Best practices and lessons learned

Real-world use cases using traditional ML & deep learning techniques

YOUR UTILITY IT PLATFORM

V1.0 Lars van Geet, Kevin Schaul

12-03-2019

Outline

1. How do we approach a business problem with data science?

- General set-up of analytics projects

2. Which techniques do we use to solve business problems?

- Traditional Machine Learning techniques applied in projects
  - K-means
  - Text mining & Boosted decision trees
- Break
- Deep learning technique applied in projects
  - DNN

3. How do we embed Data Science solutions into the business

- Solution implementation
  - Data Factory Flow
- Wrap-up

About us

YOUR UTILITY IT PLATFORM

About Itility

WE BUILD AUTOMATION RESULT BASED

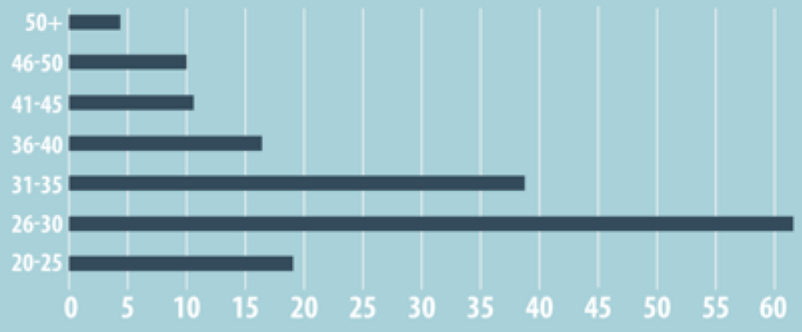
WE LOVE DEVOPS

WE FOCUS ON DATA

ROLES
(in %)

WOMEN / MEN

CATEGORY OF AGE



EMPLOYEE GROWTH

Customer base

Customer base

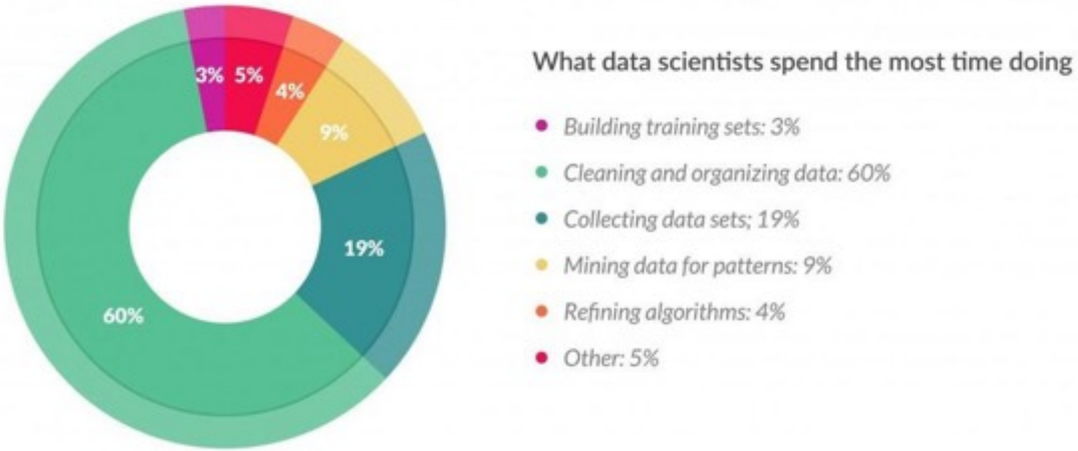
Innovation – initiatives with TU/e

General set-up of analytics projects

How do we approach a business problem with data science?

YOUR UTILITY IT PLATFORM

General set-up of analytics projects



* source: Forbes

Data science projects are complex messy

Involve many steps (time consuming)

&

people (communication)

General set-up of analytics projects

YOUR UTILITY IT PLATFORM

Guidelines & methods we use:

1. Define business problem with domain experts
2. Start small
3. Define a workflow which involves iterations
4. Standardize techniques/steps in the workflow

Define the business problem (1/3)

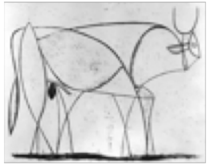
1. Define use cases based on problems of the domain experts

Name	Use case name (should be used in all references)
Description	Short description of the use case, the purpose and the business value.
Value	What does this offer.
Actors	Who is involved / should be involved and how.
Priority	How urgent is the problem.
Assumptions	What (if any) are the assumptions made before starting the use case.

Define the business problem (2/3)

1. Define use cases based on domain expert's issues
2. Summarize current situation/problem
   - Validation if your perception of the business situation is correct

Models



Simulations

Experiments

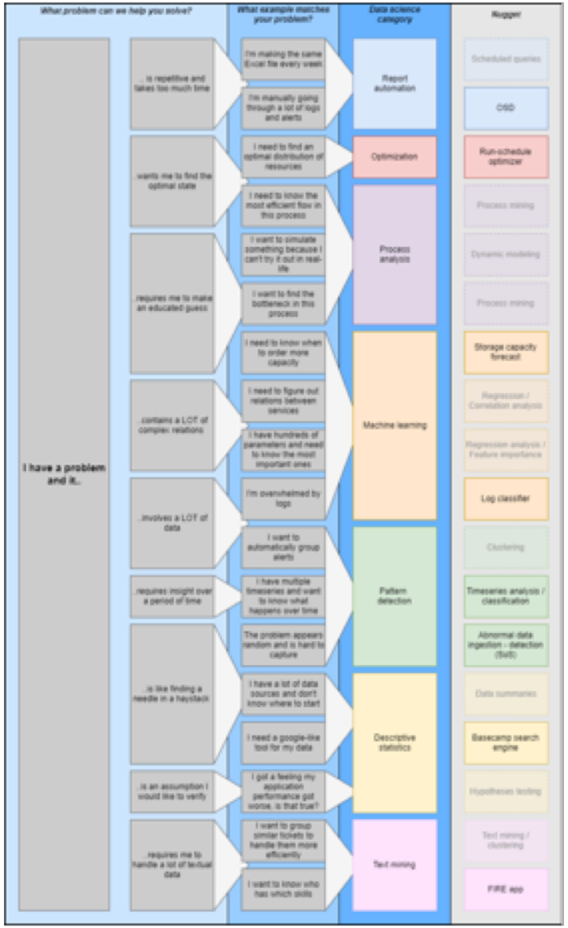
Reality

Domain experts

Data science

Define the business problem (3/3)

