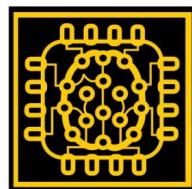


COLETÂNEA DE DICAS, ORIENTAÇÕES E REFERÊNCIAS EM INTELIGÊNCIA ARTIFICIAL

Dr. Arnaldo de Carvalho Junior

Versão 6 - 2024



EAILab

Laboratório de Inteligência Artificial Embarcada
Instituto Federal de São Paulo

Sumário

INTRODUÇÃO.....	6
1. O ICEBERG DA INTELIGÊNCIA ARTIFICIAL.....	7
2. O UNIVERSO DA INTELIGÊNCIA ARTIFICIAL.....	9
3. COMO A INTELIGÊNCIA ARTIFICIAL FUNCIONA.....	10
1. FUNCIONAMENTO DA INTELIGÊNCIA ARTIFICIAL (IA) EM POUCAS PALAVRAS.....	10
2. IA EM POUCAS PALAVRAS.....	11
3. COMO A IA APRENDE?.....	13
4. TUDO PARA APRENDER SOBRE IA EM UM QUADRO.....	16
5. INTELIGÊNCIA ARTIFICIAL EXPLICADA.....	19
1. IA EXPLICADA.....	19
2. DOZE CONCEITOS PARA SE ESPECIALIZAR EM IA.....	20
3. ROTEIRO PARA DOMÍNIO EM IA.....	22
4. ROTEIRO DO ENGENHEIRO DE IA E ML.....	24
5. FALSOS CONCEITOS SOBRE IA.....	26
6. O ECOSISTEMA DE IA – FERRAMENTAS E CURSOS.....	30
7. COMPREENDENDO IA, ML, DI E DS.....	34
8. INTELIGÊNCIA ARTIFICIAL x APRENDIZADO DE MÁQUINA X REDE NEURAL ARTIFICIAL X APRENDIZADO PROFUNDO.....	36
9. DATASETS DE ACESSO LIVRE PARA QUALQUER PROJETO DE IA!.....	39
10. PRINCIPAIS ALGORITMOS DE INTELIGÊNCIA ARTIFICIAL PARA LEIGOS.....	41
11. CONCEITOS ESSENCIAIS DE INTELIGÊNCIA ARTIFICIAL PARA INICIANTES.....	47
12. HIERARQUIA DA INTELIGÊNCIA ARTIFICIAL - DA FUNDAÇÃO AO NÍVEL AVANÇADO.....	49
13. PRINCIPAIS FERRAMENTAS DE IA.....	52
1. AS 10 FERRAMENTAS DE IA MAIS UTILIZADAS EM 2024.....	52
2. AS 24 PRINCIPAIS FERRAMENTAS DA INTELIGÊNCIA ARTIFICIAL (IA)!.....	53
14. TERMOS MAIS UTILIZADOS EM INTELIGÊNCIA ARTIFICIAL.....	56
15. VINTE ABREVIAÇÕES DE INTELIGÊNCIA ARTIFICIAL.....	59
16. ROTEIRO DE PROJETOS DE ML E IA.....	61
1. O ROTEIRO DO APRENDIZADO DE MÁQUINAS.....	61
2. FLUXO DE TRABALHO PARA GUIAR O INICIANTE EM APRENDIZADO DE MÁQUINAS.	63

3. CICLO DE VIDA DE PROJETO DE ML.....	65
4. PROJETOS DE INTELIGÊNCIA ARTIFICIAL: DA CONCEPÇÃO À IMPLANTAÇÃO.....	67
17. MELHORES RECURSOS PARA APRENDER SOBRE APRENDIZADO DE MÁQUINA.....	71
3. CURSOS IA NVIDIA.....	74
4. LIVROS GRATUITOS [5].....	75
18. ALGORITMOS DE APRENDIZADO DE MÁQUINA.....	78
1. EXPLORANDO ALGORITMOS DE ML.....	78
2. DEZ TÉCNICAS FUNDAMENTAIS DE ML.....	79
3. TIPOS DE ALGORITMOS DE ML.....	81
4. APRENDIZADO SUPERVISIONADO OU NÃO-SUPERVISIONADO?.....	89
5. RESUMO.....	90
19. A RODA DA INTELIGÊNCIA ARTIFICIAL E APRENDIZADO DE MÁQUINAS.....	95
20. TODAS AS ESTRUTURAS DE DADOS SÃO APENAS VARIÁVEIS.....	97
21. A ATMOSFERA PYTHON.....	100
22. O PODER DA ANÁLISE DE DADOS COM PYTHON.....	102
23. ROTEIRO DO CIENTISTA DE DADOS.....	105
1. ROTEIRO DO CIENTISTA DE DADOS.....	106
2. QUINZE RESPONSABILIDADES-CHAVE DO CIENTISTA DE DADOS.....	107
3. PAPEIS DO ENGENHEIRO DE SOFTWARE, ENGENHEIRO DE DADOS, CIENTISTA DE DADOS E ANALISTA DE DADOS.....	109
24. ONZE TIPOS DE VARIÁVEIS EM DATASET.....	112
25. FUNDAMENTOS DE ESTATÍSTICA.....	115
1. PENSAMENTO ESTATÍSTICO: CONCEITOS OBRIGATÓRIOS.....	115
2. AS DISTRIBUIÇÕES MAIS IMPORTANTES EM CIÊNCIA DE DADOS E IA.....	117
3. VALIDAÇÃO CRUZADA: A ABORDAGEM DO CONJUNTO DE VALIDAÇÃO.....	120
26. ANÁLISE AVANÇADA DE SÉRIES DE DADOS.....	124
1. SÉRIES TEMPORAIS AVANÇADAS: TIPOS, MÉTODOS, APlicativos e PRINCIPAIS BIBLIOTECAS PYTHON.....	124
2. ALGORITMOS DE ML PARA MODELAGEM PREDITIVA.....	128
27. SINTONIA DE HIPERPARÂMETROS EM APRENDIZADO DE MÁQUINA.....	131
1. IMPACTO DOS HIPERPARÂMETROS SINTONIZADOS EM ALGORITMOS ESPECÍFICOS DE ML.....	131
2. AJUSTE DOS HIPERPARÂMETROS EM ALGORITMOS ESPECÍFICOS DE ML.....	132

3. RESUMO DOS HIPERPARÂMETROS EM ML.....	135
28. ENTENDENDO A REGRESSÃO LINEAR: UM CONCEITO CENTRAL EM ESTATÍSTICA....	138
29. MODELOS DE CLASSIFICAÇÃO POR LIMITES DE DECISÃO.....	141
30. AGRUPAMENTOS DE DADOS K-MEANS, GMM E DBSCAN.....	148
1. AGRUPAMENTO (<i>CLUSTERING</i>).....	148
2. ALGORITMO K-MEANS.....	148
3. A PRINCIPAL LIMITAÇÃO DO K-MEANS.....	149
4. GMM.....	151
5. DBSCAN.....	154
31. MÉTRICAS DE AVALIAÇÃO DE MODELO PARA APRENDIZADO DE MÁQUINAS.....	157
32. QUAL O MODELO DE APRENDIZADO DE MÁQUINA IDEAL?.....	159
1. ESCOLHENDO O MODELO DE ML.....	159
2. RÁPIDA IMPLANTAÇÃO DE APRENDIZADO DE MÁQUINA.....	171
33. REDES NEURAIS CONVOLUCIONAIS.....	174
34. ALGORITMOS FREQUENTEMENTE UTILIZADOS PARA APRENDIZADO PROFUNDO....	178
35. ARQUITETURA DOS TRANSFORMADORES.....	181
36. EVOLUÇÃO DA ARQUITETURA DE IA: DO ML TRADICIONAL PARA A IA GENERATIVA... <td>185</td>	185
37. AS OPERAÇÕES DE APRENDIZADO DE MÁQUINA (MLOps).....	188
38. ADAPTADORES LORA e RAG.....	190
39. ROTEIRO DO CIENTISTA DE LLM.....	199
40. PASSO A PASSO DA APLICAÇÃO GENAI.....	203
41. APRENDIZADO PROFUNDO: EXEMPLOS DE CASOS.....	207
42. CONSTRUINDO UM MODELO DE APRENDIZADO DE MÁQUINAS.....	210
1. ETAPAS CHAVE PARA CONSTRUÇÃO DO MODELO DE ML.....	211
2. TREINAMENTO DO MODELO.....	211
3. AVALIAÇÃO DO MODELO.....	211
4. OTIMIZAÇÃO DO MODELO.....	211
5. APLICAÇÕES DO MUNDO REAL.....	212
6. DESAFIOS ENFRENTADOS NA CONSTRUÇÃO DE MODELOS.....	212
7. RESUMINDO, COMO A MODELAGEM DE ML FUNCIONA?.....	212
43. ROTEIRO PARA PROJETAR REDES NEURAIS.....	214
1. INTRODUÇÃO.....	214

2. COMO PROJETAR UMA REDE NEURAL.....	215
44. CHATGPT.....	219
1. ESCRITA IMEDIATA DO ChatGPT.....	219
2. VINTE REGRAS DE OURO PARA ESCREVER PROMPTS DE ChatGPT	223
3. QUINZE ERROS PARA EVITAR AO USAR O CHATGPT.....	225
4. COMO USAR O CHATGPT PARA AUXILIAR NA METODOLOGIA DA PESQUISA.....	226
ANEXO I – POSTS DO EAILAB.....	230
1. FUNÇÕES DE ATIVAÇÃO.....	230
2. REDES NEURAIS ARTIFICIAIS.....	230
3. PAL Et.....	230
4. CNN.....	230
5. ROTEIRO DE DATASETS PARA DL.....	230
6. PIPELINE PARA CLASSIFICAÇÃO DE IMAGENS.....	230

INTRODUÇÃO

Este documento reúne uma coletânea de cartões ou lâminas sobre assuntos relacionados a inteligência artificial (IA), aprendizado de máquina (*machine learning* – ML), Tiny ML e computação na borda (*edge computing*) tais como: siglas, termos mais utilizados, conceitos matemáticos, algoritmos, ferramentas, aplicativos, tutoriais e cursos.

O objetivo deste documento é servir como uma referência preliminar para que o pesquisador possa demarcar o roteiro do aprendizado em inteligência artificial, suas diferentes estratégias de aplicação, tendências e evolução.

Dr. Arnaldo de Carvalho Junior

EAILAB – IFSP

<https://eailab.labmax.org/>

Versão 6

Dezembro 2024.

1. O ICEBERG DA INTELIGÊNCIA ARTIFICIAL

Por: Denis Panjuta.

Editado por: Dr. Arnaldo de Carvalho Junior, Junho 04, 2024.

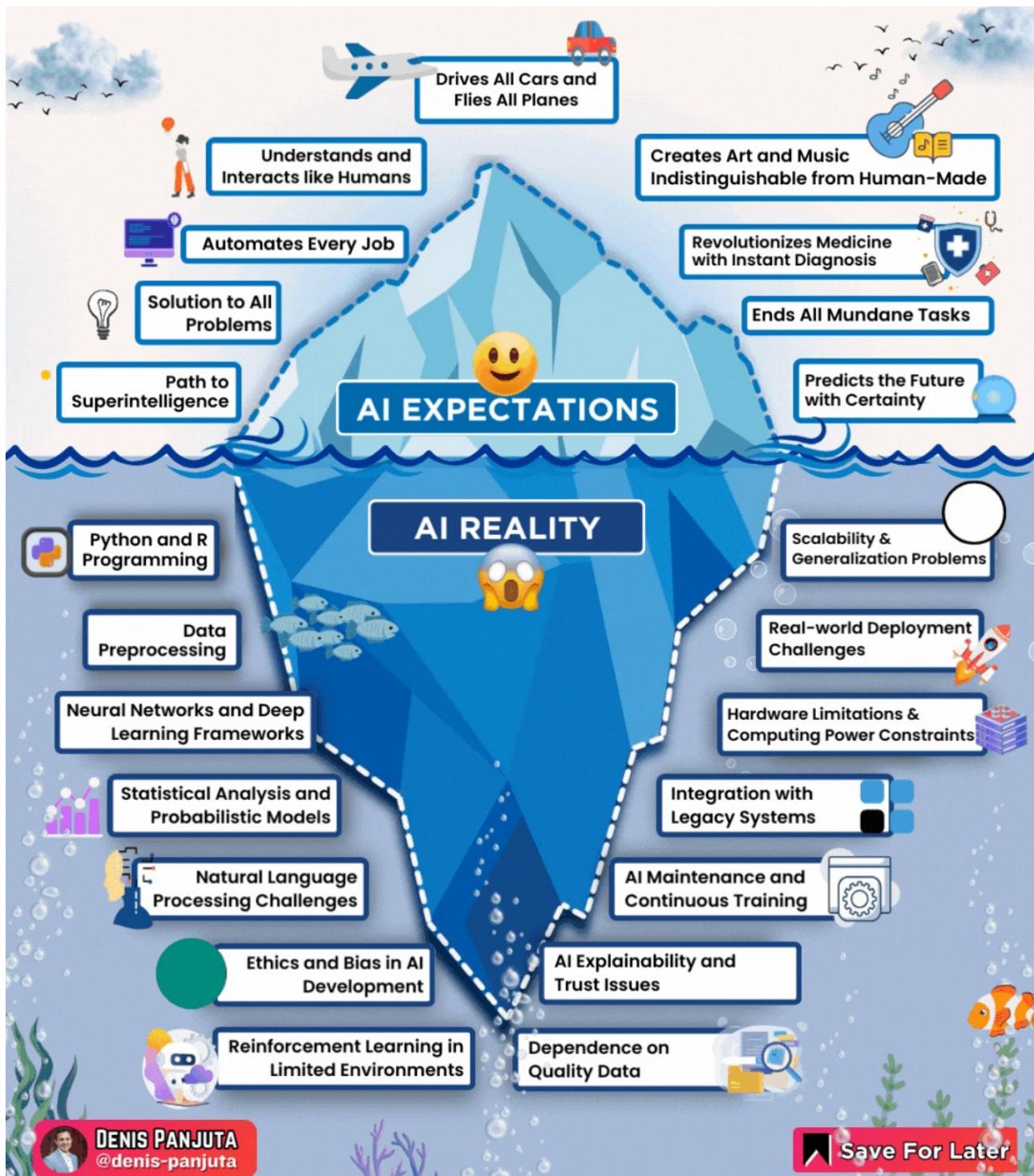


Figura 1 - O Iceberg de IA – Esclarecendo Conceitos Normalmente Equivocados de IA.

Fonte: Adaptado de [1].

O pico do iceberg brilha com aspirações:

- a) Aspirando a superinteligência e solução abrangente de problemas.
- b) Automatizando uma ampla gama de empregos e realizando interações humanas.
- c) Carros pioneiros sem motorista, aviação autônoma e arte gerada pela IA.
- d) Transformando os cuidados de saúde com diagnósticos rápidos e precisos.
- e) Erradicar tarefas monótonas e prever eventos futuros com precisão.

No entanto, a base submersa na realidade nos apresenta seus detalhes técnicos:

- a) Programação do núcleo em Python e R.
- b) A criticidade do pré -processamento meticuloso de dados.
- c) Os meandros das redes neurais e estruturas de aprendizado profundo.
- d) Análise estatística rigorosa e a imprevisibilidade de modelos probabilísticos.
- e) Superando desafios de processamento de linguagem natural.
- f) Abordando a ética e o viés nos sistemas de IA, e mais...

Referência:

[1] PANJUTA, D. The AI Iceberg, Machine Learning Community (Moderate), 2024.
Disponível em: https://media.linkedin.com/dms/image/D4E22AQG6T7Ejsgmipw/feedshare-shrink_800/0/1719096543753?e=1723075200&v=beta&t=wOO7Jmd6Wl8AyFcIzGfh3m4bllBkfCfF5pyO1OTmPk.
Acessado em Junho 27, 2024.

2. O UNIVERSO DA INTELIGÊNCIA ARTIFICIAL

Por: Abdul Rahim Roni.

Editado por: Dr. Arnaldo de Carvalho Junior, Junho 27, 2024.

Enquanto a inteligência artificial (IA) continua a remodelar o mundo, é crucial entender o amplo universo de técnicas e domínios dela. O infográfico da Figura 1 a seguir captura os principais componentes da IA.

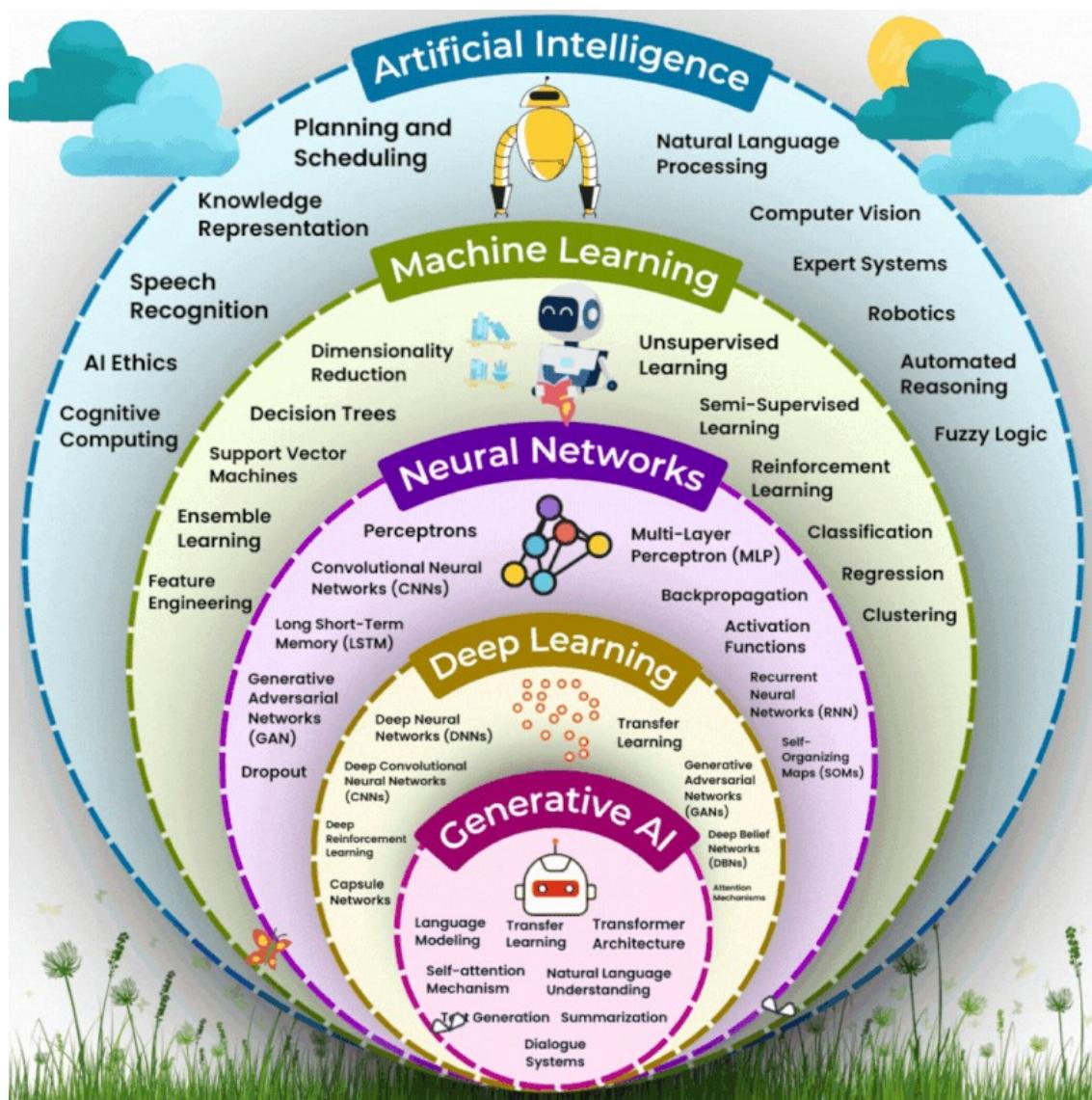


Figura 1 - Universo de Inteligência Artificial.

Fonte: Adaptado de [1].

REFERÊNCIA:

- [1] RONI, A. R. The AI Universe, Cool Tech Gadgets Newsletter, 2024. Disponível em <https://cooltech99.com/>. Acessado em Junho 27, 2024.

3. COMO A INTELIGÊNCIA ARTIFICIAL FUNCIONA

Editado por: Dr. Arnaldo de Carvalho Junior, setembro 25, 2024.

1. FUNCIONAMENTO DA INTELIGÊNCIA ARTIFICIAL (IA) EM POCAS PALAVRAS

A Figura 1 apresenta a estrutura da IA.

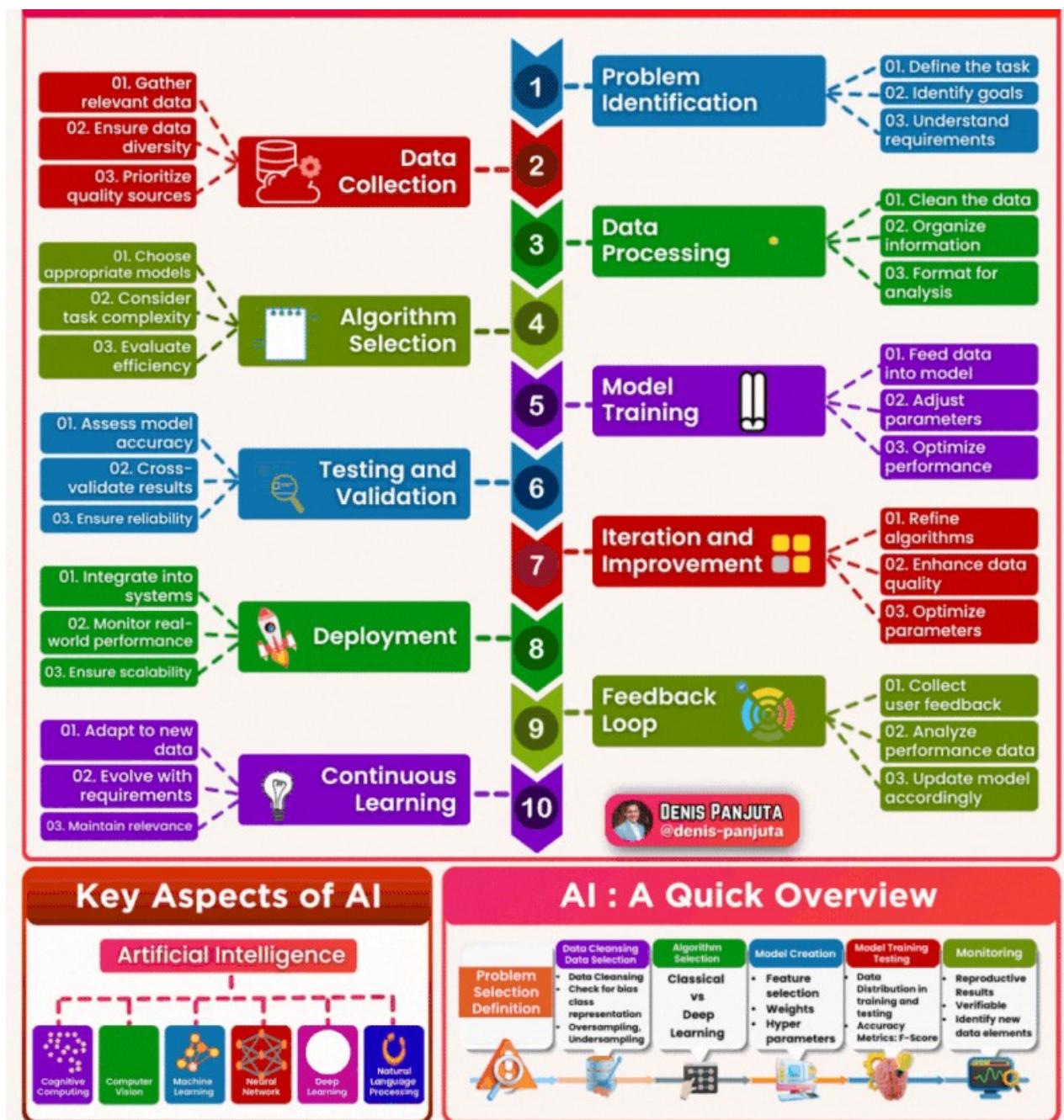


Figura 1 – Estrutura da Inteligência Artificial
Fonte: Adaptado de [1].

1.1. Aparelhamento estrutural

- a) Etapa 1: Identificação do problema;
- b) Etapa 2: Coleta de dados;
- c) Etapa 3: Processamento de dados;
- d) Etapa 4: Seleção de algoritmo;
- e) Etapa 5: Treinamento de modelo;
- f) Etapa 6: Teste e validação;
- g) Etapa 7: iteração e melhoria;
- h) Etapa 8: implantação;
- i) Etapa 9: *Laço (Loop)* de realimentação (*feedback*);
- j) Etapa 10: aprendizado contínuo.

1.2. Aspectos-chave da IA

- a) Computação cognitiva;
- b) Visão computacional;
- c) Aprendizado de máquina;
- d) Rede neural;
- e) Aprendizagem profunda;
- f) Processamento de linguagem natural.

2. IA EM POCAS PALAVRAS

A Figura 2 resume a IA em poucas palavras.

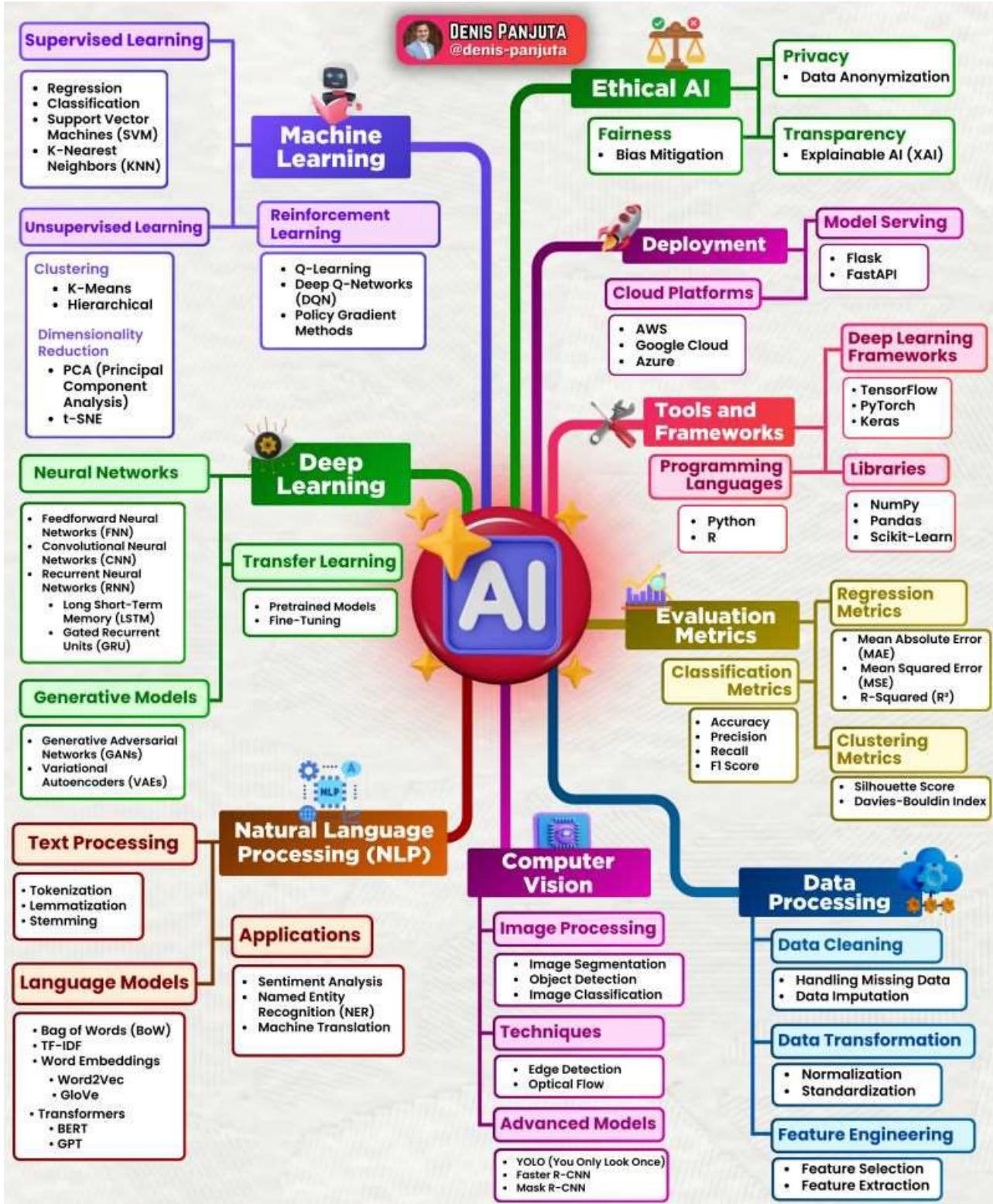


Figura 2 – Estrutura da Inteligência Artificial

Fonte: Adaptado de [2].

- **Aprendizado de máquina (*machine learning* - ML):** envolve técnicas de aprendizado supervisionado, sem supervisão e reforço para permitir que as máquinas aprendam com os dados.
- **Aprendizagem profunda (*deep learning* – DL):** usa redes neurais de várias camadas para representação avançada de dados.
- **Processamento de linguagem natural (*natural language processing* – NLP):** permite que as máquinas compreendam e processem a linguagem humana.
- **Visão computacional:** permite que as máquinas interpretem e analisem informações visuais.
- **Processamento de dados:** prepara dados por meio de limpeza, transformação e engenharia de recursos.
- **IA ética:** concentra-se na justiça, transparência e privacidade no desenvolvimento da IA.
- **Implantação:** utiliza plataformas, ferramentas e estruturas em nuvem para implementar soluções de IA.

3. COMO A IA APRENDE?

Uma pergunta interessante pode ser, como a IA aprende?

Uma analogia fácil usando o processo de aprendizado da escola está na Figura 3.

- IA como estudante:** a IA aprende absorvendo informações, como um aluno na escola. Ele processa dados para obter conhecimento e melhorar com o tempo.
- Dados como livros didáticos (*textbooks*):** Assim como os alunos usam livros didáticos para coletar informações, a IA usa grandes conjuntos de dados para aprender e derivar informações.
- Algoritmos como professores:** no processo de aprendizado da IA, os algoritmos desempenham o papel dos professores, orientando como a IA processa e entende as informações.
- Treinamento como lição de casa (*homework*):** a IA pratica através do treinamento, semelhante aos estudantes fazendo lição de casa. Isso reforça o que a IA aprendeu com os dados.

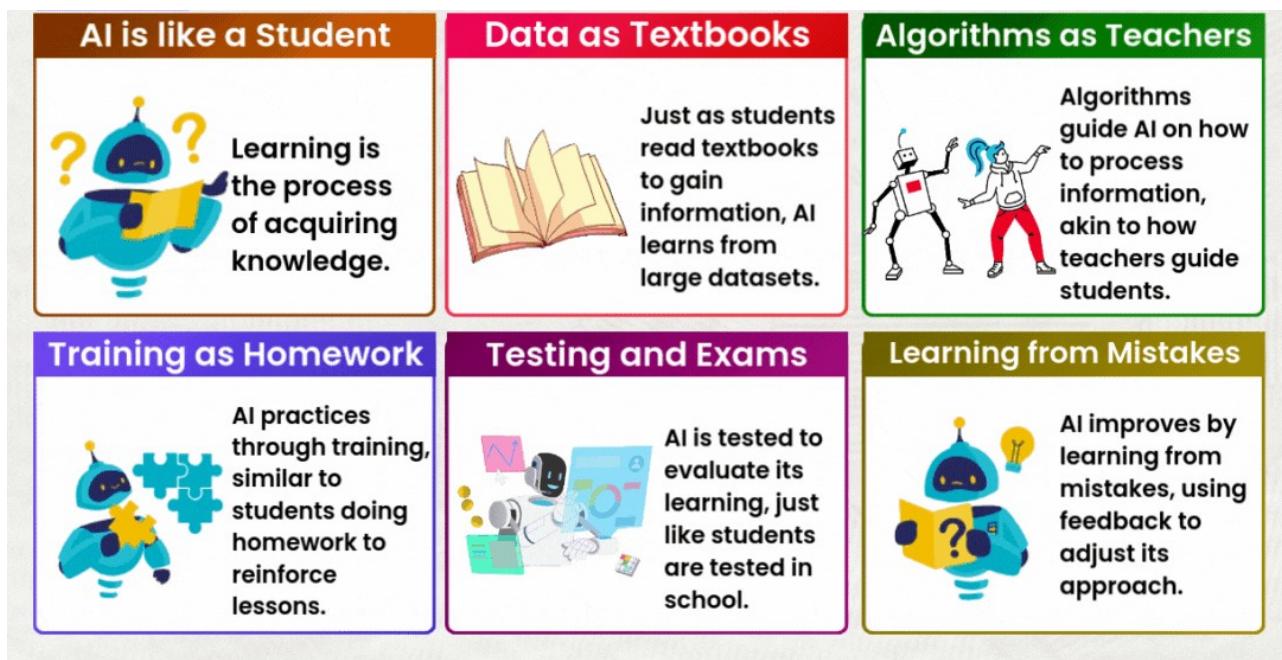
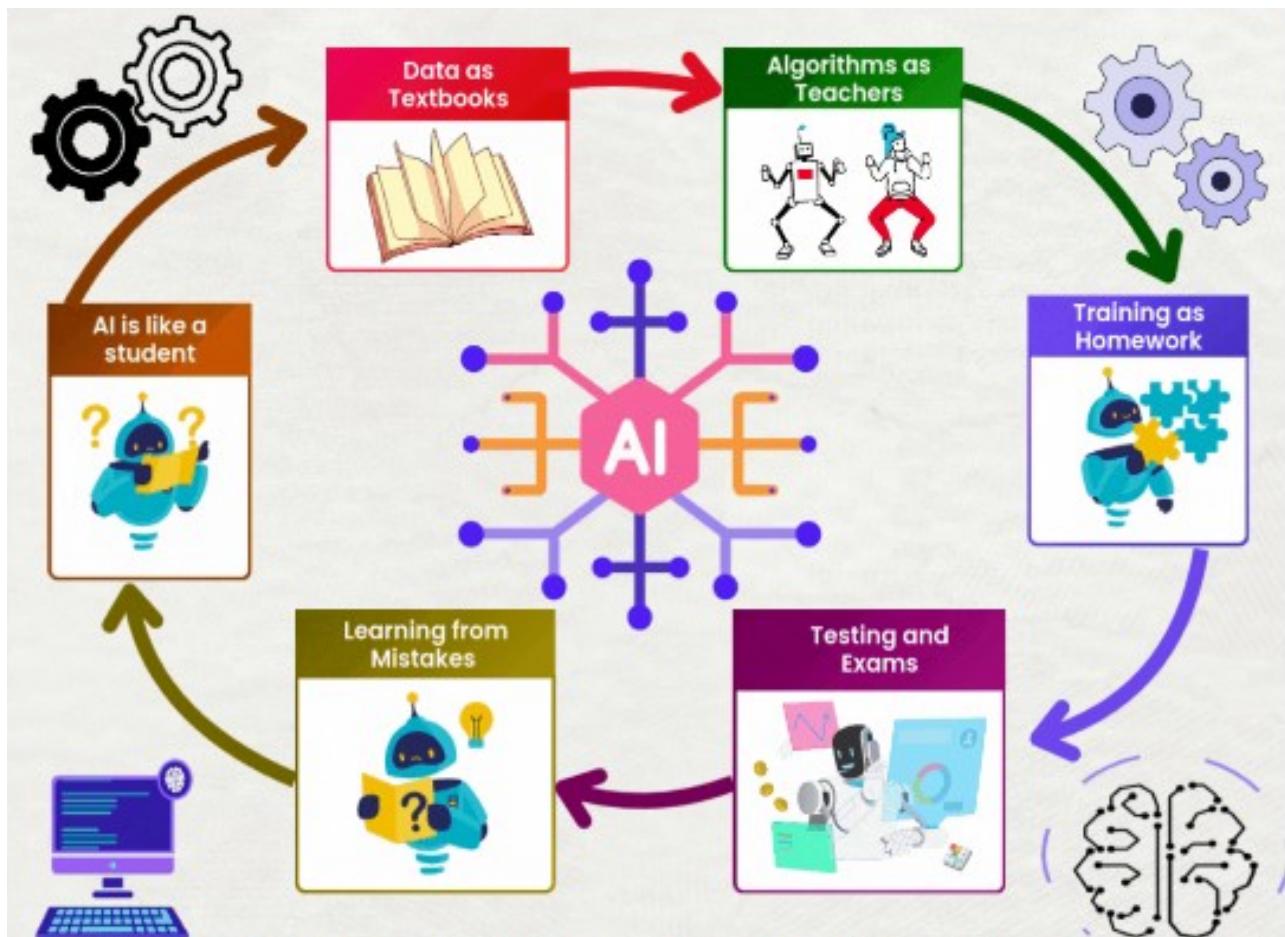


Figura 3 – Analogia de como a IA aprende.

Fonte: Adaptado de [3].

e) Testes e exames: a IA é testada para avaliar seu aprendizado, semelhante aos alunos que estão sendo testados na escola. Isso ajuda a avaliar o quanto bem a IA pode aplicar seu conhecimento.

f) Aprendendo com os erros: a IA melhora continuamente aprendendo com os erros. Usando *feedback*, ele ajusta suas abordagens, aumentando sua precisão e eficácia ao longo do tempo.

Essa analogia pode ser muito útil para compreender as etapas do processo de IA e aprendizado de máquinas.

REFERÊNCIAS

[1] PANJUTA, D. How AI Works – Explained in a Nutshell. Machine Learning Community (Moderated), 2024. Disponível em

[https://media.linkedin.com/dms/image/D4E22AQEuJWIw2jLCA/feedshare-shrink_800/0/1720437690112?](https://media.linkedin.com/dms/image/D4E22AQEuJWIw2jLCA/feedshare-shrink_800/0/1720437690112?e=1723075200&v=beta&t=jx1P1_5Qn8UEQpGOwRbDdgx4f-feNabM5jUvy_K85a4)

Acessado em Junho 27, 2024.

[2] PANJUTA, D. AI in a Nutshell. Machine Learning Community (Moderated), 2024.

Disponível em:

[https://media.linkedin.com/dms/image/v2/D4E22AQEN9C_Pn7ITZQ/feedshare-shrink_800/feedshare-shrink_800/0/1722857227031?](https://media.linkedin.com/dms/image/v2/D4E22AQEN9C_Pn7ITZQ/feedshare-shrink_800/feedshare-shrink_800/0/1722857227031?e=1726099200&v=beta&t=d9pctPRbwZ5nO8gQuQDcU7tkc6Hyw8fAxuxCeysiulI)

Acessado em Agosto 08, 2024.

[3] PANJUTA, D. How does AI learn?. AI Simplified (Linkedin Profile), 2024.

Disponível em:

https://media.linkedin.com/dms/image/v2/D5622AQFYBDb8xqZtWw/feedshare-shrink_800/feedshare-shrink_800/0/1726647167904?e=1730332800&v=beta&t=t-p-1f6sUF_YUQ_Gc89jiPaql65AFUWhHZ-iTOjkt-s

Acessado em Setembro 25, 2024.

4. TUDO PARA APRENDER SOBRE IA EM UM QUADRO

Por: Denis Panjuta.

Editado por: Dr. Arnaldo de Carvalho Junior, Junho 04, 2024.

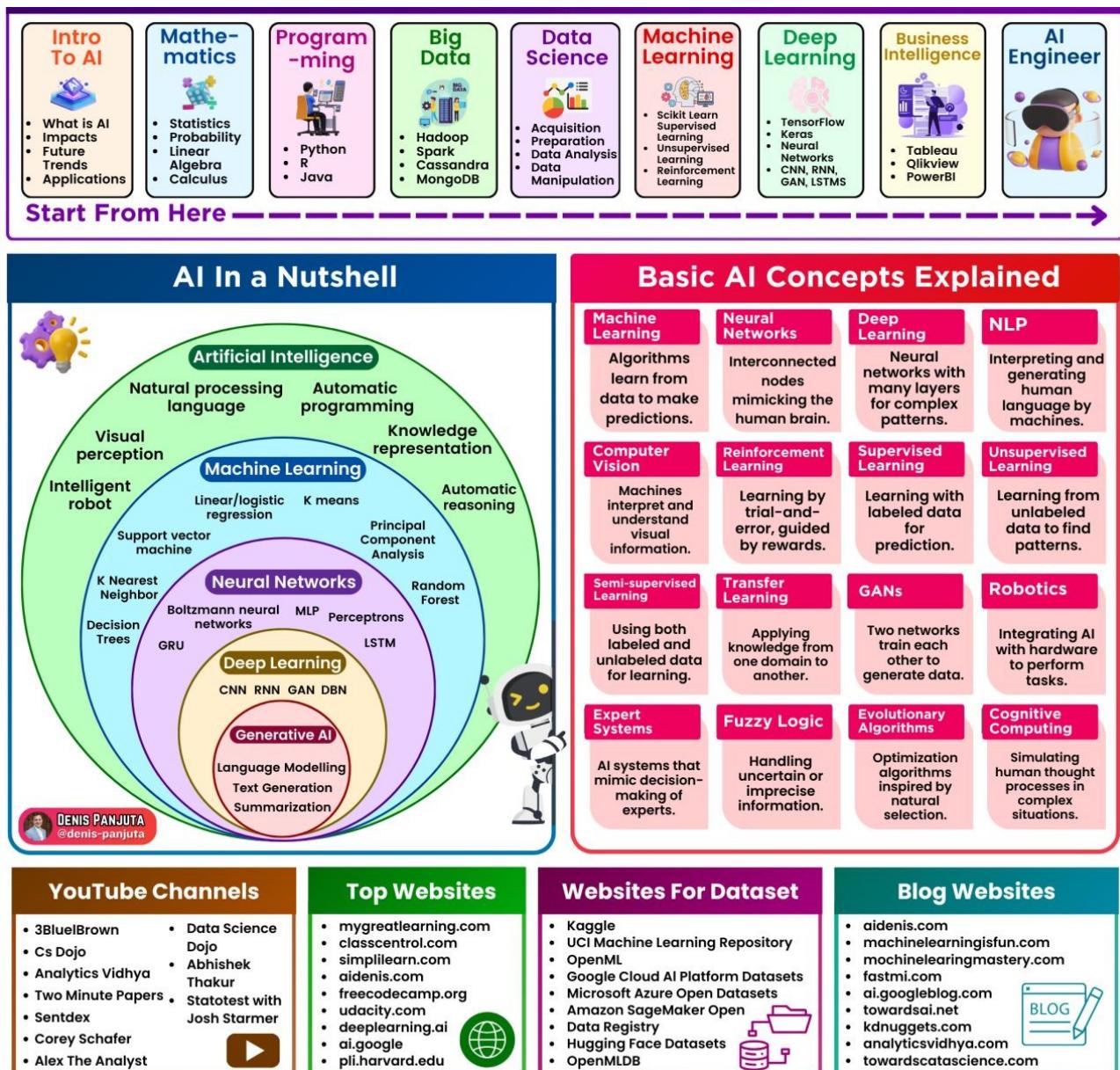


Figura 1 - Tudo para se aprender sobre IA em Um Quadro.

Fonte: Adaptado de [1].

Folha de dicas para aprender IA

1. Roteiro básico para aprender IA

- Introdução a IA;
- Matemática;
- Programação;

- Grande conjunto de dados (*big data*);
- Ciência dos dados;
- Aprendizado de máquina (*machine learning* – ML);
- Aprendizagem profunda (*deep learning* – DL);
- Inteligência de negócios;
- Inteligência Artificial (IA) em poucas palavras;
- Componentes de IA, ML, redes neurais, aprendizado profundo e IA gerativa.

A Figura 2 apresenta o roteiro de estudos de ML.

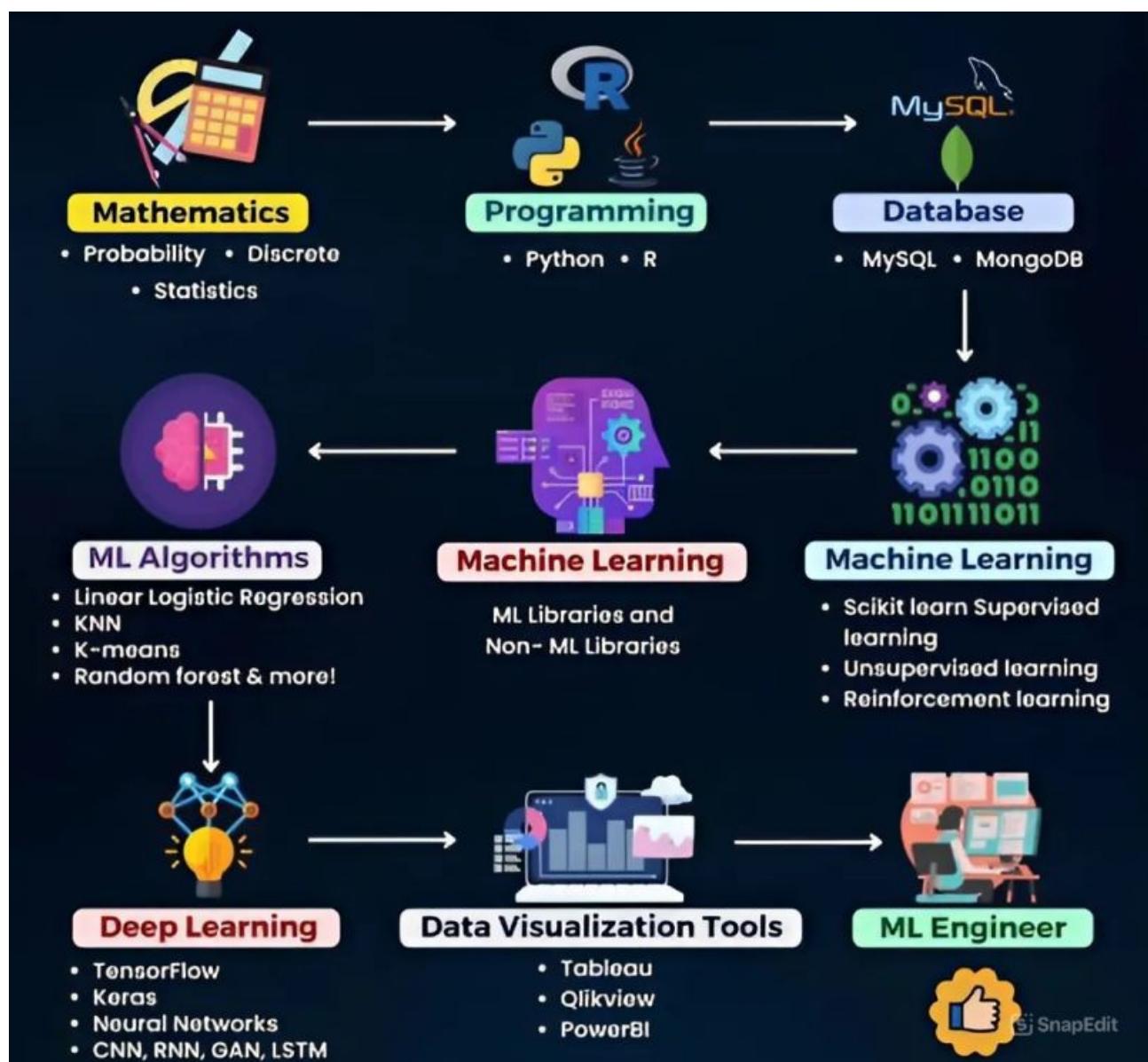


Figura 2 – Roteiro para estudar ML.

Fonte: Adaptado de [2];

2. Conceitos básicos de IA explicados

- Transferir aprendizado;
- Aprendizagem não supervisionada;
- Aprendizagem semi-supervisionada;
- Aprendizagem de reforço;
- Aprendizagem supervisionada;
- Visão computacional;
- Redes neurais;
- Programação de linguagem natural (*natural language programming* – NLP);
- Aprendizado de máquina;
- Aprendizagem profunda.

3. Outros Recursos

- Canais do YouTube;
- Sites para o conjunto de dados;
- Sites de blog;
- Sites principais

REFERÊNCIA

[1] PANJUTA, D. All in One Cheat Sheet to Learn AI, Machine Learning Community (Moderate), 2024. Disponível em:

https://media.linkedin.com/dms/image/D5622AQEwVCp0bNsWzQ/feedshare-shrink_800/0/1717666928207?e=1723075200&v=beta&t=6hlshWdJhnJWa0Ldu24bYruC8WvemqAbZ_OUQIFO6FY.
Acessado em Junho 27, 2024.

[2] RAHUL, P. Roadmap to Learn Machine Learning, Artificial Intelligence, Machine Learning, Generative AI, Python (Linkedin Group), 2024. Disponível em:

https://media.linkedin.com/dms/image/v2/D5622AQEuS0pa4IRSXA/feedshare-shrink_800/B56ZOvDFAZG8Ag-/0/1733808642413?e=1736985600&v=beta&t=ef0HngbWysbILJdCKF_Nlm_oOYD07cFG2OI-Vyg8Cc0.
Acessado em Dezembro 11, 2024.

5. INTELIGÊNCIA ARTIFICIAL EXPLICADA

Editado por: Dr. Arnaldo de Carvalho Junior, Agosto 26, 2024.

1. IA EXPLICADA

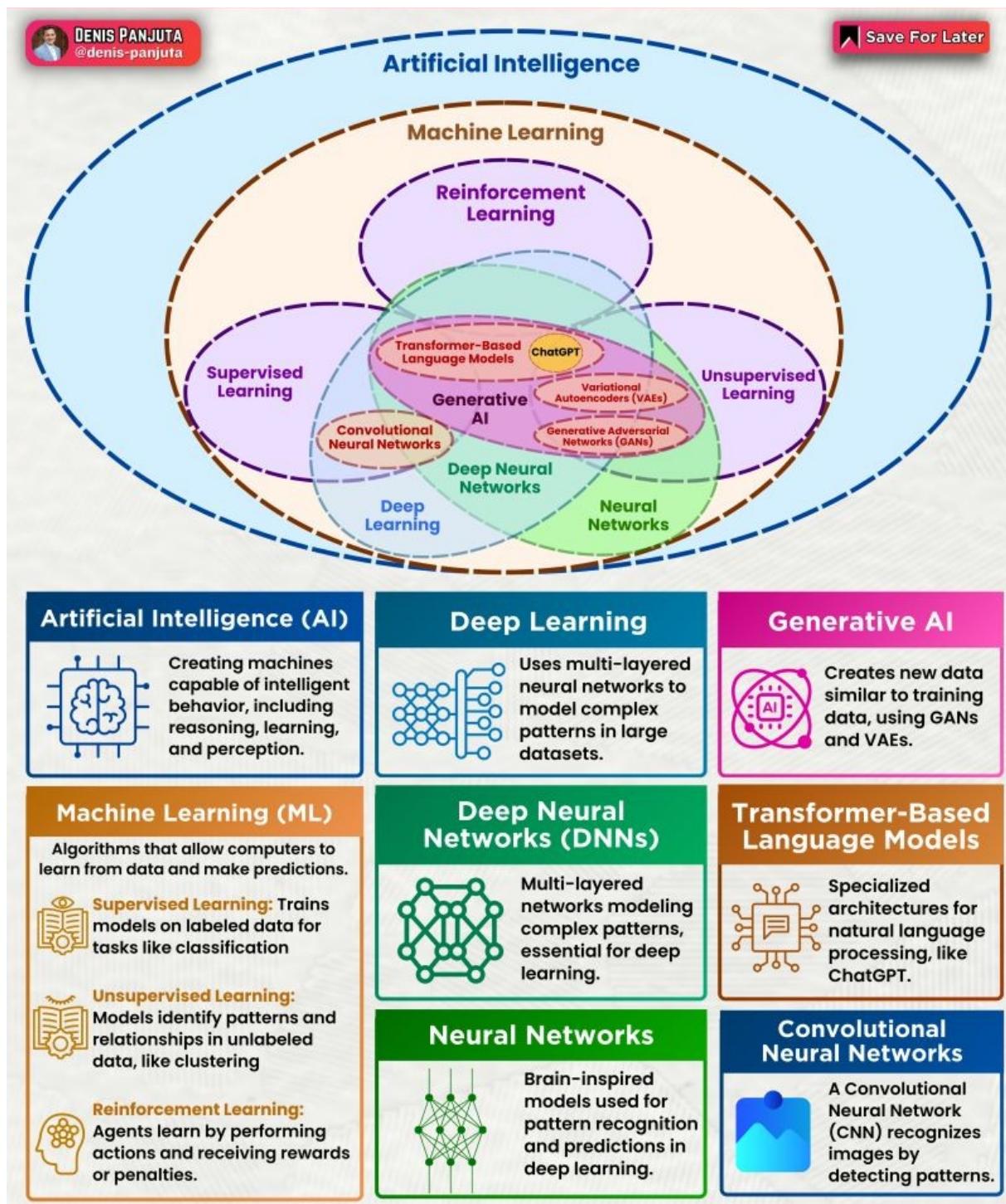


Figura 1 - Inteligência Artificial Explícada.

Fonte: Adaptado de [1].

A Figura 1 apresenta os principais conceitos sobre inteligência artificial (IA). Esses conceitos estão descritos a seguir:

- 1. Inteligência Artificial (IA):** Cria máquinas capazes de comportamentos inteligentes, incluindo raciocínio, aprendizado e percepção.
- 2. Aprendizado de máquina (*machine learning* - ML):** algoritmos que permitem aos computadores aprender com dados e fazer previsões.
 - Aprendizagem supervisionada: treina modelos em dados rotulados para tarefas como classificação.
 - Aprendizagem não supervisionada: os modelos identificam padrões e relacionamentos em dados não marcados, como o agrupamento (*clustering*).
 - Aprendizagem de reforço: os agentes aprendem executando ações e recebendo recompensas ou multas.
- 3. Aprendizagem profunda (*deep learning* - DL):** utiliza redes neurais de várias camadas para modelar padrões complexos em grandes conjuntos de dados.
- 4. Redes neurais (*artificial neural networks* - ANN):** modelos de inspiração cerebral usados para reconhecimento de padrões e previsões no aprendizado profundo.
- 5. Redes Neurais Profundas (*deep neural networks* - DNNs):** redes de várias camadas essenciais para aprendizado profundo, modelando padrões complexos.
- 6. Redes Neurais Convolucionais (*convolutional neural networks* - CNNs):** reconhece imagens detectando padrões, cruciais para tarefas de reconhecimento visual.
- 7. IA Generativa:** cria novos dados semelhantes aos dados de treinamento, usando redes adversárias generativas (*generative adversarial network* - GAN) e autocodificadores variacionais (*variational autoencoder* - VAE).
- 8. Modelos de Linguagens Baseados em Transformadores (*Transformers*):** arquiteturas especializadas para processamento de linguagem natural, como ChatGPT.

2. DOZE CONCEITOS PARA SE ESPECIALIZAR EM IA

A Figura 2 apresenta os 12 principais conceitos em inteligência artificial – IA (*artificial intelligence* - AI) e os respectivos tópicos a serem estudados em profundidade por um especialista em IA.

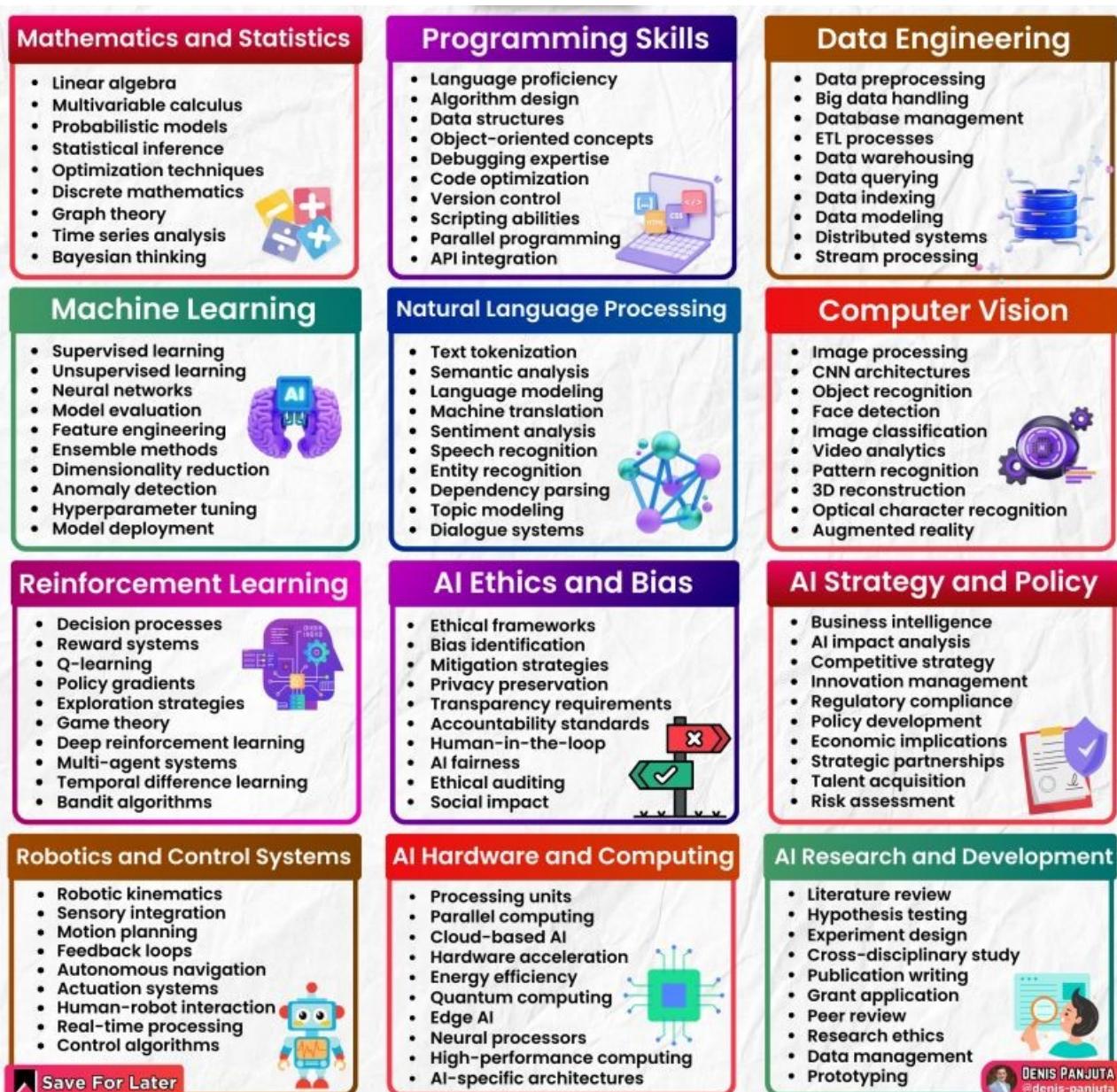


Figura 2 – 12 conceitos principais para se especializar em IA.

Fonte: Adaptado de [2]

Os 12 principais conceitos de IA cujos estudos precisam ser aprofundados são:

- Matemática e Estatística
- Habilidades de programação
- Engenharia de dados
- Visão computacional
- Aprendizado de máquina (ML)
- Processamento de linguagem natural (*natural language processing* - NLP)
- Aprendizagem de reforço

- Ética e preconceito em Inteligência Artificial (IA)
- Estratégia e política de IA
- Sistemas de robótica e controle
- Hardware IA e computação
- Pesquisa e desenvolvimento da IA

3. ROTEIRO PARA DOMÍNIO EM IA

As etapas do roteiro para domínio de IA são descritas a seguir.

- a) Linguagens de programação: Dominar Python e R para iniciar a jornada de Inteligência Artificial (IA).
- b) Matemática e Estatística: Construir uma base forte em álgebra linear, probabilidade, estatística e cálculo.
- c) Manuseio de dados: Aprender técnicas de pré-processamento, limpeza e visualização de dados para trabalhar com eficiência com dados.
- d) Aprendizado de máquina: mergulhar no aprendizado supervisionado e sem supervisão e explorar o aprendizado de reforço.
- e) Aprendizagem profunda: Entender os conceitos de redes neurais artificiais (*artificial neural networks* - ANNs), redes neurais convolucionais (*convolutional neural networks* - CNNs), redes neurais recorrentes (*recurrent neural networks* - RNNs) e redes adversárias generativas (*generative adversarial networks* – GANs).
- f) NLP: Obter experiência em pré-processamento de texto, análise de sentimentos e modelos de idiomas como representação bidirecionais de codificadores de transformadores (*bidirectional encoder representations from transformers* - BERT) e transformador gerativo pré-treinado (*generative pre-trained transformer* - GPT).
- g) Visão computacional: processamento de imagem mestre, detecção de objetos e segmentação de imagens.
- h) Tecnologias de Grandes Conjuntos de Dados (*Big Data*): Familiarizar-se com Hadoop e Spark para lidar com grandes conjuntos de dados.
- i) Estrutura de inteligência artificial (*AI Frameworks*) e bibliotecas: Obter experiência prática com Tensorflow, Pytorch e Scikit-Learn.
- j) Implantação do modelo: Aprender a implantar modelos usando o Flask/Django, Docker para contêiner e Kubernetes para orquestração.

- k) As plataformas em nuvem exploram a AWS, o Azure e o Google Cloud para soluções de IA escaláveis.
- l) Ética em IA: Entender a importância de preconceitos e justiça e privacidade na IA.
- m) Os sistemas de controle de versão usam Git para colaboração eficaz e controle de versão.
- n) O rastreamento de experimento (*experiment tracking*) rastreia os experimentos com MLFlow e pesos e bias.
- o) Operações de aprendizado de máquina (*machine learning operations* – MLOps): Implementar a integração contínua – entrega contínua (IC/EC) (*continuous integration – continuous delivery* - IC/CD) para ML e garantir monitoramento e manutenção robustos do modelo.
- p) IA em produção: Aprender sobre a inferência em tempo real e a escala de modelos de IA para ambientes de produção.

A Figura 3 apresenta o roteiro de domínio em IA.

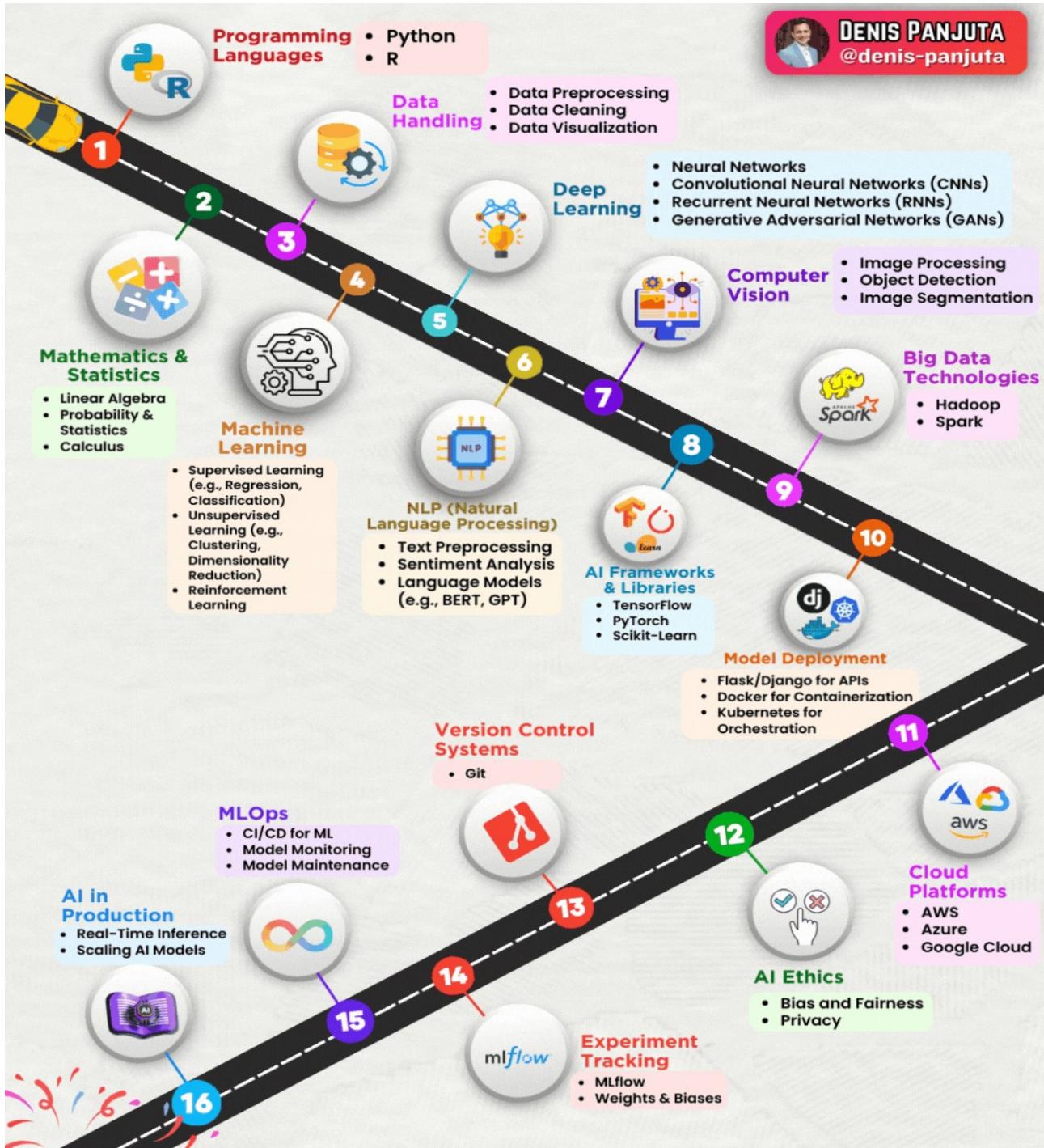


Figura 3 – Roteiro de Domínio de IA

Fonte: Adaptado de [3]

4. ROTEIRO DO ENGENHEIRO DE IA E ML

O roteiro do Engenheiro de IA e ML é descrito a seguir.

1. Começar a jornada de IA/ML com uma base sólida em matemática e estatística.
2. Aprender as principais linguagens de programação, como Python e ferramentas essenciais, como o Jupyter Notebooks.
3. Coletar dados mestre e de pré-processamento para gerenciamento robusto de dados.
4. Explorar a análise de dados exploratórios (*exploratory data analysis* - EDA) para visualização de dados perspicaz e análise estatística.
5. Entender os algoritmos de aprendizado de máquina (ML), incluindo métodos de aprendizagem supervisionados e não supervisionados.
6. Avançar com técnicas de ML, como ajuste de hiperparâmetro e métodos de conjunto.
7. Aprofundar no aprendizado profundo, com foco em redes e estruturas neurais como Tensorflow e Pytorch.
8. Se especializar em processamento de linguagem natural (PNL), cobrindo os modelos de pré-processamento de texto e PNL.
9. Adquirir habilidades em tecnologias de *big-data* como Hadoop e Spark para processamento de dados em larga escala.
10. Aprender a implantação e o serviço de modelos usando ferramentas como Flask, FASTAPI e plataformas em nuvem.
11. Adquirir experiência em MLOPs para a integração contínua e implantação contínua (*continuous integration / continuous deployment* - CI/CD) e o monitoramento de modelos.
12. Completar o conhecimento com habilidades de engenharia de dados, com foco em pipelines de dados e armazenamento.

A Figura 4 apresenta um roteiro do engenheiro de IA/ML proposto.

