

Start Here

January 26, 2017

0.1 Amazon work sample

Anselme Vignon

0.2 A. pre-data peeking brainstorm

0.2.1 task description

- Dataset
- Features are attributes ==> need to figure out what they could be, and how to interpret each of them
- Each point is a date/device_id. Need to determine the sparsity in both dimensions, to know which modelisation could make sense.
- Is there a feature bias linked to the time at which maintenance occurs ? (e.g. no more data after a failure) this could build a bias in the data.
- Problem
- Failure detection: two signal axis (which device fails, when does a device fail) to consider. Consider mixing two models ? eg:
 - model1 : which device is the most likely to require maintenance.
 - model2 : based on past signals, when is a maintenance the most likely to occur.
- Failure detection: positives could be sparse. to be checked. SVD could be worth something. This is a precision/recall problem.
- On the other hand, no obvious advantage in detecting a false positive over a false negative.
- Infos
- 3D technologies is electronics, the device sends out telemetry: multiple failure modes (wrong telemetry, bad transmission, etc...) ==> we could be detecting multiple failure modes. ensemble models ?

0.3 The Plan

1. [Exploratory Analysis](#)
2. Data shape
3. Labels
4. [features](#)
5. Modeling
6. Decide on a model / a list of models
7. Model(s) optimisation process
 1. Dataset building
 2. First model and calibration
 3. Feature optimisation
 4. Test different Models
 5. TPOT
8. Final test on validation set.

1 1. Exploratory Analysis

1.1 A. Data shape

Notebook : [here](#)

In general:

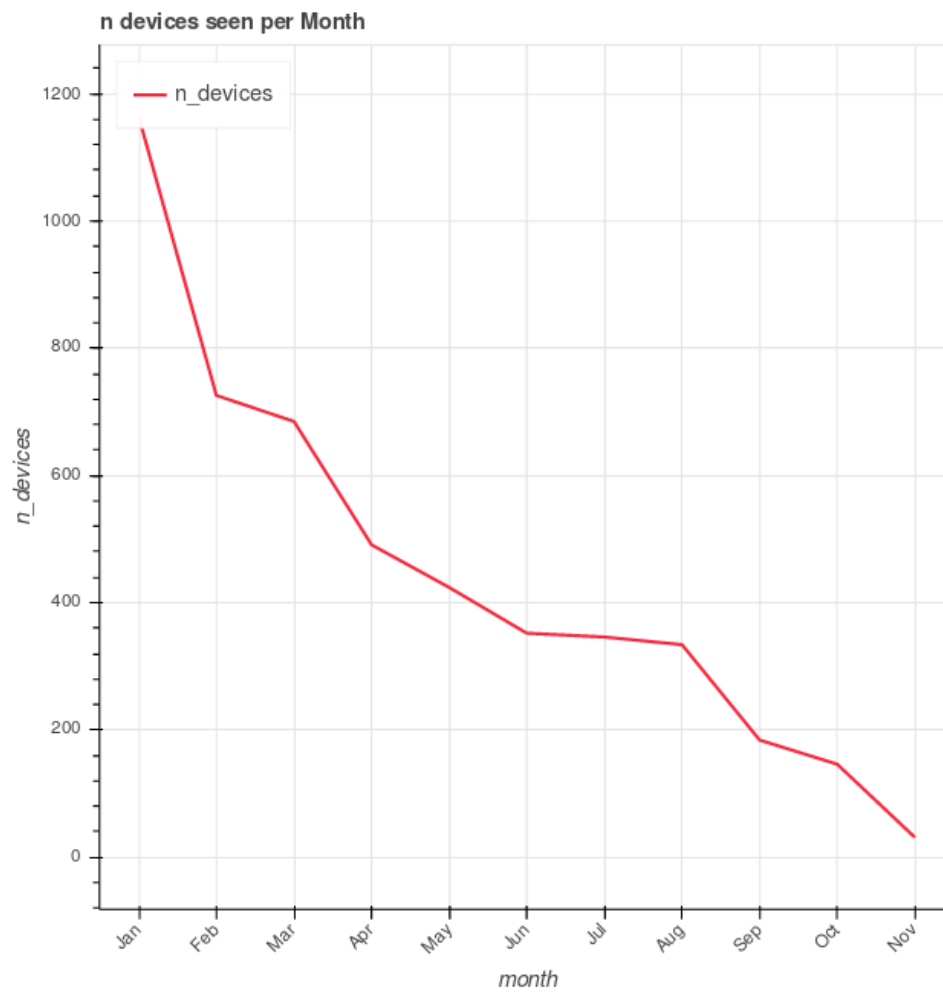
- Each line is indexed as a (device, time)
- 9 attributes (features)
- 1168 devices, 106 failing (small data, few positives)
- Date Range: daily timestamps from 2015-01-01 to 2015-11-02

Devices are not always on !

