Technical University of Denmark (DTU)

MACHINE LEARNING FOR ENERGY SYSTEMS

Course 46755

Assignment 1:
Renewable energy trading in day-ahead and balancing markets

October 30, 2024

Group 7 - Students:

Salomé Aubri,	s233033
Leonie Meier,	s233049
Manisha Talukdar,	s232203
Victor Schmitt,	s233241
Alexandre del Barco,	s232518

Table 1: Contribution Table

	Salomé	**Leonie**	**Manisha**	**Victor**	**Alexandre**
Contribution to model development	20%	20%	20%	20%	20%
Contribution to programming	20%	20%	20%	20%	20%
Contribution to analysis of the numerical results	20%	20%	20%	20%	20%
Contribution to writing the report	20%	20%	20%	20%	20%
Overall	**20%**	**20%**	**20%**	**20%**	**20%**

Contents

List of Figures

Nomenclature

Sets

T Time

Parameters

λ_t^D Day-ahead price at hour t

$\lambda_t^{\downarrow}$ Down-regulation price at hour t

$\lambda_t^{\uparrow}$ Up-regulation price at hour t

$\overline{P}$ Wind farm capacity

p_t Prediction of wind power at time t

Decision Variables

$\boldsymbol{\delta}^+$ Positive part of the difference between bid and predicted wind production at time t

$\boldsymbol{\delta}^-$ Negative part of the difference between bid and predicted wind production at time t

$\hat{\boldsymbol{p}}_t$ Bid of wind power at time t on the day-ahead market

Introduction

The aim of this assignment is to apply linear and non-linear regression in two models for renewable energy trading. These models will focus on trading in day-ahead and balancing markets, with the goal of maximizing revenue.

The project focuses on the Kalby Wind Turbines, located on the island of Bornholm, where three Vestas V80-2.0MW turbines have been installed. Each turbine has a 2MW capacity and is mounted on a 60-meter tower. The main goal is to trade energy in electricity markets by analyzing historical data. Bornholm is part of the DK2 market area, and for this assignment, any instances of negative prices have been disregarded.

The objective is to develop two models for trading in day-ahead and balancing markets and to determine which performs better. The first approach uses regression to predict wind power, which is then applied to solve the decision-making problem for offering strategies. The second approach uses regression to directly find the most effective offering strategy. This method bypasses the need to predict wind power and instead focuses on finding the optimal strategy through regression analysis. The assignment examines how regression models and trading strategies interact, aiming to find the most effective ways to maximize revenue in the complex world of wind energy trading.

1 Model 1

1.1 Step 1: Trading optimization model

The goal of this step is to formulate an optimization model for wind power trading in day-ahead and balancing markets to maximize the wind farm revenues. This analysis is presented from the traders' perspective, where decisions are made on D-1, meaning the actual wind realization is still unknown. The day-ahead and balancing prices are assumed to be known, informed by historical data.

The decision variables (denoted in bold character) and the parameters are defined in the Nomenclature section. The linear optimization problem is defined as follows.

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

Subject to:

$$p_t - \hat{\boldsymbol{p}}_t = \boldsymbol{\delta}_t^+ - \boldsymbol{\delta}_t^- \qquad \forall t \in T \tag{2}$$

$$0 \leq \hat{\boldsymbol{p}}_t \leq \overline{P} \qquad \forall t \in T \tag{3}$$

$$\boldsymbol{\delta}_t^+, \boldsymbol{\delta}_t^- \geq 0 \qquad \forall t \in T \tag{4}$$

Equation 2 is used to linearize the model. The difference between the wind prediction and the bid is decomposed into a positive and negative part. Equation 4 ensures the positive and negative parts are positive. Equation 3 ensures the wind power bid is non-negative and lower than the wind farm capacity. Finally, Equation 1 is the objective function of the problem maximising the two different revenue streams of the farm: day-ahead and balancing market. In the balancing market, only the difference between the wind realization (since we are at D-1, the best realization is the power prediction) and the bid is considered. When there is excess production, the wind farm participates in the down-regulation market. On the contrary, a lack of wind production should be settled on the up-regulation market.

1.2 Step 2: Data Collection

In order to implement the optimization problem developed in Step 1.1, an accurate prediction of wind power for the next day is needed. Therefore, regression will be used as a method for making predictions. To achieve this, a robust dataset needs to be constructed using the provided actual data.

1.2.1 Step 2.1: Chosen features

The data are sourced from the EnergyDataDK platform, providing real-world data from Bornholm island. The data of this assignment is based on the Kalby wind park. The active power of the wind park (*i.e.*, the wind power production), taken from the **Bornholm Network Manager Historical Wind Data** dataset, which serves as target variable. Along with the DMI hourly weather observations data, retrieved from **Weather Observations DMI**, the Nordpool price information for electric power data from **Nordpool**, Energinet regulating and balance power from **Energinet Balancing Prices**, and the weather forecast data from **The Norwegian Meteorological institute**. In total, 19 features are selected initially, discarding all those features that which are not related to the site or wind power production. To predict wind production, the following features are incorporated:

- The actual wind power from the previous day, obtained by shifting the initial wind production dataset by one day
- The 0.05, 0.5, and 0.90 quantiles of wind production from the previous week, derived from a statistical analysis of the actual wind power
- The hourly 0.05, 0.5 and 0.90 quantiles of wind production within the specific hour from the previous week
- Wind conditions: both historical and forecasted wind speed, wind direction, mean air temperature, accumulated precipitation, and air humidity. The historical data (actual data with a one-hour shift, measured at Nexo weather station, the closest station to the wind park) are taken from the **DMI** dataset, while the forecasted data (weather forecasts for the current hour being considered) are sourced from the **The Norwegian Meteorological Institute**.

The study period spans from June 1^{st}, 2023 to October 1^{st}, 2023, during which there were no significant gaps in the data. Appendix A summarizes all the 19 features considered. Min-Max normalization was applied to rescale the dataset, restricting feature values within the 0 to 1 range. This procedure, which involved identifying feature-specific minimum and maximum values, aimed to eliminate scale-related biases, ensuring equitable contribution of data points to subsequent analyses. With those 19 features, a correlation matrix is created being Active Power of Kalby wind park the target variable, Appendix B. Due to the very low correlation values, 5 features related to temperature and precipitation are removed, thus 14 features in total are used for the regression model, they used features are shown in Table 2 and their correlation is shown in Figure 1.

1.2.2 Step 2.2: Predictor's Performance Evaluation

Evaluating the performance of the predictor is fundamental, in order to asses the reliability of the wind power prediction model. The measures Mean Absolute Error (MAE) and Root Mean Square Error (RMSE), were employed to quantify the accuracy of the predictions.

Table 2: Used Features Summary

Previous Day Wind Production	Forecasted Accumulated Precipitation
5th Quantile Production Previous Week	5th Quantile Production Specific Hour Previous Week
50th Quantile Production Previous Week	50th Quantile Production Specific Hour Previous Week
90th Quantile Production Previous Week	90th Quantile Production Specific Hour Previous Week
Mean Wind Speed for the Previous Hour	Forecasted Mean Wind Speed
Mean Wind Direction for the Previous Hour	Forecasted Mean Wind Direction
Mean Air Humidity for the Previous Hour	Forecasted Mean Air Humidity

