Universitatea Politehnică Timișoara
Calculatoare și tehnologia informației, ANUL 3

Spătăcean Oana- Teodora
Stoichescu Iulia

Proiect Explorarea Datelor
Data Mining
Analiza setului de date - Dermatology

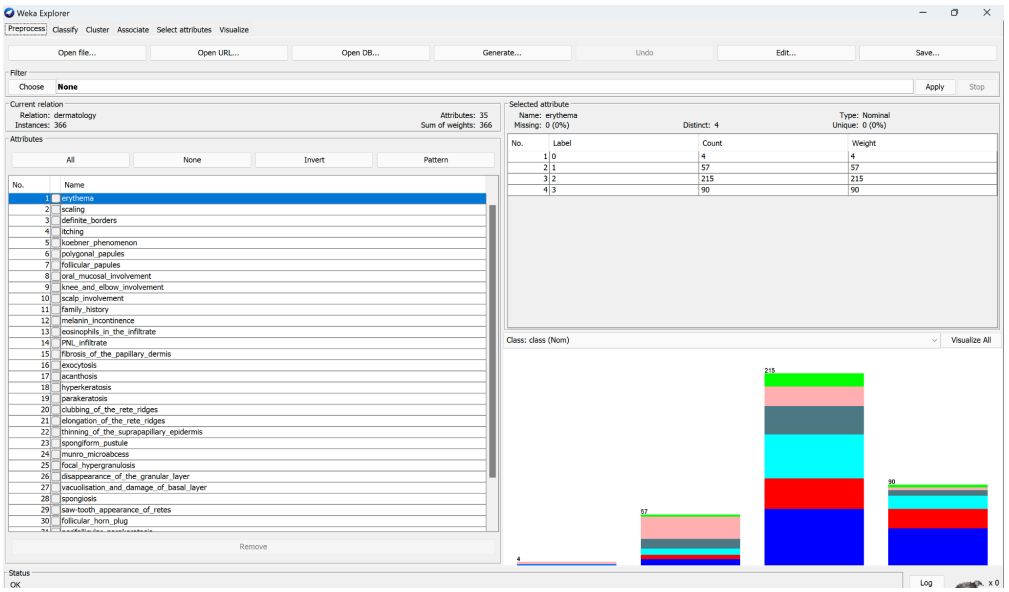
1. Descrierea setului de date

Am selectat un set de date denumit Dermatology, care furnizează informații esențiale despre diverse afecțiuni ale pielii. Acest set de date conține 34 de caracteristici, dintre care 33 sunt valori numerice continue, iar una este nominală. Numărul total de instanțe este de 366.

În cadrul acestui set de date, caracteristica de istoric familial este reprezentată printr-o valoare de 1 dacă afecțiuni cutanate au fost observate în familie și de 0 în caz contrar. Vârsta pacientului este înregistrată direct ca un număr întreg.

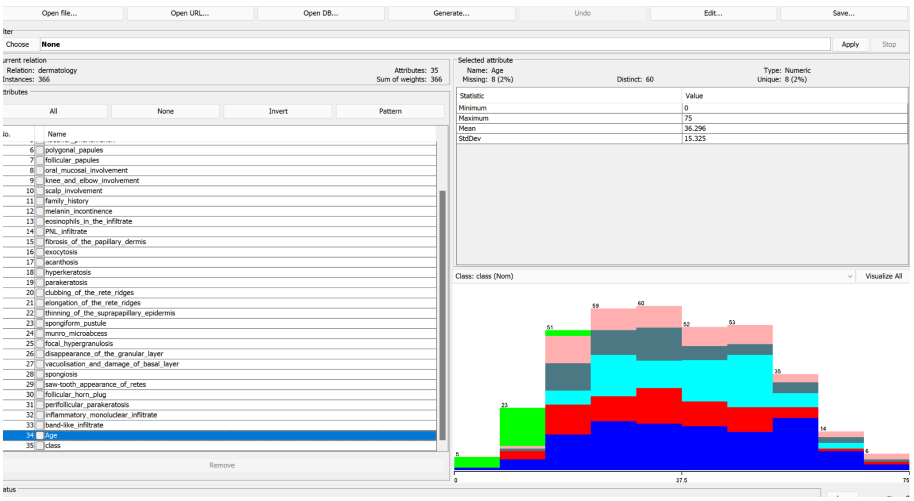
Celelalte caracteristici, fie clinice, fie histopatologice, sunt evaluate pe o scară de la 0 la 3. Valorile acestei scale indică absența caracteristicii (0), prezența caracteristicii într-o măsură scăzută (1 sau 2), sau prezența caracteristicii într-o măsură semnificativă (3).

Pentru a facilita analiza, am transformat inițial setul de date din formatul său original în format .csv folosind Microsoft Office Excel. Acest format este recunoscut automat de către Weka, însă pentru a utiliza setul de date în Weka, acesta va fi convertit în formatul arff (Attribute-Relation File Format) prin intermediul pachetului Weka.

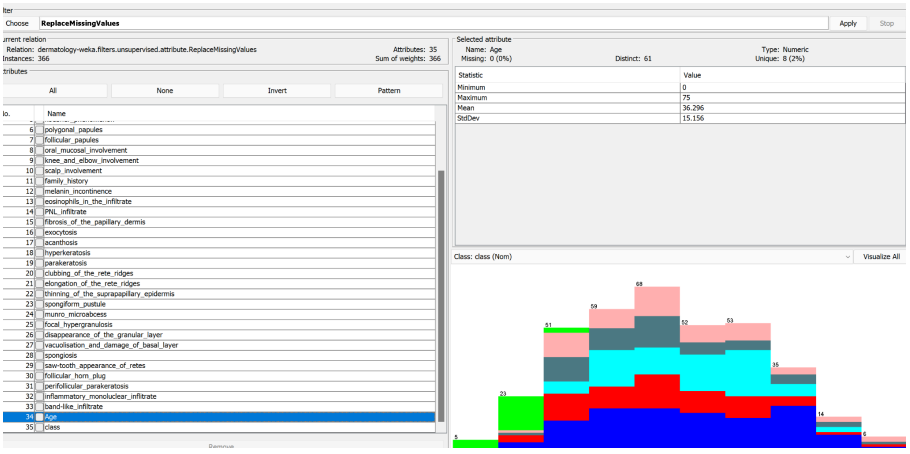


2. Analiza setului de atribute

În cadrul setului de date selectat, am identificat prezența unor valori lipsă, pe care trebuie să le înlocuim corespunzător.

