

PROJECT WORK

Implementing Machine Learning model for forecasting household Solar energy production, and advising on electricity usage

SUMMARY

Machine Learning and Artificial

Intelligence are enhancing ways how we can use data for improving and optimizing “almost anything”. This project work implements a simple, but working model, a ‘Minimum Viable Product’, that advice on use of Solar Energy in a household to optimize financial impact. The project is also a learning experience for the processes and requirements for developing operable and maintainable ML/AI services.

markku roiha

Data, Analytics & AI for Professionals,
Aalto Executive Education, Nov 2022 –
May 2023

Table of content

Acknowledgement	3
<hr/>	
Project subject and the background	4
<hr/>	
Selecting the use case	4
<hr/>	
Project walk-through	6
<hr/>	
The project interpretation of the CRISP-DM and ML workflow	7
<hr/>	
Gate 1: Business understanding	9
Insights on Business understanding	9
Gate 2: Data Understanding and Gate 3: Data Preparation	10
Insights from the data analysis and preparation	11
Gate 4: Model Development	12
Insights of the Gate 4: Model Development	13
Gate 5: Model Evaluation	14
<hr/>	
Gate 6: Model Deployment	15
Insights from the model deployment and use	18
What's next...?	18
<hr/>	
...For the project	19
<hr/>	
...For the MLOps	19
<hr/>	
Closing Thoughts	20
<hr/>	
Appendix 1: Machine Learning Canvas	21

Appendix 2: Data preparation extract – calculation Solar Energy production for training data

..... 22

Appendix 3: Example of a CI/CD pipeline for a Machine Learning project 23

Pictures and tables..... 24

Sources 25

Acknowledgement

I would like to thank Professor Jukka K. Nurminen, Department of Computer Science in University of Helsinki, for his invaluable support through interesting, clarifying discussions, and guiding towards insightful study material around MLOps helping to reach comprehensive understanding on the subject.

Project subject and the background

This project work is part of my 'Data, Analytics and AI for Professional' studies in Aalto Executive Education, and targets to utilize and test my new knowledge gained during the 'DAAP' learning program as well as in the prior 'Python for AI' training. The project has two objectives.

- Test, and deepen learnings from the 'DAAP' and 'Python for AI' sessions and thus verify my understanding on the required logic, skills and expertise and practicalities related to Machine Learning/Artificial Intelligence solutions and their development, the process phases for developing Machine Learning and Artificial Intelligence solutions.
- Implement a working ML/AI solution solving a real-life problem, consider the architecture of a solution, and be able to map out similar solutions for both business decision-makers and the experts, and lead development and operations teams not only to develop, but also to operate and maintain relevant ML/AI services.

My profession for the past seven years has been in operations, more precisely in delivering IT (mostly infra) services to customers in the Nordics. This background has affected this project, and its objectives to clarify what is relevant for effective MLOps (new paradigm for streamlining taking solutions to production and maintenance and monitoring), and which critical choices we need to do during implementation to create maintainable, operable, and scalable ML/AI services and thus optimize the life-time costs of the ML/AI services. This is important as we easily focus on development costs, while the maintenance phase constitutes 67% of the traditional software lifetime costs (IML4E project 2022). And we should not forget the people aspect, the pressure a complicated service lays on developers and operations people with negative effect on job satisfaction and employee turn-over.

Selecting the use case

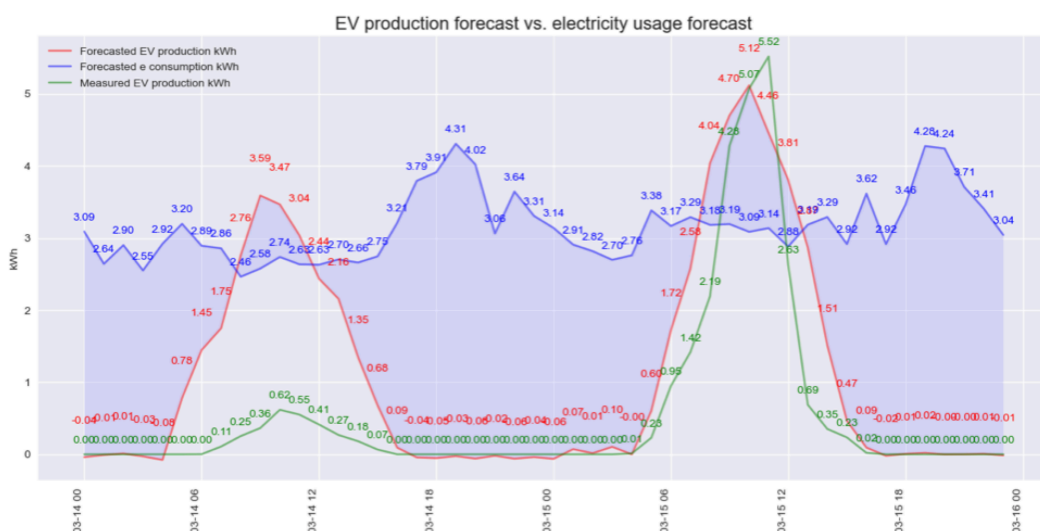
The tangible outcome of the project is a simple, working software implementation that solves a business question "how the household can optimize the use of Solar Energy from the own solar panel system, and get guidance when to operate the key electric appliances/loads". But the more important are the intangible

outcomes of improved understanding and skills related to Machine Learning software development, and abilities to facilitate collaboration and communications between various business and technical experts and lead the work for developing, and delivering (operating, maintaining) these new types of business-critical services.

This use case has two key technical requirements.

- Step 1 'Predictive analytics' solution: Forecast the hour-by-hour Solar Energy production for the next 1.5 days (afternoon 16.00 to next day 00.00). For instance, be able to answer "how much excess solar energy will be at 12.00 tomorrow?"
- Step 2 'Prescriptive analytics/Decision Support': Suggest start times for using defined electric appliances (car charging, water heating, dishwasher) to minimize/optimize electricity costs during the same period. For instance, "When should I start charging the car to minimize the electricity costs?"

The implementation should offer the information in a meaningful graphical format such as the example of prediction below.



