Assignment 1

Machine Learning for Energy Systems

Authors

Zoe Elsa Wieser - s233239
Nicole Khai Wen Wong - s242088
Daksh Jain - s241875

October 30, 2024

Contents

1 Introduction

With the current transition towards renewable energy sources instead of the traditional fossil fuels, the landscape of electricity markets have been reshaped, particularly in the context of wind power generation. In order to maximise revenue and ensure grid stability, the ability to trade in day-ahead and balancing markets become of great value.

This assignment will apply mathematical modelling techniques to optimize trading strategies through the perspective of a wind farm owner, leveraging on historical data of a wind farm to enhance decision making in energy trading.

In this study, two distinct models for trading is developed. The first model employs regression techniques to forecast wind power production, which is then applied to address the decision making problem. In the second model, regression is employed directly to identify the most effective offering strategy.

By exploring the effects of different modeling techniques and their performance, the aim is to identify the most effective strategy for trading in renewable energy markets, enhancing the understanding of market dynamics and optimizing trading strategies.

The findings from this assignment will also provide actionable insights for enhancing trading performance in the rapidly evolving landscape of renewable energy markets. Through analysis and comparison of the developed models, this study can inform future trading strategies.

2 Trading optimization model

The trading optimization model aims to maximize the revenue generated from power bidding in the day-ahead market while considering the potential for adjustments in real-time operations. The optimization problem can be formulated as follows:

$$\max_{\hat{p}_t, \Delta_t, \Delta_t^{\uparrow}, \Delta_t^{\downarrow}} \quad \sum_{t=1}^{24} \left(\lambda_t^D \cdot \hat{p}_t + \lambda_t^{\downarrow} \cdot \Delta_t^{\downarrow} - \lambda_t^{\uparrow} \cdot \Delta_t^{\uparrow} \right) \tag{1}$$

$$\tag{2}$$

$$\begin{aligned}
\text{s.t.} \quad & 0 \leq \hat{p}_t \leq P_{\max} && (3) \\
& \Delta_t = \hat{p}_t - p_t && (4) \\
& \Delta_t = \Delta_t^{\uparrow} - \Delta_t^{\downarrow} && (5) \\
& \Delta_t^{\uparrow} \geq 0 && (6) \\
& \Delta_t^{\downarrow} \geq 0 && (7)
\end{aligned}$$

Decision variable:
$\hat{p}_t$: representing the power bid in the day-ahead market.

Parameters:
p_t: actual power production (assumed to be a perfect forecast)
λ_t^D: market price in the day-ahead market
$\lambda_t^\downarrow$: down-regulation price in the balancing market
$\lambda_t^\uparrow$: up-regulation price in the balancing market

The objective function seeks to maximize the total revenue over a 24-hour period by considering the income from the day-ahead market and the balancing market. The first term in the objective function calculates revenue from the day-ahead market based on the submitted power bid $\hat{p}_t$, while the second and third terms account for the adjustments in power output, represented by the down-regulation and up-regulation prices, respectively.

The constraints ensure that the power bid is non-negative and does not exceed the maximum production capacity $P_{\max}$. The variables track the differences between the bid power, actual power production, and adjustments made for balancing purposes.

3 Data Presentation

The data utilized for this assignment is sourced from *Energy Data DK*, specifically focusing on active power from a wind farm in Kalby, Bornholm. Additionally, supplementary features are gathered from various sources: *DMI* provides geological data for the area, *Energinet* supplies spot price data for the DK2 market, and *The Wind Power* offers a power curve for the Vestas V80/2000 Wind Turbine, which closely resembles the actual turbine installed at Kalby.

3.1 Features

The chosen features used are mean wind speed, mean wind power, and wind speed. These are chosen due to their high correlation with the wind production at Kalby. The correlation is found using the *.corrwith* command from the Pandas library. The chosen features yielded a correlation of 0.82 to 0.85, completely surpassing the other possible features, whom were all in the correlation range of -0.30 to -0.35. The plot can be found in Appendix A.1.

The dataset is then split up into training and testing data using a constructed method split_data, which takes a DataFrame, a target, a list of features, and a test-split as an input. It then uses this information to run the *train_test_split* function from the sklearn.model_selection library. This returns the split up feature and target data as x_train, y_train, x_test, and y_test.

Z-score standardization is employed to scale the data such that it has a mean of 0 and a standard deviation of 1, using the *StandardScaler* class from the *sklearn.preprocessing* module. With no significant outliers, this standardization could enhance model performance and interpretability.

4 Linear Regression

In this assignment, regression models are trained to be used for prediction. The implementation of linear regression is first explored following the construction of the training dataset.

