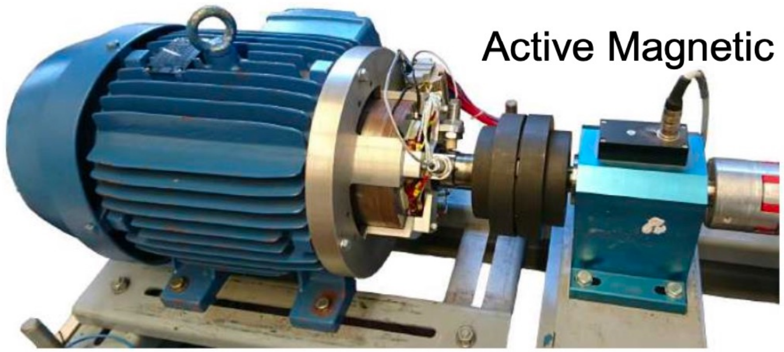
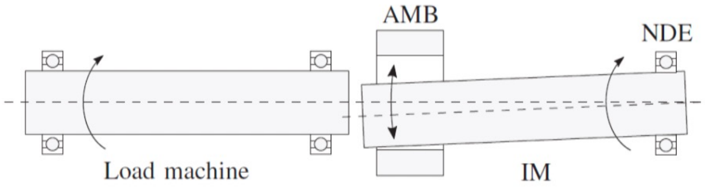
GHENT UNIVERSITY

ML CHALLENGE 2023

Misalignment of induction engine



Active Magnetic Bearing



Misalignment Fault

Misalignment between two axes of induction engine must be detected (can lead to damage).

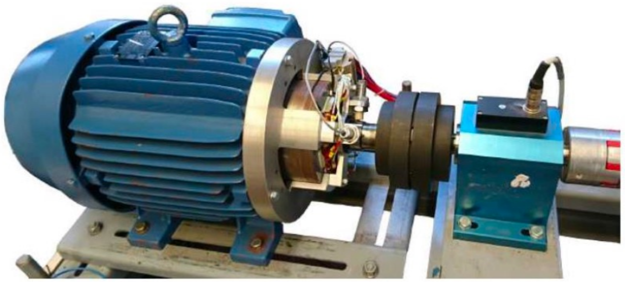
Research topic:
Can misalignment be quantified automatically from limited number of (cheap) sensor outputs

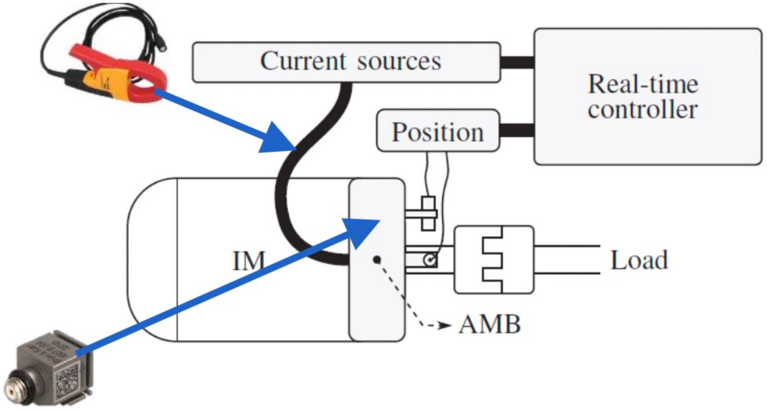
Here:
Emulated data, for total misalignments **between -0.5 and +0.5 (steps of 0.1)**
-> strictly speaking: **ordinal regression**
(but only because of the experimental setup)

Data

Available Signal Sampling hardware:

1. Current Clamp Low Resolution
2. Current Clamp High Resolution
3. Acceleration Signal in Z direction

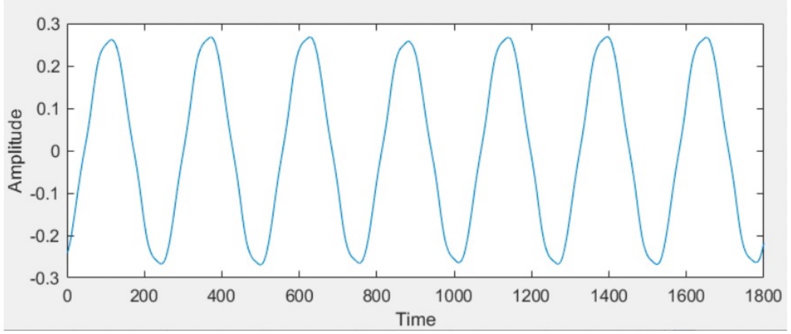




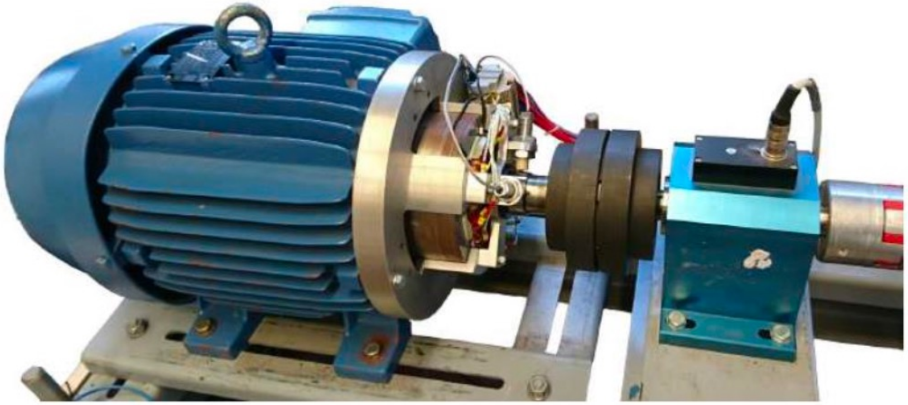
Application of Current sampling hardware, Sampling Rate 12800 Hz

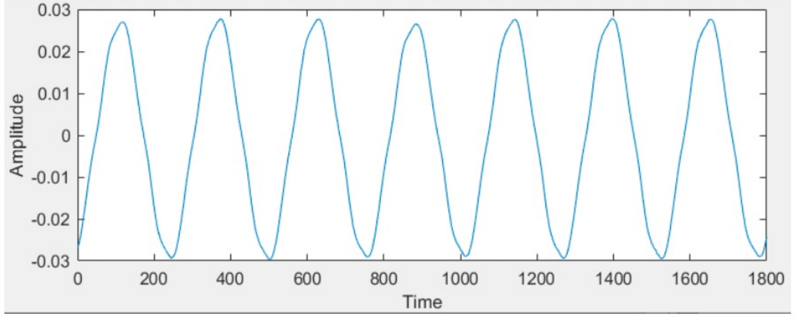
Data

Result of sampling:
- Collection of CSV files
- Duplicated dataset for testing/validation
- Inside the file 3 vectors
  - MotorCurrent: low/high Resolution and Acc
  - Duration of recordings is 1 sec.

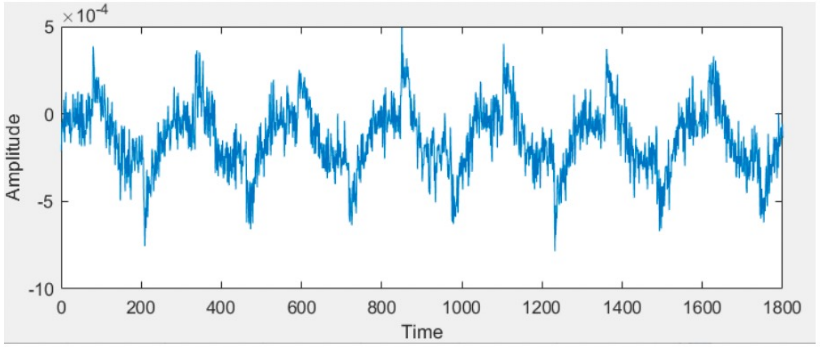


Example of the HighRes current signal





Example of the LowRes current signal



Example of the Acceleration signal

Labels: MAE (mean absolute error)

$$\frac{1}{n}\sum_{i=0}^{n}|\hat{y_i} - y_i|$$

Task

Step 1: LOOK AT YOUR DATA (**LAYD**)!

