



equinor

# Making it count - data science at scale

*Moving from single model building to automated model building*

Explore, develop and produce oil and gas



equinor

Operations in more than 30 countries

Deliver wind power to  
650,000 British households

Mounting solar panels in Brazil



## Our vision for data science at Equinor

Create **ML/AI technology** that gives Equinor **competitive advantages** in focus areas.

Avoid model by model tasks (unless we're exploring), build **products**.

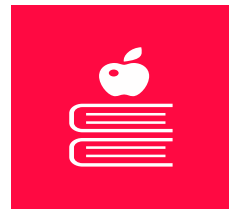


# Our structure supporting our goals and visions



**Insights & Advanced Analytics**

Builds and deliver interactive UX to access & analyse data across the value chain



**Knowledge AI**

Extracts knowledge from our vast unstructured data and make it available for use



**Machine Data ML**

Turns machine data into operation optimization capability across the value chain



**Self-Learning Robots & Vision**

Optimises highly dynamic problems and builds computer vision tools

## Core ML Teams

Build ML/NLU/RL technology at scale

**Subsurface DS**



**Data Management**

**US DS**



**Data Engineering**

**IOC**



**Software Development**

**Design Thinking**



**Data Platform**

## Business Teams

Work with business areas to implement value add solutions.

**Procurement**

## The data science team



23

by end of October



24%

And growing



5 cities

Stavanger, Bergen,  
Oslo, Austin, Houston



44% PhDs

Mathematics, geoscience,  
engineering...



60% External

Great improvement in  
attracting talent



6 Industries

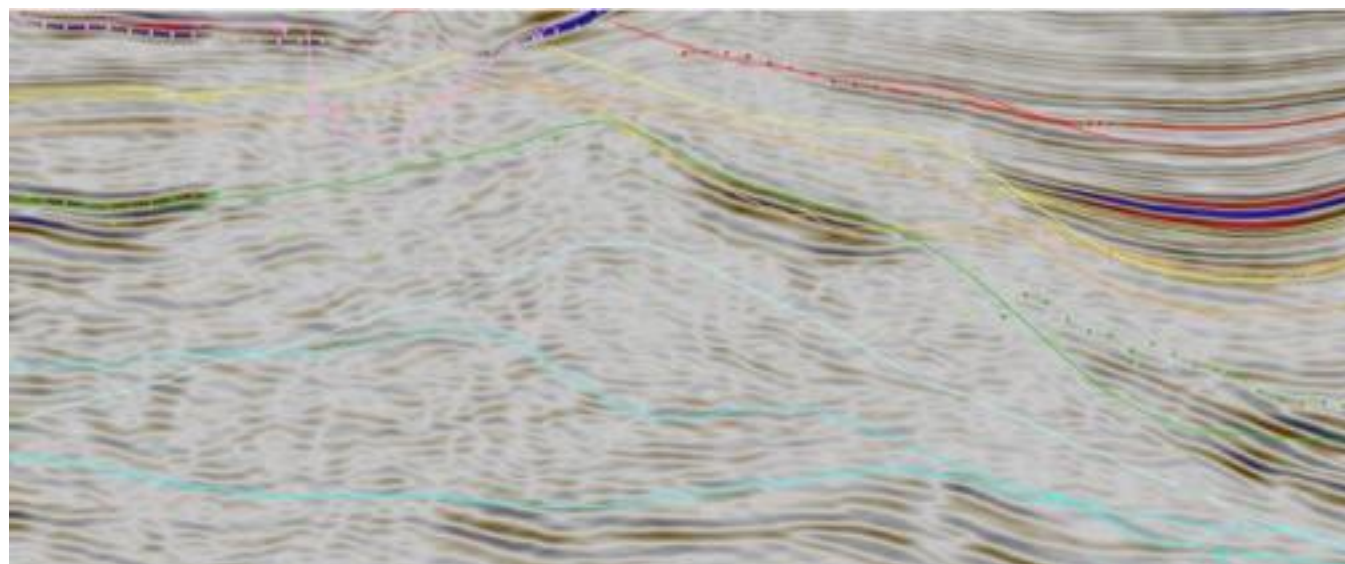
Consulting, automotive,  
tech, internet, finance



