Forecasting Process (with External Regressors)

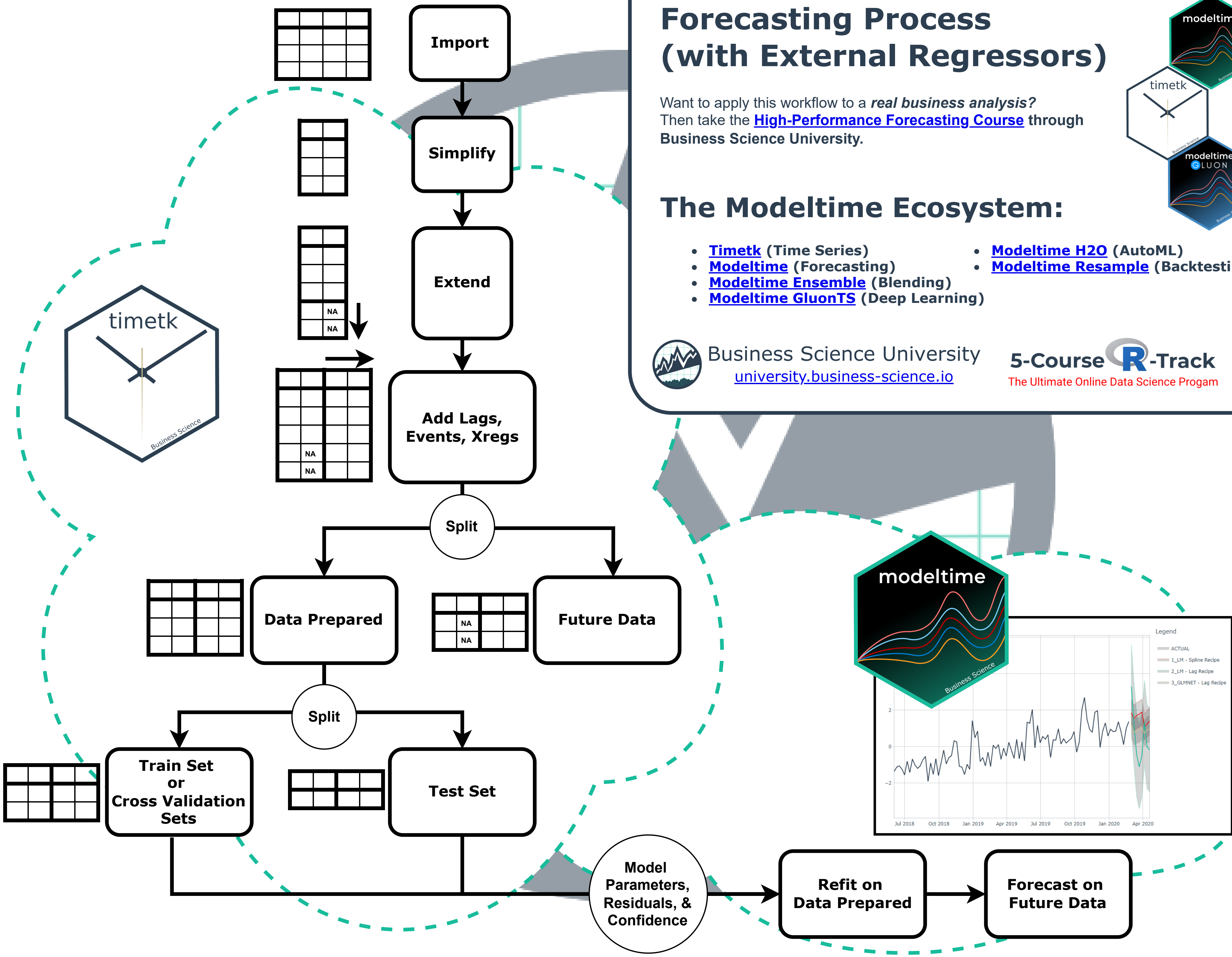
Want to apply this workflow to a ***real business analysis?***
Then take the **High-Performance Forecasting Course through Business Science University.**

The Modeltime Ecosystem:

- **Timetk (Time Series)**
- **Modeltime (Forecasting)**
- **Modeltime Ensemble (Blending)**
- **Modeltime GluonTS (Deep Learning)**
- **Modeltime H2O (AutoML)**
- **Modeltime Resample (Backtesting)**

