Projetando Sistemas de Machine Learning

Um Processo Interativo para Aplicações Prontas para Produção

Chip Huyen

ALTA BOOKS
GRUPO EDITORIAL
Rio de Janeiro, 2023

Sumário

CAPÍTULO 1
Visão Geral dos Sistemas de Machine Learning

Em novembro de 2016, a Google anunciou que havia incorporado seu sistema de tradução automática neural multilíngue ao Google Tradutor, sinalizando uma das primeiras histórias de sucesso de redes neurais artificiais profundas implementadas em produção e em escala.[1] Segundo o Google, com essa novidade, a qualidade da tradução melhorou mais em um único avanço do que havia sido visto nos últimos 10 anos combinados.

Esse sucesso do aprendizado profundo reavivou o interesse pelo machine learning (ML) em geral. Desde então, mais e mais empresas se voltaram ao ML em busca de soluções para seus problemas mais desafiadores. Em apenas cinco anos, o ML se introduziu em quase todos os aspectos de nossas vidas: como acessamos informações, como nos comunicamos, como trabalhamos, como encontramos o amor. A disseminação do ML tem sido tão rápida que já é difícil imaginar a vida sem ele. No entanto, ainda há muitos outros casos de uso de ML esperando para serem explorados em campos de atuação como assistência médica, transporte, agricultura e até mesmo para nos ajudar a compreender o universo[2]. Ao ouvirem "sistema de machine learning", muitos pensam somente nos algoritmos de ML que estão sendo usados, como regressão logística ou diferentes tipos de redes neurais. No entanto, o algoritmo é apenas uma pequena parte de um sistema de machine learning em produção. O sistema também engloba os requisitos de negócios que originaram o projeto de ML, a interface em que usuários e desenvolvedores interagem com seu sistema, a data stack e a lógica para desenvolver, monitorar e atualizar seus modelos, bem como a infraestrutura que viabiliza a entrega dessa lógica. A Figura

[1] Mike Schuster, Melvin Johnson e Nikhil Thorat, "Zero-Shot Translation with Google's Multilingual Neural Machine Translation System", *Google AI Blog*, 22 de novembro de 2016. Disponível em: *https://oreil.ly/2R1CB*.

[2] Larry Hardesty, "A Method to Image Black Holes", *MIT News*, 6 de junho de 2016. Disponível em: *https://oreil.ly/HpL2F*.

1-1 mostra os diferentes componentes de um sistema de ML e em quais capítulos deste livro eles serão abordados.

A Relação entre MLOps e Design de Sistemas de ML

Ops em MLOps, Machine Learning Operations [Operações de Machine Learning] se origina do termo DevOps, abreviação de Developments and Operations, combinação dos termos Desenvolvimentos e Operações. Operacionalizar algo significa colocá-lo em produção, o que inclui implementá-lo, monitorá-lo e fazer a manutenção. MLOps é um conjunto de ferramentas e melhores práticas para disponibilizar o machine learning em produção. O design de sistemas de ML adota uma abordagem de sistema MLOps. Ou seja, essa abordagem considera um sistema de ML de forma holística para garantir que todos os componentes e suas partes interessadas possam trabalhar juntos a fim de atender os objetivos e requisitos especificados.

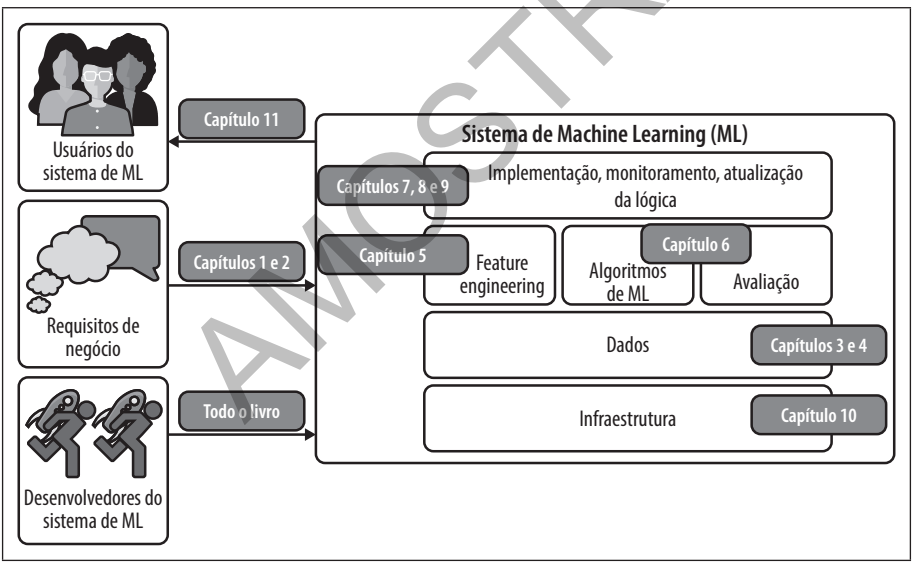

Figura 1-1. Componentes diferentes de um sistema de ML. Em geral, "algoritmos de ML" é o que as pessoas pensam quando dizem machine learning, mas isso é somente uma pequena parte de todo o sistema.