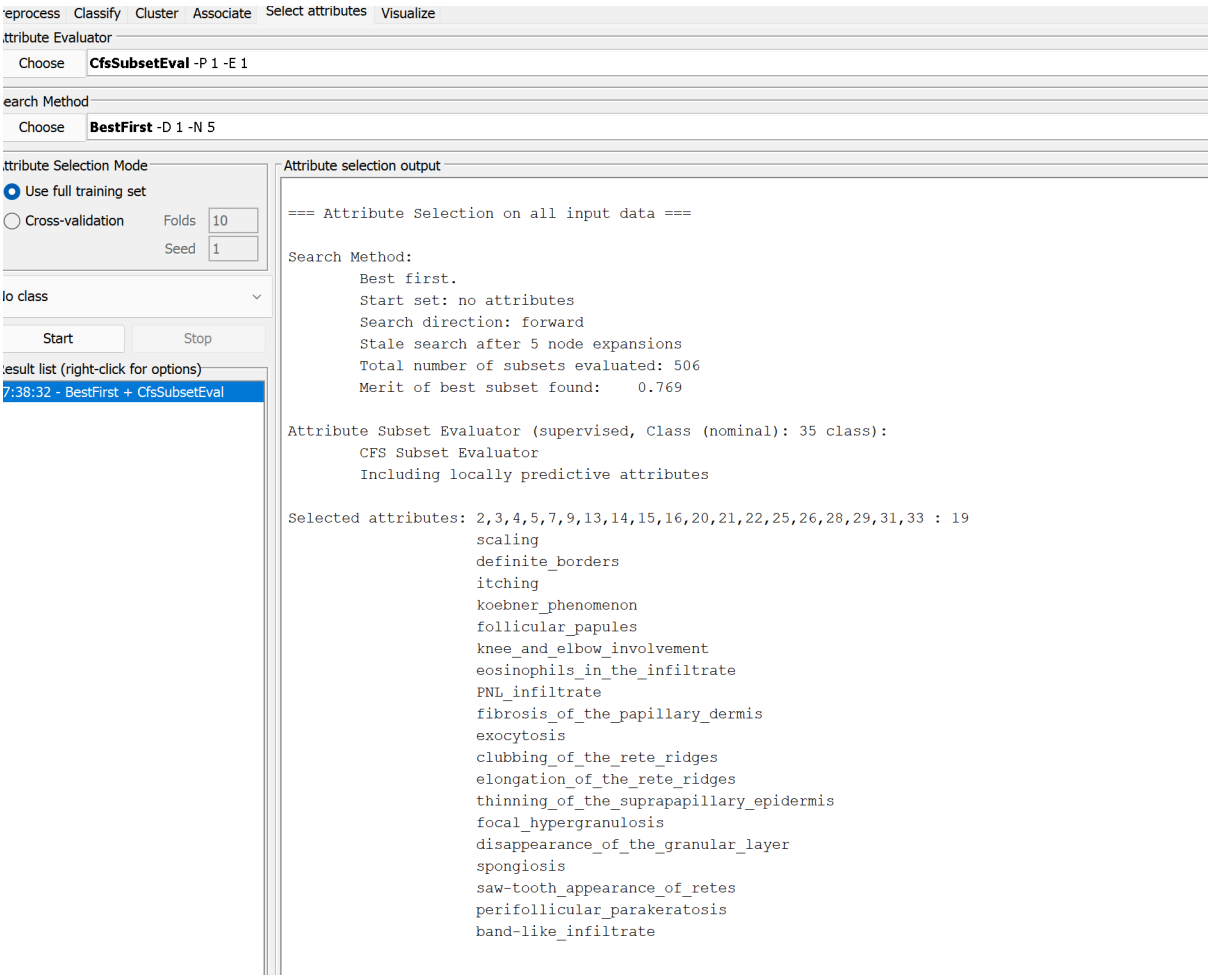


Pentru aceasta, vom utiliza opțiunea "ReplaceMissingValues" din cadrul filtrului disponibil în Weka (Choose -> filters -> unsupervised -> attribute -> ReplaceMissingValues). După aplicarea acestui filtru, toate valorile lipsă vor fi înlocuite cu valoarea medie sau moda corespunzătoare atributului în cauză.



Rezultatul va fi salvat într-un nou fișier numit "dermatologyReplacedMissingValues.arff", care va fi utilizat în continuare pentru analiză.

Următorul pas constă în identificarea celor mai importante atribute din setul de date. Aceasta se va realiza folosind funcționalitatea "Select attributes". După selectarea metodei de căutare și a evaluatorului potrivit, vom iniția procesul. Rezultatele obținute vor evidenția cele 19 cele mai importante atribute, plus atributul "class", totalizând 20 de atribute relevante.



În continuare, vom salva acest nou set de date, eliminând toate celelalte atribute care nu se află printre cele mai importante. Pentru aceasta, vom utiliza metoda "BestFirst" pentru căutare, iar opțiunea "CvsSubsetEval" pentru evaluarea valorii fiecărui subset de atribute, luând în considerare atât capacitatea lor individuală de predictibilitate, cât și gradul de redundanță între ele.

Trebuie să notăm că trei dintre cele mai importante atribute identificate sunt: "Scalling", "Define_borders" și "Itching". Acestea sunt toate atribute nominale care fac parte din categoria "Clinical Attributes", având valori în intervalul de la 0 la 3 (dacă nu este altfel specificat).

3. Analiza datelor

Tehnică:

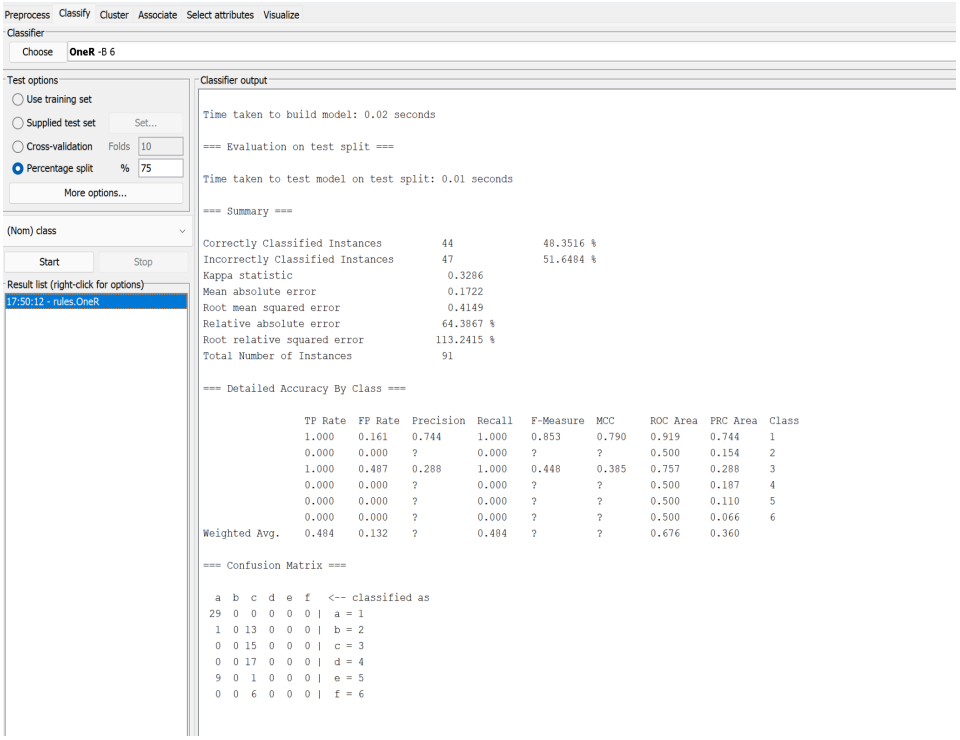
- **Clasificarea**

Clasificarea este o tehnică predictivă folosită în învățarea supervizată, unde se prezic clasele pentru instanțele noi pe baza unor predictorii. Printre metodele comune de clasificare se numără algoritmul OneR și arborii de decizie, rețelele neuronale sau algoritmii genetici.

