Aprendizagem automática

Sessão 11 - T

Modelos lineares

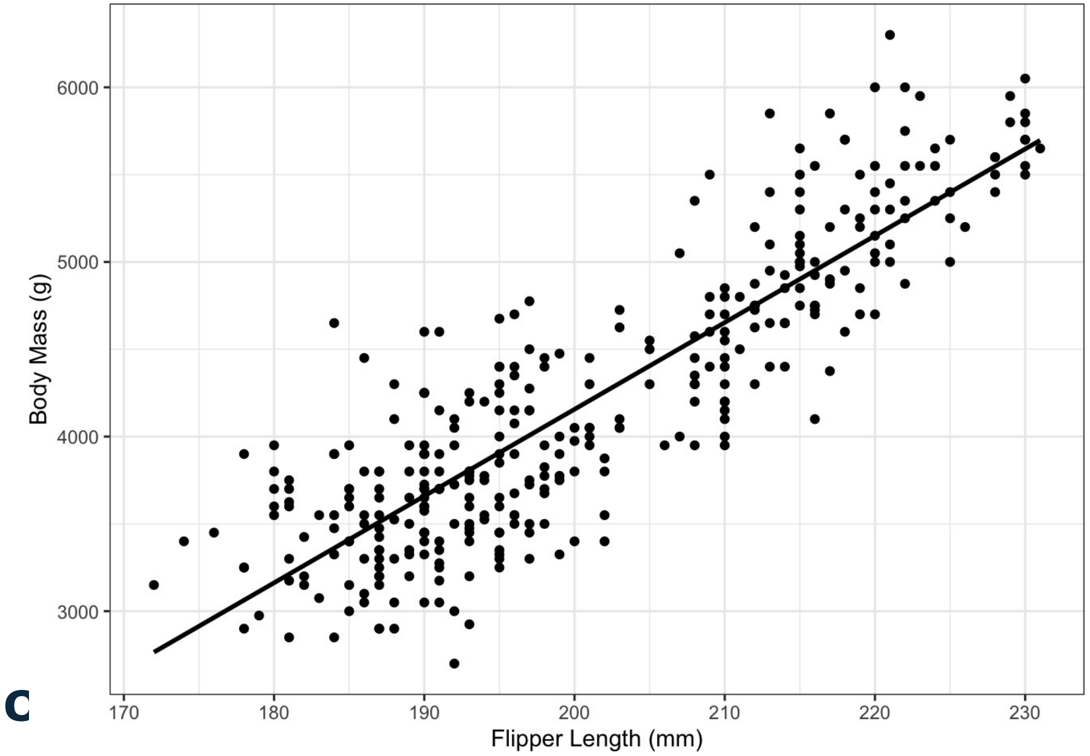
Ciência de Dados Aplicada

2023/2024

Modelos lineares

- Os modelos lineares são uma classe de algoritmos de aprendizagem automática que modelam a relação entre variáveis dependentes e independentes utilizando **uma aproximação linear**.

- **Características principais:**
  - **Fácil de interpretar e aplicar**;
  - **Eficiente** para grandes conjuntos de dados devido devido à computação linear;
  - Pode ser utilizado tanto para **regressão** ... **tarefas de classificação**.



