**Abraçar e esperar mudanças**

É de extrema importância, não apenas na experimentação e na direção do projeto, mas também no desenvolvimento do projeto, estar preparado e esperar que mudanças inevitáveis ocorram. Em quase todos os projetos de ML em que trabalhei, os objetivos definidos no início do projeto nunca foram exatamente o que foi construído no final. Isso se aplica a tudo, desde tecnologias específicas, linguagens de desenvolvimento e algoritmos, até suposições ou expectativas sobre os dados – e, às vezes, até mesmo ao uso de ML para resolver o problema em primeiro lugar (um painel de agregação simples para ajudar as pessoas a resolver um problema). problema de forma mais eficiente, por exemplo). Se você planejar a mudança inevitável, poderá ajudar a focar no objetivo mais importante de todo o trabalho do DS: resolver problemas. Essa expectativa também pode ajudar a remover o foco dos elementos insignificantes (qual algoritmo sofisticado, nova tecnologia interessante ou estrutura incrivelmente poderosa para desenvolver uma solução). Sem esperar ou permitir que mudanças aconteçam, podem ser tomadas decisões sobre a implementação de um projeto que tornem sua modificação incrivelmente desafiadora (ou impossível) sem uma reescrita completa de todo o trabalho realizado até aquele ponto. Ao pensar em como a direção do projeto poderia mudar, o trabalho é forçado a adotar um formato modular de peças de funcionalidade fracamente acopladas, reduzindo o impacto de um pivô direcional em outras partes do trabalho já concluído. O Agile adota esse conceito de design fracamente acoplado e um forte foco na construção de novas funcionalidades em sprints iterativos para que, mesmo diante de requisitos dinâmicos e mutáveis, o código ainda funcione. Ao aplicar este paradigma ao trabalho de ML, mudanças abruptas e até tardias podem ser relativamente simplificadas – dentro do razoável, é claro. (Mudar de um algoritmo baseado em árvore para um algoritmo de aprendizagem profunda não pode acontecer em um sprint de duas semanas.) Embora simplificado, isso não garante simplicidade. O fato é que antecipar mudanças e construir uma arquitetura de projeto que suporte rápida iteração e modificação tornará o processo de desenvolvimento muito mais fácil.