MÁSTER EN BIG DATA Y ANALÍTICA AVANZADA

ASSESSMENT 1

Machine Learning II

Ensemble

Group 7

Minerva Bermejo Arcos
Ainhoa Zuazola Arregui
Belén Salgado Bottaro
Blanca Martínez Rubio

April 2024

Contents

1 Introduction

In the realm of renewable energy, particularly solar power, understanding and predicting the usage of energy resources is crucial for efficient energy management and planning. This project focuses on two key metrics: irradiation and utilization.

Irradiation refers to the amount of solar energy received per unit area and is a critical factor in determining the potential energy output of solar power systems. It varies throughout the day based on geographical location, time of year, and weather conditions. Accurately measuring irradiation allows for the estimation of how much solar power will be available at any given time.

Utilization, on the other hand, measures how effectively this solar power is used. In the context of this project, utilization is quantified over three-hour intervals throughout the day, providing insights into the energy consumption patterns and efficiency of the system.

1.1 Objectives of the Study

The primary goal of this project is to estimate hourly utilization for a day using the available data on irradiation and utilization. The specific objectives are:

1. **Employ Ensemble Techniques:** To predict utilization, ensemble methods will be employed, which combine multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone.
2. **Comparison with Direct Methods:** These ensemble methods will be compared against more direct or straightforward statistical or machine learning methods to assess their efficacy and accuracy.
3. **Honest Validation:** To select the best approach among those tested, honest validation will be conducted. This involves using a portion of the data not seen during the model training phase to evaluate model performance, ensuring that the evaluation is unbiased and reflective of real-world performance.
4. **Highlighting Poorly Estimated Cases:** Part of the analysis will focus on examples where the estimation significantly deviates from actual utilization. These cases will provide valuable insights into the limitations and potential areas of improvement for the predictive models.
5. **Provision of Intermediate Results:** Throughout the analysis, intermediate results will be shared to demonstrate the progression in model training and tuning, which supports transparency and provides checkpoints for understanding model behavior.
6. **Preliminary Exploratory Analysis:** An exploratory analysis of the data will be conducted before delving into complex modeling. This will include visualizing the data distributions, checking for outliers, and understanding correlations among the variables. This analysis will provide a foundational understanding of the dataset.

2 EDA

2.1 Preprocessing

The data were collected from two CSV files, one with radiation values and one with utilization values. In both, each row refers to a different day, from 2015-01-01 to 2020-12-31, five full years. Both have the following columns: Fecha, ANNO, MES, DIA, DIASEM, and 8 columns with ration/utilization values at different times of the day: 0, 3, 6, 9, 12, 15, 18 and 21.

1. **Reading CSVs into a Pandas dataframe:** Each CSV has been loaded into a different dataframe, and in both the FECHA column has been converted from string format to datetime, to facilitate later manipulation.
2. **DataFrames union:** Both dataframes have been combined using the Pandas merge function, using the columns DATE, YEAR, MONTH, DAY and DAYTIME as merge keys.
3. **Data preparation:** With the aim of getting a single column for Radiation and another one for Usage, each of its 8 values (for the 8 different hours) has been put in 8 different rows, and the column 'FranjaHoraria' has been created, to indicate to which hour the value belongs. To do so, the Pandas melt function has been used.
4. **Lags calculation:** In order to consider them as inputs to train the models, several lags or delays of both radiation and utilization have been calculated and added as new columns of the dataframe. More specifically, the values of radiation/utilization 3 hours back (lag1), 6 hours back (lag2), 1 day back (lag8) and 1 year back (lag365) have been considered.
5. **Data cleaning:** All rows containing null values in the final DataFrame have been removed.
6. **Data transformation:** Finally, ANNO, MONTH, DAY, DIASEM and FranjaHoraria columns have been converted to categories for later use in machine learning models.

The final result is a dataframe of size 14616 x 15, with no null values, whose columns are: YEAR, MONTH, DAY, DAYSEM, TimeSlot, Irradiation, Utilization, Irradiation_lag1, Irradiation_lag2, Irradiation_lag8, Irradiation_lag365, Utilization_lag1, Utilization_lag2, Utilization_lag8 and Utilization_lag365.

2.2 Time Series

In order to be able to represent Irradiance and Utilisation values over time as Time Series, a new column Fecha_Hora of type timestamp has been created. This column is the result of the sum of the 'Fecha' and 'Hora' columns, of type string, and a subsequent transformation of the sum to type timestamp. The 'Hora' column has been created by adding the substring '00:00' to the 'FranjaHoraria' column, which has been previously converted to string, and filled with 0 on the left so that it always has 2 characters.

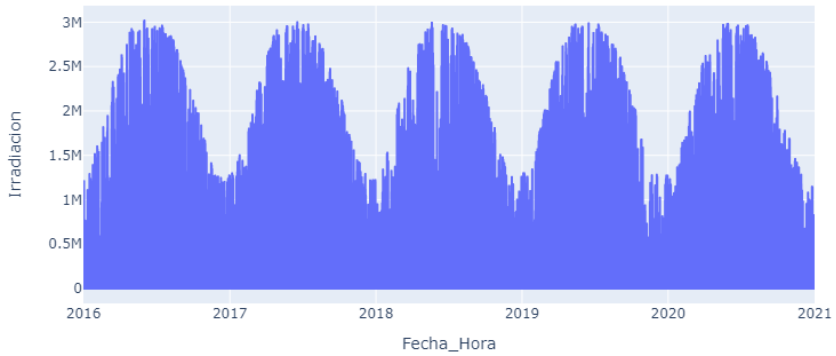


Figure 1: Irradiation Time Series

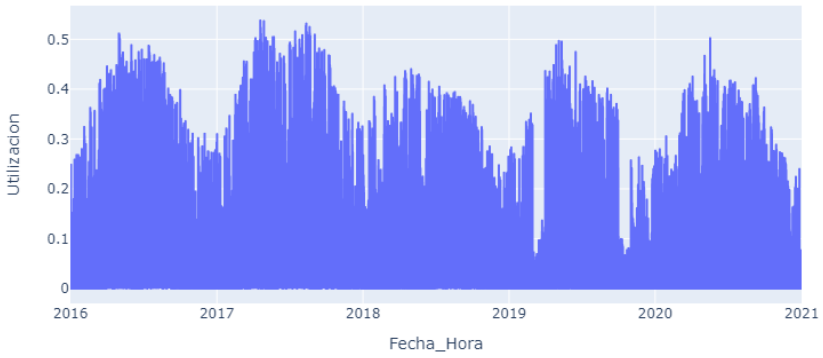


Figure 2: Utilization Time Series

Irradiance appears to have a seasonal pattern, with peaks that could correspond to the summer months of each year, which would make sense if it were related to solar irradiance. There are clear cycles that repeat with an annual periodicity. This suggests that irradiance has a predictable behavior over time and could be modeled with techniques that take seasonality into account.

The second graph shows the variable "Utilization" over the same time period. This series also shows variations over time and could have seasonality, although the patterns are not as marked and regular as in the irradiance series. The utilization varies between 0 and a little over 0.5, and appears to have periods of increase and decrease, although these are not as consistent in magnitude and time as those of the irradiance.

2.3 Main Stats

Statistic	Irradiacion	Irradiacion_lag1	Irradiacion_lag2	Irradiacion_lag8	Irradiacion_lag365
count	1.461600e+04	1.461600e+04	1.461600e+04	1.461600e+04	1.461600e+04
mean	4.721051e+05	4.721051e+05	4.721051e+05	4.722018e+05	4.726322e+05
std	7.310293e+05	7.310293e+05	7.310293e+05	7.310110e+05	7.348932e+05
min	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
25%	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
50%	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
75%	7.408039e+05	7.408039e+05	7.408039e+05	7.413618e+05	7.311085e+05
max	3.017838e+06	3.017838e+06	3.017838e+06	3.017838e+06	3.017838e+06

Table 1: Descriptive Statistics for Irradiation Variables

Statistic	Utilizacion	Utilizacion_lag1	Utilizacion_lag2	Utilizacion_lag8	Utilizacion_lag365
count	14616.000000	14616.000000	14616.000000	14616.000000	14616.000000
mean	0.076575	0.076575	0.076575	0.076602	0.081055
std	0.115443	0.115443	0.115443	0.115446	0.119549
min	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000405	0.000405	0.000405	0.000405	0.000422
50%	0.008861	0.008861	0.008861	0.008861	0.011759
75%	0.119412	0.119412	0.119412	0.119429	0.129854
max	0.537553	0.537553	0.537553	0.537553	0.537553

Table 2: Descriptive Statistics for Utilization Variables

It is observed that Irradiation statistics are generally higher than Utilization statistics, suggesting that, on average, more solar energy is received than utilized.

On the other hand, the variability in Irradiation levels appears to be greater than the variability in Utilization levels, which may indicate that the solar energy available varies more than the solar energy utilized.

2.4 Histograms

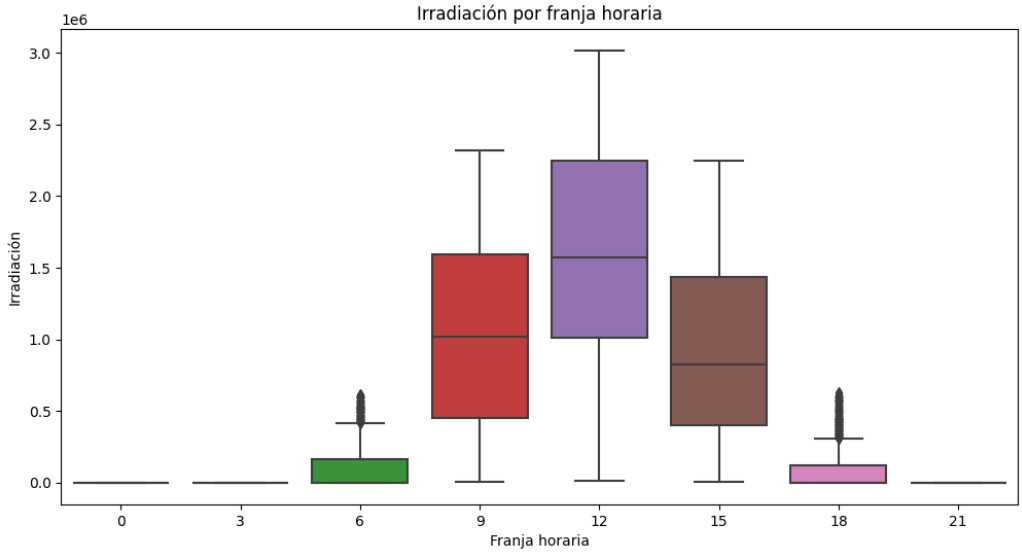


Figure 3: Irradiation by time slot

The histogram above represents the distribution of solar irradiance along different time slots of the day. The following observations can be made:

- The central time slots of the day (9, 12, 15) show the highest values of irradiance, as would be expected because these are the hours where the sun is highest in the sky and therefore there is more solar energy available.
- The 6 o'clock time slot appears to have significantly lower median irradiance compared to the central hours, which is consistent with dawn or the beginning of the day when solar intensity begins to increase.
- The 0 and 21 o'clock bands show very low values of irradiance, which makes sense as they correspond to the evening or twilight periods where there is little or no sunlight.
- The extreme values, represented by the points outside the boxplot whiskers, are more frequent during peak irradiance hours, suggesting variability in the amount of irradiance possibly due to weather conditions such as cloudiness.

