**Analiza si proiectare aplicatie**

**Scop**: personalizarea preferintelor clientilor si obtinerea de oferte personalizate pentru acestia

1. Pe scurt aplicatia trebuie sa:

* permita unui client sa introduca valorile pentru atributele unui produs ( sau sa fie preluate dintr-un fisier text). Generic le voi numi Criteriu 01, Criteriu02, ..., Criteriu N. Acestea pot fi preluate dintr-o baza de date in momentul in care se deschide interfata, clientului ramanad doar sa completeze valorile pentru criterii ( valorile vor fi numerice). Clasa/obiectul pentru client se va numi ceva de genul *AgentClientXXX* , unde XXX va reprezenta numarul clientului in momentul in care se deschide interfata pentru el).
* prin intermediul unei alte interfete care va reprezenta furnizorul produsului/serviciului si va fi doar unul singur. Acesta va analiza informatia trimisa (completata de clienti sau preluata dintr-un fisier/sau basza de date) si va decide ce oferte sa creeze pentru clienti folosind algoritmul de clusterizare SimpleKMeans. Numarul de oferte va fi egal cu numarul de profile de clienti. Acest numar va fi dat ca parametru ( a se vedea sectiunea de mai jos pentru detalieri). Clasa/obiectul afurnizorului se va numi ceva de genul *AgentFurnizor.* Ofertele vor fi trimise clientilor pe baza alocarii acestora intr-un grup.

**Aplicatia descrisa cum o vad eu**.

Daca aveti sugestii, sunt binevenite atat la functionalitati cat si la interfete. Ca forma de implementare nu ar fi nicio restrictie decat cele mentionate pentru denumirile de clase/obiecte/formuri si sa fie de preferat in Java (sau C#).

1. se deschide o interfata principala care va permite deschiderea interfetei pentru clienti si deschiderea interfetei pentru agentul furnizor (Figura 1). Cand se va deschide va fi pozitionata in centrul ecranului. Aceasta fereasta va avea optiunile:

* optiuni (butoane) *DA* si *STOP* pentru client (descrise la punctul a).
* si “*Deschide interfata pentru furnizorul de produse/servicii*” (buton) pentru furnizor – cand va fi apasat va deschide interfata pentru furnizor ( descrisa la punctul b).



Figura 1. Interfata principala

1. **Interfata Agent Client**

De cate ori se apasa DA, se va deschide o fereastra pentru un client , fereatra pozitionata in partea stanga a ecranului (vezi Figura 2). Titlul formului sa fie *AgentClientXXX* (XXX – numarul interfetei deschise pentru client si va reprezenta numarul clientului folosit la sfarsit pentru trimiterea ofertei personalizate, fiind ID-ul clientului).

**Agent Client\_XXX**

|  |  |
| --- | --- |
| Criteriu 01 (text) | Valoare (numerica/string) |
| Criteriu 02 (text) | Valoare (numerica/string) |
| .... | .... |
| Criteriu n (text) | Valoare (numerica/string) |

Trimite catre furnizor

Oferta primita de la Agentul Furnizor

Numarul ofertei: *numarul profilului (ofertei) care a fost atribuit AgentuluiClient*

|  |  |
| --- | --- |
| Criteriu 01 (text) | Valoare (numerica/string) - de la profil (oferta) |
| Criteriu 02 (text) | Valoare (numerica/string)- de la profil (oferta) |
| .... | .... |
| Criteriu n (text) | Valoare (numerica/string) - de la profil (oferta) |

Figura 2. Interfata AgentClient

Descriere interfata:

-*AgentClientXXX* – nume form

-**Criteriu 01, 02...n** - lista cu criterii pe care le introduce clientul (preluat din baza de date, in baza de date ar fi bine sa aiba ca si camp si un numar de ordine). Acestea pot fi definite generic exact asa Criteriu 1, Criteriu 2, ... Criteriu 8, putand fi particularizate ulterior. Se vor pune initial 8 criterii in baza de date. As dori ca daca actualizez/sterg/adauga criterii in baza de date, acestea sa fie preluate apoi la rularea aplicatiei cu modificarile efectuate.

-**Valoare numerica** – valoarea data fiecarui criteriu (atribut) de catre client

-**Trimite catre furnizor** – este un buton. Daca este apasat informatiile sunt trimise catre agentul furnizor si va aparea un mesaj “*Preferinta a fost trimisa*” ( fereastra de mesaj va fi inchisa prin apasarea unui buton cu OK, fapt ce va asigura salvarea datelor intr-o baza de date sau fisier). Toate informatiile (de la toti clientii) pot fi stocate intr-o baza de date (de preferat) sau intr-un fisier text. Din locul stocarii vor fi preluate de agentul furnizor. Pe masura ce un client trimite informatiile despre criterii, in interfata AgentuluiFurnizor va exista o zona care va contoriza clientii ( De exemplu, daca clientul 1 a apasat butonul *Trimite spre furnizor* si apoi a fost apasa butonul Ok de la “*Preferinta a fost trimisa*”, zona de *Contor clienti* va indica 1, trimite si al doilea client, Contorul va arata 2, etc).

*//de aici, campurile mentioante se vor completa dupa ce se va face clusterizarea (impartirea in grupe) de catre furnizor.*

-*Oferta primita de la agentul furnizor*: text fix care va aparea pe form

- *Numarul ofertei*: ( text fix), urmat de *numarul oferteri* care a fost atribuita clientului respectiv (voi incerca pe exemplul de la sfarsit sa fie clar acest aspect).

- Criteriu 01, 02...n - criterii preluate din baza de date/fisier

- Valoare 1...n valorile ofertei (profilului) pentru fiecare criteriu.

Daca se apasa STOP, nu va mai fi permisa generarea de interfete pentru clienti, afisandu-se un mesaj in acest sens ( de exemplu: *Toti clientii au fost contactati*).

1. **Interfata agent furnizor**

Pentru a aparea aceasta interfata trebuie apasat butonul numit “*Deschide interfata pentru furnizorul de produse/servicii* ” din fereastra principala. Interfata pentru agentul furnizor va arata ca in figura 3. In masura in care se poate, dar se complica un pic, pot fi mai multi furnizori, iar clientii aleg carui furnizor sa trimita preferintele personalizate.

Agent Furnizor

Preferinte personalizate clienti (*text fix*) **Afiseaza**

**Client Criteriu 1 Criteriu 2 ... Criteriu n**

Client 1 valoare valoare valoare

......

Client n valoare valoare valoare

Numar de oferte create(text fix): **.....**

**Realizarea ofertelor pe baza preferintelor clientilor** (buton)

**Oferta Criteriu 1 Criteriu 2 ... Criteriu n**

Oferta 1 valoare valoare valoare

......

Oferta M valoare valoare valoare

Alocarea oferelor pe clienti (text fix):

**Oferta Client/Clienti**

Oferta 1 Client\_001, Client\_003, Client\_XXX (ca si exemplu)

......

Oferta M Client\_002, Client\_004, Client\_XXX ( ca si exemplu)

**Trimite mesaj catre clienti cu oferta personalizata aferenta** (buton)

Figura 3. Interfata agent furnizor

Descriere interfata:

* Agent furnizaor – titlu form
* *Preferinte personalizate clienti* (text fix) – se va afisa asa in form.
* **Afiseaza**  – buton prin care se preia informatia din bd sau fisier pentru criterii pentru a fi afisata in grid. De fiecare data cand se apasa se va face refresh pe tabelul de dedesubt cu valorile criteriilor preluate.
* **Tabelul**: *Client, Criteriu 1, Criteriu 2, ... Criteriul n* – cap de tabel ( **Nr de clienti > nr criterii**, chiar daca apare **n** la ambele campuri)
  + Client – va reprezenta numele clientului cu ID (de exemplu pentru Clientul 1 va aparea Client\_001).
  + Criteriul 1, 2, ..., n pentru clientul 1 vor reprezenta valorile reluate din BD sau fisier pentru clientul 1.