- Visually inspect data,
- Try to identify how data looks different for different labels
- Try to identify how data looks different for same label

Step 2: features!

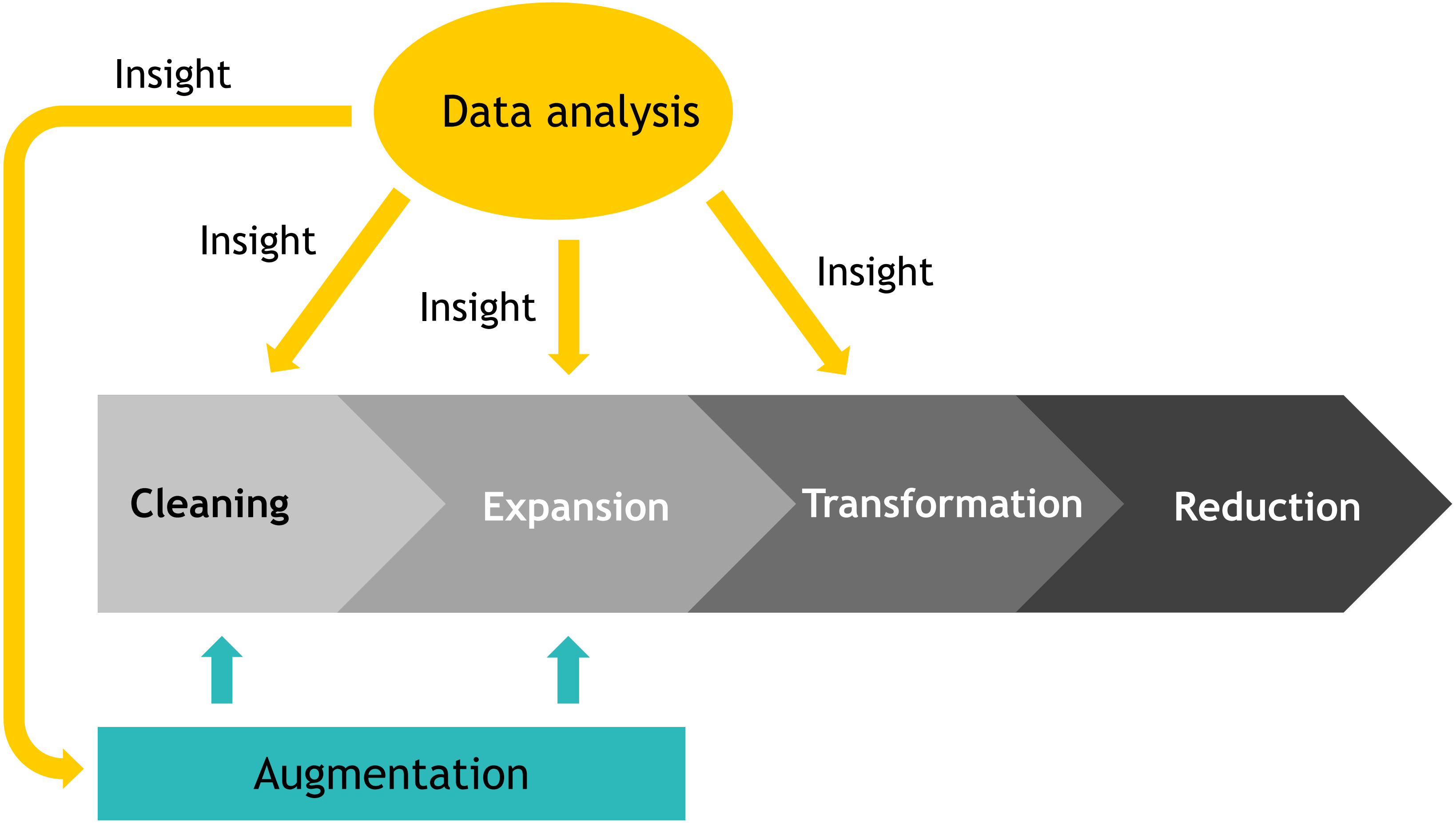
- Each sample is 1 time series window: need to extract features!
- General knowledge:
  which types of features can we use (use libraries, e.g. mentioned in class)?
- Domain knowledge (and data analysis):
  which types of features could be useful for this task? (from library, from insight)

