

# Time Series as a First-Class Citizen in Python

**Dr Markus Löning**

*Data scientist and sktime contributor*

@sktime\_toolbox



main ▾

47 branches

15 tags

Go to file

Add file ▾

Code ▾



TonyBagnall and MatthewMiddlehurst Up... ... ✓ 3a3b009 2 days ago 2,082 commits



.binder

1478 statsmodels max\_requirements (#1479)

20 days ago



.github

Fix macOS CI (#1511)

8 days ago



build\_tools

1478 statsmodels max\_requirements (#1479)

20 days ago



docs

[DOC] roles overview: added martin to cc (#1532)

6 days ago



examples

Add UnobservedComponents statsmodels wrappe...

last month



extension\_templates

Update classification.py (#1525)

2 days ago



sktime

Update classification.py (#1525)

2 days ago



.all-contributorsrc

[BUG] #1469: stripping names of index X and y (#...

14 days ago



.appveyor.yml

Fix appveyor (#1541)

2 days ago

## About



A unified framework for machine learning with time series

[sktime.org](https://sktime.org)

data-science

machine-learning

data-mining

time-series

scikit-learn

forecasting

time-series-analysis

time-series-classification

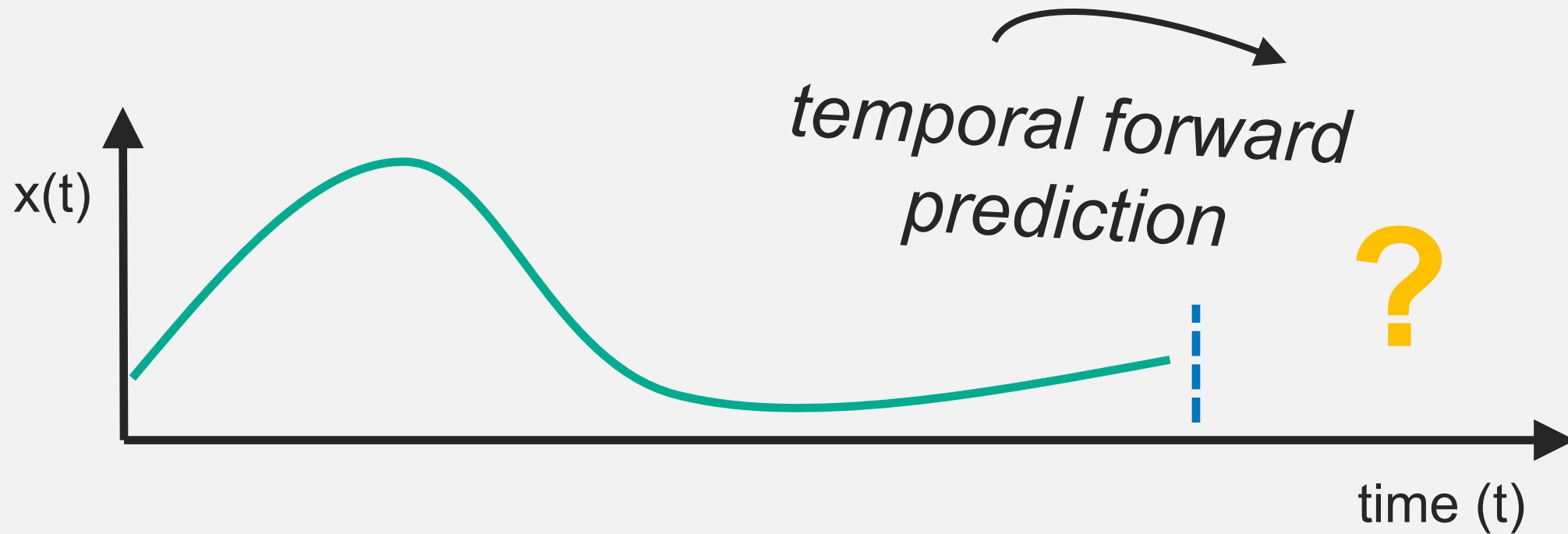
time-series-regression

Readme

BSD-3-Clause License

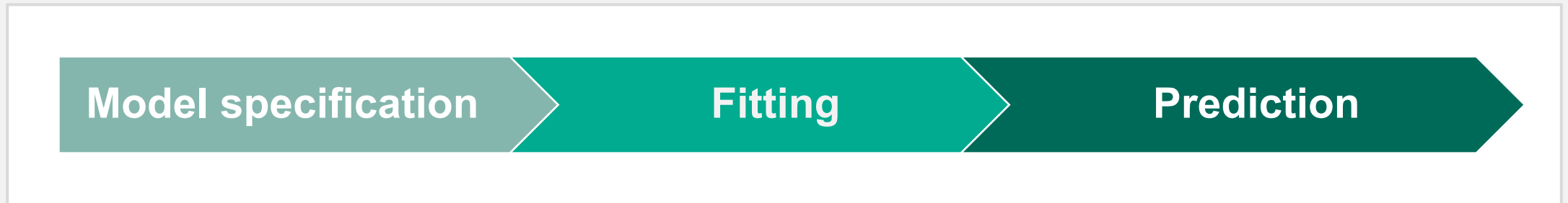
# Machine learning with time series

## Forecasting



# Machine learning with time series

## Typical user workflow



# A fragmented ecosystem

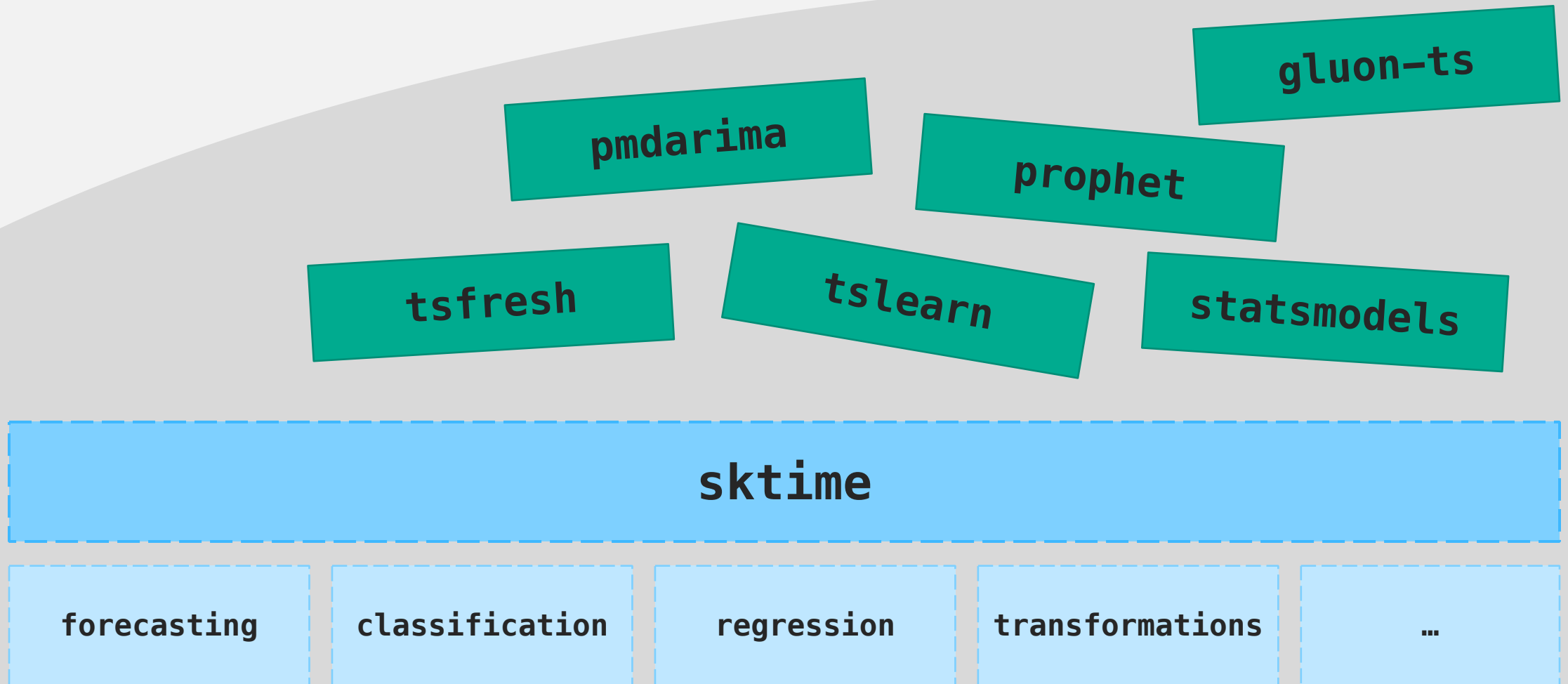
```
# statsmodels  
m = SARIMAX(y)  
mf = m.fit(**params)  
y_hat = mf.predict(start, end)
```

