Automatic Filling of Data Gaps

VII Iberian Modelling Week

Coordinator: Ricardo Enguiça - ISEL

Diogo Silva - University of Minho
Eduarda Machado - University of Minho
Julian Guillo - University of Valencia
Tiago Mota - University of Porto

1st December 2021

Outline

1. Introduction to the Problem
2. Statistical Approach
3. Algorithm Approach
4. Main Conclusions

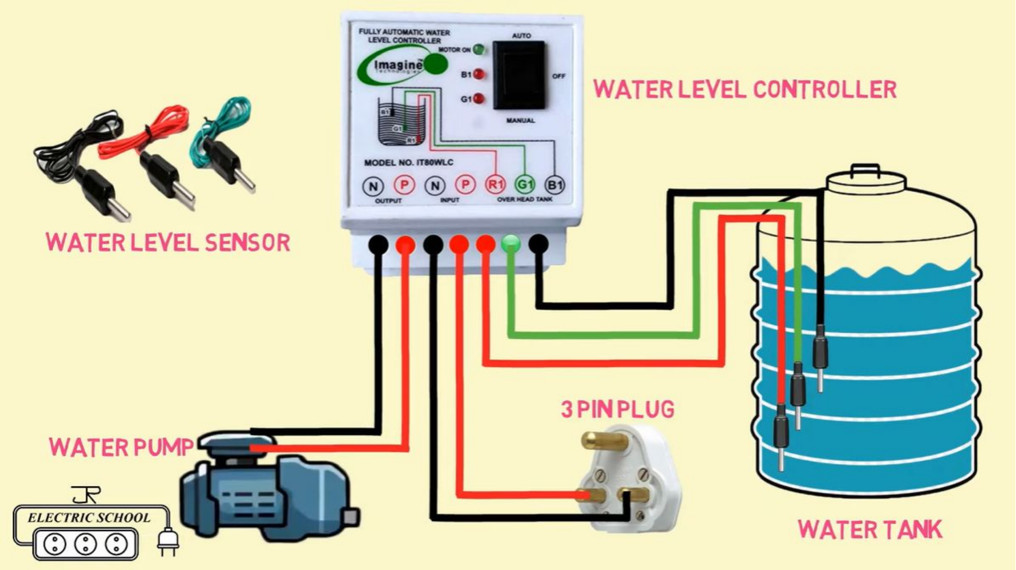
Introduction to the Problem

Introduction to the Problem

A lift station has the possibility to enable up to **three** different water **pumps**, in order to lift the water that arrives continuously to the station.

Water level thresholds

Stop 1	0,6
Stop 2	1,2
Stop 3	1,9
Start 1	2,1
Start 2	2,3
Start 3	2,5



Schematic of the water tank and pump system

Introduction to the Problem

Data (~70,000 rows):

- **Level of water** in the station, every **five minutes** for a year aprox. (2020);
- **Number of pumps** working every five minutes (0, 1, 2 or 3).

Problem:

- Some of the data has errors in it, e.g. it says 0 pumps are working when judging by the level of water there should be at least 1 or 2 pumps.

Objective:

- Correct errors in data;
- Predict the correct number of pumps that should be working given the level of water.

Introduction to the Problem

Two different **approaches** to the problem:

- Fitting a statistical/machine learning **model** to the data and then predicting with it;
- Creating an **algorithm** that resembles the data generating process.

