

10.05.2020 Projekt

“Automatyczny kelner ” raport nr. 3.

Podprojekt Adama Toppmayera

Opis Podprojektu: Celem podprojektu jest dodanie sztucznej inteligencji u klientów aby mogli dokonać zamówienia na podstawie znanych im informacji o ich zainteresowaniach. Na przykład osoba lubiąca sport chętniej zamówi rybę niż pizzę.

1. Metoda uczenia:

Drzewa decyzyjne

2. Działanie drzewa

Program tworzy drzewo decyzyjne na podstawie “tabeli” w której zapisane są dane, na podstawie których projekt będzie dalej dokonywał decyzji. Najpierw wyszukiwane jest w drzewie, która kolumna daje najlepszy podział danych wejściowych poprzez obliczenie “zanieczyszczenia” podziału jakim skutkuje podział bazy danych z użyciem tej kolumny jako pytania dzielącego. Do sprawdzenia zanieczyszczenia stosuje się index “Gini” który oblicza się za pomocą wzoru:

P_i - prawdopodobieństwo metki nr i

$$1 - P_1^2 - P_2^2 - \dots - P_N^2$$

Zanieczyszczenie potem wykorzystujemy do obliczenia informacji otrzymywanych przez podział według danej kolumny za pomocą średniej ważonej zanieczyszczeń lewego i prawego poddrzewa gdzie wagą jest prawdopodobieństwo danego poddrzewa.

Jeżeli pytanie nie dzieli danych lub jeżeli dzieli dane gorzej niż pytanie węzła powyżej, to nie dodajemy go do drzewa

Kiedy wywołujemy funkcję “zgadnij” program sprawdza czy jesteśmy w Liściu (gdzie jest podana odpowiedź) lub jeżeli jesteśmy w węźle z pytaniem - sprawdza czy dane do analizy na danej kolumnie są \geq od podanej odpowiedzi w węźle jeżeli dane w

tabeli są liczbami albo czy słowo jest == do zawartego w węźle jeżeli dane słowami. Jeżeli w Liściu możliwe są dwie odpowiedzi to program losuje którą wybrać.

3. Struktura Programu

training_db - "tabela" z wartościami na których będziemy inicjować tworzenie drzewa, w której kolumny odpowiadają odpowiednio za zainteresowanie w : sporcie, czytaniu, łucznictwie, boksie oraz ostatnia kolumna odpowiadająca za jedzenie które dana osoba lubi

Question - klasa która odpowiada za porównywanie wartości z drzewem na danej kolumnie poprzez metodę "match()"

kolumny_podzial - metoda odpowiadająca za podział danych na dwa poddrzewa - te które spełniają pytanie i te które nie spełniają

gini - funkcja odpowiadająca za obliczanie indexu "Gini"

info_gain - funkcja obliczająca przyrost informacji poprzez podział na danej kolumnie

find_best_split - funkcja która wyszukuje najlepszą kolumnę po której podzielić zestaw danych

Leaf - klasa w której trzymana jest informacja z decyzją

Decision_Node - klasa która trzyma w sobie pytanie które zadaje danym i przypisuje analizowaną daną do lewego lub prawego poddrzewa

build_tree - funkcja budująca drzewo

zgadnij - funkcja zwracająca decyzję na podstawie danego drzewa i wiersza danych

4. Integracja w Projekcie

Każdy klient tworzy drzewo decyzyjne na podstawie zadanych danych.

Kiedy kelnerka podchodzi aby zebrać zamówienie - klient zadaje sobie pytanie "Co chciałbym zjeść, zważywszy na moje zainteresowania?" używając do tego metody zgadnij i wylosowanej dla niego wiersza zainteresowań.