**Heart Failure Prediction Model**

<https://github.com/mijurek/Heart-Failure-Prediction-Model/>

**Autorzy:**

Michał Jurek

Jakub Dudajek

**Spis treści**

[I. Charakterystyka Oprogramowania 2](#_Toc187435587)

[II. Dane 2](#_Toc187435588)

[III. Specyfikacja wymagań 3](#_Toc187435589)

[Wymagania funkcjonalne 3](#_Toc187435590)

[Wymagania pozafunkcjonalne 3](#_Toc187435591)

[IV. Architektura oprogramowania 3](#_Toc187435592)

[Jupyter Notebook 3](#_Toc187435593)

[CLI: 3](#_Toc187435594)

[Wykorzystane technologie 3](#_Toc187435595)

[V. Testy 4](#_Toc187435596)

[Walidacja modelu: 4](#_Toc187435597)

[Testy CLI: 4](#_Toc187435598)

[VI. Instrukcja obsługi aplikacji CLI 4](#_Toc187435599)

# Charakterystyka Oprogramowania

|  |  |
| --- | --- |
| **Nazwa projektu** | Heart Failure Prediction Model |
| **Opis projektu** | Projekt ma na celu opracowanie narzędzia wspierającego diagnostykę niewydolności serca poprzez predykcję ryzyka zgonu pacjentów na podstawie danych klinicznych. Głównym celem jest identyfikacja kluczowych czynników ryzyka oraz wspieranie personelu medycznego w podejmowaniu decyzji diagnostycznych i terapeutycznych.  Model wykorzystuje algorytm XGBoost, który pozwala na skuteczną analizę złożonych danych i jest zoptymalizowany pod kątem niezrównoważonych zbiorów danych. |
| **Prawa autorskie i warunki licencyjne** | Kod źródłowy jest objęty licencją MIT, co pozwala na dowolne korzystanie, modyfikację oraz dystrybucję oprogramowania przy zachowaniu informacji o autorach. |
| **Wykorzystane technologie** | **Język programowania**: Python  **Biblioteki**:   * Dane i modelowanie: pandas, numpy, shap, xgboost, sklearn * Wizualizacja danych: matplotlib, seaborn * Strojenie i analiza: scipy, pickle |
| **Narzędzia** | Jupyter Notebook  CLI (Comman-Line Interface) |

# Dane

Dane pochodzą z Kaggle - [**https://www.kaggle.com/code/ecemboluk/heart-attack-prediction-with-classifier-algorithms/notebook**](https://www.kaggle.com/code/ecemboluk/heart-attack-prediction-with-classifier-algorithms/notebook)

Obejmują 299 obserwacji pacjentów z niewydolnością serca. Każdy przypadek zawiera 13 cech klinicznych.

* **age**: wiek pacjenta (lata)
* **anaemia**: obecność anemii (0 – brak, 1 – obecna)
* **creatinine\_phosphokinase**: poziom enzymu CPK we krwi (mcg/L)
* **diabetes**: obecność cukrzycy (0 – brak, 1 – obecna)
* **ejection\_fraction**: frakcja wyrzutowa serca (procent)
* **high\_blood\_pressure**: nadciśnienie tętnicze (0 – brak, 1 – obecne)
* **platelets**: liczba płytek krwi (kilopłytki/mL)
* **serum\_creatinine**: poziom kreatyniny w surowicy (mg/dL)
* **serum\_sodium**: poziom sodu w surowicy (mEq/L)
* **sex**: płeć (0 – kobieta, 1 – mężczyzna)
* **smoking**: palenie tytoniu (0 – nie pali, 1 – pali)
* **time**: czas obserwacji (dni)
* **DEATH\_EVENT**: zdarzenie śmiertelne podczas okresu obserwacji (0 – przeżył, 1 – zmarł)

# Specyfikacja wymagań

## Wymagania funkcjonalne

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Nazwa** | **Opis** | **Priorytet** | **Kategoria** |
| **WF1** | Wczytywanie danych | Import danych z pliku csv do obiektu DataFrame w Pandas. | Wyoski | Importowanie |
| **WF2** | Trening modelu | Trening modelu XGBoost z uwzględnieniem optymalizacji hiperparametrów. | Wysoki | Modelowanie |
| **WF3** | Wizualizacja wyników | Generowanie wykresów istotności cech, macierzy pomyłek i analizy SHAP. | Średni | Analizowanie |
| **WF4** | Predykcja | Implementacja interfejsu CLI do zbierania danych od użytkownika i generowania prognoz. | Wysoki | Interfejs użytkownika |

## Wymagania pozafunkcjonalne

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Nazwa** | **Opis** | **Priorytet** | **Kategoria** |
| **WNF1** | Wydajność | Model trenuje się i przewiduje wyniki w akceptowalnym czasie. | Wysoki | Optymalizacja |
| **WNF2** | Jakość | Aplikacja działa niezawodnie i bez błędów. | Wysoki | Niezawodność |
| **WNF3** | Ochrona danych | Dane wejściowe są chronione przed dostępem osób nieuprawnionych. | Wysoki | Bezpieczeństwo |

# Architektura oprogramowania

## Jupyter Notebook

* Budowanie modelu: wczytywanie danych, przetwarzanie ich, selekcja cech oraz trening modelu.
* Analiza wyników: wizualizacja, macierz pomyłek, analiza SHAP.
* Eksport modelu do pliku.

## CLI:

* Wczytywanie modelu.
* Pobieranie danych od użytkownika.
* Generowanie prawdopodobieństwa zgonu i prezentacja wyniku w terminalu.

## Wykorzystane technologie

* Python 3.11
* Sklearn, XGBoost, matplotlib, shap, pickle

# Testy

Zakres testów:

## Walidacja modelu:

* Krzywe uczenia się.
* Macierz pomyłek.

## Testy CLI:

* Wprowadzanie ręcznie danych.
* Generowanie prognozy na podstawie przykładowych danych.

# Instrukcja obsługi aplikacji CLI

1. Wprowadź wymagane dane.
2. Odczytaj wyniki przedstawiające prawdopodobieństwo zgonu.