NTNU
Norwegian University of
Science and Technology

Department of Computer Science

TDT4259 - Applied Data Science

Energy Consumption Forecasting with Machine Learning

Authors:
Ali Hashemi - 583306
Eduardo Iván Flores Peña - 583312
Eirik Drage Steen - 544632
Elias Lerheim Birkeland - 507705
Marcus Remoquin Fuglestad - 544561
Maryam Hematipour - 583307

Autumn 2023

Table of Contents

List of Figures **ii**

List of Tables **iii**

1 Context and Problem Definition **1**

1.1 Aneo 1

1.2 Problem Definition 1

1.3 Broader Suitability within the Energy Domain 2

1.4 Team, Roles, and Responsibilities 2

2 Background **4**

2.1 Objectives 4

2.2 How The Objectives Can Be Resolved With the Use of Data 6

2.3 Problem Approach 7

2.4 Data Strategy - Crisp-DM 8

2.5 Design Thinking 10

3 Method and Analysis **11**

3.1 Dataset Description 11

3.1.1 Features and Attributes 11

3.1.2 Sources 11

3.2 Data Analysis Methods and Tools 12

3.3 Data Preprocessing 15

3.4 Machine Learning Methods 16

3.5 Model Evaluation and Metrics 17

4 Evaluation and Interpretation **19**

4.1 Prediction Results 19

4.2 Error Analysis 21

4.3 Feature Importance 22

4.4 Benchmarking Against a Naïve Forecast 23

4.5 Business Value 25

4.6 Shortcomings 27

4.7 Ensuring the Reliability of the Pipeline 28

5 Deployment and Recommendations **30**

5.1 Deployment Plan . . . 30
5.1.1 Implementation Plan . . . 31
5.2 Relevant Actions Towards Different Stakeholders . . . 32
5.2.1 Aneo Actions . . . 32
5.2.2 Client Actions . . . 33
5.3 Limitations and Improvements . . . 34
5.3.1 Dataset Limitations and Improvements . . . 34
5.3.2 Method Limitations and Improvements . . . 34
5.4 Future Analysis . . . 35

6 Maintenance and Monitoring **36**
6.1 Key Performance Indicators . . . 36
6.2 Monitoring Dashboard . . . 37
6.3 Strategic Risk Management and Contingency Planning: . . . 38
6.4 Lessons Learned and Feedback . . . 38

List of Figures

1 Transforming Business Objectives into Project Objectives . . . 4
2 Business Analytics Leverage Matrix . . . 5
3 Business Model Canvas of Aneo (Aneo 2023) . . . 7
4 The Crisp-DM model (Dalal 2023a) . . . 8
5 STL decomposition applied to the Oslo consumption series for all hours . . . 12
6 STL decomposition applied to the Oslo consumption series for the hours 0 to 300 . 13
7 Correlation plot for Oslo over the entire duration of the dataset . . . 13
8 Autocorrelation of the consumption in Oslo from hour 0 to 300 . . . 14
9 Sliding window cross-validation . . . 18
10 Visualization of the model training, validation, and predictions for Bergen . . . 19
11 Visualization of the 5-day gap and day-ahead forecast for Bergen . . . 20
12 Day-ahead predictions for all Cities . . . 20
13 Error Graph for Each City . . . 21
14 Comparative predictive performance illustrating Stavanger's pronounced improvement in MAE and Oslo's notable enhancement in SD against the baseline model. . 23
15 Error comparison between the actual and baseline models . . . 24
16 Standard deviation comparison between actual and baseline models . . . 24
17 Comparions of day-ahead prediction on April 2 and cross-validation errors . . . 29

18 Google Cloud AI Platform ML Workflow 30
19 Aneo's Energy Forecasting Dashboard 37

List of Tables

1 Team members and their roles 2
2 Correlation Coefficient by City 14
3 Error Metrics in Day-Ahead Predictions for All Cities 21
4 Feature F-scores for Different Cities 22
5 Baseline Model Results 23
6 Error metrics from Cross Validation 29

Introduction

This report describes our results from the Aneo Energy Consumption Forecasting project in the course TDT4259 at the Norwegian University of Science and Technology (NTNU). First, we introduce the problem and describe the background in terms of objectives and strategy. Then, we give a description of the employed prediction and analysis methods and evaluate and interpret the results. Finally, we describe a detailed deployment plan, different actions stakeholders can take based on the results, and how we intend to monitor the implemented pipeline.

1 Context and Problem Definition

1.1 Aneo

Aneo is a Norwegian renewable energy company that produces energy and related analytics. Their areas of focus include solar power systems, energy usage optimization for construction sites, and general energy management. Aneo is headquartered in Trondheim, Norway, and has more than 200 employees across offices in Norway and Sweden. They have established themselves as a leading force within the Nordic renewable energy industry. Their stated goal is to contribute to the global green shift by ensuring that energy is produced as sustainably as possible, and consumed as efficiently as possible (Aneo 2023).

Our project is concerned with their efforts in the energy management area. Within this domain, their provided services are surveillance and monitoring of energy usage, and trading energy assets. Aneo has invested significant resources into building an advanced system for energy analytics and data mining. They have recently ramped up their efforts to incorporate state-of-the-art artificial intelligence algorithms into their systems, and this is where we come in.

1.2 Problem Definition

The problem we are tasked to solve is defined by the following problem statement:

Forecast the next day's hourly energy consumption from 00:00 to 23:00 for Bergen, Helsinki, Oslo, Stavanger, Tromsø, and Trondheim.

This problem statement sets the stage for creating predictive models that can accurately forecast energy consumption for the specified cities on an hourly basis for the subsequent day. We will refer to the forecasts as *Day-ahead predictions*.

Although Aneo is an energy producer, they sell the consumption forecasts to an energy retailer client, and they do not use them to determine the levels of their in-house energy production. Their main benefit is a fixed fee they receive from the client. However, the intended application of the forecasts within the client organization is to guide the levels of energy production. The problem that the forecasts solve for the client is related to the financial risks of over- or underproducing energy that does not meet customer demand. The price at which the residual, unconsumed energy is sold and bought is called the *regulating price* and is known to be highly unstable compared to the standard *spot price* (Norwegian Ministry of Petroleum and Energy 2023). The client seeks to minimize the amount of energy traded at the regulating price, and accurate forecasts are a highly advantageous tool towards this end.

1.3 Broader Suitability within the Energy Domain

The stated goal of the project fits well into the context of the overall renewable energy industry. In recent years, data and AI has begun to play a more important role within the industry. Vast amounts of data are available, and the patterns are often too complex to be extracted by humans. As stated by Sankhyana Consultancy Services: "From renewable energy management to smart grid optimization, data science is revolutionizing the way this industry operates" (Sankhyana 2023).

The main advantages that data science brings to the industry are increased sustainability and efficiency. Companies have used machine learning tools to predict maintenance needs and the probability of equipment failure, to identify risks related to compliance, and for smart grid optimization (Sankhyana 2023). A smart grid is a digital electricity network that uses software and data collected from a variety of sensors in order to match the supply and demand of energy. They are used in cities, and households, and are increasingly being adapted in many countries (IEA 2023). Accurate energy consumption forecasting is relevant for a tool like cost-optimizing smart grids. Our problem statement will generally be highly valuable within the modern, and more technology-driven energy industry, as it is based on making energy production more efficient and sustainable.

1.4 Team, Roles, and Responsibilities

The team for this project consists of a diverse group of students from the Norwegian University of Science and Technology (NTNU), each bringing unique skills and backgrounds to the table. The following table and descriptions provide an overview of each team member's expertise and the roles they will play in the project.

Table 1: Team members and their roles

Member	**Academic Background**
Ali	Ali is a fifth-year master's student in Petroleum Engineering, concentrating on drilling and wells. He brings a strong interest in cybernetics and automation to the project, promising to enhance report writing and idea generation with his technical expertise.
Eduardo	With a background in petroleum engineering, Eduardo applies his technical knowledge to the project's data analytical aspects. His skill set will be instrumental in gathering information and exploring advanced analytical techniques.
Eirik	As a fourth-year student in Industrial Economics and Technology Management at NTNU, Eirik majors in Computer Science with a specialization in Optimization and Artificial Intelligence. He offers his coding skills to drive the project's technical development.
Elias	Elias, studying Industrial Economics and Technology Management with dual specialization in Computer Science and Finance, contributes his AI expertise and financial acumen to the technical and strategic facets of the project.
Marcus	Marcus's academic journey in Industrial Economics and Technology Management, combined with his technological expertise in Computer Science and a focus on Finance, equips him with a unique blend of skills crucial for the project's financial and technical aspects.
Maryam	Maryam, a second-year master's student in Petroleum Engineering with a specialization in Reservoir Engineering, plays a pivotal role in her data science group project. She excels in writing and clearly presenting project goals while tackling technical challenges with an innovative mindset.

Roles and Responsibilities:
In the spirit of collaborative learning and comprehensive skill development, the team has decided to approach the project with fluid roles and responsibilities. Each member is encouraged to engage in various stages of the project, from data analysis to report writing. This approach allows team members to expand their knowledge and expertise in different areas of the project while ensuring that everyone has a comprehensive understanding of the entire process.

- Ali: Primarily focused on report writing and conceptual development, ensuring that our findings are communicated with clarity and impact.
- Eduardo: Spearheading the data visualization efforts, transforming complex datasets into clear, insightful graphical representations.
- Eirik: Tasked with data processing and model development, enhancing the robustness and accuracy of our prediction pipeline.
- Elias: Concentrating on model development and testing, ensuring our models are both accurate and reliable.
- Marcus: Overseeing the deployment of models and the integration of our analytical insights into business strategy planning.
- Maryam: Leading the market analysis and financial impact assessment, bridging the gap between our technical findings and their business implications.

Collaboration and Communication:
The team will employ a combination of synchronous and asynchronous communication tools to ensure effective collaboration. Regular meetings will be held to discuss progress, address challenges, and plan next steps. Tools like Slack, Teams, and Zoom will be used for daily communication, task management, and documentation. This structure aims to foster a collaborative environment while respecting individual schedules and work preferences.

2 Background

2.1 Objectives

Business Objectives:

The business objective was summarized in the **Problem Statement** in section 1.2 (Problem Definition). Based on the team's understanding of the industry context and of the user's need, the problem definition can be further broken down into the following business objectives:

1. Deliver precise hourly energy consumption forecasts for the following day to aid in operational planning.
2. Provide tailored day-ahead energy consumption forecasts for distinct smart grid systems.
3. Craft a machine learning model with adaptability for effortless application to Aneo's growing smart grid networks.
4. Establish an *Energy Forecasting Dashboard* that displays energy consumption forecasts alongside key business performance indicators beneficial for Aneo's strategic decisions.
5. Arrive at a model performant enough that a client who uses it directly to choose energy production levels (produces exactly the forecasted amount) does not need to trade more than x MW of energy at the regulating price hourly. This goal is motivated by the client business's desire to minimize the amount of energy traded at this price. The exact value of x needs to be determined by energy experts at Aneo. We describe potential values for x in section 4.

From Business Objectives to Technical Project Objectives:

In order to convert the business objectives into project objectives, we have to get a clearer understanding of the data available and its constraints.

- The 5-Day Gap Constraint: A notable limitation is the absence of both consumption and temperature data for the five days. Figure 1 shows 5-day gap $[Day_{t-5}, Day_t]$.
- 6-Day Predictions: To overcome the 5-day gap's data limitation, our approach is to forecast a 6-day period. Figure 1 shows 6-day predictions $[Day_{t-5}, Day_{t+1}]$.

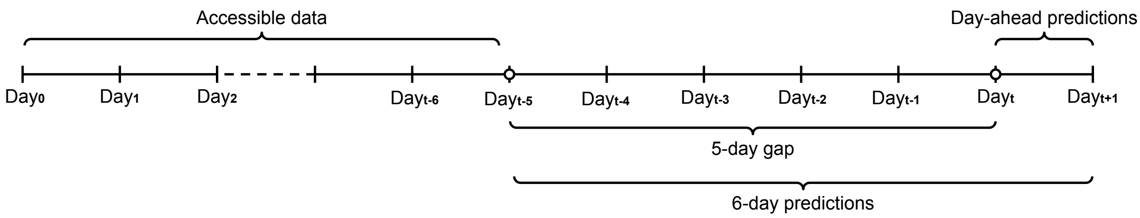

Figure 1: Transforming Business Objectives into Project Objectives

Further analysis of the accessible data and the 5-day gap constraint indicates that the solution requires forecasting a 6-day period. Only then can we attain the day-head predictions needed to fulfill the business objectives.

Technical Project Objectives:

1. Create a machine learning model that predicts 6 days forward, thus overcoming the 5-day gap constraint, and solving the main business objective of forecasting day-ahead predictions.
2. The model's predictions should have a MAPE (mean absolute percentage error, as described in section 3) for the day-ahead predictions lower than 15%. This value was chosen based on qualitative observations that the model gave satisfactory predictions when the MAPE was less than 15% (however, our qualitative opinions may differ from those of an energy professional. For this reason, we formulate the next objective to give a more absolute standard by which to judge the model).
3. The model predictions should have an MAE (mean absolute error, as described in section 3) below the max threshold of x MW that can be traded at the regulating price (from business objective 5). The MAE represents the expected amount of energy in unit MW that the client will need to trade at the regulating price hourly. This goal represents a technical translation of business objective 5.
4. Create a machine learning model that creates day-ahead energy predictions with higher accuracy (lower error) and higher consistency (lower standard deviation) than a baseline model.

Project Success Criteria

Utilizing the Business Analytics Leverage matrix, we've strategically formulated project objectives that balance feasibility and high value for Aneo, ensuring solutions are both implementable and impactful.

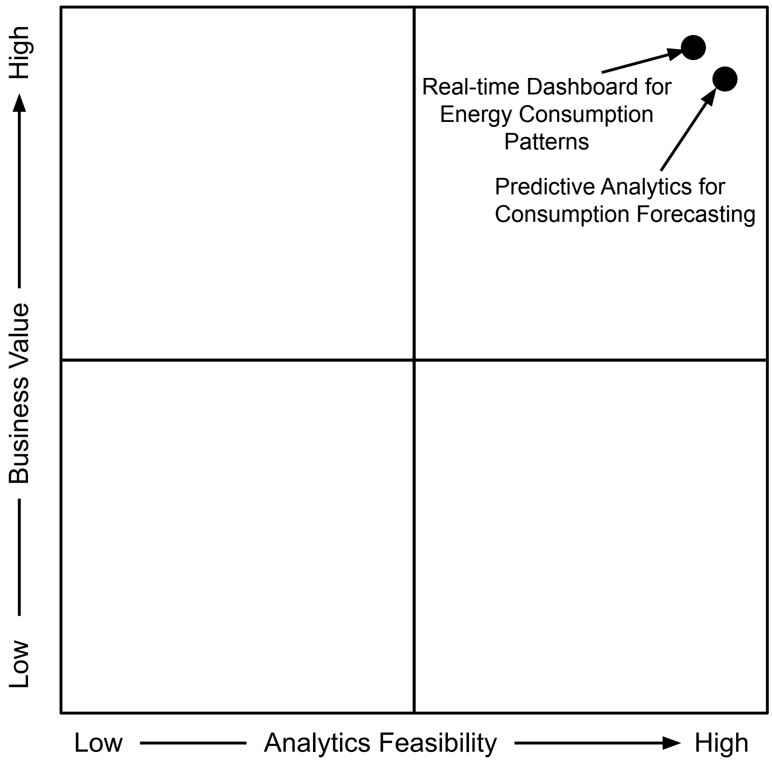

Figure 2: Business Analytics Leverage Matrix

2.2 How The Objectives Can Be Resolved With the Use of Data

Results from past papers and research projects underline that data-driven prediction methods can lead to solid results for time series forecasting. The article, "A Multi-task Learning Approach to Short-Term Load-Forecasting for Multiple Energy Loads in an Educational Building," published by the IEEE International Conference, serves as a notable example of previous research in the field of predicting future energy consumption (Khalil 2022).

