Data Science League DATEV Challenge

-Solutions and Winners-

Monday, 21.02.2021
Presentation of solutions and award ceremony

Agenda

10:00	Welcome
10:05	Team presentations (10 min each, 2 min Q&A)
10:45	Data origin, DATEV approach
11:00	Announcing the Winners
11:10	Feedback session

Give us a quick feedback: How did you like the hackathon?

Put -- / - / o / + / ++ in the chat window

Team Presentation IAB

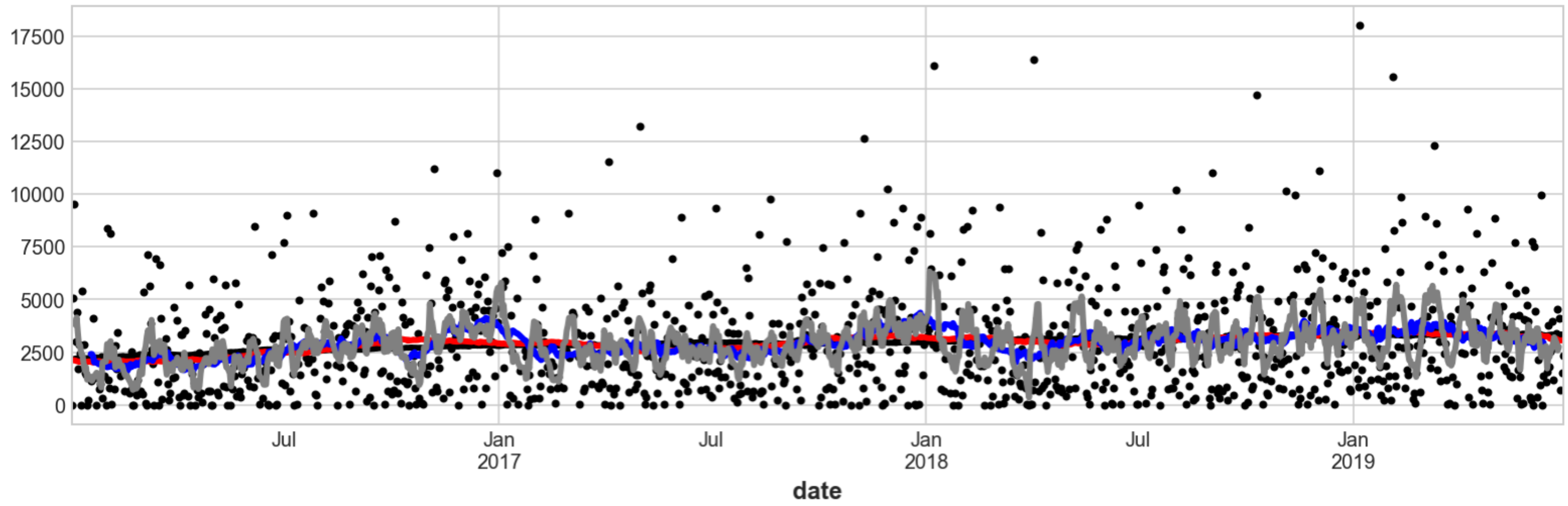
INSTITUT FÜR ARBEITSMARKT- UND BERUFSFORSCHUNG
Die Forschungseinrichtung der Bundesagentur für Arbeit

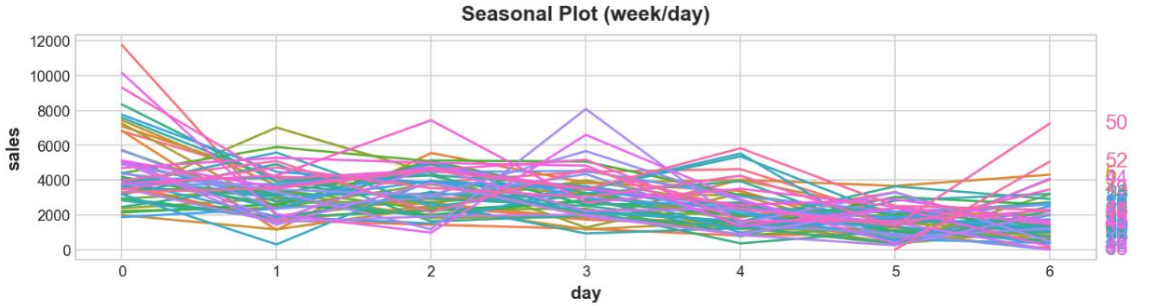
DATA SCIENCE LEAGUE CHALLENGE 3

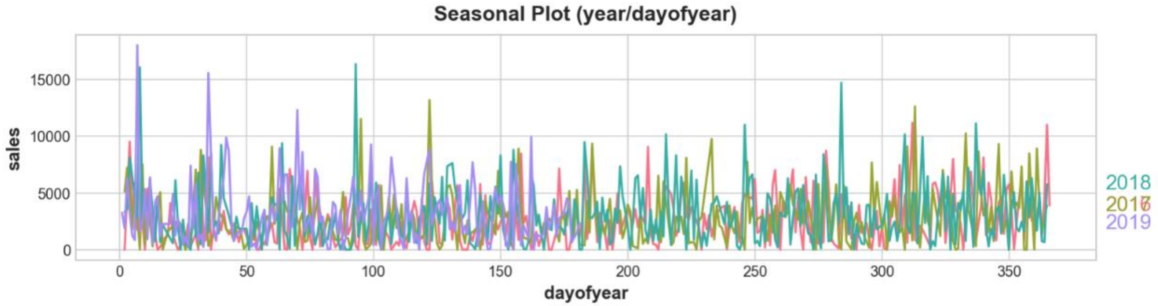
IAB

Ben Börschlein
Leonie Wicht
Lina Metzger
Michael Oberfichtner
Sophie Hensgen

DATA INSPECTION







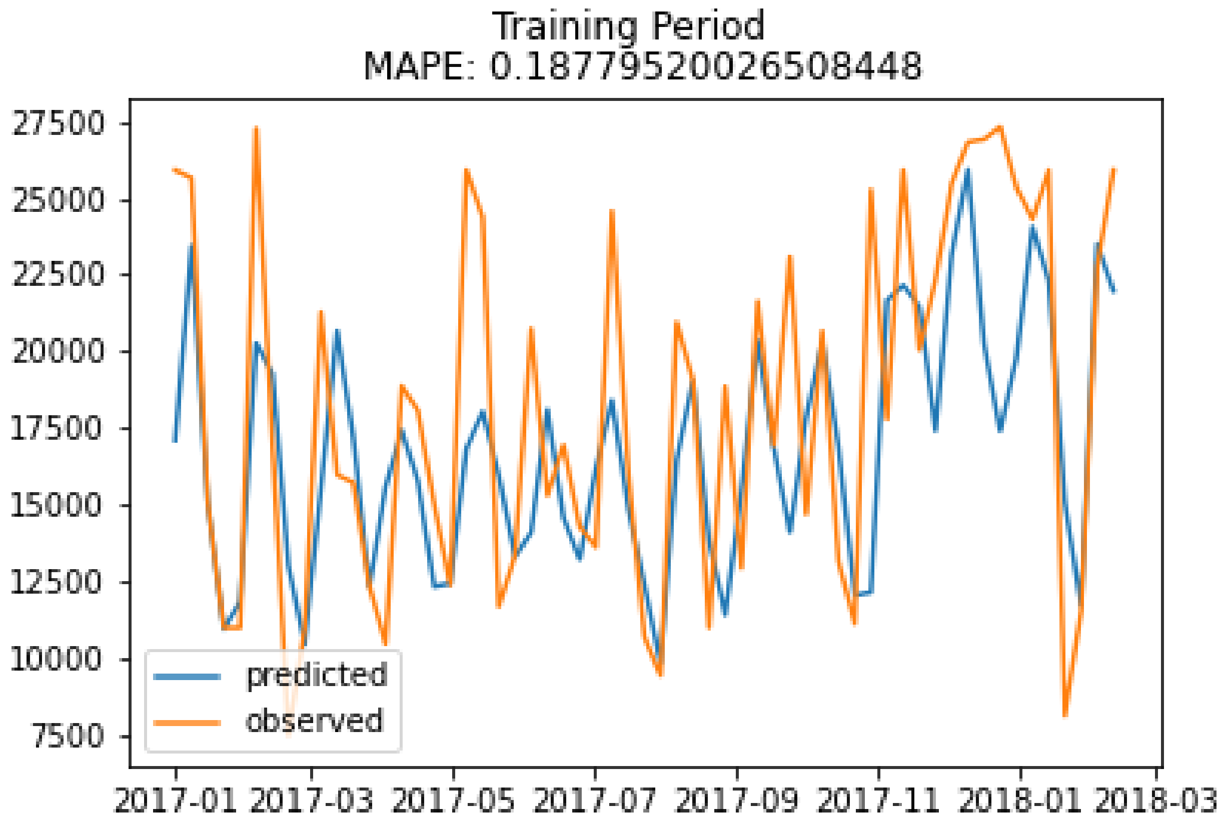
FACTS

- Multi-step forcast problem → cannot use short-run autoregressive features
- Feature engineering based on time and dates
  - date-related features (seasonality), period number (trend), sales last year (yearly similarities)
- Validation set of 1 year
- Training share of 75%
- Excluded 5 largest and 5 smallest values
- Model:
  - Deep Feed Forward Neural Network
  - 3 dense layers of 100 neurons each
  - Optimizer: Adam
  - Learning rate: 0.001

