

CADERNOS Da Hora

TZILHA 1: A Camada Lógica

Volume 1

A LÓGICA DOS ALGORITMOS: COMO O
CÓDIGO MOLDA DECISÕES PÚBLICAS

INSTITUTO
da Hora

Ficha Técnica

Dados Internacionais de Catalogação na Publicação (CIP) (Câmara Brasileira do Livro, SP, Brasil)

A Lógica dos algoritmos [livro eletrônico] : como o código molda decisões públicas /[Nina da Hora ; organização Instituto da Hora]. -- Rio de Janeiro : Instituto da Hora, 2025. -- (Cadernos da hora. A camada lógica ; 1) PDF

ISBN 978-65-84362-29-1

1. Algoritmos 2. Dados - Estruturas (Ciência da computação) 3. Direito digital 4. Gestão pública 5. Governança 6. Lógica 7. Tecnologia I. Hora, Nina da. II. Instituto da Hora. III. Série.

25-325494.0

CDD-005.1

Índices para catálogo sistemático:

1. Algoritmos : Computadores : Programação :
Processamento de dados 005.1

Eliane de Freitas Leite - Bibliotecária - CRB 8/8415



Atribuição-SemDerivações-SemDerivados
CC BY-NC-ND

Volume 1

A LÓGICA DOS ALGORITMOS: COMO O CÓDIGO MOLDA DECISÕES PÚBLICAS

Neste Volume 1, mapeamos as bases técnicas que sustentam o debate democrático: desde a regra rígida (determinística) até a otimização de sistemas, ou seja, o ajuste matemático contínuo para que a máquina atinja metas de eficiência máxima, muitas vezes invisíveis para nós.

Para entender como essas engrenagens impactam direitos fundamentais, introduzimos o conceito de inferência. É aqui que o termo técnico ganha peso social: é o momento em que a máquina deixa de apenas processar dados e passa a concluir, prever e ditar regras sobre nossas vidas.

Porém, o universo da Inteligência Artificial vai muito além, o funcionamento detalhado das Redes Neurais, os perigos reproduzidos e as ferramentas para auditar essas "caixas pretas" serão o tema exclusivo do nosso próximo caderno: Cadernos da Hora – Volume 2: Inteligência Artificial e a Caixa Preta.

Boa Leitura!

Equipe Instituto da Hora

Sumário

Introdução

Por que explicar algoritmos é uma questão de democracia? 5

Capítulo 1

O que é um algoritmo? 7

Funções e estrutura de dados como base dos algoritmos 8

Estrutura de dados são anatomia do algoritmo 8

Estruturas de dados produzem representações 11

Capítulo 2

Nem Todo Algoritmo é Igual 13

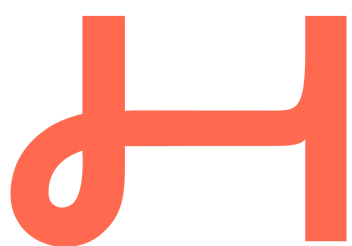
Algoritmos Determinísticos: A Arquitetura da Regra Rígida 15

Algoritmos de Otimização e Rede: A Lógica da Priorização 17

Aprendizado de Máquina: Da Inferência Matemática ao Produto de Prateleira 19

Referências

Bibliográficas 22



Introdução

Por que explicar algoritmos é uma questão de democracia?

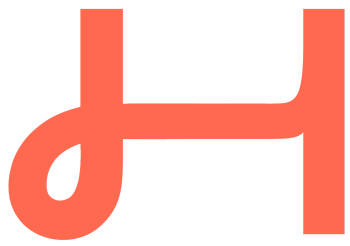
Sistemas algorítmicos já deixaram de ser apenas ferramentas de suporte para se tornarem verdadeiros árbitros da vida social. Hoje, códigos invisíveis decidem quem recebe crédito para abrir um negócio, quem passa por vigilância nas ruas, quais notícias chegam ao nosso conhecimento e até quem é considerado apto a uma vaga de emprego ou a um benefício social. No Brasil, essas decisões automatizadas ocorrem em um terreno marcado por desigualdades estruturais históricas: o racismo, o sexismo e os abismos econômicos que não desapareceram magicamente ao serem convertidos em dados. Pelo contrário, muitas vezes, a digitalização serve para esconder velhas injustiças sob uma nova camada de verniz técnico e aparentemente neutro.