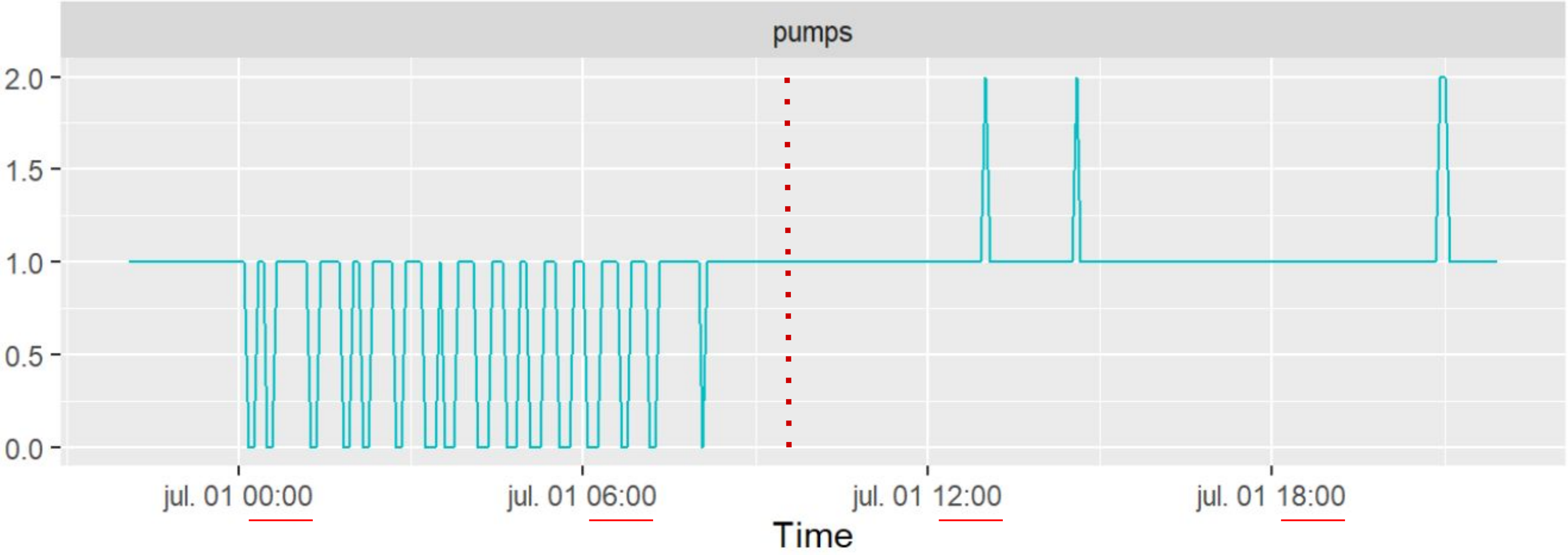
Statistical Approach

Statistical approach

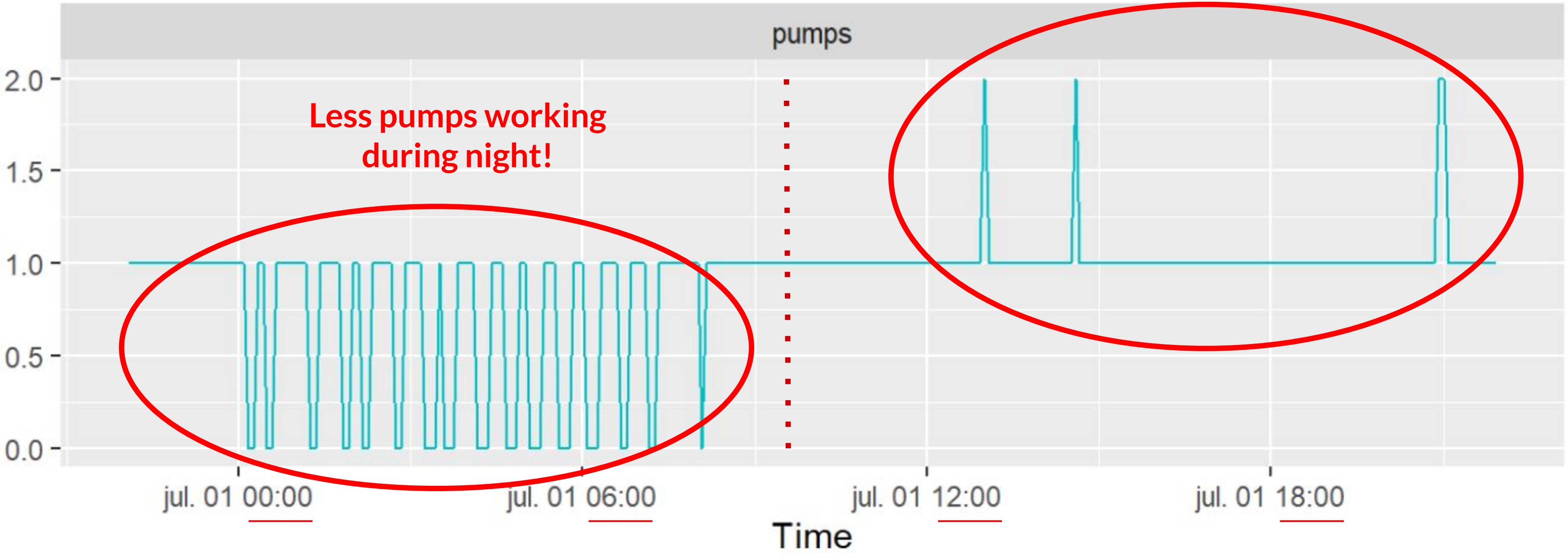
Steps:

1. Explore the data, look for patterns
2. Adjust simple baseline models
3. Improve accuracy of the models: feature engineering and hyperparameter tuning
4. Validation of models

Data exploration



Data exploration



Less pumps working during night!

Data exploration

Season	average nº of pumps working
Winter	~1.12
Spring	~1.1
Summer	~0.98
Autumn	~1.3

Baseline model - Silly algorithm

Model assumes nº of pumps is always 1

nº of pumps	% of data points
0	11.3%
1	75.8%
2	12.7%
3	0.05%

Baseline model - Silly algorithm

Model assumes nº of pumps is always 1

nº of pumps	% of data points
0	11.3%
1	75.8%
2	12.7%
3	0.05%

More sophisticated models should have better accuracy than 75%

Feature engineering

Variables we start with:

- Level of water
- Nº of pumps

data exploration

New variables added:

- Rate of change in water level

  (how much has the level of water changed with respect the previous 5-10-15 minutes)

- Nº pumps active 5 minutes before
- Season
- Day/night

Models trained

Accuracy	Linear regression	Multinomial regression	Random forest
Before adding new features	79%	82%	88%
After adding new features	87%	90%	99%

Models trained

Accuracy	Linear regression	Multinomial regression	Random forest
Before adding new features	79%	82%	88%
After adding new features	87%	90%	99%

Random forest classifier

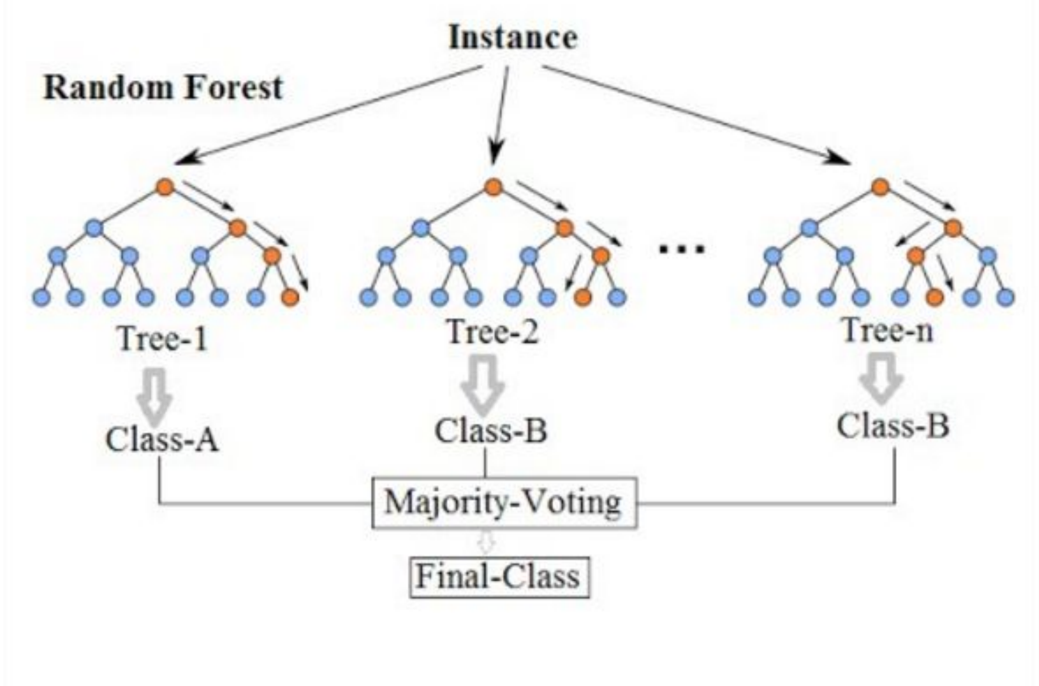
Train n different decision trees with all data

↓

Make predictions for each data point with each tree

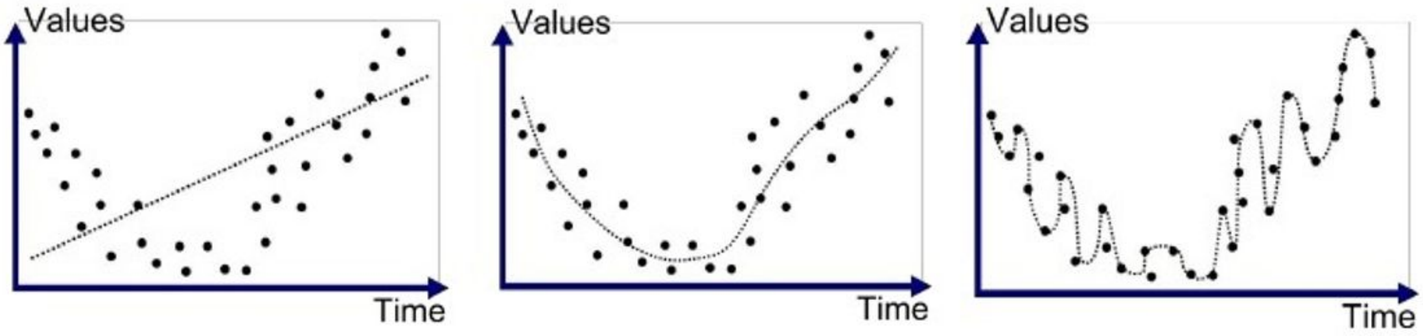
↓

Assign a class to each data point based on majority voting



Validation of the models

We don’t want our model to overfit the data → Very poor predictions outside of training data!



