**Projeto de Obesidade Infantil**

**Dataset do Kaggle**

<https://www.kaggle.com/datasets/thedevastator/childhood-obesity-levels>

**Passo a Passo para o Projeto**

1. Configuração do Ambiente

* Criar repositório no GitHub
* Configurar ambiente virtual Python
* Instalar bibliotecas necessárias

2. Coleta e Carregamento de Dados

* Download do dataset do Kaggle
* Carregar o dataset no Jupyter Notebook

3. Análise Exploratória dos Dados (EDA)

* Visualização inicial dos dados
* Análise estatística descritiva
* Tratamento de valores ausentes e outliers

4. Transformação dos Dados

* Normalização e padronização dos dados
* Criação de novas features se necessário

5. Modelagem dos Dados

* Divisão do dataset em treino e teste
* Treinamento de modelos de machine learning
* Avaliação dos modelos

6. Visualização dos Resultados

* Criação de gráficos e dashboards
* Interpretação dos resultados

7. Documentação e Publicação

* Documentar o processo no GitHub
* Publicar um README detalhado
* Compartilhar o projeto

**Etapa 1: Configuração do Ambiente**

**1.1 Criar repositório no GitHub**

* Crie um repositório no GitHub chamado childhood-obesity-project.
* Clone o repositório na sua máquina local.

git clone https://github.com/seu-usuario/childhood-obesity-project.git

cd childhood-obesity-project

**1.2 Configurar ambiente virtual Python**

* Crie um ambiente virtual para isolar as dependências do projeto.

python -m venv env

source env/bin/activate # Para Windows, use `env\Scripts\activate`

**1.3 Instalar bibliotecas necessárias**

* Instale as bibliotecas necessárias utilizando pip.

pip install pandas numpy matplotlib seaborn scikit-learn jupyter

**1.4 Verificar Bibliotecas Instaladas**

Você pode verificar as bibliotecas instaladas no ambiente virtual usando:

pip list

Quando você cria um ambiente virtual usando venv e ativa esse ambiente, ele isola as dependências do seu projeto em um diretório separado (env neste caso). Isso significa que todas as bibliotecas que você instalar depois de ativar o ambiente virtual serão instaladas apenas dentro desse ambiente e não afetarão as bibliotecas instaladas globalmente no seu sistema.

**Benefícios de Usar um Ambiente Virtual**

1. **Isolamento de Dependências**: Bibliotecas instaladas em um ambiente virtual não interferem com outras bibliotecas instaladas globalmente ou em outros ambientes virtuais.
2. **Controle de Versões**: Você pode usar diferentes versões de bibliotecas em diferentes projetos sem conflito.
3. **Facilidade de Manutenção**: Facilita a replicação do ambiente de desenvolvimento em outras máquinas, tornando a colaboração e o deployment mais simples.

**Como Funciona**

* **Criar um Ambiente Virtual**: O comando python -m venv env cria um novo diretório chamado env que contém uma cópia do interpretador Python e scripts para gerenciar pacotes.
* **Ativar o Ambiente Virtual**: O comando source env/bin/activate (ou env\Scripts\activate no Windows) altera o PATH do seu terminal para usar o Python e pip do ambiente virtual.
* **Instalar Bibliotecas**: Quando você instala bibliotecas com pip install após ativar o ambiente virtual, elas são instaladas dentro do diretório env e não globalmente.

**1.5 Iniciar Jupyter Notebook**

* Inicie o Jupyter Notebook para começar a trabalhar com os dados.

jupyter notebook

#### **Justificativa para utilização do Jupyter notebook**

O Jupyter Notebook é uma ferramenta muito útil, especialmente nas fases iniciais de um projeto de dados, por várias razões:

### **Vantagens do Jupyter Notebook**

1. **Exploração e Prototipagem Rápida**: O Jupyter Notebook permite a execução de células de código interativas, facilitando a exploração dos dados, testes de hipóteses e prototipagem rápida.
2. **Visualização**: É fácil criar e visualizar gráficos diretamente no notebook, o que é extremamente útil para a análise exploratória de dados (EDA).
3. **Documentação**: Você pode combinar código, visualizações e texto explicativo em um único documento, o que torna a documentação do processo de análise muito mais clara e detalhada.
4. **Feedback Imediato**: Você pode ver os resultados imediatamente após executar uma célula, o que ajuda a ajustar o código rapidamente.