Há muitos livros excelentes sobre uma variedade imensa de algoritmos de ML. Neste livro, não abordamos nenhum algoritmo específico em detalhes, mas ajudamos os leitores a entender o sistema de ML em sua totalidade. Em outras palavras, o objetivo desta obra é dar um referencial a fim de desenvolver uma solução que

funcione melhor para o seu problema, independentemente de qual algoritmo você acabe usando. Por mais que os algoritmos fiquem defasados com rapidez, à medida que novos algoritmos são constantemente desenvolvidos, o referencial proposto aqui ainda deve funcionar com novos algoritmos.

O primeiro capítulo tem o intuito de fornecer uma visão geral do que é preciso para disponibilizar um modelo de ML em produção. Antes de analisar como desenvolver um sistema de ML, é importante fazer uma pergunta fundamental sobre quando usar e não usar o ML. Abordaremos alguns dos casos de uso populares de ML para exemplificar esse ponto.

Após os casos de uso, abordaremos os desafios de implementação de sistemas de ML, comparando o ML em produção com o ML em pesquisa, bem como ao software tradicional. Se você esteve na linha de frente do desenvolvimento de sistemas de machine learning, talvez já esteja familiarizado com o conteúdo deste capítulo. No entanto, caso tenha experiência com ML em contextos acadêmicos, este capítulo fornecerá uma perspectiva sincera do ML no mundo real, mostrando como configurar com sucesso sua primeira aplicação.

Quando Usar o Machine Learning

À medida que sua adoção no setor cresce rapidamente, o ML provou ser uma ferramenta poderosa para um amplo leque de problemas. Apesar do tremendo entusiasmo e alarde gerado por pessoas dentro e fora da área, o ML não é uma ferramenta mágica que consegue resolver todos os problemas; ele pode até solucionar problemas, mas suas soluções podem não ser as melhores. Antes de iniciar um projeto de ML, talvez você esteja se perguntando se o machine learning é necessário ou eficaz na redução de custos.[3] Para entender do que o ML é capaz, vamos examinar o que as soluções de ML costumam fazer:

> O machine learning é uma abordagem para (1) *aprender* (2) *padrões complexos* a partir de (3) *dados existentes* e usar esses padrões para fazer (4) *predições* sobre (5) *dados desconhecidos*.

Analisaremos cada uma das frases-chave em itálico acima a fim de compreender suas consequências para os problemas que o ML pode resolver:

1. *Aprender: o sistema tem a capacidade de aprender*

   Um banco de dados relacional não é um sistema de ML porque não tem a capacidade de aprender. Você pode declarar explicitamente o relacionamento entre duas colunas em um banco de dados relacional, mas é improvável que tenha a capacidade de descobrir o relacionamento entre essas duas colunas por si só.

[3] Não perguntei se o ML é suficiente porque a resposta é sempre não.

Para que um sistema de ML aprenda, deve haver algo com o qual ele possa aprender. Na maioria dos casos, os sistemas de ML aprendem com os dados. No aprendizado supervisionado, baseados em exemplos de pares de entrada [input] e saída [output], os sistemas de ML aprendem a gerar saídas para entradas arbitrárias. Por exemplo, se o objetivo for construir um sistema de ML a fim de aprender a predizer o valor do aluguel para anúncios do Airbnb, é necessário fornecer um conjunto de dados em que cada entrada seja um anúncio com características relevantes (metros quadrados, número de quartos, bairro, comodidades, avaliação desse anúncio etc.) e a saída associada seja o valor do aluguel desse anúncio. Quando aprender, esse sistema de ML deve ser capaz de predizer o valor de um novo anúncio, dadas as suas características.

2. *Padrões complexos: há padrões para aprender e eles são complexos*

   As soluções de ML só são vantajosas quando há padrões para aprender. Pessoas sensatas não investem dinheiro na construção de um sistema de ML a fim de predizer o próximo resultado do lançamento de um dado porque não há um padrão em como esses resultados são gerados.[4] Contudo, há padrões na precificação das ações, por isso as empresas investiram bilhões de dólares na construção de sistemas de ML para aprender esses padrões.

   A existência de um padrão pode não ser óbvia ou, se existirem, seu conjunto de dados ou algoritmos de ML podem não ser suficientes para capturá-los. Por exemplo, pode haver um padrão em como os tweets de Elon Musk impactam os preços das criptomoedas. No entanto, você não saberia até ter treinado e avaliado rigorosamente seus modelos de ML nos tweets dele. Mesmo que todos os seus modelos não consigam realizar predições razoáveis dos preços das criptomoedas, isso não significa que não haja um padrão.

   Pense em um site como o Airbnb, com muitos anúncios de casas; cada anúncio é acompanhado de um CEP. Se você quiser classificar os anúncios nos estados em que estão localizados, não precisará de um sistema de ML. Como o padrão é simples — cada código postal corresponde a um estado conhecido — é possível usar uma tabela lookup.

