Who are we?

Balázs Kégl

- Directeur de recherche CNRS
  - machine learning (20 years)
    interfacing with particle physics (10 years)
- Director of the Paris-Saclay Center for Data Science
  - interfacing with biology, economy, climatology, chemistry, etc. (3 years)
- Data science consulting and training for non-IT industry (3 years)

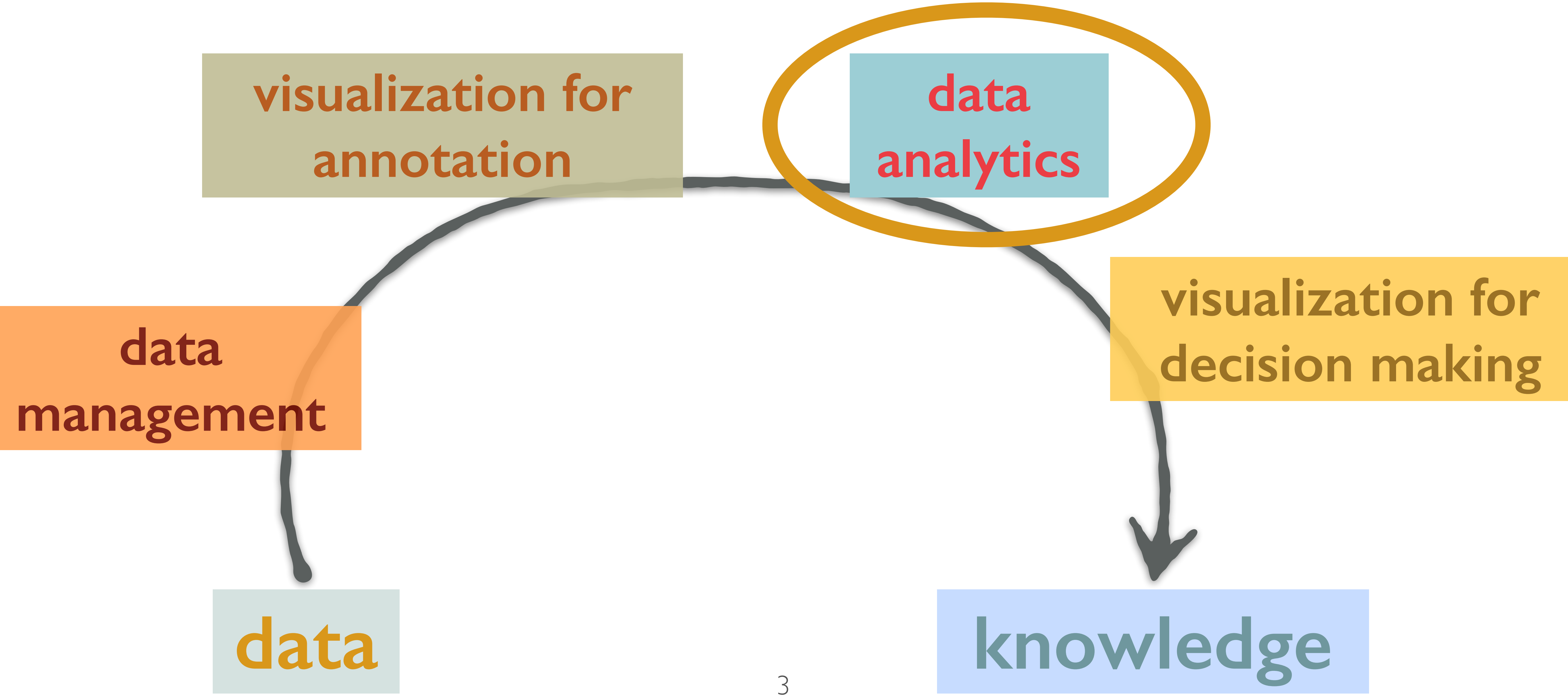
Who are we?

Alexandre Gramfort

- Chercheur at INRIA Saclay
- http://alexandre.gramfort.net
- alexandre.gramfort@inria.fr
- co-author of scikit-learn
- **Research topics:** machine learning, non-linear optimization, signal processing, applications in health care and particularly in neuroscience.

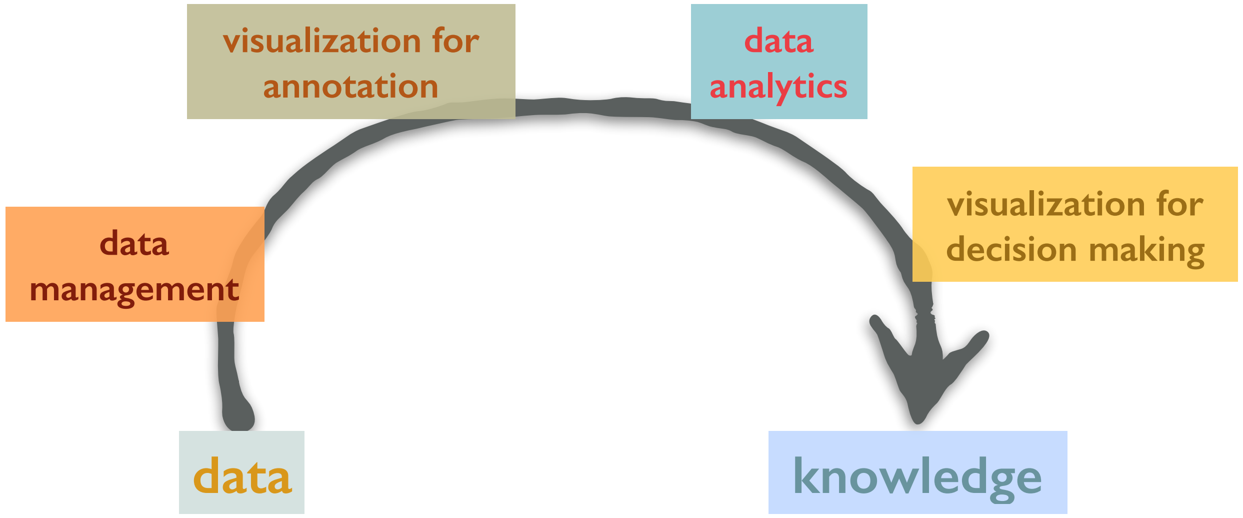
The full data chain

The products a data scientist/engineer can end up working on, after the business case is identified


visualization for annotation
data analytics
data management
visualization for decision making
data
knowledge

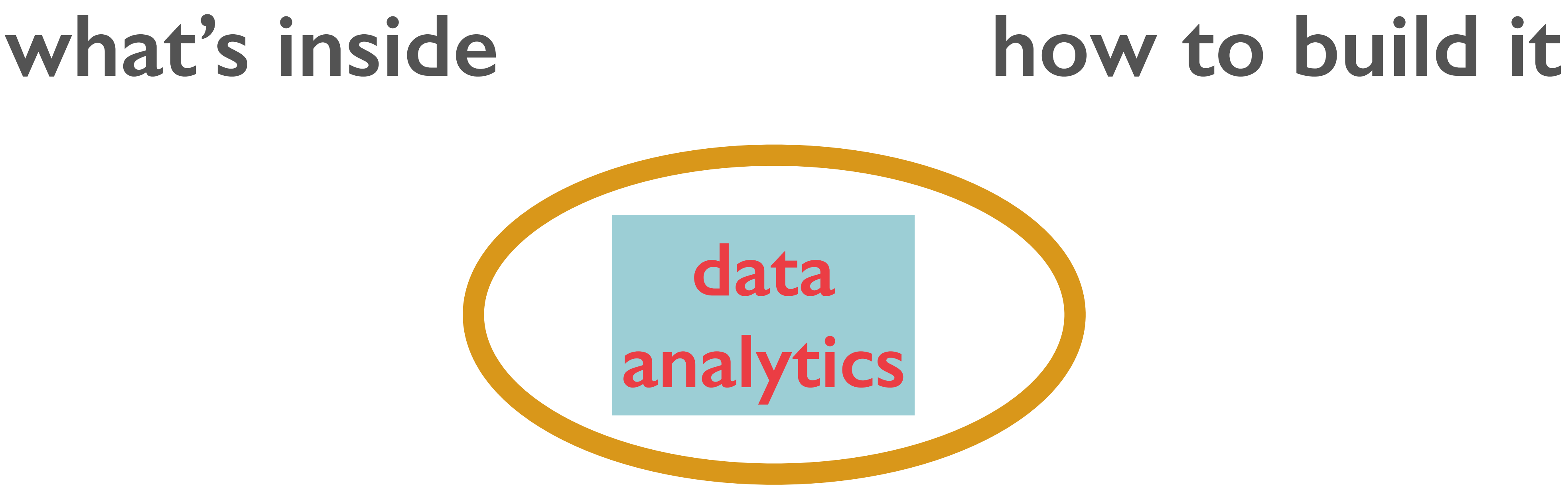
THE SUBJECT OF THIS COURSE

- We will mostly concentrate on the data analytics box, and explain what it contains and how to build it


visualization for annotation
data analytics
data management
visualization for decision making
data
knowledge

- But first we will explain what is possible with predictive analytics: the main focus is about the organizational issues of building and running the pipeline, and the business implications of these issues

The subject of our six-day segment



what's inside

how to build it

data analytics

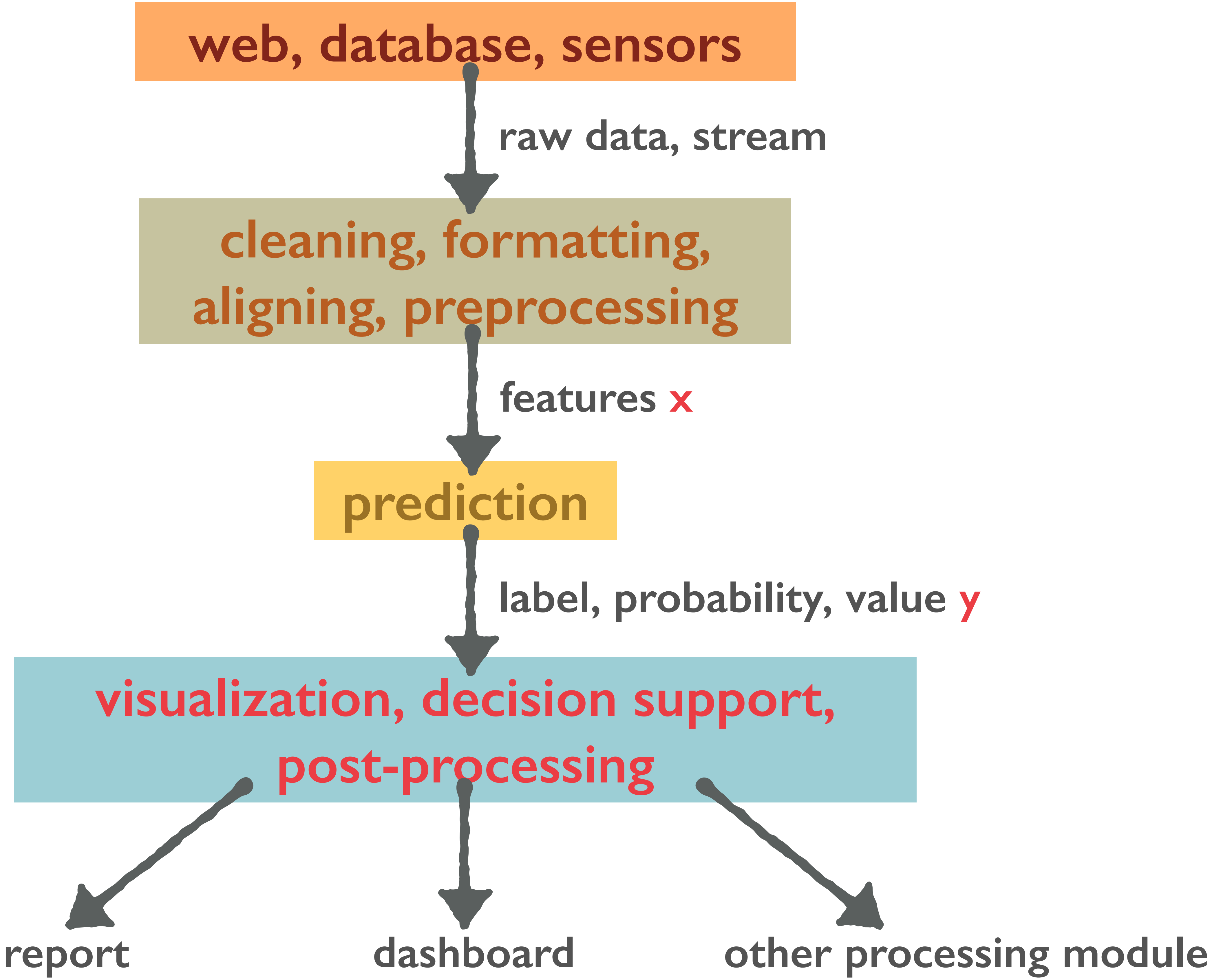
- Introduction (1 day)
- Methods (3 days)
- DataCamp, format RAMP http://www.ramp.studio (2 days)

