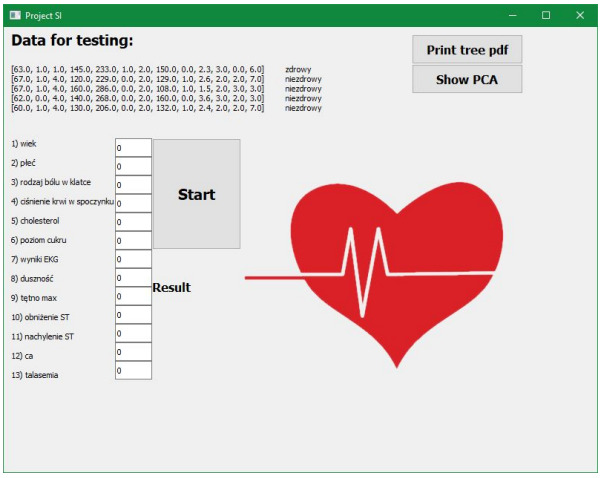
Wykrywanie chorób serca z użyciem algorytmu drzewa decyzyjnego

Władysław Jakołcewicz
Katarzyna Pietkiewicz

Cel projektu



Projekt ma na celu wykrywanie obecności chorób serca u pacjentów na podstawie danych medycznych przy użyciu algorytmu drzewa decyzyjnego. Przy jego użyciu można z wysokim prawdopodobieństwem wykryć chorobę serca u pacjenta na podstawie 13 atrybutów. Po wprowadzeniu do programu odpowiednich parametrów otrzymujemy informację czy program wykrył u pacjenta problemy kardiologiczne.

Dataset

W naszej implementacji użyliśmy zbioru danych „Heart Disease Data Set" dostępnego na stronie archive.ics.uci.edu /ml/datasets. Zbiór ten zawiera 76 atrybutów opisujących dane medyczne z których dla potrzeb naszego projektu wyselekcjonowaliśmy 14 najbardziej istotnych. Z dostępnych danych usunęliśmy atrybuty, które nie będą używane przez program oraz niepełne dane pacjentów w czego wyniku otrzymaliśmy 299 pełnych danych pacjentów zawierających 14 atrybutów do treningu i testów programu.

Atrybutem, którego wartość chcemy przewidzieć jest num- odpowiada on diagnozie choroby serca (wartość 0 dla braku wykrytej choroby serca i wartości 1, 2, 3, 4 dla wykrytej choroby serca), pozostałe trzynaście parametrów wpisywanych do programu pozwala określić czy podawane wyniki należą do pacjenta z problemami kardiologicznymi. Wybrane atrybuty mają wartości numeryczne, niektóre z nich mają przypisane wartości sztuczne (np. płeć, rodzaj bólu w klatce piersiowej).

Trudnością, która pojawiła się podczas implementacji rozwiązania była konieczność zredukowania bazy danych w przypadkach braku pełnego zestawienia atrybutów co skutkowało ograniczeniem ilości przypadków użytych do nauki naszego programu. W przypadku tego rodzaju rozwiązań ilość danych które służą do uczenia programu ma kluczowe znaczenie w skuteczności przewidywania rozwiązania.

Wybrane atrybuty:

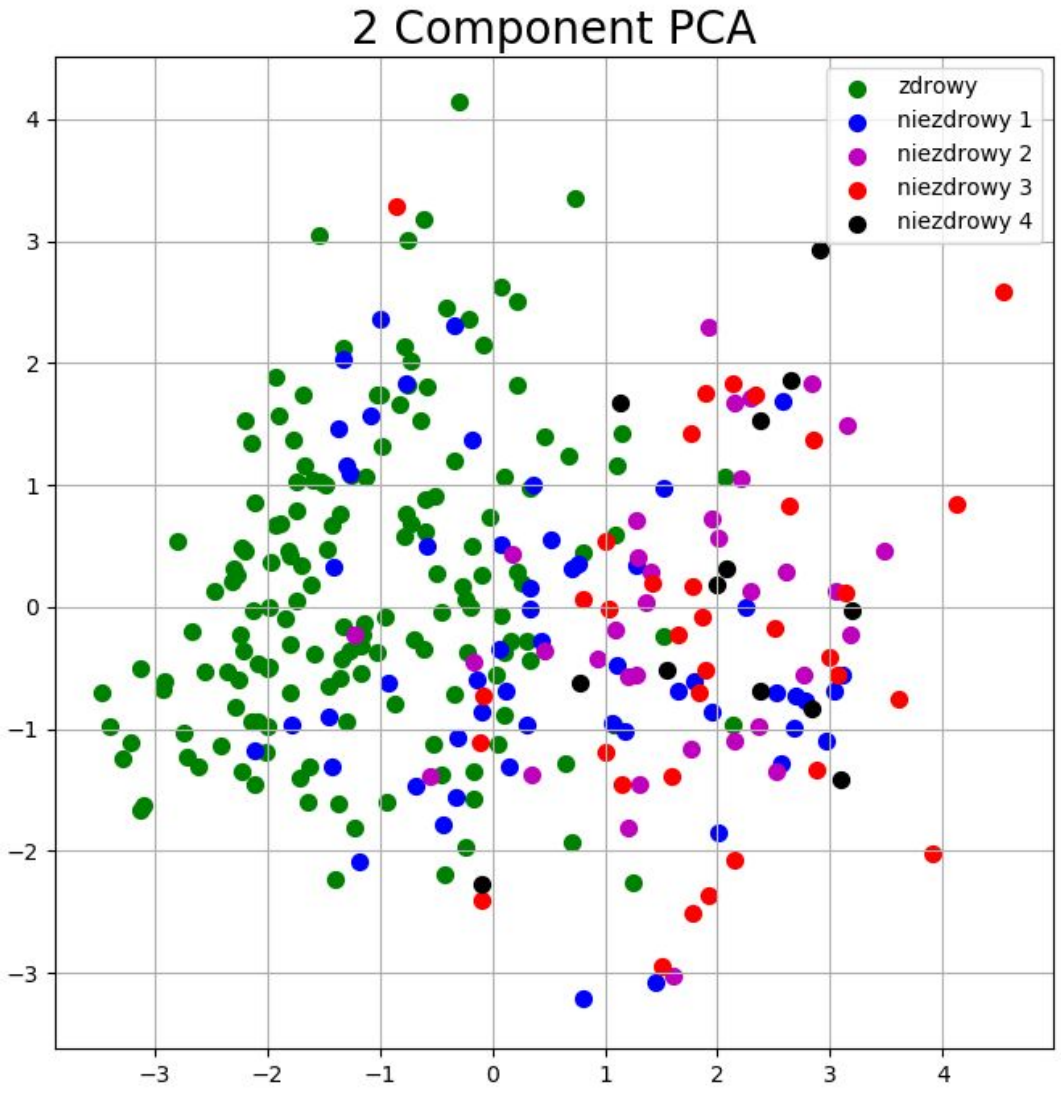
age - wiek pacjenta
sex - płeć pacjenta
(1 - mężczyzna
0 - kobieta)
cp - chest pain type- rodzaj bólu w klatce piersiowej
(1 - typowy dusznościowy
2 - nietypowy dusznościowy
3 – ból niedusznościowy
4 - bezobjawowy)
trestbps – resting blood pressure- ciśnienie krwi w spoczynku
chol - serum cholesterol in mg/dl – cholesterol w surowicy [mg/dl]

fbs- fasting blood sugar – poziom cukru we krwi na czczo
(1 – powyżej 120 mg/dl
0 – poniżej 120 mg/dl)
restecg - resting electrocardiographic results – spoczynkowe wyniki elektrokardiograficzne
(0 – w normie
1 – z nieprawidłowością fali ST-T (odwrócenie załamków T i / lub uniesienie odcinka ST o > 0,05 mV)
2 – wykazujące przerost lewej komory według kryteriów Estesa)
exang – exercise inducted angina – duszność wynikająca z testu wysiłkowego
(1 – zaobserwowana
0 - brak)
thalach – maksymalne tętno podczas testu wysiłkowego
oldpeak - ST depression induced by exercise relative to rest – obniżenie odcinka ST spowodowane testem wysiłkowym uzależnione od odpoczynku
slope - the slope of the peak exercise ST segment – nachylenie odcinka ST podczas odcinka szczytowego testu wysiłkowego
(1 - wzrost nachylenia
2 – brak zmian w nachyleniu
3 – obniżenie nachylenia)
ca - number of major vessels (0-3) colored by flouroscopy – ilość głównych naczyń zabarwionych za pomocą fluoroskopii
thal – thalassemia – talasemia (niedokrwistość tarczowatokrwinkowa)
(3 – w normie
6 – wada nieodwracalna
7 – wada odwracalna)
num – diagnosis of heart disease (angiographic disease status) - diagnoza choroby serca
(0 – choroba serca niewykryta
1, 2, 3, 4 – wykryta choroba serca)

age	sex	cp	tresrbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	num
63	1	1	145	233	1	2	150	0	2.3	3	0	6	0
67	1	4	160	286	0	2	108	1	1.5	2	3	3	2
67	1	4	120	229	0	2	129	1	2.6	2	2	7	1

przykładowe dane

Parametry, które w naszej implementacji są najbardziej istotne to **thal-** talasemia (niedokrwistość tarczowatokrwinkowa) oraz **ca**- ilość głównych naczyń zabarwionych za pomocą fluoroskopii.