Por isso, defender que a sociedade entenda como os algoritmos funcionam não é apenas uma exigência técnica, é uma questão central de governança, justiça e sobrevivência democrática. Quando um algoritmo decide, alguém delegou essa decisão. E delegar poder sem entender a lógica que o produz é abrir espaço para erros previsíveis e violências automatizadas. O risco aqui não é apenas o de uma falha pontual no sistema, mas o da escala e da invisibilidade. Diferente de um erro humano, que afeta casos individuais, um erro algorítmico ou um critério excludente inserido no código pode negar direitos a milhões de pessoas simultaneamente, sem que haja um rosto humano para ser cobrado ou um balcão onde se possa reclamar.

Essa ameaça transborda os limites da gestão pública e invade o cotidiano de toda a sociedade. Estamos falando de um cenário onde a eficiência matemática é frequentemente usada como justificativa para ignorar a complexidade da vida humana, punindo desproporcionalmente populações negras, indígenas, periféricas e LGBTQIA+. Se a regra é rígida demais, ela

exclui quem vive na exceção; se a regra é baseada apenas no passado, ela condena o futuro a repetir as mesmas discriminações de sempre. Sem transparência sobre como essas engrenagens giram, perdemos a capacidade de defesa e, em última instância, a própria soberania sobre os rumos do país.

Este primeiro volume dos **Cadernos da Hora** nasce dessa urgência. Ele foi desenhado para servir como uma bússola para tomadores de decisão, ativistas e cidadãos. Nas páginas a seguir, apresentamos de forma rigorosa, porém acessível, a anatomia dessas estruturas: como os dados traduzem pessoas em números, qual a diferença entre seguir uma regra e inferir um padrão, e onde exatamente se escondem os riscos que precisamos monitorar. Entender a arquitetura dessas decisões é o primeiro passo para garantir que a tecnologia sirva à redução das desigualdades, e não à sua automatização silenciosa.



Capítulo 1

O que é um algoritmo?

A noção de algoritmo ocupa um lugar central na ciência da computação e na governança digital contemporânea. Em termos formais, um algoritmo é um conjunto finito de instruções capaz de transformar uma entrada (*input*) em uma saída (*output*) seguindo regras bem definidas. Donald Knuth, uma das autoridades clássicas no campo, descreve algoritmos como procedimentos “precisos, finitos e efetivos” usados para resolver problemas computacionais de forma sistemática (Knuth, *The Art of Computer Programming*, 1968).

Do ponto de vista da computação, um algoritmo não é apenas uma sequência de passos, mas também uma estrutura formal composta por:

- um espaço de entrada (o universo de dados possíveis),
- um conjunto de operações permitidas (as instruções elementares da linguagem ou máquina),
- uma estratégia de controle (ordem em que as instruções são aplicadas),
- um estado interno, que pode incluir memória, variáveis ou estruturas organizadas,
- um critério de parada, que garante finitude,
- e um mapeamento determinístico ou probabilístico entre entrada e saída.

Essa visão mais completa permite entender por que algoritmos podem ser analisados matematicamente: cada operação tem um custo, cada estrutura de dados apresenta restrições e cada estratégia de controle influencia a eficiência e a precisão das decisões tomadas.

Funções e estrutura de dados como base dos algoritmos

A computação enxerga algoritmos como funções que operam sobre representações formais, conjuntos de dados organizados com precisão para que possam ser manipulados por uma máquina. Essa formulação permite analisar algoritmos matematicamente: cada operação tem um custo, cada estrutura apresenta um comportamento previsível e cada função possui um conjunto claro de entradas e saídas.

Formalmente, um algoritmo pode ser descrito como:

$$f: \mathcal{D} \rightarrow \mathcal{R}$$

Onde:

$\mathcal{D}$ é o domínio dos dados de entrada,

$\mathcal{R}$ é o conjunto de resultados possíveis,

f é a transformação computacional que associa cada dado a uma resposta.

Essa maneira de definir algoritmos está ligada à tradição da teoria da computação: se algo pode ser descrito como uma função computável, é possível analisá-lo quanto à sua corretude, eficiência, limites, impactos. Mas essa função, não opera no vazio, ela só existe por que o dado foi representado em uma estrutura formal adequada.

Estrutura de dados são anatomia do algoritmo

Em ciência da computação, uma estrutura de dados é uma forma de organizar informação na memória, de modo que operações específicas possam ser realizadas com eficiência.

