Primena metode k najbližih suseda u teoriji odlučivanja

O modelima odlučivanja indukovanim iz podataka

U klasičnoj teoriji odlučivanja postupak donošenja odluka je striktno vezan za eksperta ili grupu eksperata koji modeluju problem i rešavaju ga primenom različitih metoda. Mišljenje, iskustvo i sklonost ka riziku donosioca odluke je dodato kroz izražavanje preferencija, matrice procene i slično. Međutim, postoje situacije kada DO nije u stanju da brzo i jednostavno reši problem. U takvim situacijama se uvodi računarska podrška u proces donošenja odluka. Ideja računarske podrške u odlučivanju je da se donese slična odluka kao i u prošlosti. Dakle, neophodno je da preduzeće ima strukturiran problem, istorijske podatke o tom problemu i podatak koja je odluka doneta. Sve navedeno predstavlja „iskustvo“ preduzeća koje može da se predstavi tabelom slučajeva na sledeći način:

RB	x_1	x_2	x_3	y
1	x_{11}	x_{12}	x_{13}	y_1
2	x_{21}	x_{22}	x_{23}	y_2
3	x_{31}	x_{32}	x_{33}	y_3
4	x_{41}	x_{42}	x_{43}	y_4
5	x_{51}	x_{52}	x_{53}	y_5

Redovi predstavljaju slučajeve sa kojima se preduzeće već susrelo u prošlosti. Kolone tabele slučajeva predstavljaju atribute koji opisuju problem. Razlikujemo ulazne atribute (obeleženi su sa *x*) i izlazne atribute (obeleženi su sa *y*). Ulazni atributi su nam poznati i oni opisuju problem, dok izlazni atributi predstavljaju odluku koja je doneta. U opštem slučaju izlazni atributi nam nisu poznati. Atributi su analogni kriterijumima odlučivanja. Vrednosti u tabeli slučajeva predstavljaju stanje uzorka (red) za atribut (kolona). Tako x_{12} predstavlja stanje prvog uzorka za drugi atribut.

Koristeći iskustvo možemo napraviti model odlučivanja naučenog iz podataka koji će automatizovati proces donošenja odluka. Ovim pristupom „učimo“ kako DO donosi odluke, te možemo doneti slične odluke za slične probleme u budućnosti. Napominjemo da postoji odgovornost iza svake odluke i da ako model pogreši, DO je odgovoran za tu odluku.

Pravljenje modela odlučivanja naučenog iz podataka ima širok dijapazon primene. Uzmimo za primer bankarsko poslovanje i problem odlučivanja da li nekome dati kredit ili ne. Svaka banka prilikom odobravanja kredita uzima podatke o klijentima, poput stanja na računu, visine mesečnih prihoda, broja podignutih kredita, broja isplaćenih kredita, iznosa na štednom računu i slično. Ovi podaci predstavljaju atribute (x_1, …, x_5), a kako je reč o istorijskim podacima, znamo

da li je klijent vratio kredit ili ne (y). Zatim se uočavaju veze između ulaznih atributa i izlaznog atributa koje su se javljale u prošlosti (tabeli slučajeva) i na osnovu toga se pravi model odlučivanja. Na kraju, kada se pojavi novi klijent u banci, uzmu se podaci o stanju na računu, mesečnim prihodima, broju podignutih kredita, broju isplaćenih kredita, iznosa na štednom računu i ostalim atributima, primeni se model odlučivanja i dobije predviđanje da li će klijent vratiti kredit ili ne. Ukoliko model odlučivanja „kaže“ da će klijent vratiti kredit, njemu se odobrava kredit. U suprotnom, klijentu se ne odobrava kredit.

Sledeći primer pravljenja modela odlučivanja jeste prepoznavanje bankrota preduzeća. Svako preduzeće na kraju godine daje izveštaj o svom poslovanju. Iz izveštaja se mogu uzeti informacije npr. o kapitalu preduzeća (x_1), prodaji (x_2), ukupnim troškovima (x_3) i broju zaposlenih (x_4). Ciljni atribut (y) govori da li će preduzeće bankrotirati naredne godine. Podaci se uzimaju iz prethodnih godina, gde je ova informacija već bila poznata, i time se popunjava tabela slučajeva. Pravi se model odlučivanja i kada preduzeća predaju svoje izveštaje za tekuću godinu, propuštaju se kroz model odlučivanja i dobija se predviđanje da li će preduzeće bankrotirati ili ne. Ta informacija može da se koristi kao signal da je preduzeću možda potrebno pomoći.