   A relação entre o valor de um aluguel e todas as suas características segue um padrão bem mais complexo, que seria bastante desafiador de especificar manualmente. Uma boa solução para isso é o ML. Em vez de informar ao seu sistema como calcular o valor a partir de uma lista de características, você pode fornecer preços e características e deixá-lo descobrir o padrão. A diferença entre as soluções de ML e a solução de tabela lookup, assim como as soluções

[4] Os padrões são diferentes das distribuições. Conhecemos a distribuição dos resultados do lançamento de um dado, mas não há padrões na forma como os resultados são gerados.

de software tradicionais gerais, é mostrada na Figura 1-2. Por isso, também chamamos o ML de Software 2.0.[5]

O ML tem sido um grande sucesso com tarefas de padrões complexos, como detecção de objetos e reconhecimento de fala, pois os níveis de complexidade para máquinas e para humanos são diferentes. Os humanos têm dificuldade de fazer muitas tarefas que são fáceis para as máquinas — por exemplo, elevar um número à potência de 10. Em contrapartida, muitas tarefas fáceis para os seres humanos podem ser difíceis para as máquinas — por exemplo, decidir se há um gato em uma imagem.

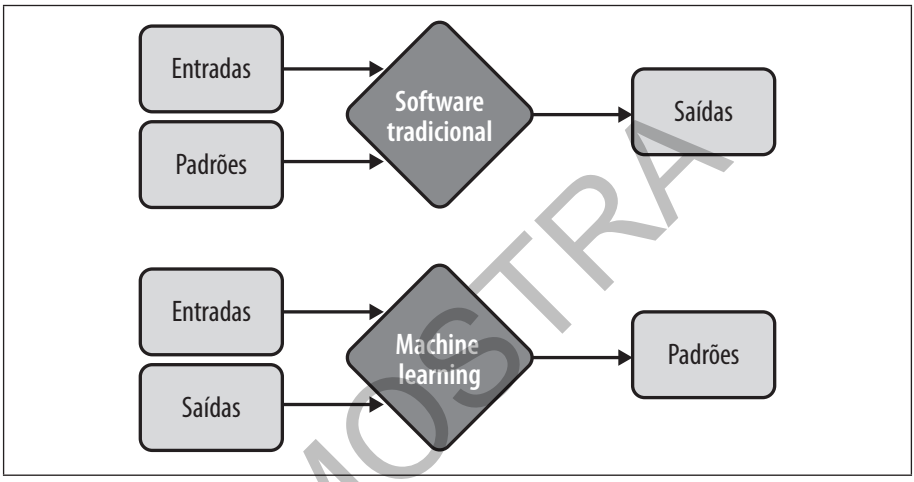

Figura 1-2. Em vez de exigir padrões especificados manualmente para calcular saídas, as soluções de ML aprendem padrões a partir de entradas e saídas.

3. Dados existentes: os dados estão disponíveis ou é possível coletar dados

Como o ML aprende com os dados, é necessário haver dados com os quais ele possa aprender. É curioso pensar em construir um modelo para predizer quanto imposto uma pessoa deve pagar por ano, só que isso não é possível a menos que você tenha acesso a dados de impostos e do salário de uma grande população.

No contexto do aprendizado zero-shot (*https://oreil.ly/ZshSg*) (às vezes conhecido como aprendizado de dados zero), é possível que um sistema de ML faça boas predições para uma tarefa, sem antes ter sido treinado em dados para ela. No entanto, esse sistema de ML foi previamente treinado em dados para outras

[5] Andrej Karpathy, "Software 2.0", *Medium*, 11 de novembro de 2017. Disponível em: *https://oreil.ly/yHZrE*.

tarefas, muitas vezes relacionadas à tarefa em questão. Portanto, mesmo que o sistema não exija dados para a tarefa, ele ainda exige dados para aprender.

É possível também lançar um sistema de ML sem dados. Por exemplo, no contexto da aprendizagem contínua, é possível fazer deploy de modelos que não tenham sido treinados em nenhum dado, já que eles aprenderão com os dados de entrada em produção.[6] No entanto, disponibilizar modelos insuficientemente treinados aos usuários apresenta certos riscos, como uma customer experience medíocre. Sem dados e sem aprendizado contínuo, muitas empresas adotam uma abordagem de "fingir até se tornar real": lançando um produto que disponibiliza predições feitas por seres humanos em vez de modelos de ML, com a esperança de usar os dados gerados para treinar posteriormente os modelos de ML.

4. Predições: um problema preditivo

Os modelos de ML realizam predições, então apenas conseguem resolver problemas que exigem respostas preditivas. O ML pode ser bastante atrativo quando você pode se beneficiar de uma grande quantidade de predições de baixo custo, ainda que aproximadas. Em inglês, a palavra "predict [predição]" significa "estimar um valor no futuro". Por exemplo, como estará o tempo amanhã? Quem ganhará o Super Bowl este ano? Qual o próximo filme que um usuário vai querer assistir?

À medida que as máquinas preditivas (por exemplo, modelos de ML) estão se tornando mais eficazes, mais e mais problemas estão sendo ressignificados como problemas preditivos. Seja qual for a pergunta que tenha, você sempre pode delimitá-la como: "Qual seria a resposta para essa pergunta?" independentemente de essa pergunta ter relação com o futuro, presente ou mesmo com o passado.

