Wykrywanie występowanie chorób serca,porównanie algorytów uczenia maszynowego nadzorowanego na podstawie zbioru danych dotyczących chorób układu krążenia z repozytorium UCI

Magdalena Szulc

Toruń,2022-05-01

Contents

Abstrakt

The aim of the work is to compare selected algorithms of supervised machine learning and build a model based on medical data, which diagnoses the presence or absence of cardiovascular disorders.

Medical data is distinguished by the fact that it is difficult to access, in most cases it is restricted information not available for public use. Therefore, a key step is to choose the features taken into account when creating the model. The data obtained from the UCI repository has already undergone pre-processing. The dataset itself, due to its small size, allows checking effects of algorithms without getting rid of redundant and insignificant features.

The main motive is to answer the question of how data deficiency strongly influences the outcome and whether there is a difference between the use of selected supervised learning algorithms requires a comparison of the difficulty level of creating a model, accuracy, complexity and time to obtain an answer.

Wykrywanie występowanie chorób serca,porównanie algorytów uczenia maszynowego nadzorowanego na podstawie zbioru danych dotyczących chorób układu krążenia z repozytorium UCI

Wstęp

Sztuczna inteligencja wśród szerokiego zakresu swoich zastosowań może zostać wykorzytana do analizy bardziej lub mniej złożonych danych medycznych, w celu przewidzenia wystąpienia choroby u konkretnej osoby, bez udziału procesu myślowego od stony specjalisty.

Do tego przeznaczenia istnieje możliwość zastosowania uczenia nadzorowanego (ang. *supervised learning*) tj. rodzaj uczenia maszynowego zakładający istnienie zbioru danych testowych zawierających odpowiedzi, na których podstawie wyszukiwane są zależności, cechy znaczące oraz budowany jest w ten sposób model służący przykładowo do przewidywania przyszłych wartości.

W przypadku danych dotyczących chorób zależności typujące występowanie choroby, bazują na podstawie konkretnych wyników badań zgormadzonych w repozytorium UCI.

W dzisiejszych czasach choroby sercowo-naczyniowe stanowią najczęstszą przyczynę zgonów, a liczba osób cierpiących na te dolegliwości stale rośnie. Głównymi przyczynami zachorowalności diagnozowanymi przez specjalistów są niski poziom świadomości i profilaktyki chorób serca.Dlatego prowadzone są intensywne prace nad zwiększeniem dostępności badań, które wspomogą diagnostykę kardiologiczną na jak najwcześniejszym etapie [1].

Powodem szukania dokładniejszych sposobów diagnozowania są również wysokie koszty leczenia generowane przez choroby układu krwionośnego. Według analityków firmy konsultingowej KPMG [2] w 2011 r. koszty diagnostyki i terapii chorób serca wyniosły ponad 15 miliardów polskich złotych.

Uczenie maszynowe poprzez przetwarzanie dużych zasobów klinicznych danych historycznych pod kątem zależności przyczynowo skutkowych, może zostać wykorzystane do wczesnej diagnostyki lub wspomagania leczenia pacjentów [3].

Słowa kluczowe: uczenie maszynowe, uczenie nadzorowane, lasy losowe, maszyna wektorów nośnych, k-najbliższych sąsiadów

Cel i zakres pracy

Celem pracy jest porównanie wybranych algorytmów uczenia maszynowego nadzorowanego, przy założeniu że dane wejściowe są wybrakowane, a w rezultacie zbudowanie modelu który na podstawie danych medycznych wystawia diagnozę o występowaniu zaburzeń sercowo-naczyniowych lub ich braku.

Dane medyczne wyróżniają się tym, że trudno uzyskać do nich dostęp, najczęściej nie są to informacje, które się udostępnia do użytku publicznego, z tego powodu, kluczowym krokiem jest wybór cech branych pod uwagę przy tworzeniu modelu. Dane pozyskane z repozytorium UCI przeszły już wstepną obróbkę , sam dataset ze względu na swoje niewielkie rozmiary pozwala na sprawdzenie działań algorytmów bez pozbywania się nadmiarowych i mało znaczących cech.

Zatem odpowiedź na pytanie jak wybrakowanie danych mocno wpływa na rezultat i czy istnieją róznicę między zastosowaniem wybranych algorytmów nauczania nadzorowanego wymaga przedstawienia porówniania łatwości tworzenia modelu, dokładności, złożoności oraz czasu uzyskania odpowiedzi.

W pracy opisano następujące algorytmu uczenia nadzorowanego:

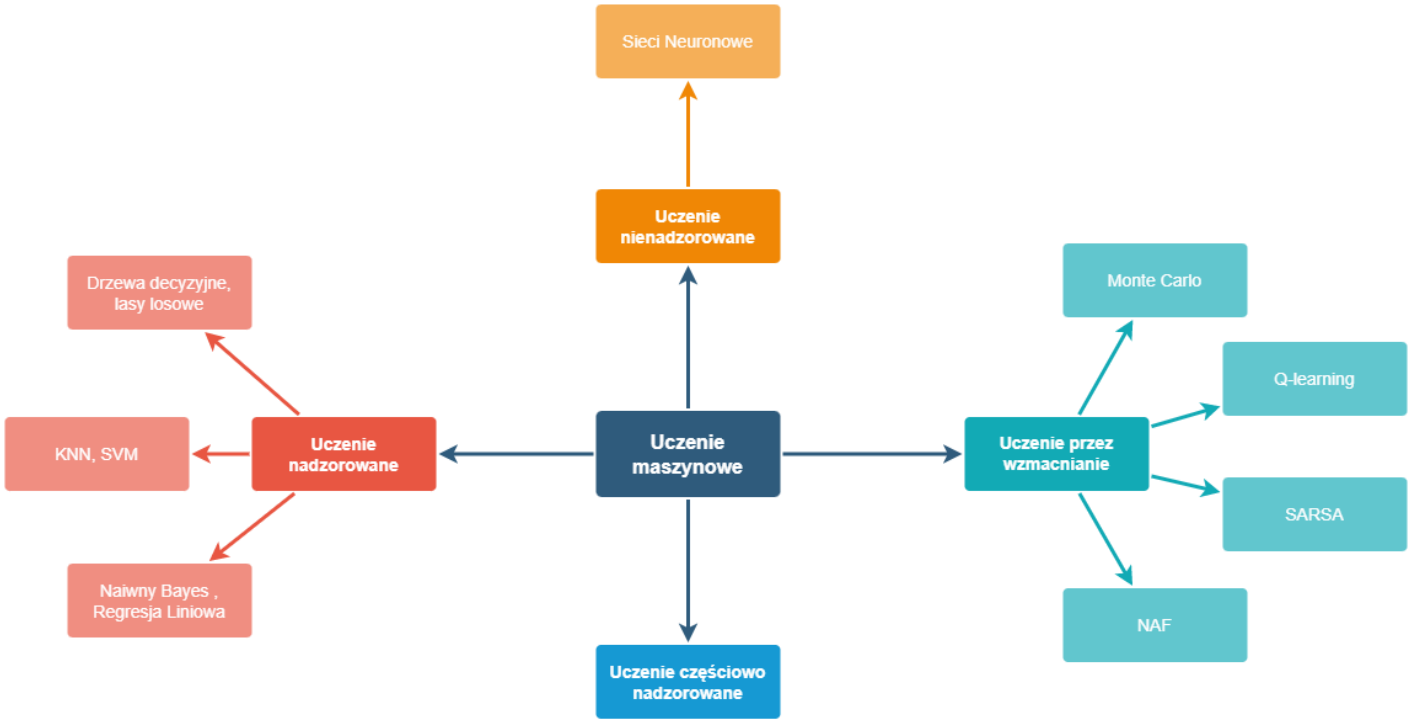
- losowe lasy decyzyjne (ang. *random decision forests*)
- maszyna wektorów nośnych (ang. *support vector machines*, SVM)
- k-najbliższych sąsiadów (ang. *k-neares neighbours*, KNN)

Wprowadzenie teorertyczne

Uczenie maszynowe (ang. *machine learning*, ML) to dziedzina zajmująca się tworzeniem modeli do analizy bardzo obszernych zasobów danch. Modele utworzone za pomocą algorytmów uczenia maszynowego są w stanie z wysokim prawdopodobieństwem wystawić predyckję lub dokonać klasyfikacji na temat zadanego problemu.

