Danmarks Tekniske Universitet

DTU

Machine Learning for Energy Systems Assignment 1

Authors

Konstantinos Kountras - s242786
Jonas Wiendl - s243543
Thorri Jokull Thorsteinsson - s242760
Bjartur Jorfi Ingvason - s242754
Kristian Jerichau Nissen - s203895

October 29, 2025

Contents

List of Figures iii

List of Tables v

Matchmatical notation vi

1 Introduction 1

2 Data preprocessing & Input rationale 1
- 2.1 Evaluation Methodology . . . 1
- 2.2 Dataset Construction . . . 1
  - 2.2.1 Exploratory Data Analysis . . . 2
  - 2.2.2 Handling Missing Data . . . 2
  - 2.2.3 Feature Selection . . . 3

3 Model 1: Wind Power Prediction 4
- 3.1 Linear Regression - Baseline Model . . . 4
- 3.2 Polynomial Regression . . . 4
- 3.3 Weighted Least Squares . . . 5
- 3.4 Regularization . . . 6
- 3.5 Results and comparison of models . . . 7

4 Revenue Evaluation 8
- 4.1 Market revenue formulation . . . 8
- 4.2 Revenue-based evaluation . . . 9

5 Unsupervised Learning 9

6 Model 2: Decision-Focused Learning 11
- 6.1 Predict-then-Optimize Results . . . 11
- 6.2 DFL Joint training Results . . . 12

7 Conclusion 12

A Appendix 13

Use of AI

Artificial Intelligence (AI), specifically LLMs, were used in the presented case study to an appropriate degree. Time-consuming tasks like creating a preliminary inclusive document layout of the report were supported by the use of ChatGPT. Furthermore, it assisted in coding scripts, used for data visualization and analysis. Finally, it streamlined the process of formatting of the LaTeX document, helping with efficient generation of tables and figures, and supported the overall writing process by drafting initial ideas for the sections of the report when necessary. It should be noted that all written material still remains an effort of personal and teamwork and is subject to copyright, securing the originality of the work done. When using AI tools, it is extremely important to rigorously verify the correctness of outputs. As expected, some inconsistencies were encountered. This happened most often with coding, where multiple iterations of running the script were needed before expected outputs were achieved.

Participation Table

Student Code	**Ch. 2 (%)**	**Ch. 3 (%)**	**Ch. 4 (%)**	**Ch. 5 (%)**	**Ch. 6 (%)**
s242786	14%	20%	29%	15%	40%
s243543	24%	20%	14%	40%	15%
s242760	24%	20%	14%	15%	15%
s242754	24%	20%	14%	15%	15%
s203895	14%	20%	29%	15%	15%
Total	**100%**	**100%**	**100%**	**100%**	**100%**

Table 1: Participation percentages per task across 5 students. Each column sums to 100%.

List of Figures

1 Example for XGBoost prediction for wind production gap 2
2 Cumulative Profit Comparison - Test Period 10
3 Linear Regression model coefficients. Wind speed features (`mean_wind_speed_Nex` and `mean_wind_speed_Hammer_Odde_Fyr`) have the strongest contribution to predicted power, while market-related variables show minor influence. 16
4 Predicted vs. observed power for the Linear Regression model on the test set. The dashed line represents the ideal prediction ($y = x$). The model demonstrates limited accuracy in capturing high variability in wind power production. 16
5 Validation RMSE across polynomial degree combinations for NexĂ¸ and Hammer Odde Fyr wind speeds ($\alpha = 1$). The optimal combination corresponds to ($\text{Nex}^\circ = 3, \text{Hammer}^\circ = 2$). 17
6 Polynomial degree tuning for Ridge regression ($\alpha = 1$). The plots show the trade-off between bias and variance across polynomial degrees for Nexo and Hammer Odde Fyr wind speeds. The chosen degrees are $\text{Nex}^\circ = 3$ and $\text{Hammer}^\circ = 2$. 17
7 Polynomial Ridge model fits for wind-power relationships at Nexo and Hammer Odde Fyr. 18
8 Predicted vs. observed power for the Polynomial Ridge model ($\lambda = 1.43 \times 10^1$, $\text{Nex}^\circ = 3$, $\text{Hammer}^\circ = 2$). 18
9 Bandwidth selection for the 2D Locally Weighted Regression model. The optimal bandwidth $\lambda = 0.20$ minimizes validation RMSE, balancing bias and variance. 19
10 Bandwidth selection for 1D Weighted Least Squares applied to mean wind speed. The best smoothing parameter is $\lambda = 0.04$. 19
11 Smoothed 1D LWR fit ($\lambda = 0.20$) for mean wind speed. The curve captures the saturation effect of power output with increasing wind speed. 20
12 Predicted vs. observed power for the grid-based LWR model ($\lambda = 0.20$, grid$=20 \times 20$). The dashed line represents the ideal prediction. While local models improve flexibility, extreme regions remain challenging to estimate. 20
13 Ridge Regression: (Left) RMSE as a function of regularization strength λ on train and validation splits; (Right) coefficient shrinkage path showing how feature weights are reduced as λ increases. 21
14 Final Ridge Regression coefficients at optimal $\lambda = 49.24$. Wind speed features dominate, while most market and temporal variables show minor influence due to regularization. 21
15 Lasso Regression: (Left) RMSE as a function of regularization strength λ on train and validation splits; (Right) coefficient shrinkage path showing variable selection as λ increases. 22
16 Final Lasso Regression coefficients at optimal $\lambda = 0.0059$. Only a few dominant predictors (notably wind speed and temperature) remain non-zero, confirming the model's sparsity-inducing behavior. 22

17 Predicted vs. observed wind power on the test set for Ridge and Lasso models. The dashed line indicates the ideal 1:1 relationship. 23
18 OLS predictions versus oracle commitments on the test set. In the predict then optimize framework, the OLS model predicts commitments $\hat{p}$ produced by the optimization model. 24
19 Decision-Focused Learning (DFL) versus oracle commitments. The DFL model directly optimizes the decision-making objective (expected profit) during training. A bid-no bid strategy is observed. 24

List of Tables

1 Participation percentages per task across 5 students. Each column sums to 100%. . . . ii
2 Selected features used in the modeling process . . . 3
3 Overview of locally weighted regression configurations used for the WLS experiments. . . . 6
4 Compact comparison of Ridge and Lasso regularization schemes. . . . 7
5 Model comparison of all models used . . . 7
6 Comparison of test set RMSEs for daily retraining. . . . 8
7 Total Revenue Comparison Across Models (Test Period) . . . 10
8 Performance comparison of global and clustered Ridge regression models. Note that in contrast to the other model validation in this report, a simple holdout validation was performed. . . . 10
9 Statistical and economic performance of the DFL model (placeholders in red). 12
10 Summary statistics: general and power. . . . 13
11 Summary statistics: weather variables. . . . 13
12 Summary statistics: market variables. . . . 13
13 Performance metrics of the OLS regression model used for imputing missing wind production. . . . 13
14 OLS regression coefficients and significance levels. . . . 14
15 Summary of weather data gap handling during the training/validation period. 14
16 Summary of weather data gap handling during the test period. . . . 14
17 Summary of wind power gap filling using OLS regression. . . . 14
18 Market data gaps during training/validation period. . . . 15
19 Market data gaps during test period. . . . 15
20 Overall dataset reduction after preprocessing. . . . 15

Regarding the **mathematical notation** used in the current study, the symbols will comprise of the following:

- m: The total number of training examples.
- n: The total number of features.
- $x^{(i)}$: The feature vector for the i-th training example, where $x^{(i)} \in \mathbb{R}^{n+1}$.
- $y^{(i)}$: The target value (output) for the i-th training example.
- X: The design matrix of size $m \times (n+1)$, where each row represents a training example. A column of ones is prepended for the intercept term x_0.
- y: The vector of target values of size $m \times 1$.
- θ: The parameter vector (weights) of the model, of size $(n+1) \times 1$.
- $h_\theta(x)$: The hypothesis function, which outputs the predicted value for an input x.
- $\phi(x)$: A function that maps the original feature vector x into a new, potentially higher-dimensional, feature space. For example, for polynomial regression of degree p, $\phi(x)$ might be $[1, x, x^2, \ldots, x^p]^T$.
- $w^{(i)}$: The weight assigned to the i-th training example for a specific query point x.
- λ: The regularization strength in Ridge regression, controlling the penalty on model complexity to prevent overfitting.

1 Introduction

Accurate prediction and decision-making are central challenges in renewable energy integration, where intermittent wind generation must be optimally traded in electricity markets. In this project, a series of regression-based models for **wind power prediction and trading optimization** are developed and compared, progressing from simple linear baselines to more advanced decision-focused approaches. In particular, the objectives are to, (1) Evaluate **linear**, **non-linear**, and **regularized regression** models for wind power prediction, (2) Investigate **locally weighted regression** for capturing nonlinear wind-power relationships, (3) Implement and assess **Decision-Focused Learning (DFL)** to directly optimize trading profit, (4) Compare all methods using both *statistical* (RMSE, R^2) and *economic* (revenue-based) metrics. The analysis follows a structured approach, building on the concepts and the workflow, upon each previous secction progressively.

