Department of Computer Science

MSc Data Science and Analytics

Academic Year 2018-2019

Powerplant constraints forecasting using machine learning

Christophe Cestonaro

Registration number 1834138

A report submitted in partial fulfilment of the requirement for the degree of Master of Science

Brunel University
Department of Computer Science
Uxbridge, Middlesex UB8 3PH
United Kingdom
Tel: +44 (0) 1895 203397
Fax: +44 (0) 1895 251686

ABSTRACT

In this dissertation, we study how to use of machine learning methods and build models for hydro powerplant related timeseries predictive analytics. This is a multi-objective regression problem with constraints between target variables, which makes it unusual.

We are implementing an effective solution, by using two approaches. First, we are decomposing the problem into sub-problems, predicting high level values and then lower level ones. Second, we transform a multi objective regression problem into a classification one, by applying clustering on the target variables. This allows to reach appreciable accuracy, as it beats the provided baseline, and to respect the given constraints.

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Professor XiaoHui Liu for his support, complete trust and precious advices. As well, I would like to sincerely thank Professors Veronica Vinciotti and Alessandro Pandini, who despite not being directly involved in this project, are behind most of the methods and an approaches I used, through to their inspiring and excellent teaching.

My gratitude goes to the energy company who provided the data used in this project. In particular, I would like to thank Mrs Julie Ancel, who proposed this topic, prepared the data and shared many of her relevant intuitions. As well, her superior, Dr Andrea Poncet, head of the analysts team, who gave me some very valuable feedback.

Finally, I want to thank my wife and kids, for supporting me through this project, and through my entire master's year. They gave me strength, motivation and inspiration which were key in succeeding in this experience.

I certify that the work presented in the dissertation is my own unless referenced

Signature 

Date 19.09.2019

TOTAL NUMBER OF WORDS: 13'036

Contents

1 CHAPTER 1: INTRODUCTION

In this chapter, this dissertation topic is introduced: we'll present the problem, its context and explain why it is an interesting question to answer in the field of Data Science and Machine Learning.

1.1 CONTEXT AND PROBLEM STATEMENT

Within a given energy company, employees need to produce a forecast of production constraints for a given hydro powerplant. They approached us to solve this problem thanks to Machine Learning techniques and historical data they are providing.

In the next chapters, we will present this task in detail and propose an approach to reach prediction results as accurate as possible. We'll identify their strengths and limitations and draw conclusions about the attained results.

In order to understand this project, it is key to get an idea of its context, therefore we present it here shortly (more details are provided in appendix 9.1).

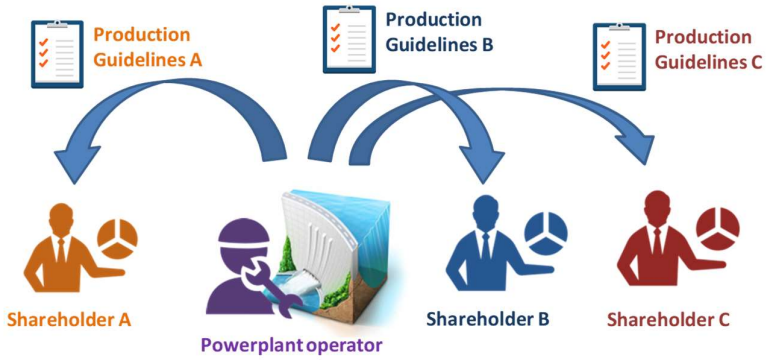


Figure 1. Production guidelines sent to powerplant shareholders

In short, the energy company is a shareholder of a given hydro powerplant. As such, it receives every day a production guideline, i.e. a document stating how much power and energy it can produce on the next day. With this information, energy traders can make decisions on what power amount to produce at what time. As the powerplant operator only delivers the guideline values and doesn't disclose how those values are generated, traders are missing a long-term forecast. This project aims filling this gap.

In details these are three quantities in the guidelines to be forecasted are:

1) **minimum energy** production constraint (MWh), for the next day
2) **maximum energy** production constraint (MWh), for the next day
3) **power blocks** that constitute the maximum energy. A power block is defined as a pair of "power value / number of hours" (MW/number). As there can be up 8 pairs on any given day, the problem really deals with 18 prediction targets (2+8*2). Number of hours in a block are whole numbers.

Concerning maximum energy production and related power blocks, quantities are linked by the following relation and must respect following constraints:

$$MaximumEnergy = \sum_{i=1}^{8} Power_i * NoHours_i \ , \qquad NoHours \in \mathbb{N}^+$$

$$\sum_{i=1}^{8} NoHours_i \leq 24$$

$$MaximumEnergy > MinimumEnergy$$

In practice, a trader will use power blocks as follows: he chooses how many hours in each power blocks to activate at the corresponding power, making sure the total energy is above the minimum constraint. Which means he is free to use all or none of the given numbers of hours in each block, at any time in the day. This flexibility makes a lot of the powerplant's energy value on the market, since hydro powerplants allow to target hours when price is the highest, contrary to renewables, like wind and solar, or nuclear.

Analysts in the energy company explain to us that both energy and power are equally important: energy values impact the reservoir content pattern in the yearly cycle whereas power values impact production decisions on the short term. Therefore, our prediction should be equally precise in the two cases.

Although the term "forecast" has been used by the analysts to introduce the problem, this is actually a regression problem, as assumption is made that the 3 target quantities mainly depend on a set of 10 input quantities: lake content, lake maximum level, natural inflows and power plant availabilities. They are communicated in the same guideline document as the 3 target quantities and we have almost perfect foresight on them during the prediction horizon since they are sufficiently accurately predictable, or results of human decisions made already.

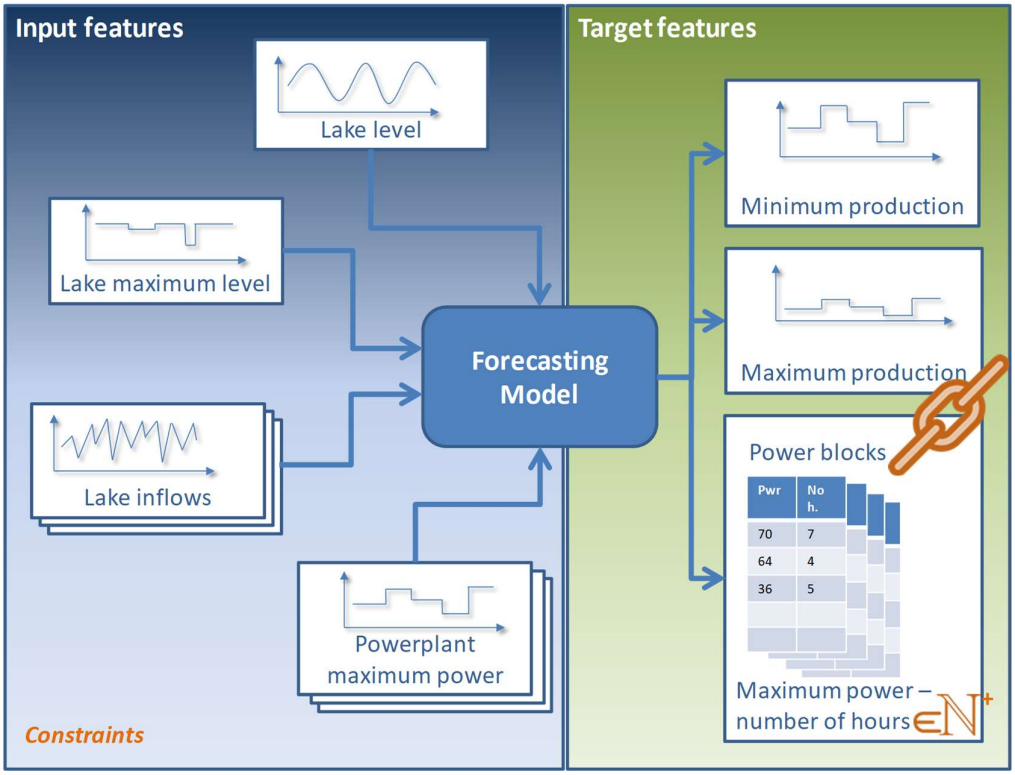


Figure 2. Forecasting problem summary.

Data history is provided from 1.4.2014 until 30.06.2019. Expected prediction horizon is 2 years.

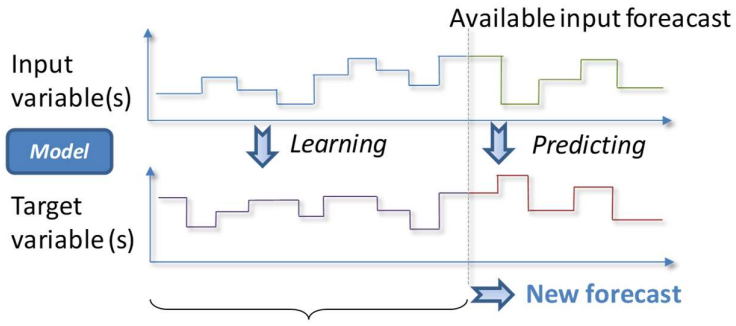


Figure 3. Model learning and prediction based on available data

About this problem, it is worth underlining:

- the 3 target quantities are not measured values, they are fabricated by the powerplant operator. We don't have any knowledge as how these numbers are generated. The project's stakeholders have only assumptions: they believe these values mainly depend on the other provided quantities; hence the problem fits a regression framework.

- consequently, the implemented solution is expected to "open the black box", i.e. not only produce accurate predictions but as well interpretable ones. One could consider this project as a "reverse engineering" task.

- a baseline prediction dataset is provided by the energy company: it is their first attempt at doing a forecast, using a set of simple algorithmic rules. Our results should beat this baseline in terms of prediction quality.

In our view, the mentioned constraints and relationships make this problem non-trivial: applying blindly ML algorithms with "brute force" will not produce results that respect them. But it is what makes this project challenging!

1.2 RESEARCH AIM AND OBJECTIVES

To sum up, our research aims at identifying and implementing ML methods that produce the most accurate predictions and respect the given constrains while providing interpretability, by extracting knowledge from the available data.

The precise objectives that make up this overall aim are the following:

1) Select adapted methods and ML models, through literature review, in order to identify similar problems and related approaches that proved to be successful
2) Collect, import and clean up data. Perform exploratory data analysis.
3) Develop and test different models and benchmark them with baseline and amongst themselves, using adapted technical framework and metrics, in order to select the solution fitting best the identified criteria for the final implementation
4) Critically assess obtained results. Identify strength and limitations of obtained models.

Globally, key qualitative objectives for this project are the following, in our view:

1) "Guarantee of results"

   We see a real risk of getting lost in the various possible predictive modelling approaches, countless algorithms, abundant literature, and internet resources. However, it is key that at least some

convincing results are delivered at the end of the project. Here, this will be achieved by first deciding on a global approach and by breaking down the original problem into manageable sub-problems. Prediction accuracy can of course not be guaranteed, on the contrary, it is important to manage stakeholders' expectations from the start.

2) Results reproducibility and traceability

Results must be reproducible, so they can be challenged and to make sure that they are not due to pure chance or errors in the code. Where and how they have been produced must be clearly documented, to address traceability.

1.3 METHODOLOGY

Although the analysts who are this project's stakeholders are only hoping for some conclusive results and fully understand that this piece of work is of exploratory nature, we believe it is key to approach it in a scientific and rigorous manner.

Therefore, the approach used in this project will be to adhere to the best practices of the field, generally used when building ML models, as inspired by our literature review, described in the next chapter.

2 CHAPTER 2: LITERATURE REVIEW

This chapter contains main elements found in the literature we reviewed with objective to identify comparable problems and effective solutions.

2.1 SIMILAR PROBLEMS AND METHODS

Our first intuition was to look within the energy forecasting field. The recent developments of renewable energy sources and smart grids have made the forecasting field very critical, which triggered important research in this area where artificial intelligence techniques have proven their effectiveness (Raza & Khosravi, 2015).

Although some of these problems relate to regression, typically load forecasting as it is very much influenced by calendar (weekdays vs. holidays) and weather (temperature, humidity, sun conditions), most problems are closer to typical timeseries forecasting problems, where few explanatory variables are available but where trends and seasonality are at play, on top of intrinsic dynamic. We note that several techniques have been used with success to address load forecasting problems: MLP, SVM in conjunction with other methods (fuzzy clustering, wavelets, genetic algorithm, ARIMA) depending on the problem nature.

Another review focusing on load forecasting lists additional methods: RNN (LSTM) and CNN, for example associated with k-means clustering, which outperform more traditional methods such as ARIMA and SVR (Almalaq & Edwards, 2017).

In the context of electricity consumption forecasting, some research compares implementation of ANN and SVM, using population and installed capacity as main explanatory variables. Although SVM shows a slightly more accurate prediction than ANN in this case, the authors recommend them both as viable methods (Kaytez, Taplamacioglu, Cam, & Hardalac, 2015)

Interestingly, to cope with very large volumes of data, CNN have been used to automatically identify useful features and k-means clustering used to create train and test datasets that are not unbalanced (Dong, Qian, & Huang, 2017).

About energy prices forecasting, a quite comprehensive survey provides a typology of possible models and

lists main ones used in “computational intelligence”, i.e. AI: they are basically the same as already identified above, i.e. MLP, RNN, SVM (Weron, 2014). The author synthesises their strengths, mentioning the ability to model complexity and non-linearity, beating statistical techniques, but this is also their weakness since it can lead to overfitting. He further mentions that comparing algorithms in general is hard, unless they are using the exact same data.

Still about price forecasting, a comparison study is conducted between deep learning approaches and traditional models, i.e. statistical ones (Lago, De Ridder, & De Schutter, 2018). The authors show how ANN models produce hourly prices values prediction for next days as 24 outputs, using various explanatory inputs (such as day of week) and previous price values. These models use hybrid topology with an MLP part coupled to a LSTM / CNN part. They benchmark 23 models, from statistical without exogenous inputs (ARIMA, GARCH), to exponential smoothing, to statistical with exogenous inputs, to ANN, SVR and ensemble models like random forests. They make the following findings about ANN: optimal activation function in all layers is ReLU. To prevent overfitting, early stopping is used in the training process. Their conclusion is that ANN (MLP, GRU, LSTM) outperform all other models, in a significant manner, perhaps because the prices in question seem to be driven by highly nonlinear influences and have a lot of spikes. Within those models, the ones with more parameters seem to perform the worst.

The area of stock prices is as well one at the root of forecasting research. Recent research compared several multi-variate regression-based predictive models (Jaydip, 2018). The results are consistent with the ones observed above: ANN produce on average the most accurate results, LSTM being ahead by a large margin.

When forecasting several targets with the same model, for example energy price and demand, implementing random forest can be an effective solution (González, Mira, & Ojeda, 2016).

Other business areas provide examples of forecasting problems whose results might be of interest: in the case of tourism, a step by step design procedure is provided to build a MLP model for expenditure forecasting (Palmer, José Montaño, & Sesé, 2006). It explains the process of defining and integrating lagged features in as input of the neural network, which helps producing accurate results and recommends using a minimal number of hidden layers and neurons. It mentions input layer neurons use linear activation functions while hidden and output use sigmoid ones. Authors are using a grid search approach to find the most effective configuration (number of hidden layers between 1 and 3, number of neurons in each layer), providing a best output in an 8 inputs-1 hidden-1 output neuron topology.

2.2 SUPPORT VECTOR MACHINE / SUPPORT VECTOR REGRESSION

In classification problems, SVM can efficiently distinguish subsets in complex data by using hyperplanes, with the help of kernel functions when linear hyperplanes are not sufficient which is often the case in real life datasets (Machines, 2012).

They are widely used in the context of load forecasting (Dong, Qian, & Huang, 2017). They can overcome over-fitting by using kernel functions and regularization techniques. Some studies show, despite both SVM and MLP being well adapted for classification and regression, SVM generalize better than MLP and offers much greater computational efficiency in the case of large data sets (Osowski, Siwek, & Markiewicz, 2004)

Based on this review, we checked into more details potential methods fitting our case:

2.3 RANDOM FORESTS (RF)

RF can be used in a context of regression, i.e. when the output depends on explanatory variables, that can include lagged values of those output variables (González, Mira, & Ojeda, 2016). One of the key features of RF is their ability to measure input variables importance, hence giving some interpretability of provided results. Compared to other algorithms, they are not very prone to overfitting and can handle missing input data.

2.4 MULTI LAYER PERCEPTRON (MLP)

Based on a review of electricity load forecasting research (Raza & Khosravi, 2015), the strength of MLPs stand in their ability to extract complex relationships between input variables. Some of their main drawbacks are their computation power need in the training process and initial weights configuration since the learning algorithm might get stuck in a local minima. Experience shows that ANN forecasts depend mainly on the following parameters: model architecture, activations functions and learning algorithm, as well as additional exogenous inputs. Additionally, drop out layers are used as a regularisation tool in order to reduce overfitting (Lago, De Ridder, & De Schutter, 2018)

Concerning short term load forecast, experience shows that they can outperform traditional statistical methods as seasonal Holt-Winters models or shallow neural networks (Ryu, Noh, & Kim, 2016).

2.5 LONG SHORT TERM MEMORY ARTIFICIAL NEURAL NETWORKS (LSMT)

LSMT are used in a wide variety of cases (Almalaq & Edwards, 2017), typically when large volumes of data are available, for example in the case of hourly resolution timeseries.
They can retain complex time related patterns over long sequences (Lago, De Ridder, & De Schutter, 2018). When designing a model, main parameters are learning window size and number of memory cells, on top of other ANN parameters already mentioned.

2.6 MULTI OUTPUT MODELS

Using a single model to predict several quantities can provide better results than several single output models, as shown in the case of price and demand of electricity (González, Mira, & Ojeda, 2016). The assumption is that output variables depend on the same set of input variables and the joint model can take advantage of the outputs's correlation. Here one of the key parameters is to set relative weights of the targets in the joint model (demand and price in this research).

2.7 METRICS

Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) seem to be widely used metrics (Almalaq & Edwards, 2017) as indicators of performance to compare models with each other, concerning the forecasting accuracy.

2.8 SUMMARY

From this literature review, we gather the following takeaways:

- The problem in question seems unusual: none of the cases met is exactly similar to it, mainly because of its "power blocks" and related optionality.
- There is no clear rule as to what model to use in what case and no single best solution for everything. Only experimenting will deliver a verdict.
- Identifying all possible exogenous influences, lagged variables or other engineered variables is key in improving accuracy
- The list of model candidates to be implemented should include the following, as they all proved successful in one case or another: SVR, MLP, RF and LSTM. Hyperparameter tuning plays a

determining role concerning results quality.

- Multi-output regression can be achieved using with RF and ANN
- Concerning ANN, adding neurons layers doesn't improve results, it only leads to more overfitting. 2 layers should be sufficient in our case
- Adapted error metrics are RMSE and MAE

3 CHAPTER 3: METHODOLOGY

3.1 INTRODUCTION

As we have deducted from our literature review, solving the problem in question will not be a simple application of a given method with guaranteed results, as no problem precisely matching ours could be identified. On the contrary, we will need to apply an exploratory approach, based on the elements gathered in the literature review, in terms of methods and models, as we will explain in the next paragraphs.

3.2 GENERAL APPROACH

Overall, we will put in practice a standard data science methodology, namely "CRISP-DM" (Azevedo & Santos, 2008). This consists in the process depicted below, whose steps will be illustrated as they are put in practice in the rest of this report.

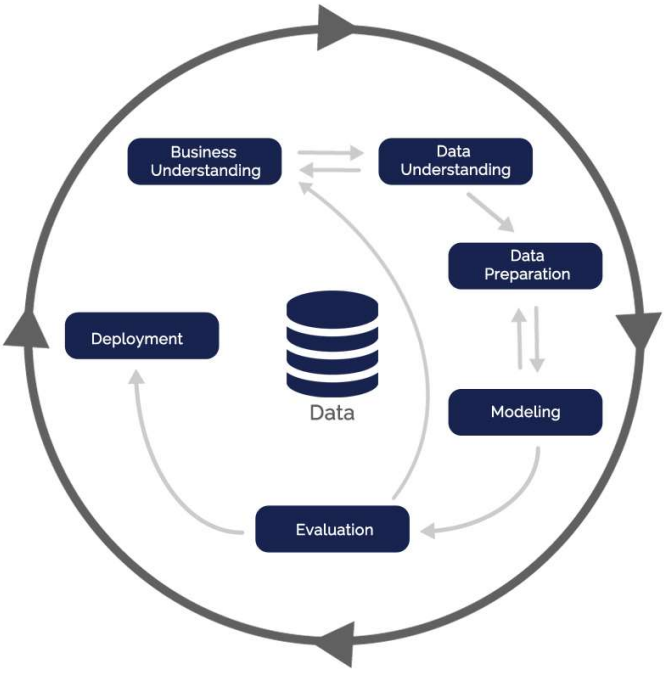


Figure 4. The CRISP-DM life cycle (Azevedo & Santos, 2008)

We note that this life cycle is no linear but cyclical and, as arrows show, not only are there dependencies between phases, but the process allows to go back and forth between them, as learnings gathered in a given phase can trigger improvement or corrections to the previous and next one, which fits well the exploratory

nature of our task. Chapter 4 will describe each of the implementation steps in the context of our project, as "business understanding" has been covered in chapter 1 already.

We believe this method is particularly adapted to our context since it stresses the importance of business understanding: indeed, in our case, it is key to grasp business requirements and processes in depth. Otherwise, key intuitions would be missing, many choices in the project could not be decided and the end result would likely not fit stakeholders' expectations, hence reducing greatly the project value.

One more step could be added to this methodology, in our view namely the "Explanation" step, which would come before deployment, and close the loop with the business stakeholders, as it is not sufficient for them to receive accurate results, they need to understand them so they can trust them.

3.3 STAKEHOLDERS RELATIONSHIP

To fulfil the "business understanding" step, we have been able to count on the close collaboration of the project's stakeholders (analysts), who not only provided the data, requirements and baseline prediction but also were available to answer our questions in detail. Additionally, they shared some their intuitions which were key in guiding our analysis and modelling work.

Outside of usual electronic means of communication, a physical meeting took place in July in order to validate scope and goal, discuss assumptions and present initial results. The feedback gathered as a result was very valuable, for example about understanding the importance of predicting power blocks and not only daily energy. A second meeting is planned in September to present our final results, which will constitute our "explanation" step.

3.4 PROBLEM DECOMPOSITION

3.4.1 Power blocks dual representation

Power blocks are provided in the original dataset in the form of value pairs. This information can be represented in a 24 hours vector equivalently and unequivocally:

Block	Power	No hours

1	74	4
2	66	4
3	43	8
4	0	0

Hour	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**…**	**24**
Power	74	74	74	74	66	66	66	66	43	43	43	43	43	43	43	43	0	0	0

Figure 5. Power blocks: value pairs and vector equivalents

In this example, the 3 power blocks are first displayed in the condensed original form and then their 24 hours vector equivalent: the vector is simply constructed by repeating the power values as indicated by the number of hours.

This equivalency is important for the rest of the project as this is one of our prediction targets, but both forms are adequate.

3.4.2 Detailed problem decomposition

The “all in one” approach would be to design one large multi output model generating all outputs at once. Although possible, how can we force it to produce results that respect the given constraints?

In order to tackle this challenge, we propose to split the initial problem into cascaded sub-problems. Treating several subproblems enables to reduce complexity and risk, by addressing more manageable decoupled work packages. We believe cascading can help impose a constraint to the sub problem, by integrating it as an input. In our case, the goal is to predict the high-level value first (maximum energy) then use this as an input to predict the constitutive elements, i.e. the power blocks:

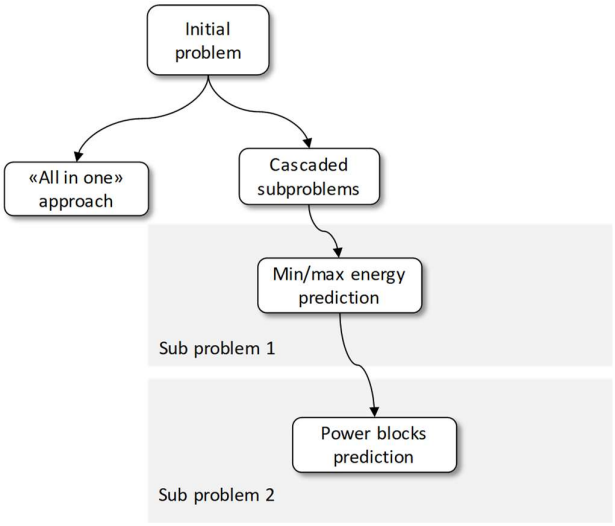


Figure 6. High level problem decomposition

At this stage, the first subproblem is a standard regression problem with 2 targets, based on the explanatory variables. The second can be treated as regression problem too, although with numerous targets, still using the same explanatory variables augmented with the result of the first sub-problem.

An important assumption can be exploited here: power blocks seem to have a limited number of configurations used in practice. Therefore, we can think of simplifying our targets set by clustering them, thus turning the 8 targets regression problem into a classification one. This enables respecting the constraint about number of hours being whole numbers, if we make sure the clusters' centres do respect it. Moreover, if we scale them, we can respect the "maximum energy=sum power*hours" constraint, i.e. by norming them, all we'll need to do will be to scale them back using the predicted maximum energy, to preserve the predicted maximum energy and number of hours:

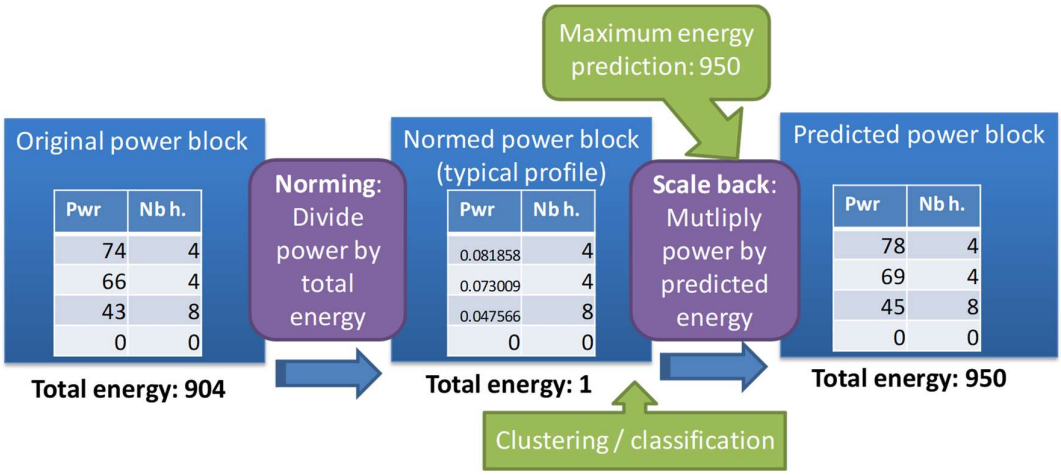


Figure 7. Power block norming and scaling to total energy prediction

To exploit the power blocks duality, i.e. either value pairs or 24 hours vectors, we propose to investigate both forms: perform clustering on targets with one representation and with the other, since distance measuring will certainly yield different results in both cases.