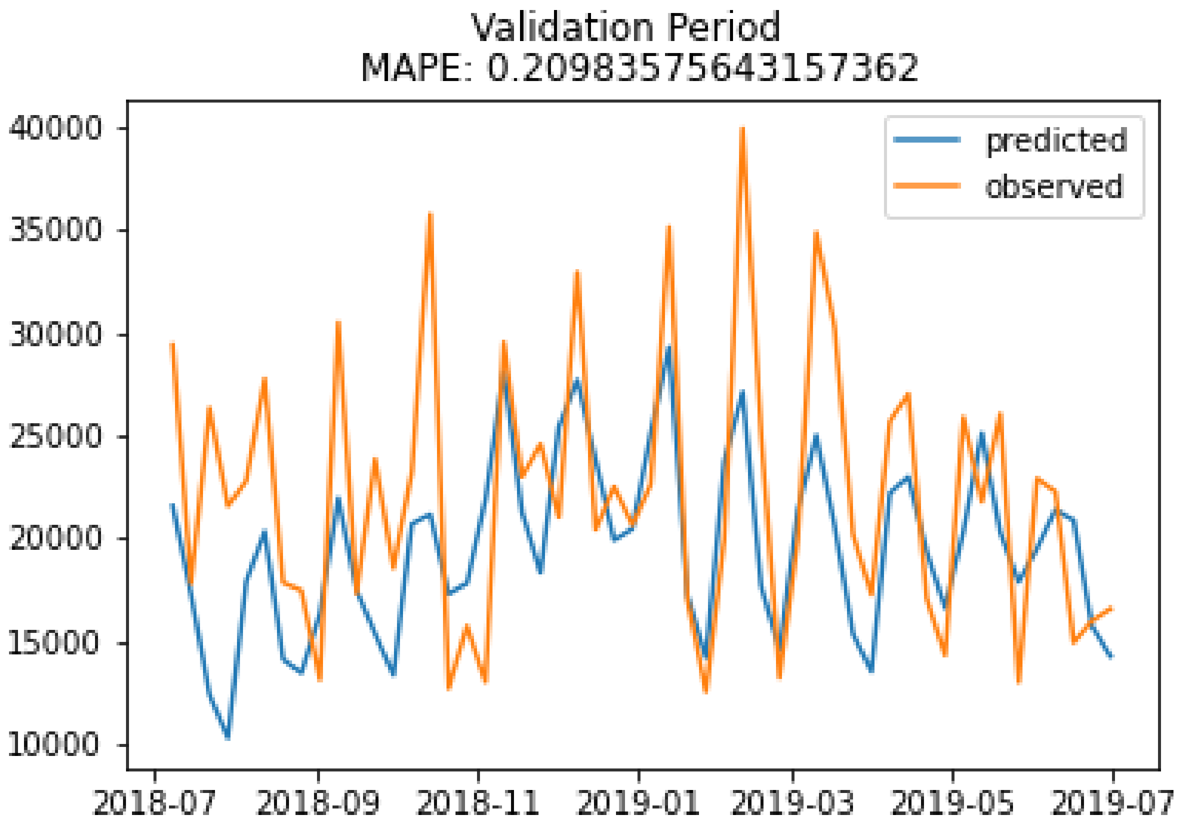
THE MODEL

```python
# set parameters
model_setup = Sequential(
    [
        Dense(100, activation="relu", ),
        Dense(100, activation="relu"),
        Dense(100, activation="relu"),
        Dense(1, activation="linear")
    ]
)

epochs = 500
loss = "mean_absolute_percentage_error"
learning_rate = 0.001  # --> smaller makes predictions more stable

# define early stopping rule
stop_early = EarlyStopping(monitor='val_loss', mode='min', verbose=1, patience=10)

def train_model(model_setup, X_train, y_train, X_test, y_test, epochs):
    model_setup.compile(optimizer=Adam(learning_rate=learning_rate), loss=loss)

    history = model_setup.fit(X_train, y_train, epochs=epochs, validation_data=(X_test, y_test), callbacks=stop_early)
    return model_setup, history


model, hist = train_model(model_setup, X_train, y_train, X_test, y_test, epochs=epochs)
```