Modelos lineares

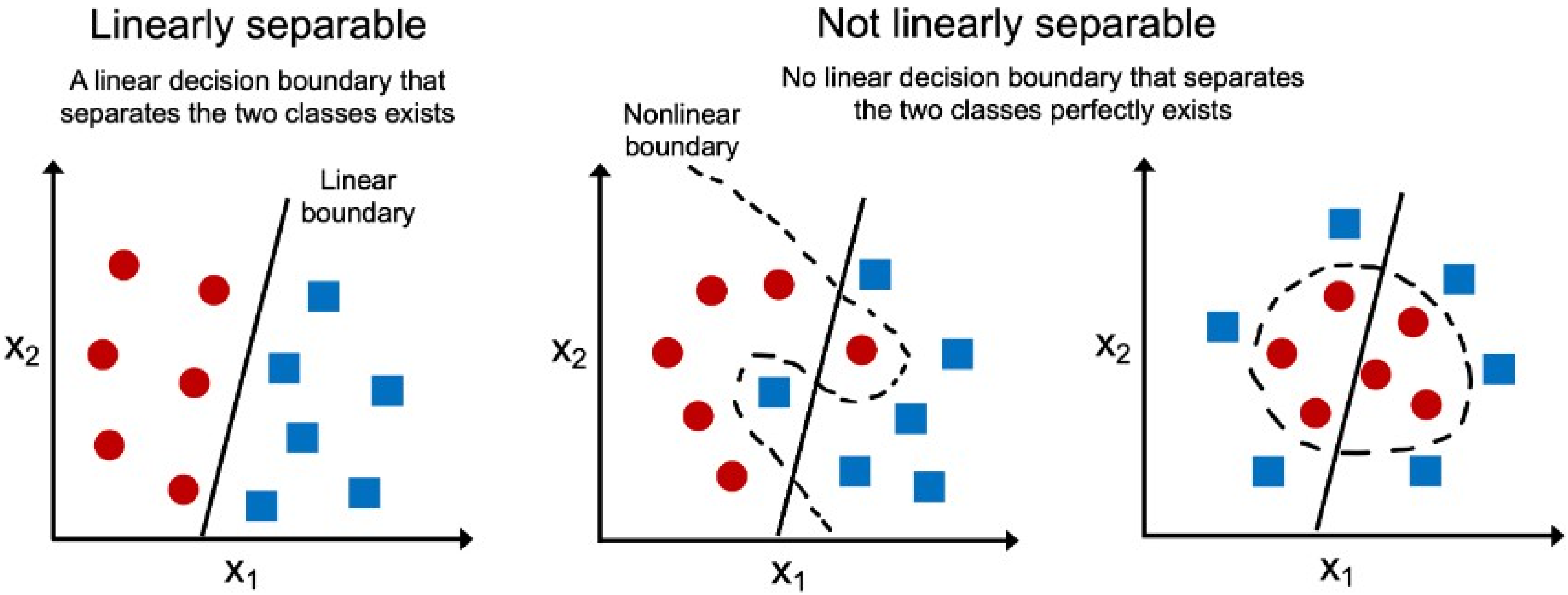
- Um modelo linear é uma **representação matemática** de uma relação entre uma variável dependente y e uma ou mais variáveis independentes $x_1, x_2, ..., x_n$, assumindo-se **que** esta **relação é linear**.

- Matematicamente, representa-se da seguinte forma:

$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n + \epsilon$ Onde y é a **variável dependente**, $x_1, x_2, ..., x_n$ são as **variáveis independentes**, β_0 é a **interceção**, $\beta_1, \beta_2, ..., \beta_n$ são os **coeficientes** e ϵ é o **termo de erro**.

Problemas não lineares

UNIVERSIDADE CATÓLICA PORTUGUESA
BRAGA



https://vitalflux.com/how-know-data-linear-non-linear/

- **Limitações:** Em casos não lineares, os modelos lineares podem ser insuficientes...

Modelos lineares

- **Regressão Linear Simples**

- **Regressão Linear Múltipla**

- **Regressão polinomial**

- **Regressão logística**

Regressão Linear Simples

- A regressão linear simples modela a relação entre **uma variável independente e uma variável dependente**. A equação para uma regressão linear simples é:

A interceção - uma constante

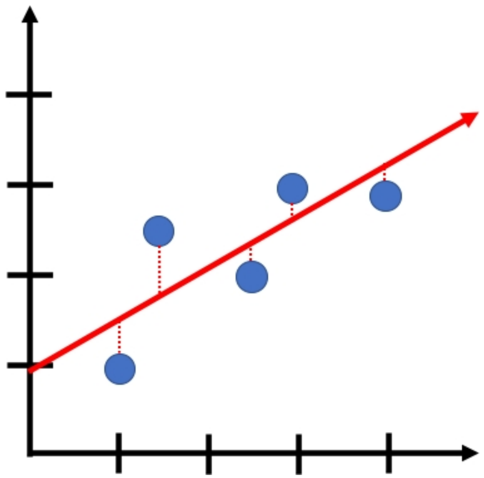
Variável independente

$$y = \beta_0 + \beta_1 x + \epsilon$$

Componente de erro

Variável dependente

Coefficient\Weight\Slope - controla a contribuição de x para y



Regressão Linear Múltipla

- A regressão linear múltipla alarga a regressão linear simples para incorporar **múltiplas variáveis independentes**. A equação para a regressão linear múltipla é:

A interceção - uma constante

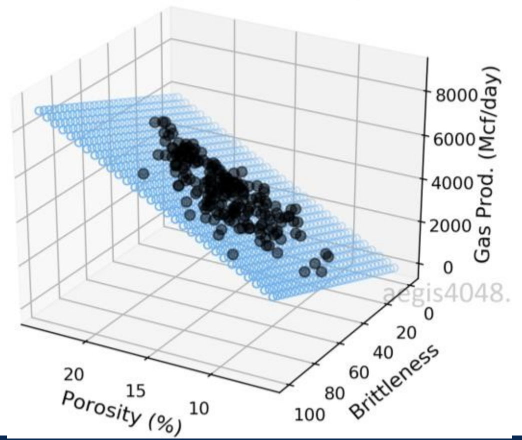
Variáveis independentes

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n + \epsilon$$

Componente de erro

Variável dependente

Coefficients\Weights\Slopes - controla a quantidade de cada
x contribui para y



Regressão polinomial

UNIVERSIDADE CATÓLICA PORTUGUESA
BRAGA

- A regressão polinomial consiste em ajustar uma curva aos dados através da introdução de
  termos polinomiais. A equação para a regressão polinomial é:

Termos polinomiais

$$y = \beta_0 + \beta_1 x_1^1 + \beta_2 x_2^2 + \dots + \beta_n x_n^n + \epsilon$$

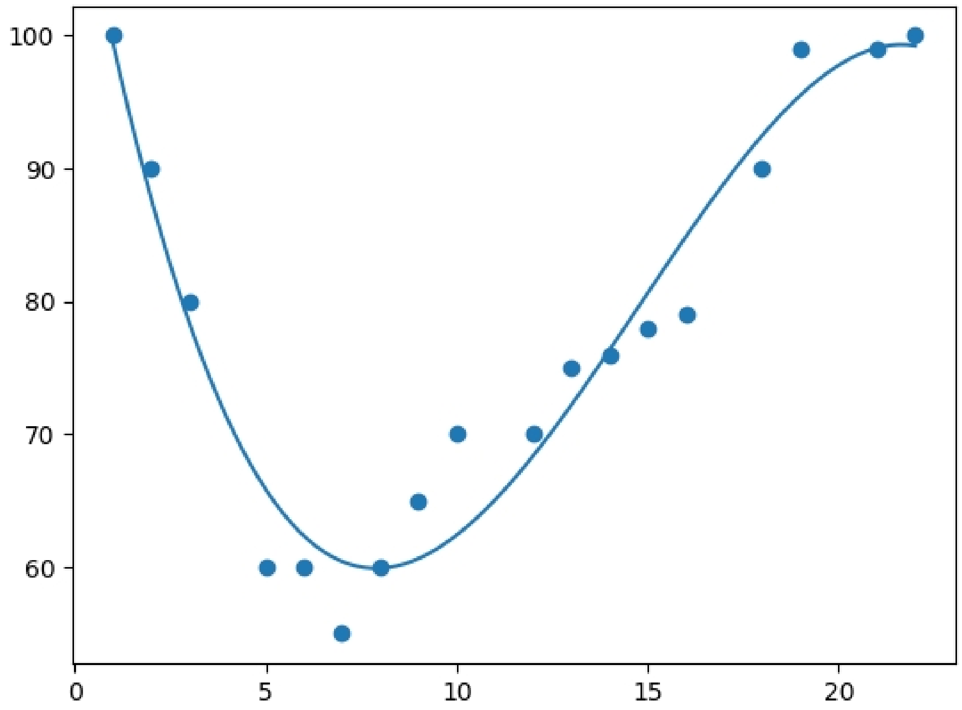
- NOTA: O termo "linear" na regressão linear refere-se à **linearidade dos coeficientes**, não necessariamente à linearidade da relação entre as variáveis independentes e dependentes.

