Test for Data Scientists

Analysis of Manufacturing data

**Context description**

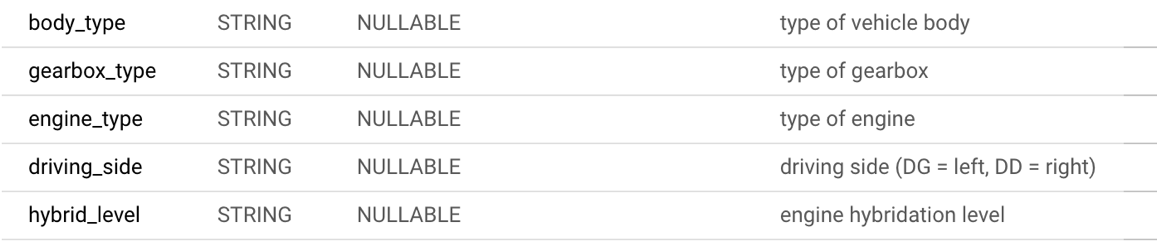
This test comes with CSV files extracted from one of Renault’s systems. In this use case, we want to read and analyze manufacturing data coming from a machine at the end of the assembly line: the filling machine. This machine fills the vehicles at the very end of the assembly line, by adding cooling fluid for the engine – called **RMFLuid** in the dataset, fluid for the braking system – called **FRFluid** in the data, and fluid for the air conditioning system – called **HFOFluid**.



Most of the machines in Renault’s plants have been equipped with sensors and the required hardware/software to send data about the process. The gathered data, after some preparation, looks like the CSV files we have in this exercise.

Une image contenant texte

Description générée automatiquement



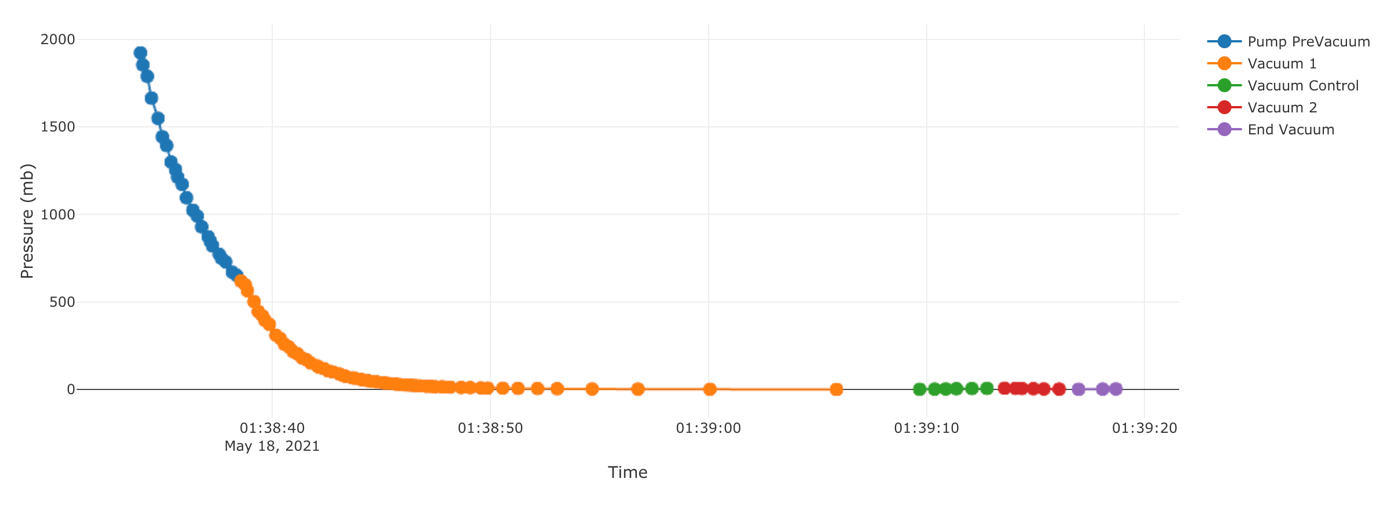
For this exercise, the data has been extracted and filtered with only 1 plant (**siteCode**) and machine (**objectUAI, machine**) at a time.