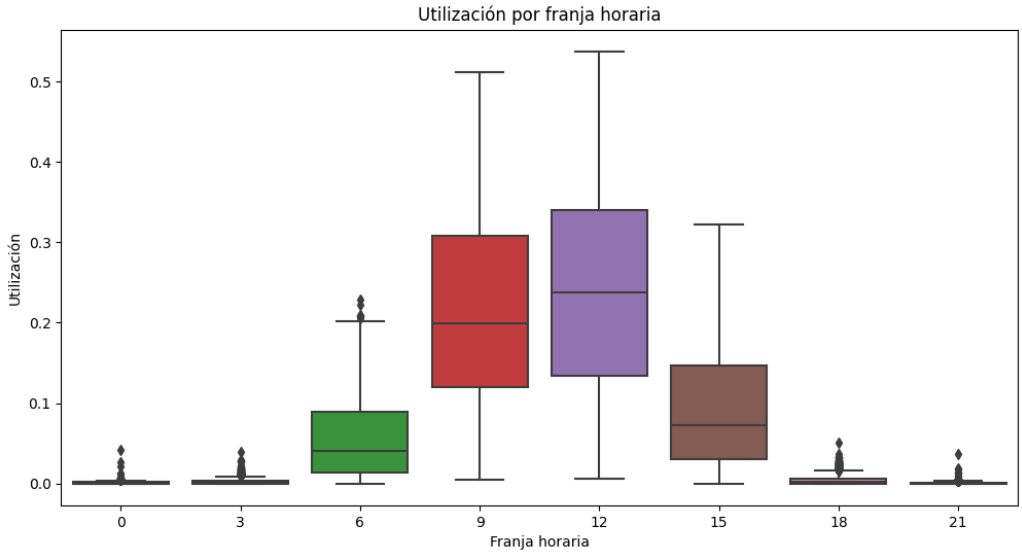


Figure 4: Utilization by time slot

This second histogram shows the distribution of solar energy utilization throughout the day, and reveals the following:

- Similar to the irradiance graph, the mid-day time slots (9, 12, 15) exhibit higher median values of utilization, which may reflect higher consumption or utilization of solar energy when it is more abundant.
- In contrast to irradiance, the morning (6 o'clock slot) and afternoon (18 o'clock slot) hours also exhibit considerable utilization, although not as high as in the middle of the day. This may indicate that, although there is less irradiation during these hours, the system is still taking advantage of the available energy.
- The night hours (0 and 21) show minimal utilization, which is expected since there is little or no solar irradiance to convert to energy.
- The presence of extreme values is less prominent in the utilization graph compared to the irradiance graph, but is still noticeable in the peak hours.

2.5 Pairplot

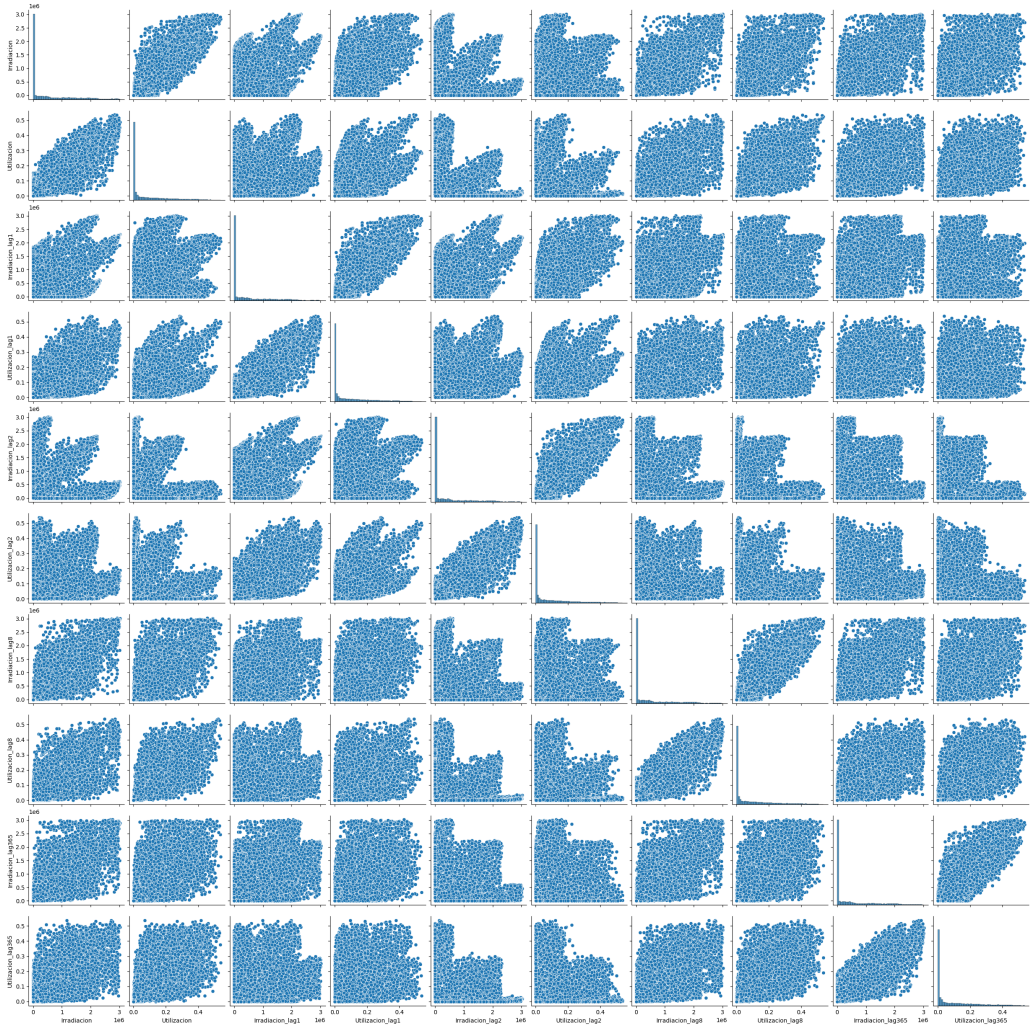


Figure 5: Pairplot of all inputs and output

The pairplot shows a linear relationship between the Irradiance and Utilisation variables both at the present time and with values from 3 hours, 6 hours, 1 day and 1 year ago (lags 1, 2, 8 and 365, respectively). On the other hand, graphs representing a distribution with three peaks are observed, suggesting the possible interaction of a third categorical variable influencing the relationship between these two variables. However, to be able to affirm these assumptions would require further exploration.

2.6 Correlation Matrix

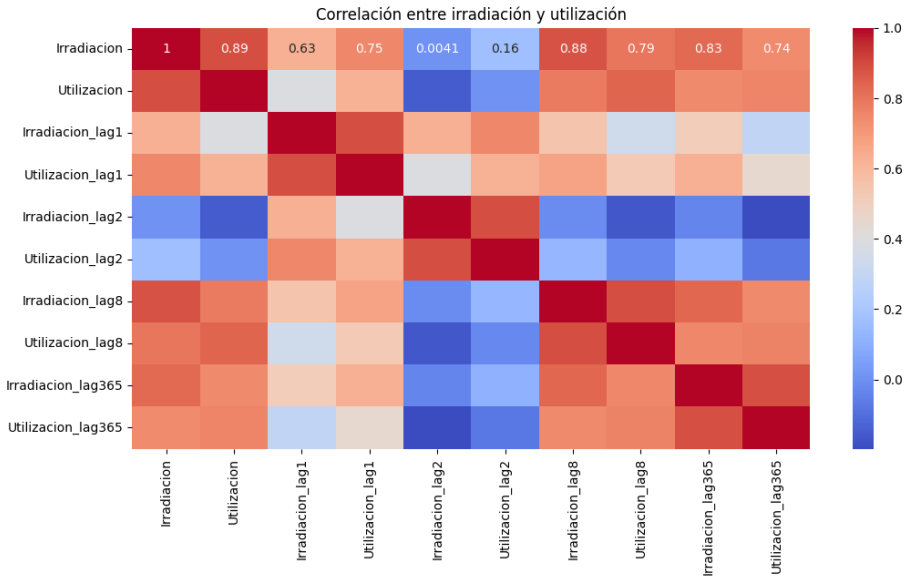


Figure 6: Correlation matrix of all inputs and output

The figure above represents a correlation matrix between the different input variables (Irradiance, Irradiance lags and Utilization lags) and the output variable (Utilization). A range of colors is used where red shades represent high positive correlation and blue shades represent high negative correlation. Values near 1 or -1 indicate a strong correlation, while values near 0 indicate weak or no correlation.

- The first cell shows a perfect correlation, as expected, since it compares the variable 'Irradiance' to itself.
- 'Irradiance' and 'Utilization' have a fairly high correlation of 0.89, suggesting that as irradiance increases, utilization also tends to increase.
- The lag values represent irradiance and utilization on previous days. For example, 'Irradiance_lag1' would be the irradiance on the previous day.
- The 'Irradiance' and 'Utilization' variables maintain a relatively high correlation with their own lagged values, especially at short lags such as 'lag1' and 'lag2'.
- Interestingly, there appears to be a weak negative correlation between 'Irradiance_lag365' and 'Utilization', as well as between 'Utilization_lag365' and 'Irradiance', which could indicate an inverse relationship on an annual cycle.

Besides the above, it is important to mention that there might be strong non-linear relationships that we are not seeing, as the correlation matrix only shows linear relationships between variables.