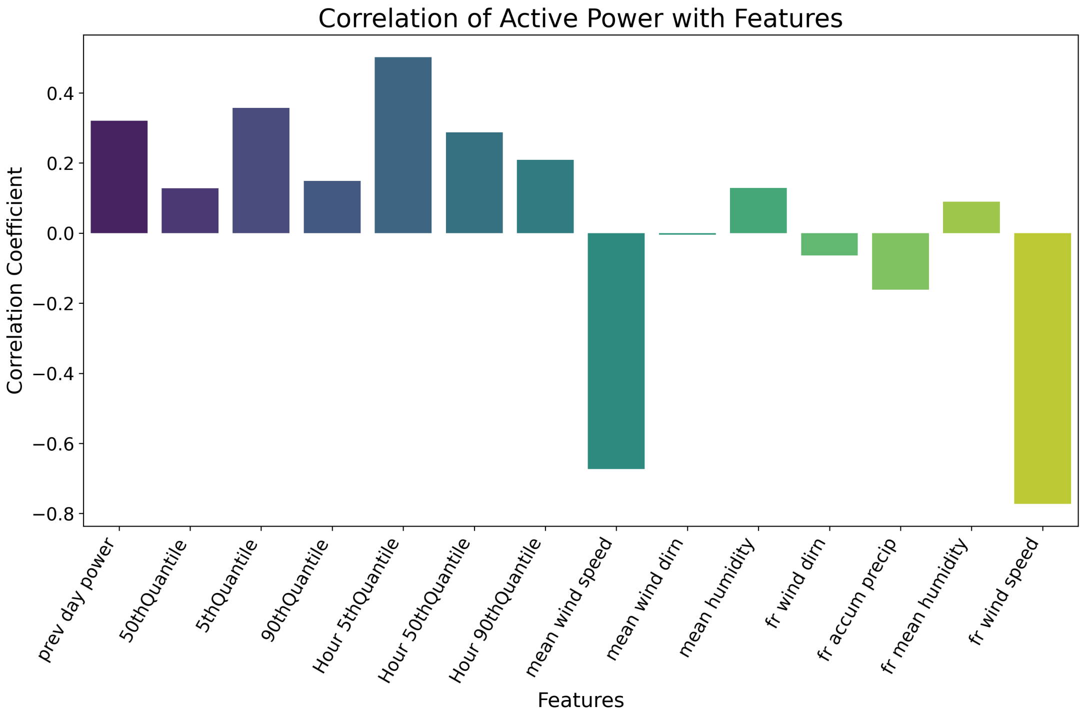

Figure 1: Correlation Matrix for final 14 features

The Mean Absolute Error (MAE) is a metric that represents the average of the absolute differences between predicted and actual values. The MAE is defined as follows, where y_i represents the actual values, $\hat{y}_i$ denotes the predicted values, and n is the total number of observations:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

The Root Mean Square Error (RMSE) is a metric that measures the square root of the average squared differences between predicted and actual values, taking into account both the magnitude and variance of prediction errors. It offers insights into the overall goodness-of-fit of the model and is defined as follows:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Overall, a lower value signifies a better predictive accuracy. To robustly assess the model's performance, **k-fold cross-validation** was employed. This technique partitions the dataset into 'k' subsets or folds, ensuring that each fold serves as both a training and testing set. Specifically, 10-fold was used to cross-validate. The dataset is divided into 10 subsets, and through each fold, training the model on 9 of the folds and evaluating its performance on the remaining fold. MAE and RMSE are then calculated for each training fold, providing insights into performance variations.

Figure 2 shows the different RMSE and MAE scores for each fold.

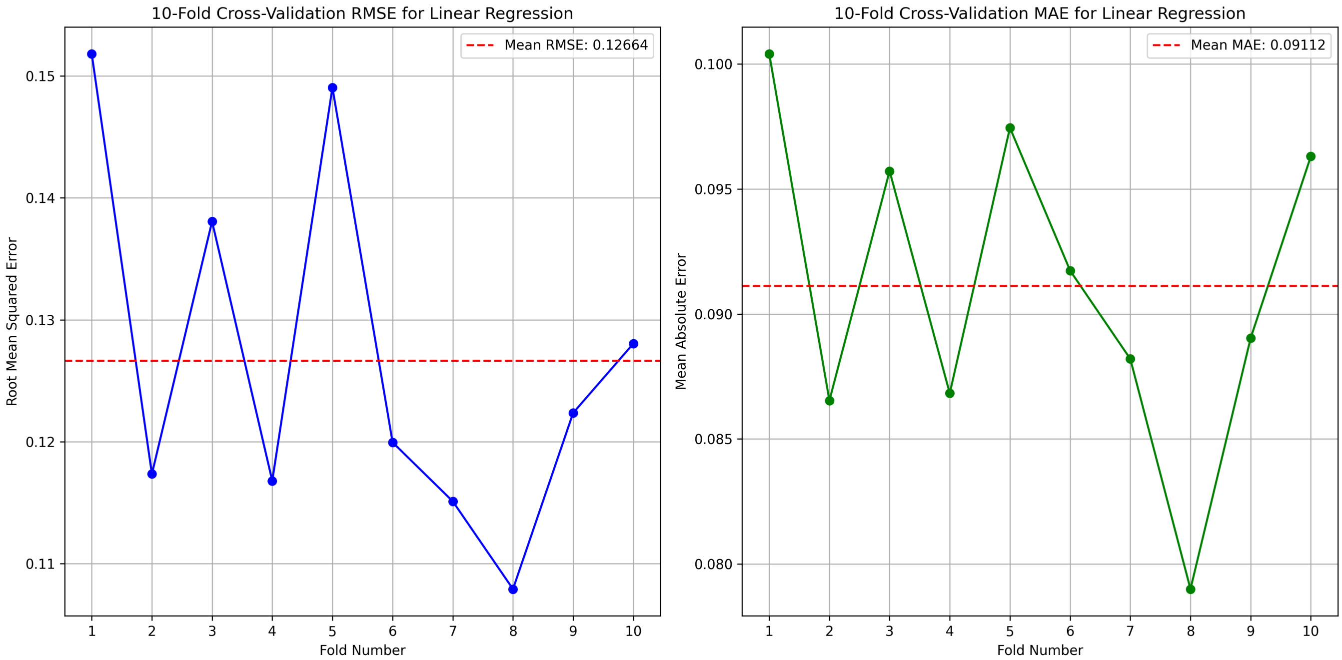

Figure 2: RMSE and MAE scores for each fold

The results obtained in the cross-validation gives a priori good results and confidence in the predictor, as it manages to obtain a low error score for each fold, resulting then in its ability to generalize across various data. This will be further assessed when comparing with the developed models.

1.3 Step 3: Linear regression

Regression models can help in forecasting wind energy based on historical data, leading to improved strategies for bidding in the electricity market. This section focuses on the evaluation of the linear regression method. Two algorithms have been implemented and tested, the gradient descent algorithm and the closed form solution.

1.3.1 The Gradient Descent Algorithm

In the gradient descent algorithm, parameters are adjusted iteratively to minimize the error. The goal is to minimize the cost function, which quantifies the error between the prediction ($\boldsymbol{\theta}^\top \boldsymbol{x}^i$) and the target ($y^i$) for each sample i:

$$J(\boldsymbol{\theta}) = \frac{1}{2}\sum_{i=1}^{n}(\boldsymbol{\theta}^\top \boldsymbol{x}^{(i)} - y^{(i)})^2, \tag{5}$$

To achieve this, we compute the partial derivatives of the cost function with respect to the parameters and find the parameter combinations for which these derivatives equal zero. Given multiple training samples, the update rule for the parameters is given by:

$$\theta_{t+1} = \theta_t - \alpha \frac{1}{n} \sum_{i=1}^{n} (\boldsymbol{\theta}^\top \boldsymbol{x}^{(i)} - y^{(i)}) \boldsymbol{x}^{(i)}.$$

The Gradient descent algorithm is therefore written as follows :

Algorithm 1 Gradient Descent Algorithm

1: Initialize θ_0, and set $t = 1$
2: **while** $|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t| \geq \varepsilon$ & $t \leq 1000$ **do**
3: $\quad \boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \frac{\alpha}{n} \left(\sum_{i=1}^{n} (\boldsymbol{\theta}_t^\top \boldsymbol{x}^{(i)} - y^{(i)} \right) \boldsymbol{x}^{(i)}$
4: $\quad$ t = t + 1
5: **end while**
6: **return** $\boldsymbol{\theta}_{t+1}$

It updates the parameters in the direction opposite to the gradient to find the local minima. The update rule involves α the learning rate, and n the number of features, as in this case a multi-features regression is applied. The learning rate controls the step size of each iteration. Here a value of $\alpha = 0.1$ has been chosen, whereas 1000 iterations at maximum were done. This combination of parameters was offering a particularly good trade-off between computation speed and accuracy, according to the closed-form solution.