Picture 1 Example graph with forecasts for Solar Energy production vs. real production vs. forecast electricity consumption in the household

This use case was selected first and foremost because it had a ‘business need’, and a strong personal interest. Electricity prices in autumn 2022 were extreme, and the household invested in an EV system (i.e. Solar Power system), and it was a major interest how to optimally consume the produced solar energy as

- Electricity prices during a day fluctuate, sometimes heavily (e.g., 1c/kWh at night, 10c/kWh during day)
- EV system produces different amounts of electricity depending on the “features” like weather or time of the year, producing excess electricity that can be either consumed by the household or provided back to the grid (with lower compensation)
- and there are some ‘appliances’ such as charging the hybrid car whose use can be controlled to optimize the electricity bill.

The other key reason to choose this use case was the availability of the data, i.e. independence on any other person or proprietary business data, giving a full control and freedom of implementation. The data is either ‘owned’ by me (inverter data for EV production), or available from the public sources like Finish Meteorological Institute for weather data, or from Fingrid (Finland’s electricity transmission operator) for solar energy production in Finland (for training purposes). Only problematic data is the hourly electricity prices where the API is chargeable (thus handled manually in this project).

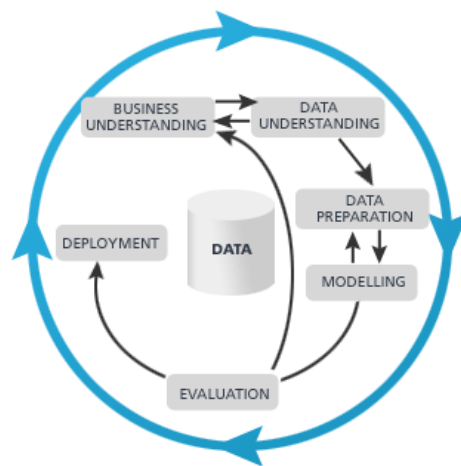
Project walk-through

The project implementation targets to consolidate the learnings from the DAAP program and follow logical steps from Definition and Requirements to final Validation and Deployment. The below two methods have given significant guidance how to best approach the Data Mining and consider the whole process of ML service implementation.

considering not only Data Mining but the overall development and operations i.e., Development and Operations of software & data solution.

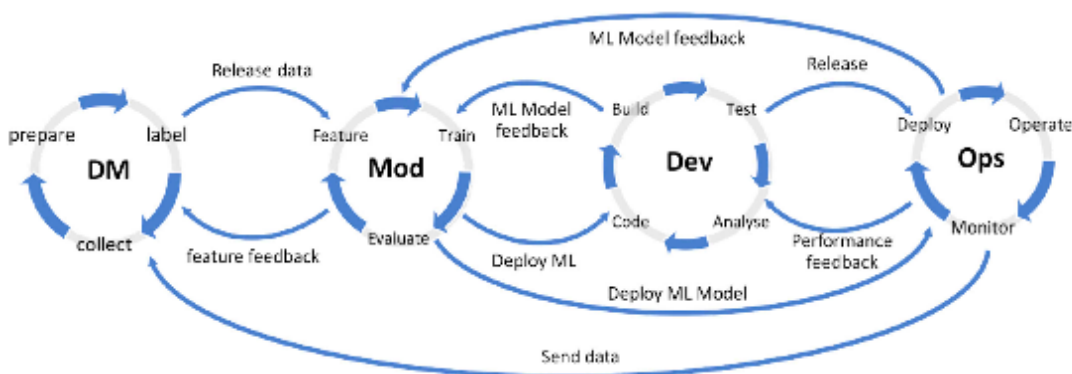
The project interpretation of the CRISP-DM and ML workflow

In developing an ML service our focus is easily in Data Mining and training the model (CRISP-DM), and while it is the “clue” of an ML service, considering also the support and the maintenance of the services in the on-going service phase, the ML iterative process and software development practices like DevOps need to be combined (Mehrdad Saadatmand (RISE 2019).



(IVVES project 2020)

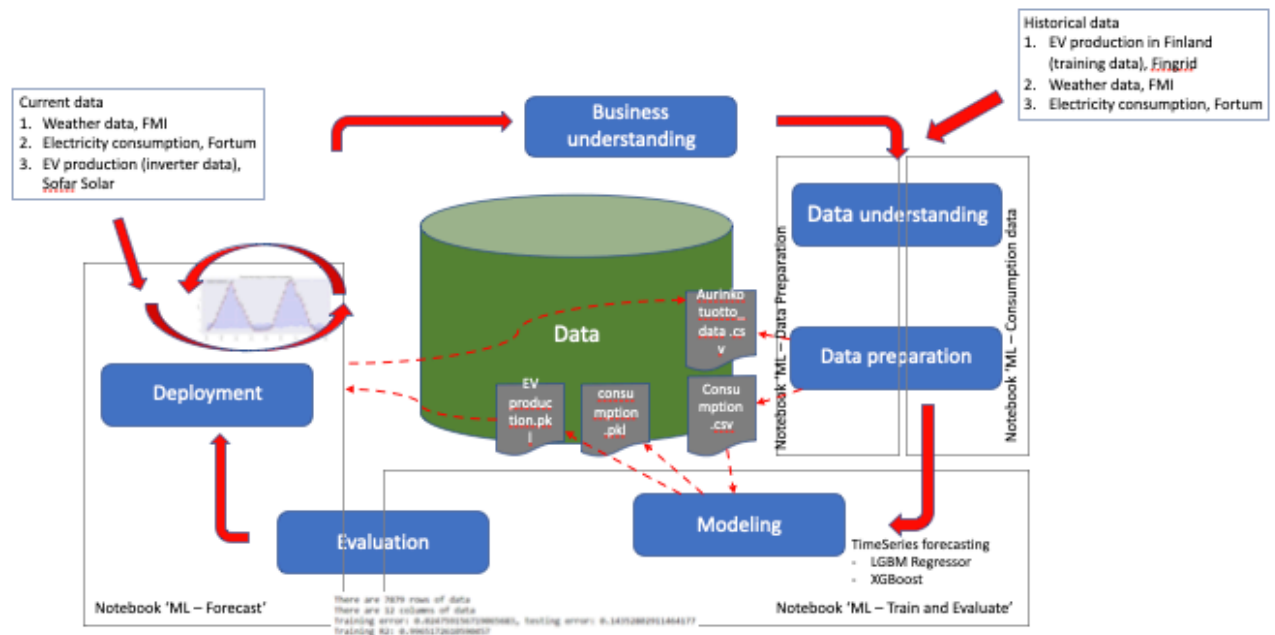
Picture 2 Cross-Industry Standard Process for Data Mining (CRISP-DM) model is guiding the steps for this project



