ECOSSISTEMA DE INTELIGÊNCIA CLÍNICA E PREDIÇÃO DE OBESIDADE.

Modelo preditivo + aplicação (Streamlit) + visão analítica (dashboard) + IA para suporte à decisão.

Equipe Técnica:

Alfonso J. T. Rodriguez – RM 361699

Gabrielle Silva Santos – RM 361901

Isac João Kahan de Barros – RM 364894

Luana Tavares Faiotto – RM 362722

Data: 08-01-2026

Sumário:

INTRODUÇÃO:

A obesidade é uma condição crônica multifatorial, associada ao aumento de risco para doenças cardiovasculares, diabetes tipo 2, hipertensão e outras comorbidades que elevam significativamente a demanda assistencial e o custo operacional de instituições de saúde. No contexto hospitalar e ambulatorial, o desafio não é apenas medir peso ou calcular o IMC, mas avaliar o risco de forma integrada, considerando fatores comportamentais, ambientais e familiares que influenciam o desenvolvimento e a progressão do quadro.

Na prática clínica, essa avaliação integrada enfrenta três obstáculos recorrentes:

1. Tempo limitado de consulta, que reduz a profundidade na coleta e interpretação de hábitos de vida;

2. Alta variabilidade entre profissionais, que pode gerar decisões inconsistentes para casos semelhantes;

3. Baixa rastreabilidade histórica, dificultando auditoria, monitoramento de prevalência e planejamento de campanhas preventivas.

Diante desse cenário, este projeto propõe a criação de um Ecossistema de Inteligência Clínica e Suporte à Decisão para avaliação do nível de obesidade, integrando Machine Learning e IA Generativa em uma solução operacional.

O objetivo é apoiar o profissional de saúde com uma classificação preditiva padronizada (em múltiplos níveis de gravidade) e, ao mesmo tempo, traduzir esse resultado em recomendações orientadas à conduta, com linguagem clara, foco em intervenção e personalização ao perfil do paciente.

A solução foi desenvolvida a partir de uma base estruturada com variáveis antropométricas, comportamentais, alimentares e de contexto familiar, permitindo prever o nível de obesidade em categorias.

Além da predição em si, o sistema foi desenhado para atender dois públicos com necessidades distintas:

- Corpo clínico: precisa de uma interface simples de coleta, resposta rápida e recomendações aplicáveis na rotina (triagem e acompanhamento).

- Gestão hospitalar: precisa de uma visão agregada que identifique padrões de risco, segmentos prioritários e evidências para direcionar ações preventivas e recursos.

Por isso, o projeto entrega um ecossistema em três camadas:

1. Camada Preditiva (CDSS): formulário avaliativo que coleta variáveis clínicas e retorna a classificação do nível de obesidade.

2. Camada Consultiva (IA Generativa): geração de suporte à decisão, orientando condutas e prioridades com base no perfil e no resultado do modelo.

3. Camada Analítica (Storytelling/Dashboard): conjunto de visuais que explicam o "porquê" por trás dos padrões observados e apontam ações estratégicas para prevenção e acompanhamento.

A partir desta introdução, o storytelling percorre as telas do sistema e do dashboard sempre na mesma lógica: pergunta clínica/gerencial → evidência nos dados → interpretação → decisão prática, garantindo que a solução seja

compreendida não como um “modelo de ML”, mas como um instrumento de transformação operacional orientado a resultado.

OBJETIVO:

Desenvolver e disponibilizar um Sistema de Suporte à Decisão Clínica (CDSS) capaz de predizer o nível de obesidade do paciente (em 7 classes, de Insuficiente a Obesidade Tipo III) a partir de variáveis antropométricas, comportamentais, alimentares e de contexto familiar, e converter essa predição em suporte prático à conduta por meio de recomendações geradas por IA, além de disponibilizar uma visão gerencial por dashboard.

Objetivos específicos:

1. Padronizar a avaliação clínica: estruturar a coleta das variáveis em um formulário único, reduzindo variações de julgamento e garantindo comparabilidade entre atendimentos.

2. Aumentar a velocidade e consistência da triagem: fornecer classificação preditiva imediata para apoiar decisões de encaminhamento e priorização.