- Each device is active during a given period, `device_period < dataset period`
- The device 'start_time' is biased toward the beginning of the dataset. (1163/1168 start the fi
- There are much fewer devices at the end of the measurement period (see graph below)

1.2 B. Labels

- 1168 devices, 106 failing (small data, few positives)
- failure ratio : 10%, ~1% per month
- inn fact, the % failures varies month to month, from 5% to 0%



Alt

Weekly view

- Less failures over the weekend: devices are less strained during the weekend ?
- Warning: fewer failures during weekend could also be explained by maintenance hapening only du

weekday	NB failures
0:Monday	27
1:Tuesday	18
2:Wednesday	15
3:Thursday	22
4:Friday	12
5:Saturday	8
6:Sunday	4

Missing data

- During device "lifetime", most devices (1077/1168) have a signal per day, without missing days

Failure mode

- Almost all devices stops measuring after a failure
- Identified a list of devices, which are still measured after having failed.

three hypothesis: - The device is still functional after maintenance - The failure was a fluke - The measurement thereafter are false

=> if we cannot distinguish between these hypothesis, need to remove these devices from the dataset

1.3 C. Features

Using the [feature analyser notebook](#) over each individual attribute.

- attribute 1: no real influence observed
- attribute 2:
 - Higher values on failures
 - Rising front before failures
- attribute 3:
 - Slightly higher for non-failing than for failing
 - Unclear temporal effect
- attribute 4
 - higher values for failing devices
 - Rising front before failures
 - failing frequency peak
- attribute 5 :
 - no clear impact
 - potential peaks when failing
- attribute 6 :
 - unclear effect on value or fronts

- on the other hand, signal on frequency distribution
- attribute 7 :
 - unclear effect
- attribute 8 :
 - idem
- attribute 9
 - frequency distribution ?

Conclusion

- Need to check out for attribute values, derivatives and DFT peaks as features

2 2. Modeling

2.1 A. Decide on a model / a list of models

There seems to be some attributes having an effect on the device failing. What we actually want is a model predicting the day devices are failing.

Also, since what we want is to implement a maintenance model, it would be acceptable, if it is more efficient, to predict devices soon to be failing, for example 7 days before the failure.

ccl: we will build two models: - A model predicting which device will fail at some point - A model predicting which device will fail and when, with an acceptable failure window

2.2 B. Models optimisation process

1. Dataset building
2. First model and calibration
3. Feature optimisation
4. Test different Models
5. TPOT

ML decision diary: [here](#)

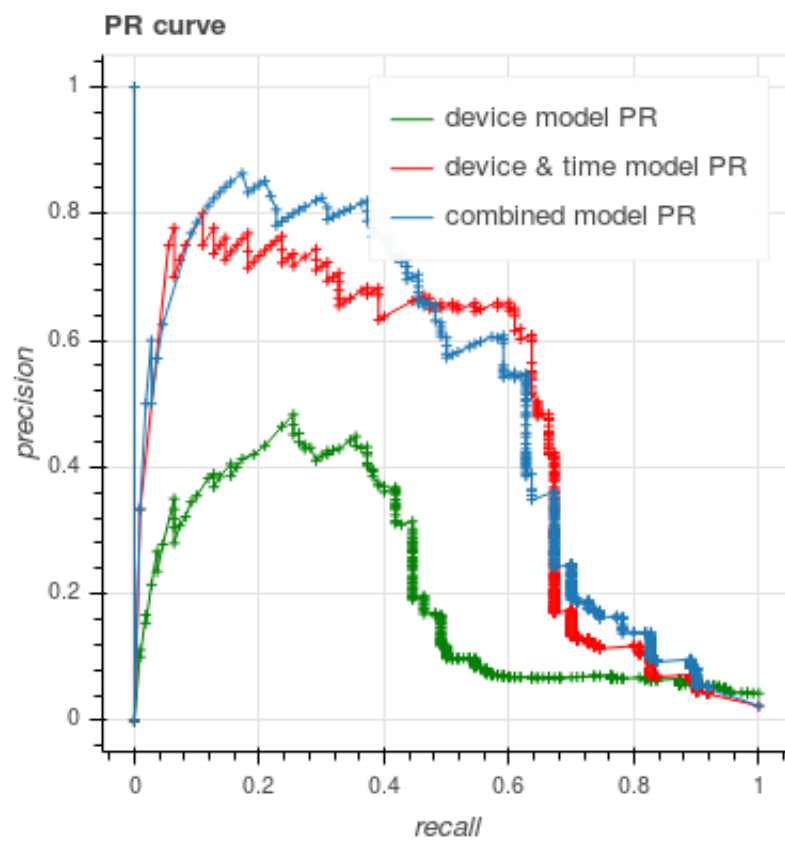
Each models are optimized on different notebooks:

- [device base model](#)
- [device base model and time](#)

2.3 C. Final Test

Test on a validation set, splitted after the device exploration

In []:



Alt