2 Data preprocessing & Input rationale

The dataset combines historical measurements of **wind production, weather variables, time variables and market prices** for the Danish DK2 bidding zone. The primary source of data used in this project was the EnergyDataDK platform, which provides real-time and historical energy data from across Denmark. Specifically, wind power production data was obtained from the Snorrebakken wind farm located on the island of Bornholm, as signed in the NDA.
The data were merged and cleaned, to accomodate the scope of the project. They were split chronologically into training and test subsets respecting and preserving temporal consistency. Missing values were handled mainly through interpolation. Suitable cross-validation scheme was employed. The main steps of the preprocessing are further analyzed in Section 2.2

2.1 Evaluation Methodology

To ensure realistic evaluation under temporal dependence, we implemented a **rolling-origin cross-validation** scheme with a training window of two months and a validation horizon of one month, sliding forward in one-month increments. For each split s, the model was trained on $\mathcal{D}^{(s)}_{\text{train}}$ and validated on the immediately subsequent $\mathcal{D}^{(s)}_{\text{val}}$, yielding K folds that span the entire time series

2.2 Dataset Construction

The required datasets were collected from *EnergyDataDK*, covering the DK2 price zone of the Snorrebakken wind farm. Four sources were used: Balancing prices (up- and down-regulation), Day-ahead market prices, Climate data from two nearby weather stations and Actual wind production from the Snorrebakken wind farm. All datasets were aligned to hourly resolution for consistency across sources. They form the basis for subsequent exploratory analysis, missing-value handling, dataset partitioning, and feature engineering.

2.2.1 Exploratory Data Analysis

Before preprocessing, the datasets were inspected to assess their coverage, consistency, and general characteristics. Summary statistics for each data category are provided in Appendix 10–12. For the wind power series, production ranges between 0 and 2.61 MW with a mean of 0.49 MW, indicating a highly skewed distribution towards low production. Weather variables cover a wide range of conditions, with wind speeds between 0 and 20 m/s and temperatures from −9.6°C to 30.8°C. Market variables exhibit strong variability, including occasional negative day-ahead prices and extreme balancing prices up to 10,000 DKK/MWh. The following subsection will explore the handling of this data.

2.2.2 Handling Missing Data

In order to avoid filling or removing long gaps, the training/validation and test periods were selected to balance the two sets. Roughly two and a half years were chosen for training and validation (01.01.2021 00:00 – 31.08.2023 23:00), and one full year for testing (15.08.2023 00:00 – 14.08.2024 23:00). This split was chosen to balance seasonal bias between the training and test sets, while still utilizing as much of the data as possible. To handle existing gaps, the following strategies were applied:
Weather data: Gaps up to six hours were filled by linear interpolation, while longer gaps were removed entirely to avoid unrealistic imputations. On average, interpolated gaps had a duration of around 2–3 hours across variables.
Wind production: Missing values were imputed using extreme gradient boosting algorithm. The model was trained on all available parameters as we are trying to reconstruct the missing power values, the prices carry information. The model achieved a normalized RMSE of 0.32% on the training set and was used to reconstruct 70 gaps (2,062 h in total, with a maximum length of 405 h). An example of the XGBoost-based reconstruction of a long gap is shown in Figure 1. Market data: Only a small number of short gaps were detected; these were handled by local averaging.
After preprocessing, the cleaned dataset retained 94.1% of the original training/validation data and 92.0% of the test data. Detailed statistics for each variable are provided in the appendix.

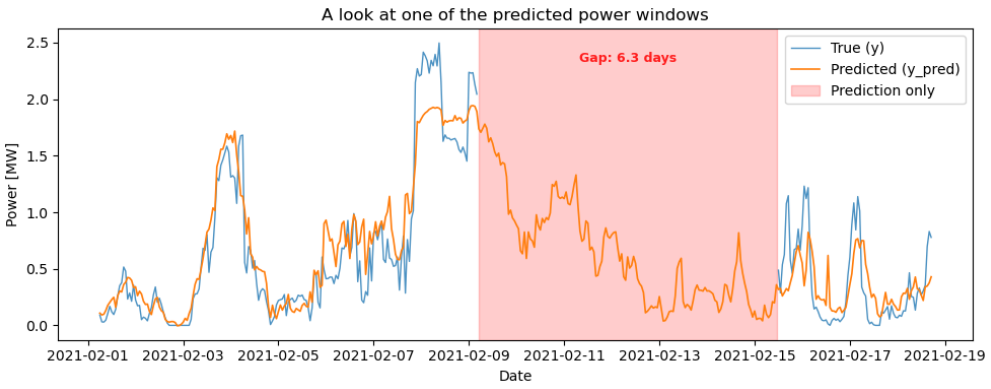


Figure 1: Example for XGBoost prediction for wind production gap

2.2.3 Feature Selection

Feature engineering can be very useful when modelling, highlighting patterns in the data, simplifying complex relationships, and improving the overall performance of a model. Calendar and cyclical features were created from the timestamp index to capture temporal patterns in the data. From the datetime index, the variables year, month, day, hour, day of the week, and day of the year were extracted to represent time-based information explicitly. To handle the cyclical nature of time variables, such as hours of the day and days of the week, sine and cosine transformations were applied. These cyclical encodings ensure that values close in time, such as hour 23 and hour 0, are represented as close in feature space, avoiding artificial discontinuities in the data. This transformation could help the model learn periodic patterns more effectively.

Feature selection was carried out using a combination of exploratory, statistical, and model-based methods. Initially, descriptive and visual analyses such as correlation matrices and summary statistics were applied to identify relationships between variables and detect potential multicollinearity. This helped highlight which features carried overlapping information and which appeared most relevant to the target variable.
Next, a Principal Component Analysis (PCA) was performed to assess the underlying structure of the data and to verify whether groups of features represented similar dimensions of variation. PCA helped confirm which features could be reduced or replaced without significant loss of information.
Finally, wrapper methods were used to evaluate feature importance directly in relation to model performance. Both forward selection and backward elimination were conducted, using validation error (such as RMSE) as the selection criterion. Forward selection started with no predictors and added features step by step based on performance improvement, while backward elimination started with all features and iteratively removed the least useful ones. Together, these methods ensured that the final set of features was both interpretable and efficient containing variables that contributed meaningfully to predictive performance while minimizing redundancy.

Table 2: Selected features used in the modeling process

Feature name	**Description**
mean_IGR_Nex	Mean solar irradiance (Nex station)
mean_temp_Nex	Mean air temperature (Nex station)
mean_hum_Nex	Mean relative humidity (Nex station)
prec_Nex	Precipitation (Nex station)
mean_wind_speed_Nex	Mean wind speed (Nex station)
mean_wind_dir_Nex	Mean wind direction (Nex station)
DK2_DKK	Day-ahead electricity price in DK2 (DKK)
SYSTEM_DKK	System price (DKK)
ImbalancePriceDKK	Imbalance price (DKK)

3 Model 1: Wind Power Prediction

3.1 Linear Regression - Baseline Model

Linear regression serves as a fundamental predictive modeling technique, establishing a linear relationship between a dependent variable and one or more independent variables. It forms the baseline for which more complex models will be compared. Both a closed form solution and a gradient descent (sklearn) are implemented without difference in results. closed form is formulated as:

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n = \theta^T x \tag{1}$$

The objective is to find the optimal parameters θ that minimize the cost function. The most common cost function is the Mean Squared Error (MSE), which measures the average squared difference between the predicted and actual values.

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 \tag{2}$$

3.2 Polynomial Regression

To capture non-linearities in the data, the linear regression model can be extended by applying non-linear transformations to the input features. This allows the model to fit more complex patterns.

$$h_\theta(\phi(x)) = \theta^T \phi(x) \tag{3}$$

The cost function remains the Mean Squared Error, but it is now optimized with respect to the transformed feature matrix $\Phi(X)$.

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (\theta^T \phi(x^{(i)}) - y^{(i)})^2 \tag{4}$$

The approach of polynomial regression, allows a linear model to fit non-linear data by transforming the input features into a space where a linear relationship exists. The model remains linear with respect to the parameters θ, so the same closed form solution is applicable.
In this project, we captured the non-linear relationship between wind speed and power by applying polynomial transformations of degree p only to the wind speed feature. This design models the physical non-linearity of wind-to-power conversion while maintaining computational efficiency. All other meteorological, time, and market variables remained linear.
Each training fold was independently standardized using `StandardScaler` to prevent look-ahead bias. The final design matrix per fold was constructed as $\mathbf{X}_{\text{train}} = [\,\phi_p(x_{\text{wind}}),\ \tilde{\mathbf{X}}_{\text{linear}}\,]$, where $\phi_p(x_{\text{wind}})$ are the polynomial wind features and $\tilde{\mathbf{X}}_{\text{linear}}$ denotes the remaining standardized non-wind features.
A two-level tuning strategy was employed: first, the polynomial degree $p \in \{1, 2, 3, 4, 5, 6\}$ was optimized, over the two mean wind speeds features derived from the Nex and Ode Fyr

stations, using rolling-validation RMSE; second, with the optimal degrees for each wind speed $p^\star$ fixed, the regularization strength λ was tuned over a logarithmic grid from 10^{-5} to 10^7. The optimal combination $(p^\star, \lambda^\star)$ was selected to minimize the mean validation RMSE across all K folds:

$$(p^\star, \lambda^\star) = \arg\min_{p,\lambda} \frac{1}{K} \sum_{k=1}^{K} \mathrm{RMSE}_{\mathrm{val}}^{(k)}(p, \lambda). \tag{5}$$

3.3 Weighted Least Squares

To increase model flexibility beyond a single global parameter vector, we employ **Weighted Least Squares** (WLS). This non-parametric method fits a unique linear model for each query point $\mathbf{x}_0$ by minimizing a weighted sum of squared errors, giving more influence to nearby training points. The local coefficients $\hat{\boldsymbol{\beta}}(\mathbf{x}_0)$ are found by solving:

$$\hat{\boldsymbol{\beta}}(\mathbf{x}_0) = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{m} w_i(\mathbf{x}_0) \left(y^{(i)} - \boldsymbol{\beta}^\top \mathbf{x}^{(i)}\right)^2, \tag{6}$$

where the weights $w_i(\mathbf{x}_0)$ depend on the distance between the query point $\mathbf{x}_0$ and each training point $\mathbf{x}^{(i)}$:

$$w_i(\mathbf{x}_0) = K\left(\frac{\|\mathbf{x}^{(i)} - \mathbf{x}_0\|_2}{\lambda}\right), \tag{7}$$

and $K(\cdot)$ is a kernel function (e.g., Gaussian, Epanechnikov, Tricube) controlling the decay of influence with distance.
The parameter λ acts as a **bandwidth** controlling the locality of the fit and directly influences the bias-variance trade-off. A small λ forces a narrow neighborhood, highly flexible but noisy fit (low bias, high variance), while large λ yields the opposite effects. Therefore λ serves as a hyperparameter that needs to be tuned through a validation process. The resulting closed-form local solution is very similar to the known OLS method and holds as follows:

$$\hat{y}(\mathbf{x}_0) = \mathbf{x}_0^\top (\mathbf{X}^\top \mathbf{W}(\mathbf{x}_0) \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{W}(\mathbf{x}_0) \mathbf{y}, \tag{8}$$

where $\mathbf{W}(\mathbf{x}_0)$ is a diagonal matrix with elements $w_i(\mathbf{x}_0)$.

There are different kernels one could select from, e.g. Gaussian, Epanechnikov, Tricube, Uniform, Triangular. To produce similar qualitive behaviour, the case study uses the gaussian kernel throughout, but bandwidth tuning remains the most critical factor. Due to the high computational complexity of locally weighted regression, which scales with the number of features and data points, the analysis was restricted to 1D and 2D selected feature model setups (subsection 2.2). The number of local regressions and matrix inversions grows as moving to higher dimensions thus restraining computational efficiency and feedback testing. The first setup (2D) included both wind speed and imbalance price to capture power output, while a power solely as a function of standardized wind speed setup (1D) was tested as well. Each configuration was similarly tuned through the process explained in subsection 2.1.

Bandwidth values were scanned over $\lambda \in [0.01, 0.6]$, and the configuration minimizing the mean validation RMSE across folds was selected:

$$\lambda^\star = \arg\min_\lambda \frac{1}{K} \sum_{k=1}^{K} \mathrm{RMSE}_{\mathrm{val}}^{(k)}(\lambda). \tag{9}$$

The above methodology was implemented in the following setups presented in Table 3

Table 3: Overview of locally weighted regression configurations used for the WLS experiments.

Model	**Kernel application**	**Query grid / points**
1D WLS (wind speed only)	Pointwise per query	40-150 evenly spaced points
2D WLS (wind speed + price)	Pointwise per grid cell	20×20 grid of representative points

3.4 Regularization

Continuing the model built in Section 3.1 to mitigate the effects of increased model variance and sensitivity to noise one may apply **regularization** to control coefficient magnitude and improve generalization. Specifically, both **Ridge** (L2) and **Lasso** (L1) regressions were implemented and compared using the same rolling-origin validation framework described in Section 2.1.

Mathematical formulation. For a linear model with parameters $\boldsymbol{\theta}$ and design matrix $\mathbf{X}$, the regularized objective reads:

$$J_{\mathrm{reg}}(\boldsymbol{\theta}) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) - y^{(i)} \right)^2 + \lambda\, \Omega(\boldsymbol{\theta}), \tag{10}$$

where $\Omega(\boldsymbol{\theta})$ penalizes model complexity and λ (hyperparameter tuning) controls the trade-off between bias and variance. The choice of penalty term $\Omega(\boldsymbol{\theta})$ determines the behavior of the estimator. The formulations examined in this study are **Ridge** (ℓ_2) and **Lasso** (ℓ_1) regression. Both share the same general optimization objective of Equation 10, but differ in how they constrain the model parameters.
Both methods share the same closed-form structure for their optimization (Equation 10), but differ in the nature of the penalty term and its impact on model sparsity.

$$\hat{\boldsymbol{\theta}}_{\mathrm{ridge}} = (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^\top \mathbf{y}. \tag{11}$$

The cleaned and standardized dataset described in Section 2.2 was used. Both Ridge and Lasso models were validated through **rolling-origin cross-validation** (window = 2 months,

Table 4: Compact comparison of Ridge and Lasso regularization schemes.

Aspect	**Ridge (ℓ_2)**	**Lasso (ℓ_1)**
Penalty term $\Omega(\boldsymbol{\theta})$	$\frac{1}{2}\|\boldsymbol{\theta}\|_2^2$	$\|\boldsymbol{\theta}\|_1$
Effect on coefficients	Continuous shrinkage ($\downarrow$ magnitude)	Sparse shrinkage (many $\to 0$)
Bias-variance trade-off	Bias $\uparrow$, Variance $\downarrow$	Bias $\uparrow$, Variance $\downarrow$, Sparsity $\uparrow$
Interpretability	All features retained (scaled)	Subset of dominant features
Typical use case	Collinear / high-dim. features	Feature selection / interpretable models

horizon = 1 month), consistent with the temporal dependency of the data. Regularization strengths were scanned over logarithmic grids:

$$\lambda_{\text{ridge}} \in [10^{-2}, 10^{7}], \quad \lambda_{\text{lasso}} \in [10^{-5}, 10^{1}]$$

and the optimal $\lambda^\star$ minimizing the mean validation RMSE across folds was selected:

$$\lambda^\star = \arg\min_{\lambda} \frac{1}{K} \sum_{k=1}^{K} \text{RMSE}_{\text{val}}^{(k)}(\lambda). \tag{12}$$

3.5 Results and comparison of models

Table 5: Model comparison of all models used

Model	**Optimal $\lambda^\star$**	**Validation RMSE [MW]**	**Test RMSE [MW]**
Linear	—-	—-	0.4593
polynomial	14.2510	0.304	0.4770
WLS 1D	0.04	0.3338	0.4823
WLS 2D	0.2	0.3364	0.4652
Ridge	49.24	0.3128	0.4590
Lasso	0.0059	0.3129	0.4550

A comparison of the predictive accuracy of the models, evaluated by test set Root Mean Squared Error (RMSE) as shown in Table 5, reveals that regularized linear models provided the best performance for this forecasting the wind power
The Linear Regression baseline achieved a test RMSE of 0.4593 MW. Applying regularization yielded clear improvements. Ridge (L2) regression slightly reduced the error to 0.4590 MW by penalizing large coefficients. The best-performing model from a statistical standpoint was Lasso (L1) regression, which achieved the lowest RMSE of 0.4550 MW. This superior performance is attributed to its ability to perform feature selection. In contrast, the non-linear approaches did not improve generalization accuracy. The Polynomial regression, despite having the best validation RMSE, resulted in a higher test RMSE (0.4770 MW) than the linear baseline. Similarly, the Weighted Least Squares (WLS) models underperformed the regularized linear methods. While the 2D WLS (0.4652 MW) was a significant improvement

over the 1D version (0.4823 MW) underscoring the value of including market features, it did not surpass the accuracy of the simpler Lasso model.

Table 6: Comparison of test set RMSEs for daily retraining.

Model	**Whole Feature Space**	**Selected Features**
Ridge	0.4201	0.4273
Lasso	0.4289	0.4288
OLS (squared parameters)	0.4201	0.4372
Baseline OLS	**0.4158**	**0.4158**

Setting the use cases of the models to a more realistic scenario, where we predict for the day ahead and then retrain the model based on the actual power output, allowing the model to be more adaptable and should reduce the error. This approach would be even more important when using forecasted data for the OOS predictions (we assume the test data is forecasted, although it is actual). Results are shown in Table 6.
When models were trained once on the entire training period, the feature-selected versions achieved lower RMSEs, confirming that removing weak or redundant predictors improved generalization. However, under the daily retraining regime, models using the full feature space slightly outperformed or matched the selected versions. This reversal occurs because frequent retraining reduces the risk of overfitting, the models are continually updated on recent data, allowing them to exploit even marginally informative variables without overfitting to long-term noise. Regularization (Ridge/Lasso) further stabilizes this behavior by automatically shrinking unhelpful coefficients. As a result, the broader feature set provides more flexibility to adapt to short term changes in the input-output relationships.
The relation between RMSE of each method and the subsequent revenue produced by the bidding strategies will be further detailed in Section 4. Individual results and figures regarding the hyperparameter tuning, validation process and final predictions of each of the aforementioned developed models will be provided in the Appendix A in relevant sections.