3. Viabilizar recomendações personalizadas: gerar orientação de apoio à decisão (não prescritiva) com base no perfil do paciente e no nível predito.

4. Criar histórico auditável: registrar cada avaliação para formar base de prevalência e permitir análises futuras (monitoramento, auditoria, estudos internos).

5. Entregar insights para gestão: disponibilizar visualizações que identifiquem padrões (idade, gênero, hábitos, transporte, atividade física) e suportem políticas de prevenção e alocação de recursos.

Critérios de sucesso (critérios de aceitação do projeto):

- Desempenho do modelo: atingir acurácia mínima de 75% na classificação do nível de obesidade (meta do desafio).

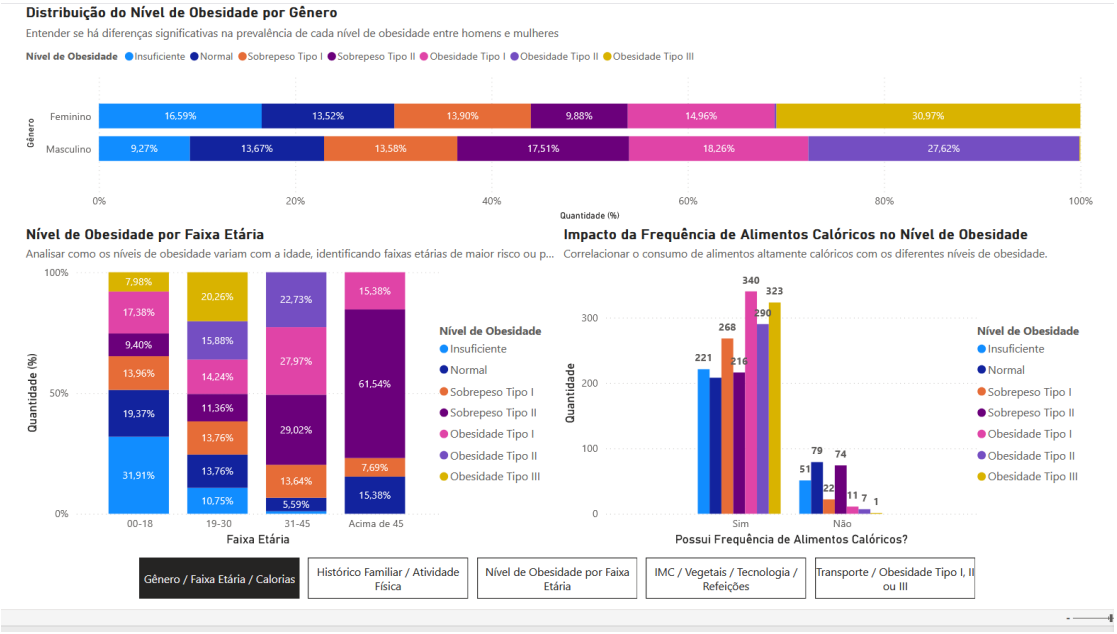
- Usabilidade clínica: permitir preenchimento rápido e retorno claro do resultado, com linguagem objetiva e acionável.