MODULE 1: ECOSYSTEM AND PIPELINES (4.5 H)

- The **data science ecosystem**
  - actors, roles, building it
- The data science ***production* pipeline**
  - the **steps the data goes through** after the pipeline has been built, tuned, and put into production
- The data science ***development* pipeline (loop)**
  - the **steps the data scientist follow** to build the pipeline

The data science ecosystem (2h)

The data analytics production pipeline



web, database, sensors

raw data, stream

cleaning, formatting, aligning, preprocessing

features x

prediction

label, probability, value y

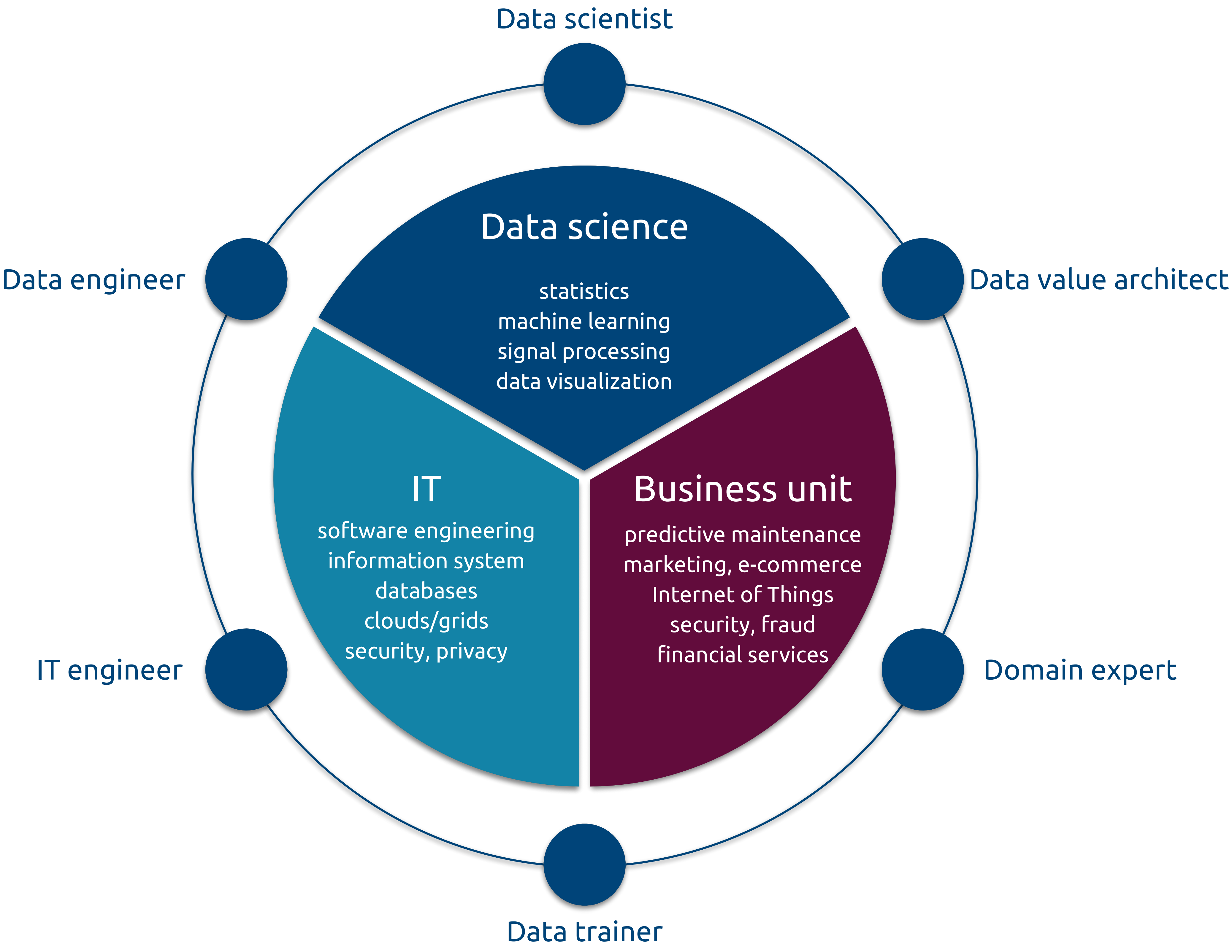
visualization, decision support, post-processing

report

dashboard

other processing module

The data science ecosystem



Data scientist

Data engineer

Data value architect

IT engineer

Domain expert

Data trainer

Data science

statistics
machine learning
signal processing
data visualization

IT

software engineering
information system
databases
clouds/grids
security, privacy

Business unit

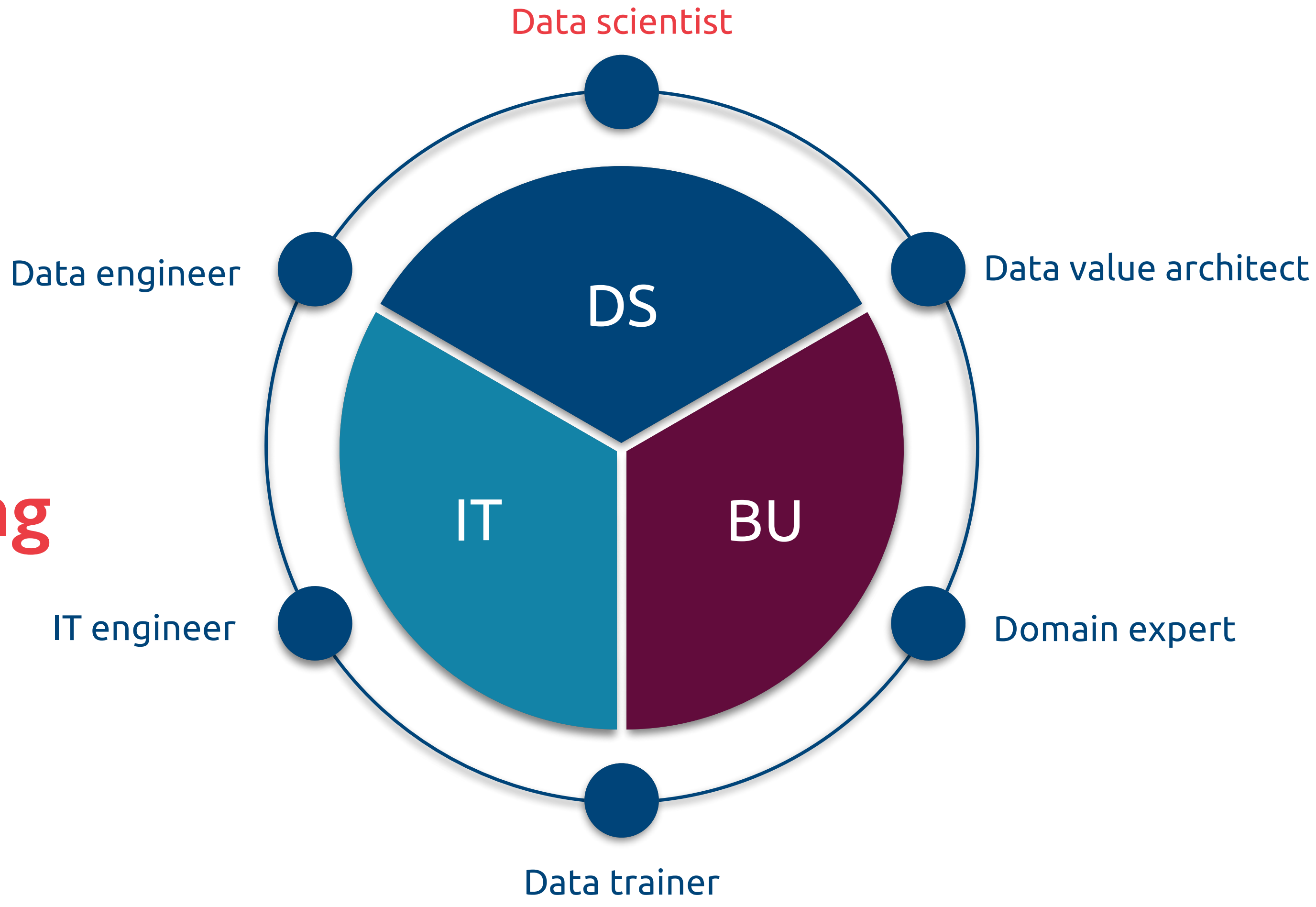
predictive maintenance
marketing, e-commerce
Internet of Things
security, fraud
financial services

DATA SCIENTIST THE *KAGGLER*

- Technical expert in machine learning, statistics, visualization, signal processing

