

UNIVERSIDADE DE BRASÍLIA
Faculdade do Gama

Aprendizado de Máquina

Mini Trabalho 8

Lançamento, monitoramento e manutenção do sistema

Grupo - 1

Ana Clara Barbosa Borges
André Emanuel Bispo da Silva
Artur Handow Krauspenhar
Gabriel Moura dos Santos
João Artur Leles Ferreira Pinheiro
João Pedro Anacleto Ferreira Machado

Brasília, DF

2025

Introdução

Durante os últimos meses desenvolvemos, a partir dos mini trabalhos, uma solução para a identificação de transações fraudulentas. Nesse período decidimos o tema do projeto, escolhemos uma base de dados, exploramos ela, treinamos diferentes modelos, selecionando os que geraram os melhores resultados e aperfeiçoando eles.

Tendo os nossos modelos treinados, nessa última entrega colocamos eles em produção para que possam ser acessados pelos usuários. Este documento tem como objetivo detalhar o processo da preparação para a produção, assim como um plano de monitoramento e estratégias de manutenção e atualização.

Preparação para a produção

Pensando em escolher a melhor tecnologia para colocar o modelo em produção e construir a interface da plataforma, o nosso grupo debateu sobre algumas possibilidades. Segue as tecnologias que foram consideradas:

- Streamlit
 - Um gerador de plataformas web com uma API simplificada em Python e provedor de hospedagem. Contém várias facilidades para criação de aplicações baseadas em dados e criação de interfaces web através de sua API.
 - Consideramos porque é uma ferramenta que muito simplifica o trabalho da criação de um sistema web que economizaria tempo para o desenvolvimento de uma interface simples o que evitaria grandes esforços com o front-end, que não é tão relevante para a disciplina.
 - Era conhecido por apenas 2 membros da equipe.
- Flask com templates
 - Uma micro framework de web para python caracterizada pelo seu minimalismo, modularidade e flexibilidade. Sua arquitetura é baseada em extensões
 - Consideramos essa integração por ser uma framework extremamente minimalista, utilizando apenas as extensões necessárias seria possível fazer uma aplicação de produção com relativa facilidade. Utilizando templates seria possível economizar mais esforço, ao custo de um pouco de dinamismo.
- FastAPI + React
 - Conjunto de frameworks para criação de APIs e Python e frontends dinâmicos com javascript/typescript respectivamente.
- Figma
 - Uma aplicação colaborativa para design de interfaces gráficas muito utilizado para prototipagem de interfaces web.
 - Consideramos para fazer um protótipo evitando retrabalho se fizéssemos a interface web por si só.

- Caso usássemos Streamlit julgamos não ser necessário porque o Streamlit traz facilidades na hora de mudar a interface.

Após conversarmos sobre as tecnologias, optamos por utilizar o Streamlit, excluindo a possibilidade de usar outras tecnologias de backend em conjunto com o figma. A nossa escolha foi fortemente baseada na facilidade e agilidade de desenvolvimento devido à componentes de interface já existentes na plataforma e facilidade de criação de lógica da web, agilizando o desenvolvimento da plataforma. Contudo, é importante ressaltar que o Streamlit foi escolhido como uma solução de hospedagem temporária e, para o futuro, a intenção é migrar a aplicação para um ambiente de produção mais definitivo.

Plano de Monitoramento

Após o lançamento do modelo treinado devemos monitorar ele, garantindo que ele continue a funcionar da maneira esperada ao longo do tempo. Pensando nisso, a forma ideal de aplicar esse monitoramento seria:

1. Coletar os dados preenchidos pelos usuários, desde que eles permitam essa coleta de dados, assim como a label atribuída à transação.
2. Fazer um script de monitoramento que analisa os dados periodicamente para calcular as métricas de performance, no nosso caso em especial o F1Score.
3. Pedir feedbacks aos usuários para sabermos se a previsão acertou sobre a transação ser fraudulenta ou não.
4. Adaptar periodicamente levando em consideração as métricas e os feedbacks dos usuários.

Esse fluxo ainda não foi implementado, mas levamos em consideração que seria a forma ideal de fazer esse monitoramento.

Estratégias de Manutenção e Atualização

Para a manutenção automatizada do sistema iremos, baseado nos dados do nosso plano de monitoramento, retreinar o modelo com base nos novos dados assim que for julgado necessário. O processo de atualização do modelo poderá ser facilitado utilizando uma arquitetura distribuída, com a separação do sistema web e serviços de treino e uso do modelo, permitindo automatização do processo de treino com os dados novos e atualização do serviço.

Com base nos dados armazenados, será possível retreinar o modelo em uma máquina com maior capacidade de processamento necessário, que disponibilizaria o modelo para os nós de previsão/uso, que receberiam uma notificação da atualização por meio de protocolos de comunicação como webhooks.