Review of Regression Models and Their Characteristics

Authors Names

[a]*Layth S. Ibrahaim*

[b]*Tasnim H.K. ALbaldawi*

Publication data: 16 /11 /2024

Keywords: *Linear* Regression(LR), Polynomial Regression(PR), Decision Tree Regression (DTR), Neural Network Regression (NNR), Random Forest Regression, Support Vector Regression (SVR), Time Series Regression(TSR), Spatial Linear Regression(SLR).

ABSTRACT

The paper provides an overview of regression models commonly used in statistics and machine learning, emphasizing their importance in predicting and understanding relationships between variables across diverse datasets. The models covered include Linear Regression (LR), polynomial regression (PR), Decision Tree Regression (DTR), Neural Network Regression (NNR), Random Forest Regression, and Support Vector Regression (SVR), With the introduction of special models of regression models, they are Time Series Regression (TSR) and Spatial Linear Regression (SLR). Linear regression focuses on linear relationships, while polynomial regression captures nonlinear patterns by introducing polynomial terms. Decision trees and random forests, as ensemble methods, partition data recursively, while support vector regression uses support vector coefficients and kernel functions to handle nonlinear relationships. GRNN is a fast and efficient model in certain situations and may struggle with performance on large datasets, while it is more flexible and easy to customize, but requires intensive training to achieve outstanding results. Time series regression (TSR) is a powerful tool for modeling and forecasting time-dependent data while spatial linear regression (SLR) is a powerful extension of traditional linear regression that incorporates spatial relationships, enabling it to analyze and forecast spatial data.

1.Introduction

In statistics, a statistical model is a component of a mathematical model that incorporates a set of assumptions and primarily focuses on generating data similar to and synthesized from a larger sample [1]. Essentially, a statistical model serves as a process for generating data. The assumptions employed by a statistical model involve a combination of probability distributions, some of which adequately characterize the specific dataset. The inherent use of probability is a distinctive feature of statistical models. Regression, within statistical modeling, closely examines how a dependent variable is affected by varying an independent variable while keeping the other independent variables constant [2]. Regression Analysis, within the realm of statistical modeling, emerges as a technique designed to uncover relationships between various variables. Mathematicians and data scientists employ regression analysis for prediction and forecasting. The process entails selecting the appropriate model based on the provided dataset and subsequently utilizing that model for making predictions. An ideal model accurately captures all relationships. Consequently, a tool based on regression analysis can offer valuable insights to economists or managers. The diverse applications and advantages of Regression Analysis include: Prediction of the future: Regression Analysis, applied to a relevant dataset, can accurately predict various valuable pieces of information, such as Stock Prices, Medical Conditions, and even Public Sentiments. Support for major decisions and policies: Results derived from regression analysis provide a scientific foundation for decisions or policies, enhancing their reliability and likelihood of success [1]. Correction of errors in thinking or dispelling misconceptions: Discrepancies between the predictions of regression analysis and a decision or thought process can sometimes help rectify the fallacies in decision-making. Provision of a new perspective: The application of Regression Analysis to large datasets unleashes their potential to introduce new dimensions to a study. Nonlinear regression models are important tools for solving optimization problems [3]. Therefore, Regression Analysis stands out as a crucial tool for Data Scientists working with diverse datasets. To obtain accurate results from different types of datasets with varied relationships, various types of Regression Analysis models are employed**.**

[a] *Department of Mathematic, College of Science, University of Baghdad, Baghdad, Iraq*, E-Mail: *laith.salem.ibrahim@gmail.com*

[b] *Department of Mathematic, College of Science, University of Baghdad, Baghdad, Iraq*, E-Mail: *tasnim.h@sc.uobaghdad.edu.iq*

2. Models

Each model is characterized by its underlying principles, equations, and key parameters. The following provides a detailed overview of the selected models:

2.1. Linear Regression

Linear regression is the most simple regression analysis technique. It is the most commonly regression analysis mechanism in predictive analysis [4]. Linear regression shows the linear relationship between the independent variable (X-axis) and the dependent variable (Y-axis), consequently called linear regression. If there is a single input variable (x), such linear regression is called **simple linear regression**. And if there is more than one input variable, such linear regression is called **multiple linear regression.** The linear regression model gives a sloped straight line describing the relationship within the variables.

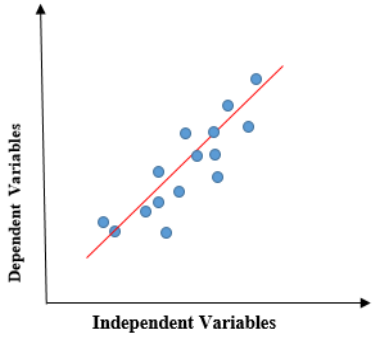

Fig. 1. Linear Relationship between the dependent variable and independent variables.

The above graph presents the linear relationship between the dependent variable and independent variables. When the value of x (**independent variable**) increases, the value of y (**dependent variable**) is likewise increasing. The red line is referred to as the best fit straight line. Based on the given data points, we try to plot a line that models the points the best. We determine the given data points that we plot in a way that best fits the variables, here regression shows a line or curve that passes through all the data points on a graph Regression shows a line or curve that passes through all the data points. Regression is mainly used for forecasting, modeling time series, and determining the relationships between variables in terms of effect and causation. Linear regression estimates the relationship between a dependent variable and an independent variable.

- **Positive Linear Relationship**

If the dependent variable expands on the Y-axis and the independent variable progress on X-axis, then such a relationship is termed a Positive linear relationship.

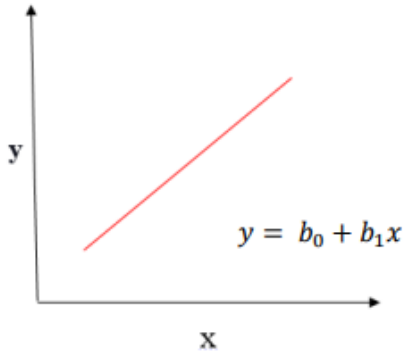

Fig. 3. Positive linear relationship

- **Negative Linear Relationship**

If the dependent variable decreases on the Y-axis and the independent variable increases on the X-axis, such a relationship is called a negative linear relationship.

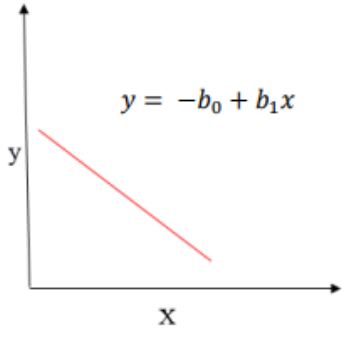

Fig. 4. Negative linear relationship

The goal of the linear regression algorithm is to get the best values for b_0 and b_1 to find the best fit line. The best fit line should have the least error means the error between predicted values and actual values should be minimized.

