CHAPTER II

LITERATURE REVIEW

This chapter presents the mathematical and statistical modeling background, along with the statistical measures essential for evaluating mathematical and statistical models. In addition, relevant prior studies are reviewed to inform the research process and support the achievement of more robust and effective results.

2.1 History and Development of Linear Modeling

The development of modern statistical modeling follows a smooth progression from Gauss's original concept of simple least squares to contemporary generalized linear modeling frameworks. This evolution, as outlined by de Jong and Heller (2008), encompasses several key stages that progressively increase in mathematical sophistication and practical applicability.

2.1.1 Simple Linear Modeling

The foundational approach aims to explain an observed response variable y through a single explanatory variable x. In the statistical literature, the response variable y is alternatively referred to as the dependent variable, outcome variable, or (in econometrics) endogenous variable. Similarly, the explanatory variable x has various designations including covariate, independent variable, predictor, driver, risk factor, exogenous variable, regressor, or simply the "x" variable. When x represents categorical data, it is termed a factor.

The simple linear model takes the form:

$$y_i = \beta_0 + \beta_1 x_{i1} + \epsilon_i, \quad i = 1, 2, \ldots, n,$$

where β_0 and β_1 are parameters to be estimated, x_{i1} represents the single explanatory variable for observation i, and ϵ_i represents the random error term.

2.1.2 Multiple Linear Modeling

This framework extends simple least squares by incorporating multiple explanatory variables. Here, $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ik})^T$ contains k explanatory variables for each observation i, with their combination serving to explain the response y_i (de Jong and Heller, 2008).

The multiple linear model is expressed as:

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_k x_{ik} + \epsilon_i,$$

or equivalently:

$$y_i = \mathbf{x}_i^T \boldsymbol{\beta} + \epsilon_i,$$

where $\mathbf{x}_i = (1, x_{i1}, x_{i2}, \ldots, x_{ik})^T$ is the $(k+1) \times 1$ covariate vector for observation i, and $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_k)^T$ is the $(k+1) \times 1$ parameter vector.

In matrix notation for all n observations:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon},$$

where $\mathbf{y} = (y_1, y_2, \ldots, y_n)^T$ is the $n \times 1$ response vector, $\mathbf{X}$ is the $n \times (k+1)$ design matrix with the ith row being $\mathbf{x}_i^T = (1, x_{i1}, x_{i2}, \ldots, x_{ik})$, and $\boldsymbol{\epsilon} = (\epsilon_1, \epsilon_2, \ldots, \epsilon_n)^T$ is the $n \times 1$ error vector.

2.1.3 Response Transformation

A natural extension involves replacing the response variable y with a transformation $g(y)$. In this approach, the aim is to model the transformed response $g(y_i)$ in terms of the observed explanatory variables in $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ik})^T$:

$$g(y_i) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_k x_{ik} + \epsilon_i,$$

or equivalently:

$$g(y_i) = \mathbf{x}_i^T \boldsymbol{\beta} + \epsilon_i.$$

Common transformations include the logarithm $g(y) = \log(y)$ and logit $g(y) = \log\left(\frac{y}{1-y}\right)$ functions. For interpretability and mathematical tractability, the transformation function g is constrained to be monotonic.

2.1.4 Classical Linear Modeling

A more conceptually sophisticated development replaces the response y_i with its expected value $\mathbb{E}[y_i]$, or more precisely, $\mathbb{E}[y_i|\mathbf{x}_i]$. This paradigm shift enables modeling of the statistical average of y in terms of the predictors $\mathbf{x}_i$:

$$\mathbb{E}[y_i|\mathbf{x}_i] = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_k x_{ik},$$

or equivalently:

$$\mathbb{E}[y_i|\mathbf{x}_i] = \mathbf{x}_i^T \boldsymbol{\beta}.$$

This modeling perspective focuses on the conditional expectation of the response rather than individual observations, thereby accounting for the inherent variability present in real data (de Jong and Heller, 2008).

However, the classical linear model relies on the assumptions of normality and constant variance, which may be restrictive when dealing with non-Gaussian outcomes such as binary responses, counts, or proportions. These limitations motivate a broader modeling framework capable of accommodating different distributional structures while preserving interpretability and analytical coherence.

The formal development of this extended framework is presented in the following section.

2.2 Generalized Linear Model (GLM)

A generalized linear model (GLM) is a flexible statistical framework that extends linear regression to accommodate various types of response variables and their associated probability distributions. It provides a way to model the relationship between a set of predictors and a response variable while accounting for non-normal and non-continuous data. GLMs allow for the use of different link functions to relate the predictors to the response, making them suitable for analyzing a wide range of data types, including binary, count, and categorical outcomes (Wood, 2017). The flexibility of GLMs makes them widely used in fields such as medicine, social sciences, and economics, where complex data structures and diverse response variables are encountered

The generalized linear models (GLMs) differ from traditional linear models in that they do not assume the normal distribution of the response variable Y_i and do not strictly require $\mathbb{E}[Y] = X\beta$. Instead, GLMs offer more flexibility by allowing the choice of distribution for Y and a corresponding link function g such that $g(\mathbb{E}[Y]) = \eta = X\beta$. These generalizations are based on three main components:

2.2.1 Random Component

The distribution of the response variable for each of the n independently sampled observations is determined by a random component. This random component defines the conditional distribution based on the values of the explanatory variables included in the model. The distribution is selected from the exponential family and can belong to various families, including Gaussian (normal), Binomial, Poisson, Gamma, or Inverse-Gaussian distributions.

The exponential dispersion family represents a class of probability density functions characterized by the following form:

$$f_y(y;\theta,\phi) = \exp\left(\frac{y\theta - b(\theta)}{a(\phi)} + c(y;\phi)\right). \tag{2.1}$$

In this expression, $\phi > 0$ denotes the dispersion parameter, $\theta \in \Theta$ represents the param-

eter specific to the distribution, and $\Theta \subset \mathbb{R}$ is an open set that includes θ. The function a allows for the incorporation of weights in the distribution, although generally $a(\phi) = \phi$. Meanwhile, $b : \Theta \to \mathbb{R}$ corresponds to the cumulant function, and c is a normalization factor that does not depend on θ.

