Feature engineering for time series forecasting

DataTalks.Club
Aug 2022
Kishan Manani

About me

- Data Science Manager
- PhD Physics: Modelling and large scale time series analysis abnormal heart rhythms
- Open source: github.com/KishManani
- Online course developer, see course: trainindata.com/p/feature-engineering-for-forecasting
- Slides: https://github.com/KishManani/DataTalksClub2022

Kishan Manani, PhD

@KishManani

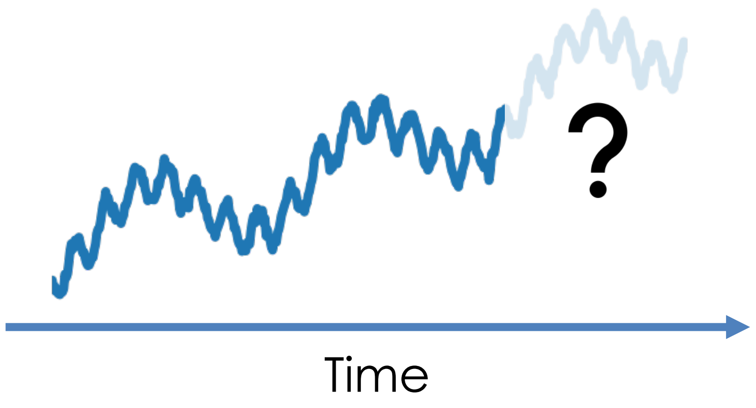
In/kishanmanani

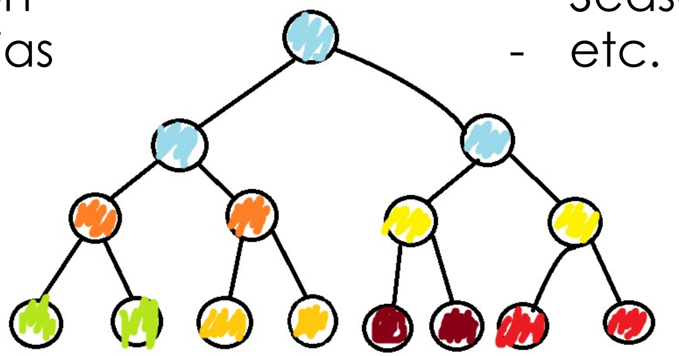
medium.com/@kish.manani

About this talk

Forecasting workflow
- Multi-step forecasting
- Cross-validation
- Look-ahead bias
- etc.

Time series features
- Lags, windows
- Seasonality
- etc.



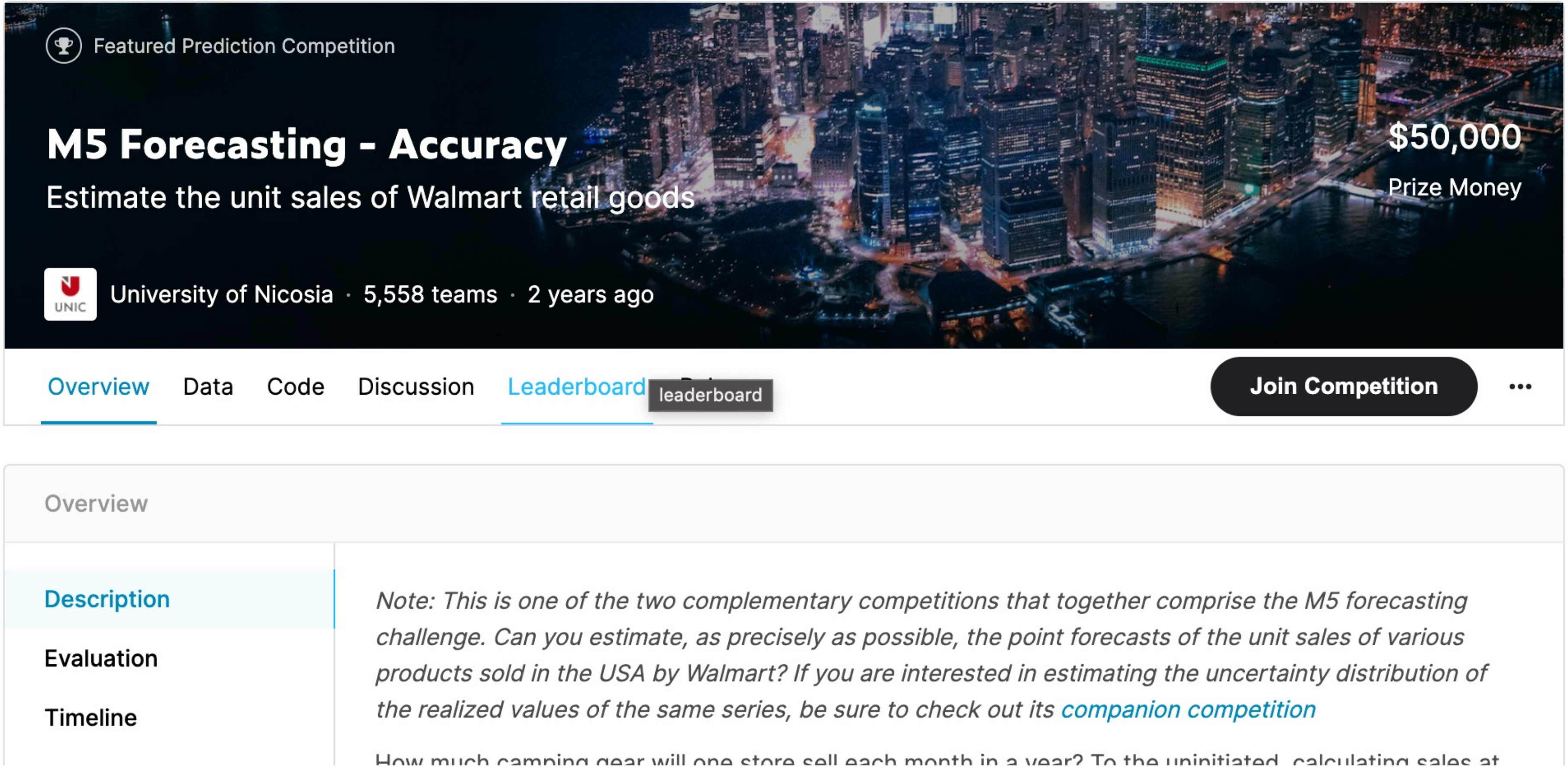


Time	Sales (UK)	Sales (Germany)
2020-02-12	35	21
2020-02-13	30	20
2020-02-14	23	19
2020-02-15	?	?
2020-02-16	?	?

x1	x2	x3	y
			35
			30
			23
			?
			?

Contents

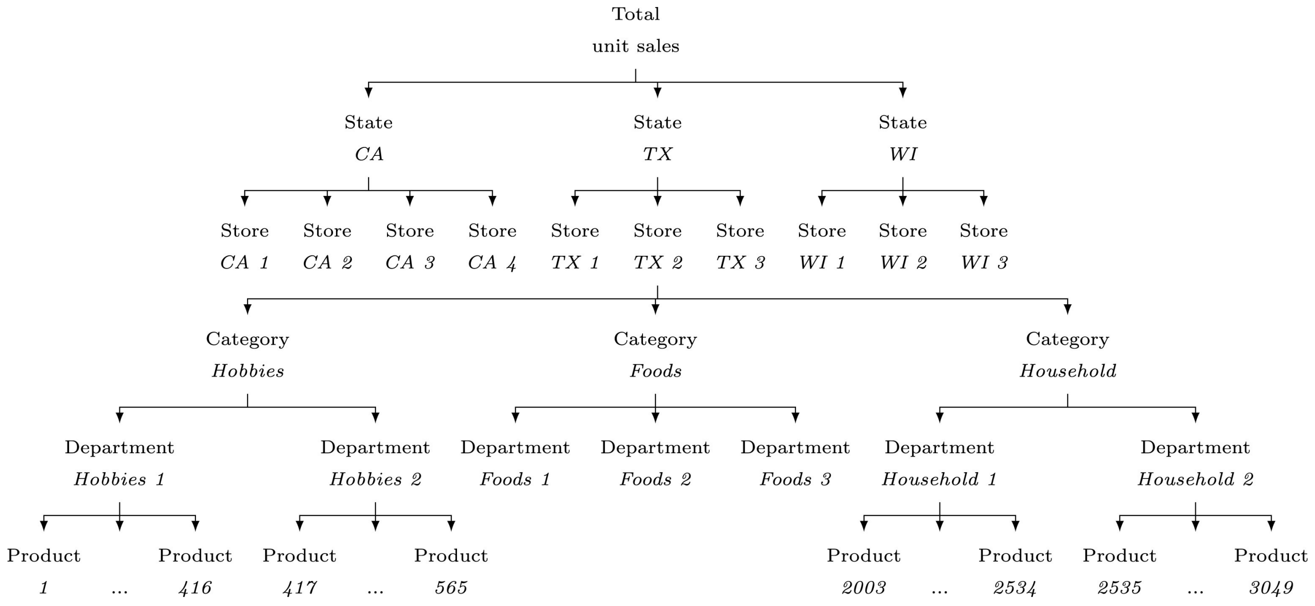
Why use machine learning for forecasting?

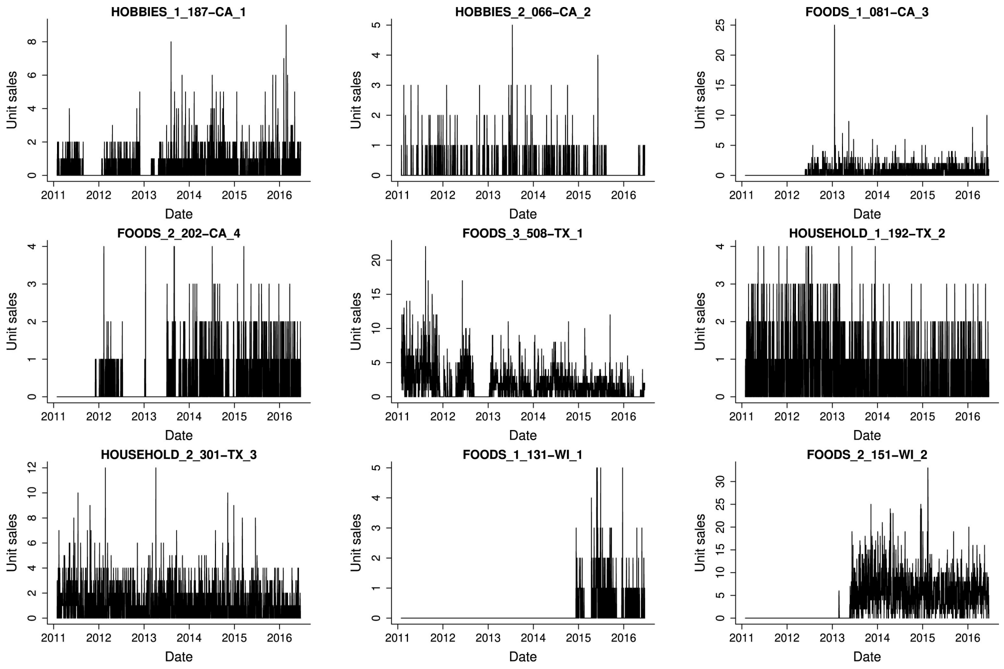


kaggle

Why use machine learning for forecasting?

- Large number of correlated time series (30,490)
- Hierarchical structure
- Varying length for each time series
- High sparsity & intermittency
- Exogenous variables (price, promos, etc.)
- Multiple seasonal patterns





[1] Makridakis, Spyros, Evangelos Spiliotis, and Vassilios Assimakopoulos. "The M5 competition: Background, organization, and implementation." *International Journal of Forecasting* (2021).

Why use machine learning for forecasting?

"… all of the **top-performing methods** were both **"pure" ML** approaches and **better than all statistical benchmarks** and their combinations. It was shown that **LightGBM** can be used effectively to process **numerous correlated series** and **exogenous**/explanatory variables, and to reduce the forecast errors." – [2]

[1] Makridakis, Spyros, Evangelos Spiliotis, and Vassilios Assimakopoulos. "The M5 competition: Background, organization, and implementation." *International Journal of Forecasting* (2021).

[2] Makridakis, Spyros, Evangelos Spiliotis, and Vassilios Assimakopoulos. "M5 accuracy competition: Results, findings, and conclusions." *International journal of forecasting* (2022).

Don’t neglect simple baselines though!

“The fact that about **92.5% of the participating teams failed to beat ES_bu [exponential smoothing]** should not be overlooked.”[2]

The top 50 entries improved on exponential smoothing by between ~15-20% [2].

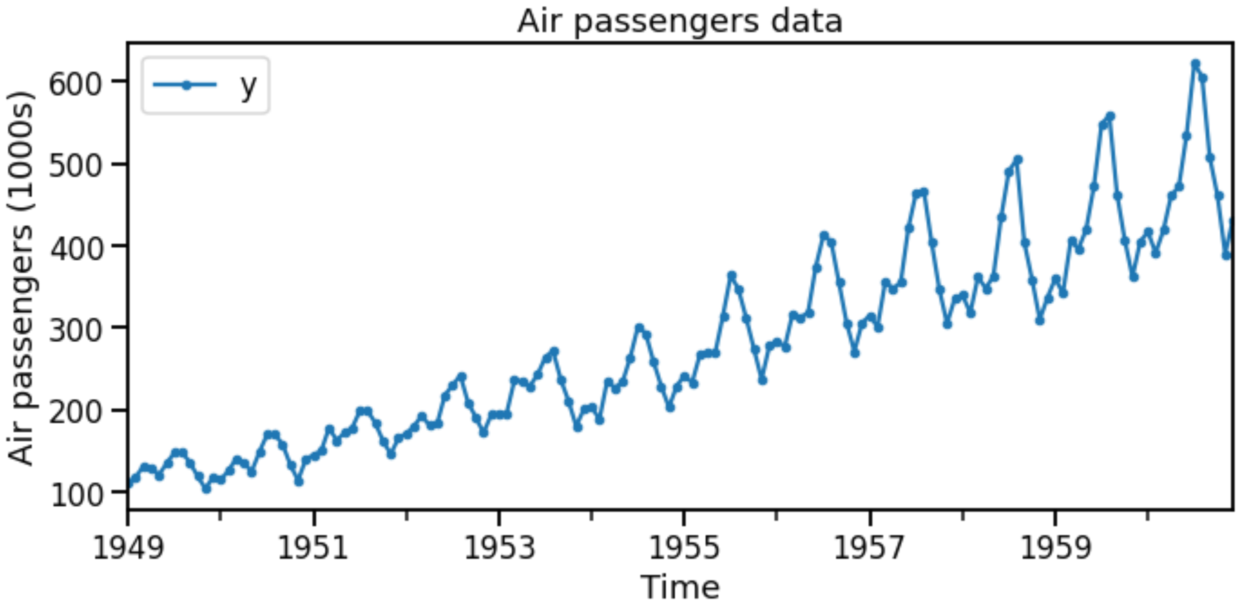
[1] Makridakis, Spyros, Evangelos Spiliotis, and Vassilios Assimakopoulos. "The M5 competition: Background, organization, and implementation." *International Journal of Forecasting* (2021).

[2] Makridakis, Spyros, Evangelos Spiliotis, and Vassilios Assimakopoulos. "M5 accuracy competition: Results, findings, and conclusions." *International journal of forecasting* (2022).

Use simpler methods for "easy" time series

Time series characteristics

- Strong seasonality and/or trend.
- Small number of time series.
- Uncorrelated time series.
- No sparsity or intermittency.
- Few or no exogenous features.



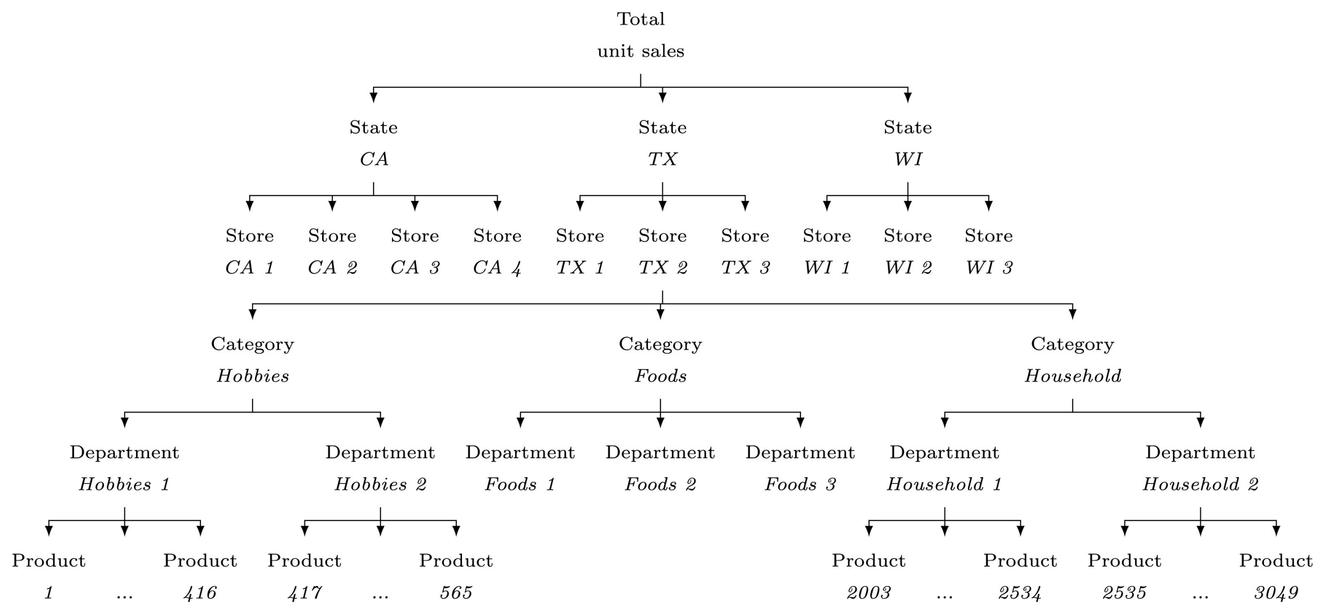
Methods

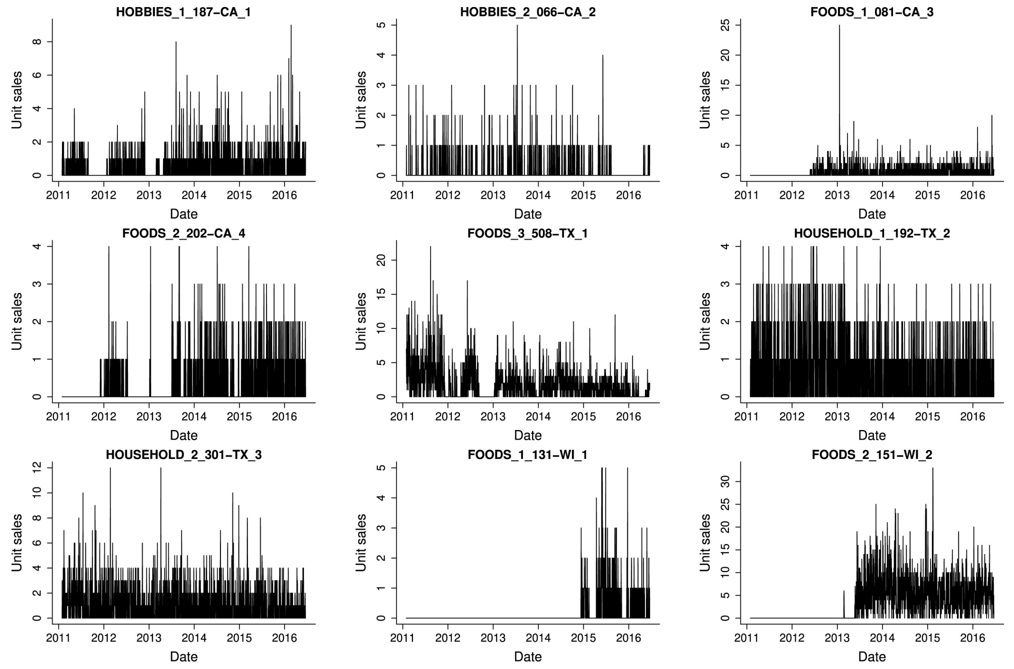
- Naïve and rules based forecasts
- ARIMA
- ETS
- Prophet

Use ML methods for “hard” time series

- Large number of correlated time series (30,490)
- Hierarchical structure
- Varying length for each time series

- High sparsity & intermittency
- Exogenous variables (price, promos, etc.)
- Multiple seasonal patterns

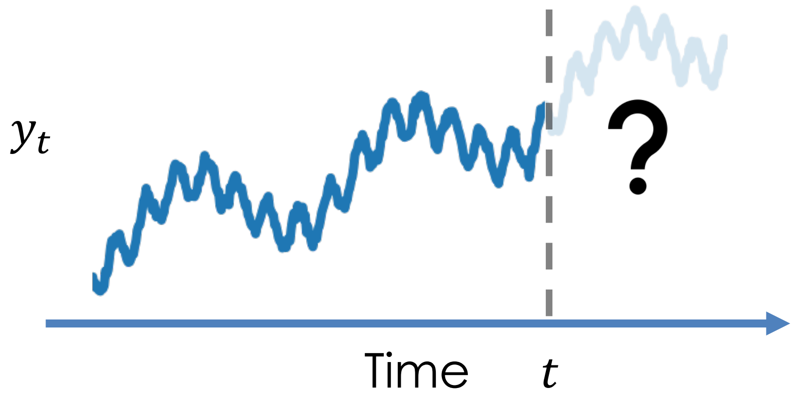




[1] Makridakis, Spyros, Evangelos Spiliotis, and Vassilios Assimakopoulos. "The M5 competition: Background, organization, and implementation." *International Journal of Forecasting* (2021).

