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

Magdalena Szulc

2022-02-11

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 it, most often it is not information that is made 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 preliminary processing, the dataset itself, due to its small size, allows you to check the operation 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 ease 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 ich podstawie wyszukiwane są zależności znaczące oraz budowany jest model do przewidywania wartości.

W przypadku danych dotyczących chorówb 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. Objawy są tym silniejsze im gorszy jest stan chorobowy pacjenta.

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.

Zastosowanie uczenia maszynowego w medycynie, pozwala na przetwarzanie dużych zasobów historycznych wyników medycznych, głównie zależności przyczynowo skutkowych , które mogą zostać wykorzystane do diagnostyki lub leczenia [3].

Słowa kluczowe: uczenie maszynowe, uczenie nadzorowane

[1] Wojciech Modrzejewski and Włodzimierz J. Musiał tyt.:"Stare i nowe i czynniki ryzyka sercowo-naczyniowego - jak zahamować epidemię miażdżycy? Część I. Klasyczne czynniki ryzyka", Forum Zaburzeń Metabolicznych 2010;1(2):106-114.

[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] Korczak, Karol. "Uczenie maszynowe w opiece zdrowotnej." Roczniki Kolegium Analiz Ekonomicznych/Szkoła Główna Handlowa 56 Technologie informatyczne w administracji publicznej i służbie zdrowia (2019): 305-316.

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:

- lasy decyzyjne (ang. *decisions-forests*)
- metoda 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 danych wykorzystując algorytmy do analizy zbiorów dancyh (zazwyczaj bardzo obszernych). W ten sposób utworzone modele są w stanie z wysokim prawdopodobieństwem wystawić predyckję lub dokonać klasyfikacji na temat zadanego problemu. Sposób wykorzystania segreguje alorytmy uczenia maszynowego na dwie kategorie , jednak powszechnie stosowanym podziałem jest zależnie od sposobu *trenowania* algorytmu. Algorytmy dzieli się na min. uczenie nadzorowane, uczenie bez nadzoru oraz uczenie przez wzmacnianie [4] .

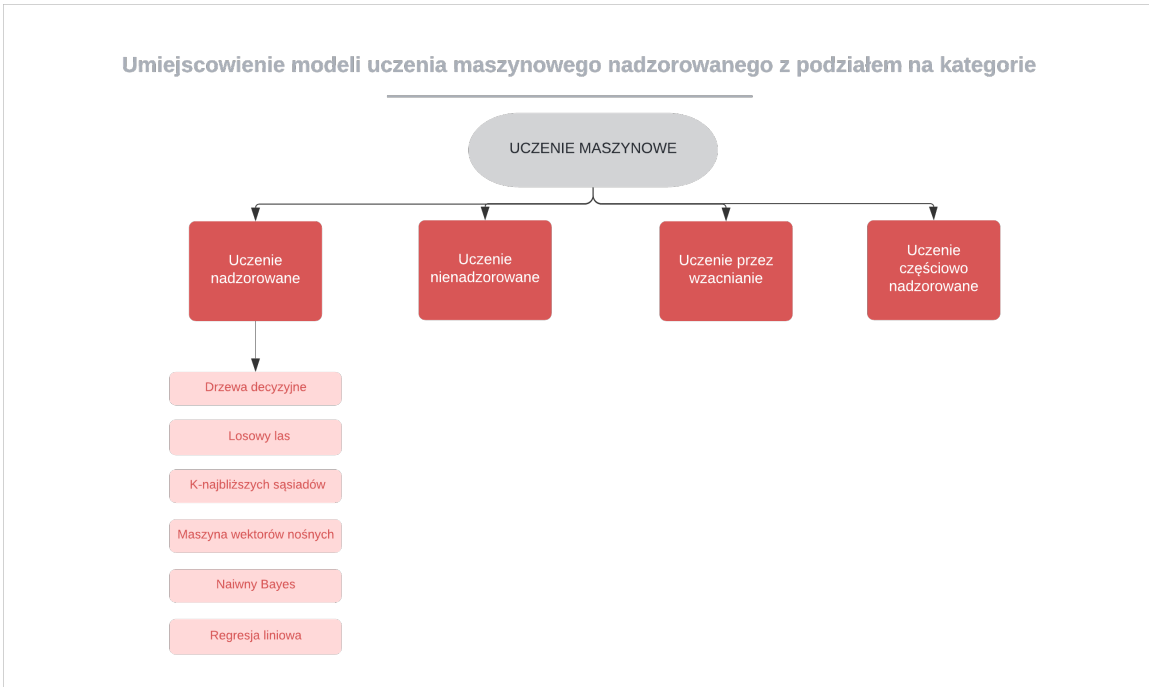

Figure 1: Schemat 1

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. Ten typ uczenia świetnie nadaje się do rozwiązywania problemów z zakresu klasyfikacji. Nadzór polega na porównaniu rezultatów działania modelu z wynikami które są zawarte w danych wejściowych(*dane oznaczone*). Algorytm po osiągnięciu żądanej efektywnosci jest w stanie dokonać klasyfikacji przykładu dla którego nie posiada odpowiedzi.

[4]Data Science from Scratch:First Principles with Python, Joel Grus, R.11,str140, Thoughtful Machine Learning with Python A Test-Driven Approach autor :Kirk Matthew r.1 str.8

Sprawdza się to obecnie w rekomendacji produktów oraz diagnozie chorób. Z matematycznego puntu widzenia dopasowanie danych oznaczonych nazywane jest aproksymacją funkcji [^machine-learning] .

Uczenie maszynowe bez nadzoru (ang. *unsupervised learning*) to klasa algroytmów uczenia maszynowego która głownie 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 [^machine-learning] .

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. [5] .

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

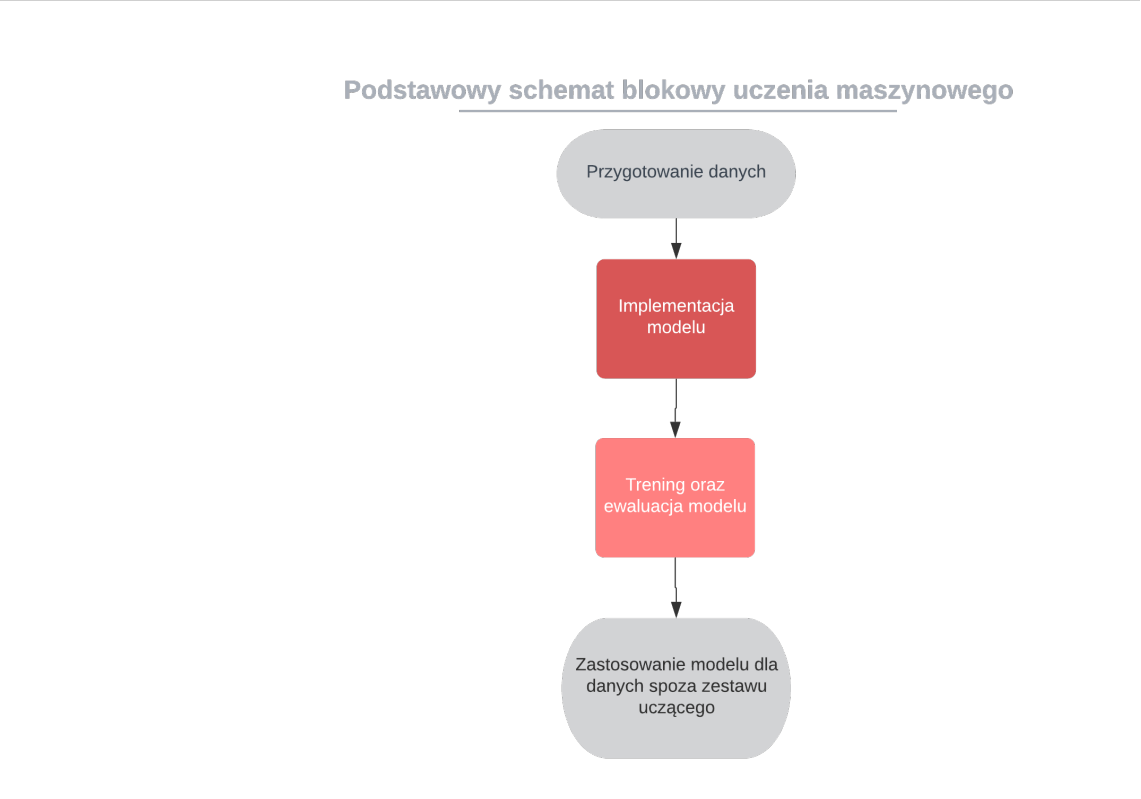

Figure 2: Schemat 2

[5] An Overview of Machine Learning Methods Used in Sentiment Analysis. Justyna Laska