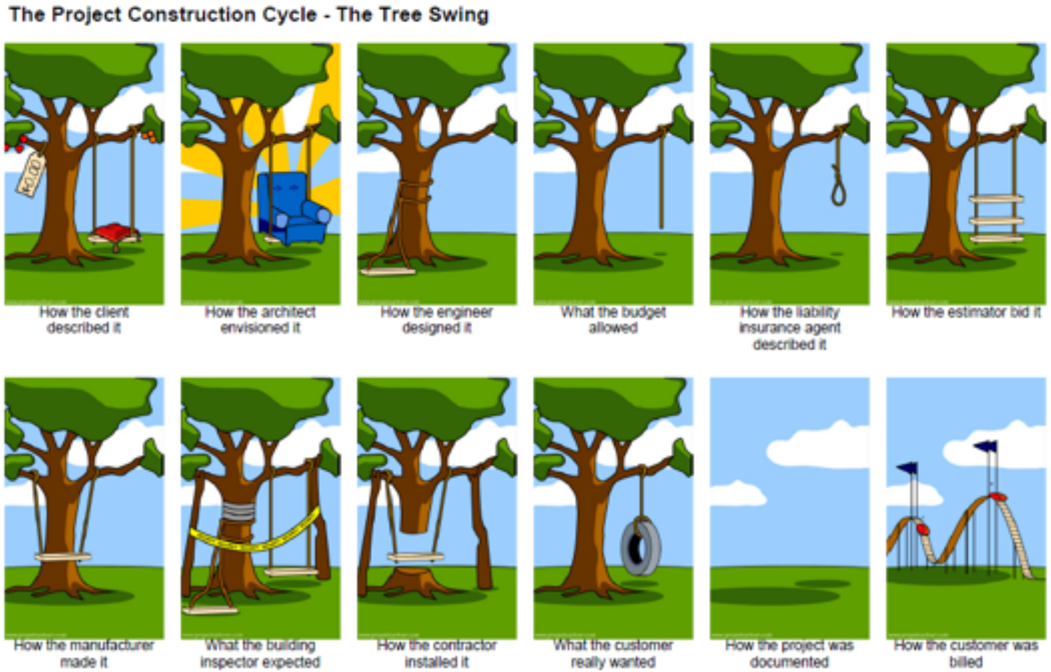
1. Define use cases based on domain expert's issues
2. Summarize current situation/problem
3. Translate the problem into solution(s)
   - Decompose problem into subtasks



Start small

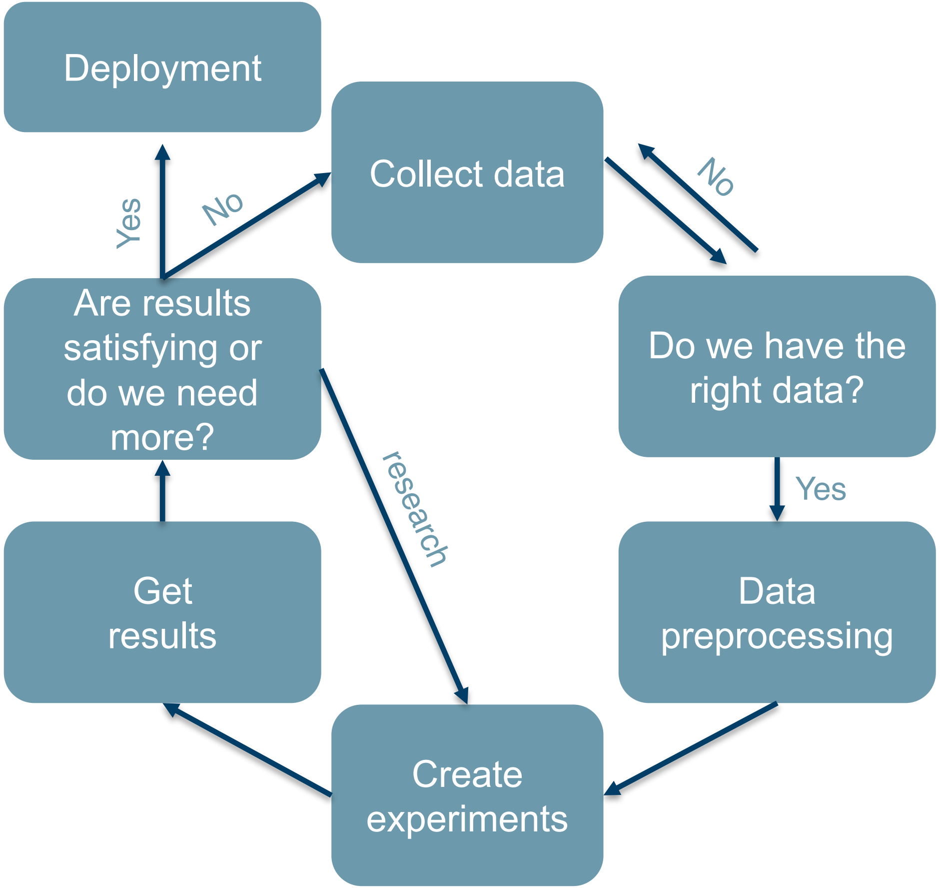
Start with a Proof of Concept

- Get results fast in order to fail fast
- Work with data dumps



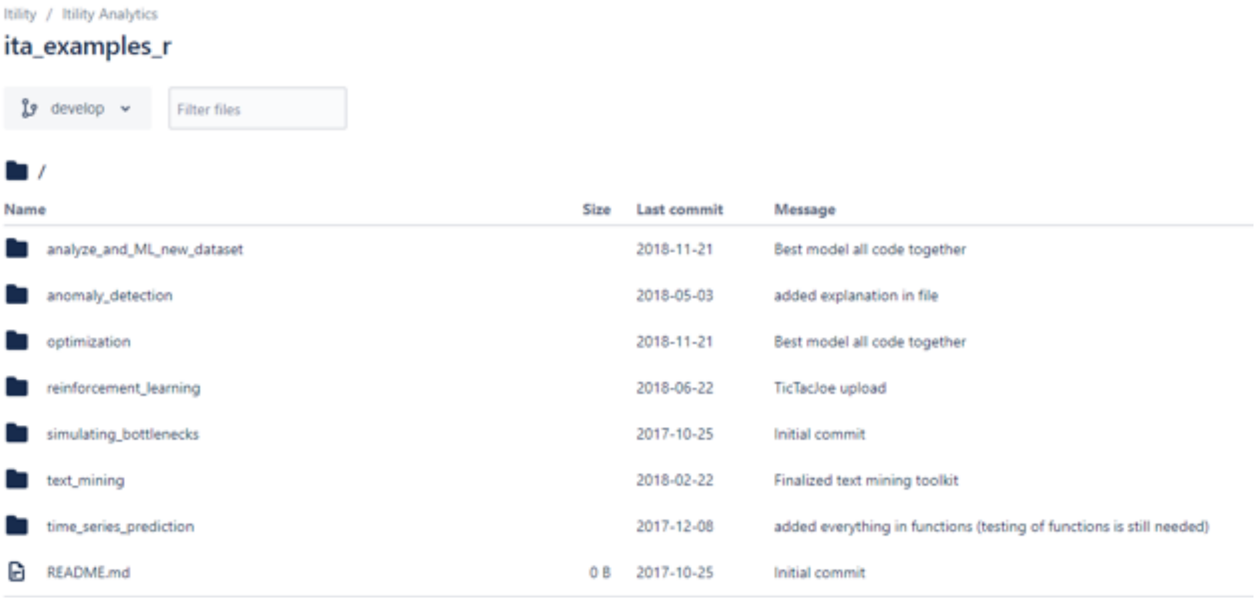
*source: projectcartoon

Define a workflow with iterations



Standardize techniques/steps in the workflow

- Create your own libraries (best practices)
- Automated ETL
- Pipelines



General set-up of analytics projects

Guidelines & methods we use:

1. Define business problem with domain experts
2. Start small
3. Define a workflow which involves iterations
4. Standardize techniques/steps in the workflow

Next: Which techniques do we use to solve business problems?

