NGBoost: Various probability distributions

Ota Zuzana Vojtěch Jakub

Advisor: Terézia

December 27, 2025

1 Introduction

The primary objective of this project is to compare the application of NGBoost with normal, lognormal, and exponential probability distributions on data with various noise to other methods of continuous variable prediction problems. By analyzing how these distributions perform on various datasets, we aim to show the model's behavior and performance metrics.

The datasets used for this analysis are on concrete compressive strength, energy efficiency, naval propulsion plants maintenance, power generation efficiency, wine quality, and yacht hydrodynamics.

Our choice of topic was based on an interest in exploring new methods of machine learning that we did not know of. How different probability distributions influence predictive accuracy and model reliability across diverse datasets. This exploration is expected to shed light on the practical applications of NGBoost, contributing to a deeper understanding of its capabilities and limitations in real-world scenarios. Through this project, we would like to provide a comprehensive analysis that not only advances our knowledge but also serves as a valuable resource for others interested in the field of machine learning.

2 Natural Gradient Boosting (NGBoost)

NGBoost is a robust machine learning method suitable for both regression and classification tasks. It works by sequentially combining several weak learners, usually decision trees. Each new learner is trained to rectify the errors made by the ensemble of previous learners. What distinguishes NGBoost from other boosting methods is its ability to predict the entire probability distribution of the target variable, rather than just a point estimate. This is achieved by estimating the parameters of a specific probability distribution for the target variiable, providing uncertainty estimates crucial in fields like healthcare and meteorology.

The main innovation in NGBoost lies in its use of natural gradients. Unlike standard gradients, which consider only the immediate direction of the steepest descent in the parameter space, natural gradients account for the geometry of the probability space. This leads to more efficient parameter updates. NGBoost starts with an initial set of parameters for the probability distribution and iteratively refines them by computing natural gradients. These gradients are based on a scoring rule that measures discrepancies between predicted and actual outcomes. By using natural gradients to adjust the training process, NGBoost provides a reliable method for multiparameter boosting.

NGBoost is highly modular and customizable. Users can choose any base learner (typically decision trees), any family of distributions (Normal, Laplace, etc.), and any scoring rule (Maximum Likelihood Estimation, Continuous Ranked Probability Score). The model starts with an initial guess of the parameters and improves them iteratively. At each stage, the natural gradients guide the updates of the base learner outputs, which are then scaled and incorporated into the final parameter estimates. This step-by-step refinement builds a strong model that not only predicts but also provides a probabilistic understanding of future outcomes, making NGBoost a powerful tool for predictive analytics.

Empirical results show that NGBoost offers competitive performance in probabilistic prediction tasks, often surpassing traditional methods in estimating predictive uncertainty. In regression tasks, NGBoost provides full probability distributions rather than point predictions, allowing for meaningful interval predictions and risk assessments. The modularity and flexibility of NGBoost, combined with the stability of natural gradients, make it an excellent choice for predictive tasks that require robust uncertainty estimation. [5]

3 Datasets

In this section, we present an overview of the datasets utilized in our project, which aims to explore the impact of different probability distributions on model performance.

We're using seven datasets from the UCI Machine Learning Repository [7], which are commonly studied in research. We'll use them to compare how well our model works in different situations. Some of the datasets have more than one variable we want to predict, so we'll focus on predicting the first one.

3.1 Dataset Combined Cycle Power Plant

The Combined Cycle Power Plant dataset [8] contains 9568 data points collected over 6 years (2006-2011) from a power plant operating at full load. It includes hourly average ambient variables such as Temperature, Ambient Pressure, Relative Humidity, and Exhaust Vacuum, collected from sensors around the plant. The target variable, Net Hourly Electrical Energy Output (PE), ranges from 420.26 to 495.76 MW. The dataset aims to predict the net hourly electrical energy output based on the ambient variables recorded.

3.2 Dataset Concrete Compressive Strength

The Concrete Compressive Strength dataset [10] contains 1030 data points and focuses on predicting the compressive strength of concrete, a crucial property in civil engineering. The target variable, Concrete Compressive Strength, is measured in megapascals (MPa) and represents the ability of the concrete to withstand pressure before breaking.

3.3 Dataset Condition Based Maintenance of Naval Propulsion Plants

The Condition Based Maintenance of Naval Propulsion Plants dataset [1], donated on September 10, 2014, contains 11934 data points. It focuses on predicting the performance degradation of Gas Turbines (GT) in a naval vessel (Frigate) with a Combined Diesel Electric And Gas (CODLAG) propulsion plant type.