Model Danych

Repozytorium uczenia maszynowego UCI

[6]

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 uczeniia. W celu gromadzenia zaufanej 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 … [7]

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 [^clevland]
2. Węgierski Instytut Kardiologii, Budapeszt [8]
3. V.A. Centrum medyczne, Long Beach, Kalifornia (long-beach-va.data) [^clevland]
4. Szpital Uniwersytecki, Zurych, Szwajcaria (switzerland.data) [9].

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

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 dataset'ów dostarcza większej różnorodności również dzięki temu że dane pochodzą z różnych krajów. Jeżeli zestaw wejściowy został by ograniczony do jednej lokalizacji to cecha dla której nie uzupełniono wartości zostałaby z autoatu pominięta jako znacząca ze względu na brak danych. Po złączeniu można przeprowadzić szereg działań w celu sztucznego uzupełnienia pustych wartości bazując na wartościach które już istnieją.

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

Obsługa brakujących wartości

Możliwości obsługi brakujących wartości są jak już przedstawiono powyżej sa 2 : mniej polecana ze względu na utrate danych, redukcja dataset'u lub uzupełnienie go zgodnie z wybrany przez siebie

[6] Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

[7] Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

[8] Węgierski Instytut Kardiologii. Budapeszt: Andras Janosi, MD

[9] Szpital Uniwersytecki, Zurych, Szwajcaria: William Steinbrunn, MD i Szpital Uniwersytecki, Bazylea, Szwajcaria: Matthias Pfisterer, MD

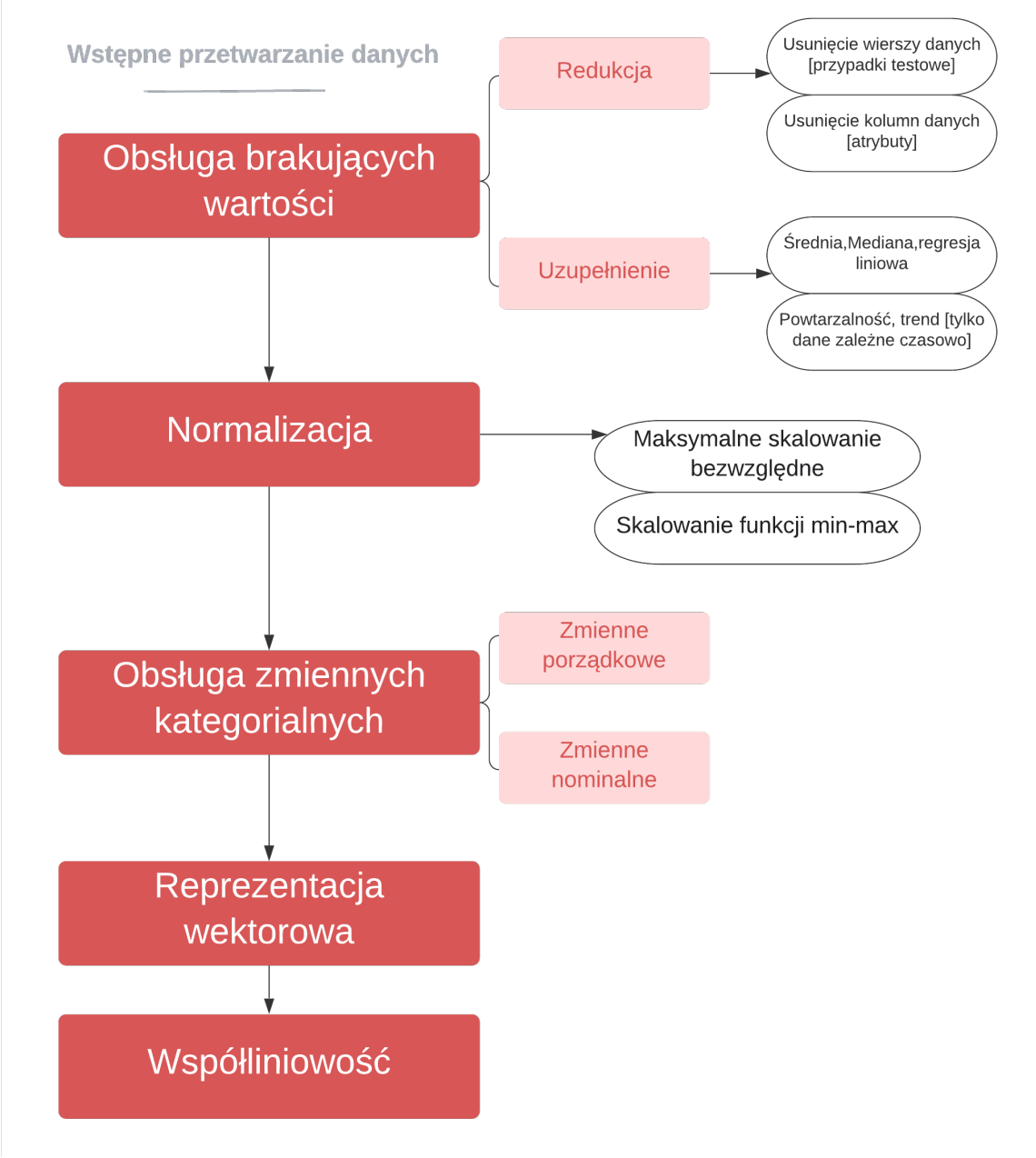

Figure 3: Schemat 4

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

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

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

Figure 4: Schemat 5

Obsługa zmiennych kategorialnych

Cechy kategorialne dzielą się na dwie zasadnicze grupy ze względu na mozliwość uprorządkowania , dane takie jak wyksztaucenie , 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.

Współliniowość cech Aby znaleść kolrelacje współliniowości należy szukać liniowej zależności pomiędzy danymi, najłatwiej zauważyć to tworząc wykresy z danych testowych dla każdej pary.

[TODO] Wwykresy dla cech

Przy zastosowaniu reprezentacji wektorowej dla cech mocno od siebie uzależnionych zalecane jest zastosowanie :

Opis praktycznej częsci projektu

Moduły projektu:

- Config - zawiera statyczne zasoby oraz konfigurację logowania projektu
- Data - moduł odpowiada za wczytywanie i obróbkę danych testowych, oraz objety danych wykorzystywanych przy uczeniu oraz zapisie 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 obrazkami , plikami stylów oraz javascript i jQuerry wykorzystywanych przez Flask
- Templates - folder z stronami html

Projekt posiada dwa tryby pracy :

- tryb nauczania na podstawie danych testowych - machine learning z wykorzystaniem 3 algorytmów
- tryb aplikacji web - wykorzystanie Flask do prezentacji i wykorzystania utworzonych modeli

Poniżej przedstawiono plan działania:

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, zestaw narzędzi do obliczeń na macierzach, wektorach oraz umożliwiający metody numeryczne takie jak całkowanie, różniczkowanie itp [10].
Do przygotowania danych wykorzystano zestaw narzędzi Pandas, ułatwiający tworzenie struktur danych i ich analizę. W celu wizualizacji wyników w postaci wykresów zastosowano Matplotlib. Część prezentacyjna czyli możliwość wprowadzenia danych w formularzu na stronie i weryfikacja wyniku dla wyuczonych już modeli wykorzystuje bibliotkę Flask. Przekazywanie obiektów o bardziej skomplikowanej budowie serializowano do formatu JSON są za pomocą biblioteki jsonpickle, a zapis modeli wykonano za pomocą joblib która zapewnia taką obsługę obiektów Pythona.