After being trained and the optimal $\boldsymbol{\theta}^*$ being obtained, the wind production can be predicted according to:

$$\hat{y}_{t+k|t} = {\boldsymbol{\theta}^*}^\top \boldsymbol{x}_{t+k} \tag{6}$$

1.3.2 The Closed-form solution

The Closed-form solution directly computes the optimal model parameters through matrix operations, minimizing the cost function.

$$\boldsymbol{\theta} = (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top y \tag{7}$$

and the prediction can be obtained through :

$$\hat{y}_{t+k|t} = \boldsymbol{\theta}^\top \boldsymbol{x}_{t+k} \tag{8}$$

1.3.3 Prediction results

For both methods, the dataset has been divided in two : 80% of the samples were used for training, whereas the 20% left were used for testing purposes.

In Table 3, we compile the results for the RMSE and MAE metrics, for both 500 samples and all the samples (2791). Both results show a close similarity in performance. As expected the model with just 500 samples performs a bit better than the one with all the samples, as with fewer samples, the model may be more likely to overfit the data, and the data can be simpler, allowing the model to perform better.

Figure 3 shows the different results got from the two methods on the training set, and the comparison with the actual data, and how the prediction follows the overall trend quite closely. This suggests that some of our features likely have a strong correlation with the actual wind power, which may contribute to the success of the linear regression model.

Table 3: Wind Production Prediction Results

	RMSE	MAE
Gradient Descent (500 samples)	0.0831	0.0698
Closed Form solution (500 samples)	0.0783	0.0666
Gradient Descent (all samples)	0.1067	0.0795
Closed Form solution (all samples)	0.1057	0.0786

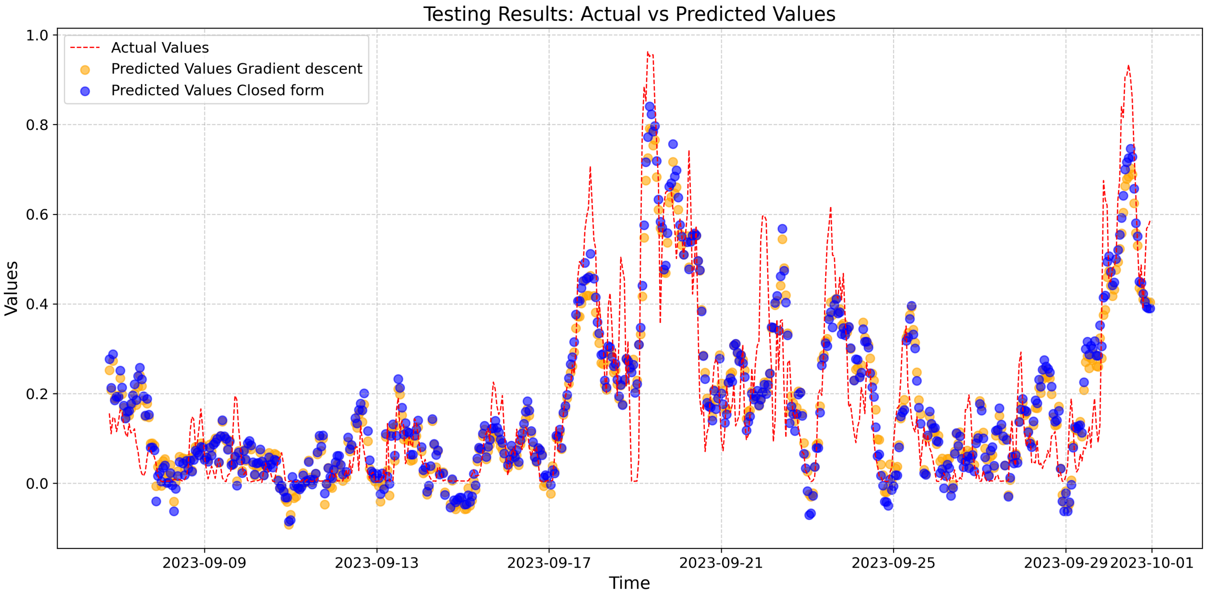

Figure 3: Comparison between predicted and actual values

1.4 Step 4: Nonlinear regression

This section extends the linear regression model from subsection 1.3 to a non-linear model by applying non-linear transformations to selected features.

1.4.1 Linear Regression with Non-Linear Features

Of the 14 features used for linear regression, we derive 10 additional non linear features for root-squared, squared and cubic forecasted and actual wind speeds, and exponentials of hourly and weekly 5^{th} and 90^{th} quantiles, with some scaling factor allowing better results. These non-linear features were selected iteratively, in order to decrease the error.
Here, the model remains itself linear although some features are now non-linear. In that case, Equation 7 and Equation 8 can still be used to predict the wind production. The performance for this model is shown and compared in Section 1.4.3. Figure 4 shows the results of the prediction with this model.

On this Figure, one can already see that some peaks are reached more accurately than previously, and the amount of predicted negative values also decreased, showing overall a better accuracy.

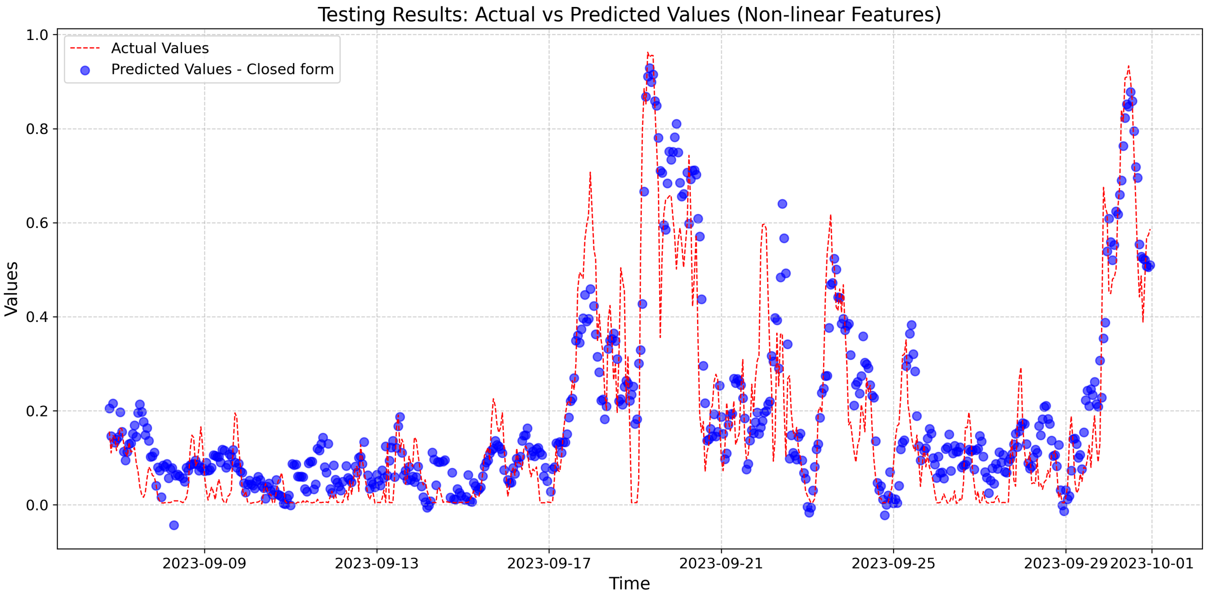

Figure 4: Comparison between predicted and actual values

1.4.2 Locally weighted regression

The locally weighted regression method is applied to the initial model without altering the features used in Step 3. This method assigns weights to data points based on their distance from the query point (x_u), using a Gaussian Kernel function, as shown in Equation 9. In this context, the query points are the data points in the test dataset, while x_t represents a point from the training dataset. As a result, each point in the test dataset is associated with its own weight ($\boldsymbol{W}$), a diagonal matrix that contains the weights for each point. The w_t function assigns higher importance to the training data points based on the proximity to the test datapoints.

$$\mathbf{W} = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{\|\mathbf{X}_t - \mathbf{X}_u\|}{\kappa}\right)^2\right) \tag{9}$$

The parameter κ is called the radius in this equation and characterizes how wide the range will be around a query point where the method assigns non-zero weights to the surrounding data points. It has been selected using a sensitivity analysis where 10 values have been compared according to its corresponding RMSE error value. The optimal value obtained for $\kappa = 0.26667$ according to Appendix C.

