

Review

Comparative Study of Regression Models for Continuous Function Approximation

Tamás Storcz ^{1,2} , Gábor Gyurák ^{2,3} and Zsolt Ercsey ^{2,3,*} 

¹ Department of Systems and Software Technologies, Faculty of Engineering and Information Technology, University of Pécs, Boszorkány Street 2, H-7624 Pécs, Hungary; storcz.tamas@mik.pte.hu

² Autonomous Technologies and Artificial Intelligence Research Team, Faculty of Engineering and Information Technology, University of Pécs, Boszorkány Street 2, H-7624 Pécs, Hungary; gyurak.gabor@mik.pte.hu

³ Department of Cybersecurity and Networks, Faculty of Engineering and Information Technology, University of Pécs, Boszorkány Street 2, H-7624 Pécs, Hungary

* Correspondence: ercsey.zsolt@mik.pte.hu

Abstract

Regression models are widely used for continuous function approximation in applied research, yet selecting an appropriate model remains challenging for applied users who must balance predictive accuracy, interpretability, robustness, computational cost, and preprocessing requirements. This methodological review provides a decision-oriented synthesis of regression model families, preprocessing strategies, and evaluation criteria for transparent and reproducible model selection. The reviewed methods are organized by modeling principle, including linear and regularized models, robust and distribution-aware estimators, online learning methods, tree-based ensembles, kernel-based and probabilistic approaches, instance-based regressors, neural networks, and symbolic regression. The main contribution is a practical framework that connects data characteristics, including linearity, dimensionality, feature scale, target distribution, noise, outliers, and sample size, with suitable model families, preprocessing choices, and performance metrics. The review distinguishes theoretical guarantees, empirical tendencies, and implementation-dependent behavior because properties such as robustness, interpretability, scalability, and approximation capacity cannot be reduced to universal binary categories. The resulting comparative tables and decision criteria provide a compact reference for applied researchers designing regression workflows that are theoretically grounded, practically feasible, and aligned with research objectives.

Keywords: comparative analysis; regression model; function approximation; data scaling; evaluation metrics



Academic Editors: Gabriele Gianini and Haridimos Kondylakis

Received: 10 April 2026

Revised: 15 June 2026

Accepted: 3 July 2026

Published: 7 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Function approximation is a central task in scientific computing, statistics, and machine learning. Its goal is to estimate an unknown relationship between input variables and a continuous output from finite and often noisy observations. Such problems arise in engineering, physics, economics, biology, energy systems, medical modeling, and many other quantitative fields. In these contexts, regression is not only a prediction tool, but part of a wider modeling workflow that includes data preparation, model selection, validation, and interpretation.

Regression models differ substantially in their assumptions and practical behavior. Linear and regularized models offer interpretability and computational efficiency, while ro-

bust estimators address outliers and non-Gaussian errors. Kernel methods, instance-based regressors, tree ensembles, neural networks, and genetic programming provide greater flexibility for nonlinear approximation, but often require more careful preprocessing, tuning, or interpretation. For this reason, model selection should be treated as a methodological design decision rather than as a simple comparison of predictive scores.

A meaningful regression workflow must consider the compatibility between the model, the data preparation procedure, and the evaluation metric. Regularized and kernel-based models are typically scale-sensitive, distance-based methods depend strongly on feature magnitudes, and neural networks often benefit from both input and output scaling. At the same time, evaluation metrics emphasize different aspects of error behavior: squared-error metrics penalize large residuals, absolute-error metrics are more robust, percentage metrics support relative interpretation under suitable conditions, and quantile losses allow asymmetric error evaluation.

The main added value of this review is its process-oriented synthesis of these elements (see Figure 1). Instead of treating regression algorithms, scaling methods, and evaluation metrics separately, the paper connects them into a unified framework for practical model selection in continuous function approximation. The core contribution is presented through comparative tables that summarize model properties, recommended scaling strategies, and metric characteristics. These tables translate the theoretical discussion into a compact decision aid for constructing transparent, reproducible, and practically applicable regression workflows.

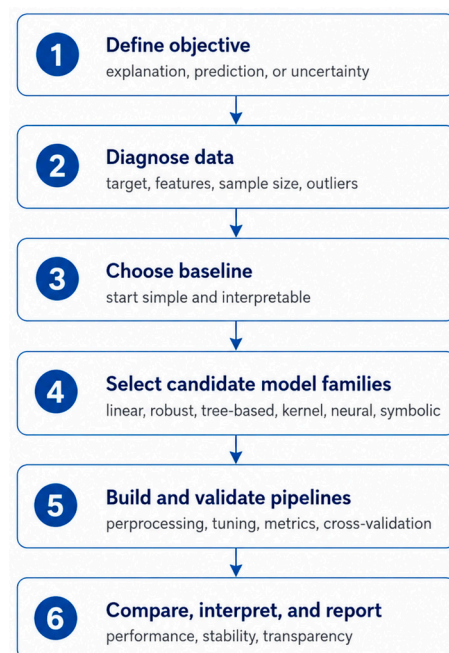


Figure 1. Main steps of regression workflow generation.

The paper is organized as follows. Section 2 defines the review scope, method-selection criteria, and dimensions of comparison. Section 3 reviews the main regression model families and their methodological foundations. Section 4 synthesizes application domains and methodological trends from the reviewed literature. Section 5 translates these observations into decision dimensions for practical model selection. Section 6 discusses input and output scaling recommendations, and Section 7 reviews evaluation metrics and diagnostic criteria. Section 8 concludes by summarizing how model assumptions, preprocessing choices, validation design, and performance measures should be aligned in continuous function approximation.

2. Review Scope and Methodology

The objective of this review is to provide a structured and practice-oriented synthesis of regression models for continuous function approximation. The paper is intended for researchers and practitioners who need to select, apply, and evaluate regression models but who may not have a formal background in computer science, statistics, or mathematical optimization. Therefore, the review does not aim to provide an exhaustive theoretical treatment of every algorithm. Instead, it focuses on methodological properties that are directly relevant to practical model selection, including data requirements, preprocessing sensitivity, interpretability, robustness, scalability, uncertainty handling, and evaluation strategy.

The review follows a methodological and comparative narrative design. It does not introduce a new regression algorithm and does not conduct a systematic meta-analysis of all regression studies. Its contribution lies in organizing widely used regression approaches into a decision-support framework that connects three elements that are often treated separately: the regression model, the preprocessing strategy, and the performance metric. This process-oriented perspective is intended to support transparent and reproducible regression workflows in applied continuous function approximation.

2.1. Scope and Selection of Regression Methods

The regression methods included in this review were selected according to four criteria. First, the method had to be applicable to continuous target variables. Second, it had to represent a distinct modeling principle, such as linear modeling, regularization, robust estimation, distribution-aware regression, online learning, kernel-based learning, instance-based learning, tree-based partitioning, ensemble learning, neural approximation, or symbolic regression. Third, the method had to be sufficiently established in the methodological literature or commonly available in widely used scientific software environments. Fourth, it had to be relevant to practical model selection problems encountered in applied research.

Based on these criteria, the review includes classical linear models, regularized linear models, robust estimators, generalized and distribution-aware regressors, online learning methods, support-vector approaches, instance-based regressors, decision-tree models, ensemble methods, neural-network models, and genetic-programming-based symbolic regression. This selection is intended to cover the main modeling families encountered in applied continuous regression tasks rather than every possible implementation, variant, or software-specific extension.

The scope of the review is limited to regression models for continuous outputs. Classification, survival analysis, causal inference, and time-series forecasting are not treated as separate methodological domains, although some of the reviewed models can be adapted to these settings as well. The review also emphasizes general model-selection principles rather than implementation-specific details. Individual software libraries may differ in optimization algorithms, default hyperparameters, missing-value handling, categorical-variable support, and uncertainty estimation. Therefore, the reported properties should be interpreted as typical characteristics of model families rather than absolute guarantees of every implementation.

2.2. Organization and Dimensions of Comparison

To improve readability and avoid treating regression algorithms as isolated techniques, the reviewed methods are grouped into model families. Linear and regularized methods are discussed first because they form the conceptual baseline for many regression workflows. Robust and distribution-aware methods are then considered as extensions that address outliers, skewed responses, heteroscedasticity, or non-standard error behavior. Online

learning methods are treated separately because their primary advantage lies in scalability and incremental updating.

Nonlinear approximation methods are organized according to their learning principle. Polynomial and kernel-based methods introduce nonlinear structure through explicit transformations or implicit mappings. Instance-based methods rely on local similarity rather than global parameter estimation. Tree-based and ensemble methods approximate nonlinear relationships through recursive partitioning and aggregation. Neural-network models are discussed as flexible approximators whose performance depends strongly on data volume, architecture, optimization, and scaling. Finally, genetic programming and memetic genetic programming are included as symbolic regression approaches, where interpretability is linked to the discovery of explicit analytical expressions.

The comparative analysis is based on practical decision dimensions that directly affect model choice in applied regression problems. These dimensions include the expected functional form, scale sensitivity, robustness, interpretability, scalability and computational cost, uncertainty or distributional information, and tuning complexity. They were selected because they frequently determine whether a regression model is suitable for a given dataset, research objective, and computational setting.

These dimensions should be interpreted as diagnostic axes rather than fixed rankings. For example, models based on distances, gradients, regularization penalties, or kernel evaluations are often affected by feature magnitudes, whereas tree-based models are usually less sensitive to monotone transformations of individual numerical features. Similarly, robustness, interpretability, and scalability depend not only on the model family but also on dataset size, noise structure, hyperparameter settings, preprocessing choices, and validation strategy.

2.3. Integration of Preprocessing, Evaluation, and Practical Synthesis

A central methodological assumption of this review is that regression model selection cannot be separated from preprocessing and evaluation. The same model may behave differently depending on how input features and target variables are scaled, transformed, encoded, or filtered. Similarly, the preferred model may change depending on whether the evaluation criterion emphasizes large errors, average absolute deviations, relative errors, explained variance, or asymmetric loss.

For this reason, data scaling methods and regression evaluation metrics are treated as integral parts of the modeling workflow. Input scaling is discussed in relation to models that depend on feature magnitudes, such as regularized linear models, support-vector regression, nearest-neighbor regression, online optimization methods, and neural networks. Output scaling is considered for target variables with large dynamic ranges, skewed distributions, or unstable loss magnitudes. Evaluation metrics are compared according to their sensitivity to outliers, interpretability, scale dependence, and suitability for different target-variable properties.

The synthesis proceeds in three steps. First, each regression method is described in terms of its modeling principle, assumptions, strengths, limitations, and typical preprocessing requirements. Second, the methods are compared across the decision dimensions described above. Third, these comparisons are translated into compact tables and workflow recommendations that support practical model selection.

