

Feature engineering for time series forecasting

PyData London 2022
Kishan Manani

About me

- Data Science Manager
- Online Course Developer
- trainindata.com/p/feature-engineering-for-forecasting
- Slides:
<https://github.com/KishManani/PyDataLondon2022>



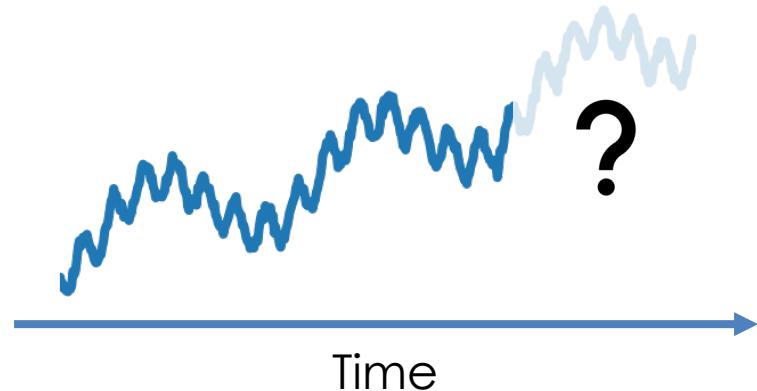
Kishan Manani, PhD

 @KishManani

 In/kishanmanani

 medium.com/@kish.manani

About this talk



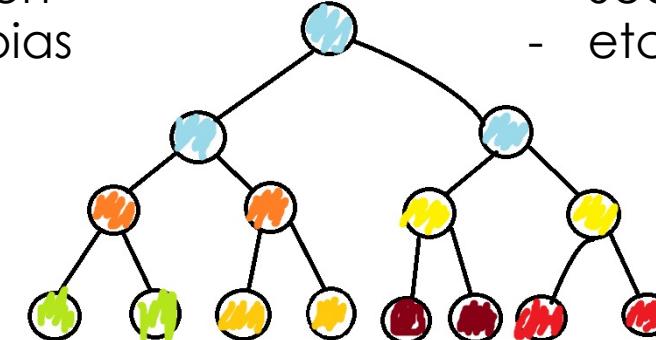
Time	Sales (UK)	Sales (Germany)
2020-02-12	35	21
2020-02-13	30	20
2020-02-14	23	19
2020-02-15	?	?
2020-02-16	?	?



x1	x2	x3	y
			35
			30
			23
			?
			?

Forecasting workflow

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



Time series features

- Lags, windows
- Seasonality
- etc.

Contents



Time series forecasting using ML models



Features for time series forecasting



Useful libraries for forecasting with ML models

Why use machine learning for forecasting?

Featured Prediction Competition

M5 Forecasting - Accuracy

Estimate the unit sales of Walmart retail goods



\$50,000 Prize Money

University of Nicosia · 5,558 teams · 2 years ago

Overview Data Code Discussion Leaderboard [Leaderboard](#) Join Competition ...

kaggle

Overview

Description

Evaluation

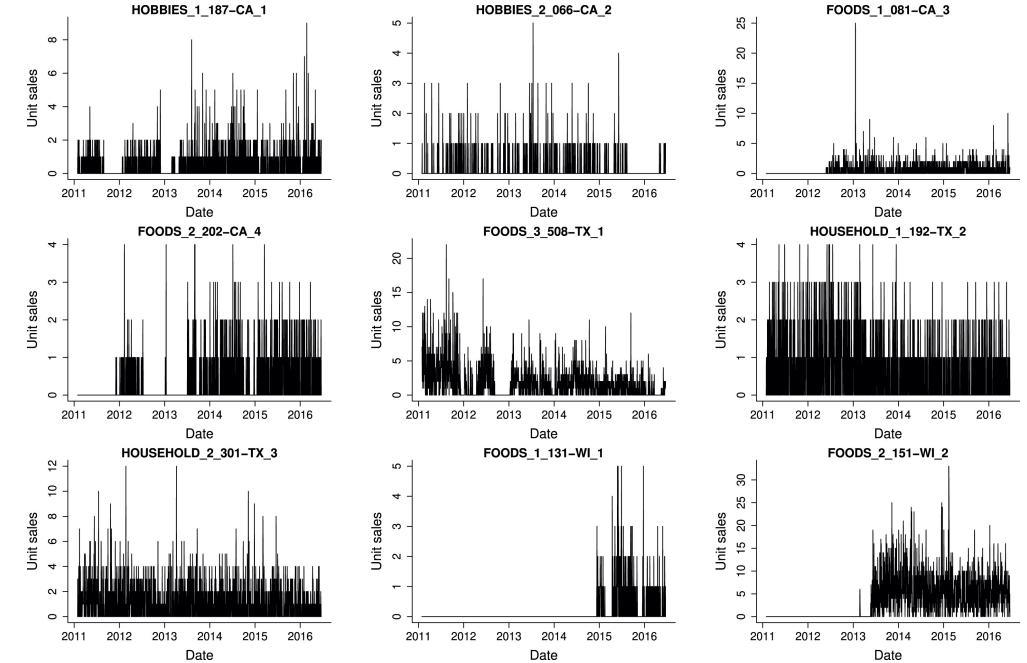
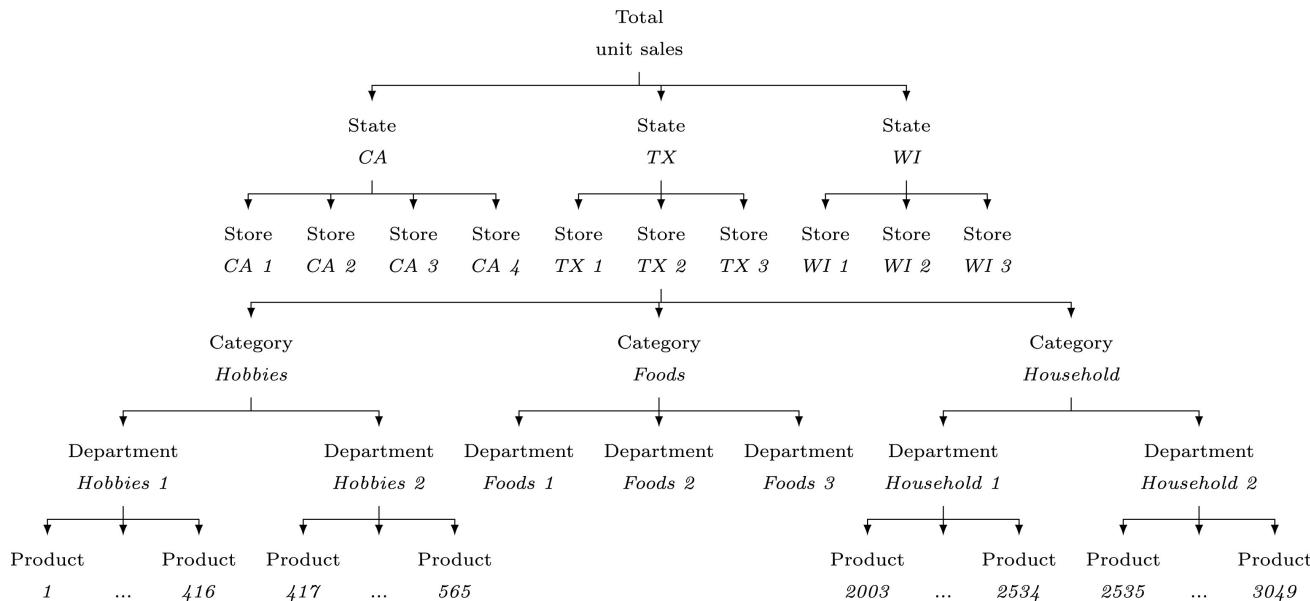
Timeline

Note: This is one of the two complementary competitions that together comprise the M5 forecasting challenge. Can you estimate, as precisely as possible, the point forecasts of the unit sales of various products sold in the USA by Walmart? If you are interested in estimating the uncertainty distribution of the realized values of the same series, be sure to check out its [companion competition](#)

How much camping gear will one store sell each month in a year? To the uninitiated, calculating sales at

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).

Forecasting with machine learning

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	?

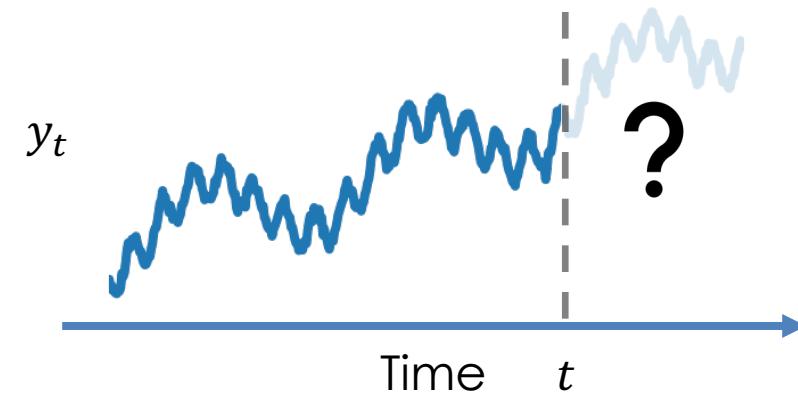
\vdots

$T - 2$

$T - 1$

T

$T + 1$



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	?

■ ■ ■

T - 2

T - 1

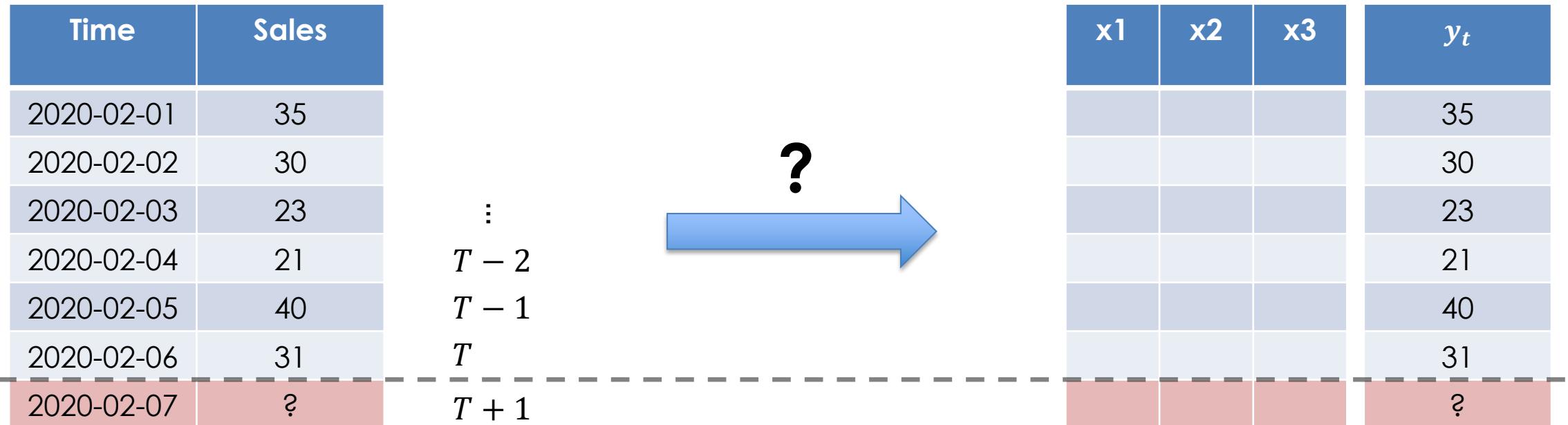
T

T + 1

?



Time series to a table of features and a target



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	?

Only use data known at time of target.
This is to avoid look-ahead bias.

\vdots

$T - 2$

$T - 1$

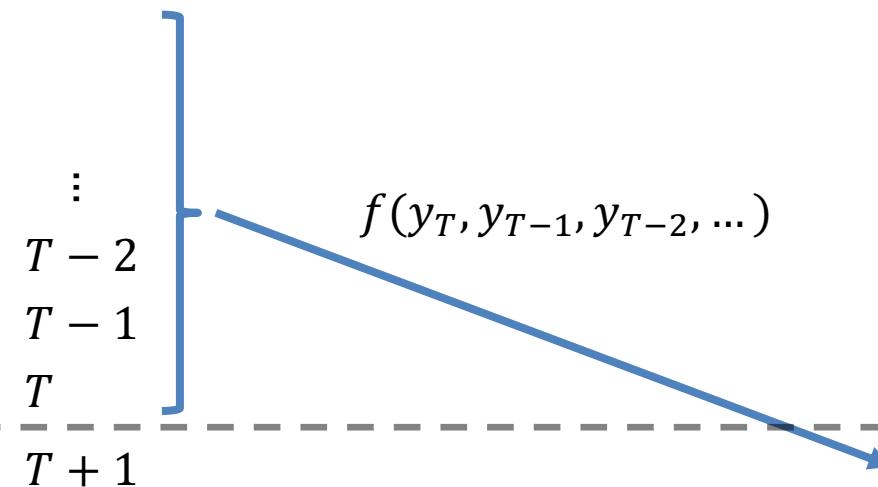
T

$T + 1$

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	?



x_1	x_2	x_3	y_t
			35
			30
			23
			21
			40
			31
			?

Time series to a table of features and a target



Time series to a table of features and a target



Time series to a table of features and a target



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	Rainfall	Ad spend	y_{t-3}	y_{t-2}	y_{t-1}	y_t
2020-02-01	35	12	100	NaN	NaN	NaN	35
2020-02-02	30	15	120	NaN	NaN	35	30
2020-02-03	23	13	116	NaN	35	30	23
2020-02-04	21	14	120	35	30	23	21
2020-02-05	40	23	101	30	23	21	40
2020-02-06	31	25	90	23	21	40	31
2020-02-07	?	?	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	Rainfall	Ad spend	y_{t-3}	y_{t-2}	y_{t-1}	y_t
2020-02-01	35	12	100	NaN	NaN	NaN	35
2020-02-02	30	15	120	NaN	NaN	35	30
2020-02-03	23	13	116	NaN	35	30	23
2020-02-04	21	14	120	35	30	23	21
2020-02-05	40	23	101	30	23	21	40
2020-02-06	31	25	90	23	21	40	31
2020-02-07	?	\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	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	?

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	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	?


 x
(features)

y
(target)

Time series to a table of features and a target

Training data

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)

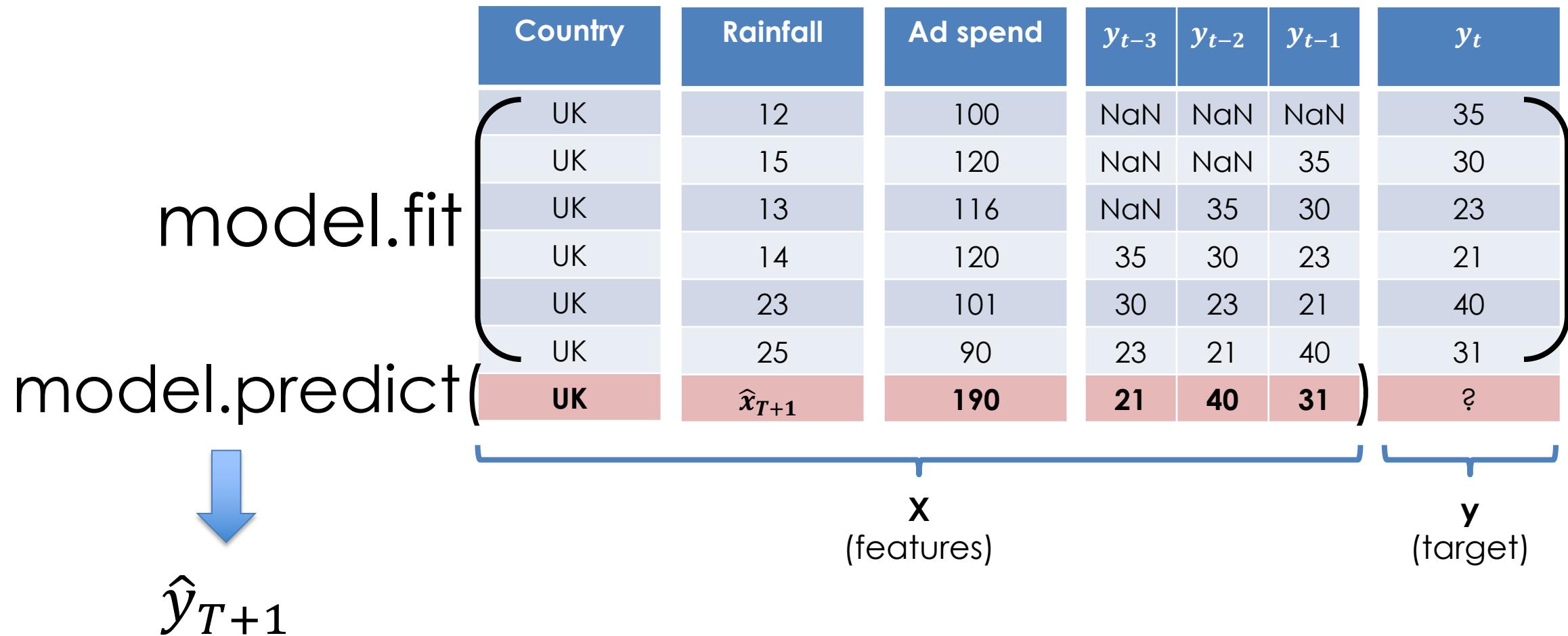
Time series to a table of features and a target

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	?

x (features) y (target)

Time series to a table of features and a 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	?

- Direct forecasting
- Recursive forecasting

:

$T - 2$

$T - 1$

T

.