A escolha da estrutura determina:

- a velocidade de acesso;
- o consumo de memória;
- a possibilidade de paralelização;
- a estabilidade da manipulação;
- e até quais operações são viáveis.

Por isso dizemos que não existem algoritmos fora de estruturas de dados.

Um algoritmo é uma coreografia matemática que dança dentro das restrições impostas pela estrutura de dados, e essa metáfora é tecnicamente precisa. As operações que um algoritmo pode executar dependem diretamente do modo como a informação está organizada na memória máquina.

A escolha da estrutura determina o desempenho. Se os dados estão armazenados em um vetor contínuo, por exemplo, o acesso é imediato, mas inserir algo novo no meio do vetor é custoso, pois exige deslocar toda a fila de informações. Se a informação está em um grafo, o tempo de busca por conexões varia conforme a densidade da rede (vértices e arestas). Já em uma árvore balanceada, a eficiência decorre da organização hierárquica, que divide o problema em subproblemas menores para acelerar a busca. É diferente de uma tabela hash, que funciona como um escaninho com endereços fixos: você vai direto ao ponto onde o dado está, sem perder tempo em procurar.

Cada estrutura, portanto, impõe seus próprios limites físicos e oportunidades lógicas.

Essa relação íntima entre algoritmo e estrutura significa que a eficiência não é uma propriedade abstrata: ela emerge da combinação entre o que o algoritmo faz e a disposição dos dados. O mesmo procedimento pode ser extremamente rápido em uma estrutura e quase impraticável em outra. Um algoritmo elegante em teoria pode ser lento na prática se a representação

escolhida não permitir que suas operações fundamentais sejam realizadas de forma eficiente.

Além disso, diferentes estruturas de dados afetam de maneiras muito distintas a forma como erros, distorções e vieses se propagam em um sistema computacional. Cada estrutura cria um tipo particular de “visão de mundo matemática”, e essa visão molda tanto o que o algoritmo consegue perceber quanto a forma como ele erra.

Quando usamos **vetores**, por exemplo, comprimimos muitas características em uma lista de números. É como fotografar uma pessoa apenas por uma sequência de medidas: altura, idade, CEP, renda aproximada. A fotografia numérica é simples e eficiente, mas não guarda nenhum elemento que não possa ser traduzido para números. O vetor não “vê” a pessoa; vê apenas aquilo que foi transformado em número. Se alguma característica importante ficou de fora, ela simplesmente deixa de existir para o algoritmo. E, quando erros acontecem, eles se espalham justamente por essa compressão: pequenas diferenças numéricas podem ser interpretadas como diferenças enormes ou irrelevantes dependendo da escala adotada.

Um exemplo do dia a dia, é o Cadastro Único do Governo Federal, onde cada família é representada por um vetor de aproximadamente 200 atributos numéricos ou categóricos. Se a composição familiar é complexa, digamos, três gerações morando no mesmo lote, mas em casas separadas, situação comum em periferias urbanas, o formulário exige uma simplificação: uma "família" precisa caber em campos predefinidos. Famílias indígenas com organizações territoriais específicas ou configurações familiares LGBTQIA + que não seguem o modelo hegemônico frequentemente não se enquadram bem no vetor. O que não cabe na estrutura é perdido antes mesmo de qualquer cálculo ser feito.

Em **árvores de decisão**, o algoritmo divide repetidamente o espaço de possibilidades em blocos cada vez menores, como quem corta um mapa em quadrantes sucessivos. Essa divisão pode amplificar atributos específicos: se o algoritmo decide primeiro separar por renda, depois por escolaridade e,

só então, por território, essas escolhas ficam gravadas na estrutura. Um atributo que aparece cedo na árvore ganha grande peso; um que aparece apenas nos ramos finais tem pouco impacto. Assim, a própria forma da árvore distribui vieses: ela destaca algumas diferenças e apaga outras.

Os **tensores**, usados em visão computacional e em redes neurais profundas, representam dados em múltiplas dimensões. Eles reorganizam padrões que os humanos não conseguem perceber diretamente, como correlações sutis entre tonalidades de pixel ou combinações de formas. O poder dos tensores está em revelar estruturas escondidas, mas esse mesmo poder pode gerar distorções se os dados utilizados forem desbalanceados. Uma leve diferença na iluminação dos rostos de pessoas negras, por exemplo, pode se tornar um padrão algorítmico que gera erros sistemáticos.