2.1.1. Simple linear regression

The purpose of simple regression analysis is to evaluate the relative impact of a predictor variable on a particular outcome [5]. A very straightforward approach to predicting a quantitative response Y on the basis of a single predictor variable X. It assumes that there is an approximately linear relationship between X and Y. Mathematically, we can write this linear relationship as follows:

$$Y \approx b_0 + b_1 X \quad (1)$$

You might read " ≈ " as "approximately modeled". We sometimes describe (1) by saying that we are regressing Y to X (or Y to X). Y is Dependent Variable, x is Independent Variable, b_0 is intercept of the line, b_1 is Linear regression coefficient.

$$\hat{y} = \hat{b}_0 + \hat{b}_1 x \quad (2)$$

where ($\hat{y}$) refers to the prediction of Y on the basis of $X = x$. Here we use the hat symbol (ˆ) , to indicate the estimated value of a parameter or to indicate the expected value of a response.

2.1.1.1. Simple Linear Regression Algorithm

Inputs	A set of points (x_i, y_i) where $i = 1, 2, \ldots, n$.	
Outputs:	The equation of the straight line: $y = b_1 x + b_0$.	
Steps:	1. Calculate the Mean	Calculate the mean of both x and y. $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ (3) $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ (4)
	2. Calculate the Slope b_1	Calculate the slope using the following formula: $b_1 = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sum_{i=1}^{n}(x_i-\bar{x})^2}$ (5)
	3. Calculate the Intercept b_0	Use the following formula to calculate the intercept: $b_0 = \bar{y} - b_1 \times \bar{x}$ (6)
	4. Equation of the Line	Based on the calculated values of m and c, the equation of the line will be (1)
Usage:	This algorithm can be used to predict future values of y based on input values of x.	

2.1.2. Multiple linear regression [6]

Multiple linear regression extends simple linear regression to include more than one explanatory variable. It is one of the important regression models that models the linear relationship between more than one independent variable and one continuous dependent variable.

In both cases, we still use the term 'linear' because we assume that the response variable is directly related to a linear combination of the explanatory variables. The equation for multiple linear regression has the same form as that for simple linear regression but has more terms:

$$y_i = b_0 + b_1 x_{1i} + b_2 x_{2i} + \cdots + b_p x_{pi} + e_i \quad (7)$$

As for the simple case, b_0 is the constant – which will be the predicted value of y when all explanatory variables are 0. In a model with p explanatory variables, each explanatory variable has its own b_coefficient.

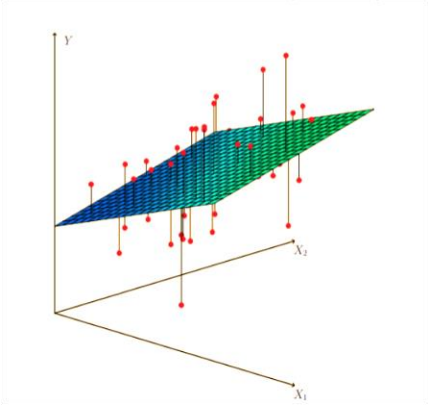

Fig. 5. Multiple linear regression

2.1.2.1. Multiple Linear Regression Algorithm

Inputs	A set of data containing multiple independent variables $x_1, x_2, \dots, x_p$ and a dependent variable y.	
Output	Multilinear equation: $y = b_0 + b_1x_1 + b_2x_2 + \cdots + b_px_p$ (8) Where b_0 is the constant (intercept), and b_1, b_2 , $\cdots$, b_p are the regression coefficients of the independent variables.	
Steps	1. Prepare the data	Arrange the data so that it contains the independent variables x_1 , x_2 , $\cdots$, x_p and the dependent variable y.
	2. Create the design matrix	We create a design matrix X that has rows representing the samples and columns representing the independent variables. We add a column of fixed values (usually 1) to represent the constant b_0 . $X = \begin{bmatrix} 1 & x_{11} & x_{21} & \cdots & x_{n1} \\ 1 & x_{12} & x_{22} & \dots & x_{n2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{1p} & x_{2p} & \cdots & x_{np} \end{bmatrix}$ (9) where n is the number of samples.
	3. Calculating regression coefficients	We use the following formula to calculate the regression coefficients b. $b = (X^TX)^{-1}yX^T$ (10) Where X^T is the transposed matrix of X , $(X^TX)^{-1}$ is the inverse matrix of X^TX.
	4. Building the regression	Based on the calculated values of b_1 and

	equation	b_0, the equation of the line will be (1)
Usage	Once you have implemented this algorithm, you can use the multiple regression equation to predict future values of y based on the input independent variables x_1 , x_2 ,⋯ , x_p	

2.1.2.2. The most important Advantages and Disadvantages of the model

Advantages	Disadvantages
1- Its ability to use more than one independent variable, which helps model complex relationships between variables and improves forecasting accuracy.	1- Sensitivity to outliers It is sensitive to outliers that can significantly affect the final results.
2- Easy to interpret each regression coefficient individually while holding the rest of the variables constant.	2- It relies on several assumptions, such as linearity, homogeneity of variance, normal distribution of errors, and independence of errors.
3- It can work effectively with large and complex data that contains many independent variables.	3- If these assumptions are not met, the model may not be accurate.
4- It can be used to predict continuous values, making it useful in many applications such as price forecasting, sales, and others.	4- If there is a large overlap between the independent variables (i.e. if the variables are highly correlated with each other), this can lead to non-stationarity of the coefficients and thus make interpretation difficult.
5- It can be used to test the importance of each independent variable and determine the extent of its effect on the dependent variable using p-values and fixed coefficients.	5- This model assumes that the relationship between the independent variables and the dependent variable is linear. If the relationship is non-linear, it may fail to provide accurate predictions. 6- It can be difficult to deal with missing values in the data when using it and this may lead to distortion of the results.

Table 1. Advantages of multiple linear regression

2.2. Polynomial Regression

Polynomial regression is a special case of multiple regression, with only one independent variable X. One-variable polynomial regression model can be expressed as [7]

$$y = b_0 + b_1 x + b_2 x^2 + \cdots + b_n x^n \quad (11)$$

In this model the original features are converted into polynomial features of the desired degree (2,3,..,n) and thus modeled using a linear model.

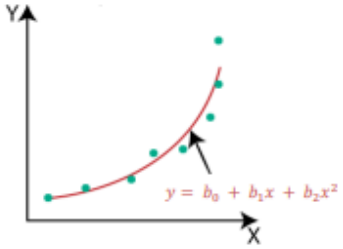

Fig. 6. Quadratic polynomial model

Polynomial regression is a special case of multiple regression, with only one independent variable X. One-variable polynomial regression model can be expressed as

$$y_i = b_0 + b_1 x_{1i} + b_2 x_i^2 + \cdots + b_p x_i^p + e_i \quad (12)$$

Where p is the degree of the polynomial. The degree of the polynomial is the order of the model [8].