In this research endeavor, a dataset similar to the one provided by Aneo was employed. This dataset contained input data pertaining to energy consumption within an educational building, specifically encompassing lighting, the Air-Handling-Unit (AHU), and Domestic Hot Water (DHW), with a particular focus on energy utilization within specific smart grids. Additionally, the study incorporated Exogenous Features, such as outdoor temperature, which were also accessible from Aneo. Importantly, the data was collected at an hourly granularity, aligning with the temporal resolution of the data offered by Aneo (Khalil 2022).

In terms of methodology, the study employed time series Long Short-Term Memory (LSTM) models, a specialized class of Recurrent Neural Networks (RNNs) (Khalil 2022).

In summary, this study mirrors our dataset and shares the common objective of forecasting future energy consumption. Consequently, it reaffirms that our business and project objectives can be effectively addressed through the utilization of data.

2.3 Problem Approach

In addressing the objectives, our approach embraces the Business Analytics Model (BAM). We followed Hindle and Vidgen's four-step BAM model: (Giles A. Hindle 2016)

1. Problem situation structuring: This stage involves identifying and defining the business problem. It includes understanding the context, stakeholders, and objectives of the project (Giles A. Hindle 2016).
2. Business model mapping: This stage involves creating Business Model Canvas (BMC). This helps to identify the key drivers of the business and the areas where analytics can add value (Giles A. Hindle 2016).
3. Analytics leverage analysis: This stage involves creating a Business Analytics Leverage Matrix identifying the analytics techniques to address the business problem. (Giles A. Hindle 2016).
4. Analytics implementation: This stage involves implementing the analytics solution and monitoring its performance (Giles A. Hindle 2016).

The Business Analytics Model as seen in our project:

- **BAM STEP 1:** In sections 1.1 Aneo, 1.2 Problem Definition and 1.3 Broader Suitability within the Energy Domain, we collected all the useful information from Aneo. From this data and understanding of the user, we have defined the clear objectives in section 2.1 Objectives, also illustrated in Figure 1.
- **BAM STEP 2:** Figure 3 illustrates the Business Model Canvas (BMC) of Aneo.
- **BAM STEP 3:** We created a Business Analytics Leverage Matrix, Figure 2, to get an overview of the user's need together in solution feasibility.
- **BAM STEP 4:** Section 3 through 6 of our report explains the implementation and monitoring of the solution.

<table>
<tr><th>Key Partners</th><th>Value Proposition</th><th>Customer relationship</th><th>Customer segments</th><th>Key activities</th></tr>
<tr>
<td rowspan="3">Partners:
-Trønderenergi, HitechVision

Energy Suppliers:
-Bessakerfjellet vindpark
-Geitfjellet vindpark
-Skomakerfjellet vindkraftverk
-Valsneset vindpark
-Frøya vindkraftverk
-Kvenndalsfjellet vindpark
-Roan vindpark
-Ytre Vikna vindpark
-Stokkfjellet vindpark
-Sørmarkfjellet vindpark
-Hundhammerfjellet vindpark
-Grimsås vindpark, Sverige
-Brännliden vindpark, Sverige</td>
<td>Aneo's services:
1. Electric car charging
2. Solar energy
3. Electrification of construction site
4. Industrial heat pumps
5. Energy optimization for grocery stores
6. Energy management</td>
<td>-Comprehensive service provision including installation, operation, and maintenance.
-Continuous support and performance optimization.
-Fixed, predictable billing under Energy as a Service agreements.
-24/7 monitoring and management of energy systems.</td>
<td>-Power production facility owners.
-Municipalities and landowners in renewable energy sites.
-Businesses seeking e nergy efficiency solutions.
-Sectors requiring electric vehicle charging infrastructure.</td>
<td>-Energy Management
-Operation of Power Facilities
-Investment in Renewable Energy</td>
</tr>
<tr><th>Channels</th><th>Key resources</th><th>Cost structure</th><th>Revenue streams</th></tr>
<tr>
<td>-Direct Service Plattform
-24/7 Monitoring
-Operations and Value Maximization Support
-Renewable Energy Site Development</td>
<td>-Technical Expertise
-Heat Pump Technology
-Engineering and Construction Partners
-Capital for Investment
-Energy Management Technology</td>
<td>-Operational expenses for facility management.
-Investments in renewables like wind and solar.
-R&D for heat pump technology.
-Salaries for staff and operational personnel.
-Costs for engineering and construction partnerships.
-Maintenance services for their EaaS offering.</td>
<td>-Fixed monthly payments for Energy as a Service contracts.
-Energy production and trading operations.
-Investments in renewable energy projects.
-Engineering and consulting services.</td>
</tr>
</table>

Figure 3: Business Model Canvas of Aneo (Aneo 2023)

2.4 Data Strategy - Crisp-DM

"CRISP-DM (Cross-Industry Standard Process for Data Mining) is a widely-used data mining process model that describes commonly-used approaches that expert data miners use to tackle problems. Its intention is to provide a structured approach to planning a data mining project and to help ensure that the project stays on track." (Saltz 2021) Our data strategy aligns with the CRISP-DM model. Although untraditional, we have chosen the CRISP-DM format suggested by our professor, with the difference being the addition of a seventh step.

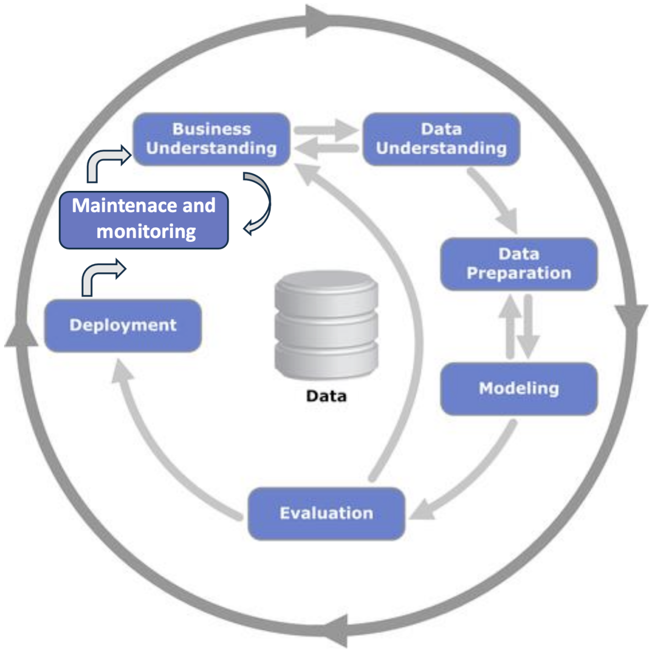

Figure 4: The Crisp-DM model (Dalal 2023a)

1. **Business Understanding**

"The Business Understanding phase focuses on understanding the project objectives and requirements from a business perspective." (Saltz 2021). It involves dialogues with stakeholders to clarify the business problem, define the project's goals, and assess the resources available. The outcome of this phase is a project plan with defined business objectives and data mining goals that align with the business's strategic direction.

Detailed in section 1.2 Problem Definition and 2.1 Objectives, this stage aligns the data strategy with Aneo's goals, ensuring our approach serves the company's strategic direction effectively.

2. **Data Understanding**

"The Data Understanding phase focuses on identifying, collecting, and analyzing data sets that can help the project."(Saltz 2021). It's a stage where exploratory data analysis is conducted to become familiar with the data, identify data quality problems, or discover initial insights into the data. It involves tasks such as collecting data, describing data, exploring data, and verifying data quality.

Outlined in section 3.1 Dataset Description and 3.2 Data Analysis Methods and Tools, this phase involves a deep dive into Aneo's data, setting the groundwork for insightful analytics and data-driven decisions.

3. Data Preparation

Often the most time-consuming phase, data preparation is about transforming raw data into a final dataset that can be used in modeling. "This phase includes tasks such as cleaning data, integrating data from multiple sources, and deriving new attributes that will be helpful for modeling."(Saltz 2021).

Section 3.3 Data Preprocessing describes the rigorous process of refining data, pivotal for the accuracy and efficacy of our subsequent modeling phase.

4. Modeling

"The Modeling phase involves selecting and applying various modeling techniques to the prepared data."(Saltz 2021). Depending on the situation, multiple techniques may be appropriate and require testing. Modeling involves selecting suitable models, configuring their parameters, and then training them on the prepared dataset. Techniques may have specific requirements on the form of data, therefore, stepping back to the data preparation phase is sometimes needed.

Explored in section 3.4 Machine Learning Methods, section 3.5 Model Evaluation and Metrics and section 4 Evaluation and Interpretation, this critical phase discusses our adaptive modeling approach, emphasizing iterative refinement and alignment with Aneo's predictive needs.

5. Evaluation

"The Evaluation phase looks more broadly at how the model meets the business and what to do next."(Saltz 2021). The model needs to be robustly evaluated to ensure it meets the business's requirements and expectations. This stage involves assessing the model, and the steps executed to construct the model, to be certain it properly achieves the business objectives. A key task is to determine if there is some important business issue that has not been sufficiently considered. At the end of this phase, a decision on the use of the data mining results should be reached.

In section 4.5 Business Value and **?? ??**, we assess our models' performance, ensuring they meet Aneo's business objectives and deliver actionable insights.

6. Deployment

The final phase of the traditional CRISP-DM process involves deploying the data mining solution into the business environment. "The Deployment phase focuses on developing and documenting a plan for deploying the model."(Saltz 2021). This could mean simply generating a report or as complex as implementing a repeatable data mining process across the organization.

Section 5 Deployment and Recommendations discusses the integration of our models into Aneo's operations, focusing on the practical application and scalability of our data mining solutions.

7. Maintenance and Monitoring

This additional section is proposed by our professor to make the CRISP-DM model to ensure further continued success for the client. Post-deployment, this phase ensures the model's continued relevance and accuracy in the dynamic business environment. We will establish processes for regular reviews, updates, and model recalibration as necessary. This continuous improvement cycle will be articulated in the 'Monitoring and Maintenance' section.

The section 6 Maintenance and Monitoring highlight the necessity of continuous model assessment and adaptation to maintain alignment with the evolving business landscape.

2.5 Design Thinking

"Design thinking makes it possible to not rush into the product design but to dedicate upstream time to generate a maximum number of ideas against the identified needs and to propose the elements upon which to build a response element to users." (Combelles et al. 2020). At its core, design thinking involves five phases:

1. **Empathize:**

This first stage of Design Thinking is about understanding the needs, motivations, and emotions of the people for whom you're designing. It involves observing, engaging, and empathizing with people to understand their experiences and motivations, as well as gaining a deeper personal understanding of the issues involved.

In section 1.1 Aneo and 1.3 Broader Suitability within the Energy Domain, we described Aneo's context, exploring their position in the energy market. This deep dive helped us form a clear picture of the problem from their perspective.

2. **Define:**

In Define phase, information gathered from the Empathize stage is used to to define the core problems of the user. This stage involves crafting a user-centric problem statement.

In section 1.2 Problem Definition and 2.1 Business Objectives:, we synthesized our insights from the empathize phase to articulate Aneo's challenges and objectives. These goals were crafted to directly address the user needs we identified.

3. **Ideate:**

With a solid grasp of the users' needs and problems, the Ideate stage is about generating ideas. The Ideate stage is about brainstorming, looking for alternative ways of viewing the problem, and identifying innovative solutions to the problem statement you've created.

In section 2.1 Objectives, the team worked on converting them into project objectives. We engaged in creative sessions, mapping out how business objectives can be solved using data science. The result was a set of project objectives tailored to the methods we chose.

4. **Prototype:**

The Prototype stage involves producing a number of inexpensive, scaled-down versions of the product or specific features found within the product. This step is iterative; prototypes are shared and tested within the team itself, with other departments, or on a small group of people outside the team.

Early in the implementation process, the team underwent a phase of testing out several solutions. This was done to identify the solution that was most optimal to meet the client's needs, yet also for us to implement within the project time frame.

5. **Test:**

Testing is a stage of is the final stage of design thinking involving refinements and improvements. Feedback from the users is critically important during this phase, as it can reveal new insights or issues with the current design.

Although we were unable to showcase our solution to the users, we iteratively reassessed our solution based on the business objectives and project objectives. Here we tried to perfect our model to suit the client's needs.

3 Method and Analysis

This section details the methods and analysis techniques used in our study, including data visualization, dataset description, data preprocessing, analysis methods, and machine learning techniques.

3.1 Dataset Description

3.1.1 Features and Attributes

The dataset contains four features: time, location, energy consumption, and temperature. As the problem statement is to predict the city-wise consumption, we are left with the input features time, location, and temperature.

Time
The time feature provides the point in time of a single observation in the format YYYY-MM-DD-HH. All time points are registered at the beginning of an hour. There are no missing time values. The first date is 2022-04-07, and the last is 2023-04-02, which adds up to 49494 rows of data in total. As for the attribute type, although time itself is a continuous value, it is a discrete attribute in this dataset. This is due to the discrete nature of the measurements, where each observation pertains to a specific hour.

Temperature
The temperature is a floating point number describing the Celsius value of the temperature at a specific time. Each temperature value is the the hourly forecasted temperature for the corresponding location. It can take both negative and positive values. As for the attribute type, it is continuous since it can take on an infinite set of values.

Location
The location is a string value that tells the city where the observation took place. There are six cities represented in the dataset: Bergen, Helsingfors, Oslo, Stavanger, Tromsø, and Trondheim. The observations for every location except Helsingfors contain 8641 rows. The observations for Helsingfors start later in time and contain only 6289 rows. The attribute type for location is categorical since the feature represents a place name.

Consumption
This is the prediction target, and it is a non-negative floating point number in the unit megawatt. Since it can take any real non-negative value, it is a continuous attribute. It represents the average hourly consumption for a given hour and city.

3.1.2 Sources

The source of the dataset is an energy retailer, who is a client of Aneo. For privacy reasons, the identity of the retailer has not been revealed.