2.2.2 Systematic Component

A linear predictor of a GLM is a linear function of the i-th regressors:

$$\eta_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \ldots + \beta_k X_{ik},$$

where the β_j are the model parameters associated with the regressors X_{ij} to be estimated.

For responses in the exponential family with cumulant function $b(\theta)$,

$$\mu_i = \mathbb{E}[Y_i] = b'(\theta_i), \qquad \text{Var}(Y_i) = \phi\, b''(\theta_i) = \phi V(\mu_i).$$

2.2.3 The Link Function

The GLM incorporates a smooth and invertible link function denoted by $g(\cdot)$. This function relates the mean $\mu_i = \mathbb{E}[Y_i]$ to the linear predictor:

$$g(\mu_i) = \eta_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \ldots + \beta_k X_{ik}. \tag{2.2}$$

For example, we might require that the mean must be strictly positive (as in claim counts and severity). In such cases, link functions that only produce positive values might be more suitable. The following link functions are commonly considered:
where $\Phi(\cdot)$ is the standard normal cumulative distribution function.

Due to the invertibility of the link function, an alternative expression can be used:

$$\mu_i = g^{-1}(\eta_i) = g^{-1}(\beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \ldots + \beta_k X_{ik}), i = 1, 2, \ldots, n.$$

This representation allows GLMs to be considered either as a linear model for a transfor-

Table 2.1 Common link functions in generalized linear models.

Link function name	Link function formula
Identity	$g(\mu) = \mu$
Reciprocal	$g(\mu) = \mu^{-1}$
Log	$g(\mu) = \log(\mu)$
Complementary log-log	$g(\mu) = \log(-\log(1-\mu))$
Logit	$g(\mu) = \log(\mu/(1-\mu))$
Probit	$g(\mu) = \Phi^{-1}(\mu)$

mation of the expected response or as a nonlinear regression model for the response. The inverse function $g^{-1}(\cdot)$ is also referred to as the mean function. It is worth noting that the identity link function simply returns its argument unchanged, so $\eta_i = g(\mu_i) = \mu_i$, resulting in $\eta_i = g^{-1}(\mu_i) = \mu_i$.

GLMs are applied to data using the maximum likelihood method, which allows for the estimation of both the regression coefficients and the estimated asymptotic standard errors of the coefficients.

2.2.4 Parameter Estimation

The parameters of a generalized linear model are typically estimated using the maximum likelihood estimation (MLE) method. Given n independent observations $(y_1, y_2, \ldots, y_n)$ with corresponding covariate vectors $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ik})^T$ for $i = 1, 2, \ldots, n$, the likelihood function is constructed based on the exponential dispersion family:

$$L(\boldsymbol{\beta}, \phi; \mathbf{y}) = \prod_{i=1}^{n} f_{y_i}(y_i; \theta_i, \phi) = \prod_{i=1}^{n} \exp\left(\frac{y_i\theta_i - b(\theta_i)}{a(\phi)} + c(y_i; \phi)\right).$$

The log-likelihood function is then given by:

$$\ell(\boldsymbol{\beta}, \phi; \mathbf{y}) = \sum_{i=1}^{n} \left[\frac{y_i\theta_i - b(\theta_i)}{a(\phi)} + c(y_i; \phi)\right],$$

where $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_k)^T$ is the vector of regression parameters.

Since $\mu_i = \mathbb{E}[Y_i] = b'(\theta_i)$ and $\eta_i = g(\mu_i) = \mathbf{x}_i^T\boldsymbol{\beta}$, the relationship between θ_i

and $\boldsymbol{\beta}$ is established through the link function. The maximum likelihood estimates $\hat{\boldsymbol{\beta}}$ are obtained by solving the score equations:

$$\mathbf{U}(\boldsymbol{\beta}) = \frac{\partial \ell(\boldsymbol{\beta}, \phi; \mathbf{y})}{\partial \boldsymbol{\beta}} = \mathbf{0},$$

which can be expressed as:

$$\mathbf{U}(\boldsymbol{\beta}) = \sum_{i=1}^{n} \frac{(y_i - \mu_i)}{a(\phi)V(\mu_i)} \cdot \frac{\partial \mu_i}{\partial \eta_i} \cdot \mathbf{x}_i = \mathbf{0},$$

where $V(\mu_i) = b''(\theta_i)$ is the variance function and $\frac{\partial \mu_i}{\partial \eta_i} = \frac{1}{g'(\mu_i)}$ is the derivative of the inverse link function.

Since closed-form solutions generally do not exist for GLMs, the maximum likelihood estimates are computed iteratively using the Fisher scoring algorithm or the iteratively reweighted least squares (IRLS) method. The Fisher scoring algorithm updates the parameter estimates according to:

$$\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} + [\mathcal{I}(\boldsymbol{\beta}^{(t)})]^{-1} \mathbf{U}(\boldsymbol{\beta}^{(t)}),$$

where $\mathcal{I}(\boldsymbol{\beta})$ is the Fisher information matrix defined as:

$$\mathcal{I}(\boldsymbol{\beta}) = \mathbb{E}\left[-\frac{\partial^2 \ell(\boldsymbol{\beta}, \phi; \mathbf{y})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^T}\right] = \sum_{i=1}^{n} \frac{1}{a(\phi)V(\mu_i)} \left(\frac{\partial \mu_i}{\partial \eta_i}\right)^2 \mathbf{x}_i \mathbf{x}_i^T.$$

Equivalently, the IRLS algorithm can be expressed as a weighted least squares problem at each iteration:

$$\boldsymbol{\beta}^{(t+1)} = (\mathbf{X}^T \mathbf{W}^{(t)} \mathbf{X})^{-1} \mathbf{X}^T \mathbf{W}^{(t)} \mathbf{z}^{(t)},$$

where $\mathbf{X}$ is the $n \times (k+1)$ design matrix, $\mathbf{W}^{(t)}$ is a diagonal weight matrix with elements:

$$w_i^{(t)} = \frac{1}{a(\phi)V(\mu_i^{(t)})} \left(\frac{\partial \mu_i^{(t)}}{\partial \eta_i^{(t)}}\right)^2,$$

and $\mathbf{z}^{(t)}$ is the adjusted dependent variable (working response) with elements:

$$z_i^{(t)} = \eta_i^{(t)} + (y_i - \mu_i^{(t)}) \frac{\partial \eta_i^{(t)}}{\partial \mu_i^{(t)}}.$$

The iteration continues until convergence is achieved, typically when $\|\boldsymbol{\beta}^{(t+1)} - \boldsymbol{\beta}^{(t)}\| < \epsilon$ for some predetermined tolerance level $\epsilon > 0$.

