



Ph.D. Professor Aluisio Igor Rego Fontes

Gustavo de Freitas Rodrigues

José Matheus Bento

**Autor da apostila**

Ph.D. Professor Aluisio Igor Rego Fontes

**Instrutor do curso**

Larissa Jéssica Alves – Analista de Suporte Pedagógico

**Revisão da apostila**



**Autor**

**Aluisio Igor Rêgo Fontes**

Aluisio I. R. Fontes possui graduação em Engenharia da Computação pela Universidade Federal do Rio Grande do Norte (2010), mestrado (2012) e doutorado (2015) em Engenharia Elétrica e Computação pela Universidade Federal do Rio Grande do Norte. Atualmente é professor do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte campus Pau dos Ferros, onde coordena o Laboratório de Análise de Dados e Inteligência Computacional (NADIC). Por mais de 13 anos, vem se dedicando a criação de soluções inovadoras utilizando inteligência artificial e análise de dados para inserir tecnologias em várias organizações. No contexto de sistemas corporativos, desenvolvi soluções com acesso massivo de usuários, alta disponibilidade e robotização de processos. Na pesquisa acadêmica sou autor/co-autor de mais de 15 publicações em periódicos internacionais nas áreas de Inteligência Artificial e teoria da informação, tendo publicado em veículos de grande reputação nestas áreas. Sou revisor de periódicos internacionais (e.g., IEEE Signal Processing, Expert System with Application, IEEE Access e eurasip journal on advances in signal processing). Tenho experiência na área de Engenharia da Computação, com ênfase em Sistemas de Computação, atuando principalmente nos seguintes temas: Processamento digital de sinais, inteligência Computacional, Teoria da Informação, Correntropia, Processamento em Big Data, desenvolvimento de sistemas corporativos e computação de alto desempenho.



**José Matheus Bento**

Fiz curso técnico em informática no IFRN, campus Pau dos Ferros. Atualmente estou cursando Análise e Desenvolvimento de Sistemas no mesmo instituto. Sempre gostei de tecnologia, e assim que aprendi a programar, procurei aprender diversas tecnologias, para me aprimorar e encontrar aquela que mais gostasse. Desde 2020 participo do NADIC onde trabalhei com desenvolvimento backend utilizando o framework Django, em 2021 iniciei um estágio na JusInvestimens ainda focado em backend. No ano de 2022 fui contratado tanto para dar continuidade ao sistema da empresa como também para desenvolver um aplicativo nativo para Android. Tenho muito interesse em redes neurais e aprendizado de máquina, pois desenvolvi um sistema de reconhecimento de placas de veículos utilizando IA.



**Gustavo de Freitas Rodrigues**

Sou técnico de informática no IFRN - Campus Pau dos Ferros, estudante na graduação de Tecnologia da Informação, com interesse em ciências da computação pela UFRN (Universidade Federal do Rio Grande Do Norte). Tem experiência com circuitos e eletrônica, já tendo trabalhado na construção de robôs, utilizando arduino. Participei no NADIC (Núcleo de Análise de Dados e Inteligência Computacional), na área de Deep Learning, Computer Vision e IoT. Atualmente, estou trabalhando e aprofundando meus conhecimentos em ciência de dados e aprendizado de máquina.

**APRESENTAÇÃO**

A presente apostila é um instrumento teórico que complementa o curso de capacitação de Inteligência Artificial. Nela, veremoso primeiro módulo daementa do curso. Este material é baseado em artigos científicos, periódicos, revistas científicas e livros científicos. É extremamente recomendável ao aluno que, ao final da leitura de cada seção, realize os exercícios propostos e acesse os materiais indicados nas referências bibliográficas, para aprofundar a leitura desse material e complementar o que foi lido aqui.

