

# Soft kompjuting

Istorija i osnovni pojmovi

# Istorija – Deduktivno zaključivanje

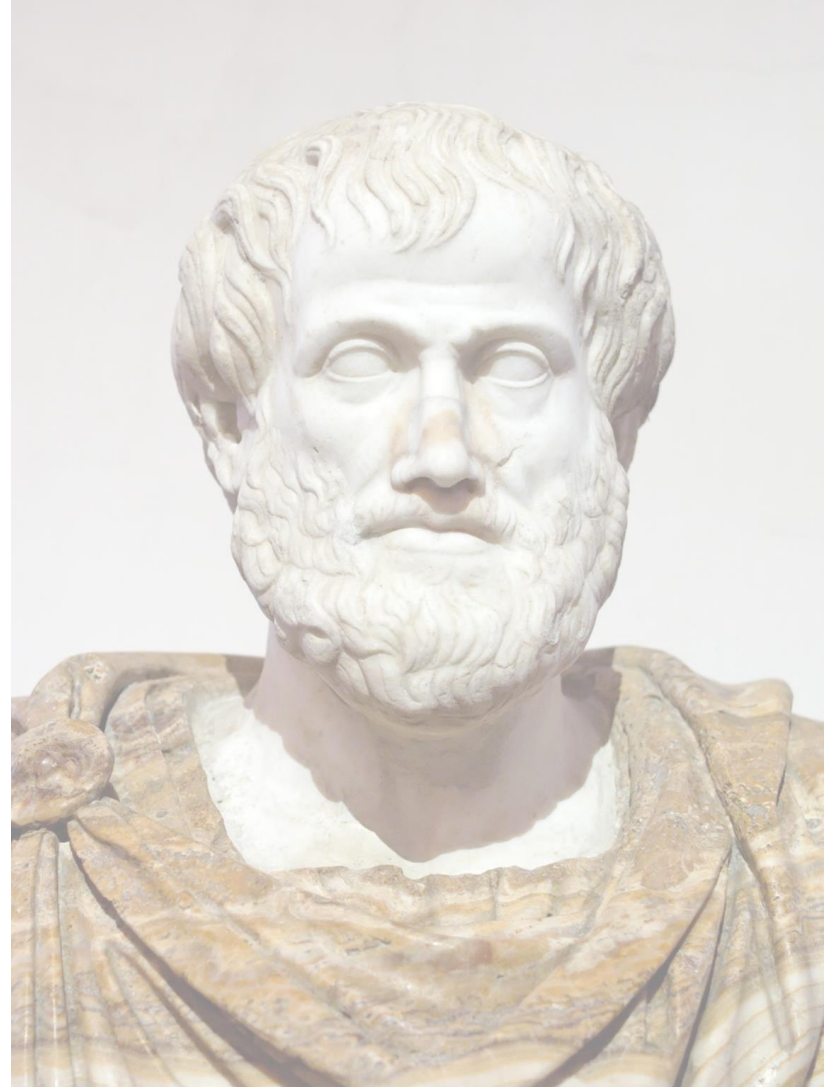
- Aristotel (384-322 p. n. e.)
  - Deduktivno zaključivanje (syllogism)
  - Rezonovanje o prirodnom svetu

A: Svi ljudi su smrtni

B: Sokrat je čovek

Na osnovu A i B možemo zaključiti:

C: Sokrat je smrtan



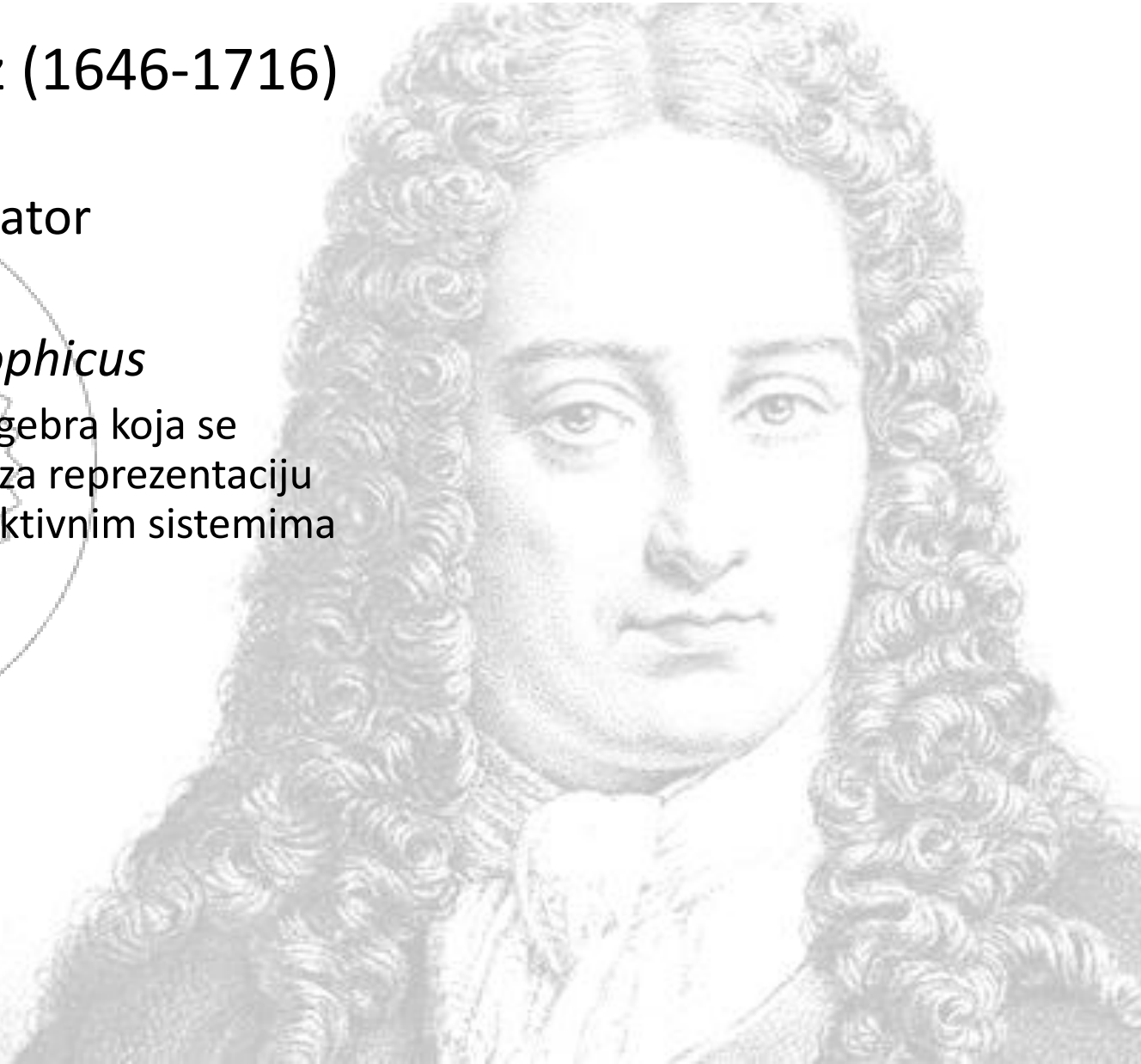
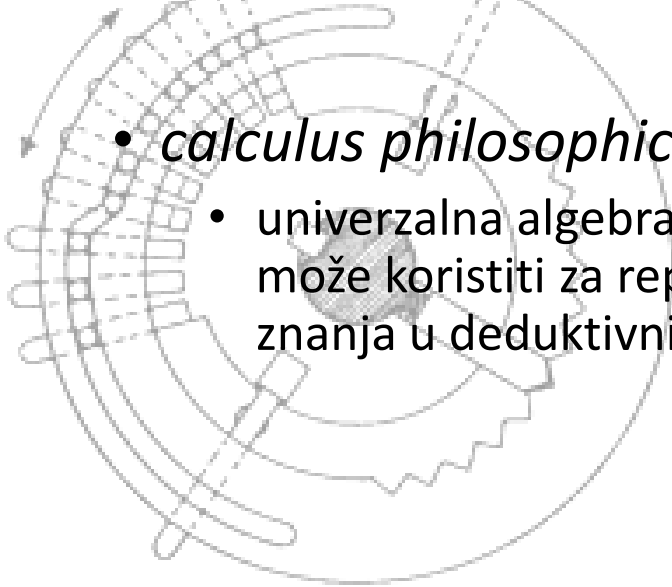
# Istorija

- Gottfried Leibniz (1646-1716)

- Leibniz wheel
- Pinwheel calculator

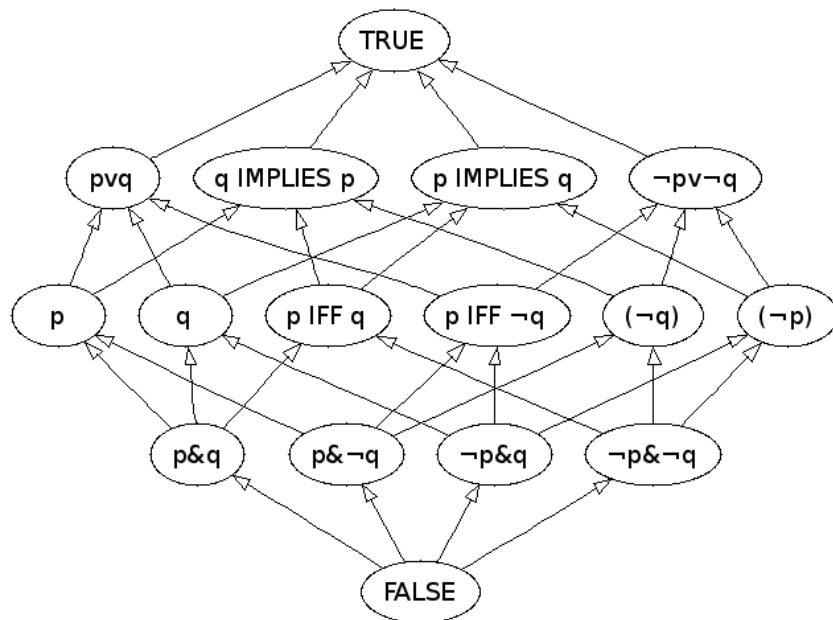
- *calculus philosophicus*

- univerzalna algebra koja se može koristiti za reprezentaciju znanja u deduktivnim sistemima



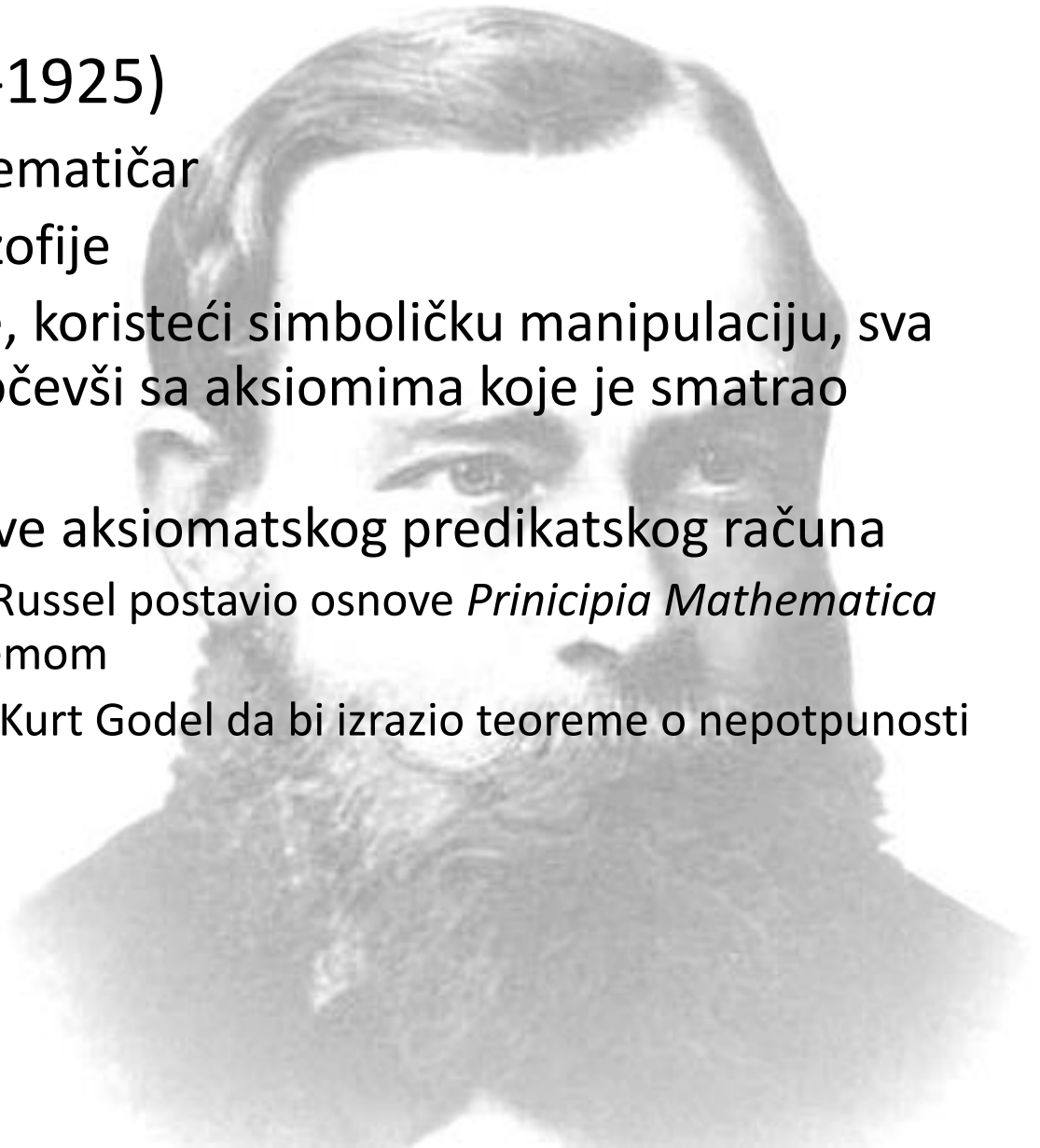
# Istorija

- George Boole (1815-1864)
  - 1854. postavio osnove propozicione logike
  - boolean (tip logičke promenjive)
  - Robustan sistem za pamćenje i rezonovanje o logičkim operacijama



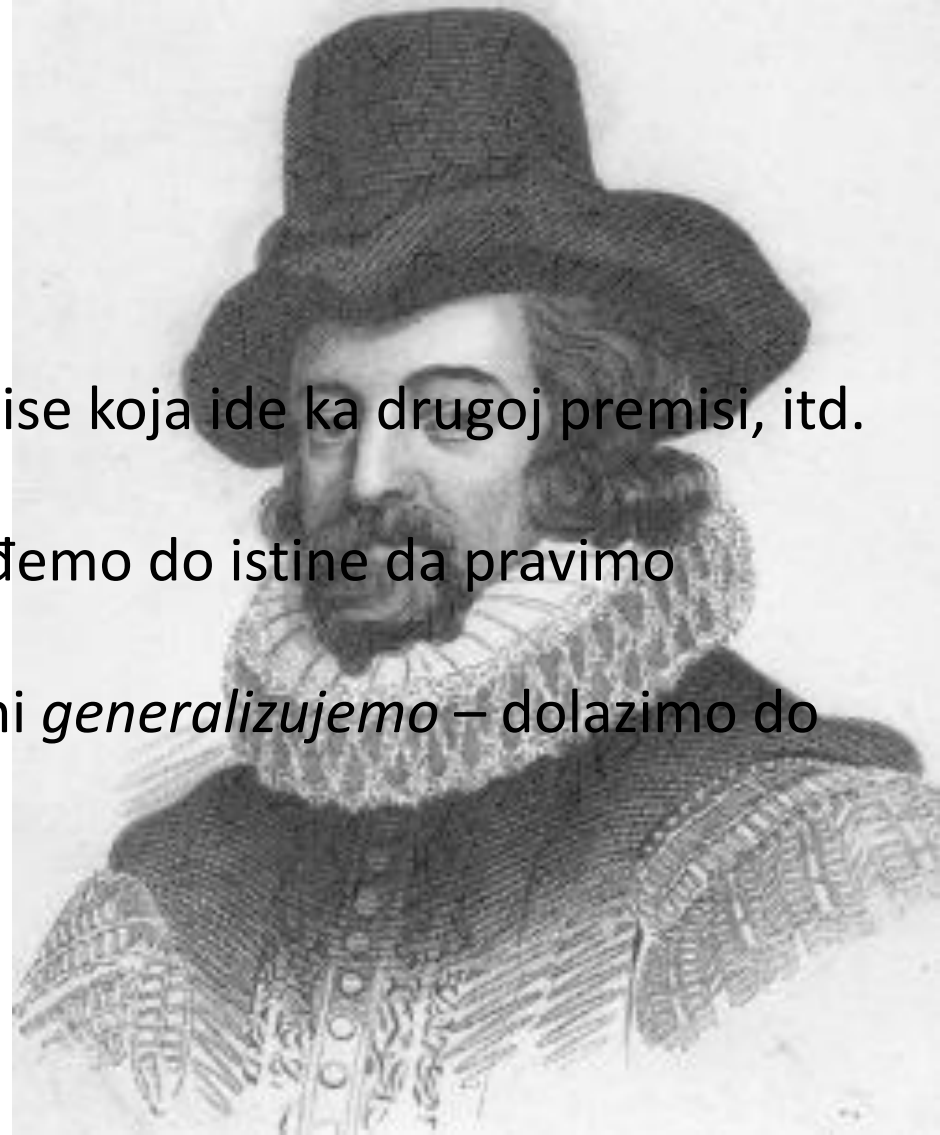
# Istorija

- Gottlob Frege (1848-1925)
  - Filozof, logičar i matematičar
  - Pionir analitičke filozofije
  - Pokušavao da izvede, koristeći simboličku manipulaciju, sva pravila aritmetike počevši sa aksiomima koje je smatrao očiglednim
  - 1879. postavio osnove aksiomatskog predikatskog računa
    - Kasnije je Bertrand Russel postavio osnove *Principia Mathematica* inspirisan ovim sistemom
    - Sistem je ključan za Kurt Godel da bi izrazio teoreme o nepotpunosti



# Istorija – Induktivno zaključivanje

- Sir Francis Bacon
  - Engleski filozof iz 17. veka
- Induktivno zaključivanje
  - Umesto da krenemo od premise koja ide ka drugoj premisi, itd. dok ne dođemo do zaključka,
  - Bacon kaže da je način da dođemo do istine da pravimo ponovljene opservacije
  - Na osnovu ovih opservacija mi *generalizujemo* – dolazimo do *verovatnog* zaključka



# Razlika deduktivnog i induktivnog zaključivanja

---

Generalizacija

DEDUKTIVNO  
*Top Down*

INDUKTIVNO  
*Bottom Up*

Specifična instanca

# Empiricizam

---

- Što više podataka imamo – veća je verovatnoća da su naši zaključci tačni

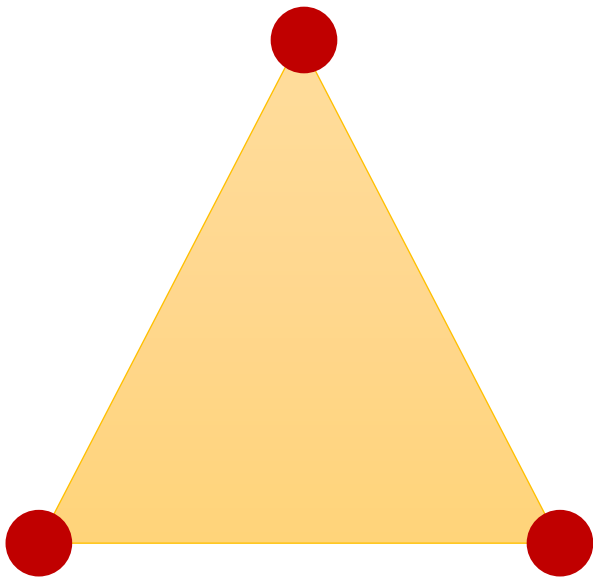




# Empiricizam

---

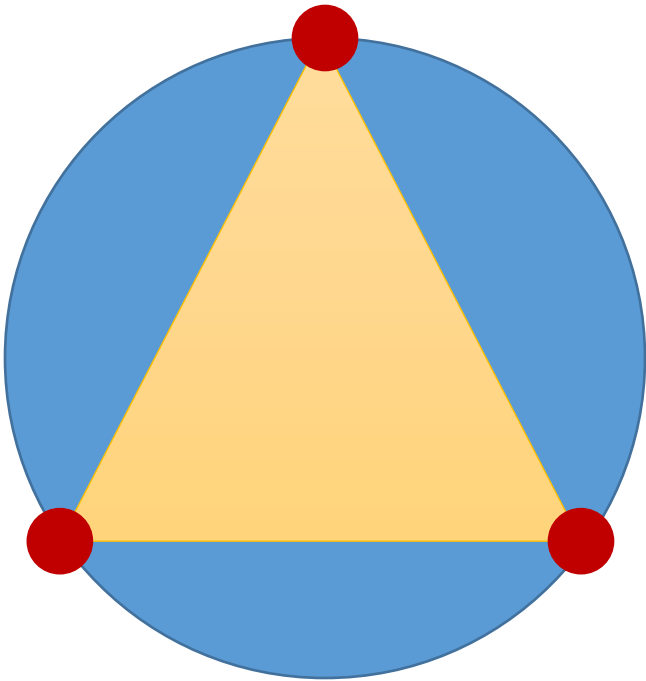
- Što više podataka imamo – veća je verovatnoća da su naši zaključci tačni



# Empiricismizam

---

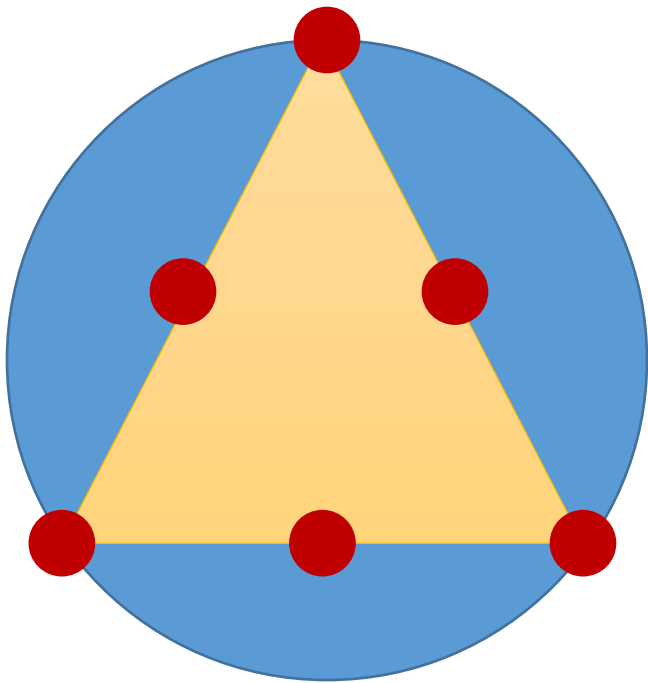
- Što više podataka imamo – veća je verovatnoća da su naši zaključci tačni



# Empiricizam

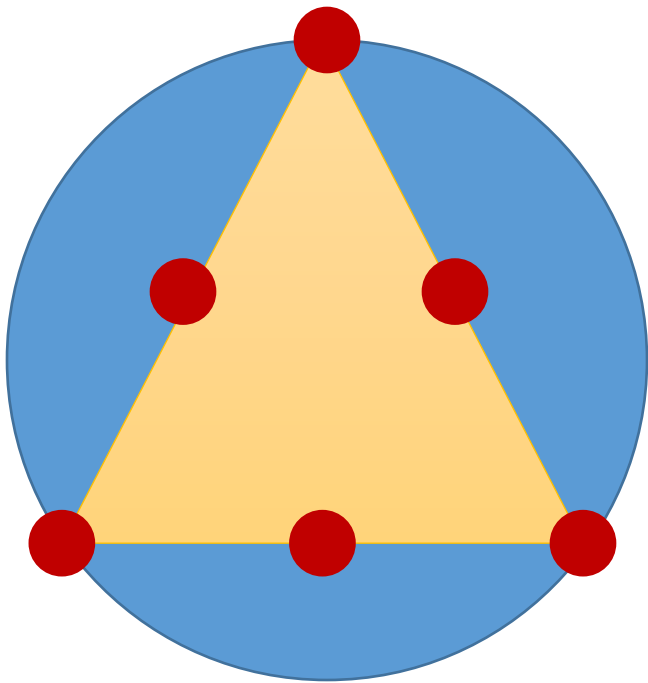
---

- Što više podataka imamo – veća je verovatnoća da su naši zaključci tačni



# Empiricizam

- Što više podataka imamo – veća je verovatnoća da su naši zaključci tačni



David Hume  
“Skeptical Scotsman”

Indukcija nas nikada ne može dovesti do *sigurnih* zaključaka jer pretpostavlja da će stvari uvek ostati iste

# Različiti ciljevi

---

- Deduktivno zaključivanje → sigurnost
- Induktivno zaključivanje → verovatnoća

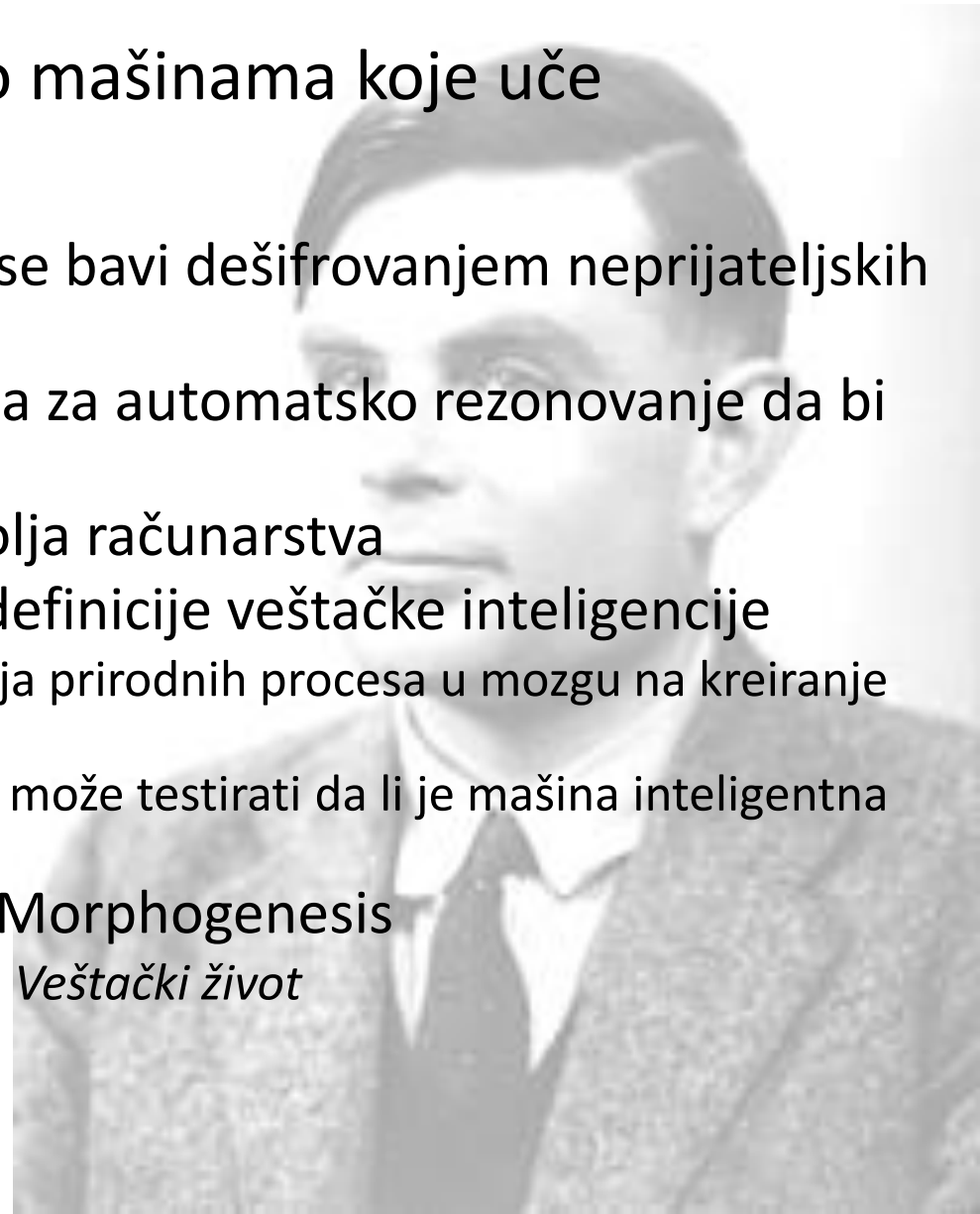
# Na koji način učimo?

---

- *Pre hiljadu godina* nauka je empirična, opisuje prirodne pojave
- *Pre nekoliko stotina godina* nauka postaje teoretska, koristi modele, aproksimacije i uopštenja
- *Nekoliko poslednjih decenija* nauka postaje sve više bazirana na izračunljivim pristupima, razne računarske simulacije
- *Danas* se nauka bavi istraživanjem podataka
  - Podaci se automatski skupljaju ili generišu simulatorima
  - Podaci se automatski obrađuju korišćenjem softvera

# Istorija

- 1950 Alan Turing razmišlja o mašinama koje uče
- Alan Turing (1912-1954)
  - Tokom drugog svetskog rata se bavi dešifrovanjem neprijateljskih poruka
  - Razvija jednu od prvih mašina za automatsko rezonovanje da bi uspeo u dešifrovanju poruka
  - Ključna osoba u osnivanju polja računarstva
  - 50'-ih godina postavio prve definicije veštačke inteligencije
    - Izučavao mogućnosti korišćenja prirodnih procesa u mozgu na kreiranje mašina
    - Definisao test prema kome se može testirati da li je mašina inteligentna ili ne
  - 1952. The Chemical Basis of Morphogenesis
    - osnove za oblast pod nazivom *Veštački život*



# Soft Computing

---

- Primena računarske obrade podataka nad problemima koji
  - Nemaju tradicionalno rešenje
    - Rešenje nije moguće izraziti u analitičkom obliku
    - Analitički oblik rešenja nije pogodan za implementaciju
  - Moraju biti robusni na šum, i operaciju u stvarnom svetu
    - Senzori iz kojih se generišu podaci su po definiciji zašumljeni
    - Znanje koje koristimo u obradi podataka često nije jasno određeno



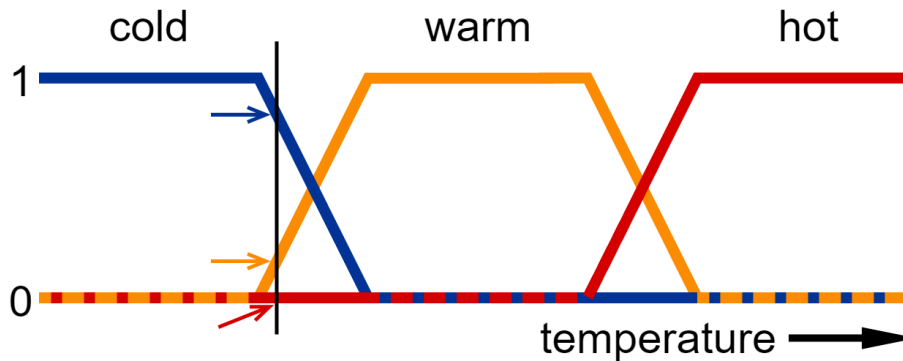
# Koncepti koje Soft Computing obuhvata

---

- Fuzzy koncepti
  - Fuzzy skupovi
  - Fuzzy logika
- Evolutivno računarstvo (*Evolutionary computation*)
  - Genetski algoritmi (*Genetic algorithms*)
  - Inteligencija mnoštva (*Swarm intelligence*)
- Mašinsko učenje
  - Neuronske mreže (*Neural Networks*)
  - Mašine potpornog vektora (*Support Vector Machines*)
  - Probabilističko rezonovanje

# Fuzzy koncepti

- Relacijama se dodeljuje vrednost tačnosti iz  $[0,1]$  uzimajući u obzir nesigurnost



- Znanje je eksplicitno reprezentovano modelom i relacijama izmedju koncepata
- Širok spektar primena
  - Reprezentaciju koncepata, znanja, relacija
  - Kontrolu industrijskih pogona
  - Medicinsku dijagnostiku

# Fuzzy koncepti

- Fuzzy koncepti
  - 1920. Lukasiewicz (two-valued  $\rightarrow$  three-valued logic)
  - 1965. Lotfi Zadeh, Fazi skup
  - 1974. Lotfi Zadeh, Fazi logika
  - 1975. Mamdani, Fazi kontrola
  - 1985. Sugeno, Fazi zaključivanje



# Fuzzy koncepti

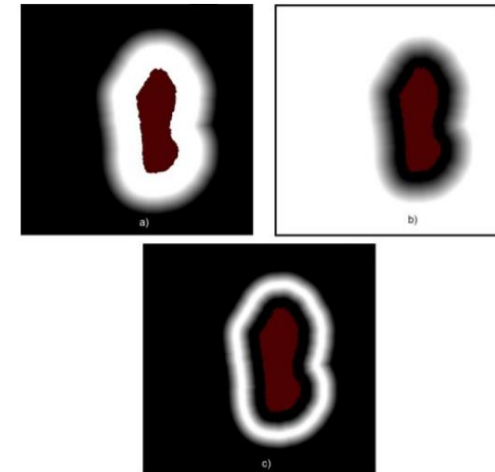
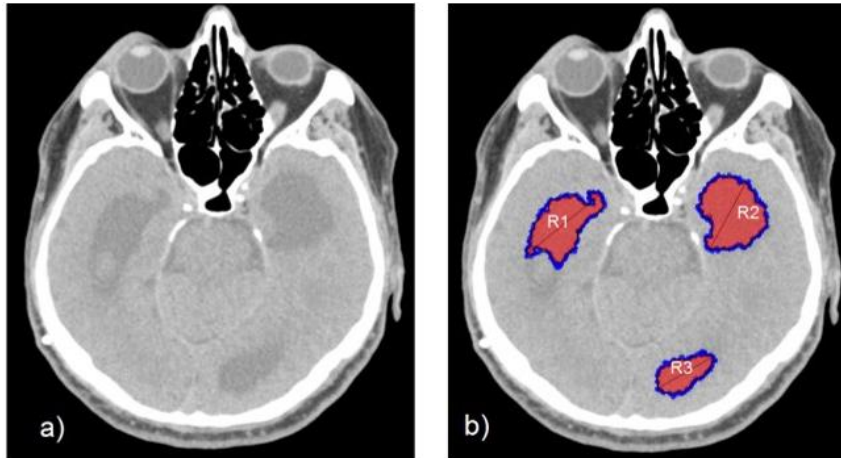


Figure 8 - Examples of fuzzy spatial relations a) *CloseTo*, b) *FarFrom* and c) *AtDistanceOf*

Jocić, M., Obradović, Đ., Konjović, Z. and Pap, E., 2013, September. 2D fuzzy spatial relations and their applications to DICOM medical images. In *Intelligent Systems and Informatics (SISY), 2013 IEEE 11th International Symposium on* (pp. 39-44). IEEE

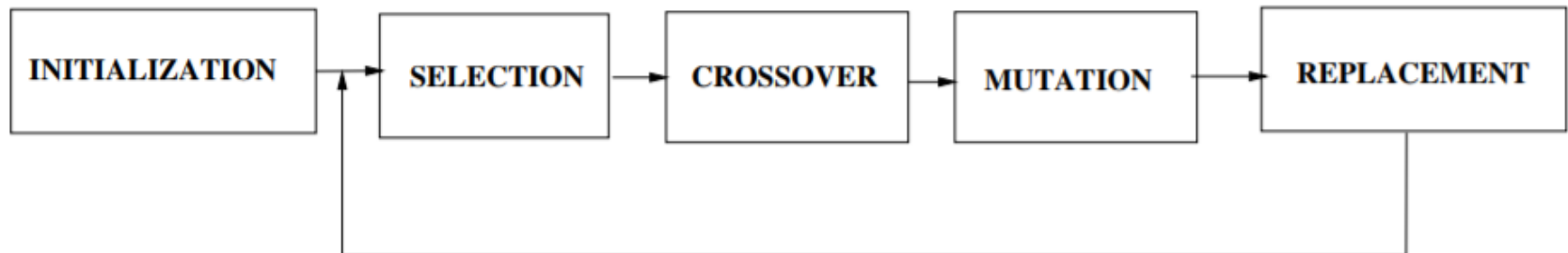
**In this paper we propose a model of fuzzy spatial relations, which can be used to describe imprecise spatial data on 2D images. This model defines relations very similar to those existing in natural language and reasoning, like “on the left side of the image”, “in the center of the image”, “right/left of”, “above/below of”, “(very) close to”, “(very) far from”, “at distance of”, etc. Composition operators and, or, not and subtract are defined in order to combine these basic fuzzy spatial relations into more complex ones. In addition, formula for calculating membership value of imprecise region to fuzzy spatial relation is defined**

# Evolutivno računarstvo

- Familija algoritama za optimizaciju inspirisana biološkom evolucijom



- Pravljenje modela se sastoji opisa prostora koji se direktno preslikava na rešenja problema
- Potrebno je efikasno pretražiti prostor rešenja da bi se našlo optimalno rešenje
- Pretražujemo prostor rešenja inspirisani prirodnom evolucijom
  - Fitness funkcija opisuje koliko dobro je rešenje problema
  - Genotip opisuje jedno rešenje problema
  - Populacija je skup genotipa
  - Moguće poboljšanje fitnessa: mutacija i kombinovanje



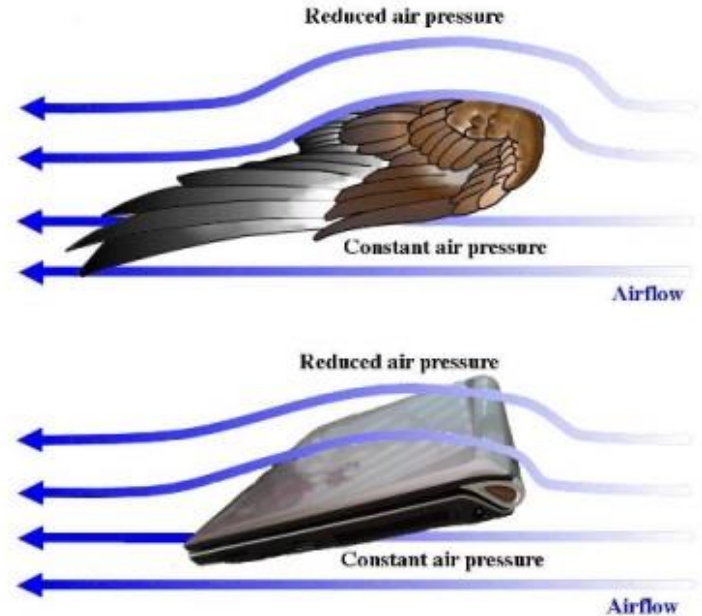
# Evolutivno računarstvo

---

- Rezultat učenja je genotip ili jedinka koja posebno dobro rešava problem
- Proizvode prilično optimalna rešenja za širok stepen problema
- Često se koristi kada nema drugih pristupa rešavanju problema “black box”
- Ali često ne rezultuju globalnim optimumom

# Evolutivno računarstvo

- Fogel – uvodi evolutivno programiranje, kasnije Holland naziva njegov metod genetski algoritam
- 1960 Fraser, simulacija genetike i populacija
- 1960-1970 Ingo Rechenberg – prva ozbiljna tehnička aplikacija veštačke evolucije: aerodinamičan dizajn krila
- Holland je zaslužan za popularizaciju
- Danas: Image processing, Computational Fluid Dynamics, Reservoir Optimisation in oil-fields, Weather prediction, Evolutionary Robotics, Medical Imaging, Financial Markets, Protein Folding, Vehicle Routing...