The asymptotic distribution of the maximum likelihood estimator is:

$$\hat{\boldsymbol{\beta}} \sim \mathcal{N}\left(\boldsymbol{\beta}, [\mathcal{I}(\boldsymbol{\beta})]^{-1}\right),$$

which provides the basis for constructing confidence intervals and hypothesis tests for the model parameters.

2.3 Double Generalized Linear Model

The double generalized linear model (DGLM) is a comprehensive and computationally efficient approach that enables the detection of variance effects while accounting for mean effects. It comprises two components: a generalized linear model for the mean and a generalized linear sub-model for the dispersion (residual variance). Unlike a Generalized Linear Model (GLM), the DGLM allows the dispersion to be dependent on its own linear predictor. The dispersion model can incorporate fixed effects for genotype and relevant covariates associated with the trait. Consequently, a DGLM facilitates the simultaneous estimation of the genotype's impact on both the mean value and the residual variance of a quantitative trait. By including genotype as a fixed effect in the model for residual variance, the DGLM enables the assessment of whether genotype significantly influences variability around the mean trait value (Paula, 2013).

To achieve a simultaneous estimation of mean and variance effects, the DGLM employs an iterative process, iteratively refining the models until convergence. This approach incorporates adjustments for estimation uncertainty at each iteration. The DGLMs

are defined by

$$g(\mu_i) = \eta_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \ldots + \beta_k X_{ik}$$

and

$$h(\phi_i) = \lambda_i = \gamma_0 + \gamma_1 Z_{i1} + \gamma_2 Z_{i2} + \ldots + \gamma_m Z_{im},$$

where β_i and γ_i are the model parameters to be estimated,X_{ij} and Z_{ij}, $j = 1, 2, \ldots, k$, contain values of explanatory variables, ϕ_i is the dispersion parameter, and $g(\cdot)$ and $h(\cdot)$ are the link functions. Various methods, such as maximum likelihood estimation and restricted maximum likelihood estimation, can be employed to fit the DGLMs.

2.3.1 Parameter Estimation for Double Generalized Linear Models

The estimation of parameters in Double Generalized Linear Models (DGLMs) involves simultaneously modeling both the mean and dispersion structures. Given the coupled nature of these two components, specialized estimation procedures are required to obtain consistent and efficient parameter estimates.

Maximum Likelihood Estimation For DGLMs, the log-likelihood function based on the exponential dispersion family can be written as:

$$\ell(\boldsymbol{\beta}, \boldsymbol{\gamma}; \mathbf{y}) = \sum_{i=1}^{n} \left[\frac{y_i \theta_i - b(\theta_i)}{a(\phi_i)} + c(y_i; \phi_i) \right],$$

where $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_k)^T$ are the mean model parameters and $\boldsymbol{\gamma} = (\gamma_0, \gamma_1, \ldots, \gamma_k)^T$ are the dispersion model parameters. The relationships $\mu_i = b'(\theta_i)$, $\eta_i = g(\mu_i) = \mathbf{x}_i^T \boldsymbol{\beta}$, and $\lambda_i = h(\phi_i) = \mathbf{z}_i^T \boldsymbol{\gamma}$ establish the connections between the canonical parameters and the linear predictors. The maximum likelihood estimates $(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\gamma}})$ are obtained by solving the system of score equations:

$$\frac{\partial \ell}{\partial \boldsymbol{\beta}} = \mathbf{0} \quad \text{and} \quad \frac{\partial \ell}{\partial \boldsymbol{\gamma}} = \mathbf{0}.$$

However, direct maximization of the joint likelihood presents computational challenges due to the interdependence between $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$. An iterative estimation procedure is employed, alternating between updating the mean and dispersion parameters until convergence (Smyth, 1989; Lee and Nelder, 1996).

Iterative Fitting Algorithm The estimation procedure follows an iterative scheme:

1. Initialize $\phi_i^{(0)}$ for all $i = 1, 2, \ldots, n$ (typically $\phi_i^{(0)} = 1$).

2. At iteration t:

   (a) Fix $\phi_i = \phi_i^{(t-1)}$ and estimate $\boldsymbol{\beta}^{(t)}$ by solving:

   $$\sum_{i=1}^{n} \frac{(y_i - \mu_i)}{a(\phi_i^{(t-1)})V(\mu_i)} \cdot \frac{\partial \mu_i}{\partial \eta_i} \cdot \mathbf{x}_i = \mathbf{0}. \tag{2.3}$$

   (b) Using $\boldsymbol{\beta}^{(t)}$, compute adjusted responses:

   $$d_i^{(t)} = \frac{(y_i - \mu_i^{(t)})^2}{V(\mu_i^{(t)})^2} + \frac{V'(\mu_i^{(t)})}{V(\mu_i^{(t)})}(y_i - \mu_i^{(t)}), \tag{2.4}$$

   where $V'(\mu_i) = \frac{\partial V(\mu_i)}{\partial \mu_i}$.

   (c) Estimate $\boldsymbol{\gamma}^{(t)}$ by fitting a GLM with response $d_i^{(t)}$ and linear predictor $\lambda_i = \mathbf{z}_i^T \boldsymbol{\gamma}$:

   $$\sum_{i=1}^{n} w_i^{(t)} (d_i^{(t)} - a(\phi_i^{(t)})) \frac{\partial \phi_i^{(t)}}{\partial \lambda_i^{(t)}} \mathbf{z}_i = \mathbf{0}, \tag{2.5}$$

   where $w_i^{(t)}$ are appropriate weights.

3. Repeat step 2 until convergence: $\|\boldsymbol{\beta}^{(t)} - \boldsymbol{\beta}^{(t-1)}\| < \epsilon_1$ and $\|\boldsymbol{\gamma}^{(t)} - \boldsymbol{\gamma}^{(t-1)}\| < \epsilon_2$.

