

Lucas da Mata Guimarães

## **Titulo do Trabalho**

São Paulo - Brasil

2025

Lucas da Mata Guimarães

## **Titulo do Trabalho**

Monografia apresentada na disciplina Trabalho de Conclusão de Curso, como parte dos requisitos para obtenção do título de Bacharel em Ciência da Computação.

Centro Universitário Senac - Santo Amaro  
Bacharelado em Ciência da Computação

Orientador: Nome do Orientador

São Paulo - Brasil  
2025

*Texto da dedicatória.*

# Agradecimentos

Texto de agradecimento.

*“A vingança nunca é plena,  
mata a alma e a envenena.  
(MADRUGA, Seu, Chaves)*

# Resumo

Texto do resumo

**Palavras-chaves:** palavra-chave 1, palavra-chave 2, palavra-chave 3.

# Abstract

Abstract text in english

**Key-words:** keyword 1, keyword 2, keyword 3

# Lista de ilustrações

Figura 1 – Regressão Linear Simples . . . . .	16
Figura 2 – Regressão Linear Multipla . . . . .	18
Figura 3 – Curva Spline . . . . .	23
Figura 4 – Operação do Insertion Sort . . . . .	26



# Lista de tabelas

Tabela 1 – Matriz de Confusão . . . . .	17
Tabela 2 – Notação Assintótica . . . . .	25
Tabela 3 – Análise Insertion Sort . . . . .	26

# Lista de abreviaturas e siglas

GAM	Generalized Additive Models
GLM	Generalized Linear Model
MARS	Multivariate Adaptive Regression Spline
ML	Maximum likelihood

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>11</b>
<b>1.1</b>	<b>Contexto</b>	<b>11</b>
<b>1.2</b>	<b>Justificativa</b>	<b>11</b>
<b>1.3</b>	<b>Objetivo</b>	<b>12</b>
1.3.1	Objetivos Específicos	12
<b>2</b>	<b>REVISÃO BIBLIOGRÁFICA</b>	<b>14</b>
<b>2.1</b>	<b>Modelos Computacionais</b>	<b>14</b>
2.1.1	Modelos Lineares	14
2.1.2	Metodos de avaliação de Modelos Computacionais	17
2.1.3	Acurácia	19
2.1.4	Modelos de Distribuição de Espécies	19
2.1.4.1	GLM - Generalized Linear Model	20
2.1.4.2	GAM - Generalized Additive Model	21
2.1.4.3	MARS - Multivariate Adaptive Regression Spline	22
<b>2.2</b>	<b>Análise de Algoritmos</b>	<b>24</b>
2.2.1	Análise de Complexidade	25
2.2.2	Análise de Espaço	27
<b>2.3</b>	<b>Linguagem R</b>	<b>28</b>
2.3.1	Packages	28
<b>2.4</b>	<b>Trabalhos relacionados</b>	<b>29</b>
<b>3</b>	<b>DESENVOLVIMENTO</b>	<b>30</b>
<b>4</b>	<b>RESULTADOS</b>	<b>31</b>
<b>5</b>	<b>CONCLUSÃO</b>	<b>32</b>
<b>5.1</b>	<b>Trabalhos Futuros</b>	<b>32</b>
	<b>REFERÊNCIAS</b>	<b>33</b>
	<b>APÊNDICES</b>	<b>36</b>
	<b>APÊNDICE A – EXEMPLO DE SEÇÃO DE ANEXO</b>	<b>37</b>

# 1 Introdução

## 1.1 Contexto

O uso de modelos computacionais, na Biologia, possibilita o avanço de diferentes estudos ([COSME, 2025a](#)). Uma destas aplicações são os modelos de distribuição de espécies, que são capazes de fornecer uma visualização da situação da fauna e flora de determinada região, podendo mostrar como estas estão se comportando no decorrer do tempo ([ELITH; LEATHWICK, 2009](#)).

Entre esses modelos, os mais utilizados são o Generalized Additive Models (GAM) ([HASTIE; TIBSHIRANI, 1986](#)) e o Generalized Linear Model (GLM) ([PAUL; SAHA, 2007](#)). Esses dois modelos usam uma função para estabelecer uma relação entre a média da variável de resposta e uma função 'suavizada' das variáveis explanatórias, sendo o GLM uma extensão de modelos lineares que não forçam o dado a escalas não naturais, e o GAM uma extensão semi-parametrizada do GLM, tendo a capacidade de atuar com relações não lineares e não monótonas ([GUISAN; EDWARDS; HASTIE, 2002](#)).

Já o Multivariate Adaptive Regression Spline (MARS) combina partição recursiva e ajustes por splines, de modo a manter seus aspectos positivos, enquanto sendo menos vulnerável a suas propriedades não favoráveis. Gerando um conjunto de regras para prever valores futuros apartir de uma análise regressiva. ([FRIEDMAN, 1991](#))

Sendo as aplicações destes modelos encontradas codificadas na linguagem de programação R, que por sua vez é a linguagem de programação mais utilizada quando tratamos de ciência de dados, sendo conhecida como a linguagem mais robusta para a área de dados, tendo sido pensada para o uso em cálculos e análises estatísticas ([AWARI, 2022](#)).

Porém, estes modelos podem requisitar uma alta demanda de processamento e memória do computador hospedeiro, como citado por ([COSME, 2025a](#)), ponto este, que não é repassado nos trabalhos referentes a análise ou uso dos modelos citados. Logo, mesmo com a facilidade de se adquirir um computador, tais modelos requerem computadores de alto desempenho para serem treinados, tornando esse processo lento ou criando a necessidade de se alugar máquinas virtuais para está finalidade ([RICHTER, 2025](#)).

E quando se coloca a necessidade de se manter um controle das populações de espécies, dentro ou próximo a centros urbanos, a velocidade de preparo destes modelos se torna mais critica, já que é necessário ir desde a coleta dos dados, ao treino e validação do modelo, e análise dos resultados obtidos.

## 1.2 Justificativa

Identificar a distribuição de espécies em um dado ambiente, em um determinado intervalo de tempo, é importante para termos noção de como as espécies estão respondendo a mudanças no ambiente, no aumento ou diminuição de outra espécie.

Uma vez que essas mudanças podem ser geradas pela ação humana, na construção civil e de infraestrutura ([AMETEPEY; ANSAH, 2014](#)), conseguir estimar o impacto dessas

ações é vantajoso para a preservação de espécies.

Além disso, estas abordagens aumentam as possibilidades para integrar a infraestrutura necessária, contribuindo para a sobrevivência de espécies que estão em níveis populacionais baixos.

Modelos estatísticos, que tem a capacidade de demonstrar estes eventos, aplicam de maneiras diferentes algumas linhas de abordagem. O Generalized Additive Models (GAM), Generalized Linear Model (GLM), e o Multivariate Adaptive Regression Spline (MARS), ambos com uma abordagem de Maximum likelihood (ML), variando em sua capacidade de atuar com um determinado tipo de dado e o custo levado para seu treinamento e utilização (NORBERG et al., 2019).

Modelos que são utilizados na modelagem de distribuição de espécies necessitam de uma quantidade elevada de dados (WISZ et al., 2008), de ocorrência e ausência, sendo os dados de ausência não necessários em todos os tipos de modelos.

Nem todas as espécies são facilmente modeláveis devido à dificuldade de coleta de dados, seja pela sua raridade ou habitat (STOCKMAN; BEAMER; BOND, 2006). A colaboração de cidadãos na coleta de dados pode auxiliar na identificação de áreas prioritárias para pesquisa. Portanto, a identificação de bons modelos que trabalham com esses dados é vantajosa.

Dentro destes modelos, além da quantidade e tipo de dados necessários, precisamos levar em consideração, o custo necessário de processamento e o espaço de memória utilizado pelo mesmo, para este fim utilizamos a análise de complexidade e espaço (CORMEN et al., 2009), já que um modelo mais barato nesse quesito pode ser criado em computadores mais acessíveis (SEDGEWICK; FLAJOLET, 2013), e ser possível a construção de mais de um modelo de modo simultâneo.