Globally, we come up with the following problem decomposition:

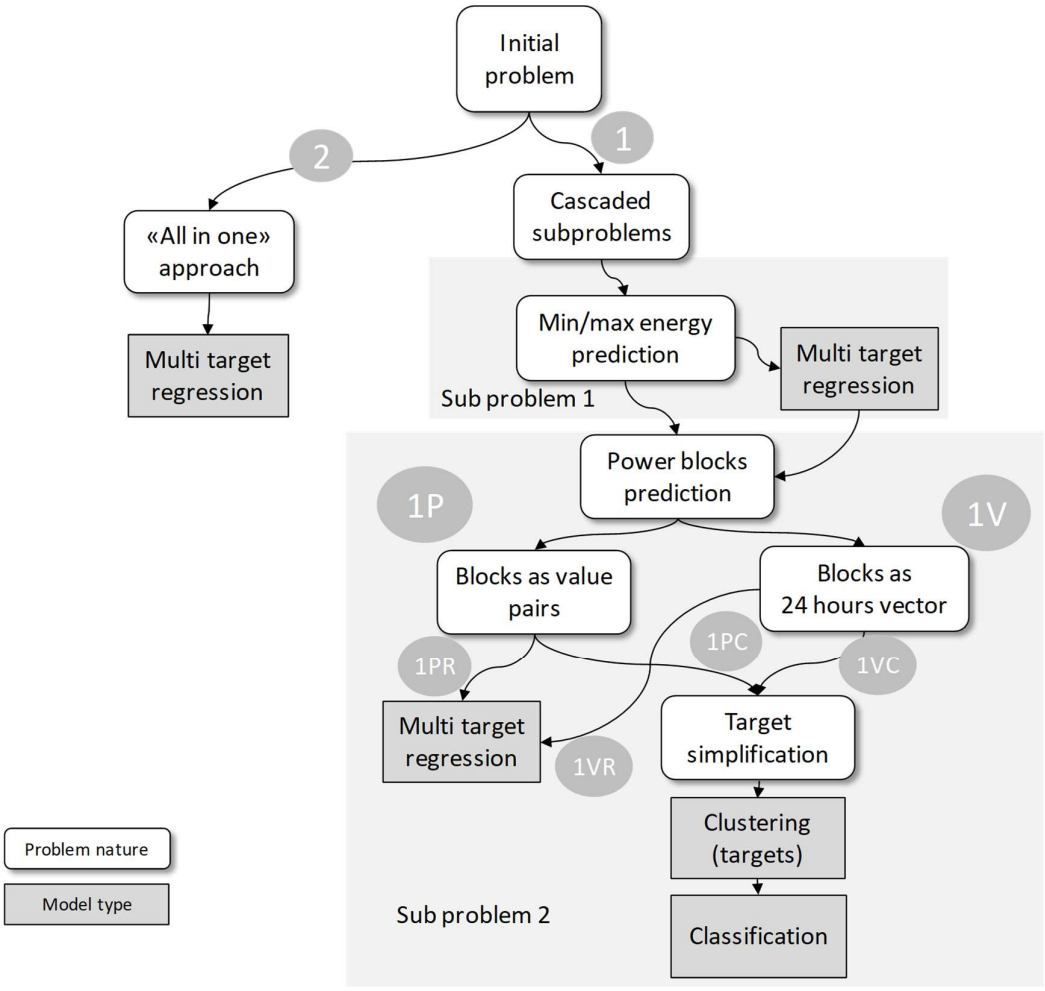


Figure 8. Complete problem decomposition

This might seem convoluted not to say overcomplicated. However, all proposed assumptions need to be tested if we want to find the optimal solution. Each path has its strength and weaknesses and seems to have a fair chance of being successful: only experiment will tell which is the most efficient.

Therefore, we will investigate paths from the simplest to the more complex where it makes sense. We note already that the cascaded approach (1) is also the less opaque one, since it produces intermediary results that

can be consulted independently and whose quality can be assessed as an influential factor on the second subproblem results' quality.

In subproblem 1, point will be to implement several of the identified ML models (RF, SVM, MLP, RNN) treating the target separately and comparing the results in terms of accuracy and interpretability.

In subproblem 2, to predict the power blocks, we will compare multi output regression (1P) with "clustering / classification" (1V). The validity of the latter approach has been shown in a univariate context, i.e. as "regression by classification" (Torgo & Gama, 1996), mainly using k-means clustering. Here it would bring key benefits, as it would allow to generate predictions that respect the given constraints: since we will use the k-medoids clustering method to output centres that are actual data points, not virtual ones, hence respecting constraints by essence. We will use grid search to find an optimal number of clusters.

Concerning multiple targets regression, we identified in the literature review two adapted models: MLP and RF. They offer no way to constraint the number of hours results to be entire numbers, which can be solved by applying an adapted rounding method.

3.5 IMPLEMENTATION

Concerning the implementation part, we will apply the steps illustrated below, that provide details of the previously defined implementation phases (top line).

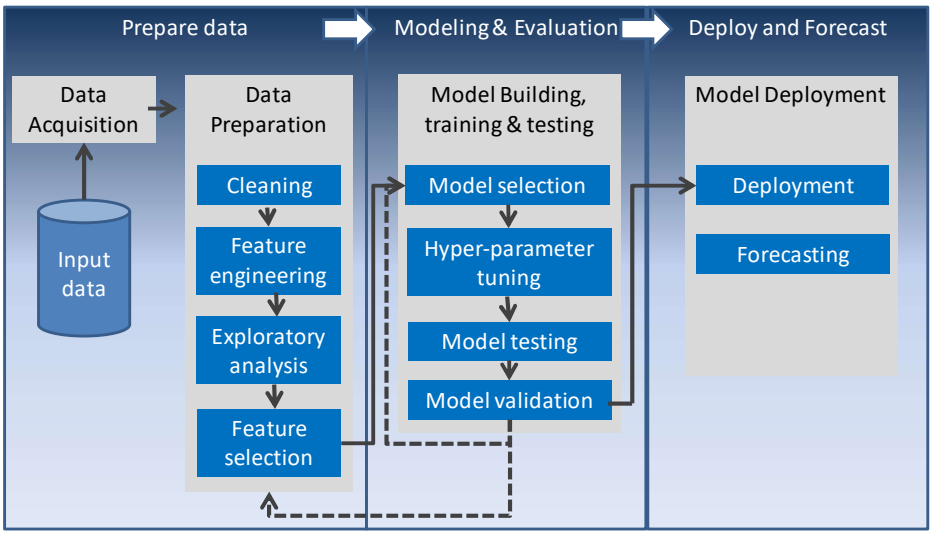


Figure 9. Implementation steps

Again, the steps are very standard in a data science project, will no describe them further, but underline a

few key considerations in the next paragraphs.

3.6 HYPERPARAMETER TUNING

Hyper parameter tuning is a core element in the implementation. Generally, we will use a grid search approach, using usual parameters taken from the literature as far as possible and evaluate the best overall configuration for each model candidate.

3.7 TIMESERIES ASPECTS

Although this problem is mainly solved by regression/classification models, i.e. no timeseries statistical model will be investigated, timeseries aspects still need to be considered in the implementation:

1) lagged features will be calculated where it sounds sensible
2) we will experiment a model that can integrate the time dimension, i.e. RNN, to observe how much gain this can produce in our predictions
3) train / test split will be done so it respects temporal order of data points, i.e. preserving the data time structure, as shown below

Figure 10. Timeseries train / test split

3.8 TESTING STRATEGY

Good practice when tuning and testing a model is to split the dataset in 3 parts: on top of the training data, it should be divided into validation and test subsets. The goal is to tune the model on the validation subset and make a final quality measure on the test subset, hence getting a truly objective performance measure since calculated on data never seen, not for learning nor for tuning.

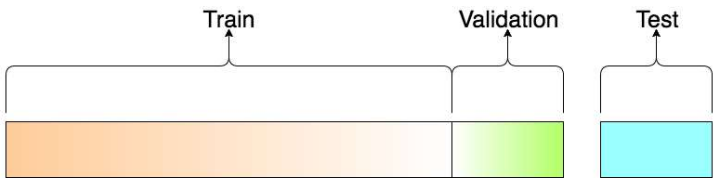


Figure 11. Recommended train / validation / test split

In this project, as the volume of data is rather low and contains very few exceptional events, we took the

decision to stick to a more simple approach, defining only train and test subsets, which allows to have more data for learning and testing and occurrences of exceptional events in both. Model tuning and performance measures will be done on the same subset, which implies performance might be overestimated.

3.9 METRICS

A key element in the implementation is to pick adapted performance metrics to compare and rank our models results as it is closely related to the cost function to minimize when tuning the models. As seen in our literature review, RMSE is a good candidate in the regression problems to measure overall error, therefore this will be our reference metric.

We will still compute MAE, which is less sensible to outliers, and R^2. While both and RMSE have no meaning on their own, i.e. they must be compared to the values targeted, R^2 indicates the proportion of variance explained by the model.

3.9.1 Error measure: energy vs. power

Whereas measuring error on the daily minimum and maximum energy values is straightforward, measuring error on the forecasted power blocks is trickier, because of the form blocks have (value pairs). As hinted by the project stakeholders, an appropriate way is to compute error on the hourly representation of the blocks, i.e. the 24 hours vector containing power values.

Using the same example as before, we now consider the corresponding fictitious forecast, delivering only 2 blocks. If we transform it into 24 vector shape, it becomes straightforward to compute a RMSE on the hourly values (as any other error measure).

Block	Power	No hours
1	74	4
2	66	4
3	43	8

Hour	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	…	24
Power	74	74	74	74	66	66	66	66	43	43	43	43	43	43	43	43	0	0	0

Figure 12. Original power block dual representation

	Power								No hours							
Block no	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
Gr. truth	74	66	43	0	0	0	0	0	4	4	8	0	0	0	0	0
Forecast	70	40	0	0	0	0	0	0	8	8	0	0	0	0	0	0

Hour no	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Gr. truth	74	74	74	74	66	66	66	66	43	43	43	43	43	43	43	43	0	0	0	0	0	0	0	0
Forecast	70	70	70	70	70	70	70	70	40	40	40	40	40	40	40	40	0	0	0	0	0	0	0	0

Table 1. Ground truth (original) and forecast power blocks – value pairs and hourly vector equivalent

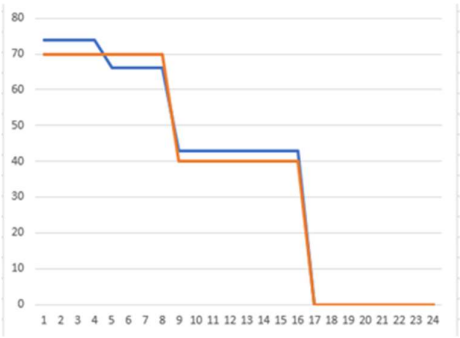


Figure 13. Actual vs. predicted power blocks values, hourly plot

Plotting the hourly values, we notice that the prediction is actually close to the original values, despite delivering only 2 blocks as opposed to the 3 original ones.

Here we underline that this principle only works when the blocks' power values are sorted by decreasing value order, which is not the case in the original file, so we need to fix this.

3.9.2 Error measure weight on several targets

Ultimately, our model's predictions must minimize error on its 3 targets. From our stakeholders' point of view, none has greater importance over the others, therefore we'll use equal weights when assessing the solution globally.

3.10 TECHNICAL FRAMEWORK

In order to deploy the solution in the target environment, and in accordance with our stakeholders, we picked the Python language as it is commonly used and adapted to data science implementation, thanks to its numerous libraries. On top of the usual libraries, worth mentioning are:

- "Keras" for deep learning models implementation, because of its simplicity

- "scikit-learn" for other ML models, pipeline and grid search functions
- "LIME" (Local Interpretable Model-Agnostic Explanations) for results interpretation

In order to reach our objectives of reproducibility and traceability, we will be developing our solution Jupyter notebooks for its integrated documentation capabilities which allow to build a clear and illustrated narrative. We therefore try to produce code that is easily linearly readable, i.e. not too condensed nor too obfuscated. The challenge is to find the right balance between modularity, making code more compact and readability, hence avoiding the "spaghetti effect". From our experience, to overcome notebooks' drawbacks, some addons are needed to work efficiently, mainly "table of content" to ease navigation in large documents.

Still very large notebooks can become slow to work with, for this reason, we decided not to write all the code in one large Jupyter "report", but to split it, as examined below.

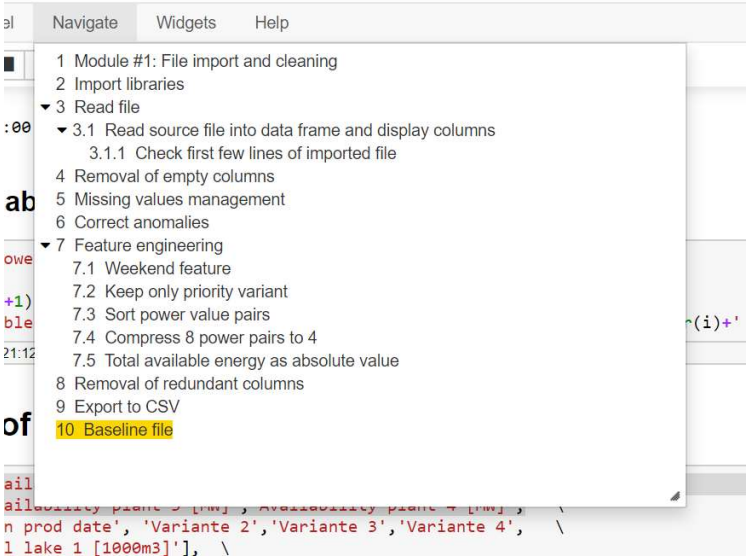


Figure 14. Juypter navigation add-on

3.11 CODE ARCHITECTURE AND GOOD PRACTICE

As our code volume is rather consequent, we decided to decompose it into "modules" of manageable size, both for writing and reading, that match the general methodology steps, i.e.

1. "Data preparation" implements data import, cleaning and initial feature engineering. Saves result in a clean CSV file for other modules to import.
2. "Data exploratory analysis" implements data plotting, statistical analysis and unsupervised learning method
3. "Subproblem 1. Modelling and evaluation" implements regression models for minimum and

maximum energy targets. Saves result to a CSV file for Step 2 import.

4. "Subproblem 2. Modelling and evaluation" implements power block prediction, using the maximum energy previously calculated

In terms of good practice, we will pay attention to setting the random seed to a chosen value in each experiment implying randomize steps (RF, ANN weights initialisation), in order to make results reproducible.

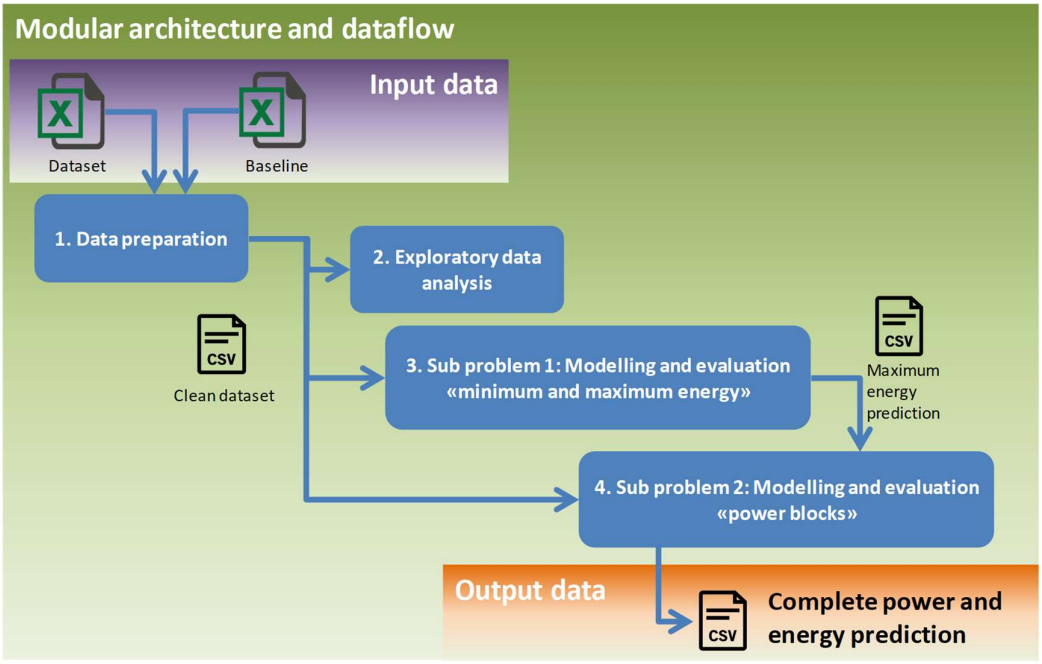


Figure 15. Modular architecture and dataflow

3.12 SUMMARY

In this chapter we have seen how to approach the given problem, in a systematic way so results could be guaranteed. The following chapter describes the actual implementation steps where this is put into practice.

4 CHAPTER 4: DATA ANALYSIS

In this chapter, we will be presenting the outcome of our data analysis, from gathering the data to building and evaluating predictive models, according our chosen methodology.

4.1 DATA PREPARATION

In our experience, real world dataset are usually "messy" for several reasons: exotic formats, various files that are hard to join, missing values, erroneous values or unstructured data. Therefore, in order to be able and use the data we first had to import and clean the data file. As we will see, we are lucky to have a rather clean data file to start with.

4.1.1 Gathering the data

The data file was provided to us in MS Excel format, after it had been anonymized, therefore no ethics approval was required (see appendix). Several versions were provided, as more data became available through the project and a minor issue was corrected at the source. Another file was provided that contained the baseline prediction in a very similar format.

4.1.2 Initial dataset presentation

The dataset originally contains 1917 lines and 94 columns, of which 6 are empty. Lines correspond to daily time points, from 2014.04.01 until 2019.06.30.

4.1.3 Cleaning the data

To easily re-import several versions of the input, the cleaning task is implemented in Python and not performed manually. It consists mainly in removing empty columns and renaming some to give them more explicit names. As well, we remove redundant information: for example, lake contents are provided in absolute value and percentages, same for turbine values. We kept only the later.

4.1.4 Handling missing data

Very few data occurrences are missing in the original file. We fill the gaps by using linear by linear

interpolation, as this sounds as the safest option from looking at the surrounding occurrences.

4.1.5 Data quality / inconsistencies

At this stage, we check for obvious inconsistencies in the data, as defined by the columns semantic. Lake level value should always remain below the maximum lake level constraint. As there are only a couple of days were the constraint is broken, which sounds like obvious erroneous values, we correct these values using surrounding ones and linear extrapolation.

Some other inconsistencies are spotted, for example maximum production being significantly below minimum production. As the reason why is not clear, we choose not to fix them.

4.1.6 Data simplification

Within the original file, there are 16 columns containing the 8 power blocks values (power and number of hours). After discussion with the stakeholders, this number is decreased from 8 pairs to 4 for simplification reasons, as this does not impact the problem definition much, meaning no useful information is lost in the process. Indeed the 4 last columns are empty most of the time. The baseline prediction uses 4 pairs as well.

To perform this simplification, we first sort the column pairs by decreasing power values, then aggregate the energy in the last 5 pairs of columns into the 4th pair: the hour value is set as the sum of the 5 pairs, while the power value is calculated as the weighted average of the power values, so the total energy is conserved in the process. For example:

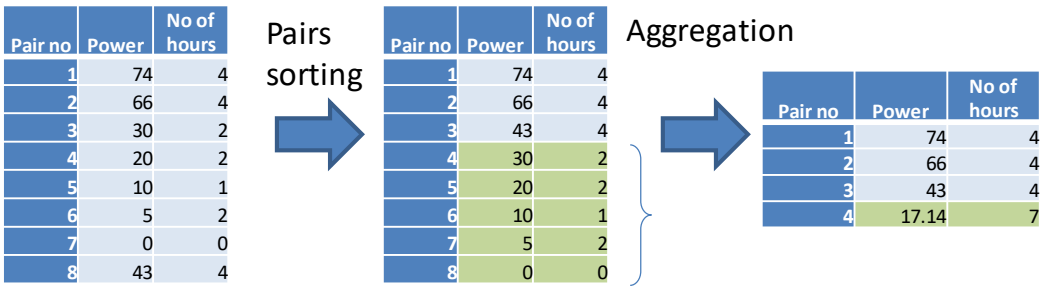


Figure 16. Power block simplification example

This helps fight the “curse of dimensionality” that impacts many aspects of data science projects: computation time, code complexity, interpretability and algorithms performance, since Euclidian measure

becomes inefficient as a distance measure when the number of dimension increases.

Another simplification concerns the number of variants, brought down from 4 to 1 (see appendix for more details).

4.1.7 Initial feature engineering

The first necessary feature to calculate is the maximum daily energy, which is the sum of block power values times the number of hours.

Then, based on the stakeholders' intuition, a categorical feature is introduced that is "true" if the date is a weekend, "false" otherwise, since they believe it could play a role.

Other additional features will be identified when we get a better feel of the data, which we are now going to look at in detail.

4.2 EXPLORATORY DATA ANALYSIS

At this stage, the dataset has been cleaned up and is ready for processing. The next step is to perform exploratory data analysis (EDA), where our goal will be to investigate the data looking for outliers, patterns, correlations or anomalies to gain useful insights for later steps. In one word, this is where we are getting intimate with our data. To do so we will compute statistics, produce plots and implement unsupervised learning models.

4.2.1 Basic statistics

We first take a look at basic statistics of the dataset's features, to get an initial feeling.

4.2.1.1 Input features

We notice we have only 11 input features containing daily values, that we can group in the following categories:

1) Lake inflows in cubic meters (one for each of the 4 lakes)
2) Lake volume in percentage of full lake (only provided for lake 1)
3) Lake constraint, i.e. maximum lake volume level in cubic meters (only provided for lake 1)

4) Powerplant constraints, i.e. availability in percentage of full availability (for the 4 powerplants)
5) Weekend indication, indicating if the date is a weekend or not (1 in this case)

	Inflow lake 1 [m3]	Inflow lake 2 [m3]	Inflow lake 3 [m3]	Inflow lake 4 [m3]	Vol lake 1 [%]	Max lake 1 [1000m3]	Availability plant 1 [%]	Availability plant 2 [%]	Availability plant 3 [%]	Availability plant 4 [%]	Weekend
count	1917.00	1917.00	1917.00	1917.00	1917.00	1917.00	1917.00	1917.00	1917.00	1917.00	1917.00
mean	279.38	55.39	172.24	86.18	0.44	26305.37	0.71	0.89	0.86	0.77	0.29
std	393.60	89.28	155.39	100.92	0.31	9228.22	0.42	0.14	0.31	0.36	0.45
min	-210.00	-482.00	-208.00	-224.00	0.00	0.00	-0.00	0.00	0.00	0.00	0.00
25%	42.00	8.70	69.00	37.30	0.15	30000.00	0.50	0.83	1.00	0.50	0.00
50%	119.00	31.00	119.40	59.30	0.43	30000.00	1.00	0.94	1.00	1.00	0.00
75%	376.00	75.00	228.00	116.00	0.70	30000.00	1.00	1.00	1.00	1.00	1.00
max	3951.60	471.00	985.60	1349.70	0.97	30000.00	1.00	1.00	1.00	1.00	1.00

Table 2. Input features basic statistics

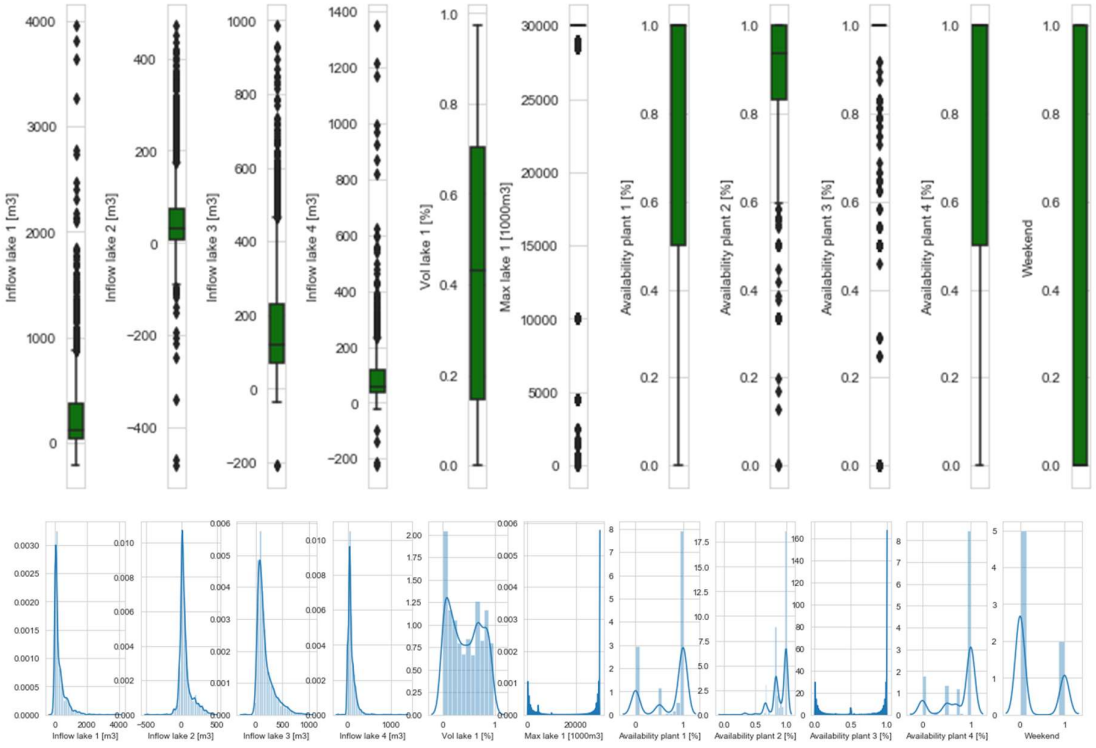


Figure 17. Input features statistics plots

Concerning input features, we see that inflows have a distribution very much skewed left, with some surprising negative values. Lake volume is skewed but not as much. Maximum lake value and plant availabilities have most of the time a maximum value, hence multi-modal type distributions. From this we cannot extract much useful information to solve our problem yet. Although in general ML methods work more efficiently with inputs that follow normal distributions, we don't see any reason to improve this at this stage.

