Smart Data Project

Benchmark Algorithm of Ensemblist Methods on Spatial Data.

GitHub Repository

Students:

Andrea Bianco
Matteo Mancini
Rodrigue Mellot

Under the direction of :

Olivier Meslin

Promotion 2026

1 Abstract

Spatial interpolation is a fundamental challenge in data analysis, necessitating robust methods for estimating unknown values from scattered observations. This study provides a comprehensive benchmark to identify the most effective algorithms currently available. We begin with a literature review to classify existing techniques and analyze their theoretical advantages. To expand the scope of possible solutions, we also introduce and test ensemble learning models integrated with coordinate rotation alongside traditional methods. Our experimental framework employs a hierarchical evaluation strategy: initial screening on small datasets is used to discriminate and filter underperforming algorithms, followed by testing of the remaining candidates on complex, high dimensional datasets. We compare these methods based on scalability, computational efficiency, and robustness. Ultimately, this study offers a definitive guide to the "state-of-the-art" in spatial interpolation, highlighting which algorithms perform best across varying levels of difficulty and data density.

2 Introduction

Spatial interpolation is a fundamental technique within the environmental sciences, used extensively to estimate the values of continuous variables at unsampled locations based exclusively on spatial coordinates. For decades, deterministic and geostatistical methods have been regarded as the premier choice for these tasks due to their mathematical foundations in spatial continuity. However, the recent advancement of ensemble learning methods (such as Random Forest and Gradient Boosting) has begun to shift this perspective. These modern algorithms offer the potential to model complex, non linear spatial surfaces that traditional methods may struggle to capture accurately.

The central research question of this benchmark involves identifying which algorithm is most effective for predicting spatial values using coordinates as the only covariates. In defining the most effective algorithm, this study looks beyond mere accuracy. We define the optimal model through the lens of both predictive precision and scalability. Precision measures the ability of a model to minimize error across diverse spatial geometries, while scalability evaluates the computational efficiency of the algorithm as it transitions from sparse datasets to large scale high density data.

Much of this research is motivated by the proposition that coordinate rotation can be integrated into ensemble methods to effectively replace or outperform traditional deterministic algorithms. While standard tree based models often struggle with axis aligned splits in spatial contexts, the coordinate rotation paradigm suggests that transforming the feature space can unlock significantly higher performance. One of the primary goals of this benchmark is to rigorously test this hypothesis and develop a clear understanding of the true capabilities and limitations of coordinate rotation in spatial modeling.

To achieve a robust evaluation, this study examines eight datasets comprising both real world topographic data and synthetic stationary random fields. We compare a dozen distinct algorithms, ranging from k-nearest neighbors and kriging to advanced tree based variants and generalized additive models. By analyzing these results across varied conditions, such as grid versus non grid structures and small versus large sample sizes, this report seeks to provide a definitive comparison that guides practitioners in choosing the most efficient method for their specific spatial interpolation needs.

3 Methodology

Overview of the benchmarking framework and experimental pipeline.

3.1 Algorithms Considered

A) Generalized Additive Models - GAM	F) Nearest Neighbor Interpolation
B) GeoSpatial Random Forest	G) Oblique Random Forest
C) Gradient Boosting	H) Ordinary Kriging
D) Inverse Distance Weighting - IDW	I) Random Forest
E) MI-GBT	J) XGBoost

3.2 Comparison of Spatial Interpolation Methods

Our benchmark compares a large range of spatial interpolation algorithms that can be grouped into three distinct families based on their theoretical foundations: deterministic methods, geostatistical methods, and machine

learning approaches. Understanding these distinctions is essential for interpreting our results and providing practical guidance for model selection.

3.2.1 Deterministic Methods

Inverse Distance Weighting (IDW) and **K-Nearest Neighbors** (K-NN) belong to the ***deterministic family*** of spatial interpolation methods. Given identical input data, they always produce the same output, as their formulation contains no stochastic component. Predictions are obtained through fixed mathematical rules that operate directly on spatial distances, rather than through an explicit data driven learning mechanism.

However, it is important to remember that in this context *deterministic* does **not** mean *assumption free*. In practice, both IDW and KNN embed **implicit assumptions** about the underlying spatial process through the choice of **hyperparameters**, which determine the effective spatial scale of smoothing and the degree of locality in the interpolation.

For **Inverse Distance Weighting**, the prediction at a target location s_0 is computed as a distance weighted average of observed values:

$$\hat{z}(s_0) = \frac{\sum_{i=1}^{n} w_i(s_0)\, z(s_i)}{\sum_{i=1}^{n} w_i(s_0)}, \qquad w_i(s_0) = \frac{1}{d(s_0, s_i)^p},$$

where $d(s_0, s_i)$ denotes the Euclidean distance and $p > 0$ is the **distance-decay exponent**. Larger values of p impose a stronger locality assumption, causing nearby observations to dominate the prediction, while smaller values yield smoother surfaces by allowing distant points to contribute more substantially. Consequently, the choice of p effectively encodes an assumption about how rapidly spatial influence should decay with distance.

For **K-Nearest Neighbors**, predictions are obtained by averaging the values of the k closest observations:

$$\hat{z}(s_0) = \frac{1}{k} \sum_{i \in \mathcal{N}_k(s_0)} z(s_i),$$

where $\mathcal{N}_k(s_0)$ denotes the set of the k nearest neighbors of s_0 (optionally using distance based weights). Here, k acts as a **bandwidth parameter**: small values of k assume highly local spatial variation and may lead to noisy predictions, whereas larger values enforce stronger smoothing and implicitly assume a more slowly varying spatial surface.

Both methods are thus based purely on geometric principles and do not rely on any underlying statistical model. This simplicity offers clear practical advantages: they are easy to implement, as they only require basic distance computations, and they are computationally efficient because no training phase is needed. Their deterministic nature also makes them highly interpretable, since it is straightforward to understand how each prediction is obtained.

