INFORMATIONSFABRIK

DATEN VERSTEHEN, ENTSCHEIDUNGEN TREFFEN

Automating Feature Engineering for Supervised Learning?

methods, open-source tools and prospects.

Dr. Thorben Jensen

Jonas Jostmann

INFORMATIONSFABRIK

MÜNSTER

Continental

DONNER & REUSCHEL
PRIVATBANK SEIT 1798

Deutsche Bank

GENERALI
Versicherungen

COMMERZBANK

Content

- Feature Engineering
- Open-Source Tools
- Case Study

INFORMATIONSFABRIK

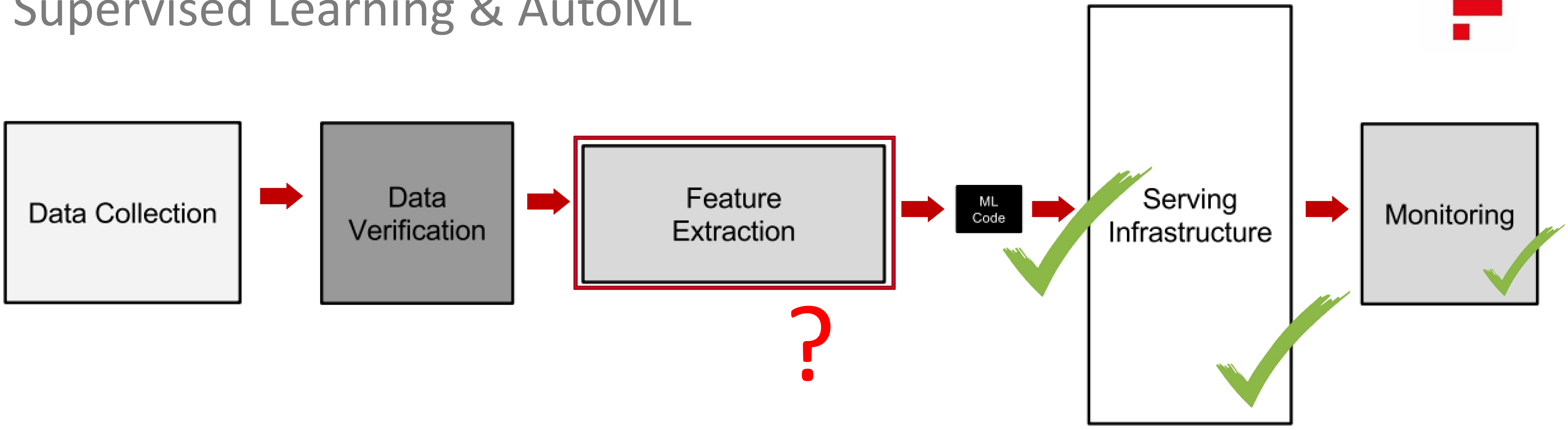
DATEN VERSTEHEN, ENTSCHEIDUNGEN TREFFEN

Feature Engineering

Workload in Supervised Learning

ML
Code

Supervised Learning & AutoML

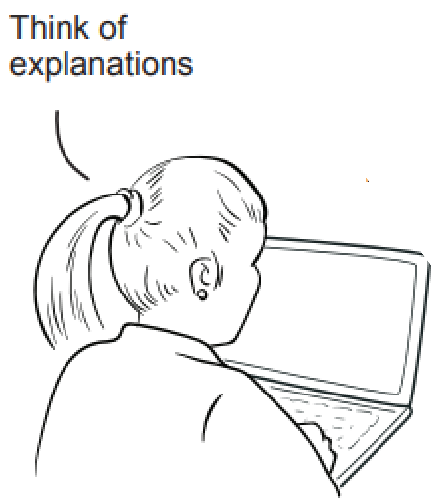


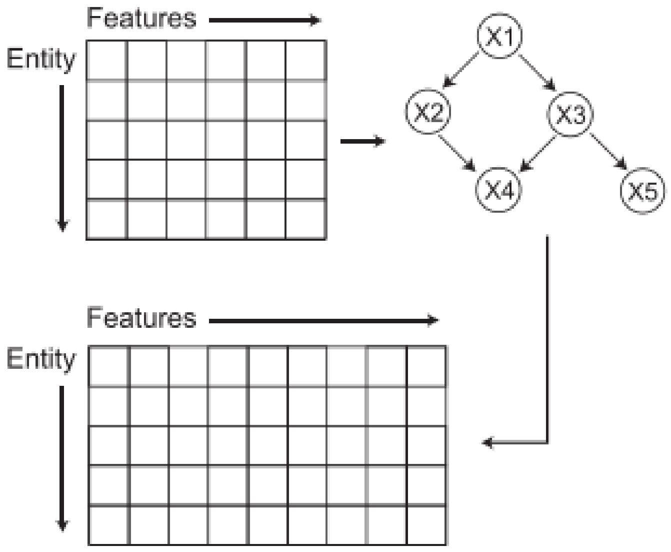
Machine Learning eliminates the need for detailed programming effort. (Arthur Samuel, 1959)

