

Mašinsko učenje

KNN

Klasifikacija

- Da li klasifikator dobro radi?
- Koliko puta se klasifikuje ispravno i pogrešno
- True positive - pogodak: „DA“ tačno prediktovan kao „DA“.
- True negative: „NE“ nije tačno prediktovan kao „NE“
- False positive: „NE“ nije predviđen kao „DA“.
- False negative - promašaj: „DA“ je predviđen kao „NE“.

		Actual class	
		Yes	No
Predictd class	Yes	TP – true positive	FP– false positive
	No	FN – false negative	TN– true negative

Performanse klasifikacije

Грешка класификације:

$$\text{Classification error} = \frac{\text{errors}}{\text{totals}} = \frac{FP + FN}{FP + FN + TP + TN}$$

Тачност класификације:

$$\text{Classification accuracy} = 1 - \text{error} = \frac{\text{correct}}{\text{totals}} = \frac{TP + TN}{FP + FN + TP + TN}$$

- Главни problem ove dve mere - ne mogu da podnesu neuravnotežene klase.
- Da li je transakcija kreditnom karticom zloupotreba ili ne.
- 99,99% bez zloupotrebe i samo mali procenat zloupotreba
- Klasifikator koji kaže da je svaka transakcija normalna je tačan 99,99%
- Krajnjeg korisnika uglavnom zanima onih nekoliko 0,01% klasifikacija

Performanse klasifikacije

- Koristiti mere koje ne uključuju true negative.
- **Preciznost** (eng. precision):

$$\text{Precision} = \frac{TP}{TP + FP}$$

- Jednaka je količniku broja pozitivnih primeraka koji su tačno prediktovani (TR) od svih ostalih prediktovani kao pozitivni (TP + FP)
- **Opoziv** (eng. recall): Jednaka je količniku broju primeraka tačno prediktovanih kao pozitivni (TR) od svih ostalih pozitivnih primeraka:

$$\text{Recall} = \frac{TP}{TP + FN}$$

Performanse klasifikacije

- Često se kombinuju ove dve mere u F-meru (eng. F-measure) koja uzima u obzir istovremeno preciznost i opoziv da se izračuna skor kao težinsku sredinu, gde skor dostiže najbolju vrednost 1 i najgoru 0:

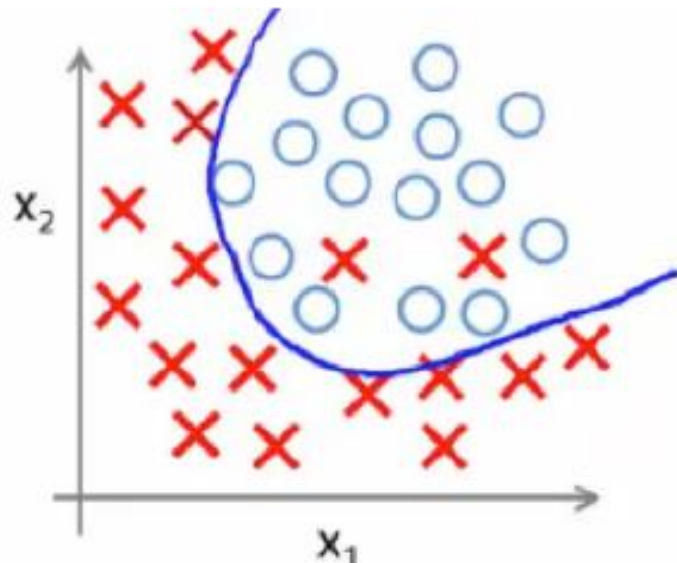
$$F - measure = \frac{2 * Precision * Recall}{Precision + Recall}$$

Problem prevelike podešenosti (overfitting)