# A fragmented ecosystem

```
# statsmodels  
m = SARIMAX(y)  
mf = m.fit(**params)  
y_hat = mf.predict(start, end)
```

```
# prophet  
m = Prophet(**params)  
m.fit(df)  
y_hat = m.predict(df)
```

# A unified framework



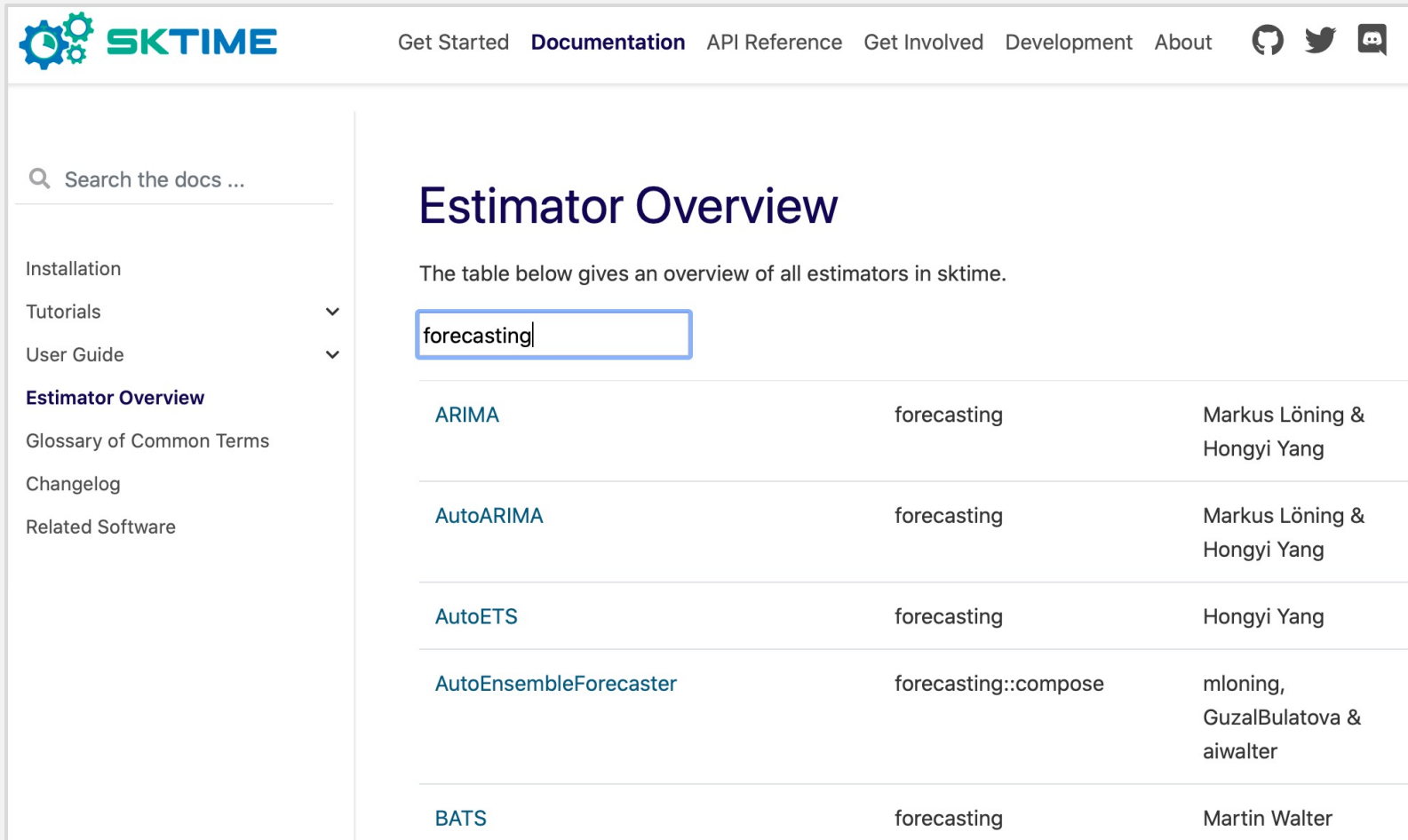
# A unified framework

```
# sktime  
m = ARIMA(**params)  
m.fit(y)  
y_hat = m.predict(fh)
```

```
# sktime  
m = Prophet(**params)  
m.fit(y)  
y_hat = m.predict(fh)
```