Kuva 3 ML workflow and DevOps process integration

(L. E. Lwakatare 2020)

The project implemented the service in the Jupyter Notebook using Python as the programming language. The functionality is divided into different Jupyter Notebooks that logically implement specific functionality or functionalities of the CRISP-DM model as in the below picture. The idea behind the split was to be able to separate the functionality into smaller independent modules that are easier to review and debug as necessary. Files are used as the means to transfer information to next steps, i.e., next module. They are also an easy way for permanent storage of data and information and artifacts such as trained models. A serious implementation for commercial solution should consider proper tooling for storing datasets and other artifacts, managing features and versioning of artifacts along the development and maintenance process.



Kuva 4 Logical architecture of the solution

The next chapters walk through the implementation steps of the 'Household Solar Energy usage optimization' service. Each chapter has an 'Insights' part adding the learnings and findings related to the development as well as the considerations related to operations i.e., extending to MLOps. Due to resource and time constraints, this project implemented only minimum requirements, a Minimum

Viable Product (MVP), but targeted to make observations for a real commercial ML/AI service development.

Gate 1: Business understanding

For the initial definition and considerations of the project scope, the 'Machine Learning Canvas' was used (see Appendix 1: Machine Learning Canvas). It is an easy, and practical tool for collecting initial information of the ML service project and communicate and develop the ideas between the relevant stakeholders such as business owners and experts, and technical teams. Such information is critical to get people with different background aligned on project directions and get an agreement.

From this learning experience, the most important aspects where to define the objectives, and understand the data sources which were described in the chapter Selecting the use case.

Insights on Business understanding

This project was only experimental, learning experience. For a commercial implementation, The Machine Learning canvas is not enough, though. While it limits itself to key functional requirements, there are much more to an operational ML/AI service coming from for instance 'Responsible AI' requirements. The project MUST define and agree on relevant non-functional requirements, and to what extend the project needs to take them into consideration. In addition to requirements for fairness, transparency (explain ability), etc. requirements from 'Responsible AI', it is crucial to have the Operations experts involved with their input related to the scalability, operability, and maintainability of the solution. By this it is possible to affect the life-time costs such as need for supporting the service (service downtimes, and incidents), or operating the service (e.g., reducing manual effort in data collection, service monitoring and alerting).

Operations (DevOps) can also gather requirements in this step and be able to support and speed up the ML solution development. Ops engineers with their expertise on CI/CD pipelines can suggest and deliver needed platforms and tooling for software and model deployment and (automating) testing.

While not emphasized by the previous ML methods, a further learning from the project is the importance of the design and architecture. Setting up the guidelines and good practices in the beginning can save from a lot of iterative work i.e., going back and forth modifying datasets or having to re-code parts of the logic.

Gate 2: Data Understanding and Gate 3: Data Preparation

Starting point for the data gathering was the consideration, what data is needed to be able to forecast solar energy production and calculate the excess production. Deciding which parameters, features in Machine Learning, was a very straight-forward decision process in this project. Intuitively the solar energy production is affected by the position of the sun (represented by the time of the year/date, and time of day), weather conditions (open data from Finish Meteorological (Institute 2023), and the qualities of the production system (such as peak capacity). EV system was installed only in Nov 2022 meaning there was no production data, and thus the Solar Energy production in Finland provided by provider Fingrid (Fingrid 2023) was selected as a “proxy” to represent production data for training purposes. Calculation of the excess solar energy also required data for household’s electricity consumption, and it is available from the consumer portal of the provider Fortum (Fortum 2023).

aurinkotuotto_data_jatkuva2

Alkuaika UTC	Dates	Year	m	d	Hour	Cloud amount (1/8)	Air temperature (degC)	Horizontal visibility (m)	Wind speed (m/s)	forecast	production	consumption	consumption forecast
2022-01-01 00:00:00	2022-01-01	2022.0	1.0	1.0	0	8.0	-1.2	2295.0	1.6	-0.0283401732898877		2.79	3.029666789002471
2022-01-01 01:00:00	2022-01-01	2022.0	1.0	1.0	1	8.0	-1.0	5757.0	1.9	-0.0237653690395282		2.21	2.626719611078561
2022-01-01 02:00:00	2022-01-01	2022.0	1.0	1.0	2	7.0	-1.1	9041.0	3.1	-0.0172503864321305		2.93	2.684403119855933
2022-01-01 03:00:00	2022-01-01	2022.0	1.0	1.0	3	8.0	-1.0	17250.0	4.8	0.0074162223989714		2.21	3.181811240623301
2022-01-01 04:00:00	2022-01-01	2022.0	1.0	1.0	4	8.0	-2.5	22036.0	8.2	0.027758836038949		3.12	3.227903772415256
2022-01-01 05:00:00	2022-01-01	2022.0	1.0	1.0	5	8.0	-3.4	48969.0	6.2	0.0354361564428012		2.49	3.1480770899557378
2022-01-01 06:00:00	2022-01-01	2022.0	1.0	1.0	6	8.0	-3.5	50000.0	4.7	0.1713056630440752		2.54	3.558498311026966

Figure 1: Extract of the master dataset

The different input datasets were analyzed, cleaned up and features engineered i.e., modified to prepare them for consolidation into a single dataset as in the below example extract. This meant for instance unifying the data such as date-time, removing rows of empty data cells. Also, to mitigate the lack of own production data, it was necessary to calculate a feature value for solar energy production. Example of this calculation is shown in the Appendix 2: Data preparation extract – calculation Solar Energy production for training data

Insights from the data analysis and preparation

“Know your data”. “Look at the data”. “Data preparation is 80% of the ML effort”. This project confirms these common “truths”. While on logical level, the work is clear, this step requires careful planning, thorough reviews, and plenty of coding to prepare a solid dataset that is ready for training and testing purposes.

