



Data Science & Business Analytics

Machine Learning Models

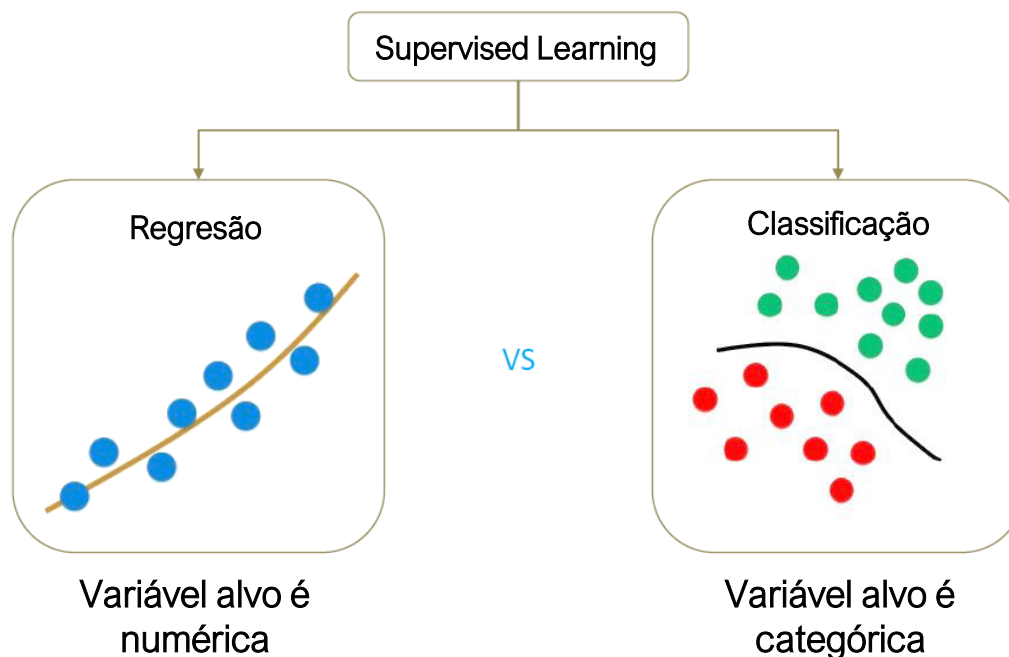
David Issá

davidribeiro.issa@gmail.com

1. Supervised Learning

1. Supervised Learning – Definição

Corresponde a um conjunto de técnicas de machine learning que utilizam um **conjunto de dados rotulados** (labeled data) para treinar algoritmos com o **objetivo de classificar dados ou prever resultados**.



1. Supervised Learning – Definição

Regressão



Prever o preço de uma casa

ID	m2	Localização	Tipo	WCs	Preço
1	234	Restelo	T5	3	1.112.000
2	107	Campolide	T2	2	365.000
3	67	Alfama	T1	1	240.000
4	86	Alavalade	T2	2	320.000
5	102	Campolide	T3	1	330.000
6	78	Benfica	T2	1	295.000
7	104	Areeiro	T2	2	367.000
8	122	Benfica	T3	1	?

1. Supervised Learning – Definição

Classificação



Prever probabilidade de ter diabetes

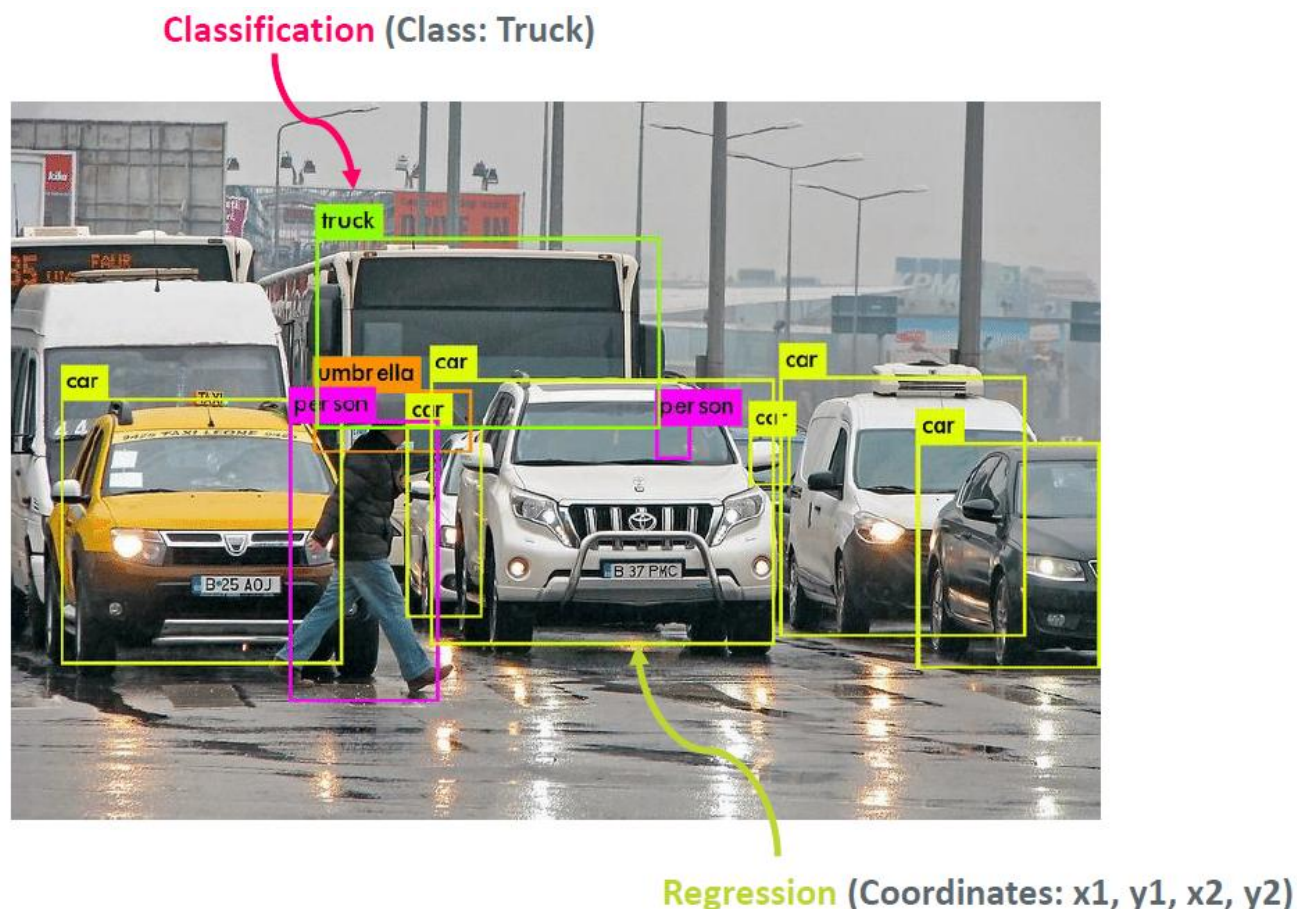
ID	Género	Peso	Altura	Gravidezes	Status
1	M	78	175	0	Sem Diabetes
2	F	66	155	3	Diabetes
3	F	91	165	1	Diabetes
4	M	89	187	0	Sem Diabetes
5	M	101	172	0	Diabetes
6	M	81	179	0	Sem Diabetes
7	F	72	169	0	Sem Diabetes
8	F	93	169	0	?