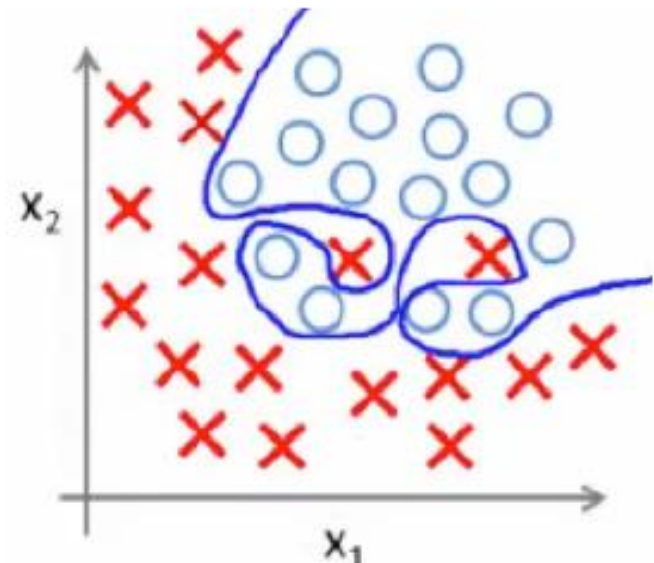
- Situacija kada model savršeno nauči da prepozna instance iz trening skupa, ali nije u mogućnosti da prepozna instance koje se *i malo razlikuju* od naučenih
 - Training skup je nužno nepotpun i ne uključuje buduće podatke koje želimo da klasifikuje.
 - Algoritam treba da bude “imun” na pamćenje celog trening skupa (već samo generalnog znanja)

Primer problema prevelike podešenosti modela

- Ako svi klijenti sa imenom “David” u trening skupu imaju visoka primanja
- => “IF (first name == “David”) THEN the customer has a high income”



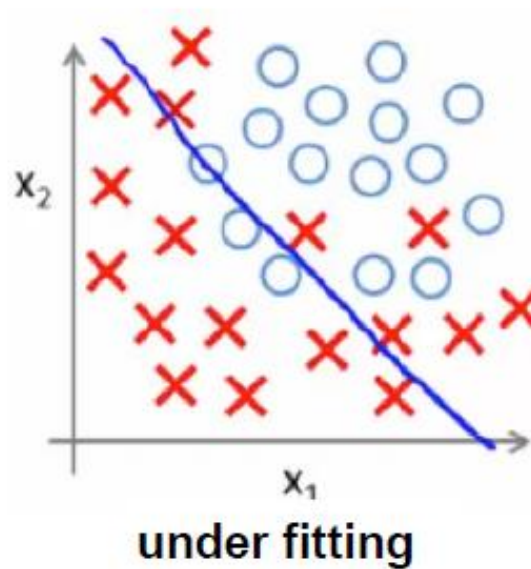
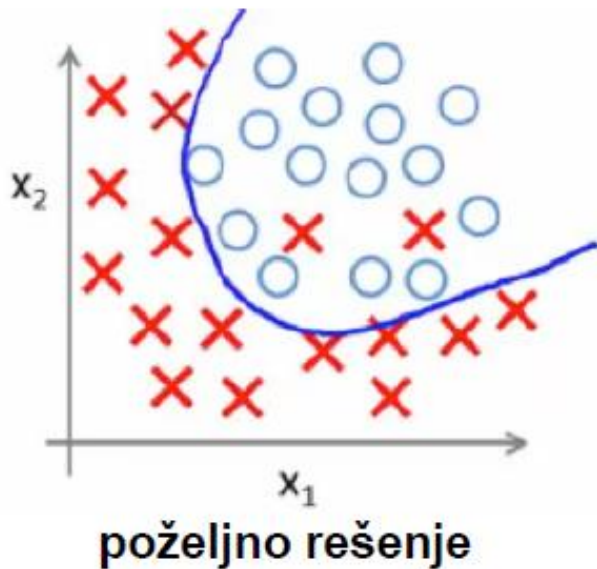
poželjno rešenje



