Regressão Linear e Logística

github.com/kvpergentino/data-science-explained/

A jornada do aprendizado de máquina pode ser compreendida como uma progressão em níveis de complexidade e abstração. Em sua base, encontram-se **algoritmos estatísticos,** como a regressão, que fornecem os fundamentos para modelar relações e realizar previsões a partir de dados. A partir dessa etapa, surgem arquiteturas mais sofisticadas, como as **redes neurais**, que, quando organizadas em múltiplas camadas, originam o campo do **deep learning**, capaz de identificar padrões complexos em grandes volumes de dados. O deep learning, por sua vez, viabiliza o desenvolvimento da **inteligência artificial generativa**, voltada não apenas à análise ou previsão, mas também à criação de novos conteúdos. Um dos resultados mais avançados dessa evolução são os **modelos de linguagem de grande escala (LLMs)**, aplicações de deep learning no contexto da IA generativa, que demonstram capacidade sem precedentes de compreender e produzir linguagem humana.

Nesse contexto, a **regressão** ocupa um papel central por ser uma das técnicas mais fundamentais do **aprendizado supervisionado**. Seu objetivo principal é **modelar a relação entre uma variável alvo contínua** (y – aquilo que se deseja prever) **e uma ou mais variáveis explicativas** (x – também chamadas de features ou preditores). Ao estabelecer essas relações matemáticas e prever valores com base em padrões observados, a regressão introduz conceitos essenciais como **treinamento de modelos, ajuste de parâmetros e avaliação de desempenho**. Essa base conceitual não apenas facilita a compreensão de métodos mais avançados, mas também oferece instrumentos práticos para enfrentar problemas reais de previsão e apoiar a tomada de decisão.

Para ilustrar, considere um conjunto de dados sobre emissões de CO_2 de diferentes modelos de automóveis. Este dataset contém características como o tamanho do motor, número de cilindros e consumo de combustível. Com esses dados, é possível treinar um modelo de regressão para aprender os padrões que relacionam as características do veículo às emissões de CO_2. A partir desse modelo, torna-se viável prever a emissão de um novo carro, com base em suas especificações.

O processo pode ser resumido em três etapas:

1. Utilizamos dados históricos (características e emissões de CO_2 de carros já conhecidos).
2. Treinamos um modelo de regressão para que ele "aprenda" o padrão e a relação entre essas variáveis.
3. Usamos o modelo treinado para estimar o valor da emissão de CO_2 para um carro novo ou hipotético.

Existem diversos algoritmos de regressão, e a escolha do mais adequado depende da natureza dos dados e da relação entre as variáveis.

1. Visão Geral: Regressão Simples e Múltipla

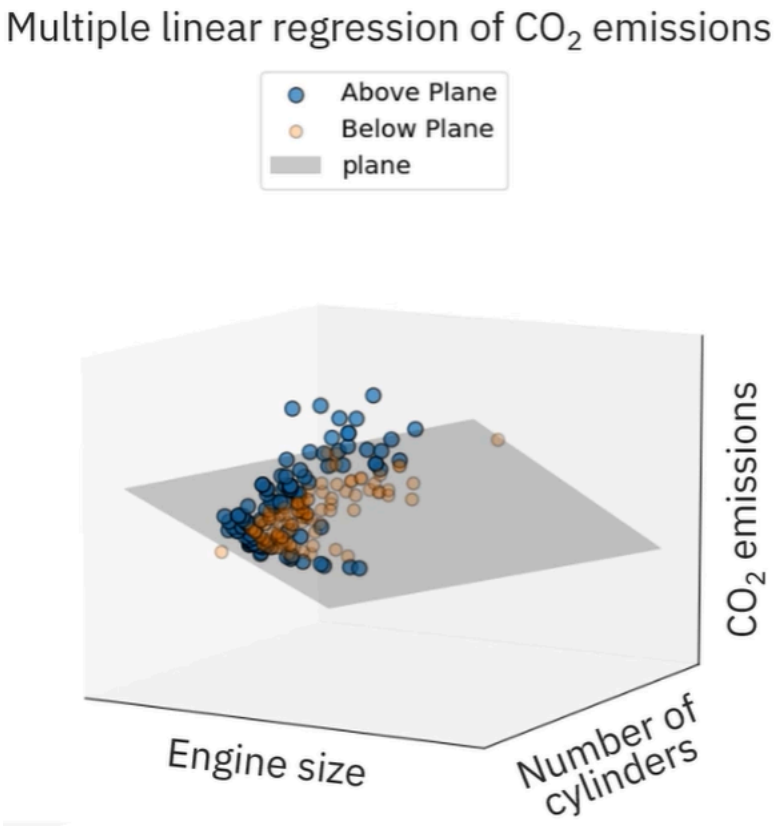
Os modelos de regressão podem ser classificados com base no número de variáveis explicativas que utilizam.

Na **regressão simples**, utilizamos uma **única variável independente** para estimar o valor de uma variável dependente (o alvo). A relação entre essas duas variáveis pode ser:

- **Linear:** Assume-se que a relação entre as variáveis pode ser representada por uma linha reta. Por exemplo, prever a emissão de CO_2 usando *apenas* o tamanho do motor.
- **Não Linear:** A relação é modelada por uma curva, sendo mais adequada para padrões complexos que não seguem uma tendência linear.

Quando utilizamos **duas ou mais variáveis independentes** para prever a variável alvo, o processo é chamado de **regressão múltipla**. Assim como na regressão simples, a relação modelada também pode ser linear ou não linear.

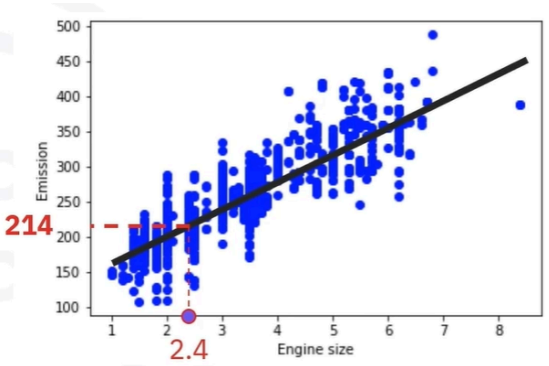
Por exemplo, um modelo de regressão múltipla poderia prever a emissão de CO_2 utilizando o tamanho do motor, o número de cilindros e o consumo de combustível simultaneamente.



2. Regressão Linear Simples

Vamos voltar ao nosso exemplo de emissões de CO_2. Se plotarmos os dados em um gráfico de dispersão (*scatter plot*), com o tamanho do motor no eixo X e as emissões de CO_2 no eixo Y, poderemos observar uma correlação: à medida que o tamanho do motor aumenta, as emissões de CO_2 também tendem a aumentar.

	ENGINESIZE	CYLINDERS	FUELCONSUMPTION_COMB	CO2EMISSIONS
0	2.0	4	8.5	196
1	2.4	4	9.6	221
2	1.5	4	5.9	136
3	3.5	6	11.1	255
4	3.5	6	10.6	244
5	3.5	6	10.0	230
6	3.5	6	10.1	232
7	3.7	6	11.1	255
8	3.7	6	11.6	267
9	2.4	4	9.2	?

