AI-Driven Analysis of Photovoltaic Module Efficiency: Predicting Performance and Categorizing Defects

Fábio Rodrigues up202107030 | Sandro Oliveira up202104703 | Tiago Castanheira up202104958

***Abstract*—The following project assesses the operational efficiency of photovoltaic (PV) modules using machine learning techniques, focusing on how common defects affect performance. Before applying the machine learning techniques, the data was pre-processed using one-Hot encoding, label encoding, and standardization. The actual models implemented include several types of regression, random forest, generative classifiers and neural networks. It was tested the impact of reducing the feature dimension using an autoencoder and analyzing the correlation between the features in the dataset.**

***Index Terms*—Photovoltaic Modules, efficiency, machine learning techniques, one-hot encoding, label encoding, normalization, regression, random forest, generative classifiers and neural networks, autoencoder, correlation.**

I. Introduction

The present project aims to assess the operational efficiency of photovoltaic (PV) modules using several machine learning techniques considering how common defects impact performance.

The dataset has several input features most of which are continuous, such as the relative area affected, the temperature, the irradiance, the voltage in open circuit (Vosc) and the current in short circuit (Isc). It also takes into account the PV module type and some categorical deffect indicators: the presence and severity of thermal spots, the amount of obstruction from bird droppings, the level of soiling from dust and the status of the junction box, which constitute the discrete input features. The output efficiency is described both continuously (as Expected Efficiency) and discretely (as Efficiency Level), providing an interpretable summary of the module's performance.

Before applying the machine learning techniques, we needed to prepare the data by transforming the categorical variables into numerical representations (one-Hot and label encoding) and also perform standardization.

To perform this task a broad range of machine learning models were applied:

- Linear, Ridge, Lasso, polynomial and logistic regressions
- Random forest
- Generative classifiers
- Neural networks

In order to evalute the results several metrics were taken into account:

- For continuous outputs: R^2 and RMSE
- For discrete outputs: accuracy, precision, recall and F-1 score
- For neural networks: AUC and PR curves

II. Methodology

A. Dataset Description

The dataset incorporates both categorical and numerical variables, which are essential for assessing PV module efficiency under various conditions. As shown in Figure 1, the categorical variables, such as Hotspot, Birddrop, Soiling, and Junction_Box, represent defects that can affect module efficiency. These are visualized in histograms, displaying the frequency distribution of each defect across the dataset.

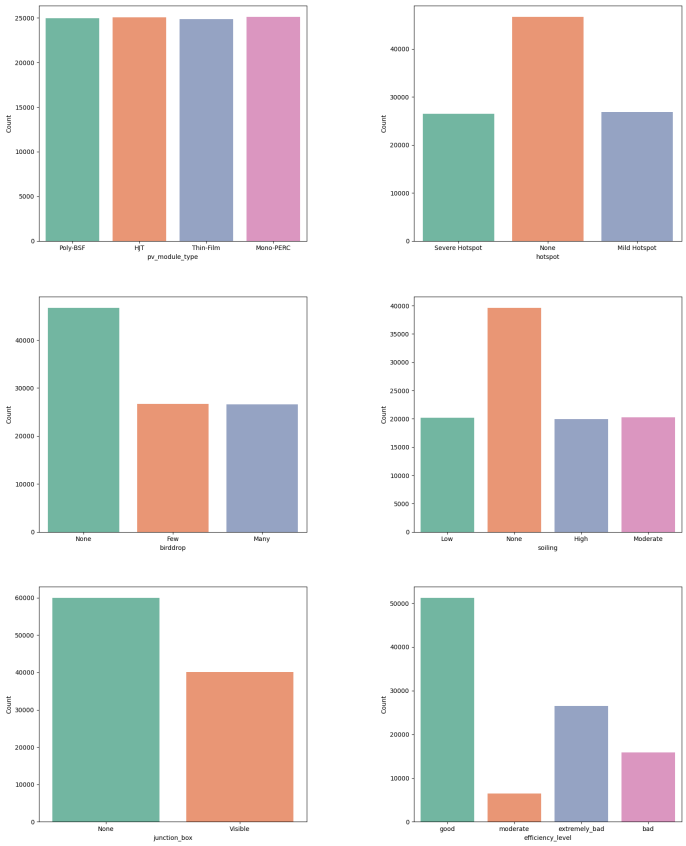

Figure 1: Histograms of the categorical variables in the dataset.

