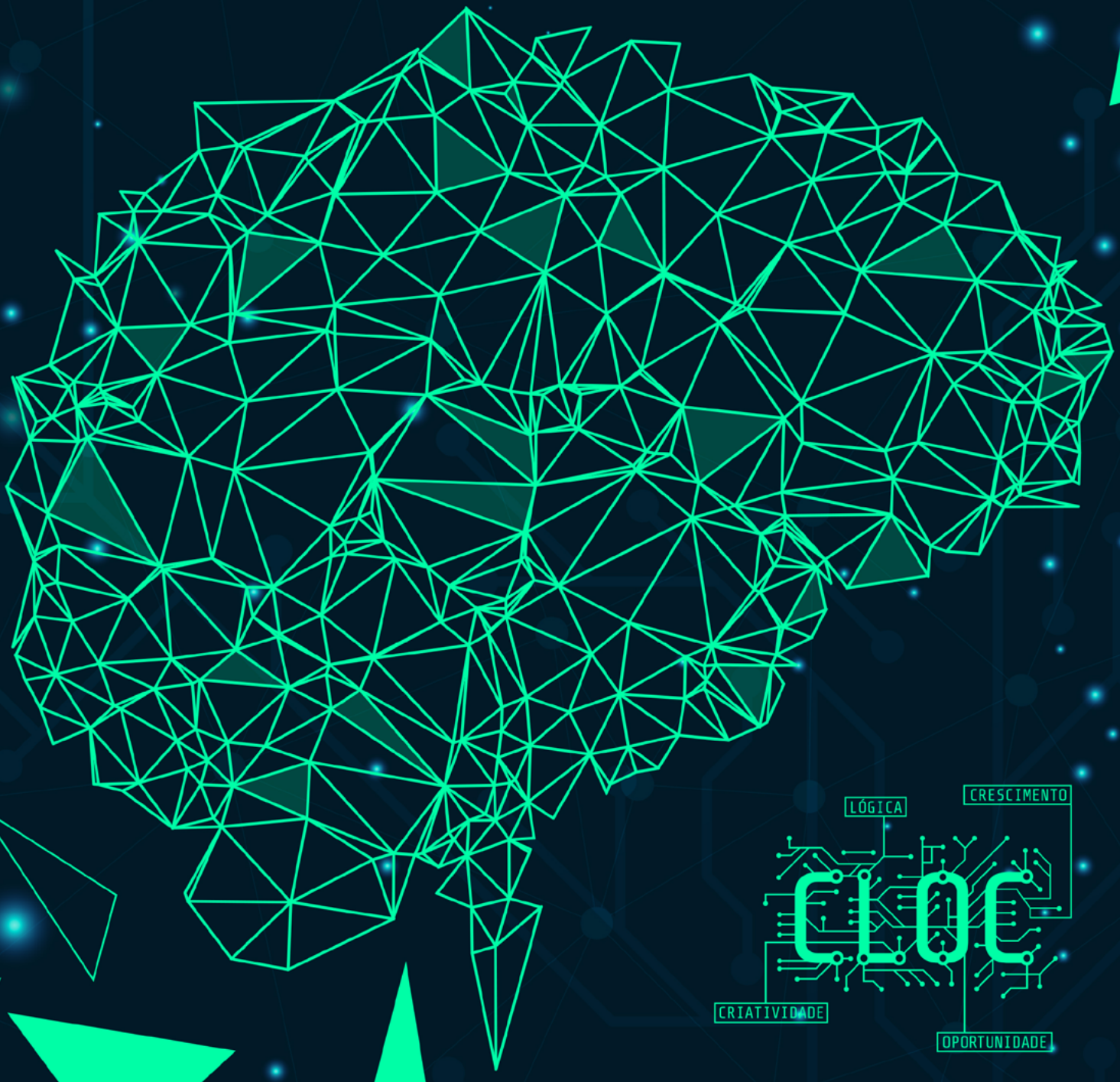
IA

inteligência artificial





Este caderno está licenciado com uma Licença Creative Commons Atribuição-NãoComercial-SemDerivações 4.0 Internacional. (CC BY-NC-ND 4.0).

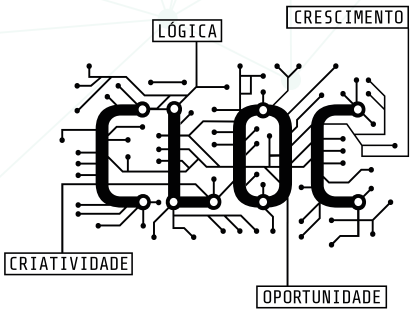
Esta licença requer que os reutilizadores deem crédito ao criador. Ela permite que os reutilizadores copiem e distribuam o material em qualquer meio ou formato em forma não adaptada e somente para fins não comerciais.

BY: O crédito deve ser dado a você, o criador.

NC: Somente o uso não comercial do seu trabalho é permitido. Não comercial significa que não é destinado ou direcionado principalmente para vantagem comercial ou compensação monetária.

ND: Não são permitidos derivados ou adaptações do seu trabalho.

Aracaju: THP, 1ª edição, 2025.



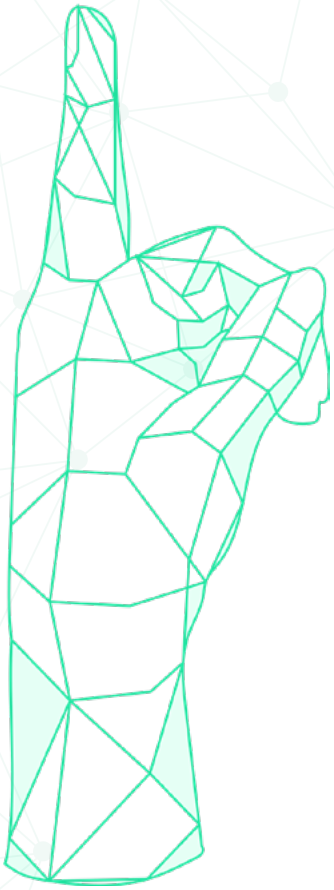
Coordenadora do Projeto

RENATA SANTOS CELESTINO

Projeto Gráfico

PATRICIA KARIN S. FABER

SUMÁRIO



1. Módulo 1: Introdução à Inteligência Artificial 04
2. Módulo 2: História da Inteligência Artificial 08
3. Módulo 3: IA vs Humana 12
4. Módulo 4: Primeiros contatos com IA 16
5. Módulo 5: Conceitos da Inteligência artificial 24
6. Módulo 6: Praticando Aprendizado de Máquina 27
7. Módulo 7: Primeiros Contatos - Uma breve Introdução a Python e Google Colab 33
8. Módulo 8: Conhecendo o Primeiro Modelo de IA 39
9. Módulo 9: Mãos à Obra! - Prevendo os dados a partir de características da flor 42
10. Módulo 10: Mãos à Obra! - Interpretabilidade do Modelo de IA (Árvore de Decisão) 46
11. Módulo 11: Mãos à Obra! - Classificando Dígitos com o MNIST 49
12. Módulo 12: Mãos à Obra! - Trabalho Final 54

Módulo 1

INTRODUÇÃO À INTELIGÊNCIA ARTIFICIAL

Tema: Introduzindo a Inteligência Artificial

Duração: 2h

Material necessários: Notebooks, projetor, Internet, caderno e lápis

Objetivo: Apresentar o conceito de IA de forma simples e suas vantagens.

CONTEXTUALIZAÇÃO/ENGAJAR

A inteligência artificial, ou IA como chamamos para simplificar, é uma tecnologia que está cada vez mais presente em nossas vidas. Mas o que é exatamente a IA? De forma bem simples, podemos dizer que é quando os computadores conseguem fazer tarefas que normalmente precisariam da inteligência humana. Pense nos assistentes de voz como Alexa ou Google Assistente. Quando você faz uma pergunta, eles entendem o que você disse, procuram a resposta e respondem de um jeito que faz sentido. Isso é IA em ação! Ou quando você usa o celular para tirar uma foto e ele reconhece automaticamente que tem um rosto na imagem, isso também é IA.

A IA não é uma tecnologia única, mas um conjunto de várias técnicas e ferramentas que permitem que os computadores "aprendam" com exemplos, reconheçam padrões e tomem decisões. É como se estivéssemos ensinando os computadores a pensar, mas de um jeito diferente de como nós humanos pensamos.

Para entender melhor a IA, é interessante comparar a inteligência humana com a inteligência artificial: A inteligência humana é muito boa em:

Criatividade
conseguimos inventar coisas novas e ter ideias originais

Emoções
sentimos alegria, tristeza, medo e amor

Intuição
às vezes "sabemos" algo sem conseguir explicar como

Aprendizado com poucas experiências
uma criança precisa ver poucos cachorros para reconhecer um cachorro novo

Já a inteligência artificial é muito boa em:

Velocidade
processa informações muito rapidamente

Precisão
não se cansa e raramente comete erros em tarefas específicas

Análise de dados
encontra padrões em grandes quantidades de informações

Repetição
faz a mesma tarefa milhões de vezes sem reclamar

O mais interessante é que a inteligência humana e a artificial não competem entre si, elas se complementam! Enquanto nós humanos somos criativos e temos intuição, a IA é rápida e precisa.

Na medicina, médicos usam IA para ajudar a analisar exames de imagem e detectar doenças mais cedo. O médico tem o conhecimento, a experiência e a empatia para tratar o paciente, enquanto a IA ajuda a identificar padrões que poderiam passar despercebidos.

Aprender sobre IA hoje é muito importante porque essa tecnologia está mudando o mundo ao nosso redor. Ela está presente em muitas áreas:

- Na **medicina**, ajudando a diagnosticar doenças
- Na **educação**, criando experiências de aprendizado personalizadas
- Nos **transportes**, com carros que dirigem sozinhos
- Na **arte**, criando músicas, imagens e textos
- Nos **jogos**, com personagens que aprendem e se adaptam.

As crianças e jovens podem aprender sobre IA de várias formas divertidas:

- Com jogos e brinquedos que usam IA
- Com cursos online feitos especialmente para iniciantes
- Participando de clubes e acampamentos de tecnologia
- Fazendo projetos práticos, como criar um pequeno robô ou um programa simples

O mais importante é lembrar que a IA é uma ferramenta criada por humanos para nos ajudar. Ela não vai substituir a inteligência humana, mas sim trabalhar junto com ela para resolver problemas e criar um mundo melhor.

ATIVIDADE DE PESQUISA E FIXAÇÃO

Em grupos, pesquisem e listem:

1. Três características da inteligência humana (como criatividade, emoções, intuição, aprendizado)
2. Três características da inteligência artificial (como velocidade, precisão, análise de dados, repetição)

Depois, pensem em como essas duas formas de inteligência podem trabalhar juntas. Por exemplo, como a criatividade humana pode se combinar com a velocidade da IA? Ou como a intuição humana pode ajudar a direcionar a análise de dados feita pela IA?

*Vamos separar 20 minutos para essa atividade. Usem os notebooks para pesquisar e anotem suas descobertas no caderno.

DESAFIO/COMPARTILHAR

Cada grupo vai apresentar um exemplo prático de como a inteligência humana e a inteligência artificial podem trabalhar juntas para resolver um problema ou criar algo novo.

Pensem em exemplos do dia a dia ou em áreas como:

- **Saúde** (diagnóstico de doenças, criação de medicamentos)
- **Educação** (ensino personalizado, correção de atividades)
- **Meio ambiente** (monitoramento de florestas, previsão de desastres naturais)
- **Arte** (criação de músicas, pinturas, histórias)
- **Esportes** (análise de desempenho, treinamento personalizado)

Na apresentação, expliquem:

1. Qual é o problema ou a tarefa?
2. Como a inteligência humana ajuda?
3. Como a inteligência artificial ajuda?
4. Por que juntas elas funcionam melhor do que separadas?

Depois das apresentações, vamos conversar sobre como a IA pode ser uma ferramenta poderosa quando usada junto com nossas habilidades humanas. Também vamos discutir a importância de usar a IA de forma responsável e ética.

RESUMO

No site ***https://cartilha-dzgm63.manus.space/introducao*** contém o resumo desta aula

Módulo 2

HISTÓRIA DA **INTELIGÊNCIA ARTIFICIAL**

Tema: Linha temporal da IA

Duração: 2h

Material necessários: Notebooks, projetor, Internet, caderno e lápis

Objetivo: Apresentar um breve histórico da IA e seus marcos importantes, contextualizando sua evolução e impacto na sociedade.