### **Limitações do Jupyter Notebook**

1. **Não Escalável**: Como você mencionou, Jupyter Notebooks não são ideais para projetos de produção escaláveis. Eles são mais indicados para análise exploratória, prototipagem e desenvolvimento inicial.
2. **Gestão de Dependências**: Pode ser difícil gerenciar dependências em notebooks se não forem bem organizadas.
3. **Desempenho**: Para grandes volumes de dados ou processamento intensivo, o desempenho pode ser limitado.

### **Uso no Projeto**

**Fase Inicial**:

* **Exploração de Dados**: Usaremos o Jupyter Notebook para carregar, explorar e entender os dados. Isso inclui a análise exploratória dos dados, visualizações iniciais e testes de hipóteses.
* **Prototipagem**: Desenvolveremos protótipos de modelos de machine learning e faremos ajustes iniciais.

**Fase de Produção**:

* **Script Python**: Após a fase de exploração, o código final pode ser convertido em scripts Python organizados e otimizados para produção.
* **Pipeline de Dados**: Podemos usar ferramentas como Apache Airflow, Luigi ou mesmo scripts agendados para construir pipelines de dados que rodem em produção.
* **Deploy**: Para a etapa final, integraremos o código em ambientes de produção usando serviços como AWS, Google Cloud Platform ou Azure.

### **Etapa 2: Coleta e Carregamento de Dados**

#### **2.1 Download do Dataset do Kaggle**

* Acesse o link do dataset de níveis de obesidade infantil no Kaggle: [Childhood Obesity Levels](https://www.kaggle.com/datasets/thedevastator/childhood-obesity-levels).
* Faça o download do dataset. Normalmente, ele estará disponível em formato .csv.

#### **2.2 Carregar o Dataset no Jupyter Notebook**

* No terminal do VSCode (com o ambiente virtual ativado), inicie o Jupyter Notebook:

jupyter notebook

* Crie um novo notebook no Jupyter.
* Carregue o dataset usando a biblioteca pandas. O código a seguir mostra como fazer isso:

import pandas as pd

# Carregar o dataset

df = pd.read\_csv('caminho/para/seu/dataset.csv')

# Visualizar as primeiras linhas do dataset

df.head()

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Dados das planilhas .csv são visualizados

### **Etapa 3: Análise Exploratória dos Dados (EDA)**

A análise exploratória dos dados nos ajuda a entender melhor os dados com os quais estamos trabalhando, identificando padrões, valores ausentes e outliers, e começando a formular hipóteses para nossa análise.

#### 3.1 Visualização Inicial dos Dados

* **Visualizar as primeiras linhas dos datasets**: Já fizemos isso com o método head().
* **Informações dos dados**: Use info() para obter um resumo das colunas, tipos de dados e valores não nulos.

# Informações dos dados

print(df\_gender.info())

print(df\_age.info())

#### 3.2 Análise Estatística Descritiva

* **Estatísticas descritivas**: Use describe() para obter estatísticas descritivas como média, mediana, desvio padrão, etc.

# Estatísticas descritivas

print(df\_gender.describe())

print(df\_age.describe())

#### 3.3 Tratamento de Valores Ausentes

* **Identificar valores ausentes**: Use isnull().sum() para ver quantos valores ausentes existem em cada coluna.

