

INFO02
Curso: UFCD 10810
UFCD/Módulo/Temática: UFCD 10810 - Fundamentos do desenvolvimento de modelos analíticos em Python
Ação: 10810_1L
Formador/a: Sandra Liliana Meira de Oliveira
Data:
Nome do Formando/a:

Conceitos Principais em Inteligência Artificial (IA)

- **Inteligência Artificial (IA):** Campo da computação que procura desenvolver sistemas capazes de realizar tarefas que normalmente exigem inteligência humana. Engloba diversas subáreas e técnicas, permitindo que máquinas tomem decisões, reconheçam padrões e **simulem inteligência** de forma autónoma. É um conceito amplo que abrange desde algoritmos simples até sistemas complexos como carros autónomos e assistentes virtuais.
- **Machine Learning (Aprendizagem Automática):** Subconjunto da IA focado em algoritmos que **aprendem a partir de dados**. Em vez de serem explicitamente programados para cada tarefa, os modelos de *machine learning* identificam e refinam padrões internos nos dados para fazer previsões ou tomar decisões. Essa abordagem revolucionou a IA ao permitir melhorias contínuas de desempenho conforme se dispõe de mais dados, tornando viável o reconhecimento de voz, recomendações em serviços de streaming, detecção de fraudes e muito mais.
- **Deep Learning (Aprendizagem Profunda):** Subcategoria do *machine learning* que utiliza **redes neurais artificiais com múltiplas camadas** (daí o termo "profunda") para modelar relações complexas nos dados. O *deep learning* destaca-se por alcançar resultados de ponta em áreas como visão computacional e processamento de linguagem natural, graças à sua capacidade de extrair automaticamente características dos dados brutos (como pixels de imagens ou texto bruto) sem necessidade de intervenção humana para *feature engineering*. Em suma, **deep learning é machine learning em escala maior**, aproveitando grandes volumes de dados e alto poder computacional para atingir alta precisão em tarefas complexas.
- **IA Generativa:** Vertente da IA cujo objetivo é **gerar novos conteúdos** (textos, imagens, áudio, etc.) semelhantes aos criados por humanos. Em vez de apenas analisar ou classificar dados, modelos de IA generativa conseguem criar algo novo a partir do que aprenderam. Exemplos incluem geradores de imagens

artísticas, composições musicais e modelos de linguagem como o ChatGPT. A IA generativa ganhou destaque recentemente por **expandir a criatividade das máquinas**, possibilitando aplicações inovadoras em arte, entretenimento e design, além de auxiliar em tarefas como criação de protótipos e brainstorming. Esses modelos generativos mostram a capacidade da IA não só em resolver problemas, mas também em produzir conteúdo original, embora sob supervisão e orientações humanas.

Exemplos Históricos e Tecnológicos Relevantes

Alguns marcos históricos ilustram o avanço impressionante da IA ao longo do tempo, destacando momentos em que máquinas superaram capacidades humanas em tarefas específicas:

- **Deep Blue (IBM) vs. Garry Kasparov (1997):** Deep Blue, um supercomputador desenvolvido pela IBM, ficou famoso por derrotar o campeão mundial de xadrez Garry Kasparov em 1997. Esse evento marcou a **primeira vez que um campeão mundial de xadrez foi vencido por uma máquina** num jogo competitivo. A vitória do Deep Blue evidenciou o poder do cálculo bruto e de algoritmos especializados: ele avaliava milhões de posições por segundo, escolhendo jogadas otimizadas. Embora baseado em técnicas menos "aprendizes" (muito da força do Deep Blue veio de força bruta de computação e regras de xadrez programadas), esse feito foi um divisor de águas que mostrou que computadores poderiam competir – e vencer – humanos até mesmo em domínios de alta complexidade estratégica.
- **AlphaGo (Google DeepMind) vs. Lee Sedol (2016):** Em outro salto significativo, o programa AlphaGo, da DeepMind (Google), derrotou Lee Sedol, um dos melhores jogadores de Go do mundo, em uma série de partidas em 2016. O jogo de Go tem um **espaço de possibilidades enormemente maior que o xadrez**, o que torna impraticável vencer apenas com força bruta. Diferentemente do Deep Blue, o AlphaGo utilizou *machine learning* avançado e *deep learning* – redes neurais treinadas com milhões de partidas para prever movimentos e avaliar posições, combinado com busca em árvore Monte Carlo. A vitória por 4 jogos a 1 sobre um campeão mundial de Go surpreendeu até especialistas, sinalizando o **poder das abordagens de aprendizado** (em vez de programação manual) e inaugurando uma nova era em que sistemas de IA podem superar humanos em tarefas intelectuais extremamente complexas. Esses exemplos históricos (entre outros, como sistemas vencedores de *Jeopardy!* e avanços em jogos complexos) evidenciam a rápida evolução da IA, do uso de algoritmos determinísticos para métodos de aprendizagem capazes de **descobrir estratégias criativas** além da capacidade humana.

Abordagens de Aprendizagem Máquina (ML)

No *Machine Learning*, existem diferentes abordagens de treino de modelos, cada uma adequada a tipos de problemas e dados específicos. As quatro principais categorias apresentadas no conteúdo são:

- **Aprendizagem Supervisionada:** O modelo é treinado com **dados rotulados**, ou seja, cada exemplo de treino inclui entradas (features) e a saída esperada. O objetivo do algoritmo é aprender a mapear de forma geral as entradas nas saídas corretas. Exemplos incluem **classificação** (e.g., identificar se um e-mail é spam ou não) e **regressão** (e.g., prever o preço de uma casa). Essa abordagem é poderosa quando há muitos dados de exemplo disponíveis, pois o modelo **aprende com exemplos corretos fornecidos pelo "supervisor"**. A qualidade do resultado depende fortemente da quantidade e qualidade das amostras rotuladas.
- **Aprendizagem Não Supervisionada:** Neste tipo de aprendizagem, o modelo recebe **dados sem rótulos** e deve descobrir a estrutura autonomamente. Os algoritmos procuram padrões ocultos ou agrupamentos (**clusters**) nos dados, encontrando similaridades para organizar os exemplos em categorias ou estruturas. Por exemplo, pode-se usar aprendizagem não supervisionada para segmentar clientes em grupos com comportamentos semelhantes, sem saber previamente quais grupos existirão. Técnicas comuns incluem análise de clusters (**como o algoritmo K-means**) e redução de dimensionalidade (**como o algoritmo PCA**). Aqui, **não há uma resposta certa pré-definida**, e o sucesso é medido pela utilidade das estruturas descobertas para entender os dados ou auxiliar em tarefas posteriores.
- **Aprendizagem Semi-Supervisionada:** Combina elementos das duas anteriores, utilizando um conjunto **misto de dados – alguns rotulados e muitos não rotulados**. Essa abordagem explora o fato de que, na prática, obter rótulos para todos os dados pode ser caro ou demorado, mas há abundância de dados brutos. O algoritmo inicialmente aprende com os poucos exemplos rotulados e, de seguida, tenta inferir rótulos ou estruturar os dados não rotulados, refinando o seu modelo. É muito usada em cenários como classificação de imagens, onde apenas algumas imagens são identificadas manualmente e as demais são incluídas para melhorar a aprendizagem. Assim, **aproveita-se volume de dados não rotulados mantendo alguma orientação supervisionada**, atingindo um equilíbrio entre custo de rotulação e desempenho.
- **Aprendizagem por Reforço:** Diferente das anteriores, nesta abordagem o algoritmo aprende mediante **tentativa e erro**, interagindo com um ambiente. Em vez de receber explicitamente respostas corretas, o agente de software realiza