CONTEXTUALIZAÇÃO/ENGAJAR

A história da inteligência artificial começou há bastante tempo. Antes mesmo de termos computadores como conhecemos hoje, as pessoas já imaginavam máquinas que pudessem pensar como humanos. Nas histórias antigas e nos mitos, existiam seres criados por humanos que ganhavam vida e tinham inteligência própria. Mas a IA como conhecemos hoje começou a tomar forma por volta dos anos 1950.

Um cientista chamado Alan Turing fez uma pergunta importante: "As máquinas podem pensar?". Ele criou um teste, chamado Teste de Turing, para verificar se uma máquina poderia se passar por um ser humano em uma conversa. Em 1956, aconteceu um evento muito importante: a Conferência de Dartmouth. Foi nessa conferência que o termo "Inteligência Artificial" foi usado pela primeira vez. Um grupo de cientistas se reuniu para discutir como criar máquinas que pudessem simular a inteligência humana. Eles estavam muito otimistas e achavam que em poucos anos teriam máquinas realmente inteligentes. Mas logo descobriram que criar uma IA não era tão fácil quanto pensavam.

Alguns anos depois, veio o que chamamos de "inverno da IA", período em que o progresso ficou mais lento e o financiamento para pesquisas diminuiu. Isso aconteceu porque as expectativas eram muito altas e a tecnologia da época não conseguia alcançá-las. Mesmo assim, a cada novo ciclo de pesquisa, os cientistas conseguiram avançar um pouco mais.

Nos anos 1960, foi criado o ELIZA, um dos primeiros programas de computador que podia conversar com pessoas. O ELIZA simulava um psicólogo e conseguia manter conversas simples, fazendo as pessoas acreditarem que estavam falando com alguém que as entendia.

Nos anos 1970 e 1980, os pesquisadores começaram a criar "sistemas especialistas" - programas que tinham conhecimento específico em áreas como medicina ou geologia. Esses sistemas poderiam ajudar especialistas humanos a tomar decisões.

Nos anos 1990 e 2000, com computadores mais potentes, a IA começou a usar mais o "aprendizado de máquina". Em vez de programar regras específicas, os cientistas criaram sistemas que podiam aprender com exemplos e dados.

Em 2011, o computador Watson da IBM venceu humanos no programa de TV Jeopardy!, mostrando que podia entender perguntas complexas em linguagem natural (linguagem humana).

A partir de 2017, começamos a ver o surgimento da IA generativa - sistemas que podem criar textos, imagens, músicas e até vídeos. Exemplos como o ChatGPT e o DALL-E mostram como a IA pode ser criativa.

Hoje, a IA está em muitos lugares do nosso dia a dia:

Nos celulares	Nos carros	Nos hospitais	Nas escolas

Ela ajuda a diagnosticar doenças, traduzir idiomas, recomendar filmes e músicas, e muito mais.



É importante salientar como os filmes e livros de ficção científica muitas vezes imaginaram a IA antes dela existir de verdade. Filmes como "2001: Uma Odisseia no Espaço" (com o computador HAL 9000) ou "Blade Runner" mostraram ideias sobre IA que inspiraram cientistas e engenheiros. Muitas das tecnologias que temos hoje foram primeiro imaginadas em histórias de ficção! Entender essa história nos ajuda a ver como chegamos até aqui e a pensar sobre o futuro da IA. Também nos lembra que a IA é uma criação humana, feita para nos ajudar a resolver problemas e melhorar nossas vidas.

ATIVIDADE DE PESQUISA E FIXAÇÃO

Cada grupo vai escolher um destes marcos para pesquisar:

1. A Conferência de Dartmouth (1956) - Quando o termo "Inteligência Artificial" foi criado
2. O programa ELIZA (1966) - Um dos primeiros chatbots da história
3. O AlphaGo (2016) - O programa de IA que venceu o campeão mundial de Go
4. A IA generativa (a partir de 2017), como o ChatGPT e o DALL-E

Na sua pesquisa, tente descobrir:

- Por que esse momento foi importante?
- Quem foram as pessoas envolvidas?
- Como isso mudou o campo da IA?

*Vamos separar 25 minutos para essa atividade. Usem os notebooks para pesquisar e anotem suas descobertas no caderno.

DESAFIO/ COMPARTILHAR

Agora cada grupo vai compartilhar o que descobriu sobre o marco histórico que pesquisou e sobre o filme ou livro que escolheu.

Explique para a turma:

1. O que aconteceu nesse momento da história da IA?
2. Por que isso foi importante?
3. Essas ideias da ficção se tornaram realidade ou ainda são apenas imaginação?

Depois que todos os grupos se apresentarem, vamos conversar sobre como a história da IA nos ajuda a entender melhor o presente e a pensar sobre o futuro dessa tecnologia. Vamos discutir como as ideias foram evoluindo ao longo do tempo e como a ficção científica muitas vezes inspirou os cientistas a criar coisas novas.

RESUMO

No site ***https://cartilha-dzgm63.manus.space/historia*** contém o resumo desta aula

Módulo 3

IA VS HUMANA

Tema: Diferenciando a Inteligência Humana da Artificial

Duração: 2h

Material necessários: Notebooks, projetor, Internet, caderno e lápis

Objetivo: Mostrar as diferenças entre a inteligência humana e a artificial de forma simples.

CONTEXTUALIZAÇÃO/ENGAJAR

Quando falamos de inteligência artificial (IA) e inteligência humana, estamos falando de duas coisas muito diferentes que podem trabalhar juntas. É como comparar um avião com um pássaro, ambos voam, mas de jeitos bem diferentes.

A inteligência humana é algo que desenvolvemos ao longo de milhões de anos de evolução. Nosso cérebro é uma máquina incrível que nos permite pensar, sentir, criar e entender o mundo ao nosso redor. Algumas coisas que só a inteligência humana consegue fazer (pelo menos por enquanto):

Diagnósticos médicos:

IA: Pode analisar milhares de exames de raio-X rapidamente e identificar padrões que indicam doenças com grande precisão.

Humanos: Médicos usam empatia para entender os sintomas descritos pelo paciente, consideram o histórico familiar e fazem perguntas importantes que uma IA não saberia fazer.

Juntos: O médico usa a análise da IA como uma segunda opinião, mas toma a decisão final considerando fatores que a IA não consegue avaliar.

Assistentes de voz:

IA: Alexa, Siri e Google Assistente reconhecem comandos de voz e executam tarefas como tocar música ou responder perguntas simples.

Humanos: Entendem o contexto, as emoções na voz e as intenções por trás das palavras.

Juntos: Usamos assistentes de voz para tarefas simples e rápidas, mas recorremos a humanos quando precisamos de compreensão mais profunda.

Aprendizado infantil:

IA: Precisa de milhares de exemplos para aprender a identificar um objeto.

Crianças: Podem ver um cachorro apenas algumas vezes e já reconhecem qualquer cachorro, mesmo muito diferentes entre si.

Juntos: Programas educacionais com IA podem se adaptar ao ritmo de cada criança, mas os professores humanos oferecem motivação, conexão emocional e orientação personalizada.

Um exemplo interessante para entender como a IA "aprende" é pensar em como ela classificaria animais.

Imagine que queremos ensinar uma IA a diferenciar entre animais com cascos (como cavalos e vacas) e animais sem cascos (como cachorros e gatos).

Para uma IA aprender isso, precisamos:

1. **Coletar dados:** Reunir muitas imagens de diferentes animais, cada uma marcada como "com casco" ou "sem casco".
2. **Identificar padrões:** A IA analisa as características de cada grupo (formato das patas, pernas, corpo).
3. **Criar regras:** A IA cria um *"modelo"* com regras como *"Se tem casco dividido, é um animal com casco. Se tem patas com garras, é um animal sem casco."*
4. **Testar e classificar:** Quando mostramos um animal novo, a IA aplica as regras que aprendeu para decidir se ele tem casco ou não.

Este é um exemplo simples de como a IA usa dados para aprender a classificar coisas. É assim que ela consegue identificar spam em e-mails, reconhecer rostos em fotos ou até mesmo diagnosticar doenças a partir de imagens médicas.

A grande diferença é que nós, humanos, não precisamos ver milhares de exemplos para aprender algo novo. Temos uma capacidade natural de generalizar a partir de poucos exemplos e de usar nosso conhecimento de um assunto para entender outro completamente diferente.

ATIVIDADE DE PESQUISA E FIXAÇÃO

Vamos analisar diferentes animais e classificá-los em grupos de acordo com a característica ungulados ou não ungulados, compreendendo como esse processo se relaciona ao modo como a inteligência artificial aprende a categorizar informações.



DESAFIO/COMPARTILHAR

1. As imagens dos animais devem ser apresentadas uma a uma.
2. A cada apresentação, os alunos devem indicar a qual grupo acreditam que o animal pertence: **ungulado** ou **não ungulado**.
3. Após a escolha, deve-se confirmar a classificação correta e posicionar o animal no grupo adequado da figura.
4. Ao final, deve-se destacar o caso do porco, que aparece com um ponto de interrogação:
   - O animal deve ser analisado e discutido coletivamente.
   - Conclui-se que o porco pertence ao grupo dos ungulados, por possuir casco fendido, mesmo não sendo tão semelhante a outros exemplos clássicos desse grupo, como cavalos ou vacas.

DISCUSSÃO FINAL:

Quais características foram utilizadas para tomar a decisão de classificação?

Como os alunos chegaram às respostas: pela observação das patas, pelo formato do corpo ou por conhecimentos prévios?

Essa comparação deve ser conectada com a diferença entre aprendizado humano e aprendizado artificial:

- Humanos conseguem aprender e generalizar a partir de poucos exemplos.
- A inteligência artificial necessita de muitos dados e repetições para formar padrões confiáveis.

CONCLUSÃO

A atividade demonstra que tanto a inteligência humana quanto a inteligência artificial utilizam dados para classificar elementos, mas com abordagens diferentes. O caso do porco ilustra como a IA pode enfrentar dificuldades quando o padrão não é tão evidente, enquanto os humanos conseguem recorrer a conhecimentos mais amplos para chegar à resposta correta.

*Vamos separar 10 minutos para essa atividade.

RESUMO

No site **https://cartilha-dzgm63.manus.space/ia-vs-humana** contém o resumo desta aula.

Módulo 2

PRIMEIROS CONTATOS COM **IA**

Tema: Explorando a IA na Prática: Brincando com IA

Duração: 2h

Material necessários: Notebooks, projetor e Internet

Objetivo: Permitir que os alunos experimentem e testem tecnologias de IA de forma interativa e divertida, compreendendo seus princípios básicos através da prática.

CONTEXTUALIZAÇÃO/ENGAJAR

Depois de aprender sobre o que é a IA, sua história e onde ela está presente, chegou a hora de colocar a mão na massa. Neste módulo, vamos explorar a inteligência artificial de uma forma muito divertida: experimentando e brincando com algumas ferramentas e jogos que usam IA. A ideia é que vocês vejam como a IA funciona na prática, de um jeito leve e interativo, sem precisar ser um especialista em programação. Vamos descobrir como a IA pode nos ajudar a criar, a jogar e a interagir com o mundo de maneiras diferentes.

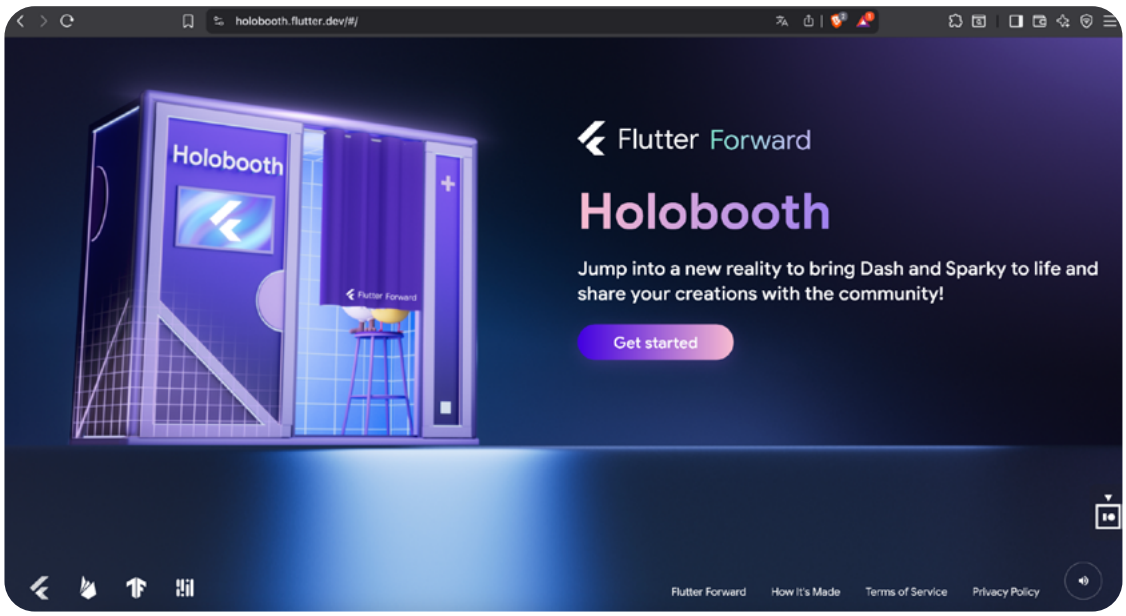
ATIVIDADE DE PESQUISA E FIXAÇÃO

Vamos dividir a turma em grupos e cada grupo terá a missão de explorar um ou mais dos links abaixo. A ideia é que vocês testem as ferramentas, entendam como elas funcionam e descubram o que a IA está fazendo por trás das cenas. Não se preocupem em entender tudo, mas sim em experimentar e observar.

