# Data Science in Finance – Machine Learning for Overdue Invoice Prediction

D.Sc.(Tech.) Andreas Henelius, Data Scientist, OP Financial Group



04.12.2020

OP

# Contents

- About OP Financial Group
- Machine learning for overdue invoice prediction
  - Background
  - Problem formulation
  - Feature engineering and challenges
  - Model choice
  - Technology and implementation

# OP Financial Group's structure


12,000 OP employees

# Transformation of the world around us is constantly creating new opportunities as well as challenges to us and our customers

OP Financial Group's worldviews 2020–2025

**Economy:** Slow growth

**Regulation:** Regulation challenges

**Technology:** Business involves technology management

**Competitive environment:** More intense and diverse competition

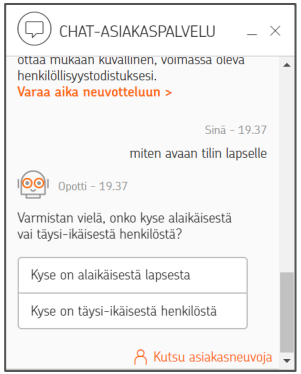
**Customer behaviour:** Customer experience is crucial

# OP AI Use Cases

## – serious impact on operational efficiency

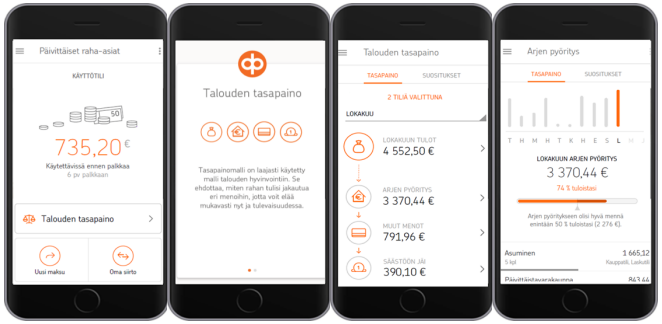
### Insurance claim analytics

### Chatbots




### Property Value Estimation


### Personal Financial Management




### Group customer profile




OP

# Contents

- Background
- Problem formulation
- Feature engineering and challenges
- Model choice
- Technology and implementation