4.2.1.2 Output features

We observe that we have only 10 output features, which might appear as a lot compares to 11 input features! The output features can be grouped in the following categories:

1) Minimum energy production in MWh
2) Maximum energy production in MWh
3) 4 power block pairs of values, each pair contains a power value in MW and a number of hours (between 0 and 23)

	Min prod	Max prod	PrioH1	PrioP1	PrioH2	PrioP2	PrioH3	PrioP3	PrioH4	PrioP4
count	1917.00	1917.00	1917.00	1917.00	1917.00	1917.00	1917.00	1917.00	1917.00	1917.00
mean	228.44	696.30	4.68	59.03	4.87	42.46	3.86	24.06	2.52	8.06
std	194.11	389.93	3.32	16.40	3.95	21.27	4.31	21.59	4.16	14.10
min	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
25%	90.00	362.10	2.00	52.50	2.00	27.00	0.00	0.00	0.00	0.00
50%	180.00	748.50	4.00	65.40	4.00	47.40	2.00	24.00	0.00	0.00
75%	300.00	933.60	7.00	69.90	7.00	61.50	8.00	45.00	4.00	10.50
max	1470.00	1612.17	24.00	84.00	22.00	75.00	24.00	68.65	20.00	60.30

Table 3. Output features basic statistics

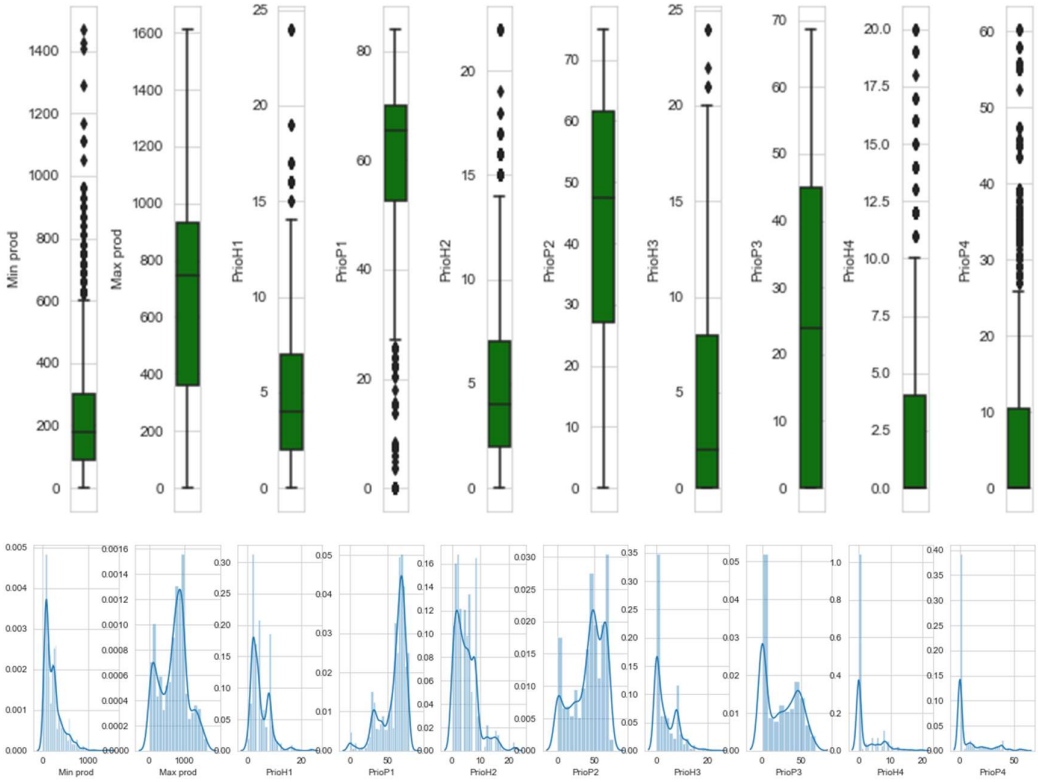


Figure 18. Output features statistics plots

Concerning output features, we see that minimum production's distribution is as expected very much skewed to the left. Maximum production seems to have a multimodal distribution pattern. Same applies to the power blocks values generally. Looking at the 4th pair we can confirm most values are at zero, hence values are concentrated on the first 3 pairs. Interesting will now be to look at these features individually on a time axis, then check correlations between them.

4.2.2 Individual Features analysis on time axis

4.2.2.1 Minimum and maximum production

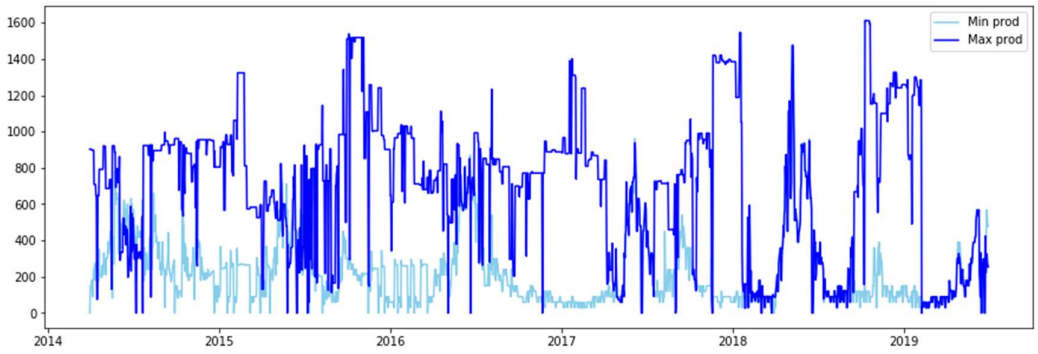
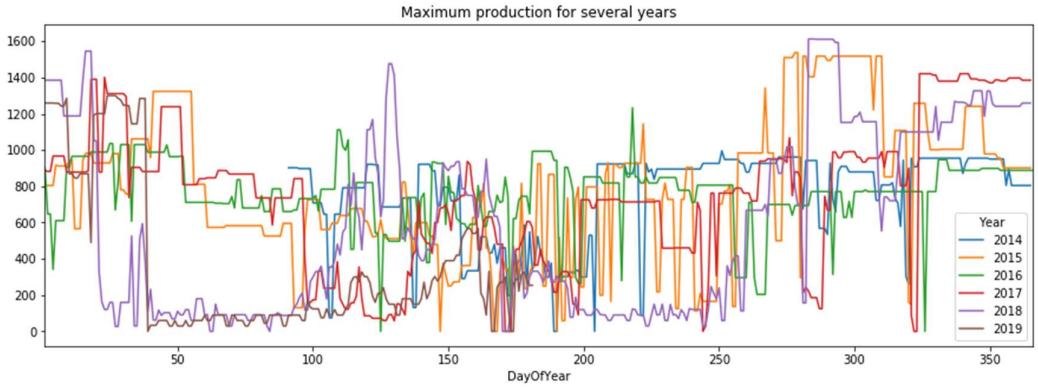


Figure 19. Minimum and maximum production over entire interval



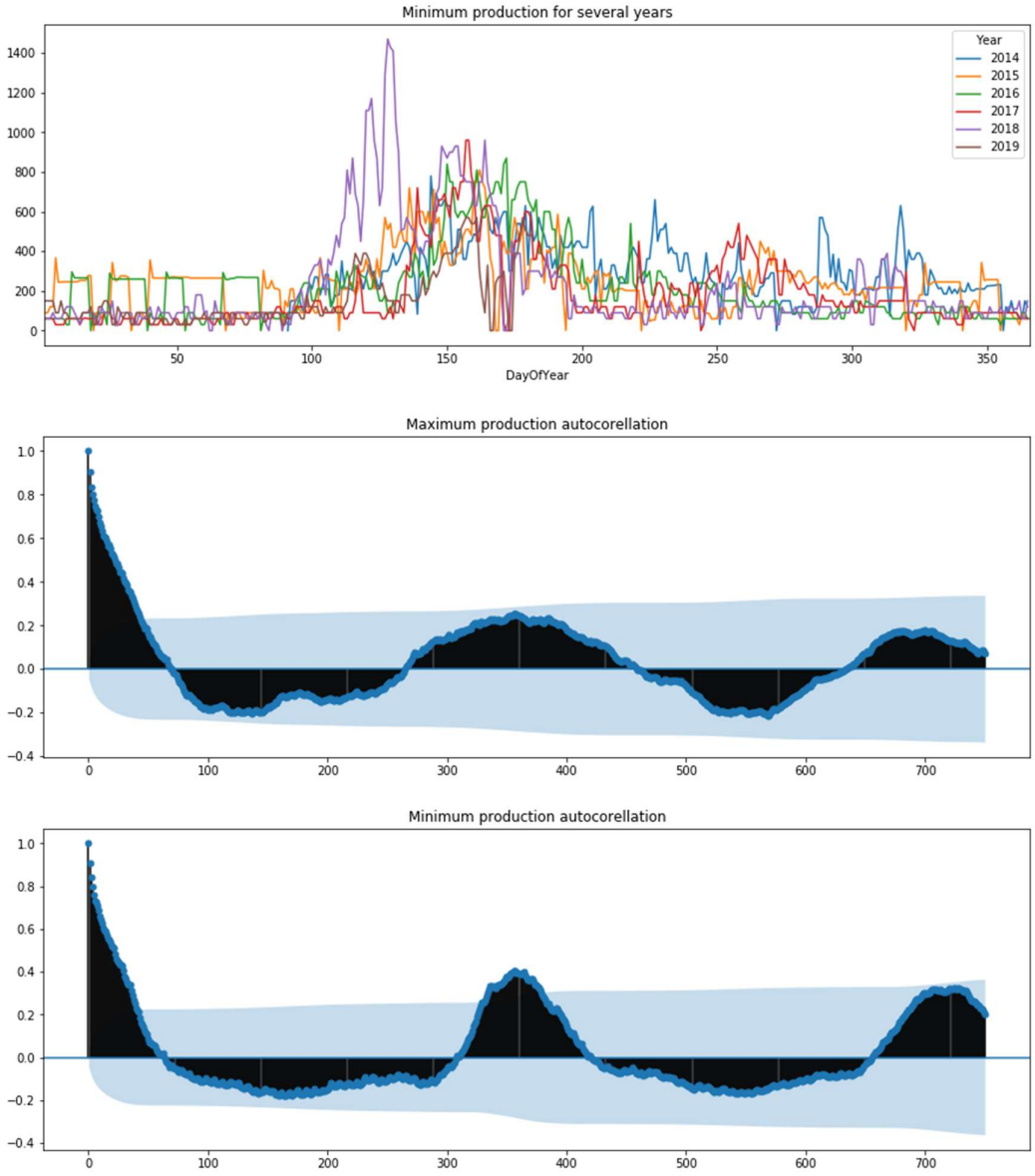


Figure 20. Minimum and maximum timeseries plots and ACF plots

Concerning minimum energy production, we notice a yearly seasonality, visible when looking at the yearly plot and confirmed by the autocorrelation function plot (ACF). Minimum production is highest in the summer month, when temperature is high, which produces important inflows.

About maximum energy production, the curve is much more erratic, no clear pattern can be observed, which the ACF indicates as well. To interpret the ACF plot, we take into considerations points if they are outside of the confidence interval (light blue).

When looking at both minimum and maximum curve, we can't spot any common behaviour, outside of the

fact that maximum is interestingly not always larger than minimum, as there are several periods where they seem to be equal, which means that production is tightly constrained during these periods.

4.2.2.2 Power blocks

Power blocks are clearly the peculiar objects in this project. While not unusual in the energy word, their nature seems to be infrequent in data science projects, based our vain efforts to find such examples in the literature.

Plotting them is already a challenge, that we took on like this:

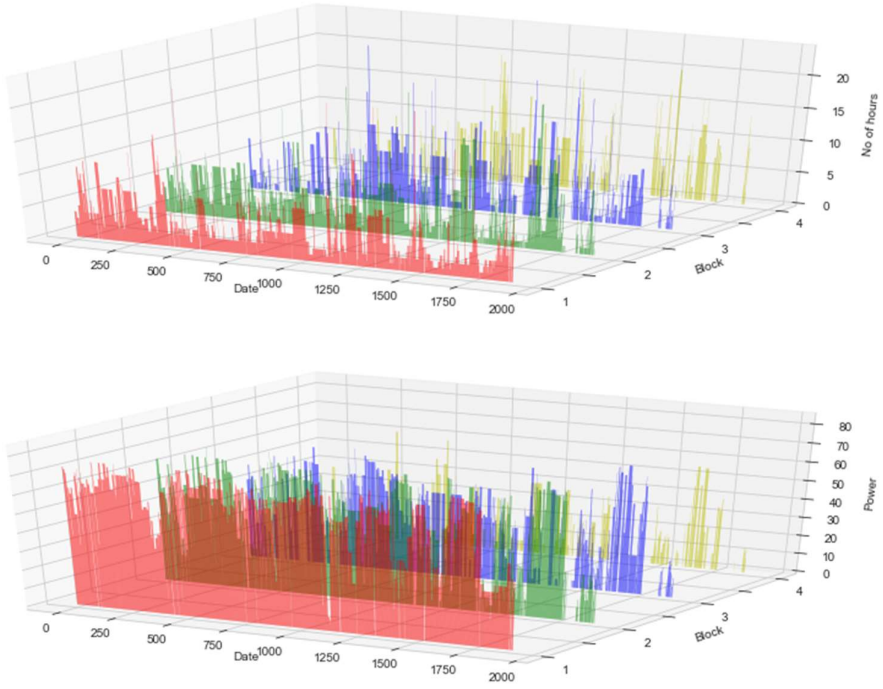


Figure 21. Power blocks 3D bar plots: number of hours (above) and power level (below)

From these plots, we can't see any clear pattern nor tendency. On the contrary the curves seem quite erratic.

Another way to look at this it to plot the energy contained in each block, i.e. power times number of hours, in a stacked bar plot, which confirms this impression.

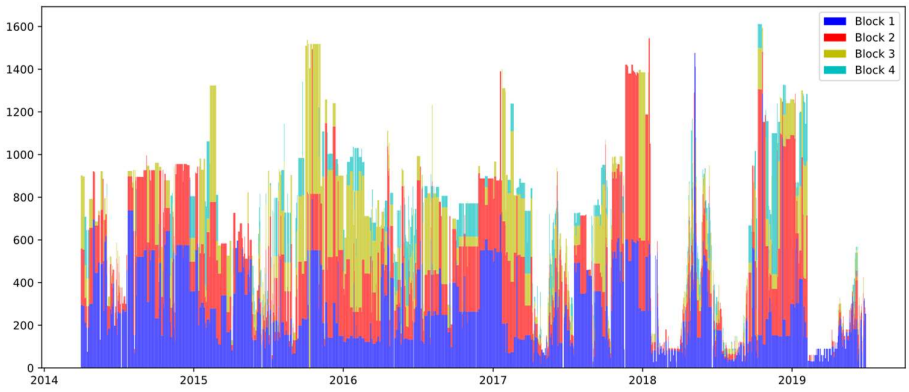


Figure 22. Power blocks' energy timeseries plot

Using the 24 hours vector equivalent, as explained in 3.9.1, we can propose another 3D graphical representation to help getting an idea of what this object really looks like, over some days in 2014 (24 hours vector representation):

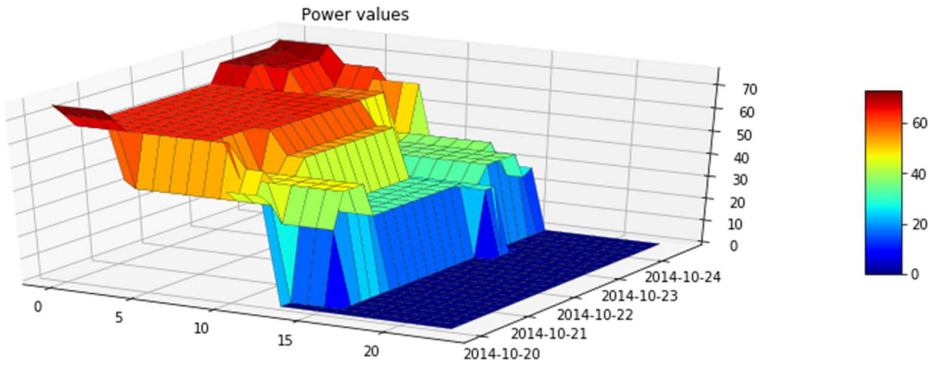


Figure 23. Power blocks 3D surface plot over several days

Finally, an interesting plot to analyse, is the total number of block hours per day. Here it seems that there are different patterns along time, which would hint that the method used to set those numbers might have changed during the period.

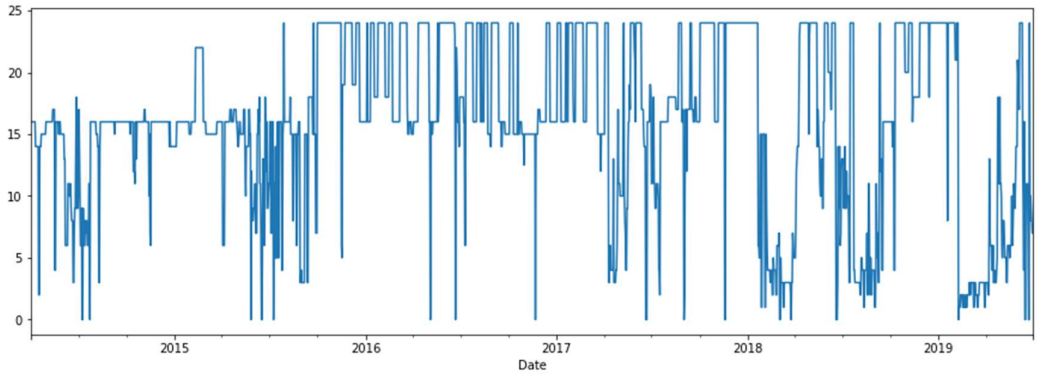
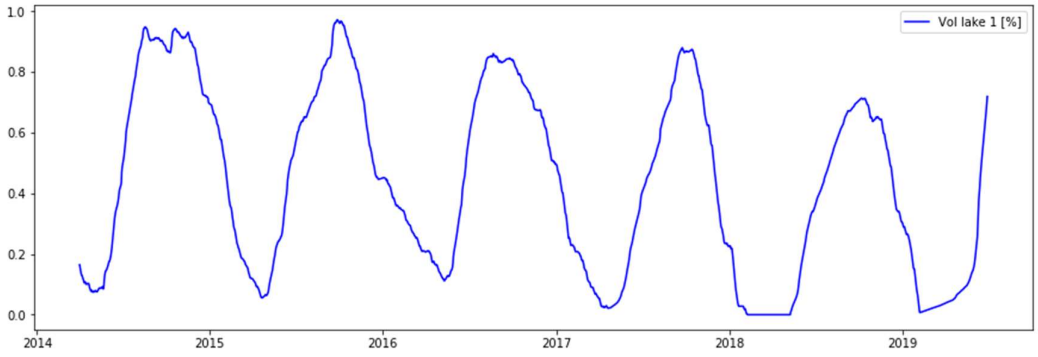
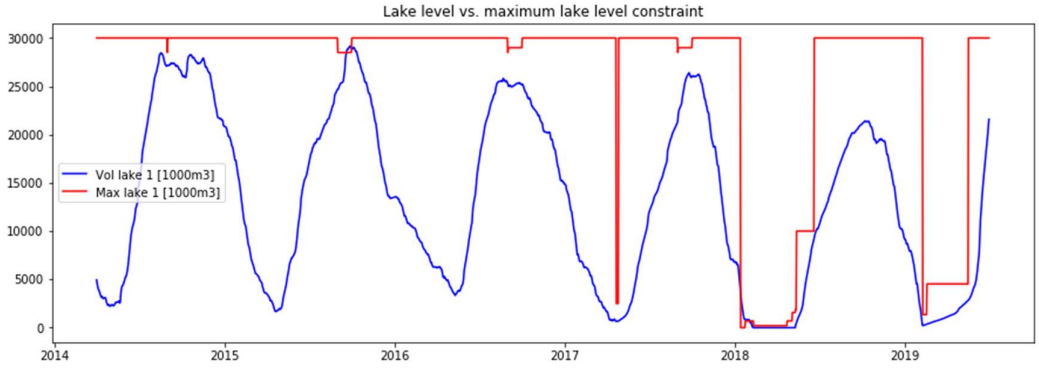


Figure 24. Maximum number of hours timeseries plot

4.2.2.3 Lakes level and maximum constraint





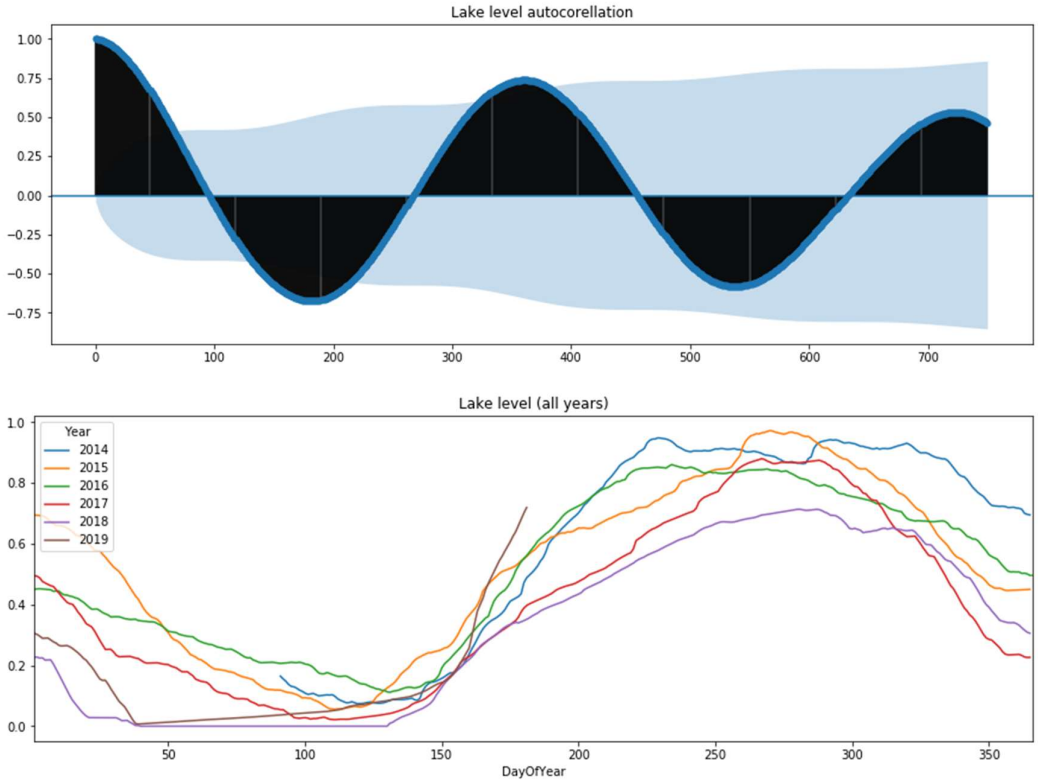


Figure 25. Lake level and maximum constraints. Timeseries and ACF plots.

Concerning lake levels, we notice striking seasonality: lakes are located at high altitude, inflows are more abundant when weather is hot and ice is melting, creating important inflows. In the winter season, market prices are high, inducing important production. It is the reason why lake levels and inflows can be forecasted by our stakeholders with relatively good accuracy.

Considering the maximum level constraint, we see it has the expected influence on the lake level, with a few small exceptions. Looking at the curves, we wonder however how long in advance this signal, i.e. maximum level drop, is taken into account so the constraint can actually be respected. We will see later if a lagged feature introduced in a predictive model can help answer this question.

4.2.2.4 Exceptional events: maximum lake constraints

The maximum lake constraint goes close to zero on 3 occasions only. In that way, they are exceptional events, that still have an important influence. Exceptional events are by definition hard to learn from but it is crucial for our model to take them into account as they will happen again.

We zoom on 2017 and 2018, to check the magnitude of this impact:

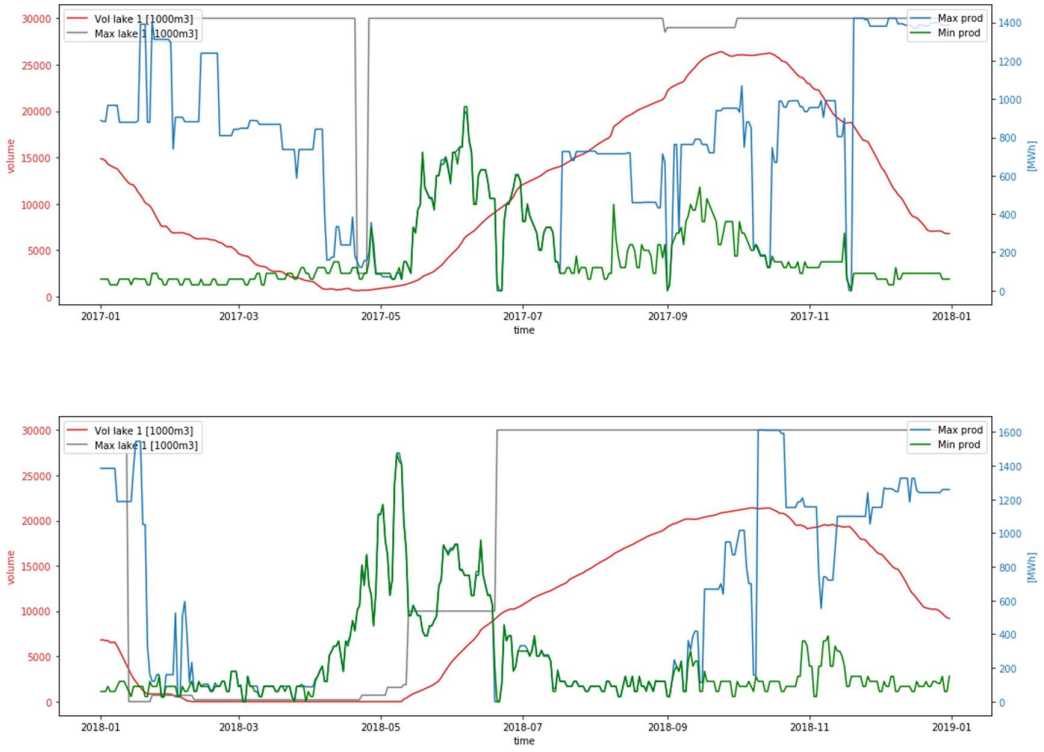
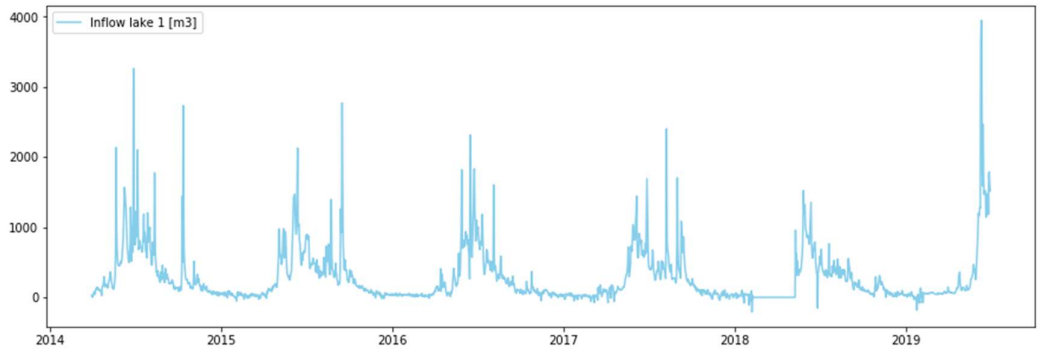


Figure 26. Maximum lake level constraint timeseries plots

Minimum production seems to increase before the constraint is effective, to force emptying the lake. We notice that both times, during the time the maximum level is low and about 3 months after it goes back to full level, the production is constraint, i.e. minimum and maximum values are equal.