4 Revenue Evaluation

The economic value of each predictive model was assessed by computing its realized revenue under the Danish day-ahead and balancing markets. At each hour t, the committed and realized power were denoted $\hat{p}_t$ and p_t, respectively, while λ_t^D, $\lambda_t^{\uparrow}$, and $\lambda_t^{\downarrow}$ represent the day-ahead, up-, and down-regulation prices.

4.1 Market revenue formulation

The total revenue is composed of two components:

$$R^D = \sum_t \lambda_t^D \hat{p}_t, \quad R^B = \sum_t \lambda_t^{\downarrow}(p_t - \hat{p}_t)^+ - \lambda_t^{\uparrow}(\hat{p}_t - p_t)^+ \tag{13}$$

where $(x)^+ = \max(x, 0)$. The overall profit, being the ultimate goal of trading, is therefore:

$$\Pi(\hat{p}) = \sum_t \left[\lambda_t^D \hat{p}_t + \lambda_t^{\downarrow}(p_t - \hat{p}_t)^+ - \lambda_t^{\uparrow}(\hat{p}_t - p_t)^+\right]. \tag{14}$$

when the plant underproduces, it must buy at the up-regulation price ($\lambda^{\uparrow} > \lambda^D$), while overproduction is remunerated at the lower down-regulation price ($\lambda^{\downarrow} < \lambda^D$).

4.2 Revenue-based evaluation

For each predictive model (Linear, Polynomial, WLS, Ridge/Lasso, and DFL), the hourly commitments $\hat{p}_t$ were used in Equation 14 to compute the realized revenue on the test set. Table 7 compares the total profits obtained by each model. It is noted that for regularized and WLS methods, the Ridge (L2) and 2D grid query setup are given.
The economic evaluation, illustrated by the cumulative profit curves in Figure 2, provides a different and more insightful picture than the RMSE comparison. While the baseline Linear, Regularized Ridge, and Polynomial Ridge models performed similarly in terms of profit, they all significantly underperformed the theoretical 'Perfect Info' benchmark. A noticeable improvement is seen with the Weighted Least Squares (WLS) model, suggesting that its adaptive, localized approach captures some economically valuable patterns missed by the global models. However, the most compelling result is the superior performance of the Decision-Focused Learning (DFL) model. Despite having the highest prediction error (RMSE), it generated the highest profit, tracking the 'Perfect Info' curve far more closely than any other model. This clearly demonstrates that optimizing directly for the economic objective, rather than for pure predictive accuracy, is the most effective strategy for optimizing revenue (Section 6).

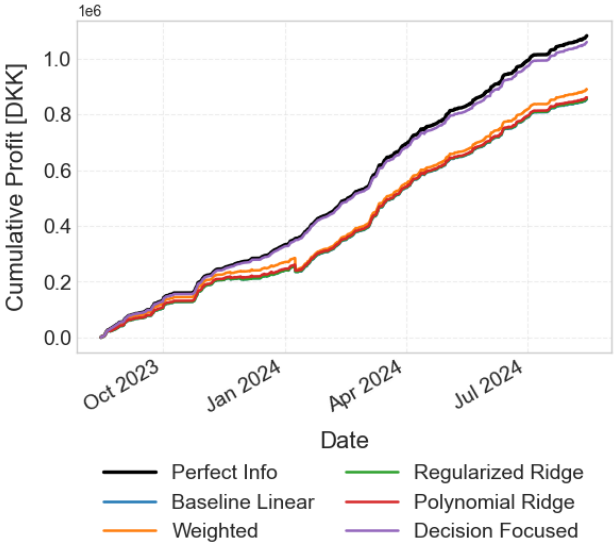
5 Unsupervised Learning

In this part of the project, we aimed to improve the accuracy of wind power forecasting by applying unsupervised learning techniques to segment the dataset into different operating regimes. Examples are high-wind versus low-wind periods, or winter versus summer days, where the relationship between inputs and output may differ. Based on these regimes, ridge regression models are fitted in order to improve accuracy.
Three main clustering approaches were applied: **K-Means Clustering**, which groups observations into k clusters by minimizing within-cluster variance; **Hierarchical Clustering**, which builds a tree of nested clusters without requiring k in advance; and **Gaussian Mixture Models (GMM)**, a probabilistic extension of K-Means that represents each cluster as a Gaussian distribution. Beyond the clustering algorithms themselves, several segmentation scenarios were tested to examine how different data representations affect model performance. These include clustering individual hourly observations based on all features, grouping by meteorological variables only (*Weather-Only*, both kmeans), enforcing temporal adjacency in the clustering process (*Temporal*, hierarchical), applying a probabilistic mixture

Table 7: Total Revenue Comparison Across Models (Test Period)

Model	**Total Profit [DKK]**
Perfect Information	1,082,036.68
Decision Focused	1,059,512.36
Weighted	889,956.24
Polynomial Ridge	861,018.43
Regularized Ridge	856,067.35
Baseline Linear	854,295.03
Secondary Models (Retrained)	
Ridge Retrain	886,777.02
OLS (squared)	886,377.52
Baseline OLS Retrain	881,807.67
Lasso Retrain	863,354.36

Figure 2: Cumulative Profit Comparison - Test Period



approach (*GMM*), and clustering full 24-hour power profiles (*Daily-Profile*). Each scenario represents a different assumption about how wind power generation patterns may vary across time or weather conditions. For each scenario method, a Ridge regression model was trained per cluster, and performance was compared against a global baseline using RMSE and R^2. Table 8 shows the results for the training and test set.

Table 8: Performance comparison of global and clustered Ridge regression models. Note that in contrast to the other model validation in this report, a simple holdout validation was performed.

Model	k	**RMSE (val)**	**RMSE (test)**	R^2 **(val)**	R^2 **(test)**
Daily-Profile Clustered Ridge	2	0.2412	**0.4544**	0.7809	**-0.1089**
GMM Clustered Ridge	7	0.2579	0.4711	0.7496	-0.1920
Global Ridge (baseline)	1	0.2621	0.4718	0.7413	-0.1956
Temporal Clustered Ridge	2	0.2620	0.4775	0.7415	-0.2247
K-Means Clustered Ridge	2	0.2638	0.4825	0.7380	-0.2504
Hierarchical Clustered Ridge	2	0.2632	0.4825	0.7391	-0.2505
Weather-Only Clustered Ridge	2	0.2566	0.4951	0.7521	-0.3164

While all clustering approaches achieve slightly lower validation errors (RMSE $\approx$ 0.24–0.26) compared to the global baseline (RMSE $=$ 0.262), most do not generalize well to the test set, where R^2 values remain negative. The *Daily-Profile Clustered Ridge* model again yields the best test performance (RMSE $=$ 0.454), improving upon the global Ridge (RMSE $=$ 0.472) by roughly 4%. This improvement arises because daily-profile clustering groups days by

their 24-hour power trajectories, effectively distinguishing high- and low-output regimes with characteristic diurnal shapes. Each local model can thus specialize in more homogeneous temporal patterns, enhancing generalization.
The *GMM Clustered Ridge* also performs comparably to the baseline, benefiting from soft probabilistic cluster assignments that smooth regime boundaries. In contrast, the remaining approaches, k-means, hierarchical, temporal, and weather-only, either overfit or create weakly separated clusters, fragmenting the data and degrading model stability. Overall, clustering provides only limited gains, with temporal (daily) segmentation and probabilistic mixture modeling showing the most promise.

6 Model 2: Decision-Focused Learning

While standard predictive models aim to minimize statistical error by predicting the day-ahead power output, in energy markets the true objective is to maximize operational profit. **Decision-Focused Learning (DFL)** bridges this gap by embedding the market decision problem directly into the learning process, so that model parameters are trained to improve financial outcomes rather than forecast accuracy. The mathematical formulation is derived by the following concept: At each hour t, the predictive model outputs a commitment $\hat{p}_t = f_\theta(z_t)$ given features z_t (forecasted wind and price signals). Instead of minimizing $\|\hat{p}_t - p_t^*\|^2$, DFL optimizes

$$\min_{\theta} \; -\Pi\big(f_\theta(z_t), p_t^{\text{real}}, \lambda_t^D, \lambda_t^{\uparrow}, \lambda_t^{\downarrow}\big), \tag{15}$$

where $\Pi(\cdot)$ is the profit function defined in Equation 14. This formulation ensures that the learned policy maximizes expected revenue.
In the framework of the current project, the following workflow was developed to derive results: **Oracle optimization**, where perfect-foresight commitments p_t^* are computed using Gurobi by maximizing Equation 14 with $p_t = p_t^{\text{real}}$; **Predict-then-Optimize baseline (OLS)**, which trains a linear regression model to approximate p_t^* from available features z_t and evaluates both its RMSE and profit gap relative to the oracle; and **Decision-Focused Learning**, where a linear model with parameters $(b_0, \mathbf{b})$ is optimized batch-wise via Gurobi to minimize the negative profit over each batch, directly aligning learning with the trading objective.

