'model_best': 'WeightedEnsemble',
 'model_paths': {'Naive': ['Naive'],
  'SeasonalNaive': ['SeasonalNaive'],
  'RecursiveTabular': ['RecursiveTabular'],
  'DirectTabular': ['DirectTabular'],
  'ETS': ['ETS'],
  'Theta': ['Theta'],
  'TemporalFusionTransformer': ['TemporalFusionTransformer'],
  'WeightedEnsemble': ['WeightedEnsemble']},
 'model_fit_times': {'Naive': 0.019258737564086914,
  'SeasonalNaive': 0.01923227310180664,
  'RecursiveTabular': 8.86068844795227,
  'DirectTabular': 6.096981763839722,
  'ETS': 0.06772780418395996,
  'Theta': 0.06740283966064453,
  'TemporalFusionTransformer': 1937.9394381046295,
  'WeightedEnsemble': 0.167924165725708},
 'model_pred_times': {'Naive': 1.360595464706421,
  'SeasonalNaive': 0.25355076789855957,
  'RecursiveTabular': 0.3201289176940918,
  'DirectTabular': 0.579308032989502,
  'ETS': 98.16336441040039,
  'Theta': 11.947529554367065,
  'TemporalFusionTransformer': 0.9321684837341309,
```

```
 'WeightedEnsemble': 1.2522974014282227},
'model_hyperparams': {'Naive': {},
 'SeasonalNaive': {},
 'RecursiveTabular': {},
 'DirectTabular': {},
 'ETS': {},
 'Theta': {},
 'TemporalFusionTransformer': {},
 'WeightedEnsemble': {}},
'leaderboard':                        model  score_val  pred_time_val  fit_time_marginal  \
0           WeightedEnsemble  -0.287953       1.252297           0.167924
1  TemporalFusionTransformer  -0.298310       0.932168        1937.939438
2           RecursiveTabular  -0.446931       0.320129           8.860688
3                      Naive  -0.512100       1.360595           0.019259
4                        ETS  -0.520564      98.163364           0.067728
5                      Theta  -0.557930      11.947530           0.067403
6              SeasonalNaive  -0.613604       0.253551           0.019232
7              DirectTabular  -0.626914       0.579308           6.096982

   fit_order
0          8
1          7
2          3
3          1
4          5
5          6
6          2
7          4  }
```

In [ ]:
```
new_predictor.plot(data=test_time_series_data_frame)
```

In [71]:
```
del new_predictor
```

In [72]:
```
del best_predictor
```

In [70]:
```
del info
```

In [ ]: