<table>
<tr><td>Título dos encontros</td><td colspan="3">Aprendizado de Máquina Supervisionado</td><td>Sequência:
07</td></tr>
<tr><td>Etapa/Ano</td><td colspan="4">Ensino Médio: ☐ 1º Ano | ☑ 2º Ano | ☐ 3º Ano</td></tr>
<tr><td>Nível de Maturidade
[link]</td><td>Escola</td><td>☐ Emergente
☐ Básico
☑ Intermediário
☐ Avançado</td><td>Docente</td><td>☐ Básico
☑ Intermediário
☐ Avançado</td></tr>
<tr><td>Competências</td><td colspan="4">☐ C1 ☑ C2 ☑ C3 ☐ C4 ☐ C5</td></tr>
<tr><td>Objetos de Conhecimento</td><td colspan="4">• Aprendizado de máquina
• uso e design
• técnicas e aplicações</td></tr>
<tr><td>Recursos Educacionais</td><td colspan="4">• Quadro branco ou projetor
• Marcadores ou canetas
• Celular, tablet ou Computadores com acesso à Internet (preferencialmente individuais)
• Acesso a um navegador de internet</td></tr>
<tr><td>Componentes Curriculares Relacionados</td><td>☐ Arte
☐ Língua Port.
☐ Ed. Física</td><td>☐ Geografia
☐ História
☐ Matemática</td><td colspan="2">☐ Língua Inglesa
☐ Ciências
☑ Computação</td></tr>
<tr><td>Palavras-chave</td><td colspan="4">Aprendizado de Máquina; Inteligência Artificial</td></tr>
<tr><td>Perguntas Importantes</td><td colspan="4">Como os dados podem ser ou são utilizados para treinar modelos?
Que tratamentos e cuidados devemos ter na preparação de dados para treinar modelos?</td></tr>
<tr><td>Encontros</td><td>1</td><td colspan="2">Total de Horas-aula</td><td>2</td></tr>
<tr><td>Objetivos</td><td colspan="4">• Entender o funcionamento de técnicas de Aprendizado Supervisionado e seu papel na IA</td></tr>
<tr><td>Habilidades Relacionadas</td><td colspan="4">BNCC:

Referencial Curricular IA no EM:

EFIA6911 Descrever e experimentar uma Inteligência Artificial que aprenda com exemplos (supervisionado)</td></tr>
<tr><td>Práticas Pedagógicas Inovadoras</td><td colspan="4">☑ Aula Enriquecida com Tecnologia
☐ Ensino Híbrido: Sala de aula invertida
☐ Ensino Híbrido: rotação por estação
☐ Ensino Híbrido: rotação individual</td></tr>
</table>

CC BY-NC-SA 4.0 VICARI, Rosa Maria; GALAFASSI, Cristiano; BRACKMANN, Christian; OLIVEIRA, Flávio.
IA@Escola. 2025. Disponível em: www.computacional.com.br/ia.

	☑ Aulas Mão na massa ☐ Aprendizagem baseada em projetos ☐ IA Desconectada

Sequência Didática - Encontro 6

1. Introdução ao Tema

Agora temos uma compreensão de como os dados podem ser trabalhados para nos ajudar a gerar **modelos de IA** que respondem questões úteis e complexas. Também vimos as Redes Neurais, o que nos trouxe os conceitos do que são modelos e de que modelos precisam ser **treinados.** Só que as redes neurais, como vimos, são um modelo bem poderoso e que, dependendo do caso, pode exigir mais do nosso hardware, ou demorar bastante para treinar. Ocorre que, dependendo do problema, existem outros algoritmos e modelos que podem ser usados, que não exigem tantos recursos, e cujos resultados atendem muito bem nossas necessidades! Vamos, então, explorar esses algoritmos, começando pelas técnicas de ***aprendizado supervisionado*.**

2. Aprendizado de Máquina Supervisionado – que é

As técnicas de aprendizado de máquina supervisionado geram modelos capazes de realizar inferências preditivas ou prescritivas a partir de um conjunto de dados ou *dataset.* Esses modelos são treinados com dados **rotulados**, ou seja, dados que já possuem uma resposta conhecida, permitindo que o modelo aprenda a fazer previsões sobre novos dados. É o uso de dados rotulados no treinamento que caracteriza o aprendizado supervisionado. É bom lembrar que, antes de usá-los no treinamento, os dados passam por aquelas etapas que vimos ao estudar a Ciência de Dados (limpeza, seleção de atributos, transformações etc.).