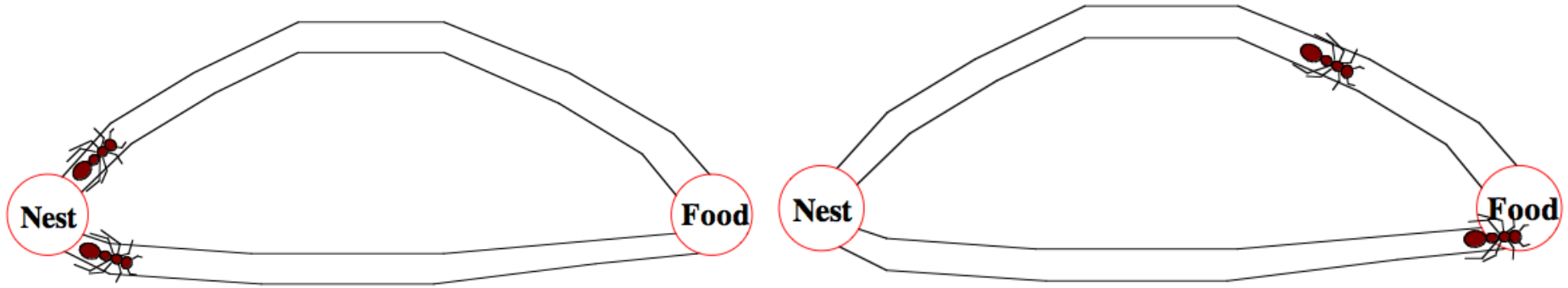
# Inteligencija mnoštva

---

- Kolektivno ponašanje mnoštva autonomnih, samo organizovajućih agenata
- Inspirisani biološkim sistemima kao što su kolonije bakterija, mrava ili jata ptica
- Mnoštva jedinki koje pojedinačno prate prosta pravila, ne postoji centralizovana kontrola
- Mnošto prostih jedinki sa prostim ponašanjem sarađuje da bi rešili komplikovane probleme iz specifičnih domena



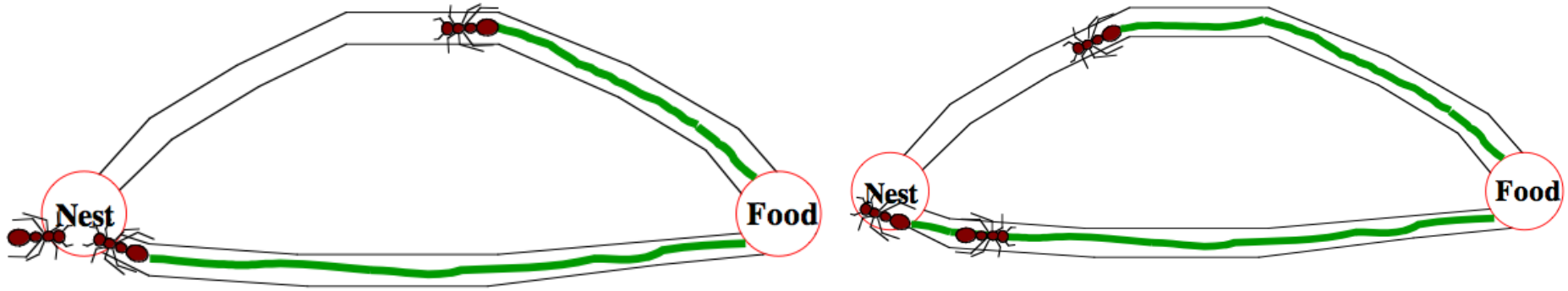
# Mravlji algoritmi – biološka inspiracija



Dva mrava imaju jednaku šansu da pronađu hranu

Mrav koji se kreće kraćim putem pre stiže na cilj

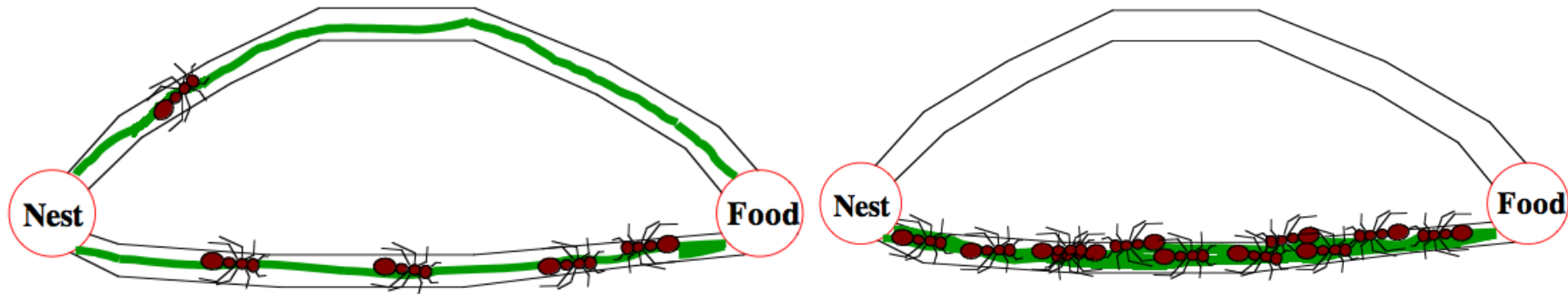
# Mravlji algoritmi – biološka inspiracija



- Intenzitet feromona zbog dva prolaza izraženiji je kod kraće putanje.

- Sledeći mrav bira kraću putanju

# Mravlji algoritmi – biološka inspiracija



- Posle nekog vremena veći deo mrava kreće se kraćim putem
- Postoje mravi koji ne idu tim putem
- U jednom trenutku mravi se koncentrišu na jednom putu

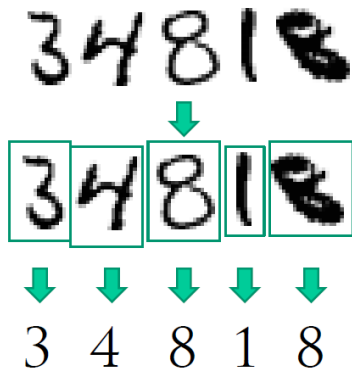
# Neuronske mreže

---

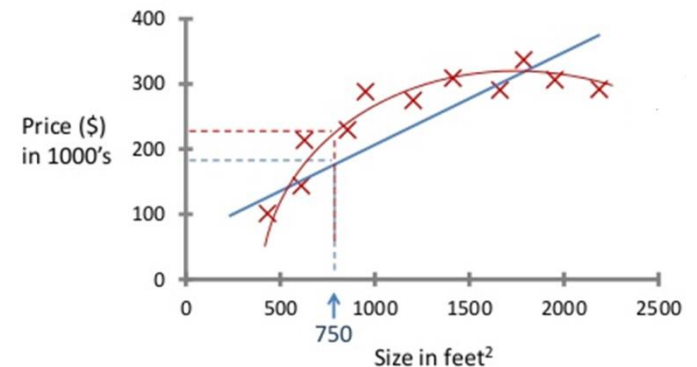
- 1943 McCulloch & Pitts *threshold logic*, preteča neuronskih mreža
  - Pokušaj da se razume kako mozak postiže komplikovano ponašanje iz prostih delova
  - Prave model neurona sličan perceptronu
- 1957 Rozenblat je napravio (u hardveru) perceptron
- 1969 Marvin Minski i Sejmur Papert kritikuju perceptron, što usporava razvoj neuronskih mreža
- 1975 Verboš formuliše algoritam propagacije unazad (*backpropagation*) za treniranje neuronskih mreža
- 1985 Sejnovski i Rozenberg prave sistem koji uči da izgovara engleske reči zasnovan na neuronskim mrežama

# Machine learning

- Učenje na osnovu iskustva u vidu istorijskih ishoda ili mera o nekom procesu, sa ne označenim ili označenim podacima



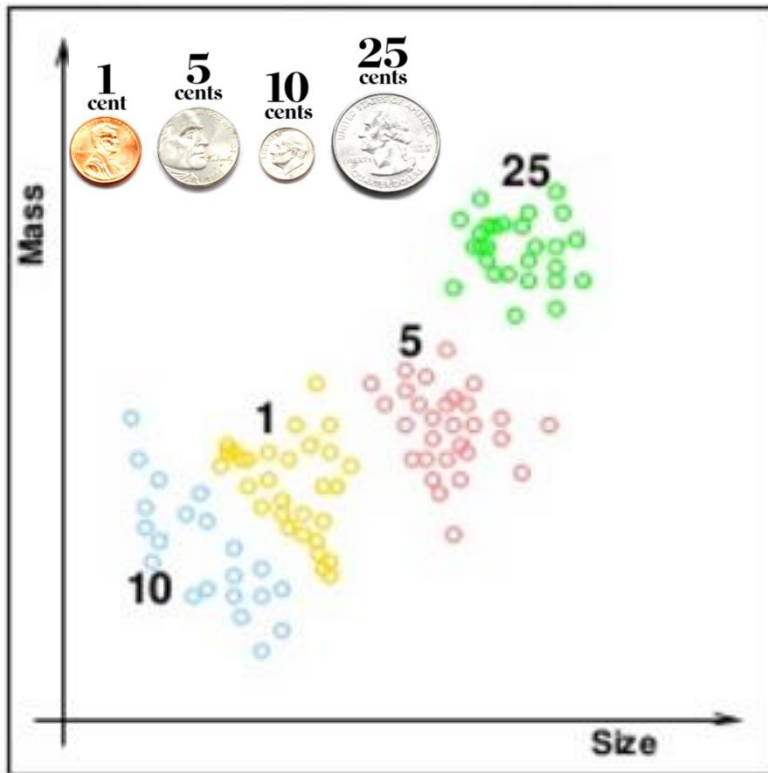
Housing price prediction.