Modeli odlučivanja naučeni iz podataka se primenjuju i u medicini kao pomoć u dijagnostikovanju bolesti. Pacijent dolazi kod doktora sa određenim simptomima. To mogu biti temperatura (x_1), glavobolja (x_2), bol u grudima (x_3), kašalj (x_4) i otežano disanje (x_5), a izlazni atribut je da li pacijent ima upalu pluća ili ne (y). Kako doktor već ima iskustva, sastavio je tabelu slučajeva na osnovu koje je napravio model odlučivanja. Kada dođe novi pacijent, doktor pogleda koje simptome od gore pomenutih pacijent ima i daje „predviđanje“ da li pacijent ima upalu pluća ili ne i na osnovu toga leči.

Razlog zašto se primenjuju modeli odlučivanja naučeni iz podataka, a ne ekspertsko znanje, može biti u prevelikom broju odluka koje je potrebno doneti, pa se proces donošenja odluke može automatizovati. Dakle, proces donošenja odluke primenom modela naučenih iz podataka je brži, a odluka može biti bolja. Zatim, modeli odlučivanja naučeni iz podataka lakše mogu da uoče zavisnosti ciljnog atributa (y) i ulaznih atributa (x_1, x_2, …, x_n), naročito ukoliko postoji veći broj ulaznih atributa.

Bitno je napomenuti da modeli odlučivanja naučeni iz podataka ne moraju nužno tačno predviđati ishod. Drugim rečima, ovakvi modeli često imaju grešku. Na kraju, modeli odlučivanja naučeni iz podataka nikako ne mogu da zamene DO i da ga oslobode odgovornosti, tj. DO je uvek odgovoran za krajnju odluku.

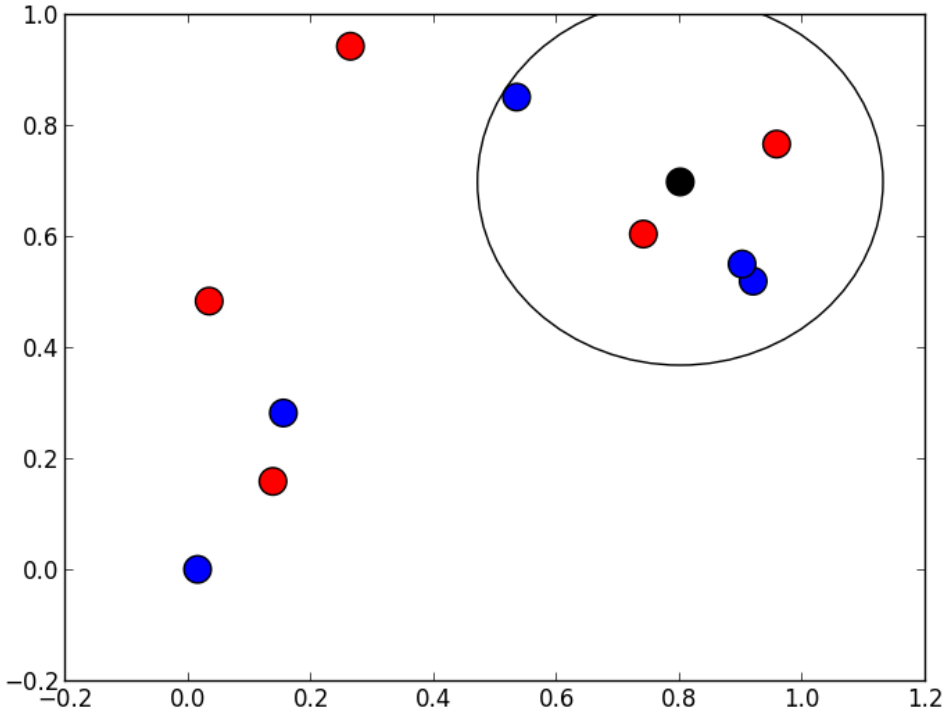
Modeli odlučivanja naučeni iz podataka predviđaju ishod (y) na osnovu ulaznih atributa (x_1, x_2, …, x_n), a primenjuju se na slučajevima (klijent, pacijent i slično) koje model nije video, odnosno primenjuju se na onim slučajevima za koje se ne zna ishod (y).

