Desafios na adoção de MLOps por time DevOps - projeto de co-desenvolvimento entre Governo e Academia para a introdução de e-gov 3.0

Carla Rocha [1]

[1]Faculdade do Gama - Universidade de Brasília (UnB))
Brasília, Brazil.

caguiar@unb.br

Abstract. *A adoção de sistemas de machine learning tem acelerado nos últimos anos, pela disponibilização de ferramentas, frameworks e bibliotecas. Enquanto a implantação de um sistema de machine learning é facilitado, os desafios relacionados à manutenção e evolução desses modelos tem sido pouco falado. Pesquisas e surveys mostram que engenheiros ainda tem dificuldade de operacionalizar e padronizar os processos para o deploy contínuo. Nesse contexto, relato minha experiência coordenando a adoção de MLOps durante uma parceria sem precedentes entre o governo e academia por 24 meses para a introdução de serviços e-gov 3.0. A partir da análise post-mortem dos dados dos projetos de sistemas de Machine learning desenvolvidos, um chatbot e um serviço de recomendação, levantei um conjunto de lições aprendidas e melhores práticas para a adoção bem sucedida de MLOps de uma equipe que já madura na cultura DevOps.*

1. Introdução

Sistemas de Machine Learning integram capabilidades de Inteligência Artificial em softwares e serviços [Amershi et al. 2019a], e tem se tornado popular na indústria de Software. Um dos grandes desafios na implantação desses sistemas é garantir a entrega contínua de sistemas de Machine Learning (ML), pois impõe mais desafios a uma organização que software tradicionais [Amershi et al. 2019b], tais como serviços web e aplicações móveis. Em sistemas de machine learning há mais artefatos a serem gerenciado além do código, como o modelo e os dados de treinamento. Novos papéis fazerem parte do time, como cientista de dados e analista de dados. Com isso, foi cunhado o termo MLOps, que aplica práticas ágeis, lean e DevOps para o contexto de sistemas de machine learning.

Alguns problemas resolvidos com sistemas de Machine learning são: assistentes pessoais, sistemas de recomendações/sugestões, detecção automática de fraude, serviços de mídia social, automação de processos, automação de análises e tomada de decisão. Todos esses sistemas são compostos por código fonte, modelos de machine learning, e dados usados para treinamento periódico desses modelos. A complexidade de manter um modelo de machine learning em produção estável e correto surge do fato que novos dados de treinamento e novas features/atualizações no modelo requer o treinamento do modelo completo, o que pode fazer com que resultados já validados do modelo alterem com novos treinamentos. É mais difícil manter as fronteiras entre os componentes de machine

learning modular com código fonte [Amershi et al. 2019b]. Algumas limitações do uso de DevOps em sistemas ML são: versionamento único de código e modelo, orquestração de ambientes para diversos modelos sendo testados, expectativas irreais de gestores de negócio em relação aos benefícios do sistema ML, pressão para colocar em produção um modelo ML ainda não calibrado para o contexto real, entre outros.

Coordenei um projeto de parceria para o co-desenvolvimento de dois sistemas de machine learning com a Secretaria Especial da Cultura. Essa parceria pioneira no desenvolvimento de serviços de governo digital 3.0 iniciou um outubro 2017, pouco depois do artigo pioneiro da google [Sculley et al. 2015] no tema, mas ainda no início do movimento MLOps, com o termo ainda não adotado pelos práticos, e soluções para MLOps ainda nos seus estágios iniciais, não existentes ou instável. Sem comunidade local ou global de MLOps, esse projeto foi uma oportunidade de fazermos pesquisa pioneira sobre desafios de adoção de MLOps por equipes DevOps.

Foram implementados dois sistemas de machine learning, o projeto do chatbot TAIS e o projeto do sistema de recomendações para o SALIC. Tínhamos no laboratório experiência prévia tanto em co-desenvolvimento com agências governamentais quanto em práticas e ferramentas/automações DevOps [Wen et al. 2020, Leite et al. 2019].

