

Sztuczna inteligencja

Sprawozdanie z projektu końcowego

Temat: Prezentacja możliwości biblioteki scikit-learn w projektach informatycznych, wymagających użycia drzew decyzyjnych.

Wykonujący projekt: **Adam Bajguz**
Magdalena Kalisz

Studia dzienne
Kierunek: Informatyka
Semestr: IV Grupa zajęciowa: **PS1**
Prowadzący ćwiczenie: **mgr inż. Dariusz Jankowski**

26 maja 2018
Data wykonania projektu

.....
Data i podpis prowadzącego

Spis treści

1 Wstęp	2
1.1 Zalety i wady drzew decyzyjnych	2
2 Idea drzew decyzyjnych	2
2.1 Drzewa klasyfikacyjne i regresyjne	2
2.2 Drzewa decyzyjne w teorii decyzji	2
2.3 Drzewa decyzyjne w typowych zastosowaniach CI: klasyfikacja danych i maszynowym uczeniu się	2
2.4 Budowa drzew decyzyjnych	2
2.5 Cechy drzew decyzyjnych	3
3 Entropia	3
4 Funkcje podziału	3
4.1 GINI	3
4.2 Information gain	3
4.3 Gain ratio	3
4.4 Random	3
5 Algorytmy budowania drzew decyzyjnych	3
5.1 Algorytm CHAID	4
5.2 Algorytm ID3	5
5.3 Algorytm C4.5	5
5.4 Algorytm C5.0	5
5.5 Algorytm CART	5
5.6 Metody poprawy wyników uzyskiwanych dzięki drzewom decyzyjnym	6
5.6.1 Przycinanie drzew decyzyjnych	6
5.6.2 Inne metody poprawy wyników	7
6 Biblioteka scikit-learn	7
7 Opis programu	7
8 Wnioski	7
9 Omówienie literatury i bieżącego stanu wiedzy w wybranej dziedzinie SI	7
10 Omówienie wykorzystanej biblioteki i ewentualnych modyfikacji kodu źródłowego	7
11 Opisanie danych wykorzystanych do testowania działania metody wybranej biblioteki	7
12 Omówienie głównych części programu/skryptu i wyników	7
13 Wnioski końcowe odnośnie biblioteki i projektu	7
Literatura	7

1 Wstęp

Drzewa decyzyjne to nieparametryczna nadzorowana metoda uczenia, która może być stosowana zarówno do klasyfikacji, jak i regresji. Drzewa decyzyjne są jednym z najczęściej stosowanych narzędzi używanych do klasyfikacji i prognozowania, ponieważ wiedza odkryta przez drzewo decyzyjne jest zilustrowana w hierarchicznej strukturze. Drzewa decyzyjne kodują zestaw reguł if-else, które mogą być używane do przewidywania zmiennej docelowej danych funkcji danych. Reguły if-else są tworzone przy użyciu zestawu danych treningowych w celu zaspokojenia jak największej liczby instancji danych treningowych [1, 2]. Pierwszy artykuł, który przedstawia podejście do klasyfikacji znane z drzew decyzyjnych, pochodzi z 1959 roku [3]. Z kolei pierwszy algorytm drzew regresyjnych został opublikowany w roku 1963 [4].

Drzewa decyzyjne to również graficzny sposób wspierania procesu decyzyjnego. Drzewo stosowane jest w teorii decyzji i ma sporo zastosowań. Może zarówno rozwiązać problem decyzyjny, jak i stworzyć plan. Metoda drzew decyzyjnych sprawdza się przede wszystkim, kiedy mamy problemy decyzyjne z wieloma rozgałęziającymi się wariantami oraz kiedy podejmujemy decyzję w warunkach ryzyka. Drzewa znalazły zastosowanie w takich dziedzinach jak botanika i medycyna. Coraz częściej sięga się do nich także w ekonomii, gdyż są w stanie ułatwiać i usprawniać komputerowe wspomaganie procesu podejmowania decyzji. Technika drzew decyzyjnych, czy też klasyfikacyjnych, jak niektórzy je nazywają, pozwala na: wyznaczenie zasad decyzyjnych opisujących reguły przypisywania obiektów do wyróżnionych klas (zasady odwołują się do wartości atrybutów opisujących obiekty) analizowanie zbioru obiektów opisywanych przez przyjęty zestaw atrybutów; celem analizy jest doskonalenie podziału obiektów na jednorodne klasy; metoda dokonywania podziału ma charakter hierarchiczny. Punktem wyjścia jest zbiór zawierający wszystkie analizowane obiekty; w trakcie analizy jest dzielony na określoną liczbę podzbiorów. W kolejnych krokach każdy z podzbiorów podlega dalszemu podziałowi; na końcu analizy każdy obiekt stanowi oddzielną klasę [5, 6, 4, 7].