Laser refills

K-means clustering

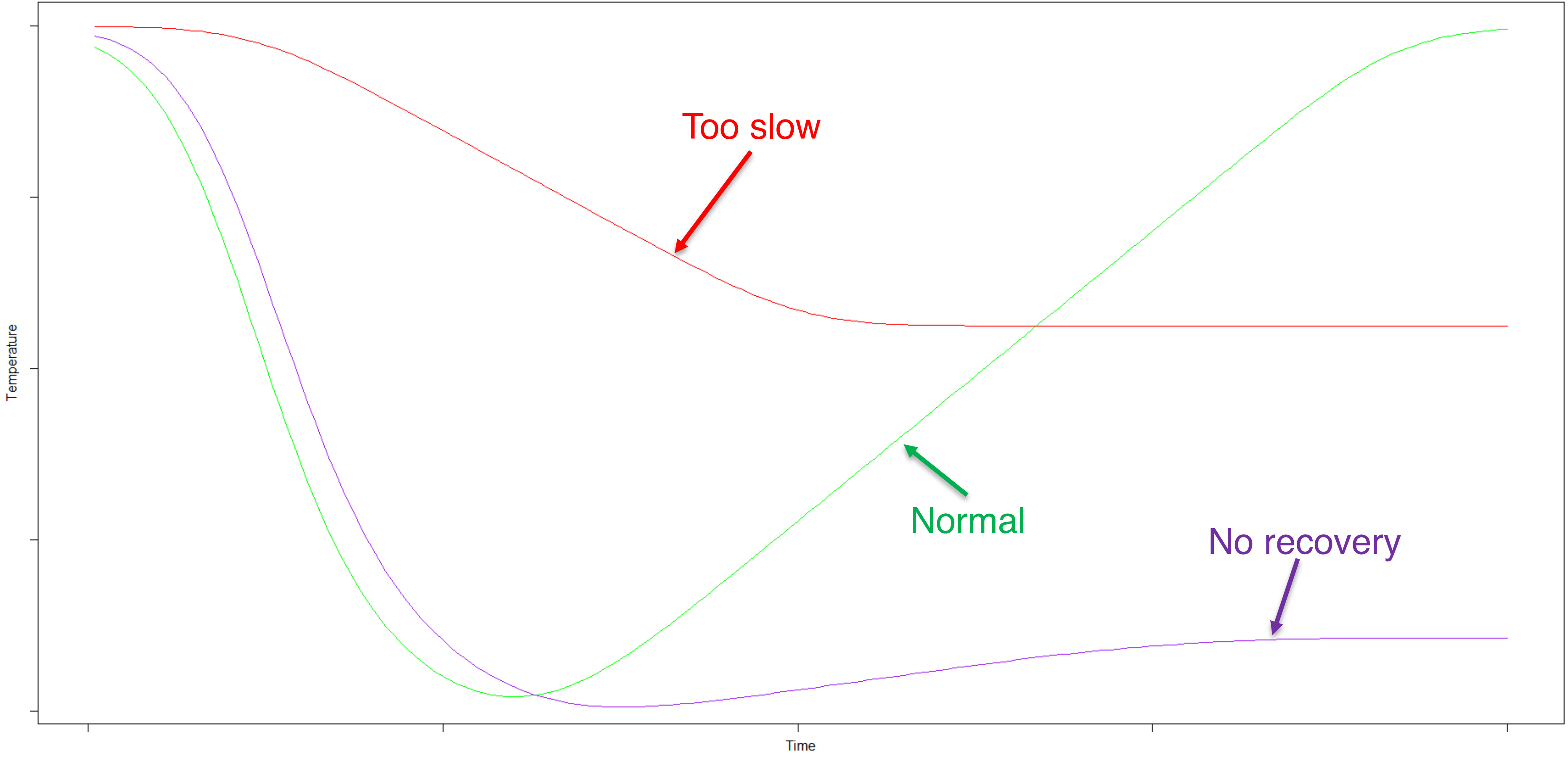
YOUR UTILITY IT PLATFORM

Business case – Laser refills

- One of our clients produces machines of which one important component is a high-power CO_2 laser
- The CO_2 has to be refilled periodically, which causes the temperature to quickly go down, then slowly get back up to normal → this is the refill signature
- In case the refill goes wrong, this could cause downtime

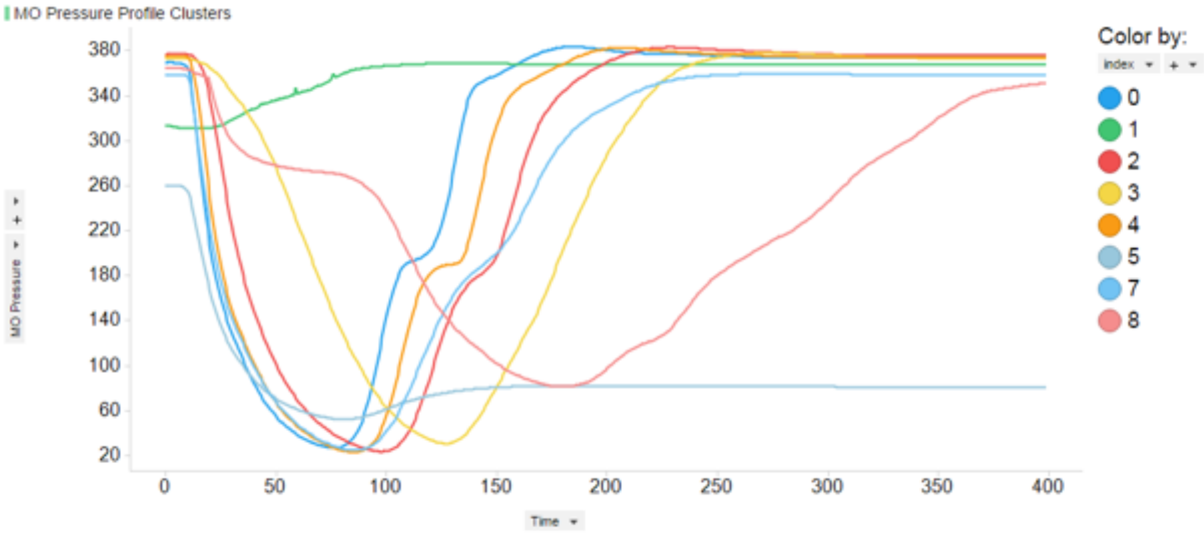
How can we automatically check each refill and fix it if it went wrong?

Business case – Laser refills



Kmeans clustering

- 9 different types of signatures are labelled by a domain expert
- Each new refill signature (time series) is assigned to a cluster
- Engineers are alerted if a signature falls within an incorrect cluster



cluster	NumFiles	%
0	18,088	29.8 %
1	1,948	3.2 %
2	6,463	10.7 %
3	1,379	2.3 %
4	23,356	38.5 %
5	483	0.8 %
6	1	0.0 %
7	8,527	14.1 %
8	381	0.6 %
Grand total	**60,626**	**100.0 %**

The gain of parallelism

	Current WoW	Central Data Lake
Ingestion	**ETL process (Local)** 12.5h	**Source to HDFS (Transatlantic)** 60m*
Preparation & Analysis	**Clustering Analysis** 5.5h**	**Clustering Analysis** 15m***

* Does not include time to zip files
** Extrapolate from 2 lasers, ignores memory swapping (exponential increase)
*** 30 executors, 8GB each

Lessons learned

- You need more than just data and a model:
  - Security
  - Development standards
  - Operational processes and global support

Log classification

Text mining & boosted decision trees

YOUR UTILITY IT PLATFORM

Business case – log classification

- An IT environment generates thousands of logs per day
- Most of them are not of interest, but you don't want to miss the important ones
- On the other hand, you can't go through them all

How can we automatically check each log and only filter out the relevant ones?

