### **Documentação do Projeto: Análise de Clientes e Classificação de Atividade (Academia RedFit)**

*Arthur Trindade*

#### **1. Objetivo do Projeto**

A academia RedFit deseja conhecer melhor o perfil de seus clientes para direcionar campanhas de marketing de forma mais eficaz. O objetivo é desenvolver um modelo de Machine Learning capaz de classificar os usuários em diferentes níveis de atividade física: **Sedentário, Ativo ou Atleta**.

#### 

#### **2. Processo de ETL (Extração, Transformação e Carga)**

O ETL é a base do nosso projeto, onde preparamos os dados brutos para que se tornem informações úteis para análise e modelagem.

**2.1. Extração (Extraction)**

A primeira etapa consiste em carregar os dados do arquivo academia\_redfit.csv para o nosso ambiente de análise.

* **Código:** df = pd.read\_csv("academia\_redfit.csv")
* **Explicação:** Utilizamos a biblioteca pandas para ler o arquivo CSV. Os dados são carregados para uma estrutura de tabela (DataFrame) que chamamos de df, permitindo a sua manipulação em Python.

**2.2. Transformação (Transformation)**

Nesta fase, realizamos a limpeza, padronização e enriquecimento dos dados.

* **Tratamento de Dados Vazios:** Identificamos que algumas colunas numéricas (tempo\_medio\_exercicio, etc.) possuíam valores em falta. Para resolver isso, preenchemos esses campos com a **mediana** da coluna correspondente, garantindo que não tenhamos dados faltantes no modelo.
* **Padronização de Categorias:** Garantimos que o texto de colunas como sexo, tipo\_atividade e estado seguisse um padrão consistente (primeira letra maiúscula), para evitar que o modelo trate "Yoga" e "yoga" como atividades diferentes, por exemplo.
* **Conversão de Datas:** A coluna data\_matricula foi convertida do formato de texto para o formato datetime. Isso permite que o Python entenda que se trata de uma data, possibilitando cálculos futuros, como o tempo de permanência do cliente na academia.
* **Criação de Novas Métricas (Feature Engineering):**
  + **evolucao\_PGC**: Criamos esta coluna para medir o progresso do cliente, calculando a diferença entre o primeiro\_PGC e o ultimo\_PGC. Um valor positivo indica uma redução de gordura corporal.
  + **nivel\_atividade (Variável Alvo)**: Como o objetivo é classificar os clientes em três níveis, criamos esta nova coluna com base em regras de negócio:
    1. Clientes no estado "Sedentária" são classificados como **sedentários**.
    2. Clientes que treinam 5 ou mais vezes por semana e possuem um PGC final abaixo de 15% são classificados como **Atleta**.
    3. Os restantes são classificados como **ativos**.
  + **Observação sobre o IMC:** O enunciado pedia o cálculo do IMC (Índice de Massa Corporal). No entanto, a fórmula requer peso e altura, e como a base de dados **não contém estas colunas**, não foi possível criar esta métrica.

**2.3. Carga (Load)**

Após a transformação, guardamos o conjunto de dados tratado.

* **Código:** df.to\_csv("academia\_redfit\_tratado.csv", index=False)
* **Explicação:** O DataFrame df, agora limpo e com as novas colunas, é guardado num novo arquivo chamado academia\_redfit\_tratado.csv.

#### **3. Análise Exploratória e Visualização**

Antes de construir o modelo, é fundamental explorar os dados para extrair insights.

* **Gráfico de Barras:** Geramos um gráfico de barras para visualizar a quantidade de clientes em cada um dos novos níveis de atividade (Sedentário, Ativo, Atleta). Isso ajuda-nos a entender a proporção de cada perfil de cliente na academia.
* **Gráfico de Pizza:** Criamos um gráfico de pizza para mostrar a distribuição percentual de clientes por sexo. Este tipo de gráfico é ideal para mostrar a composição de um todo.

#### **4. Construção do Modelo de Classificação**

Esta é a fase onde utilizamos ML para criar um modelo preditivo.

1. **Preparação e Pré-processamento:**
   * Separamos os dados em X (as características que usamos para prever) e y (o que queremos prever: nivel\_atividade). Colunas que não são úteis para a previsão (como data\_matricula) são removidas.
   * Utilizamos um ColumnTransformer para aplicar tratamentos diferentes a colunas diferentes:
     + StandardScaler: Padroniza as colunas numéricas.
     + OneHotEncoder: Converte colunas de texto em um formato numérico.
2. **Divisão, Treino e Avaliação:**
   * **train\_test\_split**: Dividimos os dados em 80% para treinar o modelo e 20% para o testar, garantindo uma avaliação justa.
   * **Treino (.fit)**: Treinamos um modelo RandomForestClassifier para que ele aprenda os padrões nos dados de treino.
   * **Avaliação (accuracy\_score)**: Medimos a **acurácia** do modelo, que representa a percentagem de previsões corretas nos dados de teste.
3. **Guardar o Modelo Final:**
   * **Código:** joblib.dump(best\_model, "redfit\_model.pkl")
   * **Explicação:** O modelo treinado é guardado num ficheiro .pkl. Este ficheiro pode ser posteriormente carregado para classificar novos clientes sem a necessidade de treinar o modelo do zero.