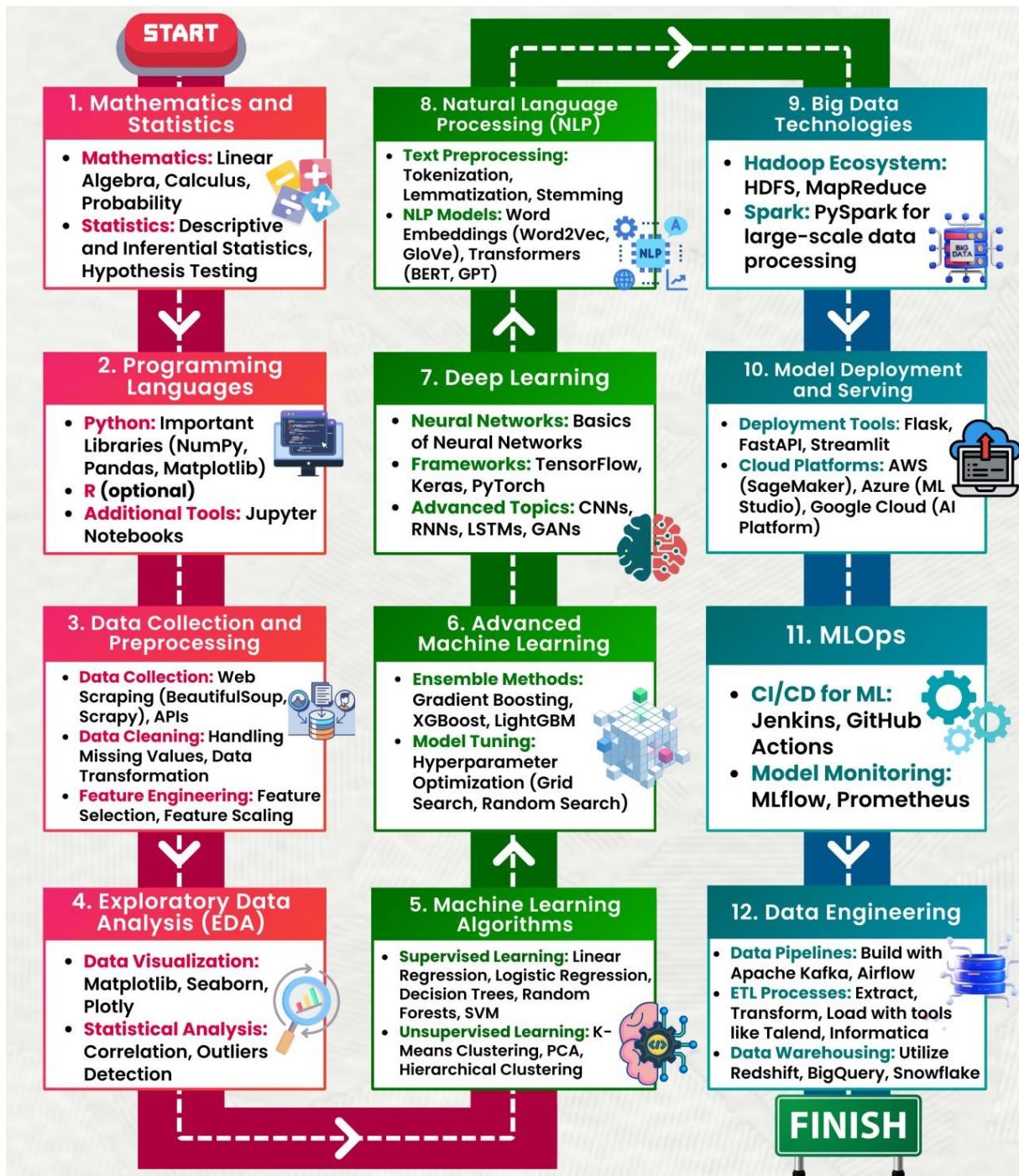


Figura 4 – Roteiro do Engenheiro de IA e ML.

Fonte: Adaptado de [4]

5. FALSOS CONCEITOS SOBRE IA

A Figura 5 apresenta vinte e duas mentiras sobre IA. As principais são explicadas a seguir:



Figura 5 – Roteiro do Engenheiro de IA e ML.

Fonte: Adaptado de [5]

- A IA pensa como os seres humanos? Realidade: a IA opera puramente em algoritmos e dados sem consciência humana.
- A IA assumirá todos os empregos? Realidade: Enquanto a IA muda de paisagens empregadas, ela também cria novos papéis e oportunidades.

- c) A IA é completamente imparcial? Realidade: os sistemas de IA podem perpetuar vieses presentes em seus dados, desafiando a noção de imparcialidade.
- d) A IA entende tudo perfeitamente? Realidade: a IA geralmente luta com nuances e compreensão contextual, ao contrário da crença de que comprehende como seres humanos.
- e) A IA nunca comete erros? Realidade: como qualquer sistema, a IA pode cometer erros, especialmente ao encontrar cenários inesperados.
- f) A IA é sempre objetiva? Realidade: a objetividade da IA está ligada aos vieses em sua programação e aos dados que recebe.
- g) Todas as inovações da IA são benéficas? Realidade: nem todas as inovações de IA são benéficas; Seus impactos podem variar amplamente.
- h) A IA opera por conta própria uma vez implantada? Realidade: os sistemas de IA requerem atualizações e manutenção regulares, desmembrando o mito que operam independentemente uma vez implantadas.
- i) A IA é sempre justa? Realidade: a justiça da IA depende do design cuidadoso do algoritmo para evitar a discriminação.
- j) Os chatbots realmente possuem empatia? Realidade: os chatbots simulam a empatia, mas não têm capacidade emocional genuína.

REFERÊNCIAS

- [1] PANJUTA, D. AI Explained. Machine Learning Community (Moderated), 2024. Disponível em [https://media.licdn.com/dms/image/D4E22AQH5ee-aWrz3Pg/feedshare-shrink_800/0/1721121338485?
e=1724284800&v=beta&t=b5v2HvC_2b7gEaxc9P7T2eFHc2jL6fUbl7IL5ToMtWc](https://media.licdn.com/dms/image/D4E22AQH5ee-aWrz3Pg/feedshare-shrink_800/0/1721121338485?e=1724284800&v=beta&t=b5v2HvC_2b7gEaxc9P7T2eFHc2jL6fUbl7IL5ToMtWc). Acessado em Julho 16, 2024.
- [2] PANJUTA, D. Top 12 Concepts You Must Know to Master AI. Artificial Intelligence, Machine Learning, Data Science & Robotics (Linkedin Group), 2024. Disponível em: [https://media.licdn.com/dms/image/D5622AQH36fgHmA7jMw/feedshare-shrink_1280/0/1721916348937?
e=1724889600&v=beta&t=xBzNPTBi7YSAUjV_hlicvbtyMRqNdBUv0li764u2-SM](https://media.licdn.com/dms/image/D5622AQH36fgHmA7jMw/feedshare-shrink_1280/0/1721916348937?e=1724889600&v=beta&t=xBzNPTBi7YSAUjV_hlicvbtyMRqNdBUv0li764u2-SM). Acessado em Julho 25, 2024.
- [3] PANJUTA, D. AI Mastery Roadmap. Machine Learning Community (Linkedin Group), 2024. Disponível em:

https://media.licdn.com/dms/image/D5622AQFsAtvIdOWAhQ/feedshare-shrink_1280/0/1721821716954?e=1724889600&v=beta&t=jFTZrAoslFgu8TXtY6Q3xgN-uPEkpYmlIPM-RHbeOs0. Acessado em Julho 25, 2024.

[4] PANJUTA, D. AI/ML Engineer Roadmap. Machine Learning Community (Linkedin Group), 2024. Disponível em: https://media.licdn.com/dms/image/v2/D4E22AQGIlvg8xRU5A/feedshare-shrink_1280/feedshare-shrink_1280/0/1724669639704?e=1727308800&v=beta&t=32AkjaYyT0CGqNtEJ1h3A6hL7bXmJNu3w91li94Rzg. Acessado em Agosto 26, 2024.

[5] PANJUTA, D. 22 Common Lies About Artificial Intelligence, Artificial Intelligence, Machine Learning, Data Science & Robotics (Linkedin Group), 2024. Disponível em: https://media.licdn.com/dms/image/v2/D4E22AQEw8w2-2zYIAA/feedshare-shrink_800/feedshare-shrink_800/0/1729499863876?e=1732147200&v=beta&t=Od1w4wbvH2UYZiDpjGf_pfOMTUX8BH-tk8uFqgtdxuc. Acessado em Outubro 21, 2024.

6. O ECOSISTEMA DE IA – FERRAMENTAS E CURSOS

Por: Denis Panjuta.

Editado por: Dr. Arnaldo de Carvalho Junior, Junho 04, 2024.

A Figura 1 apresenta o ecossistema de Inteligência Artificial (IA), principais tecnologias e ferramentas chaves [1].

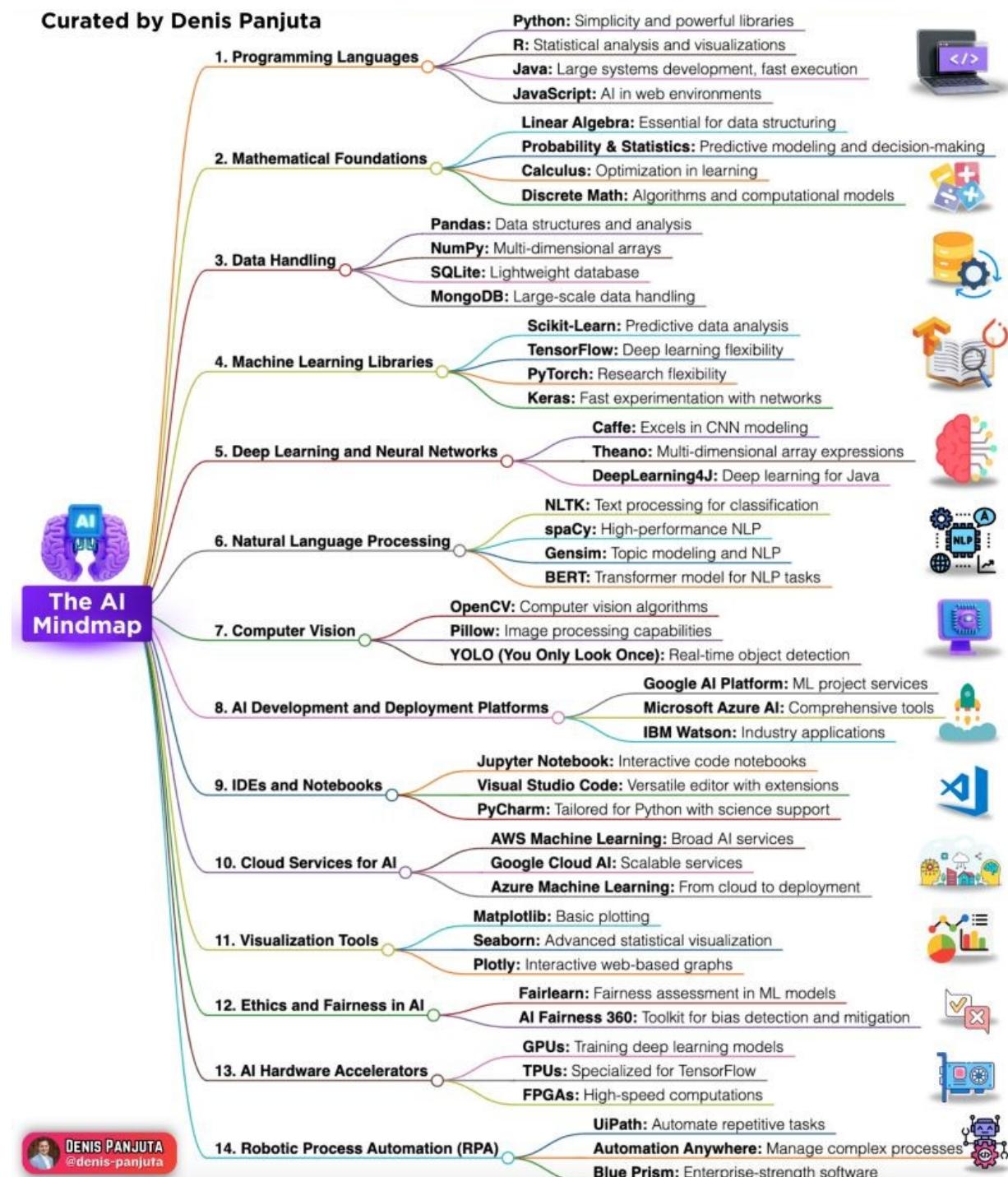


Figura 1 – Ferramentas e Tecnologias Chave para o Ecossistema de IA.

Fonte: Adaptado de [1].

Ecossistema de IA (Cursos Gratuitos Inclusive) [1], [2].

1. Fundamentos de Matemática

a) Álgebra Linear:

1)<https://www.coursera.org/learn/linear-algebra-machine-learning>

2)https://www.youtube.com/watch?v=fNk_zzaMoSs&list=PLZHQBObOWTQDPD3MizzM2xVFitgF8hE_ab

3)<https://www.khanacademy.org/math/linear-algebra>

b) Probabilidade e Estatística:

f) <https://www.coursera.org/learn/machine-learning-probability-and-statistics>

g) <https://www.khanacademy.org/math/statistics-probability>

c) Cálculo: <https://www.coursera.org/learn/machine-learning-calculus>

d) Matemática Discreta: <https://www.youtube.com/watch?v=p2b2Vb-cYCs&list=PLBlnK6fEyqRhqJPDXcvYILfXPh37L89g3>

e) Matemática Imersiva: <https://immersivemath.com/ila/learnmore.html>

2. Linguagem de Programação

a) Python:

g) <https://www.mygreatlearning.com/academy/learn-for-free/courses/artificial-intelligence-with-python>

h) https://www.coursya.com/product/programming-for-everybody-getting-started-with-python/?utm_source=LinkedIn+&utm_campaign=rv

b) R Programming: <https://www.codecademy.com/learn/learn-r>

c) Java: <https://intellipaat.com/academy/course/java-training/>

3. Manuseio de Dados

a) Pandas and Numpy: <https://www.youtube.com/watch?v=GPVsHOIRBBI>

b) SQLite: <https://www.youtube.com/watch?v=byHcYRpMgl4>

c) MongoDB: <https://youtu.be/c2M-rlkkT5o?si=jFcqav4yBZwV3hXz>

4. Redes Neurais e Aprendizado Profundo

a) Aprendizado Profundo (*Deep Learning*): <https://www.youtube.com/watch?v=VyWAvY2CF9c>

b) Visão Computacional: <https://www.youtube.com/watch?v=IA3WxTPXqQ>

- c) Rede Neural Recorrente (*recurrent neural network* – RNN) PyTorch do princípio:
<https://jaketae.github.io/study/pytorch-rnn/>
- d) Entendendo Redes de Memória Longas de Termo Curto (Long Short Term Memory - LSTM): <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>

5. Desenvolvimento e Implementação de IA

- a) Google AI Platforms : <https://www.youtube.com/watch?v=N2WSr7pt54g>
- b) Azure AI Studio: <https://www.youtube.com/watch?v=fQ9RFR1KTbY>
- c) Introdução à IA Generativa: https://www.courtesy.com/product/introduction-to-generative-ai/?utm_source=LinkedIn+&utm_campaign=rv
- d) Fundação da IA para Todos (IBM): https://www.courtesy.com/product/ai-foundations-for-everyone-specialization/?utm_source=LinkedIn+&utm_campaign=rv
- e) Aprendizado de Máquinas (machine learning) para todos:
https://www.courtesy.com/product/machine-learning-for-all/?utm_source=LinkedIn+&utm_campaign=rv

6. Ferramentas de Visualização

- 1. Matplotlib tutorial: <https://www.youtube.com/watch?v=wB9C0Mz9qSo>
- 2. Seaborn tutorial: <https://www.youtube.com/watch?v=6GUZXDef2U0>
- 3. Plotly tutorial: <https://www.youtube.com/watch?v=GGL6U0k8WYA>

7. Bibliotecas de Aprendizado de Maquina (*Machine Learning Libraries*)

- 1. Scikit-Learn: https://www.youtube.com/watch?v=0B5eIE_1vpU
- 2. Tensorflow: <https://www.youtube.com/watch?v=tPYj3fFJGjk>
- 3. Pytorch: https://www.youtube.com/watch?v=V_xro1bcAuA&t=10s

8. Processamento de Linguagem Natural (*Natural Language Processing - NLP*)

- a) RealPython - NLP with spaCy: <https://realpython.com/natural-language-processing-spacy-python/>
- b) Guia NLP Kaggle: <https://www.kaggle.com/learn-guide/natural-language-processing>
- c) Word2vec Ilustrado: <https://jalammar.github.io/illustrated-word2vec/>
- d) Engenharia de Comandos para ChatGPT: https://www.courtesy.com/product/prompt-engineering-for-chatgpt/?utm_source=LinkedIn+&utm_campaign=rv

REFERÊNCIA:

- [1] PANJUTA, D. The AI Ecossistem, Machine Learning Community (Moderate), 2024. Disponível em: https://media.licdn.com/dms/image/D4E22AQENVzMWA-33_Q/feedshare-shrink_800/0/1719101139160?e=1723075200&v=beta&t=3ynaVgluP-vRJ8NKBfNnGnPyoVJTf478kXpqZN9JZVY. Acessado em Junho 27, 2024.
- [2] SINGH, A. Free AI/ML LLM Fundamentals Courses, Linkedin Post, 2024. Disponível em: https://media.licdn.com/dms/image/D4E22AQEKel1JEDXUQ/feedshare-shrink_800/0/1721357172190?e=1725494400&v=beta&t=hls0fLGcJVPkWxc3vXI5aRbnN_6KihCDYF4NDj7uSkM. Acessado em Jul 30, 2024.

7. COMPREENDENDO IA, ML, DI E DS

Por: Denis Panjuta.

Editado por: Dr. Arnaldo de Carvalho Junior, Junho 07, 2024.

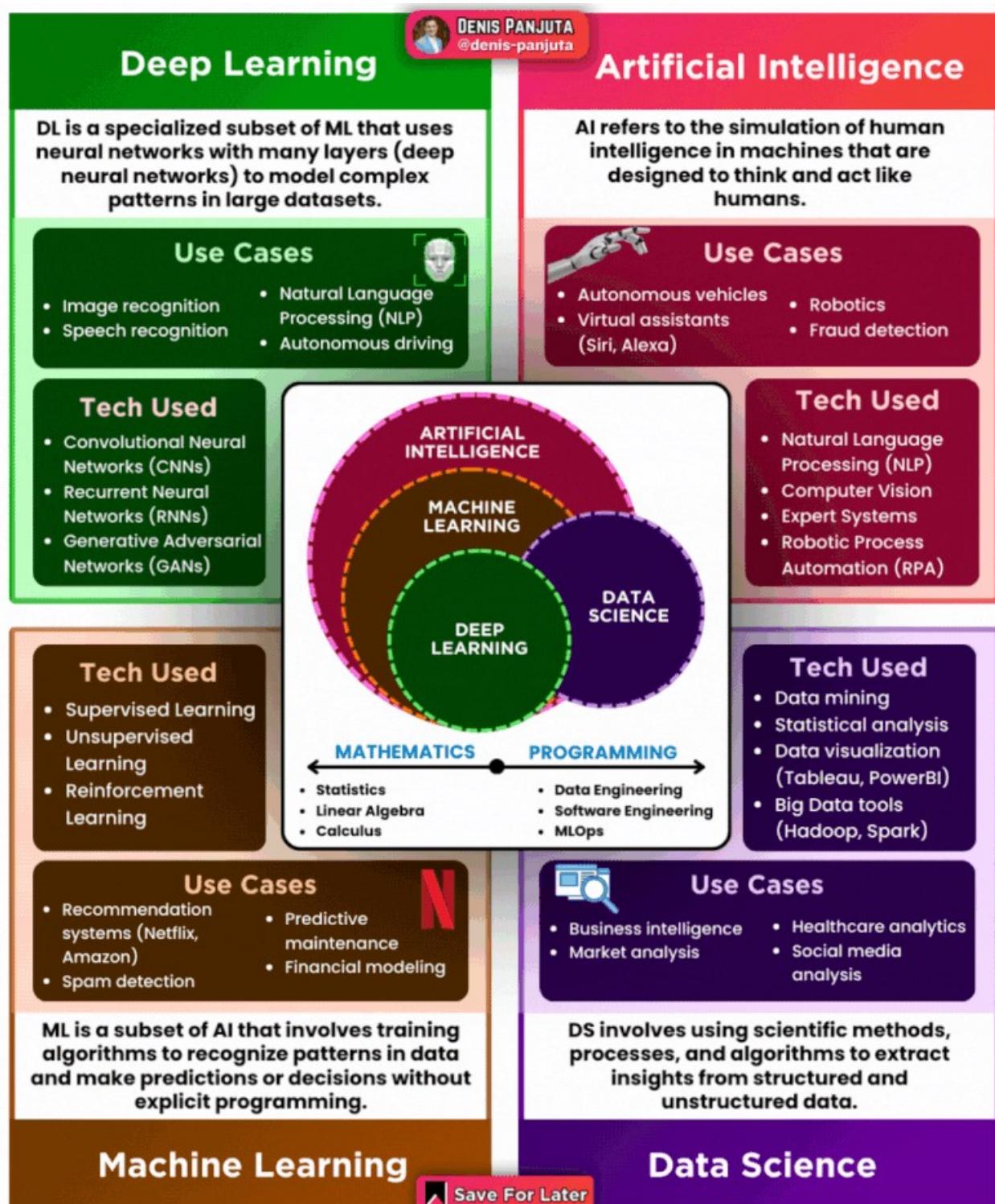


Figura 1 - Discriminação da Inteligência Artificial.

Fonte: Adaptado de [1].

Compreendendo IA, ML, DL e DS

1. **Inteligência Artificial (IA):** Simula a inteligência humana em máquinas projetadas para pensar e agir como humanos, com aplicações em veículos autônomos, assistentes virtuais, robótica e detecção de fraudes.
2. **Aprendizado de Máquina (Machine Learning - ML):** Um subconjunto de IA focado em algoritmos que reconhecem padrões em dados e tomam decisões sem programação explícita, utilizados em sistemas de recomendação, manutenção preditiva e modelagem financeira.
3. **Aprendizado Profundo (Deep Learning - DL):** Um subconjunto especializado de ML que utiliza redes neurais com muitas camadas para modelar padrões complexos em grandes conjuntos de dados, aplicável em reconhecimento de imagem e fala e direção autônoma.
4. **Ciência de dados (Data Science – DS):** Utiliza métodos e algoritmos científicos para extrair insights de dados estruturados e não estruturados, impulsionando inteligência de negócios, análise de mercado, análise de saúde e muito mais.

REFERÊNCIA

- [1] PANJUTA, D. AI Breakdown, Machine Learning Community (Moderate), 2024. Disponível em: https://media.linkedin.com/dms/image/D4E22AQFsk-rUN-ujw/feedshare-shrink_800/0/1719089411056?e=1723075200&v=beta&t=QWygd06KdISSwmUzLuas96H-MIKkyoAMmVVumJ4BdqI. Acessado em Junho 7, 2024.

8. INTELIGÊNCIA ARTIFICIAL x APRENDIZADO DE MÁQUINA X REDE NEURAL ARTIFICIAL X APRENDIZADO PROFUNDO

Por: Arif Alan

Adaptado por: Dr. Arnaldo de Carvalho Junior

Data: Julho 19, 2024.

1. Inteligência Artificial – IA (*Artificial Intelligence – AI*)

A IA é o conceito mais amplo, referindo-se a máquinas projetadas para imitar a inteligência humana. A IA abrange todos os subcampos destinados a permitir que os computadores executem tarefas que normalmente exigem inteligência humana.

Para saber mais: <https://towardsdatascience.com/>. Acessado em Jul 19, 2024.

2. Aprendizado de Máquina (*Machine Learning – ML*)

O ML é um subconjunto de IA focado na ideia de que as máquinas podem aprender com os dados e melhorar seu desempenho ao longo do tempo sem serem explicitamente programados para todas as tarefas.

Conceitos chave:

- Aprendizagem supervisionada: Aprenda com os dados rotulados.
- Aprendizagem não supervisionada: encontre padrões em dados não marcados.
- Aprendizagem de reforço: Aprenda através de tentativa e erro.

Para saber mais: <https://www.coursera.org/learn/machine-learning>. Acessado em Jul 19, 2024.

3. Rede Neural Artificial – RNA (*Artificial Neural Network – ANN*)

As ANNs são um subconjunto de ML inspiradas na estrutura do cérebro humano. Elas consistem em nós interconectados (neurônios) que processam dados em camadas. As redes neurais são a base para muitos algoritmos ML.

Conceitos chave:

- Camadas: camadas de entrada, oculto e saída.
- Funções de ativação: determine a saída de um nó.

Para saber mais: <https://www.tensorflow.org/?hl=pt-br>. Acessado em Jul 19, 2024.

4. Aprendizado Profundo (*Deep Learning* – DL)

O DL é um subconjunto especializado de ML que usa redes neurais profundas com muitas camadas. O DL se destaca em tarefas que envolvem grandes quantidades de dados não estruturados, como imagens, áudio e texto.

Conceitos chave:

- Redes Neurais Convolucionais (*Convolutional Neural Networks* - CNNs): usadas para reconhecimento de imagem.
- Redes Neurais Recorrentes (*Recurrent Neural Networks* - RNNs): usadas para dados sequenciais.

Para saber mais: <https://www.deeplearning.ai/>. Acessado em Jul 19, 2024.

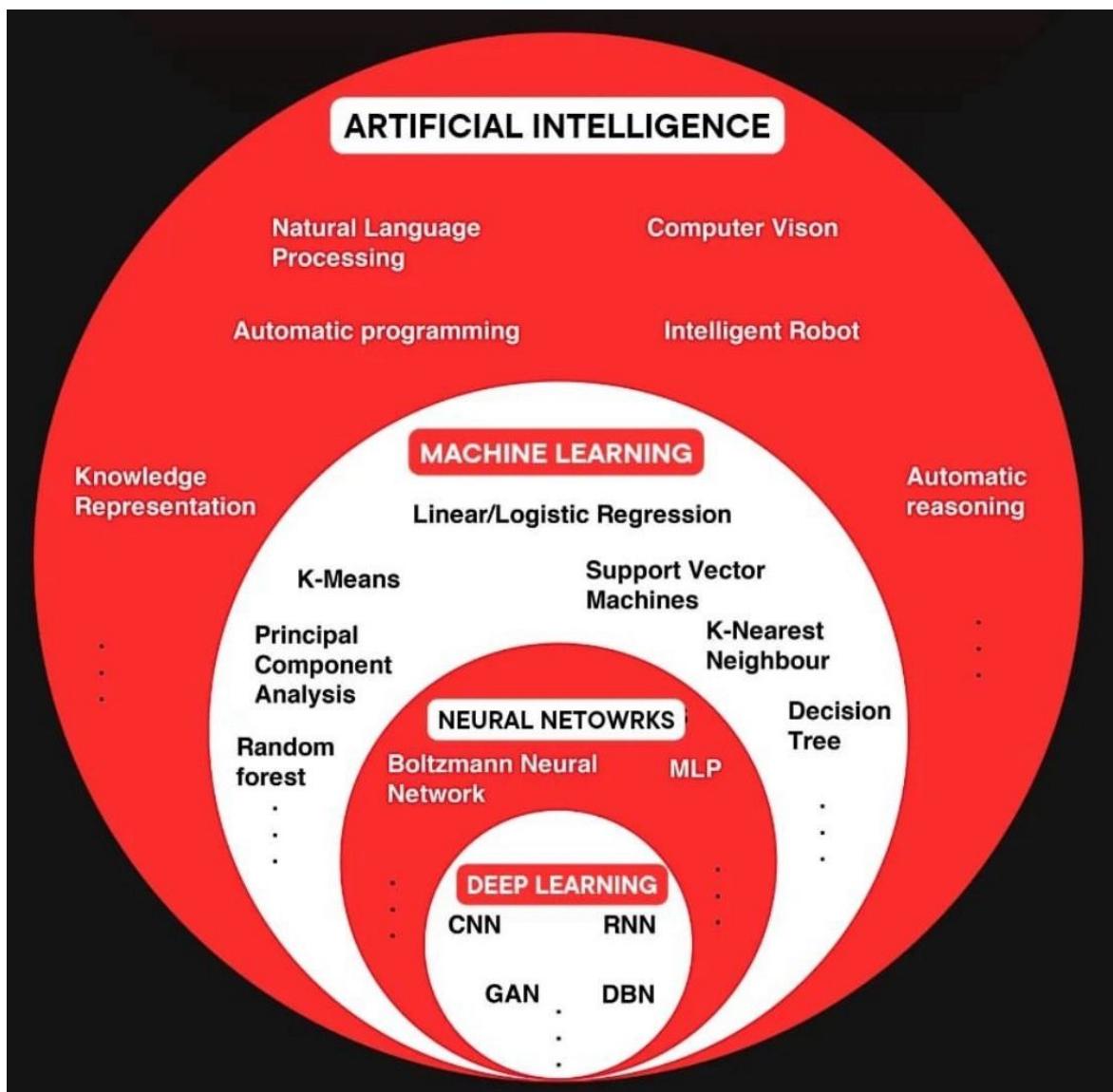


Figura 1 - AI vs ML vs ANN vs DL

Fonte: Adaptado de [1].

5. Comparação

- Inteligência Artificial (IA): O campo geral se concentrou na criação de máquinas inteligentes.
- Aprendizado de Máquina (*Machine Learning* - ML): Um subconjunto de IA que permite que as máquinas aprendam com os dados.
- Rede Neural Artificial (*Artificial Neural Network* - ANN): Um modelo específico usado em ML inspirado no cérebro humano.
- Aprendizado Profundo (*Deep Learning* - DL): Um subconjunto de ML que usa redes neurais complexas com muitas camadas.

REFERÊNCIA

[1] ALAM, A. D. AI/ML AI vs ML vs NN vs DL. Data Science Brain, 2024. Disponível em: https://media.licdn.com/dms/image/D5622AQHG2KmkeNCt6Q/feedshare-shrink_2048_1536/0/1721391498440?e=1724284800&v=beta&t=WGB2kyXJdQjTSIAzM6ElyiRY5Ax1py1GaMfRpObGW0g. Acessado em Julho 19, 2024.

9. DATASETS DE ACESSO LIVRE PARA QUALQUER PROJETO DE IA!

Compilado por: Dr. Arnaldo de Carvalho Junior

Data: 17/06/2024

A melhor forma de aprender o poder da inteligência artificial (IA) é criando projetos e aplicações com ela.

O sucesso de um projeto de IA começa pela análise, seleção, tratamento e filtragem do conjunto de dados a serem utilizados no treinamento da IA. O tempo demandado nesta etapa permitirá economizar tempo nas etapas futuras do projeto, minimizar o uso de recursos computacionais e elevar o nível de acurácia da IA.

A seguir são relacionadas alguns dos conjuntos de dados (*datasets*) de acesso livre para projetos de IA mais difundidos.

- h) UCI Machine Learning Repository - <https://archive.ics.uci.edu/datasets>
- i) Harvard Dataverse (Harvard University) - <https://data.harvard.edu/dataverse>
- j) Data .gov US - <https://data.gov/>
- k) Data .gov UK - <https://www.data.gov.uk/>
- l) European Data - <https://data.europa.eu/data/datasets?locale=en>
- m) Latin American Data Bank - <https://ropercenter.cornell.edu/latin-american-data-bank>
- n) Dados Abertos Brasil - <https://dados.gov.br/signin>
- o) Earth Data (NASA) - <https://www.earthdata.nasa.gov/>
- p) CERN Open Data Portal - <https://opendata.cern.ch/>
- q) Health Data (USA) - <https://healthdata.gov/>
- r) Centers For Disease Control And Prevention (USA) -
<https://www.cdc.gov/datastatistics/index.html>
- s) Dataset for Health Care and Public Health (USA) -
<https://researchguides.dartmouth.edu/c.php?q=517073&p=6289098>
- t) Global Health Observatory Data Repository - World Health Organization (WHO) -
<https://apps.who.int/gho/data/node.home>
- u) National Library of Medicine (NIH – USA) - <https://medpix.nlm.nih.gov/home>
- v) Center of AI in Medicine & Imaging (Stanford University) -
<https://aimi.stanford.edu/shared-datasets>

- w) Labelme (CSAIL – MIT) -
<http://labelme.csail.mit.edu/Release3.0/browserTools/php/dataset.php>
- x) ImageNet - <https://image-net.org/>
- y) Kaggle Datasets - <https://www.kaggle.com/datasets>
- z) Sigma Open Datasets - <https://sigma.ai/open-datasets/>
- aa) OpenML - <https://www.openml.org/>
- bb) Datahub .io - <https://datahub.io/collections>
- cc) Google Dataset Search - <https://datasetsearch.research.google.com/>
- dd) IBM Data Asset eXchange - <https://developer.ibm.com/exchanges/data/>
- ee) AWS Open Data - [https://aws.amazon.com/marketplace/search/results?
trk=868d8747-614e-4d4d-9fb6-
fd5ac02947a8&sc_channel=el&FULFILLMENT_OPTION_TYPE=DATA_EXCHANGE
&CONTRACT_TYPE=OPEN_DATA_LICENSES&filters=FULFILLMENT_OPTION
TYPE%2CCONTRACT_TYPE](https://aws.amazon.com/marketplace/search/results?trk=868d8747-614e-4d4d-9fb6-fd5ac02947a8&sc_channel=el&FULFILLMENT_OPTION_TYPE=DATA_EXCHANGE&CONTRACT_TYPE=OPEN_DATA_LICENSES&filters=FULFILLMENT_OPTION_TYPE%2CCONTRACT_TYPE)
- ff) BD - <https://basedosdados.org/>
- gg) Furnas Dataset (Electrical Power Transmission Lines) -
https://github.com/fredso/PTL-AI_Furnas_Dataset?tab=readme-ov-file
- hh) Antarctic Datasets - <https://www.antarcticglaciers.org/antarctica-2/antarctic-datasets/>
- ii) Nasdaq Data Link - <https://data.nasdaq.com/institutional-investors>
- jj) BFI film industry statistics (UK) - <https://www.bfi.org.uk/industry-data-insights>
- kk) NYC Taxi Trip Data - <https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page>
- ll) FiveThirtyEight - <https://data.fivethirtyeight.com/>
- mm) IMDb Non-Commercial Datasets - <https://developer.imdb.com/non-commercial-datasets/>
- nn) FBI (USA) Crime Data Explorer -
<https://cde.ucr.cjis.gov/LATEST/webapp/#/pages/home>

10. PRINCIPAIS ALGORITMOS DE INTELIGÊNCIA ARTIFICIAL PARA LEIGOS

Editado por: Dr. Arnaldo de Carvalho Junior

Data: 10/10/2024

A Figura 1 apresenta os principais algoritmos de Inteligência Artificial (IA) para leigos.

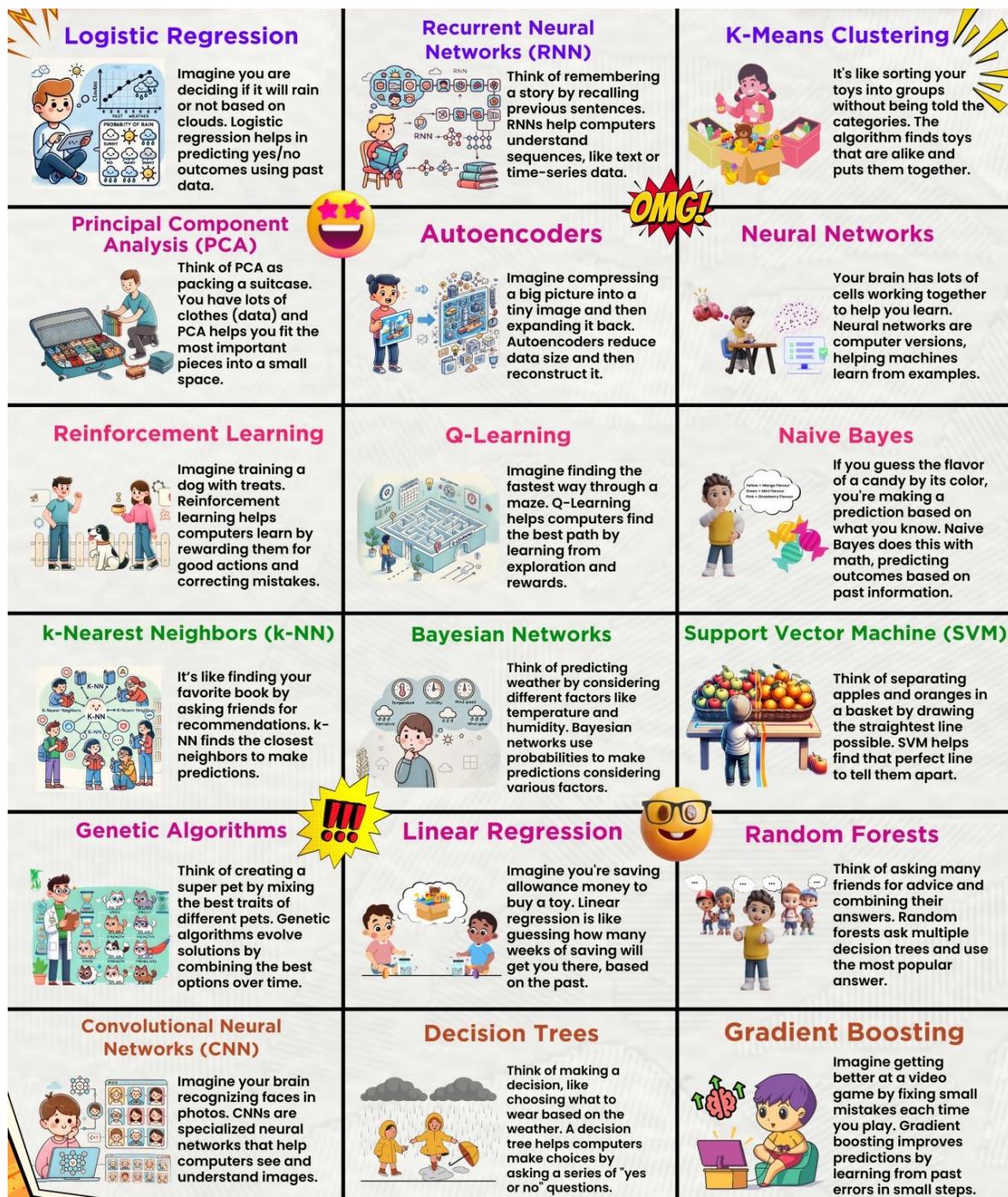


Figura 1 - Principais Algoritmos de Inteligência Artificial Explicados.

Fonte: Adaptado de [1].

- 1. Regressão Logística (*Logistic Regression* - LR):** prevê resultados sim/não, usando dados passados.
- 2. Redes Neurais Recorrentes (*Recurrent Neural Networks* - RNN):** tipo de rede neural que entende sequências como texto e séries temporais de dados.
- 3. Conjunto de Significados K (*K-Means Clustering*):** algoritmo que agrupa itens semelhantes, sem uma definição prévia.
- 4. Análise de Componentes Principais (*Principal Component Analysis* - PCA):** algoritmo que empacota dados importantes em um pequeno espaço.
- 5. Autocodificadores (*AutoEncoders*):** algoritmo que comprime e depois reconstrói imagens.
- 6. Redes Neurais Artificiais (*Artificial Neural Networks* - ANN):** redes de neurônios artificiais baseados no cérebro biológico, que aprende por exemplos.
- 7. Aprendizagem de Reforço (*Reinforcement Learning* - RL):** aprende com recompensas, ou seja, o computador é recompensado por boas ações e erros são corrigidos.
- 8. Aprendizado Q (*Q-Learning*):** encontra o melhor caminho em um labirinto, através da exploração e recompensa.
- 9. Bayes Ingênuos (*Nave Bayes* - NB):** prevê resultados com base no conhecimento prévio.
- 10. Vizinhos Mais Próximos (*K Nearest Neighbors* – K-NN):** encontra vizinhos mais próximos para fazer previsões.
- 11. Redes Bayesianas (*Bayesian Networks* - BN):** utiliza probabilidade para realizar previsão considerando diferentes fatores.
- 12. Máquina Vetorial de Suporte (*Support Vector Machine* - SVM):** separa os itens com a linha mais reta.
- 13. Algoritmos Genéticos (*Genetic Algorithm* - GA):** evoluí na solução ao combinar as melhores opções através do tempo.
- 14. Regressão Linear (*Linear Regression* - LR):** prevê resultados com base em dados anteriores.
- 15. Florestas Aleatórias (*Randon Forest* - RF):** combina várias respostas para precisão.
- 16. Redes Neurais Convolucionais (*Convolutional Neural Networks* - CNN):** são redes neurais especializadas, que permitem computadores ver e entender imagens.

17. Árvores de Decisão (Decision Tree - DT): toma decisões a partir de uma sequência de perguntas do tipo sim ou não.

18. Aumentado de Gradiente (Gradient Boosting - GB): melhora previsões através de erros passados em passos pequenos.

Aprendizado de máquina desmisticificado: abordagens supervisionadas versus não supervisionadas

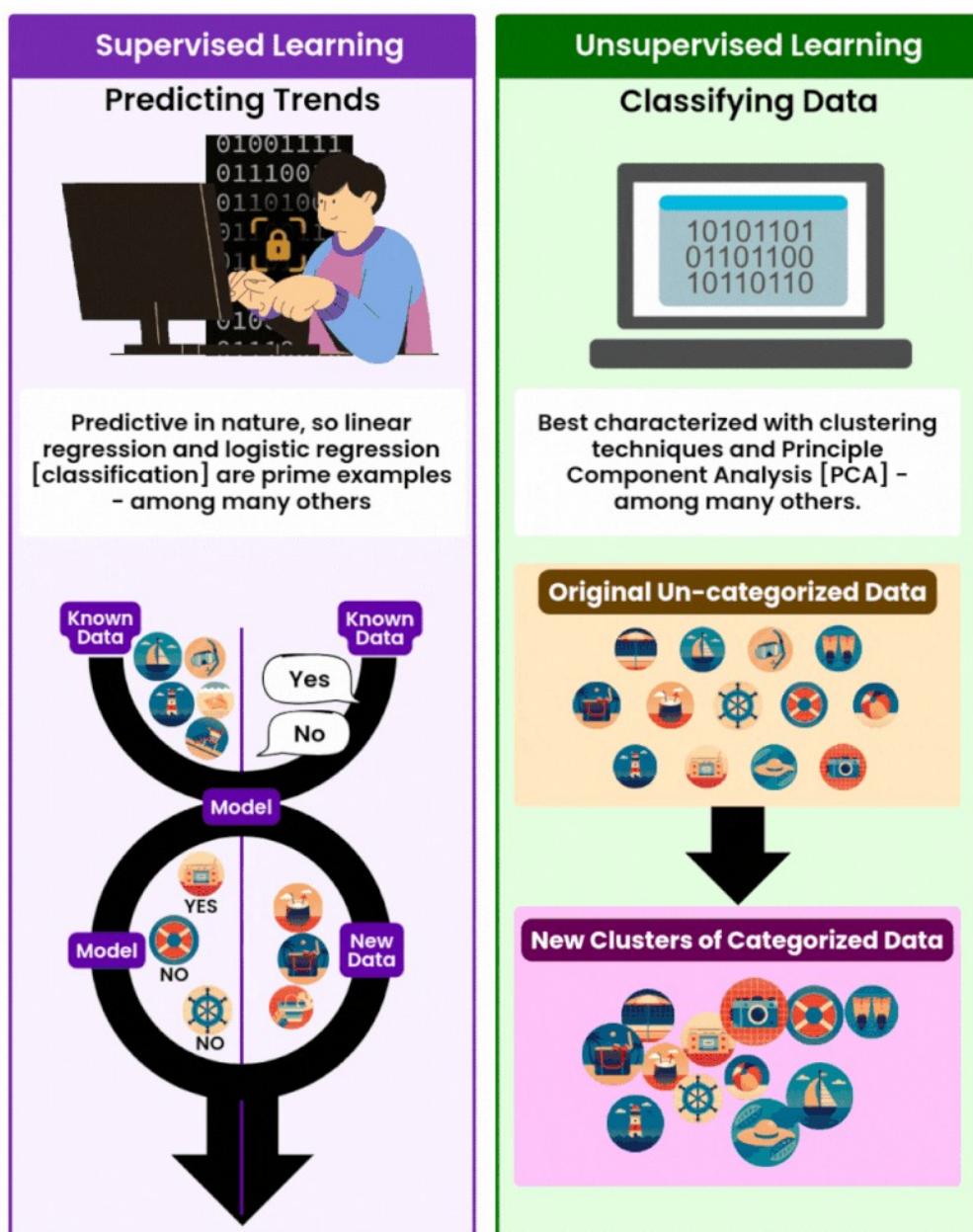


Figura 2 – Algoritmos de Aprendizado Supervisionado x Não-supervisionado.
Fonte: Adaptado de [2].

À medida que a IA continua a transformar as indústrias, a compreensão dos fundamentos do aprendizado de máquina é crucial.

1. Aprendizado Supervisionado:

- Definição: algoritmos aprendem com dados rotulados para prever resultados ou classificar novas entradas.
- Características-chave:
 - Requer dados de treinamento rotulados;
 - Função objetiva clara;
 - Os resultados são conhecidos durante o treinamento.
- Tipos:
 - a) Classificação: prevê categorias discretas.
Exemplo: Detecção de spam por e -mail (spam ou não spam).
 - b) Regressão: prevê valores contínuos.
Exemplo: Previsão de preços da casa com base nos recursos.
- Algoritmos populares:
 - Regressão linear
 - Regressão logística
 - Árvores de decisão
 - Florestas aleatórias
 - Máquinas vetoriais de suporte (SVM)
 - Redes Neurais
- Aplicativos do mundo real:
 - Pontuação de crédito
 - Reconhecimento da imagem
 - Diagnóstico da doença
 - Análise de sentimentos

2. Aprendizado Não-Supervisionado:

- Definição: Algoritmos encontram padrões em dados não marcados sem saídas predefinidas.
- Características-chave:
 - Funciona com dados não marcados

- Nenhuma variável de destino específica
 - Descobre estruturas ocultas em dados
- Tipos:
 - a) Clustering: grupos pontos de dados semelhantes.
Exemplo: segmentação do cliente para marketing direcionado.
 - b) Redução da dimensionalidade: reduz o número de recursos enquanto preservou as informações.
Exemplo: Análise de componentes principais (PCA) para compactação de dados.
 - c) Associação: descobre regras que descrevem grandes partes dos dados
Exemplo: Análise de cesta de mercado (clientes que compram x frequentemente compram y).
- Algoritmos populares:
 - Cluster de k-means
 - Cluster hierárquico
 - Dbscan
 - Análise de componentes principais (PCA)
 - T-Sne
 - AutoEncoders
- Aplicativos do mundo real:
 - Detecção de anomalia em segurança cibernética
 - Sistemas de recomendação
 - Análise de sequência genética
 - Modelagem de tópicos no processamento de linguagem natural

3. Escolhendo entre supervisionado e não supervisionado:

- i) Disponibilidade de dados: dados rotulados? Supervisionado. Dados não marcados? Não-supervisionado.
- j) Tipo de problema: previsão de resultados específicos? Supervisionado. Explorando a estrutura de dados? Não supervisionado.
- k) Recursos computacionais: supervisionados geralmente requer mais recursos para o treinamento.

- I) Interpretabilidade: Os modelos supervisionados geralmente são mais fáceis de interpretar.

Tendências emergentes:

- Aprendizagem semi-supervisionada: usa dados rotulados e não marcados.
- Transferência de aprendizado: aplica conhecimento de uma tarefa a outra.
- Aprendizagem de reforço: aprende através da interação com um ambiente

REFERÊNCIA

[1] PANJUTA, Denis. Top AI Algorithms Explained to Kids. Machine Learning Community. Published, 2024. Disponível em:
https://www.linkedin.com/feed/update/urn:li:activity:7209575140362194945?utm_source=share&utm_medium=member_desktop. Acessado em: Jun 20, 2024.

[2] PANDEY, B. K. Demystifying Machine Learning: Supervised vs. Unsupervised Approaches, Linkedin Post, 2024. Disponível em:
https://media.licdn.com/dms/image/v2/D4E22AQHiS_VRXreWog/feedshare-shrink_800/feedshare-shrink_800/0/1728269388801?e=1731542400&v=beta&t=PGV9dzN2UjTN7-S2jervdY4mK6MpCrsYhr8_YMHL5LU. Acessado em Out 10, 2024.

11. CONCEITOS ESSENCIAIS DE INTELIGÊNCIA ARTIFICIAL PARA INICIANTES

Por: Denis Panjuta

Editado por: Dr. Arnaldo de Carvalho Junior

Data: 25/06/2024

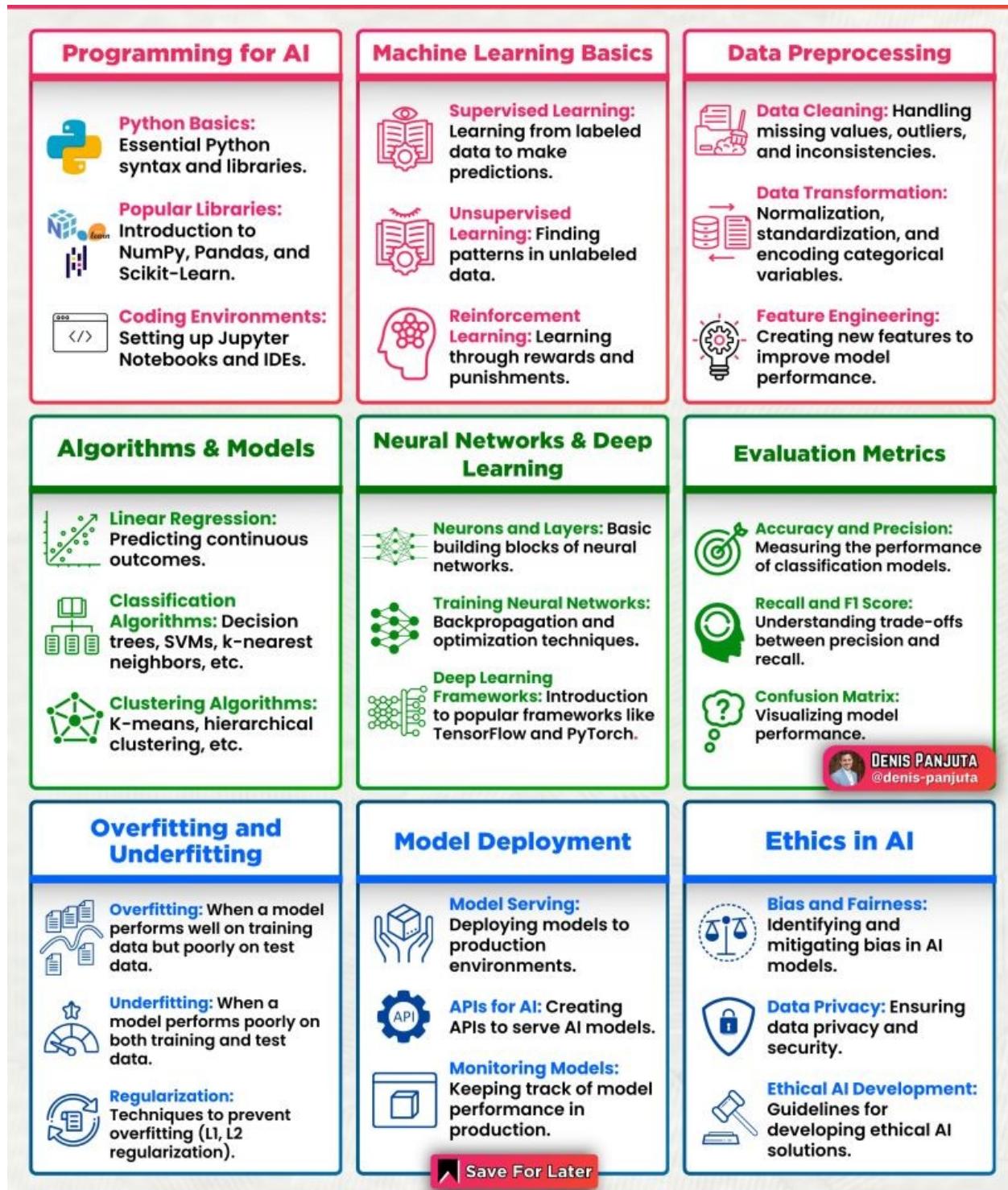


Figura 1 - IA para Iniciantes. Nove Conceitos Essenciais de IA Explicados.

Fonte: Adaptado de [1].

Nove conceitos essenciais de Inteligência Artificial (IA) explicados.

Aqui está o que está coberto:

1. Programação para IA;
2. Noções básicas de aprendizado de máquina (*machine learning* - ML);
3. Pré -processamento de dados;
4. Algoritmos e modelos;
5. Redes neurais artificiais (*artificial neural networks* – ANN) e aprendizado profundo (*deep learning* - DL);
6. Métricas de avaliação;
7. Excesso de ajuste e subjacência;
8. Implantação do modelo;
9. Ética na IA.

REFERÊNCIA:

- [1] PANJUTA, Denis. AI for Begineers. Machine Learning Community. Published, 2024. Disponível em: https://media.liecdn.com/dms/image/D4E22AQEeWYj4SDYz/wg/feedshare-shrink_800/0/1719043506730?e=1723075200&v=beta&t=dy7AjJQ8akSnXs7eojUQIGBfVa6OPYLadrBD4c4G4w. Acessado em: Jun 20, 2024.

12. HIERARQUIA DA INTELIGÊNCIA ARTIFICIAL - DA FUNDAÇÃO AO NÍVEL AVANÇADO

Por: Denis Panjuta.

Adaptado por: Dr. Arnaldo de Carvalho Junior

Data: Junho 27, 2024.

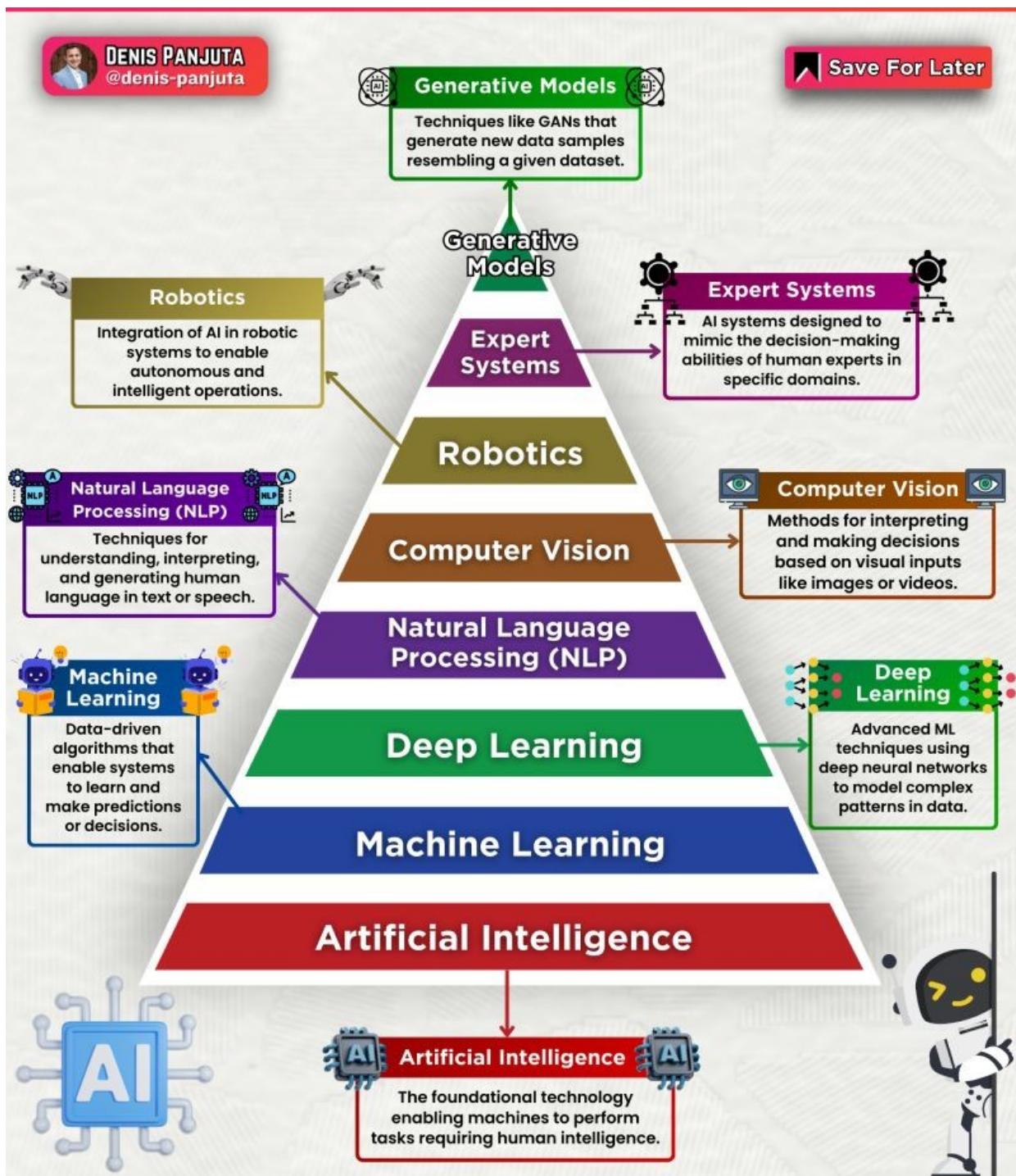


Figura 1 - Hierarquia da Inteligência Artificial – do básico ao avançado.

Fonte: Adaptado de [1].

Recursos gratuitos maciços para aprender Inteligência Artificial (IA)

1. Linguagem de Programação:

- a) Python: <https://www.mygreatlearning.com/academy/learn-for-free/courses/artificial-intelligence-with-python>
- b) R Programação: <https://www.codecademy.com/learn/learn-r>
- c) Java: <https://intellipaat.com/academy/course/java-training/>
- d) JavaScript:
<https://alison.com/topic/learn/71185/writing-your-first-javascript-application-learning-outcomes>

2. Fundamentos da Matemática

- a) Álgebra linear: <https://www.coursera.org/learn/linear-algebra-machine-learning>
- b) Probabilidade e estatística: <https://www.coursera.org/learn/linear-algebra-machine-learning>
- c) Cálculo: <https://www.coursera.org/learn/linear-algebra-machine-learning>
- d) Matemática Discreta: <https://www.youtube.com/watch?v=p2b2Vb-cYCs&list=PLBlnK6fEyqRhqJPDXcvYILfXPh37L89g3>

3. Manuseio de Dados (*Data Dandling*)

- a) Pandas e Numpy: <https://www.youtube.com/watch?v=GPVsHOIRBBI>
- b) Sqlite: <https://www.youtube.com/watch?v=byHcYRpMgl4>
- c) MongoDB: <https://www.youtube.com/watch?v=c2M-rlkkT5o>

4. Bibliotecas de Aprendizado de Máquina (*Machine Learning Libraries*)

- a) Scikit-Learn: https://www.youtube.com/watch?v=0B5eIE_1vpU
- b) TensorFlow: <https://www.youtube.com/watch?v=tPYj3fFJGjk>

c) Pytorch <https://www.youtube.com/watch?v=IA3WxTTPXqQ>

5. Aprendizagem Profunda e Redes Neurais (*Deep Learning and Neural Networks*)

a) Aprendizagem profunda: <https://www.youtube.com/watch?v=VyWAvY2CF9c>

b) Visão computacional: <https://www.youtube.com/watch?v=IA3WxTTPXqQ>

6. Desenvolvimento e Implantação de IA

a) Plataformas do Google AI: <https://www.youtube.com/watch?v=N2WSr7pt54g>

b) Azure AI Studio: <https://www.youtube.com/watch?v=fQ9RFR1KTbY>

7. Ferramentas de Visualização

a) Matplotlib: <https://www.youtube.com/watch?v=wB9C0Mz9gSo>

b) Seaborn: <https://www.youtube.com/watch?v=6GUZXDef2U0>

c) Plotly: <https://www.youtube.com/watch?v=GGL6U0k8WYA>

REFERÊNCIA

[1] PANJUTA, D. The AI Hierarchy: From Foundation to Advanced. Disponível em: https://media.lcdn.com/dms/image/D5622AQFXxvJvBDb0fg/feedshare-shrink_800/0/1719388809006?e=1722470400&v=beta&t=JxhmxDReS25AuODpcrLEdI8qV3ZgTn4ii27RkPwt_do. Acessado em Junho 27, 2024.

13. PRINCIPAIS FERRAMENTAS DE IA

Por: Denis Panjuta.

Adaptado por: Dr. Arnaldo de Carvalho Junior

Data: Junho 27, 2024.

1. AS 10 FERRAMENTAS DE IA MAIS UTILIZADAS EM 2024.

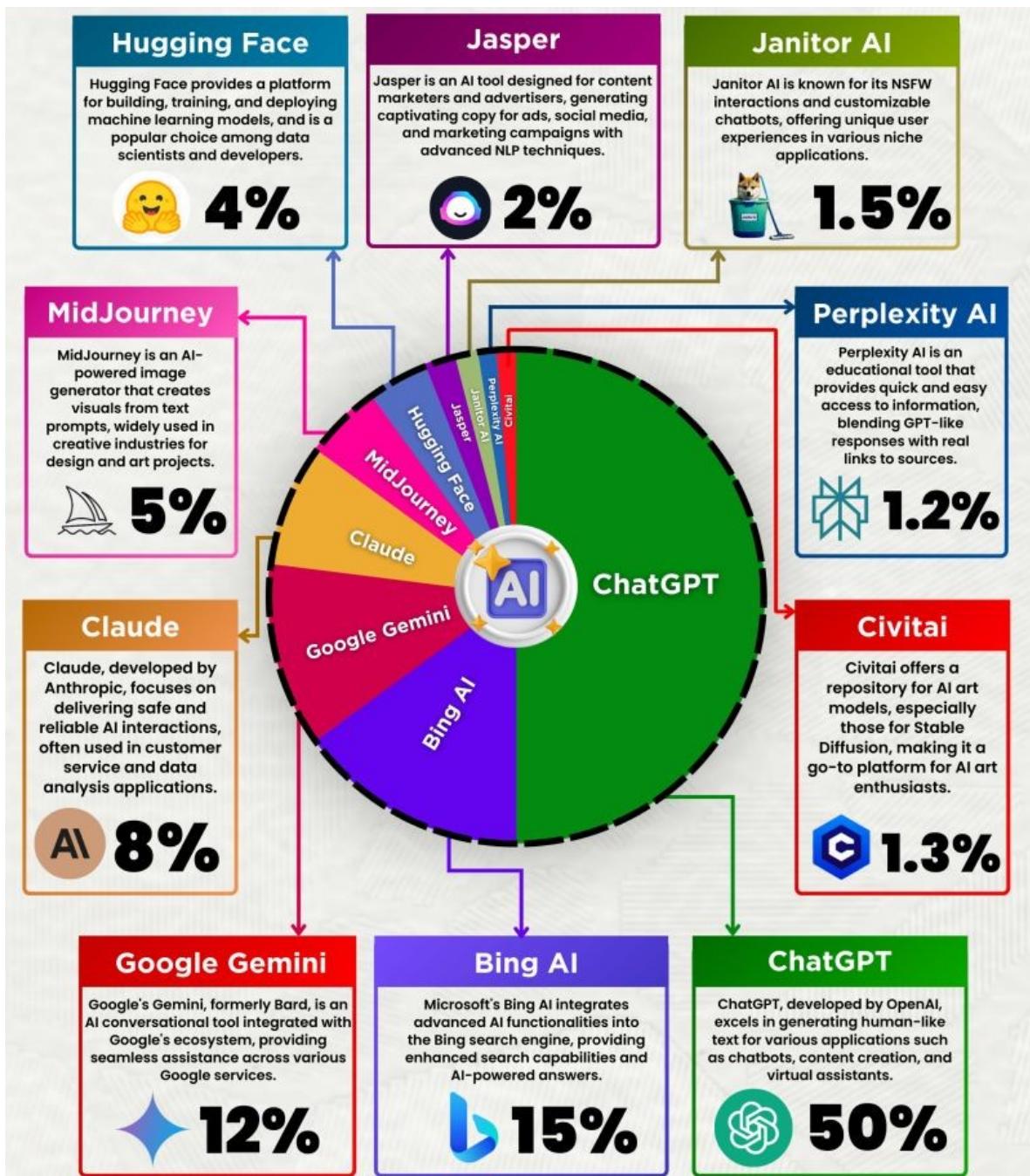


Figura 1 - Ferramentas de Inteligência Artificial Mais Utilizadas em 2024.

Fonte: Adaptado de [1].

1. ChatGPT (50%): se destaca na geração de texto semelhante ao humano para chatbots, criação de conteúdo e assistentes virtuais.
2. Bing AI (15%): aprimora a pesquisa do Bing com respostas a IA e recursos avançados de pesquisa.
3. Google Gemini (12%): uma ferramenta de conversação de IA integrada aos serviços do Google para obter assistência robusta.
4. Claude (8%): fornece interações de IA seguras e confiáveis para o atendimento ao cliente e análise de dados.
5. Midjourney (5%): transforma o texto solicita visuais impressionantes, perfeitos para criativos.
6. Hugging Face (4%): uma plataforma para construir, treinar e implantar modelos de aprendizado de máquina.
7. Jasper (2%): gera cópia cativante para anúncios, mídias sociais e campanhas de marketing.
8. Civitai (1,3%): um repositório para modelos de arte de IA, popular entre os entusiastas da AI Art.
9. Janitor AI (1,5%): chatbots personalizáveis que oferecem experiências exclusivas do usuário.
10. Perplexity AI (1,2%): fornece acesso rápido e fácil às informações com respostas do tipo GPT e links reais.

2. AS 24 PRINCIPAIS FERRAMENTAS DA INTELIGÊNCIA ARTIFICIAL (IA)!

2.1. Para projetar:

- a) Adobe Firefly: desencadeie sua criatividade com esta IA generativa livre.
- b) Canva: sua ferramenta on-line preferida para design gráfico, simplificada.
- c) FIGMA: a ferramenta de design de interface colaborativa final.
- d) Design AI: Crie sem esforço logotipos gerados pela IA.
- e) Uizard: Transforme suas ideias em designs impressionantes com ferramentas de interface do usuário movidas a IA.
- f) Prezi: Reinvente suas apresentações com esta plataforma de narrativa baseada na Web.

2.2. Para gerenciamento de mídia social:

- a) Later: Eleve seu jogo de mídia social com agendamento inteligente.
- b) Buffer: Spark criatividade sem fim para seus *feeds* sociais.
- c) Hootsuite: automatize a geração de legenda e encontre inspiração após a inspiração.
- d) Ocoya: Combine a programação com a criação de conteúdo perfeitamente.
- e) Tweetyai: envolva seu público com *tweets* gerados pela IA.
- f) Rapidely: simplificar a criação de conteúdo com áudio de tendências para bobinas.

2.3. Para Marketing:

- a) Hubspot: capacite sua empresa com ferramentas gratuitas de gerenciamento do relacionamento com o cliente (*customer relationship management* - CRM).
- b) SmartWriter: Personalize sua divulgação com e-mails frios criados pela IA.
- c) Drift: conecte-se com seus compradores por meio de uma plataforma de engajamento acionada por IA.
- d) Browse AI: automatize a coleta de dados nas páginas da web.
- e) Reply IO: Crie e-mails de vendas semelhantes a humanos em escala.
- f) Jasper: o assistente de IA para as necessidades criativas da sua equipe de marketing.

2.4. Para produtividade:

- a) Quillbot: refine sua redação com ferramentas de edição avançada.
- b) Emma: gerencie seus planos de viagem e compromissos com facilidade.
- c) GPTBOTS: Crie aplicativos de negócios orientados por IA sem codificação.
- d) Venturefy: transforme dados complexos em mapas claros e interativos.
- e) Bardeen AI: simplifique seu fluxo de trabalho automatizando tarefas de rotina.
- f) Fireflies ai: Nunca perca um momento com a captura e transcrição de reuniões de IA.

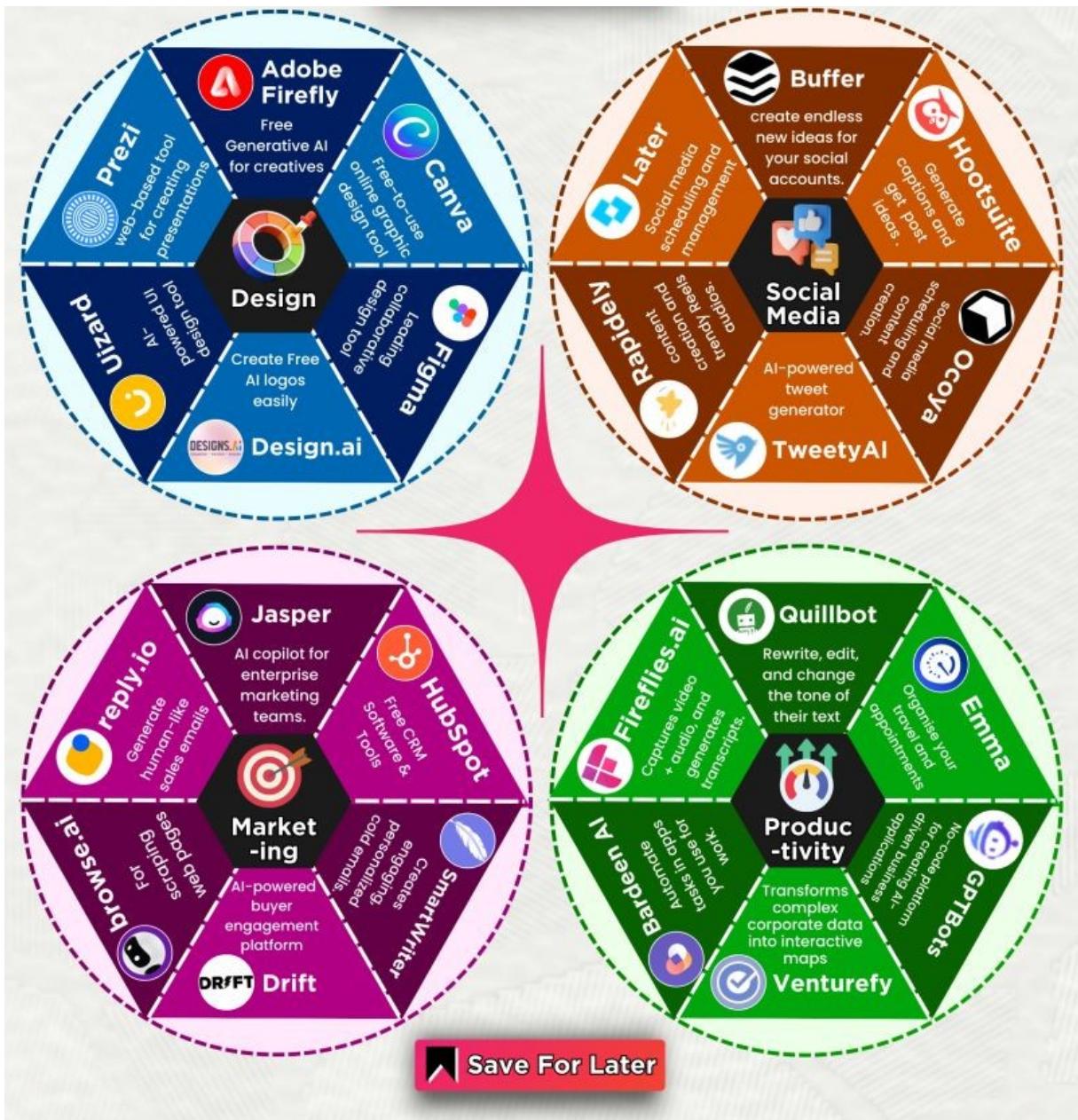


Figura 2 - Lista com as 24 ferramentas de IA essenciais.