Os pontos levantados anteriormente podem afetar a acurácia de um modelo, mesmo atendendo os requisitos, de pouco adianta se o mesmo nos entrega respostas que induzem ao erro. Identificar um modelo que tenham uma boa acurácia, quando trabalham somente com dados de ocorrência, assim como uma melhor avaliação computacional, se vê vantajoso para situações em que queremos criar uma análise inicial de um determinado cenário.

## 1.3 Objetivo

Este trabalho tem como objetivo avaliar e comparar a implementação encontrada nas bibliotecas mda e mgcv da linguagem R, dos modelos de distribuição de espécies, GAM, GLM e MARS, levantando o custo computacional de cada um destes apartir de uma análise de complexidade e espaço. Encontrando um modelo que melhor apresente um equilíbrio entre a acurácia e o custo computacional.

### 1.3.1 Objetivos Específicos

1. Análise de complexidade e espaço dos modelos.

- Generalized Additive Model;
- Generalized Linear Model;
- Multivariate Adaptive Regression Spline;

2. Avaliação da acurácia dos modelos com dados de ocorrência.
3. Comparação dos modelos.
4. Avaliação dos modelos com base na relação custo x acurácia.

## 2 Revisão Bibliográfica

### 2.1 Modelos Computacionais

Modelos computacionais são modelos que representam fenômenos de modo simplificado, gerando uma aproximação do evento real, tendo em vista a visualização ou entendimento de determinado fenômeno, codificados em alguma linguagem computacional para ser executado em um computador. Estes modelos podem ser criados por especialistas utilizando de equações matemáticas ou, automaticamente utilizando de técnicas de inteligência artificial. (AUGUSTO, 2025)

Ao processo de criação destes modelos, damos o nome de modelagem computacional, podendo ser aplicado em qualquer situação onde uma análise de um sistema complexo se vê necessária, sendo suas principais aplicações encontradas nas seguintes áreas, como apresentado por (COSME, 2025b):

1. **Ciência e Pesquisa:** Permite o teste de hipóteses de maneira mais rápida e eficiente.
2. **Engenharia:** Essencial para projetos de larga escala, utilizada para testar estruturas antes de começar sua construção.
3. **Medicina:** Permite a modelagem de epidemias, assim prevendo como doenças podem se espalhar em dada população, ajudando a planejar métodos de controle.

O tipo da modelagem depende do tipo de fenômeno ou problema que queremos tratar, onde os tipos principais segundo (COSME, 2025b) são:

1. **Modelagem determinística:** O comportamento do sistema é previsível, onde os mesmos parâmetros de entrada sempre produzem os mesmos resultados. Mais visto no campo da Física e Engenharia, onde os fenômenos naturais seguem um conjunto de regras bem definido.
2. **Modelagem estocástica:** Inclui elementos de incerteza e aleatoriedade, o sistema pode apresentar resultados diferentes para o mesmo conjunto de parâmetros de entrada. Comumente usada onde o acaso desempenha um papel importante, como na Biologia e Economia.
3. **Modelagem dinâmica:** Focada em sistemas que mudam ao longo do tempo, essencial em áreas como a Ecologia e Epidemiologia, onde se é preciso prever a evolução de sistemas biológicos ou a propagação de doenças.

#### 2.1.1 Modelos Lineares

Modelos lineares são modelos que preveem uma resposta linear utilizando de base a relação entre o resultado e as propriedades dadas como parâmetros. Sendo uma opção mais simples, possuem propriedades mais fáceis de serem entendidas e um tempo de

desenvolvimento mais curto quando comparado a outros tipos de modelos, como redes neurais, ou árvores de decisão, empregadas no mesmo problema. (IBM, 2025)

A linearidade destes modelos, implica que matematicamente a variação dos parâmetros independentes não possuem relações entre si, e podem ser separados em dois grupos clássicos (ADALARDO, 2020).

- **Modelos de Regressão:** Este grupo é utilizado para modelar relações entre variáveis quantitativas, que são um conjunto de valores de possível representação numérica, indicando quantidade ou magnitude. Com o intuito de estimar parâmetros, explicando relação ou para fazer previsões.
- **Modelos de Análise de Variância:** Estes modelos tem como questão principal comparar a importância de fatores sobre o comportamento da variável de resposta. Para encontrar a relação entre grupos de análise, de modo a indentificar oque gera a diferença entre os grupos estudados.

Ambas as abordagens ao modelo linear, irão gerar uma regressão linear, que é um modelo matemático que descrevem a relação entre as váriaveis dependentes e independentes usadas, tendo a possibilidade de ser representado graficamente. Podendo ser de dois tipos, simples ou múltipla.

Na regressão linear simples, queremos estimar os valor de  $a$  e  $b$  da equação da reta,  $y = a + bx$ , apartir de um conjunto de dados  $x$  e  $y$ , onde  $y$  representa a váriavel dependete e  $x$  á váriavel independente, que melhor represente a relação entre  $x$  e  $y$ . Em outras palavras, queremos estimar a inclinação da reta, esta que nos indica o efeito em  $y$  das mudanças ocorridas em  $x$  (CHEIN, 2019).

A essa reta, é dado o nome de reta de regressão linear, está que depende de cinco estatísticas básicas (CHEIN, 2019):

1. Média de  $X$ :  $\bar{X} = \frac{1}{N} \sum_{i=1}^N X_i$ ;
2. Desvio padrão de  $X$ :  $S_x = \sqrt{\frac{1}{N} \sum_{i=1}^N (X_i - \bar{X})^2}$ ;
3. Média de  $Y$ :  $\bar{Y} = \frac{1}{N} \sum_{i=1}^N Y_i$ ;
4. Desvio padrão de  $Y$ :  $S_y = \sqrt{\frac{1}{N} \sum_{i=1}^N (Y_i - \bar{Y})^2}$ ;
5. Correlação de  $X$  e  $Y$ :  $r = \frac{1}{n} \sum_{i=1}^N \frac{X_i - \bar{X}}{S_x} \cdot \frac{Y_i - \bar{Y}}{S_y}$

Com estas estatísticas podemos traçar a reta de regressão, sabendo que esta passa pelo ponto médio  $(\bar{X}, \bar{Y})$ . A inclinação da reta será dada por:

$$\beta_1 = \frac{r \cdot S_y}{S_x} \quad (2.1)$$

E o intercepto da reta de regressão, onde a reta corta um dos eixos do plano cartesiano, será dado por:

$$\beta_0 = \bar{Y} - \beta_1 \bar{X} \quad (2.2)$$



Assim resultamos na seguinte equação:

$$Y = \beta_0 + \beta_1 X \quad (2.3)$$

Onde:

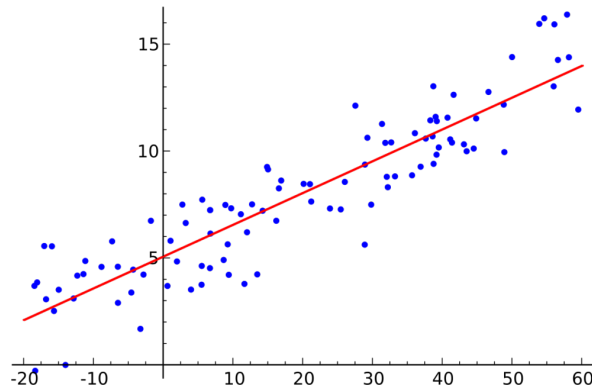
- $(X)$  é a variável independente;
- $(Y)$  é a variável dependente;
- $(\beta_0)$  é o intercepto da reta;
- $(\beta_1)$  é a inclinação da reta.