On the other hand, Table I provides descriptive statistics of the numerical variables.

As can be seen, since the mean and standard deviation of the features differ significantly, there is a need for standardization[1]. It was used standard scaler which transforms the features so that they have a distribution with a

Table I: Descriptive statistics of the numerical variables in the dataset

Variable	Mean	Std	Min	Max
affected_area	0.121	0.130	0.000	0.748
temperature	24.988	4.997	3.600	46.600
irradiance	999.820	99.736	582.700	1416.800
Voc	44.904	1.069	41.340	48.170
Isc	10.007	1.571	3.740	16.000
expected_efficiency	0.188	0.033	0.066	0.315

mean of 0 and a standard deviation of 1. The new values are represented in Table II.

Standardization is an important preprocessing step in machine learning, especially when features of the dataset are in different units or have different ranges. Without normalization, models may not perform well since features with larger ranges could dominate the learning process. In this way, these features contribute equally to the model by scaling them; this generally helps to improve both convergence speed and the overall performance of the algorithm. Also, such scaling prepares a model such as a gradient-based model, such as a neural network and linear regression.

Table II: Descriptive statistics of the standardized numerical variables in the dataset

Variable	Mean	Std	Min	Max
affected_area	0.000	1.000	-0.930	4.844
temperature	0.000	1.000	-4.280	4.325
irradiance	0.000	1.000	-4.182	4.181
Voc	0.000	1.000	-3.334	3.055
Isc	0.000	1.000	-3.988	3.814
expected_efficiency	0.000	1.000	-3.740	3.890

B. Data Transformation: One-Hot Encoding and Labeling

To prepare the data for use in machine learning models, the categorical variables were transformed into numerical representations using appropriate encoding techniques. For the pv_module types, which represent categories without an inherent order such as "Monocrystalline," "Polycrystalline," and "Thin-Film," one-hot encoding [2] was applied. This encoding creates binary variables for each category, indicating the presence or absence of that category in a given observation, and to reduce redundancy, the drop_first parameter was used, which excludes one of the binary columns, assuming that if all other columns are zero, the observation belongs to the dropped category.

For categorical variables with an inherent order, such as "Low," "Medium," and "High," label encoding [2] was applied. This technique assigns numerical labels to each category based on their order, where, for example, "Low" might be encoded as 0, "Medium" as 1, and "High" as 2, preserving the ordinal relationship between categories and ensuring that the model interprets the increasing numerical values as a meaningful progression. By transforming categorical variables into numerical formats, these encoding techniques enable machine learning models to process and learn from the data effectively while maintaining the underlying relationships within the dataset.

C. Relationship Between the Data

Given that the dataset features many variables, all of which might contribute differently to predict the target variables of expected_efficiency and efficiency_level, the relationships between features and target variables should be analyzed. This can help in the identification of those features that may contain redundant information so that the model could be simplified, or identify those features which are poorly related to the outputs, their removal would not seriously affect the performance of the models.

According to the dataset description given in Section II a), it consists of various data type measurements including continuous, categorical and ordinal variables. Their analysis needed several correlation methods in order to understand their respective dependencies. The analysis in continuous variable relation is done through the method called Pearson's Correlation[3] assuming a linear relationship (if this would always be valid). To assess the dependence between ordinal variables, Kendall Tau-B[4] was used; because it is effective in cases when the ordinal variable takes only a few distinct values, both are represented in Figure 2. Besides, in order to cover the fact that the relationship might not be linear, Spearman's Rank Correlation[3] was computed to investigate the association between ordinal and continuous variables since it does not depend on the distances between the levels of the ordinal values illustrated in Figure 3.

