SDMM
Small Data Many Models

Szymon Urbański
Dominik Lewy

27/03/2019

Big Data - what is it?

Why this is not our case?

The FOUR V's of Big Data

From traffic patterns and music downloads to web history and medical records, data is recorded, stored, and analyzed to enable the technology and services that the world relies on every day. But what exactly is big data, and how can these massive amounts of data be used?

As a leader in the sector, IBM data scientists break big data into four dimensions: **Volume, Velocity, Variety and Veracity**

Depending on the industry and organization, big data encompasses information from multiple internal and external sources such as transactions, social media, enterprise content, sensors and mobile devices. Companies can leverage data to adapt their products and services to better meet customer needs, optimize operations and infrastructure, and find new sources of revenue.

By 2015
4.4 MILLION IT JOBS
will be created globally to support big data, with 1.9 million in the United States

Volume

SCALE OF DATA

40 ZETTABYTES
[43 TRILLION GIGABYTES]
of data will be created by 2020, an increase of 300 times from 2005

2005

2020

It's estimated that
2.5 QUINTILLION BYTES
[2.3 TRILLION GIGABYTES]
of data are created each day

6 BILLION PEOPLE
have cell phones

WORLD POPULATION: 7 BILLION

Most companies in the U.S. have at least
100 TERABYTES
[100,000 GIGABYTES]
of data stored

Velocity

ANALYSIS OF STREAMING DATA

The New York Stock Exchange captures
1 TB OF TRADE INFORMATION
during each trading session

Modern cars have close to
100 SENSORS
that monitor items such as fuel level and tire pressure

By 2016, it is projected there will be
18.9 BILLION NETWORK CONNECTIONS
– almost 2.5 connections per person on earth

Variety

DIFFERENT FORMS OF DATA

As of 2011, the global size of data in healthcare was estimated to be
150 EXABYTES
[161 BILLION GIGABYTES]

By 2014, it's anticipated there will be
420 MILLION WEARABLE, WIRELESS HEALTH MONITORS

4 BILLION+ HOURS OF VIDEO
are watched on YouTube each month

30 BILLION PIECES OF CONTENT
are shared on Facebook every month

400 MILLION TWEETS
are sent per day by about 200 million monthly active users

Veracity

UNCERTAINTY OF DATA

1 IN 3 BUSINESS LEADERS
don't trust the information they use to make decisions

Poor data quality costs the US economy around
$3.1 TRILLION A YEAR

27% OF RESPONDENTS
in one survey were unsure of how much of their data was inaccurate

Sources: McKinsey Global Institute, Twitter, Cisco, Gartner, EMC, SAS, IBM, MEPTEC, QAS

IBM

BI PLATFORMS

- Microsoft
- aws
- DOMO
- salesforce Wave Analytics
- looker
- THOUGHTSPOT
- ARCADIA DATA
- ATSCALE
- Information Builders
- SISENSE
- GoodData
- MicroStrategy
- birst

VISUALIZATION

- tableau
- SAP Lumira
- Google Cloud
- celonis
- Qlik
- Periscope Data
- ZEPL
- ZOOMDATA
- plotly
- CHARTIO
- TOUCAN TOCO

MACHINE LEARNING

- Azure Machine Learning
- aws
- Google Cloud
- H2O.ai
- DataRobot
- gamalon
- ELEMENT AI
- VERSIVE
- ViSENZE
- deepsense.io
- bonsai

Open Source

- TensorFlow
- theano
- torch
- MADlib.
- learn
- Caffe
- Microsoft Cognitive Toolkit
- OpenAI
- DMTK
- Keras
- PaddlePaddle
- Apache SINGA
- FeatureFu
- mxnet
- Chainer
- VELES
- WEKA The University of Waikato
- DIMSUM
- neon™
- DSSTNE
- mllib
- DL4J
- MAHOUT
- Aerosolve

http://mattturck.com/bigdata2018/

	Locally in DSS	In Hadoop / Spark	In SQL Database	In Kubernetes / Docker
Visual Preparation Design	In-memory Sample	N/A	N/A	N/A
Visual Preparation Execution	YES Streaming	YES Spark	YES	N/A
Visual Recipes (other than Prepare)	YES Streaming or disk-copy	YES Hive, Spark, Impala	YES	N/A
Python and R recipes	YES In-memory or streaming	YES PySpark, SparkR, sparklyr	Custom code with DSS helper API	YES In-memory or streaming
Spark-Scala recipe	N/A	YES	N/A	N/A
Charts	YES	YES Hive, Impala (most charts)	YES (most charts)	N/A
Machine Learning train	YES scikit-learn, XGBoost, Keras/Tensorflow	YES MLlib, Sparkling Water	YES Vertica ML	YES scikit-learn, XGBoost, Keras/Tensorflow
Machine Learning execution	YES scikit-learn, XGBoost, MLlib, Keras/Tensorflow	YES scikit-learn, XGBoost, MLlib, Sparkling Water	YES scikit-learn (partial), XGBoost, MLlib (some models), Vertica ML	YES scikit-learn, XGBoost, MLlib, Keras/Tensorflow
Python, R, Scala notebooks	YES In-memory or streaming	YES Spark-Scala, PySpark, SparkR, sparklyr	Custom code with DSS helper API	YES
SQL-like recipe or notebook	N/A	YES Hive, Impala, Pig, SparkSQL	YES	N/A

Source: https://www.dataiku.com/learn/guide/getting-started/dss-concepts/where-does-it-all-happen.html

SDMM - Small Data Many Models

16 000 SKUs on a few markets

6 model types

seasonality

holidays

promotions

media

16k*6*2*2*2*2= **1 536 000** models

Econometrics

↓

Machine Learning

Small Data Many Models: Typical Use Cases

Time Series → forecasting, MMM

Modeling on aggregate data → MMM, CRM data, macroeconomic (money supply), web analytics without cookies

Longitudinal data → household panels, per country, macroeconomic

Optimization → supply chain optimization, product assortment, NPI

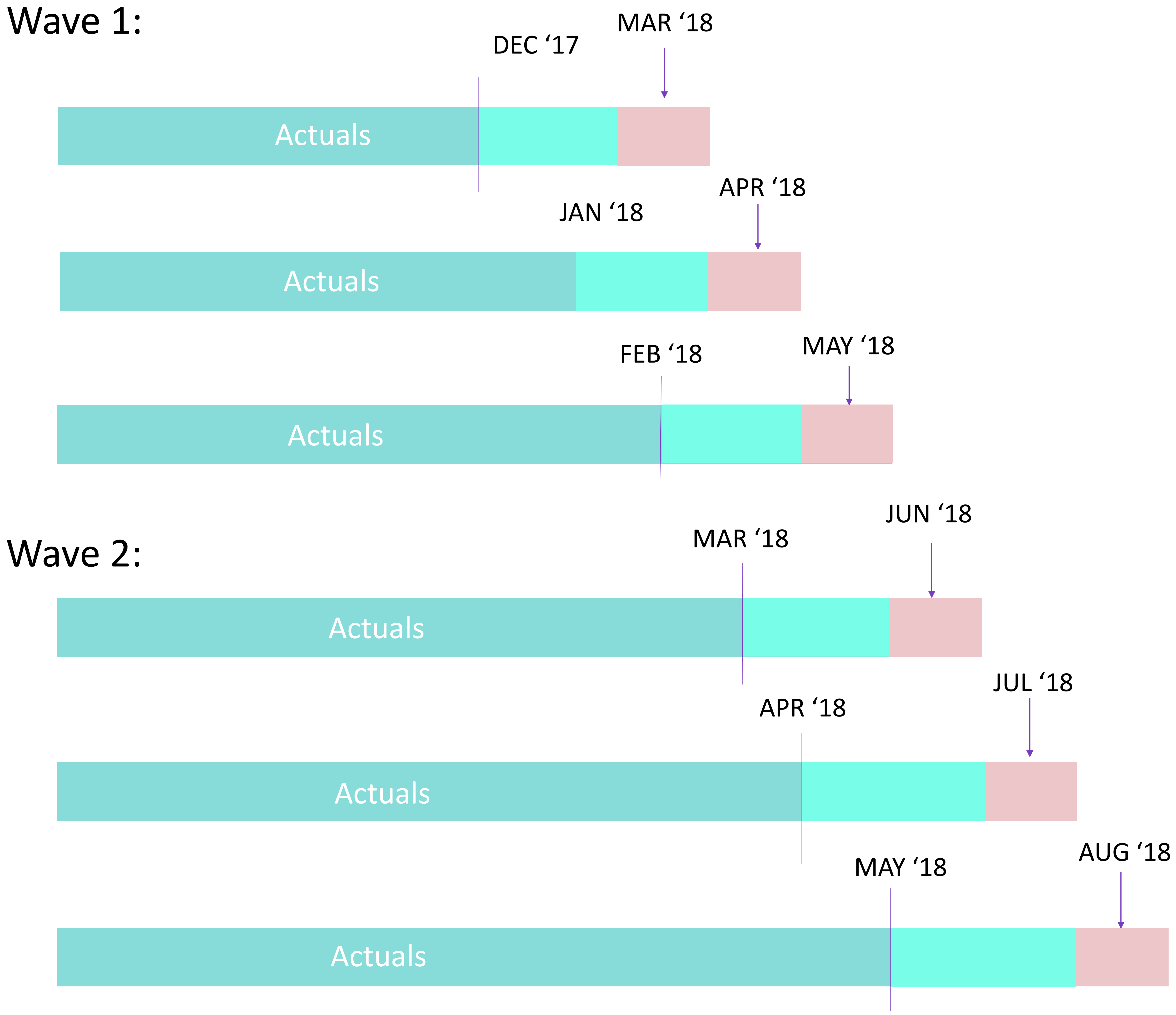
Rare phenomena → earthquakes, floods, crime