The resulting framework is not intended to rank regression models universally. There is no single best method across all datasets, domains, and modeling objectives. Instead, the framework encourages users to begin with the problem and data characteristics, then select candidate model families, choose compatible preprocessing steps, and evaluate the models using metrics aligned with the scientific or practical objective. Final model choice

should still be based on validation using the user's own data, appropriate cross-validation procedures, domain knowledge, and sensitivity analysis.

3. Regression Model Families

This section presents the regression methods considered in this review by organizing them into model families rather than treating them as isolated algorithms. The purpose of this organization is to support practical model selection. Models within the same family often share assumptions, preprocessing requirements, strengths, and limitations, even if their optimization procedures or implementation details differ. The discussion therefore focuses on the modeling principle of each family, the theoretical foundation of the individual methods, and the implications for continuous function approximation.

3.1. Linear and Regularized Regression Models

Linear regression is the natural baseline for continuous function approximation when the relationship between predictors and the target variable can be reasonably approximated by an additive linear structure. Ordinary Least Squares (OLS, [1]) estimates the coefficient vector by minimizing the residual sum of squares in the model. Under the Gauss–Markov assumptions, the OLS estimator is the Best Linear Unbiased Estimator (BLUE, [2]). Its main advantages are computational efficiency, coefficient-level interpretability, and conceptual transparency. However, OLS is sensitive to multicollinearity, high-dimensional predictor spaces, leverage points, and violations of linearity. It cannot represent nonlinear relationships in the original feature space unless explicit feature transformations are introduced.

Polynomial regression extends the linear modeling framework by applying OLS to polynomially transformed input variables [3]. Although the model remains linear in its coefficients, polynomial basis functions allow it to represent smooth nonlinear trends and interactions. This makes polynomial regression useful as an intermediate approach between simple linear models and more flexible nonlinear regressors. Its practical limitation is that the number of polynomial terms grows rapidly with the degree and the number of input variables, which increases the risk of overfitting and makes feature scaling important.

Regularized linear models modify the least-squares objective by adding penalty terms that control the coefficient magnitude and reduce variance. Lasso regression, introduced as the least absolute shrinkage and selection operator [4], adds an L1 penalty to the loss function. This encourages sparse coefficient vectors and can perform implicit feature selection by shrinking some coefficients exactly to zero. Lasso is particularly useful in high-dimensional settings, but its behavior may be unstable when predictors are strongly correlated.

Ridge regression adds an L2 penalty to the least-squares objective [5], corresponding to Tikhonov regularization [6]. Unlike Lasso, Ridge does not generally set coefficients to zero, but it shrinks them continuously. This is useful when multicollinearity is present or when many predictors contribute small but meaningful effects. Ridge regression often improves predictive stability compared with OLS, although it sacrifices the direct unbiasedness of the unpenalized estimator.

Elastic Net combines L1 and L2 penalties [7]. It inherits the sparsity-inducing property of Lasso and the grouping effect of Ridge, making it useful when predictors are numerous and correlated. From a practical perspective, Elastic Net is often preferable to Lasso when groups of correlated variables are expected to contribute jointly to the response. As with other regularized models, standardization is usually required because the penalty acts on coefficient magnitudes and can otherwise penalize variables unevenly.

Bayesian Ridge regression introduces a probabilistic interpretation of coefficient shrinkage by placing Gaussian priors on the regression coefficients and estimating posterior distributions [8]. The posterior mean has a Ridge-like form under fixed prior and noise pre-

cisions, while the Bayesian formulation additionally allows uncertainty about coefficients and hyperparameters to be incorporated. This makes Bayesian Ridge useful when regularization and uncertainty characterization are both relevant, although the interpretation of uncertainty depends on the correctness of the assumed probabilistic model.

Taken together, linear and regularized models are most appropriate when interpretability, computational efficiency, or baseline comparison is important. They are less suitable when the underlying relationship is strongly nonlinear and cannot be captured by predefined transformations. Their main methodological advantage is transparency; their main limitation is restricted functional flexibility.

3.2. Online and Large-Scale Linear Optimization Methods

Some regression settings require incremental learning, low memory consumption, or rapid model updates as new data arrive. In these cases, online and stochastic optimization methods can be more appropriate than closed-form or batch estimators.

The Stochastic Gradient Descent Regressor is based on stochastic gradient descent [9]. Instead of computing the gradient over the full dataset, it updates the model using individual observations or mini-batches. This makes it scalable to large datasets and streaming contexts. It can be combined with several loss functions, including squared, absolute, Huber, and ϵ -insensitive losses, and with L1, L2, or Elastic Net regularization. Its performance, however, depends strongly on the learning-rate schedule, feature scaling, regularization strength, and stopping criterion.

Passive-Aggressive regression is another online learning approach [10]. The model remains passive when the current prediction is sufficiently accurate and updates aggressively when the error exceeds a predefined tolerance. Its update rule is derived from a convex optimization problem that balances correction of the current error against minimal change to the existing model. Extensions of Passive-Aggressive learning provide theoretical guarantees for online regression and related tasks [11]. The method is attractive in real-time settings because it avoids explicit learning-rate tuning, but it remains sensitive to the aggressiveness parameter and the error tolerance.

Online linear methods are useful when computational constraints dominate the modeling problem. They are usually less flexible than nonlinear models, but they provide an efficient way to update regression functions under changing or continuously arriving data.

3.3. Distribution-Aware and Robust Regression Models

Classical least-squares regression is strongly influenced by large residuals and assumes a particular error structure. Distribution-aware and robust regression methods address situations in which the target variable is skewed, heteroscedastic, semicontinuous, or contaminated by outliers.

Tweedie regression models the response using the Tweedie family of exponential dispersion models [12]. This family is characterized by a mean–variance relationship of the form $\text{Var}(Y) = \varphi\mu^p$, where Y is the response variable, $\mu = \mathbb{E}(Y)$ is its mean, φ is the dispersion parameter, and p controls the variance function. Depending on the value of p , the family includes Gaussian, Poisson, Gamma, inverse Gaussian, and compound Poisson–Gamma cases. Tweedie regression is therefore appropriate when the target variable is positive, skewed, semicontinuous, or includes a mass at zero. Its usefulness depends on whether the selected distributional form adequately represents the data-generating process.

Quantile regression estimates conditional quantiles rather than the conditional mean [13]. It minimizes an asymmetric absolute loss function, often called pinball loss, and generalizes median regression to arbitrary quantile levels. This makes it useful when the

conditional distribution of the response is asymmetric or heteroscedastic, or when upper and lower prediction behaviors are of separate interest. Quantile regression is not primarily a robust replacement for all least-squares models; rather, it provides a distributional view of regression by describing different parts of the conditional response distribution.

Huber regression uses a loss function that is quadratic for small residuals and linear for large residuals [14]. This design preserves the efficiency of squared-error regression for small deviations while reducing the influence of large residuals. Huber regression is useful when moderate outliers are present, but it does not provide high-breakdown robustness against leverage points by itself. Its behavior depends on the transition parameter that determines when the loss changes from quadratic to linear.

The Theil–Sen estimator was developed by Theil [15], extended by Sen [16], and later summarized in a broader methodological form [17]. In simple linear regression, it estimates the slope as the median of all pairwise slopes. This median-based construction gives the estimator a substantially higher tolerance to contamination than OLS in simple regression settings. Its multivariate extensions are more complex, and its computational cost can increase substantially with sample size.

Random Sample Consensus, or RANSAC, fits regression models by repeatedly sampling minimal subsets, estimating candidate models, and selecting the model with the largest consensus set of inliers [18]. It is useful when a large fraction of the data may be contaminated by gross outliers. However, RANSAC is stochastic, depends on inlier thresholds and iteration counts, and may be unstable if the assumed inlier structure is poorly specified.

These methods should be considered when the main modeling challenge is not only functional approximation but also error behavior. They are especially relevant when residual distributions are non-Gaussian, when outliers are expected, or when the mean is not the only scientifically meaningful summary of the conditional response.

3.4. Tree-Based Regression and Ensemble Methods

Tree-based regression models approximate continuous responses by recursively partitioning the feature space. They are attractive in applied research because they can represent nonlinear relationships and interactions without requiring explicit feature engineering or strict parametric assumptions.

Decision tree regression follows the recursive partitioning principle of CART-style regression trees [19,20]. The feature space is split into regions, and each leaf typically predicts the mean response of observations in that region. Single trees are interpretable because their decisions can be traced through split rules. However, they are prone to overfitting and can be unstable: small changes in the training data may lead to different tree structures.

Random Forest regression reduces the variance of single trees by averaging predictions from many trees trained with sample and feature randomization [21]. Bootstrap or subsampled training sets and randomized feature selection reduce correlations among trees. Random Forests are flexible, usually strong baselines for nonlinear tabular regression, and relatively insensitive to monotone transformations of individual features. Their limitations include reduced interpretability compared with single trees and potentially high memory cost when many trees are used.

Extremely randomized trees, or Extra Trees, introduce additional randomness by selecting split thresholds randomly before choosing the best among candidate splits [22]. This often reduces variance and computational cost relative to more exhaustive split-search procedures, although it may increase bias in some settings. Extra Trees are useful when speed and robustness through aggregation are important.

Gradient Boosting regression builds an additive model sequentially [23]. Each new weak learner, usually a shallow tree, is fitted to the negative gradient of the loss function with respect to the current predictions. This allows the model to focus on observations or regions that previous learners predicted poorly. Gradient boosting can achieve high predictive accuracy, but it is sensitive to hyperparameters such as learning rate, tree depth, number of estimators, and subsampling strategy.

Histogram-Based Gradient Boosting accelerates gradient boosting by discretizing continuous features into bins and searching for approximate split points using aggregated gradient statistics [24]. This reduces computational and memory requirements and makes boosting more scalable to large datasets. The approximation introduced by binning is often acceptable when the number of bins and the binning strategy are well chosen. Handling of missing values or categorical variables depends on the specific implementation and should not be treated as an inherent property of histogram binning alone.

Tree ensembles are often strong practical candidates for nonlinear tabular regression. They generally require less feature scaling than distance-based, kernel-based, or gradient-based neural methods. Their main trade-off is interpretability: a single tree is transparent but unstable, whereas ensembles are more accurate and stable but require summary tools, feature importance measures, or post hoc explanation methods for interpretation.

3.5. Kernel-Based and Instance-Based Regression Models

Kernel-based and instance-based methods provide nonlinear flexibility by relying on similarity relationships rather than explicit global linear structure.