- *Numar de oferte create*: (text fix in form): - valoarea se va da de la tastatura in ..... si va reprezenta parametru pentru algoritmul SimplekMeans. Acest va fi egal cu numarul de grupe in care vor fi impartiti clientii si care la randul lui va fi egal cu numarul de oferte create (sa zicem M).

- **Realizarea ofertelor pe baza preferintelor clientilor** – buton prin a carui apasare va genera in tabelul de dedesubt cele M oferte mentionate ca parametru la pasul anterior.

- Oferta Criteriu 1.. Criteriu n – cap de tabel – preluate din bd

- exemplu*: Oferta 1 valoare vloare*... reprezinta cea mai reprezentativa oferta pentru clientii din grupul 1 si care va trimisa acestoa. Raportat la exemplu numeric este centroidul Ki’’’’.

- numarul liniilor idn acest tabel este egal cu *Numarul de oferte create*

* Alocarea oferelor pe clienti (text fix): sub aceasta sectiune se va afla un tabel in care vor fi afisate ofertele si clientii carora le-au fost alocate.
* **Trimite mesaj catre clienti cu oferta personalizata aferenta –** este un buton prin a carui apasare sunt trimise ofertele clientilor. Altfel spus, in fereastra clientului se vor completa campurile **Numarul ofertei, Criteriu si Valoare** (unde e marcat cu verde mai sus). Totodata sa apara un mesaj cu un text: “*Ofertele au fost trimise catre clienti*”.

1. Etape si exemplu numeric de cum este folosit algorimtul SimpleKmeans

**Etape:**

* 1.se alege numarul de clustere ( grupe in care vor fi impartite datele/ clientii) – sa zicem N
* 2.se aleg aleator valorile pentru centroizi (centroizii vor fi acele instante /clienti care vor cei mai reprezentativi pentru un grup de clienti) sau pot fi instante din cadrul setului de date
* 3.se calculeaza distantele de la instante (clienti) la centroizii alesi
* 4.se face repartizarea instantelor in in clusterul ( grupa) de al carui centroid sunt mai apropiati
* 5.se calculeaza noul centroid pentru fiecare cluster din cele N pe baza instantelor repartizate in el
* 6.algorimtul este reluat de la pasul 3
* 7.algoritmul se opreste cand instantele sunt repartizate succesiv in acelasi cluster de mai multe ori consecutiv (de exp.10 ori) sau se poate asigura rularea de 100 de ori dupa care algoritmul sa se opreasca.

**Exemplu numeric:**

-se va folosi distanta Euclidiana pentru calculul distantei:

J=

Presupunem instantele ( clientii)

**Client Criteriu 1 Criteriu 2**

Client 001 1 7

Client 002 2 8

Client 003 4 6

Client 004 1 9

Client 005 3 6

Client 006 2 9

Se stabilesc centroizii ( aleator sau pot fi instante din setul de date), in cazul nostru numarul de oferte care va fi egal cu numarul de profile identificate in date:

* numarul de clustere ( grupe/profile, dat ca parametru la noi in aplicatie): 2
* se aleg centroizii: K1 (1,4), K2 (3,10) in acest caz au fost alesi aleatoriu cat mai departati. Noi putem sa ii alegem ca fiind primele N inregistrari din fisier/baza de date.

Calcul distante de la centroizi la instante(clienti):

* pentru prima instanta (client) se calculeaza distanta pana la cei doi centroizi alesi si se alege distanta minima. Daca sunt egale se atribuie primului cluster.