AutoML *eliminates the need for detailed Machine Learning effort.*

Sculley, David, et al. "Hidden technical debt in machine learning systems." *Advances in neural information processing systems*. 2015.
Samuel, Arthur L. "Some studies in machine learning using the game of checkers. II—recent progress." *Computer Games I*. Springer, New York, NY, 1988. 366-400.

Feature Engineering in Practice





DATE	SALES (TARGET)	WEEKDAY	OPEN	STATE HOLIDAY	„BRIDGE DAY“	LAG(OPEN)
2014-04-29	5923	Friday	1	0	0	1
2014-04-30	5870	Saturday	1	0	0	1
2014-05-01	**0**	Thursday	0	1	0	1
2014-05-02	**6790**	Friday	1	0	**1**	**0**
2014-05-03	5498	Saturday	1	0	0	1

http://www.jmaxkanter.com/static/papers/DSAA_DSM_2015.pdf https://www.kaggle.com/c/rossmann-store-sales

Current Challenge & Vision for Automated Feature Engineering

CHALLENGES	VISION
Time-consuming task	• Easy setup (data and config)
Good features require domain knowledge	• Handle time in data
Features often not transferable between data	• Features self-explanatory
	• High potential complexity of features
	• Minimal „garbage" features

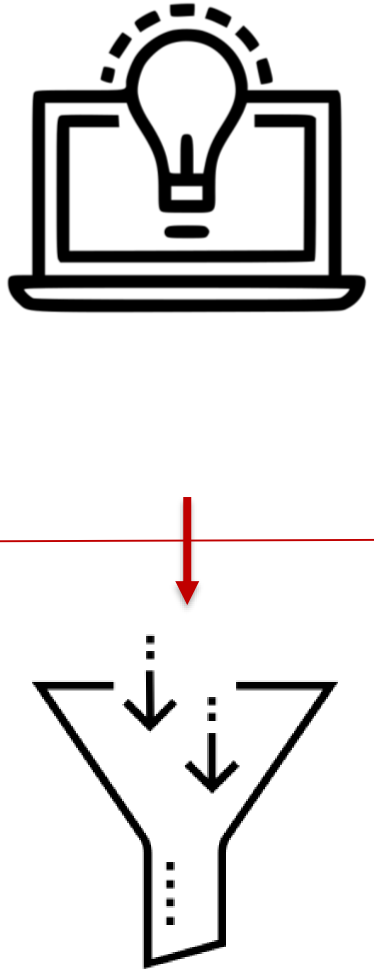
INFORMATIONSFABRIK

DATEN VERSTEHEN, ENTSCHEIDUNGEN TREFFEN

Open Source Tools

Options for Automated Feature Engineering

GENERATION



SELECTION

Options for Automated Feature Engineering

GENERATION

DIMENSIONALITY REDUCTION

Combine similar features, e.g. PCA.

TRANSFORMATION

Apply operators (+, *, Min, …)

RELATIONAL OPERATIONS

SQL-like operations

TIME-SERIES FEATURES

Pre-defined functions (e.g. trend)

PRE-TRAINED EMBEDDINGS

Extract „features“ with pre-trained models

META-LEARNING

Predict good features based on previous datasets

SELECTION

FILTERING

Select features by testing interrelation with target variable.

WRAPPING

Search best feature set, by model performance.

Options for Automated Feature Engineering

GENERATION

DIMENSIONALITY REDUCTION	TRANSFORMATION	RELATIONAL OPERATIONS	TIME-SERIES FEATURES	PRE-TRAINED EMBEDDINGS	META-LEARNING
Combine similar features, e.g. PCA.	Apply operators (+, *, Min, …)	SQL-like operations	Pre-defined functions (e.g. trend)	Extract „features“ with pre-trained models	Predict good features based on previous datasets

(autofeat)

TSFRESH

TPOT

Featuretools

SELECTION

FILTERING

Select features by testing interrelation with target variable.

TSFRESH

WRAPPING

Search best feature set, by model performance.