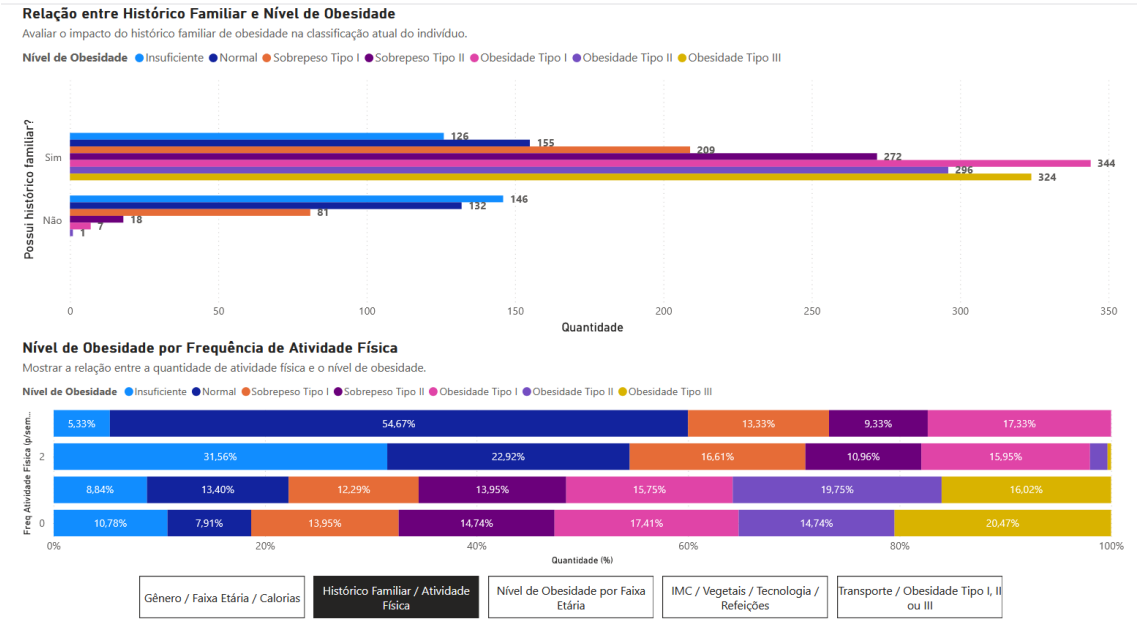
- Rastreabilidade: manter registro de entradas e saídas para auditoria e análises futuras.

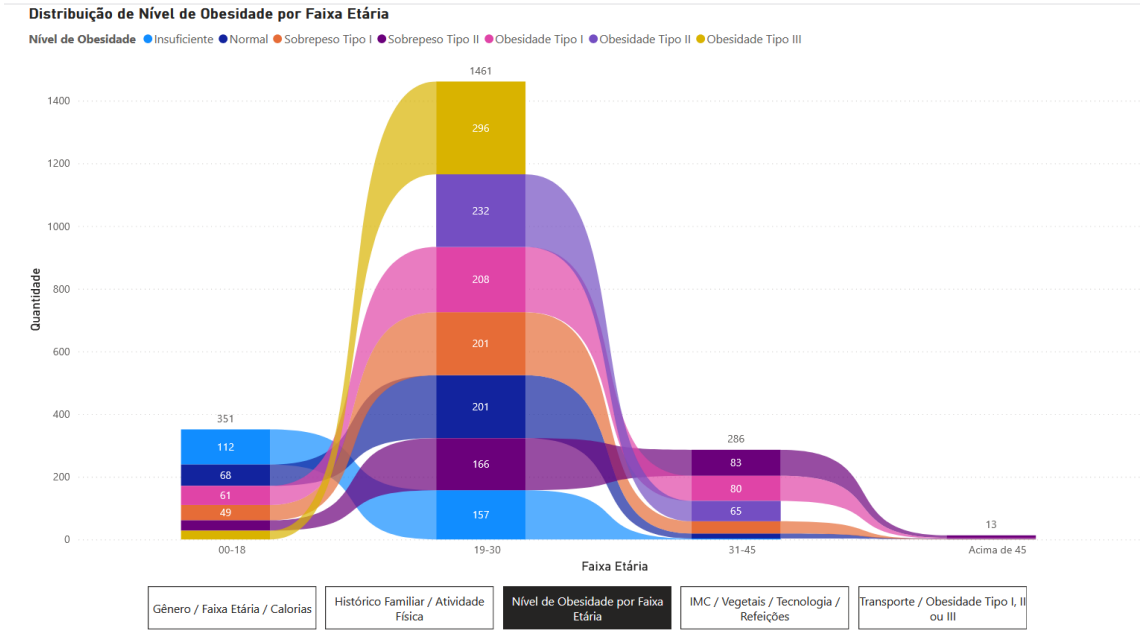
- Valor gerencial: dashboard capaz de responder “quem é mais risco”, “por quê” e “onde intervir” com evidências.