Business Science University
university.business-science.io

5-Course R-Track
The Ultimate Online Data Science Progam



MODELTIME Workflow

Create Modeltime Table

`modeltime_table()`

```
# Modeltime Table
# A tibble: 5 x 3
  .model_id .model     .model_desc
      <int> <list>     <chr>
1         1 <fit[+]> ARIMA(0,1,1)(0,1,1)[12]
2         2 <fit[+]> ARIMA(0,1,1)(0,1,1)[12] W/ XGBOOST ERRORS
3         3 <fit[+]> ETS(M,A,A)
4         4 <fit[+]> LM
5         5 <fit[+]> EARTH
```

Calibrate

`modeltime_calibrate()`

```
# Modeltime Table
# A tibble: 6 x 5
  .model_id .model     .model_desc                               .type .calibration_data
      <int> <list>     <chr>                                     <chr> <list>
1         1 <fit[+]> ARIMA(0,1,1)(0,1,1)[12]                   Test  <tibble [31 × 4]>
2         2 <fit[+]> ARIMA(0,1,1)(0,1,1)[12] W/ XGBOOST ERRORS Test  <tibble [31 × 4]>
3         3 <fit[+]> ETS(M,A,A)                                Test  <tibble [31 × 4]>
4         4 <fit[+]> PROPHET                                   Test  <tibble [31 × 4]>
5         5 <fit[+]> LM                                        Test  <tibble [31 × 4]>
6         6 <fit[+]> EARTH                                     Test  <tibble [31 × 4]>
```

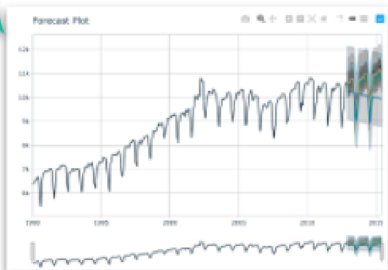
Refit

`modeltime_refit()`

```
# Modeltime Table
# A tibble: 6 x 5
  .model_id .model     .model_desc                               .type .calibration_data
      <int> <list>     <chr>                                     <chr> <list>
1         1 <fit[+]> ARIMA(0,1,1)(0,1,1)[12]                   Test  <tibble [31 × 4]>
2         2 <fit[+]> ARIMA(0,1,1)(0,1,1)[12] W/ XGBOOST ERRORS Test  <tibble [31 × 4]>
3         3 <fit[+]> ETS(A,A,A)                                Test  <tibble [31 × 4]>
4         4 <fit[+]> PROPHET                                   Test  <tibble [31 × 4]>
5         5 <fit[+]> LM                                        Test  <tibble [31 × 4]>
6         6 <fit[+]> EARTH                                     Test  <tibble [31 × 4]>
```

Forecast Test Set

`modeltime_forecast()`



`plot_modeltime_forecast()`

Test Accuracy

`modeltime_accuracy()`

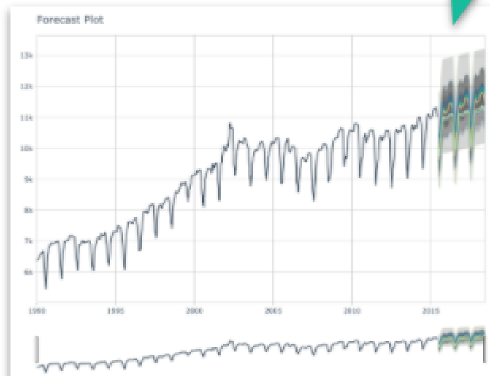
Accuracy Table

.model_id	.model_desc	.type	mae	mape	mase	smape	rmse	rsq
1	ARIMA(0,1,1)(0,1,1)[12]	Test	151.33	1.41	0.52	1.43	197.71	0.93
2	ARIMA(0,1,1)(0,1,1)[12] W/ XGBOOST ERRORS	Test	147.04	1.37	0.50	1.39	191.84	0.93
3	ETS(M,A,A)	Test	77.00	0.73	0.26	0.73	90.27	0.98
4	PROPHET	Test	177.51	1.70	0.61	1.70	234.65	0.88
5	LM	Test	629.12	6.01	2.15	5.81	657.19	0.91
6	EARTH	Test	709.83	6.59	2.42	6.86	782.82	0.55