2.7 Scatter plot

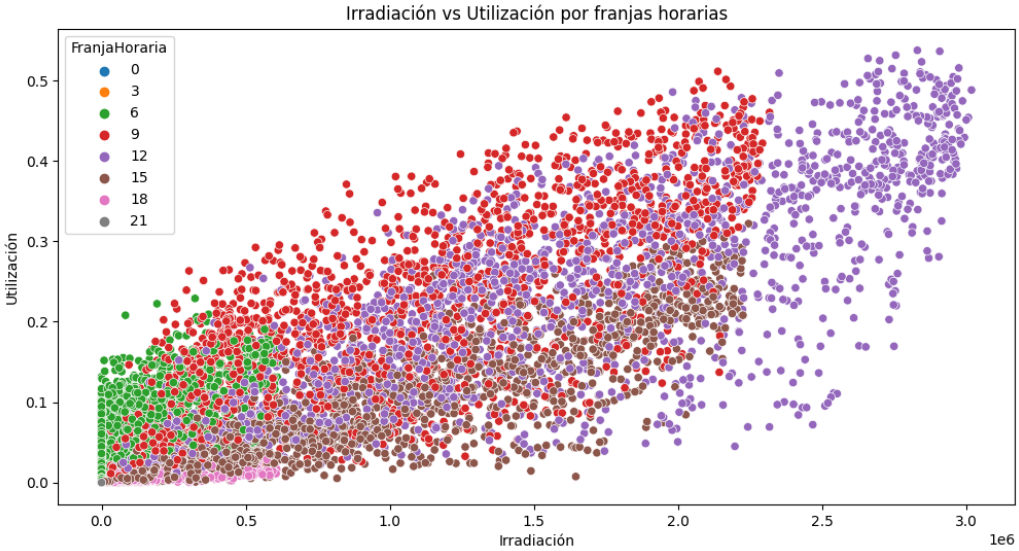


Figure 7: Scatter plot: Irradiation vs Utilization by time slots

The scatter plot above shows the relationship between irradiance and utilization across different time slots of the day. The dots are colored by hour, allowing us to see how this relationship behaves at different times of the day. The following observations can be made.

- **Positive Relationship:** There is a general trend suggesting a positive relationship between irradiance and utilization; that is, as irradiance increases, so does utilization. This is consistent with what we would expect in a system where solar energy is converted and utilized or stored.
- **Differences by Time Zone:** The earliest (00, 03) and latest (21) time slots have, in general, lower irradiance and utilization, which makes sense given that these times are near sunrise and sunset or during the night. The central time slots (06, 09, 12, 12, 15, 18), which correspond to daytime hours, show significantly higher irradiance and utilization. This indicates power generation during the hours of maximum sunlight.
- **System Saturation or Limitations:**Some points with high irradiance do not have proportionally high utilization, which could indicate that the system reaches its maximum capacity utilization or there is a threshold beyond which additional energy is not efficiently utilized.
- **Variability in Utilisation:** Although irradiance appears to increase in a continuous and concentrated manner, utilisation shows greater variability. This could be due to system efficiency, varying operating conditions, or the presence of other factors that affect how the generated energy is consumed.

- **System Efficiency:** In the time slots with the highest irradiance (probably midday), there appears to be a wide dispersion in utilisation, which could suggest that system efficiency varies. For example, some systems may be better designed or located and therefore use irradiance more efficiently.
- **Outliers:** Some outliers, especially those with low irradiance and high utilisation, or vice versa, may be of interest for further investigation, as they could indicate system anomalies or measurement errors.

In summary, this graph provides a good overview of how irradiance and utilisation are correlated and how this correlation varies at different times of the day. To draw more definitive conclusions, it would be useful to consider the specific system configuration, local conditions, and other operational or environmental factors that could influence these results.

3 Direct Methods

In this section, we explore two direct approaches for modeling the relationship between input features and target variable: Simple Regression and Multi-Layer Perceptron (MLP).

3.1 Simple Regression

Simple Regression was employed to model the linear relationship between the predictor variables and the utilization. The input variables used in the regression model were preprocessed using StandardScaler for numeric variables and OneHotEncoder for categorical variables. Grid Search Cross Validation was employed to fine-tune the model's parameters for optimal performance.

Following this, an analysis of the Variance Inflation Factor (VIF) was conducted to assess multicollinearity among the predictor variables. The VIF measures the extent to which the variance of the coefficient estimates is amplified due to multicollinearity in the model. It was found that four variables exhibited VIF values greater than 10:

Variable	VIF Factor
num__Irradiacion	10.8
num__Irradiacion_lag1	18.5
num__Irradiacion_lag2	13.1
num__Irradiacion_lag8	11.6

Table 3: Variables with VIF Values Greater Than 10

The remaining variables had VIF values below 10, indicating that multicollinearity is not a significant concern in the model. Further steps may be taken to address multicollinearity, such as removing highly correlated variables or applying dimensionality reduction techniques.

Additionally, the following metrics were calculated to evaluate the model:

	MAE	RMSE	R^2
Training	0.02026	0.03253	0.92585
Test	0.02094	0.03238	0.91937

Table 4: Simple Regression Performance Metrics

Error metrics provided insights into the average deviations and the consistency of the model's accuracy. Meanwhile, the R^2 value offered a statistical measure of how closely the predicted values correspond to the actual values, giving a sense of the model's explanatory power. This multi-metric approach ensures a robust understanding of the model's effectiveness in making accurate predictions.

The model exhibits promising performance based on the evaluation metrics. With a Training Mean Absolute Error (MAE) of 0.02026 and a Test MAE of 0.02094, the model demonstrates low average deviations from the actual values. Additionally, the Train RMSE of 0.03153 and the Test RMSE of 0.03238 indicate that the model's predictions have relatively small errors on average.

Moreover, the high Training R^2 value of 0.92585 and the Test R^2 of 0.91937 suggest that approximately 92.6% and 91.9% of the variance in the target variable is explained by the model, respectively. This indicates that the model performs well in capturing the underlying patterns in the data. It's important to note that while the Training R^2 is slightly higher than the Test R^2, suggesting a potential slight overfitting of the model to the training data, the difference is relatively small. Therefore, the model demonstrates good generalization ability to unseen data despite the presence of mild overfitting.

The final step involved studying the residuals, which are the differences between the predicted and actual values, and help identifying any systematic patterns or biases in the model's predictions.

The dispersion of the residuals does not exhibit a discernible pattern; they appear to be randomly scattered around zero. This suggests that the model has effectively captured the relationship between the independent and dependent variables. Furthermore, the variance of the residuals remains constant, indicating that the model's predictive errors are consistent across the range of input values. Additionally, there are no outliers present in the residual plot, and the residuals are centered around zero.

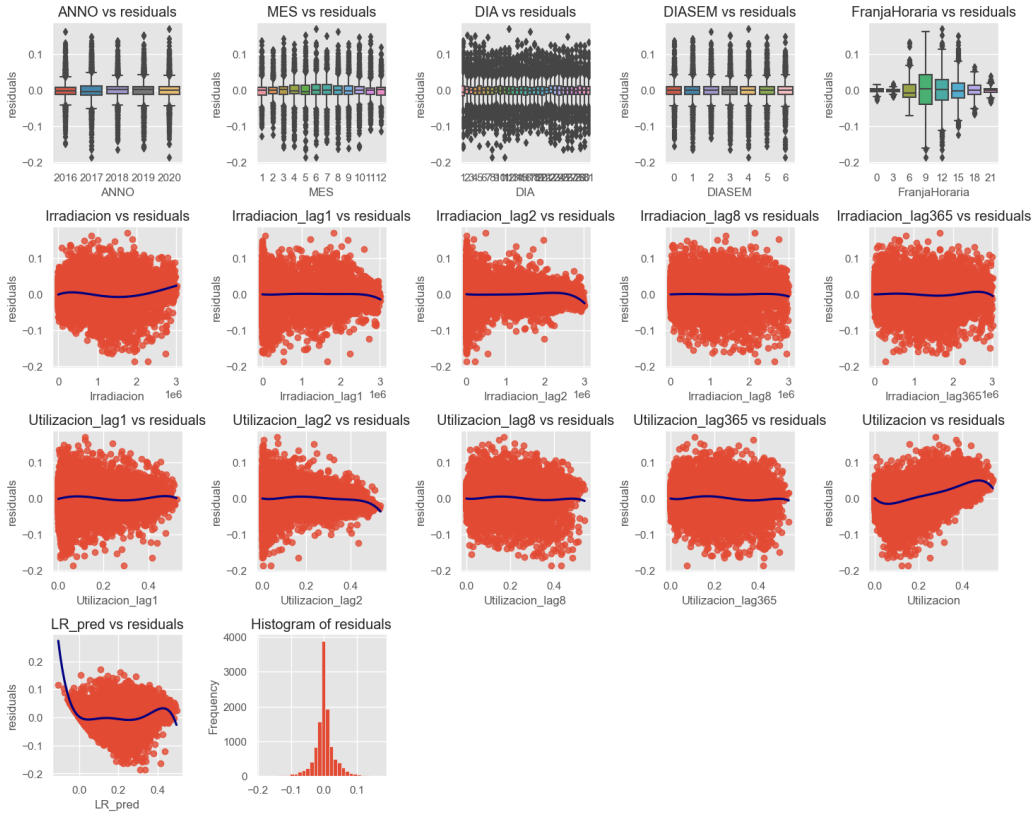


Figure 8: Residuals of Linear Regressor

In this case, as the output values represent very small quantities, when assessing the model's residuals, it's essential to consider their relative magnitude compared to the actual data values. Absolute residuals may appear insignificant when evaluated alone, but relative to the small scale of the output values, even small residuals could have a notable impact. Therefore, calculating relative residuals allows us to better understand the true extent of the model's predictive errors relative to the size of the data values, ensuring a more accurate evaluation of the model's performance.