TPOT (autofeat)

Featuretools

Database Tables:

OrderID	Customer ID
1	2
2	…
3	…
4	2
…	…

ID	OrderID	ProductId
1	1	3
2	1	1
3	…	…
4	4	3
…	…	…

ProductID	Price
1	$100
2	…
3	$200
4	…
…	…

Features on Customer?

Aggregations (per value):

- sum
- trend
- time_since_first
- …

Transformations (per entity):

- month
- weekday
- num_words
- …

Base Column

ProductID	Price
1	$100
2	…
3	$200
4	…
…	…

‚JOIN'

Deep Feature, d=1

ID	OrderID	ProductId	Product.Price
1	1	3	$200
2	1	1	$100
3	…	…	…
4	4	3	$200
…	…	…	…

SUM

Deep Feature, d=2

OrderID	Customer ID	SUM(Product.Price)
1	2	$300
2	…	…
3	…	…
4	2	$200
…	…	…

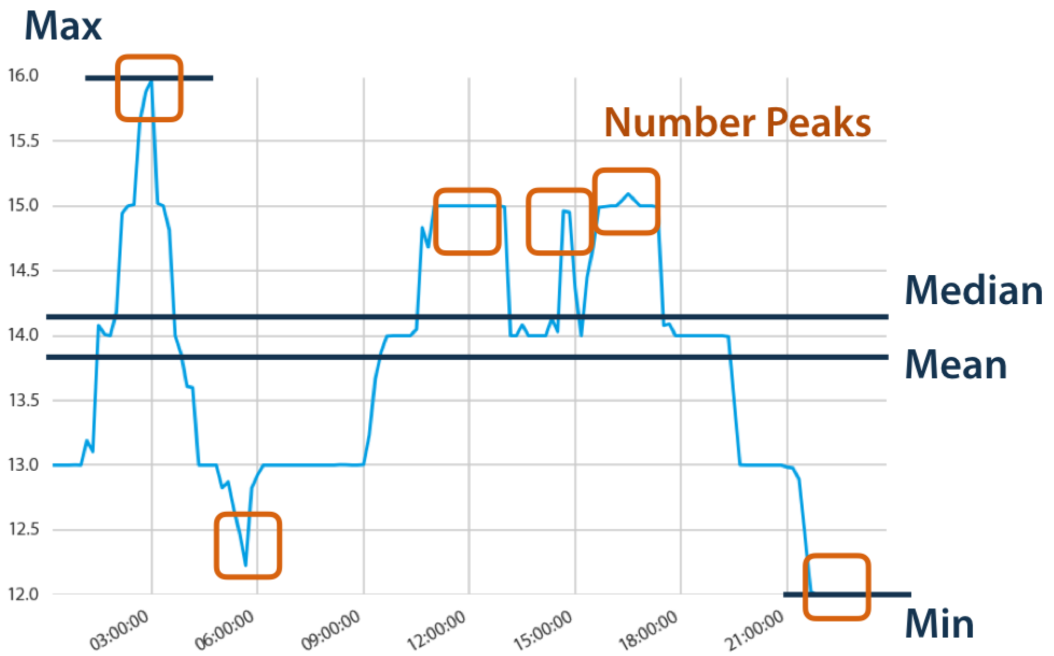
AVG

Deep Feature, d=3

CustomerID	Gender	Age	AVG(Orders.SUM(Product.Price))
1	…	…	…
2	f	45	$250
3	…	…	…
4	…	…	…
…	…	…	…

Kanter, J.M. and Veeramachaneni, K., 2015. Deep feature synthesis: Towards automating data science endeavors. *DSAA,* IEEE.