ações e obtém recompensas ou punições de acordo com o resultado da ação no ambiente. O objetivo é aprender uma política de decisões que **maximize as recompensas acumuladas** ao longo do tempo. Esse tipo de aprendizagem é inspirado em como os animais (e humanos) aprendem com feedback das suas ações. Aplicações típicas incluem jogos (um agente a aprender a jogar xadrez ou um vídeo-jogo aprendendo a maximizar a pontuação), controlo de robôs e sistemas de recomendação que se adaptam com o uso. A aprendizagem por reforço demonstrou sucesso notável em cenários complexos – por exemplo, foi central na construção de agentes como o AlphaGo mencionado acima. Embora poderoso, requer cuidadoso desenho do sinal de recompensa e geralmente envolve muitas simulações para atingir um bom desempenho.

Cada abordagem de ML tem o seu **papel e contexto de utilização**. Em projetos reais, muitas vezes são combinadas: por exemplo, pode-se usar aprendizagem não supervisionada para pré-estruturar dados e depois aplicar um modelo supervisionado. Compreender essas diferenças é importante para selecionar a técnica adequada a cada problema.

Machine Learning vs. Deep Learning

Embora relacionados, *machine learning* e *deep learning* não são sinónimos, e o conteúdo destaca suas distinções e relação:

- **Relação hierárquica:** *Deep learning* é uma **subárea dentro do machine learning**. Todo modelo de deep learning (rede neural profunda) é um tipo de modelo de machine learning, mas nem todo modelo de machine learning é deep learning. *Machine learning* abrange uma variedade de algoritmos, dos mais simples (como regressão linear ou árvores de decisão) aos mais complexos (redes neurais profundas).
- **Diferenças em complexidade e dados:** *Machine learning* tradicional frequentemente requer que os programadores realizem **engenharia de atributos (feature engineering)** manualmente – isto é, selecionar ou extrair as características mais relevantes dos dados para alimentar os algoritmos. Já *deep learning* tende a **automatizar a extração de características** por meio das múltiplas camadas da rede neural. Em contrapartida, modelos de deep learning geralmente precisam de **volumes muito maiores de dados** e poder computacional para treinar de forma eficaz, enquanto algoritmos de ML mais simples podem funcionar bem com conjuntos de dados menores e menos recursos.
- **Desempenho e interpretabilidade:** Em problemas de alta complexidade (visão, fala, linguagem), redes profundas costumam superar outras abordagens de ML em desempenho, tornando possível alcançar níveis *state-of-the-art*. Por exemplo,

reconhecimento de imagens teve um salto de qualidade com CNNs (redes neurais convolucionais) de deep learning. Entretanto, essa potência tem um custo: os modelos de deep learning **são geralmente menos interpretáveis**, funcionando como "caixas-pretas" onde entender exatamente como a decisão foi tomada é difícil. Modelos de ML mais simples, por sua vez, podem oferecer interpretações mais claras (uma árvore de decisão pode ser visualizada, uma regressão linear tem coeficientes facilmente entendidos).

Em resumo, *machine learning* é o guarda-chuva de técnicas de aprendizagem automatizada, enquanto *deep learning* refere-se aos métodos mais avançados baseados em redes neurais profundas que alcançaram enorme popularidade por resolver problemas antes intratáveis. A escolha entre um e outro depende dos requisitos do problema: se há muitos dados disponíveis e a tarefa é complexa (como visão ou linguagem), deep learning tende a ser apropriado; se os dados são escassos ou a interpretabilidade é crucial, algoritmos de ML mais simples podem ser preferíveis. Ambos, contudo, são fundamentais no desenvolvimento da IA moderna.