O ponto central é que a estrutura formal vetor, árvore, grafo, tensor determina não só o custo computacional, mas também **o que é computável, o que é visível, o que é ignorado e como o erro se distribui**. Cada estrutura oferece um tipo de lente, e toda lente enfatiza algumas coisas enquanto desfoca outras. Um algoritmo não enxerga o mundo inteiro: enxerga apenas o que a estrutura de dados permite.

Estruturas de dados produzem representações

Toda representação computacional é, necessariamente, uma simplificação. Ao converter uma pessoa em um vetor, por exemplo, perde-se tudo o que não cabe na forma numérica: história, contexto, experiência, território, nuances culturais, vínculos sociais. Ficam apenas atributos discretos :idade, renda, localização, escolaridade , escolhidos não por serem essenciais, mas por serem facilmente formalizáveis. E essa seleção não é inocente: ela define, desde o início, qual versão da realidade o algoritmo poderá operar.

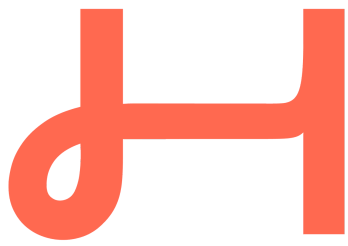
É por isso que estruturas de dados não são apenas escolhas técnicas, mas também políticas. Elas definem quais aspectos do mundo serão tornados computáveis e quais serão descartados no processo. A vida social não

chega inteira ao algoritmo: passa por uma filtragem que seleciona atributos mensuráveis e rejeita tudo o que não se adequa ao formato. Essa filtragem impõe um tipo de ordem ao mundo, uma ordem que reflete prioridades institucionais, convenções históricas e limitações tecnológicas. Representar é sempre decidir; decidir é sempre excluir; excluir é sempre produzir uma visão específica do real.

Quando uma política pública ou um sistema empresarial se baseia em algoritmos, a estrutura de dados atua como mediadora das decisões. Quem é representado de forma pobre tende a sofrer mais erros; quem é representado de forma enviesada terá seus traços amplificados pela máquina; quem não é representado desaparece da possibilidade de ser atendido ou protegido. Assim, antes que qualquer modelo seja treinado, antes mesmo de existir uma linha de código sofisticada, a semente do risco já está plantada na escolha da estrutura que representa as pessoas e os fenômenos.

É por isso que afirmamos: estruturas de dados produzem representações e representações produzem mundos computáveis. O que um algoritmo pode perceber, julgar, calcular ou ignorar depende, fundamentalmente, da forma como a informação foi estruturada. O mundo que a máquina enxerga não é o mundo real, e sim uma versão reduzida, filtrada e normalizada por essas estruturas. E toda visão reduzida produz também um mundo reduzido: um mundo de categorias rígidas, diferenças comprimidas e relações simplificadas.

Por isso, antes mesmo que qualquer modelo seja treinado, a forma escolhida para representar os dados já influencia profundamente os resultados. A estrutura não é um detalhe técnico: é o alicerce sobre o qual o sistema interpreta a realidade e, portanto, também o alicerce no qual desigualdades, ruídos e vieses podem ser silenciosamente inscritos.



Capítulo 2

Nem Todo Algoritmo é Igual

Se a estrutura de dados é a anatomia, a forma como o mundo é organizado na memória da máquina, o algoritmo é a fisiologia: é o sistema em movimento, tomando decisões e executando ações.

No livro *Algoritmos para Viver* ("Algorithms to Live By"), Brian Christian e Tom Griffiths nos lembram que a palavra "algoritmo" não pertence apenas às máquinas. Seres humanos usam pensamento algorítmico o tempo todo para lidar com a escassez. Quando decidimos organizar uma estante de livros, estamos executando um algoritmo de ordenação (*sorting*). Quando escolhemos entre ir ao restaurante favorito ou arriscar um novo, estamos resolvendo um dilema computacional clássico entre explorar o novo ou aproveitar o conhecido (*explore vs. exploit*).

Um algoritmo, portanto, nada mais é do que uma estratégia definida para resolver um problema usando recursos limitados. Na computação, esses recursos são tempo de processamento e espaço de memória. Na sociedade, os recursos são vagas em hospitais, crédito bancário, atenção nas redes sociais ou liberdade de ir e vir.