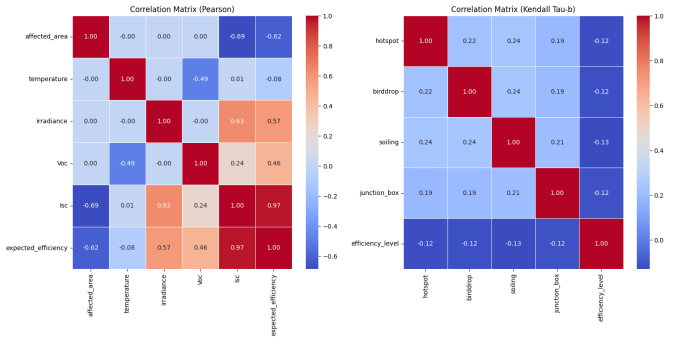

Figure 2: Pearson and Kendall Tau-b Correlation.

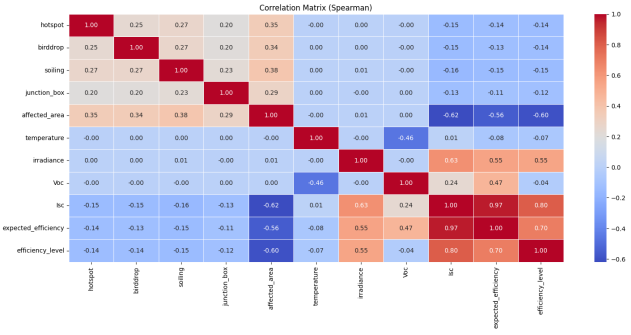

Figure 3: Spearman Correlation.

Since junction_box is a binary variable, the point-biserial correlation[5] was calculated to assess its relationship with expected_efficiency. The correlation value was -0.1124 with a p-value of 1.54e-278, indicating a weak negative relationship.

Specifically, as the value of junction_box changes (from one binary state to another), expected_efficiency tends to decrease slightly, though the association is weak.

To relate the categorical data (pv_module_type) with the target outputs, ANOVA was initially tested to check for normality and homogeneity of variances (using the Shapiro and Levene tests). However, the assumptions for ANOVA were not met, leading to the use of the Kruskal-Wallis test as an alternative. Due to the large sample size (N > 5000), the p-values from both ANOVA and Kruskal-Wallis were unreliable, making conclusions difficult. To address this, a box plot represented in Figure 4 was used to visually assess the differences between groups. [5]

Furthermore, to adjust for the large sample size, the Kruskal-Wallis Effect Size (epsilon-squared) was calculated. The epsilon-squared value of 0.2487 indicates that 24.87% of the variance in expected_efficiency can be explained by the pv_module_type, suggesting a large effect of pv_module_type on the expected performance of the modules.

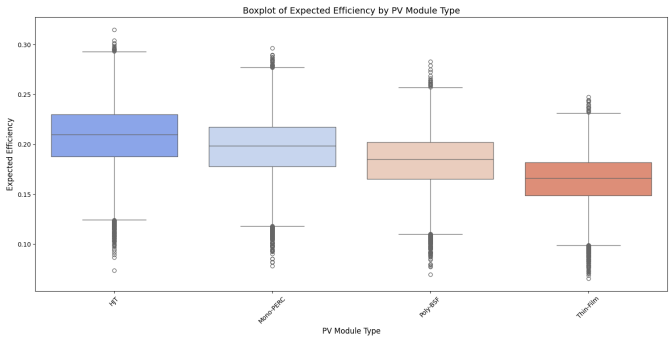

Figure 4: Box plot showing the distribution of expected efficiency for each photovoltaic module type.

As can be seen, the features that exhibit the strongest relationships with the target variables are those with the highest correlation values, such as Isc (short-circuit current), irradiance, affected area and pv_module_type (some tendency in the different groups). Conversely, features with correlation values close to 0 suggest that their variations are not significantly related to changes in the target variables, and therefore, they may not be as important for the prediction task.