Os problemas de computação intensiva são uma classe de problemas que foram reformulados com sucesso como preditivos. Em vez de calcular o resultado exato de um processo, que pode ser computacionalmente mais oneroso e demorado do que o ML, é possível delimitar o problema assim: "Como seria o resultado desse processo?" e aproximá-lo usando um modelo de ML. A saída será uma aproximação da saída exata, porém, muitas vezes, boa o suficiente. Podemos ver muito isso em renderizações gráficas, como eliminação de ruído de imagem e sombreamento de espaço de tela.[7]

[6] Falaremos sobre aprendizado online no Capítulo 9.

[7] Steke Bako, Thijs Vogels, Brian McWilliams, Mark Meyer, Jan Novák, Alex Harvill, Pradeep Sen, Tony Derose e Fabrice Rousselle, "Kernel-Predicting Convolutional Networks for Denoising Monte Carlo Renderings", *ACM Transactions on Graphics* 36, no. 4 (2017): 97. Disponível em: *https://oreil.ly/EeI3j*; Oliver Nalbach, Elena Arabadzhiyska, Dushyant Mehta, Hans-Peter Seidel e Tobias Ritschel, "Deep Shading: Convolutional Neural Networks for Screen-Space Shading', *arXiv*, 2016. Disponível em: *https://oreil.ly/dSspz*.

5. Dados desconhecidos: dados desconhecidos compartilham padrões com os dados de treinamento

Os padrões que seu modelo aprende com os dados existentes somente serão úteis se os dados desconhecidos também compartilharem esses padrões. Um modelo para predizer o número de downloads de um app no Natal de 2020 não funcionará muito bem se for treinado em dados de 2008, quando o app mais popular na App Store era o Koi Pond. Mas o que é Koi Pond? Pois é.

Em termos técnicos, significa que seus dados desconhecidos e dados de treinamento devem vir de distribuições semelhantes. Você pode perguntar: "Se os dados não forem desconhecidos, como saberemos de qual distribuição vêm?" Não saberemos, mas podemos fazer suposições — como podemos supor que os comportamentos dos usuários de amanhã não serão muito diferentes dos comportamentos dos usuários de hoje — e esperar que nossas suposições se comprovem. Caso contrário, teremos um modelo com desempenho insatisfatório, o que talvez possamos descobrir com o monitoramento, conforme abordado no Capítulo 8, e o teste em produção, como abordado no Capítulo 9.

Devido à forma como a maioria dos algoritmos atuais de ML aprendem, as soluções de ML virão à tona, ainda mais se seu problema apresentar as características adicionais a seguir:

6. É repetitivo

Os seres humanos são ótimos no aprendizado few-shot: você pode mostrar às crianças algumas fotos de gatos e a maioria delas reconhecerá um gato na próxima vez que vir um. Apesar do incrível progresso na pesquisa de aprendizado few-shot, boa parte dos algoritmos de ML ainda exige muitos exemplos para aprender um padrão. Quando uma tarefa é repetitiva, cada padrão é repetido diversas vezes, facilitando a aprendizagem das máquinas.

7. O custo de predições erradas é baixo

A menos que o desempenho do seu modelo de ML seja sempre 100%, algo bastante improvável para quaisquer tarefas significativas, ele cometerá erros. O ML é particularmente indicado quando o custo de uma predição errada é baixo. Por exemplo, um dos maiores casos de uso de ML hoje é em sistemas de recomendação, pois com sistemas de recomendação, uma recomendação ruim geralmente é perdoada — o usuário simplesmente não clica nela.

Caso um erro de predição possa ter consequências catastróficas, o ML ainda pode ser uma solução adequada se, em média, os benefícios das predições corretas superarem o custo das predições erradas. Desenvolver carros autônomos é um desafio, pois um erro algorítmico pode levar à morte. Apesar disso, muitas empresas ainda querem desenvolver carros autônomos porque eles têm o potencial de salvar muitas vidas, já que os carros autônomos são estatisticamente mais seguros do que os motoristas humanos.

8. *Está em escala*

   Em geral, as soluções de ML exigem investimentos iniciais e consideráveis em dados, processamento computacional, infraestrutura e talentos, por isso faria sentido se pudéssemos usá-las em larga escala. "Em escala" significa coisas diferentes para tarefas distintas, mas, via de regra, significa muitas predições. Como exemplo, podemos citar a classificação de milhões de e-mails por ano ou a predição para quais departamentos milhares de chamados de suporte devem ser encaminhados em um dia.

   Um problema pode parecer uma única predição, mas, na verdade, envolve uma série de predições. Por exemplo, um modelo que prediz quem vencerá uma eleição presidencial nos EUA aparentemente faz somente uma predição a cada quatro anos, mas pode estar fazendo uma predição a cada hora ou ainda com mais frequência, já que essa predição precisa ser continuamente atualizada para incorporar novas informações. Um problema em escala implica em uma abundância de dados, o que é útil para o treinamento de modelos de ML.

9. *Os padrões estão constantemente mudando*

   As culturas mudam. Os gostos mudam. As tecnologias mudam. O que é tendência hoje pode ser águas passadas amanhã. Pense na tarefa de classificação de spam por e-mail. Hoje, a indicação de um e-mail de spam é um príncipe nigeriano, mas amanhã pode ser um escritor vietnamita transtornado.