First, the quality of the data was a cause for some incoherent test results which was very difficult to identify from the training and test results. Thus, in addition to analysing the data by the data scientist, I consider it extremely important to involve a Subject Matter Expert to contribute with his/her understanding to improve the quality of the created datasets.

Second suggestion that helps to reduce workload in later phases is to plan and document the dataset and the features clearly in this phase. This project was an experimental learning experience, and shortcuts in planning slashed back in later phases requiring iterating and come back to the structures of the datasets several times triggered by new learnings during training and building new functionality. Proper documentation of the datasets saves effort in later phases even for a single developer but is truly critical when more people are involved requiring coordination between people and onboarding new people. Thus, documentation and clarity on data should be agreed from the start. The right (Data Pipeline) tooling and expertise of a Data engineers can be of significant help.

From the operational point of view, the decisions and implementation made here are critical, and affect the work effort and stability of the service in the production phase. Automating the data ingestion, building the measures to monitor and alert the changes in the input data, being able to manage gracefully the bad or missing data will directly affect the quality of the service, and thus customer experience. Also, the operational team will appreciate fewer incidents, and less needs for emergency actions.

Gate 4: Model Development

While auto-sklearn was experimented to evaluate the best possible prediction model for the work, the LGBMRegressor algorithm was selected for predictions. There are examples where the algorithm is used for similar regression jobs for time-series data (Ansoleaga 2022) successfully, and early experiments and the cross-validations (see picture) already provided good confidence that the LGBM Regressor algorithm works, and thus focus should be put on improving the data.

```
# Estimate the error of your model with the testing set
predictions_test = model.predict(production_testing_this_time[["m", "d", "Hour", "Cloud amount (1/8)", "Air temp
mses_test.append(mean_squared_error(production_testing_this_time["production"], predictions_test))

print(f'Training error: {np.mean(mses_train)}, testing error: {np.mean(mses_test)}')
print(f'Training R2: {np.mean(r2_train)}')
```

There are 7879 rows of data
There are 14 columns of data
Training error: 0.024759156719065686, testing error: 0.14352802911464177
Training R2: 0.9965172610590457

Figure 2: Cross-validation of the LGBM model with some good results

Training and evaluation of the models are implemented in a separate Jupyter Notebook mainly to help developing, testing and troubleshooting the solution. The Notebook uses the master dataset file from the 'Preparation' notebook as an input and stores the trained model in a file using the 'pickle' library.

Table 1: Visualization of the training dataset (Pandas DataFrame)

```
In [17]: # Visualizing training data and forecasted EV production
df_forecast.describe().T
```

Out[17]:

	count	mean	std	min	25%	50%	75%	max
Year	7879.0	2022.000000	0.000000	2022.000000	2022.000000	2022.000000	2022.000000	2022.000000
m	7879.0	6.066506	3.185975	1.00000	3.000000	6.000000	9.000000	12.000000
d	7879.0	15.691966	8.856386	1.00000	8.000000	16.000000	23.000000	31.000000
Hour	7879.0	11.486610	6.922634	0.00000	5.000000	11.000000	17.000000	23.000000
Cloud amount (1/8)	7879.0	4.558066	3.547509	0.00000	0.000000	7.000000	8.000000	9.000000
Air temperature (degC)	7879.0	7.683945	8.413359	-19.80000	0.700000	7.800000	13.600000	28.900000
Horizontal visibility (m)	7879.0	30300.456276	15299.828157	178.00000	17723.000000	32143.000000	44040.000000	50000.000000
Wind speed (m/s)	7879.0	4.078893	1.934920	0.00000	2.700000	3.800000	5.200000	13.600000
forecast	7879.0	1.762060	2.625320	-0.35177	0.000317	0.167607	2.945991	12.190317
production	7879.0	1.762183	2.642896	0.00000	0.000000	0.098795	2.925301	12.249275
consumption	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
consumption forecast	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN

The calculated predictions (for solar energy production) are stored into the master dataset to complement the data.

Insights of the Gate 4: Model Development

At least in this project, developing the model, that is, finding and training a suitable model, was possibly the easiest step. Investigation of similar use cases led to using the LGBMRegressor algorithm, which seems to predict the values well enough for this application of predicting the excess solar energy in the household.

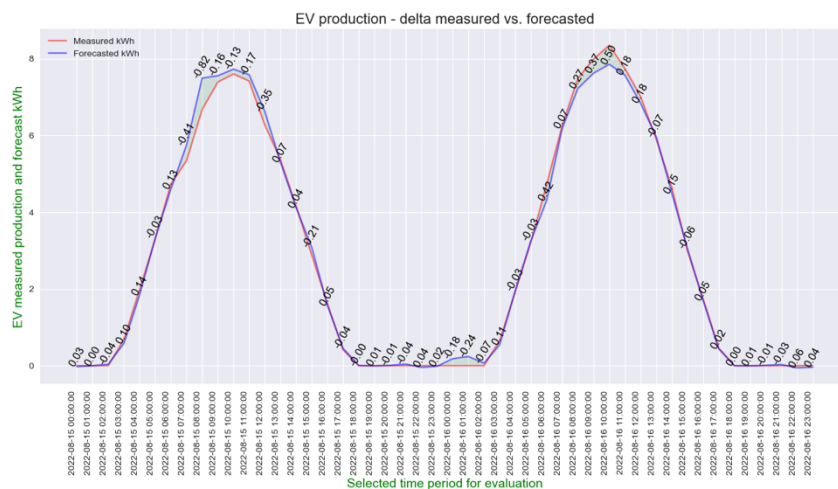


Figure 3: Testing - Real data vs. predicted data based on trained model

However, as explained earlier, there is no data yet from the Household's own EV system, and it is to be expected that the real data will be different, and the production data will only be improved over time when more and more real measurement data will be available. Still, the selected algorithm seems to fit well for the use case.

While preparing the data and developing the code around the trained model, the developers should have the clarity on the operational requirements affecting the operability, maintainability, and availability, and consider them here and not having to sticker them afterwards. Consideration should be put especially on resiliency for changes in the data. The solution should not have an outage if receiving bad or missing data but be able to gracefully manage the situation and alert the operations team. Also the inevitable drift in input data or model performance needs to be measured, logged and alerted.