OneR

Algoritmul OneR este o metodă simplă care creează o singură regulă de clasificare bazată pe o singură trăsătură. Alegerea trăsăturii se face pe baza maximizării probabilității de clasificare. Pentru fiecare atribut, se construiește un clasificator separat, rezultând în mai multe clasificatoare, iar cel mai semnificativ este ales.

În practică, pentru a aplica algoritmul OneR în Weka, se folosește opțiunea Percentage Split pentru a diviza setul de date într-un set de antrenare și unul de testare (Classify -> Choose -> classifiers -> rules -> OneR). De obicei, se alege un procentaj mai mare pentru setul de antrenare pentru a asigura o cantitate suficientă de date pentru învățare. Noi am ales 75%.



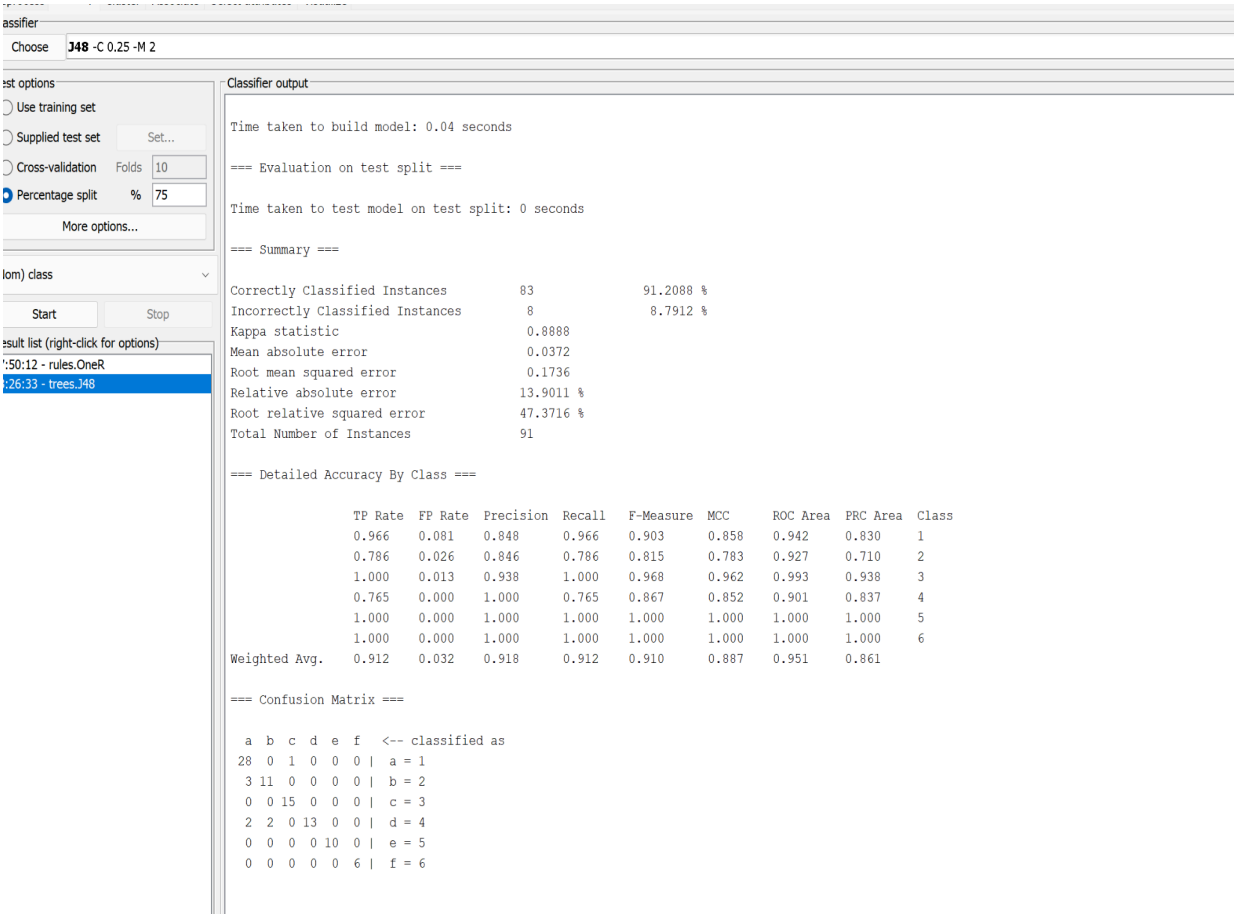
Astfel, se obține o regulă de clasificare bazată pe o trăsătură, iar în urma testării, se evaluează performanța acesteia. Într-un exemplu, clasificarea se face în funcție de atributul "elongation_of_the_rete_ridges". Acuratețea clasificării, fiind in cazul nostru 48,3516%, și statistica Kappa, fiind in cazul nostru 0,3286 (nu foarte apropiata de valoarea 1, valoarea 1 semnificand acord total intre clasificarea realizata prin aceasta tehnica si valorile observate), sunt utilizate pentru a evalua cât de bine se potrivește modelul datelor și cât de apropiată este clasificarea de cea corectă. 47 de instanțe sunt clasificate greșit dintr-un total de 91 de instanțe din setul de test.

Matricea de confuzie oferă o comparație între clasificarea făcută de model și valorile reale. Valorile de pe coloane reprezintă predicțiile modelului, în timp ce cele de pe rânduri sunt valorile reale.

J48

Algoritmul C4.5, implementat în Weka sub numele de J48, este unul dintre cei mai utilizați algoritmi de inducție a arborilor de decizie. Este o extensie a algoritmului ID3, abordând probleme precum supra-antrenarea, gestionarea datelor continue și a celor cu valori lipsă, și îmbunătățirea eficienței computaționale. Acest algoritm generează un arbore de decizie prin împărțirea recursivă a setului de date, folosind o strategie de parcurgere în adâncime.