6.1 Predict-then-Optimize Results

In the Predict-then-Optimize scenario, although it achieved moderate statistical accuracy ($\text{RMSE}_{\text{test}} = 0.44$, $R^2_{\text{test}} = 0.046$), its economic performance reached only 92% of the oracle's profit 987.309 DKK vs. 1,073,309 DKK on the test set). The the model learns an average bidding strategy centered around $\hat{p}_t \approx 0.5$, not capturing the threshold-like responses of the oracle near 0 and 1 MW. This illustrates that despite the low prediction error does not necessarily translate into a poor trading performance.

6.2 DFL Joint training Results

The DFL model achieved an optimal set of coefficients:

$$-0.4064, \quad \mathbf{b} = [0.38, 0.03, -0.09, 8.58, -6.51, -3.69, -0.67, 0.01, -0.05, 0, 02, -0.007]$$

Table 9: Statistical and economic performance of the DFL model (placeholders in red).

Metric	**Train Value**	**Test Value**
RMSE	0.5333	0.5483
R^2	-0.3207	-0.4485
Profit (DKK)	7.72x10^6	1.07x10^6
Relative to oracle	99.6%	99.2%

Despite relatively poor predictive metrics (low R^2), the DFL achieved near-oracle profits on both training and test sets, illustrating that optimizing for *economic decisions* is more effective than minimizing statistical error.
The DFL model often predicts extreme commitments (near 0 or 1), reflecting learned threshold-like rules in response to price spreads ($\lambda^{\uparrow} - \lambda^{\downarrow}$). Although less accurate in terms of RMSE, its end-to-end training aligns with the real market objective and consistently closes the **decision gap** between predictive and optimal bidding strategies. The figures provided in the relative section of the Appendix A, display the above mentioned bidding performance of the two frameworks used and the underlying correspondence between DFL and oracle commitments

7 Conclusion

The results of this assignment show that increasing model complexity does not necessarily lead to improved predictive performance in the given wind power forecasting problem and using the particular static dataset. In particular, more complex models such as polynomial and regularized regressions did not yield significantly better results compared to simpler linear approaches, likely due to the intrinsic variability of wind power data. Furthermore, improved accuracy in power prediction does not directly translate into higher trading revenues for the stakeholder. Thus optimizing for statistical performance alone is insufficient for decision making regarding profit maximization under dynamic market uncertainty.
To address this, models that explicitly incorporate the economic objective, such as **Decision-Focused Learning (DFL)**, proved more effective, as they directly optimize revenue-related loss functions rather than purely predictive errors. Additionally, continuously updating the model by incorporating information from the most recent days (after they are realized - feedback effect) improved day-ahead forecasting accuracy. This approach resembles an *online learning* process with an interacting environment, suggesting that future extensions could leverage **reinforcement learning** to dynamically adapt trading strategies based on observed market responses. Such an approach will be further explored in Assignment 2.

A Appendix

Data Summary Statistics

	ts	power
count	43848	30367
mean	2022-07-02 11:30	0.49
min	2020-01-01 00:00	0.00
25%	2021-04-01 17:45	0.06
50%	2022-07-02 11:30	0.25
75%	2023-10-02 05:15	0.73
max	2024-12-31 23:00	2.61
std	–	0.57

Table 10: Summary statistics: general and power.

	mean_IGR_Nex	max_temp	mean_temp	min_temp	hum	prec	wind_speed	wind_dir
count	42095	42174	42158	42101	42200	42118	42121	42135
mean	132.25	10.14	9.68	9.22	82.58	0.07	4.93	195.03
min	0.00	-8.50	-8.90	-9.60	26.00	0.00	0.00	0.00
25%	0.00	4.80	4.40	4.10	74.00	0.00	2.70	111.00
50%	6.00	9.60	9.20	8.70	85.00	0.00	4.50	229.00
75%	184.00	15.40	15.00	14.40	93.00	0.00	6.60	265.00
max	958.00	30.80	30.00	28.40	101.00	20.60	20.10	360.00
std	214.37	6.54	6.46	6.40	13.14	0.41	2.88	95.56

Table 11: Summary statistics: weather variables.

	DK2_DKK	SYSTEM_DKK	ImbMWh	ImbPriceDKK	BalancingUp	BalancingDown
count	43847	43847	18921	43844	40555	40555
mean	712.19	448.69	-120.93	711.65	817.20	619.09
min	-447.46	-222.73	-1265.00	-16392.20	-447.43	-16392.20
25%	233.25	120.84	-224.90	178.84	261.19	156.15
50%	508.68	299.15	-105.60	455.62	556.13	400.00
75%	871.06	582.44	-16.30	882.76	990.32	788.68
max	6982.64	5258.05	1128.60	10000.67	10000.67	6477.19
std	753.19	518.82	216.67	867.34	890.80	738.74

Table 12: Summary statistics: market variables.

OLS Regression Results

	RMSE	MAE	R^2	nRMSE (%)
Train set	0.372	0.273	0.533	14.59

Table 13: Performance metrics of the OLS regression model used for imputing missing wind production.

Variable	Coef.	Std. Err.	t	P> |t|	[0.025, 0.975]
const	0.4623	0.002	187.43	0.000	[0.457, 0.467]
mean_IGR_Nex	-0.0589	0.003	-17.26	0.000	[-0.066, -0.052]
max_temp_Nex	0.5290	0.087	6.08	0.000	[0.358, 0.700]
mean_temp_Nex	-0.2238	0.159	-1.41	0.159	[-0.535, 0.088]
min_temp_Nex	-0.2565	0.084	-3.06	0.002	[-0.421, -0.092]
mean_hum_Nex	0.0422	0.003	13.59	0.000	[0.036, 0.048]
prec_Nex	0.0044	0.002	1.81	0.071	[-0.000, 0.009]
mean_wind_speed_Nex	0.4192	0.003	154.85	0.000	[0.414, 0.425]
mean_wind_dir_Nex	-0.0175	0.003	-6.92	0.000	[-0.022, -0.013]

Table 14: OLS regression coefficients and significance levels.

Missing Data Summary

Variable	Interpolated gaps	Interpolated avg (h)	Interpolated total (h)	Removed gaps	Removed total (h)
mean_IGR	260	2.29	595	19	424
max_temp	252	2.29	578	18	413
mean_temp	307	2.23	684	18	413
min_temp	311	2.25	699	18	414
mean_hum	239	2.35	561	17	407
prec	248	2.33	578	18	418
mean_wind_speed	257	2.28	587	19	424
mean_wind_dir	262	2.30	602	18	414

Table 15: Summary of weather data gap handling during the training/validation period.

Variable	Interpolated gaps	Interpolated avg (h)	Interpolated total (h)	Removed gaps	Removed total (h)
mean_IGR	138	2.34	323	7	465
max_temp	125	2.27	284	7	465
mean_temp	113	2.28	258	7	465
min_temp	127	2.28	289	7	465
mean_hum	118	2.29	270	7	465
prec	136	2.33	317	7	465
mean_wind_speed	133	2.33	310	7	465
mean_wind_dir	131	2.31	303	7	465

Table 16: Summary of weather data gap handling during the test period.

Gaps filled	Avg gap length (h)	Min gap length (h)	Max gap length (h)	Total hours
70	29.46	1	405	2,062

Table 17: Summary of wind power gap filling using OLS regression.

Market Data Gaps (Training Period)

Column	Missing values	Gaps	Avg gap length (h)	Min (h)	Max (h)	Total missing (h)
PriceArea	1	1	1.0	1	1	1
ImbalancePriceDKK	1	1	1.0	1	1	1
BalancingPowerPriceUpDKK	25	2	12.5	1	24	25
BalancingPowerPriceDownDKK	25	2	12.5	1	24	25

Table 18: Market data gaps during training/validation period.

Column	Missing values	Gaps	Avg gap length (h)	Min (h)	Max (h)	Total missing (h)
PriceArea	0	0	0.00	0	0	0
ImbalancePriceDKK	0	0	0.00	0	0	0
BalancingPowerPriceUpDKK	31	3	10.3	1	28	31
BalancingPowerPriceDownDKK	31	3	10.3	1	28	31

Table 19: Market data gaps during test period.

Set	Original rows	Removed rows	Remaining (%)
Train/Validation	17,520	1,042	94.1
Test	8,784	8,079	92.0

Table 20: Overall dataset reduction after preprocessing.

Model 1: Power prediction models

This following subsections present the predictive effect of wind power forecasting based on the implemented baseline regression, , non-linear models, weighted local regression and regularized regression. Relevant Figures are provided for each particular model.