Gate 5: Model Evaluation

Model development and model evaluation were parallel, iterative actions and difficult to separate in this project, but in a professional implementation this is where the collaboration of the data scientist (developing the model) and the Subject Matter Expert (knowing the business needs) becomes crucial. For this project, evaluation was simple due only a few experiments and making quantitative analysis, in this case calculating their MSE (Mean Squared Error) and R2 scores (Figure 2). However, in a professional implementation project the number of experiments easily grows, and keeping track of experiments becomes a nightmare. Thus, managing the different versions of the experiments and their artifacts (trained models, dataset versions) needs to be planned well to be able to trace and troubleshoot if necessary.

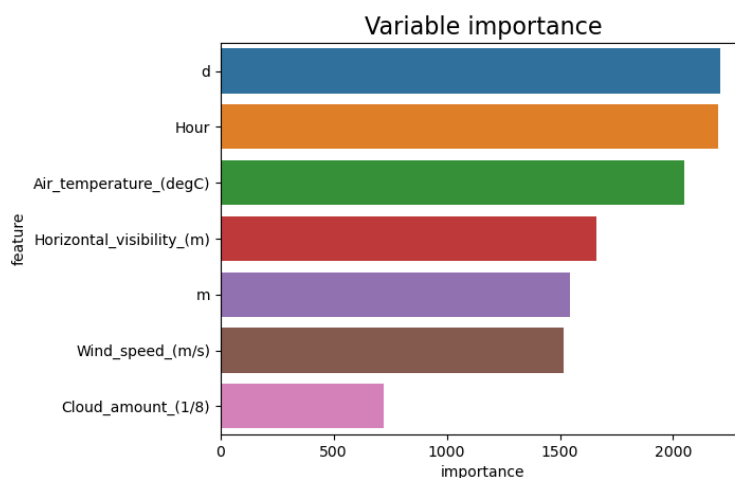
In addition, there are qualitative criteria to evaluate the model's performance. This is where the business SME plays a critical role evaluating how well the model fits the purpose, and solves the initial business problem, and to be able to do that, the model needs to be understood and explainable (Treveil 2021).

The below picture shows a technique used in this project explaining the relative importance of each feature. This should help the SME with verifying the logic, and when not happy, working with the data scientist to dive deeper. In case of predicting the Solar Energy production, some results are very logical such as the importance of the hour of the day. Some of the results are surprising instead such as the relatively low impact of the cloudiness. With more time, this would be something to investigate more.

```
#create a dataframe with the variable importance of the model
df_importances = pd.DataFrame({
    'feature': model.feature_name_,
    'importance': model.feature_importances_
}).sort_values(by='importance', ascending = False)

# plot importance
plt.title("Variable importance", fontsize = 16)
sns.barplot(x=df_importances.importance, y=df_importances.feature, orient='h')
plt.show()
```

Table 2: Variable importance of the features



Gate 6: Model Deployment

Due to experimental nature of the project, the solution runs currently only in the development environment in the Jupyter Notebook with very little requirements for the capacity and infrastructure. Using the model with new data brings up new information and knowledge. Below picture compares the prediction by the trained model (red line), and the true measurements from the household's EV system (green line).

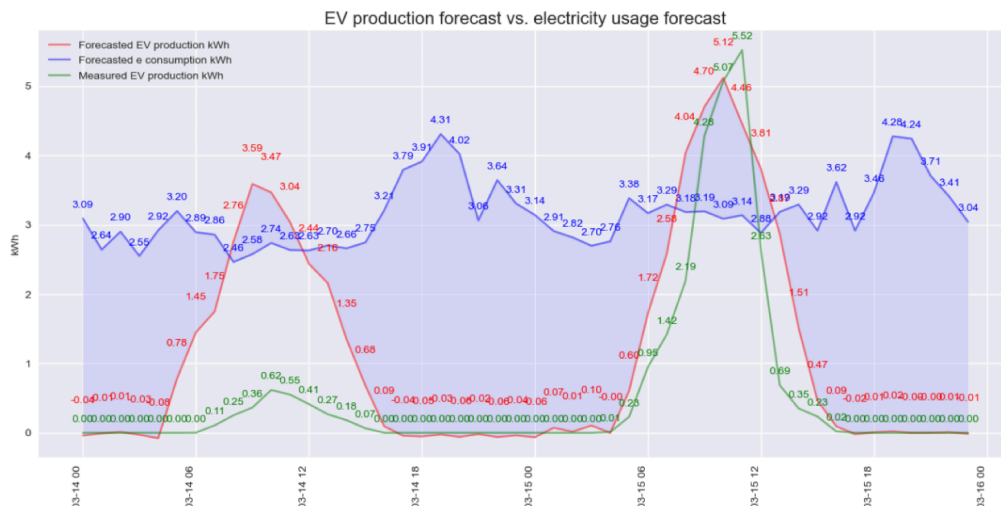


Figure 4: Comparing forecast from the model with the data from the real EV production system

What we can see from the picture, on the second day we can see similarity with the forecast and real data, the first day is completely different. The major difference, explaining such significant differences, was the slush on the solar panels, which cuts the production. Such circumstances were not taken into consideration in the model. Forgetting such exceptional conditions, the model can be considered good enough for providing predictions and make suggestions to run the selected electric appliances.

To make these recommendations, the 'Forecast' part of the solution makes a brute force calculation over the selected 32 hour time period (16.00 until 00.00 next day), and provides information on the most optimal solution to start the selected three appliances. For the sake of interest the calculation is done also for the worst option and the forecasted total amount of solar energy in the (32 hour) period as in below screenshot.

Table 3: Output from the cost optimization calculation

The period for investigation is between 2023-04-14 16:00:00 and 2023-04-16 00:00:00
Excess solar energy in that period is forecasted to be 42.66 kWh

Worst cost is -221.31 Eurocent, when starting

- CARCHARGING at 2023-04-15 19:00:00
- WATERHEATING at 2023-04-15 19:00:00
- DISHWASHER at 2023-04-15 19:00:00

The 1 most economical options are

Balance of 13.13 cent. (Positive value = compensation, negative value = need to pay)
Starting...