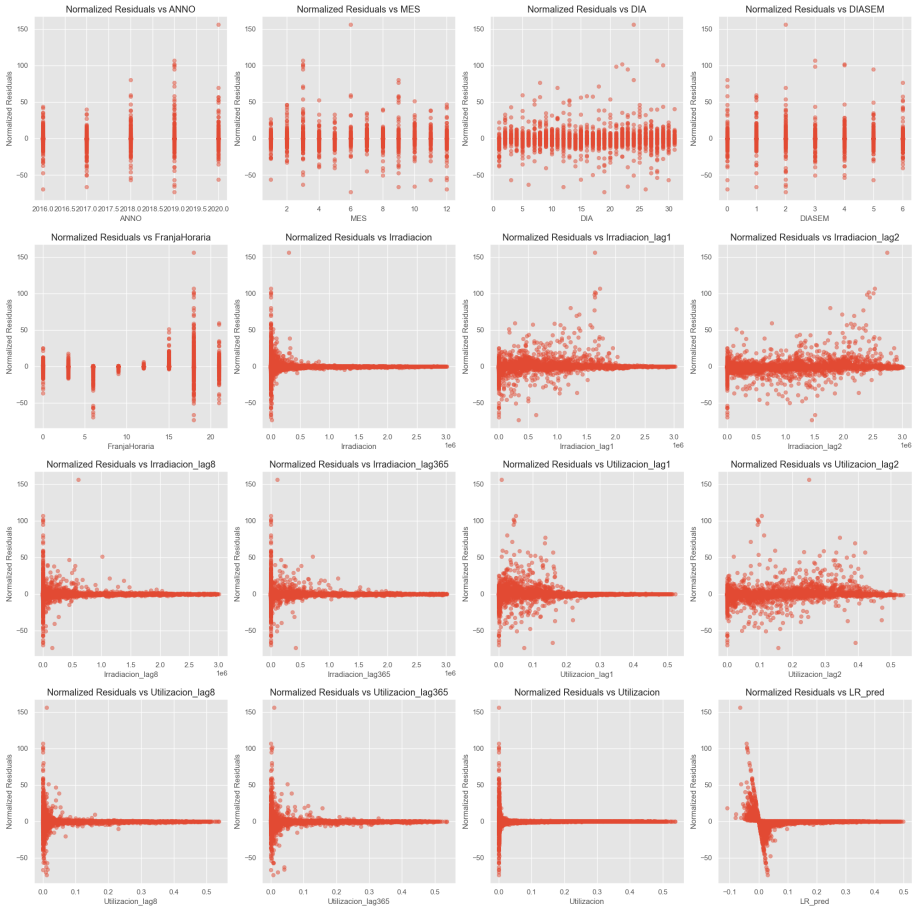


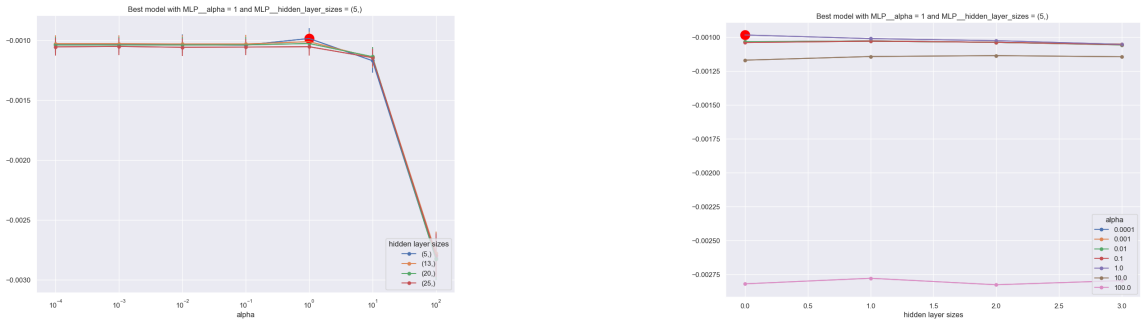
Figure 9: Normalized residuals of Linear Regressor

The residuals exhibit significant dispersion, with points far from the horizontal line at zero, indicating inaccurate model predictions for some values close to zero. Additionally, patterns are observed near zero values of the predictors, where the spread of residuals increases as the predictor value approaches zero. The presence of very high or very low residuals suggests potential outliers in the data that the model is not handling well. These findings underscore the need to address potential issues with outliers, non-constant variance of errors, and possibly model fit, especially for values close to zero, in order to improve the accuracy and reliability of the model.

3.2 Multi Layer Perceptron (MLP)

A Multi-Layer Perceptron (MLP) was employed to capture the complex non-linear relationships between the input variables and the target variable, "Utilization". Prior to model training, preprocessing was performed for the numerical and categorical variables. The numeric variables were standardized using StandardScaler, while categorical variables were encoded using OneHotEncoder. Cross-validation was then utilized to fit the hyperparameters of the model, negative mean squared error was used as the scoring metric to assess the model's performance.

The hyperparameters explored included the regularization parameter ('alpha') and the number of neurons in each hidden layer ('hidden_layer_sizes'). The optimal result was achieved with alpha = 1 and hidden_layer_sizes = (5,).



(a) Alpha

(b) Hidden Layer Sizes

Figure 10: Grid search results

Additionally, to assess the performance of the model, several metrics were calculated. The calculated metrics include Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared (R^2).

	MAE	RMSE	R^2
Training	0.01874	0.03037	0.93149
Test	0.01948	0.031327	0.92454

Table 5: MLP Performance Metrics

These results indicate that the model performs well in both the training and test sets. The mean absolute error (MAE) values are relatively low. Similarly, the root mean squared error (RMSE) values are also low, indicating small deviations between the predicted and actual values. The R^2 values further confirm the model's effectiveness.

In order to visually assess this effectiveness on performance of the Multi-Layer Perceptron (MLP) model in predicting utilization based on irradiation levels, scatter plots were created to compare the model's predictions with the actual utilization values for both the training and testing datasets. The scatter plots depict the relationship between irradiation and utilization, with actual utilization

values represented by black dots and the MLP model's predictions indicated by red dots with black borders.

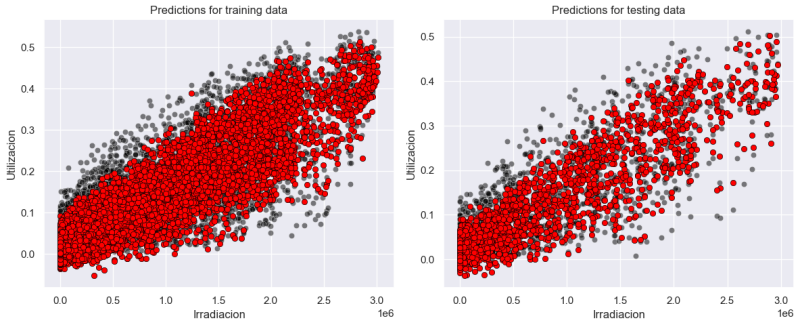


Figure 11: Predictions with the Multi-Layer Perceptron

A plot displaying the residuals is provided below. The residuals appear randomly scattered around zero, indicating that the model has effectively captured the relationship between the independent and dependent variables. Consistent variance across input values suggests stable predictive errors. Moreover, the absence of outliers and the centering of residuals around zero highlight the model's reliability.

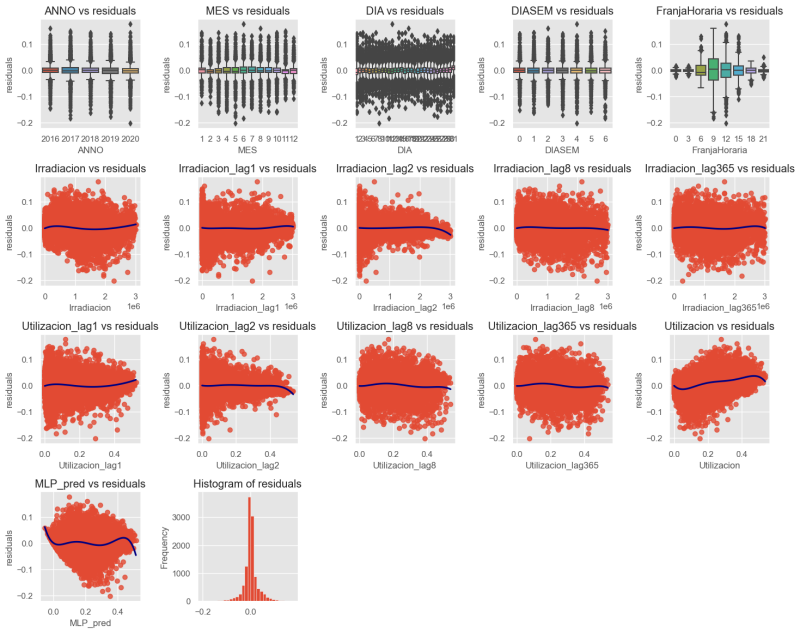


Figure 12: Residuals of Linear Regressor

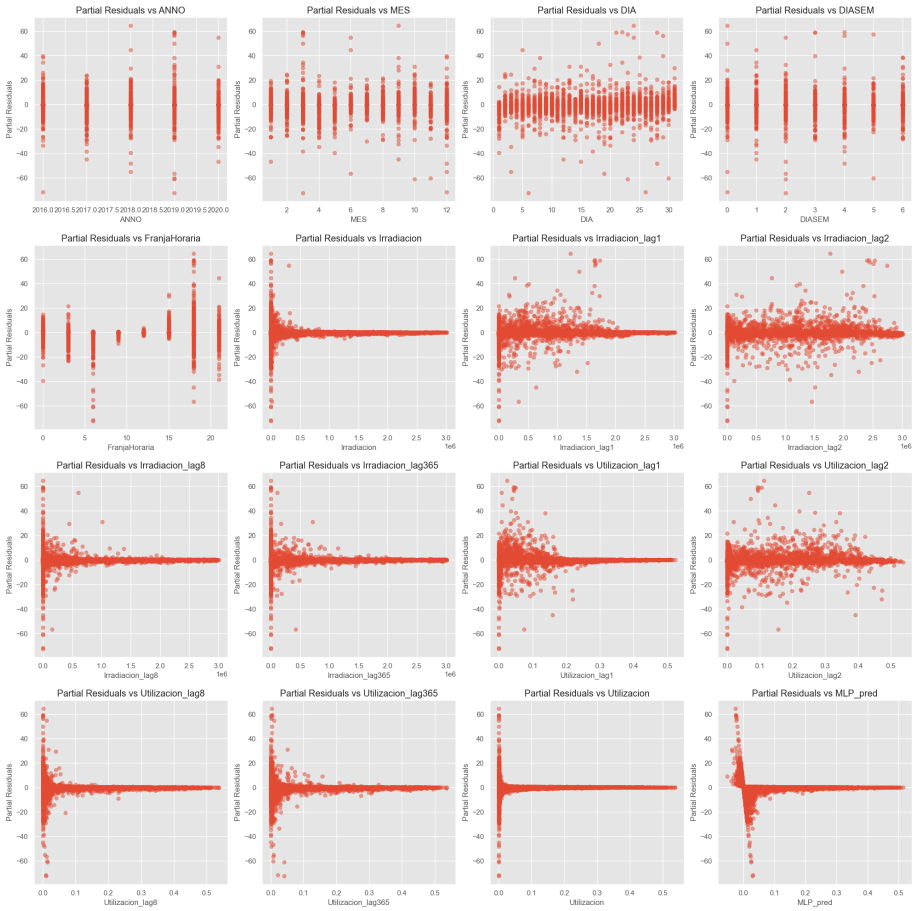


Figure 13: Normalized residuals of MLP

Just as observed with the Linear Regressor, these insights emphasize the necessity of thoroughly analyzing and resolving any discrepancies in the model's residuals. It is imperative to address issues such as outliers, varying error variances, and potential model misalignment to enhance the precision and dependability of the model's forecasts.

4 Ensemble

After having conducted a basic regression analysis of the problem, ensamble techinques were applied for estimating hourly utilizations and comparing the results.

