Uvod u znanost o podacima

Uvod u nadzirano strojno učenje

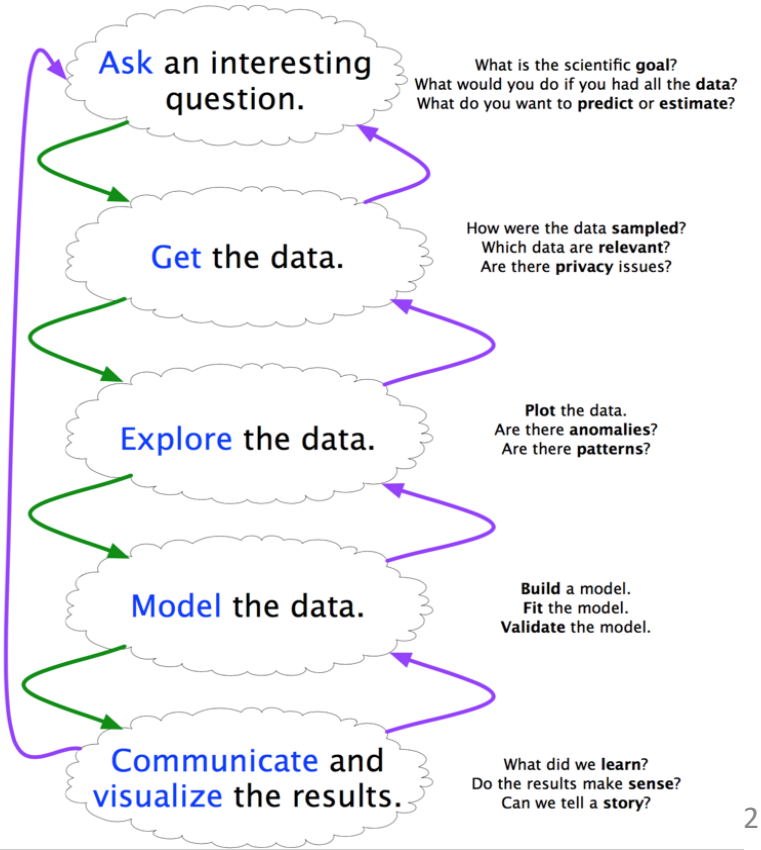
Prof. dr. sc. Bojana Dalbelo Bašić

7. predavanje, 16. studenog 2021.

ak. god. 2021./2022.

Zašto je SU dio znanosti o podacima?

- SU može doprinijeti većini koraka u ciklusu analize podataka



Sadržaj

- Uvod i primjeri
- Uvod u algoritam k najbližih susjeda (k-NN) i dilema pristranost-varijanca
- Stabla odluke, slučajne šume i *boosted trees*
- Linearna i logistička regresija
- O prenaučenosti

Strojno učenje

- **Nadzirano:** Dani su parovi ulaz/izlaz (X, y) (tj. uzorak) pomoću kojih tražimo funkciju $y = f(X)$. Naučenu funkciju f evaluiramo na novim podacima. Vrste:
  - **Klasifikacija:** output y je diskretan (oznake klasa)
  - **Regresija:** output y je kontinuiran (linearna regresija)

Strojno učenje

- **Nadzirano:** Dani su parovi ulaz/izlaz (X, y) (tj. uzorak) pomoću kojih tražimo funkciju y = f(X). Naučenu funkciju f evaluiramo na novim podacima. Vrste:
  - **Klasifikacija:** output y je diskretan (oznake klasa)
  - **Regresija:** output y je kontinuiran (linearna regresija)

- **Nenadzirano:** Dani samo podaci X, oblikujemo funkciju f tako da je y = f(X) *jednostavnija* reprezentacija podataka.
  - Diskretan y: grupiranje
  - Kontinuirani y: redukcija dimenzionalnosti

Strojno učenje: primjeri

- **Nadzirano učenje (*predavanje 7, i.e., danas*):**
  - Da li je na ovoj slici mačka ili pas ili kuća ..?
  - Kako bi ovaj korisnik rangirao ovaj restoran?
  - Da li je ovaj mail *spam*?
  - Da li je ova mrlja na slici supernova?

  Nenadzirano (*predavanje 9*):
  - Grupiraj rukom pisane znakove u 10 klasa.
  - Koje su top 20 tema na Twitteru baš sada?
  - Nađi najbolju 2D vizualizaciju 1000-dimenzijskih podataka.

Strojno učenje: primjeri

- **Nadzirano učenje (*predavanje 7, i.e., danas*):**
  - Da li je na ovoj slici mačka ili pas ili kuća ..?
  - Kako bi ovaj korisnik rangirao ovaj restoran?
  - Da li je ovaj mail *spam*?
  - Da li je ova mrlja na slici supernova?

PRIMJER

Cilj: pokazati učinak kombiniranja različitih metoda strojnog učenja, dubokog učenja i eksplorativnih tehnika

Predicting News Values from Headline Text and Emotions

Maria Pia di Buono[1] Jan Šnajder[1] Bojana Dalbelo Bašić[1]
Goran Glavaš[2] Martin Tutek[1] Natasa Milic-Frayling[3]

[1] TakeLab, Faculty of Electrical Engineering and Computing, University of Zagreb,Croatia

[2] Data and Web Science Group, University of Mannheim, Germany

[3] School of Computer Science, University of Nottingham, UK

NLP meets Journalism 2017

SVM i CNN

SVM - powerful discriminative model (OVR)

- assume additive compositionality of word embeddings - 300 dimensional vector
- RBF kernel nested 5 x 5 cross-validation model for C and γ optimisation

CNN – feed forward NN with one or more convolutionary layers

- we had one single convolutionary and one pooling layer. 64 filters, top k pooling with k=2
- CNN detect indicative word sequences

Comments on the results

	SVM		CNN	
News value	T	T+E	T	T+E
Bad news	0.652	0.763*	0.778†	**0.848*†**
Celebrity	**0.553**	0.534	0.496	0.526
Conflict	0.526	0.487	0.654†	**0.659†**
Drama	0.636	0.637	0.668	**0.681**
Entertainment	0.783	0.832*	0.803	**0.841***
Good news	0.414	0.513	0.509	**0.578**
Magnitude	0.299	**0.515***	0.438	0.507
Power elite	0.596	0.570	0.695†	**0.700†**
Shareability	0.309	0.318	**0.427†**	0.425†

Bad news and *Entertainment* easiest to predict –
highest values in all models (can we explain why?)

Exploratory Data Analysis

Answer to the first research question:

What is the relation between news values conveyed by headlines and emotions?

Factor analysis on 21 variable:

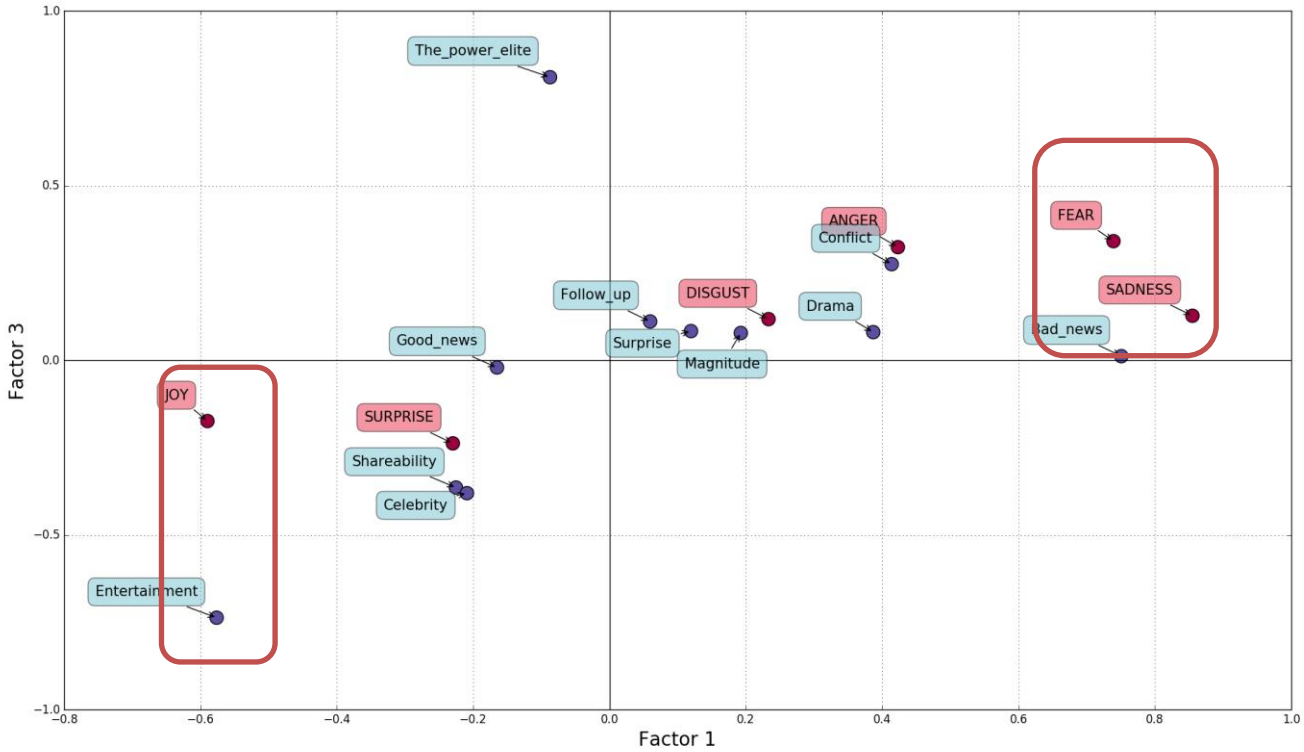
13 newsvalues + 6 emotions

FA is an explorative technique that reveals hidden patterns of data and summarises data in a smaller number of dimensions (factors)

One way of looking at these results is FA ...

Factor Structure (Correlations)							
	Factor1	Factor2	Factor3	Factor4	Factor5	Factor6	Factor7
Bad_news	0.75091	0.12346	0.01188	-0.05998	-0.00307	0.01029	-0.04510
sadness	0.85518	0.34884	0.12756	-0.16657	0.27210	0.03152	0.12622
fear	0.73940	0.46652	0.34112	0.00029	0.20410	-0.15266	0.34437
joy	-0.58946	-0.49260	-0.17384	0.43343	-0.24294	-0.01359	-0.20438
disgust	0.23344	0.83367	0.11795	-0.10880	0.17566	0.06618	0.04554
anger	0.42320	0.86706	0.32393	-0.11523	0.14058	-0.13394	0.06740
The_power_elite	-0.08702	0.06352	0.81043	-0.07032	-0.09346	-0.08074	-0.06733
Entertainment	-0.57603	-0.40295	-0.73671	-0.18412	-0.35589	0.05834	-0.31996
Good_news	-0.16483	-0.20731	-0.02053	0.71088	-0.03108	0.01609	-0.04611
Shareability	-0.22505	-0.18057	-0.36418	0.54080	-0.14910	-0.01100	0.01135
Celebrity	-0.20888	-0.22304	-0.38024	-0.52540	-0.10984	-0.19901	-0.17286
Follow_up	0.05979	0.05991	0.11119	0.04793	0.61571	-0.12195	-0.05023
Drama	0.38699	0.37760	0.08075	-0.15243	0.77361	0.14579	0.15772
Conflict	0.41409	0.46163	0.27530	0.06578	-0.44660	-0.33068	-0.09061
Surprise1	0.11974	0.02589	0.08369	-0.10091	0.08982	0.77937	0.01578
surprise	-0.22957	-0.11412	-0.23768	0.27018	-0.06000	0.66039	-0.06521
Magnitude	0.19280	-0.18114	0.07828	0.18438	0.08043	-0.07454	0.73312
Relevance	-0.07368	0.12570	-0.02862	-0.09079	-0.03507	0.03186	0.66362

Bad news and Entertainment highly correlated with emotions (high +/- values on Factor 1)



Cluster analysis

In simplified, seven - dimensional factorial space, we **further summarised original variables** (more than 60% variability) **by clustering** on factor loadings.

Tehnike strojnog učenja

- **Nadzirano strojno učenje:**
  - k-NN (k najbližih susjeda)
  - Naïve Bayes
  - Linearna + logistička regresija
  - Stroj potpornih vektora
  - Slučajne šume
  - Nadzirane neuronske mreže
  - Itd.
- **Nenazirano strojno učenje:**
  - Grupiranje
  - Redukcija dimenzionanosti: modeliranje tema, matrična daktorizacija (PCA, SVD, FA,CA)
  - Skriveni Markovljevi modeli (HMM)
  - itd.

KRITERIJI

Prediktivne peformanse (točnost, AUC/ROC, preciznost, odziv, F1-score, etc.)

Brzina i skalabilnost:

- Vrijeme za izgradnju modela
- Vrijeme uporabe modela
- Zauzeće memorije vs. obrada na disku
- Cijena na

Robustnost

- Otpornost na šum, stršeće vrijednosti *(outliers*), nedostajuće vrijednosti

Interpretabilnost

- Razumijevanje modela i (*black box vs. white box*)

Kompaktnost modela

- Mobilni/ugrađeni uređaji

Uvod u algoritam k najbližih susjeda (k-NN) i dilema pristranost-varijanca

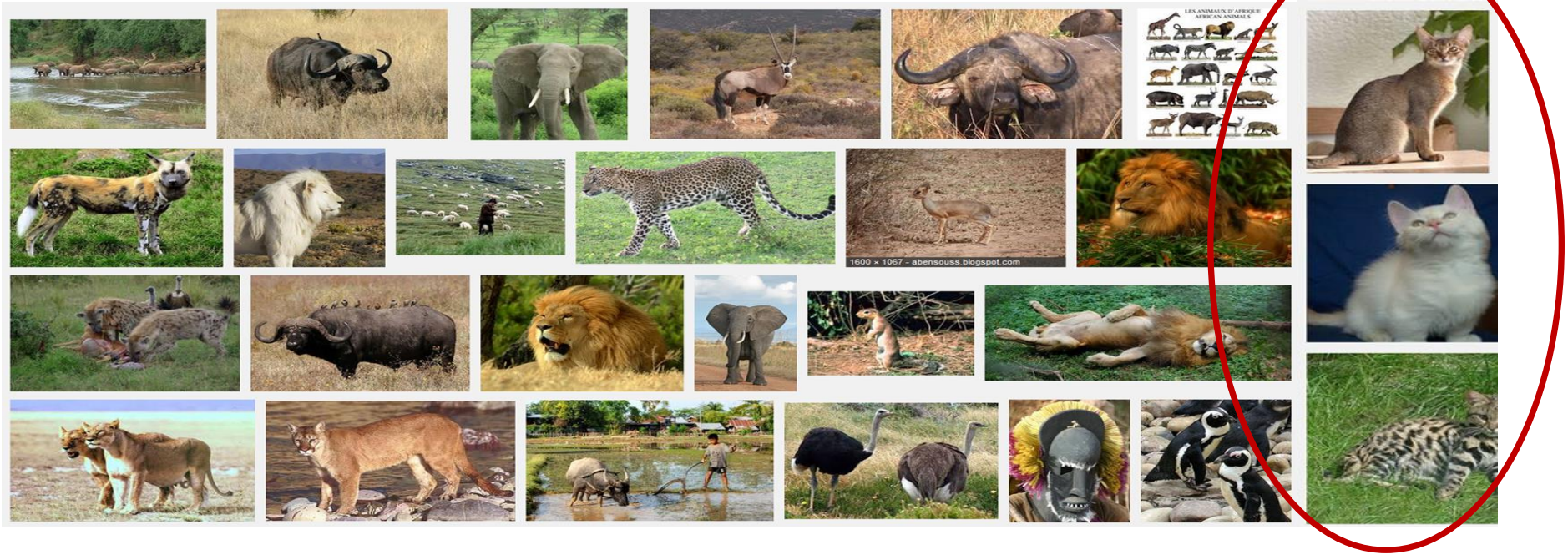
k najbližih susjeda (k-NN)

Dan je upit:

Nađi k najbližih primjera  na označenom skupu

k nearest neighbors (k-NN)

k = 3 algoritam glasa
za mačku



k-NN teme

Podaci jesu model

- Nema treniranja (nema optimizacije niti funkcije gubitka).
- Točnost se načelno poboljšava s više podataka.
- Matching is simple and fairly fast if data fits in memory.
- Obično treba podatke u memoriji.

k-NN teme

Podaci jesu model

- Nema treniranja (nema optimizacije niti funkcije gubitka).
- Točnost se načelno poboljšava s više podataka.
- Matching is simple and fairly fast if data fits in memory.
- Obično treba podatke u memoriji.

Minimalna konfiguracija

- Samo hiperparametar k (broj susjeda)
- Dva druga izbora su važna:
  - Težine susjeda (na primjer inverz udaljenosti)
  - Mjere sličnosti

k-NN teme

Klasifikacija:

- Model je y = f(X), y je diskretna vrijednost (labele).
- Dan X, izračunaj y = većina glasova k najbližih susjeda.
- Možemo koristiti težinske faktore*.

k-NN teme

Klasifikacija:

- Model je y = f(X), y je diskretna vrijednost (labele).
- Dan X, izračunaj y = većina glasova k najbližih susjeda.
- Možemo koristiti težinske faktore*.

Regresija:

- Model je y = f(X), y je realan broj.
- Ako je dan X, izračunaj y = sred.vrij. k najbližih susjeda.
- Također se može koristiti težinska funkcija* susjeda.

* Težinska funkcija inverz udaljenosti, jezgrene funkcije (Gauss).

k-NN mjere udaljenosti

- **Euclidean Distance:** Simplest, fast to compute

$$d(x, y) = \|x - y\|$$

- **Cosine Distance:** Good for documents, images, etc.

$$d(x, y) = 1 - \frac{x \cdot y}{\|x\|\|y\|}$$

- **Jaccard Distance:** For set data:

$$d(X, Y) = 1 - \frac{|X \cap Y|}{|X \cup Y|}$$

- **Hamming Distance:** For string data:

$$d(x, y) = \sum_{i=1}^{n} (x_i \neq y_i)$$

k-NN mjere udaljenosti

- **Manhattan Distance:** Coordinate-wise distance

$$d(x, y) = \sum_{i=1}^{n} |x_i - y_i|$$

- **Edit Distance:** for strings, especially genetic data.

- **Mahalanobis Distance:** Normalized by the sample covariance matrix – unaffected by coordinate transformations.

$$d(\vec{x}, \vec{y}) = \sqrt{(\vec{x} - \vec{y})^{\top} \mathbf{S}^{-1} (\vec{x} - \vec{y})}.$$

$$d(\vec{x}, \vec{y}) = \sqrt{\sum_{i=1}^{N} \frac{(x_i - y_i)^2}{s_i^2}}$$

K=1, Veronoi dijagrami

Euclidean distance

Manhattan distance

https://en.wikipedia.org/wiki/Voronoi_diagram#Applications

Predviđanje na temelju uzorka

- Većina uzoraka su iz beskonačnih populacija.
- Najviše smo zainteresirani za modele populacije, ali imam samo uzorak.

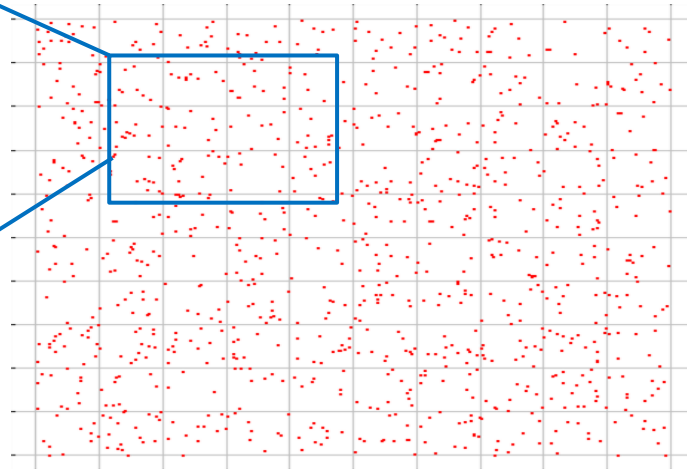
Za skupove koji se sastoje od (X, y)

- značajke X, oznaka y

Za stvarni model f:

- $y = f(X)$

Mi učimo na uzorku za učenje D
i označavamo model s $f_D(X)$



Pristranost i varijanca

Our data-generated model $f_D(X)$ is a **statistical estimate** of the true function $f(X)$.

Because of this, its subject to bias and variance:

Bias: if we train models $f_D(X)$ on many training sets D, bias is the expected difference between their predictions and the true y's.

i.e. $$Bias = \mathrm{E}[f_D(X) - y]$$

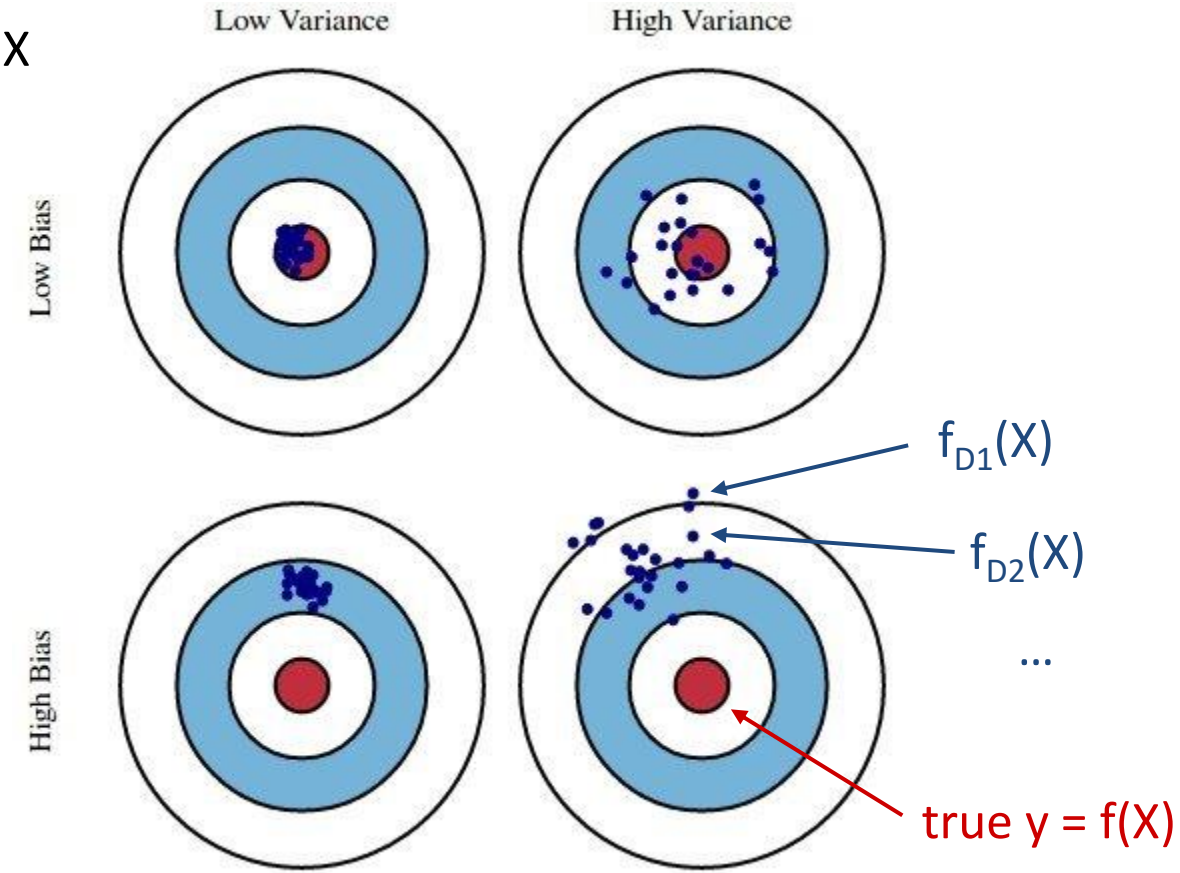
E[] is taken over points X and datasets D

Variance: if we train models $f_D(X)$ on many training sets D, variance is the variance of the estimates:

$$Variance = \mathrm{E}\left[\left(f_D(X) - \bar{f}(X)\right)^2\right]$$

Where $\bar{f}(X) = \mathrm{E}[f_D(X)]$ is the average prediction on X.