1. Supervised Learning – Definição

É possível existir uma tarefa de classificação e outra de regressão no mesmo modelo?



1. Supervised Learning – Definição

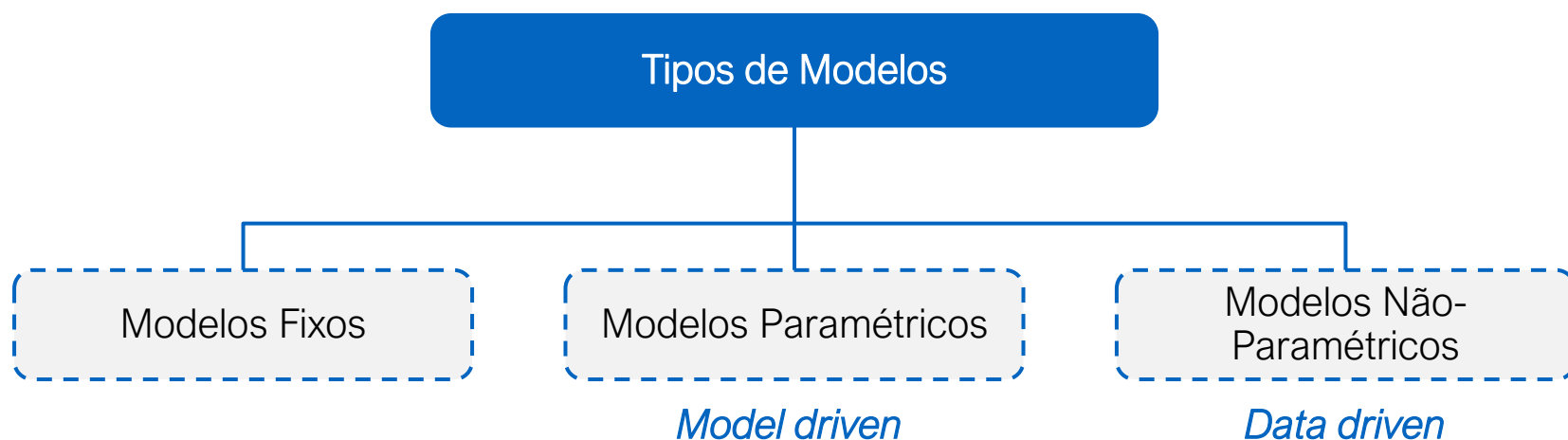


2. O que são modelos?

2. Modelos

- Como em qualquer outra tarefa informática, **a modelação requer um “programa”** que forneça instruções pormenorizadas;
- Estas instruções são tipicamente **equações matemáticas, que caracterizam a relação entre inputs e outputs**;
- A formulação destas equações é o problema central da modelação.

2. Modelos



2. Modelos – Modelos Fixos



- Equações de forma fechada que definem como os outputs são derivados dos inputs;
- A **relação entre um determinado input e o seu respetivo output é fixa**;
- **Adequados para problemas simples** e perfeitamente compreensíveis.

Exemplo: calcular quanto tempo uma maçã demora a atingir o solo na Terra.

$$t = \sqrt{\frac{2h}{9.8}}$$

2. Modelos – Modelos Paramétricos



- Sabemos o suficiente sobre o problema para definir a relação geral entre inputs e outputs;
- Mas **alguns parâmetros não são especificados**: esses são estimados examinando através de um conjunto de dados existente.

Exemplo: calcular quanto tempo uma maçã demora a atingir o solo em Marte.

$$t = \sqrt{\frac{2h}{g}}$$

Precisamos de estimar o parâmetro g !

2. Modelos – Modelos Paramétricos



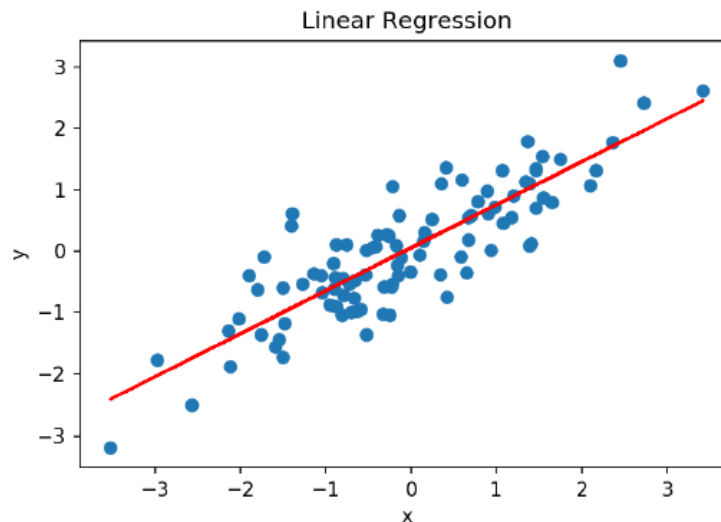
Precisamos de estimar o parâmetro g !

Teríamos de ir a Marte e recolher uma amostra:

Altura (h)	Tempo da queda (t)
0.5	0.2
1.3	0.4
1.8	0.46
4	0.68
7.3	0.7

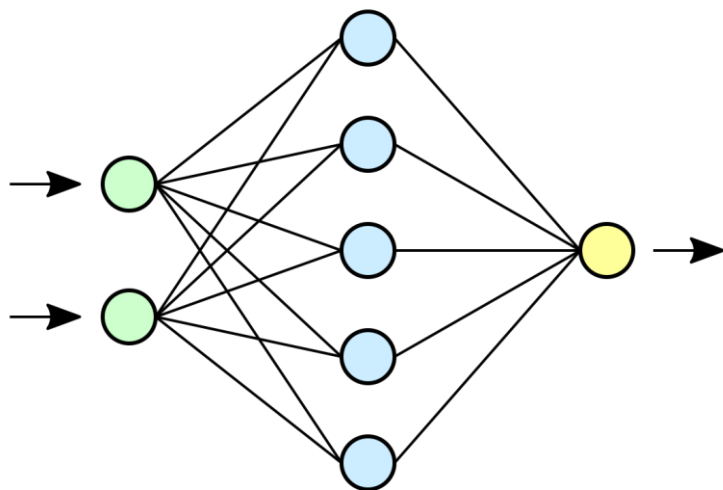
2. Modelos – Modelos Paramétricos

Um **exemplo comum de um modelo paramétrico é a regressão linear**. O pressuposto é que existe uma relação linear entre inputs e outputs:



$$y_i = \beta_0 + \beta_1 x_{i1} + \dots + \beta_k x_{ik} + \varepsilon_i, i = 1, \dots, n$$

2. Modelos – Modelos Não-Paramétricos



- Modelos que se baseiam em dados, em vez de em conhecimentos humanos.
Designados por modelos data driven;
- Utilizados em problemas complexos em que a relação entre inputs e outputs não é conhecida;
- Requer grandes quantidades de dados;
- Uma vantagem importante é o facto de os modelos não paramétricos não exigirem um conhecimento profundo do problema.

2. Modelos – Modelos Não-Paramétricos

Modelos Paramétricos



Simplicidade: fácil de perceber e interpretar.

Performance: rápido a treinar e aprender dos dados de input.

Quantidade de dados: não requer uma elevada quantidade de dados para treinar o modelo.