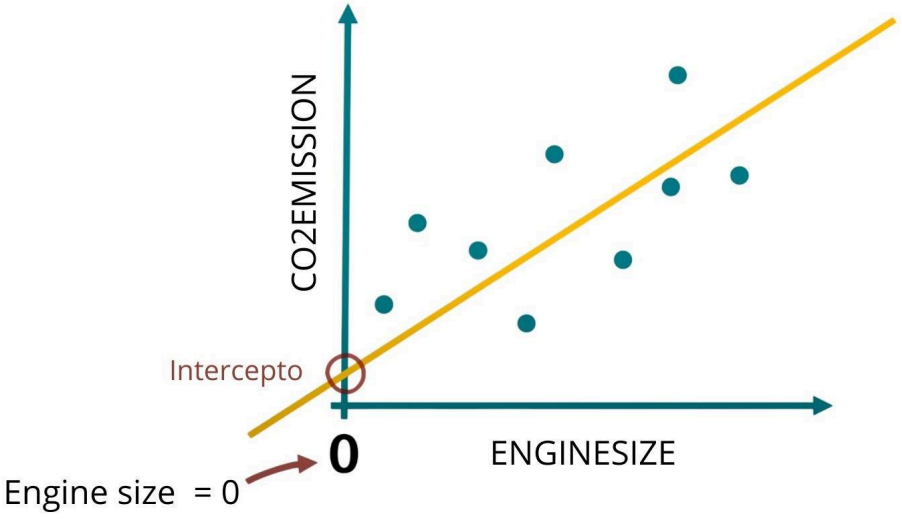


O objetivo da regressão linear simples é encontrar a **linha de melhor ajuste** (*best-fit line*) que passa por esses pontos de dados. Essa linha representa o modelo matemático que descreve a relação.

A equação que define esta linha é:

$$\hat{y} = \theta_0 + \theta_1 x_1$$

- $\hat{y}$ (lê-se "y chapéu") é o **valor previsto** para a variável alvo y (emissões de CO_2).
- x_1 é a **variável independente** (tamanho do motor) que usamos para fazer a previsão.
- θ_0 (theta zero) é o **intercepto** (ou *bias*), o valor de $\hat{y}$ quando x_1 é zero. É a linha de base, como dizer “mesmo que o carro não tivesse motor, ainda haveria um valor inicial de emissão previsto”.



- θ_1 (theta um) é a **inclinação** da linha, representando o quanto $\hat{y}$ varia para cada unidade de aumento em x_1 (taxa de mudança). Se θ_1 = 40, significa que para cada aumento de 1 no tamanho do motor, o modelo prevê que as emissões de CO_2 aumentem em 40 unidades.

Os valores de θ_0 e θ_1 são chamados de **coeficientes** do modelo, e o algoritmo de regressão os calcula para encontrar a linha que melhor se ajusta aos dados.

As representações das variáveis podem mudar a depender do contexto em que a equação é abordada, podendo tomar aparências diferentes, mas mantendo a estrutura base que se dá por:

$$Variável\ Alvo\ =\ Intercepto\ ou\ Bias\ +\ Inclinação\ ou\ Peso\ \cdot\ Variável\ independente$$

ou

$$Variável\ Alvo\ =\ Inclinação\ ou\ Peso\ \cdot\ Variável\ independente\ +\ Intercepto\ ou\ Bias$$

Medindo o Erro do Modelo

Nenhum modelo é perfeito. A diferença entre o valor real (y) e o valor previsto pelo modelo ($\hat{y}$) para um determinado ponto é chamada de **erro residual** ou **resíduos**.

Erro Residual = Valor Real - Valor Previsto

Em um modelo de regressão, o **erro residual** é a distância vertical entre o valor real de um ponto de dado (y_i) e o valor previsto pela linha de regressão ($\hat{y}_i$).

A regressão linear encontra a linha de melhor ajuste ao minimizar a soma dos quadrados de todos os erros residuais. Essa abordagem é conhecida como o método dos **Mínimos Quadrados Ordinários (OLS)**. A performance do modelo está diretamente ligada a esse princípio, sendo medida pelo **Erro Quadrático Médio (MSE)**, que transforma a soma total minimizada pelo OLS em uma média.

Calculando os Coeficientes do Modelo

A solução OLS, uma contribuição matemática de Gauss e Legendre, nos fornece fórmulas exatas para determinar os coeficientes da linha, θ_0 (intercepto) e θ_1 (coeficiente angular).

Para usá-las, precisamos calcular as médias das nossas variáveis: $\bar{x}$ (média dos valores da variável independente) e $\bar{y}$ (média dos valores da variável dependente).

Cálculo do Coeficiente Angular (θ_1): O coeficiente θ_1 é calculado primeiro, pois o valor de θ_0 depende dele. De forma técnica, a fórmula calcula o coeficiente θ_1 usando a covariância e a variância de todo o conjunto de dados (as variáveis X e Y). O valor de θ_1 que encontramos se torna a inclinação da nossa linha de regressão, nos dizendo o quanto o valor previsto (y) muda para cada unidade de mudança no valor de entrada (x).

$$\theta_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

- **Numerador:** $\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$

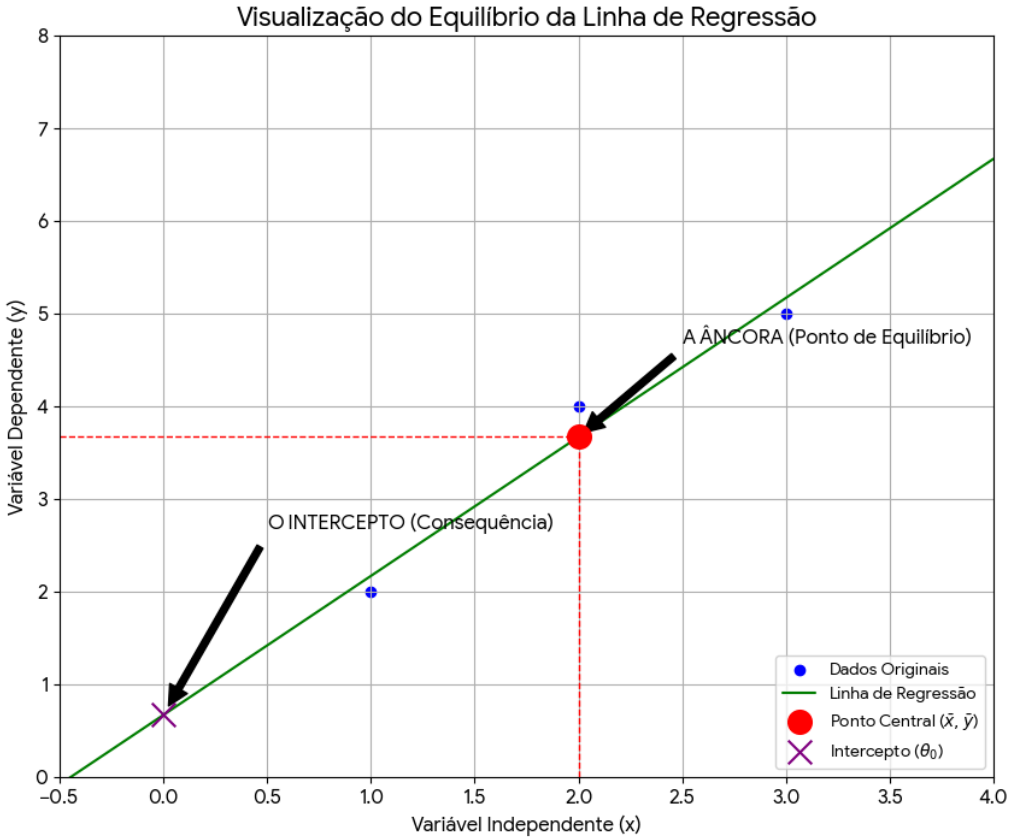
  Esta parte representa a soma dos produtos dos desvios de cada ponto de dados em relação às suas médias. Se *x* e *y* tendem a aumentar juntos, ambos os termos $(x_i - \bar{x})$ e $(y_i - \bar{y})$ serão positivos, resultando em um θ_1 positivo. Se um tende a aumentar enquanto o outro diminui, o resultado será um θ_1 negativo.