Baseline Linear Regression

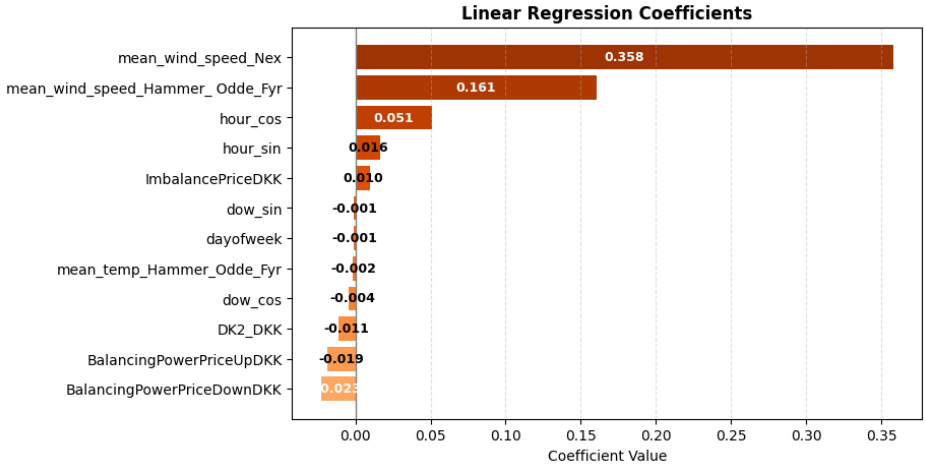


Figure 3: Linear Regression model coefficients. Wind speed features (`mean_wind_speed_Nex` and `mean_wind_speed_Hammer_Odde_Fyr`) have the strongest contribution to predicted power, while market-related variables show minor influence.

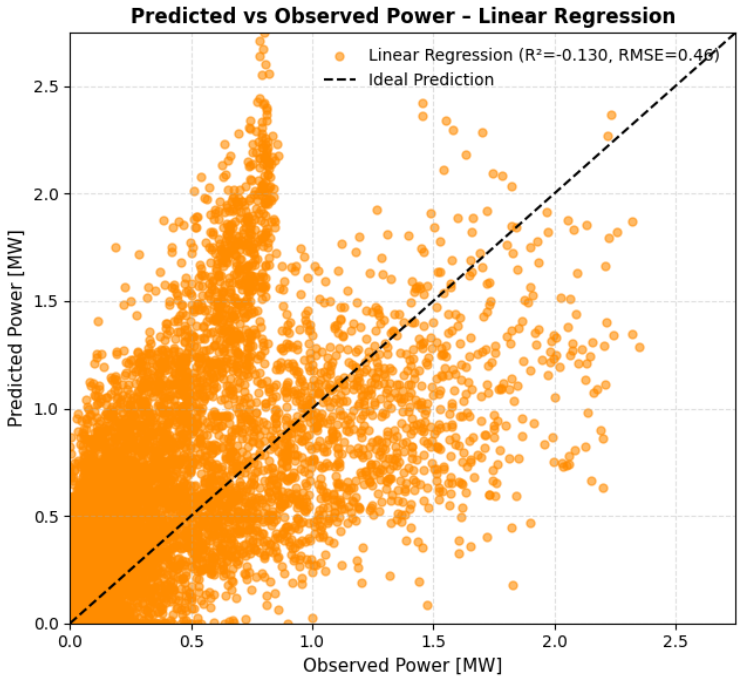


Figure 4: Predicted vs. observed power for the Linear Regression model on the test set. The dashed line represents the ideal prediction ($y = x$). The model demonstrates limited accuracy in capturing high variability in wind power production.

Non-Linear Regression (Polynomial Ridge)

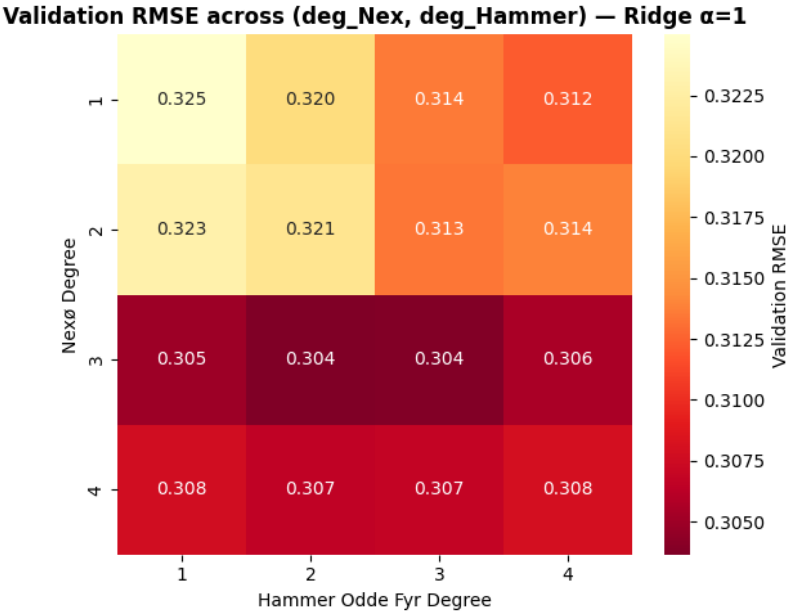


Figure 5: Validation RMSE across polynomial degree combinations for NexÃ¸ and Hammer Odde Fyr wind speeds ($\alpha = 1$). The optimal combination corresponds to (Nex$^\circ = 3$, Hammer$^\circ = 2$).

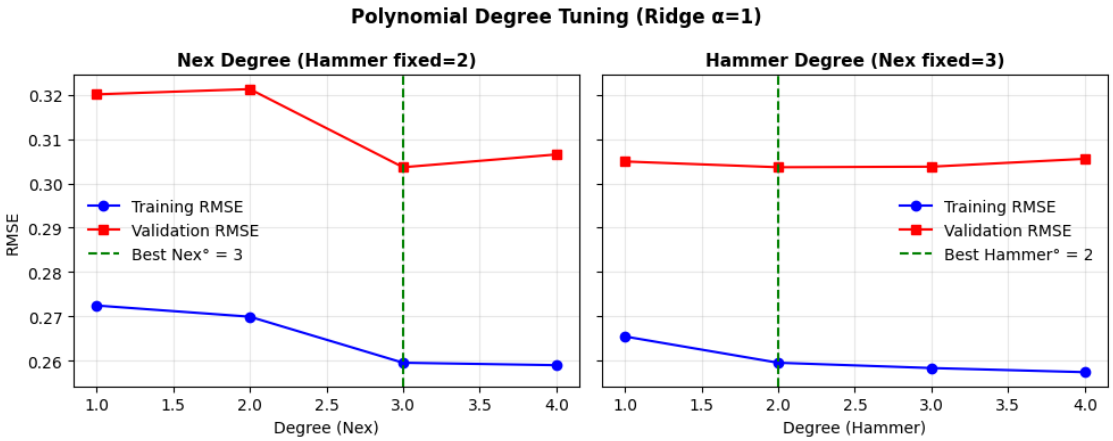


Figure 6: Polynomial degree tuning for Ridge regression ($\alpha = 1$). The plots show the trade-off between bias and variance across polynomial degrees for Nexo and Hammer Odde Fyr wind speeds. The chosen degrees are Nex$^\circ = 3$ and Hammer$^\circ = 2$.

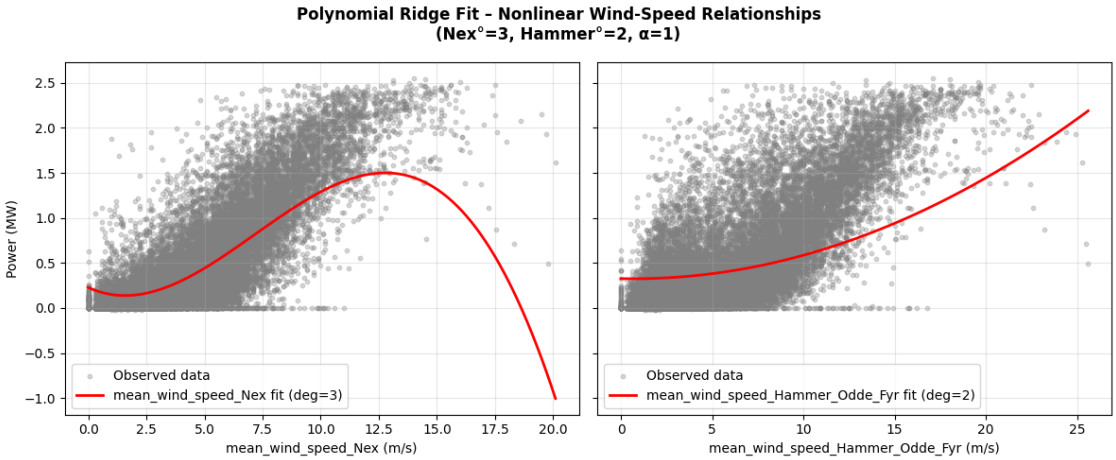


Figure 7: Polynomial Ridge model fits for wind-power relationships at Nexo and Hammer Odde Fyr.