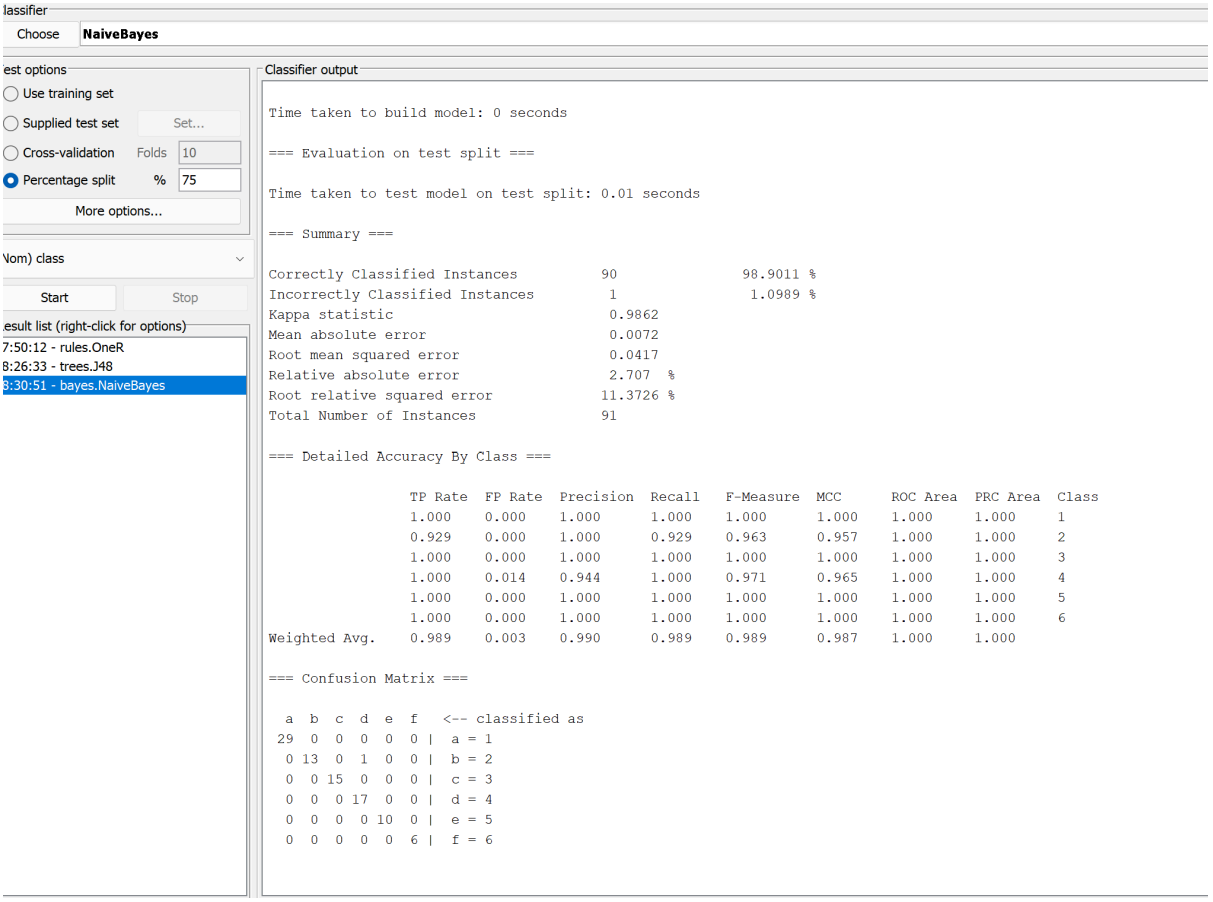
În cazul nostru, am aplicat algoritmul J48 în Weka, utilizând opțiunea "Precentage Split". Astfel, am construit regulile de clasificare pe baza a 75% din setul nostru de date, restul fiind rezervat pentru testare. Am obținut un arbore de decizie cu 40 de noduri și 30 de frunze. Acuratețea clasificării a fost de 91,2088%, cu doar 8 din cele 91 de instanțe fiind clasificate greșit, iar 83 corect. Statistica Kappa a fost de 0,8888, apropiindu-se de valoarea 1, ceea ce indică un grad ridicat de acord între clasificarea obținută și valorile observate.



Naive Bayes

Se utilizează pentru a clasifica date neetichetate prin estimări, bazându-se pe datele de antrenament etichetate. Este un algoritm generativ bazat pe ipoteza "naivă" că predictele condiționate de apartenența la o clasă sunt independente între ele.

În cazul nostru, am aplicat algoritmul Naive Bayes în Weka, selectând opțiunea "Percentage Split". Astfel, regulile de clasificare au fost construite pe baza a 75% din setul nostru de date, iar restul instanțelor au fost folosite pentru testare. Acuratețea clasificării a fost de 98,9011%. Statistica Kappa a fost de 0.9862, indicând un acord între clasificarea obținută și valorile observate.

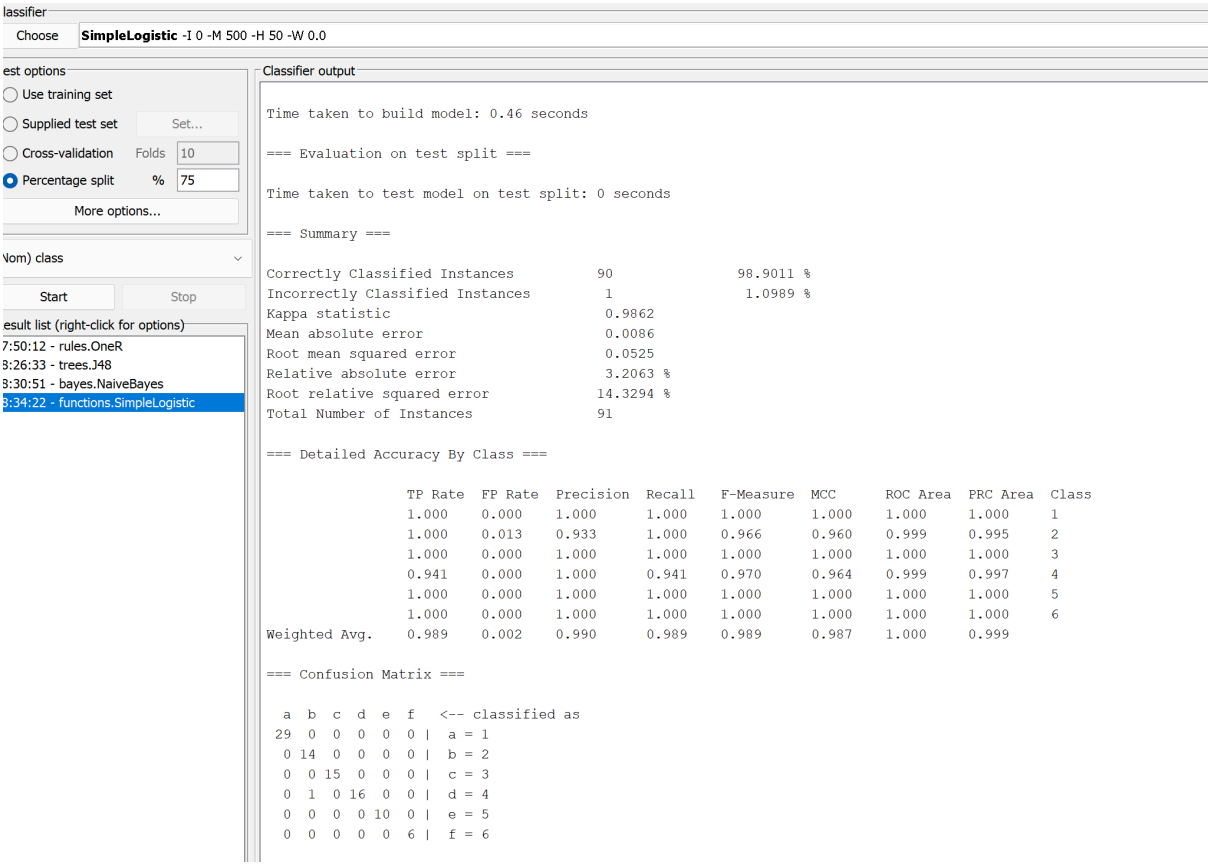


Regresia logistică

Regresia logistică este utilizată pentru a investiga relația dintre multiple variabile independente și o variabilă dependentă binară. Această variabilă dependentă se referă adesea la apartenența la două categorii, cum ar fi prezența/absența sau da/nu.

În Weka, pentru a modela o variabilă binară folosind regresia logistică, selectăm opțiunea "SimpleLogistic" din panoul "Classify". De asemenea, bifăm opțiunea "Percentage split" pentru a împărți setul de date într-un set de antrenare și unul de testare, folosind 75% pentru antrenare și 25% pentru testare.

