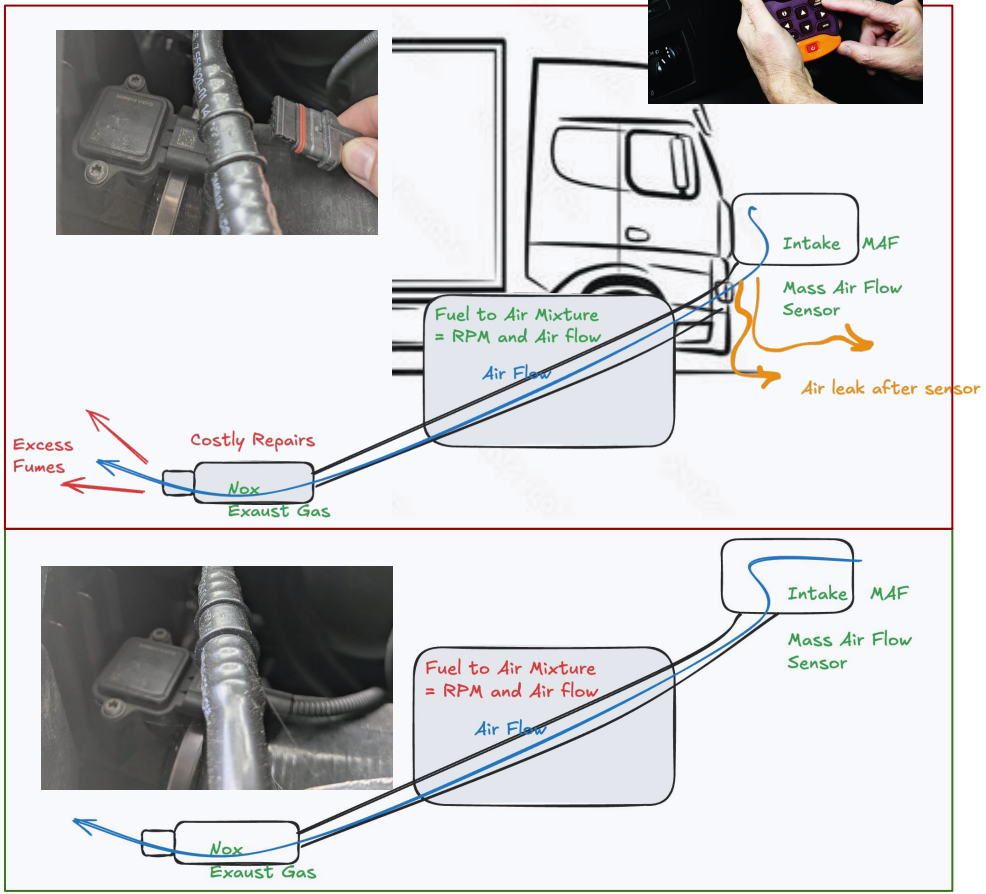
OBD-II and CANbus Data

Automotive issue

Problem

Not all engine faults are logged, when they are they create **Driver Troubleshooting Codes DTC**. That may not be addressed in time.

- **Problem:** Air leaks left unchecked can lead to error codes logged **(e.g. DTC P0401)** and cause downstream costs.
- **Sensors stream values:** accessed via the OnBoard Diagnostics **OBD** port.
- **Goal:** Access this sensor data and Alert the company if a truck is in an unsafe or unhealthy condition