At the same time, this simplicity comes with limitations. These methods do not explicitly capture complex spatial structures that may be present in the data and rely on rigid monotonic distance assumptions. For instance, IDW presumes that the influence of observations decreases smoothly with distance, an assumption that may not hold in all spatial contexts. Despite these drawbacks, their speed and robustness make them valuable baseline approaches in spatial interpolation benchmarks.

For this reason, deterministic methods provide a useful baseline for spatial interpolation, against which more complex probabilistic and learning based approaches can be meaningfully compared.

3.2.1.1 Geostatistical Method

Kriging represents the geostatistical approach to spatial interpolation. Unlike deterministic methods, it treats spatial data as realizations of an underlying stochastic process. This probabilistic framework allows the method to explicitly model spatial autocorrelation, determine prediction weights by minimizing the variance of the prediction error, and provide uncertainty estimates associated with the predictions.

At the core of the approach lies the variogram, which describes how similarity between observations changes with distance. Rather than simply assuming that nearby points are more influential, as deterministic methods do, Kriging asks how strong this influence is and how it evolves across spatial scales. Based on this information, optimal prediction weights are obtained by solving a system of linear equations that accounts for both inter point distances and the overall spatial configuration.

There are several types of Kriging methods based on different assumptions about the spatial trend: **Universal Kriging** (polynomial drift with unknown coefficients), **Simple Kriging** (known constant mean), and **Ordinary Kriging**. In this benchmark, we focus on Ordinary Kriging.

Formally, the Ordinary Kriging predictor at an unobserved location s_0 is defined as a linear combination of the observed values, $\hat{Z}(s_0) = \sum_{i=1}^{n} \lambda_i Z(s_i)$, subject to the unbiasedness constraint $\sum_{i=1}^{n} \lambda_i = 1$. The weights λ_i are chosen so as to minimize the prediction error variance $\mathrm{Var}(\hat{Z}(s_0) - Z(s_0))$ under the assumed variogram model.

These theoretical foundations give Kriging several important advantages. Under its assumptions, Ordinary Kriging is the **Best Linear Unbiased Predictor (BLUP)**, meaning that among all predictors that are linear combinations of the data and unbiased, it minimizes the variance of the prediction error. It also guarantees exact interpolation at observed locations and produces smooth, continuous surfaces, avoiding the abrupt artifacts that may arise with simpler approaches. In addition, it naturally provides measures of predictive uncertainty, enabling the construction of confidence intervals.

However, these theoretical advantages come with substantial practical challenges. Kriging has **cubic computational complexity**, as it requires the inversion of an $n \times n$ covariance matrix, where n is the number of observations. As a result, it becomes **prohibitively slow for datasets larger than roughly 10,000 points**, limiting its applicability in large scale spatial problems.

In addition, the method relies on strong statistical assumptions. It **assumes** intrinsic **stationarity**, implying a constant mean over the study area and a variogram that depends only on the distance between points, not on their absolute location, **a condition that is often violated in real-world scenarios**. For example, in topographic elevation data, the mean altitude may vary systematically between valleys and mountain ranges; in urban environments, land values or pollution levels may differ markedly between city centers and suburban areas; and in climate related applications, temperature or precipitation fields often exhibit large scale spatial trends driven by latitude, altitude, or proximity to the sea. In such cases, assuming a constant mean across the domain can lead to biased predictions and distorted uncertainty estimates.

Kriging also assumes **isotropy**, meaning that spatial correlation depends solely on distance and not on direction. This assumption may be **unrealistic in the presence of directional effects**, such as prevailing wind patterns influencing air pollution dispersion, river networks imposing anisotropic dependence in hydrological variables, or geological strata generating directional continuity in subsurface properties. When such anisotropy is present but ignored, the variogram model becomes misspecified, potentially leading to degraded predictive performance.

Beyond these computational and theoretical issues, Kriging also presents practical implementation difficulties. Prior to prediction, the variogram must be estimated from the data. This step requires selecting a theoretical variogram model—such as spherical, exponential, or Gaussian—and estimating its key parameters, including the nugget, sill, and range. This procedure is often partly subjective, demands experience and judgment, and can be sensitive to outliers. Since prediction quality depends directly on the variogram specification, a poorly fitted model can result in unreliable estimates. Moreover, as a linear method, Kriging has limited ability to represent highly nonlinear spatial patterns and may overfit when applied to small datasets. It also implicitly assumes Gaussianity of the data, making transformations necessary when variables are strongly skewed.

Taken together, these limitations imply that although Kriging is theoretically optimal under its assumptions, it is not always the most practical choice in real world settings. The gap between theoretical optimality and empirical performance is therefore a central theme in the benchmark results presented in this study.

These limitations motivate the exploration of alternative approaches that relax such assumptions while retaining strong predictive performance.

3.2.1.2 Machine Learning Methods

The machine learning (ML) approaches considered in this benchmark represent a fundamentally different paradigm from both deterministic and geostatistical methods. Rather than relying on fixed mathematical rules or explicit stochastic models of spatial dependence, ML algorithms are **data-driven** and learn predictive relationships directly from the data by optimizing a loss function. This property makes them highly flexible, scalable, and particularly effective in high dimensional settings.

Within the ML family, we distinguish **two main classes of methods**: **tree-based ensemble methods** and **Generalized Additive Models (GAM)**. This distinction is important, as it reflects different modeling philosophies and leads to markedly different behaviors when applied to spatial data.

3.2.1.2.1 Tree-based Ensemble Methods

Tree based ensemble methods are among the most widely used machine learning algorithms for tabular data and are known for their **state-of-the-art predictive performance**, **robustness, and scalability**. In this benchmark, we focus on two major subfamilies: **Random Forest (RF)** and **Gradient Boosting (GB)** methods.