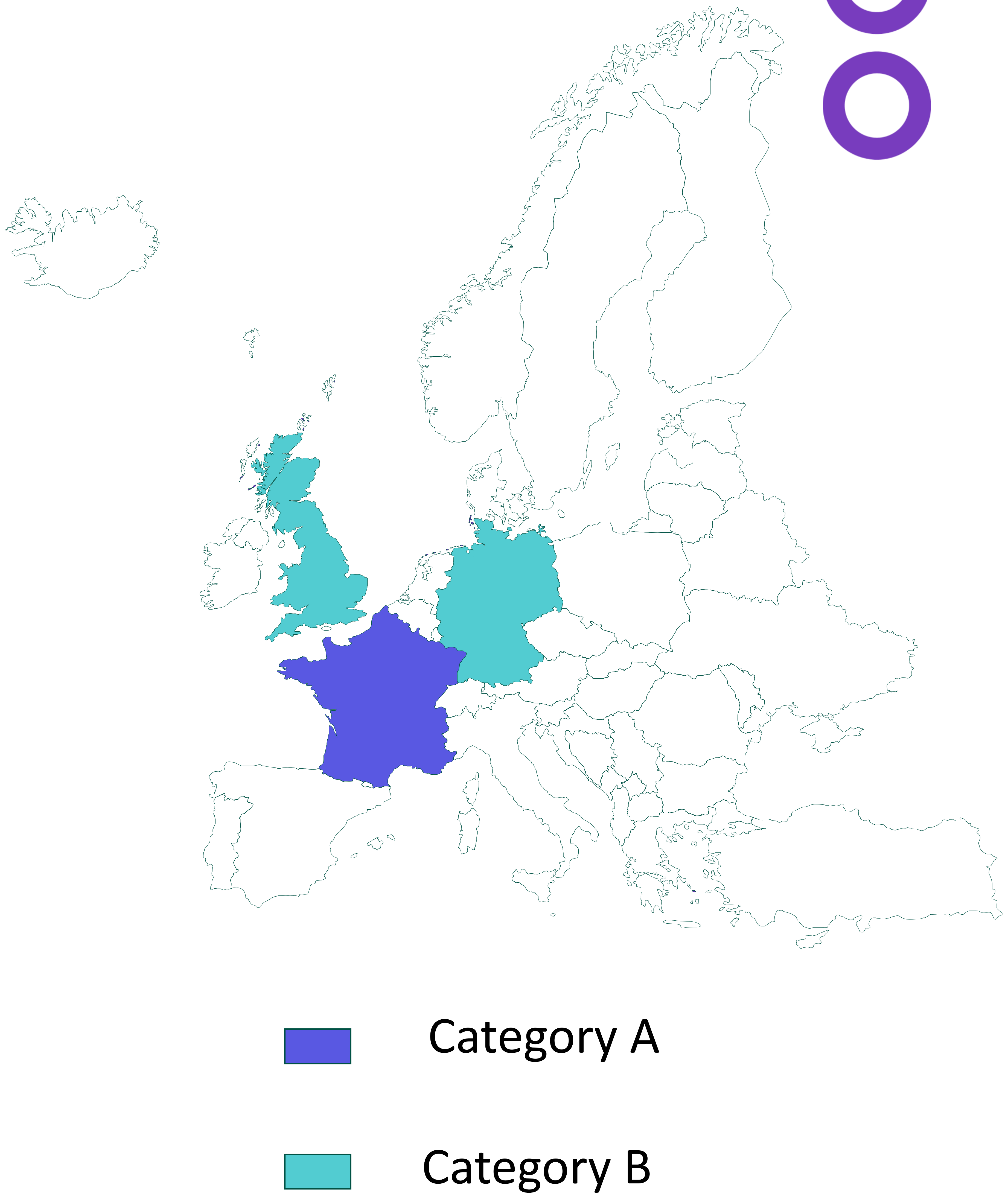
Small Data Many Models: Problems

Over-fitting (#features vs #observations)	Cross-validation
More feature engineering – be smarter with the data (take care of degrees of freedom)	Aggregate variables, reduce variables, example: holidays
Outliers	Hard to do in time-series (omitting observations, especially when variables are time-dependant) – rather have to understand WHY outlier is an outlier
Even more feature engineering - functions	Apply functions that tell more about business (for example log the price)

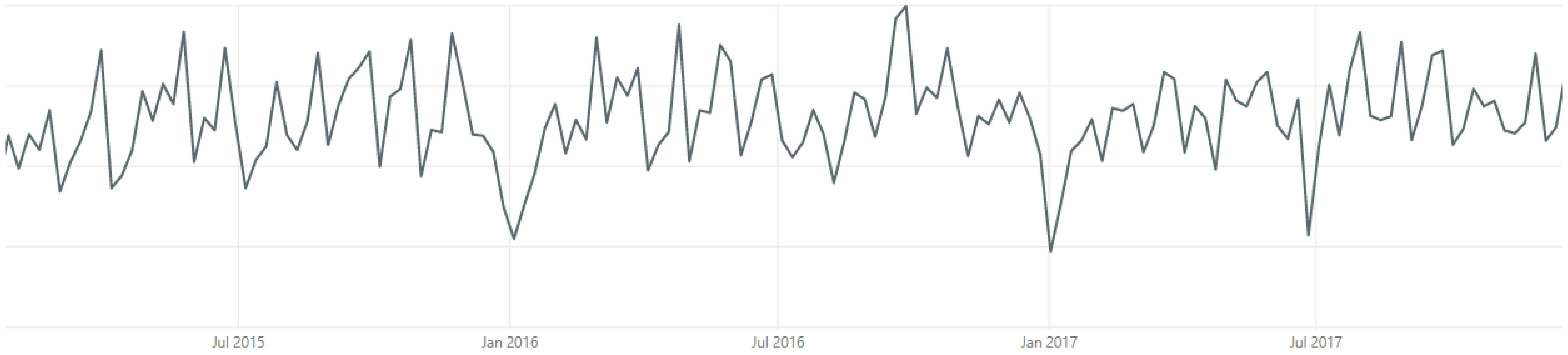
Business background

Problem – Shipments at a multinational FMCG company





Problem – product diversification



A-list products

Seasonal products

New Product Launch

Error and Bias

base code	materials	SKU % in BC	Actuals	Forecast	forecast per material	Actual-forecast	abs(Actual-forecast)
A	1	50%	500	1100	550	-50	50
	2	50%	500		550	-50	50
B	3	20%	400	1900	380	20	20
	4	80%	1600		1520	80	80
C	5	50%	1500	3100	1550	-50	50
	6	50%	1500		1550	-50	50
			Σ	6100		Σ -100	Σ 300

SFE

$$\frac{300}{6100} = 0.05$$

BIAS

$$\frac{-100}{6100} = -0.02$$

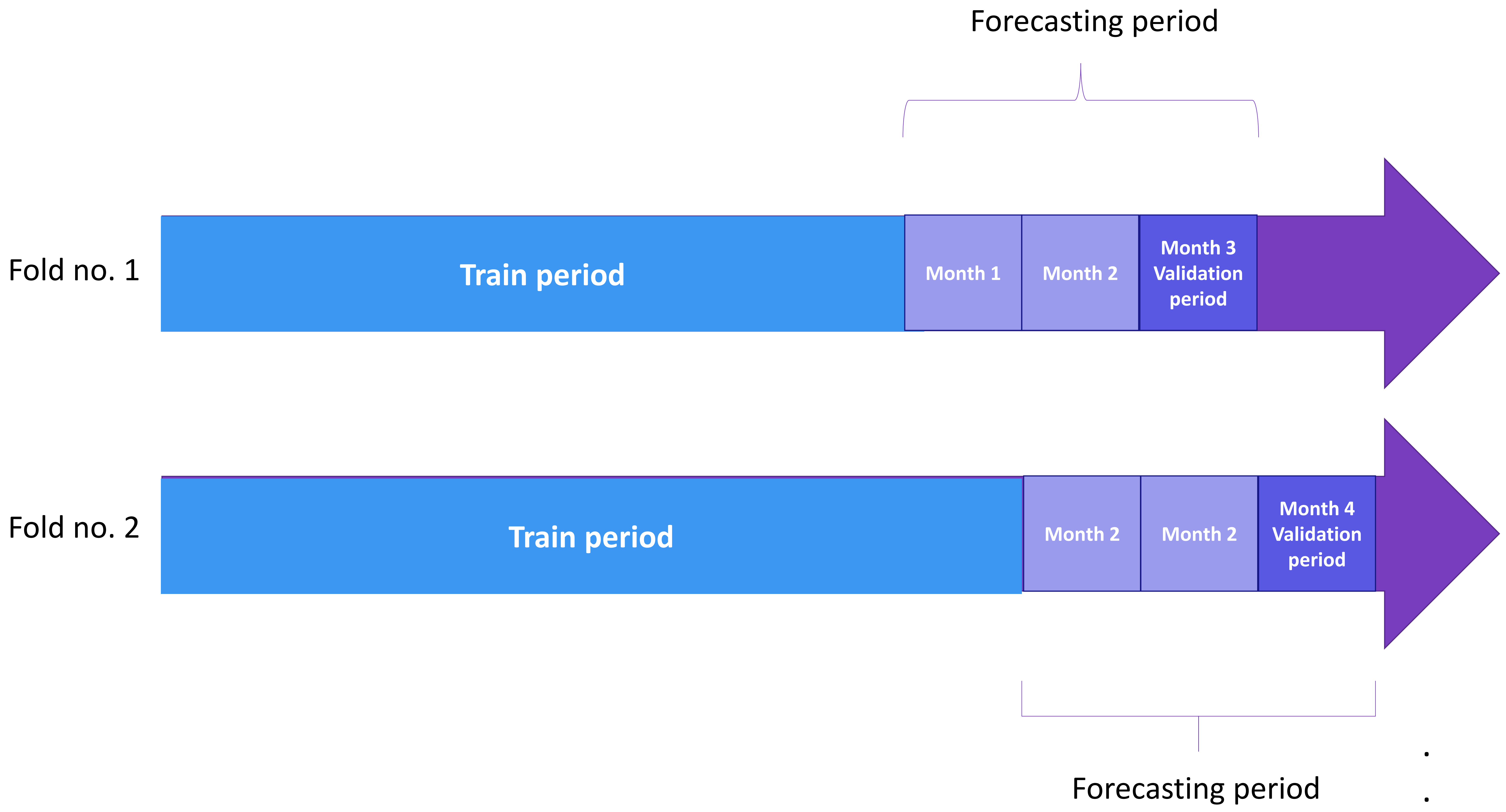
Glossary

OoS (Out of Stock) – temporary lack of products at shelf.