Regarding consumption, we know that each data point comes from the observed consumption at the start of an hour in a specific city. However, it does not represent the consumption of the whole city. The values represent the sum of consumption of all customers of the retailer in a given city. As not all electricity users are customers of the retailer, the consumption values can be looked at as a subset of the consumption for the whole city. As mentioned, we do not receive the consumption data until five days after the last observation.

The source of the temperature is an externally gathered forecast (presumably by a reputable forecaster such as Yr, but the identity of the source is unknown). We gain access to temperature information one day before the prediction target date, and we can therefore use it as a feature in our model when doing day-ahead consumption forecasting.

3.2 Data Analysis Methods and Tools

This section describes the methods and tools we used to analyze the ANEO dataset. We explain how they are appropriate with respect to the problem statement. The focus of the section is how we determined what characteristics in the data would be most helpful for the problem statement prediction task. Additionally, we used the results of the analysis for feature selection, by determining which features would be the most useful for our task. We provide descriptive statistics and corresponding visualizations showing results from the analysis.

Seasonality, Trend, and the STL Decomposition

The analysis of seasonality and trends is a central tool in any time series analysis scenario. Seasonality is a pattern repeating at a specific frequency with a corresponding period. The trend is a long-term increase or decreases in time series values, that develops while the seasonality remains the same. Seasonality and trend are highly important features in a time series dataset when it comes to predicting future values based on the past. If there is a trend, the general, long-term change in the data is indicative of the change it will go through in the future. If there is seasonality, we can estimate future values based on the specific time since a past observation (Fernandez 2023).

In order to analyze seasonality and trend, we leverage the method of STL decomposition using LOESS (Cleveland et al 1990). STL decomposes a time series into three components, one for seasonality, one for trend, and one for residuals. Summing all three components will reproduce the original time series. It is a standard method for time series analysis and is widely used due to its robustness. For a detailed explanation, we refer to (Cleveland et al 1990). Figure 5 and 6 show STL plots of the Aneo dataset. While 5 analyses the entire consumption for Oslo, 6 analyses the consumption for the first 300 hours.

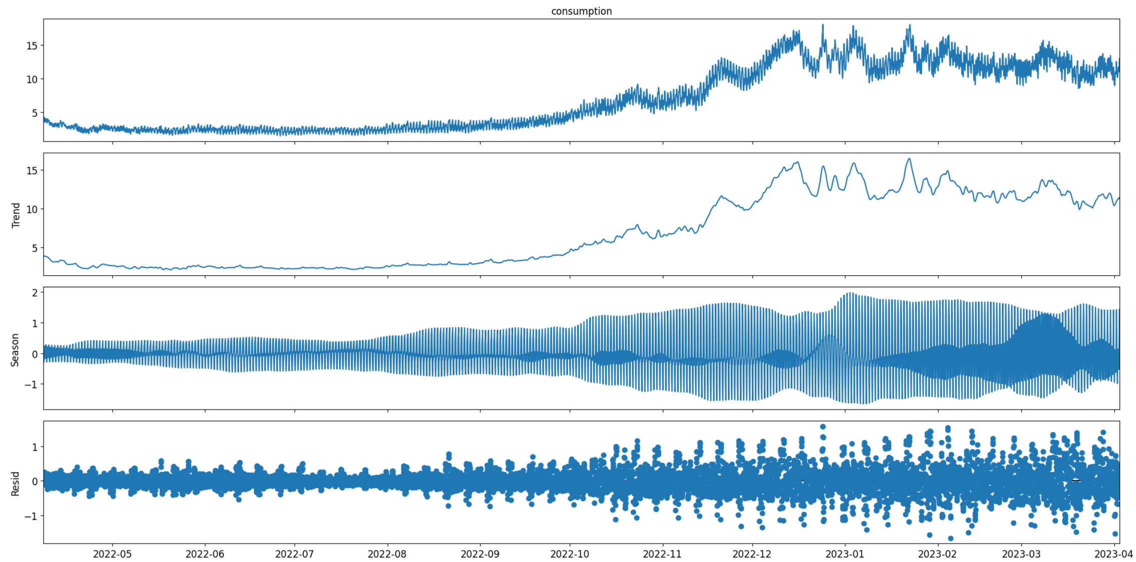

Figure 5: STL decomposition applied to the Oslo consumption series for all hours

We observe that there is a strong presence of trend and seasonality in the dataset. The trend is flat at first but turns upwards when the winter months begin. This makes sense, as more energy is required when the cold arrives. We see that there is seasonality, but the frequency is too high to analyze qualitatively in figure 5. Looking at the STL decomposition in figure 6 gives a more illustrative picture:

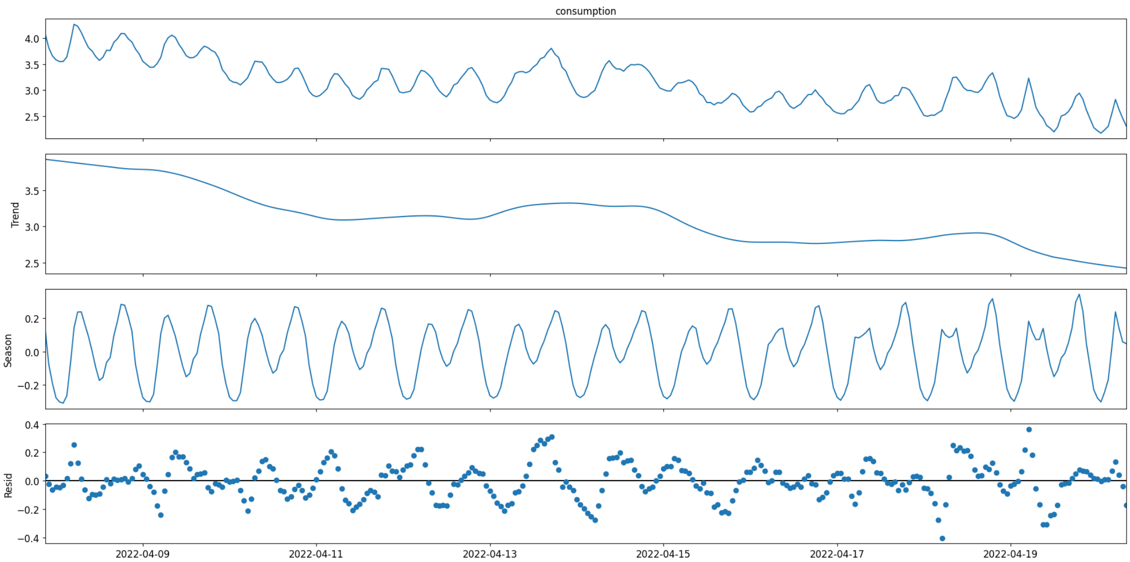

Figure 6: STL decomposition applied to the Oslo consumption series for the hours 0 to 300

From figure 6, we observe that there is a strong presence of seasonality. This makes intuitive sense, as energy consumption should trace energy demand, and energy demand depends on human activity. We observe that the seasonality is intra-daily, which makes sense since less energy is required during the night. Further based on intuition, we also hypothesized that the energy consumption would have a strong inverse correlation with temperature. This is because we require more energy for heating when the temperature drops (and vice versa). The correlation plot in figure 7 supports our hypothesis.

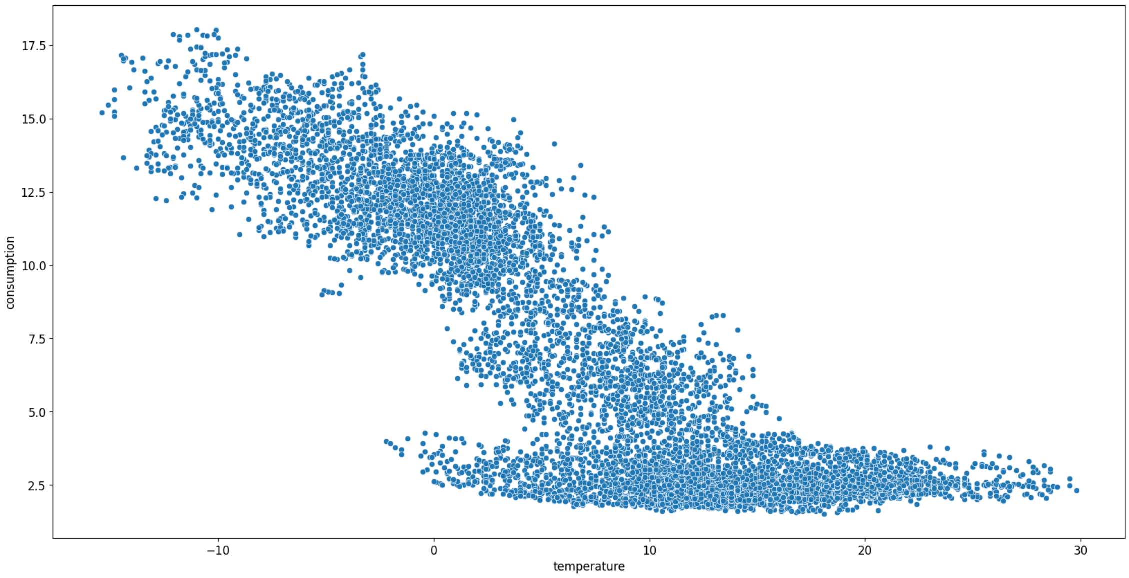

Figure 7: Correlation plot for Oslo over the entire duration of the dataset

To get a sense of the city-wise consumption dependence on temperature, here is a table showing the correlation coefficient calculated for each city over the entire duration of the dataset:

Table 2: Correlation Coefficient by City

City	Bergen	Helsingfors	Oslo	Stavanger	Tromsø	Trondheim
Correlation Coefficient	-0.78	-0.57	-0.86	-0.78	-0.86	-0.84

We see that all cities exhibit a strong negative correlation with the temperature. This analysis tells us that temperature is a highly relevant feature for the problem statement prediction task.

Autocorrelation

Autocorrelation is another widely used metric related to the analysis of seasonality and trends in a time series. Informally, autocorrelation is the correlation calculated for a time-dependent random variable between itself at different points in time. In mathematical terms, if X_t is the random variable in question at time t, and $\mu_t = E(X_t)$:

$$\gamma_X(t+h, t) = Cov(X_{t+h}, X_t) = E((X_{t+h} - \mu_{t+h})(X_t - \mu_t)) \tag{1}$$

$$\rho_X(h) = Corr(X_{t+h}, X_t) = \frac{\gamma_X(h)}{\gamma_X(0)} \tag{2}$$

The metric is relevant to our analysis since it indicates the predictive influence of past values on future values. It tells whether the value at time t tells you something about the value at time $t+h$. Specifically, for data with a strong trend, the autocorrelation between values at close time points will be positively large since the values will be close in size. If the data has seasonality, we will also observe a seasonality in the autocorrelation, since it will be highly dependent on the specific lag value h. Concretely, the values will be higher at multiples of the seasonal frequency (Zimmer et al. 2021).

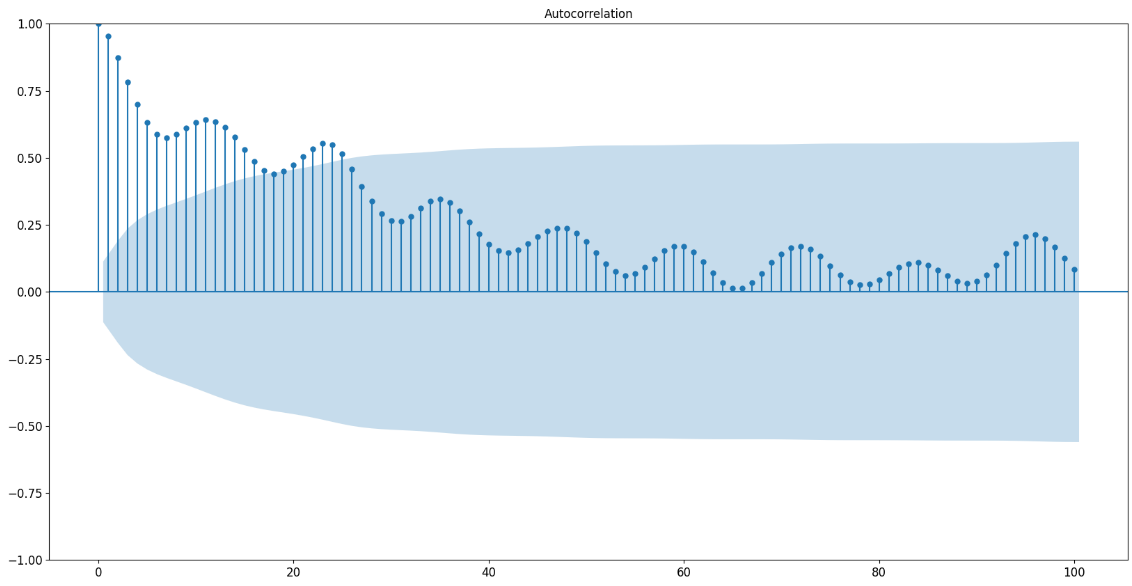

Figure 8: Autocorrelation of the consumption in Oslo from hour 0 to 300

From figure 8, we observe that the autocorrelation decreases for time points farther in the past and that the values are higher at multiples of the seasonal frequency. This shows that both trend and seasonality are present in the data. Furthermore, since the analysis shows that the consumption values correlate highly with past values, it is likely a good idea to leverage lag features. These features will let the model learn the correlation patterns with the past data. The analysis also tells us how far in the past we should go when extracting the lag features. The blue area shows where the correlation is statistically significant (where it is high enough that noise in the data is an unlikely cause). The plot shows that the autocorrelation loses significance after about 20 time points. We will use this information to guide our feature engineering, as described in the next subsection.

For all plots in this subsection, we decided to only include examples for a single city within a fixed time interval. We could show plots we made for other cities and intervals, but we observed very similar patterns in these plots, and so decided to omit them in the report to avoid redundancy.

3.3 Data Preprocessing

Feature Engineering

In order to maximize the predictive information contained in the time value, we decomposed it into five features: day of week, day of year, month, hour, and year. We did this to facilitate the detection of seasonality (as the presence of seasonality was proved in the analysis). By providing the time information in this more granular form, the model will be able to detect seasonality that is hourly, daily, weekly, monthly, or yearly.

In the analysis from the previous section, we proved that there exists a statistically significant autocorrelation pattern in the data, and deemed it a good idea to incorporate lag features. However, as the analysis also showed that the most powerful lag features would be those computed on the base of the past 20 hours, the 5-day data gap makes it problematic to implement (useful) lag features. We experimented with lag features from 6 days in the past or farther, but it did not perform better than a model trained without them. We consequently decided not to implement lag features.

