**PROJET VOLTRON: GREENTECH**

**RESUME**

Dans l’optique d’aider les vignerons dans pour qu’il améliore la production en automatisation. De ce fait pour le moment ils font tout à la main, alors nous allons apporter une solution dans l’optique de palier à tout cela.

Pour ce fait et pour la solution que nous allons mettre en place, nous nous sommes répartis en plusieurs équipes et tout au long du rapport nous en parlerons par spécificité.

Les équipes sont constituées tel que suit :

* Equipe IA
* Equipe DATA
* Equipe IOT
* Equipe CLOUD
* Equipe Sécurité
* Equipe VR

Nous allons tout au long de ce document vous présenter les spécificités et ce qui doit être fait.

Concernant la mise en place mise en place des Objectifs nous le spécifions tel de cette manière

**IA**

**Introduction**

This document purpose is to delivering some generic off-the-shelf solutions for the IA part of our project.

**Humans ressources**

**Organisation**

* Discord for fast communication
* Jira for sprint organisation
* Notion for documentation and organisation
* Github for host or code
* Agile method

**Tools**

**Data**

Currently, we don’t really have information on the exact data we will have available.

However, there is a process that we know we will be able to follow:

* Load the data
* Data visualization
* Preprocessing
* Missing values, categorical variables etc ..
* Feature Engineering
* The goal of feature engineering is simply to make your data better suited to the problem at hand.
* You might perform feature engineering to:
  + improve a model's predictive performance
  + reduce computational or data needs
  + improve interpretability of the results

**Algorithm**

Our problem seems to correspond to a supervised machine learning problem, more particularly classification problem.

Here is a list of the different algorithms we will be able to try:

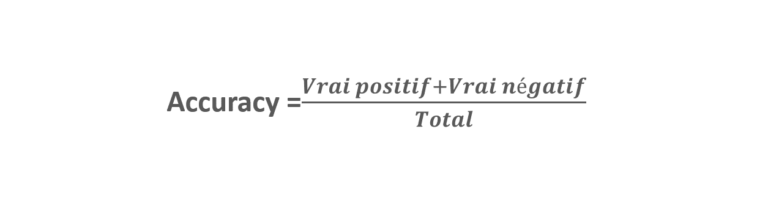
* Random Forest
* Neural network
* KNN
* XGBoost
* SVM

Then we will determine which one is the most efficient (see Metrics) for our problem and therefore which one we will use.

**Metrics**

* **Accuracy**

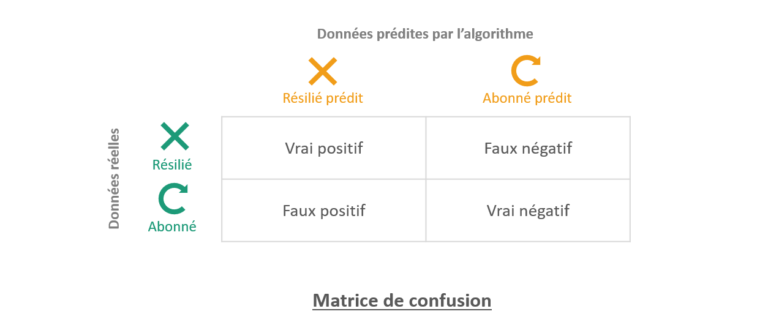
The simplest indicator is accuracy: it indicates the percentage of good predictions. It’s a very good indicator because it’s very simple to understand. The higher it is, the more accurate our model is.



* **Confusion matrix**

Matrix confusion is a cross table between real values and predictions. This matrix identifies 4 categories of results:

* + The right predictions:
    - True positives: customers who have terminated for whom the score predicted they would terminate
    - True negatives: customers who are still subscribed and for whom the algorithm correctly predicted that they would remain subscribed
  + False predictions:
    - False negatives: customers who have terminated but for whom the score wrongly predicted that they would remain subscribers
    - False positives: customers who stayed subscribed when the score incorrectly predicted they would terminate



* **Classification report**

This is one of the evaluation of the classification models. It displays your model’s accuracy, recall, F1 score and support. It provides a better understanding of the overall performance of our model.

[Classification report (1)](https://www.notion.so/0e51b3623dbb437ead27e84b6ac1a1e5)

* **Area under curve**

The area under the curve (AUC) is the measure of a classifier’s ability to distinguish classes and serves as a summary of the ROC curve.