- Na regressão polinomial, embora a **relação entre as variáveis seja modelada através de uma função polinomial** (que pode parecer curva), o processo de estimativa continua a ser linear em relação aos coeficientes.

Regressão polinomial

Termos polinomiais

$$y = \beta_0 + \beta_1 x_1^1 + \beta_2 x_2^2 + \ldots + \beta_n x_n^n + \epsilon$$



Regressão linear

- **Função de custo:** Erro quadrático médio (MSE)

$$J(\beta) = \frac{1}{2m} \sum_{i=1}^{m} \left(\hat{y}^{(i)} - y^{(i)} \right)^2$$

Valor previsto dado por:

Valor real

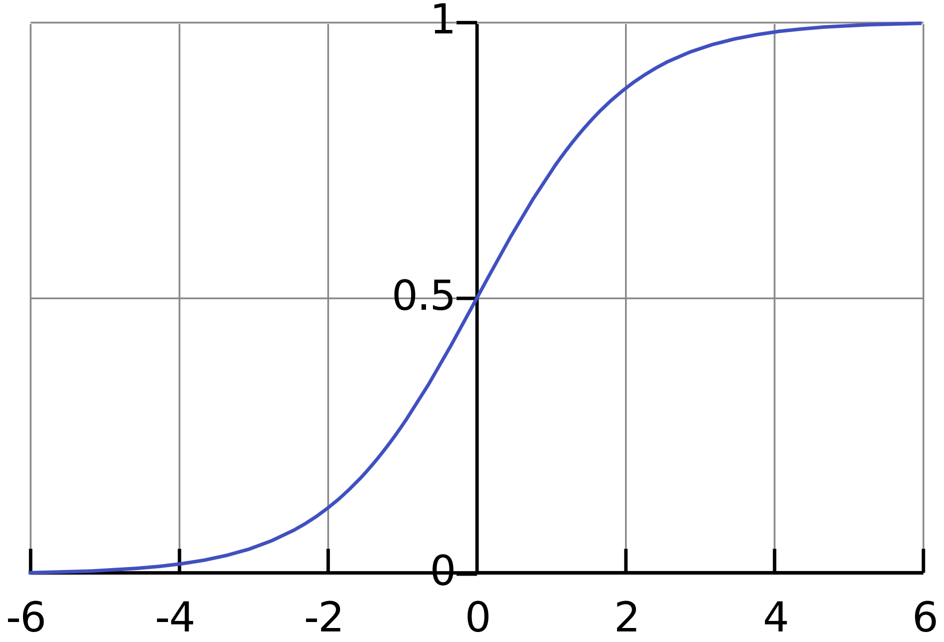
$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n$$

- **J** é uma função dos c o e f i c i e n t e s do modelo $\beta_1, \beta_1, \ldots, \beta_n$
- Objetivo: identificar os coeficientes do modelo que **minimizam J**

Regressão logística

- A regressão logística é utilizada para a **classificação binária**.

- Modela a **probabilidade de o resultado pertencer a uma determinada classe** utilizando a função logística, também conhecida como **função sigmoide**.