Razmatramo fiksni X



“Prava” pristranost/varijanca dilema: uprosječite ovu sliku kroz sve X

Pristranost /varijanca dilema

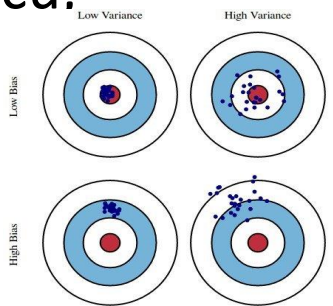
Bias/variance tradeoff

Dilema pristranost/varijanca povezana je s kompleksnošću modela.

Kompleksni modeli (puno parametara) obično imaju višu varijancu i nižu pristranost.

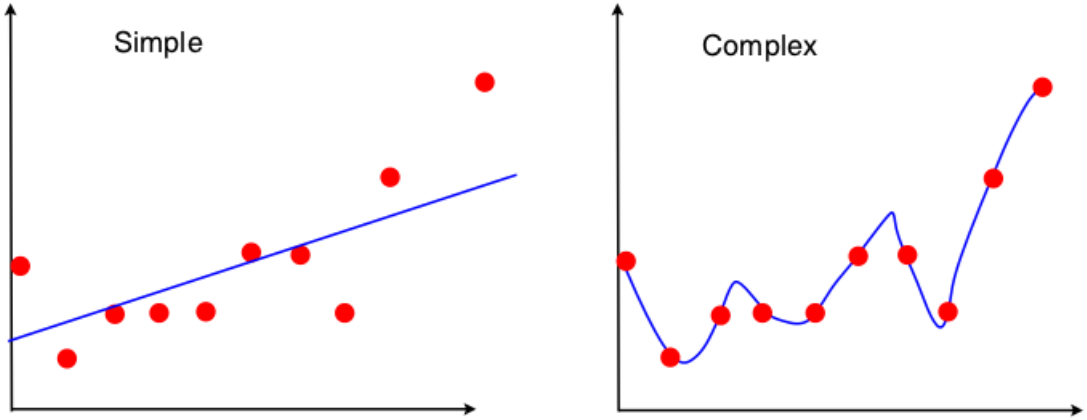
Jednostavni modeli (malo parametara) imaju nižu varijancu.

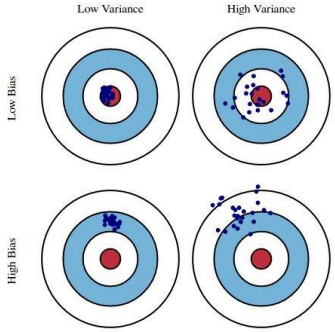
ali veću pristranost.



Pristranost/varijanca dilema

Linearni modeli oblikuju pravac (2D), polinom visokog stupnja oblikuje kompleksnu krivulju. Polinom može pristati uz individualni element, prije nego što može oblikovati populaciju. Njegov oblik jako ovisi o uzorku i zato ima visoku varijancu.





Bias/variance tradeoff

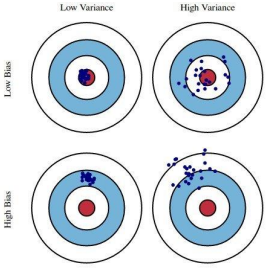
The total expected error is

$$Bias^2 + Variance$$

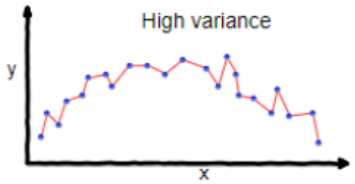
Because of the bias-variance trade-off, we want to **balance** these two contributions.

If $Variance$ strongly dominates, it means there is too much variation between models. This is called **over-fitting**.

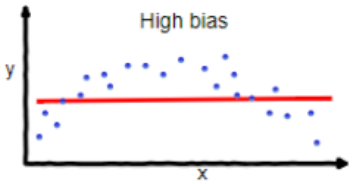
If $Bias$ strongly dominates, then the models are not fitting the data well enough. This is called **under-fitting**.



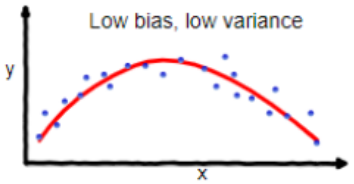
Pristranost/varijanca dilema



overfitting



underfitting

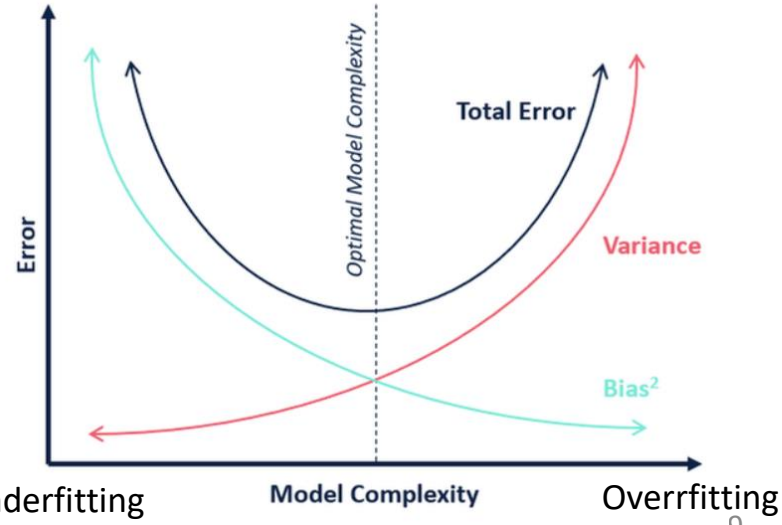


Go

$$Err(x) = E\left[(Y - \hat{f}\,(x))^2\right]$$

$$Err(x) = \left(E[\hat{f}\,(x)] - f(x)\right)^2 + E\left[\left(\hat{f}\,(x) - E[\hat{f}\,(x)]\right)^2\right]$$

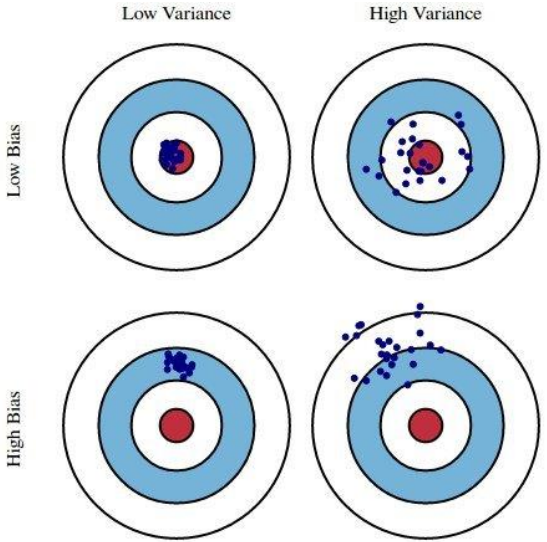
$$Err(x) = \text{Bias}^2 + \text{Variance} + \text{Irreducible Error}$$



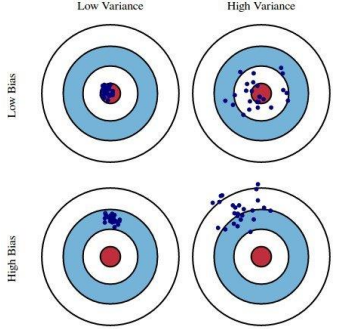
Izbor k za k-nn

Pristranost-varijanca dilema:

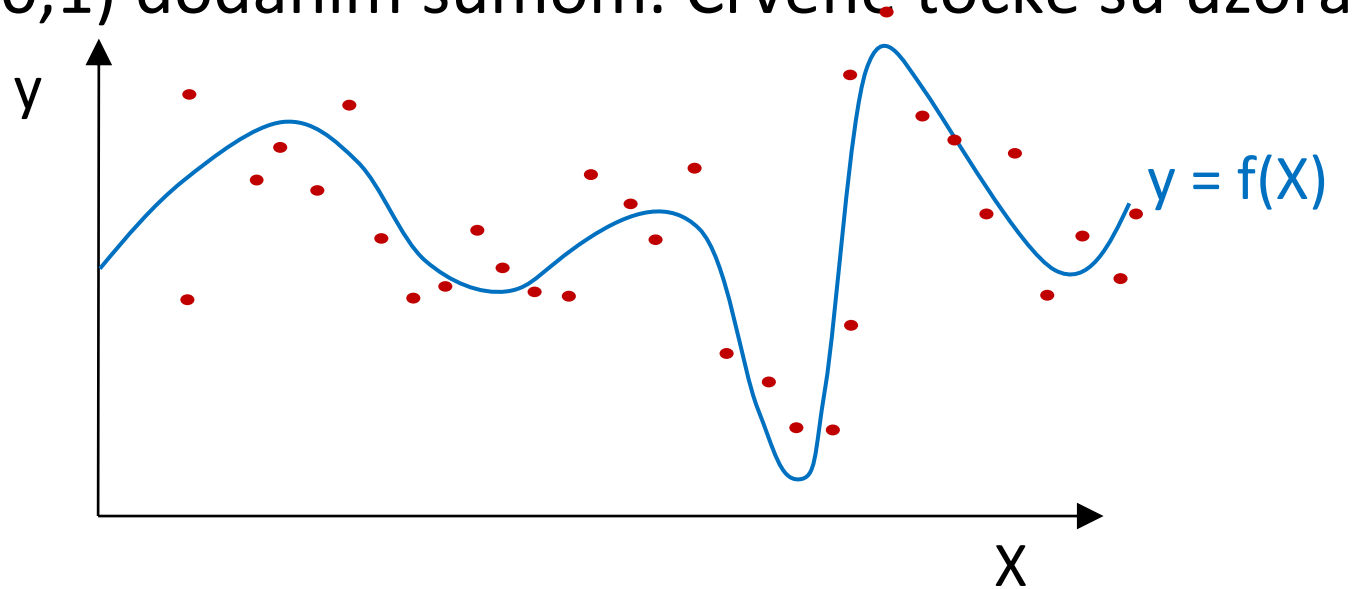
- Mali k → ?
- Veliki k → ?



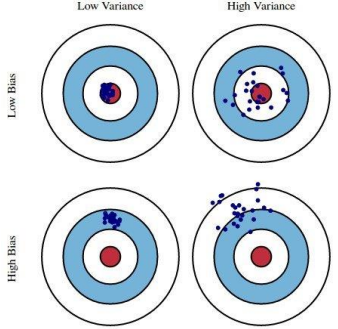
Izbor k



- Mali k → mala pristranost, visoka varijanca
- Veliki k → visoka pristranost, mala varijanca
- Pretpostavimo da pravi podaci slijede plavu krivulju sa nekim N(0,1) dodanim šumom. Crvene točke su uzorak.