Model *klasyfikacjny* służy do przewidzenia etykiety klasy poprzez mapowanie na już z góry ustalony jednowymiarowy pozdział, model *regresywny* natomiast mapuje przestrzeń ustalając liczbę klas podziału oraz grupując wartości. [4] Istnieje możliwość przekształcenia problemu regresywnego na klasyfikacyje i na odwrót poprzez zamiane wartości oczekiwanego wyniku. Taką modyfikację zastosowano w praktycznej częsci projektu. Wyniki dla danych występowały w wartościach od 0 do 4 , dla wartości <1,4> przypadek testowy uznawany był za sklasyfikowany pozytywny (chory), dlatego przekształcenie z modelu regresywnego do modelu klasyfikacyjnego polega na konwersji wyników do wartości liczbowych 0 - brak stwierdzenia stanu choroboweo oraz 1 - stwierdzenie o chorobie układu krążenia.

Sposób wykorzystania segreguje alorytmy uczenia maszynowego na dwie kategorie, jednak powszechnie stosowanym podziałem jest podział zależnie od sposobu *trenowania* algorytmu. Algorytmy dzieli się na min.: uczenie nadzorowane, uczenie częściowo nadzorowane, uczenie bez nadzoru oraz uczenie przez wzmacnianie [5] .



Dobór typu uczenia oraz algorytmu uzależniony jest od danych wejściowych oraz oczekiwanego rezultatu. Dane wyjściowe mogą przyjmować format odpowiedzi TAK/NIE , klasyfikacji do danego zbioru czy np procentowej oceny ryzyka.

Uczenie maszynowe nadzorowane (ang. *supervised learning*) to klasa algorytmów uczenia maszynowego, która bazuje na poetykietowanych danych. Nadzór polega na porównaniu rezultatów działania modelu z wynikami które są zawarte w danych wejściowych (*dane oznaczone*) [6]. Algorytm po osiągnięciu żądanej efektywnosci jest w stanie dokonać klasyfikacji przykładu dla którego nie posiada odpowiedzi. Sprawdza się to obecnie w rekomendacji produktów oraz diagnozie chorób.

Z matematycznego puntu widzenia dopasowanie danych oznaczonych nazywane jest aproksymacją funkcji [5] .

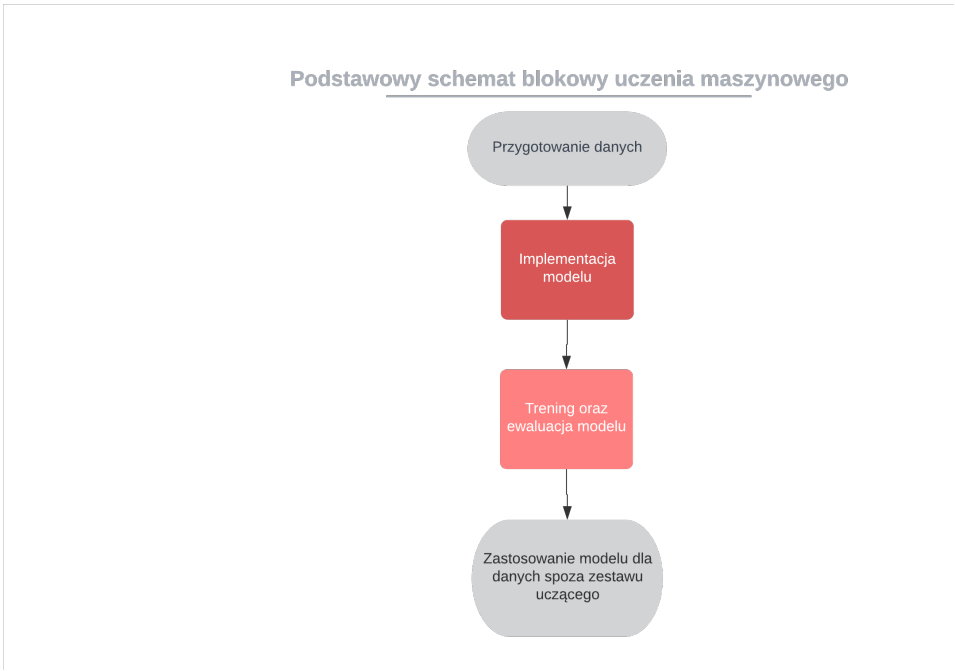
Uczenie maszynowe bez nadzoru (ang. *unsupervised learning*) to klasa algroytmów uczenia maszynowego która wiodąco rozwiązuje problemy grupowania. Dane dostarczane do modelu nie zawierają *oznaczeń*, zatem nauczanie polega na wyciąganiu konkluzji z poprzednio wykonanych iteracji. Na skuteczność modeli budownych w oparciu o uczenie bez nadzoru wpływ ma rozmiar dostarczonego do nauki zbioru danch, im jest on większy tym bardziej wzrasta effektywność. Takie zbiory można uzyskać rejestrując dane na bierząco dlatego do najczęstrzych zastosowań tej klasy algorytmów, można zaliczyć rozpoznawanie mowy czy obrazu [5] .

Uczenie maszynowe przez wzmacnianie (ang. *reinforcement learning*) to klasa algorytmów uczenia maszynowego której nauczanie nie opiera się na danych wejściowych czy wyjściowych a rezultatach otrzymanych podczas testu nazywanych tzw. sygnałami wzmocnienia który może przyjmować wartość pozytwną lub negatywną. Algorytm generując dane wejściowe dostosowuje reguły by uzyskać zwrotnie sygnał pozytywny w jak największej liczbie przypadków. [7] .

Uczenie częściowo nadzorowane (ang. *semi-supervised learning*) to klasa algorytmów uczenia maszynowego która wykorzystuje zbór danych w większości niepoetykietowany na podstawie których tworzony jest model [8] .

Podział osób na kategorie cierpiące na choroby sercowo-naczyniwe oraz zdrowe, to dylemat klasyfikayjny nadający się do rozwiązania za pomoca algorytmów uczenia maszynowego nadzorowanego i na nich skupia się dalsza część pracy.

Ścieżka działania algorytmów uczenia maszynowego nadzorowanego



Model Danych

[9]

Repozytorium uczenia maszynowego UCI

Sensem wykorzystania uczenia maszynowego jest prognoza lub klasyfikacja rzeczywistych wartości z dużego zbioru danych które mogą znaleść zastosowanie w praktycznych dziedzinach. Im bardziej dokładne i rzeczywiste dane do testowania i tworzenia modelu tym większe prawdopodobieństwo otrzymania realnych wyników na końcu ścieżki uczenia. W celu gromadzenia miarodajnej bazy dostępnych zbiorów danych testowych powstało repozytorium uczenia maszynowego UCI. Jak podaje strona informacyjna :

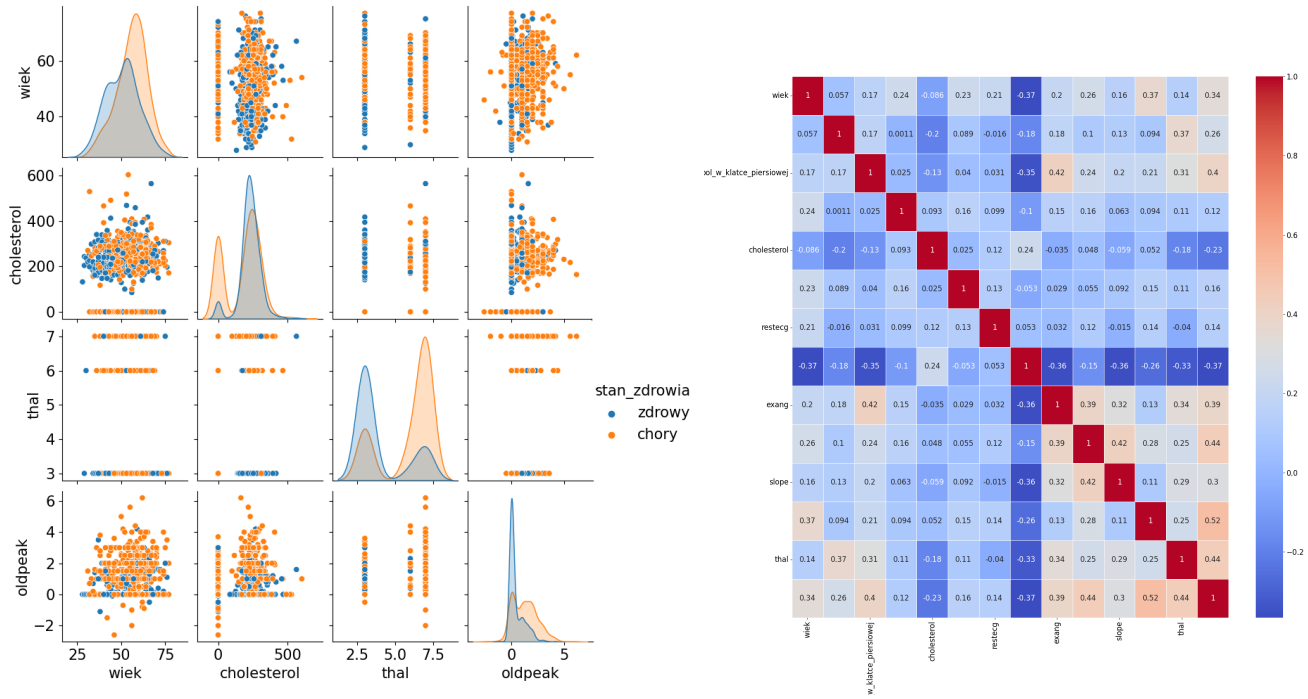
> … było ono cytowane ponad 1000 razy, co czyni je jednym ze 100 najczęściej cytowanych „artykułów” w całej informatyce … [9]

Repozytorium gromadzi dane z wielu rozbieżnych dziedzin , dane medyczne umieszczone w repozytorium nie zawierają wrażliwych danych pacjentów , a niektóre zbiory są poddane już wstępnej obróbce tak jak zbiór danych “Heart Disease Databases” wykorzystany w tym dokumencie, który powstał na podstawie realnych danych medycznych zebrany z lokalizacji

1. Fundacja Cleveland Clinic [10]
2. Węgierski Instytut Kardiologii, Budapeszt [11]
3. V.A. Centrum medyczne, Long Beach, Kalifornia [10]
4. Szpital Uniwersytecki, Zurych, Szwajcaria [12].

Stratyfikacja

Wyróżniono 14 atrybutów spośród 76 zebrancyh do wykorzystania w algorytmach uczenia maszynowego, wszystkie z nich mają wartośi liczbowe.



Rozkład chorób serca w danych testowych to 44.67% chorych czyli 509 prób pozytywnych oraz 411 negatywnych. W danych testowych znajduje się 726 przypadków osób płci męskiej oraz 194 żeńskiej. Dla zachorowań widać nierówność ale jest ona spowodowana rzeczywistą statystyką. Tylko u 25.77% badanych kobiet stwierdzono występowanie chorób wieńcowych, natomiast wśród badanych mężczyzn jest to aż 63.22%. [9]

Lista atrybutów wykorzystanych w algorytmie:

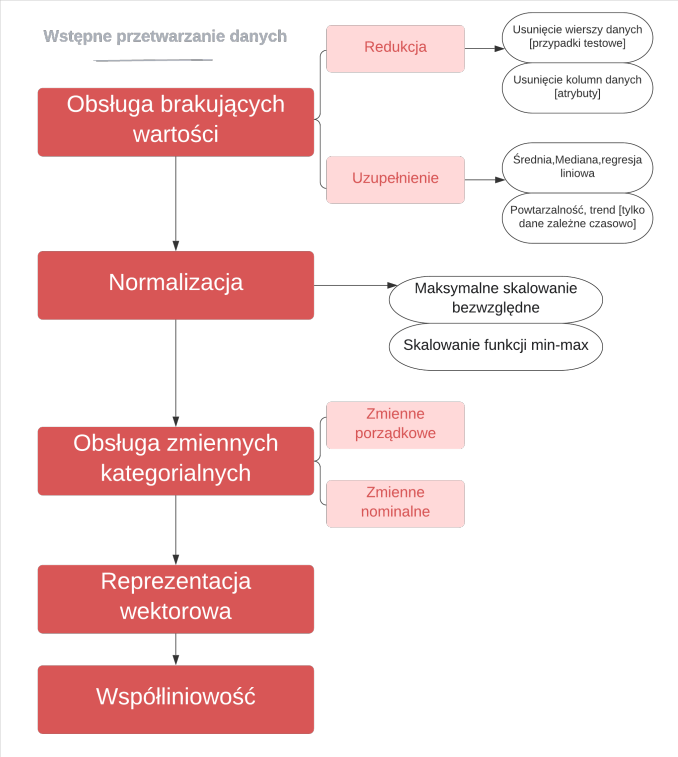
- wiek
- płeć
- rodzaj bólu w klatce piersiowej
- spoczynkowe ciśnienie krwi

- cholesterol w surowicy w mg/dl
- poziom cukru we krwi na czczo > 120 mg/dl
- spoczynkowe wyniki elektrokardiograficzne
- osiągnięto maksymalne tętno
- dławica piersiowa wywołana wysiłkiem fizycznym
- obniżenie odcinka ST wywołane wysiłkiem fizycznym w stosunku do odpoczynku
- nachylenie szczytowego odcinka ST ćwiczenia
- liczba głównych naczyń pokolorowanych fluorozopią
- skan serca z talem lub test wysiłkowy
- stan (brak choroby serca/choroba serca)

W przypadku danych testowych z repozytorium UCI, fakt iż dane pochodziły z różnych lokalizacji ma duże znaczenie ,gdyż od placówki medycznej zależy jakim badaniom poddani zostali pacjęci a co za tym idzie w jakich kolumnach tabelarycznego przedstawienia będą mieć uzupełnione bądź puste wartości. Scalenie ze sobą wyników badań dostarcza większej różnorodności również dzięki temu że dane pochodzą z wielu krajów. Jeżeli zestaw wejściowy zostałby ograniczony do jednej lokalizacji to cecha dla której nie uzupełniono wartości zostałaby pominięta podczas treningu ze względu na brak danych, co skutowało by uboższym modelem i możliwe że pominięciem kluczowej cechy wpływającej na działanie.

Wstępna obróbka danych

Proces przetwarzania danych może składać sie z wielu różnych kroków zależenie od typu, w uczeniu nadzorowanym operującym na danych tekstowo-liczbowych poprawnym będzie zastosowanie schematu przedstawionego poniżej:



Po złączeniu można przeprowadzić szereg działań w celu sztucznego uzupełnienia pustych wartości bazując na wartościach które już istnieją.

Obsługa brakujących wartości

Możliwościami obsługi brakujących wartości są : mniej polecana ze względu na utrate danych, redukcja zestawu danych lub uzupełnienie go zgodnie z wybrany przez siebie założeniem. Biblioteki do nauczania maszynowego dostarczają już gotowe rozwiązania do upuszczenie wierszy lub kolumn zawierających wartości *null*. Uzupełnienie danych inaczej *imputacja*, rozwiązuje problem w mniej stratny sposób i tak samo jak do redukcji są już gotowe rozwiązania w bibliotece sklearn. Istnieją 4 różne strategie uzupełniania wykorzystujące proste matematyczne obliczenia takie jak :

- średnia,
- mediana,
- stała,
- najczęściej występująca wartość.

Do wyznaczenia wartości uzupełniających można również użyć regresji liniowej.