Random Forest is based on a **bagging strategy**, where multiple decision trees are trained on **bootstrap samples** of the data and their predictions are averaged. Each tree partitions the feature space through **recursive binary splits** designed to **minimize prediction error within each region**. The aggregation of many decorrelated trees results in stable predictions with reduced variance.

Gradient Boosting methods, such as **XGBoost** and **MI-GBT**, adopt a different strategy. Trees are built **sequentially**, with each new tree trained to correct the residual errors of the existing ensemble. This iterative refinement **often** yields **higher predictive accuracy** than bagging based methods, at the cost of increased **sensitivity to hyperparameter tuning** and a **higher risk of overfitting** if not properly regularized.

A key advantage of tree based methods lies in their broad **applicability across a wide range of prediction tasks**, as they can be employed **without relying on strong assumptions** about the underlying data generating process. Moreover, they are highly extensible, allowing additional covariates to be seamlessly incorporated into the model. For instance, in applications such as housing price prediction, spatial coordinates can be combined with socioeconomic or structural variables, a level of flexibility that is difficult to achieve with traditional geostatistical approaches.

3.2.1.3 Generalized Additive Models

Generalized Additive Models represent a distinct class within the ML family. GAMs model the target variable as a **sum of smooth nonlinear functions of the input features**, balancing flexibility and interpretability. While not tree based, GAMs can capture nonlinear spatial trends through smooth functions of the coordinates, positioning them at the boundary between classical statistical modeling and modern machine learning. Their additive structure makes them more interpretable than ensemble methods, but also limits their ability to represent highly complex spatial interactions.

3.2.1.4 Adapting Tree-based Methods to Geospatial Data

Despite their strong performance in many domains, tree based methods are not specifically designed for geospatial data. When applied naively to spatial coordinates, they face several well known challenges inherent to spatial processes, including spatial autocorrelation, anisotropy, and the presence of large scale spatial trends.

Spatial autocorrelation refers to the tendency of nearby observations to exhibit similar values. For example, environmental variables such as air pollution or temperature often display strong local correlation, meaning that observations are highly dependent on their neighbors. Standard tree based models do not explicitly account for this dependence structure and may therefore overfit local noise. Similarly, many spatial phenomena display anisotropy, where dependence varies with direction rather than distance alone, as in the case of pollutant dispersion driven by prevailing winds or hydrological variables constrained by river networks. Axis aligned decision trees struggle to capture such directional patterns efficiently, often requiring deep trees and many splits to approximate oblique spatial gradients.

Large scale spatial trends present a particularly important challenge for tree based methods. These trends represent smooth, gradual variations that span the entire spatial domain, such as elevation decreasing from mountainous regions to lowlands, temperature declining with latitude or altitude, or housing prices increasing radially from urban centers. Tree based models approximate such trends using axis aligned rectangular partitions, which is inherently inefficient. Capturing a smooth diagonal or curved trend requires many stepwise splits, leading to deep trees, high model complexity, and potential overfitting to local variations while failing to capture the global pattern effectively.

As a consequence, achieving optimal performance with tree based methods in geospatial settings typically requires adaptation. Broadly speaking, there are two alternative strategies to address these challenges. The first strategy consists in modifying the algorithms themselves to make them more suitable for spatial data. Examples include Geographic Random Forest, which explicitly incorporates spatial information into the ensemble construction, and Oblique Random Forest, which allows decision boundaries that are not constrained to be parallel to the coordinate axes. By enabling oblique splits, these methods can more naturally represent spatial gradients that are misaligned with the original coordinate system.

The second strategy focuses on preprocessing the spatial data in a way that makes it better suited to standard tree-based algorithms, without altering their internal functioning. A prominent example of this approach is coordinate rotation, which augments the feature space with rotated versions of the original spatial coordinates.

This transformation allows axis-aligned trees to effectively learn oblique spatial patterns, mitigating the geometric limitations of standard decision trees while preserving their computational efficiency and scalability.

In this benchmark, we investigate both adaptation strategies, with particular emphasis on coordinate rotation as a simple yet powerful preprocessing technique.

3.3 Coordinate Rotation

3.3.1 Addressing Anisotropy: Oblique Trees versus Coordinate Rotation

Regarding spatial interpolation, all standard tree based ensemble methods share a fundamental limitation: they rely exclusively on axis aligned splits during tree construction. At each node, the algorithm selects a single feature and a threshold value to partition the data along that coordinate axis. This implies that when dealing with spatial data, tree based models produce geographical splits with either a North-South (vertical) orientation or a (horizontal) East-West orientation. Obviously, spatial phenomena may have any spatial orientation, implying that off the shelf tree based models must approximate diagonal spatial patterns by an accumulation of horizontal or vertical splits, leading to a staircase approximation.

To overcome these limitations, two main strategies can be considered. The first one is represented by Oblique Decision Trees, an algorithms that directly learn oblique splits by constructing linear combinations of features at each node; however, these methods are computationally demanding. The second strategy is our Coordinate Rotation, a simpler approach that augments the feature space with rotated versions of the original coordinates, allowing standard axis aligned algorithms to effectively learn oblique boundaries.

The Coordinate rotation method is based on the transformation of the original spatial coordinates (x, y) by applying a series of rotation transformation around the centroid of the training data. Using this procedure we are able to obtain an icreased number of feature space containing multiple and different representations of the same space locations, unique for their rotation angle. The precise procedure is detailed in the following pseudocode.