Equinor operates more than 40 platforms just in Norway

Thousands of potential cases (machines) and tens of thousands (towards) of different sensors



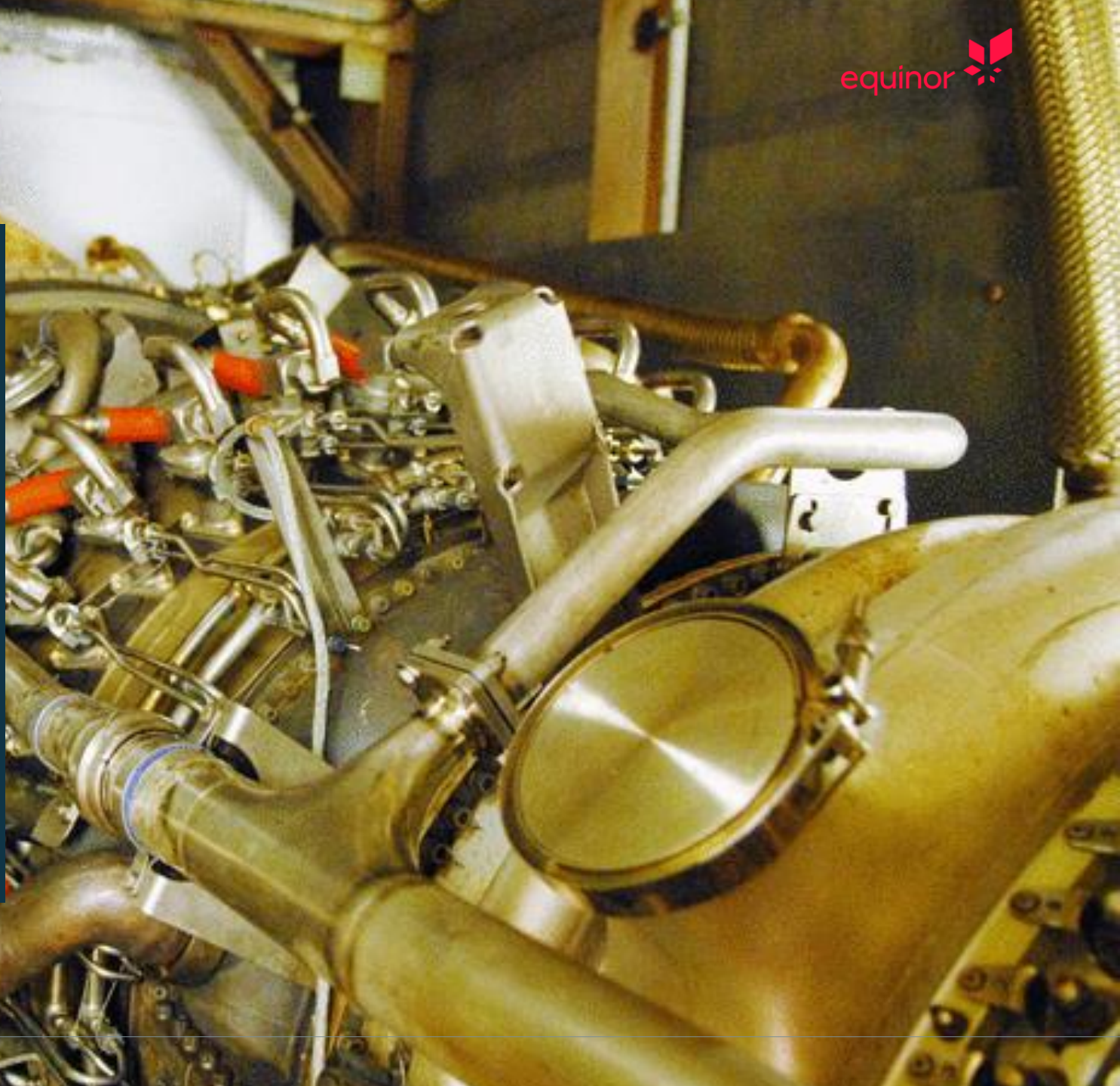




## We can't model every machine

One of our cases aim to monitor the *health* of a *machine* and reduce shutdowns.