***Boa Leitura !!***

**Sumário**

[**1 Divisão do Dataset 6**](#_heading=h.1yndlq12yvzb)

[1.1 Método K-Fold 6](#_heading=h.p4s5re954a7d)

[1.2 Método Hold-Out 6](#_heading=h.iihqvh3fx66d)

[**2 Medidas de desempenho 7**](#_heading=h.ydi4ep9vsmq8)

[2.1 Precision 7](#_heading=h.b827771ofxgn)

[2.2 Accuracy 7](#_heading=h.8j1mepp5j5nc)

[2.3 Recall 7](#_heading=h.4f2a94k1wplq)

[2.4 F1 Score 8](#_heading=h.40jwo0nxwxdy)

[**3 Algoritmos de Machine Learning 8**](#_heading=h.aj2un24uei39)

[3.1 Regressão Linear 9](#_heading=h.q1q6087omvcp)

[3.2 Regressão Logística 10](#_heading=h.2qjx2r8amp4m)

[3.3 KNN (K-Nearest Neighbors) 11](#_heading=h.vispnzfhi2lv)

[3.4 Árvore de Decisão (J48) 11](#_heading=h.qkj08iddyb1z)

# Divisão do *Dataset*

A divisão de conjuntos de dados é uma etapa crucial no processo de treinamento e avaliação de modelos de aprendizado de máquina. Duas abordagens comuns para essa divisão são o método de *hold-out* e o método de *k-fold cross-validation*.

## Método *K-Fold*

consiste em dividir o conjunto total de dados em k subconjuntos mutuamente exclusivos do mesmo tamanho. Um subconjunto é utilizado para teste e os k - 1 restantes são utilizados para estimação dos parâmetros e calcula-se a acurácia do modelo.

O modelo é treinado k vezes, cada vez usando k-1 folds como conjunto de treinamento e o fold restante como conjunto de teste. O desempenho do modelo é a média das métricas de avaliação obtidas em cada uma das k iterações.

## Método *Hold-Out*

Consiste em dividir o conjunto total de dados em dois subconjuntos mutuamente exclusivos, um para treinamento e outro para teste (validação). O conjunto de dados pode ser separado em quantidades iguais ou não. É possível ter 2/3 dos dados para treinamento e o 1/3 restante para teste. Esta abordagem é indicada para grande quantidade de dados. Quando o conjunto é pequeno, o erro calculado na predição pode sofrer muita variação

Geralmente se utiliza de 70% a 90% do *dataset* para treino e de 30% a 10% para testes. O modelo é treinado no conjunto de treinamento e avaliado no conjunto de teste para estimar sua capacidade de generalização para dados não vistos.

# Medidas de desempenho

## Precision

Precision mede a precisão das previsões positivas feitas pelo modelo. É a proporção de instâncias classificadas como positivas que realmente são positivas.

## Accuracy

Uma medida geral da corretude do modelo. Representa a proporção de todas as previsões corretas em relação ao número total de instâncias.

## Recall

Mede a capacidade do modelo de identificar todas as instâncias positivas. É a proporção de instâncias positivas que foram corretamente identificadas pelo modelo em relação ao total de instâncias positivas.

## F1 *Score*

É importante notar que existe uma compensação entre *precision* e recall. Aumentar um pode diminuir o outro, e encontrar um equilíbrio depende do contexto do problema. Às vezes, a F1 Score (uma média harmônica de *precision* e recall) é usada para fornecer uma única métrica que leva em consideração ambos os aspectos.

# Algoritmos de Machine Learning

Os algoritmos de aprendizado de máquina (machine learning) desempenham um papel crucial na análise e extração de padrões a partir de dados, possibilitando a automação de tarefas e a tomada de decisões em diversos domínios. Como já vimos anteriormente podem ser divididos em 3 tipos:

* **Aprendizado supervisionado**

Nesse tipo de aprendizado, o algoritmo é treinado usando um conjunto de dados rotulados, onde as saídas desejadas já são conhecidas.

* **Aprendizado não supervisionado**

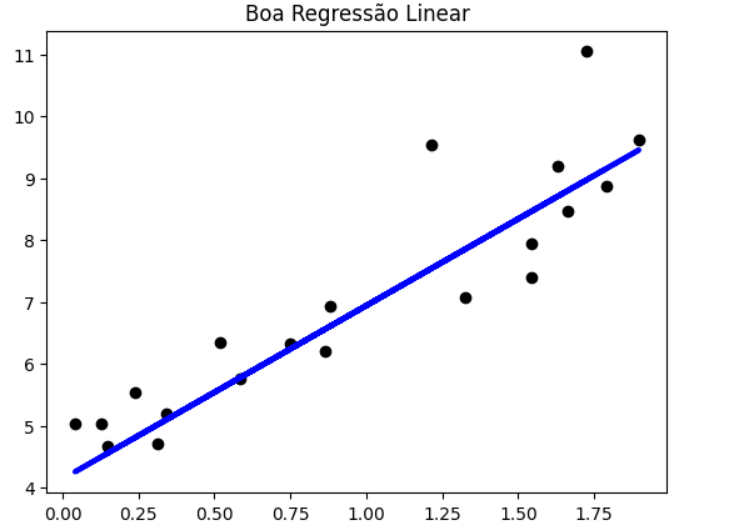
Aqui, o algoritmo é treinado com dados não rotulados, buscando identificar padrões ou estruturas intrínsecas nos dados.