D. Conventional Machine Learning Methods

1) Regressions: The first approach to a problem should usually be the simpler one. In machine learning, that would correspond to regressions. These models assume a certain relationship (e.g. linear, quadratic, etc.) between a set of input features and continuous output features. The aim of regressions is to estimate the parameters that govern this relationship, effectively capturing the underlying patterns in the data. Regression models are particularly well-suited for tasks where the output features are continuous, such as the Expected Efficiency.

However, they are versatile enough to handle discrete outputs, such as classifying Efficiency Levels. The predictions will return a continuous Efficiency Level that should then be discretized to the correct scale. It should be noted that, as this is not the main purpose of regression models, they are not expected to perform as well as they do for continuous outputs or as other models, suited for categorical outputs, would perform.

To begin the tests, we tried Linear Regression, which, as the name indicates, assumes a linear relationship between the input and output. We then implemented regularization techniques to prevent overfitting and improve model generalization, resorting to Ridge and Lasso regressions. Both these techinques penalyze models with large coefficients. The first one applies a penalty proportional to the sum of squared coefficients, while the second penalyzes the sum of their absolute value. These last two techniques require tunning of the regularization parameters [6]. For Ridge, it was used a regularization parameter that applies a moderate penalyzation to the coefficients, similar to what we've done in class. For Lasso regression, we performed a search for the optimal parameter resorting to cross validation and using R^2 to evaluate the models and corresponding parameters.

It was considered to be important also checking if the relationship between the data was non linear and how that impacted the results. So we also implemented Polynomial Regression, which operates very similarly to what was just described but now assumes that the relationship between the data is discribed by a polynom of greater degree [7]. We tried to also perform a search for the optimal degree of the polynom that would best fit the data. That showed to be very high computaional cost, taking too long to return a value, so we decided to try a few values for the degree manualy.

Afterwards, we wanted to test Logistic Regression, a type of regression specifically suited for classification problems, which was applied to predict the Efficiency Level. This model computes an indiviudal linear score for each class using the input features. These scores are passed to the softmax function, which converts them into probabilities for each class. The class that has the highest probability is chosen as the output. [8]

Finally, we also tested Random Forest Regression, a meta estimator that fits a number of tree decision regressors on various sub-samples of the dataser and uses averaging to improve the predictive accuracy and control over-fitting.[9]

2) Classifiers: In machine learning, classification models are algorithms that are used to predict categorical labels. The aim of these classifiers is to assign a given input to one of several predifined classes.

Firstly, we used Random Forest Classification, that works in the same way as its regression counterpart, but, instead of using tree decision regressors, uses tree decision classifiers.

Afterwards, we used Naive Bayes Classifier, a probabilistic classifier. It assumes that each feature contributes to the model independently of the others. [10]

E. Deep Learning Approaches

1) Autoencoders: As analyzed in Section II c), some features are less relevant for predicting the target variable. To address this, two different autoencoder[11] architectures,

illustrated in Figure 5 were trained to explore the possibility of reducing the dimensionality of the input data. The goal of this approach is to generate new, more compact features that can be used in subsequent models, thereby reducing computational complexity. Early stopping was employed during training, with the MSE Loss as the criterion. Only the architecture that resulted in the best data reconstruction will be used for further analysis.

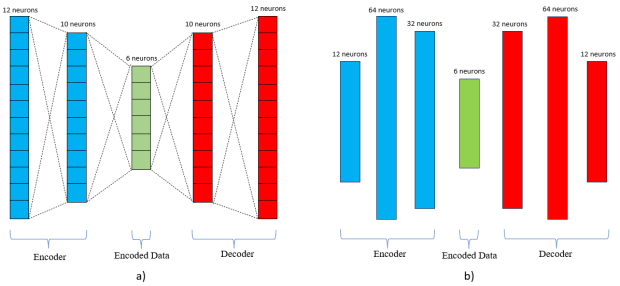

Figure 5: Autoencoders neural network structures.