- Efficient in cleaning and munging

- Knows the latest techniques and tools

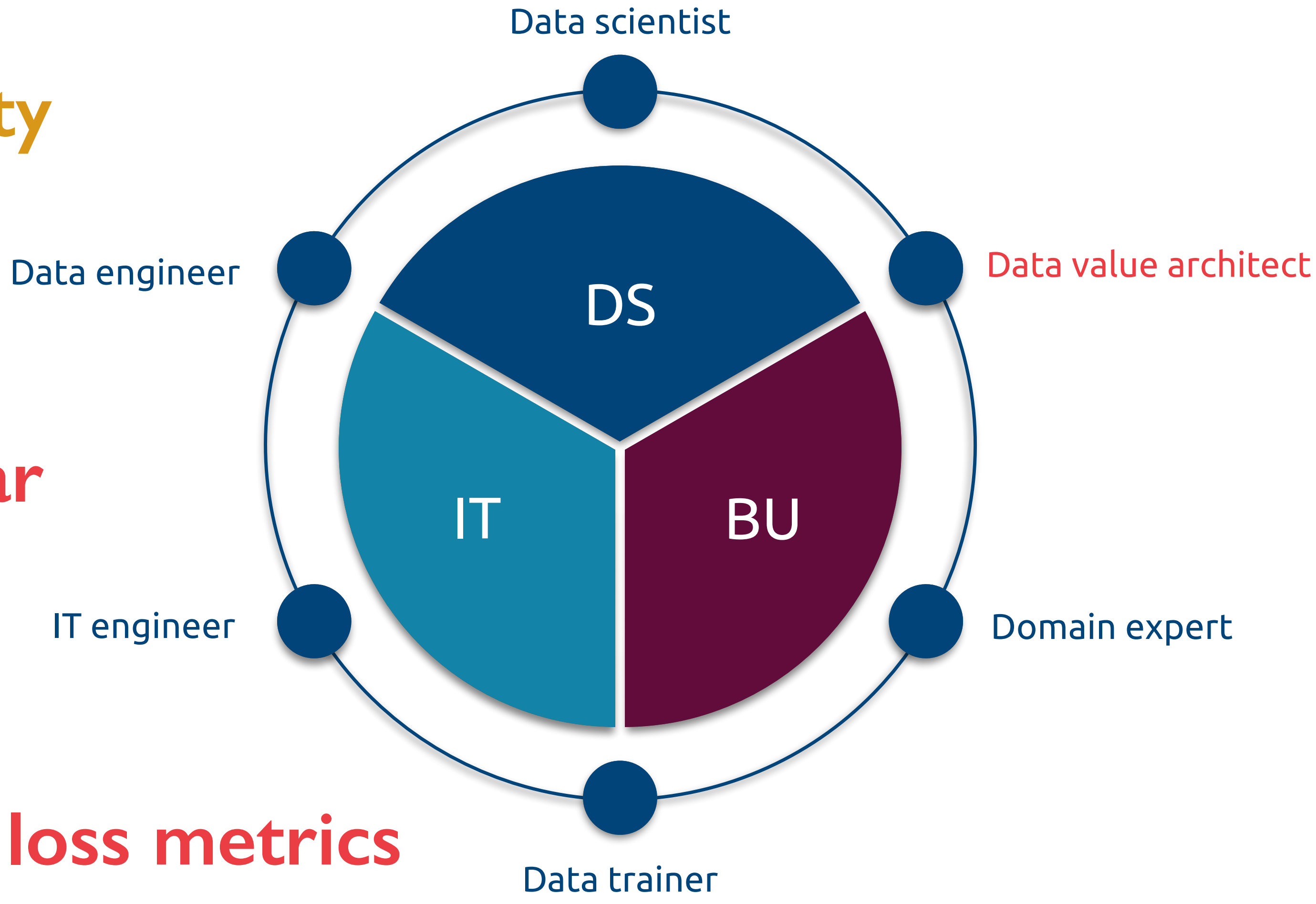


- Can handle different data types and loss metrics

- Can build adequate prototype workflows

- Knows how to tune (optimize) and blend models

DATA VALUE ARCHITECT
THE *FORMALIZER*

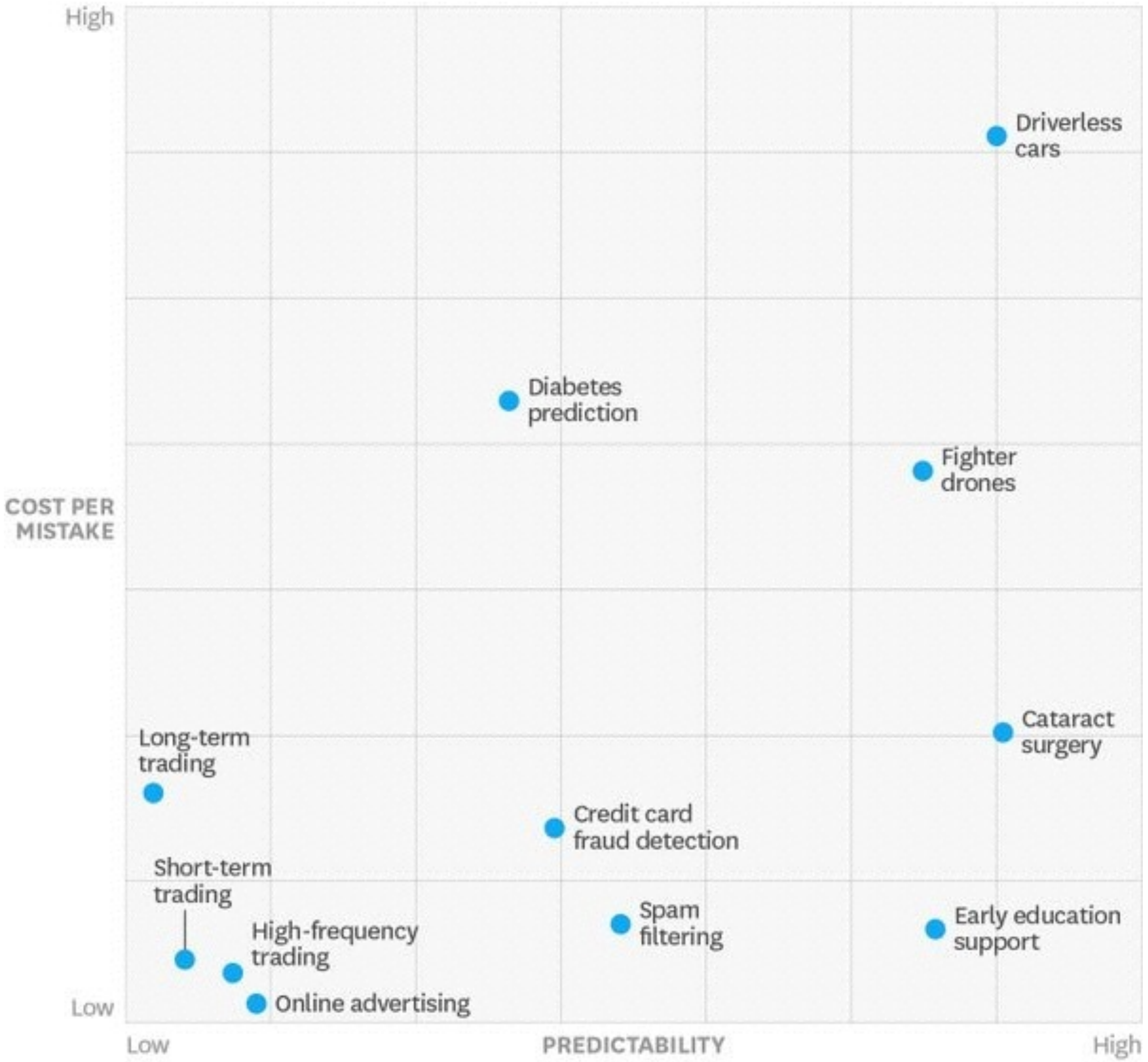


- Has experience with a **wide variety of business problems** and **technical solutions**
- Is possibly **expert in the particular domain**, or at least can **converse** with the domain expert
- Can translate **business goals into loss metrics**
- Can **formalize** adequate **prototype workflows**
- Can **estimate the costs** of building and running workflows
- Can define and dimension the **data collection** effort

DATA VALUE ARCHITECT MINDSET

The Decision Automation Map

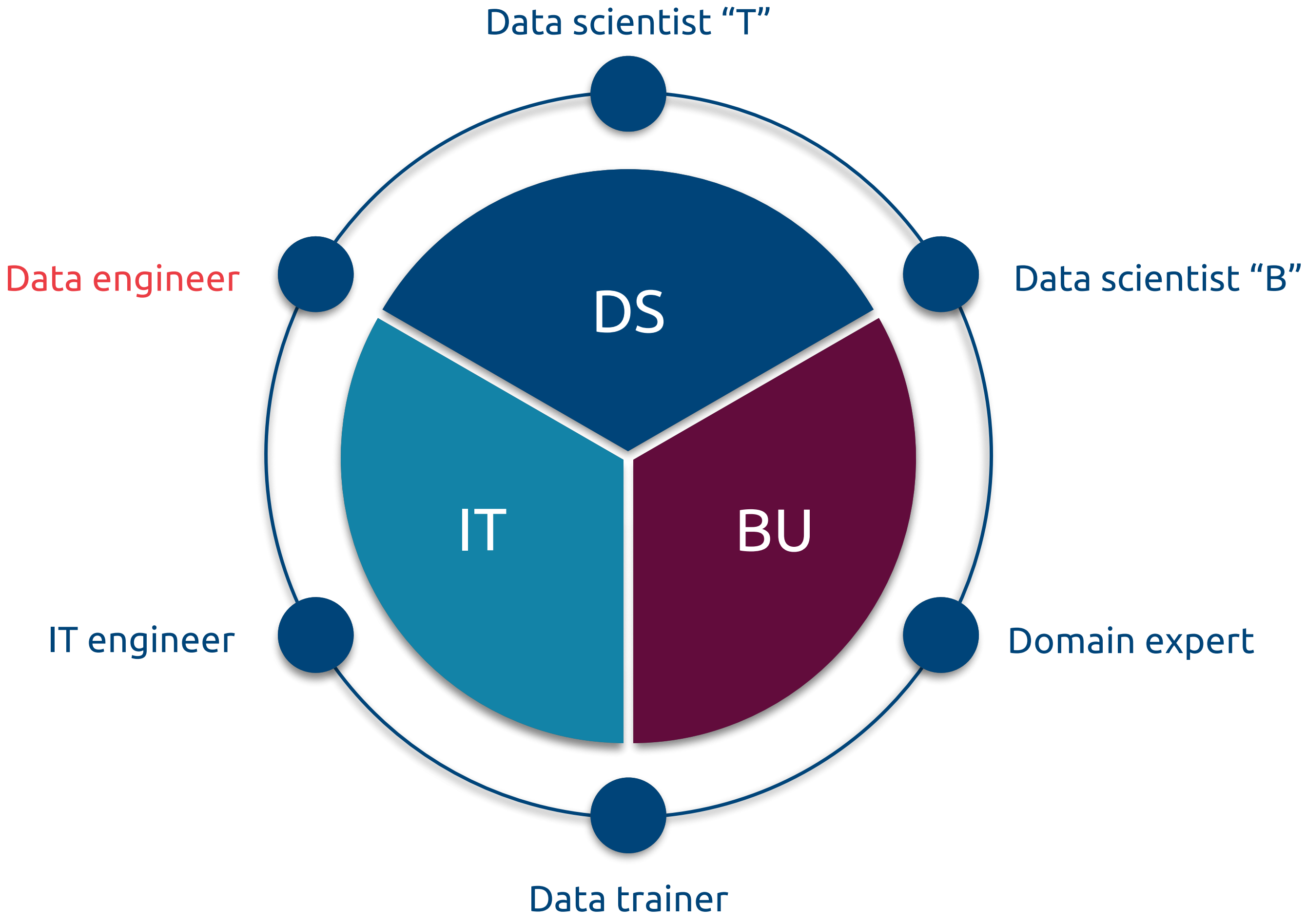
Plotting how well machines can do at making predictions against the costs of mistakes.



