Primena UML dijagrama aktivnosti u predstavljanju Data Mining modela tehnikom genetskih algoritama

Muzafer Saračević , Sead Mašović , Zoran Lončarević

***Sadržaj* — Cilj ovog rada je da se predstave faze u procesu inteligentne obrade podataka (*Data Mining*) i način kreiranja i izbora modela. U radu je primenjena tehnika genetskih algoritama i razvijen za njih model primenom UML dijagrama aktivnosti koji najbolje predstavljaju način funkcionisanja algoritama i postupak dobijanja rešenja. U radu je dat vizuelni plan koji kasnije može da posluži u daljem razvoju modela i implementaciji u programskom jeziku JAVA.**

***Ključne reči* — Genetski algoritmi, Java programiranje, *Data Mining*, Objektno-orijentisana analiza, UML dijagrami.**

I. UVOD

DATA Mining se može definisati kao proces pronalaženja skrivenih zakonitosti i veza među podacima. To je tehnika pretraživanja podataka u cilju identifikacije traženih uzoraka i njihovih međusobnih relacija. To je postupak izdvajanja interesantnih, novih i potencijalno korisnih informacija ili uzoraka, sadržanih u velikim bazama podataka, u cilju donošenja ispravnih poslovnih odluka [1]. Prilikom pretrage podataka softver pomaže analitičaru da reši neke od sledećih problema [2] :

1) Klasifikacija kod koje se analiziraju skupovi podataka, otkrivaju skrivene veze i utvrđuju elementi (funkcije) za njihovo grupisanje u jednu od nekoliko klasa.
2) Asocijacija podataka gde se utvrđuju osobine koje se javljaju zajedno kod više uzoraka, odnosno veze među proizvoljnim atributima.
3) Grupisanje (*Clustering*), proces određivanja grupa podataka koji su međusobno slični, ali različiti od ostalih grupa podataka. Pri tome se indentifikuju i promenljive po kojima se vrši najbolje grupisanje.
4) Predviđanje (*Numeric prediction*) u kojem se otkriva ponašanje objekta posmatranja tokom vremena, vrše se predviđanja, utvrđuju se pravilnosti iz primera i na osnovu toga određuju očekivane numeričke vrednosti.

Muzafer Saračević, Prirodno-Matematički fakultet, Univerzitet u Nišu, Višegradska 33,18000 Niš, Srbija (telefon: 381-60-4979797 , e-mail: muzafers@gmail.com)
Sead Mašović, Fakultet Organizacionih nauka, Univerzitet u Beogradu, Jove Ilića 154, 11000 Beograd, Srbija (telefon: 381-60-6660646 , e-mail: sekinp@gmail.com)
Zoran Lončarević, Fakultet Organizacionih nauka, Univerzitet u Beogradu, Jove Ilića 154, 11000 Beograd, Srbija (telefon: 381-64-2147204 , e-mail: zokinp@gmail.com)

Cilj UML modelovanja i analize jeste da se kreira model odnosno vizuelni plan koji će nam poslužiti kao šablon (obrazac) za konkretno programiranje u nekom od programskih jezika. Pored toga, model nam omogućava da bolje sagledamo sistem i način njegovog funkcionisanja.

II. FAZE U PROCESU DATA MINING-A

Životni ciklus jednog *data mining* projekta se sastoji iz sledećih faza:

Sakupljanje podataka je prvi korak u *data mining* projektu. Poslovni podaci su uskladišteni u brojnim sistemima , internetu, bazama podataka kompanija, i početni korak predstavlja prenos relevantnih podataka u bazu podataka gde se podaci analiziraju.

Filtriranje podataka i transformacija je najintenzivniji korak u *data mining* projektu kad su resursi u pitanju. Cilj filtriranja podataka je odstranjivanje irelevantnih i suvišnih informacija iz skupa podataka. Cilj transformacije podataka je promena izvornog podatka u drugačiji format tipa podataka. Postoje različite tehnike koje se mogu primeniti za korak filtriranja i transformaciju podataka, a najčešće korišćene su transformacija tipova podataka, neprekidna transformacija kolona, grupisanje, rad sa vrednošću koja nedostaje itd.