TSFRESH



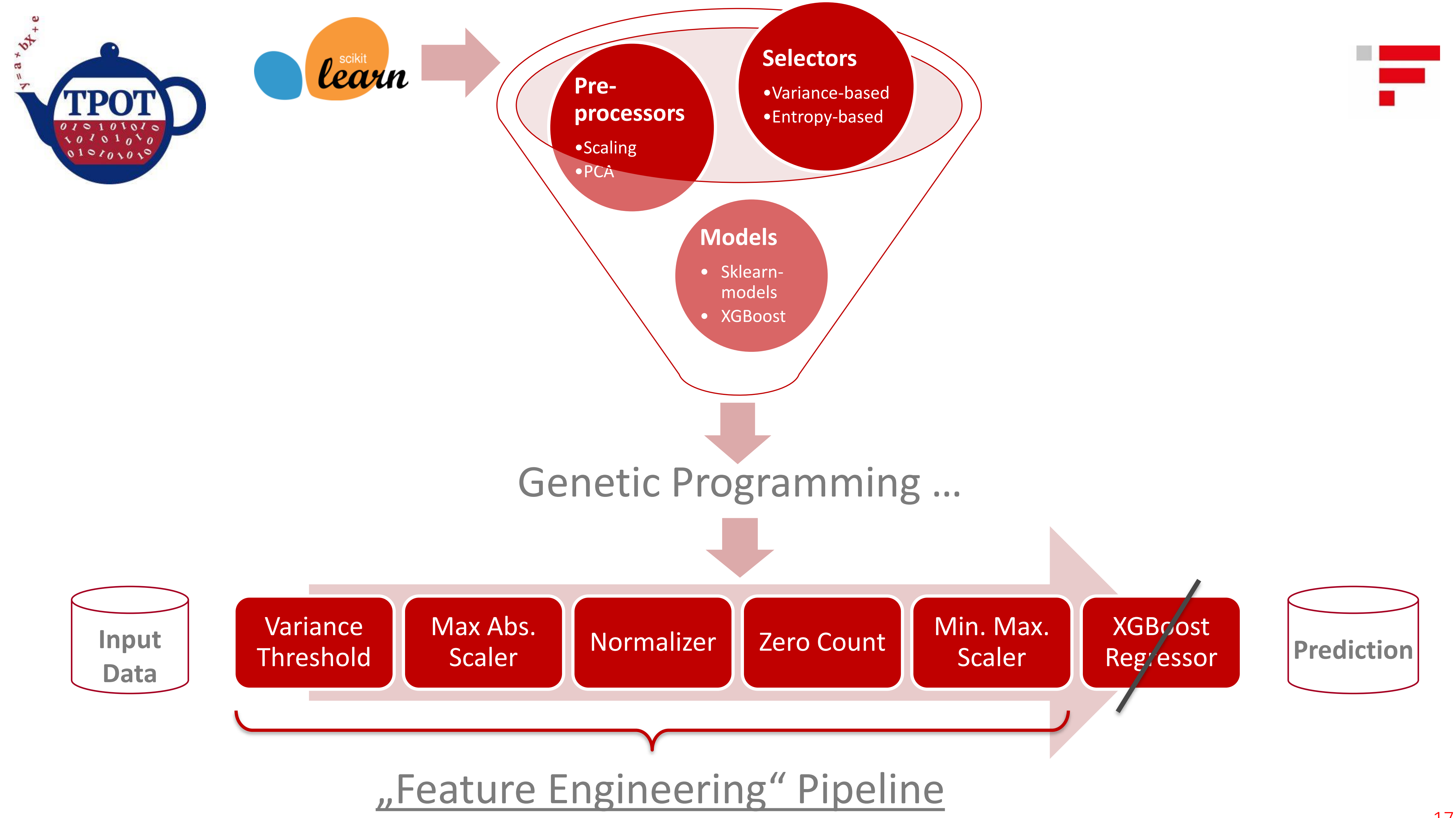
Numerous features:

- linear_trend(x, “slope”)
- mean_change(x)
- autocorrelation(x, lag: int)
- spkt_welch_density(x, …)
- …

Filtering most promising features:

1. Variables tested individually
2. Statistical tests on interaction with target variable
3. Discarding variables with low significance

https://tsfresh.readthedocs.io


Pre-processors
•Scaling
•PCA
Selectors
•Variance-based
•Entropy-based
Models
• Sklearn-models
• XGBoost
Genetic Programming …
Input Data
Variance Threshold
Max Abs. Scaler
Normalizer
Zero Count
Min. Max. Scaler
XGBoost Regressor
Prediction
„Feature Engineering“ Pipeline