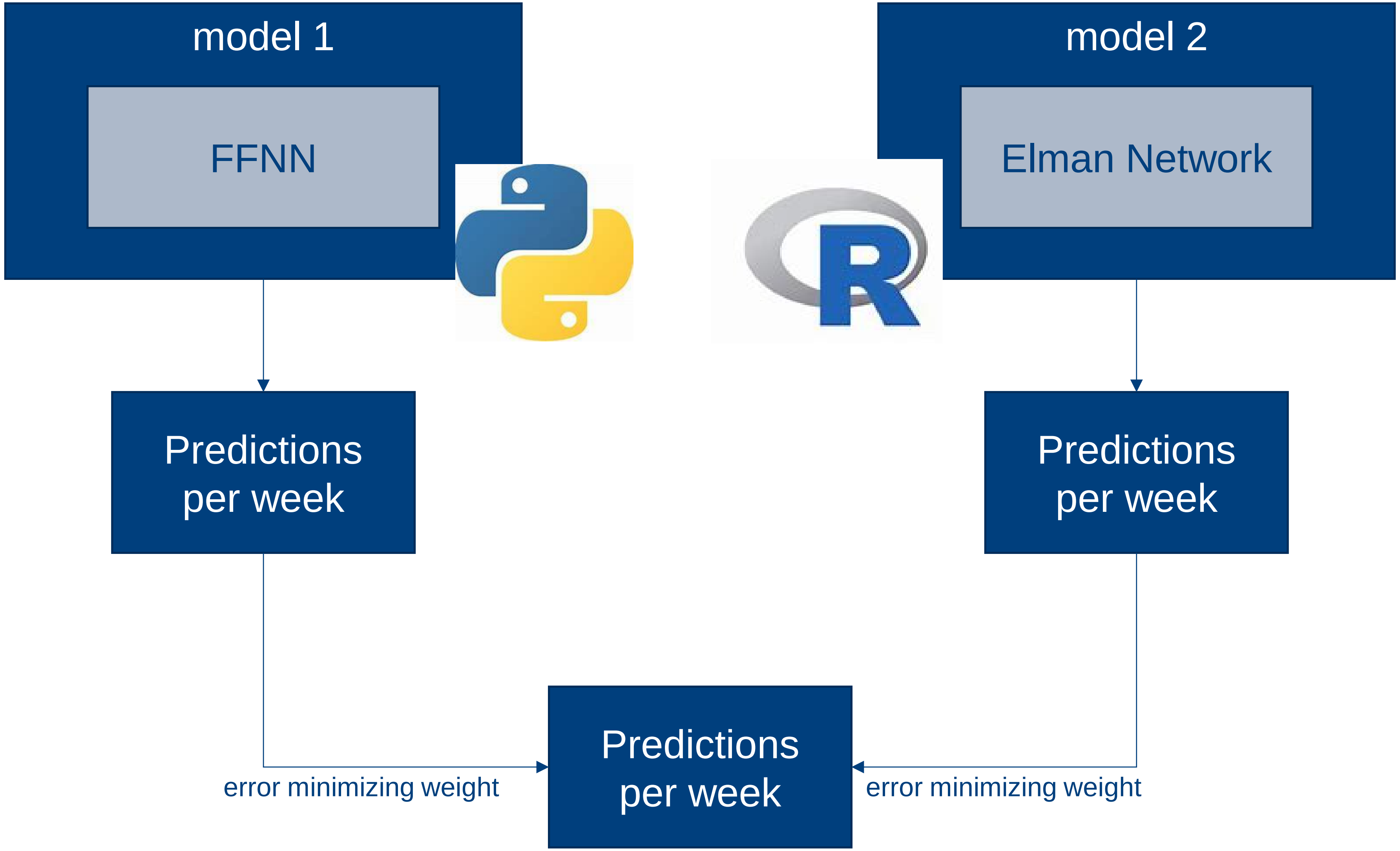
PREDICTED AND OBSERVED VALUES
TRAINING PERIOD



PREDICTED AND OBSERVED VALUES VALIDATION PERIOD



PLANNED: ENSEMBLE-DESIGN



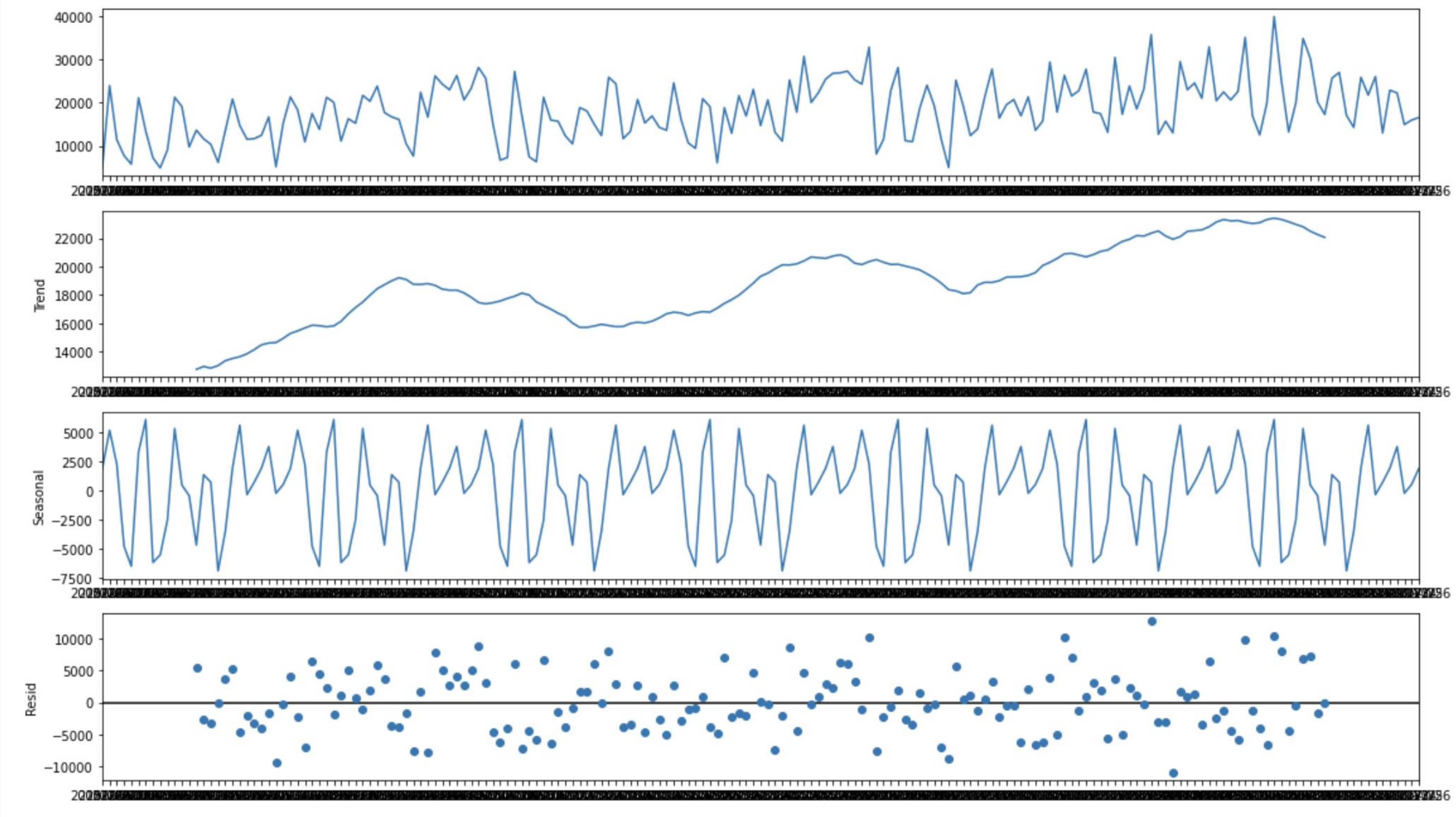
Team Presentation GfK

See DSL_Datev-Challenge_GfK_submission.pdf

Team Presentation adidas

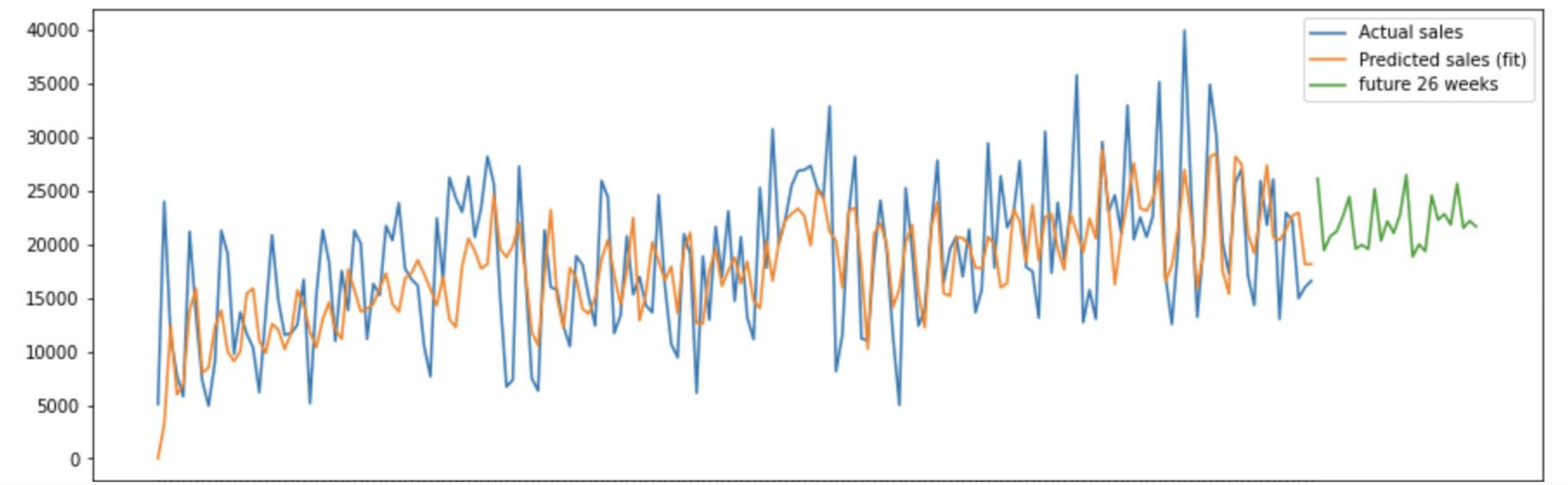

Data Science League
adidas

Visualization – Additive decomposition



SARIMA

S	Seasonal (monthly, quarterly, yearly etc.)
A R	Autoregressive (based on past values)
I	Integrated (accounts for trend trend)
M A	Moving average (takes periodic average to smooth out noise)



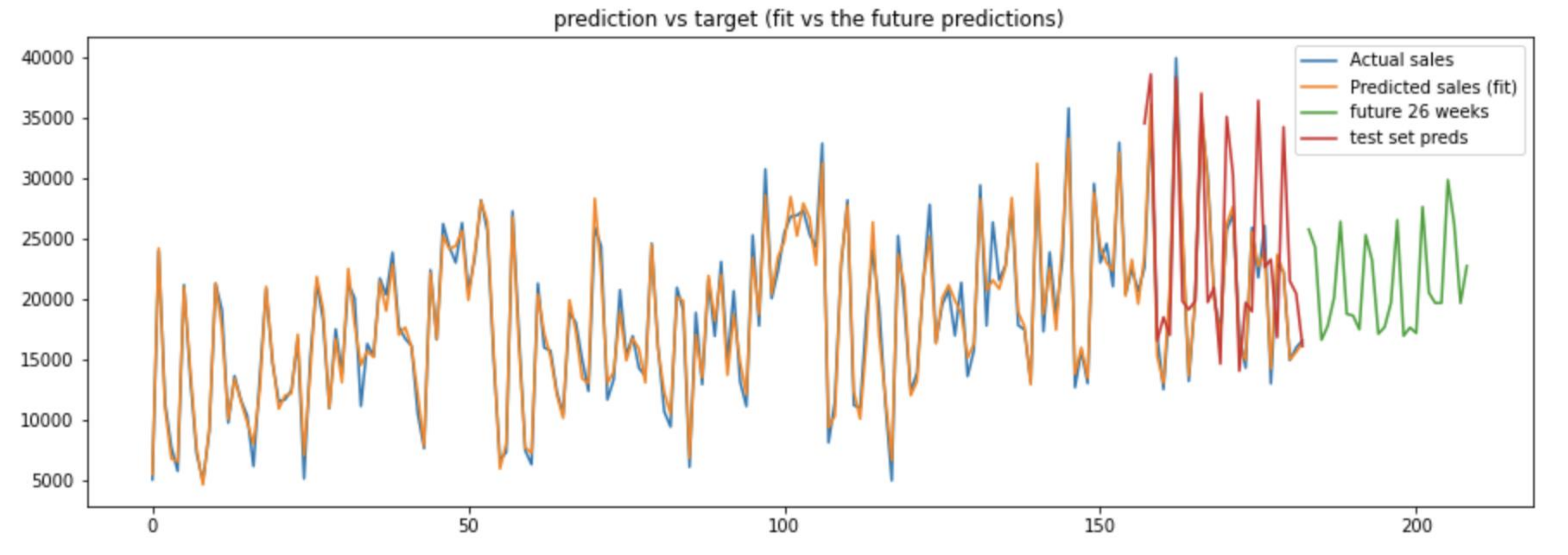
- Used weekly sales values as univariate time series
- Identified parameters for SARIMA model using pmdarima library
- Best model found at 52 weeks seasonality
- Used last 26 weeks of training data for evaluation
- Forecasted next 26 steps for final submission using all available data to train

XGBoost

date	weekofyear	zipcode	city	customer	product	ship mode	sales
2016-01-02	2015-W53	86150	Augsburg	KND-0160	PRK-0055	Standard Shipping	8
2016-01-03	2015-W53	47051	Duisburg, Stadt	KND-0250	PRK-0010	Premium Shipping	2669
2016-01-03	2015-W53	47798	Krefeld, Stadt	KND-0097	PRK-0058	Standard Shipping	1
2016-01-03	2015-W53	41061	Mönchengladbach, Stadt	KND-0177	PRK-0118	Standard Shipping	3
2016-01-03	2015-W53	30159	Hannover, Landeshauptstadt	KND-0190	PRK-0411	Standard Shipping	87
2016-01-03	2015-W53	52062	Aachen, Stadt	KND-0248	PRK-0112	Express Shipping	1117
2016-01-03	2015-W53	47051	Duisburg, Stadt	KND-0264	PRK-0073	Standard Shipping	7

- Used xgboost regressor for modeling the time series
- Feature engineering:
  - Added date-time related features
  - Only kept **ship mode** information and discarded other features
- Trained model on daily sales
- During prediction for future 26 sales added all combinations of the **ship mode**

	weekofyear	ship mode	month	year	dayofweek	dayofmonth	sales
0	2015-W53	Express Shipping	1	2016	6	3	1126
1	2015-W53	Premium Shipping	1	2016	6	3	2669
2	2015-W53	Standard Shipping	1	2016	5	2	8
3	2015-W53	Standard Shipping	1	2016	6	3	1269
4	2016-W01	Express Shipping	1	2016	2	6	4



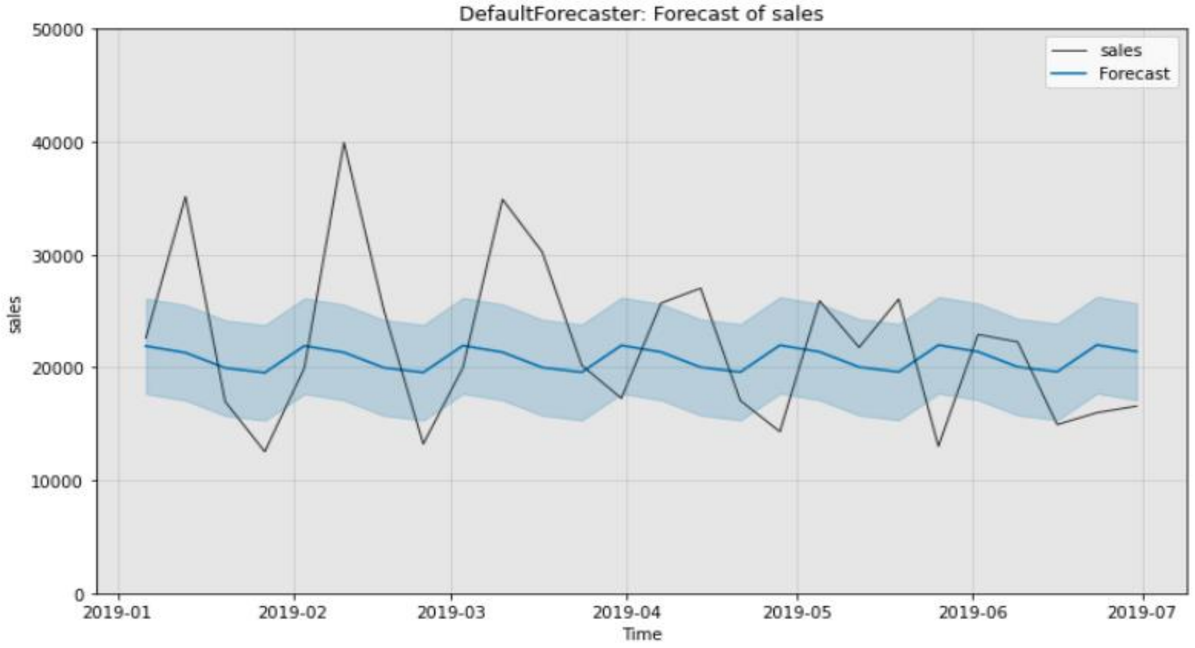
MERLION - FORECASTING FRAMEWORK FROM SALESFORCE

DATA EXPLORATION

- After an exploration of data we decided to use only **sales** and **date** features since there were no substantial pattern in other features.
- On a weekly level the sales has been shown some **trend** and there were clearly a **seasonality** pattern.