NPI (Non-productive Inventory) – goods stored at your warehouse that cannot be sold for some reason: expiration time to short for the shop to accept the goods, damaged packaging or some aspects of the packaging or bottle/container not meeting current regulations.

Cross validation for time series

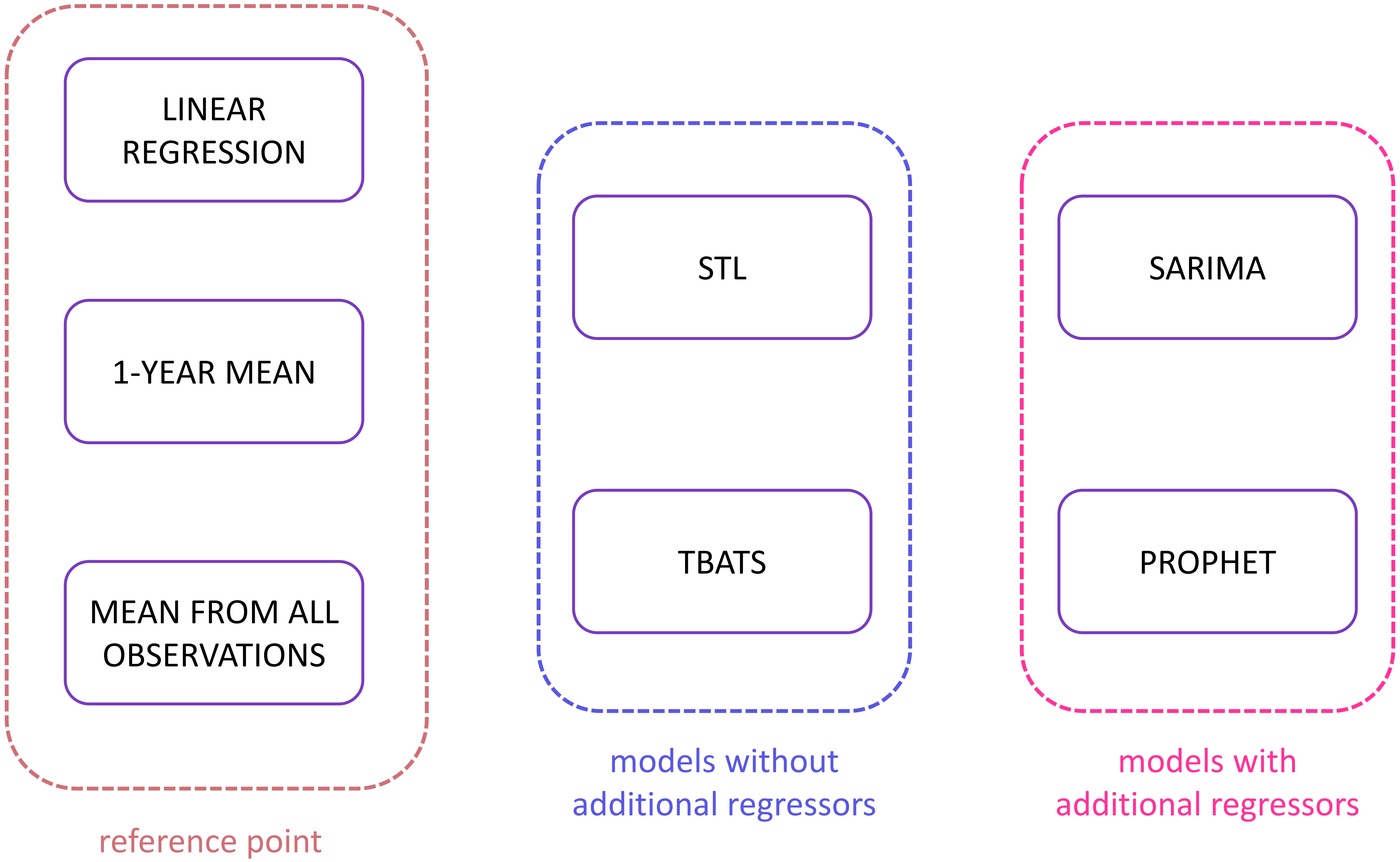
How to be sure that we're choosing the best model?



- The structure of cross validation reflects business cycle
- Minimum: 2 years of training

.
.
.
(till the end of the project we calculated 17 folds)

Model spectrum



If more advanced models don’t perform better than simple models used as reference points, this may indicate that time series is similar to white noise and no model will be able to forecast with available information.

Models

STL - “**Seasonal** and **Trend** decomposition using Loess” (Loess is a method for estimating nonlinear relationships)

STL can handle any type of seasonality, which might change over time. The method can be robust to outliers so that occasional unusual observations do not affect the estimates of the trend-cycle and seasonal components. STL, however, doesn’t handle trading day or calendar variation automatically.

STL

TBATS

models without additional regressors

TBATS model is a variations of an exponential smoothing state space model with a Box-Cox statistical transformation using ARMA (autoregressive and moving average) errors. A time series decomposition model consists of **decomposing a time series into trend, seasonal, cyclical, and irregular components**. In a TBATS model the seasonality is allowed to change slowly over time. One drawback of TBATS models, however, is that they can be slow to estimate, especially with long time series.

Models

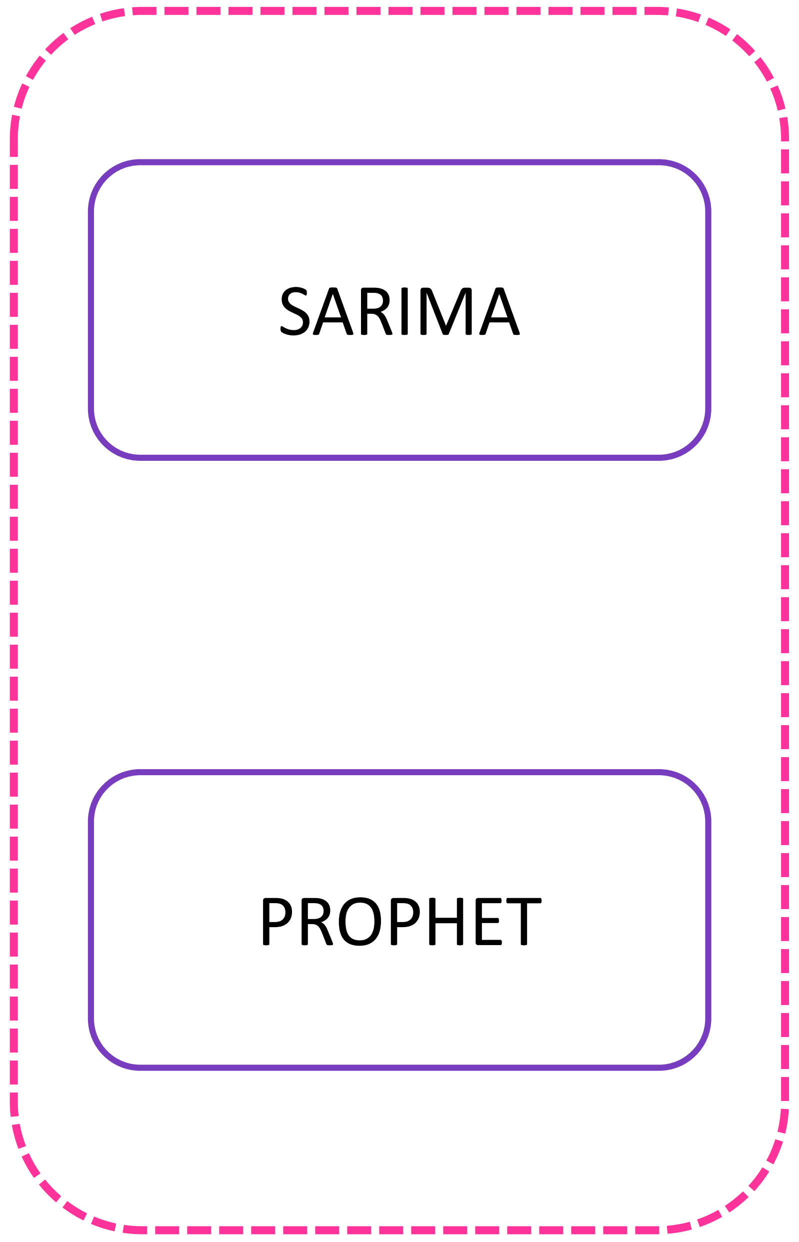
$$\text{ARIMA} \quad \underbrace{(p,d,q)}_{\substack{\uparrow \\ \left(\begin{array}{l}\text{Non-seasonal part}\\ \text{of the model}\end{array}\right)}} \quad \underbrace{(P,D,Q)_m}_{\substack{\uparrow \\ \left(\begin{array}{l}\text{Seasonal part}\\ \text{of the model}\end{array}\right)}}$$

SARIMA

- The auto-regressive parameter *p* specifies the number of lags used in the model
- The *d* represents the degree of differencing (subtracting of its current and previous d times) in the integrated **(***I(d)***)** component
- A moving average (MA(q)) component represents the error of the model as a combination of previous error terms e_t. The order *q* determines the number of terms to include in the model.
- $(P,\ D,\ Q)_m$ parameters describe the seasonal component of *m* periods.