Restricted Maximum Likelihood Estimation A critical limitation of maximum likelihood estimation in DGLMs is that the dispersion parameter estimates can be biased, particularly in small samples. This bias arises because ML estimation does not account for the loss of degrees of freedom due to estimating the mean parameters (Patterson and Thompson, 1971; Harville, 1977). Restricted Maximum Likelihood (REML) addresses this

bias by partitioning the likelihood into two components: one that depends on $\boldsymbol{\beta}$ and one that does not. The REML estimator maximizes only the portion of the likelihood that is invariant to $\boldsymbol{\beta}$, thereby providing less biased estimates of the dispersion parameters. The REML log-likelihood for the dispersion parameters can be expressed as

$$\ell_R(\boldsymbol{\gamma}; \mathbf{y}) = \ell(\hat{\boldsymbol{\beta}}, \boldsymbol{\gamma}; \mathbf{y}) - \frac{1}{2} \log |\mathbf{X}^T \mathbf{W} \mathbf{X}|, \tag{2.6}$$

where $\hat{\boldsymbol{\beta}}$ is the ML estimate conditional on $\boldsymbol{\gamma}$, $\mathbf{W}$ is the weight matrix from the mean model, and $|\cdot|$ denotes the determinant. The second term adjusts for the uncertainty in estimating $\boldsymbol{\beta}$. The REML estimates $\hat{\boldsymbol{\gamma}}_R$ are obtained by maximizing $\ell_R(\boldsymbol{\gamma}; \mathbf{y})$:

$$\frac{\partial \ell_R(\boldsymbol{\gamma}; \mathbf{y})}{\partial \boldsymbol{\gamma}} = \mathbf{0}. \tag{2.7}$$

For DGLMs, the REML procedure modifies the iterative algorithm by incorporating the adjustment term in the dispersion model estimation step. Specifically, the adjusted responses in Eq. (2.4) are replaced by REML-adjusted responses that account for the degrees of freedom correction (Smyth and Verbyla, 1999):

$$d_i^{R,(t)} = d_i^{(t)} + \text{tr}\left[(\mathbf{X}^T \mathbf{W}^{(t)} \mathbf{X})^{-1} \frac{\partial (\mathbf{X}^T \mathbf{W}^{(t)} \mathbf{X})}{\partial \phi_i} \right], \tag{2.8}$$

where $\text{tr}(\cdot)$ denotes the trace operator.

Asymptotic Properties Under regularity conditions, the REML estimators have desirable asymptotic properties. As $n \to \infty$:

$$\hat{\boldsymbol{\gamma}}_R \sim \mathcal{N}\left(\boldsymbol{\gamma}, [\mathcal{I}_R(\boldsymbol{\gamma})]^{-1}\right),$$

where $\mathcal{I}_R(\boldsymbol{\gamma})$ is the Fisher information matrix for the dispersion parameters under REML. The REML approach generally provides estimates with smaller mean squared error than ML, particularly for variance components, making it the preferred method for fitting DGLMs in practice (Smyth and Verbyla, 1999; Lee et al., 2006).

The choice between ML and REML estimation depends on the specific application

and sample size. While REML is preferred for dispersion parameter estimation, ML may be more appropriate when comparing models with different fixed effects structures, as REML estimates are not directly comparable across models with different mean structures.

2.4 Generalized Additive Model

A generalized additive model (GAM) is a statistical modeling technique that extends the concept of generalized linear models (GLMs) by allowing for nonlinear relationships between predictors and the response variable. Unlike GLMs, which assume linear relationships, GAMs can capture complex and nonlinear patterns by incorporating smooth functions of the predictors (Wood, 2017). In a GAM, the response variable is still related to the predictors through a specified probability distribution and a link function, similar to GLMs. However, instead of assuming a linear relationship, GAMs employ smoothing functions, such as splines or wavelets, to model nonlinearities and interactions. By allowing for flexible and nonparametric modeling, GAMs can handle complex data structures and capture intricate relationships between predictors and the response. They are particularly useful when dealing with data that exhibit nonlinear patterns, such as time series, spatial data, and interactions between variables. Instead of using the linear form $\Sigma\beta_i X_i$, GAM replaces it with a sum of smooth functions $\Sigma(X_i)$ (Hastie and Tibshirani, 1986). The general structure of the model can be described as follows:

$$g(\mu_i) = A_i\theta + f_1(x_{1i}) + f_2(x_{2i}) + \ldots + f_k(x_{ki}),$$

where $g(\mu_i) = \mathbb{E}(Y_i)$ and response variable Y_i follows a conditional distribution from the exponential family distribution with mean μ_i and scale parameterϕ. A_i is an explanatory variable vector for strictly parametric components, is the equivalent parameter vector, and f_i are smooth functions of non-parametric model variables, x_i for $i = 1, 2, \ldots, k$.

2.4.1 Penalized Estimation

The estimation of smooth functions in Generalized Additive Models requires balancing two competing objectives: achieving good fit to the data while maintaining appropriate smoothness to avoid overfitting. This trade-off is formalized through penalized likelihood estimation, which combines a measure of fit with a penalty for excessive roughness (Wood, 2017).

Penalized Likelihood Framework For a GAM with response variable Y_i following an exponential family distribution, the penalized log-likelihood is given by:

$$\ell_p(\boldsymbol{\theta}, \mathbf{f}) = \ell(\boldsymbol{\theta}, \mathbf{f}) - \frac{1}{2}\sum_{j=1}^{k} \lambda_j \int [f_j''(x)]^2 dx,$$

where $\ell(\boldsymbol{\theta}, \mathbf{f})$ is the log-likelihood function:

$$\ell(\boldsymbol{\theta}, \mathbf{f}) = \sum_{i=1}^{n} \left[\frac{y_i \eta_i - b(\eta_i)}{\phi} + c(y_i; \phi) \right],$$

with $\eta_i = A_i\boldsymbol{\theta} + \sum_{j=1}^{k} f_j(x_{ji})$, and the penalty term $\int [f_j''(x)]^2 dx$ measures the roughness of the j-th smooth function through its second derivative. The parameters $\lambda_j \geq 0$ are smoothing parameters that control the trade-off between fit and smoothness for each smooth function f_j.