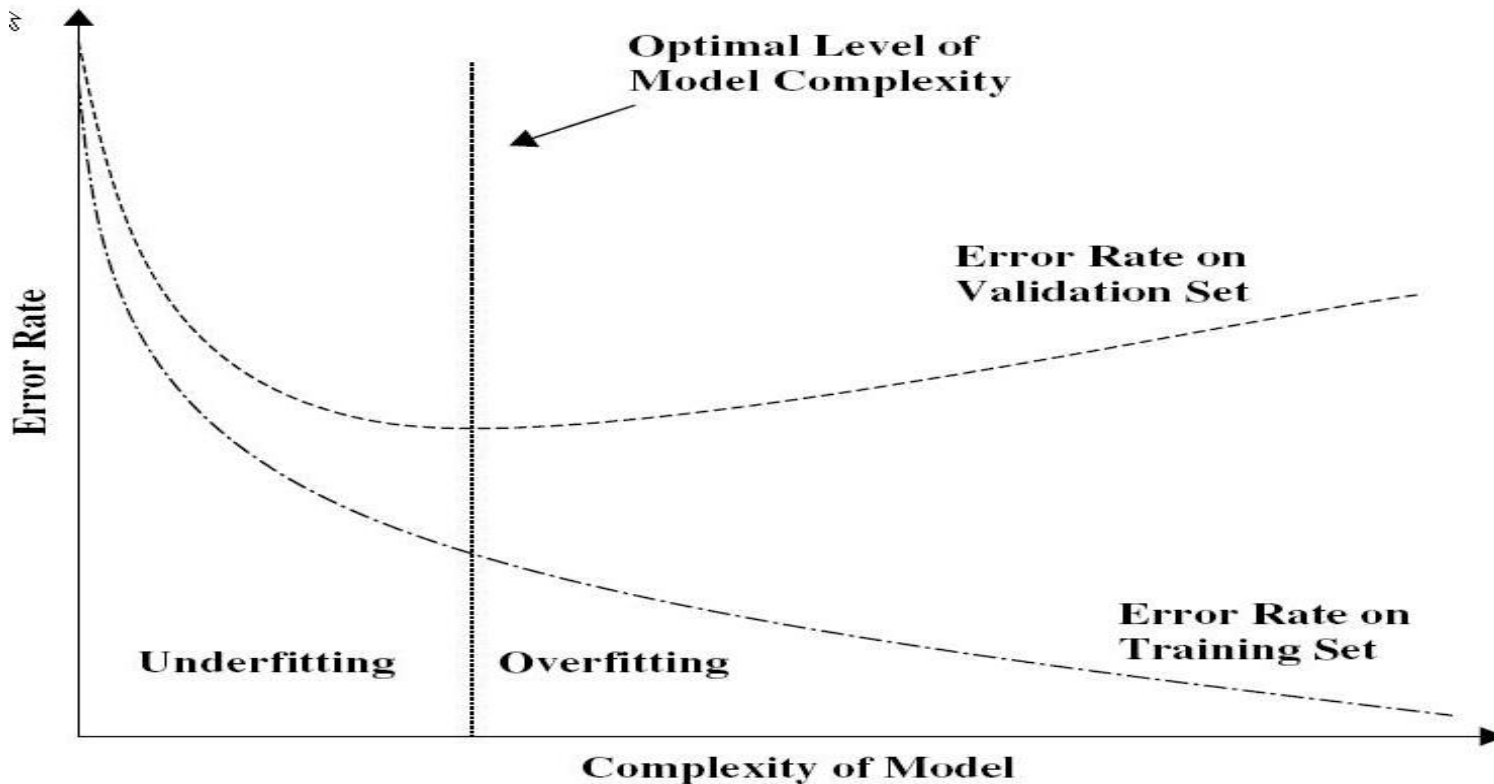
over fitting

Problem nedovoljne podešenosti (underfitting)

- slučaj kad model ne uspeva da aproksimira podatke iz trening skupa, tako da ima slab učinak čak i na trening skupu