4.1 Bagging

Initially, Bagging, also known as Bootstrap Aggregation, was performed to enhance the accuracy of single models by reducing their variance through averaging their predictions. Following data preprocessing, an initial base regression tree estimator was constructed with specific hyperparameters tailored to our case via Grid Search Cross Validation:

- Impurity measure: Squared Error
- Minimum Impurity Decrease: 0.001
- Minimum number of observations in node to keep cutting: 2
- Minimum number of observations in a terminal node: 9

The performance of this base model is summarized in the following table:

	MAE	RMSE	R^2
Training	0.01076	0.02075	0.96787
Test	0.01641	0.03137	0.92429

Table 6: Regression Tree Performance Metrics

Subsequently, bagging was employed to further enhance the performance of the single regression tree and reduce the overfit indicated by the slight difference in the coefficient of determination (R^2) between training and test. Through Grid Search Cross Validation and utilizing negative Mean Absolute Error (MAE) as the scoring criterion, the optimal number of trees to aggregate was determined to be 55.

However, the performance metrics of the bagging model, presented in Table 7, fell short of expectations. The Mean Absolute Error and Root Mean Squared Error increased, and the R^2 value decreased compared to the base regression tree model. While the base regression tree explained 92% of the variance in the test data, the bagging model only accounted for 72%.

	MAE	RMSE	R^2
Training	0.03479	0.05911	0.73936
Test	0.03564	0.05943	0.72835

Table 7: Bagging Performance Metrics

Furthermore, while studying the residuals and partial residuals (Figures 14 and 15), which help identifying any systematic biases in the model's predictions, it became evident that the model was not adequately capturing the underlying patterns within our dataset. As it can be seen in figure

14 the residuals are not distributed along the horizontal axis and clear biases can be observed in some of the pairplots. The decrease in the model's performance after having done bagging might be produced by using initial trees which are too similar or not having chosen the correct base estimator.

On the other hand, the partial residual plots (Figure 15) exhibit a concentrated number of points around zero. However, the model seems to be underestimating some values and there are numerous residuals with a consistent magnitude of -20. This deviation suggests systematic errors, the model seems to be missing some important patterns in the data.

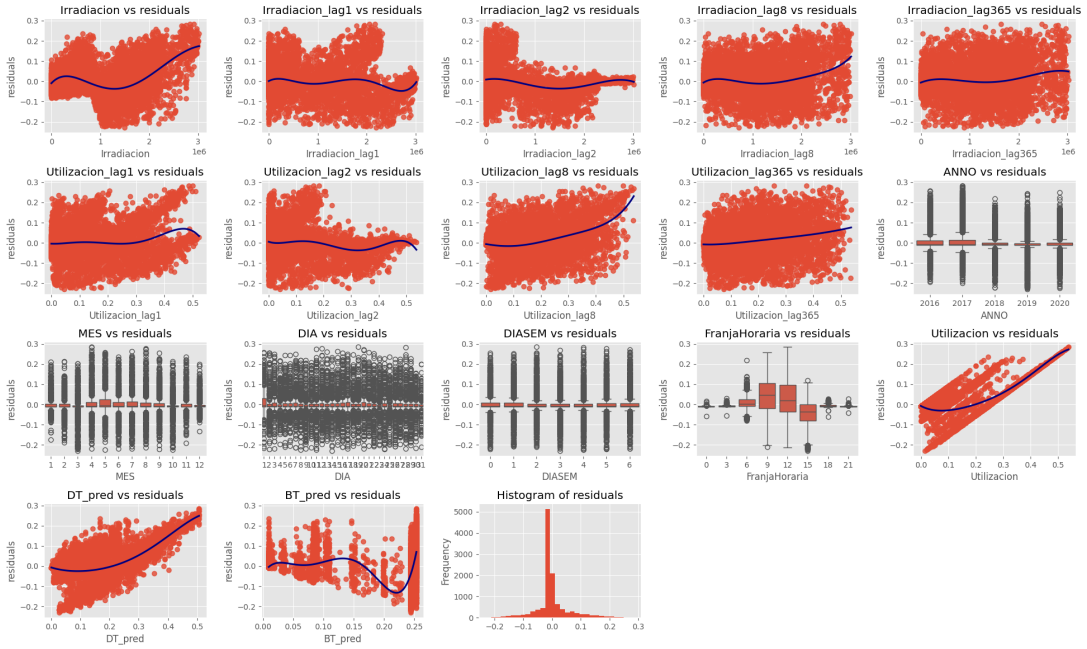


Figure 14: Residuals of Bagging

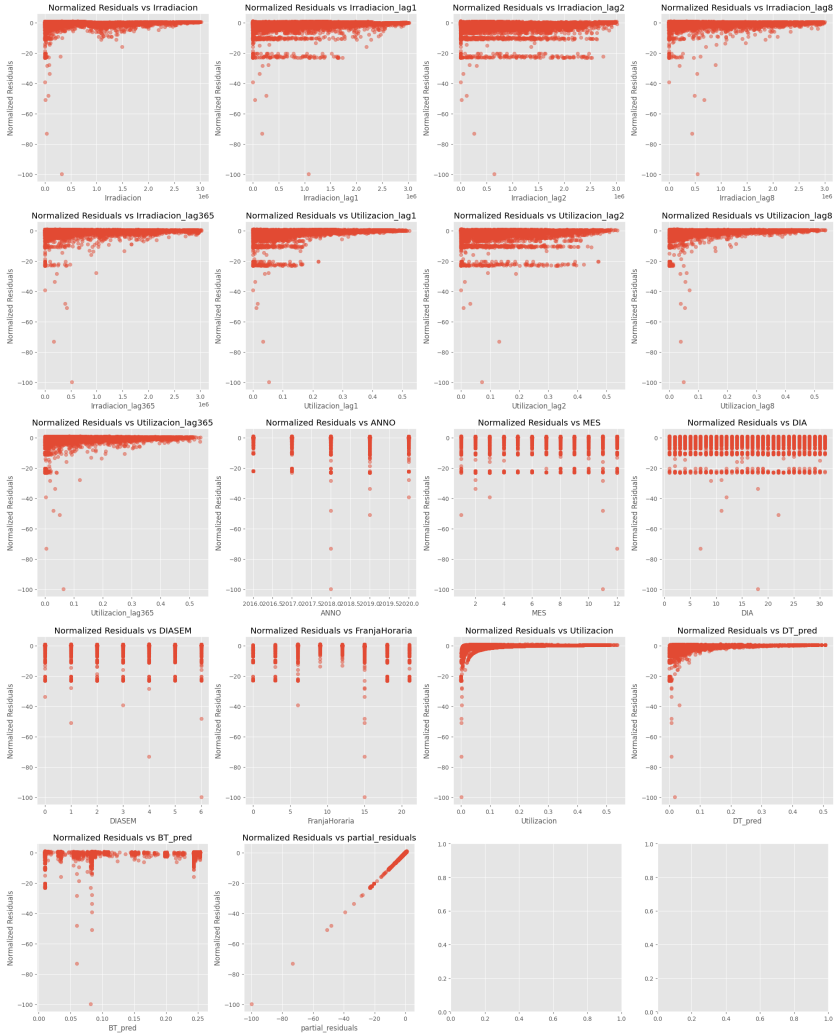


Figure 15: Normalized residuals of Bagging

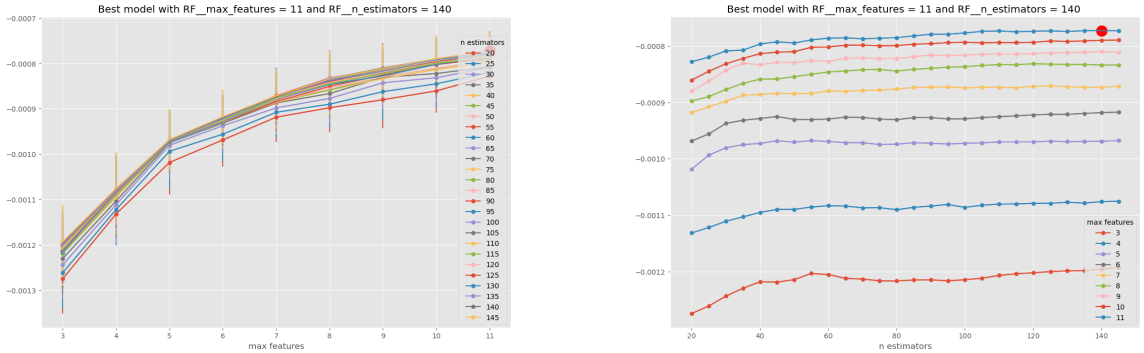
Given the decreased performance of the bagging model compared to the single tree estimator, the decision was made to discard this ensemble model.

4.2 Random Forest

Random forests is a substantial modification of bagging that builds a large collection of decorrelated trees, and then averages them. They appear to outperform bagging in many cases by enhancing its variance reduction capabilities. This improvement is achieved by lowering the correlation among the individual (bootstrap) trees, without significantly raising the variance.

Given this understanding, a random forest model was developed using a one hot encoder for categorical variables and a standard scaler for numerical variables. To identify the optimal parameters, a grid search with 10 folds was performed on the maximum features and number of estimators. The parameters for min_samples_split and min_samples_leaf were set at 5 and 1, respectively, which are commonly used values.

The results from the grid search indicate 11 estimators and 140 maximum features, as illustrated in Figure 16.



(a) Maximum features

(b) Number of estimators

Figure 16: Grid search results

After generating predictions with this model, a variety of metrics were utilized to assess performance, quantify errors, and evaluate the coefficient of determination (R^2). The performance evaluation was comprehensive, examining both the precision of the predictions and their alignment with the observed data.

	MAE	RMSE	R^2
Training	0.00709	0.01323	0.98695
Test	0.01504	0.02762	0.94136

Table 8: Random Forest Performance Metrics

The model shows a modest discrepancy between training and test performance, hinting at a slight overfitting. While the R^2 values remain high for both datasets, the increase in MAE and RMSE for the test data suggests the model fits the training data slightly better than it does with unseen data. This level of overfitting is not extreme, but it's worth monitoring and could potentially be reduced with further model tuning.

The next step is analyzing the residuals of the model.