For a linear regression model, the equation can be written as:

$$y = X \cdot \theta + \epsilon \tag{8}$$

where:

- y is the vector of target values
- X is the matrix of input features
- ϵ is the bias term

The aim of linear regression is to establish a linear equation that best fits the given data. To estimate the parameters θ, the cost function needs to be optimized, which quantifies the difference between the predicted values and the actual target values. The cost function can be defined as:

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 \tag{9}$$

where:

- m is the number of training samples
- $h_\theta(x^{(i)})$ is the predicted value for the i-th training sample
- $y^{(i)}$ is the actual target value for the i-th training sample

There are 2 primary approaches to solve the linear regression problem - gradient descent algorithm and the closed form solution.

Firstly, a subset of 100 samples is taken from the training data to ensure a manageable dataset. The feature chosen for initial testing is mean wind speed, with data being split into training and testing datasets. A column of ones is added to the training data to account for the intercept term in the linear regression equation.

4.1 Gradient descent algorithm

The gradient descent function follows these steps:

1. Gradient calculation

$$\text{gradient} = -\frac{2}{m} \cdot X^T \cdot (y - X \cdot \theta) \tag{10}$$

2. Parameter update

$$\theta_{new} = \theta - \alpha \cdot \text{gradient} \tag{11}$$

where:
m: number of training samples in dataset
X: feature matrix where each row represents a training sample and each column represents a feature
X^T: transpose of the matrix X
y: target vector
θ: the parameter vector that is to be optimized during the training process
α: the learning rate, a hyperparameter that determines the size of the steps taken towards the minimum of the cost function

Initial parameters are randomly initialized, and the algorithm iteratively updates these based on the computed gradient until convergence or the maximum number of iterations is reached.

4.2 Closed form solution

On the other hand, the closed form solution computes the parameters by setting the gradient of the cost function with respect to zero and solving for θ, given by:

$$\theta = (X^T X)^{-1} X^T y \tag{12}$$

4.3 Results for 2 methods

The results for the gradient descent algorithm (GD) and closed form solution (CF) are collated below:

$$\theta_{\text{GD}} = \begin{bmatrix} -1029.89 \\ 669.56 \end{bmatrix}, \quad \theta_{\text{CF}} = \begin{bmatrix} -1177.77 \\ 704.43 \end{bmatrix}$$

With the same dataset, the solutions are within 15% of each other and follow a similar pattern where the magnitudes are aligned.

4.4 Increased samples

To improve the accuracy of the prediction, the number of samples are increased to get a better representation of the data, reduce noise impact and variance. In this case, the closed form solution gives a direct way to compute the optimal parameters of the linear regression without iterative processes. The results are:

$$\theta_{\text{CF, }1} = \begin{bmatrix} -436.06 \\ 437.14 \end{bmatrix}$$

4.5 Model verification

The root mean squared error (RMSE), mean absolute error (MAE) and R-squared values are computed for each model, giving a metric to assess the performance of the model on the test data. The closed form solution model results is verified against the testing dataset, with results collated in Table 1.

Metrics	Values
RMSE	787.3404896324633
MAE	578.9145189737654
R-squared	0.7854927869935326

Table 1: Linear regression model results

5 Non-linear regression

As an extension of the linear regression model, a nonlinear regression model was tested. In real-world scenarios, data often exhibit complex, nonlinear patterns that a linear model may oversimplify by assuming a constant rate of change. By introducing nonlinear regression, the model gains flexibility, allowing it to capture higher-order interactions and better represent the true underlying relationships in the data.

The linear regression model is transformed to a nonlinear model by introducing polynomial features. This includes the addition of higher degree terms of the original feature.

Transitioning from linear regression to non-linear regression is done using *PolynomialFeatures* from the *sklearn.preprocessing* module which creates new feature sets of the input feature *'mean_wind_speed'* by raising the original features to different degrees. The continued use of *LinearRegression()* fits the transformed features linearly, capturing non-linear relationships in the data.

Similarly, RMSE, MAE and R-squared values are computed for each model. The results are illustrated in Figure 1, where degree 1 appears to yield the best fit. The RMSE, MAE and R-squared values seem to remain relatively constant after polynomial degree 6, indicating overfitting after polynomial degree of 6. RMSE and MAE quantify how well the model's predictions match the actual data. If these values remain constant after a certain polynomial degree, it suggests that increasing the complexity of the model does not lead to improved predictive accuracy on the test data. Additionally, R-squared measures the proportion of variance in the target variable that can be explained by the features. If R-squared values plateau, it indicates that the model is not explaining any additional variance with more complexity.