Experimento 1

Um experimento interativo que usa IA para criar efeitos visuais em tempo real com a sua imagem. Vocês podem se transformar em personagens ou interagir com objetos virtuais.

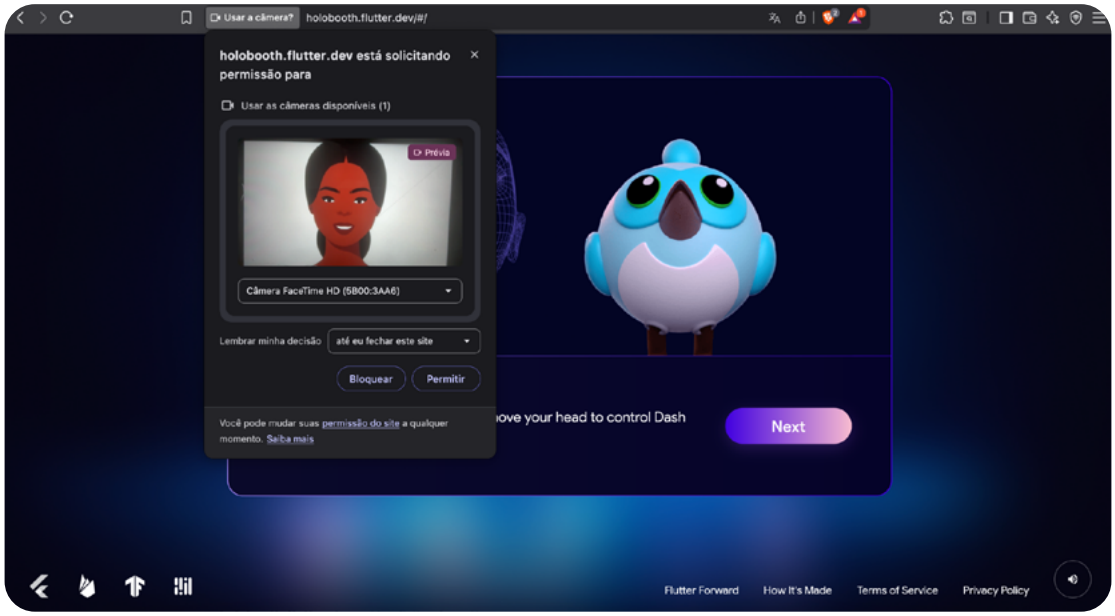
Acesse o link **https://holobooth.flutter.dev**



TELA INICIAL DA PÁGINA HOLOBOOTH

O site apresenta um botão roxo chamado **Get started**.

Esse botão é a porta de entrada para começar a experiência de criar e controlar o avatar.

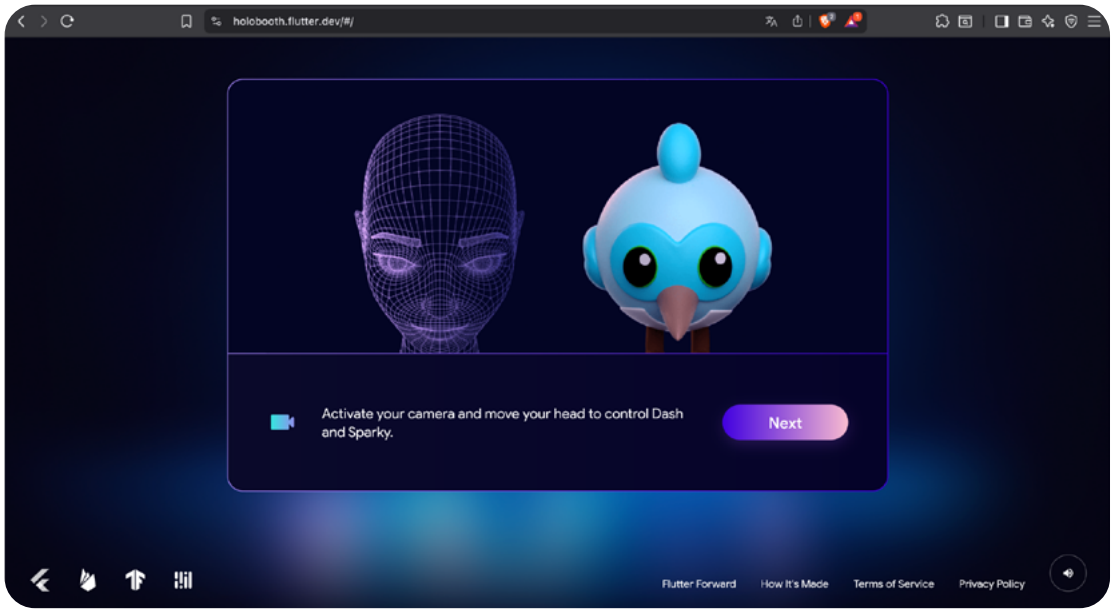


TELA INICIAL DA PÁGINA HOLOBOOTH

Ao clicar em **Get started**, o navegador solicita permissão para usar a câmera do dispositivo.

É exibida uma janela de pop-up perguntando se deseja **Permitir** ou **Bloquear** o acesso.

Para continuar, é necessário clicar em **"Permitir"**, pois a aplicação utiliza a câmera para reconhecer o rosto humano e controlar o avatar.



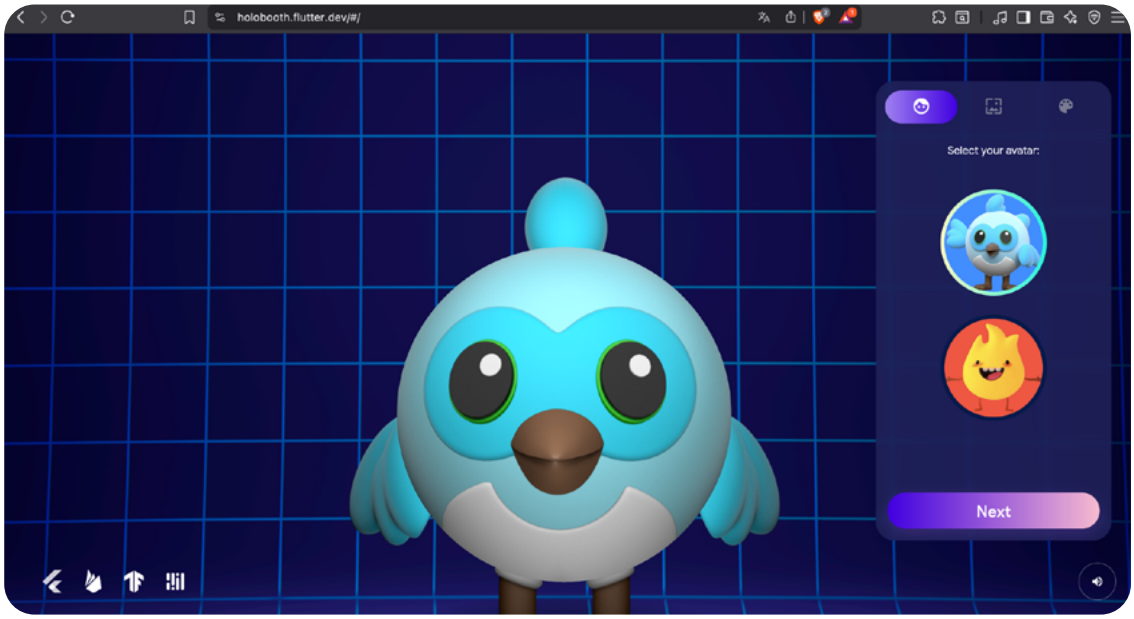
RECONHECIMENTO FACIAL E INTERAÇÃO

O sistema pede para **ativar a câmera.**

O usuário deve **mover a cabeça** para que a IA aprenda a reconhecer os movimentos e associá-los ao personagem virtual (Dash).

Depois de concluir esse processo, é necessário clicar em **Next** para continuar.

Esse passo é como um ***"treinamento inicial"***, em que o programa calibra os movimentos faciais para que o avatar responda corretamente.



ESCOLHA DO AVATAR

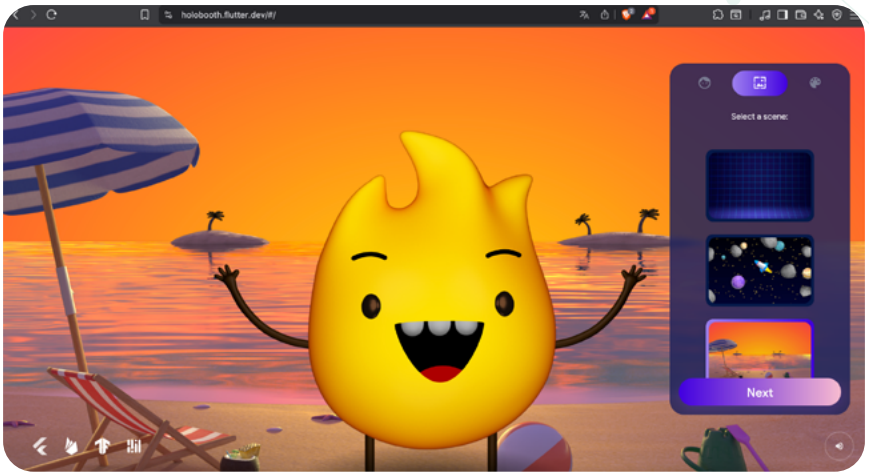
Após o reconhecimento dos movimentos, aparece a tela de seleção de avatar.

Há opções disponíveis, como o pássaro Dash (mascote do Flutter) e outros personagens.

O usuário pode se mover diante da câmera e observar como o avatar imita seus gestos.

Em seguida, basta clicar em Next para avançar para a próxima etapa.

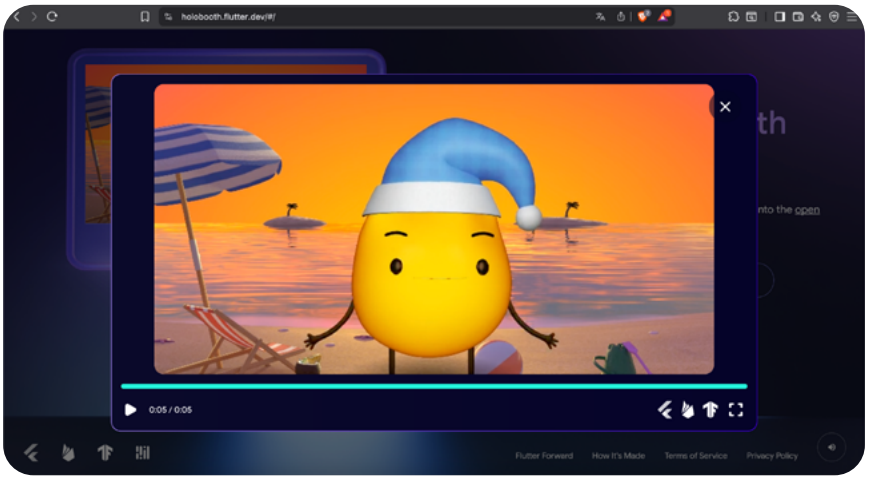
Aqui o sistema demonstra como a IA interpreta em tempo real os movimentos humanos e os traduz em animações no personagem escolhido.



PRÓXIMOS PASSOS
(não mostrados completamente, mas inferidos)

Depois da escolha do avatar, a próxima etapa permite selecionar cenários de fundo para personalizar a experiência.