Biblioteki w większości posiadają otwarty kod źródłowy , głównie napisany w języku Python.

[10] @article{scikit-learn, title={Scikit-learn: Machine Learning in {P}ython}, author={Pedregosa, F. and Varoquaux, G. and Gramfort, A. and Michel, V. and Thirion, B. and Grisel, O. and Blondel, M. and Prettenhofer, P. and Weiss, R. and Dubourg, V. and Vanderplas, J. and Passos, A. and Cournapeau, D. and Brucher, M. and Perrot, M. and Duchesnay, E.}, journal={Journal of Machine Learning Research}, volume={12}, pages={2825–2830}, year={2011}}

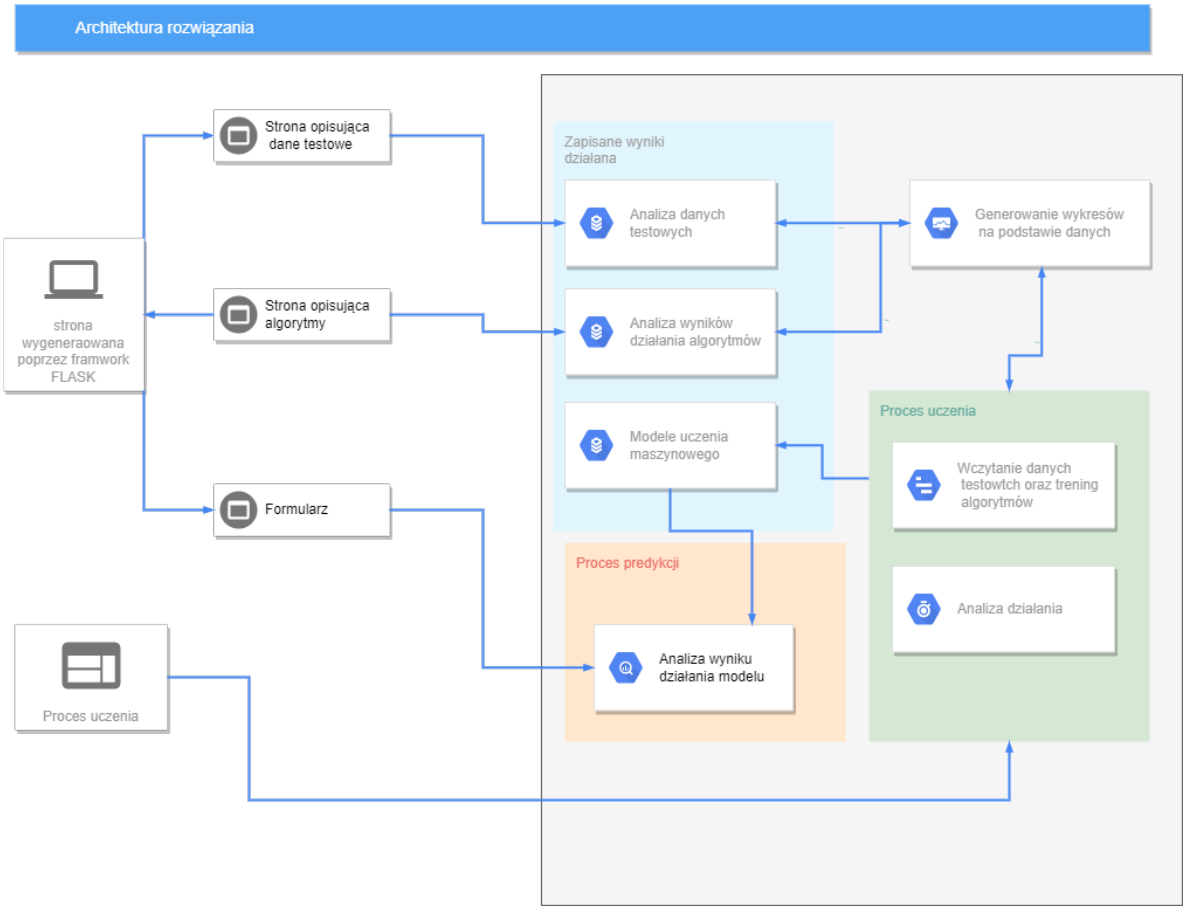

Figure 5: Schemat 6

Figure 6: Schemat 7

Trening algorytmu

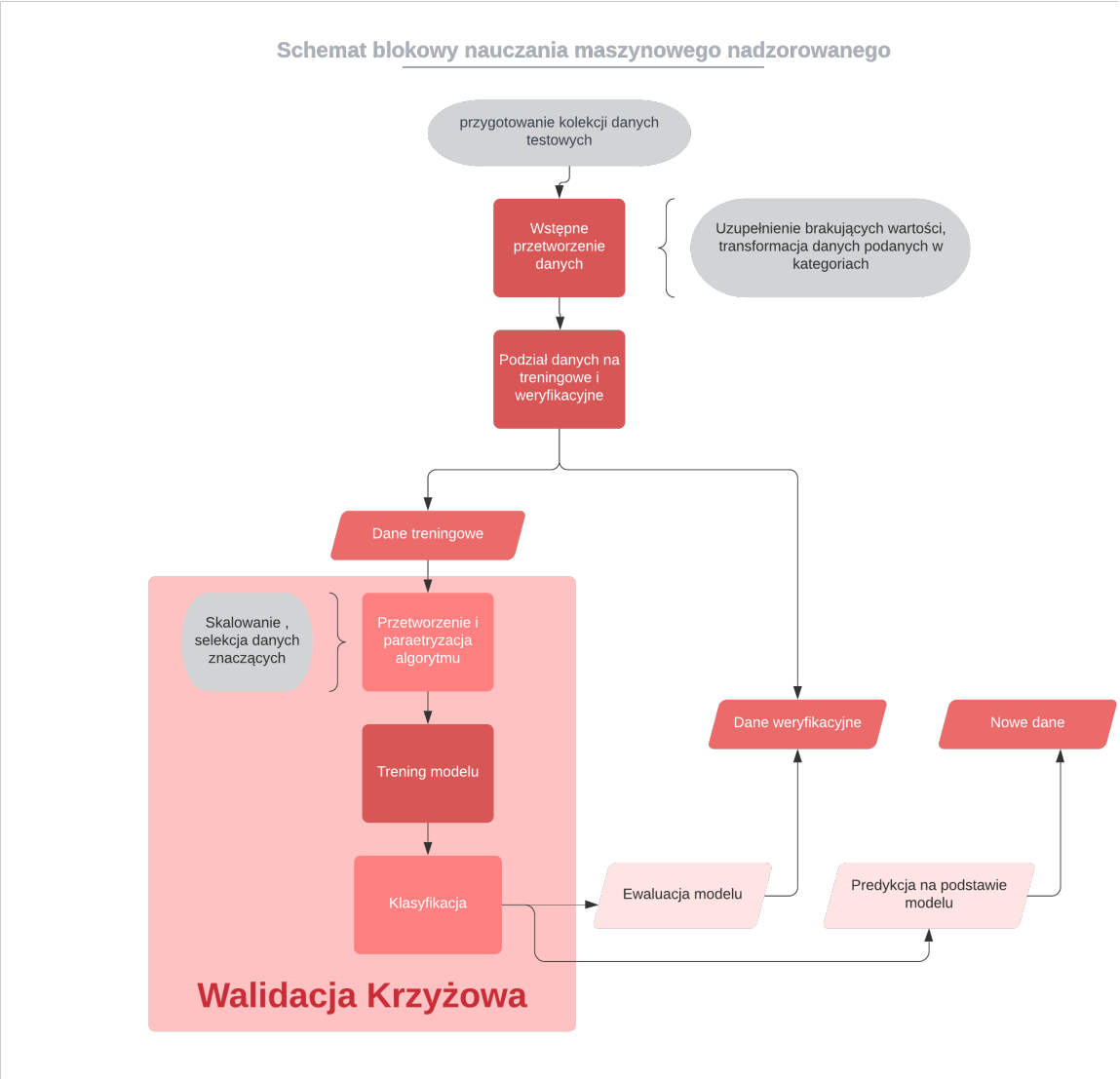

Figure 7: Schemat 8

Zgodnie z powyższym schematem po przeprocesowaniu wejściowego dataset'u, dane należy 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 [11] .

