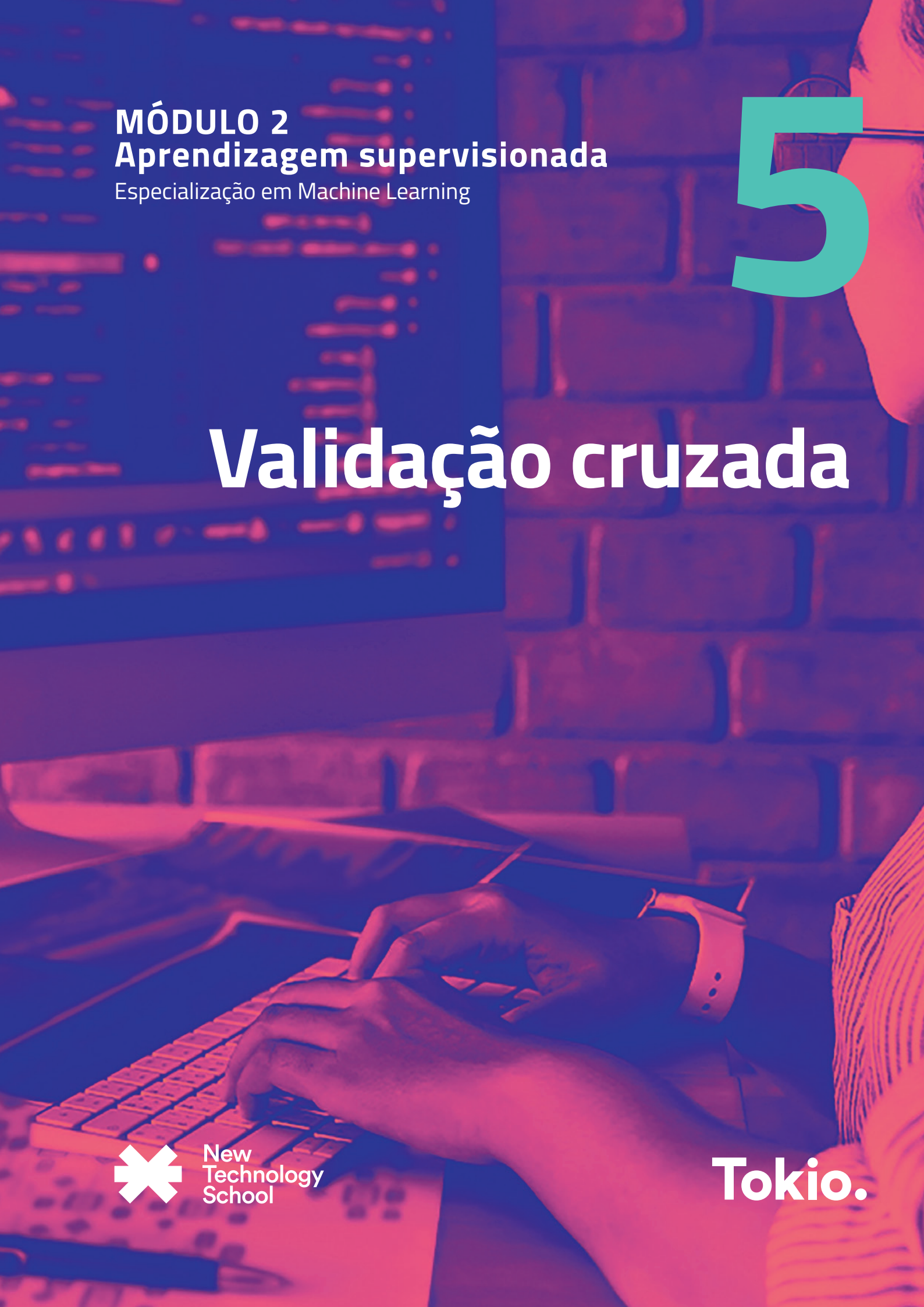
MÓDULO 2
Aprendizagem supervisionada
Especialização em Machine Learning