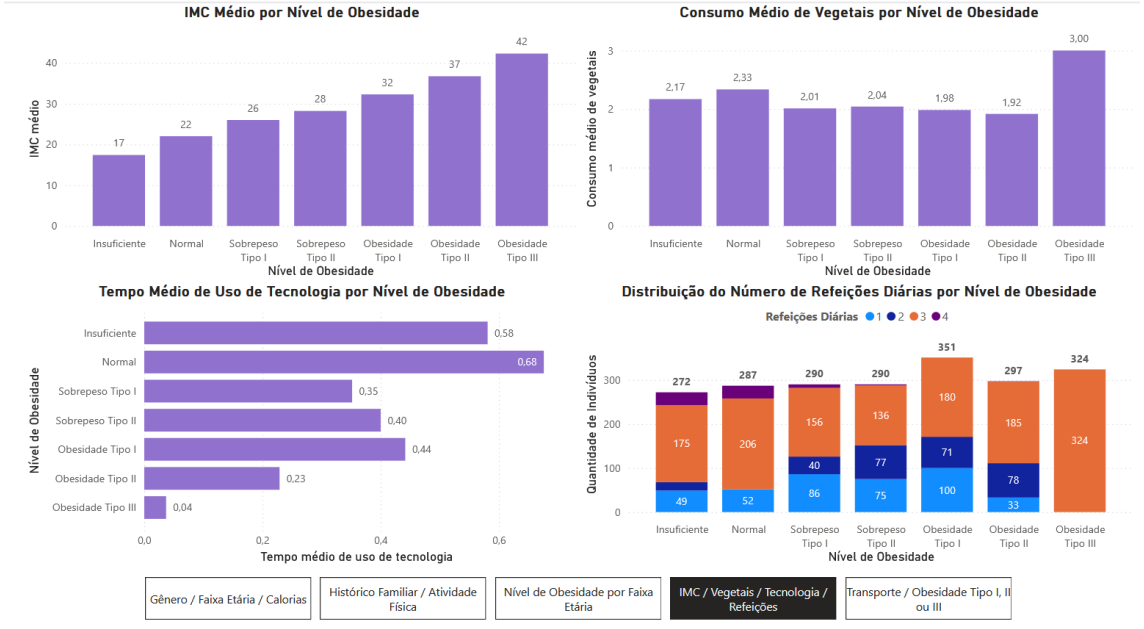
FORMULÁRIO AVALIATIVO (https://lufaiotto.github.io/tech-challenge/Fase_4/**):**

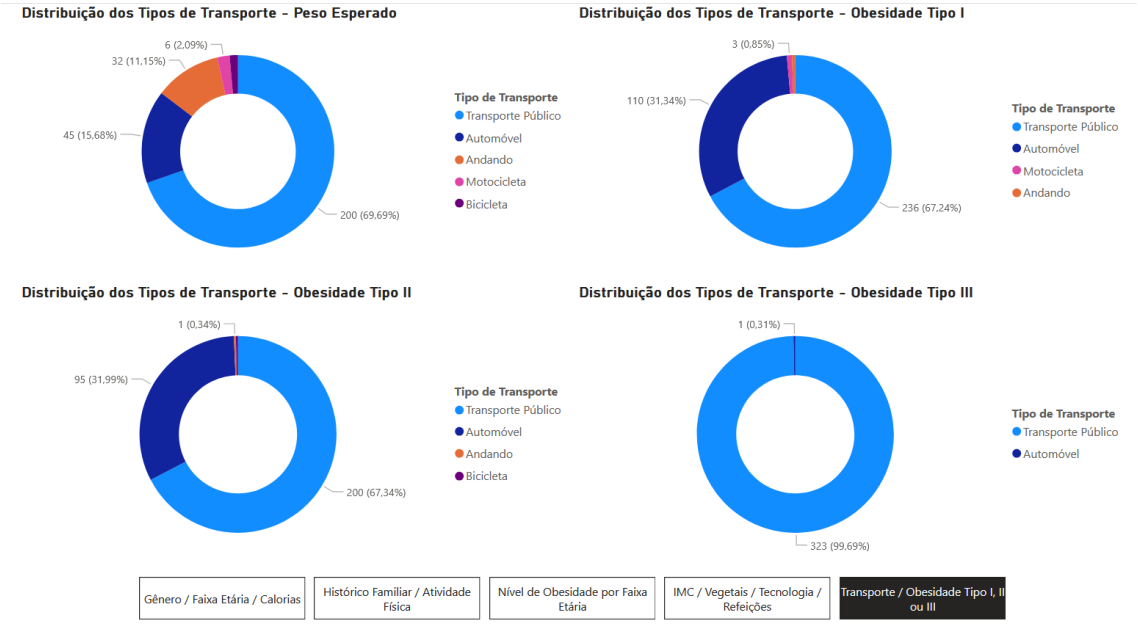
DASHBOARD ESTATÍSTICO (https://lufaiotto.github.io/tech-challenge/Fase_4/:






Distribuição de Nível de Obesidade por Faixa Etária
Nível de Obesidade Insuficiente Normal Sobrepeso Tipo I Sobrepeso Tipo II Obesidade Tipo I Obesidade Tipo II Obesidade Tipo III
Quantidade
Faixa Etária
00-18
19-30
31-45
Acima de 45
351
112
68
61
49
1461
296
232
208
201
201
166
157
286
83
80
65
13
Gênero / Faixa Etária / Calorias
Histórico Familiar / Atividade Física
Nível de Obesidade por Faixa Etária
IMC / Vegetais / Tecnologia / Refeições
Transporte / Obesidade Tipo I, II ou III


IMC Médio por Nível de Obesidade
IMC médio
Nível de Obesidade
Insuficiente
Normal
Sobrepeso Tipo I
Sobrepeso Tipo II
Obesidade Tipo I
Obesidade Tipo II
Obesidade Tipo III
17
22
26
28
32
37
42
Consumo Médio de Vegetais por Nível de Obesidade
Consumo médio de vegetais
2,17
2,33
2,01
2,04
1,98
1,92
3,00
Tempo Médio de Uso de Tecnologia por Nível de Obesidade
Nível de Obesidade
Tempo médio de uso de tecnologia
0,58
0,68
0,35
0,40
0,44
0,23
0,04
Distribuição do Número de Refeições Diárias por Nível de Obesidade
Refeições Diárias 1 2 3 4
Quantidade de Indivíduos
272
175
49
287
206
52
290
156
40
86
290
136
77
75
351
180
71
100
297
185
78
33
324
324
Gênero / Faixa Etária / Calorias
Histórico Familiar / Atividade Física
Nível de Obesidade por Faixa Etária
IMC / Vegetais / Tecnologia / Refeições
Transporte / Obesidade Tipo I, II ou III



METODOLOGIAS:

Esta metodologia descreve, de ponta a ponta, o processo aplicado para transformar a base obesity.csv em um ecossistema preditivo para apoio à decisão clínica, incluindo curadoria e padronização de dados, pipeline de Machine Learning, comparação de modelos, seleção do melhor classificador, deploy em formulário (Streamlit) com suporte consultivo via IA Generativa via API, persistência em banco (SQLite) e visão analítica (Power BI).

Enquadramento do problema e requisitos do desafio:

O desafio propõe um cenário hospitalar em que o cientista de dados deve desenvolver um modelo preditivo de obesidade utilizando a base disponibilizada e entregar um sistema preditivo para apoiar a equipe médica.

Dados: origem, dicionário e versões utilizadas:

A base de entrada fornecida no desafio é obesity.csv, com variáveis demográficas, antropométricas e comportamentais, e uma coluna-alvo relacionada ao nível de obesidade. O dicionário inclui, entre outras, Gender, Age, Height, Weight, family_history, FAVC, FCVC, NCP, CAEC, SMOKE, CH2O, SCC, FAF, TER/tempo de dispositivos, CALC e MTRANS, além da coluna-alvo de obesidade.

Base curada/normalizada (dados de treino):

Para garantir reprodutibilidade do treinamento e padronização do consumo do dataset, foi utilizada uma versão tratada denominada Obesity_normalizado_ok.csv, definida explicitamente como fonte de treino no script de modelagem.

Uma regra importante de governança foi aplicada: se existirem colunas auxiliares com prefixo N_, elas não entram no conjunto final de features, evitando redundância e "duplicação semântica". O script seleciona apenas colunas não N_ e mantém a coluna-alvo.