SOURCE VASANT DHAR

© HBR.ORG

Data Engineer



- Translates prototypes into **production workflows**, runs and maintains them

- Knows the latest **data engineering systems** and **architectures**

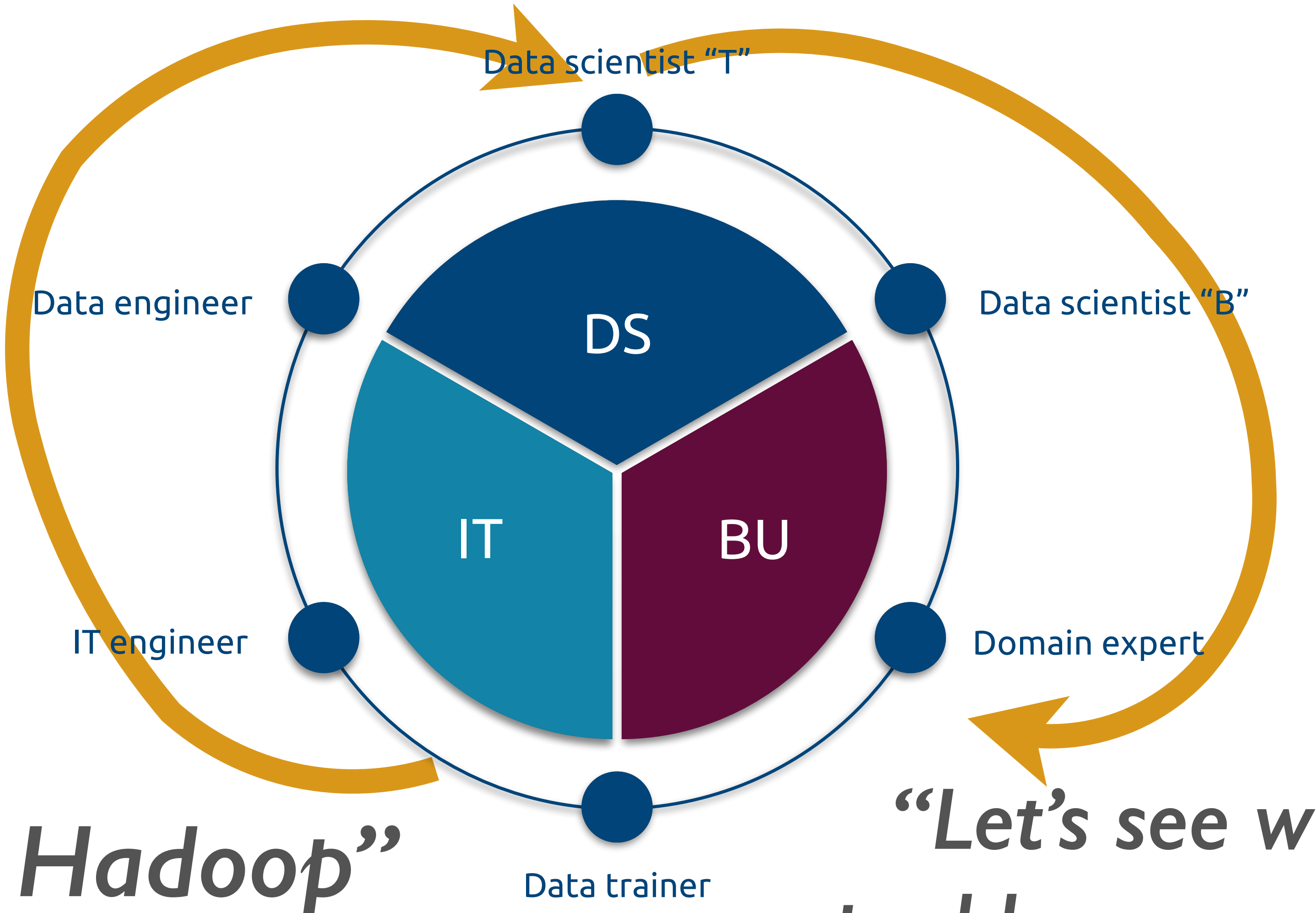
- Knows the **existing IT**

- Can **dimension the production workflows** and **estimate their costs**

- Knows the **basics of building** a data science workflow, and can feed the process by **extracting** and possibly **cleaning/munging** adequate data

BUILDING A DATA SCIENCE ECOSYSTEM DRIVEN BY IT

"Let's hire data scientists"

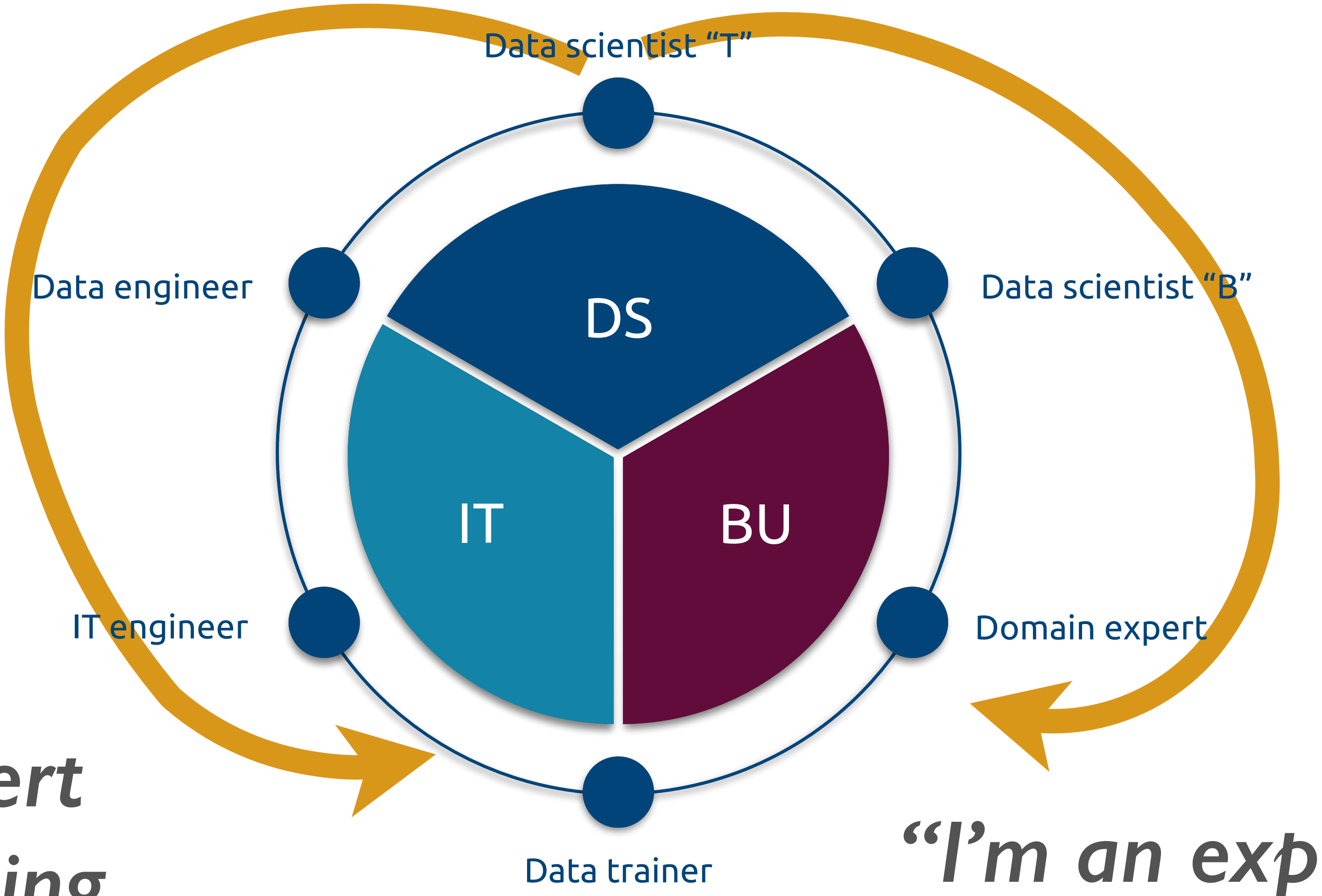


"Let's install Hadoop"

"Let's see what business problems we can solve with the existing data science team and the infrastructure we bought"

BUILDING A DATA SCIENCE ECOSYSTEM DRIVEN BY DATA SCIENTISTS

“Let’s hire data scientists.”

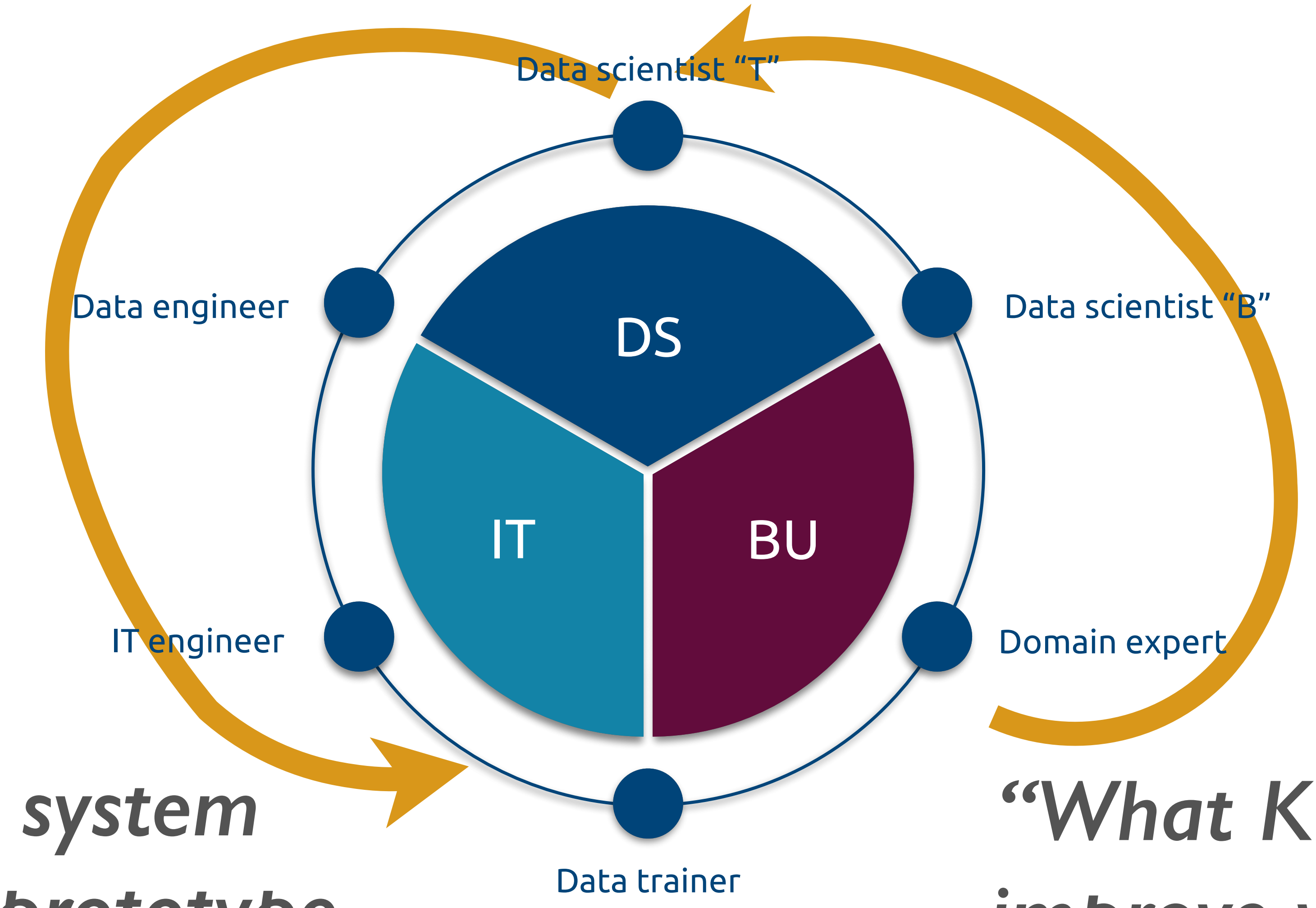


“I’m an expert of deep learning, let’s by a GPU cluster.”