- **Denominador:** $\sum_{i=1}^{n}(x_i - \bar{x})^2$

  Esta é a soma dos quadrados dos desvios de *x* em relação à sua média. Mede a dispersão total ou a variabilidade na variável independente *x*.

Cálculo do Intercepto (θ_0): Uma vez que o coeficiente angular (θ_1) tenha sido calculado, a determinação do intercepto (θ_0) é um passo direto, baseado em uma propriedade fundamental da regressão por Mínimos Quadrados Ordinários (OLS): a linha de melhor ajuste é matematicamente garantida a passar pelo **"centro de massa"** dos dados.

Este centro de massa nada mais é do que o ponto formado pelas médias de suas variáveis: $(\bar{x}, \bar{y})$. Pense nele como o ponto de equilíbrio de toda a sua nuvem de dados. Para que a linha de regressão minimize a distância total de todos os pontos, ela precisa, obrigatoriamente, passar por este centro.



Sabendo dessa regra, podemos usar a própria equação da regressão para encontrar a fórmula de θ0.

1. O princípio nos diz que o ponto ($\bar{x}$, $\bar{y}$) deve satisfazer a equação da linha. Isso significa que, se inserirmos o valor de $\bar{x}$ na equação, o valor previsto para y deve ser exatamente $\bar{y}$.
2. Aplicando essa lógica, substituímos as médias na equação da linha, o que nos dá:

$$\bar{y} = \theta_0 + \theta_1 \bar{x}$$

3. Agora, com uma simples manipulação algébrica, podemos isolar θ_0 para sua fórmula de cálculo:

$$\theta_0 = \bar{y} - \theta_1 \bar{x}$$

Esta é a fórmula exata para encontrar o intercepto que, junto com θ_1, define a única linha que minimiza o erro quadrático.

Onde:

- θ_0: O intercepto da linha de regressão, o valor que desejamos encontrar.
- $\bar{y}$: A média de todos os valores da variável dependente (y).
- $\bar{x}$: A média de todos os valores da variável independente (x).
- θ_1: O coeficiente angular, que **já deve ter sido calculado** na etapa anterior.

O intercepto θ_0 tem duas interpretações essenciais:

1. **A Definição Matemática:** Ele representa o valor previsto da variável dependente (y) quando a variável independente (x) é igual a zero. Graficamente, é o ponto onde a linha de regressão cruza o eixo vertical (eixo Y).
2. **A Função Estrutural no Modelo:** Mais importante, esta fórmula garante que a linha de regressão esteja perfeitamente **"ancorada"** no centro de equilíbrio dos dados ($\bar{x}$, $\bar{y}$). É essa ancoragem que posiciona a linha de forma a balancear as distâncias dos pontos acima e abaixo dela, uma condição indispensável para cumprir o objetivo do OLS de minimizar a soma dos erros quadrados.

Vantagens e Desvantagens da Regressão Linear Simples

Vantagens

- **Interpretabilidade:** Os coeficientes do modelo possuem uma interpretação direta, quantificando de forma clara o impacto da variável independente sobre a variável dependente.
- **Eficiência Computacional:** Apresenta baixo custo computacional, permitindo um processo de treinamento rápido mesmo em conjuntos de dados volumosos.
- **Base Teórica:** Serve como fundamento conceitual para uma gama de algoritmos mais complexos, como Regressão Logística, Regressão Múltipla e modelos com regularização.
- **Baixo Risco de *Overfitting*:** Devido à sua baixa complexidade, o modelo tende a capturar a tendência geral dos dados, reduzindo a probabilidade de se ajustar a ruídos aleatórios do conjunto de treinamento.

Desvantagens

- **Pressuposto de Linearidade:** O modelo é inerentemente incapaz de capturar relações não lineares entre as variáveis, o que pode resultar em um alto viés e performance inadequada.

- **Sensibilidade a *Outliers*:** Valores discrepantes (outliers) podem influenciar desproporcionalmente o ajuste da linha de regressão, uma vez que o método de Mínimos Quadrados Ordinários minimiza a soma dos erros quadráticos.
- **Dependência de Pressupostos Estatísticos:** A validade das inferências do modelo (testes de hipótese, intervalos de confiança) depende de pressupostos sobre os resíduos (e.g., normalidade, homocedasticidade), que frequentemente não são atendidos em dados reais.
- **Incerteza na Extrapolação:** A realização de previsões para valores fora do intervalo de dados observados é inerentemente arriscada, pois a validade da relação linear não é garantida fora desse escopo.

---

3. Regressão Linear Múltipla

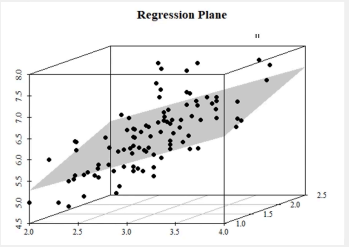
A Regressão Linear Múltipla constitui uma generalização do modelo de Regressão Linear Simples. Sua principal distinção reside na utilização de **duas ou mais variáveis independentes (preditoras)** para modelar o comportamento de uma única variável dependente (resposta), resultando em um modelo com maior poder preditivo.

- **Regressão Linear Simples:** Modela a relação entre *y* e um único preditor *x*.
- **Regressão Linear Múltipla:** Modela a relação entre *y* e um conjunto de preditores $x_1, x_2, x_3, ..., x_n$

A representação geométrica do modelo evolui com a dimensionalidade. Enquanto a regressão simples define uma **reta** em um espaço 2D, um modelo com duas variáveis explicativas define um **plano** em um espaço 3D. Com mais de duas variáveis, o modelo é representado por um **hiperplano** em um espaço multidimensional.

One feature: $y = \theta_0 + \theta_1 \boldsymbol{x}_1$ defines a line

Two features: $\hat{y} = \theta_0 + \theta_1 \boldsymbol{x}_1 + \theta_2 \boldsymbol{x}_2$ defines a plane



N features: $y = \theta_0 + \theta_1 \boldsymbol{x}_1 + \theta_2 \boldsymbol{x}_2 + \ ... + \theta_N \boldsymbol{x}_N$ an N−dimensional hyperplane

O modelo é expresso como uma função linear como na regressão simples, que combina as variáveis preditoras (x_n) para estimar a variável dependente, sendo $n >= 2$.