Forecasting with machine learning

Time	Sales	
2020-02-01	35	
2020-02-02	30	
2020-02-03	23	$\vdots$
2020-02-04	21	$T-2$
2020-02-05	40	$T-1$
2020-02-06	31	T
2020-02-07	?	$T+1$



Time series to a table of features and a target

Time	Sales	
		$\vdots$
2020-02-01	35	
2020-02-02	30	
2020-02-03	23	
2020-02-04	21	$T-2$
2020-02-05	40	$T-1$
2020-02-06	31	T
2020-02-07	?	$T+1$

?

x1	x2	x3	y

Time series to a table of features and a target

Time	Sales	
2020-02-01	35	
2020-02-02	30	
2020-02-03	23	$\vdots$
2020-02-04	21	$T-2$
2020-02-05	40	$T-1$
2020-02-06	31	T
2020-02-07	?	$T+1$

?

x1	x2	x3	y_t
			35
			30
			23
			21
			40
			31
			?

Time series to a table of features and a target

Time	Sales
2020-02-01	35
2020-02-02	30
2020-02-03	23
2020-02-04	21
2020-02-05	40
2020-02-06	31
2020-02-07	?

$\vdots$

$T-2$

$T-1$

T

$T+1$

Only use data known at time of target.

This is to avoid look-ahead bias.

x1	x2	x3	y_t
			35
			30
			23
			21
			40
			31
			?

Time series to a table of features and a target

Time	Sales
2020-02-01	35
2020-02-02	30
2020-02-03	23
2020-02-04	21
2020-02-05	40
2020-02-06	31
2020-02-07	?

$\vdots$
$T-2$
$T-1$
T
$T+1$

$f(y_T, y_{T-1}, y_{T-2}, \dots)$

x1	x2	x3	y_t
			35
			30
			23
			21
			40
			31
			?

Time series to a table of features and a target

Time	Sales
2020-02-01	35
2020-02-02	30
2020-02-03	23
2020-02-04	21
2020-02-05	40
2020-02-06	31
2020-02-07	?

$\vdots$

$T-2$

$T-1$

T

$T+1$

y_{t-3}	y_{t-2}	y_{t-1}	y_t
			35
			30
			23
			21
			40
			31
21	**40**	**31**	?

Time series to a table of features and a target

Time	Sales	
2020-02-01	35	
2020-02-02	30	
2020-02-03	23	$\vdots$
2020-02-04	21	$T-2$
2020-02-05	40	$T-1$
2020-02-06	31	T
2020-02-07	?	$T+1$

y_{t-3}	y_{t-2}	y_{t-1}	y_t
			35
			30
			23
			21
			40
23	**21**	**40**	31
21	40	31	?

Time series to a table of features and a target

Time	Sales
2020-02-01	35
2020-02-02	30
2020-02-03	23
2020-02-04	21
2020-02-05	40
2020-02-06	31
2020-02-07	?

$\vdots$

$T-2$

$T-1$

T

$T+1$

y_{t-3}	y_{t-2}	y_{t-1}	y_t
			35
			30
			23
			21
30	**23**	**21**	40
23	21	40	31
21	40	31	?

Time series to a table of features and a target

Time	Sales
2020-02-01	35
2020-02-02	30
2020-02-03	23
2020-02-04	21
2020-02-05	40
2020-02-06	31
2020-02-07	?

y_{t-3}	y_{t-2}	y_{t-1}	y_t
NaN	NaN	NaN	35
NaN	NaN	35	30
NaN	35	30	23
35	30	23	21
30	23	21	40
23	21	40	31
21	40	31	?

Time series to a table of features and a target

Time	Sales
2020-02-01	35
2020-02-02	30
2020-02-03	23
2020-02-04	21
2020-02-05	40
2020-02-06	31
2020-02-07	?

y_{t-3}	y_{t-2}	y_{t-1}	y_t
NaN	NaN	NaN	35
NaN	NaN	35	30
NaN	35	30	23
35	30	23	21
30	23	21	40
23	21	40	31
21	40	31	?

Features derived from **past values** of y_t (e.g., lag features).

Time series to a table of features and a target

Time	Sales
2020-02-01	35
2020-02-02	30
2020-02-03	23
2020-02-04	21
2020-02-05	40
2020-02-06	31
2020-02-07	?

Ad spend	y_{t-3}	y_{t-2}	y_{t-1}	y_t
100	NaN	NaN	NaN	35
120	NaN	NaN	35	30
116	NaN	35	30	23
120	35	30	23	21
101	30	23	21	40
90	23	21	40	31
190	21	40	31	?

Features with **known values in the future.**

Features derived from past values of y_t (e.g., lag features).

Time series to a table of features and a target

Time	Sales
2020-02-01	35
2020-02-02	30
2020-02-03	23
2020-02-04	21
2020-02-05	40
2020-02-06	31
2020-02-07	?

Rainfall	Ad spend	y_{t-3}	y_{t-2}	y_{t-1}	y_t
12	100	NaN	NaN	NaN	35
15	120	NaN	NaN	35	30
13	116	NaN	35	30	23
14	120	35	30	23	21
23	101	30	23	21	40
25	90	23	21	40	31
?	190	21	40	31	?

Features with **unknown values in the future**.

Features with known values in the future.

Features derived from past values of y_t (e.g., lag features).

Time series to a table of features and a target

Time	Sales
2020-02-01	35
2020-02-02	30
2020-02-03	23
2020-02-04	21
2020-02-05	40
2020-02-06	31
2020-02-07	?

Rainfall	Ad spend	y_{t-3}	y_{t-2}	y_{t-1}	y_t
12	100	NaN	NaN	NaN	35
15	120	NaN	NaN	35	30
13	116	NaN	35	30	23
14	120	35	30	23	21
23	101	30	23	21	40
25	90	23	21	40	31
$\hat{x}_{T+1}$	190	21	40	31	?

Features with **unknown values in the future**.

Features with known values in the future.

Features derived from past values of y_t (e.g., lag features).

Time series to a table of features and a target

Time	Sales
2020-02-01	35
2020-02-02	30
2020-02-03	23
2020-02-04	21
2020-02-05	40
2020-02-06	31
2020-02-07	?

Country	Rainfall	Ad spend	y_{t-3}	y_{t-2}	y_{t-1}	y_t
UK	12	100	NaN	NaN	NaN	35
UK	15	120	NaN	NaN	35	30
UK	13	116	NaN	35	30	23
UK	14	120	35	30	23	21
UK	23	101	30	23	21	40
UK	25	90	23	21	40	31
UK	$\hat{x}_{T+1}$	190	21	40	31	?

Static features.

Features with unknown values in the future.

Features with known values in the future.

Features derived from past values of y_t (e.g., lag features).

Time series to a table of features and a target

Time	Sales
2020-02-01	35
2020-02-02	30
2020-02-03	23
2020-02-04	21
2020-02-05	40
2020-02-06	31
2020-02-07	?

Country	Rainfall	Ad spend	y_{t-3}	y_{t-2}	y_{t-1}	y_t
UK	12	100	NaN	NaN	NaN	35
UK	15	120	NaN	NaN	35	30
UK	13	116	NaN	35	30	23
UK	14	120	35	30	23	21
UK	23	101	30	23	21	40
UK	25	90	23	21	40	31
UK	$\hat{x}_{T+1}$	190	21	40	31	?

X
(features)

y
(target)

Time series to a table of features and a target

Country	Rainfall	Ad spend	y_{t-3}	y_{t-2}	y_{t-1}	y_t
UK	12	100	NaN	NaN	NaN	35
UK	15	120	NaN	NaN	35	30
UK	13	116	NaN	35	30	23
UK	14	120	35	30	23	21
UK	23	101	30	23	21	40
UK	25	90	23	21	40	31
UK	$\hat{x}_{T+1}$	190	21	40	31	?

Training data

X
(features)

y
(target)

Time series to a table of features and a target

model.fit

Country	Rainfall	Ad spend	y_{t-3}	y_{t-2}	y_{t-1}	y_t
UK	12	100	NaN	NaN	NaN	35
UK	15	120	NaN	NaN	35	30
UK	13	116	NaN	35	30	23
UK	14	120	35	30	23	21
UK	23	101	30	23	21	40
UK	25	90	23	21	40	31
UK	$\hat{x}_{T+1}$	190	21	40	31	?

$\underbrace{\text{Country, Rainfall, Ad spend, } y_{t-3}, y_{t-2}, y_{t-1}}_{\textbf{X}\text{ (features)}}$ $\underbrace{y_t}_{\textbf{y}\text{ (target)}}$

Time series to a table of features and a target

Country	Rainfall	Ad spend	y_{t-3}	y_{t-2}	y_{t-1}	y_t
UK	12	100	NaN	NaN	NaN	35
UK	15	120	NaN	NaN	35	30
UK	13	116	NaN	35	30	23
UK	14	120	35	30	23	21
UK	23	101	30	23	21	40
UK	25	90	23	21	40	31
UK	$\hat{x}_{T+1}$	**190**	**21**	**40**	**31**	?

model.fit

model.predict( )

$\hat{y}_{T+1}$

X (features)

y (target)

Multi-step forecasting

Time	Sales
2020-02-01	35
2020-02-02	30
2020-02-03	23
2020-02-04	21
2020-02-05	40
2020-02-06	31
2020-02-07	32
2020-02-08	?
2022-02-09	?
2022-02-10	?

$\vdots$

$T-2$

$T-1$

T

$T+1$

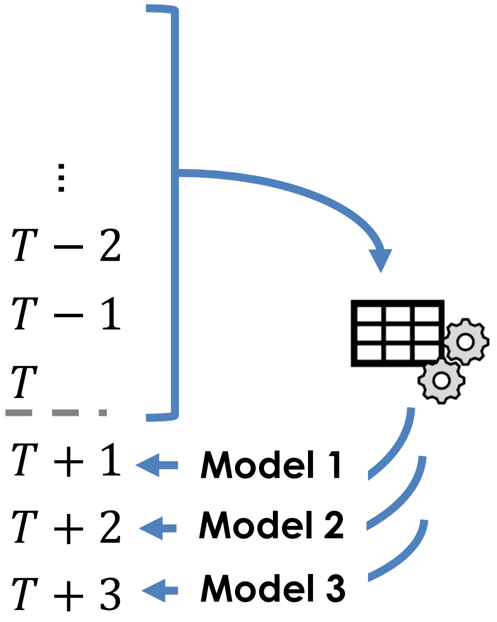
$T+2$

$T+3$

- Direct forecasting
- Recursive forecasting

Multi-step forecasting: Direct forecasting

Time	Sales
2020-02-01	35
2020-02-02	30
2020-02-03	23
2020-02-04	21
2020-02-05	40
2020-02-06	31
2020-02-07	32
2020-02-08	?
2022-02-09	?
2022-02-10	?



- Directly predict $y_{T+1}, y_{T+2}, \ldots, y_{T+h}$
- **Same features** but **different target variable** for each forecast step.
- **Multiple models** trained with different targets, one for each forecast step.

Multi-step forecasting: Direct forecasting

Time	Sales
2020-02-01	**35**
2020-02-02	**30**
2020-02-03	**23**
2020-02-04	21
2020-02-05	40
2020-02-06	31
2020-02-07	32
2020-02-08	?
2022-02-09	?
2022-02-10	?

Feature vector

y_{t-3}	y_{t-2}	y_{t-1}
35	**30**	**23**

Multi-step forecasting: Direct forecasting

Time	Sales
2020-02-01	35
2020-02-02	30
2020-02-03	23
2020-02-04	**21**
2020-02-05	40
2020-02-06	31
2020-02-07	32
2020-02-08	?
2022-02-09	?
2022-02-10	?

Feature vector

y_{t-3}	y_{t-2}	y_{t-1}
35	30	23

Multiple targets

y_{t+1}
21

Multi-step forecasting: Direct forecasting

Time	Sales
2020-02-01	35
2020-02-02	30
2020-02-03	23
2020-02-04	21
2020-02-05	**40**
2020-02-06	31
2020-02-07	32
2020-02-08	?
2022-02-09	?
2022-02-10	?

Feature vector

y_{t-3}	y_{t-2}	y_{t-1}
35	30	23

Multiple targets

y_{t+1}	y_{t+2}
21	**40**

Multi-step forecasting: Direct forecasting

Time	Sales
2020-02-01	35
2020-02-02	30
2020-02-03	23
2020-02-04	21
2020-02-05	40
2020-02-06	**31**
2020-02-07	32
2020-02-08	?
2022-02-09	?
2022-02-10	?

Feature vector

y_{t-3}	y_{t-2}	y_{t-1}
35	30	23

Multiple targets

y_{t+1}	y_{t+2}	y_{t+3}
21	40	**31**

Multi-step forecasting: Direct forecasting

Time	Sales
2020-02-01	35
2020-02-02	**30**
2020-02-03	**23**
2020-02-04	**21**
2020-02-05	40
2020-02-06	31
2020-02-07	32
2020-02-08	?
2022-02-09	?
2022-02-10	?

Feature vectors

y_{t-3}	y_{t-2}	y_{t-1}
35	30	23
30	**23**	**21**

Multiple targets

y_{t+1}	y_{t+2}	y_{t+3}
21	40	31

Multi-step forecasting: Direct forecasting

Time	Sales
2020-02-01	35
2020-02-02	30
2020-02-03	23
2020-02-04	21
2020-02-05	**40**
2020-02-06	**31**
2020-02-07	**32**
2020-02-08	?
2022-02-09	?
2022-02-10	?

Feature vectors

y_{t-3}	y_{t-2}	y_{t-1}
35	30	23
30	23	21

Multiple targets

y_{t+1}	y_{t+2}	y_{t+3}
21	40	31
40	**31**	**32**

Multi-step forecasting: Direct forecasting

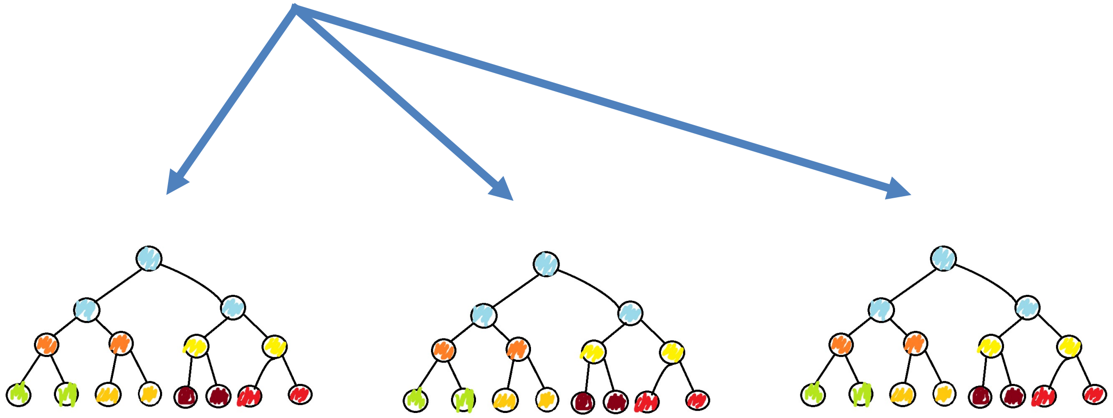
Time	Sales
2020-02-01	35
2020-02-02	30
2020-02-03	23
2020-02-04	21
2020-02-05	40
2020-02-06	31
2020-02-07	32
2020-02-08	?
2022-02-09	?
2022-02-10	?

Feature vectors

y_{t-3}	y_{t-2}	y_{t-1}
35	30	23
30	23	21

Multiple targets

y_{t+1}	y_{t+2}	y_{t+3}
21	40	31
40	31	32



Multi-step forecasting: Direct forecasting

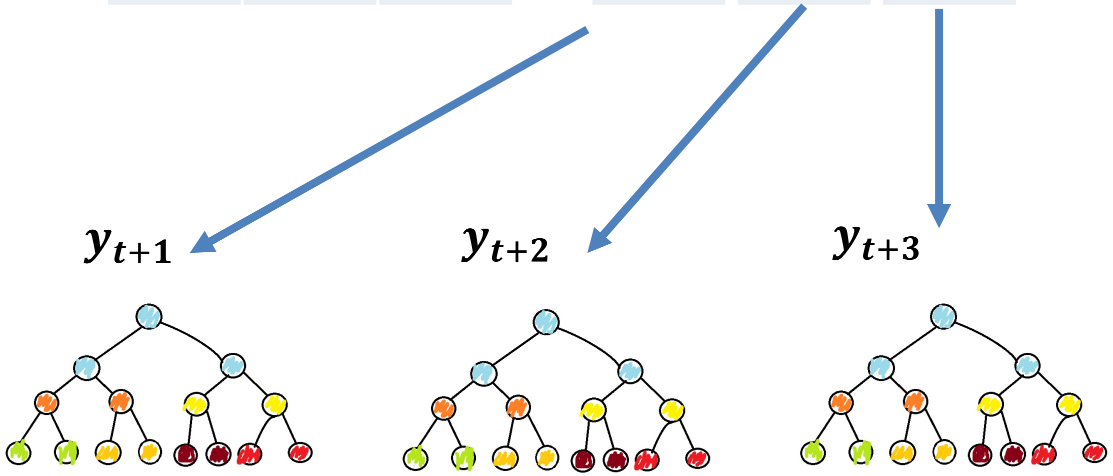
Time	Sales
2020-02-01	35
2020-02-02	30
2020-02-03	23
2020-02-04	21
2020-02-05	40
2020-02-06	31
2020-02-07	32
2020-02-08	?
2022-02-09	?
2022-02-10	?

Feature vectors

y_{t-3}	y_{t-2}	y_{t-1}
35	30	23
30	23	21

Multiple targets

y_{t+1}	y_{t+2}	y_{t+3}
21	40	31
40	31	32



Multi-step forecasting: Direct forecasting

Time	Sales
2020-02-01	35
2020-02-02	30
2020-02-03	23
2020-02-04	21
2020-02-05	**40**
2020-02-06	**31**
2020-02-07	**32**
2020-02-08	?
2022-02-09	?
2022-02-10	?