“I’m an expert of deep learning, let’s see what it can do for your business.”

BUILDING A DATA SCIENCE ECOSYSTEM DRIVEN BY BUSINESS

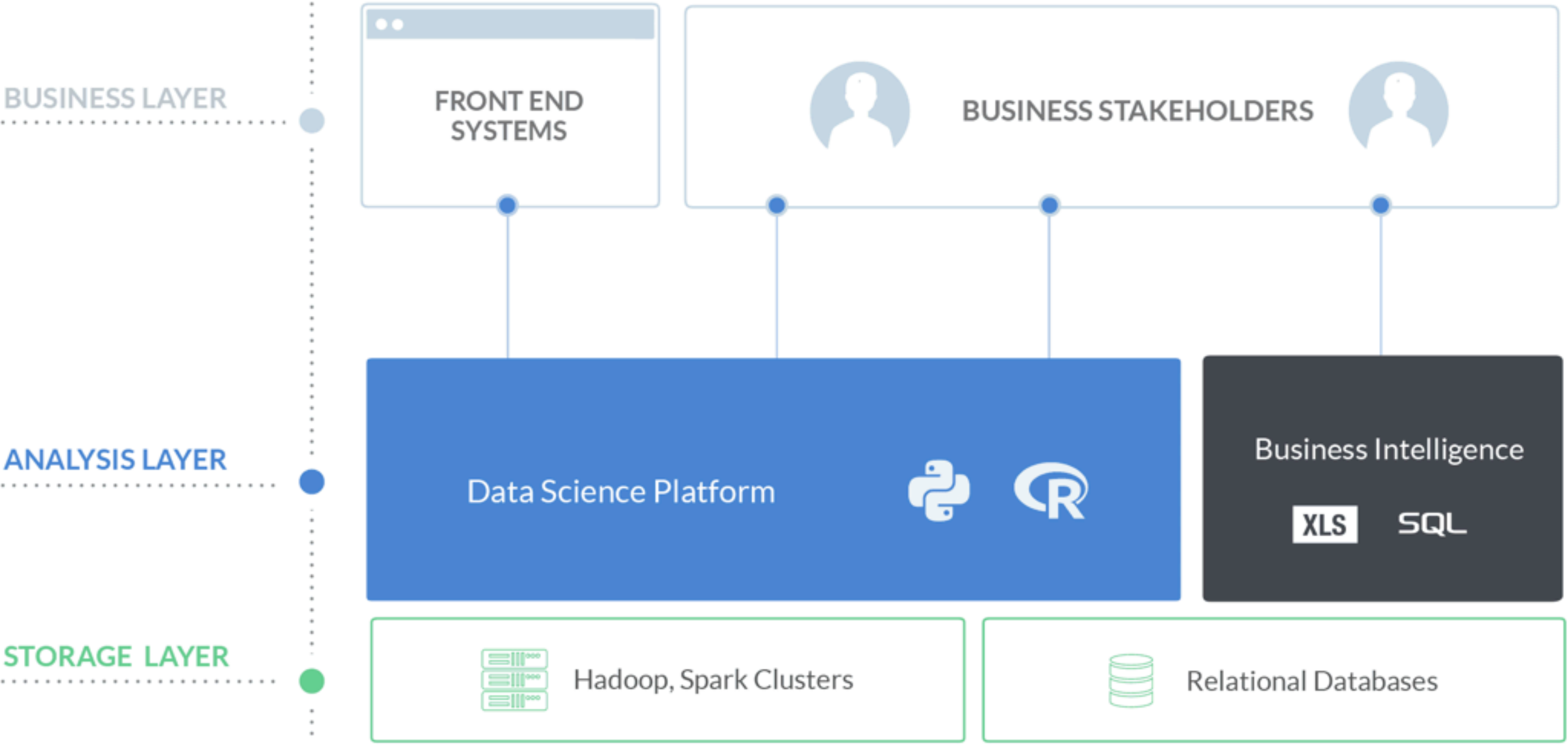
"Let's hire data scientists for prototyping the business case."



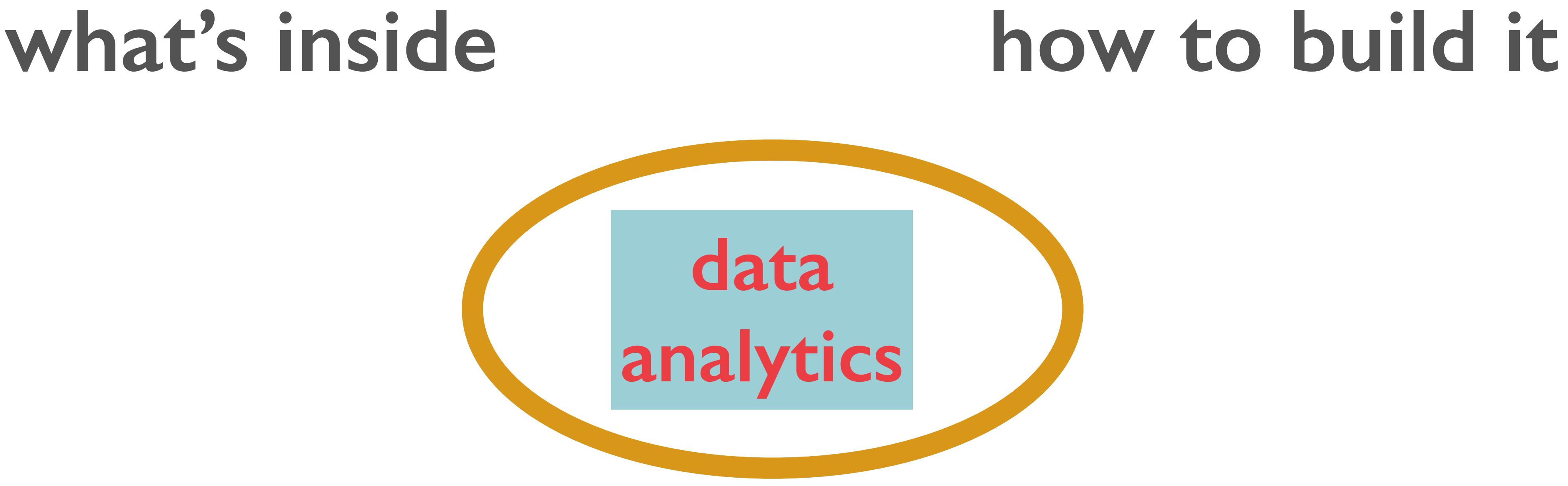
"Let's build a system for putting the prototype into production."

"What KPI can we improve with data? What data should we collect?"

THE PLACE OF A DATA SCIENCE PLATFORM



The subject of our six-day segment



what's inside

how to build it

data analytics

- Introduction (1 day)
- Methods (3 days)
- DataCamp, format RAMP http://www.ramp.studio (2 days)

A case study
(1h)

CASE STUDIES

- Real-time online ad placement

APLA is a web advertising broker company. It contracts with companies who seek placing online ads on web pages. It participates in online bidding (second-price auctions) and it pays for placing the ad. It is then getting payed if the user looking at the ad clicks on the ad.

- Predicting electrical consumption with smart meters

SmartEnergy is an electricity distribution company. It equipped some of the households with smart meters that can measure instantaneous consumption at every 5 seconds. SmartEnergy is asking itself what services it could develop with this data.

- Internal purchasing fraud detection

ANM is a large company with ~1.5M yearly purchases in its books. An estimated 0.1% of the purchases are frauds. ANM would like to develop an automated alarming system to detect fraud.

- Predictive maintenance

RBTF is a large industrial company with multiple sites. It has ~100K complex pieces of equipment that need regular maintenance and replacement. RBTF would like to develop a system to predict the wear and the lifetime of its equipment to save maintenance costs.

- Questions to ask

Identify the KPI. What should X try to predict? What are the costs of misprediction? What re the benefits of better predicting?
Do we want to fully automate, use predictions for decision support, or just visualize to better understand the data?
What data should X collect? What are the costs of data collection? What are the costs of deploying the data science pipeline?