Regressão Linear

Regressão Logística

Naive Bayes

...



Limitado: forma funcional da função limita as possibilidades de modulação e de aumento de complexidade.

Fit do modelo: na maioria dos casos, não oferece o melhor ajuste aos dados.

Pré-processamento: é necessário transformar os dados para que estes correspondam a uma distribuição específica.

2. Modelos – Modelos Não-Paramétricos

Modelos Não-Paramétricos



Flexibilidade: Capaz de se ajustar a um grande número de formas funcionais.

Resultados: pode resultar numa maior taxa de acerto.

Pré-processamento: não há pressupostos sobre a distribuição dos dados, poupando-se tempo no pré-processamento.



Quantidade de dados: requer mais dados do que os modelos paramétricos.

Performance: mais lento devido à maior complexidade dos modelos.

Interpretabilidade: mais difícil de interpretar a razão de certas predições serem feitas por em alguns algoritmos menos intuitivos.

Overfitting: maior risco de fazer overfitting nos dados de treino.

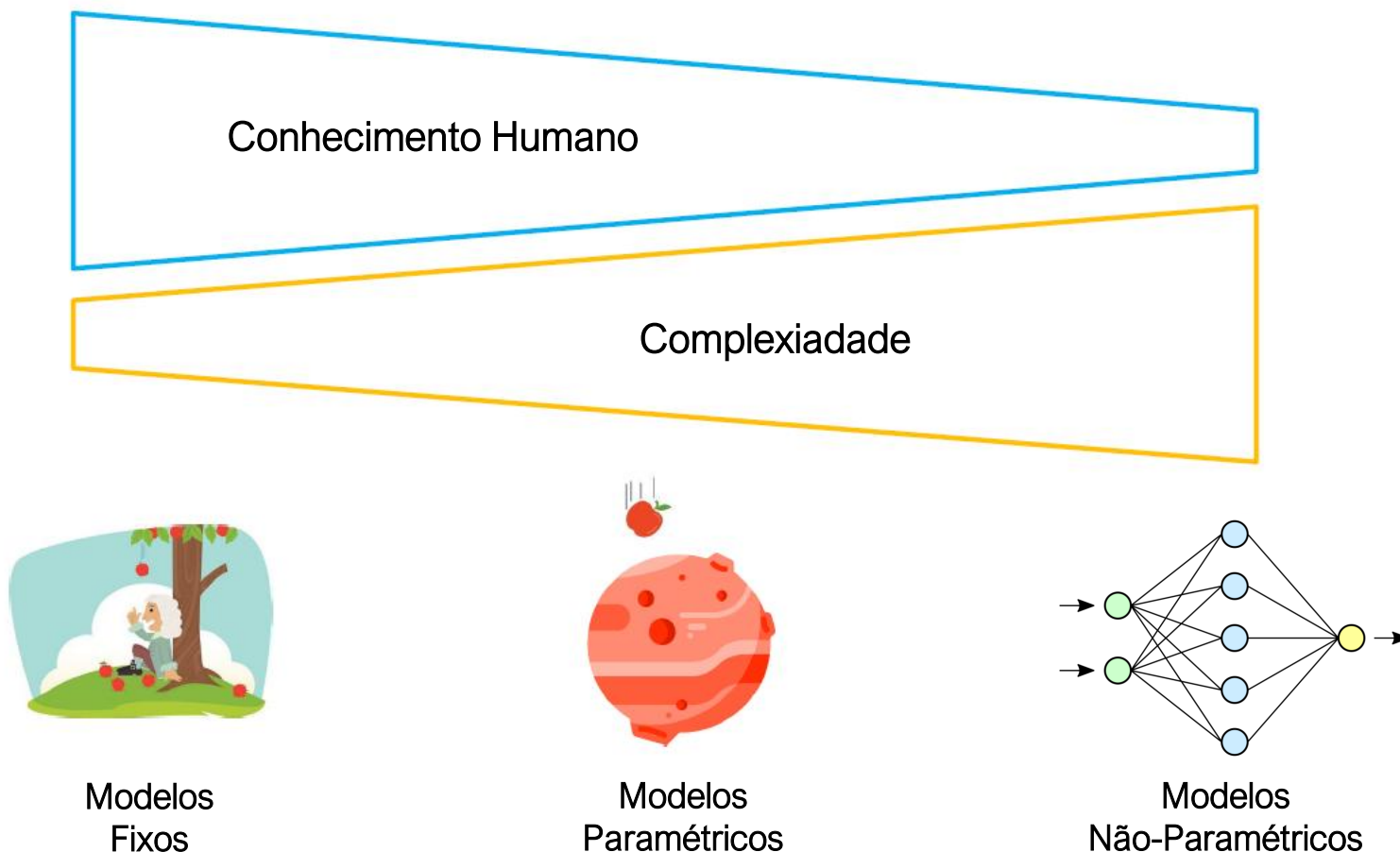
KNN

Árvores de decisão

Redes neuronais

...

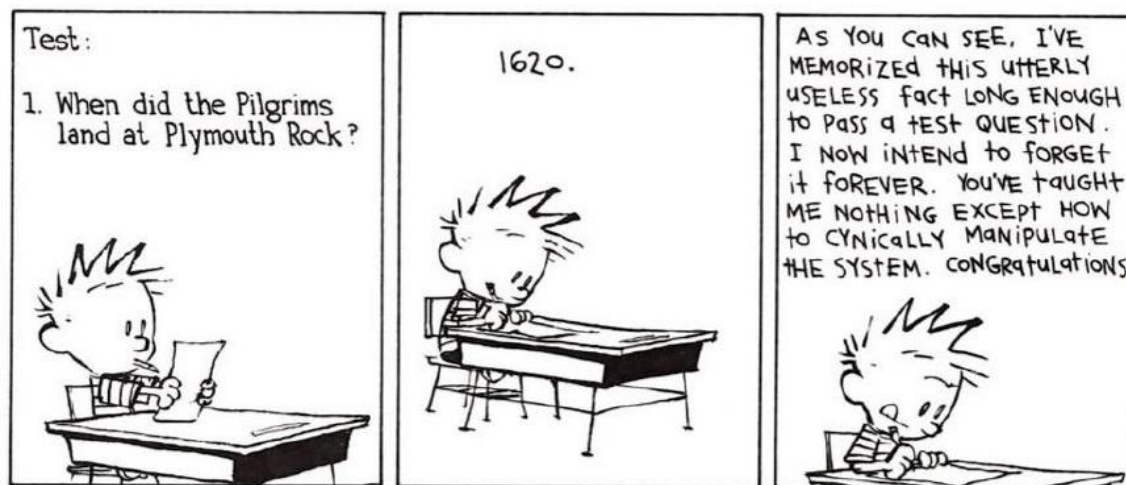
2. Modelos – Modelos Não-Paramétricos



2. Modelos – Overfitting

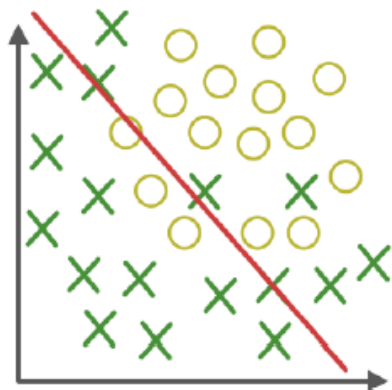
Overfitting corresponde a uma situação onde o **modelo depende demasiado dos dados de treino para aprender**.