- CARCHARGING at 2023-04-15 09:00:00
- WATERHEATING at 2023-04-15 05:00:00
- DISHWASHER at 2023-04-15 08:00:00

For a more sophisticated presentation, the solution also prints an html page with hover text providing information about excess solar energy (blue line) and plotting the recommended hours for the selected three electric appliances as in the below picture.

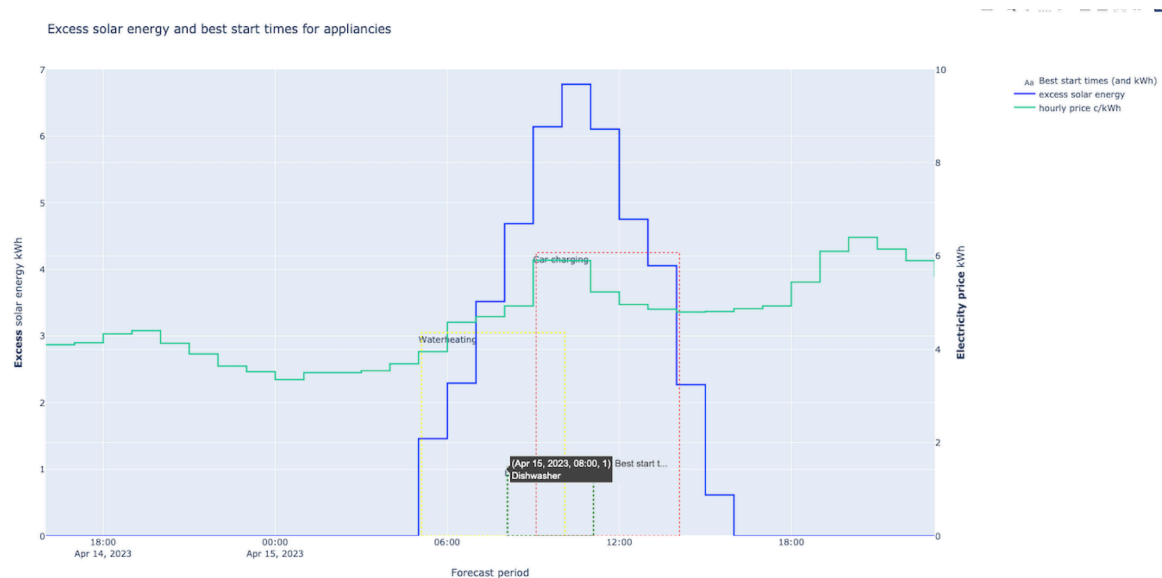


Figure 5: Final output from the optimization - predicted excess solar energy and recommendation for using the electric appliances

While the service is used for the new time period and fetching and preparing the new data, it appends (concatenates, in Pandas DataFrame terms) all new data into the master dataset. The idea is to enable easy model re-training as more and more data is gathered over time.

Insights from the model deployment and use

While deployment is yet to be done on a real server for this experiment, the deployment is something that a professional project following the best MLOps principles should start to plan parallel to the requirements definition. The criticality of the service affects the architecture and the platforms for the deployment, and the requirements for operability and maintainability may have relevant requirements for developing the software side of the solution. Examples of such critical considerations are for instance.

- Drift detection: monitoring and alarming changes in the input data affecting the accuracy of the predictions i.e., accuracy of the model, and
- therefore, trigger re-training of the model with new data.
- Deployment strategy of the re-trained model: it might make sense to deploy the new model only to limited use and monitor the performance before fully replacing the old model.

To guarantee that these requirements are incorporated in the planning from the beginning, it is therefore important to involve the operations (or DevOps, in other terms) expert early.

At this phase all the shortcuts made during the development becomes visible. As this project was more proof of concept rather than an idea of a daily tool, all those shortcuts for usability are visible while using the service. The key gap is the intake of the new data that requires some manual effort, the prime example being the hourly electricity prices that need to be entered manually (as API is available only through chargeable subscription).

The second improvement for the development backlog is the ability to monitor the performance drift. A simple calculation and logging of the RMSE (root-mean-square error) for the latest predictions using the scikit-learn library should already provide useful information and be very simple to implement.

What's next...?

...For the project

While the 'Household Solar Energy forecasting' service provides the targeted output, it is too laborious to use on daily basis, and has a lot of "technical debt" before it is ready for daily use. My plan is to finetune certain key gaps related to usability such as improving the API implementation and triggering the daily updates automatically.

Further, the solution is currently experimental and only available on a laptop. To make the results available for the larger audience, the plan is to deploy the application on a server such as Azure Cloud and be able to give any interested person access to the information on an html page.

...For the MLOps

If in this project I wanted to enhance my prior experience on the IT Operations and understand "everything" that is relevant for operating and maintaining ML/AI services in business context, I fell short of the goal. To the traditional software development, Machine Learning adds the data that needs to be gathered, managed, versioned, has its own life cycle and specialists, and changes constantly, potentially requiring changes in the solution.

The whole life cycle from the initial idea to deployment, and finally retiring old models is huge, and requires expertise of many individuals each being an expert on their specific area. We easily focus on areas of Data Scientist and the software developers doing Data Mining and the Machine Learning models and the predictions, but for a scalable service, it requires coordinated way of working, practical tooling to help in managing the code and the data, and the versions, automating deployment and testing where reasonable, and finally monitoring and alerting. After this project I appreciate more and more those (DevOps) experts who know and deliver these critical elements.

Instead, I have got a glimpse of what needs to be done in different steps of development, and what expertise is needed in various steps. No matter how excellent data scientist, if the service can't be operated, it will not fulfill the business needs, and vice versa. Creating a great ML/AI service is a






cooperation of many experts covering the business, data, software, and infrastructure experts bringing in their knowledge and tools. Deep subject expertise is needed, but there is also a need for generalists who can bridge and facilitate the collaboration, ensure there is understanding within the Team. I trust this exercise has taught me a lot on this.





Closing Thoughts

Creating a Machine Learning service is easy. Creating an AI service that can be trusted and has significant effect on business, is operation able, is maintainable, is hard. "Anyone" can do the first, but you don't want to trust the second just to anyone. Not to mention, the service is not rigid, it will need to continually learn and (self-)adapt to remain accurate. Traditionally, the operations and maintenance costs are majority (67%) of a solution life-time costs, and the decisions made in implementation have a huge effect on required maintenance effort (=costs) while also on vulnerability/resilience towards errors from humans or data.