The function `weighted_least_squares` calculates the weights for each point in the test data set and calculates the predicted y values based on these equations:

$$\theta = (\mathbf{X}^\top \mathbf{W}\,\mathbf{X})^{-1}\mathbf{X}^\top \mathbf{W}\,\mathbf{y}$$

$$\mathbf{y}_{\text{pred}} = \mathbf{X}_{\text{u}} \cdot \theta$$

1.4.3 Comparison of Non-linear and Locally Weighted regression with Linear regression

The RMSE and MAE of both non-linear models are in Table 4 compared with models from Step 3.

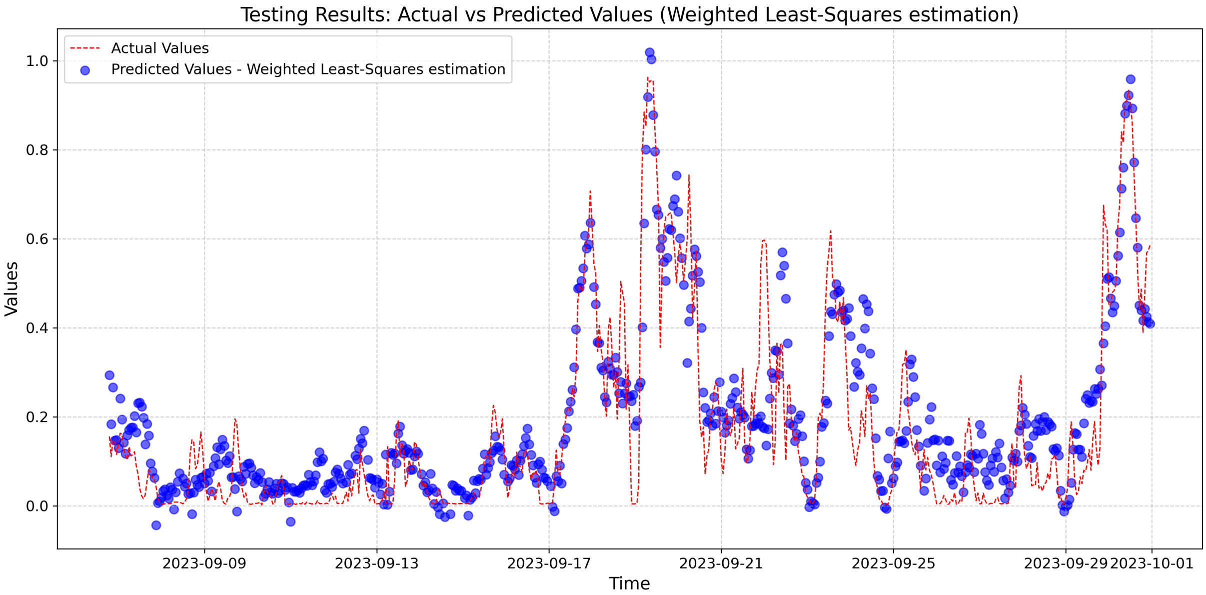

Figure 5: Comparison between predicted and actual values

Table 4: Prediction Error of Linear and Non-Linear Regression models

	RMSE	MAE
Linear regression model (Step 3)	0.1057	0.0786
Non Linear regression model (Step 4.1)	0.1033	0.0757
Locally Weighted regression model (Step 4.2)	0.1024	0.0746

Comparing errors to the Linear model, both the RMSE and MAE error obtain lower values, especially, the Locally Weighted regression model, providing the best prediction. For example, in the most extreme values, this method presents a higher accuracy. Even though the good performance this model presents, it still has some limitations, as the model sometimes predicts negative wind power production, which can not be physically possible under typical operating conditions. While some wind turbines do consume small amounts of power to autonomously manage essential functions during low-wind periods, this consumption would not result in the negative production values the model sometimes predicts. Overall, it can still be seen that adding non-linearity, improves the accuracy of the prediction by capturing higher complexities of the dataset.

1.5 Step 5: Regularization

In this section, a regularization term is added to the Linear and Non-Linear models to improve the estimation of the output.

1.5.1 Lasso (L1) Regularization

In the Lasso Regularization, the incentive term is the sum of the absolute value of the weights. The equation for Linear regression with Lasso regularization is given as:

$$\hat{\beta} = \arg\min_{\beta} \sum_i \epsilon_i^2 = \arg\min_{\beta} \sum_i (y_i - \beta^T x_i)^2 + \lambda \sum_j |\beta_j| \tag{10}$$

The regularization parameter (λ), also called hyperparameter, determines the strength of the incentive term added to the model. This parameter reduces the overfitting of the model, which means it will reduce the variance of the estimated regression parameters, but also add bias to the estimate. A large value of λ will result in less overfitting but higher bias.
The code defines a function to create and train Lasso regression models with different values of λ, calculating the Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) for each model through an iterative process. The Lasso regression model from the Sci-kit Learn package was used. The regularization parameter and number of iterations are inputs to the function. When using this method, a validation set is used which is 20% of the training set. The new training set is now 80% of the original one. The validation set is important as it allows for a decision to be made on the best parameter to choose. Then, the actual error of the model needs to be evaluated on a set that has not been used before and this justifies the need of a test set.

1.5.2 Ridge (L2) Regularization

In the Ridge regularization, the incentive term added to the model is the sum of the square of the weights. The equation for Linear regression with Ridge regularization is given as:

$$\hat{\beta} = \arg\min_{\beta} \sum_i \epsilon_i^2 = \arg\min_{\beta} \sum_i (y_i - \beta^T x_i)^2 + \lambda \sum_j ||\beta_j||^2 \tag{11}$$

The Ridge regression model from the Sci-kit Learn package was used. Once again, an iteration process and a validation set are used to find the best regularization parameter.

1.5.3 Performance evaluation of L1 & L2 Regularized model

Table 11 and Table 12 displays the Mean Absolute Error and Root Square Absolute Error for each λ in L1 and L2 regularization, for linear and non-linear models.

The reason Ridge regression shows similar error levels across different λ values is that it does not entirely remove any features from the model. In contrast, Lasso regression sets some coefficients to zero, therefore excluding certain features. When λ is high, indicating strong regularization, Lasso is more aggressive in feature selection, which can lead to increased error as important information may be disregarded.

As shown in Table 12, error values for the Lasso regularization change slightly across different λ values, with a general trend of increasing error as λ rises. However, the corresponding error in the Non Linear model is lower for the same values of λ. Similar behaviour can be seen for Ridge regularization with lower variation of RMSE with changing λ due to L2 regularization's tendency to retain all features.

Overall, the results indicate that Ridge (L2) regularization for non-linear regression is the preferred approach, yielding consistently lower error metrics (in bold in Table 12) for a value of $\lambda = 0.001$. When evaluating the model on the test set, RMSE and MAE values are computed and equal to **0.105173** and **0.078864** respectively. They represent the expected error of the model when assessed on a "real life" case meaning on a set that was not used to train or make a decision.