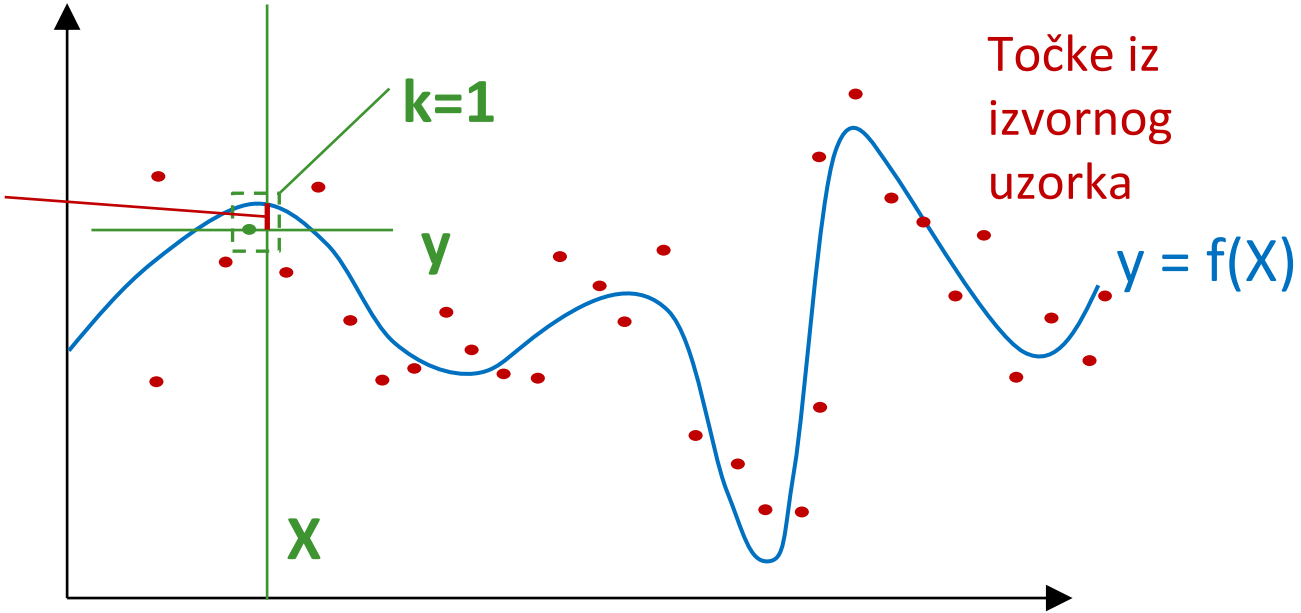


Izbor k

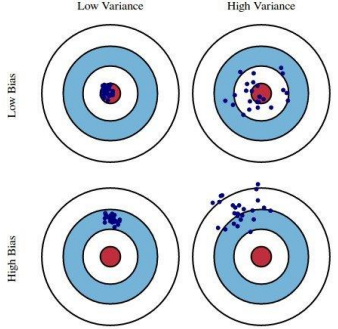


- **Mali k** → mala pristranost, visoka varijanca
- Veliki k → visoka pristranost, mala varijanca

Pristranost = srednja vrijednost odstojanja *offset*

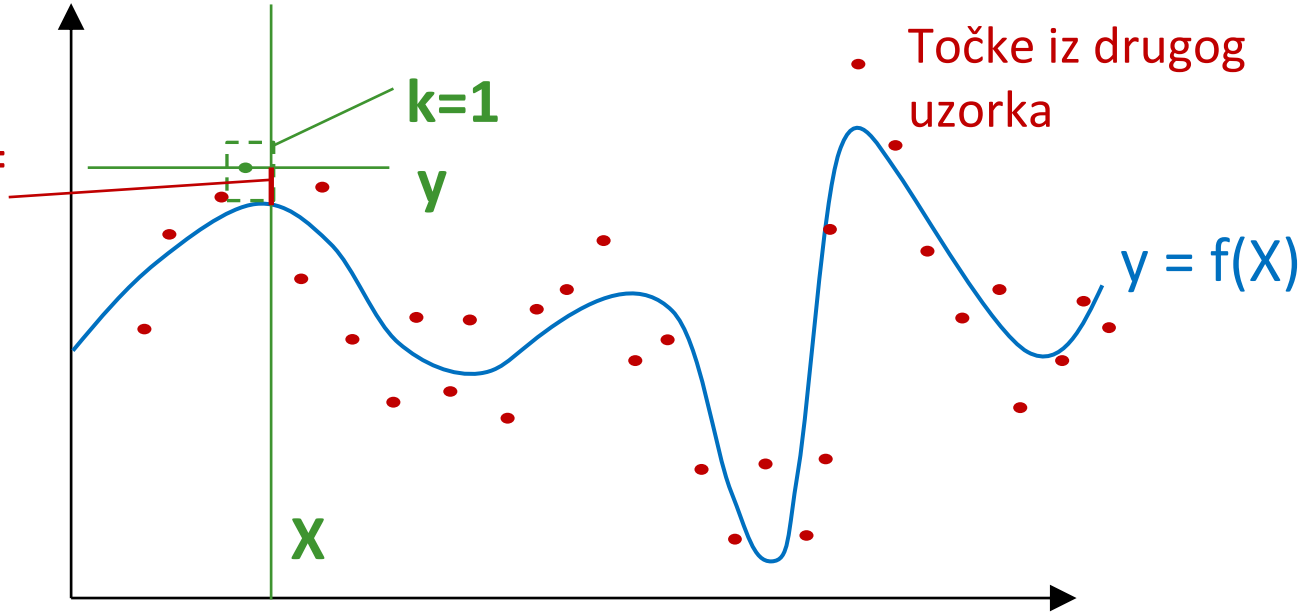


Izbor k

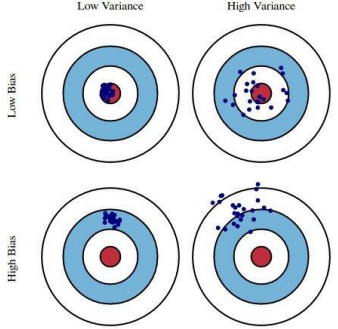


- **Mali k** → mala pristranost, visoka varijanca
- Veliki k → visoka pristranost, mala varijanca

Pristranost = srednja vrijednost odstojanja

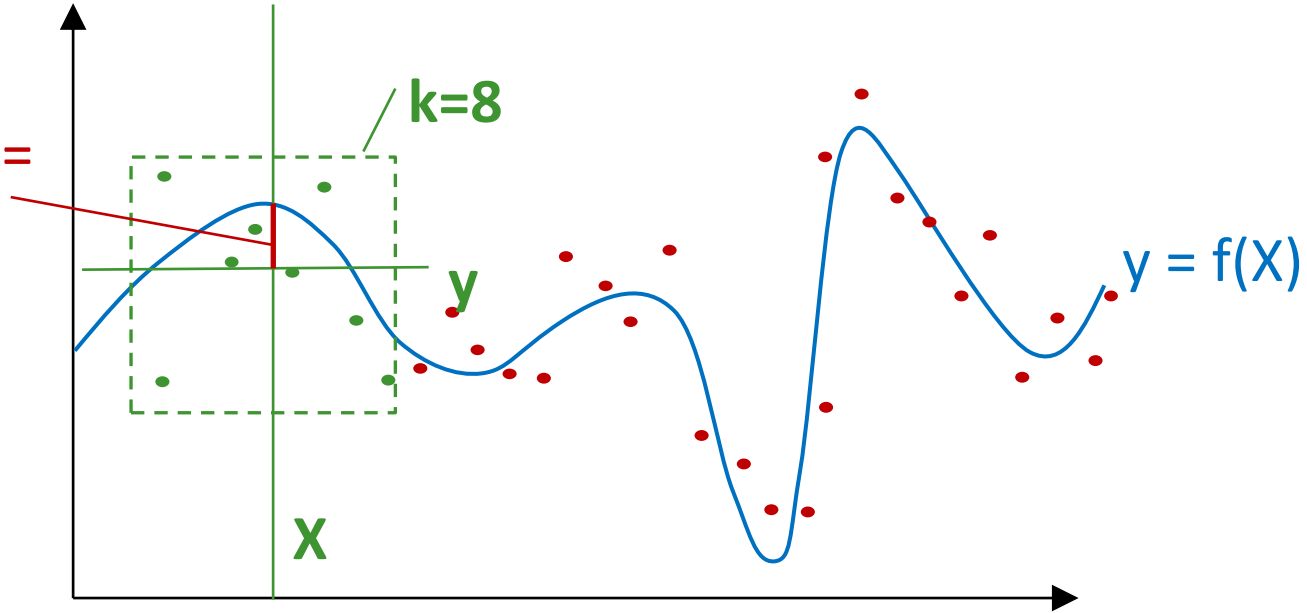


Izbor k

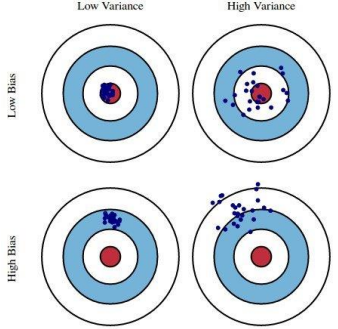


- Mali k → mala pristranost, visoka varijanca
- **Veliki k** → visoka pristranost, mala varijanca

Pristranost = srednja vrijednost odstojanja

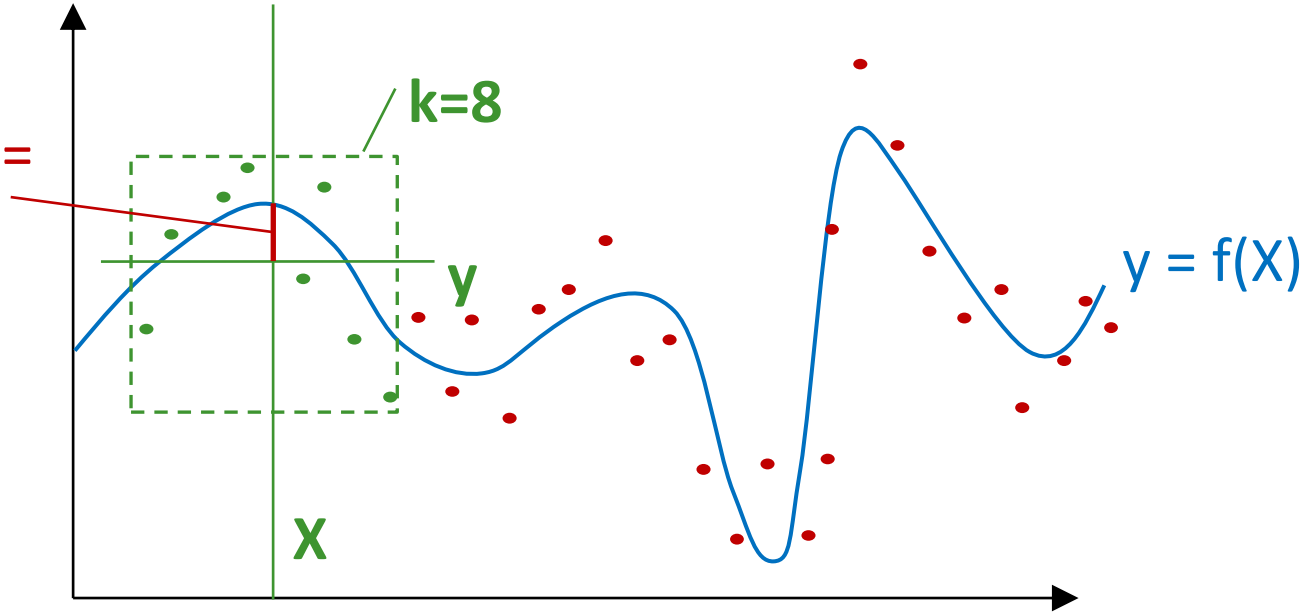


Izbor k



- Malil k → mala pristranost, visoka varijanca
- **Veliki k** → visoka pristranost, mala varijanca

Pristranost = srednja vrijednost odstojanja

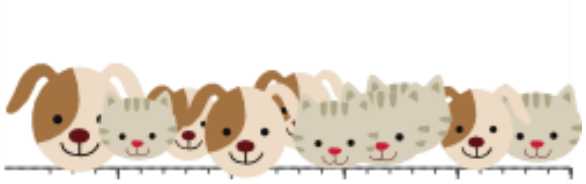


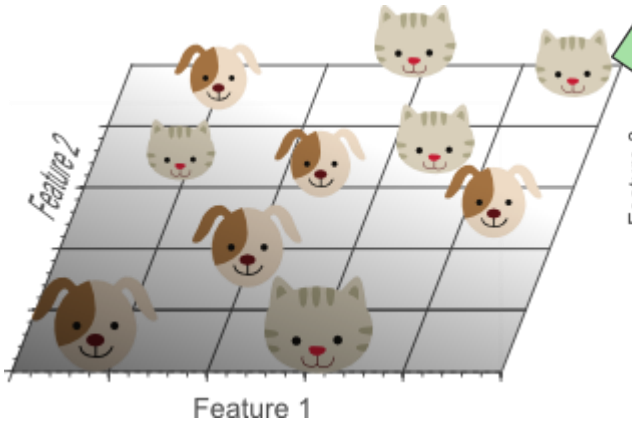
Izbor k u praksi

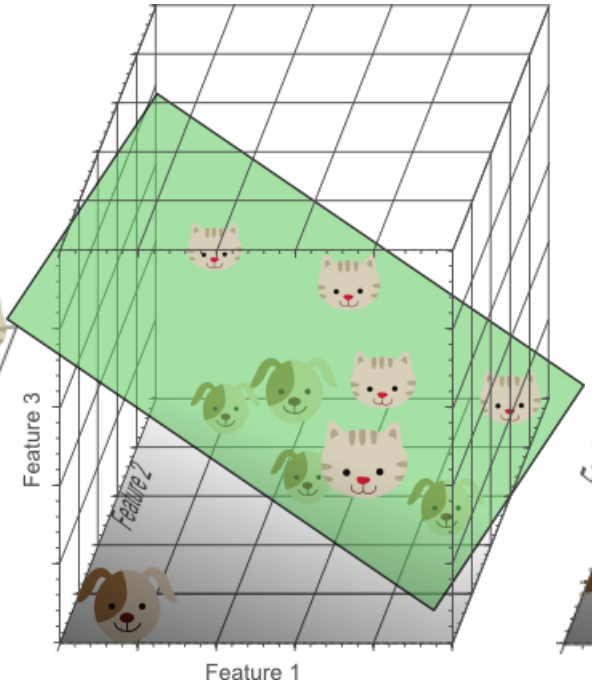
Koristi *leave-one-out* (LOO) unakrsnu provjeru:

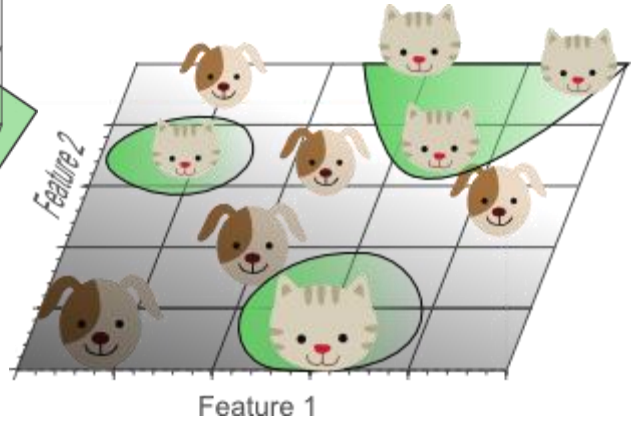
- **Podjela:** Rastavi podatke na učenje i testiranje u slučajnom omjeru, na primjer 80-20 %.
- **Predviđanje:** Za svaku točku u skupu za učenje predviđaj koristeći k –nn iz skupa svih drugih točaka u skupu za učenje. Izmjeri LOO pogrešku klasifikacije ili SSE za regresiju
- **Ugađanje modela:** Pokušaj različite vrijednosti *k* i koristi onu koja daje najmanju LOO pogrešku.
- **Evaluacija:** Testiraj na zasebnom skupu za testiranje.

Klasifikacija i kletva dimenzionalnosti



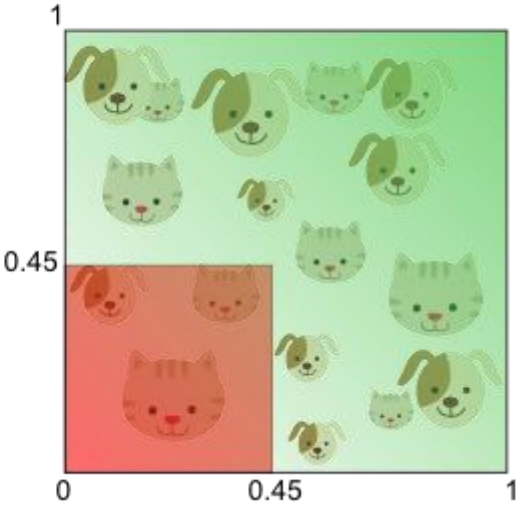


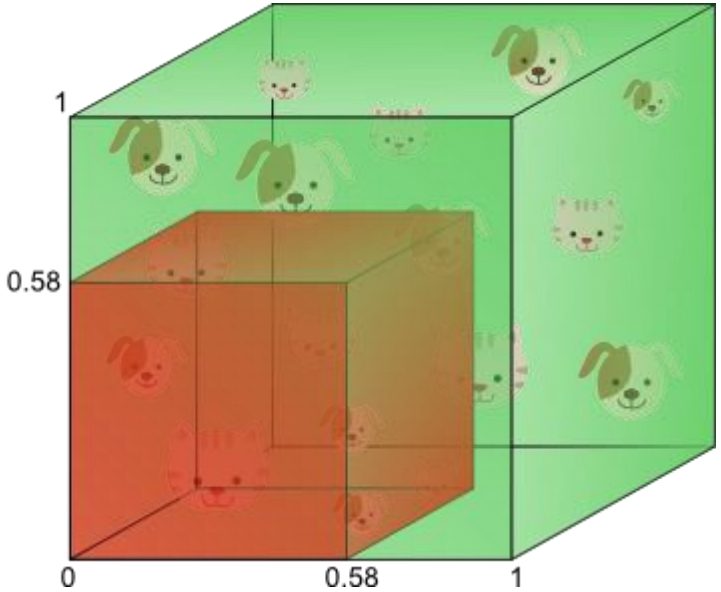




https://www.visiondummy.com/2014/04/curse-dimensionality-affect-classification/

Klasifikacija i kletva dimenzionalnosti





https://www.visiondummy.com/2014/04/curse-dimensionality-affect-classification/

k-NN i kletva dimenzionalnosti

Kletva dimenzionalnosti odnosi se na fenomen koji se dešava u visokim dimenzijama (100 do milijuni), ne u nižim, napr. u 3-dimenzijskom prostoru.

K-nn se oslanja na bliskost točaka u prostoru.

Podaci u višim dimenzijama su rijetki, manje gusti nego u manjim dimenzijama.

Za k-nn to znači da ima manje točaka koje su vrlo blizu u prostoru značajki (vrlo slične) točki X, čiji y želimo predvidjeti.

k-NN i kletva dimenzionalnosti

- Najbliža udaljenost se pribilžava prosječnoj daljenosti - knn ima neznatno bolju prediktivnu

- Nema dovoljno primjera za veliki broj dimenzija – rješenje je povećanje podataka – nedostatak – veća moć računala

k-NN i kletva dimenzionalosti

Iz te perspektive iznenađujuće je da k-nn radi dobro u visokim dimenzijama

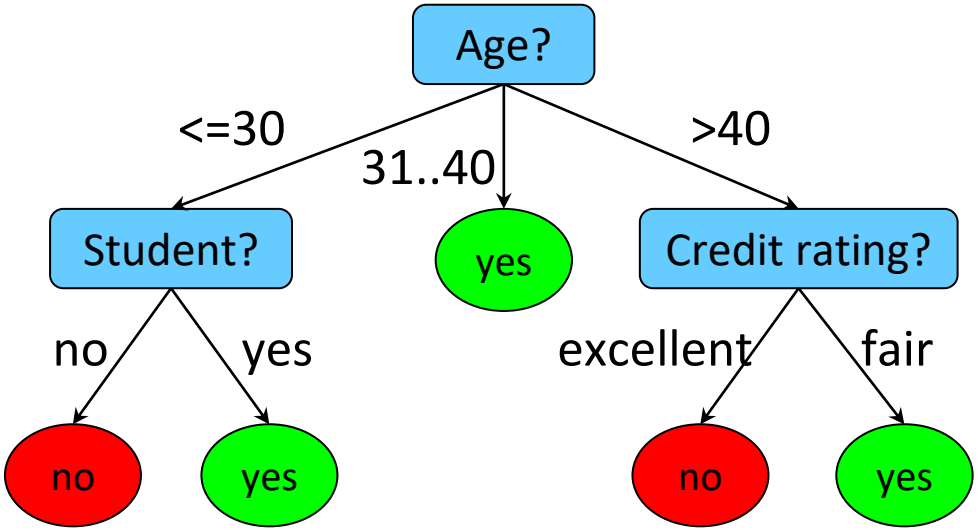
Srećom, stvarni podaci nisu kao slučajeve točke u visoko dimenzijskom prostoru, oni žive u **gustim klasterima** blizu **puno manje dimenzijskih površina.**

Također, točke mogu biti vrlo „slične” čak i kada je njihova euklidska udaljenost velika. Na primjer, dokumenti koji imaju nekoliko dominantnih zajedničkih riječi – vjerojatno su unutar iste teme.
(Druga metrika)

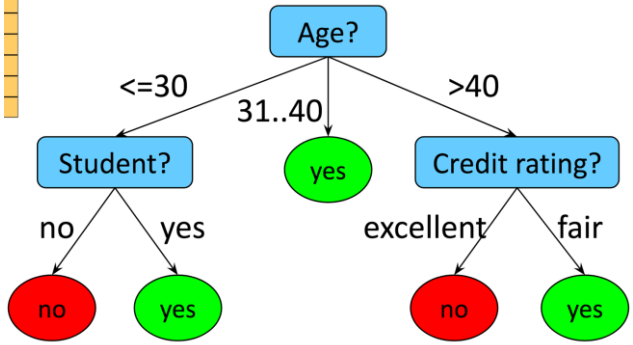
Stabla odluke, slučajne šume i *boosted trees*

Stabla odluke: primjer

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no



Stabla odluke



Model: dijagram stablaste strukture

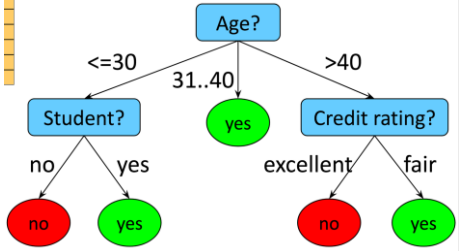
- Čvorovi su testovi pojedinog atributa
- Grane su vrijednosti atributa
- Listovi su oznake klasa

Uspješnost: točnost klasifikacije

Optimizacija:

- NP-težak problem
- Heuristika: pohlepan *top-down aloritam,* konstrukcija + orezivanje

Stabla odluke



Stabla odluke (*top-down divide-and-conquer strategy*)

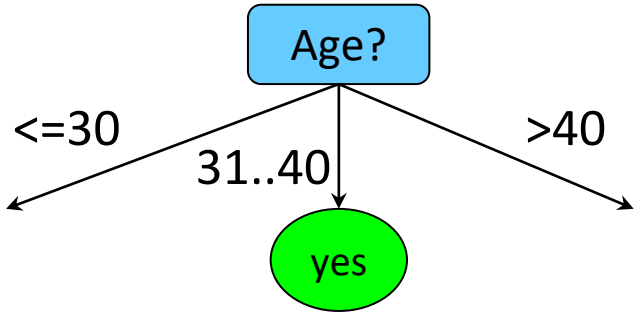
- Na početku, svi primjeri za učenje pripadaju korijenu
- Primjeri se dijele rekurzivno na temelju atributa koji najviše diskriminira
- Diskriminativna snaga se temelji na **informacijskoj dobiti** (ID3/C4.5) ili *Gini impurity* (CART)

Particioniranje prestaje kada:

- Svi uzorci pripadaju istoj klasi → pridijeli tu klasu listu
- Nema atributa za dijeljenje → većina glasa za labelu klase tog lista

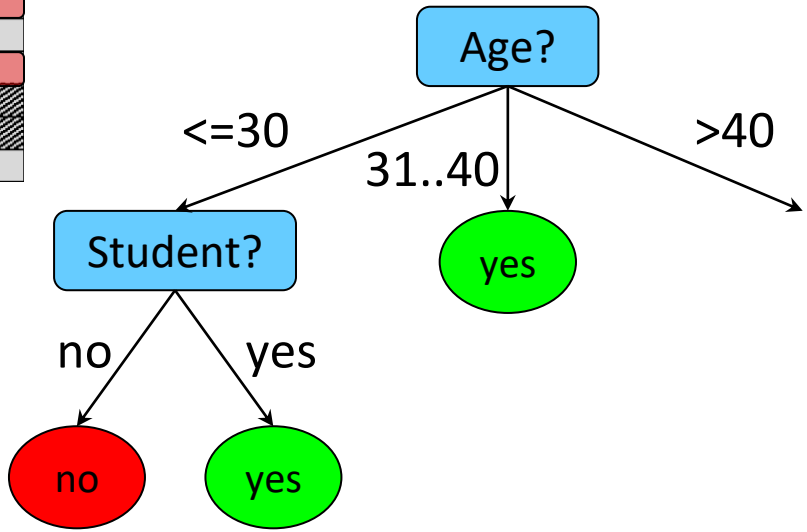
Stabla odluke

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31...40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31...40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31...40	medium	no	excellent	yes
31...40	high	yes	fair	yes
>40	medium	no	excellent	no



