Raport przyrostu indywidualnego z przedmiotu „ Sztuczna inteligencja”

Temat projektu: Automatyczny kelner

Metoda uczenia: Drzewo decyzyjne

Problem: Wybór produktu przez klienta

Drzewo decyzyjne to nic innego jak graficzny sposób wspierania procesu decyzyjnego. Drzewo stosowane jest w teorii decyzji i ma sporo zastosowań. Może zarówno rozwiązać problem decyzyjny, jak i stworzyć plan.

Do mojego projektu wykorzystałam decision-tree-id3. Jest to moduł stworzony do tworzenia drzew decyzyjnych wykorzystujący algorytm ID3.

Wymagania do instalacji decision-tree-id3:

- Python (>= 2.7 or >= 3.3)

- NumPy (>= 1.6.1)

- Scikit-learn (>= 0.17)

Instalacja:

pip install decision-tree-id3

or clone the project using:

git clone https://github.com/svaante/decision-tree-id3.git

cd decision-tree-id3

python setup.py install

Kiedy instalacja zakończy się pomyślnie, można podpiąć owy moduł do naszego projektu:

from id3 import Id3Estimator, export\_graphviz

W algorytmach konstrukcji drzew jednym z kluczowych elementów jest wybór kolejności cech, według których, na poszczególnych etapach, będzie dokonywany podział zbioru obiektów.

Hierarchiczność podejmowania decyzji jest cechą, która wyróżnia drzewo decyzyjne od innych metod.

W moim projekcie do stworzenia korzystam z trzech tabel przechowujących kolejno:

- nazwy cech:

feature\_names = ["kwasne","gorzkie","szybkie","pitne","slodkie","lekkostrawne","tanie","smaczne"]

- przypadki testowe

feature\_names = X = np.array([

[99,50,90,80,2,60,84,"tak"],

[0,80,76,0,32,33,0,"przepyszne"],

[20,30,95,34,45,22,36,"domowej roboty"],

…

- oraz ich decyzje

y = np.array(["chlodnik",

"ciasto czekoladowe",

"frytki",

"krokiety z miesem",

…

Następne linie kodu odpowiadają odpowiednio za tworzenie drzewa oraz na zobrazowaniu go

# tworzenie drzewa decyzyjnego

clf = Id3Estimator()

# fit - synonim do "find patterns in data"

clf.fit(X, y, check\_input=True)

export\_graphviz(clf.tree\_, "test.dot", feature\_names)…

Drzewo to jest zobrazowane (przy użyciu GraphViz ) w pliku PDF. Możemy również zobrazować do plików w formatach takich jak GIF, PNG, SVG czy PostScript. Graphviz to zestaw narzędzi do tworzenia diagramów za pomocą grafów. Tworzymy to poniższym poleceniem w wierszu poleceń:

dot -Tpdf tree.dot -o tree.pdf

Treść algorytmu

1. Oblicz entropię dla każdego atrybutu

2. Wybierz atrybut A z najniższą entropią

3. Podziel zbiór przykładów uczących ze względu na wartość atrybutu A na rozłączne podzbiory

4. Dodaj do drzewa krawędzie z warunkami:

jeśli A=a1 to ... (poddrzewo 1)

jeśli A=a2 to ... (poddrzewo 2)

...

5. Dla każdego poddrzewa wykonaj kroki od 1.

6. W każdej iteracji jeden atrybut jest usuwany. Algorytm zatrzymuje się,

gdy do rozpatrzenia nie pozostanie juz żaden atrybut lub wszystkie przykłady

w danym podrzewie mają tą samą wartość atrybutu decyzyjnego.

Źródła:

<https://svaante.github.io/decision-tree-id3/>

<http://graphviz.org/>