Underfitting vs. Overfitting



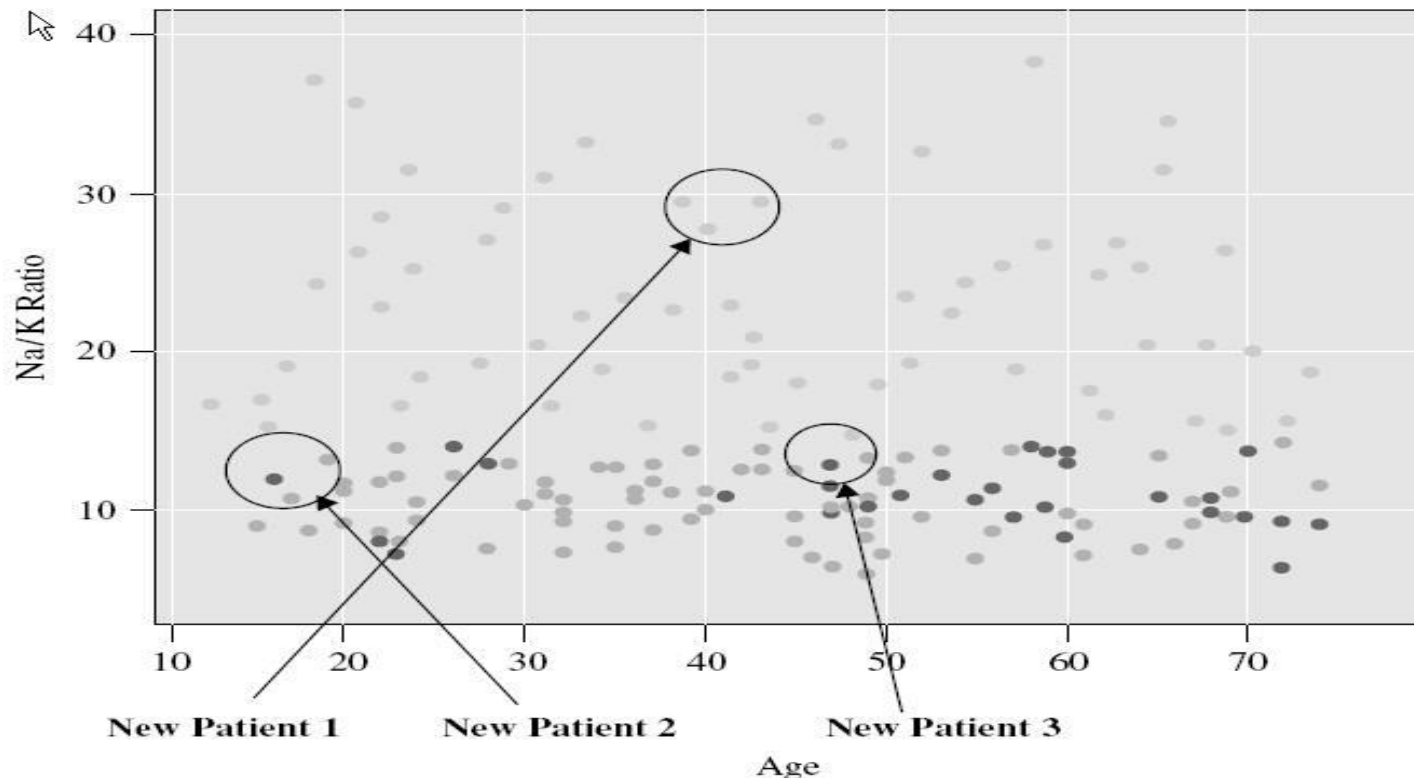
Optimalni nivo kompleksnosti modela je na minimalnoj stopi greške na validacionom skupu

- Naziva se još i odnosom pristrastnosti-varijanse (bias/variance)
- Preterano povećanje kompleksnosti modela dovodi do degradacije njegove generalnosti.

Algoritam k-najbližih suseda

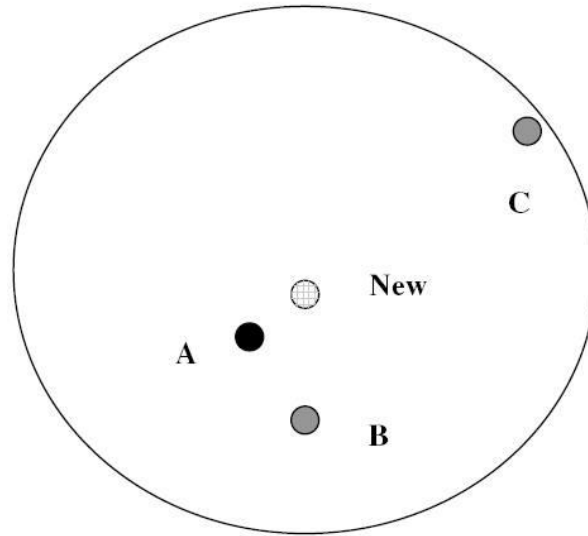
k-Nearest Neighbor (k-NN)