The target variables in this dataset include the GT measures at the steady state of the physical asset, such as the Gas Turbine shaft torque (GTT), GT rate of revolutions (GTn), Gas Generator rate of revolutions (GGn), and other related measures. Additionally, it includes coefficients representing the decay state of the GT compressor and turbine.

3.4 Dataset Energy Efficiency

The Energy Efficiency dataset [9] contains 768 samples and aims to predict two real-valued responses: Heating Load and Cooling Load. These responses represent the heating and cooling load requirements of buildings, respectively.

3.5 Dataset Wine Quality

The Wine Quality dataset [3] contains 4898 samples and aims to model the quality of red and white Vinho Verde wine from Portugal. The goal is to predict wine quality based on physicochemical tests, including features such as fixed acidity, volatile acidity, citric acid, residual sugar, etc.

The target variable, Quality, is a score ranging from 0 to 10, representing the sensory evaluation of the wine. Although it could be considered categorical, given the number of levels and to make the task more similar to the other datasets, we shall treat it as continuous.

3.6 Dataset Yacht Hydrodynamics

The Yacht Hydrodynamics dataset [6] is used to predict the hydrodynamic performance of sailing yachts based on their dimensions and velocity.

This dataset comprises 308 full-scale experiments conducted at the Delft Ship Hydromechanics Laboratory. The target variable in this dataset is the residuary resistance per unit weight of displacement, which is crucial for evaluating the ship's performance and estimating the required propulsive power.

3.7 Dataset Year Prediction MSD

The Year Prediction MSD dataset [2] aims to predict the release year of a song based on audio features.

This dataset contains 515,345 instances of mostly western commercial tracks from 1922 to 2011, peaking in the 2000s. The features are extracted from timbre features using The Echo Nest API.

The target variable is the song's release year, which ranges from 1922 to 2011. The dataset includes 90 attributes, 12 representing timbre averages and 78 representing timbre covariances.

4 Noisy Data

Attributes in all datasets are numerical, so we normalized them to values between 0 and 1. This normalization simplifies the process of adding noise to the datasets. Note that the target values are not normalized.

We introduced noise into the dataset by modifying each attribute with random values. For each feature, we randomly replaced 0%, 1%, 2%, and up to 10% of its values with random noise. The noise values were uniformly distributed within the interval [0, 1]. This approach allowed us to explore how different levels of noise affected the results of the learning algorithm. Specifically, we:

- Created a series of datasets with increasing amounts of random noise added to each column.
- For each percentage level (0% to 10%), replace the corresponding fraction of values in each attribute with random values.

5 Comparison

We used an already existing implementation of NGBoost [4]. We compared different distributions used in the NGBoost models - Normal, LogNormal and Exponential. The dataset was split 80/20 into train/test to compare the models. Model results were analyzed and compared using their predictions on the test set. To make the results as comparable as possible we did not finetune the models and instead used the default parameters given by their respective implementations.

5.1 Preprocessing

We inspected the datasets for missing values and features. We found no missing values or irregularities. Before training the models, the features were normalized.

5.2 Evaluation metrics

To compare the absolute performance of the models, we use the Mean Squared Error (MSE) since we consider the labels of all of the datasets to be continuous.

$$MSE = \sum_{i=1}^{D} (x_i - y_i)^2, \tag{1}$$

We trained two baselines to compare the NGBoost models to, a Linear Regression Model and a Gradient Boosting Regressor.

5.3 Results

There is often quite a little variation between the different distributions of the NGBoost algorithm. The only large difference is between the MSE scores on the yacht dataset (0.33 for normal, 0.29 for log-normal, 0.4 for exponential). Otherwise, the resulting scores are relatively similar across all of the datasets. Detailed table of these scores and the baseline score obtained through linear regression can be found below:

Table 1: MSE Scores for Different Distributions and Datasets

Dataset	**Normal**	**LogNormal**	**Exponential**	**Linear Regression**
CCPP	14.974	14.963	15.426	20.274
Concrete	35.246	34.813	32.291	95.975
CBMNP_1	3.650×10^{-5}	3.646×10^{-5}	3.186×10^{-5}	3.416×10^{-5}
Energy_1	0.304	0.290	0.309	9.152
Red_Wine	0.364	0.378	0.376	0.390
White_Wine	0.478	0.480	0.464	0.569
Yacht	0.330	0.296	0.400	67.603
Year Prediction	80.466	80.438	77.673	89.094