MODEL: LGBM

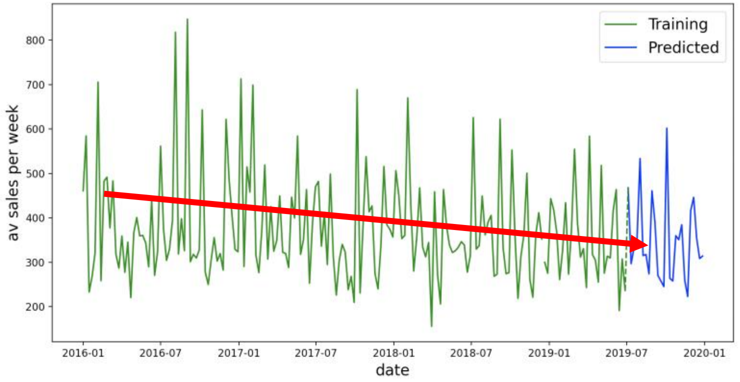
- It has been used a predictive model for forecasting the sales values
- The model is **LGBM** from **Merlion**, a Python library for time series
- The mean absolute percentage error: **0.26**



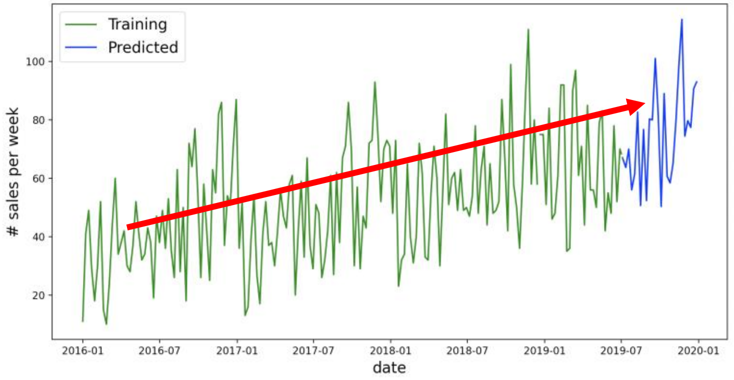
EXPONENTIAL SMOOTHING

- Split weekly sales as average x number of sales
- Both series exhibit opposite trends and somewhat indep. fluctuations (corr=0.38)
- Fit independent models to both series and combine their predictions
- Model: `statsmodels.tsa.exponential_smoothing.ets.ETSModel`

Av. sales per week ($)

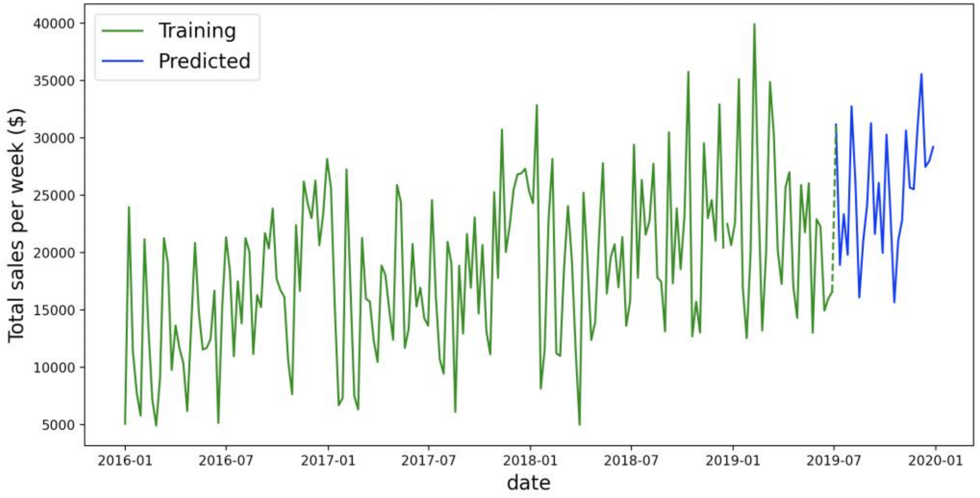


Number sales per week



x

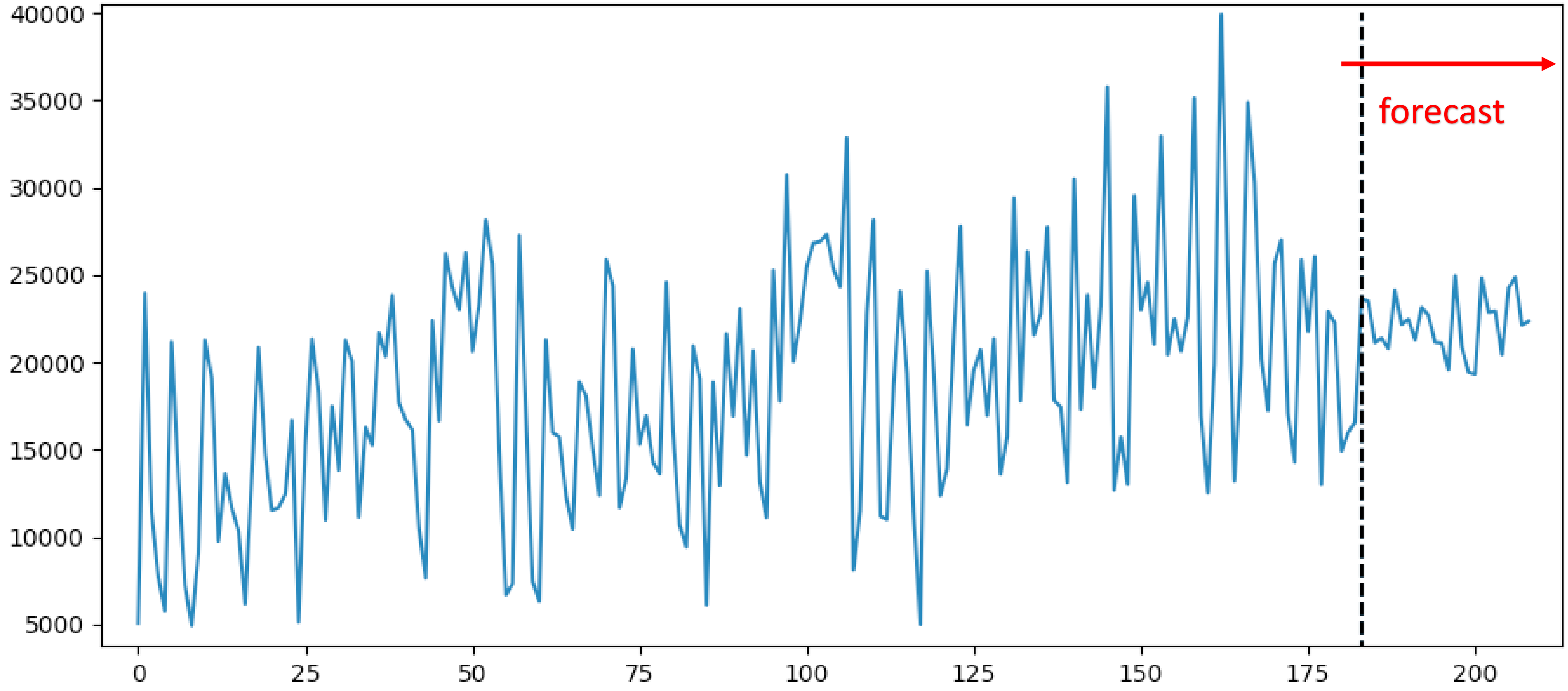
Total sales per week ($)