The higher the AUC, the more effective the model is in distinguishing positive from negative classes.

## Conclusion

After different try, we create a model that predict health of grapes in function of pictures of them.

We have a success of 86% of the predictions (accuracy).

**IOT**

**Software**

| **Name** | **Description** | **Link** | **Price** |
| --- | --- | --- | --- |
| Visual Studio Code | IDE | <https://code.visualstudio.com/> | Free |
| PlatformIO | Framework | <https://platformio.org/> | Free |
| Eclipse Mosquitto | MQTT Broker | <https://mosquitto.org/> | Free |
| Autodesk Fusion 360 | 3D Modeling | <https://www.autodesk.fr/products/fusion-360> | Free |
| Prusa Slicer | Slicer | <https://www.prusa3d.com/page/prusaslicer_424/> | Free |
| InfluxDb | Software | <https://www.influxdata.com/> | Free |
| Telegraf | Software | <https://www.influxdata.com/time-series-platform/telegraf/> | Free |
| Docker | Software | <https://www.docker.com/> | Free |
|  |  |  | 0$ |

**Hardware**

| **Name** | **Description** | **Link** | **Price** |
| --- | --- | --- | --- |
| DHT11 | Air Temperature, Humidity | <https://www.amazon.fr/> | 7$ |
| Capacitive soil moisture sensor v1.2 | Soil Moisture | <https://www.amazon.fr/> | 5$ |
| GL5539 | Photo Resistor | <https://www.amazon.fr/> | 5$ |
| Arduino nano RP2040 connect | Controller | <https://store.arduino.cc/> | 25$ |
| PETG | 3D Printing Filament | <https://www.prusa3d.com/category/prusament-petg/> | 30$ |
| MK3S+ | 3D printer | <https://www.prusa3d.com/product/original-prusa-i3-mk3s-kit-3/> | 700$ |
|  |  |  | 772$ |

## Data Collection

The data will be read and sent from the following sensors every 15min:

* DHT11
* CSMS v1.2
* GL5539

These sensors will provide us with data of the environment where the module was setup, we will measure the following data:

* Air
  + Temperature (in degress C)
  + Humidity (%)
* Soil
  + Humidity (%)
* Brightness (0 → 1000)

The data will then be sent via MQTT which is a lighter and faster protocole than standard http request and allows us to publish data to topics instead of routes, so that we can listen to theses topics from different sources if need be.

The controller will publish to the following topics:

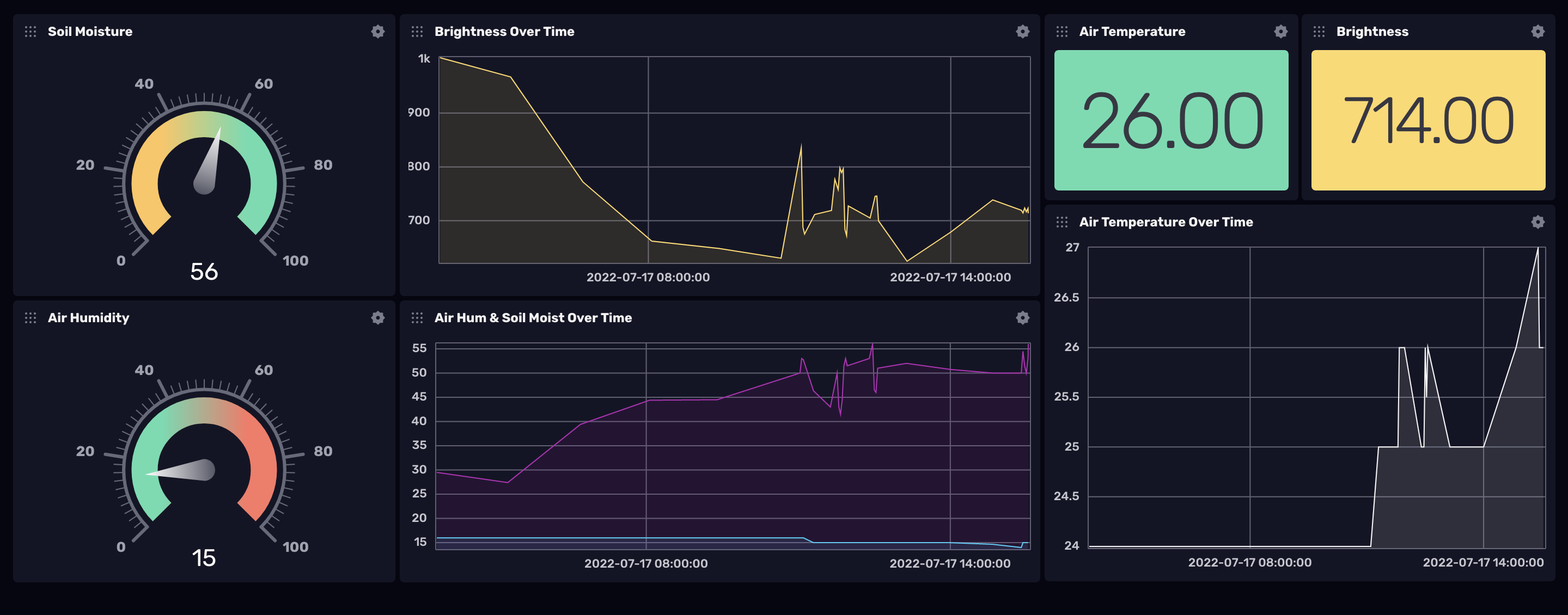
* sensors/greentech/v1/deviceName/airTemperature
* sensors/greentech/v1/deviceName/airHumidity
* sensors/greentech/v1/deviceName/soilMoisture
* sensors/greentech/v1/deviceName/brightness

