Cristóbal Sánchez Moreno - 100510664
Francisco Wagner Manetti - 100533319
Big Data Analytics - UC3M
Machine Learning - Project 1
2025/03/07

1. Dataset

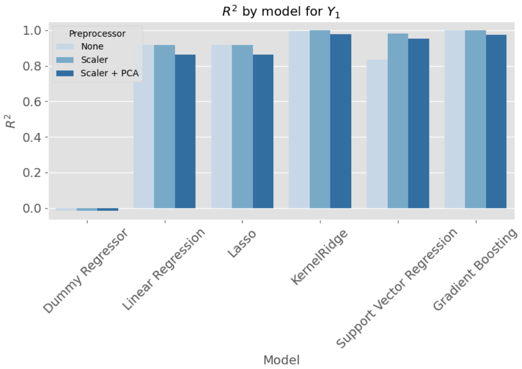
Energy Efficiency dataset published by UC Irvine. The dataset shows heating and cooling load requirements of buildings as a function of building parameters. The complete dataset and additional details can be found in the link below:
https://archive.ics.uci.edu/dataset/242/energy+efficiency

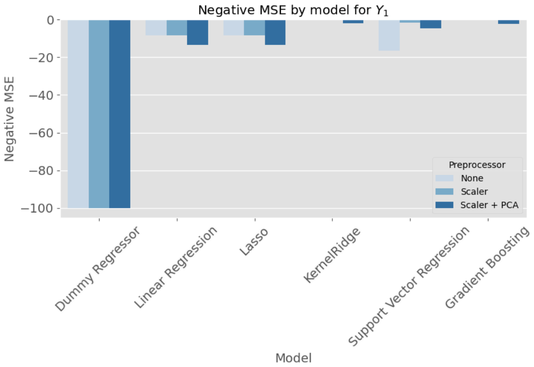
2. Goal

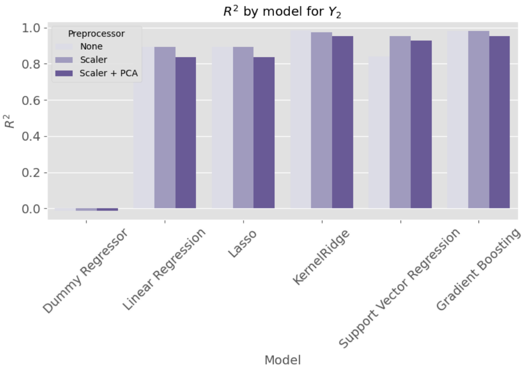
The goal of this project is to use machine learning models to build predictors of the target variables. In doing so, we must use at least 3 different scikit-learn classes of models, one of which must be a kernel method. Next, we must use at least one preprocessing method (e.g. scaling, feature selection, etc.). And lastly, we must use hyper-parameter selection in at least one stage of the data processing pipeline.

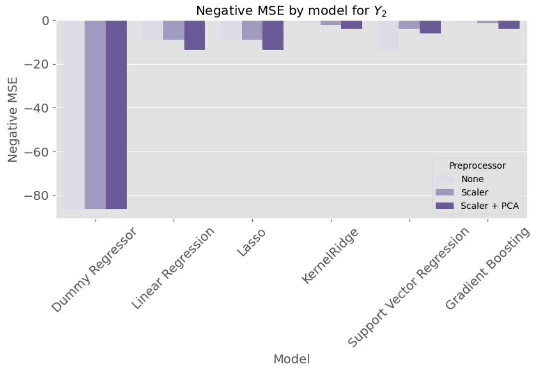
3. Code Overview

Firstly, we did exploratory data analysis and discovered that the dataset contains 8 attributes (denoted by X1 through X8) and two target variables (denoted by Y1 and Y2 and highly correlated). We split the data into train and test, using 75% for training and 25% for testing. Next, using the training data we analyzed model performance for different preprocessing, regressors, and hyper-parameters. We compared the models using the metrics R2 and Mean Squared Error and chose the best model for each target (Y1 and Y2). Lastly, these best models were used for estimation of future performance using the test data. The charts below show the performance for the different models we implemented using 5-fold cross validation on the training dataset.









4. Preprocessing:

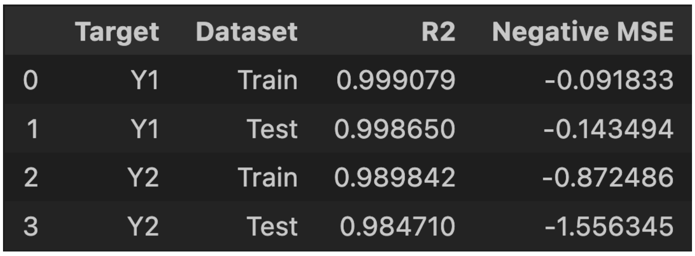
Exploratory data analysis confirmed that all variables are numeric and contain no missing values, eliminating the need for imputation. The dataset includes eight input variables with different scales, so we evaluated three preprocessing approaches: no preprocessing, standard scaling, and standard scaling with PCA. Standard scaling is beneficial as it normalizes the data by subtracting the mean and scaling to unit variance, making it suitable for methods such as linear regression and support vector machines (Support Vector Regressor in our case) among others. Additionally, PCA further reduces dimensionality by retaining the most significant variance, helping to prevent overfitting, eliminate collinearity, and reduce noise. However, our model results indicate that standard scaling alone does not significantly improve most models, except for support vector regression. Additionally, combining standard scaling with PCA reduces performance across all models compared to standard scaling alone. Therefore, standard scaling alone is the most effective preprocessing step overall.

5. Model Comparison:

We considered a series of models with different hyper-parameters combinations. In particular the models were: Dummy Regressor, Linear Regression, Lasso, Kernel Ridge, Support Vector Regression and Gradient Boosting Regressor. We found that Gradient Boosting with hyper-parameter optimization and standard scaling leads to the best scores in the prediction of both targets, closely followed by the Kernel Ridge regressor. The latter, despite obtaining a slightly worse performance as compared with Gradient Boosting, is approximately 6 times faster in terms of training time and hence less computationally costly. Still, as our dataset is small enough we will not consider this criterion for the selection of a best model, eventually choosing the Gradient Boosting Regressor.

6. Estimation of Future Performance:

Then, we estimated the performance of this best model, producing the table below.



	Target	Dataset	R2	Negative MSE
0	Y1	Train	0.999079	-0.091833
1	Y1	Test	0.998650	-0.143494
2	Y2	Train	0.989842	-0.872486
3	Y2	Test	0.984710	-1.556345

We can see that the Gradient Boosting model (along with scaling) shows high predictive performance, with R2 values near 1, indicating it effectively captures variance in both targets. However, Y1 is predicted more accurately than Y2, as seen in the lower negative MSE. The model exhibits slight overfitting, especially for Y2, where test errors increase more noticeably. Despite this, it generalizes well, with only a small performance drop from training to testing.