J1= |1-1|2 + |7-4|2 = 0 + 9 = **9** ( de la Client 001 (1,7) pana la centroidul K1(1,4))

J1 = |1-3|2 + |7-10|2 = 4 + 9 = 13( de la Client 001 (1,7) pana la centroidul K2(3,10)

Se alege distanta mai mica => Clientul 001 este incadrat in clusterul 1 cu centroidu l in K1 (9<13)

J2= |2-1|2 + |8-4|2 = 1 + 16 = 17 ( de la Client 002 (2,8) pana la centroidul K1(1,4))

J2 = |2-3|2 + |8-10|2 = 1 + 4 = **5** ( de la Client 002 (2,8) pana la centroidul K2(3,10)

Se alege distanta mai mica => Clientul 002 este incadrat in clusterul 2 cu centroidu l in K2 (5<17)

J3= |4-1|2 + |6-4|2 = 9 + 4 = **13** ( de la Client 001 (4,6) pana la centroidul K1(1,4))

J3 = |4-3|2 + |6-10|2 = 1 + 16 = 17( de la Client 001 (4,6) pana la centroidul K2(3,10)

Se alege distanta mai mica => Clientul 003 este incadrat in clusterul 1 cu centroidu l in K1 (13<17)

J4= |1-1|2 + |9-4|2 = 0 + 25 = 25 ( de la Client 001 (1,9) pana la centroidul K1(1,4))

J4 = |1-3|2 + |9-10|2 = 4 + 1 = **5** ( de la Client 001 (1,9) pana la centroidul K2(3,10)

Se alege distanta mai mica => Clientul 004 este incadrat in clusterul 2 cu centroidu l in K1 (5<25)

J5= |3-1|2 + |6-4|2 = 4 + 4 = **8** ( de la Client 001 (3,6) pana la centroidul K1(1,4))

J5 = |3-3|2 + |6-10|2 = 0 + 16 = 13( de la Client 005 (3,6) pana la centroidul K2(3,10)

Se alege distanta mai mica => Clientul 005 este incadrat in clusterul 1 cu centroidu l in K1 (8<16)

J6= |2-1|2 + |9-4|2 = 1 + 25 **=** 9 ( de la Client 001 (2,9) pana la centroidul K1(1,4))

J6 = |2-3|2 + |9-10|2 = 1 + 1 = **2** ( de la Client 001 (2,9) pana la centroidul K2(3,10)

Se alege distanta mai mica => Clientul 006 este incadrat in clusterul 1 cu centroidu l in K1 (2<9)

Prin urmare, distributia in clustere este:

*In clusterul 1: Client01, Client03, Client05*

*In clusterul 2: Client02, Client04 , Client06*

Se formeaza noii centroizi ( ofertele in cazul nostru, dar pentru ca nu s-a terminat algoritmul sunt oferte intermediare).

Mod de calcul:

K1’(clienti alocati in clusterul 1/ numar de clienti alocati) = > k1’ ( (1+4+3)/3, (7+6+6/3)) => k1’( 2.66, 6.33)

// (1+4+3)/3 se ia fiecare valoare de la criteriul 1 de la clientii alocati in clusterul 1 si se imparte la numarul de clienti.

// (7+6+6/3) se ia fiecare valoare de la criteriul 1 de la clientii alocati in clusterul 1 si se imparte la numarul de clienti.

K2’( (2+1+2)/3, (8+9+9)/3 ) => k2’ (2.5 , 8.66)

Apoi se reia algoritmul pentru K1’ si K2’ pana nu se mai modifica valorile pentru ei sau se ruleaza de 100 de ori algoritmul.