Porém, a equação 2.3 ainda não proporciona os valores de  $Y$ , mesmo possuindo os valores para  $\beta_0$  e  $\beta_1$ , visto que não é apenas a variável  $X$  que afeta os valores de  $Y$  quando tratamos de ocorrência no mundo real, assim incluímos um termo de erro  $\epsilon$ , que é o erro que se comete ao estimar os valores de  $Y$  por meio da variável  $X$  (CHEIN, 2019).

$$Y = \beta_0 + \beta_1 X + \epsilon \quad (2.4)$$

Agora com a equação 2.4, podemos criar a reta de regressão, que pode ser representada graficamente, possuindo uma estrutura semelhante ao gráfico a seguir:

Figura 1 – Regressão Linear Simples



Fonte: EBAC (2023)

A equação 2.4, pode ser escrita de forma mais geral. Visto que em nossos dados podemos trabalhar com conjuntos de valores, agrupando valores de  $Y$  distintos, para cada valor de  $X$ . Por exemplo, dados que representam a qualidade de vida nos estados brasileiros com o número de postos de saúde. Assim a equação 2.4 ficaria (CHEIN, 2019):

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i \quad (2.5)$$

Onde:

- $(i)$  representa um agrupamento de dados, um estado brasileiro seguindo o exemplo dado acima;
- $(Y_i)$  é a variável dependente;
- $(X_i)$  é a variável independente, representaria o número de postos de saúde em dado estado.
- $(\beta_0)$  é o intercepto;
- $(\beta_1)$  é a inclinação, e o efeito médio de  $X_i$  sobre  $Y_i$ ;
- $(\epsilon)$  é o erro médio ao se estimar  $Y_i$  por meio de  $X_i$ ;

Já quando tratamos da regressão linear múltipla, é levado em conta que outros fatores podem afetar a variável de resposta, este que também podem ser correlacionados com a variável independente. A formula para este modelo de regressão, pode ser representa da seguinte forma, onde temos  $k$  variáveis, explicativas (CHEIN, 2019):

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_k X_k + \epsilon \quad (2.6)$$

Onde:

- $(\beta_0)$  é o intercepto;
- $(\beta_1, \dots, \beta_k)$  são "inclinações", mesmo que na prática não sejam inclinações da função
- $(\epsilon)$  é o termo de erro.

Aqui temos  $\beta_1$  até  $\beta_k$  como coeficientes parciais da regressão (CHEIN, 2019). Neste caso, a visualização por meio de um gráfico, fica comprometida, visto que temos um número  $k$  de  $X$ , para ilustrar, usemos uma situação onde temos dois  $X$ , aqui podemos representar os valores por um gráfico de três dimensões.

Quando temos mais de dois valores de  $X$  a representação gráfica fica confusa para o entendimento humano, mas ainda podemos tratar o problema como uma reta.

### 2.1.2 Metodos de avaliação de Modelos Computacionais

Antes de apontarmos as metricas, precisamos explicar os tipos de classificação que um modelo pode chegar, categorizando a sua previsão como correta ou não, estes tipos são representados pela matriz de confusão (DEVELOPERS, 2024):

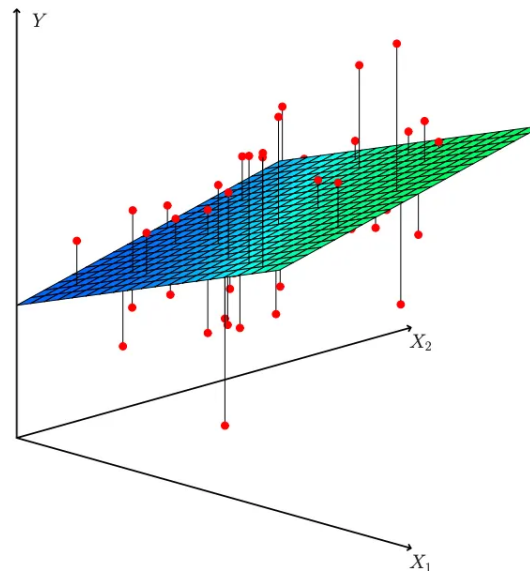
Tabela 1 – Matriz de Confusão

	Verdadeiro Positivo	Verdadeiro Negativo
Positivo Previsto	Positivo Verdadeiro	Falsos Positivos
Negativo Previsto	Falsos Negativos	Negativos Verdadeiros

Fonte: (FILHO, 2023)

Onde:

Figura 2 – Regressão Linear Múltipla



Fonte: [Novak \(2022\)](#)

- **Positivos Verdadeiros:** São as classificações positivas corretas;
- **Falsos Positivos:** São positivos que foram erroneamente classificados como negativos;
- **Falsos Negativos:** São negativos que foram erroneamente classificados como positivos;
- **Negativos Verdadeiros:** São as classificações negativas corretas;

Existem várias métricas para se avaliar a qualidade de um modelo computacional, sendo os métodos mais comuns a acurácia, precisão, recall, F1-Score e área da curva ROC (AUC-ROC) ([DUARTE, 2024](#)) onde:

- **Acurácia:** Uma métrica simples e amplamente utilizada que mede a proporção de previsões corretas feitas pelo modelo;
- **Precisão:** Uma métrica que mede a proporção de previsões positivas corretas feitas pelo modelo, muito utilizada quando um falso positivo agrega um custo muito alto;
- **Recall:** Uma métrica que mede a proporção de exemplos positivos que foram corretamente identificados pelo modelo;
- **F1-Score:** É uma média harmônica entre a Precisão e Recall, fornece um equilíbrio entre essas duas métricas, utilizado quando se quer levar em consideração tanto os falsos positivos quanto os falsos negativos;
- **AUC-ROC:** Avalia o desempenho de modelos de classificação binária em diferentes limites de decisão, onde quanto maior o valor melhor o modelo separa essas classes.

Como cada métrica possui um campo de atuação, para o problema abordado neste trabalho, as relevantes são a acurácia e precisão, onde a precisão apresenta uma métrica de avaliação onde se aceita o erro de um falso negativo, pois o erro de um falso positivo é algo prejudicial a previsão, isso fazendo um local sem ocorrência da espécie analisada pelo modelo poderia ser indicado como um lugar de ocorrência, torna a métrica por acurácia mais confiável neste caso.

### 2.1.3 Acurácia

A acurácia é um modo de avaliar a performance de um modelo, assim identificando se seus resultados podem ser considerados válidos ou não. Para chegarmos na acurácia utilizamos da seguinte fórmula:

$$A = \frac{PC}{TP} \quad (2.7)$$

Onde:

- (A) é a Acurácia
- (PC) é o total de Previsões Corretas, encontrada pela soma, (*PositivosVerdadeiros* + *NegativosVerdadeiros*);
- (TP) é o Total de Previsões, encontrada pela soma, (*PC* + *FalsosVerdadeiros* + *FalsosNegativos*).

Como a acurácia incorpora por completo a matriz de confusão 1, em um conjunto de dados equilibrado, com uma quantidade de exemplos semelhante para as duas classes, ela pode ser usada como uma medida grosseira da qualidade de um modelo (DEVELOPERS, 2024).

Onde temos mais exemplos de uma classe do que de outras, é importante considerar outras métricas de avaliação, já que esses modelos são considerados desbalanceados (FILHO, 2023).

### 2.1.4 Modelos de Distribuição de Espécies

Definimos um Modelos de Distribuição de Espécies, SDM (Spice Distribution Model), como um modelo que relaciona dados de distribuição de espécies, com informações sobre as características ambientais e/ou espaciais de certas localidades, podendo ser usados para entender e/ou prever a distribuição de uma espécie em uma dada localidade (ELITH; LEATHWICK, 2009).

SDMs contemporâneos combinam conceitos de ecologia e história natural com os avanços mais recentes em estatísticas e tecnologia da informação, as raízes destes modelos são encontradas nos estudos mais antigos que descrevem padrões biológicos em termos de relações com geografia e/ou gradientes ambientais, e estudos que indicam a resposta individual de espécies para seus ambientes, proveem um forte argumento conceitual para se modelar espécies de modo individual (ELITH; LEATHWICK, 2009).