Standaryzacja

Przekształcenie danych również bazujące na statystycznych założeniach i również ustandaryzowane w popularnych bibliotekach. Dążymy aby średnia wartoś wynosiła 0, a odchylenie standardowe 1 dla liczbowyh reprezentacji danych. Z matematyczne punktu widzenia wykonujemy działanie

$$\frac{\bar{X}}{\sqrt{\frac{\sum_{i=1}^{n}(X-\bar{X})^2}{N-1}}}$$

[13]

Obsługa zmiennych kategorialnych

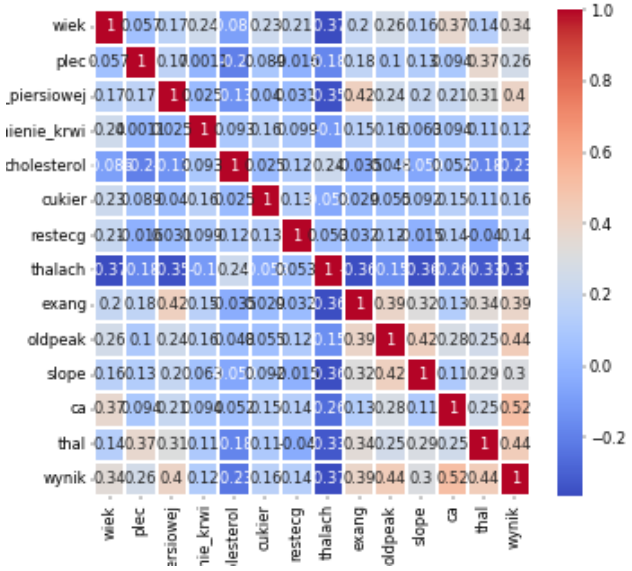
Cechy kategorialne dzielą się na dwie zasadnicze grupy ze względu na mozliwość uprorządkowania , dane takie jak wykształcenie , rozmiar podlegają mapowaniu , dane typu kolor lub płeć podlegaja kodowaniu. W ten sposób dane kategoryczne staja się wartosciami liczbowymi.

Reporezentacja wektorowa

Obsługa danych kategorialnych pozwoliła zmapować/zakodować je w postaci liczbowej, ale można pójść o krok dalej i te same dane mieć w postaci 0 lub 1 na odpowiedniej kolumnie. Rozwiązanie reprezentacji wektorowej polega na utworzeniu tylu kolumn ile jest unikalnych wartości dla kategorii i wpisanie 0 lub 1 dla każdego rekordu danych [14] .

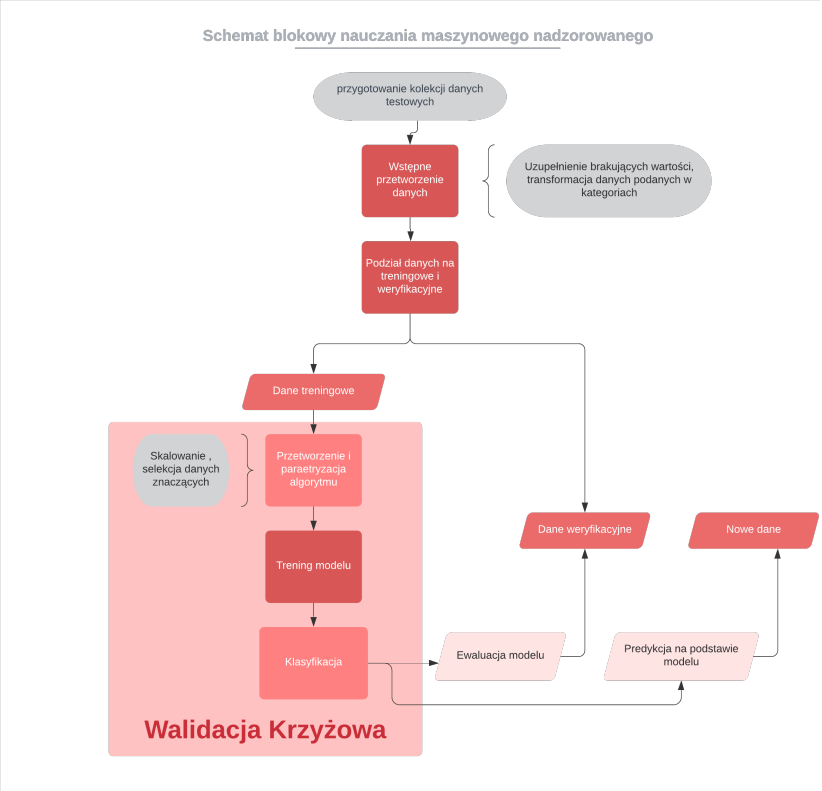
Współliniowość cech

Aby znaleść korelacje współliniowości należy szukać liniowej zależności pomiędzy danymi, najłatwiej zauważyć to tworząc wykresy z danych testowych dla każdej pary [14].



Zgodnie z poniższym schematem po przeprocesowaniu wejściowego zbioru danych, należy go podzielić na dane treingowe oraz ewaluacyjne. Powszechnie stosowana K krzyżowa walidacja umożliwia maksymalne wykorzystanie dostarczonego

wejścia do dostrajania parametrów modelu, ponieważ optymalizacja hiperparametrów połączone z ciągłą weryfikacją poprawności to sedno treningu.



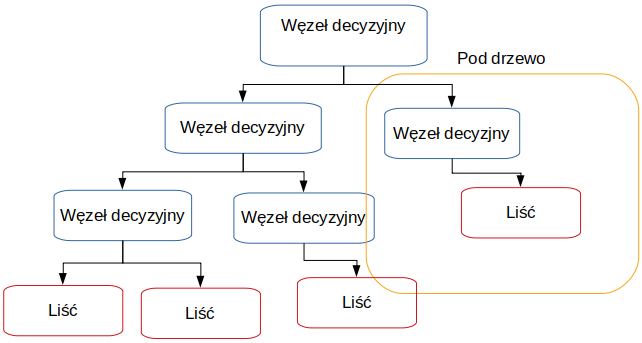
K-krotna walidacja krzyżowa (ang. *K-fold Cross Validation*, KCV) - metoda weryfikacji działająca poprzez podział zbioru danych na k podzbiorów z których każdy przynajmniej raz jest zbiorem oceniającym wydajność , zaznaczając że K musi być równe lub mniejsze niż liczba elementów w zbiorze [15], [16] .

Kluczowym elementem jest ewaluacja która odbywa się na końcu każdej z k-1 iteracji w celu dostosowania parametrów, po osiągnięciu wymaganych lub ustalonych wartości dokładności modelu lub weryfikacji wszystkich możliwych opcji i znalezienie najlepszego modelu można go wykorzystać do weryfikacji na danych spoza zestawu testowego.

Wybrane algorytmy uczenia maszynowego nadzorowanego

Losowe lasy decyzyjne

Drzewa decyzyjne (ang. *decisions trees*) są uznawane za najprostyszy i najbliższy ludzkiemu zrozumieniu algorytm uczenia, który swoją nazwę zawdzięcza graficznej reprezentacji w postaci drzewa. Każdy węzeł oznacza atrybut, na podstawie którego następuje rozróżnienie. W modelu kluczowa jest kolejnosc cech, które występują po sobie ponieważ determinuje to otrzymany rezultat [5], [17] .



Prawie każdy algorytm uczenia maszynowego nadzorowanego można podzielić na dwa etapy. W pierwszym opracowywany jest wzorzec, na którym bazuję późniejsza predykcja. Etap nauki dla drzewa decyzyjnego polega na typowaniu atrybutów, które staja się węzłami decyzyjnymi, dzielącymi rekordy na dwa mniejsze zestawy i tak aż nie ma możliwości dalszego podziału.