   Se o seu problema envolve um ou mais padrões em constante mudança, soluções baseadas em regras fixas podem ficar obsoletas rapidamente. Descobrir como seu problema mudou para que você possa atualizar suas regras manuscritas de modo adequado pode ser muito oneroso ou impossível. Como o ML aprende com os dados, é possível atualizar seu modelo de ML com novos dados sem ter que descobrir como os dados mudaram. É possível também preparar seu sistema para se adaptar às mudanças nas distribuições de dados, abordagem que analisaremos na seção "Aprendizado Contínuo" do Capítulo 9.

A lista de casos de uso pode continuar indefinidamente à medida que a adoção do ML amadurece no setor. Embora o ML consiga resolver um subconjunto de problemas muito bem, ele não pode resolver e/ou não deve ser usado em muitos problemas. A maioria dos algoritmos atuais de ML não deve ser usada em nenhuma das seguintes condições:

- É antiético. Analisaremos um estudo de caso em que o uso de algoritmos de ML pode ser considerado antiético na seção "Estudo de caso I: Vieses automatizados do avaliador" do Capítulo 11.
- Soluções mais simples resolvem o problema. No Capítulo 6, abordaremos as quatro fases do desenvolvimento do modelo de ML, em que a primeira fase deve ser soluções sem ML.
- Não é eficaz na redução de custos.

Apesar disso, mesmo que o ML não consiga resolver seu problema, pode ser possível dividi-lo em componentes menores e usá-lo para resolver alguns deles. Por exemplo, se você não conseguir criar um chatbot para responder às perguntas de todos os seus clientes, talvez seja possível criar um modelo de ML para predizer se uma consulta corresponde a uma das perguntas frequentes. Em caso afirmativo, direcione o cliente à resposta. Caso contrário, direcione-os para o atendimento ao cliente.

Gostaria também de alertar contra a rejeição de uma nova tecnologia porque não compensa tanto na redução de custos quanto às tecnologias existentes no momento. A maioria dos avanços tecnológicos é incremental. Um tipo de tecnologia pode não ser eficiente hoje, mas pode ser ao longo do tempo com mais investimentos. Se você esperar que a tecnologia prove seu valor para o resto do setor antes de ser disponibilizada, pode acabar anos ou décadas atrás de seus concorrentes.

Casos de Uso de Machine Learning

O ML está em uso crescente em aplicações corporativas e de consumo. Desde meados da década de 2010, houve a franca expansão de aplicações que aproveitam o ML para fornecer aos consumidores serviços superiores ou até então impraticáveis.

Com a explosão de informações e serviços, teria sido muito desafiador para nós encontrar o que queremos sem a ajuda do ML, integrados em um *mecanismo de busca* ou em um *sistema de recomendação*. Ao acessar um site como Amazon ou Netflix, você recebe itens recomendados a partir de predições que melhor se adéquam ao seu gosto. Se não gostar de nenhuma das suas recomendações, é possível pesquisar itens específicos e seus resultados de pesquisa provavelmente são alimentados por ML.

Caso tenha um smartphone, é bem provável que o ML já o esteja ajudando em muitas de suas atividades diárias. Digitar em seu telefone é mais fácil com a *digitação preditiva*, um sistema de ML que dá sugestões sobre o que você pode querer dizer. Um sistema de ML pode estar rodando em seu app de edição de fotos para sugerir a melhor forma de aprimorá-las. É possível usar autenticação em seu celular por meio de sua impressão digital ou rosto, o que requer um sistema de ML para predizer se uma impressão digital ou um rosto corresponde ao seu.

O caso de uso de ML que me atraiu para a área foi a *machine translation*, a tradução automática de um idioma para outro. Essa funcionalidade tem o potencial de possibilitar que pessoas de diferentes culturas se comuniquem entre si, eliminando a barreira linguística. Meus pais não falam inglês, mas graças ao Google Tradutor, agora eles podem ler minha escrita e conversar com meus amigos que não falam vietnamita.

O ML está cada vez mais presente em nossas casas por meio de assistentes pessoais inteligentes, como a Alexa e o Google Assistente. Câmeras de segurança inteligentes podem avisá-lo quando seus animais de estimação saem de casa ou se você tiver uma visita indesejada. Um amigo meu estava preocupado com a mãe idosa morando sozinha — se ela caísse, ninguém estaria lá para ajudá-la a se levantar — então ele recorreu a um sistema de monitoramento de saúde em casa que prediz se alguém pode cair.

Embora o mercado de aplicações de ML para consumidores esteja crescendo, a maioria dos casos de uso de ML ainda está no mundo corporativo. As aplicações corporativas de ML costumam ter requisitos e considerações muito diferentes das aplicações de consumo. Há muitas exceções que fogem à regra, porém, na maioria dos casos, as aplicações corporativas podem ter requisitos de acurácia mais rigorosos, ainda que mais tolerantes com os requisitos de latência. Por exemplo, talvez melhorar a acurácia de um sistema de reconhecimento de fala de 95% para 95,5% não seja perceptível para maioria dos consumidores, porém melhorar a eficiência de um sistema de alocação de recursos em apenas 0,1% pode ajudar uma empresa como o Google ou a General Motors a economizar milhões de dólares. Por outro lado, a latência de um segundo pode distrair um consumidor a abrir outra coisa, mas os usuários corporativos podem ser mais tolerantes à alta latência. Para pessoas interessadas em construir empresas a partir de aplicações de ML, as aplicações de consumo podem ser mais fáceis de distribuir, ainda que mais difíceis de monetizar. Ainda assim, a maioria dos casos corporativos de uso não é evidente, a menos que você mesmo os tenha encontrado.