Segundo (OLIVER, 2024) as principais fontes de informações para a criação destes modelos são:

- Dado de ocorrência: Geralmente coordenadas de latitude e longitude onde a espécie foi observada, conhecida como dado de **ocorrência**, alguns modelos fazem uso de dados de **ausência**, que são coordenadas geograficas onde se sabe que a espécie não ocorre;
- Dado ambiental: São a descrição do ambiente, podendo conter medições de temperatura e precipitação, como também, a ocorrência e ausência de outras espécies, como predadores, competidores ou fontes de alimento.

Dentro dos SDMs, temos varios frameworks de modelagem, sendo os mais utilizados o Generalized Linear Model (GLM), Generalized Additive Model (GAM) e Multivariate Adaptive Regression Spline (MARS), que são encontradas nos softwares mais amigaveis ao usuário e bem documentados (NORBERG et al., 2019), assim como podem ser encontrados em bibliotecas de linguagens de programação, como R nas bibliotecas mda, onde encontramos o GLM e GAM (HASTIE et al., 2024) e mgcv onde encontramos o MARS(WOOD, 2025).

#### 2.1.4.1 GLM - Generalized Linear Model

Generalized Linear Models agrupão uma grande quantidade de modelos discretos e continuos, sendo particularmente util para se trabalhar com dados discretos, sendo uma extensão de General Linear Models, apresentada por Nelder e Wedderburn (1972), que consiste na incerção da família exponencial de distribuições de erro junto com a distribuição normal (PAUL; SAHA, 2007).

Ao contrário dos modelos lineares classicos, assim como o General Linear Model, que propoem uma distribuição Gaussiana (normal) e uma função de ligação dos valores de  $X$  e  $Y$ , os GLMs permitem que a função de distribuição seja alguma da família de distribuições exponenciais, Gaussiana, Poisson ou Binomial, e a função de ligação pode ser qualquer função monotônica diferenciável, como a logaritmica (GUISAN; EDWARDS; HASTIE, 2002).

No GLM as variáveis de predição  $X_j$ , onde  $j = 1, \dots, p$  com  $p$  sendo a quantidade de  $X$ s, são combinadas para se formar um preditor linear, LP, que é relacionado ao valor esperado  $\mu = E(Y)$  da variável de resposta  $Y$  através de uma função de junção  $g()$  (GUISAN; EDWARDS; HASTIE, 2002), assim podemos chegar a seguinte formula:

$$g(E(Y)) = LP = \alpha + X^T \beta \quad (2.8)$$

Onde:

- $\alpha$ : é uma constante, chamada de intercepto;
- $X$ : é um vetor de  $p$  preditores,  $(X_1, \dots, X_p)$ ;
- $\beta$ : é um vetor de  $p$  coeficientes de regressão, uma para cada preditor,  $(\beta_1, \dots, \beta_p)$ .

Assim escrevemos o modelo para variáveis  $X$  e  $Y$  genericas, os termos correspondes para uma dada observação  $i$  da amostra é (GUISAN; EDWARDS; HASTIE, 2002):

$$g(\mu_i) = \alpha + \beta_1 X_{i1} + \beta_2 X_{i2} + \dots + \beta_p X_{ip} \quad (2.9)$$

Aqui a variancia de  $Y$  depende de  $\mu = E(Y)$  através da função de variancia  $V(\mu)$ , dado  $Var(Y) = \phi V(\mu)$ , onde  $\phi$  (phi) é o parametro de dispersão, quando se espera um  $\phi$  maior que o valor antecipado dado a distribuição escolhida, o parametro de escala pode ser estimado utilizando de quasi-likelihood ([GUISAN; EDWARDS; HASTIE, 2002](#)).

Por sua vez, quasi-likelihood é um metodo de se generalizar a abordagem pr likelihood-based para GLMs, permitindo se estimar so valores de  $\beta$  de tal forma que não se é necessário se especificar uma distribuição para a saída. Normalmente utilizado como um mecanismo para lidar com dados muitos dispersos, ou, quando não se deseja fazer afirmações solidas de distribuição sobre as variaveis de saída ([SPICKER, 2025](#)).

#### 2.1.4.2 GAM - Generalized Additive Model

Gerados a partir do GLM, este modelo possui uma automatização para se identificar os termos de polinomio apropriado e as transformações dos preditores que melhoram a qualidade do modelo linear. Logo podemos dizer que os GAMs estão aninhados dentro do GLMs que por sua vez estão aninhados em modelos lineares, LM ([GUISAN; EDWARDS; HASTIE, 2002](#)):

$$LM \subset GLM \subset GAM \quad (2.10)$$

GAMs são parametrizados como os GLMs, porém alguns preditores podem ser modelados de modo não parametrizado em adição a termos lineares e polinomias para outros preditores. Um passo crucial para se aplicar GAMs é selecionar o nível apropriado de "suavização" para os preditores.

Nestes se substitui as função de predição linear  $\eta = \sum_1^p \beta_j X_j$  pela função de predição aditiva  $\eta = \sum_1^p s_j(X_j)$ , onde  $s_j(X_j)$  é uma função de suavização do valor  $X_j$ . Onde a forma assumida para a estimativa pelo modelo de regressão linear, possui a seguinte característica ([HASTIE; TIBSHIRANI, 1986](#)):

$$E(Y|X_1, X_2, \dots, X_p) = \beta_0 + \beta_1 X_1 + \dots + \beta_p X_p. \quad (2.11)$$

Enquanto no GAM, se generaliza o modelo de Regressão linear, tornando a equação 2.11:

$$E(Y|X_1, X_2, \dots, X_p) = s_0 + \sum_{j=1}^p s_j(X_j) \quad (2.12)$$

Como exemplo, tomemos o caso de um preditor simples, neste, nosso modelo seria:

$$E(Y|X) = s(X) \quad (2.13)$$

Para estimarmos  $s(x)$  a partir de nossos dados, podemos usar um estimativa razoável de  $E(Y|X = x)$ , sendo uma dessas classes de estimativas as estimativas médias locais,  $\hat{s}(x_i) = Ave_{j \in N_i}(Y_j)$ , onde  $Ave$  representa uma operação de média como a média e  $N_i$  é a vizinhança de  $x_i$ , os valores  $x$  que estão proximos a  $x_i$ , em associação com a vizinhança, temos o tamanho  $w$  da janela, isto é a proporção de pontos contidos em cada janela ([HASTIE; TIBSHIRANI, 1986](#)).

Se assumirmos  $x$  sendo um valor inteiro, que  $wn$  é ímpar, então a span  $w$  vizinhança mais próxima simétrica conterá  $wn$  pontos, o  $i$ -ésimo ponto mais  $\frac{(wn-1)}{2}$  pontos em cada lado do  $i$ -ésimo ponto. Assumindo que os dados estão ordenados de forma crescente em  $x$ , uma definição formal seria (HASTIE; TIBSHIRANI, 1986):

$$N_i = \max(i - \frac{wn-1}{2}, 1), \dots, i-1, i, i+1, \dots, \min(i + \frac{wn-1}{2}, n) \quad (2.14)$$

A vizinhança fica truncada próxima aos pontos finais se os pontos  $\frac{wn-1}{2}$  não estão disponíveis, o span de  $w$  controla a suavidade dos resultados estimados, e geralmente é escolhido com base nos dados a serem usados (HASTIE; TIBSHIRANI, 1986). Se  $\hat{Ave}$  representar a média aritmética, então  $\hat{s}(\cdot)$  é a running lines smoother, e sua definição para estimar valores de  $x$  é:

$$\hat{s}(x_i) = \hat{\beta}_{0i} + \hat{\beta}_{1i}x_i \quad (2.15)$$

Onde  $\hat{\beta}_{0i}$  e  $\hat{\beta}_{1i}$  são as estimativas por least square para os pontos de dados de  $N_i$ :

$$\begin{aligned} \hat{\beta}_{1i} &= \frac{\sum_{j \in N_i} (x_j - \bar{x}_i)y_j}{\sum_{j \in N_i} (x_j - \bar{x}_i)^2}, \\ \hat{\beta}_{0i} &= \bar{y}_i - \hat{\beta}_{1i}\bar{x}_i, \\ \bar{x}_i &= \frac{1}{n} \sum_{j \in N_i} x_j, \\ \bar{y}_i &= \frac{1}{n} \sum_{j \in N_i} y_j \end{aligned} \quad (2.16)$$

Outros métodos de estimar  $E(Y|X)$  poderiam ser usados, acarretando na mudança do custo computacional do modelo, podendo trabalhar tão bem quanto ou melhor do que o running lines smoother (HASTIE; TIBSHIRANI, 1986).