PROPHET

- Decomposes time series into three main model components: trend, seasonality, and holidays.
- This specification is similar to a generalized additive model (GAM), a class of regression models with potentially non-linear smoothers applied to the regressors.
- Within this approach the forecasting problem is framed as a curve-fitting exercise, which is inherently different from time series models that explicitly account for the temporal dependence structure in the data



models with additional regressors

Holidays

Input data

Country	Date	Weekday	Holiday Name	Holiday Type
France	2018 Jan 1	Monday	New Year's Day	National holiday
France	2018 Mar 20	Tuesday	March equinox	Season
France	2018 Mar 25	Sunday	Daylight Saving Time starts	Clock change/Daylight Saving Time
France	2018 Mar 30	Friday	Good Friday	Local holiday
France	2018 Apr 1	Sunday	Easter Day	Observance
France	2018 Apr 2	Monday	Easter Monday	National holiday

Data gathered from https://www.timeanddate.com/holidays/ :

- France: 25 holidays per year
- Germany: 72 holidays per year
- UK: 90 holidays per year

Data used for modelling

Country	#Holidays
France	16
Germany	18
UK	15

- Mainly bank holidays and “potentially sweet” special occasions (Mother’s Day or Valentine’s Day)
- Creating holiday vector, which counts number of holidays per week

Promo vector

Customer	Category	Brand	SKU	Promo	Start	End
Sainsbury	Chocolate	A	A1234	TPR	11/07/2018	1/08/2018
…	…	…	…	…	…	…

This is the format that is required to track and manage promotions, however it is not perfect for modeling. It needs to be transformed into a promotional vector which tells about the intensity of promotions for a particular SKU in a week. Examples:

- Simple count of promotions each week
- Count of promotions weighted by the associated volume of the order
- Count of promotions weighted by the share of customer in market

Additionally the promotional vector can be processed by an analytical function:

- Square of the vector – to amplify the impact of big promotions

Approaches

Summary table

	Weekly	Monthly
National	this approach contains weekly actuals aggregated to national level (all customers for one base code)	this approach uses actuals on the most general level – contains actuals aggregated to month and national level (all customers for one base code)
Customer	this approach is the most detailed one: contains weekly actuals and actuals aggregated to customer level	*NA*

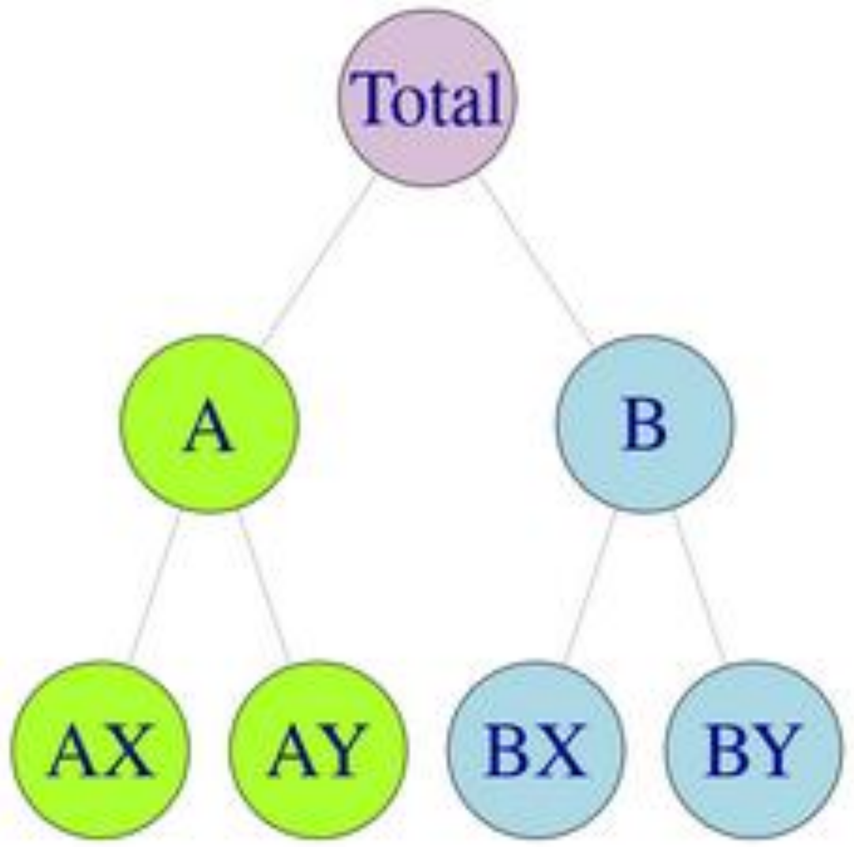
Based on them – two additional:

- **Reconciliation –** this approach combines customer and national approach through its hierarchical nature to decrease MAE
- **Ensemble** - this approach uses all above and is counted as an average of them (on national monthly level to enable comparison)

Reconciliation

4 methods of reconciliation:

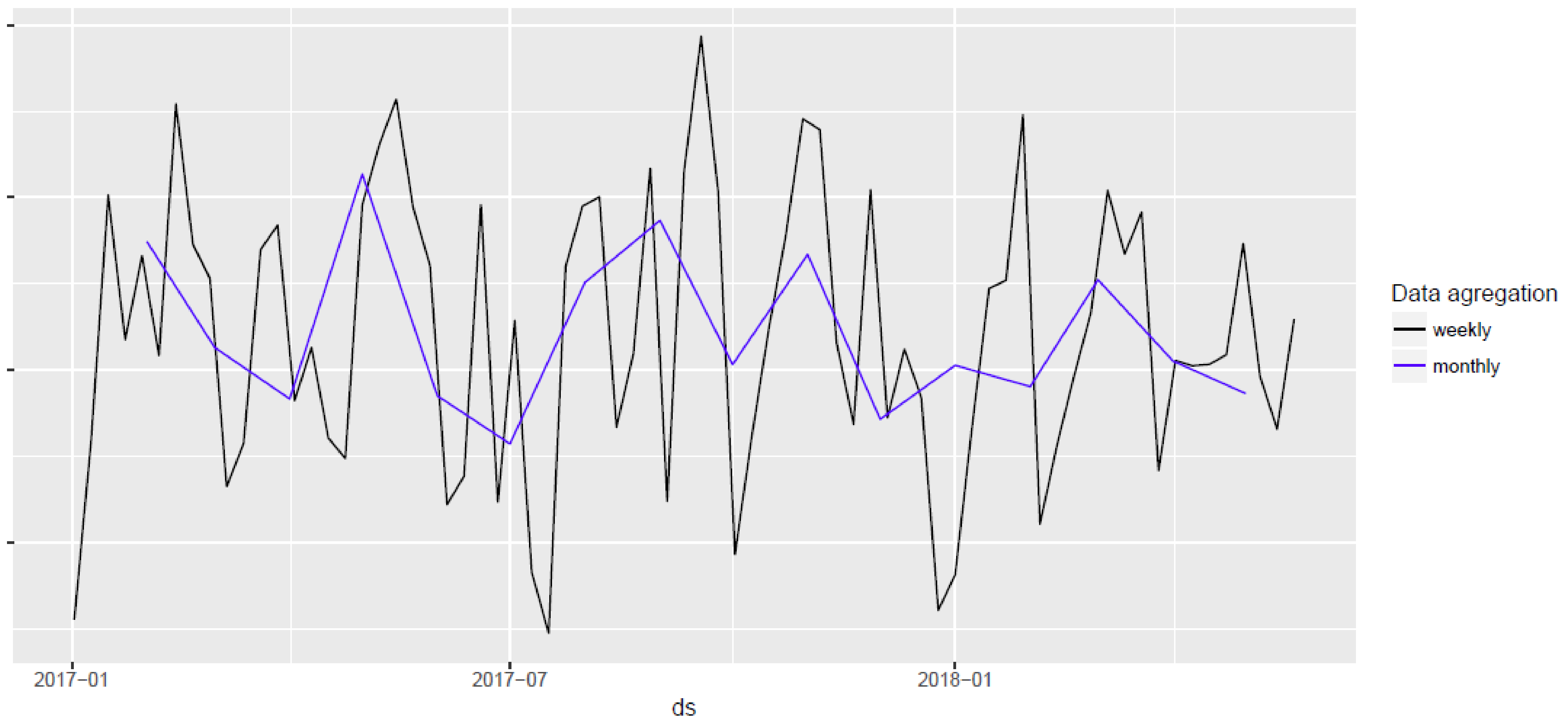
- **Bottom-up** - this approach involves first generating forecasts for each series at the bottom-level, and then summing these to produce forecasts for all the series in the structure.
- **Top-down** - They involve first generating forecasts for the Total series yt, and then disaggregating these down the hierarchy.
- The **middle-out** approach combines bottom-up and top-down approaches. First, a "middle level" is chosen and forecasts are generated for all the series at this level. For the series above the middle level, coherent forecasts are generated using the bottom-up approach by aggregating the "middle-level" forecasts upwards.
- **Optimal forecast reconciliation** will occur if we can find the P matrix which minimizes the forecast error of the set of coherent forecasts. The objective is to find a matrix P that minimizes the error variances of the coherent forecasts. To use this in practice, we need to estimate forecast error variance of the h-step-ahead base forecasts. This can be difficult, and so there are four simplifying approximations which have been shown to work well in both simulations and in practice.