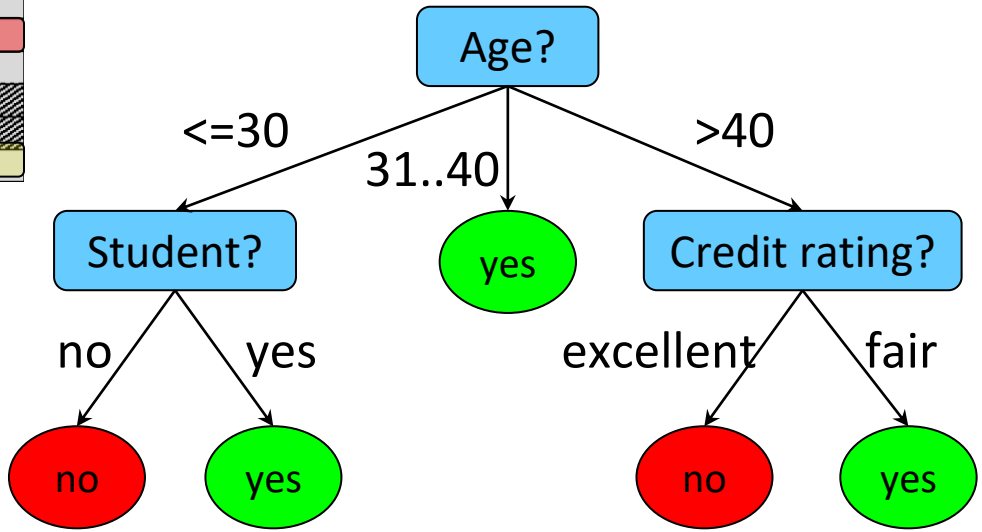
Decision tree induction

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no



Decision tree induction

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no



Izbor atributa

Kod danog skupa S, P su pozitivni, N negativni primjeri.

Iznos entropije za skup S

$$H(P,\ N) = -\frac{P}{P+N}\log_2\frac{P}{P+N} - \frac{N}{P+N}\log_2\frac{N}{P+N}$$

Uoči:

- If P=0 (ili N=0), H(P, N) = 0 → nema neizvjesnosti
- If P=N, H(P, N) = 1 → maksimalna neizvjesnost

Izbor atributa

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31...40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31...40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31...40	medium	no	excellent	yes
31...40	high	yes	fair	yes
>40	medium	no	excellent	no

$H_S = H(9, 5) = 0.94$

Age	[<=30]	H(2, 3) = 0.97
Age	[31...40]	H(4, 0) = 0
Age	[>40]	H(3, 2) = 0.97

Income	[high]	H(2, 2) = 1
Income	[med]	H(4, 2) = 0.92
Income	[low]	H(3, 1) = 0.81

Student	[yes]	H(6, 1) = 0.59
Student	[no]	H(3, 4) = 0.98

Rating	[fair]	H(6, 2) = 0.81
Rating	[exc]	H(3, 3) = 1

Izbor atributa

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31...40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31...40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31...40	medium	no	excellent	yes
31...40	high	yes	fair	yes
>40	medium	no	excellent	no

$$H_S = H(9, 5) = 0.94$$

$H_{Age} = p([<=30]) \cdot H(2, 3) + p([31...40]) \cdot H(4, 0) + p([>40]) \cdot H(3, 2) =$

$= 5/14 \cdot 0.97 + 4/14 \cdot 0 + 5/14 \cdot 0.97 = 0.69$

$H_{Income} = p([high]) \cdot H(2, 2) + p([med]) \cdot H(4, 2) + p([low]) \cdot H(3, 1) =$

$= 4/14 \cdot 1 + 6/14 \cdot 0.92 + 4/14 \cdot 0.81 = 0.91$

$H_{Student} = p([yes]) \cdot H(6, 1) + p([no]) \cdot H(3, 4) = 7/14 \cdot 0.59 + 7/14 \cdot 0.98 = 0.78$

$H_{Rating} = p([fair]) \cdot H(6, 2) + p([exc]) \cdot H(3, 3) = 8/14 \cdot 0.81 + 6/14 \cdot 1 = 0.89$

Izbor atributa

Attribute A partitions S into S_1, S_2, ... S_v

Entropija atributa A is

$$H(A) = \sum_{i=1}^{v} \frac{P_i + N_i}{P + N} H(P_i, N_i)$$

Informacijska dobit podjelom S koristeći A

$$Gain(A) = H(P, N) - H(A)$$

Gain(Age) = 0.94 – 0.69 = 0.25 ← split on age

Gain(Income) = 0.94 – 0.91 = 0.03

Gain(Student) = 0.94 – 0.78 = 0.16

Gain(Rating) = 0.94 – 0.89 = 0.05

Orezivanje

U postupku generiranja stabla ne filtrira se šum→ **prenaučenost**

Mnoge moguće tehnike orezivanja:

Orezivanje

U postupku generiranja stabla ne filtrira se šum→ **prenaučenost**

Mnoge moguće tehnike orezivanja:

- Zaustavljanje dijeljenja čvora kada broj dodijeljenih primjera ide ispod određene zadane granice.
- *Bottom-up* unakrsna provjera: Izgradi puno stablo i zamijeni čvorove je za listove s oznakom klase koja je brojnija ako se klasifikacijska točnost na skupu za provjeru (**validaciju**) se ne pogoršava .

Komentari na stabla odluke

Stabla odluke su samo primjer klasifikacijskog algoritma

- Puno drugih (k-NN, naive Bayes, SVM, neuronske mreže, logistička regresija, random forest ...)

Komentari na stabla odluke

Stabla odluke su samo primjer klasifikacijskog algoritma

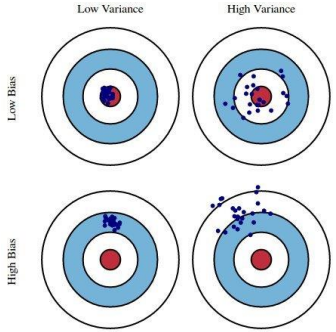
- Puno drugih (k-NN, naive Bayes, SVM, neuronske mreže, logistička regresija, random forest ...)

Ne spadaju među najbolje ...

- Osjetljiva su na male perturbacije u podacima (**visoka varijanca)**
- Skloni su **prenaučenosti**
- **Nisu inkrementalni**: Potrebno je cijeli postupak napraviti ispočetka ako se pojavi novi podatak

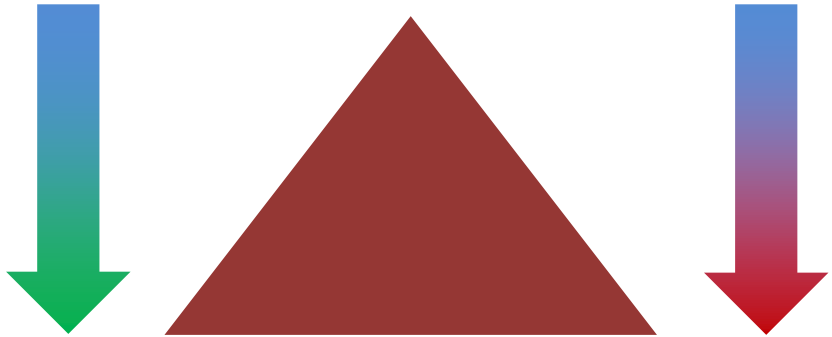
Stabla odluke

- Kako se mijenjaju pristranost i varijanca povećanjem dubine stabla?



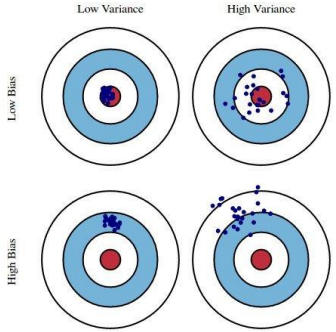
Stabla odluke

- Povećanjem dubine stabla, pristranost se smanjuje, varijanca se povećava. Zašto?



Pristranost se smanjuje s dubinom stabla

Varijanca se povećava s dubinom stabla



Zajednice metoda

su metaforički kao ***crowdsourced machine learning algorithms***:

- Uzmi skup jednostavnih ili slabih algoritama (learners)
- Kombiniraj ih da bi dobio jedan, bolji.

Zajednice metoda

su metaforički kao ***crowdsourced machine learning algorithms***:

- Uzmi skup jednostavnih ili slabih algoritama (learners)
- Kombiniraj ih da bi dobio jedan, bolji.

Vrste:

- ***Bagging***: treniraj *learners* paralelno na različitim uzorcima podataka, a zatim kombiniraj izlaze glasanjem (diskretni izlaz) ili uprosječnjavanjem (kontinuirani izlaz).
- ***Stacking***: kombiniraj izlaze iz različitih modela korištenjem *learnera* na drugoj razini.
- ***Boosting***: ponovi učenje, ali nakon filtriranja/otežavanja primjera temeljeno na prethodnom outputu

Slučajne šume

Izgradi K stabala na skupu uzorkovanom iz originalnog skupa (size N) sa vraćanjem (*bootstrap samples*), p = broj značajki.

- Izvuci K bootstrap uzoraka veličine N
- Izgradi svako stablo odluke slučajnim izborom *m* od *p* značajki u svakom čvoru i izborom najbolje značajke za podjelu.
- Agregiraj predviđanja stabala (najpopularniji glas ili prosjek) da bi dobio konačan odgovor za labelu klase ili vrijednost (primjer bagginga).

Tipično *m* može biti sqrt(p), ali i manji.

Slučajne šume

PRINCIP: zanima nas glasanje između različitih modela (learners), pa ne želimo da su modeli previše slični. Slijedeći kriteriji osiguravaju raznolikost u pojedinim stablima:

- Izvuci K bootstrap uzoraka veličine N:

- Izgradi svako stablo odluke slučajnim izborom *m* od *p* značajki u svakom čvoru i izborom najbolje značajke za podjelu.

Slučajne šume

PRINCIP: zanima nas glasanje između različitih modela (learners), pa ne želimo da su modeli previše slični. Slijedeći kriteriji osiguravaju raznolikost u pojedinim stablima:

- Izvuci K bootstrap uzoraka veličine N:
  - Svako stablo je trenirano na različitom skupu.
- Izgradi svako stablo odluke slučajnim izborom *m* od *p* značajki u svakom čvoru i izborom najbolje značajke za podjelu.
  - Odgovarajući čvorovi u različitim stablima (obično) ne koriste iste atribute za podjelu.