1.1 Zalety i wady drzew decyzyjnych

2 Idea drzew decyzji

Idea drzew opiera się na rekursywnym podziale danych na coraz to mniejsze sterty w celu jak najlepszego dopasowania. Początkowo próbka (węzeł macierzysty, korzeń) dzielona jest na dwa lub więcej podzbiorów (węzły potomne). Natomiast węzeł optymalny wyszukuje się na podstawie wszystkich punktów węzłowych dla każdej zmiennej. Następnie proces jest powtarzany dla każdego węzła potomnego, a te podczas dzielenia traktowane są jak węzły macierzyste. Węzeł, którego nie można już podzielić nazywamy liściem, bądź węzłem końcowym, a liczbę liści - wielkością drzewa

2.1 Drzewa klasyfikacyjne i regresyjne

Do rozwiązywania problemów zarówno regresyjnych, jak i klasyfikacyjnych służy metoda CART (ang. Classification and Regression Trees). Metoda ta powstała w roku 1984. Opiera się na dwóch typach drzew: Drzewa klasyfikacyjne - służą porządkowaniu klas, charakteryzuje je kategorią zmienną docelową;

Rysunek 1: Idea drzew klasyfikacyjnych Drzewa regresyjne - służą do przewidywania wartości zmiennej docelowej, charakteryzuje je zmienną docelową typu ciągłego.

Rysunek 2: Idea drzew regresyjnych [<http://www.statystyka.az.pl/analiza-skupien/metoda-cart.php>]

2.2 Drzewa decyzyjne w teorii decyzji

2.3 Drzewa decyzyjne w typowych zastosowaniach CI: klasyfikacja danych i maszynowym uczeniu się

2.4 Budowa drzew decyzyjnych

Drzewem decyzyjnym jest graf-drzewo, które składa się z korzenia, węzłów, krawędzi oraz liści. Liście to węzły, z których nie wychodzą już żadne krawędzie. Korzeń drzewa tworzony jest przez wybrany atrybut, natomiast poszczególne gałęzie reprezentują wartości tego atrybutu. Dzięki drzewu decyzyjnemu, zbudowanemu na podstawie danych empirycznych, można sklasyfikować nowe obiekty, które nie brały udziału w procesie tworzenia drzewa. Drzewa decyzyjne charakteryzują się strukturą hierarchiczną.

Znaczy to, że w kolejnych krokach dzieli się zbiór obiektów, poprzez odpowiedzi na pytania o wartości wybranych cech lub ich kombinacji liniowych. Ostateczna decyzja zależy od odpowiedzi na wszystkie pytania. W algorytmach konstrukcji drzew jednym z kluczowych elementów jest wybór kolejności cech, według których, na poszczególnych etapach, będzie dokonywany podział zbioru obiektów. Technika drzew decyzyjnych to uzupełnienie metod klasycznych. Przykładem może tu być analiza dyskryminacyjna. Hierarchiczność podejmowania decyzji jest cechą, która wyróżnia drzewo decyzyjne od innych metod. [https://mfiles.pl/pl/index.php/Drzewo_decyzyjne]

Drzewo decyzyjne buduje modele klasyfikacji lub regresji w postaci struktury drzewa. Rozkłada zbiór danych na mniejsze i mniejsze podzbiory, a jednocześnie rozwija się przyrostowo powiązane drzewo decyzyjne. Końcowym wynikiem jest drzewo z węzłami decyzyjnymi i węzłami liści. Węzeł decyzyjny (np. Outlook) ma dwie lub więcej gałęzi (np. Sunny, Overcast i Rainy). Węzeł liści (np. Play) reprezentuje klasyfikację lub decyzję. Najwyższy węzeł decyzyjny w drzewie, który odpowiada najlepszemu predyktorowi zwanemu węzłem głównym. Drzewa decyzyjne mogą obsługiwać zarówno dane kategoryczne, jak i liczbowe. [http://www.saedsayad.com/decision_tree.htm]

2.5 Cechy drzew decyzyjnych

Drzewa decyzyjne i uczenie się drzewa decyzyjnego razem stanowią prosty i szybki sposób uczenia się funkcji, która mapuje dane x na wyniki y , gdzie x może być mieszanką zmiennych jakościowych i liczbowych, a y może być kategoryczne dla klasyfikacji lub numeryczne dla regresji [5, 8].