Na metodologie drzew decyzyjnych oparta jest dokładniejsza forma nauczania nadzorowanego: *losowe lasy decyzyjne.*

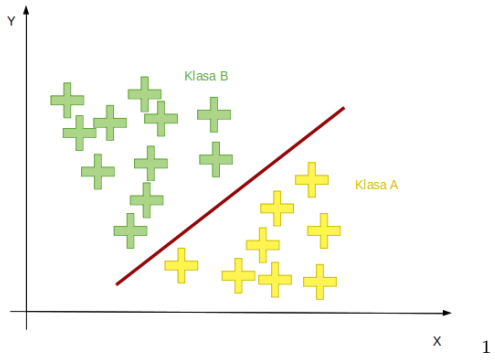
Losowe lasy decyzyjne (ang. *random decision forests*) to technika polegająca na połączeniu wielu drzew decyzyjnych w celu uniknięcia problemu z *nadmiernym dopasowaniem* do treningowego zestawu danych na którym został przeszkolony.

Utworzony szablon aby poprawnie działać na danych testowych i służacych weryfikacji, nie może stać się charakterystycznym przypadkiem rozwiązującym przypadek testowy [5], [17]. W tym celu dla loswoych lasów decyzyjnych najpierw stosuję się **agregację bootstrap'ową**. Z treningowego zestawu danch losuję się, z możliwymi powtorzeniami, wiersze danych dla których trenowany będzie model. Jako rezutat brana jest większość lub średnia wartości uzyskanych wyników dla poszczególnych drzew dezycyjnch. Dodatkowo dla drzew decyzyjnych w lasach losowych, atrybuty odpowiadające za kategoryzację są wybierane z wylosowanego podzbioru.[18]

Wśrod zalet lasów losowych nalezy wyróżnić iż potrafią one trafnie wykalkulować brakujace wartości cech. Idealnie znajdują zastosowanie dla realnych danych, których zasadniczym problemem jest ich niekompletność.
Dane medyczne posiadają szeroką wariację zmiennych z dużym prawdopodobieństwem wybrakownia, zastosowanie do nich lasów decyzyjnych ma potencjał na pozytywne rezulataty.

Maszyna wektorów nośnych

Metoda wektorów nośnych (ang. *support vector machines* , skr. ***SVM***) to algorytm uczenia maszynowego nadzorowanego, który każdy parametr z dostępnych cech dla danych wejściowych, traktuje jako punkt w przestrzeni. Na podstawie ułożenia punktów dzieli się je na 2 klasy. Graficznie jest to reprezentowane przez prostą dla której odległość między najbliższymi dwoma punktami dla wektorów jest możliwie największa.

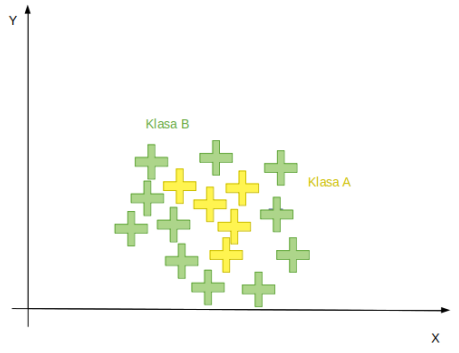


[1]

Taka prosta nazywana jest *prostą marginalną* i powstaje ona poprzez generowanie i selekcję tych prostych które rzetelnie szufladkują klasy danych [5], [17].

[1]Na podstwie materiałów opublikowanych na https://www.datacamp.com

Techinka ta gwarantuje precyzyjniejsze rezulatay niż drzewa decyzyjne, niestety dla dużych zbiorów danych czas trwania szkolenia znacznie się wydłuża oraz istnieją przypadki dla których podział jedną prostą jest niewykonalny, taki przypadek reprezentuje rozkład na schemacie nr. 2.



[2]

Z powyższego schematu widać że prosta marginalna ma zastosowanie w przypadku dwóch wymiarów, dla większej ilości stosowane jest przekształcenie do innego systemu współrzędnych i szukanie hiperpłaszczyzny brzegowej dzielącej tak samo jak prosta punkty w przestrzeni na dwa zbiory.[19]

Wyszukiwanie podziału

Idea działania maszyny wektorów nośnych opiera się na wyznaczenia minimalnej wartości wektora wag oraz przesunięcia (ang. *bias*) który geometrycznie opisuje współrzędne hiperpłaszczyzny.

$$\min_{w,b} \frac{1}{2}||w||^2$$

$$\text{pod warunkiem } y_i(w \cdot x_i + b) - 1 \geqslant 0,\ i = 1, \ldots, n.$$

[20]

K najbliższych sąsiadów

K najbliższych sąsiadów (ang. *k nearest neighbours*, skr. ***KNN***) to algorytm uczenia maszynowego nadzorowanego operający swoje estymacje dla konkretnego przypadku danch na wartościach jego K najbliższych sąsiadów(punktów) liczonych min. dla przestrzeni Euklidesowej [5]. Do wyznaczenia odległości w metryce Euklidesowej stosowany jest wzór:

$$d_{(x,y)} = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad [21]$$

popularne są również przestrzenie Manhattan:

$$d_{(x,y)} = \sum_{i=1}^{n} |(x_i - y_i)| \quad [21]$$

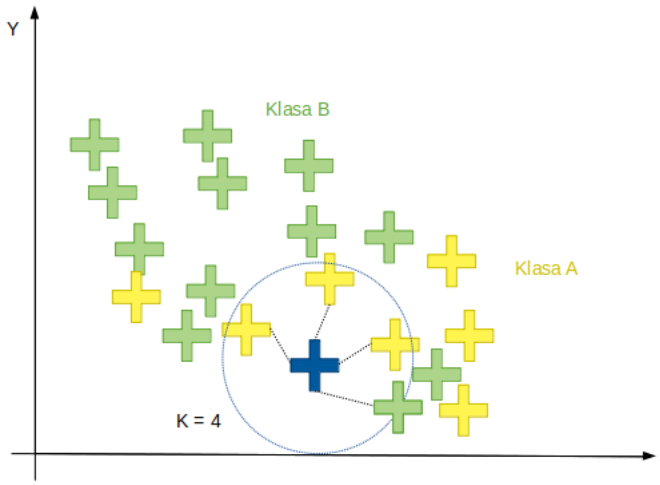
oraz Mińkowskiego:

$$d = \left(\sum_{i=1}^{m} |u_i - v_i|^p\right)^{1/p} \quad [22]$$

Atrybut który nastraja proces uczenia się modelu i ma na niego największy wpływ określany jest jako hiperparametr. Dla KNN jest to liczba sąsiadów i może przyjmować maksymalnie wartości do rozmiaru zbioru cech. Im większa ilość jednostek mających wpływ, tym potęguje się niestety złożoność czasowa algorytmu, znacząco już większa od przedstawionych powyżej innych algoryrtmów,[5] oraz tym bardziej wzrasta ryzyko nadmiernego dopasowania do modelu testowanego.

[2] Na podstwie materiałów opublikowanych na https://www.datacamp.com

W celu przewidzenia wartości dla nowych danych, należy odnaleść K najbliższych punktów wyliczając odległości, a następnie przpisać odpowiedź implikowaną przez większość sąsiadów. Dla wartości K równej jeden, metoda ta nazywana jest algorytmem najbliższego sąsiada.



[3]

Dla lekarza wartością dodatnią jest wykrycie zależności które decyzują o uznaniu lub zaprzeczeniu występowania choroby. Zastosowanie algorytmu KNN może nie tylko zakwalifikować osoby chorujące na serce, ale również ułatwić swoją graficzną reprezentacją wpływ cech na ostateczny osąd próbki.

[3] Na podstwie materiałów opublikowanych na https://www.datacamp.com

Opis praktycznej częsci projektu

Narzędzia i biblioteki zastosowane w pojekcie

Praktyczna część pracy napisana została w języku Python z wykorzystaniem *scikit-learn*, obsługującym wiele algorytmów maszynowego uczenia się w tym uczenia nadzorowanego i docelowo wybranych algorytmów przedstawionych w teoretycznej części pracy.