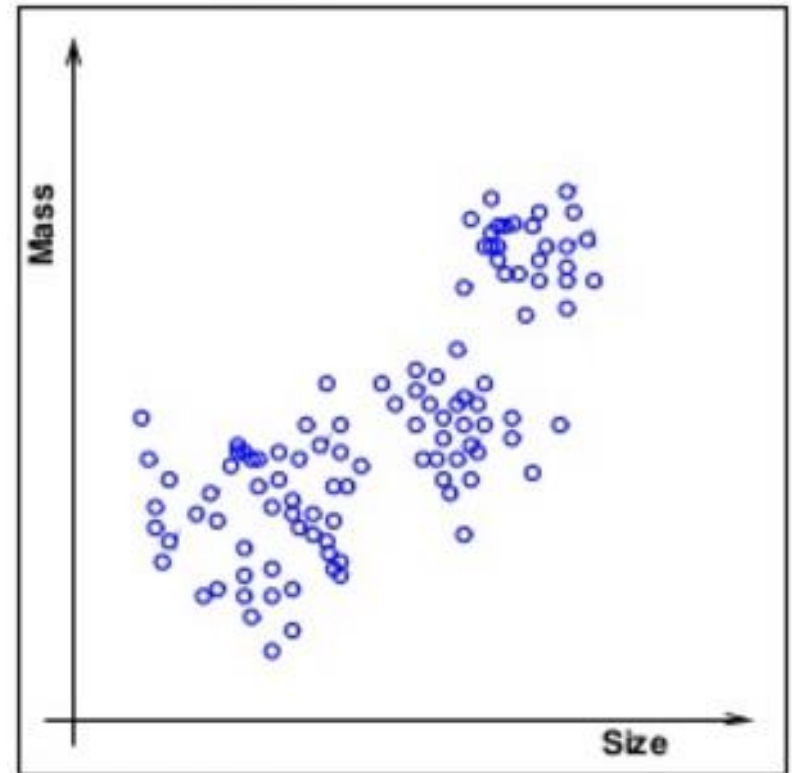
Označeni podaci

→ Nadgledano obučavanje (*Supervised learning*)

# Machine learning



Označeni podaci



Neoznačeni podaci  
→ Nenadgledano obučavanje  
(*Unsupervised learning*)

# Učenje – načini učenja

---

- Označeni podaci – nadgedano obučavanje (*supvised learning*)
  - Poznata nam je ciljna labela  $y$
  - Obično je lako ustanoviti odgovarajuću metriku greške
- Neoznačeni podaci – nenadgledano obučavanje (*unsupervised learning*)
  - Nemamo  $y$  – tražimo neku strukturu u podacima
  - Teže je proceniti metriku ishoda
  - Najčešće čovek vizualno može dobro da proceni da li su podaci organizovani na odgovarajući način
- Učenje sa povratnom spregom (*reinforcement learning*)

# Učenje

---

Nepoznata ciljna funkcija  
 $f: X \rightarrow Y$



Trening skup  
 $T = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(N)}, y^{(N)})\}$

Konačna hipoteza  
 $g \approx f$



# Učenje

---

Nepoznata ciljna funkcija  
 $f: X \rightarrow Y$

želimo da napravimo funkciju  $f$   
koja mapira sliku  $X$  na cifru  $Y$

Trening skup

$$T = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(N)}, y^{(N)})\}$$

Konačna hipoteza  
 $g \approx f$

# Učenje

---

Nepoznata ciljna funkcija  
 $f: X \rightarrow Y$



Trening skup  
 $T = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(N)}, y^{(N)})\}$

Konačna hipoteza  
 $g \approx f$

# Učenje

Nepoznata ciljna funkcija  
 $f: X \rightarrow Y$



Trening skup

$$T = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(N)}, y^{(N)})\}$$



3 4 8 1 8

Labela  $y$

- Vektor  $x \rightarrow$  pretprocesiranje:
1. skaliramo/transliramo sliku tako da sve staju u isti okvir  $m \times n$
  2. sliku predstavimo kao vektor od  $m \times n$  realnih brojeva
  3. ...

Konačna hipoteza  
 $g \approx f$

# Učenje

---

Nepoznata ciljna funkcija  
 $f: X \rightarrow Y$



Trening skup  
 $T = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(N)}, y^{(N)})\}$

Konačna hipoteza  
 $g \approx f$

# Učenje

Nepoznata ciljna funkcija  
 $f: X \rightarrow Y$



Trening skup  
 $T = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(N)}, y^{(N)})\}$

Kao rezultat želeli bismo da damo konačnu hipotezu – formulu koju ćemo koristiti da sliku pretvorimo u cifru

Konačna hipoteza  
 $g \approx f$

# Učenje

---

Nepoznata ciljna funkcija  
 $f: X \rightarrow Y$



Trening skup  
 $T = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(N)}, y^{(N)})\}$

Konačna hipoteza  
 $g \approx f$

# Učenje

Nepoznata ciljna funkcija  
 $f: X \rightarrow Y$



Trening skup  
 $T = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(N)}, y^{(N)})\}$

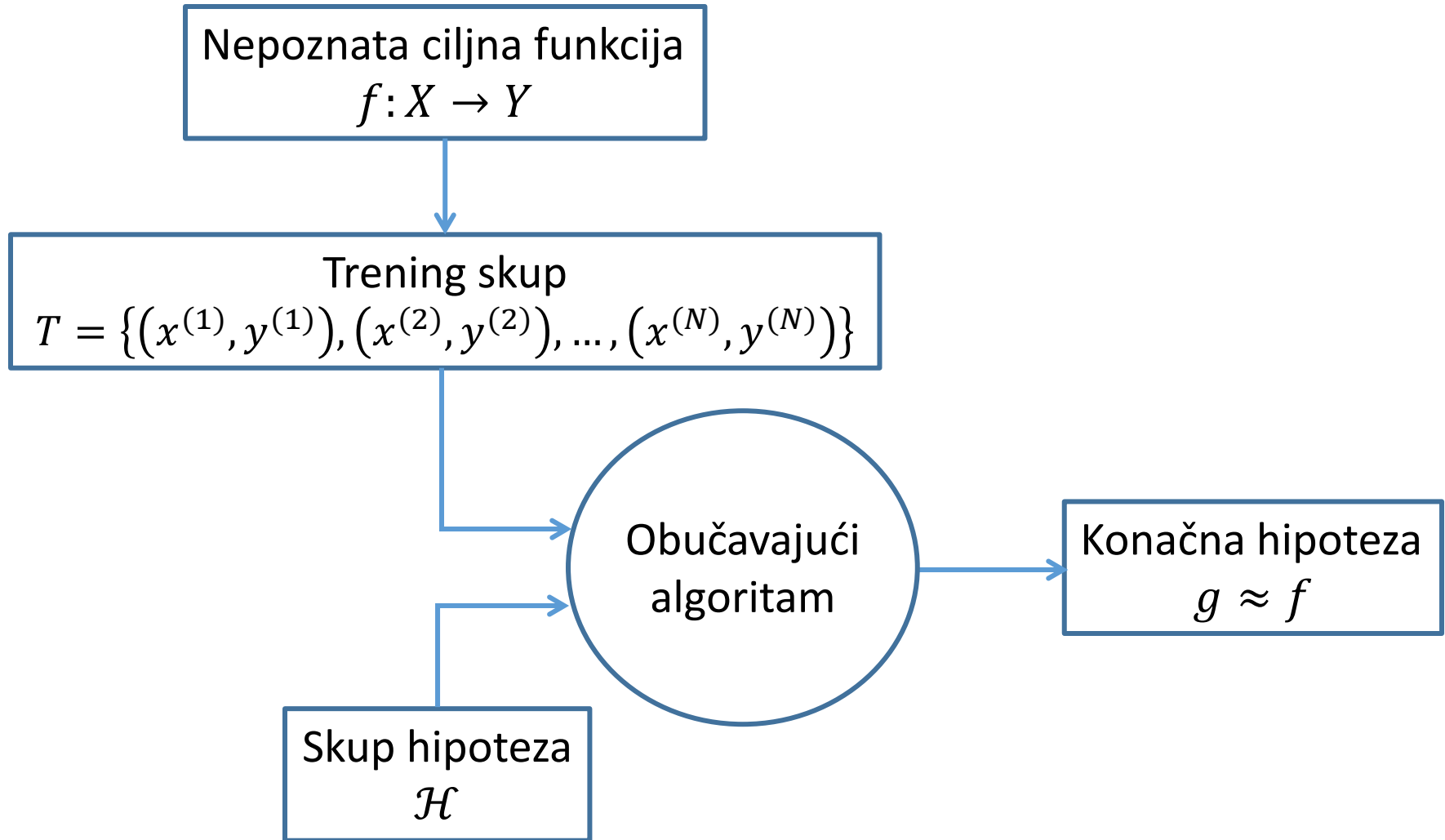


Obučavajući  
algoritam



Konačna hipoteza  
 $g \approx f$

# Učenje





# Evaluacija – tipične mere performanse

---

- Regresija:
  - MSE – zbir kvadrata greške nad svim primerima
  - $R^2$  (koeficijent determinacije)
- Klasifikacija:
  - Tačnost (*accuracy*) – procenat tačno predikovanih vrednosti u odnosu na ukupan broj testova
  - Međutim, zavisno od problema, tačnost ne mora biti dobra mera. Uzmite u obzir veoma nebalansirane klase, na primer, 99% primera pripada jednoj klasi, a svega 1% drugoj. Naivni klasifikator koji sve svrstava u većinsku klasu ima 99% tačnosti. Problem je još više izražen kod višekategorijske klasifikacije
  - U ovom slučaju obično se koriste *f*-mera (*f-measure*) ili preciznost/odziv (*precision/recall*)

# Evaluacija – mera performanse

---

- Način na koji se računa i definiše *fitness* zavisi od problema
  - Na primer, kod *traveling salesman* problema *fitness* funkcija je dužina pređene staze
- Generalan savet: pre nego što se odlučite za određenu meru performansi, proverite koja se često koristi u literaturi za isti ili sličan problem

# Evaluacija – postupak evaluacije

---

- Osnovno načelo: podaci korišćeni za procenu kvaliteta modela ni na koji način ne smeju biti upotrebljeni prilikom treninga
- Deluje jednostavno, ali se u praksi ispostavlja kao vrlo pipavo

# Kako evaluirati model?

---

- Da li je dobro trenirati model na datom skupu podataka, a zatim na istom skupu podataka evaluirati i grešku?

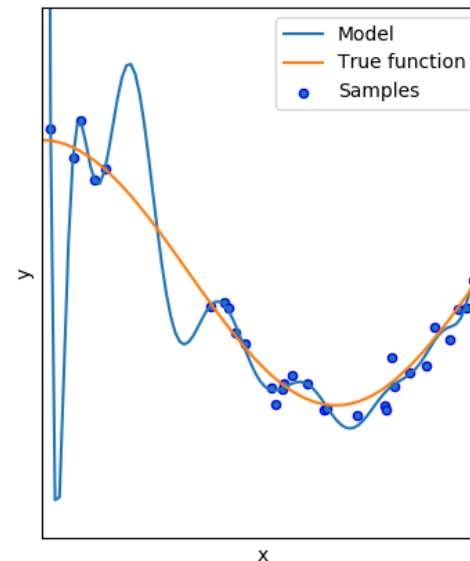
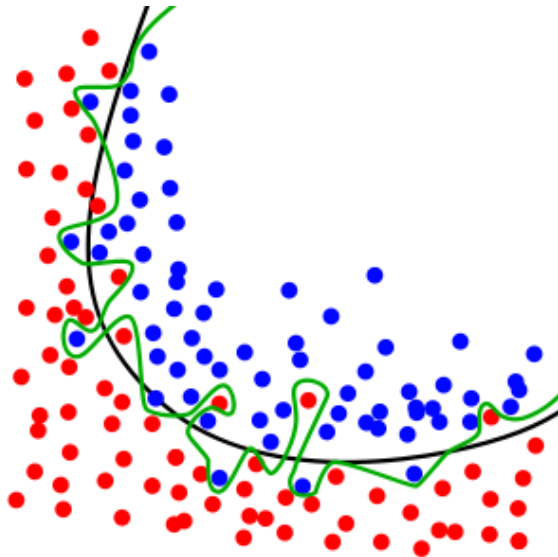
# Kako evaluirati model?