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + ... + \theta_n x_n$$

Estimação dos Parâmetros do Modelo

O treinamento do modelo consiste na estimação dos coeficientes (θi) que melhor se ajustam aos dados. Conforme abordado no estudo da regressão simples, o critério para este ajuste é a minimização do **Erro Quadrático Médio (MSE)**. Os dois métodos mais comuns para encontrar os coeficientes são:

- **Mínimos Quadrados Ordinários (OLS):** Utiliza operações de álgebra linear para calcular os valores ótimos dos coeficientes de forma direta e analítica.
- **Otimização (ex: Gradiente Descendente):** Uma abordagem iterativa que ajusta gradualmente os coeficientes para minimizar o erro. Este método é particularmente útil para conjuntos de dados muito grandes.

Desafios Técnicos e Mitigações

A implementação eficaz de um modelo de regressão múltipla requer atenção a certas armadilhas:

- **Sobreajuste (Overfitting):** Ocorre ao adicionar variáveis em excesso, fazendo com que o modelo "decore" os dados de treinamento e perca sua capacidade de generalização para dados novos. A mitigação envolve técnicas de **seleção de variáveis** e **regularização**.
- **Multicolinearidade:** Refere-se à alta correlação entre duas ou mais variáveis independentes (ex: área de um imóvel em m² e em pés²). A colinearidade torna os coeficientes instáveis e de difícil interpretação. A solução comum é a remoção de variáveis redundantes.

Tratamento de Variáveis Categóricas

Modelos de regressão exigem entradas numéricas. Variáveis como "tipo de carro" (manual/automático) precisam ser convertidas em formato numérico (ex: 0 e 1) através de técnicas de codificação para serem incluídas no modelo.

Aplicações da Análise de Regressão

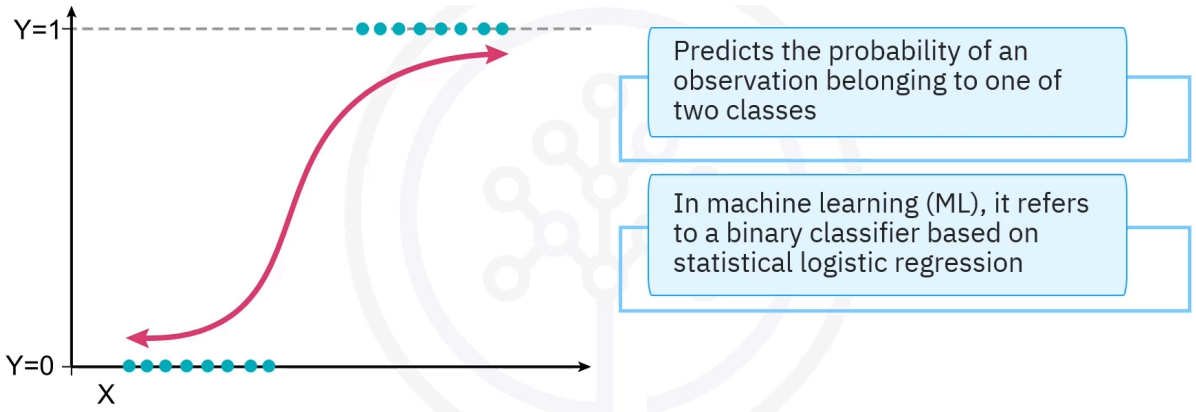
Essencialmente, as técnicas de regressão são utilizadas para estimar ou prever um valor contínuo. Suas aplicações são vastas e permeiam diversas indústrias:

- **Negócios e Finanças:** Previsão de vendas, estimativa do valor do ciclo de vida de um cliente (CLV), previsão de preços de ações e avaliação de risco de crédito.
- **Setor Imobiliário:** Estimativa do preço de imóveis com base em características como área, número de quartos e localização.
- **Saúde e Ciências Ambientais:** Previsão da disseminação de doenças infecciosas, estimativa da probabilidade de um paciente desenvolver certas condições, previsão de volumes de chuva ou determinação da severidade de incêndios florestais.
- **Engenharia:** Otimização de processos e previsão de manutenção preditiva para equipamentos industriais.

4. Regressão Logística

A **Regressão Logística** é uma técnica de modelagem estatística e um algoritmo fundamental de aprendizado de máquina supervisionado. Seu objetivo principal é prever a **probabilidade** de uma observação pertencer a uma de duas classes possíveis (um resultado binário), como `sim/não`, `verdadeiro/falso` ou `1/0`.

No contexto de machine learning, embora o nome "regressão" possa sugerir a previsão de valores contínuos, a Regressão Logística é, na prática, um dos algoritmos mais utilizados para problemas de **classificação binária**.



A transição de um preditor de probabilidade para um classificador é direta: estabelece-se um limiar de probabilidade (conhecido como **fronteira de decisão**). Se a probabilidade prevista para uma observação for maior que esse limiar, ela é atribuída a uma classe (ex: classe 1); caso contrário, é atribuída à outra classe (ex: classe 0).

Cenários de Aplicação

A Regressão Logística é uma escolha adequada em cenários específicos:

1. **Quando a Variável Alvo é Binária:** A aplicação mais direta do modelo é em problemas onde o resultado que se deseja prever possui apenas duas categorias.
2. **Quando a Saída Probabilística é Necessária:** Além de classificar, o modelo informa a probabilidade de uma ocorrência. Isso é valioso em áreas como análise de risco de crédito (probabilidade de inadimplência) ou medicina (probabilidade de um paciente ter uma doença).
3. **Quando a Interpretabilidade é Importante:** O modelo permite entender o impacto de cada variável independente (feature) na previsão. Os coeficientes (pesos) associados a cada feature indicam a direção e a força da sua influência no resultado.
4. **Quando os Dados são Linearmente Separáveis:** Em sua forma mais básica, o algoritmo funciona melhor quando uma linha, plano ou hiperplano pode separar razoavelmente bem as duas classes no espaço de features. A fronteira de decisão criada pelo modelo é linear.

Exemplos práticos de aplicação incluem:

- **Saúde:** Prever a chance de um paciente desenvolver diabetes com base em seu peso, altura, pressão arterial e resultados de exames de sangue.
- **Negócios:** Estimar a probabilidade de um cliente cancelar uma assinatura (churn) com base em seu histórico de uso e dados demográficos.
- **Finanças:** Determinar a probabilidade de um proponente de empréstimo se tornar inadimplente.
- **Engenharia:** Calcular a probabilidade de falha de um componente ou sistema.

A Limitação do Modelo Linear para Classificação

Para entender por que a Regressão Logística é necessária, é útil primeiro entender por que a Regressão Linear é inadequada para tarefas de classificação binária.

Consideremos um exemplo: prever o *churn* (cancelamento de serviço) de clientes com base em sua idade.

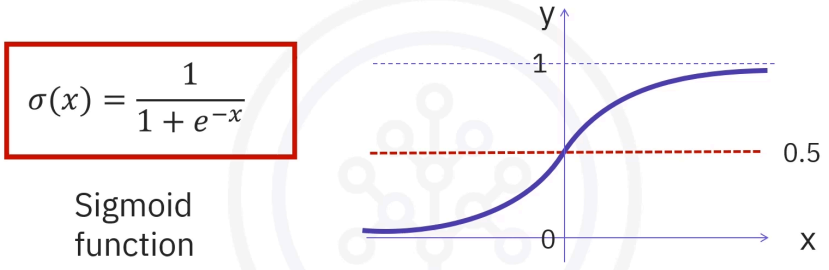
- **Variável Independente (x):** Idade.
- **Variável Alvo (y):** Churn (1 para "sim", 0 para "não").