# A unified framework

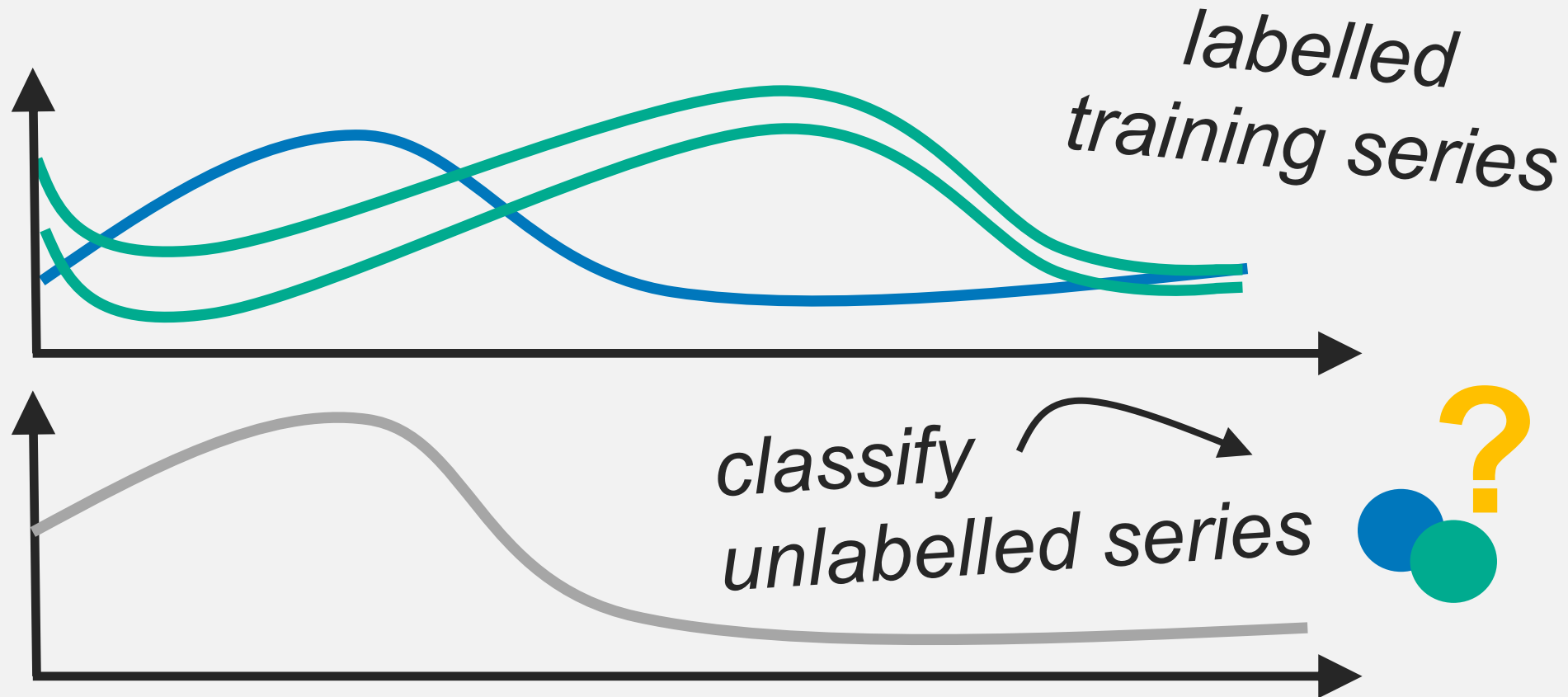


The screenshot shows the SKTime documentation website. The top navigation bar includes links for 'Get Started', 'Documentation' (highlighted), 'API Reference', 'Get Involved', 'Development', and 'About', along with social media icons for GitHub, Twitter, and Discord. The left sidebar contains a search bar and a list of navigation items: 'Installation', 'Tutorials', 'User Guide', 'Estimator Overview' (highlighted), 'Glossary of Common Terms', 'Changelog', and 'Related Software'. The main content area is titled 'Estimator Overview' and contains a table of estimators. A search filter 'forecasting' is applied to the table. The table lists five estimators: ARIMA, AutoARIMA, AutoETS, AutoEnsembleForecaster, and BATS, each with its category and authors.