Feature vectors

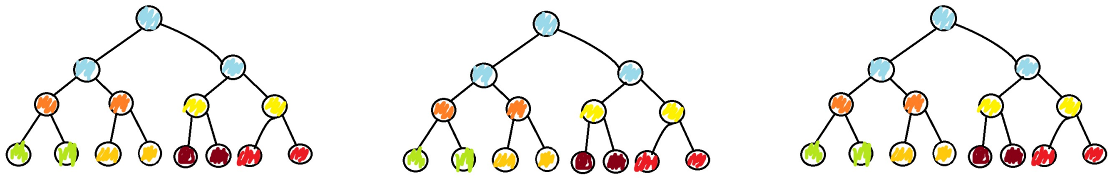
y_{t-3}	y_{t-2}	y_{t-1}
35	30	23
30	23	21

Multiple targets

y_{t+1}	y_{t+2}	y_{t+3}
21	40	31
40	31	32

Input

y_{t-3}	y_{t-2}	y_{t-1}
40	**31**	**32**



Multi-step forecasting: Direct forecasting

Time	Sales
2020-02-01	35
2020-02-02	30
2020-02-03	23
2020-02-04	21
2020-02-05	40
2020-02-06	31
2020-02-07	32
2020-02-08	$\hat{y}_{T+1}$
2022-02-09	$\hat{y}_{T+2}$
2022-02-10	$\hat{y}_{T+3}$

Input

Output

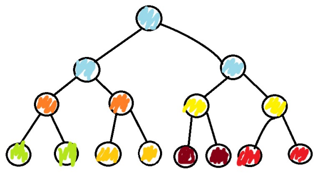
Feature vectors

y_{t-3}	y_{t-2}	y_{t-1}
35	30	23
30	23	21

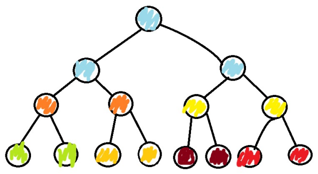
Multiple targets

y_{t+1}	y_{t+2}	y_{t+3}
21	40	31
40	31	32

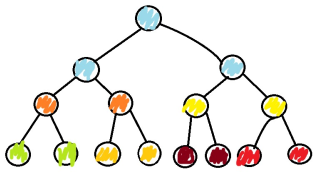
y_{t-3}	y_{t-2}	y_{t-1}
40	31	32



$\hat{y}_{T+1}$



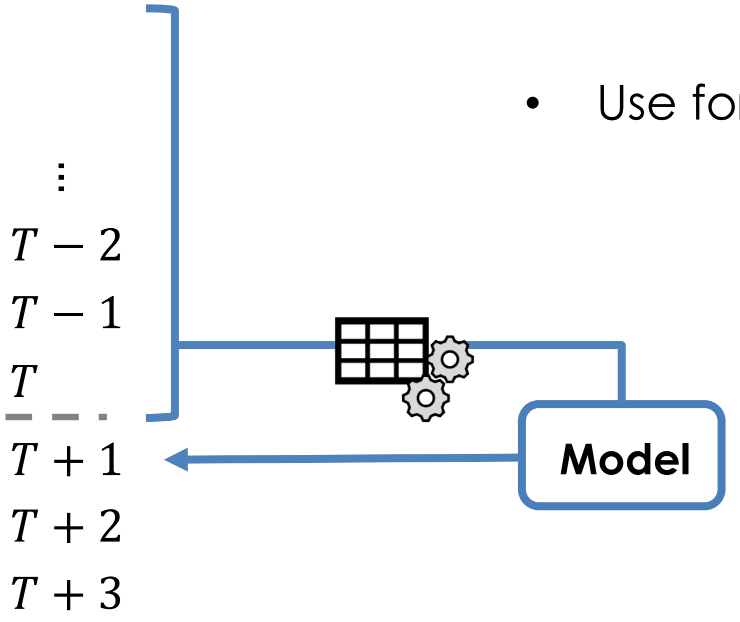
$\hat{y}_{T+2}$



$\hat{y}_{T+3}$

Multi-step forecasting: Recursive forecasting

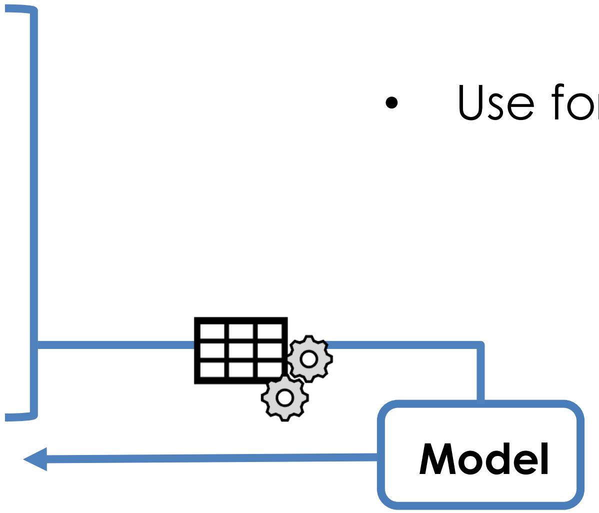
Time	Sales
2020-02-01	35
2020-02-02	30
2020-02-03	23
2020-02-04	21
2020-02-05	40
2020-02-06	31
2020-02-07	?
2022-02-08	?
2022-02-09	?



- Recursively apply a 1-step ahead forecast model.
- Use forecasted output as new input.

Multi-step forecasting: Recursive forecasting

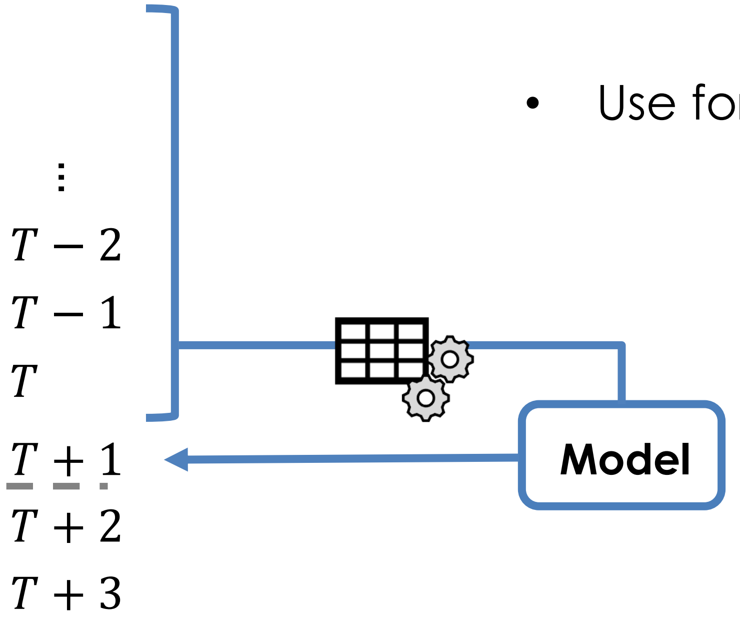
Time	Sales	
2020-02-01	35	
2020-02-02	30	
2020-02-03	23	$\vdots$
2020-02-04	21	$T-2$
2020-02-05	40	$T-1$
2020-02-06	31	T
2020-02-07	$\hat{y}_{T+1}$	$T+1$
2022-02-08	?	$T+2$
2022-02-09	?	$T+3$



- Recursively apply a 1-step ahead forecast model.
- Use forecasted output as new input.

Multi-step forecasting: Recursive forecasting

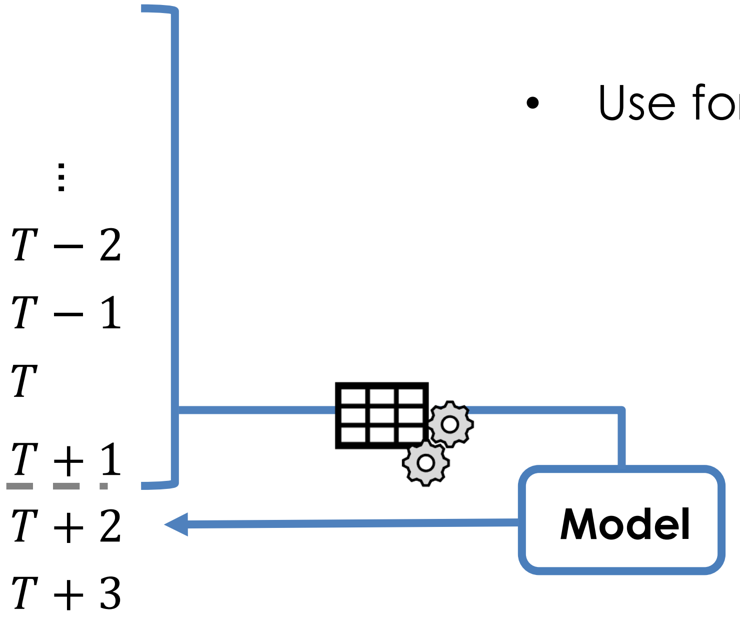
Time	Sales
2020-02-01	35
2020-02-02	30
2020-02-03	23
2020-02-04	21
2020-02-05	40
2020-02-06	31
2020-02-07	$\hat{y}_{T+1}$
2022-02-08	?
2022-02-09	?



- Recursively apply a 1-step ahead forecast model.
- Use forecasted output as new input.

Multi-step forecasting: Recursive forecasting

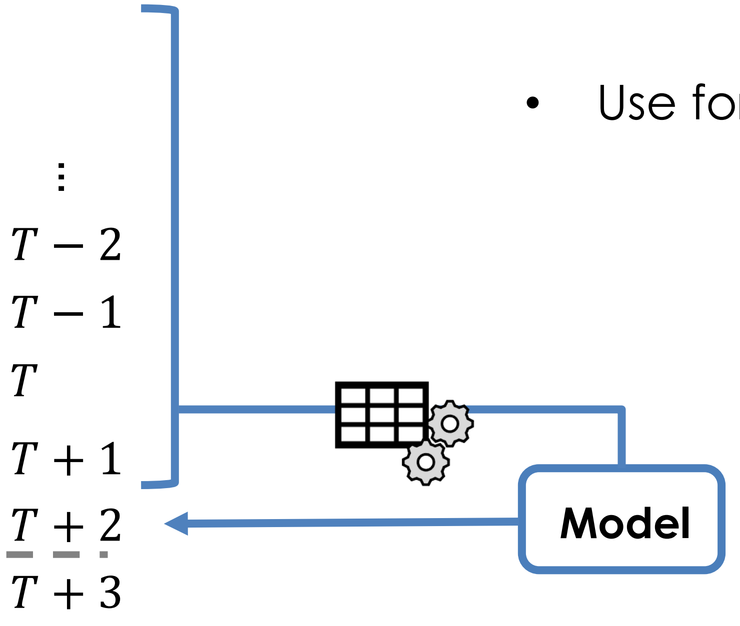
Time	Sales
2020-02-01	35
2020-02-02	30
2020-02-03	23
2020-02-04	21
2020-02-05	40
2020-02-06	31
2020-02-07	$\hat{y}_{T+1}$
2022-02-08	$\hat{y}_{T+2}$
2022-02-09	?



- Recursively apply a 1-step ahead forecast model.
- Use forecasted output as new input.

Multi-step forecasting: Recursive forecasting

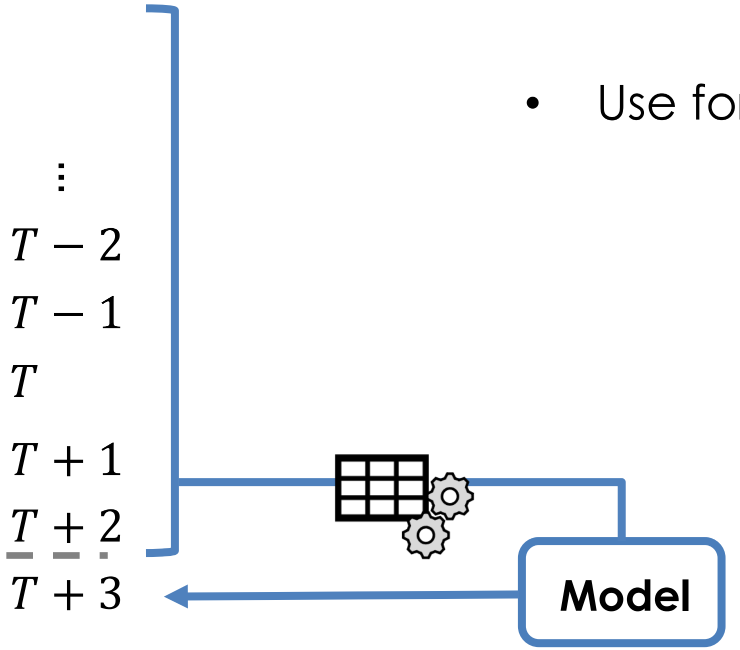
Time	Sales
2020-02-01	35
2020-02-02	30
2020-02-03	23
2020-02-04	21
2020-02-05	40
2020-02-06	31
2020-02-07	$\hat{y}_{T+1}$
2022-02-08	$\hat{y}_{T+2}$
2022-02-09	?



- Recursively apply a 1-step ahead forecast model.
- Use forecasted output as new input.

Multi-step forecasting: Recursive forecasting

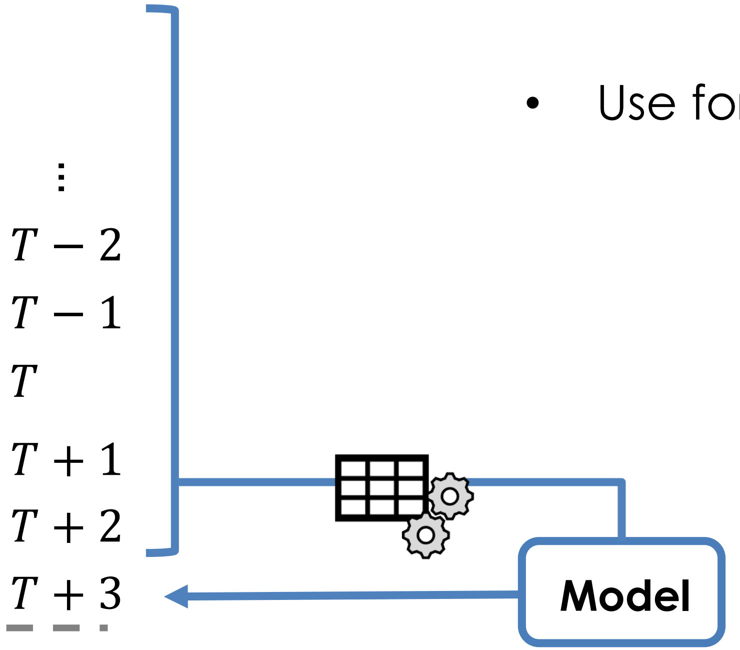
Time	Sales
2020-02-01	35
2020-02-02	30
2020-02-03	23
2020-02-04	21
2020-02-05	40
2020-02-06	31
2020-02-07	$\hat{y}_{T+1}$
2022-02-08	$\hat{y}_{T+2}$
2022-02-09	$\hat{\boldsymbol{y}}_{\boldsymbol{T+3}}$



- Recursively apply a 1-step ahead forecast model.
- Use forecasted output as new input.

Multi-step forecasting: Recursive forecasting

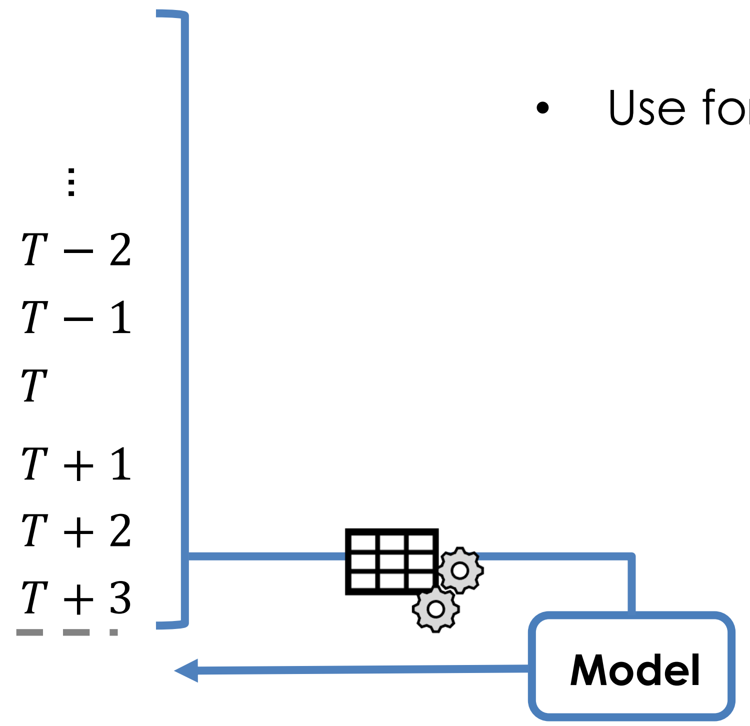
Time	Sales
2020-02-01	35
2020-02-02	30
2020-02-03	23
2020-02-04	21
2020-02-05	40
2020-02-06	31
2020-02-07	$\hat{y}_{T+1}$
2022-02-08	$\hat{y}_{T+2}$
2022-02-09	$\hat{y}_{T+3}$



- Recursively apply a 1-step ahead forecast model.
- Use forecasted output as new input.

Multi-step forecasting: Recursive forecasting

Time	Sales
2020-02-01	35
2020-02-02	30
2020-02-03	23
2020-02-04	21
2020-02-05	40
2020-02-06	31
2020-02-07	$\hat{y}_{T+1}$
2022-02-08	$\hat{y}_{T+2}$
2022-02-09	$\hat{y}_{T+3}$



- Recursively apply a 1-step ahead forecast model.
- Use forecasted output as new input.

Multi-step forecasting: Recursive forecasting

Time	Sales
2020-02-01	35
2020-02-02	30
2020-02-03	23
2020-02-04	21
2020-02-05	40
2020-02-06	31
2020-02-07	?
2022-02-08	?
2022-02-09	?

Country	Rainfall	Ad spend	y_{t-3}	y_{t-2}	y_{t-1}	y_t
UK	12	100	NaN	NaN	NaN	35
UK	15	120	NaN	NaN	35	30
UK	13	116	NaN	35	30	23
UK	14	120	35	30	23	21
UK	23	101	30	23	21	40
UK	25	90	23	21	40	31
UK	$\hat{x}_{T+1}$	190				?
UK	$\hat{x}_{T+2}$	201				?
UK	$\hat{x}_{T+3}$	110				?

- Features derived from the target variable need to created iteratively.

Multi-step forecasting: Recursive forecasting

Time	Sales	Country	Rainfall	Ad spend	y_{t-3}	y_{t-2}	y_{t-1}	y_t
2020-02-01	35	UK	12	100	NaN	NaN	NaN	35
2020-02-02	30	UK	15	120	NaN	NaN	35	30
2020-02-03	23	UK	13	116	NaN	35	30	23
2020-02-04	21	UK	14	120	35	30	23	21
2020-02-05	40	UK	23	101	30	23	21	40
2020-02-06	31	UK	25	90	23	21	40	31
2020-02-07	?	UK	$\hat{x}_{T+1}$	190				?
2022-02-08	?	UK	$\hat{x}_{T+2}$	201				?
2022-02-09	?	UK	$\hat{x}_{T+3}$	110				?

- Features derived from the target variable need to created iteratively.

Multi-step forecasting: Recursive forecasting

Time	Sales	Country	Rainfall	Ad spend	y_{t-3}	y_{t-2}	y_{t-1}	y_t
2020-02-01	35	UK	12	100	NaN	NaN	NaN	35
2020-02-02	30	UK	15	120	NaN	NaN	35	30
2020-02-03	23	UK	13	116	NaN	35	30	23
2020-02-04	21	UK	14	120	35	30	23	**21**
2020-02-05	40	UK	23	101	30	23	21	**40**
2020-02-06	31	UK	25	90	23	21	40	**31**
2020-02-07	?	UK	$\hat{x}_{T+1}$	190	**21**	**40**	**31**	?
2022-02-08	?	UK	$\hat{x}_{T+2}$	201				?
2022-02-09	?	UK	$\hat{x}_{T+3}$	110				?

- Features derived from the target variable need to created iteratively.

Multi-step forecasting: Recursive forecasting

	Country	Rainfall	Ad spend	y_{t-3}	y_{t-2}	y_{t-1}	y_t
model.fit	UK	12	100	NaN	NaN	NaN	35
	UK	15	120	NaN	NaN	35	30
	UK	13	116	NaN	35	30	23
	UK	14	120	35	30	23	21
	UK	23	101	30	23	21	40
	UK	25	90	23	21	40	31
model.predict( ) → $\hat{y}_{T+1}$	**UK**	$\hat{x}_{T+1}$	**190**	**21**	**40**	**31**	?
	UK	$\hat{x}_{T+2}$	201				?
	UK	$\hat{x}_{T+3}$	110				?

Multi-step forecasting: Recursive forecasting

model.fit

model.predict

Country	Rainfall	Ad spend	y_{t-3}	y_{t-2}	y_{t-1}	y_t
UK	12	100	NaN	NaN	NaN	35
UK	15	120	NaN	NaN	35	30
UK	13	116	NaN	35	30	23
UK	14	120	35	30	23	21
UK	23	101	30	23	21	40
UK	25	90	23	21	40	31
UK	$\hat{x}_{T+1}$	190	21	40	31	$\hat{y}_{T+1}$
UK	$\hat{x}_{T+2}$	201				?
UK	$\hat{x}_{T+3}$	110				?

$\hat{y}_{T+1}$

Multi-step forecasting: Recursive forecasting

model.fit

Country	Rainfall	Ad spend	y_{t-3}	y_{t-2}	y_{t-1}	y_t
UK	12	100	NaN	NaN	NaN	35
UK	15	120	NaN	NaN	35	30
UK	13	116	NaN	35	30	23
UK	14	120	35	30	23	21
UK	23	101	30	23	21	**40**
UK	25	90	23	21	40	**31**
UK	$\hat{x}_{T+1}$	190	21	40	31	$\hat{y}_{T+1}$
UK	$\hat{x}_{T+2}$	201	**40**	**31**	$\hat{y}_{T+1}$	?
UK	$\hat{x}_{T+3}$	110				?