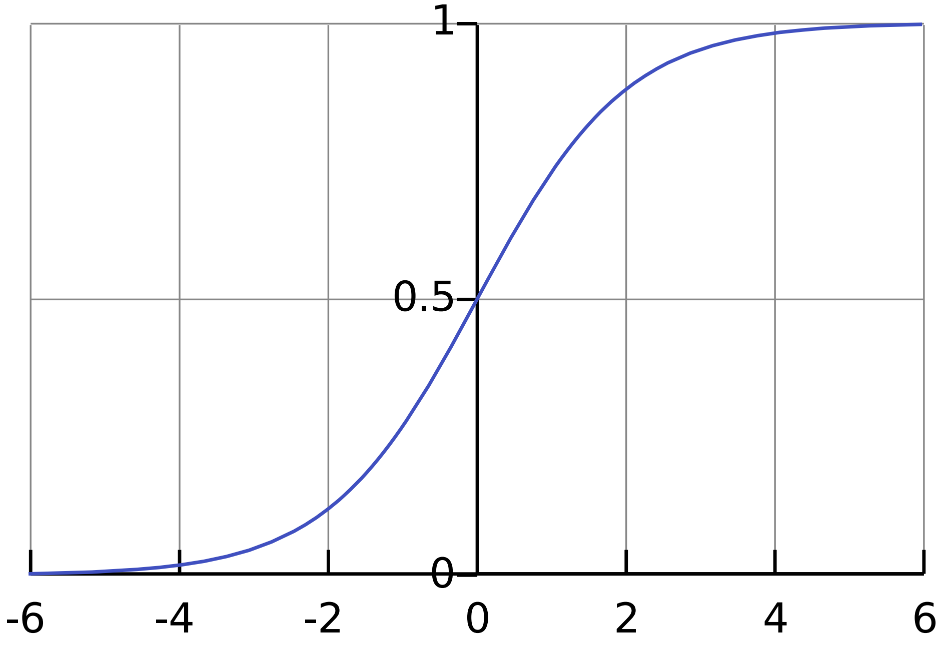


$$f(x) = \frac{1}{1+e^{-(x)}}$$

Regressão logística

- A classe prevista é dada pela **aplicação da função sigmoide ao resultado da regressão linear**.

- Dá-nos a **probabilidade** de y (resultado) ser 1 para o exemplo x.



$$f(x) = \frac{1}{1+e^{-(x)}}$$

$$p = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n)}}$$

Regressão logística

- Treinar **modelos "binários" separados para cada classe**;
- Tratar **as outras classes como uma entidade única** durante a formação;
- **Cada modelo estima a probabilidade** de o exemplo pertencer a uma classe;
- Aplicar todos modelos, **selecionar classe commais elevada probabilidade prevista.**

Regressão logística - Multiclasse

- **Função de custo**:

$$J(\beta) = \begin{cases} -\log(\hat{y}) & \text{if} \quad y = 1 \\ -\log(1-\hat{y}) & \text{if} \quad y = 0 \end{cases}$$

- **Exemplo:**
  - Se rótulo verdadeiro y=1, previsão=0,8, erro = -log(0,8) , erro baixo
  - Se rótulo verdadeiro y=0, previsão=0,8, erro = -log(1 - 0,8), erro elevado

- **Combinando, obtemos:**

$$J(\beta) = \frac{1}{m} \sum_{i=1}^{m} \left[-y^{(i)} \log(\hat{y}^{(i)}) - (1 - y^{(i)}) \log(1 - \hat{y}^{(i)}) \right]$$

$$p = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n)}}$$

Valor previsto dado por:

Valor real

Estimativa dos parâmetros (coeficientes) β

UNIVERSIDADE CATÓLICA PORTUGUESA
BRAGA

- A estimativa dos parâmetros consiste em determinar os **valores dos coeficientes** ($\beta_1, \beta_2, ..., \beta_n$) que **melhor ajustam o modelo** aos dados de treino.

- Este processo envolve encontrar o parâmetros parâmetros que **minimizar uma função de perda predefinida**.

- Para modelos lineares, podemos utilizar o **método analítico** dos **mínimos quadrados**, minimizando a função de erro **MSE**.

- Podem também ser utilizados outros **métodos iterativos**, como a **descida do gradiente.**

Regressão Linear - Mínimos Quadrados

- Método algébrico que envolve a resolução de um **sistema de equações**:

$$\frac{\partial}{\partial \beta_j} J(\beta) = 0, j = 1, ..., n$$

$$\beta = (X^T X)^{-1} X^T y$$

Versão matricial
X consiste nos dados de treino
+
primeira coluna com uns (para ter em conta o beta zero)

Regressão linear\Logística - Gradiente Descendente

UNIVERSIDADE CATÓLICA PORTUGUESA

BRAGA

- Só pode ser aplicado se a **função de custo for diferenciável**.