How it works

- We start with a training set with *(classified)* textual data:
  - Many logs and how severe they are / what category they belong to

```
classlabel,severitylabel,datehour,message,source
Functional,1,0,AlarmActionTriggeredEvent Appliance Management Health Alarm ,vcenter
Storage,2,0,AlarmActionTriggeredEvent Cannot connect to storage ,vcenter
Unclassified,0,0,AlarmActionTriggeredEvent Consolidation Needed ,vcenter
Functional,1,0,AlarmActionTriggeredEvent Data Service Health Alarm ,vcenter
Storage,2,0,AlarmActionTriggeredEvent Datastore OverAllocation ,vcenter
Functional,1,0,AlarmActionTriggeredEvent Health status changed alarm ,vcenter
Functional,0,0,AlarmActionTriggeredEvent Host IPMI System Event Log status ,vcenter
Functional,1,0,AlarmActionTriggeredEvent Host connection and power state ,vcenter
```

- Leveraging text mining techniques, the algorithm (tm/tidytext-package):
  - Strips 'noise' in the text
    - Punctuation, stop words
  - Makes all text lowercase
  - Stems words to reduce variability
  - Creates its own 'dictionary' (Document Term Matrix)

	action	alarm	alarmactiontriggeredev	alarmclearedev	alarmemailcompletedev	alarmemailfailedev	alarmsnmpcompletedev	alarmstatuschange
1	0	1	1	0	0	0	0	
2	0	0	1	0	0	0	0	

How it works

- Machine learning algorithm classifies new data based on the created DTM
- The classifier contains two decision tree models:
  - For **category**
  - For **severity**
  - One model's outcome is input for the other ('model ensembling')

How it works - example

Mine the data for relevant info:

Log	Severity	Category
AlarmActionTriggeredEvent Host connection and power state	2	Network
AlarmActionTriggeredEvent Host hardware fan status	1	CPU
AlarmEmailCompletedEvent connection and power state	0	Network

How it works - example

Mine the data for relevant info:

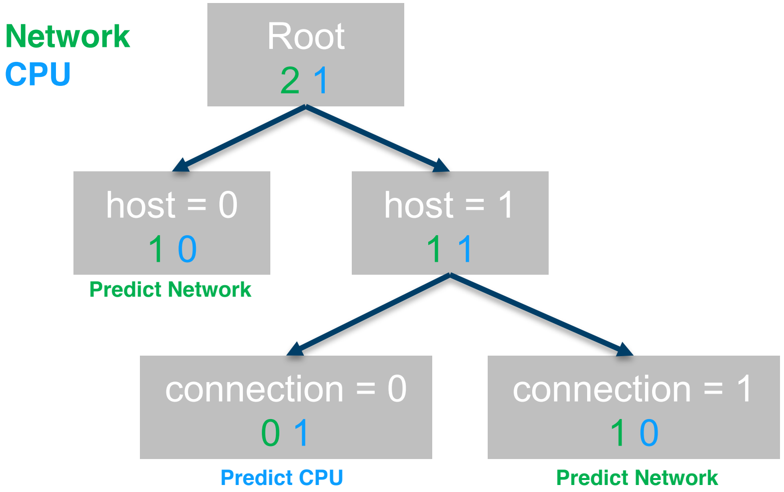
Log	Severity	Category
alarmactiontriggeredevent host connection power state	2	Network
alarmactiontriggeredevent host hardware fan status	1	CPU
alarmemailcompletedevent connection power state	0	Network

alarmactiontriggeredtvent	host	connection	power	state	alarmemailcompletedevent	hardware	fan	status	Severity	Category
1	1	1	1	1	0	0	0	0	**0**	**Network**
1	1	0	0	0	0	1	1	1	**1**	**CPU**
0	0	1	1	1	1	0	0	0	**0**	**Network**

How it works - example

Build the first tree based on the DTM for classifying category:

alarmactiontriggeredtvent	host	connection	power	state	alarmemailcompletedevent	hardware	fan	status	Severity	Category
1	1	1	1	1	0	0	0	0	**0**	**Network**
1	1	0	0	0	0	1	1	1	**1**	**CPU**
0	0	1	1	1	1	0	0	0	**0**	**Network**



How it works - example

A new log comes in:

Log Content
AlarmEmailCompletedEvent host hardware fan status

Text pre-processing

Log Content
alarmemailcompletedevent host hardware fan status

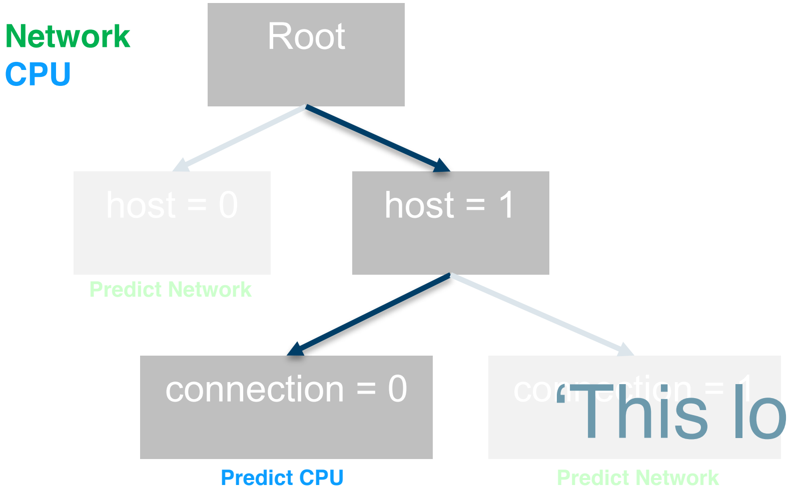
DTM

alarmactiontriggeredtvent	host	connection	power	state	alarmemailcompletedevent	hardware	fan	status
0	1	0	0	0	1	1	1	1

How it works - example

The new log goes through the model:

alarmactiontriggeredtvent	host	connection	power	state	alarmemailcompletedevent	hardware	fan	status
0	1	0	0	0	1	1	1	1



'This log is about CPU'

Lessons learned

- Data is cheap, labels are expensive
- Feedback mechanism
  - Make it simple by embedding it into business process / software
- Efficiently store dictionary (remove sparse terms)
- “Simple” vs “Complex” modelling
  - *Bag of words; n-grams; word2vec*
    - *KISS; “If it works it ain't stupid”*
  - *Extreme gradient boosting vs. gradient boosted*
    - *Use the industry best practice →*
      Xgboost used as a regularized model to control for over-fitting

Break

YOUR UTILITY IT PLATFORM

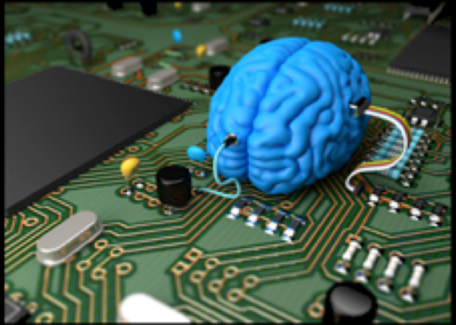
Deep learning applied in projects

Deep neural networks

YOUR UTILITY IT PLATFORM

Deep Learning

What society thinks I do



What my friends think I do

What other computer scientists think I do

What mathematicians think I do

What I think I do

```
In [1]:
import keras
Using TensorFlow backend.
```

What I actually do

Case

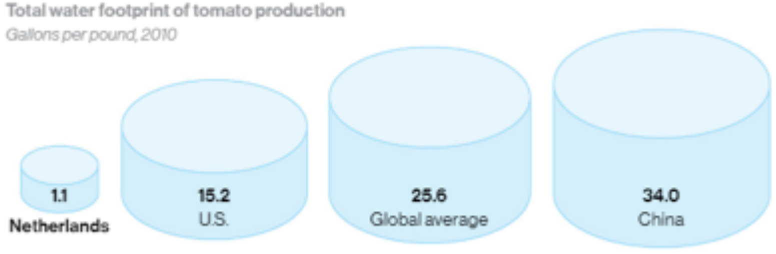
- Challenge
  - Feed a growing population using less resources (energy, water, land).