Neste trabalho, compartilho as principais lições aprendidas na adoção de MLOps por uma equipe madura em DevOps. Registro os principais riscos na gestão de projeto para a entrega contínua de modelos de machine learning. O cenário apresentado é similar à maioria das organizações que conhecem os benefícios do DevOps mas ainda não aplicaram esses conceitos em projetos de sistemas de machine learning. Dessa forma, compartilho com a comunidade acadêmica um entendimento sobre as questões críticas e práticas, ainda muito pouco documentada, na adoção de MLOps, que podem beneficiar práticos, tanto desenvolvedores quanto gestores, e outras iniciativas de adoção similares.

2. Projeto Ecossistemas de Software Livre

Os projetos que gerenciei, usado como estudo de caso neste trabalho, foram o desenvolvimento de um chatbot e um sistema de recomendação. Os sistemas ML desenvolvidos nessa parceria foram: um chatbot FAQ, o projeto TAIS[1] (Tecnologia de Aprendizado Interativo do Salic), e o projeto SALICML [2] (Sistema de Apoio às Leis de Incentivo à Cultura), um sistema de recomendação. TAIS é um FAQ chatbot para responder a questões do usuário sobre a Lei de Incentivo à Cultura, conhecida como Lei Rouanet. Adotamos o framework Open Source de chatbot orientado a modelos de machine learning Rasa [3], ainda em suas primeiras releases. O SalicML é um microsserviço da plataforma legada da secretaria SALIC[4], e ele usa os mais de 25.0000 (vinte e cinco mil) projetos já submetidos na plataforma para identificar outliers, anomalias e indicadores que tornam um projeto complexo para ser avaliado.

O desenvolvimento desses sistemas ML ocorreu no Laboratório Avançado de Pesquisa, Produção e Inovação em Software da UnB (LAPPIS/UnB) com ampla experiência no desenvolvimento de projeto de software livre, cultura DevOps, e foco em

[1]https://github.com/lappis-unb/tais

[2]https://github.com/lappis-unb/salic-ml

[3]https://rasa.com

[4]http://salic.cultura.gov.br/autenticacao/index/index

entrega contínua [Wen et al. 2020].

3. Desafios na adoção de MLOps

Ambos projetos sofreram diversas mudanças para viabilizar MLOps. Inicialmente, tratamos ambos projetos como software tradicional e aplicamos conceitos de DevOps: a arquitetura de ambos o modelo de machine learning era acoplada ao código fonte, tendo assim um único controle de versão para ambos código, modelo e dataset de treinamento; os estágios do pipeline de deploy contínuo continham somente testes relacionados ao código fonte; gestão manual dos modelos e dos datasets. Com as dificuldades em manter e evoluir esses modelos continuamente e estável no ambiente de produção, diversas alterações relacionadas ao que hoje sabemos que é o foram realizadas. Para o chatbot TAIS, a cada nova inserção de conteúdo (tanto adição/edição de intenções ou exemplos de diálogos), todo o modelo era treinado. Em um pipeline de entrega contínua, diversas pessoas trabalhavam em paralelo tanto no conteúdo do chatbot quanto em modelos e hiperparâmetros mais robustos de conversação, e tínhamos alguns ambientes de teste/homologação. Inicialmente, testamos somente as novas funcionalidade, ou as mudanças realizadas. Um problema recorrente em chatbots é a similaridade na modelagem de intenções, o que faz com que o chatbot classifique de forma errada frases do usuário, e responda de forma incorreta. Como validamos inicialmente somente a parte alterada do chatbot, tivemos situações em que percebemos que uma nova intenção atrapalhava a classificação de uma outra intenção já validada somente ambiente de produção. Isso fazia com que a resposta do chatbot não fosse como esperado, causando uma insatisfação do usuário e muitas vezes uma baixa taxa de retenção de usuários. Isso é um exemplo de como usar práticas de DevOps nesse contexto não garante uma alta confiabilidade do modelos implantados em produção, uma vez que esses modelos apresentavam alta acurácia no treinamento, apesar dos erros evidentes. Abaixo descrevo os principais desafios e lições aprendidas na adoção do MLOps nesses dois projetos.