Se aplicarmos um modelo de Regressão Linear, $\hat{y} = \theta_0 + \theta_1 x_1$, ele tentará ajustar uma reta aos dados. Isso gera dois problemas fundamentais:

1. **Saída Não-Limitada:** A reta de regressão produz valores que podem ser menores que 0 ou maiores que 1. Isso não faz sentido conceitual, pois o que buscamos é uma probabilidade, que deve estar estritamente contida no intervalo [0,1].
2. **Falta de Sensibilidade Probabilística:** Uma abordagem seria usar um limiar (ex: se $\hat{y} > 0.5$, classificar como 1). No entanto, essa "função degrau" é abrupta e não captura nuances. Por exemplo, ela trataria um cliente com previsão de 0.6 e outro com 0.99 da mesma forma (ambos classe 1), embora a certeza da previsão seja muito diferente. O objetivo é ter uma curva suave que represente a transição de probabilidade.

A Função Sigmoide (Logit)

Para resolver as limitações da Regressão Linear, a Regressão Logística emprega uma transformação matemática em sua saída. O componente central desta transformação é a **Função Sigmoide** (também conhecida como função logística ou *logit*). A imagem a seguir detalha a fórmula e o comportamento gráfico desta função:



Conforme ilustrado, a função sigmoide possui propriedades matemáticas ideais para a modelagem de probabilidades:

1. **Fórmula e Transformação:** A equação recebe a saída do modelo linear como sua entrada. Para o modelo logístico, é convencional denotar essa entrada como **z** (onde $z = \theta_0 + \theta_1 x_1 \ldots$). A imagem, por sua vez, exibe a forma matemática genérica da função, utilizando ***x*** como a variável de entrada. A fórmula é:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

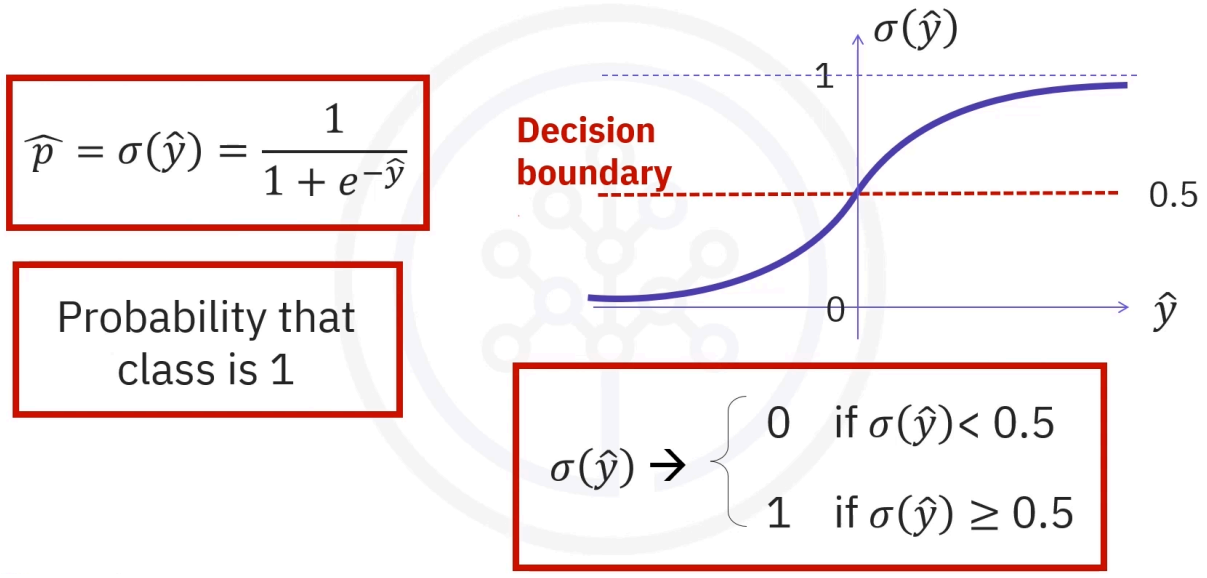
2. **Saída Limitada:** O gráfico demonstra visualmente a característica mais importante da função: independentemente do valor de entrada (seja x ou z), a saída σ(z) é sempre "esmagada" para um valor contido estritamente **entre 0 e 1**.

3. **Comportamento da Curva:** A curva em formato de "S" mostra que para valores de entrada muito positivos, a função se aproxima de 1, e para valores muito negativos, aproxima-se de 0. O ponto central mostra que uma entrada igual a 0 resulta em uma saída de exatamente 0.5.

Esta capacidade de converter qualquer valor real em uma saída que pode ser interpretada como uma probabilidade é o que torna a função sigmoide a peça-chave da Regressão Logística.

Da Saída Probabilística à Classificação Categórica

A etapa final do processo da Regressão Logística é traduzir a saída de probabilidade do modelo em uma classificação categórica definitiva.



Conforme detalhado acima, a saída da função sigmoide, σ(z), nos fornece a probabilidade estimada de a observação pertencer à classe positiva (1), que denotamos como $\hat{p}$:

$$\hat{p} = \sigma(z) = P(Y = 1 \mid X)$$

Para converter essa saída probabilística em uma classificação final (0 ou 1), aplicamos um critério de decisão. Este critério é um limiar, conhecido como **fronteira de decisão**, que é convencionalmente estabelecido em 0.5. A regra de classificação, também ilustrada na imagem, é direta:

- Se a probabilidade estimada $\hat{p}$ for **maior ou igual a 0.5** (correspondendo à metade superior da curva sigmoidal), a observação é classificada como pertencente à **classe 1**.
- Se a probabilidade estimada $\hat{p}$ for **menor que 0.5** (correspondendo à metade inferior da curva), a observação é classificada como **classe 0**.

É importante notar que, como o modelo estima a probabilidade para um evento binário, a probabilidade da **classe 0** (o não evento) é simplesmente o complemento da probabilidade da **classe 1** (o evento de interesse).

Matematicamente, a relação é expressa como:

$$P(Y = 0|X) = 1 - P(Y = 1|X)$$

Suponha que um modelo de regressão logística foi treinado para prever a probabilidade de evasão (*churn*) de um cliente com base em suas características (X), como `Renda` e `Idade`. Para um cliente específico, o modelo calcula a probabilidade de evasão como 0.8:

$$P(Churn|Renda,\ Idade) = 0.8$$

Utilizando a regra do complemento, podemos calcular facilmente a probabilidade de o mesmo cliente permanecer fiel à empresa:

$$P(Permanência|Renda,\ Idade) = 1 - 0.8 = 0.2$$

---

5. Treinamento de Um Modelo de Regressão Logística

Uma vez definida a arquitetura do modelo de Regressão Logística, o próximo passo é o seu **treinamento**. O objetivo deste processo é encontrar o conjunto de parâmetros (ou coeficientes, θ) que melhor mapeia as variáveis de entrada (X) para a variável alvo binária (Y), minimizando o erro de predição (função de custo – cost function).