`table_modeltime_accuracy()`

Forecast Future

`modeltime_forecast()`



`plot_modeltime_forecast()`

Resources:

- This workflow is covered in depth in the [High-Performance Time Series Course (Modules 7 to 18)]
- A beginner tutorial is available here: [Getting Started with Modeltime]
- The ["Global Forecasting Workflow"] is used to create scalable high-performance model(s) that forecast many time series
- This is NOT the "Iterative Forecasting Process" (Nested Workflow), which is used to forecast iteratively. The Iterative Forecasting Procedure is shown on Page 5.

modeltime

START
Cross Validation

1. Cross Validation Plan

Can be used for more than 1 model

Sequential Time Series CV
`time_series_cv()`
Ex. Arima Boost

Non-Sequential 10-Fold CV
`vfold_cv()`
Ex. Prophet Boost

2. Identify Tuning Parameters

Specific to Each Model

Model Spec & Recipe Identify Parameters
`tune(id)`

3. Make Grid for Parameters

Specific to Each Model

Grid Spec

```
grid_latin_hypercube(
  param_set
)
```

Tune Grid

```
tune_grid(
  resamples,
  grid,
  metrics,
  control
)
```

4. Hyperparameter Tune

Specific to Each Model

Use Parallel Processing to Speed Up

Review Parameters Adjust & Re-Tune (if necessary)
`autoplot()`

Model Parameters, Residuals, & Confidence

Split

Cross Validation

Test

Model Parameters, Residuals, & Confidence

Hyperparameter Tuning & Cross-Validation for Time Series

Cross-Validation and Hyperparameter Tuning are absolutely critical for Machine Learning algorithms.

Additional Resources:

- [The NEW Hyperparameter Tuning & Parallel Processing Process]