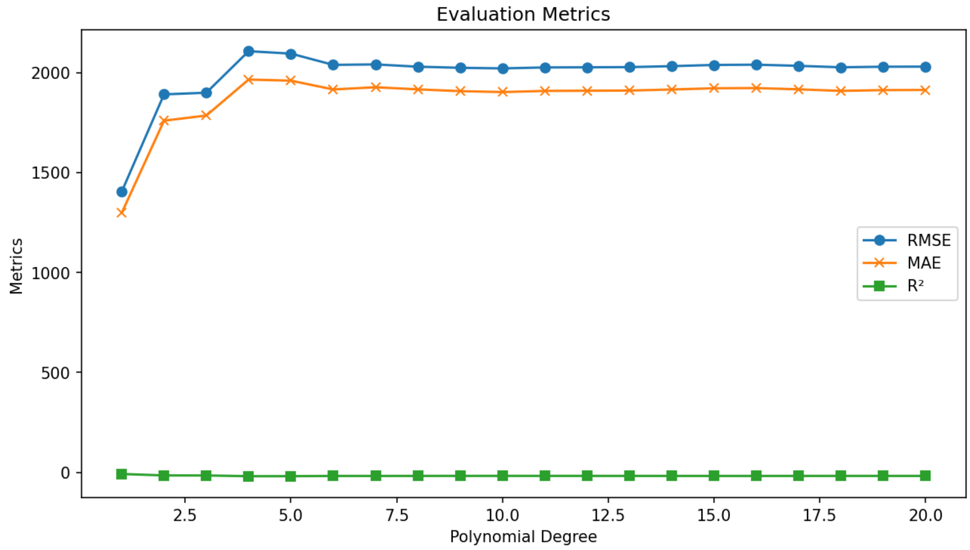

Figure 1: Evaluation metrics for different polynomial degrees

5.1 Weighted linear regression

From the linear regression model, a weighted linear regression model using the Epanechnikov kernel is implemented, to define weights based on the distances between training and test data points. The Epanechnikov kernel function calculates weights for each training point relative to a test point, where closer points receive higher weights. For each test data point, the distances to all training points are computed, and the corresponding weights are calculated. These weights are used to compute the regression coefficients (theta) via weighted least squares, used to predict the target variable at the test points. The model's performance is evaluated using RMSE, MAE, and R-squared metrics.

5.2 Comparison of models

Table 2 presents a summary of the performance metrics for the different models for comparison.

	RMSE	MAE	R-squared
Linear regression	787.34	578.91	0.79
Non-linear regression	1403.93	1297.51	-7.94
Weighted regression	1845.17	1723.02	-14.44

Table 2: Comparison of Model 1 regression results

The incorporation of nonlinear features allows the model to capture complex patterns within the data that a linear model may overlook. However, the results indicate that the non-linear regression model performed poorly, with a high RMSE and negative R-squared value. Similarly, the performance of the weighted regression model fell short, suggesting that increased complexity did not translate into improved performance.

Overall, while adding non-linear features has the potential to enhance model accuracy by capturing more complex relationships, the specific non-linear model tested did not yield better results than linear regression. Moreover, the locally weighted regression approach did not provide substantial improvements over linear regression either.

6 Regularization

Regularization is used in machine learning to prevent overfitting and improve generalization to unseen data. It does so by adding a penalty term to the loss function, also called the regularization term. In this section, two methods of regularization: Lasso (L1) and Ridge (L2) regularization are explored. Each method uses a parameter that controls the strength of the regularization, which will be fine-tuned based on the performance of the model on the testing set.

6.1 Lasso regularization (L1)

Lasso regularization, or L1 regularization, is a technique that encourages sparsity in the model. This means that Lasso can shrink some model weights to zero, thereby getting rid of irrelevant or redundant features. It will effectively simplify the model and improve the feature selection.

The algorithm adds a penalty term corresponding to the sum of the absolute values of the weights.

$$\widehat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \sum_i \epsilon_i^2 = \arg\min_{\boldsymbol{\beta}} \sum_i \left(y_i - \boldsymbol{\beta}^T \boldsymbol{x}_i\right)^2 + \lambda \sum_j |\boldsymbol{\beta}_j| \tag{13}$$

Since there is no closed-form solution for this problem, new linear constraints were implemented such that the quadratic linear constrained optimization problem is optimized. The new formulation is described as follows:

$$\min_{\boldsymbol{\beta},\ \boldsymbol{\beta}_{aux}} \quad \sum_i \left(y_i - \boldsymbol{\beta}^T \boldsymbol{x}_i\right)^2 + \lambda \sum_j \boldsymbol{\beta}_{aux,j} \tag{14a}$$

$$\text{s.t} \quad -\boldsymbol{\beta}_{aux} \leq \boldsymbol{\beta} \leq \boldsymbol{\beta}_{aux} \tag{14b}$$

$$\boldsymbol{\beta}_{aux} \geq 0 \tag{14c}$$

The new set of constraints ensures that $\boldsymbol{\beta}_{aux}$ takes the absolute value of $\boldsymbol{\beta}$. This optimization problem will then be used to include L1 regularization in the linear and non-linear models.

