



Predição de Doença Cardíaca (Dataset Cleveland)

Aplicação web construída com **Streamlit** para estimar a probabilidade de um paciente apresentar **doença cardíaca**, utilizando um modelo de Machine Learning treinado sobre o *Heart Disease Dataset (Cleveland)* da UCI.

A interface permite inserir manualmente os principais atributos clínicos de um paciente (idade, sexo, pressão arterial, colesterol, etc.) e retorna:

- Probabilidade de **presença** de doença cardíaca
 - Probabilidade de **ausência** de doença cardíaca
 - Mensagem destacando **alto** ou **baixo risco** segundo o modelo
-



Estrutura do Projeto

```
mvp-heart-disease-function/  
├─ app.py           # Aplicativo Streamlit  
├─ heart_model.pkl  # Modelo treinado (arquivo gerado por você)  
├─ requirements.txt # Dependências Python  
└─ README.md       # Este arquivo
```

O arquivo **heart_model.pkl** **não está versionado** e precisa ser gerado a partir de um treinamento prévio do modelo.



Tecnologias Utilizadas

- **Python**
 - **Streamlit** (interface web)
 - **scikit-learn** (modelo de ML)
 - **pandas** (manipulação de dados)
 - **NumPy** (operações numéricas)
-



Instalação e Configuração

1. Pré-requisitos

- Python 3.8+ instalado
- **pip** configurado

2. Clonar o repositório

```
git clone <URL_DO_REPOSITORIO>
cd mvp-heart-disease-function
```

3. (Opcional, mas recomendado) Criar ambiente virtual

```
python -m venv .venv
source .venv/bin/activate # Linux/macOS
# .venv\Scripts\activate # Windows (PowerShell)
```

4. Instalar dependências

```
pip install -r requirements.txt
```

5. Adicionar o modelo treinado `heart_model.pkl`

O arquivo `app.py` espera encontrar um modelo salvo no arquivo `heart_model.pkl` na **mesma pasta** do app:

```
with open("heart_model.pkl", "rb") as f:
    model = pickle.load(f)
```

Você deve treinar um modelo de classificação (por exemplo, `RandomForestClassifier`, `LogisticRegression`, etc.) usando o dataset Cleveland processado e salvá-lo com `pickle` nesse arquivo.

Exemplo simplificado de treinamento do modelo

Atenção: este é apenas um exemplo ilustrativo. Ajuste métricas, validação e pré-processamento conforme o seu TCC/projeto.

```
import pandas as pd
import pickle

from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier

# Carregar dataset já pré-processado
df = pd.read_csv("heart_cleveland_processed.csv")

FEATURES = [
    "age", "sex", "cp", "trestbps", "chol", "fbs",
    "restecg", "thalach", "exang", "oldpeak", "slope",
```

```

    "ca", "thal",
]

X = df[FEATURES]
y = df["target"] # coluna alvo (0 = sem doença, 1 = com doença)

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)

model = RandomForestClassifier(
    n_estimators=200,
    max_depth=None,
    random_state=42,
)

model.fit(X_train, y_train)

with open("heart_model.pkl", "wb") as f:
    pickle.dump(model, f)

```

Depois de gerar o `heart_model.pkl`, copie-o para a raiz deste projeto, ao lado do arquivo `app.py`.

12 Features esperadas pelo modelo

O app coleta as seguintes variáveis e monta um `DataFrame` com estas **colunas**, nesta ordem:

1. `age` — Idade (anos)
2. `sex` — Sexo (`1` = Masculino, `0` = Feminino)
3. `cp` — Tipo de dor torácica
 - `1` = Angina típica
 - `2` = Angina atípica
 - `3` = Dor não anginosa
 - `4` = Assintomático
4. `trestbps` — Pressão arterial em repouso (mm Hg)
5. `chol` — Colesterol sérico (mg/dl)
6. `fbs` — Glicemia de jejum > 120 mg/dl (`1` = Sim, `0` = Não)
7. `restecg` — Resultado do ECG em repouso
 - `0` = Normal
 - `1` = Anormalidade onda ST-T
 - `2` = Hipertrofia ventricular
8. `thalach` — Frequência cardíaca máxima alcançada (bpm)
9. `exang` — Angina induzida por exercício (`1` = Sim, `0` = Não)
10. `oldpeak` — Depressão do segmento ST induzida por exercício
11. `slope` — Inclinação do segmento ST no pico do exercício

- 1 = Ascendente
 - 2 = Plano
 - 3 = Descendente
12. **ca** — Número de vasos principais coloridos por fluoroscopia (0–4)
13. **thal** — Resultado do exame Thal
- 3 = Normal
 - 6 = Defeito fixo
 - 7 = Defeito reversível

Certifique-se de que o pré-processamento do dataset e o modelo treinado utilizam **exatamente a mesma codificação** de variáveis e a mesma ordem das colunas.

► Executando o Aplicativo

Com o ambiente configurado e o arquivo **heart_model.pkl** na pasta do projeto, execute:

```
streamlit run app.py
```

O Streamlit abrirá automaticamente o app no navegador (por padrão, em **http://localhost:8501**).



Uso do Aplicativo

1. **Preencha** todos os campos com os dados do paciente.
2. Clique em "**Fazer predição**".
3. O sistema exibirá:
 - Mensagem indicando **alta** ou **baixa** probabilidade de doença cardíaca.
 - Probabilidade (em %) de **doença** e de **ausência de doença**.

PROF



Observações

- Este projeto é um **MVP acadêmico** (ex.: TCC) e **não** substitui avaliação médica profissional.
- A qualidade das predições depende diretamente da qualidade do dataset, do pré-processamento e do modelo utilizado.
- Ajuste o treinamento, escolha de algoritmo e métricas de acordo com os objetivos do seu estudo.