Challenge

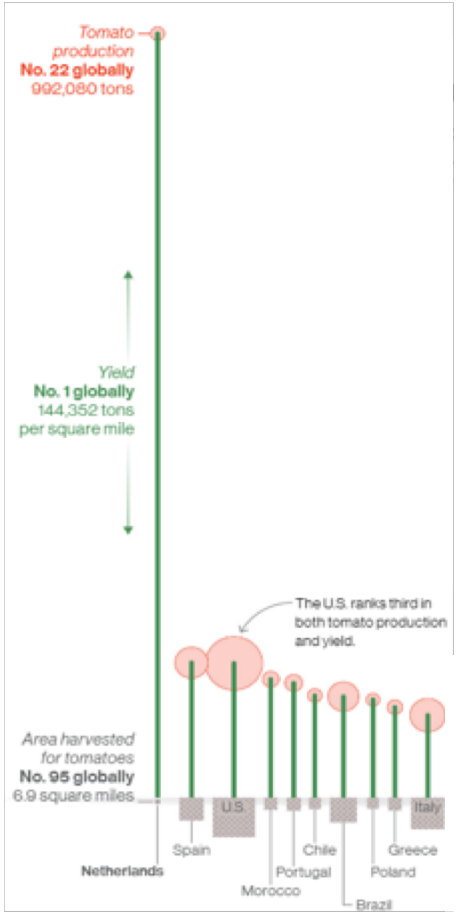
- Feed a growing population using less resources (energy, water, land).

The Netherlands

- Big exporter of seeds, fruits and vegetables.
- **World leader in efficiency.**



*source: National geographic



PoC goals

- Automatically tell what species a seedling belongs to
- Predict early which plants are worth investing in

- The challenge is, young plants tend to look a lot like each other..

Common Chickweek

Fat Hen

Loose Silky

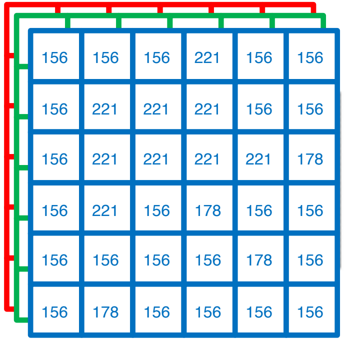
Scentless Mayweed

Small-flowered Cranesbill

Approach – Architecture of neural networks

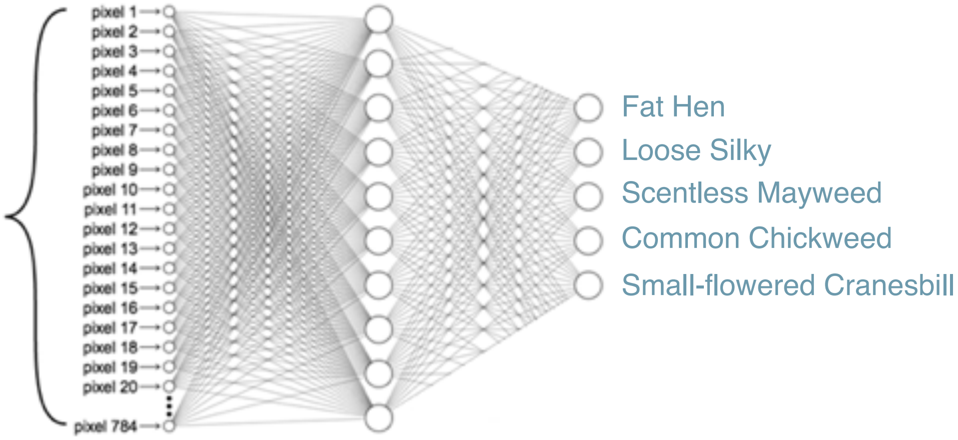
Scentless Mayweed

256x256 image



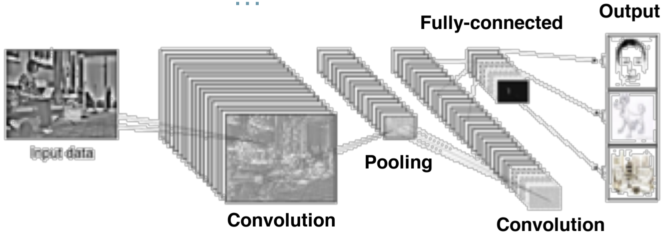
Input layer

Output layer

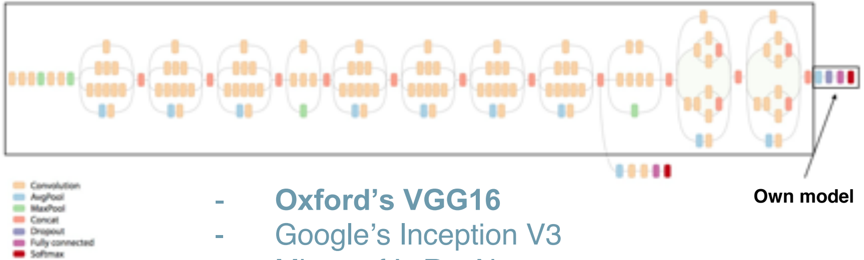


1. Simplest model
- One fully connected layer

2. Adding more layers
- Convolutional layers
- Pooling layers
- …



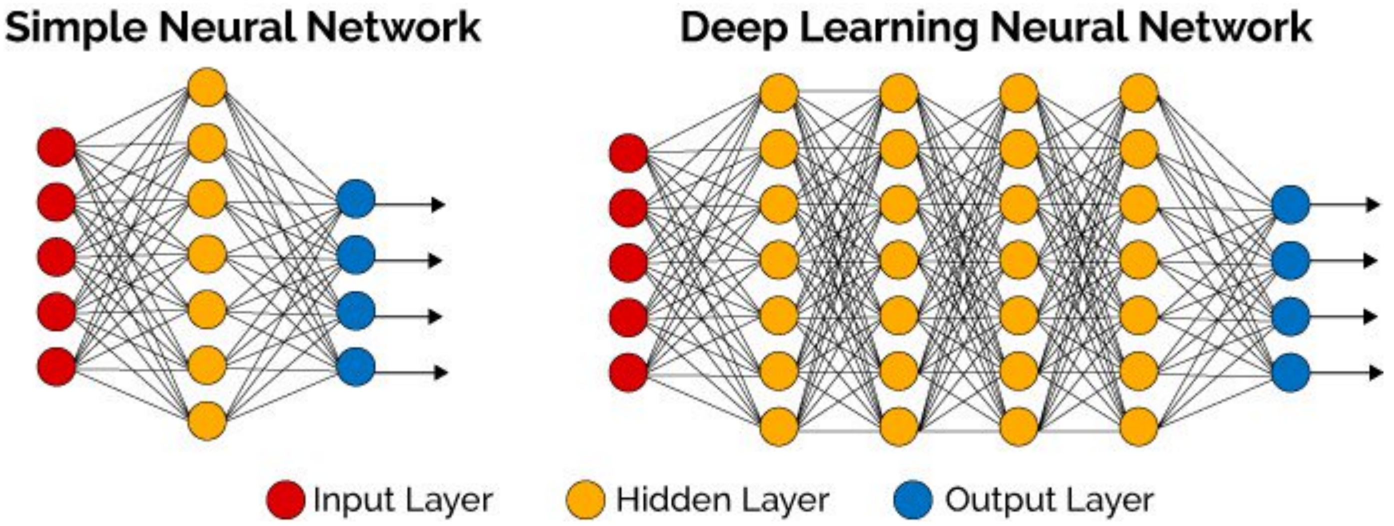
3. Use state-of-the-art pretrained network