OP

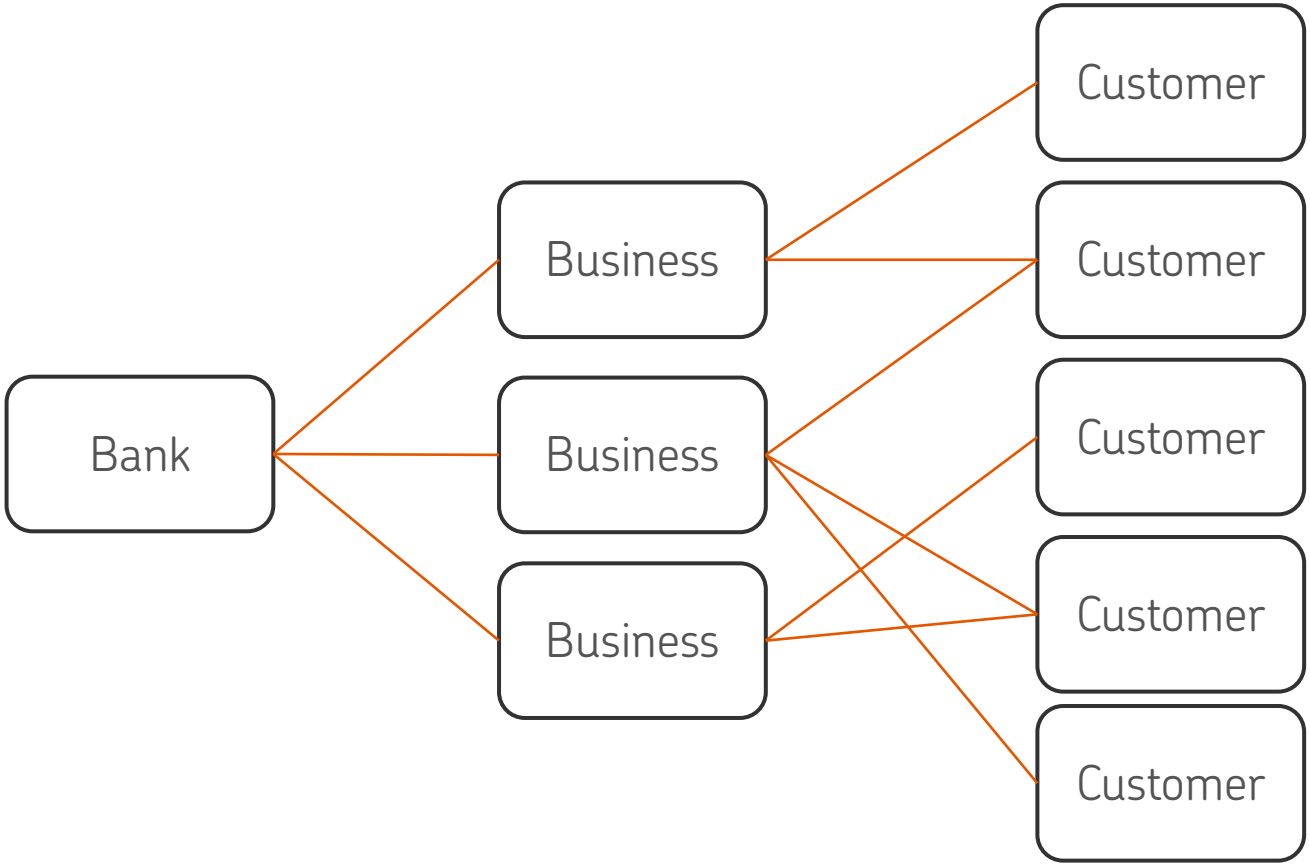
# Background - Application areas

## Optimising cash collection

- See Zeng, Sai, et al. *Using predictive analysis to improve invoice-to-cash collection. KDD 2008.*
- Prioritise and focus resources

## Invoice financing

- Considered here
- Short-term borrowing by companies against their invoices
- E.g., to manage cash flow



# Estimating invoice value

## Net present value (NPV)

- Mostly influenced by the payback time
- The later the expected payback, the lower the value

## Problem formulation: classify invoices

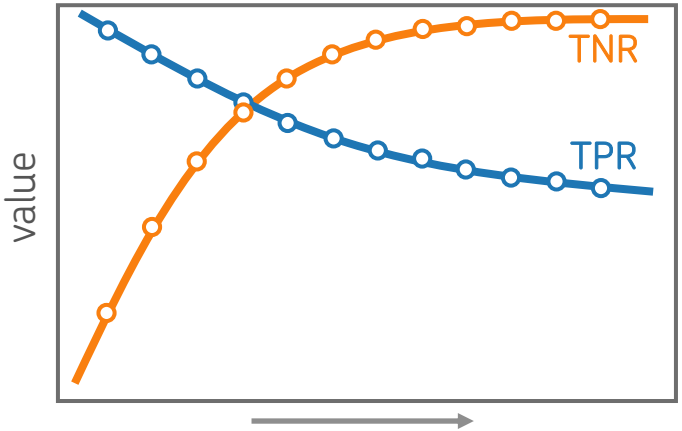
- On time — less than 60 days late
- Overdue — at least 60 days late