Kreiranje i izbor modela je treći korak koji se primenjuje nakon filtriranja i transformacije podataka. U radu je akcenat na ovom delu procesa *Data Mining*-a. Tek kada se podaci filtriraju i kada se promenljive transformišu u pogodne tipove podataka, može se započeti sa kreiranjem modela. Pre kreiranja modela treba da razumemo cilj *data mining* projekta i vrstu *data mining* zadatka koji će se koristiti. Za svaki *data mining* problem postoji nekoliko odgovarajućih algoritama [3]. Konkretno, za realizaciju i kreiranje modela u ovom radu su odabrani genetski algoritmi . Preciznost algoritma zavisi od prirode podataka kao što su broj stanja atributa koji se koriste za predviđanje, prenos vrednosti svakog atributa, veza između atributa itd.

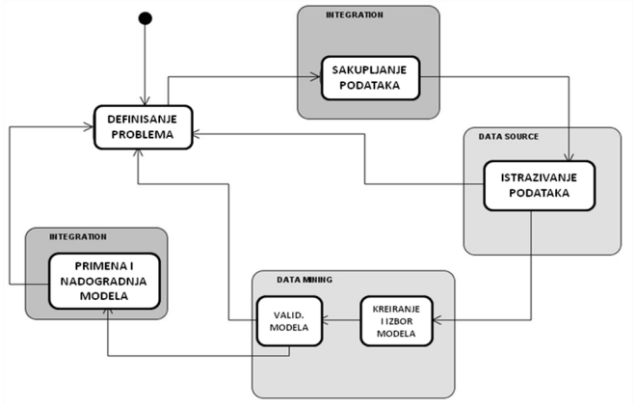
Procena kvaliteta modela koja se realizuje preko alata za evaluaciju kvaliteta modela. Najpoznatiji je *lift dijagram*. On koristi već istreniran model kako bi predvideo vrednosti koje će se dobiti iz skupa podataka koji se testira. Na osnovu vrednosti koje se dobiju i verovatnoće on grafički prikazuje model na dijagramu.

Kreiranje izveštaja se vrši nakon kreiranja modela i evaluacije kvaliteta koji se dostavlja menadžerima na uvid. Većina *data mining* alata ima osobinu kreiranja izveštaja

koji omogućuje korisnicima da generišu prethodno definisan izveštaj sa tekstualnim i grafičkim detaljima *data mining* modela.

Ocenjivanje ili predviđanje modela *(scoring)* gde važi pravilo da moramo da imamo već istrenirani model i skup novih podataka da bi dobili predviđene vrednosti.

Integracija data mining modela u aplikaciju predstavlja ponovnu primenu poslovne inteligencije na poslovni sistem tj. zatvaranje petlje za analizu. Sve više poslovnih aplikacija uključuje i data mining komponentu a prednosti *data mining*-a su velike.



Sl. 1. Dijagram aktivnosti u procesu *Data Mining*-a

Integrisanje data mining osobina, pogotovo komponente za predviđanje u aplikacije jedan je od bitnijih koraka *data mining* projekta. Ovo je ključni korak za uvođenje *data mining*-a u masovnu upotrebu.

Upravljanje modelom je završna faza. Trajanje jednog *data mining* modela je ograničeno. Nova verzija modela se mora praviti često.

III. GENETSKI ALGORITMI I POSTUPAK MODELOVANJA

Uopšteno govoreći, sve *Data Mining* tehnike se mogu podeliti u dve grupe [1] :

- *Discovery data mining* - tehnike za otkrivanje novih znanja (informacija)
- *Predictive data mining* - tehnike za predvidjanja .

Analitičke tehnike koje se koriste u *data mining-u* u najvećem broju slučajeva su odavno poznate matematičke tehnike i algoritmi, znači u samom procesu analize podataka nema ničeg novog.

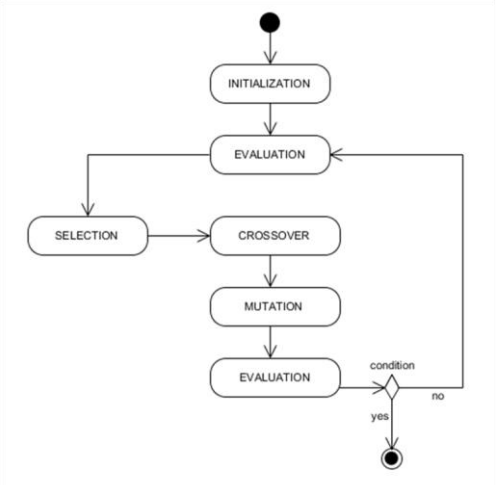
Genetski algoritmi predstavljaju tehniku koja se koristi za klasifikaciju i klasterovanje. Genetski algoritmi se baziraju na principu genetske modifikacije, mutacije i prirodne selekcije. Genetski algoritam kreira određen broj nasumičnih rešenja problema. Sva ta rešenja ne moraju biti dobra, neka grupa rešenja može da bude sasvim preskočena, a može da dođe i do preklapanja rešenja. Loša rešenja se odbacuju, a dobra zadržavaju. Dobra rešenja se zatim hibridizuju i ceo proces se ponavlja. Na kraju, slično procesu prirodne selekcije, ostaju samo najbolja rešenja [4].