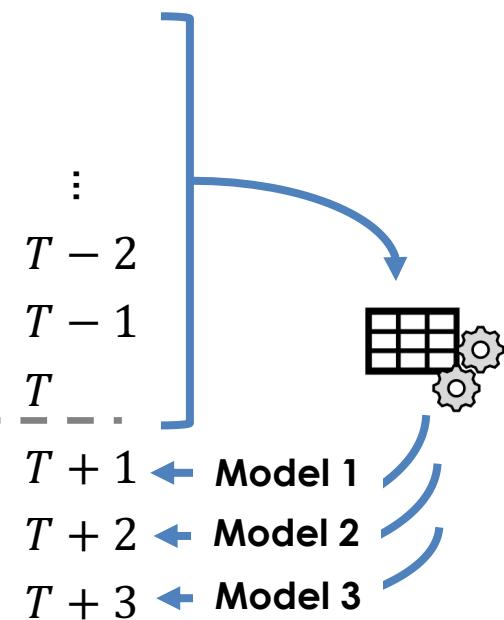
$T + 1$

$T + 2$

$T + 3$

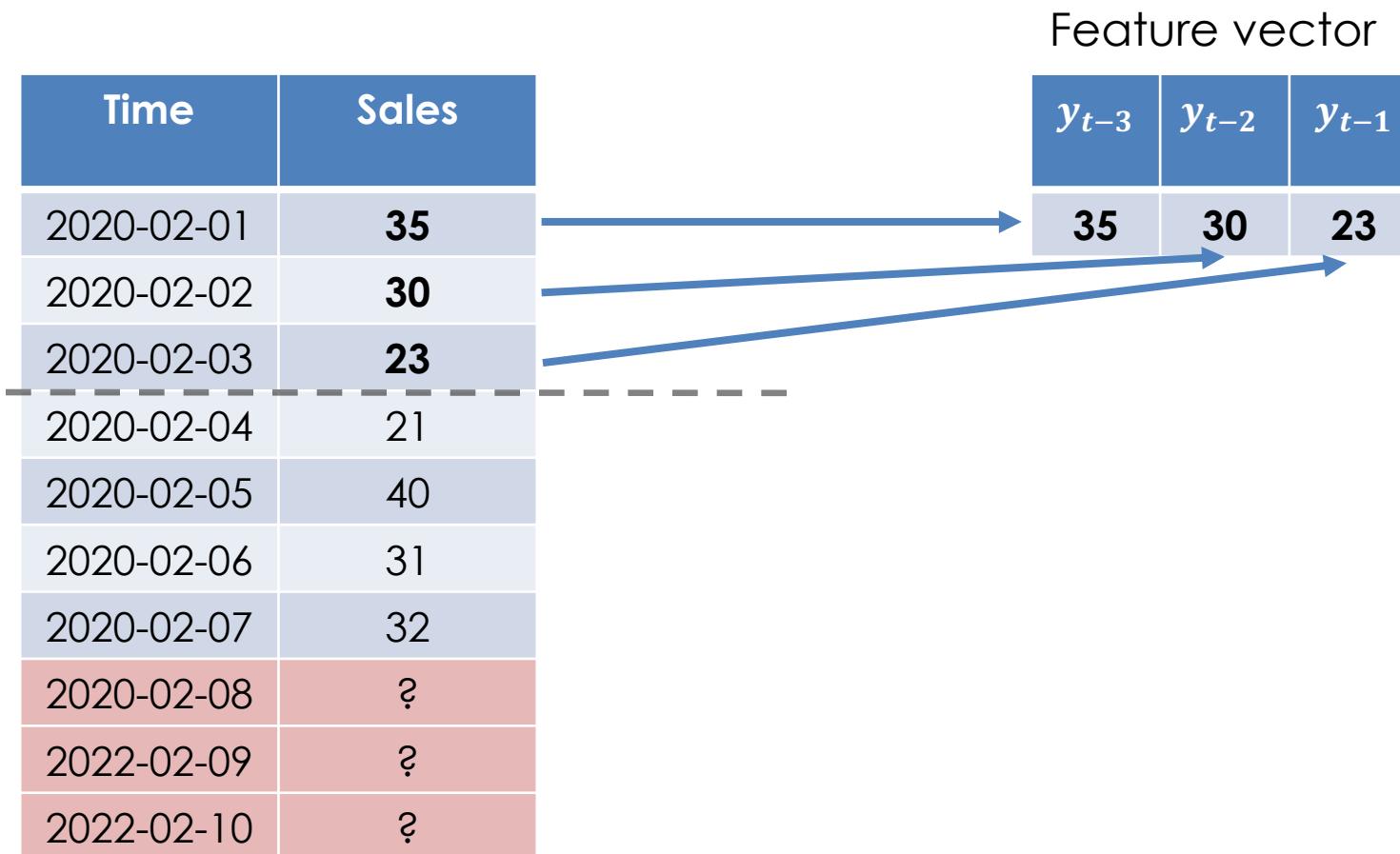
Multi-step forecasting: Direct forecasting

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	?

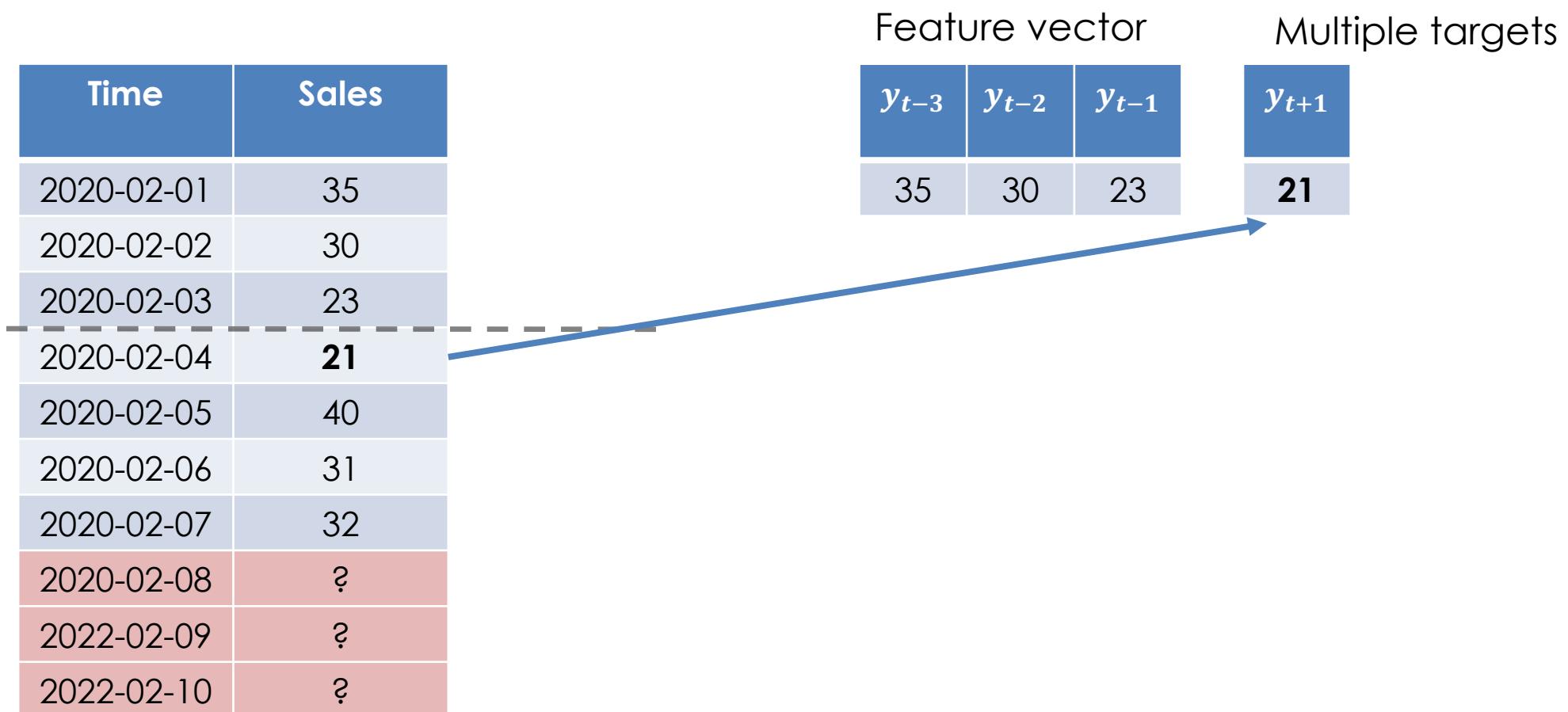


- Directly predict $y_{T+1}, y_{T+2}, \dots, y_{T+h}$
- **Same features** but **different target variable** for each forecast step.
- **Multiple models** trained with different targets, one for each forecast step.

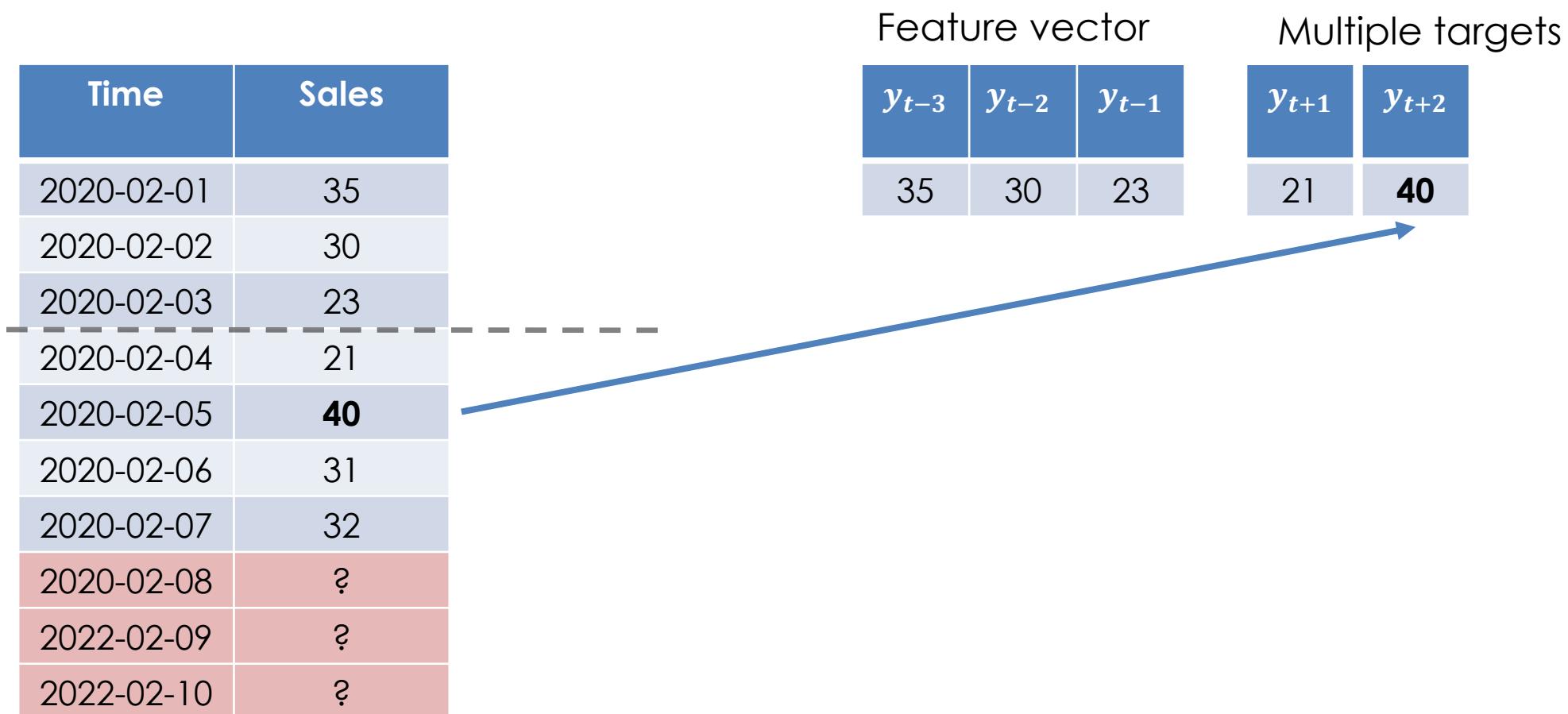
Multi-step forecasting: Direct forecasting



Multi-step forecasting: Direct forecasting



Multi-step forecasting: Direct forecasting

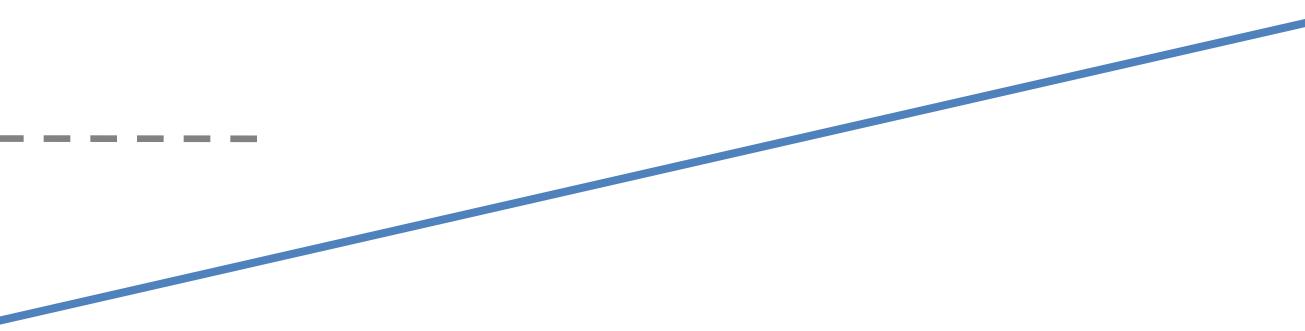


Multi-step forecasting: Direct forecasting

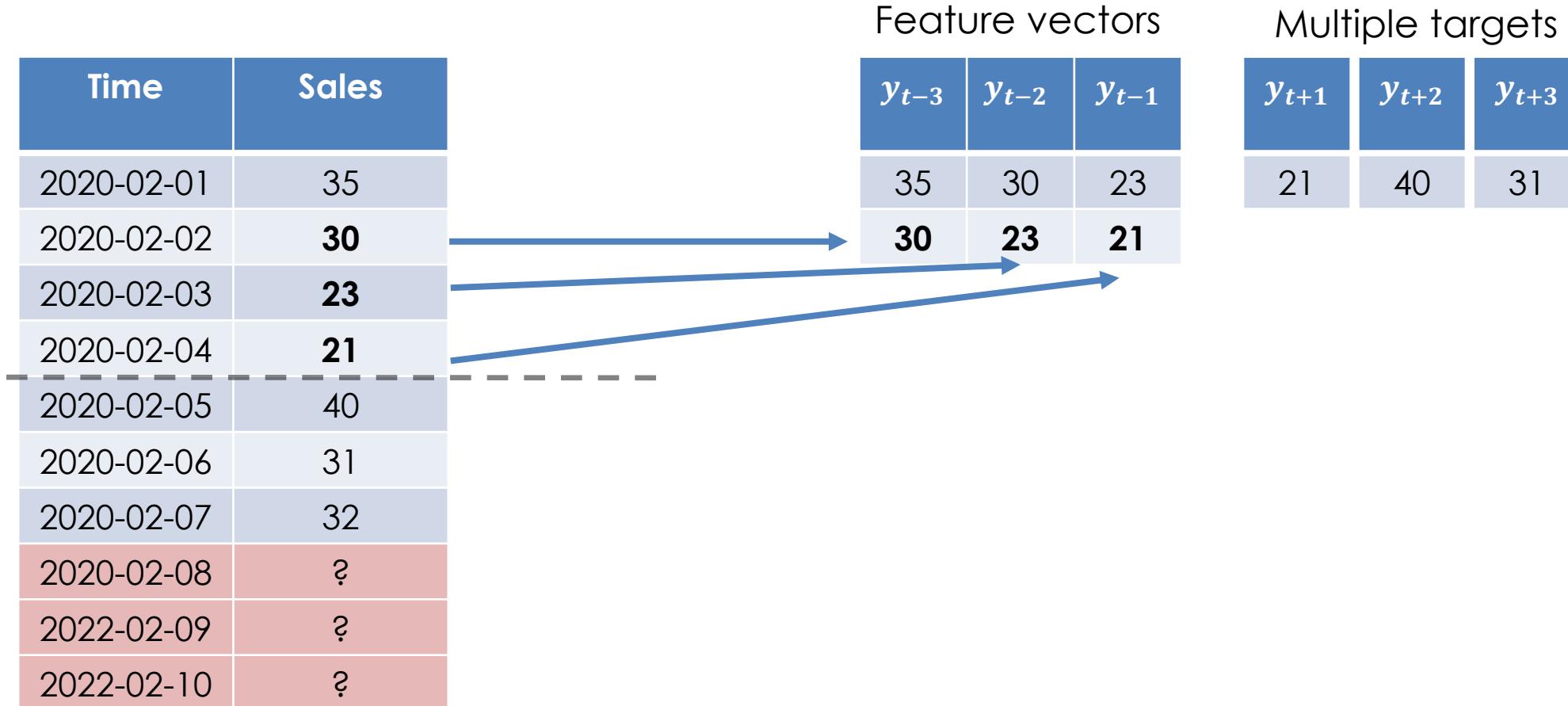
Time	Sales
2020-02-01	35
2020-02-02	30
2020-02-03	23
2020-02-04	21
2020-02-05	40
2020-02-06	31
2020-02-07	32
2020-02-08	?
2022-02-09	?
2022-02-10	?

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

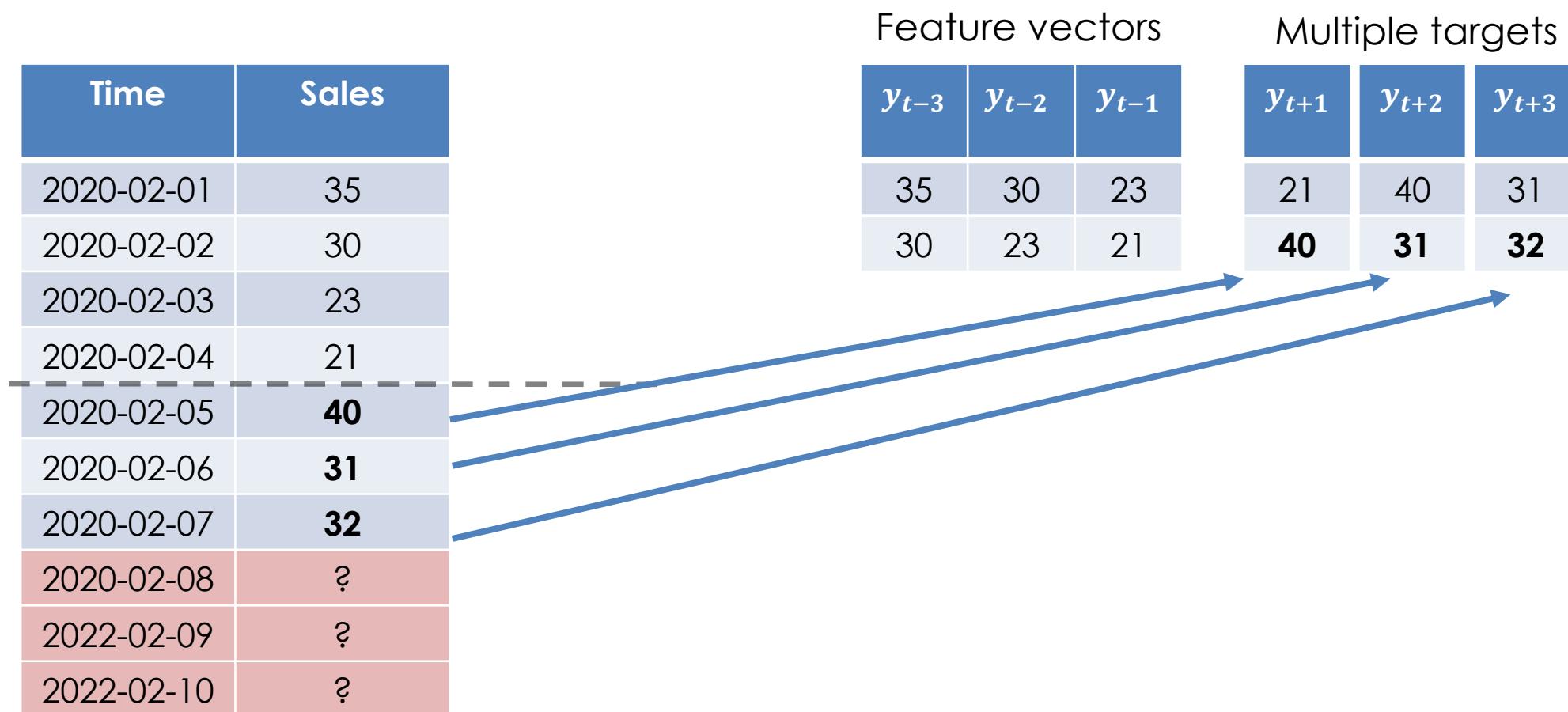
Multiple targets		
y_{t+1}	y_{t+2}	y_{t+3}
21	40	31