# Considerations

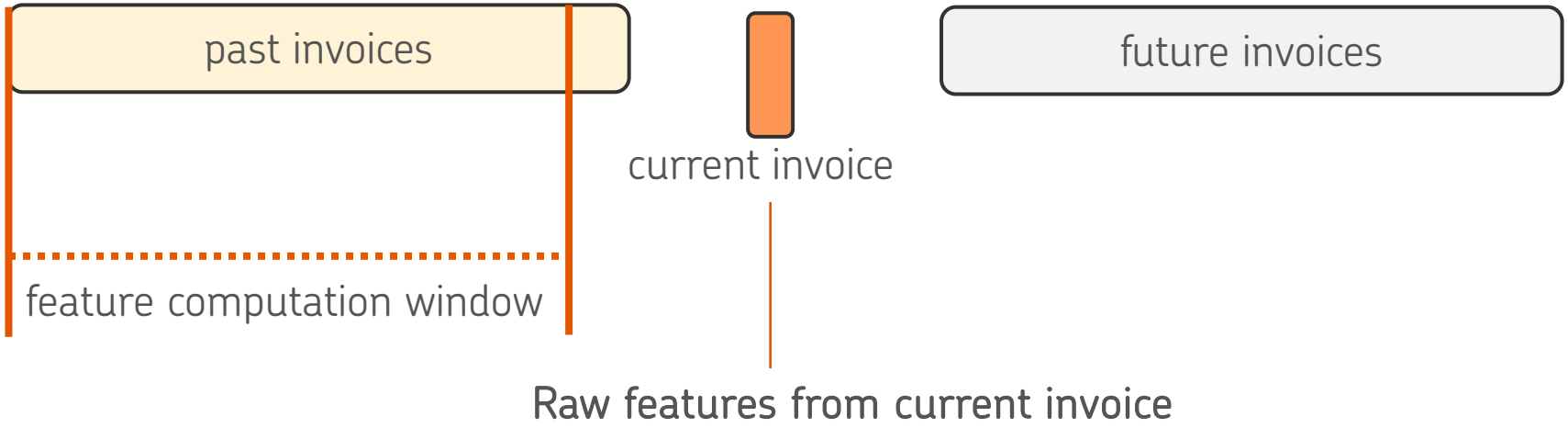
- Class imbalance

- Some strategies
  - Resampling techniques (used here)
  - Cost-sensitive learning  $C(i, j)$
    - *Elkan. The foundations of Cost-Sensitive Learning. IJCAI 2001.*
    - *Domingos. Metacost: A general method for making classifiers cost-sensitive. KDD 1999.*

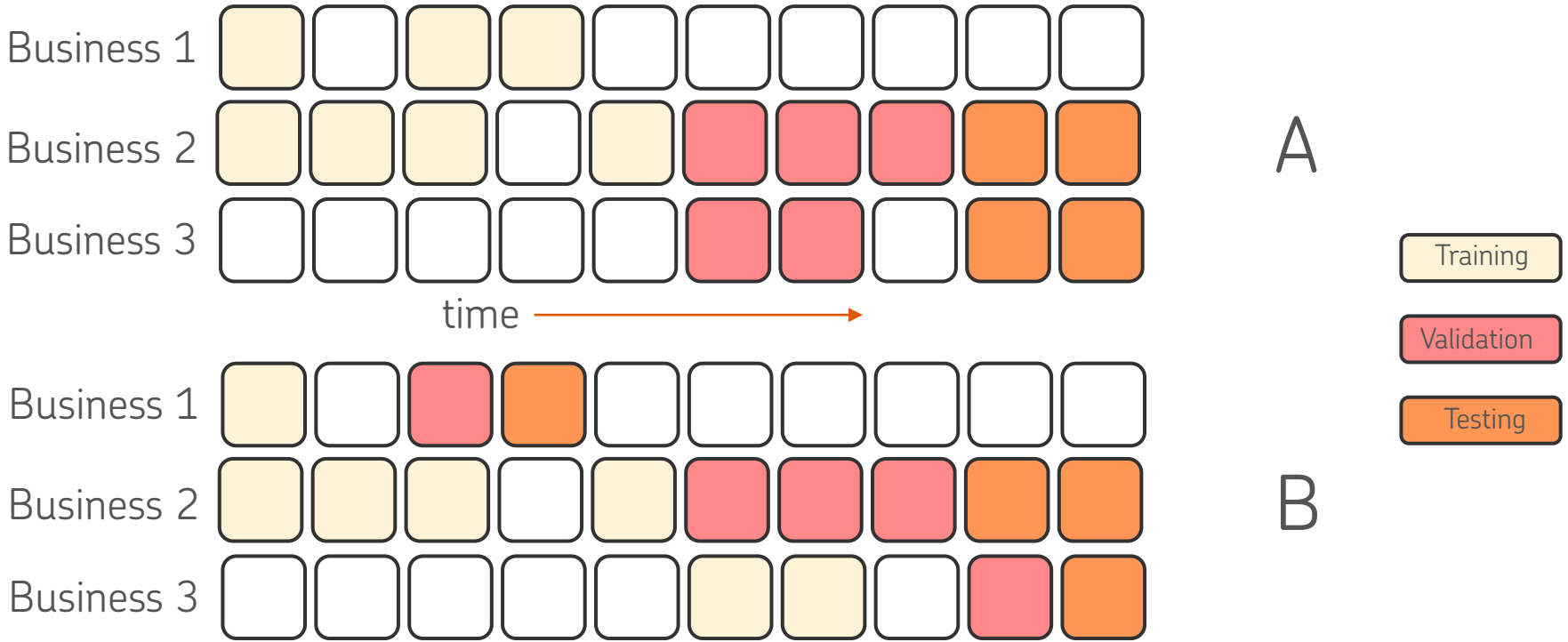
| Predicted class \ True class | + | – |
|---|---|---|
| + | TP | FP |
| – | FN | TN |