#### 2.1.4.3 MARS - Multivariate Adaptive Regression Spline

O foco na modelagem por regressão, é estimar uma função  $\hat{f}(x_1, \dots, x_n)$ , que melhor se assemelhe a função  $f(x_1, \dots, x_n)$ , que descreve a relação entre as propriedades de um dado fenômeno, e o seu resultado real. (FRIEDMAN, 1991).

$$y = f(x_1, \dots, x_n) + \epsilon \quad (2.17)$$

A função de  $n$ -dimensões  $f$  captura a relação de predição de  $y$  em  $x_1, \dots, x_n$ , onde o alvo da análise regressiva é usar os dados para construir a função  $\hat{f}(x_1, \dots, x_n)$  que serve como uma aproximação razoável para  $f(x_1, \dots, x_n)$  sobre um domínio  $D$  de interesse. Onde MARS prove uma abordagem natural para a modelagem de variáveis categóricas, variáveis aninhadas (variável que contém outra variável) e valores faltantes (FRIEDMAN; ROOSEN, 1995).

O procedimento MARS, é baseado em uma generalização de métodos de spline para ajuste de funções. Consideramos o caso de apenas um preditor,  $x(n = 1)$ , para se estimar

a função  $f(x)$ , uma função splien de aproximação  $\hat{f}_q(x)$  é obtida por dividir a abrangencia de  $x$  valore em  $k + 1$  regiões separadas por  $k$  pontos (FRIEDMAN; ROOSEN, 1995).

$$\hat{f}_q(x) = \sum_{k=0}^{k+q} a_k B_k^{(q)}(x) \quad (2.18)$$

Onde  $B_k^{(q)}(x)$  é um conjunto de funções base que englobam o espaço de dota a função spline de ordem  $q$  e  $a_k$  é o valor do coeficiente de espanção sendo sem limitadores (FRIEDMAN; ROOSEN, 1995).

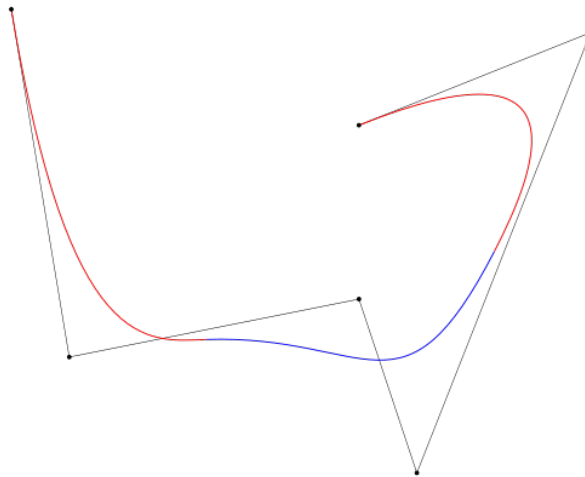
A base mais popular é a 'B-spline' possuindo a superioridade em numeros de propriedades usada em conjunto com o least-squares fitting. Sendo a 'B-spline' definida por,  $K + 2$  locais de pontos adjacentes, onde o função limitadora tem o maior suporte mas é definida cada por um por um único local de pontos. Para a seleção destas funções base, que gera um grupo de resultados não válidos sera removida, se tivermos um modelo aditivo, onde

$$\hat{f}(x) = \sum_{j=1}^n f_j(x_j) \quad (2.19)$$

foce considerado adequado, qualquer função que envolvesse mais de uma váriavel se tornaria ilegítima para inclusão no modelo (FRIEDMAN; ROOSEN, 1995).

Em outras palavras, podemos definir um spline, como a aproximação de uma curva de formato complexo utilizando do menor número possível de retas, sendo essa quantidade de retas o prametro  $q$ , como pode ser visto na figura a seguir:

Figura 3 – Curva Spline



Fonte: [Wikipedia \(2025\)](#)

O algoritmo MARS usa de uma estrategia de paços forward/backward para produzir o seu conjunto de funções base. A parte de forward é um processo recursivo, a cada interação simultaneamente controí um lista expansiva de funções base a serem consideradas e então decide quais considerar naquele paço, se repetindo até uma quantidade relativamente grande de funções bases for selecionada. Um conjunto final de funções base de tamanho



apropriado é então selecionado por meio de um procedimento de seleção de subconjunto de variáveis passo a passo regressivo, usando as funções base produzidas pelo algoritmo progressivo como 'variáveis' candidatas (FRIEDMAN; ROOSEN, 1995).

O paço de forward começa com uma unica função base no modelo:

$$B_0(x) = 1 \quad (2.20)$$

Após a Méximo iteração teremos  $2M + 1$  funções:

$$\{B_m(\mathbf{x})\}_0^{2M} \quad (2.21)$$

No modelo, cada iteração ( $M + 1$ ) adiciona duas novas funções bases:

$$\begin{aligned} B_{2M+1}(x) &= B_{l(M+1)}(x) \left\{+(x_{v(M+1)} - t_{M+1})\right\}_+^q \\ B_{2M+2}(x) &= B_{l(M+1)}(x) \left\{-(x_{v(M+1)} - t_{M+1})\right\}_+^q \end{aligned} \quad (2.22)$$

Aqui  $B_{l(M+1)}(x)$  é uma das funções base  $2M + 1$  já selecionada,  $0 \leq l(M + 1) \leq 2M$ ,  $v(M + 1)$  é uma variavel preditora, não representada em  $B_{l(M_1)}(x)$ , e  $t_{M+1}$  é um ponto nestá váriavel. Os três parametros  $l(M + 1)$ ,  $v(M + 1)$  e  $t_{M+1}$  que definem as duas novas funções base, são escolhidos para serem os que melhor melhoram o enciaxe do "novo" modelo com os dados.

Para pequenas amostras o algoritmo MARS ira tentar produzir modelos envolvem interações de baixa ordem, e com grandes amostras ira favoravelmente entrar em interações de alta ordem para os pociveis candidatos.

## 2.2 Análise de Algoritmos

A análise de algoritmos é o processo de se identificar uma formula matemática que melhor represente o custo de utilização de uma dado algoritmo, podendo ser o tempo que o algoritmo leva para terminar com uma quantidade  $n$  de dados, ou de espaço, quanto da memoria do computador o algoritmo ira usar durante seu processo.

Neste processo identificamos a qual família de problemas esse algoritmo pertence, que corresponde ao seu custo de computação (CORMEN et al., 2009) e assim podemos categoriza-lo com base na notação assintótica.

A formula matemática identificada como a representação do custo computacional é "arredondada" para uma das familias apresentadas acima, reduzindo a formula a sua caracteristica mais presente, visto que aqui assumimos que  $n$  seja um valor muito grande. Por exemplos uma função que tenha a forma de  $n^2 + n + c$ , onde  $c$  é uma constante, pode ser reduzida a  $n^2$ , já que está parte tera maior peso durante a computação, a caracterizando como  $O(n^2)$ , onde  $O$  é a notação "O grande", que representa a complexidade do algoritmo (CORMEN et al., 2009).

Com está análise, encontramos um algoritmo que melhor se encaixe em determinado problema, antes de se desenvolver o mesmo em uma linguagem de programação especifica,

Tabela 2 – Notação Assintótica

Complexidade	Nome	Eficiente
$O(1)$	Constante	Sim
$O(\log n)$	Logartimica	Sim
$O(n)$	Linear	Sim
$O(n \log n)$	"Linearitimica"	Sim
$O(n^2)$	Quadrática	Sim
$O(n^3)$	Cúbica	Sim
$O(2^n)$	Exponencial	Não
$O(n!)$	Fatorial	Não

Fonte: (BIG..., 2025)

ou de utilizarmos algoritmos já implementados de forma cega, podendo perder em tempo e em consumo desnecessário de memória.