1.6 Step 6: Revenue calculation for evaluation

At this step, the performance of the predicted power production from the regression models is evaluated. Predicted and realized revenues are calculated based on Equation 1.1, using historical data for day-ahead and balancing market prices. The optimal bid is characterized based on the relationships between the prices for

$\lambda_t^{\downarrow}$, $\lambda_t^{\uparrow}$, and λ_t^{D}, as here, the prices at hour t are assumed as known information.
Full Capacity Bid: When $\lambda_t^{\downarrow} < \lambda_t^{\mathrm{D}}$, it is advantageous to bid at full capacity. This is because the price for selling energy in the day-ahead market exceeds the price for providing downward regulation.
Zero Bid: Conversely, when $\lambda_t^{\uparrow} > \lambda_t^{\mathrm{D}}$, the model will choose to bid zero. In this case, the cost of upward regulation is higher than the market price. Therefore, the model decides not to bid in the day-ahead market to avoid needing upward regulation.
Bidding the Prediction: In the case where $\lambda_t^{\uparrow} > \lambda_t^{\mathrm{D}} > \lambda_t^{\downarrow}$, the optimal bid is according to the prediction, as the day-ahead price still offers a better return than the downward regulation price.

First, the validation set is used to evaluate the regression methods. The day-ahead bid is determined with the optimization method from Section 1.1, using the predicted values from each model as p_t^{real}. The optimal bid is then applied to calculate the actual revenue based on the true values of p_t^{real}. Comparing these two revenue calculations allows for an assessment of the prediction model's accuracy. The results are shown in Table 5.

Table 5: Comparison between forecasted and real revenues

Model	Forecasted Revenue	Real Revenue	Deviation (%)
Linear Regression (Closed form)	27450.65	22540.89	+21.78
Linear Regression (Gradient Descent)	27022.41	22536.98	+19.90
Closed form with non-linear features	30259.52	22537.95	+34.26
Non-linear Regression (locally weighted)	30471.35	22559.78	+35.07
Linear Regression L1	26753.67	22541.33	+18.69
Linear Regression L2	29837.31	22602.36	+32.01
Non-linear Regression L1	32462.86	22514.65	+44.19
Non-linear Regression L2	32871.20	22536.74	+45.86

All regression models tend to predict higher revenues compared to actual revenue. This could be caused due to certain hours, in which the prediction is high but unrealistic, leading to high revenues. These outlier predictions significantly exceed historical values, resulting in bids that inflate revenue forecasts artificially. Since the actual revenue is consistently lower, the model with the lowest predicted revenue offers the best accuracy. In this case, the linear regression models outperform the non-linear ones, and it can be noticed that less accurate models leads to better revenue predictions.

The Linear model with L1 regularization has shown the best performance. With all previously defined data sets utilized, a new test set has been established for the period from November 1st to November 8th, 2023. The deviation between actual and forecasted revenues on this test set is **11.36%**, indicating a significantly improved prediction accuracy. This improvement may be attributed to the shorter duration of the test period.

2 Model 2

In contrast to the first model, in the second approach, we utilize regression to directly determine the most effective offering strategy, *i.e.* the optimal bid the wind farm should place on the day-ahead market.

2.1 Step 1:

In order to adjust our training and testing dataset in terms of features and target, it is important to note that the label in these datasets is not wind power production; instead, it represents now the optimal offering strategy.

To obtain this, one run the optimization problem from step 1 for each datapoint of the actual historical wind production of the wind farm. It gives than the optimal bid for each data point in case of a 100% certainty scenario. Figure 6 shows the obtained optimal bids.

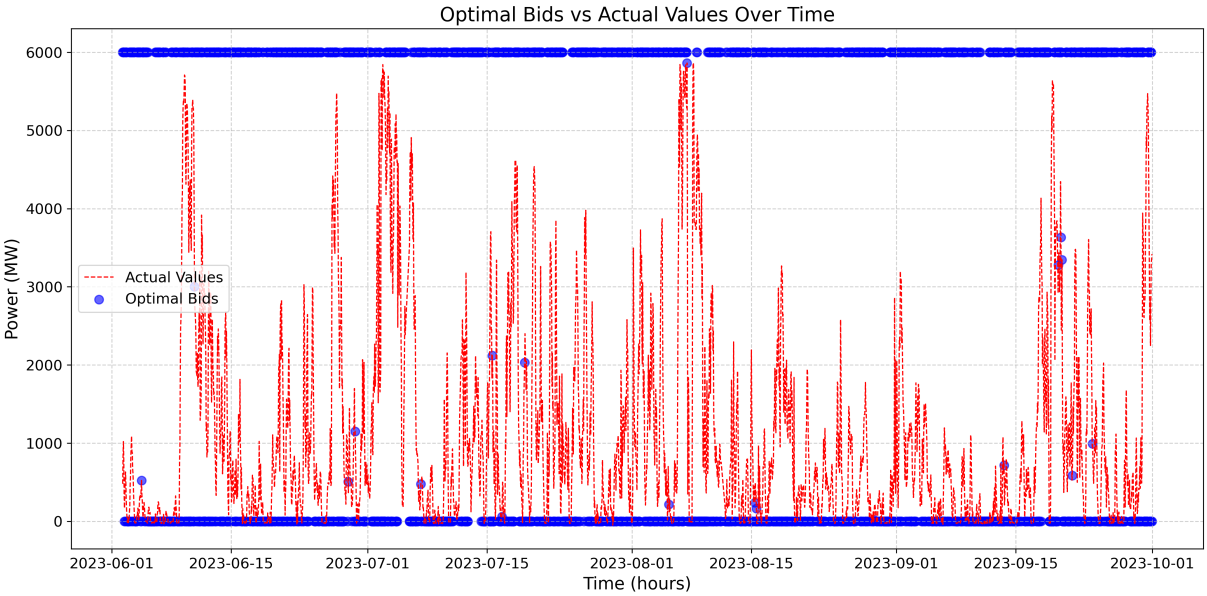

Figure 6: Optimal bidding strategy

Once the values for the historic optimal bids were obtained, the prices of the Day-Ahead and Up and Down Balancing markets were added to the dataset. These values were normalized to keep them within the 0 to 1 range.

Finally, the actual wind power was dropped from the features dataset, as this variable is now unknown.

Consequently, the final fitting dataset contained a total of 15 distinct features : 13 from the first model without the actual wind power, and 3 prices features (Day-ahead power prices; Up-Regulation power prices and Down-regulation power prices).

2.2 Step 2:

Using the new dataset obtained from the previous step, various models can now be tested to predict the bids with greater accuracy. Three different regression models were evaluated, and their accuracies are compiled in Table 6.

Table 6: Prediction Error for different Regression models

	RMSE	MAE
Linear regression model	0.4710	0.4227
Non Linear regression model	0.4469	0.3903
Locally Weighted regression model	0.4507	0.3917

The predicted bids are illustrated in Figure 7.