A CASE: CAR RENTAL PREDICTIVE MAINTENANCE

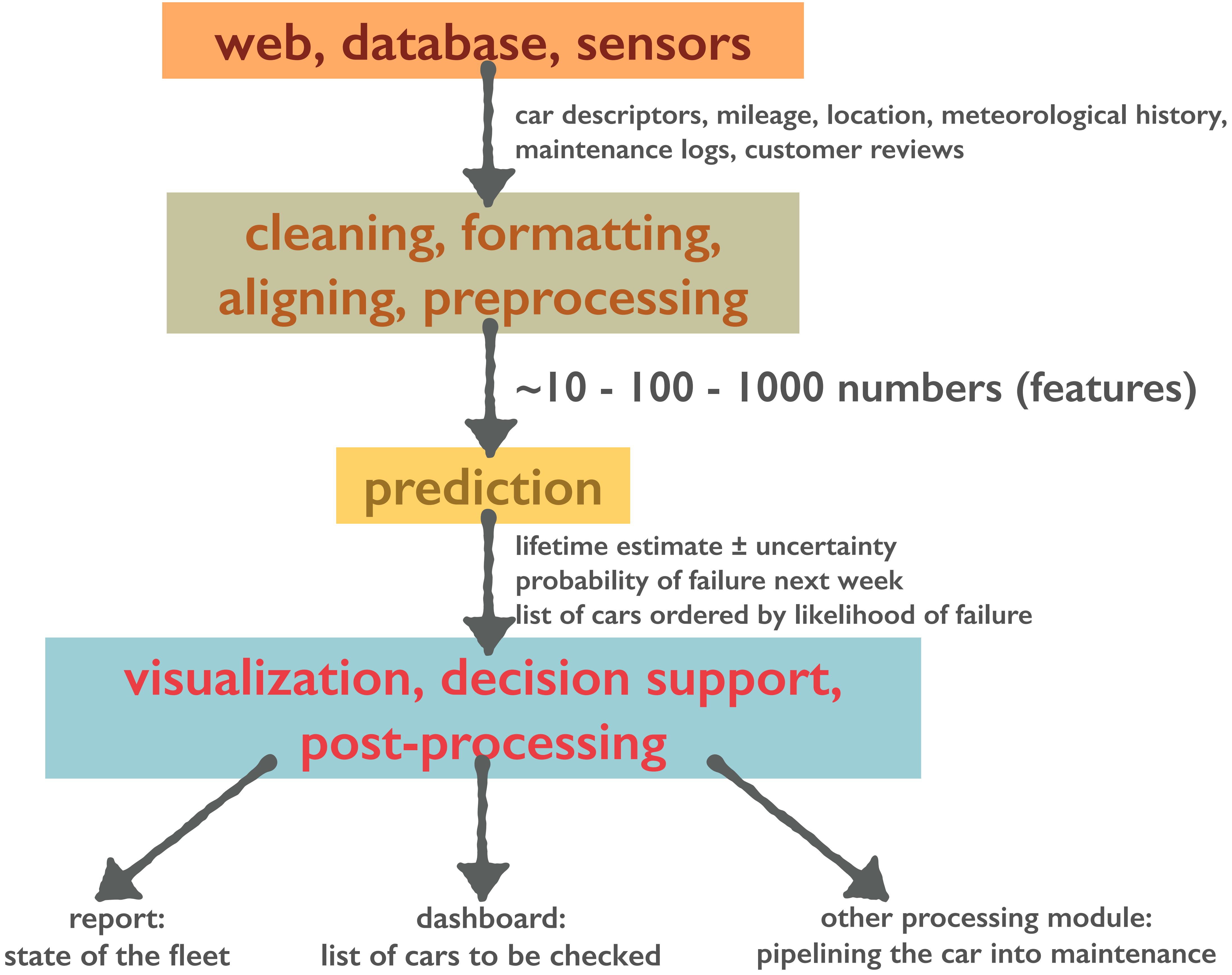
AB-Rent is a multinational car rental company. They have a fleet of 500K vehicles, ranging from sub-economy cars to 12 foot trucks. Annual sales are about 6B$ per year with a profit of 100M$. AB-Rent follows an extensive maintenance schedule to avoid the rental of defective cars. There are check-ups (pre-determined by the car manufacturer) scheduled by mileage and age of the car. After every rental, mechanics visually inspect and drive the cars to check for defects. A defective car, if it were rented, would generate both direct costs (replacement, towing, refunds) and indirect costs (reputation, customer churn). Maintenance costs total about 10% of annual sales (~600M$).

To decrease maintenance and defect costs, AB-Rent decided to launch a Big Data project. The goal is to predict more accurately when a car needs a check-up and repair.

1. What do we want to predict and how do we measure the quality of the prediction? How will a better prediction improve the bottomline?
2. Do you want to have decision support, a fully automated system, or to know just the factors which are important? How will the agent use the system? Why are these important questions to ask?
3. What should be the quantitative prediction? How do we measure success?
4. What data do we need to develop a predictor?

The data analytics production pipeline and development loop (1h30)

The data analytics production pipeline



web, database, sensors

car descriptors, mileage, location, meteorological history, maintenance logs, customer reviews

cleaning, formatting, aligning, preprocessing

~10 - 100 - 1000 numbers (features)

prediction

lifetime estimate ± uncertainty

probability of failure next week

list of cars ordered by likelihood of failure

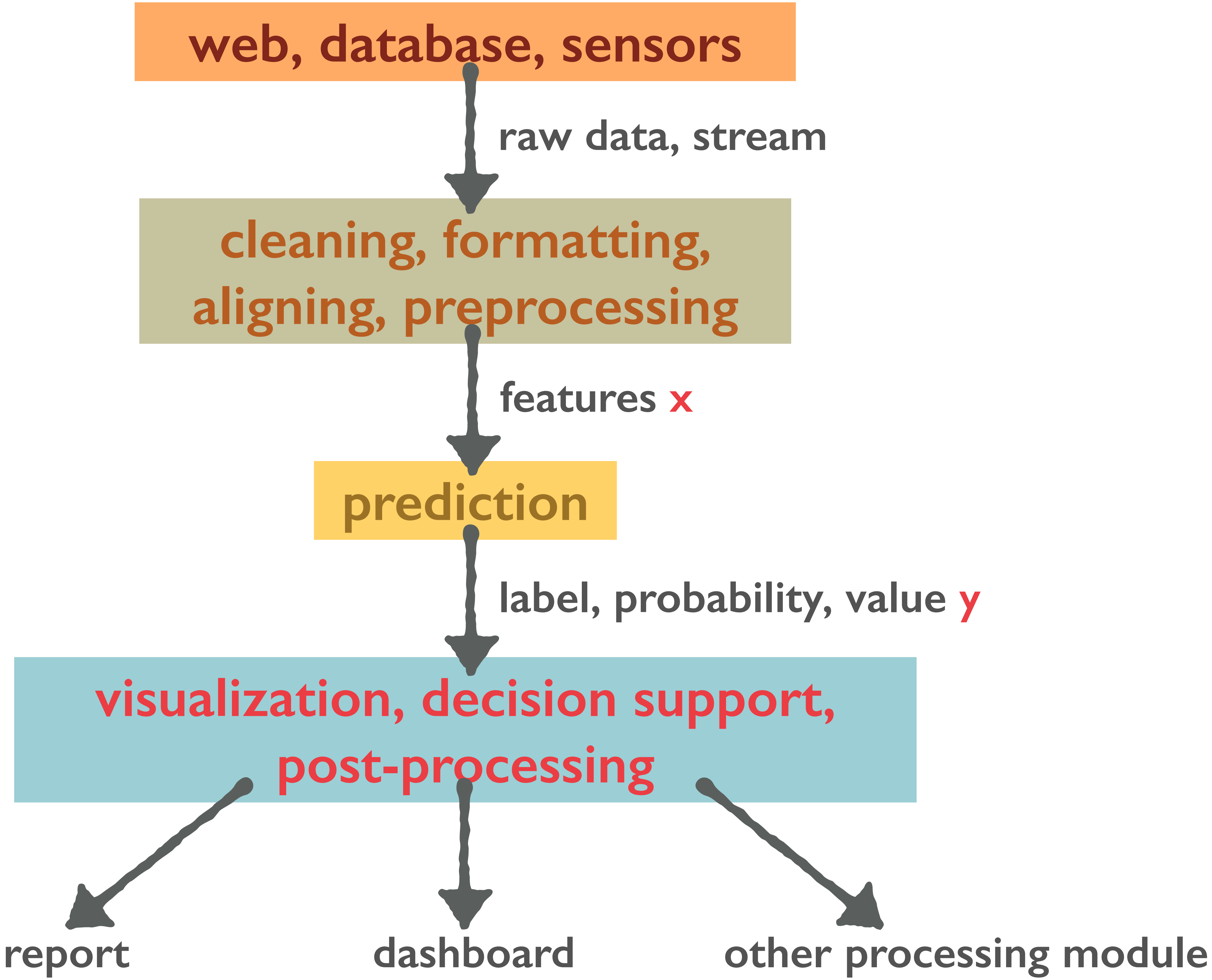
visualization, decision support, post-processing