Zaletą drzew decyzyjnych jest to, że mogą modelować dowolny typ funkcji do klasyfikacji lub regresji, czego inne techniki nie potrafią. Ale drzewo decyzyjne jest bardzo podatne na przeuczenie i nastawienie na dane szkoleniowe. Tak więc drzewa decyzyjne są używane w przypadku bardzo dużych zbiorów danych, które są uważane za dobrze reprezentujące gruntową prawdę. Dodatkowo, niektóre algorytmy przycinania drzew są również używane do rozwiązywania problemu przeuczenia [5, 1].

Znacznie szybsze trenowanie w porównaniu z prostymi sieciami neuronowymi przy porównywalnej wydajności (złożoność czasowa drzew decyzyjnych jest funkcją zależną od liczby cech oraz wierszy w zestawie danych, podczas gdy dla sieci neuronowych jest funkcją zależną od liczby cech, wierszy w zestawie danych, warstw ukrytych oraz węzłów w każdej ukrytej warstwie) [5, 8]

Łatwo interpretowalne, odpowiednie do wyboru zmiennych - Ponadto jedną z największych zalet korzystania z drzew decyzyjnych i losowych lasów jest łatwość, dzięki której możemy zobaczyć, jakie cechy lub zmienne przyczyniają się do klasyfikacji lub regresji oraz ich względne znaczenie w oparciu o ich położenie w drzewie [5, 8].

3 Entropia

Drzewo decyzyjne jest budowane odgórnie z węzła głównego i polega na dzieleniu danych na podzbiory zawierające instancje o podobnych wartościach (jednorodnie). Algorytm ID3 wykorzystuje entropię do obliczenia homogeniczności próbki. Jeśli próbka jest całkowicie jednorodna, entropia wynosi zero, a jeśli próbka jest równo podzielona, ma entropię jednego.

4 Funkcje podziału

4.1 GINI

4.2 Information gain

4.3 Gain ratio

4.4 Random

5 Algorytmy budowania drzew decyzyjnych

Popularne pomiary dzielenia obejmują minimalizację Gini Impurity (używanego przez CART) lub maksymalizację Information Gain (używanego przez ID3, C4.5).

Dostępnych jest kilka algorytmów służących do klasyfikacji i analizy segmentacji. Wszystkie te algorytmy zasadniczo realizują to samo zadanie: dzieląc dane na kolejne podgrupy, analizują wszystkie

Tabela 1: My caption

Decision tree algorithms	Data types	Numerical data splitting method
CHAID (Kass, 1980)	Categorical	N/A
ID3 (Quinlan, 1986)	Categorical	No restrictions
C4.5 (Quinlan, 1993)	Categorical, numerical	No restrictions
C5.0	Categorical, numerical	No restrictions
CART (Breiman et al., 1984)	Categorical, numerical	Binary splits

zmienne w zbiorze danych, by znaleźć zmienną zapewniającą najlepszą klasyfikację lub predykcję. Proces jest rekursywny, a grupy są dzielone na coraz mniejsze jednostki aż do ukończenia drzewa (zgodnie z określonym kryterium zatrzymania). Zmienne przewidywane i wejściowe używane do budowania drzewa mogą być ilościowe (przedział liczbowy) lub jakościowe, w zależności od algorytmu. Jeśli zmienna przewidywana jest ilościowa, generowane jest drzewo regresji; jeśli zmienna przewidywana jest jakościowa, generowane jest drzewo klasyfikacji.