Se permitirmos demasiada complexidade, o modelo irá “memorizar” os dados de treino, em vez de extrair relações úteis para observações “não vistas”.

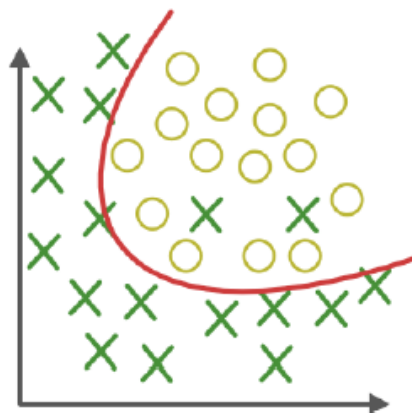


2. Modelos – Overfitting

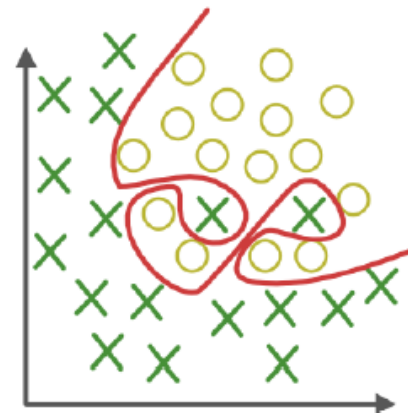
Num problema de classificação:



Underfitting
(solução demasiado simples para “separar” os dados)



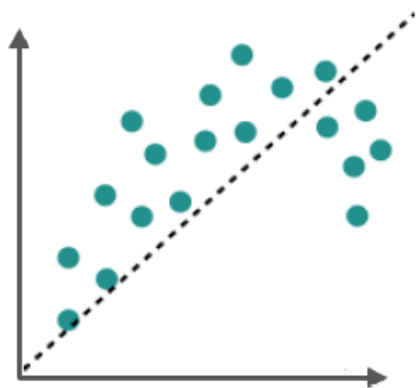
Fit apropriado



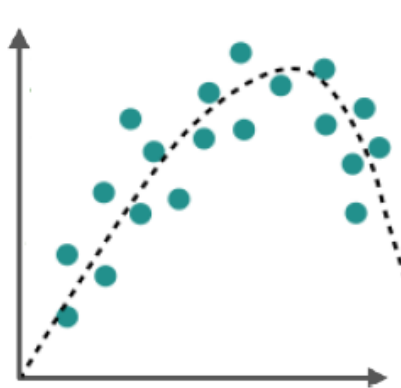
Overfitting
(solução “cola” demasiado aos dados de treino)

2. Modelos – Overfitting

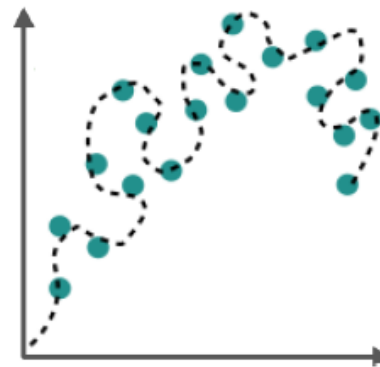
Num problema de regressão:



Underfitting
(solução demasiado simples para “separar” os dados)



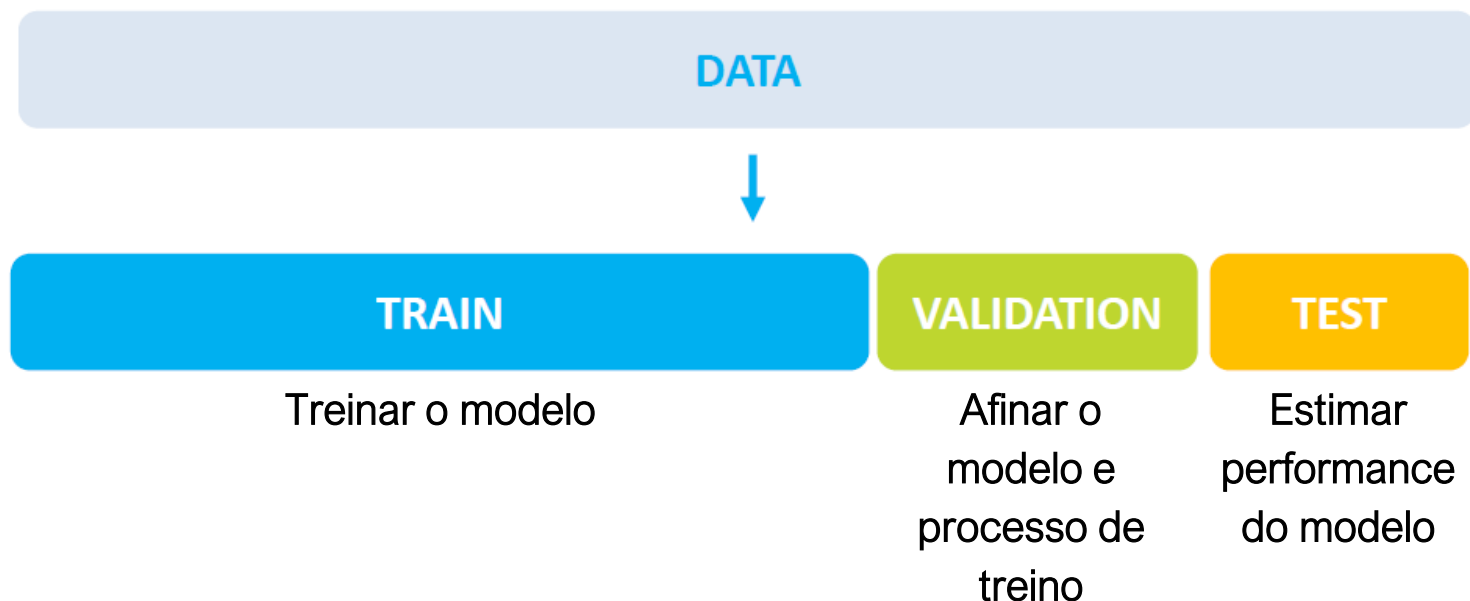
Fit apropriado



Overfitting
(solução “cola” demasiado aos dados de treino)