În rezultatele obținute, observăm o acuratețe de clasificare de 98,9011%. Din cele 91 de instanțe, 90 au fost clasificate corect, iar 1 a fost clasificata greșit. Valoarea indicelui Kappa statistics este de 0,9862, apropiindu-se de valoarea maximă de 1, ceea ce indică o concordanță excelentă între clasificarea realizată prin regresia logistică și valorile observate.



- **Gruparea**

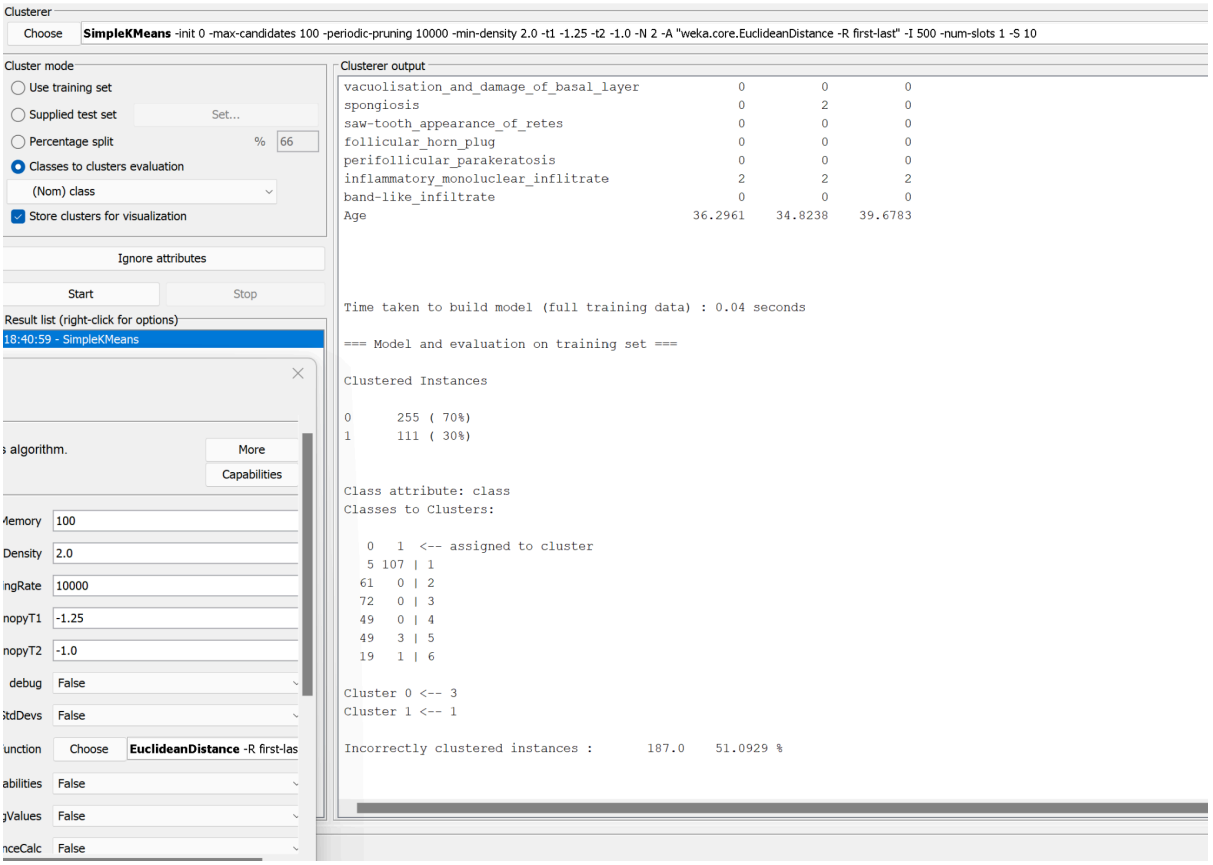
Gruparea, cunoscută și sub numele de analiză de tip cluster, este o tehnică utilizată pentru a împărți un set de date în clase sau grupuri, fără a avea clase prestabilite. Fiecare clasă este definită de un model, care poate fi un obiect abstract sau un exemplu reprezentativ al clasei.

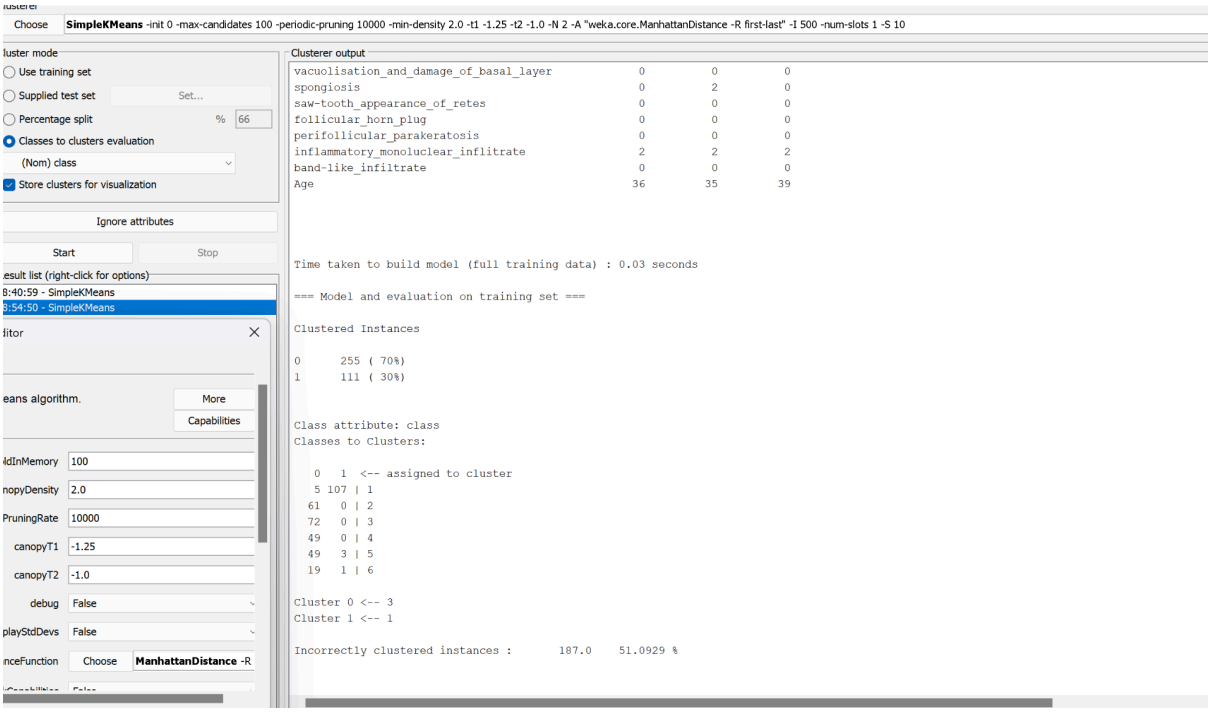
Simple Kmeans

Algoritmul SimpleKMeans este unul dintre algoritmii utilizați în gruparea datelor. Acesta începe prin crearea unei partiționări inițiale a datelor și apoi încearcă să îmbunătățească această partiționare prin mutarea iterativă a obiectelor dintr-un grup în altul. Procesul se oprește atunci când se reușește împărțirea datelor în k grupuri, astfel încât fiecare grup să conțină cel puțin un obiect și fiecare obiect să aparțină unui singur grup.