Algorithm 1 Spatial Feature Augmentation by Coordinate Rotation

Require: Training dataset $\{(x_i, y_i)\}_{i=1}^{n}$, number of axes k
Ensure: Augmented feature matrix with original and rotated coordinates
Compute centroid:
$\bar{x} \leftarrow \frac{1}{n}\sum_{i=1}^{n} x_i \quad \bar{y} \leftarrow \frac{1}{n}\sum_{i=1}^{n} y_i$
Generate rotation angles and apply rotation:
for $j = 0$ to $k-1$ **do**
  $\theta_j \leftarrow \frac{360^\circ \cdot j}{k}$
  for $i = 1$ to n **do**
    $x'_{ij} \leftarrow \bar{x} + (x_i - \bar{x})\cos(\theta_j) - (y_i - \bar{y})\sin(\theta_j)$
    $y'_{ij} \leftarrow \bar{y} + (x_i - \bar{x})\sin(\theta_j) + (y_i - \bar{y})\cos(\theta_j)$
  end for
end for
Return augmented features: $\{(x'_{i0}, y'_{i0}, x'_{i1}, y'_{i1}, \ldots, x'_{i,k-1}, y'_{i,k-1})\}_{i=1}^{n}$

3.3.2 Model Selection for Coordinate Rotation

Not all algorithms benefit equally from coordinate rotation. For this reason, we apply the technique selectively based on each algorithmic properties:

Models with Coordinate Rotation Variants: - Random Forest (RF vs RF-CR) - XGBoost (XGB vs XGB-CR) - MI-GBT (MI-GBT vs MI-GBT-CR) - GAM (GAM vs GAM-CR)

The remaining algorithm are used only for a comparative purpose. This allow us to see how the coordinate rotations based models perform with respect to other existing strategy and isolate the specific impact of coordinate rotation on standard axis aligned tree algorithms.

3.3.3 Geometric Interpretation

From a geometric perspective, coordinate rotation can be interpreted as a change of reference system applied to the spatial domain. Each rotation defines a new coordinate system whose axes are oriented at a specific angle with respect to the original one. Within these rotated systems, spatial patterns that appear oblique in the original coordinates may become aligned with the axes, allowing tree based models to represent them through simple axis aligned splits.

4 Experimental Setup and Evaluation Metrics

4.1 Datasets

There will be plenty datasets to cover combinations of, **Real or Synthetic**, **Large or Small** and **Grid or No Grid**. It will also be about, very large, extremly large datasets and noisy datasets.

For the **synthetic ones**, they are built following the idea that we need some data spatially correlated, be able to respect our different criteria:

- **Spatial Correlation**: We use a **Matérn covariance model** (dimension=2, variance=1, length scale=10) to generate a Stationary Random Field (SRF). This ensures the synthetic data mimics the spatial continuity and "smoothness" often found in real world environmental phenomena.
- **Structure**: If there is a grid, points are generated on a regular Cartesian grid using a meshgrid of and coordinates. If it's not the case, points are sampled using a*Uniform Random Distribution across the spatial domain to simulate irregular sampling.
- **Size**: Ranging from 5,000 points for "Small" Datasets, 100,000 points for "Large" datasets, 1,000,000 points for "Very Large" datasets and 10,000,000 points for "Extrem Large"
- **Consistency**: A fixed seed (20170519) is applied to both the random field generation and the coordinate sampling to ensure the experiments are fully reproducible across different benchmark runs.

Noisy: For the noisy datasets, they are build using gaussian noise, with a variance equal to 0,5 for the low noise, 1 for medium noise and two for high noise.

For the **real datasets**, we utilize high quality topographic data provided by the **IGN (Institut National de l'Information Géographique et Forestière)**, the French national mapping agency.

- **BD ALTI**: This dataset represents the "unstructured" real world scenario. The points are derived from various sources (photogrammetry, digitization, etc.) where the spatial distribution of samples is irregular. So we can use this dataset as our no grid, large, real dataset.
- **RGE ALTI**: It is the highest resolution elevation model available nationally. It is provided as a 5 meter regular grid. The full national dataset contains over 22 billion points. So we can use this dataset as our grid, large, real dataset.
- **California Housing**: A standard machine learning benchmark derived from the 1990 U.S. Census. We include this dataset to test model robustness in a "medium sized, noisy, real world" scenario, contrasting with the relatively smooth topographic surfaces of the IGN data.

For the Small real world datasets, we use a subset of the French territory by filtering for Department 48 (Lozère). This department was chosen because its diverse topography—ranging from deep canyons and plateaus to mountainous terrain—offers a representative sample of various geographic challenges for spatial interpolation.

4.2 Dataset Reference Table

Dataset Name	Origin	Size Category	Structure	Noise	Row Count
bdalti	Real	Large	No Grid	Smooth	~7,000,000
bdalti_48	Real	Small	No Grid	Smooth	~40,000
rgealti	Real	Large	Grid	Smooth	~22,000,000,000
rgealti_48	Real	Small	Grid	Smooth	~1,500,000
S-G-Sm	Synthetic	Small	Grid	Smooth	5,000
S-G-Lg	Synthetic	Large	Grid	Smooth	100,000
S-NG-Sm	Synthetic	Small	No Grid	Smooth	5,000
S-NG-Lg	Synthetic	Large	No Grid	Smooth	100,000
S-NG-VLg	Synthetic	Very Large	No Grid	Smooth	1,000,000
S-NG-ELg	Synthetic	Extremely Large	No Grid	Smooth	10,000,000
S-NG-Lg-N1	Synthetic	Large	No Grid	Low Noise	100,000
S-NG-Lg-N2	Synthetic	Large	No Grid	Med Noise	100,000
S-NG-Lg-N3	Synthetic	Large	No Grid	High Noise	100,000
cal_housing	Real	Medium	No Grid	Noisy	~20,000

4.3 Performance Metrics

We evaluate all models using three complementary metrics computed on the held-out test set:

- **R^2 (Coefficient of Determination)**: Measures the proportion of variance in the target variable explained by the model. This metric is scale-invariant and provides an intuitive measure of overall model fit.
- **RMSE (Root Mean Squared Error)**: Quantifies prediction error in the original units of the target variable. Penalizes large errors more heavily than small ones due to the squaring operation. Lower values indicate better performance.
- **MAE (Mean Absolute Error)**: Measures average absolute prediction error in original units. More robust to outliers than RMSE. Lower values indicate better performance.
- **Training Time**: Wall-clock time (in seconds) required for model fitting on the training set. Provides insight into computational efficiency and practical scalability.