adidas

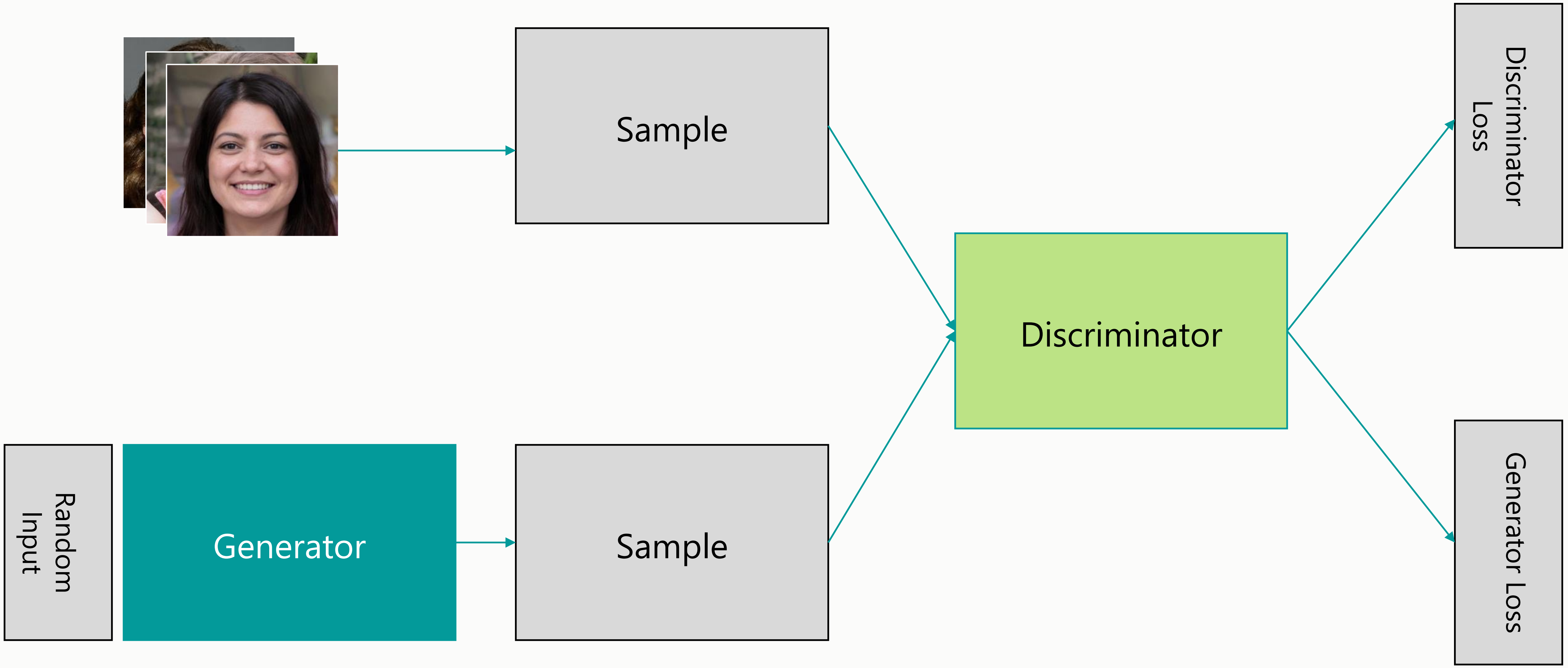
FINAL SUBMISSION

- Our MAPEs were very similar & our models were diverse
- Ensemble model: **voting regressor** made from **averaging** our different solutions
- Might lose accuracy in capturing the fluctuations but might improve the overall trend



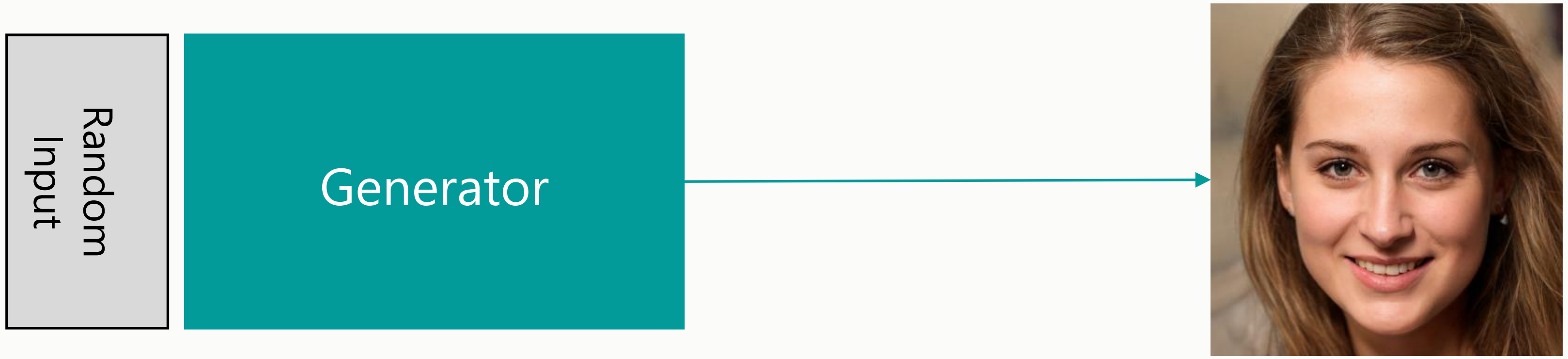
Data Origin

Generative Adversarial Networks (GAN)

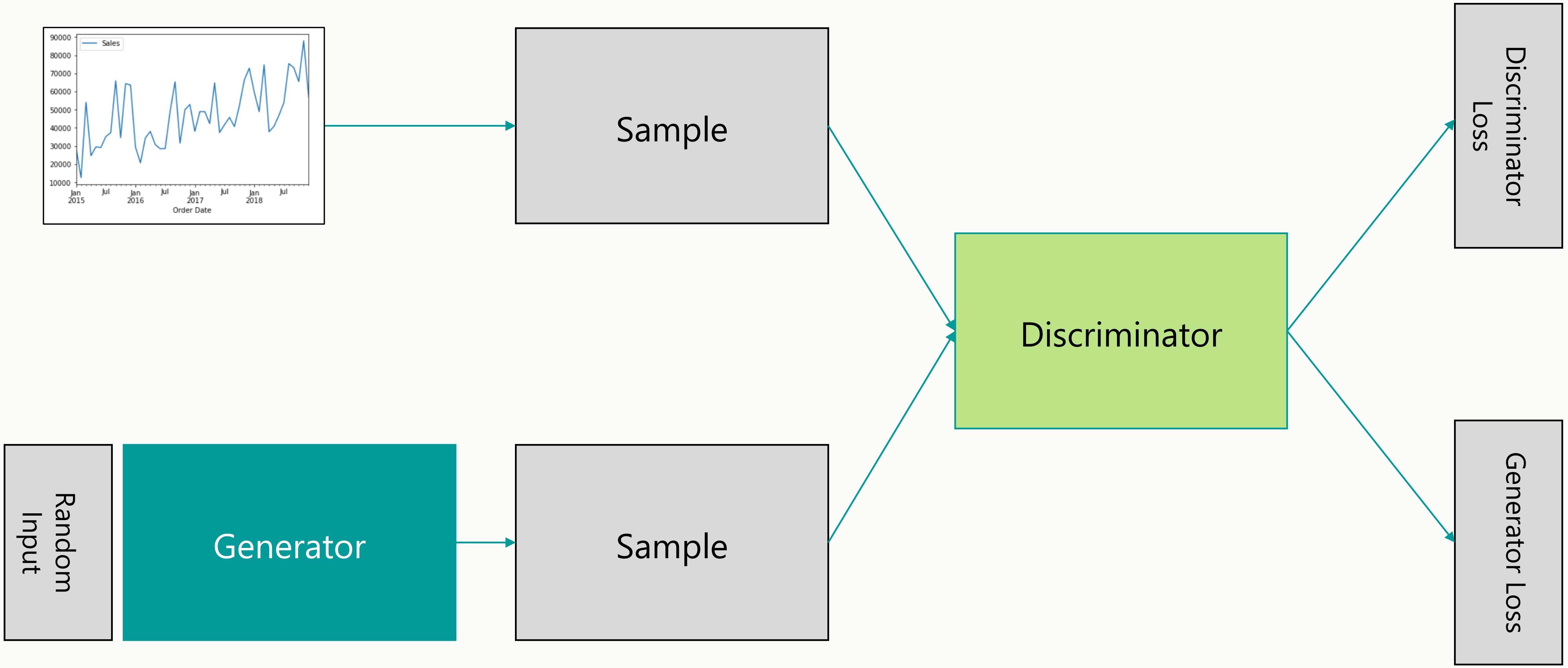


Generative Adversarial Networks (GAN)

Example: this-person-does-not-exist.com

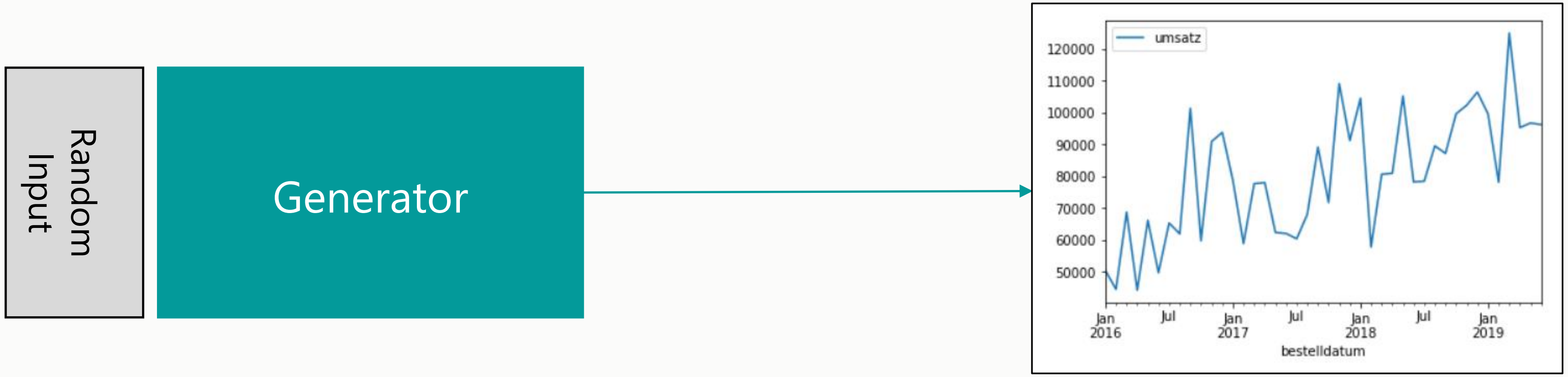


Generative Adversarial Networks (GAN)



Generative Adversarial Networks (GAN)