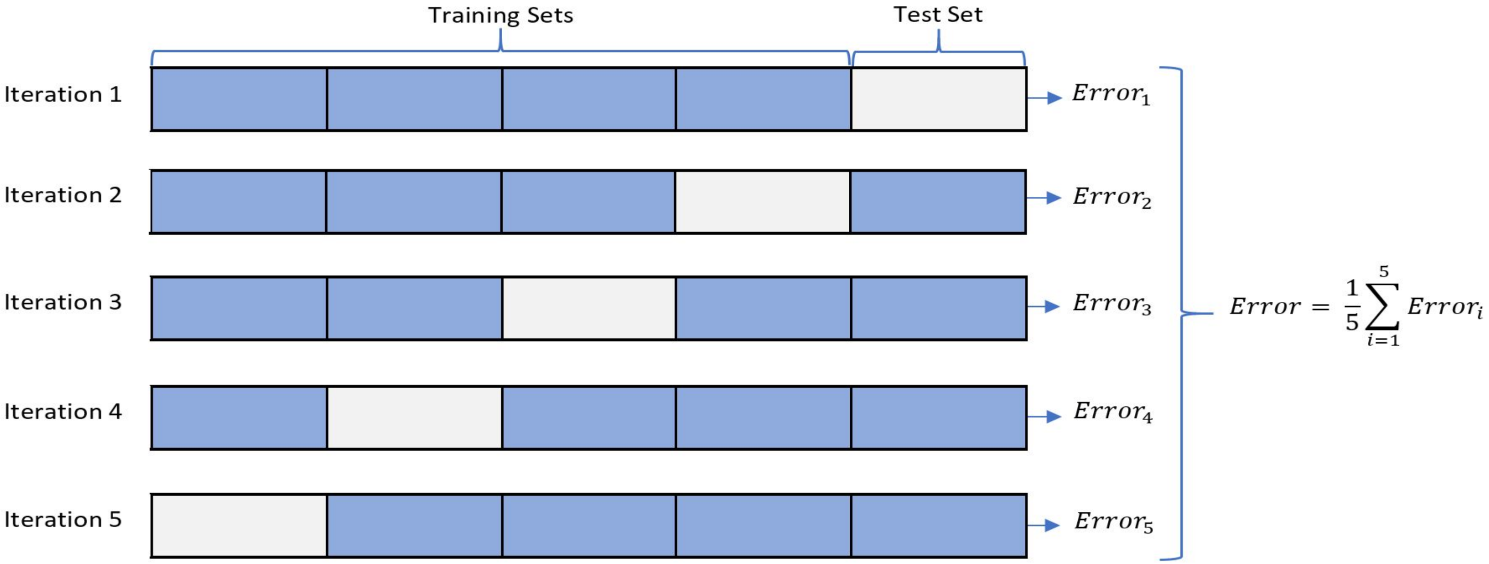
Underfitted

Good Fit/Robust

Overfitted

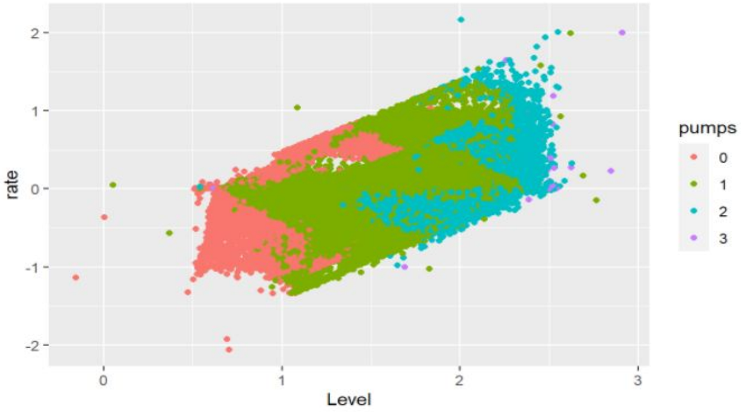
Validation of the models

K-fold cross validation → We validate the models with data they have not seen previously!

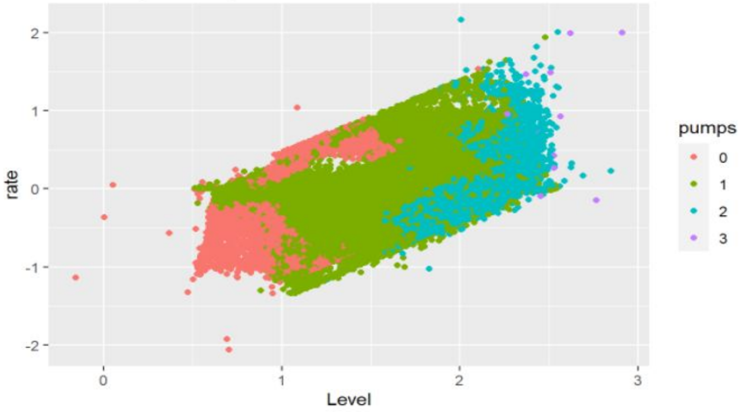


$$Error = \frac{1}{5}\sum_{i=1}^{5} Error_i$$

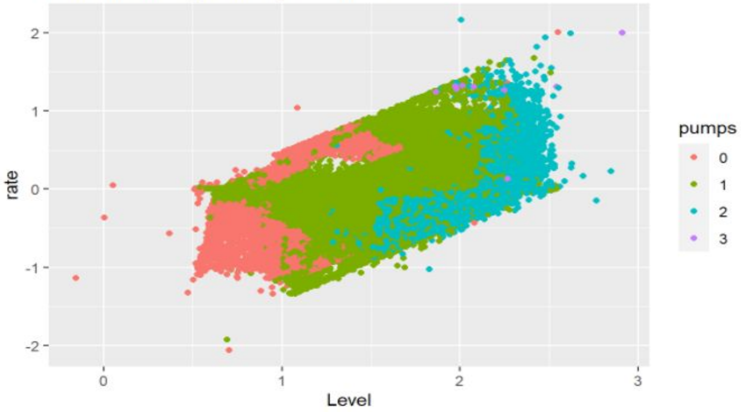
Real data



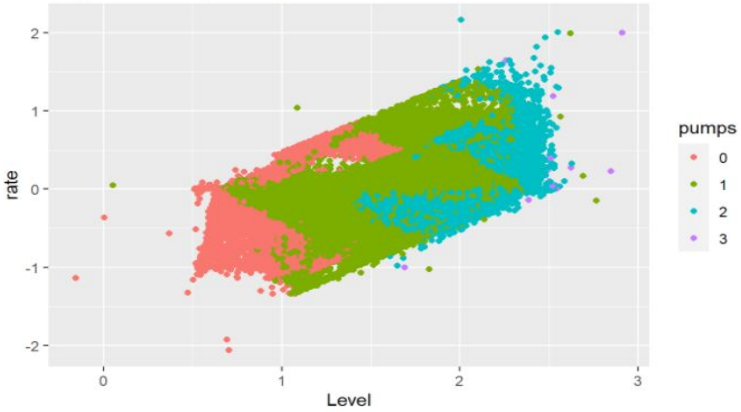
Linear regression predictions (87% accuracy)



Multinomial regression predictions (90% accuracy)



Random forest predictions (99% accuracy)



Statistical approach

Main problem of this approach:

We are training the models to fit data that has errors on it!