Fonte: Adaptado de [2].

REFERÊNCIAS

- [1] PANJUTA, D. Most Used AI Tools in 2024. Disponível em: https://media.linkedin.com/dms/image/D4E22AQEnAdWisuj-IA/feedshare-shrink_800/0/1719475225887?e=1722470400&v=beta&t=edlh-aMQvf2QVHcvMoLNJfIMtojhHvDpFECZqygFAio. Acessado em Junho 27, 2024.
- [2] PANJUTA, D. List of 24 Must Have AI Tools. Disponível em: https://media.linkedin.com/dms/image/D4E22AQGsCCKiTyQcsw/feedshare-shrink_800/0/1719410456948?e=1722470400&v=beta&t=Q231p0HpNAH0bhbfAW22_IQAbnowyEO2XbOdMGFFD0. Acessado em Junho 27, 2024.

14. TERMOS MAIS UTILIZADOS EM INTELIGÊNCIA ARTIFICIAL

Por: Gina Acosta Gutiérres

Adaptado por: Dr. Arnaldo de Carvalho Junior

Data: Julho 01, 2024.

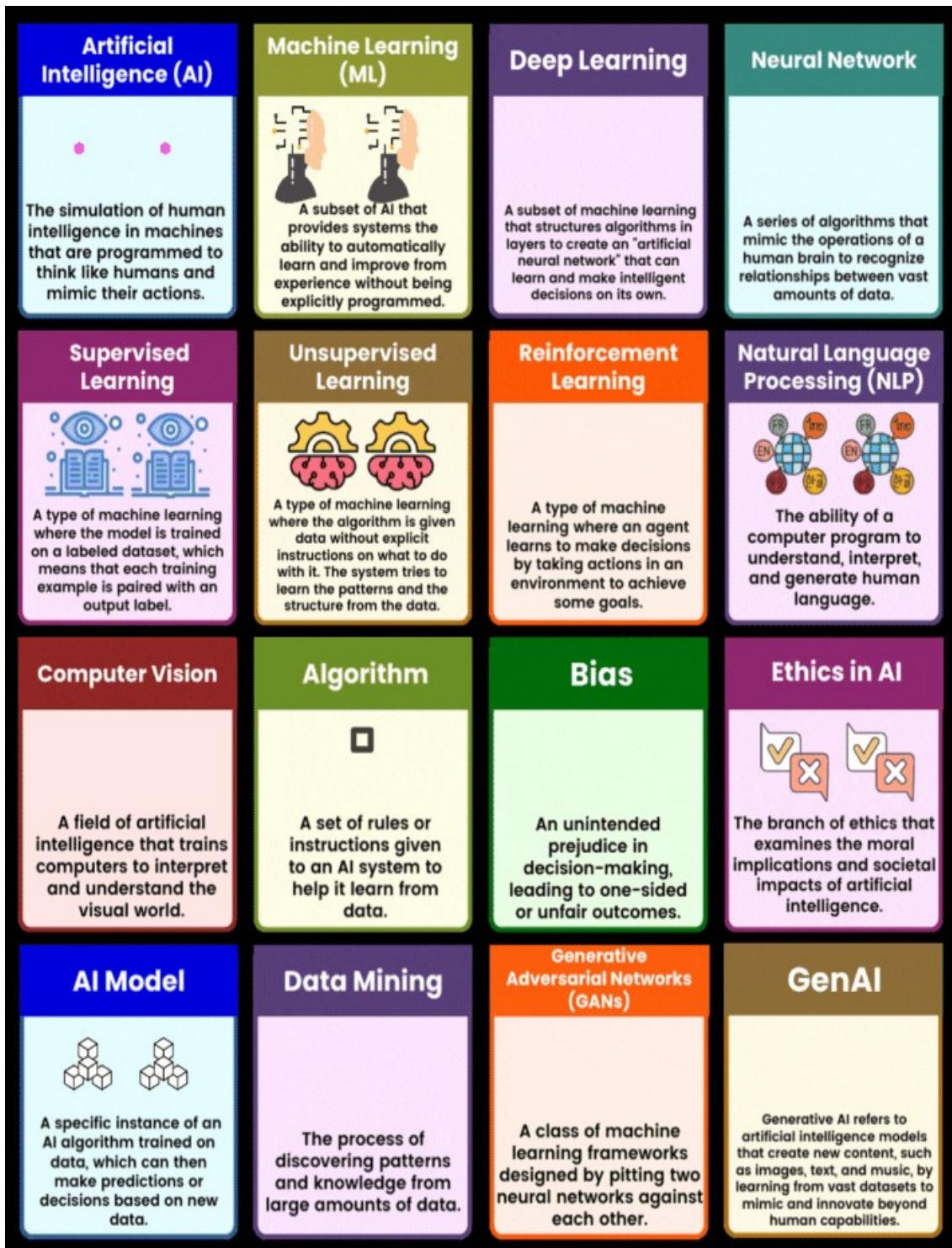


Figura 1 - Termos que se deve conhecer sobre Inteligência Artificial.

Fonte: Adaptado de [1].

Básico de Inteligência Artificial (IA) em Resumo: Termos Essenciais de Forma Simplificada.

Como a IA molda o futuro da humanidade, conhecer seus termos básicos se torna importante para todos. Aqui está um guia fácil de entender:

- a) **Algoritmo (*algorithm*):** Instruções para os computadores resolverem problemas.
- b) **Viés (*Bias*):** preconceito injusto em dados ou algoritmos.
- c) **Inteligência Artificial – IA (*artificial intelligence – AI*):** máquinas projetadas para pensar e agir como seres humanos.
- d) **Aprendizado de Máquinas – AM (*machine learning – ML*):** sistemas que aprendem com dados para melhorar automaticamente.
- e) **Aprendizado Profundo – AP (*deep learning – DL*):** Uma forma de ML poderoso usando redes complexas e inspiradas no cérebro.
- f) **Redes Neurais Artificiais – RNA (*artificial neural networks – ANN*):** Uma rede interconectada de neurônios artificiais que processa informações.
- g) **Aprendizado Supervisionado – AS (*supervised learning – SL*):** Treinamento com dados rotulados (entradas e saídas desejadas).
- h) **Aprendizado Não-supervisionado - AN (*unsupervised learning – UL*):** Encontrar padrões em dados não marcados.
- i) **Aprendizado por Reforço – AR (*reinforcement learning – RL*):** Aprendendo através de tentativa e erro com recompensas e penalidades.
- j) **Processamento de Linguagem Natural – PLN (*natural language processing – NLP*):** máquinas que entendem e processam a linguagem humana.
- k) **Visão Computacional (*computer vision*):** máquinas que veem e interpretam informações visuais.
- l) **Ética em IA (*ethics in AI*):** Considerando as implicações morais do desenvolvimento da IA.
- m) **Redes Adversárias Generativas – RAG (*generative adversarial networks – GAN*):** Duas redes concorrentes que se treinam para gerar dados realistas.

- n) **IA Generativa (generative AI – GenAI):** tecnologia para criar conteúdo semelhante ao trabalho feito pelo homem.
- o) **Modelo de IA (AI model):** Programa de computador treinado para fazer previsões ou decisões.
- p) **Minearação de Dados (data mining – DM):** Descobrindo conhecimentos e percepções (*insights*) de grandes conjuntos de dados.

REFERÊNCIA

[1] GUTIÉRRES, G. A. Artificial Intelligence – terms You Must Know. Disponível em: https://media.liecdn.com/dms/image/D4E22AQENcHDHk_m4Q/feedshare-shrink_2048_1536/0/1719816903619?e=1722470400&v=beta&t=AD8kS9qzXEGaWnFk4_p6cU3xFXG_ErC76mjeW4RBUDg.
Acessado em Julho 01, 2024.

15. VINTE ABREVIACÕES DE INTELIGÊNCIA ARTIFICIAL

Por: Generative AI

Adaptado por: Dr. Arnaldo de Carvalho Junior

Data: Julho 02, 2024.

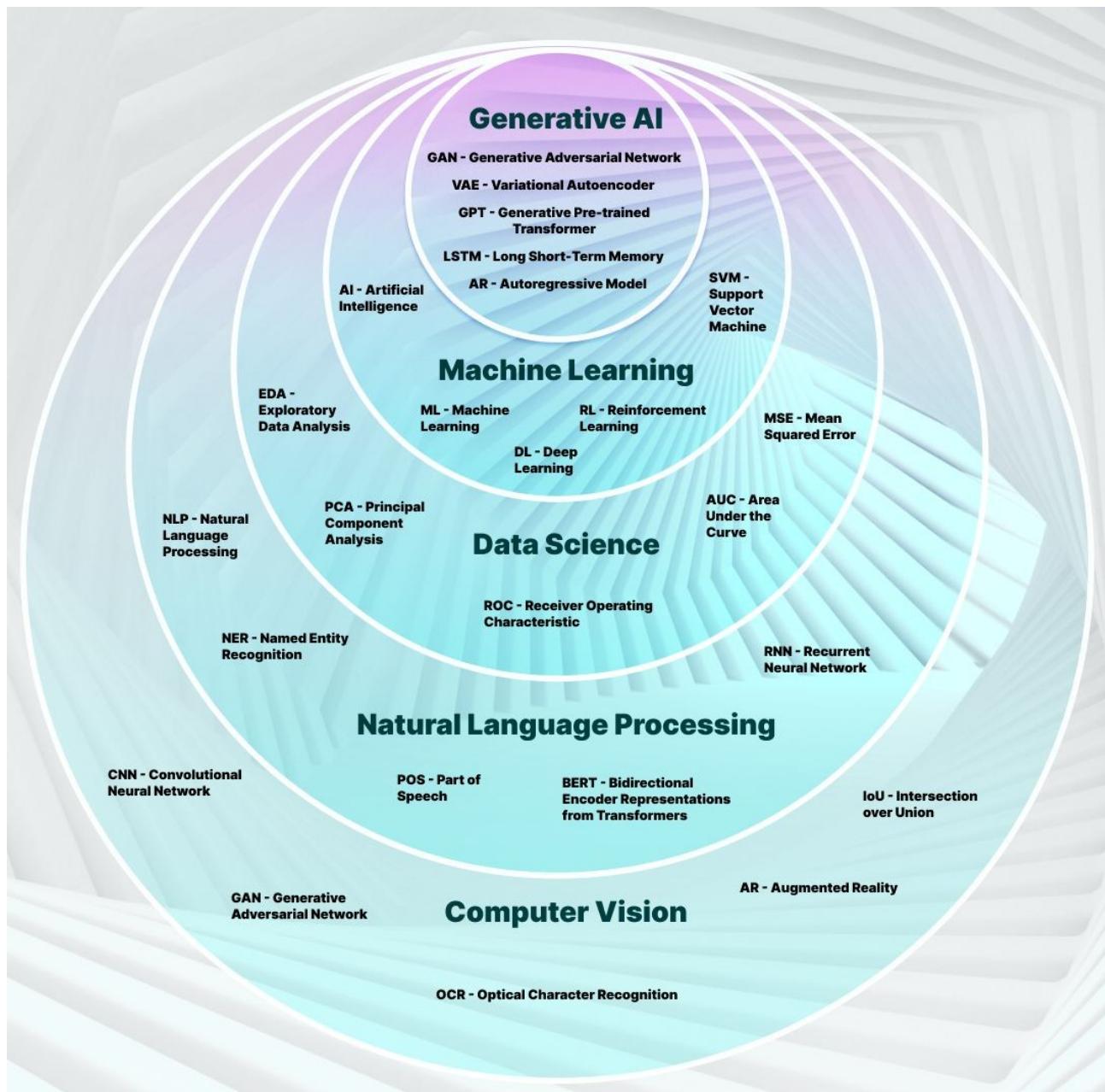


Figura 1 - 20 abreviações essenciais para compreender inteligência artificial(IA).

Fonte: Adaptado de [1].

Compreender as abreviações do universo da inteligência artificial (IA) ajudará o leitor a compreender a literatura de IA de forma mais efetiva:

1. Aprendizado de Máquina

- oo) Inteligência Artificial - IA (*Artificial Intelligence – AI*)
- pp) Aprendizado de Máquina – AM (*Machine Learning – ML*)
- qq) Aprendizado Profundo – AP (*Deep Learning – DL*)
- rr) Aprendizado por Reforço – AR (*Reinforcement Learning – RL*)
- ss) Máquina Vetorial de Suporte – MVS (*Support Vector Machine – SVM*)

2. Ciência de Dados (Data Science)

- m) Análise Exploratória de Dados – AED (*Exploratory Data Analysis – EDA*)
- n) Análise do Componente Principal – ACP (*Principal Component Analysis – PCA*)
- o) Características Operacionais do Receptor – COR (*Receiver Operating Characteristic – ROC*)
- p) Área Sob a Curva – ASC (*Area Under the Curve – AUC*)
- q) Erro Médio Quadrático – EMQ (*Mean Squared Error – MSE*)

3. Processamento de Linguagem Natural

- Rede Neural Recorrente – RNR (*Recurrent Neural Network - RNN*)
- Processamento da Linguagem Natural – PLN (*Natural Language Processing – NLP*)
- Reconhecimento de Entidade Nomeada – REN (*Named Entity Recognition – NER*)
- Parte da Conversa - POS (*Part of Speech*)
- Representação de Codificador Bidirecional de Transformadores – BERT(*Bidirectional Encoder Representations from Transformers*)

4. Visão Computacional (Computer Vision)

- Rede Neural Convolucional – RNC (*Convolutional Neural Network – CNN*)
- Realidade Aumentada – RA (*Augmented Reality – AR*)
- Rede Adversária Generativa – RAG (*Generative Adversarial Network – GAN*)
- Reconhecimento Ótico de Caracteres – ROC (*Optical Character Recognition – OCR*)
- Interseção Sobre a União – ISU (*Intersection Over Union - IOU*)

REFERÊNCIA

- [1] GENERATIVE AI – 20 essential abbreviations to understand AI, 2024. Disponível em: https://media.linkedin.com/dms/image/D4D22AQHK9G6FiZ6ivA/feedshare-shrink_1280/0/1719916486548?e=1723075200&v=beta&t=ZGPm_Gj6LfO6-cMdS9DJcdUHI97OPw-Dvl8LXqPXGCI. Acessado em Julho 02, 2024.

16. ROTEIRO DE PROJETOS DE ML E IA

Adaptado por: Dr. Arnaldo de Carvalho Junior

Data: Agosto 19, 2024.

Computadores usam dados para aprender e realizar melhores previsões ou decisões através do tempo, aplicadas em tarefas como reconhecimento de imagens e recomendações de sistemas.

1. O ROTEIRO DO APRENDIZADO DE MÁQUINAS

A Figura 1 apresenta o processo de aprendizado de máquinas (*machine learning – ML*).

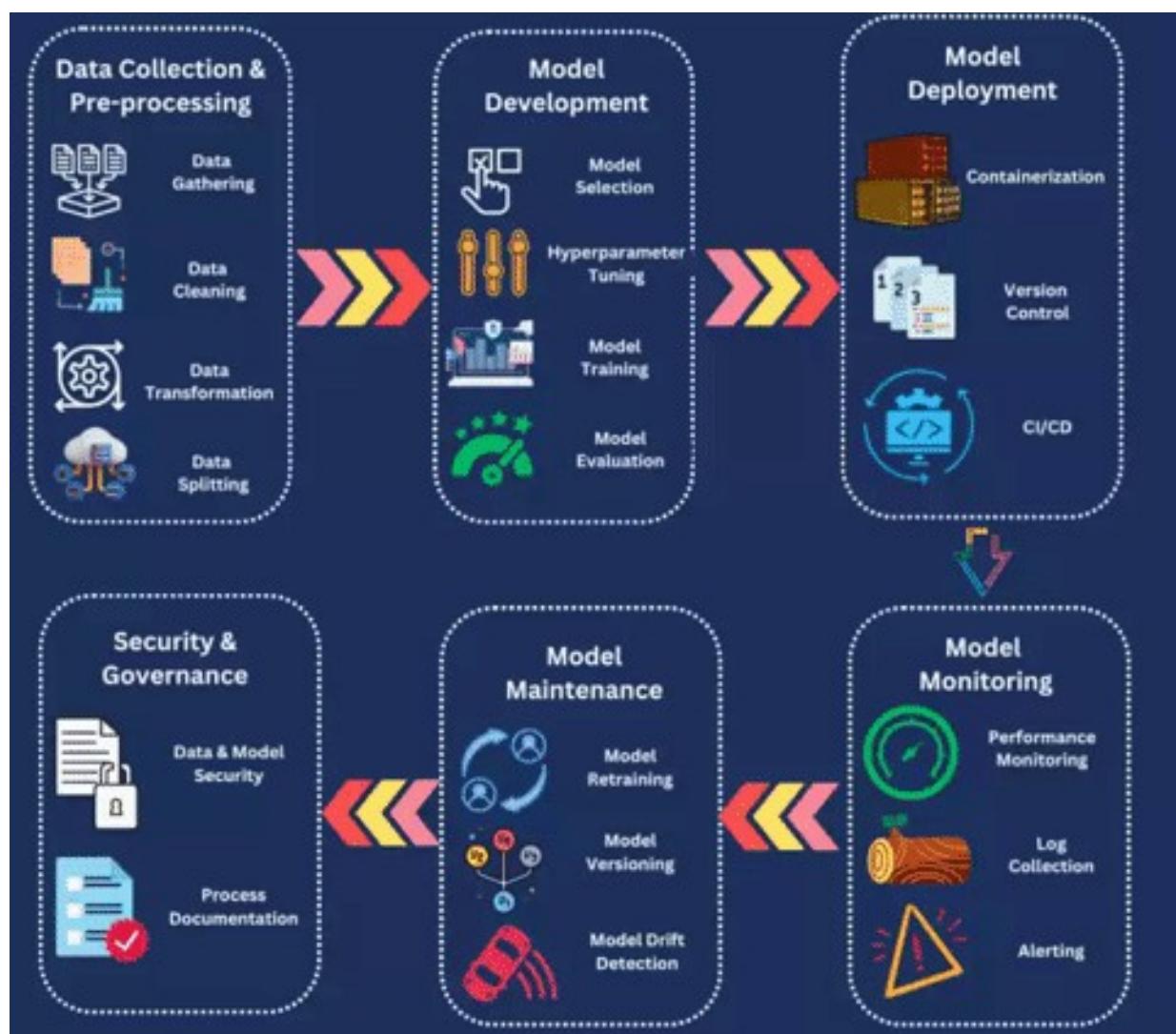


Figura 1 – Processo de ML.

Fonte: Adaptado de [1].

1.1. Entenda o básico:

- Aprenda conceitos fundamentais (aprendizado supervisionado versus não supervisionado).
- Fique confortável com conceitos matemáticos (álgebra linear, estatísticas).

1.2. Escolha suas ferramentas:

- tt) Linguagens de programação mestre (Python, R).
- uu) Explore bibliotecas e Estruturas (*Libraries and Frameworks*) (Tensorflow, Scikit-Learn).

1.3. Mergulhe em dados:

- r) Aprenda técnicas de pré -processamento de dados.
- s) Entenda a disputa e a limpeza de dados.

1.4. Explore os Algoritmos:

- Estude algoritmos populares (regressão, classificação, agrupamento).
- Implementar modelos de aprendizado de máquina e avaliar seu desempenho.

1.5. Trabalhe em projetos:

- Aplique seu conhecimento a problemas do mundo real.
- Construa um portfólio de projetos de aprendizado de máquina.

1.6. Mantenha-se atualizado:

- Siga as últimas tendências e pesquisas em ML.
- Leia blogs, papéis e participem on-line.

1.7. Rede e Colaboração:

- k) Junte -se às comunidades e fóruns da ML.
- l) Participe de hackathons e desafios.

1.8. Aprendizagem contínua:

- g) Faça cursos e certificações avançadas

2. FLUXO DE TRABALHO PARA GUIAR O INICIANTE EM APRENDIZADO DE MÁQUINAS.

O fluxo de trabalho de ML é apresentado na Figura 2.

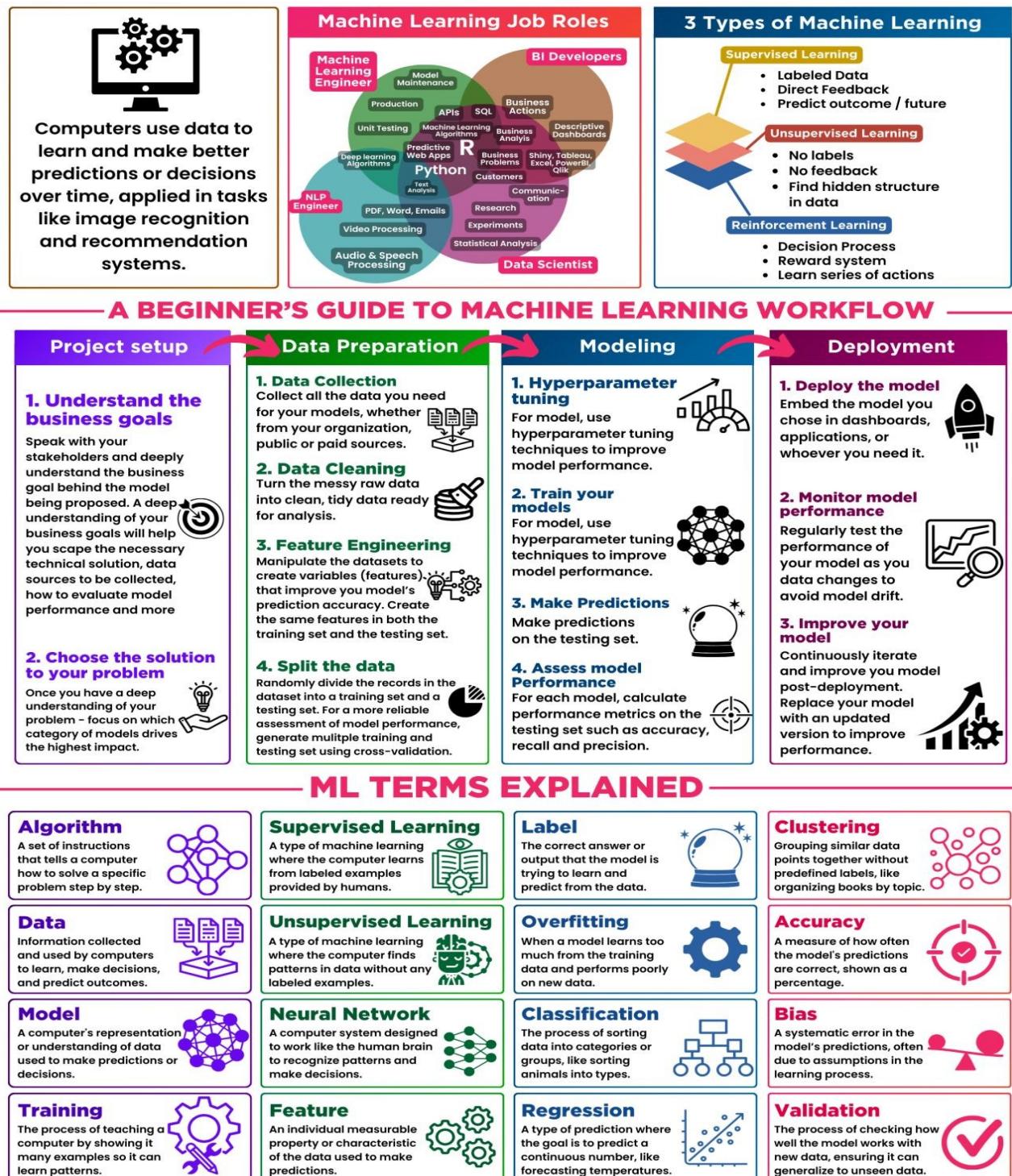


Figura 2 - Aprendizado de Máquinas Explicado em Poucas Palavras.

Fonte: Adaptado de [2].

2.1. Configuração do Projeto

- a) Entenda os gols do negócio:** O desenvolvedor deve conversar com as partes interessadas (*stakeholders*) e profundamente compreender o gol do negócio por trás do modelo sendo proposto. Uma compreensão profunda dos gols do negócio ajudarão o desenvolvedor a procurar a solução técnica necessária, fontes de dados a serem coletados como avaliar a performance do modelo e muito mais.
- b) Escolha a solução para o problema:** Uma vez que o desenvolvedor adquira uma profunda compreensão do problema pode se concentrar em quais categorias de modelos conduzem ao impacto mais elevado.

2.2. Preparação dos Dados

- a) Coleção de Dados:** O desenvolvedor deve coletar todos os dados necessários para o modelo, tanto da organização, órgãos públicos e fontes pagas.
- b) Limpeza dos Dados:** O desenvolvedor deve transformar o massivo conjunto de dados brutos em dados limpos, organizados e prontos para análise.
- c) Engenharia de Recursos:** O *dataset* deve ser manipulado para criar variáveis (*features*), que melhoram a acurácia do modelo. Criar os mesmos *features* em ambos os conjuntos, treinamento e de teste.
- d) Separe os Dados:** Os registros devem ser aleatoriamente divididos no *dataset* entre conjunto de dados de treinamento e de teste/validação. Para uma avaliação da performance do modelo, devem ser gerados múltiplos conjuntos de treinamento e teste usando avaliação cruzada.

2.3. Modelagem

- a) Sintonia dos Hiperparâmetros:** Para modelagem, técnicas de sintonia de hiperparâmetros devem ser usadas, de modo a melhorar a performance do modelo.
- b) Treine o modelo:** O modelo deve ser treinado e os hiperparâmetros sintonizados.
- c) Faça Predições:** Predições devem ser realizadas utilizando o conjunto de dados de teste (*testing set*).
- d) Avalie o Desempenho do Modelo:** Para cada modelo, métricas de performance devem ser calculadas no conjunto de teste, como acurácia, recalibração e precisão.

2.4. Implementação

- a) Implemente o Modelo:** O pesquisador pode embutir o modelo escolhido em painéis gráficos (*dashboards*), aplicações ou onde o modelo for requerido.
- b) Monitore a Performance do Modelo:** A performance do modelo deve ser testada regularmente conforme os dados de entrada variem para se evitar desvios do modelo.
- c) Melhore o Modelo:** O pesquisador deve continuamente interagir e melhorar o modelo após a implementação, e eventualmente substituir/atualizar o modelo por uma versão com performance melhorada.

3. CICLO DE VIDA DE PROJETO DE ML

O ciclo de vida do aprendizado de máquina é uma jornada abrangente, desde a definição de objetivos claros do projeto até a manutenção dos modelos após a implantação.

- a) Definir os objetivos do projeto:** Tudo começa com a identificação de problemas de negócios, adquirindo a experiência certa do assunto e considerando cuidadosamente os riscos e os critérios de sucesso antes de avançar.
- b) Adquirir e explorar os dados:** A próxima etapa, adquirindo e explorando dados, é crucial seja dividindo dados, executando a análise de dados exploratórios e os recursos de engenharia para garantir que os dados estejam limpos e prontos para modelagem.
- c) Modelar os dados:** Depois que os dados são preparados, é hora de modelar dados selecionando as variáveis corretas, construindo modelos candidatos e validando-os rigorosamente para escolher o melhor ajuste para o problema em questão.
- d) Interpretar e Comunicar:** Interpretar e comunicar os resultados do modelo de maneira eficaz às partes interessadas garante que as ideias sejam açãoáveis e compreensíveis.
- e) Implementar, Documentar e Manter:** Por fim, a implementação, documentação e manutenção do modelo garante que ele continua a agregar valor ao longo do tempo, com um plano claro para monitorar e atualizar.

A Figura 3 apresenta um resumo do ciclo de vida de um projeto de ML.

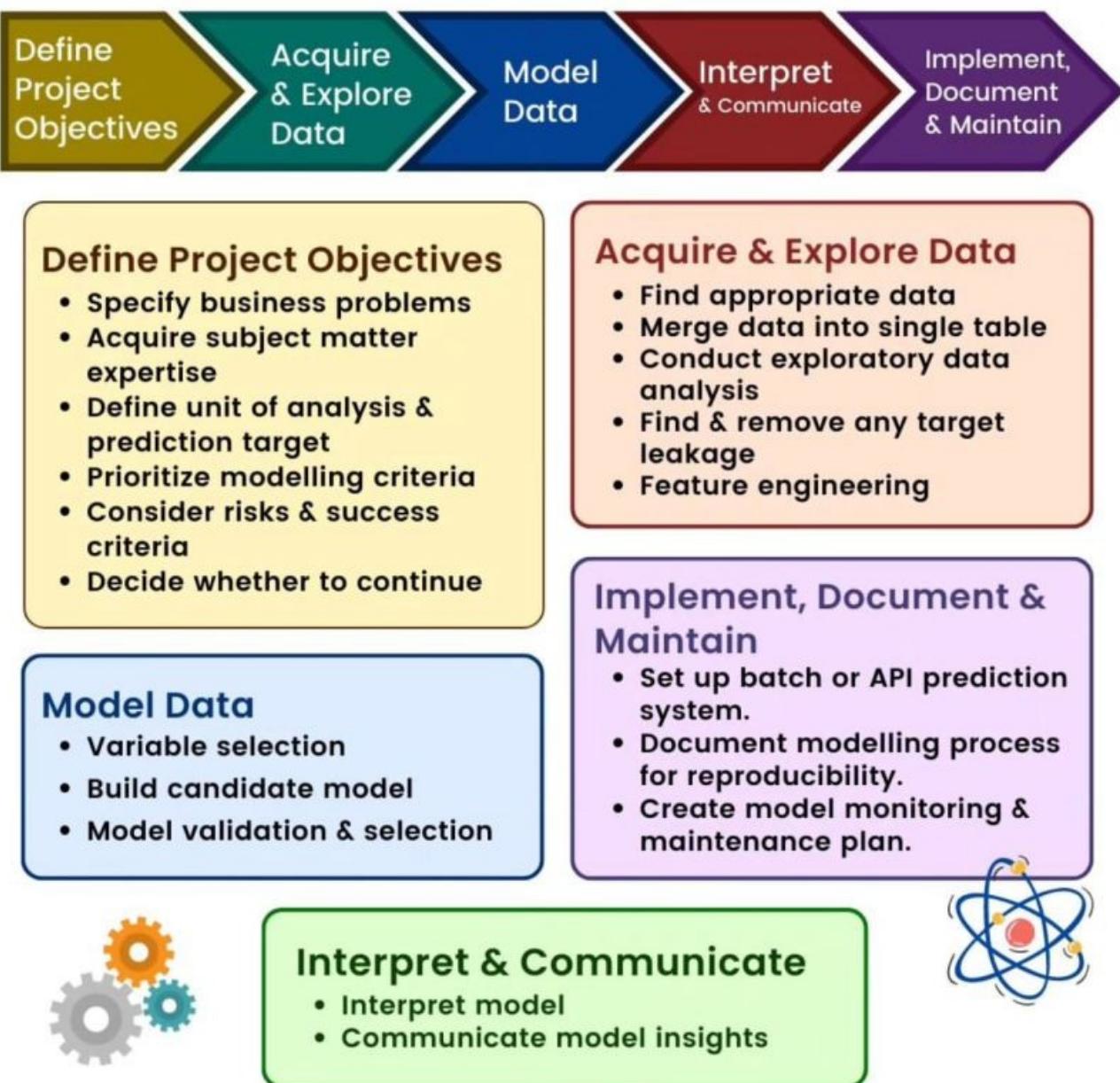


Figura 3 – Ciclo de Vida de Projeto de ML.

Fonte: Adaptado de [3].

4. PROJETOS DE INTELIGÊNCIA ARTIFICIAL: DA CONCEPÇÃO À IMPLANTAÇÃO

A Figura 4 apresenta de forma comprehensiva um guia para gerentes de projeto e empresários, atravessarem cada estágio em um projeto de inteligência artificial (IA), da concepção inicial à implantação bem-sucedida.

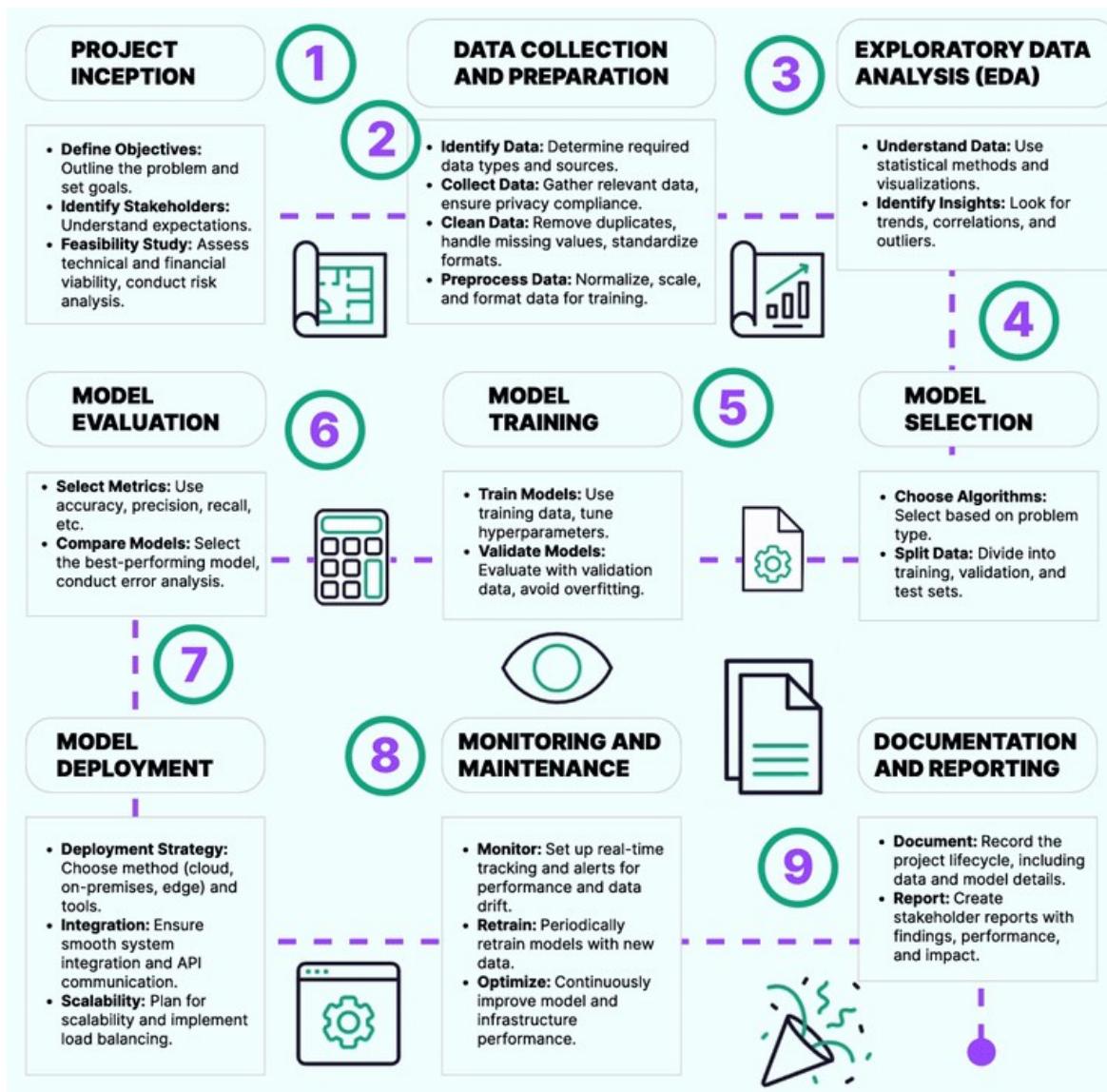


Figura 4 – Projeto de IA: Da Concepção Inicial à Implantação.
Adaptado de [4].

À medida que cada vez mais a sociedade se aproveita do poder transformador da IA, a jornada desde a concepção à implantação pode parecer assustadora. Ter um roteiro claro é crucial para navegar nas complexidades dos projetos de IA.

Este documento contém dicas abrangentes que encapsula todos os estágios críticos de um projeto de IA, oferecendo um guia passo a passo para garantir o sucesso. As fases essenciais são:

3.1. Concepção do Projeto:

- Defina os objetivos: comece com um esboço claro do problema e defina metas tangíveis.
- Identifique as partes interessadas: entenda as expectativas e papéis de todas as partes envolvidas.
- Estudo de viabilidade: Avalie a viabilidade técnica e financeira, junto com a análise de risco.

3.2. Coleta e Preparação de Dados:

- Identifique dados: determine os tipos e fontes de dados necessários. Colete dados: colete dados relevantes, garantindo a conformidade com a privacidade.
- Dados limpos: remova duplicatas, manipule valores ausentes e padronize formatos.
- Dados de preprocessamento: normalize, escala e formate os dados para treinamento.

3.3. Análise de Dados Exploratórios (*Exploratory Data Analysis – EDA*):

- Entenda os dados: utilize métodos e visualizações estatísticas para entender os dados.
- Identifique *insights*: procure tendências, correlações e *outliers* que possam afetar o modelo.

3.4. Seleção do Modelo:

- Escolha Algoritmos: selecione Algoritmos apropriados com base no tipo de problema.
- Dados divididos: divida os dados em conjuntos de treinamento, validação e teste para avaliação robusta.

3.5. Treinamento do Modelo:

- Modelos de trem: use os dados de treinamento e os hiperparâmetros ajustados.

- Validar modelos: avalie com dados de validação para evitar o excesso de ajuste.

3.6. Avaliação do Modelo:

- Selecione métricas: use precisão, precisão, *recall*, etc., para medir o desempenho.
- Compare modelos: selecione o modelo de melhor desempenho e conduza a análise de erros.

3.7. Implantação do Modelo:

- Estratégia de implantação: escolha métodos (nuvem, local, borda) e ferramentas necessárias.
- Integração: garanta integração suave com sistemas e APIs existentes. Escalabilidade: planeje escalabilidade e implemente o balanceamento de carga.

3.8. Monitoramento e Manutenção:

- Monitor: configure rastreamento e alertas em tempo real para desempenho e desvio de dados.
- Retreinamento: periodicamente retreine modelos com novos dados.
- Otimizar: Melhore continuamente o desempenho do modelo e da infraestrutura.

3.9. Documentação e Relatórios:

- Documento: Registre o ciclo de vida do projeto, incluindo dados e detalhes do modelo.
- Relatório: Crie relatórios para as partes interessadas detalhando as descobertas, desempenho e impacto.

3.10. Dicas rápidas:

- Abordagem iterativa: siga um processo iterativo para melhoria contínua.
- Colaborar: mantenha uma comunicação clara com todas as partes interessadas.
- Considerações éticas: Garanta o uso ético da IA e a conformidade com os regulamentos.

Seguindo essas etapas estruturadas, pode-se otimizar os processos, mitigar riscos e aumentar a eficácia de soluções de IA e ML.

REFERÊNCIAS:

- [1] PATEL, H. Machine Learning Process. Artificial Intelligence, Machine Learning, Data Science & Robotics, Linkedin Group, 2024. Disponível em: https://media.licdn.com/dms/image/D4D22AQGXmiFQ-IYgpA/feedshare-shrink_800/0/1721897777963?e=1725494400&v=beta&t=I_RvMDVell5RzWUU2uA1KQ1P8-AKvvXUyB9ybL6JB0k. Acessado em Jul 30, 2024.
- [2] PANJUTA, D. Machine Learning Explained in a Nutshell. Machine Learning Community (Moderate), 2024. Disponível em: https://media.licdn.com/dms/image/D4E22AQH41orBhr8Jdg/feedshare-shrink_1280/0/1721121265915?e=1724284800&v=beta&t=yxOtfYUK7UgTbhzrbxirNw6q2V_KcC7Wucl7lvXto4. Acessado em Julho 16, 2024.
- [3] KHINVASARA, A. Machine Learning Life Cycle, The Ravit Show (Linkedin Group), 2024. Disponível em: <https://www.linkedin.com/feed/?showCollaborativeArticleModal=true>. Acessado em Agosto 19, 2024.
- [4] ANBARJAFARI, S.. Streamlining AI Projects: From Inception to Deployment, Machine Learning Community (Linkedin Group), Jul 2024. Disponível em: https://media.licdn.com/dms/image/D4D22AQEg5dp26aYqxw/feedshare-shrink_2048_1536/0/1721973151119?e=1725494400&v=beta&t=9JFD7SMBfMamqlwk5wgbGviZz-OjP1fXI7SwWEzOqYc. Acessado em Jul 30, 2024.

17. MELHORES RECURSOS PARA APRENDER SOBRE APRENDIZADO DE MÁQUINA

Adaptado por: Dr. Arnaldo de Carvalho Junior

Data: Setembro 10, 2024.

1. SEQUÊNCIA PARA APRENDER SOBRE ML

A Figura 1 apresenta uma sequência estruturada para o estudar sobre aprendizado de máquina (*machine learning* – ML).

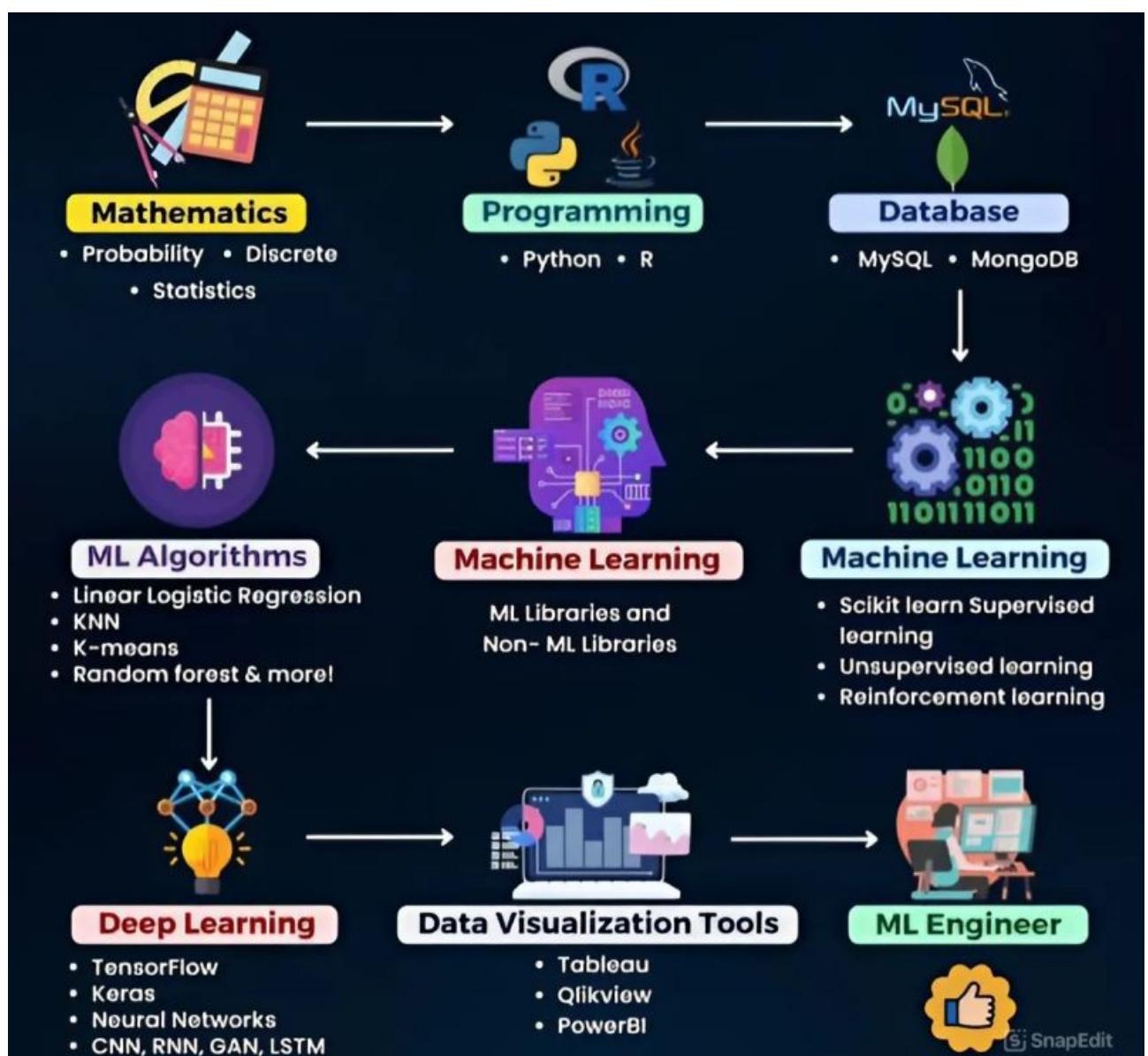


Figura 1: Sequência de estudo sobre ML

Fonte: Adaptado de [1].

a) Fundamentos da Matemática: deve-se começar pelo básico. O ML depende de sólidos fundamentos matemáticos. Focar em:

- h) Probabilidade e Estatística;
- i) Matemática Discreta.

b) Habilidades em Programação: O ML depende muito de estrutura de programação.

- t) Python: Alinguagem principal para o ML Já o
- u) R: Ótimo para análise estatística.

c) Gerenciamento de Base de Dados: Aprender a gerenciar dados é crucial para o ML.

- vv) MySQL é recomendado para dados estruturados.
- ww) MongoDB: para lidar com quantidades enormes de dados (*big data*) não estruturados.

d) Conceitos de ML: A prática mão na massa (*hands-on*) com ferramentas como o Scikit-Learn ajudará a solidificar esses conceitos. Entender os principais tipos de ML, como:

- Aprendizado supervisionado;
- Aprendizado não supervisionado;
- Aprendizagem de reforço.

e) Algoritmos ML Chave: Estar familiarizado com algoritmos ML populares, é essencial para resolver problemas de diferentes tipos.

- Regressão linear e logística;
- Vizinhos mais antigos (KNN);
- Cluster de k-means
- Florestas aleatórias.

f) Técnicas de aprendizado profundo (*deep learning – DL*): Inteligência Artificial avançada com estruturas de aprendizado profundo:

- Tensorflow e Keras;
- Redes Neurais CNN, RNN, LSTM e GAN.

g) Visualização de Dados: Ferramentas de visualização eficazes são essenciais para a comunicação e insights:

- m) Tableau;
- n) PowerBI;
- o) QlikView.

2. TREINAMENTOS, GUIAS E TUTORIAIS



Figura 2 - Melhores recursos gratuitos para aprendizado de máquinas.

Fonte: Adaptado de [1].

- 1. TensorFlow:** tutoriais oficiais - <https://www.tensorflow.org/tutorials?hl=pt-br>
- 2. Scikit-learn:** Guias comprehensivos e exemplos - https://scikit-learn.org/stable/user_guide.html

- 3. Kaggle:** trilhas para estudar aprendizado de máquina (*machine learning* – ML), Python, Panda, etc. - <https://www.kaggle.com/learn>
- 4. Google AI:** como aprender a plataforma de inteligência artificial – IA (*artificial intelligence* – AI) do Google - <https://ai.google/build>
- 5. FreeCodeCamp:** curso de Python com ML -
<https://www.freecodecamp.org/learn/machine-learning-with-python/>
- 6. Stanford CS224N:** Aprendizado profundo (*deep learning* – DL) para processamento de linguagem natural (*natural language processing* NLP) -
<https://web.stanford.edu/class/cs224n/>
- 7. Fast.ai:** DL prático para programadores - <https://course.fast.ai/>
- 8. DataCamp:** tutoriais gratuitos de Python e R - <https://www.datacamp.com/tutorial>
- 9. Canais do Youtube:**
 - **Sentdex:** tutoriais Python e ML - <https://www.youtube.com/user/sentdex>
 - **Corey Schafer:** Programação Python - <https://www.youtube.com/user/schafer5>
- 10. MIT OpenCourseWare:** introdução à DL - <https://ocw.mit.edu/courses/6-s191-introduction-to-deep-learning-january-iap-2020/>
- 11. TinyMLEdu:** iniciativa aberta de treinamento em TinyML, com plataforma Edge Impulse - <https://tinyml.seas.harvard.edu/>
- 12. Deep Learning Book:** plataforma online, em português, abordando conceitos de redes neurais, IA, ML, DL, em constante evolução [3] -
<https://www.deeplearningbook.com.br/>

3. CURSOS IA NVIDIA

8 Excelentes cursos gratuitos da Nvidia para entrar no universo da IA [3].

- 1. Construindo um cérebro em 10 minutos:** Entenda como as redes neurais se baseiam na biologia e na psicologia e usam dados para aprender. Duração: 10 min. Disponível em: <https://courses.nvidia.com/courses/course-v1:DLI+T-FX-01+V1/>
- 2. Primeiros passos com IA na Jetson Nano:** Use o Jetson Nano Developer Kit da Nvidia para construir um modelo de classificação e regressão de aprendizado profundo. Duração: 8 horas. Disponível em:

https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+S-RX-02+V2.

3. IA generativa explicada: Conheça os conceitos, aplicações, ameaças e oportunidades da IA Generativa. Duração: 2 horas. Disponível em: <https://courses.nvidia.com/courses/course-v1:DLI+S-FX-07+V1/>.

4. Introdução à Ethernet: Entenda como os dados são encaminhados em uma rede Ethernet. Duração: 1 hora. Disponível em: <https://academy.nvidia.com/en/course/introduction-to-networking/?cm=51347>.

5. Desenvolva ferramentas avançadas de layout 3D: Crie ferramentas de layout 3D personalizadas usando o código NVIDIA Omniverse e o script Python. Duração: 90 min. Disponível em: <https://courses.nvidia.com/courses/course-v1:DLI+S-OV-07+V1/>

6. Uma Introdução Ainda Mais Fácil para CUDA: Descubra como melhorar o desempenho do seus aplicativos CUDA C/C++ por sobreposição de memória transferências de e para a GPU com cálculos na GPU. Duração: 1 hora. Disponível em: <https://courses.nvidia.com/courses/course-v1:DLI+T-AC-01+V1/>

7. Criação de aplicativos de IA de vídeo: Aprenda aplicativos de análise inteligente de vídeo (*Intelligent Video Analytics* – IVA) e use-os para extrair percepções (*insights*) de fluxos (*streams*) de vídeo usando análise de aprendizagem profunda. Duração: 8 horas. Disponível em: <https://courses.nvidia.com/courses/course-v1:DLI+S-IV-02+V2/>

8. Monte um Robô Simples: Monte um robô móvel de duas rodas e aprenda a construir seus próprios projetos de simulação de robôs. Duração: 30 min. Disponível em: <https://courses.nvidia.com/courses/course-v1:DLI+T-OV-01+V1/>.

4. LIVROS GRATUITOS [5]

1. HAGAN, M. T.; Howard, B. D. Neural Network Design, Marting Hagan, 2º Edition, 1012 p., 2002. ISBN-13: 978-0971732117. Disponível em: <https://hagan.okstate.edu/NNDesign.pdf>

2. ROSA, J. L. G. R. Artificial Neural Networks – Models and Applications, In-Tech, 416 p., 2016. ISBN-13: 978-9535127055. Disponível em: https://mts.intechopen.com/storage/books/5191/authors_book/authors_book.pdf
3. LUGER, George F. Tradução de Daniel Vieira. Inteligência Artificial. 6a edição, Pearson, 631p, 2014. Disponível em: <https://livrariapublica.com.br/livros/inteligencia-artificial-george-f-luger/>
4. MEHLIG, B. Machine Learning with Neural Networks, University of Gothenburg, Sweden, 241 p., 2021. Disponível em: <https://arxiv.org/pdf/1901.05639.pdf>
5. MOHRI, M; ROSTAMIZADEH, A; TALWALKAR, A. Foundations of Machine Learning, MIT Press, 2º edition, 505 p., 2018. learning. ISBN: 9780262039406. Disponível em: <https://www.dropbox.com/s/38p0j6ds5q9c8oe/10290.pdf?dl=1>
6. GOODFELLOW, I; Bengio, Y; COURVILLE, A. Deep Learning. MIT Press, 2016. Disponível online: <http://www.deeplearningbook.org>
7. ABE, Jair Minoro. Aspectos de Computação Inteligente Paraconsistente, Instituto de Estudos Avançados da USP. São Paulo, 2013. Disponível em: <http://www.iea.usp.br/pesquisa/grupos-anteriores/logica-e-teoria-da-ciencia/publicacoes/E-book%20Aspectos%20de%20Computacao%20Inteligente%20Paraconsistente.pdf>
8. SIMÕES, Marcelo Godoy; SHAW, Ian S. Controle e modelagem fuzzy. Editora Blucher, 2007. Disponível online: https://www.academia.edu/29968068/Controle_e_modelagem_fuzzy_shaw_simoes
9. HALTERMAN, R. L. Fundamentals of Python Programming, Southern Adventist University, 669 p., 2018. Draft: Disponível em: https://folk.ntnu.no/sversti/INGG1001-H2019/pythonbook_20191015.pdf
10. LEE, K. D. Python Programming Fundamentals, 2º edition, Springer, 241 p., 2014. ISBN: 978-1-4471-6642-9. Disponível em: https://aulavirtual.fio.unam.edu.ar/pluginfile.php/149985/mod_resource/content/1/2014_Book_PythonProgrammingFundamentals.pdf

REFERÊNCIAS

- [1] KUMAR, S. Your Roadmap to Becoming a Machine Learning Engineer, Machine Learning, Artificial Intelligence, Deep Learning, Computer Vision, Robotics, DataOps, GenAI (Linkedin Group), 2024. Disponível em: https://media.linkedin.com/dms/image/v2/D4D22AQExq5t58x_47Q/feedshare-shrink_800/feedshare-shrink_800/0/1728034523017?e=1730937600&v=beta&t=oMNTmPJW5JLwMogyP0fTxQGEW9IvbHsRxlfkdo-B5fo. Acessado em Set 04, 2024.

- [2] ALAM A. The Best Free Resources to Master Machine Learning. Art of Data Science, 2024. Disponível em:
https://media.licdn.com/dms/image/D5622AQHbFxS0qniQ/feedshare-shrink_800/0/1720438696292?e=1723075200&v=beta&t=4Rulg0quCep9a4wOBjMtPHPqxV4DI-AN58jGc_E02Js. Acessado em Julho 08, 2024.
- [3] DSA. Deep Learning Book, Data Science Academy (DSA), 2024. Disponível em: <<https://www.deeplearningbook.com.br/>>. Acessado em Julho 8, 2024.
- [4] CARVALHO JUNIOR, A. 8 Excelentes cursos gratuitos da NVIDIA, para entrar no universo da Inteligência Artificial. EAILab Posts, Abril 2, 2024. Disponível em: <https://eailab.labmax.org/2024/04/02/8-excelentes-cursos-gratuitos-da-nvidia-para-entrar-no-universo-da-inteligencia-artificial/>. Acessado em Jul 8, 2024.
- [5] CARVALHO JUNIOR, A. Hot Links – Free Ebooks, EAILab, 2024. Disponível em: <<https://eailab.labmax.org/hot-links/>>. Acessado em Jul 8, 2024.

18. ALGORITMOS DE APRENDIZADO DE MÁQUINA

Adaptado por: Dr. Arnaldo de Carvalho Junior

Data: Agosto 19, 2024.

1. EXPLORANDO ALGORITMOS DE ML

O aprendizado de máquina (*machine learning* – ML) é um campo vasto, abrangendo uma ampla gama de algoritmos e técnicas. A Figura 1 apresenta uma visão geral abrangente:

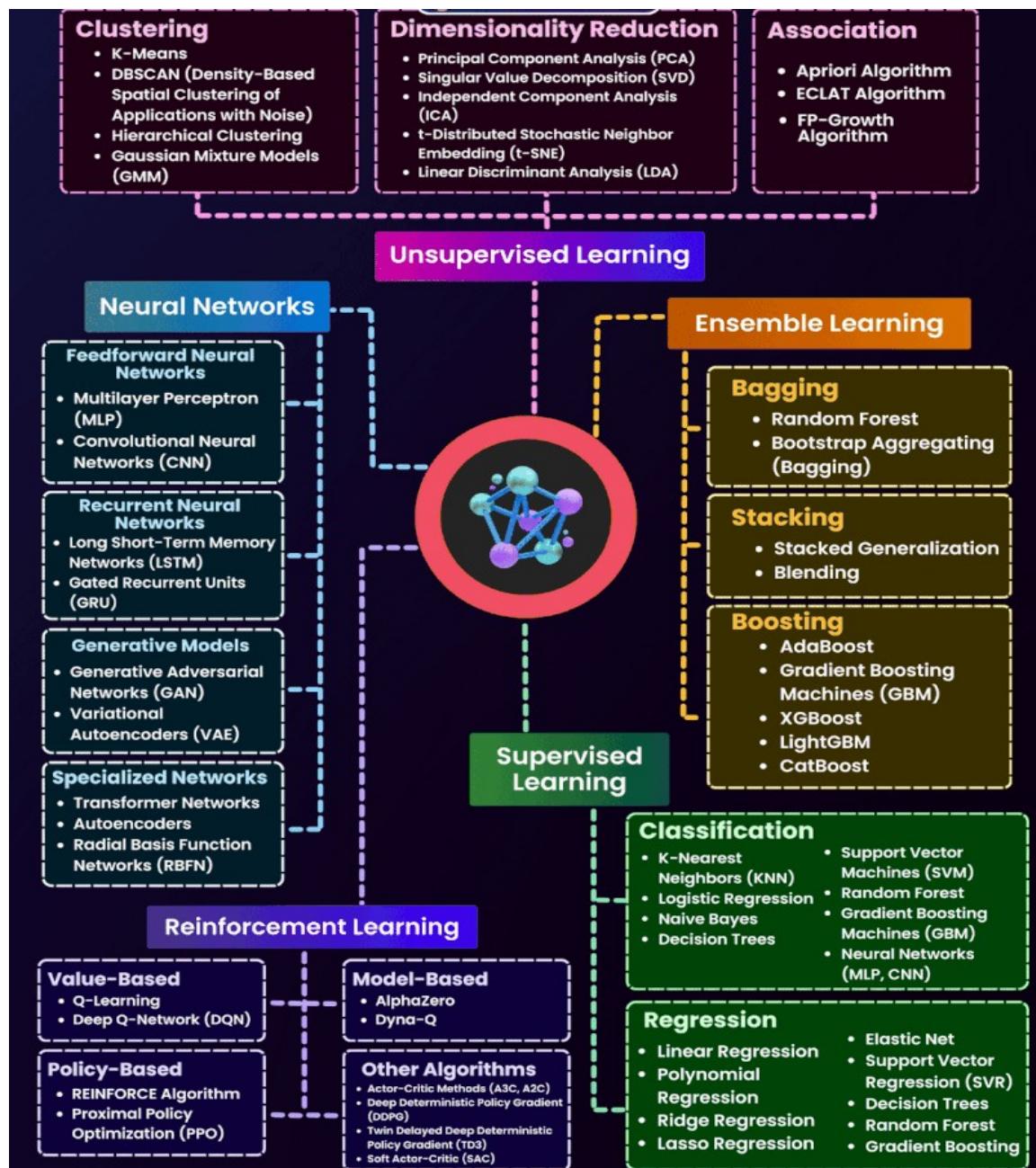


Figura 1 - Algoritmos de Aprendizado de Máquina.

Fonte: Adaptado de [1].

2. DEZ TÉCNICAS FUNDAMENTAIS DE ML

A Figura 2 apresenta 10 técnicas fundamentais de ML.

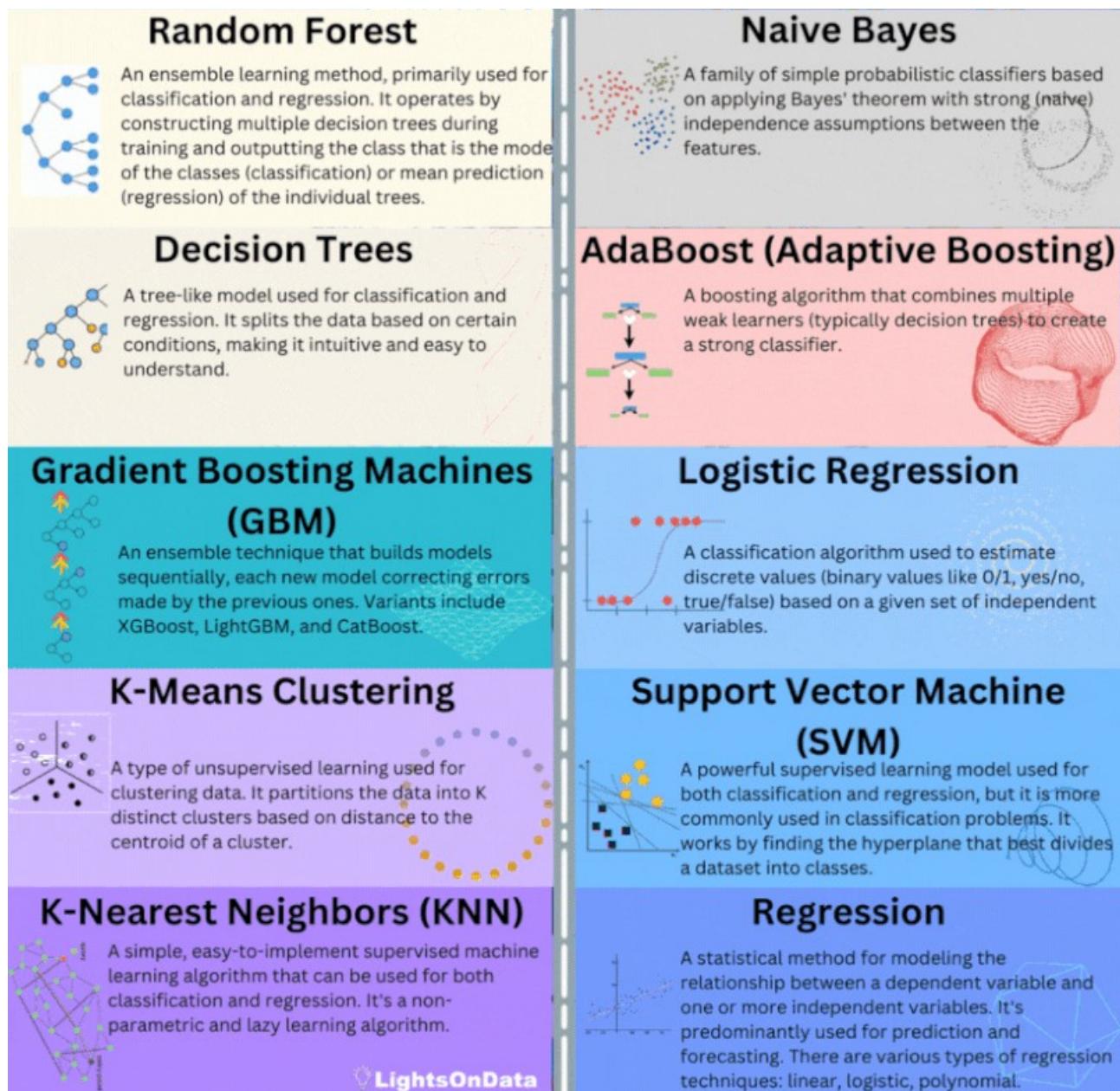


Figura 2 - Algoritmos Frequentemente Utilizados em de Aprendizado de Máquina.

Fonte: Adaptado de [2].

À medida que se navega pelo complexo cenário da ciência de dados e da inteligência artificial (IA), é crucial entender os algoritmos principais que alimentam as aplicações modernas de aprendizado de máquina.

As dez técnicas fundamentais que formam a espinha dorsal de informações orientadas a dados são:

2.1. Floresta aleatória (*random forest*): Um conjunto de árvores de decisão que melhora a precisão e reduz o excesso de ajuste, geralmente usado na análise de estoque e na detecção de fraude. Método de conjunto que alavanca a sabedoria de várias árvores de decisão, oferecendo desempenho robusto nas tarefas de classificação e regressão. Sua força está em mitigar o excesso de ajuste através da tomada de decisão coletiva.

2.2. Bayes ingênuos (*Naive Bayes*): Com base no teorema de Bayes, esse algoritmo é perfeito para tarefas que assumem a independência do recurso, como filtragem de spam e análise de sentimentos. Enraizada na teoria probabilística, esse algoritmo se destaca na classificação de texto e na filtragem de spam. Sua eficiência decorre da suposição de independência de recursos, permitindo treinamento e implantação rápidos.

2.3. Árvores de decisão (*Decision Trees*): Esses modelos intuitivos fornecem processos transparentes de tomada de decisão, tornando-os inestimáveis tanto para modelagem preditiva quanto para análises explicativas em contextos de negócios.

2.4. Reforço Adaptativo (*Adaptive Boosting - Adaboost*): Um algoritmo pioneiro de impulso que melhora iterativamente o desempenho do modelo, concentrando -se em instâncias classificadas incorretamente, demonstrando o poder do aprendizado de conjunto no tratamento de conjuntos de dados complexos.

2.5. Máquinas de reforço de gradiente (*Gradient Boosting Machine - GBM*): Conhecidas pela correção de erro sequencial, os GBMs são ótimos para detecção de anomalias e previsão de rotatividade de clientes. É uma técnica avançada de conjunto que cria modelos sequencialmente para corrigir erros, oferecendo desempenho de ponta em vários domínios, desde finanças até assistência médica.

2.6. Técnicas de Regressão: Melhor para prever valores contínuos, como preços da casa e previsão de salários, assumindo uma relação linear entre variáveis. Dos modelos

lineares a polinomiais, a análise de regressão forma a base da modelagem preditiva, oferecendo informações sobre relacionamentos variáveis e tendências de previsão.

2.7. Regressão Logística: Uma ferramenta preferida para tarefas de classificação binária, como filtragem de email e diagnóstico de doenças. Apesar de sua simplicidade, esse algoritmo continua sendo uma pedra angular da classificação binária, fornecendo resultados interpretáveis e saídas probabilísticas cruciais para avaliação de riscos e análise de limite de decisão.

2.8. Cluster de K-Means: Este algoritmo partitiona dados em *clusters* k, úteis para segmentação de mercado e agrupamento de documentos. Uma abordagem de aprendizado não supervisionada essencial para a segmentação de mercado, detecção de anomalia e descoberta de padrões em espaços de dados de alta dimensão.

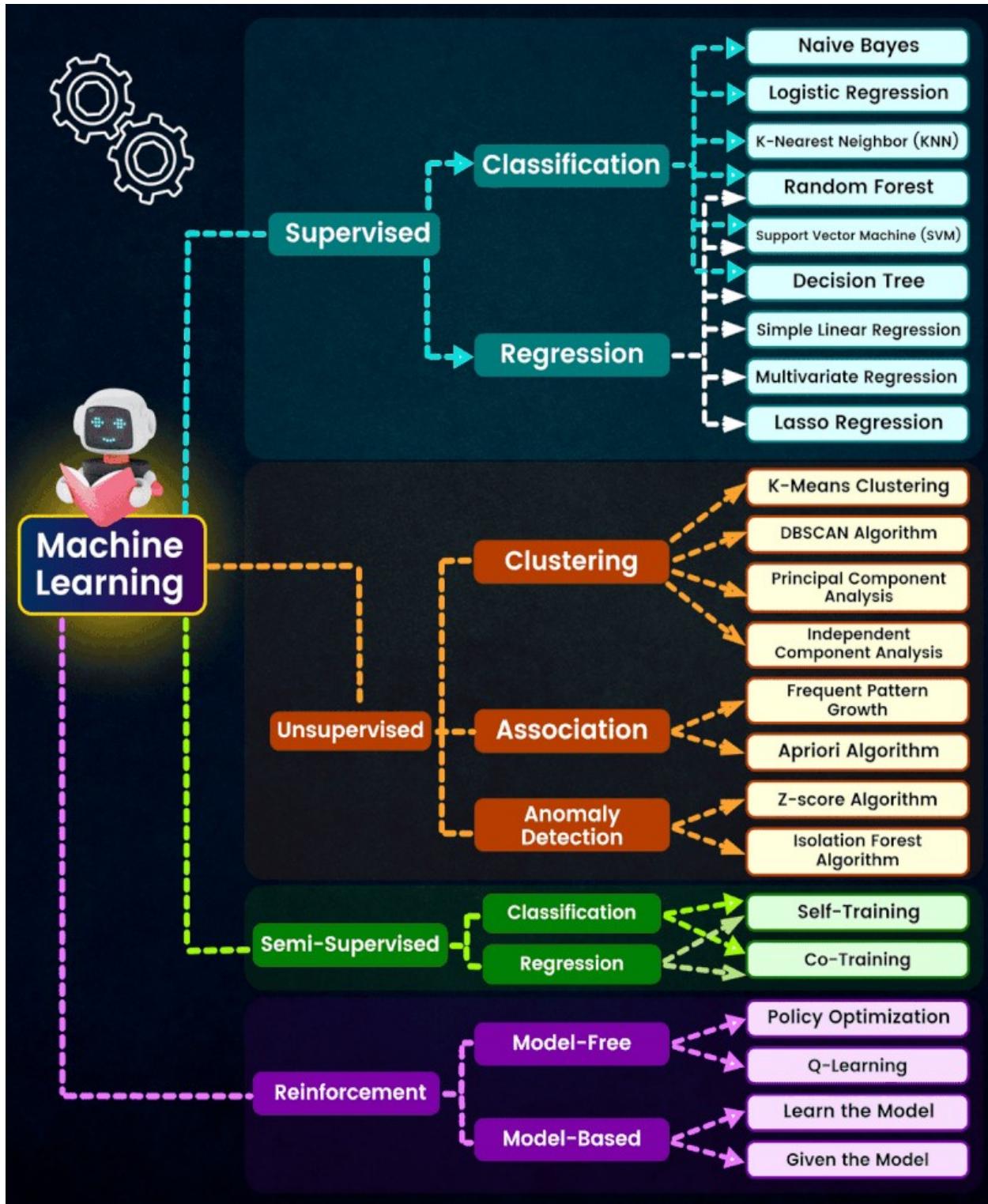
2.9. Máquina de Vetor de Suporte (Support Vector Machine - SVM): Conhecida por sua eficácia em espaços de alta dimensão, a capacidade da SVM de definir hiperplanos ótimos o torna indispensável na classificação de imagens e na bioinformática.

2.10. Vizinhos Mais Próximos (K Nearest Neighbours - KNN): Um algoritmo intuitivo baseado na similaridade, geralmente usado em sistemas de recomendação e detecção de manuscrito. O KNN é um método versátil e não paramétrico que brilha nos sistemas de recomendação e tarefas de reconhecimento de padrões, alavancando o princípio de que os pontos de dados semelhantes se agrupam.

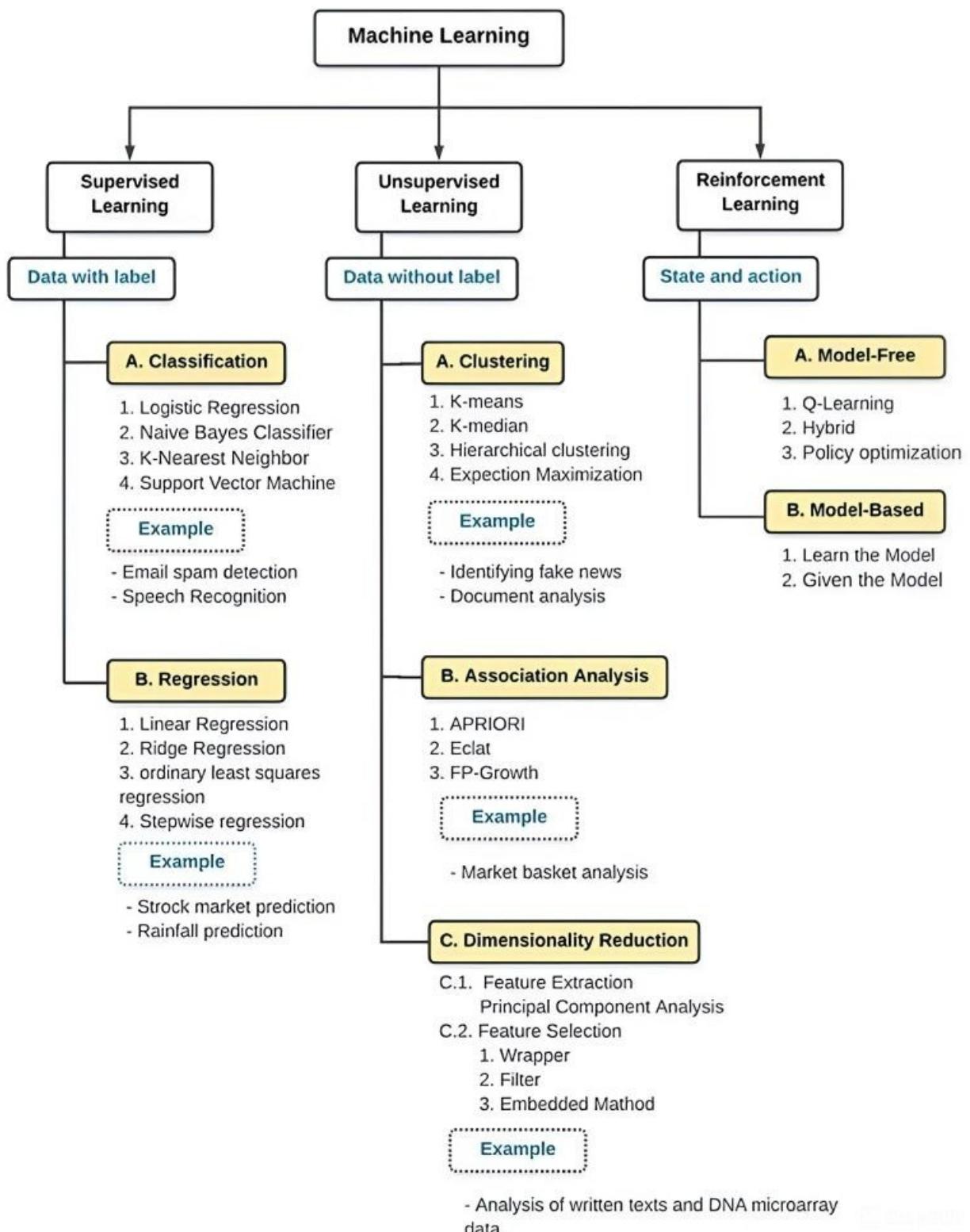
3. TIPOS DE ALGORITMOS DE ML

O domínio desses algoritmos capacita os cientistas de dados a extrair *insights* significativos, impulsionar a inovação e resolver desafios de negócios complexos.

À medida que a sociedade continua a ultrapassar os limites da IA, uma profunda compreensão dessas técnicas fundamentais permanece primordial. Segue uma relação detalhada para orientar a compreensão do leitor, conforme Figura 3.



(a)



(b)

Figura 3 – Diferentes Tipos de Algoritmos de Aprendizado de Máquina.
Fonte: Adaptado de [3].

3.1. Aprendizagem Não Supervisionada:

a) Clustering: técnicas para segmentação de dados e reconhecimento de padrões. Exemplos de aplicações: são essenciais para identificar padrões em dados não marcados, como detecção de notícias falsas.

- K-Means, agrupamento espacial de aplicações com ruído baseado em densidade (*density-based spatial clustering of applications with noise* – DBSCAN).
- Clustering hierárquico, modelos de mistura gaussiana (*gaussian mixture model* – GMM),

b) Redução de Dimensionalidade: análise do componente principal (*principal component analysis* - PCA), decomposição de valor singular (*singular value decomposition* - SVD), análise de componentes independentes (*independent components analysis* - ICA), incorporação de vizinhos distribuídos em T (*t-distributed neighbor embedding* - t-SNE), análise discriminante linear (*linear discriminant analysis* – LDA). São usadas para otimizar conjuntos de dados complexos para análise, aplicáveis nos dados de microarrays de DNA.

- O PCA é ideal para redução de dimensionalidade e recursos, sendo comumente utilizado na compressão da imagem.

c) Associação: Métodos para Análise de Mercado e Sistemas de Recomendação.

Exemplos de aplicações: são usadas para tarefas como a análise da cesta de mercado.

- xx) Algoritmo Apriori, Algoritmo de agrupamento de classes de equivalência e travessia de rede de baixo para cima (*equivalent class clustering and bottom-up lattice transversal* – ECLAT)
- yy) Algoritmo de crescimento padrão de frequência (*frequency pattern* – FP).

3.2. Aprendizagem Supervisionada:

a) Classificação: K-vizinhos mais próximos (*k-nearest neighbor* - KNN), regressão logística, Naive Bayes, árvores de decisão, máquinas de vetores de suporte (*support vector machine* - SVM), floresta aleatória, máquinas de reforço de gradiente (*gradient boosting machine* - GBM), redes neurais perceptron multicamadas (*multilayer perceptron* - MLP), convolucionais (*convolutional neural networks* – CNN). Estes algoritmos ajudam em tarefas como detecção de spam de e-mail e reconhecimento de fala.

- b) Regressão:** regressão linear, regressão polinomial, regressão Ridge, egressão Lasso, rede elástica, regressão de vetor de suporte (*support vector regression* - SVR), Árvores de Decisão (*decision trees*), Floresta Aleatória (*random forest*), crescimento de gradiente (*gradient boosting*). Essas técnicas são essenciais para o mercado de ações e previsão de chuvas.
- c) Modelos lineares:** algoritmos-chave para modelagem preditiva.
- d) Modelos baseados em árvores:** ferramentas poderosas para classificação e regressão.

3.3. Aprendizagem em conjunto:

- a) Agregação (*Bagging*):** Floresta Aleatória, Inicialização da Agregação (*bootstrap bagging*);
- b) Empilhamento (*stacking*):** generalização empilhada, combinação (*blending*);
- c) Impulso (*Boosting*):** AdaBoost, GBM, XGBoost, LightGBM, CatBoost.

3.4. Redes Neurais Artificiais (*artificial neural networks* - ANN):

- a) Redes Neurais Artificiais:** são algoritmos inspirados no cérebro humano. A ANN se destaca no reconhecimento e previsão de padrões, utilizando várias camadas ocultas para o processamento complexo de dados. Existem várias arquiteturas. As mais comuns são as do tipo Feedforward, Perceptron Multicamadas (MLP), Redes Neurais Recorrentes (*recurrent neural networks* – RNN) e Redes Neurais Convolucionais (*convolutional neural networks* - CNN);
- b) Redes Neurais Recorrentes (*recurrent neural networks* - RNN):** Redes de Memória Longa e de Curto Prazo (*long short-term memory* - LSTM), Unidades Recorrentes Fechadas (*gated recurrent units* - GRU);
- c) Modelos Generativos:** redes adversariais gerativas (*generative adversarial networks* - GAN), autocodificadores variacionais (*variational autoencoders* - VAE);
- d) Redes especializadas:** redes de transformadores, auto-codificadores, redes de funções de base radial (*radial basis function network* – RBFN).

3.5. Aprendizagem por Reforço:

Desde métodos sem modelos, como Q-learning até abordagens baseadas em modelos, esse campo permite que os sistemas aprendam e otimizem por meio de interações.

- a) Baseado em valor:** aprendizado Q (*Q-learning*), rede Q profunda (*deep Q-network* - DQN);
- b) Baseado em Políticas:** Algoritmo REINFORCE, Otimização de Política Proximal (*proximal policy optimization* - PPO);
- c) Baseado em modelo:** AlphaZero, Dyna-Q;
- d) Outros algoritmos:** métodos ator-crítico assíncrono (*asynchronous advantage actor critic* – A3C) e síncrono (*advantage actor critic* - A2C), gradiente de política determinística profunda (*deep deterministic policy gradient* - DDPG), gradiente de política determinística profunda com atraso duplo (*twin delayed deep deterministic policy gradient* - TD3), ator crítico suave (*soft actor-critic* – SAC).

Esteja o pesquisador iniciando no aprendizado de máquina ou precisa aprimorar, a Figura 3 é super útil. Ela abrange algoritmos importantes em vários estilos de aprendizagem e fornece dicas práticas para o desempenho de primeira linha.

Compreender esses algoritmos é crucial para quem deseja se aprofundar no aprendizado de máquina e em suas aplicações. Cada categoria oferece abordagens e ferramentas exclusivas para lidar com diferentes tipos de dados e problemas, conforme Figuras 4,5 e 6.

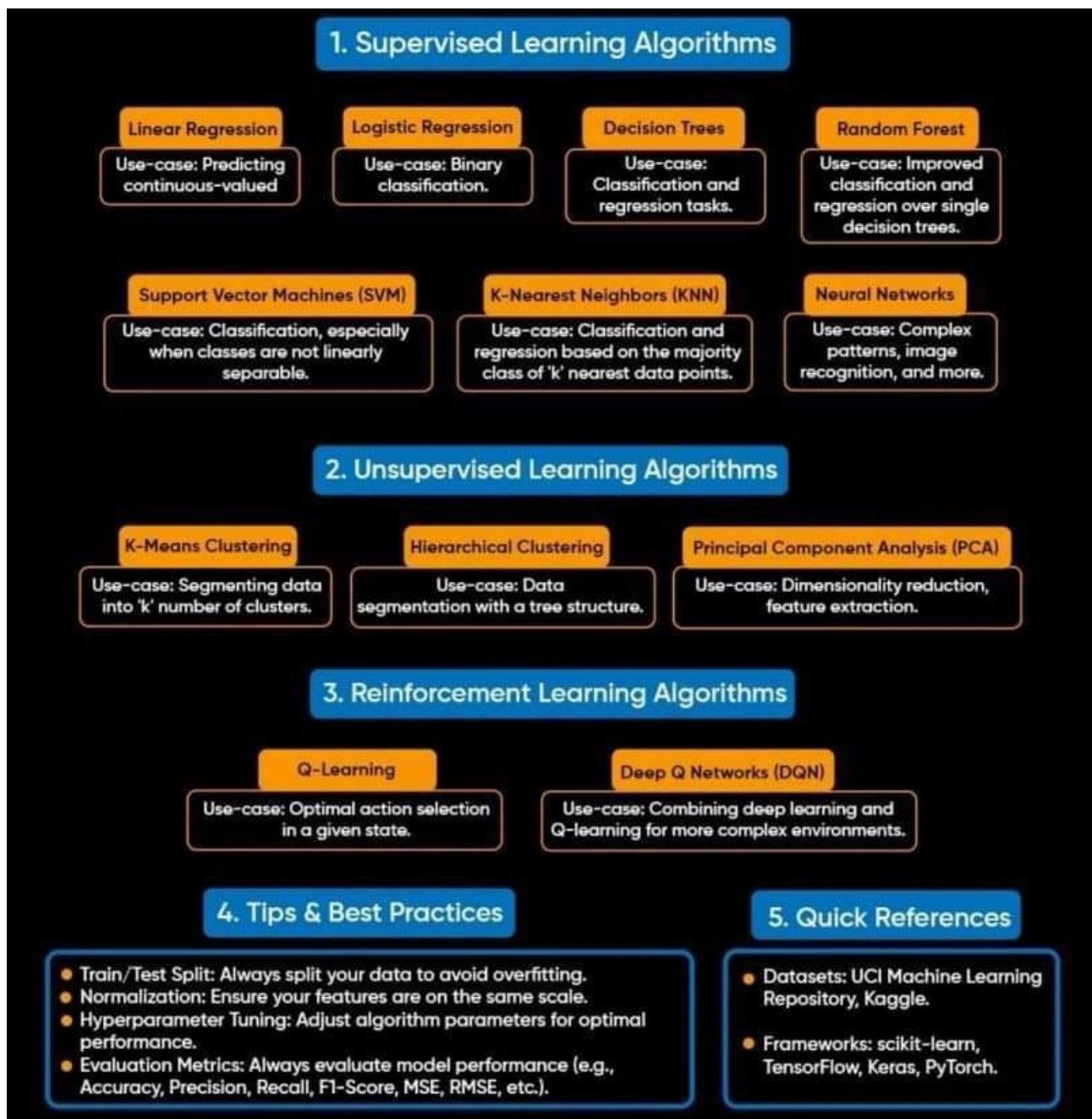


Figura 4 – Folha de Dicas de Aprendizado de Máquinas

Adaptado de [4].

Supervised Learning	Linear Models	Linear Regression	A simple algorithm modeling relationships between dependent and independent variables using a straight line.	Use Cases	<ul style="list-style-type: none"> Predicting housing market trends. Forecasting company sales revenue.
		Logistic Regression	Models binary outcomes by estimating probabilities using a logistic function, suitable for classification tasks.	Use Cases	<ul style="list-style-type: none"> Binary disease outcome prediction. Classifying email as spam.
		Ridge Regression	Linear regression with L2 regularization to prevent overfitting by penalizing large coefficients.	Use Cases	<ul style="list-style-type: none"> Handling multicollinear predictors. Regularized model for sales forecasting.
		Lasso Regression	Linear regression with L1 regularization to encourage sparsity by penalizing absolute coefficient values.	Use Cases	<ul style="list-style-type: none"> Sparse modeling in genetics. Identifying key predictive features.
Tree-Based Models	Decision Tree	Decision Tree	A model that splits data into branches based on feature values for classification or regression.	Use Cases	<ul style="list-style-type: none"> Loan approval predictions. Classifying customer purchasing behaviors.
		Random Forests	An ensemble of decision trees that votes for the most popular class or average prediction in regression.	Use Cases	<ul style="list-style-type: none"> Feature selection and importance ranking. Credit scoring and risk assessment.
	Gradient Boosting Regression	Gradient Boosting Regression	Ensembles weak predictive models, like decision trees, iteratively minimizing errors to boost performance.	Use Cases	<ul style="list-style-type: none"> Fraud detection in transactions. Accurate energy demand forecasting.
	XGBoost	XGBoost	Optimized gradient boosting algorithm that enhances performance and scalability through regularization and parallel processing.	Use Cases	<ul style="list-style-type: none"> Optimizing fraud detection algorithms. Boosting credit risk predictions.
	LightGBM Regressor	LightGBM Regressor	A fast, gradient boosting framework for large datasets using histogram-based learning to enhance efficiency.	Use Cases	<ul style="list-style-type: none"> Efficient large-scale data processing. Ranking results in search engines.
Unsupervised Learning	Clustering	K-Means	Clusters data into k groups by minimizing variance within each group through iterative centroid updates.	Use Cases	<ul style="list-style-type: none"> Customer market segmentation analysis. Image compression and simplification.
		Hierarchical Clustering	Groups data into a tree of clusters using either agglomerative or divisive methods based on distances.	Use Cases	<ul style="list-style-type: none"> Organizing document collections hierarchically. Hierarchical grouping of genes.
		Gaussian Mixture Models	Probabilistic models representing data as mixtures of several Gaussian distributions for clustering or density estimation.	Use Cases	<ul style="list-style-type: none"> Clustering mixed-type data points. Detecting anomalies in datasets.
Association	Aprori Algorithm	Aprori Algorithm	A method for finding frequent itemsets and association rules in large datasets by iteratively identifying item correlations.	Use Cases	<ul style="list-style-type: none"> Market basket association analysis. Recommending frequently bought products.

Figura 5 - Principais Algoritmos de Aprendizado de Máquina.

Fonte: Adaptado de [5].

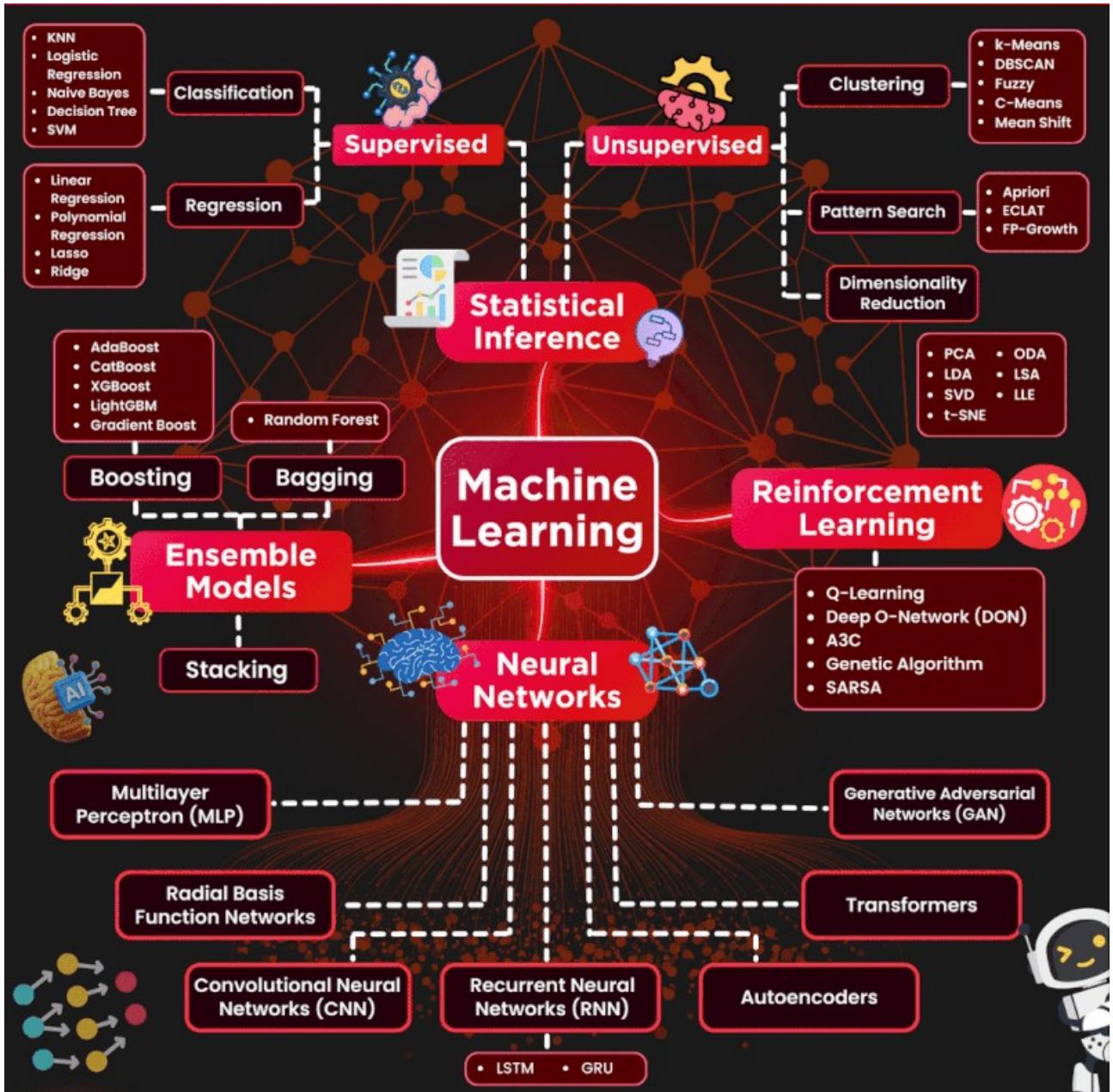


Figura 6 - Algoritmos de Modelos de Aprendizado de Máquina

Fonte: Adaptado de [6].

4. APRENDIZADO SUPERVISIONADO OU NÃO-SUPERVISIONADO?

Escolher a abordagem certa de aprendizado de máquina (*machine learning* – ML) pode tornar o projeto um sucesso ou total fracasso. Se o pesquisador estiver confuso entre o aprendizado supervisionado e não supervisionado, a Figura 7 apresenta uma divisão com exemplos de algoritmos e aplicações típicas em biomedicina, para ajudar o pesquisador a tomar uma decisão melhor informado!

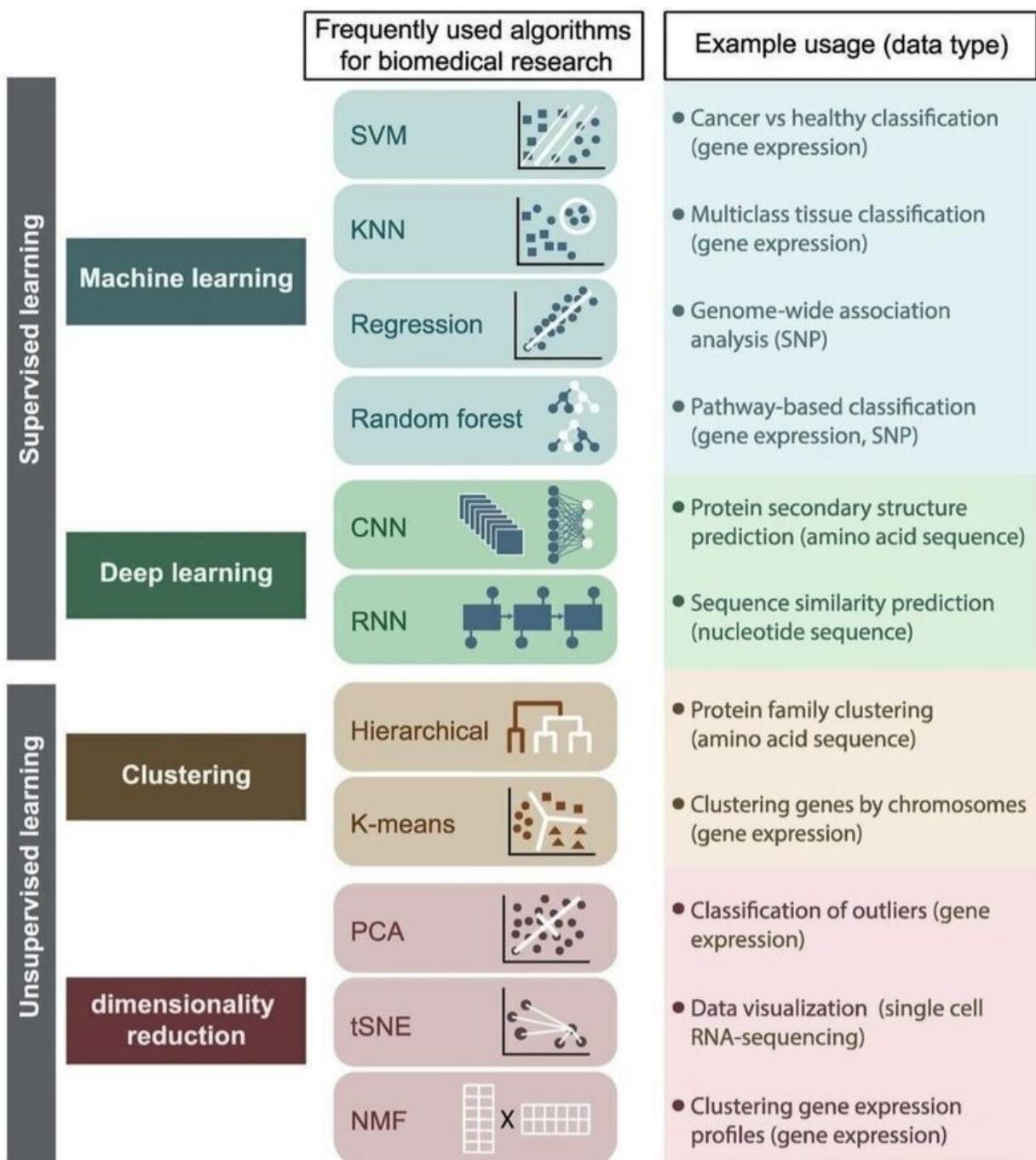


Figura 7 – Algoritmos de ML e aplicações típicas em biomedicina.

Fonte: Adaptado de [1].

5. RESUMO

Mergulhar no mundo vibrante do aprendizado de máquina (*machine learning* – ML) é como se tornar um explorador digital. Cada algoritmo e modelo é uma paisagem distinta, rica em oportunidades e ideias. A Figura 8 apresenta um resumo dos algoritmos de ML.

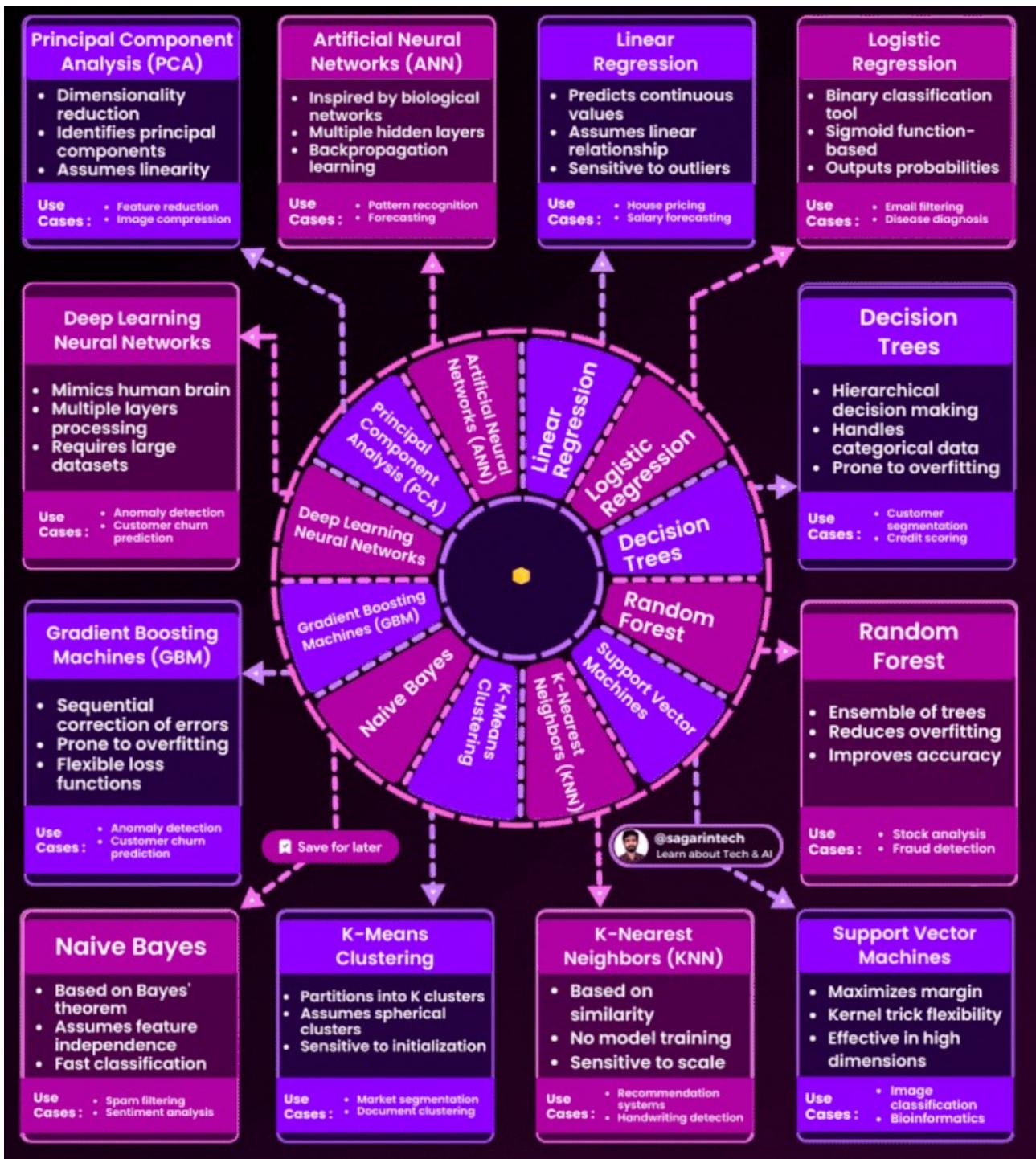


Figura 8 – Em resumo, os algoritmos mais utilizados em ML.

Fonte: Adaptado de [8].

O aprendizado supervisionado é o guia do desenvolvedor, oferecendo instruções claras com dados rotulados para nos ajudar a alcançar previsões precisas com eficiência. Pode-se imaginar uma visita guiada pelas complexidades da inteligência artificial (IA). É aqui que o pesquisador tem sinalizações (*signposts*) claras ou rótulos (*labels*), e um mapa

(dados). É perfeito quando os objetivos são definidos e o pesquisador precisa de resultados precisos. Deve-se Pensar nisso como se houvesse um tutor pessoal para o modelo de IA do projeto.

A **aprendizagem não supervisionada** convida o desenvolvedor a traçar o desconhecido, descobrindo estruturas e padrões ocultos sem caminhos predefinidos. Pode-se imaginar uma aventura no deserto dos dados. Não há mapa e o pesquisador é o explorador, encontrando tesouros e padrões ocultos por conta própria. Este é o objetivo para cavar as estruturas intrínsecas dos dados e revelar o inesperado.

O **aprendizado de reforço** transforma a jornada em uma experiência interativa, onde cada decisão leva a recompensas que guiam nossas ações futuras.

As **redes neurais** são companheiras de viagem versáteis, capazes de navegar em terrenos complexos, desde decifrar imagens até a compreensão de idiomas.

Os **modelos de conjunto** reúnem a sabedoria coletiva de vários algoritmos, levando a decisões mais robustas e confiáveis.

As **CNNs** ajudam o desenvolvedor a focar nos detalhes nas imagens, identificando padrões e recursos com precisão incrível.

As **RNNs** tecem sequências de dados, prevendo o próximo elemento em uma série ou a nota futura em uma melodia com precisão.

Os **Transformadores (transformers)** fornecem uma chave universal para a linguagem, oferecendo o poder de traduzir e resumir os diversos dialetos da expressão humana.

As **GANs** são a dupla desafiadora de Criador e Crítica, empurrando os limites da criatividade e da autenticidade.

Os **Autocodificadores (Autoencoders)** são como empacotadores especializados, condensando informações à sua essência e reconstruindo -as com alta-fidelidade, ideal para manuseio de dados eficientes.

Os **PCAs** são excelentes na redução de dimensionalidade e recursos, sendo ideais na compressão da imagem.

Cada modelo abre uma porta para um mundo de ideias, alimentando a inovação e a compreensão de maneiras novas e emocionantes. A compreensão desses conceitos principais capacita o pesquisador a aproveitar o potencial total dos dados, selecionando o paradigma de aprendizado apropriado. Seja o pesquisador um cientista de dados experiente ou no início da sua jornada, ele deve-se lembrar de que cada passo nesse domínio é um passo em direção ao futuro.

REFERÊNCIAS:

- [1] GUTIÉRRES, G. A. Machine Learning Algorithms. Disponível em: https://media.linkedin.com/dms/image/D5622AQHjtSbvWeZD-g/feedshare-shrink_800/0/1719497072314?e=1722470400&v=beta&t=dRv98oMQ6szY6IZZjFU7WEr3MfUObp3iwTtVh2eB7RE. Acessado em Junho 27, 2024.
- [2] CHANDRAGIRI, Frequently Used Machine Learning Algorithms. LightsonData, 2024. Disponível em: https://media.linkedin.com/dms/image/D5622AQGACUXLe_g9fw/feedshare-shrink_2048_1536/0/1721522561420?e=1725494400&v=beta&t=l8XXIDR5toiR2MunaH9soFs1_DqHsomAN4uF6iML84I. Acessado em Jul 30, 2024.
- [3] GUTIÉRRES, G. A. Different Types of Machine Learning Algorithms. Disponível em: https://media.linkedin.com/dms/image/D4E22AQFVcnlwdul6EA/feedshare-shrink_2048_1536/0/1719675363342?e=1723075200&v=beta&t=iFpQME_CIDMh8OfdPjRampgCLac93Td1coFzbfkTTI8. Acessado em Julho 08, 2024.
- [4] MAITY, P. Machine Learning Cheatsheet, Machine Learning Community (Linkedin Group), 2024. Disponível em: https://media.linkedin.com/dms/image/D5622AQHSYwQhx2f3Ww/feedshare-shrink_2048_1536/0/1722244721869?e=1724889600&v=beta&t=GosgJWE7y3wJgPOHi2jJhB9SXfqadxoqJIdKJz9tRLA. Acessado em Jul 29, 2024.
- [5] PANJUTA, D. Top Machine Learning Algorithms. AI Simplified, 2024. Disponível em: https://media.linkedin.com/dms/image/D4E22AQGi34WNA-YUEQ/feedshare-shrink_800/0/1720426087300?e=1723075200&v=beta&t=rYVBM-Kg_a7758O_ySjfOozm86jpZhXj9BzsrtUf0. Acessado em Jul 08, 2024.
- [6] PANJUTA, D. Machine Learning Algorithms and Models. AI Simplified, 2024. Disponível em: https://media.linkedin.com/dms/image/D5622AQFiHceRWYaBkA/feedshare-shrink_800/0/1713363339603?e=1724284800&v=beta&t=sH8pn7LL-EHUsQ-WZt0R0B9MVpsuTJAmd48CftnaGjU. Acessado em Jul 16, 2024.
- [7] DAKSH, K. Supervised vs Unsupervised Learning! Data Science and AI (Linkedin Group), 2024. Disponível em: <https://media.linkedin.com/dms/image/v2/D5622AQHILOGAWC8kyw/feedshare->

shrink_2048_1536/feedshare-shrink_2048_1536/0/1724761797540?
e=1727913600&v=beta&t=6eHIGE6Ay7MfAwWeQ-49zi3yOE4p96m_7cG8h_SAMSE.
Acessado em Ago 27, 2024.

[8] SAGARINTECH, Most Used Machine Learning Algorithms, Love Open Source Community (Linkedin Group), 2024. Disponível em:
https://media.licdn.com/dms/image/D5622AQEYLHsGTI5-Tw/feedshare-shrink_2048_1536/0/1722512275091?e=1725494400&v=beta&t=-0h9P2_ZLCTBBiyO2olvkgvuqdbmvxpkQwYHK_1fZPc.
Acessado em Ago 01, 2024.

19. A RODA DA INTELIGÊNCIA ARTIFICIAL E APRENDIZADO DE MÁQUINAS

Por: Denis Panjuta

Adaptado por: Dr. Arnaldo de Carvalho Junior

Data: Julho 16, 2024.

Visão Geral do Ecossistema da Tecnologia

Compreendendo o ecossistema de tecnologia Inteligência Artificial (IA) e Aprendizado de Máquinas (*Machine Learning* - ML): O infográfico da figura a seguir fornece uma visão geral detalhada das várias tecnologias que compõem a IA e o cenário de ML.

1. Áreas-chave:

1.1. Aprendizado de Máquina (ML): explora como os algoritmos aprendem com os padrões de dados para executar tarefas como sistemas de detecção e recomendação de fraude.

1.2. Aprendizagem Profunda: um subconjunto de ML que usa redes neurais para interpretar dados complexos, como imagens e fala.

1.3. Grandes Dados (Big Data): concentra-se no gerenciamento de conjuntos de dados extremamente grandes que ajudam na modelagem preditiva sofisticada.

1.4. Mineração de Dados (Data Mining): envolve a extração de padrões de grandes conjuntos de dados, aprimorando a tomada de decisões e o planejamento estratégico.

1.5. Análise Preditiva (Predictive Analytics): usa dados históricos para prever resultados futuros, aplicáveis em áreas como pontuação de crédito e gerenciamento de riscos.

1.6. Computação em Nuvem (Cloud Computing): essencial para armazenar e acessar grandes quantidades de dados pela Internet, suportando escalabilidade nos projetos de IA.

Cada segmento da roda apresentada no infográfico é interconectado, demonstrando a sinergia necessária para desenvolver soluções robustas de IA. Se você está envolvido no desenvolvimento de tecnologias de IA ou simplesmente curioso sobre como a IA afeta vários setores, entender esses elementos é crucial.

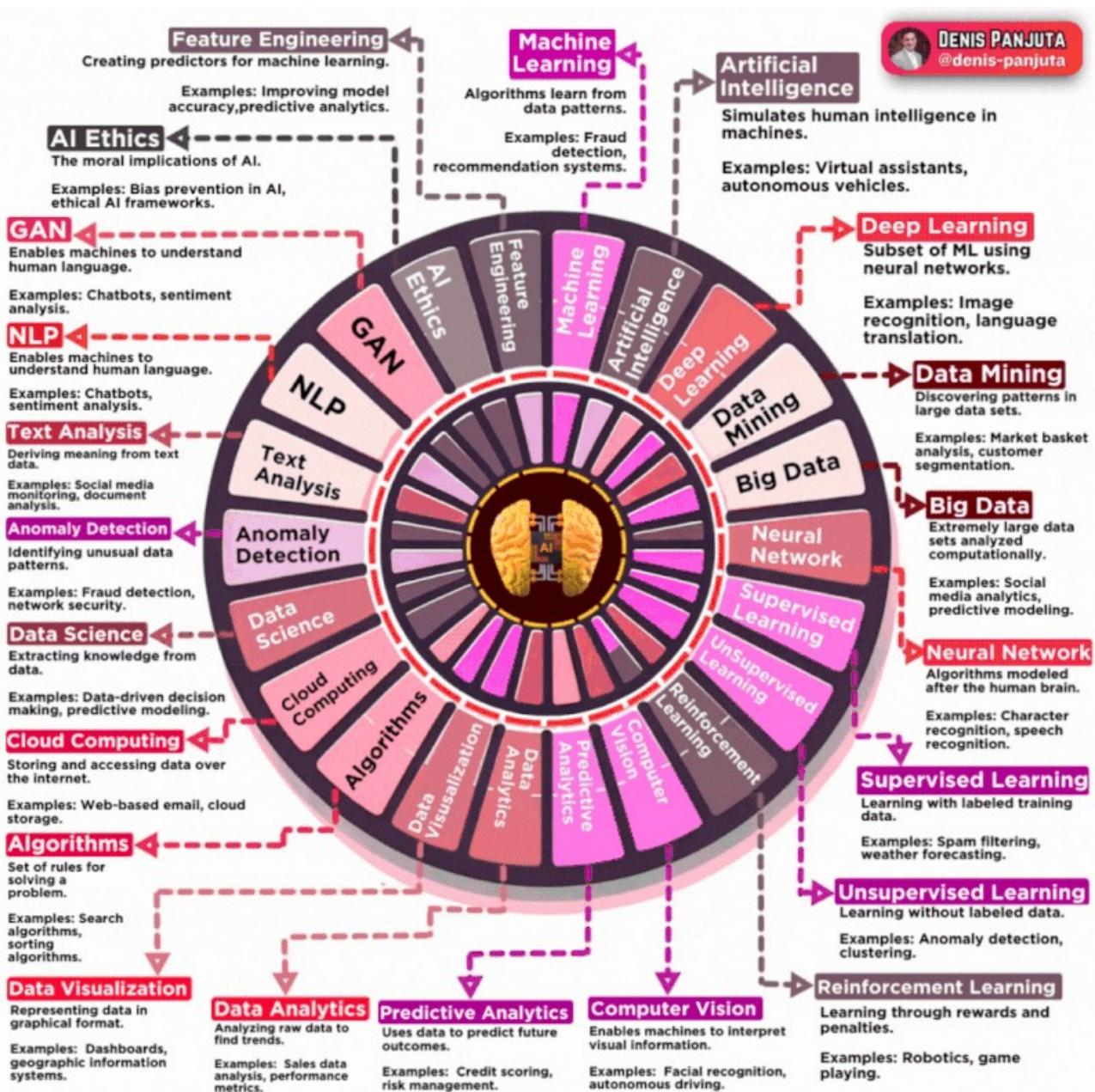


Figura 1 - Roda da IA/ML.

Fonte: Adaptado de [1].

2. REFERÊNCIA

- [1] PANJUTA, D. AI/ML Wheel – Technology Ecosystem Overview. Machine Learning Community (Moderate), 2024. Disponível em: https://media.linkedin.com/dms/image/D5622AQEm0qPZ68GE5g/feedshare-shrink_2048_1536/0/1715308560314?e=1724284800&v=beta&t=VWfKKVEPVpm4W34jcO_LHZohMCJeEMxuB2hLOhmtNQw. Acessado em Julho 16, 2024.

20. TODAS AS ESTRUTURAS DE DADOS SÃO APENAS VARIÁVEIS

Por: Hayder Zaeem

Adaptado por: Dr. Arnaldo de Carvalho Junior

Data: Julho 15, 2024.

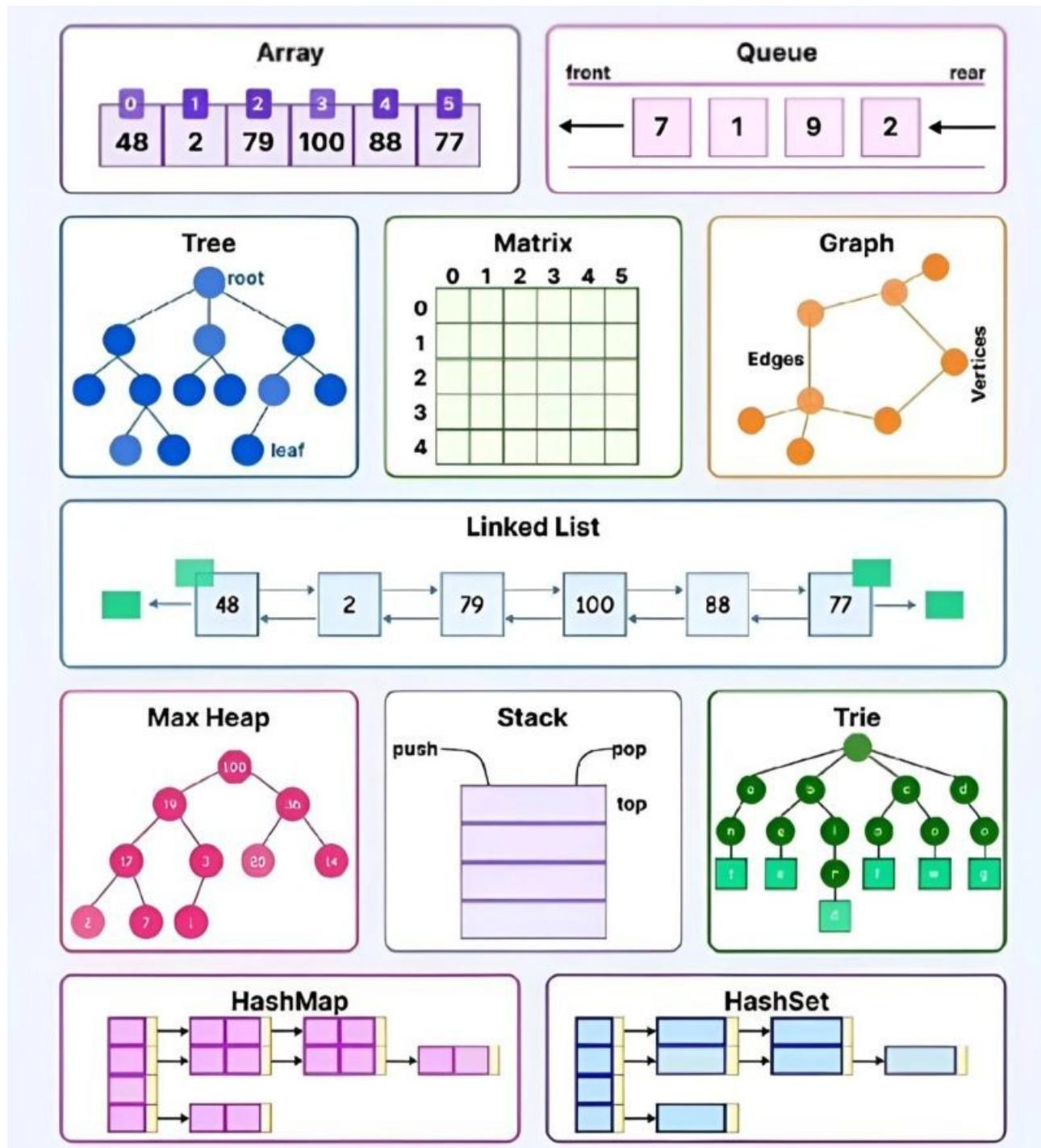


Figura 1 - Revelando a Verdade – Todas as Estruturas de Dados São Apenas Variáveis
Fonte: Adaptado de [1].

Todos sabem a importância das estruturas de dados, especialmente com o recente desenvolvimento da inteligência artificial (IA). A pedra angular da inteligência artificial está nos dados, mais especificamente, nas estruturas de dados.

Este artigo apresenta uma interessante visão que explica o significado das estruturas de dados, algo que com certeza a maioria das pessoas nunca encontrou antes. Este *insight* faz parte da pesquisa do autor que propõe um padrão global para projetar estruturas de dados. De uma maneira simples e fácil e não muito profunda, a ideia que se deseja transmitir é de que a compreensão das estruturas de dados não requer complexidade, dificuldade ou mágica. Por fim, eles são simplesmente variáveis. Vários tipos de estruturas de dados podem ser vistos como filosofias, métodos ou ideias na distribuição de dados em diferentes espaços de memória para gerenciá-los de maneira eficaz. No entanto, sua implementação e armazenamento dependem apenas de variáveis.

Os computadores não lidam com estruturas de dados como os seres humanos imaginam. Eles lidam apenas com variáveis. Os computadores não entendem matrizes, filas, pilhas, listas vinculadas, árvores e outras, etc. Em vez disso, eles gerenciam e manipulam dados usando variáveis sós.

As matrizes são feitas apenas de variáveis, o que significa de uma maneira muito simplificada em um nível muito baixo. Quando uma matriz é criada na linguagem de programação em que está escrita, é criada uma variável que contém o tamanho da matriz e outra variável contém um ponteiro ou variável que armazena o primeiro valor. Em seguida, ele assume as posições das próximas variáveis ao lado para armazenar os outros valores e determina o número de espaço ou variáveis após o uso do tamanho da matriz que foi especificada e salva como uma variável, como mencionamos. Dessa forma, foi formado um espaço completo de armazenamento contíguo. É claro que existem outros detalhes de baixo nível.

Em relação a estruturas de dados, como filas, pilhas, listas vinculadas, árvores, etc., o mesmo conceito se aplica. Algumas estruturas de dados utilizam matrizes em sua construção, enquanto em outras, as variáveis são usadas diretamente.

2. REFERÊNCIA

- [1] ZAEEM, H. Unveiling The Truth: All Data Structures Are Just Variables! Java Puzzlers, Java Champions, Java Developers, 2024. Disponível em: https://media.licdn.com/dms/image/D4E22AQHqhefyHGyWdg/feedshare-shrink_800/0/1720996788917?e=1723680000&v=beta&t=SgpZepWEFaR8mLGPB1E-prmHuImTMaRZrc1LdQxb2ek. Acessado em Julho 15, 2024.

21. A ATMOSFERA PYTHON

Por: Abdul Rahim Roni (Python Developers Community)

Adaptado por: Dr. Arnaldo de Carvalho Junior

Data: Julho 04, 2024.

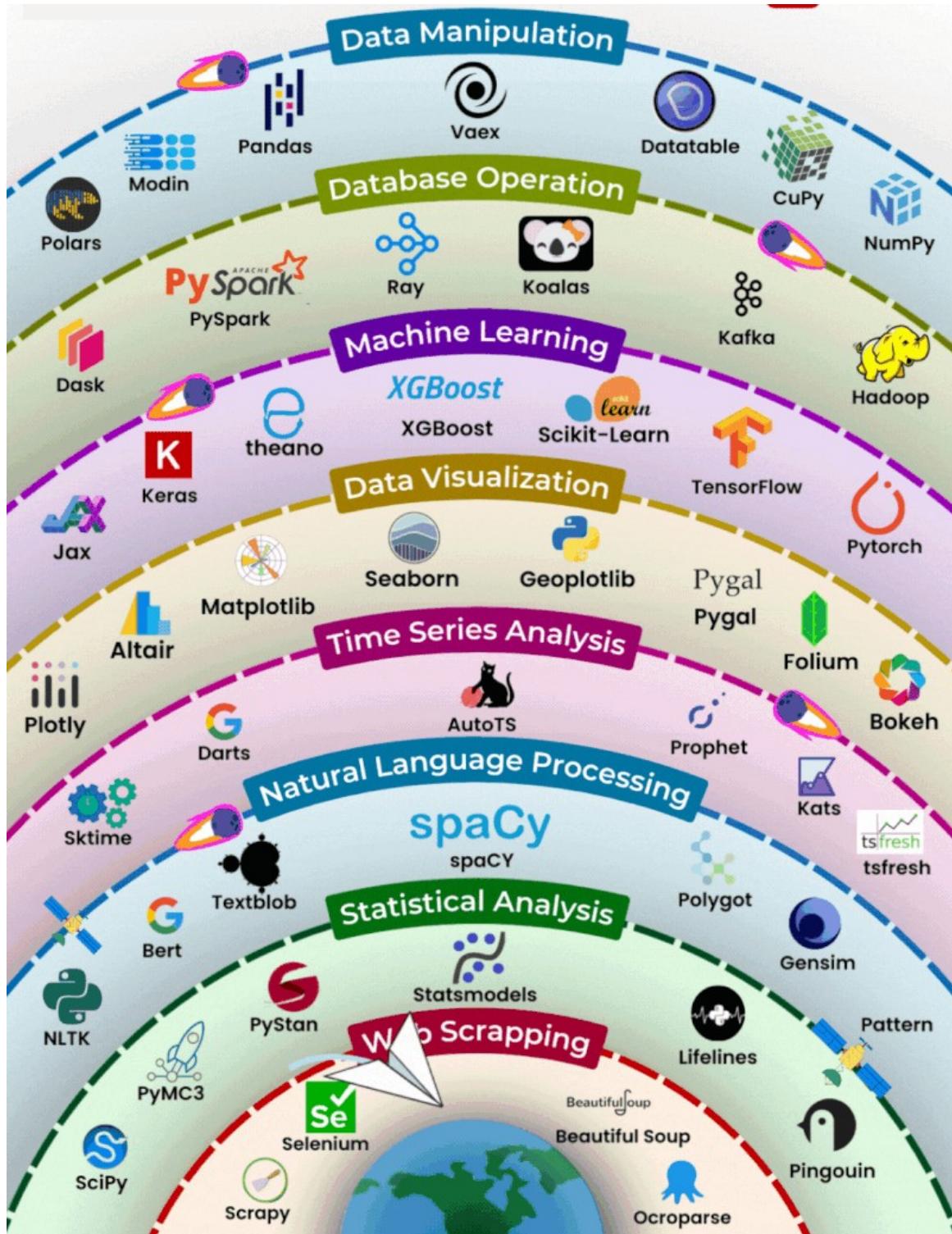


Figura 1 - A Atmosfera Python.

Fonte: Adaptado de [1].

Conhecer Python pode abrir um mundo de possibilidades interessantes.

Python é uma linguagem de programação versátil e amigável para iniciantes que encontra aplicações em uma ampla variedade de domínios, desde desenvolvimento web e análise de dados até aprendizado de máquina e inteligência artificial IA).

O ecossistema Python é incrivelmente rico e diversificado, com inúmeras bibliotecas, estruturas e ferramentas que atendem a vários casos de uso.

Esteja o pesquisador interessado em manipulação de dados (**pandas**, **numpy**), aprendizado de máquina (**scikit-learn**, **TensorFlow**) ou desenvolvimento web (**Django**, **Flask**), o Python tem o que ele precisa.

Investir tempo aprendendo Python pode aumentar significativamente as perspectivas de carreira e permitir que se resolva problemas complexos com facilidade.

Então, o pesquisador deve considerar mergulhar no mundo do Python e desbloquear seu potencial para agilizar seu trabalho e impulsionar a inovação em seus projetos.

A Atmosfera Python (de baixo para cima), conforme a Figura 1:

- Demolição de Web (*web scrapping*)
- Análise Estatística
- processamento de Linguagem Natural (*natural language processing* – NLP)
- Análise de Séries de Dados
- Visualização de Dados
- Aprendizado de Máquina (*machine learning* – ML)
- Operação de Base de Dados
- Manipulação de Dados

REFERÊNCIA

[1] RONI, A. R. The Python Atmosphere. Machine Learning Community, 2024. Disponível em: https://media.liecdn.com/dms/image/D5622AQFQ3moHDeEQwQ/feedshare-shrink_2048_1536/0/1719998782170?e=1723075200&v=beta&t=6Jt63xplNJ5bnq3j3glBTob37ja5P2l9yeMf8U8_j8s. Acessado em Julho 04, 2024.

22. O PODER DA ANÁLISE DE DADOS COM PYTHON

Por: Python Developers Community

Adaptado por: Dr. Arnaldo de Carvalho Junior

Data: Julho 02, 2024.

No cenário em constante evolução da ciência de dados, O **Python** continua a ser a pedra angular que une a inovação e a aplicação prática. À medida que a era das grandes dados, inteligência artificial (IA) e aprendizado de máquina (*machine learning* – ML) avançam, ter um kit de ferramentas robusto não é apenas uma vantagem – é uma necessidade!

Essa visualização abrangente resume a essência do que torna a codificação Python uma ferramenta indispensável no arsenal de cientistas e analistas de dados em todo o mundo. A seguir é apresentado um vislumbre do mundo multifacetado das bibliotecas Python que estão redefinindo a análise de dados:

- 1. Manipulação de dados:** Bibliotecas como **Pandas**, **NumPy** e **Polars** oferecem recursos incomparáveis para organização de dados, permitindo manipulação e transformação contínuas de conjuntos de dados complexos.
- 2. Visualização de dados:** de **Matplotlib** a **Seaborn** e **Plotly**, o Python fornece uma infinidade de ferramentas de visualização que transformam dados brutos em histórias perspicazes, tornando informações complexas acessíveis e envolventes.
- 3. Análise estatística:** **SciPy**, **PyStan** e **Statsmodels** estão na vanguarda da computação estatística, fornecendo métodos robustos para testes de hipóteses, análise de regressão e muito mais.
- 4. Análise de série temporal:** com bibliotecas como **PyFlux**, **Prophet** e **Darts**, a previsão de série temporal nunca foi tão acessível, oferecendo técnicas sofisticadas para análise de dados temporais.
- 5. Operações de banco de dados:** **Dask**, **PySpark** e **Hadoop** garantem o manuseio eficiente de dados em grande escala, permitindo computação escalonável e distribuída.
- 6. Web Scraping:** **BeautifulSoup**, **Scrapy** e **Selenium** são as ferramentas essenciais para extrair informações valiosas da web, impulsionando a tomada de decisões baseada em dados.

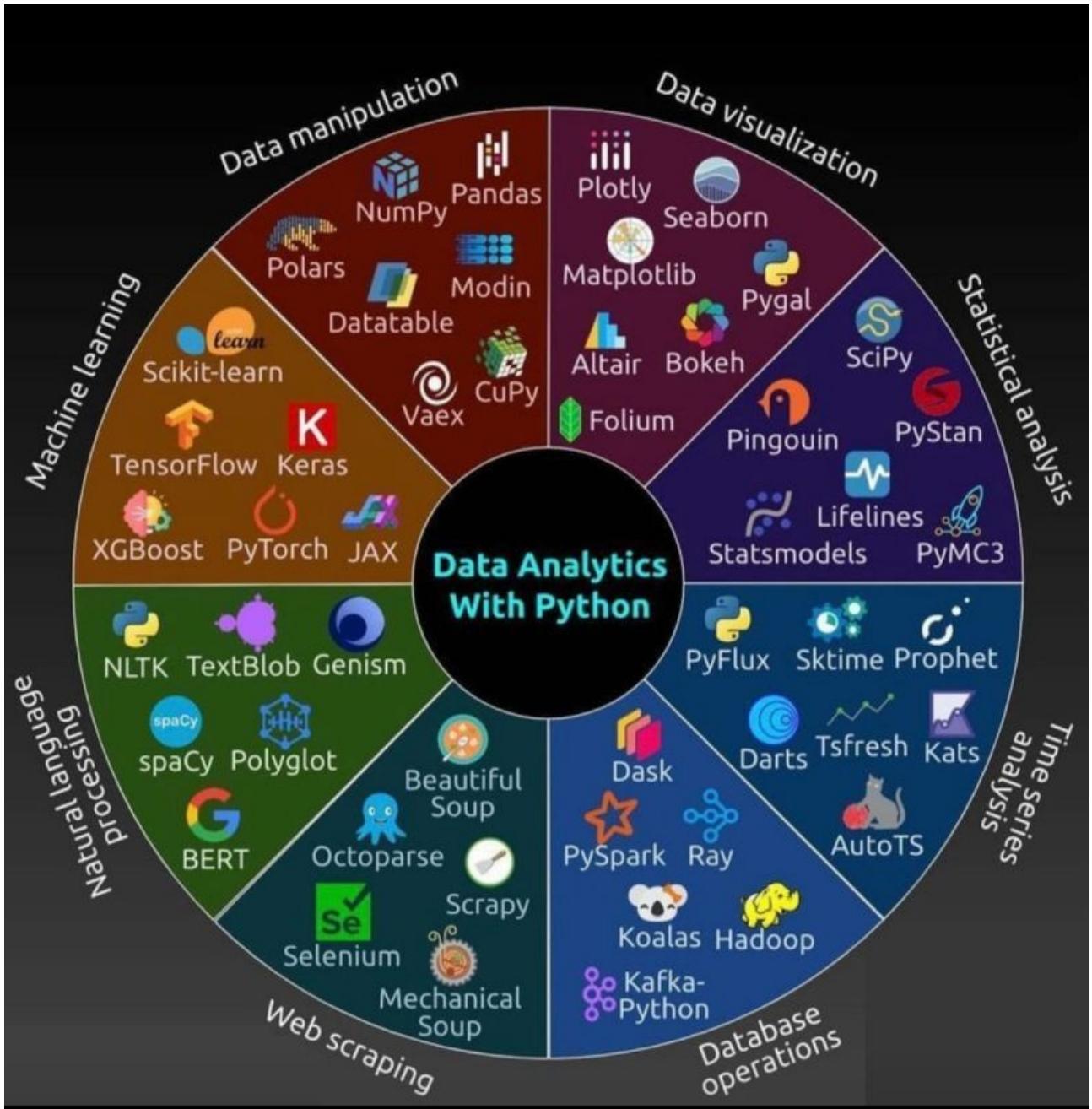


Figura 1 - Análise de Dados com Python.

Fonte: Adaptado de [1].

7. Processamento de linguagem natural (natural language processing - NLP): NLTK, *spaCy* e **BERT** revolucionam a análise de texto, capacitando aplicações desde análise de sentimentos até tradução de idiomas.

8. Aprendizado de máquina: com **TensorFlow**, **Scikit-learn** e **PyTorch**, o Python está no centro da revolução da IA, permitindo o desenvolvimento de modelos e algoritmos de ponta.

À medida que os limites do que é possível com os dados são ampliados, fica claro que a versatilidade e o extenso ecossistema do Python estão impulsionando o futuro da análise de dados. Essas ferramentas são a chave para desbloquear novas percepções (*insights*) e inovações.

Que a humanidade possa continuar aproveitando o poder do Python para transformar **dados** em **conhecimento** e este em **ação!**

REFERÊNCIA

- [1] PYTHON DEVELOPERS COMMUNITY. Data Analytics with Python. Disponível em: https://media.liecdn.com/dms/image/D4D22AQGYWcwt6QE8Mw/feedshare-shrink_2048_1536/0/1719898561178?e=1723075200&v=beta&t=0wLQO85zeNk1t2-0KXBd24Lf3y_Xbd0cRcoicQ4O4QA. Acessado em Julho 02, 2024.

23. ROTEIRO DO CIENTISTA DE DADOS

Adaptado por: Dr. Arnaldo de Carvalho Junior

Data: Julho 08, 2024.

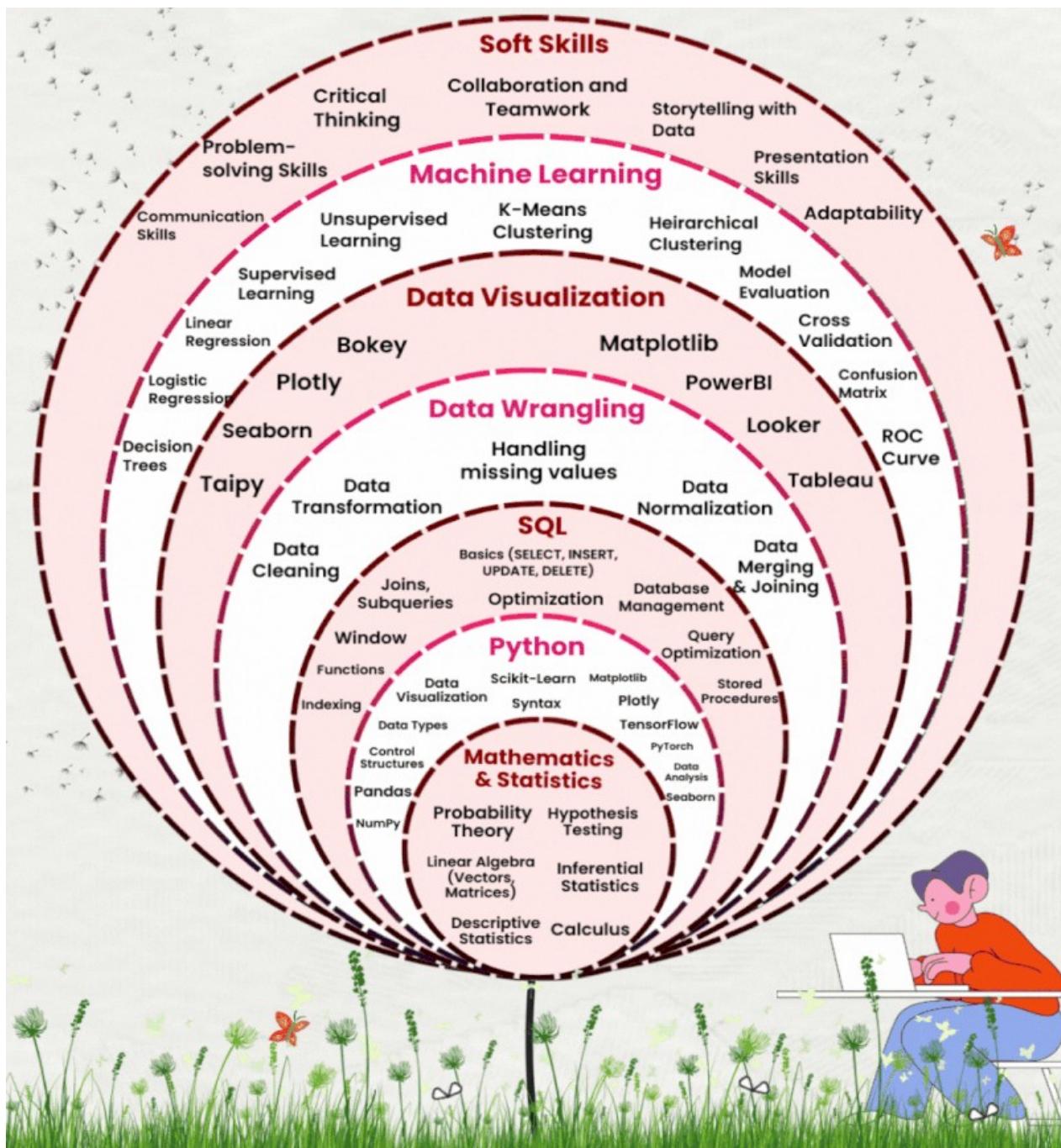


Figura 1 - Roteiro do Cientista de Dados.

Fonte: Adaptado de [1].

1. ROTEIRO DO CIENTISTA DE DADOS

1.1. Matemática e Estatística: Desenvolva uma sólida compreensão da teoria da probabilidade, teste de hipóteses, álgebra linear (vetores, matrizes), estatística inferencial, estatística descritiva e cálculo. Esses conceitos são fundamentais para análise de dados e algoritmos de aprendizado de máquina.

1.2. Python: proficiência de ganho no Python, incluindo funções, tipos de dados, estruturas de controle, pandas, numpy, scikit-learn, sintaxe, visualização de dados, matplotlib, plotay, tensorflow, pytorch, análise de dados e mareben. A linguagem Python é a espinha dorsal da programação da ciência de dados.

1.3. SQL: Master SQL Basics, como selecionar, inserir, atualizar, excluir, além de junções, subconsências, funções de janela, otimização de dados, gerenciamento de banco de dados, otimização de consultas e procedimentos armazenados. Essas habilidades são essenciais para trabalhar com grandes conjuntos de dados.

1.4. Divisão de dados (*Data Wrangling*): Aprenda a lidar com valores ausentes, transformação de dados, normalização de dados e limpeza de dados com eficiência. Essas tarefas são essenciais na preparação de dados para análise.

1.5 Visualização de dados: use ferramentas como bokey, plotly, SeaBorn, Taipy, Matplotlib, Powerbi, Looker e Tableau para criar visualizações atraentes que comunicam informações de maneira eficaz.

1.6. Aprendizado de máquina (*machine learning - ML*): mergulhe em aprendizado supervisionado, aprendizado não supervisionado, regressão linear, regressão logística, árvores de decisão, agrupamento de meanos, agrupamento hierárquico, avaliação de modelos, validação cruzada e curva ROC. O domínio dessas técnicas é fundamental para a construção de modelos robustos.

1.7. Habilidades sociais (*soft skills*): Aprimore suas habilidades de solução de problemas, pensamento crítico, habilidades de comunicação, colaboração e trabalho em equipe, narrativa com dados, habilidades de apresentação e adaptabilidade. Essas

habilidades sociais são vitais para colaborar com equipes e partes interessadas de maneira eficaz.

2. QUINZE RESPONSABILIDADES-CHAVE DO CIENTISTA DE DADOS

Conforme a Figura 2, são elas:



Figura 2 – As Quinze Responsabilidades do Cientista de Dados

Fonte: Adaptado de [2].

1. **Coleta de Dados:** Reúna dados de várias fontes, incluindo bancos de dados, APIs e conjuntos de dados externos. Isso envolve entender a arquitetura de dados e garantir a integridade dos dados.
2. **Limpeza de Dados:** Limpe e preprocessado dados brutos para prepará-los para análise. Isso inclui lidar com valores ausentes, corrigir inconsistências e remover duplicatas.
3. **Exploração de Dados:** Explore dados por meio de métodos estatísticos e técnicas de visualização para entender padrões e distribuições. Ferramentas como Pandas, Numpy e Matplotlib são comumente usadas.
4. **Engenharia de Recursos:** Crie e selecione Recursos significativos a partir de dados brutos para melhorar o desempenho do modelo. Isso envolve o conhecimento e a criatividade do domínio para transformar dados em entradas úteis.
5. **Construção do Modelo:** Desenvolva modelos preditivos usando algoritmos de aprendizado de máquina. Isso pode incluir regressão, classificação, agrupamento e muito mais.
6. **Avaliação do Modelo:** Avalie o desempenho dos modelos usando métricas como precisão, precisão, *recall* e escore F1. A avaliação ajuda a selecionar o melhor modelo e entender seus pontos fortes e fracos.
7. **Ajuste do Modelo:** Otimize os parâmetros do modelo para melhorar o desempenho. Técnicas como pesquisa de grade e validação cruzada são frequentemente usadas.
8. **Visualização de Dados:** Crie visualizações para apresentar informações de dados de forma clara e eficaz. Ferramentas como Tableau, Power BI e Bibliotecas Python como SeaBorn e Matplotlib são comumente usadas.

9. **Relatórios:** Prepare relatórios e painéis para comunicar as descobertas às partes interessadas. A comunicação clara e concisa é essencial para tomar decisões orientadas a dados.
10. **Colaboração:** Trabalhe com equipes multifuncionais, incluindo engenheiros, analistas e partes interessadas de negócios. A colaboração garante que as soluções de dados estejam alinhadas com as metas de negócios.
11. **Segurança de Dados:** Garanta a privacidade dos dados e a conformidade de segurança com regulamentos como o GDPR. Proteger dados sensíveis é uma responsabilidade crítica.
12. **Manter-se Atualizado:** Acompanhe as últimas tendências e tecnologias em ciência de dados. Aprendizagem e adaptação contínuas são essenciais neste campo em rápida evolução.
13. **Experimentação:** Execute experimentos para testar hipóteses e validar modelos. Essa abordagem científica ajuda a refinar modelos e melhorar a precisão.
14. **Automação:** Automatize os processos de dados e os fluxos de trabalho para melhorar a eficiência. Ferramentas como *scripts python* e sistemas de gerenciamento de fluxo de trabalho são úteis.
15. **Considerações Éticas:** Considere implicações éticas no uso e modelagem de dados. Garantir justiça, transparência e responsabilidade nas práticas de dados é crucial.

3. PAPEIS DO ENGENHEIRO DE SOFTWARE, ENGENHEIRO DE DADOS, CIENTISTA DE DADOS E ANALISTA DE DADOS

À medida que o cenário orientado a dados continua a evoluir, é essencial entender os papéis diversos, porém interconectados, que impulsionam a inovação e as ideias, conforme a Figura 3. Para os profissionais, conhecer as habilidades e responsabilidades

específicas de cada função ajuda a identificar as áreas para se concentrar no avanço ou na transição da carreira no campo de dados.

- Os **engenheiros de software** se concentram na criação de aplicativos e sistemas robustos, alavancando sua experiência em programação e design do sistema.
- Os **engenheiros de dados** criam e mantêm a infraestrutura necessária para a coleta, armazenamento e processamento de dados.
- Os **cientistas de dados** analisam e interpretam dados complexos, usando modelos estatísticos e técnicas de aprendizado de máquina para extrair *insights* valiosos.
- Os **analistas de dados** transformam dados em informações acionáveis, apoiando a tomada de decisões por meio de visualização e relatório de dados.

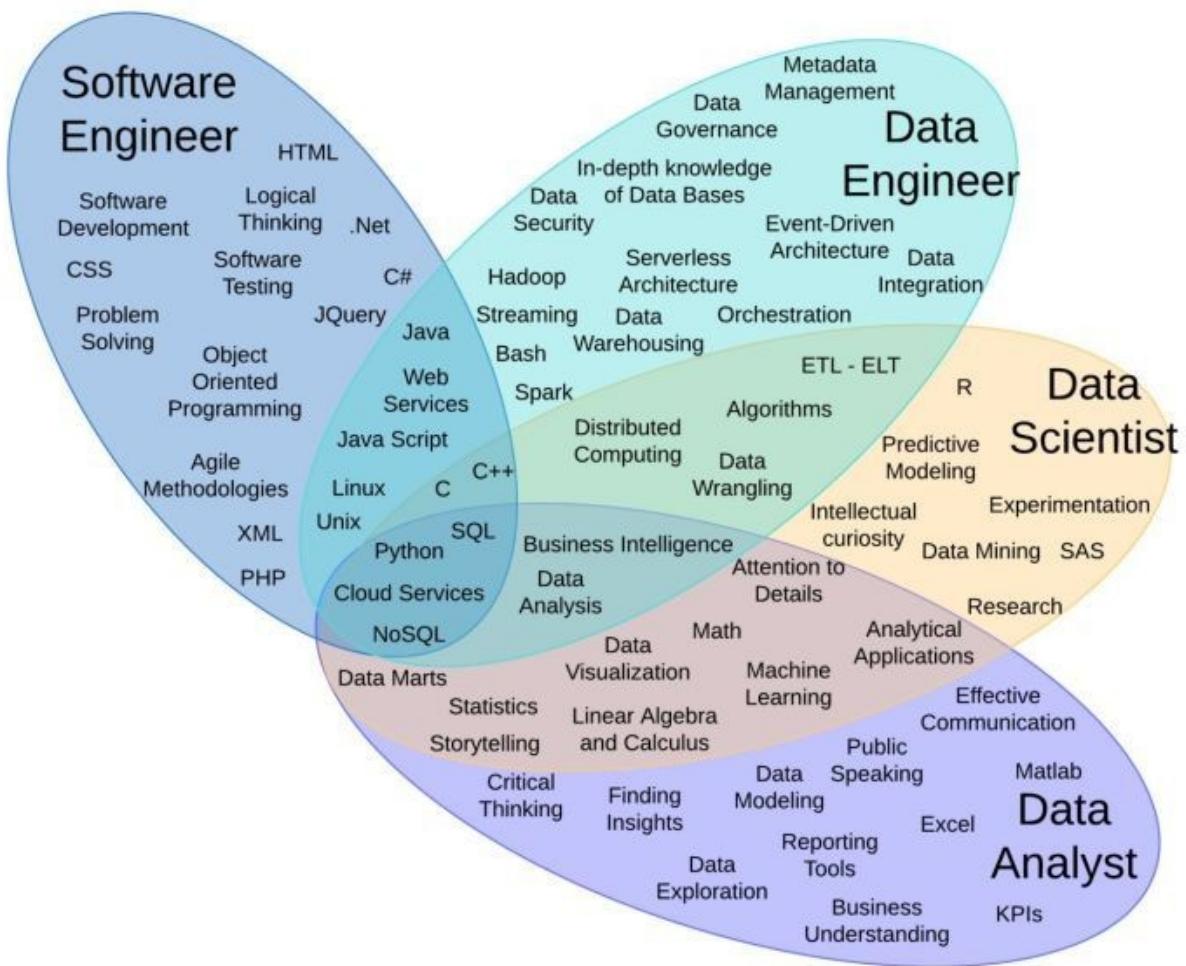


Figura 3 – Papel do Engenheiro de Software, Engenheiro de Dados, Cientista de Dados e Analista de Dados.

Fonte: Adaptado de [3].

Compreender essas funções e suas interseções pode melhorar a colaboração e impulsionar soluções mais eficientes e eficazes de dados. Ao identificar as habilidades certas para desenvolver, os profissionais podem navegar melhor em suas carreiras e atingir seus objetivos no ecossistema de dados.

REFERÊNCIA

- [1] PANJUTA, D. Data Scientist Roadmap. AI Simplified, 2024. Disponível em: https://media.linkedin.com/dms/image/D4E22AQFZOcBUeCQadQ/feedshare-shrink_800/0/1720188167601?e=1723075200&v=beta&t=DaTTdo2wsFSUSvV3ueySWJ0qa0LPZF0W0GnEiMTwo-8. Acessado em Julho 08, 2024.
- [2] ALAM, A. 15 Key Responsibilities of a Data Scientist, Linkedin Post, 2024. Disponível em: https://media.linkedin.com/dms/image/D5622AQGIzk8C46FZGw/feedshare-shrink_800/0/1721984608882?e=1725494400&v=beta&t=e_wcsiRgpsD7sEoTsWtk1NHEgoGcJo2x8MWSJGqeaWI. Acessado em Jul 30, 2024.
- [3] ANSARI, M. Z. Software Engineer vs Data Scientist vs Data Engineer vs Data Analyst. Machine Learning Community (Linkedin Group). Disponível em: [https://media.linkedin.com/dms/image/v2/D5622AQFLy107E852nA/feedshare-shrink_2048_1536/0/1724519499648?e=1727308800&v=beta&t=oM-Au-ljgeaPS-uSeTxMTwV4OhEJtqWGZfUM15i5aRg](https://media.linkedin.com/dms/image/v2/D5622AQFLy107E852nA/feedshare-shrink_2048_1536/feedshare-shrink_2048_1536/0/1724519499648?e=1727308800&v=beta&t=oM-Au-ljgeaPS-uSeTxMTwV4OhEJtqWGZfUM15i5aRg). Acessado em Agosto 26, 2024.

24. ONZE TIPOS DE VARIÁVEIS EM DATASET

Por: Avi Chawla

Editado por: Dr. Arnaldo de Carvalho Junior, Junho 16, 2024.

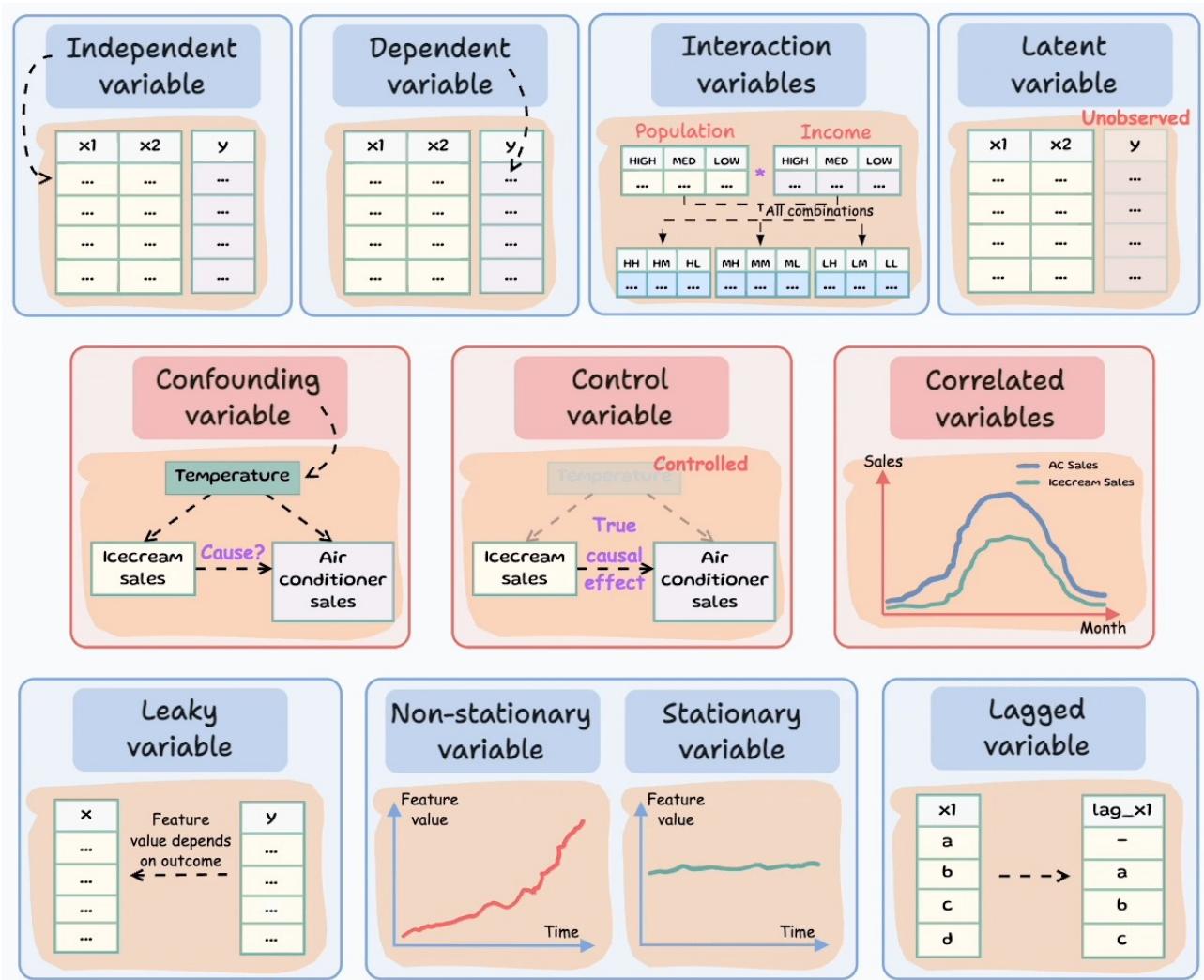


Figura 1 - 11 Tipos de variáveis em um Dataset.

Fonte: Adaptado de [1].

As colunas geralmente são categorizadas como um recurso ou um destino. No entanto, existem tantas variáveis que se pode encontrar/definir em um conjunto de dados.

1. Variáveis independentes: Recursos usados como entrada para prever o resultado.

2. Variável dependente: o resultado que está sendo previsto.

3. Variáveis de interação: representam o efeito de interação entre duas ou mais variáveis e são frequentemente usados na análise de regressão.

4. Variável latente: Uma variável que não é observada diretamente, mas é inferida de outras variáveis observadas. Por exemplo, usamos algoritmos de agrupamento porque os rótulos verdadeiros não existem e queremos inferi-los de alguma forma. O rótulo verdadeiro é uma variável latente neste caso.

5-7. Variáveis de confusão (*confounding variables*), correlatas e de controle: são normalmente observadas em um estudo de causa e efeito (inferência causal). Considere que se deseje medir o efeito das vendas de sorvetes nas vendas de ar-condicionado. Essas duas medidas estão altamente correlacionadas. No entanto, há uma variável de confusão, a temperatura, que influênciaria as duas medidas. Para estudar o verdadeiro impacto casual, deve-se garantir que a temperatura permaneça inalterada ao longo do estudo. Uma vez controlada, a temperatura se torna uma variável de controle. Mais formalmente, essas são variáveis que não são o foco principal do estudo, mas são cruciais para garantir que o efeito que se pretende medir não seja tendencioso ou confundido por outros fatores.

8. variável com vazamento (*leaky*): Variáveis que (sem querer) fornecem informações sobre a variável de destino, que não estariam disponíveis no momento da previsão.

9-10) Variáveis estacionárias e não estacionárias: Variáveis estacionárias são aquelas cujas propriedades estatísticas (média, variação) não mudam com o tempo. Preservar a estacionariedade na aprendizagem estatística é fundamental porque esses modelos dependem fundamentalmente da suposição de que as amostras são distribuídas de forma idêntica. Mas se a distribuição de probabilidade de variáveis estiver evoluindo ao longo do tempo (não estacionária), a suposição acima será violada. É por isso que, normalmente, o uso de valores diretos do recurso não estacionário (como o valor absoluto do preço das ações) não é recomendado. Em vez disso, sempre achei melhor definir recursos em termos de mudanças relativas: $\Delta P/P$.

11) Variável atrasada (*lagged*): Isso representa os valores dos pontos de tempo anteriores de uma determinada variável, mudando essencialmente a série de dados por um número especificado de períodos/linhas. Por exemplo, ao prever os números de vendas do próximo mês, pode-se incluir os números de vendas do mês anterior como uma variável atrasada.

REFERÊNCIAS

[1] CHAWLA, A. 11 types of variables in a dataset, explained visually. Daily Dose of Data Science, 2024. Disponível em:

https://media.liecdn.com/dms/image/D5622AQGJBWaFnHSEJg/feedshare-shrink_1280/0/1721131094442?e=1724284800&v=beta&t=VxpMnFBQml-Y-PFIAP1UslaBPmBpTqQaexVH2e-iZVc. Acessado em Julho 16, 2024.

25. FUNDAMENTOS DE ESTATÍSTICA

Adaptado por: Dr. Arnaldo de Carvalho Junior

Data: Agosto 21, 2024.

1. PENSAMENTO ESTATÍSTICO: CONCEITOS OBRIGATÓRIOS

A compreensão dos conceitos estatísticos é crucial para análise de dados, tomada de decisão e interpretação de resultados com precisão. A Figura 1 apresenta um resumo dos conceitos essenciais no pensamento estatístico.

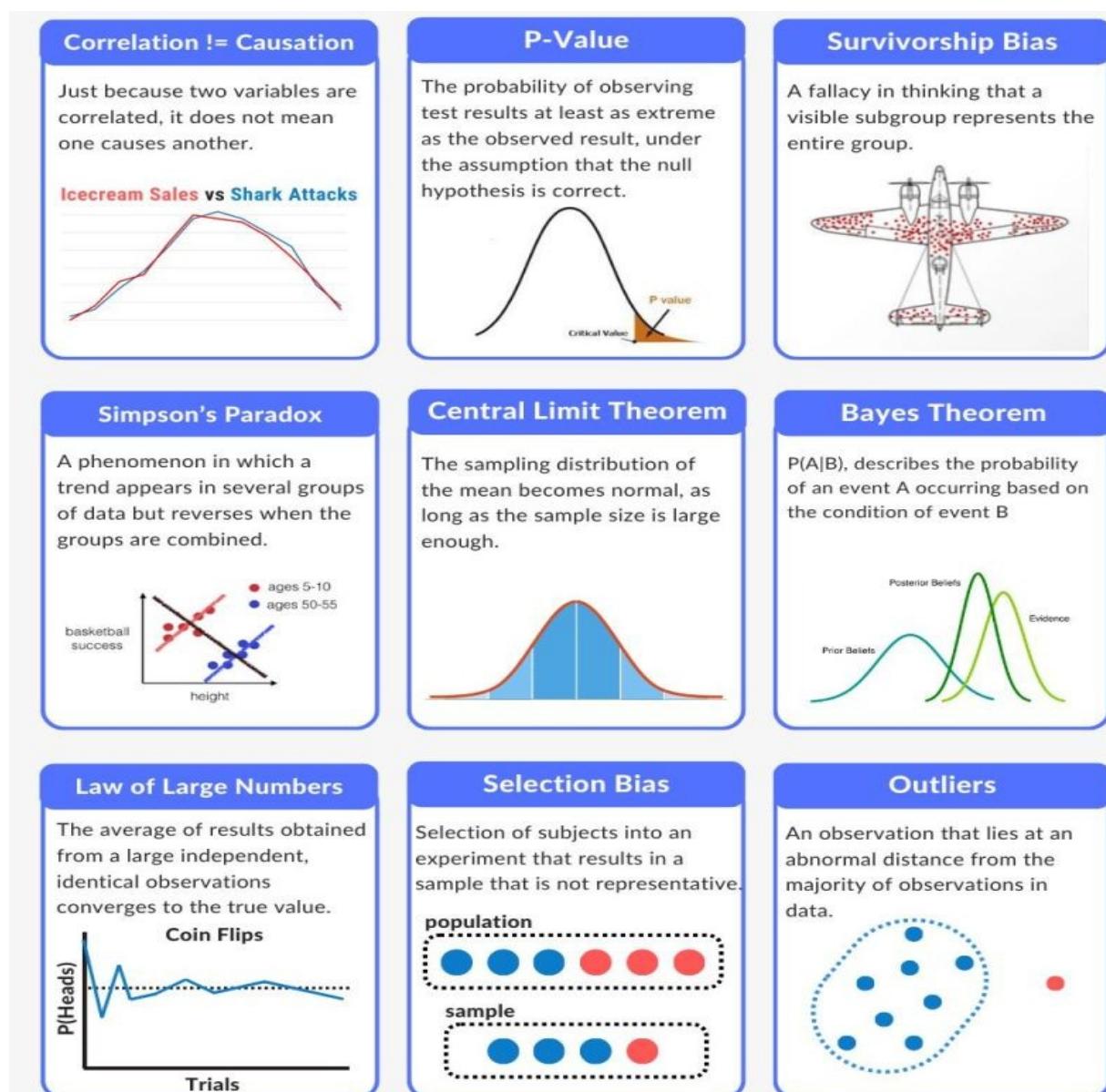


Figura 1 – Conceitos essenciais no pensamento estatístico.

Fonte: Adaptado de [1].

Os conceitos essenciais no pensamento estatístico são:

1. Correlação ≠ Causa

Só porque duas variáveis mostram uma correlação, isso não significa que um causa o outro. Por exemplo, um aumento nas vendas de sorvetes pode se correlacionar com um aumento de ataques de tubarão, mas comer sorvete não causa ataques de tubarão.

2. Valor p

O valor p mede a probabilidade de obter resultados dos testes pelo menos tão extremos quanto os resultados observados, assumindo que a hipótese nula seja verdadeira. Ajuda a determinar a significância estatística do efeito observado.

3. Bias de Sobrevivência

Esse viés (*bias*) ocorre quando um subgrupo visível é confundido com todo o grupo. Por exemplo, examinar apenas aeronaves sobreviventes da Segunda Guerra Mundial sem considerar aqueles que foram abatidos pode levar a conclusões incorretas sobre quais partes do avião precisava de reforço.

4. Paradoxo de Simpson

Uma tendência que aparece em vários grupos de dados pode reverter quando esses grupos são combinados. Esse paradoxo ilustra a importância de considerar o contexto e o agrupamento de dados em análise.

5. Teorema do Limite Central

Este teorema afirma que a distribuição de amostragem da média se aproxima de uma distribuição normal à medida que o tamanho da amostra se torna grande o suficiente, independentemente da distribuição da população. Este princípio sustenta muitos procedimentos estatísticos.

6. Teorema de Bayes

O teorema de Bayes calcula a probabilidade de um evento com base no conhecimento prévio das condições relacionadas ao evento. Ele atualiza a probabilidade à medida que

mais evidências se tornam disponíveis, fundamental em áreas como aprendizado de máquina e modelagem preditiva.

7. Lei de Grandes Números

Esta lei afirma que, à medida que o número de testes aumenta, a média dos resultados convergirá para o valor esperado. Por exemplo, virar uma moeda muitas vezes resultará na probabilidade de cabeças se aproximando de 0,5.

8. Bias de Seleção

Ocorre quando a amostra não é representativa da população, levando a resultados distorcidos. Técnicas adequadas de amostragem aleatória são essenciais para evitar esse *bias*.

9. Outliers

Outliers são pontos de dados que diferem significativamente de outras observações. Identificar e analisar outliers é crucial, pois eles podem influenciar os resultados e *insights* extraídos dos dados.

Compreender e aplicar esses conceitos pode aprimorar as habilidades de análise de dados do pesquisador, levando a conclusões mais precisas e confiáveis.

2. AS DISTRIBUIÇÕES MAIS IMPORTANTES EM CIÊNCIA DE DADOS E IA

A distribuição de dados descreve como os pontos de dados são espalhados por um conjunto de dados. A Figura 1 apresenta as distribuições mais importantes para se conhecer. As distribuições essenciais na ciência de dados e seu significado são apresentados a seguir:

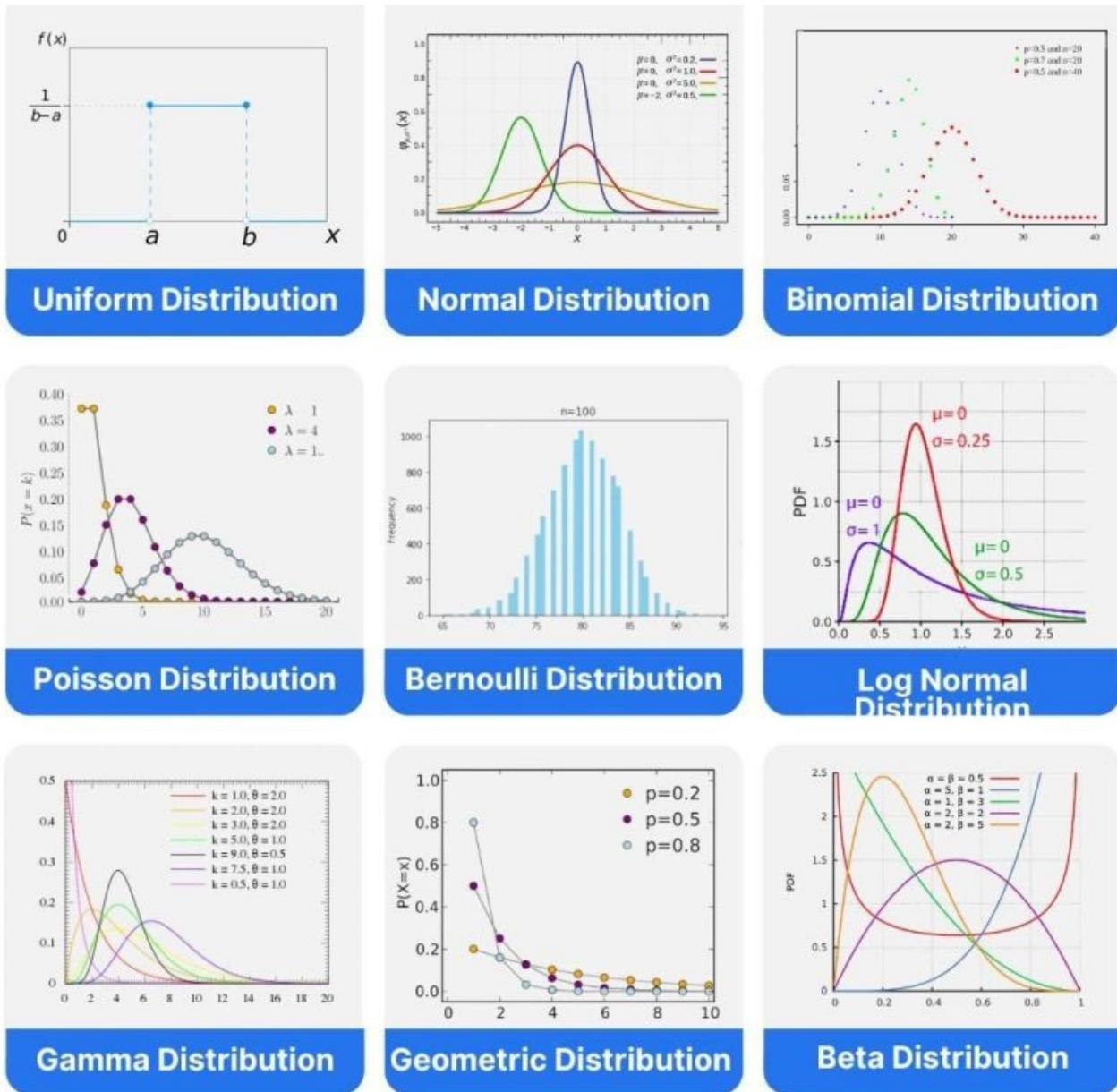


Figura 2 – As Distribuições Mais Importantes em Ciência de Dados e IA
Adaptado de [2].

1. Distribuição Uniforme:

- O que: todos os resultados são igualmente prováveis (como rolar um dado justo).
- Função: Útil quando todos os valores são igualmente prováveis, por exemplo, amostragem aleatória ou selecionando um número aleatório.
- Quando usar: quando você não tem conhecimento prévio sobre os dados.

2. Distribuição Normal (Gaussiana):

- O quê: a famosa curva de sino! Simétrico em torno da média.

- Papel: comum em ciências naturais e sociais. Muitos fenômenos do mundo real seguem esta distribuição.
- Quando usar: modelagem alturas, pontuações de teste, erros e muito mais.

3. Distribuição Binomial:

- O que: descreve o número de sucessos em um número fixo de experimentos independentes (por exemplo, jogadas de moedas).
- Função: Ótimo para perguntas sim/não, como respostas da pesquisa ou controle de qualidade.
- Quando usar: resultados binários, cenários de aprovação/falha.

4. Distribuição de Poisson:

- O quê: modelos de eventos raros (por exemplo, acidentes, telefonemas, defeitos) que ocorrem ao longo do tempo ou espaço.
- Função: Útil para contagem de dados com taxa de eventos constantes.
- Quando usar: contagem de ocorrências, como hits ou defeitos no site.

5. Distribuição de Bernoulli:

- O que: mais simples! Apenas dois resultados (sucesso ou falha).
- Função: frequentemente usada nas probabilidades de classificação binária ou modelagem.
- Quando usar: cenários sim/não, como testes A/B.

6. Distribuição Log-Normal:

- O que: O logaritmo de dados segue uma distribuição normal.
- Função: se encaixa em dados com valor positivo (por exemplo, renda).
- Quando usar: quando os dados abrangem diferentes ordens de magnitude.

7. Distribuição Gama (Gamma Distribution):

- O que: modelos de tempo de espera (por exemplo, tempo até o próximo cliente chegar).
- Função: Usado na análise de confiabilidade, reivindicações de seguro e muito mais.
- Quando usar: cenários de tempo até o evento.

8. Distribuição Geométrica:

- O que: representa o número de ensaios necessários para o primeiro sucesso (por exemplo, rolando um dado até obter um 6).
- Função: Útil para modelar eventos raros.
- Quando usar: probabilidades raras de eventos.

9. Distribuição Beta:

- O que: descreve a incerteza sobre uma probabilidade (por exemplo, taxas de conversão no teste A/B).
- Função: Comum em estatísticas bayesianas e proporções de modelagem.
- Quando usar: quando você precisar modelar proporções.

Ao entender e aplicar essas distribuições, o cientista de dados pode criar modelos robustos, tomar melhores decisões e extrair informações valiosas dos dados.

3. VALIDAÇÃO CRUZADA: A ABORDAGEM DO CONJUNTO DE VALIDAÇÃO

É o tipo de validação cruzada mais simples. A Figura 3 resume o método.

O conjunto de dados (*dataset*) é dividido em 2 grupos: treinamento e validação.

O modelo é ajustado utilizando-se o *dataset* de treinamento e avaliada a predição do modelo com o conjunto de validação.

A Figura 4 apresenta uma explanação mais completa da validação cruzada.

3.1. Vantagens e Desvantagens

- Vantagens
 - Estratégia simples
 - Fácil de Executar
- Desvantagens
 - Forte dependência da distribuição: os *datasets* de treinamento e validação podem ter propriedades diferentes.
 - O modelo é ajustado somente no *dataset* de treinamento.

The Validation Set Approach

- Validation set approach is the simplest cross-validation type
- Divide dataset into training and validation datasets
- Fit model using training dataset and evaluate model's prediction on validation set

Validation Error Rate

$$MSE = \frac{1}{n_v} \sum_{i=1}^{n_v} (y_i - y_i^{\text{predicted}})^2$$

n_v : Number of observations (validation dataset)

y_i : class label for the i -th observation

$y_i^{\text{predicted}}$: predicted class label for the i -th observation

Visual Explanation

training
dataset

validation
dataset

n_t

$n_t > n_v$

n_v

n_t : Number of observations (training dataset)

n_v : Number of observations (validation dataset)



Advantages

- Simple strategy; Quick to execute



Disadvantages

- Strong dependence on distribution:
Training and validation datasets may have different properties.
- Fit the model only on the training dataset.

Figura 3 – A Abordagem do Conjunto de Validação.

Fonte: Adaptado de [3].

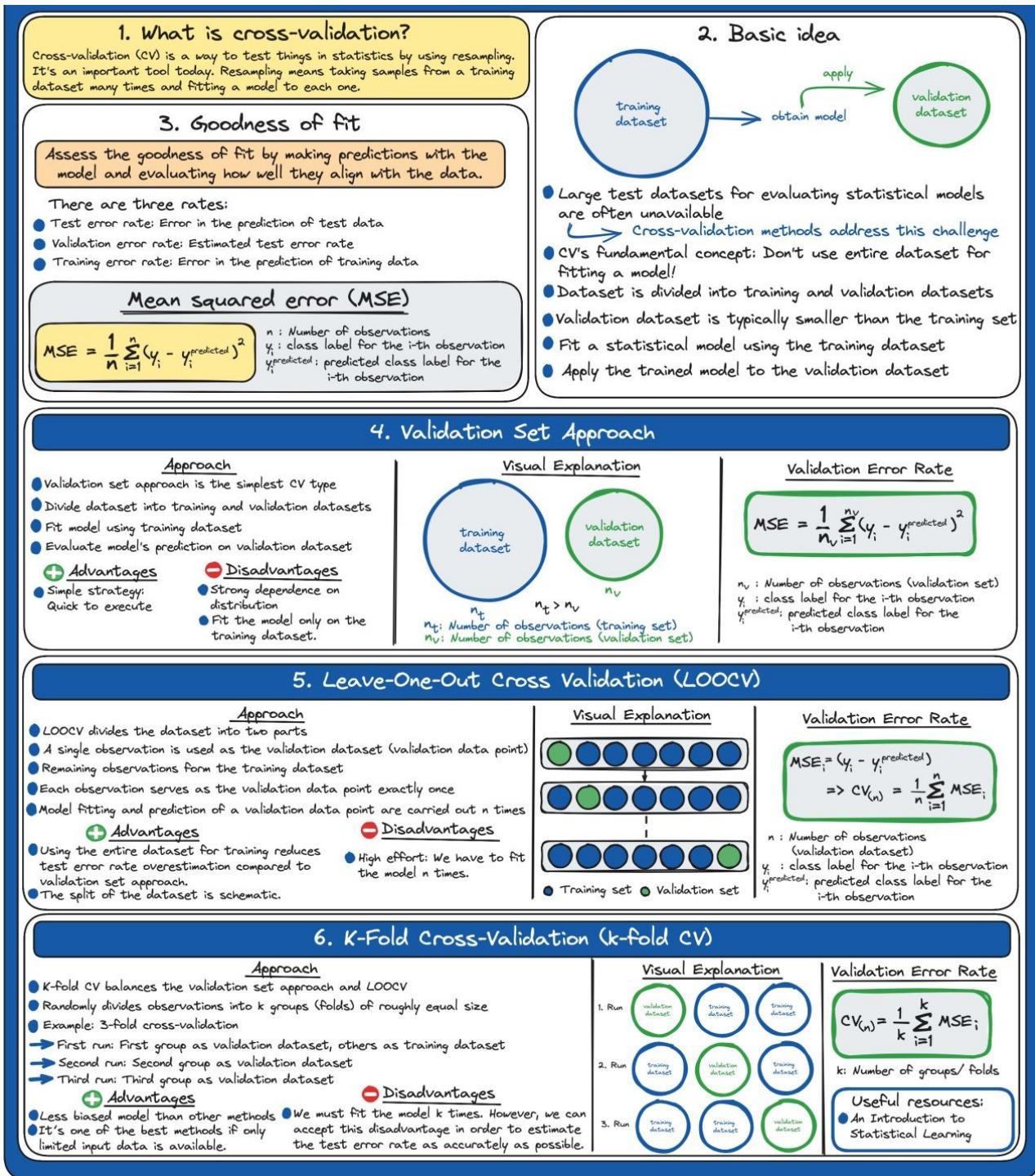


Figura 4 – Explanação de validação cruzada.

Fonte: Adaptado de [4].

REFERÊNCIA

- [1] CHANDRAGIRI, A. Statistical Thinking: Must-Know Concepts, Machine Learning Community (Linkedin Group), 2024. Disponível em:

https://media.licdn.com/dms/image/D5622AQEwppxe7IJYgg/feedshare-shrink_800/0/1722822204706?e=1725494400&v=beta&t=ksR-R5agc_S_EWP41qFDdTkt11yrQdNCjNloKkJPzbA. Acessado em Ago 05, 2024.

[2] MAITY, P. The Most Important Distributions to Know in Data Science, Artificial Intelligence, Machine Learning, Data Science & Robotics (Linkedin Group), Jul 2024. Disponível em: https://media.licdn.com/dms/image/D5622AQH4P9wL1ph8Jg/feedshare-shrink_800/0/1722138744716?e=1725494400&v=beta&t=cdfhPCJKkJ9WwBuUGmEw0upCjJchg3OH3mhXs2DAj0I. Acessado em Jul 30, 2024.

[3] TINZ, J. The Validation Set Approach. Python Developers Community (Linkedin Group), 2024. Disponível em: https://media.licdn.com/dms/image/D5622AQGgNwDMhYfmZQ/feedshare-shrink_800/0/1722442467519?e=1725494400&v=beta&t=mlsxWRpUQ6_B9Psax8uKZ2tedVqlBUdBxwqzgG7dsVw. Acessado em Ago 01, 2024.

[4] TINZ, J. Cross Validation Cheat Sheet . Machine Learning Community (Linkedin Group), 2024. Disponível em: https://media.licdn.com/dms/image/v2/D4E22AQGZ8N96fed5fA/feedshare-shrink_800/feedshare-shrink_800/0/1722704464286?e=1725494400&v=beta&t=4ZcHvkUvYgeb_IT5TORH35xa-MTv-xVVxObnh4TWjps. Acessado em Ago 05, 2024.

26. ANÁLISE AVANÇADA DE SÉRIES DE DADOS

Adaptado por: Dr. Arnaldo de Carvalho Junior

Data: Julho 08, 2024.

1. SÉRIES TEMPORAIS AVANÇADAS: TIPOS, MÉTODOS, APLICATIVOS E PRINCIPAIS BIBLIOTECAS PYTHON

A previsão avançada de séries temporais (*time series prediction*) envolve o uso de aprendizado de máquina e técnicas de aprendizado profundo para prever valores futuros de dados dependentes do tempo, levando em consideração padrões complexos e tendências sazonais.

1.1. Tipos de série temporal:

- Univariada
- Multivariada
- Estacionário
- Não Estacionário
- Sazonal
- Não sazonal
- Irregular
- Normal
- Aditivo
- Multiplicativo
- Periódico
- Não periódico

1.2. Métodos Avançados de Previsão de Série Temporal:

- a) **Redes LSTM (Long Short-Term Memory)**: Um tipo de rede neural recorrente (RNN) capaz de aprender dependências de longo prazo.
- b) **Redes GRU (Gated Recurrent Unit)**: Semelhantes às LSTM, mas com arquitetura mais simples.
- c) Modelos de transformadores (*transformer models*): usam mecanismos de atenção para capturar dependências sem depender de dados sequenciais.

- d) **TBATS (Trigonométrico, Box-Cox, ARMA, Tendência, Sazonal)**: Lida com múltiplas sazonalidades e padrões sazonais complexos.
- e) **XGBoost (Extreme Gradient Boosting)**: Uma implementação de árvores de decisão com gradiente aprimorado projetadas para velocidade e desempenho.
- f) **N-BEATS (Análise de Expansão de Base Neural)**: Uma abordagem baseada em rede neural projetada especificamente para previsão de séries temporais.
- g) **TFT (Transformadores de Fusão Temporal)**: Combina a interpretabilidade de transformadores com fusão temporal para previsão de séries temporais.
- h) **Modelos de linguagem grande (LLMs)**: LLMs como o GPT-4 podem ser adaptados para previsão de séries temporais codificando séries temporais como texto, usando incorporações, ajustando modelos pré-treinados, combinando com métodos tradicionais e aproveitando a compreensão contextual do texto. dados baseados.

1.3. Aplicações:

- v) Manutenção Preditiva
- w) Monitoramento e previsão de saúde
- x) Previsão de consumo de energia
- y) Otimização da Cadeia de Suprimentos
- z) Processamento de linguagem natural para dados temporais
- aa) Análise de dados de sensores
- bb) Previsão de fluxo de tráfego
- cc) Previsão de vendas e receitas
- dd) Previsão de Indicadores Econômicos
- ee) Modelagem Climática
- ff) Previsão do preço das ações
- gg) Previsão de preço de criptomoeda
- hh) Previsão de rotatividade de clientes
- ii) Análise de tendências de mídia social
- jj) Detecção de fraude
- kk) Detecção e resposta a eventos em tempo real

1.4. Bibliotecas para Séries Temporais baseadas nas Estrelas do GitHub

- Sktime
- Darts (Dardos)
- tsfresco
- NeuralProphet (Profeta Neural)
- STUMPY
- pmdarima
- tslearn
- GlúonTS
- Pytorch-forecasting
- StatsForecast
- Streamz
- Uber/órbita
- pyts
- NeuralForecast (Previsão Neural)
- greykite
- TSFEL
- segleaprender
- tick
- Auto_TS
- DeepAR

2. ALGORITMOS DE ML PARA MODELAGEM PREDITIVA

O aprendizado de máquina é uma grande parte de como se usam dados atualmente.

Name	Type	Description	Advantages	Disadvantages
Linear Regression		Finds the straight line that best fits the data points in a dataset.	<ul style="list-style-type: none"> Easy to explain to people who aren't data experts Performs well when the data can be separated by a straight line Overfitting can be reduced by regularization 	<ul style="list-style-type: none"> Assumes linear relationship between variables Sensitive to outliers in data Limited for complex, non-linear patterns
Logistic Regression		Binary classification algorithm using probability estimation	<ul style="list-style-type: none"> Easy to understand and explain Works well with linearly separable classes Provides probability estimates for predictions 	<ul style="list-style-type: none"> Assumes features are independent of each other Not suitable for complex, non-linear problems May underperform with highly correlated features
Decision Tree		A tree-like model that makes decisions based on feature conditions. It splits data into branches, leading to final predictions at the leaves.	<ul style="list-style-type: none"> Handles both numerical and categorical data Can be visualized for clear decision-making process Sensitive to small data changes 	<ul style="list-style-type: none"> Prone to overfitting with complex trees Can become complex with large datasets Sensitive to small data changes
Random Forest		Group of decision trees that work together to make predictions. It creates multiple trees using random data samples, then combines their results for better accuracy.	<ul style="list-style-type: none"> Handles large datasets with many features well Reduces overfitting by averaging multiple trees Resistant to outliers and noise in the data 	<ul style="list-style-type: none"> Biased towards features with more levels in categorical variables Requires more memory and resources than simpler models Less interpretable than single decision trees
Gradient Boosting		An ensemble method that builds trees sequentially, each correcting errors of previous ones. It combines weak learners to create a strong predictive model.	<ul style="list-style-type: none"> Works well with both numerical and categorical features High accuracy, often outperforming simpler, traditional algorithms Robust to outliers, enhancing prediction reliability 	<ul style="list-style-type: none"> Can be prone to overfitting if not tuned properly Computationally intensive and may be slow to train Requires careful tuning of hyperparameters for best results
Neural Network		A model inspired by the human brain, with interconnected nodes in layers. It learns patterns from data to make predictions or classifications.	<ul style="list-style-type: none"> Suitable for large datasets, improving performance with more data Effective for image, text, and speech data analysis, offering broad applications Can handle both structured and unstructured data types efficiently 	<ul style="list-style-type: none"> Requires large datasets for effective training and generalization Computationally expensive, needing significant processing power and resources Can be prone to overfitting if not properly managed

Figura 1 – Principais Algoritmos de ML para Modelagem Preditiva.

Fonte: Adaptado de [1].

A Figura 1 apresenta um infográfico dos principais algoritmos usados na modelagem preditiva. São eles:

a) Regressão Linear

- Encontra uma linha reta que melhor se encaixa nos pontos de dados.
- Fácil de explicar e funciona bem quando os dados podem ser separados por uma linha.
- Assume uma relação direta entre variáveis, o que nem sempre é o caso.

b) Regressão Logística

- Usado para classificação binária, ajudando a decidir entre duas opções.
- Simples de entender e funciona bem com dados que podem ser divididos em dois grupos claros.
- Lida com problemas complexos e não lineares.

c) Árvores de Decisão

- Como os fluxogramas, eles dividiram dados em filiais para tomar decisões.
- Lidar bem com números e categorias e é fácil de visualizar.
- Pode sobre ajuste (*overfit*), o que significa que eles podem funcionar muito bem nos dados de treinamento, mas não em novos dados.

d) Floresta Aleatória (*Random Forest*)

- Um grupo de árvores de decisão trabalhando juntas.
- Lida com grandes conjuntos de dados e reduz o excesso de ajuste, com a média dos resultados.
- Pode ser tendencioso em direção a certos recursos e requer mais recursos do que modelos mais simples.

e) Gradiente Aumentado (*Gradient Boosting*)

- Construa árvores uma após a outra, corrigindo erros ao longo do caminho.
- Preciso e funciona bem com números e categorias.

- Pode apresentar sobre ajuste (*overfit*), se não estiver sintonizado corretamente e demorar para treinar.

f) Redes Neurais Artificiais

- Inspirado no cérebro humano, usando camadas de nós para aprender com os dados.
- Ótimo para grandes conjuntos de dados e tipos de dados complexos, como imagens e texto.
- Precisa de muitos dados e potência de computação e pode demais, se não for bem gerenciado.

Cada algoritmo tem seus pontos fortes e fracos. Escolher o certo depende do problema que o pesquisador está tentando resolver e dos dados que ele possui.

REFERÊNCIAS

- [1] MIRADI, M. Advanced Time-Series: Types, Methods, Applications and Top 20 Python Libraries, 2024. Disponível em: https://www.linkedin.com/posts/maryammiradi_advanced-time-series-types-methods-applications-activity-7214536530101497857-HJM-/?utm_source=share&utm_medium=member_android. Acessado em Julho 08, 2024.
- [2] GUTIERREZ, G. A. Essential Machine Learning Algorithms for Predictive Modeling, Linkedin, 2024. Disponível em: https://media.licdn.com/dms/image/v2/D4E22AQG675mYuPkuRw/feedshare-shrink_800/feedshare-shrink_800/0/1726484161326?e=1729123200&v=beta&t=Ji3SzCp8lrylu5cVr4ylft8r_r8bT5ubzq6PP_h6LJg. Acessado em Set 16, 2024.

27. SINTONIA DE HIPERPARÂMETROS EM APRENDIZADO DE MÁQUINA

Adaptado por: Dr. Arnaldo de Carvalho Junior

Data: Julho 15, 2024.

1. IMPACTO DOS HIPERPARÂMETROS SINTONIZADOS EM ALGORITMOS ESPECÍFICOS DE ML

No mundo do aprendizado de máquina, os hiperparâmetros de ajuste fino são essenciais para a construção de modelos eficazes.

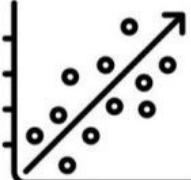
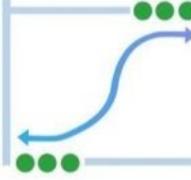
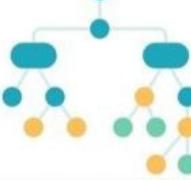
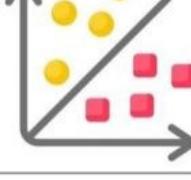
Representation	Algorithm Name	Hyperparameter
	Linear Regression	Regularization parameter (alpha for Ridge/ Lasso Regression)
	Logistic Regression	C (Inverse of regularization strength), penalty (L1, L2)
	Decision Tree	Max_depth, min_samples_splits, min_samples_leaf, criterion
	K- Nearest Neighbors	n_neighbors, weights, metric
	Support Vector Machines	C, Kernel, gamma, degree (for polynomial kernel)

Figura 1 - Sintonia de hiperparâmetros em aprendizado de máquina.

Fonte: Adaptado de [1].

O ajuste do hiperparâmetro é o processo de encontrar o melhor conjunto de parâmetros para um modelo de aprendizado de máquina (*machine learning* - ML).

Ao contrário dos parâmetros aprendidos durante o treinamento, os hiperparâmetros são definidos com antecedência e controlam o processo de aprendizado.

2. AJUSTE DOS HIPERPARÂMETROS EM ALGORITMOS ESPECÍFICOS DE ML

O ajuste do hiperparâmetro é um processo iterativo, e entender as compensações ajuda a criar modelos robustos de aprendizado de máquina.

1.1. Regressão Linear:

- a) Hiperparâmetro sintonizado: força de regularização (penalidade L1 ou L2) para evitar o excesso de ajuste..
- b) Solucionador: algoritmo de otimização para encontrar coeficientes de modelo.
- c) Peso da classe (logística): ajusta os pesos inversamente proporcionais às frequências de classe.
- d) Impacto: A regularização adequada impede o excesso de ajuste, levando a melhores previsões.
- e) Caso de uso do mundo real: previsão dos preços da habitação com base em recursos como área, quartos e localização.

A Figura 2 descreve o passo a passo da regressão linear.

1. Visual Explanation

- Linear Regression is a simple statistical regression method, ideal for beginners.
- You can perform Linear Regression with multiple variables or just one. In this sheet, we use a single variable known as Simple Linear Regression.

Goal: Find a line "Best fit" that represents the trend in the data.

2. Assumptions

- Linearity: Linear Relationship between independent and dependent variables
- Independence: The residuals should be independent and uncorrelated.
- Homoscedasticity: The variance of the residuals must be constant.
- Normal distribution: The errors of the prediction follow a normal distribution with a mean close to zero.

3. Mathematical Explanation

- To calculate the best-fit line for the data, we use the following formula:

$$Y_i = b_0 + b_1 X_i + \epsilon_i \quad i = 1 \dots n$$

Y_i : Dependent Variable
 b_0 : Intercept
 b_1 : Slope
 X_i : Independent Variable
 ϵ_i : Random Errors (Residuals)

4. How can we calculate the Residuals?

- The best-fit line has the least error.
- The Residuals are the difference between the observed values of the dependent variable and the predicted values.

$\epsilon_i = Y_i - \hat{Y}_i$

Goal: We have to minimize this error to get the best-fit line.

5. How can we calculate the Intercept and the Slope?

- We use the least squares method to calculate the intercept and the slope. This method is known as OLS regression (Ordinary Least Squares).
- We get the best fit line when the sum of the distance squares is minimal.

$$\sum_{i=1}^n (y_i - (b_0 + b_1 x_i))^2$$

$b_0^{\text{predicted}} = \bar{y} - b_1^{\text{predicted}} \bar{x}$

$$b_1^{\text{predicted}} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}$$

6. Evaluation

- Formula R-squared (R^2) or Coefficient of Determination:

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

Values between 0 and 1!

The coefficient of determination describes the percentage of variance of the dependent variable (y) that can be explained by the independent variable (x).

7. Use of Linear Regression:

Pros	Cons
<ul style="list-style-type: none"> Easy to interpret and understand Efficient method, relatively fast! Suitable for large datasets Example: Increase in sales as a result of marketing activities 	<ul style="list-style-type: none"> Statistical Assumptions Not suitable for complex data Sensitive to outliers

Figura 2 – Simples Regressão Linear Explicada.
Adaptado de [2].

1.2. Regressão Logística:

- a) Hiperparâmetro sintonizado: taxa de aprendizado ou força de regularização.
- b) Solucionador: algoritmo de otimização para encontrar coeficientes de modelo.
- c) Peso da classe (logística): ajusta os pesos inversamente proporcionais às frequências de classe.
- d) Impacto: os hiperparâmetros equilibrados melhoram a precisão da classificação.
- e) Caso de uso do mundo real: identificando *e-mails* de *spam* ou previsão de rotatividade de clientes.

1.3. Árvore de Decisão (*Decision Tree*) e Floresta Aleatória (*Randon Forest*):

- a) Hiperparâmetro ajustado: profundidade máxima ou amostras mínimas por folha.
- b) Critério: métrica para medir a qualidade das divisões (por exemplo, Gini, entropia).
- c) Profundidade máxima: profundidade máxima da árvore para controlar o ajuste excessivo.
- d) Amostra mínima: número mínimo de amostras necessárias para dividir um nó.
- e) Impacto: A poda controla a complexidade das árvores e evita o excesso de ajuste.
- f) Caso de uso do mundo real: detectando transações fraudulentas.

1.4. Árvores Aumentadas de Gradiente (*Gradient Boosted Trees*):

- a) Taxa de aprendizagem: diminui a contribuição de cada árvore para evitar o excesso de ajuste.
- b) N Estimadores: Número de estágios de reforço a serem executados.

1.5. Vizinhos K Próximos (*K-Nearest Neighbors – KNN*):

- a) Hiperparâmetro sintonizado: Número de vizinhos (K) para usar para os vizinhos mais antigos.
- b) Pesos: função em pontos de peso (por exemplo, uniforme, distância).
- c) Impacto: Escolhendo um *bias* e variação de equilíbrio apropriado.
- d) Caso de uso do mundo real: recomendando produtos semelhantes aos usuários.

1.6. K Médio (*K-Means*):

- a) n Clusters: Número de agrupamentos (*clusters*) para formar.
- b) Init: Método para inicialização (por exemplo, K-Means ++, aleatório).

1.7. Máquinas Vetoriais de Suporte (*Support Vector Machines – SVM*):

- a) Hiperparâmetro sintonizado: Função de base do *kernel* (função linear, polinomial ou radial).
- b) Impacto: o núcleo direito afeta a flexibilidade do limite da decisão.
- c) Caso de uso do mundo real: Classificando imagens (por exemplo, gato vs. Cão).

1.8. Naives Ingênuos (*Naive Bayes*)

- a) Alfa: suavizando o parâmetro para lidar com frequências zero.
- b) Ajuste antes: se deve aprender ou não os anteriores da aula.

1.9. Análise de Componente Principal (*Principal Component Analysis – PCA*):

- a) Componente N: Número de componentes principais a se manter.
- b) Solver SVD: Algoritmo para calcular a decomposição do valor singular.

1.10. Redes Neurais Densas (*Dense Neural Networks – DNN*):

- a) Tamanhos de camada oculta: Número de neurônios em cada camada oculta.
- b) Ativação: Função de ativação para a camada oculta.
- c) DROPOUT: Fração das unidades de entrada para cair para evitar o excesso de ajuste.

3. RESUMO DOS HIPERPARÂMETROS EM ML

O domínio dos hiperparâmetros pode melhorar significativamente o desempenho dos modelos de aprendizado de máquina. Mergulhar mais fundo em cada algoritmo, ajustando os modelos para assim obter melhor precisão e eficiência. A Figura 2 apresenta um resumo para os principais algoritmos de ML.

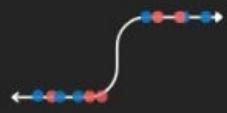
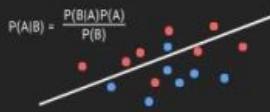
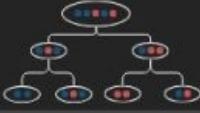
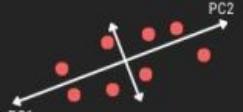
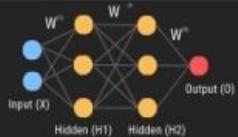
ML Algorithms	Hyperparameters
Linear Regression 	<ul style="list-style-type: none"> • L1/L2 Penalty • Solver • Fit Intercept
Logistic Regression 	<ul style="list-style-type: none"> • L1/L2 Penalty • Class Weight • Solver
Naive Bayes 	<ul style="list-style-type: none"> • Alpha • Fit Prior • Binarize
Decision Tree 	<ul style="list-style-type: none"> • Criterion • Max Depth • Min Sample Split
Random Forest 	<ul style="list-style-type: none"> • Criterion • Max Depth • N Estimators • Max Features
Gradient Boosted Trees 	<ul style="list-style-type: none"> • Criterion • Max Depth • N Estimators • Min Sample Split • Learning Rate
Principal Component 	<ul style="list-style-type: none"> • N Component • SVD Solver • Iterated Power
K-Nearest Neighbor 	<ul style="list-style-type: none"> • N Neighbors • Weights • Algorithm ('kd_tree', 'brute')
K-Means 	<ul style="list-style-type: none"> • N Clusters • Init • Max Iter
Dense Neural Networks 	<ul style="list-style-type: none"> • Hidden Layer Sizes • Activation • Dropout • Solver • Alpha • Learning rate

Figura 2 – Hiperparâmetros de Aprendizado de Máquinas.

Fonte: Adaptado de [3].

REFERÊNCIAS

- [1] MAITY, P. Hyperparameter Tuning in Machine Learning. Machine Learning Community (Moderated), 2024. Disponível em:

https://media.licdn.com/dms/image/D5622AQHTaSbAq5iN1g/feedshare-shrink_800/0/1721033750040?e=1723680000&v=beta&t=6UEJE_oXhSSP5KYgzeDVFyTefORQazepRM3hOzl4dqk.
Acessado em Julho 15, 2024.

[2] TINZ, P. Linear Regression Cheat Sheet, Python Developers Community (Linkedin Group), Jul 2024. Disponível em: https://media.licdn.com/dms/image/D4E22AQHaSx57PeMq-g/feedshare-shrink_1280/0/1722186042378?e=1724889600&v=beta&t=2Mn5N7xX-1ufR8KU0o7wGRD_V6Qei98_ofXB0xqCBMq.
Acessado em Jul 29, 2024.

[3] CHANDRAGINI, Mastering Machine Learning Hyperparameters. Machine Learning Community (Moderated), Linkedin Group, 2024. Disponível em: https://media.licdn.com/dms/image/D5622AQF52pM8S0J_cw/feedshare-shrink_800/0/1722304013740?e=1725494400&v=beta&t=wlfyC68EMRnMgfLiOsSNIK6KWt0iYxay8IpsFanEYhE.
Acessado em Jul 30, 2024.

28. ENTENDENDO A REGRESSÃO LINEAR: UM CONCEITO CENTRAL EM ESTATÍSTICA

Editado por: Dr. Arnaldo de Carvalho Junior

Data: 01 Ago 2024

A regressão linear é um método estatístico fundamental que nos ajuda a entender a relação entre variáveis e fazer previsões. A Figura 1 apresenta um sumário da regressão linear.

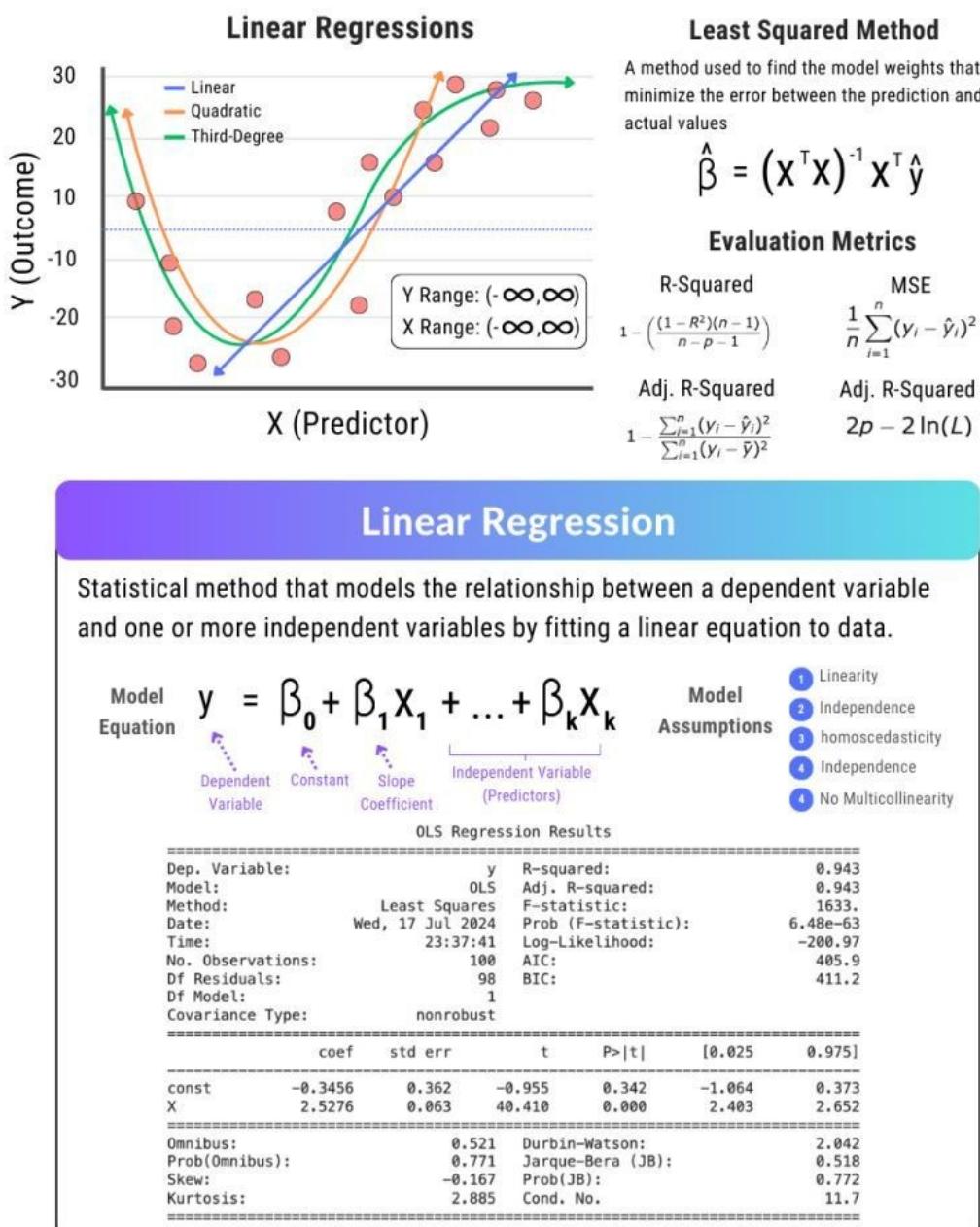


Figura 1 – A Abordagem do Conjunto de Validação.

Fonte: Adaptado de [1].

Este artigo traz uma visão geral simplificada de seus principais conceitos e usos:

1. O que é regressão linear?

Modelos de regressão linear A relação entre uma variável dependente (resultado) e uma ou mais variáveis independentes (preditores) ajustando uma linha reta nos pontos de dados.

2. Tipos de regressões lineares:

- **Régressão linear simples:** envolve um preditor e uma variável de resultado.
- **Régressão linear múltipla:** envolve vários preditores para explicar a variável de resultado.

3. Como a Regressão Linear Funciona?

A regressão linear encontra a linha mais adequada através dos pontos de dados, minimizando as diferenças entre os valores observados e os valores previstos pela linha.

4. Métricas-Chave

- **R-quadrado (R-squared - R^2):** indica quão bem os preditores explicam a variabilidade na variável de resultado. Valores mais altos significam melhor explicação.
- **Error médio ao quadrado (mean square error - MSE):** mede a média das diferenças quadradas entre os valores observados e previstos. Valores mais baixos indicam melhor precisão.

5. Suposições do Modelo

- a) Linearidade: a relação entre os preditores e o resultado é linear.
- b) Independência: as observações são independentes uma da outra.
- c) Homoscedasticidade: variação constante dos erros em todos os níveis dos preditores.
- d) Sem multicolinearidade: os preditores não estão altamente correlacionados entre si.

6. Aplicações Práticas

Aqui estão algumas aplicações práticas:

- II) Previsão de negócios: previsão de vendas, receita e tendências de mercado.
- mm) Gerenciamento de riscos: estimando riscos e retornos potenciais em finanças.
- nn) Saúde: Prevendo os resultados dos pacientes e a eficácia do tratamento.
- oo) Marketing: Analisando o impacto dos gastos com publicidade nas vendas.
- pp) Economia: Entendendo os indicadores econômicos e seu impacto nos mercados.
- qq) Ciências sociais: estudando relacionamentos entre variáveis sociais.

7. Conclusão

Entender os conceitos apresentados neste artigo é essencial para qualquer pessoa envolvida na análise de dados.

A regressão linear ajuda a tomar previsões precisas e decisões informadas com base nos dados.

REFERÊNCIA

- [1] CHANDRAGIRI, A. Linear Regression: Core Concept in Statistics. DataInterview.com, Machine Learning Community (Linkedin Group), 2024. Disponível em: https://media.linkedin.com/dms/image/D5622AQHKS7TrVKnolA/feedshare-shrink_2048_1536/0/1722482328289?e=1725494400&v=beta&t=AKst4IxWRe6E86vXvqgXV4gfORg-rLCI53aL-5h_Dr0. Acessado em Ago 01, 2024.

29. MODELOS DE CLASSIFICAÇÃO POR LIMITES DE DECISÃO

Editado por: Dr. Arnaldo de Carvalho Junior, Julho 19, 2024.

1. Introdução

Há muitas informações úteis a serem obtidas na visualização de um limite de decisão, informações que nos darão uma compreensão intuitiva dos modelos de aprendizagem [1]. Ao treinar um classificador em um conjunto de dados, usando um algoritmo de classificação específico, é necessário definir um conjunto de hiperplanos, denominado Limite de Decisão, que separa os pontos de dados em classes específicas, onde o algoritmo muda de uma classe para outra. De um lado, um limite de decisão, é mais provável que um ponto de dados seja chamado de classe A — do outro lado do limite, é mais provável que seja chamado de classe B [1].

2. Importância do Limite de Decisão

Um limite de decisão é uma superfície que separa pontos de dados pertencentes a diferentes classes (*label class*). Os limites de decisão não se limitam apenas aos pontos de dados que fornecemos, mas também abrangem todo o espaço de recursos em que são treinados. O modelo pode prever um valor para qualquer possível combinação de entradas em nosso espaço de recursos. Se os dados nos quais são treinados não forem ‘diverso’, a topologia geral do modelo será generalize mal para novas instâncias. Portanto, é importante analisar todos os modelos que podem ser mais adequados para o conjunto de dados ‘diverso’, antes de usar o modelo em produção [1].

Examinar os limites da decisão é uma ótima maneira de aprender como os dados de treinamento que forem selecionados afetam o desempenho e a capacidade de generalização do modelo. A visualização dos limites de decisão pode ilustrar o quanto sensíveis os modelos são a cada conjunto de dados, o que é uma ótima maneira de entender como algoritmos específicos funcionam e suas limitações para conjuntos de dados específicos [1].

O objetivo é construir o limite de decisão para vários algoritmos classificadores e decidir qual é o melhor algoritmo para o conjunto de dados.

3. Tipos de Classificadores de Limites

A Figura 1 apresenta os modelos de classificação e os limites de decisão.

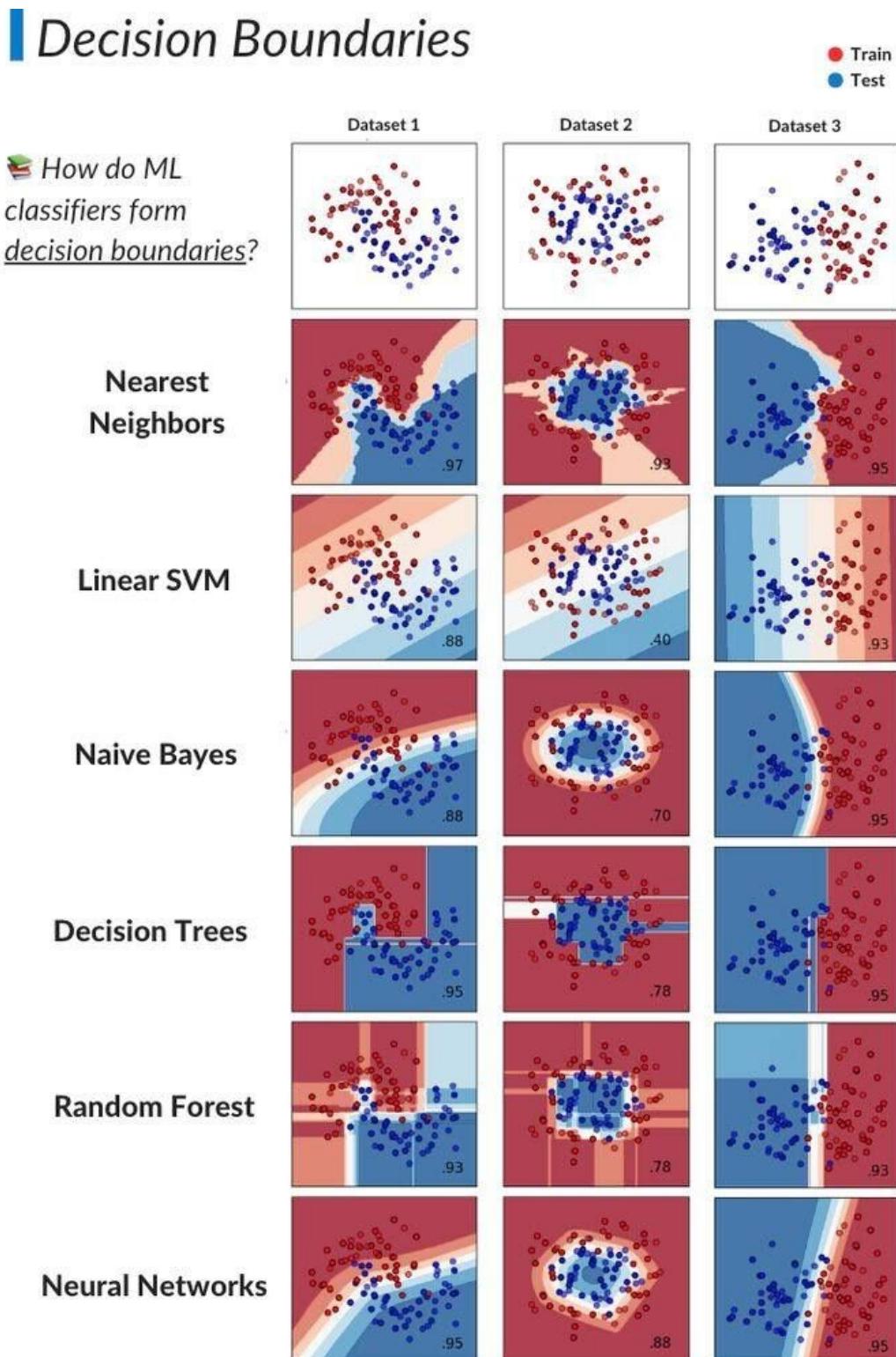


Figura 1 - Modelo de Classificação – Limites de Decisão
Fonte: Adaptado de [1].