Creation of the (base-)dataset



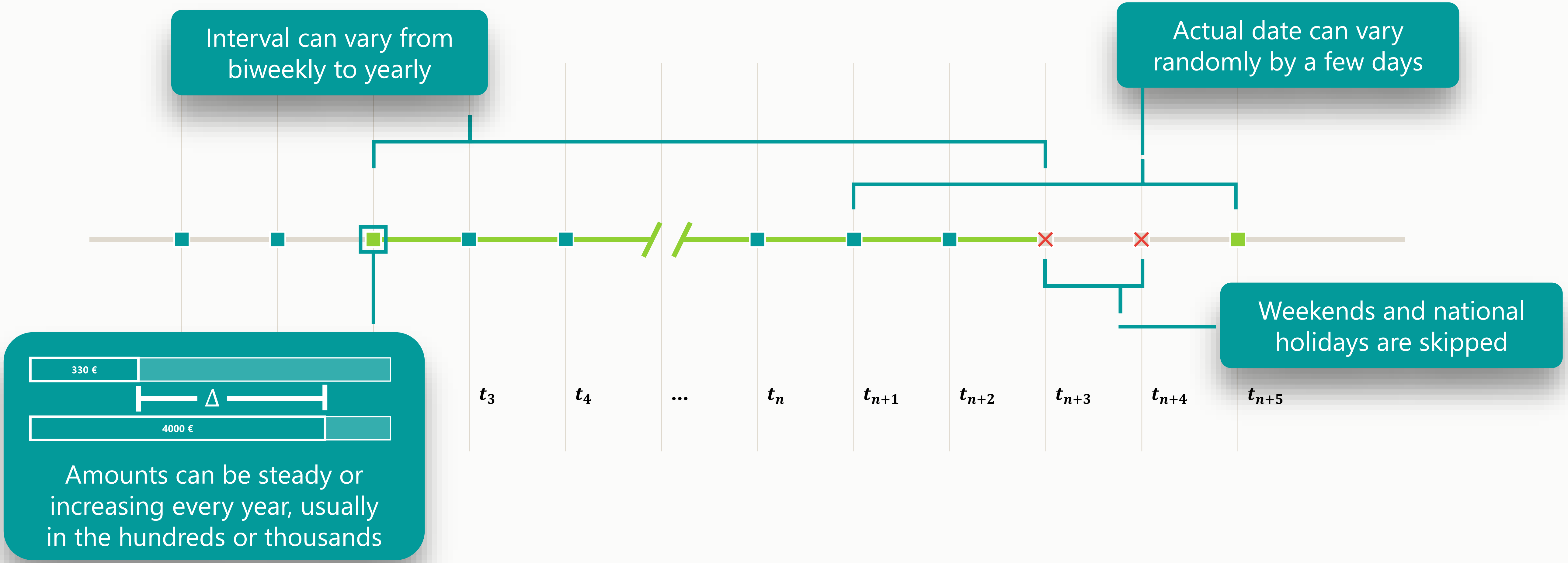
Christmas Spending

Christmas spending was added by scaling the december transactions by a Beta-distributed parameter.

December Transactions → **Scaling by $\kappa \sim \mathrm{B}(\alpha, \beta)$** → **Christmas Spending**

Reoccurring Postings

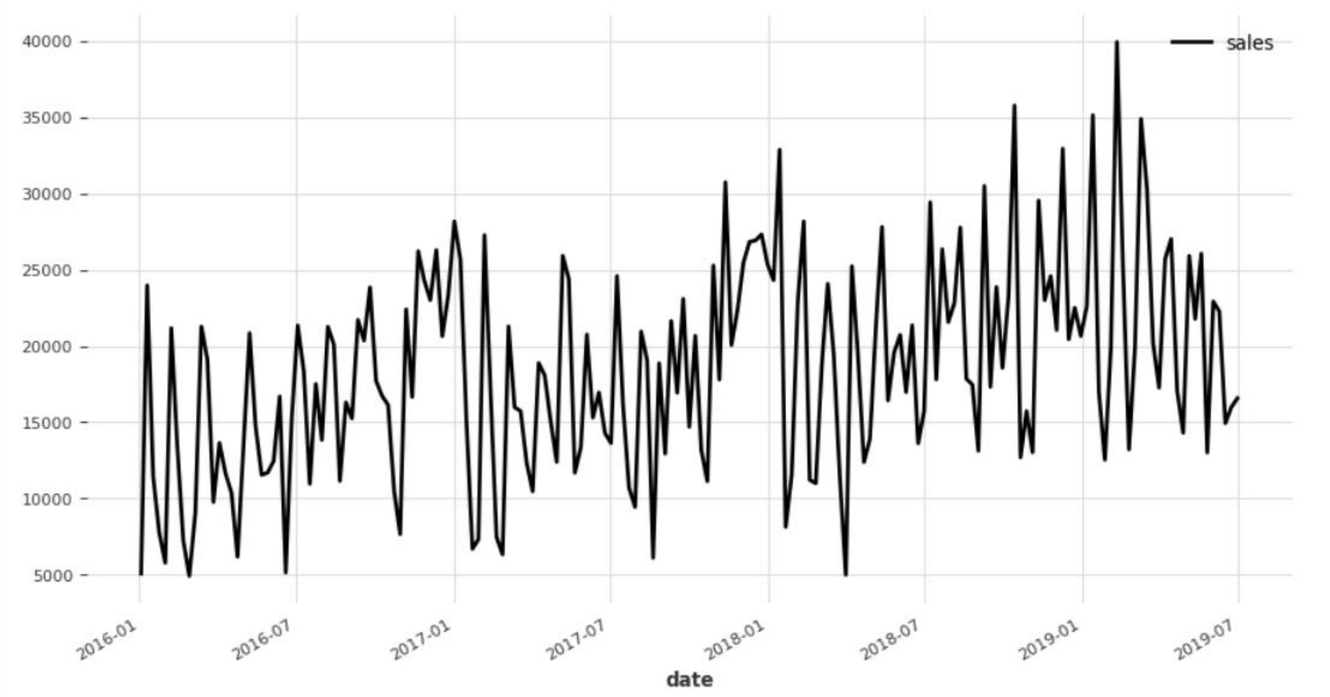
Typical data sets at DATEV contain a variety of reoccurring high-volume postings. This was simulated using some sensible boundaries.



DATEV Approach to DSL

Looking at the Data

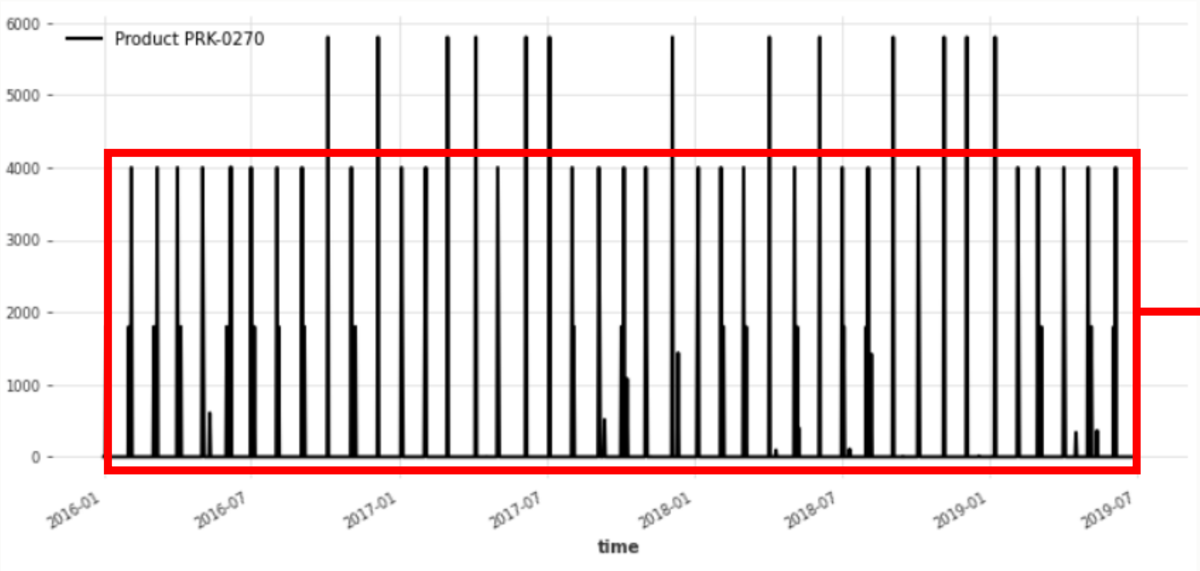
	date	weekofyear	ship mode	sales	customer	product	city	zipcode
0	2016-01-02	2015-W53	Standard Shipping	8	KND-0160	PRK-0055	Mainz, Stadt	55116
1	2016-01-03	2015-W53	Premium Shipping	2669	KND-0250	PRK-0010	Kassel, documenta-Stadt	34117
2	2016-01-03	2015-W53	Standard Shipping	1	KND-0097	PRK-0058	Hamburg, Freie und Hansestadt	20038
3	2016-01-03	2015-W53	Standard Shipping	3	KND-0177	PRK-0118	Rostock, Hansestadt	18055
4	2016-01-03	2015-W53	Standard Shipping	87	KND-0190	PRK-0411	Magdeburg, Landeshauptstadt	39104



