Podcast

Disciplina: Linguagens de programação para ciência de dados
Título do tema: Introdução: por que Python, Spark e Hadoop?
Autoria: Yuri Vasconcelos de Almeida Sá
Leitura crítica: Henrique Salustiano Silva

Abertura:

Olá! No podcast de hoje vamos falar sobre a importância de utilizar as ferramentas o mais próximo possível da produção.

Em ciências de dados temos uma infinidade de ferramentas para desenvolver um código ou modelo e assim executá-lo em produção. Porém uma situação bastante comum é o modelo ser desenvolvido em uma plataforma diferente da plataforma de produção, ainda que utilizem a mesma linguagem, os Workflows podem ser diferentes, gerando alguns problemas de incompatibilidade entre o protótipo e a rotina de execução.

O que costuma acontecer é gerenciamento de memória ser muito diferente entre a prototipagem e a produção, e memória é, atualmente, o item mais caro em uma infraestrutura, seja ela de servidores físicos ou distribuídos na nuvem.

Há também ainda problemas estruturais da arquitetura das bibliotecas que podem ou não ser compatíveis com execução em larga escala (sendo o *Multithreading* o pior vilão).

A melhor solução para estes problemas é o modelo ser desenvolvido já na plataforma e infraestrutura utilizada para produção. Hoje em dia as principais plataformas estão disponíveis de forma gratuita e muito acessível. Até mesmo empacotadas em Máquinas virtuais ou Containers de fácil instalação e utilização em PCs comuns. Existem guias e tutoriais para a correta utilização e instalação destes pacotes.

Este tipo de abordagem pode até gerar um certo desconforto no cientista de dados em um primeiro momento, afinal estamos todos acostumados com nossas ferramentas e modo de trabalho, mas evita algumas situações muito chatas, por exemplo:

O puxão de orelha do pessoal da infra de produção, sempre são e-mails muito chatos expondo o problema.

Outra é cliente ou chefe (caso você trabalhe *in-house*) reclamando que a conta de servidor está ficando cara demais. Essa sim é perigosa.

Em cenários menores onde nós temos que fazer o papel de Analista, Estatístico, Modelador, Cientista de Dados, Programador e Engenheiro é pior ainda, pois se a aplicação não está otimizada o trabalho aumenta e geralmente o pagamento não!

Caso essa adaptação seja de difícil adoção ou realmente impossível de implementar em hardware não especializado, procure guias e tutoriais específicos para o seu código e sua implementação na infraestrutura alvo. Deixe comentários ao longo do projeto e código, guiando assim a implementação real do modelo. Aliás, esta é uma boa prática que deve ser seguida em todos os projetos, pois existem projetos com elevado sigilo, onde a modelagem é anonimizada e a infraestrutura de produção também, ficando então o projeto dependente da modelagem às cegas, se você comentou tudo e guiou a implementação para diversas plataformas durante o modelo, ponto para você.

Procure na documentação meios de executar este código em diferentes cenários mesmo que não seja da sua prática e, deixe disponível para o pessoal da engenharia junto com o código, pois qualquer tempo economizado entre a finalização do modelo e a produção é vital para a melhoria contínua do modelo.

O ambiente de trabalho em ciência de dados é extremamente dinâmico e ágil e exige que todos os pontos da cadeia produtiva estejam alinhados e coesos para a evolução da prática como um todo, então se o time trabalha para que a operação flua bem, estamos um passo mais perto do sucesso e prontos para o próximo desafio.

Fechamento:

Este foi nosso podcast de hoje! Até a próxima!