Redes Neurais Artificiais e Modelos de Linguagem de Grande Escala (LLMs)

Um dos alicerces do *deep learning* são as **redes neurais artificiais**, inspiradas livremente no cérebro humano. O conteúdo explica como elas funcionam e discute um tipo específico e moderno de rede neural aplicada à linguagem:

- **Redes Neurais Artificiais:** São modelos compostos por **neurónios artificiais organizados em camadas**. Cada neurónio realiza operações matemáticas simples: ele recebe um conjunto de entradas (valores numéricos que podem ser os dados brutos ou saídas de neurónios da camada anterior), multiplica-os por pesos (que representam a importância de cada entrada), soma esses resultados e então aplica uma função de ativação não-linear. A saída de um neurónio alimenta neurónios da próxima camada, e assim sucessivamente. Numa rede *deep learning*, existem múltiplas camadas "escondidas" entre a entrada e a saída, permitindo à rede aprender **representações hierárquicas dos dados** – por exemplo, numa rede que processa imagens: camadas iniciais detectam bordas, camadas intermediárias detectam formas ou texturas, e camadas profundas reconhecem objetos inteiros. O treino de uma rede neural ocorre ajustando iterativamente os pesos (através de algoritmos como **backpropagation** e otimização por gradiente) para minimizar erros nas previsões. A importância das redes neurais está na sua **flexibilidade e poder de aproximação** – dados suficientes, elas conseguem modelar relações muito complexas. Este facto torna-as a espinha dorsal de aplicações como reconhecimento de fala, tradução automática e até veículos autónomos.

- **Modelos de Linguagem de Grande Escala (Large Language Models - LLMs):** São redes neurais profundas especializadas em **processamento de linguagem natural**, treinadas em coleções massivas de texto. O texto destaca que modelos como *GPT-3* ou *GPT-4* (da qual o ChatGPT é derivado) e outros (por exemplo, o Codex do GitHub Copilot) conseguem gerar texto e até código-fonte a partir de comandos em linguagem natural. Esses LLMs tipicamente possuem **milhões ou bilhões de parâmetros** (pesos da rede) e são treinados com quantidades gigantescas de dados de texto obtidos da internet, livros, código, etc. Como resultado, desenvolvem uma capacidade notável de **entender contexto e produzir respostas coerentes** em linguagem humana. Por exemplo, conseguem continuar um parágrafo, responder perguntas, traduzir idiomas e até escrever programas simples a partir de descrições. O conteúdo ressalta a utilidade desses modelos em auxiliar programadores (gerando pedaços de código ou sugerindo soluções) e noutras tarefas automatizadas de redação e atendimento via *chatbots*. Entretanto, também devemos comentar brevemente as **limitações**: apesar de impressionantes, LLMs podem apresentar falta de precisão em certos casos, **cometer erros ou "alucinações"**, produzindo afirmações incorretas que soam confiáveis. Também podem gerar código que parece plausível mas contém bugs subtis. Assim, destaca-se que, embora sejam ferramentas poderosas e inovadoras, requerem supervisão e verificação humana, especialmente em aplicações críticas. OS LLMs representam o estado da arte em redes neurais aplicadas à linguagem, ampliando as fronteiras do que sistemas de IA podem realizar em termos de compreensão e geração de texto.

Ferramentas e Bibliotecas em Python para IA e Ciência de Dados

A prática em IA e *machine learning* é grandemente facilitada por ferramentas modernas, em especial a linguagem Python e seu rico ecossistema de bibliotecas. Destacam-se os seguintes pontos:

- **Python como Linguagem Predominante:** Python é a **linguagem de programação preferida na ciência de dados e IA**. Isso deve-se à sua sintaxe simples e legibilidade, que permitem aos pesquisadores e programadores implementarem ideias rapidamente, bem como à vasta comunidade que contribui com ferramentas. No contexto apresentado, Python é utilizado nos exemplos práticos para demonstrar algoritmos e análise de dados de forma acessível.
- **Bibliotecas para Cálculo Numérico e Dados:** A linguagem em si é lenta em operações numéricas puras, mas possui bibliotecas otimizadas em C/Fortran. Temos bibliotecas fundamentais como **NumPy** (que fornece estruturas de dados de alto desempenho para matrizes e funções matemáticas vetorizadas) e **Pandas** (que oferece estruturas de dados como *DataFrames* para manipulação

eficiente de conjuntos de dados tabulares). Essas bibliotecas simplificam tarefas comuns, desde leituras de dados, limpeza e transformação, até cálculos estatísticos básicos, permitindo que cientistas de dados sejam produtivos sem reinventar funções básicas.