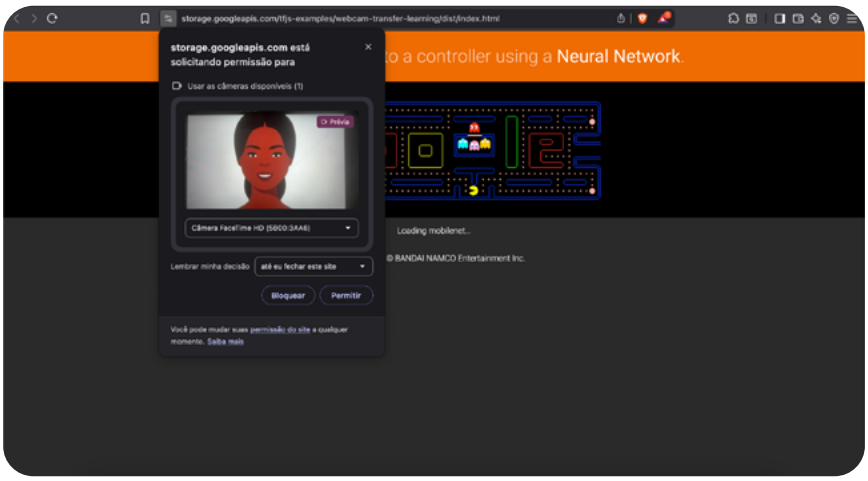
O usuário poderá gravar vídeos, tirar fotos ou interagir com o ambiente virtual controlando o avatar pela própria expressão facial e movimentos da cabeça.



Experimento 2

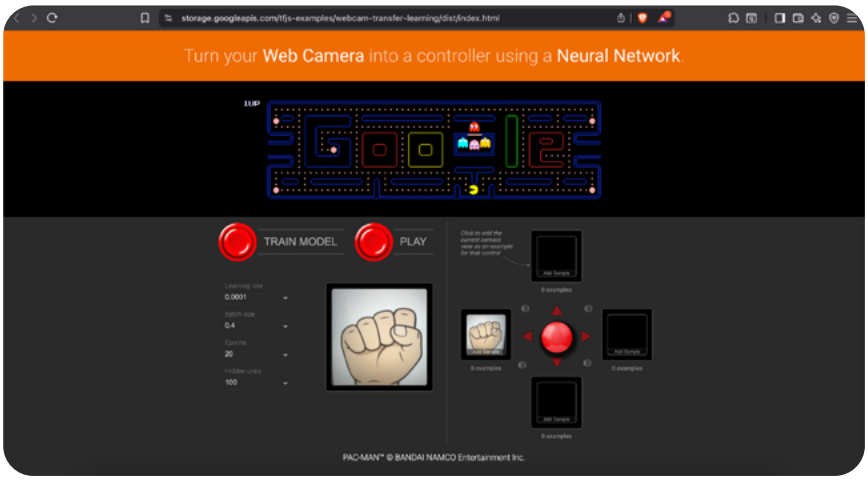
É uma aplicação interativa que permite usar a webcam do seu dispositivo como um controle (joystick) usando técnicas de aprendizado de máquina. Ou seja: aproveita um modelo já treinado como base, e "ajusta" para reconhecer novos gestos ou poses que o usuário forneça.

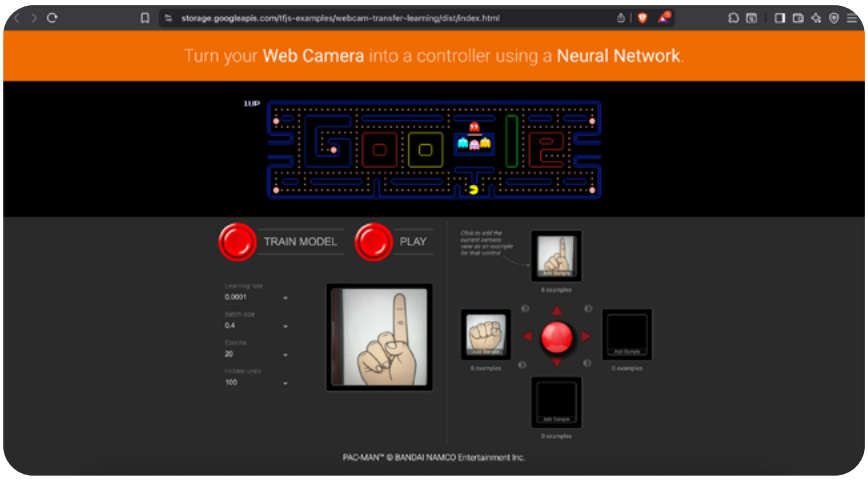
Acesse o link **Webcam Pacman**(https://storage.googleapis.com/tfjs-examples/webcam-transfer-learning/dist/index.html

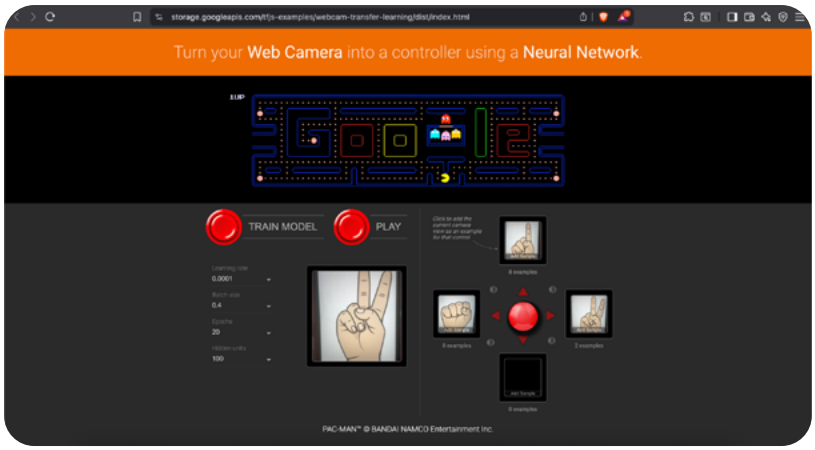


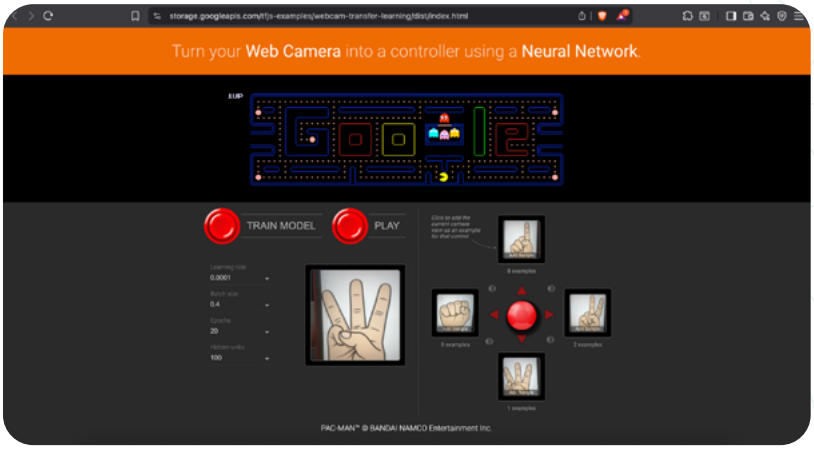
CARREGAMENTO DO MODELO BASE

O usuário pode usar a câmera para capturar exemplos de diferentes gestos ou posições. Há botões do tipo "Add Sample" para diferentes controles. Por exemplo, um gesto pode corresponder ao comando "esquerda", outro gesto ao comando "direita", etc.









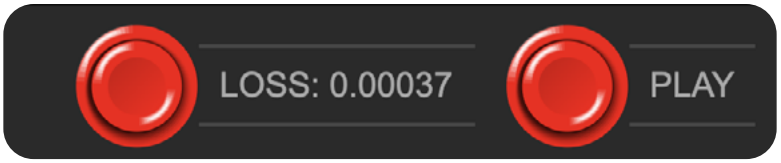
COLETA DE EXEMPLOS (à direita da tela)

Existem botões em formato de joystick (setas) para indicar cada direção.

O usuário posiciona a mão em frente à câmera e clica no botão correspondente para registrar exemplos.

Por exemplo:
- Gesto com 1 dedo levantado pode ser usado para "cima".
- Gesto com 2 dedos levantados pode ser usado para "direita".

Cada clique adiciona várias imagens capturadas da webcam à classe selecionada.



BOTÃO *"TRAIN MODEL"*

Após coletar as amostras para cada gesto/botão de controle e ajustar os parâmetros, inicia-se o processo de treinamento.

O modelo “aprende” a distinguir seus gestos com base nas imagens capturadas

Loss: é o valor do erro

MODO DE USO (“PLAY”)

Depois que o modelo estiver treinado, passa-se para o modo Play, onde os gestos capturados via webcam ativam os controles no estilo Pac-Man, ou outro jogo/ controlador definido. Ou seja: mover a cabeça ou mostrar a mão por exemplo pode corresponder a comandos como mover à esquerda, direita, cima, baixo.

Vamos separar 40 minutos para essa atividade, incluindo a exploração dos links e a discussão em grupo. É importante que os alunos se sintam à vontade para pesquisar e tirar dúvidas entre si.

DESAFIO/ COMPARTILHAR

Fazer uma roda de conversa para compartilhar as experiências, tirar dúvidas e refletir sobre como a IA pode ser usada de forma criativa e útil. O objetivo é que todos saiam com uma compreensão mais prática e divertida do que a IA é capaz de fazer, e que se sintam inspirados a explorar mais essa tecnologia por conta própria. Pode apresentar o site: Demonstrações do TensorFlow.js para que os alunos explorem outras técnicas como essas.

Módulo 5

CONCEITOS DA **INTELIGÊNCIA ARTIFICIAL**

Tema: Entendendo sobre com dados

Duração: 2h

Material necessários: Notebooks, projetor, Internet, caderno e lápis

Objetivo: Apresentar, de forma simples e prática, o que são dados e como eles estão presentes no nosso dia a dia. A partir disso, mostrar como a Inteligência Artificial utiliza esses dados para aprender, identificar padrões e tomar decisões.

CONTEXTUALIZAÇÃO/ENGAJAR

Nos módulos anteriores, vimos que a IA está em todo lugar: ela ajuda a encontrar músicas, sugerir filmes, reconhecer rostos, traduzir frases e até criar desenhos. Mas como isso é possível?

A resposta está em um campo chamado Aprendizado de Máquina. Assim como nós aprendemos com a prática, por exemplo, uma criança aprende a reconhecer um cachorro depois de ver vários cachorros diferentes, o computador também pode aprender a partir de exemplos.

Em vez de darmos uma lista de regras prontas, nós mostramos muitos exemplos para o computador e deixamos que ele descubra os padrões sozinho. Para o computador aprender, ele precisa de dados. Mas o que seriam esses dados?

Imagine um homem chamado João. Todos os dias, ele tomava café no mesmo copo de vidro. Ele queria saber se o café estava quente ou frio, mas sem encostar no copo (porque já havia se queimado algumas vezes). Então, ele pensou em uma ideia: colocar o copo sobre uma folha de papel.

Quando a marca era bem forte e escura, ele sabia que o café estava bem quente.

Quando a marca era fraquinha e mais clara, o café já estava esfriando.

E quando quase não tinha marca, o café estava frio.



Dia	Circunferência da marca (cm)	Força da marca (intensidade)	Copo
1	8,5 cm	Forte	Cheio
2	6,0 cm	Fraca (falhando)	Vazio
6	6,0 cm	Normal	Meio Cheio

Com o tempo, João foi observando essas marcas todos os dias e anotando:

Marca escura = café quente	Marca clara = café morno	Sem marca = café frio

O João não tinha uma "regra escrita" para medir a temperatura. Ele usava as marcas para entender a situação e escrevia os valores em um guardanapo em formato de tabela, como na figura. O que isso tem a ver com dados? Essas anotações do João são dados. Cada vez que ele media a circunferência e a força da marca, ele estava registrando informações.

Portanto, dados são informações que coletamos do mundo real e colocamos em uma tabela para entender melhor o que está acontecendo.

É exatamente aí que entra o Aprendizado de Máquina, em vez de programarmos todas as regras, deixamos o computador olhar para os dados (como as tabelas do café do João) e descobrir sozinho os padrões. Assim como aprendemos a reconhecer um cachorro depois de ver muitos cachorros diferentes, o computador aprende a reconhecer situações olhando para muitos exemplos nos dados.

ATIVIDADE DE PESQUISA E FIXAÇÃO

Nesta atividade, os alunos vão observar e registrar informações simples do seu dia a dia, colocando em tabelas. Assim, eles vão perceber que dados são informações que coletamos do mundo real.