Multi-step forecasting: Recursive forecasting

model.fit(

model.predict(

Country	Rainfall	Ad spend	y_{t-3}	y_{t-2}	y_{t-1}	y_t
UK	12	100	NaN	NaN	NaN	35
UK	15	120	NaN	NaN	35	30
UK	13	116	NaN	35	30	23
UK	14	120	35	30	23	21
UK	23	101	30	23	21	40
UK	25	90	23	21	40	31
UK	$\hat{x}_{T+1}$	190	21	40	31	$\hat{y}_{T+1}$
UK	$\hat{\boldsymbol{x}}_{T+2}$	**201**	**40**	**31**	$\hat{\boldsymbol{y}}_{T+1}$	?
UK	$\hat{x}_{T+3}$	110				?

$\hat{y}_{T+2}$

Multi-step forecasting: Recursive forecasting

model.fit

model.predict(

$\hat{y}_{T+2}$

Country	Rainfall	Ad spend	y_{t-3}	y_{t-2}	y_{t-1}	y_t
UK	12	100	NaN	NaN	NaN	35
UK	15	120	NaN	NaN	35	30
UK	13	116	NaN	35	30	23
UK	14	120	35	30	23	21
UK	23	101	30	23	21	40
UK	25	90	23	21	40	31
UK	$\hat{x}_{T+1}$	190	21	40	31	$\hat{y}_{T+1}$
UK	$\hat{x}_{T+2}$	201	40	31	$\hat{y}_{T+1}$	$\boldsymbol{\hat{y}_{T+2}}$
UK	$\hat{x}_{T+3}$	110				?

Multi-step forecasting: Recursive forecasting

model.fit

Country	Rainfall	Ad spend	y_{t-3}	y_{t-2}	y_{t-1}	y_t
UK	12	100	NaN	NaN	NaN	35
UK	15	120	NaN	NaN	35	30
UK	13	116	NaN	35	30	23
UK	14	120	35	30	23	21
UK	23	101	30	23	21	40
UK	25	90	23	21	40	**31**
UK	$\hat{x}_{T+1}$	190	21	40	31	$\hat{\boldsymbol{y}}_{\boldsymbol{T+1}}$
UK	$\hat{x}_{T+2}$	201	40	31	$\hat{y}_{T+1}$	$\hat{\boldsymbol{y}}_{\boldsymbol{T+2}}$
UK	$\hat{x}_{T+3}$	110	**31**	$\hat{\boldsymbol{y}}_{\boldsymbol{T+1}}$	$\hat{\boldsymbol{y}}_{\boldsymbol{T+2}}$	?

Multi-step forecasting: Recursive forecasting

model.fit

model.predict

Country	Rainfall	Ad spend	y_{t-3}	y_{t-2}	y_{t-1}	y_t
UK	12	100	NaN	NaN	NaN	35
UK	15	120	NaN	NaN	35	30
UK	13	116	NaN	35	30	23
UK	14	120	35	30	23	21
UK	23	101	30	23	21	40
UK	25	90	23	21	40	31
UK	$\hat{x}_{T+1}$	190	21	40	31	$\hat{y}_{T+1}$
UK	$\hat{x}_{T+2}$	201	40	31	$\hat{y}_{T+1}$	$\hat{y}_{T+2}$
UK	$\boldsymbol{\hat{x}_{T+3}}$	**110**	**31**	$\boldsymbol{\hat{y}_{T+1}}$	$\boldsymbol{\hat{y}_{T+2}}$	?

$\hat{y}_{T+3}$

Multi-step forecasting: Recursive forecasting

model.fit

Country	Rainfall	Ad spend	y_{t-3}	y_{t-2}	y_{t-1}	y_t
UK	12	100	NaN	NaN	NaN	35
UK	15	120	NaN	NaN	35	30
UK	13	116	NaN	35	30	23
UK	14	120	35	30	23	21
UK	23	101	30	23	21	40
UK	25	90	23	21	40	31
UK	$\hat{x}_{T+1}$	190	21	40	31	$\hat{y}_{T+1}$
UK	$\hat{x}_{T+2}$	201	40	31	$\hat{y}_{T+1}$	$\hat{y}_{T+2}$
UK	$\hat{x}_{T+3}$	110	31	$\hat{y}_{T+1}$	$\hat{y}_{T+2}$	$\boldsymbol{\hat{y}_{T+3}}$

model.predict( )

$\hat{y}_{T+3}$

Direct vs recursive multistep forecasting

Direct

Pros

- Directly optimised for each forecast step

Cons

- Multiple models
- Computational cost
- Forecasts are independent

Recursive

Pros

- Only 1 model
- Less computation time
- Forecasts are dependent

Cons

- Code complexity in making features from forecasts
- Errors propagate
- Model optimises on one step forecast

Cross-validation: Tabular vs Time series

country	spend	credit risk	churn
'France'	$1200	low	0
'Japan'	$300	medium	1
'UK'	$20	low	0
'Italy'	$100,000	high	0
'France'	$310	medium	0

Randomly split

Training data

country	spend	credit risk	churn
'France'	$1200	low	0
'UK'	$20	low	0
'Italy'	$100,000	high	0

Testing data

country	spend	credit risk	churn
'Japan'	$300	medium	1
'France'	$310	medium	0

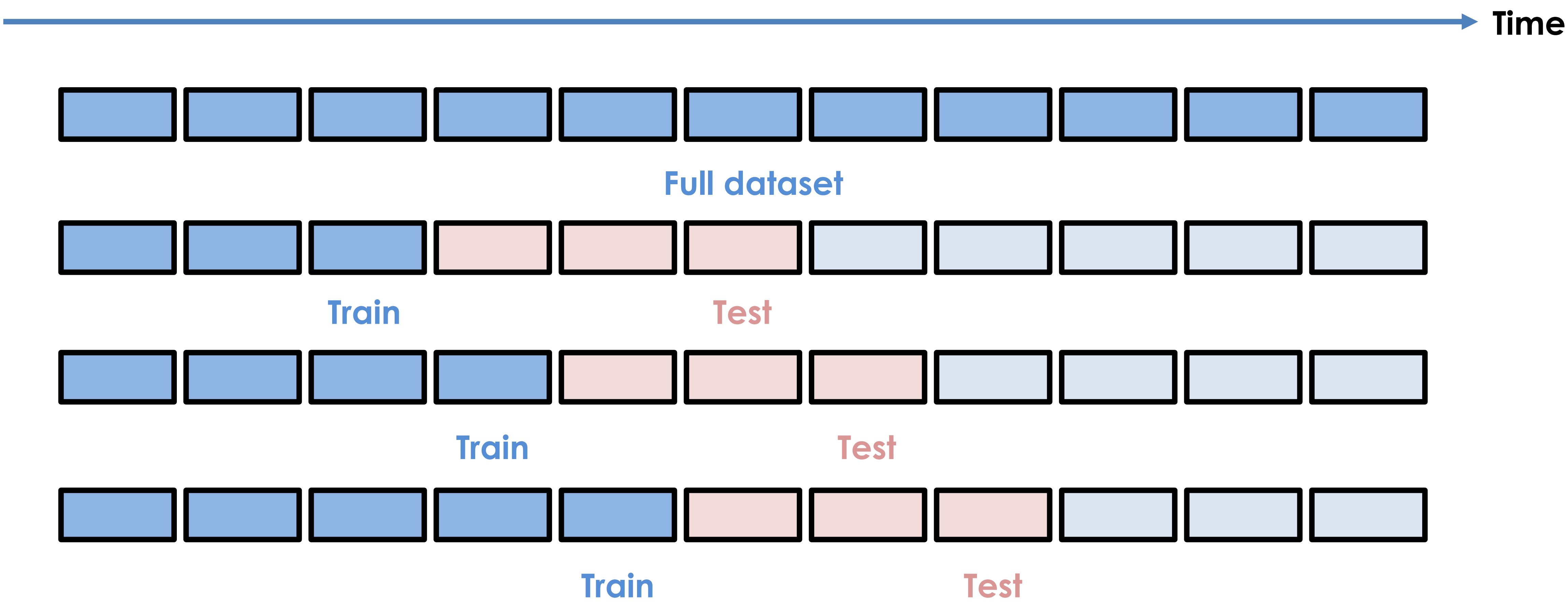
- Because **each row is independent** we can **randomly split** the data into train and test.

Cross-validation: Tabular vs Time series

Time	Sales (UK)	Country	Rainfall	Ad spend	y_{t-3}	y_{t-2}	y_{t-1}	y_t
2020-02-12	35	UK	100	100	NaN	NaN	NaN	35
2020-02-13	30	UK	120	120	NaN	NaN	35	30
2020-02-14	23	UK	116	116	NaN	35	30	23
2020-02-15	21	UK	120	120	35	30	23	21
2020-02-16	40	UK	101	101	30	23	21	40
2020-02-17	31	UK	90	90	23	21	40	31

- **Cannot split randomly** because the time ordering means **each row is not independent**.
- Instead **need to split by time** to replicate the actual forecasting process.

Cross-validation: Tabular vs Time series



Machine learning workflow

	ML on tabular data (regression & classification)	ML on tabular data (forecasting)
Train/test split	Random allocation.	Split by time.
Creating the feature and target	Can pre-compute features and target before predict time.	Features built from target created ”on demand” at predict time for test set.
Prediction	Only the trained model required at predict time.	Need trained model & training set at predict time.
Feature engineering		Time series specific feature engineering and data leakage issues.

Machine learning workflow

	ML on tabular data (regression & classification)	ML on tabular data (forecasting)
Train/test split	**Random allocation.**	**Split by time.**
Creating the feature and target	Can pre-compute features and target before predict time.	Features built from target created ”on demand” at predict time for test set.
Prediction	Only the trained model required at predict time.	Need trained model & training set at predict time.
Feature engineering		Time series specific feature engineering and data leakage issues.

Machine learning workflow

	ML on tabular data (regression & classification)	ML on tabular data (forecasting)
Train/test split	Random allocation.	Split by time.
Creating the feature and target	**Can pre-compute features and target before predict time.**	**Features built from target created ”on demand” at predict time for test set.**
Prediction	Only the trained model required at predict time.	Need trained model & training set at predict time.
Feature engineering		Time series specific feature engineering and data leakage issues.

Machine learning workflow

	ML on tabular data (regression & classification)	ML on tabular data (forecasting)
Train/test split	Random allocation.	Split by time.
Creating the feature and target	Can pre-compute features and target before predict time.	Features built from target created "on demand" at predict time for test set.
Prediction	**Only the trained model required at predict time.**	**Need trained model & training set at predict time.**
Feature engineering		Time series specific feature engineering and data leakage issues.

Machine learning workflow

	ML on tabular data (regression & classification)	ML on tabular data (forecasting)
Train/test split	Random allocation.	Split by time.
Creating the feature and target	Can pre-compute features and target before predict time.	Features built from target created "on demand" at predict time for test set.
Prediction	Only the trained model required at predict time.	Need trained model & training set at predict time.
Feature engineering		**Time series specific feature engineering and data leakage issues.**

Contents

Time series forecasting using ML models

Features for time series forecasting

Useful libraries for forecasting with ML models

Feature engineering for time series forecasting

Imputation	Outliers	Transformation	Encoding	Temporal	Past features	Trend and seasonality
Forward fill, backfill	Identify	Log, Box Cox	One hot	Calendar Day, week, month Holidays	Lag features	Time variable, changepoint, step changes
Interpolate	Dummy variable	Seasonal & trend adjustment	Target mean, integer	Cyclical feature encoding	Window features	Fourier series, seasonal dummies, seasonal lags

Feature engineering for time series forecasting

Imputation	Outliers	Transformation	Encoding	Temporal	Past features	Trend and seasonality
Forward fill, backfill	Identify	Log, Box Cox	One hot	Calendar Day, week, month Holidays	**Lag features**	Time variable, changepoint, step changes
Interpolate	Dummy variable	Seasonal & trend adjustment	**Target mean,** integer	Cyclical feature encoding	**Window features**	Fourier series, seasonal dummies, **seasonal lags**

Which data can I use as features?

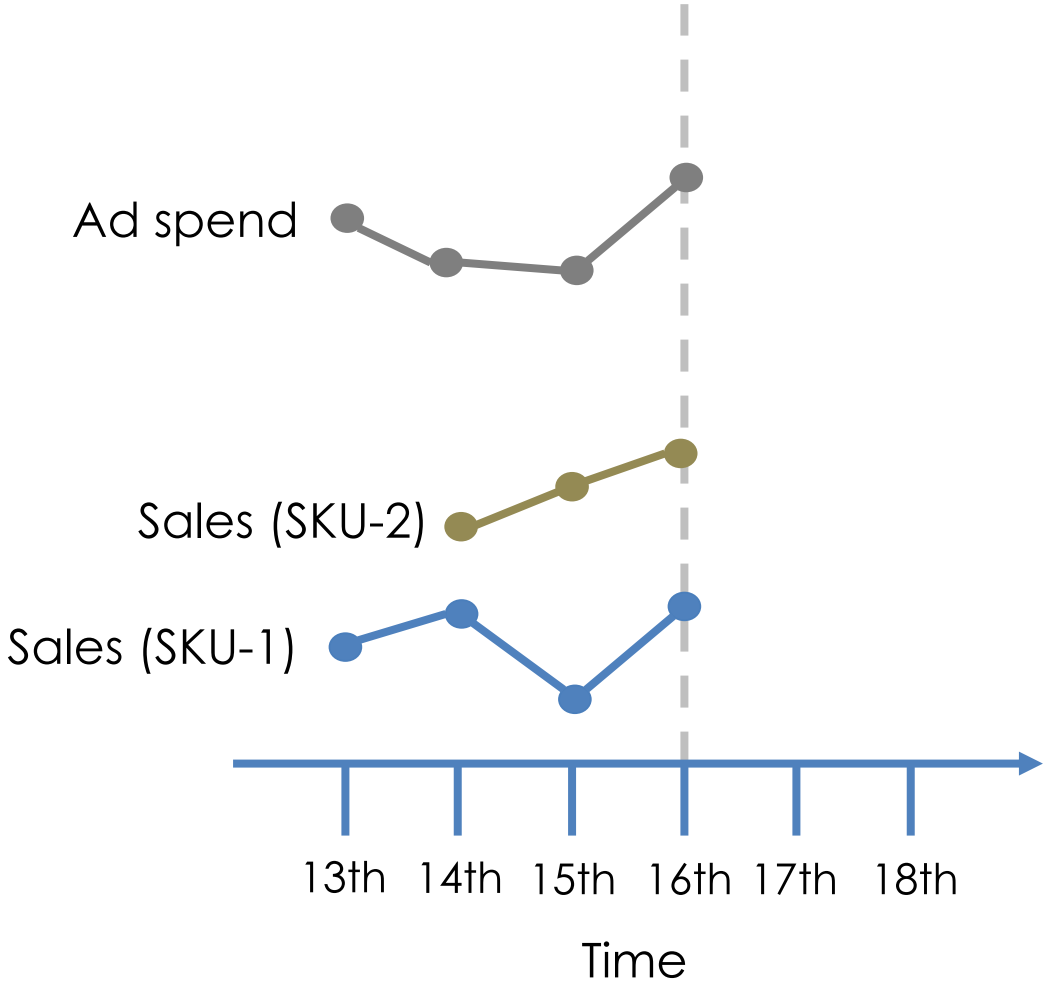
Any data that is known at the time of prediction.

This includes knowledge about future values of a feature.

We need to be **very careful** not to accidentally use data that is not available at the time of prediction.

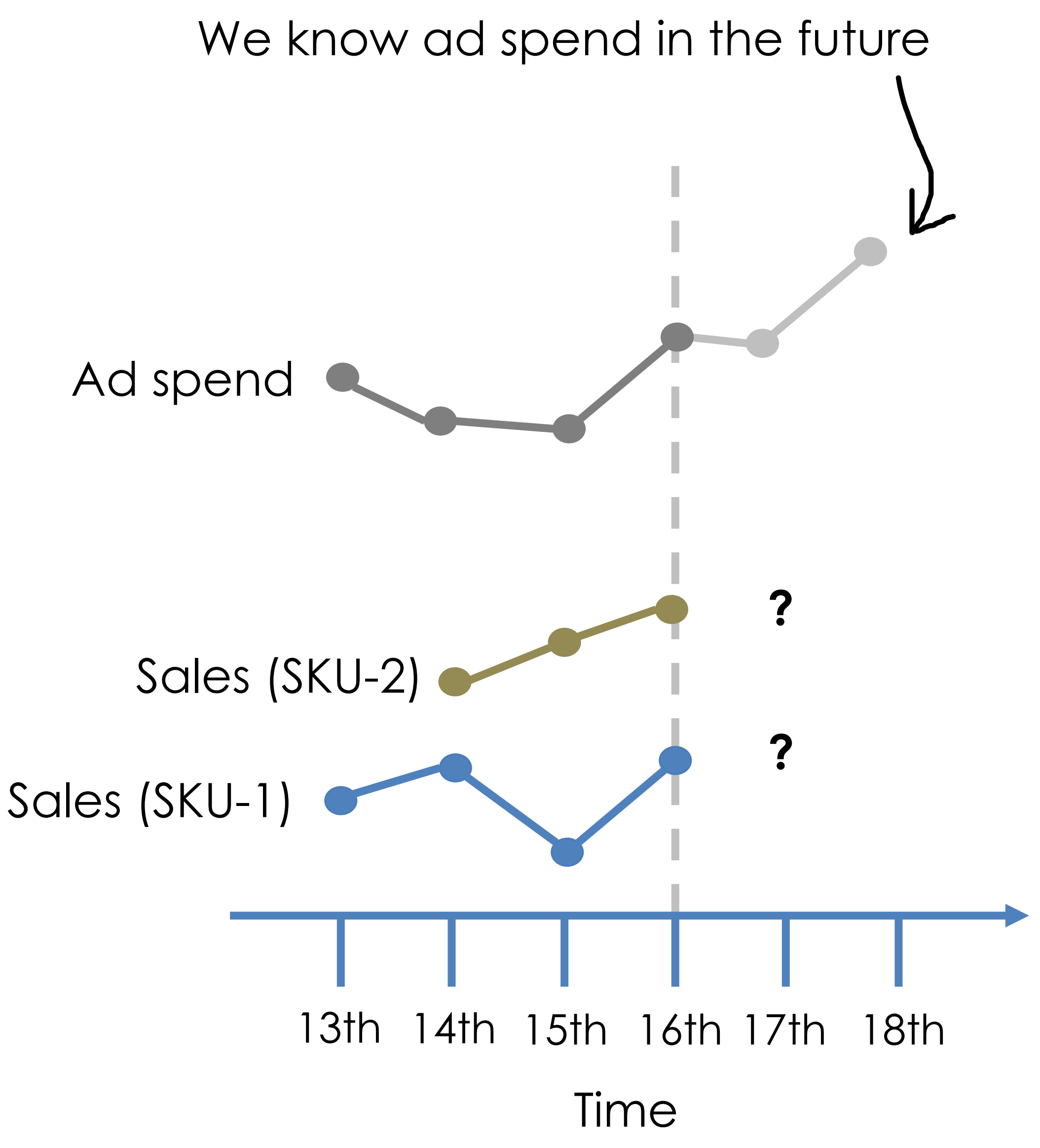
An example

Time	Product ID	Ad spend	Sales
…	…	…	…
2020-02-13	SKU-1	100	30
2020-02-14	SKU-1	120	32
2020-02-15	SKU-1	110	25
2020-02-16	SKU-1	101	34
…	…	…	…
2020-02-14	SKU-2	120	32
2020-02-15	SKU-2	110	21
2020-02-16	SKU-2	101	25



An example

Time	Product ID	Ad spend	Sales
...	...	...	...
2020-02-13	SKU-1	100	30
2020-02-14	SKU-1	120	32
2020-02-15	SKU-1	110	25
2020-02-16	SKU-1	101	34
2020-02-17	**SKU-1**	**102**	**?**
...	...	...	...
2020-02-14	SKU-2	120	32
2020-02-15	SKU-2	110	21
2020-02-16	SKU-2	101	15
2020-02-17	**SKU-2**	**102**	**?**



Target variable

Time	Product ID	Ad spend	Sales
...	...	...	...
2020-02-13	SKU-1	100	30
2020-02-14	SKU-1	120	32
2020-02-15	SKU-1	110	25
2020-02-16	SKU-1	101	34
2020-02-17	SKU-1	102	?
...	...	...	...
2020-02-14	SKU-2	120	32
2020-02-15	SKU-2	110	21
2020-02-16	SKU-2	101	15
2020-02-17	SKU-2	102	?