All of the variations were able to beat the linear regression baseline on most of the datasets (with the exception of the CBMNP_1 dataset), often by a significant amount. However, it is important to note that

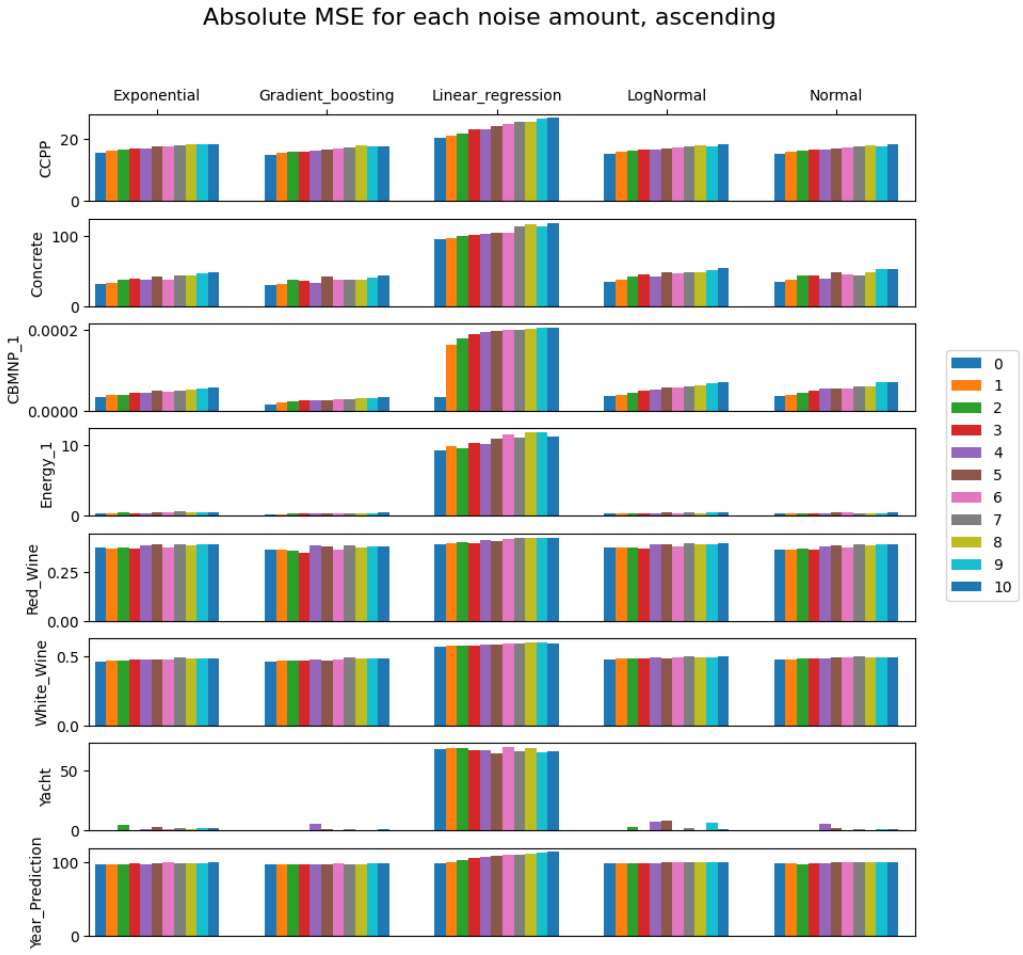
the significance is based on the predicted values. For example, for the Concrete dataset, the models beat the baseline MSE score by two thirds (approximately for the models 35 compared to 95 for the baseline), for the energy datasets, the relative difference is much greater (0.3 for the models compared to 9 for the baseline). In the case of the Yacht dataset, the MSE for the models is approximately 200 times smaller than for the baseline. For the rest of the datasets the MSE score is higher by at least 10 %, with the exception of the red wine datasets, where the difference is smaller, but the scores are still above the baseline.