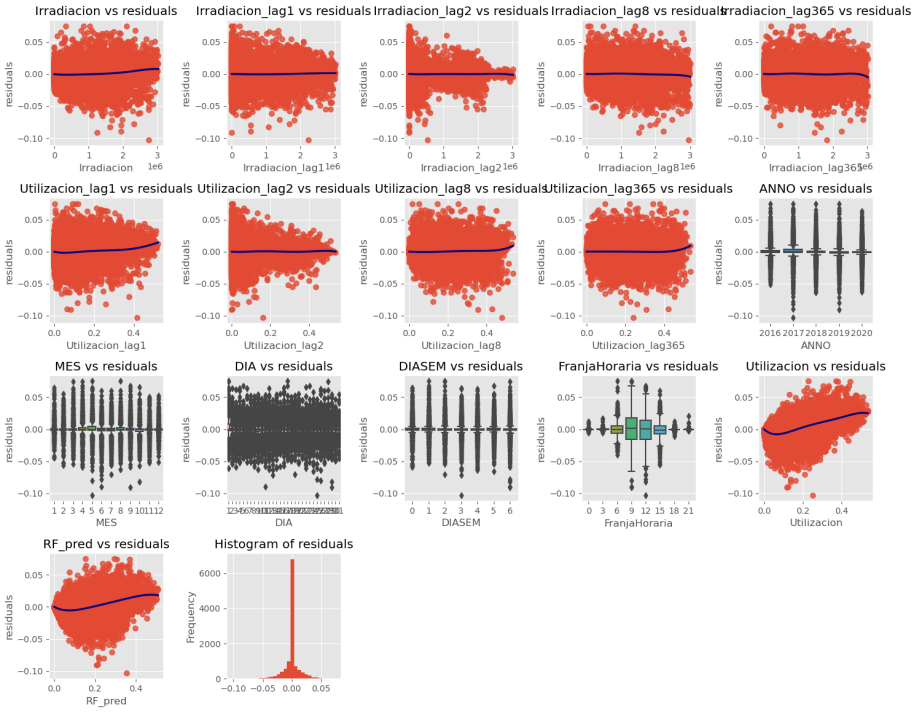


Figure 17: Residuals of Random Forest

Figure 17 reveals that the residuals are well-dispersed, not excessively large, and centered around zero, suggesting the absence of model bias. However, it is essential to evaluate the residuals relative to the magnitude of the data values to fully assess their significance. Residuals may appear small in absolute terms, but they can be substantial if the data values themselves are around similar magnitudes.

To address this, the analysis includes normalized residuals to provide a clearer picture of the error's impact in proportion to the data's scale. The results are shown in Figure 18

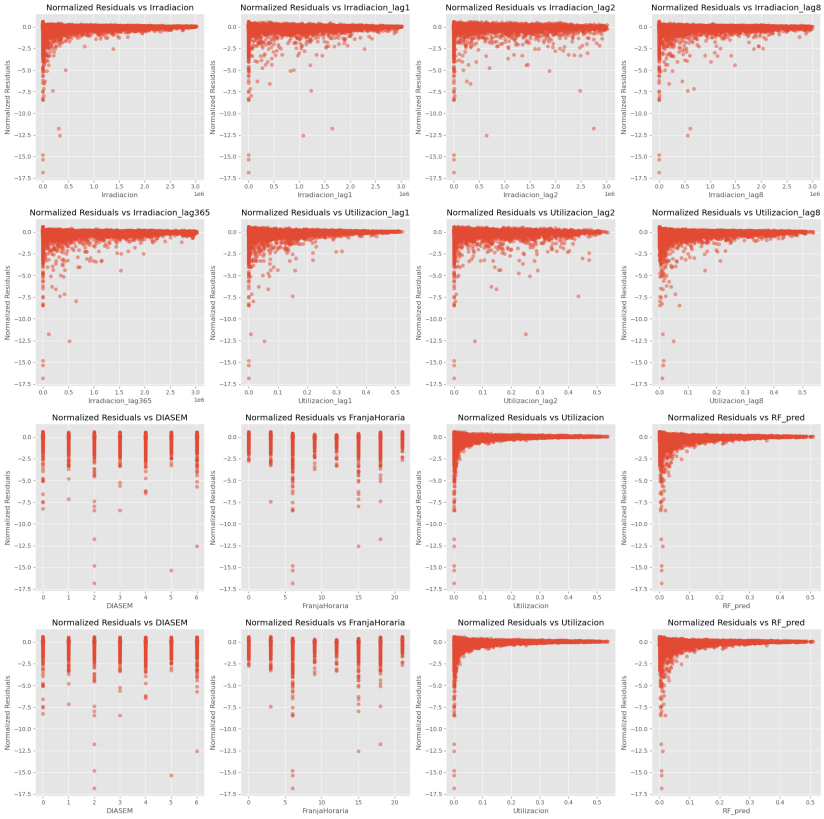


Figure 18: Normalized residuals of Random Forest

The partial residual plots exhibit a concentrated number of points around zero, suggesting no significant bias in the model. However, the presence of some notable outliers, particularly in the irradiation features, indicates instances where the model's predictions are far from the actual values.

For the utilization variables, the residuals show a narrower spread, which might indicate a variance in prediction accuracy compared to the irradiation features.

The time-based features reveal some periodic patterns in the residuals, hinting at potential seasonal effects or cyclic trends that the model may not be capturing. This is especially visible in the plots against days and months.

In general the residuals shows a tendency for underestimation at lower predicted values.

4.3 Boosting

Boosting is one of the most powerful learning ideas introduced in the last two decades. Like bagging, it is a general approach that can be applied to many statistical learning methods for regression or classification. The motivation for boosting was a procedure that combines the outputs of many "weak" classifiers to produce a powerful "committee".

4.3.1 Ada Boost

Ada Boost is the most popular boosting algorithm due to Freund and Schapire. It is an ensemble method that aims to improve the classification performance by combining multiple weak learning models. A weak learner is a model that performs slightly better than random guessing. By iteratively adjusting the weights of misclassified instances, AdaBoost focuses on the examples that previous models found challenging, thereby adapting to the unique characteristics of the data.

In order to train the model, a grid search was conducted, which determined that the optimal parameters include a maximum depth of 10, a minimum impurity decrease of 0, and a total of 100 estimators.

With this parameters, the following results are obtained:

	MAE	RMSE	R^2
Training	0.00748	0.01021	0.99222
Test	0.01518	0.02580	0.94881

Table 9: Ada Boost Performance Metrics

The model exhibits strong performance on the training data, as indicated by the low values of MAE and RMSE, along with a high R-squared (R2) value, close to 1. These metrics suggest that the model captures a large proportion of the variance in the target variable and makes accurate predictions on the data it was trained on. However, the larger MAE, RMSE, and lower R2 values observed on the test data compared to the training data suggest a potential issue with overfitting.

A further analysis into the residuals, shown in Figure 19 indicates that the values are centered at 0, therefore the model is not biased. However, it can be observed that in the first and last values, the line representing the mean of the residuals is not completely straight, so there are values that will not be properly predicted. This is especially noticeable in the plots of Utilization vs residuals, DT_pred vs residuals and AB_pred vs residuals. Generally speaking, it seems that the model is good, although worse than others that have been fitted. Nevertheless, the normalized residuals will be analyzed to get a better idea of the magnitude of the residuals.

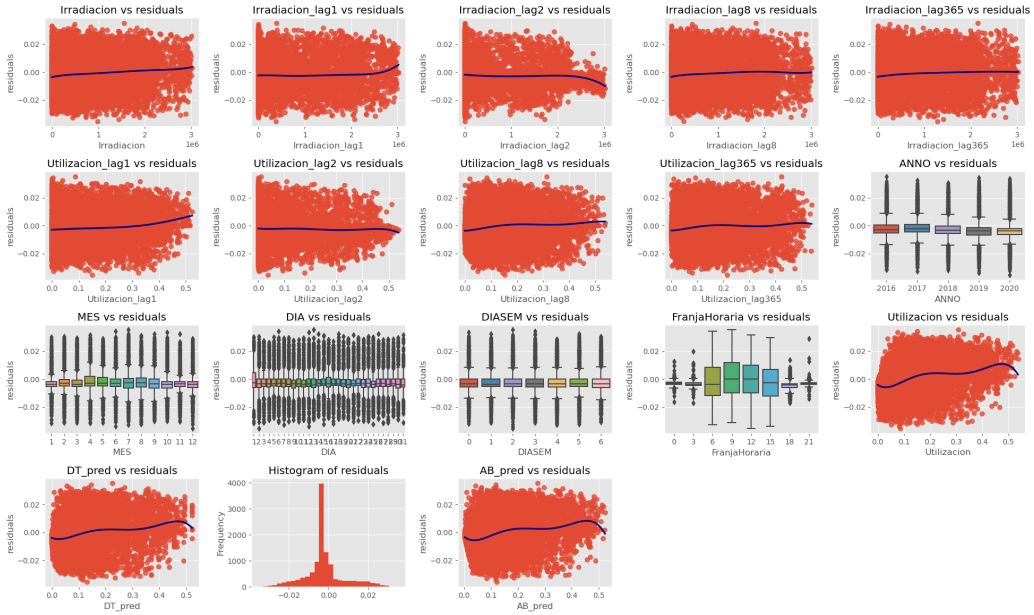


Figure 19: Residuals of Ada Boost

The figure below displays the normalized residuals of the Ada Boost model. The presence of numerous residuals with a magnitude around -20, particularly those far from zero, can be problematic as it indicates that the model's predictions may be significantly lower than the actual values. This deviation suggests systematic errors, the model seems to be missing some important patterns in the data or not accounting for some unusual but influential points.

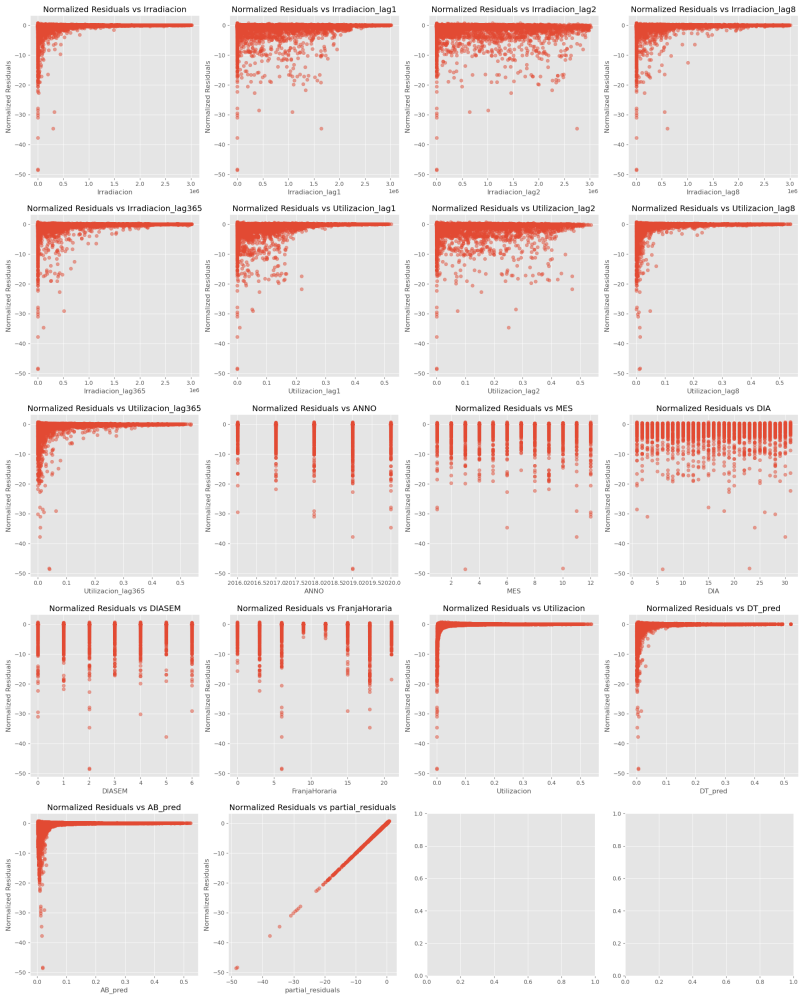


Figure 20: Normalized residuals of Ada Boost

4.3.2 Gradient Boosting

Gradient boosting, similar to AdaBoost, is an ensemble learning technique where weak learners are sequentially combined to form a strong learner. However, the key difference lies in how they build the ensemble. While AdaBoost focuses on adjusting the weights of incorrectly classified instances to prioritize them in subsequent iterations, gradient boosting takes a gradient descent approach. Specifically, gradient boosting fits each new weak learner to the residuals or errors of the previous ensemble, thereby placing more emphasis on correcting the mistakes made by the existing model. This iterative refinement process allows gradient boosting to handle more complex relationships within the data and often leads to improved performance compared to AdaBoost, especially in scenarios with noisy data or high-dimensional feature spaces.