Bitno je napomenuti da kvalitet zaključaka zavisi od uzorka u tabeli slučajeva. Što je veći uzorak (ima više slučajeva), zaključak koji će se dobiti će biti bolji.

Algoritam k najbližih suseda

Algoritam k najbližih suseda je algoritam otkrivanja zakonitosti u podacima (eng. data mining) koji se koristi za probleme predviđanja ishoda (bilo da je numerička ili nenumerička vrednost), a na osnovu sličnosti slučaja za koji treba doneti odluku sa slučajevima iz tabele slučajeva (baze znanja). Kada je potrebno doneti odluku posmatra se k najbližih (najsličnijih) suseda i donosi se ona odluka koja se najčešže pojavljivala kod posmatranih k najbližih suseda.

Uzmimo za primer da imamo dva ulazna atributa (kriterijuma) koja predstavljaju npr. koeficijent inteligencije i vrednost sa testa ličnosti i deset alternativa u tabeli slučajeva. Svaka alternativa ima pridodat izlazni atribut koji predstavlja npr. da li se zaposleni uklopio u firmu. Crvena vrednost govori da nije, plava da jeste. Nakon nekog vremena treba da zaposlimo novog kandidata i treba da donesemo odluku da li će on biti pogodan za kolektiv. Pogledaćemo kom bivšem kandidatu je novi kandidat najsličniji. Ovakav način rezonovanja predstavlja situaciju k = 1. Dakle, novi kandidat najsličniji je crvenom slučaju i zaključujemo da se kandidat neće uklopiti. Ukoliko posmatramo pet najbližih suseda dobijamo situaciju koja je prikazana na slici 1. Imamo tri kandidata koja su plava i dva kandidata koja su crvena. Kako ima više plavih, zaključujemo da će novi kandidat biti plavi, odnosno da će se dobro uklopiti u firmu.



Slika 1. Vizualizacija problema zapošljavanja u dve dimenzije

Matematički, algoritam k najbližih suseda se može predstaviti na sledeći način:

$$U(S_0, S_m) = \sum_{i=1}^{n} f\left(S_{0,i}, S_{m,i}\right) * w_i \,, i \ \in \{1, n\}$$

gde je S_0 slučaj (alternativa) čija se udaljenost (U) meri sa ostalim alternativama, S_m slučajevi koji se upoređuju, $f\left(S_{0,i}, S_{m,i}\right)$ sličnost slučajeva po atributu i, w_i težina atributa (ponder) i, n broj atributa i m broj slučajeva (alternativa).

Ključni pojam u algoritmu k najbližih suseda je udaljenost. Postoji mnogo načina da se izmeri udaljenost između dva slučaja (ili dve tačke u prostoru u opštem slučaju). Međutim, ovde ćemo koristiti Euklidsko odstojanja, odnosno kvadratno odstojanje:

$$U(S_0, S_m) = \sqrt{\sum_{i=1}^{n} \left(S_{0,i}, S_{m,i}\right)^2 * w_i \,, i \ \in \{1, n\}}$$

Primetimo da Euklidsko odstojanje radi samo sa numeričkim podacima, što znači da ako u tabeli slučajeva imamo kategoričke atribute, onda ne možemo da koristimo ovo odstojanje. U takvim slučajevima, koristimo preklapanje (eng. *overlap*) ili Gudol 3 (eng. *Goodall3*) mere sličnosti.

$$Preklapanje(Overlap) = \begin{cases} 0 : X = Y \\ 1 : X \ \neq Y \end{cases}$$

$$Gudol3(Goodall3) = \begin{cases} p(X)^2 : X = Y \\ 1 : X \ \neq Y \end{cases}$$

gde je $p(X)$ verovatnoća pojavljivanja stanja X.

Drugi ključni pojam je broj suseda, odnosno parametar k. Izbor parametra k je izuzetno važan i kompleksan korak. Ukoliko izaberemo malu vrednost k, onda možemo doći do situacije da nam šum u podacima (npr. pogrešno doneta odluka) utiče na dalje odluke. Ako izaberemo preveliko k, onda nam slučajevi koji nisu slični slučaju za koji se predviđa ishod utiču na odluku. Jednostavna heuristika je da se izabere najbliži neparni broj od korena broja slučajeva. Treba imati u vidu da ova heurstika ne predstavlja optimalno rešenje i da možda za neke probleme nije odgovarajuća.