Tecnologias e stack empregadas:

Camada de ciência de dados e modelagem.

- Python como linguagem de implementação;
- Pandas/Numpy para manipulação numérica e de dados;
- Scikit-learn para pipeline, pré-processamento e modelos clássicos (LogReg, Tree, KNN, RF, GB, AdaBoost, SVM, Naive Bayes);
- XGBoost (XGBClassifier) como candidato de alto desempenho em ensembles;

treino_de_modelo

- Joblib para serialização do pipeline treinado e do encoder;

Camada de produto (deploy e decisão):

- Streamlit (formulário clínico e inferência);
- SQLite (persistência local de respostas e recomendações) via sqlite3;
- Generative AI via API para suporte consultivo pós-predição.
- Power BI (painel analítico), acessado via link publicado no portal HTML;
- HTML/JavaScript (index.html) para portal unificado com embed em iframe do Streamlit e do Power BI.

Curadoria e limpeza dos dados (tratamento de dados brutos):

A ingestão e limpeza foram implementadas no script treino_de_modelo.py para lidar com problemas práticos (separador, tipos, ausentes), garantindo um dataset consistente antes do treinamento.

Seleção de colunas e prevenção de redundância:

Após leitura, o script monta o conjunto final de features removendo colunas N_ e mantendo a coluna-alvo Obesity.

Feature engineering e padronização do pré-processamento (pipeline reprodutível):

Para garantir consistência entre treino e produção, toda transformação foi encapsulada em pipeline, evitando divergências entre "pré-processamento manual" e "inferência em produção".

Geração de variável derivada: IMC (BMI):

Foi criado um transformer customizado BMICalculator que:

- substitui altura 0 por NaN para evitar divisão por zero;
- preenche altura ausente com mediana;
- calcula BMI = Weight / Height^2 e adiciona como feature.

Pré-processamento por tipo de variável:

O pipeline utiliza ColumnTransformer com.

- StandardScaler nas numéricas (incluindo BMI);
- OneHotEncoder(handle_unknown='ignore') nas categóricas para tratar variáveis nominais sem perda de informação e tolerância a categorias novas no deploy.

Como o problema é multiclasse com rótulos textuais, foi aplicado LabelEncoder ao target e o encoder foi salvo como label_encoder.pkl para garantir decodificação consistente no Streamlit.

Split treino/teste:

A validação foi realizada via holdout estratificado (80%/20%), preservando a proporção das classes no conjunto de teste e fixando RANDOM_SEED.

O script imprime explicitamente os tamanhos de treino e teste, reforçando rastreabilidade da rodada de avaliação.

Modelagem preditiva: benchmarking e seleção do melhor modelo:

A seleção do modelo não foi feita por escolha subjetiva, mas por benchmarking de múltiplas famílias de algoritmos, todos treinados sob o mesmo pipeline (feature engineering + pré-processamento + classificador).

Conjunto de modelos avaliados:

O script define um dicionário de classificadores incluindo:

- Logistic Regression (baseline)
- Decision Tree
- KNN
- Random Forest
- XGBoost
- Gradient Boosting
- AdaBoost
- SVM (RBF)
- Naive Bayes

Pipeline por modelo:

Para cada classificador, é construída uma pipeline com as etapas: BMICalculator → preprocessor → classifier, garantindo padronização total da comparação.

Métricas de avaliação (indicadores do modelo):

Para cada modelo, são calculados e impressos:

- Acurácia (métrica principal de cumprimento do requisito >75%);
- Classification report (precision, recall, f1-score por classe);

- Matriz de confusão (mapeamento de erros entre classes).

Critério de escolha e resultado do experimento (acurácia):

O melhor modelo é selecionado automaticamente pelo maior valor de acurácia e o script imprime um comparativo final com o vencedor.

Resultados da execução fornecida por você (rodada reportada no log do projeto):

- Logistic Regression (Baseline): 89,60%
- Decision Tree: 97,16%
- KNN: 91,02%
- Random Forest: 97,40%
- XGBoost: 97,87% (melhor)
- Gradient Boosting: 97,40%
- AdaBoost: 42,55%
- SVM (RBF): 61,23%
- Naive Bayes: 69,27%

Conclusão metodológica: o modelo escolhido para produção foi o XGBoost, por apresentar a maior acurácia no conjunto de teste nessa rodada, superando com folga o critério mínimo.