This project work implemented a (simple) Machine Learning experiment and turned some weather data and household electricity measurements into a 'Prescriptive' AI service. It does what it was originally planned to do, give support decisions on optimizing the household electricity consumption with excess solar energy. From this perspective I'm personally satisfied with the MVP (Minimum Viable Product) implementing the initial requirements of the customer. Development will, however, continue to make it usable on daily basis.

Appendix 1: Machine Learning Canvas

PREDICTION TASK 	DECISIONS 	VALUE PROPOSITION 	DATA COLLECTION 	DATA SOURCES 
<p>Type of task? Entity on which predictions are made? Possible outcomes? Wait time before observation?</p> <p>Ph1. Regressio: paljonko tiettyinä tunteina tulee aurinkosähköä</p> <p>Ph2. "kannattaako tietynä hetkenä käyttää ylimääräinen sähkö omiin tarpeisiin vai käyttää pesukonetta, sähköauton latausta, yms. Muuna aikana</p>	<p>How are predictions turned into proposed value for the end-user? Mention parameters of the process / application that does that.</p> <ul style="list-style-type: none"> - Laske jokapäivä EV-systeemin tuotantoennuste seuraavalle päivälle - Arvioi tulos ja käytäkö sähköä itse? - Vertaa edellisen päivän ennuste ja toteutunut (evaluation) - Ota talteen (osa) datasta uutta koulutusta (model training) varten 	<p>Who is the end-user? What are their objectives? How will they benefit from the ML system? Mention workflow/interfaces.</p> <p>Able to plan electricity consumption to most cost-effective time of the day.</p> <p>Ph1. /Diagnostics. Predict hourly production of the own solar panels (EV) for the next 24h, possible excess production.</p> <p>Ph2. /Decision support. Optimize own electricity bill. Receive recommendation to consume own solar panel production such as <u>schedule actions hourly</u> like dishwasher, charging the car, accordingly, or sell all excess, and schedule own consumption to other times <u>e.g.</u> at night.</p>	<p>Strategy for initial train set & continuous update. Mention collection rate, holdout on production entities, cost/constraints to observe outcomes.</p> <p><u>Miten jatketaan datan keräämistä</u></p> <ul style="list-style-type: none"> - Lukea säännöllisesti inverteristä todellisia tuottoja (Sofar Solar) - Säätiötojen hakeminen automaattisesti (FMI) - Sähkön kulutus (Fortum Tarkka) - Dataa aurinkosähkön tuotannosta Suomessa (Fingrid) 	<p>Where can we get (raw) information on entities and observed outcomes? Mention database tables, API methods, websites to scrape, etc.</p> <p>Features</p> <ul style="list-style-type: none"> - Calendar (<u>i.e.</u> daylight and significant effect on solar panel production) - Weather forecast (sun/cloud/temp) <p>Label</p> <ul style="list-style-type: none"> - EV system electricity production (source: data from inverter) <p>Ph2. Features</p> <ul style="list-style-type: none"> - Nordpool electricity prices - Sähkön kokonaiskulutus kiinteistöissä (source: Fortum app)

IMPACT SIMULATION 	MAKING PREDICTIONS 	BUILDING MODELS 	FEATURES 
<p>Can models be deployed? Which test data to assess performance? Cost/gain values for (in)correct decisions? <u>Fairness constraint?</u></p>	<p>When do we make real-time / batch pred.? Time available for this + featurization + post-processing? Compute target?</p>	<p>How many prod models are needed? When would we update? Time available for this (including featurization and analysis)?</p>	<p>Input representations available at prediction time, extracted from raw data sources.</p> <ul style="list-style-type: none"> - Date - Time of day (hour) - Temperature (celsius) - Sun/cloud (togg.) - Rain forecast (mm) <p>Ph2.</p> <ul style="list-style-type: none"> - Cost of kWh (€) - Hourly consumption in the building (kWh)
	<p>MONITORING</p> <p>Metrics to quantify value creation and measure the ML system's impact in production (on end-users and business)?</p>	<p><u>miten mitata?</u></p> <ul style="list-style-type: none"> - Kuinka lähelle regressio arvioi tuotannon - Mikä vaikutus sähkölaskuun 	