2.2.1. Polynomial Regression Algorithm

Inputs	• A dataset containing an independent variable x and a dependent variable y. • The desired degree of the polynomial regression d.	
Outputs	The polynomial regression equation: $y_i = b_0 + b_1 x + b_2 x^2 + \cdots + b_d x^d$ (13) Where $b_0, b_1, \ldots, b_d$ are the regression coefficients.	
Steps	1. Prepare the Data	Collect the data containing the independent variable x and the dependent variable y.
	2. Transform the Independent Variable	Expand the independent variable x into a set of polynomial features up to the degree d. For example, if x is your original variable, it will be transformed into: $X_{poly} = [1, x, x^2, \ldots, x^d]$ (14) Where each row contains different powers of x from x_0 (representing the intercept) to x_d.
	3. Create the Design Matrix	Create the design matrix X_{poly} where rows represent the samples and columns represent the polynomial features.
	4. Calculate the Regression Coefficients	Use the following formula to calculate the regression coefficients b: $b = \left(X_{poiy}^T X_{poly}\right)^{-1} X_{poiy}^T y$ (15)

		Where X_{poiy}^T is the transpose of X_{poly}, and $\left(X_{poiy}^T X_{poly}\right)^{-1}$ is the inverse of the matrix $X_{poiy}^T X_{poly}$.
	5. Construct the Polynomial Regression Equation	After calculating the coefficients $b_0, b_1, \dots, b_d$ the polynomial regression equation will be: $y = b_0 + b_1 x_1 + b_2 x_2 + \cdots + b_d x_d$ (16)
Usage	Once you implement this algorithm, you can use the polynomial regression equation to predict future values of y based on the input value x.	

2.2.2. The most important Advantages and Disadvantages of the model

Advantages	Disadvantages
1- Polynomial regression allows for modeling complex, non-linear relationships between the independent and dependent variables. By increasing the degree of the polynomial, the model can capture more intricate patterns in the data.	1- As the degree of the polynomial increases, the model becomes more complex and may fit the training data too closely, leading to overfitting. This reduces the model's ability to generalize to new, unseen data.
2- When data does not follow a linear trend, polynomial regression can provide a better fit compared to linear regression, resulting in more accurate predictions.	2- Higher-degree polynomials require more computational resources and time to calculate, especially with large datasets or very high degrees.
3- Although more complex than simple linear regression, polynomial regression still maintains some level of interpretability. The coefficients can indicate the influence of each polynomial term on the dependent variable.	3- As the degree of the polynomial increases, the model becomes harder to interpret. Understanding the influence of higher-degree terms on the outcome can be challenging.
4- Polynomial regression can be applied to a wide range of problems in various fields, such as economics, engineering, and natural sciences, where relationships between variables are non-linear.	4- Polynomial regression is sensitive to outliers. A single outlier can significantly distort the curve, leading to inaccurate predictions.
5- It is a straightforward extension of linear regression, allowing the same foundational methods to be applied to more complex data structures.	5- Polynomial models can behave unpredictably outside the range of the training data. Extrapolating beyond the observed data can result in nonsensical predictions, especially with high-degree polynomials.
	6- When polynomial terms are added, especially in higher degrees, multicollinearity can become an issue, where independent variables (or their polynomial terms) are highly correlated with each other. This can lead to unstable coefficient estimates.

Table 2. Advantages and Disadvantages of Polynomial Regression

2.3. Decision Tree Regression

Decision Trees (D T) recursively split the data based on features, aiming to maximize information gain (classification) or variance reduction (regression) at each node. Decision Tree is a Supervised Machine Learning approach to solve classification and regression problems by continuously splitting data based on a certain parameter [9].

2.3.1. Decision Tree Regression Algorithm

Inputs	• A dataset with features $x_1, x_2, \dots, x_p$ and a target variable y. • A splitting criterion, typically the reduction in variance (or another metric such as mean squared error). • A stopping criterion, like the maximum depth of the tree or minimum number of samples per leaf.	
Outputs	A decision tree model that predicts the target variable y for a given input x.	
Steps:	**1. Initialization**	Collect the data containing the independent variable x and the dependent variable y.
	2. Splitting Criterion	At each node, calculate the optimal split by considering all possible splits for each feature. The goal is to choose the split that minimizes the variance (or other error metric) in the target variable within the resulting subsets. **Variance Reduction (Variance Minimization):** For a given split S that divides the dataset into two subsets S_1 and S_2, the variance before the split is: $Var_{total} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2$ (17) where n is the total number of samples, y_i is the target value for the ith sample, and $\bar{y}$ is the mean of the target values. After the split, the variance for each subset is calculated as: $Var(S_1) = \frac{1}{n_1}\sum_{i=1}^{n}(y_i - \bar{y}_1)^2$ (18) $Var(S_2) = \frac{1}{n_2}\sum_{i=1}^{n}(y_i - \bar{y}_2)^2$ (19) where n_1 and n_2 are the number of samples in subsets S_1 and S_2 , and $\bar{y}_1$ and $\bar{y}_2$ are the means of the target values in those subsets. The total variance after the split is: $Var_{split} = \frac{n_1}{n}Var(S_1) + \frac{n_2}{n}Var(S_2)$ (20) The variance reduction (or reduction in mean squared error) achieved by the split is: $Variance\ Reduction = Var_{total} - Var_{split}$ (21)

		Choose the split that maximizes this variance reduction.
	3. Recursive Splitting	Recursively apply the splitting criterion to each subset, creating new nodes in the tree until a stopping criterion is met. Common stopping criteria include: Maximum depth of the tree. Minimum number of samples required to split a node. Minimum variance reduction required to perform a split.
	4. Stopping Condition	If a stopping criterion is met, stop splitting and assign a predicted value to the leaf node. The predicted value is typically the mean of the target values in that node: $\hat{y}_{leaf} = \frac{1}{n_{leaf}} \sum_{i \in leaf} y_i$ (22) Where n_{leaf} is the number of samples in the leaf.
	5. Prediction	To predict the target value for a new input x, traverse the tree starting from the root, following the decisions at each node based on the input feature values, until reaching a leaf. The predicted value is the value associated with that leaf.
Usage:	Once you implement this algorithm, you can use the polynomial regression equation to predict future values of y based on the input value x.	

2.3.2. The most important Advantages and Disadvantages of the model

Advantages	Disadvantages
1- Based on certain features, the decision tree is one of the easiest machine learning models to interpret and understand, as the path that the model follows to make a decision can be traced.	1- If the depth of the tree or the size of the nodes is not controlled, the decision tree can lead to overfitting, which leads to poor performance when predicting new data.
2- The decision tree can deal with non-linear relationships between the target variable and the features without the need for data transformation or complex models.	2- The decision tree may be unstable, because small changes in the data may lead to the creation of a significantly different tree.
3- The decision tree does not require initial data settings before using it, which makes it easy to use directly on raw data.	3- In case of small data or data with many attributes, the decision tree may be inefficient and thus give inaccurate predictions.
4- The decision tree has the ability to deal with quantitative and qualitative variables easily in the same model.	4- If the relationship between the target variable and the attributes is linear, the decision tree will not be the optimal model for new predictions.

5- The decision tree is robust in dealing with data that contains missing values and does not require complex methods. 6- The decision tree is less sensitive to outliers. 7- Based on certain features, the decision tree is one of the easiest machine learning models to interpret and understand, as the path that the model follows to make a decision can be traced.	5- Predictions in a decision tree are not smooth and the predictive values may change when moving from one node to another.

