# Challenges in feature engineering

1. ML workflow
2. Challenges

* How am I going to make forecasts in new data?
* How am I going to send data to deploy the model?
* How do I make multistep forecasting?

1. Opportunities

* What can I bring forward from supervised learning strategies?
* How is forecasting different from supervised learning strategies?

1. Summary

* ML workflow
* supervised learning workflow
* Workflow in forecasting
* Multistep forecasting
* Why should I care?

# ML workflow

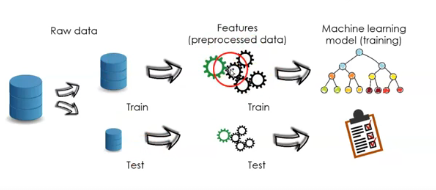
1. ML workflow

* Take raw data
* Separate into train and test set
* Extract features – learn parameters just from the train set and use them to extract features in the test set
* Train and evaluate the model
* Is this really possible in forecasting?

# Feature engineering in tabular data

1. ML workflow

* Tabular data – each row is an independent observation



1. Feature engineering – tabular data

A blue squares with white text

Description automatically generated

* Some feature transformations do not learn parameters from data
* Doesn’t matter when we split the data
* Imputation – Arbitrary
* Encoding – one hot
* Transformations – log, exp, combination
* Many feature transformations learn parameters from the data
* Best to split the data before feature engineering

1. Feature engineering libraries

* Fit() -> learns and stores parameters
* Transform() -> transforms data

1. Feature engineering pipelines

A screenshot of a computer code

Description automatically generated

* We can pass raw data to the pipeline and obtain a prediction
* We can deploy the pipeline to production

1. ML workflow forecasting

* Sometimes, the raw data does not contain the input features

1. Summary

* Many feature engineering procedures for tabular data learn parameters
* Best to split raw data before any transformation
* Feature transformation steps and ml model trained with a pipeline
* Pipeline can score and raw dataset and be deployed to production

# Feature engineering in forecasting – considerations

1. ML learning workflow: forecasting
2. Feature engineering in forecasting

A diagram of different types of information

Description automatically generated

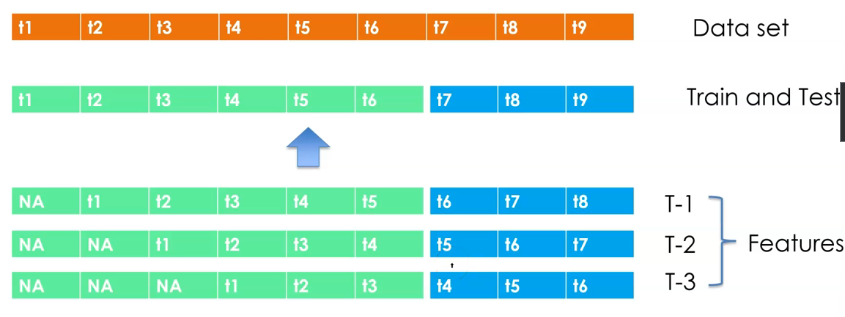
* Some transformations do not learn parameters:
* Encoding: one hot
* Transformation: log, exp, combination
* Temporal: datetime
* We can split the data or not
* Some transformations do learn parameters:
* Encoding: target mean, other
* Outliers: identify, capping
* Transformation: seasonality
* We should split the raw data
* Some transformations rely on past values:
* Imputation: last value forward, interpolation, etc.
* Past features: lag features, window features
* We can’t split the raw

1. Train test split

A screenshot of a number

Description automatically generated

* Lag features – rely on past values
* When we actually know the data -> bring value forward



* Window features – rely on past values
* When we actually know the data

A diagram of numbers and arrows

Description automatically generated

* Train test split in forecasting
* In forecasting – does not always work
* Careful with the data that we use to extract the features
* No standardized approach workflow

# Feature engineering in forecasting – pipelines

1. Traditional feature engineering methods

A screenshot of a computer

Description automatically generated

1. Can I still use pipelines?

* Yes

1. Summary

* Some features can’t be created after we split the data
* We can use all data available at the time of pred iction
* No unified approach to create features from past data

# Multistep forecasting

1. Forecasting framework: single point

* We have data upto a certain point in time
* We want to forecast 1 single point ahead
* E.g. forecast sales next day

1. Forecasting framework: multiple points

* We have data up to a certain point in time
* We want to forecast various points ahead
* E.g., forecast sales next 3 days

1. Multi-step forecasting approaches

* Direct: one model to predict each forecast step
* Recursive: forecast as inputs for next step

1. Multistep forecasting

* We want to predict multiple steps ahead
* We have data up to the first point of forecast

# Direct multistep forecasting

1. Direct forecasting

* 1 model per forecasting point -> multiple models
* 1 target per forecasting point
* Each forecasting point is estimated independently -> computational cost & forecasts are independent
* Less estimation variance – we are not using predictions to make further predictions in the future
* Does not propagate estimation errors

# Recursive multistep forecasting

1. Recursive forecasting

* Use forecast output as new input
* Recursively apply a 1-step ahead forecast model
* Only 1 time series model -> less computation time
* Each forecasting point is estimated using previous forecasts -> propagates the errors
* More code to take the forecasts as inputs, and recreate the features