Multi-step forecasting: Direct forecasting

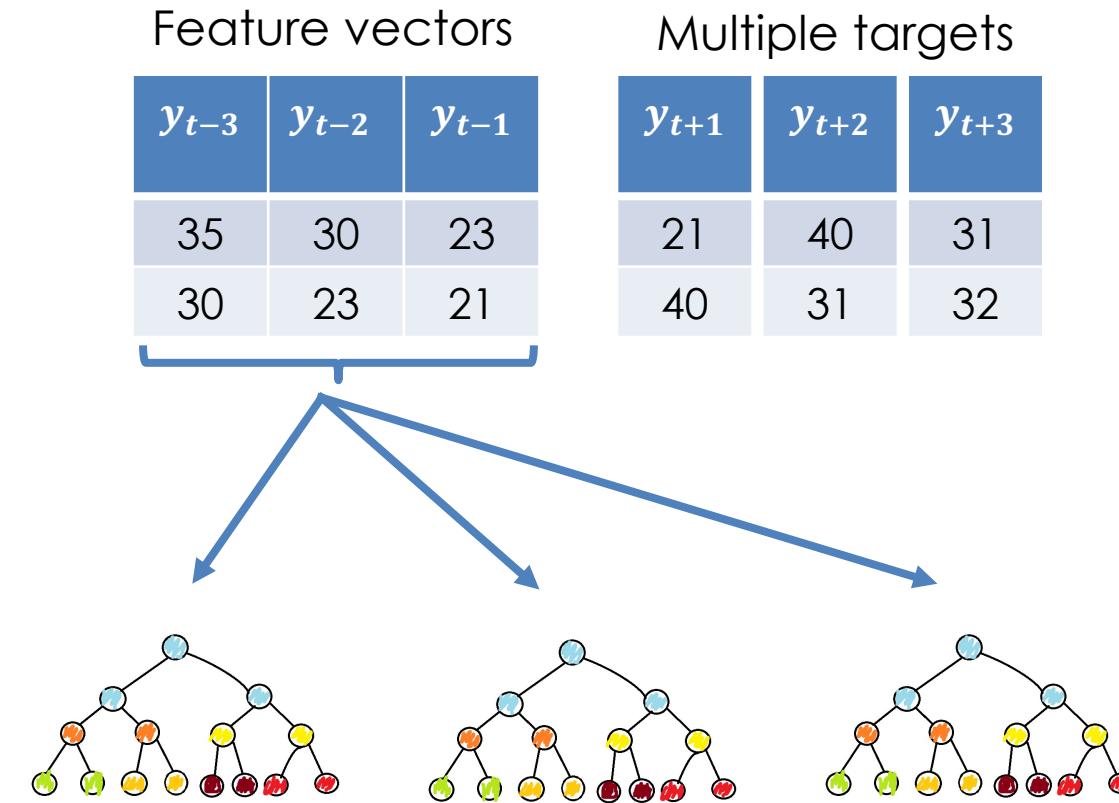


Multi-step forecasting: Direct forecasting



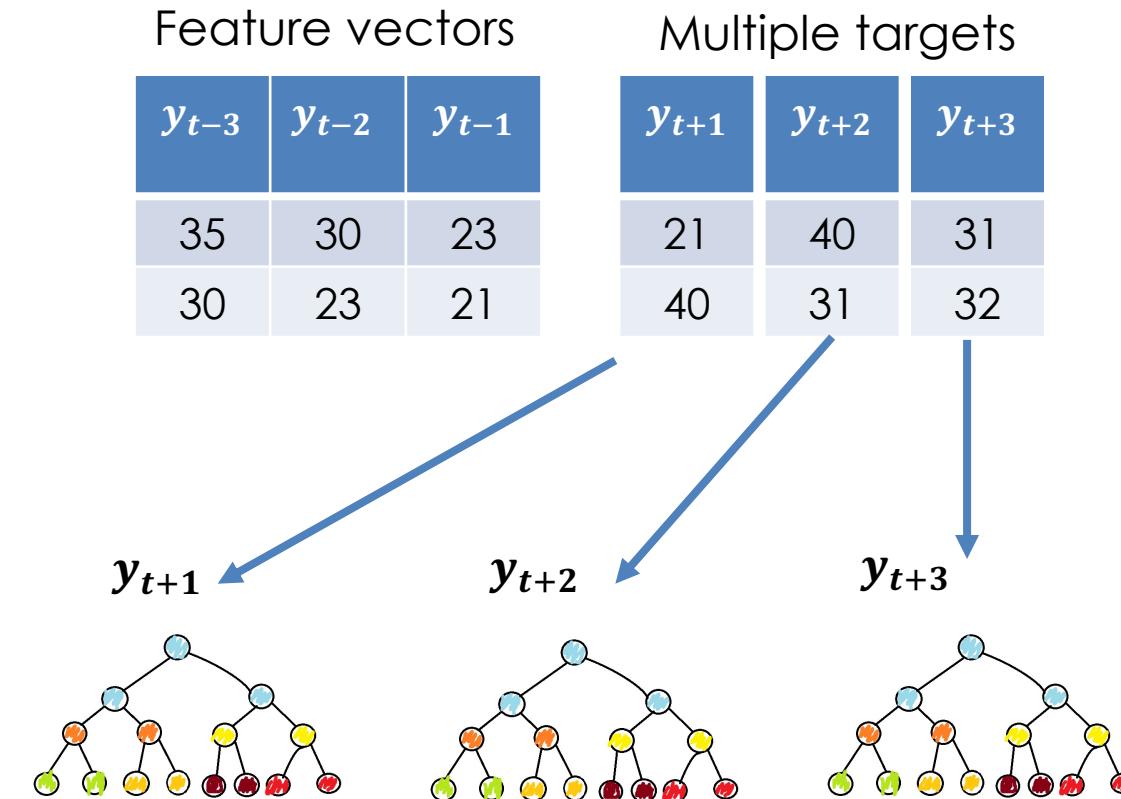
Multi-step forecasting: Direct forecasting

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	?



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	?



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	?

Input

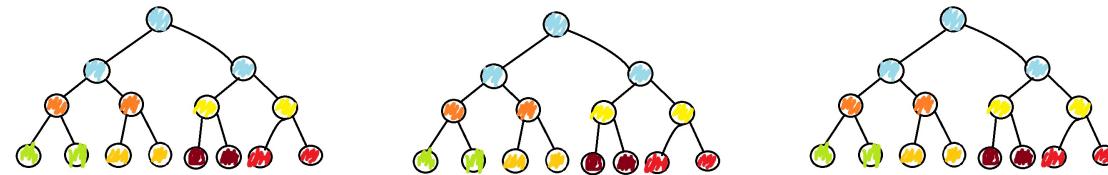
Feature vectors

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

Multiple targets

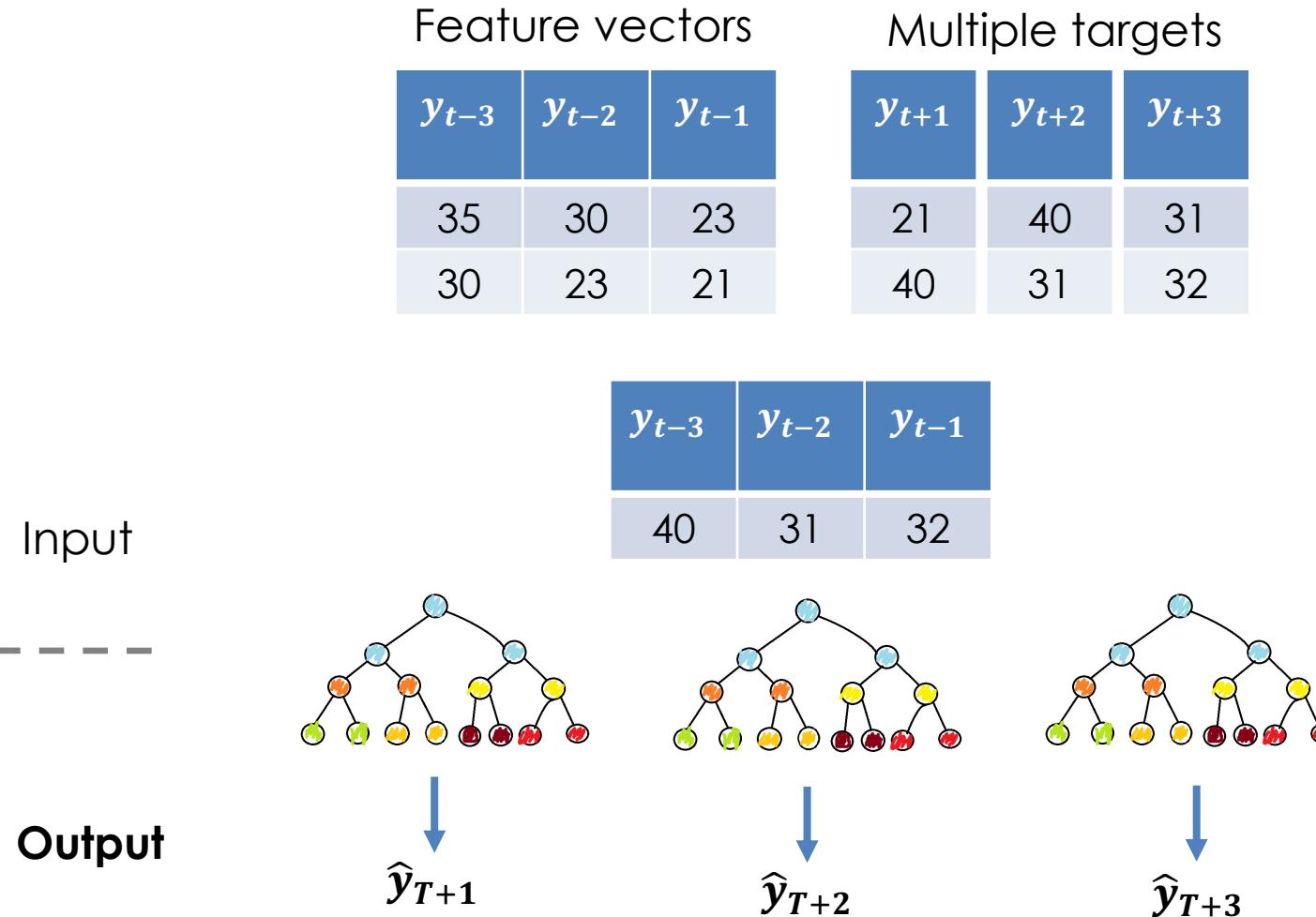
y_{t+1}	y_{t+2}	y_{t+3}
21	40	31
40	31	32

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



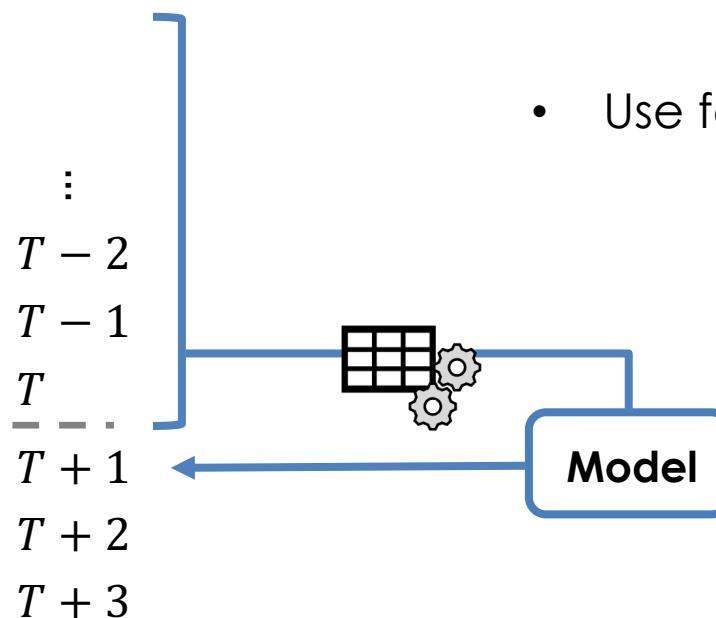
Multi-step forecasting: Direct forecasting

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}



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	?



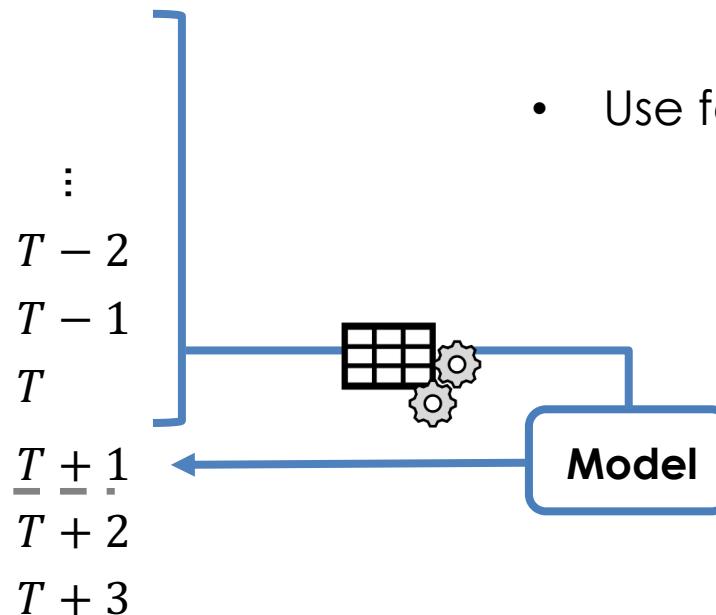
- Recursively apply a 1-step ahead forecast model.
- Use forecasted output as new input.

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	?
2022-02-09	?

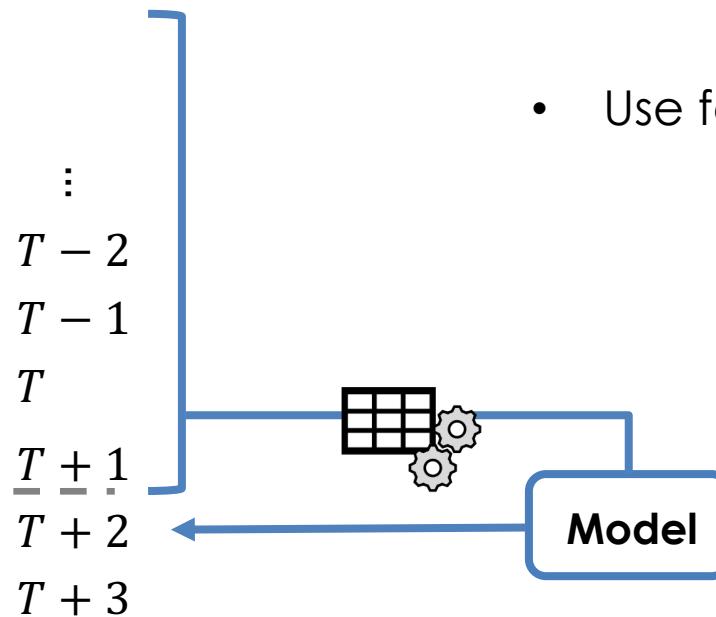
- Recursively apply a 1-step ahead forecast model.

- Use forecasted output as new input.



Multi-step forecasting: Recursive forecasting

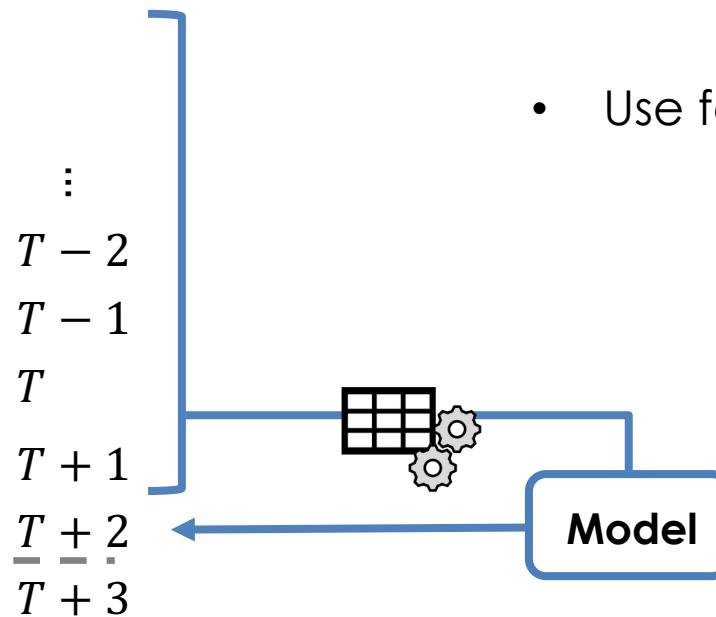
Time	Sales
2020-02-01	35
2020-02-02	30
2020-02-03	23
2020-02-04	21
2020-02-05	40
2020-02-06	31
2020-02-07	\hat{y}_{T+1}
2022-02-08	\hat{y}_{T+2}
2022-02-09	?



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

Multi-step forecasting: Recursive forecasting

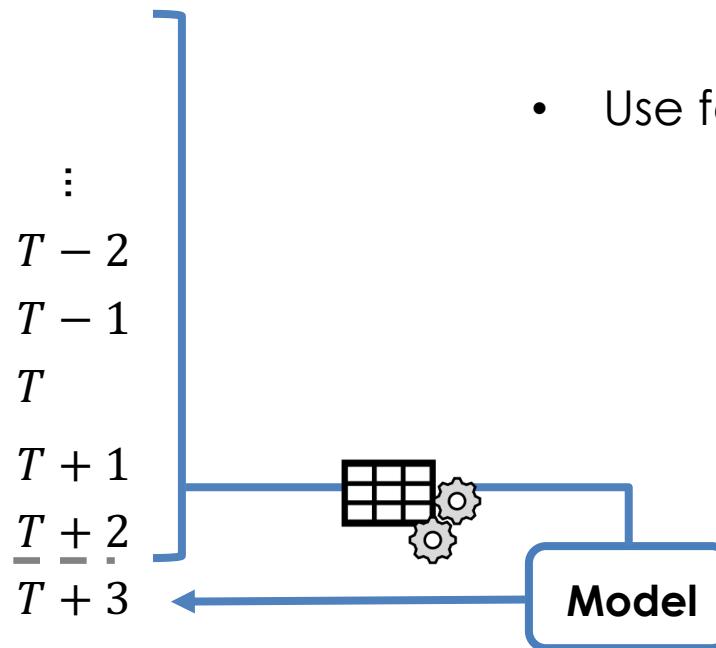
Time	Sales
2020-02-01	35
2020-02-02	30
2020-02-03	23
2020-02-04	21
2020-02-05	40
2020-02-06	31
2020-02-07	\hat{y}_{T+1}
2022-02-08	\hat{y}_{T+2}
2022-02-09	?



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

Multi-step forecasting: Recursive forecasting

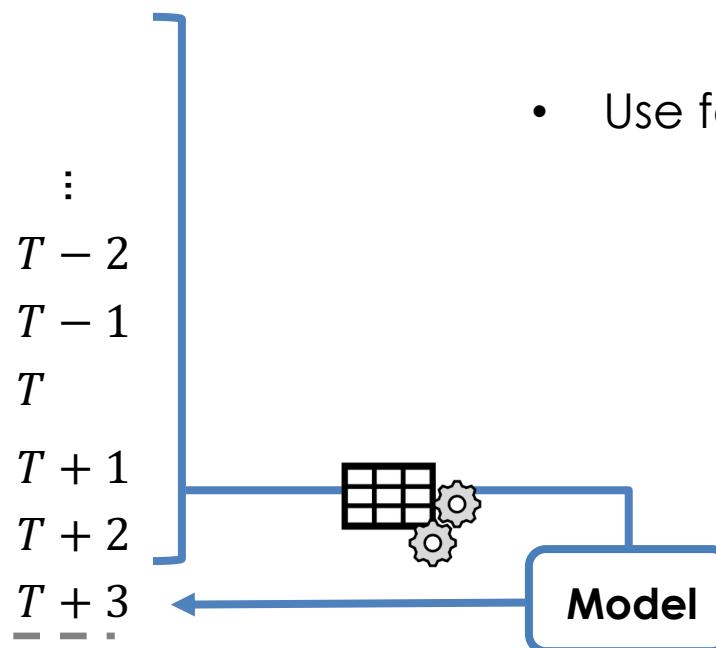
Time	Sales
2020-02-01	35
2020-02-02	30
2020-02-03	23
2020-02-04	21
2020-02-05	40
2020-02-06	31
2020-02-07	\hat{y}_{T+1}
2022-02-08	\hat{y}_{T+2}
2022-02-09	\hat{y}_{T+3}



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

Multi-step forecasting: Recursive forecasting

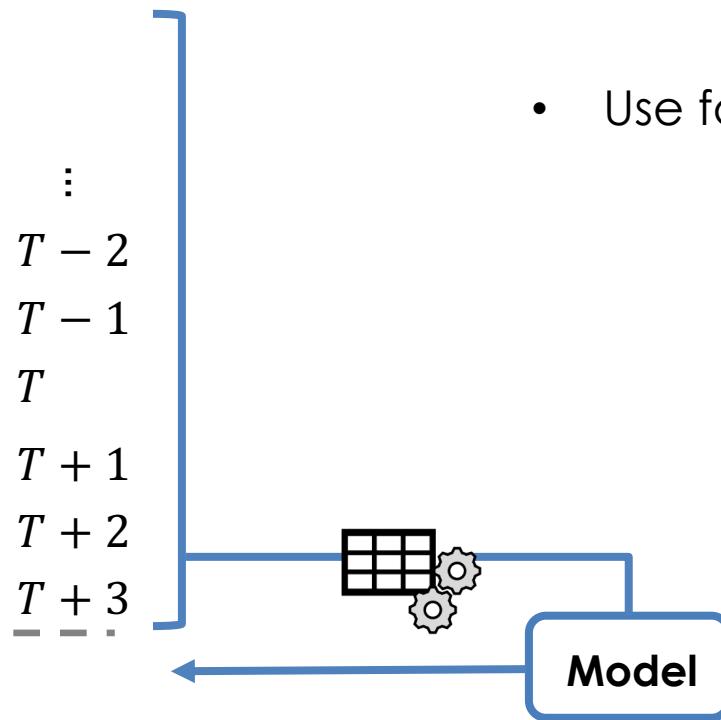
Time	Sales
2020-02-01	35
2020-02-02	30
2020-02-03	23
2020-02-04	21
2020-02-05	40
2020-02-06	31
2020-02-07	\hat{y}_{T+1}
2022-02-08	\hat{y}_{T+2}
2022-02-09	\hat{y}_{T+3}



- 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	\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	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 be 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 be 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 be created iteratively.