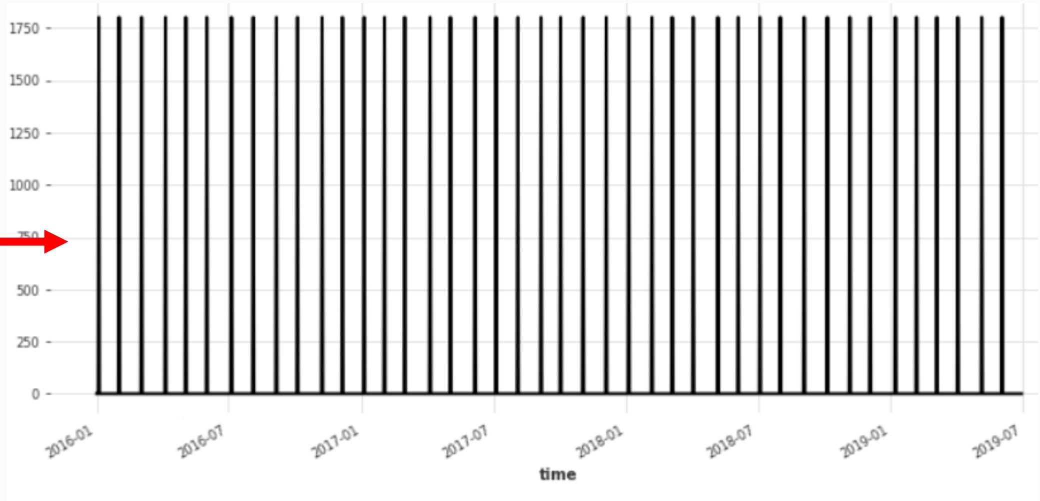
Examining the Data

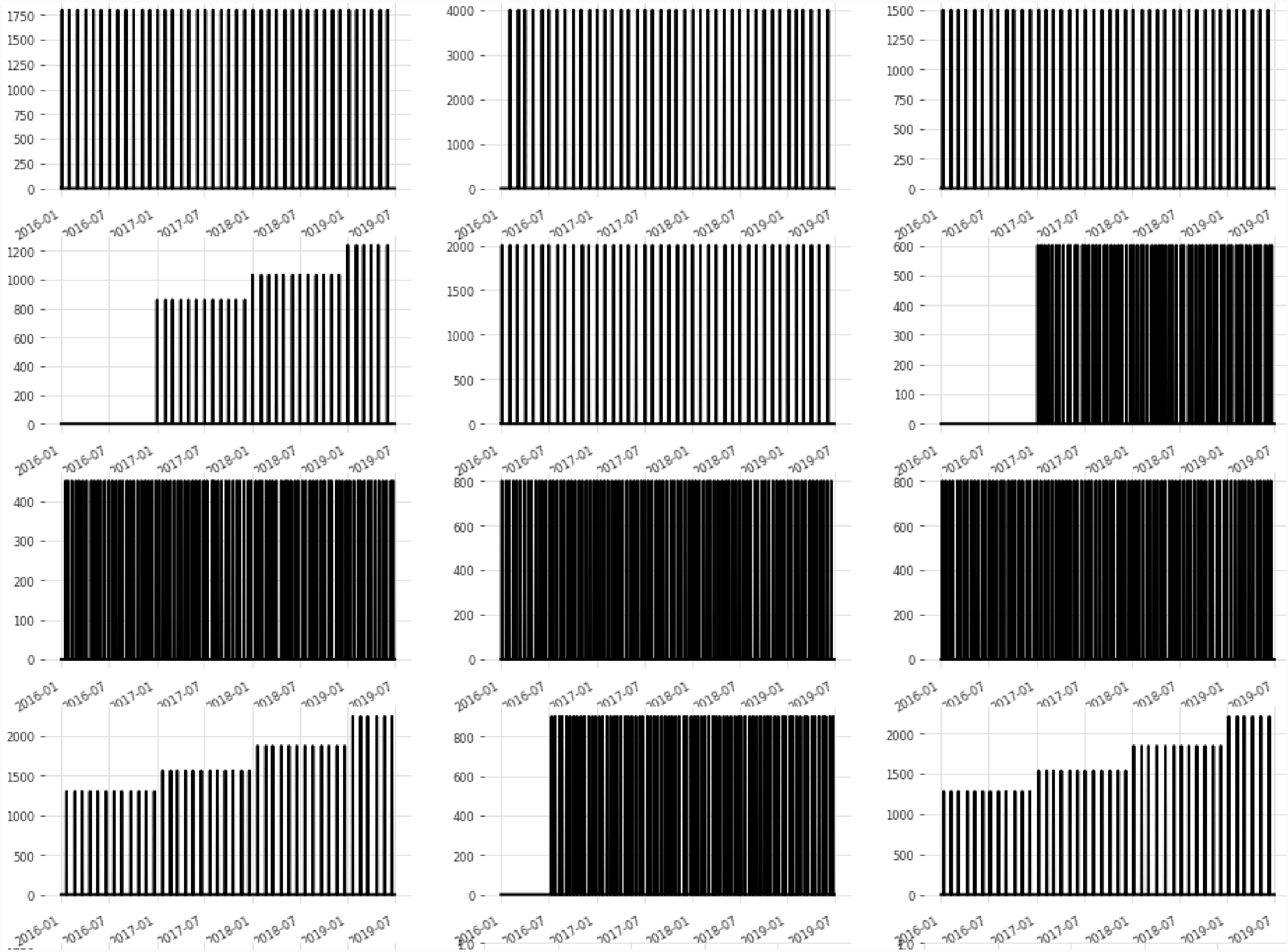
Selecting specific products or product-customer combinations:

PRK-0270



PRK-0270 + KND-0191





Modeling Regular Transactions

Model Time Intervals:

$\Delta t \sim \mathrm{Exp}(\lambda)$

Squeeze in regular buckets:

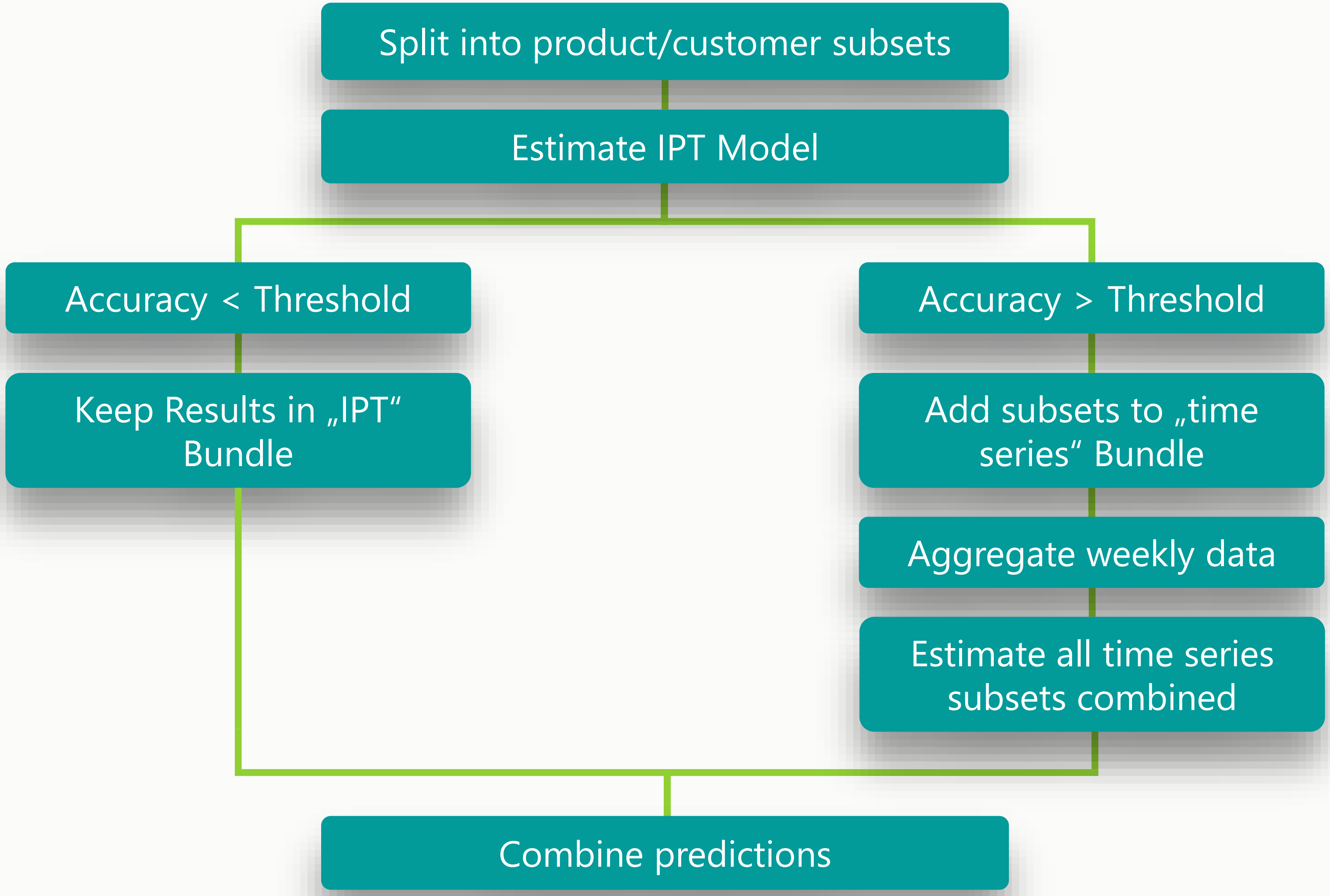
30.3 days => Monthly

Adjust for increasing amounts

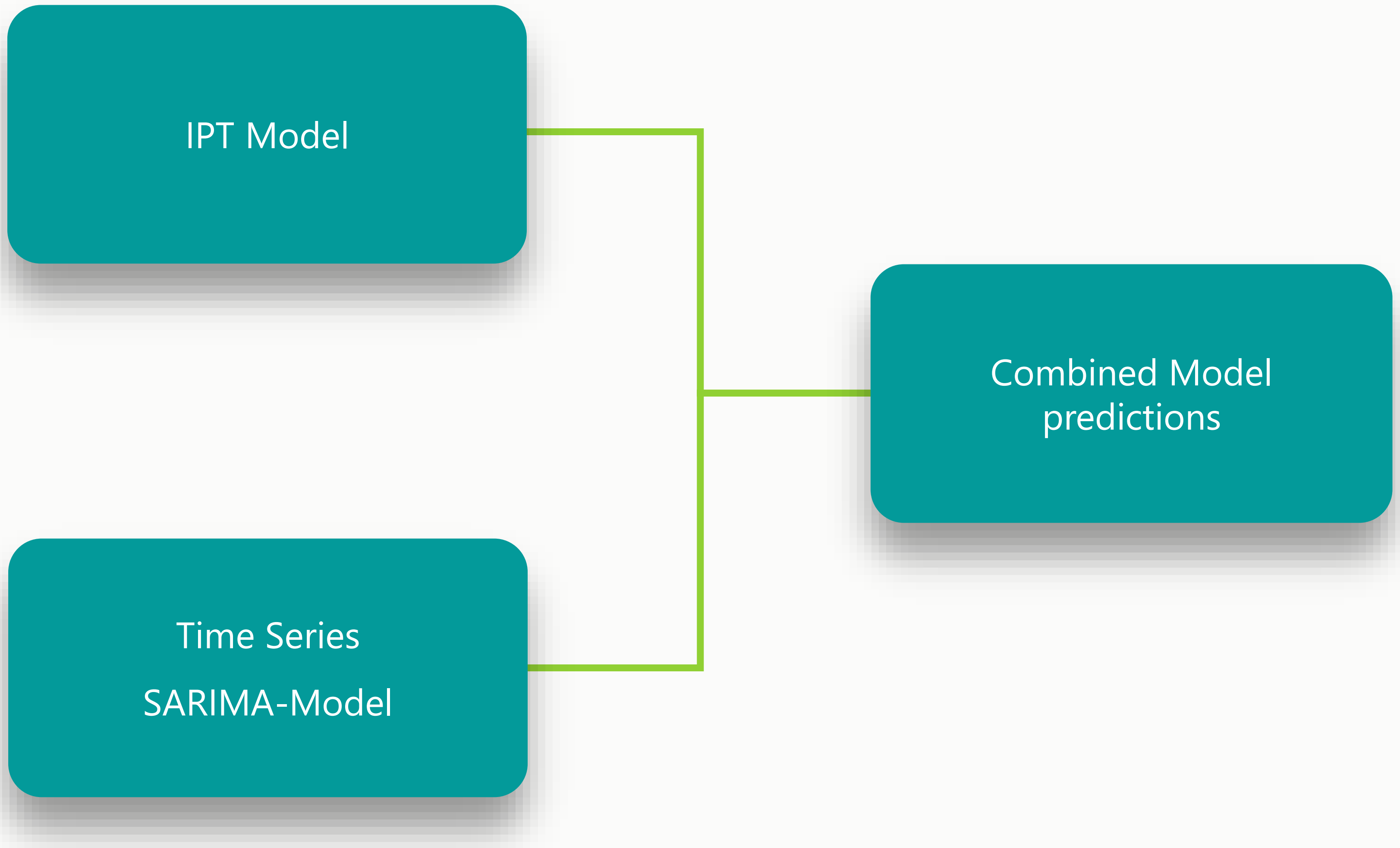
Adjust results for weekends and holidays