O processo é iterativo e consiste nos seguintes passos:

1. **Inicialização:** Escolhe-se um conjunto inicial de parâmetros θ, frequentemente de forma aleatória.
2. **Previsão:** Utilizando os parâmetros atuais, o modelo calcula a probabilidade $\hat{p}$ para cada observação no conjunto de dados.
3. **Cálculo do Erro:** Mede-se o erro entre as probabilidades previstas e as classes reais através de uma **função de custo**.
4. **Atualização dos Parâmetros:** Os parâmetros θ são ajustados em uma direção que reduza o erro calculado.
5. **Repetição:** Os passos 2 a 4 são repetidos até que o erro atinja um valor suficientemente baixo ou um número máximo de iterações seja alcançado.

A Função de Custo para Classificação: *Log Loss*

Qualquer modelo de Regressão Logística com parâmetros iniciais pode ser considerado **preliminar**. Para encontrar o **modelo ótimo**, é necessário um processo de otimização que ajuste os parâmetros θ da melhor forma possível. Este processo, por sua vez, requer uma métrica que meça a **adequação do ajuste** (*goodness of fit*) do modelo.

Na Regressão Logística, essa métrica é uma função de custo padrão chamada ***Log Loss*** (também conhecida como Perda Logarítmica ou Entropia Cruzada Binária). A *Log Loss* mede a performance de um modelo de classificação cuja saída é uma probabilidade, quantificando a divergência entre as probabilidades previstas ($\hat{p}_i$) e as classes reais (y_i).

A fórmula é definida como:

$$J(\theta) = -\frac{1}{n}\sum_{i=1}^{n}[y_i \log(\hat{p}_i) + (1 - y_i)\log(1 - \hat{p}_i)]$$

Onde:

- n é o número de observações.
- y_i é a classe real da observação i (0 ou 1).
- $\hat{p}_i$ é a probabilidade prevista pelo modelo para a observação i de pertencer à classe 1.
- log é o logaritmo natural.

O sinal negativo no início da fórmula é necessário para garantir que a função de custo seja positiva, uma vez que o logaritmo de um número entre 0 e 1 (uma probabilidade) é sempre negativo.

Intuição por Trás da *Log Loss*

A intuição por trás da *Log Loss* é que ela **favorece previsões confiantes e corretas**, ao mesmo tempo em que **penaliza severamente as previsões confiantes e incorretas**:

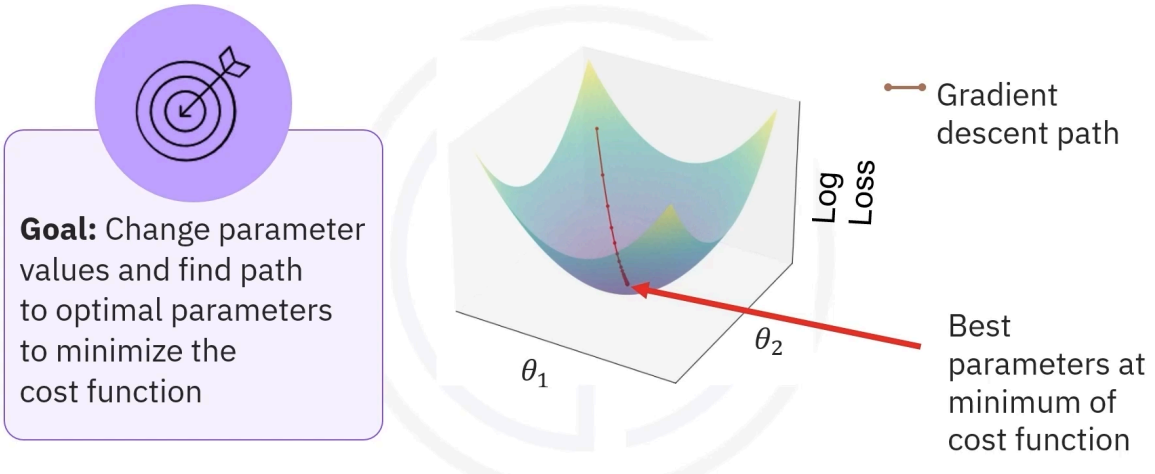
- **Quando a previsão está correta e confiante:** Se a classe real é 1 (y_i=1) e o modelo prevê uma probabilidade alta ($\hat{p}_i \approx 1$), ambos os termos da soma tendem a zero. O primeiro termo, $y_i \, log(\hat{p}_i)$, se anula porque $log(1)$ é 0, e o segundo termo é zerado pelo fator $(1-y_i)$. O custo para essa observação é, portanto, mínimo.
- **Quando a previsão está incorreta e confiante:** Se a classe real é 0 (y_i=0) mas o modelo prevê uma probabilidade alta ($\hat{p}_i \approx 1$), o primeiro termo é zerado, mas o segundo termo, $(1-y_i) \, log(1-\hat{p}_i)$, se torna um número muito grande, pois $log(1-\hat{p}_i)$ tende a $-\infty$. O custo total é, portanto, muito alto.

O objetivo do treinamento é encontrar o conjunto de parâmetros θ que **minimiza** esta função de custo. O processo iterativo de otimização continua até que o valor da *Log Loss* seja satisfatoriamente baixo ou outro critério de parada, como um número máximo de iterações, seja atingido.

Algoritmo de Otimização: Gradiente Descendente (*Gradient Descent*)

Para encontrar os parâmetros θ que minimizam a função de custo, o algoritmo de otimização mais comum é o **Gradiente Descendente**. Trata-se de uma abordagem iterativa inteligente para encontrar o ponto mínimo de uma função.

Para visualizar este processo, imagine a função de custo como uma superfície tridimensional, onde os eixos horizontais representam os valores de dois parâmetros do modelo (como θ_1 e θ_2) e o eixo vertical representa o valor do custo (o erro). O objetivo do algoritmo é "descer" por esta superfície até encontrar o ponto mais baixo do vale.



Conforme ilustrado na imagem, o processo funciona da seguinte maneira:

1. **Ponto de Partida:** O algoritmo começa em um ponto aleatório da superfície, que corresponde a um conjunto inicial de parâmetros θ.

2. **Cálculo da Direção:** Para navegar pela superfície, o algoritmo calcula o **gradiente** da função de custo nesse ponto. O gradiente é um vetor que aponta na direção da subida mais íngreme (maior **ascensão**).

3. **O Passo de Descida:** Para "descer" em direção ao mínimo, o algoritmo dá um passo na **direção oposta** à do gradiente. Daí o nome "Gradiente Descendente".

4. **Tamanho do Passo:** A magnitude de cada passo é ajustada dinamicamente. Em áreas mais íngremes da superfície, o gradiente é maior, resultando em passos mais longos. À medida que o algoritmo se aproxima do fundo do vale, a inclinação diminui, o gradiente se torna menor e os passos ficam naturalmente mais curtos, permitindo uma convergência suave. O tamanho geral dos passos é controlado por um hiperparâmetro chamado **taxa de aprendizado (*learning rate*)**.

5. **Convergência:** O processo é repetido até que o algoritmo alcance o fundo do vale, um ponto onde a inclinação e, consequentemente, o gradiente, são praticamente zero. As coordenadas desse ponto correspondem aos **parâmetros ótimos** (θ) que minimizam a função de custo.

