Aplicație Practică 1

Cirjontu Ionela

January 4, 2025

1 Descrierea problemei

Scopul aplicației este de a analiza modelele de consum energetic din perioada anterioară, în special din luna noiembrie, pentru a prezice soldul de energie electrică din luna decembrie. Aceste prognoze sunt esențiale pentru furnizorii de energie, care pot astfel să își ajusteze capacitatea de producție și distribuție, pentru a răspunde mai eficient cerințelor din perioada respectivă.

2 Justificarea abordării

Am ales algoritmul de clasificare ID3 datorită simplității și eficienței sale în abordarea problemelor de învățare automată, fiind un algoritm foarte intuitiv și ușor de implementat. ID3 este cunoscut pentru faptul că utilizează măsura entropiei pentru a construi arborele de decizie, alegând, la fiecare pas, atributul care imparte cel mai bine instantele de antrenament. În cazul aplicatiei, entropia este folosită pentru a calcula informația câștigată de la fiecare atribut, iar alegerea celui mai bun atribut la fiecare pas se face pe baza acestei măsuri. Deși algoritmul ID3 este, în mod tradițional, folosit pentru probleme de clasificare, am adaptat acest algoritm pentru o problemă de regresie, având etichete care reprezintă intervale de valori ale soldului, nu clase discrete.

2.1 Soluția 1

În soluția denumită soluția1 am considerat toate coloanele (Consum[MW], Medie Consum[MW], Producție[MW], Carbune[MW], Hidrocarburi[MW], Ape[MW], Nuclear[MW], Eolian[MW], Foto[MW], Biomasa[MW]) ca atribute de intrare continue, iar coloana Sold[MW] este eticheta codificată în cod cu -1, 0, 1, 2, cu semnificația: -1 sold în [0, 200), 0 sold în [200, 800), 1 sold în [800, 1400), iar 2 pentru sold mai mare de 1400. Am ales aceste intervale deoarece majoritatea valorilor se concentrează în 200-1400.

2.2 Soluția 2

Soluția denumită soluția2. Deși rata de eroare la validare pentru soluția anterioară este destul de mică, intervalele de solduri cuprind prea multe valori, iar predictia nu este destul de satisfăcătoare. Adițional soluției anterioare am creat 3 subarbori, astfel: pentru eticheta anterioară 0 am creat alte 3 intervale: [200, 400), [400, 600), [600, 800); pentru 1 am creat: [800, 1000), [1000, 1200), [1200, 1400); pentru 2 am creat: [1400, 1600), [1600, 1800), [1800, 2000). Fiecare subarbore folosește toate atributele de intrare, dar modelul este antrenat doar cu acele instanțe din setul de antrenament care corespund etichetei principale corespunzătoare.

2.3 Soluția 3

Soluția denumită soluția3. Am încercat evitarea overfitting-ului. Crearea arborilor folosind toate atributele ar fi învățat perfect setul de antrenament, însă pentru setul de validare nu ar fi fost relevante toate atributele. Încercând să folosesc mai puține atribute, rezultatul este un arbore mai simplu care surprinde mai mult dependențele generale și semnificative între atributele de intrare și eticheta. Pentru a îmbunătăți performanța soluției anterioare am testat crearea arborilor de decizie doar în funcție de

anumite atribute. Am făcut prin încercări, iar la secțiunea rezultate voi prezenta rata de eroare pentru fiecare încercare.

3 Prezentarea rezultatelor

3.1 Soluția 1

Am antrenat modelul cu datele din luna noiembrie a aceluiasi an deoarece condițiile climatice ar trebui să fie destul de asemănătoare și datele fiind expuse pe ore dispunem de multe date de antrenament. Am testat cu datele din luna decembrie. Rata de eroare la validare este de 0.1562 (clasifică corect 3807/4512 instanțe). Rata nu este foarte mare deoarece intervalele sunt destul de vaste. Acest aspect facilitează ignorarea zgomotelor, însă nu oferă cu precizie un rezultat satisfăcător.

3.2 Soluția 2

Am antrenat modelul cu același set ca în soluția precedentă. De asemenea, am testat cu același set de validare, dar de această dată am obținut o eroare la validare de 0.5217 (clasifică corect 2158/4512 instanțe). Am restrâns intervalele, deci rezultatul este mai exact decât la soluția anterioară, însă se realizează overfitting pe subarbori deoarece folosim toate atributele de intrare, fără să facem o selecție a celor mai relevante.

3.3 Soluția 3

Am antrenat și validat modelul cu aceleași seturi ca la soluțiile anterioare.Pentru a identifica atributele cele mai relevante in raport cu setul de validare, am testat algoritmul implementat anterior pentru mai multe seturi de atribute. Cele mai semnificative rezultate:

Atribut	Rata eroare
1,2,3	0.4313
2,3,4	0.6017
3,4,5	0.9003
3,5,6	0.8548
1,3,4	0.4333
1,3,4,5	0.4439
1,3,5	0.4082
1,2,3,5	0.4614
2,3,5	0.6095
2,6,7	0.8648
1,3	0.4003

În urma testelor făcute pe diferite configurații de atribute, am observat că cel mai important în raport cu setul de validare este coloana de Consum. Inițial, am încercat să extind varianta 1,3,5, însă dacă mai adăugăm atribute, rata de eroare la validare era din ce în ce mai mare. De asemenea am observat că eliminarea atributului Consum duce la o creștere mult mai mare a erorii. După mai multe încercări, cele mai bune atribute s-au dovedit a fi Consum și Producție, ceea ce ar fi fost de zis și intuitiv, fiind aspectele care descriu cel mai bine problema.

4 Concluzii

Soluția 1 oferă o abordare simplă, dar cu o precizie limitată din cauza intervalelor mari.Soluția 2 aduce o îmbunătățire în precizie, dar suferă din cauza overfitting-ului și a selecției incomplete a atributelor relevante.Soluția 3 este cea mai eficientă abordare, combinând selecția atributelor relevante și evitarea overfitting-ului pentru a prezice mai bine soldul de energie.
În soluțiile propuse pentru prezicerea soldului de energie electrică, am învățat că alegerea corectă a intervalelor și a caracteristicilor importante este esențială pentru ca modelul sa ofere un rezultat cu o acuratete mai mare. Dacă intervalele sunt prea mari, precizia scade, iar dacă sunt prea mici, modelul poate învăța prea multe detalii care nu sunt utile. Atributele precum Consum și Producție sunt cele

mai importante pentru prezicerea soldului, iar folosirea tuturor caracteristicilor disponibile poate face modelul prea complicat. În final, un model simplu poate da rezultate mai bune decât unul prea specializat.În concluzie, un model care folosește intervale corecte și caracteristici relevante, evitând în același timp complexitatea inutilă, va oferi cele mai bune predicții.
Pentru îmbunătățirea performanței algoritmului s-ar putea crea criterii separate de alegere a atributelor pentru arborele principal și cei secundari, crescând relevanța acestora. De asemenea, se pot aplica diferite metode de pruning, simplificând arborii.