- **Método iterativo** que, em cada iteração, altera os valores dos parâmetros de modo a minimizar o erro entre as previsões e as etiquetas verdadeiras.

$$\beta_j := \beta_j - \alpha \frac{\partial}{\partial \beta_j} J(\beta)$$

Taxa de aprendizagem

Os parâmetros são actualizados da seguinte

$$\beta_j := \beta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (\hat{y}^{(i)} - y^{(i)}) x_j^{(i)}$$

forma:

Mínimos quadrados vs Gradiente Descendente

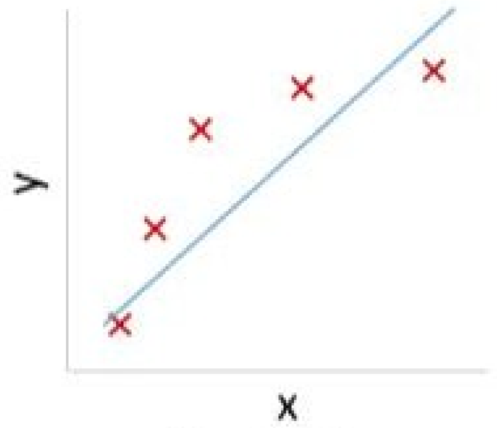
UNIVERSIDADE CATÓLICA PORTUGUESA
BRAGA

- Os mínimos quadrados garantem **uma solução óptima**. A descida de gradiente pode
  não convergem / ficam presos em óptimos locais.

- Os mínimos quadrados são adequados e **computacionalmente** eficientes para pequenos conjuntos de dados;

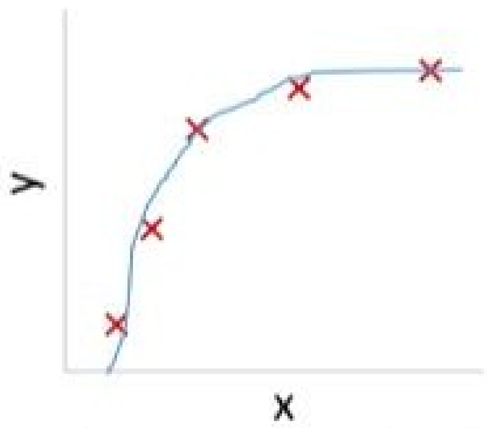
- A descida de gradiente é adequada para **grandes conjuntos de dados** e pode tratar
  modelos não lineares.