- **Bibliotecas para Machine Learning e Deep Learning:** É de notar que o ecossistema Python inclui **bibliotecas especializadas para ML e DL**. Por exemplo, o scikit-learn (muito usado para algoritmos clássicos de *machine learning* supervisionado e não supervisionado) e frameworks de *deep learning* como TensorFlow e PyTorch (que facilitam a construção e treino de redes neurais profundas, aproveitando aceleração por GPU). O Python, aliado às suas bibliotecas, fornece um **ambiente completo para experimentar e criar em IA** – desde a exploração de dados até à implementação de modelos complexos.
- **Visualização de Dados:** Outra vantagem ressaltada é a capacidade de Python gerar gráficos e visualizações facilmente, usando bibliotecas como **Matplotlib** ou seaborn. Visualizações são componentes essenciais da ciência de dados, ajudando a **interpretar resultados e comunicar insights**. Com poucas linhas de código Python, pode-se criar gráficos esclarecedores, o que torna a linguagem popular também para análise exploratória.

Em suma, o a combinação de Python + bibliotecas é uma peça chave na aplicação prática de IA: graças a isso, os profissionais podem focar-se no **problema e na lógica** em vez de detalhes de implementação de baixo nível, acelerando pesquisas e desenvolvimentos.

Dataficação: Dados em Todo Lugar e Impactos

Outro tópico relevante abordado é o fenômeno da **dataficação**, que se refere ao processo de transformar aspetos da vida quotidiana e operações de negócios em dados quantificáveis. Devemos discutir como a dataficação está a moldar setores inteiros e a afetar o nosso dia a dia:

- **Definição de Dataficação:** Trata-se da **conversão massiva de fenómenos em dados**. Com a digitalização crescente, praticamente toda atividade – transações financeiras, interações em redes sociais, passos medidos por wearables, padrões de navegação na internet – gera algum tipo de dado armazenável. Essa abundância de dados é combustível para IA e *machine learning*, pois fornece material para que os algoritmos aprendam e extraiam insights. Recolher e processar grandes quantidades de dados tornou-se uma prática central e estratégica em muitas organizações, inaugurando uma era orientada por dados.
- **Impacto em Diversos Setores:** Vários exemplos mostram como diferentes indústrias aproveitam a dataficação:

- Na **saúde**, a análise de grandes bases de dados clínicos e de pacientes (resultados de exames, históricos médicos, dados genómicos) ajuda a diagnosticar doenças com mais precisão e até antecipar riscos, personalizando tratamentos. A IA pode detectar padrões subtis em imagens médicas ou correlacionar sintomas e fatores genéticos que seriam imperceptíveis a olhos humanos, auxiliando médicos em decisões críticas.
- No setor do **retalho e alimentação**, dados de vendas, preferências de clientes e logística são recolhidos constantemente. Isso permite, por exemplo, prever necessidades de stock, personalizar ofertas e melhorar a gestão da cadeia de distribuição. Restaurantes e supermercados usam dados para ajustar menus e stocks conforme o comportamento de consumo, enquanto lojas físicas analisam fluxo de clientes e tempo de permanência para otimizar layouts.
- No setor **financeiro**, cada transação bancária ou operação de investimento gera dados que, quando analisados, auxiliam em detectar fraudes em tempo real e quantificar riscos de crédito ou de mercado com mais precisão. Instituições financeiras empregam algoritmos de *machine learning* para avaliar perfis de clientes, determinar taxas de empréstimo adequadas e até orientar estratégias de investimento, tudo baseado em grandes históricos de dados económicos.
- Em **recursos humanos** e gestão empresarial, a dataficação manifesta-se no uso de dados de desempenho, currículos e comportamento organizacional para melhorar contratações e avaliações. Por exemplo, alguns sistemas analisam dados de candidatos (formação, experiências, traços de perfil em redes sociais profissionais) para filtrar aqueles mais alinhados a uma vaga, ou monitorizam métricas de produtividade de funcionários para identificar necessidades de formação.
- **Impacto no Quotidiano:** Para além dos setores específicos, a dataficação afeta diretamente a vida de pessoas comuns. Hoje, **atividades rotineiras estão entrelaçadas a dados**: ao usar um smartphone, deixar um rastro de localização GPS e preferências de utilização de aplicações; ao assistir filmes numa plataforma de *streaming*, gerar dados de preferência que alimentarão recomendações personalizadas; ao fazer uma postagem numa rede social, contribuir para um vasto banco de dados sobre opiniões e tendências culturais. Essa onipresença de dados traz conveniências – como serviços sob medida e maior eficiência (por exemplo, trânsito otimizado via apps tipo Waze) – mas também levanta questões sobre privacidade e uso responsável das informações pessoais. Contudo é de destacar o **caráter transformador da dataficação**:

organizações e governos tomam decisões mais sustentadas em evidências empíricas e padrões extraídos dos dados. Em última análise, vivemos numa sociedade cada vez mais guiada por dados, onde compreender e saber extrair valor desse novo "recurso" se tornou fundamental.

- **Dataficando para IA:** Um ponto implícito é que sem dataficação não haveria o recente boom de IA. Os algoritmos de aprendizagem profunda famintos por dados só atingem bom desempenho porque contam com *datasets* massivos que vêm dessa conversão intensa de atividades em dados digitais. Portanto, a dataficação impulsiona diretamente o progresso da IA, ao passo que a IA, por sua vez, permite analisar e **descobrir insights acionáveis** a partir desses oceanos de dados, fechando um ciclo virtuoso.

Aplicações Práticas Demonstradas (Código em Python e Gráficos)

Podemos apontar alguns **exemplos práticos ilustrativos**, nos quais conceitos e ferramentas são aplicados em problemas reais, acompanhados de código em Python e visualizações gráficas. Esses estudos de caso breves mostram passo a passo como utilizar IA/data science na prática e servem para consolidar os tópicos discutidos:

- **Exemplo 1 – Cálculo de Estatísticas Simples:** O material inclui um exercício de gerar um conjunto de dados aleatórios e calcular medidas estatísticas básicas. Com auxílio do Python (e inclusive recorrendo ao ChatGPT para obter o código, mostrando a utilidade de LLMs como assistentes de programação), foram gerados 100 números aleatórios entre 1 e 20 e em seguida calculados a **média, mediana e desvio padrão** desses valores. Este exemplo evidencia como, com poucas linhas de código usando bibliotecas padrão (possivelmente usando random ou NumPy para geração dos números e funções do NumPy ou métodos manuais para as estatísticas), é possível extrair rapidamente informações sumarizadas de dados. A inclusão desse caso ressalta a importância de entender **estatísticas descritivas** – média e mediana dão ideia de tendência central dos dados, enquanto o desvio padrão indica a dispersão. Saber obter e interpretar esses números é fundamental no início de qualquer análise de dados. O uso do ChatGPT para auxiliar no código também demonstra na prática a sinergia entre IA generativa e tarefas de programação, agilizando o desenvolvimento.

```
import numpy as np
from statistics import median, stdev

# Gerar 100 observações aleatórias entre 1 e 20 (inclusive)
observacoes = np.random.randint(1, 21, 100)
```

```
# Calcular a média, mediana e desvio padrão
media = np.mean(observacoes)
mediana = median(observacoes)
desvio_padrao = np.std(observacoes)

# Exibir os resultados formatados
print(f'Média: {media:.2f}')
print(f'Mediana: {mediana}')
print(f'Desvio Padrão: {desvio_padrao:.2f}')
```