Data Splitting: Input and Output

Due to the 5-day delay in data availability, as well as the problem statement goal of day-ahead forecasting, we train the model to predict the consumption for six days in the future ($6 * 24$ hourly time points). We do this since we first need to close the five-day data gap, and then predict the next day after the time point at the end of the data gap (which represents the present time). We extract the last 24 hours of the predicted data as the final output we give to the customer. We treat the size of the input window as a hyperparameter, as further explained in section 3.5.

Further, we train separate models for the different cities, as the problem statement dictates that we want to predict the consumption per city, and not in aggregation.

Preprocessing

We considered preprocessing the data through the standard mean-centering and normalization procedure, but we ended up selecting a model invariant to such transformations and consequently did not spend development time on this. The model selection is described in detail in the next subsection.

Data Cleaning

As the provided dataset was remarkably clean, we did not need to conduct data cleaning. After analysis, we concluded that there were no missing values or noteworthy inconsistencies. The solid model performance as described in the next section underlines that data cleaning was not necessary.

3.4 Machine Learning Methods

Model Choice and Motivation

We used XGBoost regression as our forecasting model. The reasons are that XGBoost is considered a robust and well-performing method (Chen and Guestrin 2016), and it is quick to implement (with the convenient XGBoost library). We initially experimented with a simple baseline model that creates predictions based on the direct past. A comparison between the methods, as well as an empirical justification for our selection of XGBoost, is provided in the results section.

Baseline: Naive Forecasting

A naive forecast approach for predicting the next day's energy consumption would simply replicate the most recent day's data available. In our context, the baseline model predicts based on the consumption data from six days prior, embodying the most straightforward prediction technique. Concretely, if $\hat{y_i}$ is the prediction and y_i is the consumption at time i:

$$\hat{y_i} = y_{i-6*24} \tag{3}$$

The model is evaluated in the results section.

XGBoost

XGBoost is an ensemble model that consists of regularized decision trees. Many different learners are trained, and each learner learns from the mistakes of the last. Concretely, the gradient of the loss function is computed for the current ensemble of learners, and a new learner is trained to minimize the gradient. In the case of XGBoost, the learners are decision trees.

For brevity but completeness, we will include a condensed mathematical description of XGBoost. The model prediction is calculated by summing the predictions from K additive functions (K decision trees):

$$\hat{y_i} = \sum_{k=1}^{K} f_k(x_i) \tag{4}$$

$\hat{y_i}$ is the prediction, and f_k belongs to the set of regression trees. The function f_k aggregates (usually sums) the predictions from the decision tree leaves to arrive at the final output. The optimization objective is the following regularized loss function:

$$L(\phi) = \sum_i l(\hat{y_i}, y_i) + \sum_k \Omega(f_k) \tag{5}$$

$$\Omega(f_k) = \gamma T + (1/2)\lambda||w||^2 \tag{6}$$

l is a convex and differentiable loss function. In our model, we selected the squared error function. T is the number of leaves in tree k, w is the tree weights, and everything else is regularization hyperparameters. As we can see, the objective regularizes the model by penalizing weights of high magnitude (large weights induce sensitivity to input perturbations), and by rewarding a few leaves in the tree. Thus the model complexity and variance are reduced.

The following equation describes the gradient tree-boosting mechanism:

$$L^{(t)} = \sum_{i}^{n} l(\hat{y_i}^{(t-1)} + f_t(x_i), y_i) + \Omega(f_k) \tag{7}$$

This is a greedy algorithm, where we add the learner that achieves minimum loss according to the objective in equation 5. For a more detailed treatment, see the original paper (Chen and Guestrin 2016).

In summary, many weak learners are combined to create a strong learner, and the model is trained to be generalizable. Consequently, the model is well suited for the problem statement task, as it catches complex patterns (through the decision trees), while it maintains generalisability (which we want across time intervals) due to the regularisation and boosting.

3.5 Model Evaluation and Metrics

Metrics

We leveraged standard time series forecasting metrics to evaluate the efficacy of the model. We used the mean absolute error, root mean squared error, and the mean absolute percentage error. Let y be the true value i, $\hat{y_i}$ prediction i, and n the size of the data sample. The equations 8, 9, and 10 define the MAE, RMSE, and the MAPE respectively.

$$MAE = \frac{1}{n} \sum |y_i - \hat{y}| \tag{8}$$

$$RMSE = \sqrt{\frac{1}{n} \sum (y_i - \hat{y})^2} \tag{9}$$

$$MAPE = \frac{1}{n} \sum \frac{|y_i - \hat{y}|}{y_i} \tag{10}$$

The MAE tells us the magnitude of the expected error. The RMSE gives a similar value, and we can expect it to be large when the MAE is large. The difference is that the RMSE is based on squared errors, and consequently penalizes larger errors relatively higher than the MAE. We use both metrics as the MAE gives us valuable information about the average error, while the RMSE compliments it by telling whether there are some unusually high error outliers. If the predictions are accurate on average, but some time points have exceptionally high errors, the RMSE will indicate this and help with further analysis. Our usage of the third metric is motivated by the fact that neither the MAE nor RMSE provides an intuitively interpretable value. While they represent unbounded magnitudes, the MAPE provides the average percentage error. The percentage ranges from 0-1 and gives us a reference interval that lets us intuitively quantify the model performance (as 0 is perfect and 1 is completely wrong).

Hyperparameter Tuning

Since our model may be used at any time, it is important that the predictions remain satisfactory regardless of the specific time interval upon which it runs inference. It is possible that certain intervals are easier to forecast than others, and to avoid this bias in our evaluation, we use sliding window cross-validation. Sliding window cross-validation entails training the model and computing metrics for different time intervals. The input and output data size stay constant, while the start and end time change.

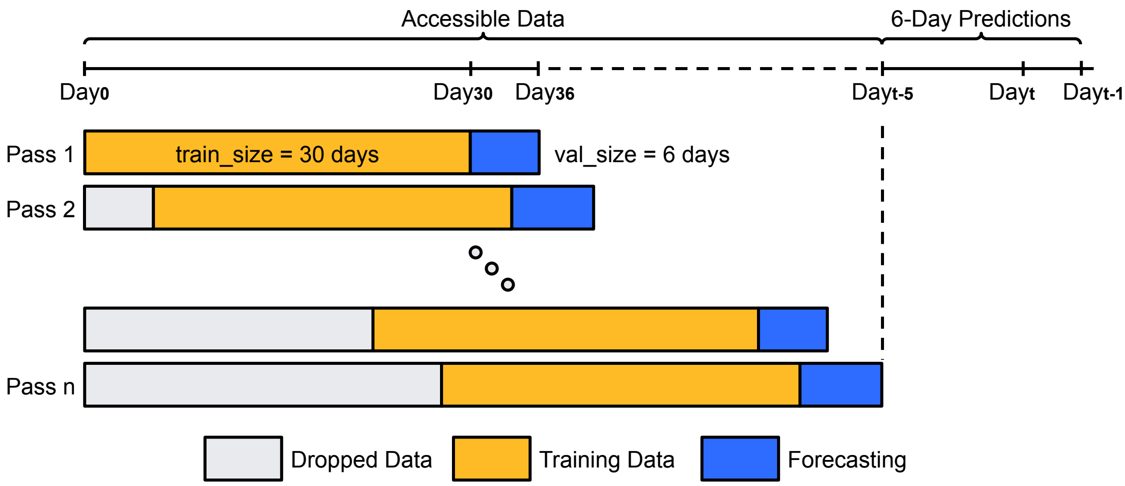

Figure 9: Sliding window cross-validation

We implemented sliding window cross-validation bypassing the train input size n as a parameter (while the output size $= 6 * 24$ remains constant, as per the definition of 6-day Predictions in 2.1Objectives). The model was then separate trained for the intervals $[0, n], [n, 2n], [2n, 3n]$, etc. The corresponding output intervals would be $[n, n + 6 * 24], [2n, 2n + 6 * 24]$, etc. Metrics were computed for each model and corresponding interval and then averaged to quantify the complete model performance. The RMSE was the target metric used for hyperparameter tuning. The learning rate, decision tree max depth, and input data size were the parameters we selected to tune. We chose to tune the learning rate and max depth as they are known to be the most important hyperparameters for boosting algorithms, and the input data size since we observed from experimentation that it had a substantial effect on the model performance.

Train, test, validation split

We split our data into train, test, and validation sets according to best practices. We train the model on data in the window $[t_i, t_i + n]$, validate the hyperparameters on the interval $[t_i + n, t_i + n + 6 * 24]$, and finally compute test metrics on the completely unseen data in the interval $[t_i + n + 6 * 24, t_i + n + 2 * 6 * 24]$. The value of t_i is determined by what interval we are currently using for the sliding window cross-validation mechanism. t_i is shifted for the different intervals such that $t_i = (n + 6 * 24) * i$. Consequently, if N is the total length of the time series, the sliding window operation is applied to $\lfloor \frac{N-2*6*24}{n} \rfloor$ intervals in total. Note that there is no overlap in the train and validation sets between the different intervals, but the train set of the next interval starts where the test set of the current interval ends. As new models are trained separately on different intervals, we could overlap more data. We chose not to do so to reduce the running time of the hyperparameter tuning script. As separate models are trained for each interval, and the test set is never seen during training or hyperparameter tuning, we avoid leakage of future data to the model. We can therefore be confident that the computed metrics represent how the model would perform on truly unseen future data (or, as confident as we can be, as an inherent limitation of time series forecasting is that there is no guarantee that the future will behave similarly to the past.).

4 Evaluation and Interpretation

4.1 Prediction Results

As described in section 3, individual XGBoost models were trained for each location to forecast energy consumption. The model performance was evaluated quantitatively with the use of metrics (as shown in *Table 3*), and qualitatively by looking at the model's predictions plotted against the ground truth. *Figure 10* exemplifies this comparison for the city of Bergen, which also shows the train, test, and validation splits (similar plots were made and evaluated for each location, and can be made available on request). As described in section 3, the test dataset was concealed during training and serves to evaluate predictive accuracy on completely unseen data. Preliminary observations suggest the models proficiently track daily energy consumption patterns, yet occasionally falter in accurately predicting periods of significantly high or low demand.

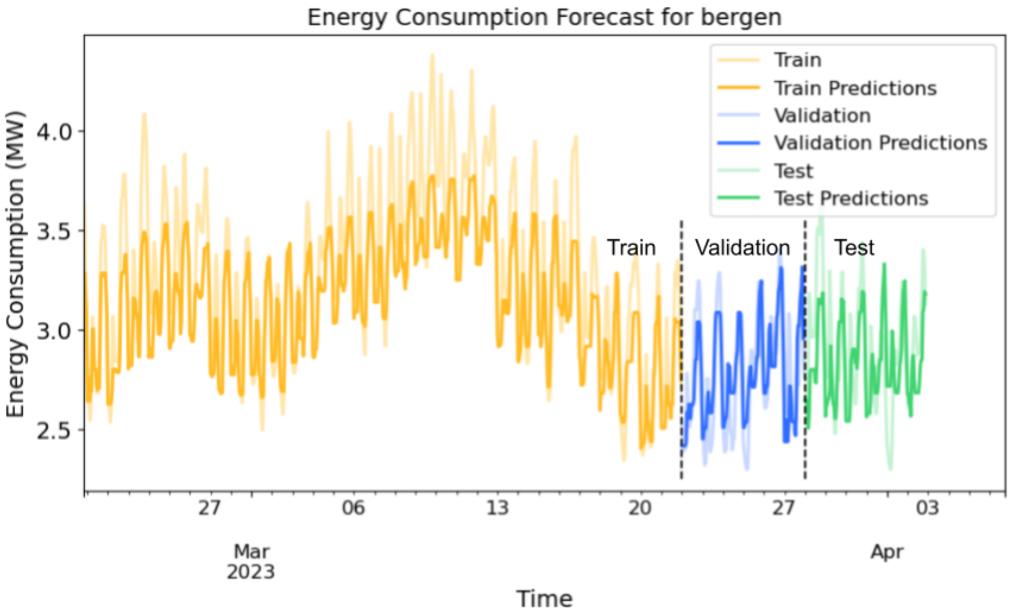

Figure 10: Visualization of the model training, validation, and predictions for Bergen

Relating to our objective of day-ahead forecasting, we illustrate how the model would act in a real-life scenario in *Figure 11* and *Figure 12*. The dotted green line represents predictions during the 5-day data gap, while the solid green line indicates the 'Day-ahead predictions' (corresponding to the 6th day). These forecasts project hourly energy consumption for the succeeding day, April 2 in the given instance, thereby showcasing the model's prospective utility in real-world energy management applications.

The model has successfully predicted the 6-day period. Therefore, the first technical project objective is achieved, "Create a machine learning model that predicts 6 days forward, thus overcoming the 5-day gap constraint, and solving the main business objective of forecasting day-ahead predictions".