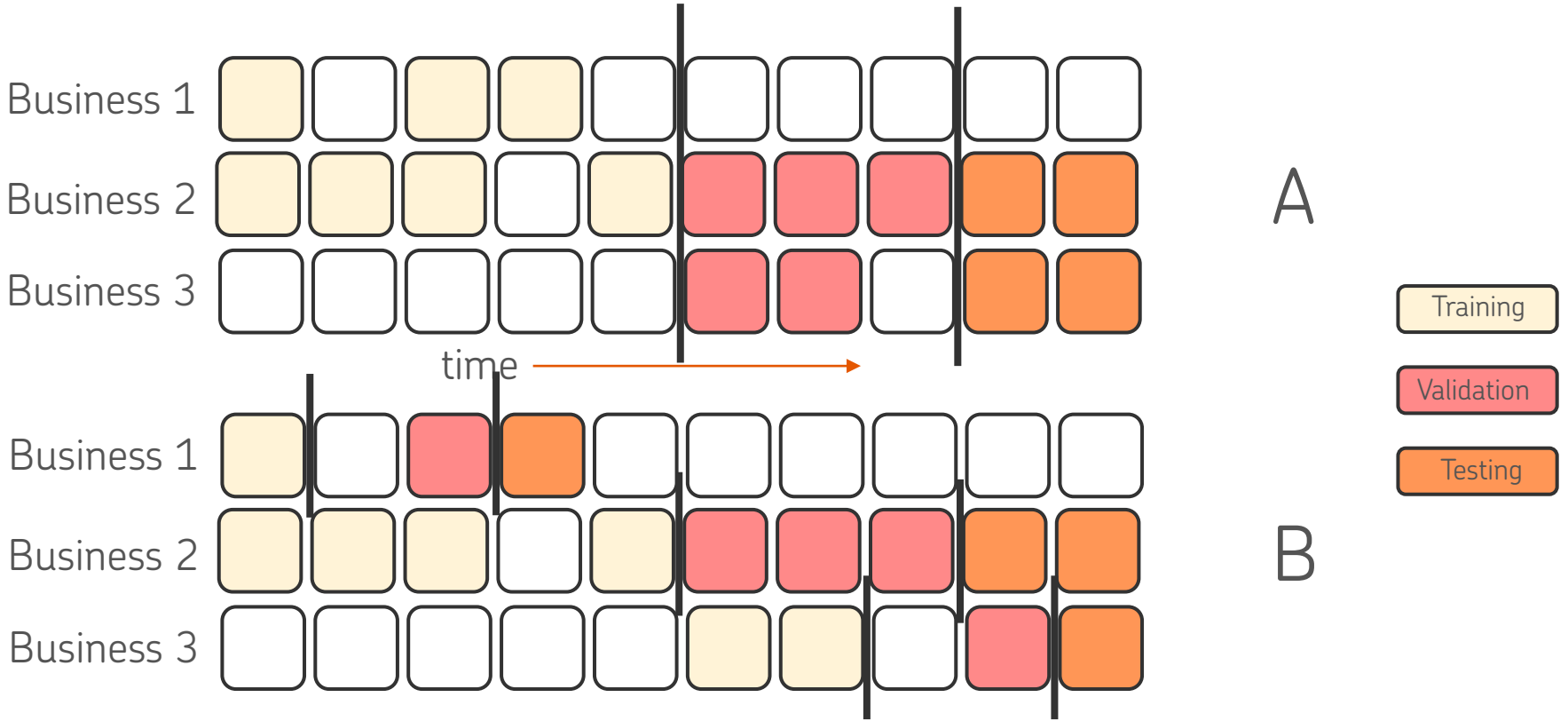


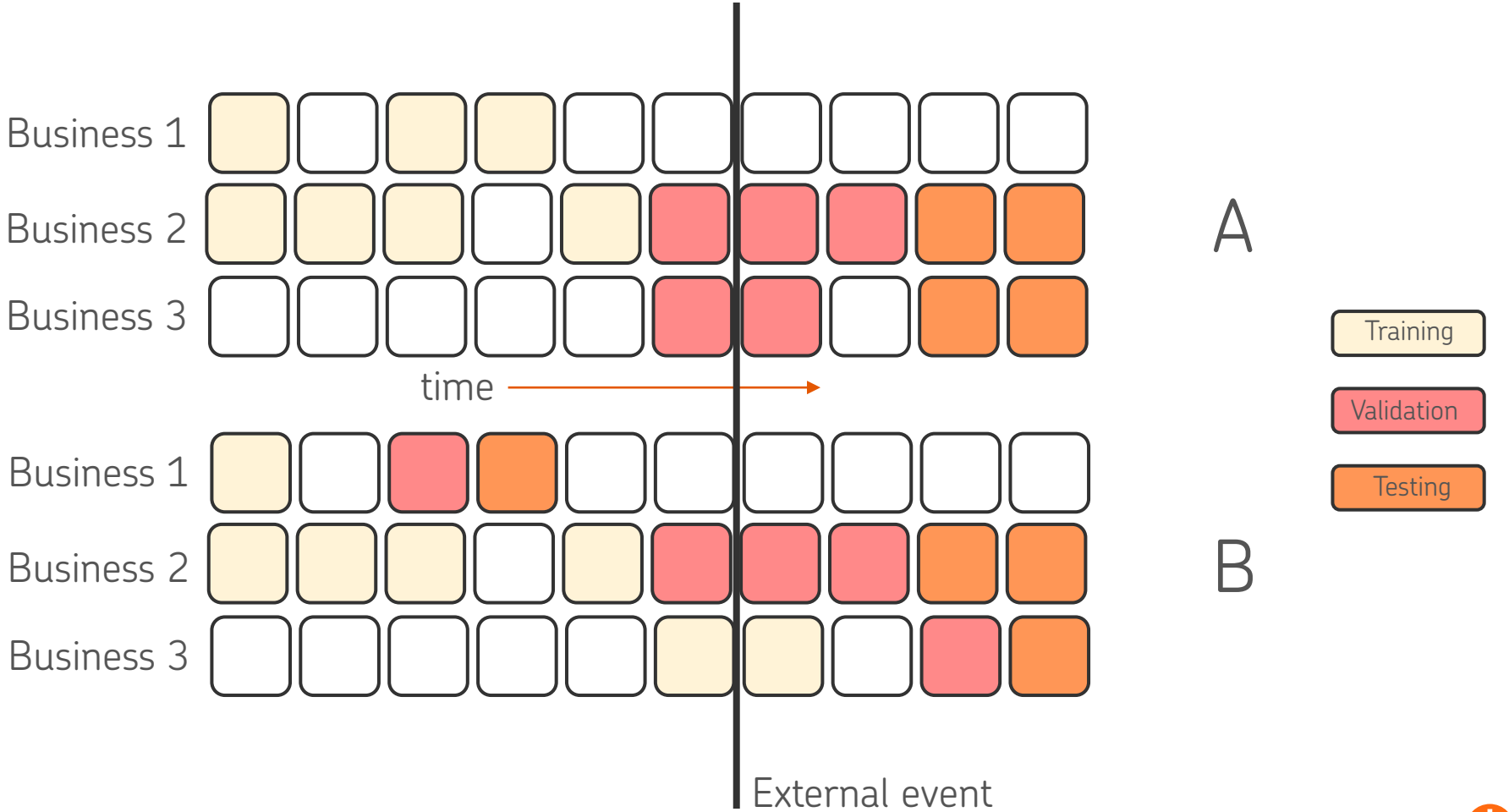
# Feature engineering




Cumulative features from history







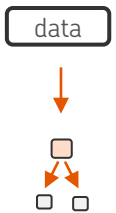
# Model choice

- Baseline models
  - No model, “simple” models
  - Decision trees
  - Logistic regression
  - Naïve Bayes

