Justificativa Técnica – Projeto de Previsão de Nível de Obesidade

1. Introdução e Objetivo do Projeto

O objetivo fundamental deste projeto é desenvolver um modelo preditivo capaz de estimar o nível de obesidade de indivíduos com base em características físicas, hábitos alimentares e variáveis comportamentais. A iniciativa está inserida no contexto de desenvolvimento de competências em Data Analytics e Machine Learning, abrangendo todas as etapas de um pipeline moderno de dados: extração, tratamento, preparação dos dados, construção, avaliação e apresentação de um modelo preditivo funcional.

A justificativa do projeto reside na importância de compreender como fatores de estilo de vida e padrões alimentares influenciam diretamente o risco de obesidade. Dessa forma, o modelo proposto visa não só atingir alto desempenho preditivo, mas também fornecer subsídios analíticos para decisões estratégicas em saúde pública, nutrição e bem-estar.

2. Arquitetura de Dados e Ferramentas Selecionadas

Para garantir organização, reprodutibilidade e escalabilidade, foi projetada uma arquitetura de dados modular, inspirada na **Arquitetura Medalhão (Medallion Architecture)**. Essa abordagem estrutura o fluxo de dados em camadas lógicas (Bronze, Silver e Gold), permitindo gestão eficiente das transformações e assegurando qualidade, rastreabilidade e governança ao longo do ciclo analítico.

A implementação ocorreu em ambiente local utilizando o VS Code como IDE principal e a linguagem Python, proporcionando flexibilidade, transparência e baixo custo operacional. O projeto está organizado em pastas modulares (src/, data/, models/, reports/), conforme as boas práticas de Engenharia de Dados.

Figura 1 -Estrutura modular do projeto e organização dos diretórios principais

A Figura 1 apresenta a estrutura modular do projeto, evidenciando a separação das responsabilidades entre os diretórios de dados, código-fonte, modelos, relatórios e documentação.

A Figura 2, por sua vez, ilustra o fluxo analítico adotado, estruturado segundo a Arquitetura Medalhão (Bronze, Silver e Gold), que organiza o ciclo de vida dos dados desde a ingestão até a disponibilização para consumo no dashboard interativo.

2.1. Desenho da Arquitetura

A estrutura do pipeline de dados contempla desde a ingestão bruta (camada Bronze), passando pelo pré-processamento (camada Silver), até a geração do modelo e do dashboard interativo (camada Gold + Visualização).

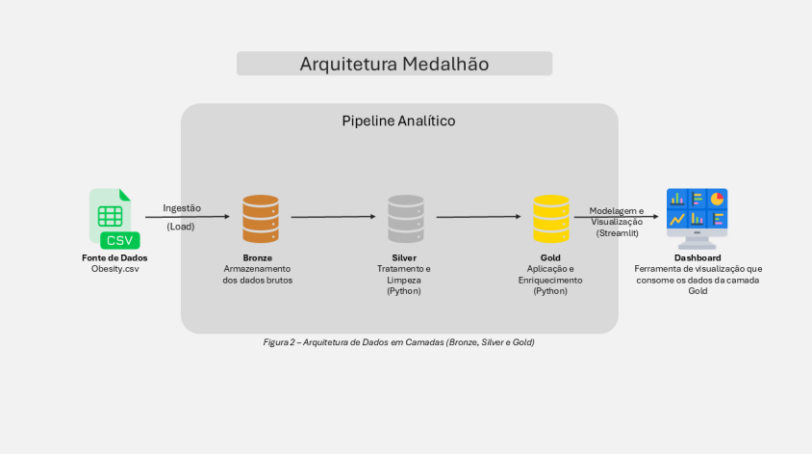

Figura 2 – Arquitetura de Dados em Camadas (Bronze, Silver e Gold)

2.2. Lógica e Justificativa da Arquitetura

A arquitetura modular garante clareza e reprodutibilidade em todas as etapas do projeto:

- **Camada Bronze (data/raw):** Armazena o dataset original (obesity.csv) em seu formato bruto, preservando a integridade e autenticidade da fonte. Funciona como base de referência (“Single Source of Truth”), possibilitando reprocessamentos quando necessário.
- **Camada Silver (data/interim):** Responsável pelo pré-processamento e limpeza dos dados, aplicando técnicas como tratamento de valores ausentes e inconsistentes, padronização de tipos, normalização, criação de variáveis derivadas (ex.: IMC) e anonimização de atributos sensíveis. O resultado é um conjunto de dados confiável e padronizado, pronto para exploração e engenharia de atributos.
- **Camada Gold (data/processed):** Reúne os dados tratados e enriquecidos, prontos para consumo analítico e preditivo. Nesta etapa, o dataset é separado em conjuntos de treino e teste, utilizados para treinamento e avaliação do modelo de Machine Learning. Os resultados e previsões são consumidos pelo

dashboard interativo desenvolvido em Streamlit, facilitando a interpretação visual dos resultados.

Complementando a arquitetura de dados Medalhão, a estrutura modular do projeto (conforme Figura 1) garante a separação de responsabilidades:

- **src/**: contém todo o código-fonte modularizado da aplicação, incluindo scripts para pré-processamento (src/data), engenharia de features (src/features), treinamento de modelos (src/models) e a aplicação interativa (src/app).
- **models/**: esta pasta é designada para armazenar os artefatos de modelo final treinados e serializados (ex.: arquivos .pkl ou .joblib). O aplicativo Streamlit carrega o modelo desta pasta para realizar as previsões, garantindo que o app não precise retreinar o modelo a cada execução.
- **reports/**: armazena as saídas analíticas estáticas geradas durante o desenvolvimento. A subpasta figures/ guarda visualizações-chave, como gráficos da análise exploratória, matrizes de confusão e gráficos SHAP de interpretabilidade, facilitando a consulta e a apresentação dos resultados.