- Divida os alunos em pequenos grupos
- Cada grupo deve escolher um objeto ou situação do cotidiano para observar (ex.: temperatura da sala, quantas vezes alguém ri durante a aula, músicas favoritas dos colegas, cor das mochilas, etc.)
- Eles devem coletar os dados (observando ou perguntando)
- Organizar em tabela
- Apresentar a conclusão para os colegas
- No final, cada grupo compartilha a tabela e discute: **"O que esses dados podem nos dizer?"**

DESAFIO/COMPARTILHAR

Nesta dinâmica, os alunos vão organizar e agrupar dados coletados em diferentes categorias, criando seus próprios critérios. Depois, vão compartilhar com os colegas, entendendo que há várias formas de olhar para os mesmos dados.

Modelos de exemplos que o re-aplicador pode dar para cada grupo

Quantos alunos usam o celular durante a aula (ou deixam o celular sobre a mesa)?
Observação direta + contagem.
Qual é a distribuição de alturas na sala?
Medir (em palmos ou aproximado) e agrupar em faixas: baixo, médio, alto.
Quantos alunos vieram de ônibus, a pé, de bicicleta ou de moto hoje?
Perguntar e organizar.
Qual é o estilo musical mais popular na sala?
Coletar preferências e agrupar por gênero musical.

Módulo 6

PRATICANDO APRENDIZADO DE MÁQUINA

Tema: Aprendendo Aprendizado de Máquina e suas aplicações

Duração: 2h

Material necessários: Notebooks, projetor, Internet, caderno e lápis

Opcional - Dinâmica 1: Aprendizado Não Supervisionado

- Cartões ou post-its com informações diferentes (símbolos ou combinações de características)
- Formas (círculo, triângulo, quadrado)
- Cores (vermelho, azul, verde)
- Números (1, 2, 3 desenhados)

Opcional - Dinâmica 2: Aprendizado por Reforço

- 3 objetos diferentes para esconder na sala
- Vendas (para tapar os olhos dos agentes)

Objetivo: Explicar de forma simples os conceitos básicos de Aprendizado de Máquina e os principais tipos de aprendizado, mostrando como a IA "aprende" com os dados.

CONTEXTUALIZAÇÃO/ENGAJAR

Existem diferentes formas de ensinar um computador a aprender com os dados. Na Inteligência Artificial, essas formas de aprender são chamadas de tipos de aprendizado de máquina:

Aprendizado Supervisionado - quando alguém ensina

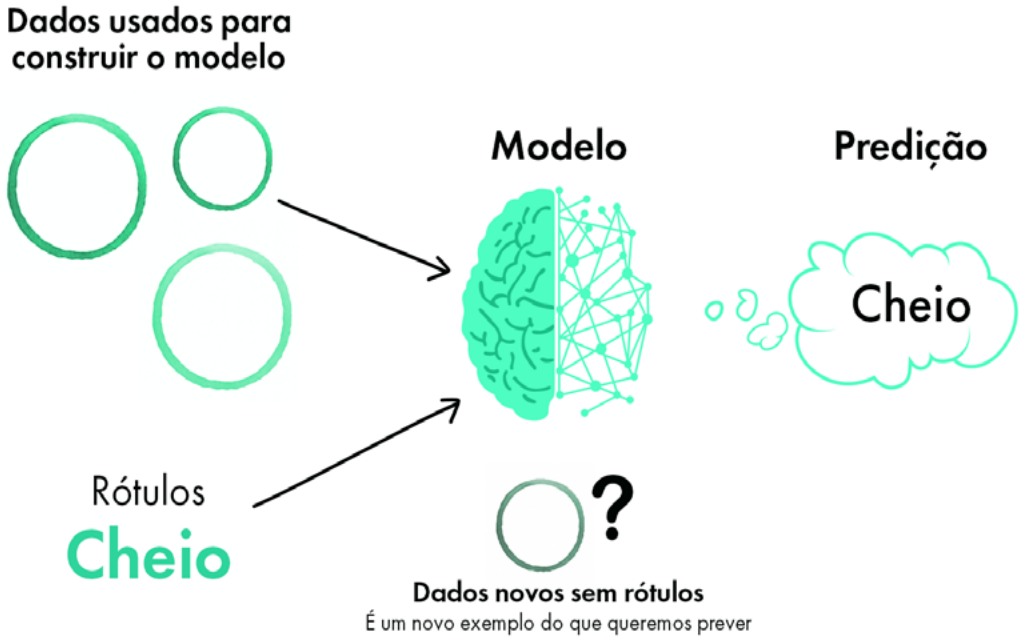
Imagine que o João mostra para o computador a tabela dele, mas além das medidas (circunferência e intensidade), ele também escreve a resposta certa: **"copo cheio"**, **"meio cheio"**, **"vazio"**.

O computador recebe exemplos com a resposta já pronta.

Ele aprende a associar:

Circunferência grande + marca forte = Cheio

Circunferência pequena + marca fraca = Vazio



É como se fosse um professor corrigindo a lição de casa: o aluno aprende porque sabe se a resposta estava certa ou errada.

Aprendizado Não Supervisionado - quando ninguém dá a resposta

Agora, imagine que o João entrega para o computador apenas as medidas, mas sem dizer se o copo estava cheio ou vazio.

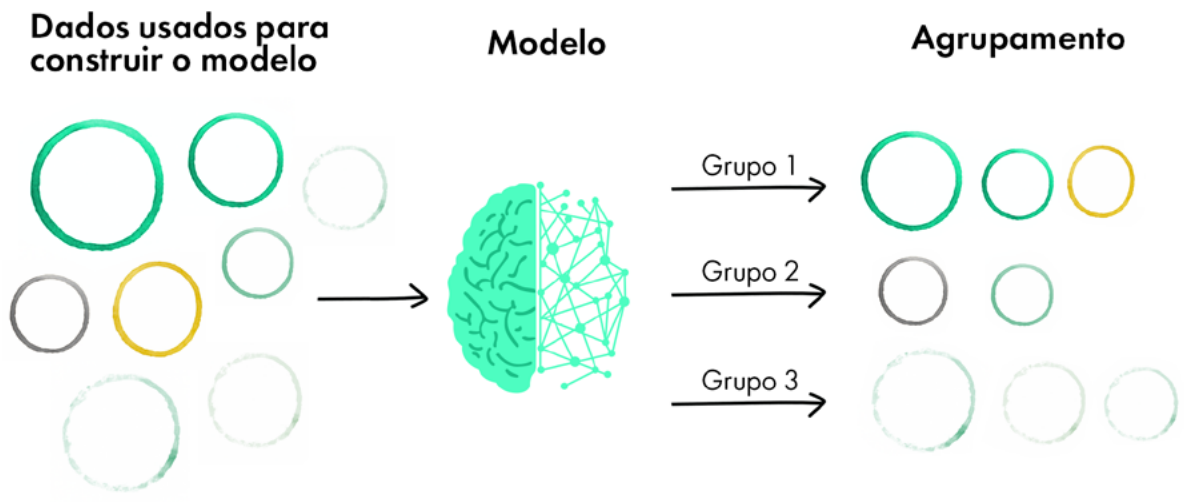
O computador fica com os números e tenta organizar os dados em grupos por conta própria.

Ele pode descobrir que existem dois ou três padrões:

Grupo A = marcas grandes e fortes (provavelmente copo cheio)

Grupo B = marcas médias (meio cheio)

Grupo C = marcas pequenas e fracas (copo vazio)



Esse é o agrupamento: o computador descobre "grupos" de dados parecidos, sem ninguém dar a resposta.

Aprendizado por Reforço - quando aprende tentando e errando

Imagine que o João faz um joguinho com o computador:

O computador olha a marca e adivinha se o café está cheio ou vazio.

Se acertar, ganha um ponto (recompensa).

Se errar, perde um ponto.

Com o tempo, o computador vai tentando, errando, e aprendendo qual é a melhor decisão para ganhar mais pontos.



É como quando você joga videogame: no começo erra muito, mas depois de várias tentativas aprende a jogar melhor.

ATIVIDADE DE PESQUISA E FIXAÇÃO

Agora que entendemos como os computadores aprendem, vamos fazer de conta que nós somos o computador.

Dinâmica 1

Aprendizado não supervisionado: Fazer os alunos descobrir sozinhos padrões escondidos e criar grupos (clusters), sem que ninguém dê a resposta pronta.

EXEMPLO DE CARTÕES:



COMO JOGAR

1. Distribua os cartões entre os alunos (cada aluno deve ter um desses cartões).
2. Não explique o que significa. Só diga: *"Vocês precisam se organizar em grupos de acordo com semelhanças nos cartões."*
3. Deixe eles discutirem como se organizar. Alguns vão agrupar por cor, outros por forma, outros por número e outros por imagens.
4. Diga que não há resposta certa única. O computador também pode agrupar de jeitos diferentes dependendo de como "vê" os dados.
5. Cada grupo explica qual foi a lógica que usou para se organizar e o tipo de aprendizado que eles acreditam que estão trabalhando.
6. A turma vota em qual agrupamento parece mais "inteligente".
7. Explique: *"Assim como vocês criaram critérios diferentes, os algoritmos de aprendizado não supervisionado também testam vários jeitos de organizar os dados até achar o que faz mais sentido."*

Dinâmica 2

Aprendizado Por reforço: Fazer os alunos vivenciarem como um computador aprende tentando, errando e recebendo recompensas. Em grupos, cada agente precisa encontrar um objeto escondido na sala, mas só pode se mover com comandos básicos ("direita, esquerda, frente, trás"). A cada passo certo recebe pontos positivos, a cada passo errado perde pontos. Assim, a turma percebe que, tal como em um videogame, o computador aprende qual caminho funciona melhor a partir de tentativas e do feedback recebido.

COMO JOGAR

1. Divida a turma em 3 grupos (Grupo A, Grupo B e Grupo C).
2. Cada grupo escolhe um Agente IA (a pessoa que vai procurar o objeto).
3. O Agente sai da sala. Enquanto isso, o re-aplicador ou grupo adversário esconde o objeto em um ponto da sala.

4. Quando o Agente volta, ele fica de olhos vendados ou de costas inicialmente.

5. O restante do grupo só pode dar comandos simples:
   - **a.** "Frente"
   - **b.** "Trás"
   - **c.** "Esquerda"
   - **d.** "Direita"

6. A cada movimento:
   - **a.** Se o agente se aproximar → ganha +1 ponto.
   - **b.** "Trás"

7. O jogo termina quando o objeto é encontrado.

8. **Competição entre grupos:** cada grupo tem 1 minuto para ajudar seu agente a encontrar o objeto. Ganha quem chegar mais rápido ou com mais pontos positivos.

9. Depois de cada rodada, os agentes trocam de grupo, mostrando que eles podem aprender mais rápido se já passaram pela experiência antes (reforço acumulado).

Observação

Deve ser feito por um grupo de cada vez e não todos de uma vez

DESAFIO/COMPARTILHAR

O agente = o computador

O grupo que dá instruções = o ambiente que fornece feedback.

Os comandos simples = ações possíveis do computador.

Os pontos ganhos ou perdidos = recompensas que guiam o aprendizado.

Assim como no jogo, os computadores aprendem por tentativa e erro, recebendo recompensas ou penalidades até descobrir a melhor estratégia.

Módulo 7

PRIMEIROS CONTATOS - UMA BREVE
INTRODUÇÃO A PYTHON E GOOGLE COLAB

Tema: Primeiros passos usando a base para codificar uma IA

Duração: 1h30

Material necessários: Notebooks, projetor, Conexão à Internet, acesso ao Google Colab.

Objetivo: Dominar o uso do Google Colab e os comandos básicos da linguagem Python, estabelecendo a base para a implementação de algoritmos de IA.

CONTEXTUALIZAÇÃO/ENGAJAR

Para desenvolver um modelo de IA eficaz, é fundamental entender que o aprendizado da máquina é impulsionado por dois elementos principais: dados e algoritmos.

Dados (Input)

Representam o conjunto de informações que o modelo utilizará para identificar padrões e realizar o treinamento. No contexto de Machine Learning, a qualidade e a quantidade dos dados são determinantes para a performance do modelo.

Bibliotecas (Frameworks)

São coleções de funções e classes pré-escritas que simplificam o processo de codificação, permitindo que o desenvolvedor se concentre na lógica do modelo. Utilizaremos a biblioteca Scikit-learn, uma ferramenta robusta e amplamente adotada para Machine Learning.

Programador (Desenvolvedor)

É o agente que integra os dados e as bibliotecas, definindo a arquitetura e os parâmetros do modelo através da linguagem Python.

PREPARAÇÃO DO AMBIENTE: GOOGLE COLAB

O Google Colaboratory (Colab) é um ambiente de notebook Jupyter hospedado na nuvem que permite a execução de código Python diretamente no navegador. Sua principal vantagem é a configuração zero, ideal para prototipagem e aprendizado em Machine Learning, pois já inclui as principais bibliotecas de IA pré-instaladas.