INFORMATIONSFABRIK

DATEN VERSTEHEN, ENTSCHEIDUNGEN TREFFEN

Case Study

Goals and Dataset

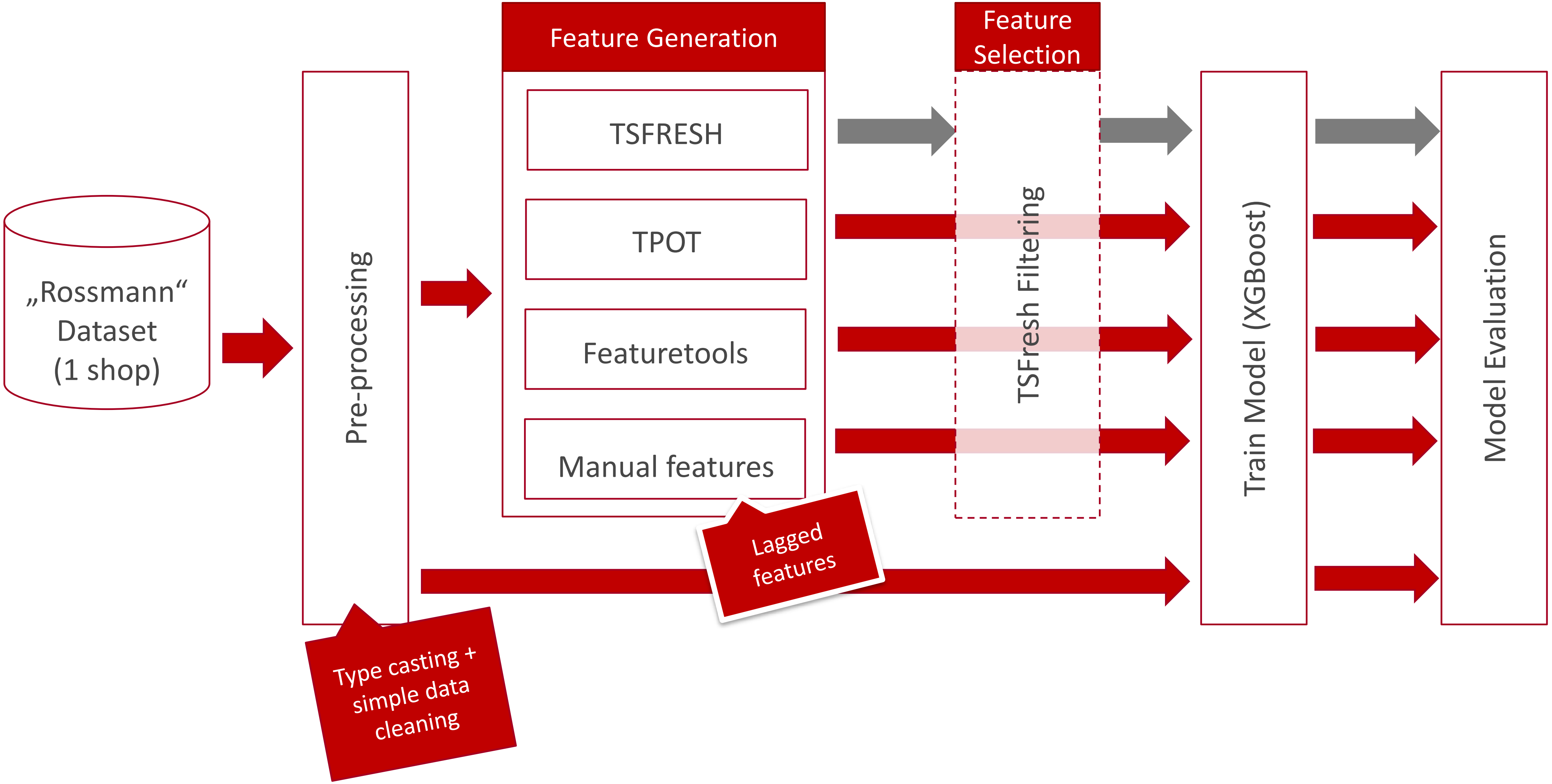
- Goals:
  - Test libraries **at default settings**
  - Achieve **general understanding** of tools
  - **Compare performance**

- Machine-learning Task
  - **Time series forecasting** („Rossmann Challenge“)
  - **XGBoost with fixed settings**