6.2 Ridge regularization (L2)

Ridge regularization, or L2 regularization, is a technique that discourages large model weight values. This means that model weights will be shrinked towards 0, while remaining non-null. As a result, Ridge regularization typically retains all features in the model but reduces their impact, allowing for a more stable and generalized solution.

The algorithm adds a penalty term corresponding to the sum of the squared weights (15). The problem admits a closed-form solution (16) which will be used to include L2 regularization in the linear and non-linear models.

$$\widehat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \sum_i \epsilon_i^2 = \arg\min_{\boldsymbol{\beta}} \frac{1}{2} \sum_i \left(y_i - \boldsymbol{\beta}^T \boldsymbol{x}_i\right)^2 + \frac{\lambda}{2} \|\boldsymbol{\beta}\|^2 \tag{15}$$

$$\left(\boldsymbol{X}^T \boldsymbol{X} + \lambda \boldsymbol{I}\right)^{-1} \boldsymbol{X}^T \boldsymbol{y} \tag{16}$$

6.3 Results for linear model

First, the regularization on the linear model is implemented. To find the optimal level of regularization, different values of λ are experimented with, each time computing the corresponding RMSE. This results in a plot representing the RMSE as a function of λ in Figure 2. From this plot, the optimal values for the regularization parameter can be deduced: λ as 1.9 and 572 for L1 and L2 regularization, respectively. They represent the optimal strength of regularization in our linear model.

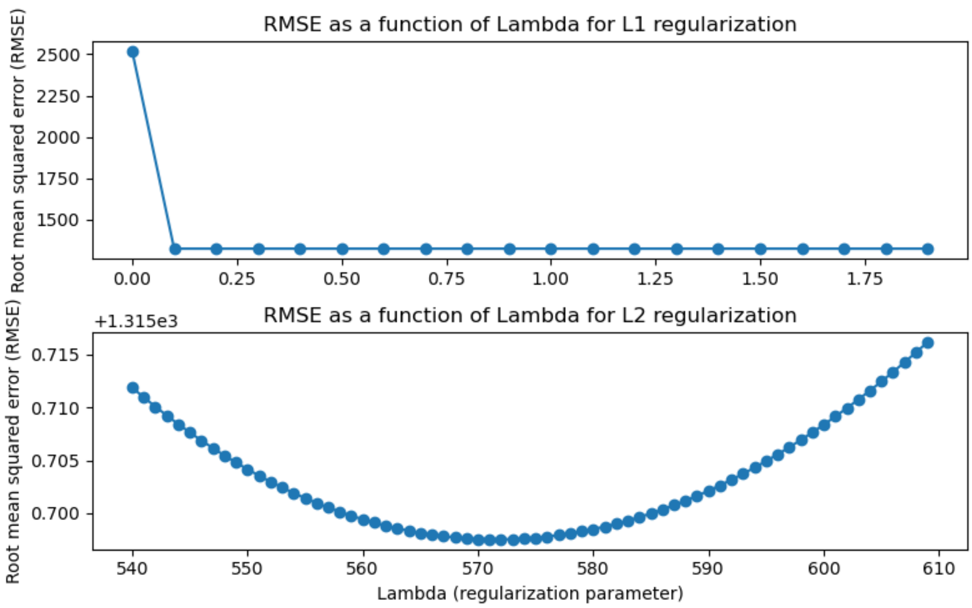

Figure 2: RMSE as a function of the regularization parameter

Once the regularization parameters are chosen for each method, the regularized model is evaluated. Table 3 shows how the usual evaluation metrics (Root mean squared error, Mean absolute error, R-squared) change from the linear regression model to the L1 or L2-regularized model, while Table 4 shows the corresponding weights for each feature.

Overall, L1-regularization has a negligible effect on our linear model. It can be interpreted that since L1 aligns closely with the original model, no feature selection was needed. The influence was negligible, likely because there was no irrelevant or redundant feature.

L2-regularization results in a higher MAE, likely due to the regularization penalty term. However, it improves RMSE and R-squared, suggesting that L2 will smooth out the predictions better than L1 and will give a better model fit. As for the features, weights are shrunk while remaining non-zero since the model distributes the penalty among all features. Here, all weights are decreased to promote a more stable model.

In this case, since all our features were relevant, L2 regularization has proven to be more suitable.