Za praktičnu primenu genetski algoritmi dobijaju još jedan plus zbog toga što u praksi nije potrebno naći optimalno rešenje već je dovoljno dobro rešenje i ono koje je u okolini optimuma [5]. U konkretnom modelovanju i aplikaciji smo naveli sledeću kombinaciju operatora prilikom realizacije ovih algoritama : prirodna selekcija, *2opt* metoda (mutacija) i *Greedy Subtour crossover* (ukrštanje).

Ovakav način analize i dizajna za genetske algoritme formira model na osnovu kojeg ćemo moći poboljšati efikasnost algoritama na tri načina i to povećanjem verovatnoće postizanja dobrih rešenja, povećanjem kvaliteta dobijenih rešenja i skraćenjem trajanja izvođenja. Trajanje izvođenja se može skratiti na tri načina i to povećanjem brzine konvergencije (čime se smanjuje broj iteracija), smanjenjem trajanja izvođenja jedne iteracije i paralelnim izvođenjem celog algoritma ili samo pojedinih genetskih operatora [6]. Navedeni ciljevi mogu se ostvariti podešavanjem parametara, optimizacijom izvornog programa i paralelizacijom pomenutih algoritama. Naveden je algoritam na osnovu kojeg ćemo raditi modelovanje i analizu [1] :

```
ULAZ:
P // populacija
IZLAZ:
P1 // poboljsana populacija
GENETSKI ALGORITAM:
repeat
N=|P|; P1=0;
repeat
i1,i2= select(P);
o1,o2= cross(i1,i2);
o1= mutate(o1); o2= mutate(o2);
P1=P1 U {o1,o2};
until |P1|=N;  P=P1;
until <zadovoljen kriterijum zaustavljanja>
```

Dijagram aktivnosti predstavlja akcije koje se izvode, konkretno dijagram (Sl.1) daje generalni pogled na aktivnosti koje se dalje dekomponuju.



Sl.2.- Način funkcionisanja genetskih algoritama

A. *Dijagrami aktivnosti*

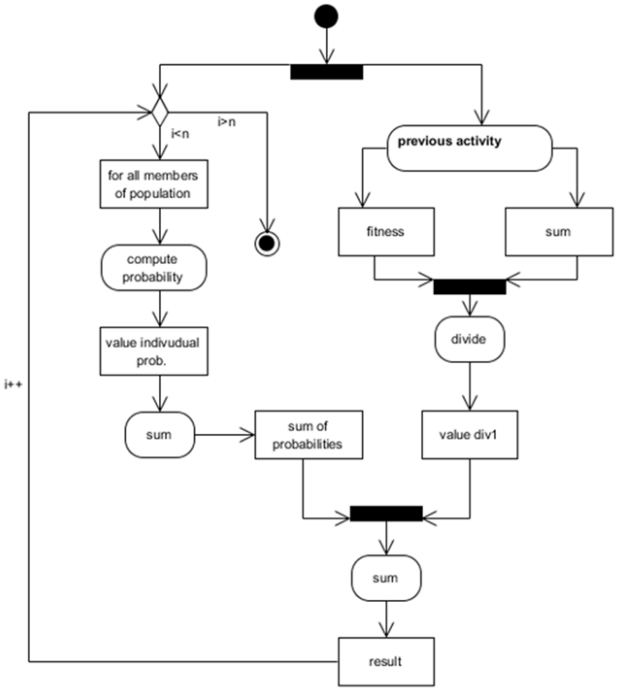
Dijagrami aktivnosti služe za modelovanje dinamičkih aspekata sistema. Oni imaju mogućnost da prikažu: proceduralnu logiku, poslovni proces ili tok posla. Slični su blok-dijagramima za opis algoritama (dodatno, podržavaju paralelno ponašanje). Akcije se implemetiraju kao metodi klasa ili neke podaktivnosti. Dijagrami aktivnosti opisuju šta se radi, ali ne kažu ko šta radi. Ako želimo istaći ko šta radi (koja klasa je odgovorna za

određenu akciju), podelićemo dijagram na particije. Akcije mogu biti razbijene na podakcije.