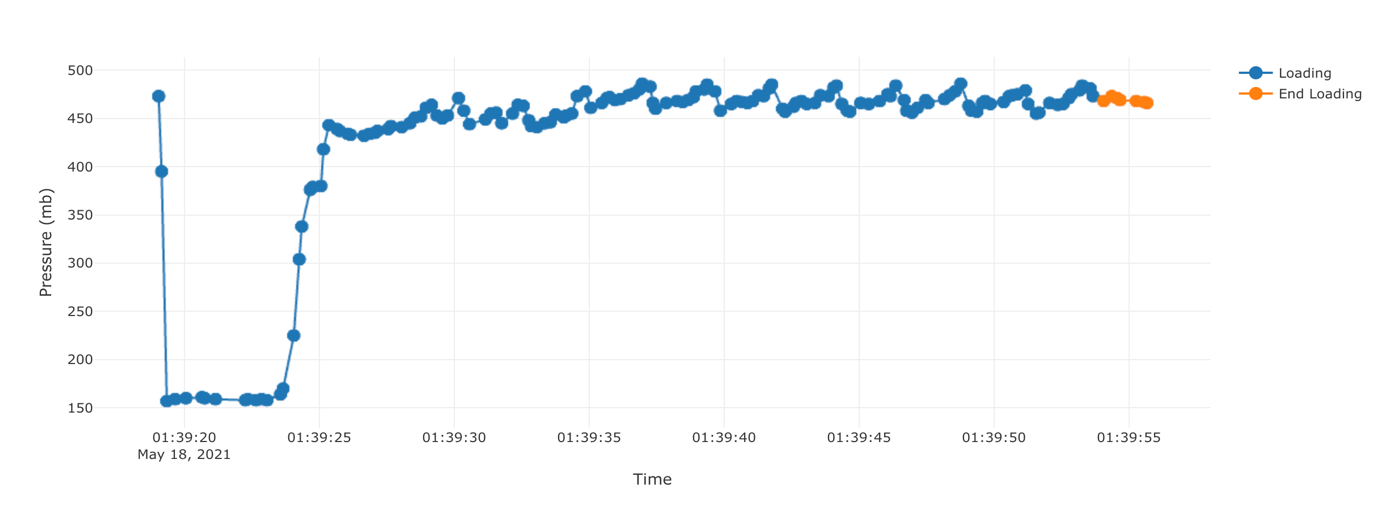
Fluids are filled one after the other (sequentially) during the filling process.

For each of them:

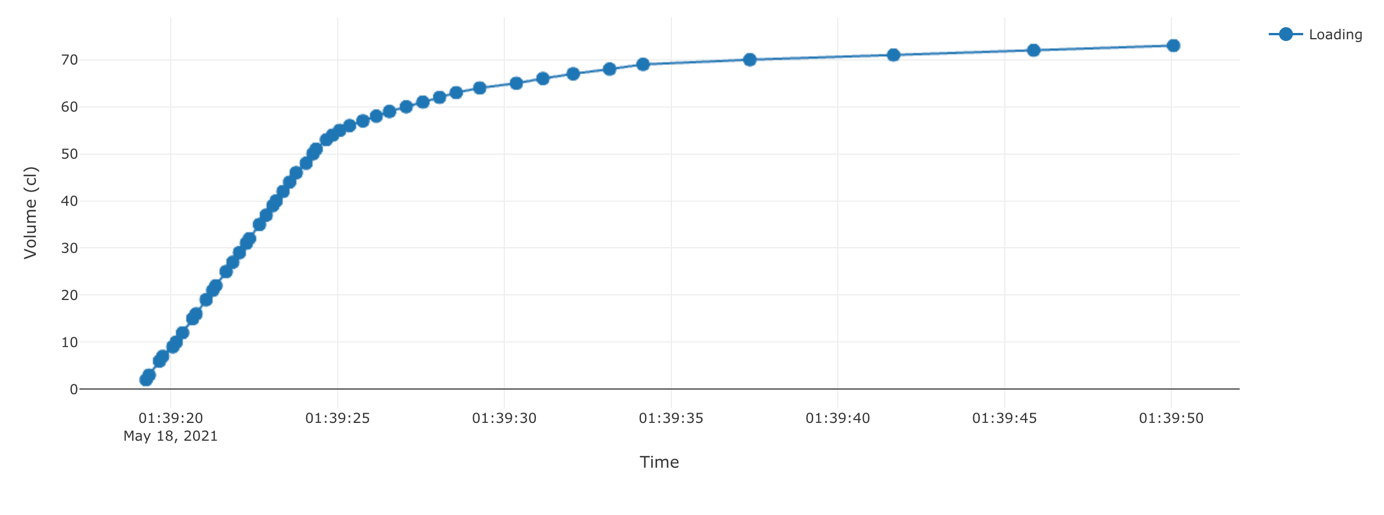
1. The machine starts by emptying the circuits to reach a vacuum. Indeed, any air bubble left in the circuit would cause serious trouble. This vacuum process is divided in smaller steps:
   1. **PreVacuum:** Beginning of the vacuum with the main pump
   2. **Vaccum1:** First big pressure fall with the main pump
   3. **VacuumControl:** The machine stops and check the pressure value. It is normal to experience a small increase of pressure during this phase.
   4. **Vacuum2:** Another pump is used to reach the final pressure value
   5. **EndVacuum:** The machine measures the final value, which was the old reference monitored by maintenance personnel.
2. Then it fills the circuit with the fluid.
   1. **Loading:** The main loading step
   2. **EndLoading:** Check of the final value