Support Vector Regression formulates regression as a convex optimization problem with an ϵ -insensitive loss and margin-based regularization [25]. Through the kernel trick, SVR can implicitly map input data into high-dimensional feature spaces and represent nonlinear relationships. The hyperparameters (C), ϵ , and the kernel parameters control the trade-off between model complexity, tolerance to errors, and generalization. SVR can be effective for small-to-medium datasets, but its computational cost and prediction time may increase substantially with the number of support vectors.

ν -SVR is a related formulation in which the parameter μ controls the fraction of support vectors and margin errors [26]. This can make model calibration more interpretable than independently tuning (C) and ϵ , although practical performance still depends on the kernel and scaling of the features. Like classical SVR, ν -SVR inherits the ability to model nonlinear relationships through kernels while remaining sensitive to hyperparameter choices.

K-Nearest Neighbors regression predicts a response by averaging the target values of the (k) nearest training observations in feature space [27]. It is non-parametric and requires no explicit training beyond storing the data. Its flexibility comes from local averaging, but its performance depends strongly on the choice of (k), the distance metric, feature scaling, and the density of the training data. Prediction can be expensive for large datasets unless efficient indexing or approximation methods are used.

Radius Neighbors regression generalizes the nearest-neighbor idea by averaging observations within a fixed radius around the query point [28]. Unlike KNN, the number of contributing neighbors can vary by region: dense regions may contain many neighbors, while sparse regions may contain few or none. This makes the method adaptive to local density, but it also introduces sensitivity to radius selection and fallback strategies for empty neighborhoods.

Kernel-based and instance-based models are useful when local similarity or nonlinear structure is central. They are usually scale-sensitive because distances and kernels depend directly on feature magnitudes. For non-specialist users, this means that preprocessing

and distance or kernel choice should be treated as part of the model, not as a separate technical detail.

3.6. Neural-Network Regression Models

Neural networks approximate nonlinear functions through layers of weighted transformations and nonlinear activation functions [29]. In regression, the output layer is typically designed to produce continuous predictions, while the hidden layers learn intermediate representations of the input variables. Their training is usually based on backpropagation and stochastic optimization. Neural networks are highly flexible, but they require careful tuning, sufficient data, and appropriate scaling.

Deep Neural Network regression extends standard feedforward neural regression by using multiple hidden layers. Its theoretical motivation is related to universal approximation results, which show that neural networks with suitable activation functions can approximate continuous functions on compact domains under sufficient capacity [30]. Depth can also allow the network to represent compositional structures more efficiently than shallow networks in certain problem classes [31]. Training is commonly performed using adaptive optimizers such as Adam [32]. Since the optimization problem is non-convex, performance depends on architecture, initialization, learning rate, batch size, regularization, and dataset size. Overfitting can be mitigated through early stopping, weight decay, dropout, batch normalization, and related techniques [33,34].

Modern neural-network regression architectures adapt the general neural modeling principle to specific data structures. Residual networks use shortcut connections to improve the trainability of deeper models [35], and dense residual-style variants can be adapted to tabular or continuous-input regression settings [36]. Convolutional and Temporal Convolutional Networks are useful when input variables have spatial, temporal, spectral, or ordered local structure; TCNs use causal and dilated convolutions to model long-range dependencies efficiently [37]. Recurrent neural networks and Transformer-based models are relevant for sequential regression problems, with Transformers replacing recurrence by self-attention mechanisms [38].

Graph Neural Networks extend neural regression to data represented as graphs [39]. They are suitable when observations are linked by relational structure, such as molecules, networks, power systems, or spatial interactions. Tabular neural models such as TabNet use attention-based feature selection mechanisms to improve neural modeling of structured datasets [40]. Physics-informed neural networks incorporate physical laws, often expressed as differential equations, directly into the loss function [41], making them relevant when prior scientific knowledge constrains the regression problem.

Neural regression models are most appropriate when nonlinear interactions are complex, sufficient data are available, and predictive flexibility is more important than direct interpretability. They are less suitable as default models for small tabular datasets unless the architecture is justified by the structure of the input data. Their scaling requirements and hyperparameter sensitivity should be explicitly considered in any practical workflow.

3.7. Symbolic and Evolutionary Regression Models

Symbolic regression differs from most black-box regression methods because it seeks an explicit mathematical expression that maps inputs to a continuous output. Genetic Programming regression represents candidate models as symbolic expression trees and evolves them using evolutionary operators [42]. Internal nodes represent functions or operators, while terminal nodes represent variables or constants. The search process typically includes initialization, fitness evaluation, selection, crossover, mutation, and replacement.

The fitness function usually combines prediction error with a complexity penalty. This is important because unconstrained Genetic Programming can generate unnecessarily large expressions, a phenomenon often referred to as bloat. Parsimony pressure, expression simplification, validation-based selection, and constraints on the primitive set are therefore important for practical use. Recent symbolic regression systems often combine evolutionary search with parameter optimization, simplification rules, dimensional constraints, or Pareto-based selection to balance accuracy and complexity [43].

Memetic Genetic Programming extends symbolic regression by combining evolutionary structural search with local numerical optimization of constants. In this setting, the evolutionary component searches over expression structures, while a local optimizer refines continuous parameters within promising expressions. This separation of structural and parametric optimization can improve numerical accuracy and convergence, but it also increases computational cost and may reduce population diversity if local search is applied too aggressively.

Symbolic and memetic regression models are most useful when the analytical form of the resulting model is important. They are especially attractive in scientific and engineering contexts where the goal is not only prediction but also model discovery, interpretability, and hypothesis generation. Their limitations are high computational cost, sensitivity to the choice of primitives and evolutionary parameters, and vulnerability to overfitting noisy data.

3.8. Practical Implications of the Model-Family View

The model-family view clarifies that regression model selection should begin with the modeling objective and data characteristics rather than with algorithm popularity. Linear and regularized models provide transparent baselines and are well-suited to interpretable, low-complexity workflows. Robust and distribution-aware methods should be considered when residual behavior, skewness, heteroscedasticity, or outliers are central. Online methods are appropriate when data arrive sequentially or when memory constraints are important. Tree ensembles are strong candidates for nonlinear tabular data, while kernel and instance-based methods are useful when the similarity structure is meaningful. Neural networks are justified when the data structure and sample size support high-capacity representation learning. Symbolic regression is most relevant when explicit mathematical form is a scientific requirement.

This organization also highlights that preprocessing and evaluation are not independent of model choice. Regularized, kernel-based, distance-based, online, and neural methods are usually scale-sensitive. Robust and distribution-aware models require attention to residual structure and target-variable properties. Ensemble models often reduce preprocessing demands but introduce interpretability and tuning considerations. Therefore, a regression workflow should select the model family, preprocessing strategy, and evaluation metric jointly.

4. Synthesis of Application Domains and Methodological Trends

The reviewed application examples are synthesized here at the domain and methodological level rather than repeated after each individual model description. Their purpose is not to demonstrate that a specific regression method is universally superior in a given field, but to illustrate how model families are selected according to data structure, modeling objective, interpretability requirements, robustness needs, and computational constraints.

Energy, building, and environmental applications show that regression workflows often range from simple interpretable models to more flexible nonlinear methods. Linear regression has been used in building energy modeling [44], polynomial regres-

sion in photovoltaic–thermal system modeling [45], Ridge regression in wind-speed forecasting [46], and RANSAC in wind-power data analysis [47]. Gradient boosting regression trees and Support Vector Regression have also been applied in energy-related prediction tasks [48,49]. These examples indicate that linear and regularized models remain useful when transparency and efficient estimation are important, while robust, kernel-based, and ensemble methods become more relevant when nonlinear relationships, abnormal measurements, or complex tabular interactions are present.

Industrial, manufacturing, and engineering applications further demonstrate the need to balance accuracy, robustness, and operational interpretability. Bayesian Ridge regression has been applied to mixed oil length prediction [50], Passive-Aggressive regression to drug release modeling [51], Extra Trees to aluminum manufacturing [52], Radius Neighbors regression to membrane separation processes [53], and neural-network-based approaches to rubber blend mixing and dynamic-system approximation [54,55]. These cases suggest that engineering data are often nonlinear, correlated, noisy, or collected under changing operating conditions. Consequently, regularized and Bayesian models are useful for stable estimation; instance-based methods may be suitable when local similarity is meaningful, and ensemble or neural models may be justified when nonlinear interactions dominate.

Biomedical, pharmaceutical, and health-related examples emphasize robustness, nonlinear approximation, and interpretability. Huber regression has been used in pharmaceutical solubility estimation [56], ν -SVR in pharmaceutical modeling [57], and decision-tree-based approaches in clinical decision contexts such as orthognathic surgery [58]. These applications illustrate that biomedical regression problems often involve small or moderate datasets, noisy measurements, and a need for interpretable or clinically meaningful outputs. Robust regression and support-vector methods can be useful under such conditions, while decision trees provide transparent rule-based structures, provided that instability and overfitting are controlled through validation or comparison with ensemble alternatives.

Cybersecurity, reliability, and monitoring applications show the relevance of regression models in high-dimensional and operational settings. Elastic Net regression has been used for malicious attack detection [59], SGD-based optimization for intrusion detection system performance evaluation [60], Random Forest regression for power-grid operation under cyber-attacks [61], and KNN-based regression for latent defect identification in transistor testing [62]. These examples highlight the importance of regularization, scalability, incremental learning, nonlinear modeling, and feature-scaling awareness in monitoring-oriented regression workflows.

Distribution-aware modeling appears in domains where the conditional mean is not sufficient to describe the response. Poisson–Tweedie models have been used for cyclist crash frequency estimation [63], while Quantile Regression has been applied to sports analytics involving NBA game outcomes [64]. These examples show that Tweedie-type models are appropriate when skewness, count-like behavior, zero inflation, or mean–variance relationships are relevant, whereas quantile-based models are useful when lower, median, and upper response regimes need to be distinguished.

Remote sensing, geospatial analysis, and scientific discovery emphasize robustness and analytical transparency. Theil–Sen estimation has been used for UAV-based coral mapping under sun glint correction [65], while symbolic regression and Genetic Programming have been explored for recovering or approximating physical laws from data [66]. These examples illustrate that, in some scientific contexts, the objective is not only predictive accuracy but also the recovery of explicit relationships that can be inspected, simplified, or compared with domain theory.

Financial and market-related regression provides an example of scalable nonlinear modeling for structured data. Histogram-based gradient boosting has been adapted to stock

market analysis [67]. This reflects a broader trend in which boosting methods are attractive for large, nonlinear, tabular datasets. However, in financial applications, validation must explicitly account for temporal dependence, non-stationarity, and the risk of overfitting historical patterns.