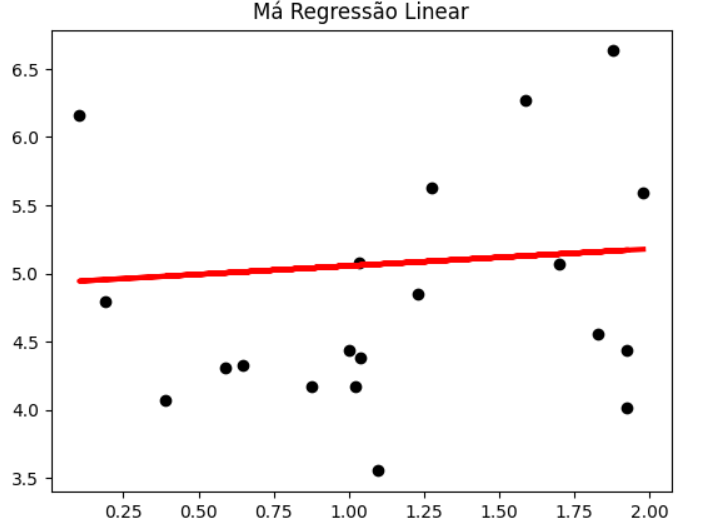
* **Aprendizado por reforço**

Nesse paradigma, o modelo (agente) aprende a tomar decisões sequenciais para maximizar uma recompensa ao interagir com um ambiente dinâmico.

## Regressão Linear

A regressão linear é um método estatístico que tenta modelar a relação linear entre uma variável dependente (ou resposta) e uma ou mais variáveis independentes (ou preditoras). Em outras palavras, a regressão linear procura encontrar a linha reta que melhor se ajusta aos dados, de modo a prever a variável dependente com base nas variáveis independentes.





Esse algoritmo é muito utilizado para fazer previsões em que a variável desejada há uma relação linear com outras variáveis, por exemplo, prever o preço de uma casa, com base no terreno, metragem e número de quartos.

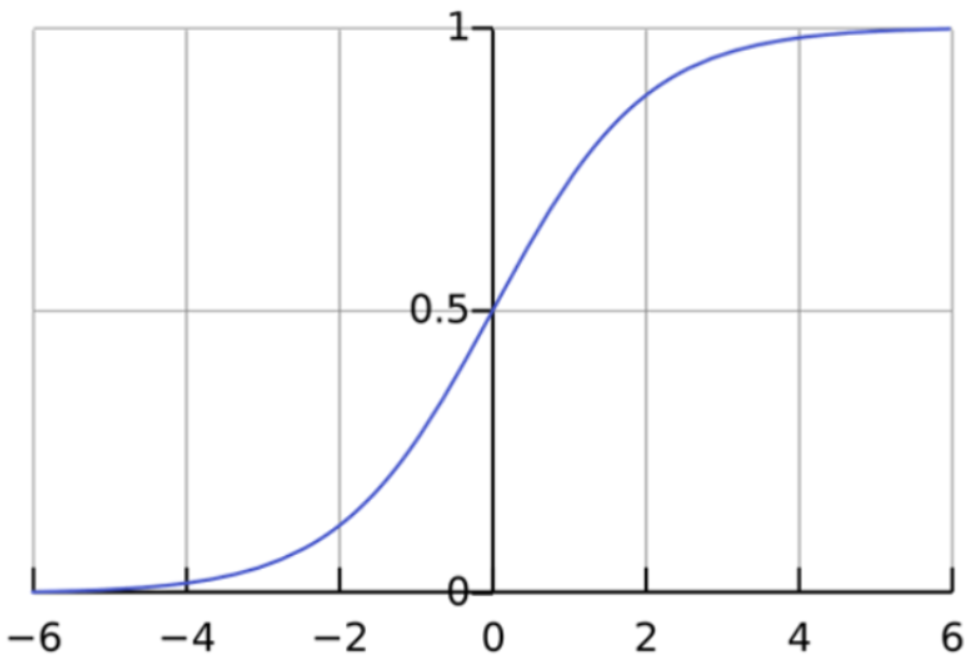
Os três índices mais comuns para a validação de uma regressão linear são:

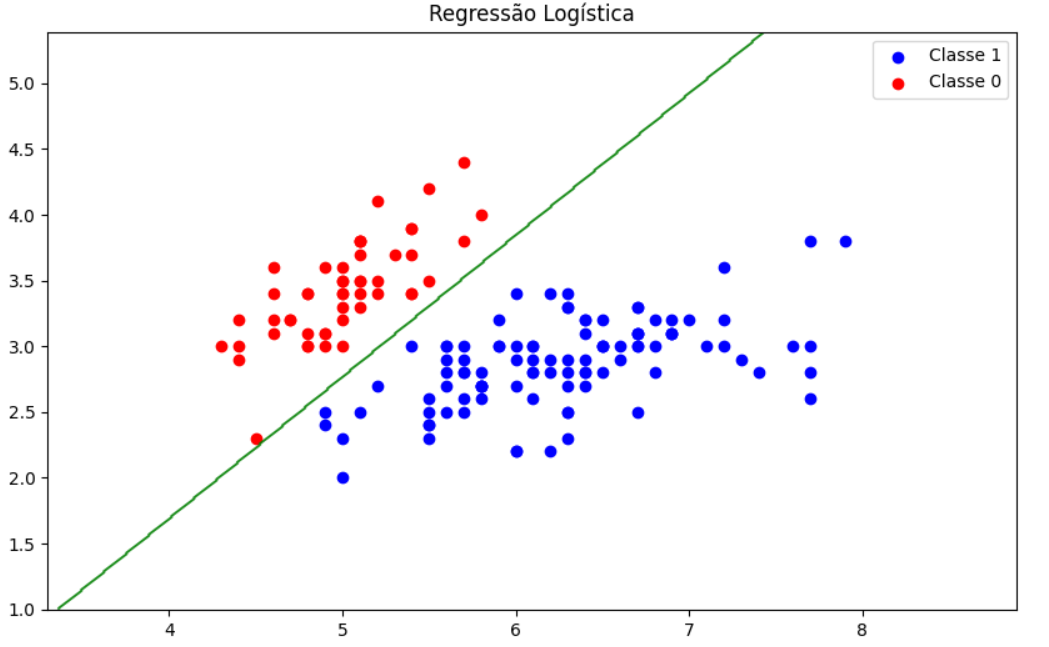
* **R2**
  + Mede a proporção da variabilidade nos dados explicada pelo modelo. Ou seja, quantifica quanto a nossa regressão explica o conjunto de dados.
* **MSE (*Mean Squared Error)***
  + Média das diferenças ao quadrado entre as predições e os valores reais. Ou seja, quantifica se nossa regressão está errando muito ou não os valores preditos.
* **RMSE *(Root Mean Squared Error)***
  + Raiz quadrada do MSE

## Regressão Logística

A regressão logística é um método estatístico utilizado para modelar a probabilidade de uma variável dependente binária, ou seja, aquela que tem apenas dois possíveis resultados, como 0 ou 1, verdadeiro ou falso, positivo ou negativo, etc. Diferentemente da regressão linear, que é usada para prever valores contínuos, a regressão logística é específica para problemas de classificação binária.

O modelo de regressão logística utiliza a função logística (também conhecida como função sigmóide) para transformar a combinação linear das variáveis independentes ponderadas pelos coeficientes em uma probabilidade.





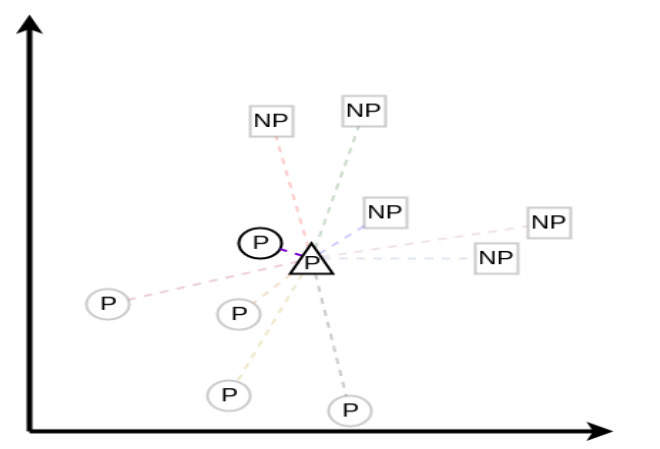
## KNN (K-Nearest Neighbors)

É um algoritmo de aprendizado supervisionado utilizado para classificação e regressão. É um método simples, mas eficaz, baseado no princípio da proximidade, que se encaixa tanto em tarefas de classificação quanto em tarefas de regressão.