Kao ključna ograničenja algoritma k najbližih suseda možemo da navedemo da u osnovnoj verziji ne radi sa nenumeričkim atributima (iako se taj problem rešava primenom neke mere odstojanja koja radi sa kategoričkim atributima[1]). Zatim, određivanje parametra k (broj suseda), u opštem slučaju, predstavlja optimizacioni problem. Na kraju, algoritam nije skalabilan, odnosno sa većim uzorkom algoritam gubi efikasnost.

[1] http://www-users.cs.umn.edu/~sboriah/PDFs/BoriahBCK2008.pdf

Postupak rada algoritma će detaljnije biti opisan kroz primere u domenu višeatributivnog odlučivanja.

Primer 1: Korišćenje Euklidske sličnosti

Za date podatke odrediti najsličniji slučaj traženom slučaju.

	Cena (eur)	Internet	Udaljenost od grada (km)	Čistoća
Zahtev korisnika	40	3	0	5

Tabela slučajeva i ponderi.

Tabela slučajeva	Cena (eur)	Internet	Udaljenost od grada (km)	Čistoća
	min	**max**	**min**	**max**
A1	55	3	0,7	4
A2	65	1	0,4	3
A3	40	0	0,7	4
A4	25	2	4	3
A5	40	1	2	5
A6	36	1	1	3
A7	43	2	3	5
A8	54	2	2	4
A9	63	1	3	3
A10	32	2	5	5
Ponderi	**0,35**	**0,2**	**0,3**	**0,15**

Prvi korak u rešavanju je normalizacija tabele slučajeva i zahteva korisnika. Zahtev korisnika se normalizuje podacima iz tabele slučajeva. Za normalizaciju koristimo L∞ normu.

Kao rezultat dobijamo normalizovan zahtev korisnika.

	Cena (eur)	Internet	Udaljenost od grada (km)	Čistoća
Zahtev korisnika	0,62	1	0	1

Takođe, dobijamo normalizovanu tabelu slučajeva.

Norm. tabela slučajeva	Cena (eur)	Internet	Udaljenost od grada (km)	Čistoća
	min	**max**	**min**	**max**
A1	0,846	1,000	0,140	0,800
A2	1,000	0,333	0,080	0,600
A3	0,615	0,000	0,140	0,800
A4	0,385	0,667	0,800	0,600
A5	0,615	0,333	0,400	1,000
A6	0,554	0,333	0,200	0,600
A7	0,662	0,667	0,600	1,000
A8	0,831	0,667	0,400	0,800
A9	0,969	0,333	0,600	0,600
A10	0,492	0,667	1,000	1,000

Nakon normalizacije, računamo matricu udaljenosti. Za udaljenost koristimo Euklidsko odstojanje za numeričke atribute ili odgovarajuću meru udaljenosti za kategoričke podatke. Što je vrednost veća, to su slučajevi udaljeniji; odnosno što je udaljenost manja, slučajevi su sličniji.

Za alternativu A1 i kriterijum Cena imamo:

$$U(S_{0,cena}, S_{1,cena}) = (0{,}62 - 0{,}846)^2 = 0{,}053$$

Tabela slučajeva	Cena (eur)	Internet	Udaljenost od grada (km)	Čistoća
	min	**max**	**min**	**max**
A1	0,053	0,000	0,020	0,040
A2	0,148	0,444	0,006	0,160
A3	0,000	1,000	0,020	0,040
A4	0,053	0,111	0,640	0,160
A5	0,000	0,444	0,160	0,000
A6	0,004	0,444	0,040	0,160
A7	0,002	0,111	0,360	0,000
A8	0,046	0,111	0,160	0,040
A9	0,125	0,444	0,360	0,160
A10	0,015	0,111	1,000	0,000

Kada smo popunili matricu udaljenosti, računamo koren sume otežane udaljenosti po alternativama. Ovim korakom kompletiramo računanje otežane Euklidske udaljenosti (pogledati formulu sa 4. strane). Za prvu alternativu imamo:

$$U(S_0, S_1) = \sqrt{0{,}053 * 0{,}35 + 0 * 0{,}2 + 0{,}020 * 0{,}3 + 0{,}040 * 0{,}15} = 0{,}175$$

Računanjem za preostale alternative dobijamo sledeće udaljenosti:

	Agregatna udaljenost
A1	**0,175**
A2	0,408
A3	0,460
A4	0,507
A5	0,370
A6	0,355
A7	0,362
A8	0,304
A9	0,515
A10	0,572

Želimo da vidimo koja alternativa je najsličnija traženoj i to je alternativa A1.