4.2.2.5 Lake inflows

We look only at one inflow since they are quite similar in shape and behaviour, from the time series perspective.



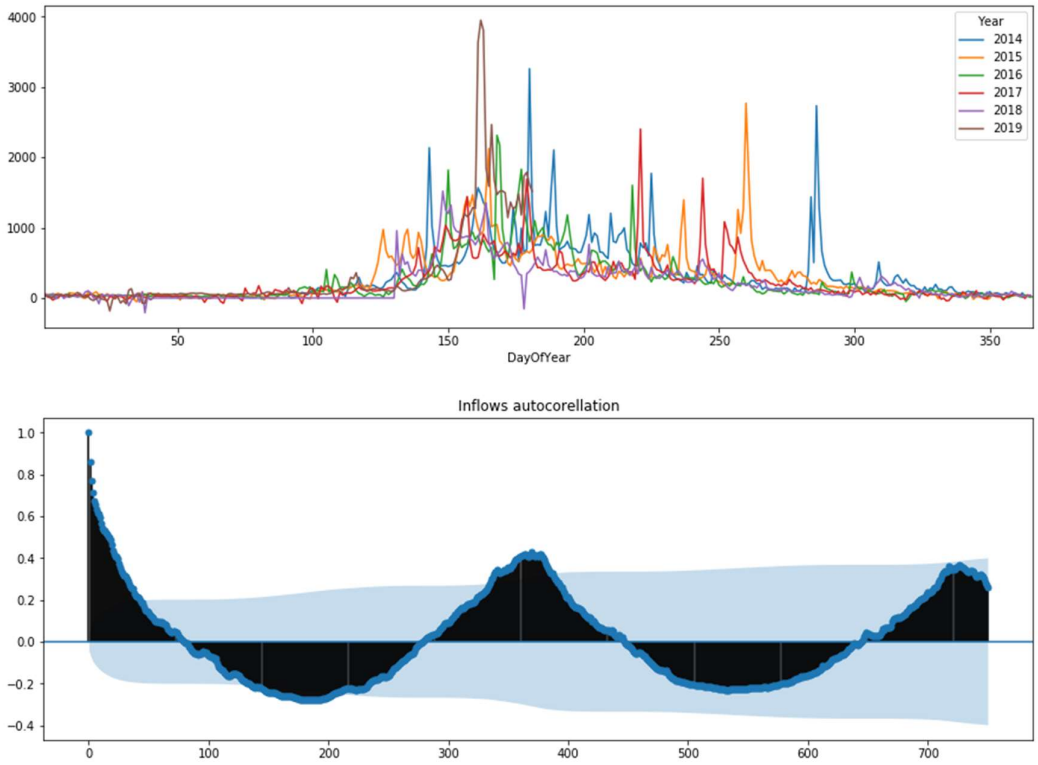


Figure 27. Lake inflows timeseries and ACF plots.

As expected, we witness a clear seasonal pattern here, as already explained when looking at the lake levels, which are directly related.

4.2.2.6 Powerplants availability

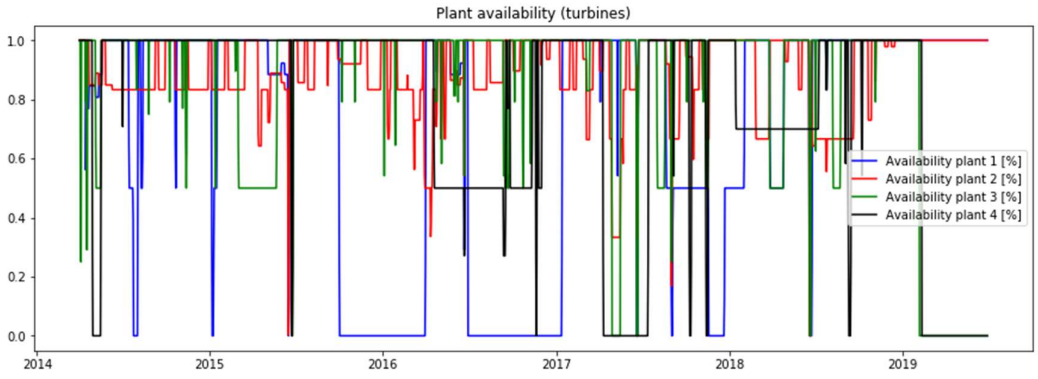


Figure 28. Power plants availability timeseries plot.

Power plants availability is the result of planned human decisions, related to power plants maintenance. As one could expect, we see no obvious pattern here, except the fact that maintenance periods seem to be happening more in the wintertime, when inflows are low.

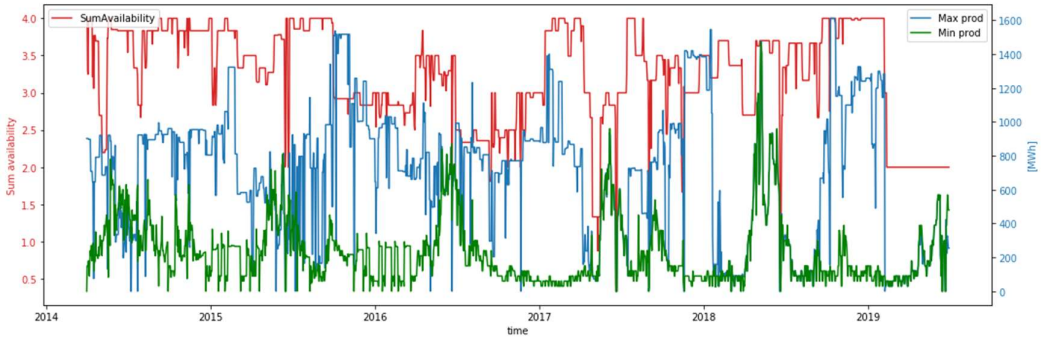


Figure 29. Total powerplant availability vs production plot

Looking at the global picture, i.e. sum of all power plants availabilities, we notice this seems to be influencing the maximum production, as expected.

At this point, we notice the 2019 period seems unusual, since availability is constant and minimum and maximum energy are equal over the whole time interval.

4.2.3 Correlations

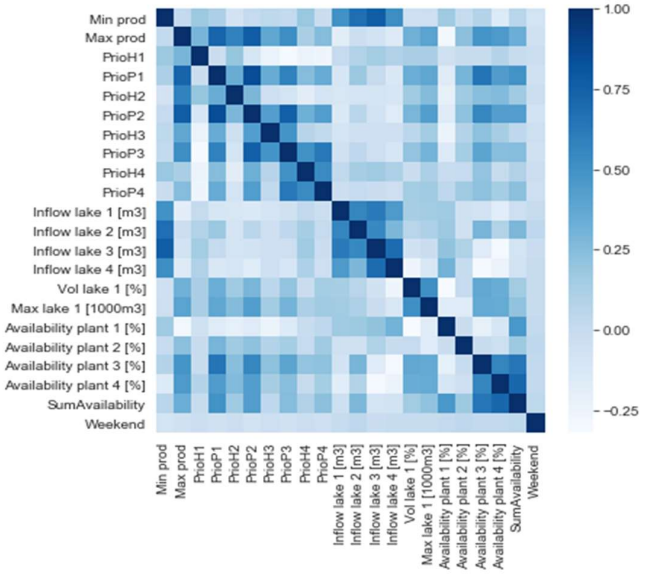


Figure 30. Correlation matrix.

We observe that maximum production seems quite correlated with availability of plants (turbines), which makes perfect sense. Minimum production is highly correlated with inflows, which is in line with the fact that when lakes are almost full, incoming inflows must be used right away for production. Inflows are of course correlated with each other, since the lakes have geographically close to each other.

At this stage, “weekend” does not seem to be influencing any of the other features.

Pairwise scatter plots will help us get a more detailed view on these correlations in the next paragraphs.

4.2.4 Pair wise scatter plots

4.2.4.1 Minimum and maximum production vs. inflows

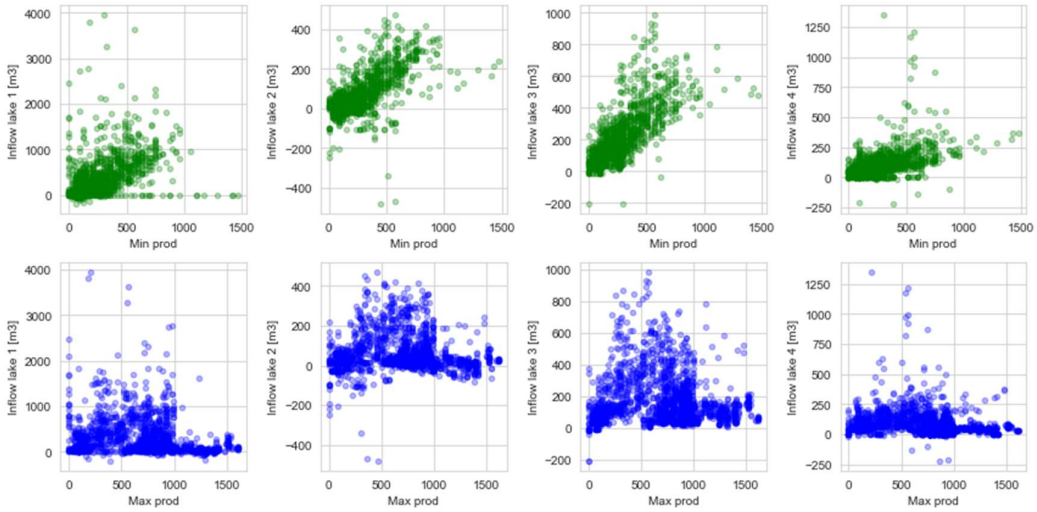


Figure 31. Minimum and maximum production vs. inflows scatter plots

The above scatter plots show us:

- For minimum power: a rather linear dependency between lakes inflows is observed, very visible for lakes 2 and 3, although negative values raise questions. However, for lake 1, many values around hint at a non-linear dependency.
- For maximum power: a dependency is harder to spot.

4.2.4.2 Minimum and maximum production vs plants availability

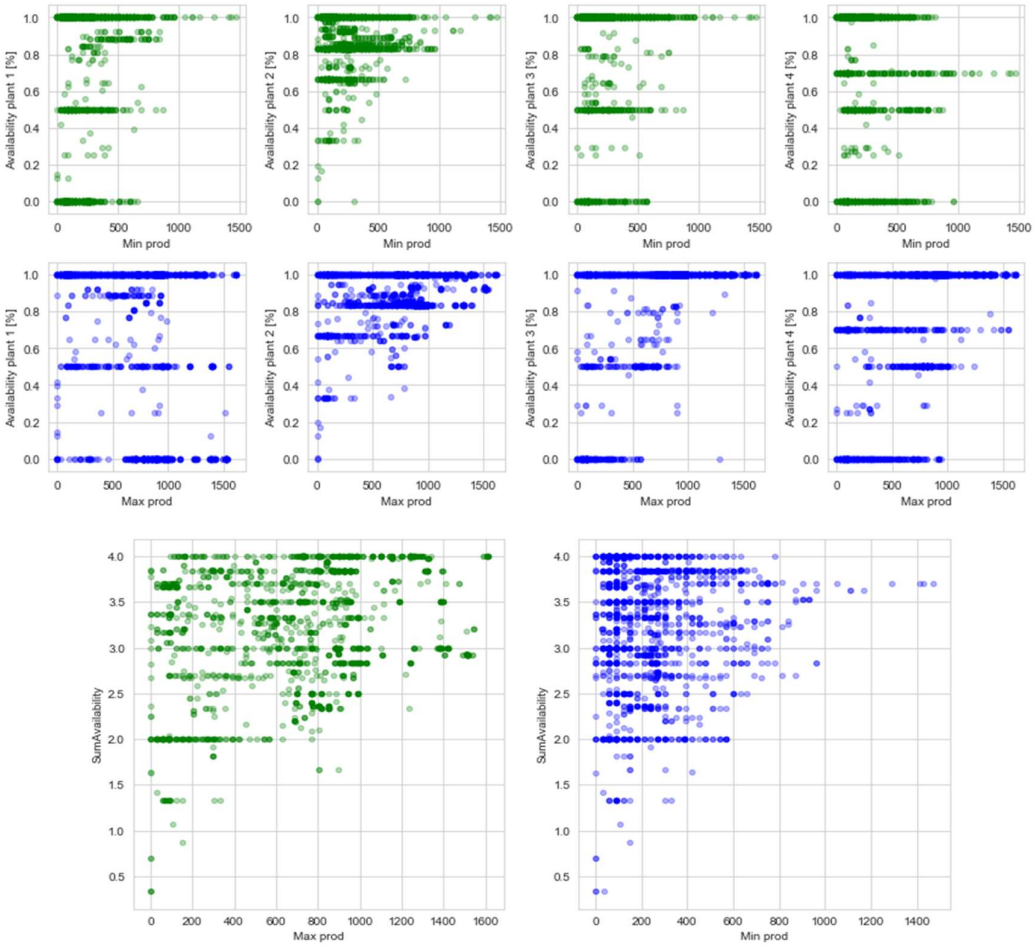


Figure 32. Min. and max. production vs. powerplant availability scatter plots. Individual (above) and global (below)

Here we confirm that values of availability are not continuous. The influence of global availability (i.e. sum of powerplants availability) can clearly be observed on maximum production, less on minimum production, but is still seems to be present.

4.2.5 Clustering on target data

Here we will use unsupervised learning in an unusual way. In machine learning projects, unsupervised techniques are usually used to discover structure in the input variables, disregarding the output variables. Which leads to better understanding of the data and more accurate targeting of adapted predictive models, or even to developing several distinct models (Rushdi & Perera, 2018).

Here, we will use such techniques to get clarity on our output variable structure, and more specifically on the power blocks. In case clear groups of values appear, this would simplify the prediction problem greatly, at

the cost of an acceptable error, i.e. clustering error, since we would need to target a finite number of profiles, thus transforming a multivariate regression into a univariate classification problem

In other words, our prediction would not target the full spectrum of possible values but values that are usually used, provided that such a set of "typical power profiles" exists. Performing clustering aims at checking this assumption.

First, we check how many different combinations actually exist, out of 1917 datapoints: there are about one third (688 precisely), which is still a lot.

Second, we plot our data points, i.e. power blocks, along the PCA dimensions. Important to note is that we are dealing here with a subset of the original dataset, containing only power blocks variables, hence PCA are obtained on this sub dataset, not the full one. NB: we are using the value pair representation of power block in this analysis.

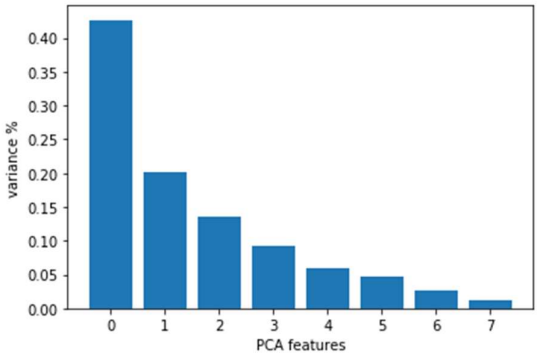


Figure 33. PCA loadings for power blocks analysis.

We note that the first two PCA dimensions account for more than 60% of the data variability, which is considerable and confers some value of the graph built on those:

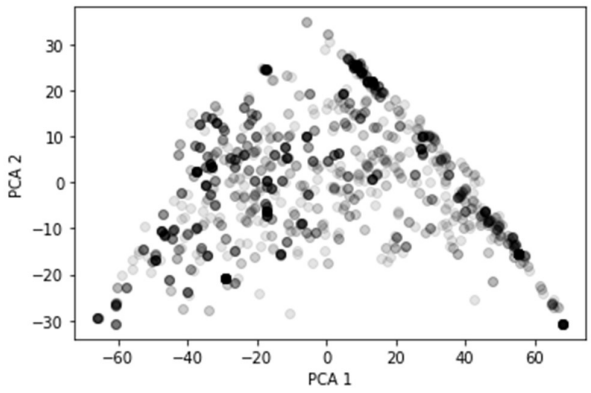


Figure 34. Power block data point along PCA dimensions

This projection gives a good idea of the closeness of that data points. To identify our clusters, we will use the "k-medoids" algorithm, that works similarly to the "k-means" algorithm, only using existing data points as centroids. The reason for doing so is the need to use coherent data points as target for our future model, not "average" values that are in practice not present in the dataset and that would no longer respect the given constraints. To run the algorithm, we first scale our data use in a "min-max" scaling, because we want the distances to be comparable without giving more importance to any of the dimensions in particular.

The only hyperparameter to set in this case is the number of clusters, outside of the distance measure (standard Euclidian distance since features are numerical). To pick a value, we look at the total distance between data points in the clusters and the cluster centre. Of course, this quantity diminishes with the number of clusters. The "elbow method" can help us here: the inflexion point will designate a good candidate for the k-value, here it seems to be around 7.

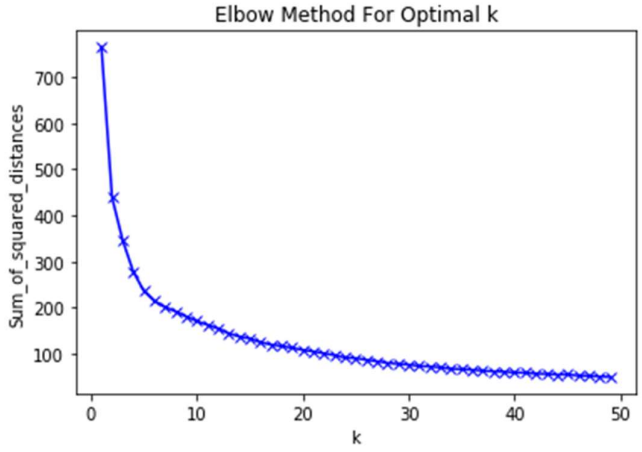


Figure 35. Elbow method for optimal k

The resulting cluster allocation can be visualized using a T-distributed Stochastic Neighbour Embedding (t-

SNE) plot, which allows to represent high-dimensional data in 2 dimensions, in our case, in such a way that similar objects are grouped together and dissimilar are distant (unlike the PCA plot above). The colours represent the k-medoids cluster allocation, with k=7.

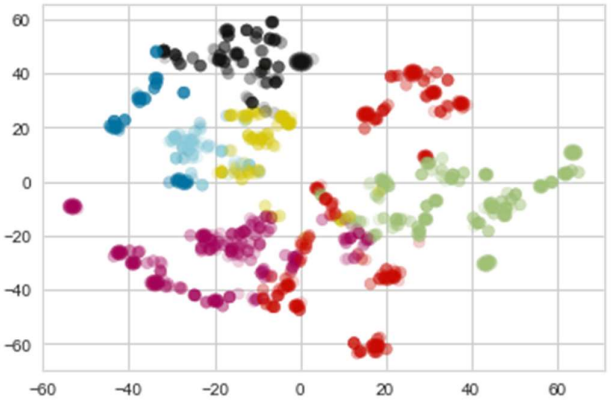


Figure 36. Clustering result with k=7 (t-SNE plot)

This shows a clustering does indeed produce interesting results, same for the silhouette graph below, since few points have negative silhouette score. However, at this stage, it also hints that a higher number of clusters might be possible which would diminish the clustering error. We'll see later how to approach this.

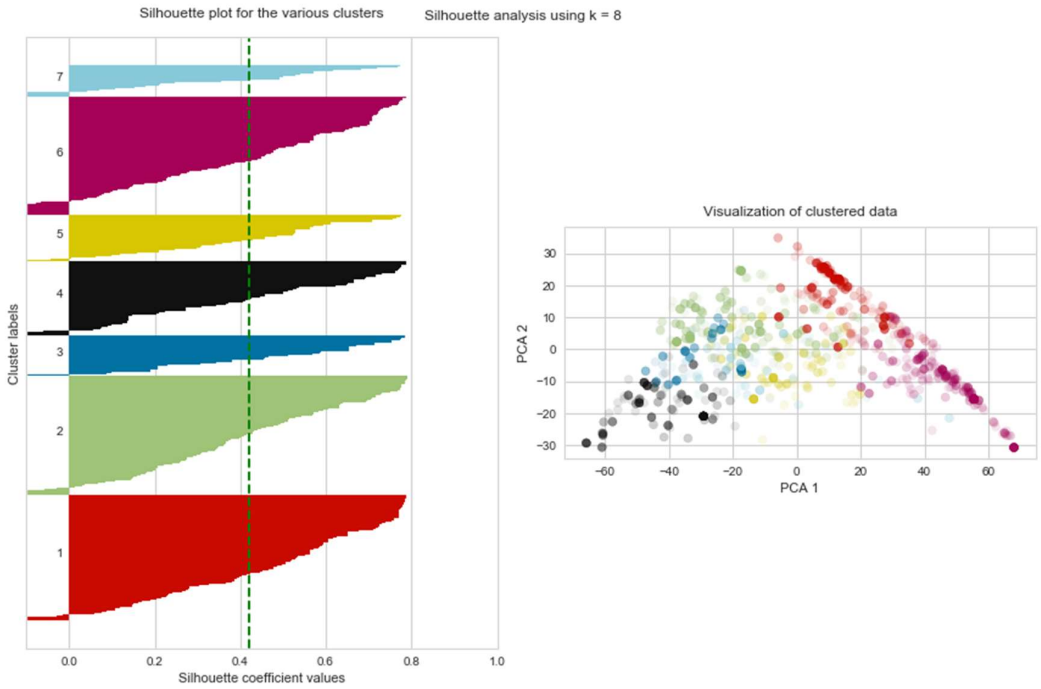


Figure 37. Silhouette plot of power block clusters (k=7)

4.2.6 Insight gathered thanks to EDA

Although the EDA step helped us get familiar with our dataset, it is not obvious if globally, we can find a predictive model demonstrating good performance by exploiting them.

We have spotted some influences that seem promising, either direct or lagged, that are worth taking into account in the next steps, i.e. feature selection, feature engineering and model experimentations.

We have also seen it can be worth reducing the target value set into clusters. Hence, we will check what results can be achieved by targeting these clusters in a classification problem, to ultimately come up with a power blocks prediction.

4.3 FEATURE ENGINEERING, FEATURE SELECTION

In this short section, we will present how some intuitions developed in the EDA step can be guide us to compute additional feature and select the ones to construct predictive models.

4.3.1 Calculated features

Based on the stakeholders' input, we introduce a "weekend" feature, calculated on the datapoint associated date, whose importance could not be confirmed by EDA. To make sure it is bringing useful information, we will check how models behave, with and without it.

4.3.2 Lagged features

Based on the EDA, two lagged features are worth experimenting:

1) Minimum production.

Since the ACF showed a significant autocorrelation at lag "one year", we will add a variable containing this lagged value. This is straight forward, except for the first year of data: in this case, we use the average value of all years for the datapoint, since this is our best guess.

2) Maximum lake level

Since we spotted a definitive influence of maximum lake level, we deducted the signal of this constraint coming must be integrated before it actually becomes effective, so there is time to take it into consideration and empty the lake. How long in advance is not clear. We propose to experiment with several values, from 0 to 60 days, using a step of 5 (lag in the future). Models' behaviour will be measured against all of those and best value selected.

4.3.3 Feature selection, dimensionality reduction

Dimensionality reduction was already performed in the data cleaning step, by limiting the number of power blocks to 4. Further reduction on output will tested thanks to clustering.

No PCA was performed on the input features because they are not numerous, and each seem to play a role in influencing some of the output features, so we saw this analysis as not necessary. Consequently, we do not propose to discard any of the input features.

4.4 MODELING AND EVALUATION

At this stage, we have some intuitions there are relationships between our input features, including engineered ones, and the target ones that could be reproduced by a predictive model. Thanks to our literature review, we a have a good idea of the model types to implement and how to compare them. This process is depicted in the next section.

4.4.1 Baseline prediction

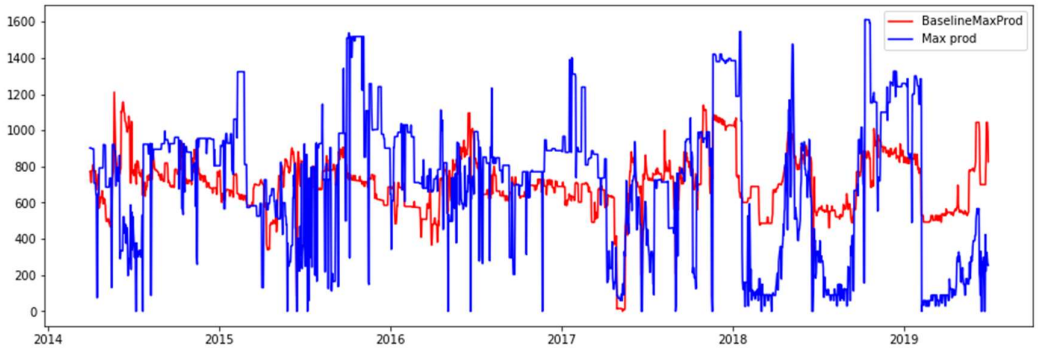


Figure 38. Baseline timeseries plot.