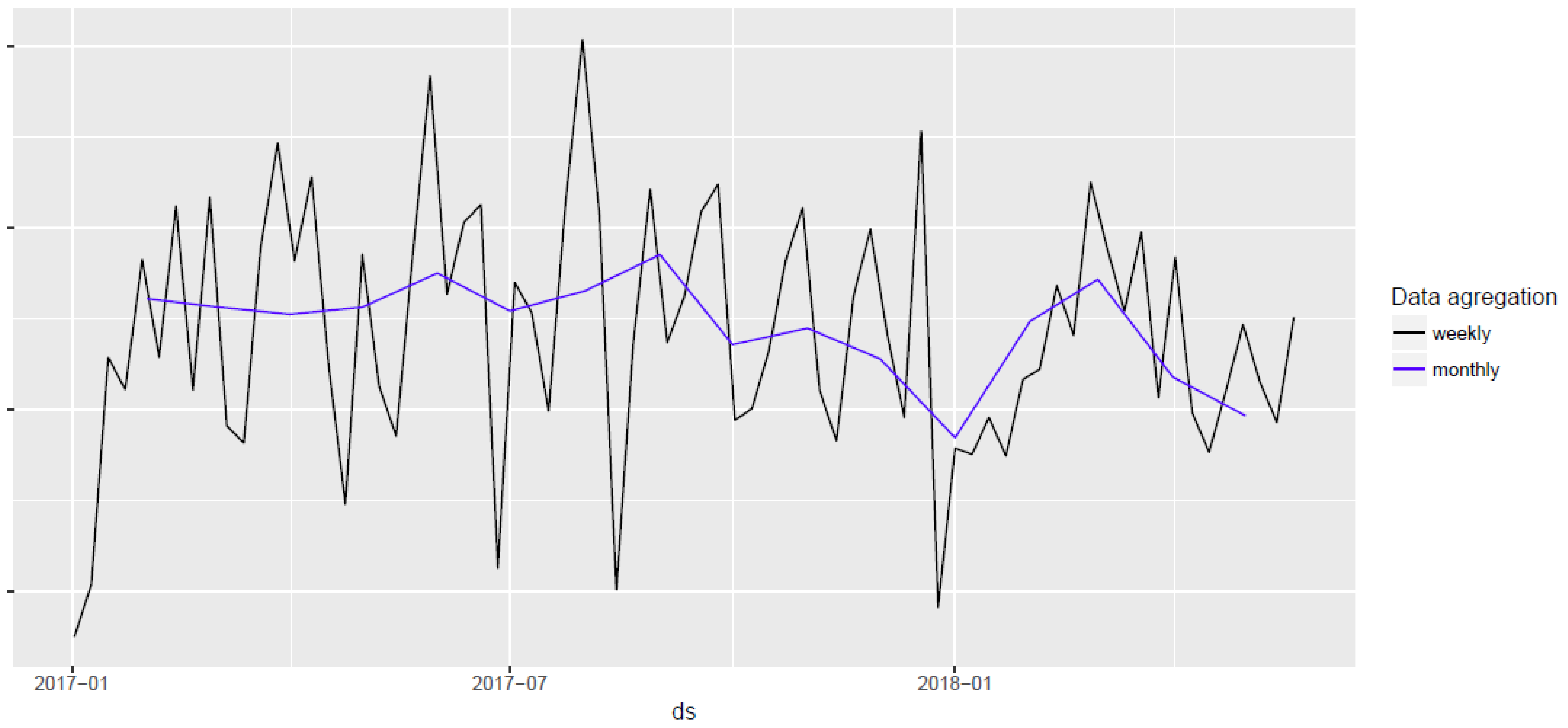
More info: "Forecasting: Principles and Practice" by George Athanasopoulos and Rob J. Hyndman

Why do we use different approaches?

National Weekly



National Monthly



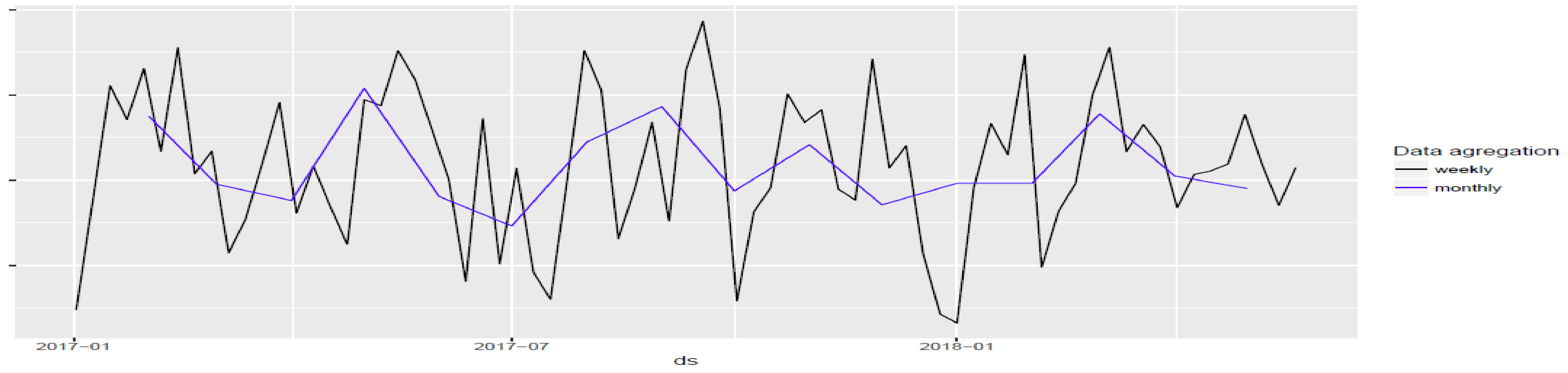
- National monthly approach was designed for time series that show more stability when aggregated to longer time period.

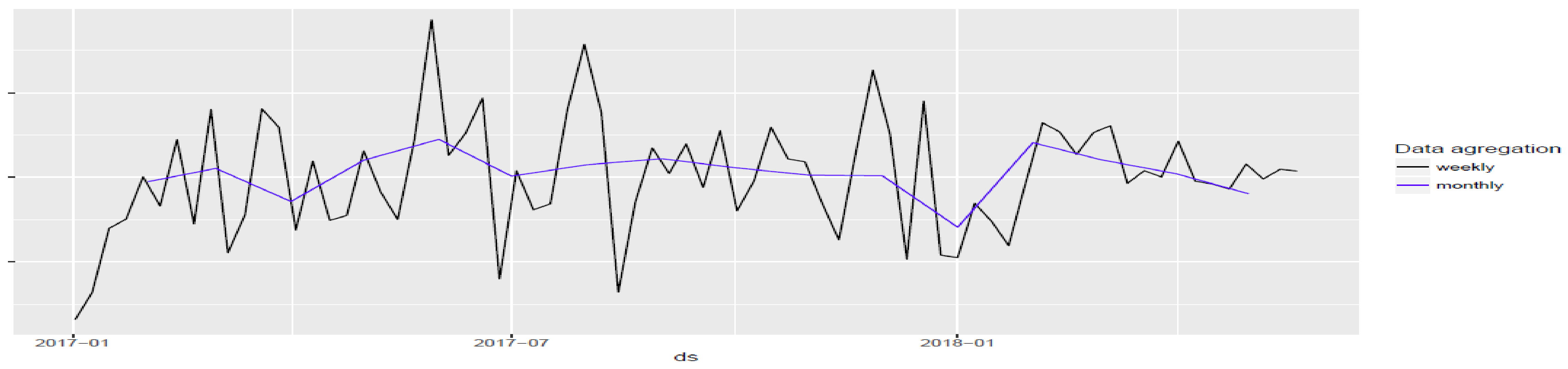
Comment to the example: although the variability on weekly level between two presented Base Codes is similar, when aggregated to months one of the Base Codes becomes more stable.

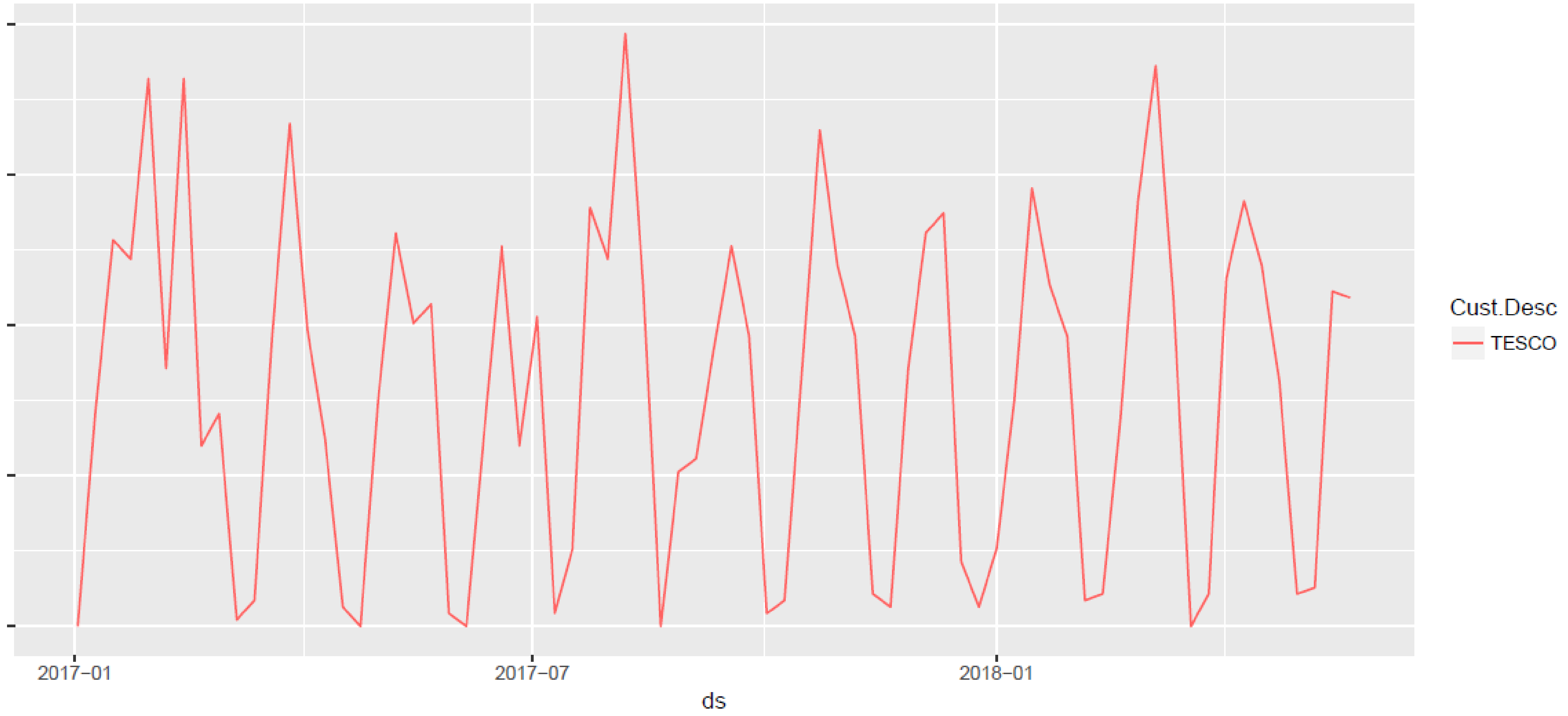
Why do we use different approaches?