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.

[11] The 'K' in K-fold Cross Validation Authors: D. Anguita,L. Ghelardoni,A. Ghio,ONETO, LUCA,S. Ridella

Wybrane algorytmy uczenia maszynowego nadzorowanego

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 kolejnośc cech, które występują po sobie ponieważ determinuje to otrzymany rezultat [12] .

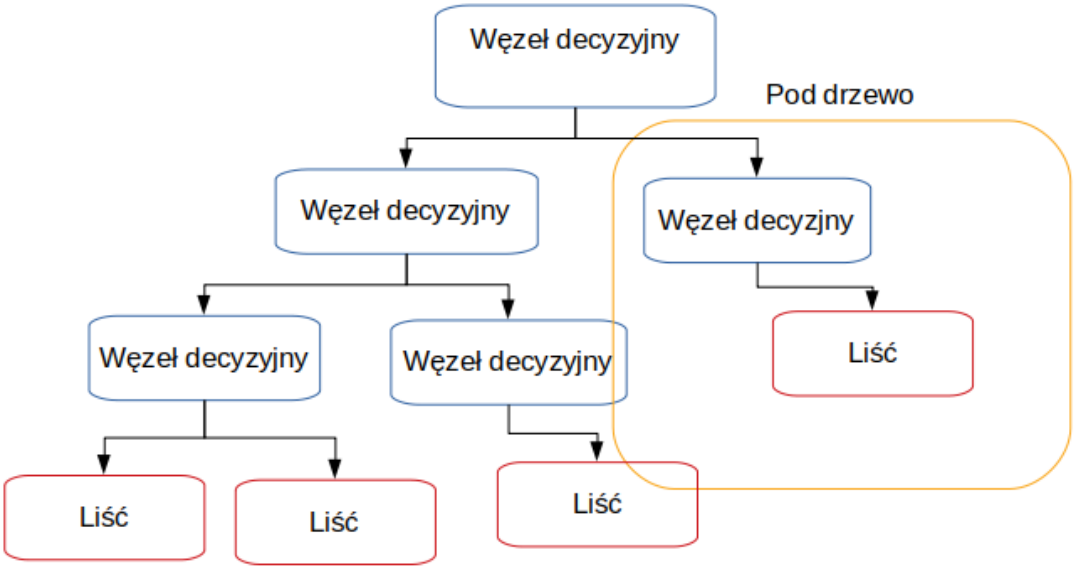

Figure 8: Schemat 9

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 [13].

W tym celu dla loswoych lasów decyzyjnych najpierw stosuję się **agregację bootstrap'ową**.

Z treningowego zestawu danch losuję się, co ważne 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.

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.

[12] Data Science from Scratch:First Principles with Python, Joel Grus, R.11,str140, Thoughtful Machine Learning with Python A Test-Driven Approach autor :Kirk Matthew r.1 str.8

[13] Data Science from Scratch:First Principles with Python, Joel Grus, R.11,str140, Thoughtful Machine Learning with Python A Test-Driven Approach autor :Kirk Matthew r.1 str.8

Dane medyczne posiadają szeroką wariację zmiennych z dużym prawdopodobieństwem wybrakownia, zastosowanie do nich lasów decyzyjnych ma potencjał na pozytywne rezulataty.

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.Taka prosta nazywana jest *prostą marginalną* i powstaje ona poprzez generowanie i selekcję tych prostych które rzetelnie szufladkują klasy danych [14].

[Schemat 10][15](img/10svm_schemat.png "Schmat SVM")

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.

[Schemat 11] [16](img/9svm_niemozliwy_podzial_schemat.png "Schmat SVM niemożliwy podział")

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 [17] .

Atrybut który nastraja proces uczenia się modelu i ma na niego największy wpływ określany jest jako hiperparametr. Dla KNN jest to liczba sąsiadów, im większa ilość jednostek mających wpływ, tym wierniejsze będą wyniki. Potęguje się wtedy niestety złożoność czasowa algorytmu, znacząco już większa od przedstawionych powyżej innych algoryrtmów.

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.

[Schemat 12][18](img/5knn_schemat.png "Schmat KNN")

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.

Budowa modelu

Budowanie zbiorów danych

Ta faza obejmuje podział przetworzonych danych na trzy zestawy danych — szkolenie, walidację i testowanie:

Zestaw uczący — służy do wstępnego uczenia algorytmu i uczenia go, jak przetwarzać informacje. Ten zestaw definiuje klasyfikacje modeli za pomocą parametrów. Zestaw walidacyjny — używany do oszacowania dokładności modelu. Ten zestaw danych służy do dostrajania parametrów modelu. Zestaw testowy — służy do oceny dokładności i wydajności modeli. Ten zestaw ma na celu ujawnienie wszelkich problemów lub błędów w modelu. Szkolenie i doskonalenie

Gdy masz już zestawy danych, możesz rozpocząć trenowanie modelu. Wiąże się to z wprowadzeniem zestawu treningowego do algorytmu, aby mógł nauczyć się odpowiednich parametrów i cech używanych w klasyfikacji.

Po zakończeniu szkolenia możesz udoskonalić model, korzystając ze swojego zestawu danych do walidacji. Może to obejmować modyfikację lub odrzucenie zmiennych i obejmuje proces dostrajania ustaw-

[14]Data Science from Scratch:First Principles with Python, Joel Grus, R.11,str140, Thoughtful Machine Learning with Python A Test-Driven Approach autor :Kirk Matthew r.1 str.8

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

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

[17]Data Science from Scratch:First Principles with Python, Joel Grus, R.11,str140, Thoughtful Machine Learning with Python A Test-Driven Approach autor :Kirk Matthew r.1 str.8

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

ień specyficznych dla modelu (hiperparametrów) aż do osiągnięcia akceptowalnego poziomu dokładności.

Ocena uczenia maszynowego

Wreszcie, po znalezieniu akceptowalnego zestawu hiperparametrów i zoptymalizowaniu dokładności modelu, możesz przetestować swój model. Testowanie wykorzystuje Twój testowy zestaw danych i ma na celu sprawdzenie, czy Twoje modele używają dokładnych funkcji. Na podstawie otrzymanej opinii możesz wrócić do trenowania modelu, aby poprawić dokładność, dostosować ustawienia wyjściowe lub wdrożyć model w razie potrzeby.

Jakie są najlepsze praktyki uczenia maszynowego dla wydajnych przepływów pracy? Podczas definiowania przepływu pracy dla projektu uczenia maszynowego można zastosować kilka najlepszych praktyk. Poniżej kilka na początek.