Para classificar ou prever um ponto de dados, o algoritmo KNN procura pelos "k" vizinhos mais próximos desse ponto no espaço de características. A classe ou valor previsto é determinado pela maioria dos "k" vizinhos mais próximos em tarefas de classificação ou pela média em tarefas de regressão.

1. Escolha do parâmetro k: O número de vizinhos mais próximos a serem considerados (k) é um hiperparâmetro do modelo. Escolher um valor apropriado para k é importante, pois pode afetar o desempenho do modelo.
2. Cálculo da Distância: A distância entre os pontos de dados é calculada utilizando métricas como a distância euclidiana. Quanto menor a distância, mais próximos os pontos estão.
3. Votação ou Média: Para tarefas de classificação, é feita uma votação entre os k vizinhos mais próximos para determinar a classe mais frequente. Para tarefas de regressão, é calculada a média dos valores dos k vizinhos mais próximos.

KNN é considerado um algoritmo não-paramétrico, pois não faz suposições explícitas sobre a forma funcional dos dados. Seu desempenho pode ser afetado por outliers, pois eles podem influenciar significativamente a determinação dos vizinhos mais próximos. E por fim, o custo computacional do KNN aumenta à medida que o tamanho do conjunto de dados cresce, pois é necessário calcular as distâncias entre o ponto de interesse e todos os pontos no conjunto de treinamento.



## Árvore de Decisão (J48)

J48 é um algoritmo de árvore de decisão utilizado para tarefas de classificação. O nome "J48" é derivado do nome do algoritmo original, C4.5, desenvolvido por Ross Quinlan. A implementação Java do C4.5 ficou conhecida como J48.

Princípios Básicos da Árvore de Decisão:

* Divisão Recursiva: A árvore de decisão é construída de forma recursiva, dividindo o conjunto de dados com base em diferentes atributos. A cada nó da árvore, a escolha do atributo de divisão é feita com base em critérios como ganho de informação.
* Critério de Parada: O processo de construção da árvore continua até que um critério de parada seja atendido. Isso pode incluir a profundidade máxima da árvore, o número mínimo de amostras em um nó ou a pureza dos nós.
* Folhas da Árvore: As folhas da árvore representam as classes de saída. Quando uma instância percorre a árvore, a decisão é tomada com base nas condições encontradas nos nós.

Porém existe o problema de *Overfitting* que pode ocorrer quando a árvore de decisão se ajusta demais aos dados de treinamento, capturando padrões específicos que podem não generalizar bem para novos dados. Para evitar isso existe a Poda (pruning) que remove partes da árvore que não contribuem significativamente para a precisão do modelo em dados não vistos.

Existem dois tipos principais de poda em árvores de decisão:

* **Pré-Poda (*Pre-pruning*):**

A pré-poda envolve a definição de critérios de parada durante a construção da árvore, impedindo que ela cresça além de certos limites. Exemplos de critérios de pré-poda incluem limitar a profundidade máxima da árvore, o número mínimo de amostras em um nó ou o número mínimo de amostras necessárias para continuar dividindo um nó.

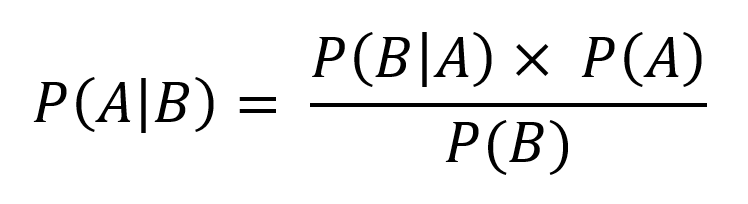
* **Pós-Poda (*Post-pruning*):**

A pós-poda envolve a construção inicial da árvore sem restrições e, em seguida, a remoção de partes da árvore que não contribuem para a melhoria do desempenho do modelo. Geralmente, a pós-poda é realizada retrocedendo da folha em direção à raiz da árvore e removendo ramos que não contribuem significativamente para a redução do erro de generalização.