PROCEDIMENTO DE ACESSO

Acesse o Google Colab: https://colab.research.google.com/

Crie um novo notebook.

O ambiente está pronto para a execução de código Python.

O AMBIENTE DE DESENVOLVIMENTO: GOOGLE COLAB

O Colab utiliza o conceito de células. Você pode alternar entre células de Texto (para documentação) e células de Código (para execução de Python).

O AMBIENTE DE DESENVOLVIMENTO: GOOGLE COLAB

O Colab utiliza o conceito de células. Você pode alternar entre células de Texto (para documentação) e células de Código (para execução de Python).

Para executar o código em uma célula, clique no ícone "Play" ou utilize o atalho Shift + Enter.

1. Variáveis e Tipos de Dados

Variáveis são containers nomeados utilizados para armazenar valores na memória.

Exemplo:

Python

```python
# Atribuição de variáveis
nome_projeto = "Classificador Iris"  # String (texto)
versao = 1.0                         # Float (número decimal)
max_iteracoes = 100                  # Integer (número inteiro)

# Exibição de valores
print(f"Projeto: {nome_projeto}")
print(f"Versão: {versao}")
print(f"Iterações: {max_iteracoes}")
```

2. Operadores Aritméticos e Lógicos

O Python suporta operações matemáticas e lógicas essenciais para o processamento de dados.

Operadores Aritméticos:

Operação	Símbolo	Exemplo
Adição	+	5 + 3
Subtração	-	10 - 4
Multiplicação	*	2 * 6
Divisão	/	10 / 2
Exponenciação	**	2 ** 3

Python

```python
# Exemplo de operações aritméticas
resultado_soma = 15 + 7
resultado_multi = 4 * 5
print(f"Resultado da soma: {resultado_soma}")
print(f"Resultado da multiplicação: {resultado_multi}")
```

Operadores Lógicos (Booleanos): Resultam em True (Verdadeiro) ou False (Falso).

Operador	Símbolo	Significado	Exemplo
Igualdade	==	Verifica se os valores são idênticos	5 == 5 (True)
Diferença	!=	Verifica se os valores são distintos	5 != 6 (True)
Maior que	>	Verifica se o primeiro é maior	10 > 5 (True)
Menor que	<	Verifica se o primeiro é menor	5 < 10 (True)

Python

```python
# Exemplo de operações lógicas
print(f"10 é maior que 5? {10 > 5}")
print(f"7 é igual a 8? {7 == 8}")
```

3. Estruturas de Controle de Fluxo: if, elif, else

As estruturas condicionais permitem que o programa execute blocos de código específicos com base na avaliação de expressões lógicas.

```python
# Exemplo de if e else
score = 7.5

if score >= 7.0:
    print("Status: Aprovado.")
else:
    print("Status: Necessita de revisão.")
```

O elif (else if) permite encadear múltiplas condições.

```python
# Exemplo de if, elif e else
temperatura = 25

if temperatura > 30:
    print("Alerta: Temperatura elevada.")
elif temperatura >= 20:
    print("Condição: Temperatura ideal.")
else:
    print("Condição: Temperatura baixa.")
```

4. Estruturas de Repetição: for (Iteração)

O laço for é essencial para automatizar tarefas repetitivas, como o processamento de grandes volumes de dados em Machine Learning.

```python
# Exemplo de laço for
print("Contagem de 1 a 5:")
for i in range(1, 6): # Itera de 1 até 5
    print(f"Iteração {i}")
```

5. Estruturas de Dados: Listas e Tuplas

Estas estruturas são fundamentais para organizar e manipular coleções de dados.

Listas ([]): Coleções ordenadas e mutáveis (podem ser alteradas após a criação).

Tuplas (()): Coleções ordenadas e mutáveis (podem ser alteradas após a criação).

Estrutura	Símbolo	Mutabilidade	Exemplo
Lista	[]	Mutável	features = ["sepal_length", "sepal_width"]
Tupla	()	Imutável	dimensoes = (1920, 1080)

```python
# Exemplo de Lista
dataset_colunas = ["sepal_length", "sepal_width", "petal_length"]
print(f"Colunas do Dataset: {dataset_colunas}")
print(f"Primeiro elemento (índice 0): {dataset_colunas[0]}")

# Modificando um elemento da lista
dataset_colunas[2] = "petal_width"
print(f"Lista após modificação: {dataset_colunas}")

# Exemplo de Tupla
coordenadas_fixas = (40.71, -74.00)
print(f"Coordenadas: {coordenadas_fixas}")
# Tentar modificar coordenadas_fixas[0] resultaria em um erro (TypeError)
```

5. Funções: Modularização e Reutilização de Código

Funções são blocos de código nomeados que realizam uma tarefa específica. Elas são cruciais para a modularização (dividir um programa grande em partes menores) e a reutilização de código, tornando-o mais limpo e fácil de manter.

Sintaxe Básica: Uma função é definida pela palavra-chave def, seguida pelo nome da função, parênteses (que podem conter parâmetros de entrada) e dois pontos. O valor de saída é retornado pela palavra-chave return.

```python
# Exemplo de função sem parâmetros
def saudacao():
    print("Olá! Bem-vindo ao módulo de Python.")

saudacao() # Chamada da função

# Exemplo de função com parâmetros e retorno
def calcular_area_retangulo(largura, altura):
    area = largura * altura
    return area

# Chamada da função com argumentos
area_terreno = calcular_area_retangulo(10, 5)
print(f"A área calculada é: {area_terreno}")
```

Funções em IA:

Em Machine Learning, usamos funções para tarefas repetitivas como calcular métricas, pré-processar dados ou treinar modelos. A organização do código em funções é uma prática profissional indispensável.

Atividade Prática: Exercício de Programação em Python

Para consolidar o aprendizado das estruturas fundamentais (Variáveis, Listas, Condicionais, Loops e Funções), propomos o seguinte desafio:

Atividade de Pesquisa e Fixação

Crie um programa em Python que:

Cadastre alunos

- O usuário deve informar:
  - Nome do aluno
  - Idade
  - Lista de notas (mínimo 2, máximo 5 notas)

Armazene os alunos em uma lista

- Cada aluno deve ser armazenado como um dicionário no seguinte formato:

```python
{
    "nome": "Maria",
    "idade": 16,
    "notas": [7.5, 8.0, 9.2]
}
```

Após o cadastro, o programa deve:

- Exibir o total de alunos cadastrados
- Calcular a média de cada aluno
- Mostrar:
  - Quem tem a maior média
  - Quem tem a menor média
  - Quantos alunos têm média acima de 7

Regras do programa:

- O usuário pode cadastrar quantos alunos quiser
- O cadastro só deve parar quando o usuário digitar “sair”
- Se o usuário digitar uma nota inválida (ex: texto, número negativo ou acima de 10), o programa deve pedir novamente
- Organize o código usando Funções, por exemplo:
  - cadastrar_aluno()
  - calcular_media(notas)
  - exibir_relatorio(alunos)



Módulo 8

CONHECENDO O PRIMEIRO MODELO DE IA

Tema: Implementando sua primeira IA, prevendo os dados a partir de características da flor.

Duração: 2h

Material necessários: Notebooks, projetor, Conexão à Internet, acesso ao Google Colab.

Objetivo: Implementar um modelo de Machine Learning simples (Árvore de Decisão) para classificação utilizando o Iris Dataset e a biblioteca Scikit-learn.

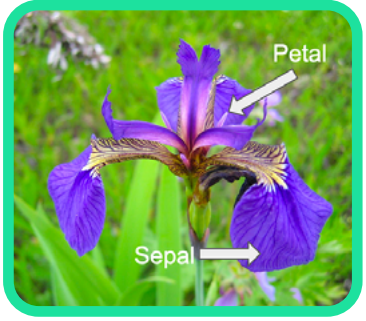
CONTEXTUALIZAÇÃO/ENGAJAR

1. O Dataset Clássico: Iris

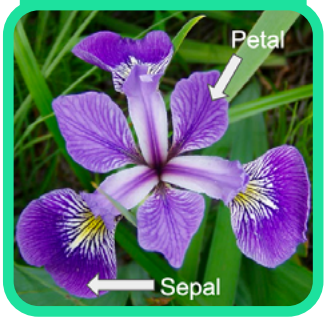
O Iris Dataset (Base de dados) é um dos conjuntos de dados mais famosos e utilizados na história do Machine Learning. Foi introduzido em 1936 pelo estatístico Ronald Fisher e é frequentemente usado como o primeiro estudo para classificar sistemas.

Estrutura do Dataset: O dataset é composto por 150 amostras de flores Iris, com 50 amostras para cada uma das três espécies:

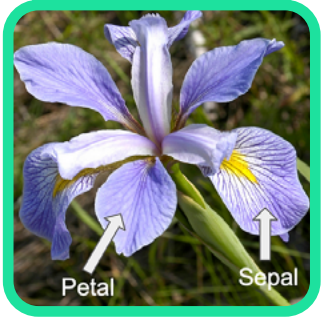
Iris setosa



Iris versicolor



Iris virginica



Features (Variáveis de Entrada): Para cada amostra, são fornecidas quatro medições em centímetros:

1. Sepal Length (Comprimento da Sépala)
2. Sepal Width (Largura da Sépala)
3. Petal Length (Comprimento da Pétala)
4. Petal Width (Largura da Pétala)

Target (Variável de Saída): A variável target é a espécie da flor, que o modelo de IA deve aprender a prever com base nas quatro características.

Importância: O Iris Dataset é ideal para iniciantes porque é pequeno, limpo (sem dados faltantes) e apresenta uma clara separação entre as classes, permitindo que modelos simples atinjam alta precisão.

2. A Biblioteca de Machine Learning: Scikit-learn

O Scikit-learn é a biblioteca de Machine Learning mais popular e acessível em Python. Ela fornece uma coleção consistente de ferramentas para tarefas de pré-processamento de dados, modelagem e avaliação.

Características Chave:

Consistência: Todos os modelos no Scikit-learn seguem uma interface padronizada

.fit() - para treinar uma algoritmo,

.predict() - para prever um algoritmo,

É otimizada para o aprendizado supervisionado e não supervisionado, sendo a ferramenta de escolha para a prototipagem e o desenvolvimento de modelos de ML tradicionais.

3. O Algoritmo de Classificação: Árvore de Decisão (Decision Tree)

A Árvore de Decisão é um algoritmo de Machine Learning supervisionado que pode ser usado tanto para classificação quanto para regressão. É um dos modelos mais intuitivos e interpretáveis.

Funcionamento Detalhado: O algoritmo constrói um modelo na forma de uma estrutura de árvore, onde:

Nós Internos: Representam um teste em uma feature (ex: "Comprimento da Pétala > 2.45 cm?").

Ramos: Representam o resultado desse teste (Sim ou Não).

Nós Folha: Representam a decisão final ou a classe (ex: Iris setosa).



Processo de Treinamento: Durante o treinamento (.fit()), o algoritmo busca a melhor característica e o melhor valor de corte para dividir o conjunto de dados de forma que a mistura de classes seja minimizada em cada subconjunto. Esse processo é repetido até que os nós folha contenham amostras de uma única classe ou até que um critério de parada seja atingido.

Módulo 9

MÃOS À OBRA! - PREVENDO OS DADOS A PARTIR DE CARACTERÍSTICAS DA FLOR

Tema: Classificação Supervisionada com Scikit-learn.