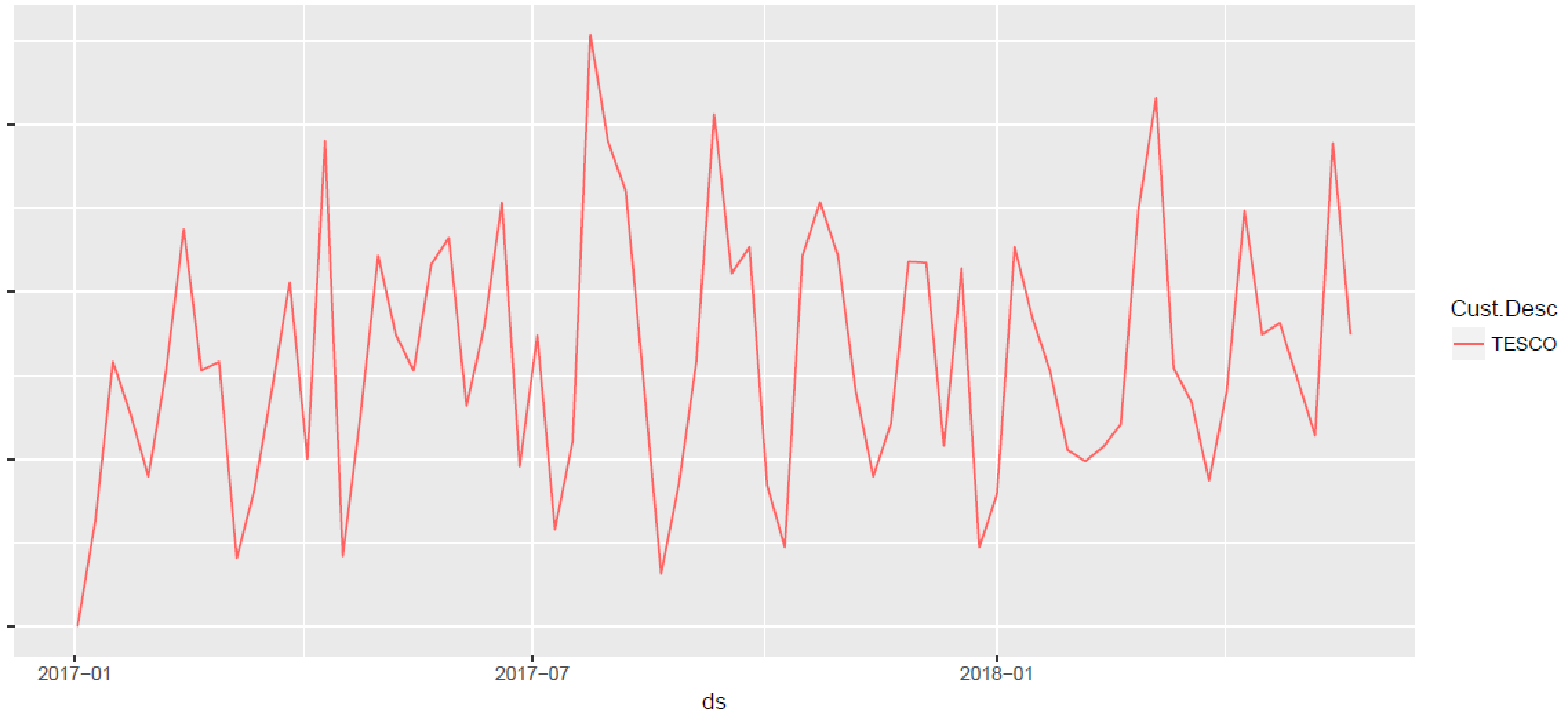
Customer Weekly

National Monthly









- Customer weekly approach was designed for situations when we can see a clear pattern of sales at a lower level (customer) which is not visible on a higher level (national).

Comment to the example: there is no visible pattern on national level for neither of those Base Code. However at a customer level pattern emerges for the one on the left.

Work flow

1. Cross validation

3 main approaches

Calculate forecasts for up to all combinations of models per BC and test each of them on weeks from testing months

National Monthly (NM)

National Weekly (NW)

Customer Weekly (CW)

2. Model selection

Select for each approach model/models with lowest error

1 best model per BC

1 best model per BC

1 best model per customer per BC (up to 4 models per BC)

3. Create derivative approaches

Part 1

Reconciliation (recon)

Reconcile best model from NW level with best models from on CW level

1 model per BC (obtained from best models on NW and CW approaches)

Part 2

Ensembling (ensemble)

Aggregate forecasts from NW, CW and recon approach to monthly level to average them with NM approach

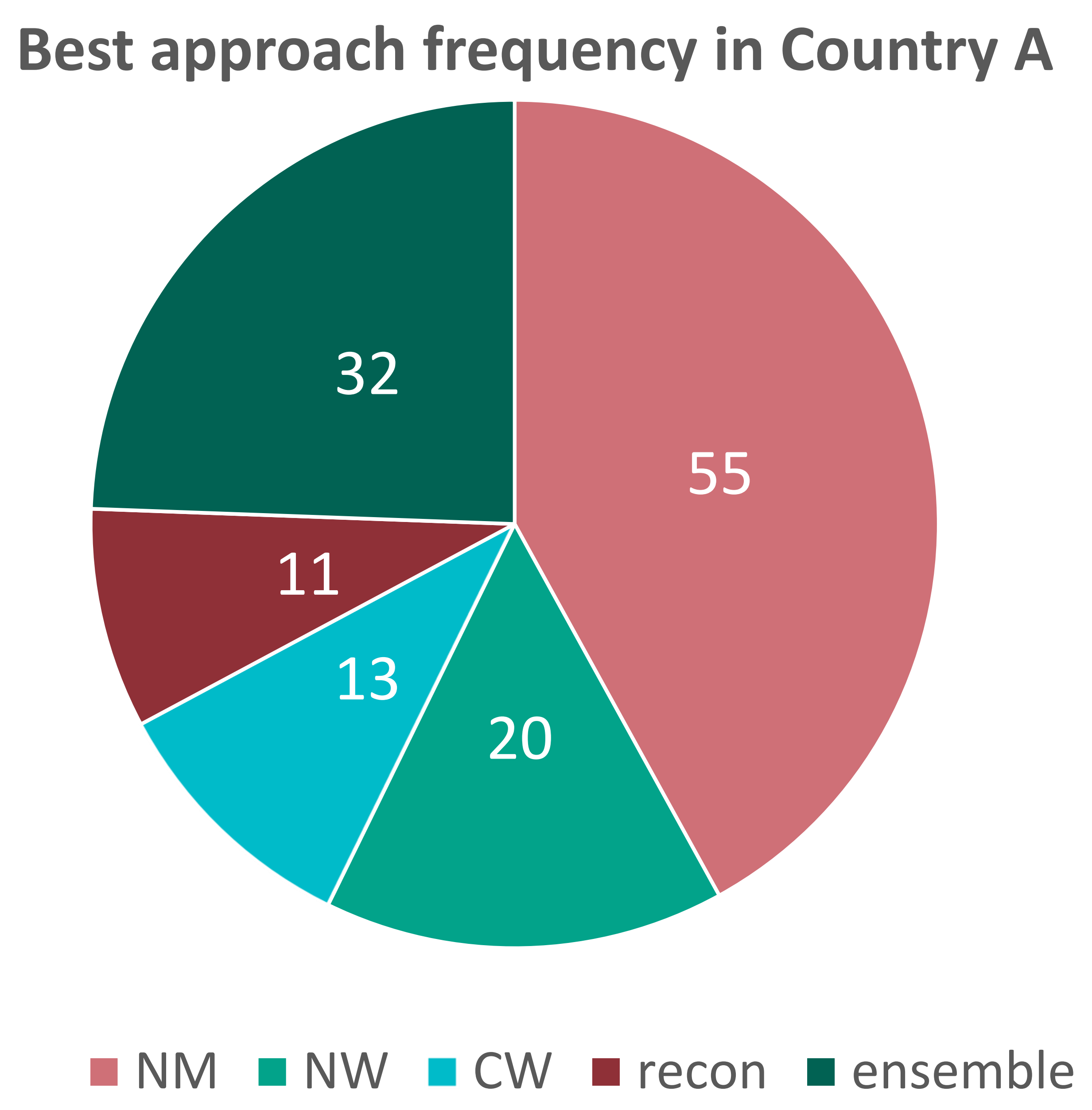
1 model per BC (obtained from models selected in NW, CW, NM and recon approaches)