Przegląd istniejącej literatury pokazuje, że do najczęściej stosowanych algorytmów drzewa decyzyjne należą algorytm Iterative Dichotomiser 3 (ID3), algorytm C4.5, algorytm automatycznego detektora interaktywnego Chi-kwadrat (CHAID) oraz drzewo klasyfikacji i regresji (CART) algorytm. Wśród tych algorytmów są pewne różnice, z których jedną jest możliwość modelowania różnych typów danych. Ponieważ zestaw danych może być skonstruowany na podstawie różnych typów danych, np. Danych kategorycznych, danych liczbowych lub kombinacji obu, istnieje potrzeba użycia odpowiedniego algorytmu drzewa decyzyjnego, który może obsługiwać określony typ danych wykorzystywanych w zestawie danych. Wszystkie wyżej wymienione algorytmy mogą wspierać modelowanie danych jakościowych, podczas gdy tylko algorytm C4.5 i algorytm CART mogą być używane do modelowania danych liczbowych (patrz tabela 1). Różnica ta może być również wykorzystana jako wytyczna dla wyboru odpowiedniego algorytmu drzewa decyzyjnego. Inną różnicą między tymi algorytmami jest proces opracowywania modeli, szczególnie na etapie sadzenia drzew i przycinania drzew. Pod względem tego pierwszego algorytmy ID3 i C4.5 dzielą model drzewa na tak wiele potrzebnych operacji, podczas gdy algorytm CART może obsługiwać jedynie binarne podziały. Jeśli chodzi o te ostatnie, mechanizmy przycinania zlokalizowane w algorytmach C4.5 i CART wspierają usuwanie nieistotnych węzłów i rozgałęzień, ale algorytm CHAID utrudnia proces wzrostu drzewek przed nadużywaniem danych szkoleniowych (Tabela 1).

[<http://what-when-how.com/artificial-intelligence/decision-tree-applications-for-data-modelling-artificial-intelligence/>]

5.1 Algorytm CHAID

CHAID to algorytm do uczenia drzew decyzyjnych zaproponowany przez Kassa (1980). Działa podobnie do CART w tym sensie, że oba mogą być używane zarówno do klasyfikacji, jak i regresji. Ale w przeciwieństwie do CART, CHAID wewnętrznie obsługuje tylko kategoryczne funkcje. Funkcje ciągłe są najpierw konwertowane na porządkowe cechy kategoryczne algorytmu CHAID, aby móc z nich korzystać. Konwersja odbywa się poprzez binowanie wartości funkcji. Liczba pojemników (K) musi być dostarczona przez użytkownika. Biorąc pod uwagę K, predyktor jest podzielony w taki sposób, że wszystkie pojemniki mają taką samą liczbę (mniej więcej) różnych wartości predykcyjnych. Maksymalna wartość funkcji w każdym pojemniku jest używana jako punkt przzerwania.

Ważnym parametrem w procesie wzrostu drzewa CHAID jest wartość p. Wartość p jest miarą używaną do decydowania o tym, które kategorie wartości predykcyjnych mają się łączyć podczas łączenia, a także do decydowania o najlepszym atrybucie podczas dzielenia. Wartość p jest obliczana przy użyciu różnych metod testowania hipotez w zależności od rodzaju zmiennej zależnej (nominalnej, porządkowej lub ciągłej). Bardziej szczegółową dyskusję na temat algorytmu CHAID można znaleźć w dokumentacji klasy CCHAIDTree w Shogun. Przejdźmy do bardziej interesującego tematu, który uczy się używać CHAID przy użyciu API python Shoguna.

[<http://www.shogun-toolbox.org/static/notebook/current/DecisionTrees.html>]

5.2 Algorytm ID3

ID3 jest prostym algorytmem uczenia drzewa decyzyjnego opracowanym przez Rossa Quinlana. ID3 ma zastosowanie tylko w przypadkach, w których atrybuty (lub cechy) definiujące przykłady danych mają charakter kategoriowy, a przykłady danych należą do wcześniej zdefiniowanych, wyraźnie odróżnialnych (tj. Dobrze zdefiniowanych) klas. ID3 to iteracyjny chciwy algorytm, który rozpoczyna się od węzła głównego i ostatecznie buduje całe drzewo. W każdym węźle wybiera się "najlepszy" atrybut do klasyfikacji danych. Atrybut "najlepszy" jest wybierany przy użyciu metryki przyrostu informacji. Po wybraniu atrybutu w węźle, przykłady danych w węźle są podzielone na podgrupy na podstawie wartości atrybutów, które mają. Zasadniczo wszystkie przykłady danych o tej samej wartości atrybutu są umieszczane w tej samej podgrupie. Te podgrupy tworzą dzieci obecnego węzła, a algorytm jest powtarzany dla każdego z nowoutworzonych węzłów potomnych. To trwa dopóki wszystkie elementy danych węzła nie należą do tej samej klasy lub wszystkie atrybuty zostaną wyczerpane. W tym ostatnim przypadku przewidywana klasa może być błędna i ogólnie tryb klas pojawiających się w węźle jest wybierany jako klasa predykcyjna [<http://www.shogun-toolbox.org/static/notebook/current/DecisionTrees.html>]

