ARUNDO

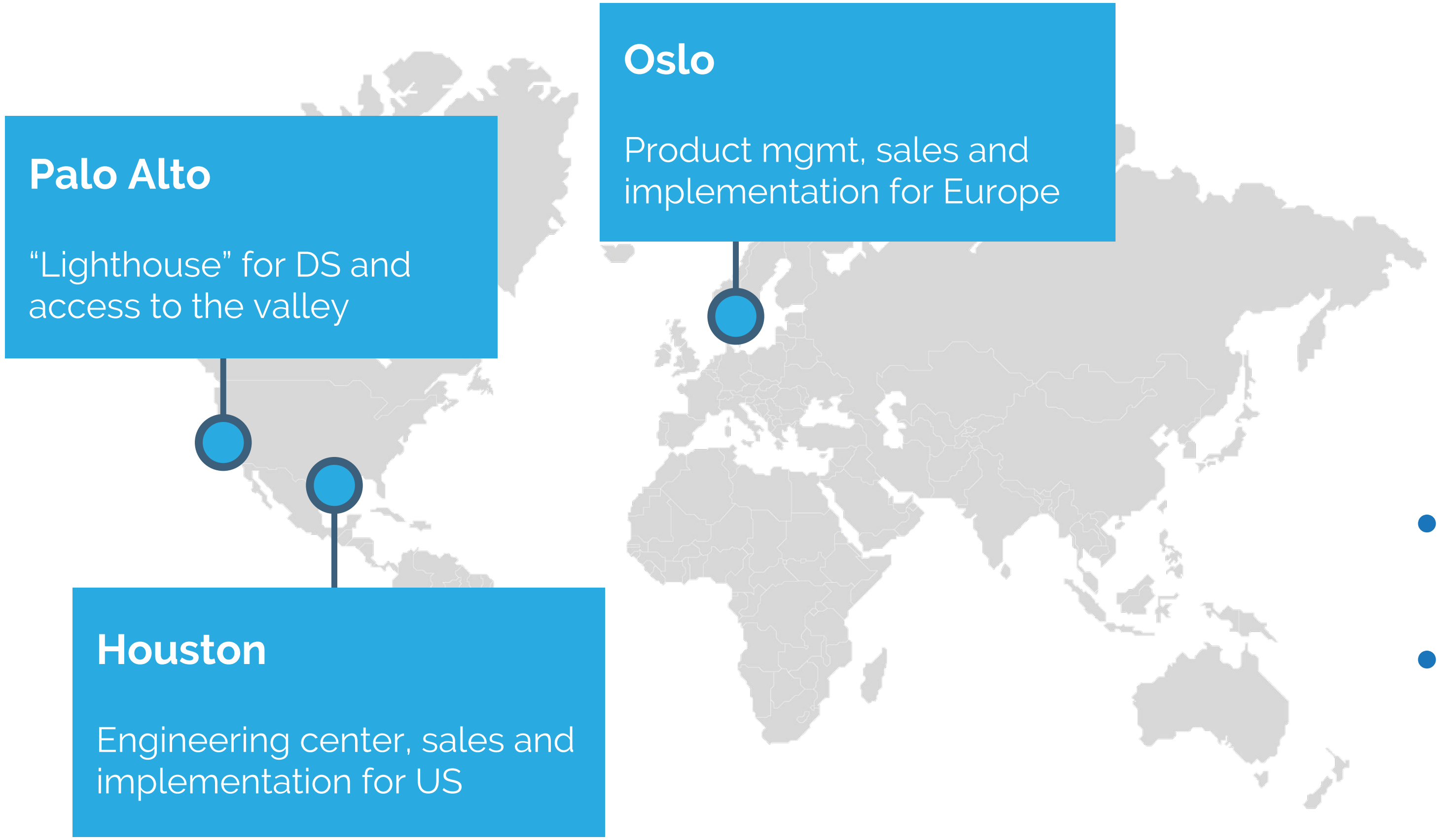
HOUSTON | OSLO | PALO ALTO

Machine Learning with Industrial Data

Mark Tibbetts

Introduction to Arundo & Myself

Arundo: who we are

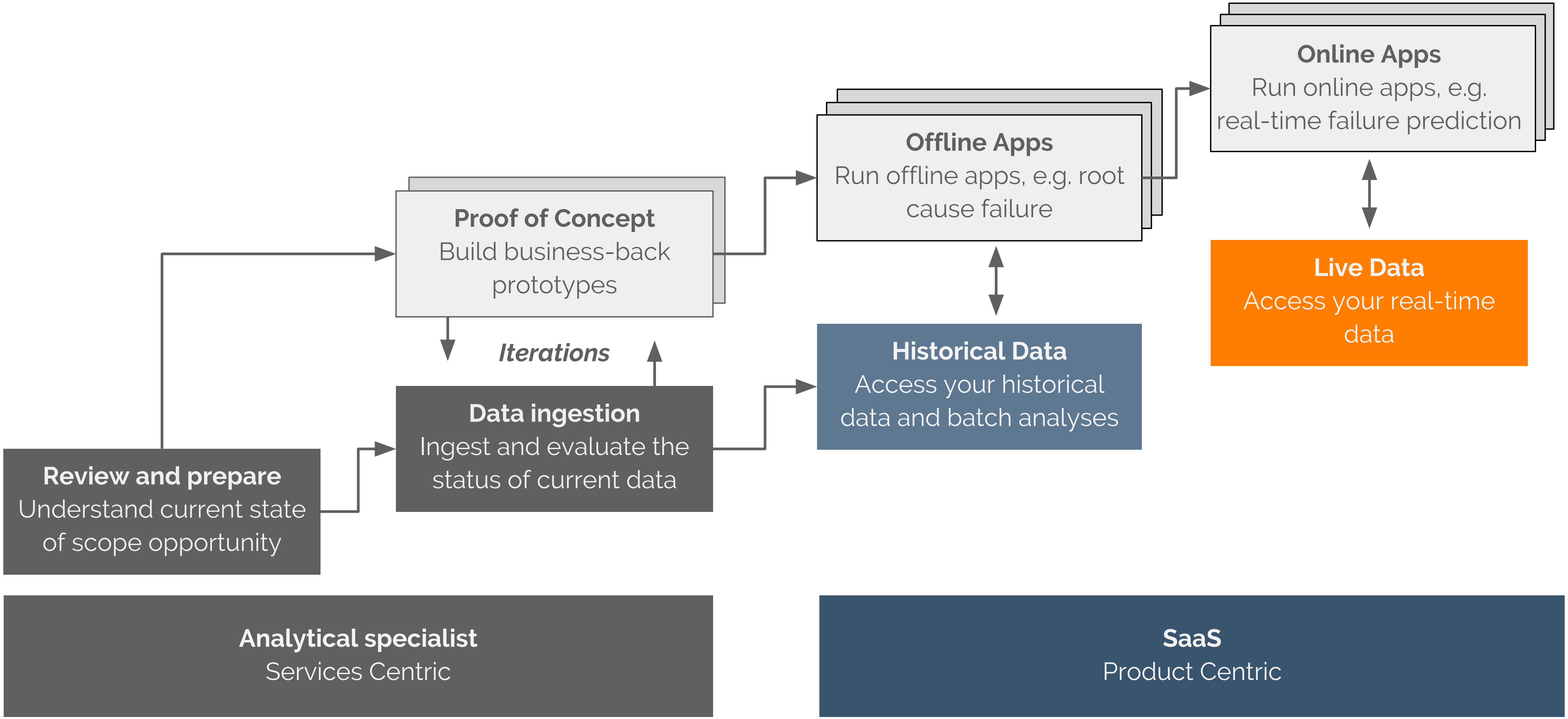


- Bringing "Silicon Valley" into oil & gas, marine and utilities
  - Leveraging MSFT Azure backbone
  - First customers running "microservice" analytics models on base platform
  - Closed second financing round bringing new strategic investors
  - Funding and support from Stanford through StartX accelerator
- Built industrial grade cloud architecture (deployable in private cloud)
- Growing team in all locations, strong support from investors to accelerate

https://www.arundo.com

ARUNDO

Arundo: what we do



ARUNDO

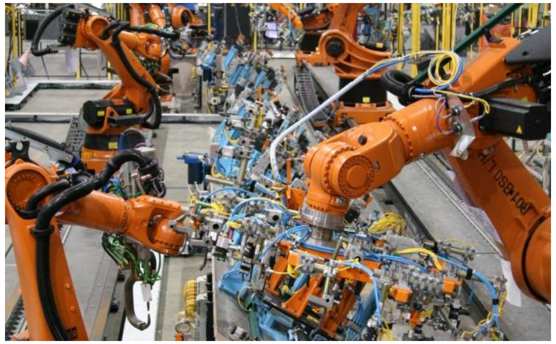
Thanks for inviting me to Geilo!

- PhD at Imperial College in High Energy Physics
  - Thesis analysis on radiative penguin processes in B-meson decays
  - 2005-2010
- Postdoctoral researcher at Berkeley Lab, USA analysing data from CERN
  - 2010-2016
- Data Scientist at Arundo
  - Since September this year
  - My first position outside of academia!
  - You’re welcome to connect with me on linkedin

The Corporate Data Science Presentation

ARUNDO

Which industries can find insights from data?



Any industry with access to data!

Industrial assets contain a wealth of data

Equipment are fitted with numerous sensors constantly recording data

Historical data might span over a decade

What can these data tell us about known equipment failures?

Is it possible to predict future failures before they occur?

Not all industries are alike

Number of Devices

You might start with large statistics and small feature space

… or …

small statistics and large feature space

Number of Sensors

ARUNDO

How do we gain insights from data?