MODELTIME ENSEMBLE

Multi-Level Stacking

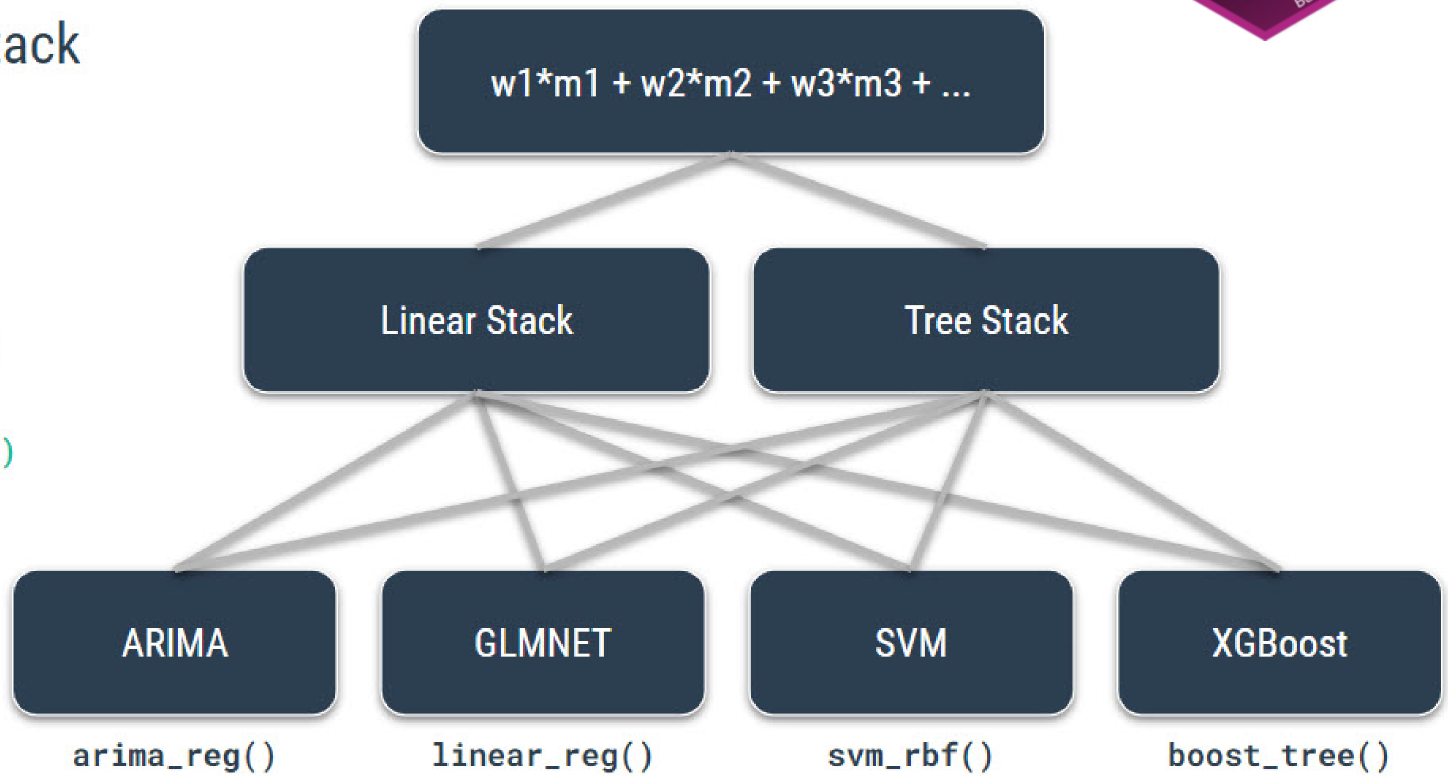
Level 3: Weighted Stack

`ensemble_weighted()`
`ensemble_average()`

Level 2:
Stacking Algorithms

`ensemble_model_spec()`
`modeltime_fit_resamples()`

Level 1:
Sub-Models



Additional Resources:

- The Modeltime Ensemble process is covered in-depth in the **[High-Performance Time Series Course Module 14]()**
- The **[Lost Time Series Module 5]()** shows how to use Ensembles to improve *Hierarchical Forecasting Performance*
- Additional Resource: **[Getting Started with Modeltime Ensemble]()**

Iterative Forecasting (Nested Workflow)

Used to make individual models for many time series. Can be more accurate than Global Forecasting, but is less scalable.

- This process is covered in-depth in the [Lost Time Series Modules (Module 2 - Iterative Forecasting at Scale) and (Module 3 - Recursive Iterative Forecasting)]
- Additional Resource: [Getting Started with Nested Forecasting].

Nested Time Series Data

`extend_timeseries()`
`nest_timeseries()`
`split_nested_timeseries()`

```
# A tibble: 7 × 5
  id    .actual_data       .future_data      .splits
  <fct> <list>             <list>            <list>
1 1_1   <tibble [104 × 2]> <tibble [52 × 2]> <split [52/52]>
2 1_3   <tibble [104 × 2]> <tibble [52 × 2]> <split [52/52]>
3 1_8   <tibble [104 × 2]> <tibble [52 × 2]> <split [52/52]>
4 1_13  <tibble [104 × 2]> <tibble [52 × 2]> <split [52/52]>
5 1_38  <tibble [104 × 2]> <tibble [52 × 2]> <split [52/52]>
6 1_93  <tibble [104 × 2]> <tibble [52 × 2]> <split [52/52]>
7 1_95  <tibble [104 × 2]> <tibble [52 × 2]> <split [52/52]>
```

Many Models

Model	Function
ARIMA	`arima_reg()`
PROPHET	`prophet_reg()`
ETS	`exp_smoothing()`
Random Forest	`rand_forest()`
SVM	`svm_rbf()`
XGBoost	`boost_tree()`

`modeltime_nested_fit()`

Nested Modeltime Table

```
# Nested Modeltime Table
  Trained on: .splits | Model Errors: [0]
# A tibble: 7 × 5
  id    .actual_data       .future_data      .splits         .modeltime_tables
  <fct> <list>             <list>            <list>          <list>
1 1_1   <tibble [104 × 2]> <tibble [52 × 2]> <split [52/52]> <mdl_time_tbl [2 × 5]>
2 1_3   <tibble [104 × 2]> <tibble [52 × 2]> <split [52/52]> <mdl_time_tbl [2 × 5]>
3 1_8   <tibble [104 × 2]> <tibble [52 × 2]> <split [52/52]> <mdl_time_tbl [2 × 5]>
4 1_13  <tibble [104 × 2]> <tibble [52 × 2]> <split [52/52]> <mdl_time_tbl [2 × 5]>
5 1_38  <tibble [104 × 2]> <tibble [52 × 2]> <split [52/52]> <mdl_time_tbl [2 × 5]>
6 1_93  <tibble [104 × 2]> <tibble [52 × 2]> <split [52/52]> <mdl_time_tbl [2 × 5]>
7 1_95  <tibble [104 × 2]> <tibble [52 × 2]> <split [52/52]> <mdl_time_tbl [2 × 5]>
```