2) MLP Neural Networks: It was also implemented two different neural network[12] architectures, represented in Figure 6, to solve the problem of classifying the efficiency_level and predicting the value of expected_efficiency. For the classification task, CrossEntropyLoss was used, and for the regression task, MSELoss was applied. In both neural networks, the early stopping strategy was implemented, and the learning rate was set to 0.01 and 0.003, respectively.

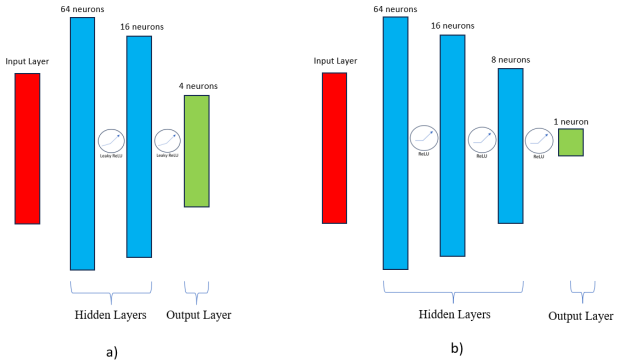

Figure 6: a) Classification Neural Network b) Regression Neural Network

F. Evaluation of the Models

The performance of the models will be evaluated using various metrics depending on the nature of the outputs, whether continuous or discrete. For the continuous output, which is expected efficiency, the main evaluation metrics include the R^2 score and Root Mean Squared Error. The R^2 is a statistical measure in a regression model that, by definition, determines the proportion of variance in the dependent variable that can be explained by the independent variable. While RMSE is a measure of the root-mean-square difference between the predicted and actual values. [13]

In the case of discrete output, several metrics of classification, such as accuracy, precision, recall, and the F1 score. Accuracy tells the proportion of correctly predicted instances from all instances. Precision measures the true positives across all the instances that are predicted as positive. On the other hand, recall means a proportion of true positives identified against all actual positives. The F1 score is the balanced mean of precision and recall. [14]

For the Neural Network, it was also calculated the AUC of the ROC and PR curve which shows the trade-off between true positives and false positives across different decision thresholds. An AUC score close to 1 indicates that a model is a good discriminator, while scores closer to the lower bound indicate poor performance. [15]

III. Results and Discussion

A. Models using all features

Initially, the models used the 10 features available with the goal of maximizing the learning capacity by leveraging all potential information. No feature selection or dimensionality reduction was done, so the models were trained and evaluated on the full dataset. This was a comparison baseline for further experiments on feature selection and dimensionality reduction techniques that will be aproached in the next section. The results obtained for each model previsously described are represented in Tables III and IV. Regarding the degree used in the polynomial regression, we opted to use a second degree polynom because the results were already great and it has a relatively low computational cost. Higher degree polynoms bring a heavy computational cost and the improvement in the results is not significative.

Table III: Model Performance Comparison in terms of Classification of the efficiency_level. Best results for each metric highlighted in bold

Model	Accuracy	Precision	Recall	F1 Score
Linear Regression	0.5777	0.8553	0.5777	0.6420
Ridge Regression	0.5779	0.8553	0.5779	0.6421
Lasso Regression	0.2607	0.0679	0.2607	0.1078
Polynomial Regression	0.6276	0.8614	0.6276	0.6871
Logistic Regression	0.9887	**0.9886**	**0.9887**	**0.9887**
Random Forest	0.9768	0.9769	0.9461	0.9523
Naive Bayes	0.7743	0.7501	0.7743	0.7588
Classification Neural Network	**0.9959**	0.9833	0.9840	0.9837

Table IV: Model Performance Comparison in terms of predicting the expected_efficiency. Best results for each metric highlighted in bold

Model	R^2	MSE
Linear Regression	0.9986	0.0012
Ridge Regression	0.9986	0.0012
Lasso Regression	0.9985	0.0013
Polynomial Regression	**0.9999**	**0.0001**
Random Forest	0.9998	0.0004
Regression Neural Network	0.9956	0.0022

As it is possible to visualize in Figure 1, the dataset is very imbalanced, with many more data points in the efficiency levels that are good, compared to those that are moderate. Therefore, the classification report represented in Table V corroborates this imbalance, since the neural network had

many more samples from the good class, which led to a better learning process, resulting in better precision values. Since the samples for the Moderate class were much fewer, the model had more difficulties in learning features that are relevant for classifying this class. In Figure 7, it is possible to see that all values of AUC (Area Under the Curve) for the ROC and PR curves were very high, with means that the neural network was able to perform the classification task very successfully.