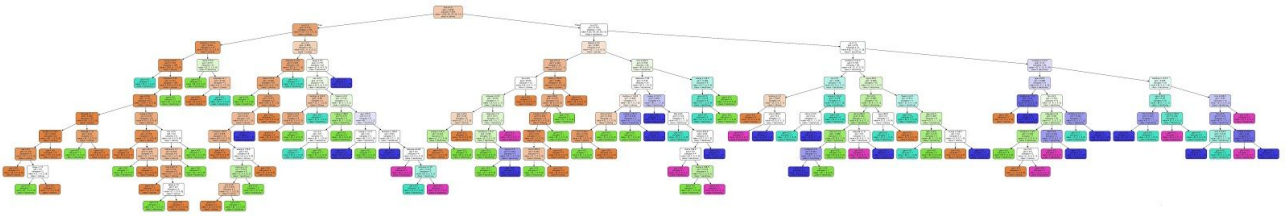
Wizualizacja PCA (Analiza głównych składowych)

Użyte rozwiązania

Projekt zaimplementowany jest w języku Python, a użytym przez nas narzędziem do uczenia maszynowego jest scikit-learn z biblioteką Tree. Scikit-learn jest oprogramowaniem Open source pozwalającym użytkownikowi na zgłębienie zagadnień związanych z technologią sztucznej inteligencji. Oferuje on wachlarz rozwiązań pozwalających na wprowadzenie i analizę danych przez program, który na ich podstawie uczy się i tym samym jest, w zależności od ilości i jakości otrzymanych danych, zdolny odpowiadać na zadawane pytania w oparciu o poznane wzorce.
Algorytm drzewa decyzyjnego stosowany jest w uczeniu maszynowym do pozyskiwania wiedzy na podstawie przykładów i do rozwiązywania problemu klasyfikacji. Jego działanie opiera się na analizie otrzymanych przykładów i na ich podstawie określeniu wartości szukanej.
Zaletami metody algorytmu drzewa decyzyjnego jest prostota jego interpretacji wynikająca m.in. z możliwości jego wizualizacji, niski koszt użycia w przypadku przewidywania danych (logarytmiczny zależny od ilości danych użytych do treningu drzewa), wysoka skuteczność w

przypadku dużej ilości danych wejściowych oraz niewielkie wymagania dotyczące przygotowania danych wejściowych.

Trudnością w implementacji tego rozwiązania jest brak możliwości korzystania z niepełnych zbiorów danych- drzewa decyzyjne nie obsługują brakujących danych. Innym problemem może być mniejsza dokładność w porównaniu do innych metod.

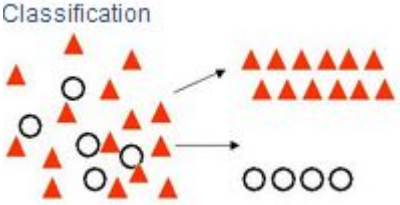


Schemat stworzonego na potrzeby programu drzewa decyzyjnego - załączony w pliku tree_d.pdf

Drzewa CART

Jednym z najpopularniejszych algorytmów jest algorytm CART, czyli Classification and Regression Tree. Jego autorami są Breiman, Friedman, Olshen oraz Stone, którzy wykorzystali w procesie decyzyjnym zastosowanie dwóch kryteriów podziału zebranych cech. Pierwszym z nich jest podział na dwie równe grupy. Kolejnym jest kryterium podziału Giniego, czyli mierze koncentracji zmiennej losowej.

Drzewa CART umożliwiają zarówno rozwiązywanie problemów regresyjnych jak i klasyfikacyjnych. Ich użycie pozwala na bardzo prostą interpretację wyników oraz tworzy prostszy model wyjaśniający sposób w jaki zostało otrzymane rozwiązanie. W naszym projekcie rozwiązujemy problem klasyfikacyjny.



Rozwiązywanie problemów klasyfikacyjnych polega na odnalezieniu kategorialnej zmiennej zależnej, której wartość chcemy przewidzieć na podstawie predykcyjnych zmiennych.

Źródła

[1] http://archive.ics.uci.edu/ml/datasets/heart+Disease – „Heart Disease Data Set”
1. Hungarian Institute of Cardiology. Budapest: Andras Janosi, M.D.
2. University Hospital, Zurich, Switzerland: William Steinbrunn, M.D.
3. University Hospital, Basel, Switzerland: Matthias Pfisterer, M.D.
4. V.A. Medical Center, Long Beach and Cleveland Clinic Foundation: Robert Detrano, M.D., Ph.D.
[2] http://scikit-learn.org/stable/ - Scikit-learn – oprogramowanie do uczenia maszynowego dla języka Python
[3] https://www.datasciencecentral.com
[4] https://github.com/Graidaris/decision_Tree_SI