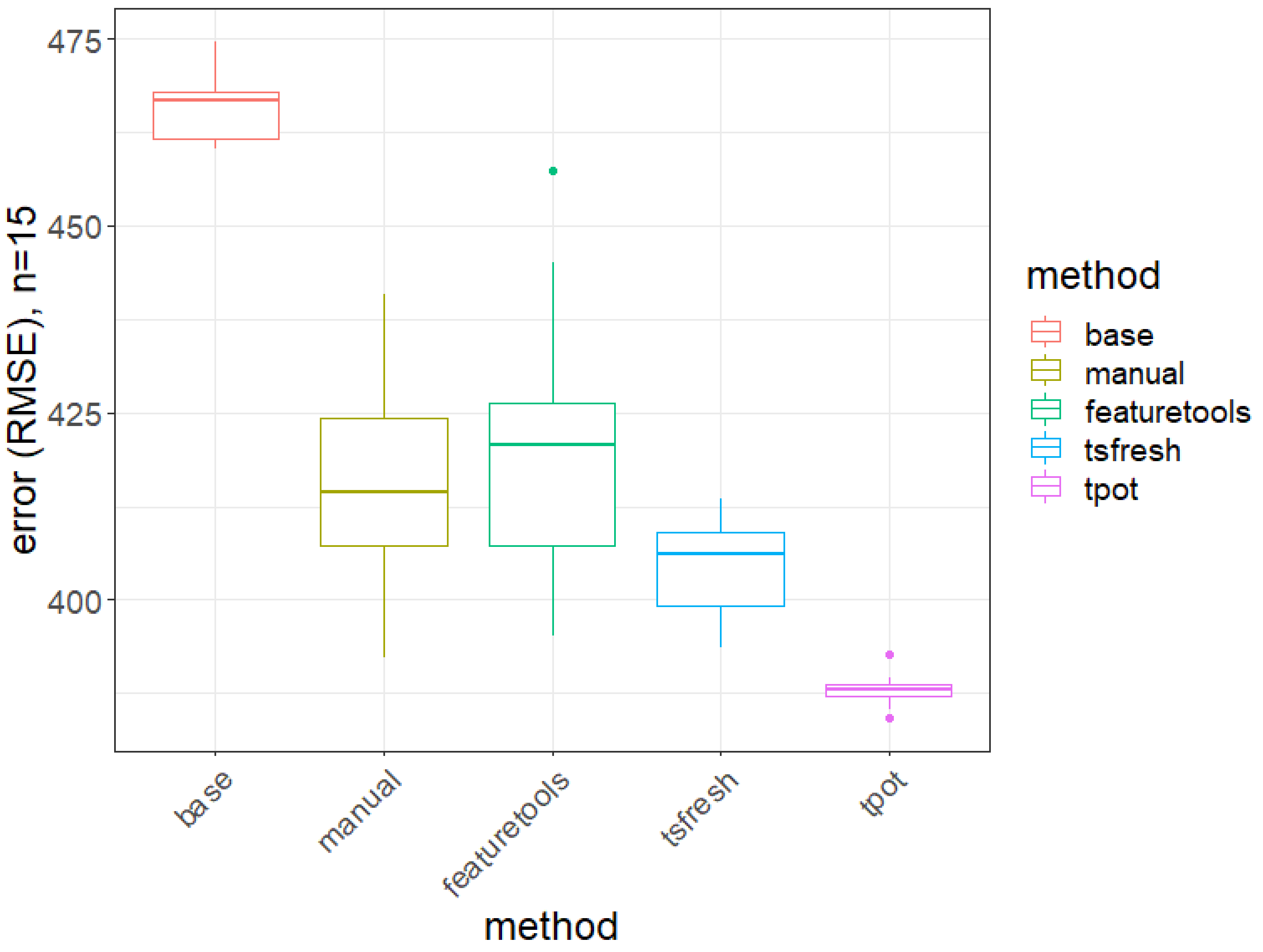
- Out of scope
  - Not finding best library *in general*