Table V: Classification Report on predicting efficiency_level

Class	Precision	Recall	F1-Score	Support
Extremely Bad	0.99	1.00	0.99	6650
Bad	0.99	0.98	0.98	3885
Moderate	0.96	0.96	0.96	1639
Good	1.00	1.00	1.00	12826

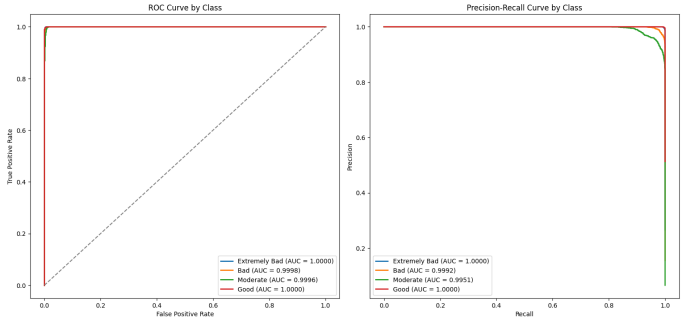

Figure 7: ROC and PR Curves.

As shown in Table IV, the models performed extremely well in predicting the expetcted efficiency of the photovoltaic modules. All the R^2 values are above 0.99 and RMSE values below 0.001. Regarding the regression first four regressions (linear, ridge, lasso and polynomial) they perform great but polynomial regression takes the edge and therefore will be the only one considered for further experiments on feature selection and dimensionality reduction techniques. Unexpectedly, the neural network performed the worst in this task, however this can be disregarded as it still recorded very good marks.

For the classification task the results were very good for some of the models are relatively average for others. As expected, the first four regressions performed the worst which is normal as they are primarily suited for regression tasks and were adapted to classify the efficiency level. The polynomial regression keeps maintains a better performance than the first three. The Naive Bayes classifier did not show very promising results (scoring around 0.75 on the evaluation metrics) due to the severe class imbalance in the dataset which affects its performance. Logistic regression, random forest and the neural network performed exceptionally well (scoring above 0.97 in most evaluation metrics) with the neural network taking a slight edge.

B. Feature Selection

As explained in the methodology, two different autoencoder (AE) architectures were trained, with the selection of the one that allowed the best reconstruction of the features using lower dimensionality. As can be seen in Figure 8, autoencoder 1 achieved a lower loss (0.0418) compared to autoencoder 2, which had a loss of 0.1961.

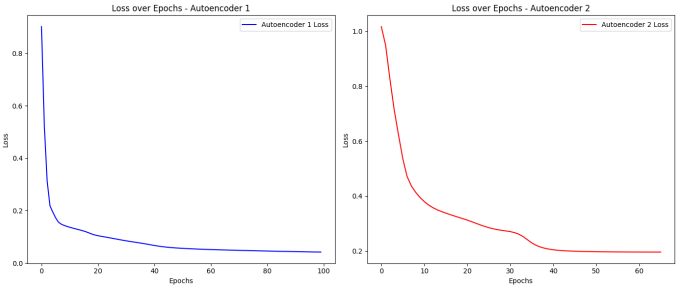

Figure 8: Loss Evolution of Autoencoder 1 and 2.

For the performance impact brought about by dimensionality reduction, an autoencoder was used to produce for input new data with only six features, referred to as Autoencoder 1. Dimensionality reduction plays an important role in model efficiency by reducing a number of input variables. Besides accelerating the process of model training, the computational cost of the whole process is reduced. Besides, with a basis on the correlation established in Section II c) between different features of the dataset, we took the four most highly correlated features representing CF in this context to the target variables. This will include Isc, which represents short-circuit current, irradiance, affected_area, and pv_module_type since they are the ones that determine the outcomes of the models.

