**Implantação**

*Não planejar um projeto em torno de uma estratégia de implantação é como organizar um jantar sem saber quantos convidados comparecerão. Você estará desperdiçando dinheiro ou arruinando experiências.*

Talvez a parte mais confusa e complexa do trabalho do projeto de ML para equipes mais novas seja como construir uma estratégia de implantação econômica. Se tiver pouca potência, a qualidade da previsão não importa (já que a infraestrutura não pode atender adequadamente às previsões). Se for sobrecarregado, você estará efetivamente queimando dinheiro em infraestrutura e complexidade não utilizadas. Como exemplo, vejamos um problema de otimização de estoque para uma empresa de fast-food. A equipe do DS tem sido bastante bem-sucedida no fornecimento de previsões para gerenciamento de estoque em agrupamentos em nível de região há anos, executando grandes previsões em lote para as demandas diárias de contagens esperadas de clientes em nível semanal e enviando previsões como extratos em massa a cada semana. Até este ponto, a equipe de DS estava acostumada com uma arquitetura de ML que se parece efetivamente com a mostrada na Figura 1.11. Essa arquitetura relativamente padrão para fornecer previsões de lotes programados concentra-se na exposição dos resultados à equipe interna de análise, que fornece orientação sobre as quantidades de materiais a serem encomendados. Essa arquitetura de serviço de previsão não é particularmente complexa e é um paradigma com o qual os membros da equipe de DS estão familiarizados. Com a natureza síncrona programada do design, bem como a grande quantidade de tempo entre o retreinamento subsequente e a inferência, a sofisticação geral da sua pilha de tecnologia não precisa ser particularmente alta (o que é uma coisa boa; veja a barra lateral a seguir). . À medida que a empresa percebe os benefícios da modelagem preditiva ao longo do tempo com essas abordagens em lote, sua confiança na equipe de DS aumenta. Quando surge uma nova oportunidade de negócio que exige previsão de estoque quase em tempo real por loja, os executivos da empresa pedem à equipe de DS que forneça uma solução para esse caso de uso. Os membros da equipe de ML entendem que sua arquitetura padrão de fornecimento de previsão não funcionará para este projeto. Eles precisam construir uma interface de programação de aplicativos (API) REST para os dados previstos para suportar o volume de solicitações e a frequência de atualização da previsão. Para se adaptar ao nível granular de uma previsão de estoque por loja (e à volatilidade envolvida nisso), a equipe sabe que precisa regenerar as previsões com frequência ao longo do dia. Munidos desses requisitos, eles recorrem à ajuda de alguns engenheiros de software da empresa e desenvolvem a solução. Somente após a primeira semana de entrada em operação é que a empresa percebe que os custos de computação em nuvem da implementação são mais do que uma ordem de magnitude superiores às economias de custos observadas com o sistema de gerenciamento de inventário mais eficiente. A nova arquitetura, juntamente com modelos autoregressivos de média móvel integrada (ARIMA) necessários para resolver o problema, é mostrada na figura 1.12. Não demora muito para que o projeto seja cancelado e uma reformulação completa da arquitetura dessa implementação seja encomendada para manter os custos baixos. Esta é uma história que se repete continuamente em empresas que implementam ML para resolver problemas novos e interessantes (e para ser justo, um que eu pessoalmente causei três vezes em minha carreira). Sem focar na implantação e no atendimento no início de um projeto, o risco de construir uma solução com engenharia insuficiente — que não atenda ao acordo de nível de serviço (SLA) ou às necessidades de volume de tráfego — ou com engenharia excessiva — excede as especificações técnicas a um custo inaceitavelmente elevado – é elevado. A Figura 1.13 mostra alguns (não todos, por qualquer esforço de imaginação) elementos a serem considerados em relação ao fornecimento de resultados de previsão e aos custos associados aos extremos das faixas desses paradigmas. Pode não parecer particularmente excitante ou importante pensar no custo quando nos deparamos com um problema novo para resolver de forma inteligente com um algoritmo. Embora a equipe do DS possa não estar pensando no custo total de propriedade de um projeto específico, tenha certeza de que os executivos estão. Ao avaliar estas considerações suficientemente cedo no processo de construção de um projecto, podem ser realizadas análises para determinar se o projecto vale a pena. Afinal, é melhor cancelar um projeto na primeira semana de planejamento do que desligar um serviço de produção depois de passar meses construindo-o. A única maneira de saber se uma arquitetura relativamente cara vale o custo de sua operação, entretanto, é medindo e avaliando seu impacto nos negócios.

**Uma breve nota sobre a arquitetura simplista**

No mundo do ML, sempre busca o design mais simples possível ao construir uma arquitetura. Se o projeto exigir uma periodicidade de inferência de uma semana, use um processo em lote (não streaming em tempo real). Se os volumes de dados estiverem nos megabytes, use um banco de dados e uma máquina virtual simples (não um cluster do Apache Spark de 25 nós). Se o tempo de execução do treinamento for medido em minutos, atenha-se às CPUs (não às GPUs). Usar arquitetura, plataformas e tecnologia complexas simplesmente para usá-los criará uma condição da qual você inevitavelmente se arrependerá, pois introduz uma complexidade desnecessária em uma solução já complexa. Com cada nova complexidade introduzida, aumentam as chances de que algo seja quebrado (geralmente de uma maneira espetacularmente complexa). Manter a tecnologia, a pilha e a arquitetura tão simples quanto são necessárias para resolver as necessidades de negócios iminentes do projeto é sempre uma prática recomendada para oferecer uma solução consistente, confiável e eficaz para uma empresa.