We monitor the following variables:

Pressure in the circuit during the vacuum phase (**ActVacuum** in the dataset):

Pressure of the machine during the filling phase (**ActLoadingPressure**): 

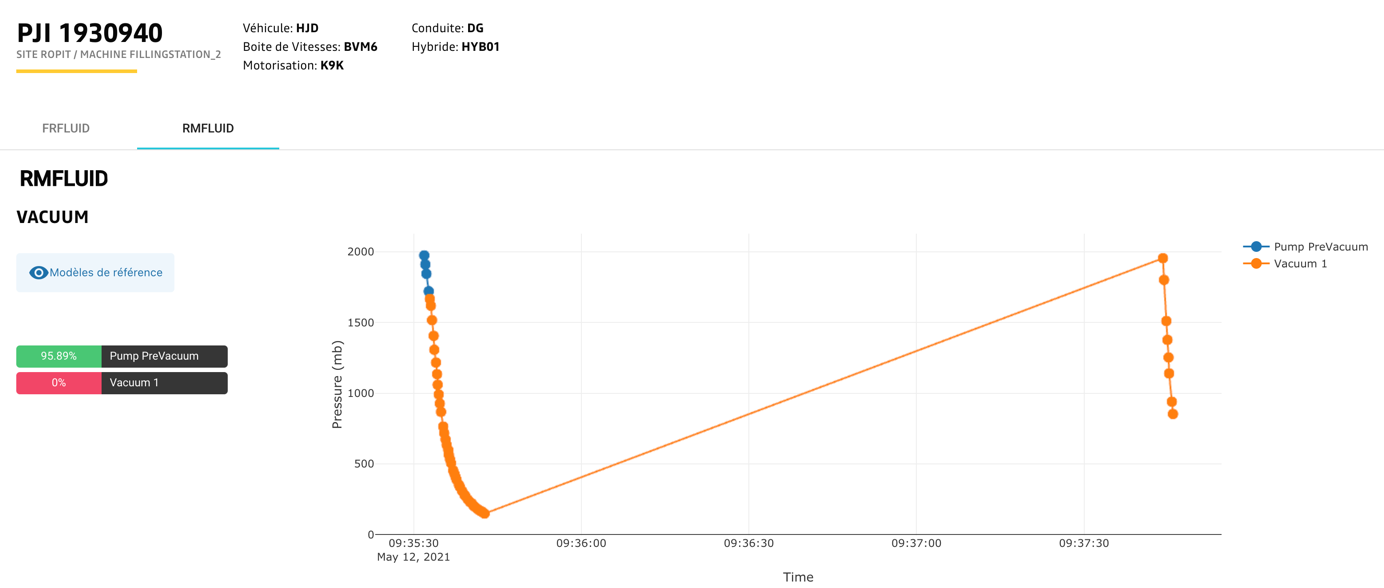
* Volume filled in the car during the filling phase (**ActVolume** or **ActMass**):



It is important to detect the anomalies in the process for several reasons:

* In case there is an anomaly on a car (e.g. car not fully filled, or circuit has a leak and vacuum can’t be reached), it is important to check that actions have been taken and we did not let a car with a quality issue get out of the assembly line.
* There may be a decay in the process over time due to machine degradations. We need to find these before any major problem occurs on the assembly line.

Here is an example of a car ID where we get an obvious anomaly (vacuum level could not be reached, despite the machine trying twice)



As you can see, we have an anomaly detection algorithm that yields a 0% score for this example on the Vacuum1 Step, therefore the car has been detected.

**Your Tasks**

1. Data exploration

We will be using df\_ropit.csv for this section.

* Open and read the data file.
* Write a function that allows you to visualize the data for a given PJI, fluid, and measurement
* How is the data balanced between the different car types (body\_type, gearbox\_type, engine\_type, driver\_side, hybrid\_level)?
* How does the vehicle diversity impact the process? Would you need to take it into account when designing an anomaly detection algorithm?
* How self-similar is the process for a given vehicle diversity?

1. Anomaly Detection

* Using df\_ropit.csv, filter the data to get the most represented type of car diversity.
* Design an algorithm that detects anomalies for each car of this type in an unsupervised approach.
* Plot a few examples of anomalies you can find.
* According to the business, a false negative (missing a real problem on a car) has 10 times the cost of false positive. How would you take this into account?
* Train your anomaly detection on train\_df\_lha.csv. Let’s assume this was our training sample before putting the algorithm into production.
* Search for anomalies in inference\_df\_lha.csv. Do you notice something happening starting on Jan-29? How would you react to this in a production environment?