We decided to keep the most important aspects of the data by choosing those features which are most highly correlated as a way of keeping the predictive power of the models. We have picked those models which have shown superior performance both for regression and classification tasks so far and test them with these reduced sets of features. This allowed us to investigate the trade-off between feature dimensionality reduction and model performance in both computational efficiency and accuracy.

Results obtained from testing both the autoencoder-generated features and the manually selected high-correlation features are detailed in Tables VI and VII. These tables show the performance of the models for these reduced feature sets, comparing their original performance using all the features against that obtained after reduction. The analysis will hence reveal not only how much efficiency has been gained through this dimensionality reduction but also the importance of choosing relevant features for good model performance.

Regarding the results in Table VI all the models' performance take a considerable hit with the autoencoder, which makes sense considering that some information was lost as only the most impactful features were used. Nonetheless, some models still performed well, namely the neural network and the random forest with thelast reaching values above 0.85 for accuracy, precision, recall and F-1 score. The polynomial regression and Naive Bayes keep performing the worst, following the same tendency as before.

When only four features were used surprisingly, the models' results got closer to the ones in Table III. This results are

Table VI: Model Performance Comparison in terms of Classification of the efficiency_level. AE means autoencoder and CF means features with the highest correlation to the target. Best results for each metric highlighted in bold

Model	Accuracy	Precision	Recall	F1 Score
Polynomial Regression AE	0.5516	0.8408	0.5516	0.6240
Polynomial Regression CF	0.6160	0.8564	0.6160	0.6752
Logistic Regression AE	0.7234	0.6424	0.7234	0.6676
Logistic Regression CF	0.9131	0.9069	0.9131	0.9091
Random Forest AE	0.8862	0.8598	0.8662	0.8504
Random Forest CF	0.9186	**0.9430**	**0.9444**	**0.9434**
Naive Bayes AE	0.5190	0.3850	0.5195	0.4327
Naive Bayes CF	0.7734	0.7475	0.7734	0.7572
Neural Network AE	0.9289	0.7269	0.6733	0.6638
Neural Network CF	**0.9582**	0.8254	0.8146	0.8193

great considering so few features were used and it reduces signifincatly the computational cost of the models' operations. This might be explained by the fact of the remove of unecessary noise of the data, that does not influence considerebly the output.

Table VII: Model Performance Comparison in terms of predicting the expected_efficiency.AE means autoencoder and CF means features with the highest correlation to the target. Best results for each metric highlighted in bold

Model	R^2	MSE
Polynomial Regression AE	0.9660	0.0060
Polynomial Regression CF	0.9912	0.0031
Random Forest AE	0.9708	0.0056
Random Forest CF	**0.9931**	**0.0027**
Neural Network AE	0.9246	0.0089
Neural Network CF	0.9861	0.0038

Taking a look at Table VII, the models performed very well for the regression task when less features were used, even though scoring slighlty lower than in Table IV, as expected. The autoencoder still shows to be slighlty less efficient.

It is also important to mention that the results using only 4 features are very similiar from those obtained when using the entire dataset which can be explained due to the fact that , by its very nature, Random Forest performs feature selection during the classification itself; it gives higher importance to the most relevant features when constructing decision trees. It will, therefore, tend to rely more on the features most responsible for the prediction. This mimics the effect of manually selecting important features based on their correlation whereby the less relevant features contribute very little toward the decisions made by the model. Hence, reducing the number of features to only the most important, we simply align the model's embedded feature selection process with those manually selected and get similar performance.

IV. Conclusion

By analysing the obtained results, we can conclude that most of the models were implemented with success, confirming the possibility of implementing Machine Learning algorithms in order to determine whether or not a certain photovoltaic panel will have a good efficiency level. Initially, when the models still used all available features, the best results were obtained with the Classification Neural Network and with the Regression Neural Network. The classification models achieved better accuracy and precision values for Good class, due to this having more values, allowing for a better learning process.