Sobreajuste em modelos lineares



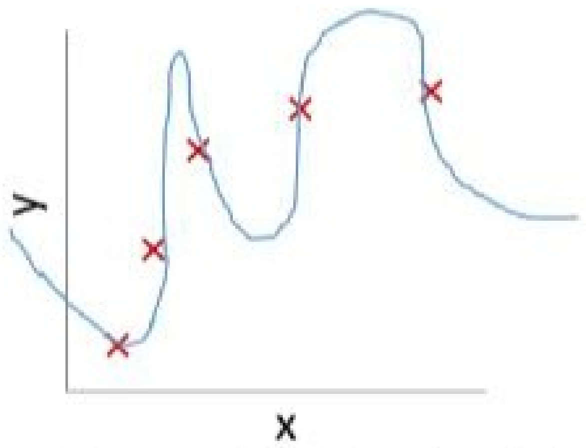
$$\theta_0 + \theta_1 x$$

Underfitting:
Insufficient model complexity



$$\theta_0 + \theta_1 x + \theta_2 x^2$$

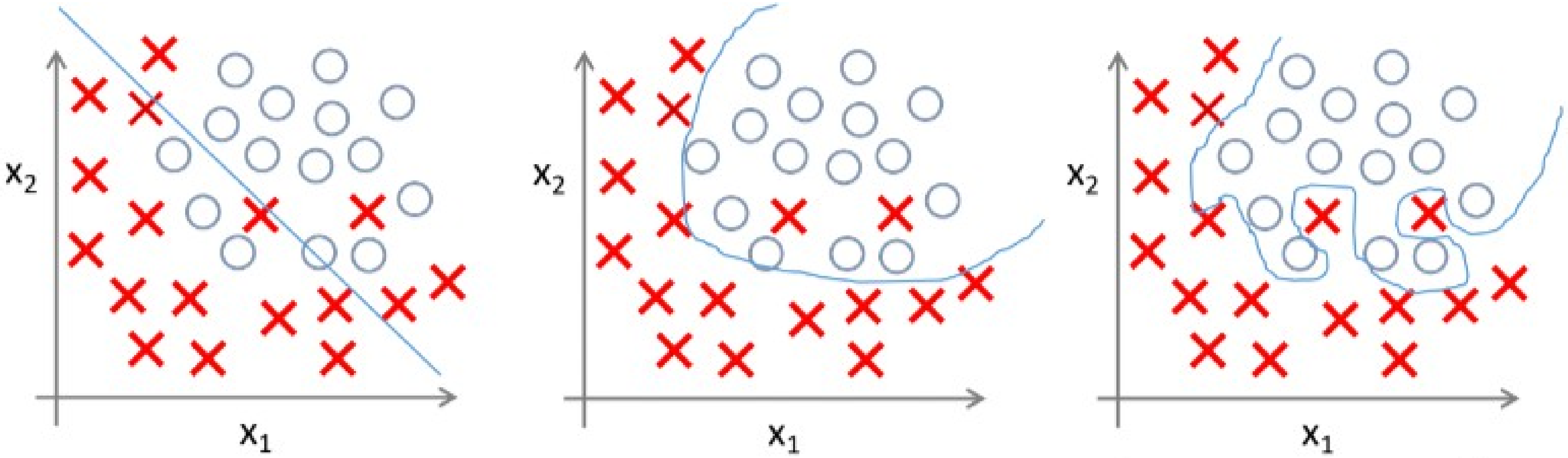
"Adequate" model complexity



$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

Overfitting:
Excessive complexity

Sobreajuste em modelos lineares



$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$

g = sigmoid function

Underfitting:
Insufficient model complexity

$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 \\ + \theta_3 x_1^2 + \theta_4 x_2^2 \\ + \theta_5 x_1 x_2)$$

“Adequate” model complexity

$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 \\ + \theta_3 x_1^2 x_2 + \theta_4 x_1^2 x_2^2 \\ + \theta_5 x_1^2 x_2^3 + \theta_6 x_1^3 x_2 + \dots)$$

Overfitting:
Excessive complexity

Sobreajuste em modelos lineares

- Nos modelos lineares, como a regressão linear simples ou a regressão linear múltipla, **o sobreajuste não envolve curvas ou superfícies complexas que se dobram para se ajustarem aos dados**.

- Em vez disso, implica que o modelo linear **capte o ruído ou flutuações irrelevantes nos dados**, o que pode acontecer quando o modelo é demasiado complexo em relação à quantidade de dados disponíveis.

- Os modelos lineares são propensos ao sobreajuste nos casos em que o **número de características é elevado** (especialmente

Sobreajuste em modelos lineares

quando comparado com o número de exemplos).

Ultrapassar o sobreajuste em modelos lineares

UNIVERSIDADE CATÓLICA PORTUGUESA
BRAGA

- Reduzir o número de características (coeficientes) - seleção de características.

- Regularização: Mantém todas as características, tentando reduzir a magnitude dos valores dos parâmetros.