Engineer's approach:

I expect flow to increase before just before a seal failure in a compressor

Monitor flow and raise an alarm when it goes over a threshold

Data Scientist's approach:

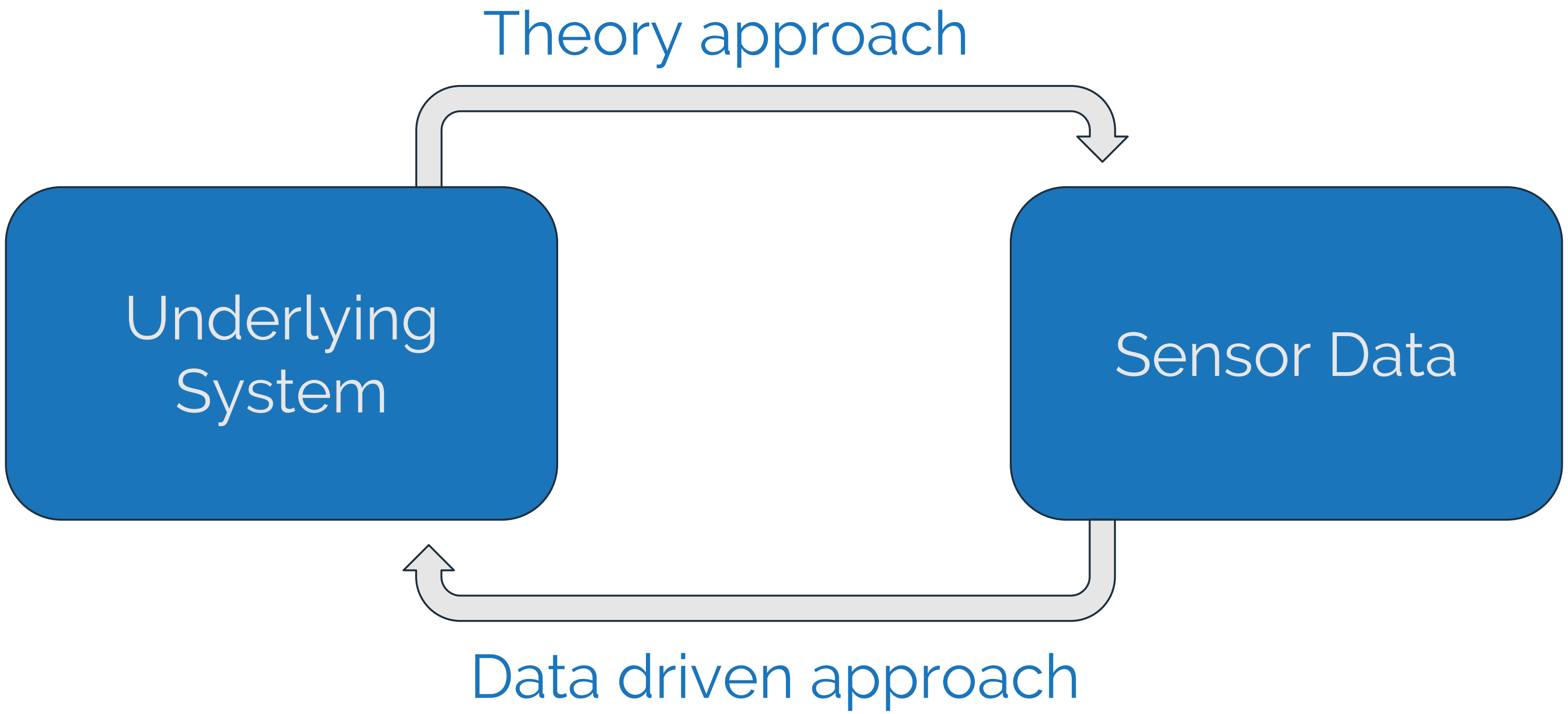
I know failure happened at time t

What can I infer from data far away from time t vs. just before time t

Is it possible to model?

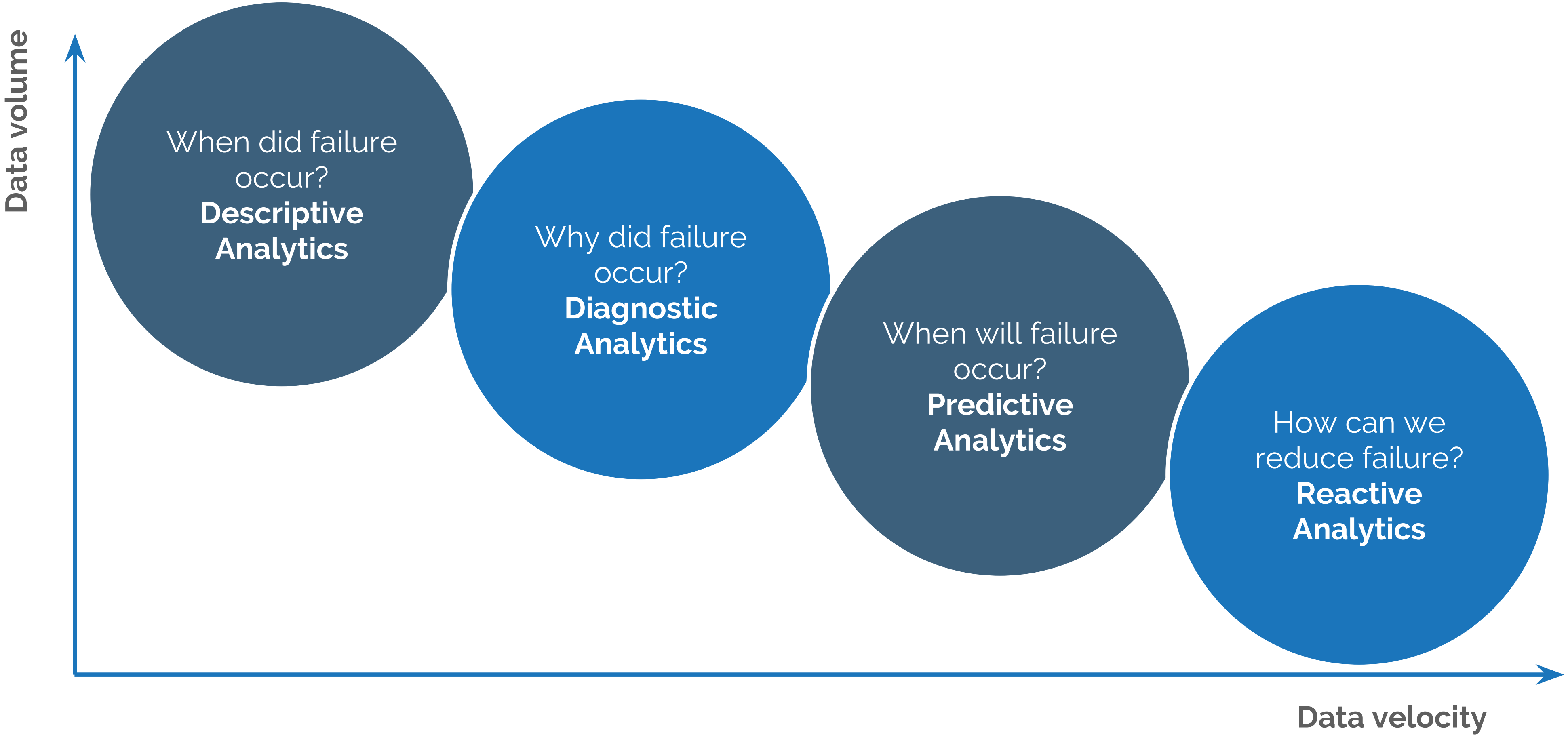
Raise an alarm based on the model output

How do we gain insights from data?