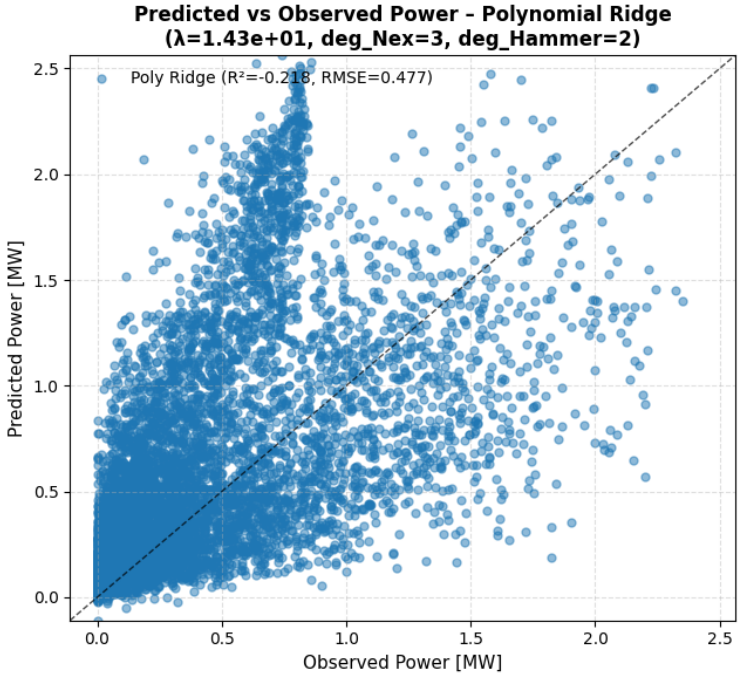


Figure 8: Predicted vs. observed power for the Polynomial Ridge model ($\lambda = 1.43 \times 10^1$, Nex$^\circ = 3$, Hammer$^\circ = 2$).

Weighted Local Regression (WLS)

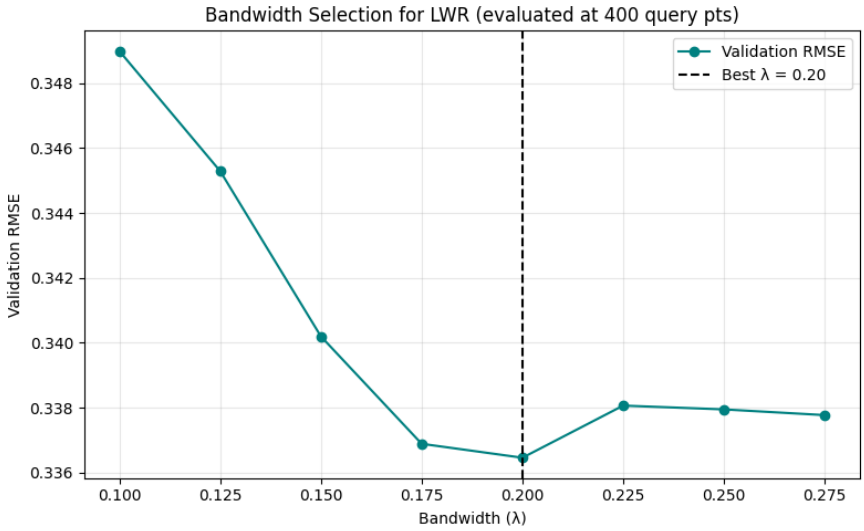


Figure 9: Bandwidth selection for the 2D Locally Weighted Regression model. The optimal bandwidth $\lambda = 0.20$ minimizes validation RMSE, balancing bias and variance.

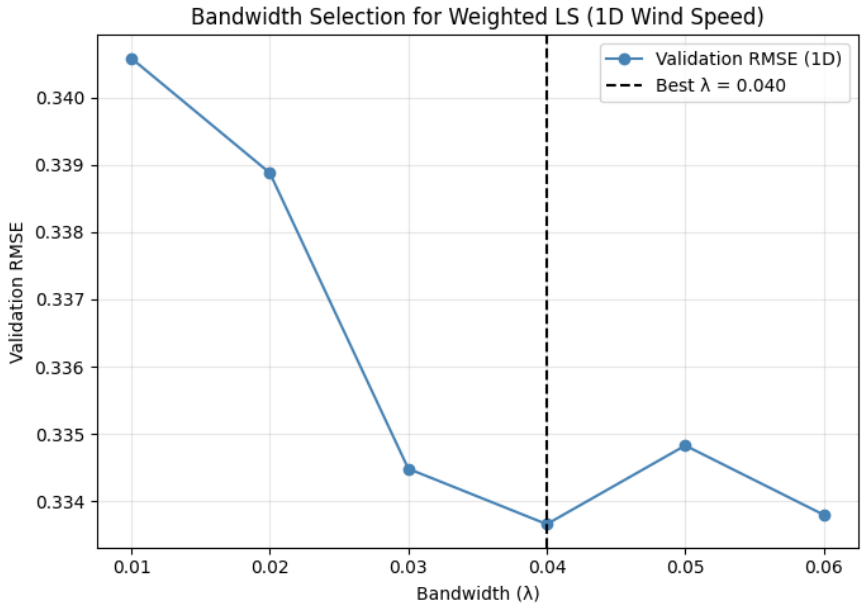


Figure 10: Bandwidth selection for 1D Weighted Least Squares applied to mean wind speed. The best smoothing parameter is $\lambda = 0.04$.

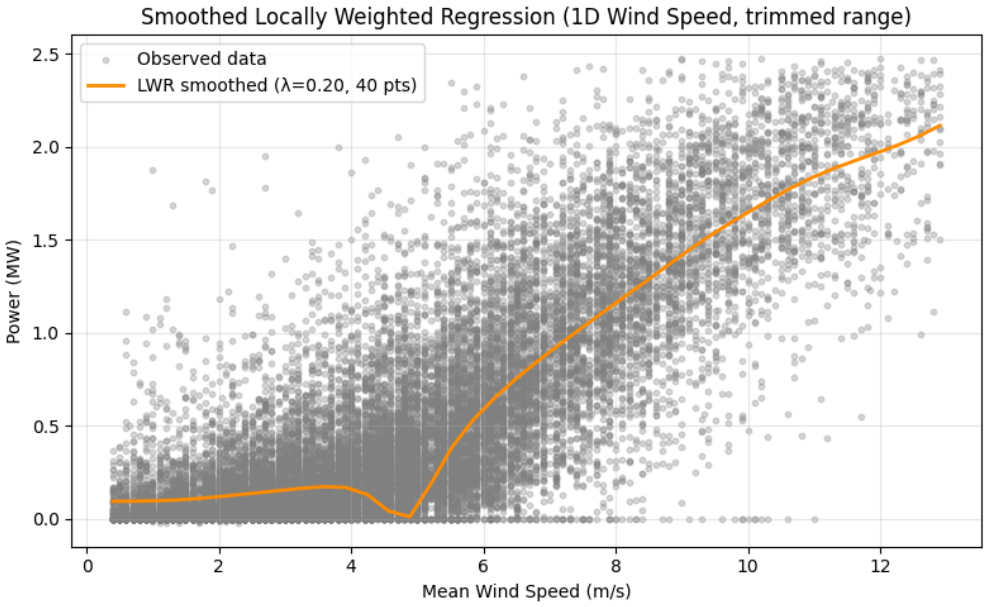


Figure 11: Smoothed 1D LWR fit ($\lambda = 0.20$) for mean wind speed. The curve captures the saturation effect of power output with increasing wind speed.

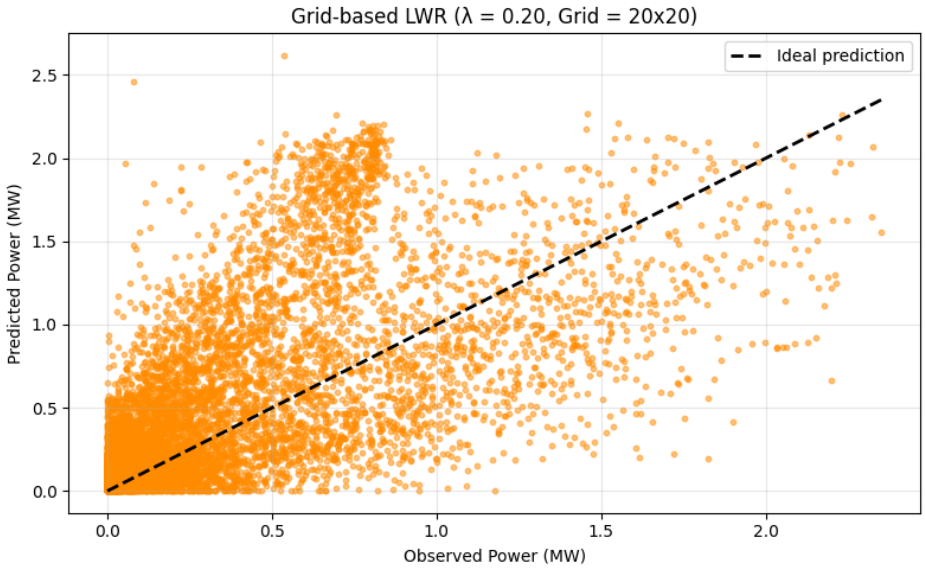


Figure 12: Predicted vs. observed power for the grid-based LWR model ($\lambda = 0.20$, grid=20×20). The dashed line represents the ideal prediction. While local models improve flexibility, extreme regions remain challenging to estimate.