Table 3. Advantages and Disadvantages of Decision Tree Regression

2.4.General Regression Neural Network(GRNN) and Neural Network Regression (NNR)

GRNN is a single-pass learning algorithm with a perfectly parallel architecture. This algorithm provides a flexible transition from one value to another, even with sparse data in a multi-dimensional measurement space [10]. For each layer l in the network (excluding the input layer):

$$z_l = W_l a_{l-1} + b_l \tag{23}$$

Where z_l: weighted input , W_l : weights, b_l: biases and a_{l-1} : the output

$$a_l = f_l(z_l) \tag{24}$$

Where f_l activation function to obtain the outputs of layer l. Equations (23) and (24) are applied to all layers until the last layer.

2.4.1.1. General Regression Neural Network (GRNN) Algorithm

Inputs	A dataset with features $x_1, x_2, \dots, x_p$ and a target variable y. A smoothing parameter σ, which controls the width of the kernel function.	
Outputs	A predicted value $\hat{y}$ for a new input vector x.	
Steps:	1. Network Structure	The GRNN consists of four layers: Input Layer: Each input feature x_i is passed directly to the Pattern layer. Pattern Layer (or Radial Basis Layer): This layer calculates the distance between the input vector x and each training sample vector x_j. The output of each neuron in this layer is computed using the Gaussian kernel: $G_j(x) = \exp\left(\frac{\|x-x_j\|^2}{2\sigma^2}\right)$ (25) Where $\|x - x_j\|$ is the Euclidean distance

		between the input vector x and the training sample x_j, and σ is the smoothing parameter. Summation Layer: This layer has two nodes: Node S: The weighted sum of the outputs from the Pattern layer, weighted by the target values: $S = \sum_{j=1}^{n} y_j G_j(x)$ (26) Node D: The sum of the outputs from the Pattern layer: $D = \sum_{j=1}^{n} G_j(x)$ (27) Output Layer: The predicted value $\hat{y}$ is computed as the ratio of the sums from the Summation layer: $\hat{y} = \frac{S}{D} = \frac{\sum_{j=1}^{n} y_j G_j(x)}{\sum_{j=1}^{n} G_j(x)}$ (28)
	2. Training	Unlike traditional neural networks, GRNN does not require iterative training. It simply stores the training data and computes the necessary values at prediction time.
	3. Prediction	For a new input vector x, the GRNN calculates the predicted value $\hat{y}$ using the formula from the Output layer. The network automatically adjusts to new data by adding new training samples to its memory, making it inherently adaptable.
	4. Smoothing Parameter σ	The performance of GRNN is highly dependent on the choice of the smoothing parameter σ. A small σ\sigmaσ leads to a model that closely follows the training data (risking overfitting), while a large σ produces a smoother, more generalized model. The optimal σ can be selected using cross-validation.

2.4.2. Neural Network Regression (NNR)

refers to a subfield of artificial intelligence that is similar to the brain, where computers have the option to understand things and make decisions in a human-like way [11]. It is a neural network consisting of two or more layers, and NNR is used to predict time series [12]

$$y_t = g(y_{t-1}, y_{t-2}, \ldots y_{t-p}) + e_t \quad (29)$$

where e_t is the error at time t . The sigmoid function is represented in the following form, knowing that it is a biased function:

$$sigmoid(x) = \frac{1}{1+e^{-y}} \quad (30)$$

2.4.2.1.Neural Network Regression (NNR) Algorithm

<table>
<tr><td>Inputs</td><td colspan="2">• A dataset with features $x_1, x_2, \dots, x_p$ and a target variable y.
• Number of hidden layers and the number of neurons in each layer.
• An activation function for the hidden layers, such as $ReLU$, Sigmoid, or $Tanh$.
• A loss function, such as Mean Squared Error (MSE).
• An optimization algorithm, such as Gradient Descent or Adam.</td></tr>
<tr><td>Outputs</td><td colspan="2">A trained neural network model that can predict the target variable y for new input data.</td></tr>
<tr><td rowspan="5">Steps:</td><td>1. Initialize the Network</td><td>Set up the network architecture by determining the number of input features, the number of hidden layers, and the number of neurons in each layer.
Initialize the weights and biases of the network, often with small random values or using specific initialization methods like Xavier initialization</td></tr>
<tr><td>2. Forward Propagation</td><td>For a given input x, propagate the input through the network:
Compute the linear combination of inputs and weights for each neuron in the hidden layers.
Apply the activation function to the result.
Repeat this process for each hidden layer, passing the output of one layer as the input to the next.
Finally, compute the output of the network in the output layer (which typically does not use an activation function for regression).</td></tr>
<tr><td>1. Compute the Loss</td><td>Calculate the loss using a loss function like Mean Squared Error (MSE):
$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2$ (31)
Here, $\hat{y}_i$ is the predicted output, and y_i is the actual target value.</td></tr>
<tr><td>2. Backward Propagation σ</td><td>Compute the gradient of the loss function with respect to each weight and bias in the network using backpropagation:
Calculate the derivative of the loss with respect to the output of the network.
Propagate this error backward through the network, layer by layer, calculating the gradient with respect to the weights and biases.
Use the chain rule to update the gradients for each layer.</td></tr>
<tr><td>3. Update Weights and Biases</td><td>Update the weights and biases of the network using an optimization algorithm such as Gradient Descent:
$w_{ij} = w_{ij} - \eta \frac{\partial Loss}{\partial w_{ij}}$ (32)</td></tr>
</table>

		Here, w_{ij} represents the weights, η is the learning rate, and $\frac{\partial Loss}{\partial w_{ij}}$ is the gradient of the loss with respect to the weight.
	4. Iterate	Repeat steps 2-5 for a specified number of iterations (epochs) or until the loss converges to a minimum value.
	5. Prediction	Once the network is trained, use it to predict the target variable y for new input data x.
	6. Evaluate the Model	After training, evaluate the model's performance on a test dataset by calculating metrics like Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), or R^2.

The most important differences between the properties of the generalized regression neural network (GRNN) and the neural network regression (NNR)

Criteria	General Regression Neural Network (GRNN)	Neural Network Regression (NNR)
1- Structure and Design	Fixed four-layer structure (Input, Pattern, Summation, Output)	Multilayer structure (Input, multiple Hidden layers, Output)
2- Training Method	No iterative training required; relies on storing data	Requires iterative training to optimize weights using optimization algorithms
3- Activation Functions	Relies on Gaussian function as the kernel function	Uses multiple activation functions like $ReLU$, Sigmoid, and $Tanh$
4- Data Handling	Treats each point in the training set individually	Generalizes patterns across hidden layers
4- Efficiency with Large Data	May be slow due to storing and evaluating all data during prediction	More efficient with large datasets due to its generalization capabilities
5- Stability and Reliability	Does not require training and relies on full data storage, making it stable	Depends on the quality of training and can be unstable if not trained properly
6- Customization Flexibility	Limited flexibility, primarily depends on the smoothing parameter σ	Highly customizable by adjusting the number of hidden layers and neurons
7- Performance with Non-linear Relationships	Performs well with non-linear and complex relationships	Can be effective with non-linear relationships if trained properly
8- Generalization	Can lead to overfitting in some cases	Has strong generalization capabilities if trained well
9- Sensitivity to Outliers	Less sensitive to outliers	Can be more sensitive to outliers if not handled properly
10- Practical Application	Suitable for problems requiring	Suitable for problems requiring

	quick results without intensive training	customized models with strong generalization

Table 4. Comparison Table between General Regression Neural Network (GRNN) and Neural Network Regression (NNR)

2.5.Random Forest Regression (RFR)

Random forest is a type of supervised learning algorithm that uses ensemble methods to solve both regression and classification problems [11].
It is a mixture of tree predictions $\{h(x,\Theta_i), i = 1,2,\dots\}$, such that each tree depends on the values of the random vector $\{\Theta_i\}$ sampled independently and with the same distribution for all trees in the forest . [13], [14]. Under a given independent variable , each Decision Trees gives an opinion on choosing the best outcome [14]. The generalization error of any Decision Trees $h(x)$

$$h(x) = E_{X;Y}\left(Y - h(X)\right)^2 \tag{33}$$

Where X is input vector and Y is output vector. The expected value of Random Forest Regression is equal to the average value of i Decision Trees $h(x,\Theta_k)$.