- Nadgledano učenje - primer učenja na bazi instanci (*Instance-based learning*)
- Klasifikacija za novi primer vrši se upoređivanjem sa najbližim primerima u trening skupu. (S kim si, takav si 😊)



Na/K odnos nasuprot broja godina, sa vrstom prepisanog leka

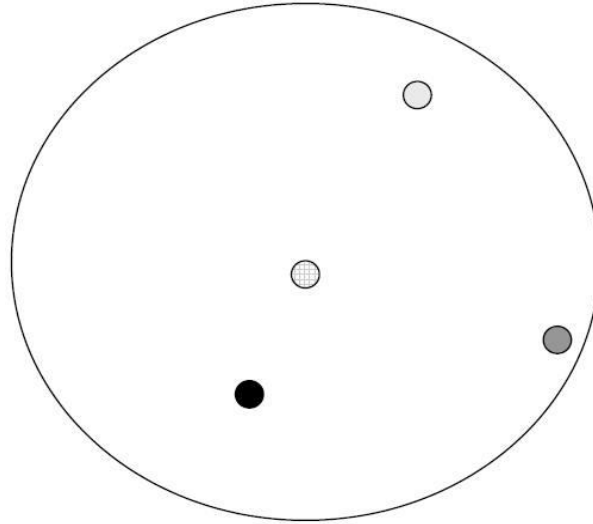
Algoritam k-najbližih suseda



Tri najbliža suseda za pacijenta 2

- Pacijent 2 je 17 godina star i ima Na/K odnos 12.5.
- Za različito $k = 1, 2, 3$ menja se klasa dodeljena ovom pacijentu

Algoritam k-najbližih suseda



Tri najbliža suseda za pacijenta 3

- Pacijent 3, star 47 godina i ima Na/K odnos 13.5.
- Za $k = 2$ ili 3 obično glasanje ne pomaže.

Algoritam k-najbližih suseda

- Pitanja na koja treba odgovoriti:
 - Kako se meri distanca?
 - Kako se kombinuju (agregiraju) vrednosti više instanci?
 - Da li sve tačke treba da imaju istu težinu, ili neke tačke treba da imaju više uticaja od drugih?
 - Koliko suseda treba razmatrati, tj. koliko je k ?

Funkcija rastojanja

- Za novi primer, algoritam k-najbližih suseda dodeljuje klasu najbližnje instance
- Kako definišemo slično?
- Najčešće se koristi Euklidsko rastojanje:

$$d_{\text{Euclidean}}(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_i (x_i - y_i)^2}$$

- Pri čemu $x = x_1, x_2, \dots, x_m$, i $y = y_1, y_2, \dots, y_m$ su vrednosti atributa dve instance.

Funkcija rastojanja

- Atributi mogu imati vrednosti u opsezima koji se značajno mogu razlikovati.
 - Npr. godine i prihodi
- Da bi se ovo izbeglo koristi se normalizacija (vrsta transformacije podataka)
- Za kategoričke vrednosti Euklidsko rasojanje nije odgovarajuće
- Definiše se funkcija različitosti (different) kako bi se uporedila dva atributa *nara instanci*.

$$\text{different}(x_i, y_i) = \begin{cases} 0 & \text{if } x_i = y_i \\ 1 & \text{otherwise} \end{cases}$$