Slučajne šume

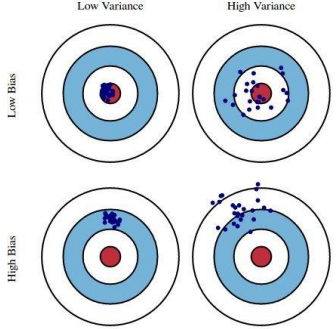
- **Vrlo popularne u praksi,** vjerojatno najpopularniji klasifikator za guste podatke (do više tisuća značajki)
- **Jednostavne za implementaciju** (jednostavno treniranje puno običnih stabala odluke)
- **Laka paralelizacija**

- **Potrebno je puno prolazaka kroz podatke** – barem/najmanje maksimalne dubine stabla (<< *boosted trees*)

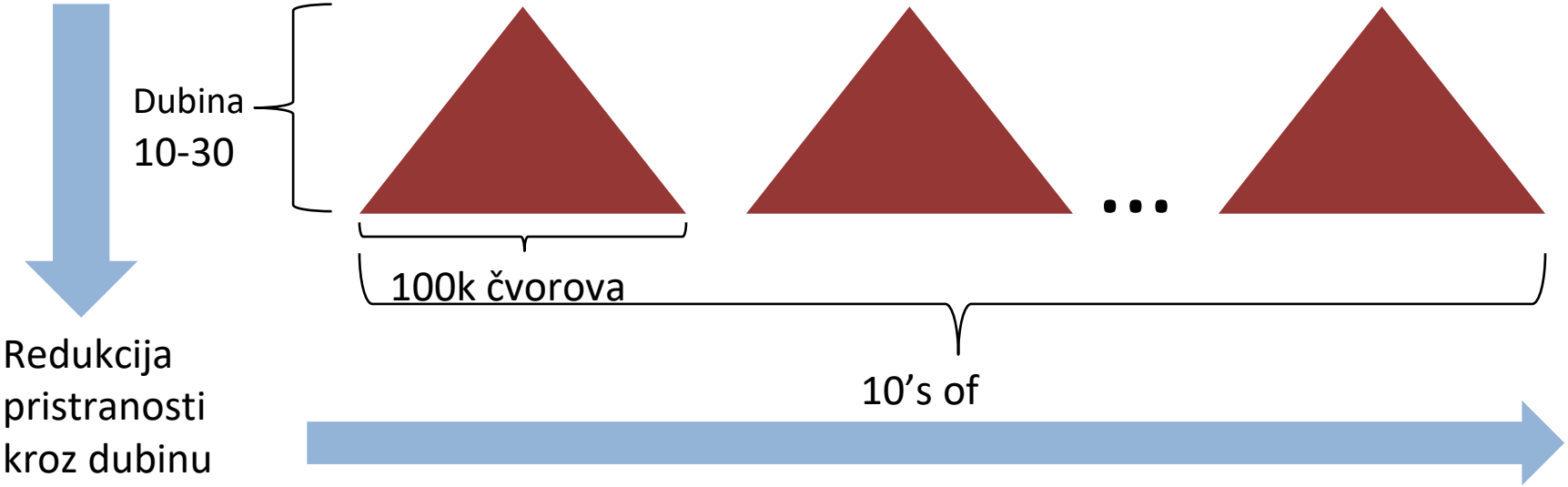
Boosted trees (BDT)

- Novija alternativa slučajnim šumama (RF) [dobar intro ovdje i ovdje]
- Za razliku od RF, čija su stabla trenirana **nezavisno**,
  BDT stabla su trenirana sekvencijalno koristeći ***boosting***: Svako stablo je trenirano da predviđa korektno, ali korigira se pogreška (rezidual) iz prethodnog stabla (redukcija pristranosti)
- Oba modela RF i BDT mogu proizvesti vrlo kvalitetne modele. Superiornost jedne metode nad drugom zavisi o skupu podataka.
- Vrlo su različiti modeli i zahtjevi za resursima - netrivijalno uspoređivati metode.

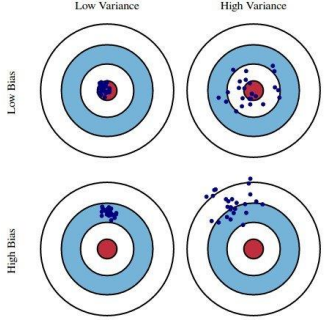
Slučajne šume vs. *boosted trees*



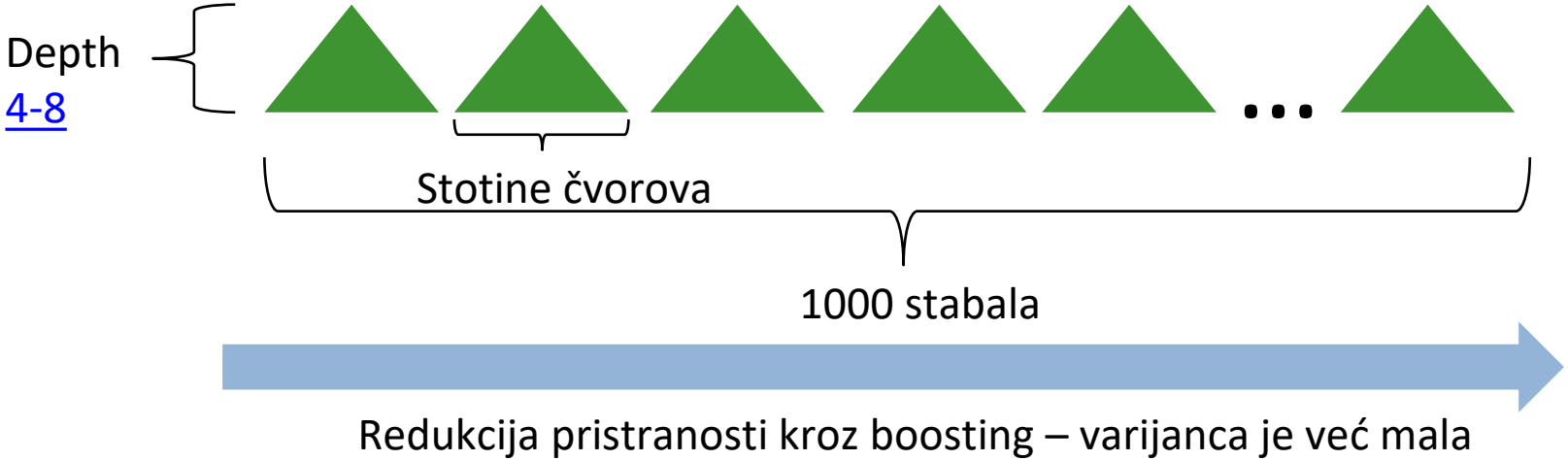
- "Geometrija" metoda je vrlo različita:
- Slučajne šume koriste desetke dubokih širokh stabala:



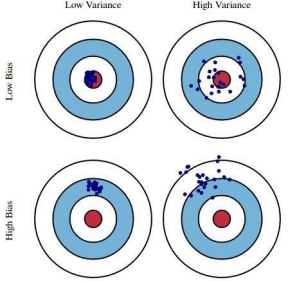
Slučajne šume vs. *boosted trees*



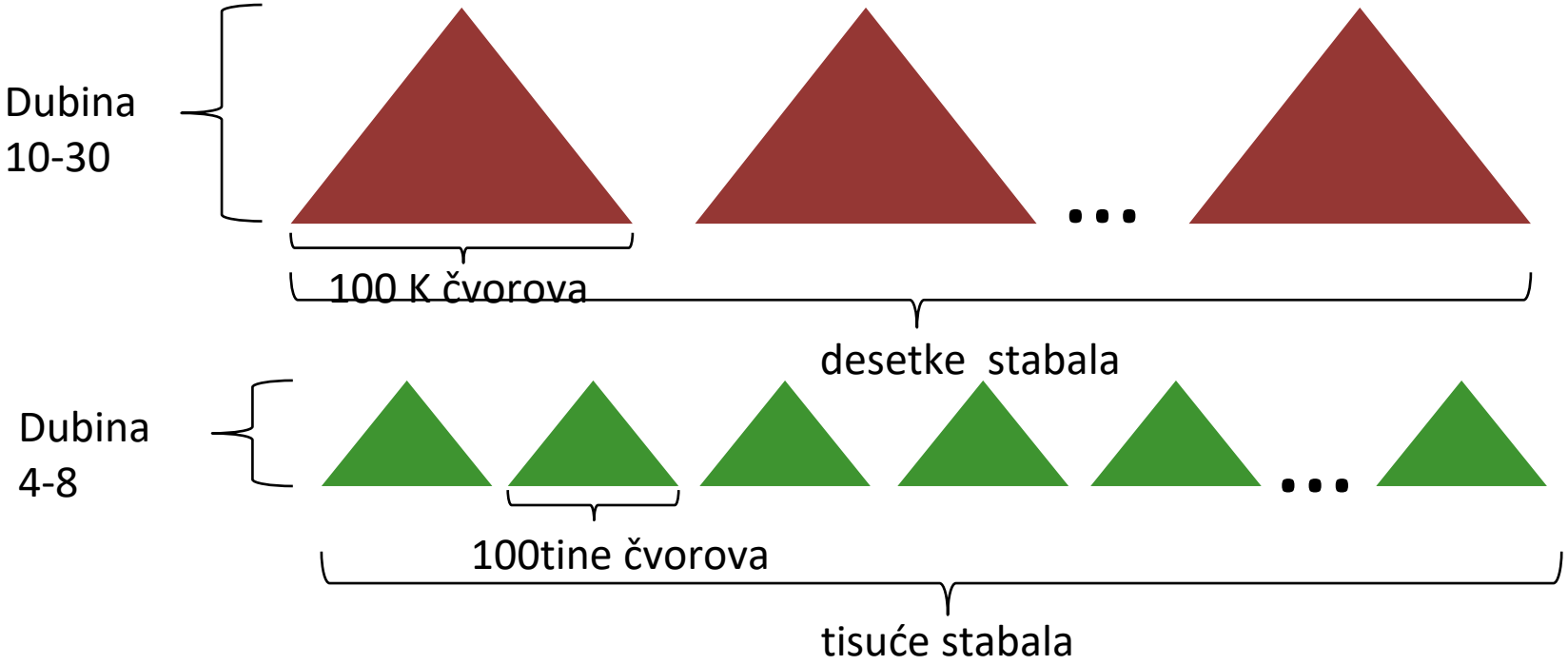
- "Geometrija" metoda je vrlo različita:
- BDT koriste na tisuće plitkih, malih stabala:



Slučajne šume vs. *boosted trees*



- RF uče paralelno, mogu biti vrlo brze
- Brža evaluacija (*runtime*) također bolje kod RF



Transparentnost modela

Duboke neuronske mreže smatraju se teškim *crnim kutijama*.

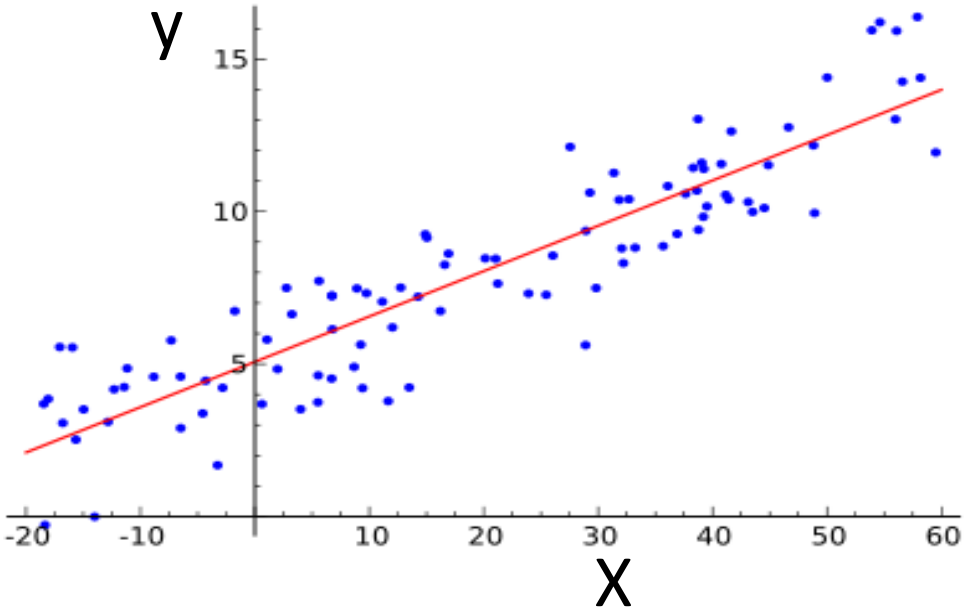
Da li je lakše interpretirati model s 1000 stabala?!

http://www.r2d3.us/visual-intro-to-machine-learning-part-1/

Linear and logistic regression

Linearna regresija

- U 5. predavanju
- Cilj: naći najbolju linearnu funkciju y=f(X) koja objašnjava podatke

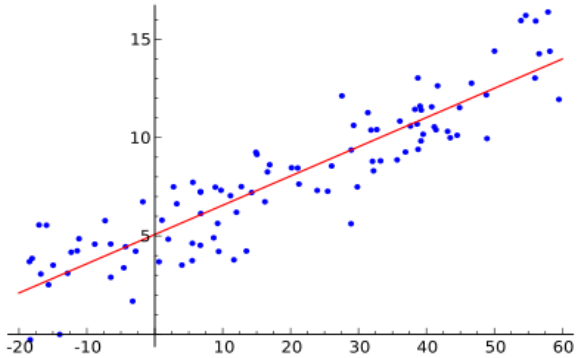


Linear regression

The predicted value of y is given by:

$$\hat{y} = \hat{\beta}_0 + \sum_{j=1}^{p} X_j \hat{\beta}_j$$

The vector of coefficients $\hat{\beta}$ is the regression model.