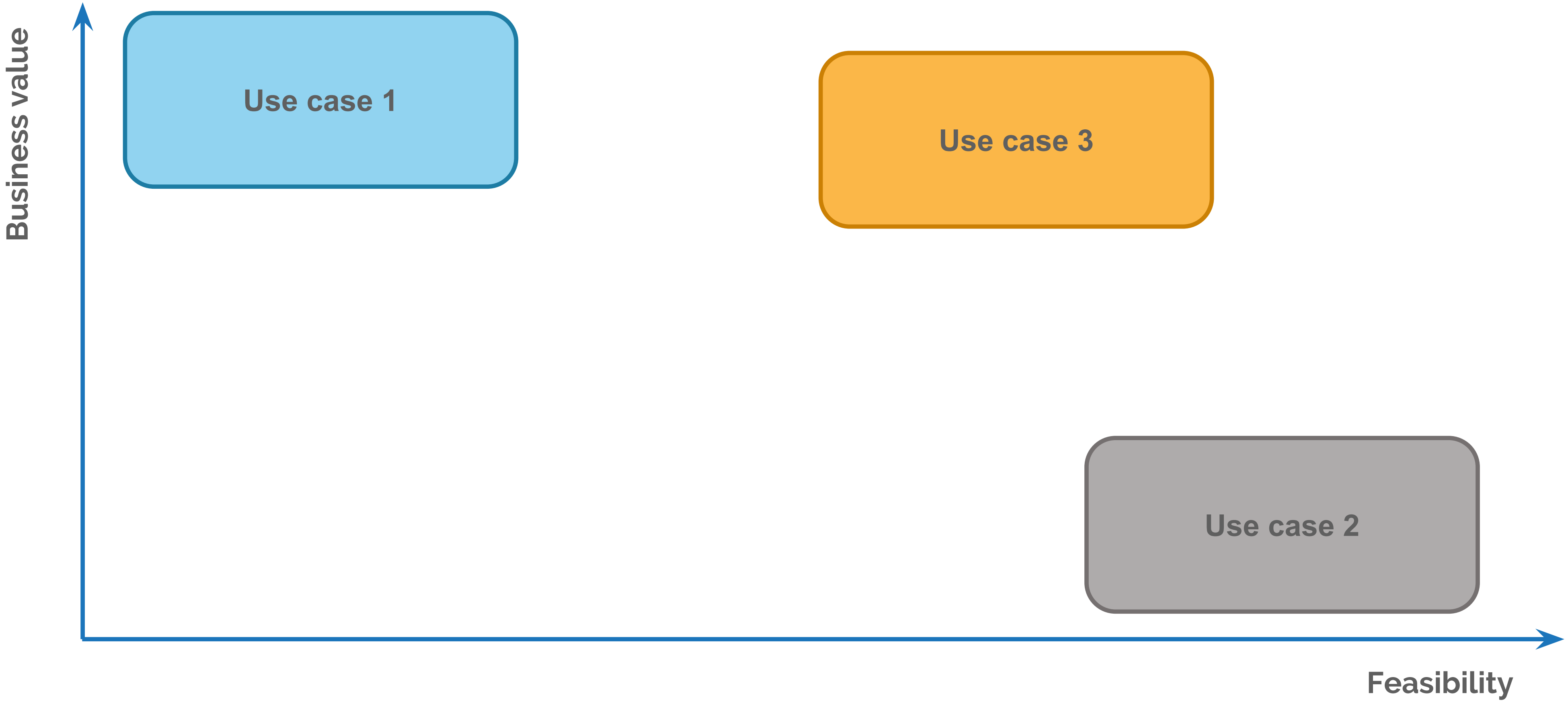
ARUNDO

Which questions **CAN** we answer?



ARUNDO

Which questions **SHOULD** we answer?

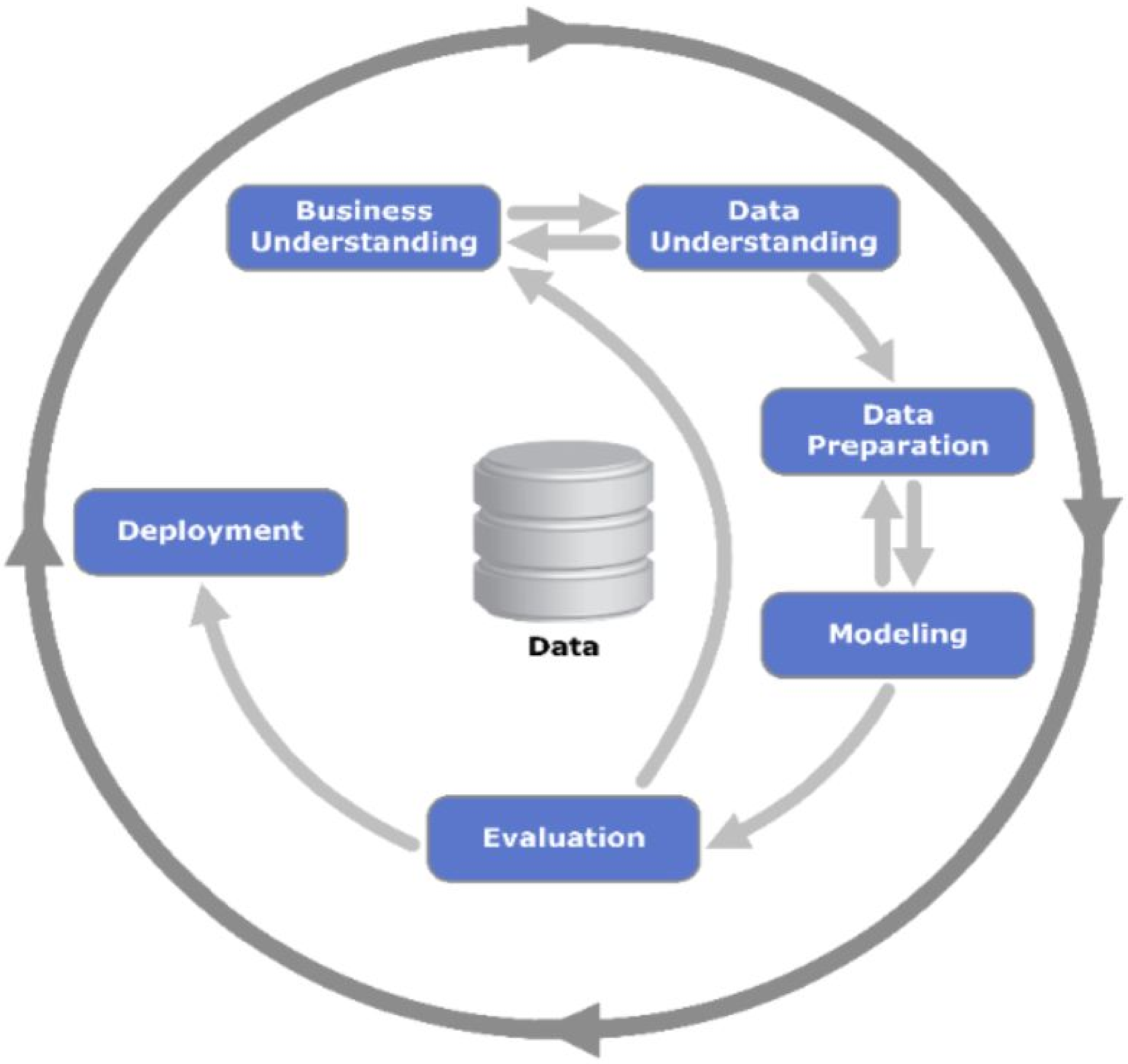


ARUNDO

CRISP-DM:

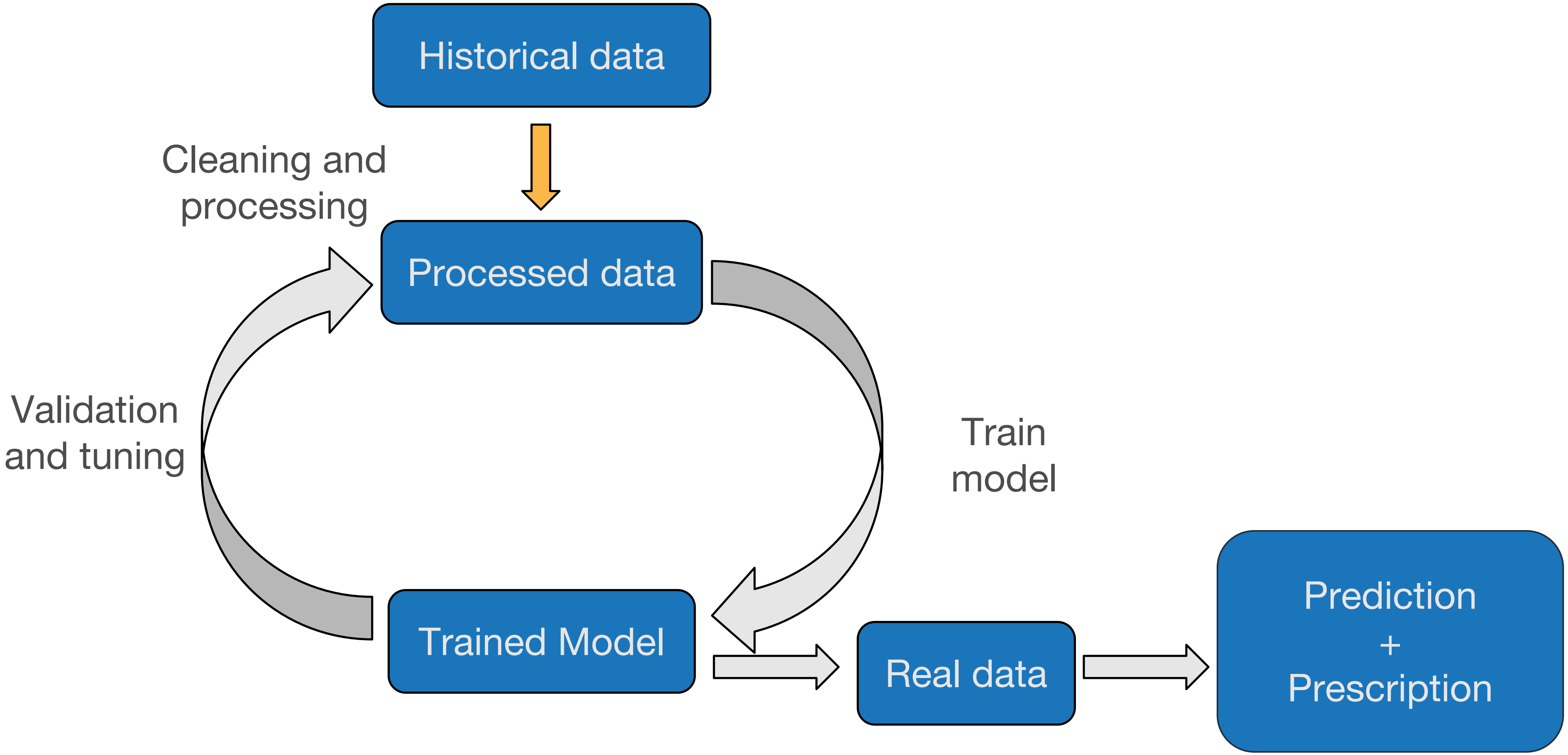
Cross Industry Standard Process for Data Mining

