Case | Vaga Analista de Machine Learning com foco em IA

💡Contextualização do Case

Cenário

Você foi contratado para construir um chatbot que ajude a prever o sucesso de novos projetos com base em dados históricos de projetos anteriores. O objetivo é usar aprendizado de máquina para analisar os projetos passados e, a partir disso, fornecer previsões e recomendações sobre projetos futuros. O chatbot deve ser capaz de interagir com os usuários (gerentes de projeto ou membros da equipe) e combinar os dados fornecidos pelo usuário com informações de uma base de dados de usuários. A partir dessa combinação de dados, o chatbot rodará o modelo de machine learning tradicional para oferecer previsões e recomendações úteis.

🎯Objetivo

Você deve construir uma aplicação que inclua os seguintes componentes:

1. **Treinamento de modelo de ML tradicional** para prever o sucesso de novos projetos com base em dados históricos. O modelo deve ser implementado em Python.
2. **Deploy simplificado** do modelo em uma API que possa ser acessada por outras aplicações.
3. **Chatbot** que interage com o usuário, acessa a base de dados de usuários para coletar informações adicionais, combina essas informações com os dados fornecidos no chat e oferece previsões e recomendações sobre o sucesso do projeto.

🧩 Componentes do Projeto

1. Treinamento de Modelo de ML Tradicional

Utilize uma base de dados pública ou crie seu próprio banco de dados com informações sobre projetos anteriores, como duração, orçamento, equipe, e sucesso do projeto. O modelo de aprendizado de máquina deve ser implementado em Python, utilizando bibliotecas como **scikit-learn** ou **XGBoost** para treinamento de modelos de classificação (ex.: Random Forest ou Regressão Logística).
O objetivo do modelo é prever se o projeto será bem-sucedido (sucesso ou fracasso).

Entradas para o Modelo:
Variáveis como duração do projeto, orçamento, número de membros da equipe, recursos disponíveis, entre outras.

---

2. Deploy do Modelo

Crie uma API simples em Python, utilizando frameworks como **Flask** ou **FastAPI**, para expor o modelo treinado.
A API deve aceitar dados de um novo projeto (como duração, orçamento, número de membros da equipe) e retornar a previsão do modelo (ex.: "Probabilidade de sucesso: 85%").
O deploy simplificado da API deve permitir que ela seja acessada de forma simples para integrar com o chatbot.

---

3. Chatbot para Interação

Crie um chatbot que interage com o usuário (gerente de projeto ou membro da equipe). O chatbot deve:

- Fazer perguntas sobre o novo projeto (ex.: duração, orçamento, etc.).
- Acessar uma base de dados de usuários para coletar informações adicionais sobre o usuário que está interagindo (ex.: histórico de projetos passados, cargos, experiência, etc.).
- Combinar as informações fornecidas pelo usuário no chat com os dados da base de usuários e rodar a previsão usando o modelo treinado.
- Fornecer previsões personalizadas, como:

> “Com base nos dados fornecidos e no seu histórico de projetos, o seu projeto tem 75% de chance de ser bem-sucedido.”
>
> “Seu orçamento está abaixo da média dos projetos de sucesso. Considere ajustar o orçamento.”

📊 Base de dados sugerida

Para tornar o caso acessível, você pode utilizar um conjunto de dados público ou criar um banco de dados com informações sobre projetos. Algumas opções incluem:

- **Kaggle: "Project Management Dataset":** Um conjunto de dados com informações sobre projetos, como orçamento, equipe, duração e sucesso do projeto.
- **Kaggle: "Software Project Data":** Um conjunto de dados com informações sobre projetos de software, como tempo de desenvolvimento, número de bugs, tamanho da equipe, entre outros.

Ou, você pode criar sua própria base de dados com informações sobre projetos que você achar relevante para o caso.
A base de dados pode ser em formato **CSV**, **JSON** ou qualquer outro formato adequado.

📌 Exemplo de estrutura da base de dados

1. Base de Dados de Projetos (para treinamento do modelo de ML):

Projeto_ID	Duração (meses)	Orçamento (R$)	Tamanho da Equipe	Recursos Disponíveis	Sucesso (1=sim, 0=não)
1	6	500,000	10	Alto	1
2	12	1,000,000	15	Médio	0
3	9	750,000	8	Baixo	1
4	18	2,000,000	20	Alto	1

2. Base de Dados de Usuários (para o *chatbot* acessar):

Usuario_ID	Nome	Cargo	Histórico de Projetos	Experiência (anos)	Sucesso Médio (percentual)
1	João	Gerente de TI	15	5	80%
2	Maria	Analista de Projetos	10	3	65%
3	Pedro	Coordenador de Projetos	25	8	90%

⚠️ Restrições

- O relatório gerado pelo chatbot deve ser conciso e direto, com informações claras sobre a previsão de sucesso ou fracasso do projeto.
- O modelo de ML deve ser treinado de forma simples e eficiente, sem necessidade de um grande volume de dados.
- O deploy da API deve ser simples, mas funcional, permitindo acesso a dados do projeto e retornando uma previsão de sucesso.
- O chatbot deve ser interativo e fácil de usar, fornecendo previsões baseadas no modelo treinado e nas informações coletadas da base de usuários.

✅ Entregáveis

- **Modelo de ML Tradicional:** Treinamento e validação de um modelo em Python para prever o sucesso de novos projetos com base em dados históricos.
- **API de Deploy:** Criação de uma API simples para expor o modelo treinado e fazer previsões sobre novos projetos.
- **Chatbot Funcional:** Implementação de um chatbot que interage com o usuário, coleta dados sobre o novo projeto, acessa a base de dados de usuários, combina as informações e retorna uma previsão de sucesso com base no modelo treinado.
- **Documentação:** Explicação da abordagem adotada, incluindo detalhes sobre as escolhas técnicas, o processo de treinamento do modelo, e a implementação do chatbot.

Critérios de avaliação

- **Capacidade Técnica:**
  - Implementação eficaz do modelo de aprendizado de máquina em Python e deploy simplificado da API.
  - Qualidade e organização do código.
- **Desempenho do Modelo:**
  - Acuracidade e outras métricas de avaliação do modelo, como precisão, recall e F1-score.
- **Funcionalidade do Chatbot:**
  - Integração do chatbot com a API do modelo e a base de dados de usuários.
  - Personalização das respostas do chatbot com base nos dados fornecidos pelo usuário.
- **Explicabilidade e Justificativas:**
  - Justificativas claras para as escolhas feitas durante o treinamento do modelo e a construção do chatbot.
- **Inovação e Criatividade:**
  - Demonstração de soluções inovadoras ou personalizações que mostrem habilidades adicionais do candidato.

Submissão

Por favor, envie seu case de uma das seguintes maneiras:

- **GitHub:** Compartilhe seu repositório no GitHub com duas pastas:
  - Uma pasta contendo o código do modelo de ML (incluir README explicando como rodá-lo).
  - Uma pasta contendo o código do chatbot (geralmente um arquivo JSON ou estrutura similar).

> ⚠ **Prazo de 7 dias corridos.**
>
> Após esse período, não serão aceitas alterações no repositório Git.