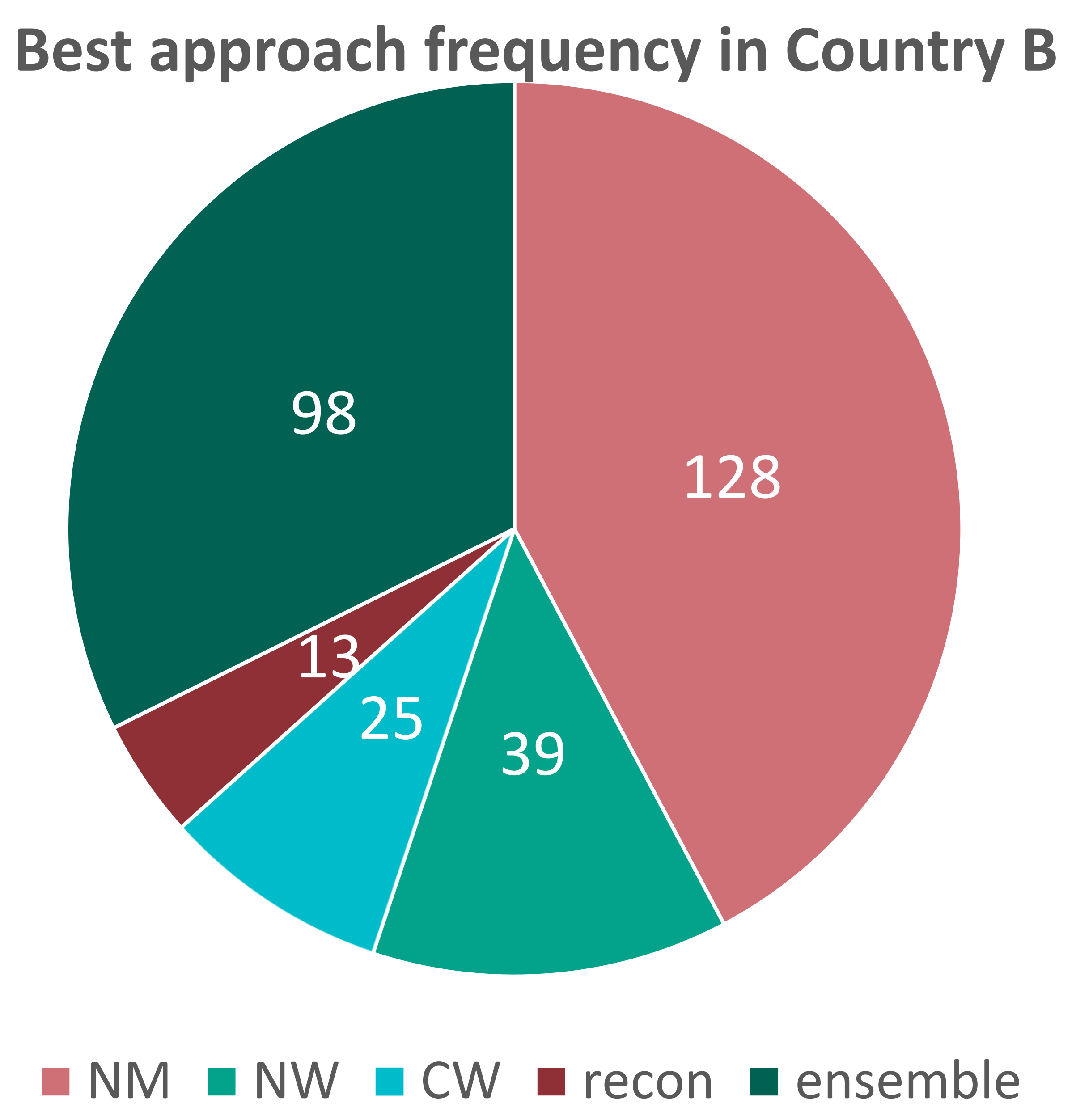
4. Select best approach

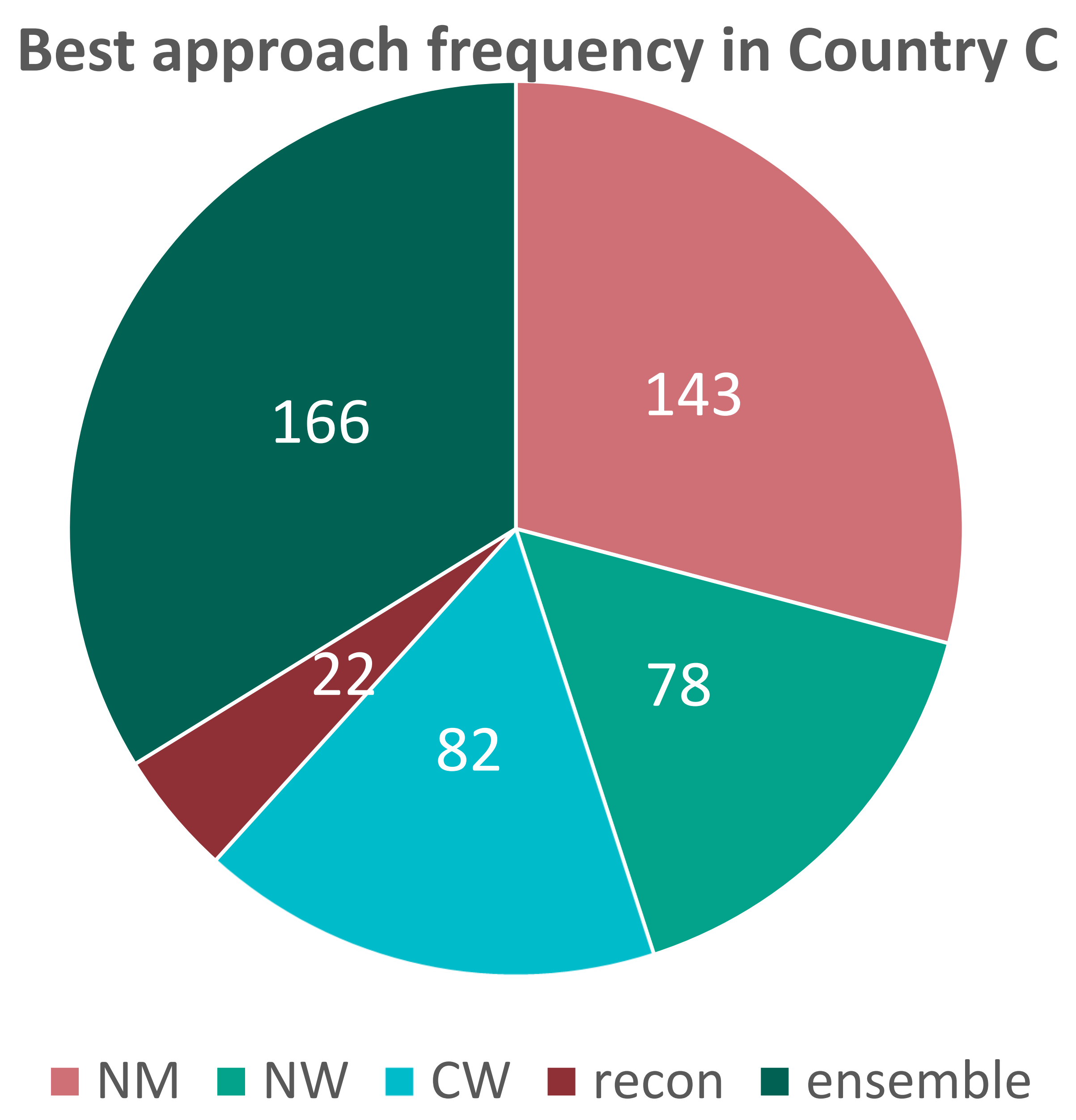
We have 5 approaches (NW, CW, NM, recon and ensemble) to choose from. Select for each BC one approach, which has the lowest error on average for all testing months.

1 model with the best result per BC

How often each approach was used?







Deploying on production

Going to production

Reference monthly workload (typical should not deviate by more than 50%):

> **1942 base codes** from Country A, Country B and Country C
> **3 approaches** based on granularity Customer Weekly,
> National Weekly and Monthly

This yields **5826 forecasting jobs**, each estimating on average 50 different models.

Processing would take around 45 hours, using typical single threaded R approach and one computer per country.