Data science as an agile process



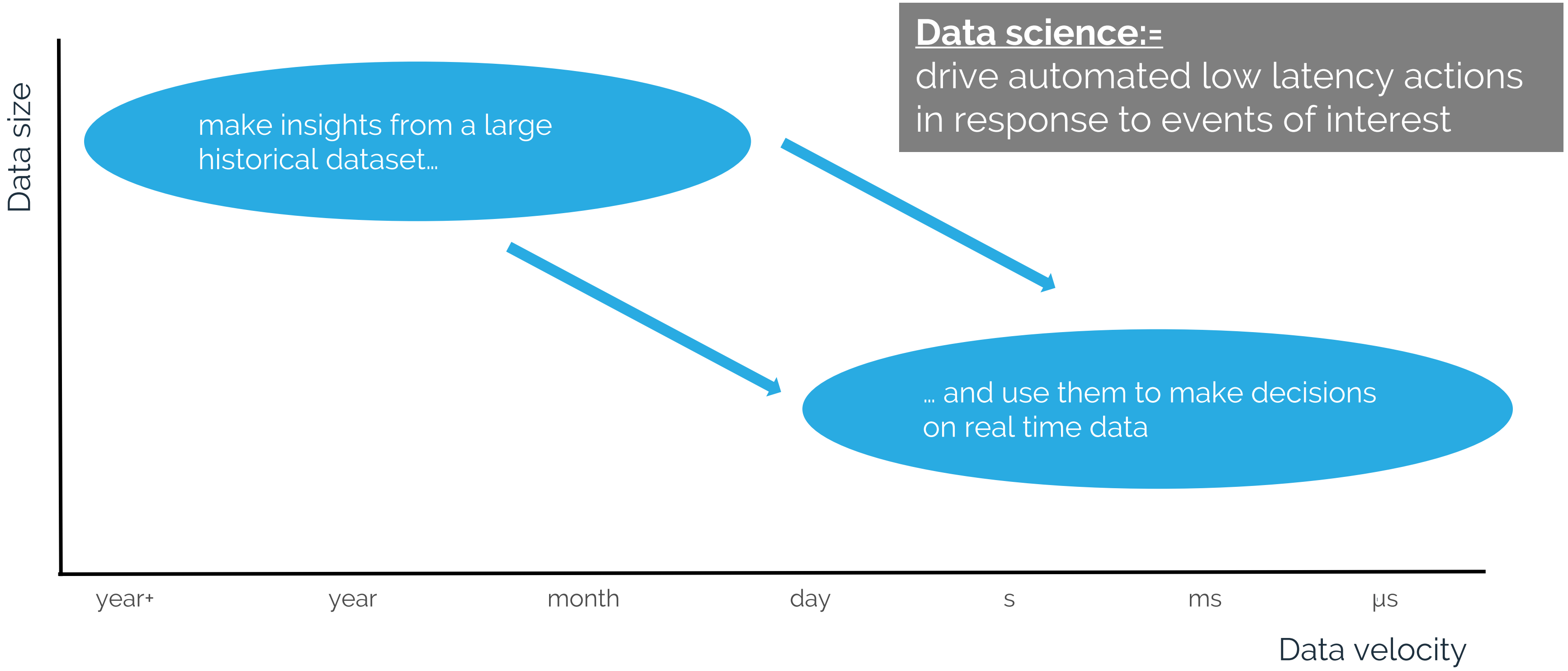
ARUNDO

Example data science workflow



ARUNDO

Big Data vs. Fast Data

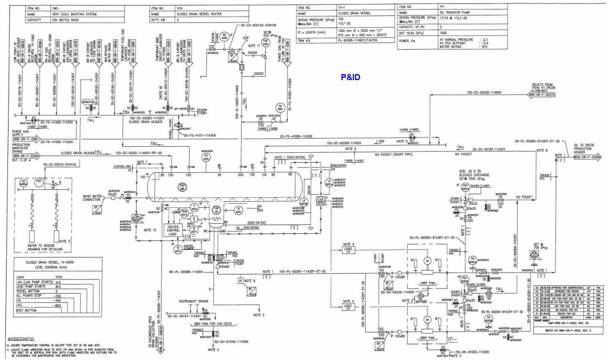


ARUNDO

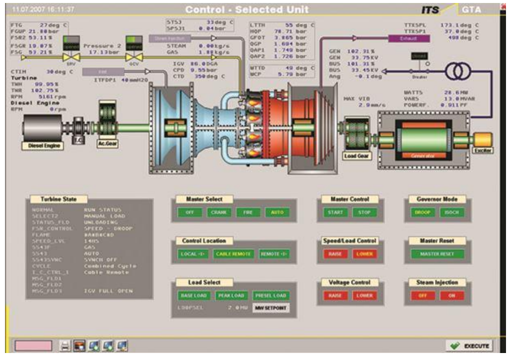
What does that look like in practice?

Where is the data? - Oil Rig Example

- DB containing all signals for all assets
- No direct indication of
  - Asset type
  - Physical meaning



- P&ID schematic shows which signals correspond to given asset
- Partial information on what is a pressure, temp, etc.
- Not all signals of interest are ‘part of’ the asset



- Control room monitoring SW displays physical meaning for each signal

Do I have to map between these manually?
How does that process scale and become automatic?

ARUNDO

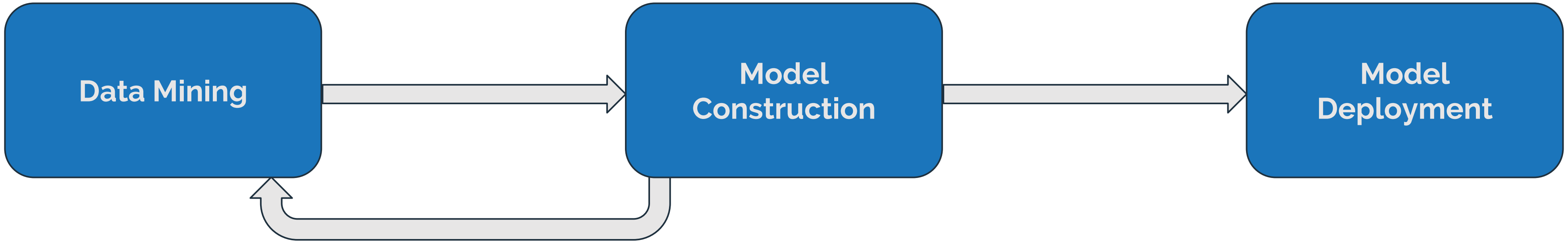
Where is the data? - Power Grid Example

- DB containing dynamic data from all smart meters
    - Consumption
    - Leakage current
    - ...

- DB containing static meter information
    - Parent transformer
    - City region
    - Annual consumption
    - 1-phase vs. 3-phase
    - ...

ARUNDO

Simplified Data Science Workflow



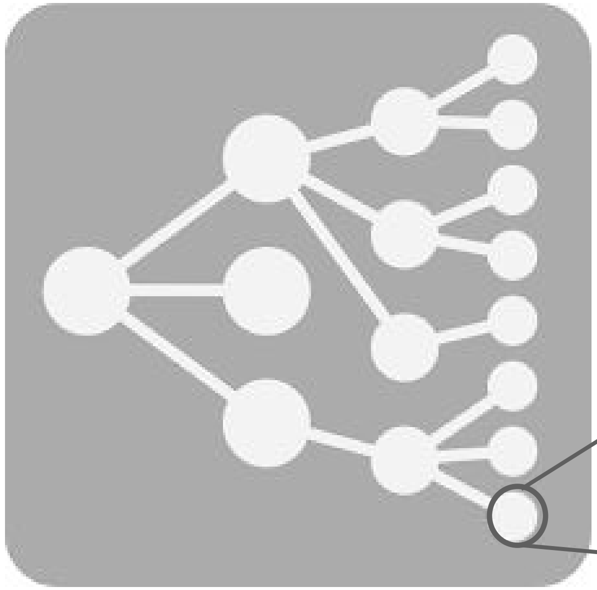
- Understand data structure.
- Address missing/null data.
- Identify which dependent variables provide insights.
- Determine and/or engineer input features.
- Assign required labels.
- ...

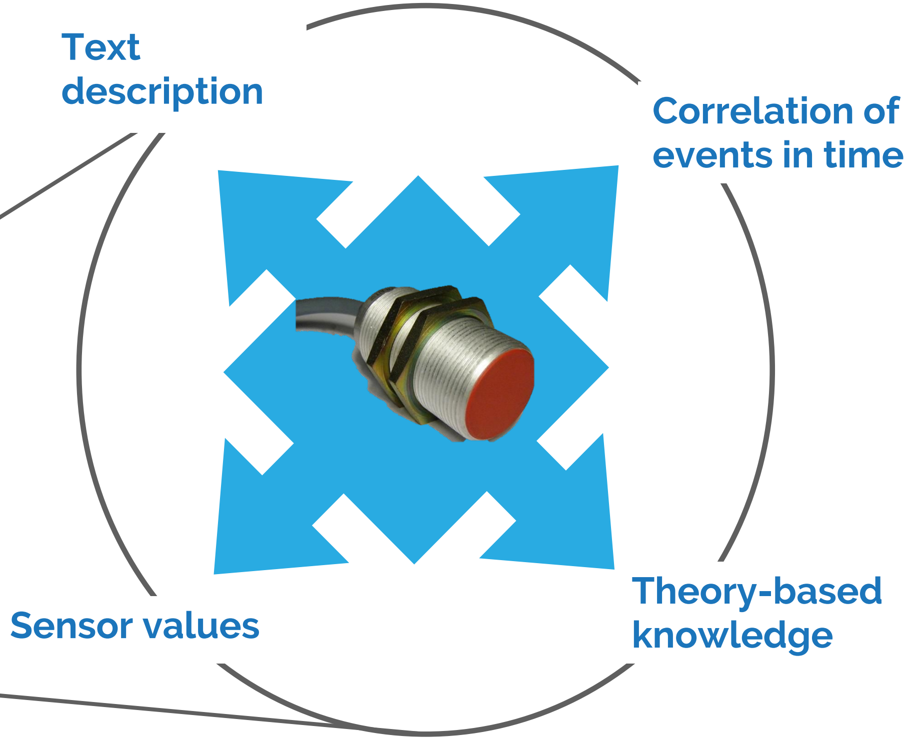
- Identify class of algorithm(s) to provide required insight.
- Apply algorithm(s) to input features.
- Assess performance with statistical metrics.
- ...