Pentru a folosi algoritmul SimpleKMeans în Weka, trebuie să accesăm opțiunea "Cluster" și să selectăm "SimpleKMeans" din lista de opțiuni disponibile. Înainte de a începe procesul, bifăm opțiunea "Classes to clusters evaluation". De asemenea, putem alege distanța de evaluare a similarității între obiecte, cum ar fi distanța euclidiană sau Manhattan.

În rezultatele obținute, observăm instantele incorecte de clustere sunt 51,0929%, iar algoritmul a efectuat 4 iterații. Interesant este că valorile obținute pentru distanța euclidiană sunt similare cu cele pentru distanța Manhattan. Precizia datelor este de 48,90% (100% - (187 / 366) * 100 = 48.09%).

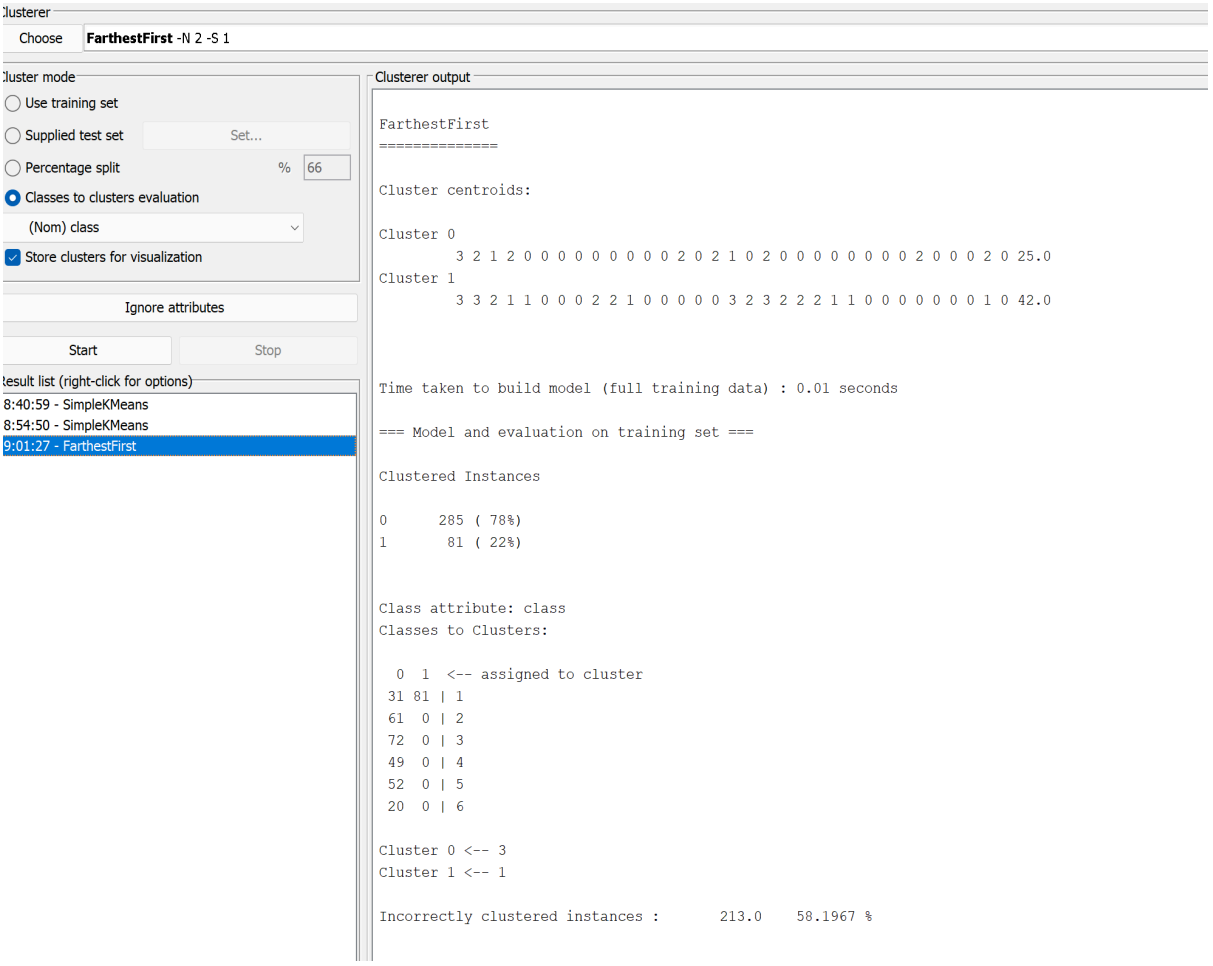




Farthest First

Algoritmul Furthest First este o altă metodă de grupare în care clasificarea unei noi observații nu se face în funcție de valoarea medie a clasei, ci prin selectarea instanței reale cea mai aproape de medie. Această abordare permite luarea deciziei de clasificare în funcție de o observație concretă, ceea ce poate conduce la clustere ce reflectă datele mai realistic.

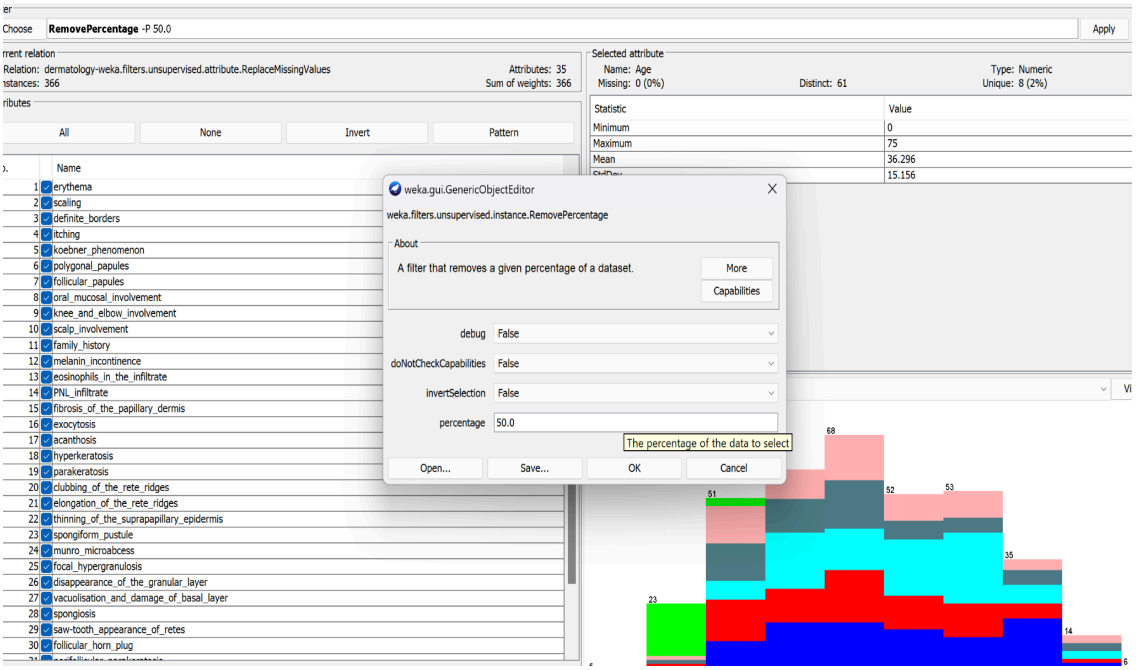
În rezultatele obținute, se observă o acuratețe a setului de date de 42,62% (100% - (213 / 366) * 100 = 42.62%). Comparativ cu algoritmul SimpleKMeans, această tehnică de grupare produce rezultate mai slabe cu câteva procente. Este important de menționat că performanța algoritmilor de grupare poate varia în funcție de natura și distribuția datelor, precum și de parametrii specifici utilizați în algoritm.