---

- Da li je dobro trenirati model na datom skupu podataka, a zatim na istom skupu podataka evaluirati i grešku?
- Ne, pošto će naša procena greške biti previše optimistična
  - Model se prilagodio baš tim trening podacima
  - Nema garancije da će se novi primeri (koje nismo koristili za obučavanje) ponašati baš identično
  - A nama je stalo do tačne klasifikacije novih primera

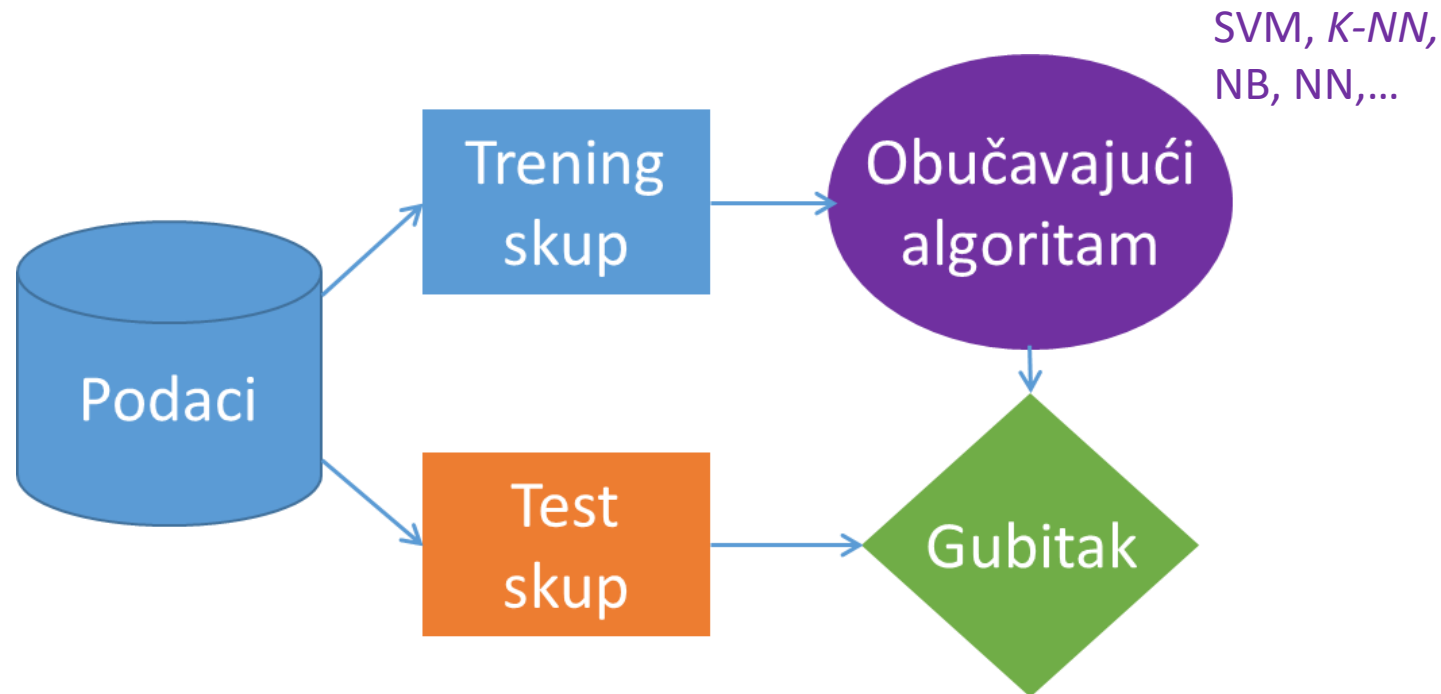
# Kako evaluirati model?

- Da li je dobro trenirati model na datom skupu podataka, a zatim na istom skupu podataka evaluirati i grešku?
- Ne, pošto će naša procena greške biti previše optimistična
  - Model se prilagodio baš tim trening podacima
  - Nema garancije da će se novi primeri (koje nismo koristili za obučavanje) ponašati baš identično
  - A nama je stalo do tačne klasifikacije novih primera



# Kako evaluirati model?

- Podeliti podatke na trening i test skup
  - Uobičajeni odnosi su 90/10, 80/20, 70/30

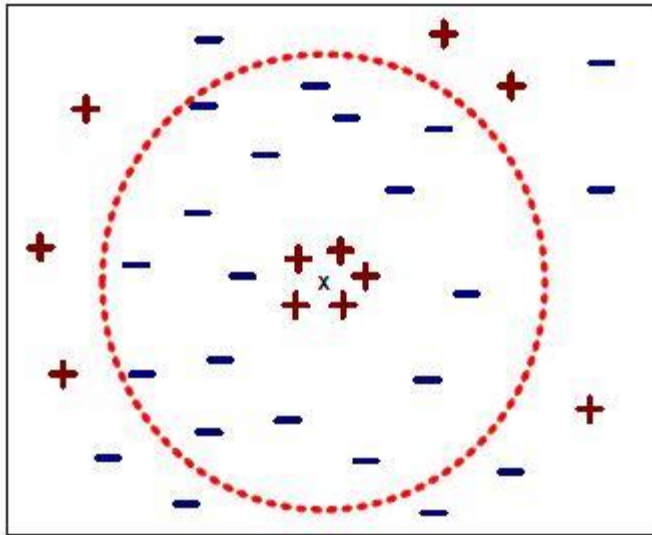


Npr. tačnost (accuracy):

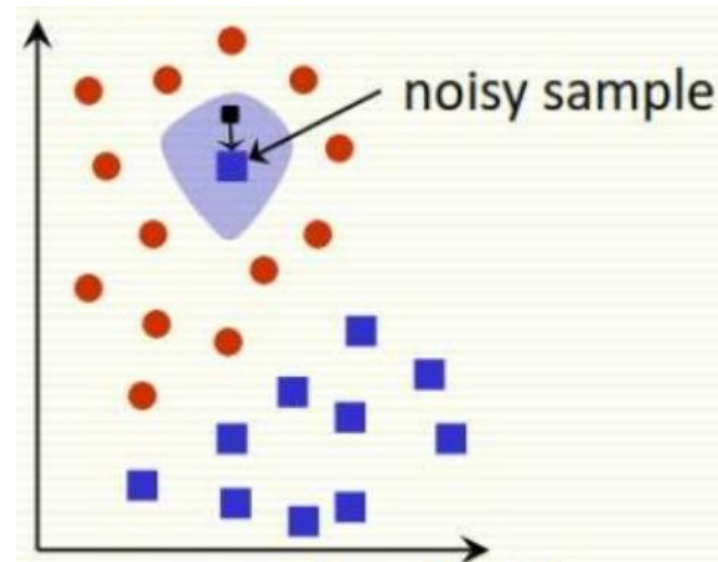
$$\frac{\text{br. tačno klasifikovanih primera}}{\text{ukupan br. primera}}$$

# Optimizacija hiper-parametara modela

- Gotovo svaki model ima hiper-parametre koje treba optimizovati
  - Npr. kod  $K$ -NN treba da odredimo optimalno  $K$  (broj najbližih suseda koje uzimamo u obzir prilikom klasifikacije primera)



Preveliko  $K$



Premalo  $K$  – osetljivost na šum (*outliere*)



# Optimizacija hiper-parametara modela

---

- Hiper-parametre bismo isprobavali
  - Za svaku kombinaciju vrednosti parametra obučiti model i izmeriti njegove performanse. Odabrati one parametre koji rezultuju najboljim performansama
  - Npr. treniramo  $K$ -NN model za  $K = 1, 2, \dots$  i odaberemo  $K$  koje daje najveću tačnost
- Ali na kojim podacima trenirati model a na kojim meriti performanse modela?

# Optimizacija hiper-parametara – pogrešno

## 1. Selekcija modela

- Za svaku moguću kombinaciju vrednosti hiper-parametara:
  - i. Trenirati model sa tim vrednostima hiper-parametara na **trening** podacima
  - ii. Proceniti performanse dobijenog modela na **test** podacima
  - iii. Odabrati vrednosti parametara koje rezultuju najmanjom greškom na **test** skupu

## 2. Evaluacija modela

- Model (sa odabranim vrednostima hiper-parametara) evaluirati na **test** skupu
- Zašto je ovo pogrešno?