2.5.1. Random Forest Regression Algorithm

Inputs	• A dataset D consisting of n observations $(X_1,y_1),(X_2,y_2),\dots,(X_n,y_n)$, where $X_i = [x_{1i},x_{2i},\dots,x_{ip}]$ represents the feature vector, and y_i is the target variable. • The number of trees M to grow in the forest. • The number of features *m* to consider when splitting a node in each tree (often $\sqrt{p}$ or $\log(p)$.	
Outputs	• A regression model that predicts the target variable $\hat{y}$ for new input data.	
Steps:	**1. Bootstrapping**	For each tree T_b where $b \in \{1,2,\dots,M\}$, generate a bootstrap sample D_b from the original dataset D. This is done by sampling *n* observations from D with replacement.
	2. Growing the Trees	For each tree T_b , grow a decision tree using the following process: **1. Node Splitting:** At each node, randomly select mmm features from the total p features in the dataset. Evaluate all possible splits across these mmm features. Select the feature x_j and split point s that minimizes the impurity measure (such as variance) in the target variable y. Split the node into two child nodes based on the best split (x_j,s) . **2. Stopping Criteria:**

		Continue splitting nodes until a stopping criterion is met, such as a maximum depth, minimum number of samples in a node, or no further improvement in impurity.
	3. Prediction for Each Tree	For a new input vector $X = [x_1, x_2, \dots, x_p]$, pass X down each tree T_b in the forest to obtain a predicted value $\hat{y}_b$.
	4. Aggregating the Predictions	The final prediction $\hat{y}$ is obtained by averaging the predictions from all trees: $\hat{y} = \frac{1}{M}\sum_{b=1}^{M} \hat{y}_b$ (34) This aggregation reduces the variance of the predictions, leading to a more stable and accurate model. **Mathematical Details:** **Node Splitting Criterion:** For each feature x_j in the selected subset mmm, and for every possible split point s , calculate the reduction in variance (or another impurity measure) from splitting the node: Reduction in Variance $= Var(y) - \left(\frac{nL}{n}Var(yL) + \frac{nR}{n}Var(yR)\right)$ (35) Where $Var(y)$ is the variance of the target variable in the parent node. nL and nR are the number of observations in the left and right child nodes, respectively. $Var(yL)$) and $Var(yR)$ are the variances in the left and right child nodes. **Out-of-Bag (OOB) Error Estimation:** During the bootstrapping process, about one-third of the data is not included in each bootstrap sample. This data is called Out-of-Bag (OOB) data. The OOB error is calculated by averaging the prediction error of each tree on the OOB data. This provides an unbiased estimate of the model's performance.

2.5.2. The most important Advantages and Disadvantages of the model

Advantages	Disadvantages
1- By averaging the predictions of multiple decision trees, Random Forest significantly reduces the risk of overfitting compared to using a single decision tree. The randomness introduced in both the data sampling and	1- Training a large number of decision trees and making predictions with them can be computationally expensive, especially with large datasets and high numbers of trees. This can lead to slower model training and

feature selection helps in achieving this robustness.	prediction times.
2- Random Forest can effectively handle datasets with a large number of features (high dimensionality). It automatically selects the most important features during the training process, making it well-suited for complex datasets.	2- Random Forest models are less interpretable compared to single decision trees. Understanding the overall model and how individual decisions are made can be challenging due to the ensemble nature of the method.
3- Random Forest generally provides high accuracy and good performance for a wide range of regression tasks. The ensemble approach ensures that the model is less sensitive to noise and anomalies in the data.	3- Since Random Forest stores multiple decision trees, it can require a significant amount of memory, especially when dealing with large datasets and many trees.
4- It provides an internal mechanism to rank the importance of different features. This is useful for understanding the model and for feature selection in other machine learning tasks.	4- If the dataset is highly unbalanced, Random Forest can produce biased predictions towards the majority class. Special techniques like balancing the data or using weighted trees may be needed to address this issue.
5- Random Forest can handle both categorical and continuous data. It is versatile and can be used for a variety of applications in regression and classification.	5- While Random Forest is generally robust, excessive noise in the dataset can still lead to less accurate predictions, particularly if the noise is present in a significant portion of the data.
6- Unlike many machine learning algorithms, Random Forest does not require the data to be scaled or normalized, making it easier to apply to raw datasets.	6- The resulting Random Forest model can be quite large, making it difficult to deploy in resource-constrained environments or when storage and memory are limited.

Table 5. Features of Random Forest Regression: Advantages and Disadvantages

2.6.Support Vector Regression (SVR)

The mathematical relationship representing the SVR model is as follows:
For the linear case:

$$y = w \cdot x + b \tag{36}$$

Where:

- *y* is the target value.
- w is the weight (coefficients) for features.
- x is the feature vector.
- *b* is the bias term.

For the non-linear case, can use a kernel function to express the relationship in a more complex form. For instance, when using a Radial Basis Function (RBF) kernel, the relationship may look like this:

$$y = \sum_{i=1}^{N} \alpha_i K(x_i, x) + b \tag{37}$$

Where:

- y is the target value.
- α_i are the support vector coefficients.
- $K(x_i, x)$ is the kernel function that assesses similarity between data points.
- b is the bias term.

an analytical technique to investigate the relationship between one or more predictor variables and a real-valued (continuous) [15].
The kernel function *K* introduces non-linearity. This algorithm aims to make the best decision boundary or line that can segregate − space into classes [16]. The supervised learning technique, Support Vector Regression, is used to predict discrete values [17].
The choice of kernel function and the coefficients ($\alpha_i$and b) are adjusted during the training and optimization process to achieve the best possible fit with the available data.