  - Regularização L1 (**regressão Lasso**)
  - Regularização L2 (**refressão de cumeeira**)
  - **Redes elásticas** - utilizar a regularização L1 e L2

Regressão de cumeeira

- Ideia: **penalizar valores elevados dos parâmetros** (coeficientes) na função de custo.

$$J(\beta) = \frac{1}{2m}\left[\sum_{i=1}^{m}(\hat{y}(i) - y^{(i)})^2 + \lambda \sum_{j=1}^{n} \beta_j^2\right]$$

Parâmetro de regularização:
- valores mais elevados penalizam mais os valores dos parâmetros

Se for demasiado elevado: risco de subadaptação Se for demasiado baixo: risco de sobreadaptação

Regressão de cumeeira

- **Método analítico (mínimos quadrados):**

$$\beta = \left(X^T X + \lambda \begin{bmatrix} 0 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix} \right)^{-1} X^T y$$

Regressão de cumeeira

- **Descida de gradiente:**

$$\beta_0 := \beta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(\hat{y}^{(i)} - y^{(i)} \right) x_0^{(i)}$$

$$\beta_j := \beta_j (1 - \alpha \frac{\lambda}{m}) - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(\hat{y}^{(i)} - y^{(i)} \right) x_j^{(i)}$$

$$j = 1, \ldots, n$$

Termo imposto pela regularização É sempre < 1

Regressão Lasso

UNIVERSIDADE CATÓLICA PORTUGUESA
BRAGA

- Ideia: em vez de penalizar os valores quadrados dos parâmetros, penaliza os **valores absolutos**

$$J(\beta) = \frac{1}{2m}\left[\sum_{i=1}^{m}(\hat{y}(i) - y^{(i)})^2 + \lambda\sum_{j=1}^{n}\beta_j^2\right]$$ **Cumeeira**

$$J(\beta) = \frac{1}{2m}\left[\sum_{i=1}^{m}(\hat{y}(i) - y^{(i)})^2 + \lambda\sum_{j=1}^{n}|\beta_j|\right]$$ **Laço**

- Tudo o resto fica na mesma!

Regularização na regressão logística

- Função de custo:

Regularização de cumeeira

$$J(\beta) = \left[-\frac{1}{m}\sum_{i=1}^{m} y^{(i)} \log\left(\hat{y}^{(i)}\right) + \left(1 - y^{(i)}\right) \log(1 - \hat{y}^{(i)})\right] + \frac{\lambda}{2m}\sum_{j=1}^{n} \beta_j^2$$

$$J(\beta) = \left[-\frac{1}{m}\sum_{i=1}^{m} y^{(i)} \log\left(\hat{y}^{(i)}\right) + \left(1 - y^{(i)}\right) \log(1 - \hat{y}^{(i)})\right] + \frac{\lambda}{2m}\sum_{j=1}^{n} |\beta_j|$$

Regularização Lasso

Regularização na regressão logística

Gradiente

$$\frac{\partial}{\partial \beta_0} J(\beta) \qquad \frac{1}{m} \sum_{i=1}^{m} (\hat{y}^{(i)} - y^{(i)}) x_0^{(i)}$$

$$\frac{\partial}{\partial \beta_1} J(\beta) \qquad \frac{1}{m} \sum_{i=1}^{m} (\hat{y}^{(i)} - y^{(i)}) x_1^{(i)} - \frac{\lambda}{m} \beta_1$$

$$\frac{\partial}{\partial \beta_2} J(\beta) \qquad \frac{1}{m} \sum_{i=1}^{m} (\hat{y}^{(i)} - y^{(i)}) x_2^{(i)} - \frac{\lambda}{m} \beta_2$$

(...) (...)

$$\frac{\partial}{\partial \beta_j} J(\beta) \qquad \frac{1}{m} \sum_{i=1}^{m} (\hat{y}^{(i)} - y^{(i)}) x_j^{(i)} - \frac{\lambda}{m} \beta_j$$

Recursos

- Rencher, A. C., C Schaalje, G. B. (2007). Linear Models in Statistics (2nd ed.) [PDF]. doi:10.1002/9780470192610

- Pillonetto, G., Chen, T., Chiuso, A., De Nicolao, G., C Ljung, L. (2022). Regularização de Modelos de Regressão Linear. Em Regularized System Identification (pp. 33-93). Springer International Publishing. https://doi.org/10.1007/978-3-030-95860-2_3

- Tran-Dinh, Q., C van Dijk, M. (2022). Métodos do tipo gradiente descendente: Antecedentes e Análise de Convergência Unificada Simples (Versão 1). arXiv. https://doi.org/10.48550/ARXIV.2212.09413