We are building a general process engineers can apply themselves.





## Looking at single machines – but only if re-use is possible

```

train_df.dropna(thresh=len(train_df) - 10, axis=1,
inplace=True)
train_df.dropna(inplace=True)
train_col = train_df.columns

test_df = pd.concat(test_df, axis=1)
test_df.dropna(thresh=len(test_df) - 10, axis=1, in
place=True)
test_df.dropna(inplace=True)
test_col = test_df.columns

col = [ c for c in train_col if c in test_col]
train_df = train_df[col]
test_df = test_df[col]

print('Data loading complete!')
Data loading complete!
2j: train_df.head(5)
2j:


|                              | AMHX5XIN | DSVLVH2C | DSVLVLQI | PRCAX5XIN | PRDIFXIN |
|------------------------------|----------|----------|----------|-----------|----------|
| 2017-01-01<br>02:15:00+00:00 | 0        | 0.0      | 0.0      | 3.439829  | 75.43910 |
| 2017-01-01<br>02:30:00+00:00 | 0        | 0.0      | 0.0      | 3.439829  | 50.16793 |
| 2017-01-01<br>02:45:00+00:00 | 0        | 0.0      | 0.0      | 3.439829  | 27.78900 |
| 2017-01-01<br>03:00:00+00:00 | 0        | 0.0      | 0.0      | 3.439829  | 16.71324 |
| 2017-01-01<br>03:15:00+00:00 | 0        | 0.0      | 0.0      | 3.439829  | 0.00000  |


5 rows x 21 columns
3j: train_df.plot(figsize=(20, 15))
3j: %matplotlib.axes._subplots.AxesSubplot at 0x7f01e012b0b0

4j: test_df.plot(figsize=(20, 15))
4j: %matplotlib.axes._subplots.AxesSubplot at 0x7f01d94075c0

5j: #Elements well configuration
well_config = MachineConfig(well, 'tag', sensor_list)
6j: #Creates path for scaler and model storage. They will be us
ed to
return the models after training and recover the models if
condition