Sa presupunem ca ajungem ca valori finale pentru centroizi ( oferte) la: K1’’’’(2, 10) si K2’’’’(4, 15) iar:

- in clusterul 1 ( grupa de instante reprezentata de K1’’’’ avematribuiti *Client01, Client03*

- in clusterul 2 ( grupa de instante reprezentata de K1’’’’ avematribuiti *Client02, Client04, Client 5, Client 6*

**In acest caz:**

- clientii 1 si 3 vor primi oferta constituita din K1’’’’(2,10)

- clientii 2,4,5,6 vor primi oferta constituita din K2’’’’(4,15)

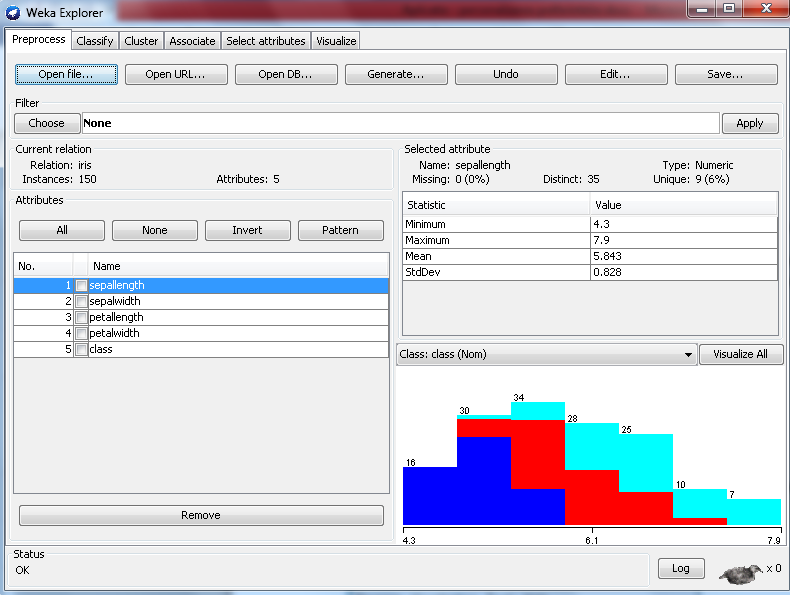
**Exmeplu de aplicatie care implementeaza algoritmul SimpleKMeans**: Weka ( se poate descarca gratuit de la <http://www.cs.waikato.ac.nz/ml/weka/downloading.html>)

Un ghid de folosire a sa este mai jos redat in imagini si comenatrii:

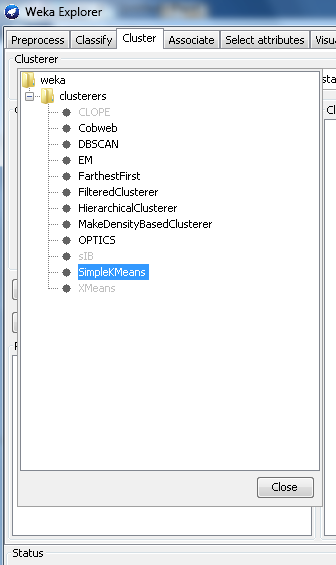
* dupa instalare rulati executabilul si se va deschide fereastra:



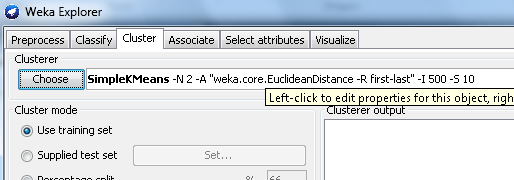
* se alege optiunea Explorer.
* se va deschide un fisier de lucru (Open file...) si in direcotrul unde a fost instalat Weka exista un director Data din care puteti prelua un fisier pentru analiza. Presupunem ca alegem fisierul “iris.arff”
* dupa deschiderea fisierului, prima interfata ar trebui sa arate asa ca in figura de mai jos. Se va elimina atributul **class** utilizand butonul **Remove**