Linear regression

The regression formula

$$\hat{y} = \hat{\beta}_0 + \sum_{j=1}^{p} X_j \hat{\beta}_j$$

if $X_0 = 1$, can be written as a matrix product with X a row vector:

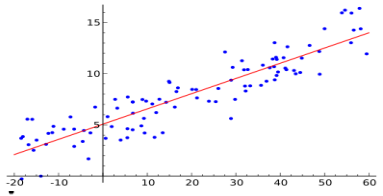
$$\hat{y} = \mathrm{X}\,\hat{\beta}$$

We get this by writing all of the input samples in a single matrix **X**:

i.e. **rows of X** $= \begin{pmatrix} X_{11} & \cdots & X_{1n} \\ \vdots & \ddots & \vdots \\ X_{m1} & \cdots & X_{mn} \end{pmatrix}$

are **distinct observations**, **columns of X** are **input features**.

Podsjetnik:



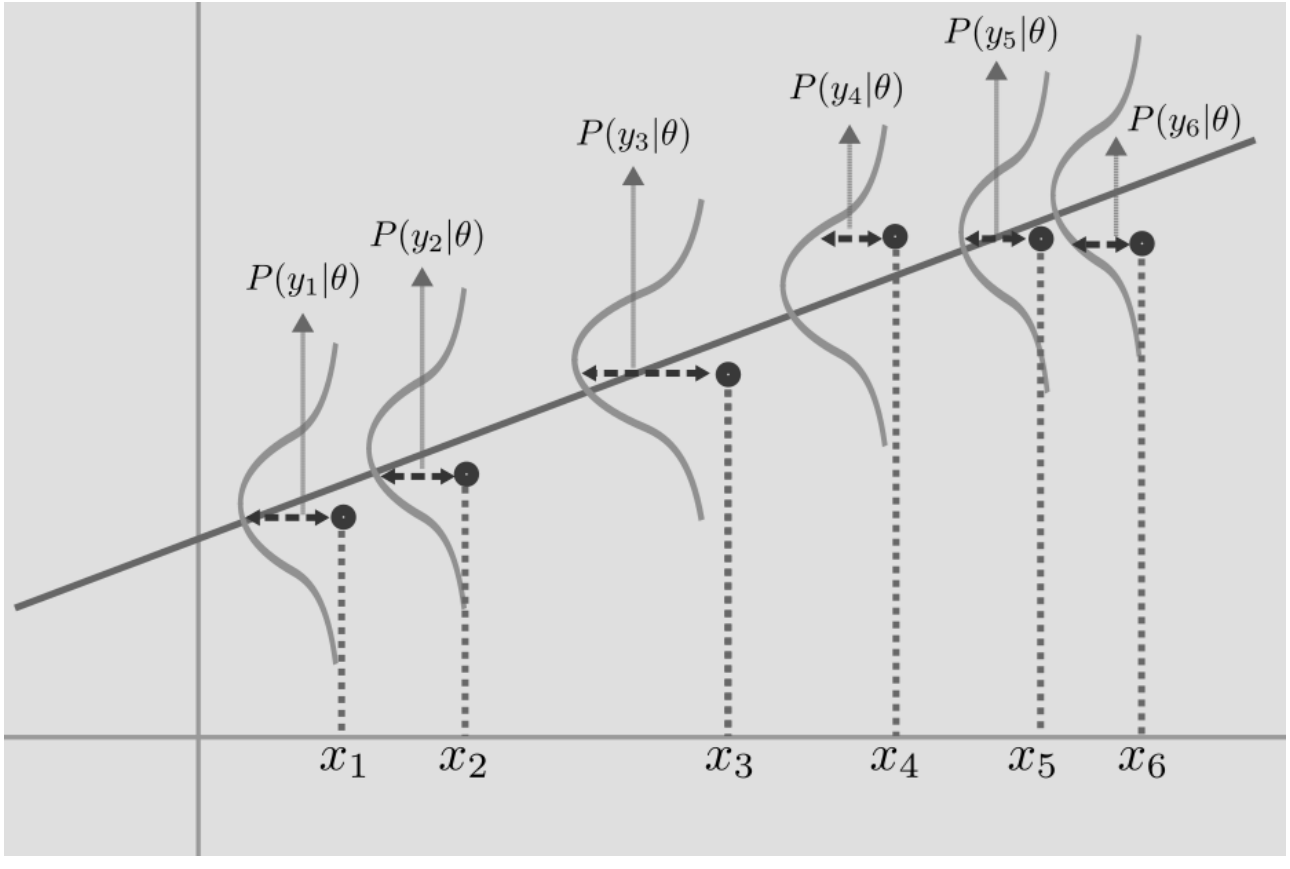
The most common measure of fit between the line and the data is the **least-squares fit**.

There is a good reason for this: If the points are generated by an ideal line with additive Gaussian noise, the least squares solution is the ***maximum likelihood solution***.

Probability of a point y_j is $\Pr(y_j) = \exp\left(\frac{-(y_j - X_j\beta)^2}{2\sigma^2}\right)$ and the probability for all points is the product over j of $\Pr(y_j)$.

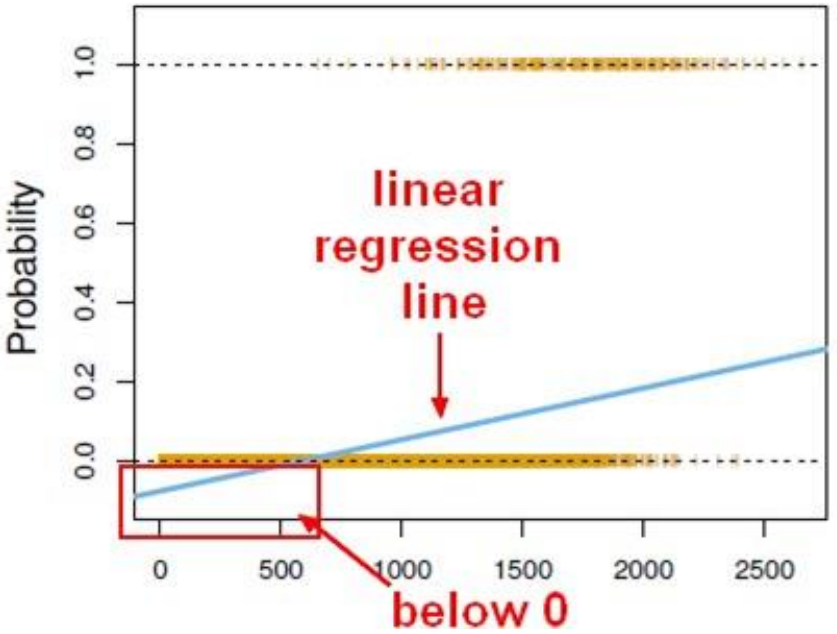
We can **easily maximize the log** of this expression $\frac{-(y_j - X_j\beta)^2}{2\sigma^2}$ for one point, or the sum of this expression at all points.

Linearna regresija



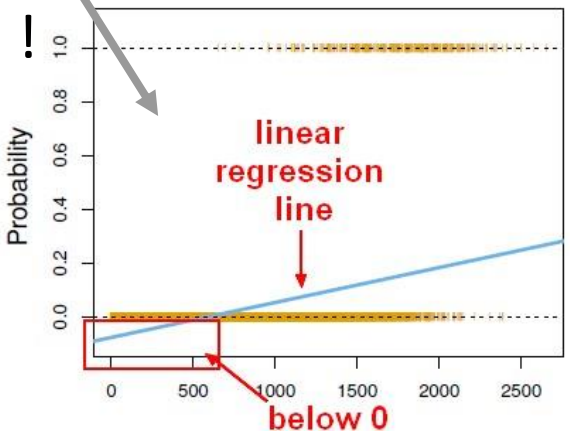
Kako modelirati binarne izlaze?

- Primjer: X - značajke studenta; y: je li student položio MAT?
- Željeni izlaz: f(X) = vjerojatnost prolaska MAT, ako je dan X
- Problem s linearnom regresijom:
  f(X) može biti ispod 0 ili iznad 1

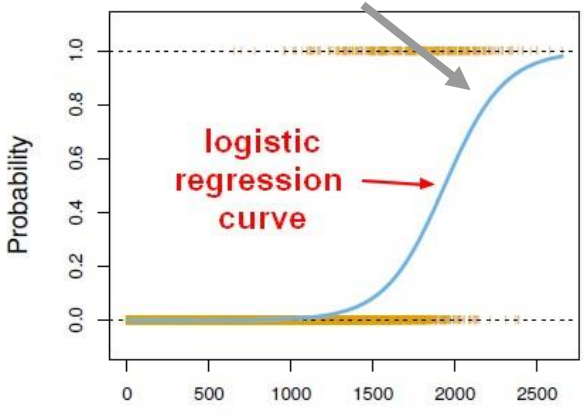


Logistička regresija

Loše !



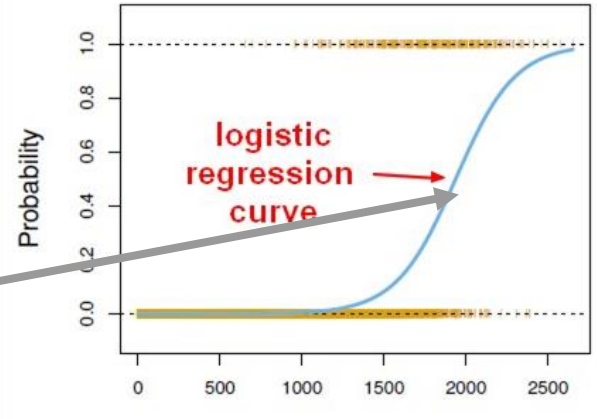
Ovo želimo!



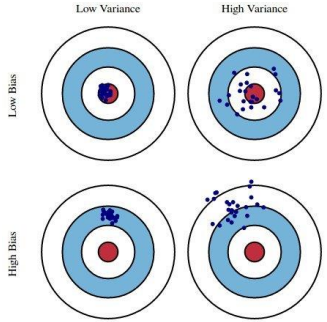
- Trick: Ne raditi s vjerojatnostima koje idu od 0 do 1 nego s *log odds*, koji su u rasponu -inf do +inf
- Vjerojatnost y ⇔ odds y/(1-y) ⇔ log odds log[y/(1-y)]
- Model *log odds* kao linearna funkcija X, *log odds* – LOGARITAM OMJERA ŠANSI

Logistička regresija

- Model *log odds* kao linearna funkcija X
- $\beta^T X = \log[y/(1-y)]$
- Rješenje za y: $y = 1 / (1 + \exp(-\beta^T X))$
- Naći najbolji model i β s maksimalnom vjerodostojnošću:
  - Ne koristi kvadratni gubitak kao u linearnog regresiji
  - Koristi *cross-entropy* funkciju gubitka

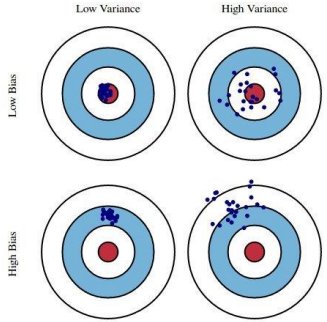


Prenaučenost



- Što više značajki to bolje?
  - **NE!**
  - Više značajki znači manje pristranosti, ali veću varijancu
    -> vodi u prenaučenost

Prenaučenost



- Što više značajki to bolje?
  - **NE!**
  - Više značajki znači manje pristranosti, ali veću varijancu
  - Vodi u prenaučenost
- Pažljiv odabir značajki može poboljšati točnost modela
  - Treba zadržati značajke koje koreliraju s izlazom y
  - Unaprijed/unatrag selekcija
  - Regularizacija (kažnjavanje norme težine vektora)
- Više o takvim praktičnim stvarima u sljedećoj lekciji – primijenjeno SU

Današnje teme

- Primjeri primjene
- Uvod u algoritam k najbližih susjeda (k-NN) i dilema pristranost-varijanca
- Stabla odluke, slučajne šume i *boosted trees*
- Linearna i logistička regresija
- O prenaučenosti

Ovo predavanje temelji su na nastavnim materijalima predmeta *Applied Data Analysis* (ADA) EPFL-a, autora Roberta Westa.