Biblioteka opiera się o *Numpy* oraz *Scipy*, daje zestaw narzędzi do obliczeń na macierzach, wektorach oraz umożliwiający metody numeryczne takie jak całkowanie, różniczkowanie i temu podobne [23]. W rezultacie można za jej pomoca wykonać elementy procesu nauczania algorytmu, takie jak: przetwarzanie wstępne, redukcja wymiarowości, klasyfikacja, regresja. [24]

Do przygotowania danych wykorzystano zestaw narzędzi *Pandas*, ułatwiający tworzenie struktur danych i ich analizę.

W celu wizualizacji wyników w postaci wykresów zastosowano, opartą na *Matplotlib*, bibliotekę *Seaborn* powszechnie stosowaną do rysowania estetycznej grfiki statystycznej.

Część prezentacyjna czyli możliwość wprowadzenia danych w formularzu na stronie i weryfikacja wyniku dla wyuczonych już modeli wykorzystuje bibliotkę *Flask*. Framework Flask ułatwia pisanie aplikacji internetowch i jest rozwiązaniem które daje duży zakres dowolności oraz możliwości. Flask sam z siebie nie definiuje warstwy bazy danych czy formularzy, pozwala za to na obsługę rozszerzeń które ubogacają aplikację o wybraną funkcjonalność. [25]

Przekazywanie obiektów o bardziej skomplikowanej budowie i ich *serializacja* oraz *deserializacja* do formatu JSON wykonane są za pomocą biblioteki *jsonpickle*, a zapis modeli wykonano za pomocą *joblib* która zapewnia obsługę obiektów Pythona i jest zoptymalizowana pod kątem pracy na dużych tablicach Numpy. [24]

Biblioteki w większości posiadają otwarty kod źródłowy, napisany w języku Python [24].

Moduły projektu:

- Config - zawiera statyczne zasoby oraz konfigurację logowania projektu
- Data - moduł odpowiada za wczytywanie i obróbkę danych testowych, oraz zawiera definicje objektów wykorzystywanych przy uczeniu oraz zapisu modelu
- Management:
  - PlotGeneration - moduł odpowiedzialny za prezentację wyników w postaci wykresów porównujących algorytmy oraz odpowiedzi na zadany problem
  - Prediction :
    - RF - implepmentacja treningu algorytmu Lasów losowych
    - KNN - implepmentacja treningu algorytmu K-najbliższych sąsiadów

        * SVM - implepmentacja treningu algorytmu Maszyny wektorów nośnych

- Static - forlder z grafikami, plikami stylów, skryptami javascript i jQuerry
- Templates - folder z stronami html wykorzystującymi dyrektywy Flask

Projekt posiada dwa tryby pracy :

- tryb nauczania na podstawie danych testowych
  machine learning z wykorzystaniem 3 algorytmów (*Run_Learning_Proces.xml*)
- tryb aplikacji web
  wykorzystanie Flask do prezentacji i wykorzystania utworzonych modeli (*Run_Web_Application.xml*)

Trening algorytmu

Głównym zadaniem trybu nauczania jest utworzenie i wytrenowanie modeli dla 3 algorytmów nauczania nienazdorowanego, w tym celu wykonywany jest preprocesing danych czyli kolejno:

2. Załadowanie i konkatenacja dataset'u
3. Uzupełnienie pustych wartości - dla późniejszego porównania tworzone są imputery dla 4 różnych form uzupełnienia
4. Standaryzacja
5. Konwersja danych dla kategorii
6. Normalizacja z wykorzystaniem MinMaxScaler.

Następnie wykonywany jest podział na dane treningowe i testowe z wykorzystaniem zdefiniowanej w bibliotece sklearn predefiniowanej metody. Tak spreparowany zestaw danych poddawany jest treningowi modelu kolejno dla każdego z algorytmów. Do dostrojenia parametrów oraz znalezienia najlepszego modelu wykorzystuwany jest:

GridSearchCV

W projekcie dla każdego algorytmy zapróbkowano większość dostępnych dla danego modelu klasyfikacji hiperparametrów przekazywane w param_grid.

Wykorzystane parametry wykonania GridSearchCv [23]:

- estimator: implementacja interfejsu obiekt estymatora scikit-learn,
- param_grid: słownik parametrów które są potem testowane w dowolnej sekwencji ustawień,
- refit: dopasowanie best_estimator_ ,best_index_, best_score_ i best_params_ dla najlepszej sekwencji ustawień parametrów
- cv: parametr k dla KFold walidacji krzyżowej,
- verbose: obszerność logowanych informacji

Estymatory to impementacja z skleran która powstała w oparciu o dokumntacje sklearn oraz dostępną dla nich parametryzacje.

Zestawienie najlepszych osiągniętych estymatorów przedstawia się następująco:

KNeighborsClassifier [23] :

- n_neighbors: 8 - liczba sąsiadów z których wnioskowany jest jednostkowy resultat
- weights: distance - wagi na podsatwie których wyliczana jest predykcja , można zastosować wagę 1:1 lub nałożyć wagi zgodnie z dystansem.
- algorithm: auto - algorytm zastosowany do znalezienia najbliższych sąsiadów, w projekcie wykorzystano : brute-force oraz auto
- leaf_size: 1 - rozmiar liścia dla algorytmów BallTree or KDTree
- p: 1 - wykorzystanie miar odległości dla manhattan
- metric: canberra -metryka odległości

RandomForestClassifier [23] :

- criterion: [todo] - funkcja pomiaru dokładności rozgałęzienia
- min_samples_leaf: [todo] -minimalna liczba probek wymagana na liściu.
- min_weight_fraction_leaf: [todo] -minimalny ułamek sumy wag wymagany na liściu

- min_impurity_decrease: [todo] - większe lub równe zmniejszenie zanieczyszczenia powoduje podział danego węzła Zmniejszenie zanieczyszczenia liczone jest zgodnie z wzorem:

```
N_t / N * (impurity - N_t_R / N_t * right_impurity
                    - N_t_L / N_t * left_impurity)
```

gdzie N to całkowita liczba próbek, N_t to liczba próbek w bieżącym węźle, N_t_L to liczba próbek w lewym liściu, a N_t_R to liczba próbek w prawym liściu.

- max_features: [todo] - liczba funkcji najlepszego podziału
- random_state: [todo] - wykorzystywany przy próbkowaniu cech przy poszukiwaniu najlepszego podziału w węźle
- cpp_aplha: [todo] - zastosowanie to przycinanie drzewa o największej złożoności mniejszej niż cpp_alpha

SVC [23] :

- C: [todo]float, default=1.0 Regularization parameter. The strength of the regularization is inversely proportional to C. Must be strictly positive. The penalty is a squared l2 penalty.
- kernel: [todo]- jądro wykorzystane w algorytmie
- degree: [todo] - stopień dla funkcji jądra *poly*
- gamma: [todo] - współczynnik jądra dla wartości *scale* parametr jądra ustawiany jest na wartość:

```
1 / (n * X.var())
```

dla wartości auto jest to :

```
1 / n
```

gdzie n to liczba cech.

