Teoria Transmisiunii Informației – Mini-proiect

Tema 17 – Decision Tree Learning

Abrașu Cătălin, grupa 432A

Iliescu Dan, grupa 432A

Radu Vlad-Andrei, grupa 432A

Tudosia Ștefan, grupa 432A

Tema noastră propune implementarea unui algoritm de învățare prin arbori de decizie. Am decis să implementăm un algoritm de clasificare cu ajutorul arborilor de decizie, prin două metode: categorizare după entropie, respectiv prin impuritate Gini. Atât entropia, cât și impuritatea Gini sunt măsuri pentru incertitudinea clasificării în funcție de variabila aleasă. Impuritatea Gini reprezintă frecvența cu care obiectele sunt clasificate greșit. Impuritatea Gini necesită o putere de procesare mai mică decât calcularea entropiei, deoarece nu sunt necesare calcule logaritmice.

Algoritmul preia o bază de date de pe site-ul archive.ics.uci.edu, unde se găsesc o multitudine de baze de date utilizate pentru machine learning. Am ales două baze de date, cu format similar -clasa dorită se află pe coloana 0, iar variabilele utilizate pe clasificare pe coloanele următoare-, baza de date “Wine”, care determină calitatea unui vin după 13 factori precum concentrația de alcool, intensitatea culorii etc. și baza de date “Balance Scale” pentru a prezice poziția unui balansoar în funcție de greutățile pe fiecare braț și distanța la care acestea se află față de centrul balansoarului (4 factori). Aceste date sunt preluate din baza de date (*funcția importdata*), 70% urmând a fi utilizate pentru antrenarea arborilor de decizie, iar 30% fiind utilizate pentru testarea acestora (*funcția split*).

După preluarea datelor și salvarea lor într-un format potrivit, se vor antrena doi arbori de decizie cu funcțiile *gini\_train* și *entropy\_train.* Adâncimea maximă a acestor arbori a fost impusă la 3 niveluri, cu condiția suplimentară de a avea minim 5 obiecte în fiecare frunză. Astfel se evită situațiile de supraclasificare, în care fiecare obiect ar avea propria lui clasă, respectiv de subclasificare, în care clasele ar fi prea generale pentru a oferi informația necesară.

Funcția *prediction* este utilizată pentru a clasifica seturile de date de test în funcție de arborii de decizie creați anterior. În final, funcția *accuracy* returnează informații utile despre precizia fiecărui arbore: matricea de confuzie (o matrice de n\*n clase, unde pe diagonala principală sunt clasificările corecte, iar celelalte valori sunt erori de clasificare), precizia clasificării în procente, și un raport de clasificare care conține alte informații utile.

Majoritatea funcțiilor au fost implementate cu ajutorul funcțiilor prestabilite din biblioteca sklearn.

La final, datele de ieșire vor fi scrise într-un fișier, *output.txt*

**Bibliografie:**

<https://archive.ics.uci.edu/ml/machine-learning-databases/wine/> <https://archive.ics.uci.edu/ml/machine-learning-databases/balance-scale/>

<https://archive.ics.uci.edu/ml/machine-learning-databases/>

<https://medium.com/machine-learning-101/chapter-3-decision-trees-theory-e7398adac567>

<https://medium.com/machine-learning-101/chapter-3-decision-tree-classifier-coding-ae7df4284e99>

<http://dataaspirant.com/2017/02/01/decision-tree-algorithm-python-with-scikit-learn/>

<https://www.geeksforgeeks.org/decision-tree-implementation-python/>

<https://www.quora.com/What-is-the-interpretation-and-intuitive-explanation-of-Gini-impurity-in-decision-trees/answer/Duane-Rich>

<http://www.bel.utcluj.ro/dce/didactic/sisd/SISD_seminar_4_RNA.pdf>

<https://scikit-learn.org/>

<https://scikit-learn.org/stable/modules/tree.html>

<https://docs.python.org/3/>

<https://en.wikipedia.org/wiki/F1_score>