A Gradient Boosting Regressor model was trained using a grid search approach to optimize its hyperparameters. The optimal parameters identified through this process include a learning rate of 0.01, a maximum depth of 5, a minimum impurity decrease of 0.000134, and 3000 estimators. These parameters were selected based on their ability to minimize the loss function and improve the model's predictive accuracy.

To assess the performance of the model obtained with these parameters, several metrics have been calculated.

	MAE	RMSE	R^2
Training	0.00783	0.01377	0.98586
Test	0.01347	0.02536	0.95056

Table 10: Gradient Boosting Performance Metrics

The model demonstrates excellent performance, with low MAE and RMSE values, and high R^2 scores for both training and test datasets, indicating accurate predictions and strong generalization capability. The slight overfitting observed is negligible, given the high performance metrics and minimal discrepancy between training and test results.

Examining the residuals will offer further insights into how well the model is performing.

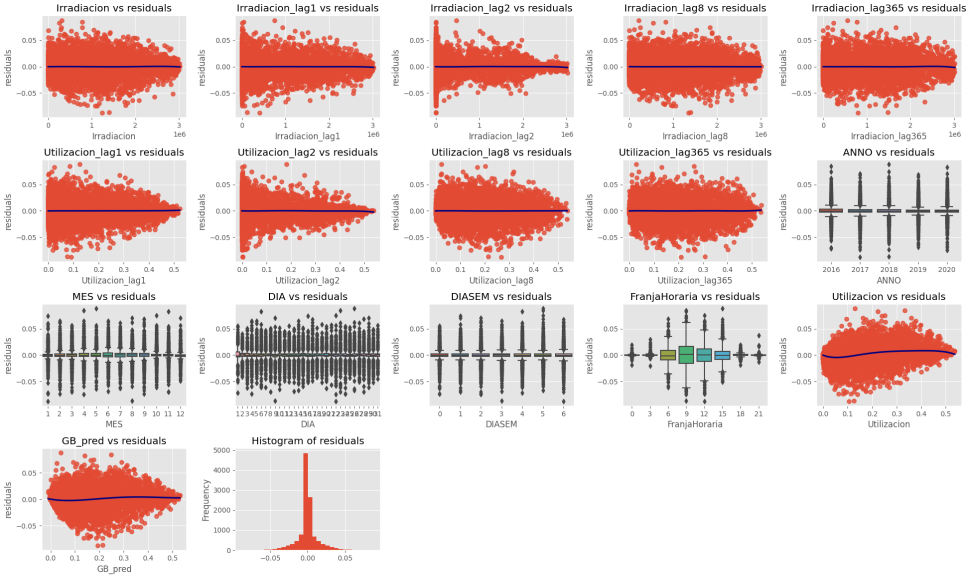


Figure 21: Residuals of Gradient Boosting

The residuals show no discernible pattern and are evenly scattered around zero, suggesting the model captures the relationship well. Consistent variance across input values and absence of outliers further support model effectiveness.

Nevertheless, as it has been explained before, it is essential to assess normalized residuals to understand the magnitude of the residuals relative to the variability of the response.

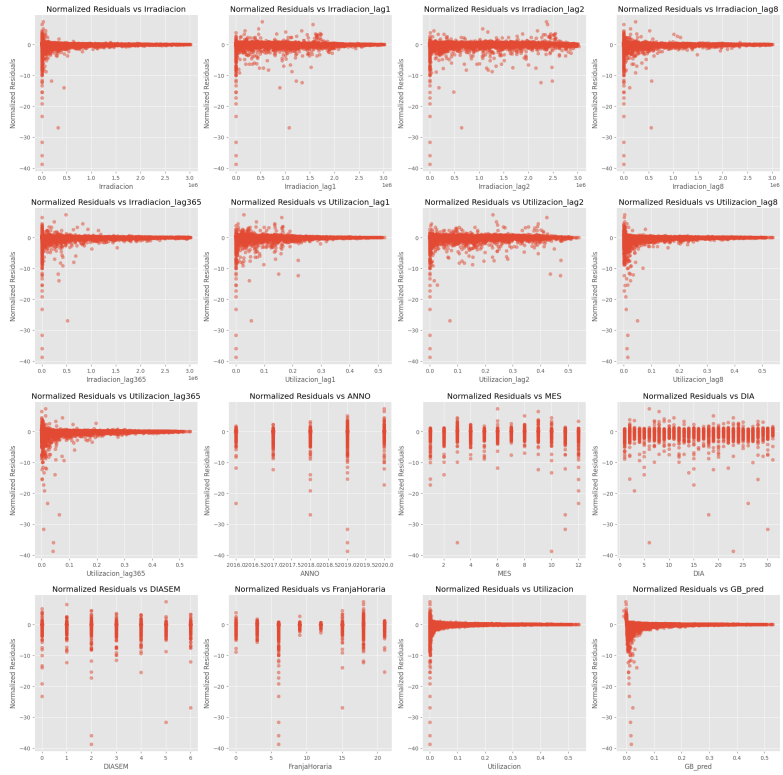


Figure 22: Normalized residuals of Gradient Boosting

The plot illustrates the normalized residuals of the Gradient Boosting model. In spite of the improvement on normalized residuals over direct methods, the model still exhibits some pronounced errors when values are close to zero, indicating potential systematic errors or overlooked patterns in the data. These deviations may suggest that the model inadequately accounts for these influential data points.

4.3.3 Extreme Gradient Boosting (XGBoost)

XGBoost is a version of gradient boosting that incorporates regularization to enhance model performance. It is well-known for its highly efficient implementation, which provides significant computational speed and the capability to scale up to handle larger datasets.

Unlike traditional Gradient Boosting Machines (GBMs) that build trees sequentially, XGBoost employs a parallelized approach to tree construction. Additionally, XGBoost uses standard regularization parameters to control both the size of the trees and the magnitude of the weights, further refining the model's accuracy and complexity.

An XGBoost model was optimized using a grid search technique, which identified the best parameters as 1000 estimators, a maximum depth of 6, and a learning rate of 0.01.

As with previous cases, various error metrics and R^2 have been evaluated to assess the model's performance.

	MAE	RMSE	R^2
Training	0.00873	0.01595	0.98103
Test	0.01340	0.02529	0.95081

Table 11: XGBoost Performance Metrics

The model exhibits better accuracy on the training data, which is a typical sign of overfitting. Although the differences in performance metrics between the training and test datasets point towards this issue, the model still shows a strong ability to explain a significant portion of the variance in both datasets, as reflected by the high R^2 values. This indicates that despite some overfitting, the model maintains a robust predictive capacity.

An analysis of the residuals will provide more insights about the model's performance.

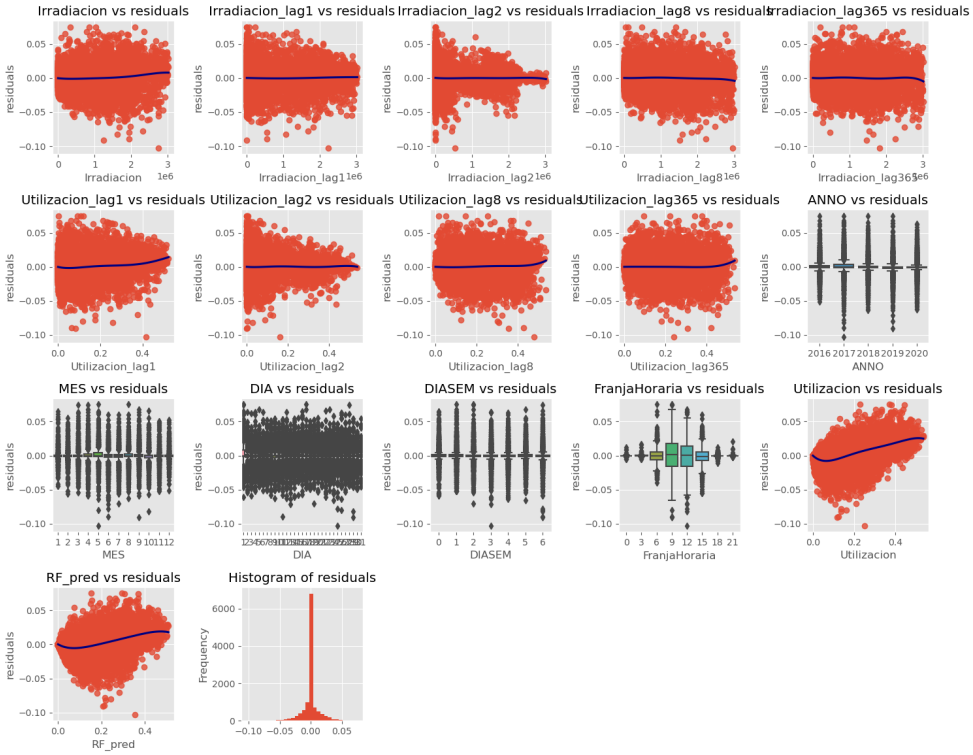


Figure 23: Residuals of XGBoost

Overall, the concentration of residuals around zero across different features suggests the model has a decent fit for the data, without bias. However, the spread and patterns observed in certain plots imply that there are specific conditions under which the model's predictions deviate more from the actual values. The presence of systematic patterns in residuals associated with time-based features could indicate that temporal dynamics are influencing the model's performance.

The histogram's peak near zero is promising but the skewness indicates a bias in the prediction errors.

However, normalized residuals must be evaluated to assess the magnitude of the residuals in terms of the response variability.