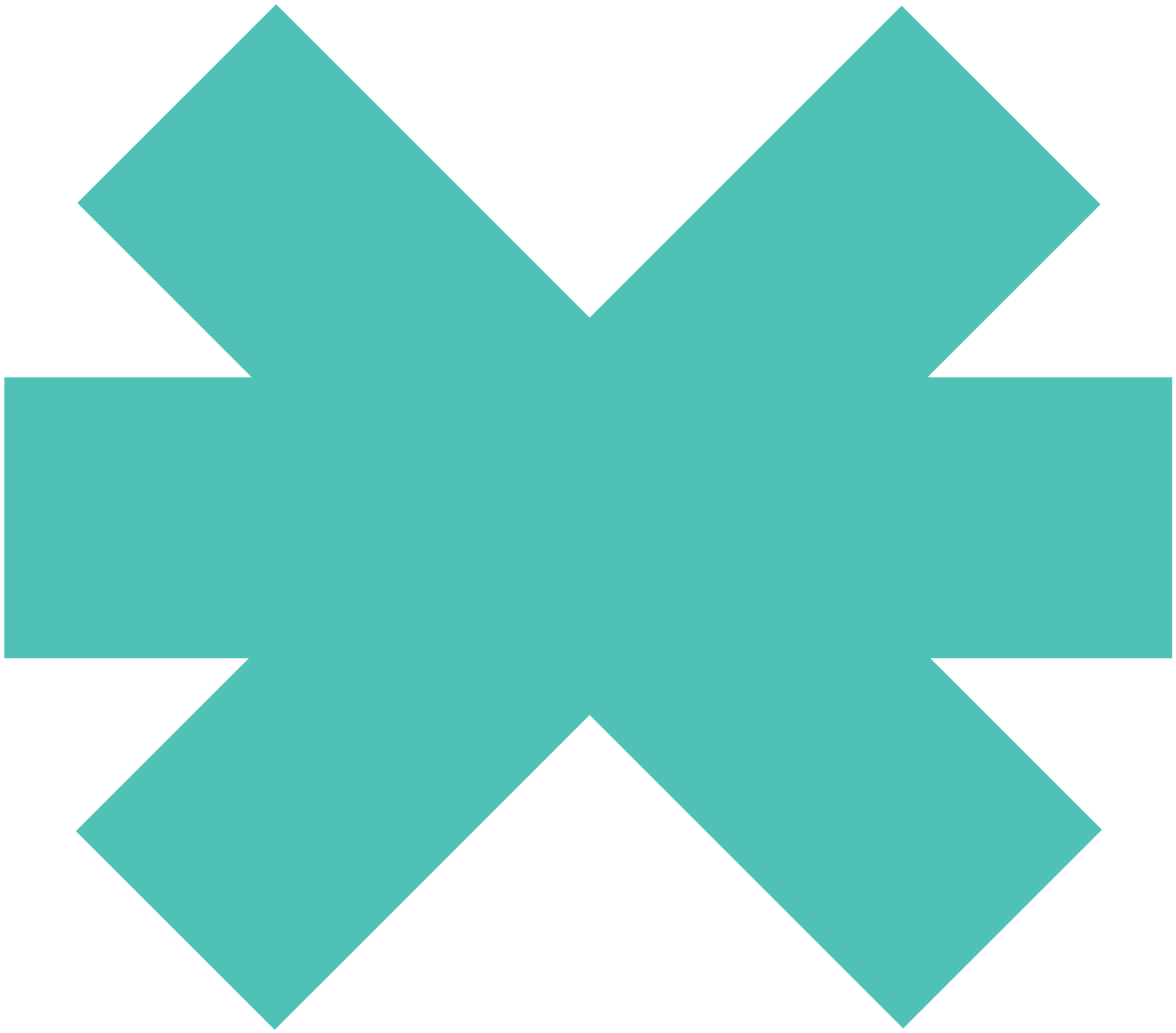
5

Validação cruzada



New Technology School

Tokio.



5 Validação cruzada

Sumário

A validação cruzada permite-nos avaliar e validar o nosso modelo, antes de o usar com novos exemplos, escolhendo os hiperparâmetros ótimos para isso.

Logo, ajudar-nos-á a encontrar o valor de λ idóneo, para cada modelo.

Neste sentido, realizaremos a validação cruzada através de uma avaliação prévia do modelo, sobre um subconjunto de dados. Para isso, reordenamos aleatoriamente o conjunto de dados (mantendo as relações entre os vetores X e Y) e dividimo-lo em três subconjuntos:

- O **subconjunto de treino**, utilizado para treinar o modelo (obter a Θ ótima).
- O **subconjunto de validação,** destinado a regularizar o modelo (obter a λ ótima).
- O **subconjunto de teste,** dirigido a avaliar definitivamente o modelo, a obter a sua precisão final.