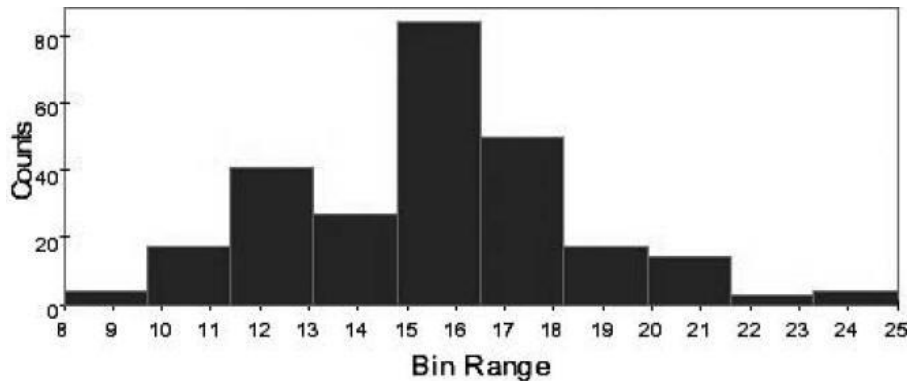
gde su x_i i y_i kategoričke vrednosti.

Transformacija podataka

- Za neke algoritme razlike u opsegu mogu dovesti do tendencija da atributi sa većim opsegom imaju *veći uticaj na rezultate* (k-NN).
- Rešenje -> normalizovati numeričke vrednosti, kako bi se standardizovale razmere efekta koji svaki atribut ima na krajnji rezultat.
- **Min-Max Normalizacija**

$$X^* = \frac{X - \min(X)}{\text{range}(X)} = \frac{X - \min(X)}{\max(X) - \min(X)}$$

Transformacija podataka - primer



Count:	261
Missing:	0
Max:	25.0
Min:	8.0
Mean:	15.548
Std dev:	2.911

$$X^* = \frac{X - \min(X)}{\max(X) - \min(X)} = \frac{8 - 8}{25 - 8} = 0$$

$$X^* = \frac{X - \min(X)}{\max(X) - \min(X)} = \frac{15.548 - 8}{25 - 8} = 0.444$$

$$X^* = \frac{X - \min(X)}{\max(X) - \min(X)} = \frac{25 - 8}{25 - 8} = 1.0$$

Min-Max Normalizacija

- Normalizovane vrednosti su u opsegu $[0,1]$
 - Osim ako novi podaci ne predstavljaju ekstremne vrednosti koje leže van originalnog opsega (*Outliers*)
- Takođe, ekstremne vrednosti mogu učiniti da ostale vrednosti budu blizu ekstrema (0 ili 1)
 - Npr. prosek primanja od 100.000€ može učiniti ostale normalizovane vrednosti bliskim 0.
- Potencijalan problem – Min-max normalizacija zavisi od opsega.

Funkcija rastojanja - primer

Patient	Age	Age _{MMN}	Gender
A	50	$\frac{50 - 10}{50} = 0.8$	Male
B	20	$\frac{20 - 10}{50} = 0.2$	Male
C	50	$\frac{50 - 10}{50} = 0.8$	Female

Vrednosti atributa za broj godina (Age) i pol (Gender)

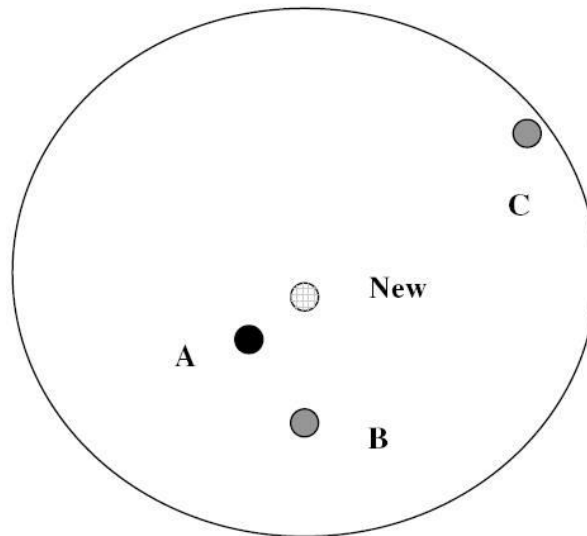
- Koji je pacijent sličniji pacijentu A?
- Ako ne bismo izvršili normalizaciju:
 $d(A,B)^1 = \sqrt{(50 - 20)^2 + 0^2} = 30$
 $d(A,C) = \sqrt{(50 - 50)^2 + 1^2} = 1$
- Sa MMN:
 $d_{MMN}(A,B) = \sqrt{(0.8 - 0.2)^2 + 0^2} = 0.6$
 $d_{MMN}(A,C) = \sqrt{(0.8 - 0.8)^2 + 1^2} = 1.0$

Funkcija agregacije

- Kako kombinovati slične instance za određivanje klasifikacije?