5.3 Algorytm C4.5

Algorytm C4.5 jest zasadniczo rozszerzeniem algorytmu ID3 do uczenia drzewa decyzyjnego. Ma dodatkową możliwość obsługi ciągłych atrybutów i atrybutów z brakującymi wartościami. Proces uprawy drzew w przypadku C4.5 jest taki sam jak w przypadku ID3, to jest znajdowanie najlepszego podziału w każdym węźle przy użyciu metryki przyrostu informacji. Jednak w przypadku atrybutu ciągłego algorytm C4.5 musi wykonać dodatkowy krok przekształcania go w dwuwartościowy atrybut kategoriowy, dzieląc około odpowiedniego progu. Ten próg jest wybierany w taki sposób, że wynikowy podział daje maksymalne wzmocnienie informacji. Rozpocznijmy odkrywanie algorytmu Shoguna C4.5 za pomocą zabawkowego przykładu. [<http://www.shogun-toolbox.org/static/notebook/current/DecisionTrees.html>]

5.4 Algorytm C5.0

C5.0 to najnowsza wersja Quinlana na podstawie licencji firmowej. Wykorzystuje mniej pamięci i buduje mniejsze zestawy reguł niż C4.5, a jednocześnie jest bardziej dokładny.

[<http://scikit-learn.org/stable/modules/tree.html#tree-algorithms-id3-c4-5-c5-0-and-cart>]

5.5 Algorytm CART

Klasyfikacja i regresja Drzewa lub CART w skrócie to akronim wprowadzony przez Leo Breimana w celu odnoszenia się do algorytmów Drzewo decyzyjne, które można wykorzystać do klasyfikacji lub modelowania predykcyjnego problemów z modelowaniem.

Skupimy się na użyciu CART do klasyfikacji w tym samouczku.

Reprezentacja modelu CART jest drzewem binarnym. To jest to samo binarne drzewo z algorytmów i struktur danych, nic nadzwyczajnego (każdy węzeł może mieć zero, jeden lub dwa węzły potomne).

Węzeł reprezentuje pojedynczą zmienną wejściową (X) i punkt podziału na tej zmiennej, zakładając, że zmienna jest wartością numeryczną. Węzły liści (zwane również węzłami końcowymi) drzewa zawierają zmienną wyjściową (y), która służy do prognozowania.

Po utworzeniu drzewa można nawigować za pomocą nowego wiersza danych następujących po każdym oddziale z podziałem aż do ostatecznej prognozy.

Tworzenie binarnego drzewa decyzyjnego jest w rzeczywistości procesem dzielenia przestrzeni wejściowej. Chciwe podejście jest używane do dzielenia przestrzeni zwanej rekurencyjnym rozdzielaniem binarnym. Jest to procedura numeryczna, w której wszystkie wartości są wyrównane, a różne punkty podziału są testowane i testowane za pomocą funkcji kosztów.

Dokonano podziału z najlepszym kosztem (najniższy koszt, ponieważ minimalizujemy koszty). Wszystkie zmienne wejściowe i wszystkie możliwe punkty podziału są oceniane i wybierane chciwie w oparciu o funkcję kosztów.

Regresja: funkcja kosztu, która jest zminimalizowana, aby wybrać punkty podziału, jest sumą kwadratów błędów we wszystkich próbkach treningowych, które mieszczą się w prostokącie. Klasyfikacja: Używana jest funkcja kosztu Gini, która zapewnia wskazanie, jak czyste są węzły, gdzie czystość węzła

odnosi się do mieszanych danych treningowych przypisanych do każdego węzła. Dzielenie jest kontynuowane, dopóki węzły nie zawierają minimalnej liczby przykładów treningu lub osiągnięta zostanie maksymalna głębokość drzewa [9].