t_0 t_1 t_{n+1} t_{n+2} t_{n+3} t_{n+4} t_{n+5}

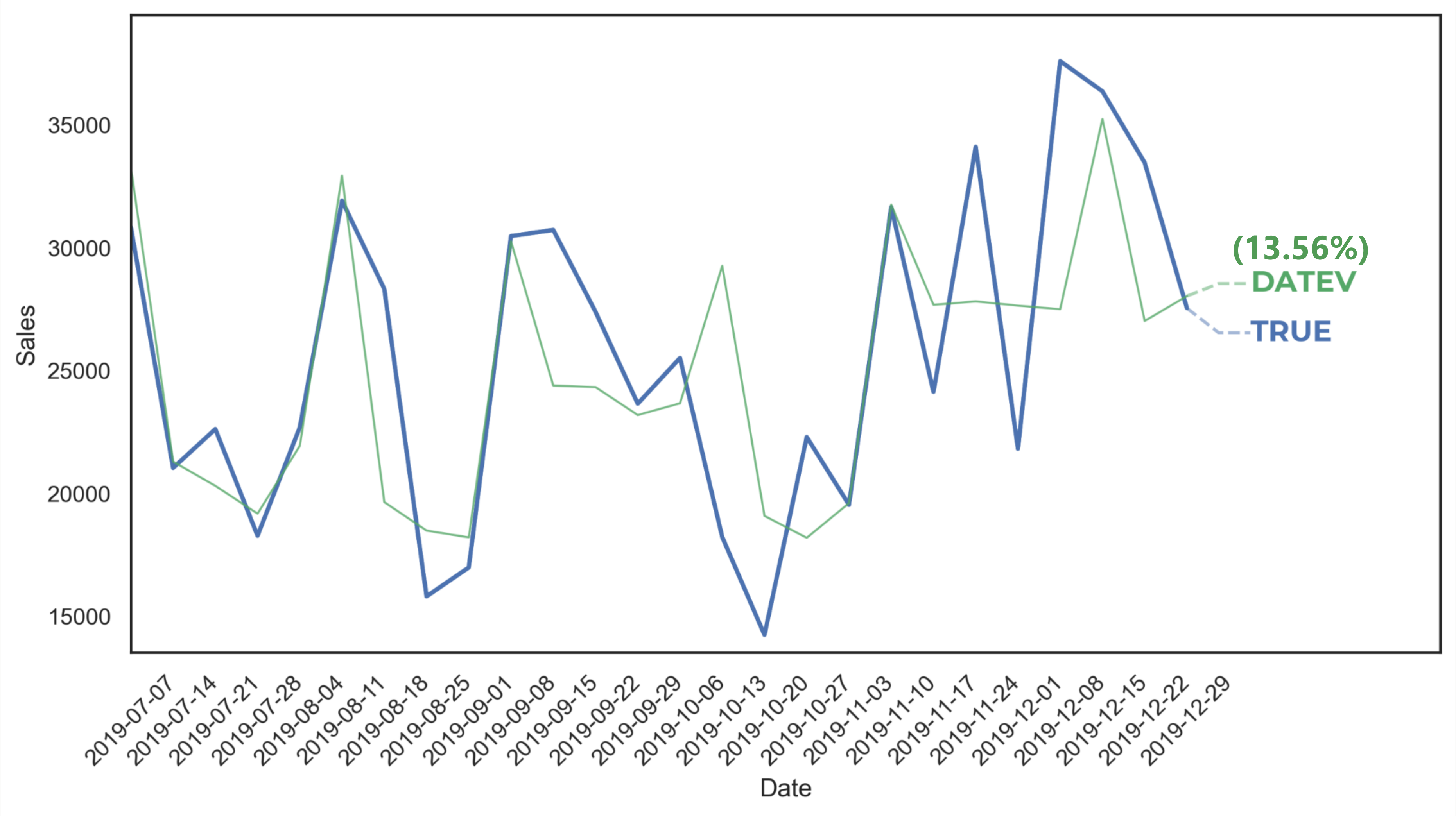
Model Approach



Results


IPT Model
Combined Model predictions
Time Series
SARIMA-Model

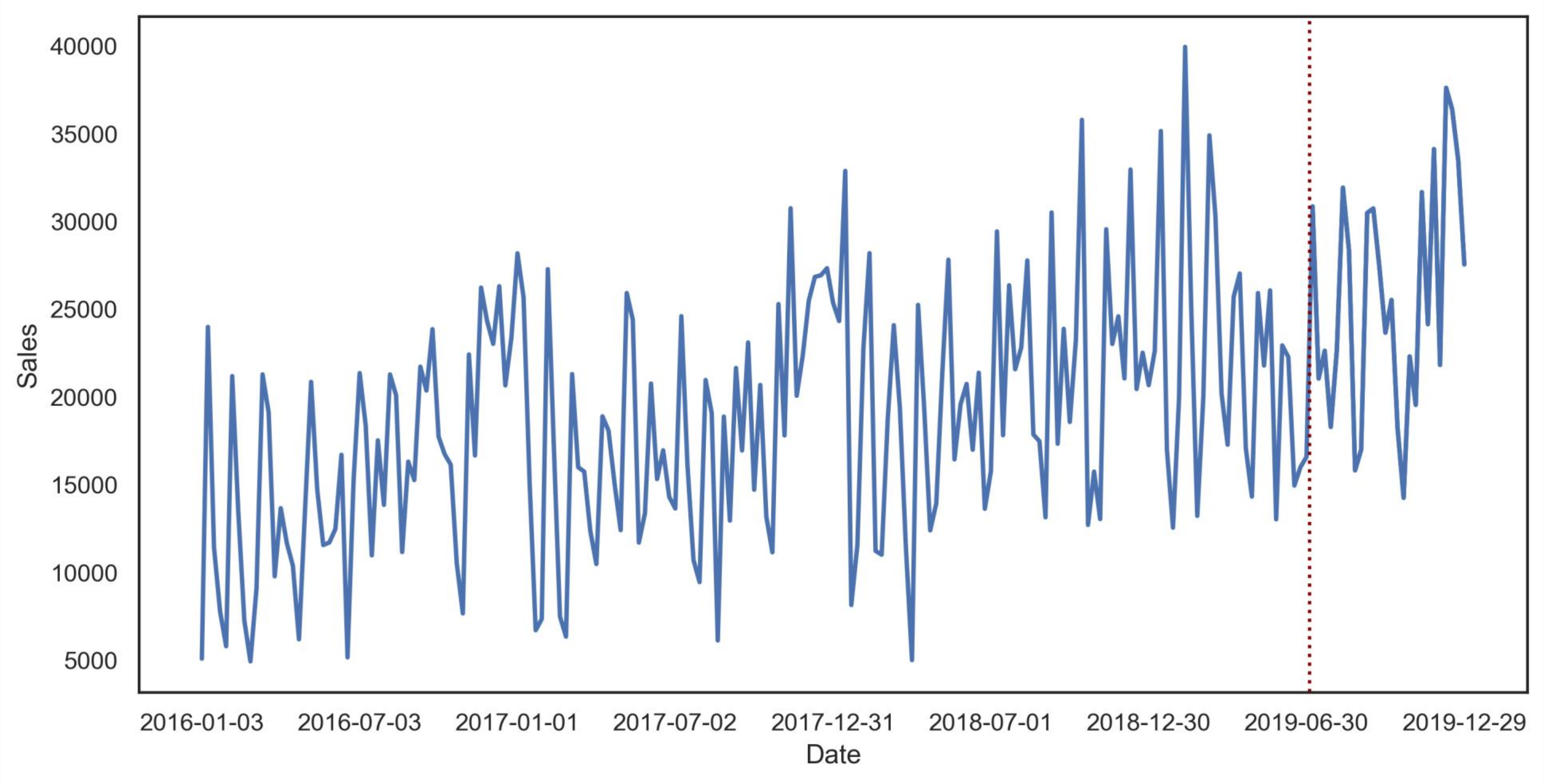
Results



Sales
35000
30000
25000
20000
15000
Date
2019-07-07
2019-07-14
2019-07-21
2019-07-28
2019-08-04
2019-08-11
2019-08-18
2019-08-25
2019-09-01
2019-09-08
2019-09-15
2019-09-22
2019-09-29
2019-10-06
2019-10-13
2019-10-20
2019-10-27
2019-11-03
2019-11-10
2019-11-17
2019-11-24
2019-12-01
2019-12-08
2019-12-15
2019-12-22
2019-12-29
(13.56%)
DATEV
TRUE

Results + Winners

Evaluation: Input Data



Evaluation: Forecast Plot