Over the entire time interval, we compute the following error measure, for the baseline prediction:

Measure	Minimum production	Maximum production	Maximum power (hourly)
RMSE	237.33	447.80	20.31
MAE	181.59	430.38	14.21
R^2	-192.26	6.06	42.34

Table 4. Baseline error measures

RMSE and MAE should be compared to the magnitude of the target values, as seen in the plots.

These values will be the targets we will try to surpass. Although it doesn't look extremely ambitious at this stage, we have no certainty to achieve this.

4.4.2 Data splitting / testing strategy

As explained in section 3.7, we will split the data keeping its time structure, hence use a consecutive time interval for training and the rest for testing. The ratio is set to 90% training and 10% testing.

4.4.3 Subproblem 1. Model definition and evaluation.

In subproblem 1, we tackle minimum and maximum energy prediction.

4.4.3.1 Regression with Linear regression, SVR, RF, MLP

For this double regression problem, out of the literature review, we consider the following model candidates: SVR, MLP, RF and LSTM and a simple perceptron model for ANN. To this we add linear regression and decision tree, as they come at very low cost in terms of implementation effort and are simply interpretable.

We keep in mind that hyperparameter tuning plays a determining role in ensuring best results, to test all models (but deep learning ones for now). We adopt a grid search method, for the simplest models, using parameters found in the literature review and compare achieved results. In this first step we compared at total of 98 models-parameters combinations, with following results, order by target and RMSE value:

No	Algorithm	Target	RMSE	MAE	R^2

8	**SVR**	**Min prod**	**75.9431**	**61.4425**	**70.0738**
0	DecisionTreeRegressor	Min prod	105.711	80.0949	42.0148
2	LinearRegression	Min prod	114.696	87.2005	31.7395
6	RandomForestRegressor	Min prod	127.36	92.2321	15.8338
4	MLPRegressor	Max prod	220.617	171.203	-152.554
3	**LinearRegression**	**Max prod**	**246.364**	**211.871**	**71.565**
1	DecisionTreeRegressor	Max prod	278.049	221.357	63.7807
5	MLPRegressor	Max prod	357.076	283.407	40.2664
9	SVR	Max prod	369.214	346.986	36.1365
7	RandomForestRegressor	Max prod	554.363	534.052	-43.9742

Table 5. Models benchmark (all but deep learning models)

The ranking shows that

1) For minimum production, SVR provides the best results, by a large margin. This indicates that our dataset can be separated rather accurately with linear hyperplanes using the right kernel function, i.e. radial basis.
2) For maximum production, to our great surprise, a simple linear model performs best out of the list. This is interesting as this is a very easily interpretable model. This raises question about the perceptron model configured (default hyperparameters) as it should perform in an equivalent manner at least. We'll spend time on MLP tweaking, in the next step.

As both models beat the baseline by a comfortable margin, we'll keep these new values as the target to beat when designing MLP models.

At this stage, we test the previously defined additional features: weekend, lagged minimum production and lagged lake maximum level. Using the same list of models and best hyperparameters, we find out the performance

1) improves when adding "weekend", for both minimum and maximum production, we decide to include to the input features list
2) degrades when adding the lagged minimum production, we therefore decide not to keep it as an input.
3) improves slightly when adding the maximum lake constraint lagged by 55 days in the futures, concerning maximum production. We therefore add it.

We provide here several plots to visually assess results of both models:

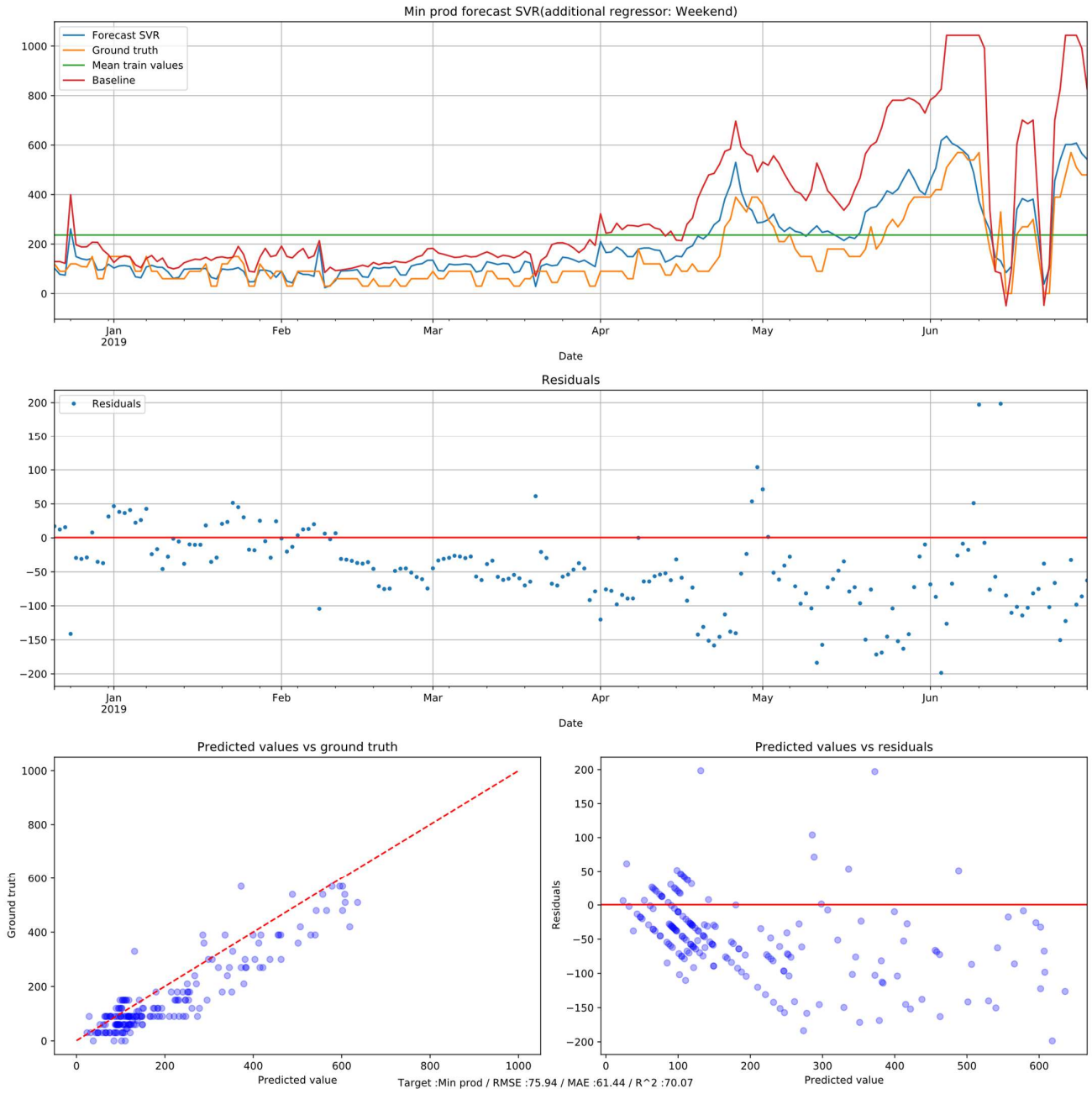


Figure 39. Minimum production regression with SVR plots

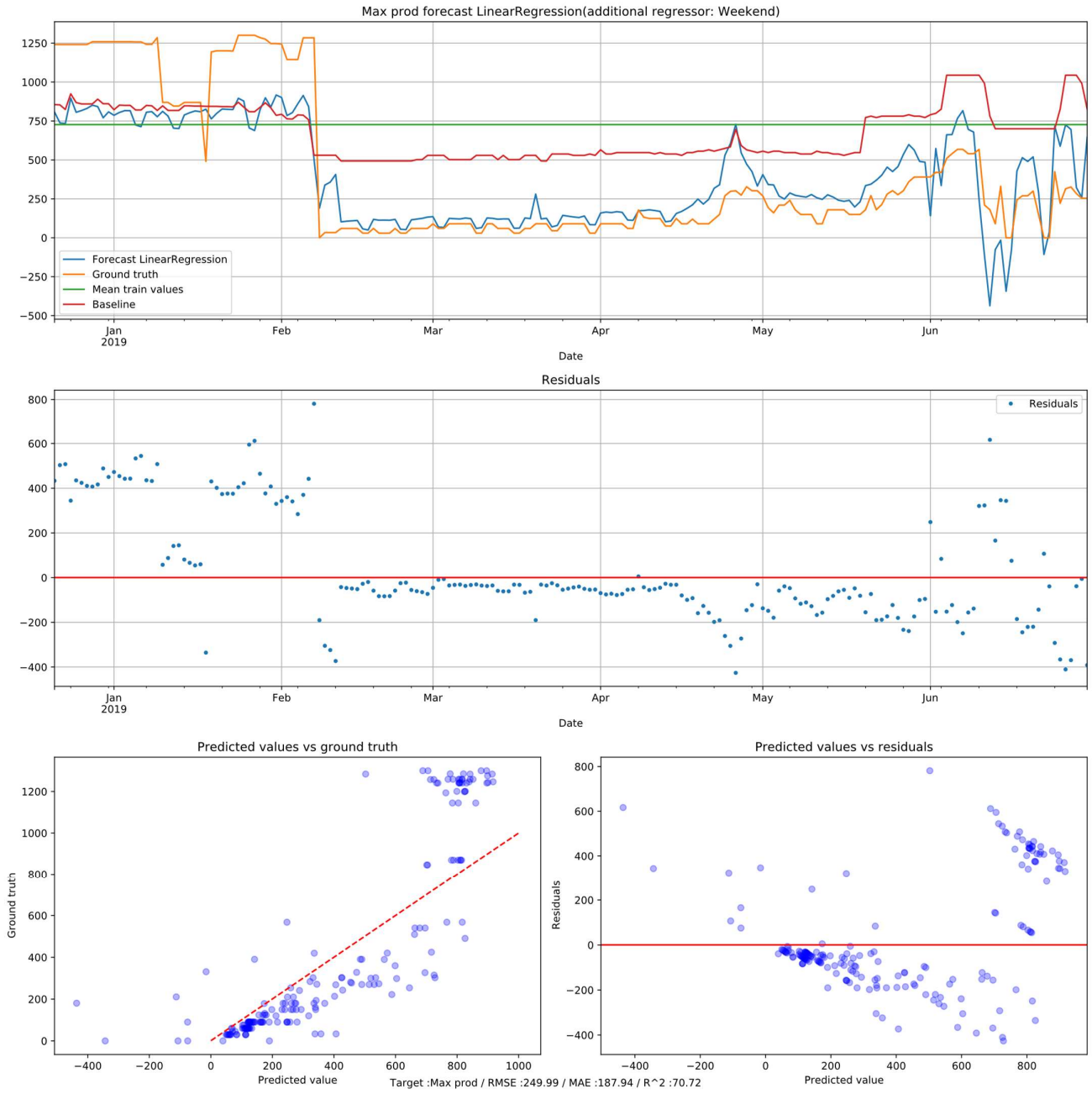


Figure 40. Maximum production regression with linear regression plots

From those plots, we see that both models produce decent predictions. Linear regression produce aberrant negative values and struggles in the beginning of the period, when values are high but it is noticeably close to the baseline.

General regression checks can be made here:

- the residuals mean should be equal to 0 and no trend visible.
- the residuals vs. fits plot should have no shape, i.e. there shouldn't be any pattern (information) left in the residuals.

As it doesn't seem to be the case, we conclude there is room for modelling improvement.

4.4.3.2 Regression with MLP

At this stage, we will try to beat the new baselines using the augmented list of inputs and MLP models, working on key hyperparameters: number of layers, number of neurones, activation functions and drop out regularization factors (Josh & Adam, 2017).

We reach the best results with the following hyper parameters combinations:

Parameters	Target	RMSE	MAE	R^2
Hidden layers size: 10,7,1 Activation fct: Relu, Relu, Lin. Drop out : 0, 0	Max production	192.09	129.86	82.72
Hidden layers size: 7,5,1 Activation fct: Relu, Lin, Lin. Drop out : 0.5, 0.5	Min production	56.64	40.37	83.50

Table 6. MLP model, 10 inputs, hidden layers (10, 7, 1)

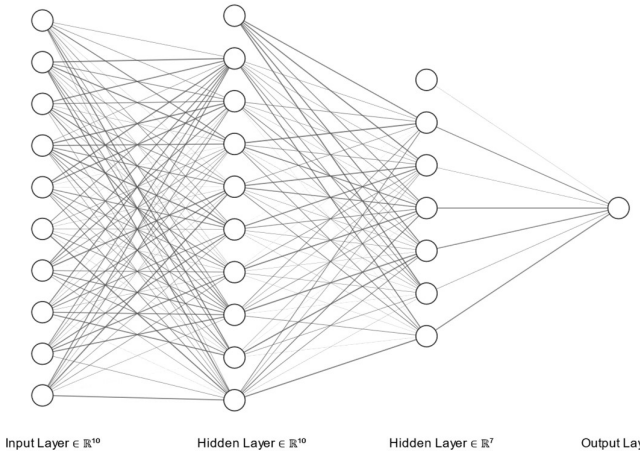


Figure 41. Example topology: MLP 10,7,1

We observe that the results reached show important improvement over the previous models, for both minimum and maximum production. Interesting is the fact that the best hyper parameter configuration is quite different for both targets, even number of neurons in layers. To us, this reflects probably the fact that the "logic" behind these targets is different, i.e. the rules used don't have the same complexity even though they share the same inputs here. May some other influential factors missing from our inputs might also play

a different role and these factors could be different in both cases.

Here are the plots from both models' results, illustration the model's good performance.

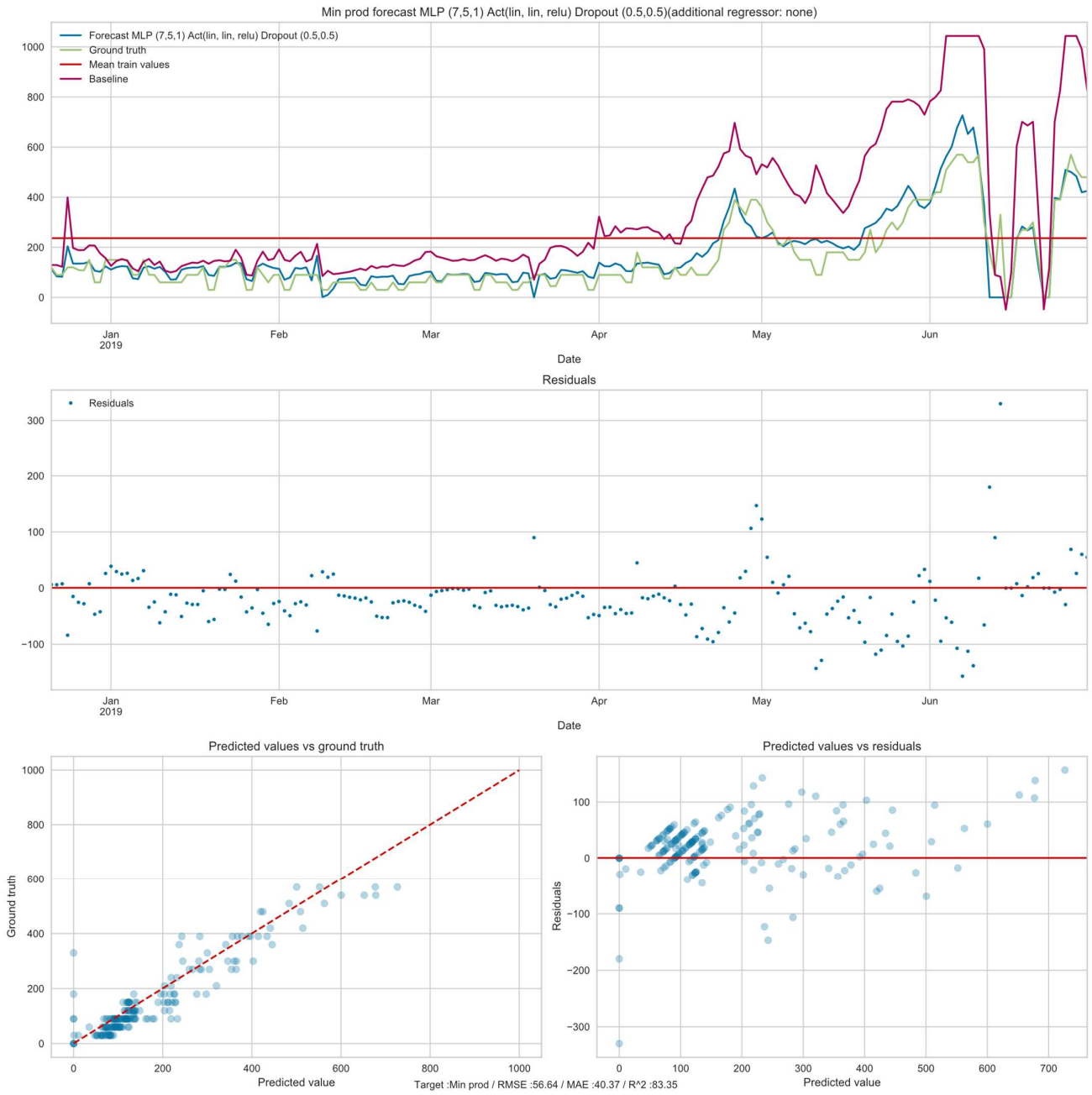


Figure 42. Minimum production - MLP (7,5,1) regression plots.

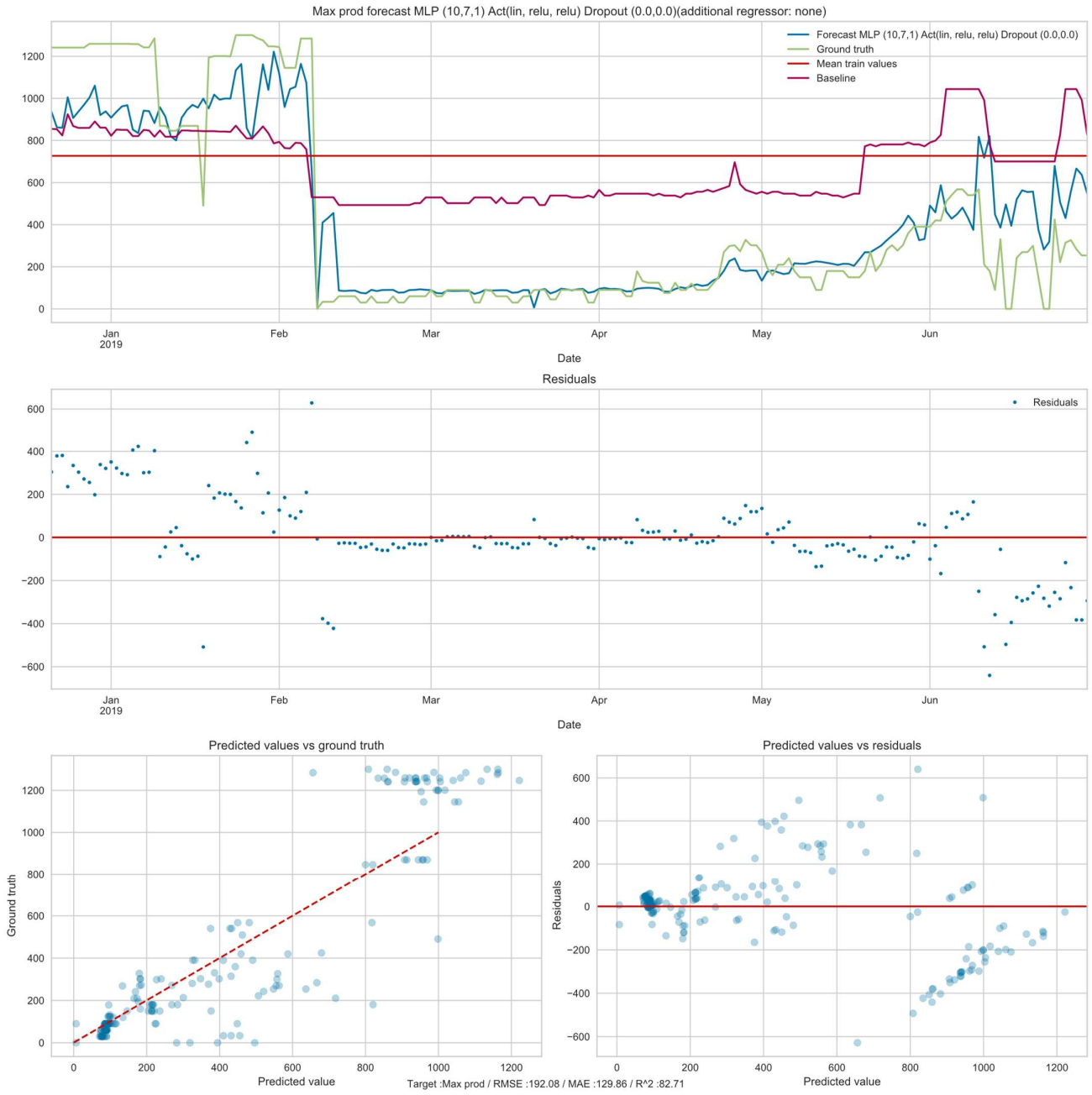


Figure 43. Maximum production - MLP (10,7,1) regression plots

4.4.3.3 Regression with RNN – LSTM / GRU

Following findings of our literature review, we implement RNN models searching for better accuracy.

Here the main hyper parameter to tweak is the learning window, i.e. size of history to include in the training process, of which the RNN network will learn and keep in "memory" the history of past datapoints, building a "context". The other hyper-parameter is the type of memory unit: GRU or LSTM.

Despite our efforts and trying several configurations, we could not do any better than the MLP models.

For example, concerning maximum production, using a rather close to the MLP topology, we are getting the following results:

Model	RMSE	MAE	R^2
GRU (15, 7, 1) **Drop out: 0.1, 0.2**	231.57	147.73	74.88

Table 7. RNN error for maximum production regression

Best results are attained with GRU units, and two regularization drop out layers and a dense output layer using a linear activation function. This is worse than the best MLP model, we therefore eliminate this model from the final solution.

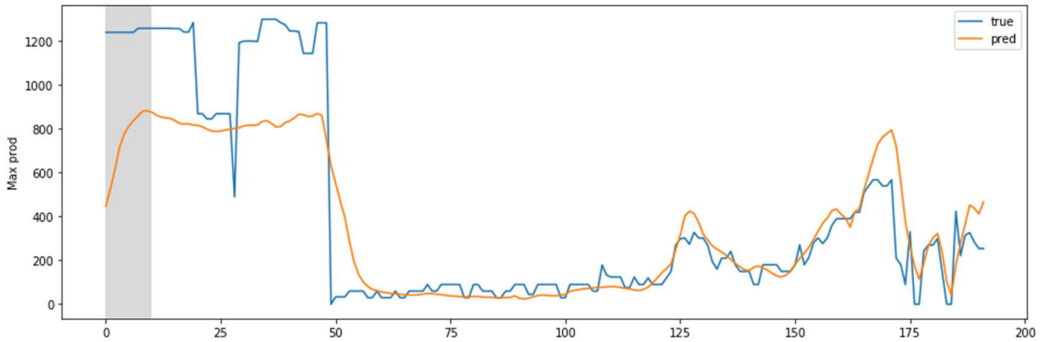


Figure 44. RNN maximum energy prediction timeseries plot

Interesting to note is the shape of the curve: it is less sharp-edged than the MLP one, probably showing the influence of the "historic" context in the prediction, provoking some inertia in the model's behaviour along the time dimension. Although effective in many forecasting problems, it doesn't bring improvements in this case.

4.4.3.4 Results interpretability

Our best models, i.e. MLP, suffer from the "black box" syndrome. On the other hand, the simpler models deliver a good insight on how the input variables are used.

Concerning maximum production, the simplest and most interpretable model is delivering the best of the

predictions (outside of MLP). It is worth looking at it into more details:

RMSE	MAE	R^2
246.364	**211.871**	**71.565**

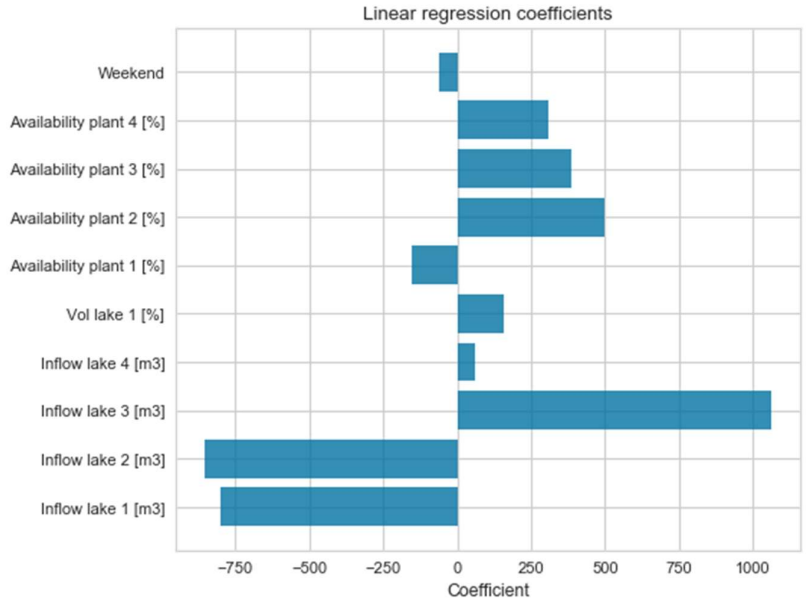


Figure 45. Linear regression error and coefficients.

Inputs have been scaled using a MinMax scaler, so the coefficients can be compared. The interpretation is straightforward: most important factor is the inflows in a small lake (3), then maximum production is directly related to power plants availability (2, 3, 4). The negative value of coefficient for plant 1 availability could be related to its cascading topology: when it is available, it limits the flow, when it is not, the flow is diverted, hence more energy can be produced. Volume of lake 1 plays a less important role, probably because water volume increases the pressure in the penstock, hence power. Weekends influence the value negatively; it makes sense as demand is generally lower on weekends too. The other inflows' role could be that they force to produce on less powerful powerplants.

Concerning minimum production, we refer to the best "simple" model, i.e. SVR with radial basis function kernel, which is a black box. We therefore use LIME which is a method to measure input variables influence locally by linearizing the model's behaviour (Ribeiro, Singh, & Guestrin, 2016).