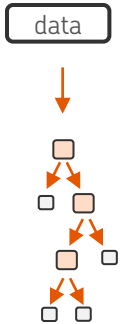
- Choice: gradient boosted tree (LightGBM)

# Stumps, trees and forests

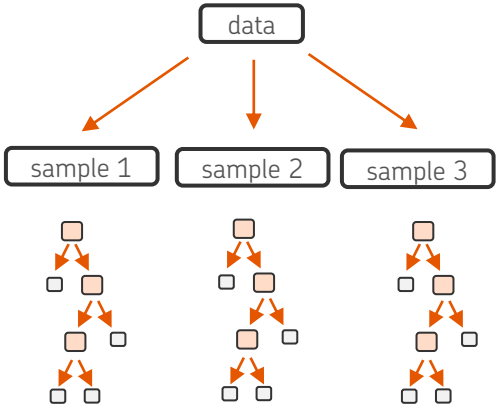
Decision stump



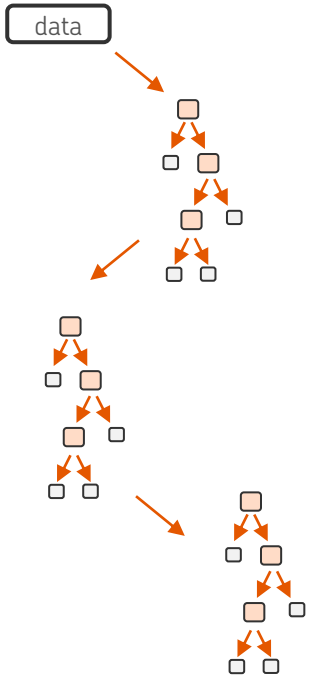

Decision tree




Bagging (Random Forest)



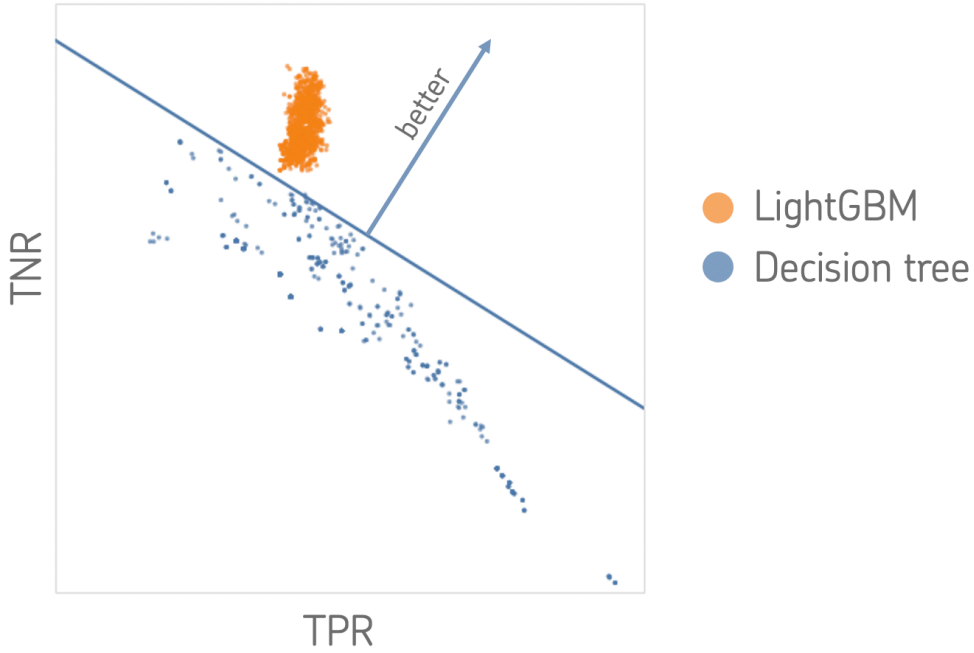

Boosting (XGBoost, LightGBM, ...)






OP

# Model stability



# Technical implementation

Amazon EMR



Amazon Simple Storage Service (S3)



AWS Lambda



Users





OP

# Summary

- Class imbalance
  - Sampling strategies, cost-sensitive learning
- Take temporal structures in the data into account
- Different types of tree models
- Compare models
  - Baseline models
  - Stability
- Consider the interpretability of the model



OP