AIRBUS Group Innovations - TX5-MD

Scikit-learn for predictive maintenance at Airbus

AIRBUS GROUP

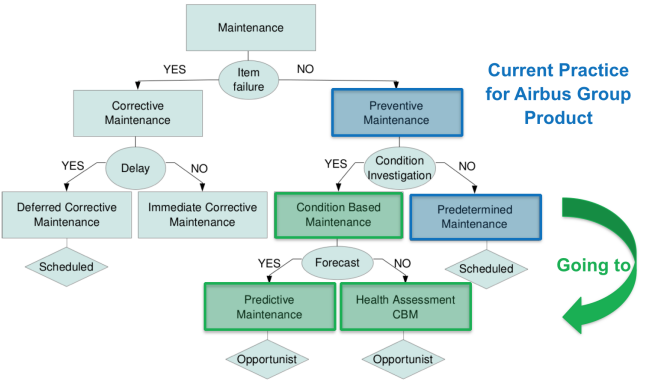
Outline

AIRBUS GROUP

Outline

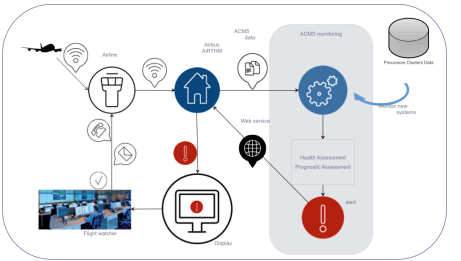
AIRBUS GROUP

Maintenance policy at Airbus



Exemple of AiRTHM

It is an advanced **service** through which A380 and A350 XWB operators can receive guidance on optimized maintenance and real-time troubleshooting actions. **A dedicated 24/7 team investigates and *anticipates* warnings/fault consequences.**



Benefits for the Customers (A/L)

Enhanced engineering advice for complex failure cases

- Ex : Failures having an impact on OI

Reduce maintenance cost

- Ex : **Predictive maintenance advice** avoiding an OI

Benefits for Airbus

Improve OR of our product
Increase learning curve of in service A/C

Setting a new OI Alert Item

Key questions :

- Can we find early symptoms announcing this OI?
- ***What are the A/C monitored parameters which can detect these early symptoms?***
- ***What are the best Indicators to monitor the degradation indicator before failure?***
- ***What are the characteristics of the behavior of this degradation?***
- ***What method do we use for detecting arising of these symptoms?***
- ***What method do we use for estimating Health indicator?***
- What method do we use for estimating Remaining Useful Life (RUL)?
- What are the needed data sets for implementing these methods and V&V processes?

How can we answer ?

Mainly by :

- Expertise
- **Assisted by Data Analysis means**

$\Rightarrow$ **Airbus needs a reference Machine Learning Toolbox**

AIRBUS GROUP

Outline

AIRBUS GROUP

The 0_2 bottles example

System function

The system delivers oxygen to passengers and crew in case of a depressurization of the A/C until it reaches an altitude where air is breathable.

Available data

For safety, the pressure is tested and recorded at each flight (so the pressure decreases at each flight).

The aim is to *prognose* (i.e. to *predict*) when servicing of the bottle should be done and detect faults like leaks, intermittent failures of sensors, etc.

AIRBUS GROUP

The 0_2 bottles example

Remarks

For this example, ***scikit-learn*** have been used for :

- **Assisting the clustering of the different operating modes**
- **Learning classifiers**
- **Predicting the operating mode (being given some features)**

Scikitlearn classifiers are currently tested for A380 fleet monitoring at AiRTHM

(*the software perennity and its compatibility with Airthm platform still has to be assessed before a full deployment*)

AIRBUS GROUP

Outline

AIRBUS GROUP

Main required functionalities

- **Data Munging** (i.e. converting raw data into a format that allows more convenient consumption of the data)
- **Data visualization** for helping in the "understanding" of the data
  - Simple scatter-plots can be very useful for understanding the relationships (dependence) between the data
  - Dimension reduction is often required for the visualization of high-dimensional data
- **Unsupervised Learning**
  - Anomaly detection (for filtering outliers and/or detecting unexpected operating modes)
  - Clustering capabilities (for assisting the identification of different operating modes)
- **Supervised Learning**
  - Classification for predicting on operating mode
  - Regression for predicting a "health status"
- **Easy to use** (quite subjective, it mainly depends on the user)
  - Well documented
  - Possibility of building a graphical workflow

AIRBUS GROUP

Main required functionalities

and :

- **Stable and reliable (open-source)**
  - Stable in the sense that the algorithms that are used will not be released too often (that makes us believe that it is a perennial solution).
  - Reliable in the sense that it implements well known algorithms and that there exists an important user community.
- **Compatibility**
  - Possibility of being integrated in an existing architecture (ex: Open System Architecture for Condition Based Maintenance)
- **Perenity**
  - Be sure that it is a long-term solution

AIRBUS GROUP

Outline

AIRBUS GROUP

Some possible Machine Learning tools

Tested Machine Learning tools by AGI

This is a non exhaustive benchmark where pros & cons are done according AGI experience.

- **KNIME**
  - Pros : Quite complete, Easy to use, possibility of graphical workflow
  - Cons : Not adapted for "complex" cases (high-dimensional data), oriented for having the commercial version
  - User profile : Domain Engineer for the reuse of packaged algorithms
- **RapidMiner**
  - Pros : Quite complete, Easy to use, possibility of graphical workflow
  - Cons : Not adapted for "complex" cases (high-dimensional data), oriented for having the commercial version
  - User profile : Domain Engineer for the reuse of packaged algorithms
- **Orange**
  - Pros : Easy to use, possibility of graphical workflow
  - Cons : Not complete enough
  - User profile : Domain Engineer for the reuse of packaged algorithms

AIRBUS GROUP

Some possible Machine Learning tools

Tested Machine Learning tools by AGI

The best ?

- **Scikit-learn**
  - Pros : Complete, BSD licence, stable, python library (possibility to use other libraries for data munging & data visualization)
  - Cons : Difficult to use for non data or computer scientist
  - User profile : data or computer scientist
- **R**
  - Pros : Very complete, huge user community
  - Cons : many techniques rely on packages that are not so stable (developed by teachers and/or students)
  - User profile : data scientist

AIRBUS GROUP

Outline

AIRBUS GROUP

Why scikit-learn ?

- **Python**

```
In [1]: import this
The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
```

AIRBUS GROUP

Why scikit-learn ?

- Python
- **Complete for our use in AGI (R & T)**,

  Currently, our practice in AGI is mainly the following :

  - **Data-Munging mainly with pandas** (and other python libraries)
  - **Machine Learning with scikit-learn and openturns**
  - **Visualization with matplotlib** and ... **bokeh** for interactive plots.
- **But**, for a deployment within our BU's,

  - **What about software pernity** ?
  - **What about compatibility** with "in-house" platforms ? (python makes us believe that it may be possible ...)

AIRBUS GROUP