Estimator	Category	Authors
ARIMA	forecasting	Markus Löning & Hongyi Yang
AutoARIMA	forecasting	Markus Löning & Hongyi Yang
AutoETS	forecasting	Hongyi Yang
AutoEnsembleForecaster	forecasting::compose	mloning, GuzalBulatova & aiwalter
BATS	forecasting	Martin Walter

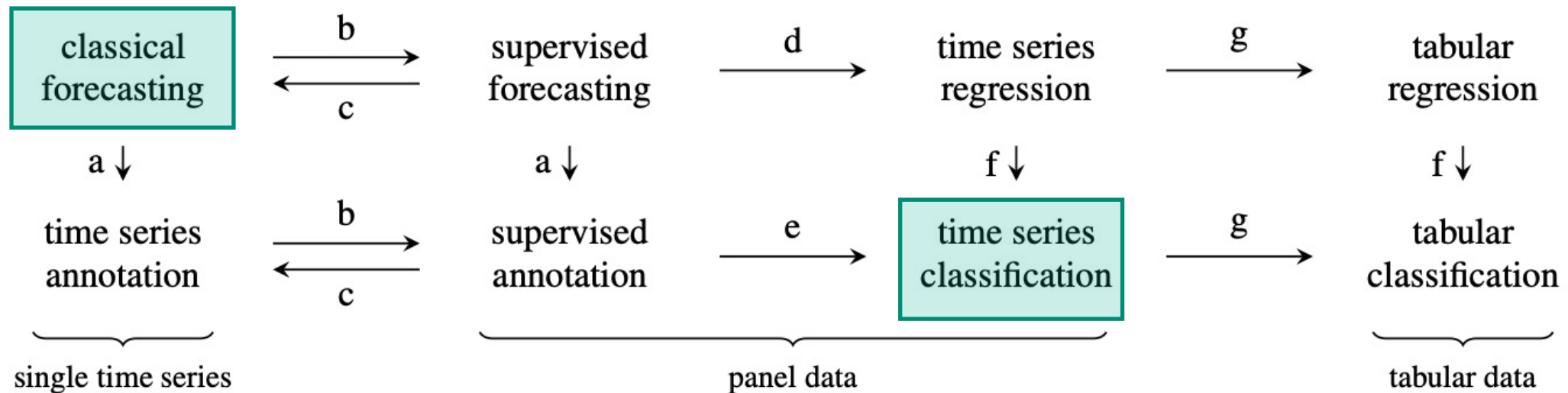
# Machine learning with time series

## Time series classification



# More learning problems ...

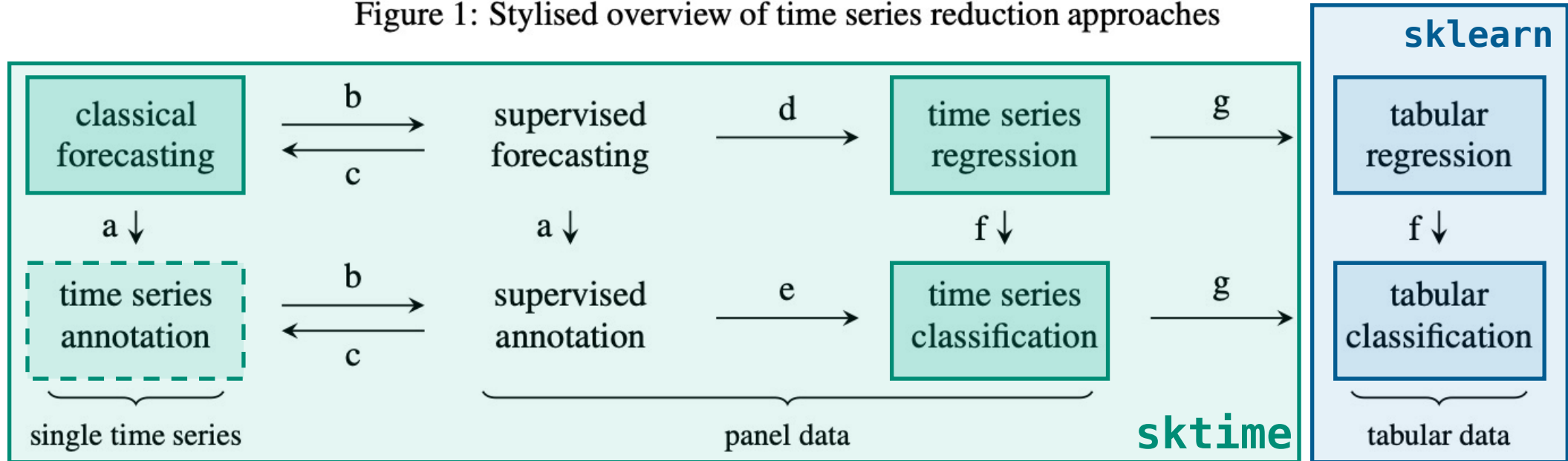
Figure 1: Stylised overview of time series reduction approaches