O problema surge quando essas estratégias saem da esfera individual e ganham escala industrial. Uma coisa é eu usar um critério pessoal para decidir quais livros ficam na altura dos olhos na minha estante. Outra coisa, muito diferente, é uma plataforma digital usar um critério opaco para decidir quais notícias ficam "na altura dos olhos" de 200 milhões de brasileiros.

Nesse sentido, algoritmos não são ferramentas neutras de cálculo; eles são **agentes de priorização**.

- Todo algoritmo de **classificação** cria hierarquias: define quem está em cima e quem está embaixo.
- Todo algoritmo de **filtragem** define fronteiras: decide o que passa e o que é barrado.
- Todo algoritmo de **otimização** define valores: escolhe qual métrica (lucro, velocidade, justiça) será maximizada em detrimento das outras.

Portanto, antes de perguntar "qual tecnologia foi usada?" (se é IA, blockchain ou Excel), a governança precisa perguntar: **"Qual é a estratégia de decisão que foi codificada aqui?"**.

Para responder a isso, precisamos parar de tratar todos os algoritmos como uma coisa só. Um algoritmo que verifica se um CPF é válido opera com uma lógica completamente diferente de um algoritmo que decide se um rosto é suspeito. Para fins de auditoria, regulação e defesa de direitos, podemos dividir os sistemas que impactam a sociedade em três grandes famílias lógicas.

Entender a qual família um algoritmo pertence é o primeiro passo para saber quais perguntas devemos fazer a ele.

Algoritmos Determinísticos: A Arquitetura da Regra Rígida

Antes de falarmos sobre inteligências artificiais que "aprendem" ou "alucinam", precisamos compreender a infraestrutura que sustenta a maior parte da digitalização do Estado e dos serviços bancários: **o algoritmo determinístico**.

Diferente do aprendizado de máquina, que opera no terreno da probabilidade ("talvez isto seja uma fraude"), o algoritmo determinístico opera no terreno da certeza absoluta. Ele é a tradução literal da vontade do programador (ou do legislador) em uma sequência de instruções imutáveis. Ele não improvisa. Ele obedece.

A ideia de que decisões sociais poderiam ser transformadas em cálculo matemático é antiga. No século XVII, o filósofo Gottfried Wilhelm Leibniz sonhava com uma *Mathesis Universalis* — uma linguagem universal onde, diante de qualquer conflito, dois filósofos não precisariam discutir. Bastaria sentar à mesa e dizer: "*Calculemos!*".

O algoritmo determinístico é a tentativa técnica de realizar esse sonho. Para isso, ele utiliza a **Lógica Booleana**, desenvolvida pelo matemático George Boole no século XIX. Nessa lógica, toda a complexidade do universo deve ser reduzida a apenas dois estados possíveis:

- **1 (Verdadeiro / True)**
- **0 (Falso / False)**

Onde mora o problema: A vida real raramente é binária. A realidade social é "nebulosa" (fuzzy), cheia de zonas cinzentas. Imagine a pergunta: "Você tem residência fixa?"

- Para um ser humano, a resposta pode ser: "Estou dormindo na casa da minha tia há 3 dias, mas semana passada estava na rua".

- Para o algoritmo determinístico do banco de dados, essa nuance é impossível. O campo no formulário aceita apenas **SIM** ou **NÃO**.

No momento em que a vida real (complexa) é forçada a entrar na estrutura lógica (binária), ocorre uma perda massiva de contexto. A "injustiça" aqui não é um erro de cálculo, é um erro de **tradução**. O que não cabe no binário, deixa de existir para o sistema.

O Código como Lei (Lawrence Lessig)

Se a lógica booleana define como o sistema "pensa", o conceito de "Code is Law" (O Código é a Lei), cunhado pelo jurista Lawrence Lessig, define como ele **governa**.

No mundo físico, uma lei é uma ordem que pode ser desobedecida. Se a lei diz "proibido pisar na grama", você pode pisar (e arcar com a multa depois). Há espaço para a desobediência civil ou para a exceção de emergência.

