**AmIPhat: Dokumentacja**

**Przegląd**

**AmIPhat** to aplikacja webowa zaprojektowana do przewidywania poziomu otyłości na podstawie danych wprowadzonych przez użytkownika. Aplikacja zapewnia kompatybilność ze wszystkimi aktualnymi urządzeniami, będąc dostępną za pomocą najnowszych przeglądarek internetowych.

**Dataset**

Aplikacja wykorzystuje zbiór danych "Obesity Levels" pobrany z Kaggle. Zbiór danych składa się z 17 atrybutów i 2111 rekordów, a zmienna zależna reprezentuje poziom otyłości. Kluczowe zmienne obejmują:

Obraz zawierający tekst, zrzut ekranu, menu, numer

Opis wygenerowany automatycznie

**Predykcja**

Główna funkcjonalność aplikacji polega na przewidywaniu stopnia otyłości na podstawie danych wprowadzonych przez użytkownika.

Stopien otyłości jest opisywany w danych wartościach:

* Insufficient\_Weight,
* Normal\_Weight,
* Obesity\_Type\_I,
* Obesity\_Type\_II,
* Obesity\_Type\_III,
* Overweight\_Level\_I,
* Overweight\_Level\_II

**Uzasadnienie funkcjonalności**

**AmIPhat** służy wielu celom:

1. **Specjaliści z dziedziny zdrowia**:
   * Lekarze i dietetycy mogą wykorzystać aplikację do oceny ryzyka otyłości u swoich pacjentów.
   * Szybka prognoza otyłości umożliwia pracownikom służby zdrowia szybsze wdrożenie planu leczenia.
2. **Użytkownicy indywidualni**:
   * Osoby mogą samodzielnie ocenić, czy ich obecny styl życia naraża je na ryzyko otyłości.
   * Aplikacja może zachęcić użytkowników do podejmowania bardziej świadomych decyzji dotyczących diety i stylu życia.

**Model uczenia maszynowego**

Aplikacja wykorzystuje model drzewa decyzyjnego, który jest powszechnie stosowanym algorytmem uczenia maszynowego. Ten model został wybrany ze względu na swoją przydatność do analizy tego rodzaju danych. Drzewa decyzyjne są skuteczne zarówno w przypadku danych kategorycznych, jak i numerycznych, a także dostarczają jasne, interpretowalne wyniki.

**Struktura projektu**

Struktura projektu jest zorganizowana w następujący sposób:

* **API**
  + fastapi: Zawiera pliki serwera FastAPI.
    - libs: Pliki biblioteki do ładowania modelu uczenia maszynowego.
    - ml\_model: Zawiera zserializowany model drzewa decyzyjnego.
    - app.py zawiera konfiguracje FastApi
    - requirements.txt: Lista zależności dla serwera FastAPI.
  + streamlit: Zawiera pliki serwera Streamlit dla interfejsu webowego.
* **conf**
  + create\_env\_conda.ps1: Skrypt do tworzenia środowiska Conda.
  + environment.yml: Plik konfiguracyjny dla środowiska Conda.
* **data**
  + kaggle.json: Klucz API do uzyskiwania dostępu do datasetów Kaggle.
  + dataset: Zawiera pliki datasetu "Obesity Levels".
* **models**
  + model.joblib: Zawiera zserializowany model drzewa decyzyjnego.
* **src**
  + modules: Kod źródłowy do przetwarzania danych i zadań z zakresu nauki o danych.
    - data\_processing: Skrypty do przetwarzania datasetu.
      * clean\_data.py: Oczyszcza surowe dane.
      * download\_dataset.py: Pobiera dataset z Kaggle.
      * prepare\_data.py: Przygotowuje dane do treningu.
      * preprocess\_data.py: Przetwarza dane.
      * split\_data.py: Dzieli dane na zbiory treningowe i testowe.
    - data\_science: Skrypty do treningu i ewaluacji modelu uczenia maszynowego.
      * evaluate\_model.py: Ewaluacja wytrenowanego modelu.
      * save\_model.py: Zapisuje wytrenowany model.
      * show\_model\_data.py: Wizualizacja danych modelu.
      * training\_model.py: Trenuje model uczenia maszynowego.
  + main.py: Główny skrypt do uruchomienia aplikacji.
* **.gitignore**: Określa, które pliki i katalogi mają być ignorowane w kontroli wersji.
* **README.md**: Zawiera przegląd i instrukcje dotyczące projektu.