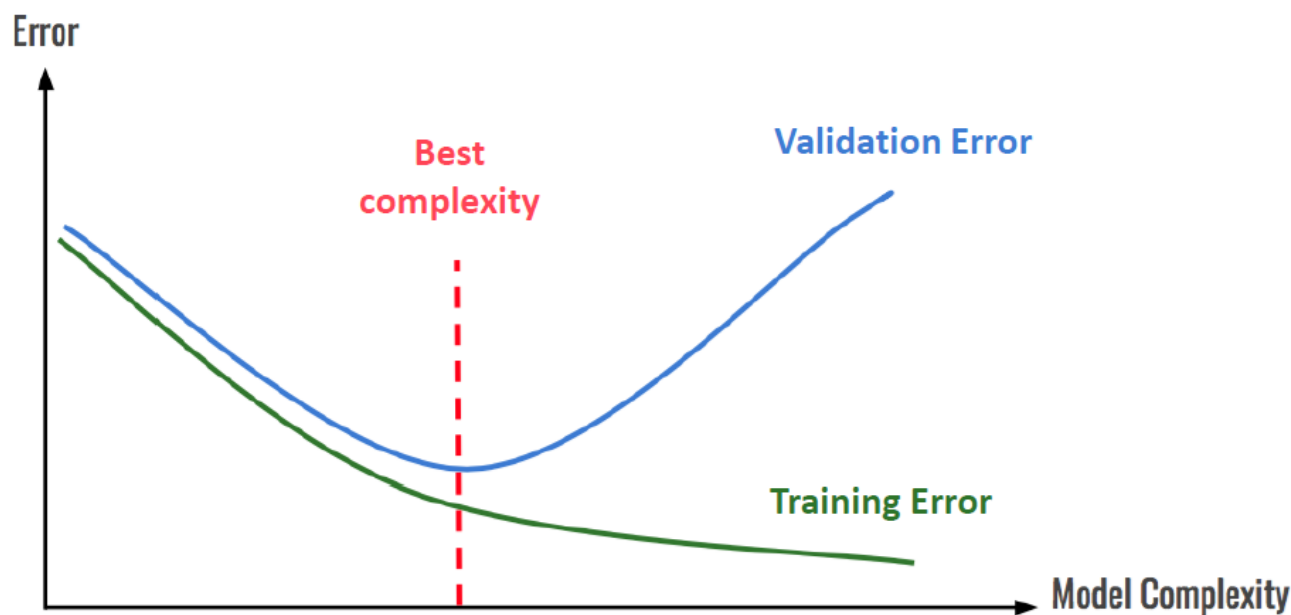
2. Modelos – Overfitting

Como detetar e evitar overfitting? Manter dados de lado além dos dados de treino!



2. Modelos – Overfitting

Como detetar e evitar overfitting? Manter dados de lado além dos dados de treino!



2. Modelos – Overfitting

Como dividir os dados?

Train set

Quanto maior, melhor o classificador/regressor.

Validation set

Quanto maior, melhor a estimação para o processo de treino ótimo.

Test set

Quanto maior, melhor é a estimativa do desempenho do classificador/regressor em dados não vistos.

3. Avaliação de modelos

3. Avaliação de modelos

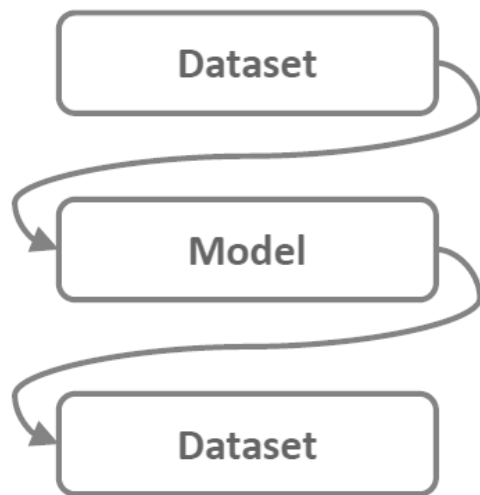
A **avaliação do desempenho** de um modelo preditivo é feita **com base na utilização de um conjunto de dados de validação e/ou de teste**.

É fundamental medir o quão bem o modelo desempenha uma determinada tarefa.

3. Avaliação de modelos

A **avaliação do desempenho** de um modelo preditivo é feita **com base na utilização de um conjunto de dados de validação e/ou de teste**.

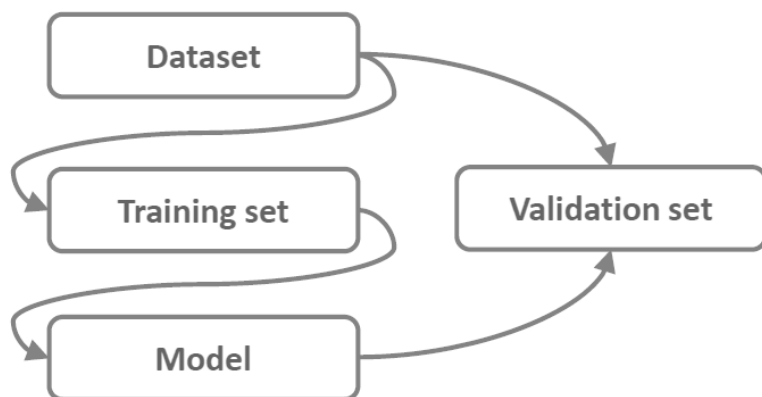
É fundamental medir o quão bem o modelo desempenha uma determinada tarefa.



- Podemos estimar e avaliar o desempenho do modelo nos dados de treino.
- No entanto, as **estimativas baseadas apenas nos dados de treino não são bons indicadores do desempenho do modelo em dados não vistos**.
- Os novos dados provavelmente não serão exatamente iguais aos dados de treino, pelo que o **modelo sofrerá de overfitting**.

3. Avaliação de modelos

Abordagem #1: o método Hold-Out



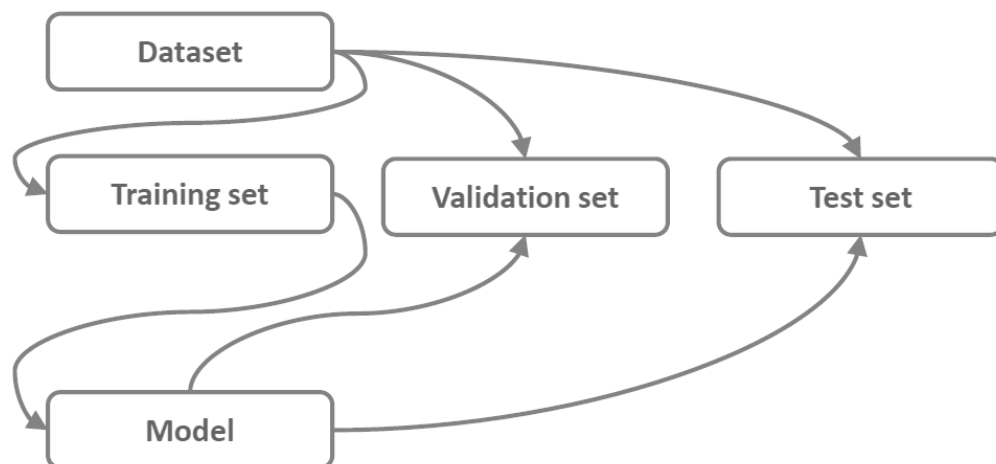
- Este método consiste em **usar um conjunto de dados de validação**, independente dos dados de treino.
- Devemos usar quando existem dados suficientes.

Regra geral:

- 70% para treino
- 30% para validação

3. Avaliação de modelos

Abordagem #1: o método Hold-Out



- Quando existem bastantes dados, podemos dividir em dados de treino, validação e teste.

Regra geral:

- 70% para treino
- 15% para validação
- 15% para teste

3. Avaliação de modelos

Abordagem #1: o método Hold-Out

Q: E se os dados forem desequilibrados relativamente às classes da variável alvo (se categórica)?

A: Certas classes podem não estar (bem) representadas. Precisamos de garantir que a partição dos dados não alterará as proporções originais de cada classe nos novos conjuntos de dados.

