Universidade Estadual Paulista - UNESP, Faculdade de Ciências

Curso: Inteligência Artificial

Clayton Pereira

Deploy vs. Dashboard

O que é Deploy?

Deploy (ou implantação) é o processo de colocar um sistema, aplicativo ou site em produção, tornando-o disponível para uso. No contexto de ciência de dados e desenvolvimento de software, deploy significa mover o código de um ambiente de desenvolvimento (onde você cria e testa) para um ambiente de produção (onde os usuários finais podem acessar e interagir com a aplicação).

Tipos de Deploy:

1. Deploy Local: Quando você executa um aplicativo em sua própria máquina ou em uma rede local.

2. Deploy na Nuvem: Quando o aplicativo é implantado em um servidor remoto, como AWS, Google Cloud, Heroku ou Streamlit Cloud.

3. Deploy Contínuo: Integração com sistemas de automação (como CI/CD) para que o código seja automaticamente implantado assim que novas mudanças sejam enviadas (via Git, por exemplo).

Quando o Deploy deve ser feito?

1. Após o desenvolvimento: Quando o código está pronto e testado, ele deve ser colocado em produção para que os usuários possam acessá-lo.

2. Atualizações e melhorias: Cada vez que há uma nova funcionalidade, correção de bugs ou otimização no código, o deploy é necessário para aplicar essas mudanças no ambiente de produção.

3. Prototipagem: Se você deseja compartilhar um protótipo de aplicação, o deploy permite que outros usuários testem o aplicativo e forneçam feedback.

Boas Práticas de Deploy:

1. Automação: Sempre que possível, use sistemas de automação para garantir que o deploy seja feito de maneira consistente e eficiente (ex.: GitHub Actions, Jenkins).

2. Versionamento: Certifique-se de que as versões do código estejam bem documentadas para evitar conflitos ou problemas de compatibilidade entre diferentes ambientes.

3. Rollback: Mantenha uma estratégia de rollback para que, se algo der errado com o deploy, você possa retornar rapidamente a uma versão anterior.

4. Monitoramento: Depois do deploy, é fundamental monitorar o desempenho da aplicação em produção para garantir que tudo está funcionando corretamente e para detectar possíveis problemas.

5. Segurança: Certifique-se de que o ambiente de produção esteja seguro, com permissões adequadas, uso de HTTPS, autenticação, etc.

O que é um Dashboard?

Dashboard é uma interface visual que apresenta dados e métricas importantes de maneira condensada e interativa. Dashboards são usados para monitorar, analisar e visualizar informações de forma eficiente, facilitando a tomada de decisões.

Exemplos de Dashboards:

1. Dashboard Financeiro: Pode mostrar métricas como receita, despesas e lucros ao longo do tempo.

2. Dashboard de Vendas: Mostra o desempenho de vendas por produto, região ou período.

3. Dashboard de Ciência de Dados: Pode exibir visualizações de dados, como gráficos de dispersão, histogramas e tabelas interativas, permitindo a exploração de dados.

Quando um Dashboard deve ser feito?

1. Monitoramento contínuo: Quando há necessidade de monitorar indicadores de desempenho ou dados em tempo real, como vendas diárias ou visitas a um site.

2. Apresentação de resultados: Para comunicar insights de uma análise de dados de forma visual,

simplificando o entendimento de informações complexas.

3. Prototipagem e validação de modelos: Um dashboard pode ser usado para mostrar o desempenho de modelos preditivos ou fornecer uma maneira interativa de validar as predições.

Boas Práticas na Criação de Dashboards:

1. Simplicidade: Mantenha o design limpo e simples. Não sobrecarregue o dashboard com gráficos desnecessários.

2. Relevância: Apresente apenas os dados mais relevantes para o público-alvo. Cada gráfico ou visualização deve ter um propósito claro.

3. Interatividade: Dashboards interativos permitem que os usuários filtrem, ajustem e explorem os dados de diferentes maneiras, facilitando a análise exploratória.

4. Atualização automática: Sempre que possível, configure o dashboard para que os dados sejam atualizados automaticamente.

5. Responsividade: Garanta que o dashboard funcione bem em diferentes dispositivos (computadores, tablets, celulares).

Relacionando Deploy e Dashboards

O deploy de um dashboard é um processo em que você publica ou coloca o dashboard em produção para que os usuários finais possam acessá-lo. Muitas vezes, dashboards são usados por equipes de negócios ou outras partes interessadas para tomar decisões baseadas em dados.

Passos para criar e fazer deploy de um dashboard:

1. Desenvolver o Dashboard: Use ferramentas como Streamlit, Dash, Tableau ou Power BI para criar visualizações interativas dos seus dados.

2. Testar Localmente: Antes de fazer o deploy, teste o dashboard localmente para garantir que ele funciona conforme esperado e que as interações com os dados estão corretas.

3. Fazer o Deploy: Use plataformas como Streamlit Cloud, Heroku, ou AWS para colocar o dashboard em produção.

4. Monitorar o Uso: Após o deploy, acompanhe o uso e a performance do dashboard para garantir

que ele esteja funcionando corretamente e fornecendo o valor esperado.

Ferramentas Comuns para Deploy de Dashboards:

1. Streamlit Cloud: Simples de usar, especialmente para aplicativos Python interativos.

2. Dash: Baseado no Plotly, excelente para visualizações interativas.

3. Heroku: Plataforma em nuvem versátil, usada para hospedar aplicativos Python, Node.js, e mais.

4. Tableau Online: Popular para criar e compartilhar dashboards interativos sem a necessidade de codificação.

Vantagens e Desvantagens: Dash vs. Streamlit

Dash e Streamlit são duas das ferramentas mais populares para criar dashboards interativos em Python. Ambas oferecem formas rápidas de criar visualizações de dados interativas, mas têm características e casos de uso distintos.