Primer 2: Rad sa funkcijama preferencije

Umesto konkretnih vrednosti možemo da računamo udaljenost, odnosno sličnost dva slučaja prema mišljenju DO o toj vrednosti. Za razliku od Promethee metode gde smo poredili svaku alternativu sa svakom alternativom, u ovom slučaju poredimo svaku alternativu sa alternativom koja predstavlja zahtev korisnika.

Za date podatke odrediti najsličniji slučaj traženom slučaju.

	Cena (eur)	Internet	Udaljenost od grada (km)	Čistoća
Zahtev korisnika	40	3	0	5

Funkcije preferencije.

	Cena (eur)	Internet	Udaljenost od grada (km)	Čistoća
Funkcija	3	2	3	1
Parametri	20	1	5	

Tabela slučajeva i ponderi.

Tabela slučajeva	Cena (eur)	Internet	Udaljenost od grada (km)	Čistoća
	min	**max**	**min**	**max**
A1	55	3	0,7	4
A2	65	1	0,4	3
A3	40	0	0,7	4
A4	25	2	4	3
A5	40	1	2	5
A6	36	1	1	3
A7	43	2	3	5
A8	54	2	2	4
A9	63	1	3	3
A10	32	2	5	5
Ponderi	**0,35**	**0,2**	**0,3**	**0,15**

Prvi korak je kreiranje tabele udaljenosti preko preferencija. Za računanje preferencija pogledajte skriptu Modelovanje preferencija.

Udaljenost preko preferencija	Cena (eur)	Internet	Udaljenost od grada (km)	Čistoća
A1	0,75	0	0,14	1
A2	1	1	0,08	1
A3	0	1	0,14	1
A4	0,75	0	0,8	1
A5	0	1	0,4	0
A6	0,2	1	0,2	1
A7	0,15	0	0,6	0
A8	0,7	0	0,4	1
A9	1	1	0,6	1
A10	0,4	0	1	0

Sada, umesto Euklidske udaljenosti računamo otežanu sumu. Razlog zašto ne računamo Euklidsko odstojanje je taj što smo poređenje alternativa već odradili preko Promethee metode, tako da sada samo proveravamo koliko preferiramo alternativu korisnika u odnosu na ostale alternative. Za alternativu A1 imamo:

$$U(S_0, S_1) = 0{,}75 * 0{,}35 + 0 * 0{,}2 + 0{,}14 * 0{,}3 + 1 * 0{,}15 = 0{,}455$$

	Agregatna udaljenost
A1	0,455
A2	0,724
A3	0,392
A4	0,653
A5	0,320
A6	0,480
A7	**0,233**
A8	0,515
A9	0,880
A10	0,440

Nakon izračunate udaljenosti, kao najprihvatljiviju biramo onu alternativu čija je vrednost najbliža nuli. Odnosno, u ovom primeru zaključujemo da je zahtevu korisnika najpribližnija alternativa A7.

Primer 3: Rad sa kategoričkim podacima

Za date podatke odrediti najsličniji slučaj traženom slučaju. Za udaljenost kategoričkih podataka koristiti Gudol 3 meru odstojanja.

Tabela slučajeva	Cena (eur)	Internet	Udaljenost od grada (km)	Čistoća	Stil hotela
	min	**max**	**min**	**max**	
A1	55	3	0,7	4	Tradicionalni
A2	65	1	0,4	3	Fensi
A3	40	0	0,7	4	Tradicionalni
A4	25	2	4	3	Fensi
A5	40	1	2	5	Tradicionalni
A6	36	1	1	3	Fensi
A7	43	2	3	5	Fensi
A8	54	2	2	4	Hipster
A9	63	1	3	3	Hipster
A10	32	2	5	5	Poslovni
Ponderi	**0,3**	**0,2**	**0,3**	**0,1**	**0,1**

Zahtev korisnika je:

	Cena (eur)	Internet	Udaljenost od grada (km)	Čistoća	Stil hotela
Zahtev korisnika	40	3	0	5	Hipster

Prvi korak je normalizacija tabele slučajeva. Za normalizaciju koristimo L∞ normu.