Vamos considerar o conjunto de dados abaixo, que é uma amostra do *dataset* disponível em https://dadosabertos.poa.br/dataset/acidentes-de-transito-acidentes. Os dados mostram diversas informações de acidentes de trânsito, como número de veículos envolvidos e número de feridos ou mortos nestes acidentes:

data	feridos	fatais	auto	taxi	lotacao	onibus_urb	moto	tipo_acid	dia_sem	noite_dia
17/10/2020 00:00	1	0	3	0	0	0	1	ABALROAMENTO	SÁBADO	NOITE
01/01/2020 00:00	1	0	0	1	0	0	1	ABALROAMENTO	QUARTA-FEIRA	NOITE
01/01/2020 00:00	1	0	1	0	0	0	0	ATROPELAMENTO	QUARTA-FEIRA	NOITE
02/01/2020 00:00	2	0	0	0	0	0	1	ATROPELAMENTO	QUINTA-FEIRA	NOITE
02/01/2020 00:00	1	0	1	0	0	0	1	ABALROAMENTO	QUINTA-FEIRA	DIA
02/01/2020 00:00	1	0	1	0	0	0	1	ABALROAMENTO	QUINTA-FEIRA	DIA
02/01/2020 00:00	1	0	2	0	0	0	1	COLISÃO	QUINTA-FEIRA	DIA

CC BY-NC-SA 4.0

02/01/2020 00:00	7	0	2	0	0	0	0	ABALROAMENTO	QUINTA-FEIRA	DIA
02/01/2020 00:00	0	0	2	0	0	0	0	ABALROAMENTO	QUINTA-FEIRA	NOITE
02/01/2020 00:00	0	0	2	0	0	0	0	COLISÃO	QUINTA-FEIRA	DIA
02/01/2020 00:00	0	0	2	0	0	0	0	COLISÃO	QUINTA-FEIRA	DIA
02/01/2020 00:00	0	0	1	0	0	1	0	ABALROAMENTO	QUINTA-FEIRA	DIA
02/01/2020 00:00	0	0	2	0	0	0	0	CHOQUE	QUINTA-FEIRA	DIA
02/01/2020 00:00	0	0	2	0	0	0	0	CHOQUE	QUINTA-FEIRA	DIA
02/01/2020 00:00	0	0	2	0	0	0	0	ABALROAMENTO	QUINTA-FEIRA	DIA
03/01/2020 00:00	0	0	2	0	0	0	0	ABALROAMENTO	SEXTA-FEIRA	DIA
02/01/2020 00:00	0	0	1	0	0	1	0	CHOQUE	QUINTA-FEIRA	DIA
02/01/2020 00:00	1	0	1	0	0	0	1	ABALROAMENTO	QUINTA-FEIRA	NOITE
03/01/2020 00:00	1	0	1	0	0	0	1	ABALROAMENTO	SEXTA-FEIRA	DIA
03/01/2020 00:00	2	0	1	0	0	0	1	ABALROAMENTO	SEXTA-FEIRA	DIA

Os dados acima são apenas uma amostra; o *dataset* original tem mais atributos (colunas) e um total de 68661 linhas, de 2019 a 2025. Será que podemos treinar um modelo capaz de, dado um registro de acidente, prever com boa taxa de acertos o número de vítimas fatais (que chamaremos de nosso **atributo-alvo**), por exemplo, a partir dos valores de outros atributos? Note que, no *dataset*, esta informação já veio preenchida com o número correto; é isso o que torna este um caso de aprendizado supervisionado. Claro que nosso objetivo é, com esses dados, conseguir prever esse atributo para acidentes que ainda não aconteceram. Isso pode ter diversas aplicações; por exemplo, se conseguirmos prever o que mais impacta no número de vítimas (se o tipo de acidente ou os veículos envolvidos, ou o dia da semana etc.), o governo pode fazer campanhas mais direcionadas, ou se organizar melhor. Vamos, ver, então, como podemos treinar esse modelo.