https://www.kaggle.com/c/rossmann-store-sales

Design of Experiment



Results Performance: methods at default usage

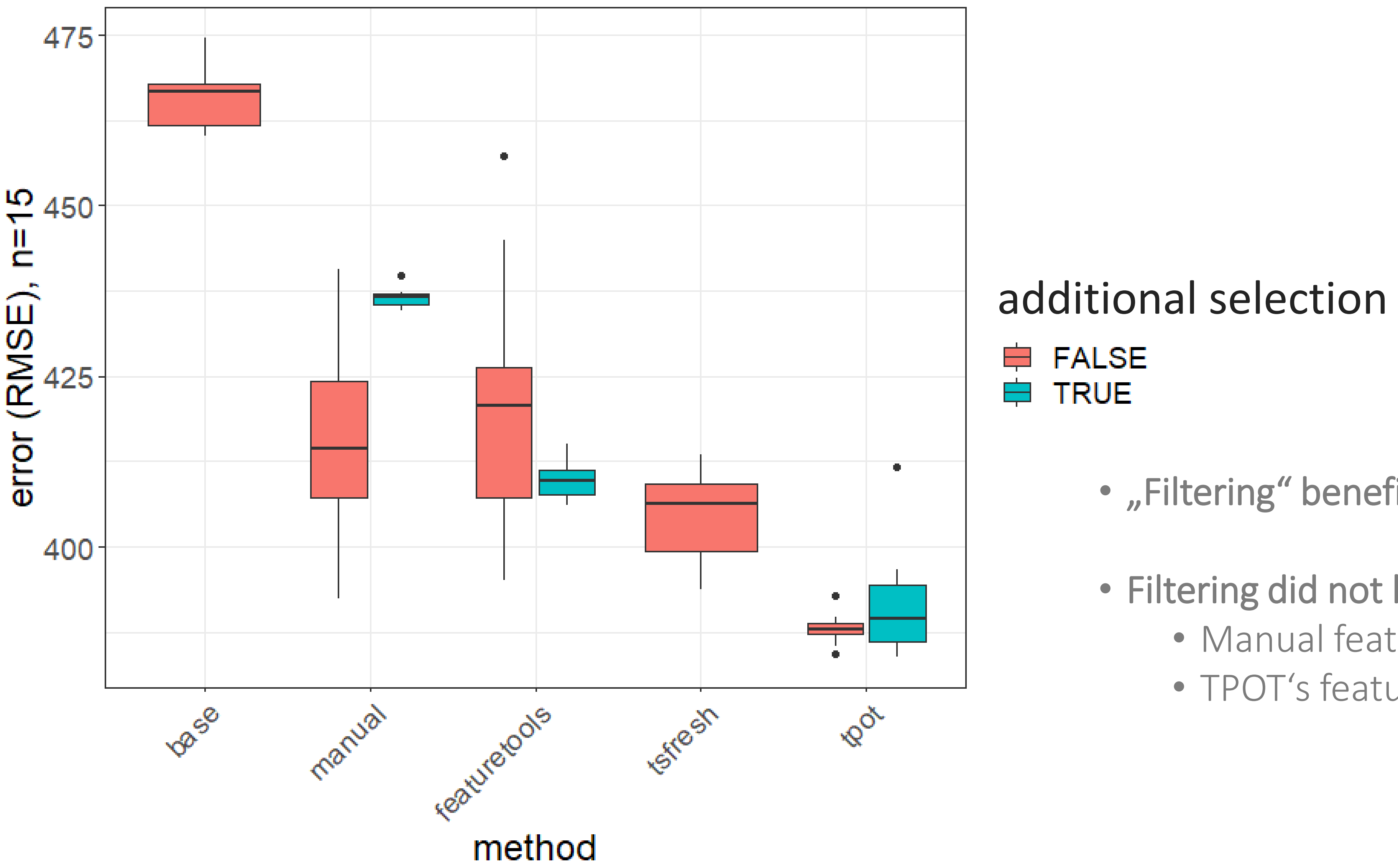


Order of performance:

1. TPOT
2. TSFRESH
3. Featuretools / Manual
5. Base

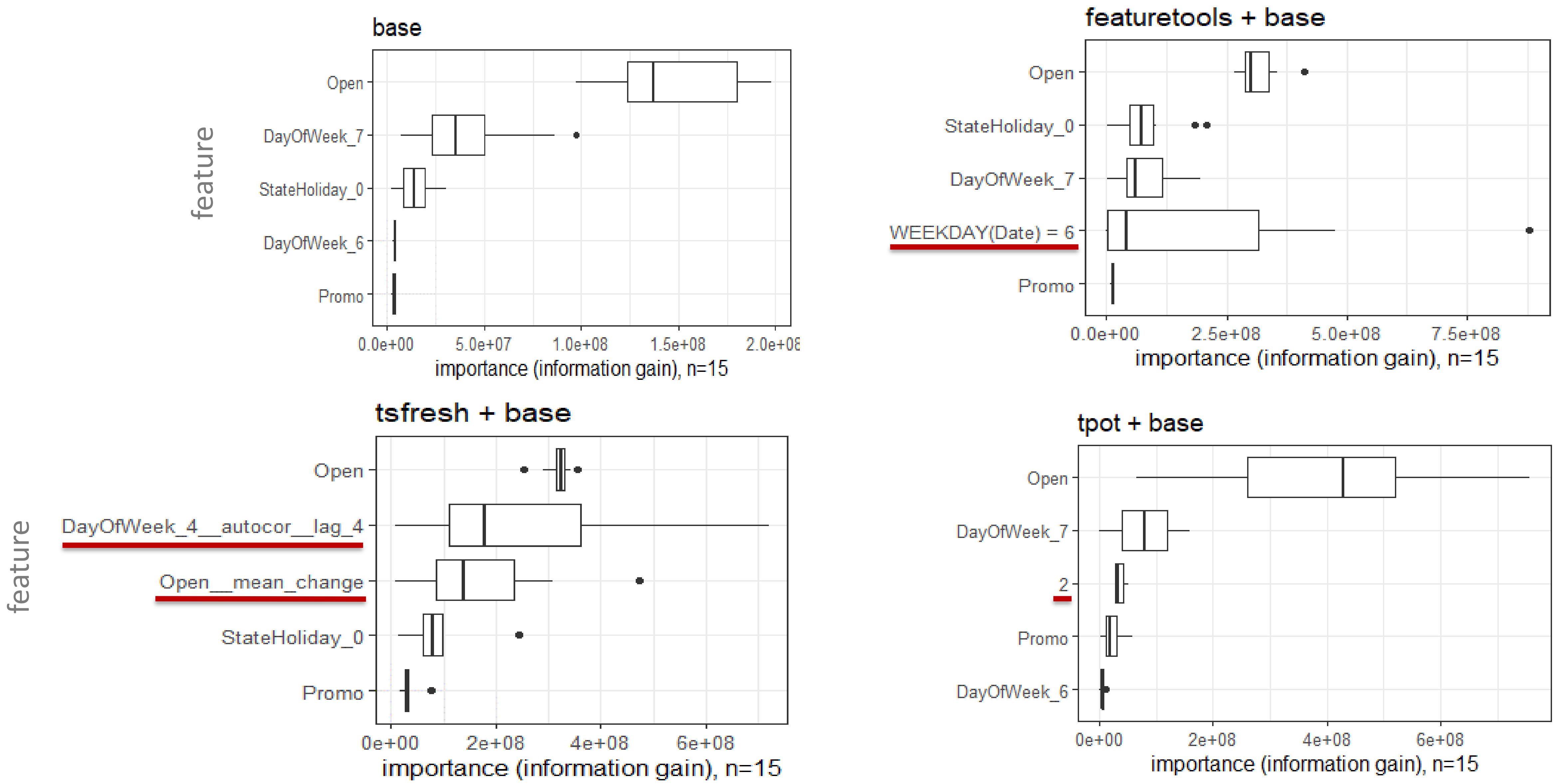
Adding features helped

Results Performance: selecting features with TSFRESH-method



- „Filtering“ benefited featuretools
- Filtering did not help others:
  - Manual features are often meaningful
  - TPOT‘s features selected by optimizer

Comparing engineered features to ‚base‘ features



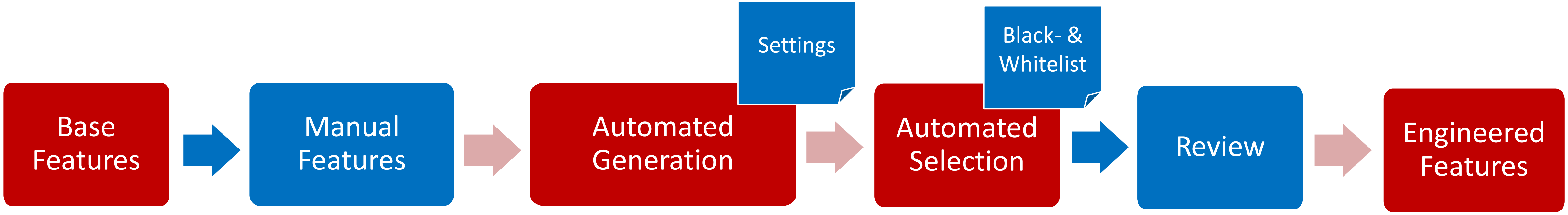
Comparison of Libraries (at default settings)

	Featuretools	TSFRESH	TPOT
Easy setup (data and config)	◐	✓	✓
Handle time in data	✓	✓	✗
Features self-explanatory	✓	◐	✗
High potential complexity of features	◐	◐	◐
Minimal „garbage“ features	✗	◐	✓

Discussion & Conclusion

Conclusion on „Automated Feature Engineering“

- Preparation of data was still required
  - E.g. preventing time-related data leakage
- „Full automation“ possible with default settings, but stays below potential
  - Default settings might be limiting, e.g. featuretools
- Human intervention would be beneficial



Outlook

- Code open to Reviews and Pull Requests: *github.com/informationsfabrik/feature-engineering*
- Can we find „better“ default parameters for libraries?
- Open Source Libraries for other methods, e.g. Meta Learning?
- Best way to combine manual and automated Feature Engineering?
  - What do you do or recommend?

TAK

DANK U WEL

谢谢

GRACIAS

KÖSZÖNÖM

CHOKRANE

СПАСИБО

TERIMA KASIH

THANK YOU

VIELEN DANK

GRAZIE

TESEKKÜR EDERIM

DZIĘKUJĘ

MERCI

ขอบคุณครับ

TÄNAN

ARIGATÔ

HVALA