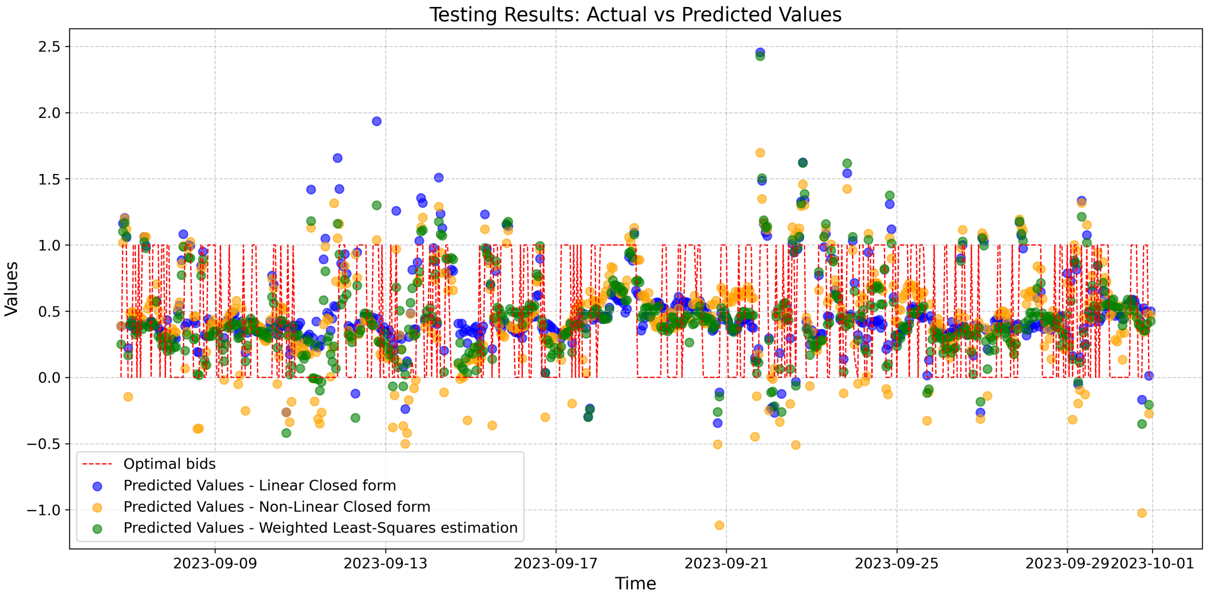

Figure 7: Comparison between predicted and actual values for 3 different models

As can be seen, the error values have increased significantly in comparison to the previous model. The most accurate technique here is the linear closed-form solution using non-linear features. These inaccurate predictions are caused by the fact that the optimal bid obtained through optimization represents an almost binary output (in a single price scheme, it is either more profitable to bid your entire capacity on the Day-Ahead Market or nothing, as revenues can still be gained from the Balancing market). Therefore, the optimal bid is almost a step function, and linear regression techniques struggle to fit a continuous function to a step function, indicating that these methods are not ideal for predicting the optimal bid.
To achieve better accuracy, classification could be a solution, as its goal is to predict discrete class labels.
A K-Nearest Neighbour (KNN) classifier was implemented as a classification method. Given that the optimal bids primarily consist of all-or-nothing values, the few remaining values were rounded to enhance suitability for classification. To determine the best value for k (the number of neighbours), a range of k values was tested using a validation set. This approach allowed us to evaluate the performance of the classifier across different k values and identify the one that yielded the lowest RMSE score. The optimal k value, denoted as $k_{\text{best}} = 10$, was then stored for use in predicting values on the actual test dataset. Figure 8 illustrates the classification results for the optimal k.

Despite the application of this technique, the results indicated poor accuracy (see Table 7).

Table 7: Prediction Error for the KNN classifier

	RMSE	MAE
10-Nearest Neighbours classifier	0.6518	0.4268

2.3 Step 3:

In the final evaluation phase, the classification model (10-Nearest Neighbours classifier) was selected as the best-performing method for Model 2. Performance was then evaluated by calculating revenue, as defined by Equation 1.1, but applied as a direct equation rather than a maximization function. This revenue was computed using both the predicted values and the actual values from the test set.

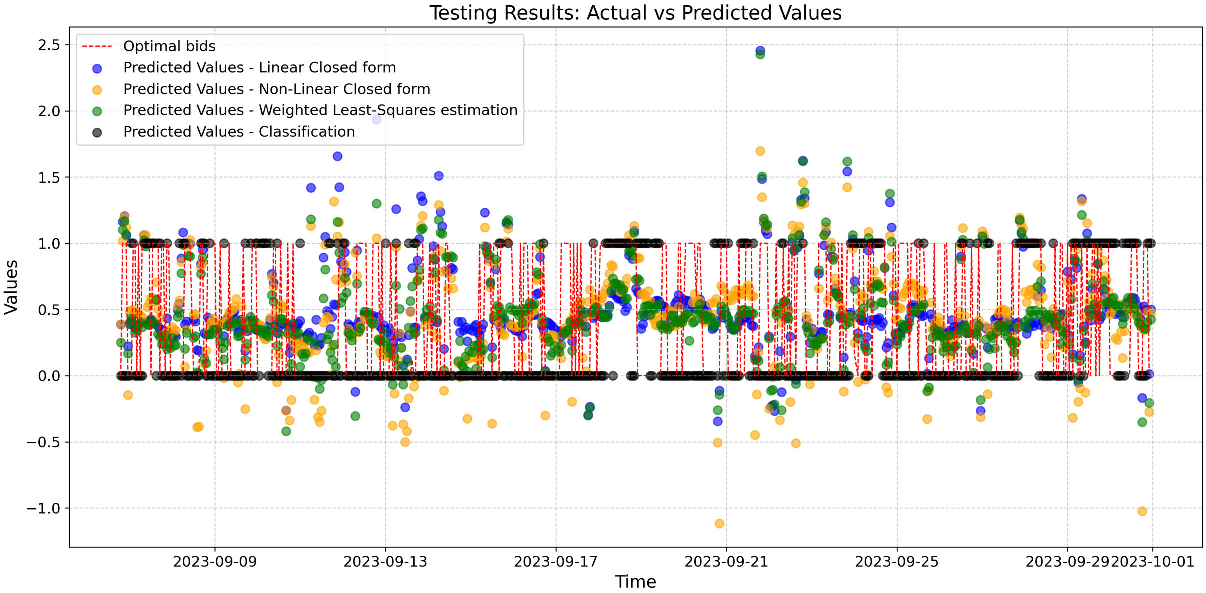

Figure 8: Comparison between predicted and actual values for 3 different models and a 6-Nearest Neighbours classifier

Table 8: Comparison between forecasted and real revenues - Model 2

Model	Forecasted Revenue	Real Revenue	Deviation (%)
KNN classifier	9395.86	10346.07	-9.18

The results, summarized in Table 8 for Model 2, are contrasted with the findings in Table 5, which reflect the performance of Models 1. One of the main insights that can be stand out is that the previous model tends to present a positive deviation from the real values, whereas Model 2 the deviation is negative. Moreover, Model 2 shows the lower absolute deviation among all with **9.18 %**. Overall, Model 1 gives, with all techniques implemented, substantially higher revenues. At the end, it is a trade-off. Some companies might prefer to forecast the most accurately their revenues, even though they are lower. Some others would prefer higher revenues, and therefore less predictability.

2.4 Step 4:

To enhance classifier performance, only price-based features (specifically Day-Ahead and Up-and Down-Balancing prices) were considered. This adjustment allowed the classification process to resemble a decision tree that follows the current dynamics of bidding within a single price scheme, as implemented in Denmark since November 2021. Consequently, after focusing solely on price-based features, Figure 9 was produced, utilizing a 6-Nearest Neighbours classifier.
Figure 9 shows a much more accurate prediction of the optimal bidding strategy, which is however unfeasible in real-life, as prices are supposedly not known in advance. Table 9 shows also the improvement of accuracy with those selected features.