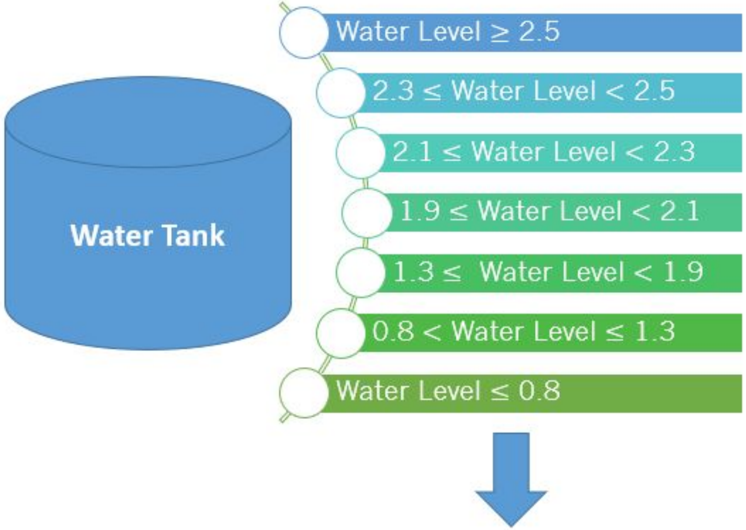
Using only the data, we won’t be able to fix the errors

We need some external knowledge for that

Algorithmic approach

Development of an Algorithm

The algorithm must be capable of determining how many pumps are active by analysing various parameters such as the water level and the flow rate.



Data regarding the water level in the tank is taken every 5 minutes.

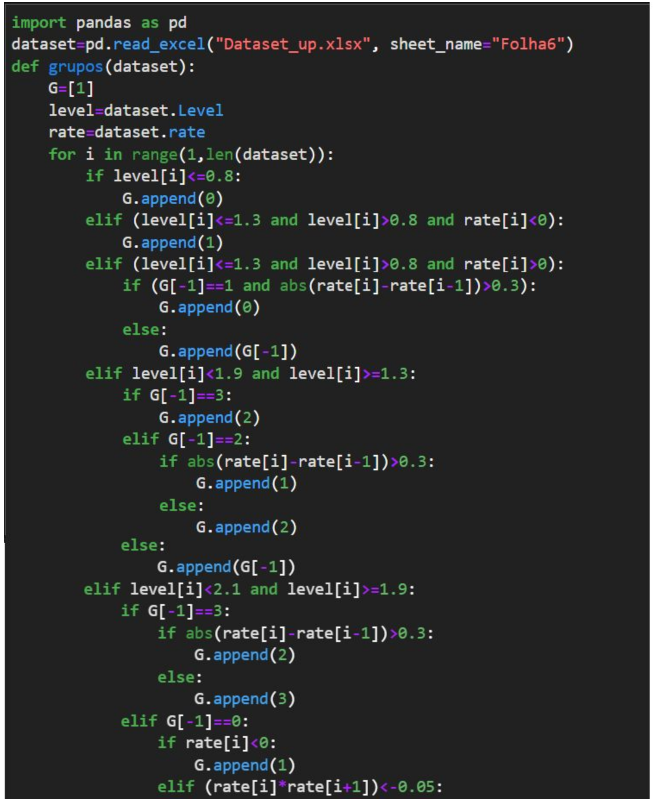
Algorithm Parameters

- Water level thresholds; →
- Rate of change - if more water is entering or exiting the tank;
- How many pumps were active in the previous or the next period;
- Intensity of rate of change – if the rate varies significantly between two consecutive periods.