Basis Function Representation In practice, each smooth function f_j is represented using a basis function expansion:

$$f_j(x) = \sum_{m=1}^{M_j} b_{jm}(x)\beta_{jm} = \mathbf{b}_j(x)^T \boldsymbol{\beta}_j,$$

where $\{b_{j1}(x), b_{j2}(x), \ldots, b_{jM_j}(x)\}$ are basis functions (typically B-splines or thin plate splines), $\boldsymbol{\beta}_j = (\beta_{j1}, \beta_{j2}, \ldots, \beta_{jM_j})^T$ are the corresponding coefficients, and M_j is the dimension of the basis for the j-th smooth term.

The roughness penalty can then be expressed in matrix form as:

$$\int [f_j''(x)]^2 dx = \boldsymbol{\beta}_j^T \mathbf{S}_j \boldsymbol{\beta}_j,$$

where $\mathbf{S}_j$ is a positive semi-definite penalty matrix with elements:

$$[\mathbf{S}_j]_{mn} = \int b_{jm}''(x) b_{jn}''(x) dx.$$

The complete penalized log-likelihood becomes:

$$\ell_p(\boldsymbol{\theta}, \boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_k) = \ell(\boldsymbol{\theta}, \boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_k) - \frac{1}{2} \sum_{j=1}^{k} \lambda_j \boldsymbol{\beta}_j^T \mathbf{S}_j \boldsymbol{\beta}_j. \tag{2.9}$$

Penalized Iteratively Reweighted Least Squares The penalized maximum likelihood estimates are obtained by maximizing Eq. (2.9). Let $\boldsymbol{\beta} = (\boldsymbol{\theta}^T, \boldsymbol{\beta}_1^T, \ldots, \boldsymbol{\beta}_k^T)^T$ denote the complete parameter vector. The estimation is performed using a penalized iteratively reweighted least squares (P-IRLS) algorithm (Wood, 2017).

At iteration t, given $\boldsymbol{\beta}^{(t)}$, compute $\boldsymbol{b}^{(t)}$ by

$$\boldsymbol{\beta}^{(t+1)} = (\mathbf{X}^T \mathbf{W}^{(t)} \mathbf{X} + \mathbf{S}_\lambda)^{-1} \mathbf{X}^T \mathbf{W}^{(t)} \mathbf{z}^{(t)},$$

where:

- $\mathbf{X}$ is the model matrix with columns corresponding to the parametric terms and basis function evaluations,
- $\mathbf{W}^{(t)}$ is the diagonal weight matrix with elements $w_i^{(t)} = \frac{1}{\phi V(\mu_i^{(t)})} [g'(\mu_i^{(t)})]^{-2}$,
- $\mathbf{z}^{(t)}$ is the working response vector with elements $z_i^{(t)} = \eta_i^{(t)} + (y_i - \mu_i^{(t)}) g'(\mu_i^{(t)})$,
- $\mathbf{S}_\lambda = \text{diag}(\mathbf{0}, \lambda_1 \mathbf{S}_1, \lambda_2 \mathbf{S}_2, \ldots, \lambda_k \mathbf{S}_k)$ is the block-diagonal penalty matrix.

Smoothing Parameter Selection The choice of smoothing parameters $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_k)$ is crucial for achieving the optimal balance between fit and smoothness. Generalized Cross-Validation (GCV) is commonly employed for automatic selection (Wood, 2006):

$$\text{GCV}(\boldsymbol{\lambda}) = \frac{n\mathcal{D}(\boldsymbol{\lambda})}{(n - \text{tr}(\mathbf{A}(\boldsymbol{\lambda})))^2},$$

where $\mathcal{D}(\boldsymbol{\lambda})$ is the deviance and $\mathbf{A}(\boldsymbol{\lambda}) = \mathbf{X}(\mathbf{X}^T\mathbf{W}\mathbf{X} + \mathbf{S}_\lambda)^{-1}\mathbf{X}^T\mathbf{W}$ is the influence matrix. The term $\text{tr}(\mathbf{A}(\boldsymbol{\lambda}))$ represents the effective degrees of freedom.

Alternatively, Restricted Maximum Likelihood (REML) treats the smoothing parameters as variance components (Wood, 2011):

$$\text{REML}(\boldsymbol{\lambda}) = -\frac{1}{2}\left[\log|\mathbf{X}^T\mathbf{W}\mathbf{X} + \mathbf{S}_\lambda| + \log|\mathbf{S}_\lambda^-| + \mathbf{z}^T(\mathbf{W} - \mathbf{W}\mathbf{X}\mathbf{C}\mathbf{X}^T\mathbf{W})\mathbf{z}\right],$$

where $\mathbf{C} = (\mathbf{X}^T\mathbf{W}\mathbf{X} + \mathbf{S}_\lambda)^{-1}$ and $\mathbf{S}_\lambda^-$ denotes the generalized inverse.

The optimal smoothing parameters are obtained by minimizing GCV or maximizing REML:

$$\hat{\boldsymbol{\lambda}} = \arg\min_{\boldsymbol{\lambda}} \text{GCV}(\boldsymbol{\lambda}) \quad \text{or} \quad \hat{\boldsymbol{\lambda}} = \arg\max_{\boldsymbol{\lambda}} \text{REML}(\boldsymbol{\lambda}).$$

Effective Degrees of Freedom The effective degrees of freedom (EDF) provides a measure of the complexity of each smooth term (Hastie and Tibshirani, 1990). For the j-th smooth function:

$$\text{EDF}_j = \text{tr}(\mathbf{F}_j),$$

where $\mathbf{F}_j$ is the submatrix of $\mathbf{A}(\boldsymbol{\lambda})$ corresponding to the j-th smooth term. When $\lambda_j = 0$ (no penalty), $\text{EDF}_j = M_j$ (fully flexible), and as $\lambda_j \to \infty$, $\text{EDF}_j \to 0$ (linear or constant term).

The penalized estimation framework thus provides a principled approach to controlling model complexity, ensuring that GAMs achieve an optimal balance between capturing genuine patterns in the data and avoiding overfitting.

2.5 Methods for Parameter Estimation

In the construction of statistical models, it is often necessary to employ efficient methods for estimating model parameters. This section presents parameter estimation using classical approaches, as well as more recent methods derived from machine learning techniques.