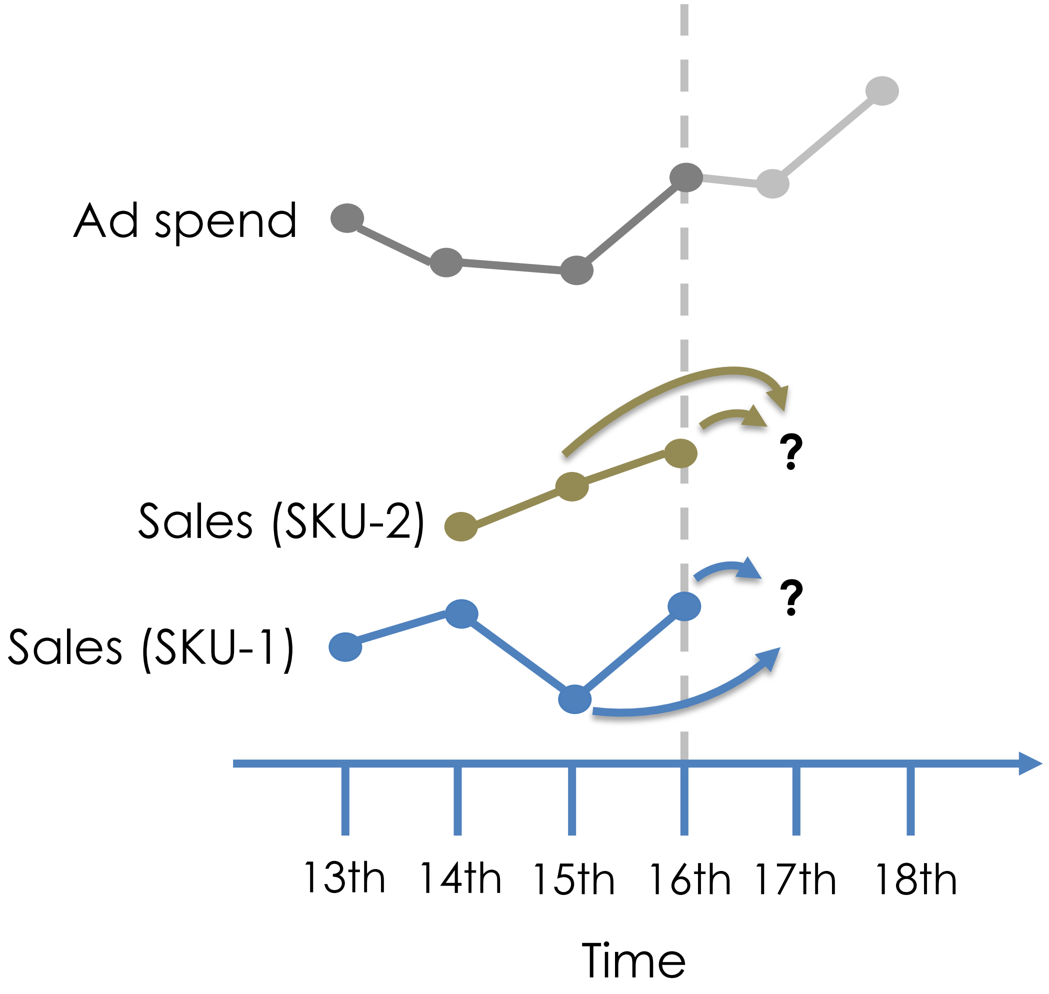
y_t
...
30
32
25
34
?
...
32
21
25
?

Lag features: Past values of target & features

- Recent values of the target are likely to be predictive:

$$\hat{y}_t \leftarrow y_{t-1}, y_{t-2}, \ldots y_{t-k}$$

- Seasonal lags good for seasonality (e.g., lag of 7 for weekly seasonality).



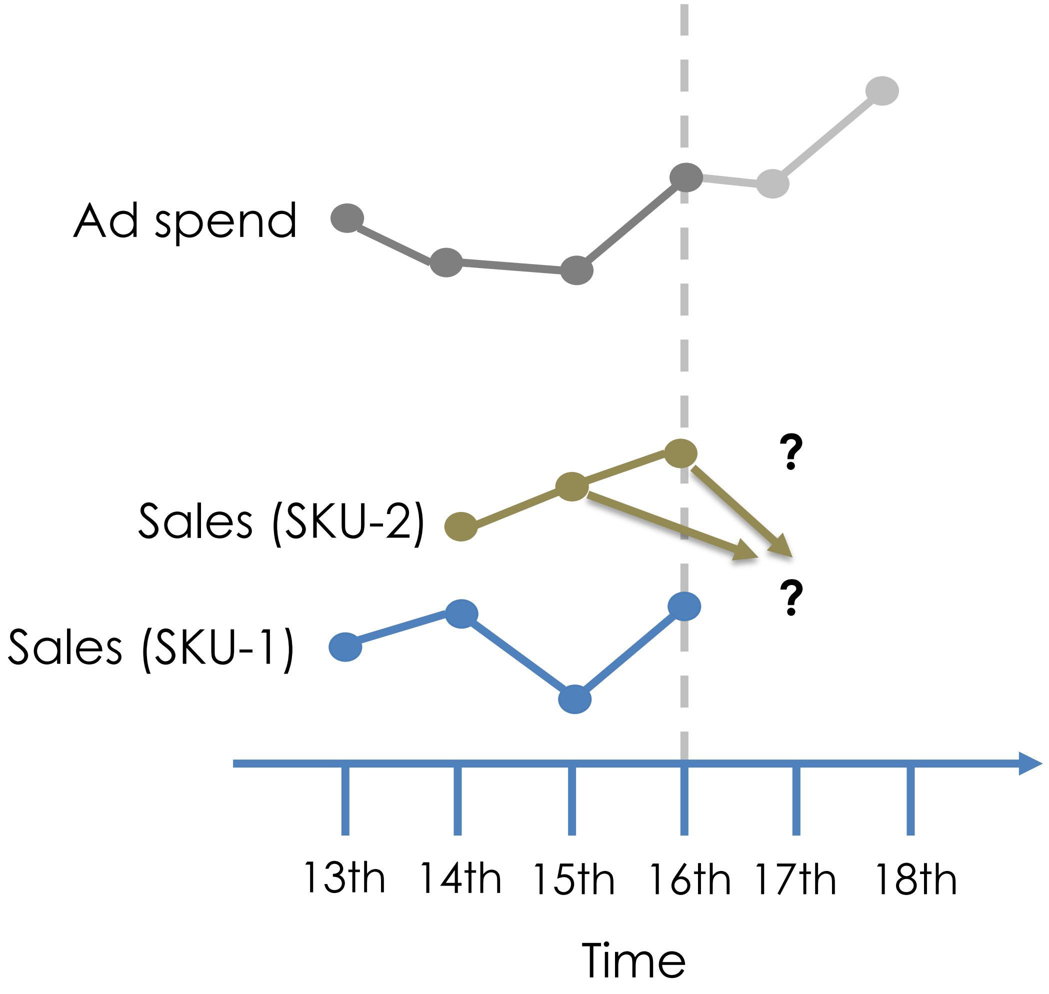
Lag features: Past values of target & features

- Recent values of the target are likely to be predictive:

$$\hat{y}_t \leftarrow y_{t-1}, y_{t-2}, \dots y_{t-k}$$

- Seasonal lags good for seasonality (e.g., lag of 7 for weekly seasonality).

- Can use lags of other target time series.

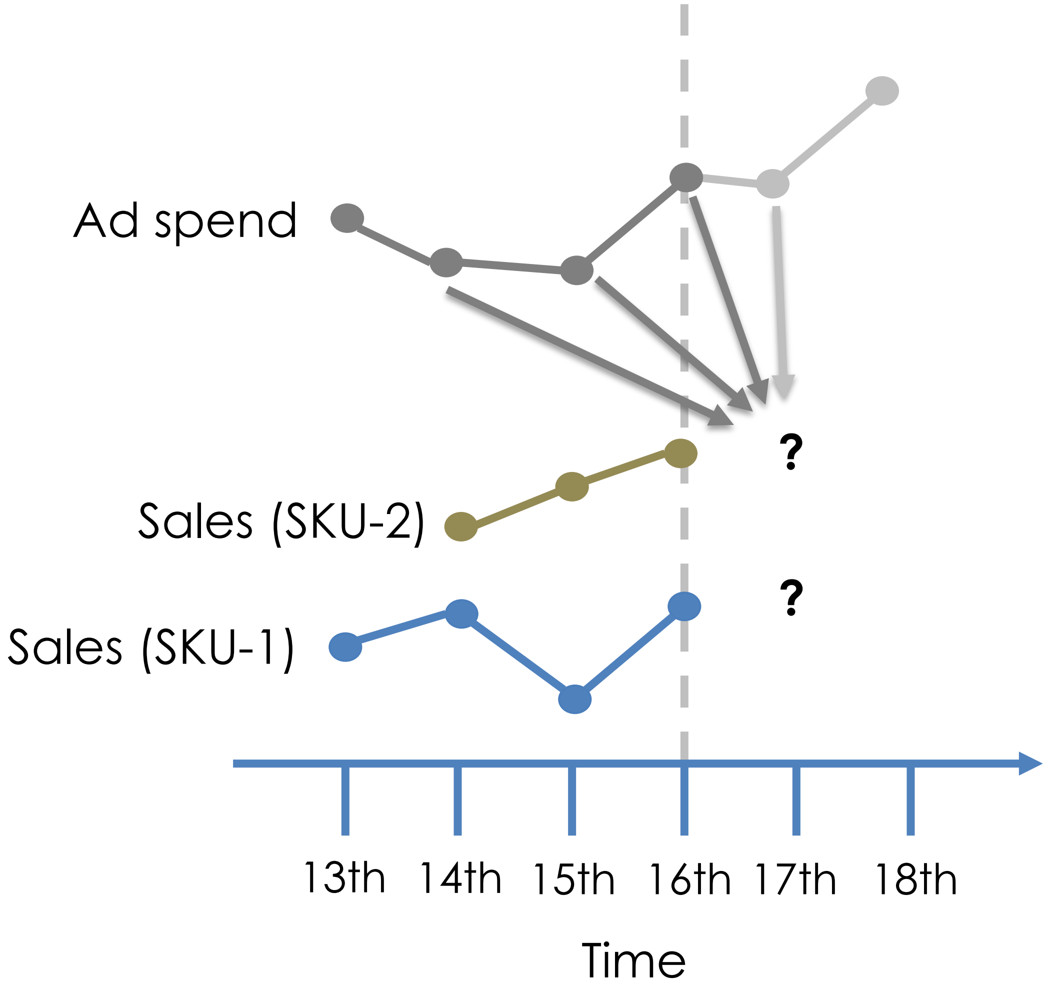


Lag features: Past values of target & features

- Recent values of the target are likely to be predictive:

$$\hat{y}_t \leftarrow y_{t-1}, y_{t-2}, \dots y_{t-k}$$

- Seasonal lags good for seasonality (e.g., lag of 7 for weekly seasonality).
- Can use lags of other target time series.
- Lag values of exogenous feature could also be predictive (e.g., distributed lags).



Lag features: Past values of target & features

Time	Product ID	Ad spend	Ad spend Lag 1	Ad spend Lag 2	y_{t-3}	y_{t-2}	y_{t-1}	y_t
…	…	…						…
2020-02-13	SKU-1	100						30
2020-02-14	SKU-1	120						**32**
2020-02-15	SKU-1	**110**						**25**
2020-02-16	SKU-1	**101**						**34**
2020-02-17	SKU-1	102	**101**	**110**	**32**	**25**	**34**	?
…	…	…	…	…	…	…	…	…
2020-02-14	SKU-2	120						**32**
2020-02-15	SKU-2	**110**						**21**
2020-02-16	SKU-2	**101**						**25**
2020-02-17	SKU-2	102	**110**	**110**	**32**	**21**	**25**	?

Lag features: Past values of target & features

Time	Product ID	Ad spend	Ad spend Lag 1	Ad spend Lag 2	y_{t-3}	y_{t-2}	y_{t-1}	y_t
…	…	…						…
2020-02-13	SKU-1	100						**30**
2020-02-14	SKU-1	**120**						**32**
2020-02-15	SKU-1	**110**						**25**
2020-02-16	SKU-1	101	**110**	**120**	**30**	**32**	**25**	34
2020-02-17	SKU-1	102	101	110	32	25	34	?
…	…	…	…	…	…	…	…	**…**
2020-02-14	SKU-2	**120**						**32**
2020-02-15	SKU-2	**110**						**21**
2020-02-16	SKU-2	101	**110**	**120**	**…**	**32**	**21**	25
2020-02-17	SKU-2	102	101	110	32	21	25	?

Lag features: Past values of target & features

Time	Product ID	Ad spend	Ad spend Lag 1	Ad spend Lag 2	y_{t-3}	y_{t-2}	y_{t-1}	y_t
...	...	...	...	...	...	...	...	...
2020-02-13	SKU-1	100	...	...	...	...	...	30
2020-02-14	SKU-1	120	100	...	...	...	30	32
2020-02-15	SKU-1	110	120	100	...	30	32	25
2020-02-16	SKU-1	101	110	120	30	32	25	34
2020-02-17	SKU-1	102	101	110	32	25	34	?
...	...	...	...	...	...	...	...	...
2020-02-14	SKU-2	120	100	...	...	...	...	32
2020-02-15	SKU-2	110	120	100	...	...	32	21
2020-02-16	SKU-2	101	110	120	...	32	21	25
2020-02-17	SKU-2	102	101	110	32	21	25	?

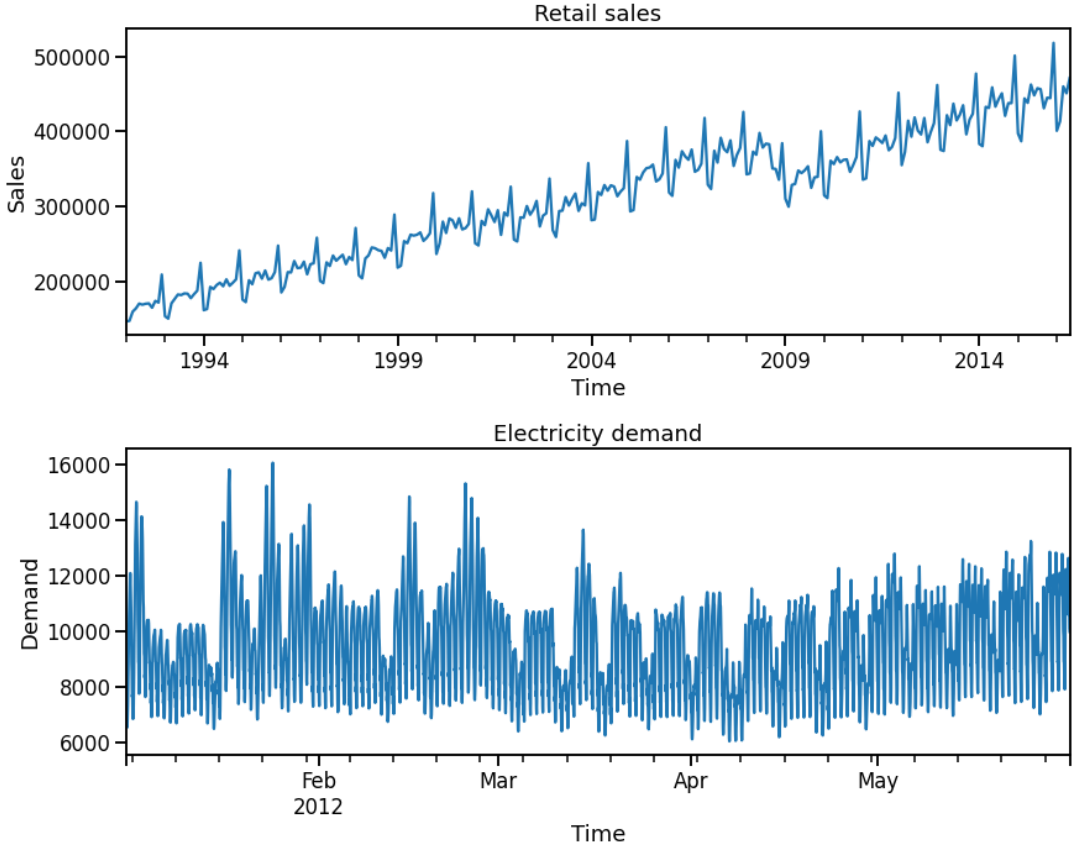
How to choose the lags

- Domain knowledge
- Feature selection and modelling
- Time-series correlation methods

Domain knowledge

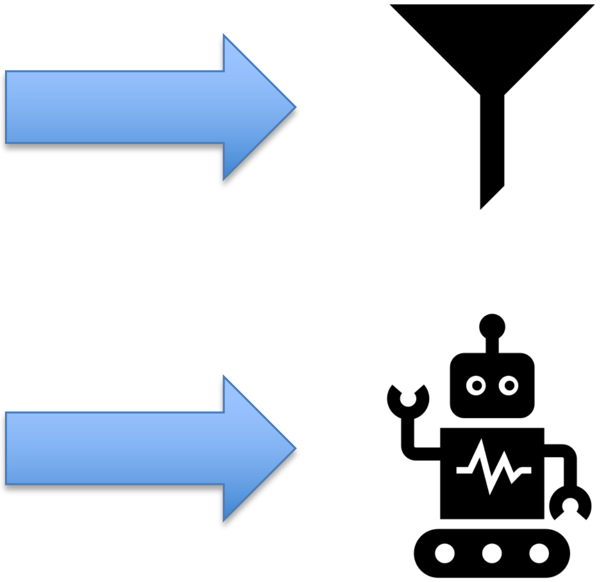
When lagging the target

- If seasonality is known use a lag of the same seasonal order (aka seasonal lag).
- Example retail sales: yearly seasonality ➔ use lag of 1 year.
- Example electricity demand: multiple seasonalities such as yearly, weekly, and daily ➔ use lag of 1 year, 1 week, and 1 day.
- Most recent values tend to be predictive ➔ use recent lags.



Feature selection and modelling

Sales Lag 1	Sales Lag 3	Ad spend Lag 1	Ad spend Lag 2
NaN	NaN	NaN	NaN
23	NaN	100	NaN
30	NaN	120	100
35	23	90	120
30	30	80	90



- Create a bunch of different lags which are reasonable given the feature and use case (e.g., ad spend more than 1 year ago unlikely to be help for sales forecasting).
- Use feature selection and/or modelling (e.g., LASSO) to best utilize the features and determine a subset which minimizes forecast error.

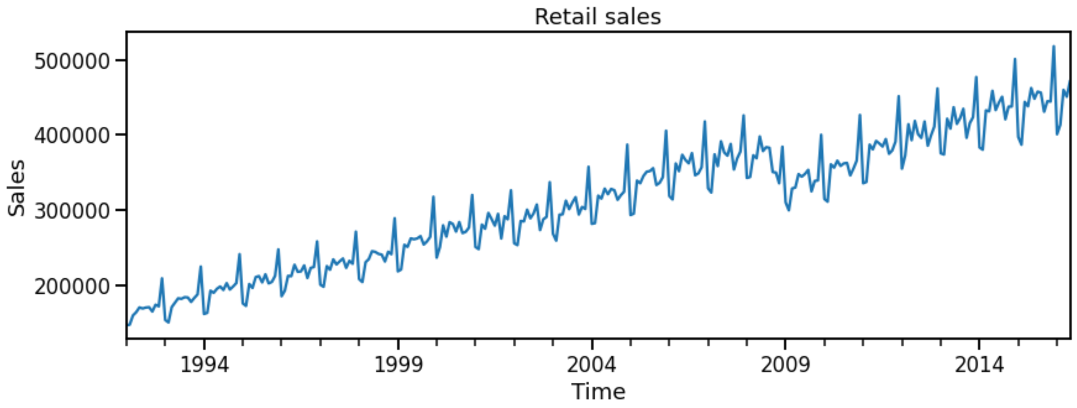
Time-series correlation methods

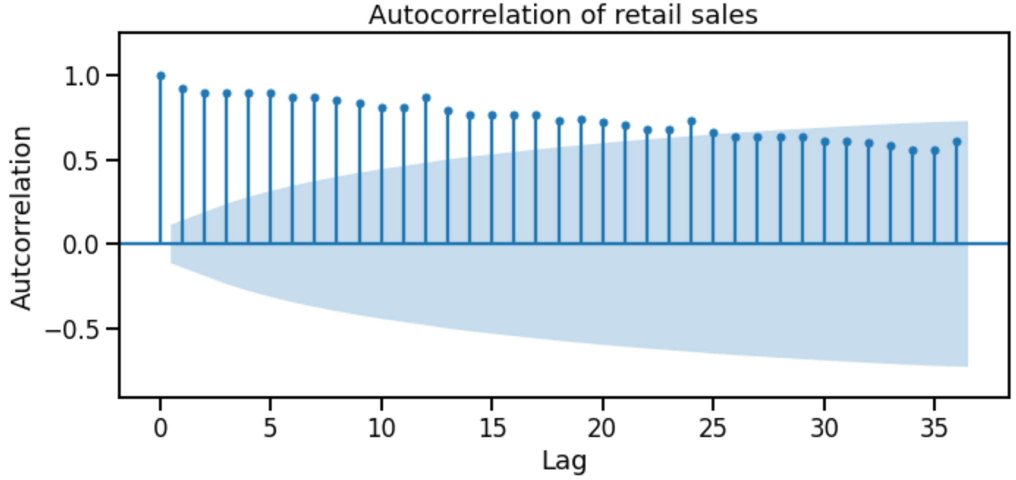
The main idea

- Measure how correlated the lag features are with the target.
- If the lag feature is highly correlated to the the target then it might be helpful.