Across these domains, several methodological trends can be identified. First, simple and regularized linear models remain important because they provide interpretable, computationally efficient, and stable baselines, especially when coefficient-level reasoning is valuable [44,46,59,68]. Second, nonlinear and ensemble methods are frequently selected when tabular predictors contain interactions, thresholds, or complex nonlinear patterns, as shown by applications involving decision trees, Random Forests, Extra Trees, Gradient Boosting, Histogram-Based Gradient Boosting, SVR, KNN, and Radius Neighbors [48,49,52,53,58,61,62,67]. Third, robust, probabilistic, online, and distribution-aware methods should be viewed as responses to specific data conditions rather than generic replacements for standard regression. Bayesian Ridge, SGD, Passive-Aggressive regression, Tweedie regression, Quantile Regression, Huber regression, Theil–Sen estimation, and RANSAC are most relevant when uncertainty, scalability, streaming updates, skewness, heteroscedasticity, or outliers are central modeling concerns [47,50,51,56,60,63–65].

Overall, the application literature supports the central argument of this review: regression models should not be ranked independently of the data and domain. Practical model selection should begin with target-variable properties, sample size, dimensionality, expected nonlinearity, outlier structure, interpretability requirements, and computational constraints. The reviewed examples show that applied regression is not a single algorithmic task, but a workflow design problem in which the model family, preprocessing strategy, validation procedure, and evaluation metric must be selected together. The growing use of predictive analytics across domains such as agriculture, environmental assessment, software engineering, and symbolic reasoning calls for systematic guidance on model selection, preprocessing, and evaluation. Recent studies demonstrate that model effectiveness depends strongly on data characteristics and analytical objectives [69–71].

5. Decision Dimensions for Model Selection

The previous sections reviewed regression model families and synthesized their application domains. This section translates these methodological and empirical observations into a practical decision framework for selecting regression models in continuous function approximation. The purpose is not to rank algorithms universally, but to describe how researchers can construct a transparent, reproducible, and objective-aligned modeling workflow.

Regression model selection should not be reduced to choosing the algorithm with the best score on a single metric. A model that performs well according to one criterion may still be unsuitable if it is difficult to interpret, unstable under resampling, sensitive to preprocessing, inconsistent with the target distribution, computationally infeasible, or poorly aligned with the scientific objective. Model selection should therefore be treated as a workflow design problem in which the modeling objective, data characteristics, preprocessing strategy, validation procedure, and evaluation metrics are considered jointly.

The decision dimensions discussed in this section should be interpreted as diagnostic axes rather than fixed rankings. Properties such as interpretability, robustness, scalability, uncertainty support, and tuning complexity are not absolute characteristics of a model family. They depend on dataset size, dimensionality, noise structure, hyperparameter settings, implementation choices, and the way the model is embedded in the complete workflow.

5.1. Modeling Objective and Scientific Role of the Model

A regression workflow should begin with the role that the model is expected to play. In some studies, the main objective is explanation: the researcher wants to understand how input variables are associated with the continuous response. In other cases, the primary goal is prediction, and interpretability may be secondary. A third situation is uncertainty-aware decision-making, where prediction intervals, conditional quantiles, or asymmetric errors are more important than a single point estimate. A fourth objective is model discovery, where the aim is to identify an explicit functional relationship that can be inspected, simplified, or compared with domain theory.

When explanation is the main goal, linear regression, regularized linear models, generalized linear or additive formulations, simple decision trees, and symbolic regression may be appropriate candidates. These models can provide coefficient-level, rule-based, or equation-based interpretation, although such interpretation remains valid only under appropriate assumptions and sufficient data quality. A transparent model is not automatically scientifically meaningful if its assumptions are violated or if the data do not support the intended interpretation.

When predictive accuracy is the primary objective, more flexible nonlinear models may be considered. Random Forests, Extra Trees, Gradient Boosting, Histogram-Based Gradient Boosting, Support Vector Regression, neural networks, and modern boosting frameworks can represent nonlinearities and interactions more effectively than simple linear models. However, their additional flexibility should be justified by validation results, not by algorithmic popularity. If a complex model improves only marginally over a simpler baseline, the gain may not justify the loss of transparency, increased tuning effort, or higher computational cost.

When uncertainty or distributional behavior is central, the workflow should include models or extensions that provide more than point predictions. Bayesian Ridge, Gaussian Process Regression, Quantile Regression, Tweedie Regression, probabilistic neural approaches, and ensemble-based uncertainty approximations may be relevant, depending on the target distribution and the type of uncertainty required. In such cases, uncertainty estimates should be evaluated explicitly, because unvalidated intervals or quantiles can create a misleading impression of reliability.

When the purpose is model discovery, symbolic regression and memetic Genetic Programming are especially relevant because they can produce explicit analytical expressions. These methods are attractive in scientific and engineering contexts where the final model should support hypothesis generation or comparison with known physical laws. Nevertheless, symbolic models require strict validation, because flexible symbolic search can overfit noisy data or produce expressions that are accurate but unnecessarily complex or physically implausible.

5.2. Target Variable, Input Features, and Data Structure

After the modeling objective has been clarified, the dataset should be examined before candidate models are selected. The target variable should be inspected for scale, skewness, boundedness, zero or near-zero values, heavy tails, outliers, and dynamic range. These properties influence both model choice and metric selection. If the response is approximately symmetric and continuous, least-squares-based methods may be suitable starting points. If the target is strictly positive and strongly skewed, transformations such as logarithmic, Box–Cox, or Yeo–Johnson transformations may improve model behavior. Alternatively, distribution-aware models such as Tweedie Regression may be more appropriate when the mean–variance relationship is part of the data-generating process.

Targets with many zero or near-zero values require special attention. Percentage-based metrics such as MAPE can become unstable or misleading in such cases, and relative errors may no longer reflect the scientific importance of prediction errors. Absolute, squared, scaled, or distribution-aware losses may be more appropriate. If the scientific interest concerns lower or upper outcomes rather than the conditional mean, Quantile Regression or quantile-based gradient boosting may be preferable to mean-based regression.

The dynamic range of the target variable also affects the workflow. Neural networks, kernel-based methods, and gradient-based optimization procedures can be sensitive to very large target magnitudes because loss gradients may become unstable. Output scaling or target transformation can improve numerical behavior, but the inverse transformation must be handled carefully during evaluation. A model trained on a transformed target should normally be assessed on the original scale as well, because error behavior on the transformed scale may not correspond to practical error behavior in the application domain.

Input features should be examined for scale differences, missing values, categorical variables, multicollinearity, irrelevant predictors, outliers, and measurement errors. Scale differences are especially important for regularized linear models, online learning methods, kernel methods, distance-based methods, and neural networks. In these models, feature magnitudes influence penalties, gradients, distances, or kernel values. Missing values, feature selection, target transformations, categorical encoding, and scaling should be performed inside the validation pipeline rather than before data splitting, because otherwise information from the validation or test data may leak into training.

Sample size and dimensionality further constrain model choice. For small datasets, simple and regularized models are often preferable because they reduce variance and are easier to validate. Ridge, Lasso, Elastic Net, Bayesian Ridge, Support Vector Regression, Gaussian Process Regression, and carefully constrained tree models can be reasonable candidates, but flexible models can overfit when the number of observations is limited. In high-dimensional settings, regularization, feature selection, dimensionality reduction, or strong structural assumptions become important. Lasso can produce sparse models, Elastic Net can handle groups of correlated predictors, and Ridge can stabilize coefficient estimates under multicollinearity. For very large datasets, scalability becomes a dominant criterion, making stochastic optimization, Passive-Aggressive regression, Histogram-Based Gradient Boosting, Extra Trees, and other efficient ensemble or online methods more attractive.

The relationship between sample size and number of predictors is particularly important. When the number of predictors is large relative to the number of observations, models with explicit regularization or dimensionality control should be preferred. When the sample size is large and the feature space is moderate, more flexible nonlinear models can be considered. Exact kernel methods, Gaussian Process Regression, KNN, Radius Neighbors, and some symbolic regression approaches may become computationally expensive as sample size increases unless approximate, sparse, or distributed variants are used.

5.3. Functional Form, Robustness, and Approximation Behavior

The expected functional relationship between inputs and outputs should guide the selection of candidate model families. Some regression problems are approximately linear, while others involve curvature, thresholds, interactions, local effects, discontinuities, sequential dependence, graph structure, or physical constraints. The practical question is not simply whether a model is linear or nonlinear, but what kind of structure the data are expected to contain.

When the relationship is approximately additive and linear, ordinary least squares and regularized linear models are natural baselines. If mild curvature is expected, polynomial, spline-based, or additive formulations can extend linear modeling while preserving some

interpretability. Polynomial regression can capture smooth nonlinear trends, but the number of terms grows rapidly with polynomial degree and feature count, increasing the risk of overfitting.

When nonlinear interactions, thresholds, or heterogeneous feature effects are expected, tree-based methods and ensembles are often effective. Single decision trees provide rule-based interpretation, but they are unstable and prone to overfitting. Random Forests and Extra Trees reduce variance through aggregation, while boosting methods sequentially correct residual structure and often achieve strong predictive performance on structured tabular data. These models are usually less sensitive to monotone transformations of individual numerical features, but they still require careful validation, hyperparameter control, and interpretation.

Kernel-based and instance-based models are useful when similarity in feature space is meaningful. Support Vector Regression and Gaussian Process Regression can model smooth nonlinear relationships through kernels, while KNN and Radius Neighbors rely on local averaging. Their behavior depends strongly on scaling, kernel choice, distance metric, neighborhood size, and local data density. These methods may perform well on small-to-medium datasets, but computational cost and prediction time can become limiting for large datasets.

Neural networks are appropriate when nonlinear interactions are complex, sufficient data are available, and the input structure justifies high-capacity representation learning. Feedforward networks can model general nonlinear functions, while convolutional, temporal, recurrent, attention-based, graph-based, tabular-attention, or physics-informed architectures may be justified when the data contain spatial, temporal, sequential, relational, structured-tabular, or physical information. Without such justification, neural networks may be unnecessarily complex for small or moderate tabular datasets.

Robustness should be treated as a workflow requirement rather than a binary model property. Real-world regression data often contain measurement noise, sensor faults, data-entry errors, rare events, leverage points, or heavy-tailed residuals. Standard least-squares models are sensitive to large residuals because squared loss penalizes large errors disproportionately. Ridge and Lasso reduce coefficient variance but do not inherently make the loss robust to outliers. Huber Regression can reduce the influence of vertical outliers, while Quantile Regression can describe different parts of the conditional response distribution. Theil–Sen and RANSAC may be useful under stronger contamination, but their suitability depends on dimensionality, threshold choices, sampling strategy, and the assumed inlier structure.