Variações do Gradiente Descendente

A forma como o gradiente é calculado em cada passo define diferentes variações do algoritmo.

- **Gradiente Descendente em Lote (*Batch Gradient Descent*):** Na sua forma padrão, o gradiente é calculado utilizando **todo o conjunto de dados de treinamento** a cada iteração. Embora o caminho de descida seja estável e direto, este método se torna computacionalmente muito lento e inviável para *datasets* grandes.
- **Gradiente Descendente Estocástico (SGD - *Stochastic Gradient Descent*):** Para resolver o problema de escalabilidade, o SGD aproxima o gradiente utilizando apenas um **subconjunto aleatório e pequeno** dos dados (um *mini-batch*) ou até mesmo uma única observação a cada passo.
    - **Vantagens:** É muito mais rápido, escala bem para grandes volumes de dados e, devido à sua natureza mais "ruidosa", tem uma chance maior de escapar de mínimos locais e encontrar o mínimo global da função de custo.
    - **Desvantagens:** A convergência pode ser menos estável, com o algoritmo "oscilando" em torno do ponto mínimo em vez de convergir diretamente para ele. Essa oscilação pode ser controlada por técnicas como a diminuição gradual da taxa de aprendizado ao longo do treinamento.

---

6. Algoritmos de Regressão

> *Em estatística e aprendizado de máquina, **algoritmos de regressão** são uma classe de métodos supervisionados utilizados para modelar a relação entre um conjunto de variáveis independentes (preditoras ou features) e uma variável dependente (alvo ou resposta) que é de natureza **contínua e numérica**. O objetivo central não é classificar dados em categorias, mas sim **estimar ou prever um valor específico** em uma escala contínua.*

Princípios fundamentais:

1. **Objetivo Principal: Previsão de Valores Contínuos** A característica que define a regressão é a natureza de sua saída. Diferente dos algoritmos de classificação, que preveem um rótulo discreto (ex: `sim/não`, `fraude/não fraude`), os algoritmos de regressão preveem uma quantidade (ex: `preço` de um imóvel, `temperatura` de amanhã).
2. **Aprendizado de uma Função de Mapeamento** No seu núcleo, um algoritmo de regressão busca aprender uma **função de mapeamento** (f) a partir dos dados de treinamento. Esta função descreve a relação matemática entre as variáveis de entrada (X) e a variável de saída (Y), de tal forma que $Y \approx f(X)$. O processo de treinamento consiste em encontrar os parâmetros ideais para essa função.
3. **Dupla Finalidade: Predição e Inferência** A análise de regressão possui duas finalidades distintas:
    - **Predição:** O foco é a **acurácia** para fazer previsões precisas em novos dados.
    - **Inferência:** O foco é o **entendimento** da relação entre as variáveis, quantificando o impacto de cada preditor sobre o alvo.
4. **"Ajuste de Curva":** Como abordado, um algoritmo de regressão busca encontrar a função matemática (seja uma reta, uma parábola ou uma curva complexa) que melhor "se ajusta" a uma nuvem de pontos de dados, minimizando o erro geral.

Modelos Lineares nos Parâmetros

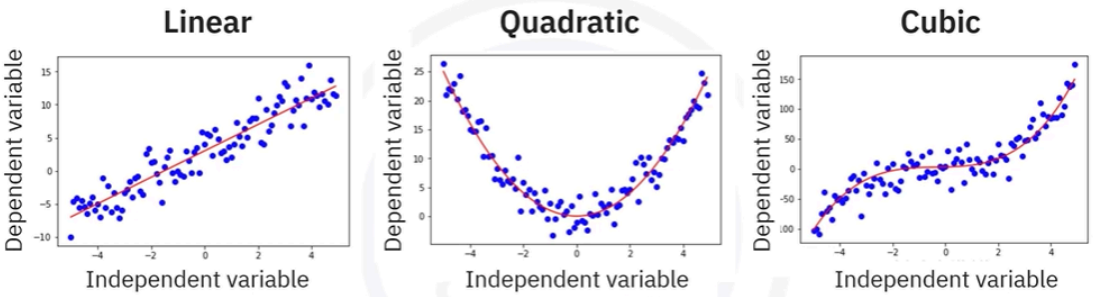
Esta categoria agrupa modelos que, apesar de poderem capturar relações curvas nos dados, possuem uma estrutura matemática que é **linear em seus parâmetros (coeficientes)**. Isso permite que sejam resolvidos com métodos eficientes como os Mínimos Quadrados Ordinários (OLS).

- **Regressão Linear:** O modelo fundamental que serve como base para esta análise, modelando relações estritamente lineares entre as variáveis.

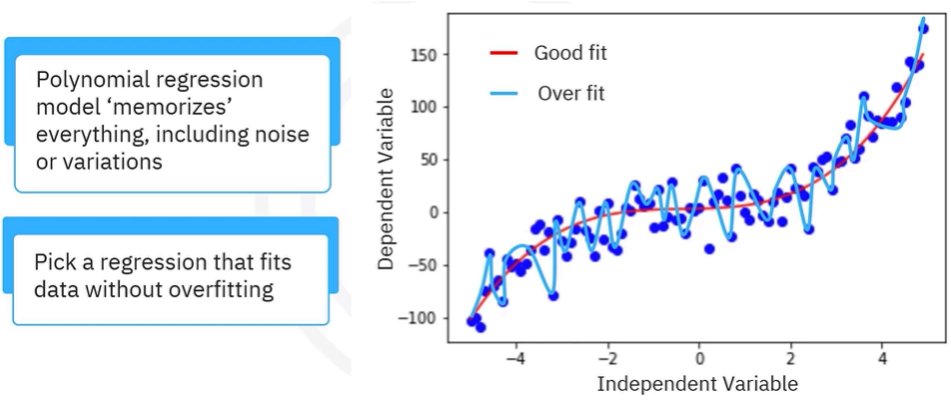
- **Regressão Polinomial:** Esta é uma forma especial de regressão que permite modelar relações não-lineares nos dados, mas que pode ser transformada em um problema de regressão linear. Isso é feito criando novas variáveis que são potências da variável original. Por exemplo, para um polinômio de grau 3 (cúbico), a equação:

$$\hat{y} = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$$

Como o modelo se torna uma combinação linear das *novas* variáveis, ele pode ser resolvido com as mesmas técnicas da regressão linear múltipla.



- **Risco de Overfitting:** Um cuidado essencial na regressão polinomial é a escolha do grau. Um polinômio de grau suficientemente alto pode passar perfeitamente por todos os pontos dos dados de treinamento. No entanto, isso significa que o modelo está "decorando" os dados, incluindo ruídos e variações aleatórias, em vez de aprender o padrão subjacente. Este fenômeno é conhecido como **overfitting** e resulta em um modelo com péssimo desempenho em dados novos.