Three main methods

- Autocorrelation function (ACF)
- Partial autocorrelation function (PACF)
- Cross-correlation function (CCF)



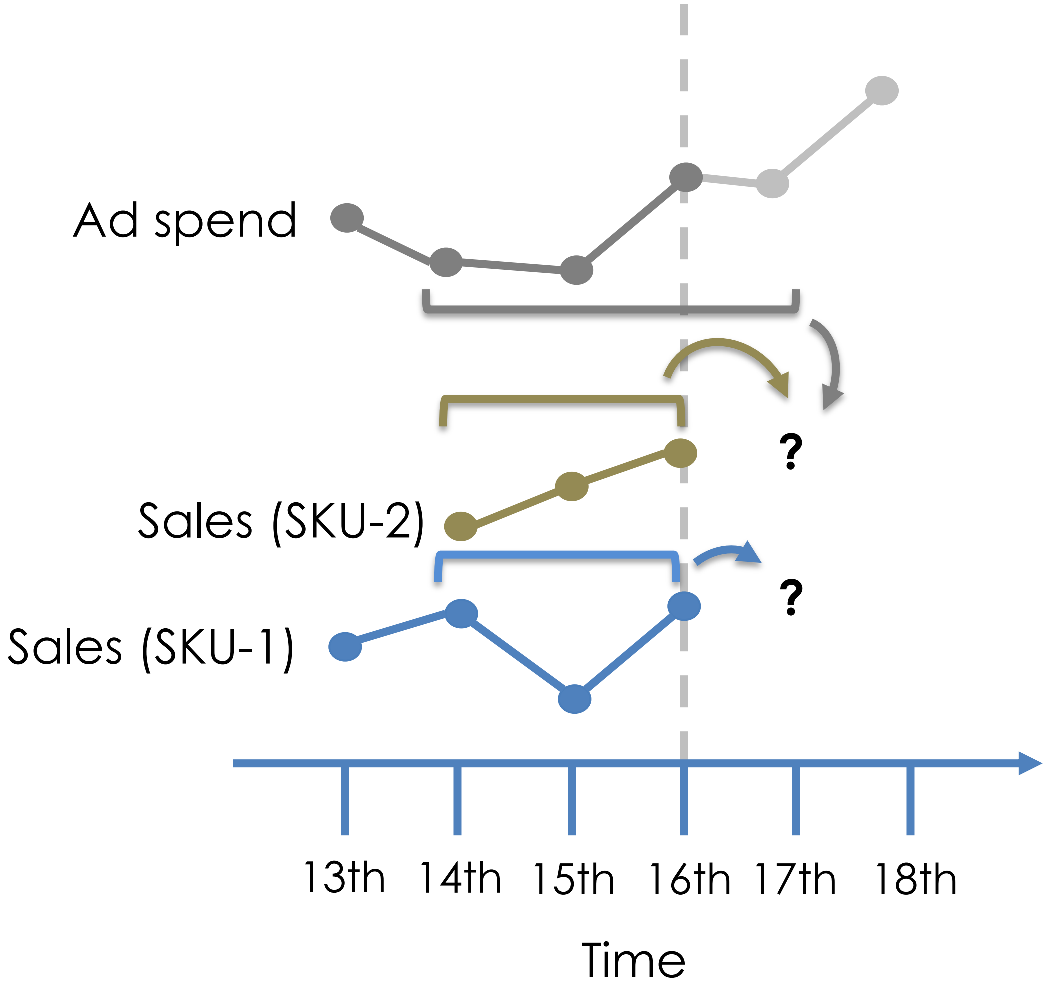


Window features: Function over a past window

- Compute a summary statistic over a window of past data

$$\hat{y}_t \leftarrow f(y_{t-1}, y_{t-2}, \dots y_{t-k})$$

- Mean & standard deviation are common.



Window features: Function over a past window

Time	Product ID	Ad spend	Rolling std y_{t-1}	Rolling mean y_{t-1}	y_t
…	…	…	…	…	…
2020-02-13	SKU-1	100			**30**
2020-02-14	SKU-1	120			**32**
2020-02-15	SKU-1	110			**25**
2020-02-16	SKU-1	101	**2.94**	**29.0**	34
2020-02-17	SKU-1	102			?
…	…	…	…	…	…
2020-02-14	SKU-2	120			32
2020-02-15	SKU-2	110			21
2020-02-16	SKU-2	101			25
2020-02-17	SKU-2	102			?

Window features: Function over a past window

Time	Product ID	Ad spend	Rolling std y_{t-1}	Rolling mean y_{t-1}	y_t
…	…	…	…	…	…
2020-02-13	SKU-1	100			30
2020-02-14	SKU-1	120			**32**
2020-02-15	SKU-1	110			**25**
2020-02-16	SKU-1	101	2.94	29.0	**34**
2020-02-17	SKU-1	102	**3.85**	**30.3**	?
…	…	…	…	…	…
2020-02-14	SKU-2	120			32
2020-02-15	SKU-2	110			21
2020-02-16	SKU-2	101			25
2020-02-17	SKU-2	102			?

Window features: Nested window features

Time	Product ID	Ad spend	y_t
…	…	…	…
2020-02-13	SKU-1	100	30
2020-02-14	SKU-1	120	32
2020-02-15	SKU-1	110	25
2020-02-16	SKU-1	101	34
2020-02-17	SKU-1	102	?
…	…	…	…
2020-02-14	SKU-2	120	32
2020-02-15	SKU-2	110	21
2020-02-16	SKU-2	101	25
2020-02-17	SKU-2	102	?

Months

Weeks

Days

Window features: Nested window features

Time	Product ID	Ad spend	Rolling mean (months)	Rolling mean (weeks)	Rolling mean (days)	y_t
...	...	...				...
2020-02-13	SKU-1	100				30
2020-02-14	SKU-1	120				32
2020-02-15	SKU-1	110				25
2020-02-16	SKU-1	101				34
2020-02-17	SKU-1	102				?
...	...	...				...
2020-02-14	SKU-2	120				32
2020-02-15	SKU-2	110				21
2020-02-16	SKU-2	101				25
2020-02-17	SKU-2	102				?

Static features

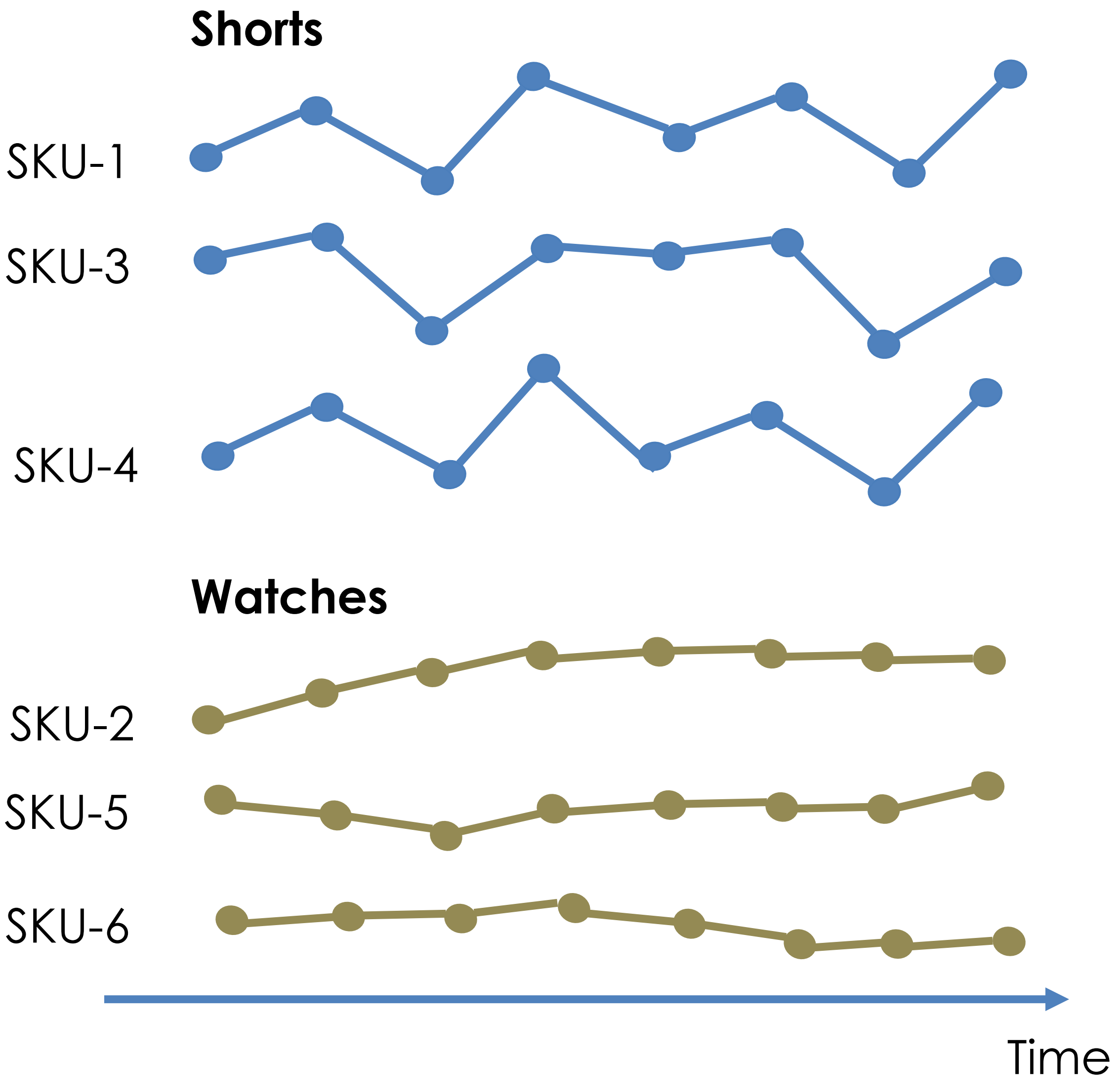
Time	Product ID	y_t
…	…	…
2020-02-13	SKU-1	30
2020-02-14	SKU-1	32
2020-02-15	SKU-1	25
2020-02-16	SKU-1	34
2020-02-17	SKU-1	?
…	…	…
2020-02-14	SKU-2	32
2020-02-15	SKU-2	21
2020-02-16	SKU-2	25
2020-02-17	SKU-2	?

Static features

Time	Product ID	Product category	y_t
…	…	…	…
2020-02-13	SKU-1	Shorts	30
2020-02-14	SKU-1	Shorts	32
2020-02-15	SKU-1	Shorts	25
2020-02-16	SKU-1	Shorts	34
2020-02-17	SKU-1	Shorts	?
…	…	…	…
2020-02-14	SKU-2	Watches	32
2020-02-15	SKU-2	Watches	21
2020-02-16	SKU-2	Watches	25
2020-02-17	SKU-2	Watches	?

Static features

Time	Product ID	Product category
…	…	…
2020-02-13	SKU-1	Shorts
2020-02-14	SKU-1	Shorts
2020-02-15	SKU-1	Shorts
2020-02-16	SKU-1	Shorts
2020-02-17	SKU-1	Shorts
…	…	…
2020-02-14	SKU-2	Watches
2020-02-15	SKU-2	Watches
2020-02-16	SKU-2	Watches
2020-02-17	SKU-2	Watches



Static features: Target encoding

Time	Product ID
…	…
2020-02-13	SKU-1
2020-02-14	SKU-1
2020-02-15	SKU-1
2020-02-16	SKU-1
2020-02-17	SKU-1
…	…
2020-02-14	SKU-2
2020-02-15	SKU-2
2020-02-16	SKU-2
2020-02-17	SKU-2

y_t
…
30
32
25
34
?
…
32
21
25
?

SKU-1 = mean

SKU-2 = mean

Static features: Target encoding

Time	Product ID (encoded)
…	…
2020-02-13	30
2020-02-14	30
2020-02-15	30
2020-02-16	30
2020-02-17	30
…	…
2020-02-14	24
2020-02-15	24
2020-02-16	24
2020-02-17	24

- No leakage between **train** and **test set**.
- But, the **target is leaked** from future values to past values in training set.
- Beware of overfitting.

SKU-1

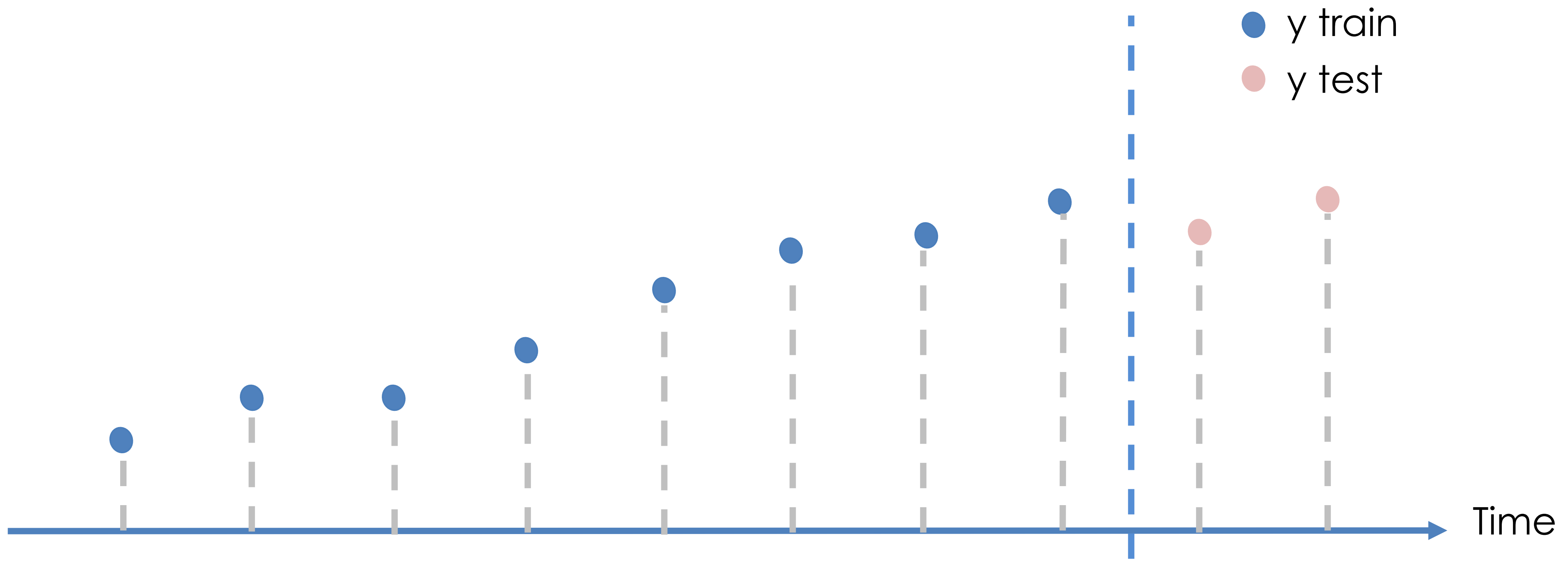
30 = mean

SKU-2

24 = mean

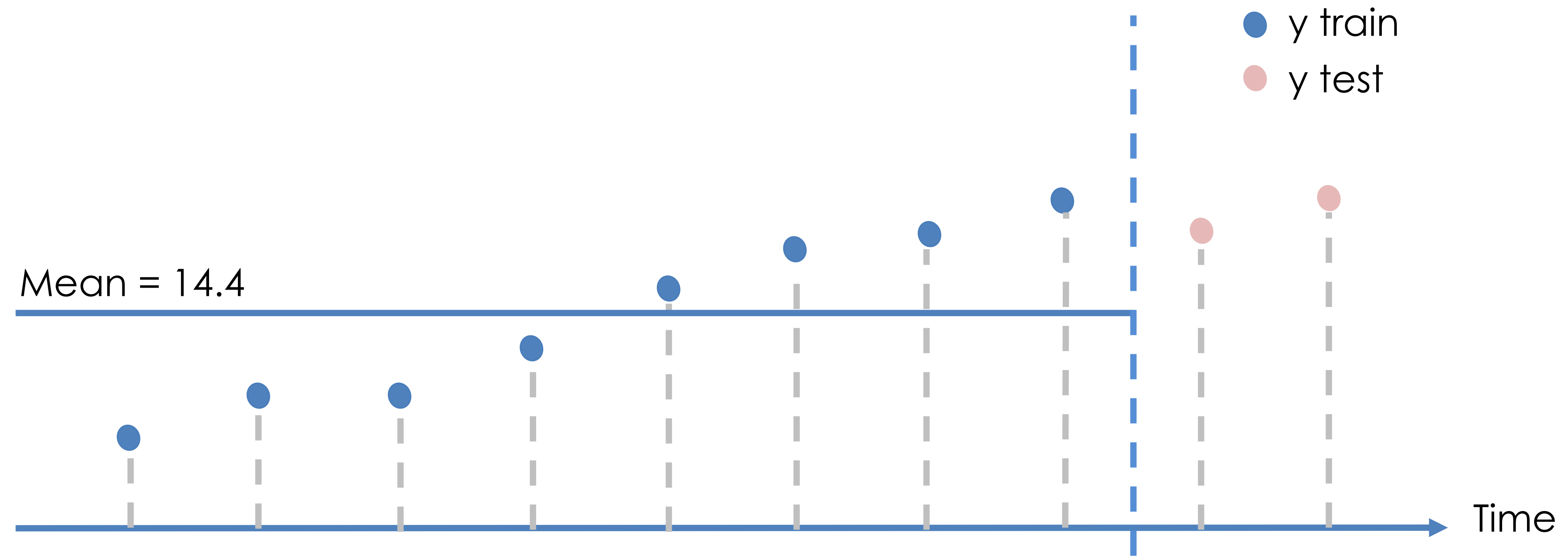
y_t
…
30
32
25
34
?
…
32
21
25
?