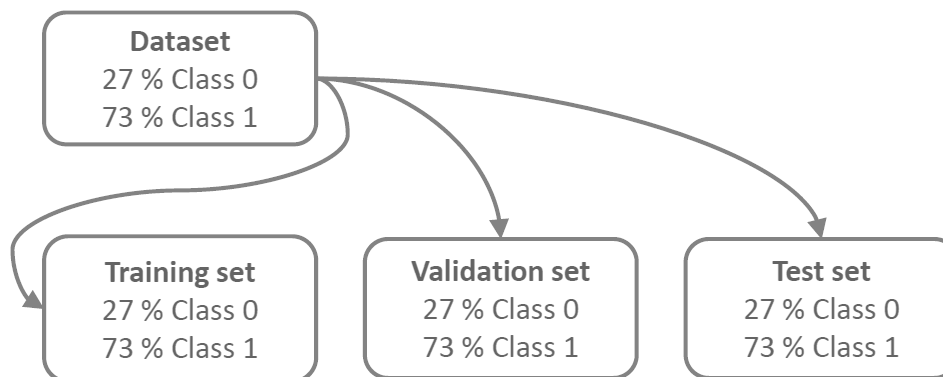
3. Avaliação de modelos

Abordagem #1: o método Hold-Out

Q: E se os dados forem desequilibrados relativamente às classes da variável alvo (se categórica)?

A: Certas classes podem não estar (bem) representadas. Precisamos de garantir que a partição dos dados não alterará as proporções originais de cada classe nos novos conjuntos de dados.

Stratified Sampling: Amostrar cada classe de forma independente, de modo a que cada conjunto de dados tenha o mesmo rácio de uma dada classe.



3. Avaliação de modelos

Abordagem #1: o método Hold-Out

Limitações do método Hold-Out:

1. Resulta apenas numa única estimacção.
2. Requer um conjunto de dados suficientemente grande.

Solução: Usar um repeated Hold-Out

- Executar o método base várias vezes. Cada execução terá um conjunto diferente de treino, validação e teste, conduzindo a resultados diferentes.
- Assim, pode-se utilizar a média dos diferentes resultados, obtendo um resultado mais robusto.
- Mesmo assim, não é o ideal: os diferentes conjuntos de teste sobrepõem-se. Como podemos evitar a sobreposição? *Dica: é a mesma solução para a segunda limitação.*

3. Avaliação de modelos

Abordagem #1: o método Hold-Out

Limitações do método Hold-Out:

1. Resulta apenas numa única estimação.
2. Requer um conjunto de dados suficientemente grande.

Solução: Usar K-Fold cross validation!

3. Avaliação de modelos

Abordagem #2: K-fold Cross Validation

	DATA				
1 st Iteration	TEST	TRAIN	TRAIN	TRAIN	TRAIN
2 nd Iteration	TRAIN	TEST	TRAIN	TRAIN	TRAIN
3 rd Iteration	TRAIN	TRAIN	TEST	TRAIN	TRAIN
4 th Iteration	TRAIN	TRAIN	TRAIN	TEST	TRAIN
5 th Iteration	TRAIN	TRAIN	TRAIN	TRAIN	TEST

Passo #1:

Dividir os dados em k conjuntos de igual dimensão.

Passo #2:

Em cada iteração, usar um subconjunto para teste e o restante para treino.

Passo #3:

Calcula a média das estimativas de cada iteração para obter um valor final.

Nota: De forma a garantir um conjunto de dados de validação, podemos começar por separar o conjunto de dados em treino e teste, e só depois correr o K-Fold CV nos dados de treino, separando em treino e validação de acordo com a figura acima.

3. Avaliação de modelos

Abordagem #3: Leave-One-Out Cross Validation

- É uma configuração do K-Fold cross validation, em que o k é definido como o número de observações no conjunto de dados.
- A LOOCV é uma versão extrema da validação cruzada k-fold que tem o custo computacional máximo. Requer a criação e avaliação de um modelo para cada observação no conjunto de dados.
- Deve ser usada apenas quando temos um conjunto de dados muito reduzido.

3. Avaliação de modelos

Como comparar diferentes modelos?

Passo #1: Escolher a(s) métrica(s).

Passo #2: Escolher um método de avaliação.

Passo #3: Correr o processo de treino e avaliação para cada modelo.

Passo #4: Comparar a(s) métrica(s) entre modelos.

Passo #5: Escolher o modelo com a melhor performance.

3. Avaliação de modelos

Como comparar diferentes modelos?

Passo #1: Escolher a(s) métrica(s).

Passo #2: Escolher um método de avaliação.

Passo #3: Correr o processo de treino e avaliação para cada modelo.

Passo #4: Comparar a(s) métrica(s) entre modelos.

Passo #5: Escolher o modelo com a melhor performance.

└─ Q: Como ter a certeza que um determinado modelo é melhor que outro para todos os casos, sendo que a sua performance está dependente do conjunto de dados usado para treino e teste?

R: Usando diferentes conjuntos de dados para treino e teste (K-Fold CV), e tirar a média e desvio padrão dos resultados!

4. Métricas de avaliação

4. Métricas de avaliação - Classificação

Tudo começa com...

		Classe Verdadeira	
		Positivo	Negativo
Classe Estimada	Positivo	TP	FP
	Negativo	FN	TN

Erro Tipo I

Erro Tipo II

Em problemas de classificação, as estimações são corretas ou erradas. Assim, os resultados da estimação podem ser representados numa matriz.