2.5.1 Smoothing Splines

Chouldechova and Hastie (2015) demonstrate for instance the univariate regression condition where we observe data on a single predictor x and a response variable y. On the real line, one assumes that x has values that occur in some compact interval. A choice of $\lambda \geq 0$ and observations $(y_1, x_1), \ldots, (y_n, x_n)$, the smoothing spline estimator, $\hat{f}_\lambda$, is defined by,

$$\hat{f} = \underset{f \in C^2}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \int [f''(x)]^2 dx$$

Although the uncountably infinite parameter space C^2 may make this optimization problem seem challenging at first, it can be demonstrated that the problem can be reduced to a much simpler form. A natural cubic spline with knots at the specific x values is the answer. The smoothing spline objective can be expressed as a finite dimensional problem because it can be demonstrated that the space of such splines is n-dimensional.

2.5.2 Explainable Boosting Machine

The Explainable Boosting Machine (EBM), another interpretability algorithm, is an element of the InterpretML framework. EBM is a transparent model with an emphasis on interpretability and explainability with the intent to achieve accuracy comparable to that of more sophisticated machine learning methods like Random Forest and Boosted Trees. It adopts a Generalized Additive Model (GAM) structure and is represented by the form:

$$g(E[y]) = \beta_0 + \sum f_j(x_j),$$

The link function $g(\cdot)$ adapts the Generalized Additive Model (GAM) to different contexts, such as regression or classification. The Explainable Boosting Machine (EBM) incorporates several notable advancements compared to traditional GAMs. Firstly, EBM utilizes modern machine learning techniques like bagging and gradient boosting to learn each feature function f_j. The boosting procedure is carefully constrained to train on one feature at a time in a round-robin fashion with a very low learning rate.

This approach ensures that the order of features does not impact the results. By cycling through features in a round-robin manner, EBM mitigates the effects of co-linearity and learns the optimal feature function f_j for each feature, revealing how each feature contributes to the model's prediction. Secondly, EBM has the capability to automatically identify and incorporate pairwise interaction terms. These terms capture interactions between pairs of features and are expressed in the following form:

$$g(\mathbb{E}[y]) = \beta_0 + \sum_{j=1}^{p} f_j(x_j) + \sum_{1 \le i < j \le p} f_{ij}(x_i, x_j).$$

The most significant of the primary benefits of EBM is its ease of comprehension. By plotting the corresponding feature function f_j, the contribution of each feature to the final prediction can be visualized and easily understood. The additive model structure of EBM ensures that each feature contributes to the predictions in a modular manner, making it easier to interpret the impact of each feature on the overall prediction.

2.5.3 XGBoosT

Extreme gradient boosting (XGBoost) is an algorithm that combines weak learners, typically regression trees, to create a strong predictive model (Chen and Guestrin, 2016). Boosting is the sequential ensemble of trees, where new trees are built based on residuals from previous trees, focusing on the unexplained variance in the training set. The algorithm adds new trees until reaching the maximum number or when they do not significantly improve predictive power. At the beginning of the algorithm, each observation is predicted as the mean of all response variables with equal weights. These weights are adjusted throughout each iteration, increasing for observations that are poorly predicted

to force the algorithm's attention to them.

The fitting process of XGBoost involves sequentially tuning a series of model parameters to achieve the most accurate model, which sets the maximum number of trees, is tested for multiple values in each tuning iteration. The algorithm returns the optimal combination of input parameters by minimizing the Loss function,

$$\mathcal{L}(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k),$$

where

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \|w\|^2$$

denoted as $\mathcal{L}(\phi)$is the estimated predictive loss (l), provides a second-degree approximation of the residuals. It quantifies the discrepancy between the predicted and actual values. To prevent overfitting and promote simpler models, a penalty function $\Omega(f)$ is applied. This penalty function aims to control the complexity of the model. One specific aspect that is penalized is the number of leaf nodes (T) in a tree, and this is achieved through the Gamma (γ) parameter (Harvri, 2019).

2.5.4 Random Forest

The random forest algorithm, introduced by Breiman (2001), is a powerful ensemble method that improves classification accuracy by combining multiple decision trees. A random forest consists of a collection of tree-based classifiers, denoted as$f(\cdot, \Theta_k) : k \geq 1$, where Θ_k represents independent and identically distributed random vectors. The classification process involves a majority voting scheme, where each tree casts a vote, $f(x, \Theta_k)$, for a test point x, and the class with the most votes is chosen. Here is a step-by-step outline of the random forest algorithm:

1. Input: The algorithm takes a dataset $S = \{(Y^1, X^1), \ldots, (Y^n, X^n)\}$, where Y belongs to the set $\{1, \ldots, J\}$ and $X \in \mathbb{R}^{\text{I}}$ is in p-dimensional space. Additionally, specify the number of trees K, positive integers d (much smaller than p and x).

2. For each k from 1 to K:

(a) Create a bootstrap sample S_k by randomly selecting data points from the training set S.

(b) At each node of the tree, randomly select d input variables from $X_1, \ldots, X_p$. Based on these variables and S_k, determine the best split to partition the data.

(c) Grow the tree until each terminal node contains no more than ℓ training samples. No pruning is performed. Let $f(\cdot, S_k)$ represent the constructed tree.

3. Output: The random forest classifier, denoted as $f_{\text{RandFrst}}(x, S)$, is defined as the class that has the highest sum of votes over K trees, i.e.

$$\underset{1 \le k \le K}{\text{argmax}} \sum_{K} I[f(x, S_k)].$$

In summary, the random forest algorithm combines the predictions of multiple decision trees to make accurate classifications. Each tree is trained on a bootstrap sample and selects random subsets of input variables, resulting in a diverse ensemble of trees. The final classification is determined by majority voting among the individual tree predictions (Shim and Lee, 2011).

2.5.5 Support Vector Regression

Support Vector Regression (SVR), introduced by Vapnik (1995), is a powerful machine learning algorithm based on statistical learning theory that extends the principles of Support Vector Machines to regression problems. SVR aims to find a function that approximates the relationship between input features and continuous output values while maintaining a balance between model complexity and prediction accuracy (Smola and Schölkopf, 2004).

Unlike traditional regression methods that minimize the sum of squared errors, SVR employs an ε-insensitive loss function that ignores errors smaller than a threshold ε. This approach creates a margin of tolerance around the regression function, focusing computational effort on predictions that deviate significantly from observed values. The

algorithm identifies support vectors data points that lie outside or on the boundary of this ε-tube which are critical in determining the regression function.

