

1. Charakterystyka oprogramowania

a. Nazwa skrócona:

Porównywarka modeli

b. Nazwa pełna:

Analiza porównawcza wybranych modeli uczenia maszynowego

c. Krótki opis:

Aplikacja wykorzystuje trzy różne zbiory danych związane z medycyną, aby wspierać diagnozowanie chorób. Po udzieleniu odpowiedzi na pytania przez użytkownika, program analizuje informacje i prezentuje prawdopodobieństwo wystąpienia danej choroby (raka płuc, cukrzyca, choroby serca). Wyniki są obliczane w oparciu o trzy wybrane modele uczenia maszynowego: regresję logistyczną, drzewo decyzyjne oraz las losowy.

2. Prawa autorskie

a. Autorzy:

- Adam Wrzątek

- Bartosz Deptuła

- Mikołaj Mazur

b. Warunki licencyjne do oprogramowania wytworzonego przez grupę

MIT License

Copyright (c) [2025] [Adam Wrzątek, Bartosz Deptuła, Mikołaj Mazur]

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

3. Specyfikacja wymagań

Identyfikator	Nazwa	Opis	Priorytet	Kategoria
I01	Model1: Drzewo decyzyjne	Podzielenie dostępnych danych na coraz mniejsze podzbiory na podstawie wartości wybranych cech.	Wymagane	funkcjonalne
I02	Model2: Las losowy	Tworzenie wielu drzew decyzyjnych na losowych podzbiorach danych, a następnie łączenie ich wyników	Wymagane	funkcjonalne
I03	Model3: Regresja Logistyczna	Metoda pozwalająca na modelowanie szansy i prawdopodobieństwa wystąpienia zdarzenia.	Wymagane	funkcjonalne
I04	Aplikacja	Aplikacja zbudowana w frameworku Flask, odpowiedzialna za realizację funkcjonalności związanych z przetwarzaniem danych i prezentacją wyników działania modeli.	Wymagane	funkcjonalne
I05	Dane	Dobranie trzech zbiorów danych dotyczących zagadnień związanych z medycyną	Wymagane	funkcjonalne
I07	Interfejs Graficzny	Graficzny interfejs użytkownika zapewniający intuicyjną obsługę aplikacji, z wizualizacją danych i wyników działania modeli.	Przydatne	funkcjonalne
I08	Porównywanie danych	Możliwość porównania modeli na przestrzeni różnych danych	Przydatne	funkcjonalne
I09	Zapisywanie prób	Zapisywanie otrzymanych prawdopodobieństw dla każdej próby	Przydatne	funkcjonalne
I10	Rejestracja oraz logowanie	Możliwość rejestracji kont oraz późniejsza możliwość zalogowania się do nich	Opcjonalne	funkcjonalne
I11	Model4: Klasteryzacja K-średnich	Podział zbioru danych na k klastrów przy wykorzystaniu centroidy	Opcjonalne	funkcjonalne

4. Architektura systemu/oprogramowania

Aplikacja opiera się na architekturze klient-serwer z trzema głównymi komponentami: frontendem (HTML, CSS), backendem (Flask Framework) oraz modelami uczenia maszynowego (las losowy, regresja logistyczna, drzewo decyzyjne). Jako główny język programowania został wykorzystany Python.

Frontend umożliwia użytkownikowi interakcję z systemem poprzez formularze wprowadzania danych, wyniki predykcji oraz historię poprzednich analiz.

Backend przetwarza dane użytkownika, obsługuje logikę aplikacji oraz komunikuje się z modelami uczenia maszynowego, które są dynamicznie wczytywane na podstawie

wybranego zbioru danych (np. choroby serca, cukrzyca, rak płuc). Dane wejściowe są odpowiednio skalowane, a wyniki predykcji są zwracane i zapisywane.

Aplikacja korzysta z bazy danych do przechowywania użytkowników, wprowadzonych danych oraz wyników predykcji poprzez wykorzystanie SQLAlchemy.

Architektura aplikacji wspiera filtrowanie wyników (np. wyświetlanie historii na podstawie zbioru danych), zarządzanie predykcjami (np. usuwanie wybranych analiz), a także prezentację wyników w sposób przejrzysty i zrozumiały dla użytkownika. Dzięki modułowej budowie, system można łatwo rozszerzać o nowe modele predykcyjne lub dodatkowe funkcjonalności.

Biblioteki użyte w aplikacji:

Biblioteka	Wersja	Opis	Użyte funkcje
Flask	3.1.0	Tworzenie aplikacji webowej	Flask, render_template, request, redirect, url_for, flash, Blueprint,
Flask-Login	0.6.3	Zarządzanie sesjami użytkowników i autoryzacją	LoginManager, login_required, current_user, login_user, logout_user, login_required, UserMixin
Flask-SQLAlchemy	3.1.1	Prosta konfiguracja bazy danych w aplikacji Flask	SQLAlchemy
DateTime	5.5	Obsługa dat i czasu	datetime
joblib	1.2.0	Ładowanie modeli uczenia maszynowego	joblib.load, joblib.dump
Werkzeug	3.1.3	Haszowanie haseł	generate_password_hash, check_password_hash
scikit-learn	1.3.0	Klasyfikacja, regresja, podział danych, ocena modeli	RandomForestClassifier, DecisionTreeClassifier, LogisticRegression, train_test_split, StandardScaler, classification_report
Pandas	2.0.3	Wczytywanie danych z plików csv	pd.read_csv