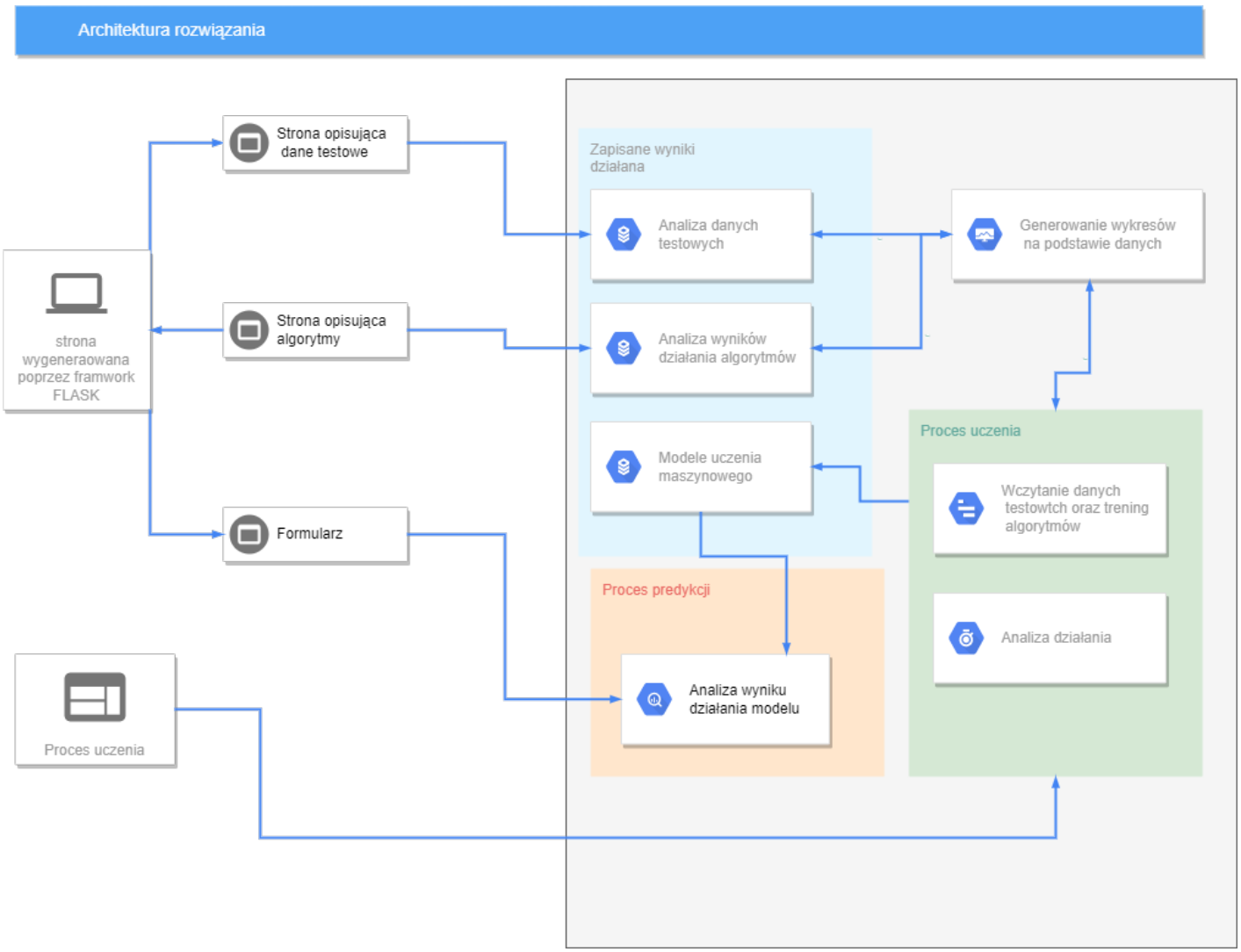
- coef0: [todo] - niezależny parametr funkcji jądra , wykorzystywany tylko przy jądrach *poly* i *sigmoid*.
- shrinking: [todo] - heurystyka kurcząca
- cache_size: [todo] - cache jądra (w MB).

Po odnalezieniu najlepszego estymatora model jest zapisywany oraz generowane sa wykresy dla trybu aplikacji webowej:

- wykresy modeli datasetu wejściowego i rozłożenia cech
- wykresy prezentujące zestawienia danych zebranych na temat algorytmu podczas wykonywania treningu.

Opis działania aplikacji webowej

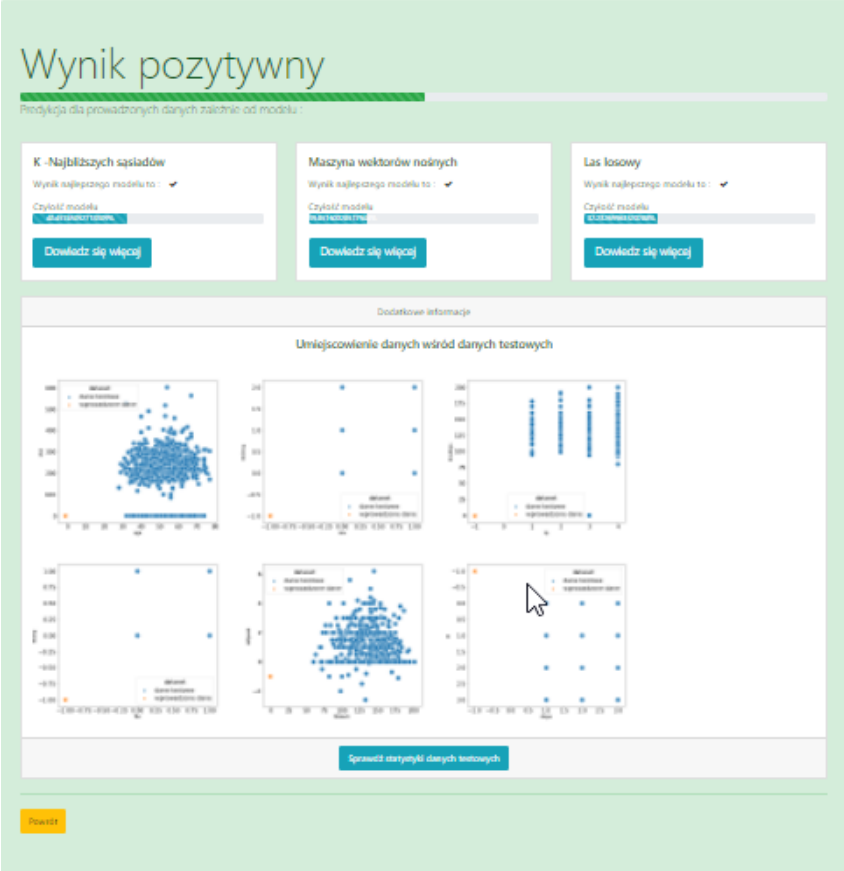
Poniżej przedstawiono architektówe działania:



Aplikacja posiada 4 widoki :

- widok głowny strony
- widok prezentacji danych wejściowych
- widok omówienia treningu algorytmów
- widok formularza pozwalającego na wykonanie predycji na wyuczonych modelach na podstawie własnych danych wejściowych

Zatwierdzenie formularza wyzwala odczytanie zapisanych modeli , iteracje i wykonanie predykcji na każdym z nich , następnie prezentowane są wyniki dla najlepszych estymatorów oraz wykresy wskazujące na umiejscowienie nowych danych na tle zbior testowego.



Porównanie działania modeli

W tym podrozdziale zamieszczone zostały wyniki oraz wykresy wygenerowane podczas treningu i weryfikacji danych testowych.

Algorytm	Dokładność w %
Losowe lasy decyzyjne	98.2
Maszyna wektorów nośnych	98.2
K-najbliższych sąsiadów	97.8

Zestawienie efektywności działania algorytmów

Konfrontacja technik ucznia maszynowego zaleznie od zestawu danch będzie dawała odmienne wyniki ze względu na ich predyspozycje do zajmowania się odpowiednimi zbiorami danych.

Potencjał algorytmów dla niewielkiego kompletu danych zawierającego wartości

Zczynając od drzew decyzyjnych,można od razu stwierdzić ich niski potencjał. Istnieje zbyt duże prawdopodobieństwo dopasowania się do modelu treningowego, gdyż wspomniany zbiór dancyh wejściowych nie jest wystarczająco liczny. Dlatego w pracy omówione zostały lasy decyzyjne.

Większej dokładności można się spodziewać po metodzie wektorów nośnych, ale jego złożoność czasowa oraz pamięciowa mogą zaniżyc jego ogólną klasyfikację.

K-najbliższego sąsiada może być przydatny w przypadku danych nieliniowych oraz łatwo wykorzystany w problemach regresji. Wartość wyjściowa obiektu jest obliczana przez średnią k wartości najbliższych sąsiadów. Niestety tak samo jak w przypadku maszyny wektorów nośnych jest wolniejsza i bardziej kosztowna pod względem czasu i pamięci. Wymaga

dużej pamięci do przechowywania całego zestawu danych treningowych do przewidywania oraz nie nadaje się również do dużych danych wymiarowych.