Core Functions | Nested Forecasting

1: Nested Fitting

`modeltime_nested_fit()`

- **Trains** each model on training split
- **Logs** test accuracy, test forecast with confidence intervals on testing split
- **Logs** additional information including error reports

2: Select Best

`modeltime_nested_select_best()`

- **Selects best model** using accuracy metric
- **Filters** test forecasts to just those of best models
- **Logs** best models

3: Nested Refitting

`modeltime_nested_refit()`

- **Retrains** selected models on actual data
- **Logs** future forecast on future data

Extracting Nested Modeltime Table Logs

Nested Modeltime Table

```
# Nested Modeltime Table
  Trained on: .actual_data | Model Errors: [0]
# A tibble: 7 × 5
  id    .actual_data       .future_data      .splits         .modeltime_tables
  <fct> <list>             <list>            <list>          <list>
1 1_1   <tibble [104 × 2]> <tibble [52 × 2]> <split [52/52]> <mdl_time_tbl [1 × 5]>
2 1_3   <tibble [104 × 2]> <tibble [52 × 2]> <split [52/52]> <mdl_time_tbl [1 × 5]>
3 1_8   <tibble [104 × 2]> <tibble [52 × 2]> <split [52/52]> <mdl_time_tbl [1 × 5]>
4 1_13  <tibble [104 × 2]> <tibble [52 × 2]> <split [52/52]> <mdl_time_tbl [1 × 5]>
5 1_38  <tibble [104 × 2]> <tibble [52 × 2]> <split [52/52]> <mdl_time_tbl [1 × 5]>
6 1_93  <tibble [104 × 2]> <tibble [52 × 2]> <split [52/52]> <mdl_time_tbl [1 × 5]>
7 1_95  <tibble [104 × 2]> <tibble [52 × 2]> <split [52/52]> <mdl_time_tbl [1 × 5]>
```

Contains Logged Attributes for fast extraction

Test Accuracy

`extract_nested_test_accuracy()`

Accuracy Table

id	.model_id	.model_desc	.type	mae	mape	mase	smape	rmse	rsq
1_1	1	PROPHET	Test	10071.47	45.88	1.89	59.97	11776.87	0.38
1_1	2	XGBOOST	Test	6236.79	26.31	1.23	24.57	9017.22	0.19
1_3	1	PROPHET	Test	3539.80	20.87	1.37	25.46	4707.77	0.80
1_3	2	XGBOOST	Test	3085.78	18.81	1.20	20.40	5085.81	0.79
1_8	1	PROPHET	Test	4282.98	11.15	1.82	11.96	4845.08	0.00
1_8	2	XGBOOST	Test	3585.77	9.33	1.53	9.80	4003.47	0.30
1_13	1	PROPHET	Test	6861.13	17.02	2.53	18.78	7309.61	0.15
1_13	2	XGBOOST	Test	2338.42	5.83	0.86	6.02	2721.47	0.54
1_38	1	PROPHET	Test	20007.21	32.57	2.22	39.60	27938.83	0.08
1_38	2	XGBOOST	Test	6847.04	8.47	0.68	8.75	8825.28	0.40
1_93	1	PROPHET	Test	17165.30	21.37	1.73	24.46	19123.17	0.03
1_93	2	XGBOOST	Test	7333.95	9.11	0.73	9.66	8879.21	0.49
1_95	1	PROPHET	Test	22836.06	18.30	2.76	20.37	24094.49	0.48
1_95	2	XGBOOST	Test	10789.75	8.54	1.30	8.96	12843.50	0.14

Test Forecast

`extract_nested_test_forecast()`

Error Reporting

`extract_nested_error_report()`

```
# A tibble: 4 × 4
  id    .model_id .model_desc .error_desc
  <fct>     <int> <chr>       <chr>
1 1_1           2 BOOST_TREE  "'data' has class 'character' and length 52.\n  'data…
2 1_3           2 BOOST_TREE  "'data' has class 'character' and length 52.\n  'data…
3 1_8           1 BOOST_TREE  "`x` should be an `rsplit` object"
4 1_8           2 BOOST_TREE  "`x` should be an `rsplit` object"
```