Segundo a pesquisa de machine learning corporativo da Algorithmia de 2020, as aplicações de ML nas empresas são diversas, atendendo a casos de uso interno (redução de custos, geração de insights e inteligência do cliente, automação de processamento interno) e casos de uso externo (melhoria da customer experience, retenção de clientes, interação com os consumidores), conforme mostrado na Figura 1-3.[8]

[8] "2020 State of Enterprise Machine Learning", *Algorithmia*, 2020: Disponível em: *https://oreil.ly/wKMZB*.

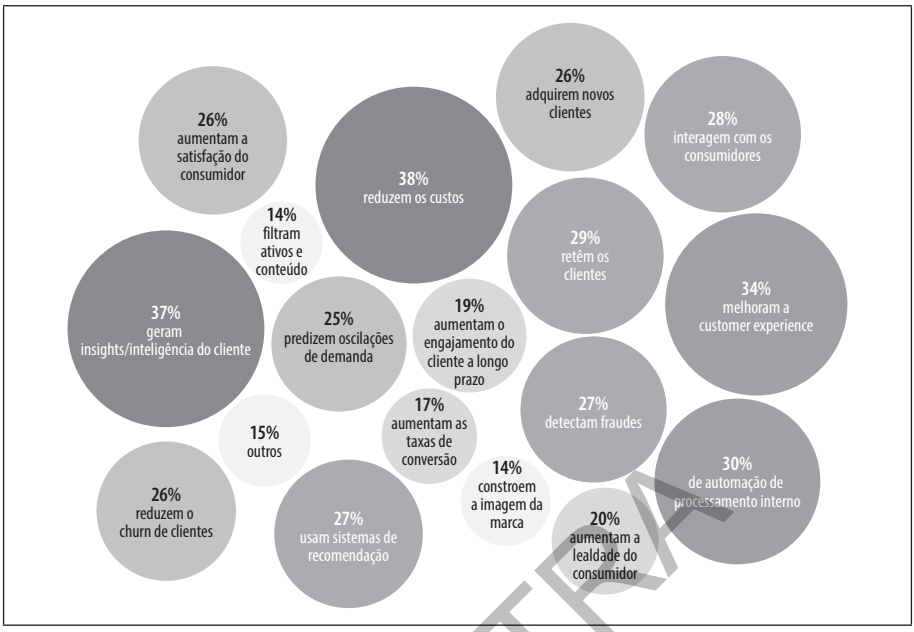

Figura 1-3. Machine learning corporativo de 2020. Fonte: Adaptado de uma imagem da Algorithmia.

A detecção de fraudes está entre as aplicações mais antigas do ML no mundo corporativo. Se o seu produto ou serviço envolver transações de qualquer valor, está suscetível à fraude. Ao aproveitar as soluções de ML para detecção de anomalias, é possível ter sistemas que aprendem com transações de fraude históricas e predigam se uma transação futura é fraudulenta.

Decidir quanto cobrar pelo seu produto ou serviço é provavelmente uma das decisões de negócios mais difíceis; então por que não deixar o ML fazer isso para você? A *otimização de preços* é o processo de estimar um preço em um determinado período de tempo a fim de maximizar uma função objetivo definida, como a margem, receita ou taxa de crescimento da empresa. A otimização de preços baseada em ML é mais adequada para casos com um grande número de transações, em que a demanda oscila e os consumidores estão dispostos a pagar um preço dinâmico — por exemplo, anúncios na internet, passagens aéreas, reservas de acomodação, ride-sharing e eventos.

Para administrar uma empresa, é importante ser capaz de prever a demanda do cliente, assim você consegue preparar um orçamento, o inventário do estoque, alocar recursos e atualizar a estratégia de preços. Por exemplo, na administração de uma mercearia, é bom estocar o suficiente para que os clientes encontrem o que

estão procurando, mas não estocar demais, caso contrário, as mercadorias podem estragar e haverá prejuízo.

Adquirir um novo usuário custa caro. A partir de 2019, o custo médio para um app adquirir um usuário que fará uma compra nele era de US$86,61.[9] O custo de aquisição da Lyft está estimado em US$158 por motorista.[10] Esse custo é bem maior para os clientes corporativos. Os investidores declaram publicamente que o custo de aquisição do cliente "mata" as startups.[11] Reduzir um pouco custos de aquisição de clientes pode resultar em um grande aumento no lucro. É possível fazer isso por meio de uma melhor identificação de clientes em potencial, mostrando anúncios mais bem direcionados, fornecendo descontos no momento certo etc. — todas essas tarefas são adequadas para o ML.

