

LUNG RISK



UWARUNKOWANIA ZACHOROWALNOŚCI NA RAKA PŁUC

LungRisk

Dokumentacja wymagań aplikacji LungRisk zrealizowanej w ramach zajęć
Inżynierii oprogramowania w roku akademickim 2024/2025

I. Charakterystyka oprogramowania

Nazwa skrócona: LungRisk

Pełna nazwa: LungRisk - uwarunkowania zachorowalności na raka płuc

Krótki opis

LungRisk to aplikacja webowa opracowana za pomocą pakietu **R Shiny**, której celem jest interaktywna wizualizacja wpływu różnych czynników na zachorowalność na raka płuc. Użytkownicy mogą przeglądać wybrane modele analizy statystycznej i symulować wpływ zmiennych (np. wieku, płci, palenia tytoniu, alergie) na prawdopodobieństwo zachorowania.

II. Prawa autorskie

Autorzy

- Krzysztof Lesiewicz - programista R-shiny
- Beata Kisiel - analityczka danych
- Monika Kwiatkowska - specjalistka ds. UX/UI

Warunki licencyjne do oprogramowania

Freeware - możliwość bezpłatnego korzystania z oprogramowania bez mocy wprowadzania zmian w kodzie źródłowym

III. Specyfikacja wymagań

Wymagania funkcjonalne:

Identyfikator	Nazwa	Opis	Priorytet	Kategoria
WF1	Import danych	Możliwość zaimportowania własnego zbioru danych.	1	Funkcjonalne
WF2	Szacowanie parametrów modeli predykcyjnych	Oszacowanie parametrów opisujących zależności pomiędzy zmiennymi dla poszczególnych modeli.	1	Funkcjonalne
WF3	Interaktywne wizualizacje	Umożliwia użytkownikom generowanie wykresów pokazujących wpływ różnych zmiennych.	1	Funkcjonalne
WF3.1	Filtrowanie danych	Pozwala użytkownikowi wybierać zakresy i wartości zmiennych.	1	Funkcjonalne
WF4	Eksport wyników	Umożliwia eksport wygenerowanych analiz do pliku PDF lub CSV.	2	Funkcjonalne

Wymagania pozafunkcjonalne:

Identyfikator	Nazwa	Opis	Priorytet	Kategoria
WNF1	Wydajność	Aplikacja powinna działać płynnie przy obsłudze zaimportowanego zbioru danych.	1	Pozafunkcjonalne
WNF2	Dostępność	Kod źródłowy opublikowany na platformie GitHub, umożliwiający użytkownikom lokalne uruchomienie aplikacji.	1	Pozafunkcjonalne
WNF3	Skalowalność	Możliwość przeniesienia na większe serwery.	2	Pozafunkcjonalne

IV. Architektura systemu

Architektura rozwoju - stos technologiczny:

- Język programowania: **R**
- Framework: **R Shiny**
- Biblioteki:
 - Tworzenie aplikacji Shiny: **shiny, shinythemes**
 - Przetwarzanie i analiza danych: **dplyr, readxl, tidyverse, naniar**
 - Wizualizacja danych: **ggplot2, corrplot, RColorBrewer, gridExtra**
 - Modelowanie i analiza statystyczna: **car, lmtest, pscl, pROC**
 - Obsługa braków danych i diagnostyka danych: **VIM, naniar**
 - Specjalistyczne wizualizacje i analiza zależności: **vcd**
- Zarządzanie wersjami: **GitHub**

Architektura uruchomieniowa - stos technologiczny

- Hosting: **Shinyapps.io**
- Opcjonalne: własny serwer z **Dockerem**
- System operacyjny: Linux/Windows

V. Testy

Scenariusze testów wymagań funkcjonalnych:

Identyfikator	Nazwa	Scenariusz	Wynik
TWF1	Import danych	Sprawdzenie, czy zbiór danych został poprawnie wgrany do systemu.	Pozytywny
TWF2	Generowanie wykresów	Sprawdzenie, czy aplikacja poprawnie generuje wykres na podstawie wybranych danych wejściowych.	Pozytywny
TWF3	Filtrowanie danych	Test możliwości filtrowania zmiennych na wykresie.	Pozytywny
TWF4	Eksport wyników	Test eksportu danych analizy do pliku PDF i CSV.	Pozytywny

Scenariusze testów wymagań pozafunkcyjnych:

Identyfikator	Nazwa	Scenariusz	Wynik
TWNF1	Test wydajności	Test płynności działania aplikacji przy obsłudze zaimportowanego zbioru danych.	Pozytywny
TWNF2	Test dostępności	Sprawdzenie czy opublikowany kod źródłowy jest dostępny do pobrania przez użytkowników.	Pozytywny