Zdefiniuj projekt

Dokładnie zdefiniuj cele projektu przed rozpoczęciem, aby upewnić się, że modele dodają wartość do procesu, a nie redundancję. Definiując swój projekt, weź pod uwagę następujące aspekty:

Jaki jest Twój obecny proces — zazwyczaj modele są zaprojektowane w celu zastąpienia istniejącego procesu. Ważne jest zrozumienie, jak działa istniejący proces, jakie są jego cele, kto go wykonuje i co liczy się jako sukces. Zrozumienie tych aspektów pozwala wiedzieć, jakie role musi pełnić Twój model, jakie ograniczenia mogą istnieć w implementacji oraz jakie kryteria musi spełniać lub przekraczać model. Co chcesz przewidzieć — dokładne zdefiniowanie tego, co chcesz przewidzieć, jest kluczem do zrozumienia, jakie dane należy zbierać i jak należy trenować modele. Chcesz być jak najbardziej szczegółowy na tym etapie i upewnić się, że wyniki zostały określone ilościowo. Jeśli twoje cele nie są mierzalne, będziesz miał trudności z zapewnieniem, że każdy z nich zostanie osiągnięty. Jakie są Twoje źródła danych — oceń, na jakich danych opiera się Twój bieżący proces, w jaki sposób są gromadzone i w jakiej objętości. Z tych źródeł należy określić, jakie konkretne typy danych i punkty są potrzebne do tworzenia prognoz. Znajdź podejście, które działa

Celem wdrożenia przepływów pracy uczenia maszynowego jest poprawa wydajności i/lub dokładności bieżącego procesu. Aby znaleźć podejście, które pozwoli osiągnąć ten cel, musisz:

Badania — przed wdrożeniem podejścia należy poświęcić czas na badanie, w jaki sposób inne zespoły wdrożyły podobne projekty. Możesz być w stanie pożyczyć metody, których używali lub uczyć się na ich błędach, oszczędzając czas i pieniądze. Eksperyment — niezależnie od tego, czy znalazłeś istniejące podejście, aby zacząć od lub stworzyłeś własne, musisz z nim poeksperymentować. Jest to zasadniczo faza uczenia i testowania Twojego modelu. Zbuduj rozwiązanie na pełną skalę

Opracowując swoje podejście, końcowy rezultat jest zazwyczaj dowodem koncepcji. Musisz jednak umieć przełożyć ten dowód na funkcjonalny produkt, aby osiągnąć swój cel końcowy. Aby przejść z rozwiązania testowego do rozwiązania wdrażalnego, potrzebne są:

Testy A/B — umożliwiają porównanie bieżącego modelu z istniejącym procesem. Może to potwierdzić lub zaprzeczyć, czy Twój model jest skuteczny i może stanowić wartość dodaną dla Twoich zespołów i użytkowników. API uczenia maszynowego — tworzenie interfejsu API do implementacji modelu umożliwia komunikację ze źródłami danych i usługami. Ta dostępność jest szczególnie ważna, jeśli planujesz oferować swój model jako usługę uczenia maszynowego. Dokumentacja przyjazna dla użytkownika — obejmuje dokumentację kodu, metod i sposobu korzystania z modelu. Jeśli chcesz stworzyć produkt rynkowy, musi być jasne dla użytkowników, w jaki sposób mogą wykorzystać model, jak uzyskać dostęp do jego wyników i jakich wyników mogą się spodziewać. Automatyzacja przepływów pracy uczenia maszynowego Automatyzacja przepływów pracy uczenia maszynowego umożliwia zespołom wydajniejsze wykonywanie niektórych powtarzalnych zadań związanych z tworzeniem modeli. Istnieje wiele modułów i coraz więcej platform do tego celu, czasami określanych jako autoML.

Co to jest zautomatyzowane uczenie maszynowe? AutoML zasadniczo stosuje istniejące algorytmy uczenia maszynowego do opracowywania nowych modeli. Jego celem nie jest automatyzacja całego procesu tworzenia modelu. Zamiast tego ma zmniejszyć liczbę interwencji, które ludzie muszą wykonać, aby zapewnić pomyślny rozwój.

AutoML pomaga programistom znacznie szybciej rozpocząć i ukończyć projekty. Może również usprawnić procesy uczenia głębokiego i nienadzorowanego uczenia maszynowego, potencjalnie umożliwiając samokorektę w opracowanych modelach.

Train Test Split train/test splitting techniques, exploiting three different Python libraries:

Zwykle proces uczenia/podziału testów jest jednym z zadań uczenia maszynowego, które są uznawane za oczywiste . W rzeczywistości naukowcy zajmujący się danymi koncentrują się bardziej na wstępnym przetwarzaniu danych lub inżynierii funkcji, delegując proces dzielenia zestawu danych na wiersz kodu. W tym krótkim artykule opiszę trzy techniki dzielenia uczenia/testowania, wykorzystujące trzy różne biblioteki Pythona: nauka-scikit pandy NumPy W tym samouczku zakładam, że cały zestaw danych jest dostępny jako plik CSV, który jest ładowany jako Pandas Dataframe. Rozważam heart.csvzbiór danych, który ma 303 wiersze i 14 kolumn: importuj pandy jako PD df = pd.read_csv('źródło/serce.csv') Ramka danych Pandy Obraz autora Kolumna wyjściowa odpowiada kolumnie docelowej, a wszystkie pozostałe odpowiadają cechom wejściowym: Y_col = „wyjście" X_cols = df. loc [:, df.columns != Y_col].columns 1 Nauka scikitu Scikit-learn udostępnia funkcję o nazwie train_test_split(), która automatycznie dzieli zbiór danych na zbiór uczący i testowy. Jako parametry wejściowe funkcji można przekazać listy lub ramki danych Pandas. ze sklearn.model_selection importuj train_test_split X_train, X_test, y_train, y_test = train_test_split (df[X_cols], df[Y_col],test_size=0.2, random_state=42) Inne parametry wejściowe obejmują: test_size: część zbioru danych, która ma zostać uwzględniona w zbiorze danych testowych. random_state: liczba nasion, która ma zostać przekazana do operacji tasowania, dzięki czemu eksperyment będzie powtarzalny. Oryginalny zbiór danych zawiera 303 rekordy, train_test_split()funkcja test_size=0.20przypisuje 242 rekordy do zestawu uczącego i 61 do zestawu testowego. 2 pandy Pandas udostępnia funkcję Dataframe o nazwie sample(), która może być używana do dzielenia Dataframe na zestawy pociągowe i testowe. Funkcja otrzymuje jako dane wejściowe fracparametr, który odpowiada proporcji zbioru danych, który ma zostać uwzględniony w wyniku. Podobnie jak scikit-learn train_test_split(), również sample()funkcja dostarcza random_stateparametr wejściowy. Funkcji sample()można użyć do wyodrębnienia zestawu treningowego: df_pociąg = df. próbka (frac=0.8, random_state=1) natomiast zestaw testowy można wyodrębnić, upuszczając zestaw uczący z oryginalnego zestawu danych: df_test=df.upuszczać(df_train.index) Zmienne X i Y można wyodrębnić, wybierając odpowiednie kolumny ze zbiorów uczących i testowych: X_train = df_train[X_cols] X_test = df_test[X_cols] y_train = df_train[Y_col] y_test = df_test[Y_col]3 np.random.rand() Podobnie do train_test_split()funkcji, funkcja sample() z frac=0.8przypisuje 242 rekordy do zbioru uczącego i 61 do zbioru testowego. Oczywiście rekordy zawarte w zbiorach danych tworzonych przez sample()różnią się od tych tworzonych przez train_test_split(). 3 Nudny W pakiecie Numpy możemy wykorzystać tę rand()funkcję do wygenerowania listy losowych elementów od 0 do 1. Dokładniej, możemy wygenerować listę o tej samej długości co Dataframe. Następnie możemy stworzyć maskę o wartościach <0,8, a następnie użyć tej maski do zbudowania zbiorów uczących i testowych: importuj numer jako np maska = np.losowy.rand(len(df)) <0,8 df_train = df[maska] df_test = df[~maska] W przeciwieństwie do train_test_split()i sample(), ta strategia nie generuje stałej liczby próbek dla zbiorów uczących i testowych. W tym konkretnym przykładzie liczba próbek w zbiorze uczącym wynosi 256 (w porównaniu z 242 innymi strategiami) i 47 dla zbioru testowego. Streszczenie W tym krótkim artykule zilustrowałem trzy strategie dzielenia zbioru danych, dostarczonego jako Ramka danych Pandas, na zbiory pociągowe i testowe.