Após gastar tanto dinheiro adquirindo um cliente, seria uma pena se ele fosse embora. O custo de aquisição de um novo usuário fica aproximadamente de 5 a 25 vezes mais caro do que reter um existente.[12] A *Churn prediction* prediz quando um cliente específico está prestes a parar de usar seus produtos ou serviços, assim você pode tomar as medidas apropriadas para recuperá-los. A *churn prediction* pode ser usada não apenas com clientes, mas também com colaboradores.

Para evitar perder clientes, é importante mantê-los felizes, respondendo às suas preocupações assim que surgirem. A classificação automatizada de chamados de suporte pode ajudar nisso. Antes, quando um cliente abria um ticket de suporte ou enviava um e-mail, era necessário primeiro processá-lo e depois passá-lo para diferentes departamentos até chegar à caixa de entrada de alguém que pudesse atendê-lo. Um sistema de ML pode analisar o conteúdo do ticket e predizer para onde deve ir, o que pode reduzir o tempo de resposta e melhorar a satisfação do cliente. Ele também pode ser usado para classificar tickets de chamados internos de TI.

Outro caso corporativo de uso popular da ML é o monitoramento da marca. A marca é um ativo valioso de um negócio.[13] É importante monitorar como o público e seus clientes percebem sua marca. Você pode querer saber quando/onde/como é mencionado, tanto explicitamente (por exemplo, quando alguém menciona "Google") ou implicitamente (por exemplo, quando alguém diz "o gigante da pesquisa"), bem como o sentimento associado a ela. De repente, se houver uma onda

[9] "Average Mobile App User Acquisition Costs Worldwide from September 2018 to August 2019, by User Action and Operating System", *Statista*, 2019. Disponível em: *https://oreil.ly/2pTCH*.

[10] Jeff Henriksen, "Valuing Lyft Requires a Deep Look into Unit Economics", *Forbes*, 17 de maio de 2019. Disponível em: *https://oreil.ly/VeSt4*.

[11] David Skok, "Startup Killer: The Cost of Customer Acquisition", *For Entrepreneurs*, 2018. Disponível em: *https://oreil.ly/L3tQ7*.

[12] Amy Gallo, "The Value of Keeping the Right Customers", *Harvard Business Review*, 29 de outubro de 2014. Disponível em: *https://oreil.ly/OlNkl*.

[13] Marty Swant, "The World's 20 Most Valuable Brands", *Forbes*, 2020. Disponível em: *https://oreil.ly/4uS5i*.

de sentimentos negativos em suas menções de marca, talvez seja bom resolver a situação o mais rápido possível. A análise de sentimentos é uma típica tarefa do ML.

Um conjunto de casos de uso de ML que tem gerado muito entusiasmo recentemente está no setor de assistência médica. Existem sistemas de ML que podem detectar câncer de pele e diagnosticar diabetes. Ainda que muitas aplicações de assistência médica sejam voltadas aos consumidores, devido aos seus requisitos rigorosos com acurácia e privacidade, elas geralmente são fornecidas por meio de um prestador de serviços de assistência médica, como um hospital, ou usadas para auxiliar os médicos no diagnóstico.

Entendendo os Sistemas de Machine Learning

Compreender os sistemas de ML será útil para projetá-los e desenvolvê-los. Nesta seção, veremos como os sistemas de ML são diferentes dos sistemas de ML na área de pesquisa acadêmica (ou como muitas vezes ensinado na escola) e do software tradicional, o que motiva a necessidade deste livro.

Machine Learning em Pesquisa versus em Produção

Como o uso de ML na indústria ainda é relativamente novo, a maioria das pessoas com experiência em ML adquiriu conhecimento no mundo acadêmico: fazendo cursos e pesquisas, lendo trabalhos acadêmicos. Se esse for seu histórico, você pode enfrentar uma curva de aprendizado íngreme para entender os desafios de implementar sistemas de ML em ambiente de desenvolvimento ou fora do controle e para se orientar devido a um conjunto esmagador de soluções para esses desafios. O ML em produção é bem diferente do ML em pesquisa. A Tabela 1-1 mostra cinco das principais diferenças.

Tabela 1-1. Principais diferenças entre o ML na pesquisa e o ML em produção

	Na pesquisa acadêmica	Em produção
Requisitos	Desempenho do modelo de última geração em conjuntos de dados comparativos	Partes interessadas diferentes têm requisitos diferentes
Prioridade computacional	Treinamento rápido, taxa de requisição alta	Inferência rápida, baixa latência
Dados	Estáticos[a]	Mudança constante
Imparcialidade	Muitas vezes não é o foco	Devem ser considerados
Interpretabilidade	Muitas vezes não é o foco	Devem ser considerados

[a] Um subcampo de pesquisa se concentra no aprendizado contínuo: desenvolver modelos para trabalhar com a mudança nas distribuições de dados. No Capítulo 9, abordaremos o aprendizado contínuo.

Partes interessadas e requisitos diferentes

As pessoas envolvidas em um projeto de pesquisa e classificação muitas vezes têm um único objetivo. O objetivo mais comum é o modelo de desempenho — desenvolver um modelo que alcance os resultados mais avançados em conjuntos de dados de referência. Para superar uma pequena melhoria no desempenho, os pesquisadores muitas vezes recorrem a técnicas que tornam os modelos muito complexos para serem úteis.