4.4 Benchmark Procedure

All experiments ran in a Python 3.13 environment on SSPCloud with full parallelization (n_jobs=-1). To ensure reproducibility, a fixed random seed of 42 was applied across all generators. Data preprocessing included log transforming skewed elevation targets and applying a 23 angle Coordinate Rotation (CR) for tree based models to mitigate orthogonal split biases.

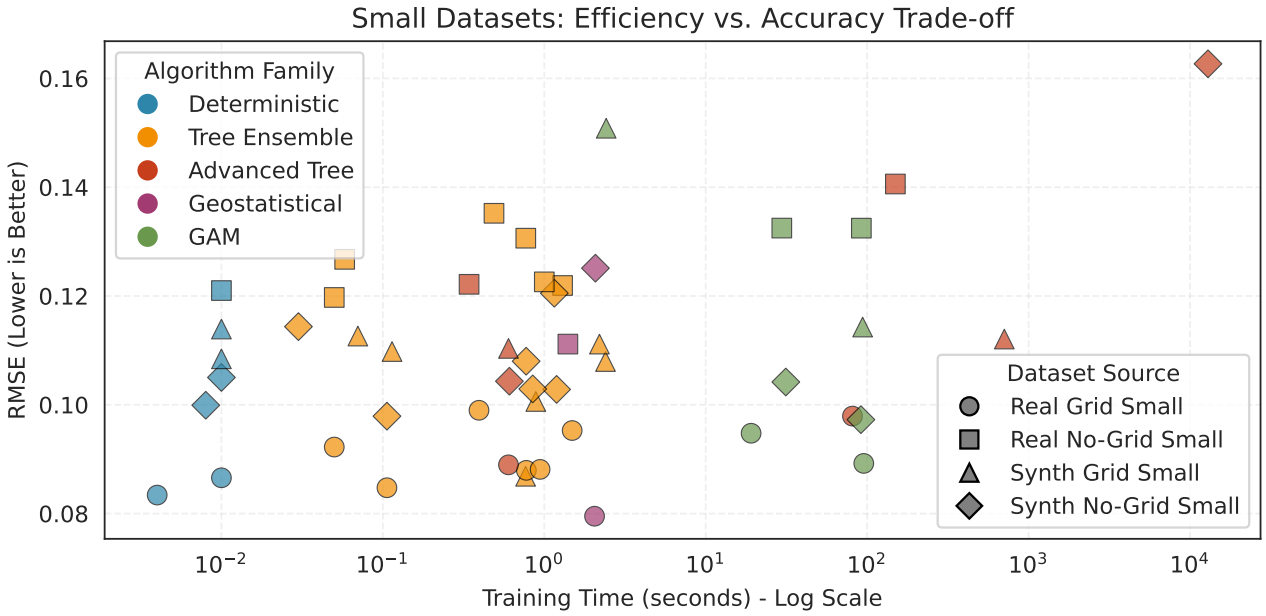
Our validation strategy employs a robust Randomized Search with K-Fold Cross-Validation. The procedure iterates through datasets and algorithms (a double loop), randomly sampling hyperparameter spaces. Each configuration is evaluated across K folds, with the "Best Model" selected based on the lowest average RMSE. We adapted tuning intensity to dataset size: high intensity search (20 iterations, 5-fold, 10-min timeout) for small data, and an efficiency focused regime (5 iterations, 3-fold) for large data.

5 Results

The study follows a two-phase selection process. First, we benchmark all algorithm families on small datasets (N 5,000) to map the initial tradeoffs between precision and computational cost. Based on these results, we preselect the most efficient candidates to undergo further tests on larger datasets and noisy scenarios, assessing their true scalability and robustness in different conditions.

5.1 Small Dataset Screening: Computational Efficiency vs Accuracy

As mentioned previously, we begin our analysis by running all algorithms on small datasets. This step allows us to evaluate the tradeoff between **Training Cost** and **Predictive Accuracy** before scaling up.



The figure below plots the Training Time (x-axis, log scale) against the RMSE (y-axis). The ideal algorithm would be located in the **bottom-left corner** (fast and accurate). It reveals different computational profiles for the families of algorithms. Using the results obtained, a selection of models is made for evaluation on larger datasets. This is a crucial step that avoids wasting computational resources on models that are not competitive in terms of training time.

The GAM, GeoRF exhibit extreme training times even on modest sample sizes. Extrapolating these computational demands to datasets an order of magnitude larger suggests prohibitive resource requirements that would render them impractical for operational use. For the case of Kriging, its complexity leads to training times that grow rapidly with sample size, making it infeasible for larger datasets. These methods are therefore excluded from further analysis, as their computational inefficiency outweighs any potential accuracy benefits they may offer.

The remaining candidates demonstrate more favorable computational profiles. Ensemble methods cluster in the efficient region of the accuracy time tradeoff space, delivering low prediction error with manageable training costs. Deterministic methods also remain in the analysis due to their near instantaneous training and competitive predictive performance. Their simplicity establishes a demanding baseline: any machine learning enhancement must deliver substantial accuracy gains to justify increased algorithmic complexity.