Multi-step forecasting: Recursive forecasting

model.fit

model.predict(

\hat{y}_{T+1}

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	?
UK	\hat{x}_{T+2}	201				?
UK	\hat{x}_{T+3}	110				?

Multi-step forecasting: Recursive forecasting

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				?

Multi-step forecasting: Recursive forecasting

model.fit

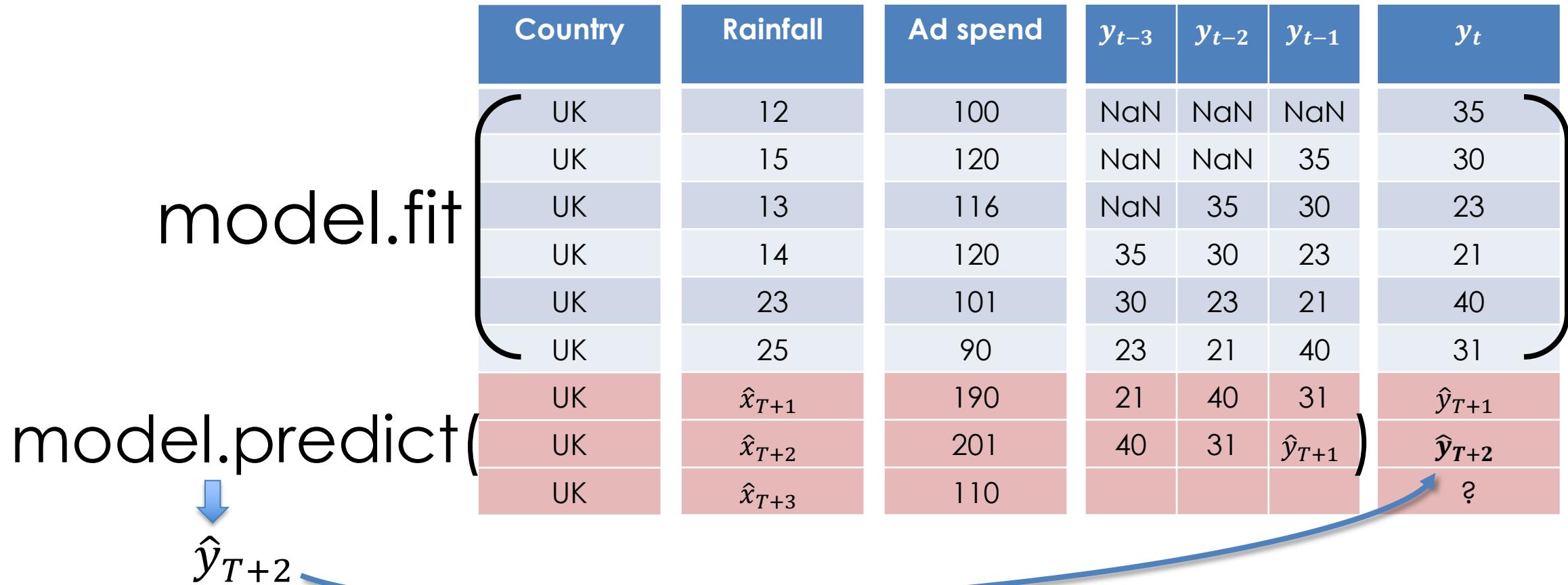
Country	Rainfall	Ad spend	y_{t-3}	y_{t-2}	y_{t-1}	y_t
UK	12	100	NaN	NaN	NaN	35
UK	15	120	NaN	NaN	35	30
UK	13	116	NaN	35	30	23
UK	14	120	35	30	23	21
UK	23	101	30	23	21	40
UK	25	90	23	21	40	31
UK	\hat{x}_{T+1}	190	21	40	31	\hat{y}_{T+1}
UK	\hat{x}_{T+2}	201	40	31	\hat{y}_{T+1}	?
UK	\hat{x}_{T+3}	110		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

Multi-step forecasting: Recursive forecasting

model.fit
model.predict()
↓
 \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}	?
UK	\hat{x}_{T+3}	110				?

Multi-step forecasting: Recursive forecasting



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}	?

Multi-step forecasting: Recursive forecasting

model.fit

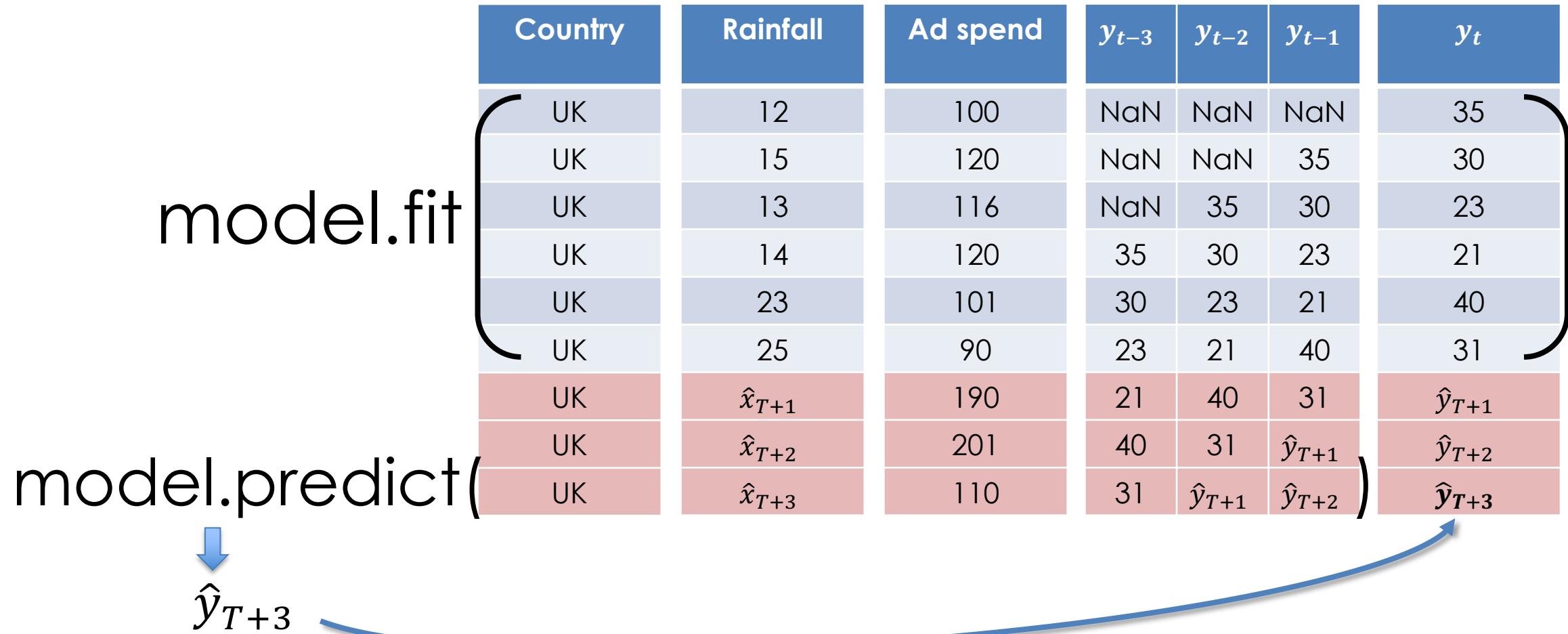
Country	Rainfall	Ad spend	y_{t-3}	y_{t-2}	y_{t-1}	y_t
UK	12	100	NaN	NaN	NaN	35
UK	15	120	NaN	NaN	35	30
UK	13	116	NaN	35	30	23
UK	14	120	35	30	23	21
UK	23	101	30	23	21	40
UK	25	90	23	21	40	31
UK	\hat{x}_{T+1}	190	21	40	31	\hat{y}_{T+1}
UK	\hat{x}_{T+2}	201	40	31	\hat{y}_{T+1}	\hat{y}_{T+2}
UK	\hat{x}_{T+3}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

model.predict(



\hat{y}_{T+3}

Multi-step forecasting: Recursive forecasting

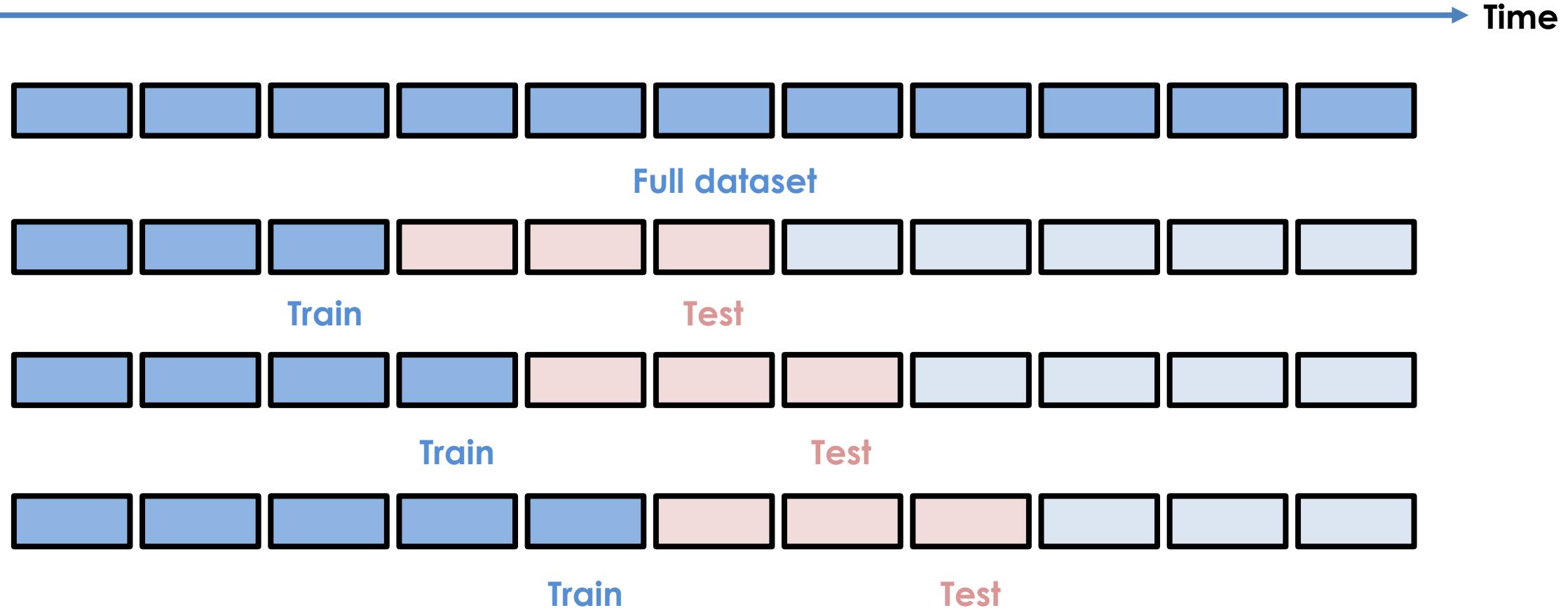


Cross-validation: Tabular vs Time series

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?

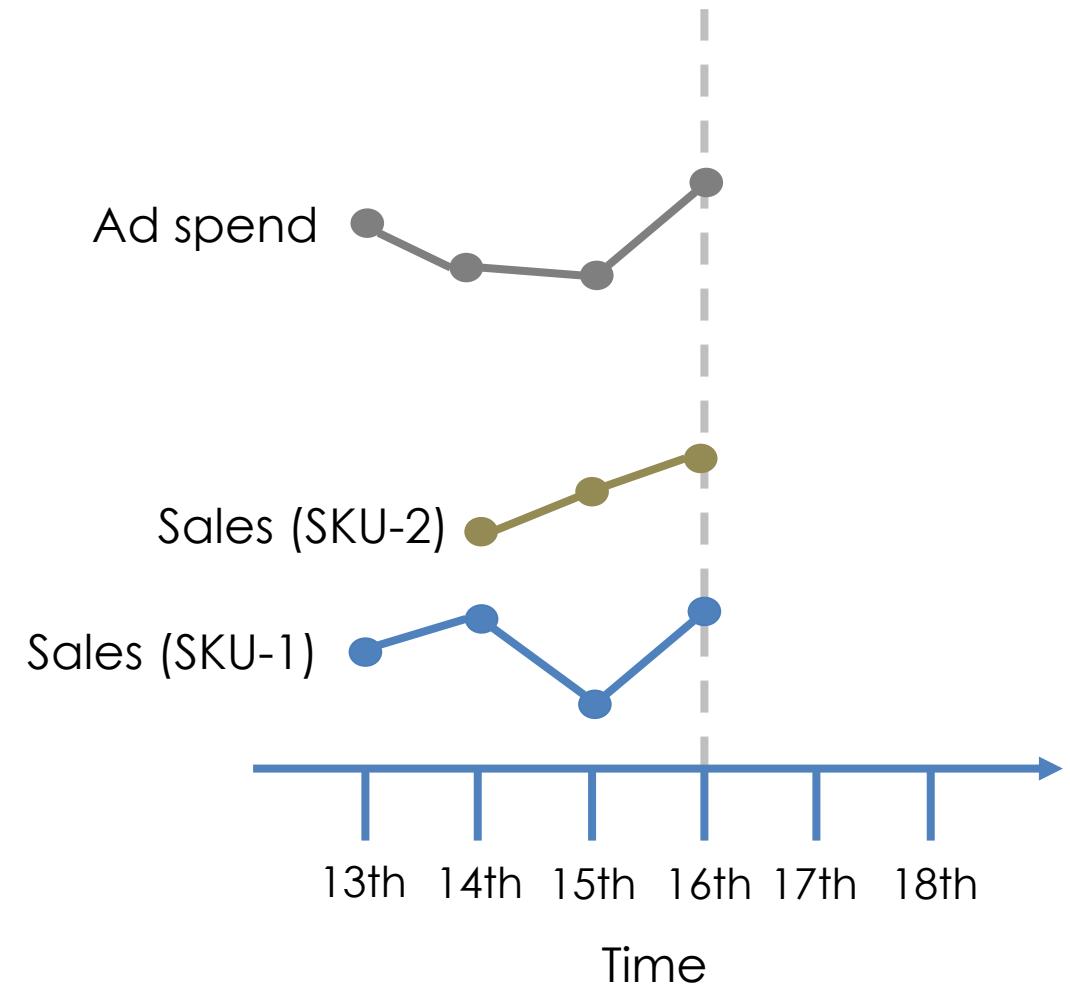
Any data that is known at the time of prediction.

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.