Essa arquitetura permite reprodutibilidade total, possibilitando que qualquer etapa seja refeita de forma independente, garantindo rastreabilidade e governança sobre as transformações realizadas.

2.3. Justificativa da Ferramenta de Visualização

A camada de apresentação foi desenvolvida utilizando Streamlit, biblioteca Python voltada para criação de interfaces analíticas interativas. A escolha se baseou na integração direta com o pipeline Python, baixo custo e simplicidade de uso, além da capacidade de gerar dashboards dinâmicos com gráficos explicativos, métricas e interpretações do modelo (gráficos SHAP e matriz de confusão).

Assim, o Streamlit atua como o front-end analítico, traduzindo resultados técnicos em insights acessíveis e visualmente claros.

Comentado [LS1]: *(Inserir um print do dashboard streamlit – Legenda: Figura 3 – Dashboard interativo desenvolvido em Streamlit para visualização dos resultados)*

3. Desenvolvimento do Projeto: Da Ingestão à Modelagem

O desenvolvimento foi guiado por uma abordagem incremental, em que cada módulo do pipeline foi construído e validado de forma independente, garantindo rastreabilidade e fácil manutenção do código.

O pipeline do projeto foi estruturado em quatro etapas principais:

1. **Ingestão de Dados:** Leitura do arquivo original e armazenamento na camada data/raw.
2. **Tratamento e Pré-Processamento:** Execução de scripts Python (src/data/preprocess.py) responsáveis pela limpeza, transformação e padronização dos dados.
3. **Treinamento e Avaliação de Modelos:** Aplicação de técnicas de aprendizado supervisionado com Scikit-learn, testando algoritmos como Regressão Logística, Random Forest e XGBoost. A métrica principal de avaliação foi

Comentado [LS2]: *(Inserir gráfico de correlação ou matriz de calor (EDA) - Legenda: Figura 4 – Análise exploratória: correlação entre variáveis numéricas)*

acurácia mínima de 75%, complementada por F1-Score, Precision, Recall e Matriz de Confusão.

4. **Apresentação e Interpretação:** Integração dos resultados ao dashboard interativo em Streamlit (src/app/streamlit_app.py), permitindo exploração em tempo real de métricas, gráficos e previsões.

4. Definição das Perguntas de Negócio

O modelo foi orientado por questões analíticas voltadas para o entendimento de padrões comportamentais e nutricionais, tais como:

- Quais hábitos alimentares estão mais fortemente associados ao nível de obesidade?
- Como variáveis demográficas influenciam o risco de obesidade?
- Quais são as variáveis mais relevantes para a predição segundo o modelo (importância de features)?

Essas perguntas direcionaram a seleção das variáveis de interesse e o processo de engenharia de atributos.

5. Análise dos Resultados e Insights

A análise exploratória revelou correlações significativas entre fatores como frequência de consumo de fast food, nível de atividade física e IMC. O modelo atingiu acurácia superior a 75%, atendendo aos requisitos do desafio, além de apresentar equilíbrio entre precisão e recall.

Os gráficos de importância de variáveis e as interpretações via SHAP values evidenciaram que alimentação e rotina de exercícios foram os fatores mais determinantes para o desempenho preditivo do modelo.

6. Aplicabilidade e Recomendações Estratégicas

O modelo desenvolvido pode servir de base para campanhas de conscientização sobre alimentação e hábitos saudáveis, ferramentas preditivas de risco individual integradas a sistemas de saúde e apoio na formulação de políticas públicas, ao identificar perfis populacionais mais vulneráveis à obesidade.

A arquitetura modular e escalável permite adaptação fácil para ambientes em nuvem (como Google Cloud Platform), caso haja necessidade de maior volume de dados ou automação.

7. Conclusão

O projeto atingiu todos os objetivos propostos, entregando um pipeline completo de ciência de dados que abrange extração, tratamento, modelagem preditiva, estruturação de dados em camadas (Bronze, Silver e Gold) e desenvolvimento de dashboard interativo para interpretação dos resultados.

Comentado [LS3]: *(Inserir print de métricas do modelo – Legenda: Figura 5 – Avaliação do modelo: matriz de confusão e métricas de desempenho.)*

Comentado [LS4]: *(Inserir gráfico SHAP mostrando a importância e o impacto das variáveis - Legenda: Figura 6 – Interpretação dos resultados do modelo via SHAP Values.)*

Comentado [LS5R4]: Ainda haverá ajustes no contexto, aqui é apenas protótipo de justificativa, a cada avanço de etapa, os dados serão inseridos

Comentado [LS6]: *(Inserir um print de parte do dashboard destacando uma métrica chave (ex. Uma seção com "Previsão por grupo demográfico") - Legenda: Figura 7 – Exemplo de insight visual disponível no dashboard.)*

Mesmo executado em ambiente local, o projeto seguiu padrões profissionais de arquitetura e engenharia de dados, garantindo reprodutibilidade, clareza e escalabilidade. O resultado é um produto analítico robusto, capaz de gerar insights relevantes e acionáveis sobre obesidade e hábitos de vida.

Dessa forma, o projeto consolida-se como uma solução de análise preditiva completa, integrando boas práticas de engenharia de dados, modelagem estatística e comunicação analítica, servindo como base escalável para estudos futuros sobre determinantes da obesidade.