Regularized Regression (Ridge & Lasso)

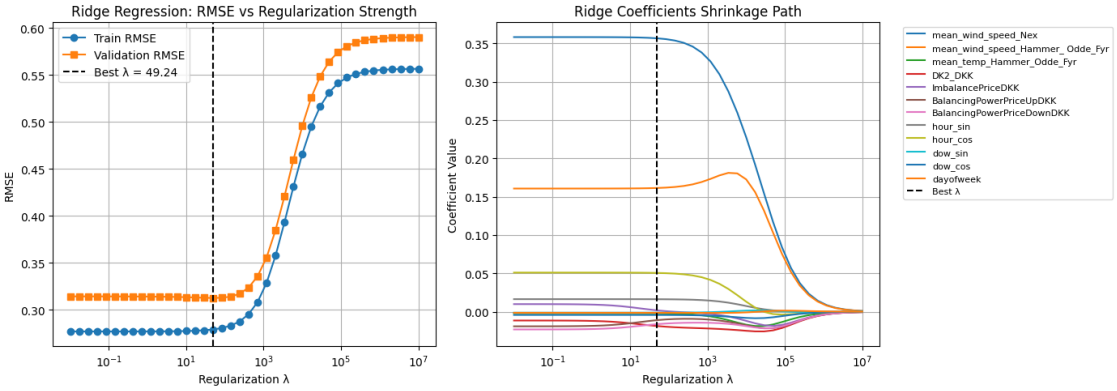


Figure 13: Ridge Regression: (Left) RMSE as a function of regularization strength λ on train and validation splits; (Right) coefficient shrinkage path showing how feature weights are reduced as λ increases.

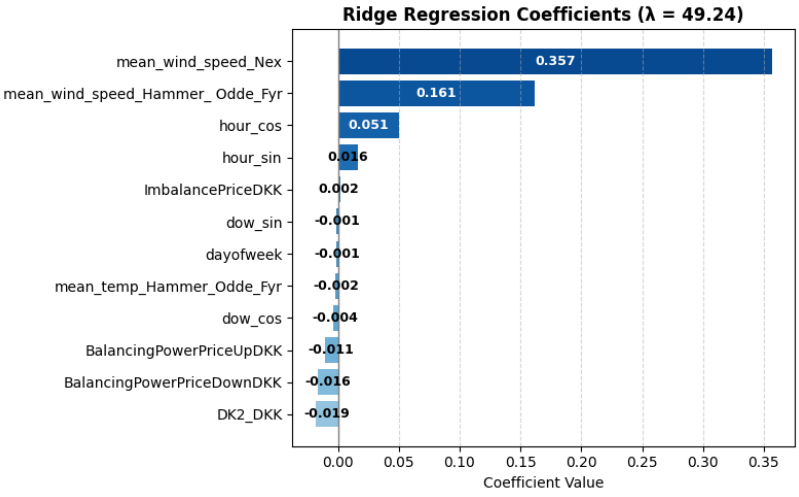


Figure 14: Final Ridge Regression coefficients at optimal $\lambda = 49.24$. Wind speed features dominate, while most market and temporal variables show minor influence due to regularization.

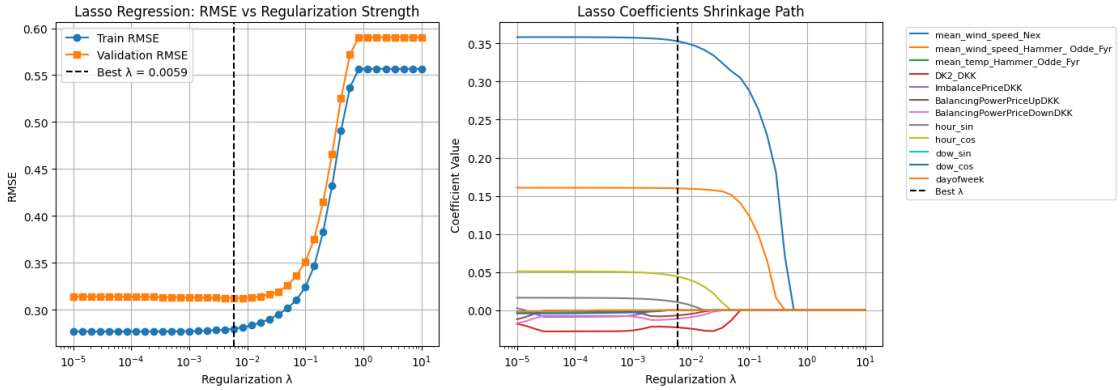


Figure 15: Lasso Regression: (Left) RMSE as a function of regularization strength λ on train and validation splits; (Right) coefficient shrinkage path showing variable selection as λ increases.

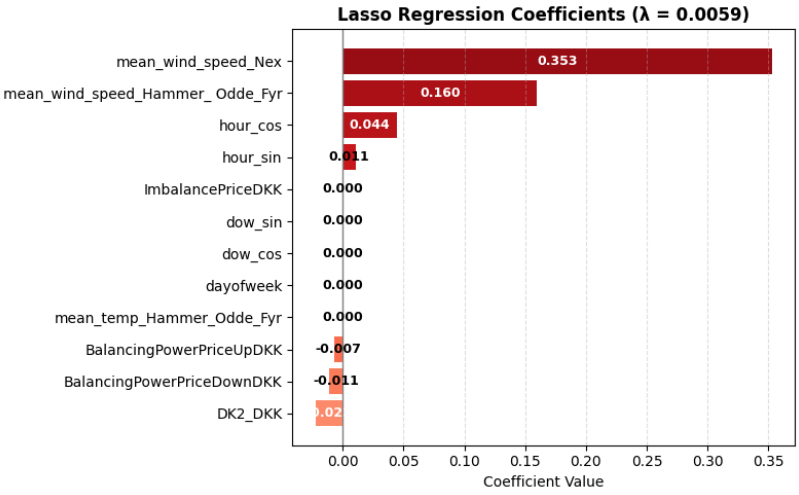


Figure 16: Final Lasso Regression coefficients at optimal $\lambda = 0.0059$. Only a few dominant predictors (notably wind speed and temperature) remain non-zero, confirming the model's sparsity-inducing behavior.

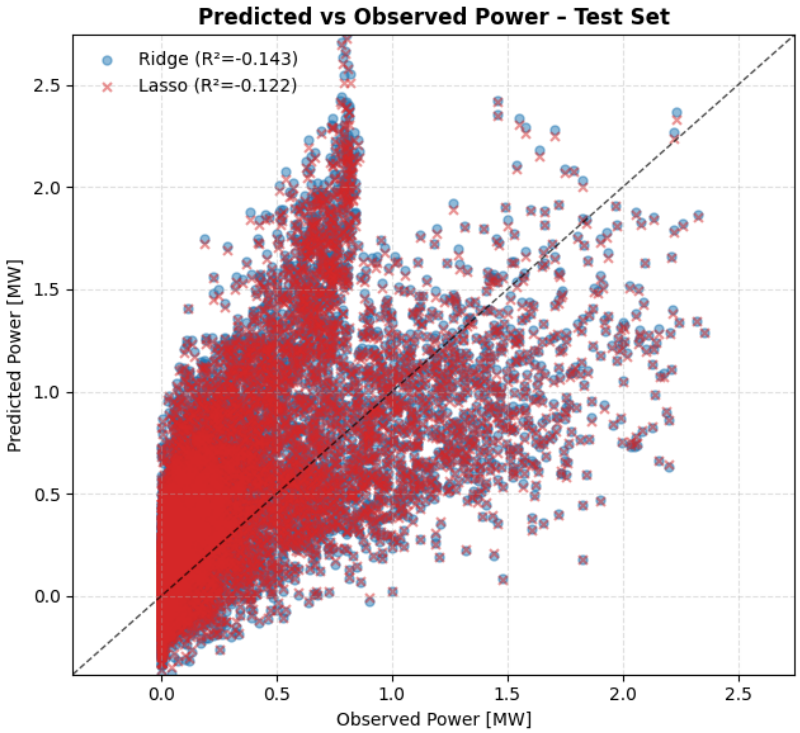


Figure 17: Predicted vs. observed wind power on the test set for Ridge and Lasso models. The dashed line indicates the ideal 1:1 relationship.

Model 2: Decision Focused Learning

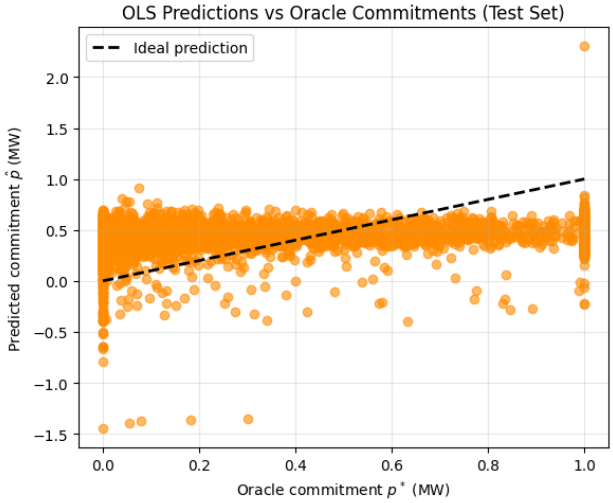


Figure 18: OLS predictions versus oracle commitments on the test set. In the predict then optimize framework, the OLS model predicts commitments $\hat{p}$ produced by the optimization model.

Figure 19: Decision-Focused Learning (DFL) versus oracle commitments. The DFL model directly optimizes the decision-making objective (expected profit) during training. A bid-no bid strategy is observed.