The SVR optimization problem can be formulated as follows. Given a training dataset $\{(x_i, y_i)\}_{i=1}^n$ where $x_i \in \mathbb{R}^p$ and $y_i \in \mathbb{R}$, SVR seeks to find a function:

$$f(x) = \langle w, \phi(x) \rangle + b,$$

where w is the weight vector, $\phi(x)$ maps input features to a higher-dimensional feature space, and b is the bias term. The optimization objective is:

$$\min_{w,b,\xi,\xi^*} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*),$$

subject to the constraints:

$$y_i - \langle w, \phi(x_i) \rangle - b \leq \varepsilon + \xi_i,$$
$$\langle w, \phi(x_i) \rangle + b - y_i \leq \varepsilon + \xi_i^*,$$
$$\xi_i, \xi_i^* \geq 0,$$

where $C > 0$ is a regularization parameter that controls the trade-off between model flatness and tolerance for deviations larger than ε, and ξ_i, ξ_i^* are slack variables that allow some points to fall outside the ε-tube.

The dual formulation of this optimization problem involves Lagrange multipliers α_i and α_i^*, leading to the prediction function:

$$f(x) = \sum_{i=1}^{n}(\alpha_i - \alpha_i^*)K(x_i, x) + b,$$

where $K(x_i, x) = \langle \phi(x_i), \phi(x) \rangle$ is a kernel function. Common kernel choices include:

- Linear kernel: $K(x_i, x_j) = \langle x_i, x_j \rangle$
- Polynomial kernel: $K(x_i, x_j) = (\gamma\langle x_i, x_j \rangle + r)^d$ (degree d, i.e., the polynomial order)
- Radial Basis Function (RBF) kernel: $K(x_i, x_j) = \exp(-\gamma\|x_i - x_j\|^2)$
- Sigmoid kernel: $K(x_i, x_j) = \tanh(\gamma\langle x_i, x_j \rangle + r)$.

The kernel function allows SVR to capture nonlinear relationships by implicitly mapping data to higher-dimensional spaces without explicitly computing the transformation $\phi(x)$, a technique known as the "kernel trick."

The fitting process of SVR involves tuning several hyperparameters to achieve optimal performance. The key hyperparameters include:

- C: The regularization parameter that balances model complexity and training error
- ε: The width of the ε-insensitive tube
- Kernel-specific parameters: γ (for RBF and polynomial kernels), d (degree for polynomial kernel), r (coefficient for polynomial and sigmoid kernels)

2.6 Model Evaluation

Model evaluation is a critical component in the development and validation of statistical and machine learning models. For Generalized Additive Models (GAM) and Double GAM (DGAM), which estimate both the conditional mean and variance, comprehensive assessment requires metrics that evaluate point prediction accuracy, model complexity, and uncertainty quantification. The following metrics provide a robust framework for evaluating model performance from multiple perspectives.

2.6.1 Point Prediction

Several metrics are employed to evaluate point prediction performance. For all metrics, y_i denotes the observed value, $\hat{y}_i$ denotes the predicted value, and N denotes the total number of observations.

Mean Absolute Error

Mean Absolute Error (MAE) provides a straightforward measure of average prediction error by calculating the mean of absolute differences between predicted and observed values:

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|.$$

Root Mean Squared Error

Root Mean Squared Error ($RMSE$) measures prediction accuracy by taking the square root of the average squared error:

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

A smaller value for either metric signifies more accurate predictions. $RMSE$ is particularly sensitive to large errors, making it well suited for evaluating mean prediction performance in GAM and DGAM models (James et al., 2013).

In contrast, the Mean Absolute Error (MAE) assigns equal weight to all prediction errors regardless of their magnitude, providing a more robust and interpretable measure of average prediction accuracy, especially in the presence of outliers.

Coefficient of Determination

The Coefficient of Determination, or R^2, is a statistical measure that represents the proportion of variance in the dependent variable that is explained by the independent variables in the regression model. It is calculated as:

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2},$$

where SS_{res} is the residual sum of squares, SS_{tot} is the total sum of squares, y_i is the observed value, $\hat{y}_i$ is the predicted value, and $\bar{y}$ is the mean of observed values. The R^2 value ranges from 0 to 1, where a value closer to 1 indicates that the model explains a larger proportion of the variance. For GAM and DGAM, R^2 provides an intuitive measure of how well the smooth functions capture the underlying relationship in the data (James et al., 2013).

2.6.2 Likelihood Based

Akaike Information Criterion

The Akaike Information Criterion, or AIC, is a model selection criterion that balances the goodness of fit with model complexity. It estimates the relative amount of information lost by a given model by dealing with the trade-off between overfitting and underfitting. The AIC is calculated using the equation:

$$AIC = -2\ln(\hat{L}) + 2k,$$

where $\hat{L}$ is the maximum value of the likelihood function for the model, and k is the number of estimated parameters. A smaller AIC value indicates a better model, as it suggests less information loss. This criterion is particularly important for GAM and DGAM, where the selection of smooth functions and the effective degrees of freedom directly affect model complexity (Akaike, 1974).

Negative Log-Likelihood

The Negative Log-Likelihood, or NLL, is a fundamental loss function derived from maximum likelihood estimation that quantifies how well a probabilistic model fits the observed data. For a dataset with N observations, the NLL is defined as:

$$NLL = -\sum_{i=1}^{N} \log p(y_i \mid x_i; \theta),$$

where $p(y_i \mid x_i; \theta)$ represents the predicted probability density of the observed value y_i given the input x_i and model parameters θ. Minimizing NLL is equivalent to maximizing the likelihood of observing the data under the model. For distributional regression models such as DGAM, which estimate both location and scale parameters, NLL serves as the primary training objective and evaluation metric (Hastie and Tibshirani, 1990).

In this thesis, we focus on the Gaussian negative log-likelihood, which corresponds to the NLL obtained under the assumption that the response variable follows a normal distribution.

2.6.3 Probabilistic Forecasts

Continuous Ranked Probability Score

The Continuous Ranked Probability Score, or $CRPS$, is a strictly proper scoring rule used to evaluate probabilistic forecasts. It measures the integrated squared difference between the predicted cumulative distribution function and the empirical cumulative distribution function of the observation. The $CRPS$ is defined as:

$$CRPS(F, y) = \int_{-\infty}^{\infty} [F(x) - \mathbf{1}(x \geq y)]^2 \, dx,$$

where F is the predicted cumulative distribution function, y is the observed value, and $\mathbf{1}(x \geq y)$ is the indicator function that equals 1 if $x \geq y$ and 0 otherwise. A smaller $CRPS$ value indicates a better probabilistic forecast. The $CRPS$ generalizes the Mean Absolute Error to probabilistic predictions and is particularly important for DGAM, which outputs a full predictive distribution rather than a single point estimate (Gneiting and Raftery, 2007).