Kao rezultat dobijamo normalizovan zahtev korisnika. Kako je atribut „Stil hotela“ kategorički podatak, njega ne normalizujemo.

	Cena (eur)	Internet	Udaljenost od grada (km)	Čistoća	Stil hotela
Zahtev korisnika	0,62	1	0	1	Hipster

Takođe, dobijamo normalizovanu tabelu slučajeva. Računanje sličnosti za “Stil hotela” se radi na sledeći način. Ako se vrednosti (kategorije) razlikuju onda je sličnost jednaka 1. Međutim ako su iste onda računamo preko formule sa 4. strane. Za alternativu A8 dobijamo:

$$U(S_{0,stil\_hotela}, S_{8,stil\_hotela}) = \left(\frac{2}{10}\right)^2 = 0{,}2^2 = 0{,}04$$

U tabeli slučajeva imamo dve alternative koje imaju vrednost “Hipster” od ukupno deset slučajeva. Dakle, verovatnoća da postoji “Hipster” stil hotela je 0,2. Kvadriramo tu vrednost i nju prikazujemo kao sličnost. Kako je maksimalna vrednost 1, nema potrebe za daljom normalizacijom ove kolone.

Norm. tabela slučajeva	Cena (eur)	Internet	Udaljenost od grada (km)	Čistoća	Stil hotela
	min	**max**	**min**	**max**	
A1	0,846	1,000	0,140	0,800	1
A2	1,000	0,333	0,080	0,600	1
A3	0,615	0,000	0,140	0,800	1
A4	0,385	0,667	0,800	0,600	1
A5	0,615	0,333	0,400	1,000	1
A6	0,554	0,333	0,200	0,600	1
A7	0,662	0,667	0,600	1,000	1
A8	0,831	0,667	0,400	0,800	0,04
A9	0,969	0,333	0,600	0,600	0,04
A10	0,492	0,667	1,000	1,000	1

Računanje Gudol3 udaljenosti ima sledeće tumačenje. Ukoliko u tabeli slučajeva ima dosta pojavljivanja neke vrednosti, onda će vrednost udaljenosti biti velika. Ako je pojavljivanje te kategorije često, onda posedovanje te vrednosti ne pravi nikakvu razliku. Sa druge strane, ukoliko imamo mali broj pojavljivanja neke kategorije u tabeli odlučivanja, tada posedovanje te kategorije treba da se prikaže kao veća sličnost, jer manje slučajeva ima tu osobinu.

Kada imamo izračunate vrednosti, odnosno popunjenu matricu udaljenosti, računamo agregatnu udaljenost preko Euklidskog odstojanja. Kao najprihvatljiviju alternativu biramo onu koja ima najmanju udaljenost.

	Agregatna udaljenost
A1	0,3548
A2	0,5012
A3	0,5567
A4	0,5884
A5	0,4867
A6	0,4669
A7	0,4805
A8	**0,3035**
A9	0,5044
A10	0,6533

Kao najprihvatljiviju alternativu biramo alternativu A8.

Primer 4: k-NN za predviđanje nedostajućih vrednosti

Pri rešavanju problema odlučivanja, donosioci odluka se vrlo često sreću sa situacijom da nije moguće oceniti sve kriterijume za pojedine alternative. Na primer: pri izboru novog zaposlenog, nije moguće svima utvrdi nivo poznavanja engleskog jezika, jer neki od njih nisu taj podatak naveli u svom radnom rezimeu. Još jedan primer bi bio da je novi automobil izašao na tržište, ali da mu još uvek nije formirana cena iako su poznate karakteristike. Pretpostavimo da DO želi da kupi auto i da su mu na raspolaganju alternative i kriterijumi dati u tabeli ispod.