5.2 Large-Scale Performance Evaluation

5.2.1 Scalability Analysis

In the following step the scalability of the algorithms is evaluated using as comparison metric the **Augmentation Factor** defined as the ratio of time for large dataset over the time for small dataset. This metrics depends on the algorithm's parameter and specific dataset, so it is not rigorous from a statistical point of view, but it provides a useful measure of how well the training time scales as the dataset size increases. riscrivere che augmentation e 20X

Table 3

Model	Avg Time (Small)	Avg Time (Large)	Augmentation Factor
XGB	0.949s	0.59s	0.6x
Oblique RF	0.537s	0.57s	1.1x
XGB-CR	1.414s	1.55s	1.1x
MiGBT	0.730s	1.00s	1.4x
MiGBT-CR	1.246s	2.19s	1.8x
KNN	0.010s	0.03s	3.1x
RF	0.052s	0.17s	3.2x
IDW	0.005s	0.02s	4.2x
RF-CR	0.094s	0.48s	5.1x

The analysis on larger datasets shows different behaviour for the considered algortihms. **Deterministic methods** exhibit one of the highest augmentation factor both for the KNN and for IDW, however this not implies a bad scalability because it still remains the fastest one. Also the **Tree ensemble methods** are able to scale well, with a reasonable augmentation factors. The XGBoost based model shows a particular pattern, infact, its training time for the large datasets is lower than one of the small datasets.Next step is to move from large dataset of 100,000 rows to very large 1O times larger and extremely large 100 times larger than the large one.

Table 4

Model	Time (S-NG-Lg)	Time (S-NG-VLg)	Time (S-NG-ELg)
IDW	0.01s	0.38s	5.12s
KNN	0.02s	0.42s	4.78s
RF	0.08s	3.27s	44.33s
RF-CR	0.28s	12.86s	274.23s
Oblique RF	0.49s	12.41s	136.11s
XGB	0.78s	1.15s	12.34s
MiGBT	0.80s	18.91s	219.20s
MiGBT-CR	1.55s	55.78s	3821.62s
XGB-CR	1.87s	5.48s	34.19s

This table highlights the dramatic increase in training time as dataset size grows, especially for complex models like MiGBT-CR, whose time explodes to over an hour for the largest dataset. Simple models (IDW, KNN) scale much more gracefully, while tree-based methods (RF, XGB) show moderate scaling. The CR variants, which include additional computational steps, consistently require more time across all dataset sizes. We can conclude that ensemblist methods seems to be more adapted to scale large dataset but determinitic methods stays quicker on our example. Also all ensemblist methods are not equivalent, in fact Mi-GBT is 18 times slower than gradiant boosting, but it's important to keep in the memory that this part is hard to interprate because of the wrappers that are build allowing us to maintain the code of each algorithm.

5.2.2 Robustness to Noise

The previous data used, whether real or synthetic, was relatively smooth. However, real world scenarios often involve noisy data, such as housing prices, where variability cannot be perfectly modeled by coordinates alone. To test the models' resilience, we evaluated them on **three large synthetic datasets with increasing noise levels** (`S-NG-Lg-N1`, `N2`, `N3`) and one real noisy dataset: **California Housing**, trying to highlight the impact of coordinate rotation on RMSE in these conditions.

Table 5

Model	RMSE (Clean)	RMSE (N1)	RMSE (N2)	RMSE (N3)	Max Degr.	Housing RMSE
KNN	0.105	0.453	0.763	1.351	+1189%	0.556
IDW	0.100	0.451	0.747	1.453	+1358%	0.560
Oblique RF	0.146	0.451	0.755	1.321	+804%	0.518
Random Forest	0.156	0.452	0.748	1.316	+745%	0.546
Random Forest + CR	0.116	0.444	0.749	1.329	+1047%	0.510
XGBoost	0.163	0.459	0.756	1.318	+706%	0.547
XGBoost + CR	0.129	0.453	0.748	1.318	+924%	0.528
MiGBT	0.154	0.461	0.768	1.350	+775%	0.548
MiGBT + CR	0.124	0.454	0.761	1.338	+980%	0.529

Noise Robustness:

A primary observation is that while the integration of CR consistently enhances model accuracy in low noise environments—notably yielding a substantial reduction in **Root Mean Square Error (RMSE)** for both Random Forest (RF) and XGBoost—this advantage tends to diminish as the signal to noise ratio decreases. As noise levels intensify, the performance gap between standard and CR augmented versions converges. In certain high noise scenarios, the CR variants even exhibit a marginal increase in error, suggesting a practical threshold for the technique. From a computational efficiency standpoint, this indicates that the overhead of applying CR may not be justified when data quality is significantly compromised, as the incremental gains in precision no longer offset the increased processing time.

Furthermore, the data reveals a critical shift in model hierarchy as environmental complexity grows. While the **K-Nearest Neighbors (KNN)** algorithm demonstrates superior precision on clean datasets, it proves particularly susceptible to stochastic interference. This vulnerability is quantified by the **Maximum Degradation (Max Degr.)**, where KNN shows a significantly higher rate of error escalation compared to its ensemble based counterparts. Such a trend points to a structural limitation inherent in deterministic models: their foundational assumptions of simplicity and local continuity struggle to capture underlying patterns once they are obscured by heavy noise. Finally, the analysis of the **California Housing** dataset assests with real world examples.

5.2.3 Impact of Coordinate Rotation

Having established computational feasibility and noise robustness patterns, attention turns to quantifying the specific contribution of coordinate rotation across diverse data conditions. The comparative evaluation considers high volume datasets characterized by varying degrees of noise and structural complexity, consolidating observations from previous experiments into a coherent assessment framework. For each ensemble algorithm, performance is contrasted between standard implementations and their coordinate rotated counterparts, examining both prediction error reduction and the associated computational burden required to achieve these gains.