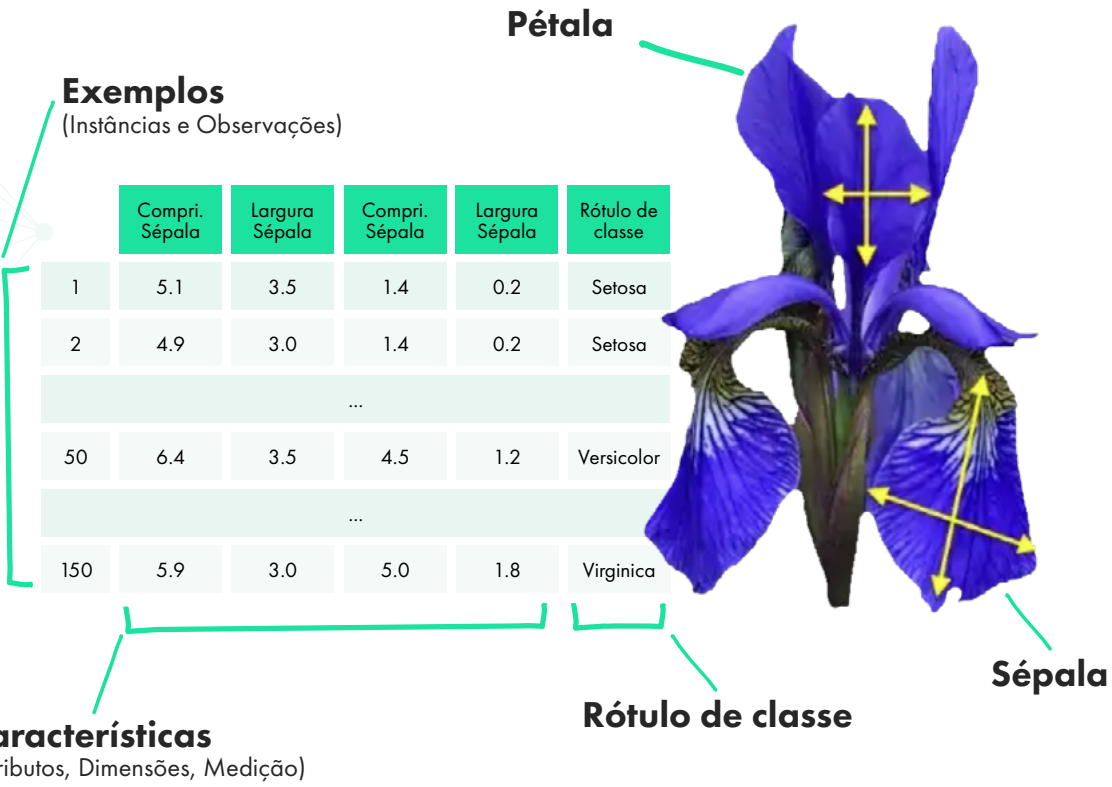
Duração: 2h

Material necessários: Notebooks, projetor, Conexão à Internet, acesso ao Google Colab.

Objetivo: Implementar um modelo de Machine Learning simples (Árvore de Decisão) para classificação utilizando o Iris Dataset e a biblioteca Scikit-learn.

CONTEXTUALIZAÇÃO/ENGAJAR

Utilizaremos o Iris Dataset, um conjunto de dados clássico em Machine Learning, que contém medições de quatro características (comprimento e largura da sépala, comprimento e largura da pétala) para três espécies de flores Iris: Setosa, Versicolor e Virginica.



Observação

Treinar um modelo de IA para classificar a espécie da flor com base nas quatro medições de entrada.

PROCEDIMENTO DE ACESSO:

Acesse o Google Colab: https://colab.research.google.com/

Crie um novo *notebook*.

O ambiente está pronto para a execução de código Python.

IMPLEMENTAÇÃO PRÁTICA

O processo de desenvolvimento do modelo segue etapas padronizadas:

Passo 1: Importação de Bibliotecas e Módulos

Importamos as ferramentas necessárias do Scikit-learn.

```python
# 1. Importação de módulos
from sklearn.datasets import load_iris
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
```

load_iris: Função para carregar o dataset.

DecisionTreeClassifier: O algoritmo de Machine Learning (Árvore de Decisão).

train_test_split: Função para dividir os dados em conjuntos de treino e teste.

accuracy_score: Métrica para avaliar a performance do modelo.

Passo 2: Carregamento e Preparação dos Dados

Os dados são carregados e separados em características (variáveis de entrada, X) e target (variável de saída, y).

```python
# 2. Carregamento dos dados
iris = load_iris()
X = iris.data  # Features (as 4 medições)
y = iris.target # Target (o tipo de flor: 0, 1 ou 2)
```

Passo 3: Divisão dos Dados (Treino e Teste)

É crucial separar uma porção dos dados para teste (geralmente 30%) para avaliar a capacidade de generalização do modelo.

```python
# 3. Divisão dos dados em conjuntos de treino e teste (70% treino, 30% teste)
X_treino, X_teste, y_treino, y_teste = train_test_split(X, y, test_size=0.3,
random_state=42)
```

Passo 4: Treinamento do Modelo

Instanciamos o classificador e o treinamos utilizando os dados de treino.

```python
# 4. Instanciação e Treinamento do Modelo
modelo_ia = DecisionTreeClassifier()
modelo_ia.fit(X_treino, y_treino) # O modelo aprende a mapear X_treino para y_treino
```

Passo 5: Avaliação da Performance

O modelo treinado é testado com os dados que ele nunca viu (X_teste), e sua precisão é calculada.

```python
# 5. Previsão e Avaliação
previsoes = modelo_ia.predict(X_teste)
precisao = accuracy_score(y_teste, previsoes)

print(f"A Acurácia do Modelo é de: {precisao * 100:.2f}%")
```

Resultado Esperado: Uma precisão tipicamente superior a 95%, indicando alta performance do modelo.

APLICAÇÃO: REALIZANDO UMA PREVISÃO

O modelo treinado pode ser usado para classificar novas amostras de dados.

```python
# Exemplo de nova amostra de dados (4 medições)
nova_flor = [[5.1, 3.5, 1.4, 0.2]] # Exemplo de Setosa
previsao_nova = modelo_ia.predict(nova_flor)

# Traduzindo o resultado numérico (0, 1 ou 2) para o nome da espécie
nomes_flores = iris.target_names
print(f"O modelo classifica a nova amostra como:
{nomes_flores[previsao_nova[0]]}")
```

Módulo 10

MÃOS À OBRA! - INTERPRETABILIDADE DO MODELO DE IA (ÁRVORE DE DECISÃO)

Tema: Visualizando o Processo de Decisão do Algoritmo.

Duração: 1h

Material necessários: Notebooks, projetor, Conexão à Internet, acesso ao Google Colab.

Objetivo: Entender e visualizar o funcionamento interno do algoritmo Decision Tree, um modelo de alta interpretabilidade.

CONTEXTUALIZAÇÃO/ENGAJAR

Como vimos anteriormente, o algoritmo Árvore de Decisão é um modelo de Machine Learning que simula um fluxograma de decisões. Ele aprende uma série de regras condicionais (perguntas "se/então") baseadas nas features para chegar a uma conclusão de classificação.

Exemplo de Regra:

Condição:	petal_length <= 2.45 cm?
Se Verdadeiro:	Classificar como Setosa.
Se Falso:	Avaliar a próxima condição.

ATIVIDADE: VISUALIZAÇÃO DA ÁRVORE

Podemos gerar uma representação gráfica da árvore de decisão que o modelo construiu.

Passo 1: Instalação de Dependência

Passo 2: Geração e Exibição da Árvore

Python

```
# Instalação da biblioteca de visualização
pip install graphviz
```

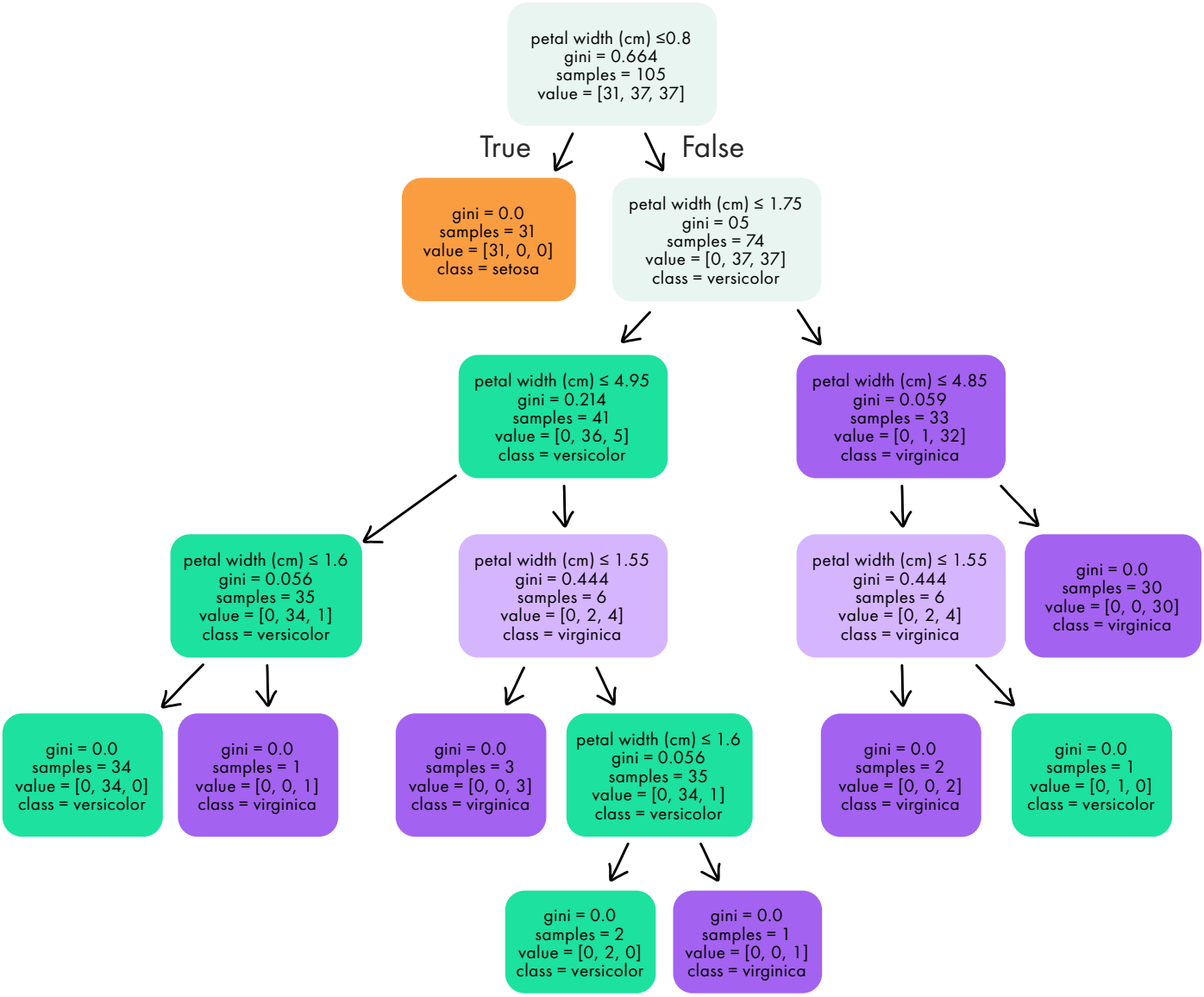
RESULTADO DO PROCESSAMENTO:

Python

```
# Geração da representação gráfica da Árvore de Decisão
import graphviz
from sklearn.tree import export_graphviz

# Exportar o modelo para o formato DOT
dot_data = export_graphviz(
    modelo_ia,
    out_file=None,
    feature_names=iris.feature_names,
    class_names=iris.target_names,
    filled=True,
    rounded=True,
    special_characters=True
)

# Renderizar e exibir o gráfico
graph = graphviz.Source(dot_data)
graph.render("arvore_de_decisao", view=True, format='png')
print("O arquivo 'arvore_de_decisao.png' foi gerado no seu ambiente do Colab,
representando o modelo.")
```



Análise da Árvore:

Nó Raiz:	O primeiro nó no topo da árvore. Ele representa a característica que o modelo considerou mais importante para iniciar a classificação.
Nós Internos:	Representam as regras condicionais.
Nós Folha:	Representam a classificação final (a espécie da flor).

Segue Abaixo o código do projeto anterior:

https://colab.research.google.com/drive/15dCp3y7YT8f16lxCayp8DVgcDysn11Vy?usp=sharing

Módulo 11

MÃOS À OBRA! - CLASSIFICANDO DÍGITOS COM O MNIST

Tema: Programando uma Inteligência Artificial com o MNIST

Duração: 2h

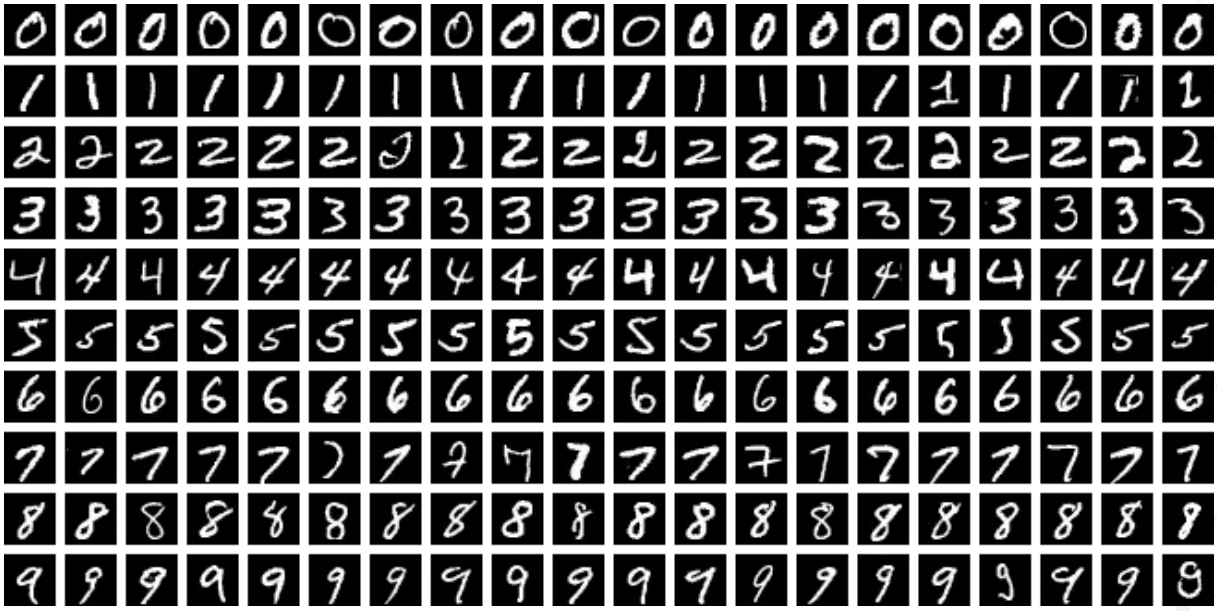
Material necessários: Notebooks, projetor, Internet, caderno e lápis

Objetivo: Apresentar como treinar um modelo simples de IA usando o famoso conjunto de imagens MNIST (números escritos à mão). Usaremos uma biblioteca pronta para usar como base.