- Make model available to provide insights on independent or future data.
- Monitor continued performance of model.
- ...

ARUNDO

Equipment hierarchies





ARUNDO

Equipment hierarchies

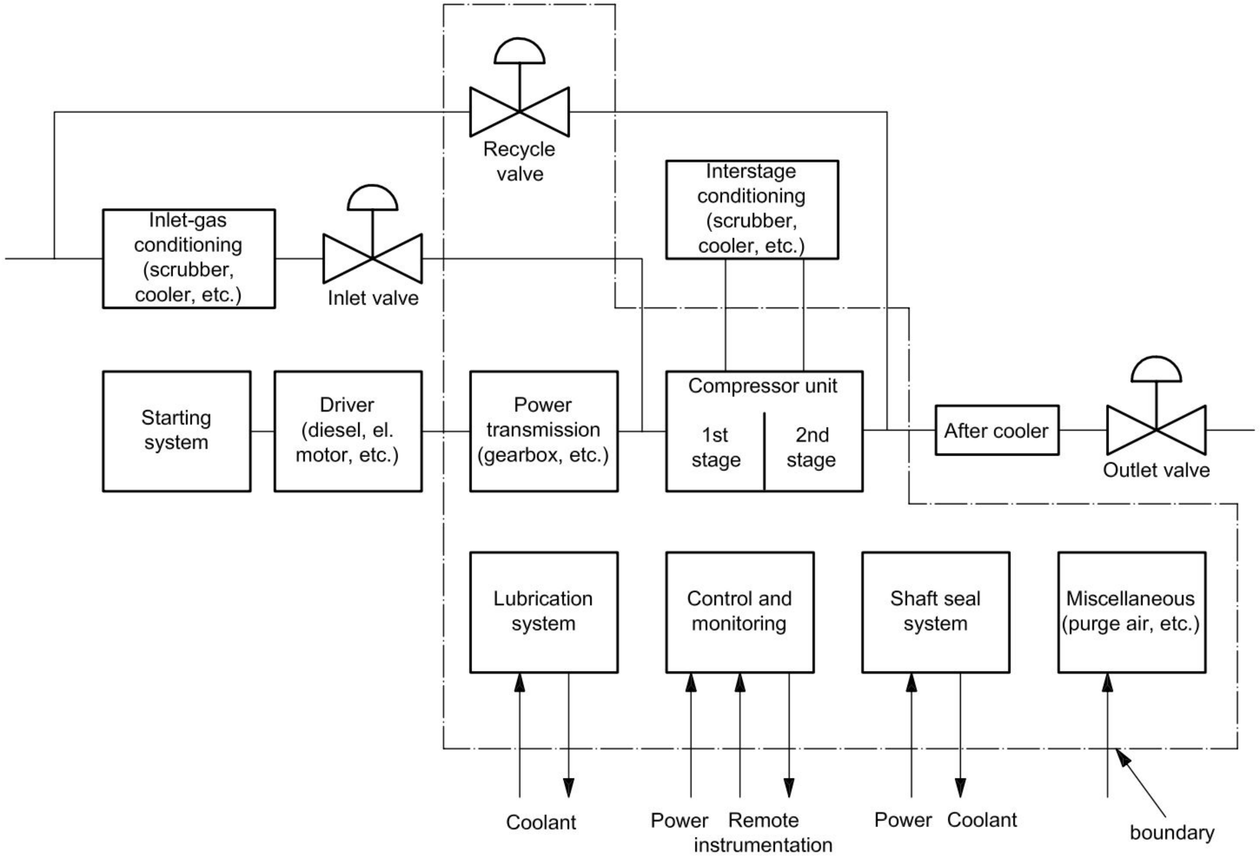


Figure A.2 — Boundary definition — Compressors

Table A.9 — Equipment subdivision — Compressors

Equipment class	**Compressors**					
Subunit	**Power transmission**	**Compressor**	**Control and monitoring**	**Lubrication system**	**Shaft seal system**	**Miscellaneous**
Maintainable item/Part	Gearbox/ variable drive Bearings Coupling to the driver Coupling to the driven unit Lubrication Seals	Casing Rotor with impellers Balance piston Interstage seals Radial bearing Thrust bearing Shaft seals Internal piping Valves Antisurge system [b] Piston Cylinder liner Packing	Actuating device Control unit Cables and junction boxes Internal power supply Monitoring Sensors [a] Valves Wiring Piping Seals	Oil tank with heating system Pump Motor Check valves Coolers Filters Piping Valves Lube oil	Oil tank with heating Reservoir Pump Motor Gear Filters Valves Seal oil Dry gas seal Mechanical seal Scrubber	Base frame Piping, pipe support and bellows Control valves Isolation valves Check valves Coolers Silencers Purge air Magnetic-bearing control system Flange joints

[a] Specify type of sensor, e.g. pressure, temperature, level, etc.

[b] Including recycle valve and controllers.

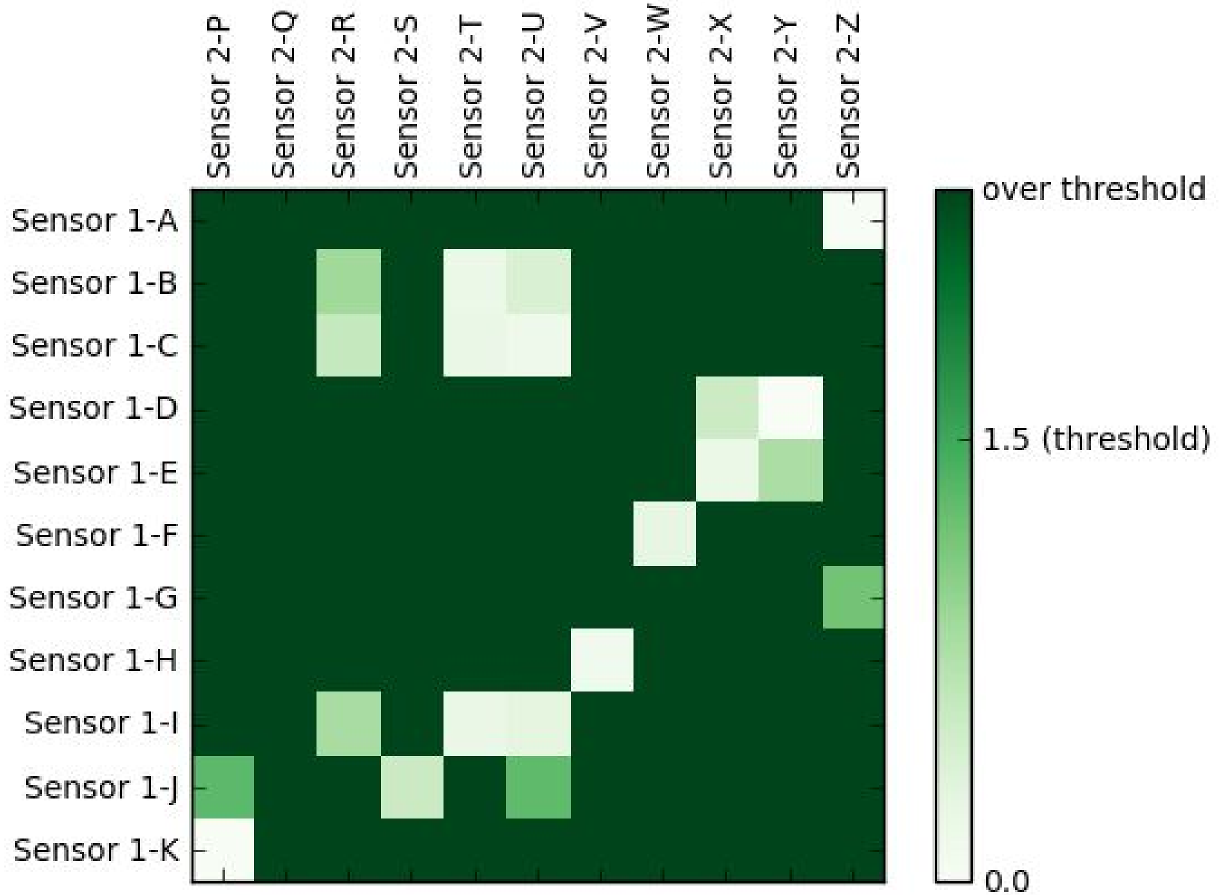
ARUNDO

Was mapping correct for all devices?