Prediction Interval Coverage Probability at 95%

The Prediction Interval Coverage Probability at 95%, or $PICP_{95}$, measures the reliability of uncertainty estimates by calculating the proportion of observed values that fall within the 95% prediction interval. It is defined as:

$$PICP_{95} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}(L_i \leq y_i \leq U_i),$$

where L_i and U_i are the lower and upper bounds of the 95% prediction interval for observation i, y_i is the observed value, and $\mathbf{1}(\cdot)$ is the indicator function. For a well-calibrated model, $PICP_{95}$ should be close to 0.95. This metric is essential for evaluating DGAM, as it directly assesses whether the estimated variance component produces reliable prediction intervals. A $PICP_{95}$ significantly below 0.95 indicates overconfident predictions, while a value significantly above 0.95 suggests overly conservative uncertainty estimates (Kuleshov et al., 2018).

2.7 Related Researches

Gijbels and Prosdocimi (2012) address the problem of real data showing a larger or smaller variability than what is assumed in exponential family models, which are the basis of Generalized Linear Models (GLMs) and additive models, in their research. They address this issue by introducing extended generalized additive models with P-spline smooth estimation techniques for the mean and the dispersion function. Using this methodology as a foundation, the authors broaden their analysis by allowing some covariates to be modeled parametrically while keeping others nonparametric. The article's main contribution is a simulation study that looks at how well the P-spline estimation method performs in these extended models. This study includes comparisons with a standard generalized additive modeling approach and a hierarchical modeling approach, providing insights into the finite-sample behavior of the proposed technique.

Croux, Gijbels, and Prosdocimi (2012) examined how Generalized Linear Models (GLMs) are frequently used to generate parametric estimates for the mean function in their study. They introduce the idea of generalized additive models (GAMs) to increase the adaptability of the relationship between the mean function and the covariates. They discover, however, that the fixed variance structure imposed by GLMs can frequently be overly restrictive. The researchers suggest using the Extended Quasilikelihood (EQL) framework, which enables the estimation of both the mean and the dispersion (variance) as functions of the covariates, in order to overcome this limitation. However, they point out that EQL estimates are susceptible to the impact of outliers, just like other maximum likelihood techniques. Therefore, there is a need for robust estimation methods that can produce reliable estimates for both the mean and the dispersion functions.

Lou, Caruana, and Gehrke (2013) investigated common Generalized Additive Models (GAMs) that are frequently used to model the dependent variable as a sum of univariate models in their study. Although earlier studies have shown that these common GAMs can be understood, they are less accurate than more intricate models that take interactions into account. By incorporating specific terms of interacting pairs of features, the authors of this paper suggest improving standard GAMs. The resulting models, referred to

as GA2M-models (Generalized Additive Models plus Interactions), include univariate terms as well as a select few pairwise interaction terms. GA2M models can be visualized and understood by users, improving their interpretability, by limiting the model components to one- and two-dimensional components.

Harvei, Jørgensen, and Ådland (2019) studied comparing XGBoost and GAM in modeling complex relationships among multiple variables, aligning with prior research in maritime economics that highlights the limitations of linear models for vessel valuation. The study identifies vessel age at sale, timecharter rates, and fuel efficiency index as the key variables in the XGBoost model. Additionally, predicting the prices of vessels priced over 20 million poses significant challenges, potentially due to limited data, unaccounted vessel characteristics, or the influence of the financial super cycle between 2003 and 2009. Overall, the XGBoost algorithm enables accurate predictions of vessel prices based on vessel characteristics and market conditions, serving as a valuable machine learning framework for desktop valuation. Its flexibility makes it highly applicable for investors, ship owners, and other stakeholders in the maritime industry.

Chang, Tan, Lengerich, Goldenberg, and Caruana (2021) ask whether Generalized Additive Models (GAMs) are truly interpretable and trustworthy. While GAMs have grown in popularity as model class for interpretable machine learning, the existence of multiple algorithms for training GAMs presents a challenge: these algorithms can learn different, even contradictory models while achieving the same level of accuracy. As a result, determining which GAM should be trusted becomes critical. The authors extensively investigate various GAM algorithms using quantitative and qualitative analyses of real and simulated datasets. They find that GAMs with high feature sparsity, utilizing only a few variables for predictions, can miss important patterns and exhibit unfairness towards rare subpopulations. The study highlights the crucial role of inductive bias in shaping interpretable models and concludes that tree-based GAMs achieve the best balance of sparsity, fidelity, and accuracy. Thus, tree-based GAMs are deemed the most trustworthy models, offering a reliable solution for interpretable machine learning tasks.

Chang, Caruana, and Goldenberg (2022) describe Support Vector Regression (SVR) as a supervised learning technique grounded in statistical learning theory that has been

widely applied across various domains, including time series prediction, financial forecasting, and healthcare applications. Unlike traditional regression models that minimize empirical risk, SVR seeks to minimize an upper bound on the generalization error by employing a loss function that defines an ϵ-insensitive tube around the regression function, within which deviations are ignored. This formulation, together with kernel functions that map input data into high-dimensional feature spaces, enables SVR to effectively capture complex nonlinear relationships while maintaining strong generalization performance. However, SVR models are often regarded as black-box methods due to their limited interpretability, which can hinder their deployment in high-risk or decision-critical environments where transparency and explainability are essential. In contrast, Generalized Additive Models (GAMs) provide an interpretable alternative by extending generalized linear models through the replacement of the linear predictor with a sum of smooth, nonparametric functions. GAMs retain the ability to learn nonlinear transformations for individual features while combining these effects additively, thereby offering modular interpretability that allows each feature's contribution to be examined independently. Despite their long history of successful application, traditional GAMs typically rely on spline-based smoothers and backfitting algorithms, whereas modern extensions such as NODE-GAM integrate the interpretability advantages of GAMs with the flexibility and scalability of deep learning approaches.