The models weren't able to beat the Gradient Boosting Regressor convincingly on any of the datasets. The models with LogNormal and Normal distributions were able to beat this baseline on the Yacht dataset. Otherwise, the results are either comparable (CCPP, Red Wine, White Wine), or the baseline beats the NGBoost models by a noticeable amount, especially on the CBMNP dataset ($1.5e^{-5}$ to $3.5e^{-5}$).

5.4 Noisy Data[1]

We also explored the behavior of our models with the addition of some degree of noise to the datasets. As explained in the previous section, to achieve this, for each feature, a random portion of the data (0%, 1%, 2%, up to 10%), from the training datasets, was replaced with random noise.

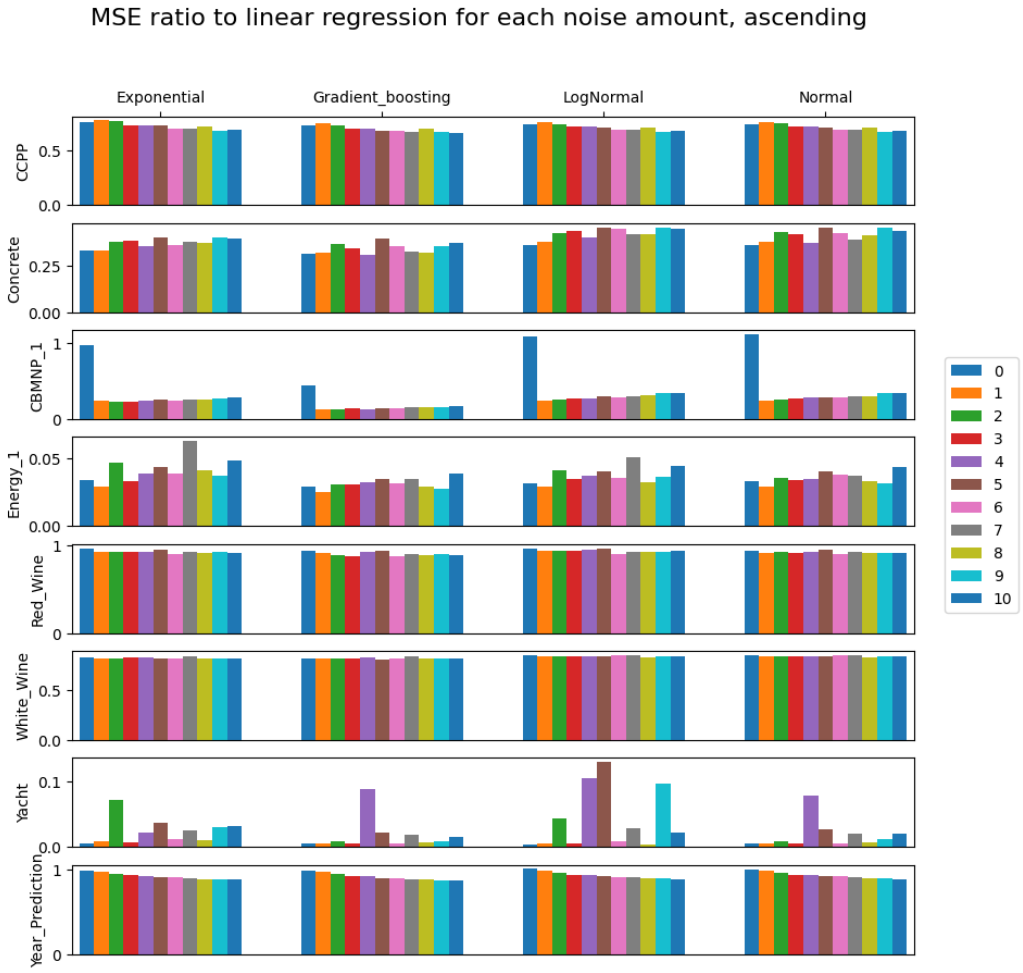
We decided to include a naïve Linear regression baseline and a Gradient boosting baseline, since that is the closest comparison to NGB, which is widely used in practice. To make the scores comparable across the datasets, we first took the ratio of the model scores to linear regression (M/LR) and visually evaluated that. We then also took the ratio of the NGB models MSE with different distributions and compared their MSEs to GB (NGB/GB).



In the figure above, we can see the absolute MSE for each percentage value of noise amount and for each dataset. It is apparent that the values are highest in the case of linear regression in all of the models. Linear regression seems to be quite sensitive to noise addition (although not always) and the other methods also seem to generally show higher MSE values with increasing noise, which is expected in our opinion. There is an interesting jump that can be observed in the case of the Condition Based Maintenance of Naval Propulsion Plants dataset between noise levels of 0% and 1%.