Afterwards, the results somewhat dropped when using the autoencoder, due to losing a bit of information, as only the most impactful features were used. Nevertheless, the neural network and the random forest were still able to perform well. Finally, when only four features were used, the results rose, as unnecessary noise was removed from the data. This is great considering the possible reduction in computational cost.

If we were to continue this project, several improvements could be made to enhance the performance of the models. First, refining the hyperparameters and the neural network's architecture would be essential. This could involve adjusting the number of layers, experimenting with different activation functions, and using techniques like dropout to prevent overfitting. We would also test additional models, such as K-Nearest Neighbors (KNN), Support Vector Machines (SVM), and Gradient Boosting Machines (GBM), to compare their performance against the existing models.

References

[1] Khaled Mahmud Sujon, Rohayanti Binti Hassan, Zeba Tusnia Towshi, Manal A. Othman, Md Abdus Samad, and Kwonhue Choi. When to use standardization and normalization: Empirical evidence from machine learning models and xai. *IEEE Access*, 12:135300–135314, 2024.

[2] Nishoak Kosaraju, Sainath Reddy Sankepally, and K. Mallikharjuna Rao. Categorical data: Need, encoding, selection of encoding method and its emergence in machine learning models—a practical review study on heart disease prediction dataset using pearson correlation. In *Proceedings of International Conference on Data Science and Applications*, pages 369–382, 2023.

[3] Ahmed Faris Amiri, Aissa Chouder, Houcine Oudira, Santiago Silvestre, and Sofiane Kichou. Improving photovoltaic power prediction: Insights through computational modeling and feature selection. *Energies*, 17(13), 2024.

[4] F Essam, Hashash El, and Shiekh Raga Hassan Ali. A comparison of the pearson, spearman rank and kendall tau correlation coefficients using quantitative variables. *Asian J. Probab. Stat*, pages 36–48, 2022.

[5] D. J. Sheskin. *Handbook of Parametric and Nonparametric Statistical Procedures*. Chapman & Hall, London, 2nd edition, 2000.

[6] Trevor Hastie Robert Tibshirani Gareth James, Daniela Witten. *An Introduction to Statistical Learning*. Springer, 2013.

[7] Eva Ostertagova. Modelling using polynomial regression. *Procedia Engineering*, 48:500–506, 12 2012.

[8] K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

[9] Daniel Borup, Bent Jesper Christensen, Nicolaj Søndergaard Mühlbach, and Mikkel Slot Nielsen. Targeting predictors in random forest regression. *International Journal of Forecasting*, 39(2):841–868, 2023.

[10] Or Peretz, Michal Koren, and Oded Koren. Naive bayes classifier – an ensemble procedure for recall and precision enrichment. *Engineering Applications of Artificial Intelligence*, 136:108972, 2024.

[11] Dor Bank, Noam Koenigstein, and Raja Giryes. *Autoencoders*, pages 353–374. Springer International Publishing, Cham, 2023.

[12] Hervé Abdi, Dominique Valentin, and Betty Edelman. *Neural networks*. Number 124. Sage, 1999.

[13] Domenico Chicco, Marius J. Warrens, and Giuseppe Jurman. The coefficient of determination r-squared is more informative than smape, mae, mape, mse and rmse in regression analysis evaluation. *PeerJ Computer Science*, 7(e623), Jul 2021.

[14] Hercules Dalianis. *Evaluation Metrics and Evaluation*, pages 45–53. 05 2018.

[15] J. Miao and W. Zhu. Precision–recall curve (prc) classification trees. *Evolutionary Intelligence*, 15:1545–1569, 2022.

V. CONTRIBUTIONS

Fábio Rodrigues (40%): enconding of the data, dataset analysis, feature reduction and development of the neural network model.

Sandro Oliveira (30%): development of the linear, ridge, lasso, polynomial and logistic regression models

Tiago Castanheira (30%): development of the random forest and generative classifier algorithms.

The report was collectively done by all the group members.