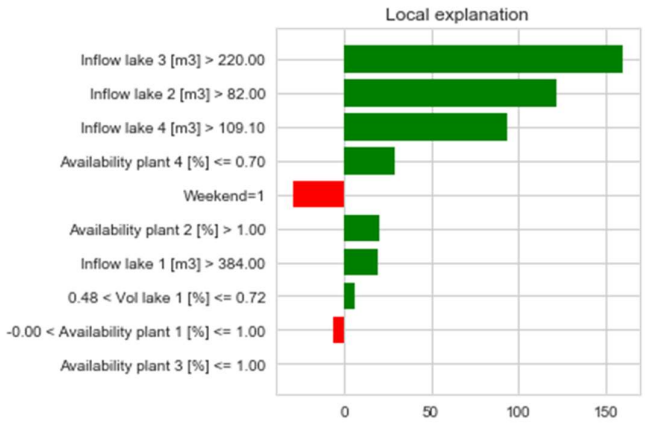


Figure 46. Variables importance in the SVR model

Looking at a sample towards the end of the test period, we see that minimum production is mainly influenced by inflows, especially in small lakes, which is in line with our expectations: small lakes filling up trigger obligatory production to avoid spilling. Again, weekend has got a negative impact, same for powerplant availability. This is all very sensible. Global behaviour should be assed looking at other samples.

4.4.3.5 Regression wrap up

This concludes our subproblem 1 task, as we have attained the best possible predictions using the proposed methodology, doing considerably better than the baseline. About what algorithm to keep in the final solution, we are facing the accuracy vs. interpretability trade-off since deep learning models offer the better accuracy while simpler one offer better interpretability. We showed tools exist to explain local results, which would fit deep learning models too.

But the power blocks are not yet predicted, and the final accuracy numbers must take those into account, in a global coherent prediction. The next step consists precisely in predicting those power blocks, using these energy predictions as inputs.

4.4.4 Subproblem 2: power blocks prediction

The first approach here is to consider this problem a multi-objective regression.

4.4.4.1 Subproblem 2: power blocks prediction using regression

Out of the literature review, we build the following list of model candidates to perform this multi-objective

regression: RF and MLP.

As a first experiment, in order to underline the usefulness of the maximum power values as input, we run the RF algorithm with the 4 power blocks value as targets, using the rest of the variables as inputs only.

As all results are decimal numbers, we round the number of hours using a simple rounding function. We compute the maximum power using the resulting power blocks. A quick visual check is enough to get convinced this is not a good approach, as shown below, the values are totally off.

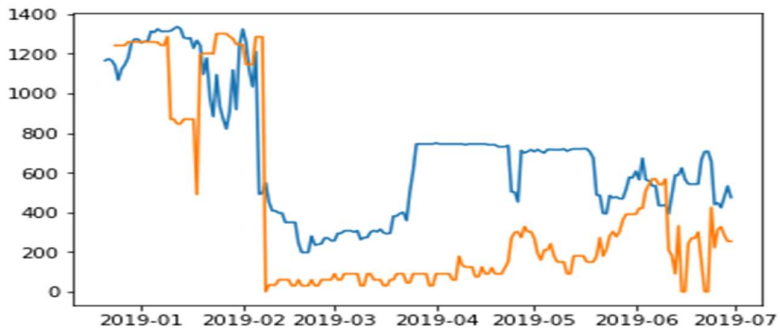


Figure 47. Prediction plot: multi objective regression RF without maximum energy (blue = prediction, orange = ground truth)

We now integrate the maximum energy as additional input. We start by normalizing the outputs. This goes along with the idea that typical power profiles are ultimately behind power blocks, that are scaled by the global maximum production. This approach is confirmed by our experimentation.

We train the RF algorithm using the previously obtained best prediction and other regressors. We then scale back the results and round them using a “ceiling” function to obtain the results plotted below:

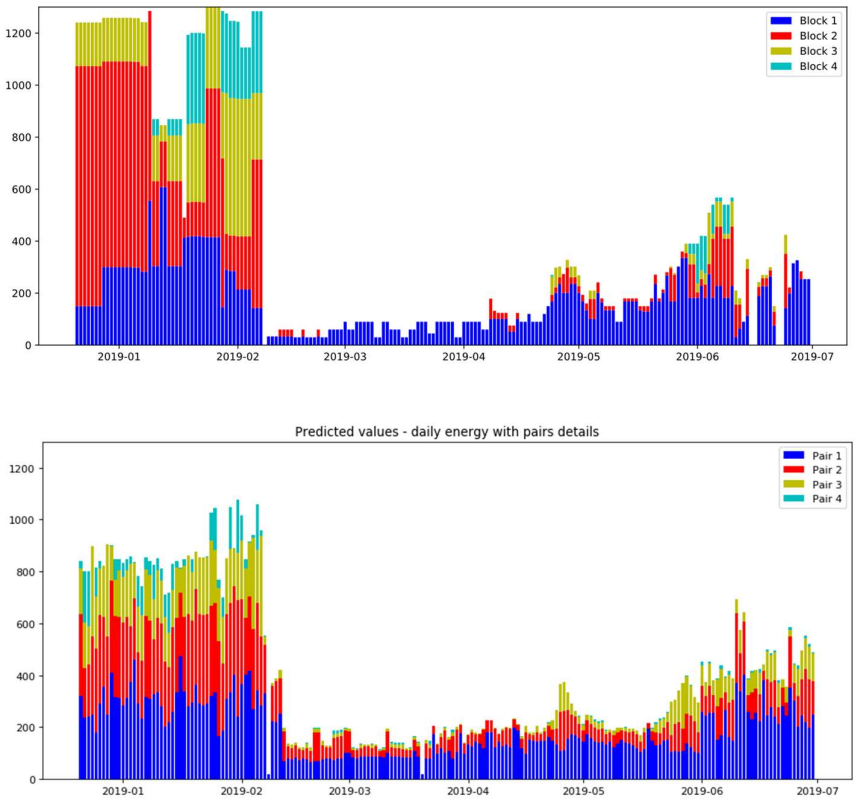


Figure 48. Power blocks timeseries plot: ground truth (above) and Rf multi-objective prediction (below)

The hourly error measure gives the interesting results over the test period (2 first columns):

Measure	Predicted power	Maximum power	Baseline
RMSE	**14.06**	87.28	24.18
MAE	**8.78**	78.03	19.05
R^2	**64.56**	92.08	-4.39

Table 8. Error on power blocks, RF multi objective model

We see that this algorithm beats the baseline by a large margin (which does particularly poorly on this time period).

The second column shows the error measure concerning the reconstructed maximum power, i.e. reconstructed vs prediction. We see that a significant error results from this regression: it has not been able to reproduce the input values accurately.

We now try to do better job using an MLP multi-objective regression model, using the same process.

Unfortunately, despite our efforts in identifying a fitting topology, the obtained MLP model, although it can learn from the input data, does not generalize well at all on the test data, meaning it overfits and produces very inaccurate predictions, as the learning plot shows:

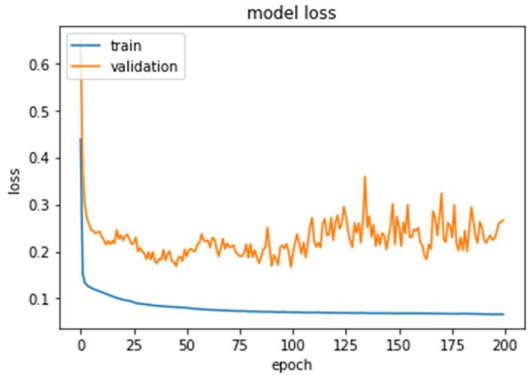


Figure 49. Loss curve during MLP training

Not only does the validation loss value remain a lot higher than the training loss during the training process, it does not converge regularly as it should: the model is not stable with that regards, meaning that the weights adjustments happening in the learning process produce chaotic effects on the training loss, that do not contribute consistently to improving the prediction quality. In one word, the model fails at generalizing what it learns from the data. Our interpretation is that the power blocks values are too erratic: there is no relationship the model can learn from the train dataset that can successfully be applied to the test subset.

We are not saying that no MLP model could do better but based on the chosen methodology and set of hyperparameters, although disappointing, this is the only result we have been able to achieve.

Based on these 2 experiments, we can infer conclusions about the “all in one” approach without implementing it: we can’t enforce constraints in the model, the multiplicative relation in particular and MLPs fail at predictive the blocks. We could apply the multiplicative constraint in postprocessing, by calculating the maximum energy on the blocks’ values, but errors on rounded number of hours and power would get multiplied too. Bottomline: it would at best deliver inconsistent results or very low accuracy results.

Our final attempt at improving the power blocks prediction is using a strong assumption: simplifying the target dataset to a limited number of typical profiles, thus turning our 4x2 targets regression problem into a

single target classification one.

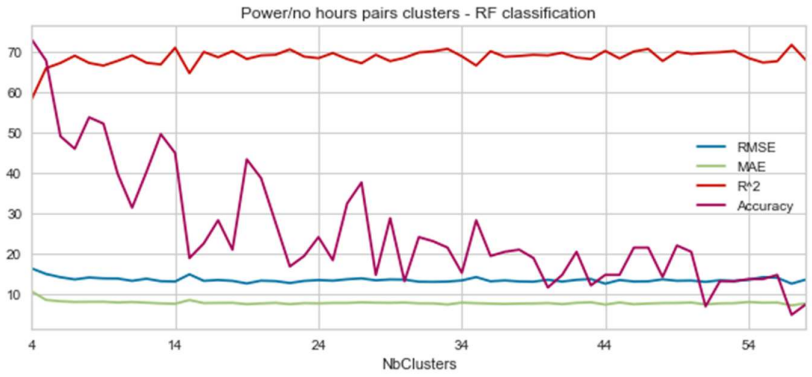
4.4.4.2 Subproblem 2: power blocks prediction using clustering and classification

Clustering data is always possible: the question is "do the clusters make sense", i.e. is the created partition bringing together elements that really belong together. The first thing to do to answer this question is to define the distance measure, i.e. what closeness means in our context. When blocks are considered as pairs of values, we simply use the usual Euclidian measure, as in the EDA. When considering them as 24 hours vectors, we will use a Manhattan distance that doesn't suffer from the large dimensionality.

Then, the question is how many clusters our data is intrinsically made of. Graphically, we could see that 7 made sense, but also more could be considered. As our final goal is to use those clusters to perform a classification, it seems like there could be an optimum value to find: the greater the number of clusters, the smaller the error due to clustering will be, but a number too large will certainly imply a greater classification error rate. We will experiment an range of values for this number to see which value is optimal, and then compare the result to the best results we got until now, i.e. the one delivered by the RF model.

We build a classification algorithm testing two common ones: RF and k-NN, as the first one generally offers a good accuracy/interpretability trade-off, and k-NN seems like a natural choice here: in case of misclassification, it is reassuring that the classification label comes from the closest data points, even if wrong. Both must deliver a verdict in terms of classification accuracy but also in terms of hourly power RSME calculated from the resulting power blocks classification.

We are implementing a grid search on k = number of clusters, between 4 and 60, using the 2 representations (value pairs and vectors) and plot the results below:



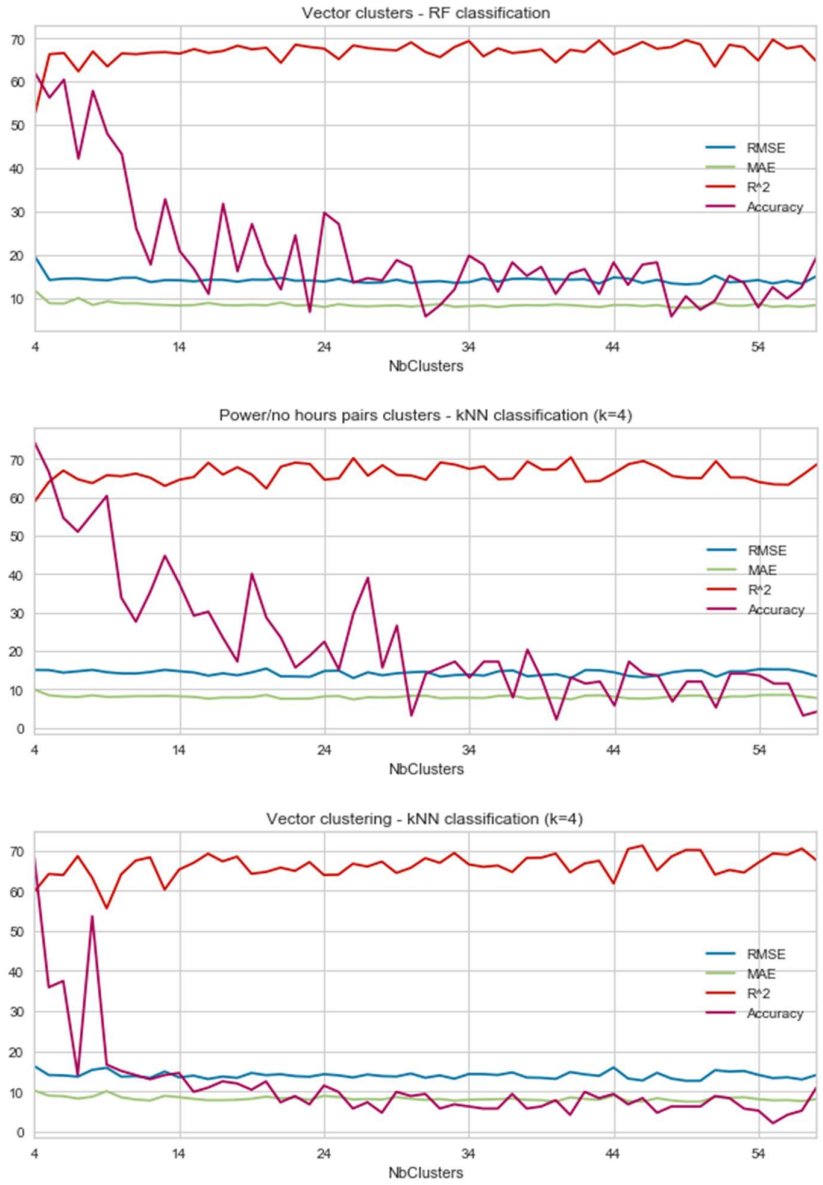


Figure 50. Grid search for best number of clusters on power blocks (k-NN, RF classifiers, value pairs and vectors representations)

We observe a common behaviour in the four combinations and a striking stability in the error values as *k* grows, especially with the RF classifier. We conclude the principle is valid. Although there is an optimal value of *k,* minimizing the error measures, it is not clearly standing out. To us, this is very surprising: as expected the classification accuracy is decreasing as k increases, but the final error, measured on hourly power values, remains stable. Our interpretation is that the classification accuracy error is compensated by the clustering error (not represented here), which is decreasing as *k* grows.

We see this approach produces an error value that is lower than the RF multi-objective one, while preserving

the maximum energy error value (daily), as expected.

NbClusters	Algorithm	NbClusters	RMSE	MAE	R^2	Accuracy	Daily MAE	Daily R2	Daily RMSE	Target Type
57	RF	57	12.41	7.03	71.61	4.6875	121.74	84.24	183.41	Power-Nohours

Table 9. Optimal value of k in k-medoids clustering (57, RF classifier)

At this stage, we have identified a solution that produces the lowest errors for all 3 targets, while beating the baseline and respecting all constraints without applying any post processing (rounding). This looks like a clear winner.

We check the results visually:

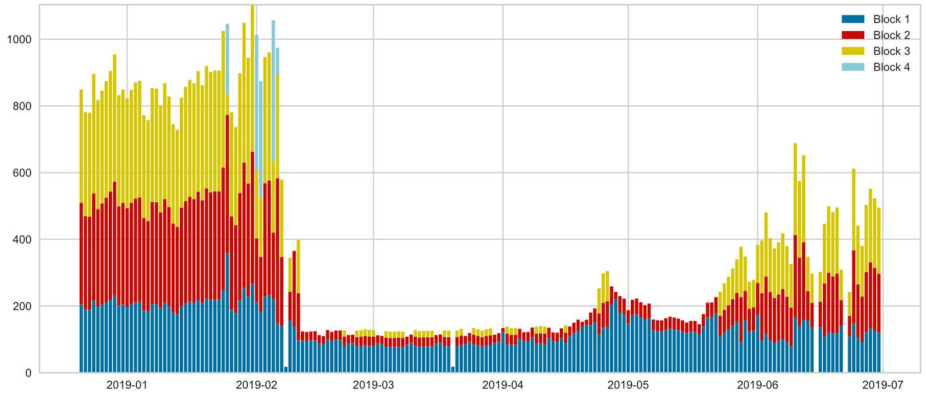


Figure 51. Predicted power blocks'energy with clustering (k=57, RF, vector)

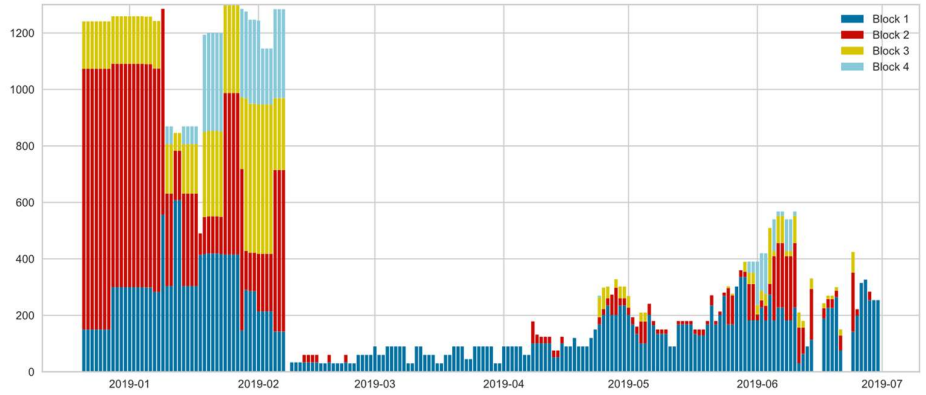


Figure 52. Power blocks'energy - ground truth

Difference in the size of the coloured bars illustrates how much the corresponding blocks energy values are different (power or number of hours or both) but large differences don't imply large RMSE on hourly power numbers, as shown in 3.9.1.

The final step is to bring some clarity in those results.

4.4.4.3 Results interpretability

First, we can measure the "intrinsic" clustering error, i.e. the error made if performing perfectly in the classification problem, i.e. 100% accuracy over the entire dataset:

RMSE	MAE	R2	Daily RMSE	Daily MAE	DailyR2
15.92	9.49	72.26	0.0	0.0	100.0

Figure 53. Intrinsic "pairs clustering" error on total dataset (8 clusters)

RMSE	MAE	R2	Daily RMSE	Daily MAE	DailyR2
8.84	4.88	89.55	0.0	0.0	100.0

Figure 54. Intrinsic "vector clustering" error on total dataset (8 clusters)

Although not visible on the test period, the vector clustering is performing best, introducing much less error. These numbers along with the silhouette plot (below) validate the classification approach: the results on the test sub-set are not due to "chance".

The silhouette plot to illustrates the clustering error visually, over the entire dataset. NB: The PCA 2D plot doesn't highlight clear clusters, since the PCA were constructed on the value pairs representation, not the vector one used here!

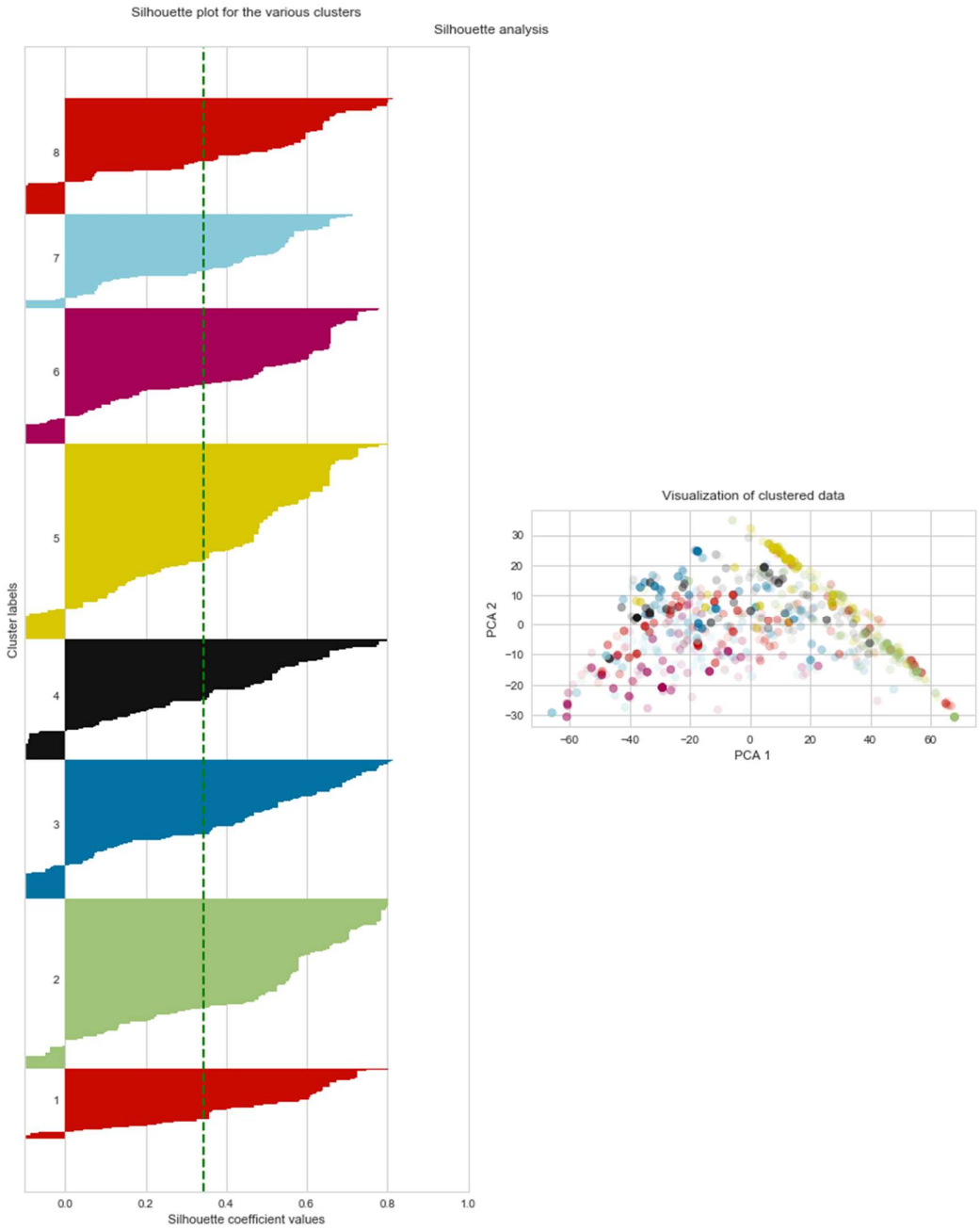
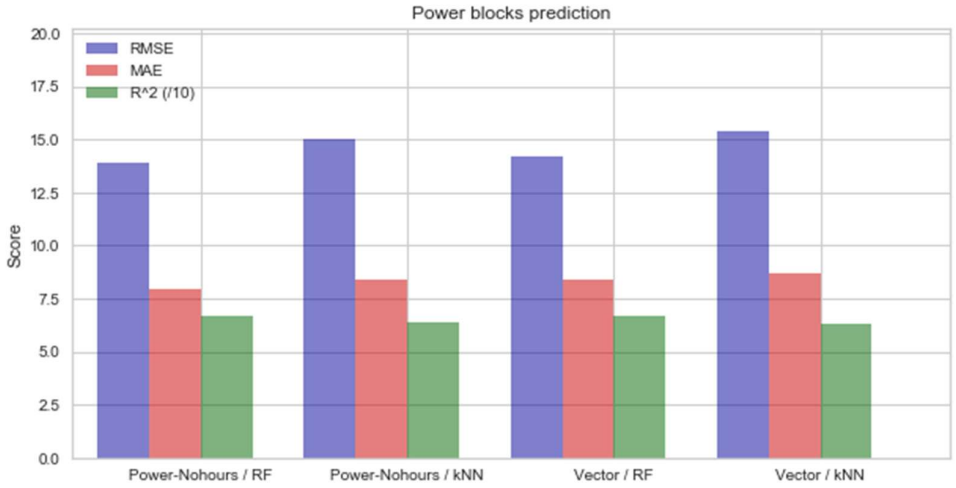


Figure 55. Clusters silhouette, k-medoids, k=8 on vector representation

As *k* value has little influence on the final error values, we choose a value that allows for interpretation, i.e. *k* relatively low still enabling decent accuracy with the RF classifier: we settle for 8, since accuracy is reaching 57.8% (vector representation).