Netežinsko glasanje:

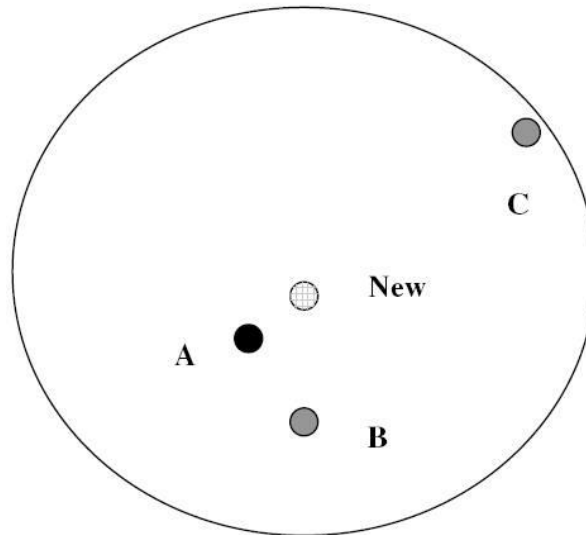
- U ovom primeru klasifikacija će imati pouzdanost $2/3 = 66.67\%$,
- Nivo pouzdanosti je određen brojem instanci “pobedničke” klase, podeljen sa k.



Funkcija agregacije

Težinsko glasanje

- Susedi koji su bliži imaju veću težinu



Funkcija agregacije – primer

Record	Age	Na/K	Age _{MMN}	Na/K _{MMN}
New	17	12.5	0.05	0.25
A (dark gray)	16.8	12.4	0.0467	0.2471
B (medium gray)	17.2	10.5	0.0533	0.1912
C (medium gray)	19.5	13.5	0.0917	0.2794

Godine i Na/K odnosi

- Rastojanja:

$$d(\text{new}, A) = \sqrt{(0.05 - 0.0467)^2 + (0.25 - 0.2471)^2} = 0.004393$$

$$d(\text{new}, B) = \sqrt{(0.05 - 0.0533)^2 + (0.25 - 0.1912)^2} = 0.58893$$

$$d(\text{new}, C) = \sqrt{(0.05 - 0.0917)^2 + (0.25 - 0.2794)^2} = 0.051022$$

Funkcija agregacije – primer

- Broj glasova je obrnuto proporcionalan kvadratu rastojanja.
- Za instancu A:

$$\text{votes (dark gray)} = \frac{1}{d(\text{new}, A)^2} = \frac{1}{0.004393^2} \simeq 51817.63$$

- Za B i C:

$$\begin{aligned} \text{votes (medium gray)} &= \frac{1}{d(\text{new}, B)^2} + \frac{1}{d(\text{new}, C)^2} = \frac{1}{0.058893^2} + \frac{1}{0.051022^2} \\ &\simeq 672 \end{aligned}$$

- Tamno sivi (dark gray) – pobednik !!!