- **Oxford's VGG16**
- Google's Inception V3
- Microsoft's ResNet
- MatLabs' AlexNet

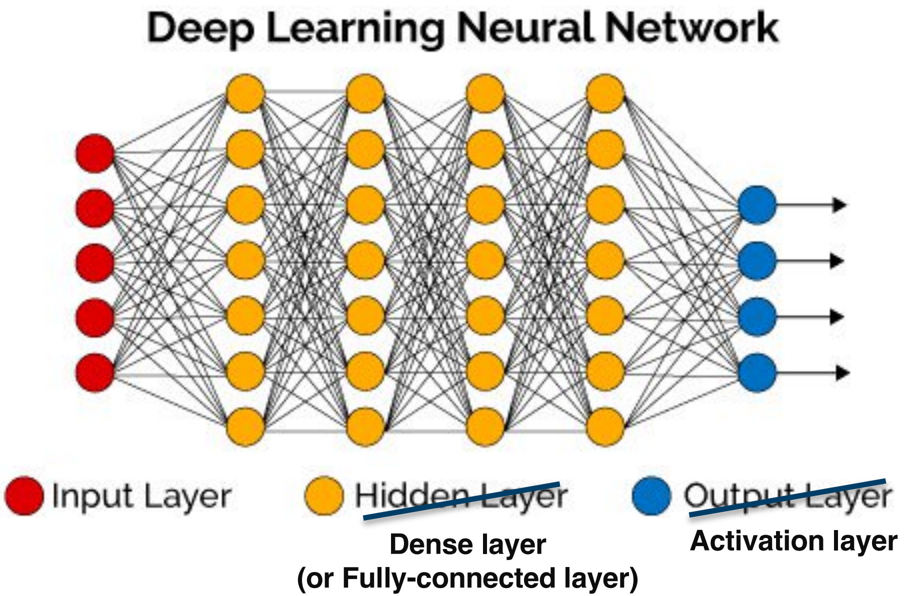
From NN to DNN

- Adding more ‘Hidden’ layers, makes a NN ‘Deep’
- More layers can make a network better if data gets more complex, varied, and/or increases in volume
- There are many types of hidden layers, that can all be combined
- Different data requires different architectures, with different layers

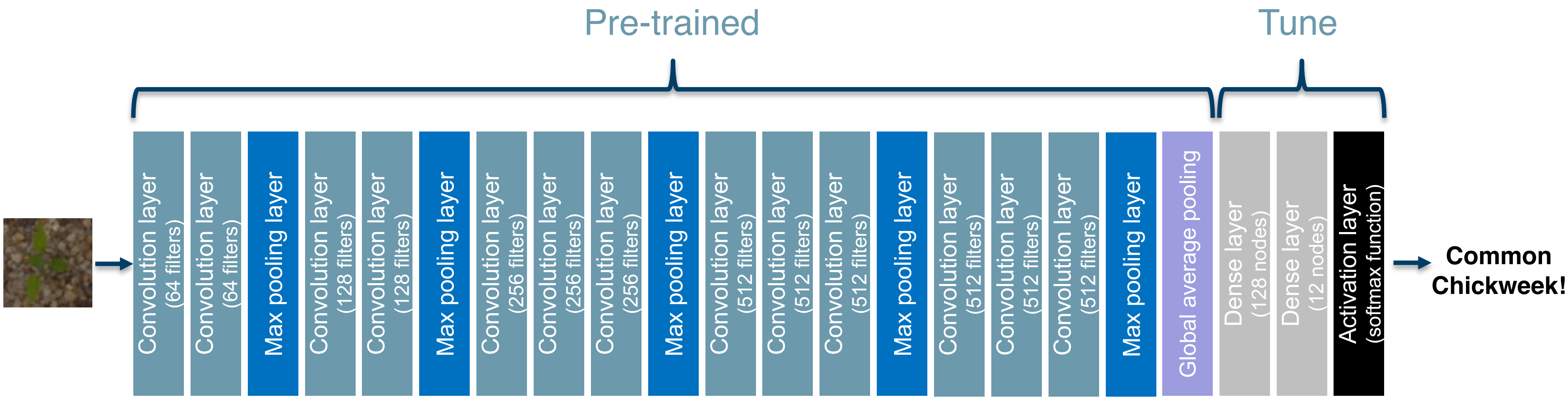


Input, dense, activation layers

- You already met these, only by a different name



Model architecture



The model is based on the VGG16 architecture, from the Visual Geometry Group of the University of Oxford.

Outcome

Training data:

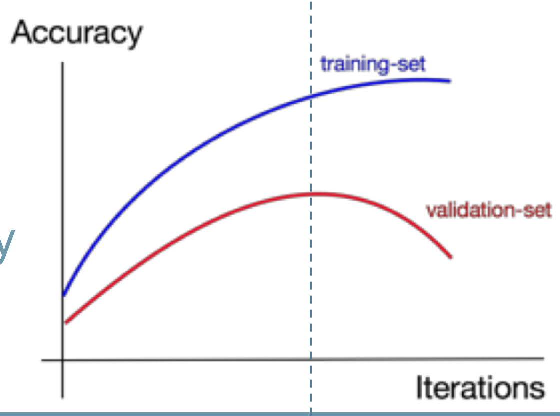
- ~6500 images of 12 different species
- Each image is labelled with its actual species

Validation data:

- We hold back 5% of the images, randomly selected. They will not be used for training
- Validation data is used to test the accuracy of the model and test for possible overfitting

The training images are fed to the model over and over again, until an optimum is found.

Our model managed to classify ~95% of the images correctly



Lessons learned

- Depending on the complexity or variation of the data, more layers might be needed for better results
- More layers = more compute power required
- Transfer learning = use readily available models
  - Leverage pre-trained model’s weighted layers has advantages:
    - No need for large training set
    - No large-scale computational set-up required
  - Only need to add dense/softmax-classification layer

Solution implementation

How do we embed Data Science solutions into the business

YOUR UTILITY IT PLATFORM

Problem

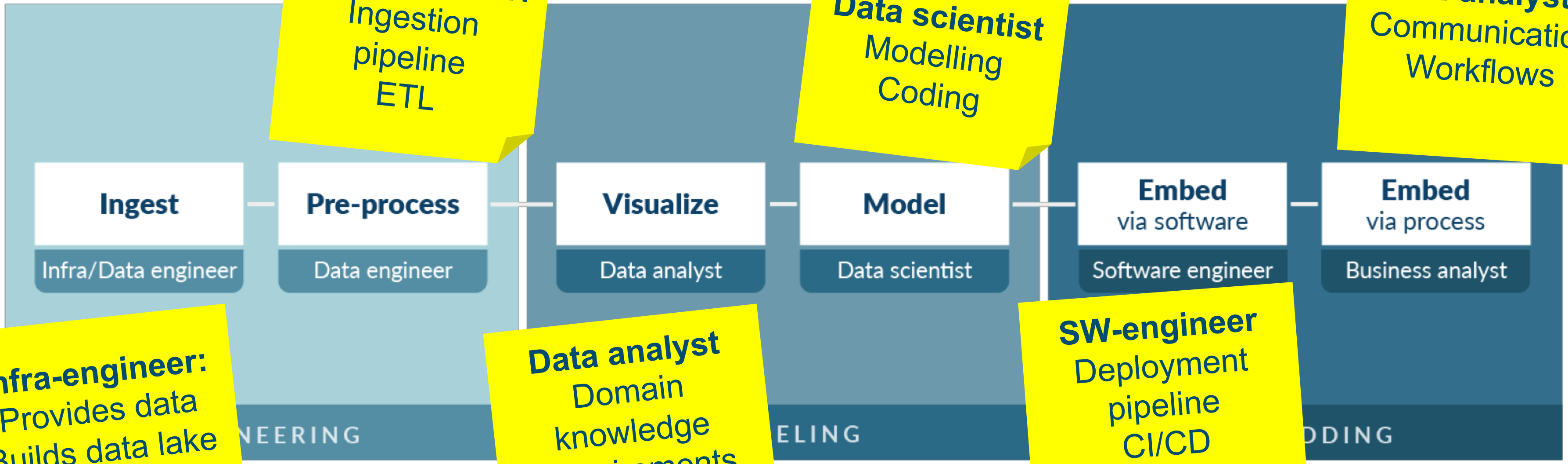
85% of ML models is getting to Production