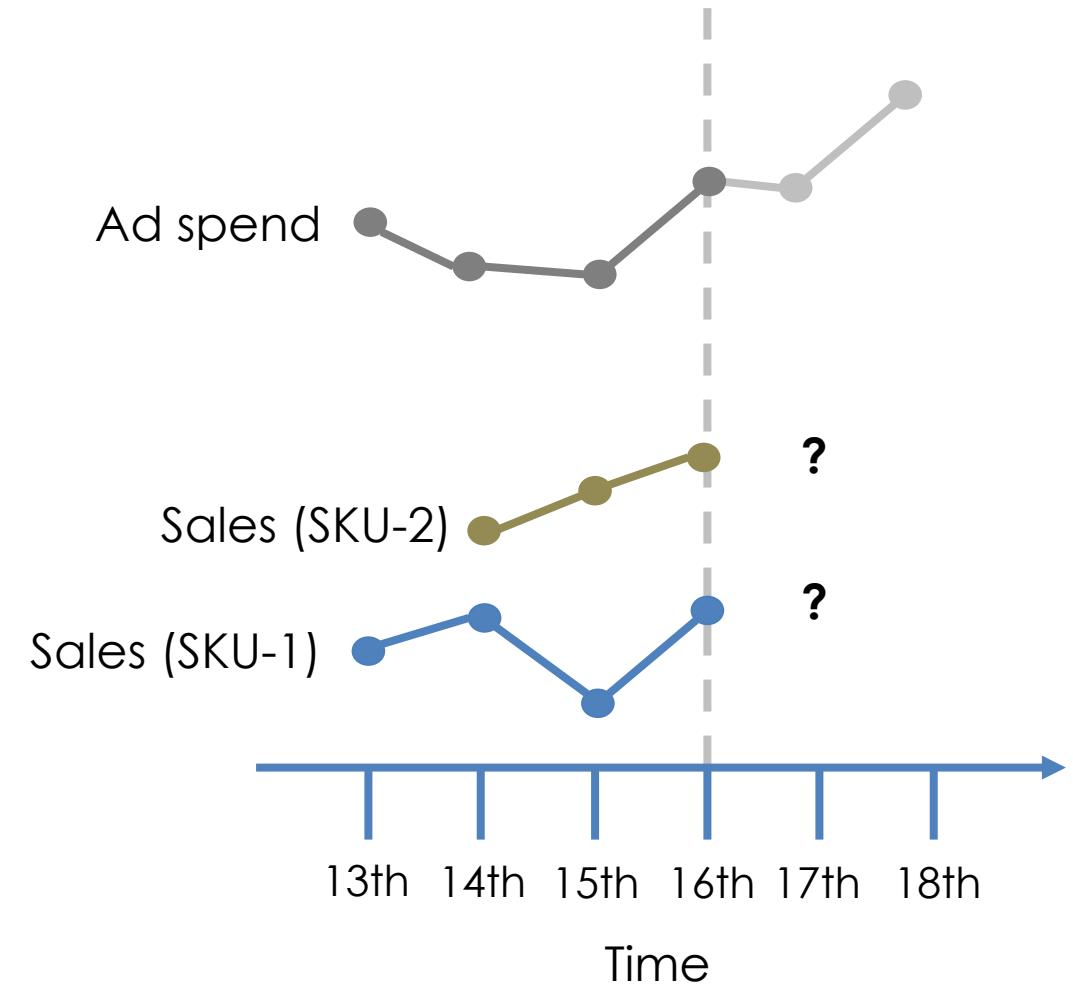
An example

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



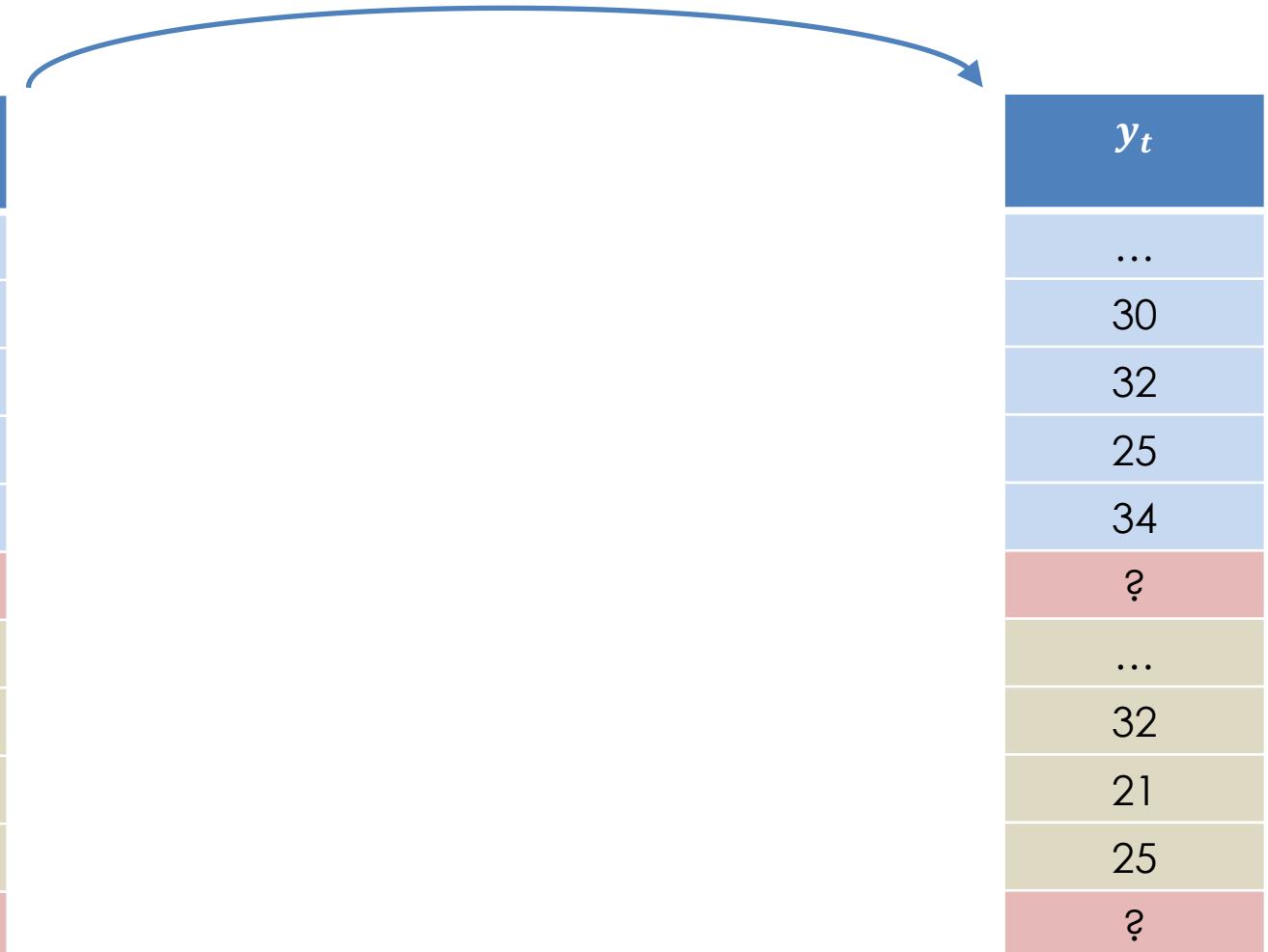
An example

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	?

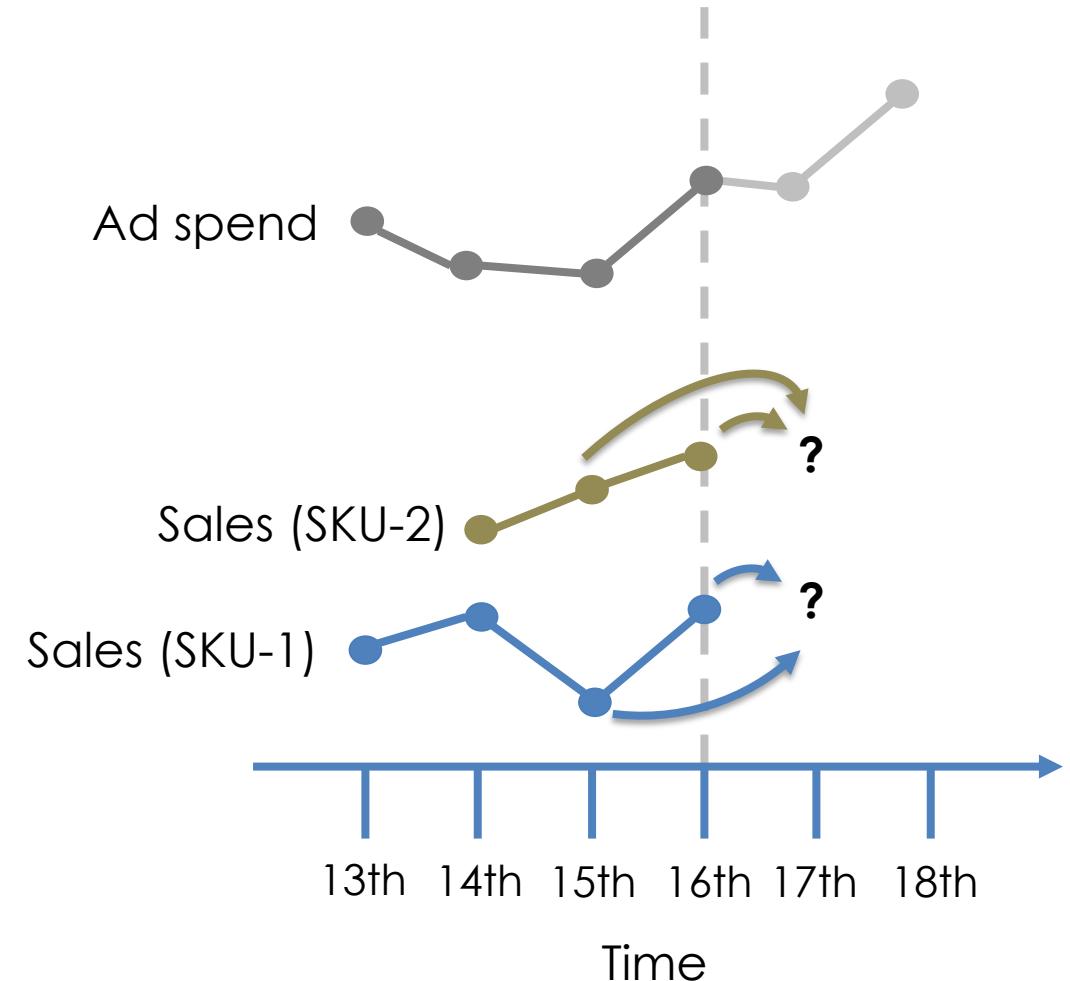


Lag features: Past values of target & features

- Past 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).

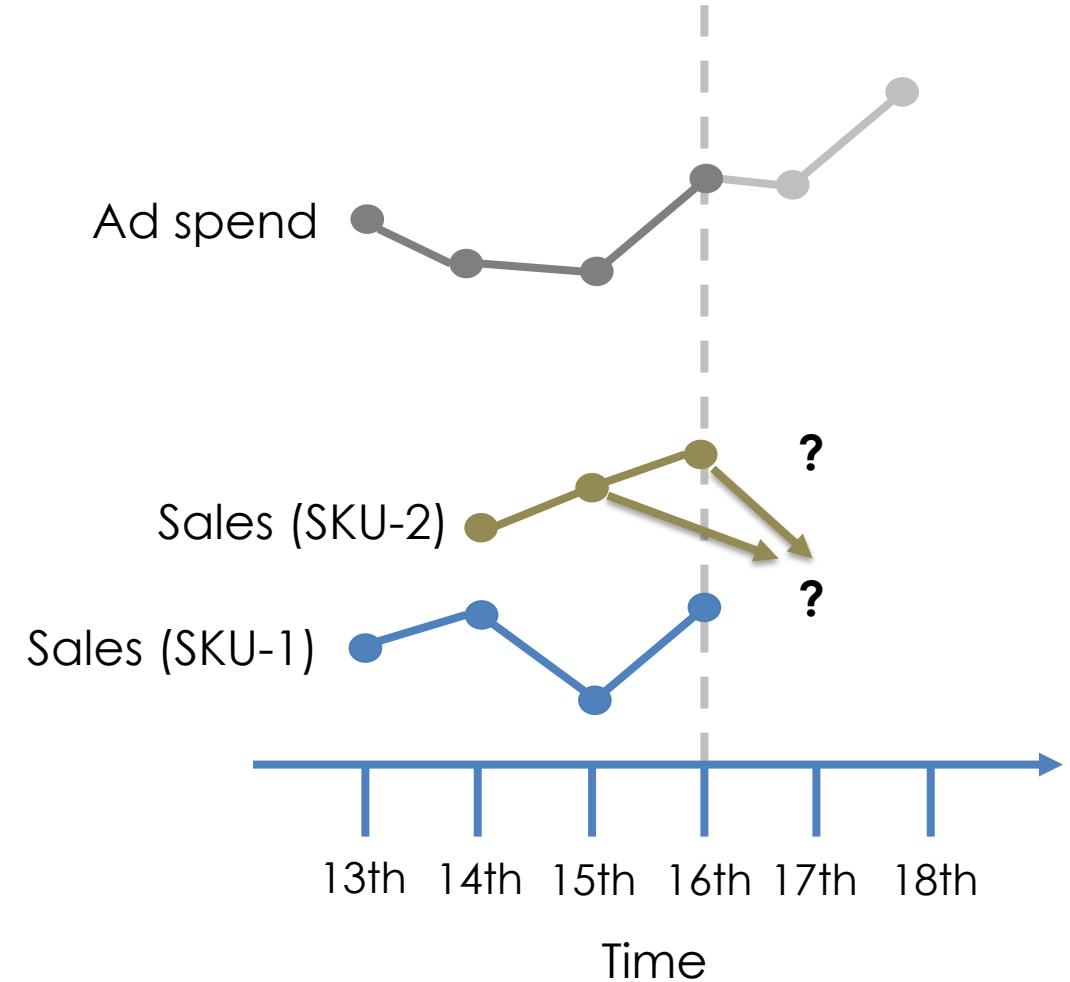


Lag features: Past values of target & features

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

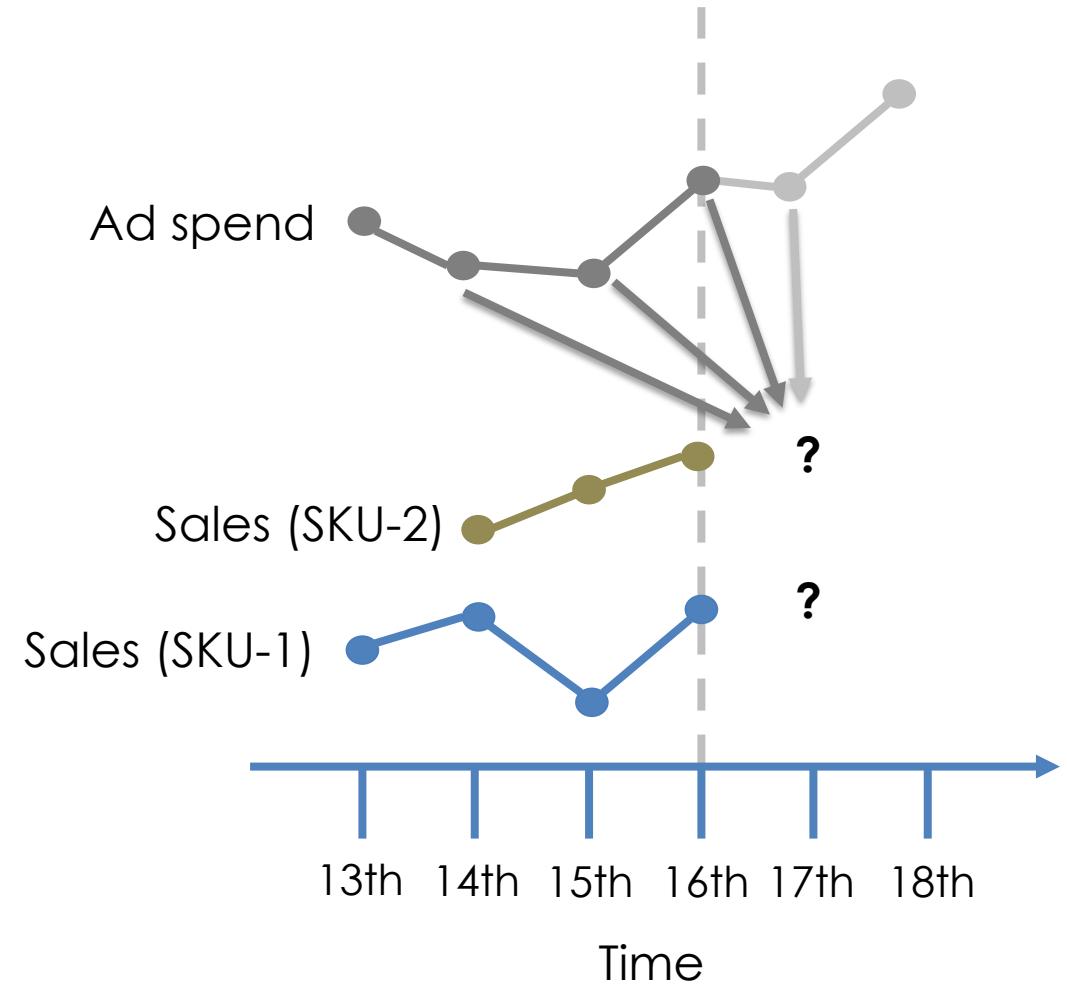
$$\hat{y}_t \leftarrow y_{t-1}, y_{t-2}, \dots, y_{t-k}$$

- 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

- Past 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.
- Past 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
...
2020-02-13	SKU-1	100
2020-02-14	SKU-1	120
2020-02-15	SKU-1	110
2020-02-16	SKU-1	101
2020-02-17	SKU-1	102
...
2020-02-14	SKU-2	120
2020-02-15	SKU-2	110
2020-02-16	SKU-2	101
2020-02-17	SKU-2	102

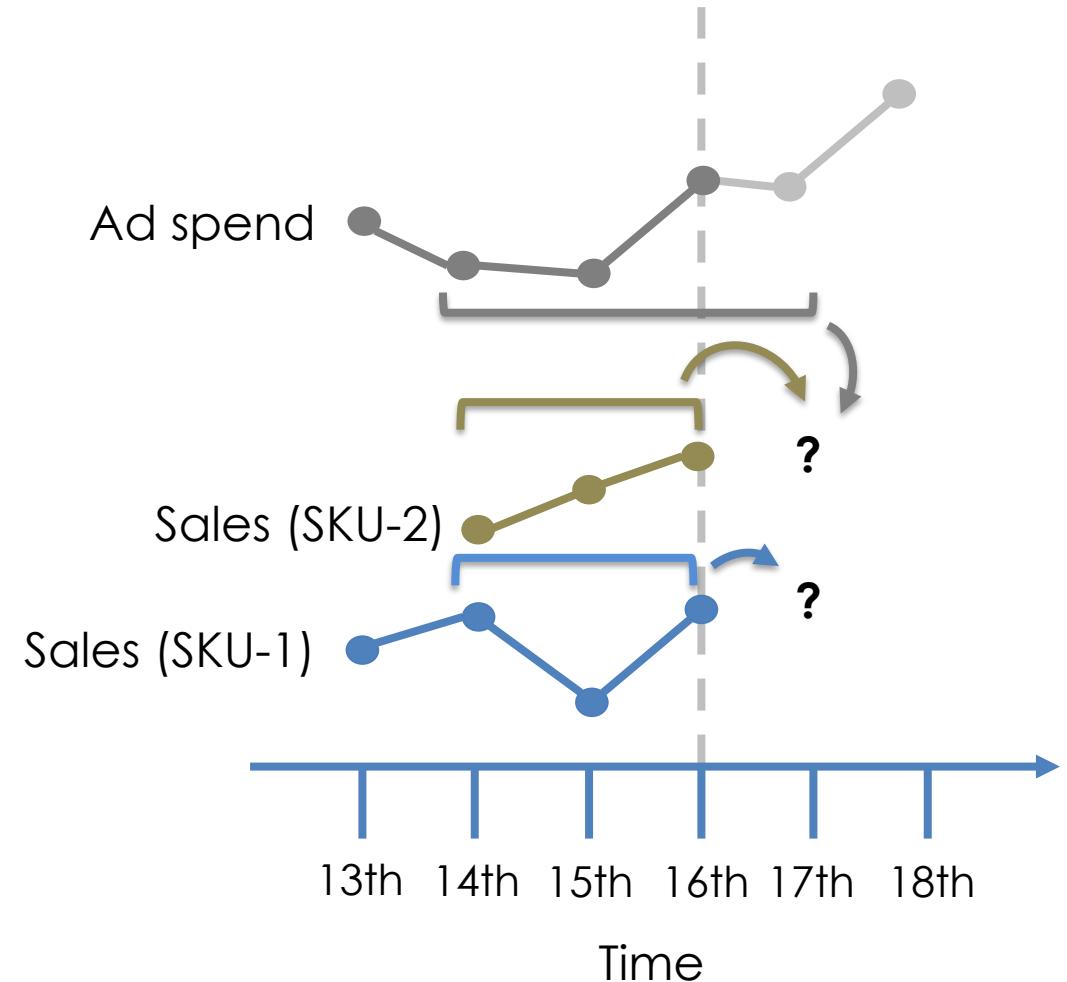
Ad spend Lag 1	Ad spend Lag 2	y_{t-3}	y_{t-2}	y_{t-1}	y_t
...
...	30
100	30	32
120	100	32	25
110	120	30	32	25	34
101	110	32	25	34	?
...
100	32
120	100	32	21
110	120	...	32	21	25
101	110	32	21	25	?

Window features: Function over a past window

- Compute a summary statistic over a window of past data

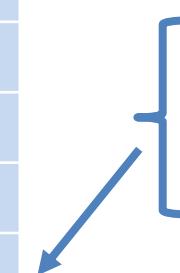
$$\hat{y}_t \leftarrow f(y_{t-1}, y_{t-2}, \dots, y_{t-k})$$

- Mean & standard deviation are common.



Window features: Function over a past window

Time	Product ID	Ad spend
...
2020-02-13	SKU-1	100
2020-02-14	SKU-1	120
2020-02-15	SKU-1	110
2020-02-16	SKU-1	101
2020-02-17	SKU-1	102
...
2020-02-14	SKU-2	120
2020-02-15	SKU-2	110
2020-02-16	SKU-2	101
2020-02-17	SKU-2	102



Rolling std y_{t-1}	Rolling mean y_{t-1}	y_t
...
		30
		32
		25
		34
		?
...
		32
		21
		25
		?

Window features: Function over a past window

Time	Product ID	Ad spend
...
2020-02-13	SKU-1	100
2020-02-14	SKU-1	120
2020-02-15	SKU-1	110
2020-02-16	SKU-1	101
2020-02-17	SKU-1	102
...
2020-02-14	SKU-2	120
2020-02-15	SKU-2	110
2020-02-16	SKU-2	101
2020-02-17	SKU-2	102

Rolling std y_{t-1}	Rolling mean y_{t-1}	y_t
...
		30
		32
		25
		34
2.94	29.0	?
3.85	30.3	...
...	...	32
		21
		25
		?