### 2.2.1 Análise de Complexidade

Na análise de complexidade, estimamos o tempo de execução de um algoritmo dado uma entrada de tamanho  $n$ , analisando seus comandos, como exemplo podemos utilizar o algoritmo de ordenação insertion sort.

O insertion sort é um algoritmo eficiente para ordena uma sequência pequena de números, funciona de modo semelhante a como muitas pessoas organizam cartas de um baralho na mão. Começamos com uma mão vazia, e a cada momento se pega uma carta da mesa e se insere na mão em sua posição correta, verificando da direita para a esquerda (CORMEN et al., 2009).

Recebendo um vetor  $A[1..n]$  contendo a sequência de tamanho  $n$  que será ordenada, o algoritmo representado pelo pseudo código 2.1 ordena a sequência encontrada no vetor  $A$ , tendo no máximo um valor da sequência armazenado fora do vetor a cada dado momento, ao final, o vetor  $A$ , irá conter a sequência ordenada (CORMEN et al., 2009).

Listing 2.1 – Insertion-Sort

```

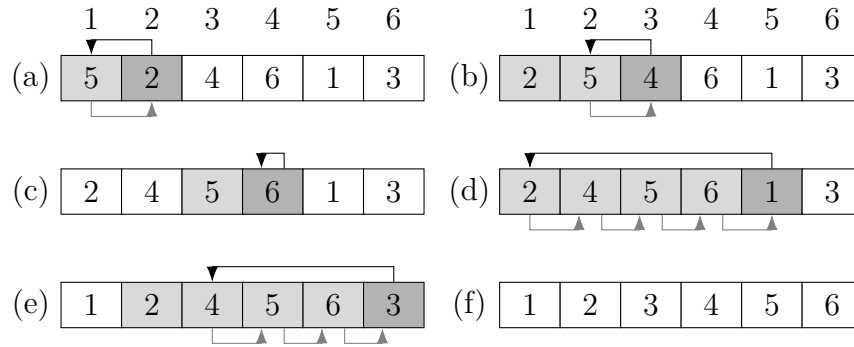
1  Insertion-Sort(A)
2  for j = 2 to n
3      key = A[j]
4      i = j - 1
5      while i > 0 and A[i] > key
6          A[i + 1] = A[i]
7          i = i - 1
8      A[i + 1] = key

```

Fonte: Cormen et al. (2009)

Para visualizar a sequência de passos que o insertion sort executa para a ordenação de dado vetor, tomemos o vetor  $A = [5, 2, 4, 6, 1, 3]$ , a sequência pode ser vista na imagem 4.

Figura 4 – Operação do Insertion Sort



Fonte: Cormen et al. (2009)

A figura 4, mostra que o algoritmo 2.1 funciona para se ordenar um vetor, onde, no início de cada interação do loop for, controlado pelo valor de  $j$ , o subvetor de elementos  $A[1..j-1]$  consiste de uma sequência ordenada, e os valores restantes  $A[j+1..n]$  são os elementos ainda não verificados e  $j$  é o elemento que está sendo verificado (CORMEN et al., 2009).

Agora com o algoritmo 2.1, conseguimos analisar o custo computacional do insertion sort, mas primeiro precisamos definir, dois conceitos, "tempo de execução" e "tamanho da entrada", que segundo (CORMEN et al., 2009) é:

- Tamanho da entrada: Varia dependendo do problema a ser abordado, porém é comumente o número de itens na entrada, por exemplo, o tamanho do vetor  $A$ .
- Tempo de execução: É o número de operações ou 'paços' executados, aqui desconsideramos o hardware, e assumimos que mesmo cada paço leva um tempo diferente de outros, o tempo da  $i$  paço é sempre uma constante,  $c_i$ .

Em nosso algoritmo 2.1, para cada  $j = 2, 3, \dots, n$ , dizemos que  $t_j$  representa o número de vezes que o loop **while** da linha 5 foi executado, para cada valor de  $j$ . Sempre que temos um loop for ou while o teste é executado uma vez a mais do que o corpo do loop.

Assim conseguimos chegar a seguinte análise do algoritmo 2.1:

Tabela 3 – Análise Insertion Sort

Linha	Custo	Veze
2	$c_2$	$n$
3	$c_3$	$n - 1$
4	$c_4$	$n - 1$
5	$c_5$	$\sum_{j=2}^n t_j$
6	$c_6$	$\sum_{j=2}^n (t_j - 1)$
7	$c_7$	$\sum_{j=2}^n (t_j - 1)$
8	$c_8$	$n - 1$

Fonte: Cormen et al. (2009)

O tempo de execução total do algoritmo será a soma dos tempos de cada linha, onde uma linha que leve  $c_i$  paços para executar e execute  $n$  vezes ira contribuir com  $c_i n$  para o tempo total de execução. Para encontrar o tempo de execução do insertion- sort 2.1, em uma entrada de tamanho  $n$ , somamos o produto do Custo e Vezes da tabela 3:

$$T(n) = c_2 n + c_3(n-1) + c_4(n-1) + c_5 \sum_{j=2}^n t_j + c_6 \sum_{j=2}^n (t_j - 1) + c_7 \sum_{j=2}^n (t_j - 1) + c_8(n-1) \quad (2.23)$$

O tempo de execução de um algoritmo vária dependendo do dado de entrada, onde podemos cair nó melhor ou pior caso, durante a análise de complexidade, se leva em consideração apenas o pior caso (CORMEN et al., 2009), onde, no nosso caso de exemplo, o algoritmo será executado por completo percorrendo cada elemento do vetor, que ocorre quando o vetor se econtra em ordem decrescente.

O pior caso no algoritmo 2.1, pode ser representado pela seguinte equação:

$$\begin{aligned} T(n) &= c_2 n + c_3(n-1) + c_4(n-1) + c_5\left(\frac{n(n+1)}{2} - 1\right) + \\ &\quad c_6\left(\frac{n(n-1)}{2}\right) + c_7\left(\frac{n(n+1)}{2}\right) + c_8(n-1) \\ &= \left(\frac{c_5}{2} + \frac{c_6}{2} + \frac{c_7}{2}\right)n^2 + (c_2 + c_3 + c_4 + \frac{c_5}{2} - \\ &\quad \frac{c_6}{2} - \frac{c_7}{2} + c_8)n - (c_3 + c_4 + c_5 + c_8) \end{aligned} \quad (2.24)$$

Podemos expressar a equação 2.24, como  $an^2 + bn - c$ , para constantes  $a, b$  e  $c$ , que dependem do custo de  $c_i$ , mas é a taxa de crescimento que realmente nos interessa, logo consideramos apenas o maior termo da equação, isto é  $an^2$ , já que os outros termos são insignificantes para valores muitos grandes de  $n$ . Com isto, ficamos com o fator de  $n^2$  para o crescimento, portanto o pior caso de tempo de execução como  $\theta(n^2)$ .

Agora para encontrarmos a qual classe apresentada na tabela 2 o algoritmo 2.1 pertence, convertemos da notação  $\theta$  (theta) para a  $O$  (O-grande), esta conversão é simples já que o grupo de problemas abordados pela notação  $\theta$  engloba a notação  $O$ , isto é  $\theta(g(n)) \subseteq O(g(n))$  onde  $g(n)$  é a função de crescimento ( $n^2$ ), portanto, o algoritmo 2.1 é  $O(n^2)$  logo pertence ao grupo de probelmas Quadráticos (CORMEN et al., 2009).

### 2.2.2 Análise de Espaço

Na análise de espaço, levamos em conta as váriaveis que o algoritmo utiliza e/ou cria, como por exemplo, estruturas auxiliares como vetores ou matrizes, conseguimos representar o consumo esperado de memória atravez de uma formula matemática (SEGEWICK; FLAJOLET, 2013), assim como ná análise anterior, como exemplo tomemos o algoritmo insertion sort.