Tree ensembles often show empirical robustness through aggregation, but they are not automatically robust to all forms of noise. Boosting methods may focus heavily on difficult or noisy observations if regularization and loss functions are not chosen carefully. KNN and Radius Neighbors can be strongly affected by local noise and irrelevant features. Neural networks can overfit noise unless regularization, early stopping, and validation are used. Therefore, the relevant question is not whether a model is robust in general, but whether the workflow handles the specific type of noise or outlier structure present in the data.

5.4. Preprocessing, Metrics, and Validation as Part of Model Selection

Preprocessing should be designed together with the candidate models. A comparison between an unscaled scale-sensitive model and a tree ensemble is not a fair comparison of model families; it is a comparison of incomplete workflows. Regularized linear models should usually include feature standardization because penalties act on coefficient magnitudes. SGD-based models and neural networks usually benefit from input scaling because

it improves numerical conditioning and convergence. Kernel and distance-based methods require careful scaling because feature magnitudes directly define similarity. Tree-based models are usually less sensitive to monotone transformations of individual numerical variables, but missing values, categorical encoding, extreme outliers, and implementation-specific behavior still need to be handled explicitly.

Metric selection should also be part of the workflow design rather than an afterthought. Squared-error metrics such as MSE and RMSE are appropriate when large errors should be penalized strongly. MAE is easier to interpret in the original target units and is less dominated by extreme residuals. Percentage-based metrics can support relative interpretation, but they are unreliable when targets are zero or close to zero. R^2 and adjusted R^2 describe explained variance and can be useful for comparison, but they should not be the only evidence of predictive usefulness. Huber loss is relevant when moderate robustness is needed, while quantile loss is appropriate when asymmetric errors or conditional quantiles are part of the modeling objective.

The selected metrics should reflect the scientific or practical cost of prediction errors. A model intended for operational prediction should not be judged only by average error if rare but large errors are critical. A model intended for scientific explanation should not be evaluated only by predictive accuracy if the interpretation of parameters or discovered relationships is central. A model intended to support uncertainty-aware decisions should be evaluated not only by point-prediction error, but also by calibration, interval coverage, or quantile behavior.

Validation should evaluate complete workflows rather than isolated algorithms. Scaling parameters, imputers, feature selection procedures, target transformations, hyperparameter tuning, and model fitting should be applied within the training portion of each validation split. The validation or test data should not influence preprocessing or model selection decisions. This is especially important when comparing models with different preprocessing requirements, because leakage can make performance estimates overly optimistic.

The validation design should reflect the structure of the data. Random cross-validation may be appropriate when observations are independent and identically distributed. Temporal data require time-aware splits to avoid training on future information. Grouped data, such as repeated measurements from the same subject, machine, location, or experiment, require group-aware validation so that observations from the same group do not appear in both training and testing sets. If the data contain hierarchical, spatial, temporal, or experimental structure, the validation design should respect that structure.

5.5. Baseline Models, Workflow Comparison, and Model Justification

A practical regression workflow should include at least one simple baseline model. The baseline provides a transparent reference point for judging whether more complex models are actually necessary. In many applications, ordinary least squares, Ridge regression, Lasso, Elastic Net, or another regularized linear model can serve as a starting point. For nonlinear problems, a shallow decision tree or a simple tree ensemble may provide an interpretable nonlinear baseline.

The purpose of the baseline is not necessarily to achieve the best performance. Its role is to create a reproducible comparison. If a complex model improves only slightly over a simple baseline, the researcher should consider whether the improvement justifies the additional tuning burden, reduced interpretability, higher computational cost, or lower transparency. Conversely, if the baseline performs poorly and residual diagnostics reveal systematic nonlinear structure, more flexible models may be justified.

After the baseline is established, the researcher should compare a small set of candidate model families that represent meaningfully different assumptions and approximation strategies. The goal is not to test every available algorithm, but to evaluate models that are plausible given the modeling objective and data conditions. A reasonable comparison may include an interpretable baseline, a regularized or robust alternative, a nonlinear ensemble method, and, where justified, a kernel-based, neural, probabilistic, or symbolic model.

Model comparison should consider multiple criteria. Predictive performance is important, but it should be assessed together with validation stability, interpretability, robustness, scalability, computational cost, tuning complexity, uncertainty support, and reproducibility. A boosting model may provide the lowest RMSE, but a Ridge or Elastic Net model may be preferable if the performance difference is small and interpretability is essential. A neural network may be justified if it substantially improves performance on a large and structurally complex dataset, but it may be inappropriate when simpler models perform similarly on a small tabular dataset.

The bias–variance trade-off provides a useful conceptual lens for interpreting these comparisons. Simple linear models may underfit complex relationships but often have low variance and strong transparency. Polynomial models, trees, ensembles, kernels, nearest-neighbor methods, neural networks, and symbolic regression increase flexibility in different ways, but they also introduce additional risks of overfitting, instability, or computational burden. Regularization, aggregation, margin control, neighborhood size, architecture design, early stopping, and parsimony pressure are all mechanisms for managing this trade-off. In applied research, the most reliable model is not necessarily the one with the lowest training error, but the one whose validation performance is stable and whose assumptions match the data and objective.

Sensitivity analysis should accompany model comparison. If rankings change substantially under different validation splits, preprocessing choices, hyperparameter ranges, or evaluation metrics, the results should be interpreted cautiously. Such instability may indicate that the dataset is too small, the models are too flexible, the preprocessing is influential, or the chosen metric does not adequately represent the modeling objective. Stability across reasonable workflow choices is an important indicator of reliability.

5.6. Interpretation, Reporting, and Limitations

After model selection, the final model should be interpreted using methods appropriate to its model family and the research objective. For linear models, coefficients, standardized effects, confidence intervals, and residual diagnostics may be relevant. For regularized models, the stability of selected variables should be examined. For decision trees, split rules and decision paths can support interpretation. For tree ensembles and other complex models, feature importance, partial dependence, accumulated local effects, SHAP-type explanations, or other post hoc tools may provide useful summaries, but they should be presented as approximations rather than definitive causal explanations.

Residual analysis remains important regardless of the model family. Residual plots can reveal heteroscedasticity, systematic bias, nonlinear patterns, outliers, or regions of poor performance. Prediction errors should also be examined across relevant subgroups, operating conditions, or target ranges. A model that performs well on average may still fail in scientifically or practically important regions. If uncertainty estimates are provided, they should be checked for calibration and coverage, because unreliable uncertainty estimates can be more misleading than point predictions alone.

The final workflow should be reported transparently. A regression study should describe the modeling objective, dataset characteristics, preprocessing steps, validation design, model families compared, hyperparameter search space, evaluation metrics, soft-

ware environment, and random seeds where applicable. The selected model should be justified using the decision dimensions of the study rather than only a performance table. For example, the chosen model may be preferred because it provides an acceptable balance between accuracy and interpretability, because it handles outliers more reliably, because it scales to the available dataset, or because it supports uncertainty estimation required by the application.

Limitations should be stated explicitly. These may include small sample size, restricted validation data, sensitivity to hyperparameters, limited interpretability, lack of external validation, unvalidated uncertainty estimates, or uncertain generalization to new domains. Transparent reporting helps prevent overstatement of model performance and supports responsible use of regression models in applied continuous function approximation.

Overall, the proposed decision framework emphasizes that regression model selection is not an isolated algorithmic choice. It is a structured workflow in which the modeling objective, target-variable properties, input-feature characteristics, expected functional form, robustness requirements, preprocessing pipeline, validation design, evaluation metrics, interpretability needs, and computational constraints must be aligned. This perspective supports reproducible and scientifically grounded regression workflows while leaving final model choice to empirical validation on the user's own data. For an overview, please consider Table 1.

Table 1. Summary of the regression model families.

Model Family	Best Aligned with	Main Selection Cautions
Linear, polynomial, and regularized models OLS, Polynomial, Lasso, Ridge, Elastic Net, Bayesian Ridge	Transparent baseline modeling, coefficient-level interpretation, low-complexity workflows, and high-dimensional settings where regularization is useful.	Limited flexibility unless nonlinear features are introduced; regularized and polynomial models are scale-sensitive and require validation of feature choice and penalty settings.
Online linear methods SGDRegressor, Passive-Aggressive	Large-scale, streaming, or incrementally updated regression tasks where memory use and update speed are important.	Mostly linear unless features are engineered; sensitive to feature scaling, learning settings, stopping criteria, and tolerance parameters.
Robust and distribution-aware models Tweedie, Quantile, Huber, Theil–Sen, RANSAC	Skewed, heteroscedastic, zero-inflated, asymmetric, or outlier-contaminated response behavior.	Require suitable distributional, quantile, threshold, or inlier assumptions; robustness should be checked with residual diagnostics and validation stability.
Tree-based and ensemble models Decision Tree, Random Forest, Extra Trees, Gradient Boosting, HistGradientBoosting	Nonlinear tabular data, interactions, threshold effects, and cases where extensive feature engineering should be avoided.	Single trees may be unstable; ensembles reduce variance but increase tuning burden and reduce direct interpretability; overfitting must be controlled.
Kernel-based models SVR, ν -SVR	Small-to-medium nonlinear problems where margin-based fitting and kernel similarity are appropriate.	Strongly affected by scaling, kernel choice, and hyperparameters; computational cost can increase with dataset size and number of support vectors.
Instance-based models KNN, Radius Neighbors	Local similarity-based regression where nearby observations are expected to have similar target values.	Sensitive to feature scaling, distance metric, dimensionality, neighborhood size or radius, sparse regions, and prediction-time cost.

Table 1. Cont.

Model Family	Best Aligned with	Main Selection Cautions
Neural-network models MLP, DNN, structured neural regressors	Complex nonlinear or structured high-dimensional inputs when sufficient data and tuning resources are available.	Require careful input and often output scaling; hyperparameter-sensitive, data-hungry, less directly interpretable, and prone to overfitting without regularization.
Symbolic and evolutionary models Genetic Programming, Memetic GP	Model discovery, explicit analytical expressions, hypothesis generation, and comparison with domain theory.	Computationally expensive; sensitive to primitive set, parsimony pressure, local search settings, and validation against noisy or physically implausible expressions.

Note: The table summarizes diagnostic tendencies rather than fixed rankings. Model suitability depends on the modeling objective, dataset size, dimensionality, noise and outlier structure, preprocessing choices, validation design, implementation details, and evaluation metrics.

6. Data Scaling Recommendations

Feature scaling is an important preprocessing step when regression algorithms are sensitive to the magnitude, range, or distribution of the input variables. Although tree-based models are usually less affected by monotone transformations, regularized linear models, kernel methods, distance-based regressors, and neural networks often require careful scaling for stable optimization and meaningful parameter penalization.