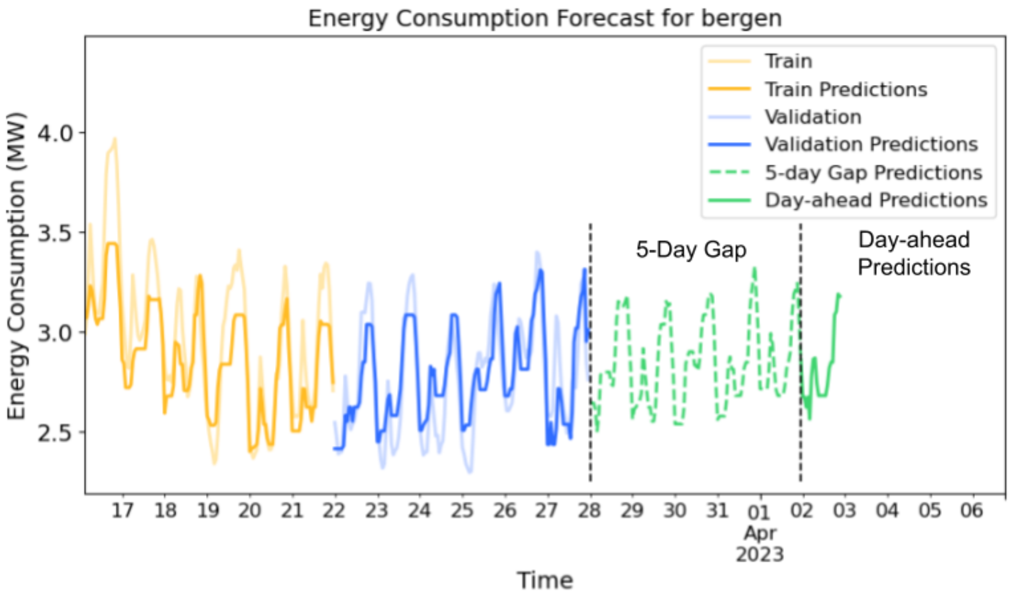

Figure 11: Visualization of the 5-day gap and day-ahead forecast for Bergen

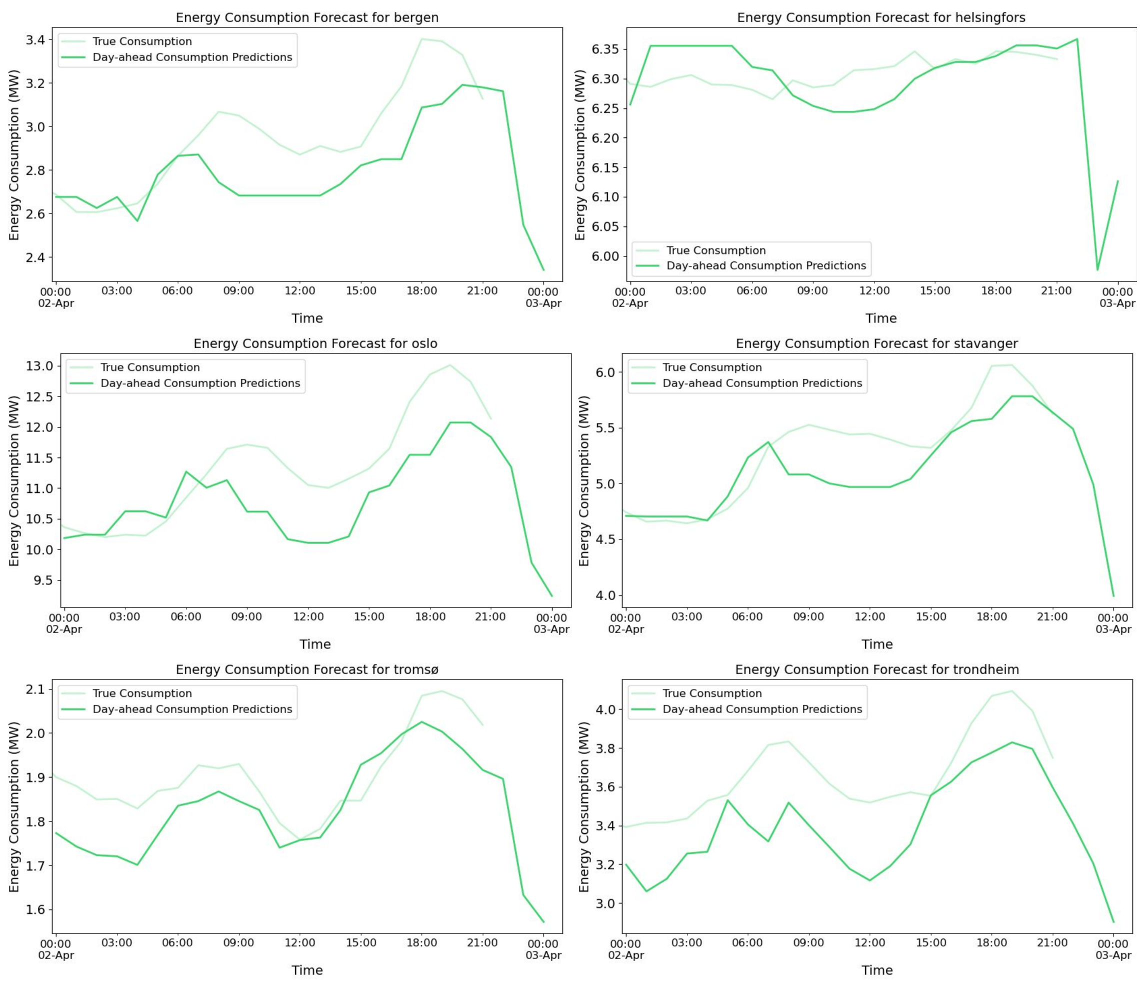

Figure 12: Day-ahead predictions for all Cities

4.2 Error Analysis

The quantitative evaluation of the model performance consisted of calculating the different metrics presented in section 3.5 on the test set for each location. These values are contained in *Table 3*. We also provide a visualization of the actual error on the test set for each city in *Figure 13*.

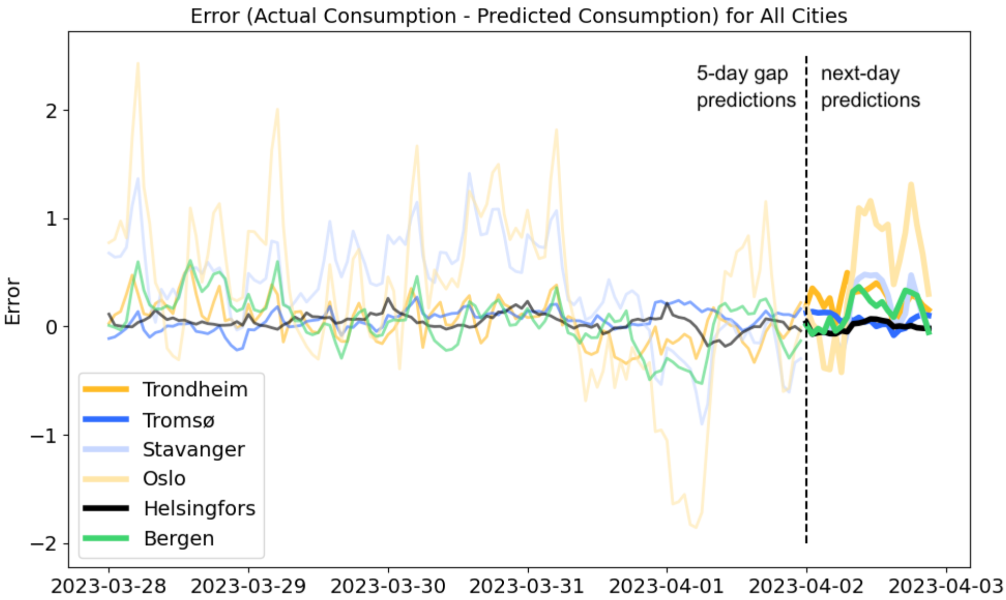

Figure 13: Error Graph for Each City

Table 3: Error Metrics in Day-Ahead Predictions for All Cities

	TRD	TOS	SVG	OSL	HEL	BGO	Average
MAE	0.17	0.09	0.45	0.68	0.06	0.19	0.27
RMSE	0.21	0.11	0.55	0.84	0.08	0.24	0.34
MAPE	0.05	0.05	0.08	0.06	0.01	0.07	0.05
SD	0.20	0.09	0.44	0.78	0.07	0.23	0.29

Comparing metrics such as MAE and RMSE between locations may not provide meaningful insights, as the consumption values in each location exhibit varying distributions, including differences in their maximum, minimum, and average values. The MAE and RMSE are however useful when it comes to evaluating the business value of the forecasting model, discussed in section 4.5 Business Value.

As described in section 3.5 Model Evaluation and Metrics, looking at the normalized MAPE score gives a more interpretable result that is independent of the specific consumption distribution at each location. We notice that all the Norwegian cities are quite similar, while Helsingfors has a significantly lower score.

The error analysis reveals that while some cities such as Helsingfors benefit from relatively accurate and consistent predictions, others such as Oslo and Stavanger experience higher variability and instances of significant over/under-estimation. This suggests that the model may require further tuning to account for factors specific to each city or that there are inherent challenges in the data or the model's structure that need to be addressed (Saha et al. 2022).

From table 3, we can see that the MAPE for all cities are less than 15%. Thus we have achieved our second technical project objective, "The model's predictions should have a MAPE for the day-ahead predictions lower than 15%.".

4.3 Feature Importance

The importance score of a feature represents its contribution to the model's predictive power. Higher scores indicate that the feature is more influential in mg predictions, while lower scores suggest that the feature has less impact. The F-scores presented in the table below provide valuable insights into the significance of each feature in our predictive model. The F-scores are calculated using 'weight', ie. how many times the number of times a feature is used to split the data across all trees (XGBoost Developers 2023). Therefore, an isolated examination of the F-scores for each location is essential, placing emphasis on their mutual proportions rather than their absolute magnitudes. This approach is driven by the variations in data distribution across locations and also influenced by factors such as early stopping during model training.

We notice the absence of the 'month' and 'year' features. This is because the data points in the test set are from the same month and year, resulting in a feature importance of 0, therefore excluded.

Table 4: Feature F-scores for Different Cities

City	Feature	F-score	F-scores visualized
TRD	hour	939	
	dayofweek	443	
	dayofyear	650	
	temperature	1167	
OSL	hour	199	
	dayofweek	77	
	dayofyear	136	
	temperature	132	
TOS	hour	891	
	dayofweek	310	
	dayofyear	568	
	temperature	778	
SVG	hour	214	
	dayofweek	82	
	dayofyear	193	
	temperature	131	
HEL	hour	215	
	dayofweek	81	
	dayofyear	94	
	temperature	230	
BGO	hour	168	
	dayofweek	59	
	dayofyear	163	
	temperature	139	

Hour:
The hour of the day consistently stands out as a highly influential factor across all cities. This suggests that our model heavily relies on the time of day for accurate energy consumption predictions, which aligns with daily demand fluctuations.

Day of Year:
Another significant feature is the day of year. This feature captures seasonal variations and is valuable for anticipating energy usage trends during holidays, seasonal transitions, or annual events.

Temperature:
Its impact on energy needs is well-established, making it a vital variable in our forecasting model. We notice that in colder cities such as Helsingfors, Tromsø, and Trondheim its importance generally is greater.

Day of the Week:
Although the day of the week has lower F-scores it remains relevant. This feature captures weekly consumption cycles likely associated with workweek and weekend variations in energy usage patterns.

In conclusion, our analysis using XGBoost and F-scores provides valuable insights into the relationships between features within each city are essential, emphasizing the significance of time-related and weather-related variables in energy forecasting. Furthermore, these findings underscore the importance of customizing models for each location. Tailoring models to the specific characteristics and patterns of individual cities is essential for accurate and effective energy consumption forecasting.

4.4 Benchmarking Against a Naïve Forecast

Here we compare the XGBoost model performance against the naive model as described in section 3.4. The naive model simply replicates and copies the true value from the most recent data available.

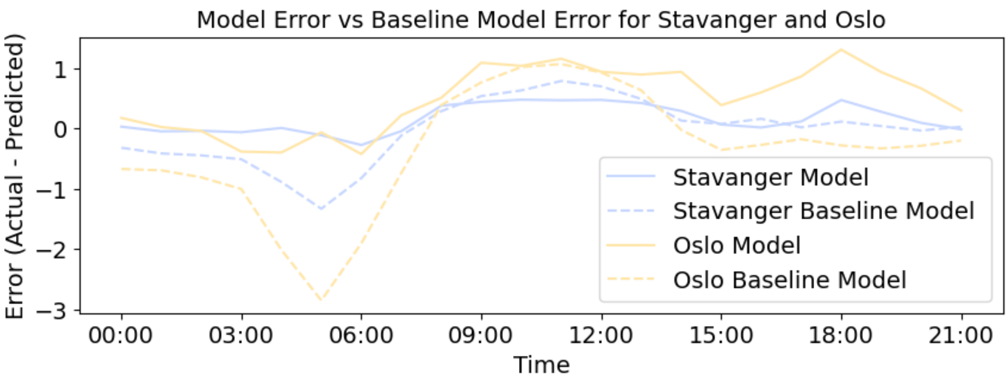

Figure 14: Comparative predictive performance illustrating Stavanger's pronounced improvement in MAE and Oslo's notable enhancement in SD against the baseline model.

Table 5: Baseline Model Results

		TRD	TOS	SVG	OSL	HEL	BGO	Average
Baseline Model	MAE	0.30	0.09	0.40	0.79	0.07	0.22	0.31
	SD	0.27	0.12	0.54	1.00	0.04	0.22	0.36

Referencing *Table 5*, we observe a marked enhancement in performance metrics with our custom-designed machine learning model when compared against the naive baseline. Our model demonstrates superior precision in forecasting, as evidenced by its lower mean absolute error (MAE) and standard deviation (SD) across all locations, with the sole exception of the SD for Helsinki. This suggests a consistent and reliable predictive strength inherent in our machine-learning approach.

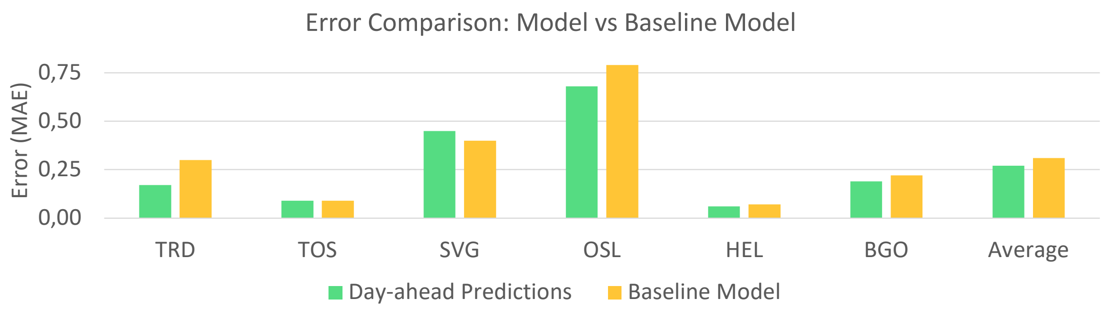

Figure 15: Error comparison between the actual and baseline models

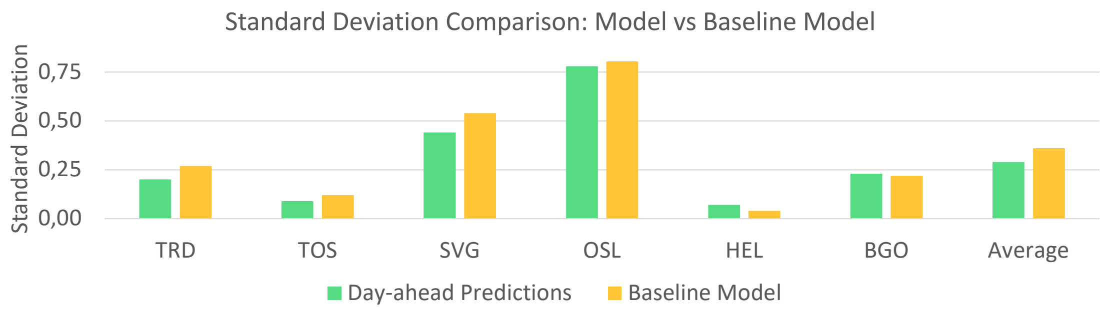

Figure 16: Standard deviation comparison between actual and baseline models

In quantifiable terms, the MAE for our model's day-ahead predictions is recorded at 0.27, which shows a notable improvement over the baseline's 0.31. This translates to a 0.04 increment in the accuracy of our model's MAE. Furthermore, the SD of our predictions at 0.29 is less than the baseline's 0.36, indicating a 0.07 enhancement in the consistency of our model's SD. Thus, the model achieves the fourth technical project objective being achieved, "Create a machine learning model that creates day-ahead energy predictions with higher accuracy (lower error) and higher consistency (lower standard deviation) than a baseline model.".

