

Trabalho Final

Valor (5.0)



Aplicação Web com Análise de Dados e Machine Learning

Desenvolver uma aplicação web utilizando Python, que permita ao usuário fazer o upload de arquivos de dados no formato .csv e realizar análises visuais e predições com machine learning. O objetivo é oferecer insights interativos e predições personalizadas, com base em variáveis presentes nos dados, possibilitando diferentes tipos de visualizações e uso de classificadores configuráveis pelo usuário.



TEMA

O tema do conjunto de dados e os objetivos de análise com recursos de aprendizado de máquina ficarão a encargo da equipe e devem ser apresentados ao professor antes de iniciar as atividades de desenvolvimento, para evitar temas duplicados na turma.

Então, lembre-se de dizer seu tema durante a aula ou, caso não consiga estar em sala, via mensagem no blackboard. Assim, caso haja temas iguais a ordem de horário será respeitada com prioridade aos grupos que estavam em sala.





Funcionalidades

1. Upload e Flexibilidade dos Dados
2. Análise de Dados e Visualização
3. Machine Learning com Personalização de Modelos
4. Treinamento Dinâmico do Modelo



Upload e Flexibilidade dos Dados

- A aplicação deverá permitir o upload de um arquivo .csv, que contenha dados estruturados em colunas consistentes. A aplicação deve ser capaz de gerar gráficos e análises automaticamente, considerando que o usuário poderá enviar diferentes bases com as mesmas colunas. A flexibilidade dos dados é essencial para permitir a análise de dados variados, como de imóveis de cidades diferentes, jogos novos, ou atualização de dados cursos online.



Análise de Dados e Visualização

A aplicação deve gerar gráficos de análise com as variáveis do dataset, usando tanto gráficos básicos (como gráficos de barras e pizza) quanto gráficos mais avançados e interativos, como mapas para visualização de dados geográficos. Por exemplo:

- **Imóveis:** Visualizar a distribuição de preços por bairro, a localização dos imóveis no mapa, e a relação entre o preço e características como tamanho, número de quartos e ano de construção.
- **Jogos:** Analisar popularidade por país, tipos de jogos mais jogados por região, distribuição de avaliações por gênero e gênero predominante em cada país.
- **Cursos Online:** Visualizar a popularidade por temas de cursos em cada país, preços médios por idioma e país, e a distribuição de alunos por área de estudo ou tipo de curso.





Machine Learning

A aplicação deve permitir ao usuário realizar previsões com base em atributos relevantes, como por exemplo: estimar o preço de uma nova casa com base em combinações como bairro, metragem (m^2), número de quartos, ano de construção, número de banheiros e proximidade de serviços essenciais (escolas, transporte público). Outros exemplos no próximo slide.

- **Escolha e Configuração de Classificadores:** O usuário deverá poder escolher entre diferentes classificadores/preditores e ajustar seus parâmetros (como profundidade em árvores de decisão ou número de vizinhos em KNN). Essa funcionalidade permite que o usuário teste múltiplos modelos e visualize como diferentes configurações influenciam as previsões.





Exemplos

- **Jogos:** Prever a probabilidade de sucesso de um jogo considerando atributos como preço, nicho, gênero, número de jogadores simultâneos, faixa etária do público, e presença de recursos multiplayer. Também é possível estimar o preço ideal para um jogo com base em avaliações, gênero, nicho e média de avaliações dos jogos no mesmo gênero.
- **Cursos Online:** Prever o preço ideal de um curso com base em país, idioma, tema, duração do curso, e nível de dificuldade. Outra predição pode envolver estimar o número de alunos que o curso atrairá com base em variáveis como tipo de curso, linguagem, nível de engajamento esperado, e popularidade do tema em determinada região.





Treinamento Dinâmico do Modelo

- Como a base de dados pode ser alterada com frequência, a aplicação deve incluir uma funcionalidade para re-treinar o modelo sempre que novos dados forem inseridos. Essa opção deve estar disponível para permitir ao usuário treinar o modelo conforme necessário para garantir previsões atualizadas.
Exemplo: cidades diferentes.



Critérios de Avaliação

1. Qualidade Técnica da Implementação
2. Análise de Dados
3. Machine Learning
4. Documentação
5. Apresentação Final / Defesa de Código



Critérios de Avaliação

Qualidade Técnica da Implementação

- **Uso de Ferramentas:** Implementação correta de bibliotecas (ex.: pandas, matplotlib/seaborn, scikit-learn).
- **Código Límpo:** O código é legível, bem comentado e segue boas práticas de programação.
- **Estrutura do Projeto:** Organização lógica e modular do código e dos arquivos.
- **Divisão de Tarefas:** As responsabilidades foram claramente distribuídas entre os membros da equipe.

Análise de Dados

- **Manipulação de Dados:** Uso adequado do pandas para manipulação e limpeza dos dados.
- **Visualização:** Criação de gráficos informativos e bem apresentados, utilizando matplotlib, seaborn, ou gráficos interativos para melhorar a experiência do usuário.





Critérios de Avaliação

Machine Learning

- **Modelagem e Predição:** Aplicação de técnicas de machine learning com scikit-learn, incluindo a seleção de modelos adequados ao problema.
- **Configuração de Modelos:** Implementação de opções de escolha de modelos e ajuste de parâmetros.
- **Treinamento Dinâmico:** Funcionalidade para re-treinar o modelo com novos dados.

Documentação

- **README e Documentação:** Disponibilização de um README detalhado e documentação explicativa para facilitar a compreensão e uso do código.





Critérios de Avaliação

Defesa de Código

- **Explicação Individual:** Cada aluno deverá apresentar partes específicas do código, explicando sua funcionalidade, escolhas de implementação, e mostrando seu entendimento.
- **Demonstração Prática:** Demonstração prática das principais funcionalidades da aplicação durante a apresentação.
- **Respostas a Perguntas:** Cada membro deve estar preparado para responder perguntas técnicas sobre o código e as decisões tomadas no projeto.





Como entregar?

Criar uma conta e um repositório no GitHub.

Este repositório deve conter todo os arquivos necessários para rodar o projeto.

Enviar link do repositório na entrega do Blackboard. E mesmo com a entrega do blackboard se encerrando dia delimitado, não podem conter commits realizados após a data de entrega.

Tamanho do grupo: 4 pessoas.

Data de Entrega: Aulas do dia 13/11 e 14/11.