	Cena	Enterijer	Snaga
Dacia Sandero	7500	1	75
Chevrolet Aveo	9000	3	86
Honda Civic		5	140
Hyundai Elantra	14000	5	120
Porsche Panamera	120000	9	320

Takođe pretpostavimo da nije poznata prodajna cena za alternativu Honda Civic i da će se cena saznati u roku od dve nedelje, kada je i potrebno da se donese odluka o željenom automobilu. Jedan od načina da se oceni nedostajuća vrednost koji se često primenjuje pri donošenju odluka jeste **procena na osnovu aritmetičke sredine svih vrednosti kriterijuma**. Intuicija ovog pristupa je da će greška u proceni (odstupanje stvarnih vrednosti od procenjenih) biti najmanja ukoliko se koristi aritmetička sredina. Ova intuicija daje relativno dobra rešenja u situacijama kada kriterijum ima **Normalnu raspodelu** (najveći deo populacije uzima aritmetičku sredinu). Međutim, kod ovog pristupa postoje dva velika nedostatka zbog kojih se najčešće dolazi do loših procena:

- ovaj pristup ne uzima u obzir sve dostupne karakteristike (kriterijume) i
- aritmetička sredina je veoma osetljiva na ekstremne vrednosti.

U našem primeru, procenjena cena Honde Civic na osnovu aritmetičke sredine bi iznosila 37625 eur: (7500+9000+14000+120000)/4. U praksi, ovaj automobil najčešće košta između 15000 i 20000. Ukoliko detaljnije pogledamo početnu tabelu odlučivanja, možemo primetiti da tri alternative imaju cenu između 7500 i 14000, dok je poslednja alternativa skuplja za jedan red veličine (Cena Porsche Panamere je 120000). Ova drastična razlika u ceni dovela je do toga da se aritmetička sredina celog kriterijuma značajno uveća u odnosu na većinu alternativa.

U ovakvim situacijama k-nn može da postigne daleko veću tačnost procene, korišćenjem koncepta sličnosti i svih dostupnih atributa, kao i istorijskih podataka koji su dostupni. Pretpostavimo da DO ima bazu podataka automobila gde se vodi evidecija o svim kriterijumima koji su mu potrebni za odlučivanje („Cena“, „Enterijer“ i „Snaga“).

	Cena	Enterijer	Snaga
Renault Clio	12000	5	95
Opel Corsa	11000	3	105
Mazda 3	16000	5	110
Toyota Corolla	14000	5	110
Opel Insignia	22000	5	120
BMW 3	32000	7	140
BMW 5	45000	7	180
L∞	**45000**	**7**	**180**

Honda Civic		**5**	**140**

Dakle, DO želi da iskoristi dostupne vrednosti kriterijuma za alternativu Honda Civic i da proceni njenu cenu na osnovu najsličnijih slučajeva u dostupnoj bazi automobila za koje su poznati svi kriterijumi (tabela iznad).

Bitno je napomenuti da je **pri svakom modelovanju u kome se koristi koncept sličnosti/udaljenosti neophodno izvršiti normalizaciju podataka**, kako jedinice mere pojedinih kriterijuma ne bi uticale na ukupno odstojanje/sličnost. Zbog toga vršimo normalizaciju ulaznih kriterijuma (kriterijuma na osnovu kojih se meri sličnost/udaljenost, u ovom slučaju „Enterijer“ i „Cena“). Jasno je da je neophodno normalizovati sve slučajeve u bazi podataka, kao i novom slučaju za koji je potrebno odrediti cenu. U tabeli ispod su prikazane normalizovane vrednosti (L_∞ normom) kriterijuma „Cena“ i „Snaga“. Kriterijum „Cena“ nije normalizovan jer se ne koristi pri određivanju odstojanja (nije dostupan za novi slučaj) i DO želi da vidi originalne vrednosti procenjene cene.

	Cena	Enterijer	Snaga
Renault Clio	12000	0.714	0.528
Opel Corsa	11000	0.429	0.583
Mazda 3	16000	0.714	0.611
Toyota Corolla	14000	0.714	0.611
Opel Insignia	22000	0.714	0.667
BMW 3	32000	1.000	0.778
BMW 5	45000	1.000	1.000
Težine		**0.3**	**0.7**
Honda Civic		**0.714**	**0.778**

Kao što je objašnjeno ranije, odstojanje je moguće otežati (ponderisati), kako bi se modelovao značaj sličnosti određenih kriterijuma. U ovom primeru pretpostavljeno je da „Snaga“ utiče 70% a „Enterijer“ 30% na konačnu cenu automobila.