4.5 Business Value

Addressing the Business Objectives

In this section, we will outline how the technical achievements of our model result in real business value for Aneo. In order to do so, we will reassess each of the five Business Objectives and how our model achieves them.

1. The first business objective was to generate precise next-day hourly energy forecasts. In section 4.1 (Prediction Results) we presented the capabilities of the model in creating day-head prediction. We achieved this by forecasting a 6-day period since the dataset we only have access to data from 5 days ago. In section 4.4 (Benchmarking Against a Naïve Forecast) we showed that our model was more precise in generating day-ahead predictions than a baseline model, scoring better in both mean absolute error and standard deviation. In terms of business value, this means that our model is highly beneficial for Aneo as it enables the company to engage in more strategic and cost-effective energy procurement. Accurate predictions equate to optimized energy usage, minimized waste, and reduced operational costs. Furthermore, such precision in forecasting facilitates better load balancing, contributing to stability within the power grid and enhancing Aneo's ability to offer competitive pricing and reliable service to its customers. By leveraging our model's forecasts, Aneo can strengthen its market position and uphold its commitment to sustainable energy practices.
2. The second business objective was to offer day-ahead forecasts for specific locations. Section 4.1 (Prediction Results) also showcased our model's capability in creating location-specific forecasts. This creates business value for Aneo because it allows for a tailored approach to energy management across different regions. By providing location-specific forecasts, Aneo can optimize resource allocation and energy distribution to meet the unique demands of each area. This targeted forecasting supports Aneo in enhancing operational efficiency, reducing the risk of energy shortfalls or surpluses, and delivering a more reliable energy supply to diverse geographic locations. Ultimately, this granular level of forecasting empowers Aneo to maintain a high standard of service quality and customer satisfaction in all serviced areas.
3. The third business objective was to develop a model that could be implemented in new locations. Since the model is uses location-specific data for training, the model can easily be implemented by Aneo in any future energy grid expansions. This is vital because it ensures the scalability and adaptability of Aneo's energy management solutions. As the company grows and enters new markets, the ability to rapidly deploy accurate forecasting models in these areas becomes a competitive advantage. This scalability supports Aneo's strategic business expansion plans, enabling a swift response to new opportunities without the need for extensive redevelopment of forecasting tools. The model's flexibility in accommodating new data from emerging smart grids ensures that Aneo can maintain a consistent level of service excellence and operational efficiency, irrespective of location.
4. The fourth business objective was to create a dashboard that Aneo can use in its day-to-day forecasting operations. An example illustration of a Monitoring Dashboard can be seen in section 6.2 (Monitoring Dashboard). The development of the Monitoring Dashboard represents a crucial interface between the complex analytics of the machine learning model and the practical, decision-oriented environment of business operations. By presenting predictive outcomes alongside key performance indicators (KPIs), the dashboard empowers Aneo to not only observe but also to interpret and act on the data with confidence and clarity. The user-centric design ensures that the information is accessible, actionable, and aligns with the daily workflow, thus streamlining the decision-making process. Moreover, this tool is designed to be dynamic and interactive, enabling Aneo to drill down into specifics when needed and to customize views to focus on different aspects of energy consumption and forecasting. This enhances the operational agility of Aneo, making the Monitoring Dashboard an invaluable asset for both strategic planning and day-to-day management.
5. The fifth business was to achieve a model that limits energy forecast error to under x MW hourly, with x defined by Aneo experts. In section 4.2 (Error Analysis) we used error met-

rics to assess the accuracy of our model. By aligning our model's error margins with the thresholds set by Aneo's experts, we provide a tailored solution that respects the company's risk appetite and operational benchmarks. From *Table 3*, we saw that our model had an average MAE of 0.27 across all the locations with a standard deviation of 0.29. Our initial suggestion for x would therefore be 0.6 MW. The ability to limit energy error to under the specified MW threshold ensures that the model operates within a safe margin, thus safeguarding against significant financial losses due to prediction inaccuracies. This precise error calibration allows Aneo to maintain a competitive edge by optimizing its energy procurement strategies and managing its resource allocations more effectively. In an industry where marginal gains are crucial, this attention to detail in error management can translate into considerable cost savings and improved operational resilience. Additionally, this approach provides Aneo with a clear criterion for model acceptance, ensuring that only the most reliable and efficient models are deployed in live environments, further protecting the company from volatile market conditions and fluctuating energy demands.

Proposal for Future Iterations & Rationale for New Business Objective

"The process of CRISP-DM is not linear, and it is an iterative cycle in which the answers to the initial questions almost always generate additional questions"(Clancy et al. 2023). The CRISP-DM process shows us that as we understand more, we often uncover new questions. Based on our findings, we suggest a new business goal for future research.

The equations for potential savings and earnings are as follows:

- C ($X per kWh): Cost per unit error of day-ahead predictions.
- MWh: Total MWh supplied to the specific location.
- OH (Operational Hours): Number of hours in the forecast period (in this case 24 hours).
- OM: Operational margin or profit per kWh.

The potential savings and earnings can be calculated using the following equations:

$$\text{Potential Savings} = (MAE_{\text{Baseline}} - MAE_{\text{Model}}) \times C \times MWh \times OH \tag{11}$$

$$\text{Potential Earnings} = PS \times OM \tag{12}$$

The monetary value of a predictive model is inherently tied to its precision; as the error margin of the model decreases, its financial worth escalates correspondingly. For Aneo, this inverse relationship means that each incremental improvement in the model's accuracy can lead to significant cost savings. Lower error rates in energy forecasts directly translate into more efficient energy distribution, minimizing instances of overproduction or underproduction that typically result in financial losses or missed opportunities for profit.

Understanding the monetary impact of a machine learning model is critical for businesses like Aneo, as it quantifies the value of technological investments and guides strategic decision-making. By assigning a dollar figure to the predictive accuracy of our model, we enable Aneo to measure the direct financial benefits, such as cost reductions from more precise energy allocation and increased earnings from operational efficiencies. This financial quantification is not just a measure of return on investment; it is an essential metric that influences budgeting, resource allocation, and future investments in technology. Moreover, it provides Aneo's leadership with concrete data to justify the adoption of the model to stakeholders and can be a persuasive element in securing funding for scaling up the implementation across new locations. The ability to monetize the model's accuracy ensures that our project outcomes are aligned with Aneo's business objectives, demonstrating the practical and fiscal viability of our solution in the real-world energy market.

4.6 Shortcomings

Lack of Lag Features

As explained in 3.2 Data Analysis Methods and Tools, we identified that the autocorrelation of our dataset diminished after 20 hours, indicating that past values had little influence on future predictions beyond this time frame. Given that our data access was restricted to less than 5 days, incorporating lag features extending further back seemed irrelevant. However, one alternative approach could have been to generate synthetic data points for the missing five-day period by predicting energy consumption for those days. This could potentially create a more robust dataset that might still benefit from lag features within the significant 20-hour window, thus possibly enhancing the predictive accuracy of our model.

Lack of Accuracy Estimates

Unlike probabilistic models that can offer confidence intervals or posterior probabilities, XGBoost outputs do not automatically convey the certainty of the predictions made. This can be a significant disadvantage since the cost of incorrect predictions for Aneo's energy allocation has a high cost.

XGBoost as a Black-Box

XGBoost is often criticized for its "black-box" nature, as it provides little insight into the specific data points or features that drive its predictions (Rodriguez 2023). This makes it difficult for the end user to understand or trust the decision-making process of the model. In the context of Aneo, who has access to industry experts who often revise predictions, it would be especially insightful to gain access to this model's assumptions and decision-making.

The Relativity of Baseline Model Comparisons

Using baseline models for comparison carries inherent limitations due to the arbitrary nature of what constitutes a baseline. A baseline model is typically a simple, less complex algorithm that serves as a point of reference. However, the choice of this baseline can vary widely. This variability means that the perceived performance of a new model can be misleadingly inflated or deflated depending on the chosen baseline. A model might seem exceptionally performant when compared to an overly simplistic baseline, or conversely. Therefore, the assessment our model requires in relation to the baseline model should be interpreted from a critical standpoint.

Feature Selection Reconsideration:

While the current model incorporates temporal factors and temperature, it may not encapsulate the full scope of variables affecting energy consumption. Excluding additional weather parameters and crucial socioeconomic indicators, such as electricity pricing, could result in a model that inadequately represents consumption patterns.

Model Complexity:

Employing only one model in a machine learning ensemble can lead to several issues: it reduces the diversity of patterns captured from the data, increases the risk of overfitting, and retains the inherent bias and variance of the single model used. Ensembles typically outperform individual models by combining their strengths and mitigating weaknesses such as individual errors, lack of robustness, and performance ceilings. Without the ensemble's collective decision-making, the model may also be less adaptable to changes in data over time, potentially resulting in less accurate predictions and a less robust system overall.

Lack of Generalizability:

Utilizing a non-global machine learning model like XGBoost when seeking generalizability poses significant drawbacks, such as limited applicability beyond the specific conditions for which it was trained, a propensity to overfit to its training dataset, and poor performance when faced with data that differ from its initial scope. These models may not scale well to broader data sets, can be biased, and often require substantial retraining or modification to adapt to new tasks, making them less robust and more resource-intensive. Consequently, they may incur higher costs and complexities in maintenance and deployment in diverse real-world situations, contrasting with the broader applicability and efficiency of global models.

Data Set Limitations:

The range of variables in our model does not reflect all possible factors influencing electricity consumption. Notably absent are considerations of economic activity, demographic changes, and extreme weather conditions, all of which significantly impact energy demand. This exclusion can affect the model's capacity to fully predict consumption fluctuations, potentially limiting the reliability of the results when applied to broader scenarios.

4.7 Ensuring the Reliability of the Pipeline

Before extracting any business value from the model, the reliability and robustness of the complete pipeline need to be verified. The pipeline encompasses every stage from data processing, feature engineering, training and validation, and finally running model inference. These are the steps taken to ensure the reliability of the pipeline, and tests we have thought about.

Testing the Pipeline Logic

The main step to ensure reliability was the usage of robust, well-documented, and very well-tested libraries. Specifically, we used Pandas and Numpy for data processing, XGBoost for model training and inference, and Sklearn for computing metrics. The heavy lifting is handled by these libraries, so we need not conduct extensive low-level tests. However, there is some custom logic in the code, such as for splitting the train, test, and validation data. We ran simple tests to evaluate the reliability of the data splitting by plotting the splits for different time intervals and cities. We included testing edge cases (at the beginning and end of a time series), and everything worked as expected.

Synthetic Tests

Although we use robust and reliable libraries, and the custom logic in the code is simple, we deemed it a good idea to establish some tests to evaluate how the model performs on data coming from different distributions. After all, we only had a time series coming from six cities to work with, and the distributional breadth covered by this data is unlikely to cover the possibilities of future input coming from different sources. Consequently, we devised a test to generate synthetic data with artificial seasonality and trend components (where we make some synthetic time series easier to predict than others). The tests will be passed if the model is able to perform better than a specific threshold for metrics such as the MAE, MAPE, and RMSE.

Tests Based on Expert Feedback

In addition to running tests with the synthetic data, we will implement tests that are based on the actual energy consumption data. Specifically, we will have an energy expert look at a past consumption time series, and draw an upper bound (both for over- and underestimation) for how inaccurate the model would reasonably be allowed to be in a business setting (for example, how large an error would you tolerate from a human energy analyst tasked with predicting the consumption?). In case the bound is exceeded, the test is failed. Before we can implement this test, we will need to collaborate and coordinate with energy experts from Aneo.

Cross-validation and Error Analysis

The next step we implemented to ensure pipeline reliability was the computation of average metrics across the entire dataset using sliding-window cross-validation (as described in section 3.5). There are several reasons why this is important to ensure reliability. Two of the most important are reliable error estimation and bias reduction. Training and testing the model across different folds of the dataset gives us a better indication of real-world performance. It also prevents evaluation that is biased to a particular subset of the data. The error metrics from the cross-validation represent the average errors across the different validation sets shown in *Figure 9*. From *Table 6* and *Figure 17* we can see that the model performs better than average on our selected demonstration of day-ahead forecasting. When implementing the model and predicting on several different dates, we would expect results closer to the cross-validation scores on average.

Table 6: Error metrics from Cross Validation

	TRD	TOS	SVG	OSL	HEL	BGO	Average
MAE	0.27	0.12	0.43	0.80	0.15	0.27	0.34
RMSE	0.30	0.14	0.47	0.89	0.19	0.30	0.38
MAPE	0.10	0.09	0.11	0.10	0.03	0.13	0.09

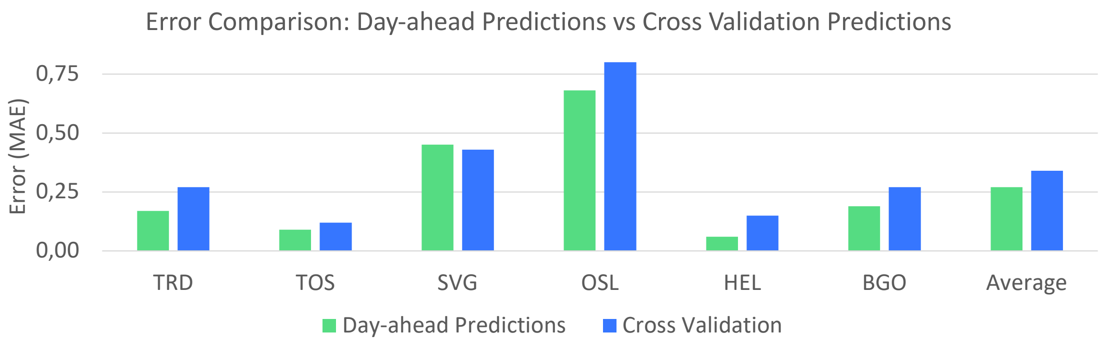

Figure 17: Comparions of day-ahead prediction on April 2 and cross-validation errors

5 Deployment and Recommendations

In this section, we describe a deployment and implementation plan for our model, relevant future actions for stakeholders, limitations of the data and our method, as well as suggestions for how future analysis can be conducted to best complement the results from this report.

5.1 Deployment Plan

We plan to host our model through an API service and let the client use this API to get predictions for their data. The API will also serve as the frontend dashboard application, and the client will be able to use this to connect to and get predictions from the API. As the data comes directly from the client, we choose to build a service where they can upload new data to the API (through the dashboard interface), and get the most recent and relevant predictions at any time.

To achieve this goal, we opted to use the Google Cloud AI Platform to streamline data preparation, model training and validation, deployment, monitoring, and code management. We consider the cost of using the Google platform to be justified, as it automatically handles important issues such as security and scalability. The ML workflow that the AI Platform provides is described in figure 18.