NbClusters	Algorithm	NbClusters	RMSE	MAE	R^2	Accuracy	Daily MAE	Daily R2	Daily RMSE	Target Type
8	RF	8	13.94	7.92	67.12	53.645833	129.86	82.71	192.08	Power-Nohours
8	kNN	8	15.03	8.39	63.72	55.729167	129.86	82.71	192.08	Power-Nohours
8	RF	8	14.23	8.39	66.93	57.812500	129.86	82.71	192.08	Vector
8	kNN	8	15.36	8.67	63.21	53.645833	129.86	82.71	192.08	Vector

Figure 56. Error measure for clustering with k=8

We see all algorithms perform about as accurately, with both power blocks representations. We now look at the actual typical profiles delivered as cluster centres:

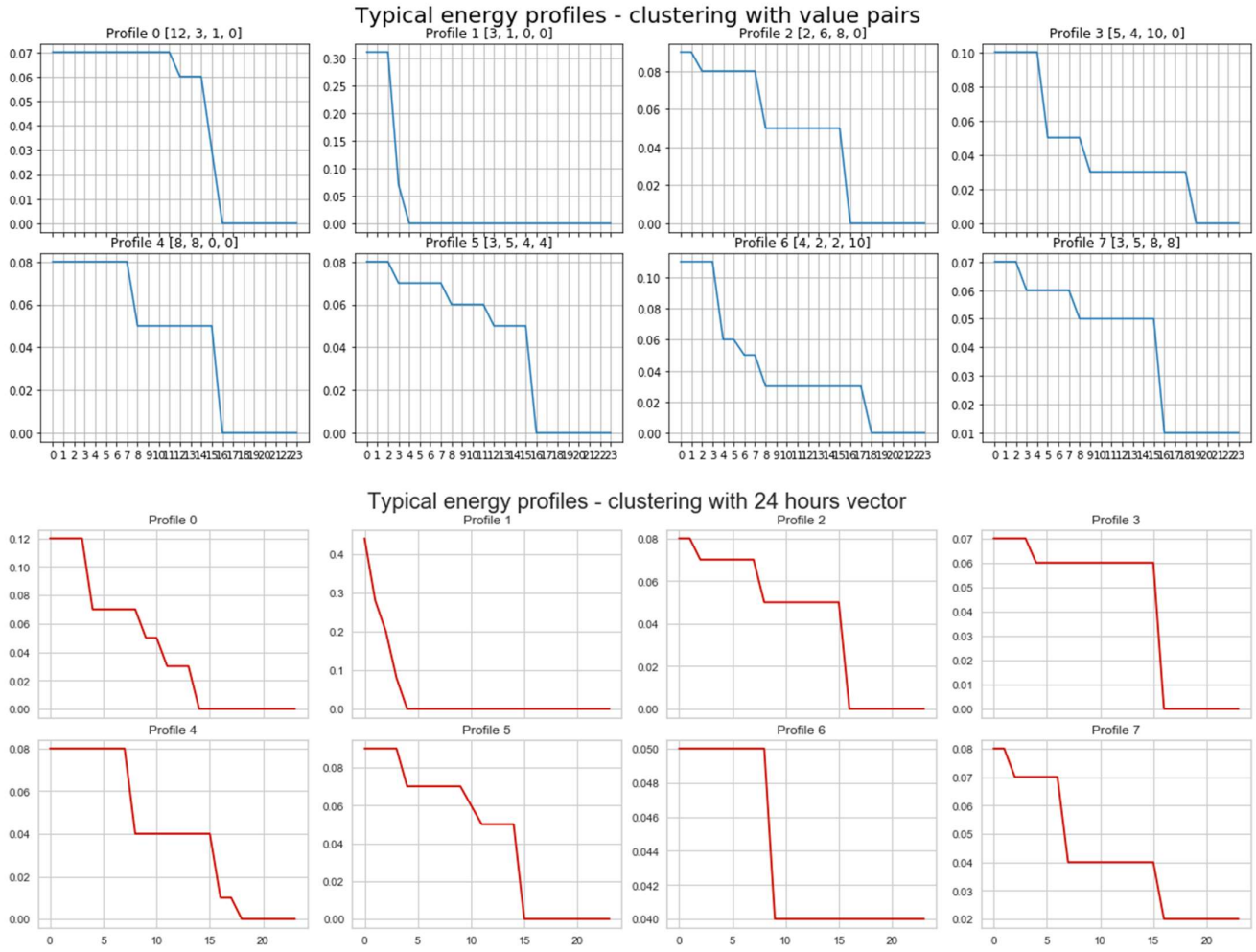


Figure 57. Typical profiles for values pairs (top) and vector representations (bottom).

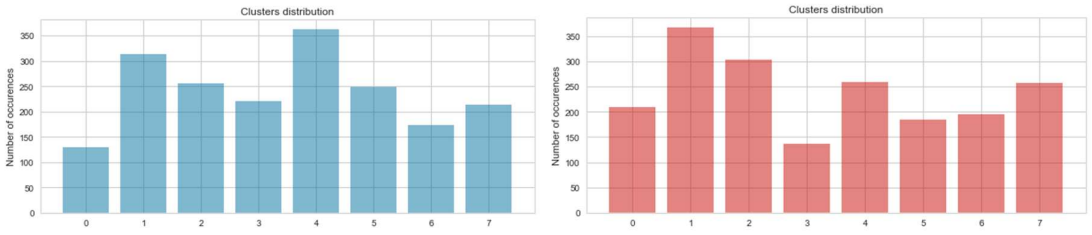


Figure 58. Typical profiles distribution over entire time period (value pairs left, vector pairs right)

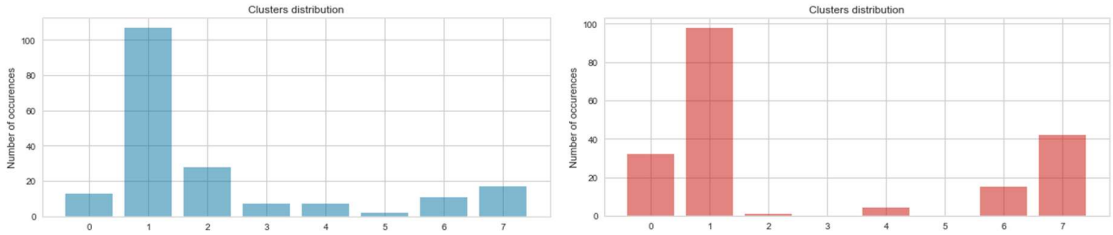


Figure 59. Typical profiles distribution over test period (value pairs left, vector pairs right)

We notice that the defined test period is not representative at all from this point of view. This is not an issue since the classification learning has been made on the rest of dataset.

In our opinion, these plots bring great clarity in the data by removing noise and presenting a clear and synthetic representation of the target values. Expected simplification is really materializing and bearing its fruits.

However clustering has got a price in the error in introduces. In the k-medoid algorithm, datapoints in one cluster can present an important distance from their centroid, as shows C1 centroid below. Although this maximum distance shrinks with the number of clusters, so does the classification error, as we have seen.

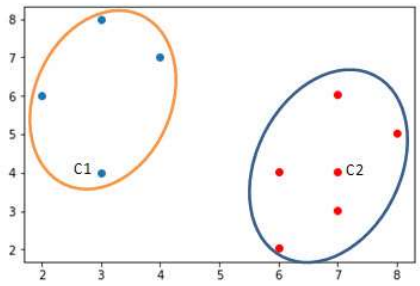


Figure 60. k-medoid example of high distance cluster (Kanakalatha, s.d.)

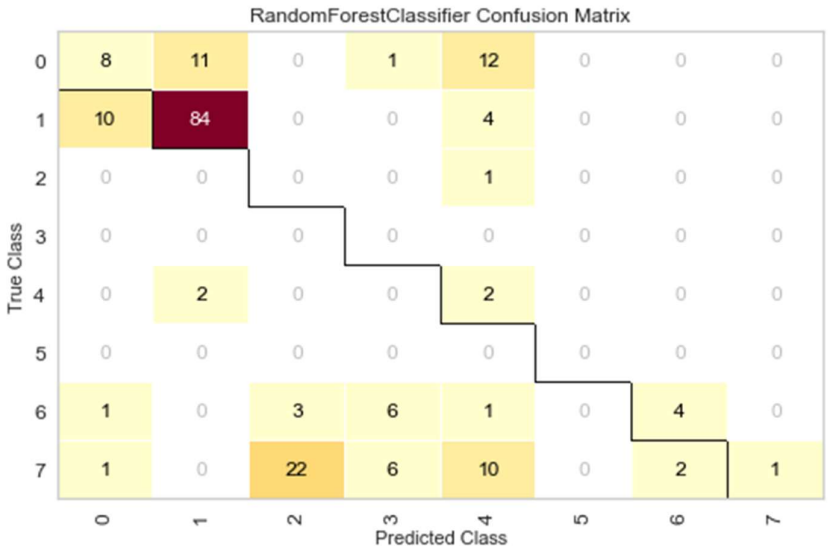


Figure 61. Confusion matrix for RF classification on vector representation

Looking at the classification confusion matrix, we see which profiles are correctly classified and which ones are not: in detail, we notice that the ones misclassified are close to the correct ones, visually. Which explains

why the overall numerical error, RMSE on power values, is rather low even when the classification error gets high:

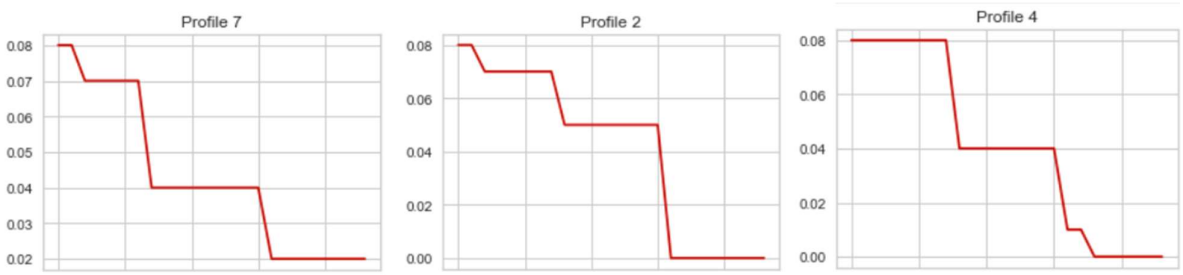


Figure 62. Profile 7 is misclassified as 2 and 4 mostly.

Finally, we extract the variable importance score from the RF model, in order to bring clarity on what influences the classification. We confirm the maximum energy plays the greatest role, then volume of lake 1, then inflows of all lakes.

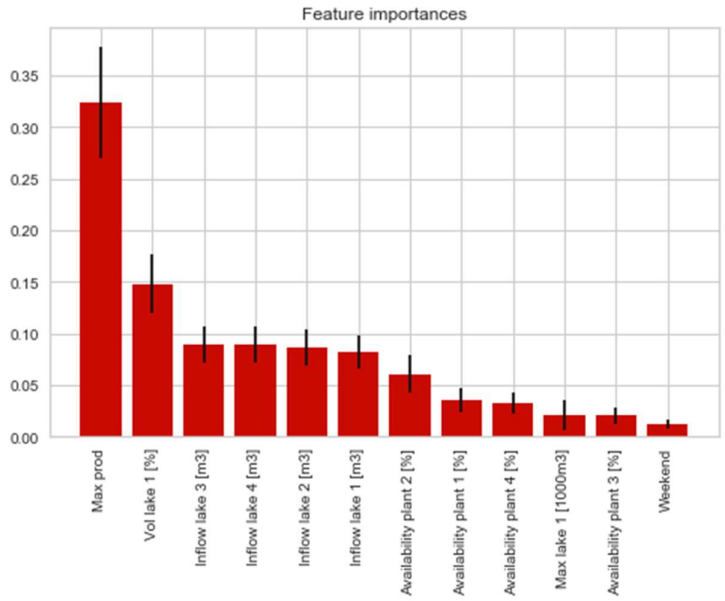


Figure 63. Classification variables importance ranking

To conclude, we believe that this clustering approach is enabling to gain very appreciable clarity on the original dataset, thanks to a classification variable importance ranking, typical profiles plotting and distribution. This is producing more accurate results than multi-objective regression, therefore we keep this approach for the final model. A better accuracy in the classification rate would still bring better results.

4.5 SUMMARY

In this large chapter we have described how we have implemented several solutions according to the chosen

methodology and approach, compared them and chosen the best one.

We note that the final solution

1) Respects all the given constraints: number of hours are entire numbers; their sum is lower or equal to 24 and maximum energy is equal to the sum of power blocks value.
2) It provides clear and detailed insights on the data hence appreciable interpretability
3) Although it accumulates 3 errors from the 3 cascaded models (maximum energy regression, clustering, classification), the result is still noticeably above the given baseline.

Constraint are respected at the cost of precision, as we have seen the clustering intrinsic error cannot be ignored.

In the next chapter, we will discuss the results we got using a critical point of view.

5 CHAPTER 5: RESULTS AND DISCUSSION

In this chapter we are discussing the results gathered in the course of our experimentation. Goal is to shed a critical light on the accomplished work, spotting weaknesses and proposing ways forward.

5.1 RESULTS RECAPITULATION

We have seen that predicting the 3 quantities in question was best solved by splitting the problem into subproblems in a cascading approach, depicted in the “greenlights path” below.

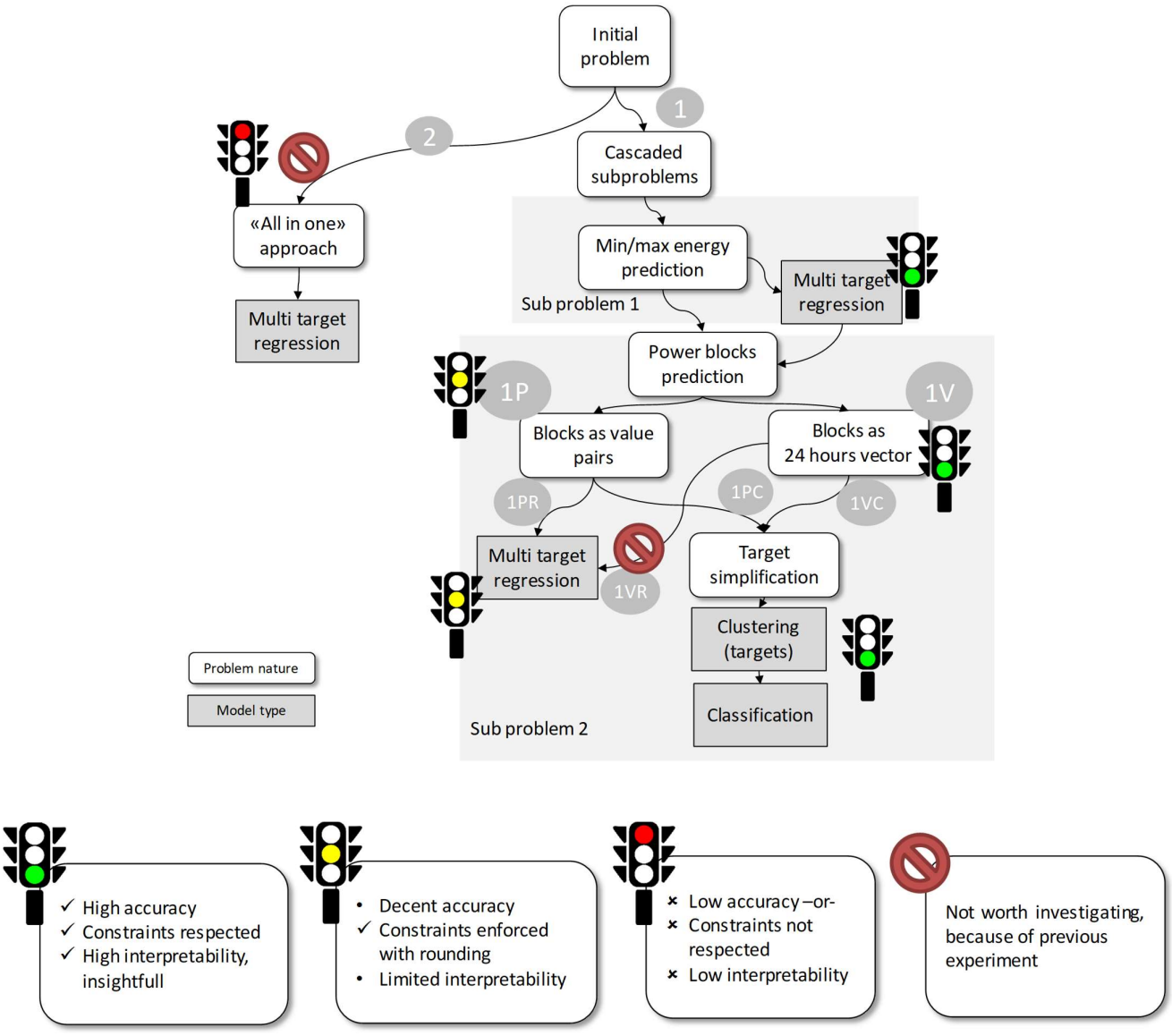
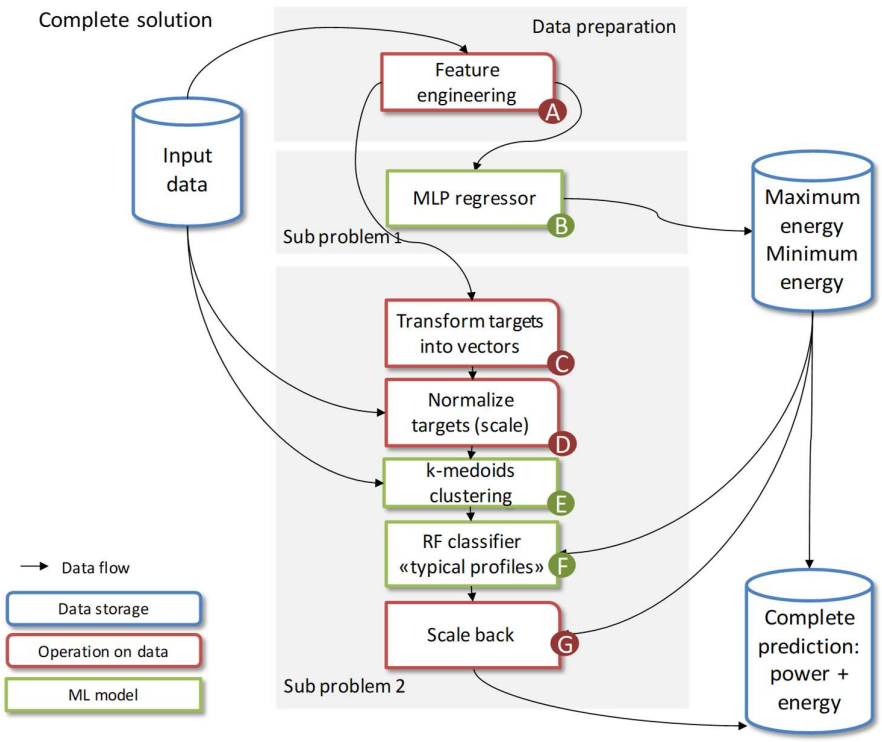


Figure 64. Problem decomposition and experiments results

First, predicting daily power values was achieved with best accuracy using MLP models with tuned hyperparameters (1). This is not very surprising to us, since analysing the dataset during the EDA step, we could identify several dependencies presenting non-linearities. Our experience shows that MLP are successful at capturing those relationships. With the proper drop out regularization settings, we manage capture them in a general enough way, avoiding overfitting. One weakness usually attributed to MLPs is their opacity. However, we saw that other algorithms presenting acceptable performance with the advantage of being much easier to interpret: linear regression for maximum production for example. As our project aimed at producing accurate predictions but also at bringing clarity on them, they might be preferable as final solution in the productive environment.

Then we have used these values in a multi-regression model used to predict the 4 pairs of values in the power blocks (1PR). A task at which MLP models struggled. Results attained with a RF model surpassed the proposed baseline. In terms of interpretability, the RF could to deliver a ranking of input features importance. Unfortunately, this regression introduced additional error on the maximum energy, despite being an input, it was degraded once reconstructed, partly because the number of hours delivered could not be restricted to a whole number. Bottom line: we could not force this model to respect constraints perfectly, accuracy was not optimal, despite bringing some interpretability.

Finally, in the context of power blocks, we have seen that simplifying the set of targets using clustering, did produce results respecting constraints, bringing overall best accuracy and very clear insights, in particular when using the power vector representation (1VC). For all these reasons, it is the solution we select as the productive one, as shown below, along with error at each stage and details of constraints implementation.



						Baseline		
Step	**Model**	**Result**	**RMSE**	**MAE**	**R^2**	***RMSE***	***MAE***	***R^2***
B	MLP	**Minimum prod.**	56.64	40.37	83.50	237.33	181.5	-192.26
B	MLP	**Maximum prod.**	192.09	129.86	82.72	447.8	430.3	6.06
E	*k-medoids*	*Clustering (k=8)*	*8.84*	*4.88*	*89.55*			
F	*RF*	*Classification*	*Accuracy: 57.81%*					
G	E+F+Scaling	**Maximum power**	14.23	8.39	66.93	20.31	14.21	42.34

Constraint	**Implementation**	**Step**
$MaximumEnergy = \sum_{i=1}^{8} Power_i * NoHours_i$	- Scaling power profiles - Clustering / centres are datapoints - Classification on clusters	D,G E F
$NoHours \in \mathbb{N}^+, \sum_{i=1}^{8} NoHours_i \leq 24$	- Clustering / centres are datapoints - Classification on clusters	E F
$MaximumEnergy > MinimumEnergy$	Not explicitly implemented, depends on regression quality!	B

Figure 65. Solution pipeline, error measures and constraints "implementation"

5.2 DISCUSSION

Overall, we have reached our goal in terms of prediction, as the baseline has been beaten by a comfortable margin and all constraints have been respected by our solution. Since we used a rigorous approach and have been attacking the problem under several angles, we can be confident that these results are solid and reliable. To come back to the initial question, we believe that our models can be used to produce a prediction over two years with trustable accuracy. Provided that the underlying context doesn't change, but we have no control over this, nor any way to know if it isn't the case anymore, typically if there is a fundamental change of processes by the powerplant operators.

Throughout the above steps, we have tried to make our work understandable and transparent, by describing in detail those steps and illustrating them abundantly. More details are contained within the Jupyter notebooks.

Looking forward, improving the overall accuracy still seems possible to us:

1) Spending time tuning MLP / RNN models for regression
2) Improving classification error with more advanced classifiers, which we tried with MLP, attaining limited gains

In the final solution, we would however advise the simpler models, as they allow a gain in interpretability even if their predicting power is slightly lower than then complex ones, as argued by some research (Green & Armstrong, 2015). They might not be state of the art but fulfil the need for understanding and trust that is key for most forecasts requested by human stakeholders who have to make decisions based on them.

However, this should not hide several shortcomings in this project:

- As we covered a wide theoretical area, maybe too wide, we didn't go very deep in certain areas, such as hyperparameter tuning, RNN and distance measures that might have deserved a deeper analysis
- Some potential inconsistencies remain in our input data (minimum > maximum, negative inflows), which might impact the results quality. Additional discussions with the stakeholders would be necessary to improve this.
- We observed that the total number of hours in power blocks seems to have a very different structure along time: it was quite different in 2015 and in 2018 for example. A clustering approach of the dataset as seen in other projects (Rushdi & Perera, 2018) might have provided better results by

segmenting the time horizon and building several models on them, which might prove challenging because of the limited amount of data.

- An important weakness in our project is the test/ train split strategy. Since no “validation” data set was used, the attained performance might be overestimated. The deployment phase will enable further measures and tuning.
- As very few exceptional events are were observed in our dataset, it is difficult to make sure they have correctly been taken into account. We tried to integrate them as well as possible, by having occurrences in both train and test subsets. Future will tell as well how successful this was.
- Another weakness relates to the fact that we have not tried to integrate other exogenous data. We didn’t have access to any other data related to the powerplant itself, but one could have tired of measure the influence of temperature forecasts or energy prices. Time was the limiting factor here.

6 CONCLUSION

In this project, we have shown that imposing constraints to predictive models can be done without changing their inner algorithm. The key here was to split the problem into cascading subproblems, forcing the multi-variate output's shape to reproduce input's through normalizing, reshaping and clustering, at the cost of an acceptable error, reaching satisfactory accuracy overall.

As we did not find any exactly similar problem during our literature research, we also hope that we contributed, if not to fill a small gap in the machine learning field, at least to shed light on it, concerning prediction with constraints in the context of energy forecasting.

Throughout the project, its stakeholders have shown great interest in its progress. According to them, "*the attained results exceed our initial expectations. Even if no decision has been made yet about deploying this solution and using it operationally, the insights it delivered will definitely help us improving our processes regarding powerplant management in the long term. The conclusion that the constraints can effectively be forecasted with a good accuracy is of great value to us.*"

On a personal key, we enjoyed this project very much, because of its challenging complexity and because of the variety of topics addressed. Maybe too many, as we spend lots of time reading material that could not be applied to this peculiar problem. But for us, this is the great benefit of this project, it allowed to go deeper in the theory addressed during the teaching but also other areas of ML, experiencing it all hands-on. This entire journey was a very rewarding and gratifying one, as in the end the strategy applied bore its fruits.

7 REFERENCES

Almalaq, A., & Edwards, G. (2017). A Review of Deep Learning Methods Applied on Load Forecasting. *16th IEEE International Conference on Machine Learning and Applications*.

Azevedo, A., & Santos, M. F. (2008). KDD, SEMMA AND CRISP-DM: A PARALLEL OVERVIEW. *IADIS European Conference on Data Mining* . Amsterdam.

Dong, X., Qian, L., & Huang, L. (2017). Short-Term Load Forecasting in Smart Grid: A. *2017 IEEE International Conference on Big Data and Smart Computing (BigComp)* (pp. 119 - 125). IEEE.

González, C., Mira, J. M., & Ojeda, J. A. (2016). Applying Multi-Output Random Forest Models to Electricity Price Forecast. *Preprints*.

Green, K. C., & Armstrong, J. S. (2015, 8). Simple versus complex forecasting: The evidence. *Journal of Business Research vol 68*, pp. 1678 - 1685.

Hong, T., & Shu, F. (2016). Probabilistic electric load forecasting: A tutorial review. *International Journal of Forecasting 32*, 914-938.

Jaydip, S. (2018). Stock Price Prediction Using Machine Learning and Deep Learning Frameworks. *6th International Conference on Business Analytics and Intelligence (ICBAI 2018).* Bangalore, INDIA.

Josh, P., & Adam, G. (2017). *Deep Learning - A practitioner's approach.* O'Reilly.

Kanakalatha, V. (n.d.). *ML | K-Medoids clustering with example*. Retrieved from geeksforgeeks.org: https://www.geeksforgeeks.org/ml-k-medoids-clustering-with-example/

Kaytez, F., Taplamacioglu, M. C., Cam, E., & Hardalac, F. (2015). Forecasting electricity consumption: A comparison of regression. *Electrical Power and Energy Systems 67*, pp. 431-438.

Lago, J., De Ridder, F., & De Schutter, B. (2018, 7). Forecasting spot electricity prices: Deep learning approaches and empirical comparison of traditional algorithms. *Applied Energy vol 221*, pp. 386 - 405.

Machines, A. I. (2012). An Introduction to Support Vector Machines. In L. Yuh-Jye, Y. Yi-Ren, & P. Hsing-Kuo, *Handbook of computational finance.*

Osowski, S., Siwek, K., & Markiewicz, T. (2004). MLP and SVM Networks – a Comparative Study. *Proceedings of the 6th Nordic Signal Processing Symposium - NORSIG 2004.* Espoo, Finland.

Pal, A., & PKS, P. (2018). *Practical Time Series Analysis.* Packt Publishing.

Palmer, A., José Montaño, J., & Sesé, A. (2006, 5). Designing an artificial neural network for forecasting tourism time series. *Tourism Management vol. 27*.

Raza, M. Q., & Khosravi, A. (2015). A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings. *Renewable and Sustainable Energy Reviews 50*, 1352–1372.

Ribeiro, M., Singh, S., & Guestrin, C. (2016). "Why Should I Trust You?": Explaining the Predictions of Any Classifier. *Conference: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations.*

Rushdi, M., & Perera, A. (2018). K-Medoids Clustering Based Approach to Predict the Future Water Height of a Reservoir. *International Conference on Advances in ICT for Emerging Regions (ICTer)*, (pp. 279-286).