The device name will change to differentiate each module and compare data bewteen them.

To save the data we will be using influxDB a time series database, and to listen to our topics we will be using the telegraf plugin of influxdb.

Telegraf will subscribe to our previously defined topics and store the data everytime the data is sent.

InfluxDB will then provide us a customizable dashboard to preview our data:



This data can be fetched outside of influxdb for other use using the influxdb client available in multiple languages.

## 🔧 Maintainability & Resilience

Our modules will be designed with high maintainability in mind, each sensor will be “plug & play” meaning they are easily swappable by the end user.

The main circuit board will have a port dedicated for each of these modules requiring zero to little experience.

3D model of the module casing will be available to reprint if needed.

Circuit Schematics & Maintenance Documentation will aslo be publicaly available.

Each component of a module will be sold by us if replacement was ever needed.

Critical sensors will be protected by the casing designed to be weatherproof and withstand the harshest conditions

Exposed sensors will all be picked for their weatherproof rated and tested

## 🔋 Power

This project being remote we will power it using solar power and storing energy in lithium cells.

To save on power consumption we will only measure data on given intervals then go into a “Deep Sleep” mode which will reduce our power consumption to nearly 0.

To determine the required power will need to calculate it based on the power consumption of each sensor and the controller, based on the operating time and solar exposure of the region it will be deployed.