→ Productions solution requires different approach

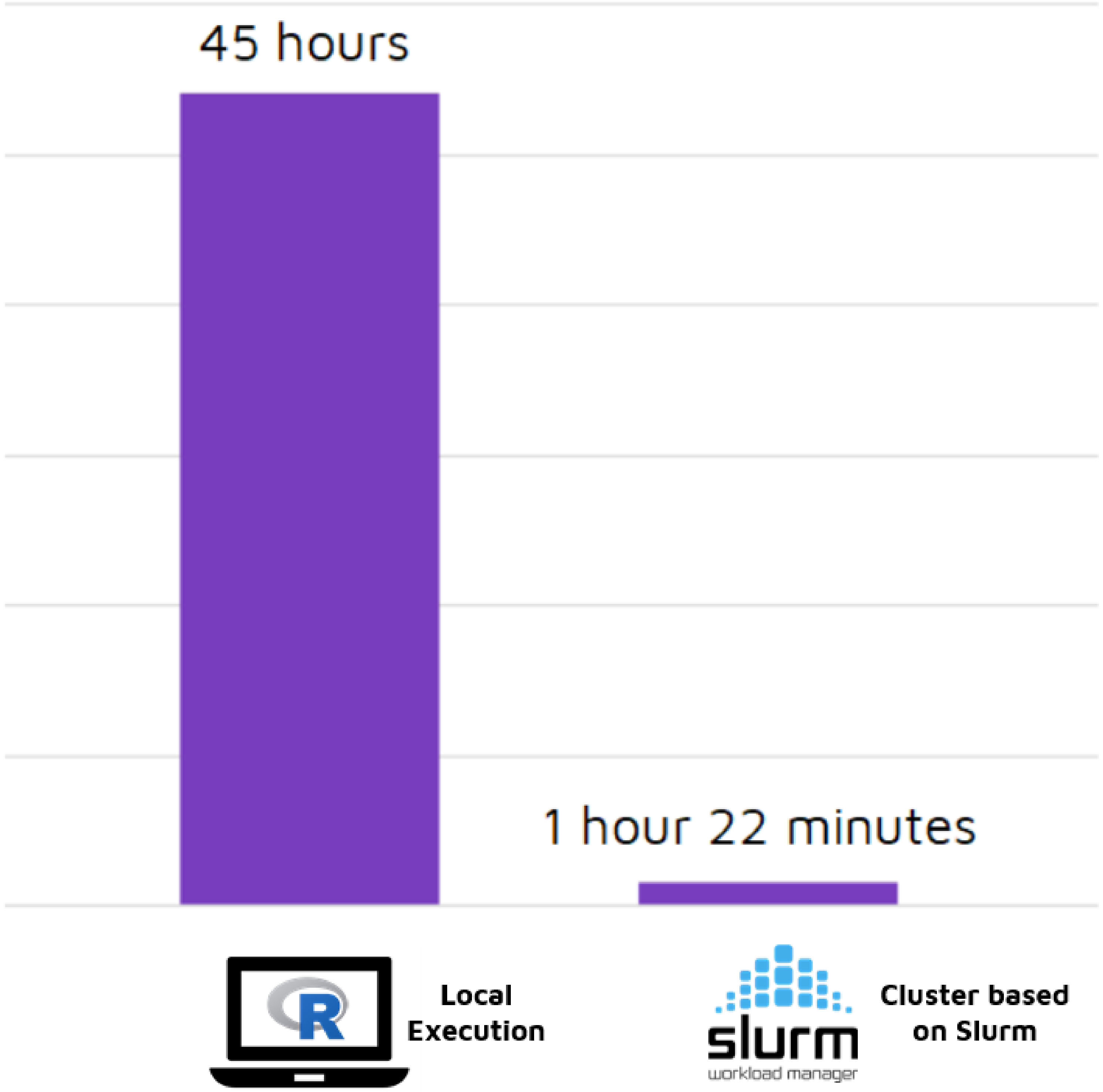
Going to production

Our new approach involved running those **5826 forecasting jobs** in parallel using ephemeral Slurm cluster on MS Azure platform consisting of:

128 compute nodes
2 CPU threads each

which translates to **256 jobs** running in parallel.

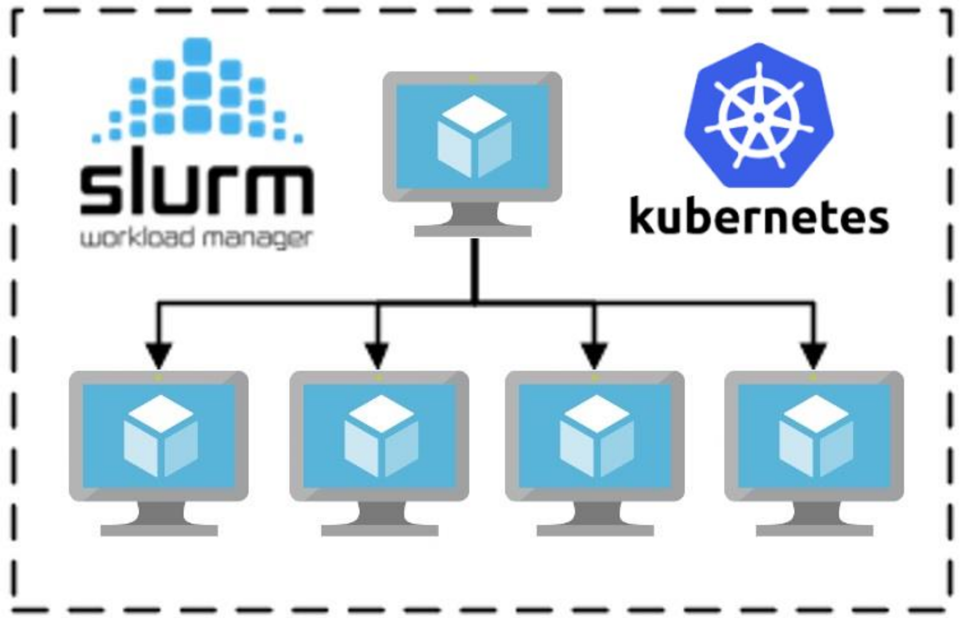
Considering additional 43 minutes overhead for cluster deployment and tear down this gives **1 hour 22 minutes total processing time.**



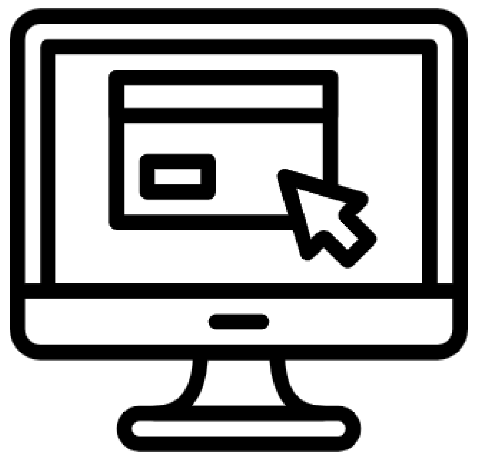
Going to production

We offer full scale production solution seamlessly introducing our exceling algorithms into existing business environment:

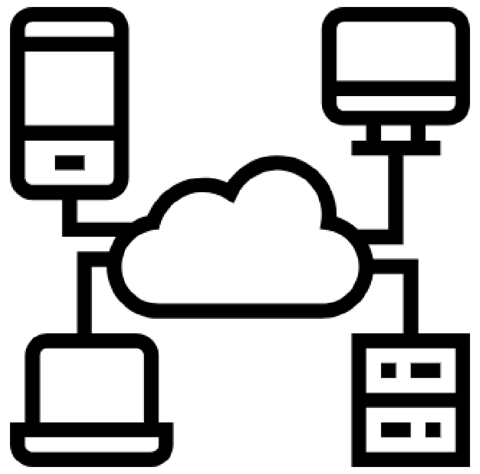
- Ephemeral cluster, being tore down when job is down enables great computing power with negligible infrastructure cost per monthly processing, scaling linearly with more data.



- User friendly front-end application for forecast performance monitoring and adjustment



- Fully automated connection to existing IT infrastructure, including SAP and interface for manual inputs



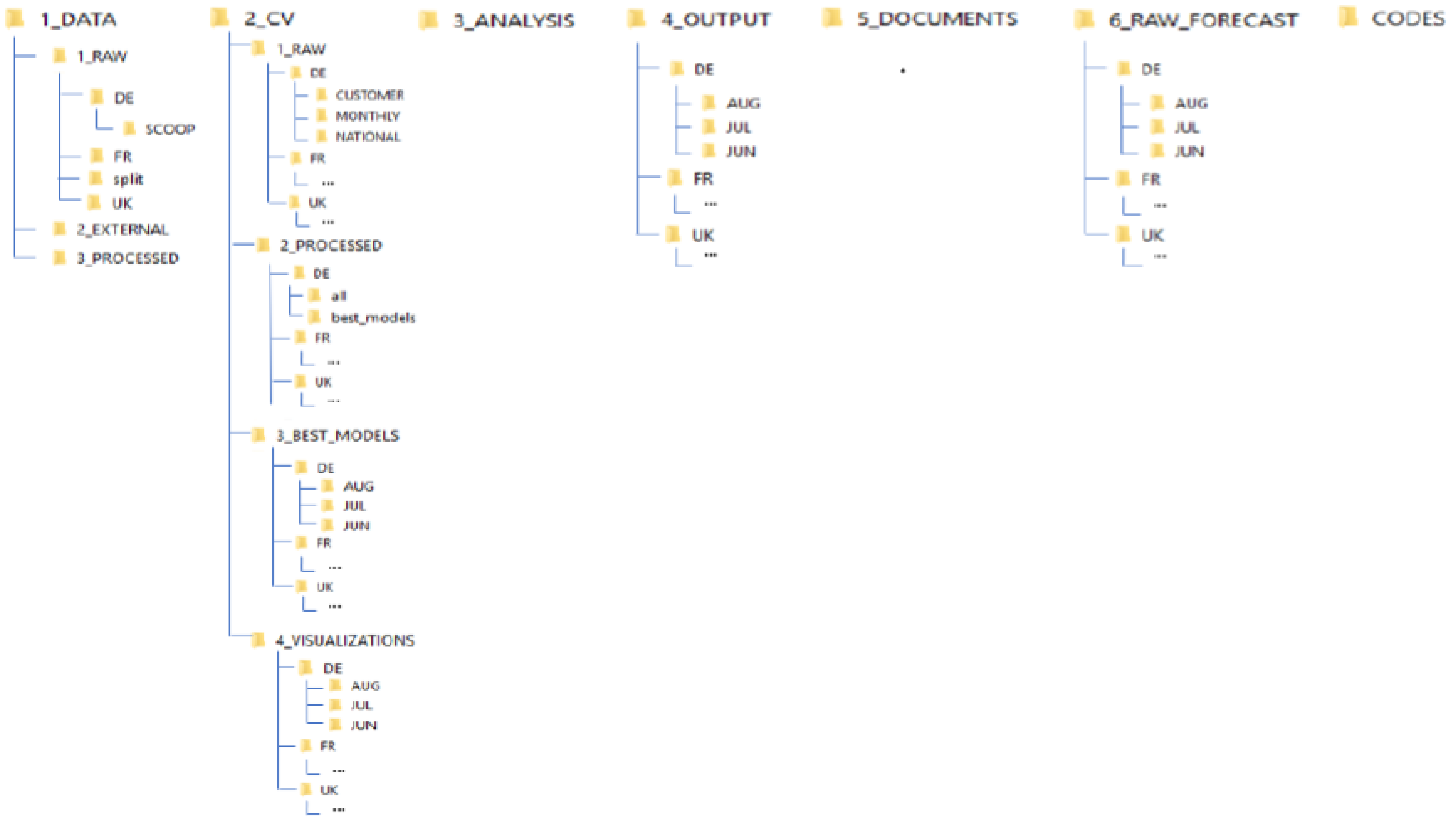
Summary

Lessons learned

Organizational perspective

1) Team project + lack of common environment – we need to **have the same structure of local files** to avoid wasting too much time on adjusting codes to local settings (we started it at the beginning of Wave 2)

1) Dealing with **results of CV**: handling lots of files with calculated errors per fold per BC (csv), using RDS for best models per BC to fasten loading data, rearranging codes due to RAM limitations of loading csv for searching best models



lingaro

Digitize. Disrupt. Lead.

Thank you for attention!

lingarogroup.com