Sensors on different devices corresponding to the same physical value should look similar

Not a perfect science:

Outputs of some devices are inputs to other devices



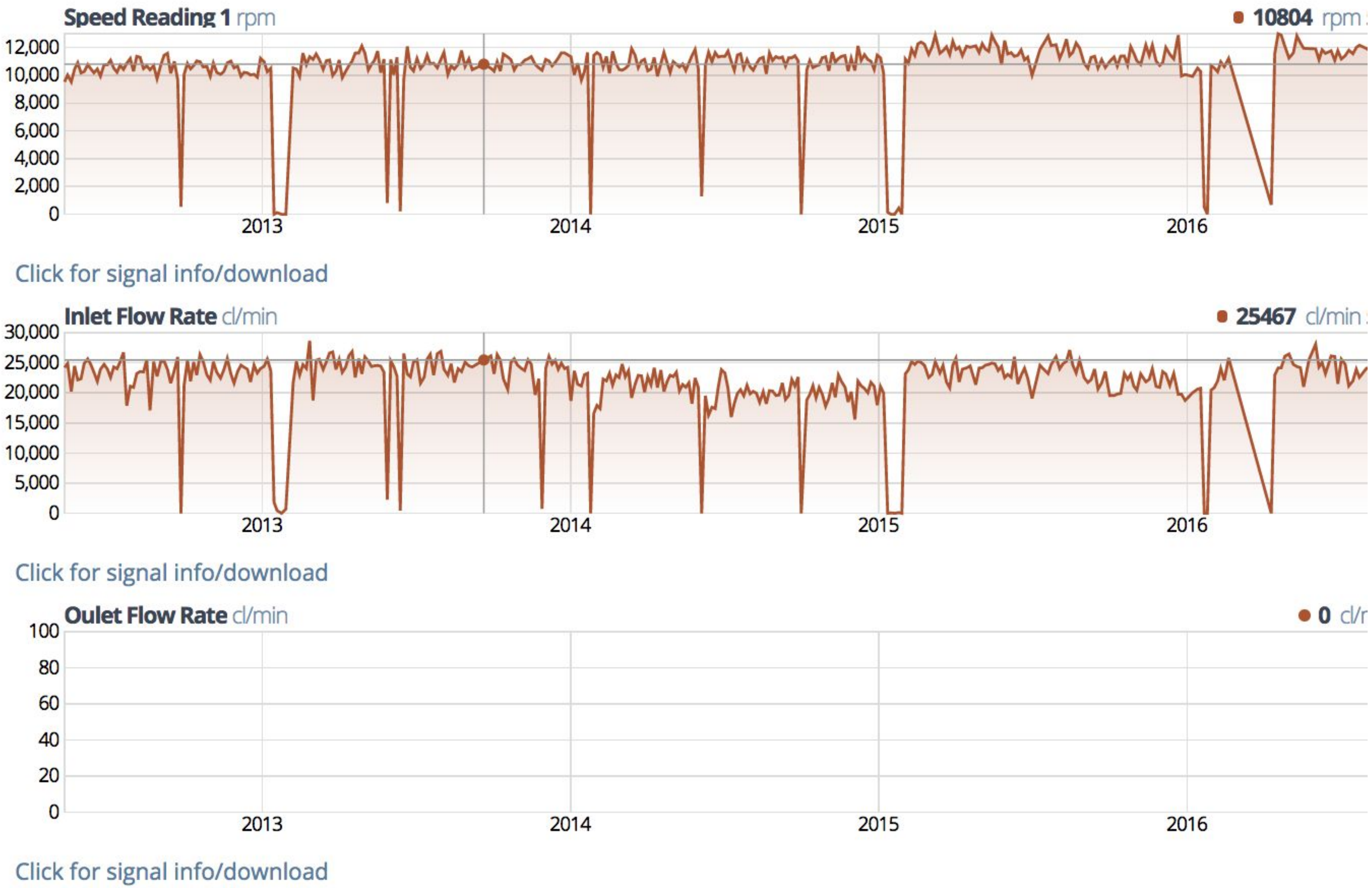
ARUNDO

Single device data quality example: oil rig pump

Periods where device switched off correlated between signals

Sensor values ramp down: just removing zero data might not work

Some signals just weren't in the DB!



ARUNDO

Correctly labeling failures, leaks, etc. is not trivial

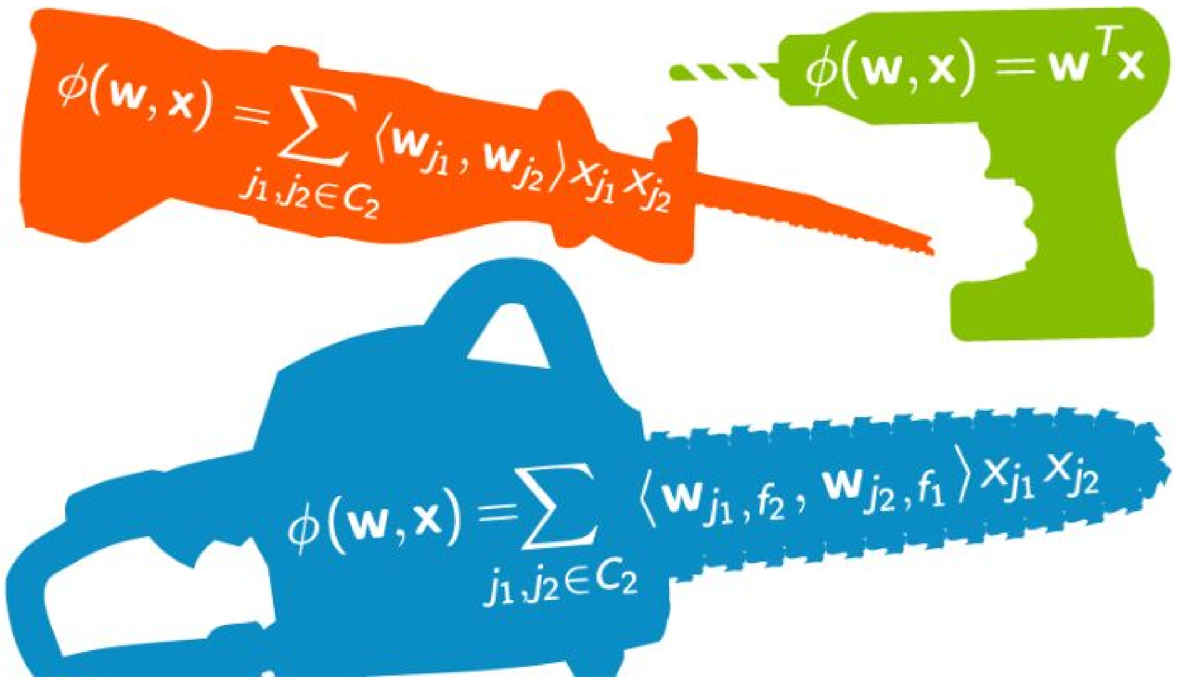
- Labels from maintenance logs
  - May require text mining
  - Reported dates can be inconsistent with where data should be labeled for modeling
  - Typos, different languages, incorrect reporting
  - Usually in different DB
- Can use sensor information to assign labels
  - Example: HC detector on oil rig asset can label leak events

ARUNDO

Feature normalization and feature engineering

- Depending on ML technique feature normalization is probably necessary
  - What does this mean for future data?

- Model may perform better when trained with differential or ratio variables
  - Ratios can be more stable as a function of time



ARUNDO

Data completeness is not guaranteed across devices

	Device 1	Device 2	Device 3	Device 4	Device 5
Inlet Pressure	✓	✗	✓	✗	✓
Outlet Pressure	✓	✓	✓	✓	✗
Inlet Temperature	✗	25% null data	✓	✗	70% null data
Outlet Temperature	✓	✓	✗	Actually outlet T for Device 1	✗
Flow	✗	✗	✓	✓	✓
Surface Temperature	only after 2012	✗	only after 2012	✗	✗
Axial Vibration	✓	✗	✗	✗	✗
Valve position	✓	✓	✓	✓	✓
Cooling level	✓	✓	✓	✓	✗

Device 3 has a reading once every 60 minutes while all other devices have a reading once every 35mins

ARUNDO

Isn't This Talk About Machine Learning?

ARUNDO

Data Science Tools

1 Find Data

Platforms
- Hadoop (other)
- SAS HPA
- AWS

2 Write Code

Editing Tools
- Vi/Vim
- Emacs
- Smultron
- TextWrangler
- Eclipse
- Notepad++
- IPython
- Sublime
- Atom

Languages
- SQL
- Bash scripting
- C
- C++
- C#
- Java
- Python
- R

3 Run Code

Interfaces
- pgAdminIII
- psql
- psycopg2
- Terminal
- Cygwin
- Putty
- Winscp
- Jupyter

4 Big Data

Hadoop
- Pig
- Hive
- Java
- (py)Spark

Cloud service
- MS Azure
- Amazon
- Google

5 Algorithms

Libraries

Java
- Mahout

R
- (Too many to list!)

Text
- OpenNLP
- NLTK
- GPText

C++
- opencv

Python
- numpy
- scipy
- scikit-learn
- Pandas

Programs
- Rstudio
- MATLAB
- Octave
- SAS
- Stata

6 Show Results

Visualization
- python-matplotlib
- python-networkx
- D3.js
- Tableau
- GraphViz
- Gephi
- R (ggplot2, lattice, shiny)
- Office

7 Collaborate

Sharing Tools
- Confluence
- Socialcast
- Github
- Google Drive & Hangouts

ARUNDO

Notebook Workflow Example: Simple Regression

Which is the best algorithm for gaining insights from data?