Window features: Nested window features

Time	Product ID	Ad spend
...
2020-02-13	SKU-1	100
2020-02-14	SKU-1	120
2020-02-15	SKU-1	110
2020-02-16	SKU-1	101
2020-02-17	SKU-1	102
...
2020-02-14	SKU-2	120
2020-02-15	SKU-2	110
2020-02-16	SKU-2	101
2020-02-17	SKU-2	102

The diagram illustrates the concept of nested window features. On the left, a table shows data for two products (SKU-1 and SKU-2) over four days. To the right, curly braces group the data by day (Days), week (Weeks), month (Months), and finally the target variable y_t .

Days	Weeks	Months	y_t
30	32	25	34
?	...	32	32
32	21	25	25
?	...	32	32

Window features: Nested window features

Time	Product ID	Ad spend
...
2020-02-13	SKU-1	100
2020-02-14	SKU-1	120
2020-02-15	SKU-1	110
2020-02-16	SKU-1	101
2020-02-17	SKU-1	102
...
2020-02-14	SKU-2	120
2020-02-15	SKU-2	110
2020-02-16	SKU-2	101
2020-02-17	SKU-2	102

Rolling mean (months)	Rolling mean (weeks)	Rolling mean (days)	y_t
			...
			30
			32
			25
			34
			?
			...
			32
			21
			25
			?

Static features

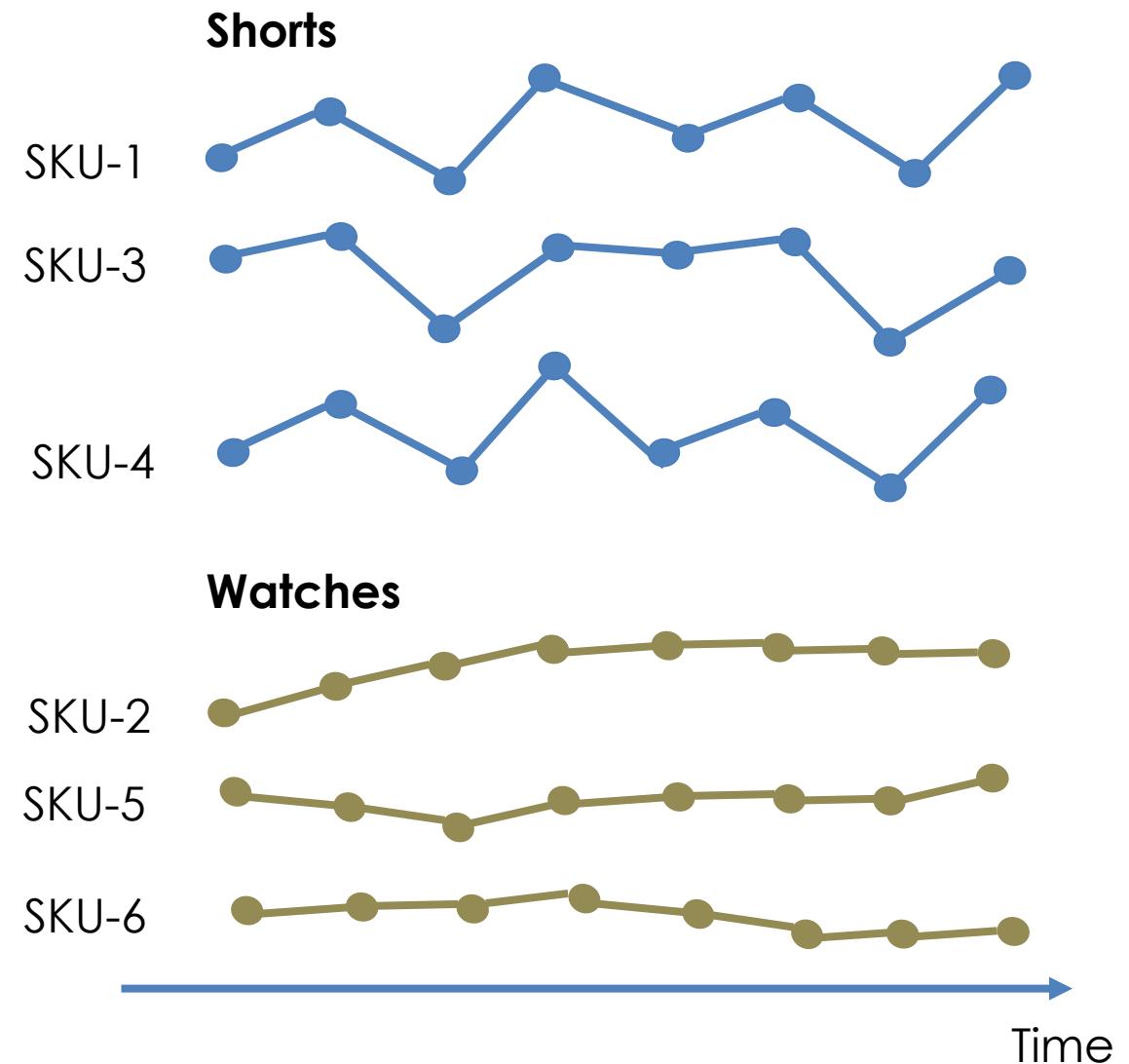
Time	Product ID	y_t
...
2020-02-13	SKU-1	30
2020-02-14	SKU-1	32
2020-02-15	SKU-1	25
2020-02-16	SKU-1	34
2020-02-17	SKU-1	?
...
2020-02-14	SKU-2	32
2020-02-15	SKU-2	21
2020-02-16	SKU-2	25
2020-02-17	SKU-2	?

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	?

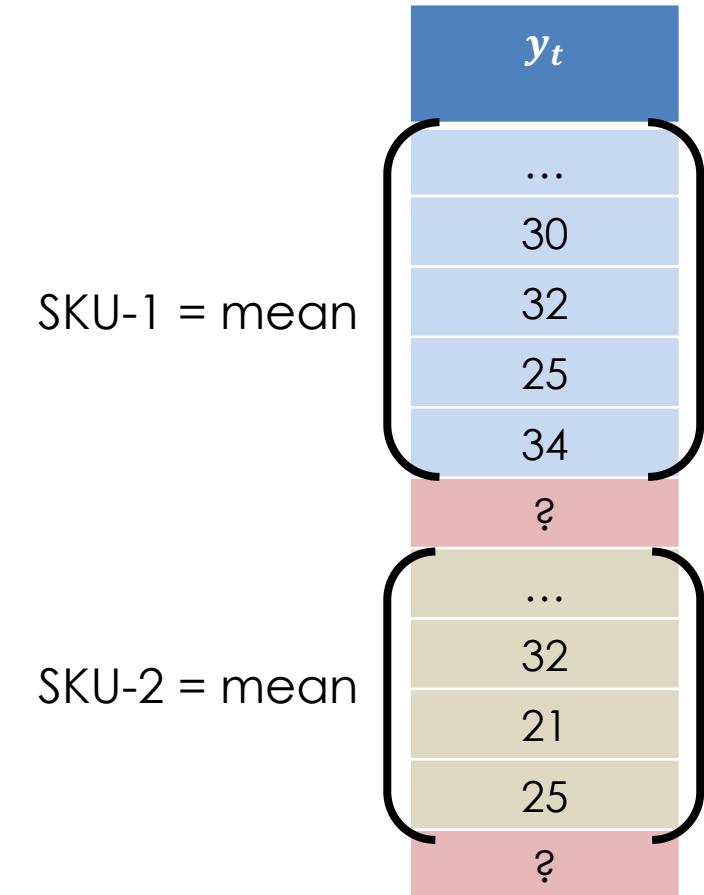
Static features

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

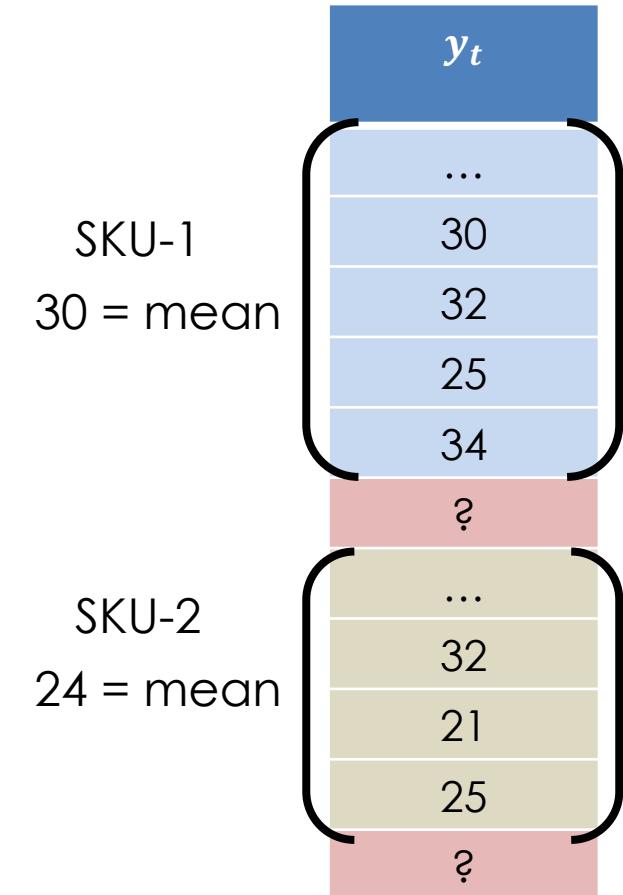
Time	Product ID
...	...
2020-02-13	SKU-1
2020-02-14	SKU-1
2020-02-15	SKU-1
2020-02-16	SKU-1
2020-02-17	SKU-1
...	...
2020-02-14	SKU-2
2020-02-15	SKU-2
2020-02-16	SKU-2
2020-02-17	SKU-2



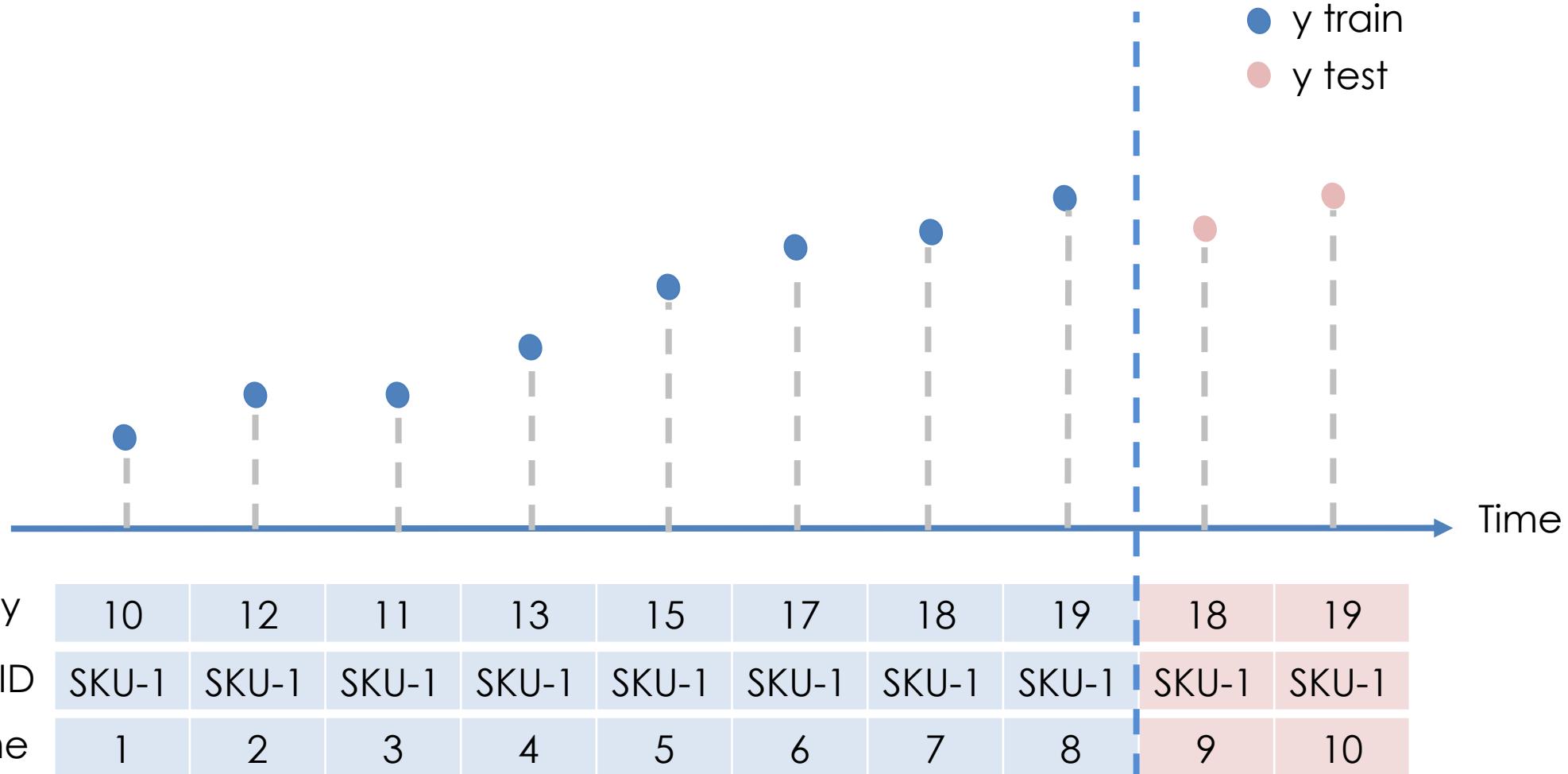
Static features: Target encoding

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

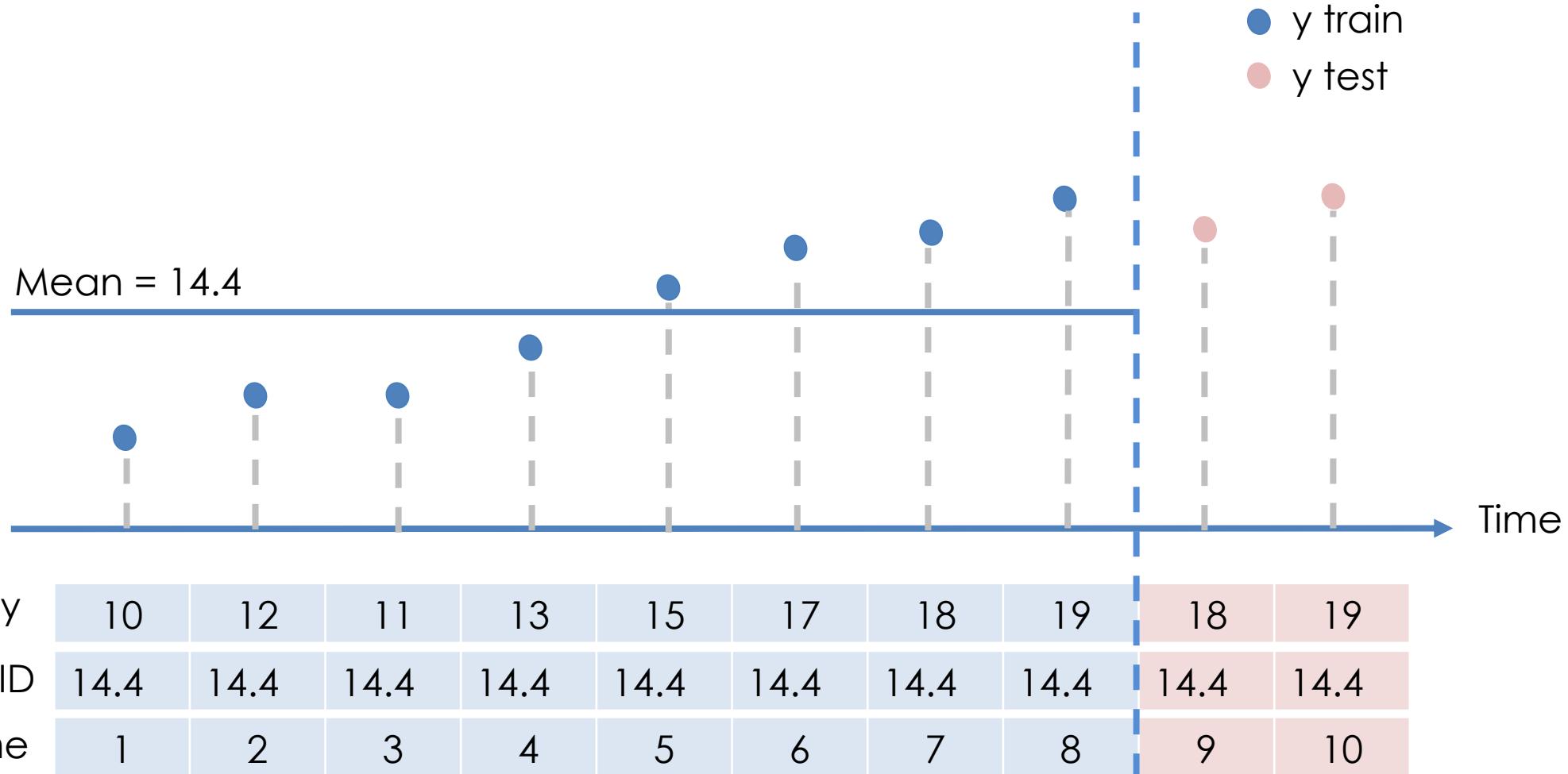
- No leakage between **train** and **test set**.
- But, the **target is leaked** from future values to past values in training set.
- Beware of overfitting.