Conjuntos de treino e teste

O primeiro passo, depois de aplicarmos as técnicas que vimos em Ciência de Dados, é dividir nosso *dataset* em:

1. **Conjunto de Treino**: dados utilizados para treinar o modelo.
2. **Conjunto de Teste**: dados utilizados para avaliar o desempenho do modelo, ao longo do treinamento (para ver se o treinamento está progredindo bem).
3. **Conjunto de validação**: dados utilizados para avaliar o desempenho final do modelo.

A ideia é que, para termos uma avaliação realista do modelo, devemos testá-lo com dados que ele ainda não "viu", ou seja, que não foram usados durante o treinamento. Assim, podemos garantir que a chance do modelo acertar as predições – e assim, ter um bom desempenho – porque já conhecia a resposta não ocorre. É bem comum usar a proporção 80/10/10: 80% dos dados para treino, 10% para teste e 10% para validação. Às vezes, o conjunto de validação é separado no início e aplicamos 80/20 sobre o conjunto restante. Em nosso exemplo, considerando

CC BY-NC-SA 4.0 VICARI, Rosa Maria; GALAFASSI, Cristiano; BRACKMANN, Christian; OLIVEIRA, Flávio.
IA@Escola. 2025. Disponível em: www.computacional.com.br/ia.

apenas treino e teste, temos 20 linhas: se aplicarmos essa proporção, teremos 16 linhas para treino e 4 linhas para teste, que devemos escolher ao acaso. Algo como:

Conjunto de treino:

data	feridos	fatais	auto	taxi	lotacao	onibus_urb	moto	tipo_acid	dia_sem	noite_dia
17/10/2020 00:00	1	0	3	0	0	0	1	ABALROAMENTO	SÁBADO	NOITE
02/01/2020 00:00	2	0	0	0	0	0	1	ATROPELAMENTO	QUINTA-FEIRA	NOITE
02/01/2020 00:00	1	0	1	0	0	0	1	ABALROAMENTO	QUINTA-FEIRA	DIA
02/01/2020 00:00	1	0	1	0	0	0	1	ABALROAMENTO	QUINTA-FEIRA	DIA
02/01/2020 00:00	1	0	2	0	0	0	1	COLISÃO	QUINTA-FEIRA	DIA
02/01/2020 00:00	7	0	2	0	0	0	0	ABALROAMENTO	QUINTA-FEIRA	DIA
02/01/2020 00:00	0	0	2	0	0	0	0	ABALROAMENTO	QUINTA-FEIRA	NOITE
02/01/2020 00:00	0	0	2	0	0	0	0	COLISÃO	QUINTA-FEIRA	DIA
02/01/2020 00:00	0	0	1	0	0	1	0	ABALROAMENTO	QUINTA-FEIRA	DIA
02/01/2020 00:00	0	0	2	0	0	0	0	CHOQUE	QUINTA-FEIRA	DIA
02/01/2020 00:00	0	0	2	0	0	0	0	ABALROAMENTO	QUINTA-FEIRA	DIA
03/01/2020 00:00	0	0	2	0	0	0	0	ABALROAMENTO	SEXTA-FEIRA	DIA
02/01/2020 00:00	0	0	1	0	0	1	0	CHOQUE	QUINTA-FEIRA	DIA
02/01/2020 00:00	1	0	1	0	0	0	1	ABALROAMENTO	QUINTA-FEIRA	NOITE
03/01/2020 00:00	1	0	1	0	0	0	1	ABALROAMENTO	SEXTA-FEIRA	DIA
03/01/2020 00:00	2	0	1	0	0	0	1	ABALROAMENTO	SEXTA-FEIRA	DIA

Conjunto de teste:

data	feridos	fatais	auto	taxi	lotacao	onibus_urb	moto	tipo_acid	dia_sem	noite_dia
01/01/2020 00:00	1	0	0	1	0	0	1	ABALROAMENTO	QUARTA-FEIRA	NOITE
01/01/2020 00:00	1	0	1	0	0	0	0	ATROPELAMENTO	QUARTA-FEIRA	NOITE
02/01/2020 00:00	0	0	2	0	0	0	0	COLISÃO	QUINTA-FEIRA	DIA
02/01/2020 00:00	0	0	2	0	0	0	0	CHOQUE	QUINTA-FEIRA	DIA

Você pode achar que é pouco para termos um resultado confiável – e é mesmo! A tabela do exemplo é apenas para fins didáticos. Na amostra original, teríamos 68661*0,8 = 54929 linhas para treino e 13732 linhas para teste. Quanto