No mundo digital determinístico, o código não regula a conduta; ele a **arquiteta**. Se o algoritmo de um formulário de solicitação de auxílio foi programado para exigir um CPF ativo, e o seu está pendente, o botão "Enviar" simplesmente não funciona. Não há como "conversar" com o algoritmo, não há como explicar a exceção. A regra torna-se uma barreira física intransponível. O algoritmo determinístico transforma a burocracia em uma "arquitetura de controle" perfeita, onde o comportamento desviante torna-se tecnicamente impossível.

A Triage Digital e a Desumanização (Virginia Eubanks)

Quando aplicados a políticas públicas, esses sistemas criam o que a pesquisadora Virginia Eubanks, em sua obra *Automating Inequality*, chama de "Digital Poorhouse" (A Casa dos Pobres Digital).

Ao estudar sistemas de bem-estar social nos EUA, Eubanks notou que algoritmos determinísticos eram usados para realizar uma "triagem digital". A promessa era eficiência e combate à fraude. A realidade foi a punição da

vulnerabilidade. Como o sistema não aceita nuances, pessoas em situações atípicas — idosos que não conseguiam assinar digitalmente, famílias com composições não-nucleares — eram automaticamente "cortadas" pelo sistema.

O efeito mais perverso desse modelo é o desengajamento moral. Quando um assistente social nega um benefício cara a cara, ele sente o peso da decisão. Quando o algoritmo nega baseada em uma regra **IF/THEN** rígida, o funcionário diz: "*Sinto muito, o sistema não autorizou*". A responsabilidade dilui-se na máquina. O algoritmo determinístico serve como um escudo que protege a instituição da culpa de negar direitos.

Algoritmos de Otimização e Rede: A Lógica da Priorização

Se o algoritmo determinístico atua como o burocrata que carimba papéis na entrada de um sistema, o algoritmo de otimização age como um gerente de tráfego invisível, decidindo quem circula rápido e quem fica parado no engarrafamento. Nesta segunda família de algoritmos, saímos da lógica binária da permissão onde a resposta é sim ou não e entramos na lógica hierárquica da alocação e da priorização. Estes são os motores que movem as grandes plataformas digitais contemporâneas, como aplicativos de transporte, entregas e redes sociais. Eles não foram desenhados para julgar a verdade ou aplicar uma lei fixa, mas para organizar o caos de milhões de conexões simultâneas.

Para compreender como essas decisões são tomadas, é preciso entender que esses algoritmos não enxergam o mundo como uma lista de formulários, mas como um **Grafo**. Na ciência da computação, um grafo é uma estrutura composta por "nós" (que podem ser pessoas, roteadores, restaurantes) conectados por "arestas" (que representam amizades, ruas ou transações). O trabalho do algoritmo, portanto, não é verificar uma regra isolada, mas navegar por essa teia gigantesca para encontrar o caminho "mais eficiente" ou a conexão "mais forte". É aqui que a técnica encontra a

política: a definição do que significa ser "eficiente" nunca é neutra.

O coração político desses sistemas reside em uma equação matemática chamada **Função Objetivo** (ou função de custo). Todo algoritmo de otimização é programado com uma meta clara: maximizar ou minimizar uma variável específica. Um aplicativo de rotas pode ter a função de minimizar o tempo de viagem; uma rede social, a de maximizar o tempo de retenção do usuário na tela; um aplicativo de entregas, a de maximizar o número de corridas por hora. Como alerta a matemática Cathy O'Neil em *Weapons of Math Destruction*, algoritmos são, essencialmente, opiniões embutidas em código. Quando uma plataforma define sua função objetivo, ela está fazendo uma escolha moral e comercial sobre o que tem valor e o que é descartável.

O impacto social dessa lógica manifesta-se brutalmente na chamada "gestão algorítmica" do trabalho. Em plataformas da gig economy, o algoritmo substitui o chefe humano. Ele não dá ordens diretas, mas manipula as variáveis do grafo para induzir comportamentos. Se a função objetivo do sistema é a eficiência extrema da entrega, o algoritmo pode punir um entregador que recusa rotas perigosas não com uma demissão formal, mas com a invisibilidade. O trabalhador sofre um rebaixamento silencioso em sua pontuação de prioridade, parando de receber chamadas lucrativas. Alex Rosenblat, em *Uberland*, descreve isso como um gerenciamento por vigilância contínua, onde a otimização matemática da empresa é transferida como precarização para o trabalhador.