```

[illegible]

```
#####
#####

|0<00:00, 304901.37it/s|

#####

|1<00:00, 57555.81it/s|

#####

#####
dictionary...')

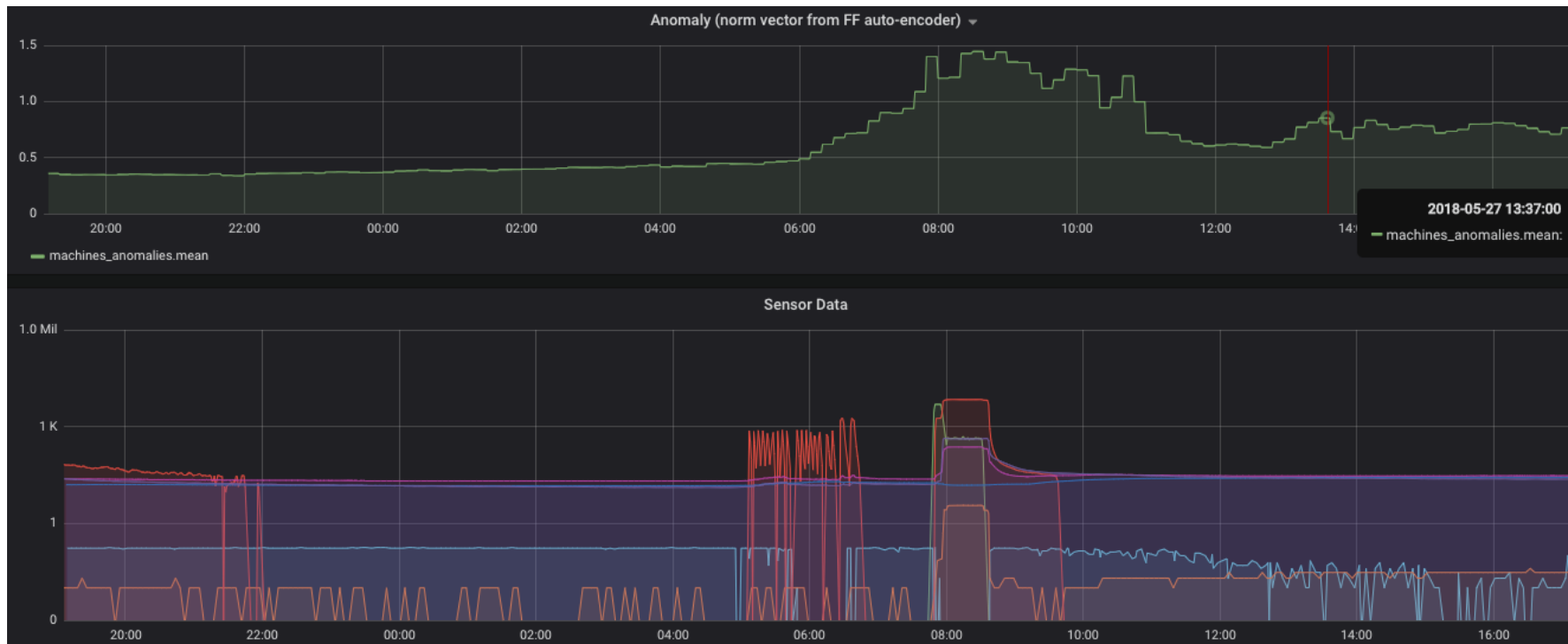
[l1 == key] for key in df_test.well.unique()
df.drop('well', axis=1).wells)
```

*Jupyter notebooks for trying out things.*

- Not easy to use for scientific studies
- Not the tool for solving a problem in a general way

What about the next 120 machines of the same type?

## The machine «health» index – from single machines to a reusability



The problem has many instances.

We must be able to solve all of them the same way.