Como os classificadores de aprendizado de máquina (*machine learning* – ML) formam limites de decisão?

Exemplos:

a) **Regressão Logística:** Limite de Decisão é uma linha linear, que separa classe A e classe B. Alguns dos pontos da classe A também chegaram à região da classe B, porque no modelo linear é difícil obter a linha limite exata que separa as duas classes.

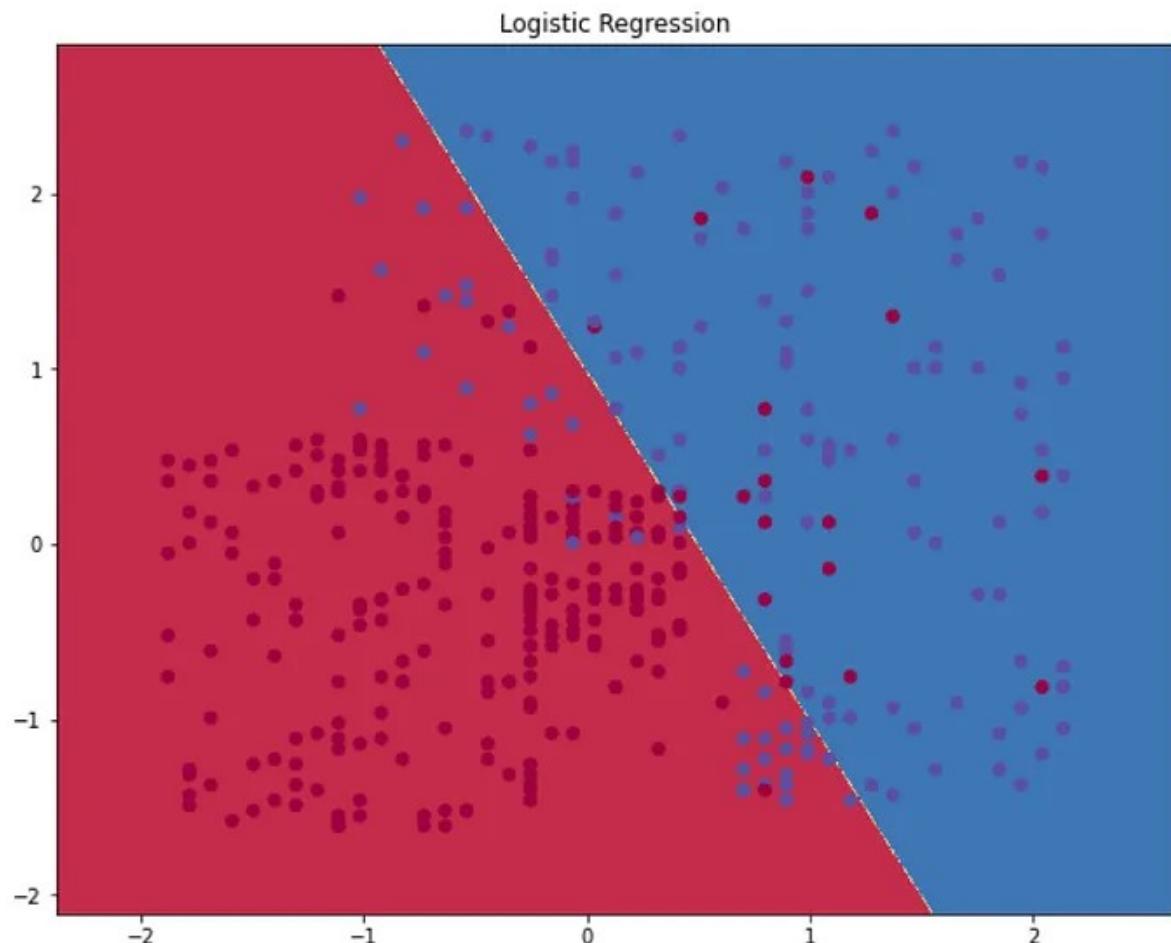


Figura 2 - Exemplo de limite de classificação por Regressão Logística [2].

b) **Árvore de Decisão e Floresta Aleatória (*Decision Tree and Random Forest*):** As superfícies de decisão da Árvore de Decisão e da Floresta Aleatória são muito complexas. A Árvore de Decisão é de longe a mais sensível, mostrando apenas probabilidades extremas de classificação que são fortemente influenciadas por pontos únicos. A Floresta Aleatória apresenta menor sensibilidade, com pontos isolados apresentando probabilidades de classificação muito menos extremas.

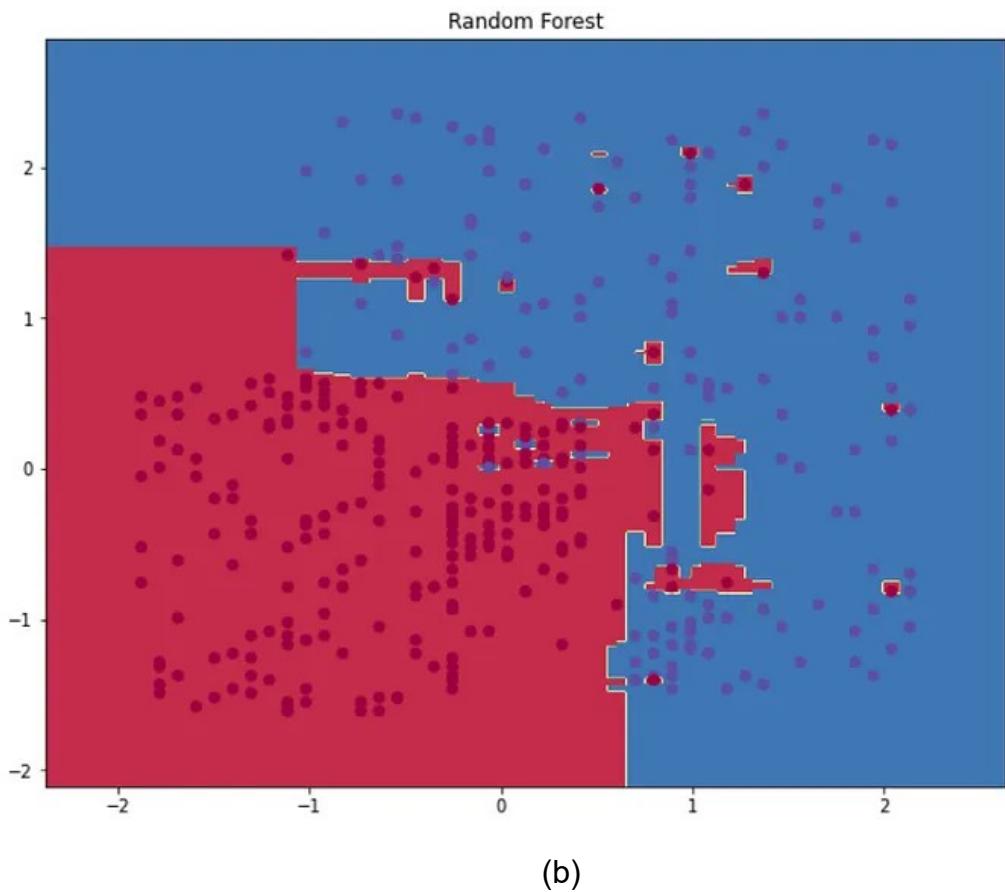
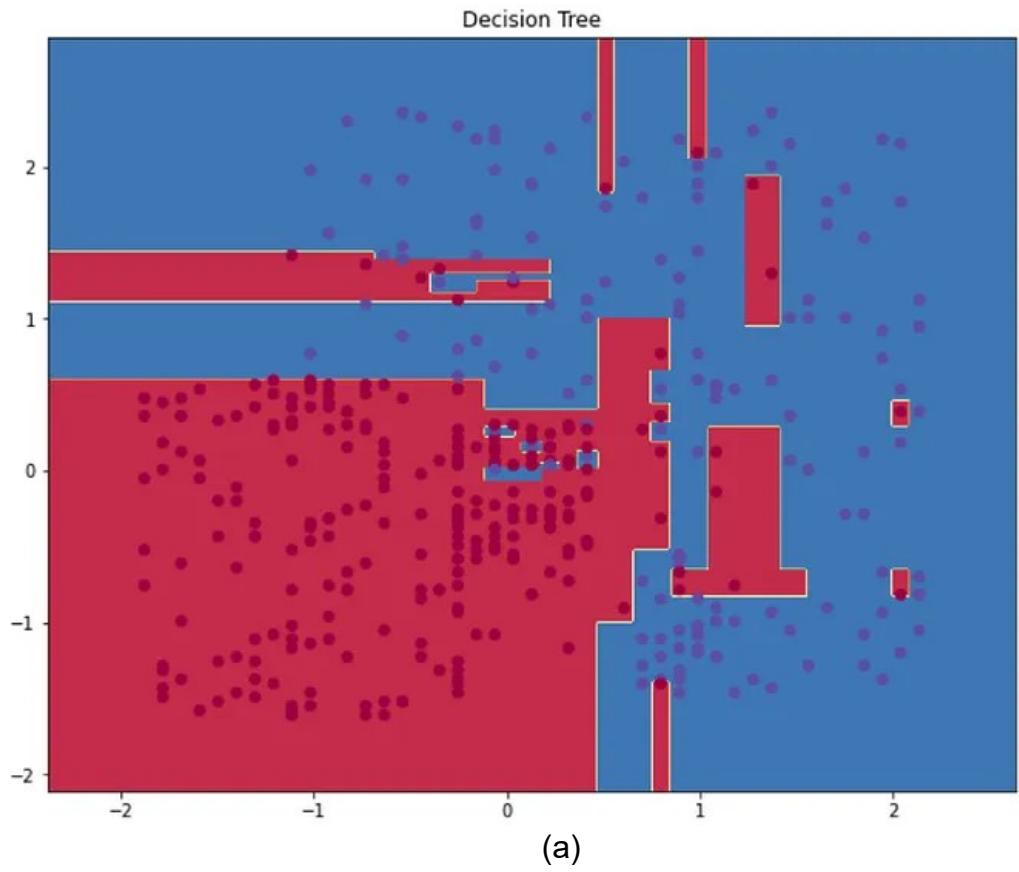


Figura 3 - limite de classificação por Árvore de Decisão e Floresta Aleatória [2].

c) **Máquina Vetorial de Suporte (Support Vector Machine – SVM):** encontra um hiperplano que separa o espaço de recursos em duas classes com a margem máxima. Se o problema não for originalmente separável linearmente, o truque do kernel é usado para transformá-lo em linearmente separável, aumentando o número de dimensões. Assim, uma hipersuperfície geral num espaço de pequena dimensão é transformada num hiperplano num espaço com dimensões muito maiores.

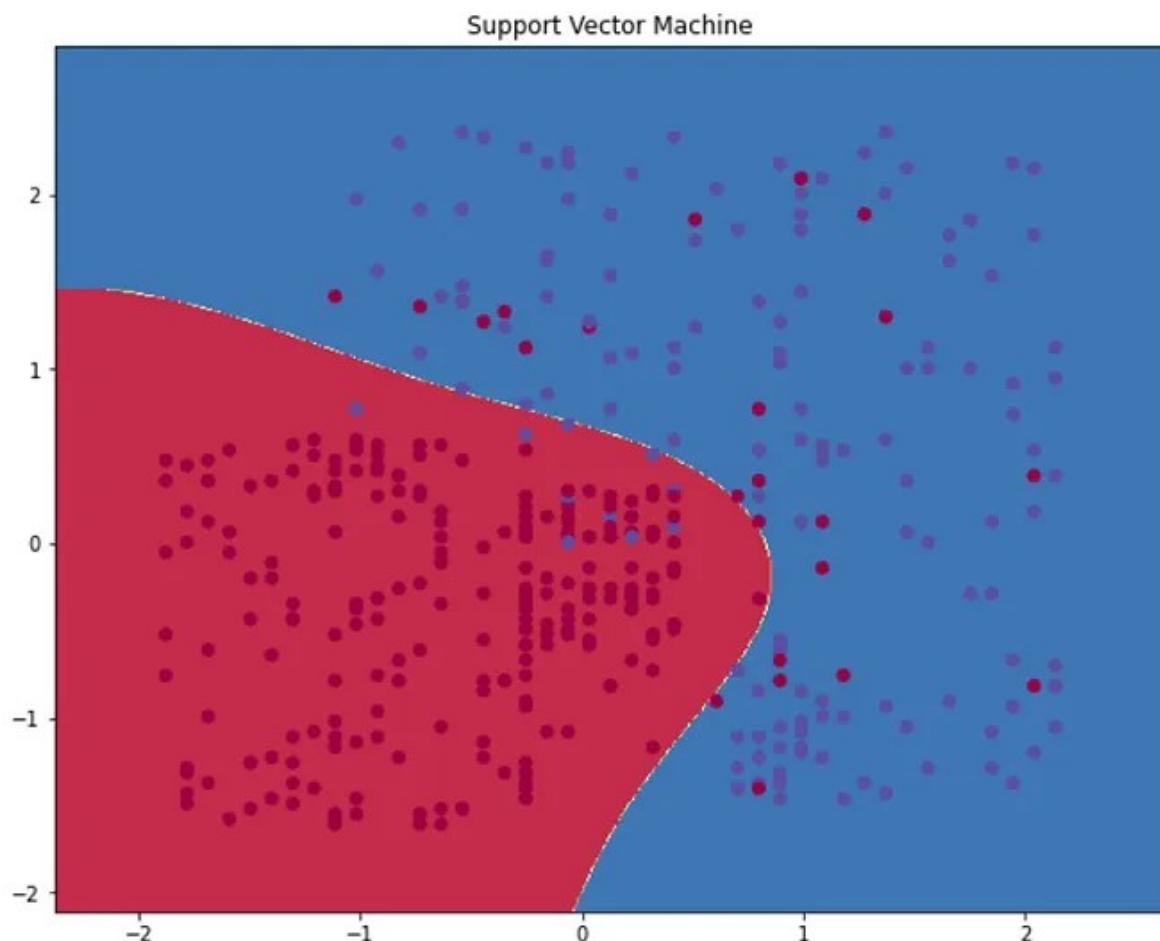


Figura 4 - Exemplo de limite de classificação por SVM [2].

d) **Modelo Gaussiano Bayesiano Ingênuo (Naive Bayes - NB):** apresenta um bom desempenho, tendo uma linha limite de curva suave.

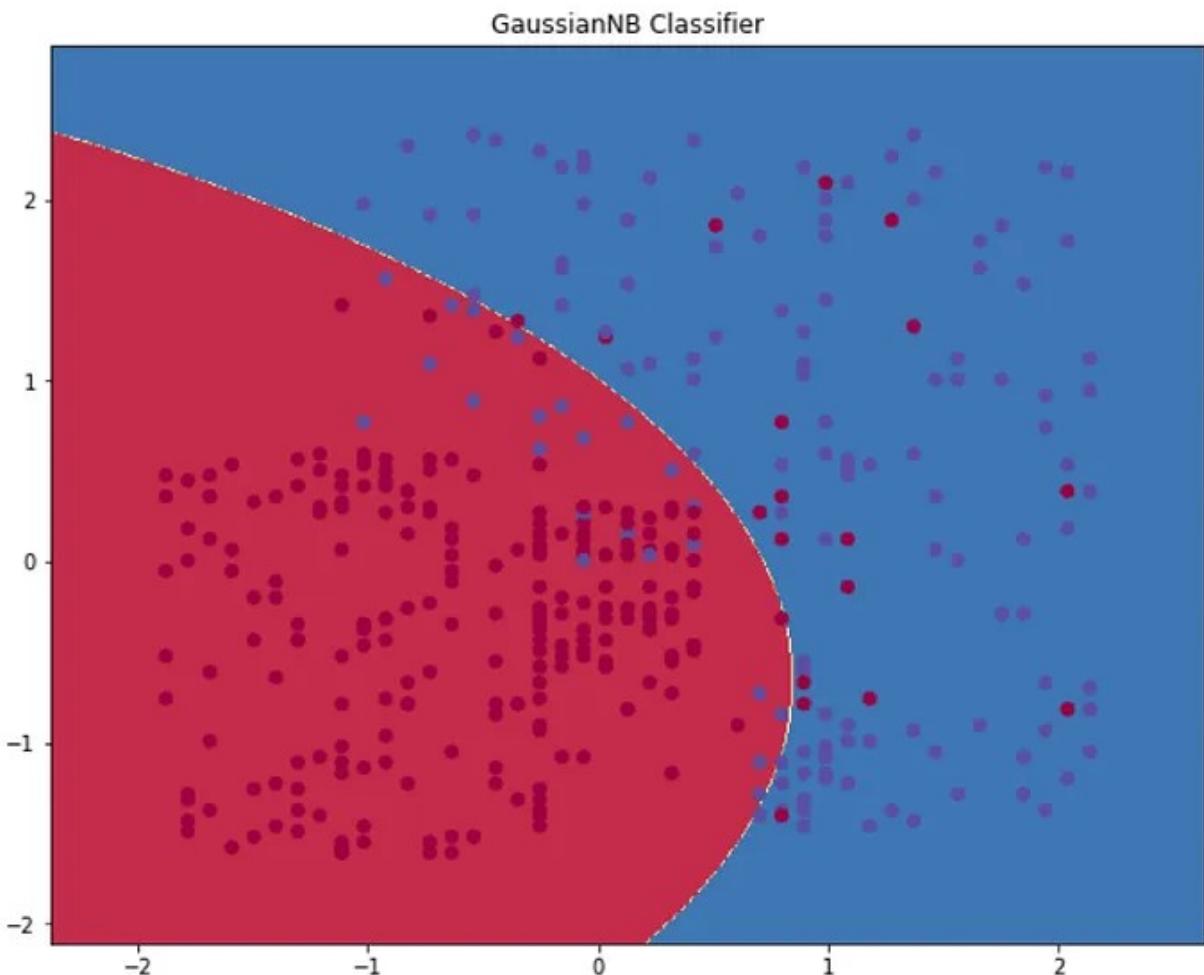


Figura 5 - Exemplo de limite de classificação por Naive Bayes [2].

4. Limites de Decisão para Dados de Dimensão Superior

Os limites de decisão podem ser facilmente visualizados para conjuntos de dados 2D e 3D. Generalizar além do 3D forma um desafio em termos da visualização onde se tem que transformar a fronteira que está presente em multidimensão para uma dimensão inferior, isso pode ser exibido e compreendido pelos especialistas é difícil [1]. No entanto, um Limite de Decisão pode ser plotado, usando tSNE, onde as dimensões dos dados podem ser reduzidas em várias etapas. Por exemplo: Se a dimensão dos dados for 150, então, inicialmente, isso será reduzido para 50 e depois para 2 dimensões.

5. Conclusão

Este artigo apresentou o papel do Limite de Decisão na determinação de um modelo classificador, apresentou modelos classificadores e traçou seus respectivos limites de decisão para selecionar o melhor modelo e também sabia que traçar um limite de decisão

para dados de dimensões superiores é uma tarefa complexa, onde as dimensões dos dados podem ser reduzidas em várias etapas.

REFERÊNCIAS

- [1] DHARMISTHA, B. Classification Model Decision Boundaries, Data Interview, 2024. Disponível em: https://media.lcdn.com/dms/image/D4D22AQGxSc1DUNqu_g/feedshare-shrink_800/0/1721309693487?e=1724284800&v=beta&t=BaseHEoVrdM_rTW7-9dfG4puQsL453j1wuTmRSCJgxM. Acessado em Jul 19, 2024.
- [2] SAHU, S. DECISION BOUNDARY FOR CLASSIFIERS: AN INTRODUCTION. ANALYTICS, Medium, 2021. Disponível em: <https://medium.com/analytics-vidhya/decision-boundary-for-classifiers-an-introduction-cc67c6d3da0e>. Acessado em Jul 19, 2024.

30. AGRUPAMENTOS DE DADOS K-MEANS, GMM E DBSCAN

Editado por: Dr. Arnaldo de Carvalho Junior, Julho 17, 2024.

1. AGRUPAMENTO (CLUSTERING)

Agrupamento é uma das técnicas de análise exploratória de dados mais comuns usadas para ter uma intuição sobre a estrutura dos dados. Pode ser definida como a tarefa de identificar subgrupos nos dados de forma que os pontos de dados no mesmo subgrupo (*cluster*) sejam muito semelhantes, enquanto os pontos de dados em diferentes *clusters* sejam muito diferentes. Ou seja, tentamos encontrar subgrupos homogêneos dentro dos dados, de modo que os pontos de dados em cada *cluster* sejam tão semelhantes quanto possível de acordo com uma medida de similaridade, como distância baseada em euclídeos ou distância baseada em correlação. A decisão de qual medida de similaridade usar é específica do aplicativo [1].

A análise de agrupamento pode ser feita com base em recursos onde tentamos encontrar subgrupos de amostras com base em recursos ou com base em amostras onde tentamos encontrar subgrupos de recursos com base em amostras. Neste artigo será abordado o agrupamento baseado em recursos. *Clustering* é usado na segmentação de mercado; onde tentamos encontrar clientes que são semelhantes entre si seja em termos de comportamentos ou atributos, segmentação/compressão de imagens; onde tentamos agrupar regiões semelhantes, agrupamento de documentos com base em tópicos, etc [2]. O *clustering* é considerado um método de aprendizagem não supervisionado, uma vez que não temos a verdade fundamental para comparar a saída do algoritmo de *clustering* com os rótulos verdadeiros para avaliar seu desempenho. Queremos apenas tentar investigar a estrutura dos dados agrupando os pontos de dados em subgrupos distintos [1].

2. ALGORITMO K-MEANS

O K-Means é um algoritmo iterativo que tenta partitionar o conjunto de dados em subgrupos K distintos, não sobrepostos e pré-definidos (*clusters*), aos quais cada ponto de dados pertence apenas um grupo. Ele tenta tornar os pontos de dados intra-agrupamento o mais semelhante possível, ao mesmo tempo que mantém os *clusters* o mais diferente ou distante (*far*) possível. Ele atribui pontos de dados a um *cluster* de modo que a soma da distância quadrada entre os pontos de dados e a média (aritmética

centróide do *cluster* de todos os pontos de dados que pertencem a esse *cluster*) está no mínimo. Quanto menor a variação que temos dentro dos *clusters*, mais homogêneos (similar) os pontos de dados estão dentro do mesmo cluster [1].

A forma como o algoritmo K-Means funciona é a seguinte:

1. Especifique o número de *clusters* K.
2. Inicialize os centróides primeiro embaralhando o conjunto de dados e depois selecionando aleatoriamente K pontos de dados para os centróides sem reposição.
3. Continue iterando até que não haja alteração nos centróides, ou seja, a atribuição de pontos de dados aos *clusters* não está mudando.
 - Calcule a soma da distância ao quadrado entre os pontos de dados e todos os centróides.
 - Atribua cada ponto de dados ao *cluster* mais próximo (centroide).
 - Calcule os centróides para os *clusters* tomando a média de todos os pontos de dados que pertencem a cada cluster.

3. A PRINCIPAL LIMITAÇÃO DO K-MEANS

O K-Means é amplamente utilizado por sua simplicidade e eficácia como algoritmo de agrupamento. Mas tem muitas limitações. Para começar:

- a) Não leva em conta a variância dos agrupamentos (*cluster*);
- b) Só pode produzir aglomerados esféricos. Conforme mostrado abaixo, mesmo que os dados tenham *clusters* não circulares, eles ainda produzem *clusters* redondos.
- c) Ele executa uma tarefa difícil. Não há estimativas probabilísticas de cada ponto de dados pertencente a cada *cluster*.

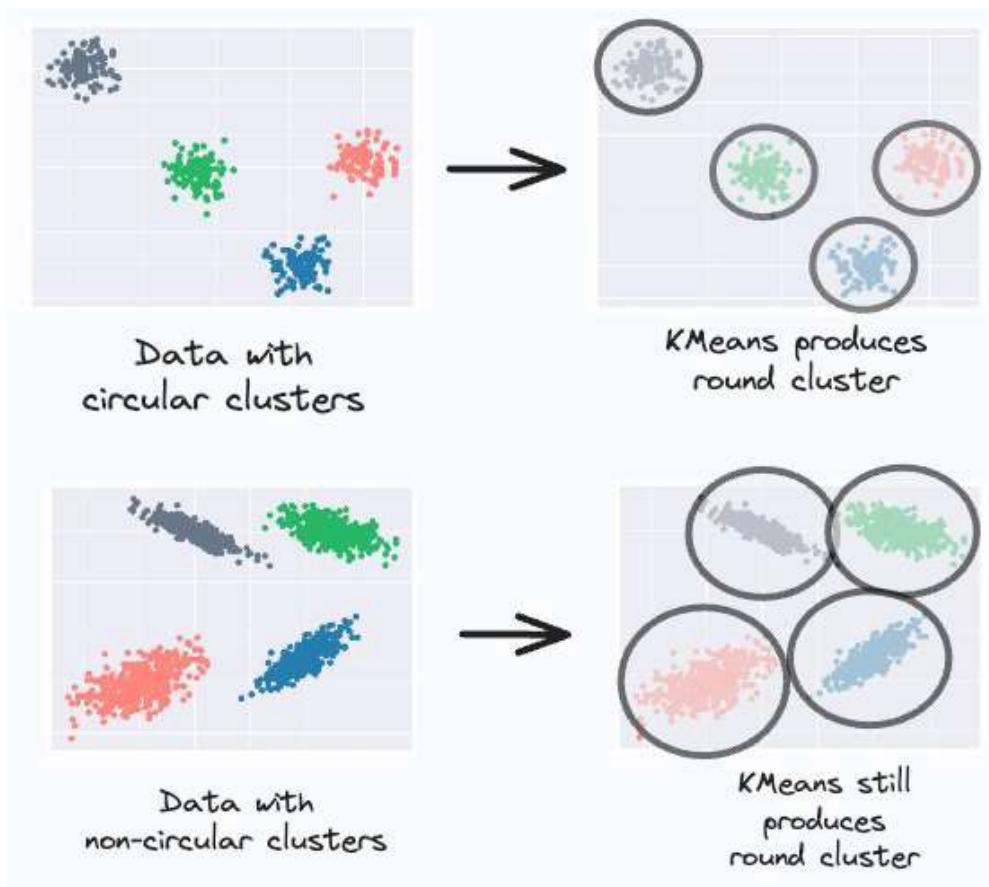


Figura 1 - Agrupamento (clustering) K-Means.

Fonte: Adaptado de [2]

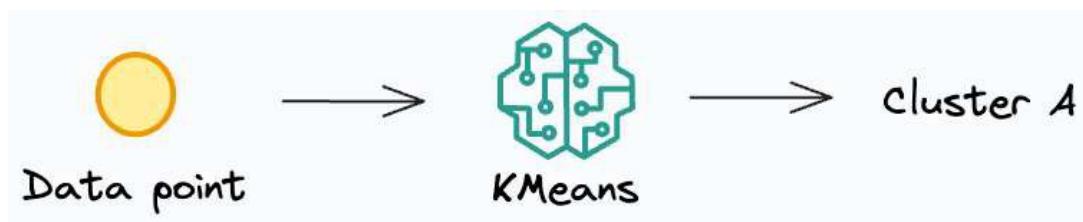


Figura 2 - K-Means executando tarefas difíceis.

Fonte: Adaptado de [2]

Essas limitações geralmente tornam o K-Means uma escolha não ideal para agrupamento.

Os modelos de mistura gaussiana (*gaussian mixture model* – GMM) são frequentemente um algoritmo superior a esse respeito. Como o nome sugere, eles podem agrupar um conjunto de dados que possui uma mistura de muitas distribuições gaussianas. Eles podem ser considerados um gêmeo mais flexível dos K-Means. A diferença primária é que:

- K-Means aprende centróides.
- Modelos de mistura gaussiana aprendem a distribuição.

Por exemplo, em 2 dimensões:

- Os K-Means só podem criar *clusters* circulares
- GMM pode criar aglomerados de formato oval.

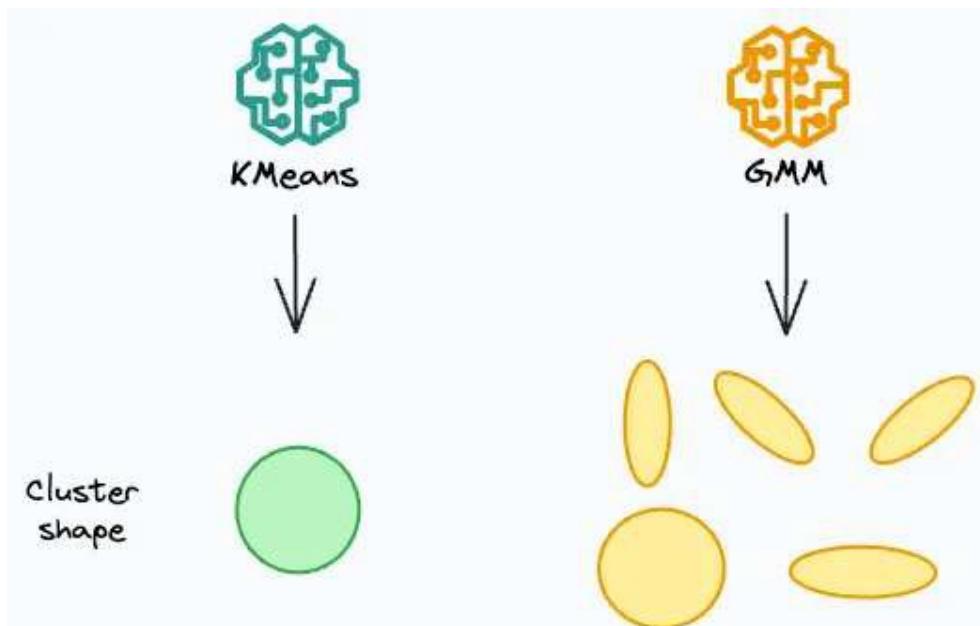


Figura 3 - Comparação K-Means e GMM

Fonte: Adaptado de [2]

4. GMM

Um modelo de mistura gaussiana (Gaussian Mixture Model -GMM) é um modelo probabilístico que assume todos os pontos de dados são gerados a partir de uma mistura de um número finito de Distribuições gaussianas com parâmetros desconhecidos. Pode-se pensar em modelos de mistura como generalização de agrupamento k-Means para incorporar informações sobre a estrutura de covariância dos dados, bem como o centros dos gaussianos latentes [3].

Os GMMs são modelos estatísticos que representam os dados como uma mistura de distribuições gaussianas (normal). Esses modelos podem ser usados para identificar grupos dentro do conjunto de dados e para capturar a estrutura complexa e multimodal das distribuições de dados [3].

Os GMMs são usados em uma variedade de aplicações de aprendizado de máquina, inclusive agrupamento, estimativa de densidade e reconhecimento de padrões [3].

Pode-se considerar o GMM como K-Means probabilísticos porque o ponto de partida e o processo de treinamento dos K-Means e GMM são os mesmos. No entanto, O K-Means usa uma abordagem baseada na distância e o GMM usa uma abordagem probabilística. Há uma suposição primária no GMM: o conjunto de dados consiste em vários gaussianos, em outras palavras, uma mistura do gaussiano [3].

Pode-se estimar os parâmetros dos modelos GMM usando uma técnica poderosa conhecida como expectativa-maximização (*Expectation-Maximization – EM*) [4].

A eficácia dos GMMs sobre os K-Means é evidente na figura a seguir [2].

- K-Means depende apenas da distância e ignora a distribuição de cada *cluster*;
- GMM aprende a distribuição e produz melhor agrupamento.

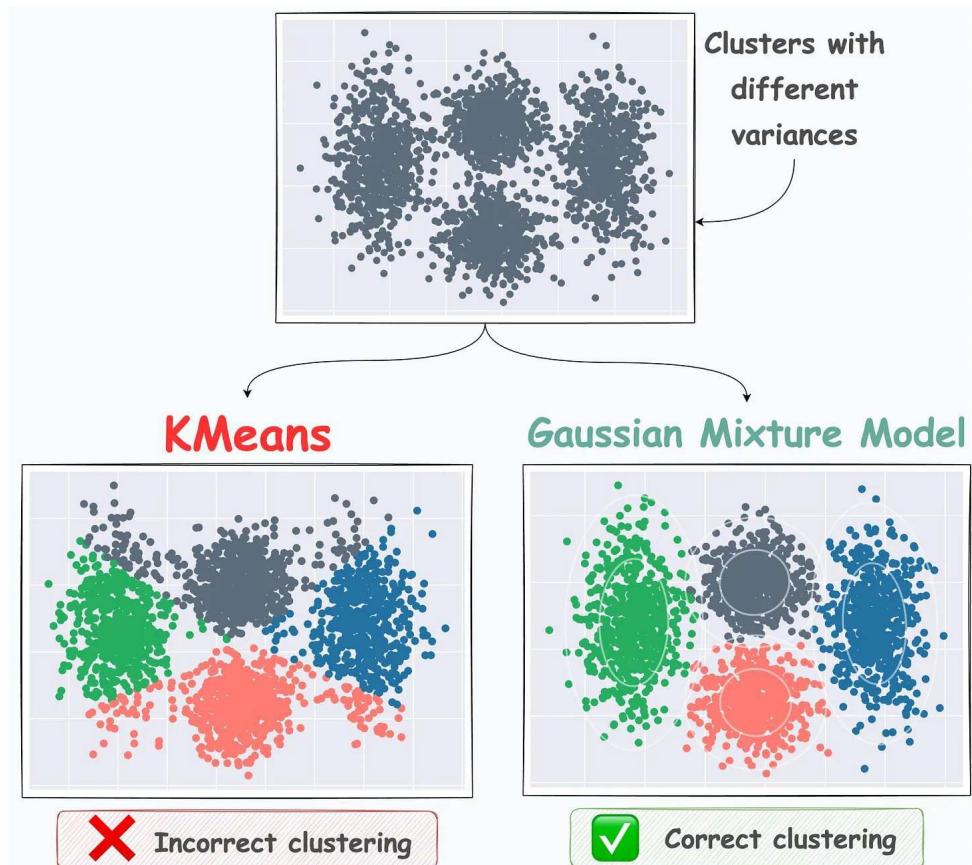


Figura 4 - Comparação K-Means e GMM para clusters com variâncias diferentes.

Fonte: Adaptado de [2]

Toda a ideia e formulação de modelos de mistura gaussiana são extremamente atraente e intrigante quando estudados primeira vez.

A noção de que um único modelo pode aprender diversas distribuições de dados é verdadeiramente cativante. Aprender sobre eles tem sido extremamente útil na construção de algoritmos de agrupamento mais flexíveis e confiáveis. Assim, entender como eles funcionam de ponta a ponta será imensamente valioso se o pesquisador estiver procurando expandir sua experiência além de algoritmos tradicionais como K-Means, agrupamento baseado em densidade espacial para aplicações com ruído (*Density-Based Spatial Clustering of Applications with Noise* – DBSCAN) [3], etc.

Se o pesquisador usar K-Means o tempo todo, leia o seguinte:

Conhecer um algoritmo é de apenas 20% de conhecimento. O restante 80% vem da compreensão das limitações práticas.

Aqui estão algumas limitações de K-Means que muitos ignoram:

- Ele não explica a variação do cluster (o *spread* em 2D, por exemplo) - só pode produzir aglomerados globulares (sem formas ovais em 2D, apenas circulares);
- Ele se baseia apenas em medidas baseadas em distância para atribuir pontos de dados a *clusters*:
 - Para entender melhor, considere dois clusters em 2D, com A e B. O *cluster* A tem uma propagação mais alta que B.
 - Agora considere uma linha que está no meio do caminho entre os centróides de A e B → A | B.
 - Embora A tenha uma propagação mais alta, mesmo que um ponto seja um pouco certo para a linha média, ele será atribuído ao *cluster* B.
 - Idealmente, no entanto, o *cluster* A deve ter uma área de influência maior.
- Ele executa uma tarefa difícil. Assim, não há estimativas probabilísticas de cada ponto de dados pertencentes a um *cluster*.

Essas limitações geralmente fazem de K-Means uma opção não ideal para o agrupamento. É possível encontrar modelos de mistura gaussiana (GMM) como um algoritmo bastante superior a esse respeito. Como o nome sugere, eles podem agrupar um conjunto de dados que tem uma mistura de muitos gaussianos. Então, pode-se pensar nos GMMs como um gêmeo generalizado de K-Means.

A principal diferença é que:

- Kmeans aprende centróides.
- GMM aprende uma distribuição.

Por exemplo, em 2 dimensões:

- Enquanto Kmeans só podem criar aglomerados circulares
 - Um GMM pode criar *clusters* de forma oval.
- K-means apenas depende da distância e ignora a distribuição de cada *cluster*.
- O GMM aprende a distribuição e produz melhor *cluster*.

Como funciona?

Em uma essência, ele usa maximização de expectativa (*expectation-maximization* – EM), que é uma técnica de otimização iterativa para estimar os parâmetros dos modelos estatísticos [4].

A ideia principal por trás de EM é a seguinte:

- Faça um palpite bastante sobre os parâmetros
- Etapa E: Calcule as probabilidades posteriores da variável não observada usando parâmetros de corrente.
- Defina a função de “probabilidade esperada” usando as probabilidades posteriores.
- M-ETP: Atualize os parâmetros atuais, maximizando a “probabilidade esperada”.
- Use os parâmetros atualizados para recomputar as probabilidades posteriores, isto é, de volta ao e-Step.
- Repita até a convergência.

5. DBSCAN

O DBSCAN é um popular algoritmo de **clusterização** que agrupa pontos de dados baseado em densidade espacial, em vez de distância. Isso significa que ele forma *clusters* identificando regiões de alta densidade de pontos, separados por regiões de baixa densidade de pontos.

- DBSCAN é um algoritmo de clusterização baseado em densidade espacial.
- Ele forma *clusters* identificando regiões de alta densidade de pontos.
- Diferentemente de outros algoritmos, ele não se baseia na distância entre os pontos.

5.1. Vantagens do DBSCAN

O DBSCAN é especialmente útil para identificar *clusters* de formatos arbitrários, ao contrário de algoritmos como K-Means que pressupõem *clusters* esféricos. Além disso, o

DBSCAN é robusto à presença de pontos fora da curva (*outliers*), pontos que estão distantes de quaisquer *clusters*.

- DBSCAN é capaz de identificar *clusters* de formatos arbitrários.
- É mais flexível do que algoritmos como K-Means, que pressupõem *clusters* esféricos.
- É robusto à presença de *outliers*, que podem distorcer os resultados de outros algoritmos de clusterização.

5.2. Algoritmo DBSCAN

O algoritmo DBSCAN é uma ferramenta poderosa para identificar *clusters* em conjuntos de dados, funcionando de maneira automatizada e eficiente. Ele opera identificando *core points* e agrupando-os com base na densidade, além de atribuir *border points* aos clusters correspondentes e marcar os *outliers*. O DBSCAN é uma ótima opção para identificar padrões em dados complexos e multidimensionais.

- O DBSCAN identifica *core points* e os agrupa com base na densidade;
- Atribui *border points* aos *clusters* correspondentes e marca *outliers*;
- É eficiente para identificar padrões em conjuntos de dados complexos e multidimensionais.

REFERÊNCIAS

[1] DABBURA, I. K-means Clustering: Algorithm, Applications, Evaluation Methods, and Drawbacks, Towards Data Science, Medium, 2018. Disponível em:
<https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluationmethods-and-drawbacks-aa03e644b48a>. Acessado em Jul 17, 2024.

[2] CHAWLA, A. Modelos de mistura gaussiana (GMMs). Daily Dose of Data Science, 2024. Disponível em:<https://www.dailydoseofds.com/gaussian-mixture-models-gmm/>. Acessado em Julho 17, 2024.

[3] YEHOSHUA, R. Modelos de Mistura Gaussiana (GMMs): da Teoria à Implementação, Towards Data Science, Medium, 2023. Disponível em:
<https://towardsdatascience.com/gaussian-mixture-models-gmms-from-theory-toimplementation-4406c7fe9847>. Acessado em Julho 17, 2024.

[4] BROWNLEE, J. A Gentle Introduction to Expectation-Maximization (EM Algorithm), Machine Learning Mastery, 2020. Disponível em:
<https://machinelearningmastery.com/expectation-maximization-em-algorithm>. Acessado em Julho 17, 2024.

31. MÉTRICAS DE AVALIAÇÃO DE MODELO PARA APRENDIZADO DE MÁQUINAS

Por: Karan Ajay Pisay

Editado por: Dr. Arnaldo de Carvalho Junior, Julho 19, 2024.

1. Introdução

Um aspecto crucial do aprendizado de máquina que garante que os modelos gerados tenham desempenho preciso e confiável.

1.1. Por que as métricas de avaliação são importantes?

Construir modelos de aprendizado de máquina – (*machine learning* - ML) não se trata apenas de fazer previsões. Trata-se de fazer previsões precisas. As métricas de avaliação nos ajudam a entender o desempenho de nossos modelos e onde precisamos melhorar.

2. Métricas-chave

a) Acurácia: a proporção de instâncias previstas corretamente para o total de instâncias. Ideal para conjuntos de dados equilibrados.

Exemplo: se o filtro de spam identificar corretamente 90 dos 100 e-mails, a precisão será de 90%.

b) Precisão: a proporção de observações positivas corretamente previstas para o total de positivos previstos. Crucial quando falsos positivos são caros.

Exemplo: em um modelo de diagnóstico médico, alta precisão significa menos pessoas são diagnosticadas incorretamente com uma doença.

c) Sensibilidade (*Sensitivity* ou *Recall*): a proporção de observações positivas previstas corretamente para os positivos reais. Importante quando a falta de positivos é cara.

Exemplo: Na detecção de fraude, o alto *recall* significa que mais transações fraudulentas são identificadas corretamente.

d) Pontuação F1 (F1 Score): a média harmônica de precisão e recall. Equilibra a precisão e o *recall*, especialmente útil para conjuntos de dados desequilibrados.

Exemplo: Útil em situações em que falsos positivos e falsos negativos são caros, como na contratação de algoritmos.

e) Pontuação do ROC-AUC: mede a capacidade de um modelo de distinguir entre classes. Quanto maior a AUC, melhor o modelo.

Exemplo: uma pontuação ROC-AUC próxima a 1 indica um ótimo modelo, enquanto 0,5 sugere adivinhação aleatória.

3. Exemplo do mundo real:

Imagine que o pesquisador está construindo um modelo para detectar transações fraudulentas para um banco. A alta recall garante que a maioria das fraudes seja capturada, reduzindo as perdas financeiras, enquanto a alta precisão garante que as transações genuínas não sejam sinalizadas, mantendo a satisfação do cliente. Equilibrar-se através da pontuação da F1 pode ajudar a otimizar o desempenho do modelo.

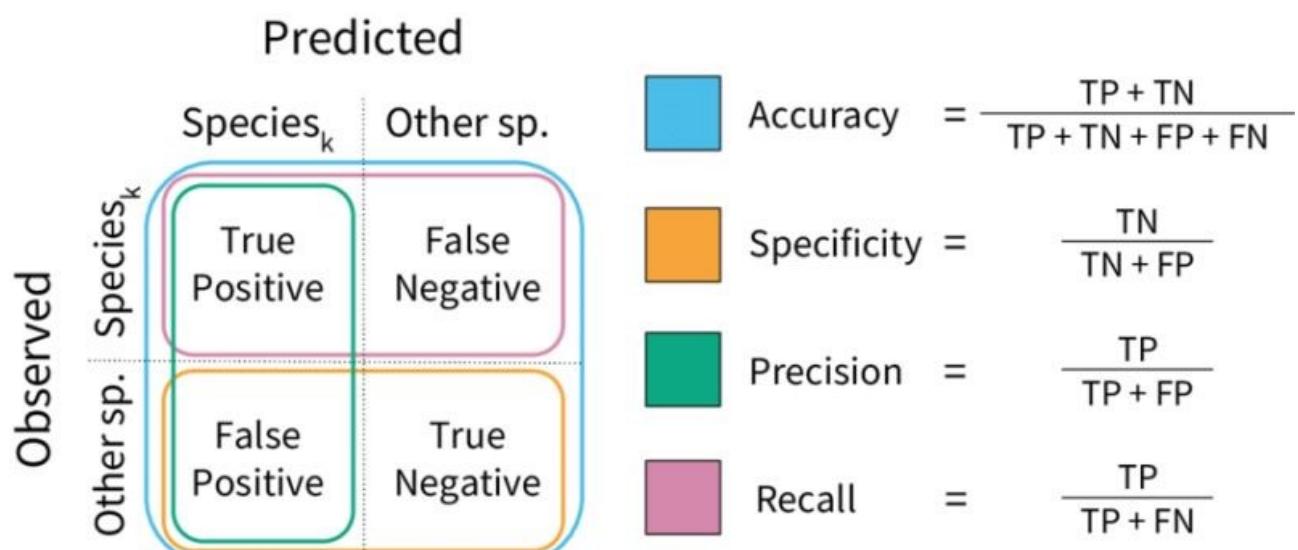


Figura 1 - Métricas de Avaliação de Modelo

Fonte: Adaptado de [1].

REFERÊNCIAS

- [1] PISAY, K. A. Understanding Model Evaluation Metrics for Machine Learning. Artificial Intelligence, Machine Learning, Data Science & Robotics (Linkedin Group), 2024. Disponível em: https://media.linkedin.com/dms/image/D5622AQGcE77r0tlxzQ/feedshare-shrink_800/0/1721310004487?e=1724284800&v=beta&t=JYAs9H8nyS6bloO-2WpVwkt-JkWDgo5ZzyQDINWg2zo. Acessado em Jul 19, 2024.

32. QUAL O MODELO DE APRENDIZADO DE MÁQUINA IDEAL?

Editado por: Dr. Arnaldo de Carvalho Junior

Data: Julho 29, 2024.

1. ESCOLHENDO O MODELO DE ML

Dúvida de todo o pesquisador é definir qual modelo de aprendizado de máquina é adequado para o projeto? Este documento lista os prós e contras de cada modelo de aprendizado de máquina [1].

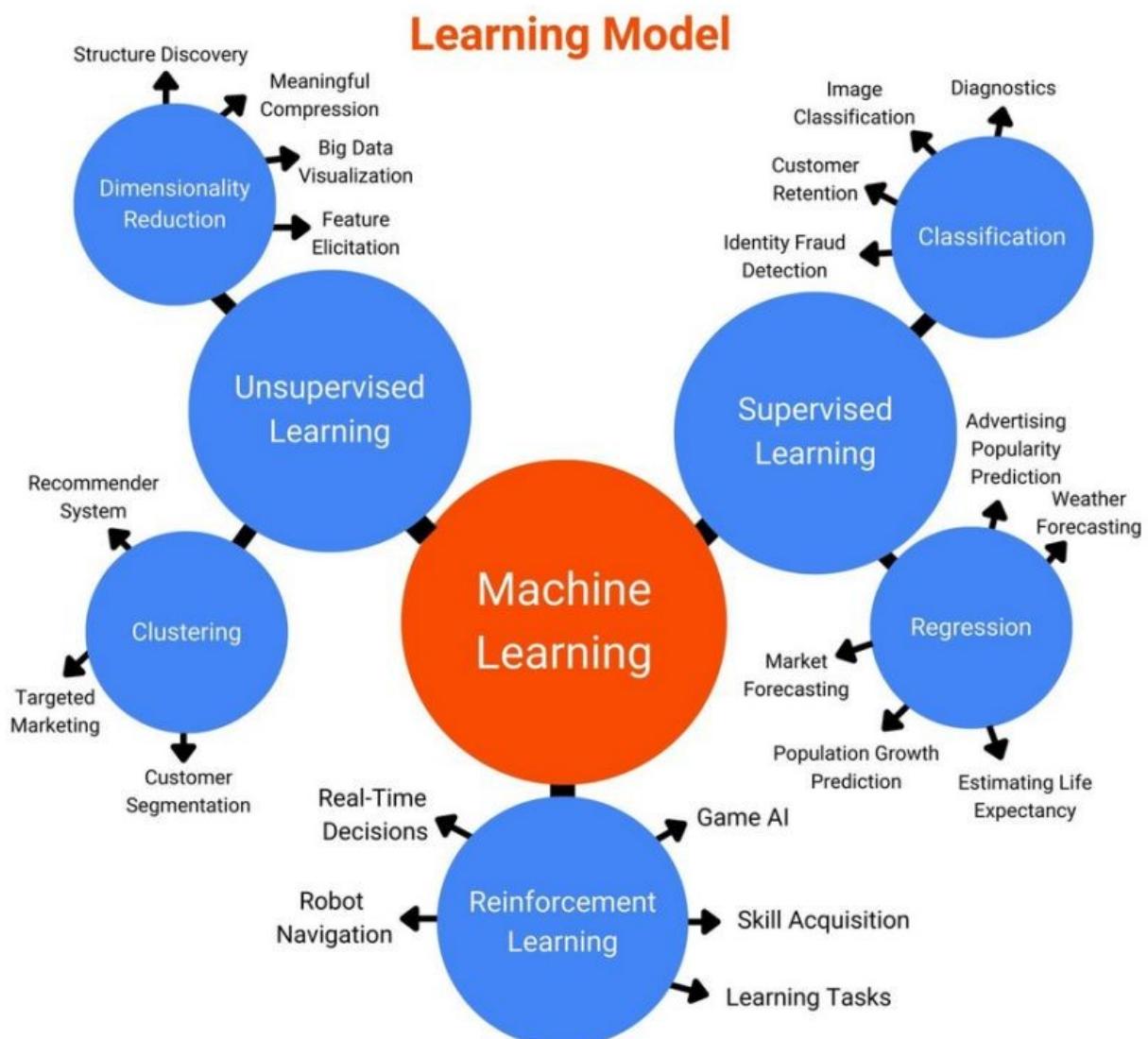


Figura 1 - Escolhendo o modelo correto de aprendizado de máquinas.

Fonte: Adaptado de [2].

Escolher o modelo certo de aprendizado de máquina é crucial para o sucesso de qualquer projeto orientado a dados. O processo de seleção envolve entender a natureza dos

dados, o problema que o pesquisador está tentando resolver e os pontos fortes e fracos de vários algoritmos, tais como [2].

- a) **Classificação:** Se a tarefa envolver prever rótulos discretos, considerar modelos como regressão logística, árvores de decisão, florestas aleatórias ou máquinas vetoriais de suporte (SVM).
- b) **Regressão:** Para prever valores contínuos, regressão linear, árvores de decisão e redes neurais são opções adequadas.
- c) **Agrupamento (*clustering*):** se precisar agrupar pontos de dados semelhantes sem rótulos predefinidos, algoritmos como K-Means ou *cluster* hierárquico são apropriados.
- d) **Redução da Dimensionalidade:** técnicas como PCA (análise de componentes principais) podem ser benéficas se você precisar reduzir o número de recursos, mantendo informações essenciais.
- e) **Tamanho do Conjunto de Dados:** alguns modelos, como algoritmos de aprendizado profundo, requerem grandes conjuntos de dados para ter um bom desempenho, enquanto modelos mais simples, como a regressão logística, podem funcionar efetivamente com conjuntos de dados menores.
- f) **Tipos de Características (*Feature Types*):** considere se os recursos são categóricos, numéricos ou uma mistura. Alguns algoritmos lidam com dados categóricos melhor do que outros (por exemplo, árvores de decisão).
- g) **Valores Ausentes (*Missing Values*):** algoritmos como a floresta aleatória podem lidar com valores ausentes de maneira mais eficaz do que outros, o que pode exigir conjuntos de dados completos.
- h) **Interpretabilidade:** Se a compreensão das decisões do modelo é crítica (por exemplo, na área da saúde), modelos mais simples, como árvores de decisão ou regressão logística, podem ser preferíveis.

- i) **Excedente (Overfitting):** modelos mais complexos (como aprendizado profundo) podem exceder se não forem gerenciados corretamente. Técnicas como validação cruzada e regularização podem ajudar a mitigar isso.
- j) **Tempo de Treinamento:** alguns modelos, como SVMs e redes de aprendizado profundo, podem ser computacionalmente intensivas e demoradas para treinar, especialmente em grandes conjuntos de dados.
- k) **Escalabilidade:** Considerar se o modelo pode escalar com eficiência com o aumento do volume de dados. Florestas aleatórias e máquinas de reforço de gradiente (*gradient boosting machine* - GBMs) são geralmente mais escaláveis que os SVMs.
- l) **Experiência:** o teorema “sem almoço grátis” sugere que nenhum algoritmo único funciona melhor para todos os problemas. É essencial experimentar vários algoritmos e comparar seu desempenho usando validação cruzada.
- m) **Métricas de avaliação:** escolher métricas apropriadas para avaliar o desempenho do modelo com base no tipo de problema. Para classificação, considere precisão, precisão, *recall* e pontuação de F1. Para regressão, usar o erro quadrado médio (*mean square error* – MSE) ou o R-Squared.
- n) **Engenharia de combate:** Invistir tempo na seleção e engenharia de recursos, pois a qualidade dos dados de entrada afeta significativamente o desempenho do modelo.
- o) **Melhoria Iterativa:** começar com modelos mais simples e evoluir gradualmente para os mais complexos, conforme necessário. Essa abordagem ajuda a entender os dados e refinar o modelo iterativamente.

Os modelos abordados são apresentados a seguir.

1. Regressão Linear (*Linear Regression* - LR)

Um modelo fundamental que prevê uma variável de resultado contínua com base em uma ou mais variáveis preditivas. Vide Figura 1.

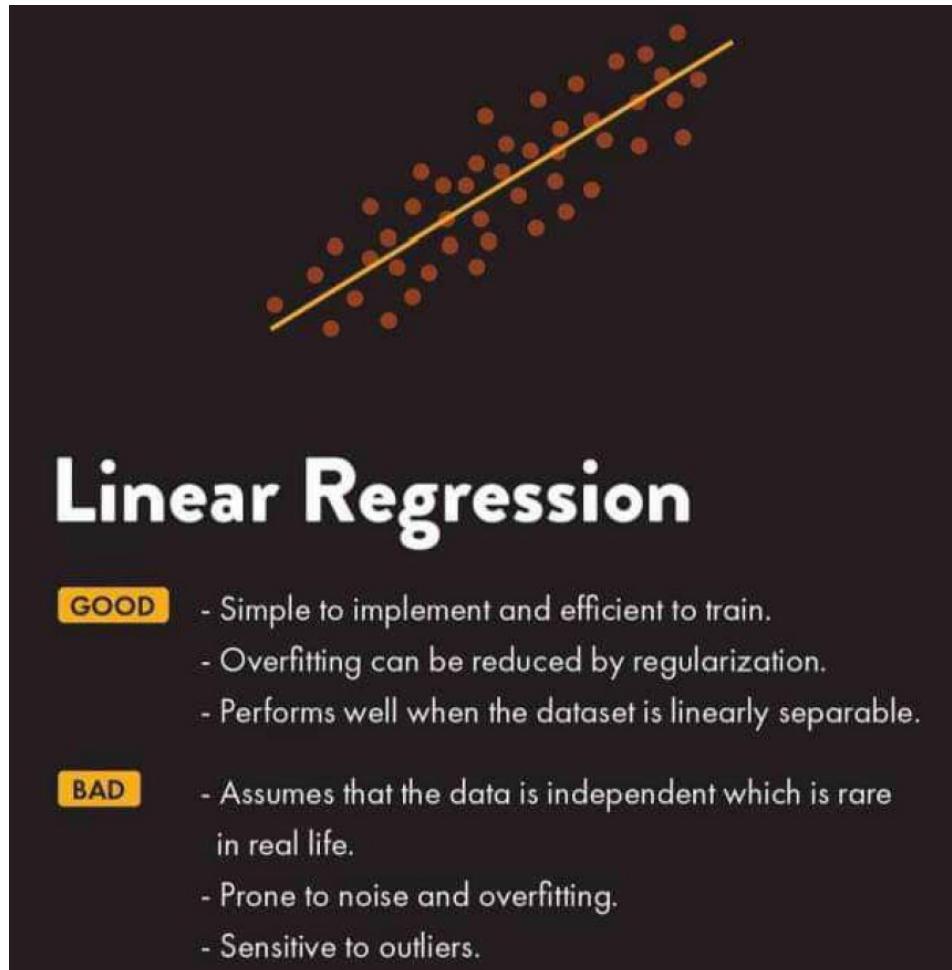


Figura 2 – Prós e Contras da Regressão Linear.
Adaptado de [1].

2. Regressão Logística (*Logistic Regression*)

Usado para tarefas de classificação binária. Estima a probabilidade de uma determinada instância pertencer a uma categoria específica.



Logistic Regression

GOOD

- Less prone to over-fitting but it can overfit in high dimensional datasets.
- Efficient when the dataset has features that are linearly separable.
- Easy to implement and efficient to train.

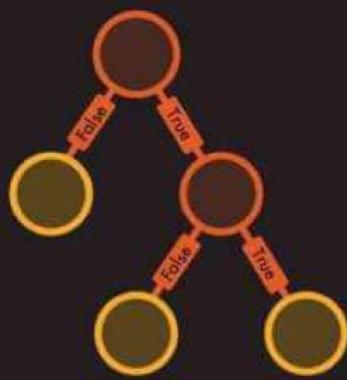
BAD

- Should not be used when the number of observations are lesser than the number of features.
- Assumption of linearity which is rare in practise.
- Can only be used to predict discrete functions.

Figura 3 – Prós e Contras da Regressão Logística.
Adaptado de [1].

3. Árvore de Decisão (*Decision Tree - DT*)

Uma estrutura do tipo Fluxchart, onde cada nó representa um recurso, cada ramificação de uma regra de decisão e cada folha uma etiqueta de classe



Decision Tree

GOOD

- Can solve non-linear problems.
- Can work on high-dimensional data with excellent accuracy.
- Easy to visualize and explain.

BAD

- Overfitting. Might be resolved by random forest.
- A small change in the data can lead to a large change in the structure of the optimal decision tree.
- Calculations can get very complex.

Figura 4 – Prós e Contras da Árvore de Decisão.

Adaptado de [1].

4. Vizinho k Mais Antigo (k Nearest Neighbour - KNN)

Classifica um ponto de dados com base em como seus vizinhos são classificados

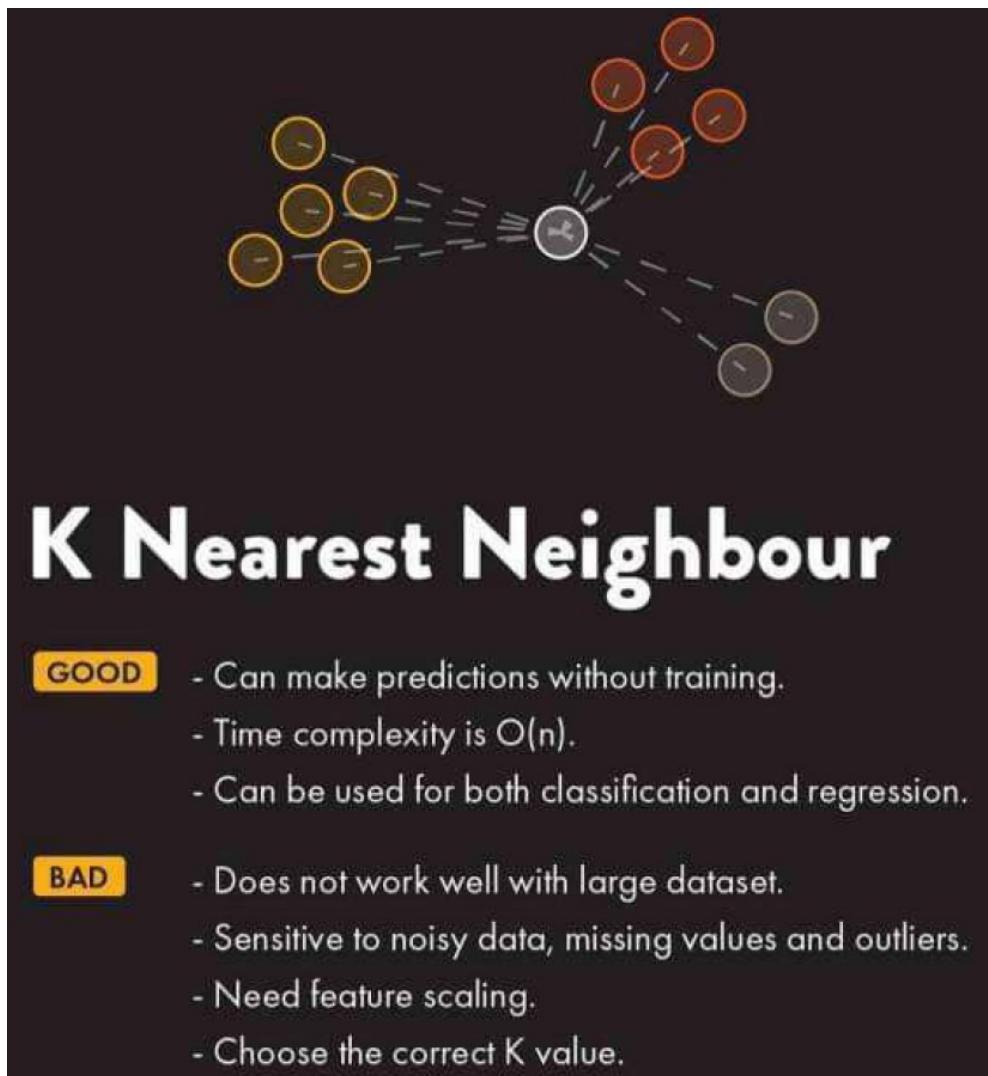


Figura 5 – Prós e Contras do KNN.

Adaptado de [1].

5. Média K (K-Means)

Um algoritmo de agrupamento (*cluster*) não supervisionado que agrupa os dados em 'k' número de centroides (*clusters*) definidos previamente. O termo “Means” refere-se à média dos pontos em cada *cluster* que determina a posição de seu centroide.

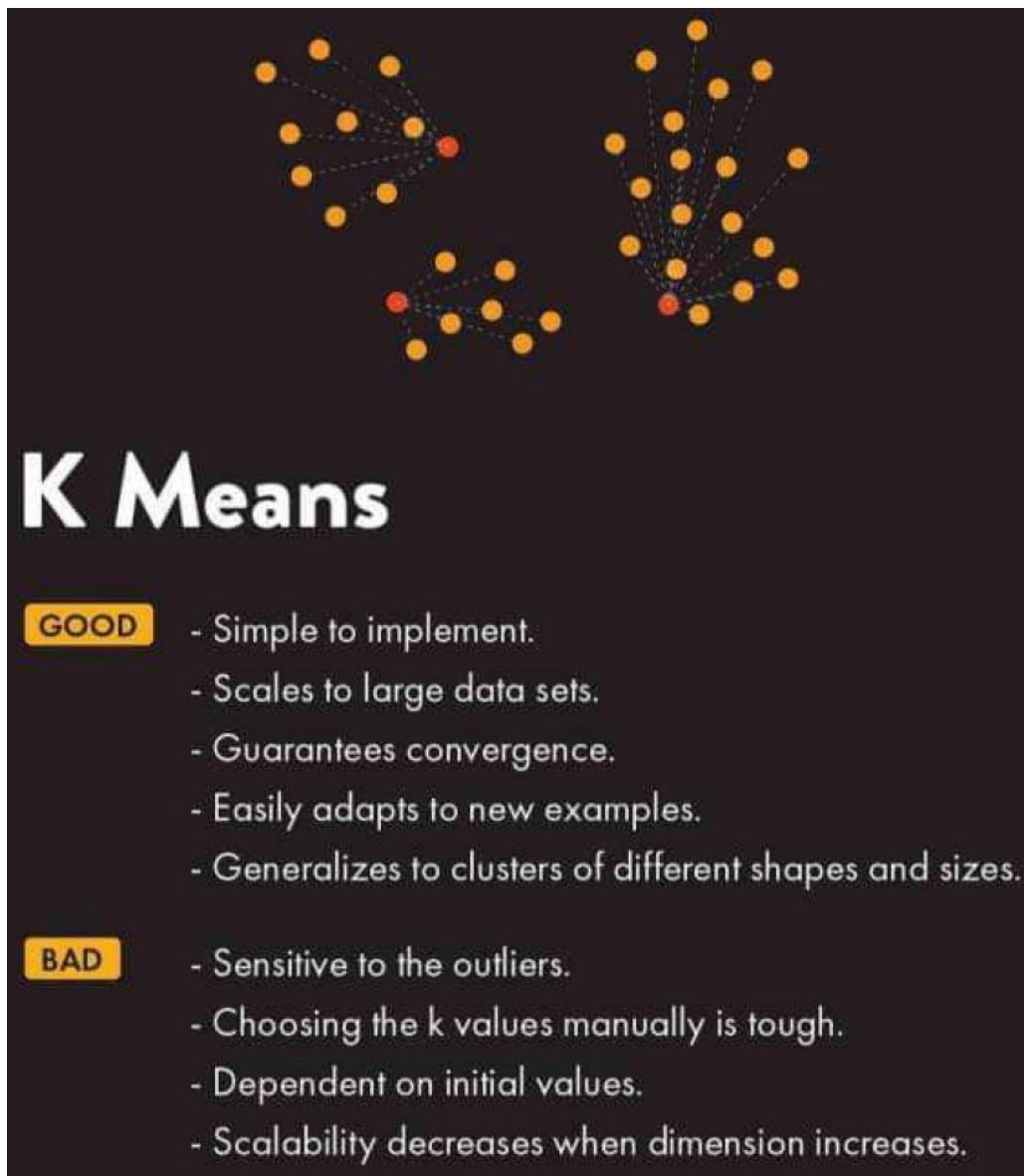


Figura 6 – Prós e Contras do K-Means.

Adaptado de [1].

6. Máquina de Suporte Vetorial (*Support Vector Machine - SVM*)

Encontra o hiperplano que melhor divide um conjunto de dados em classes.