*Notes:* (a) annotate time series with future values, (b) rolling window method to convert single series into panel data with multiple output time periods [12], (c) ignore training set (e.g. fit forecaster on test set only) or use training set for model selection, (d) iterate over output periods, optionally time binning/aggregation of output periods [12], (e) rolling window method to convert single series into panel data with single output period [23], (f) discretise output into one or more bins, (g) feature extraction [26, 19] or time binning/aggregation of input time points.

# More learning problems ...

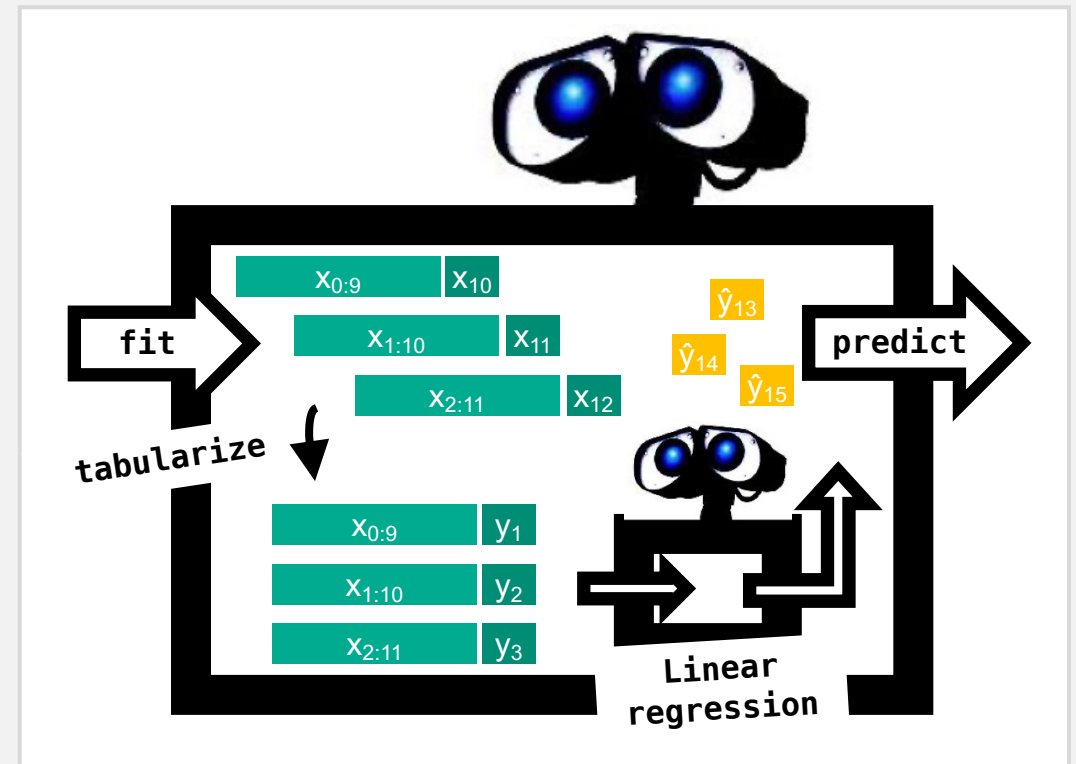
Figure 1: Stylised overview of time series reduction approaches



*Notes:* (a) annotate time series with future values, (b) rolling window method to convert single series into panel data with multiple output time periods [12], (c) ignore training set (e.g. fit forecaster on test set only) or use training set for model selection, (d) iterate over output periods, optionally time binning/aggregation of output periods [12], (e) rolling window method to convert single series into panel data with single output period [23], (f) discretise output into one or more bins, (g) feature extraction [26, 19] or time binning/aggregation of input time points.