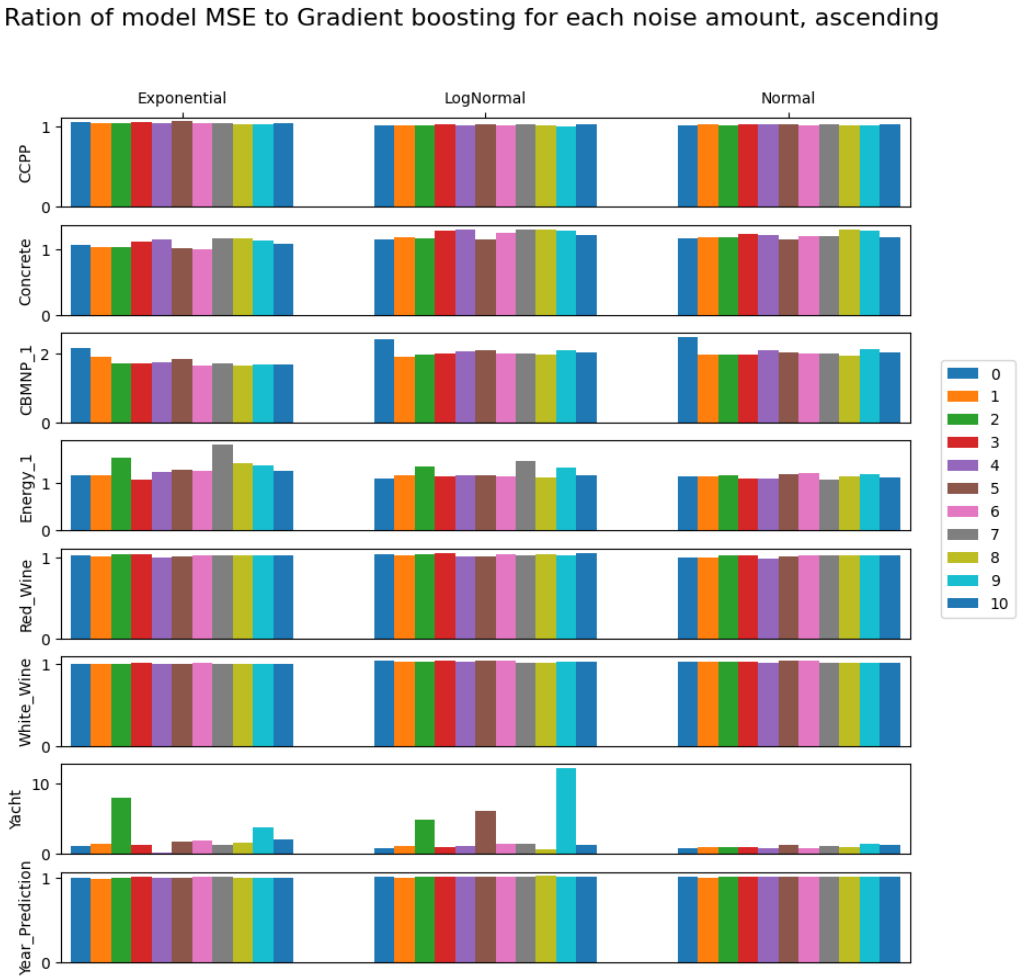
[1]MSE results: `https://drive.google.com/file/d/1-0l3LPd_PHDg4okhZAVXSsx7m87Q4L1V/view?usp=sharing`

We further compared and evaluated the ratio of the model scores to linear regression. The results can be seen in the following figure:



We can see that the way models respond to noise seems to be different than the way linear regression does. In terms of MSE, all of the models beat linear regression for all noise levels. It is unlikely that they learn anything meaningful on Wine and Year datasets, probably due to the sharp label distribution that the datasets have. In the Yacht dataset, all of the models show different results with respect to noise. This could be caused by different sensitivity to different samples and it would be interesting to investigate this phenomenon further.

Finally, we focused on the ratio of model MSE to Gradient boosting. The results that we obtained are summarized and can be seen in the figure below:



There seems to be no obvious difference in interaction with noise to gradient boosting and also between NGB distributions. NGB performs the same or worse than GB on all of the datasets and noise levels. For the CBMNP dataset, the performance is worse than for the other datasets, which might be caused by the uniform label distribution.