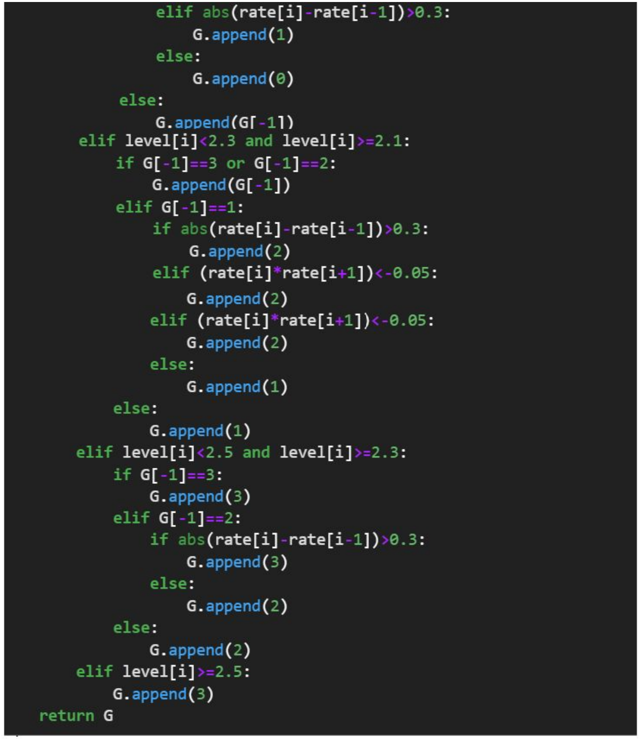
Water level thresholds

Start 3	2,5
Start 2	2,3
Start 1	2,1
Stop 3	1,9
Stop 2	1,3
Stop 1	0,8

The code

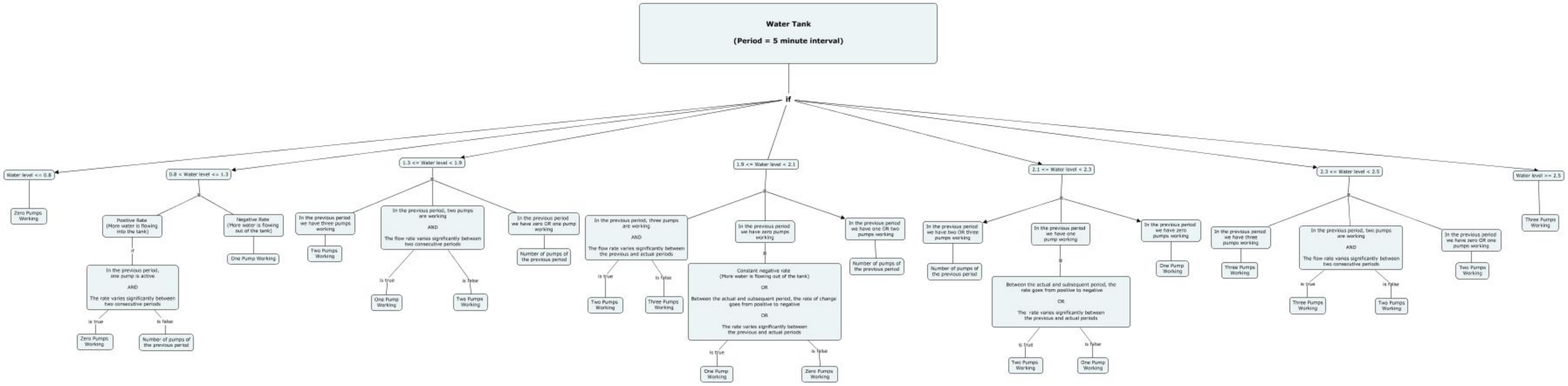


```python
import pandas as pd
dataset=pd.read_excel("Dataset_up.xlsx", sheet_name="Folha6")
def grupos(dataset):
    G=[1]
    level=dataset.Level
    rate=dataset.rate
    for i in range(1,len(dataset)):
        if level[i]<=0.8:
            G.append(0)
        elif (level[i]<=1.3 and level[i]>0.8 and rate[i]<0):
            G.append(1)
        elif (level[i]<=1.3 and level[i]>0.8 and rate[i]>0):
            if (G[-1]==1 and abs(rate[i]-rate[i-1])>0.3):
                G.append(0)
            else:
                G.append(G[-1])
        elif level[i]<1.9 and level[i]>=1.3:
            if G[-1]==3:
                G.append(2)
            elif G[-1]==2:
                if abs(rate[i]-rate[i-1])>0.3:
                    G.append(1)
                else:
                    G.append(2)
            else:
                G.append(G[-1])
        elif level[i]<2.1 and level[i]>=1.9:
            if G[-1]==3:
                if abs(rate[i]-rate[i-1])>0.3:
                    G.append(2)
                else:
                    G.append(3)
            elif G[-1]==0:
                if rate[i]<0:
                    G.append(1)
                elif (rate[i]*rate[i+1])<-0.05:
```



```python
                elif abs(rate[i]-rate[i-1])>0.3:
                    G.append(1)
                else:
                    G.append(0)
            else:
                G.append(G[-1])
        elif level[i]<2.3 and level[i]>=2.1:
            if G[-1]==3 or G[-1]==2:
                G.append(G[-1])
            elif G[-1]==1:
                if abs(rate[i]-rate[i-1])>0.3:
                    G.append(2)
                elif (rate[i]*rate[i+1])<-0.05:
                    G.append(2)
                elif (rate[i]*rate[i+1])<-0.05:
                    G.append(2)
                else:
                    G.append(1)
            else:
                G.append(1)
        elif level[i]<2.5 and level[i]>=2.3:
            if G[-1]==3:
                G.append(3)
            elif G[-1]==2:
                if abs(rate[i]-rate[i-1])>0.3:
                    G.append(3)
                else:
                    G.append(2)
            else:
                G.append(2)
        elif level[i]>=2.5:
            G.append(3)
    return G
```

Logical Thinking - Idea Map



Water Tank

(Period = 5 minute interval)

if

Water level <= 0.8

Zero Pumps Working

0.8 < Water level <= 1.3

Positive Rate (More water is flowing into the tank)

if

In the previous period, one pump is active

AND

The rate varies significantly between two consecutive periods

is true

Zero Pumps Working

is false

Number of pumps of the previous period

Negative Rate (More water is flowing out of the tank)

One Pump Working

1.3 <= Water level < 1.9

In the previous period we have three pumps working

Two Pumps Working

In the previous period, two pumps are working

AND

The flow rate varies significantly between two consecutive periods

is true

One Pump Working

is false

Two Pumps Working

In the previous period we have zero OR one pump working

Number of pumps of the previous period

1.9 <= Water level < 2.1

In the previous period, three pumps are working

AND

The flow rate varies significantly between the previous and actual periods

is true

Two Pumps Working

is false

Three Pumps Working

In the previous period we have zero pumps working

Constant negative rate (More water is flowing out of the tank)

OR

Between the actual and subsequent period, the rate of change goes from positive to negative

OR

The rate varies significantly between the previous and actual periods

is true

One Pump Working

is false

Zero Pumps Working

In the previous period we have one OR two pumps working

Number of pumps of the previous period

2.1 <= Water level < 2.3

In the previous period we have two OR three pumps working

Number of pumps of the previous period

In the previous period we have one pump working

Between the actual and subsequent period, the rate goes from positive to negative