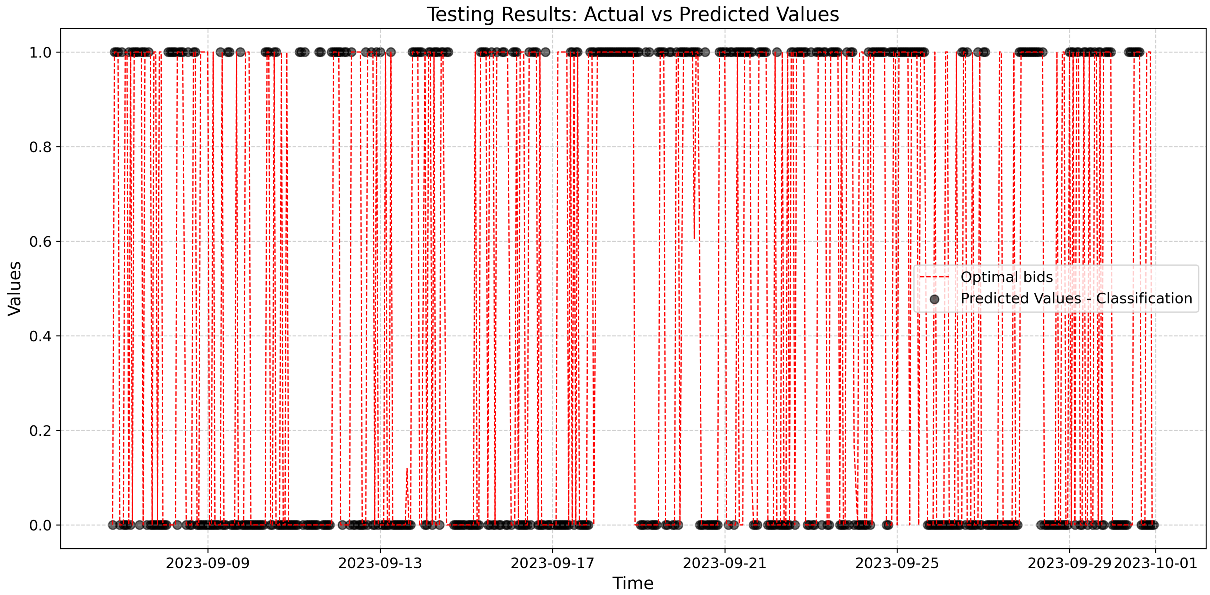

Figure 9: Comparison between predicted and actual values for a 6-Nearest Neighbours classifier

Table 9: Prediction Error for the KNN classifier

	RMSE	MAE
6-Nearest Neighbours classifier	0.5359	0.2890

It has been chosen to keep the one-price scheme in this assignment as it is the current rule in Denmark. This assumption leads to incoherent results as the balancing prices are assumed to be true predictions. The first improvement of the model would be to forecast these prices. However, under a one-price scheme, forecasting the balancing prices implies forecasting the imbalance of the system at each hour. This task is not trivial at all.

Another key consideration is the "all-or-nothing" bidding approach, where strategic bidding is actually restricted in real energy systems under one-price scheme. The main reason for that is to support system stability : what would happen if all renewable plants would bid strategically on the same hour and decide not to produce? More expensive power plants would have to be activated, leading to a lower social welfare and endangering the grid stability. In such cases, participants cannot bid adaptively or respond to short-term changes. To address this limitation, leveraging the Intraday Market becomes crucial, as it allows stochastic (variable-output) generators to adjust their bids closer to real-time, thereby improving their position within the market and enhancing overall adaptability. The current model, however, does not offer an alternative to the "all-or-nothing" strategy, indicating that future models might benefit from incorporating mechanisms to enhance bid flexibility, especially in Intraday trading contexts.

3 Appendix

A Initial features summary - Model 1 : Step 2

Table 10: Initial features summary

Previous Day Wind Production	
5th Quantile production previous week	5th Quantile production specific hour previous week
50th Quantile production previous week	50th Quantile production specific hour previous week
90th Quantile production previous week	90th Quantile production specific hour previous week
Mean Wind speed for the previous hour	Forecasted Mean Wind speed
Mean Wind direction for the previous hour	Forecasted Mean Wind direction
Mean Air Temperature for the previous hour	Forecasted Mean Air Temperature
Mean Air Humidity for the previous hour	Forecasted Mean Air Humidity
Max Air Temperature for the previous hour	Min Air Temperature for the previous hour
Accumulated precipitation in the previous hour	Forecasted Accumulated precipitation

B Correlation Matrix of initial features - Model 1 : Step 2

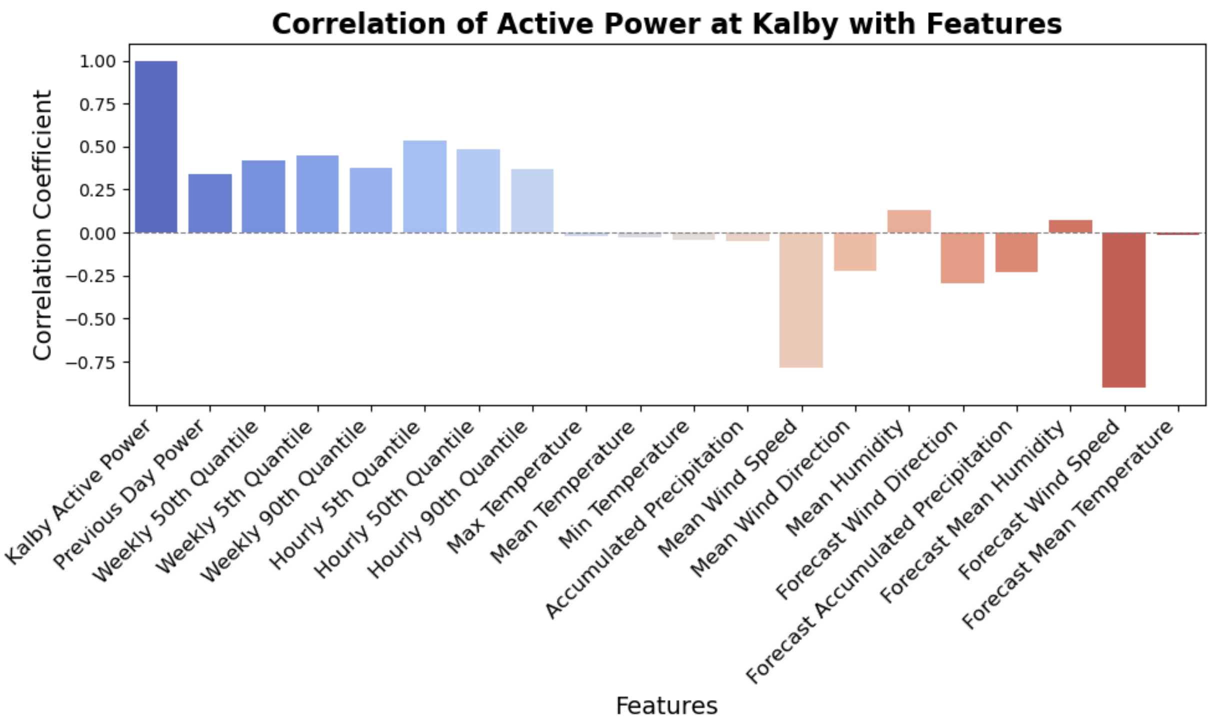

Figure 10: Correlation Matrix initial 19 features

C Optimal parameter κ - Model 1 : Step 4

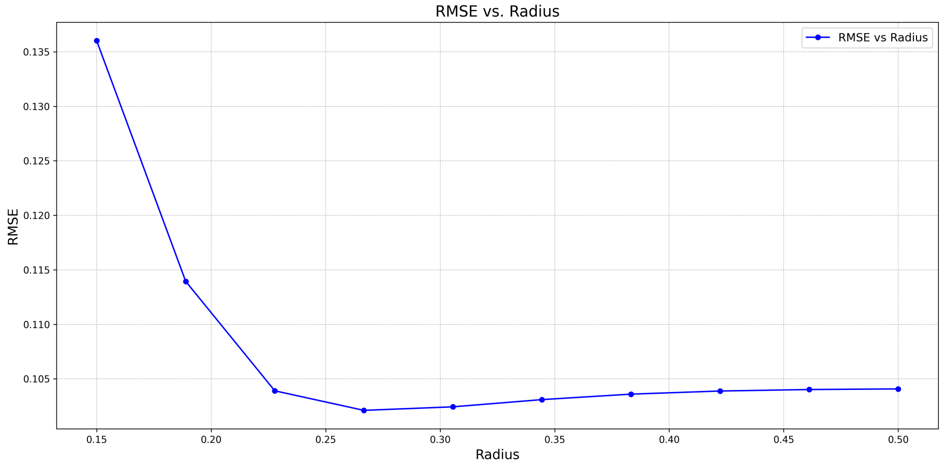

Figure 11: RMSE against radius κ

D Results - Model 1 : Step 5

Table 11: Results for the linear and non-linear models under Lasso regularization