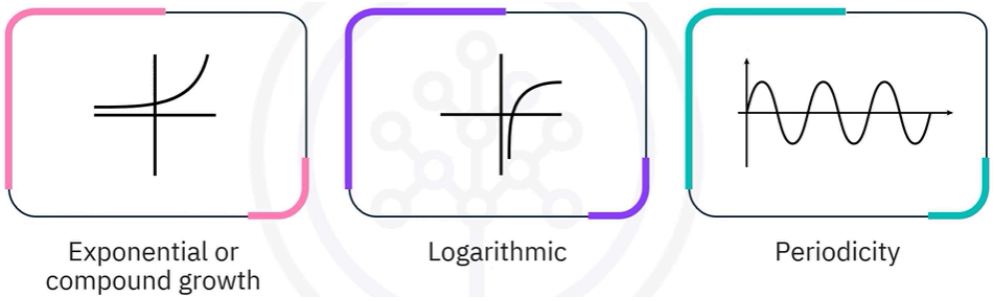


Modelos Não-Lineares nos Parâmetros

A regressão não-linear genuína ocorre quando a equação que relaciona as variáveis **não é linear em seus parâmetros**. Estes modelos não podem ser transformados em um problema linear e requerem métodos de otimização iterativos (como o Gradiente

Descendente) para encontrar os melhores parâmetros. Eles são usados para descrever relações complexas do mundo real que não seguem padrões polinomiais.

- **Regressão Exponencial:** Modela relações onde a taxa de crescimento aumenta com o tempo, como em uma curva de "J".
    - **Exemplo:** O crescimento do Produto Interno Bruto (PIB) de um país ao longo de décadas ou o crescimento de investimentos com juros compostos.
- **Regressão Logarítmica:** Descreve fenômenos de **retornos decrescentes**, onde o crescimento inicial é rápido, mas depois desacelera e se aproxima de um platô.
    - **Exemplo:** A produtividade de um trabalhador que aumenta com as horas trabalhadas, mas após um certo limite, cada hora adicional gera um ganho de produtividade menor que a anterior.
- **Regressão Periódica (Sinusoidal):** Utilizada para modelar dados que exibem ciclos ou sazonalidades regulares.
    - **Exemplo:** Variações de temperatura ou volume de chuvas ao longo dos meses do ano.



Modelos Modernos de Aprendizado de Máquina

Estes algoritmos são, em geral, mais complexos e flexíveis. Frequentemente, oferecem **maior poder preditivo**, muitas vezes ao custo de uma menor interpretabilidade direta.

- **Random Forest (Florestas Aleatórias):** Um método de *ensemble* que constrói um grande número de árvores de decisão e agrega suas predições. Esta abordagem o torna robusto e menos propenso a sobreajuste.
    - **Uso prático:** Muito eficaz em dados tabulares, é relativamente fácil de usar e oferece bom desempenho com pouca sintonia de hiperparâmetros.
- **XGBoost (Extreme Gradient Boosting):** Um algoritmo avançado que constrói árvores de decisão de forma sequencial, onde cada nova árvore corrige os erros da anterior. É amplamente reconhecido por sua eficiência e precisão.
    - **Uso prático:** Frequentemente o modelo com melhor performance em competições, mas requer uma sintonia mais cuidadosa de seus hiperparâmetros para extrair o máximo de seu potencial.
- **k-Nearest Neighbors (k-Vizinhos Mais Próximos - KNN):** Um método **não-paramétrico** que prevê o valor de um novo dado com base na média dos valores de seus 'k' vizinhos mais próximos no conjunto de treinamento.

    - **Uso prático:** É um método intuitivo e simples, mas pode se tornar lento e computacionalmente caro com grandes volumes de dados, além de ser sensível à escala das variáveis.
- **Support Vector Regression (SVR):** Uma adaptação do algoritmo de Máquinas de Vetores de Suporte (SVM) para problemas de regressão. Busca encontrar um hiperplano que melhor se ajusta aos dados, mas com uma margem de tolerância para erros.
    - **Uso prático:** É particularmente eficaz em espaços de alta dimensão (muitas *features*) e quando a relação entre as variáveis não é óbvia.
- **Neural Networks (Redes Neurais):** Inspiradas no cérebro humano, são capazes de aprender e modelar relações extremamente complexas e não-lineares. São a base do *Deep Learning*.
    - **Uso prático:** Demonstram performance de ponta em problemas com grandes volumes de dados e padrões muito complexos, como em visão computacional e processamento de linguagem natural.

Interpretabilidade e Poder Preditivo

A escolha de um algoritmo de regressão raramente se resume a selecionar o modelo com o menor erro. Uma consideração estratégica fundamental é uma relação de compromisso (*trade-off*) **entre a interpretabilidade do modelo e seu poder preditivo**.

- **Modelos Interpretáveis (Clássicos):** Modelos como a Regressão Linear são considerados "caixas brancas". Seus coeficientes nos permitem entender de forma clara e direta o impacto de cada variável no resultado final. Em um contexto de negócios, onde o **"porquê"** de uma predição é tão importante quanto a predição em si, a interpretabilidade é crucial para a tomada de decisões estratégicas.
- **Modelos de Alta Performance (Modernos):** Algoritmos como XGBoost e Redes Neurais são frequentemente tratados como "caixas pretas". Eles podem oferecer previsões extremamente acuradas, mas é muito mais difícil (às vezes impossível) discernir a contribuição exata de cada variável individual. O foco aqui é a precisão do **"o quê"** será previsto.

Portanto, a seleção do algoritmo ideal depende intrinsecamente do objetivo do projeto: se a meta é **entender os fatores e suas relações (inferência)**, um modelo mais simples e interpretável é geralmente a melhor escolha. Se o objetivo é puramente a **máxima acurácia preditiva**, modelos mais complexos são os mais indicados

---

Resumo

- A regressão modela as relações entre uma **variável alvo contínua** e **variáveis explicativas**, abrangendo os tipos de **regressão simples** e **múltipla**.

- A **regressão simples** utiliza uma única variável independente para estimar uma variável dependente, enquanto a **regressão múltipla** envolve mais de uma variável independente.

- A regressão possui ampla aplicabilidade, desde **previsão de vendas** e **estimativa de custos de manutenção** até a **previsão de chuvas** e a **propagação de doenças**.

- Na **regressão linear simples**, uma **reta de melhor ajuste** minimiza os erros, medidos pelo **Erro Quadrático Médio (MSE)**; essa abordagem é conhecida como **Mínimos Quadrados Ordinários (OLS)**.

- A regressão OLS é **fácil de interpretar**, mas **sensível a outliers**, que podem impactar a precisão.

- A **regressão linear múltipla** estende a regressão linear simples ao utilizar várias variáveis para prever resultados e analisar as relações entre elas.

- No entanto, adicionar muitas variáveis pode levar ao **overfitting**, por isso é necessária uma **seleção criteriosa de variáveis** para construir um modelo equilibrado.

- A **regressão não linear** modela relações complexas utilizando funções **polinomiais, exponenciais ou logarítmicas**, quando os dados não se ajustam a uma linha reta.

- A **regressão polinomial** pode se ajustar bem aos dados, mas corre o risco de **overfitting**, capturando ruídos aleatórios em vez de padrões reais.

- A **regressão logística** é um **preditor de probabilidades** e um **classificador binário**, adequada para alvos binários e para avaliar o impacto das variáveis explicativas.

- A regressão logística minimiza erros usando o **log-loss** e otimiza o ajuste com **gradiente descendente** ou **gradiente descendente estocástico**, visando maior eficiência.

- O **gradiente descendente** é um processo iterativo para minimizar a **função de custo**, essencial para o treinamento de modelos de regressão logística.