CC BY-NC-SA 4.0

maior for o volume de dados, podemos considerar proporções maiores para o treino – por exemplo, 90% para treino e 10% para teste. Quanto mais dados de treino, melhor; só precisamos garantir as seguintes condições:

1. O conjunto de teste deve ter um tamanho mínimo, compatível com nosso universo de dados. Se for muito pequeno, o teste pode não ser confiável.
2. Os dados – tanto de treino quanto de teste – devem estar **balanceados.** Em nosso exemplo, se a cidade possui mais carros do que motos, por exemplo, isso deve se refletir em nossa amostra, para podermos gerar um modelo capaz de fazer predições realistas. Não é o caso de entrarmos em todos os detalhes estatísticos aqui, mas podemos resumir a questão da seguinte forma: quanto mais próximo do universo real for o *dataset* que usarmos na aprendizagem, mais confiável será o modelo de IA treinado.

Ok, agora temos nossos dados. O que fazer com eles? É hora de olharmos para os algoritmos que fazem a mágica: as *técnicas de aprendizagem supervisionada.*

Técnicas de Aprendizagem Supervisionada

Classificação

Os algoritmos de classificação aprendem a responder, dado um exemplo, a qual classe ele pertence, dentro de um conjunto de classes possíveis. Por exemplo, no nosso conjunto de dados, temos a coluna "tipo de acidente", que tem um conjunto de valores possíveis, colisão, atropelamento, choque e abalroamento. Outro exemplo é a classificação de animais, quer vimos ao discutir árvores de decisão. Por sinal, as árvores de decisão constituem um exemplo muito popular de modelos de classificação e podem ser treinadas a partir de *datasets*. valores alvos a partir de dados rotulados. Outro exemplo é a **regressão logística**, muito popular para classificação binária (ou seja, quando temos duas classes possíveis, como dia e noite), que calcula uma combinação linear dos valores dos atributos e aplica a função logística sobre o resultado para gerar uma probabilidade de pertencimento à classe.

A título de curiosidade: treinamos uma árvore de decisão para prever o tipo de acidente usando o nosso *dataset* de exemplo, mas o resultado não foi nada bom: a acurácia (taxa de acertos) foi pouco maior do que 50%. Por outro lado, treinamos outra árvore para responder se o acidente foi de noite ou de dia e o resultado foi um pouco melhor, em torno de 71%. O interessante é que temos o mesmo resultado aplicando regressão logística.

Existem muitas técnicas para treinar modelos de classificação, tais como Random Forest, XGBoost, SVM etc.; inclusive, Redes Neurais podem ser treinadas para fazer classificação. Não é nossa ideia esgotar o tema aqui. Vamos ver apenas mais uma, cuja estratégia é diferente e bastante intuitiva: a técnica do **vizinho mais próximo** (*nearest neighbors* ou, mais exatamente, *K-nearest neighbors - KNN*). O conceito é bem simples: classificar um objeto como pertencendo à mesma classe que os objetos conhecidos mais parecidos com ele. Mas o que é "ser parecido" quando um objeto é descrito por um conjunto de valores de atributos? Na técnica do vizinho mais próximo, interpretamos os valores dos atributos como coordenadas em um espaço com **n** dimensões, onde **n** é o número de atributos no *dataset* (menos o atributo relativo à classe). No exemplo dos acidentes, temos 11 atributos; logo, cada acidente é representado por um ponto em um espaço de 10 dimensões. Interpretando dessa forma, podemos definir alguma forma de *distância* entre os objetos, de tal forma que, quanto mais parecidos, menor a distância entre eles. Dois acidentes envolvendo apenas um táxi e uma moto, ocorridos de dia, com o mesmo número de feridos, são mais próximos entre si (portanto, similares) do que de outro envolvendo dois ônibus à noite, por exemplo. Assim, se estamos tentando prever o tipo de acidente, é intuitivo imaginar que dois acidentes parecidos possam ser do mesmo tipo (classe).