- *Gestão de projeto* - um dos maiores desafios foi a gestão de expectativa dos gestores de negócio do ministério. Demonstrações de ”caminhos felizes” na TAIS mascararam problemas dos modelos e geraram expectativas irreais dos gestores de negócio.
- *Abstrair a complexidade para os cientistas de dados* - No SalicML percebemos a necessidade de abstrair conceitos de engenharia de software e trabalhar nas tarefas de ciência de dados para otimizar o workflow dos cientistas de dados [López García et al. 2020].
- *Isole e versione seu modelo* - Após cada execução do pipeline, salve seu modelo em um registry adequado com versionamento para uso posterior [de Lacerda and Aguiar 2019]. Essa separação reduz a sobrecarga de pipelines pois permite o reuso de modelos sempre que necessário.
- *Uma abordagem holística é complexa* - alguns stakeholders (principalmente de negócio) podem não estar disponíveis continuamente para validar e testar hipóteses.
- *O processo de treinamento é não homogêneo* - Entender a dependência dos dados pode reduzir e otimizar o pipeline, e também permite a paralelização dos cálculos.
- *Não tema mudanças arquiteturais* - Mudanças arquiteturais podem ser necessárias para otimizar tanto os procedimentos no CI/CD e quanto o fluxo de desenvolvimento, a fim de isolar os modelos e dados de treinamento do código.

- *Ative tarefas do CI/CD somente quando necessário* - Use imagens base de dockers que são atualizadas por agendamento ou com gatilhos específicos.
- *Não use em excesso ferramentas de automação* - A adoção de ferramentas aumenta a complexidade de manutenção do software. Entenda primeiro os gargalos do processo.
- *Escopo dos primeiros projetos de sistemas de machine learning podem ser difíceis de definir* - Comece por áreas de negócio de menor risco para capacitar os times.
- *Os dados impactam enormemente a capacidade de entrega contínua de sistemas de machine learning* - Caso os dados sejam do sistema, como o SalicML, pense em utilizar DataOps para criar pipelines de dados independente e versionado dos dados usados no treinamento.

MLOps vai se tornar uma disciplina essencial para as organizações que desejam ser competitivas. Todo desenvolvimento, artefatos e documentação produzidos nesse projeto podem ser acessados em `http://github.com/lappis-unb`.

References

Amershi, S., Begel, A., Bird, C., DeLine, R., Gall, H., Kamar, E., Nagappan, N., Nushi, B., and Zimmermann, T. (2019a). Software engineering for machine learning: A case study. In *Proceedings of the 41st International Conference on Software Engineering: Software Engineering in Practice*, ICSE-SEIP '19, page 291–300. IEEE Press.

Amershi, S., Begel, A., Bird, C., Deline, R., Gall, H., Kamar, E., Nagappan, N., Nushi, B., and Zimmermann, T. (2019b). Software engineering for machine learning: A case study. pages 291–300.

de Lacerda, A. R. T. and Aguiar, C. S. R. (2019). Floss faq chatbot project reuse: How to allow nonexperts to develop a chatbot. In *Proceedings of the 15th International Symposium on Open Collaboration*, OpenSym '19, New York, NY, USA. Association for Computing Machinery.

Leite, L., Rocha, C., Kon, F., Milojicic, D., and Meirelles, P. (2019). A survey of devops concepts and challenges. *ACM Comput. Surv.*, 52(6).

López García, , De Lucas, J. M., Antonacci, M., Zu Castell, W., David, M., Hardt, M., Lloret Iglesias, L., Moltó, G., Plociennik, M., Tran, V., Alic, A. S., Caballer, M., Plasencia, I. C., Costantini, A., Dlugolinsky, S., Duma, D. C., Donvito, G., Gomes, J., Heredia Cacha, I., Ito, K., Kozlov, V. Y., Nguyen, G., Orviz Fernández, P., Šustr, Z., and Wolniewicz, P. (2020). A cloud-based framework for machine learning workloads and applications. *IEEE Access*, 8:18681–18692.

Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., Chaudhary, V., Young, M., Crespo, J.-F., and Dennison, D. (2015). Hidden technical debt in machine learning systems. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, page 2503–2511, Cambridge, MA, USA. MIT Press.

Wen, M., Siqueira, R., Lago, N., Camarinha, D., Terceiro, A., Kon, F., and Meirelles, P. (2020). Leading successful government-academia collaborations using floss and agile values. *Journal of Systems and Software*, 164:110548.