	Linear	**L1-regularized**	**L2-regularized**
MAE	924.990201	924.990261	951.117173
RMSE	1324.939629	1324.939564	1315.697545
R-squared	0.476335	0.476335	0.483615

Table 3: Evaluation metrics of linear models

Features	**Linear**	**L1-regularized**	**L2-regularized**
1	-1089.753998	-1089.753631	-771.048526
Mean wind speed	289.792468	289.791040	219.181579
Max wind speed 10min	-316.144780	-316.142963	-173.694006
Max wind speed 3sec	380.072022	380.071620	292.157705

Table 4: Feature weights of linear models

6.4 Results for non linear model

Then, we implement the regularization on our non-linear model. We also optimize the strength of both regularization models (Figure 3). The resulting optimal values for the regularization parameter are found to be λ as 0.01 and 13 for L1 and L2 regularization respectively. In Table 5 and Table 6 (the non-truncated version can be found in the Appendix A.2) are shown the corresponding evaluation metrics and weights.

The non-linear model shows the lowest MAE indicating better performance than both L1 and L2 regularized model. It can suggest that regularization will smooth out the data, capturing less complexity. On the contrary, L1 and L2 shows lower RMSE, indicating slights improvements of the non linear model.

L1 regularization encourages sparsity in the model as it drastically decreases some weights (feature 6) and drastically increases others (features 0, 1). It suggests that some features may be redundant or irrelevant, which is to be expected considering the construction of our non-linear model. Indeed, each feature represents a degree 3 combination of the original features.

L2 regularization seems to reduce the influence of most features, while staying conservative. Though, its behaviour is less typical than it should. That may be because of the irrelevance of some features, since L2 regularization is not commonly good at feature selection.

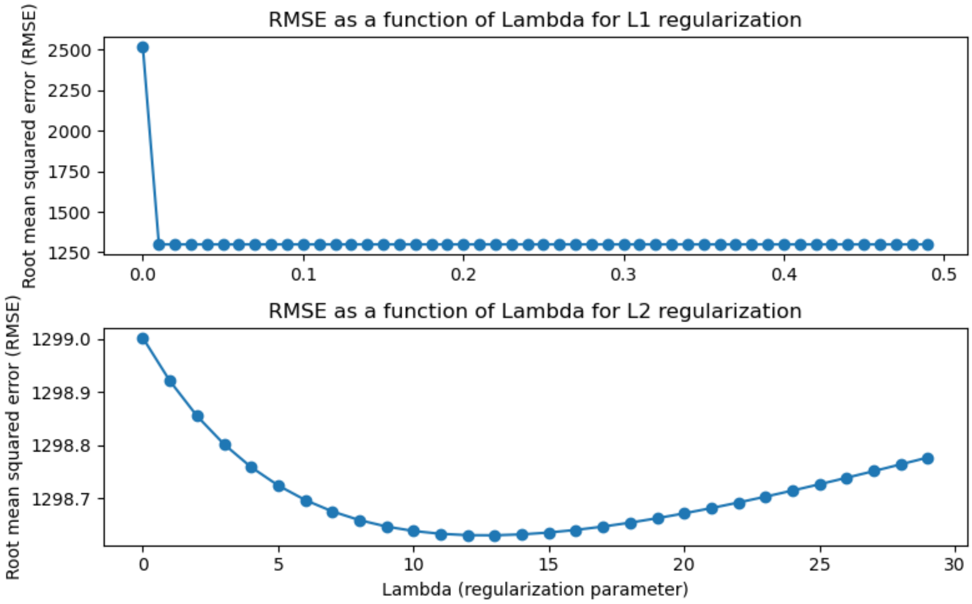

Figure 3: RMSE as a function of the regularization parameter

	Non-Linear	L1 Regularized	L2 Regularized
MAE	814.939662	817.799258	816.949700
RMSE	1301.251597	1299.001810	1298.631123
R-squared	0.494892	0.496637	0.496924

Table 5: Evaluation metrics of non-linear models

Feature	Weights	L1 Regularized	L2 Regularized
0	-652.506127	-928.979177	-624.891572
1	575.696937	828.655629	507.198942
2	-253.245596	-137.042100	-97.309615
3	-47.307781	-94.811435	-1.312067
4	153.358444	229.559313	88.744426
5	98.414295	150.533988	82.170816
6	36.006065	2.080707	49.883383
⋮	⋮	⋮	⋮
18	-11.958803	-11.575192	-11.100896

Table 6: Feature weights of non-linear models (polynomial degree 3)

7 Revenue Calculation

The total revenue generated by each regression model will be assessed based on the power selected for bidding into the system at each time step. To estimate the highest possible revenue for the producer, an optimization problem is employed to identify the optimal bid values. The predicted power values from the regression models are assumed to be the actual power values, allowing for the calculation of expected revenue based on these predictions and the prevailing market prices. Specifically, the expected revenue is determined through an optimization function that evaluates the predicted bids against the spot price and balancing prices. Subsequently, total revenue is calculated using the actual power production values alongside the established bid production values, with the actual revenue based on the real market conditions during operation.