B. Operatori selekcije

U konkretnom primeru smo koristili tzv. prirodnu selekciju**.** Selekcija je proces kojim se osigurava prenošenje boljeg genetskog materijala iz generacije u generaciju. Postupci selekcije međusobno se razlikuju po načinu odabira jedinki koje će se preneti u sledeću generaciju. Proces selekcije eliminiše najgore jedinke iz generacije. Jedinke se eliminišu tako da se sačuva različitost populacije, odnosno eliminišu se slične jednike. Za početak cela se populacija sortira prema pogodnosti (*fitness* funkcija) [8].

Sledeći dijagrami aktivnosti predstavljaju razlaganje pomenutih aktivnosti na podaktivnosti koje sadrže konkretne akcije. Način funkcionisanja operatora selekcije je realizovan kroz sledeće prikazane UML dijagrame aktivnosti. Dijagram (sl.3.) daje vizuelni prikaz kako se u postupku selekcije izračunava tzv.fitness funkcija (funkcija cilja) i verovatnoća za svakog člana pojedinačno, dok se u dijagramu (sl.4.) navodi način upotrebe dobijenih vrednosti iz prethodne aktivnosti i postupak selekcije članova na osnovu njihovih verovatnoća.

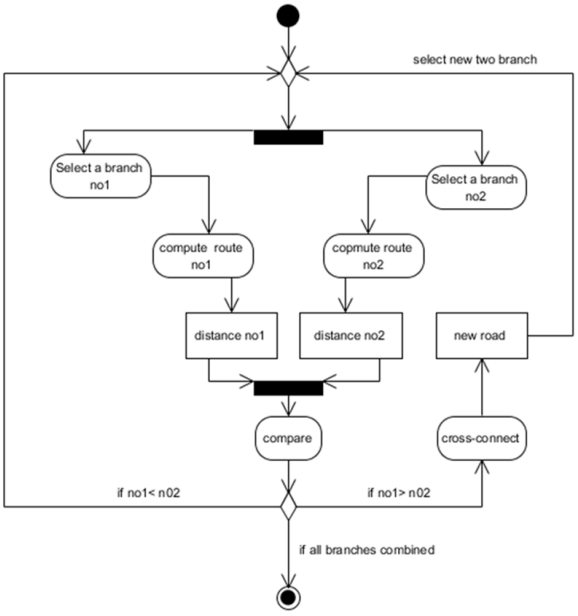


Sl. 3. Način funkcionisanja operatora selecije

U ovom postupku se upoređuje sličnost *fitness* vrednosti susednih jedniki i ukoliko je njihova razlika manja od predefinisanog malog realnog pozitivnog broja ε, eliminše se jedna od n-torki. To se ponavlja dok je broj eliminisanih jedniki manji od R. Ako je nakon ovog postupka broj jedinki koje smo eliminisali i dalje manji od R eliminišu se jedinke s lošijom vrednošću *fitness* funkcije.

