

# Documentação Técnica: Sistema Inteligente de Predição de Obesidade

---

## 1. Introdução e Visão Geral

Este documento detalha a arquitetura, o pipeline de Machine Learning (ML) e os componentes de software do **Sistema Inteligente de Predição de Obesidade**. O projeto foi desenvolvido como uma solução completa para o Tech Challenge de Data Analytics, com o objetivo de fornecer uma ferramenta robusta e acessível para auxiliar equipes médicas na triagem, diagnóstico preditivo e intervenção precoce do nível de obesidade em pacientes.

A solução é composta por dois módulos principais, ambos implementados em Streamlit, garantindo acessibilidade e usabilidade: a Aplicação de Predição e o Painel Analítico.

## 2. Arquitetura da Solução

A arquitetura do sistema é modular e centrada no modelo de Machine Learning, que é o núcleo preditivo da solução.

Componente	Função Principal	Tecnologia	Acesso
<b>ML Pipeline</b>	Treinamento, validação e exportação do modelo preditivo.	Python, Scikit-learn, Pandas	<a href="#">ml_pipeline_obesity.py</a>
<b>Aplicação de Predição</b>	Interface de usuário para entrada de dados do paciente e predição em tempo real.	Streamlit, Python	<a href="https://predicao-obesidade-tech-4.streamlit.app/">https://predicao-obesidade-tech-4.streamlit.app/</a>
<b>Painel Analítico</b>	Dashboard estratégico para visualização de dados e insights epidemiológicos.	Streamlit, Python	<a href="https://painel-obesidade--tech-4.streamlit.app/">https://painel-obesidade--tech-4.streamlit.app/</a>
<b>Repositório</b>	Controle de versão e código-fonte completo do projeto.	Git, GitHub	<a href="https://github.com/gustmacena/predicao-obesidade">https://github.com/gustmacena/predicao-obesidade</a>

### 3. Pipeline de Machine Learning

O modelo preditivo é um **Gradient Boosting Classifier (GBC)**, escolhido por sua alta precisão e capacidade de lidar com a classificação multiclasse dos 7 níveis de obesidade.

#### 3.1. Feature Engineering e Pré-processamento

O pipeline de ML ([ml\\_pipeline\\_obesity.py](#)) realiza etapas cruciais de pré-processamento, incluindo a tradução das variáveis para o português e a preparação dos dados para o algoritmo.

#### Features Utilizadas

O modelo utiliza 16 features, divididas em numéricas e categóricas, que abrangem dados demográficos, hábitos alimentares e estilo de vida.

Categoria	Features Numéricas	Features Categóricas
Demográficas	Idade, Altura, Peso	Gênero
Hábitos Alimentares	FCVC (Frequência de Consumo de Vegetais), NCP (Número de Refeições Principais), Água por dia (CH2O)	FAVC (Consumo de Alimentos Calóricos), CAEC (Consumo de Lanches), Álcool (CALC)
Estilo de Vida	Atividade Física (FAF), Tempo em Telas (TUE)	Histórico Familiar, Fuma (SMOKE), Conta Calorias (SCC), Transporte (MTRANS)

#### Etapas de Pré-processamento

O pré-processamento é encapsulado em um [ColumnTransformer](#) e um [Pipeline](#) do Scikit-learn, garantindo que as transformações sejam aplicadas corretamente e de forma consistente:

- 1 **Normalização (StandardScaler):** Aplicada às features numéricas para padronizar a escala dos dados.
- 2 **Codificação (OneHotEncoder):** Aplicada às features categóricas para convertê-las em um formato binário adequado para o modelo.
- 3 **Pipeline:** O [Pipeline](#) final combina o pré-processamento com o classificador GBC, garantindo que o modelo treinado seja exportado com todas as etapas de transformação necessárias.

#### 3.2. Performance do Modelo

O modelo demonstrou uma performance excepcional, superando a meta de 75% de assertividade.

Métrica	Resultado	Detalhes
Acurácia Média (CV 5-Fold)	95.74%	Média de acurácia obtida na validação cruzada (folds: 95.51%, 95.50%, 96.68%, 95.26%, 95.73%).
Acurácia (Holdout Test)	95.27%	Acurácia final obtida em um conjunto de teste (Holdout) de 20% dos dados.

## Relatório de Classificação (Holdout Test)

Classe	Precisão	Recall	F1-Score	Suporte
Baixo Peso	0.96	0.89	0.92	54
Peso Normal	0.85	0.91	0.88	58
Sobrepeso I	0.95	0.91	0.93	58
Sobrepeso II	0.95	0.97	0.96	58
Obesidade I	0.97	0.99	0.98	70
Obesidade II	0.98	1.00	0.99	60
Obesidade III	1.00	0.98	0.99	65
<b>Média Ponderada</b>	<b>0.95</b>	<b>0.95</b>	<b>0.95</b>	<b>423</b>

## 4. Deployment e Execução

O projeto utiliza o **Streamlit Community Cloud** para o deployment contínuo (CI/CD), garantindo que as aplicações estejam sempre acessíveis e atualizadas.

### 4.1. Pré-requisitos e Execução Local

Para executar o projeto localmente, é necessário ter Python 3.x e as dependências listadas no [requirements.txt](#).

```
# Instalar dependências

pip install -r requirements.txt

# Executar a Aplicação de Predição
streamlit run app.py

# Executar o Painel Analítico

streamlit run app_dashboard.py
```

## 4.2. Guia de Uso (Aplicação de Predição)

A aplicação de predição é projetada para ser intuitiva para o usuário final (médicos e pacientes):

- 4 **Entrada de Dados:** O usuário insere os dados do paciente em abas organizadas: "Dados Pessoais", "Hábitos Alimentares" e "Estilo de Vida".
- 5 **Predição:** Ao clicar no botão de predição, o modelo carregado (`obesity_pipeline.pkl`) é invocado.
- 6 **Saída:** O resultado é exibido em um **Card de Predição** destacado, acompanhado de um **Card de Peso Ideal** (faixa de peso saudável) e um **Gráfico de Probabilidades** (confiança do modelo em cada classe).

## 5. Referências

Aplicação de Predição - Streamlit: <https://predicao-obesidade-tech-4.streamlit.app/>

Painel Analítico - Streamlit: <https://painele-obesidade--tech-4.streamlit.app/>

Repositório GitHub: <https://github.com/gustmacena/predicao-obesidade>