Na técnica do vizinho mais próximo, definimos quantos vizinhos vamos examinar para decidir a classe de um objeto; esse valor é chamado *K* (daí o nome em inglês dessa técnica: *K-nearest neighbors*). Isso é necessário porque seria impraticável comparar cada novo objeto que desejamos classificar com todos os outros em nossos dados! Note que, apesar de ser muito intuitiva e fácil de implementar, isso não garante que o algoritmo

CC BY-NC-SA 4.0 VICARI, Rosa Maria; GALAFASSI, Cristiano; BRACKMANN, Christian; OLIVEIRA, Flávio.
IA@Escola. 2025. Disponível em: www.computacional.com.br/ia.

do vizinho mais próximo vá funcionar bem em todos os casos: isso depende do quanto a classe que desejamos aprender se relaciona com os outros atributos.

Regressão

A regressão é um conjunto de técnicas de aprendizado supervisionado onde os valores alvos são valores contínuos – diferente da classificação, onde o atributo alvo tem um conjunto limitado de valores (dia/noite, tipo de acidente). Em nosso exemplo, se quisermos treinar um modelo para prever o número de vítimas fatais, devemos usar regressão, porque esse número pode ser um inteiro positivo qualquer (incluindo 0) – embora números muito altos, na prática, não ocorram. Exemplos de algoritmos de regressão incluem:

- **Regressão Linear**: Modela a relação entre uma variável dependente e uma ou mais variáveis independentes de forma linear.
- **Regressão Polinomial**: Modela a relação como um polinômio quando o erro da regressão linear é muito alto.
- **Regressão usando Árvores de Decisão**: Técnicas originalmente usadas para classificação podem ser aplicadas em regressão.

As técnicas de regressão possuem um grande número de aplicações: predição de tempo de viagens, previsão de preços ou de demanda de produtos, e muito mais.

Como Avaliar os Modelos?

Como dissemos anteriormente, para saber se o modelo já está pronto para ser usado, temos que fazer uma validação do desempenho: aplicamos o modelo sobre o conjunto de validação e medimos como ele se saiu. Mas como medir? Quando falamos de classificação, usamos o exemplo dos acidentes e consideramos a taxa de acertos do modelo para falar do desempenho dele. Essa é uma das métricas mais clássicas e é chamada de **acurácia** – simplesmente vemos o percentual de acertos. A fórmula da acurácia é bem simples:

A = (número de acertos) / (número total de linhas)

Essa é a métrica mais simples para medir a qualidade de modelos de classificação. Há outras, como F1, AUC, precisão, *recall* etc., que são mais robustas e/ou podem ser usadas para situações específicas.

Note que a acurácia só faz sentido para aprendizagem supervisionada por classificação, porque, neste cenário, ou o modelo acerta – predizendo a classe correta – ou erra. Na regressão, precisamos avaliar de outra forma: se a resposta era 10 e o modelo predisse 9, isso é diferente de uma predição de 1 ou de 10,5, pois estamos lidando com uma faixa de valores numéricos, e não classes. Para modelos de regressão, medimos o desempenho através do conceito de **erro médio**, onde o erro é a diferença (em valor absoluto) entre o valor correto e o valor obtido pelo modelo. No exemplo, acima, o erro foi de |10-9| = 1 no primeiro caso, 9 o segundo e 0,5. Se nosso conjunto de teste fosse composto apenas desses exemplos, o erro médio seria (1+9+0,5)/3=3,5. Isso é pouco ou muito? Este é outro detalhe importante: o erro é medido nas mesmas unidades do atributo alvo, e para avaliar se o erro é pequeno ou não, temos que ver a escala de valores do *dataset* e do problema que estamos estudando. Um erro de 3,5 minutos em uma viagem que pode levar 2 horas é pequeno; já 3,5 segundos em uma corrida de 100 metros rasos é decisiva.

O erro médio é chamado de MAE (*mean absolute error* – erro médio absoluto) e é a métrica mais simples para avaliar modelos de regressão. Na prática, é mais frequente usar as variações MSE (*mean squared error*

CC BY-NC-SA 4.0 VICARI, Rosa Maria; GALAFASSI, Cristiano; BRACKMANN, Christian; OLIVEIRA, Flávio.
IA@Escola. 2025. Disponível em: www.computacional.com.br/ia.

–erro médio quadrático) ou RMSE (*root mean squared error* – raiz quadrada do erro médio quadrático), quando queremos dar mais importância a erros maiores.