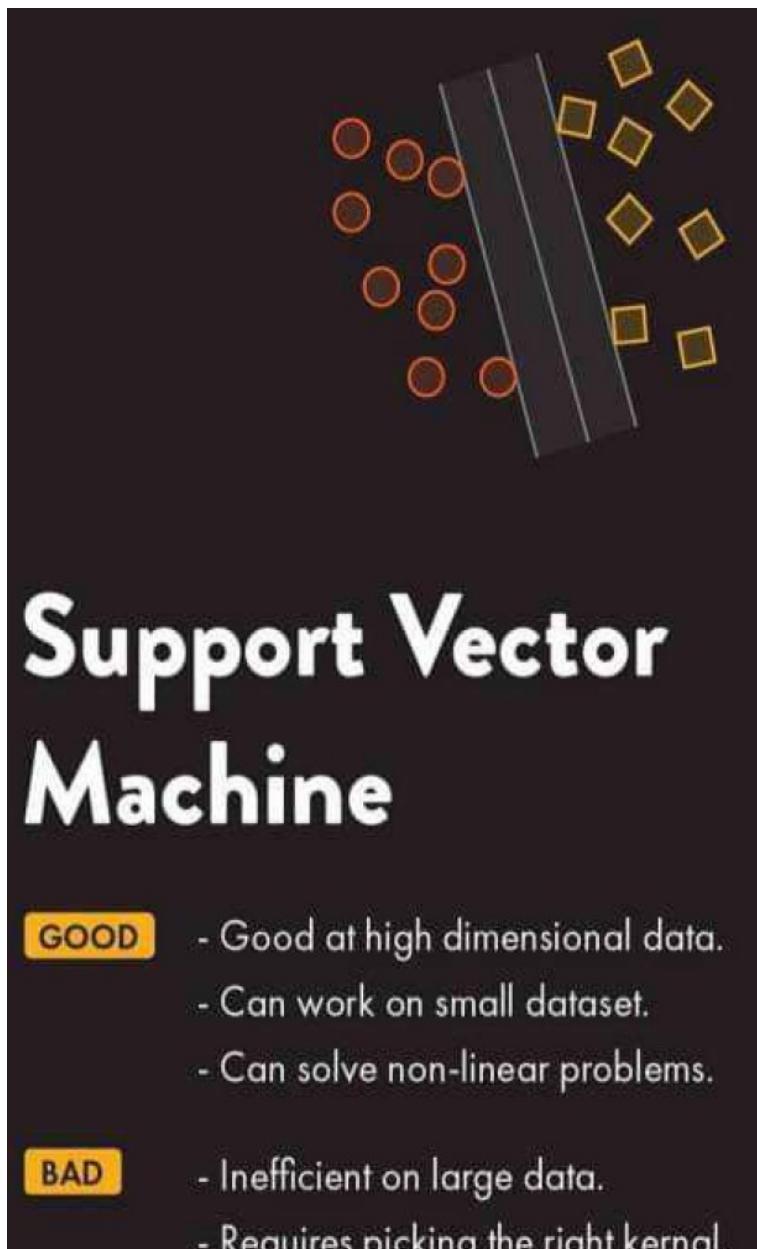
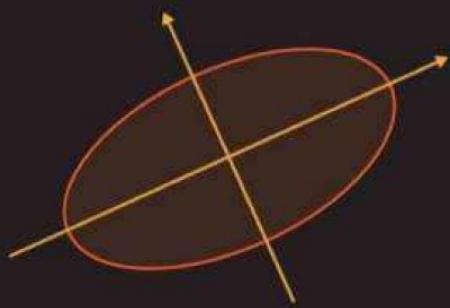


Figura 7 – Prós e Contras do SVM.

Adaptado de [1].

7. Análise de Componentes Principais (*Principal Component Analysis - PCA*)

Uma técnica de redução de dimensionalidade que transforma dados em um novo sistema de coordenadas.



Principal Component Analysis

GOOD

- Reduce correlated features.
- Improve performance.
- Reduce overfitting.

BAD

- Principal components are less interpretable.
- Information loss.
- Must standardize data before implementing PCA.

Figura 8 – Prós e Contras do PCA.

Adaptado de [1].

8. Bayes Ingênuos (*Naive Bayes*)

Baseado no teorema de Bayes, é particularmente adequado para conjuntos de dados de alta dimensão.

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)}$$

Naive Bayes

GOOD

- Training period is less.
- Better suited for categorical inputs.
- Easy to implement.

BAD

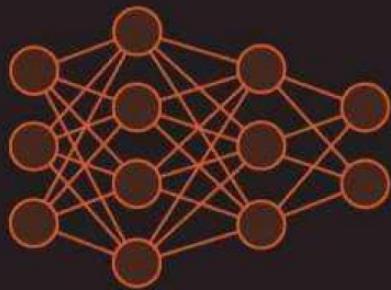
- Assumes that all features are independent which is rarely happening in real life.
- Zero Frequency.
- Estimations can be wrong in some cases.

Figura 9 – Prós e Contras do Naive Bayes

Adaptado de [1].

9. Redes neurais Artificiais – RNA (*Artificial Neural Networks - ANN*)

Inspirado no cérebro humano, consiste em neurônios interconectados.



ANN

GOOD

- Have fault tolerance.
- Have the ability to learn and model non-linear and complex relationships.
- Can generalize on unseen data.

BAD

- Long training time.
- Non-guaranteed convergence.
- Black box. Hard to explain solution.
- Hardware dependence.
- Requires user's ability to translate the problem.

Figura 10 – Prós e Contras das RNAs.

Adaptado de [1].

10. Adaboost

Um método de conjunto que ajusta os pesos dos pontos de dados classificados incorretamente.

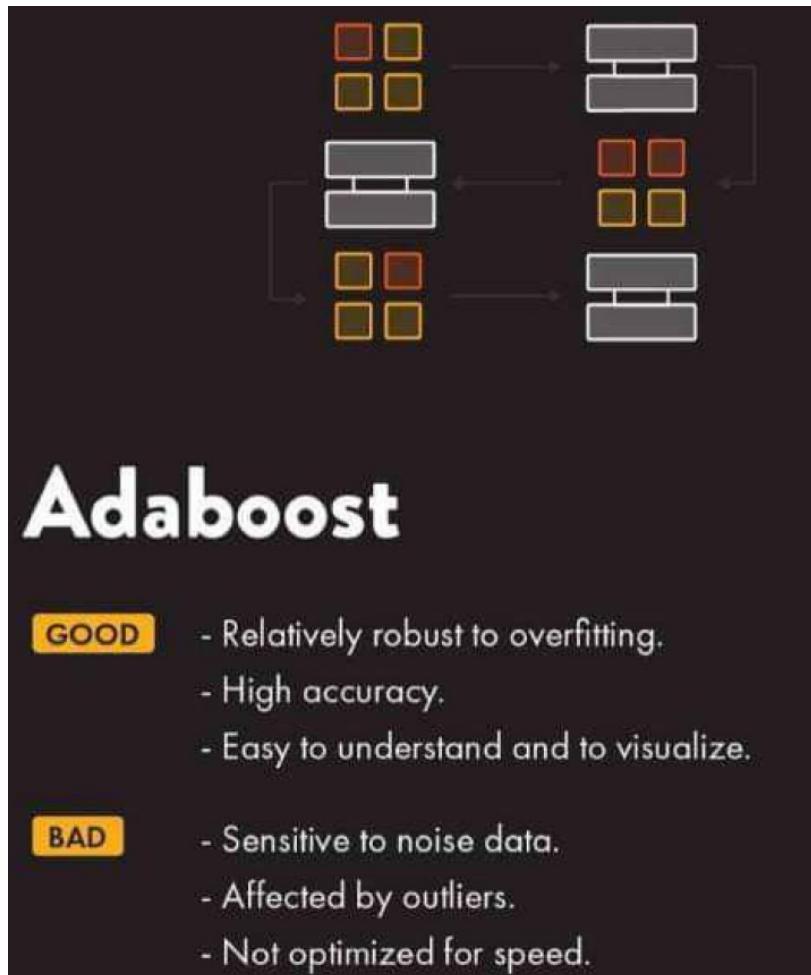


Figura 11 – Prós e Contras do Algoritmo AdaBoost.
Adaptado de [1].

2. RÁPIDA IMPLANTAÇÃO DE APRENDIZADO DE MÁQUINA

Como saber se o pesquisador está implantando seus modelos de aprendizado de máquina (machine learning - ML) de forma rápida e eficaz?

A Figura 12 apresenta um fluxograma do processo de modelagem de ML.

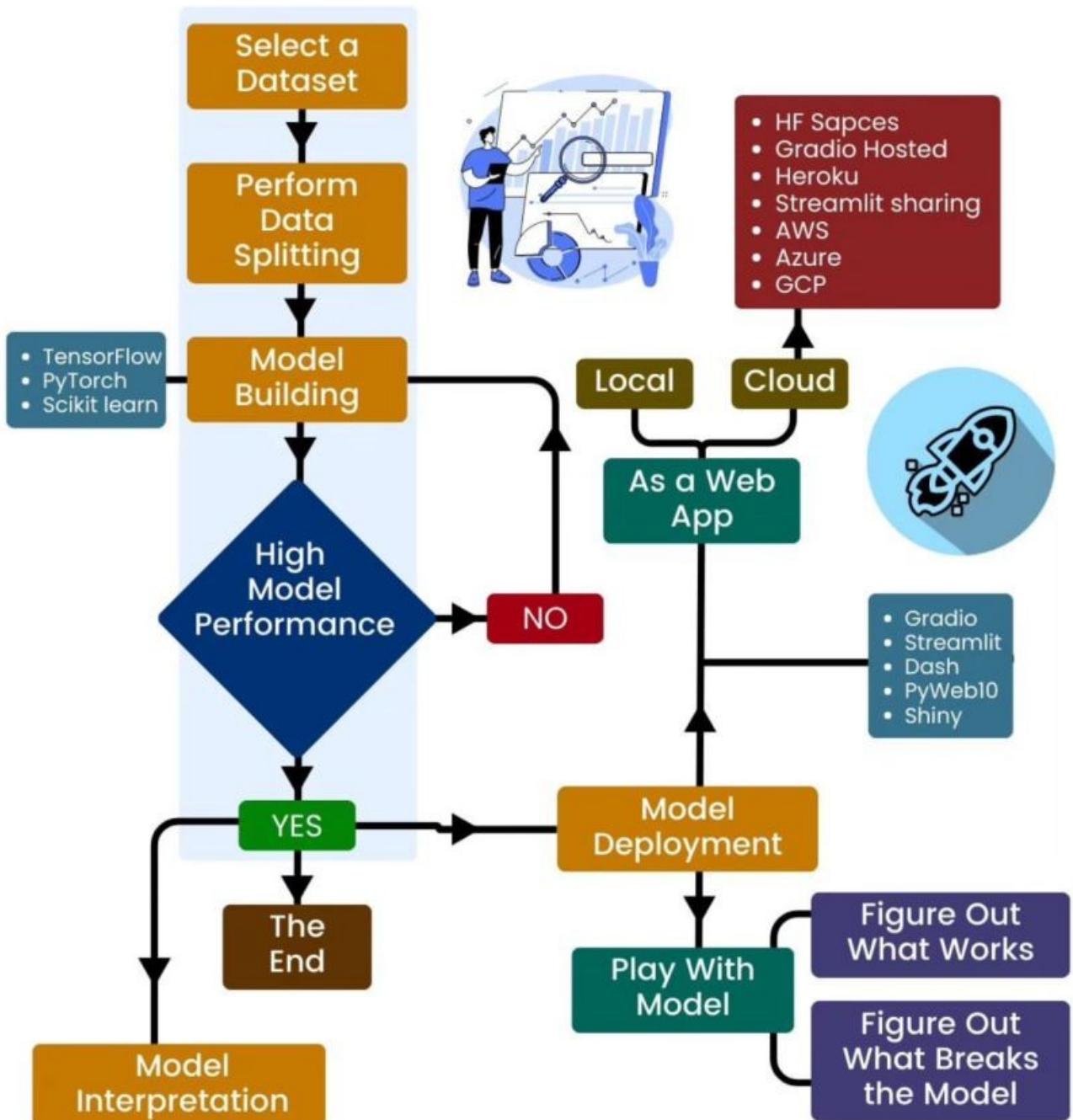


Figura 12 – Fluxograma para rápida implantação de modelo de ML.

Fonte: Adaptado de [3].

A capacidade de passar rapidamente da construção do modelo para a implantação é crucial. O fluxograma da Figura 12 simplifica o processo em etapas gerenciáveis, desde a seleção de um conjunto de dados e executando a divisão de dados, até a criação e a avaliação do desempenho.

- Se o modelo atingir o limite de desempenho, o pesquisador estará pronto para implantar o modelo, seja localmente ou na nuvem, como um aplicativo da web usando ferramentas como Graduio, Streamlit ou Shiny.
- Se o modelo não estiver tão parecido com o dataset, o processo incentiva a iteração:
 - Implante o modelo,
 - Altere parâmetros e valores, teste com ele,
 - Descubra o que funciona
 - Identifique o que o quebra.

Essa abordagem iterativa garante que, quando o pesquisador estiver pronto para implantar, o modelo será robusto e eficaz.

REFERÊNCIAS:

- [1] KHINVAZARA, A. Do you know which Machine Learning model is right for your project?, Linkedin Post, Jul 2024. Disponível em: https://www.linkedin.com/posts/aditi-khinvasara-bb99601aa_linear-regression-activity-7223260299011739648-FQ1h?utm_source=share&utm_medium=member_desktop. Acessado em Jul 29, 2024.
- [2] NAYAK, M. Choosing the Right Machine Learning Model. Artificial Intelligence, Deep Learning, Machine Learning (Linkedin Group), 2024. Disponível em: https://media.licdn.com/dms/image/D5622AQGL-9-fGPYufw/feedshare-shrink_800/0/1722322659514?e=1725494400&v=beta&t=3Db4ADMztRwmTQ2JnE4FlsnknXjzxZ9zcdRml9NOppQ. Acessado em Jul 30, 2024.
- [3] KHINVASARA, A. Quickly Deploy ML Models, Aditi Khinvasara (Linkedin Posts), The Ravit Show, 2024. Disponível em:<https://media.licdn.com/dms/image/v2/D5622AQGo2jrB2kdnmA/feedshare-shrink_800/feedshare-shrink_800/0/1724749974172?e=1727913600&v=beta&t=0dUBoG3h7zFRh46mKqecIRBd3vWDTPQiXNVYB6pDGM>. Acessado em Ago 29, 2024.

33. REDES NEURAIS CONVOLUCIONAIS

Editado por: Dr. Arnaldo de Carvalho Junior, Out 03, 2024.

1. COMO OS COMPUTADORES “ENXERGAM”?

Quando se olha para uma imagem, se reconhece instantaneamente rostos, objetos e padrões. Para um computador, porém, não é tão simples.

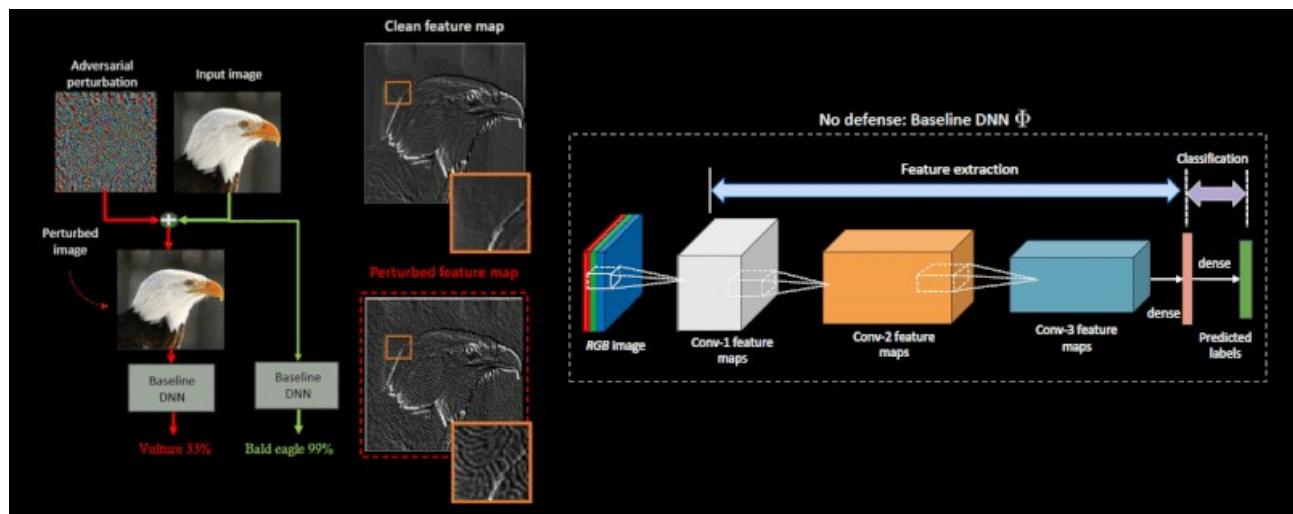


Figura 1 – Como o computador enxerga.

Fonte: Adaptado de [1].

Imagine a imagem da Figura 1. Enquanto os humanos veem objetos familiares, um computador vê uma grade de valores de pixels, cada um representando o quanto claro ou escuro é um ponto. Esses números compõem a "visão" bruta do computador. Então, como os computadores entendem esses números? Para isso entram as redes neurais convolucionais (*convolutional neural network* - CNNs), a espinha dorsal do processamento moderno de imagem.

2. O QUE É UMA CNN?

Uma CNN é um tipo especializado de algoritmo de aprendizado profundo (*deep learning* - DL) projetadas para interpretar imagens, permitindo que as máquinas identifiquem objetos, faces e padrões complexos nas imagens. Inspiradas no córtex visual humano. As CNNs usam operações de convolução para extrair recursos e identificar padrões dentro

das imagens. A CNN é um tipo de rede neural do tipo “olhar para a frente” (*feedforward neural network* - FNN) que aprende uma característica por meio de otimização de filtros [2]. Essa arquitetura permite que as CNNs interpretem e classifiquem imagens com eficiência, tornando-as inestimáveis em aplicativos de visão computacional. A Figura 2 apresenta uma arquitetura de uma CNN.

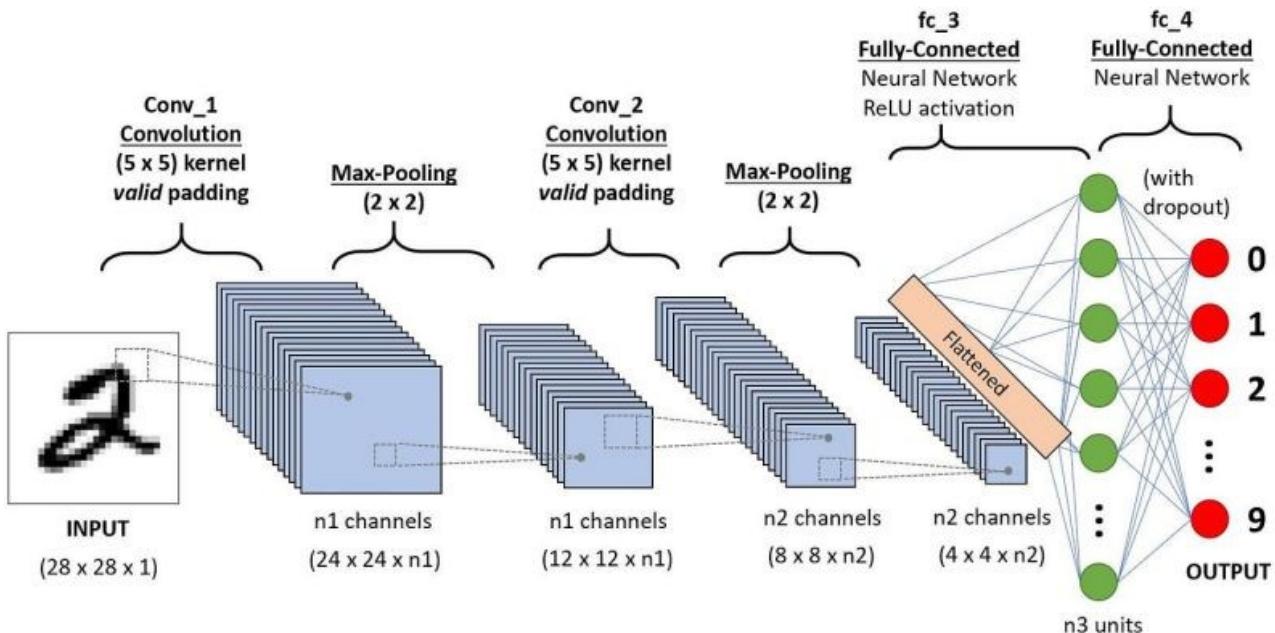


Figura 2 – Arquitetura de CNN

Fonte: Adaptado de [2].

3. CAMADAS-CHAVE DA CNN

a) Camada convolucional (*convolutional layer*): este é o bloco de construção principal de uma CNN. Essa camada digitaliza a imagem, agindo como um filtro que detecta recursos fundamentais, como bordas, cantos e texturas. É semelhante à maneira como os olhos do ser humano primeiro pegam contornos e formas ásperas. Os filtros (*kernels*) deslizam sobre a imagem de entrada para detectar recursos como bordas e texturas. A saída é um mapa de recursos que destaca esses recursos detectados [1].[2],[3].

b) Camada de agrupamento (*pooling layer*): após a camada convolucional, as camadas de agrupamento diminuem os mapas de recursos, reduzindo suas dimensões espaciais. Esta etapa diminui a complexidade computacional e ajuda a evitar o excesso de ajuste. Depois que os recursos são detectados, a coleta de zooms, reduzindo os dados enquanto preservava as principais informações. É como resumir a imagem sem perder o essencial. O máximo de pool, que seleciona o valor máximo de uma região, é uma técnica comum usada aqui [1].[2],[3].

c) Camada de ativação da unidade retificadora linear (*rectified linear unit* - ReLU):

Funções de ativação como Relu ou Sigmoid ajudam a rede a se concentrar em padrões importantes, orientando o modelo a uma previsão precisa. A ReLU introduz a não linearidade no modelo, permitindo que ele aprenda padrões e relacionamentos complexos nos dados. A ReLU é uma das funções de ativação mais utilizadas atualmente em redes neurais pela sua rapidez no processo de aprendizagem [1],[4].

d) Camada totalmente conectada (*fully connected layer*): Após o pool, os dados restantes são achatados em um único vetor, que é passado para as camadas totalmente conectadas. Essas camadas reúnem todos os recursos detectados para fazer a previsão final, como identificar um cão, carro ou rosto. Nos estágios finais, as camadas totalmente conectadas usam os recursos de alto nível aprendidos pelas camadas anteriores para fazer previsões. Cada neurônio nessas camadas está conectado a todos os neurônios da camada anterior, integrando todos os recursos extraídos para produzir a saída final [1].[2], [3].

3. CNN X ANN:

Embora as redes neurais artificiais (*artificial neural networks* - ANNs) sejam versáteis, elas lutam com dados de alta dimensão, como imagens devido à sua natureza totalmente conectada. As CNNs, no entanto, são projetadas especificamente para lidar com estruturas de dados semelhantes à grade, tornando-as altamente eficientes para o processamento de imagens. Sua capacidade de capturar hierarquias espaciais e padrões locais os diferencia das ANNs tradicionais [4].

3.1. Por que não usar a ANN para dados de imagem?

As ANNs exigem engenharia extensa de recursos e lutam com a alta dimensionalidade dos dados da imagem. Por outro lado, as CNNs aprendem automaticamente representações hierárquicas de recursos, melhorando significativamente a precisão e reduzindo a necessidade de intervenção manual [4].

4. APLICAÇÕES DA CNN

Do reconhecimento facial à direção autônoma, os CNNs estão por trás de muitas inovações [1]. As CNNs transformaram indústrias com sua alta precisão em tarefas relacionadas à imagem [4]. Elas são amplamente utilizadas em:

- Análise de imagem médica: auxiliando no diagnóstico de doenças através do reconhecimento de imagem.
- Veículos autônomos: permitindo a detecção de objetos e o entendimento da cena.
- Reconhecimento facial: alimentando sistemas de segurança e autenticação.

Além das aplicações que envolvem imagens, existem pesquisas utilizando CNNs também em processamento de linguagem natural (*natural language processing – NLP*), interfaces cérebro-computador e séries temporais como financeiras, por exemplo.

5. CONCLUSÃO

As CNNs continuam a ultrapassar os limites do que é possível na aprendizagem profunda, oferecendo soluções poderosas para desafios complexos de dados visuais.

REFERÊNCIA

- [1] ALAM, A. How Computers See Images & How CNNs Think. Linkedin Profile, 2024. Disponível em: https://media.linkedin.com/dms/image/v2/D5622AQF6p44ruXDEfA/feedshare-shrink_800/feedshare-shrink_800/0/1727934055718?e=1730937600&v=beta&t=N-2yDnMFnNTdQrMdCZL3Mdfp3IFSJHcLYZT7Fg-TRNo. Acessado em Out 03, 2024.
- [2] CARVALHO JUNIOR, A. Redes Neurais Artificiais: Algoritmos poderosos para aplicações de IA e ML. EAILab Posts. 2024. Disponível em: <<https://eailab.labmax.org/2024/04/03/redes-neurais-artificiais-algoritmos-poderosos-para-aplicacoes-de-ia-e-ml/>>. Acessado em Ago 13, 2024.
- [3] VERMA, J. Convolutional Neural Networks, Machine Learning Community (Linkedin Group), 2024. Disponível em: <https://media.linkedin.com/dms/image/v2/D4D22AQHp54hbofrRjQ/feedshare-shrink_800/feedshare-shrink_800/0/1723566397982?e=1726704000&v=beta&t=pme9UxCdvRgXd3jNoGq1ETfuRRxVqUYpxY9P9h09Pb8>. Acessado em Ago 13, 2024.
- [4] CARVALHO JUNIOR, A. Função de Ativação, o Núcleo da Composição de Neurônios Artificiais. EAILab Posts. 2024. Disponível em: <<https://eailab.labmax.org/2024/02/28/funcao-de-ativacao-o-nucleo-da-composicao-de-neuronios-artificiais/>>. Acessado em Ago 13, 2024.

34. ALGORITMOS FREQUENTEMENTE UTILIZADOS PARA APRENDIZADO PROFUNDO

Por: Abhishek Chandragiri

Editado por: Dr. Arnaldo de Carvalho Junior, Julho 19, 2024.

1. Algoritmos de aprendizado profundo: alimentando a inovação da IA

Enquanto a IA continua a remodelar as indústrias, é crucial entender os principais algoritmos que impulsionam essa revolução.

Este artigo apresenta uma visão geral concisa dos modelos de aprendizado profundo (*deep learning - DL*) frequentemente usados.

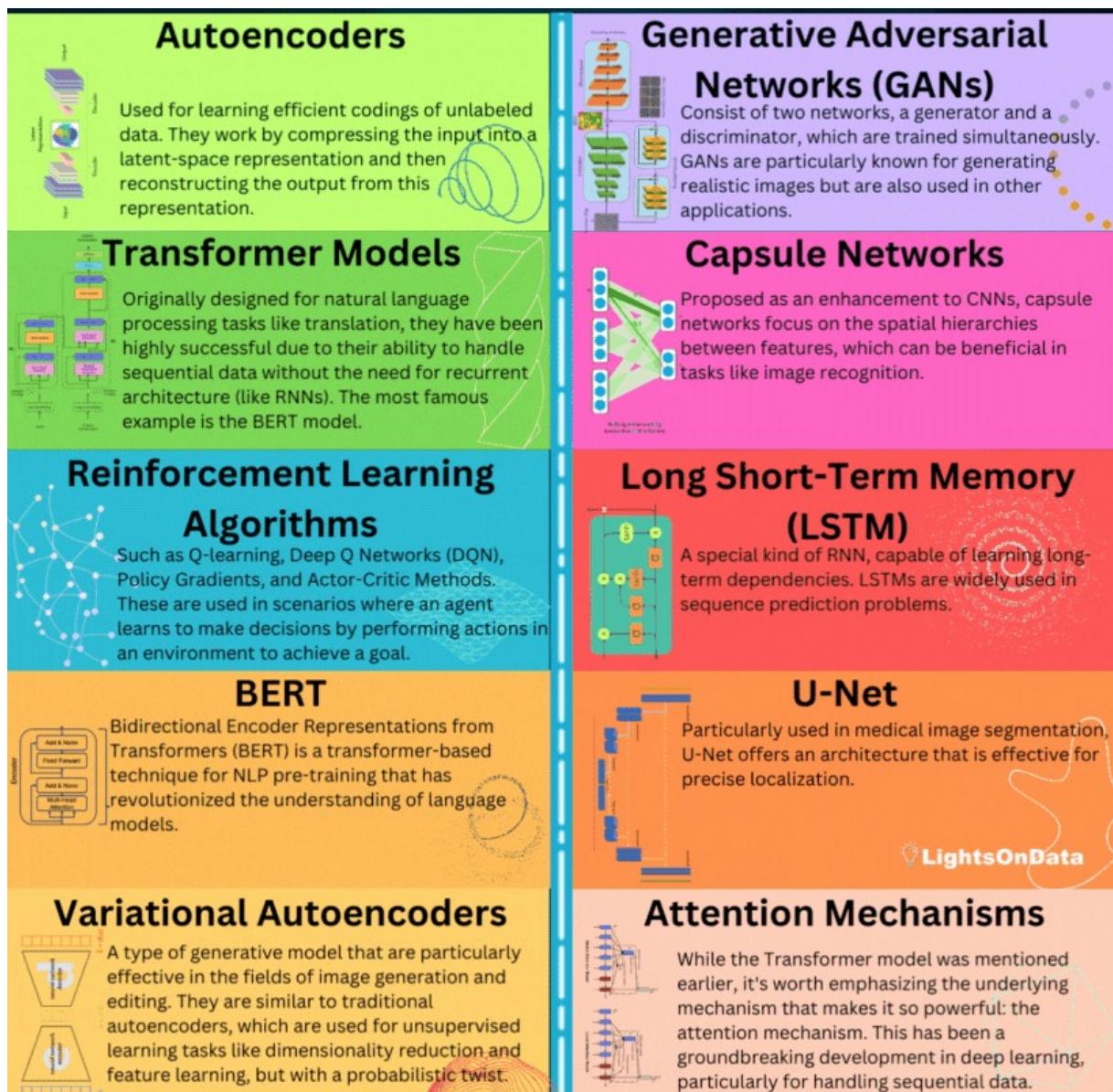


Figura 1 - Algoritmos usados frequentemente em aprendizado profundo

Fonte: Adaptado de [1].

1.1. Autocodificadores (autoencoders): esses algoritmos comprimem e reconstruem dados, aprendendo representações eficientes. Eles são inestimáveis para a redução da dimensionalidade e o aprendizado de recursos em configurações não supervisionadas.

1.2. Redes Adversárias Generativas (generative adversarial networks - GAN): GANS consistem em duas redes neurais concorrentes - uma geradora e uma discriminadora. Eles se destacam na criação de dados sintéticos realistas, particularmente na geração de imagens.

1.3. Transformadores (Transformers): originalmente projetados para processamento de linguagem natural, os Transformers revolucionaram o manuseio de dados seqüenciais. Seu mecanismo de atenção permite que eles processem dependências de longo alcance com eficiência, tornando-as altamente eficazes para tarefas como tradução e geração de texto.

1.4. Redes de cápsulas (Capsule Networks): Essas redes têm como objetivo abordar as limitações dos CNNs modelando relacionamentos hierárquicos. Eles mostram promessas na preservação de relacionamentos espaciais em tarefas como reconhecimento de imagem.

1.5. Aprendizagem de reforço (Reinforcement Learning): Este paradigma treina agentes para tomar decisões, recompensando comportamentos desejados em ambientes simulados. É particularmente poderoso para cenários que exigem tomada de decisão estratégica, como jogo ou robótica.

1.6. Memória Longa de Prazo Curto (Long Short-Term Memory - LSTM): um tipo de rede neural recorrente, os LSTMs são projetados para capturar dependências de longo prazo em dados seqüenciais. Eles são amplamente utilizados em aplicativos como reconhecimento de fala e previsão de séries temporais.

1.7. Representações Bidirecionais do Codificador dos Transformadores (Bidirectional Encoder Representations from Transformers – BERT): O BERT estabeleceu novos parâmetros de referência nas tarefas de entendimento de linguagem

natural. É pré-treinado em vastas quantidades de texto, permitindo capturar contextos de linguagem sutis.

1.8. Rede U (U-NET): É uma rede neural artificial convolucional (convolutional neural network – CNN) que foi originalmente desenvolvida para segmentação de imagens biomédicas. Esta arquitetura é particularmente eficaz para tarefas de segmentação de imagens. É amplamente utilizado em imagens médicas para localização precisa e segmentação de estruturas. A rede consiste em um caminho de contração (*contraction*) e um caminho expansivo (*expansion*), o que lhe confere a arquitetura em forma de U. O caminho de contração é uma CNN típica que consiste em aplicação repetida de convoluções, cada uma seguida por uma unidade linear retificada (*Rectifier Linear Unit - ReLU*) e uma operação de pool máximo. Durante a contração, as informações espaciais são reduzidas enquanto as informações de recursos são aumentadas. O caminho expansivo combina o recurso e as informações espaciais através de uma sequência de devoluções e concatenações com recursos de alta resolução do caminho de contratação.

1.9. Auto-Codificadores Variacionais (*Variational Autoencoders*): esses aspectos combinam aspectos dos autoencoders com modelagem probabilística. Eles são particularmente úteis para gerar novas amostras de dados e executar tarefas complexas de edição de imagens.

1.10. Mecanismos de Atenção (*Attention Mechanisms*): permitem que os modelos se concentrem em partes relevantes dos dados de entrada ao processar sequências. Eles são um componente chave dos transformadores e melhoraram drasticamente o desempenho em tarefas envolvendo dados sequenciais.

REFERÊNCIAS

- [1] CHANDRAGIRI, A. Frequently Used Deep Learning Algorithms. Lights on Data, 2024. Disponível em: https://media.lcdn.com/dms/image/D5622AQH5FXoND-XLIg/feedshare-shrink_1280/0/1721357249064?e=1724284800&v=beta&t=6bU5k1oj7I83TArsvwyorCB-QtejSaCRQuhkCoZsm7Q Acessado em Jul 19, 2024.

35. ARQUITETURA DOS TRANSFORMADORES

Editado por: Dr. Arnaldo de Carvalho Junior

Data: Julho 30, 2024.

A arquitetura dos transformadores (*Transformers*) explicada. 

O básico da arquitetura do transformador reside no conceito de codificador/decodificador.

1. O que é o Transformador?

Um transformador é uma rede neural que se destaca em entender o contexto de dados sequenciais e gerar novos dados a partir dela.

Eles são os primeiros a confiar apenas em auto-distribuição, sem usar redes neurais recorrentes (*recurrent neural networks* - RNNs) ou redes neurais convolucionais (*convolution neural networks* – CNNs).

2. Transformador como uma Caixa Preta

Pode-se imaginar um transformador para a tradução de idiomas como uma caixa-preta.

- Entrada: uma frase em um idioma.
- Saída: sua tradução.

Mas o que acontece dentro desta caixa-preta?

3. Arquitetura Codificador / Decodificador (*Encoder / Decoder Architecture*)

- Entrada: frase em espanhol “De quién es?”
- Codificador: o transforma em um formato estruturado que captura sua essência.
- Decodificador: recebe esses dados codificados e gera a tradução.
- Saída: a frase traduzida: “De quem é?”

4. A Arquitetura Por Trás dos Transformadores

Cada codificador e decodificador são compostos de camadas. Aqui está como eles funcionam:

zz) Codificadores: processe a entrada sequencialmente, camada por camada.

aaa) Decodificadores: pegue os dados codificados e gere o passo a passo.

Ambos usam redes neurais de autoatendimento e avanço, permitindo a geração de linguagem natural.

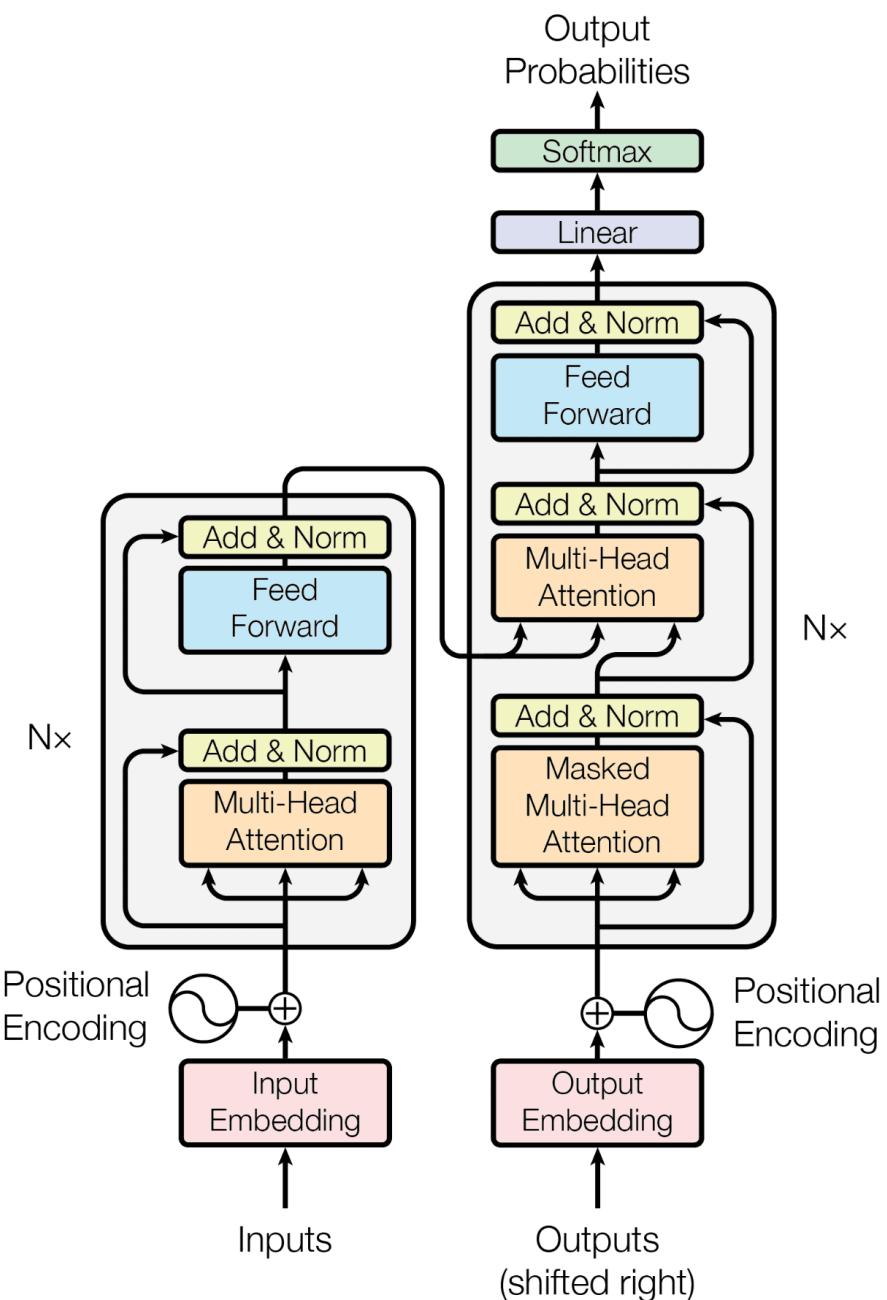


Figura 1 – O Modelo Transformer
Adaptado de [1].

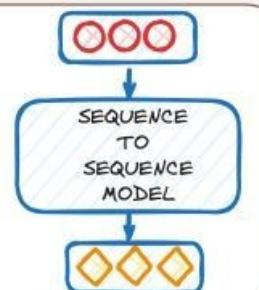


ML BASICS - THE TRANSFORMER ARCHITECTURE

#1 What is a Transformer?

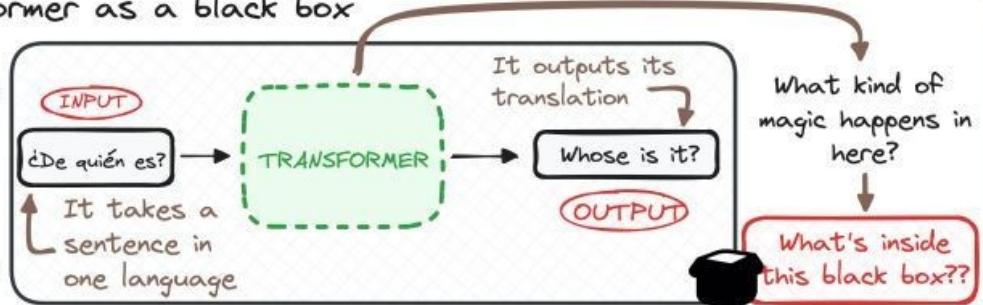
It is a type of neural network that excels at understanding the context of sequential data and generating new data from it.

They are the first transduction models to rely entirely on self-attention, without the usage of RNNs nor convolution.



#2 Transformer as a black box

Let's image a Transformer for language translation as a BLACK BOX



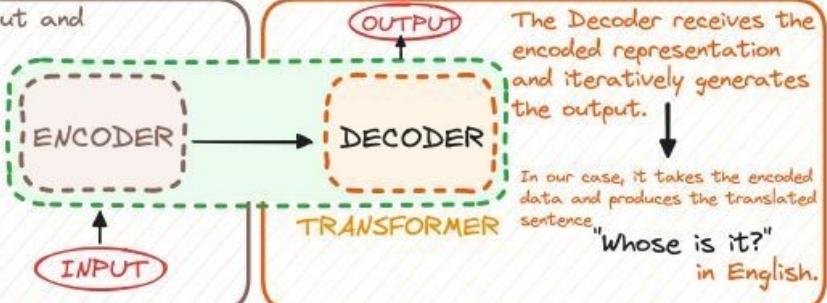
#3 The Encoder / Decoder architecture

The Encoder takes the input and converts it into a matrix representation.

But... what does this mean?

It processes the Spanish sentence "¿De quién es?"

Then it transforms it into a structured format that captures the essence of the sentence.

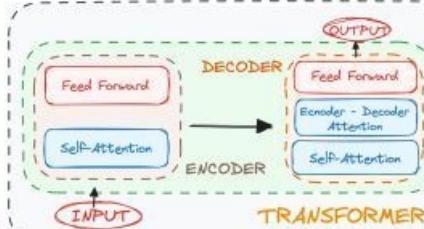


#4 The architecture behind the Transformers

Each encoder follows an identical structure, sequentially processing the input through each layer. Then the input passes to the next one.

Both the encoder and decoder are stacks comprising multiple layers. Each stack containing the same number of layers ($N=6$ in the original paper).

Each decoder follows a uniform structure, receiving input from the final encoder layer and the previous decoder layer.



Both the encoder and the decoder have an internal architecture that makes possible the generation of natural language.

TRANSFORMERS ARCHITECTURE IN DETAIL

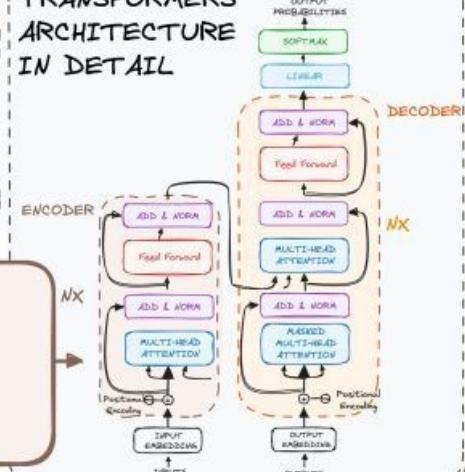


Figura 2 – Arquitetura Transformer Explicada.

Adaptado de [2].

REFERÊNCIAS:

- [1] STEFANIA, C. The Transformer Model. Machine Learning mastery. Disponível em: <https://machinelearningmastery.com/the-transformer-model/>. Acessado em Jul 30, 2024.
- [2] FERRER, J The Transformers Architeture Clearly Explained. Machine learning Community (Linkedin Group), Jul 2024. Disponível em: https://media.linkedin.com/dms/image/D4D22AQEP0sRasr5ijg/feedshare-shrink_2048_1536/0/1722175071467?e=1725494400&v=beta&t=l16mvO8m79ITackZwwNlxsb1fEcM_koaRLuQeqQ1Ds. Acessado em Jul 30, 2024.

36. EVOLUÇÃO DA ARQUITETURA DE IA: DO ML TRADICIONAL PARA A IA GENERATIVA

Editado por: Dr. Arnaldo de Carvalho Junior

Data: Julho 22, 2024.

A Figura 1 apresenta uma comparação entre a arquitetura clássica de aprendizado de máquinas (machine learning – ML) e a inteligência artifical (IA) generativa.

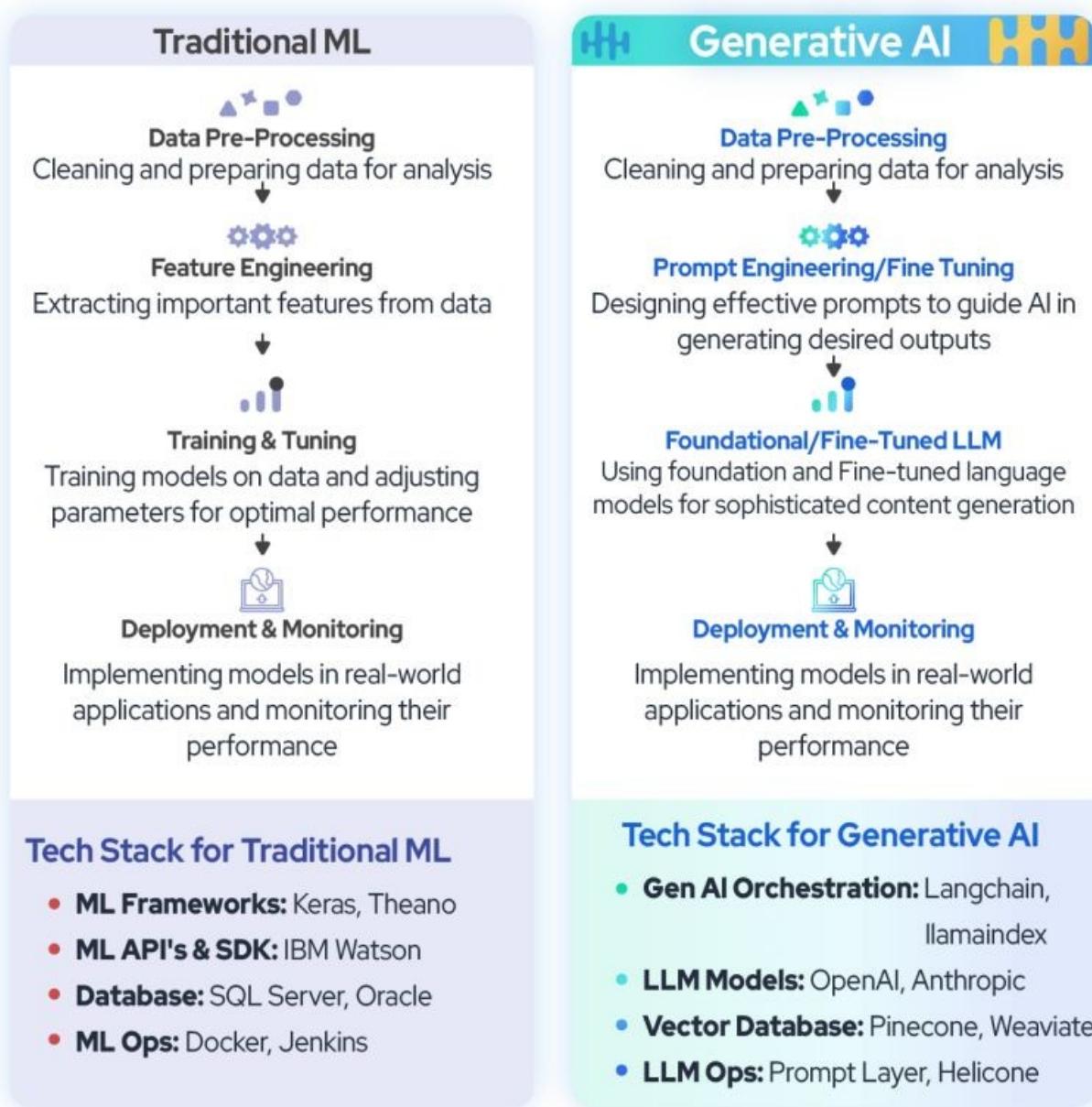


Figura 1 - Evolução da Arquitetura de IA: Do ML Tradicional para a IA Generativa.

Fonte: Adaptado de [1].

A seguir são apresentados os pontos que indicam a fascinante evolução do aprendizado de máquina tradicional até a IA generativa de ponta:

a) ML tradicional: a base da IA, concentrando-se no pré-processamento de dados, engenharia de recursos e treinamento de modelos.

- bbb) Pré-processamento de dados: limpeza e preparação de dados para análise;
- ccc) Engenharia de recursos: extraindo recursos importantes dos dados;
- ddd) Treinamento e ajuste: otimizando parâmetros do modelo para desempenho;
- eee) Implantação e monitoramento: implementando modelos em aplicativos do mundo real.

b) AI generativa: a próxima fronteira, alavancando a engenharia rápida e os grandes modelos de idiomas (*large language models* - LLMs) para criar conteúdo sofisticado.

- Pré-processamento de dados: semelhante ao ML tradicional;
- Engenharia imediata/ajuste fino: projetando instruções eficazes para orientar as saídas de IA;
- LLM fundamental/ajustado: Usando modelos de linguagem sofisticados para geração de conteúdo;
- Implantação e monitoramento: implementação e rastreamento de desempenho em aplicativos.

c) Diferenças-chave:

- Engenharia de recursos versus engenharia imediata: o ML tradicional se concentra na identificação e extração de recursos relevantes dos dados, enquanto a IA generativa enfatiza a elaboração de instruções eficazes para direcionar a saída da IA.
- Treinamento e ajuste vs. LLMs de ajuste fino: o ML tradicional se concentra na identificação e extração de recursos relevantes dos dados, enquanto a IA generativa enfatiza a elaboração de instruções eficazes para direcionar a saída da IA.
- ML Frameworks vs. Gen AI Orquestração: As ferramentas e estruturas diferem, com o ML tradicional usando estruturas como Keras e Theano e IA generativa usando ferramentas de orquestração como Langchain e Llandeindex.

d) Exemplos do mundo real:

- ML tradicional: análise preditiva em finanças, sistemas de recomendação em comércio eletrônico e ferramentas de diagnóstico na área da saúde.
- IA generativa: criação de conteúdo (por exemplo, artigos de redação, geração de imagens), assistentes virtuais e tradução avançada de linguagem.

Ambos compartilham etapas cruciais no pré-processamento e implantação de dados, mostrando a inovação contínua na IA.

REFERÊNCIAS:

[1] ALPHA, Evolution of AI Architecture: From Traditional ML to Generative AI. AIM Research, 2024. Disponível em:

https://media.linkedin.com/dms/image/D4D22AQGDZoBZUnZw9w/feedshare-shrink_2048_1536/0/1721830814125?e=1725494400&v=beta&t=5s220FF4F-afkmpoo5Ax5RnUf99jC2Lt0-wYchh7wCs. Acessado em Ago 01, 2024.

37. AS OPERAÇÕES DE APRENDIZADO DE MÁQUINA (MLOps)

Por: Deepak Bhardwaj

Editado por: Dr. Arnaldo de Carvalho Junior

Data: Julho 24, 2024.

O que são operações de aprendizado de máquina (*machine learning operations – MLOps*)?



Figura 1 - Operações Fim a Fim de Aprendizado de Máquinas

Fonte: Adaptado de [1].

A Figura 1 apresenta um detalhamento visual que simplifica a jornada complexa desde os dados brutos até modelos implementáveis.

Etapas críticas em MLOps:

- Ingestão de Dados (*Data Ingestion*): Coleta dados de várias fontes.
- Preparação de Dados (*Data Preparation*): preparando dados para análise.
 - Validar: Certifique-se de que os dados estão corretos.
 - Limpar: Remova erros e inconsistências.
 - Padronizar: Uniformizar os dados.
 - Curadoria: Organize os dados de forma eficaz.
 - Anonimizar: Proteja informações pessoais.
- Lago de Dado e Delta (*Data Lake & Delta Lake*): Armazenar dados brutos e processados.
- Recursos de Engenharia (*Feature Engineering*): Cria recursos de dados úteis.
 - Recursos de extração: selecione pontos de dados essenciais.
 - Dividir conjunto de dados: divida os dados para treinamento e teste.
- Treinamento do Modelo (*Model Training*): Desenvolver e refinar modelos.
 - Código: Escrever algoritmos.
 - Treinar: Ensinar modelos usando dados.
 - Avaliar o desempenho do modelo.
 - Otimizar: Melhore a precisão do modelo.
- Registro e Implantação dos Modelos (*Model Registry & Deployment*) Gerenciar e implantar modelos.
 - Pacote: modelos de pacote para implantação.
 - Contentorizar: Utilizar recipientes para manter a consistência.
 - Implantar: Implementar modelos em produção.
- Interface de Programação de Aplicações de Inferência (*Inference Application Programming Interface - API*): Fornece previsões em tempo real.
- Armazenagem de Recursos (*Feature Store*): Gerencie recursos de dados reutilizáveis.

REFERÊNCIA:

- [1] BHARDWAJ D. Machine Learning Operations. Machine LearningCommunity (Linkedin Group), 2024. Disponível em: https://media.linkedin.com/dms/image/D5622AQGqosDTPTYyKg/feedshare-shrink_800/0/172182723474?e=1724889600&v=beta&t=aFxZV1Ee1obfpNd5xXyoLfq2t0wlYJvy11f8no_82po. Acessado em Julho 24, 2024.

38. ADAPTADORES LORA e RAG

Editado por: Dr. Arnaldo de Carvalho Junior
Data: Julho 29, 2024.

1. Introdução

O mundo dos grandes modelos de linguagem (*large language models* - LLMs) está evoluindo rapidamente, com novos avanços surgindo a cada dia. Embora a grande escala destes modelos tenha trazido capacidades notáveis em tarefas como geração de texto, tradução e sumarização, o seu imenso tamanho também coloca desafios significativos. Treinar e implantar esses modelos colossais pode ser computacionalmente caro, consumir muita memória e consumir muito tempo. É aqui que o conceito de ajuste fino eficiente (*efficient fine-tuning*) entra em jogo, oferecendo uma solução para o problema de ambientes com recursos limitados [1].

Dois métodos populares para ajuste fino de LLM eficiente são Adaptação de Baixa Classificação (*Low-Rank – LoRA Adaptation*) e Recuperação Geração Aumentada (*Retrieval Augmented Generation - RAG*). Essas técnicas, embora compartilhem o objetivo de otimizar o desempenho do modelo com carga computacional mínima, adotam estratégias distintas e atendem a diferentes casos de uso [1].

2. O que são adaptadores LoRA?

Os adaptadores da LORA são, uma das estratégias mais inteligentes usadas no aprendizado de máquina nos últimos anos. Os adaptadores da LORA vieram como uma estratégia muito natural para modelos de ajuste fino. A ideia é perceber que qualquer matriz de parâmetros do modelo em uma rede neural de um modelo treinado é apenas uma soma dos valores iniciais e as seguintes atualizações de descida de gradiente aprendidas nos mini-lotes (*mini-batch*) de dados de treinamento:

$$\theta \text{ (modelo treinado)} = \theta \text{ (valor inicial)} + \text{Atualizações de descida de gradiente}$$

A partir daí, pode-se entender um modelo de ajuste fino como um conjunto de parâmetros

do modelo, onde se continua a agregar ainda mais os gradientes em algum conjunto de dados especializado:

$$\theta \text{ (ajuste fino)} = \theta \text{ (modelo treinado)} + \text{Atualizações de descida mais gradiente}$$

Quando se percebe que é possível decompor a aprendizagem pré-expedida e o aprendizado de ajuste fino nesses 2 termos:

$$\theta \text{ (ajuste fino)} = \theta \text{ (modelo treinado)} + \Delta W$$

Então entende-se que não precisa que essa decomposição aconteça na mesma matriz; Pode-se somar a saída de 2 matrizes diferentes. Essa é a ideia por trás da Lora: aloca-se novos parâmetros de peso que se especializam no aprendizado dos dados de ajuste fino e congelam-se os pesos originais. Como tal, não é muito interessante porque novas matrizes de parâmetros do modelo dobrariam apenas a memória necessária alocada ao modelo. Portanto, o truque é usar uma aproximação da matriz de baixo *ranking* para reduzir o número de operações e a memória necessária. Introduz-se 2 novas matrizes A e B para aproximar ΔW :

$$\Delta W \sim \text{ba}$$

É importante perceber que, normalmente, a quantidade de dados de treinamento usados para ajuste fino é muito menor que os dados usados para pré-treinamento. Como consequência, é improvável que se possa até ter dados suficientes para obter uma boa convergência estatística na matriz completa ΔW . A aproximação de baixo *ranking* atua como uma técnica de regularização que ajudará o modelo a generalizar melhor em dados invisíveis. Ajuste fino significa que se agregam os gradientes em dados especializados, conforme Figura 1.

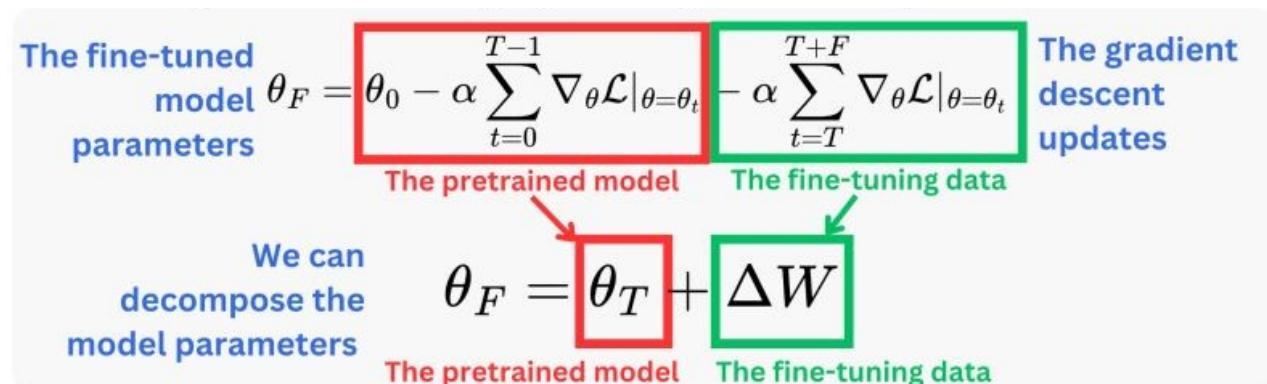


Figura 1 – Ajuste Fino do Adaptador LoRA
Adaptado de [1]

Pode-se imitar a decomposição adicionando novos parâmetros do modelo, conforme Figura 2.

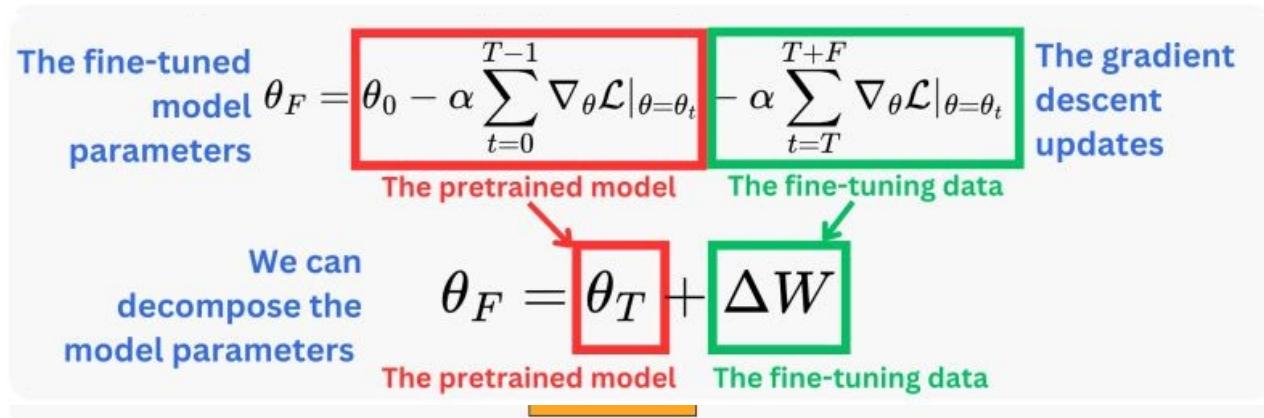


Figura 2 – Novos parâmetros do modelo do Adaptador LoRA
Adaptado de [1]

Substituem-se esses pesos por uma aproximação de baixa classificação (*ranking*) para latência e memória, conforme Figura 3.

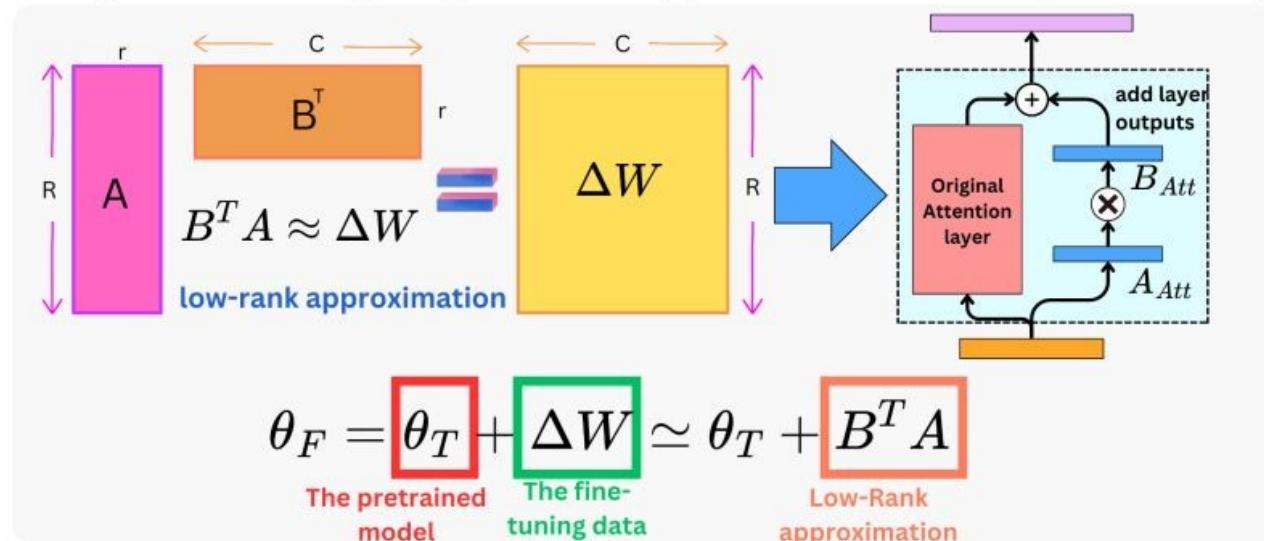


Figura 3 – Aproximação de baixa classificação do modelo do Adaptador LoRA
Adaptado de [2]

3. RAG

A geração de recuperação aumentada (retrieval-augmented generation – RAG) é uma abordagem de mudança de jogo que transforma e otimiza como se interage com LLMs.

a) **O que é o RAG?** Imagine acessar vastas quantidades de dados em tempo real, aprimorando-os com as ideias mais recentes e gerando respostas que não são apenas precisas, mas contextualmente ricas. Isso é o RAG em ação.

b) Como o RAG funciona?

A Figura 4 apresenta um fluxo do funcionamento do RAG.

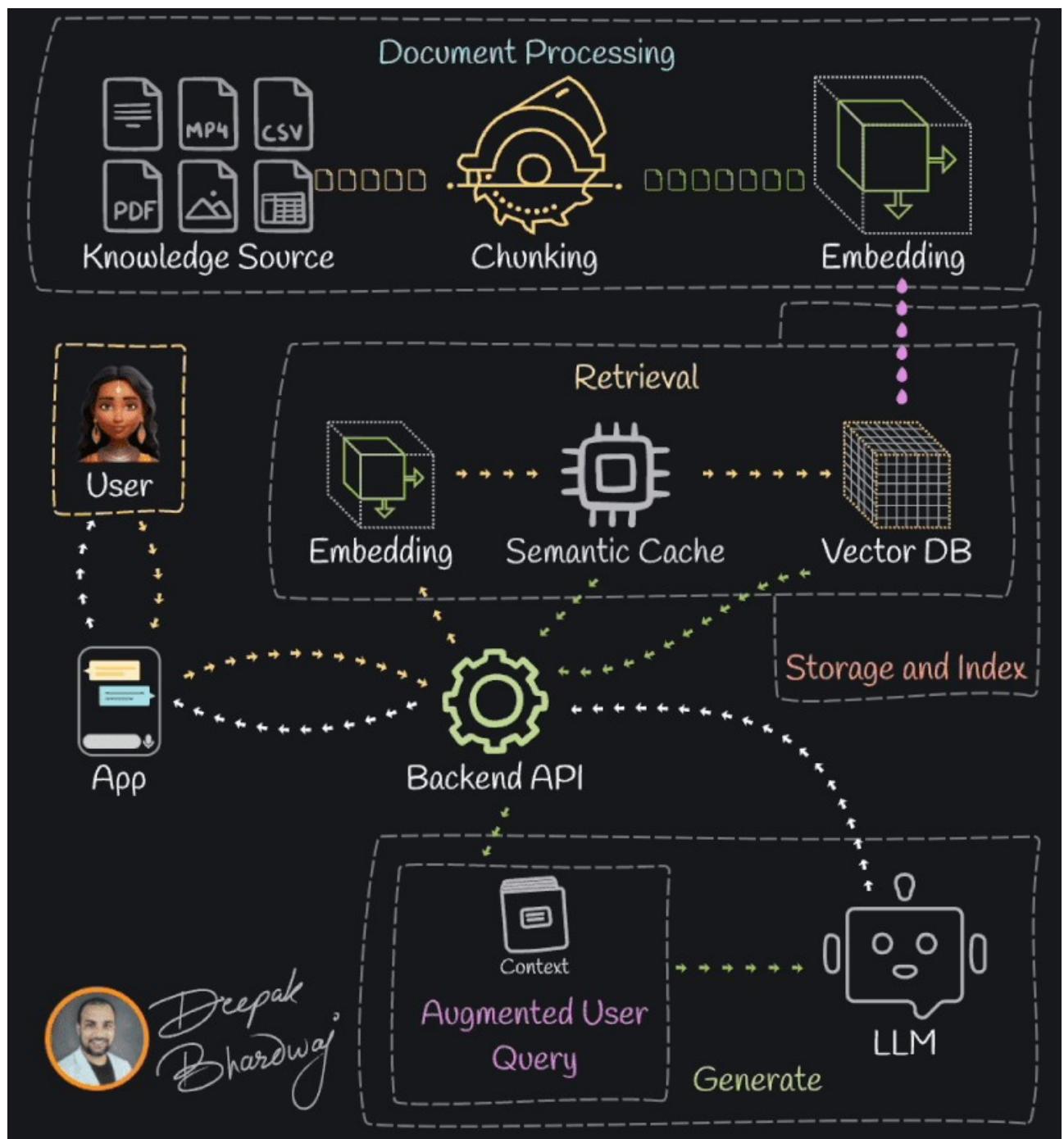


Figura 4 – Funcionamento do RAG.

Adaptado de [3]

- Processamento de Documentos: Inicia-se quebrando conjuntos de dados enormes, sejam PDFs, vídeos ou arquivos CSV, em partes (*chunks*) gerenciáveis.
- Incorporação (*Embedding*): Essas partes são transformados em incorporações - vetores capturando a essência dos dados.

- Recuperação (*Retrieval*): Quando uma consulta de usuário chega, o aplicativo RAG converte a consulta do usuário em incorporação e busca dados relevantes de um banco de dados vetorial com base em semelhanças, garantindo que as informações sejam específicas do contexto.
- Consulta Aumentada (*Augmented Query*): A consulta do usuário é enriquecida com os dados recuperados, adicionando profundidade ao contexto.
- Geração (*Generation*): Finalmente, um LLM gera uma resposta que não é apenas relevante, mas aprimorada por dados precisos de suas fontes de conhecimento.

c) Porque o RAG é importante? O RAG preenche a lacuna entre dados estáticos e necessidades dinâmicas do usuário, tornando-o uma ferramenta poderosa para empresas e desenvolvedores que visam soluções mais inovadoras, mais rápidas e mais precisas.

3.1. RAG: Injetando Conhecimento por meio da Recuperação

O RAG é uma mudança de paradigma no ajuste fino de LLMs, com foco na alavancagem de conhecimento externo para melhorar seu desempenho [1]. Em vez de ajustar diretamente os pesos do modelo, a RAG emprega um processo de duas etapas:

- a) **Recuperação:** Quando apresentado a uma consulta, o RAG primeiro recupera informações relevantes de uma vasta base de conhecimento. Esta base de conhecimento pode ser uma coleção de documentos, artigos ou quaisquer outras fontes de texto relevantes.
- b) **Geração:** A informação recuperada é então alimentada no LLM, que gera uma resposta baseada tanto na consulta quanto no contexto recuperado.

4. Ajuste fino de modelo completo vs. LoRA vs. RAG

Todas as três técnicas são usadas para aumentar o conhecimento de um modelo existente com dados adicionais.

4.1. Ajuste Fino do Modelo Completo: Isso envolve ajustar todos os pesos de um modelo pré-treinado em dados específicos da tarefa. Embora isso funcione muito bem, não é praticamente viável em grandes modelos – (*large language model* - LLM), por exemplo, principalmente por causa de:

- a) Seu tamanho.
- b) O custo envolvido no ajuste fino de todos os pesos.
- c) O custo envolvido na manutenção de todos os grandes modelos de ajuste fino.

4.2. Ajuste Fino do LoRA: O ajuste fino de LoRA aborda as limitações do ajuste fino tradicional. A ideia é decompor as matrizes de peso (algumas ou todas) do modelo original em matrizes de baixo *rank*. Em seguida, treina-se apenas a rede LoRA e congela-se o modelo grande. No gráfico da Figura 2:

- A rede superior representa o modelo com camadas LoRA.
- A rede inferior representa o grande modelo pré-treinado

Observa-se a diferença no número de conexões que ambas as redes têm. Isso reduz imensamente os requisitos computacionais.