OR

The rate varies significantly between the previous and actual periods

is true

Two Pumps Working

is false

One Pump Working

In the previous period we have zero pumps working

One Pump Working

2.3 <= Water level < 2.5

In the previous period we have three pumps working

Three Pumps Working

In the previous period, two pumps are working

AND

The flow rate varies significantly between two consecutive periods

is true

Three Pumps Working

is false

Two Pumps Working

In the previous period we have zero OR one pumps working

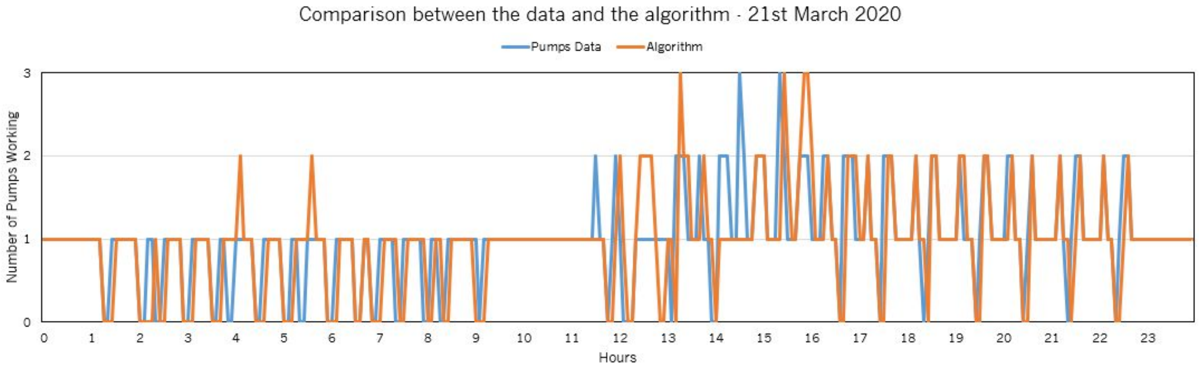
Two Pumps Working

Water level >= 2.5

Three Pumps Working

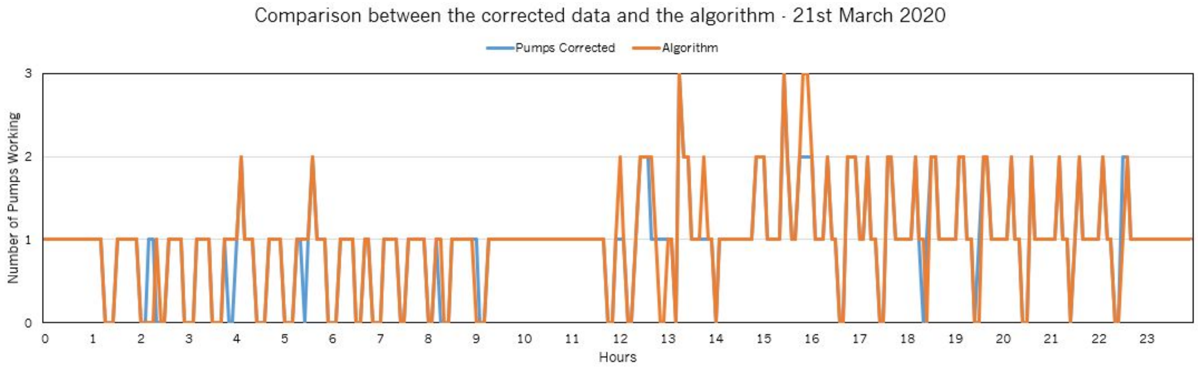
Accuracy of the Algorithm

Without Adjustments	Error	25.69%
	Deviation	4.86%



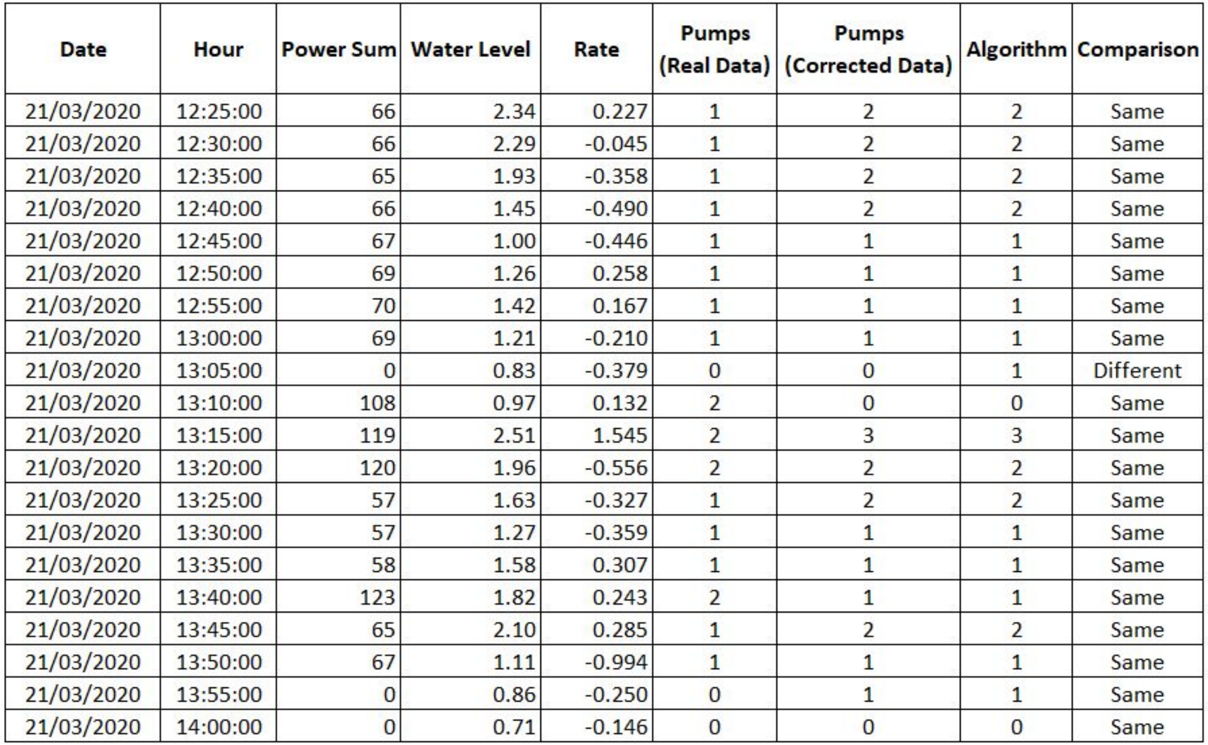
Comparison between the data and the algorithm - 21st March 2020

With Adjustments	Error	6.60%
	Deviation	1.04%



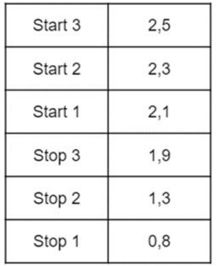
Comparison between the corrected data and the algorithm - 21st March 2020

Example:



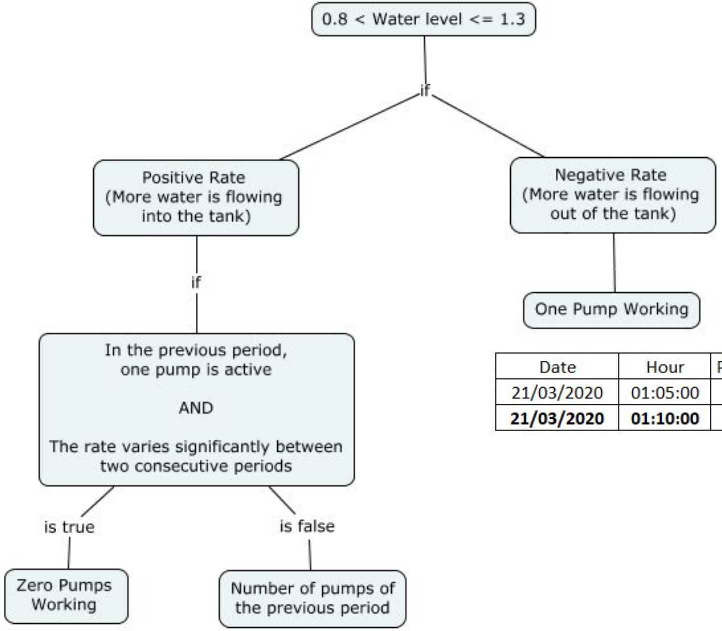
Date	Hour	Power Sum	Water Level	Rate	Pumps (Real Data)	Pumps (Corrected Data)	Algorithm	Comparison
21/03/2020	12:25:00	66	2.34	0.227	1	2	2	Same
21/03/2020	12:30:00	66	2.29	-0.045	1	2	2	Same
21/03/2020	12:35:00	65	1.93	-0.358	1	2	2	Same
21/03/2020	12:40:00	66	1.45	-0.490	1	2	2	Same
21/03/2020	12:45:00	67	1.00	-0.446	1	1	1	Same
21/03/2020	12:50:00	69	1.26	0.258	1	1	1	Same
21/03/2020	12:55:00	70	1.42	0.167	1	1	1	Same
21/03/2020	13:00:00	69	1.21	-0.210	1	1	1	Same
21/03/2020	13:05:00	0	0.83	-0.379	0	0	1	Different
21/03/2020	13:10:00	108	0.97	0.132	2	0	0	Same
21/03/2020	13:15:00	119	2.51	1.545	2	3	3	Same
21/03/2020	13:20:00	120	1.96	-0.556	2	2	2	Same
21/03/2020	13:25:00	57	1.63	-0.327	1	2	2	Same
21/03/2020	13:30:00	57	1.27	-0.359	1	1	1	Same
21/03/2020	13:35:00	58	1.58	0.307	1	1	1	Same
21/03/2020	13:40:00	123	1.82	0.243	2	1	1	Same
21/03/2020	13:45:00	65	2.10	0.285	1	2	2	Same
21/03/2020	13:50:00	67	1.11	-0.994	1	1	1	Same
21/03/2020	13:55:00	0	0.86	-0.250	0	1	1	Same
21/03/2020	14:00:00	0	0.71	-0.146	0	0	0	Same

