O que é MLOps?

Last updated by | Gabriel Pehls | 6 de jul. de 2023 at 23:43 BRT

Devops?

Antes de falar de MLOps, precisamos definir o que é DevOps:

> Um composto de Dev (desenvolvimento) e Ops (operações), o DevOps é a união de pessoas, processos e tecnologias para fornecer continuamente valor aos clientes.
>
> O que o DevOps significa para as equipes? O DevOps permite que funções anteriormente isoladas – desenvolvimento, operações de TI, engenharia da qualidade e segurança – atuem de forma coordenada e colaborativa para gerar produtos melhores e mais confiáveis. Ao adotar uma cultura de DevOps em conjunto com as práticas e ferramentas de DevOps, as equipes ganham a capacidade de responder melhor às necessidades dos clientes, aumentar a confiança nos aplicativos que constroem e cumprir as metas empresariais mais rapidamente.

[O que é o DevOps? DevOps explicado | Microsoft Azure]

Quando modelos de ML (e seus processos) encontram o ambiente de produção, também encontramos os mesmos problemas que o time de DevOps se propõe a resolver, ao integrar e operar sistemas em ambientes produtivos.

DevOps vs MLOps

Basicamente, existem dois conceitos principais nas esteiras de desenvolvimento de sistemas de software utilizados (como core) em DevOps, a **Integração Contínua (CI)** e a **Entrega Contínua (CD)**, e podemos comparar, de forma direta, um sistema de ML com um sistema de software, aplicando os mesmos conceitos a este segundo sistema, de modo a garantir que sistemas de ML sejam criados e operados de maneira confiável e em escala.
No entanto, sistemas de ML possuem algumas características peculiares, como:

- Habilidades da equipe, dado que geralmente a equipe inclui pesquisadores/cientistas de dados que não necessariamente são engenheiros de software experientes, capazes de criar serviços de classe de produção;
- Desenvolvimento, dado que ML é, essencialmente, experimental, com testes de diferentes recursos, algoritmos, técnicas de modelagens e configurações de parâmetros para encontrar a melhor solução para o problema sendo trabalhado, o mais rápido possível, buscando acompanhar o que funcionou/não funcionou e manter a reprodutibilidade enquanto maximiza a reutilização do código;
- Testes: sistemas de ML exigem testes específicos, como validações de dados, avaliação de qualidade de modelo treinado, identificar variações nos dados, e validar o modelo em teste, além dos tradicionais testes de unidade e integração, que devem ser aplicados a toda estrutura do sistema de ML, da adquirência do dado até a validação do modelo aplicado;
- Implantação: um sistema de ML não se limita a aplicar um modelo treinado off-line como um serviço de previsão. O mesmo pode exigir um pipeline complexo para treinar e implantar automaticamente o modelo, exigindo que os passos feitos manualmente antes da implantação por cientistas de dados sejam automatizados, em todas as etapas do treinamento e teste dos modelos;
- Produção: modelos de ML podem ter desempenho reduzido, tanto pela codificação, quanto pela mudança contínua do perfil dos dados e pela mudança da resposta esperada do modelo no tempo (drift dos dados de entrada / do modelo de ML), ou seja, podemos ter uma degradação do sistema de ML mais rápida do que em um sistema de software, e isto precisa ser levado em conta. O rastreamento

das estatísticas resumidas dos dados, e do desempenho dos modelos precisam ser levados, de modo a gerar "gatilhos" em resposta a tais mudanças.

De forma geral, o ML e outros sistemas de software são semelhantes na integração contínua de controle de origem, teste de unidade, teste de integração e entrega contínua do módulo de software ou do pacote. No entanto, no ML, existem algumas diferenças notáveis:

- A CI não se trata mais de apenas testar e validar código e componentes, mas também testar e validar dados, esquemas de dados e modelos.
- O CD não é mais sobre um único pacote de software ou serviço, mas um sistema (um pipeline de treinamento de ML) que deve implantar automaticamente outro serviço (serviço de predição de modelo).
- O TC é uma nova propriedade, exclusiva para sistemas de ML, que se preocupa em treinar e exibir automaticamente os modelos.