2.6.1. Support Vector Regression (SVR) Algorithm

<table>
<tr><td>Inputs</td><td colspan="2">• A dataset $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, where $X_i \in R^p$ represents the feature vector and $y_i \in R$ is the target variable.
• A kernel function $K(x_i, x_j)$ such as linear, polynomial, or RBF (Radial Basis Function).
• Regularization parameter C that controls the trade-off between achieving a low error on the training data and minimizing model complexity.
• Epsilon $\boldsymbol{\epsilon}$, which defines a margin of tolerance within which no penalty is given to errors.</td></tr>
<tr><td>Outputs</td><td colspan="2">A regression function $f(x)$ that predicts the target variable y for new input data.</td></tr>
<tr><td rowspan="2">Steps:</td><td>1. Objective Function</td><td>The goal of SVR is to find a function $f(x)$ that has at most ϵ deviation from the actual target values y_i for all training data, while keeping the model as flat (simple) as possible.
The function $f(x)$ can be defined as:
$f(x) = w^T \phi(x) + b$ (38)
Where:
w is the weight vector.
$\phi(x)$ is the feature mapping to a higher-dimensional space.
b is the bias term.</td></tr>
<tr><td>2. Minimization Problem</td><td>The optimization problem for SVR is formulated as:
$\min_{w,b,\xi_i,\xi_i^*} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*)$ (39)
subject to:
$y_i - (w^T\phi(x_i) + b) \le \boldsymbol{\epsilon} + \xi_i$ (40)
$(w^T\phi(x_i) + b) - y_i \le \boldsymbol{\epsilon} + \xi_i^*$ (41)
$\xi_i , \xi_i^* \ge 0$ (42)
Here:
ξ_i and ξ_i^* are slack variables that represent the amount by which predictions are allowed to deviate from the ϵ\epsilon$\epsilon$ margin.</td></tr>
</table>

		The first term $\frac{1}{2}\|w\|^2$ represents the model's complexity (flatness of the function), and the second term $C\sum_{i=1}^{n}(\xi_i + \xi_i^*)$ represents the penalty for errors that are outside the ϵ margin.
	3. Dual Formulation	To solve the optimization problem, we typically use the dual formulation. The Lagrangian is introduced, leading to the dual problem: $\min_{\alpha_i,\alpha_i^*} \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)K(x_i, x_j) +$ $\epsilon\sum_{i=1}^{n}(\alpha_i - \alpha_i^*) - \sum_{i=1}^{n} y_i(\alpha_i - \alpha_i^*)$ (43) subject to: $\sum_{i=1}^{n}(\alpha_i - \alpha_i^*) = 0$ (44) $0 < \alpha_i, \alpha_i^* < C$ (45) Where α_i and α_i^* are Lagrange multipliers, and $K(x_i, x_j) = \phi(x_i)^T\phi(x_j)$ is the kernel function that allows computation in the high-dimensional feature space without explicitly mapping the data to that space.
	4. Solution to the Dual Problem	Once the dual problem is solved, the weight vector www can be expressed as: $w = \sum_{i=1}^{n}(\alpha_i - \alpha_i^*)\phi(x_i)$ (46) The bias term b can be computed using the support vectors (data points where α_i or α_i^* is non-zero): $b = y_k - \sum_{i=1}^{n}(\alpha_i - \alpha_i^*)K(x_i, x_k)$ (47) Where x_k is any support vector.
	5. Prediction	The final regression function $f(X)$ is used to predict new data points: $f(x) = \sum_{i=1}^{n}(\alpha_i - \alpha_i^*)K(x_i, x) + b$ (48) The prediction is essentially a weighted sum of the kernel functions applied to the support vectors and the input data point x, plus the bias term b.

2.6.2. The most important Advantages and Disadvantages of the model

Advantages	Disadvantages
1- SVR employs regularization (controlled by the parameter C), which helps in controlling the complexity of the model and prevents overfitting, especially in high-dimensional spaces. 2- SVR can handle non-linear relationships between features and the target variable using kernel functions. Common kernels include linear, polynomial, and RBF (Radial Basis Function), making SVR versatile for different types of data.	1- SVR requires careful tuning of Hyperparameter such as C , ϵ , and the kernel parameters (e.g., the gamma parameter in the RBF kernel). Poor choices can lead to suboptimal performance. 2- SVR can be computationally expensive, especially with large datasets, as the complexity scales with the number of training examples. The training process involves solving a quadratic programming problem, which can be slow for large

	datasets.
3- Only a subset of the training data, known as support vectors, is used to define the decision boundary. This sparsity can make the model more efficient in terms of memory and computation, especially for large datasets.	3- The performance of SVR heavily depends on the choice of the kernel function and its parameters. If the wrong kernel is chosen, the model may not capture the underlying data distribution effectively.
4- SVR performs well in high-dimensional spaces, making it suitable for datasets with a large number of features. The kernel trick allows for efficient computation in these spaces without explicitly computing the high-dimensional mappings.	4- While the model's decision boundary is defined by support vectors, interpreting the overall model, especially when non-linear kernels are used, can be challenging. This makes SVR less transparent compared to simpler models like linear regression.
5- The regularization parameter C and the ϵ margin allow for fine-tuning the model's complexity and the degree of tolerance to errors. This provides flexibility in managing the trade-off between bias and variance.	5- SVR is sensitive to the scale of the data, often requiring feature scaling or normalization before training. This adds an extra preprocessing step, which may not be needed for other regression techniques.
6- The ϵ-insensitive loss function provides a margin of tolerance around the predictions, which can be interpreted as a measure of confidence in the predictions, useful in certain applications where prediction intervals are important.	6- Although SVR uses only the support vectors for prediction, storing and processing these vectors can still require significant memory, particularly when the number of support vectors is large.

Table 6. Features of Support Vector Regression (SVR): Advantages and Disadvantages

Some special Regression models in machine learning

2.7.Time Series Regression(TSR)

The TSR It is one of the statistical methods used to predict time series data to arrive at future data based on historical data records. It is called dynamic autoregressive

$$y_t = b_0 + \sum_{j=1}^{k} b_j x_{jt} + u_t \quad (49)$$

Time series analysis is typically suitable for investigations on relatively direct and short-term effects of exposures [18].

2.7.1. Time Series Regression (TSR) Algorithm

Inputs	• A time series dataset $\{y_1, y_2, \dots, y_t\}$ where y_t is the observed value at time t. • Lagged values of the series (past observations) and possibly other explanatory variables x_{jt} where j indexes the explanatory variables, and