Appendix 2: Data preparation extract – calculation Solar Energy production for training data

```
# Data preparation step1: luetaan ja puhdistetaan Fingridin aurinkotuotantodata vuodelta 2022

# Opetusdata: toistaikseksi ei ole omaa mittausdataa EV-järjestelmästä, joten mallin ensimmäiseen versioon käytetään
# aurinkosähkötuotantodataa kuvaamaan parasta ennustetta kunakin vuoden aikana
# Lähde Fingrid open data https://data.fingrid.fi/open-data-forms/search/fi/?selected\_datasets=241

tiedosto_grid = "Fingrid-aurinko-tammi-marras-2022.csv"

# puhdistetaan Fingrid -data

df_grid = pd.read_csv(tiedosto_grid, delimiter = ",")

df_grid = df_grid.drop(["Lopetusaika UTC", "Alkuaika UTC+02:00", "Lopetusaika UTC+02:00", "Aurinkovoiman tuotantoennuste"])
df_grid['Alkuaika UTC'] = df_grid['Alkuaika UTC'].astype('datetime64[ns]')

# extract hour from the timestamp column to create an time_hour column
df_grid['Year'] = df_grid['Alkuaika UTC'].dt.year
df_grid['Hour'] = df_grid['Alkuaika UTC'].dt.hour

df_grid['Dates'] = df_grid['Alkuaika UTC'].dt.date
df_grid['Dates'] = pd.to_datetime(df_grid['Dates'])

#df_grid.info()
#print(df_grid.describe())
#print("xxxxxxx")

# Poistetaan datasta rivit joissa kokonaiskapasiteetti on selkeästi virheellinen esim. '0'

df_grid.loc[df_grid["Aurinkovoimaennusteessa käytetty kokonaiskapasiteetti "] < 200,
            "Aurinkovoimaennusteessa käytetty kokonaiskapasiteetti "] = 350

df_grid["Aurinkotuotto %"] = df_grid["Aurinkovoiman tuotantoennuste - päivitys tunneittain"] / df_grid["Aurinkovoimaennusteessa käytetty kokonaiskapasiteetti "]
df_grid["production"] = df_grid["Aurinkotuotto %"] * 12
# '12' on teoreettinen max. kW kapasiteetti omalle EV-systeemille,
# jonka perusteella lasketaan odotettu tuotanto omalle järjestelmälle

df_grid = df_grid[df_grid["production"] >= 0]

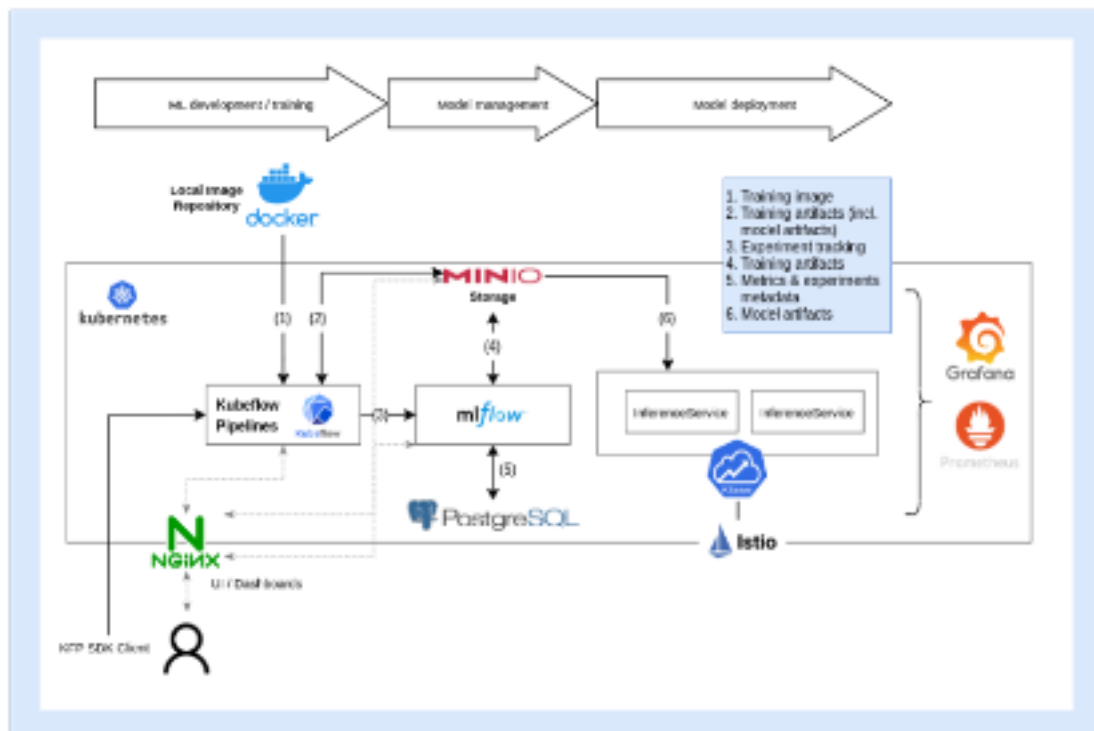
new_dataset1 = df_grid[["Alkuaika UTC", "Dates", "Year", "Hour", "Aurinkotuotto %", "production"]]
print(new_dataset1.head(10))

new_dataset1.to_csv("new_dataset1.csv", index = False)
```

Appendix 3: Example of a CI/CD pipeline for a Machine Learning project

(Mehrdad Saadatmand (RISE, IVVES - D4.4 – Data-driven engineering methods and techniques: final version 2022)

Industrial Machine Learning for Enterprises



Pictures and tables

<i>Figure 1: Extract of the master dataset.....</i>	<i>10</i>
<i>Figure 2: Cross-validation of the LGBM model with some good results</i>	<i>12</i>
<i>Figure 3: Testing - Real data vs. predicted data based on trained model</i>	<i>13</i>
<i>Figure 4: Comparing forecast from the model with the data from the real EV production system.....</i>	<i>16</i>
<i>Figure 5: Final output from the optimization - predicted excess solar energy and recommendation for using the electric appliances.....</i>	<i>17</i>
<i>Table 1: Visualization of the training dataset (Pandas DataFrame).....</i>	<i>13</i>
<i>Table 2: Variable importance of the features.....</i>	<i>15</i>
<i>Table 3: Output from the cost optimization calculation.....</i>	<i>17</i>

Sources

Ansoleaga, Unai Lopez. 2022. *Time Series Forecasting with Supervised Machine Learning*.

<https://towardsdatascience.com/time-series-forecasting-with-machine-learning-b3072a5b44ba>.

Fingrid. 2023. *Fingrid open data*. <https://data.fingrid.fi/en/dataset/>.

Fortum. 2023. *Oma Fortum*. <https://web.fortum.fi/dashboard>.

IML4E project, Harry Souris (Silo AI), Jürgen Großmann (Fraunhofer FOKUS), Johan Himberg (Reaktor).

2022. *Initial MLOps methodology and the architecture of the IML4E framework*. Work package, IML4E project.

Institute, Finnish Meteorological. ei pvm. *The Finnish Meteorological Institute's open data*.

<https://en.ilmatieteenlaitos.fi/open-data>.

Institute, Finnish Meteorological. 2023. *FMI Open Data*. <https://en.ilmatieteenlaitos.fi/open-data>.

IVVES project, subproject for ITEA. 2020. *Quality AI framework*.

<https://learn.ivves.eu/course/view.php?id=13>.

L. E. Lwakatare, I. Crnkovic and J. Bosch. 2020. "DevOps for AI – Challenges in Development of AI-enabled Applications." *International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*.

Mehrdad Saadatmand (RISE, SWE), Niclas Ericsson (RISE, SWE), Yaping Luo (ING, NLD), Tim Soethout (ING, NLD), Juan Leandro (Aunia, ESP), and WP4 partners. 2019. *D4.4 – Data-driven engineering methods and techniques: final version*. ITEA3, IVVES.

—. 2022. *IVVES - D4.4 – Data-driven engineering methods and techniques: final version*.

Treveil, Mark. 2021. *Introducing MLOps, How to Scale Machine Learning in the Enterprise*. O'Reilly.