Abaixo, vamos comparar as vantagens e desvantagens de cada uma.

Dash

Vantagens:
1. Altamente customizável e permite o controle total da interface do usuário.
2. Baseado no Plotly, possui gráficos interativos e visualizações de alta qualidade.
3. Integra-se bem com aplicativos de produção e permite uma arquitetura mais robusta.
4. Excelente para aplicações empresariais de longo prazo.

Desvantagens:
1. Requer mais conhecimento de front-end e callbacks, o que pode aumentar a complexidade.
2. Curva de aprendizado maior em comparação ao Streamlit.
3. Menos adequado para prototipagem rápida.

Streamlit

Vantagens:
1. Simplicidade: muito fácil de usar, basta adicionar widgets e gráficos com poucas linhas de

código.
2. Rápido para prototipagem e experimentação. Ideal para Data Scientists.
3. Permite criar dashboards interativos com um esforço mínimo.
4. Ambiente de desenvolvimento extremamente rápido: rodar um app Streamlit é quase instantâneo.

Desvantagens:
1. Menos flexível do que o Dash em termos de customização de design e layout.
2. Não é tão adequado para aplicações de produção robustas e complexas.
3. Foco em simplicidade pode limitar o controle detalhado da interface.

Comparação Direta: Dash vs. Streamlit

1. Complexidade: Dash oferece mais controle e flexibilidade, mas tem uma curva de aprendizado maior. Streamlit é muito mais fácil de aprender e usar.

2. Uso em produção: Dash é mais adequado para aplicativos de longo prazo e em ambientes de produção, enquanto o Streamlit é mais usado para protótipos e projetos rápidos.

3. Desenvolvimento: O desenvolvimento em Streamlit é muito mais rápido, mas Dash oferece maior personalização para aplicativos mais detalhados.

Conclusão

Ambas as ferramentas têm seus méritos, e a escolha depende do objetivo do projeto. O Dash é ideal para projetos complexos e de longo prazo, enquanto o Streamlit brilha em prototipagem rápida e criação de dashboards interativos com o mínimo de esforço. Avalie a complexidade do projeto e o tempo disponível para escolher a melhor ferramenta para cada situação.

Construindo um Deploy do Streamlit no GitHub e Streamlit Cloud

Neste documento, você encontrará um passo a passo para realizar o deploy de um aplicativo Streamlit no GitHub e no Streamlit Cloud. Além disso, há uma lista de tarefas para os alunos, sugerindo diferentes datasets que podem ser utilizados para construir dashboards interativos com o Streamlit.

Passo 1: Subir o código no GitHub

1. Crie um repositório no GitHub:
   - Vá para o GitHub (https://github.com) e crie um novo repositório.
   - Dê um nome ao repositório e escolha se deseja que ele seja público ou privado.

2. Inicialize seu repositório localmente:
   - No terminal, vá para o diretório do seu projeto e execute:
     git init
     git add .
     git commit -m "Primeiro commit"

3. Adicione o repositório remoto do GitHub e faça o push:
   - git remote add origin https://github.com/seu-usuario/nome-repositorio.git
   - git push -u origin master

Passo 2: Configurar o Deploy no Streamlit Cloud

1. Vá até o Streamlit Cloud (https://streamlit.io/cloud) e crie uma conta.

2. Conecte sua conta do GitHub ao Streamlit:
   - No Streamlit Cloud, clique em "New App" e selecione o repositório onde você subiu o projeto.

3. Escolha a branch correta e o arquivo principal:
   - Escolha a branch onde está o código (geralmente "main" ou "master").
   - Selecione o arquivo principal que roda o seu aplicativo (por exemplo, `app.py`).

4. Clique em "Deploy" e aguarde:
   - O Streamlit Cloud vai fazer o deploy automaticamente. Um link será gerado para que você possa compartilhar o aplicativo.

Lista de Tarefas para os Alunos

Vamos agora, fazer alguns exercícios, utilizando o Streamlit para desenvolver dashboards interativos com diferentes datasets.

Tarefa 1:
- Dataset: Titanic (https://www.kaggle.com/c/titanic)
- Crie um dashboard interativo para prever a sobrevivência dos passageiros, usando gráficos de dispersão, histogramas e tabelas interativas.

Tarefa 2:
- Dataset: Wine Quality (https://archive.ics.uci.edu/ml/datasets/Wine+Quality)
- Desenvolva um aplicativo que mostre as características dos vinhos (tinto e branco) e crie gráficos

interativos para explorar a relação entre a qualidade e as propriedades químicas.

Tarefa 3:
- Dataset: Global Energy Consumption (https://www.kaggle.com/datasets/pralabhpoudel/world-energy-consumption)
- Crie um aplicativo que permita comparar o consumo de energia de diferentes países da América do Sul, com gráficos de séries temporais e comparações entre energia renovável e não-renovável.

Tarefa 4:
- Dataset: Iris (https://archive.ics.uci.edu/ml/datasets/Iris)
- Construa um aplicativo que visualize as três espécies de flores e explore as correlações entre as características das sépalas e pétalas.

Tarefa 5:
- Dataset: MovieLens (https://grouplens.org/datasets/movielens/)
- Desenvolva um sistema de recomendação de filmes simples usando Streamlit, com base no histórico de avaliações dos usuários.

Para cada uma dessas tarefas, os alunos devem:
1. Fazer o download do dataset correspondente.
2. Criar um aplicativo Streamlit interativo com visualizações como gráficos de dispersão, histogramas, ou séries temporais.
3. Fazer o deploy do aplicativo no Streamlit Cloud e enviar o link para avaliação.