	t indexes time.	
Outputs	• A regression model that predicts future values $y_t + hy$ for h steps ahead based on past values and explanatory variables.	
Steps:	**1. Regression Equation**	The basic TSR model can be written as: $y_t = b_0 + \sum_{j=1}^{k} b_j x_{jt} + u_t$ (50) where: y_t is the dependent variable (time series value) at time t. b_0 is the intercept term. x_{jt} represents the $j-th$ explanatory variable at time t, which could include past values of y_t (i.e., lagged values such as $y_{t-1}, y_{t-2}, \dots, y_{t-p}$ b_j are the coefficients associated with each explanatory variable. u_t is the error term or residual, assumed to be a white noise process with $E(u_t) = 0$ and $Var(u_t) = \sigma^2$.
	2. Autoregressive Component	In the autoregressive (AR) part of the model, the current value of the series y_t depends on its previous values (lags): $y_t = b_0 + \sum_{j=1}^{p} \phi_j y_t + u_t$ (51) where: p is the order of the autoregressive model, indicating how many past values of y_t are used in the regression. ϕ_j are the autoregressive coefficients.
	3. General Model (Including Exogenous Variables)	If we include exogenous variables $X_t = [x_{1t}, x_{2t}, \dots, x_{kt}]$, the model becomes: $y_t = b_0 + \sum_{j=1}^{p} \phi_j y_t + \sum_{i=1}^{k} b_i x_{it} + u_t$ (52) This model is sometimes referred to as an ARX (Autoregressive with Exogenous inputs) model.
	4. Estimation of Coefficients	The coefficients b_j and ϕ_j are estimated using ordinary least squares (OLS) or other methods such as maximum likelihood estimation (MLE). The objective is to minimize the sum of squared residuals: $\min_{b_0 b_j \phi_j} \sum_{t=1}^{T} (y_t - \hat{y}_t)^2$ (53) where $\hat{y}_t$ is the predicted value of y_t based on the model.
	5. Prediction	Once the model is estimated, future values $y_{T+1}, y_{T+2}, \dots, y_{T+h}$ can be predicted using the estimated coefficients and past observed data: $\hat{y}_{T+h} = \hat{b}_0 + \sum_{j=1}^{p} \hat{\phi}_j \, y_{T+h-j} + \sum_{i=1}^{k} \hat{b}_i x_{i,\mathrm{T+h}}$ (54) where $\hat{b}_0, \hat{\phi}_j, \hat{b}_i$ are the estimated coefficients.

	6. Error Term Analysis	The residuals u_t are analyzed to check for autocorrelation (using the Durbin-Watson statistic or autocorrelation function), heteroscedasticity, and normality. This step is crucial to ensure that the model assumptions are not violated.
	7. Model Diagnostics and Selection	**AIC/BIC:** Model selection criteria such as the Akaike Information Criterion (AIC) or Bayesian Information Criterion (BIC) can be used to select the best model in terms of predictive accuracy. **Cross-Validation:** Time series cross-validation can be used to assess model performance and prevent overfitting.

2.7.2. The most important Advantages and Disadvantages of the model

Advantages	Disadvantages
1- TSR models are specifically designed to capture the temporal dependencies in data, making them highly effective for time series forecasting where past values influence future values.	1- TSR models often assume that the time series is stationary, meaning its statistical properties (mean, variance, autocorrelation) do not change over time. However, many real-world time series are non-stationary, requiring transformations like differencing, which can complicate the modeling process.
2- TSR can incorporate exogenous variables (other relevant features or external factors) in addition to lagged values of the time series itself. This allows for more comprehensive modeling of factors that impact the time series.	2- The performance of TSR heavily depends on the correct specification of the model, including the selection of the appropriate lag order, inclusion of relevant exogenous variables, and proper handling of seasonality. Incorrect specification can lead to poor forecasts.
3- TSR offers flexibility in terms of model structure. It can be purely autoregressive (based solely on past values), include moving average components, or integrate exogenous inputs. This adaptability makes it suitable for a wide range of time series data types.	3- If the model includes multiple lagged variables or exogenous inputs that are highly correlated with each other, multicollinearity can become an issue, leading to unstable estimates and reducing the interpretability of the model.
4- The model parameters in TSR, such as coefficients for lagged values and exogenous variables, are interpretable. This allows for a better understanding of the relationship between past values and future predictions, as well as the impact of external variables.	4- Adding too many lagged variables or exogenous inputs can lead to overfitting, where the model performs well on the training data but poorly on unseen data. This is particularly a concern when the dataset is small.
5- When properly tuned and validated, TSR models can achieve high predictive accuracy,	5- Traditional TSR models like autoregressive models are linear in nature.

particularly for time series with strong autoregressive properties. This makes them valuable for tasks like demand forecasting, financial time series prediction, and economic forecasting. 6- TSR is grounded in well-established statistical methods, making it a robust approach with strong theoretical backing. It provides statistical measures like confidence intervals, which help in assessing the uncertainty in predictions.	They may struggle to capture complex non-linear relationships in the data unless combined with non-linear methods, which can increase the model's complexity. 6- As the number of lagged terms and exogenous variables increases, the model can become computationally intensive, particularly for large datasets or when using higher-order models. 7- Time series data often have missing values, and TSR models may require additional preprocessing to handle these gaps, such as imputation methods or interpolation, which can introduce bias or errors if not done carefully.

Table 7. Features of Time Series Regression (TSR): Advantages and Disadvantages

2.8. Spatial Linear Regression(SLR)

Spatial linear regression models may be viewed as generalization of standard linear regression models such that spatial autocorrelation is allowed and accounted for explicitly by spatial models [19].

$$y_i = b_0 + \sum_{i=1}^{k} b_i x_i + \epsilon \quad (54)$$

Spatial regression models, typically with a linear additive specification, in which the relationship among areal units is specified exogenously using weights that mimics the spatial structure and the spatial interaction pattern [20].

2.8.1. Spatial Linear Regression (SLR) Algorithm

Inputs	• Dependent Variable y_iThe response variable observed at location iii. • Explanatory Variables x_{ij} : The independent variables (or covariates) associated with location i . • Spatial Weight Matrix W: A matrix representing the spatial relationships between different locations. The elements w_{ij} indicate the degree of spatial influence between locations iii and j.	
Outputs	A regression model that incorporates both the covariates and spatial dependencies to predict the dependent variable y .	
Steps:	**1. Basic Linear Regression Model**	The standard linear regression model without spatial considerations is: $y_i = b_0 + \sum_{i=1}^{k} b_j x_{ij} + \epsilon_i$ (55) Where: y_i is the dependent variable at location i. b_0 is the intercept term.

		b_j are the coefficients associated with the explanatory variables x_{ij}. ϵ_i is the error term (residual) at location iii, assumed to be independent and identically distributed $(i.i.d.)$ in standard regression.
	2. Incorporating Spatial Autocorrelation	In SLR, we account for spatial autocorrelation by incorporating the spatial structure into the model. This can be done in several ways, including: **Spatial Lag Model (SLM):** $y_i = \rho \sum_j w_{ij} y_j + b_0 + \sum_{j=1}^{k} b_j w_{ij}$ (56) Here, ρ is the spatial autoregressive coefficient, and $\sum_j w_{ij} y_j$ represents the spatially lagged dependent variable. This model assumes that the dependent variable at a location is influenced by the dependent variable at neighboring locations. **Spatial Error Model (SEM):** $y_i = b_0 + \sum_{j=1}^{k} b_j x_{ij} + u_i$ (57) Where the error term w_{ij} follows a spatial autoregressive process: $u_i = \lambda \sum_j w_{ij} u_j + \epsilon_i$ (58) In this model, spatial autocorrelation is captured in the error term, with λ being the spatial error coefficient.
	3. Compute the Loss	The spatial weight matrix W represents the spatial structure of the data. Elements w_{ij} define the spatial relationship between locations iii and j . For example, w_{ij} could be 1 if locations i and j are neighbors, and 0 otherwise. W is typically row-normalized so that the weights sum to 1 for each location.
	4. Estimation of Parameters	The parameters β_j , ρ , and λ are typically estimated using Maximum Likelihood Estimation (MLE) or Generalized Method of Moments (GMM). The likelihood function accounts for the spatial dependencies introduced by the spatial lag or spatial error components. For the Spatial Lag Model (SLM), the log-likelihood function is: $\log L(\beta, \rho) = -\frac{N}{2} \log 2\pi\sigma^2 + \log\lvert I - \rho W\rvert - \frac{1}{2\sigma^2} e'e$ (59) where $e = y - \rho W y - X\beta$ is the vector of residuals.
	5. Model Diagnostics	After estimating the parameters, it's essential to check the residuals for any remaining spatial autocorrelation using tests like Moran's I. If significant spatial autocorrelation remains, the model may need further refinement.
	6. Prediction	Once the model is fitted, predictions for the dependent variable at each location i can be made using the estimated coefficients and the spatial structure:

		$\hat{y}_i = \hat{\rho} \sum_j w_{ij} \hat{y}_j + \hat{b}_0 + \sum_{j=1}^{k} \hat{b}_j x_{ij}$ (60) The spatial lag model incorporates the influence of neighboring locations directly into the prediction.	