This section distinguishes between input scaling, applied to predictors, and output scaling, applied to the target variable. It reviews standardization, min–max rescaling, robust scaling based on medians and interquartile ranges, and power or quantile transformations for skewed or heavy-tailed data. The aim is to relate each preprocessing choice to the model families for which it is most relevant (see Table 2).

Table 2. Data scaling method recommendations for regression models.

Regressor Model	Input Scaler	Output Scaler
Linear Regression	Std, MaxAbs ^a	NR ^b
Polynomial Regression	Poly + Std, MaxAbs ^a	NR ^b
Lasso/Ridge/Elastic Net	Std, MaxAbs ^a	NR ^b
Bayesian Ridge	Std, MaxAbs ^a	NR ^b /Std ^c
SGD/Passive-Aggressive	Std, MaxAbs ^a	Std ^c
Tweedie	Rob, Power ^c	Rob, Power ^c
Quantile/Huber	Rob	Rob, Power ^c
Theil–Sen/RANSAC	Rob	NR ^b
Decision Tree/Random Forest/Extra Trees	NR ^b	NR ^b
Gradient Boosting/HistGB/XGBoost/LightGBM/CatBoost	NR ^b /Quantile ^c	NR ^b
SVR/ ν -SVR	Std, Power ^c , Quantile ^c , MaxAbs ^a , Norm ^d	Std, Power ^c
Gaussian Process Regression	Std, Power ^c /Quantile ^c	Std/Power ^c
KNN/Radius Neighbors	Std, Rob, MinMax, MaxAbs ^a , Norm ^d	NR ^b
MultiLayer Perceptron	Std, Power ^c	Std, Power ^c
Deep Neural Network	Std, Rob, Power ^c , Quantile ^c	Std, Rob, Power ^c
Genetic Programming/Memetic GP	Std, Rob, Power ^c	Std, Rob, Power ^c

Abbreviations: Std = StandardScaler; Rob = RobustScaler; MinMax = MinMaxScaler; MaxAbs = MaxAbsScaler; Quantile = QuantileTransformer; Power = PowerTransformer; Norm = Normalizer; Poly = PolynomialFeatures; NR = normally not required; HistGB = Histogram-Based Gradient Boosting. Notes: ^a Prefer MaxAbs for sparse or high-dimensional inputs. ^b NR means normally not required, not forbidden. ^c Use only when justified by skewness, heavy tails, large dynamic range, or validation results. ^d Use Normalizer only when angular or compositional similarity is meaningful.

6.1. Input and Output Scaling

Input scaling is used to normalize or standardize predictors so that they have comparable ranges or distributions. This is important for models that rely on gradient descent, regularization, distance calculations, or kernel evaluations, because large differences in feature magnitude can produce uneven learning, slow convergence, or unstable solutions.

Output scaling transforms the target variable. It is mainly used to stabilize learning in models that are sensitive to the magnitude of prediction errors, including neural networks and some kernel-based regressors. Output scaling is particularly useful when target values have a large dynamic range or a strongly skewed distribution. After prediction, the inverse transformation should be applied so that model performance and interpretation can be assessed on the original target scale.

6.2. Standard Scaling

Standard Scaler centers each input feature and scales it to unit variance. It is commonly used with linear, regularized, support-vector, and gradient-based models because it improves numerical conditioning and makes coefficient penalties more comparable across predictors. The transformation does not require Gaussian feature distributions, but it is sensitive to outliers because it relies on the mean and standard deviation. For neural networks and other scale-sensitive models, standardizing the target variable can also stabilize loss magnitudes and gradient updates.

6.3. Min–Max Scaling

Min–Max Scaler rescales data to a bounded range (e.g., $[0, 1]$), preserving the feature's relative distribution. Min–max scaling can be useful for bounded or saturation-prone activation functions such as sigmoid or tanh. For ReLU networks, scaling may still help optimization, but ReLU itself is not bounded. For outputs, min–max scaling is most appropriate when the target values are naturally bounded or the model expects outputs within a fixed range (e.g., image pixel intensities in deep learning). Output min–max scaling is also appropriate when the modeling pipeline explicitly includes the inverse transformation. For unbounded regression problems, however, this scaling may cause loss of distortion or poor extrapolation after inverse transformation.

6.4. Robust Scaling

Robust Scaler uses the median and interquartile range (IQR), making it highly resistant to outliers and skewed distributions. This is beneficial for both inputs and outputs when the dataset contains anomalies or heavy tails. Scaling targets using this method is particularly effective when training models on financial, biomedical, or real-world data where extreme values might otherwise dominate the loss function.

6.5. Power Transformer

The Power Transformer (Box–Cox or Yeo–Johnson) helps transform skewed distributions into more Gaussian-like shapes; Box–Cox requires strictly positive data, whereas Yeo–Johnson can handle zero and negative values. It is useful for inputs when features exhibit skew and improves model assumptions for linear or kernel regressors. When applied to outputs, these transformations can make exponential or multiplicative relationships easier to model and can improve numerical stability. Examples include log sales, incidence rates, or other strictly positive targets with strong skewness, provided that inverse-transformed predictions are evaluated on the original scale.

6.6. Quantile Transformer

The Quantile Transformer maps features or targets to follow a uniform or normal distribution using empirical quantiles. While powerful for reshaping marginal feature distributions, especially for methods sensitive to feature distributions or distances, it is generally less essential for tree-based models, which are mostly insensitive to monotonic feature scaling. It can distort the actual numeric relationships if the inverse transformation is imperfect, especially at distribution tails.

6.7. Max-Abs Scaling

Max-Abs Scaler rescales each feature by dividing it by its maximum absolute value, mapping values into the interval $[-1, 1]$ while preserving the sign of the original data. This method is especially useful for sparse or high-dimensional input data, because it does not subtract the mean and therefore preserves sparsity. It may be beneficial for linear, regularized, online, kernel-based, and distance-based regression models when feature magnitudes differ substantially. However, since the scaling factor depends on the maximum absolute value, it remains sensitive to extreme observations. For output scaling, MaxAbsScaler is less commonly used, but it may be appropriate when the target variable has a meaningful symmetric range around zero.

6.8. Vector Normalization

Vector normalization rescales each input sample rather than each feature, usually to unit L_2 -norm. It is useful when the direction or relative composition of a feature vector is more important than its absolute magnitude, such as in sparse, frequency-based, or text-like representations. In regression workflows, it can support distance-based, kernel-based, and linear models when angular or cosine-like similarity is meaningful. However, it should not be used as a general replacement for feature-wise scaling, because it removes magnitude information from each sample. For standard scalar regression targets, vector normalization is generally not recommended as an output transformation.

7. Evaluation Metrics and Diagnostics

Evaluation metrics are essential components of regression model selection because they determine how prediction errors are quantified, compared, and interpreted. Different metrics emphasize different aspects of error behavior. Squared-error metrics penalize large residuals strongly, absolute-error metrics provide more robust average error estimates, percentage-based metrics support relative interpretation under suitable conditions, and quantile-based losses allow asymmetric error evaluation. Therefore, the choice of metric should be aligned with the target-variable properties, residual structure, modeling objective, and practical consequences of prediction errors.

No single metric is universally appropriate for all regression problems. A model that performs well according to RMSE may not be preferable if the application requires robustness to outliers, relative error interpretation, uncertainty-aware prediction, or balanced performance across different target ranges. For this reason, evaluation metrics should be selected before final model comparison and interpreted together with residual diagnostics, validation stability, and domain-specific requirements.

7.1. Squared and Absolute Error Metrics

The most widely used regression metrics are based on residuals $r_i = y_i - \hat{y}_i$, where y_i is the observed value and $\hat{y}_i$ is the predicted value. Mean Squared Error (MSE) measures the average squared residual:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (1)$$

MSE is smooth and well-suited for optimization, but it is highly sensitive to large errors because residuals are squared. It is useful when large deviations should be penalized strongly or when Gaussian error assumptions are appropriate.

Root Mean Squared Error (RMSE) is the square root of MSE:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (2)$$

RMSE preserves the strong penalty on large errors while returning the result to the original unit of the target variable. This makes it easier to interpret than MSE in applied contexts, although it remains sensitive to outliers.

Mean Absolute Error (MAE) measures the average absolute residual:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (3)$$

MAE is less sensitive to extreme residuals than MSE or RMSE and is directly interpretable in the original target units. It is appropriate when all deviations should be penalized approximately equally. However, because the absolute value is non-differentiable at zero, MAE can be less convenient for some gradient-based optimization procedures.

Taken together, MSE and RMSE are preferable when large errors are especially undesirable, while MAE is preferable when robustness and direct interpretability of average error are more important.

7.2. Relative and Explanatory Metrics

Mean Absolute Percentage Error (MAPE) expresses prediction error as a percentage of the observed value:

$$MAPE = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (4)$$

MAPE can be useful when the target variable is strictly positive and relative error interpretation is important. However, it becomes unstable or undefined when $y_i = 0$ and can be misleading when target values are close to zero. Therefore, MAPE should not be used when the response contains zero or near-zero values unless an appropriate modification is introduced and justified.

The coefficient of determination, R^2 , measures the proportion of variance in the target variable explained by the model:

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2} \quad (5)$$

where $\bar{y}$ is the mean of the observed target values. R^2 is useful for describing explanatory fit relative to a mean baseline, but it should not be interpreted as a complete measure of predictive usefulness. A high R^2 does not guarantee good generalization, robustness, or appropriate residual behavior.

Adjusted R^2 modifies R^2 by accounting for the number of predictors:

$$\hat{R}^2 = 1 - \left(1 - R^2\right) \frac{n-1}{n-p-1} \quad (6)$$

where p is the number of predictors. It is particularly useful when comparing models with different numbers of input variables, because it penalizes unnecessary model complexity. However, like standard R^2 , it should be interpreted together with validation results and residual diagnostics.

7.3. Robust and Asymmetric Loss Functions

Huber loss combines the local sensitivity of squared error with the robustness of absolute error. For a residual $r = y - \hat{y}$, it is defined as:

$$L_{\delta}(r) = \begin{cases} \frac{1}{2} \cdot r^2 & \text{if } r \leq \delta \\ \delta \left(|r| - \frac{1}{2} \delta \right) & \text{if } r > \delta \end{cases} \quad (7)$$

For small residuals, Huber loss behaves like squared-error loss and provides smooth gradients. For large residuals, it becomes linear and reduces the influence of outliers. It is therefore appropriate when most residuals are expected to be small, but occasional large deviations should not dominate the evaluation or training objective.