# Optimizacija hiper-parametara – pogrešno

## 1. Selekcija modela

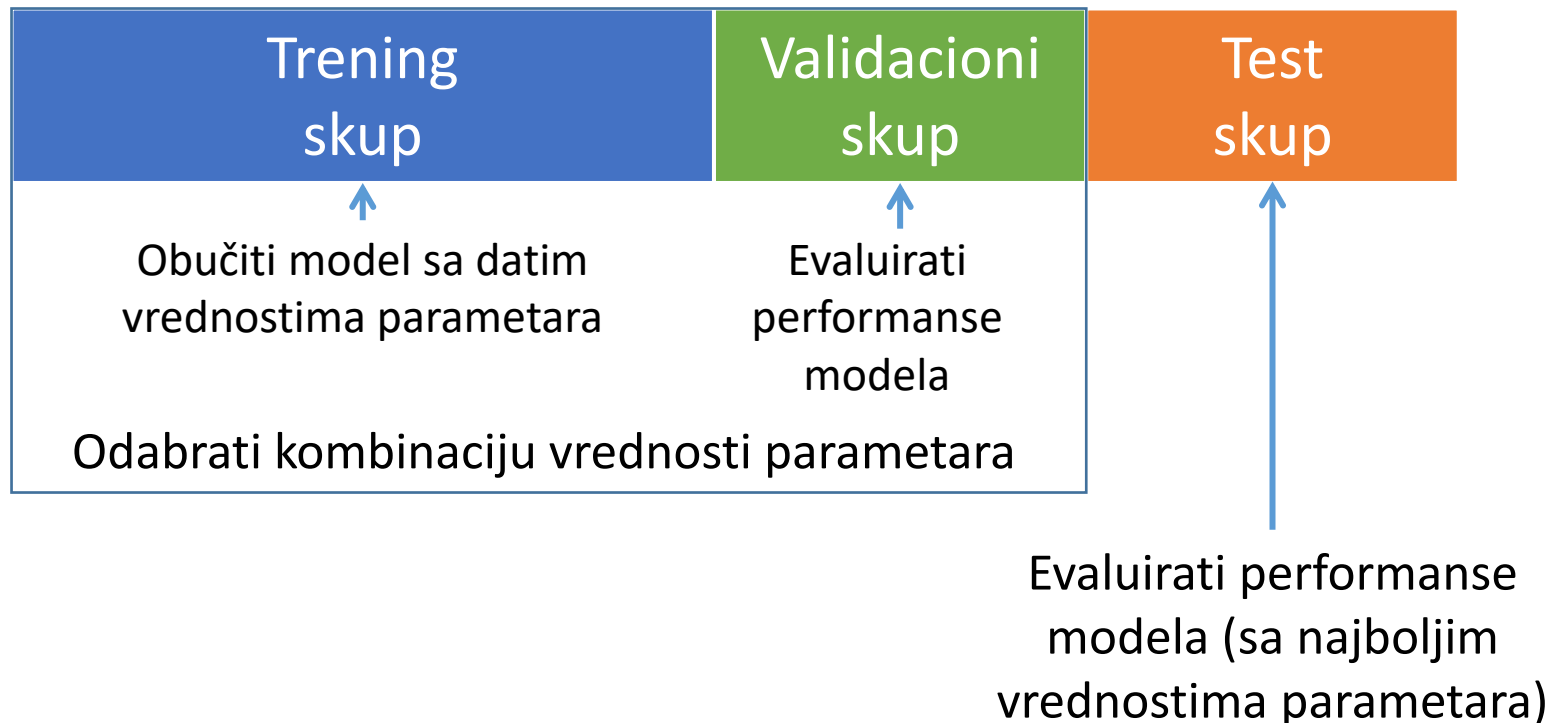
- Za svaku moguću kombinaciju vrednosti hiper-parametara:
  - i. Trenirati model sa tim vrednostima hiper-parametara na **trening** podacima
  - ii. Proceniti performanse dobijenog modela na **test** podacima
  - iii. Odabrati vrednosti parametara koje rezultuju najmanjom greškom na **test** skupu

## 2. Evaluacija modela

- Model (sa odabranim vrednostima hiper-parametara) evaluirati na **test** skupu
- Zašto je ovo pogrešno?
  - Parametri modela su takođe deo modela koje treba optimizovati
  - Ovaj postupak je veoma sličan tome da treniramo model na trening skupu i na istom trening skupu računamo i grešku

# Optimizacija parametara modela – ispravno

- Podeliti podatke na trening, validacioni i test skup
  - Npr. 80/10/10 ili 60/20/20



# Rezultat učenja

---

- Rezultat je klasterizacija, prediktivni model ili sažetak podataka
- Model
  - Model je sažetak podataka koji očuvava reprezentaciju znanja
  - U linearnoj regresiji je znanje o problemu posle učenja predstavljeno koeficijentima
  - U k-means znanje je predstavljeno parametrima centara
- Sažetak podataka može biti i vizuelizacija

# Vizualizacija

---

- Predstavljanje podataka tako da čovekov vizualni sistem može shvati znanje
- *Najveća korist od slike je kada nas natera da vidimo ono što nismo očekivali.* – John Tukey
- Učenje se često sastoji od otkrivanja projekcije podataka koja umanjuje efekat šuma i povećava signal

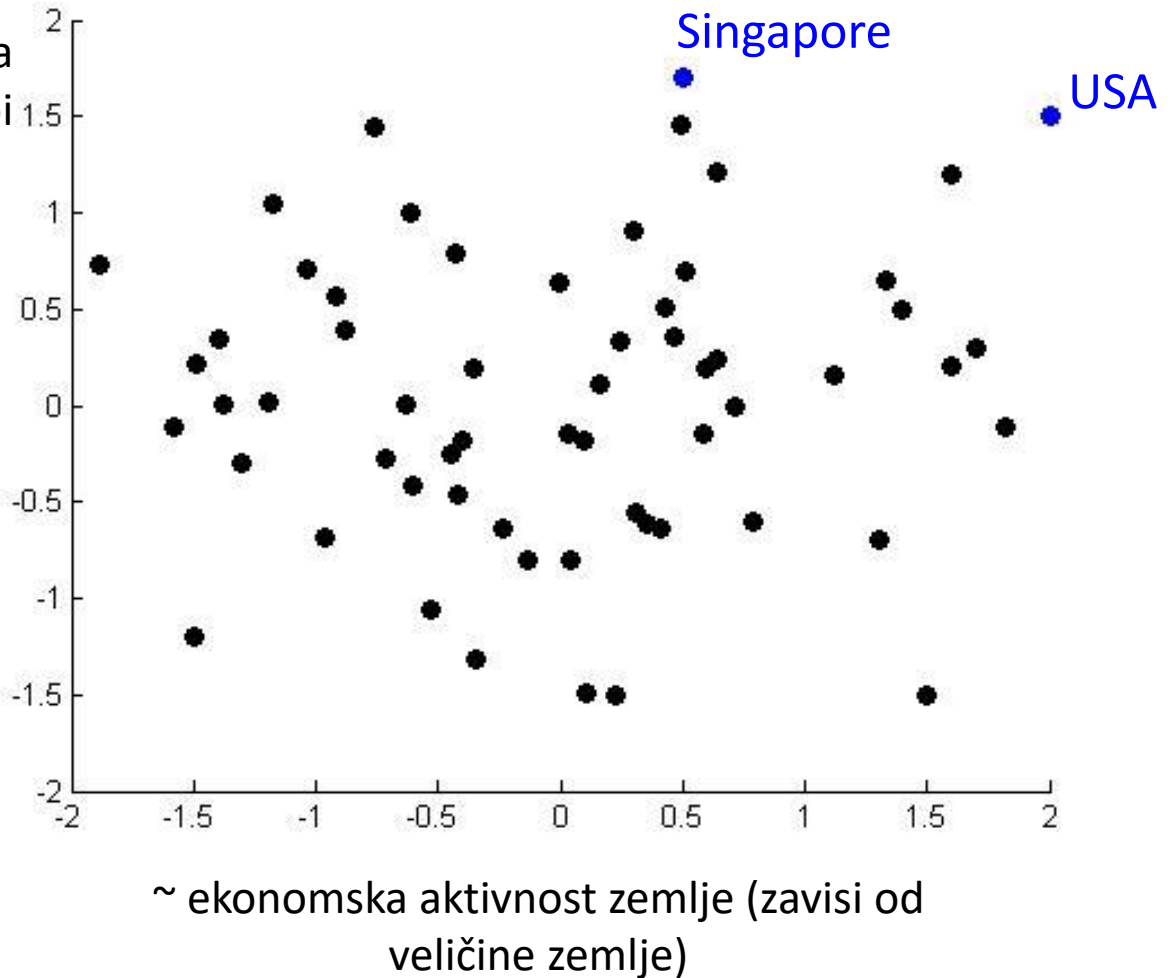
# Vizuelizacija

Country	GDP (trillions of US\$)	Per capita GDP (thousands of intl. \$)	Human Develop- ment Index	Life expectancy	Poverty Index (Gini as percentage)	Mean household income (thousands of US\$)	...
Canada	1.577	39.17	0.908	80.7	32.6	67.293	...
China	5.878	7.54	0.687	73	46.9	10.22	...
India	1.632	3.41	0.547	64.7	36.8	0.735	...
Russia	1.48	19.84	0.755	65.5	39.9	0.72	...
Singapore	0.223	56.69	0.866	80	42.5	67.1	...
USA	14.527	46.86	0.91	78.3	40.8	84.3	...
...	...	...	...	...	...	...	...

# Vizualizacija

~ GDP per  
capita/ekonomska  
aktivnost po osobi

Country		
Canada	1.6	1.2
China	1.7	0.3
India	1.6	0.2
Russia	1.4	0.5
Singapore	0.5	1.7
USA	2	1.5
...	...	...





# Učenje – šta su problemi?

---

- Visoke dimenzije
  - Prostor podataka je tipično visoko dimenzionalan zbog toga što su senzori i memorija sve jeftiniji pa ima mnogo izvora merenja
  - Prostor rešenja je često manje dimenzije od prostora podataka zato što koristimo znanje da eliminišemo nepotrebne delove informacija iz podataka
- Treba proceniti nivo neodređenosti
  - U realnom svetu svaki senzor proizvodi neki nivo šuma u podacima koje generiše
- Faktori su nekad diskretni a nekada kontinualni
  - Često je potrebno integrisati heterogene izvore podataka i proceniti njihovu relevantnost za problem

# Kako prezentovati rezultate?

---

- Analiza grešaka modela
  - Uzmite uzorak primera na kojima je model pogrešio, toliki da možete ručno da ga pregledate
  - Pokušajte da identifikujete razloge zbog kojih vaš model greši
    - Loše osvetljenje, loša rezolucija, veoma sličan prikaz dve različite klase, velika varijacija izgleda objekata iste klase, modelu je dat mali broj primera za klasu na kojoj je pogrešio
    - Klase koje model najčešće meša
  - Na osnovu ovoga definišite ograničenja svog sistema

# Primeri analize grešaka modela

## Potencijalni problemi

- Slabo osvetljenje
- Slični znaci za dva ili više znakova
- Rotacija šake
- Koža korisnika nije bela



# Primeri analize grešaka modela

---

- Isti otisak, ali različita pozicija, orijentacija i veličina



# Primeri analize grešaka modela



Улаз (узор)



Улаз (извор)



Излаз 1



Излаз 2

Prenos vizualnih svojstava je uspeo sa stila na fotografiju, ali u suprotnom smeru nije. Razlog za to jeste što umetničkim delima često nedostaju detalji u korist kreativnosti i apstrakcije



Улаз (узор)



Улаз (извор)



Излаз

Algoritam koji pronalazi korespondencije između slika ne uspeva da pronađe određeni objekat (travu) sa slike izvora na slici uzora

# Primeri analize grešaka modela



- Sistem loše odvaja slike ljudi od pozadine ukoliko nose garderobu sličnu boji pozadine