CONTEXTUALIZAÇÃO/ENGAJAR

Passaremos agora para um novo paradigma de programação, onde o foco está no Aprendizado de Máquina a partir de dados. Iremos desenvolver um modelo para atuar como um 'robô leitor', capaz de interpretar e classificar números escritos à mão.

Para isso, usaremos o conjunto de dados MNIST, que é uma coleção vasta de imagens, consistindo em 60.000 exemplos de treinamento e 10.000 exemplos de teste de dígitos (0 a 9). Ao ser alimentada com milhares dessas imagens, a Inteligência Artificial (IA) aprende a mapear as características visuais (linhas, curvas, formas) de cada dígito, construindo a capacidade de reconhecer com precisão novos números manuscritos."



Pense assim:

Quando você era pequeno, aprendeu a reconhecer o número 3 vendo vários tipos de 3, alguns tortos, alguns redondos, alguns grandes, outros pequenos.

A IA faz a MESMA coisa! Ela olha muitos exemplos e percebe padrões. Nesse caso, com poucas linhas de código e uma biblioteca pronta, conseguimos treinar uma IA de verdade!

PASSO A PASSO PARA CRIAR ESSE NOVO EXPERIMENTO:

Procedimento de Acesso:

Acesse o Google Colab: https://colab.research.google.com/

Crie um novo *notebook*.

O ambiente está pronto para a execução de código Python.

IMPLEMENTAÇÃO PRÁTICA

O processo de desenvolvimento do modelo segue etapas padronizadas:

Passo 1: Importação de Bibliotecas e Módulos

Esta célula carrega todas as bibliotecas necessárias para o projeto.

```python
import tensorflow as tf
from tensorflow.keras.datasets import mnist
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Flatten
from tensorflow.keras.utils import to_categorical
import matplotlib.pyplot as plt
import numpy as np
```

tensorflow e keras:	O framework principal para construir e treinar a rede neural.
mnist:	A função para carregar o conjunto de dados.
Dense, Flatten, Sequential:	As ferramentas para definir a arquitetura do modelo.
matplotlib e numpy:	Usadas para visualização de dados e manipulação de arrays.

Passo 2: Carregamento e Pré-processamento dos Dados

Esta é a etapa de preparação dos dados, crucial para o bom funcionamento de qualquer modelo de Machine Learning.

Passo 3: Definição e Compilação do Modelo

Aqui, a arquitetura da rede neural é definida e configurada para o treinamento.

```python
modelo = Sequential([
    Flatten(input_shape=(28, 28)),
    Dense(128, activation='relu'),
    Dense(10, activation='softmax')
])

modelo.compile(optimizer='adam',
               loss='categorical_crossentropy',
               metrics=['accuracy'])
```

Sequential:

Define que o modelo será uma pilha linear de camadas.

Flatten:	Prepara a entrada (784 pixels).
Dense(128, ...):	A camada oculta.
Dense(10, ...):	A camada de saída, com 10 classes.
optimizer='adam':	O algoritmo que ajusta os pesos da rede durante o treinamento.
loss='categorical_crossentropy':	A função que mede o quão "errado" o modelo está. O objetivo do treinamento é minimizar essa perda.
metrics=['accuracy']:	A métrica que será monitorada (a porcentagem de acertos).

Passo 4: Treinamento do Modelo

Esta célula inicia o processo de aprendizado.

```python
# Exemplo de como a célula de treinamento deve ser
modelo.fit(x_train, y_train, epochs=10, batch_size=32, validation_data=(x_test,
y_test))
```

modelo.fit(...):	Inicia o treinamento.
x_train, y_train:	Os dados de treinamento.
epochs=10:	O número de vezes que o modelo verá todo o conjunto de dados.
validation_data:	Os dados de teste são usados para monitorar o desempenho do modelo em dados que ele nunca viu.

Passo 5: Teste Interativo (Opcional/Visual)

As células finais (que usam *ipycanvas* e *ipywidgets*) criam uma interface simples onde você pode desenhar um dígito.

Lógica:

O código captura o desenho, redimensiona-o para 28x28 pixels (o formato que o modelo espera), normaliza os pixels e, em seguida, usa modelo.predict() para obter a previsão.

Previsão:

O resultado é um array de 10 probabilidades. O código encontra o índice da maior probabilidade (que é o dígito previsto) e o exibe.

Ao rodar todas as células em ordem, você terá um modelo treinado e uma interface para testar a sua própria escrita.



Segue Abaixo o código do projeto anterior:

https://colab.research.google.com/drive/1fL_lfAUUbDRAFDZdPWrKy7N-z7mlIYry?usp=sharing

Módulo 12

MÃOS À OBRA! - CLASSIFICANDO DÍGITOS COM O MNIST

Tema: Programando uma Inteligência Artificial com o MNIST

Duração: 2h

Material necessários: Notebooks, projetor, Internet, caderno e lápis

CONTEXTUALIZAÇÃO/ENGAJAR

INTRODUÇÃO

O Módulo 12 representa a culminação prática dos conhecimentos adquiridos ao longo do curso, especialmente a integração entre os modelos de Aprendizado de Máquina (Machine Learning - ML) abordados nos Módulos 10 e 11 e a implementação em Sistemas Embarcados utilizando a plataforma Arduino.

O objetivo principal deste trabalho final é permitir que o aluno demonstre a capacidade de aplicar um modelo de ML treinado em um cenário real, onde a saída do modelo é utilizada para controlar um sistema físico. O aluno deverá escolher uma das duas propostas de atividade a seguir para desenvolver seu projeto.

PROPOSTA DE ATIVIDADE 1: RECONHECIMENTO DE DÍGITOS MANUSCRITOS E CONTROLE DE MÃO ROBÓTICA

Esta atividade foca na aplicação de um modelo de Classificação de Imagens (Módulo 10 ou 11) para o controle de um atuador mecânico.

Objetivo Específico

Desenvolver um sistema onde o reconhecimento de um dígito manuscrito (0 a 9) pelo modelo de IA resulte em um movimento pré definido de uma mão robótica controlada por servo motores via Arduino.

Componentes Necessários

Software:

Modelo de ML treinado para o reconhecimento de dígitos (ex: MNIST).

Hardware:

Placa Arduino (Uno, Mega ou similar), 5 a 6 Servo Motores (ex: SG90 ou MG996R), fonte de alimentação externa para os servos, e uma estrutura de mão robótica (impressa em 3D ou construída com materiais simples).

Requisitos de Implementação

1. **Classificação do Dígito:** O aluno deve implementar a interface para que o usuário insira um dígito (desenhando-o em uma tela ou utilizando uma imagem pré-existente). O modelo de IA deve classificar o dígito com alta precisão.

2. **Controle da Mão Robótica:** O código no Arduino deve interpretar o dígito recebido e acionar os servo motores para realizar um gesto correspondente.

3. **Exemplo de Mapeamento:**

   Dígito 1: Apenas o dedo indicador se estende.

   Dígito 5: Todos os dedos se fecham (mão fechada).

   Dígito 0: Todos os dedos se abrem (mão aberta).

4. **Demonstração:** O projeto deve ser demonstrado com o aluno desenhando o dígito e a mão robótica respondendo ao comando em tempo real.

PROPOSTA DE ATIVIDADE 2: CLASSIFICAÇÃO DE ESPÉCIES DE IRIS E SISTEMA DE IRRIGAÇÃO INTELIGENTE

Esta atividade foca na aplicação de um modelo de Classificação Multiclasse (Módulo 10 ou 11) para simular um sistema de automação agrícola.

Objetivo Específico

Desenvolver um sistema onde a classificação de uma planta (utilizando os dados do dataset Iris) pelo modelo de IA acione um LED de cor específica, simulando a identificação da espécie, e implemente um temporizador para simular a irrigação automatizada.

Componentes Necessários

Software:

Modelo de ML treinado para a classificação das três espécies do dataset Iris (Setosa, Versicolor, Virginica).

Hardware:

Placa Arduino, 3 LEDs (Verde, Vermelho, Azul), resistores, e um módulo de relé (opcional, para simular o acionamento de uma bomba de água).

Requisitos de Implementação

1. **Classificação da Espécie:** O aluno deve criar uma interface (via terminal ou interface gráfica simples) para que o usuário insira as quatro características da planta (comprimento e largura da sépala, comprimento e largura da pétala). O modelo de ML deve classificar a espécie.

2. **Mapeamento de Cores:** A classificação do modelo deve ser mapeada para uma cor de LED, informando visualmente a espécie identificada:

   - **Iris Setosa:** LED Verde (Planta A)
   - **Iris Versicolor:** LED Vermelho (Planta B)
   - **Iris Virginica:** LED Azul (Planta C)

   Controle de LEDs e Temporização: O Arduino deve acender o LED correspondente à espécie classificada.

3. **Solução de Problema (Irrigação):** O aluno deve implementar um temporizador onde o LED (ou o relé, simulando a bomba de água) permaneça aceso por um tempo determinado (ex: 10 segundos) e, em seguida, se apague.

4. **Justificativa:** Este temporizador simula uma solução para o problema de aguar a planta em um determinado tempo, garantindo que a irrigação seja controlada e específica para a espécie identificada, otimizando o uso de recursos hídricos.

ESTRUTURA DE ENTREGA DO PROJETO

ETAPA 1: APRESENTAÇÃO DA PROPOSTA

O aluno deve criar uma proposta inicial do projeto, contendo:

- Tema escolhido (ex: IA na saúde, IA na educação, IA para transporte etc.)
- Problema que deseja resolver
- Objetivo da solução
- Público-alvo

Essa etapa é para mostrar a ideia geral antes de construir o projeto completo.

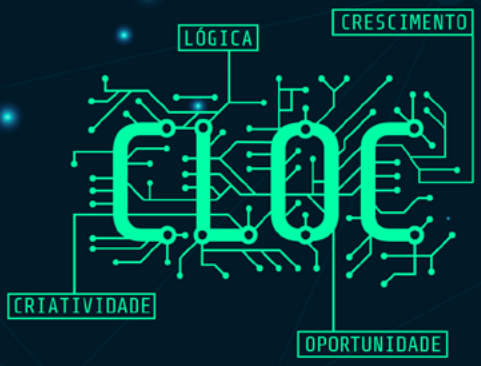
ETAPA 2: PROJETO FINAL

Depois que a proposta for aceita, o aluno deve desenvolver um projeto utilizando algum dos modelos de IA estudados.

EXPLICAÇÃO SOBRE IA (OBRIGATÓRIA NO PROJETO)

O projeto deve incluir uma parte teórica explicando:

- O que é Inteligência Artificial?
- Como funciona o modelo escolhido?
- Por que esse modelo é adequado para o projeto?
- Exemplos práticos de uso da IA no mundo real.



Realização:

The Human Project

Apoio:

CMDCA
CONSELHO MUNICIPAL DOS DIREITOS DA CRIANÇA E DO ADOLESCENTE DE SANTA LUZIA DO ITANHI

FMDCA

SECRETARIA DE ESTADO DO DESENVOLVIMENTO ECONÔMICO E DA CIÊNCIA E TECNOLOGIA

SERGIPE GOVERNO DO ESTADO

Parceria:

btg pactual

instituto localiza

Localiza&co