A mesma lógica se aplica ao consumo de informação. Na era da abundância de dados, a censura moderna raramente opera apagando conteúdos; ela opera através do ranqueamento. O algoritmo decide o que aparece na primeira tela do seu celular e o que é enterrado na décima página de resultados. Estar matematicamente posicionado na periferia do grafo significa, para todos os efeitos práticos, deixar de existir no debate público. Isso afeta de forma desproporcional vozes negras, periféricas e dissidentes: se o conteúdo produzido por esses grupos não gera o tipo de engajamento imediato que a função objetivo prioriza muitas vezes calibrada para um

usuário médio padrão, o sistema os empurra para o esquecimento. A invisibilidade aqui não é um acidente técnico, é o resultado direto de uma otimização que escolheu não incluir a diversidade como métrica de sucesso.

Aprendizado de Máquina: Da Inferência Matemática ao Produto de Prateleira

Chegamos, finalmente, à terceira e mais complexa família de algoritmos, que representa uma ruptura tectônica na história da computação: o **Aprendizado de Máquina** (Machine Learning). Para compreender a dimensão do risco que enfrentamos hoje, é preciso entender que houve uma mudança fundamental na lógica de programação. Se nos algoritmos determinísticos (vistos na Seção Anterior) o ser humano detinha o controle absoluto ao escrever cada regra explícita como "se a pessoa tiver menos de 18 anos, bloqueie o acesso", no Aprendizado de Máquina ocorre uma inversão de autoria. O ser humano deixa de escrever a regra e passa a fornecer os dados, delegando à máquina a tarefa de inferir os padrões ocultos.

Essa "lógica de inferência" é a capacidade de encontrar uma estrutura matemática dentro de um amontoado de dados caóticos. Pense na tarefa de ensinar um computador a reconhecer um gato. No modelo antigo, um programador precisaria descrever o animal em código: "tem bigodes, orelhas pontudas e cauda". No modelo de Aprendizado de Máquina, o programador apenas alimenta o sistema com dez mil fotos de gatos e dez mil fotos de não-gatos. O algoritmo processa esses pixels e cria, sozinho, seu próprio modelo matemático do que define um felino. Ninguém disse a ele o que procurar; ele encontrou correlações estatísticas e padrões visuais que se repetem nas imagens rotuladas como corretas.

Contudo, o perigo social não reside apenas na matemática, mas no momento em que essa estatística sai do laboratório e é empacotada como mercadoria para governos e empresas. No ambiente controlado da ciência, sabe-se que esses modelos operam puramente com probabilidades: o

resultado nunca é uma afirmação absoluta como "isto é um gato", mas sim uma estimativa como "há 94% de chance estatística de ser um gato". O problema surge quando essa tecnologia vira um produto de prateleira vendido para uma Secretaria de Segurança Pública ou para o RH de uma grande corporação.

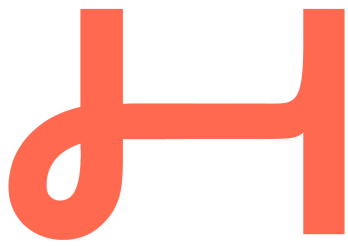
Nesse processo de comercialização, a incerteza inerente ao cálculo é deliberadamente escondida. A interface do software apaga a porcentagem de dúvida e exhibe apenas um alerta vermelho e definitivo: "SUSPEITO DETECTADO" ou "CANDIDATO REJEITADO". O produto transforma, assim, uma dúvida estatística que pode estar errada em uma certeza administrativa que gera ações reais, como uma abordagem policial truculenta ou uma demissão sumária. É aqui que a inferência se torna perigosa: ela automatiza o preconceito sob o disfarce irrefutável da eficiência tecnológica.

O caso do reconhecimento facial ilustra tragicamente essa dinâmica. O que o mercado vende como identificação biométrica é, na verdade, um cálculo de geometria probabilística. O sistema converte o rosto humano em um mapa numérico (um vetor), medindo a distância entre os olhos, a largura do nariz e o formato do queixo. A falha técnica e ética desse sistema amplamente denunciada por pesquisadoras como Joy Buolamwini no projeto *Gender Shades* ocorre na fase de treinamento. Como muitos desses "produtos" foram treinados majoritariamente com fotos de homens brancos (o padrão demográfico do Vale do Silício), o algoritmo aprendeu a inferir detalhes nesse tipo de rosto com alta precisão. Quando a mesma ferramenta é aplicada no Brasil, sobre rostos negros, indígenas ou femininos, a inferência falha. A máquina confunde as geometrias, mas, por ser um produto fechado, ela não admite que "não sabe". Ela força um "Match Encontrado". O resultado não é um mero *bug* técnico; é uma violação de direitos humanos baseada em uma estatística mal calibrada vendida como verdade absoluta.