2.8.2. The most important Advantages and Disadvantages of the model

Advantages	Disadvantages
1- SLR explicitly incorporates spatial relationships between observations, which is essential for modeling geographic or spatial data where nearby locations may influence each other. This leads to more accurate and realistic models compared to traditional linear regression.	1- Specifying an SLR model requires careful consideration of the spatial weight matrix and the form of spatial autocorrelation. Incorrect specification of these components can lead to biased or inefficient estimates.
2- By including spatial dependencies, SLR often provides better predictive performance for spatial data than non-spatial models, especially in cases where spatial autocorrelation is strong.	2- Estimating the parameters of an SLR model, especially with large datasets or complex spatial relationships, can be computationally demanding. This is due to the need to invert large matrices and solve complex likelihood functions.
3- SLR can be adapted to different types of spatial data structures. For instance, it can include spatial lag models or spatial error models, depending on whether the spatial dependency is in the dependent variable or in the error term.	3- SLR models often assume that the spatial relationship is stationary, meaning that the spatial dependency is consistent across the study area. However, in many real-world situations, spatial relationships can vary, requiring more sophisticated modeling techniques.
4- The spatial parameters (e.g., spatial lag coefficient ρ or spatial error coefficient λ) provide insights into the strength and direction of spatial dependencies, making it easier to understand the spatial processes driving the data.	4- The inclusion of spatial dependencies, especially in spatial lag models, can make the interpretation of the regression coefficients more complex, as the effect of explanatory variables is mediated by spatial interactions.
5- Like standard linear regression, SLR allows the incorporation of multiple explanatory variables (exogenous factors), enabling a comprehensive analysis that includes both spatial and non-spatial factors.	5- The choice of the spatial weight matrix W (which defines how spatial units influence each other) is crucial. Different choices of W can lead to different model results, making the model sensitive to the specification of spatial relationships.
6- SLR models come with diagnostic tools (such as Moran's I for residuals) that help detect and correct for any remaining spatial autocorrelation, leading to more robust and reliable models.	6- SLR models may struggle to handle non-stationarity (where the relationship between variables changes across space) and heteroscedasticity (where the variance of errors varies across locations), which are common in spatial data.
	7- Due to the inclusion of spatial terms and the

	complexity of the model, there is a risk of overfitting, particularly when the model is applied to small datasets or when too many spatial lags or variables are included.

Table 8. Features of Spatial Linear Regression (SLR): Advantages and Disadvantage

3. Conclusion

This review aids in understanding the strengths and weaknesses of various machine learning models. The choice of the most suitable model depends on the specific requirements of the predictive analytics task at hand. Future work involves exploring ensemble methods or hybrid approaches regression for enhanced predictive accuracy.

References

[1] A. Gupta, S. Akshay and G. Amita , "Review of regression analysis models," *International Journal of Engineering Research & Technology (IJERT),* no. 6(08), pp. 58-61, 2017.

[2] S. Hrvoje, "Regression," 8 December 2023. [Online]. Available: https://graphite-note.com/a-comprehensive-guide-to-regression-analysis-in-machine-learning/.

[3] Z. A. Abdalkareem and A. K. Bayda , "Comparison Different Estimation Methods for the Parameters of Non-Linear Regression," *Iraqi Journal of Science,* pp. 1662-1680, 30 4 2022.

[4] A. Gupta, A. Sharma and A. Goel, "Review of Regression Analysis Models," *International Journal of Engineering Research & Technology (IJERT),* pp. 58-61, 8 August 2017.

[5] K. H. Zou, K. Tuncali and S. . G. Silverman, , "Correlation and simple linear regression," *Radiology,* p. 227:617–628, 2003.

[6] M. Tranmer and M. Elliot, "Multiple linear regression," *The Cathie Marsh Centre for Census and Survey Research (CCSR),* pp. 1-5, 5 5 2008.

[7] E. Ostertagová, "Modelling using polynomial regression," *Procedia engineering, 48,* pp. 500-506, 2012.

[8] E. Ostertagová, "Modelling using polynomial regression," *Procedia engineering 48,* pp. 500-506, 2012.

[9] S. A., G. Hommel and M. Blettner, "Linear Regression Analysis," in *Part 14 of a Series on Evaluation of Scientific Publications.*, vol. 107(44), Deutsches Ärzteblatt International., 2010, pp. 776-782.

[10] D. F. e. SPECHT, "A general regression neural network," *EEE transactions on neural networks,* pp. 568-576, 1991.

[11] A. F. Alnuaimi and . T. H. Albaldawi, "An overview of machine learning classification techniques.," in *In BIO Web of Conferences*, 2024.

[12] G. Zhang, "Neural networks for time-series forecasting. In: Handbook of Natural Computing," *Springer, Berlin, Heidelberg.*, p. 461–477, 2012.

[13] L. Breiman, Machine learning, 45 ed., 2001, pp. 5-32.

[14] B. e. a. DAI " ،Statistical model optimized random forest regression model for concrete dam deformation monitoring "،*tructural Control and Health Monitoring* ،p. e2170 .2018 6 25 ،

[15] Z. Fan and J. O. Lauren, "Support vector regression," in *Machine Learning Methods and Applications to Brain Disorders*, Academic Press, 2020, pp. 123-140.

[16] F. H. Fadel and F. B. Suhad , "A Comparative Study for Supervised Learning Algorithms to Analyze Sentiment Tweets," *Iraqi Journal of Science*, pp. 2712-2724, 30 6 2022.

[17] R. A. Redha and A. Y. Suhad, "Crude Oil Price Forecasts Using Support Vector Regressio," *Iraqi Journal of Science*, pp. 2512-2520, 30 5 2023.

[18] C. Imai, B. Armstrong , Z. Chalabi, P. Mangtani and . M. Hashizume, "Time series regression model for infectious disease and weather," *Environmental research, 142*, pp. 319-327, 2015.

[19] J. DURBIN, "Estimation of parameters in time-series regression models," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 1, pp. 139-153, 1960.

[20] G. CHI and J. ZHU, "Spatial regression models for demographic analysis," *Population Research and Policy Review*, pp. 17-42, 2008.