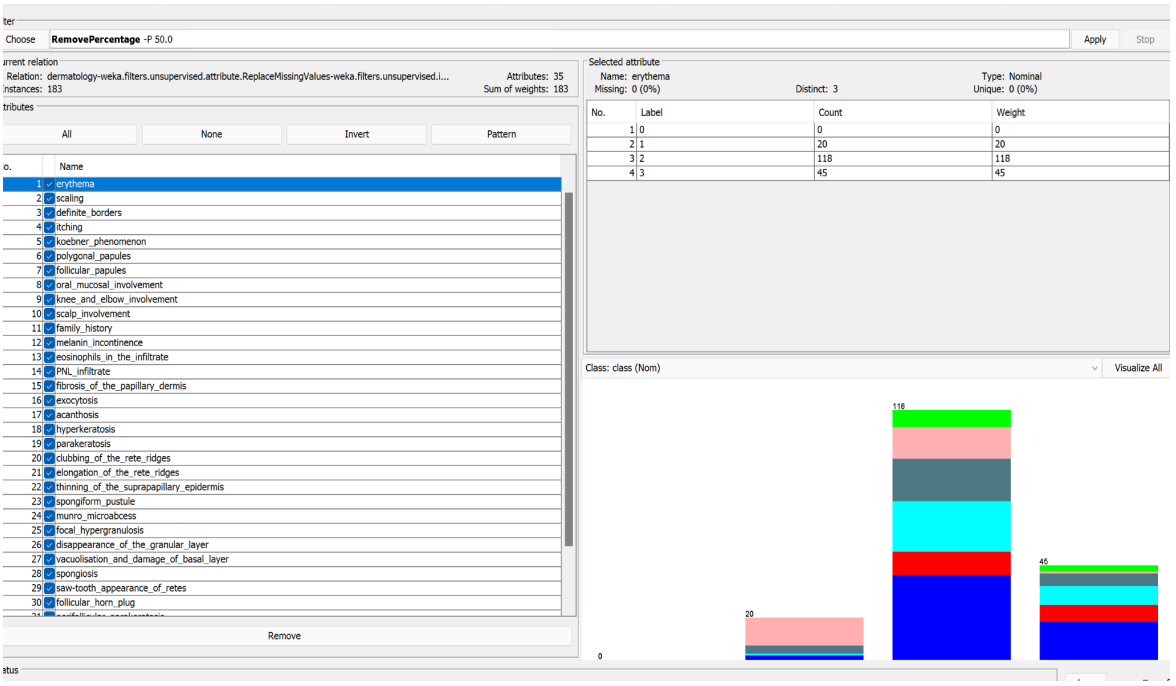
4. Obtinere de noi cunostiinte

Pentru a obține noi cunoștințe din setul de date original, am împărțit inițial setul de date într-un set de training și un set de testare. Am folosit o tehnică de grupare pentru a extrage aceste noi cunoștințe.

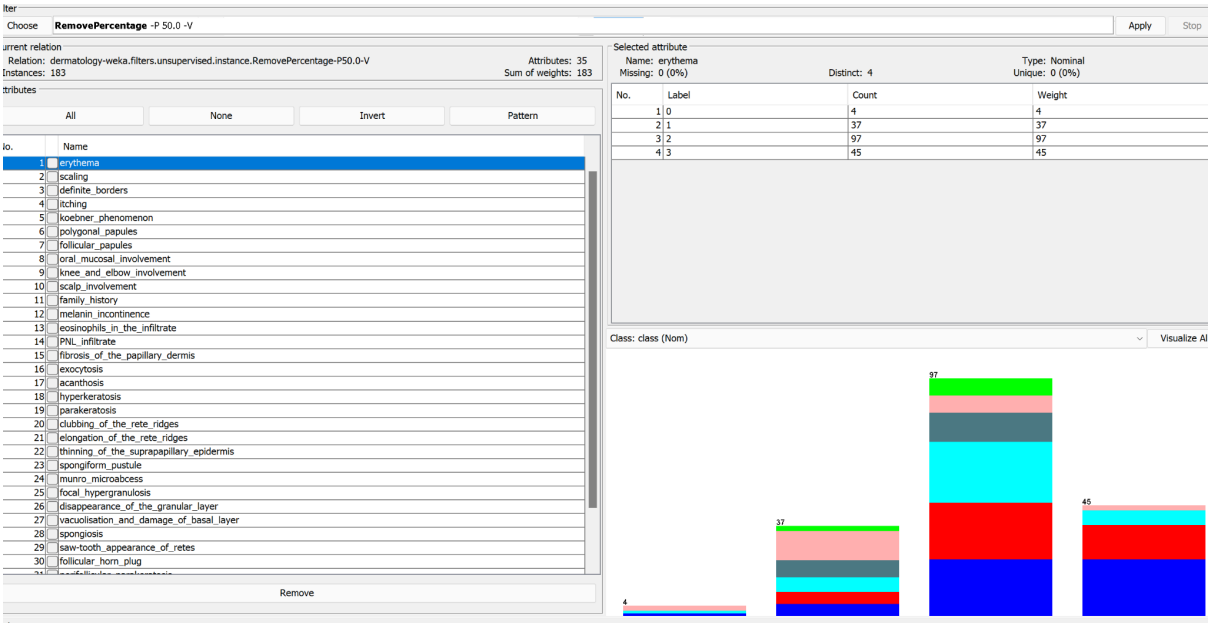
Procesul de împărțire a setului de date a implicat utilizarea opțiunii "RemovePercentage" din meniul de preprocesare în Weka, unde am setat un procentaj de 50% și am aplicat această modificare.



Setul de date rezultat a fost salvat sub denumirea "dermatologyTrainSet.arff".



Apoi, am revenit la starea inițială a setului de date folosind funcția "undo" și am repetat procedura, dar de data aceasta am setat opțiunea "invertSelection" la true. Astfel, am obținut setul de date de test.



