Modelos de Machine Learning e Processo de Seleção

Contexto e Dataset

- Base de obesidade (variáveis clínicas e de estilo de vida) documentada em `doc/dicionario_obesity_fiap.pdf`.
- Pré-processamento definido no notebook `analise/model_comparison.ipynb`: limpeza, cálculo de IMC, binning de variáveis contínuas para faixas interpretáveis e normalização de nomes/labels para português.

Modelos Avaliados

- **RandomForest (com e sem SMOTE)**: ensamble de árvores, robusto a não linearidades; a versão com SMOTE buscou mitigar desbalanceamento, mas apresentou ganho limitado em relação ao custo computacional.
- **LightGBM (LGBMClassifier)**: gradiente de árvores com folhas otimizadas; bom tempo de treino e capacidade para variáveis categóricas após one-hot.
- **XGBoost**: boosting de gradiente com regularização; avaliado com `multi:softprob`, 400 estimadores, profundidade 5, taxas de amostragem em 0.8.

Métricas e Critério

- Métricas monitoradas: **Acurácia**, **F1-macro** e **Recall para classes de risco** (sobrepeso e obesidade). O notebook imprime as três métricas por modelo (`accuracy`, `f1`, `recall_risk`) e gera um `summary` ordenado por recall de risco.
- Critério de seleção: melhor equilíbrio entre recall de risco (prioridade clínica), acurácia global e ausência de overfitting (desempenho semelhante entre treino e validação).

Resultado da Comparação

- O XGBoost apresentou o melhor recall de risco e acurácia global entre os candidatos, sem sinal de overfitting (performance consistente em validação).

- RandomForest (com/sem SMOTE) e LightGBM ficaram atrás em recall de risco, embora LightGBM tenha mantido bom tempo de treinamento.

Pipeline Final (usado no app)

- **Pré-processamento**: `engineer_features` calcula IMC, aplica binning nas variáveis contínuas e preenche valores faltantes; encode de categorias com label encoder salvo no artefato.
- **Modelo**: classificador XGBoost treinado e serializado com `dill` em `models/xgboost.pkl` (contém pipeline + label encoder).
- **Inferência no Streamlit** (`app.py`):
  i. Coleta de dados + cálculo de IMC com feedback de faixa.
  ii. Transformação de entrada pela pipeline carregada.
  iii. Predição de probabilidades por classe e cálculo de risco agregado.
  iv. Visualização em gráfico horizontal (Altair) e geração de relatório HTML para download/impressão.

Reprodutibilidade

- Reexecute o notebook `analise/model_comparison.ipynb` com as dependências de `requirements.txt` para retrenar/atualizar métricas.
- Salve o novo artefato com `dill` em `models/xgboost.pkl` mantendo a mesma assinatura (pipeline + label encoder) para compatibilidade com o app.