Há muitas partes interessadas envolvidas quando se trata de colocar um sistema de ML em produção. Cada parte interessada tem os próprios requisitos. Requisitos diferentes, não raro conflitantes, podem dificultar o design, o desenvolvimento e a seleção de um modelo de ML que atenda a todos os requisitos. Considere um app móvel que recomende restaurantes aos usuários. O app gera receita ao cobrar uma taxa de serviço de 10% dos restaurantes por cada pedido. Ou seja, os pedidos de maior valor geram mais dinheiro ao aplicativo do que os de menor valor. O projeto envolve engenheiros de ML, vendedores, product managers, engenheiros de infraestrutura e um gerente:

Engenheiros de ML
: Querem um modelo que recomende restaurantes, por meio do qual é bem provável que os usuários façam um pedido. Acreditam que podem fazer isso usando um modelo mais complexo com mais dados.

Equipe de vendas
: Querem um modelo que recomende os restaurantes mais caros, já que esses restaurantes oportunizam mais taxas de serviço.

Equipe de produtos
: Percebe que cada aumento na latência resulta em queda nos pedidos por meio do serviço, assim, querem um modelo que possa retornar os restaurantes recomendados em menos de 100 milissegundos.

Equipe da plataforma de ML
: À medida que o tráfego aumenta, essa equipe tem que acordar no meio da noite por causa de problemas com o dimensionamento de seu sistema existente. Então, querem adiar as atualizações do modelo para priorizar a melhoria da plataforma de ML.

Gerente
: Quer maximizar as margens de lucro. Para tal, talvez seja necessário dispensar a equipe de ML.[14]

[14] Não é incomum que as equipes de ML e ciência de dados estejam entre as primeiras durante uma demissão em massa de uma empresa, como foi relatado na IBM (disponível em: *https://oreil.ly/AfUB5*), Uber (disponível em: *https://oreil.ly/t0QpY*), Airbnb (disponível em: *https://oreil.ly/ q4M4E*). Veja também a análise de Sejuti Das: "How Data Scientists Are Also Susceptible to the Layoffs Amid Crisis", *Analytics India Magazine*, 21 de maio de 2020. Disponível em: *https://oreil.ly/jobmz*.

"Recomendar os restaurantes em que os usuários têm maior probabilidade de clicar" e "recomendar os restaurantes que gerarão mais dinheiro ao aplicativo" são dois objetivos diferentes e, na seção "Desacoplando objetivos" do Capítulo 2, analisaremos como desenvolver um sistema de ML que satisfaça objetivos diferentes. Spoiler: desenvolveremos um modelo para cada objetivo e combinaremos suas predições.

Por ora, vamos imaginar que temos dois modelos diferentes. O modelo A é o que recomenda os restaurantes em que os usuários têm maior probabilidade de clicar, e o modelo B é o que recomenda os restaurantes que gerarão mais dinheiro ao aplicativo. A e B podem ser modelos bem diferentes. Qual modelo deve ser implementado para os usuários? Para dificultar ainda mais essa decisão, nem A nem B satisfazem o requisito estabelecido pela equipe do produto: os modelos não podem retornar recomendações do restaurante em menos de 100 milissegundos. Quando se trata de desenvolver um projeto de ML, é importante que os engenheiros de ML compreendam os requisitos de todas as partes interessadas envolvidas e o grau de exigência desses requisitos. Por exemplo, se retornar recomendações dentro de 100 milissegundos for um requisito obrigatório — e a empresa descobrir que, se seu modelo levar mais de 100 milissegundos para recomendar restaurantes, 10% dos usuários perderão a paciência e fecharão o app — então nem o modelo A nem o B funcionarão. No entanto, se for apenas um requisito opcional, talvez você ainda queira considerar o modelo A ou o modelo B.

Um das razões pelas quais os projetos bem-sucedidos de pesquisa nem sempre podem ser usados em produção é que ambos têm requisitos diferentes. Por exemplo, o ensemble é uma técnica popular entre os vencedores de muitas competições de ML, incluindo o famoso Prêmio Netflix de US$1 milhão, e ainda não é amplamente usado em produção. O ensemble combina "múltiplos algoritmos de aprendizagem para obter melhor desempenho preditivo do que poderia ser obtido a partir de qualquer um dos algoritmos de aprendizagem constituintes sozinhos".[15] Mesmo que possa fornecer uma melhoria sucinta de desempenho ao seu sistema de ML, o ensemble costuma inviabilizar a utilidade de um sistema em produção: o sistema fica mais lento para fazer predições ou fica mais difícil de interpretar os resultados. Falaremos mais sobre o ensemble na seção "Ensembles" do Capítulo 6.

Crítica às Classificações de ML

Nos últimos anos, as classificações de machine learning têm sido alvo de muitas críticas, tanto em competições como o Kaggle e classificação de pesquisa como o ImageNet ou o GLUE. O argumento óbvio é que nessas competições muitas das etapas difíceis necessárias para construir sistemas

[15] Wikipédia, s.v. "Ensemble learning". Disponível em: *https://oreil.ly/5qkgp*.