Water level thresholds



Start 3	2,5
Start 2	2,3
Start 1	2,1
Stop 3	1,9
Stop 2	1,3
Stop 1	0,8

0.8 < Water Level <= 1.3



0.8 < Water level <= 1.3

if

Positive Rate (More water is flowing into the tank)

Negative Rate (More water is flowing out of the tank)

if

One Pump Working

In the previous period, one pump is active

AND

The rate varies significantly between two consecutive periods

is true

is false

Zero Pumps Working

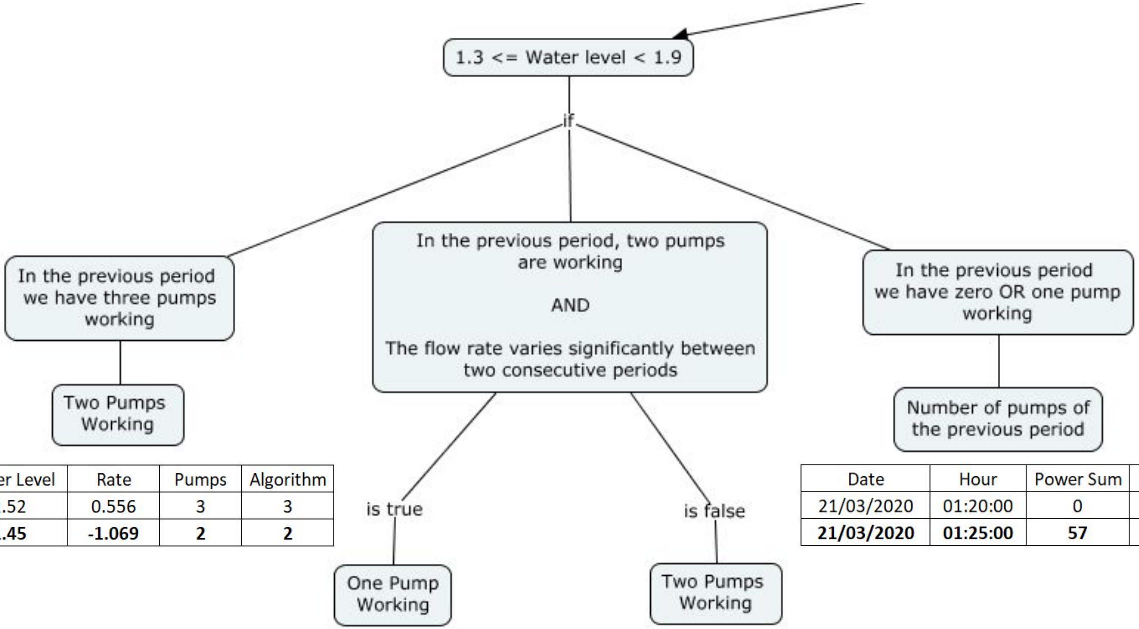
Number of pumps of the previous period

Date	Hour	Power Sum	Water Level	Rate	Pumps	Algorithm
21/03/2020	01:05:00	61	1.05	-0.129	1	1
21/03/2020	**01:10:00**	**61**	**0.92**	**-0.133**	**1**	**1**

Date	Hour	Power Sum	Water Level	Rate	Pumps	Algorithm
21/03/2020	02:20:00	0	0.94	-0.878	0	1
21/03/2020	**02:25:00**	**0**	**0.96**	**0.028**	**0**	**0**

Date	Hour	Power Sum	Water Level	Rate	Pumps	Algorithm
21/03/2020	06:50:00	0	0.65	-0.955	0	0
21/03/2020	**06:55:00**	**0**	**1.16**	**0.510**	**0**	**0**