report: state of the fleet

dashboard: list of cars to be checked

other processing module: pipelining the car into maintenance

The data analytics production pipeline



web, database, sensors

raw data, stream

cleaning, formatting, aligning, preprocessing

features x

prediction

label, probability, value y

visualization, decision support, post-processing

report

dashboard

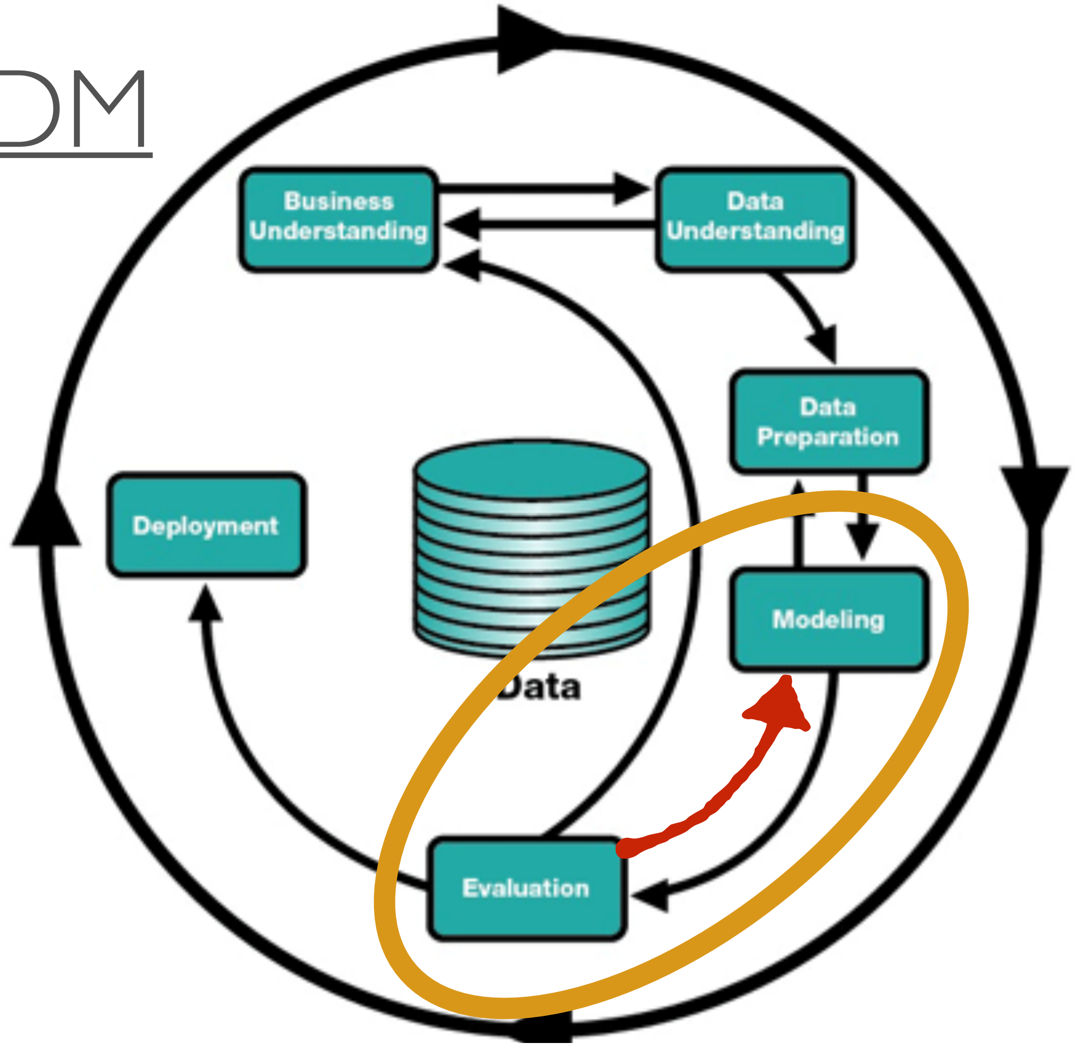
other processing module

THE DATA ANALYTICS BUILDING PIPELINE

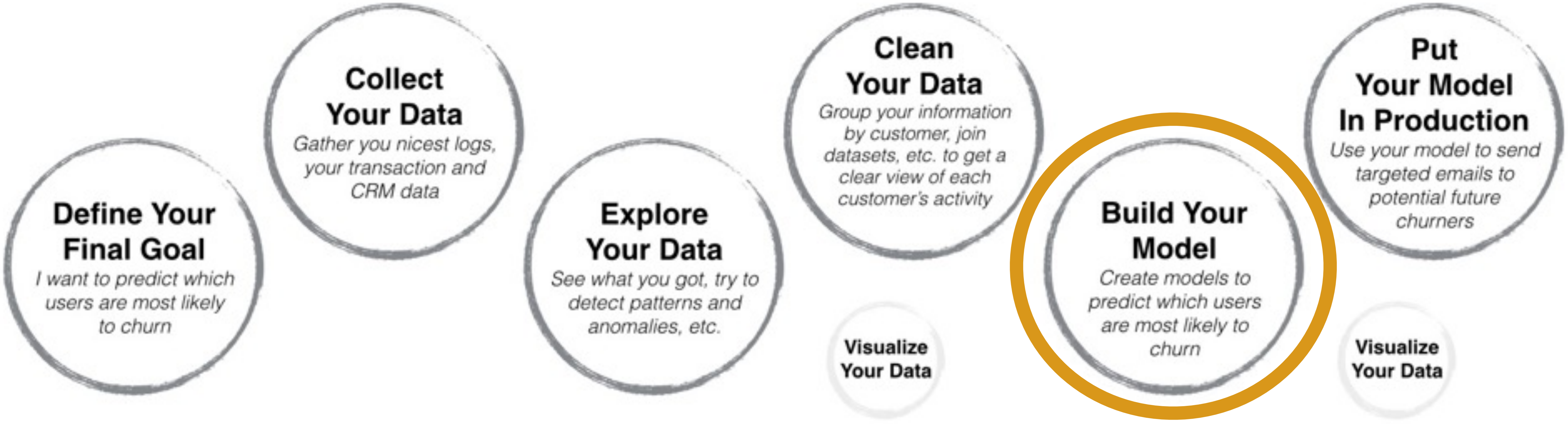
IBM CRISP-DM

1996

2016



Dataiku



Define Your Final Goal
I want to predict which users are most likely to churn

Collect Your Data
Gather you nicest logs, your transaction and CRM data

Explore Your Data
See what you got, try to detect patterns and anomalies, etc.

Clean Your Data
Group your information by customer, join datasets, etc. to get a clear view of each customer's activity

Visualize Your Data

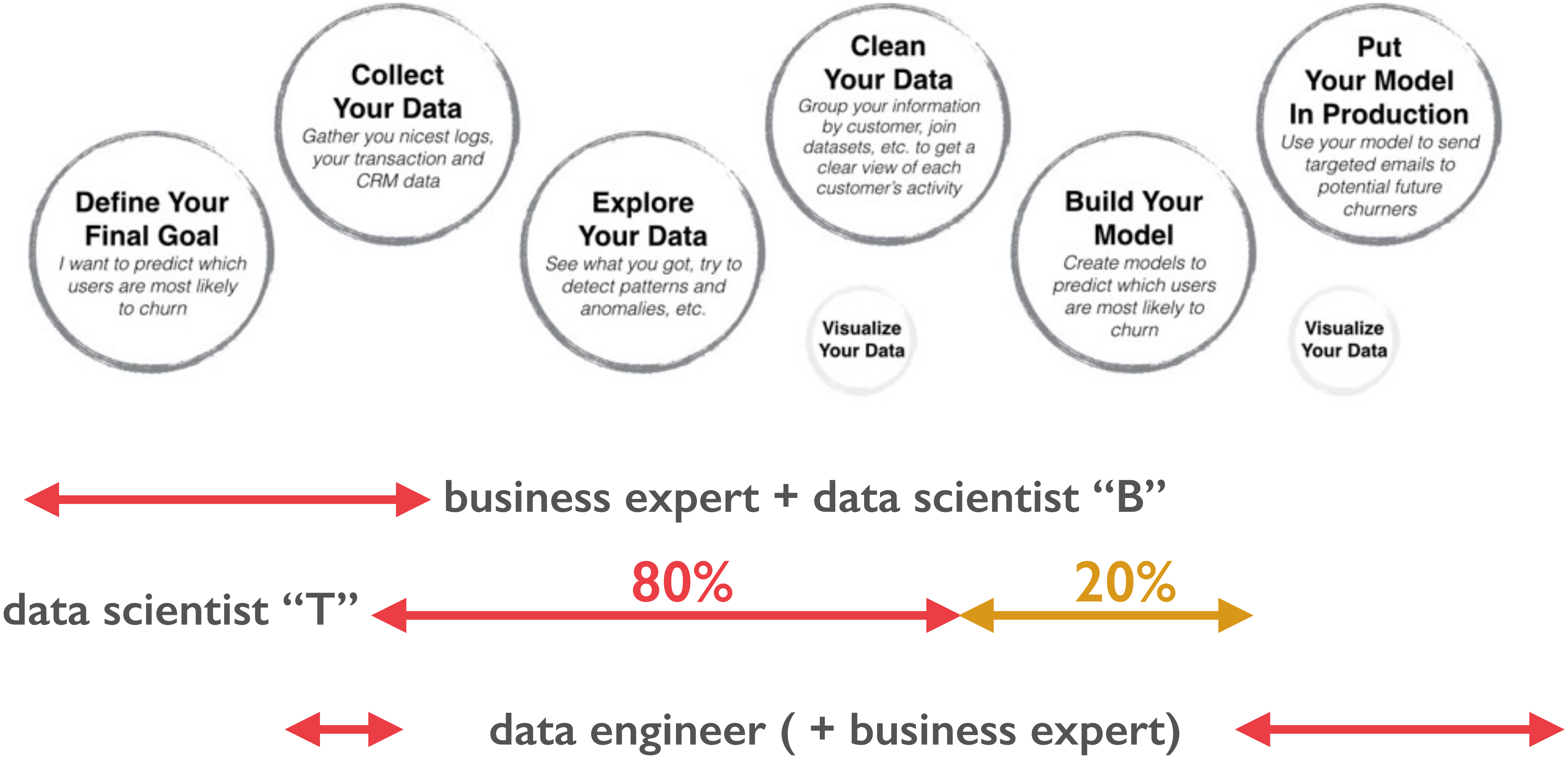
Build Your Model
Create models to predict which users are most likely to churn

Visualize Your Data

Put Your Model In Production
Use your model to send targeted emails to potential future churners

THE DATA ANALYTICS BUILDING PIPELINE

WHO DOES WHAT AND WHEN



Define Your Final Goal
I want to predict which users are most likely to churn

Collect Your Data
Gather you nicest logs, your transaction and CRM data

Explore Your Data
See what you got, try to detect patterns and anomalies, etc.

Clean Your Data
Group your information by customer, join datasets, etc. to get a clear view of each customer's activity

Visualize Your Data

Build Your Model
Create models to predict which users are most likely to churn

Put Your Model In Production
Use your model to send targeted emails to potential future churners

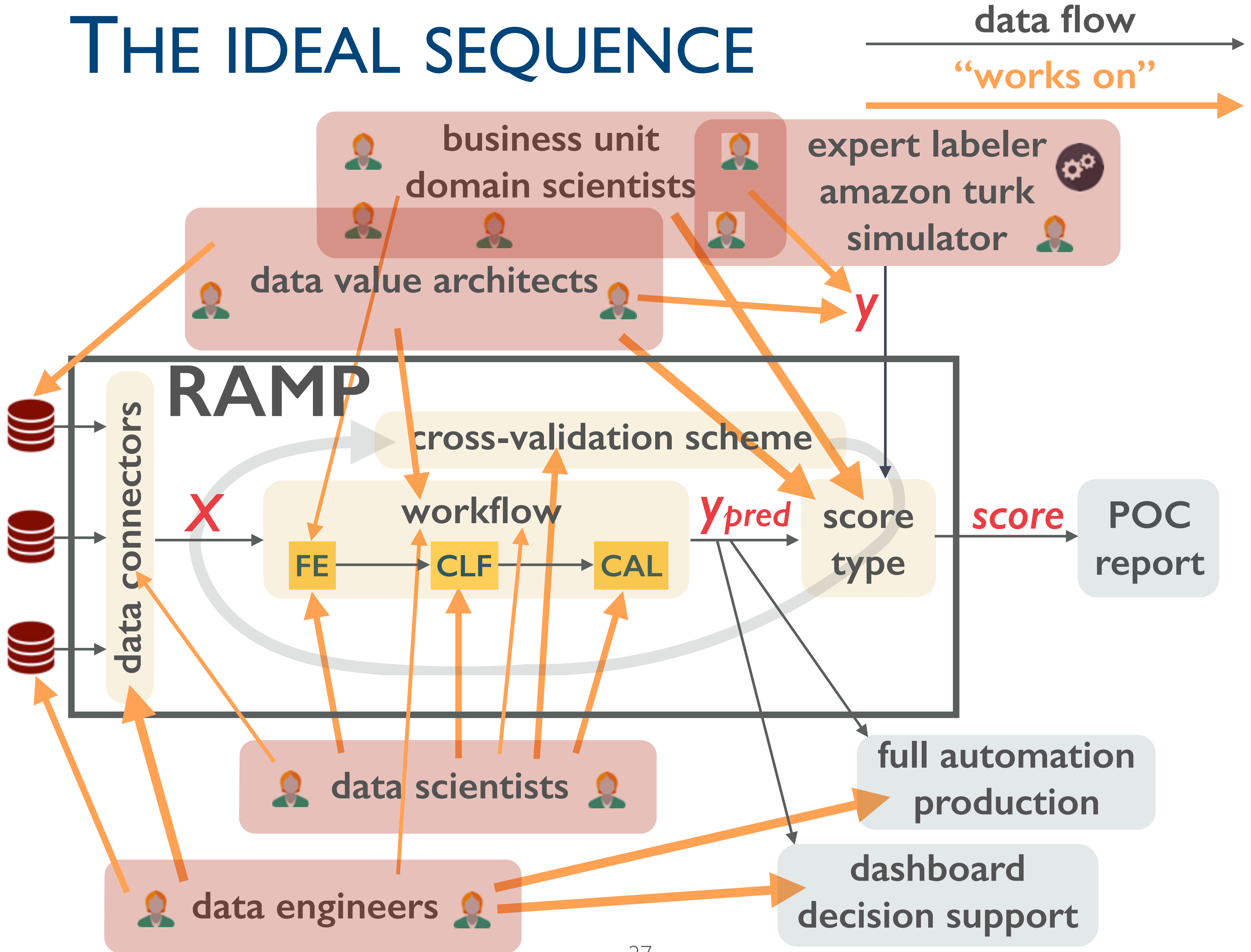
Visualize Your Data

business expert + data scientist "B"

data scientist "T" 80% 20%

data engineer (+ business expert)