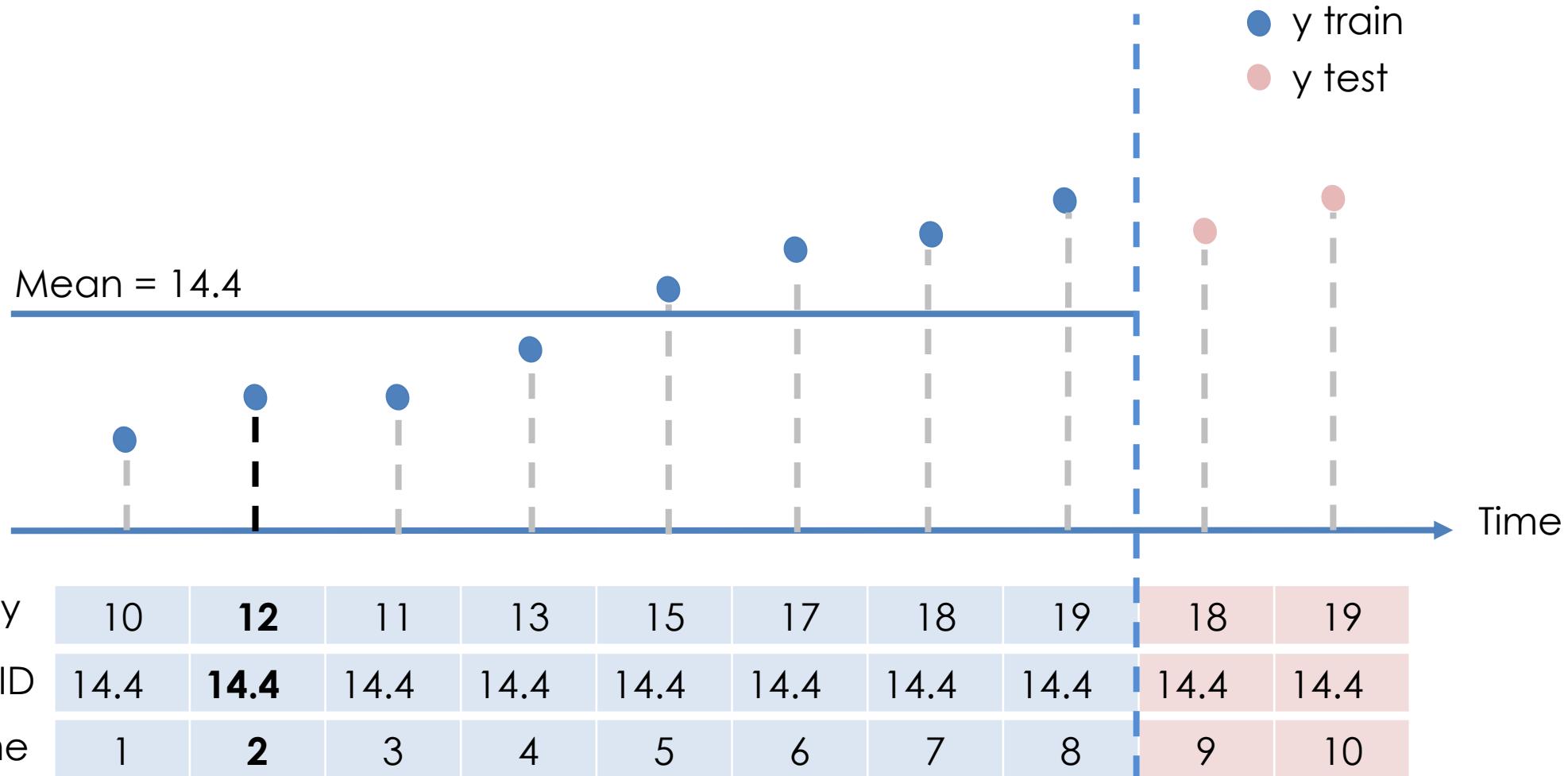
Static features: Target encoding



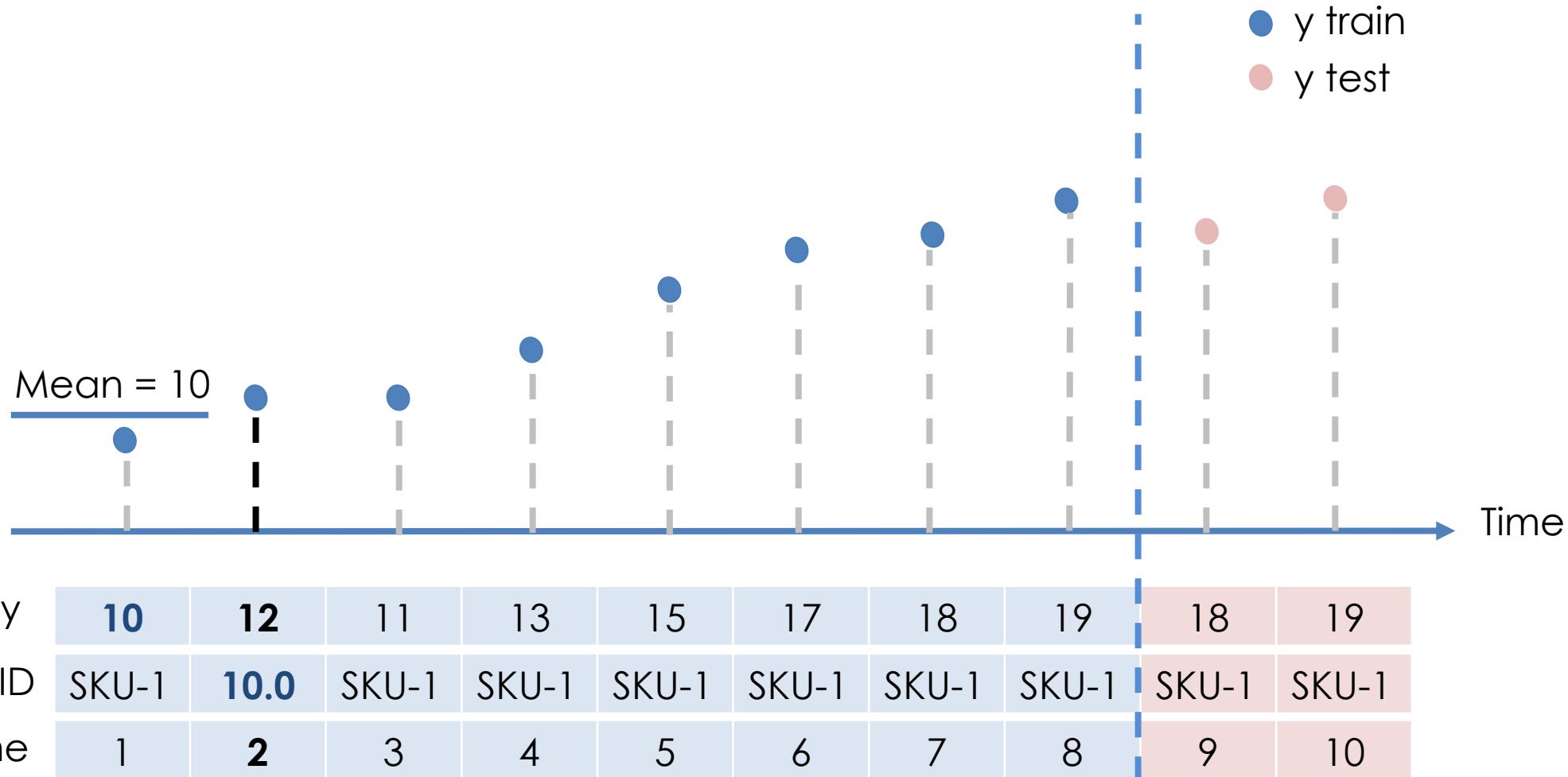
Static features: Target encoding



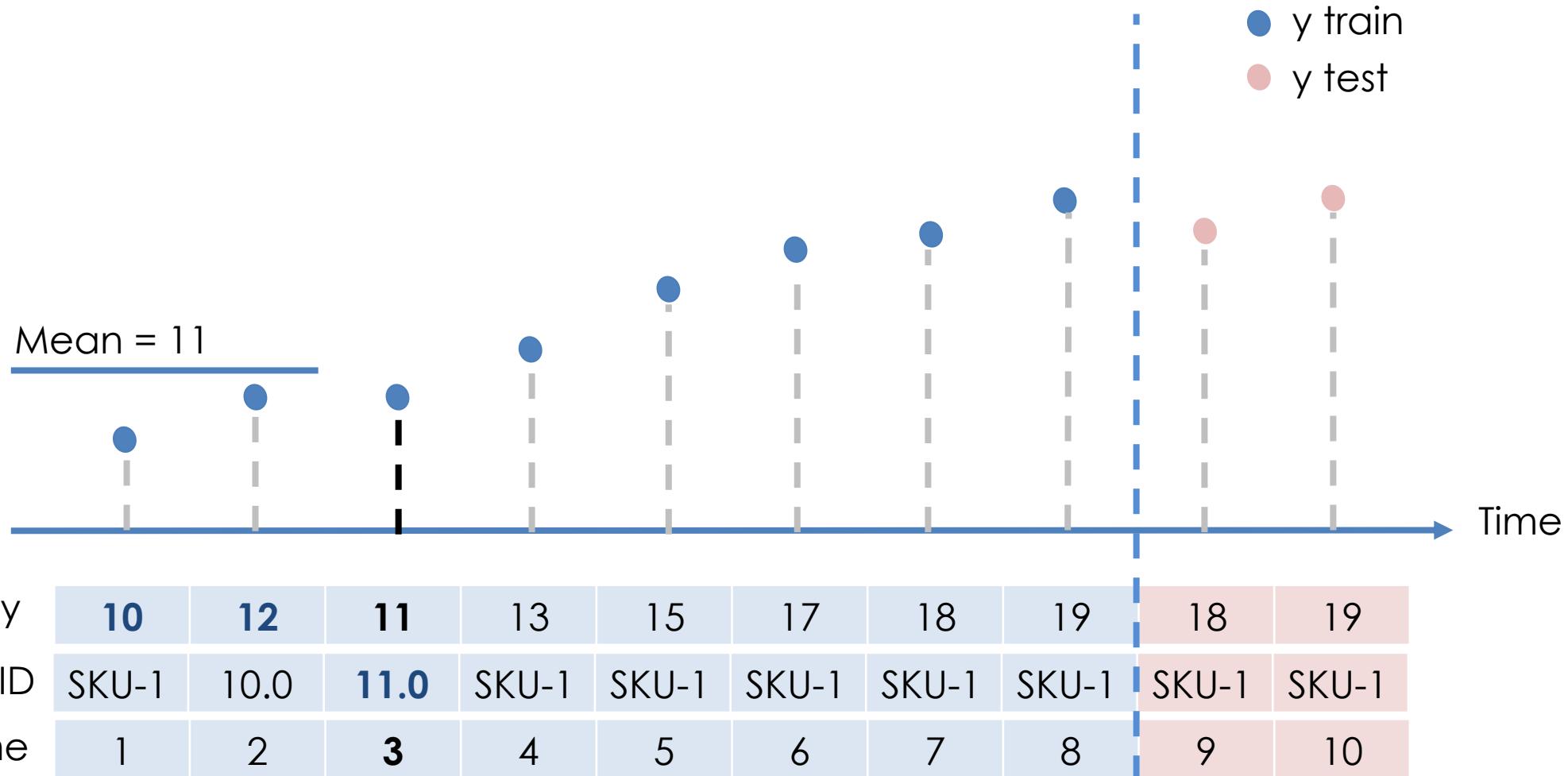
Static features: Target encoding



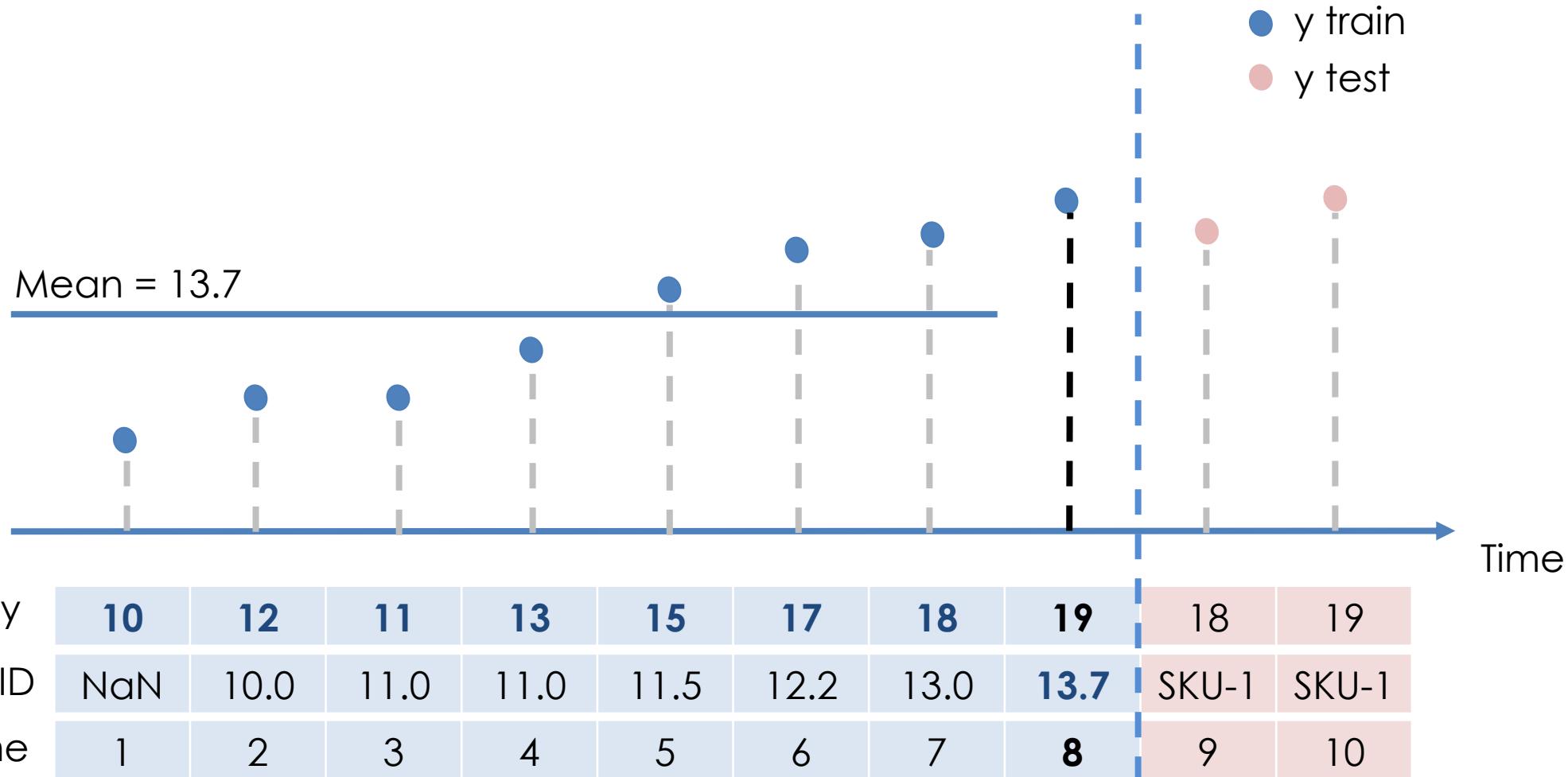
Static features: Target encoding



Static features: Target encoding



Static features: Target encoding



Static features: Target encoding

Time	Product ID (encoded)
...	...
2020-02-13	29
2020-02-14	31
2020-02-15	30
2020-02-16	28
2020-02-17	30
2020-02-18	SKU-1
...	...
2020-02-14	24
2020-02-15	26
2020-02-16	23
2020-02-17	24
2020-02-18	SKU-2

One step ahead-forecast

SKU-1

30 = mean

y_t
...
30
32
25
34

?
?

SKU-2

24 = mean

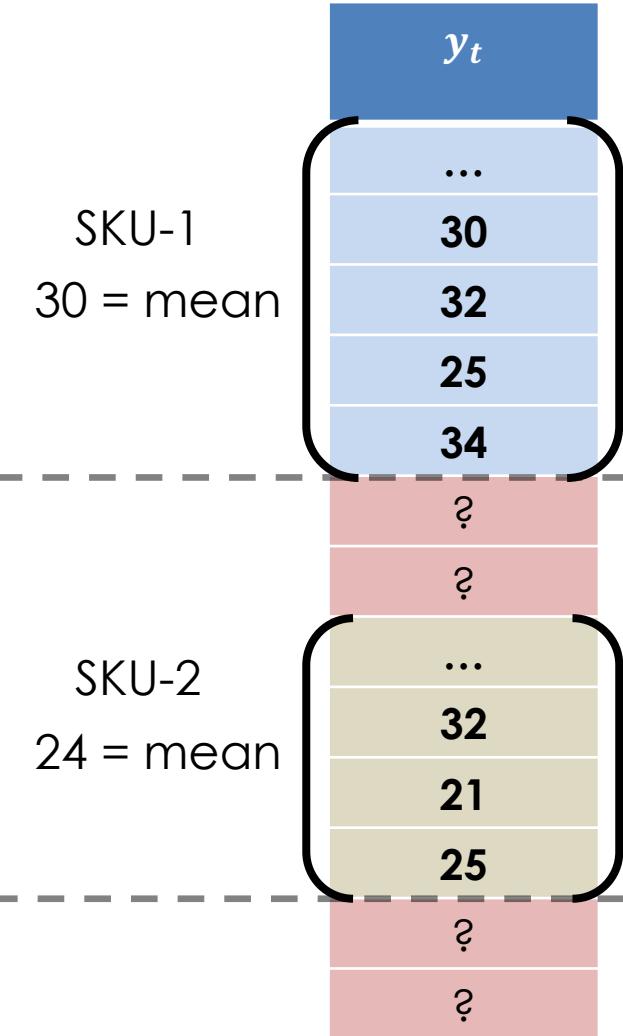
...
32
21
25

?
?

Static features: Target encoding

Time	Product ID (encoded)
...	...
2020-02-13	29
2020-02-14	31
2020-02-15	30
2020-02-16	28
2020-02-17	30
2020-02-18	SKU-1
...	...
2020-02-14	24
2020-02-15	26
2020-02-16	23
2020-02-17	24
2020-02-18	SKU-2

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

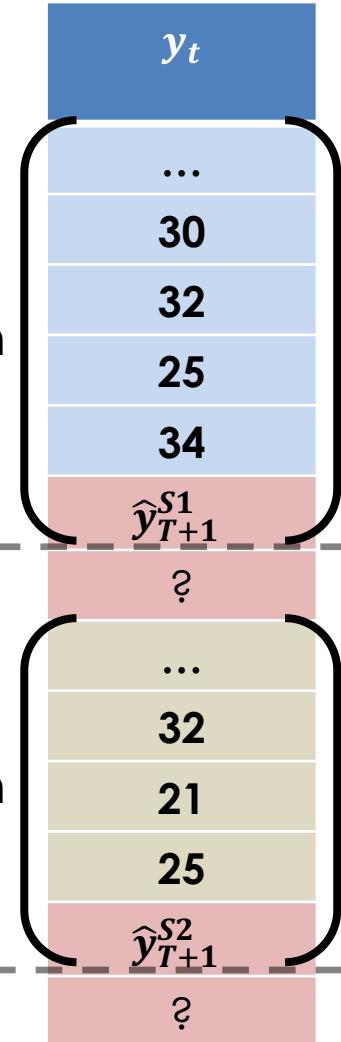


Static features: Target encoding

Time	Product ID (encoded)
...	...
2020-02-13	29
2020-02-14	31
2020-02-15	30
2020-02-16	28
2020-02-17	30
2020-02-18	SKU-1
...	...
2020-02-14	24
2020-02-15	26
2020-02-16	23
2020-02-17	24
2020-02-18	SKU-2

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

SKU-1 = 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



Time series forecasting using ML models



Features for time series forecasting



Useful libraries for forecasting with ML models

Overview of some useful libraries

Feature engineering & pre-processing



- Large number of time series features.
- Supports ML forecasting workflows.
- Multiple time series.
- Create time series features on top of Pandas.



Feature-engine

Forecasting workflow using ML on tabular data



- Recursive strategy.
- Some helper methods to create features.
- Multiple time series.
- Exogenous features.
- Time series cross-val.



- 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



```
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)
```

Period	y
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

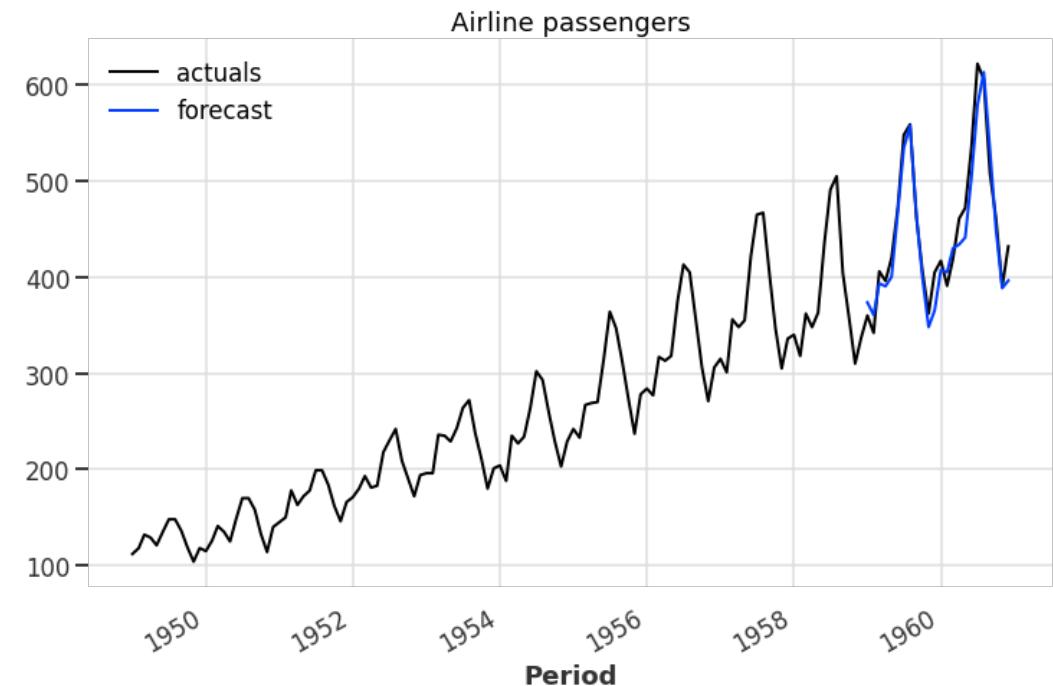
# Convert pandas DataFrame to TimeSeries
y = TimeSeries.from_series(df['y'])

# 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])
```

	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



```
# 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.