- Different sensors
- Different patterns of operation



## The machine «health» index – from single machines to a reusability

**Input:**

New machine is (automatically) registered with a barebone of information

**Action:** Model is built

**Action:** Model is made available

**Action:** Predictions are being done automatically



Must be fully automated

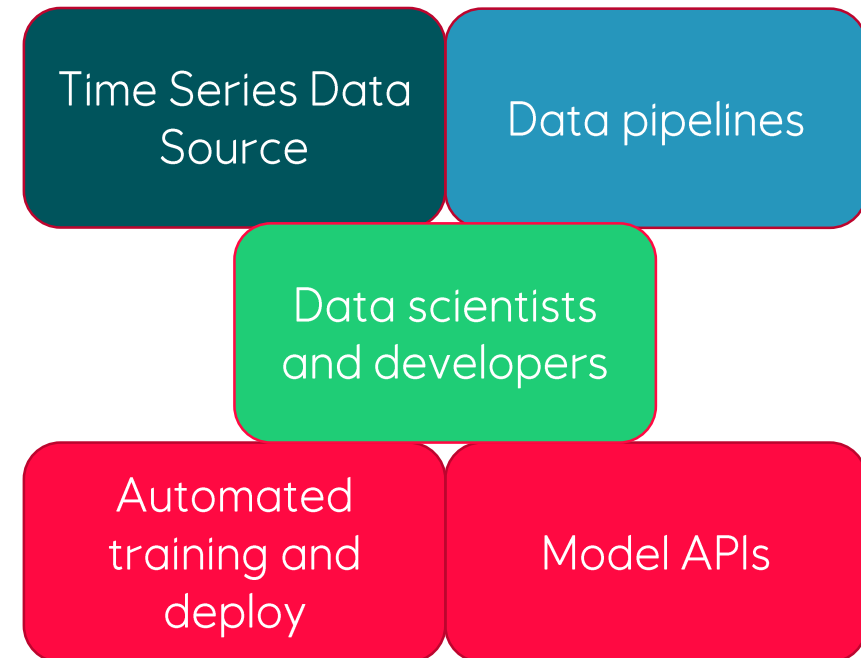
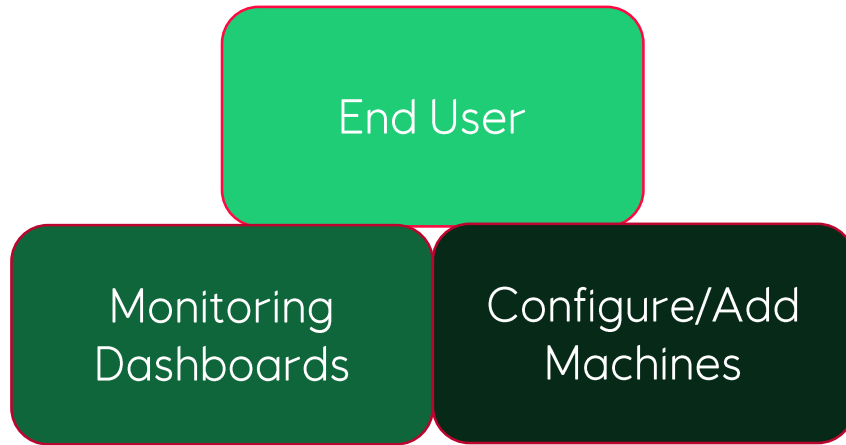
## Use case driven system development

Data science team collaborate  
with software developers in  
**one team**

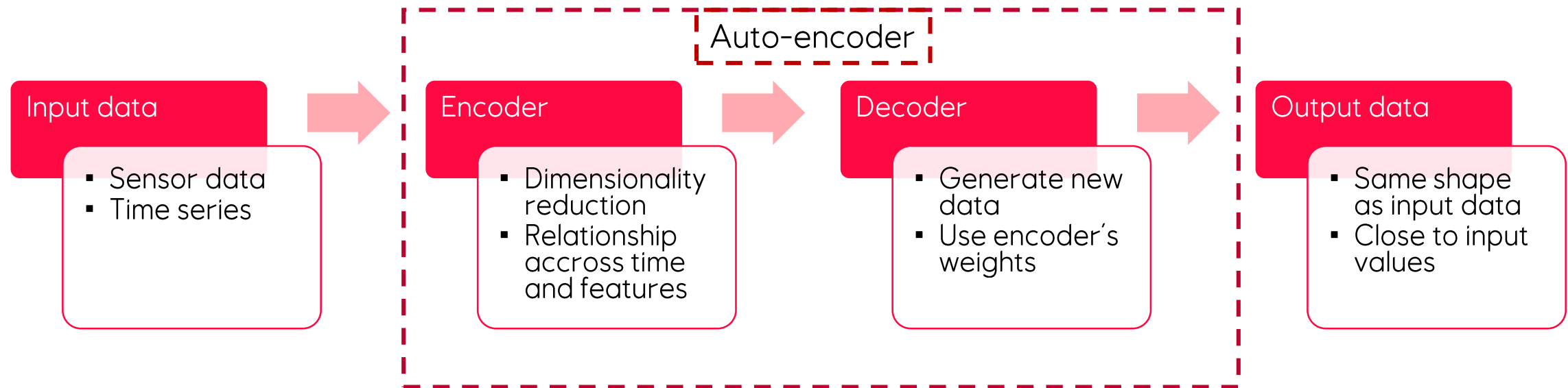
Mature **use cases** and  
**develop solution**  
together.