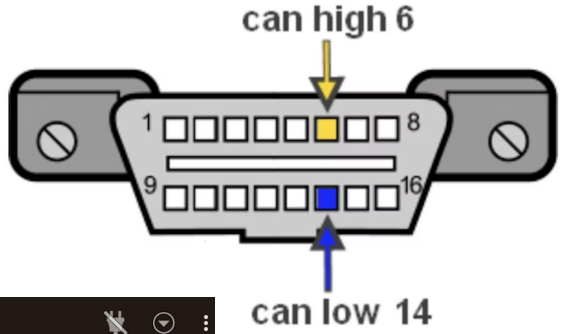


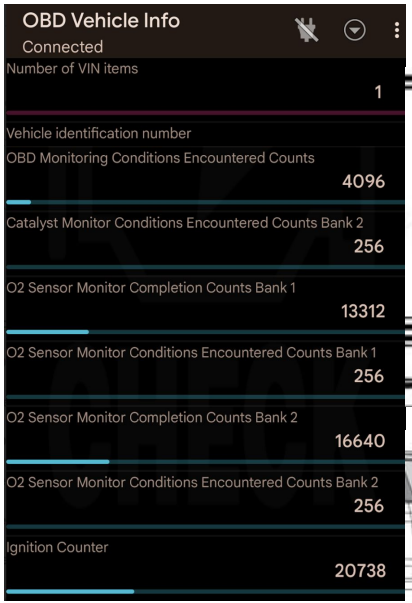
OBD-II / CANbus Data

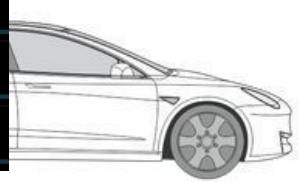
OBD-II PIDs (On-board diagnostics Parameter IDs) are codes used to request data from a vehicle, used as a diagnostic tool.
Range of options are available for data capture from sensors as frames, we can store as we do for sensors.

Automotive **Industrial**

- Standardized since 1994 (global adoption) From trucks to Teslas
- 16-pin connector with CAN-H / CAN-L differential pair
- Works across brands (even EVs have a service port)
- We can capture NOx, MAF, RPM, O2, etc.
- Download the open source android app - AndrOBD to get started capturing data as CSV







Automotive issue

Python-OBD

Python-OBD is a library for handling data from a car's **O**n-**B**oard **D**iagnostics port (OBD-II). It can stream real time sensor data, perform diagnostics (such as reading check-engine codes), and is fit for the Raspberry Pi.

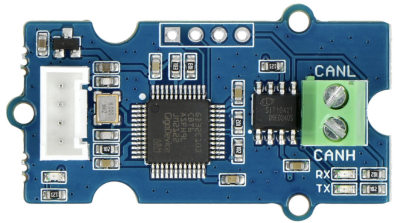
This library is designed to work with standard ELM327 OBD-II adapters.

```
import obd

connection = obd.OBD() # auto-connects to USB or RF port

cmd = obd.commands.SPEED # select an OBD command (sensor)

response = connection.query(cmd) # send the command, and parse the response

print(response.value) # returns unit-bearing values thanks to Pint
print(response.value.to("mph")) # user-friendly unit conversions
```

```
import obd

obd.OBD             # main OBD connection class
obd.Async           # asynchronous OBD connection class
obd.commands        # command tables
obd.Unit            # unit tables (a Pint UnitRegistry)
obd.OBDStatus       # enum for connection status
obd.scan_serial     # util function for manually scanning for OBD adapters
obd.OBDCommand      # class for making your own OBD Commands
obd.ECU             # enum for marking which ECU a command should listen to
obd.logger          # the OBD module's root logger (for debug)
```

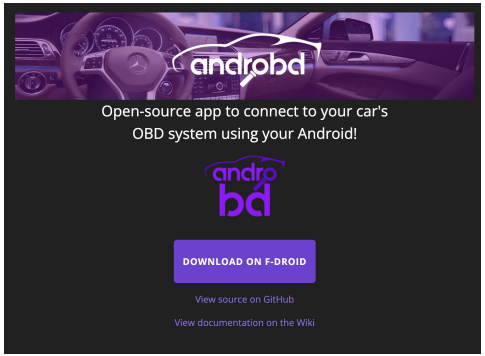
Data Capture / Implementation

Live OBD-II Pi/MCU (CAN HAT / Bluetooth ELM327 / grove) **Many options** are available for data capture and implementation and full integration, Can wires are internally available too not just via port



CSV (used today) for demo purposes

```
ts,label,nox_ppm,maf_gps,rpm

001,healthy,18,7.2,1800

002,leak,55,8.9,2100
```



Grab your data via a **AndrOBD** + **Edge Impulse CSV Wizard** to build initially.