Quantile loss, also known as pinball loss, is used to estimate conditional quantiles rather than the conditional mean. For the quantile level $\tau \in (0, 1)$, it is defined as:

$$L_{\tau}(r) = \begin{cases} \tau \cdot r & \text{if } r \geq 0 \\ (\tau - 1) \cdot r & \text{if } r < 0 \end{cases} \quad (8)$$

or equivalently:

$$L_{\tau}(r) = \max(\tau r, (\tau - 1)r) \quad (9)$$

For $\tau = 0.5$, quantile loss estimates the conditional median and corresponds to minimizing absolute error. For other quantile levels, it penalizes underestimation and overestimation asymmetrically. This makes it useful when lower or upper parts of the conditional response distribution are more important than the mean, or when different practical consequences are associated with under- and over-prediction.

7.4. Metric Selection Based on Data and Modeling Requirements

Metric selection should be guided by the target distribution, residual behavior, and application objective. When the target variable is approximately symmetric and large errors should be penalized strongly, MSE or RMSE are appropriate. When the response is skewed, heavy-tailed, or affected by outliers, MAE, Huber loss, or Quantile loss may provide more stable and interpretable assessments. When the target is strictly positive and the relative error is meaningful, MAPE can be used, but it should be avoided when zero or near-zero values occur. When explanatory fit or model complexity is important, R^2 and adjusted R^2 can be informative, but they should not replace validation-based error metrics.

The practical cost of prediction errors should also influence metric choice. RMSE is suitable when rare but large errors are critical, because it penalizes large residuals more strongly. MAE is more appropriate when deviations of different magnitudes should contribute more uniformly to the evaluation. Quantile loss is suitable when asymmetric risks are present, for example, when under-prediction is more costly than over-prediction, or when upper and lower predictive bounds are required.

In applied regression studies, it is often useful to report more than one metric. A combination of RMSE and MAE can distinguish large-error sensitivity from average absolute deviation. Adding R^2 or adjusted R^2 can help describe explanatory fit, while Huber or Quantile loss can support robustness or risk-aware evaluation. However, the primary metric should be selected before final model comparison and justified according to the modeling objective.

Residual diagnostics should accompany numerical metrics. Residual plots can reveal systematic bias, heteroscedasticity, nonlinear structure, outliers, or poor performance in specific target ranges. A model with a favorable average error may still be unsuitable if residuals show systematic patterns or if errors are concentrated in scientifically important regions. Therefore, regression evaluation should combine numerical metrics with diagnostic analysis and validation stability (see Table 3).

Table 3. Summary table of evaluation metrics.

Metric	Sensitivity to Outliers	Scale-Dependent	Relative Error	Interpretation	Optimization Suitability
MSE	High	Yes	No	Moderate	Excellent (smooth)
RMSE	High	Yes	No	Easy	Excellent
MAE	Low	Yes	No	Easy	Moderate (non-diff at 0)
MAPE	High (low y)	No	Yes	Easy	Poor for $y = 0$ cases
R^2	Moderate	No	No	Easy	Good (interpretable)
Adj. R^2	Moderate	No	No	Easy	Good for model selection
Huber	Low	Yes	No	Easy	Excellent (smooth, convex)
Quantile/Pinball	Low	Yes	No	Moderate	Moderate (non-diff at 0)

Overall, evaluation metrics should not be treated as neutral reporting tools. They encode assumptions about which errors matter, how residuals should be penalized, and what kind of model behavior is considered desirable. A transparent regression workflow should therefore justify metric choice explicitly and interpret performance results in relation to the data, the validation design, and the scientific or practical objective.

8. Conclusions

This review synthesized regression models for continuous function approximation from a workflow-oriented perspective. The central finding is that no model family is universally optimal: suitability depends on the modeling objective, target-variable properties, feature structure, sample size, noise and outlier patterns, interpretability requirements, computational constraints, and validation design.

Linear and regularized models remain valuable transparent baselines, while robust and distribution-aware methods are useful when residual behavior, skewness, heteroscedasticity, or outliers are central. Tree ensembles are strong candidates for nonlinear tabular data, whereas kernel-based, instance-based, neural, probabilistic, and symbolic approaches should be selected when their assumptions match the data structure and research objective.

The proposed framework therefore recommends evaluating complete workflows rather than isolated algorithms. Model family, preprocessing strategy, validation procedure, and evaluation metrics should be chosen jointly, and final selection should be justified through empirical validation, diagnostic checks, and domain knowledge. Under this perspective, regression modeling becomes a structured methodological design problem aimed at producing accurate, reproducible, interpretable, and practically feasible continuous-function approximations.

Author Contributions: T.S.: Conceptualization, methodology, writing—review and editing. G.G.: Writing—original draft, visualization, data curation. Z.E.: Conceptualization, supervision, writing—review and editing, funding acquisition. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially funded by the Hungarian National Office for Research, Development and Innovation, grant number 2024-1.1.1-KKV_FÓKUSZ-2024-00048.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created or analyzed in this study.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ML	Machine Learning
OLS	Ordinary Least Squares
BLUE	Best Linear Unbiased Estimator
LASSO	Least Absolute Shrinkage and Selection Operator
SGD	Stochastic Gradient Descent
PA(R)	Passive-Aggressive (Regression)
UAV	Unmanned Aerial Vehicle
RANSAC	Random Sample Consensus
SVR	Support Vector Regression
KNN	k-Nearest Neighbors
RNR	Radius Neighbors Regression
GLM	Generalized Linear Model (Tweedie family)
ReLU	Rectified Linear Unit
IQR	Interquartile Range
MSE	Mean Squared Error
RMSE	Root Mean Squared Error
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error

References

1. Kariya, T.; Kurata, H. *Generalized Least Squares*; Wiley: Hoboken, NJ, USA, 2004.
2. Hayashi, F. *Econometrics*; Princeton University Press: Princeton, NJ, USA, 2001.
3. Montgomery, D.C.; Peck, E.A.; Vining, G.G. *Introduction to Linear Regression Analysis*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2021.
4. Tibshirani, R. Regression Shrinkage and Selection Via the Lasso. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **1996**, *58*, 267–288. [CrossRef]
5. Hoerl, A.E.; Kennard, R.W. Ridge Regression—1980: Advances, Algorithms, and Applications. *Am. J. Math. Manag. Sci.* **1981**, *1*, 5–83. [CrossRef]
6. Hilt, D.E.; Seegrist, D.W. *Ridge, a Computer Program for Calculating Ridge Regression Estimates*; Biodiversity Heritage Library: Washington, DC, USA, 1977. [CrossRef]
7. Zou, H.; Hastie, T. Regularization and Variable Selection Via the Elastic Net. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **2005**, *67*, 301–320. [CrossRef]
8. Yang, Y.; Yang, Y. Hybrid Prediction Method for Wind Speed Combining Ensemble Empirical Mode Decomposition and Bayesian Ridge Regression. *IEEE Access* **2020**, *8*, 71206–71218. [CrossRef]
9. Amari, S. Backpropagation and stochastic gradient descent method. *Neurocomputing* **1993**, *5*, 185–196. [CrossRef]
10. Crammer, K.; Dekel, O.; Keshet, J.; Shalev-Shwartz, S.; Singer, Y. Online Passive-Aggressive Algorithms. *J. Mach. Learn. Res.* **2006**, *7*, 551–585.
11. Shi, R.; Ying, J.; Palomar, D.P. Adaptive passive-aggressive framework for online regression with side information. *Adv. Neural Inf. Process. Syst.* **2024**, *37*, 125073–125097. Available online: https://proceedings.neurips.cc/paper_files/paper/2024/file/e2393d306d44fdec33a8100e31c3dc23-Paper-Conference.pdf (accessed on 1 April 2026). [CrossRef]
12. Jørgensen, B. Exponential Dispersion Models. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **1987**, *49*, 127–145. [CrossRef]
13. Koenker, R. *Quantile Regression*; Cambridge University Press: Cambridge, UK, 2005.
14. Huber, P.J. Robust Estimation of a Location Parameter. *Ann. Math. Stat.* **1964**, *35*, 73–101. [CrossRef]
15. Theil, H. A rank-invariant method of linear and polynomial regression analysis. *Indag. Math.* **1950**, *12*, 173.
16. Sen, P.K. Estimates of the regression coefficient based on Kendall's tau. *J. Am. Stat. Assoc.* **1968**, *63*, 1379–1389. [CrossRef]
17. Rousseeuw, P.J.; Leroy, A.M. *Robust Regression and Outlier Detection*; Wiley-Interscience: Hoboken, NJ, USA, 2003.
18. Fischler, M.A.; Bolles, R.C. Random sample consensus. *Commun. ACM* **1981**, *24*, 381–395. [CrossRef]