1.3 < Water Level <= 1.9



1.3 <= Water level < 1.9

if

In the previous period we have three pumps working

Two Pumps Working

In the previous period, two pumps are working

AND

The flow rate varies significantly between two consecutive periods

is true

One Pump Working

is false

Two Pumps Working

In the previous period we have zero OR one pump working

Number of pumps of the previous period

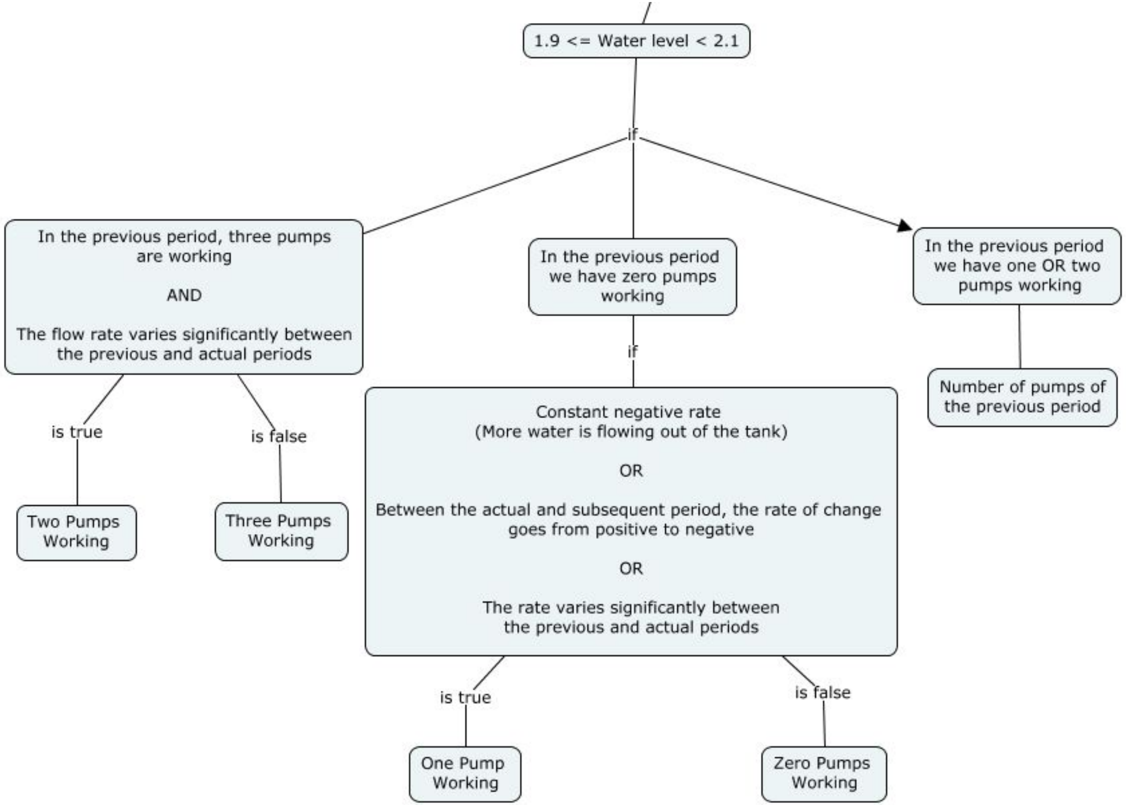
Date	Hour	Power Sum	Water Level	Rate	Pumps	Algorithm
21/03/2020	15:25:00	127	2.52	0.556	3	3
21/03/2020	**15:30:00**	**61**	**1.45**	**-1.069**	**2**	**2**

Date	Hour	Power Sum	Water Level	Rate	Pumps	Algorithm
21/03/2020	01:20:00	0	0.96	0.227	0	0
21/03/2020	**01:25:00**	**57**	**1.83**	**0.873**	**1**	**0**

Date	Hour	Power Sum	Water Level	Rate	Pumps	Algorithm
21/03/2020	17:10:00	124	2.35	0.895	2	2
21/03/2020	**17:15:00**	**65**	**1.81**	**-0.535**	**1**	**1**

Date	Hour	Power Sum	Water Level	Rate	Pumps	Algorithm
21/03/2020	16:50:00	120	2.09	-0.368	2	2
21/03/2020	**16:55:00**	**58**	**1.44**	**-0.65**	**1**	**2**

1.9 <= Water Level < 2.1



1.9 <= Water level < 2.1

if

In the previous period, three pumps are working

AND

The flow rate varies significantly between the previous and actual periods

is true

Two Pumps Working

is false

Three Pumps Working

In the previous period we have zero pumps working

if

Constant negative rate
(More water is flowing out of the tank)

OR

Between the actual and subsequent period, the rate of change goes from positive to negative

OR

The rate varies significantly between the previous and actual periods

is true

One Pump Working

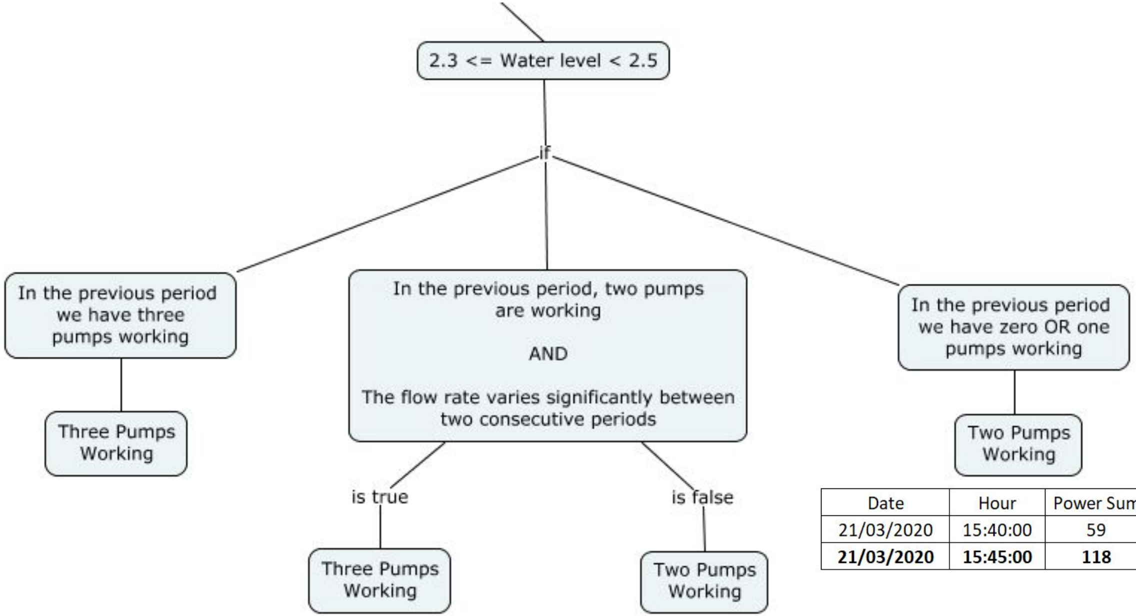
is false

Zero Pumps Working

In the previous period we have one OR two pumps working

Number of pumps of the previous period

2.3 <= Water Level < 2.5



2.3 <= Water level < 2.5

if

In the previous period we have three pumps working

Three Pumps Working

In the previous period, two pumps are working

AND

The flow rate varies significantly between two consecutive periods

is true

Three Pumps Working

is false

Two Pumps Working

In the previous period we have zero OR one pumps working

Two Pumps Working

Date	Hour	Power Sum	Water Level	Rate	Pumps	Algorithm
21/03/2020	15:40:00	59	2.05	0.476	1	1
21/03/2020	**15:45:00**	**118**	**2.34**	**0.290**	**2**	**2**

Date	Hour	Power Sum	Water Level	Rate	Pumps	Algorithm
21/03/2020	15:45:00	118	2.34	0.290	2	2
21/03/2020	**15:50:00**	**119**	**2.3**	**-0.042**	**2**	**3**

Date	Hour	Power Sum	Water Level	Rate	Pumps	Algorithm
21/03/2020	22:25:00	58	1.34	0.635	1	0
21/03/2020	22:30:00	115	2.27	0.930	2	1
21/03/2020	22:35:00	**118**	**2.39**	**0.118**	**2**	**2**

Main Conclusions

Main conclusions

1. Machine learning models can be trained to make very accurate predictions, but they are not useful to correct data with errors, as models adjusted to this data will have errors too. It would have been a great approach if instead of errors we would have had missing values on the data.
2. The algorithm allowed to develop an approach capable of predicting accurately the behaviour of the pumps in terms of various parameters and also to correct the actual data into values that make more sense.

Thank you for your time