*source: Gartner

How do we make sure solutions are being embedded into the business?

Itility Data Factory Flow

Different role are required to make a solution production ready



Data-engineer:
Ingestion
pipeline
ETL

Data scientist
Modelling
Coding

Biz-analyst
Communication
Workflows

Infra-engineer:
Provides data
Builds data lake

Data analyst
Domain
knowledge
Requirements

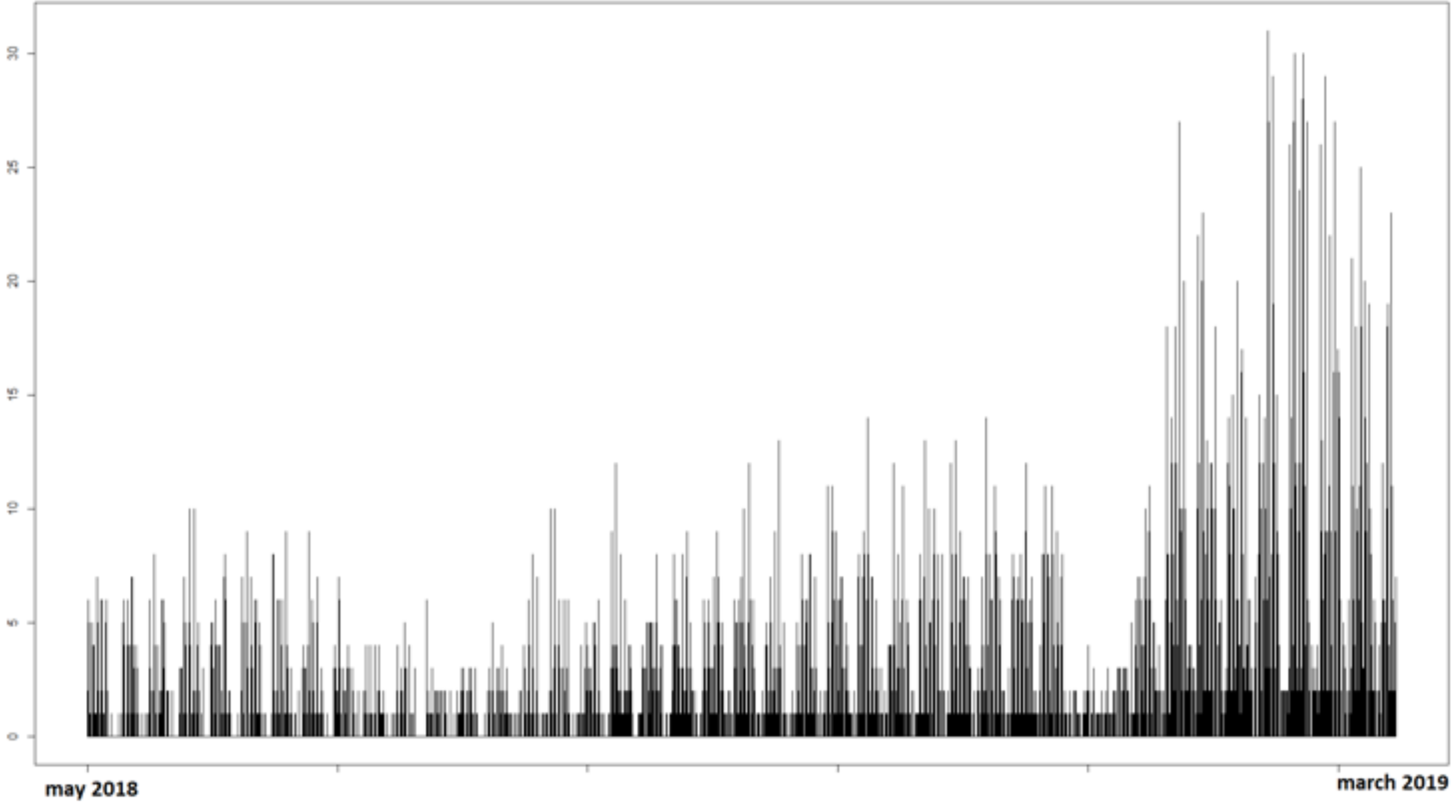
SW-engineer
Deployment
pipeline
CI/CD

Business case - Amber

- Car-sharing company
- All electric BMW i3's
- Founded in Eindhoven by three TU/e students in 2017
- USP: guaranteed availability
  - Planners make an estimate per hub, per shift
  - Students drive around to redistribute cars between low-, and high-demand hubs
- We are helping Amber with a demand prediction model