Static features: Target encoding



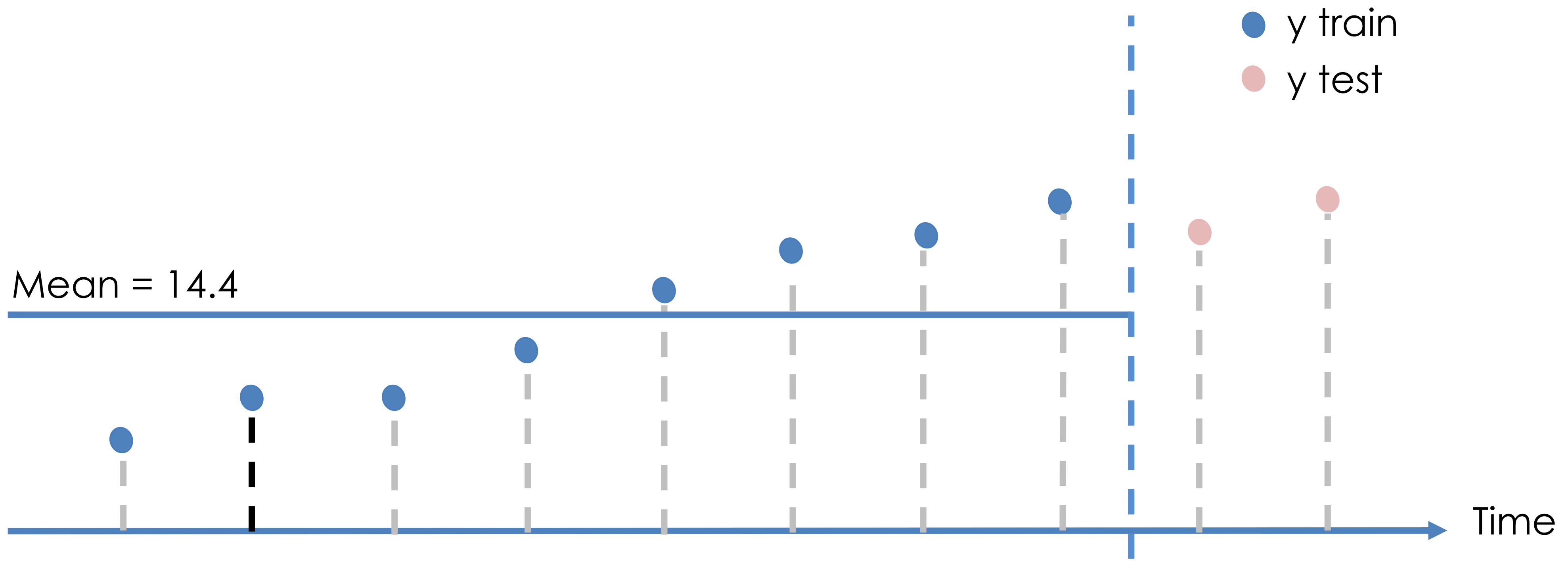
y	10	12	11	13	15	17	18	19	18	19
Product ID	SKU-1	SKU-1	SKU-1	SKU-1	SKU-1	SKU-1	SKU-1	SKU-1	SKU-1	SKU-1
Time	1	2	3	4	5	6	7	8	9	10

Static features: Target encoding



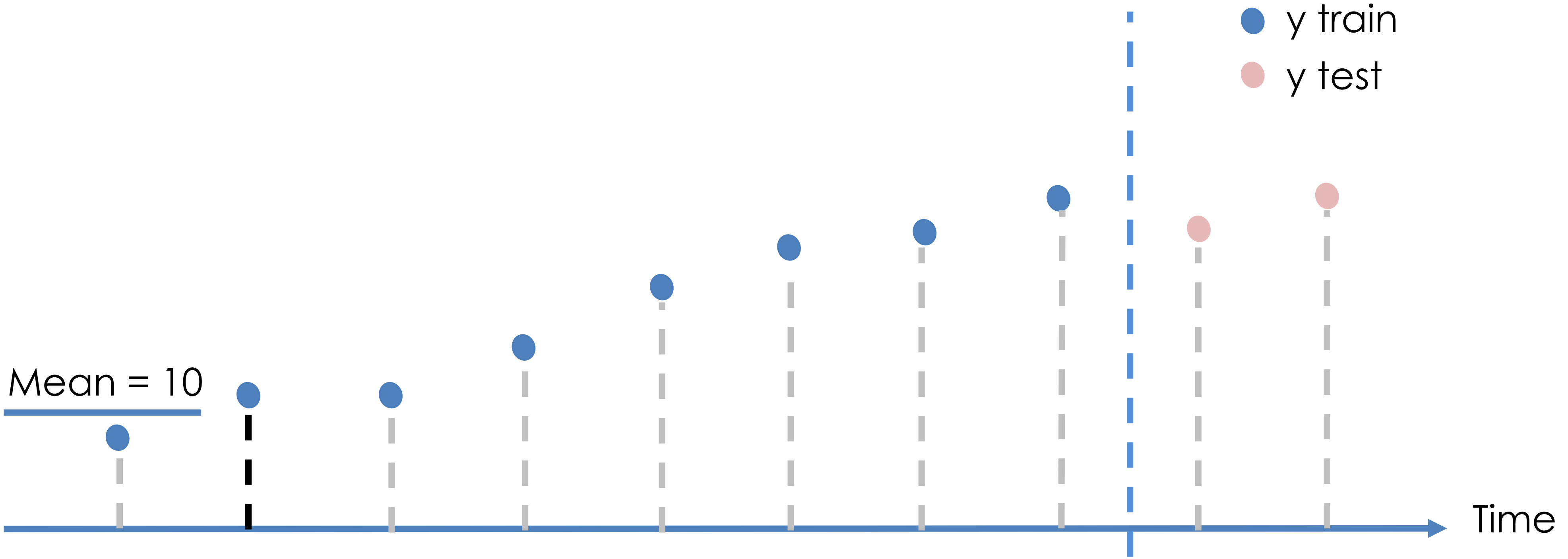
y	10	12	11	13	15	17	18	19	18	19
Product ID	14.4	14.4	14.4	14.4	14.4	14.4	14.4	14.4	14.4	14.4
Time	1	2	3	4	5	6	7	8	9	10

Static features: Target encoding



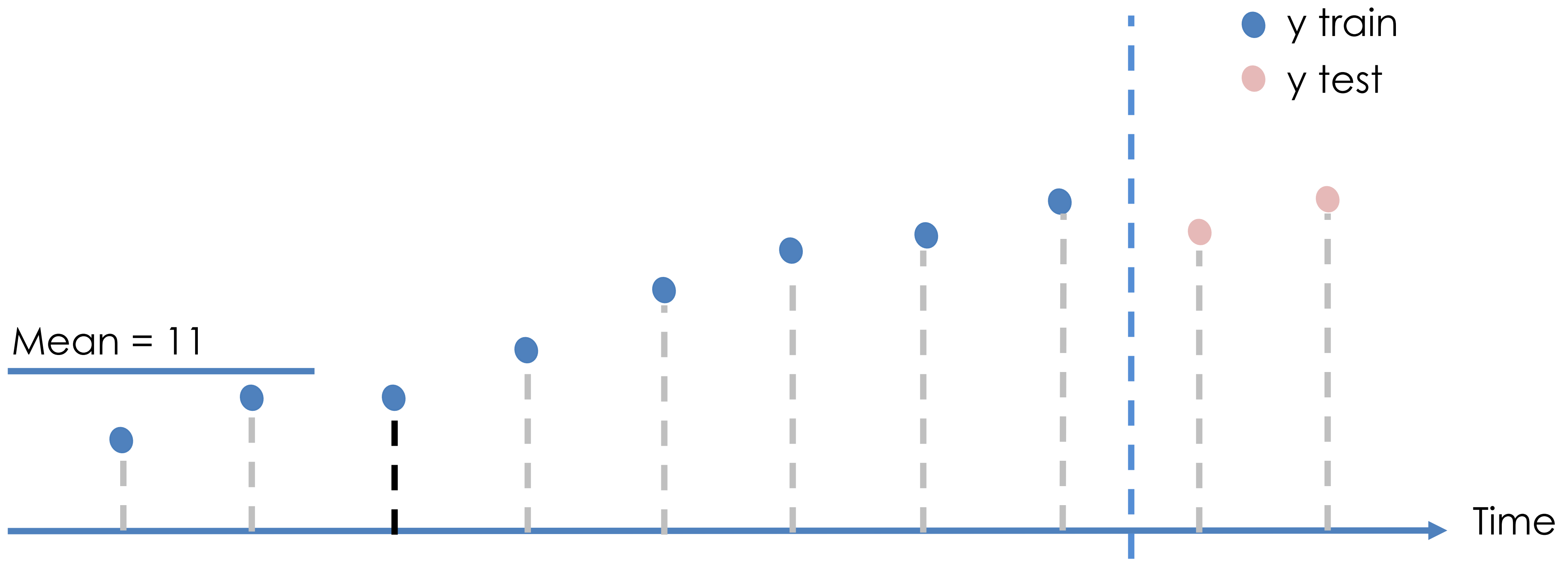
y	10	**12**	11	13	15	17	18	19	18	19
Product ID	14.4	**14.4**	14.4	14.4	14.4	14.4	14.4	14.4	14.4	14.4
Time	1	**2**	3	4	5	6	7	8	9	10

Static features: Target encoding



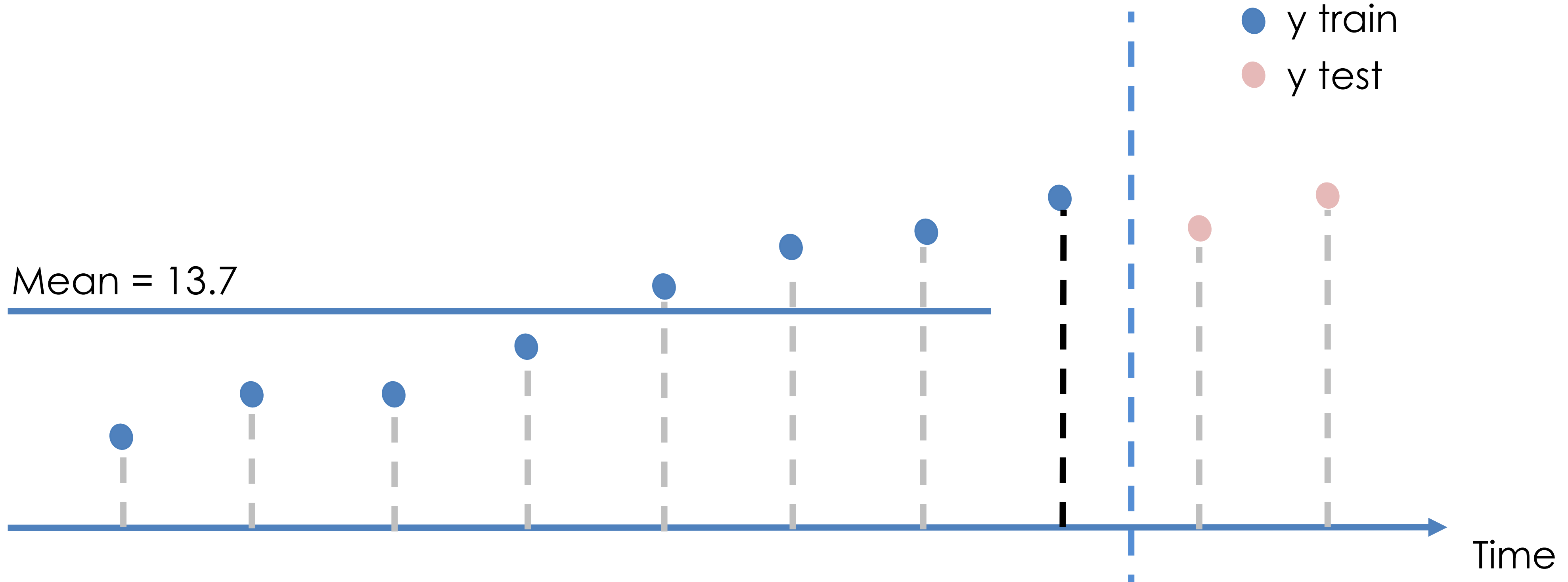
y	**10**	**12**	11	13	15	17	18	19	18	19
Product ID	SKU-1	**10.0**	SKU-1	SKU-1	SKU-1	SKU-1	SKU-1	SKU-1	SKU-1	SKU-1
Time	1	**2**	3	4	5	6	7	8	9	10

Static features: Target encoding



y	**10**	**12**	**11**	13	15	17	18	19	18	19
Product ID	SKU-1	10.0	**11.0**	SKU-1	SKU-1	SKU-1	SKU-1	SKU-1	SKU-1	SKU-1
Time	1	2	**3**	4	5	6	7	8	9	10

Static features: Target encoding



y	**10**	**12**	**11**	**13**	**15**	**17**	**18**	**19**	18	19
Product ID	NaN	10.0	11.0	11.0	11.5	12.2	13.0	**13.7**	SKU-1	SKU-1
Time	1	2	3	4	5	6	7	**8**	9	10

Static features: Target encoding

One step ahead-forecast

SKU-1

30 = mean

SKU-2

24 = mean

Time	Product ID (encoded)	y_t
…	…	…
2020-02-13	29	**30**
2020-02-14	31	**32**
2020-02-15	30	**25**
2020-02-16	28	**34**
2020-02-17	**30**	?
2020-02-18	SKU-1	?
…	…	…
2020-02-14	24	**32**
2020-02-15	26	**21**
2020-02-16	23	**25**
2020-02-17	**24**	?
2020-02-18	SKU-2	?

Static features: Target encoding

Time	Product ID (encoded)	y_t
...	...	...
2020-02-13	29	**30**
2020-02-14	31	**32**
2020-02-15	30	**25**
2020-02-16	28	**34**
2020-02-17	**30**	?
2020-02-18	SKU-1	?
...	...	...
2020-02-14	24	**32**
2020-02-15	26	**21**
2020-02-16	23	**25**
2020-02-17	**24**	?
2020-02-18	SKU-2	?

For **recursive forecasting**, we need to dynamically compute the encoding at predict time.

SKU-1

30 = mean

SKU-2

24 = mean

Static features: Target encoding

Time	Product ID (encoded)	y_t
…	…	…
2020-02-13	29	**30**
2020-02-14	31	**32**
2020-02-15	30	**25**
2020-02-16	28	**34**
2020-02-17	30	$\hat{y}_{T+1}^{S1}$
2020-02-18	**SKU-1**	?
…	…	…
2020-02-14	24	**32**
2020-02-15	26	**21**
2020-02-16	23	**25**
2020-02-17	24	$\hat{y}_{T+1}^{S2}$
2020-02-18	**SKU-2**	?

For **recursive forecasting**, we need to dynamically compute the encoding at predict time.

SKU-1 = mean

SKU-2 = mean

Key takeaways

- Data leakage is a risk when creating features from the target and future-unknown variables.
- Only use data that is known at the time of the target.
- Handling features at predict time can differ for direct vs recursive forecasting.

Time	Product ID	Ad spend	Sales
…	…	…	…
2020-02-13	SKU-1	100	30
2020-02-14	SKU-1	120	32
2020-02-15	SKU-1	110	25
2020-02-16	SKU-1	101	34
2020-02-17	SKU-1	x_{T+1}	?
2020-02-18	SKU-1	x_{T+2}	?
…	…	…	…
2020-02-14	SKU-2	120	32
2020-02-15	SKU-2	110	21
2020-02-16	SKU-2	101	15
2020-02-17	SKU-2	x_{T+1}	?
2020-02-18	SKU-2	x_{T+2}	?

Contents

Overview of some useful libraries

Feature engineering & pre-processing

tsfresh

- Large number of time series features.
- Supports ML forecasting workflows.
- Multiple time series.

Feature-engine

- Create time series features on top of Pandas.

Forecasting workflow using ML on tabular data

Darts

- Direct, Recursive, & mixed strategies.
- Some helper methods to create features.
- Multiple time series.
- Exogenous features.
- Time series cross-val.

SKTIME

- Direct, recursive, DirRec, & multi-output strategies.
- Lots of transformers for feature engineering
- Multiple time series.
- Exogenous features.
- Time series cross-val.

Forecasting with tabular data using Darts

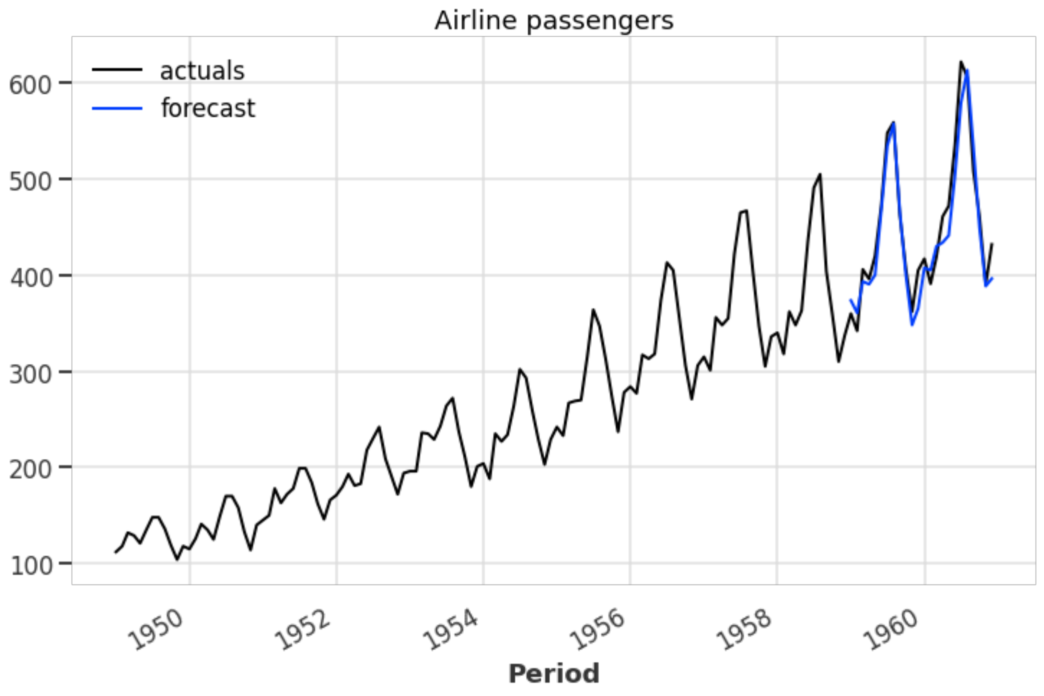
```python
from darts import TimeSeries
from darts.models import RegressionModel
from sklearn.linear_model import LinearRegression

# Convert pandas DataFrame to TimeSeries
y = TimeSeries.from_series(df['y'])

# Hold-out last 24 data points
y_train = y[:-24]

# Specify and train model
model = RegressionModel(
                        lags=[-1, -2, -12],
                        model=LinearRegression()
                       )

model.fit(series=y_train)

# Forecast
y_pred = model.predict(n=24, series=y_train)
```

	y
Period	
1949-01-01	112.0
1949-02-01	118.0
1949-03-01	132.0
1949-04-01	129.0
1949-05-01	121.0
...	...
1960-08-01	606.0
1960-09-01	508.0
1960-10-01	461.0
1960-11-01	390.0
1960-12-01	432.0

Recursive forecast using linear regression, single time series, and only lag features.

Forecasting with tabular data using Darts

```
from darts import TimeSeries
from darts.models import RegressionModel
from sklearn.linear_model import LinearRegression

# Convert pandas DataFrame to TimeSeries
y = TimeSeries.from_series(df['y'])

# Hold-out last 24 data points
y_train = y[:-24]

# Specify and train model
model = RegressionModel(
                        lags=[-1, -2, -12],
                        model=LinearRegression()
                       )

model.fit(series=y_train)

# Forecast
y_pred = model.predict(n=24, series=y_train)
```



Recursive forecast using linear regression, single time series, and only lag features.

Forecasting with tabular data using Darts

```
# Convert pandas DataFrame to TimeSeries
y = TimeSeries.from_series(df['y'])
features = ['ad_spend', 'month','year']
future_cov = TimeSeries.from_dataframe(df[features])
```

period	y	ad_spend	month	year
2020-01-01	100	25.51	1	2020
2020-01-02	103	208.82	1	2020
2020-01-03	105	44.47	1	2020
2020-01-04	107	11.96	1	2020
2020-01-05	106	145.02	1	2020
...	...	...	...	...
2020-04-05	311	16.97	4	2020
2020-04-06	341	62.60	4	2020
2020-04-07	358	16.93	4	2020
2020-04-08	377	19.31	4	2020
2020-04-09	385	110.41	4	2020

Recursive forecast using linear regression, single time series, and with lag & future-known features.