λ	Linear		Non-linear	
	RMSE	MAE	RMSE	MAE
1	0.15985	0.14235	0.1598	0.1423
0.1	0.15985	0.14235	0.1598	0.1423
0.05	0.15985	0.14235	0.1598	0.1423
0.01	0.09212	0.07126	0.0921	0.0712
0.005	0.08764	0.06720	0.0876	0.0672
0.001	0.08675	0.06858	0.0867	0.0685
0.0005	0.08898	0.07060	0.0917	0.0735
0.0001	0.09207	0.07315	0.0853	0.0668
0.00001	0.09288	0.07378	0.0833	0.0661

Table 12: Results for the linear and non-linear models under Ridge regularization

λ	Linear		Non-linear	
	RMSE	MAE	RMSE	MAE
1	0.091981	0.07304	0.08966	0.07050
0.1	0.092868	0.07377	0.08706	0.06826
0.05	0.092921	0.07381	0.08637	0.06804
0.01	0.092963	0.07384	0.08358	0.06626
0.005	0.092968	0.07385	0.08285	0.06560
0.001	0.092973	0.07385	**0.08237**	0.06526
0.0005	0.092973	0.07385	0.08238	0.06531

Table 12: Results for the linear and non-linear models under Ridge regularization

λ	Linear		Non-linear	
	RMSE	MAE	RMSE	MAE
0.0001	0.092974	0.07385	0.08241	0.06538
0.00001	0.092974	0.07385	0.08243	0.06540

E Model 1 : Step 7 : Suggesting for improvement

One potential way to improve the accuracy of the prediction is to use different regression models for different groups of data points. Therefore, in this step, the data will be clustered in groups by using the k-means algorithm from the Sklearn library. After fitting the model with the training dataset, the K-means method will be able to cluster the test dataset following the patterns discovered in the training phase. Figure 12 shows the resulting clustering for the testing dataset. Here, $K = 5$, as this amount of clusters provided the best prediction results.

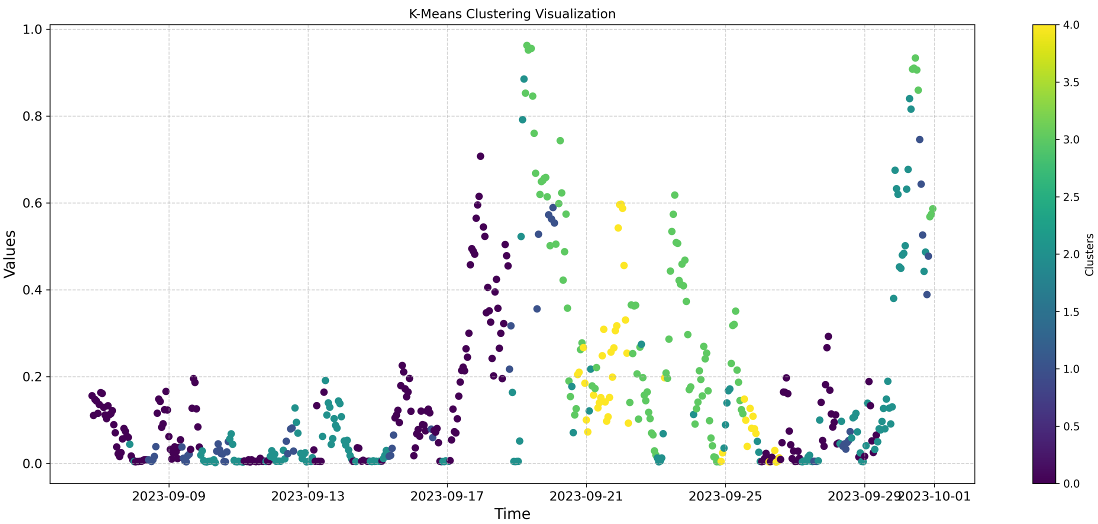

Figure 12: Clustering using K-Means with K=5

Once the dataset is clustered, we combine two different regression techniques: Linear Regression and Weighted Least Squares. To find the best combination of these models, we train and test each possible combination of models applied to the different clusters. There are 5 clusters, so there are $2^K = 32$ combinations of models to evaluate. The training and testing datasets remain the same, but the choice of regression model applied depends on which cluster a data point belongs to.

For each combination, predictions are made based on the models assigned to each cluster. These predictions are then combined across all clusters and evaluated over the entire test period to calculate the overall error. Figure 13 shows the best prediction after clustering. The optimal combination involved using Linear Regression for clusters 3 and 4, and Weighted Least Squares for clusters 0, 1, and 2.

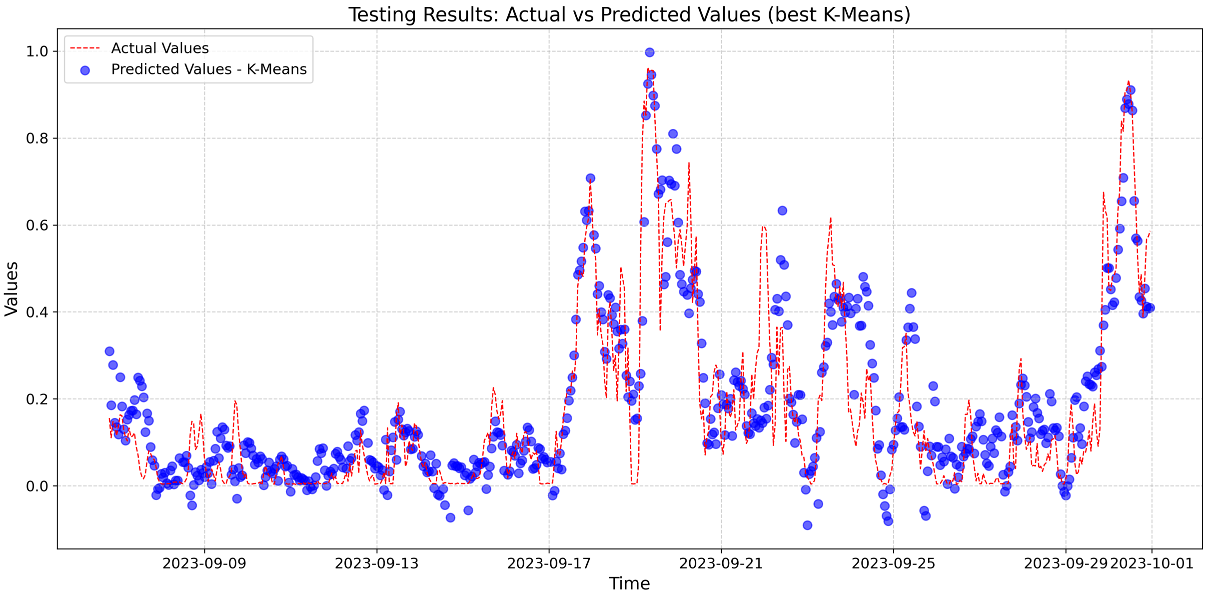

Figure 13: Comparison between predicted and actual values

In the end, the best performance achieved resulted in an RMSE of 0.1061. After analyzing the results, it can be concluded that clustering the data before applying a regression technique does not necessarily lead to improved prediction performance. Clustering the data prior to applying Weighted Least Squares can lead to overfitting, as the data is divided into smaller groups. Consequently, the amount of data available to train each regression model for the clusters becomes limited, which may negatively impact the model's generalization ability.