19. Breiman, L.; Friedman, J.; Stone, C.J.; Olshen, R.A. *Classification and Regression Trees*; Taylor & Francis: Abingdon, UK, 2017.
20. Von Winterfeldt, D.; Edwards, W. *Decision Analysis and Behavioral Research*; Cambridge University Press: Cambridge, UK, 1986.
21. Ho, T.K. The random subspace method for constructing decision forests. *IEEE Trans. Pattern Anal. Mach. Intell.* **1998**, *20*, 832–844. [[CrossRef](#)]
22. Geurts, P.; Ernst, D.; Wehenkel, L. Extremely randomized trees. *Mach. Learn.* **2006**, *63*, 3–42. [[CrossRef](#)]
23. Rao, H.; Shi, X.; Rodrigue, A.K.; Feng, J.; Xia, Y.; Elhoseny, M.; Yuan, X.; Gu, L. Feature selection based on artificial bee colony and gradient boosting decision tree. *Appl. Soft Comput.* **2019**, *74*, 634–642. [[CrossRef](#)]
24. Guryanov, A. Histogram-Based Algorithm for Building Gradient Boosting Ensembles of Piecewise Linear Decision Trees. In *International Conference on Analysis of Images, Social Networks and Texts*; Springer: Cham, Switzerland, 2019; pp. 39–50. [[CrossRef](#)]
25. Awad, M.; Khanna, R. Support Vector Regression. In *Efficient Learning Machines*; Apress: Berkeley, CA, USA, 2015; pp. 67–80. [[CrossRef](#)]
26. Hu, L.; Che, X.-L. A Nu-support Vector Regression Based System for Grid Resource Monitoring and Prediction. *Acta Autom. Sin.* **2010**, *36*, 139–146. [[CrossRef](#)]
27. Cover, T.; Hart, P. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory* **1967**, *13*, 21–27. [[CrossRef](#)]
28. Urrego, B.C.; Alsup, A.; Thompson, J.A.; Koestler, D.C. A stochastic approach to k-nearest neighbors search using a fixed radius method. *Comput. Stat.* **2026**, *41*, 27. [[CrossRef](#)] [[PubMed](#)]
29. Mitchell, T.M. *Machine Learning*; McGraw-Hill Science: New York, NY, USA, 1997.
30. Cybenko, G. Approximation by superpositions of a sigmoidal function. *Math. Control Signals Syst.* **1989**, *2*, 303–314. [[CrossRef](#)]
31. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [[CrossRef](#)] [[PubMed](#)]
32. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. In *3rd International Conference for Learning Representations; ICLR*: San Diego, CA, USA, 2017.
33. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958. Available online: https://www.jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf?utm_content=buffer79b4 (accessed on 5 May 2026).
34. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proceedings of the 32nd International Conference on Machine Learning*; Bach, F., Blei, D., Eds.; PMLR: Lille, France, 2015; pp. 448–456. Available online: <http://proceedings.mlr.press/v37/ioffe15.pdf> (accessed on 5 May 2026).
35. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2016; pp. 770–778. [[CrossRef](#)]
36. Chen, D.; Hu, F.; Nian, G.; Yang, T. Deep Residual Learning for Nonlinear Regression. *Entropy* **2020**, *22*, 193. [[CrossRef](#)] [[PubMed](#)]
37. Bai, S.; Kolter, J.Z.; Koltun, V. An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. *arXiv* **2018**, arXiv:1803.01271.
38. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Polosukhin, I. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*; Curran Associates Inc.: Long Beach, CA, USA, 2017; pp. 6000–6010.
39. Gilmer, J.; Schoenholz, S.S.; Riley, P.F.; Vinyals, O.; Dahl, G.E. Neural Message Passing for Quantum Chemistry. In *Proceedings of the 34th International Conference on Machine Learning*; PMLR: Lille, France, 2017; pp. 1263–1272. Available online: <https://proceedings.mlr.press/v70/gilmer17a.html> (accessed on 5 May 2026).
40. Arik, S.Ö.; Pfister, T. TabNet: Attentive Interpretable Tabular Learning. *Proc. AAAI Conf. Artif. Intell.* **2021**, *35*, 6679–6687. [[CrossRef](#)]
41. Raissi, M.; Perdikaris, P.; Karniadakis, G.E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **2019**, *378*, 686–707. [[CrossRef](#)]
42. Koza, J.R. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*; MIT Press: Cambridge, MA, USA, 2000.
43. Kronberger, G.; Burlacu, B.; Kommenda, M.; Winkler, S.M.; Affenzeller, M. *Symbolic Regression*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2024. [[CrossRef](#)]
44. Chen, S.; Zhou, X.; Zhou, G.; Fan, C.; Ding, P.; Chen, Q. An online physical-based multiple linear regression model for building's hourly cooling load prediction. *Energy Build.* **2022**, *254*, 111574. [[CrossRef](#)]
45. Moradi, K.; Jafari, F.; Moghaddam, F.; Shafiee, Q. Dynamic modeling of photovoltaic–thermal systems using polynomial regression. *Renew. Energy* **2025**, *250*, 123242. [[CrossRef](#)]
46. Zheng, Y.; Ge, Y.; Muhsen, S.; Wang, S.; Elkamchouchi, D.H.; Ali, E.; Ali, H.E. New ridge regression, artificial neural networks and support vector machine for wind speed prediction. *Adv. Eng. Softw.* **2023**, *179*, 103426. [[CrossRef](#)]
47. Yang, H.; Tang, J.; Shao, W.; Yin, J.; Liu, B. Wind power data cleaning using RANSAC-based polynomial and linear regression with adaptive threshold. *Sci. Rep.* **2025**, *15*, 5105. [[CrossRef](#)] [[PubMed](#)]
48. Xu, N.; Wang, Z.; Dai, Y.; Li, Q.; Zhu, W.; Wang, R.; Finkelman, R.B. Prediction of higher heating value of coal based on gradient boosting regression tree model. *Int. J. Coal Geol.* **2023**, *274*, 104293. [[CrossRef](#)]

49. Cai, W.; Wen, X.; Li, C.; Shao, J.; Xu, J. Predicting the energy consumption in buildings using the optimized support vector regression model. *Energy* **2023**, *273*, 127188. [\[CrossRef\]](#)
50. Yuan, Z.; Chen, L.; Liu, G.; Shao, W.; Zhang, Y.; Yang, W. Physics-based Bayesian linear regression model for predicting length of mixed oil. *Geoenery Sci. Eng.* **2023**, *223*, 211466. [\[CrossRef\]](#)
51. Alshahrani, S.M.; Alotaibi, H.F.; Alqarni, M. Modeling and validation of drug release kinetics using hybrid method for prediction of drug efficiency and novel formulations. *Front. Chem.* **2024**, *12*, 1395359. [\[CrossRef\]](#) [\[PubMed\]](#)
52. Minkowitz, L.; Arneitz, S.; Effertz, P.S.; Amancio-Filho, S.T. Laser-powder bed fusion process optimisation of AlSi10Mg using extra trees regression. *Mater. Des.* **2023**, *227*, 111718. [\[CrossRef\]](#)
53. Thajudeen, K.Y.; Ahmed, M.M.; Alshehri, S.A. Integration of machine learning and CFD for modeling mass transfer in water treatment using membrane separation process. *Sci. Rep.* **2024**, *14*, 23970. [\[CrossRef\]](#) [\[PubMed\]](#)
54. Kopal, I.; Labaj, I.; Vršková, J.; Harničárová, M.; Valíček, J.; Bakošová, A.; Tozan, H.; Khanna, A. Predictive Modelling and Optimisation of Rubber Blend Mixing Using a General Regression Neural Network. *Polymers* **2025**, *17*, 1868. [\[CrossRef\]](#) [\[PubMed\]](#)
55. Mihalev, G. Approximation of Dynamic Systems Using Deep Neural Networks and Laguerre Functions. *Eng. Proc.* **2025**, *104*, 22. [\[CrossRef\]](#)
56. Liu, J.; Hao, J.; Dunabuqi, S.; Zhao, H.; Li, Y.; Wu, Q.; Bai, M. Intelligence modeling of pharmaceutical solubility in supercritical CO₂: Polynomial regression, extra trees, and Huber regression models and validation. *Chin. J. Phys.* **2025**, *97*, 1024–1035. [\[CrossRef\]](#)
57. Cysewski, P.; Jeliński, T.; Przybyłek, M. Exploration of the Solubility Hyperspace of Selected Active Pharmaceutical Ingredients in Choline- and Betaine-Based Deep Eutectic Solvents: Machine Learning Modeling and Experimental Validation. *Molecules* **2024**, *29*, 4894. [\[CrossRef\]](#) [\[PubMed\]](#)
58. Rasmussen, C.M.; Viozzi, C.F. A Decision Tree for Orthodontic and Surgical Management of the Maxillary Transverse Dimension in Orthognathic Surgery. *J. Oral Maxillofac. Surg.* **2020**, *78*, 1236–1238. [\[CrossRef\]](#) [\[PubMed\]](#)
59. Srinivasan, S.; Deepalakshmi, P. ENetRM: ElasticNet Regression Model based malicious cyber-attacks prediction in real-time server. *Meas. Sens.* **2023**, *25*, 100654. [\[CrossRef\]](#)
60. Shyla, S.; Bhatnagar, V.; Bali, V.; Bali, S. Optimization of Intrusion Detection Systems Determined by Ameliorated HNADAM-SGD Algorithm. *Electronics* **2022**, *11*, 507. [\[CrossRef\]](#)
61. Aflaki, A.; Gitizadeh, M.; Kantarci, B. Accuracy improvement of electrical load forecasting against new cyber-attack architectures. *Sustain. Cities Soc.* **2022**, *77*, 103523. [\[CrossRef\]](#)
62. Huang, L.; Song, T.; Jiang, T. Linear regression combined KNN algorithm to identify latent defects for imbalance data of ICs. *Microelectron. J.* **2023**, *131*, 105641. [\[CrossRef\]](#)
63. de Barros Christ, A.K.; Roque, C.; Moura, F. Estimate traffic cyclist crashes using Poisson-Tweedie models. *Accid. Anal. Prev.* **2025**, *223*, 108256. [\[CrossRef\]](#) [\[PubMed\]](#)
64. Zhou, W.; Sansone, P.; Jia, Z.; Gomez, M.-A.; Li, F. Determining the key performance indicators on game outcomes in NBA based on quantile regression analysis. *Int. J. Perform. Anal. Sport* **2024**, *24*, 519–534. [\[CrossRef\]](#)
65. Chong, W.S.; Zaki, N.H.M.; Hossain, M.S.; Muslim, A.M.; Pour, A.B. Introducing Theil-Sen estimator for sun glint correction of UAV data for coral mapping. *Geocarto Int.* **2022**, *37*, 4527–4556. [\[CrossRef\]](#)
66. Schmidt, M.; Lipson, H. Distilling Free-Form Natural Laws from Experimental Data. *Science* **2009**, *324*, 81–85. [\[CrossRef\]](#) [\[PubMed\]](#)
67. Li, S. Estimating Stock Market Prices with Histogram-based Gradient Boosting Regressor: A Case Study on Alphabet Inc. *Int. J. Adv. Comput. Sci. Appl.* **2024**, *15*, 53. [\[CrossRef\]](#)
68. Schaeffer, J.; Lenz, E.; Chueh, W.C.; Bazant, M.Z.; Findeisen, R.; Braatz, R.D. Interpretation of high-dimensional linear regression: Effects of nullspace and regularization demonstrated on battery data. *Comput. Chem. Eng.* **2024**, *180*, 108471. [\[CrossRef\]](#)
69. Natarajan, R.; Krishna, S.; Ganesan, P.; Al Kaabi, A.; Kala, R.; Madlambayan, A.M. Revolutionizing Agriculture Through Smart Farming by Employing Advanced Machine Learning Techniques for Optimal Crop Selection. *J. Comput. Cogn. Eng.* **2025**, *4*, 546–560. [\[CrossRef\]](#)
70. Thangamani, R.; Regulagedda, P. A Computational Complexity-Based Framework for Predicting the Difficulty of Sudoku Puzzles Using CSP-Based Symbolic Modeling and Constraint Metrics. *J. Comput. Cogn. Eng.* **2025**, *5*, 115–132. [\[CrossRef\]](#)
71. Mohsen, B.M.; Mohsen, M. AI-Driven Prediction of Greenhouse Gas Emissions in Livestock Supply Chains: Towards a Data-Driven Model for Sustainable Agri-food Systems. *J. Comput. Cogn. Eng.* **2025**, *5*, 169–175. [\[CrossRef\]](#)

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.