C. Operatori mutacije

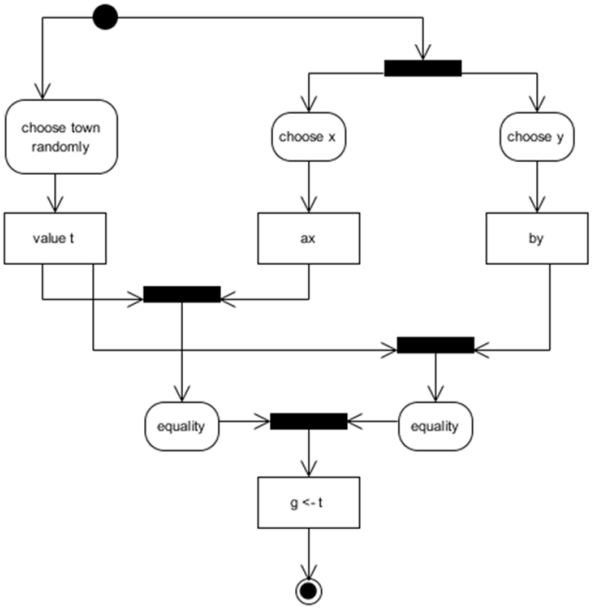
U radu smo kao operator mutacije koristili tzv. *2opt metoda.* Ova metoda je jedna od najpoznatijih metoda lokalnog pretraživanja u algoritmima koji rešavaju problem. Mutacija pomaže izbegavanju lokalnih optimuma funkcije cilja. Primenom operatora mutacije postiže se raznolikost genetskog materijala i omogućava pretraživanje novih potencijalno najboljih rešenja. Mutacije takođe obnavljaju genetski materijal. Kad nam sve jedinke na jednom delu hromozoma imaju istu vrednost, ta se vrednost tokom ukrštanja nikad neće menjati. Tokom procesa mutacije postoji mogućnost da se upravo taj deo hromozoma promeni[9]. Način funkcionisanja ovog operatora je realizovan kroz UML dijagram aktivnosti (sl.5). Prikazani dijagram je dekompozija aktivnosti „mutacija“ koji predstavlja skup akcija koje se izvode unutar operatora [10].



Sl. 5. Način funkcionisanja operatora mutacije

D. Operatori ukrštanja

Kao operator ukrštanja koristili smo GSX *(greedy subtour crossover)* koji radi tako da iz oba roditelja uzima što je moguće duži podskup rešenja. Na taj je način najbolje sačuvan genetski materijal roditelja. Način funkcionisanja ovog operatora je realizovan kroz UML dijagram aktivnosti (sl.6).



Sl. 6. Način funkcionisanja operatora ukrštanja

To zapravo znači da ako postoje dva hromozoma koja oba sadržavaju podskupove optimalnog rešenja, ovim ukrštanjem se može vrlo brzo doći do spajanja tih delova što naravno dovodi do brže konvergencije problema[11].

Navedeni dijagram predstavljaju akcije koje se izvršavanju unutar aktivnosti „ukrštanje“ i opisuju operator *GSX* sa aspekta ponašanja modela odnosno prikazuju se akcije, njihov redosled izvršavanja i uslovi koji postoje.

IV. ZAKLJUČAK

Data mining je nova i moćna tehnologija koju firme razvijenog sveta koriste u istraživanju tržišta i otkrivanju potencijalnih klijenata. To je metoda pretraživanja podataka koja je doživela vrhunac zahvaljujući razvoju računarske tehnologije jer je tek razvitkom brzih računarskih sistema postalo moguće efikasno pretraživati velike količine sirovih informacija. U ovom radu smo naveli UML modelovanje koje je specificno za ovakve probleme, odnosno glavni cilj je da se bliže predstavi procedura i funkcionisanje genetskih algoritama i način rešavanja problema primenom istih.

Primena genetskih algoritama je vrlo široka, oni su zapravo samo princip, ideja odnosno smernica kako neki problem rešiti na drugačiji način od klasičnih metoda, jer je sve na korisniku da se sam odluči da li će razvijati svoj vlastiti algoritam ili će probati svoj problem prilagoditi već nekom postojećem algoritmu koji rešava neku sličnu klasu problema. Takođe, vidi se da su genetski algoritmi korisni za one klase problema koje se ne mogu rešiti na klasične načine. Postupak UML modelovanja pre svega ima za cilj i glavni zadatak da realizuje vizuelno planiranje i predstavljanje problema. Glavne prednosti ovakvog načina rešavanja problema, konkretno u ovom slučaju, su što je problem razradjen i izvršena je analiza, što rezultuje konkretnim modelom koji može poslužiti za dalji razvoj i nadogradnju.