The models are studied between January 2021 and January 2022. Table 7 presents the revenue calculated, with expected being lower than actual, indicating that the models are conservative. Notably, the Linear regression model with L2 regularization shows the highest expected and actual revenue.

Model	Expected Revenue	Actual Revenue
Linear L2	5,645,305,994.04	6,836,336,423.14
Non-linear	5,437,685,690.21	6,798,372,190.86
Non-linear L1	5,434,836,915.87	6,797,055,323.53
Linear L1	5,534,624,860.99	6,795,625,420.81
Linear	5,534,624,469.13	6,795,625,326.06
Non-linear L2	5,431,396,761.69	6,790,096,746.18

Table 7: Expected and actual revenues for models

Based on the results and previous evaluation metrics, it can be observed that a model that shows slightly higher error could demonstrate greater robustness and adaptability, making it better suited for real-world scenarios. While it may not perform as perfectly on the training data, it might handle new data more effectively, aligning better with actual conditions and uncertainties. It is essential to prioritize models that balance accuracy with the ability to generalize, especially when making critical decisions like determining optimal power bids.

8 Model 2

In this model, regression is leveraged to directly determine the most effective offering strategy. The model is developed by adjusting the training and testing datasets to reflect the new target variable to be the optimal offering strategy instead of wind power production. This bid aims to maximize revenue based on market conditions.

The selection of relevant features is achieved through systematic evaluation of several candidate variables, including *mean_wind_speed*, *SpotPriceDKK*, *BalancingPowerPriceUpDKK*, *BalancingPowerPriceDownDKK*, *mean_wind_power*, *max_wind_speed_3 sec*, and *max_wind_speed_10 min*. By examining all possible combinations of these features, the model identifies the most effective predictors for the optimal bidding strategy.

The model's performance is assessed using the same metrics. Among the various combinations tested, the features identified as yielding the best performance are *SpotPriceDKK*, *BalancingPowerPriceDownDKK*, and *mean_wind_power*. These selected features serve as the foundation for enhancing the model's complexity by incorporating non-linear relationships through weighted regression techniques. This approach allows for a more nuanced understanding of how different factors influence the optimal bidding strategy, ultimately leading to improved prediction accuracy and a more robust decision-making process in energy market trading.

In this context, the target variable—optimal bid—is continuous. The objective of the model is to predict a specific quantity rather than to categorize observations into discrete classes. Consequently, regression models provide superior insights into the relationships between the features and target variables, thereby enhancing the understanding of how various factors influence bidding strategies. Moreover, regression techniques enable the application of performance metrics that are well-suited for continuous outcomes, offering a clear evaluation of model accuracy and effectiveness in predicting continuous values. Therefore, regression is chosen as the technique to build Model 2.

8.1 Model Comparison

Regressions with regularization are carried out for Model 2. The summarized findings for the regression models are presented in Tables 8 and 9. An analysis of the results indicates that nonlinear regression for Model 1 outperforms the other models across the majority of performance metrics, indicating that the nonlinear approach may be more suitable for trading strategies. The weighted regression for Model 2 produces an R^2 value of 0.00, indicating a poor fit and will not be used for comparison. Overall, the linear models exhibit less variability in their performance, while the nonlinear models show greater potential, especially for Model 1.

	Linear		L1 Regularized		L2 Regularized	
Metric	Model 1	Model 2	Model 1	Model 2	Model 1	Model 2
MAE	924.990201	1943.820611	924.990261	1943.821174	951.117173	1947.395653
RMSE	1324.939629	2177.818720	1324.939564	2177.819050	1315.697545	2180.380307
R-squared	0.476335	0.373024	0.476335	0.373024	0.483615	0.371548

Table 8: Evaluation metrics linear models

	Nonlinear		L1 Regularized		L2 Regularized	
Metric	Model 1	Model 2	Model 1	Model 2	Model 1	Model 2
MAE	814.939662	2019.578242	817.799258	2233.792028	816.949700	2271.791746
RMSE	1301.251597	2775.633919	1299.001810	2958.363479	1298.631123	2999.826023
R-squared	0.494892	-0.018431	0.496637	-0.156939	0.496924	-0.189596

Table 9: Evaluation metrics of non-linear models

The strength of regularization is computed for the models, where the feature weights can be found in the appendix A.4. For the linear model, a λ value of 1 indicates a moderate level of regularization that effectively balances model complexity and prediction accuracy. In contrast, the high λ values of 33185 for L1 and 6 for L2 in the nonlinear model suggest an aggressive regularization approach, potentially leading to an overly constrained model. Such high values may prevent the predicted outcomes from falling within desirable ranges, particularly if the model is not designed to confine outputs to specific intervals, hindering the model's applicability.