Forecasting with tabular data using Darts

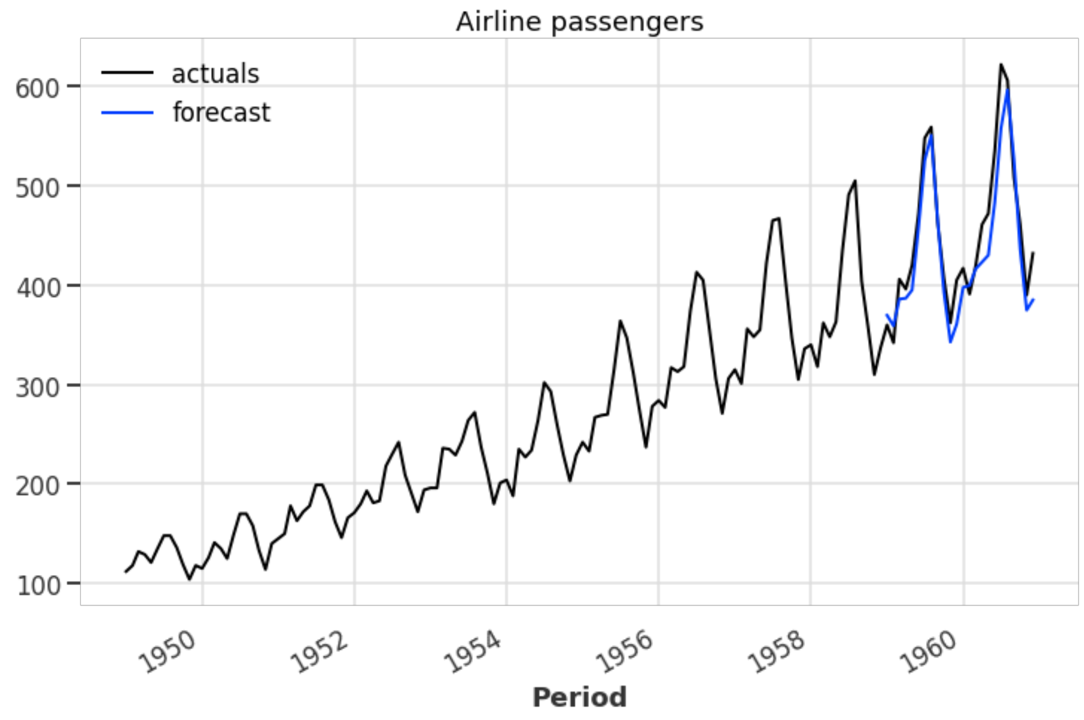
```
# Specify and train model
model = RegressionModel(
                        lags=[-1, -2, -12],
                        lags_future_covariates=[0],
                        model=LinearRegression()
                       )

model.fit(y_train, future_covariates=future_cov)

# Forecast
y_pred = model.predict(
                       n=24,
                       series=y_train,
                       future_covariates=future_cov
                      )
```

	y	ad_spend	month	year
period				
2020-01-01	100	25.51	1	2020
2020-01-02	103	208.82	1	2020
2020-01-03	105	44.47	1	2020
2020-01-04	107	11.96	1	2020
2020-01-05	106	145.02	1	2020
...	...	...	...	...
2020-04-05	311	16.97	4	2020
2020-04-06	341	62.60	4	2020
2020-04-07	358	16.93	4	2020
2020-04-08	377	19.31	4	2020
2020-04-09	385	110.41	4	2020

Recursive forecast using linear regression, single time series, and with lag & future-known features.

Forecasting with tabular data using Darts

```python
# Specify and train model
model = RegressionModel(
                        lags=[-1, -2, -12],
                        lags_future_covariates=[0],
                        model=LinearRegression()
                       )

model.fit(y_train, future_covariates=future_cov)

# Forecast
y_pred = model.predict(
                      n=24,
                      series=y_train,
                      future_covariates=future_cov
                      )
```



Recursive forecast using linear regression, single time series, and with lag & future-known features.

Forecasting with tabular data using Darts

```
# Convert pandas DataFrame to TimeSeries
y_all = TimeSeries.from_group_dataframe(df,
                                        group_cols=['country', 'product_id'],
                                        time_col='period',
                                        value_cols=['y'])
#     = [y1, y2, y3, ...]

future_cov_all = TimeSeries.from_group_dataframe(df,
                                        group_cols=['country', 'product_id'],
                                        time_col='period',
                                        value_cols=['month', 'year', 'ad_spend'])
#              = [X1, X2, X3, ...]
```

	period	country	product_id	y	ad_spend	month	year
0	2020-01-01	UK	SKU-1	100	53.61	1	2020
1	2020-01-02	UK	SKU-1	103	41.32	1	2020
2	2020-01-03	UK	SKU-1	105	65.91	1	2020
3	2020-01-04	UK	SKU-1	107	46.71	1	2020
4	2020-01-05	UK	SKU-1	106	15.58	1	2020
...	...	...	...	...	...	...	...
35	2020-04-05	Germany	SKU-2	319	39.85	4	2020
36	2020-04-06	Germany	SKU-2	329	2.60	4	2020
37	2020-04-07	Germany	SKU-2	369	101.07	4	2020
38	2020-04-08	Germany	SKU-2	365	3.29	4	2020
39	2020-04-09	Germany	SKU-2	390	11.68	4	2020

Recursive forecast using linear regression, multiple time series, and with lag & future-known features.

Forecasting with tabular data using Darts

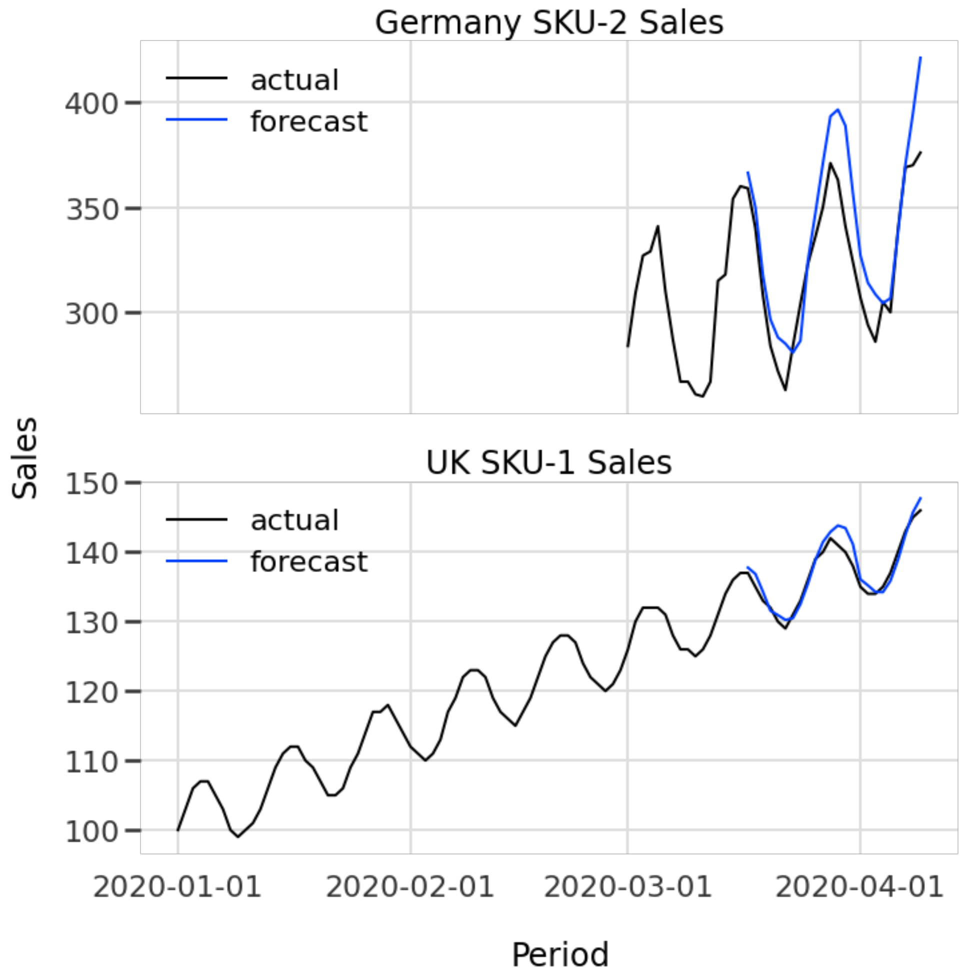
```
# Specify and train model
model = RegressionModel(
                        lags=[-1, -2, -12],
                        lags_future_covariates=[0],
                        model=LinearRegression()
                       )

model.fit(y_train_all, future_covariates=future_cov_all)

# Forecast any subset of time series
y_pred = model.predict(n=24,
                       series=y_train_all[:2],
                       future_covariates=future_cov_all[:2])
```

	period	country	product_id	y	ad_spend	month	year
0	2020-01-01	UK	SKU-1	100	53.61	1	2020
1	2020-01-02	UK	SKU-1	103	41.32	1	2020
2	2020-01-03	UK	SKU-1	105	65.91	1	2020
3	2020-01-04	UK	SKU-1	107	46.71	1	2020
4	2020-01-05	UK	SKU-1	106	15.58	1	2020
...	...	...	...	...	...	...	...
35	2020-04-05	Germany	SKU-2	319	39.85	4	2020
36	2020-04-06	Germany	SKU-2	329	2.60	4	2020
37	2020-04-07	Germany	SKU-2	369	101.07	4	2020
38	2020-04-08	Germany	SKU-2	365	3.29	4	2020
39	2020-04-09	Germany	SKU-2	390	11.68	4	2020

Recursive forecast using linear regression, multiple time series, and with lag & future-known features.

Forecasting with tabular data using Darts

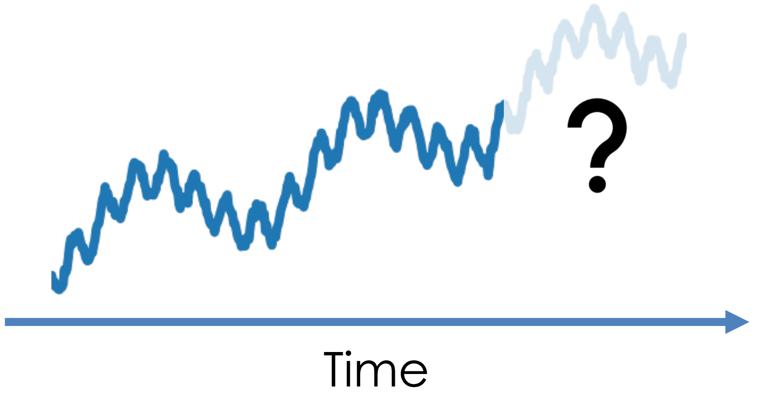
```
# Specify and train model
model = RegressionModel(
                        lags=[-1, -2, -12],
                        lags_future_covariates=[0],
                        model=LinearRegression()
                       )

model.fit(y_train_all, future_covariates=future_cov_all)

# Forecast any subset of time series
y_pred = model.predict(n=24,
                       series=y_train_all[:2],
                       future_covariates=future_cov_all[:2])
```



Recursive forecast using linear regression, multiple time series, and with lag & future-known features.

Conclusions

1. Forecasting can be treated as a tabular ML task and compete with statistical models.

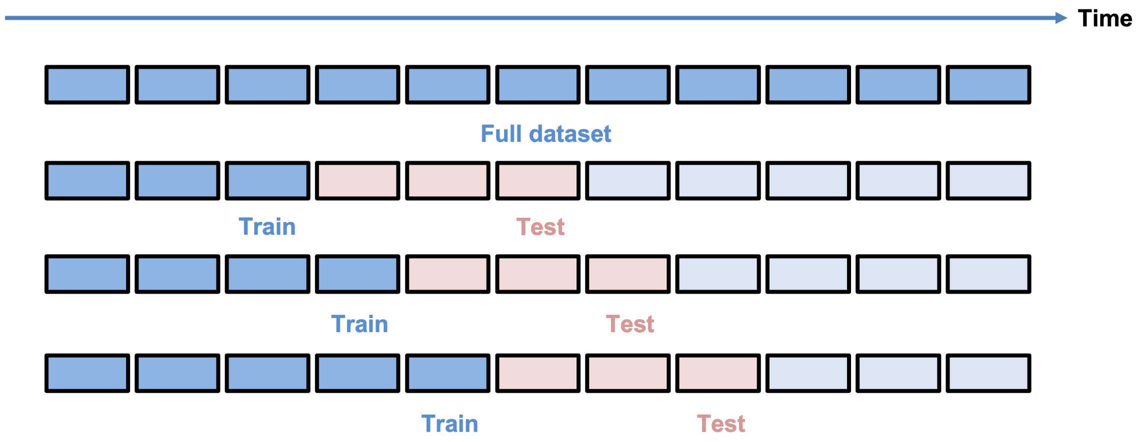


Time	Sales (UK)
2020-02-12	35
2020-02-13	30
2020-02-14	23
2020-02-15	?
2020-02-16	?
2020-02-17	?

Conclusions

1. Forecasting can be treated as a tabular ML task and compete with statistical models.

2. The feature engineering and ML workflow is different for time series forecasting.

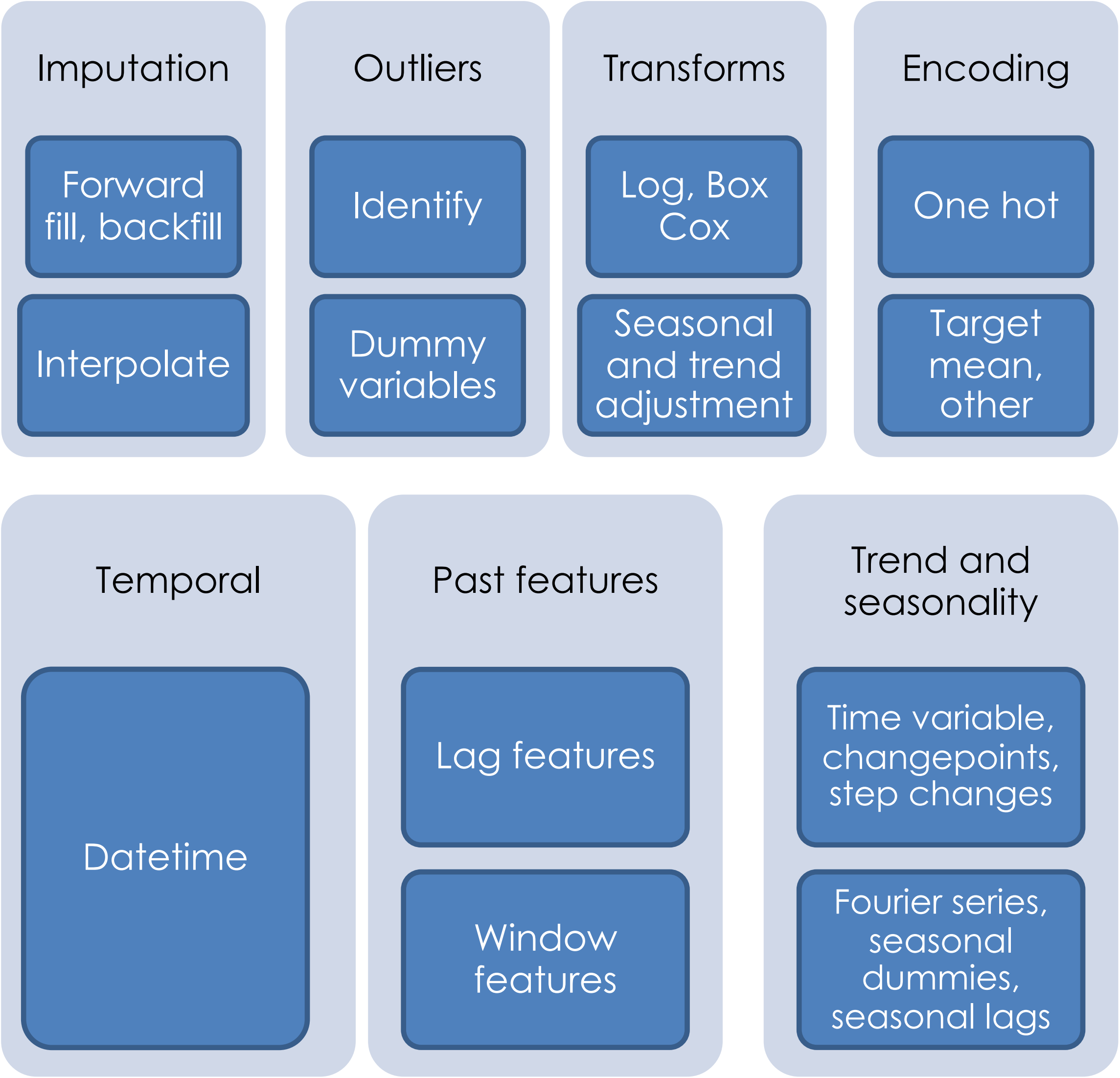


Conclusions

1. Forecasting can be treated as a tabular ML task and compete with statistical models.

2. The feature engineering and ML workflow is different for time series forecasting.

3. Forecasting comes with its own set of feature engineering methods and concerns.



Conclusions

1. Forecasting can be treated as a tabular ML task and compete with statistical models.

2. The feature engineering and ML workflow is different for time series forecasting.

3. Forecasting comes with its own set of feature engineering methods and concerns.

4. More support is increasingly becoming available for time series tasks in Python.

SKTIME

tsfresh

Feature-engine

Darts

If you'd like to learn more …

Feature engineering for time series forecasting online course

trainindata.com/p/feature-engineering-for-forecasting

@KishManani

In/kishanmanani

@Soledad_Galli

In/soledad-galli

References

[1] Makridakis, Spyros, Evangelos Spiliotis, and Vassilios Assimakopoulos. "The M5 competition: Background, organization, and implementation." *International Journal of Forecasting* (2021).

[2] Makridakis, Spyros, Evangelos Spiliotis, and Vassilios Assimakopoulos. "M5 accuracy competition: Results, findings, and conclusions." *International journal of forecasting* (2022).

[3] Sorjamaa, Antti, and Amaury Lendasse. "Time series prediction using DirRec strategy." In *Esann*, vol. 6, pp. 143-148. 2006.

[4] Bontempi, Gianluca, Souhaib Ben Taieb, and Yann-Aël Le Borgne. "Machine learning strategies for time series forecasting." In *European business intelligence summer school*, pp. 62-77. Springer, Berlin, Heidelberg, 2012.

[5] Taieb, Souhaib Ben, and Rob J. Hyndman. *Recursive and direct multi-step forecasting: the best of both worlds*. Vol. 19. Department of Econometrics and Business Statistics, Monash Univ., 2012.

[6] Petropoulos, Fotios, Daniele Apiletti, Vassilios Assimakopoulos, Mohamed Zied Babai, Devon K. Barrow, Souhaib Ben Taieb, Christoph Bergmeir et al. "Forecasting: theory and practice." *International Journal of Forecasting* (2022).

Any questions?

Appendix

Static features: Integer encoding

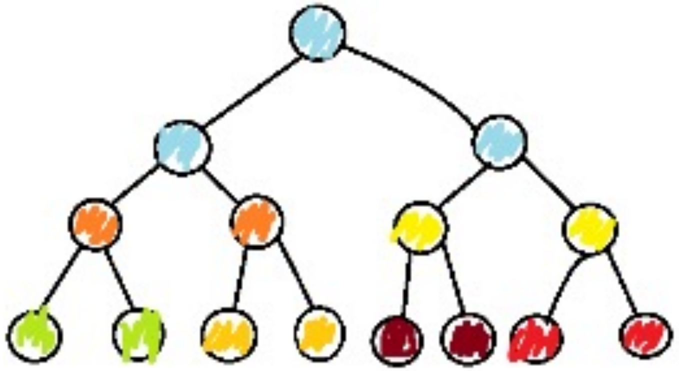
Time	Product ID	y_t
…	…	…
2020-02-13	SKU-1	30
2020-02-14	SKU-1	32
2020-02-15	SKU-1	25
2020-02-16	SKU-1	34
2020-02-17	SKU-1	?
…	…	…
2020-02-14	SKU-2	32
2020-02-15	SKU-2	21
2020-02-16	SKU-2	25
2020-02-17	SKU-2	?

Static features: Integer encoding

Time	Product ID (encoded)
…	…
2020-02-13	1
2020-02-14	1
2020-02-15	1
2020-02-16	1
2020-02-17	1
…	…
2020-02-14	2
2020-02-15	2
2020-02-16	2
2020-02-17	2

Map each category to an integer:

SKU-1 ↔ 1
SKU-2 ↔ 2
…



y_t
…
30
32
25
34
?
…
32
21
25
?

Other multi-step forecasting strategies exist

Combine direct and recursive strategies

- DirRec [3, 4]
- Rectify [5]

Directly predict the whole output sequence

- Multi-output [4]

DirRec

Model 1: $\hat{y}_{T+1} = \mathrm{f}_1(y_T, y_{T-1}, \dots)$
Model 2: $\hat{y}_{T+2} = \mathrm{f}_2(\hat{y}_{T+1}, y_T, y_{T-1}, \dots)$
...
Model h: $\hat{y}_{T+h} = \mathrm{f}_\mathrm{h}(\hat{y}_{T+h-1}, \hat{y}_{T+h-2}, \dots, y_T, y_{T-1} \dots)$

[3] Sorjamaa, Antti, and Amaury Lendasse. "Time series prediction using DirRec strategy." In *Esann*, vol. 6, pp. 143-148. 2006.

[4] Bontempi, Gianluca, Souhaib Ben Taieb, and Yann-Aël Le Borgne. "Machine learning strategies for time series forecasting." In *European business intelligence summer school*, pp. 62-77. Springer, Berlin, Heidelberg, 2012.

[5] Taieb, Souhaib Ben, and Rob J. Hyndman. *Recursive and direct multi-step forecasting: the best of both worlds*. Vol. 19. Department of Econometrics and Business Statistics, Monash Univ., 2012.