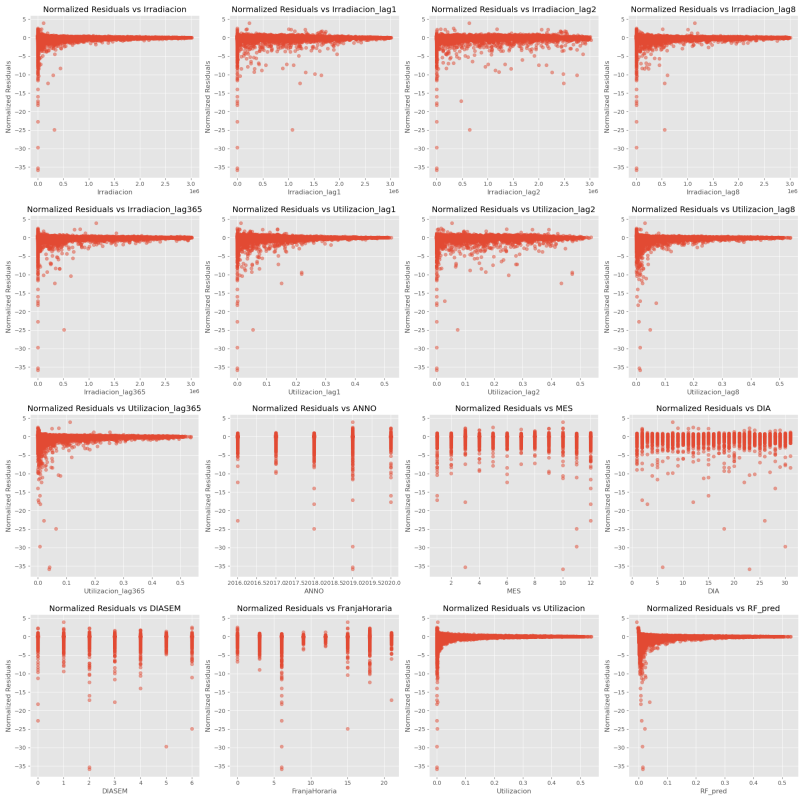


Figure 24: Normalized residuals of XGBoost

The plots in Figure 24 suggest that while the model generally captures the central trend of the data, there are noticeable deviations, especially at lower predictor values. Most residuals cluster around zero but with a substantial spread up to approximately -10 in magnitude, which indicates significant underestimation in certain areas. The model's consistency varies across different features.

4.4 Stacking

Finally, the Stacking ensemble method was integrated. The idea behind this approach is to combine different types of Machine Learning models, equally well fitted for the same whole dataset, in order to balance out their individual weaknesses. The final output is trained through cross-validation to best combine the base-models and obtained by averaging the predictions of the stacked models.

In our particular scenario, the following models were stacked together using VotingRegressor from scikit-learn, considering the corresponding weights provided below.

- Polynomial model: weight 0.1

	MAE	RMSE	R^2
Training	0.01490	0.02530	0.95225
Test	0.01583	0.02665	0.94535

Table 12: Polynomial Model Performance Metrics

- B-spline model: weight 0.2

	MAE	RMSE	R^2
Training	0.01881	0.02923	0.93622
Test	0.01978	0.03105	0.92584

Table 13: B-Spline Model Performance Metrics

- Simple Regression Tree: weight 0.7

	MAE	RMSE	R^2
Training	0.01025	0.01983	0.97064
Test	0.01652	0.03193	0.92160

Table 14: Regression Tree Performance Metrics

The performance of the final model is summarized in Table 15. As it can be seen, the model demonstrates strong performance on the training data, characterized by low MAE and RMSE, along with high R-squared (R2) value. These metrics suggest that the model captures a large proportion of the variance in the target variable and makes accurate predictions on the data it was trained on. However, the larger MAE, RMSE, and lower R2 values observed on the test data compared to the training data suggest a potential issue with overfitting.

A further analysis into the residuals, shown in Figure 25 indicates that the values are centered at 0, therefore the model is not biased. Nevertheless, consistent with prior observations, it is apparent that the line representing the mean of the residuals is not perfectly straight at the extreme values, suggesting that certain values may not be accurately predicted.

	MAE	RMSE	R^2
Training	0.01074	0.01954	0.97149
Test	0.01541	0.02857	0.93719

Table 15: Regression Tree Performance Metrics

Finally, Figure 26, which illustrates the normalized residuals of the Stacking model, demostrates that the model still exhibits some pronounced errors when values are close to zero.

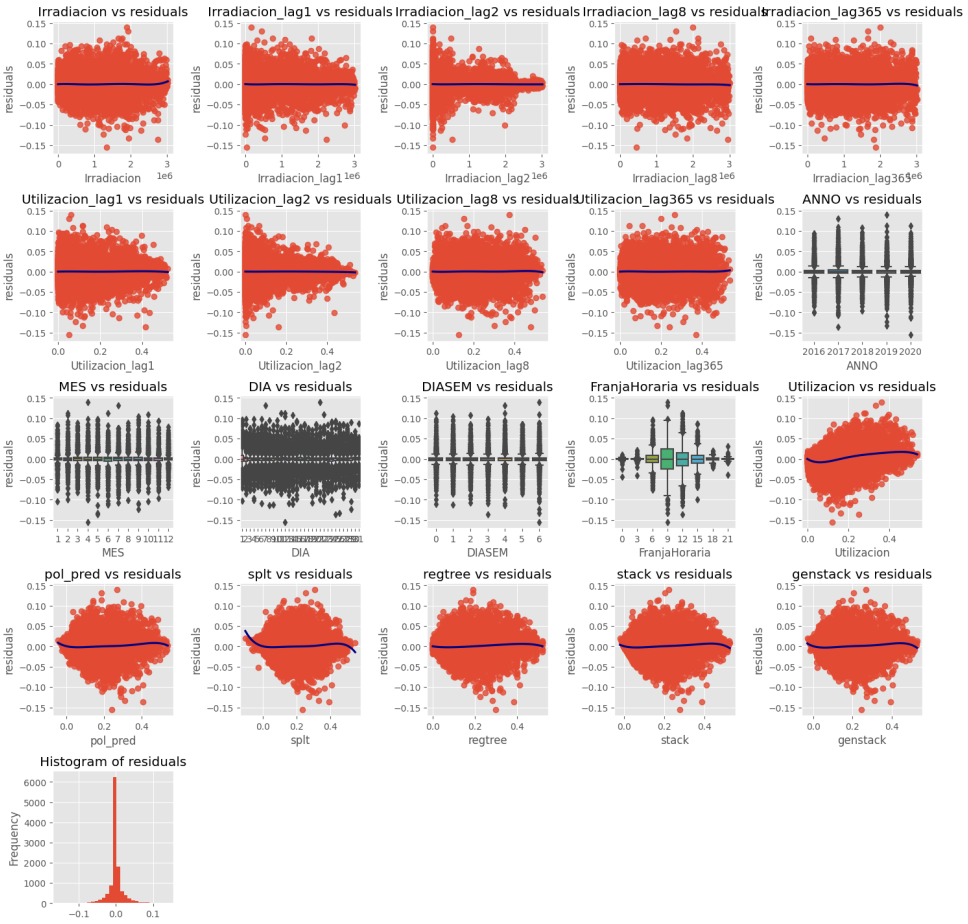


Figure 25: Residuals of Stacking

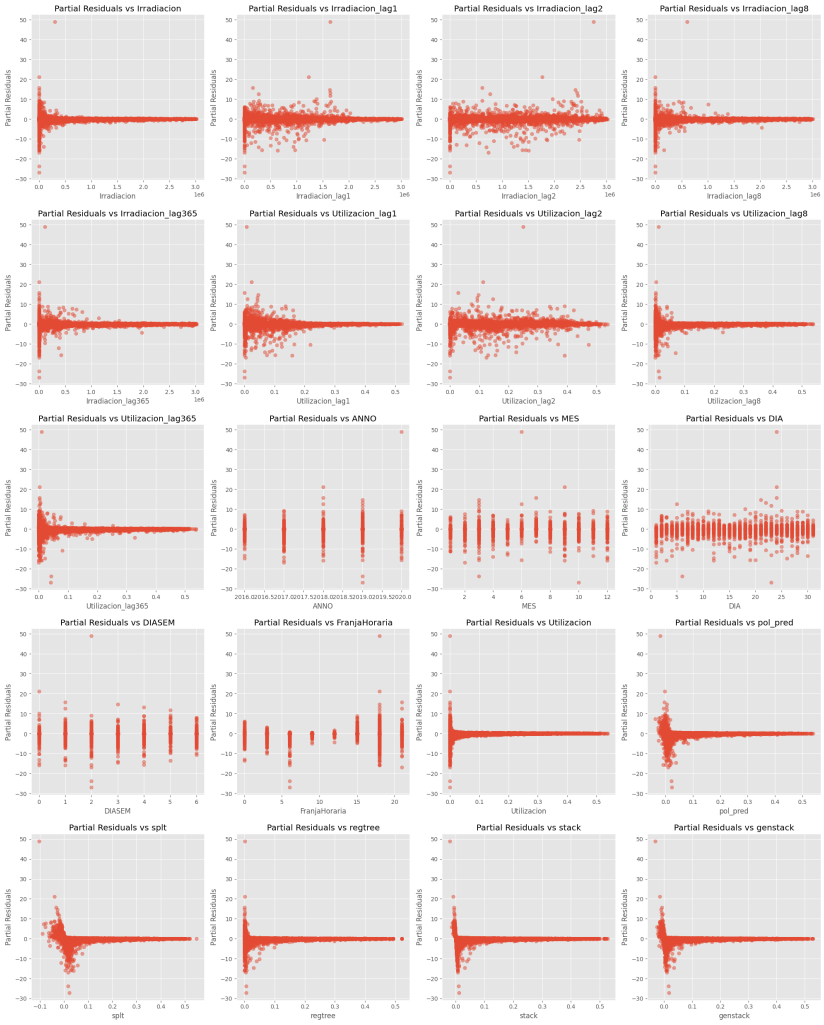


Figure 26: Normalized residuals of Stacking

5 Conclusion

After evaluating the performance of all models, it is observed that both direct models and those using ensemble methods tend to achieve high R2 scores and maintain low MSE and MAE values. These results indicate robust predictive ability and strong alignment between predicted and observed values. The normalized residuals, in particular, show a significant decrease when ensemble methods are applied, suggesting higher prediction accuracy.

These results imply that, although direct models are effective, the application of ensemble techniques substantially improves the quality of predictions. Ensemble methods, by exploiting the strengths of several models, help to reduce prediction errors and improve overall prediction performance. This approach appears to be a sensible option for achieving more reliable and accurate model predictions in a variety of analytical tasks.