Step 3: validation strategy??

Step 4: well-tuned **linear model**

Step 5: any model you like!

Remember the data pipeline



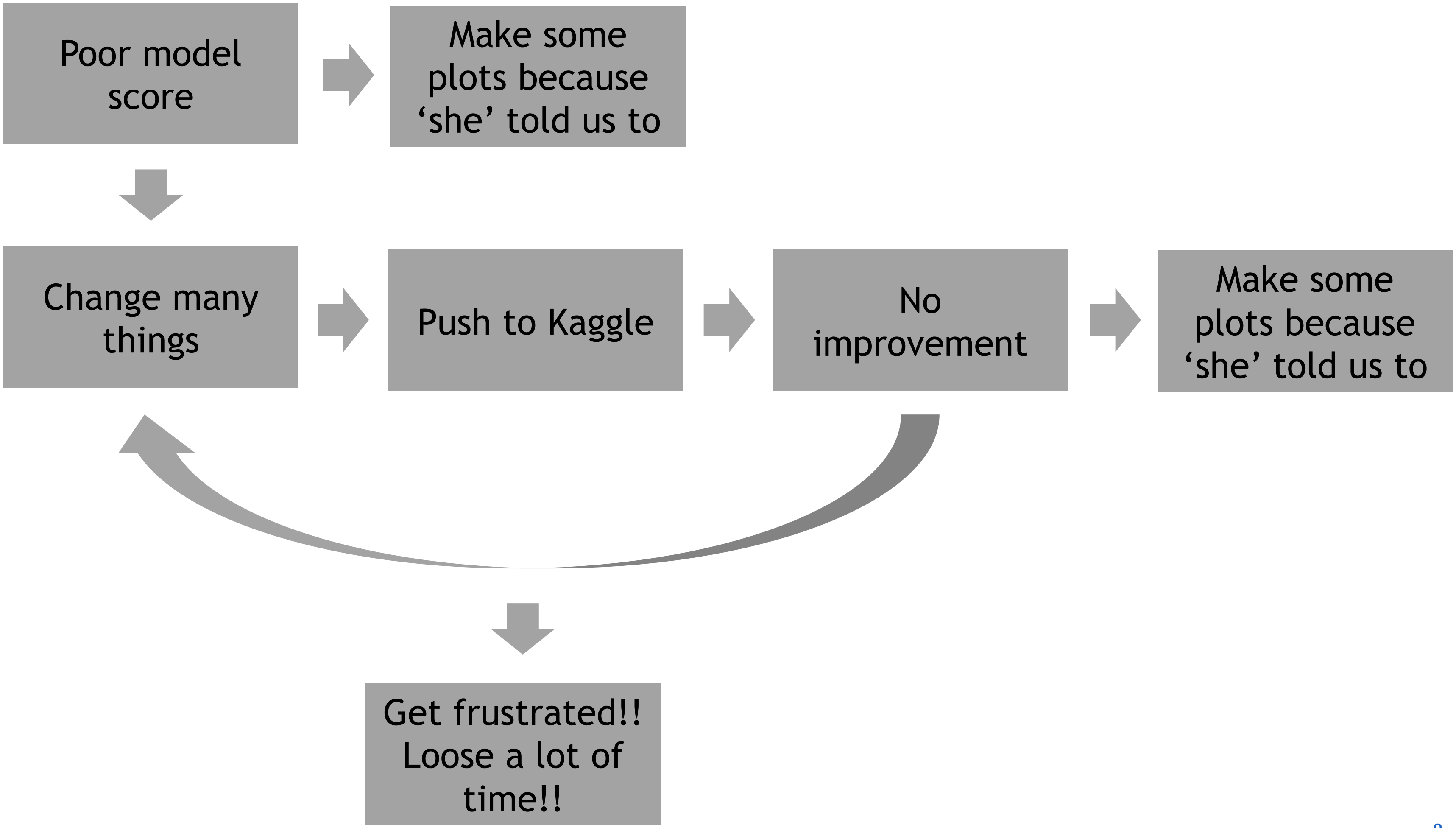
Main learning goal:
analysis-driven iterative model improvement