No algoritmo de insertion sort, temos as seguintes váriaveis:

- *vetor A* a sequencia de  $n$  inteiros;

- *key* uma variável que armazena um valor presente no vetor *A*;
- *i* um valor inteiro que representa uma posição na sequência;
- *j* um valor inteiro que representa uma posição na sequência;

Assim, o algoritmo não cria nenhuma variável a mais durante sua execução, acarretando em um custo de memória constante durante a sua execução. Sendo o tamanho de 3 inteiros mais o tamanho de um inteiro multiplicado pelo tamanho  $n$  da sequência de inteiros.

Logo o consumo de memória pode ser descrito por uma função linear,  $f(n) = n + 3$ , onde  $n$  é o tamanho da sequência de entrada, portanto o crescimento do algoritmo 2.1 é de tipo linear.

## 2.3 Linguagem R

R foi desenvolvido e pensado como uma linguagem e ambiente para computação estatística e gráficos, possui a capacidade de trabalhar com vários processos estatísticos, como modelagem linear e não-linear, testes clássicos de estatística, séries temporais, entre outros, e ser altamente extensível ([THE... , 2025](#)).

Devido ao seu desenvolvimento focado em aplicações de computação estatística, ser semelhante e compatível com a linguagem S já existente, e capaz de atuar com códigos feitos nas linguagens C, C++ e Fortran ([THE... , 2025](#)), tornou o R uma linguagem popular dentro da área de computação estatística ([AWARI, 2022](#)).

O ambiente do R, pode ser facilmente incrementado, se utilizando de bibliotecas, nomeadas como packages, por padrão, temos oito packages que são encontrados junto da distribuição comum do R, outros packages podem ser encontrados em sites especializados na distribuição dos mesmos ([THE... , 2025](#)).

Além de ser de fácil incrementação, o ambiente R, possui um interface de desenvolvimento gratuita, o R Studio, onde a utilização da linguagem fica concentrada em uma única aplicação, facilitando a visualização dos dados, gráficos, alteração e manutenção do código e packages ([RSTUDIO... , 2025](#)).

### 2.3.1 Packages

Packages, também conhecidos como bibliotecas ou pacotes, dentro do cenário de computação, se refere a agrupamentos de códigos desenvolvidos por terceiros ou por si mesmo, para encapsular alguma atividade ou processo, que pode ser utilizado em mais de um projeto.

Como exemplo, podemos utilizar os packages, *mda* e *mgcv*, presentes neste trabalho, o package *mda* engloba funções para se aplicar um grupo de modelagem de dados, sendo estes o Mixture and Flexible Discriminant Analysis, Multivariate Adaptive Regression Splines (MARS) e Vector-response Smoothing Splines ([HASTIE et al., 2024](#)), já o *mgcv*, engloba Generalized Additive Models e algumas de suas variações, Generalized Cross Validation e similares ([WOOD, 2025](#)).

Estes packages permitem ao usuário, apenas chamar a função que representa o processo a ser executado, não precisando se preocupar com os pormenores da execução em si, apenas passar os parametros necessários de modo correto.

## 2.4 Trabalhos relacionados

### 3 Desenvolvimento

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed sollicitudin tempor sapien in maximus. Quisque in vulputate dui, ac vestibulum sem. Suspendisse urna velit, dapibus nec egestas a, rhoncus vitae neque. Mauris quis efficitur augue. Aliquam quis tellus eget orci aliquet aliquam. Sed luctus, quam vitae elementum malesuada, quam lacus imperdiet urna, sed ullamcorper libero magna non elit. Cras laoreet arcu a augue volutpat, suscipit pretium tellus tempus. Sed eros tortor, imperdiet eu neque id, interdum egestas tortor.

## 4 Resultados

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed sollicitudin tempor sapien in maximus. Quisque in vulputate dui, ac vestibulum sem. Suspendisse urna velit, dapibus nec egestas a, rhoncus vitae neque. Mauris quis efficitur augue. Aliquam quis tellus eget orci aliquet aliquam. Sed luctus, quam vitae elementum malesuada, quam lacus imperdiet urna, sed ullamcorper libero magna non elit. Cras laoreet arcu a augue volutpat, suscipit pretium tellus tempus. Sed eros tortor, imperdiet eu neque id, interdum egestas tortor.



## 5 Conclusão

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed sollicitudin tempor sapien in maximus. Quisque in vulputate dui, ac vestibulum sem. Suspendisse urna velit, dapibus nec egestas a, rhoncus vitae neque. Mauris quis efficitur augue. Aliquam quis tellus eget orci aliquet aliquam. Sed luctus, quam vitae elementum malesuada, quam lacus imperdiet urna, sed ullamcorper libero magna non elit. Cras laoreet arcu a augue volutpat, suscipit pretium tellus tempus. Sed eros tortor, imperdiet eu neque id, interdum egestas tortor.

### 5.1 Trabalhos Futuros

- Trabalho Futuro 1
- Trabalho Futuro 2
- Trabalho Futuro 3

# Referências

- ADALARDO. 7. *Modelos Lineares*. 2020. Acesso em: 26 de Abril de 2025. Disponível em: <[http://ecor.ib.usp.br/doku.php?id=03\\_apostila:06-modelos#:~:text=SÃo%20chamados%20modelos%20lineares%20aqueles,dos%20demais%20parÃmetros%20do%20modelo.](http://ecor.ib.usp.br/doku.php?id=03_apostila:06-modelos#:~:text=SÃo%20chamados%20modelos%20lineares%20aqueles,dos%20demais%20parÃmetros%20do%20modelo.)> Citado na página 15.
- AMETEPEY, S. O.; ANSAH, S. K. Impacts of construction activities on the environment : the case of ghana. *Journal of Construction Project Management and Innovation*, v. 4, n. sup-1, p. 934–948, 2014. Disponível em: <<https://journals.co.za/doi/abs/10.10520/EJC162729>>. Citado na página 11.
- AUGUSTO, D. A. *Entenda o que são modelos computacionais e como o SISS-Geo os utiliza*. 2025. Acesso em: 25 de Abril de 2025. Disponível em: <<https://www.biodiversidade.ciss.fiocruz.br/entenda-o-que-sao-modelos-computacionais-e-como-o-siss-geo-os-utiliza>>. Citado na página 14.
- AWARI. *Conheça as principais linguagens de programação para Ciência de Dados*. 2022. Acesso em: 25 de Abril de 2025. Disponível em: <<https://awari.com.br/linguagens-de-programacao-para-ciencia-de-dados/>>. Citado 2 vezes nas páginas 11 e 28.
- BIG O Notation. 2025. Acesso em: 13 de Maio de 2025. Disponível em: <<https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.ml-science.com%2Fbig-o-notation&psig=AOvVaw0et-yABeIQN9psuih6aZk&ust=1747766347288000&source=images&cd=vfe&opi=89978449&ved=0CAMQjB1qFwoTCLDKmJeXsI0DFQAAAAAdAAAAABAE>>. Citado na página 25.
- CHEIN, F. *Introdução aos Modelos de Regressão Linear*. Enap, 2019. ISBN 9788525601155. Disponível em: <[https://repositorio.enap.gov.br/bitstream/1/4788/1/Livro\\_RegressÃo%20Linear.pdf](https://repositorio.enap.gov.br/bitstream/1/4788/1/Livro_RegressÃo%20Linear.pdf)>. Citado 3 vezes nas páginas 15, 16 e 17.
- CORMEN, T. et al. *Introduction to Algorithms, third edition*. MIT Press, 2009. (Computer science). ISBN 9780262033848. Disponível em: <<https://books.google.com.br/books?id=i-bUBQAAQBAJ>>. Citado 5 vezes nas páginas 12, 24, 25, 26 e 27.
- COSME, A. L. *Modelagem computacional: o que é, qual sua aplicação*. 2025. Acesso em: 17 de Abril de 2025. Disponível em: <<https://123ecos.com.br/docs/modelagem-computacional/>>. Citado na página 11.
- COSME, A. L. *Modelagem computacional: o que é, qual sua aplicação*. 2025. Acesso em: 25 de Abril de 2025. Disponível em: <<https://123ecos.com.br/docs/modelagem-computacional/#:~:text=Os%20principais%20tipos%20sÃo%20a,em%20diferentes%20Ãreas%20do%20conhecimento.>> Citado na página 14.
- DEVELOPERS, G. for. *Classificação: precisão, recall, precisão e métricas relacionadas*. 2024. Acesso em: 3 de Maio de 2025. Disponível em: <<https://developers.google.com/machine-learning/crash-course/classification/accuracy-precision-recall?hl=pt-br>>. Citado 2 vezes nas páginas 17 e 19.