Serialização e artefatos para produção (modelo "pronto para deploy"):

Após identificar o melhor modelo, o pipeline correspondente é salvo em disco como:

- obesity_prediction_model_pipeline.pkl

  O encoder do target é salvo como:

- label_encoder.pkl

Esses dois artefatos são essenciais para o deploy: o primeiro executa a inferência com o mesmo pré-processamento do treino; o segundo converte a classe numérica prevista de volta ao rótulo clínico.

Deploy no Formulário (Streamlit) e uso do modelo em tempo real:

O sistema preditivo foi implementado em app_streamlit_medico.py, operando como CDSS (Clinical Decision Support System).

Coleta estruturada das variáveis:

O formulário coleta as variáveis de entrada e aplica mapeamentos de interface (Português → categorias esperadas pelo dataset/modelo), por exemplo:

- "Sim/Não" → yes/no;

  app_streamlit_medico
- categorias de CAEC/CALC (Não/Às vezes/Frequentemente/Sempre) → valores categóricos do dataset;
- MTRANS (Transporte Público/Andar/Automóvel/Bicicleta/Outros) → rótulos do dataset;

Inferência e apresentação do nível previsto:

Ao acionar "Obter Predição e Suporte", o app:

  monta um DataFrame com as entradas;

  remove o campo Name (não utilizado pelo modelo);

  executa predict() no pipeline carregado;

- decodifica o resultado com o LabelEncoder e apresenta o nível de obesidade.

Ponto-chave: o "nível de obesidade" do formulário é calculado pelo pipeline serializado (o mesmo selecionado no benchmarking).

IA Generativa via API como suporte à decisão clínica (pós-predição):

A IA Generativa não substitui o classificador. Ela atua como camada consultiva após o modelo prever a classe.

Prompt clínico estruturado:

O prompt de IA incorpora o perfil do paciente e orienta a resposta a focar em dieta, exercícios e estilo de vida, em linguagem útil para decisão.

Persistência e rastreabilidade (SQLite):

Para auditoria e histórico clínico, o app salva:

- respostas do paciente + classe prevista na tabela respostas;
- texto do Gemini na tabela gemini_responses;
- em um arquivo SQLite obesity_data.db.

Isso cria uma base de acompanhamento para revisões futuras e análise de resultados em ambiente real.

Portal unificado e Dashboard (visão gerencial):

A entrega é apresentada ao usuário final via um portal index.html, que oferece dois acessos:

- Formulário Avaliativo (Streamlit, embed em iframe);
- Dashboard Estatísticas (Power BI publicado).

O HTML implementa a abertura do sistema via overlay/iframe e um botão "voltar", organizando a experiência de navegação para demonstração do projeto.

Em síntese, a abordagem metodológica adotou um fluxo completo e reprodutível:

I - Curadoria e padronização do dataset .

II - Feature engineering (IMC) incorporado ao pipeline.

III - Pré-processamento formal (scaling + one-hot encoding).

IV - Validação por holdout estratificado.

V - Benchmarking de múltiplos algoritmos e seleção automática do melhor classificador.

VI - Serialização do pipeline e do encoder para produção

VII - Deploy em CDSS (Streamlit) com inferência em tempo real e camada consultiva via IA.

VIII - Persistência em SQLite para rastreabilidade.

IX - Consolidação em portal HTML com dashboard Power BI para visão de negócio.

CONCLUSÃO:

Este projeto entregou uma solução completa e operacional para apoio à decisão clínica em obesidade, indo além de um experimento de modelagem e consolidando um ecossistema end-to-end: dados → pipeline preditiva → aplicação clínica → suporte consultivo por IA → rastreabilidade em banco → visão analítica por dashboard. O resultado incluindo a implementação de um sistema preditivo em produção (Streamlit) e uma camada analítica de negócio (Power BI).