## *Naive Bayes*

O algoritmo Naive Bayes é um método estatístico baseado no teorema de Bayes, que é utilizado para realizar classificação e é especialmente aplicado em problemas de aprendizado de máquina supervisionado. Ele é chamado de "naive" (ingênuo) porque faz uma suposição simplificada e forte de independência condicional entre os atributos.

O teorema de Bayes é uma fórmula que descreve a probabilidade condicional de um evento A, dado que um evento B ocorreu. Para dois eventos A e B, tem-se:



Na classificação com o Naive Bayes, os eventos A e B referem-se à classe da instância e aos atributos da instância, respectivamente. O O algoritmo irá calcular as probabilidades a priori de cada classe no conjunto de treinamento. A probabilidade a priori de uma classe é a probabilidade de uma instância pertencer a essa classe, sem levar em consideração os atributos.

Também, para cada atributo, o algoritmo calcula as probabilidades condicionais, que representam a probabilidade de um valor específico do atributo ocorrer, dado que a instância pertence a uma determinada classe.

Para que no fim a classificação seja realizada atribuindo a instância à classe com a maior probabilidade condicional, dada a instância.

## Redes Neurais Artificiais (MLP)

## Comitê de Classificadores

A ideia fundamental por trás dos *ensembles* é que a combinação de vários modelos fracos pode levar a um modelo mais robusto e poderoso do que qualquer um dos modelos individuais.

Existem várias técnicas de ensemble, sendo as mais comuns o Bagging, o Boosting e o Stacking.

* *Bagging (Bootstrap Aggregating)*:

Múltiplos modelos são treinados independentemente em diferentes subconjuntos aleatórios (com reposição) do conjunto de treinamento original. Os modelos são frequentemente treinados usando o mesmo algoritmo de base, mas em diferentes subconjuntos de dados. A previsão final é obtida agregando as previsões de cada modelo. Para problemas de classificação, uma votação pode ser realizada para determinar a classe mais frequente, enquanto em problemas de regressão, pode ser calculada a média das previsões.

* *Boosting*:

Modelos são treinados sequencialmente, onde cada novo modelo é treinado para corrigir os erros do modelo anterior. Logo, Instâncias mal classificadas pelo modelo atual recebem mais peso nas iterações subsequentes, dando mais atenção a casos difíceis. Para que por fim, a previsão final seja uma combinação ponderada das previsões dos modelos individuais.

* *Stacking*:

Diferentes modelos de aprendizado de máquina são treinados e suas previsões são usadas como entrada para um meta-modelo, que realiza a previsão final. Os modelos de base podem ser de tipos diferentes e são escolhidos para explorar diversas abordagens. A ideia é capturar as nuances que cada modelo de base pode ter em diferentes partes do espaço de características.

**Considerações finais**

Dessa forma terminamos o conteúdo do curso de Inteligência Artificial. Nele foi possível se aprofundar nessa área, vendo desde a preparação dos dados, as formas e estratégias de treinamento, o funcionamento e criação de redes neurais, as tecnologias utilizadas na área e os principais algoritmos utilizados atualmente.

Esperamos que a partir dos desafios e projetos desenvolvidos possa se levar uma experiência real de como se desenvolve a Inteligência Artificial atualmente. Assim como entender a importância dessas tecnologias atualmente, automatizando tarefas repetitivas ou solucionando problemas práticos da realidade.

Ter esses conhecimentos pode abrir um futuro novo, pois é uma área com muitas possibilidades e que avança constantemente, cada vez mais presente no nosso cotidiano. Logo, dominar seu desenvolvimento pode abrir novas oportunidades profissionais e pessoais.

**Referências**

FAYYAD, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). **From data mining to knowledge discovery in databases.** AI Magazine, 17(3), 37–54.

GÉRON, A. (2022). **Mãos à Obra: Aprendizado de Máquina com Scikit-Learn & TensorFlow**. Porto Alegre: Editora Artmed.

HAYKIN, Simon S. **Neural Networks and Learning Machines**. 3rd ed. Pearson Education, 2009.

WITTEN, I. H.; FRANK, E.; HALL, M. A.; PAL, C. J. **Data Mining: Practical Machine Learning Tools and Techniques.** 4. ed., Morgan Kaufmann, 2016.

**BOM CURSO!**