Automotive issue

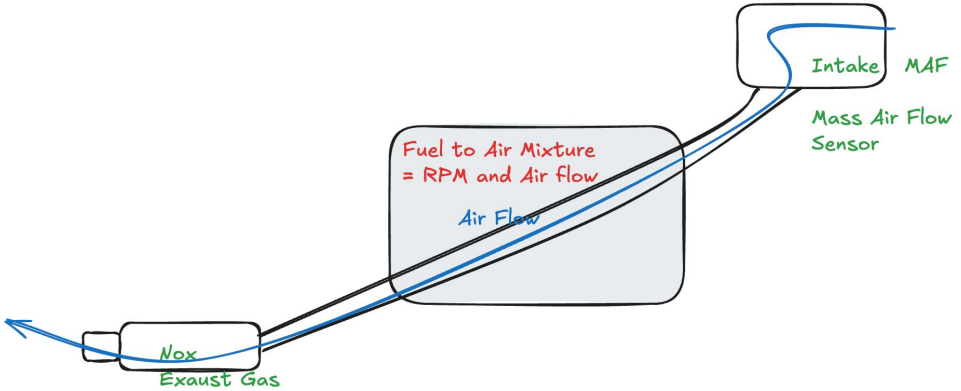
OBD-II Time-Series Fault Classification

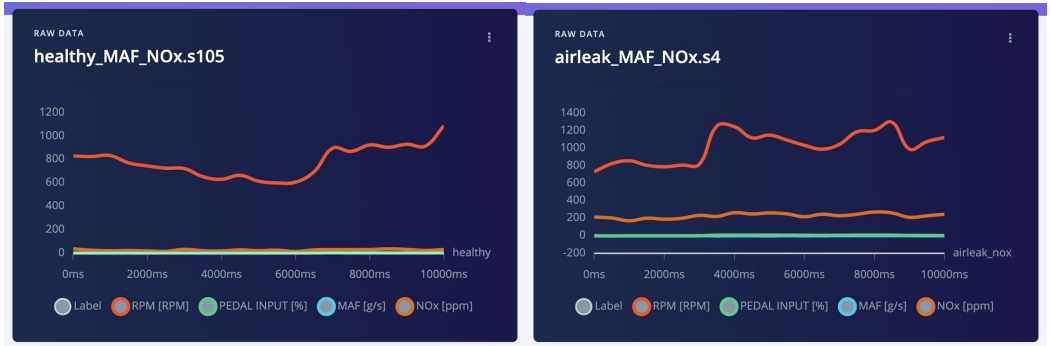
Healthy vs. Intake Air-Leak using NOx + Airflow (MAF)

- A leak in this system leads to lean mixture and **NOx** rises disproportionately relative to **MAF** at a given **RPM**.
- Inputs (minimal set):
- NOx (ppm) : exhaust gases

  MAF (g/s) : intake airflow

- RPM : Engine revolutions per minute





Impulse

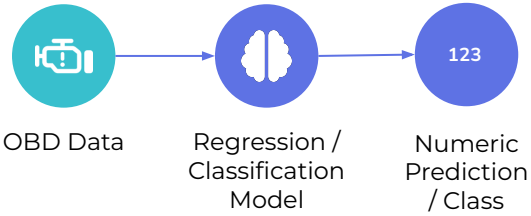
Edge Impulse | Impulse Design

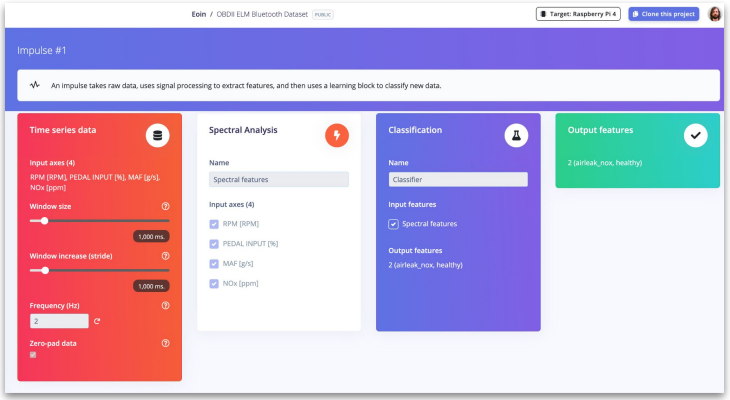
Classification: Classify an intake air-leak condition, part condition monitoring.