The ideal sequence



THE DATA ANALYTICS BUILDING PIPELINE

- It is **trial and error**
  - little if any theory-based, model-based design
  - even research (development of new algorithms) is (mostly) trial and error
  - the data scientist's best friend is a **well-designed experimental studio** for facilitating fast iterations of
    - **what data to use**: car descriptors, mileage, location, meteorological history, maintenance logs, customer reviews
    - **what features to select or engineer**: which descriptors to feed into the predictor, how to digest meteorological history into a small set of numbers, predictive about car failure
    - **what predictors to use**: linear regression (classical statistics), random forests (scikit learn), neural networks (deep learning)
    - **how to parametrize the predictors**: how many trees in the forests, how many neurons

THE DATA ANALYTICS BUILDING PIPELINE

- Data-driven predictors should **work well on future (unseen) data**. This simple fact drives everything, the mindset and the design.
  - this is the **same paradigm as in classical statistics**
  - use **historical data to select and fit a model,** then use the model to **make predictions on new data**
  - but **we only have historical data**: we need to **"simulate" past and future** on existing data
  - the **train-test loop**
    - use (eg) 80% of the data for selecting the features, selecting the models, optimizing the models
    - use 20% of the data to test the model, to measure the predictive quality
  - we will need a **quantitative measure of quality/performance** because the data scientist will have to **test a lot of different choices**
    - qualitative, human-in-the-loop measures slow down the development loop
    - designing the quantitative measure is crucial: it should be tightly coupled to the KPI that we want to improve

STATISTICS VS MACHINE LEARNING

- Similar techniques and principles but different mindset (even epistemology)
  - depending on the business case, you may need to whiten the black box: decision support vs full automatization, legal issues.
  - trade-off between performance and whiteness
- Tendency
  - ML was always strong in domains where there was no model to start with: speech, vision
  - it slowly took over model-driven statistics in domains where models are dubious and complex (natural language, biology)
  - it is sweeping over all science and industry, even in domains with strong models (physics, material science)

statistics	ML
simple parametrized models	black box models
top-down, theory-driven	bottom-up, data-driven
needs analytical knowledge about the world	needs a lot of data

READING MATERIAL ON MEDIUM.COM/@BALAZSKEGL

- The data science ecosystem (industrial edition)
- Teaching the data science process
- How to build a data science pipeline

The subject of our six-day segment

what’s inside

how to build it

data analytics

- Introduction (1 day)
- Methods (3 days)
- DataCamp, format RAMP http://www.ramp.studio (2 days)

MACHINE LEARNING AKA PREDICTIVE ANALYTICS

- The basic setup
- Data types
- Problem types
- Loss and error metrics
- The train/test paradigm
- Overfitting

THE BASIC SETUP

- Objects/instances are represented by a **vector of features** x
  - car descriptors, mileage, location, meteorological history, maintenance logs, customer reviews, digested into a finite number of numbers
- **Predictions** (usually a scalar) are represented by y
  - lifetime, probability of failure, rank order (sorted by probability)
- **Prediction** or **inference** problem: we are looking for the function f: $y = f(x)$

DATA TYPES AND FEATURE ENGINEERING

- How to obtain a vector of features *x* from a
  - database row? (easy): car descriptors, mileage, location
  - time series history or signal? (moderately easy): meteorological history, GPS history
  - text? (moderately easy to difficult but doable): maintenance logs, customer reviews
  - photo? (difficult but doable)
  - chess/go table? (state of the art)
  - video? (we're getting there but for now it's research): on-board camera
- Feature engineering is a difficult problem, the design is usually part of the data scientist's trial-and-error process
- The deep learning revolution, to a large extent, is about automating this step

Two basic problem types

- Regression: y is a real number
- Classification: y is coming from a finite set of labels/ classes/categories
  - important detail: often the direct output of f is a probability vector $(p_1, \ldots, p_k)$, where p_i = Prob{predicted label = label$_i$}
  - then the predicted label index is argmax$_i$ $(p_1, \ldots, p_k)$
- Ranking: y is an index in an ordered list
  - "Car number i fail before car number j"

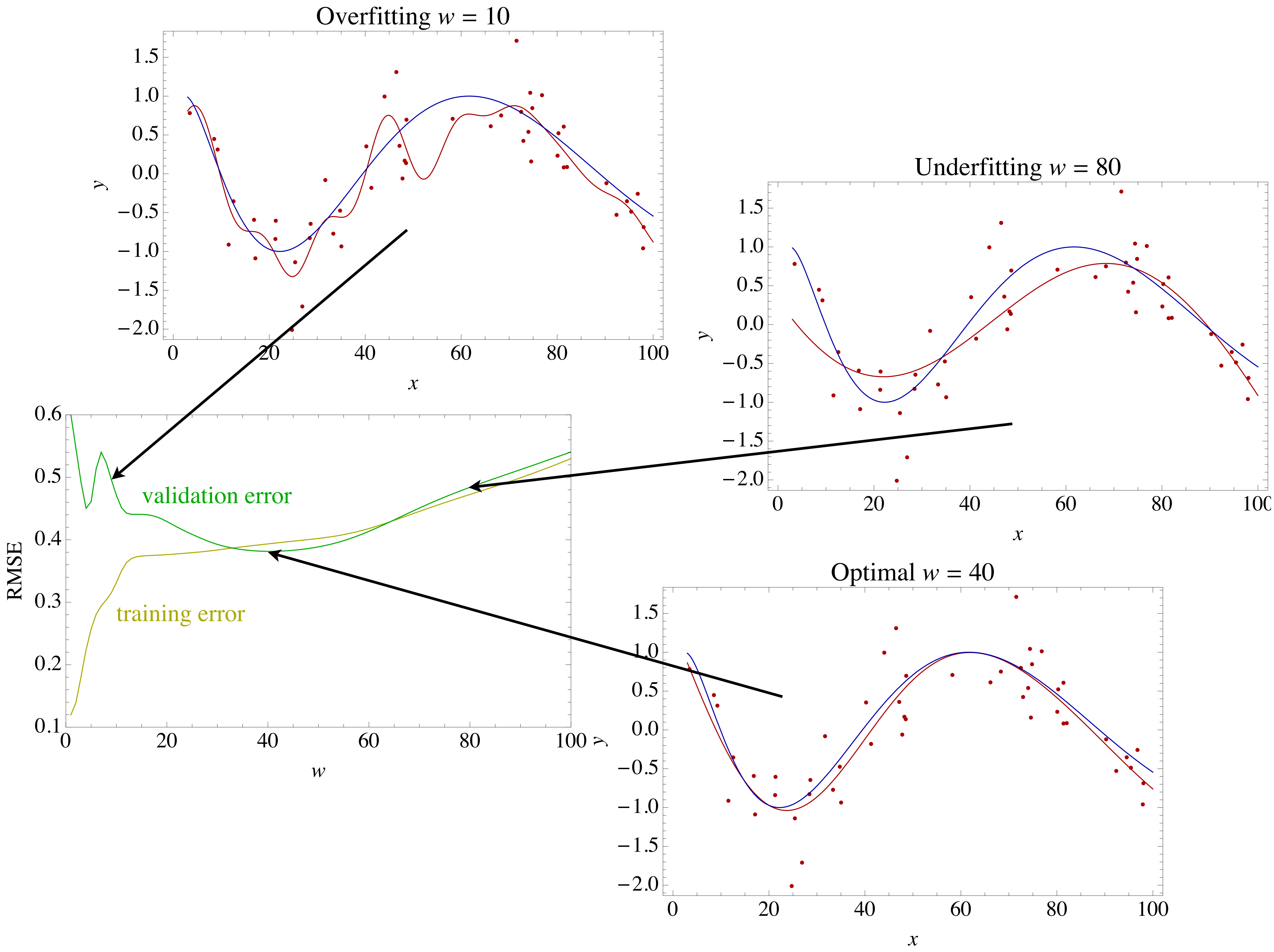
HOW TO QUANTIFY MISPREDICTION

- Regression
  - mean squared error (MSE):
    $(\text{real lifetime - predicted lifetime})^2$
  - mean absolute relative error (MARE), useful when target varies over orders of magnitude
    |real lifetime - predicted lifetime| / real lifetime
- Classification
  - probability of error: the probability that a car fails when we predict no failure, or vice verse
- Ranking
  - the number of mis-ordered pairs
  - normalized discounted cumulative gain
    - The web search industry went a long way to design metrics that captured the perception of search result quality by users. E.g. mis-ordering hits on first page (even top of the first page) is way more important than mis-ordering lower order hits. This process was a great illustration of getting precise KPIs into the data science development pipeline (even research).

THE TRAIN/TEST PARADIGM

- The problem: the future is unknown
  - so
    - 1) we need to estimate the error (performance/quality) on a data set, and
    - 2) we need to select and optimize the predictor *f* using a data set
  - train/test cut
    - one part (typically 80%) of the data is used for training, the other part is used for performance evaluation
    - sampling bias and nonstationarity: the data should reflect the future. E.g.: if old items are in the database only if they survived the time the database was set up, the first rule the predictor finds is that older items live longer than newer items. The problem is that the distribution of the historical data is not the same as the distribution of future data.
  - notice the paradox: we're optimizing a function we don't know: the future prediction performance of the predictor trained on the past
  - which is why machine learning is not equivalent to optimization: optimizing the predictor too well on the past leads to "memorization", or "overfitting". Symptom: the prediction performance is great on training data, bad on test data.

Overfitting



Overfitting $w = 10$

Underfitting $w = 80$

Optimal $w = 40$

The train/test paradigm

- The simplest case: linear (or logistic) regression
  - learn/train a linear function $f(x) = \sum_j w_j x_j$
  - training means: find $w_1, \ldots, w_d$ that minimizes the error on the training data
  - the error on the training set is "tainted": since the function was optimized to minimize this error, it is no longer an independent estimate of the performance of f on future points, thus we need an independent test set
  - overfitting occurs when we "overly" optimize f on the training data, so it "memorizes" it
  - symptom: small training error, large test error

The train/test paradigm

- We go all in: **generalized linear functions**:

$$f(x) = \sum_j w_j \, h_j(x)$$

  - $h_j(x)$ can be **application/domain-dependent cues/features**
  - $h_j(x)$ can be "neurons" or trees (making f a neural net or a forest)
  - how to choose the number of cues/neurons/features? **hyperparameter optimization**
  - how to choose among different models? **model selection**
  - same principle as train/test, except that this is a slow loop, and needs a third set (usually called validation set)

The train/test paradigm

- Operational issue: models will have to be retrained if the world changes
  - online advertising: every couple of hours
  - predictive maintenance: every 6 months
  - certain products are even adversarial: your predictions change the world. In highly automated markets AI bots are playing each other.
- Advanced topics / limitations
  - causality vs correlation: machine learning is designed for picking up correlations in data. They are good at it, can discover hidden correlations. If the data sample is biased (see the old item in a young database example), they will zoom into spurious correlations present in your data.
  - A/B testing: designing experiments to discover causal relationships in the world (ultimately: how can I cause sales/profit go up?). They are mostly designed by experts, but data-driven A/B testing is coming.
  - machine learning + operations research = reinforcement learning. Data-driven design of a sequence of actions, for optimizing cumulative reward (profit). Many times the predictor is part of an optimization process. E.g., retail: optimize the scheduling of orders to minimize out of stock events needs a good predictor of sales but it is also part of a larger logistics optimization system. See AlphaGo and R&D at Google DeepMind: they are "playing" with games, but the business goal is to develop fully autonomous intelligent agents that can act adaptively.