Cuidados e considerações

A essa altura, deve ter ficado visível que os resultados de um modelo de aprendizagem supervisionado dependem *muito* de termos um bom conjunto de dados e de conduzirmos um bom treinamento, independentemente de ser uma aplicação de classificação ou de regressão. Alguns cuidados são especialmente importantes (já falamos de alguns deles, mas incluímos aqui devido a sua importância):

- Codificar atributos categóricos: os algoritmos de aprendizagem supervisionada esperam números como entrada; assim, se tivermos um atributo como dia da semana, por exemplo, temos que substituir os nomes dos dias por números (p.ex. 1=domingo, 2=segunda etc.).
- Normalização: se tivermos atributos com escalas muito diferentes (idade e altura, p.ex.), é fundamental colocá-los na mesma escala.
- Tratamento de *outliers*: um *outlier* é um valor excepcional ou até mesmo impossível, como uma altura negativa ou um salário de R$1 milhão por mês. Mesmo que não seja impossível, é bom filtrar esses casos, pois prejudicam o funcionamento do algoritmo de aprendizagem. Se não é um erro, devem ser tratados em separado.
- Pressupostos: por exemplo, regressão linear só faz sentido se existe uma proporcionalidade linear ou bem próxima de linear entre os atributos de entrada e o atributo alv. Devemos conferir se o algoritmo escolhido pode ser aplicado com segurança no problema que estamos estudando.
- *Overfitting/underfitting*: *overfitting* é quando o modelo fica muito ajustado aos dados de treinamento, tendo um desempenho muito bom – até demais – seguido de maus resultados na validação. Já o *underfitting* ocorre quando o treinamento não consegue gerar um modelo com bom desempenho. No primeiro caso, devemos reavaliar a composição dos conjuntos de treino e teste, possivelmente aumentando o volume e variedade dos dados de treino. No segundo, devemos considerar também se estamos usando a melhor técnica para o problema.
- Seleção/engenharia de atributos: é importante escolher atributos que realmente influenciem o nosso alvo. Aqui é importante estudar bem o problema! Às vezes é fácil descartar um atributo irrelevante (a cor da roupa do paciente não influencia sua condição de saúde), mas outros casos podem exigir uma avaliação estatística (será que a data do acidente influencia o tipo?).
- Balanceamento do *dataset*: como vimos, é importante fornecer exemplos de todos os casos que podem ocorrer na realidade, em proporções realistas. Porém, é importante entender se uma classe, por exemplo, está mal representada no dataset ou se é naturalmente rara (por exemplo, acidentes de carro com explosões são naturalmente mais raros). No primeiro caso, devemos corrigir o balanceamento; no segundo, talvez seja melhor usar outra técnica.

Resumo

Nesta aula, abordamos diversos métodos de aprendizagem supervisionada, quando temos exemplos já rotulados com as respostas corretas, o que nos permite um treinamento eficiente do modelo. Um ponto importante é que, pelo que discutimos, fica muito claro que não há mágica em modelos de IA criados por aprendizagem: é preciso muito trabalho humano e cuidados éticos para termos um resultado confiável!

3. Atividade – estudando os conceitos – atividade desconectada

CC BY-NC-SA 4.0 VICARI, Rosa Maria; GALAFASSI, Cristiano; BRACKMANN, Christian; OLIVEIRA, Flávio.
IA@Escola. 2025. Disponível em: www.computacional.com.br/ia.

Uma atividade interessante é criar um jogo com peças de papel ou de LEGO, por exemplo. As peças devem ser rotuladas com os rótulos “sim” e “não”, ou “certa” e “errada”, por exemplo. Para isso, você cria uma regra (que só você conhece) para classificar as peças. Os alunos devem, a partir do exame das peças, descobrir a regra. Eles podem fazer isso simulando o algoritmo do vizinho mais próximo, por exemplo.	
Avaliação	- Os alunos podem ser avaliados de acordo com sua interação e participação nas atividades e discussões.
Material Complementar	Aqui um pouco mais de detalhe sobre regressão linear: https://www.youtube.com/watch?v=-PGDAbkLzSw&list=PLuHvZ6hWZut1P51MFngSWl7neCg8V0nEf Aqui você encontra um exemplo usando KNN: https://www.youtube.com/watch?v=_3uA9tGBx0s

CC BY-NC-SA 4.0
VICARI, Rosa Maria; GALAFASSI, Cristiano; BRACKMANN, Christian; OLIVEIRA, Flávio. IA@Escola. 2025. Disponível em: www.computacional.com.br/ia.