# Building the system



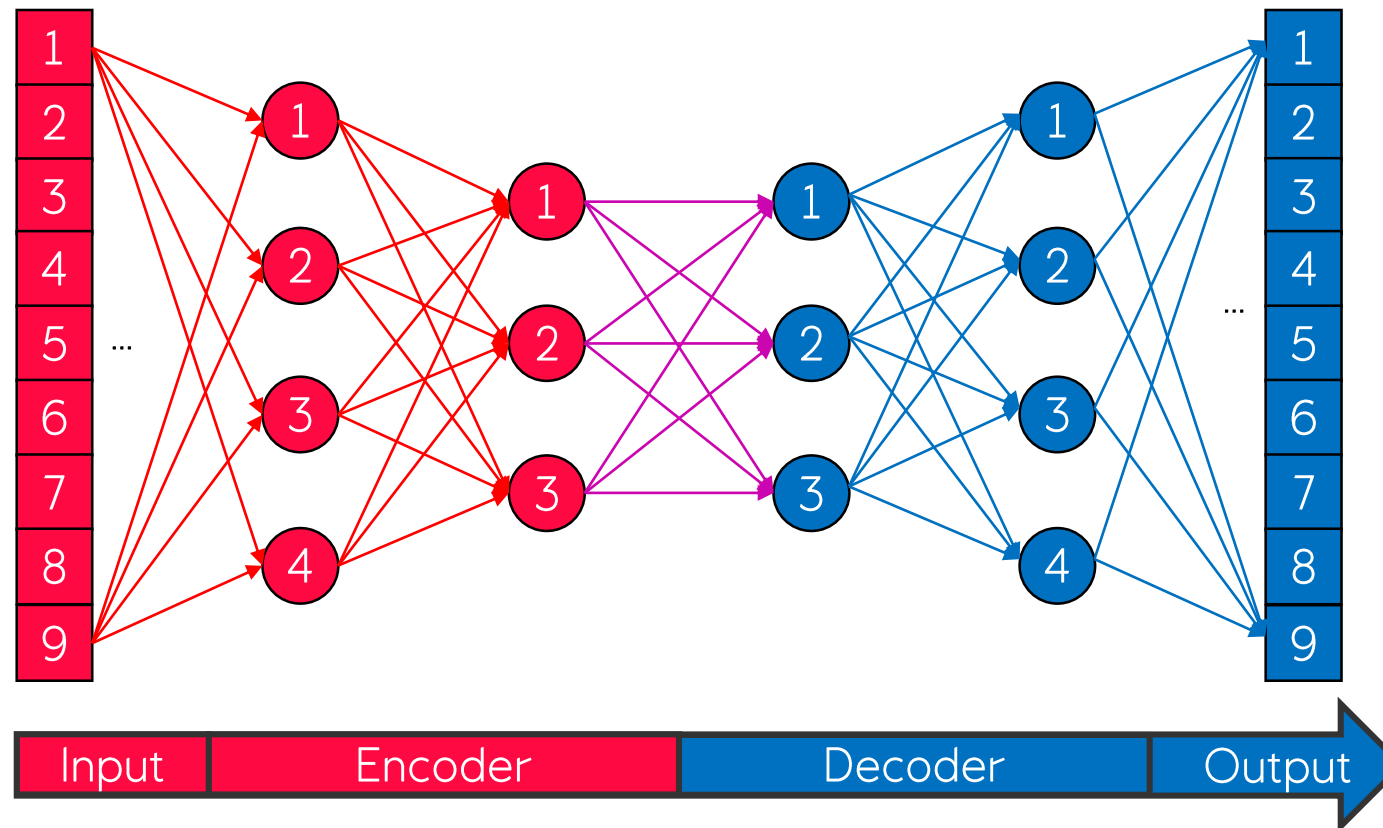
## Auto-encoder on sensor time series



*The difference between the «Input Data» and the «Output data» indicates normality or anomaly*



## The auto-encoder structure: example of feedforward neural network



## Summary

To make a difference, we must be able to create products that solve many instances

We need business understanding, analysis and software development skills within one tight group

There are too many cases and problems to be solved one by one