Feature Scaling Why Should we Use Feature Scaling? The first question we need to address – why do we need to scale the variables in our dataset? Some machine learning algorithms are sensitive to feature scaling while others are virtually invariant to it. Let me explain that in more detail. Training and Predictions Evaluating the Algorithm

Implementacja algorytmu 1: Losowe lasy decyzyjne

++ Drzewa decyzyjne to metody wykorzystujące szereg zasad decyzyjnych do wytrenowania modelu w oparciu o zbiór uczący w celu generowania przyszłych predykcji na podstawie zmiennych objaśniających. Poniżej znajduje się schemat przedstawiający przykładową strukturę drzewa decyzyjnego (patrz rysunek 3.6) oraz opis jego najważniejszych składowych. Rys. 3.6. Poglądowy schemat przedstawiający przykład-

ową strukturę modelu drzewa decyzyjnego. 32 Metody uczenia maszynowego • Gałąź - krawędź łącząca pozostałe elementy drzewa • Węzeł - wierzchołek łączący co najmniej jedną gałąź • Korzeń - główny węzeł drzewa • Liść - węzeł z którego nie wychodzi żadna gałąź Głównym problemem w efektywnym korzystaniu z drzew decyzyjnych jest dobór odpowiedniej struktury dla rozważanego zagadnienia. W tym celu stosuje się algorytmy rekurencyjne umożliwiające maksymalizację zdobywania najistotniejszych informacji z punktu widzenia rozwiązywanego problemu podczas dokonywania decyzji oraz podziału w każdym węźle. W przypadku przewidywania cen na giełdzie papierów wartościowych wyznacznikiem dla reguły decyzyjnej mogłaby być wartość błędu średniokwadratowego postaci: MSE = 1 n Xn i=1 (yi − yˆi) 2 • yi - rzeczywista wartość wyniku • yˆi - przewidziana wartość Algorytm rekurencyjny po znalezieniu najskuteczniejszych reguł decyzyjnych wykrytych na podstawie minimalizacji wartości błędu średniokwadratowego oraz zmiennych objaśniających mógłby generować przewidywania notowań spółek zamknięcia kolejnego dnia działa giełdy dla nieoznaczonych danych wejściowych. W celu przewidywania jeszcze dokładniejszych wyników mógłby zostać wykorzystany algorytm lasów losowych. Metoda ta polega na wykorzystaniu wielu drzew decyzyjnych w celu wygenerowania predykcji, a następnie wybrania z nich najczęściej występującego rezultatu.
++

Implementacja algorytmu 2: Metoda wektorów nośnych

++
Maszyna wektorów nośnych jest techniką wykorzystującą wielowymiarową przestrzeń w celu znalezienia maksymalnej hiperpłaszczyzny brzegowej umożliwiającej najbardziej precyzyjny sposób podziału danych na klasy. Wykres przedstawiający przykładową predykcję modelu znajduje się na rysunku 3.5. Nowy obiekt Y X Rys. 3.5. Poglądowy wykres przedstawiający przykładową predykcję modelu maszyny wektorów nośnych. Idea działania maszyny wektorów nośnych opiera się na wyszukiwaniu maksymalnych odległości pomiędzy najbliższymi punktami nazywanymi marginesami zgodnie ze wzorem: f(x) = 0 + X i S iK(xi , xi

0)

• 0 - wyraz wolny • S - zbiór wszystkich obserwacji wektora nośnego • - parametry modelu przeznaczone do nauki • (xi , xi

0) • pary obserwacji wektora nośnego • K - funkcja nazywana kernelem porównująca podobieństwo pomiędzy xi i xi 0 3.7 Drzewa decyzyjne i lasy losowe 31 W przypadku przewidywania notowań spółek na giełdzie papierów wartościowych maszyna wektorów nośnych mogłaby zostać wykorzystana podobnie jak w przypadku regresji logistycznej oraz algorytmu k najbliższych sąsiadów w celu predykcji przyszłych ruchów kursów akcji. Algorytm na podstawie danych wejściowych oznaczających historyczne wartości spółek przewidywałby ruch wzrostowy, spadkowy lub horyzontalny na zakończenie kolejnego dnia działania giełdy względem poprzedniego. Oczywiście predykcje odbywałyby się przy wykorzystaniu wytrenowanego wcześniej modelu sekwencyjnie minimalizującego wybraną funkcję błędu w oparciu o próby uczące
++

Implementacj algorytmu 3: K najbliższych sąsiadów

Algorytm ten służy do rozwiązywania problemów modelu klasyfikacyjnego. K-najbliższy sąsiad lub algorytm K-NN w zasadzie tworzy wyimaginowaną granicę do klasyfikacji danych. Gdy pojawią się nowe punkty danych, algorytm spróbuje przewidzieć to z dokładnością do najbliższej linii granicznej.

Dlatego większa wartość k oznacza gładsze krzywe separacji, co skutkuje mniej złożonymi modelami. Natomiast mniejsza wartość k powoduje przepełnienie danych i prowadzi do złożonych modeli.

Uwaga: Bardzo ważne jest, aby podczas analizowania zestawu danych mieć odpowiednią wartość k, aby uniknąć nadmiernego i niedopasowanego zestawu danych.

Używając algorytmu k-najbliższego sąsiada dopasowujemy dane historyczne (lub trenujemy model) i przewidujemy przyszłość.

++ Algorytm k najbliższych sąsiadów Algorytm k najbliższych sąsiadów, w odróżnieniu od omówionych powyżej metod, nie opiera się na trenowaniu modelu w celu generowania predykcji zmiennych objaśnianych. Z tego powodu nazywany jest również często algorytmem leniwym. Idea jego działania polega na przyporządkowaniu wszystkim danym wejściowym zestawu cech oraz umieszczeniu ich w wielowymiarowej przestrzeni w oparciu o miarę podobieństwa. W przypadku przekazania do algorytmu próby nieoznaczonej następuje wyszukanie k najbliższych obiektów przy pomocy określonej metody. Najczęściej wykorzystywane są do tego celu następujące miary odległości: • Euklidesowa: d = vuutXn i=1 (xi − yi) 2 3.5 Algorytm k najbliższych sąsiadów 29 • Manhattan: d = Xn i=1 |xi − yi | • Minkowskiego: d = Xn i=1 |xi − yi | p !1 p – xi i yi - obserwacje służące do obliczania odległości, – P - parametr przyjmujący wartość 1 dla odległości Manhattan oraz wartość 2 w przypadku odległości Euklidesowej. W kolejnym kroku następuje zliczenie wystąpień wszystkich klas z wybranego zbioru najbliższych sąsiadów i przyporządkowanie etykiety najczęściej występującej grupy do zmiennej objaśnianej. Wykres przedstawiający przykładową predykcję modelu znajduje się na rysunku 3.4. Nowy obiekt Y X Rys. 3.4. Poglądowy wykres przedstawiający przykładową predykcję modelu algorytmu k najbliższych sąsiadów. W przypadku przewidywania notowań spółek na giełdzie papierów wartościowych algorytm k najbliższych sąsiadów mógłby znaleźć zastosowanie w predykcji przyszłych ruchów kursów akcji. Na podstawie danych wejściowych oznaczających historyczne wartości spółek przydzielałby on je do jednej z trzech grup oznaczających ruch cen instrumentów finansowych w kierunku wzrostowym, spadkowym lub horyzontalnym w czasie kolejnego dnia działania giełdy. ++++++++++++++++++++++++++++++++++