Do ponto de vista técnico, a solução foi construída sobre uma pipeline reprodutível com curadoria de dados, pré-processamento estruturado e feature engineering incorporado (IMC calculado de forma consistente em treino e inferência).

Essa abordagem reduziu riscos de divergência entre "modelo treinado" e "modelo em produção", sustentando a robustez do sistema ao longo do ciclo de vida.

Na etapa de modelagem, foi adotado benchmarking de múltiplos algoritmos e seleção empírica do classificador mais performático. Na execução reportada, o melhor desempenho foi obtido pelo XGBoost, com acurácia de 97,87% no conjunto de teste, superando com ampla margem o critério mínimo de assertividade exigido pelo desafio (>75%).

Em termos de aplicação, o formulário clínico implementado em Streamlit utiliza diretamente o artefato do pipeline treinado (obesity_prediction_model_pipeline.pkl) para gerar o nível de obesidade em tempo real, assegurando padronização na classificação e reduzindo variação na triagem. A IA generativa (Gemini) atua como camada consultiva pós-predição: não substitui o classificador, mas transforma o resultado do modelo em recomendações orientadas à conduta (hábitos, rotina e mudanças comportamentais), aumentando a utilidade prática no contexto assistencial.

Além disso, o sistema registra respostas e recomendações em banco SQLite, criando rastreabilidade e histórico auditável, o que habilita monitoramento, revisões e evolução futura do modelo com dados reais do hospital. Por fim, o dashboard em Power BI consolida a camada gerencial, fornecendo evidências e padrões para apoiar decisões

de prevenção e alocação de recursos, completando a visão de valor do projeto para operação clínica e gestão.

Limitações e continuidade (proposta objetiva):

Como continuidade natural para ambiente real, recomenda-se: (i) validação externa com dados do hospital, (ii) monitoramento de deriva de dados e performance ao longo do tempo, e (iii) calibração de métricas por classe para reduzir eventuais confusões entre categorias adjacentes. Esses passos consolidam a transição de uma solução acadêmica para um CDSS com governança e validação clínica contínua.

Em síntese, o projeto demonstra como Data Analytics e IA podem ser aplicadas de forma prática à saúde: predizer risco com alta precisão, padronizar triagem, gerar recomendações orientadas e fornecer inteligência gerencial, convertendo dados em ações com impacto direto no cuidado e na eficiência operacional.

RECOMENDAÇÕES (OPERACIONAIS E DE GOVERNANÇA):

A seguir estão recomendações objetivas para evoluir a solução de um protótipo acadêmico para um CDSS mais próximo de produção, mantendo rastreabilidade, segurança e valor clínico.

Recomendações clínicas e operacionais (uso no hospital).

Implantar como ferramenta de triagem e acompanhamento, não como diagnóstico definitivo: o nível de obesidade deve ser interpretado como classificação de apoio, sempre com validação do profissional.

Definir protocolo de ação por classe (Insufficient → Obesity III): para cada nível, padronizar encaminhamentos (nutrição, educação física, endocrinologia), metas iniciais e frequência de retorno.

Criar checklist mínimo de anamnese associado ao formulário: garantir que as variáveis comportamentais capturadas sejam consistentes entre profissionais.

Treinamento rápido da equipe: orientar como interpretar probabilidade/classe e como usar o texto de Suporte a Decisão (IA) como “guia de conversa” com o paciente (não prescrição).

Anexos (Links úteis):

No Gitnub:

Fase 4 (todos os arquivos):

https://github.com/LuFaiotto/tech-challenge/tree/main/Fase_4

Menu Interativo (Formulário Avaliativo + Dashboard)

https://lufaiotto.github.io/tech-challenge/Fase_4/

Bases de Dados – SQL - Gold:

https://github.com/LuFaiotto/tech-challenge/blob/main/Fase_4/streamlit/obesity_data.db

Dados Históricos Silver:

https://github.com/LuFaiotto/tech-challenge/blob/main/Fase_4/streamlit/Obesity_normalizado_ok.csv

Dados Históricos Bronze:

https://github.com/LuFaiotto/tech-challenge/blob/main/Fase_4/streamlit/Obesity.csv

No YouTube:

Vídeo de Apresentação:

https://youtu.be/aGr4Ih86w4E