- Exemplo 2 – Análise de Desempenho Desportivo (Gráfico de Dispersão):** Outro exercício prático apresentado envolveu a análise de dados da **Premier League (campeonato inglês de futebol)**, especificamente explorando a relação entre número de golos marcados e número de vitórias das equipas numa temporada (a de 2018-2019). Com o Python e bibliotecas de visualização (como Matplotlib), o conteúdo mostra a construção de um **gráfico de dispersão (scatter plot)**, onde cada ponto representa uma equipa, posicionado conforme seus golos marcados (num eixo) e vitórias (no outro eixo). Além disso, cada ponto foi rotulado com o nome da respectiva equipa, facilitando a identificação no gráfico. Essa visualização permite verificar, por exemplo, se existe correlação positiva entre marcar muitos golos e ganhar muitos jogos – algo intuitivo, mas que o gráfico quantifica e deixa claro visualmente (equipas com mais vitórias tendem a estar também entre os que mais marcaram golos). Por fim, o exemplo vai além ao incorporar uma **linha de regressão linear** sobre os pontos do gráfico, indicando a tendência geral. Esse ajuste linear serve como uma aproximação simples da relação entre golos e vitórias: se a linha tiver inclinação positiva acentuada, confirma que geralmente mais golos se associam a mais vitórias. A demonstração mostra como pedir ao ChatGPT para gerar código adicional (no caso, para calcular e desenhar a linha de melhor ajuste) pode rapidamente adicionar valor à análise. O resultado apresentado combina todos esses elementos – dispersão de pontos, rótulos de equipas e linha de tendência – proporcionando uma visualização rica. Esta seção prática evidencia a aplicação conjunta de **ciência de dados e aprendizagem de máquina** num caso concreto, reforçando conceitos: visualmente, fica evidente a importância de dados esportivos e como técnicas simples de análise podem extrair insights (por exemplo, identificando equipas fora da curva que marcaram muitos golos mas não tiveram tantas vitórias, ou vice-versa).

Input:

Cria um **gráfico em Python** que represente a relação entre o número de vitórias (wins) e o número de golos marcados (goals_scored) pelas equipas da Premier League na época 2018-2019.

```
import matplotlib.pyplot as plt
import numpy as np

# Dados das equipas
teams = [
    "Arsenal", "Tottenham Hotspur", "Manchester City", "Leicester City",
    "Crystal Palace", "Everton", "Burnley", "Southampton", "AFC Bournemouth",
    "Manchester United", "Liverpool", "Chelsea", "West Ham United", "Watford",
    "Newcastle United", "Cardiff City", "Fulham", "Brighton & Hove Albion",
    "Huddersfield Town", "Wolverhampton Wanderers"
]

wins = [21, 23, 32, 15, 14, 15, 11, 9, 13, 19, 30, 21, 15, 14, 12, 10, 7, 9,
3, 16]
goals_scored = [73, 67, 95, 51, 51, 54, 45, 45, 56, 65, 89, 63, 52, 52, 42,
34, 34, 35, 22, 47]

# Criar o gráfico de dispersão
plt.figure(figsize=(12, 7))
plt.scatter(wins, goals_scored, color='blue', label='Equipas')

# Adicionar os nomes das equipas
for i, team in enumerate(teams):
    plt.text(wins[i] + 0.3, goals_scored[i], team, fontsize=9)

# Calcular e desenhar a linha de regressão
coef = np.polyfit(wins, goals_scored, 1) # grau 1 = reta
regressao = np.poly1d(coef)
x_vals = np.linspace(min(wins), max(wins), 100)
plt.plot(x_vals, regressao(x_vals), color='red', linestyle='--',
label='Regressão Linear')

# Títulos e eixos
plt.title('Relação entre Vitórias e Golos Marcados - Premier League 2018/19')
plt.xlabel('Vitórias (Wins)')
plt.ylabel('Golos Marcados (Goals Scored)')
plt.grid(True)
plt.legend()
plt.tight_layout()
plt.show()

# Mostrar o coeficiente de correlação
correlacao = np.corrcoef(wins, goals_scored)[0, 1]
print(f'Correlação entre vitórias e golos marcados: {correlacao:.2f}')
```

```
#Este gráfico:

#Mostra os pontos com os nomes das equipas.

#Inclui uma linha de tendência vermelha tracejada (regressão linear).

#Apresenta a correlação no terminal.
```