Wnioski i walidacja rozwiązania

++
W dalszej części bieżącej sekcji znajdują się wyniki przeprowadzonych eksperymentów. W badaniach zostało wykonanych 25 pełnych iteracji przy wykorzystaniu całego zbioru treningowego. Zgodnie z przyjętą metodyką ewaluacji opisaną we wcześniejszym fragmencie niniejszego rozdziału wszystkie iteracje zakończone były procesem weryfikacji skuteczności każdej z architektur na podstawie zbioru testowego. Niższa uzyskiwana wartość błędu oznaczała wyższą skuteczność generowanych predykcji. W oparciu o uzyskane wyniki powstały poniższe wykresy odzwierciedlające cały proces nauki każdej opracowanej na potrzeby przeprowadzenia eksperymentów
++

Algorytm 1:Resultaty wnioski: Losowe lasy decyzyjne

Algorytm 2: Rezultaty wnioski: Metoda wektorów nośnych

Algorytm 3 : Rezultaty wnioski: K najbliższych sąsiadów

Plusy Faza uczenia klasyfikacji K-najbliższego sąsiada jest znacznie szybsza w porównaniu z innymi algorytmami klasyfikacji. Nie ma potrzeby uczenia modelu do uogólniania, dlatego KNN jest znany jako prosty algorytm uczenia oparty na instancjach. KNN może być przydatny w przypadku danych nieliniowych. Może być używany z problemem regresji. Wartość wyjściowa obiektu jest obliczana przez średnią k wartości najbliższych sąsiadów.

Cons Faza testowania klasyfikacji najbliższych sąsiadów K jest wolniejsza i bardziej kosztowna pod względem czasu i pamięci. Wymaga dużej pamięci do przechowywania całego zestawu danych treningowych do przewidywania. KNN wymaga skalowania danych, ponieważ KNN wykorzystuje odległość euklidesową między dwoma punktami danych, aby znaleźć najbliższych sąsiadów. Odległość euklidesowa jest wrażliwa na wielkości. Obiekty o dużych jasnościach będą miały większą wagę niż obiekty o niskich jasnościach. KNN nie nadaje się również do dużych danych wymiarowych.

Jak ulepszyć KNN? Aby uzyskać lepsze wyniki, zdecydowanie zaleca się normalizację danych w tej samej skali. Ogólnie rzecz biorąc, rozważany zakres normalizacji między 0 a 1. KNN nie jest odpowiedni dla danych wielkowymiarowych. W takich przypadkach wymiar musi się zmniejszyć, aby poprawić wydajność. Również obsługa brakujących wartości pomoże nam poprawić wyniki.

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

Podsumowanie i opisanie wpływu danych na model

porównanie do dnych statystycznych

todo variants of user data preparatrio

```
        ## preparation all -> all test
        ## preparation best for best
        ## best from other to best in another  -> result and reasons for data anlayse
        ## fast not best - why is it faster
        ##
# todo prediction
# todo percentage na true false
```

porównanie wyników klasfikacji do regresji

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 wybrakowane zostanie omówony w późniejszych rozdziałach pracy.

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 dalszej części pracy omówione zostaną lasy decyzyjne.

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

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.

Spis ilustracji

Spis tabel

Bibliografia

@article{scikit-learn, title={Scikit-learn: Machine Learning in {P}ython}, author={Pedregosa, F. and Varoquaux, G. and Gramfort, A. and Michel, V. and Thirion, B. and Grisel, O. and Blondel, M. and Prettenhofer, P. and Weiss, R. and Dubourg, V. and Vanderplas, J. and Passos, A. and Cournapeau, D. and Brucher, M. and Perrot, M. and Duchesnay, E.}, journal={Journal of Machine Learning Research}, volume={12}, pages={2825–2830}, year={2011} }

- @article{https://ichi.pro/pl/uczenie-maszynowe-proste-wprowadzenie-96150019624312}
- @article{https://zpjn.wmi.amu.edu.pl/wp-content/uploads/2019/10/praca_magisterska.pdf t}
- @article{https://pdf.helion.pl/alguma/alguma.pdf}
- @article{https://towardsdatascience.com/introduction-to-data-preprocessing-in-machine-learning-a9fa83a5dc9d}
- @article{http://www.mif.pg.gda.pl/homepages/kdz/BIGDATA/AniaPielowska.pdf}
- @article{https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/}
- @article{https://myservername.com/what-is-support-vector-machine-machine-learning}
- @article{https://scikit-learn.org/stable/modules/svm.html}
- @article{https://www.hackerearth.com/practice/machine-learning/machine-learning-algorithms/ml-decision-tree/tutorial/}

- @article{https://scikit-learn.org/stable/modules/neighbors.html}
- @article{file:///C:/Users/User/Downloads/od_pojedynczych_drzew_do_losowego_lasu.pdf}
- @article{https://scikit-learn.org/stable/modules/naive_bayes.html}
- @article{https://scikit-learn.org/stable/modules/tree.html}
- @article{https://scikit-learn.org/stable/modules/feature_selection.html}
- @article{http://pages.cs.wisc.edu/~dpage/kuusisto.thesis.pdf}
- @article{http://www.bme.teiath.gr/medisp/pdfs/PhD_Glotsos_Dimitrios.pdf}
- @article{https://www.springboard.com/blog/how-to-become-a-machine-learning-engineer/}
- @article{http://www.diva-portal.org/smash/get/diva2:920202/FULLTEXT01.pdf}
- @article{https://www.techsparks.co.in/hot-topic-for-project-and-thesis-machine-learning/}
- @article{https://machinelearningmastery.com/k-fold-cross-validation/}
- @article{https://www.writemythesis.org/master-thesis-topics-in-machine-learning/}
- @article{http://mediatum.ub.tum.de/doc/1368117/47614.pdf}
- @article{https://pdfs.semanticscholar.org/0e06/561dbab0581feebe6638dc2671f94c9abf68.pdf}
- @article{https://www.cir.meduniwien.ac.at/assets/Uploads/Masterthesis-SeeboeckPhilipp-Version28-03-2015.pdf}
- @article{https://www.quora.com/Is-there-any-machine-learning-thesis-idea-in-health-care}
- @article{https://digitalcommons.odu.edu/cgi/viewcontent.cgi?referer=- @article{https://www.google.pl/&httpsred
- @article{https://www.mobt3ath.com/uplode/book/book-60163.pdf}
- @article{https://www.ilovephd.com/thesis-bank-machine-learning-2/}
- @article{https://www.digitalocean.com/community/tutorials/how-to-handle-plain-text-files-in-python-3}
- @article{https://machinelearningmastery.com/naive-bayes-for-machine-learning/}
- @article{https://machinelearningmastery.com/compare-machine-learning-algorithms-python-scikit-learn/}
- @article{https://machinelearningmastery.com/compare-machine-learning-algorithms-python-scikit-learn/}
- @article{https://elitedatascience.com/machine-learning-algorithms}
- @article{https://www.dataschool.io/comparing-supervised-learning-algorithms/}
- @article{https://medium.com/value-stream-design/online-machine-learning-515556ff72c5}
- @article{https://hackernoon.com/choosing-the-right-machine-learning-algorithm-68126944ce1f}
- @article{https://www.kaggle.com/aldemuro/comparing-ml-algorithms-train-accuracy-90}
- @article{https://www.kaggle.com/aldemuro/comparing-ml-algorithms-train-accuracy-90}
- @article{https://machinelearningmastery.com/start-here/}
- @article{https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/}
- @article{https://blog.statsbot.co/machine-learning-algorithms-183cc73197c}
- @article{https://www.analyticsvidhya.com/blog/2016/01/complete-tutorial-ridge-lasso-regression-python/}
- @article{https://scikit-learn.org/stable/modules/clustering.html}#overview-of-clustering-methods}
- @article{https://towardsdatascience.com/predicting-presence-of-heart-diseases-using-machine-learning-36f00f3edb2c}
- @article{https://towardsdatascience.com/predicting-presence-of-heart-diseases-using-machine-learning-36f00f3edb2c}
- @article{https://www.kaggle.com/cdabakoglu/heart-disease-classifications-machine-learning}
- @article{https://medium.com/@dskswu/machine-learning-with-a-heart-predicting-heart-disease-b2e9f24fee84}
- @article{https://pdfs.semanticscholar.org/d0a5/d4b8e8da3ee2a6bf8ac5d44196fb0365cf1c.pdf}
- @article{file:///home/szulce/Pobrane/Heart_Disease_Detection_by_Using_Machine_Learning_.p}df}
- @article{file:///home/szulce/Pobrane/jcm-08-01050.pdf}
- @article{http://www.cs.put.poznan.pl/alabijak/emd/12_Reprezentacja_wektorowa_slow.pdf}
- @article{https://www.hindawi.com/journals/misy/2018/3860146/}
- @article{https://pub.towardsai.net/3-different-approaches-for-train-test-splitting-of-a-pandas-dataframe-d5e544a5316}
-