Under load, a leak > lean mixture > elevated NOx vs. healthy baseline, given airflow/RPM context.

Regression: would also work here for sensor failure alerting

- Standardized since 1994 (global adoption)
- 16-pin connector with CAN-H / CAN-L differential pair
- Works across brands (even EVs have a service port)
- We can capture NOx, MAF, RPM, O2, etc.





OBD DATA

Recap

Access sensor data from live vehicles or stream recorded CSVs

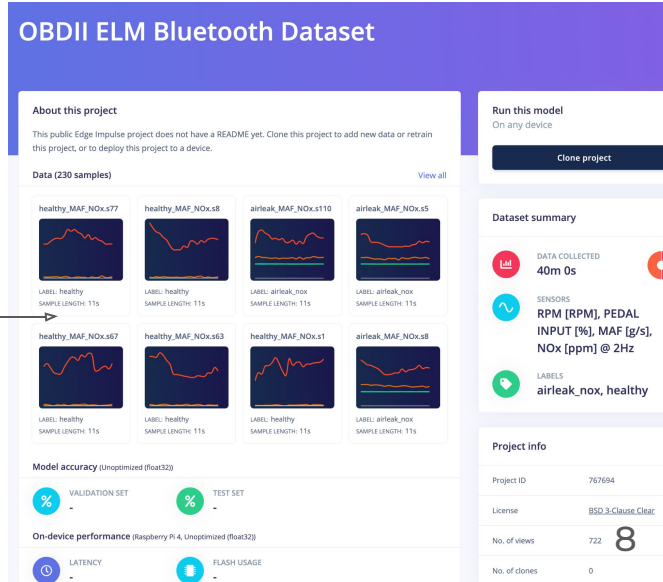
Edge Impulse **OBD** **Inference IoT**

- Live OBD-II captured via Pi/MCU (CAN HAT / ELM327)
- CSV (used today for demo simulation)
- Download the open source Android app - AndrOBD to capture data today on your own car.
- Fully integrate with a Linux device for Blues wireless remote connectivity.

Capture Sensor Data related to errors

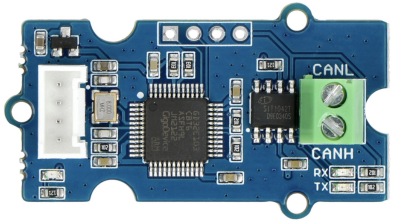
Add DTC codes as labels for Unhealthy State

```
time (ms.),fault_label,RPM [RPM],PEDAL INPU
[C],MAP [kPa]
0,healthy,666.000,10.980,2.930,0.000,48.000
574,healthy,672.000,10.980,2.930,0.000,48.0
1135,healthy,663.000,10.980,2.910,0.000,48.
1696,healthy,680.000,10.980,2.910,0.000,48.
2245,healthy,677.000,10.980,2.910,0.000,48.
```



References

Access sensor data from live vehicles or stream recorded CSVs



Edge Impulse - OBD Tutorial - https://docs.edgeimpulse.com/tutorials/end-to-end/obd-automotive-data

AndrOBD - fr3ts0n.github.io/AndrOBD/

AI Mechanic FOSDEM 2022 - https://archive.fosdem.org/2022/schedule/event/lt_car_whispering/

Federated Automotive Diagnostics: Conference: IECON 2024 - 50th Annual Conference of the IEEE Industrial Electronics Society Nov 2024 DOI:10.1109/IECON55916.2024.10905565