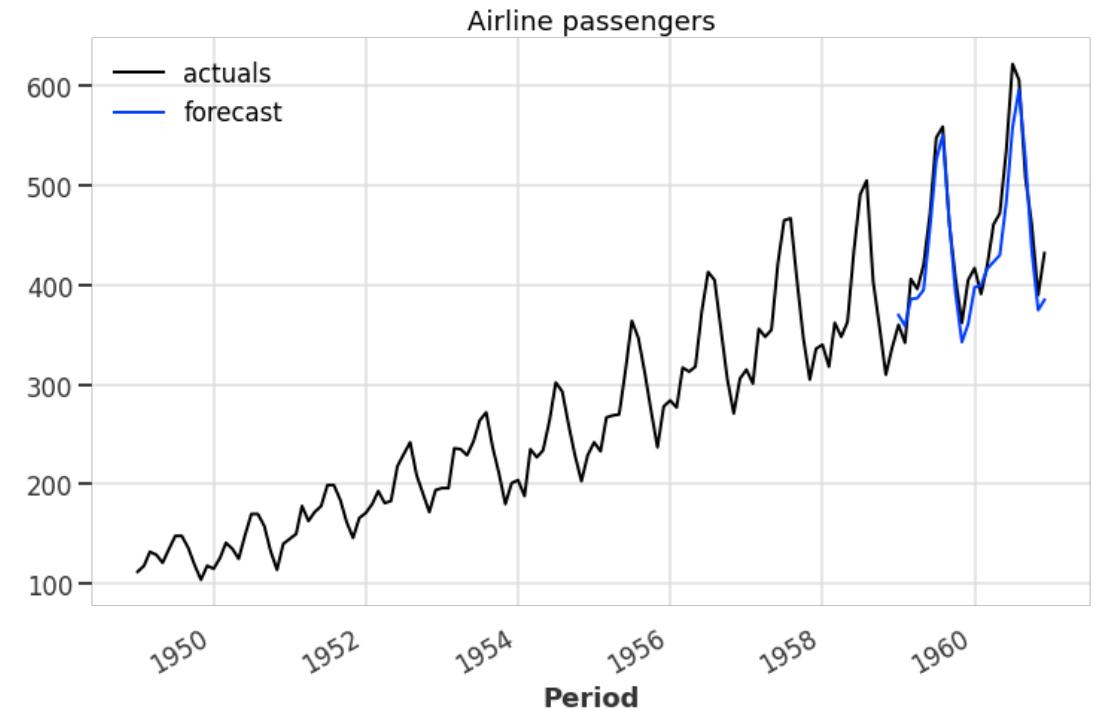
Forecasting with tabular data using Darts



```
# 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)

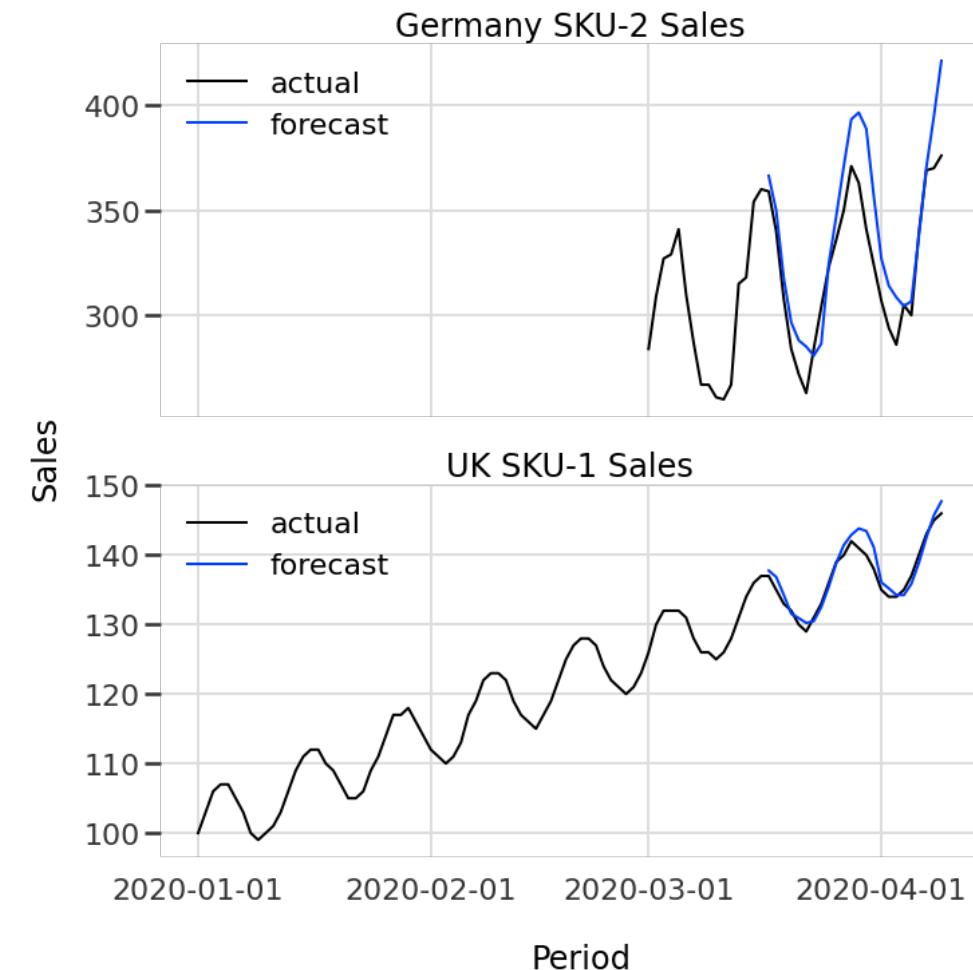
# Forecast any subset of time series
y_pred = model.predict(n=24,
    series=y_train_all[:2],
    future_covariates=future_cov_all[:2])
```

	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

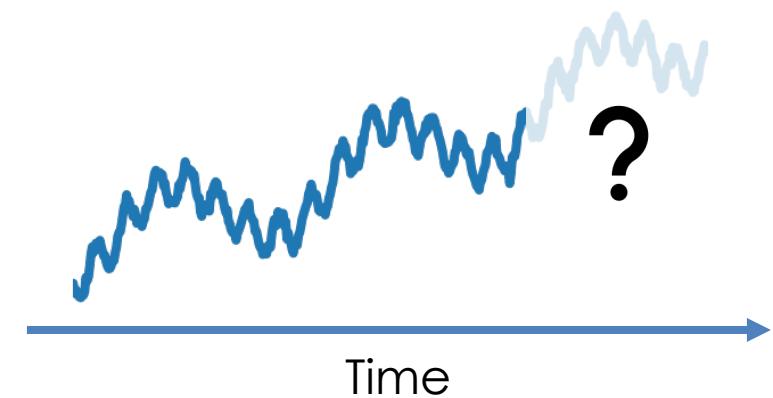
```
● ○ ●  
  
# Specify and train model  
model = RegressionModel(  
    lags=[-1, -2, -12],  
    lags_future_covariates=[0],  
    model=LinearRegression()  
)  
  
model.fit(y_train_all, future_covariates=future_cov_all)  
  
# Forecast any subset of time series  
y_pred = model.predict(n=24,  
    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

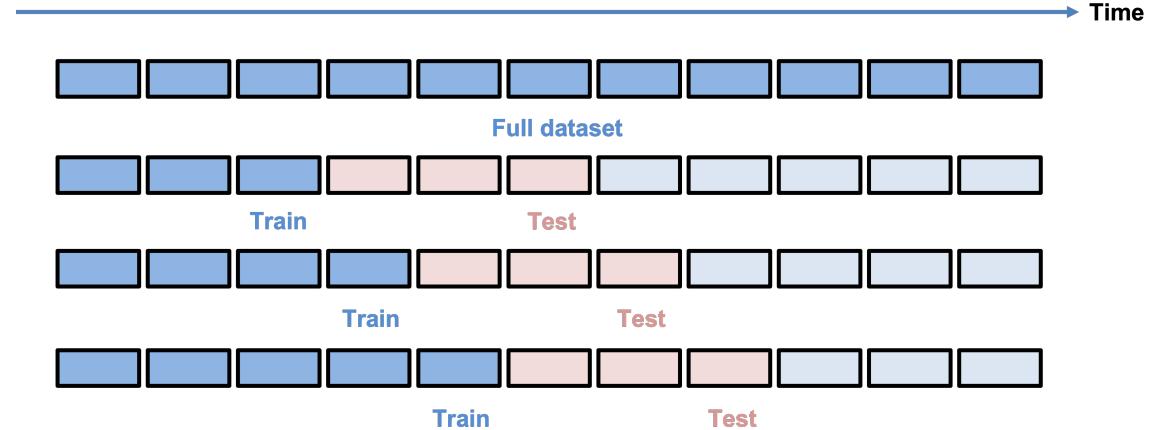
1. Forecasting can be treated as a tabular ML task and compete with statistical models.



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

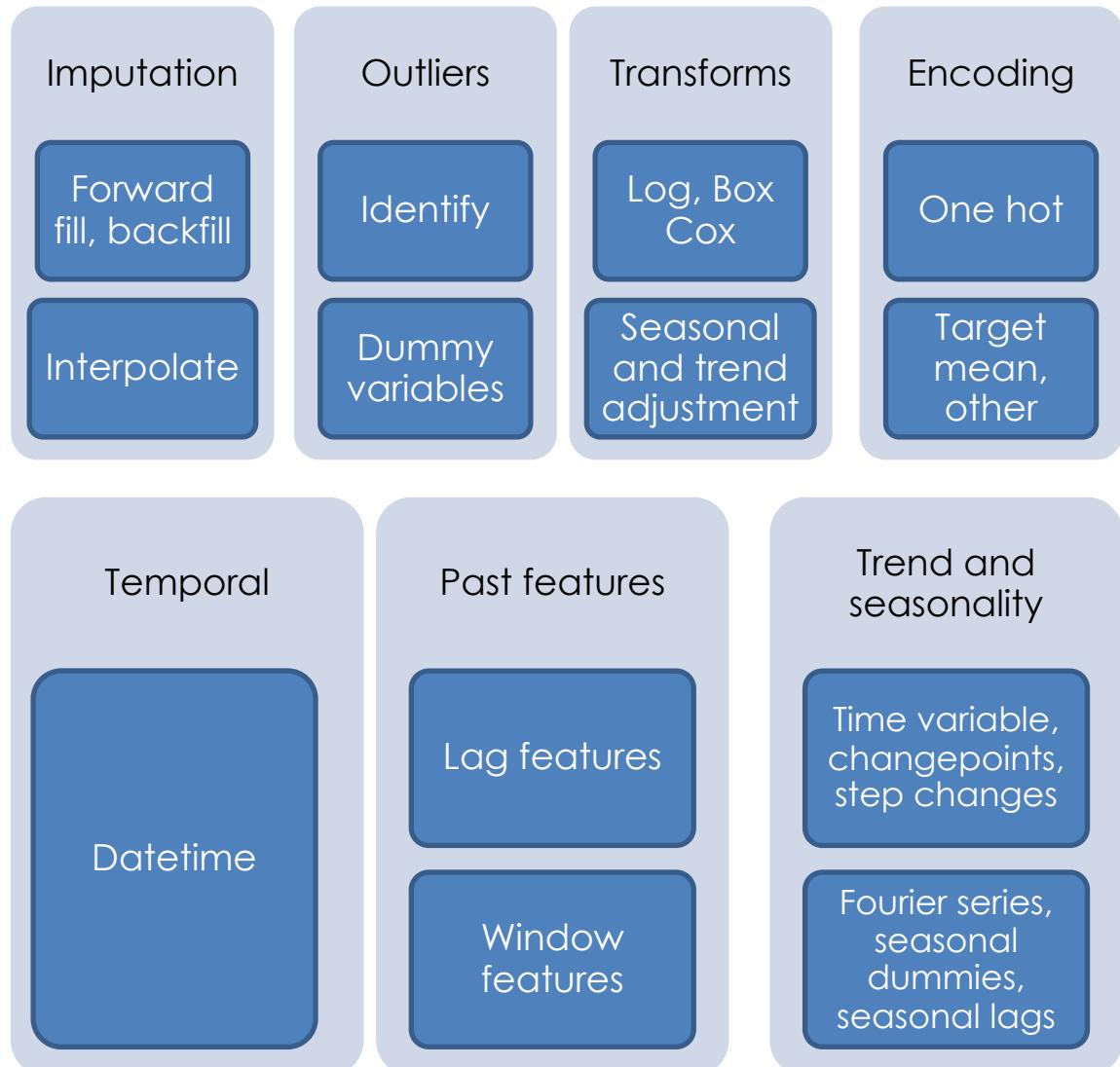
Conclusions

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.



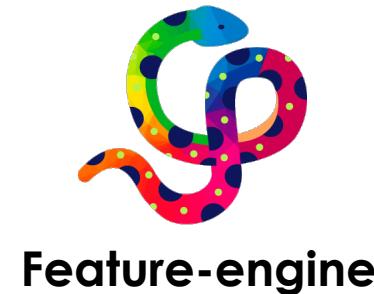
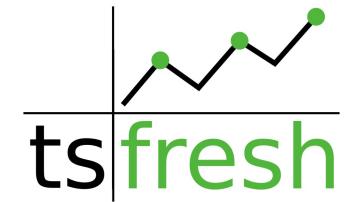
Conclusions

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.



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

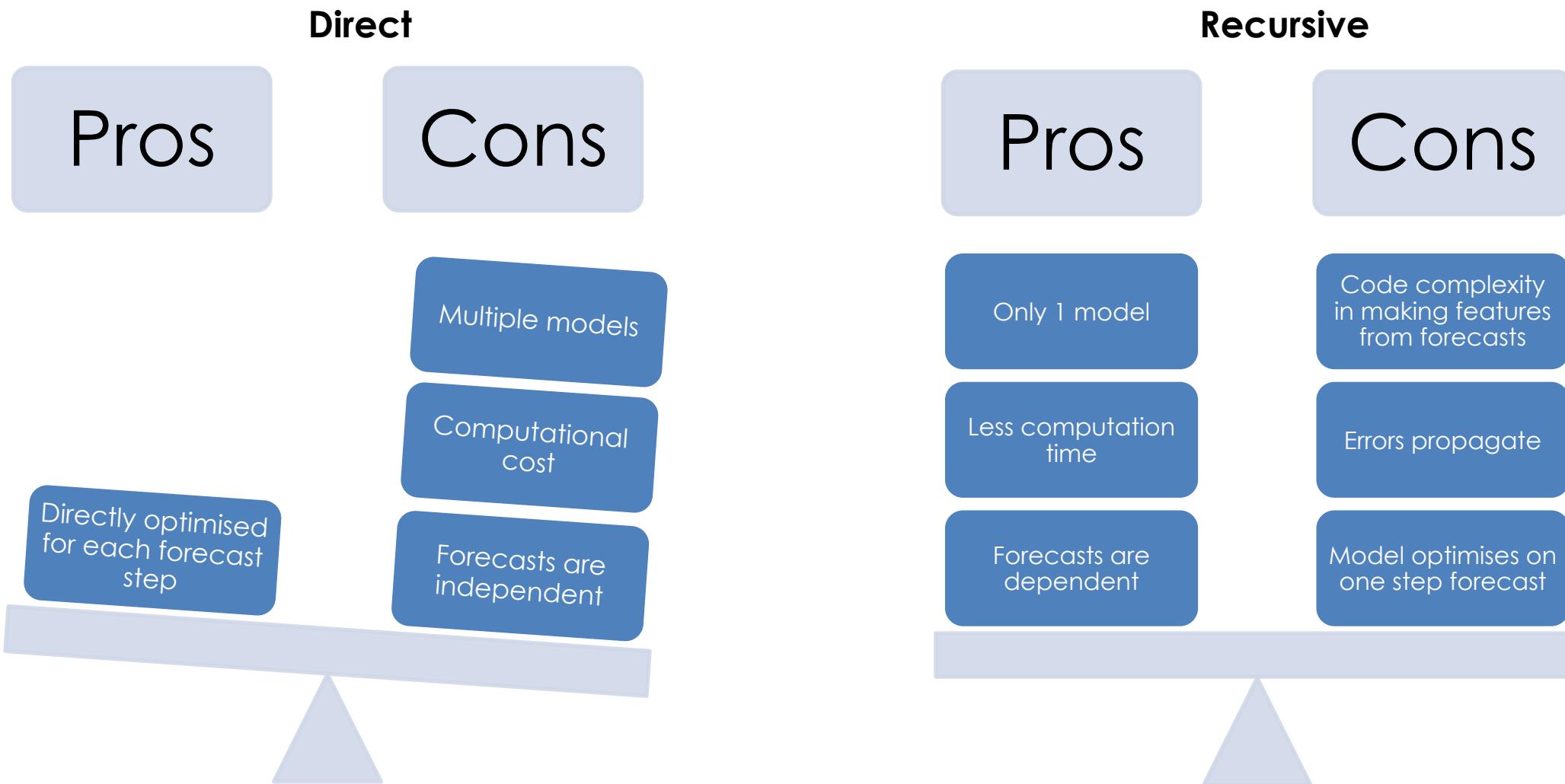
- [1] Makridakis, Spyros, Evangelos Spiliotis, and Vassilios Assimakopoulos. "The M5 competition: Background, organization, and implementation." *International Journal of Forecasting* (2021).
- [2] Makridakis, Spyros, Evangelos Spiliotis, and Vassilios Assimakopoulos. "M5 accuracy competition: Results,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

Direct vs recursive multistep forecasting



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

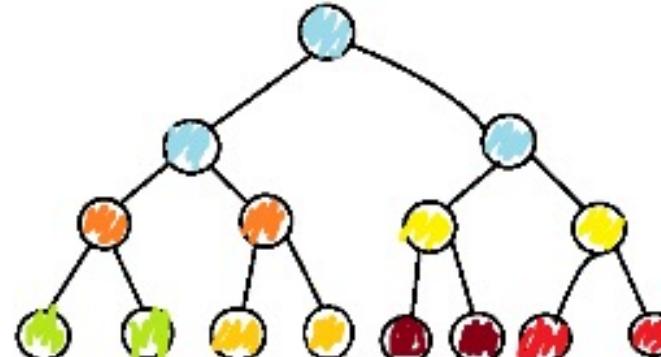
Time	Product ID (encoded)
...	...
2020-02-13	1
2020-02-14	1
2020-02-15	1
2020-02-16	1
2020-02-17	1
...	...
2020-02-14	2
2020-02-15	2
2020-02-16	2
2020-02-17	2

Map each category to an integer:

SKU-1 \leftrightarrow 1

SKU-2 \leftrightarrow 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]

DirRec

$$\text{Model 1: } \hat{y}_{T+1} = f_1(y_T, y_{T-1}, \dots)$$

$$\text{Model 2: } \hat{y}_{T+2} = f_2(\hat{y}_{T+1}, y_T, y_{T-1}, \dots)$$

...

$$\text{Model h: } \hat{y}_{T+h} = f_h(\hat{y}_{T+h-1}, \hat{y}_{T+h-2}, \dots, y_T, y_{T-1}, \dots)$$

Directly predict the whole output sequence

- Multi-output [4]

[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.