# Reduction: from one solution to another

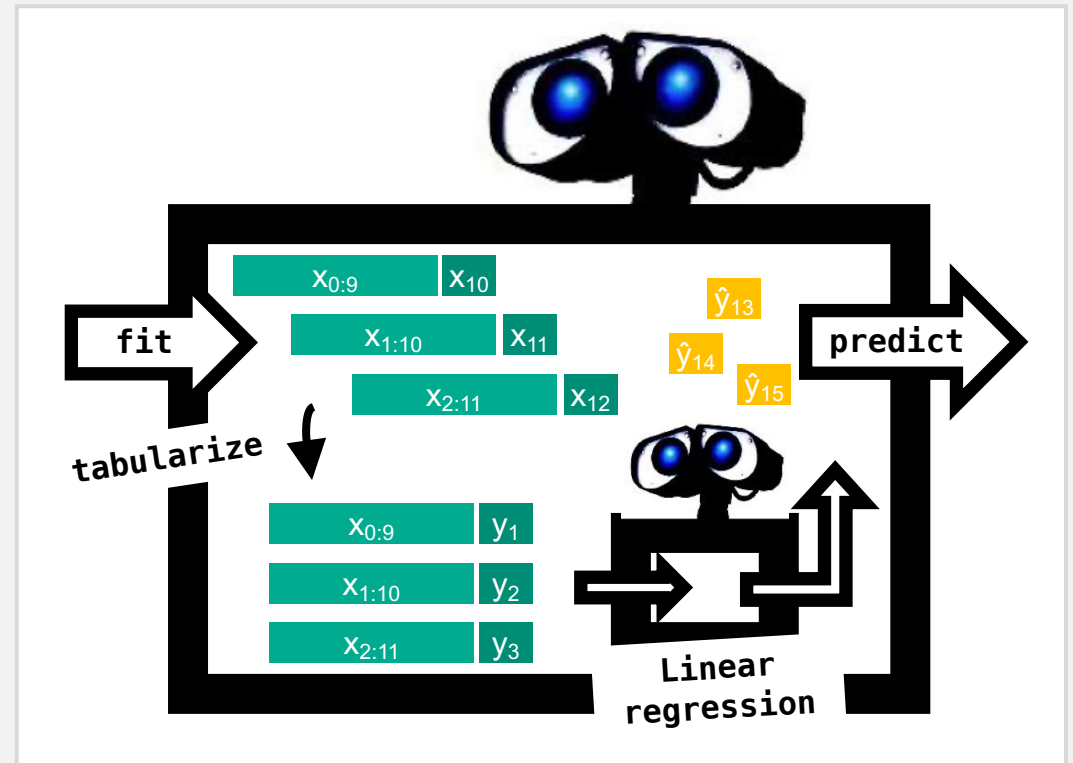
```
# sktime & scikit-learn  
r = LinearRegression(**lparams)  
m = make_reduction(r, **rparams)  
m.fit(y)  
y_pred = m.predict(fh)
```



# Reduction: from one solution to another

```
# sktime & scikit-learn  
r = LinearRegression(**lparams)  
m = make_reduction(r, **rparams)  
m.fit(y)  
y_pred = m.predict(fh)
```

- Modular
- Parametric
- Tuneable
- Adaptive





# Community

- Academia, industry & early-career data scientists
- 100+ contributors
- 13 core developers
- Mentoring
- Internships (Google Summer of Code, Major League Hacking, Outreachy)
- Regular contributor meetups
- Clear governance guidelines

@sktime\_toolbox



# Summary

- Multiple distinct but related learning tasks
- Fragmented Python ecosystem
- sktime – a unified, scikit-learn compatible framework
- Reduction approaches
- Community of practice



# How to find out more

- Website: <https://www.sktime.org>
- GitHub: <https://github.com/alan-turing-institute/sktime>
- Deep learning package: <https://github.com/sktime/sktime-dl>
- Twitter: [@sktime\\_toolbox](https://twitter.com/sktime_toolbox)
- Papers:
  - Markus Löning et al. (2019): “sktime: A Unified Interface for Machine Learning with Time Series”, *Workshop on Systems for ML at NeurIPS*.

**Check out our PyData Global 2021 tutorial!**

<https://github.com/sktime/sktime-tutorial-pydata-global-2021>