Table 6

Dataset	Algo	RMSE (Std)	RMSE (CR)	Imp.	T-Std(s)	T-CR(s)	Delay
RGEALTI	RF	0.0052	0.0020	+61.3%	0.3	0.7	+179%
	XGB	0.0063	0.0024	+61.4%	0.6	1.5	+134%
	MixGB	0.0063	0.0022	+65.2%	1.2	2.9	+141%
BDALTI	RF	0.3328	0.2878	+13.5%	0.2	0.6	+173%
	XGB	0.3537	0.2962	+16.2%	0.5	1.2	+124%
	MixGB	0.3536	0.2965	+16.1%	1.2	2.5	+110%
Housing	RF	0.5455	0.5100	+6.5%	0.0	0.1	+275%
	XGB	0.5466	0.5283	+3.4%	0.5	0.9	+76%
	MixGB	0.5485	0.5292	+3.5%	0.5	1.1	+132%
Syn-Grid	RF	0.1617	0.1248	+22.8%	0.1	0.3	+172%
	XGB	0.1883	0.1216	+35.4%	0.4	1.6	+307%
	MixGB	0.1550	0.1329	+14.3%	0.8	1.8	+121%
Syn-NoGrid	RF	0.1557	0.1158	+25.6%	0.1	0.3	+270%
	XGB	0.1635	0.1287	+21.3%	0.8	1.9	+141%
	MixGB	0.1543	0.1239	+19.7%	0.8	1.6	+93%
Syn-NG-N1	RF	0.4517	0.4445	+1.6%	0.1	0.2	+100%
	XGB	0.4594	0.4526	+1.5%	0.5	1.0	+90%
	MixGB	0.4608	0.4540	+1.5%	0.7	1.3	+84%
Syn-NG-N2	RF	0.7480	0.7494	-0.2%	0.1	0.2	+115%
	XGB	0.7564	0.7481	+1.1%	0.3	0.5	+59%
	MixGB	0.7679	0.7611	+0.9%	0.7	1.3	+89%
Syn-NG-N3	RF	1.3161	1.3289	-1.0%	0.1	0.2	+124%
	XGB	1.3184	1.3182	+0.0%	0.4	1.0	+172%
	MixGB	1.3504	1.3380	+0.9%	0.8	1.6	+98%
Syn-VeryLarge	RF	0.5462	0.4994	+8.6%	3.3	12.9	+293%
	XGB	0.5835	0.5775	+1.0%	1.2	5.5	+375%
	MixGB	0.5836	0.5733	+1.8%	18.9	55.8	+195%
Syn-ExtremLarge	RF	0.5331	0.3844	+27.9%	44.3	274.2	+519%
	XGB	0.5742	0.5540	+3.5%	12.3	34.2	+177%
	MixGB	0.5818	0.5683	+2.3%	219.2	3821.6	+1643%

The results reveal a clear relationship between noise levels and the effectiveness of coordinate rotation. On structured datasets with low noise, CR delivers substantial improvements RMSE reductions can exceed 60% on the RGEALTI dataset. However, these gains diminish progressively as noise increases. In highly noisy synthetic scenarios, the benefit becomes marginal or disappears entirely. Real world datasets like California Housing show different behavior: when spatial autocorrelation dominates the signal, CR remains effective despite noise. This pattern indicates that CR is not a universal solution but works best when geographic structure genuinely drives the predictions rather than serving as a weak proxy for other factors.

The computational cost of coordinate rotation deserves attention. Training time increases substantially because the algorithm must process the expanded feature space. However, the absolute time penalty remains modest. Even with rotation, training completes within seconds rather than minutes for these datasets, maintaining practical feasibility. The augmented algorithms stay competitive with their standard counterparts in terms of wall clock time, simply shifting from near instantaneous to still fast execution.

These observations suggest that coordinate rotation offers favorable tradeoffs for operational spatial modeling. The accuracy improvements justify the computational overhead when working with real world geographic data where spatial patterns matter. However, practitioners should assess their data characteristics before committing to the approach. When noise dominates the signal or coordinates correlate poorly with outcomes, the added complexity yields diminishing returns. The technique works when spatial structure exists; it cannot create predictive power where coordinates lack information.

5.3 Discussion

This benchmark evaluated the efficacy of ensemble learning methods against traditional spatial interpolation techniques, specifically rigorously testing the hypothesis that Coordinate Rotation (CR) can overcome the geometric limitations of axis aligned decision trees.

Our hierarchical evaluation confirms that while Geostatistical methods (Ordinary Kriging) offer theoretical robustness, they hit a computational ceiling with larger datasets, making them unsuitable for high density applications. On the other hand, Deterministic methods (IDW, KNN) provide the fastest execution but lack resilience, exhibiting significant performance degradation in the presence of noise. Ensemble methods therefore emerge as the most balanced solution for large scale interpolation, offering superior scalability compared to Kriging and greater robustness than deterministic baselines.

The impact of Coordinate Rotation, however, is shown to be dependent by the context of the application rather than universally superior. On structured, gridded datasets, CR delivers substantial accuracy gains . However, this advantage diminishes in noisy or irregularly sampled environments. In these scenarios, the prediction error is driven by the noise rather than geometric constraints, rendering the augmented feature space less effective.

Ultimately, while CR introduces a slight computational overhead, the training time remains in the order of seconds orders of magnitude faster than Geostatistical approaches. This establishes Ensemble Learning with Coordinate Rotation as a reccomended strategy for large, structured spatial datasets, while in case noise dominates the signal or when spatial coordinates provide weak predictive information simpler methods may suffice.