- DUARTE, R. *Métricas de Avaliação em Modelos de Classificação em Machine Learning*. 2024. Acesso em: 17 de Maio de 2025. Disponível em: <<https://sigmoidal.ai/metricas-de-avaliacao-em-modelos-de-classificacao-em-machine-learning/>>. Citado na página 18.
- EBAC. *Regressão Linear: teoria e exemplos*. 2023. Acesso em: 27/04/2025. Disponível em: <<https://ebaonline.com.br/blog/regressao-linear-seo>>. Citado na página 16.
- ELITH, J.; LEATHWICK, J. R. Species distribution models: Ecological explanation and prediction across space and time. *Annual Review of Ecology, Evolution, and Systematics*, Annual Reviews, v. 40, n. Volume 40, 2009, p. 677–697, 2009. ISSN 1545-2069. Disponível em: <<https://www.annualreviews.org/content/journals/10.1146/annurev.ecolsys.110308.120159>>. Citado 2 vezes nas páginas 11 e 19.
- FILHO, M. *O Que É Acurácia Em Machine Learning?* 2023. Acesso em: 3 de Maio de 2025. Disponível em: <<https://mariofilho.com/o-que-e-acuracia-em-machine-learning/>>. Citado 2 vezes nas páginas 17 e 19.
- FRIEDMAN, J. H. Multivariate Adaptive Regression Splines. *The Annals of Statistics*, Institute of Mathematical Statistics, v. 19, n. 1, p. 1 – 67, 1991. Disponível em: <<https://doi.org/10.1214/aos/1176347963>>. Citado 2 vezes nas páginas 11 e 22.
- FRIEDMAN, J. H.; ROOSEN, C. B. An introduction to multivariate adaptive regression splines. *Statistical Methods in Medical Research*, v. 4, n. 3, p. 197–217, 1995. PMID: 8548103. Disponível em: <<https://doi.org/10.1177/096228029500400303>>. Citado 3 vezes nas páginas 22, 23 e 24.
- GUISAN, A.; EDWARDS, T. C.; HASTIE, T. Generalized linear and generalized additive models in studies of species distributions: setting the scene. *Ecological Modelling*, v. 157, n. 2, p. 89–100, 2002. ISSN 0304-3800. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0304380002002041>>. Citado 3 vezes nas páginas 11, 20 e 21.
- HASTIE, T.; TIBSHIRANI, R. Generalized Additive Models. *Statistical Science*, Institute of Mathematical Statistics, v. 1, n. 3, p. 297 – 310, 1986. Disponível em: <<https://doi.org/10.1214/ss/1177013604>>. Citado 3 vezes nas páginas 11, 21 e 22.
- HASTIE, T. et al. *Package 'mda'*. 0.5-5. ed. [S.l.], 2024. <https://CRAN.R-project.org/package=mda>. Disponível em: <<https://cran.r-project.org/web/packages/mda/mda.pdf>>. Citado 2 vezes nas páginas 20 e 28.
- IBM. *Modelos lineares*. 2025. Acesso em: 26 de Abril de 2025. Disponível em: <<https://www.ibm.com/docs/pt-br/spss-modeler/18.5.0?topic=node-linear-models>>. Citado na página 15.
- NORBERG, A. et al. A comprehensive evaluation of predictive performance of 33 species distribution models at species and community levels. *Ecological Monographs*, v. 89, n. 3, p. e01370, 2019. Disponível em: <<https://esajournals.onlinelibrary.wiley.com/doi/abs/10.1002/ecm.1370>>. Citado 2 vezes nas páginas 12 e 20.
- NOVAK, H. *Modelos de Regressão Linear e como aplicamos a técnica na Loft*. 2022. Acesso em: 27/04/2025. Disponível em: <<https://medium.com/loftbr/regressão-linear-65fc8caeb729>>. Citado na página 18.

- OLIVER, J. 2024. Acesso em: 4 de Maio de 2025. Disponível em: <[https://jcoliver.github.io/learn-r/011-species-distribution-models.html#:~:text=\(%20predicts%20-,Components%20of%20the%20model,these%20data%20\(see%20below\).>](https://jcoliver.github.io/learn-r/011-species-distribution-models.html#:~:text=(%20predicts%20-,Components%20of%20the%20model,these%20data%20(see%20below).>)> Citado na página 19.
- PAUL, S.; SAHA, K. K. The generalized linear model and extensions: a review and some biological and environmental applications. *Environmetrics*, v. 18, n. 4, p. 421–443, 2007. Disponível em: <<https://onlinelibrary.wiley.com/doi/abs/10.1002/env.849>>. Citado 2 vezes nas páginas 11 e 20.
- RICHTER, F. *Amazon and Microsoft Stay Ahead in Global Cloud Market*. 2025. Acesso em: 19 de Abril de 2025. Disponível em: <<https://www.statista.com/chart/18819/worldwide-market-share-of-leading-cloud-infrastructure-service-providers/>>. Citado na página 11.
- RSTUDIO Desktop. 2025. Acesso em: 10 de Maio de 2025. Disponível em: <<https://posit.co/download/rstudio-desktop/>>. Citado na página 28.
- SEDGEWICK, R.; FLAJOLET, P. *An Introduction to the Analysis of Algorithms*. Pearson Education, 2013. ISBN 9780133373486. Disponível em: <<https://books.google.com.br/books?id=P3tCB8Q7mA8C>>. Citado 2 vezes nas páginas 12 e 27.
- SPICKER, D. *Quasi-Likelihood Theory in Full*. 2025. Acesso em: 23 de Maio de 2025. Disponível em: <[https://dylanspicker.com/courses/STAT437/course\\_index.html](https://dylanspicker.com/courses/STAT437/course_index.html)>. Citado na página 21.
- STOCKMAN, A. K.; BEAMER, D. A.; BOND, J. E. An evaluation of a garp model as an approach to predicting the spatial distribution of non-vagile invertebrate species. *Diversity and Distributions*, v. 12, n. 1, p. 81–89, 2006. Disponível em: <<https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1366-9516.2006.00225.x>>. Citado na página 12.
- THE R Project for Statistical Computing. 2025. Acesso em: 10 de Maio de 2025. Disponível em: <<https://www.r-project.org>>. Citado na página 28.
- WIKIPEDIA. *B-Spline*. 2025. Acesso em: 13 de Maio de 2025. Disponível em: <<https://en.wikipedia.org/wiki/B-spline>>. Citado na página 23.
- WISZ, M. S. et al. Effects of sample size on the performance of species distribution models. *Diversity and Distributions*, v. 14, n. 5, p. 763–773, 2008. Disponível em: <<https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1472-4642.2008.00482.x>>. Citado na página 12.
- WOOD, S. *Package 'mgcv'*. 1.9-3. ed. [S.l.], 2025. <https://CRAN.R-project.org/package=mgcv>. Disponível em: <<https://cran.r-project.org/web/packages/mgcv/mgcv.pdf>>. Citado 2 vezes nas páginas 20 e 28.

## Apêndices

# APÊNDICE A – Exemplo de seção de anexo

1 EXEMPLO DE CODIGO A SER ADICIONADO