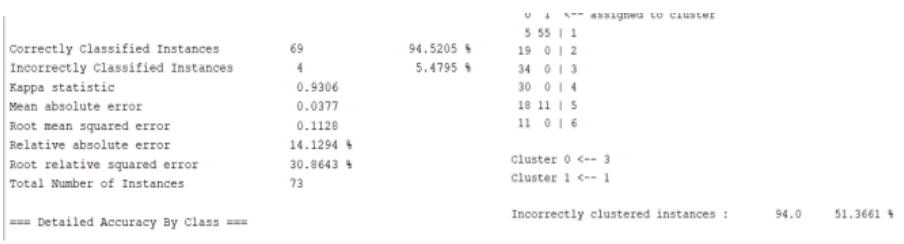
Setul de date de test este similar celui original sau celui de training, având aceeași dimensiune. Singura diferență constă în faptul că valorile atributului clasă au fost înlocuite cu "?", reprezentând datele necunoscute.

```
@attribute focal_hypergranulosis {0,1,2,3}
@attribute disappearance_of_the_granular_layer {0,1,2,3
@attribute spongiosis {0,1,2,3}
@attribute saw-tooth_appearance_of_retes {0,1,2,3}
@attribute perifollicular_parakeratosis {0,1,2,3}
@attribute band-like_infiltrate {0,1,2,3}
@attribute class {1,2,3,4,5,6}

@data
2,0,3,0,0,1,0,0,0,3,0,0,0,0,0,3,0,0,0,?
3,3,2,1,0,1,0,1,0,1,2,2,2,0,0,0,0,0,0,?
1,2,3,1,0,0,0,0,0,0,1,0,0,0,2,0,3,2,0,3,?
2,2,0,0,0,3,0,3,0,0,2,2,2,0,3,0,0,0,0,?
3,2,2,2,0,0,0,0,0,1,0,0,0,2,2,2,3,0,3,?
3,2,0,0,0,0,0,2,1,0,2,0,0,0,0,0,2,0,0,0,?
1,0,2,0,0,0,0,0,0,3,1,0,2,0,0,0,0,0,0,0,?
2,3,3,3,0,0,0,0,0,0,2,0,0,0,0,2,3,2,0,3,?
```

Pentru a aplica setul de date de test în Weka, am urmat următorii pași:

1. Am ales o tehnică de grupare, în acest caz am selectat FarthestFirst.
2. Am selectat opțiunea "Supplied test set" și am încărcat setul de date de test.
3. Am reevaluat gruparea inițială folosind setul de date de test.
4. Am obținut rezultatele analizei.



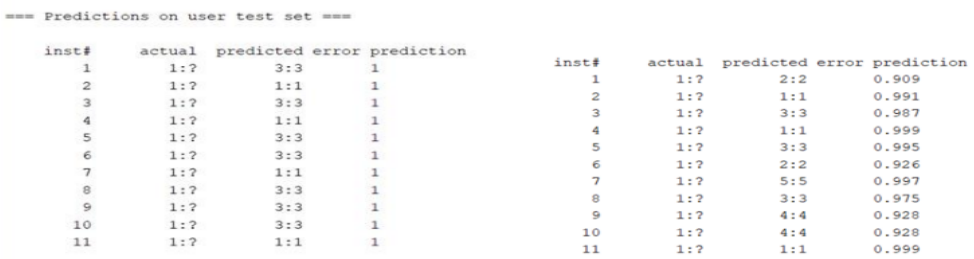
Este important să remarcăm că setul de date de test a fost utilizat pentru a evalua performanța modelului de grupare pe date noi, neetichetate anterior. Această evaluare ne permite să înțelegem cât de bine generalizează modelul nostru pe datele pe care nu le-a văzut anterior.

Graficul acestui tip de grupare este:

Predictii:

Pentru a face predicții, am utilizat fișierul de test și am reaplicat evaluarea setului de date pentru a observa diferențele și rezultatele.

- **OneR:** Toate instanțele sunt prezise cu o șansă de 100% ca fiind negative.
- **J48**: Există o tendință de a prezice anumite clase datorită structurii arborelui de decizie. Acest lucru se datorează probabil prezenței unor caracteristici semnificative care sunt asociate puternic cu anumite clase dermatologice. De exemplu, anumite atributui precum "melanin incontinence" sau "eosinophils in the infiltrate" pot avea o influență semnificativă asupra predicțiilor clasificatorului.
- **NaiveBayes:** Predicțiile diferă în funcție de instanță, însă majoritatea sunt apropiate de 1, indicând o probabilitate mare de a fi negative. Am creat un model Naive Bayes pentru fiecare pereche posibilă de clase. De exemplu, avem un model pentru a distinge între psoriazis și dermatită seboreică, altul pentru a distinge între psoriazis și lichen plan, și tot așa, până când acoperim toate posibilele perechi de clase. Pentru fiecare pereche de clase, am antrenat modelul Naive Bayes folosind doar datele corespunzătoare celor două clase. Aceasta înseamnă că am creat subseturi de date care conțin doar exemplele pentru cele două clase respective. După antrenarea modelelor pentru fiecare pereche de clase, vom clasifica un nou exemplu prin aplicarea tuturor modelelor antrenate și votarea pentru clasa care primește cele mai multe voturi.



```
=== Predictions on user test set ===

inst#     actual  predicted error prediction
    1        1:?        3:3       1
    2        1:?        1:1       1
    3        1:?        3:3       1
    4        1:?        1:1       1
    5        1:?        3:3       1
    6        1:?        3:3       1
    7        1:?        1:1       1
    8        1:?        3:3       1
    9        1:?        3:3       1
   10        1:?        3:3       1
   11        1:?        1:1       1
```

```
inst#     actual  predicted error prediction
    1        1:?        2:2       0.909
    2        1:?        1:1       0.991
    3        1:?        3:3       0.987
    4        1:?        1:1       0.999
    5        1:?        3:3       0.995
    6        1:?        2:2       0.926
    7        1:?        5:5       0.997
    8        1:?        3:3       0.975
    9        1:?        4:4       0.928
   10        1:?        4:4       0.928
   11        1:?        1:1       0.999
```

Concluzii:

- Setul de date este extrem de complex, cu un număr mare de instanțe și date. Folosirea unor algoritmi simpli, cum ar fi OneR, duce la o clasificare cu un procentaj mai mare de instanțe incorect clasificate.
- Surprinzător, clasificatorul NaiveBayes a avut o precizie de 98,9011%, apropiata de 100%, indicând că este cel mai potrivit pentru acest set de date, cu 1 instanța clasificata incorect.
- Regresia liniară a avut o acuratețe de 98,9011%, egala cu cea de la NaiveBayes. J48 a avut o acuratețe de 91,2088%.
- Algoritmii mai complexi, precum NaiveBayes, Regresia Liniară și J48, sunt mai buni în procesarea unor astfel de date.
- Problema se repetă și în cazul tehnicilor de grupare, unde numărul mare de atribute duce la o eroare ridicată. Pentru SimpleKMeans, acuratețea este de 48,90%, iar pentru FarthestFirst este de doar 42,62%, ambele cu erori considerabile.
- Impactul numărului mare de caracteristici în comparație cu dimensiunea setului de date este evident în performanța algoritmilor. Acest aspect conduce la variații semnificative în ritmul de învățare între diversele tehnici. De exemplu, Naive Bayes se evidențiază prin viteză și precizie.
- Metodele de clusterizare nu ating nivelul de precizie dorit pe acest set de date, datorită numărului mare de caracteristici implicate. Interpretarea rezultatelor este puternic influențată de aceste caracteristici, ceea ce poate genera clusterizări extrem de polarizate. Această problemă devine evidentă în cazul abordării FarthestFirst.

Bibliografie:

http://weka.sourceforge.net/doc.dev/weka/classifiers/bayes/NaiveBayes.html

https://www.saedsayad.com/oner.html

http://inf.ucv.ro/documents/rstoean/7.%20Evaluarea%20performantei.pdf

http://weka.sourceforge.net/doc.dev/weka/classifiers/trees/J48.html

http://ionut.mironica.ro/teaching/TACAI_lab1.pdf

Laboratoarele facute in cadrul semestrului, insotite de lucrarile de laborator