A mesma lógica opera nos filtros de seleção de Recursos Humanos. Plataformas de recrutamento vendem algoritmos que prometem ler milhares de currículos e selecionar os "melhores", mas raramente explicam como a máquina define qualidade. Na prática, o algoritmo aprende olhando para o passado da empresa: ele analisa quem foi contratado e promovido na última década para buscar padrões semelhantes nos novos currículos. Se a empresa historicamente privilegiou homens brancos de universidades de elite, o algoritmo inferirá que atributos como "gênero masculino" e "CEP nobre" são preditores matemáticos de competência. Como alerta a matemática Cathy O'Neil, o algoritmo funciona aqui como um "espelho conservador": ele automatiza e amplifica o viés de contratação que já existia. Currículos de mulheres ou pessoas de periferia são descartados não por falta de qualificação, mas porque não correspondem ao padrão estatístico do sucesso anterior. O produto foi desenhado para otimizar a eficiência, e a forma mais eficiente de contratar para a máquina é repetir o passado.

Para agravar esse cenário, esses produtos chegam ao mercado como "Caixas Pretas" (*Black Boxes*). Esse termo carrega dois sentidos que se sobrepõem: o técnico e o comercial. Tecnicamente, em redes neurais profundas (*Deep Learning*), o cálculo é tão complexo e passa por tantas camadas abstratas que é difícil rastrear exatamente como a decisão foi tomada. Comercialmente, as empresas alegam "segredo industrial" para impedir auditorias externas. Isso cria um perigoso vácuo de justiça: o cidadão tem seu crédito negado ou sua liberdade cerceada e, ao perguntar o porquê, recebe como resposta apenas que "foi o sistema". Sem acesso à lógica da inferência, o direito de defesa desaparece.

Essa transição da regra escrita (determinística) para a regra inferida (IA) exige um novo tipo de governança. Não basta mais ler o código; precisamos auditar os dados, testar os resultados e questionar a validade ética de usar o passado desigual como bússola para o futuro. É sobre os detalhes desse universo as alucinações das IAs Generativas, o funcionamento das Redes Neurais e as ferramentas de auditoria que nos debruçaremos exclusivamente no **Cadernos da Hora – Volume 2: Inteligência Artificial e a Caixa Preta**.



Referências Bibliográficas

Livros e Obras Principais

- CHRISTIAN, Brian; GRIFFITHS, Tom. Algorithms to Live By: The Computer Science of Human Decisions. Henry Holt and Co., 2016. (Citado no texto como: "Algoritmos para Viver").
- EUBANKS, Virginia. Automating Inequality: How High-Tech Tools Profile, Police, and Punish the Poor. St. Martin's Press, 2018. (Citado no texto como: "Automating Inequality").
- KNUTH, Donald E. The Art of Computer Programming, Vol. 1: Fundamental Algorithms. Addison-Wesley, 1968. (Citado pela definição de algoritmo).
- LESSIG, Lawrence. Code: and Other Laws of Cyberspace. Basic Books, 1999. (Referência ao conceito "Code is Law").
- O'NEIL, Cathy. Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy. Crown, 2016. (Citado no texto como: "Weapons of Math Destruction").
- ROSENBLAT, Alex. Uberland: How Algorithms Are Rewriting the Rules of Work. University of California Press, 2018. (Citado no texto como: "Uberland").

Artigos e Projetos de Pesquisa

- BUOLAMWINI, Joy; GEBRU, Timnit. "Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification". Proceedings of the 1st Conference on Fairness, Accountability and Transparency, PMLR 81:77-91, 2018. (Referência ao projeto Gender Shades e ao trabalho de Joy Buolamwini).

Referências Históricas e Conceituais (Mencionados no texto)

- BOOLE, George. The Mathematical Analysis of Logic (1847) ou An Investigation of the Laws of Thought (1854). (Referência à Lógica Booleana).
- LEIBNIZ, Gottfried Wilhelm. (Referência ao conceito filosófico de Mathesis Universalis).

INSTITUTO da HORA

Website

institutodahora.com

Contato

contato@institutodahora.com