8.2 Revenue Comparison

The revenue calculations derived from the various regression methods utilized in Model 2, as illustrated in Table 10, indicate a persistent under performance when compared to Model 1. This raises concerns about the effectiveness of the supervised learning techniques applied in Model 2, as they do not appear to yield optimized bidding strategies capable of significantly improving revenue.

This suggest that the current supervised learning framework lacks the flexibility and responsiveness necessary for effective decision-making in bid optimization. This limitation results in a failure to capture the intricate dynamics of the market, ultimately leading to suboptimal revenue outcomes.

Model	Actual Revenue (Model 1)	Actual Revenue (Model 2)
Linear L2	6,836,336,423.14	494,810.72
Non-linear	6,798,372,190.86	737,389.03
Non-linear L1	6,797,055,323.53	639,190.98
Linear L1	6,795,625,420.81	498,378.70
Linear	6,795,625,326.06	498,379.18
Non-linear L2	6,790,096,746.18	687,022.86

Table 10: Revenue comparison

9 Conclusion

In summary, the results from Model 1 indicate that the nonlinear regression model with L2 regularization yielded the best performance based on evaluation metrics, while the linear model with L2 regularization generated the highest revenue. In Model 2, the linear regression model with L2 regularization also demonstrated superior predictive capabilities, although the nonlinear model achieved the highest revenue generation.

These findings affirm that while the models can predict the power output of the wind farm with a degree of accuracy, they also highlight the limitations of using supervised learning for decision-making in this context. Specifically, the reliance on historical data and the inherent complexities of the market suggest that supervised learning may not adequately capture the nuances required for effective bidding strategies.

To enhance trading strategies, the following recommendations are proposed:

- Utilize unsupervised learning: Implement clustering techniques to identify patterns and group similar trading days or market conditions.
- Implement real-time data integration: Incorporation of real-time data feeds, such as live weather information, to enable dynamic adjustments in bidding strategies, facilitating quicker and more informed decision-making.
- Incorporate advanced machine learning techniques: Explore ensemble methods such as random forests, and deep learning models to capture complex relationships in the data, enhancing predictive accuracy and adaptability to evolving market conditions.

A more refined approach that leverages multiple methods could provide a more robust framework for optimizing bidding strategies and maximizing revenue in day-ahead and balancing markets.

A Appendix

A.1 Model 1 Figures

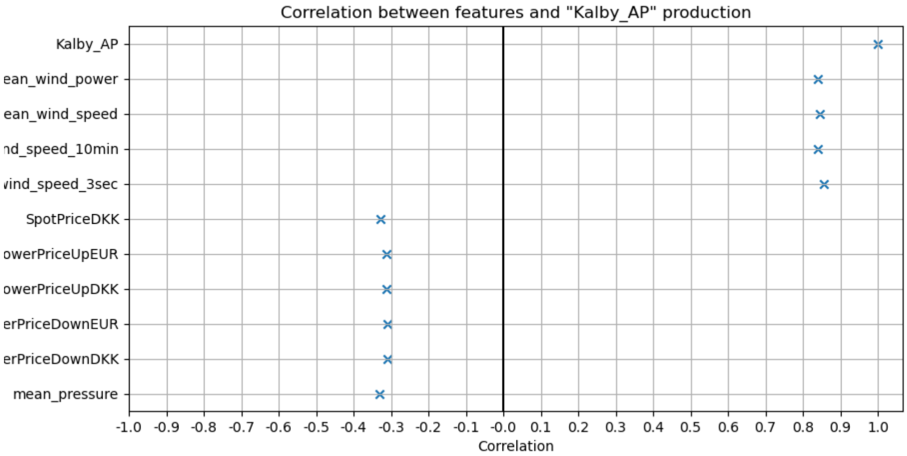

Figure 4: Correlation between features and target

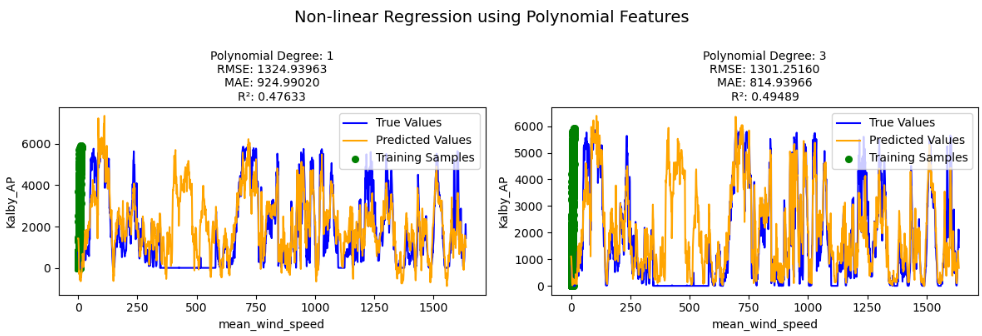

Figure 5: Nonlinear regression for different polynomial degrees

A.2 Regularization of non linear model (Model 1)