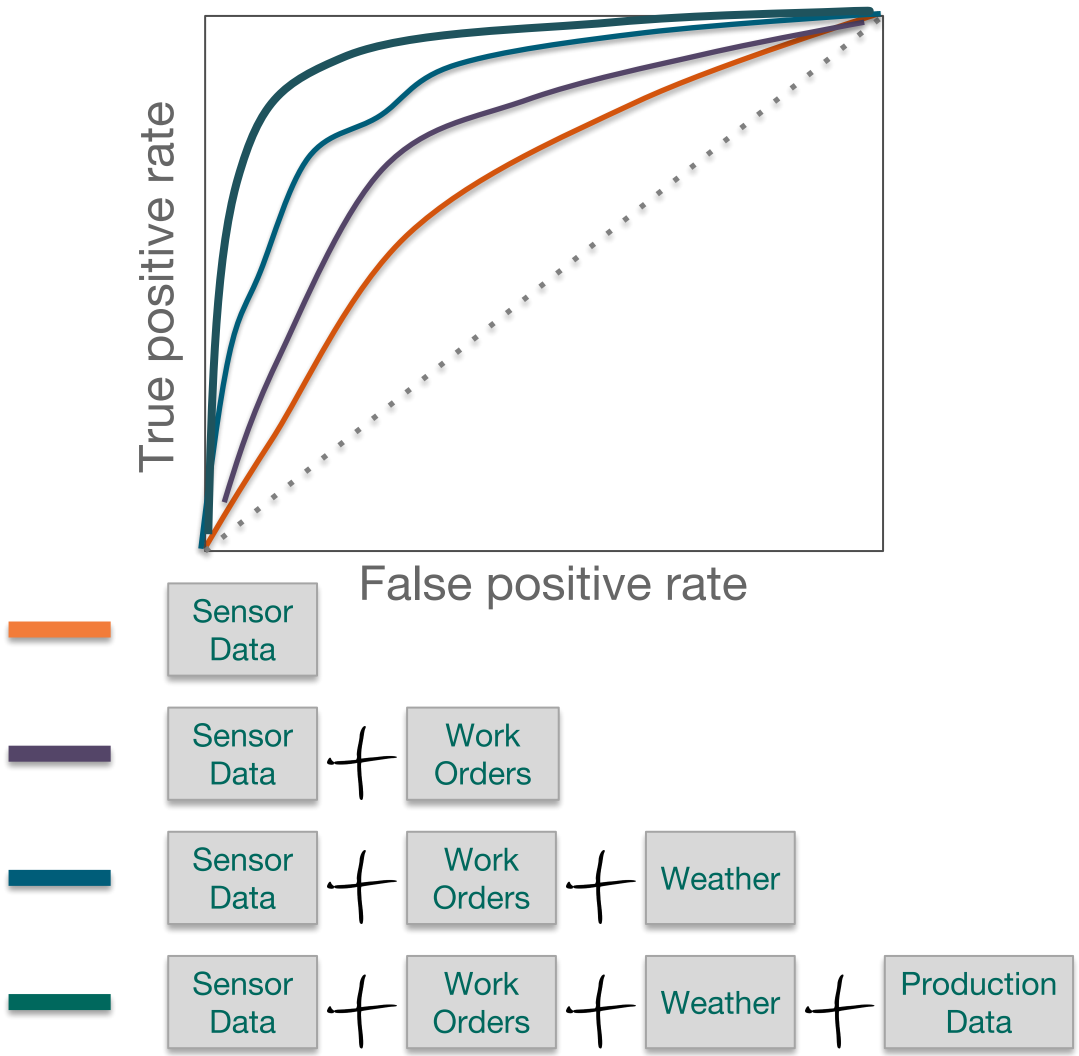
- Different models can be selected to determine predicted values
- Statistical figures of merit (FoM) determined using validation data
  - R^2, LMSE, etc. for regression
  - Accuracy, precision, recall for classification
- Typically choose the model with best FoM with preference for simpler models in the case of comparable FoMs.

ARUNDO

Options for improving algorithm performance

- In the case where no model can provide insights on available data two strategies can be explored
  - Obtain greater statistics for training and validation
  - Explore options for expanding the number of input features

- If neither of these is straightforward it may be best to just move to another use case



ARUNDO

Notebook Workflow Example: Classification of Downtime

Time series data can be complicated

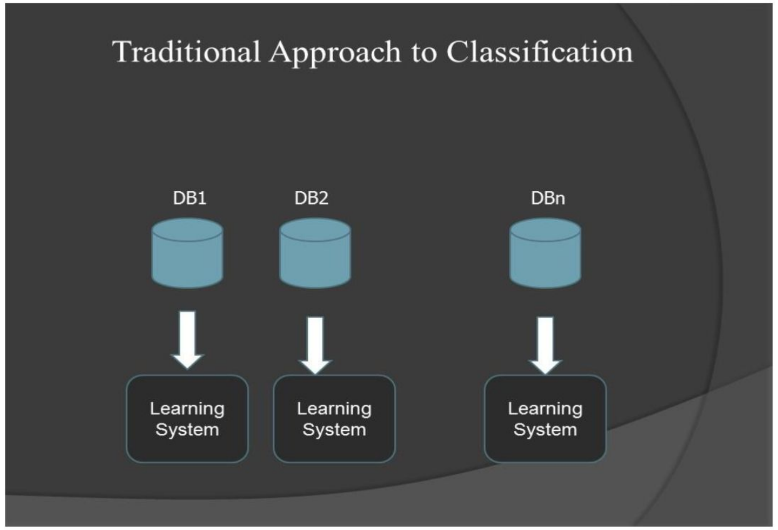
- Time series data may not naturally fit into what we're used to with ML because sequential data are not independent
    - Training strategy and model selection to predict future batches of data
    - Sensor values drift in time which can degrade model performance
    - Seasonal variations can impact modeling performance

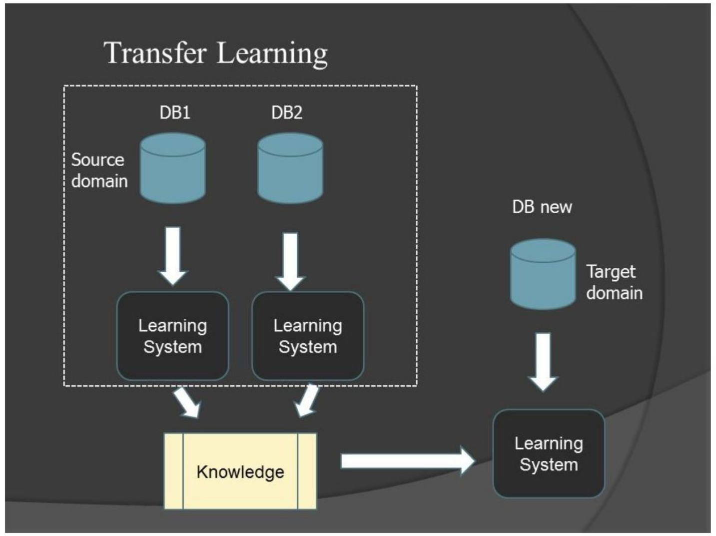
ARUNDO

More advanced techniques can also help

Reinforcement Learning, Adaptive Learning, Transfer Learning:

When a new problem is representative of an existing problem, or when data are known to change significantly over time



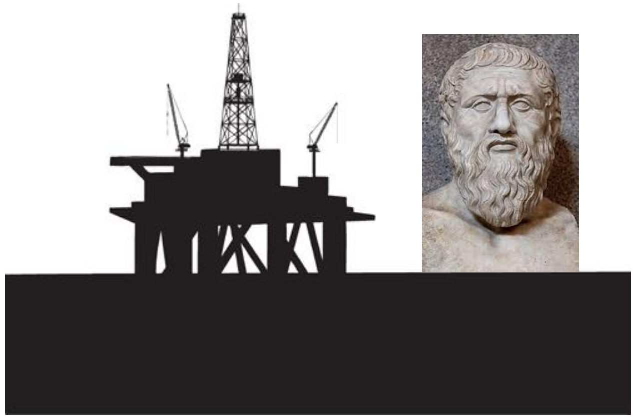


ARUNDO

Ensemble Learning

- One company may have limited data from their assets
- Idealized models can be constructed utilizing asset experience across companies
  - What is the ideal hierarchy?
  - What is the ideal set of sensors for a model?