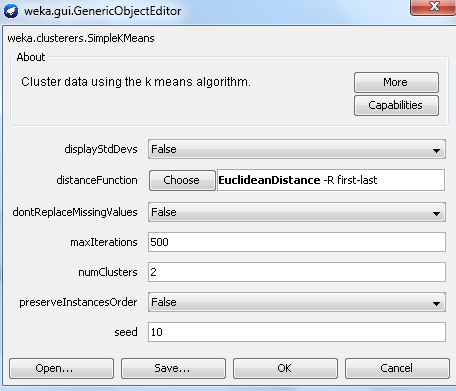
* pentru a folosi algoritmul Simple Kmeans, este nevoie a parcurge etapele:
  + accesati tab-ul Cluster (al treilea)
  + din sectiunea Clusterer se alege algoritmul SimpleKMeans apasand “Choose”



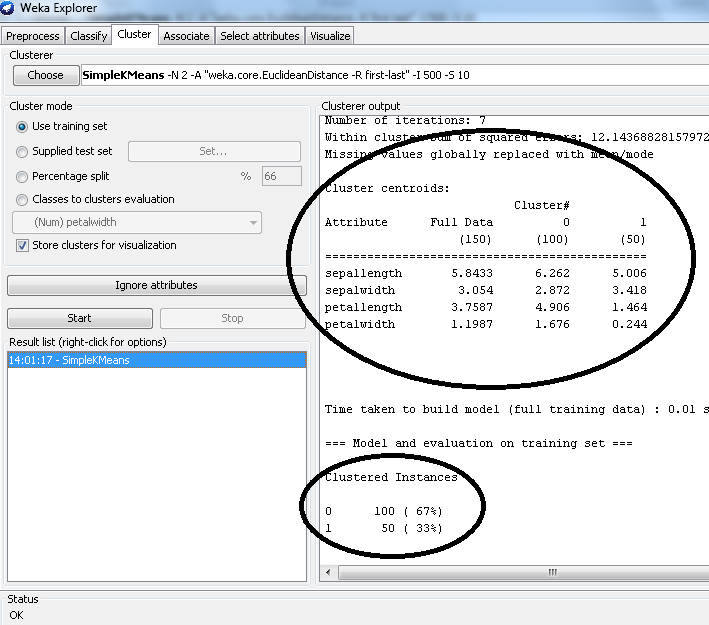
* + dupa alegerea algoritmului, se pot alege parametrii dand click pe numele sau



* + in fereastra pentru parametrii, acestia se pot modfica. **numClusters** reprezinta numarul de cluster ( grupe – 2 in cazul de fata).



* Rulare algoritm:
  + se apasa butonul start. Rezultatele sunt redate mai jos:



* + Interpretare rezultate:
    - coloana 1 - atributele ( la noi Criteriu 1, Criteriu 2....)
    - coloana 2 – mediile pentru atributul respectiv tinand cont de toate instanele
    - coloana 3 – clusterul 0 ( grupa 0) – pentru fiecare atribut se calculaeaz valoarea tinand cont doar de instantele din clusterul respectiv (100 de instante, pentru noi clienti).
    - coloana 4 – clusterul 4 ( grupa 1) – pentru fiecare atribut se calculeaza valoarea tinand cont doar de instantele din clusterul respectiv (50 de instante).

In cerculatul mic avem informatia despre ponderea si numarul de instante din fiecare grupa ( cluster).

Algoritmul care face acest lucru in gasest in directorul in care ai instalat weka de exemplu

(D:\Weka-3-6\_ok\doc\weka\clusterers).

Pentru distanta Euclidiana gasesti in directorul “core” functia EuclideanDistance.html.

In masura inc are se poate, mi-ar fi de folos si acea documentatie din Java care este generata pentru proiect.

Pe net mai sunt diverse implementari ale algoritmului SimpleK Means. Unele dintre ele sunt:

<http://stackoverflow.com/questions/21111070/implementation-of-k-means-clustering-algorithm> (aici putem aplica sa ruleze doar de 100 de ori).

<http://mnemstudio.org/clustering-k-means-example-1.htm> ( are si codul java, il accesezi prin primul link **View Java Code) – cred ca va ajuta cel mai mult**

**Pentru Jade (agenti):**

**http://jade.tilab.com/**