Ryu, S., Noh, J., & Kim, H. (2016). Deep Neural Network Based Demand Side Short Term Load Forecasting. *Energies vol.10*, 1-20.

Torgo, L., & Gama, J. (1996). Regression by classification. *Advances in Artificial Intelligence, 13th Brazilian Symposium on Artificial Intelligence, SBIA'96* , (pp. 51-60). Curitiba, Brazil.

Weron, R. (2014, 10). Electricity price forecasting: A review of the state-of-the-art. *International Journal of Forecasting vol. 30*, pp. 1030 - 1081.

8 ABREVIATIONS

Abbreviations:

AI	Artificial Intelligence
ACF	AutoCorrelation Function
ANN	Artificial neural network
AR	Auto-Regressive
ARIMA	Auto-Regressive Integrated Moving Average
ARMA	Auto-Regressive Moving Average
CNN	Convolutional Neural Network
DL	Deep Learning
EDA	Exploratory Data Analysis
GRU	Gated Recurrent Unit
k-NN	K Nearest Neighbour
LIME	Local Interpretable Model-Agnostic Explanations
LSTM	Long-short term memory
MAE	Mean Absolute Error
MAPE	Mean Absolute Percent Error
ML	Machine Learning
MLP	Multi-Layer Perceptron
PCA	Principal Component Analysis
ReLU	Rectifier linear unit
R or R^2	Correlation Coefficient
RF	Random forest
RNN	Recurrent neural network
SVR	Support vector regressor
SVM	Support Vector Machine
RMSE	Root Mean Square Error
TS	Time Series
t-SNE	t-Distributed Stochastic Neighbour Embedding

9 APPENDIX

9.1 DETAILED PROBLEM PRESENTATION

A Swiss energy company is a shareholder in a hydroelectric powerplant. This company is retributed by receiving energy from the power plant every day. The value of this energy lies in the power plant flexibility, i.e. energy production can be triggered on demand, typically when energy prices are at the highest. To allow such flexibility, the power plant operator informs its shareholders on the previous day, using a dedicated guideline document, called an "offer", which contains the minimum energy production that must at least be produced and maximum power that can be produced on this given day (details provided below).

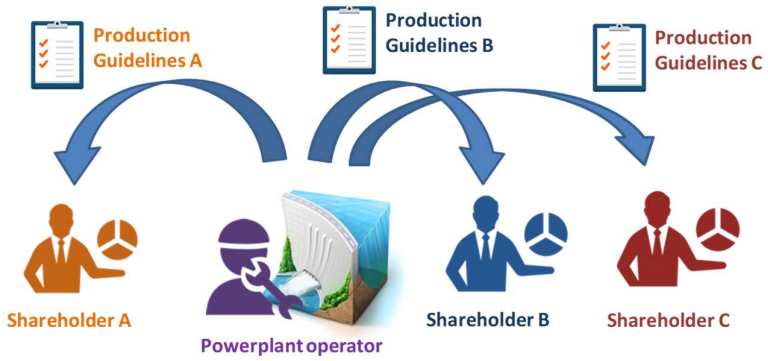


Figure 66. Production guidelines sent to powerplant shareholder

Using this guideline, the energy company traders can decide when to produce energy at what power, hence exercising the production's optionality. On top of this, the offer contains important information that influence these energy boundaries: how much natural inflows are expected, how full its reservoirs are, what constraints exist on its reservoirs (maximum level) and what constraints exist on its powerplant's turbines (in terms of power). These constraints result from maintenance work, both on the reservoirs and on the turbines. The diagram below illustrates these concepts (Figure 67).

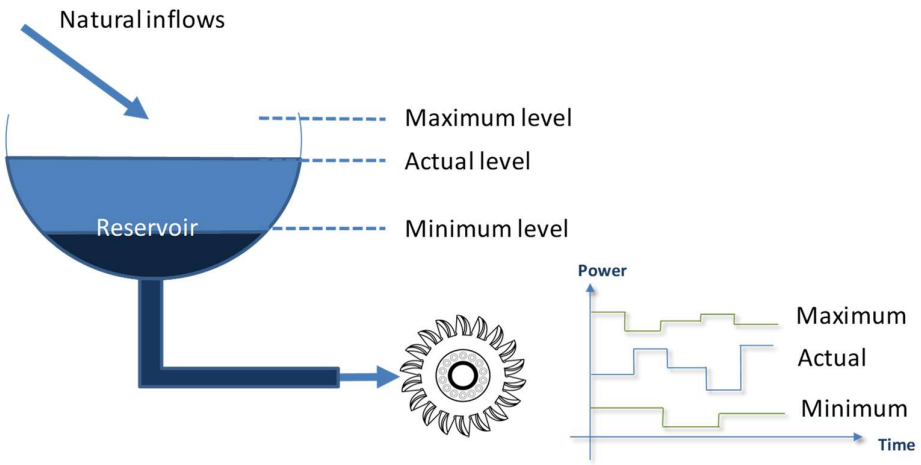


Figure 67. Hydroelectric power plant diagram

In order to develop a vision of its profits and financial risks, the energy company must forecast the guidelines values, so it can have an idea of how much energy it can produce on what day and hours in the long term. Therefore, we have been proposed us to develop a predictive model that would deliver a forecast for these quantities as accurately as possible, i.e. to forecast maximum and minimum daily energy and maximum power / number of hours (Figure 68).

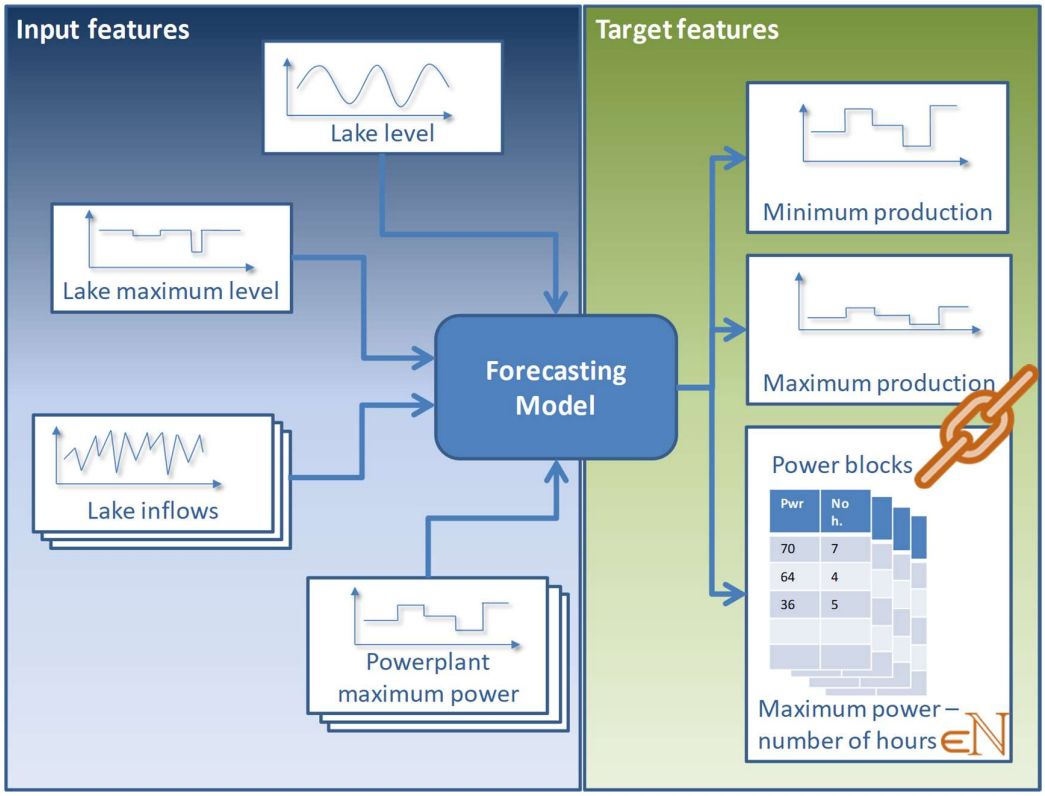


Figure 68. Model diagram

Assumption is that these values depend mainly of the other quantities, i.e. reservoirs level, inflows and constraints on reservoirs and turbines. But "how" is to be determined. Historical values are provided since 1.4.2014. Forecast horizon is two years, as forecast for explanatory variables exist over this time period, since they are already forecasted (average inflows and lakes levels) or result of human decisions that are known in for the future (maintenance time periods). The sketch below illustrates the actual offer: arrows point at target values to be forecasted; the rest of values are considered as explanatory variables (Figure 69).

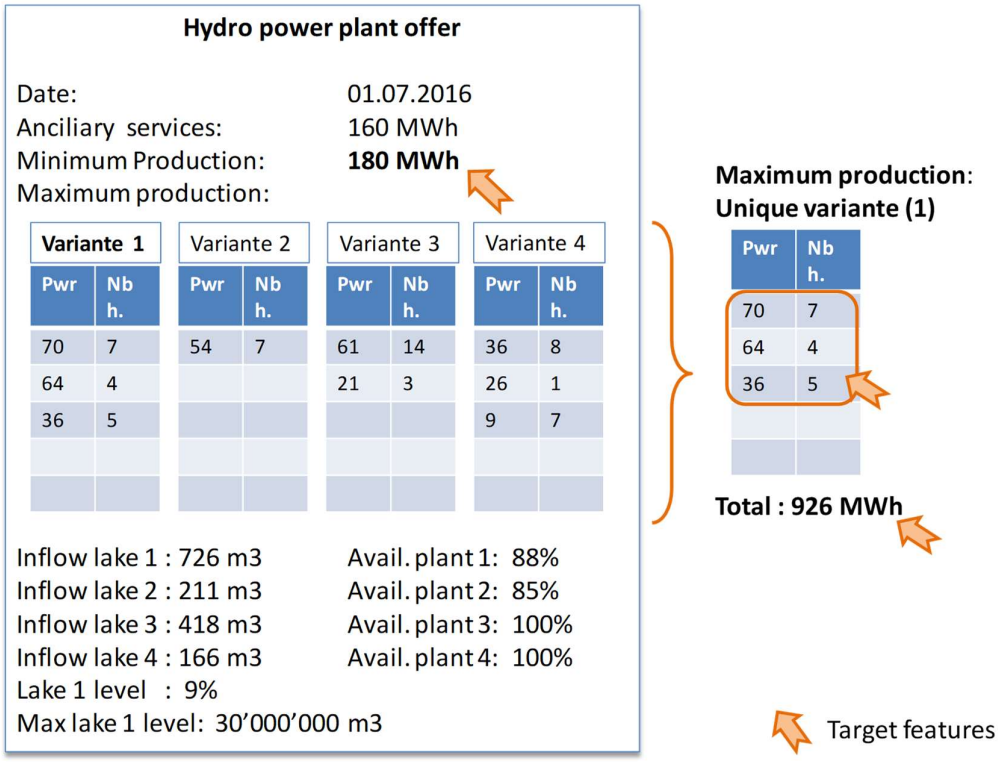


Figure 69. Offer document diagram

In details, one can see that the offer contains 4 "variants", i.e. options. This makes the problem unnecessarily complex and after discussion with the stakeholders, we consider only a "unique" variant, defined as the existing variant with lowest order (if 1 exist, it is 1, else if 2 exists, it is 2, etc). As well, we see that the maximum production is expressed as pairs of maximum power and number of hours. On the above example, traders can decide to produce up to 7 hours at 70 MWh, up 4 at 65 MWh, at up to 5 at 36 MWh. There can be up to 8 of those pairs.

We note that the forecast has in scope

1) Two energy values (as defined by the unique variant):

a. **Minimum production**
b. **Maximum production**, calculated as the scalar product of power times number of hours (for all existing pairs)

2) Several **pairs** of **power value** and associated **number of hours**, that we will refer to as "power blocks". Hours are whole numbers.

Important to keep in mind is the fact that the two last quantities are directly related, since maximum production is the sum of power values times number of hours. Hence our model must produce consistent values to this regard.

The power blocks are used by traders to trigger energy production decisions. Using the human readable representation, they know they can produce up to 7 hours at 70 MW power, etc. These hours can be used any time in the day and don't have to be in a row. So this a compact way to communicate the optionality offered by the hydro powerplant.

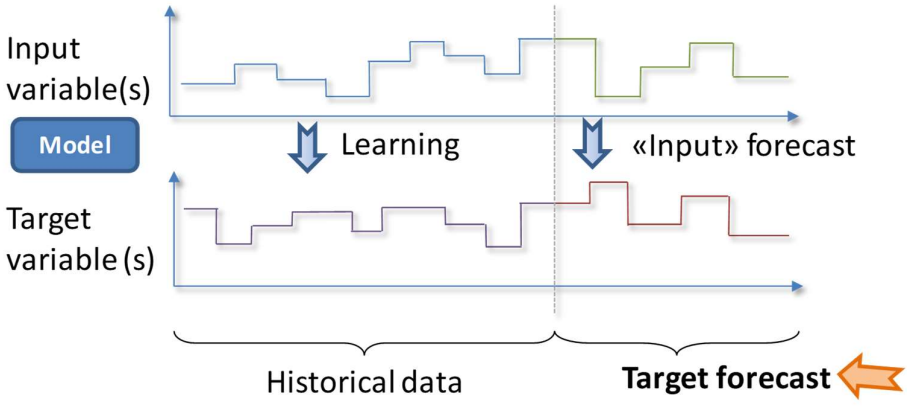


Figure 70. Forecasting problem time diagram

Although the problem is initially presented as a forecasting problem, we note that this is actually a regression problem, since the target variables are thought to be dependant of explanatory ones mainly and those are available in the future (Figure 70). Again "how" and "how much" are the questions our model will need to answer. On top, this is a time series problem, since we have data recorded on a regular time point, every day, on the entire time interval for both inputs and outputs. So, to sum up, we are facing a multi-objective regression time series problem.

At this stage, it is worth highlighting that there is no knowledge at all concerning how the target variables are generated at the powerplant operator side: it could be a manual process or an algorithmic one or a mix.

To us, it is a black box. However, some assumptions do exist: minimum power should be related to inflows; maximum power is clearly limited by turbine constraints, for example. It will be the model's aim to capture such relationships, that could be quite complex and time dependant, and reproduce them, extracting them from the provided data. As far as possible, this model's results should be interpretable gain insight on the shareholders' side about how the offer is actually built, hence opening the black box. The mentioned assumptions enabled the company to develop a simple deterministic algorithm (rule based) to produce a first basic forecast. We will use this forecast as a baseline for benchmarking our solution: it should at least perform as accurately to bring an additional value.

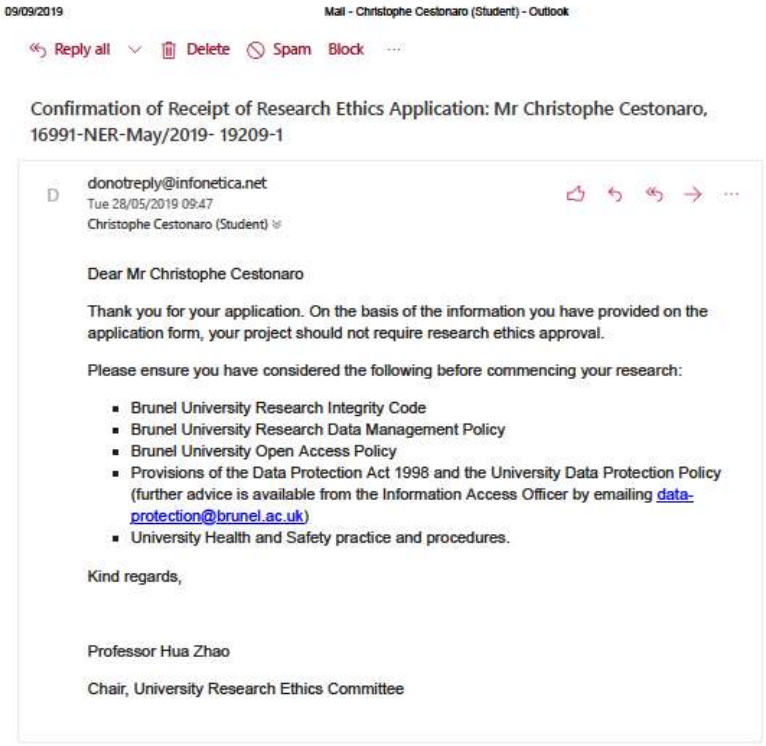
This problem is both specific and general in our view:

- it is specific, because in most energy forecasting problems, for example concerning load or renewables production, very little is known about the future and major factors as weather are not possible to forecast long in advance. It is specific as well since the nature of the process to model is not known, usually we would know its origin, be it a natural phenomenon (wind, solar production) or global human activity (energy load), etc. Here it could be an algorithmic process or manual one or a mix of both.

- it is general as it is key to forecast both energy and power in the energy sector as both have strong influence on infrastructure size and operational equilibrium. Same problem type could be faced in the different contexts although most of the time, energy is handled in a form of a continuous hourly timeseries (load curve, production curve) not hours blocks offering execution freedom. The constraints of number of hours being entire numbers is however proper to powerplants that offer high flexibility, i.e. whose power can be modified very quickly (mainly hydro and gas turbines)

9.2 ETHICS APPROVAL

Email response:



09/09/2019 Mail - Christophe Cestonaro (Student) - Outlook

Reply all | Delete | Spam | Block | ...

Confirmation of Receipt of Research Ethics Application: Mr Christophe Cestonaro, 16991-NER-May/2019- 19209-1

D donotreply@infonetica.net
Tue 28/05/2019 09:47
Christophe Cestonaro (Student)

Dear Mr Christophe Cestonaro

Thank you for your application. On the basis of the information you have provided on the application form, your project should not require research ethics approval.

Please ensure you have considered the following before commencing your research:

- Brunel University Research Integrity Code
- Brunel University Research Data Management Policy
- Brunel University Open Access Policy
- Provisions of the Data Protection Act 1998 and the University Data Protection Policy (further advice is available from the Information Access Officer by emailing data-protection@brunel.ac.uk)
- University Health and Safety practice and procedures.

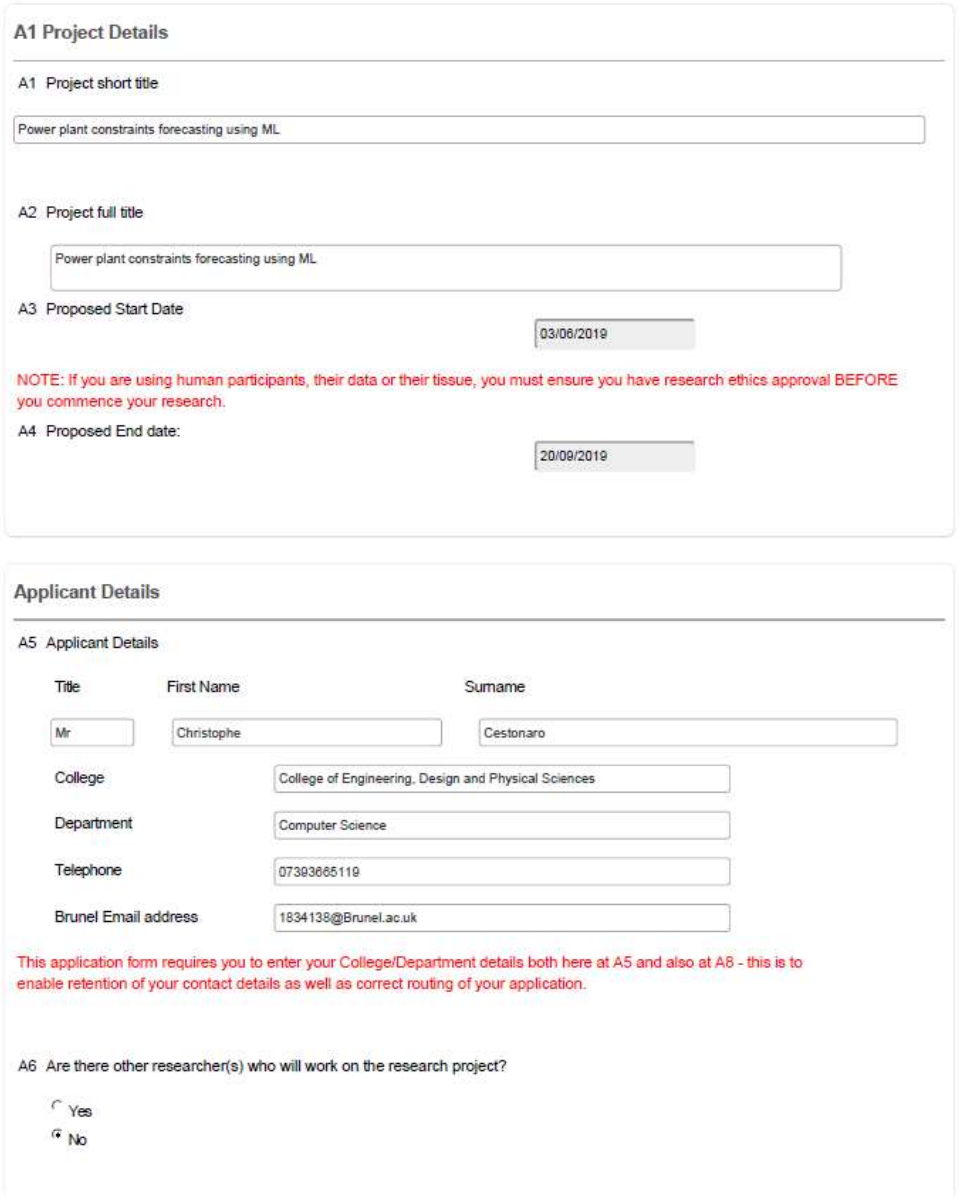
Kind regards,

Professor Hua Zhao

Chair, University Research Ethics Committee

https://outlook.office.com/mail/deeplink?version=2019090201.07&popoutv2=1 1/1

Full request form



A1 Project Details

A1 Project short title

Power plant constraints forecasting using ML

A2 Project full title

Power plant constraints forecasting using ML

A3 Proposed Start Date

03/06/2019

NOTE: If you are using human participants, their data or their tissue, you must ensure you have research ethics approval BEFORE you commence your research.

A4 Proposed End date:

20/09/2019

Applicant Details

A5 Applicant Details

Title	First Name	Surname
Mr	Christophe	Cestonaro

College	College of Engineering, Design and Physical Sciences
Department	Computer Science
Telephone	07393665119
Brunel Email address	1834138@Brunel.ac.uk

This application form requires you to enter your College/Department details both here at A5 and also at A8 - this is to enable retention of your contact details as well as correct routing of your application.

A6 Are there other researcher(s) who will work on the research project?

- ○ Yes
- ◉ No

Page 1 of 4

A8 Applicant Status: Please select the capacity in which you are carrying out the research:

Masters

A8-1 Please select your College (If you do not belong to a College, please select 'No College'):

College of Engineering, Design and Physical Sciences

A8-2a Please select your Department:

Computer Science

A8-3 Please select your Institute (if you do not belong to an Institute, select 'No Institute'):

No Institute

Student Details

A9 Student Number

18341838

A10 Module Name and Number

CS5500 Dissertation

A11 Supervisor Details

Title	First Name	Surname
Pr	XiaoHui	Liu

College	College of Engineering, Design and Physical Sciences
Department	Computer Science
Telephone	+44 1895 265989
Brunel Email address	xiaohui.liu@brunel.ac.uk

Risk Factors

A12 Are you submitting an application which will involve recruitment of NHS patients?

- ○ Yes
- ◉ No

A13

Does your research fit into any of the following security-sensitive categories?

- ☐ Commissioned by the military;
- ☐ Commissioned under an EU security call;
- ☐ Involve the acquisition of security clearances;
- ☐ Concerns terrorist or extreme groups.

- ☑ None of the above

A13 Does this research involve human participants, their data and/or their tissue? N.B. This includes any data obtained from or about human participants, including questionnaires or surveys.

- ○ Yes
- ◉ No

A13-1 Please justify:

No human related data is used in my research.

Research Integrity

A19 Have you completed the Research Integrity Online Training relevant to your field of research (via Blackboard Learn - Brunel Graduate School Research & Teaching Courses - Research Integrity)?

- ◉ Yes
- ○ No

If yes, please provide details:

Attempt Score 38 out of 57 points

Researcher/Applicant

J1 Researcher/Applicant Signature

- I understand that I cannot commence my research until full research ethics approval has been granted by the relevant research ethics committee.
- I confirm that the research will be undertaken in accordance with the Brunel University London Ethical Framework, Brunel University London Code of Research Ethics, and Brunel University London Research Integrity Code.
- I shall ensure that any changes in approved research protocols are reported promptly for approval by the relevant University Research Ethics Committee.
- I shall ensure that the research study complies with the law and Brunel University London policies on the use of human material (if applicable) and health and safety.
- I am satisfied that the research study is compliant with the Data Protection Act 1998, and that necessary arrangements have been, or will be, made with regard to the storage and processing of participants' personal information and generally, to ensure confidentiality of such data supplied and generated in the course of the research.

  (Note: Where relevant, further advice is available from the Information Access Officer, e-mail data-protection@brunel.ac.uk).

- I will ensure that all adverse or unforeseen problems arising from the research project are reported in a timely fashion to the Chair of the relevant Research Ethics Committee.
- (For members of staff and PhD students) I will undertake to provide notification to the Chair of the relevant Research Ethics Committee when the study is complete, or if it fails to start or is abandoned.

Signed: This form was signed by Christophe Cestonaro (1834138@brunel.ac.uk) on 17/05/2019 2:59 PM

Supervisor

Signature of Supervisor

- I have met and advised the student on the ethical aspects of the study design and his/her responsibilities in relation to the submission of this application and the research.
- The student has been made aware of and advised to read the University's Code of Research Ethics and other relevant documentation.
- The topic merits further research.
- The student has the skills to carry out the research.
- The consent form is appropriate (where relevant).
- The participant information sheet is appropriate.
- The procedures for recruitment and obtaining informed consent are appropriate.
- An initial risk assessment has been completed (where relevant).
- If there are issues of risk in the research, a full risk assessment has been undertaken and a risk assessment is attached.
- A DBS check has been obtained (where appropriate).

Signed: This form was signed by Prof Xiaohui Liu (xiaohui.liu@brunel.ac.uk) on 28/05/2019 9:47 AM

9.3 CODE – JUPYTER NOTEBOOKS

Sample file provided here as illustration. Full code provided as separate archive file, for size reasons