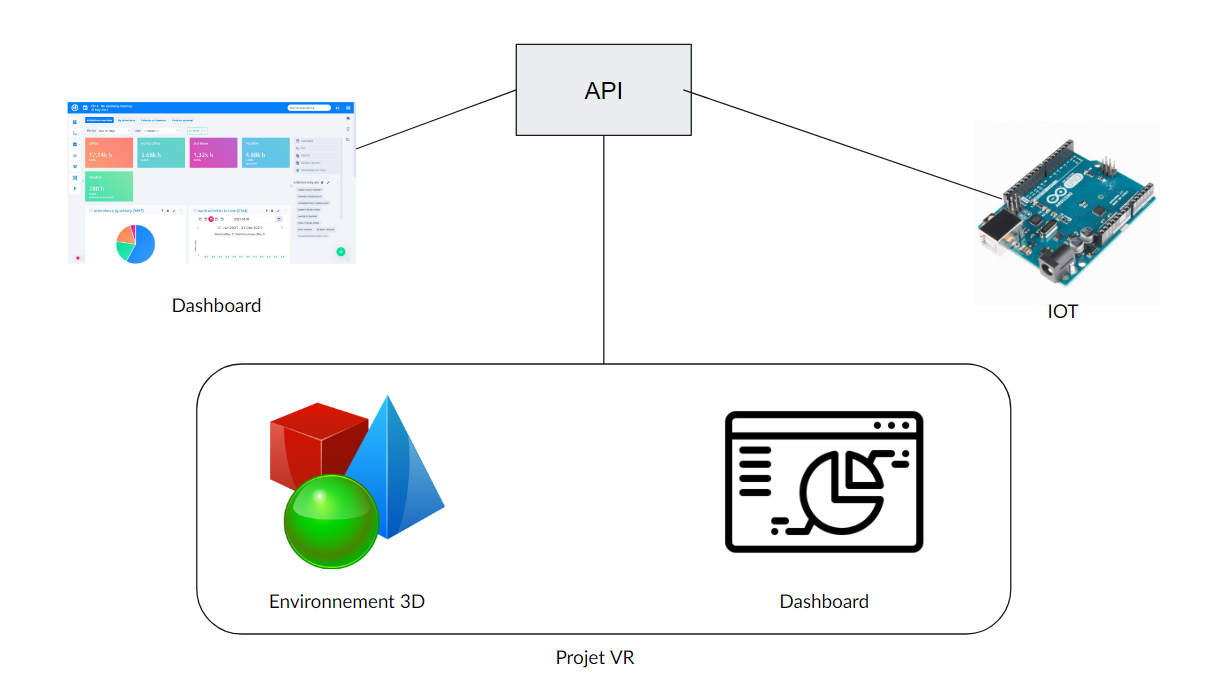
**Power Comsumptiono formula:** (Uptime) x (Days Running) x (Total Module Wattage) / 1000.

**Solar Cell Power:** solarCellPower = Power (Wc) / area (m²) × 1000

**VR**

## Project

For the project, we will create a virtual reality experience to put the user in real condition. Our solution will communicate with the API of the project and adapt the virtual elements like vines, meteo and IOT project around the user. With the controllers, the user can open a menu where more data about his vines will be displayed. We will use Unity to make the software and Unity asset store, Turbo squid and Free3D to get free 3D models and textures. We also use Blender to create 3D models.



**Humans ressources**

| **Name** | **Role** |
| --- | --- |
| Tom LEAL | XR developper |
| Joseph SALEON-TERRAS | XR developper |
|  |  |

**Tools and price**

| **Name** | **Description** | **Type** | **Price** |
| --- | --- | --- | --- |
| Unity | 3D motor to make virtual and augmented reality software | Software | 1 656 € by user. 1 656 \* 2 = 3 312 € |
| Unity Asset store | Assets bank (3D models, textures, …) for Unity. We only use free assets. | Service | Free |
| Oculus quest 2 128GB | The most popular VR headset at the moment to develop our solution. | VR headset | 350 € |
| Blender | Software to make 3D models | Software | Free |
| TurboSquid | Website where you can download textures | Website | Free |
| free 3D | Website where you can download textures | Website | Free |
| TOTAL |  |  | 3 662 € |

**BIG DATA**

For this part of project we will create a solution who can respond all subject. We know we will receive data from IOT. First we must understand what technologies and what kind of data they will give us. Second this data is treated and classified (per volume , date , etc ) . Finally we give an access of our data for IA .

For that we will choose a database. This choose is important because different technologies will use the database. The database will be easy to use, and fast installation. Technologies like broker mqtt and python wil be recommended . And our chose was Mongo db or mysql .

Finally we will have a dashboard for a view of all data. We know a lot of dashboards was on internet in html . But we have a conviction broker mqtt and node red have the capacity to give ours the dashboard we want.

### Version 2.0

In order not to wait for our IOT comrades to get data, we had to look for free data to continue our work. We found free data on the websiteNous avons donc puis avoir un fichier csv que nous avons manipulé pour faire les dashboard. [public.opendatasoft.com](http://public.opendatasoft.com/) .

We then had a csv file that we manipulated to make the dashboard.

## Humans ressources

| **Name** | **Role** |
| --- | --- |
| Dimitri RAYMOND | Big Data developper |
| Sitti Fatouma ALIAMANI | Big Data developper |

## Tools

| **Name** | **Description** | **Type** | **Price** |
| --- | --- | --- | --- |
| pandas | base de donnée | Python framework | free |
| dash | data dashboard | Python framework | free |
| ploty | data package | Python framework | free |
| Python | laguage | language | free |
| internet | documentation | software | free |