6 Conclusion

The results of this benchmark show that Coordinate Rotation (CR) is a powerful tool for improving tree based models, but it is not a universal fix for all spatial interpolation problems. On structured, gridded datasets, adding rotation allowed algorithms like Random Forest to reduce prediction errors by up to 60%, effectively overcoming the geometric limitations of standard decision trees. Crucially, these methods achieve high accuracy while remaining computationally efficient, avoiding the scaling issues that make Kriging unusable on datasets larger than 10,000 points.

However, our tests also reveal a clear limitation: the benefits of coordinate rotation disappear in high noise environments. When the signal quality is low, the added complexity of the rotated features does not translate into better predictions. Practically, this means that for large, clean topographic data, ensemble methods with CR are the best choice. In contrast, when dealing with noisy data or smaller sample sizes, simpler deterministic methods or traditional Kriging often provide a better balance of effort and accuracy.

Future research should explore how these methods perform when additional covariates are included, beyond simple coordinates, to see if the advantages of machine learning hold in more complex multivariate scenarios.

7 References

Aalto, Juha, et al. “Spatial Interpolation of Monthly Climate Data for Finland: Comparing the Performance of Kriging and Generalized Additive Models.” *Theoretical and Applied Climatology*, vol. 112, no. 1, 2013, pp. 99-111.

Anava, Oren, and Kfir Levy. “k-*Nearest Neighbors: From Global to Local.”* Advances in Neural Information Processing Systems*, vol. 29, 2016.

Barry, Ronald Paul, and M. Jay Ver Hoef. “Blackbox Kriging: Spatial Prediction without Specifying Variogram Models.” *Journal of Agricultural, Biological, and Environmental Statistics*, 1996, pp. 297-322.

Bentéjac, Candice, Anna Csörgő, and Gonzalo Martínez-Muñoz. “A Comparative Analysis of Gradient Boosting Algorithms.” *Artificial Intelligence Review*, vol. 54, no. 3, 2021, pp. 1937-1967.

Borm, Conrad. “Kriging Models: An Extensive Analysis of Their Current State, Applicability, Advantages and Limitations.” 2024. Utrecht University Student Theses, https://studenttheses.uu.nl/handle/20.500.12932/48040.

Bostan, Pınar. “Basic Kriging Methods in Geostatistics.” *Yuzuncu Yıl University Journal of Agricultural Sciences*, vol. 27, no. 1, 2017, pp. 10-20.

Chen, Meifang, Changho Lee, and Yongwan Chun. “A Machine Learning Approach Using Spatially Explicit K-Nearest Neighbors for House Price Predictions.” *ISPRS International Journal of Geo-Information*, vol. 15, no. 1, 2026, p. 46.

Ciampiconi, Lorenzo, et al. “A Survey and Taxonomy of Loss Functions in Machine Learning.” *arXiv preprint arXiv:2301.05579*, 2023.

Cressie, Noel. *Statistics for Spatial Data*. John Wiley & Sons, 2015.

Dobson, Annette J., and Adrian G. Barnett. *An Introduction to Generalized Linear Models*. Chapman and Hall/CRC, 2018.

Gaudart, Jean, et al. “Oblique Decision Trees for Spatial Pattern Detection: Optimal Algorithm and Application to Malaria Risk.” *BMC Medical Research Methodology*, vol. 5, no. 1, 2005, p. 22.

Geerts, Margot, Seppe Vanden Broucke, and Jochen De Weerdt. “A Spatial Loss Function for Gradient Boosted Trees.” *CEUR Workshop Proceedings*, R. Piskac c/o Redaktion Sun SITE Informatik V RWTH Aachen, 2024.

Geerts, Margot, Seppe Vanden Broucke, and Jochen De Weerdt. “GeoRF: A Geospatial Random Forest.” *Data Mining and Knowledge Discovery*, vol. 38, no. 6, 2024, pp. 3414-3448.

Hu, Hongda, and Hong Shu. “An Improved Coarse-Grained Parallel Algorithm for Computational Acceleration of Ordinary Kriging Interpolation.” *Computers & Geosciences*, vol. 78, 2015, pp. 44-52.

Kern, Christoph, Thomas Klausch, and Frauke Kreuter. “Tree-Based Machine Learning Methods for Survey Research.” *Survey Research Methods*, vol. 13, no. 1, 2019.

Lu, George Y., and David W. Wong. “An Adaptive Inverse-Distance Weighting Spatial Interpolation Technique.” *Computers & Geosciences*, vol. 34, no. 9, 2008, pp. 1044-1055.

Nelder, John Ashworth, and Robert WM Wedderburn. “Generalized Linear Models.” *Journal of the Royal Statistical Society Series A: Statistics in Society*, vol. 135, no. 3, 1972, pp. 370-384.

Oliver, M. A., and R. Webster. “A Tutorial Guide to Geostatistics: Computing and Modelling Variograms and Kriging.” *Catena*, vol. 113, 2014, pp. 56-69.

Pilz, Jürgen, and Gunter Spöck. “Why Do We Need and How Should We Implement Bayesian Kriging Methods.” *Stochastic Environmental Research and Risk Assessment*, vol. 22, no. 5, 2008, pp. 621-632.

Syam, Niladri, and Rajeeve Kaul. “Random Forest, Bagging, and Boosting of Decision Trees.” *Machine Learning and Artificial Intelligence in Marketing and Sales: Essential Reference for Practitioners and Data Scientists*, Emerald Publishing Limited, 2021, pp. 139-182.

Wong, David WS. “Interpolation: Inverse-Distance Weighting.” *International Encyclopedia of Geography: People, the Earth, Environment and Technology: People, the Earth, Environment and Technology*, 2016, pp. 1-7.

Wood, Simon N., Yannig Goude, and Simon Shaw. “Generalized Additive Models for Large Data Sets.” *Journal of the Royal Statistical Society Series C: Applied Statistics*, vol. 64, no. 1, 2015, pp. 139-155.