@article{https://www.run.ai/guides/machine-learning-engineer/machine-learning-workflow/#:~:text=Machine%20learning%20workflows%20define%

@article{https://www.dovepress.com/ensemble-approach-for-developing-a-smart-heart-disease-prediction-syst-peer-reviewed-fulltext-article-RRCC}

- @article{https://machinelearningmastery.com/save-load-machine-learning-models-python-scikit-learn/}
-

@article{https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn?utm_source=adwords_ppc&utm_medium=cpc&utm_campaignid=1455363063&utm_adgroupid=65083631748&u
392016246653:dsa-429603003980&utm_loc_interest_ms=&utm_loc_physical_ms=1011615&gclid=Cj0KCQiA0eOPBhC

- @article{https://www.geeksforgeeks.org/k-nearest-neighbor-algorithm-in-python/}
- @article{https://m.scirp.org/papers/88650}
- @article{https://link.springer.com/chapter/10.1007/978-3-540-24668-8_21}
- @article{https://erogol.com/machine-learning-work-flow-part-1/}
- @article{https://www.annualreviews.org/doi/pdf/10.1146/annurev-fluid-010719-060214}
- @article{https://towardsdatascience.com/workflow-of-a-machine-learning-project-ec1dba419b94}
- @article{https://cloud.google.com/ai-platform/docs/ml-solutions-overview}
- @article{https://ai.ia.agh.edu.pl/_media/pl:dydaktyka:mbn:uczenie_maszynowe.pdf}
-

@article{https://www.researchgate.net/profile/Krzysztof-Krawiec/publication/235352247_Sieci_neuronowe_i_uczenie_
neuronowe-i-uczenie-maszynowe.pdf}

- @article{https://iopscience.iop.org/article/10.1088/1742-6596/1142/1/012012/pdf}
-

@article{https://www.statystyczny.pl/co-to-jest-machine-learning/#:~:text=Niekt%C3%B3rzy%20wspominaj%C4%85%

- @article{https://www.sciencedirect.com/science/article/pii/S1877050915024928}
- @article{https://machinelearningmastery.com/types-of-classification-in-machine-learning/}
- @article{https://data-flair.training/blogs/types-of-machine-learning-algorithms/}
- @article{https://ichi.pro/pl/co-to-jest-kodowanie-one-hot-i-jak-uzywac-funkcji-pandas-get-dummies-160729382340976}
- @article{https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5640485/}
- @article{https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5863635/}
- @article{https://towardsdatascience.com/heart-disease-prediction-73468d630cfc}
- @article{https://www.sciencedirect.com/science/article/pii/S187705091630638X}
-

@article{https://www.ices.on.ca/Publications/Journal-Articles/2014/Janu
Disease-Population-Risk-Tool-predictive-algorithm-for-assessing-CVD-risk}

@article{https://www.ctvnews.ca/health/test-your-risk-of-heart-disease-with-a-new-online-lifestyle-calculator-1.4030088}

- @article{https://nevonprojects.com/heart-disease-prediction-project/}
- @article{https://scikit-learn.org/stable/modules/neighbors.html}
- @article{https://searchenterpriseai.techtarget.com/definition/machine-learning-ML}
- @article{https://www.forcepoint.com/cyber-edu/machine-learning}
- @article{https://en.wikipedia.org/wiki/Supervised_learning}
- @article{https://www.techopedia.com/definition/8181/machine-learning}
- @article{https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/}

- @article{https://searchenterpriseai.techtarget.com/definition/supervised-learning}
- @article{https://deepai.org/machine-learning-glossary-and-terms/supervised-learning}
-

@article{https://towardsdatascience.com/what-are-supervised-and-unsupervised-learning-in-machine-learning-dc76bd67795d}

@article{https://towardsdatascience.com/what-are-supervised-and-unsupervised-learning-in-machine-learning-dc76bd67795d}

- @article{https://www.geeksforgeeks.org/ml-types-learning-supervised-learning/}
- @article{http://www.cs.ucr.edu/~mwile001/papers/thesis.pdf}
- @article{https://pl.wikipedia.org/wiki/Las_losowy}
- @article{https://python-graph-gallery.com/111-custom-correlogram/}
- @article{https://python-graph-gallery.com/242-area-chart-and-faceting/}
- @article{https://en.wikipedia.org/wiki/Random_forest}
- @article{https://web.stanford.edu/~hastie/Papers/ESLII.pdf}
- @article{https://www.sciencedirect.com/topics/computer-science/random-decision-forest}
- @article{https://flask.palletsprojects.com/en/1.1.x/tutorial/install/}
- @article{https://towardsdatascience.com/introduction-to-data-preprocessing-in-machine-learning-a9fa83a5dc9d}
- @article{https://scikit-learn.org/stable/modules/generated/sklearn.impute.SimpleImputer.html}
- @article{https://stackabuse.com/k-nearest-neighbors-algorithm-in-python-and-scikit-learn/}
- @article{https://dev.to/alod83/3-different-approaches-for-traintest-splitting-of-a-pandas-dataframe-31p0}
- @article{https://pub.towardsai.net/3-different-approaches-for-train-test-splitting-of-a-pandas-dataframe-d5e544a5316}
- @article{https://www.analyticsvidhya.com/blog/2020/04/feature-scaling-machine-learning-normalization-standardization/}
- @article{https://docs.python.org/3/library/itertools.html#itertools.zip_longest}
- @article{https://realpython.com/train-test-split-python-data/}
- @article{https://towardsdatascience.com/flask-and-chart-js-tutorial-i-d33e05fba845}
- @article{https://www.sciencedirect.com/science/article/pii/S2352914820300125 - pobrane jako pdfy}
- @article{https://en.wikipedia.org/wiki/Ejection_fraction}
- @article{https://docs.python.org/3/library/zipfile.html}
- @article{https://flask.palletsprojects.com/en/2.0.x/quickstart/}
- @article{https://machinelearningmastery.com/save-load-machine-learning-models-python-scikit-learn/}
- @article{https://joblib.readthedocs.io/en/latest/}
- @article{https://www.kaggle.com/prmohanty/python-how-to-save-and-load-ml-models}
- @article{https://machinelearningmastery.com/machine-learning-in-python-step-by-step/}
-

@article{https://dobrebadania.pl/zmienna-dyskretna-ang-discrete-variable/#:~:text=Zmienna%20dyskretna%20to%20ka

- https://towardsdatascience.com/data-normalization-in-machine-learning-395fdec69d02
- https://www.ritchieng.com/machine-learning-efficiently-search-tuning-param/
- https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html
- https://www.vebuso.com/2020/03/svm-hyperparameter-tuning-using-gridsearchcv/
- http://jsonpickle.github.io/#jsonpickle-usage
- https://www.geeksforgeeks.org/svm-hyperparameter-tuning-using-gridsearchcv-ml/
- https://ichi.pro/pl/jak-najlepiej-ocenic-model-klasyfikacji-51518447076743