[<https://machinelearningmastery.com/implement-decision-tree-algorithm-scratch-python/>]

Algorytm CART jest popularnym algorytmem uczenia drzewek decyzyjnych wprowadzonym przez Breimana i in. W odróżnieniu od ID3 i C4.5, drzewo uczenia się w tym przypadku może być używane zarówno do klasyfikacji wieloklasowej, jak i do regresji w zależności od rodzaju zmiennej zależnej. Proces uprawy drzew składa się z rekurencyjnego dwójkowego podziału węzłów. Aby znaleźć najlepszy podział w każdym węźle, rozważane są wszystkie możliwe podziały wszystkich dostępnych atrybutów predykcyjnych. Najlepszy podział to taki, który maksymalizuje pewne kryterium podziału. Do zadań klasyfikacyjnych, tj. gdy atrybut zależny jest kategoriyczny, jako kryterium podziału stosuje się indeks Giniego. Do zadań regresyjnych, tj. gdy zmienna zależna jest ciągła, stosowane jest odchylenie najmniejszych kwadratów. Daj nam poznać implementację Shogun's CART, pracując nad dwoma problemami z zabawkami, jeden na temat klasyfikacji, a drugi na temat regresji. [<http://www.shogun-toolbox.org/static/notebook/current/DecisionTrees.html>]

5.6 Metody poprawy wyników uzyskiwanych dzięki drzewom decyzyjnym

5.6.1 Przycinanie drzew decyzyjne

Wydażność drzewa można dodatkowo zwiększyć przez przycinanie. Polega na usunięciu gałęzi, które korzystają z funkcji o niskim znaczeniu. W ten sposób zmniejszamy złożoność drzewa, a tym samym zwiększamy jego moc predykcyjną, zmniejszając przeuczenie.

Przycinanie może rozpocząć się od korzenia lub liści. Najprostsza metoda przycinania rozpoczyna się na liściach i usuwa każdy węzeł z najbardziej popularną klasą w tym liściu, zmiana ta jest zachowana, jeśli nie pogarsza dokładności. Nazywa się to również zmniejszeniem przycinania błędów. Można zastosować bardziej wyrafinowane metody przycinania, takie jak przycinanie kosztów, gdy parametr uczenia (alfa) jest używany do ważenia, czy węzły mogą być usunięte w oparciu o rozmiar pod-drzewa. Jest to również znane jako najsłabsze przycinanie linków. [<https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052>]

5.6.2 Inne metody poprawy wyników

6 Biblioteka scikit-learn

7 Opis programu

8 Wnioski

9 Omówienie literatury i bieżącego stanu wiedzy w wybranej dziedzinie SI

10 Omówienie wykorzystanej biblioteki i ewentualnych modyfikacji kodu źródłowego

11 Opisanie danych wykorzystanych do testowania działania metody wybranej biblioteki

12 Omówienie głównych części programu/skryptu i wyników

13 Wnioski końcowe odnośnie biblioteki i projektu

Literatura

- [1] M. P., Decision trees.
URL <http://www.shogun-toolbox.org/static/notebook/current/DecisionTrees.html>
- [2] J. R. Quinlan, Induction of decision trees, Machine Learning 1 (1) (1986) 81–106. doi:10.1007/bf00116251.
URL <https://doi.org/10.1007/bf00116251>
- [3] B. W., Matching and prediction on the principle of biological classification, JRSS, Series C, Applied Statistics 8 (2).
- [4] J. Morgan, J. Sonquist, Problems in the analysis of survey data, and a proposal, J. Amer. Statist. Assoc. 58 (1963) 415–434.
- [5] B. L., F. J., O. R., S. C., Classification and Regression Trees, Taylor & Francis Ltd, Wadsworth, Belmont, CA, 1984.
- [6] G. E., Symboliczne metody klasyfikacji danych, PWN, Warszawa, 1998.
- [7] J. Quinlan, Decision trees and decision-making, IEEE Transactions on Systems, Man, and Cybernetics 20 (2) (1990) 339–346. doi:10.1109/21.52545.
URL <https://doi.org/10.1109/21.52545>
- [8] L. H., Decision trees and random forests for classification and regression pt.1.
URL <https://towardsdatascience.com/decision-trees-and-random-forests-for-classification-and-regression-pt-1-dbb65a458df>
- [9] B. J., How to implement the decision tree algorithm from scratch in python (2016).
URL <https://machinelearningmastery.com/implement-decision-tree-algorithm-scratch-python/>