Plato's Oil Rig



ARUNDO

Data Science Good Practice

Data science project structures

Collaborative data science is simplified using standard project structures

Cookiecutter is a good standard for projects in Python

- Clear documentation is as important as clear code
- https://drivendata.github.io/cookiecutter-data-science/

Directory structure

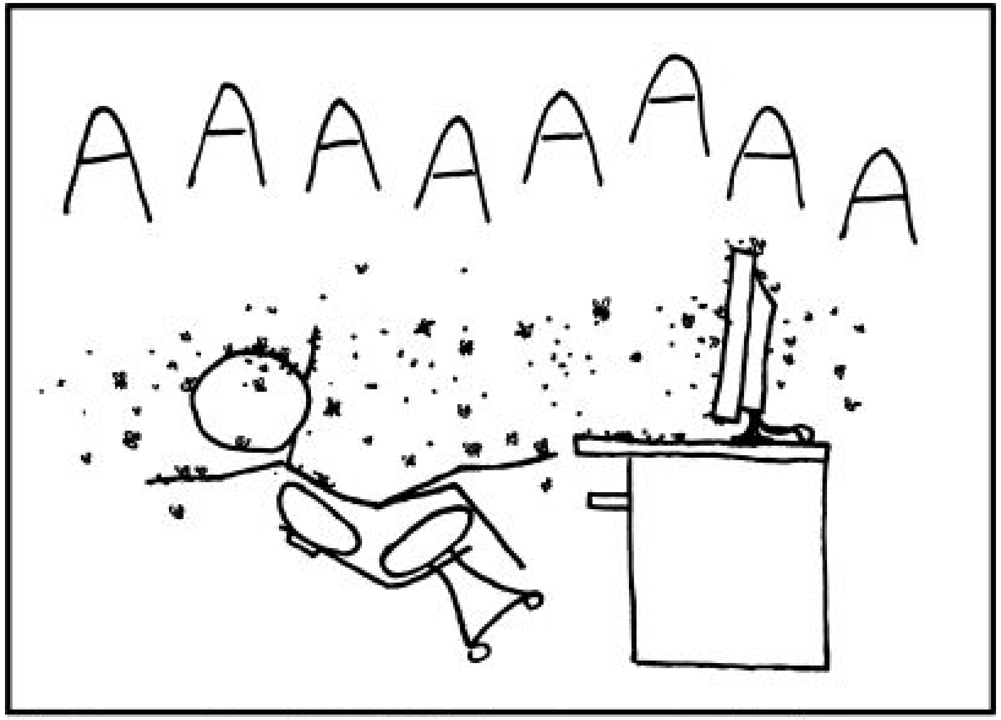
```
├── LICENSE
├── Makefile           <- Makefile with commands like `make data` or `make train`
├── README.md          <- The top-level README for developers using this project.
├── data
│   ├── external       <- Data from third party sources.
│   ├── interim        <- Intermediate data that has been transformed.
│   ├── processed      <- The final, canonical data sets for modeling.
│   └── raw            <- The original, immutable data dump.
│
├── docs               <- A default Sphinx project; see sphinx-doc.org for details
│
├── models             <- Trained and serialized models, model predictions, or model summaries
│
├── notebooks          <- Jupyter notebooks. Naming convention is a number (for ordering),
│                         the creator's initials, and a short `-` delimited description, e.g.
│                         `1.0-jqp-initial-data-exploration`.
│
├── references         <- Data dictionaries, manuals, and all other explanatory materials.
│
├── reports            <- Generated analysis as HTML, PDF, LaTeX, etc.
│   └── figures        <- Generated graphics and figures to be used in reporting
│
├── requirements.txt   <- The requirements file for reproducing the analysis environment, e.g.
│                         generated with `pip freeze > requirements.txt`
│
├── src                <- Source code for use in this project.
│   ├── __init__.py    <- Makes src a Python module
│   │
│   ├── data           <- Scripts to download or generate data
│   │   └── make_dataset.py
│   │
│   ├── features       <- Scripts to turn raw data into features for modeling
│   │   └── build_features.py
│   │
│   ├── models         <- Scripts to train models and then use trained models to make
│   │   │                 predictions
│   │   ├── predict_model.py
│   │   └── train_model.py
│   │
│   └── visualization  <- Scripts to create exploratory and results oriented visualizations
│       └── visualize.py
│
└── tox.ini            <- tox file with settings for running tox; see tox.testrun.org
```


ARUNDO

Environment control simplifies application of models

Collaborative data science requires clearly defined environments so everyone using your serialized model knows it will run

- Documentation, automatic or otherwise, of things like program version, library versions, dependencies, etc.
- Conda environments are a good solution for Python
    - http://conda.pydata.org/docs/using/envs.html



MY PACKAGE MADE IT INTO DEBIAN-MAIN BECAUSE IT LOOKED INNOCUOUS ENOUGH; NO ONE NOTICED "LOCUSTS" IN THE DEPENDENCY LIST.

ARUNDO

Apply good SW development practices to DS projects

- Write modular code
    - Can reuse when relevant
    - Reduce copy & paste errors
- Documentation
    - For yourself as well as others
- Version Control
- Testing
    - Good review of SW testing in data science:
        - https://www.youtube.com/watch?v=GEqM9uJi64Q
- Logging

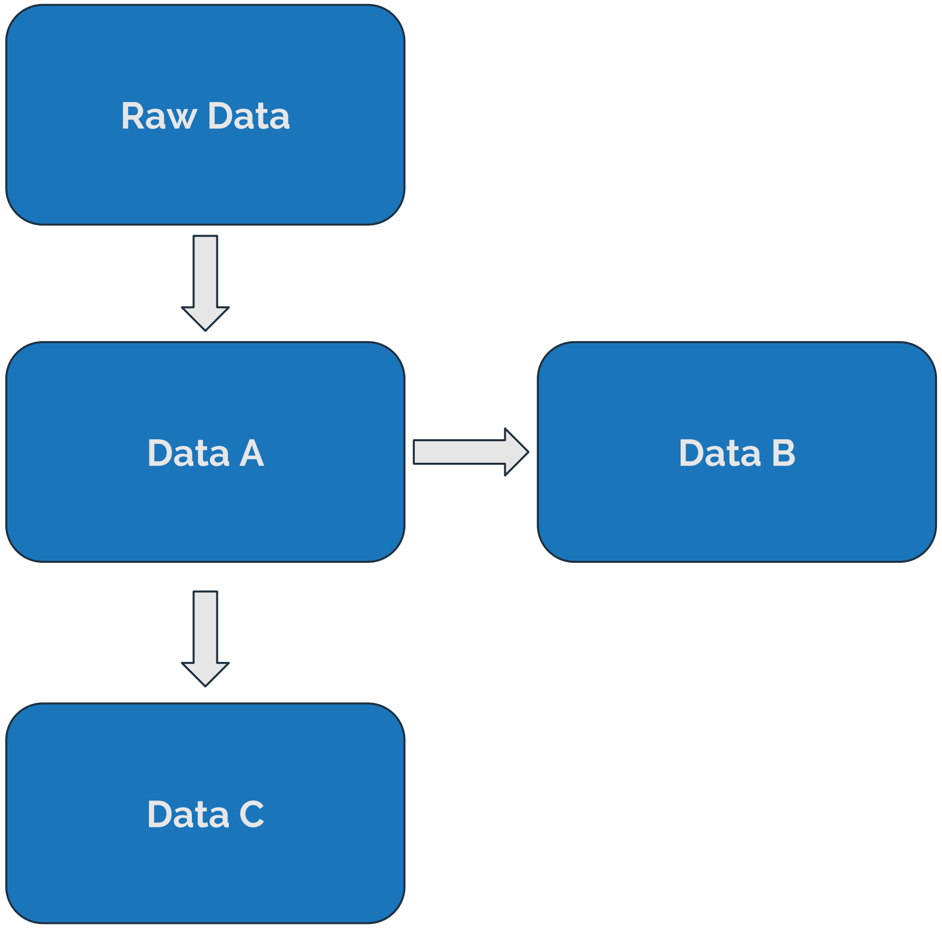
Bookkeeping the data processing workflow

Projects may require multiple data transforms each with dedicated code

Important to bookkeep so code-data dependencies are known

OS tools available such as Spotify's Luigi

https://pypi.python.org/pypi/luigi



ARUNDO

What happens when data is really big?

When data becomes too large for one device or data transforms become a bottleneck on a single processor cloud solutions become vital

- All data in one place
- Numerous tools exist for moving workflows to cloud
- Most of the tech big players offer cloud solutions

We are in the age of the (industrial) internet of things

Data security in the cloud worries many companies

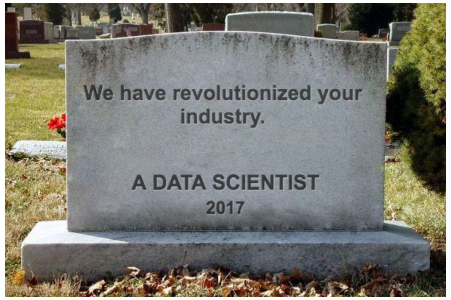
ARUNDO

I've made the perfect model what next?

Any model providing perfect insights on historical data is not the end of the story…

- How is that model going to be applied to future data?
  - Is the model going to be deployed to live streamed data?
  - Where is the future data? Local DB? Cloud DB?
- What happens when future data no longer look like historical data?
  - Retrain models? Adaptive learning?
  - Will an engineer take actions on a black box ML model output?

Many good models have died in powerpoint presentations



ARUNDO

Thank You For Listening!

ARUNDO

Some general definitions...

- **Data Mining:** the process of exploring and understanding the structure of data for further use.
- **Machine Learning:** computer algorithms which learn concepts and make subsequent predictions in the presence of data without explicitly being programmed to understand those data.
- **Input Features/Independent Variables:** the set of variables which are given as input to a machine learning algorithm.
- **Labels:** discrete classes identifying distinct properties of a given set of input features.
- **Predicted Values/Dependent Variables:** the set of variables or labels to be determined by the machine learning algorithm.
- **Insights:** conclusions drawn from the observation of a set of predicted variables.
- **Regression Algorithms:** the class of algorithms implementing statistical methods to predict a set of continuous dependent variables given a set of input features.
- **Classification Algorithms:** the class of algorithms implementing statistical methods to predict a set of discrete labels given a set of input features.
- **Supervised Learning:** class of machine learning methods which can be used to determine insights from data having been **trained** with independent data where the dependent variables are known.
- **Unsupervised Learning:** class of machine learning methods which can be used to determine insights from data with no prior information.

ARUNDO

Confusion matrix

		Model	
		1	0
Truth	1	True Positive	False Negative
	0	False Positive	True Negative

Accuracy = (TP+TN)/Total

Precision = TP/(TP+FP)

Recall = TP/(TP+FN)

F Score = 2*TP/(2*TP+FP+FN)

ARUNDO