Em conjunto, essas aplicações práticas cumprem um papel duplo no conteúdo. Didaticamente, **solidificam a teoria** mostrando como usar Python e bibliotecas para implementar desde cálculos básicos até visualizações e modelos simples. Conceitualmente, ligam de volta aos tópicos principais: no Exemplo 1, temos estatística e geração de dados (ligação com dataficação e ferramentas Python); no Exemplo 2, temos visualização e até uma pitada de *machine learning* (regressão linear é um modelo preditivo simples) aplicado em um contexto real (desportivo). Além disso, a presença do ChatGPT nessas interações ilustra, na prática, **como os LLMs podem auxiliar no trabalho de análise de dados**, seja escrevendo código boilerplate ou explicando passos. Assim, não apenas se aprendem os conceitos isoladamente, mas também se vê **a integração entre eles** – IA fornecendo código para IA, por assim dizer – num ciclo que demonstra a potência das ferramentas modernas disponíveis.

Conclusão

Temos assum uma **visão abrangente e estruturada sobre Inteligência Artificial e Ciência de Dados**, começando pelos conceitos fundamentais e evoluindo até aplicações práticas. Cada seção temática aborda um aspecto crucial:

- Conceitos como IA, *machine learning*, *deep learning* e IA generativa definem o panorama e importância de cada um.
- Exemplos históricos (Deep Blue, AlphaGo) **contextualizam os avanços**, mostrando marcos que provam o potencial da IA.
- Diferentes abordagens de aprendizado (supervisionado, não supervisionado, etc.) esclarecem como **modelos aprendem de formas variadas** conforme o tipo de problema.
- A comparação entre ML e DL ajuda a entender quando e por que usar técnicas mais simples ou mais profundas, um conhecimento prático para quem trabalha na área.

- A explicação de redes neurais e LLMs conecta a teoria à prática atual, destacando **por que redes profundas revolucionaram o campo** e como isso culmina nos modernos modelos de linguagem.
- A parte de ferramentas enfatiza que, embora a teoria seja vital, **é o ecossistema prático (Python e bibliotecas) que torna possível aplicar essas ideias** eficientemente no dia a dia.
- A discussão sobre dataficação expande o horizonte, mostrando que **IA e dados não existem no vácuo**, mas sim em um contexto social e industrial que se transforma por causa deles – impactando saúde, negócios e nossa rotina pessoal.
- Finalmente, os exemplos práticos atuam como uma síntese viva, provando em pequenos projetos como todos esses elementos (dados, algoritmos, ferramentas e modelos preditivos) se combinam para gerar insights e soluções.

Em tom analítico, percebemos que o conteúdo não apenas informa, mas também **comenta a relevância**: seja ao lembrar a significância histórica de uma máquina vencer um humano, ou ao apontar as limitações de um modelo de linguagem apesar de sua sofisticação. Essa perspectiva crítica é importante para formar uma visão equilibrada – entusiasmar-se com as possibilidades da IA, sem ignorar desafios e contextos.

Em suma, este é um **guia introdutório** que alinha teoria e prática. Do conceito de IA ao código Python executável, somos levados por um percurso que demonstra como **dados e algoritmos estão a transformar o mundo** 😊, e quão essencial é compreender os seus fundamentos para aproveitar seu potencial de forma consciente e eficaz.