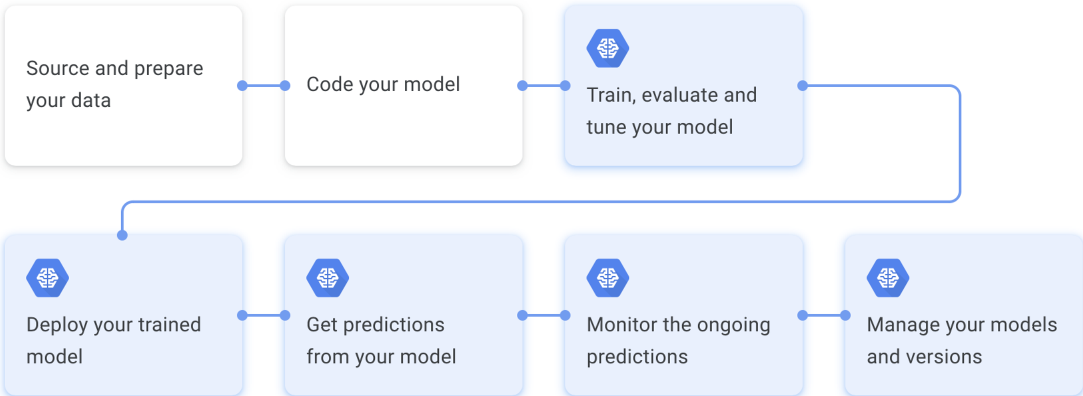

Figure 18: Google Cloud AI Platform ML Workflow

Concretely, we will orchestrate the deployment of the model using the Kubeflow Pipelines SDK service provided by the Google AI Platform. This is a flexible service that lets you set up a custom deployment pipeline. The steps to setting up the pipeline are (1) to build the code and scripts that define each stage of the pipeline, (2) create a Docker container for the code in each step, and (3) finally define the pipeline with a special Python script integrated with Kubeflow Pipelines SDK. These are the steps of the deployment pipeline:

1. **Client data upload**
   The pipeline is triggered when a client uploads new data to the API. This means that a new model will be trained, evaluated, and deployed.

2. **Data preparation and cleaning**
   We start by preparing and cleaning the data. This procedure will follow the feature engineering and data cleaning steps described in section 3.3. If there are inconsistencies in the data provided by the client or anything that makes it different enough from the Aneo dataset that our current data preprocessing method will not work, the input from the client will not be accepted and the pipeline is aborted. This will subsequently be communicated to the client.

3. **Hyperparameter tuning and metrics computation over the whole dataset**
   We will tune the model hyperparameters according to the sliding window cross-validation

strategy detailed in section 3.5. It is imperative that this is done on the entire dataset, that the model is trained separately on different intervals, and that the final metric used to select the optimal hyperparameter is the average of the final metric over the different intervals. This is to make sure the model performance is satisfactory regardless of when it is used (as described in depth in section 3.5).

The data uploaded from the client will dictate the details of the hyperparameter tuning scheme. We will only process the last 10000 hours of the provided dataset. This is to save computational Google Cloud resources, and since patterns too far in the past are less likely to have a predictive influence on the near future. If the length of the time series is shorter, we will tune the hyperparameters on all the available data.

The optimal average metrics (corresponding to the optimal hyperparameters) computed over all the time intervals will be compared to the MAPE threshold of 0.15 (from the business objectives section). If the MAPE for the uploaded dataset is higher than the threshold, we abort the pipeline, as it means that the model performance was not good enough. This will be communicated both to the client and technical staff on our end who will look into the issue.

4. **Training and evaluating the final model**
After step 2, we will know if the model is sufficiently performant from a business perspective on average over all time intervals in the dataset. The final step is to train and evaluate the model on the most recent data provided, as this is the model we will deploy to the client. We use the hyperparameters selected in step 2, split the last 6*24 as the test set, and choose the n hours before as the train set (n chosen during hyperparameter tuning). We finally compute the metrics over this interval. Again, if the MAPE exceeds the threshold of 0.15, the pipeline will be aborted, and the same response as in step 2 will occur.

5. **The final predictions are returned**
If all the preceding steps succeed, the predictions for the data uploaded from the client will be returned. This process will be repeated when new data is uploaded by the client, but we will require a wait time between uploads. We decided that the train and validation pipeline could only run 5 times a week for a single client. If the upload frequency exceeds this rate, predictions acquired from running inference with the most recent model will be returned. This will be communicated to the client.

5.1.1 Implementation Plan

As we have already created most of the code necessary to make the deployment plan possible, there is not much more to implement. It only remains to code the frontend dashboard application, and a simple backend API to pair it with. We therefore deem it realistic to have all steps described in the deployment plan ready within 1-3 months. However, to ensure that the model is completely ready for client usage, we will need sufficient testing before final deployment. The timeline for the implementation plan is as follows (note that the actions are to be implemented sequentially, so cumulatively summing the time estimate column at a time step will give the time until that step is completed):

Time Estimate	What to implement
1-3 months	Finalize code pertaining to each step in the deployment plan, as well as the frontend dashboard application. This includes the infrastructure code that defines the interactions with the Google Cloud platform and the API that serves customer requests and takes data uploads.
1 month	Run tests on the ML pipeline. This includes testing for correctness, scalability, load handling, and security. The reason we plan that this phase will not last long is that the Google AI platform automatically handles security and scalability. We will therefore focus on testing whether the model and code behave correctly and as expected. The planned tests are described in detail in section 4.
1 month	Make the final preparations for deployment. We will spend this time running final tests, communicating the release of the product to the clients, and finally deploying the pipeline.
Until the pipeline is retired	Conduct monitoring, and maintenance, and continue running tests to account for unexpected events. This part is described in depth in section 6.

5.2 Relevant Actions Towards Different Stakeholders

This section will detail relevant actions different stakeholders are recommended to take based on results from our analysis, as well as suggestible actions based on the model predictions. Since we conducted this project for Aneo, but the model and predictions are made for one of Aneo's clients, we can differentiate between Aneo as a stakeholder and the client as a stakeholder. There are actions that can be taken by Aneo to make their consumption prediction service as attractive as possible to their clients, and there are actions the clients can take based on the provided predictions.

5.2.1 Aneo Actions

The listed actions for Aneo are mainly based on improving the accuracy of the model since that is what defines the problem statement, and a more accurate model will equate to satisfying the problem statement task to a greater degree. Consequently, Aneo can expect more satisfied clients and increased revenue by following these actions.

- Invest resources into building an even stronger model, in order to minimize the risk of inaccurate forecasts, and the corresponding monetary value. This action is aimed at the management and resource-allocating stakeholders at Aneo.
- Collaborate with the retailer clients to gather data from more sources, as one of the primary limitations of the dataset is the few available features. Underline that you will be able to provide better predictions if the client is able to provide better data. This action is mainly aimed at the client relationship managing stakeholders since it depends on communication with the client.
- Invest in collaborations with technical data science staff, and staff with more domain expertise in the field of energy. This may guide the analysis and interpretations in a better direction, and the modelers may be able to incorporate performance-enhancing inductive biases into the model (information about the data that the model is unlikely to find itself). This action is aimed at engineering- and business managers at Aneo, who are responsible for putting together the engineering teams.
- Focus on building a service that does not solely rely on individual data samples sent from client retailers. With the new innovations within the field of machine learning and partic-

ularly deep learning, you should leverage the power of big data to create a general model that can work on data provided by any client. Spend resources on constructing a reliable and far-reaching data collection pipeline, collect proprietary data from different sources on consumption and weather, and strive towards attaining a large data collection upon which future models can be based. While the models we trained for this project were fitted locally on an individual time series, an idea for the future can be to build a global model trained to extract patterns from much different time series. A large, high-quality dataset paired with such a global model can lead to significant performance enhancements. This action is aimed at engineering manager stakeholders who choose how to leverage the in-house talent, the actual engineers who will build the new data pipeline, as well as senior managers responsible for resource allocation (such as the CTO).

5.2.2 Client Actions

These actions will depend on the needs of the specific client, as all the information we have is that the client stakeholder seeks accurate consumption forecasts. We can not know for sure what they want to do with the forecasts. However, we know that the client is an energy retailer, and we consequently consider it safe to assume that the client wants to use the information to guide the volume of their energy production.

- The first action is to optimize energy production by using the predictions to choose how much energy to produce at different times for consumption in different cities. Clearly, the analysis tells us that there is seasonality and trend related to when the consumption goes up or down. Take advantage of the STL analysis, and optimize the energy production distribution according to the time of day and year. As the price for residual produced energy that was not consumed is highly unstable, optimizing the production according to demand will alleviate financial risks. Concretely, the client can increase energy production that will generate energy usable during the day, and increase energy production that will be usable during the colder months.

  In the end, the client can use the model to guide the exact choice of how many MW of energy to produce for consumption at a given hour. However, the client should use the results from the error analysis to evaluate when the model is likely to make a mistake. For example, as Trondheim has the highest and most volatile error, consider using other methods in combination with our model to arrive at a final production target for this city.

  This action is aimed at stakeholders responsible for choosing the final energy production value. This includes energy analysts and their executive managers.

- The next action the client can take is to use the demand predictions to optimize prices. A higher price can be charged when demand goes up, as the customer will be willing to pay more, and a lower price should be offered when low consumption is expected. This will ensure that consumption aligns with production, and that the financial risks of not producing the exact demanded amount are reduced. This second action is particularly relevant when the energy is already produced and it is too late to change the amount.

  This action is aimed at business stakeholders who select energy pricing, such as energy/price analysts and their managers.

- The last action is to gather more data from new sources in addition to the old, as this will give more accurate predictions. Furthermore, this measure should be tailored to the individual cities. Again, Trondheim has the highest error, while the error for Helsingfors is consistently low. More resources should thus be spent on improving the data quality (and gathering more data) from Trondheim than Helsingfors. This action is aimed at the client stakeholders responsible for the company data strategy. This may include senior data scientists, engineering managers, the chief technology officer, and other members of the management team.

5.3 Limitations and Improvements

5.3.1 Dataset Limitations and Improvements

Our dataset, while free from missing values, lacks depth in terms of the number of features available to us. The only provided features are time, temperature and consumption. While we hypothesize that the industry segment of the retailer's customers is not sensitive to weather variables other than temperature, research shows that the household segment definitely is (source). The article by Kang & Reiner found that for households, weather features such as rain and sunshine have great potential to affect people's consumption behavior. The effects of rain and sunshine depended on the time of day and whether it was the weekend or a workday. For example, sunshine in the afternoon leads to lower energy consumption caused by an increase in outdoor activities. By implementing forecasted weather features such as direct sun radiation and rainfall, combined with our time series data, we can gain predictive power by utilizing the synergies of these different factors. The effect of humidity and wind were also studied, but these factors were deemed negligible. It is important to note that the research was conducted in Ireland, and we do need to keep in mind the climatic differences between that region and the Nordics. Different weather features might impact consumer behavior differently for Aneo's client. However, as weather forecasts are easy to obtain, hypothesis testing can be done at a low cost of time and capital.

Another relationship that would be interesting to explore is that of electricity prices towards consumption. Data from the Norwegian Water Resources and Energy Directorate (NVE) showed a decrease in consumption in all southern regions of Norway (NO1, NO2, NO5) during the winter of 2021/22, when they experienced high electricity prices (Norges vassdrags- og energidirektorat (NVE) 2022). Both the actual and temperature-adjusted consumption went down in the period. Meanwhile, the northernmost region of NO4 increased their consumption in the period, as they experienced reduced prices. The shift in consumption was mainly caused by households and the service sector. Industrial customers usually have long-term power contracts and are therefore not susceptible to short-term price fluctuations. The relationship of prices with consumption should be explored in both Aneo's household and industry segments to confirm or debunk these findings.

A big limitation in our dataset is the 5-day gap between the time day-ahead-forecasts are made, and the latest available consumption data. In our autocorrelation analysis in section 3.2, we found that using lag features would only be significant up to 20 hours in the past, which is quite a lot smaller than 5 days. Therefore we did not use lag features in our model. The reason for the gap in the data is unknown, but with the widespread use of smart power meters in Norway, the possibility of obtaining more current data should be discussed with the client, as this would likely better model predictions.

5.3.2 Method Limitations and Improvements

With shortcomings inherent to our XGBoost implementation already discussed in section 4.6, this part will instead focus on limitations of the method in a broader sense, ie. what limitations we face in the deployment of the model. One issue is that of scalability and generalizability with respect to feature engineering. Using individual XGBoost models for each location allows for customization to specific local patterns, but increases the complexity of the overall forecasting system. Managing multiple models can be challenging, especially when scaling to more locations or updating the models with more features over time. The introduction of new features such as different weather features must be tested individually for each forecasting region. In addition, over time the underlying processes driving energy consumption might change, making previously selected features less relevant. Continuous monitoring and updating of the feature set are necessary to maintain model accuracy.

The use of one single predictor, ie. XGBoost might limit adaptability and performance. While XGBoost is renowned for its efficiency and accuracy in various machine learning tasks, its singular approach may not fully capture the complex, dynamic nature of time series data. To address these challenges, the integration of AutoML solutions, specifically AutoGluon-TimeSeries, presents a compelling alternative. AutoGluon-TimeSeries is an advanced AutoML library designed for probabilistic time series forecasting, which synergizes statistical models, machine learning approaches, and ensembling techniques. This comprehensive approach enables the system to automatically select, optimize, and combine multiple models, thereby enhancing predictive accuracy and robustness. As demonstrated in the study "AutoGluon-TimeSeries: AutoML for Probabilistic Time Series Forecasting" (Shchur et al. 2023), AutoGluon-TimeSeries exhibits strong empirical performance across various benchmark datasets, outperforming traditional methods in handling complex time series forecasting tasks. By leveraging AutoGluon-TimeSeries, forecasters can benefit from a more adaptive, scalable, and efficient forecasting framework, overcoming the limitations inherent in using a single model like XGBoost.

In our journey to refine our energy consumption forecasting model, it's crucial to remember that the performance of any model we choose is deeply intertwined with the quality and relevance of the data we provide. This principle highlights that while exploring various models is beneficial, the real determinant of success lies in the data itself (Dalal 2023b). Moreover, the most accurate predictive model is not necessarily the one that creates the most value. While achieving high predictive accuracy is crucial for your business objectives, the practicality of implementing and using the model also plays a critical role. A model that is slightly less accurate but much easier to implement and use can sometimes be more beneficial than a highly accurate but complex model.

This perspective aligns with the practical wisdom of the 80-20 rule (Dalal 2023b), emphasizing that an operational, albeit imperfect, system can be more valuable than a theoretically perfect but undeployed model. The key is to strike a balance between accuracy and practicality, ensuring that our model not only predicts well but also adapts to real-world complexities and provides tangible benefits in its application.