5.1 Métodos de resolução

Para otimizar o valor de lambda, pegamos nos resultados do processo de treino, sobre o subconjunto de treino (o valor ótimo de Θ), e calculamos, no subconjunto de validação, para cada valor de λ que se vá estimar, o custo total.

Deste modo, selecionamos o valor de lambda que minimize o custo total como o fator a utilizar para o nosso modelo.

Finalmente, pegamos em ambos os fatores (Θ e λ), como integrantes do nosso modelo, e realizamos predições sobre o subconjunto de teste, para encontrar o seu custo total.

5.2 Subdivisão do conjunto de dados

É muito importante que cada fase se realize única e exclusivamente sobre o seu subconjunto de dados atribuído, isto é, que quando realizemos a validação não a efetuemos sobre dados que já tenhamos usado para treinar o modelo ou que, quando realizemos o teste final, não o façamos sobre dados que já tenhamos usado para treinar ou validar o modelo.

Face a subdividir o conjunto de dados, devemos selecionar algum destes rácios sugeridos, em função do número de exemplos disponíveis no total:

- 60%, 20% e 20%.
- 70%, 15% e 15%.
- 50%, 25% e 25%.

É importante que todos os subconjuntos contem com um mínimo de dados disponíveis, uma vez que, se fossem demasiado escassos, não poderíamos realizar a validação cruzada ou o teste final corretamente.

No caso de existirem poucos dados, nos dois subconjuntos mais pequenos, podemos usar rácios mais elevados ou outros métodos como o *k-fold*.

5.3 *K-fold*

O método *k-fold* permite-nos dispor de um número *k* de "dobras" ou subconjuntos com reposição dos dados, para poder avaliar múltiplos valores de λ.

Neste método estabelecemos *k* partições de dados, treinamos o modelo sobre *k-1* das partições e validamo-lo sobre a partição restante, podendo validar um número *k* de valores de λ diferentes.

Desta forma, nunca validamos λ sobre os mesmos dados que temos usado para treinar o modelo e examinamos os seus diferentes valores usando um subconjunto de dados distinto, ainda que partilhem a maioria dos exemplos.

Podemos escolher este método quando não dispusermos de um número de exemplos inicial, que nos permita simplesmente dividir o conjunto de dados em vários subconjuntos independentes com segurança.

	X	Y
Grupo 1	1	12,95
	2	16,09
	3	19,23
	4	22,37
Grupo 2	5	25,51
	6	28,66
	7	31,80
	8	34,94
Grupo 3	9	38,08
	10	41,22
	11	44,36
	12	47,51
Grupo 4	13	50,65
	14	53,79
	15	56,93
	16	60,07
Grupo 5	17	63,21
	18	66,36
	19	69,50
	20	72,64

Iteração	X_*train*, Y_*train*	X_cv, Y_cv
1	Grupos 1, 2, 3 e 4	Grupo 5
2	Grupos 2, 3, 4 e 5	Grupo 1
3	Grupos 3, 4, 5 e 1	Grupo 2
4	Grupos 4, 5, 1 e 2	Grupo 3
5	Grupos 5, 1, 2 e 3	Grupo 4

5.4 Algoritmo de treino atualizado

O algoritmo de treino completo de um problema, de regressão linear múltipla, atualizado com normalização, seria:

1. Compilar os exemplos *X* e os seus resultados previamente conhecidos *Y*.
2. Normalizar os exemplos *X*.
3. Dividir o conjunto de dados em subconjuntos de treino, validação e teste.
4. Escolher um rácio de aprendizagem α.
5. Inicializar os pesos Θ, de forma aleatória.
6. Iterativamente, calcular o custo, assim como suas derivadas/inclinações e atualizar Θ sobre o subconjunto de treino.
7. Finalizar quando Θ convirja num valor ótimo.
8. Modificar o rácio de aprendizagem α se necessário.
9. Obter a Θ ótima.
10. Otimizar λ iterativamente, sobre o subconjunto de validação.
11. Obter a λ ótima.
12. Avaliar o modelo sobre o subconjunto de teste e obter o custo total final.
13. Usar Θ e λ para formar o nosso modelo e realizar predições.