Korišćenjem otežanog Euklidskog odstojanja, DO računa odstajanje novog slučaja (Honda Civic) u odnosu na sve ostale slučajeve u bazi.

	Cena	Enterijer	Snaga	**Odstojanje**
Renault Clio	12000	0.714	0.528	**0.209**
Opel Corsa	11000	0.429	0.583	**0.226**
Mazda 3	16000	0.714	0.611	**0.139**
Toyota Corolla	14000	0.714	0.611	**0.139**
Opel Insignia	22000	0.714	0.667	**0.093**
BMW 3	32000	1.000	0.778	**0.156**
BMW 5	45000	1.000	1.000	**0.243**

Nakon određivanja odstojanja potrebno je odrediti koji slučajevi imaju najmanje odstojanje od novog slučaja i na osnovu njih odrediti nedostajući kriterijum („Cena“). Kako bismo lakše

analizirali rešenje, sortiraćemo alternative (modele) i njihove cene u opadajućem poretku u odnosu na odstojanje od novog slučaja.

Odstojanje	Model	Cena
0.093	Opel Insignia	22000
0.139	Mazda 3	16000
0.139	Toyota Corolla	14000
0.156	BMW 3	32000
0.209	Renault Clio	12000
0.226	Opel Corsa	11000
0.243	BMW 5	45000

Iz tabele iznad vidimo da je nasličniji slučaj sa Hondom Civic, Opel Insignia sa cenom od 22000 eur. Zatim slede Mazda 3 i Toyota Corolla sa cenama 16000 i 14000 eur, respektivno. Kako bi DO procenio konačnu cenu automobila, neophodno je da odredi broj najbližih suseda na osnovu kojih će odrediti prosečnu cenu. DO posmatra prosečne cene za k=1, k=2 i k=3.

k	Cena
1	22000
2	19000
3	17333.3333

Vidimo, iz tabele iznad, da ukoliko posmatramo samo jednog najbližeg suseda, procenjena cena Honde Civic bi bila 22000 eur. Ukoliko posmatramo dva najbliža suseda, cena bi bila 19000: (22000+16000)/2, a ako bismo posmatrali tri najbliža suseda, bila bi 17333.3333: (22000+16000+14000)/3.

DO se odlučuje za K=3, odnosno, procenjenu cenu od 17333.3333 eur. Konačno, DO može da popuni procenjenu vrednost cene za Hondu Civic i vrati se inicijalnom problemu odlučivanja.

	Cena	Enterijer	Snaga
Dacia Sandero	7500	1	75
Chevrolet Aveo	9000	3	86
Honda Civic	**17333.333**	5	140
Hyundai Elantra	14000	5	120
Porcshe Panamera	120000	9	320

Nakon invertovanja i normalizacije L1 normom, DO računa očekivanu korist i bira najbolju alternativu:

	Cena	Enterijer	Snaga	OK
Dacia Sandero	0.349	0.043	0.101	0.214
Chevrolet Aveo	0.291	0.130	0.116	0.206
Honda Civic	0.151	0.217	0.189	0.176
Hyundai Elantra	0.187	0.217	0.162	0.186
Porsche Panamera	0.022	0.391	0.432	0.219
Težine	0.5	0.2	0.3	

Nakon donošenja odluke, DO je saznao pravu cenu Honde Civic koja je iznosila 17000 eur. To znači da je **njegova greška procene iznosila** 333.333 eura (17333.333-17000). U tabeli ispod su date greške procene ukoliko bi se odlučio za k=1, k=2, k=3 ili za procenu na osnovu aritmetičke sredine (AS).

k/AS	Cena
1.000	5000 (22000-17000)
2.000	2000 (19000-17000)
3.000	333.333 (17333.333-17000)
AS	4714.286 (21714.286-17000)

Možemo primetiti da je minimalna greška nastala kada je korišćena k-NN metoda i kada je k=3. Takođe se iz tabele može videti da je uključivanje dodatnih kriterijuma drastično popravilo tačnost procenjene vrednosti u odnosu na procenu uz pomoć aritmetičke sredine.