DODATAK

Navedene su tri metode *selekcija() , ukrstanje() i mutacija()* koje realizuju rad genetskih algoritama i koje se baziraju na prethodno navedenim UML dijagramima.

```
public static void selekcija() {
    int Vr_popul[] = new int[nM];
    double d[][] = new double[nM][nB];
    double e[] = new double[nM];
    for (int i=0; i<nM; ++i){
   for (int l=0; l<nB; ++l) d[i][l] = c[i][l];
      e[i] = f[i]; Vr_popul[i] = i; }
    shuffle(Vr_popul); int k = 0;
    for (int i=0; i<nA; ++i) {
      if (e[Vr_popul[k]] < e[Vr_popul[k+1]]){
          for (int l=0; l<nB; ++l)
  c[i][l]=d[Vr_popul[k]][l];
       f[i] = e[Vr_popul[k]]; }
      else {
        for (int l=0; l<nB; ++l)
c[i][l]=d[Vr_popul[k+1]][l];
        f[i] = e[Vr_popul[k+1]];
      }
      k += 2; }

}

public static void ukrstanje() {
    Random rVrednost = new Random();
    int k = 0;
    for (int i=nA; i<nA+nA/2; ++i) {
   int nx = 1 + (int)(nB*rVrednost.nextDouble());
for (int l=0; l<nx; ++l){
    c[i][l] = c[k][l]; c[i+nA/2][l] = c[k+1][l]; }
      for (int l=nx; l<nB; ++l){
    c[i][l] = c[k+1][l]; c[i+nA/2][l] = c[k][l]; }
      k += 2;
    }}
```

```
public static void mutacija() {
    Random rVrednost = new Random();
    double r[] = new double[nB];
    for (int i=0; i<ne; ++i) {
      for (int l=0; l<nB; ++l) r[l] = c[i][l];
      int mb = (int)(nB*pMutate+1);
      for (int j=0; j<mb; ++j){
      int ib = (int)(nB*rVrednost.nextDouble());
      r[ib] = rVrednost.nextDouble(); }
      double e = cena(r);
      if (e<f[i]){
for (int l=0; l<nB; ++l) c[i][l] = r[l];
f[i] = e;
}
}
    int mmax = (int)((nM-ne)*nB*pMutate+1);
    for (int i=0; i<mmax; ++i) {
int ig = (int)((nM-ne)*rVrednost.nextDouble()+ne);
      int ib = (int)(nB*rVrednost.nextDouble());
      c[ig][ib] = rVrednost.nextDouble();
    }
}
```

LITERATURA

[1] U.M. Fayyad, “Knowledge Discovery and Data Mining”, Portland, OR, *AAAI Press*, pp. 226-231, 1996.

[2] Jiawei H., Kamber M.: “ Data Mining: concepts and techniques”, *Academic Press*, San Diego, 2001.

[3] Xindong Wu, V.Kumar,“Top 10 Algorithms in Data Mining”, *Knowledge and Information Systems*, 141: 1-37,2008.

[4] Nagar, P., Srivastava, S.,. “Application of Genetic Algorithms in Data Mining”, *2nd National Conference on Challenges & Opportunities in Information Technology*. 4p. 2008.

[5] Freitas, A.A., E. Noda and H.S. Lopes,. “Discovering interesting prediction rules with a genetic algorithm”’. *Proc. Conf. Evolutionary Computation*, pp: 1322–1329, 1999.

[6] Minaei-Bidgoli, B., Punch, W., “Using Genetic Algorithms for Data Mining Optimization in an Educational Web-based System”. *Genetic and Evolutionary Computation*, Part II. pp.2252–2263, 2003.

[7] Falkenauer E. “Genetic Algorithms and Grouping Problems”, *John Wiley & Sons*, 1998.

[8] Shah SC, Kusiak A. ,” Data mining and genetic algorithm based gene/SNP selection”, *Artif.Intell.Med*, 31:183, 2004.

[9] Park Y, Song M.,” A genetic algorithm for clustering problems”, *Genetic Programming 1998: Proceeding of 3rd Annual Conference*, 568-575. Morgan Kaufmann, 1998.

[10] R.S.Sexton, N.A Sikander, ”Data mining using a genetic algorithm-trained neural network”. *Int. J. Intell Systems Account, Finance Manage*, 10:201-210, 2001.

[11] M. Pei, E. D. Goodman, F. Punch, “Feature Extraction using genetic algorithm”, *Case Center for Computer-Aided Engineering and Manufacturing*, W. Department of Computer Science, 2000.

ABSTRACT

The purpose of this paper is to present the stages in the process of intelligent data processing (*Data Mining*) and a way of creation and selection models. The paper has applied genetic algorithms and techniques developed to model them using UML activity diagram that best represent the way the algorithms and procedures for obtaining solutions. The paper gives a visual plan that can later be used in the further development of models and implementation in a programming language JAVA.

APPLICATION OF UML ACTIVITY DIAGRAM IN REPRESENTATION DATA MINING MODEL USING GENETIC ALGORITHMS

Muzafer Saračević , Sead Mašović, Zoran Lončarević