4. Métricas de avaliação - Classificação



Classe Verdadeira: Goat

Classe Estimada: Goat

TRUE POSITIVE

		Classe Verdadeira	
		Goat	No Goat
Classe Estimada	Goat	1	
	No Goat		

4. Métricas de avaliação - Classificação



Classe Verdadeira: No Goat

Classe Estimada: Goat

FALSE POSITIVE

		Classe Verdadeira	
		Goat	No Goat
Classe Estimada	Goat	1	1
	No Goat		

4. Métricas de avaliação - Classificação



Classe Verdadeira: Goat

Classe Estimada: No Goat

FALSE NEGATIVE

		Classe Verdadeira	
		Goat	No Goat
Classe Estimada	Goat	1	1
	No Goat	1	

4. Métricas de avaliação - Classificação



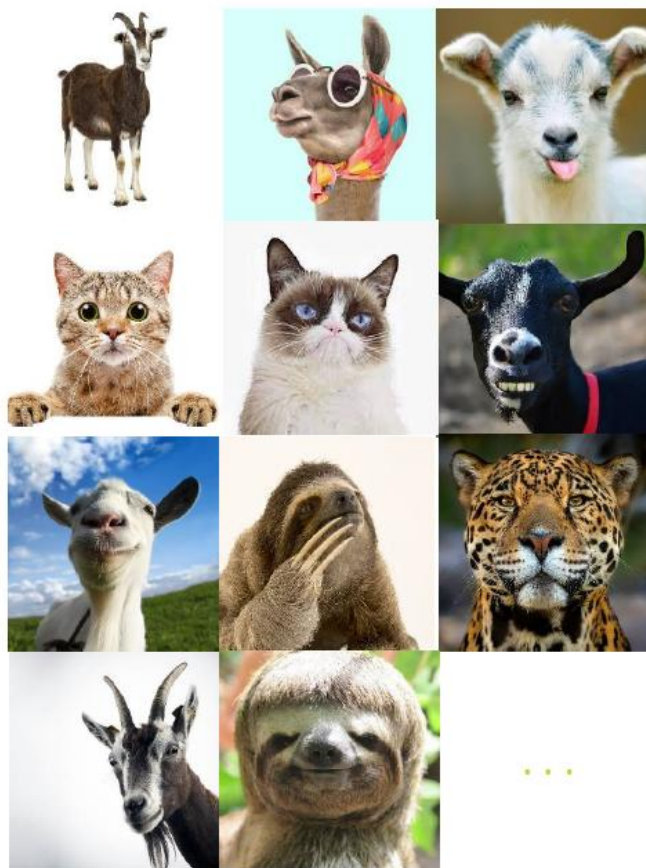
Classe Verdadeira: No Goat

Classe Estimada: No Goat

TRUE NEGATIVE

		Classe Verdadeira	
		Goat	No Goat
Classe Estimada	Goat	1	1
	No Goat	1	1

4. Métricas de avaliação - Classificação



		Classe Verdadeira	
		Goat	No Goat
Classe Estimada	Goat	30	4
	No Goat	6	20

4. Métricas de avaliação - Classificação

Métrica #1: Accuracy

		Classe Verdadeira	
		Positivo	Negativo
Classe Estimada	Positivo	TP	FP
	Negativo	FN	TN

		Classe Verdadeira	
		Goat	No Goat
Classe Estimada	Goat	30	4
	No Goat	6	20

Accuracy: Proporção de observações corretamente estimadas (positivas ou negativas).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

No nosso exemplo:

$$Accuracy = \frac{20 + 30}{20 + 30 + 6 + 4} = 0.83$$

↳ Parece um bom resultado, mas veremos que pode ser enganador (slide 50)

4. Métricas de avaliação - Classificação

Métrica #2: Error Rate

		Classe Verdadeira	
		Positivo	Negativo
Classe Estimada	Positivo	TP	FP
	Negativo	FN	TN

		Classe Verdadeira	
		Goat	No Goat
Classe Estimada	Goat	30	4
	No Goat	6	20

Error Rate: Proporção de observações incorretamente estimadas (positivas ou negativas).

$$Error\ Rate = \frac{FP + FN}{TP + TN + FP + FN}$$

No nosso exemplo:

$$Error\ Rate = \frac{4 + 6}{20 + 30 + 6 + 4} = 0.167$$

↳ Quanto mais baixo melhor!

4. Métricas de avaliação - Classificação

Métrica #3: Precision

		Classe Verdadeira	
		Positivo	Negativo
Classe Estimada	Positivo	TP	FP
	Negativo	FN	TN

Precision: Proporção de observações corretamente estimadas como positivas, de todas as observações estimadas como positivas.

$$Precision = \frac{TP}{TP + FP}$$

		Classe Verdadeira	
		Goat	No Goat
Classe Estimada	Goat	30	4
	No Goat	6	20

No nosso exemplo:

$$Precision = \frac{30}{30 + 4} = 0.882$$

Faz sentido usar quando queremos penalizar mais um Falso Positivo (ex: detecção de emails de spam)!

4. Métricas de avaliação - Classificação

Métrica #4: Recall / Sensitivity / True Positive Rate (TPR)

		Classe Verdadeira	
		Positivo	Negativo
Classe Estimada	Positivo	TP	FP
	Negativo	FN	TN

Recall: Proporção de observações corretamente estimadas como positivas, de todas as observações positivas.

$$Recall = \frac{TP}{TP + FN}$$

		Classe Verdadeira	
		Goat	No Goat
Classe Estimada	Goat	30	4
	No Goat	6	20

No nosso exemplo:

$$Recall = \frac{30}{30 + 6} = 0.833$$

Faz sentido usar quando queremos penalizar mais um Falso Negativo (ex: detecção de pacientes doentes)!

4. Métricas de avaliação - Classificação

Métrica #5: Specificity / True Negative Rate (TNR)

		Classe Verdadeira	
		Positivo	Negativo
Classe Estimada	Positivo	TP	FP
	Negativo	FN	TN

Specificity: Proporção de observações corretamente estimadas como negativas, de todas as observações negativas.

$$Recall = \frac{TN}{FP + TN}$$

		Classe Verdadeira	
		Goat	No Goat
Classe Estimada	Goat	30	4
	No Goat	6	20

No nosso exemplo:

$$Recall = \frac{20}{4 + 20} = 0.833$$

4. Métricas de avaliação - Classificação

O problema de dados desequilibrados (imbalanced data)

		Classe Verdadeira	
		Positivo	Negativo
Classe Estimada	Positivo	TP	FP
	Negativo	FN	TN

		Classe Verdadeira	
		Goat	No Goat
Classe Estimada	Goat	1	0
	No Goat	9	9990

Neste exemplo alternativo:

$$Accuracy = \frac{1 + 9990}{1 + 0 + 9 + 9990} = 0.9991$$

Este parece ser um modelo quase perfeito!

Mas, na realidade, o modelo não é capaz de identificar bem a classe Goat, o que é o nosso principal objetivo!

O Recall é igual a 0,1!

Este é o problema dos dados desequilibrados...

4. Métricas de avaliação - Classificação

Para ter em conta dados desequilibrados (imbalanced data)

		Classe Verdadeira	
		Positivo	Negativo
Classe Estimada	Positivo	TP	FP
	Negativo	FN	TN

		Classe Verdadeira	
		Goat	No Goat
Classe Estimada	Goat	1	0
	No Goat	9	9990

$$\text{Balanced Accuracy} = \frac{\text{Sensitivity} + \text{Specificity}}{2} = \frac{\frac{TP}{TP+FN} + \frac{TN}{FP+TN}}{2} = \frac{\frac{1}{10} + \frac{9990}{9990}}{2} = 0.55$$

$$F1 \text{ Score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} = 2 \times \frac{1 \times 0.1}{1 + 0.1} = 0.18$$

4. Métricas de avaliação - Classificação

Valor de corte (cutoff) para classificação

A maioria dos algoritmos de previsão classifica através de um processo de 2 etapas. Para cada observação:

- Calcular a probabilidade de pertencer à classe “True”
- Comparar com o valor de corte e classificar em conformidade

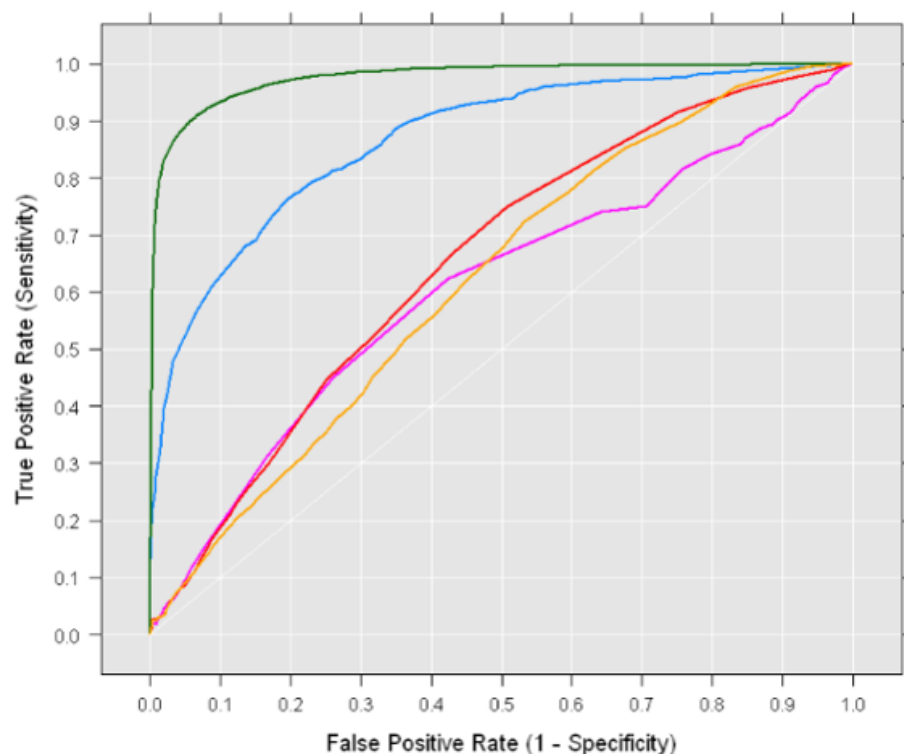
O valor de corte predefinido é 0,5:

- Se probabilidade estimada $\geq 0,5$, classificar como “1”
- Se probabilidade estimada $< 0,5$, classificar como “0”

Podemos testar utilizar diferentes valores de corte (normalmente, a taxa de erro é mais baixa para o corte = 0,5).

4. Métricas de avaliação - Classificação

A curva ROC para determinar o melhor modelo

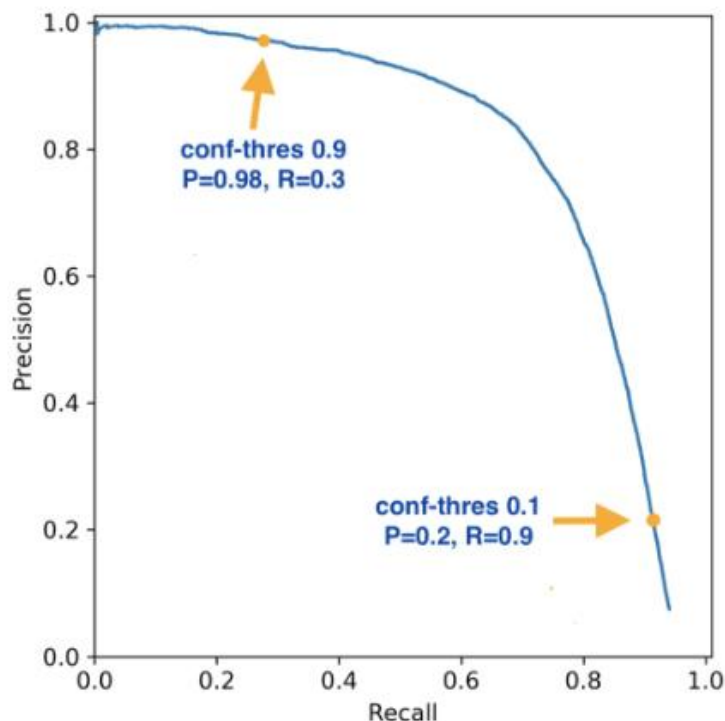


A curva ROC simula o TPR e o FPR para diferentes thresholds, permitindo:

- Comparar as curvas para diferentes modelos: a curva ROC com maior área abaixo da curva, corresponde ao melhor modelo.

4. Métricas de avaliação - Classificação

A curva Precision-Recall para determinar o valor de corte (cutoff) ótimo



A curva Precision-Recall simula as 2 métricas para diferentes thresholds, permitindo:

- Identificar o valor de corte que produz os melhor resultados, estabelecendo o threshold no ponto em que o F1-Score é maior.

4. Métricas de avaliação - Regressão

- Quando a variável alvo é numérica, deixamos de poder representar os resultados do modelo preditivo numa matriz.
- Assim, necessitamos de **avaliar o modelo medindo o quão longe a estimacão (\hat{y}) está do valor verdadeiro (y)**.

Modelo 1		
y	\hat{y}	$ y - \hat{y} $
3	4	1
6	5	1
4	6	2
12	11	1
5	5	0

Modelo 2		
y	\hat{y}	$ y - \hat{y} $
3	4	1
6	5	1
4	6	2
8	7	1
5	14	9

4. Métricas de avaliação - Regressão

Métrica #1: Mean Absolute Error (MAE)

Modelo 1

y	\hat{y}	$ y - \hat{y} $
3	4	1
6	5	1
4	6	2
12	11	1
5	5	0

Modelo 2

y	\hat{y}	$ y - \hat{y} $
3	4	1
6	5	1
4	6	2
8	7	1
5	14	9

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

Modelo 1:

$$MAE = \frac{1 + 1 + 2 + 1 + 0}{5} = 1$$

Modelo 2:

$$MAE = \frac{1 + 1 + 2 + 1 + 9}{5} = 2.5$$

A MAE mede a distância média absoluta entre os dados reais e os dados estimados, mas **não consegue punir os erros de maiores magnitude.**

4. Métricas de avaliação - Regressão

Métrica #2: Mean Squared Error (MSE)

Modelo 1

y	\hat{y}	$(y - \hat{y})^2$
3	4	1
6	5	1
4	6	4
12	11	1
5	5	0

Modelo 2

y	\hat{y}	$(y - \hat{y})^2$
3	4	1
6	5	1
4	6	4
8	7	1
5	14	81

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

Modelo 1:

$$MSE = \frac{1 + 1 + 4 + 1 + 0}{5} = 1.33$$

Modelo 2:

$$MSE = \frac{1 + 1 + 4 + 1 + 81}{5} = 14.83$$

A MSE mede a distância média ao quadrado entre os dados reais e os dados estimados.

Aqui, **os erros maiores influenciam mais negativamente a métrica.**

Nota: precisamos tirar a raiz quadrada para trazer a métrica de volta à unidade da variável alvo.

4. Métricas de avaliação - Regressão

Métrica #3: R-Squared (R^2)

Modelo 1

y	\hat{y}	$(y - \hat{y})^2$	$(y - \bar{y})^2$
3	4	1	9
6	5	1	0
4	6	4	4
12	11	1	36
5	5	0	1

Modelo 2

y	\hat{y}	$(y - \hat{y})^2$	$(y - \bar{y})^2$
3	4	1	9
6	5	1	0
4	6	4	4
8	7	1	36
5	14	81	1

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

Modelo 1:

$$R^2 = 1 - \frac{7}{50} = 0.86$$

Modelo 2:

$$R^2 = 1 - \frac{88}{50} = -0.76$$

O R^2 mede a **proporção da variação na variável alvo que é explicada a partir do modelo** (ou por outras palavras, a partir das variáveis independentes).

Se negativo, significa que é pior usar o modelo para prever do que usar simplesmente a média da variável.

4. Métricas de avaliação - Regressão

Métrica #4: R-Squared ajustado (*Adjusted R²*)

Para comparar modelos:

- À medida que o **número de variáveis independentes aumenta**, o valor de ***R²* nunca diminuirá**.
- Motivo: qualquer variável independente tem tendência para se correlacionar e explicar ligeiramente a variável alvo.
- Assim, **para comparar modelos com diferentes números de variáveis independentes**, podemos **utilizar o *R²* ajustado**:

$$\text{Adjusted } R^2 = 1 - \frac{(1 - R^2)(n - 1)}{n - p - 1}$$

Onde *n* é o número de observações, e *p* o número de variáveis independentes usado no modelo.

Obrigado!