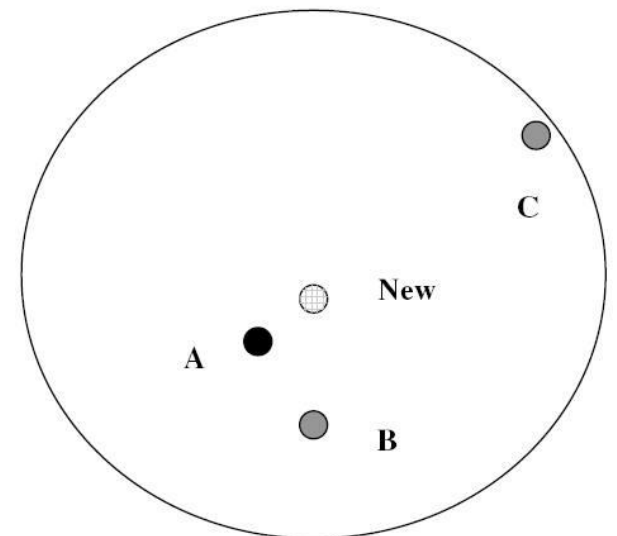
Kvantifikovanje relevantnosti atributa

- Nisu svi atributi podjednako relevantni za klasifikaciju.
- Određivanje atributa koji su više ili manje relevantni svodi se na pronalaženje koeficijenta z_j kojim se množi osa j
 - Veća vrednost $z_j \Rightarrow$ vrednosti tog atributa imaju veći uticaj.
- Primer: $z_{Na/K} = 3$ and $z_{Age} = 1$.

$$d(\text{new}, A) = \sqrt{(0.05 - 0.0467)^2 + [3(0.25 - 0.2471)]^2} = 0.009305$$

$$d(\text{new}, B) = \sqrt{(0.05 - 0.0533)^2 + [3(0.25 - 0.1912)]^2} = 0.17643$$

$$d(\text{new}, C) = \sqrt{(0.05 - 0.0917)^2 + [3(0.25 - 0.2794)]^2} = 0.09756$$



k-NN za regresiju

Metod lokalnog težinskog usrednjavanja

(*Locally weighted averaging*):

$$\hat{y}_{\text{new}} = \frac{\sum_i w_i y_i}{\sum_i w_i}$$

- Težinska srednja vrednost k-najbližih suseda
- W_i – recipročna vrednost kvadrata rastojanja od i-tog suseda
- Y_i – vrednost ciljnog atributa i-tog suseda

k-NN regresija – primer

Record	Age	Na/K	BP	Age _{MMN}	Na/K _{MMN}
New	17	12.5	?	0.05	0.25
A	16.8	12.4	120	0.0467	0.2471
B	17.2	10.5	122	0.0533	0.1912
C	19.5	13.5	130	0.0917	0.2794

$k = 3$ NN algoritam za regresiju

- Procena nivoa krvnog pritiska pacijenta – u opsegu [80,160]
- $z_{\text{Na/K}} = 3$

$$\hat{y}_{\text{new}} = \frac{\sum_i w_i y_i}{\sum_i w_i} = \frac{\frac{120}{0.009305^2} + \frac{122}{0.17643^2} + \frac{130}{0.09756^2}}{\frac{1}{0.009305^2} + \frac{1}{0.17643^2} + \frac{1}{0.09756^2}} = 120.0954.$$

Odabir k

- Za malo k (npr. $k = 1$) algoritam će vraćati ciljanu vrednost najbližeg suseda
 - Problem overfitting-a (nedostatak generalnosti)
- Ako je k preveliko, previdećemo lokalno ponašanje
- k birati tako da bude “odlučivo” (3,5...)
- Odabrati vrednost k tako da se *minimizuje greška nad validacionim skupom*.

Upotreba k-NN algoritma

- Može se koristiti i za regresiju i za klasifikaciju (sličan K-means algoritam se koristi za grupisanje)
- Kolaborativno filtriranje (Collaborative filtering) je primer upotrebe k-NN
 - Pronađi k najbližnjih ljudi korisniku x koji su ocenili film y
 - Predvidi dopadanja filma y za korisnika x kao srednju vrednost svih y_k

Implementacija

- Retke klasifikacije moraju biti dovoljno zastupljene
 - kako algoritam ne bi predviđao samo uobičajene klasifikacije.
- S druge strane, velika količina podataka može da iziskuje značajno vreme za obradu (zbog računanja rastojanja)
- Ubrzanja:
 - Grupisanje: npr. pomoću LSH (Locality Sensitive Hashing) algoritma
 - Odsecanje trening skupa: u ovom primeru sve instance sa $N_a/K > 19$ mogu se izostaviti, jer su klasifikovane na isti način