# Identificar valores ausentes

print(df\_gender.isnull().sum())

print(df\_age.isnull().sum())

#### 3.4 Visualização dos Dados

* **Histogramas**: Para visualizar a distribuição dos dados.
* **Gráficos de dispersão**: Para identificar possíveis correlações entre variáveis.

import matplotlib.pyplot as plt

import seaborn as sns

# Histogramas

df\_gender.hist(bins=30, figsize=(10, 8))

plt.suptitle('Distribuição dos Dados - Gender')

plt.show()

df\_age.hist(bins=30, figsize=(10, 8))

plt.suptitle('Distribuição dos Dados - Age')

plt.show()

# Gráficos de dispersão

sns.pairplot(df\_gender)

plt.suptitle('Correlação dos Dados - Gender')

plt.show()

sns.pairplot(df\_age)

plt.suptitle('Correlação dos Dados - Age')

plt.show()

### Etapa 4: Transformação dos Dados

#### 4.1 Limpeza de Dados

1. **Verificação de Valores Negativos**:
   * Identificamos valores negativos nas colunas PercentOW e PercentOB.
   * Tratamos esses valores negativos, substituindo-os por zero.

# Verificar e tratar valores negativos

df\_gender['PercentOW'] = df\_gender['PercentOW'].apply(lambda x: max(x, 0))

df\_gender['PercentOB'] = df\_gender['PercentOB'].apply(lambda x: max(x, 0))

#### 4.2 Padronização dos Dados

* Decidimos que a padronização não era necessária para este projeto específico, pois não estávamos usando algoritmos sensíveis à escala dos dados.

#### 4.3 Criação de Novas Features

1. **Criação da Feature OW\_to\_OB\_Ratio**:
   * Calculamos a razão entre PercentOW e PercentOB para entender melhor a relação entre sobrepeso e obesidade.

# Criar a nova feature 'OW\_to\_OB\_Ratio'

df\_gender['OW\_to\_OB\_Ratio'] = df\_gender['PercentOW'] / df\_gender['PercentOB']

#### 4.4 Análise de Outliers e Distribuições

1. **Identificação de Outliers**:
   * Usamos boxplots para identificar outliers nas colunas PercentOW, PercentOB e OW\_to\_OB\_Ratio.

# Boxplot para identificar outliers

plt.figure(figsize=(10, 6))

sns.boxplot(data=df\_gender[['PercentOW', 'PercentOB', 'OW\_to\_OB\_Ratio']])

plt.title('Boxplot - Identificação de Outliers')

plt.show()

1. **Análise de Distribuições**:
   * Criamos histogramas para analisar as distribuições das variáveis e garantir que os dados estavam devidamente preparados para a modelagem.

# Histogramas para analisar distribuições

df\_gender[['PercentOW', 'PercentOB', 'OW\_to\_OB\_Ratio']].hist(bins=15, figsize=(15, 6), layout=(2, 3))

plt.suptitle('Distribuições das Variáveis')

plt.show()

### Conclusão da Etapa 4

Na etapa 4, nos concentramos em garantir que os dados fossem limpos, transformados e preparados para a análise subsequente. Tratamos valores negativos, criamos novas features e analisamos outliers e distribuições. Houve a decisão de não aplicar Winsorização para manter a integridade dos dados.

 5 **Análise de Dados**

* Utilize técnicas estatísticas ou de machine learning para analisar os dados e extrair insights.

 6 **Visualização dos Dados**

* Crie dashboards e gráficos para visualizar os resultados da análise. Ferramentas: Tableau, Power BI, matplotlib, seaborn.

 7 **Conclusão e Comunicação dos Resultados**

* Resuma os resultados obtidos e prepare uma apresentação ou relatório.

Fonte do novo dataset

<https://catalog.data.gov/dataset/obesity-among-children-and-adolescents-aged-219-years-by-selected-characteristics-united-s-3ead4>

<https://data.cdc.gov/api/views/9gay-j69q>

Fonte do primeiro dataset

<https://www.kaggle.com/datasets/thedevastator/childhood-obesity-levels?select=obesity_child_age.csv>