(i.e.: the stuff we really could not squeeze into a lab)

- Understand your data
- Analyse model properties and errors: visualise in any useful way
  - **gaps, learning curves and fold scores!**
  - error distribution // confusions (can look at this both ways)
  - look at data for (samples or classes or label ranges with) best and worst predictions
  - which features are used (and which are not)?
  - which information may be missing?
- Try to identify where to invest your effort next
  - better features (very often)
  - better data pipeline (very often)
  - more powerful model (**only when everything else is OK**)
- Change one thing at the time, keep a logbook of what you have tried and what the result was
- Try to understand why some thing (you thought was great) didn't help

The importance of methodology

In first attempts, we often see this:



Poor model score

Make some plots because ‘she’ told us to

Change many things

Push to Kaggle

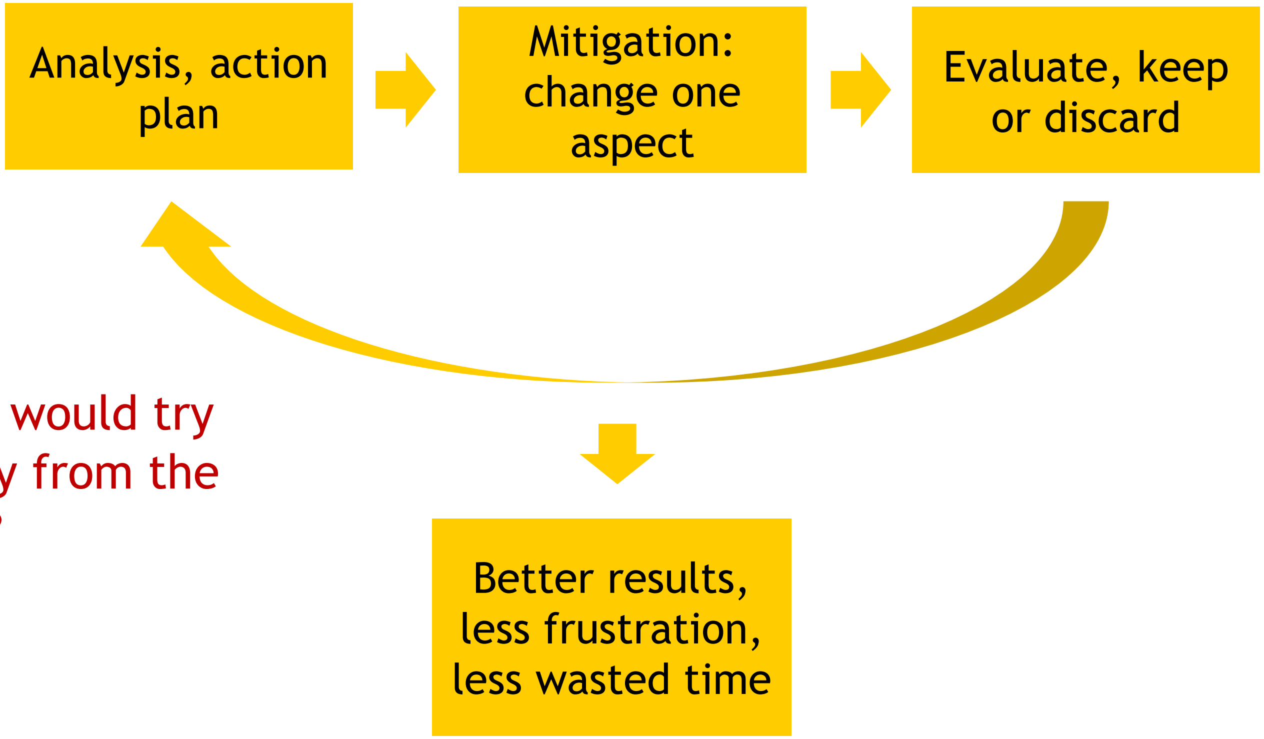
No improvement

Make some plots because ‘she’ told us to

Get frustrated!! Loose a lot of time!!

Systematic approach

Gradually, this (more or less) evolves into something more like this:



Analysis, action plan

Mitigation: change one aspect

Evaluate, keep or discard

Better results, less frustration, less wasted time

So, what if you would try to do it this way from the start?

The aim:
data- and error analysis driven iterative model improvement!

Machine learning = research!

Do not randomly try out stuff, but

i. think,

ii. formulate hypotheses
(e.g. why it doesn’t work well enough)

iii. check the result of your fix (did it help?)

iv. and iterate

Use a data-driven approach

Analyse the hell out of your data, your model and its mistakes

Usually, time invested in analysing and improving the data pipeline yields faster and larger improvements than focusing on the model!!

Discipline:

Keep a logbook of what you tried and what the results were

“I had a score of XXX once, but I cannot reproduce it!!”

Timeline

Weeks 4 and 5:

- free exploration, only based on validation scores (data is on Ufora)
- alone or in groups (free to think together, not to broadcast solutions!!)
- End of week 5: decide whether you will participate, make groups
- if you do: engagement towards your team members (don't 'drop' them)

Weeks 6 & 7:

- work in groups on data exploration, feature extraction, validation strategy, linear model
- group support session in week 6 (Monday)

Weeks 8-11:

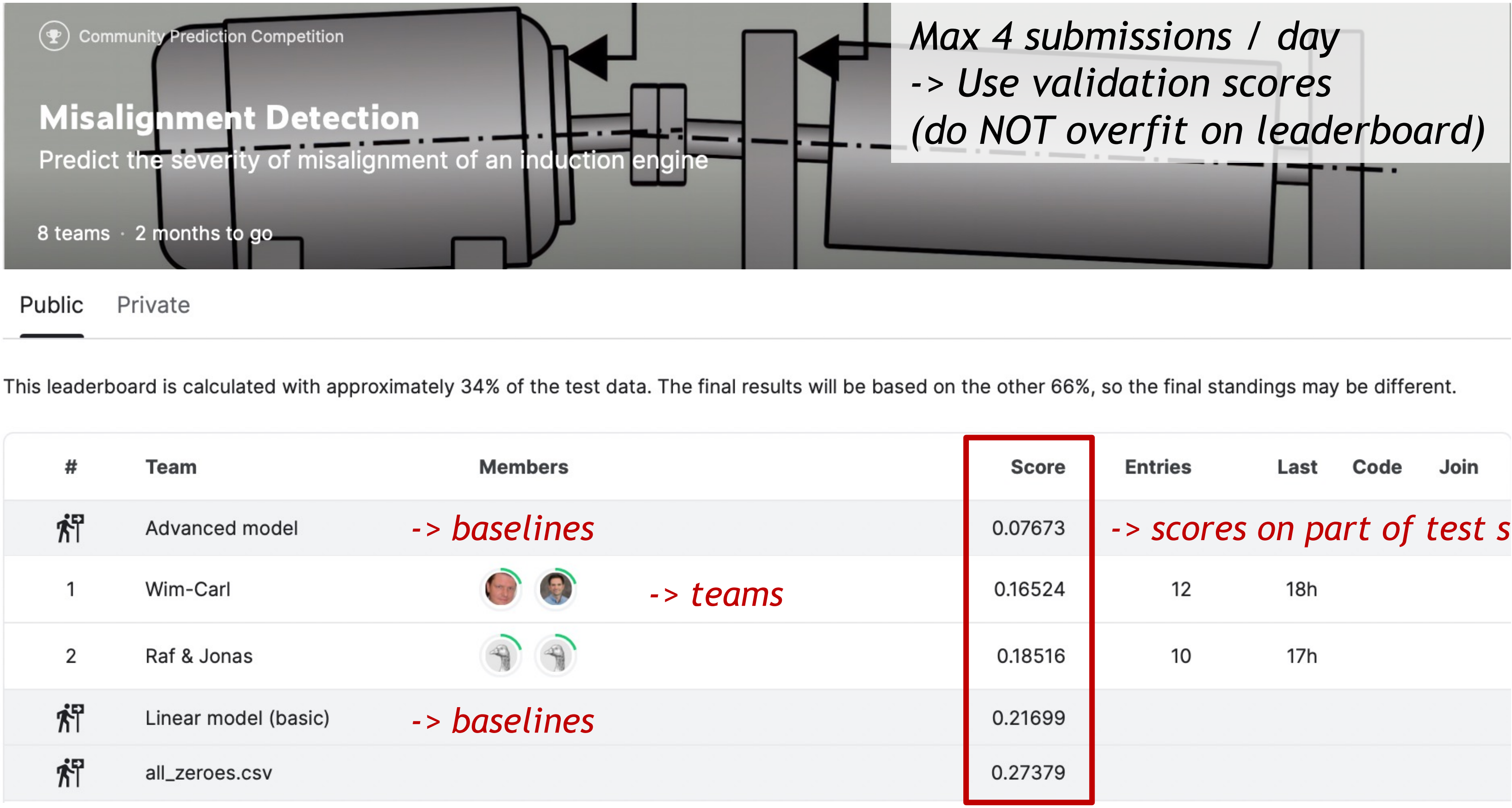
- Beginning of week 8: finalise group formation (ideally 3 people, max 4)
- Kaggle entry -> check validate-test gap with best model up to that point!!
- work in groups, compete with other groups – Kaggle closes end of week 11

Week 12: report& hand-in of code

- presentation + Q&A (scheduled based on participants and availabilities)
- **evaluation:** mainly based on analysis, understanding & iterative improvement methodology (final score matters, but methodology matters more)!!

Support & ‘peer (leaderboard) feedback’

- Through Teams: reply ASAP (but be sure to tag us)
  -> will get per-group private channels after group formation
- In-personsupport per group possible **during Lab hours**
  (from week 8 onwards: by appointment, since next labs are coached by different team)



Community Prediction Competition

Misalignment Detection

Predict the severity of misalignment of an induction engine

8 teams · 2 months to go

Max 4 submissions / day
-> Use validation scores
(do NOT overfit on leaderboard)

Public Private

This leaderboard is calculated with approximately 34% of the test data. The final results will be based on the other 66%, so the final standings may be different.

#	Team	Members	Score	Entries	Last	Code	Join
	Advanced model		0.07673				
1	Wim-Carl		0.16524	12	18h		
2	Raf & Jonas		0.18516	10	17h		
	Linear model (basic)		0.21699				
	all_zeroes.csv		0.27379				

-> baselines

-> scores on part of test set

-> teams

-> baselines