There seems to be no obviously best model. All of them have similar variances in the linear regression ratio. Gradient boosting has the overall best mean, the NGB distributions can be ranked as follow: Exponential, Normal, Lognormal. But the differences are not statistically significant as determined by a Kruskal Wallis test (H=4.02, p=0.4) on alpha = 0.05, thus we infer that the NGB distributions do not differ in a significant way. However, the Yacht dataset difference should be investigated further.

6 Conclusion

All models predictions on the test set were visually compared to the original data. The two baseline models, linear regression and gradient boosting regressor, were compared to NGBoost models with normal, log-normal, and exponential distributions using the MSE score, the Pearson correlation coefficient, the Wilcoxon signed-rank test, and cross-validation.

We found little difference between the distributions used in the model. The only apparent is in the Yacht dataset, where it would seem that the Exponential distribution would return the best results, given the highly skewed label distribution. This is, counterintuitively, not the case. However, the label distribution most likely causes the significant score difference between the Linear Regression baseline and all of the other models.

The fact that the NGBoost results are comparable to Gradient Boosting points to the quality of the algorithms. However, since we used the default model parameters, there might still be use-cases where one of the models is preferable. In conclusion, the models are able to beat naive linear regression baseline, but perform comparably or slightly worse to Gradient Boosting, when no parameter tuning is done.

7 Division of tasks

All tasks had to click together, so the classification of who did what was not binary. In **bold** are tasks that a person focuses on. Other items show tasks where a person also helped and did a significant part.

- Ota
  - **Baseline training**
  - NGBoost training
  - **Comparison**
  - Result interpretation
  - **Final report**
- Zuzana
  - **Datasets exploration**
  - **Datasets preprocessing**
  - NGBoost training
  - Result interpretation
  - **Final report**
- Vojta
  - Baseline training
  - **NGBoost training**
  - **Comparison**
  - **Result interpretation**
  - Final report
- Jakub
  - Baseline training
  - **NGBoost training**
  - Comparison
  - **Result interpretation**
  - **Final report**

References

[1] Coraddu Andrea, Oneto Luca, Ghio Alessandro, Savio Stefano, Anguita Davide, and Figari Massimo. Condition Based Maintenance of Naval Propulsion Plants. UCI Machine Learning Repository, 2014. DOI: https://doi.org/10.24432/C5K31K.

[2] T. Bertin-Mahieux. Year Prediction MSD. UCI Machine Learning Repository, 2011. DOI: https://doi.org/10.24432/C50K61.

[3] Cortez, Paulo, Cerdeira, A. Almeida, F. Matos, T., and Reis J. Wine Quality. UCI Machine Learning Repository, 2009. DOI: https://doi.org/10.24432/C56S3T.

[4] Tony Duan, Anand Avati, Daisy Yi Ding, Sanjay Basu, Andrew Y. Ng, and Alejandro Schuler. Ngboost: Natural gradient boosting for probabilistic prediction. *CoRR*, abs/1910.03225, 2019.

[5] Tony Duan, Anand Avati, Daisy Yi Ding, Khanh K. Thai, Sanjay Basu, Andrew Ng, and Alejandro Schuler. NGBoost: Natural Gradient Boosting for Probabilistic Prediction. Proceedings of the 37th International Conference on Machine Learning, 2020. URL: https://proceedings.mlr.press/v108/duan20a.html.

[6] Gerritsma J., Onnink R., and Versluis A. Yacht Hydrodynamics. UCI Machine Learning Repository, 2013. DOI: https://doi.org/10.24432/C5XG7R.

[7] Markelle Kelly, Rachel Longjohn, and Kolby Nottingham. The uci machine learning repository. `https://archive.ics.uci.edu`.

[8] Pnar Tfekci and Heysem Kaya. Combined Cycle Power Plant. UCI Machine Learning Repository, 2014. DOI: https://doi.org/10.24432/C5002N.

[9] Athanasios Tsanas and Angeliki Xifara. Energy Efficiency. UCI Machine Learning Repository, 2012. DOI: https://doi.org/10.24432/C51307.

[10] I-Cheng Yeh. Concrete Compressive Strength. UCI Machine Learning Repository, 2007. DOI: https://doi.org/10.24432/C5PK67.