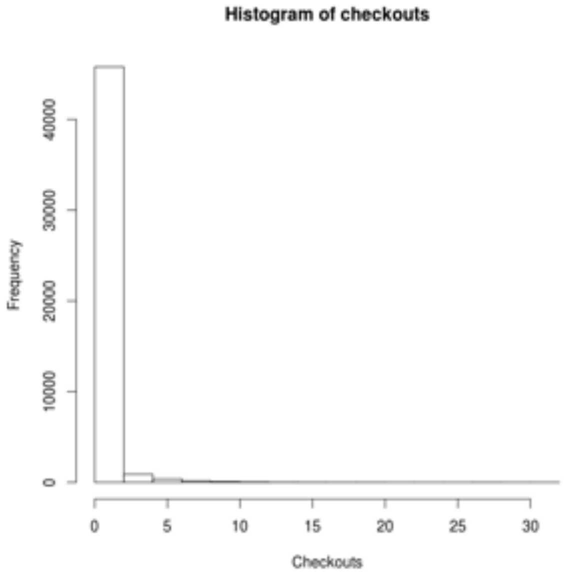
Challenges

- They grow pretty fast, making historic data quickly become less relevant

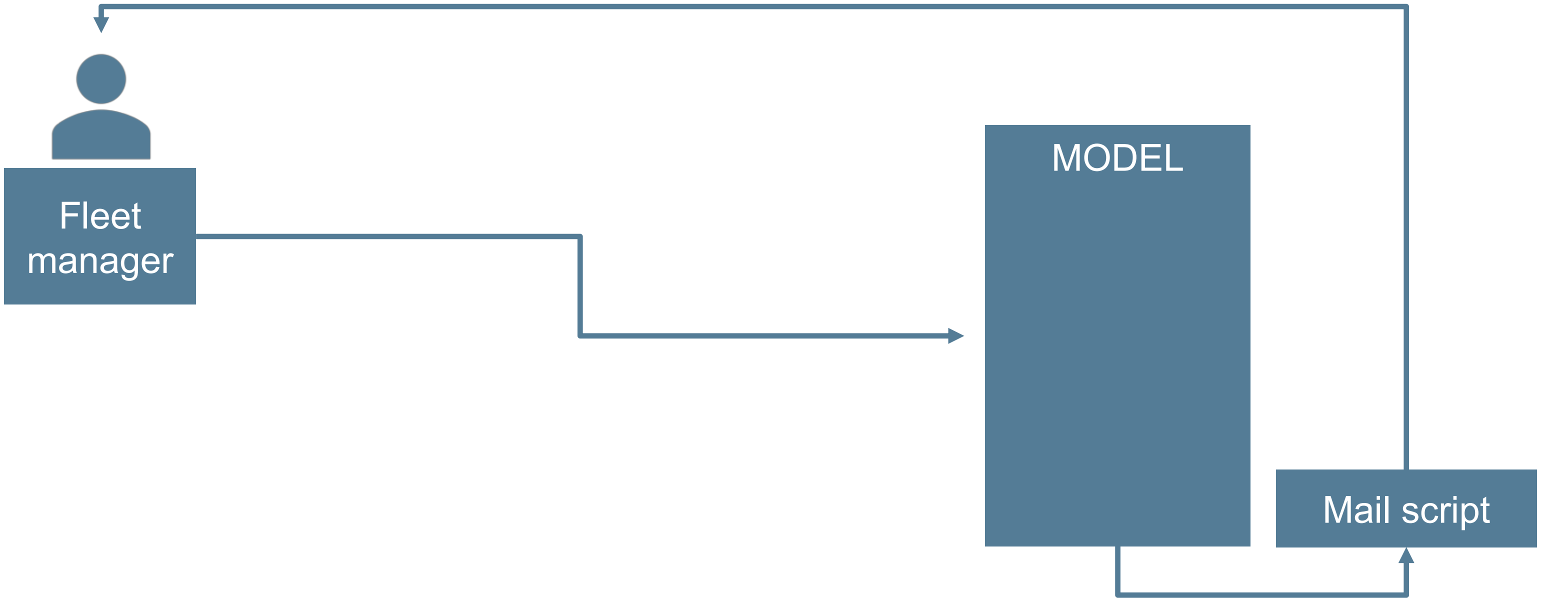


Model

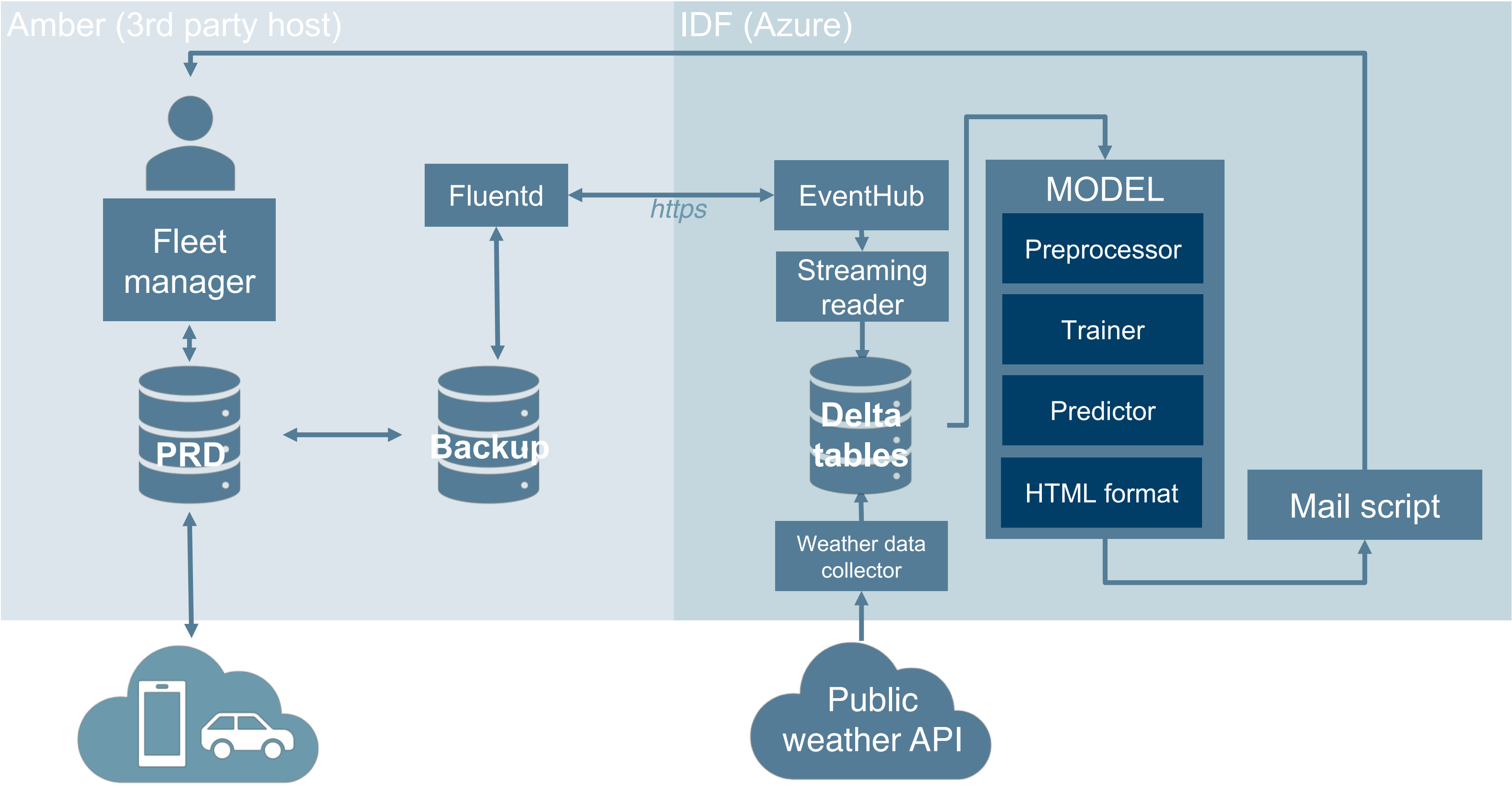
- Prediction per hub / per 6 hours (00, 06, 12, 18)
- XGBoost regressive model
- Challenge: the # day-parts the checkouts are zero:



Setup v1.0



Setup v1.0



Amber (3rd party host)
IDF (Azure)
Fleet manager
Fluentd
https
EventHub
Streaming reader
MODEL
Preprocessor
Trainer
Predictor
HTML format
PRD
Backup
Delta tables
Mail script
Weather data collector
Public weather API

Lessons learned

- Getting in live data, keep training a model, and implementing predictions in the workflow can be challenging
  - External dependencies: how/where is the data stored?

Wrap-up

YOUR UTILITY IT PLATFORM

Lessons learned

- Create a data science workflow which involves iterations & best practices
- Communication is key, make sure you speak the same language
- *“Don’t reinvent the wheel, just realign it”*
- Data is cheap, labels are expensive
- It is not only about training a model, implementing solutions in the business can be challenging

Try it yourself!

Kernel example of VGG16-model with Keras in R:

- https://www.kaggle.com/dkoops/keras-r-vgg16-base

Participate in hackathons:

- https://www.meetup.com/NL-Itility-Hackabrain/

Questions?

Thank you for your attention

You can contact us at:

YOUR UTILITY IT PLATFORM

lars.van.geet@itility.nl; kevin.schaul@itility.nl