Wskaźniki wydajności

Określenie stopnia, w jakim skonstruowany model z powodzeniem realizuje wyznaczone zadanie należy do wskaźnika wydajności. Przykładem nieprawidłowego wyboru może być próba przewidzenia wystąpienia rzadkiej choroby u pacjenta i określenie głownym miernikiem *dokładność*. W takim scenariuszu klasyfikacja wszystkich pacjentów jako zdrowych , daje niewiele odbiegającą od perfekcji dokładność, a jednocześnie błędnie osądzać każde wystąpienie choroby.

Losowe lasy decyzyjne

##todo liczenie błędów macież pomysłek

Losowe lasy decyzyjne

###OCENA PODELI ORAZ UŻYTYCH PARAMETRÓW -OCENA SZYBKOŚCI WYKONANIA -OCENA ZALEŻNIE OD UZUPELNIANIA DANYCH -OCENA ZALEŻNIE OD DOBRANEJ PARAMERYZACJI : - które parametry mają i wpływ i dlaczego: - ZALEŻNIE OD METRYKI(SHORT OPIS METRYK)

Maszyna wektorów nośnych

###OCENA PODELI ORAZ UŻYTYCH PARAMETRÓW -OCENA SZYBKOŚCI WYKONANIA -OCENA ZALEŻNIE OD UZUPELNIANIA DANYCH -OCENA ZALEŻNIE OD DOBRANEJ PARAMERYZACJI : - które parametry mają i wpływ i dlaczego: - ZALEŻNIE OD METRYKI(SHORT OPIS METRYK)

K-najbliższych sąsiadów

###OCENA PODELI ORAZ UŻYTYCH PARAMETRÓW -OCENA SZYBKOŚCI WYKONANIA -OCENA ZALEŻNIE OD UZUPELNIANIA DANYCH -OCENA ZALEŻNIE OD DOBRANEJ PARAMERYZACJI : - które parametry mają i wpływ i dlaczego: - ZALEŻNIE OD METRYKI(SHORT OPIS METRYK)

Porównianie całościowe algorytmów : złożoność czasowa , dokładność , złożoność implementacyjna , wpływ danych wykorzytywanych w modelu

Schemat 20

Schemat 20

Wybrane najlepsze modele klasyfikacji:

[CV 5/15] END C=0.1, cache_size=200, coef0=0.0, degree=1, gamma=scale, kernel=linear, shrinking=False;, score=0.982 total time= 0.0s [CV 4/15] END C=0.1, cache_size=200, coef0=0.0, degree=1, gamma=scale, kernel=poly, shrinking=True;, score=0.982 total time= 0.0s [CV 4/15] END C=0.1, cache_size=200, coef0=0.0, degree=1, gamma=scale, kernel=poly, shrinking=False;, score=0.982 total time= 0.0s [CV 4/15] END C=10, cache_size=500, coef0=0.3, degree=5, gamma=scale, kernel=linear, shrinking=True;, score=0.982 total time= 0.0s [CV 4/15] END C=10, cache_size=200, coef0=0.3, degree=5, gamma=auto, kernel=rbf, shrinking=True;, score=0.982 total time= 0.0s

Wybrane najgorsze modele klasyfikacji :

[CV 10/15] END C=100, cache_size=200, coef0=0.3, degree=5, gamma=scale, kernel=poly, shrinking=True;, score=0.800 total time= 0.0s [CV 10/15] END C=100, cache_size=500, coef0=0.3, degree=5, gamma=scale, kernel=poly, shrinking=False;, score=0.800 total time= 0.0s [CV 10/15] END C=100, cache_size=200, coef0=0.3, degree=5, gamma=scale, kernel=poly, shrinking=False;, score=0.800 total time= 0.0s

Najlepsze modele i wartości dla regresji :

Najgorsze modele i wartości dla regresji :

Maszyna wektorów nośnych

Schemat 20

Schemat 20

Schemat 20

K-najbliższych sąsiadów

###OCENA PODELI ORAZ UŻYTYCH PARAMETRÓW

Schemat 20

Schemat 20

Schemat 20

problem multiklasyfikacji - problem regresji kategrycznej - zwykła regresja , mierzyć będe metoda prównania - tzrea było wprowadzić reguły do float na int -> inne metody do liczenia błędów na dzień dobry widzimy nie dokładność ze wględu na klasyfiakcję po przecinku regresja kategoryczna -> rzutowanie przedziału wartości na wartość graniczną

Spis ilustracji

Spis tabel

Bibliografia

[1] W. Modrzejewski and W. J. Musiał, "Stare i nowe i czynniki ryzyka sercowo-naczyniowego - jak zahamować epidemię miażdżycy? Część i. Klasyczne czynniki ryzyka," *Forum Zaburzeń Metabolicznych*, vol. 1(2), pp. 106–114, 2010.

[2] "Międzynarodowa sieć firm audytorsko-doradczych ze szczególnym uwzględnieniem branży dóbr konsumpcyjnych, usług finansowych, nieruchomości i budownictwa, technologii informacyjnych, mediów i komunikacji (TMT), transportowej (TSL), produkcji przemysłowej, a także sektora publicznego."

[3] K. Karol, *Uczenie maszynowe w opiece zdrowotnej.* Roczniki Kolegium Analiz Ekonomicznych/Szkoła Główna Handlowa 56, 2019, pp. 305–316.

[4] V. Nasteski, "An overview of the supervised machine learning methods."

[5] J. Grus, *Data science from scratch:first principles with python.* pp. r11 pp140.

[6] T. Łysiak, *The use of machine learning methods in predicting stock prices on the stock exchange.*

[7] J. Laska, "An overview of machine learning methods used in sentiment analysis."

[8] H. van Engelen J. E., "A survey on semi-supervised learning," *Technologie informatyczne w administracji publicznej i służbie zdrowia*, pp. 373–440, 2020.

[9] D. Dua and C. Graff, "UCI machine learning repository [http://archive.ics.uci.edu/ml]," *Machine Learning Technical Reports*, vol. 1(1), pp. 1–6, 2019.

[10] dr n. med. Robert Detrano,.

[11] M. Andras Janosi,.

[12] W. S. M. M. P. MD,.

[13] R. H. F. Peshawa J. Muhammad Ali, "Data normalization and standardization: A technical report," *Machine Learning Technical Reports*, vol. 1(1), pp. 1–6, 2014.

[14] D. Kumar, "Introduction to data preprocessing in machine learning beginners guide for data preprocessing."

[15] A. G. D. Anguita L. Ghelardoni, "The 'k' in k-fold cross validation," *Journal of Machine Learning Research.*

[16] G. Bonaccorso, *Mastering machine learning algorithms: Expert techniques to implement popular machine learning algorithms and fine-tune your models.*

[17] K. Matthew, *Thoughtful machine learning with python a test-driven approach.* pp. r.1 pp8.

[18] T. D. Breiman L., *Random forests, machine learning.* StatSoft Polska Sp. z o. o, 2001.

[19] S. HUANG, N. CAI, P. P. PACHECO, S. NARRANDES, Y. WANG, and W. XU, "Applications of support vector machine (SVM) learning in cancer genomics," *Cancer Genomics & Proteomics*, vol. 15, no. 1, pp. 41–51, 2018, Available: https://cgp.iiarjournals.org/content/15/1/41

[20] A. Pielowska, "Maszyna wektorów nośnych."

[21] H. S. R. Sudip Karki, "Comparison of a*, euclidean and manhattan distance using influence map in ms. Pac-man," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[22] C. B. Binbin Lua Martin Charltonb and P. Harrisc, "The minkowski approach for choosing the distance metric in geographically weighted regression."

[23] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[24] V. S. W. Samrudhi Rajendra Kaware, *Podejście porównawcze do algorytmów uczenia się maszynowego.* Reading, Massachusetts: Addison-Wesley, 1993.

[25] M. Grinberg, *Flask web development: Developing web applications with python.* " O'Reilly Media, Inc.", 2018.