5.4 Future Analysis

To further complement and enhance the results obtained from our current model, future analysis can be directed towards several key areas:

- **Comparative Analysis with Enhanced Models:** Implement the suggested improvements in the dataset and method, as discussed in previous sections, and conduct a comparative analysis against the original model. This involves integrating additional weather features, exploring the impact of electricity prices, and employing advanced methods like AutoGluon. The objective is to quantitatively assess how these enhancements affect the model's performance in terms of accuracy, scalability, and practical applicability. Such a comparative study will not only validate the effectiveness of the improvements but also provide insights into areas where further refinements are needed.
- **Testing on Unseen Data:** Continuously testing the model on new, unseen data is crucial for assessing its robustness and adaptability to changing patterns in energy consumption. This will help in identifying any overfitting issues and in ensuring that the model remains relevant and accurate over time.
- **User Feedback Incorporation**: Regularly incorporating feedback from the end-users of the forecasting system can provide valuable insights into how the model can be fine-tuned to better meet the practical needs of the business.
- **Integration with Real-Time Data Streams**: Leveraging real-time data, especially from smart meters and IoT devices, can significantly enhance the model's responsiveness and accuracy in real-time forecasting scenarios.

6 Maintenance and Monitoring

The deployment of Aneo's energy consumption forecasting model marks a significant milestone in harnessing data science for operational optimization. However, the real test begins post-deployment—ensuring the model's resilience, reliability, and relevance over time. This is where vigilant monitoring and meticulous maintenance come into play, laying the foundation for continuous improvement and sustained success (Sculley et al. 2015).

6.1 Key Performance Indicators

Model Performance KPIs

For Aneo, the model's ability to predict energy consumption accurately is paramount. Our error metrics defined in section 3.5 provide natural model performance KPIs. They enable Aneo's client to pinpoint the model's prediction accuracy down to the megawatt-hour, which is vital for optimizing energy production schedules, grid maintenance, and load distribution.

MAE gives a straightforward average of error magnitude, allowing operational teams to grasp the model's performance without a complex statistical background. MSE assigns greater weight to more significant errors, which is crucial in scenarios where over or under-prediction of energy needs can lead to substantial operational inefficiencies or even blackouts. We use RMSE, and by providing a measure in the same units as the forecast, it allows for direct interpretation of the errors in the context of actual energy consumption—vital for communicating with stakeholders who may be less familiar with statistical measures (Parmenter 2015). The last error metric, the MAPE value also provides a clear and intuitive way to benchmark the model across different regions and time periods.

Business KPIs

In the context of energy forecasting and management, Business Key Performance Indicators (KPIs) serve as vital metrics to evaluate and guide the performance and strategic direction of a company. For Aneo, these KPIs not only measure the success and impact of their predictive models but also reflect how their services contribute to broader business objectives. On the other hand, KPIs for Aneo's clients focus on the tangible benefits they receive from using Aneo's forecasting services, such as operational efficiency and cost savings.

For Aneo:

1. Forecast Accuracy Improvement: This KPI gauges the progress in the precision of Aneo's energy forecasting models over time. Enhancing forecast accuracy is pivotal for Aneo as it directly influences the reliability and value of the service provided to clients. Improved accuracy leads to better decision-making in energy production and distribution, ultimately enhancing Aneo's reputation and trustworthiness in the market.
2. Client Acquisition and Retention Rate: This metric monitors the growth in Aneo's client base and the company's effectiveness in maintaining long-term relationships with its clients. A higher rate indicates successful market penetration and client satisfaction with Aneo's services, signifying the company's stability and growth potential in the competitive energy sector.
3. Sustainability Contribution Score: This KPI reflects Aneo's contribution towards sustainable energy practices. It measures the impact of Aneo's forecasting services on reducing unnecessary energy production, thereby decreasing environmental footprint and aiding clients in achieving their sustainability targets.

For Aneo's Clients:

1. Operational Cost Savings: This KPI measures the reduction in financial risk and operational costs due to Aneo's precise energy forecasts. By closely aligning energy production with actual consumption, clients can avoid costly imbalances between forecasted and actual energy use. Accurate forecasting helps minimize the need to settle energy at the fluctuating regulating price, which can significantly deviate from the regular spot price, thereby stabilizing operational expenses and reducing financial uncertainty.

2. Demand Response Efficiency: This KPI measures how effectively clients manage and adapt to changing energy demands based on Aneo's forecasts. It focuses on the clients' ability to adjust their energy consumption and production in response to peak and off-peak periods as predicted by Aneo. Effective demand response management leads to operational benefits like enhanced grid stability, reduced energy costs, and potential incentives from energy conservation programs. This KPI is crucial for clients looking to optimize their energy usage and contribute to a more sustainable energy ecosystem.

6.2 Monitoring Dashboard

A well-constructed monitoring dashboard is essential for Aneo's operational team to maintain a real-time pulse on the forecasting model's performance. The design of this dashboard should incorporate:

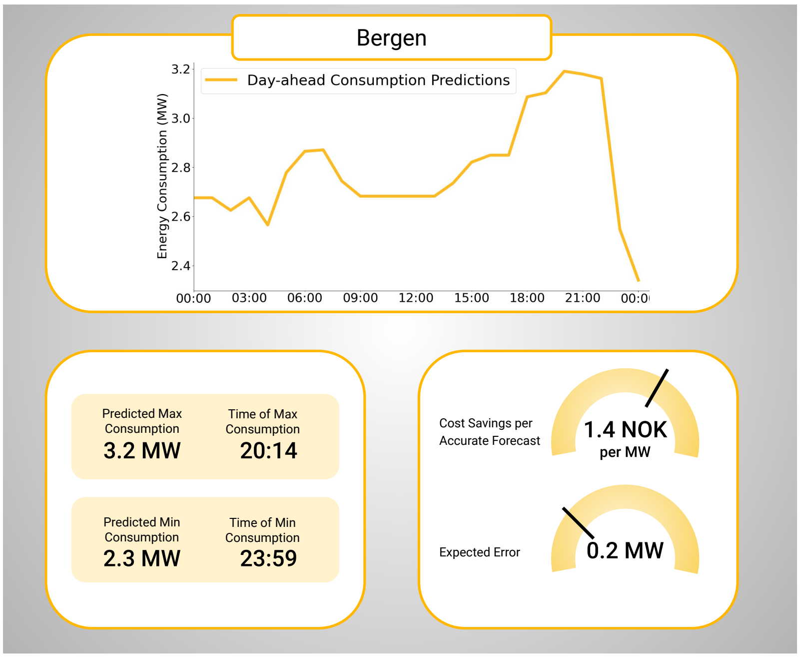

Figure 19: Aneo's Energy Forecasting Dashboard

Figure 19 illustrates an example of what a dashboard could look like for Bergen. A similar dashboard would be made for Aneo's other cities. The dashboard features an intuitive interface with clear, customizable sections for various KPIs, enabling quick assessments of system health and predictive accuracy. With predictive alert capabilities and full IT infrastructure integration, it offers a proactive tool for operational teams to manage and anticipate system performance, ensuring Aneo's readiness for future challenges and business scalability.

6.3 Strategic Risk Management and Contingency Planning:

When deploying a complex machine learning model, especially in a dynamic field like energy consumption forecasting, numerous factors can lead to deviations from expected performance. Here are potential risks and mitigation strategies:

Model Performance Deviation:
Post-deployment, the model's performance might degrade due to changes in energy consumption patterns or external factors not included in the training dataset. Regular monitoring of prediction accuracy against real-time data is essential. Anomalies in prediction errors should trigger alerts, and if a consistent pattern of errors emerges, retraining the model with updated data or revising the model's architecture may be necessary.

Infrastructure Failures:
Dependence on cloud services like Google Cloud AI Platform entails risks of downtime or service disruptions. Establishing a robust failover strategy, including backup services and clear communication channels with the service provider, can mitigate these risks. Additionally, a local version of the model could serve as a temporary solution during cloud outages.

Security Breaches:
As the system involves data transmission over networks, there's a risk of unauthorized access. Continuous security monitoring, regular penetration testing, and swift incident response plans are necessary. Alarms for unusual access patterns or data breaches should be in place to enable immediate action.

Compliance and Regulatory Changes:
Energy sector regulations can change, impacting data handling or model requirements. Staying abreast of regulatory developments and maintaining a flexible system architecture that can adapt to new compliance needs is crucial.

Human Intervention:
Despite automation, the need for human oversight remains. A dedicated team should monitor system health, performance metrics, and respond to alerts. Training for quick troubleshooting and having detailed documentation can empower this team to handle issues effectively.

6.4 Lessons Learned and Feedback

The true challenge of this project lay not in the intricacies of machine learning, but in crafting of solutions that align with our client's unique requirements. The technical aspects, while complex, were easy to overcome; it was the discernment of Aneo's specific needs and translating these into actionable, data-driven objectives that truly tested our adaptability and client-focused strategy. This nuanced understanding and client alignment is what defined our project's success.

Embracing a user-centric approach, our project synergizes the principles of the Business Analytics Model and Design Thinking, prioritizing the identification of clear objectives that resonate with our client's needs. This philosophy ensures that our solutions are not just technologically advanced but also pragmatically aligned with user requirements. We find immense value in setting these objectives upfront, which serves as a beacon throughout our development process, guiding our coding and analytical efforts toward delivering tangible value to Aneo.

The Cross-Industry Standard Process for Data Mining (CRISP-DM) has been instrumental in shaping our project's trajectory. This structured approach has illuminated the entire lifecycle of our data-driven initiative, from understanding Aneo's operational context to deploying the final predictive model. It has fostered a disciplined yet flexible methodology, reinforcing the importance of iterative phases that align with real-world business scenarios.

References

Aneo (2023). *Et nordisk fornybarkonsern*. URL: https://www.aneo.com/ (visited on 24th Nov. 2023).

Chen, Tianqi and Carlos Guestrin (Aug. 2016). 'XGBoost: A Scalable Tree Boosting System'. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '16. ACM. DOI: 10.1145/2939672.2939785. URL: http://dx.doi.org/10.1145/2939672.2939785.

Clancy, R., D. O'Sullivan and K. Bruton (2023). 'Data-driven quality improvement approach to reducing waste in manufacturing'. In: *The TQM Journal* 35.1, pp. 51–72. DOI: 10.1108/TQM-02-2021-0061.

Cleveland et al (1990). 'STL: A Seasonal-Trend Decomposition Procedure Based on Loess'. In: *Journal of Official Statistics* 6.1, pp. 3–73.

Combelles, Annie, C. Ebert and Percival Lucena (2020). 'Design Thinking'. In: *IEEE Software* 37.2, pp. 5–9. DOI: 10.1109/MS.2019.2959328.

Dalal, Nisha (Nov. 2023a). *Lecture 5: Lifecycle of a Data Science Project*. Slide 7 in TDT4259 – Applied Data Science. Lecture presented at Norwegian University of Science and Technology (NTNU).

— (2023b). *Lecture 8: Lifecycle of a Data Science Project*. Slide 39 in TDT4259 – Applied Data Science. Lecture presented at Norwegian University of Science and Technology (NTNU).

Fernandez, Javier (2023). *Time-Series Forecasting Based on Trend and Seasonal components*. URL: https://towardsdatascience.com/time-series-forecasting-based-on-the-trend-and-seasonal-components-26b92866e548 (visited on 24th Nov. 2023).

Giles A. Hindle, Richard Vidgen (2016). 'Developing a business analytics methodology: A case study in the foodbank sector'. In: *European Journal of Operational Research*. DOI: 10.1016/j.ejor.2017.06.031.

IEA (2023). *Smart Grids*. URL: https://www.iea.org/energy-system/electricity/smart-grids (visited on 24th Nov. 2023).

Khalil, Mohamad (2022). 'A Multi-task Learning Approach to Short-Term Load-Forecasting for Multiple Energy Loads in an Educational Building'. In: *IEEE*. DOI: 10.1109/AEECA55500.2022.9919082.

Norges vassdrags- og energidirektorat (NVE) (2022). *Høy strømpris har gitt redusert forbruk*. Accessed on 2023-11-26. URL: https://www.nve.no/nytt-fra-nve/nyheter-energi/hoey-stroempris-har-gitt-redusert-forbruk/ (visited on 26th Nov. 2023).

Norwegian Ministry of Petroleum and Energy (2023). *The Power Market*. URL: https://energifaktanorge.no/en/norsk-energiforsyning/kraftmarkedet/ (visited on 24th Nov. 2023).

Parmenter, David (Apr. 2015). *Key Performance Indicators: Developing, Implementing, and Using Winning KPIs*. en. 1st ed. Wiley. ISBN: 9781118925102 9781119019855. DOI: 10.1002/9781119019855. URL: https://onlinelibrary.wiley.com/doi/book/10.1002/9781119019855 (visited on 24th Nov. 2023).

Rodriguez, Carlos (2023). *The Notorious XGBoost*. Accessed on 2023-11-26. URL: https://towardsdatascience.com/the-notorious-xgboost-c7f7adc4c183 (visited on 26th Nov. 2023).

Saha, Enkanta, Runa Saha and Krishna Mridha (Oct. 2022). 'Short-Term Electricity Consumption Forecasting: Time-Series Approaches'. In: *2022 10th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*. Noida, India: IEEE, pp. 1–5. ISBN: 9781665474337. DOI: 10.1109/ICRITO56286.2022.9964624. URL: https://ieeexplore.ieee.org/document/9964624/ (visited on 24th Nov. 2023).

Saltz, J. (2021). 'CRISP-DM for Data Science: Strengths, Weaknesses and Potential Next Steps'. In: *IEEE Big Data*. DOI: 10.1109/BigData52589.2021.9671634. URL: https://dx.doi.org/10.1109/BigData52589.2021.9671634.

Sankhyana (2023). *Importance of Data Science in the Energy and Utilities Industry*. URL: https://www.linkedin.com/pulse/importance-data-science-energy-utilities#:~:text=Data%20science%20enables%20the%20energy,maintenance%2C%20and%20optimize%20asset%20utilization. (visited on 24th Nov. 2023).

Sculley, D. et al. (Dec. 2015). 'Hidden technical debt in Machine learning systems'. In: *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*. NIPS'15. Cambridge, MA, USA: MIT Press, pp. 2503–2511. (Visited on 24th Nov. 2023).

Shchur, Oleksandr et al. (2023). *AutoGluon–TimeSeries: AutoML for Probabilistic Time Series Forecasting*. arXiv:2308.05566 [cs.LG]. arXiv: 2308.05566 [cs.LG]. URL: https://arxiv.org/pdf/2308.05566.pdf (visited on 26th Nov. 2023).

XGBoost Developers (2023). *XGBoost Python API Documentation*. Accessed on 2023-11-26. URL: https://xgboost.readthedocs.io/en/stable/python/python_api.html#module-xgboost.plotting (visited on 26th Nov. 2023).

Zimmer, Marcel, Thiemo Pesch and Andrea Benigni (Sept. 2021). 'Time-Series Analysis and Forecasting of Power Consumption using Gaussian Process Regression'. In: *2021 International Conference on Smart Energy Systems and Technologies (SEST)*. Vaasa, Finland: IEEE, pp. 1–6. ISBN: 9781728176604. DOI: 10.1109/SEST50973.2021.9543253. URL: https://ieeexplore.ieee.org/document/9543253/ (visited on 24th Nov. 2023).