4.3. RAG: Tanto o modelo completo quanto o ajuste fino de LoRA discutidos acima envolvem treinamento adicional. O RAG ajuda a aumentar as informações adicionais, sem ajustar o modelo. Existem 7 etapas, que também estão marcadas no visual da Figura 5:

- **Passos 1-2:** pegam-se os dados adicionais e os despeja em um banco de dados vetorial após a incorporação. (Isso é feito apenas uma vez).
- **Passo 3:** Usa-se o mesmo modelo de incorporação para incorporar a consulta do usuário.
- **Passos 4 - 5:** encontram-se os vizinhos mais próximos no banco de dados vetorial na consulta incorporada.
- **Passos 6 – 7:** forneça a consulta original e os documentos recuperados (para mais contexto) ao LLM para obter uma resposta.

Claro, também existem muitos problemas com o RAG, como:

- rr) Envolve a correspondência de similaridade entre a consulta e os vetores despejados. No entanto, as perguntas são estruturalmente muito diferentes das respostas. Assim, normalmente recuperam-se muitos documentos irrelevantes.
- ss) Os sistemas RAG típicos são bem adequados apenas para sistemas de resposta a perguntas baseados em pesquisas. Por exemplo, não se pode criar um *pipeline* RAG para resumir os dados adicionais. O LLM nunca recebe informações sobre todos os documentos em seu *prompt*. Isso ocorre porque a etapa de correspondência de similaridade recupera apenas as melhores correspondências.

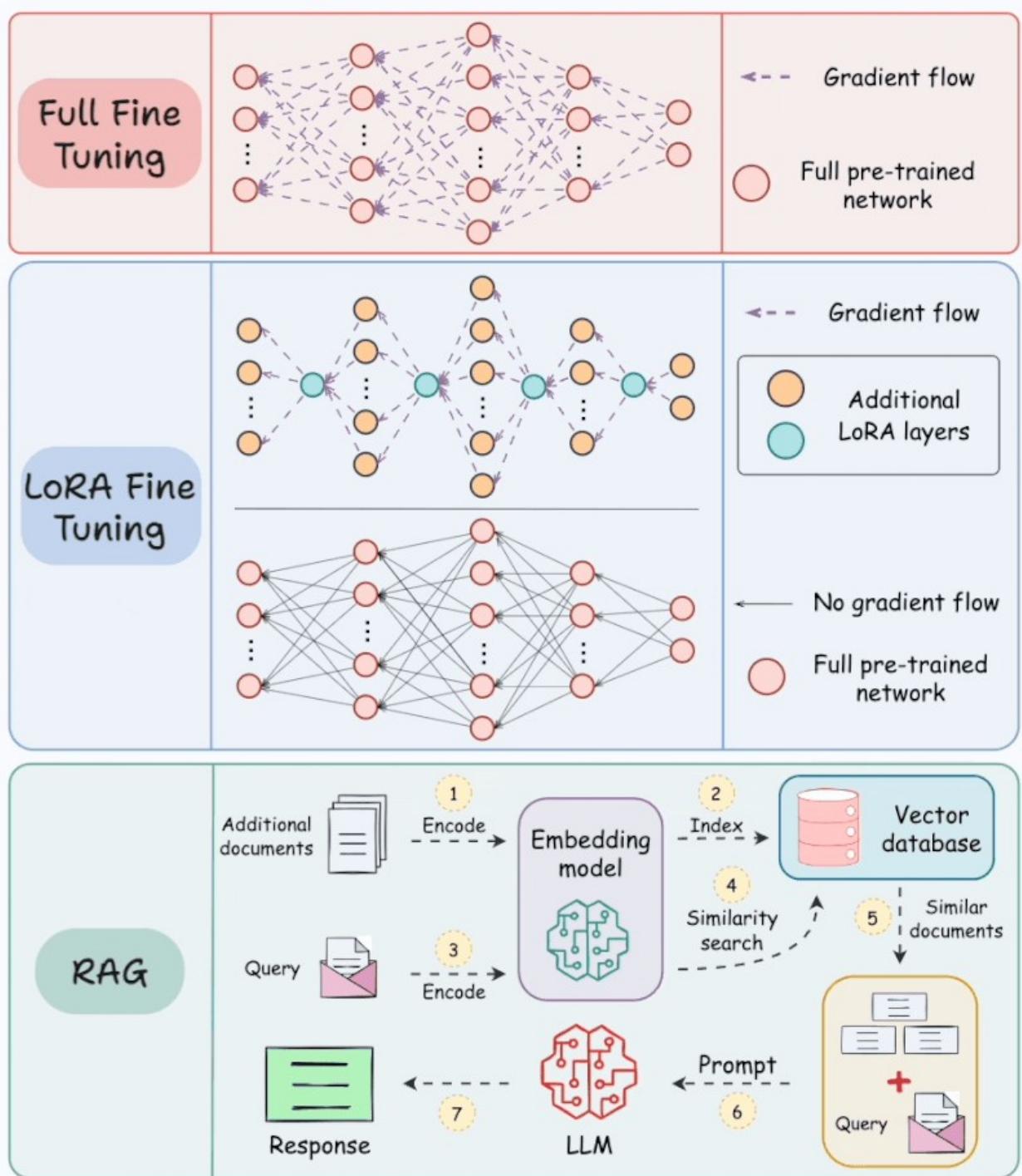


Figura 5 – Comparação de Ajuste Fino entre LoRA e RAG
Adaptado de [4].

REFERÊNCIAS:

- [1] VARMAM, A. J. S. R. R. RAG vs. LoRA for Efficient Language Model Tuning, Medium, Jul 2024. Disponível em: <https://medium.com/@rajaravivarman/rag-vs-lora-for-efficient-language-model-tuning-c6f675e710c5>. Acessado em Jul 29, 2024.
- [2] BENVENISTE D. What Are LoRA Adapters. The AiEdge.io, Machine Learning Community (Linkedin Group), 2024. Disponível em: <https://media.linkedin.com/dms/image/D4D22AQGTjAyfic-ykQ/feedshare->

shrink_800/0/1721831210243?

e=1724889600&v=beta&t=BxPWKYsJv28JBCTL5JR6z78qysG8apHnCetjcHGn9wl.

Acessado em Julho 29, 2024.

[3] BHARDWAJ, D. Retrieval Augmented Generation - RAG. Data.AI.Architecture (Linkedin Group). Disponível em: https://media.licdn.com/dms/image/v2/D5622AQG2jpncLgElrw/feedshare-shrink_800/feedshare-shrink_800/0/1724244153835?e=1727913600&v=beta&t=DTx5HLJA-sRzY_i46xG1yD3Sq8ZnYW7XxBVEciF476g. Acessado em Ago 27, 2024.

[4] CHAWLA, A. Full-Model Fine-Tuning vs. LoRA vs. RAG Explained Visually. Blog.DailyDoseofDS.com, 2024. Disponível em: https://media.licdn.com/dms/image/D5622AQHLTQW72BbNzQ/feedshare-shrink_800/0/1721648001821?e=1724889600&v=beta&t=1_brrq1UCLxje-gpP2nfjk4P-XfMTFG1LydEnWtmf50. Acessado em Jul 29, 2024.

39. ROTEIRO DO CIENTISTA DE LLM

Editado por: Dr. Arnaldo de Carvalho Junior

Data: 19 Ago 2024

1. PROPOSTA DE ROTEIRO DE LLM

Definitivamente cada pesquisador de inteligência artificial (IA) e aprendizado de máquina (*machine learning* - ML) tem sua própria jornada de desenvolvimento. Este artigo apresenta uma proposta de roteiro para o cientista de modelo longo de linguagem (*large language model* – LLM).

1.1. MARCOS IMPORTANTES:

- 1. A arquitetura LLM:** O pesquisador deve entender os componentes fundamentais do LLMS, incluindo tokenização, mecanismos de atenção e geração de texto.
- 2. Construindo um conjunto de dados de instruções (*building the instruction dataset*):** Aprender a criar e refinar os conjuntos de dados para instruir o LLMS de maneira eficaz.
- 3. Modelos de pré-treinamento:** Deve-se se aprofundar no processo de modelos de pré-treinamento, com foco em dutos de dados, escala e computação de alto desempenho.
- 4. Ajuste fino supervisionado (*supervised fine tuning*):** Técnicas mestras de ajuste fino para melhorar o desempenho do modelo em tarefas específicas.
- 5. Aprendizagem de reforço com o retorno humano (*Reinforcement Learning from Human Feedback* – RLHF):** Deve-se explorar os métodos para otimizar modelos usando *feedback* humano.
- 6. Avaliação:** Deve-se avaliar o desempenho do modelo com uma variedade de métricas e *benchmarks* para garantir resultados de alta qualidade.
- 7. Quantização:** Descobrir as técnicas para tornar os modelos mais eficientes e mais rápidos através da quantização.
- 8. Otimização da interface:** Otimizar a interação com os LLMs usando técnicas avançadas como atenção *flash* e decodificação especulativa.

A Figura 1 apresenta o roteiro de um cientista de LLM.

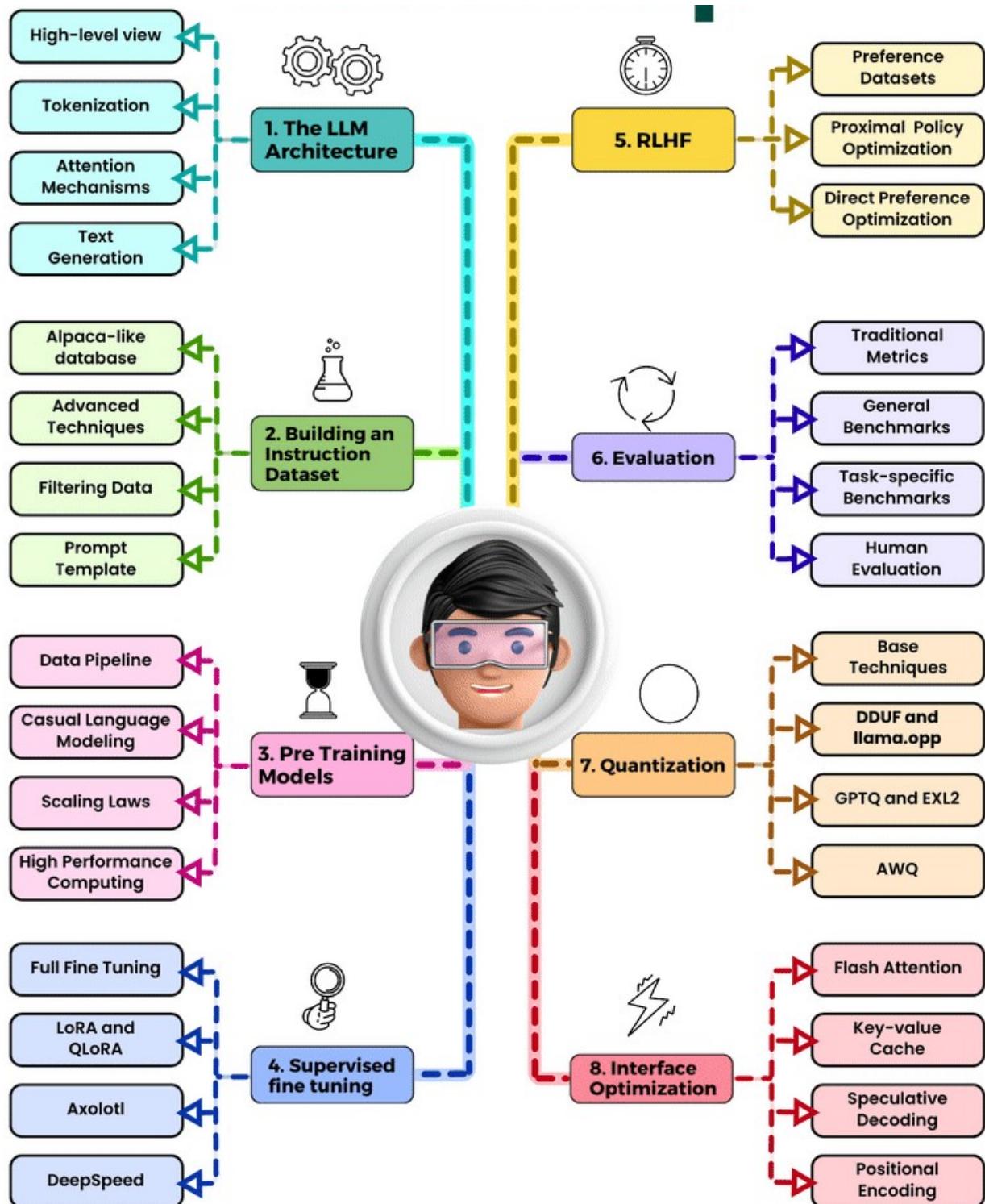


Figura 1 – Proposta de Roteiro de LLM.

Fonte: Adaptado de [1].

2. PREVISÃO CLÍNICA BASEADA EM MODELOS DE LINGUAGEM

Como funciona a previsão clínica baseada em modelos de linguagem? A Figura 2 apresenta uma proposta de sequência para previsão clínica baseada em LLM.

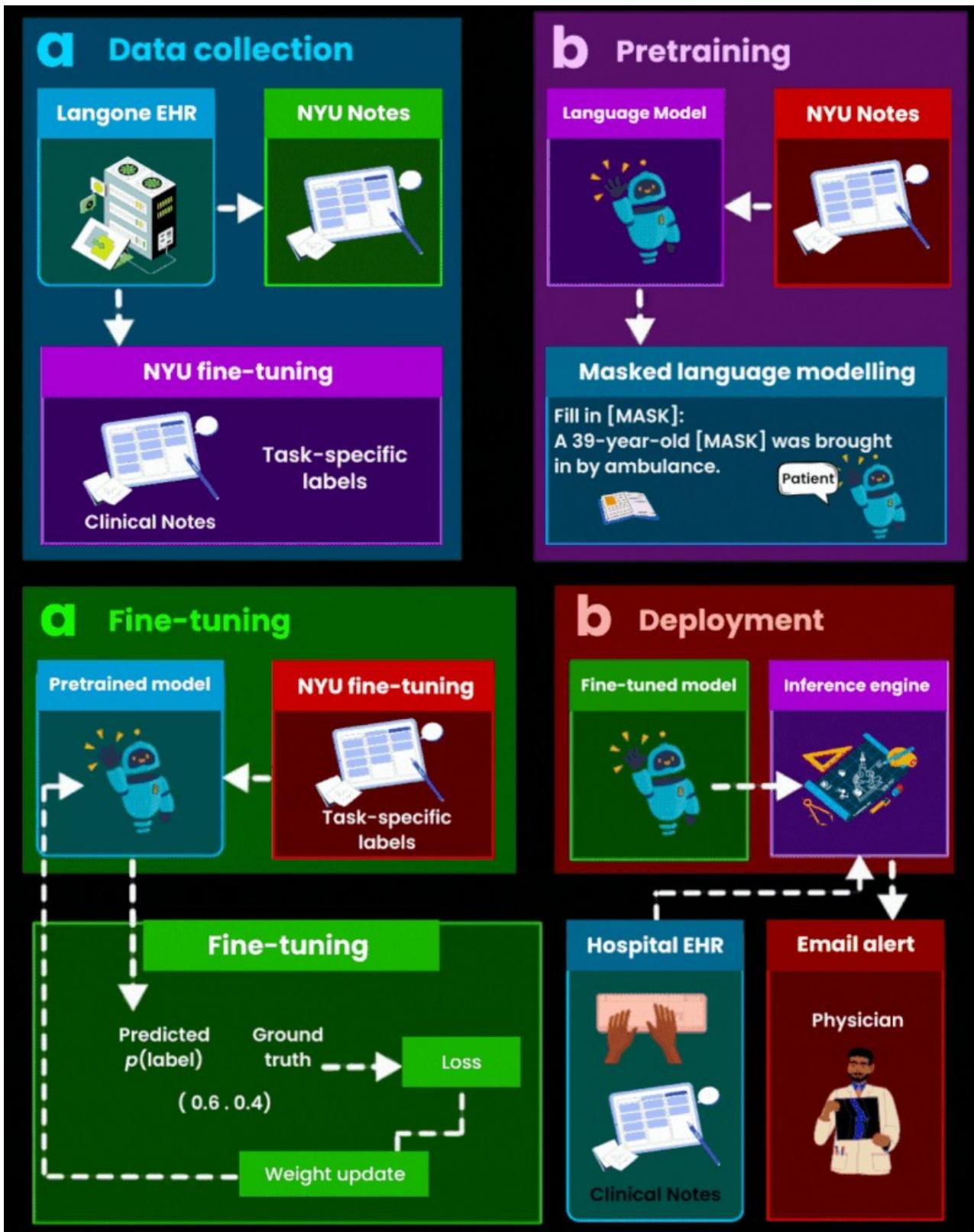


Figura 2 – Roteiro Previsão Clínica Baseada em LLM.

Fonte: Adaptado de [2].

- **Coleta de dados:** os dados clínicos são coletados de registros eletrônicos de saúde (EHR) e notas médicas, formando a base para o treinamento de modelos.
- **Preencher:** o modelo de idioma passa por pré-treinamento usando técnicas de modelagem de idiomas mascaradas para aprender com os dados de texto clínico.
- **Ajuste fino:** o modelo pré-treinado é ajustado com rótulos específicos de tarefas, otimizando-o para previsões clínicas ajustando os pesos com base nos resultados previstos.
- **Implantação:** o modelo de ajuste fino é implantado nos sistemas de saúde, integrados aos mecanismos de inferência para fornecer previsões e alertas em tempo real para os médicos com base nos dados de EHR.

O pesquisador deve seguir as etapas estruturadas apresentadas neste artigo como um guia para aprofundar sua compreensão e experiência.

REFERÊNCIA

- [1] JAIN, R. The LLM Scientist Roadmap, @RAVITJAIN, 2024. Disponível em: https://media.liecdn.com/dms/image/D4D22AQHog2V8PTmeag/feedshare-shrink_2048_1536/0/1722169641966?e=1725494400&v=beta&t=b4YTexXpkqmUMbwG9MR_brGyH9kdyVntQryvNrzJel. Acessado em Ago 01, 2024.
- [2] JAIN, R. Language Model-Based Clinical Prediction, @RAVITJAIN, 2024. Disponível em: https://media.liecdn.com/dms/image/v2/D4D22AQH6nMifir5ETA/feedshare-shrink_800/feedshare-shrink_800/0/1724069687382?e=1726704000&v=beta&t=x6k6JpQ6fhXY_aSQQVMe5hyOTPmipkrjpPBA0uEpSBw. Acessado em Ago 19, 2024.

40. PASSO A PASSO DA APLICAÇÃO GENAI

Editado por: Dr. Arnaldo de Carvalho Junior

Data: Setembro 12, 2024

Um guia completo para auxiliar na compreensão da inteligência artificial gerativa (generative – GenAI) e como construir aplicações GenAI. A Figura 1 apresenta o percurso para a construção de GenAI.

1. O QUE É GenAI?

GenAI refere-se a modelos de IA que criam novas amostras de dados através de padrões de aprendizado de um determinado conjunto de dados. Inclui aplicativos como geração de texto, síntese de imagens, composição musical e muito mais.

2. MODELOS DE FUNDAÇÃO E LLM

- a) Modelos de Fundação (Foundation Models) são modelos de aprendizado de máquina em larga escala treinados em vastos conjuntos de dados que servem como base para a construção de modelos mais especializados.
- b) Modelos de Linguagem Grande (Large Language Models – LLM) são um tipo de modelo de fundação projetado especificamente para entender e gerar linguagem humana.

3. MECANISMO DE ATENÇÃO

O Mecanismo de Atenção (*Attention Mechanism*) é um componente essencial nos modelos modernos de IA, principalmente em transformadores. Ele permite que os modelos se concentrem em partes específicas dos dados de entrada, melhorando a qualidade da compreensão e da geração [2].

4. GenAI vs MODELOS TRADICIONAIS

- a) Modelos Tradicionais (Traditional Models): são modelos pré-treinados em dados específicos de domínio, generalização limitada.
- b) GenAI: utiliza conjuntos de dados grandes, altamente adaptáveis e podem ser ajustados para tarefas específicas.

5. TREINAMENTO DE UM MODELO DE FUNDAÇÃO

O treinamento envolve o uso de grandes conjuntos de dados com conteúdo diversificado para garantir que o modelo possa generalizar bem em várias tarefas. As principais técnicas incluem aprendizado auto-supervisionado, aprendizado de transferência e ajuste contínuo.

6. PILHA DE DESENVOLVIMENTO DE GenAI

- a) Linguagem: Python
- b) Bibliotecas: Tensorflow, PyTorch, Hugging Face Transformers
- c) Ferramentas: Jupyter, Colab, Github
- d) Recursos Computacionais: GPUs, TPUs.

7. APLICAÇÕES GenAI

- a) Geração de texto (chatbots, criação de conteúdo);
- b) Síntese de imagem (arte, fotografia);
- c) Geração de música e áudio
- d) Síntese de vídeo (DeepFakes, Animação);
- e) Geração de código (codificação automatizada).

8. CONSTRUINDO UMA APLICAÇÃO SIMPLES DE GenAI

Inicia-se selecionando um modelo de Hugging Face ou OpenAI, ajustá-lo em um conjunto de dados específico e implantá-lo usando APIs.

9. ATIVIDADE DO ENGENHEIRO DE IA

Um engenheiro de IA desenvolve e otimiza os modelos de IA, gerencia *pipelines* de dados e integra modelos em aplicativos.

A Figura 1 apresenta o percurso completo para a construção de GenAI.

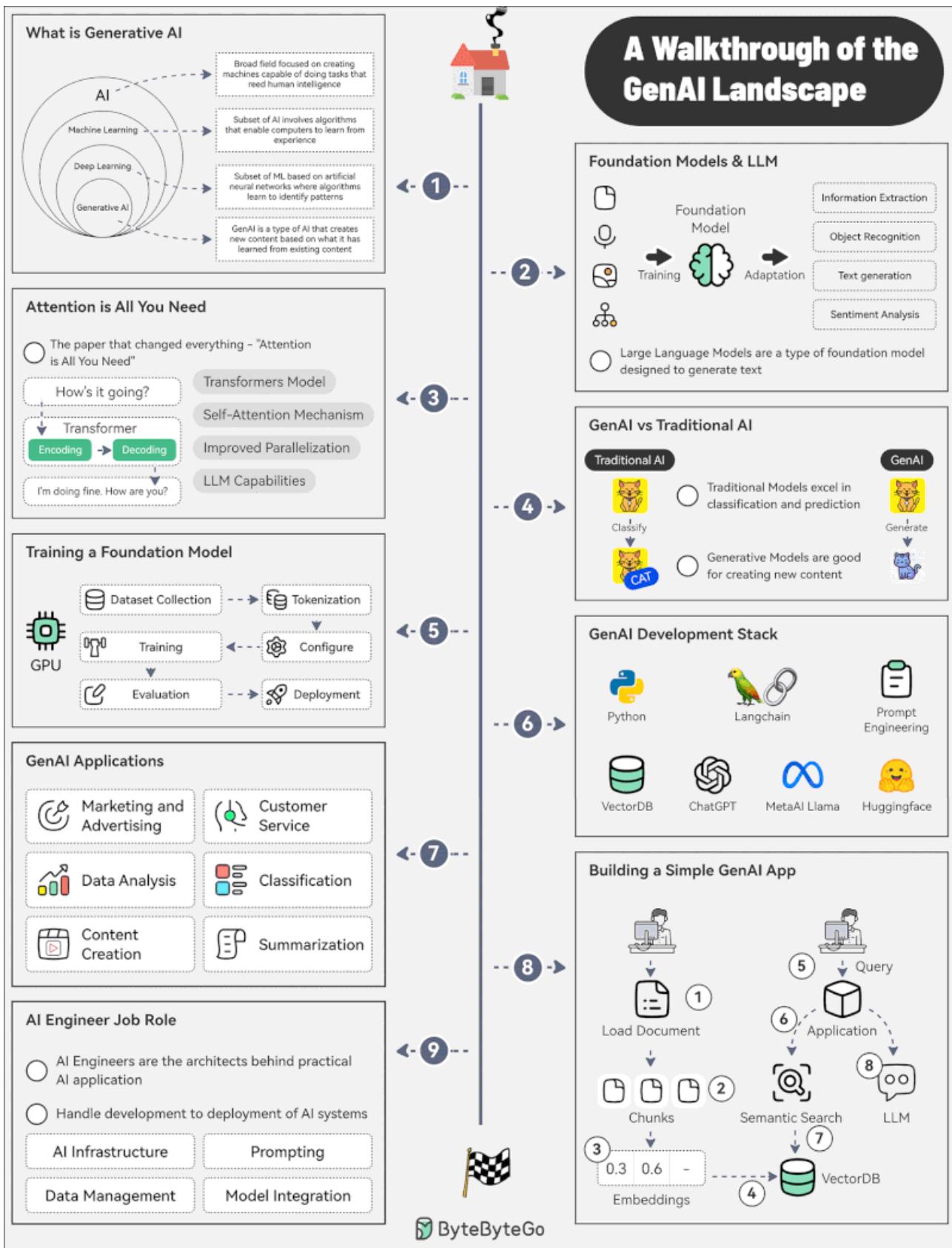


Figura 1 – Percurso para Construção de GenAI
Fonte: Adaptado de [1].

REFERÊNCIAS

- [1] ALAM, A. A Walkthrough of the GenAI Application, Linkedin Post, 2024. Disponível em: https://media.licdn.com/dms/image/v2/D5622AQHbGuB4kRk_KA/feedshare-shrink_2048_1536/feedshare-shrink_2048_1536/0/1726136745397?e=1729123200&v=beta&t=fJ6fUakppZXfQ1_nOBMd5kpBhO-xVVwrjxVL4isi6eM. Acessado em Setembro 12, 2024.
- [2] VASWANI, A. et al. Attention is all you need. Advances in Neural Information Processing Systems, 2017. Disponível em: <https://user.phil.hhu.de/~cwurm/wp-content/uploads/2020/01/7181-attention-is-all-you-need.pdf>. Acessado em Setembro 12, 2024.

41. APRENDIZADO PROFUNDO: EXEMPLOS DE CASOS

Editado por: Dr. Arnaldo de Carvalho Junior, Set 18, 2024.

O aprendizado profundo se tornou a espinha dorsal da IA moderna, com várias arquiteturas liderando a acusação em diferentes domínios. A Figura 1 apresenta quatro categorias principais e seus aplicativos do mundo real:



Figura 1 – Casos de Usos de Redes de Aprendizado Profundo.

Fonte: Adaptado de [1].

a) Redes neurais recorrentes (*Recurring Neural Networks - RNNs*):

- Caso de uso: Previsão de séries temporais: previsão dos preços das ações, padrões climáticos e demanda de energia analisando dados históricos.
- Caso de uso: Processamento de Linguagem Natural (*Processing Natural Language - PNL*): Reforçando Máquinas de Tradução, Análise de Sentimentos e Reconhecimento da Fala através da compreensão de dados sequenciais.

b) Redes neurais convolucionais (*Convolutional Neural Networks – CNNs*)

- Caso de uso: Reconhecimento e classificação da imagem: Detectar objetos, faces e padrões em imagens, amplamente utilizados na área da saúde (por exemplo, detecção de tumores) e segurança.
- Caso de uso: Análise de vídeo: aprimorando a análise de conteúdo de vídeo para aplicações em vigilância, análise esportiva e entretenimento.

c) Auto-Codificadores (AutoEncoders)

- Caso de uso: Detecção de Anomalia: Identificando padrões incomuns em dados, críticos na detecção de fraudes, segurança de rede e monitoramento industrial.
- Caso de uso: Compressão de dados: reduzindo a dimensionalidade dos dados, mantendo seus recursos essenciais, usados na compressão de imagem e vídeo.

d) Redes Adversárias Generativas (*Generative Adversarial Networks – GANS*)

- Caso de uso: Síntese de imagem: criando imagens, arte e animações realistas, revolucionando as indústrias e o design criativos.
- Caso de uso: Aumento de dados: gerando dados sintéticos para treinar modelos de aprendizado de máquina quando dados reais são escassos, cruciais em pesquisa médica e direção autônoma.

O aprendizado profundo não se trata apenas de modelos, trata-se de resolver problemas complexos com abordagens inovadoras. Compreender essas arquiteturas e suas aplicações pode abrir novas portas para inovação em seu campo.

REFERÊNCIA

- [1] RAJATH, K. Transforming Industries with Deep Learning: Essential Use Cases to Know, Artificial Intelligence, Machine Learning, Data Science & Robotics (Linkedin Group), 2024. Disponível em:<https://media.licdn.com/dms/image/v2/D4E22AQGQARRzgmvv8w/feedshare-shrink_800/feedshare-shrink_800/0/1724524602700?e=1729728000&v=beta&t=MPvExmZn9u6tPZDL1a6nfBzQUgsKnLvGu0MhZMg2y14>. Acessado em Set 18, 2024.

42. CONSTRUINDO UM MODELO DE APRENDIZADO DE MÁQUINAS

Editado por: Dr. Arnaldo de Carvalho Junior

Data: Outubro 22, 2024.

A Figura 1 a seguir apresenta ao fluxograma para a construção de um modelo de aprendizado de máquinas (*machine learning* – ML):

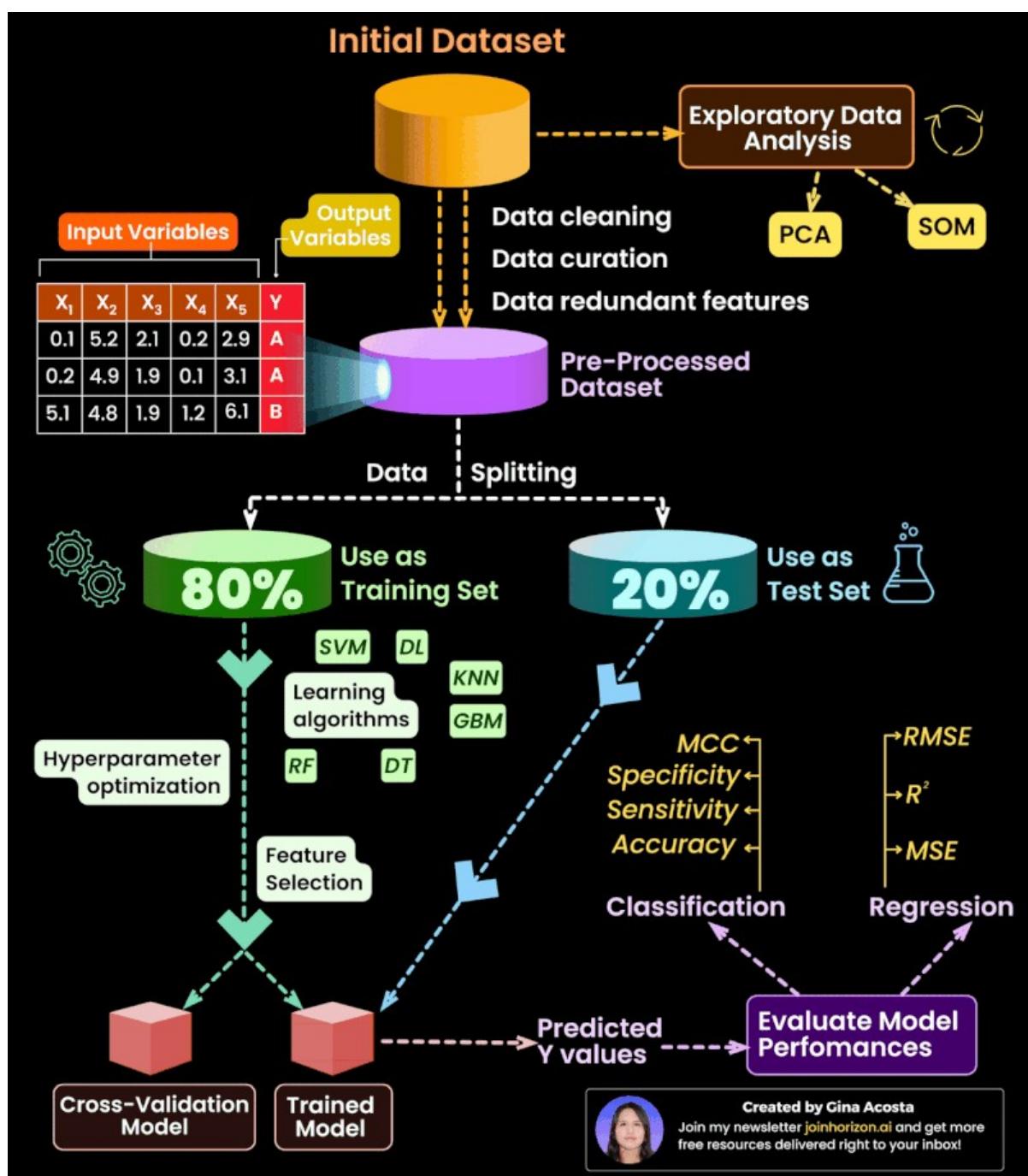


Figura 1 – Fluxograma de Construção de Modelo de ML

Fonte: Adaptado de [1]

1. ETAPAS CHAVE PARA CONSTRUÇÃO DO MODELO DE ML

- a) **Coleta de Dados:** reunir e agregar dados relevantes para o problema que você deseja resolver.
- b) **Limpeza de dados (*data cleaning*):** lidar com valores ausentes, remover duplicatas e corrigir inconsistências.
- c) **Engenharia de Recursos (*feature engineering*):** Transforme dados brutos em recursos significativos que melhoram a precisão do modelo.
- d) **Divisão de Dados (*data splitting*):** Divida o conjunto de dados em conjuntos de treinamento, validação e teste para garantir uma avaliação imparcial.

2. TREINAMENTO DO MODELO

- a) **Selecionar o Algoritmo:** escolha um algoritmo de aprendizado de máquina apropriado com base no tipo de problema (por exemplo, classificação, regressão).
- b) **Treinamento do Modelo:** alimente o algoritmo com os dados de treinamento para encontrar padrões e aprender com eles.
- c) **Ajuste do Hiperparâmetro:** ajuste os parâmetros do modelo para otimizar o desempenho e evitar o excesso de ajuste.

3. AVALIAÇÃO DO MODELO

- a) **Validação Cruzada:** use técnicas como validação cruzada em K-dobras (*k-fold*) para validar o desempenho do modelo.
- b) **Análise de Métricas:** Avalie o modelo usando métricas como precisão, precisão, recall e escore F1.
- c) **refinamento do Modelo:** Com base na avaliação, refina e ajuste o modelo para aumentar a precisão.

4. OTIMIZAÇÃO DO MODELO

- a) **Técnicas de Regularização:** aplique técnicas como a regularização de L1 ou L2 para evitar o excesso de ajuste.

b) Seleção de Algoritmo: considere o uso de técnicas de conjunto como florestas aleatórias ou aumento de gradiente para melhor precisão.

5. APLICAÇÕES DO MUNDO REAL

a) Análise Preditiva (Predictive Analytics): construa modelos para prever tendências e comportamentos do cliente.

b) Processamento de Linguagem Natural (natural language processing - NLP): use o aprendizado de máquina para analisar texto e sentimento.

c) Visão computacional: implemente modelos para identificar objetos e padrões nas imagens.

6. DESAFIOS ENFRENTADOS NA CONSTRUÇÃO DE MODELOS

a) Qualidade dos Dados: a má qualidade dos dados pode levar a previsões imprecisas.

b) Viés (Bias) e *trade-off* de Variação: o viés de equilíbrio e a variação é crucial para evitar o excesso de ajuste ou subajuste.

c) Interpretabilidade do Modelo: modelos complexos como redes neurais podem ser desafiadoras para interpretar.

7. RESUMINDO, COMO A MODELAGEM DE ML FUNCIONA?

a) Preparação de Dados: Verifique se os dados são limpos, relevantes e bem estruturados para o modelo.

b) Treinamento do Algoritmo: o modelo aprende padrões do conjunto de dados de treinamento para fazer previsões.

c) Avaliação e Teste: teste o modelo contra dados invisíveis para validar seu desempenho.

Um pipeline de como criar uma inteligência artificial de classificação de imagens está disponível em [2].

REFERÊNCIAS

- [1] ALAM, A. Building the Machine Learning Model. Linkedin Profile, 2024. Disponível em: https://media.licdn.com/dms/image/v2/D5622AQFjWo-JFNeEzw/feedshare-shrink_2048_1536/feedshare-shrink_2048_1536/0/1729409124612?e=1732752000&v=beta&t=W3pRS156S_LfIRplrQAinkg6qqmHZ_RnF01ONTWSjow. Acessado em Outubro 22, 2024.
- [2] SIMÕES, B. G. Como Criar uma Inteligência Artificial de Classificação de Imagens. EAILab, IFSP, Outubro 31, 2024. Disponível em: <<https://eailab-ifsp.github.io/AI-CLASSIFICATION-PIPELINE/>>. Acessado em Dezembro 11, 2024.

43. ROTEIRO PARA PROJETAR REDES NEURAIS

Editado por: Dr. Arnaldo de Carvalho Junior

Data: Dezembro 16, 2024.

1. INTRODUÇÃO

As redes neurais passam uma entrada, como uma imagem por exemplo, através de múltiplas camadas de neurônios digitais. Cada camada revela recursos (features) adicionais da entrada.

As pesquisas têm demonstrado para quais tipos de tarefas uma determinada rede neural é boa, ou seja, quantos neurônios, quantas camadas e como os neurônios são conectados. A Figura 1 apresenta um exemplo de rede neural para analisar a imagem de um cão.

- Entrada: imagem é fragmentada em pixels
- Camada (Layer - L1): valores dos pixels são detectados.
- Camada L2: bordas e contornos são identificados.
- Camada L3: combinações de contornos identificados.
- Camada L4: features são identificados.
- Camada L5: combinações de features são identificados.
- Saída: a imagem é classificada como “cão”.

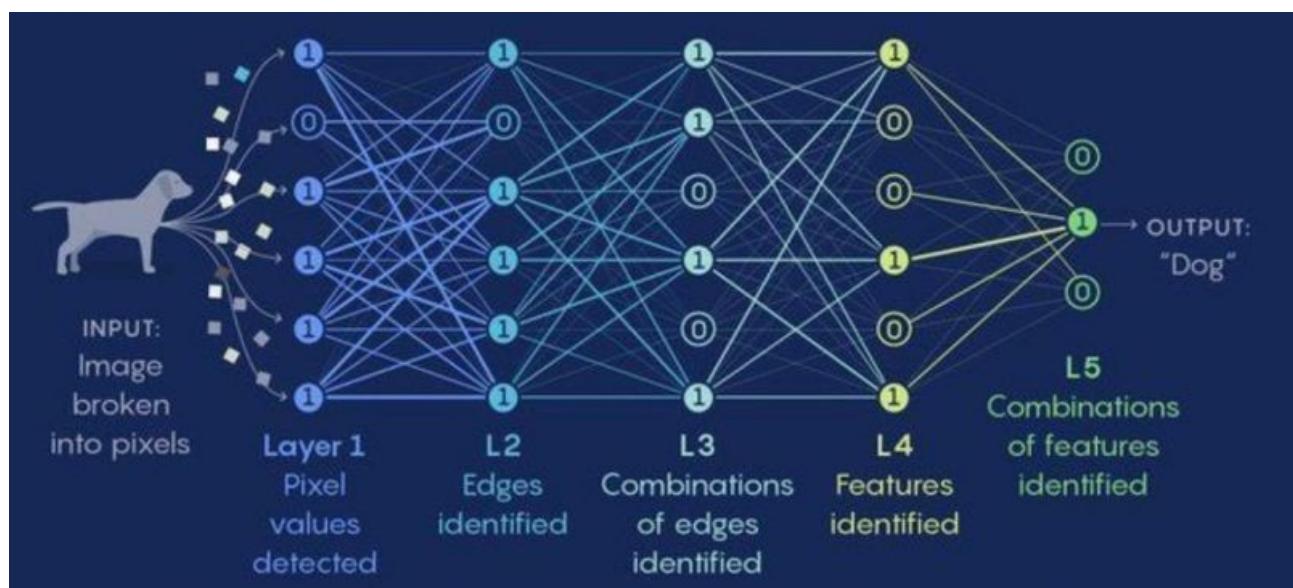


Figura 1 – Exemplo de Rede Neural Para Identificação/Classificação de Imagem
Fonte: Adaptado de [1].

Quando o dado (ou imagem) é alimentado para dentro da rede, cada neurônio artificial que dispara (nominado como 1 na Figura 1) transmite sinais para certos neurônios na próxima camada, que provavelmente dispararão se vários sinais forem recebidos. Esse processo revela informação abstrata sobre a entrada.

Uma rede neural rasa (*shallow network*) tem poucas camadas, porém muitos neurônios por camada. Essas redes “expressivas” são computacionalmente intensivas. Uma rede profunda (*deep network*) possui muitas camadas e relativamente poucos neurônios por camada. Essas redes podem atingir elevados níveis de abstração usando relativamente poucos neurônios. A Figura 2 apresenta exemplos de rede neural rasa e profunda.

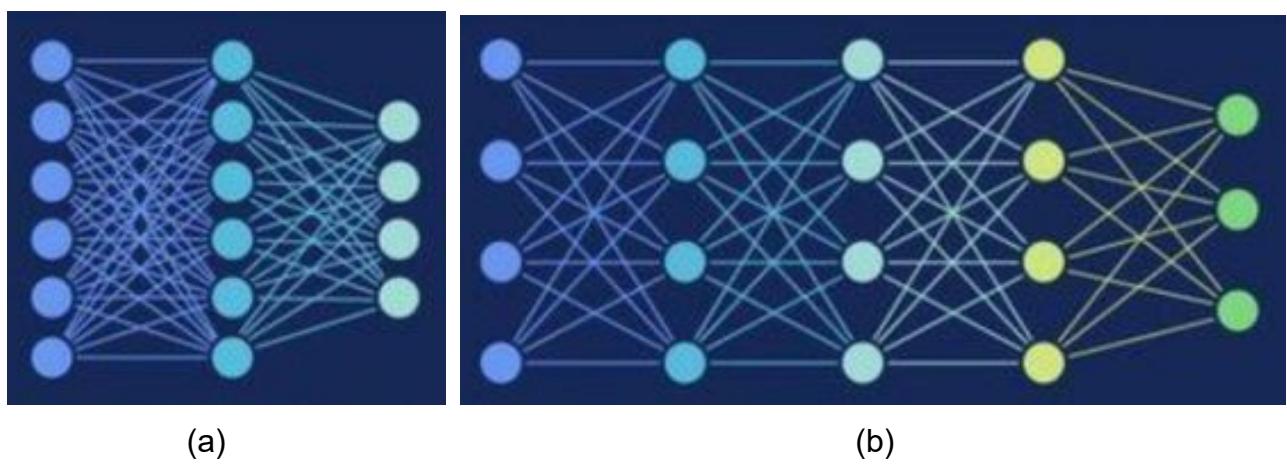


Figura 2 – Exemplo de Rede Neural Rasa (*shallow*) (a), e Profunda (*deep*) (b).
 Fonte: Adaptado de [1].

2. COMO PROJETAR UMA REDE NEURAL

1. Defina O Problema

Descreva claramente o tipo de tarefa:

- **Classificação:** preveja rótulos discretos (por exemplo, gatos vs cães).
 - **Regressão:** preveja valores contínuos (por exemplo, preços das ações).
 - **Clustering:** encontre padrões em dados não supervisionados.

2. Preprocesse os Dados

A qualidade dos dados é crítica para o desempenho do modelo.

- Normalize e padronize recursos (*features*) (MinMaxScaler, StandardScaler).
 - Lide com valores ausentes pontos fora da curva, dados que se diferenciam drasticamente de todos os outros dados (*outliers*).

- Divilde seus dados: treinamento (70%), validação (15%), teste (15%).

3. Projete a Arquitetura da Rede

a) **Camada de Entrada:** O número de neurônios é igual aos recursos de entrada.

b) **Camada Oculta:**

- Comece com algumas camadas e aumente conforme necessário.
- Use funções de ativação [2]:
 - Unidade Retificadora Linear (*Rectifier Linear Unit - ReLU*): Uso Geral. Rápido e eficiente.
 - Unidade Retificadora Linear com vazamento (*Leaky ReLU*): corrige problemas de neurônios moribundos.
 - Tangente Hiperbólica (*Tanh*) / sigmoide: Use com moderação para casos específicos.

c) **Camada de Saída:**

- **Classificação:** use Softmax ou Sigmoide para obter saídas de probabilidade.
- **Rregressão:** ativação linear (sem ativação aplicada).

4. Inicialize os Pesos

A inicialização adequada do peso ajuda na convergência mais rápida:

- Inicialização He: melhor para ativações baseadas em ReLU.
- Inicialização Xavier: Ideal para ativações sigmoides / tanh.

4. Escolha a Função Perda (*Loss Function*)

- **Classificação:** perda de entropia cruzada.
- **Rregressão:** erro ao quadrado médio (*mean square error – MSE*) ou erro absoluto médio (*mean absolute error – MAE*).

5. Selecione o Método de Otimização (*Optimizer*)

Escolha o otimizador certo para minimizar a perda:

- **Adam:** Escolha mais popular para velocidade e estabilidade.
- **Gradiente Estocástico Descendente (*stochastic gradient descent - SGD*):** mais lento, mas confiável para modelos menores.

6. Especifique Épocas (*Epochs*) e Tamanho do Lote de Dados (*Batch Size*)

- **Epochs:** Defina o total de passos sobre o conjunto de treinamento. Comece com 50 a 100 épocas.
- **Batch Size:** pequenos lotes treinam mais rápido, mas são menos estáveis. Lotes maiores estabilizam gradientes.

7. Previna Ajuste Exagerado (*Overfitting*)

- Adicione camadas de abandono a desativar aleatoriamente os neurônios.
- Use a regularização de L2 para penalizar pesos grandes.

8. Ajuste de Hiperparâmetros (*Hyperparameter Tuning*)

Otimize os parâmetros do seu modelo para melhorar o desempenho:

- Ajuste a taxa de aprendizado, a taxa de abandono, o tamanho da camada e as ativações.
- Use a pesquisa da grade ou a pesquisa aleatória por otimização de hiperparâmetro.

9. Avalie e Melhore os Resultados

Monitore as métricas para desempenho:

11. **Classificação:** precisão, acurácia, recall, Score F1, AUC-Roc.
12. **Rregressão:** Pontuação RMSE, MAE, R².

10. Aumento de Dados (*Data Augmentation*)

13. Para tarefas de imagem, aplique transformações como rotação, escala e lançamento para expandir seu conjunto de dados.

A Figura 3 apresenta um pipeline para criação de inteligência artificial (IA) de classificação de imagens [3].



Figura 3 – Como criar uma IA de classificação de imagens.

Fonte: Adaptado de [3].

O modelo de aprendizado de máquina (*machine learning* – ML) utilizando IA pode ser de diferentes tipos, como redes neurais rasas, profundas, recorrentes (*recurrent neural networks* – RNN), convolucionais (*convolutional neural network* – CNN) ou ainda mais complexas como as de memórias longas de termo curto (*long short-term memory* – LSTM) e transformadoras (*Transformer*) [4].

REFERÊNCIAS

- [1] ALAM, A. How to Design a Neural Network. Linkedin Profile, 2024. Disponível em: https://media.linkedin.com/dms/image/v2/D5622AQFgVnimuuK99g/feedshare-shrink_800/B56ZPO6jiMG4Ag-/0/1734343278719?e=1736985600&v=beta&t=ak23Fs8d1a04hAv54b0YZdlQ5IKrcGlp_zdEoLb2mi4. Acessado em Dezembro 16, 2024.
- [2] CARVALHO, A. Função de Ativação, o Núcleo da Composição de Neurônios Artificiais. EAILab, IFSP, 2024. Disponível em: <https://eailab.labmax.org/2024/02/28/funcao-de-ativacao-o-nucleo-da-composicao-de-neuronios-articiais/>. Acessado em Dezembro 16, 2024.
- [3] SIMÕES, B. G., CARVALHO, A. Como Criar uma Inteligência Artificial de Classificação de Imagens. EAILab, Github, 2024. Disponível em: <https://eailab-ifsp.github.io/AI-CLASSIFICATION-PIPELINE/>. Acessado em Dezembro 16, 2024.
- [4] CARVALHO, A. Redes Neurais Artificiais: Algoritmos Poderosos Para Aplicações de IA e ML. EAILab, IFSP, 2024. Disponível em: <https://eailab.labmax.org/2024/04/03/redes-neurais-articiais-algoritmos-poderosos-para-aplicacoes-de-ia-e-ml/>. Acessado em Dezembro 16, 2024.

44. CHATGPT

Editado por: Dr. Arnaldo de Carvalho Junior

Data: Junho 7, 2024.

1. ESCRITA IMEDIATA DO ChatGPT

- a) SONDAR (PROBE): propósito, razão, obstáculos, plano, avaliar;
- b) ESCOPO (SCOPE): sujeito, contexto, objetivo, precisão, expectativa;
- c) FORMA (SHAPE): situação, obstáculo, mira, plano, execução;
- d) TRACAR (TRACE): tópico, relevância, ângulo, clareza, objetivo;
- e) GUIAR (GUIDE): objetivo, entender, informar, direto, examinar;
- f) ALCANCE (REACH): reconhecer, avaliar, montar, conduzir, honrar;
- g) BUSCAR (QUEST): consulta, compreenda, espere, especifique, tempo de tempo;
- h) ALINHAR (ALIGN): avalie, vincule, identifique, gerar, navegar;
- i) ARTE (CRAFT): contexto, requisito, objetivo, estrutura, técnica;
- j) APRENDER (LEARN): olhe, pergunte, avalie, refine, observe;
- k) MODELAR (MODEL): matéria, objetivo, design, execução, aprendizado;
- l) ASPIRAR (ASPIRE): avalie, especifique, planeje, implemente, reflita;
- m) CONDUZIR (DRIVE): direção, raciocínio, insight, verificar, expandir;
- n) IDEALIZAR (IDEAL): identifique, definir, explorar, agir, revisar e refinar;
- o) FORMATAR (SHAPE): situação, obstáculo, mira, plano, execução;
- p) BRAVURA (BRAVE): compare com referências (benchmark), refletir, analisar, visualizar, executar;
- q) INVESTIGAR (DELVE): diagnosticar, explorar, aprender, verificar, expandir.

PROBE:		SCOPE:		SHAPE:	
<ul style="list-style-type: none"> Purpose: Define goal. Reason: Explain why. Obstacles: Anticipate issues. Blueprint: Outline approach. Evaluate: Judge response. 		<ul style="list-style-type: none"> Subject: Choose focus. Context: Provide background. Objective: Aim of interaction. Precision: Detail exactness. Expectation: Anticipate type of reply. 		<ul style="list-style-type: none"> Situation: Set context. Hindrance: Note challenges. Aim: Define objectives. Plan: Desired structure. Execute: Deliver prompt. 	
TRACE:		GUIDE:		REACH:	
<ul style="list-style-type: none"> Topic: Identify subject. Relevance: Confirm importance. Angle: Approach direction. Clarity: Seek clear response. End-goal: Define desired outcome. 		<ul style="list-style-type: none"> Goal: Define endpoint. Understand: Know the field. Inform: Provide necessary info. Direct: Steer conversation. Examine: Review responses. 		<ul style="list-style-type: none"> Recognize: Identify need. Evaluate: Potential responses. Assemble: Formulate prompt. Conduct: Ask question. Honor: Reflect and feedback. 	
QUEST:		ALIGN:		FOCUS:	
<ul style="list-style-type: none"> Query: Formulate question. Understand: Know background. Expect: Set response anticipation. Specify: Detail requirements. Timeframe: Limit for reply. 		<ul style="list-style-type: none"> Assess: Know vs. Need. Link: Connect knowledge. Identify: Spot gaps. Generate: Craft prompt. Navigate: Direct chat. 		<ul style="list-style-type: none"> Find: Select topic. Organize: Order questions. Clarify: Ensure clarity. Understand: Seek comprehension. Synthesize: Integrate into prompt. 	
CRAFT:		LEARN:		STEER:	
<ul style="list-style-type: none"> Context: Set scenario. Requirement: Define need. Aim: State purpose. Framework: Structure prompt. Technique: Approach for response. 		<ul style="list-style-type: none"> Look: Identify topic. Enquire: Ask relevant question. Assess: Evaluate information. Refine: Improve query. Note: Acknowledge feedback. 		<ul style="list-style-type: none"> Scan: Review past interactions. Target: Detail needs. Engage: Engage AI. Evaluate: Check usefulness. Readjust: Modify if needed. 	
MODEL:		ASPIRE:		BRAVE:	
<ul style="list-style-type: none"> Matter: Main topic. Objective: Target aim. Design: Plan query. Execution: Deliver question. Learning: Assess answer. 		<ul style="list-style-type: none"> Assess: Judge the current knowledge. Specify: Clearly define the prompt. Plan: Set out the conversation flow. Implement: Put prompt into action. Reflect: Consider response effectiveness. 		<ul style="list-style-type: none"> Benchmark: Set standards. Reflect: Ponder phrasing. Analyze: Break down prompt. Visualize: Ideal response. Execute: Send prompt. 	
DRIVE:		IDEAL:		DELVE:	
<ul style="list-style-type: none"> Direction: Set query path. Reasoning: Justify topic. Insight: Seek deep understanding. Verify: Confirm accuracy. Expand: Grow conversation. 		<ul style="list-style-type: none"> Identify: Choose a topic. Define: Specify the query. Explore: Consider angles. Act: Write the prompt. Look-back: Review and refine. 		<ul style="list-style-type: none"> Diagnose: Pinpoint issue. Explore: Look at ways to ask. Learn: Gain insights. Verify: Check relevancy. Expand: Build on info. 	

Figura 1 – Estrutura de Incitação (*prompt*) do ChatGPT.

Fonte: Adaptado de [1].



Figura 2 – Passo a passo de como o ChatGPT funciona.

Fonte: Adaptado de [2].

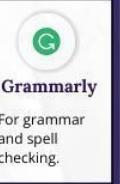
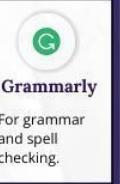
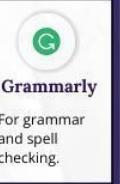
<ul style="list-style-type: none"> Text Generation: Create coherent and contextually relevant text based on a given prompt. Text Completion: Finish incomplete sentences or paragraphs in a logical manner. Vision Capabilities: Can analyse and interpret images, charts, and diagrams with precision, describe visual elements in detail. Audio Capabilities: can process and generate audio data, including speech recognition, text to speech conversion, and audio analysis. Summarization: Condense long articles or documents into concise summaries. Paraphrasing: Rephrase sentences or paragraphs to provide a new perspective while retaining the original meaning. Translation: Translate text between multiple languages with high accuracy. Question Answering: Provide accurate answers to factual questions. Conversation Simulation: Engage in human-like dialogue, maintaining context over multiple turns. 	<h2>Features</h2>  <ul style="list-style-type: none"> Real-time Conversation: Engage in real-time, back-and-forth conversations across multiple modalities. Improved Multilingual Support: Understand and generate content in over 50 languages. Multimodal Generation: Generate outputs that combine text, images, and audio. Contextual Awareness: Provide more relevant and coherent responses based on user intent, background knowledge, and conversational history. Enhanced Safety and Ethical Guardrails: Ensure responsible, unbiased, and factually accurate outputs. 								
Use Cases									
Content Creation <ul style="list-style-type: none"> Blog posts. Articles. Social media updates. Marketing copy. 	Content Repurposing <ul style="list-style-type: none"> Turning articles into social media snippets. Creating video scripts from blog posts. Developing newsletters from web content. 	Research Assistance <ul style="list-style-type: none"> Summarising research papers. Extracting key points from lengthy documents. Generating literature reviews. 	Educational Tools <ul style="list-style-type: none"> Crafting study guides. Creating quiz questions. Simplifying complex topics for better understanding. 	Customer Support <ul style="list-style-type: none"> Automating responses to common inquiries. Generating FAQ content. Crafting user manuals. 					
<h3>Expertise</h3>  <ul style="list-style-type: none"> Creative Writing: Craft engaging and compelling narratives. Technical Writing: Explain complex concepts in an accessible manner. SEO Knowledge: Optimise content for search engines. Language Translation: Translate text accurately between languages. Research Skills: Summarise and synthesise information effectively. Content Strategy: Plan and implement effective content repurposing strategies. 		<h3>Format</h3>  <ul style="list-style-type: none"> Blog Posts: In-depth articles on specific topics. Social Media Posts: Short, engaging updates for platforms like Twitter, Facebook, LinkedIn. Infographics: Visual representations of data and concepts. Newsletters: Concise updates sent to subscribers. E-books: Comprehensive guides or collections of related content. Webinars/Podcasts: Audio or video content providing detailed discussions on topics. Slide Decks: Visual presentations for talks or online courses. 							
<h3>Advanced Prompts</h3> <ul style="list-style-type: none"> Summarisation: "Summarise the main points of this article in 200 words." Paraphrasing: "Rewrite this paragraph with a more conversational tone." Expansion: "Expand this bullet point into a detailed paragraph." Condensation: "Condense this 500-word section into a 100-word summary." Translation: "Translate this English text into Spanish." 		<h3>Alternatives</h3> <table border="1" style="width: 100%; text-align: center;"> <tr> <td> QuillBot For advanced paraphrasing.</td> <td> Hemingway Editor For readability improvements.</td> <td> Canva For creating visual content like infographics.</td> <td> Grammarly For grammar and spell checking.</td> <td> Hootsuite For social media management.</td> </tr> </table>			 QuillBot For advanced paraphrasing.	 Hemingway Editor For readability improvements.	 Canva For creating visual content like infographics.	 Grammarly For grammar and spell checking.	 Hootsuite For social media management.
 QuillBot For advanced paraphrasing.	 Hemingway Editor For readability improvements.	 Canva For creating visual content like infographics.	 Grammarly For grammar and spell checking.	 Hootsuite For social media management.					
<h3>Prompting Technique</h3>  <ul style="list-style-type: none"> Context Provision: Provide clear context for the content to be repurposed. Specific Instructions: Give detailed and precise instructions for the desired outcome. Iterative Refinement: Use follow-up prompts to refine and improve the output. Tone and Style Guidance: Specify the tone and style to match the target audience. 		<h3>Emotion Prompts</h3>  <ul style="list-style-type: none"> Inspiration: "Generate a motivational quote based on this content." Empathy: "Rewrite this paragraph to express empathy towards the reader." Excitement: "Transform this announcement into an exciting social media post." Urgency: "Create a call-to-action that conveys a sense of urgency." Humour: "Add a humorous twist to this section." 							
<h3>Plugins</h3> <table border="1" style="width: 100%; text-align: center;"> <tr> <td> Yoast SEO For optimising content for search engines.</td> <td> WordPress For managing and publishing content.</td> <td> Buffer For scheduling social media posts.</td> <td> Adobe Creative Cloud For advanced design needs.</td> <td> Google Analytics For tracking content performance.</td> </tr> </table>		 Yoast SEO For optimising content for search engines.	 WordPress For managing and publishing content.	 Buffer For scheduling social media posts.	 Adobe Creative Cloud For advanced design needs.	 Google Analytics For tracking content performance.	<h3>Limitations</h3>  <ul style="list-style-type: none"> Provide inaccurate information. Perpetuate biases and stereotypes. Create "hallucinations" (Information that is not based on facts). Lack human knowledge and skills in certain domains. Be vulnerable to exploitation and social engineering. 		
 Yoast SEO For optimising content for search engines.	 WordPress For managing and publishing content.	 Buffer For scheduling social media posts.	 Adobe Creative Cloud For advanced design needs.	 Google Analytics For tracking content performance.					

Figura 3 – Recursos do ChatGPT.

Fonte: Adaptado de [3].

2. VINTE REGRAS DE OURO PARA ESCREVER PROMPTS DE ChatGPT

Aqui está o que está incluído:

1. Seja claro e específico.
2. Use frases completas.
3. Forneça contexto.
4. Evite ambiguidade.
5. Inclua exemplos.
6. Use gramática e ortografia adequadas
7. Especifique a saída desejada.
8. Defina termos técnicos.
9. Seja educado.
10. Use linguagem positiva.
11. Seja conciso.
12. Evite fazer várias perguntas.
13. Teste e refine.
14. Forneça restrições.
15. Seja aberto quando necessário.
16. Indique a forma de resposta.
17. Peças soluções passo a passo.
18. Use o sequenciamento lógico.
19. Incentive a criatividade.
20. Inclua os detalhes necessários.



Figura 4 – 20 Regras de Ouro para Escrever Comandos (prompts) no ChatGPT
Fonte: Adaptado de [4].

3. QUINZE ERROS PARA EVITAR AO USAR O CHATGPT

A Figura 5 apresenta os 15 erros a serem evitados quando utilizar o ChatGPT.



Figura 5 – 15 erros a serem evitados quando utilizar o ChatGPT.

Fonte: Adaptado de [5].

São eles:

1. Contexto não especificando.
2. Fazer muitas perguntas de uma vez.

Ignorar as limitações da IA.

3. Esperar precisão perfeita.
4. Usar *prompts* vagos.
5. Negligenciar o refinamento dos *prompts*.
6. Confiar apenas no chatgpt para decisões críticas.
7. Pular as respostas do modelo.
8. Sobrecarga com textos longos.
9. Não dar retorno (*feedback*) sobre as respostas.
10. Usar linguagem ou conteúdo inadequado.
11. Esquecer de verificar informações sobre informações.
12. Não fazer perguntas de acompanhamento.
13. Com vista para as opções de personalização da IA.
14. Não considerar a Ética.

4. COMO USAR O CHATGPT PARA AUXILIAR NA METODOLOGIA DA PESQUISA

Como pesquisadores podem utilizar a inteligência artificial (IA) para aprimorar a sua metodologia na pesquisa. A Figura 6 apresenta um roteiro para utilizar corretamente o ChatGPT para auxiliar com a metodologia da pesquisa.



Methodology Selection and Justification

1.1 Merits of employing a methodology

Outline the merits of employing a [your methodology e.g., "mixed methods"] when studying [topic], emphasizing how it enriches the research by combining the strengths of both qualitative and quantitative data.

1.2 Choice of approach

Justify the choice of [qualitative/quantitative/mixed methods] approach for [topic], outlining how it aligns with the research objectives and expected outcomes.

1.3 Pros and cons of methodology

Weigh the pros and cons of the selected research methodology, considering its impact on the study's [make sure you've mentioned study details] validity, reliability, and applicability.



Research Design and Procedures

2.1 Collecting data

Suggest best practices for collecting data in [field/research context], ensuring methodological rigor and relevance.

2.2 Instrument refinement

Formulate a strategy for conducting pilot tests or refining the research instruments, which may include questionnaires, interviews, or observation protocols.

2.3 Research design

(useful prompt when reading other relevant paper)

Provide a comprehensive description of the research design, encompassing independent and dependent variables, control measures, and considerations of potential confounding factors.



Ethical Considerations and Data Protection

3.1 Ethical Challenges

Address potential ethical challenges in researching [topic] and suggest practical solutions to uphold ethical standards.

3.2 Impact of cultural sensitivities and differences

Explain how cultural sensitivities or differences may impact the ethical conduct of research in [field/research context] and suggest ways to address them.

3.3 Managing and storing research data securely

Develop a plan for securely managing and storing research data in compliance with relevant data protection regulations and guidelines [mention specific guidelines as applicable].

Figura 6 – Roteiro de uso do ChatGpt para auxiliar com a metodologia da pesquisa.

Fonte: Adaptado de [6].

Recomendações ao pesquisador:

1. Seleção da Metodologia:

- A tormenta de idéias (*brainstorm*) aproxima o pesquisador com ChatGPT ou outros LLMs;
- Refinar os prós/contras para o contexto específico da pesquisa;
- Fazer a escolha final.

2. Projeto da Pesquisa:

- Gerar diversas idéias de coleta de dados;
- Colaborar em estratégias de refinamento de instrumentos;
- Nova perspectiva sobre possíveis fatores de confusão.

3. Considerações Éticas:

- Explorar as sensibilidades culturais que possa ter perdido;
- Brainstorm de soluções criativas para desafios éticos;
- Fortalecer o plano de proteção de dados.

Por fim, os LLMs não são o escritor fantasma do pesquisador!

REFERÊNCIAS

- [1] PANJUTA D. ChatGPT Prompting Frameworks. Disponível em: https://media.licdn.com/dms/image/D4E22AQFQOfdshl47_g/feedshare-shrink_800/0/1719064806957?e=1722470400&v=beta&t=hB-saJDSaMeGN-x6dezmoyk9WqarSax9dDM2wt4Myo4. Acessado em Junho 27, 2024.
- [2] PANJUTA, D. How GhatGPT Works. Disponível em: https://media.licdn.com/dms/image/D5622AQHOheni6AWRIA/feedshare-shrink_800/0/1719216008170?e=1722470400&v=beta&t=odNwTYRpSWDikoGRHGIOo36JAgQ8u1M_NpZEo67TvB0. Acessado em Junho 27, 2024.
- [3] VILLAGE, M. ChatGPT – 40 capabilities. Disponível em: https://media.licdn.com/dms/image/D4E22AQGIX67sdPI_A/feedshare-shrink_800/0/1718612529917?e=1722470400&v=beta&t=akZepHUIIntklcdrDt7fGVqWG7t18_Wj6blYJ67Majpk. Acessado em Junho 27, 2024.
- [4] PANJUTA D. 20 Golden Rules To Write ChatGPT Prompts. Disponível em: https://media.licdn.com/dms/image/D4E22AQGFHAsLmScudg/feedshare-shrink_800/0/1722256069667?e=1725494400&v=beta&t=U3xTEmiEYm2I1E3Ke8g6n1uoAzisJo-JmAlD4qbC7Wc. Acessado em Julho 30, 2024.
- [5] PANJUTA, D. 15 Mistakes to Avoid While Using ChatGPT, 2024. Disponível em: https://media.licdn.com/dms/image/D5622AQH36fgHmA7jMw/feedshare-shrink_1280/0/1721916348937?e=1724889600&v=beta&t=xBzNPTBi7YSAUjV_hlicvbtyMRqNdBUv0li764u2-SM. Acessado em Julho 25, 2024.
- [6] ALIANI, R. How to use ChatGpt to assist with Research Methodology, Razia Aliani (X), 2024. Disponível em:<<https://www.google.com.br/url?>

sa=i&url=https%3A%2F%2Ftwitter.com%2Ffraziaaliani&psig=AOvVaw1FKprniumfo_yOb46KoGa0&ust=1723642949722000&source=images&cd=vfe&opi=89978449&ved=0CBQQjRxqFwoTCPCK28qM8ocDFQAAAAAdAAAAABAK>. Acessado em Ago 13, 2024.

ANEXO I – POSTS DO EAILAB

1. FUNÇÕES DE ATIVAÇÃO

DE CARVALHO JUNIOR, A. Função de Ativação, o Núcleo da Composição de Neurônios Artificiais, EAILab, IFSP, Fevereiro 28, 2024. Disponível em:
<https://eailab.labmax.org/2024/02/28/funcao-de-ativacao-o-nucleo-da-composicao-de-neuronios-articiais/>. Acessado em Agosto 19, 2024.

2. REDES NEURAIS ARTIFICIAIS

DE CARVALHO JUNIOR, A. Redes Neurais Artificiais: Algoritmos poderosos para aplicações de IA e ML, EAILab, IFSP, Abril 03, 2024. Disponível em:
<https://eailab.labmax.org/2024/04/03/redes-neurais-articiais-algoritmos-poderosos-para-aplicacoes-de-ia-e-ml/>. Acessado em Agosto 19, 2024.

3. PAL ET

DE CARVALHO JUNIOR, A. PAL ET: A Solução para os Desafios de AIoT em Edge Computing, EAILab, IFSP, Janeiro 01, 2024. Disponível em:
<https://eailab.labmax.org/2024/01/09/pal-e%cf%84-a-solucao-para-os-desafios-de-aiot-em-edge-computing/>. Acessado em Agosto 19, 2024.

4. CNN

DE CARVALHO JUNIOR, A. O Poder das CNNs em Aplicações de ML Envolvendo Identificação e Classificação de Imagens. EAILab, IFSP, Agosto 13, 2024. Disponível em:
<https://eailab.labmax.org/2024/08/13/o-poder-das-cnns-em-aplicacoes-de-ml-envolvendo-identificacao-e-classificacao-de-imagens/>. Acessado em Agosto 19, 2024.

5. ROTEIRO DE DATASETS PARA DL

DE CARVALHO JUNIOR, A. Roteiro Para Criação de Dataset de Imagens Para modelos de Aprendizagem Profunda. EAILab, IFSP, Setembro 24, 2024. Disponível em:
<https://eailab.labmax.org/2024/09/24/roteiro-para-criacao-de-dataset-de-imagens-para-modelos-de-aprendizagem-profunda/>. Acessado em Dezembro 11, 2024.

6. PIPELINE PARA CLASSIFICAÇÃO DE IMAGENS

SIMÕES, B. G. Como Criar uma Inteligência Artificial de Classificação de Imagens. EAILab, IFSP, Outubro 31, 2024. Disponível em: <https://eailab-ifsp.github.io/AI-CLASSIFICATION-PIPELINE/>. Acessado em Dezembro 11, 2024.