Feature	Weights	L1 Regularized	L2 Regularized
0	-652.506127	-928.979177	-624.891572
1	575.696937	828.655629	507.198942
2	-253.245596	-137.042100	-97.309615
3	-47.307781	-94.811435	-1.312067
4	153.358444	229.559313	88.744426
5	98.414295	150.533988	82.170816
6	36.006065	2.080707	49.883383
7	-287.819837	-330.934731	-256.030471
8	135.129412	122.559556	113.637739
9	-34.875266	-44.144060	-27.343045
10	230.740073	245.580578	201.651419
11	-103.789590	-94.950136	-98.834780
12	-249.857339	-261.529167	-217.314318
13	90.766683	84.152816	80.146683
14	22.549343	18.218867	24.679301
15	97.735117	103.799618	87.279276
16	-65.737362	-67.760902	-60.538019
17	25.062504	29.064117	22.333587
18	-11.958803	-11.575192	-11.100896

Table 11: Feature weights of non-linear models (polynomial degree 3)

A.3 Model 2 Figures

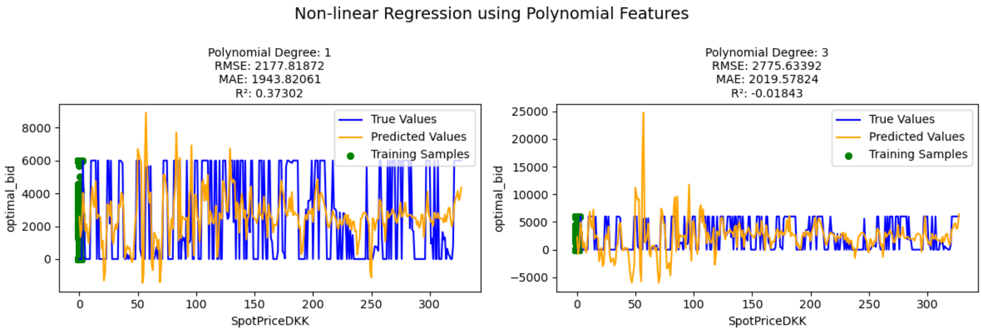

Figure 6: Nonlinear regression for different polynomial degrees for Model 2

A.4 Regularization (Model 2)

Feature	Weights	L1 Regularized	L2 Regularized
0	2566.388439	2566.388250	2565.148478
1	4254.496424	4254.488678	4202.742868
2	-1694.432922	-1694.430328	-1683.208214
3	-2837.626380	-2837.620969	-2795.850818
4	174.645160	174.645037	175.530577

Table 12: Feature Weights for linear model (Model 2)

Feature	Weights	L1 Regularized	L2 Regularized
0	8213.623048	6.734107e+03	6093.316080
1	-3118.938215	-9.583672e+02	-1031.920624
2	-5222.568156	-5.656869e+03	-5050.981889
3	885.458322	-1.204156e+02	-112.414800
4	963.217915	8.613865e+02	1610.029655
5	5798.383951	3.349276e-12	-88.610318
6	-9647.492680	-2.611800e+03	-2503.718888
7	-768.232648	3.107913e+02	612.673743
8	-2299.821126	6.402890e+02	1441.264235
9	-123.860003	-4.037656e-10	-1421.121208
10	-918.261221	1.813366e-12	-410.592031
11	5394.168599	3.566254e+03	3371.631548
12	1525.445879	2.724680e+02	530.615016
13	-189.383211	1.806630e+03	1856.599052
14	-2152.220406	-7.692954e+02	-1181.066999
15	3320.533457	1.134947e+03	2030.612551
16	4487.388460	1.331379e+03	1692.713031
17	21.776553	-3.832515e+02	-903.451732
18	-949.829587	-1.268741e+03	-1764.904952
19	-8479.069959	-2.448141e-11	-889.422884
20	273.645312	1.703918e-11	418.206893
21	622.439018	-1.475232e+03	-1728.598928
22	-118.261742	-9.804253e-13	313.696797
23	-14.760023	1.513020e-11	0.852210
24	310.947869	-1.374312e-10	-103.109791
25	672.551668	-8.696058e-12	16.395066
26	553.473829	3.632254e-08	63.723664
27	3944.958616	1.449799e-12	1078.494361
28	-282.283663	6.440902e-12	10.127106
29	-20.330342	-1.859156e+02	-78.396674
30	-1945.598136	1.804197e-12	-112.871132
31	-546.848891	-1.942286e+02	-492.379595
32	-22.438924	2.244148e-09	-182.119274
33	-31.501535	-5.038910e+02	-521.153326

Table 13: Feature weights for non-linear models (polynomial degree 3) (Model 2)