



Uvod u Mašinsko Učenje

Predavač: Aleksandar Kovačević

Slajdovi preuzeti sa CS 4641, Georgia Tech

<https://www.cc.gatech.edu/~zk15/ML2017/>

Šta je Mašinsko Učenje?

“Učenje je proces u kome sistem poboljšava svoje performanse kroz iskustvo.”

-Herbert Simon

Tom Mitchell (1998):

Mašinsko učenje je proučavanje algoritama koji:

- poboljšavaju svoje performanse P
- za neki zadatak T
- kroz iskustvo E.

Dobro definisan zadatak mašinskog učenja dat je sa $\langle P, T, E \rangle$.

Šta je Mašinsko Učenje?

Grana (pod-oblasc) veštačke inteligencije, koja se bavi konstrukcijom i proučavanjem sistema koji uče iz podataka.



Šta je Mašinsko Učenje?

Programiranje računara da optimizuju neki kriterijum performansi koristeći primere (podatke) iz prošlosti.

-- Ethem Alpaydin

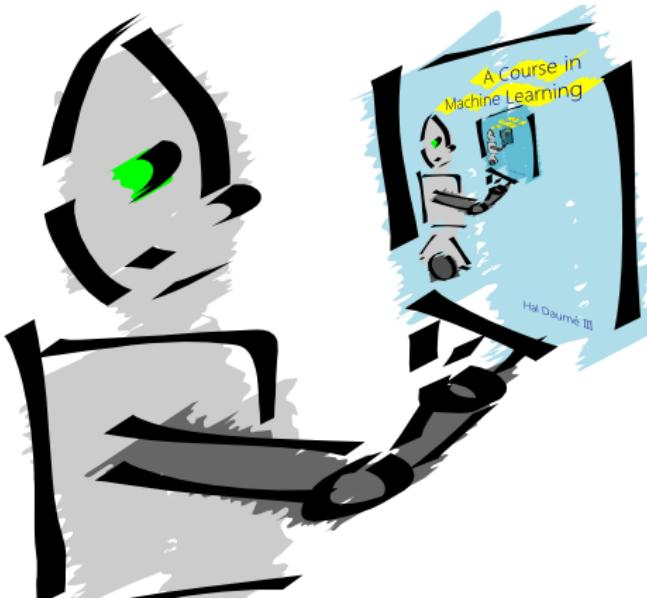
Cilj mašinskog učenja je razvoj algoritama koji mogu automatski da detektuju šablonе u podacima, i onda da iskoristi te šablonе za predviđanje budućih vrednosti.

-- Kevin P. Murphy

Šta je Mašinsko Učenje?

Mašinsko Učenje je predikcija budućnosti na osnovu prošlosti.

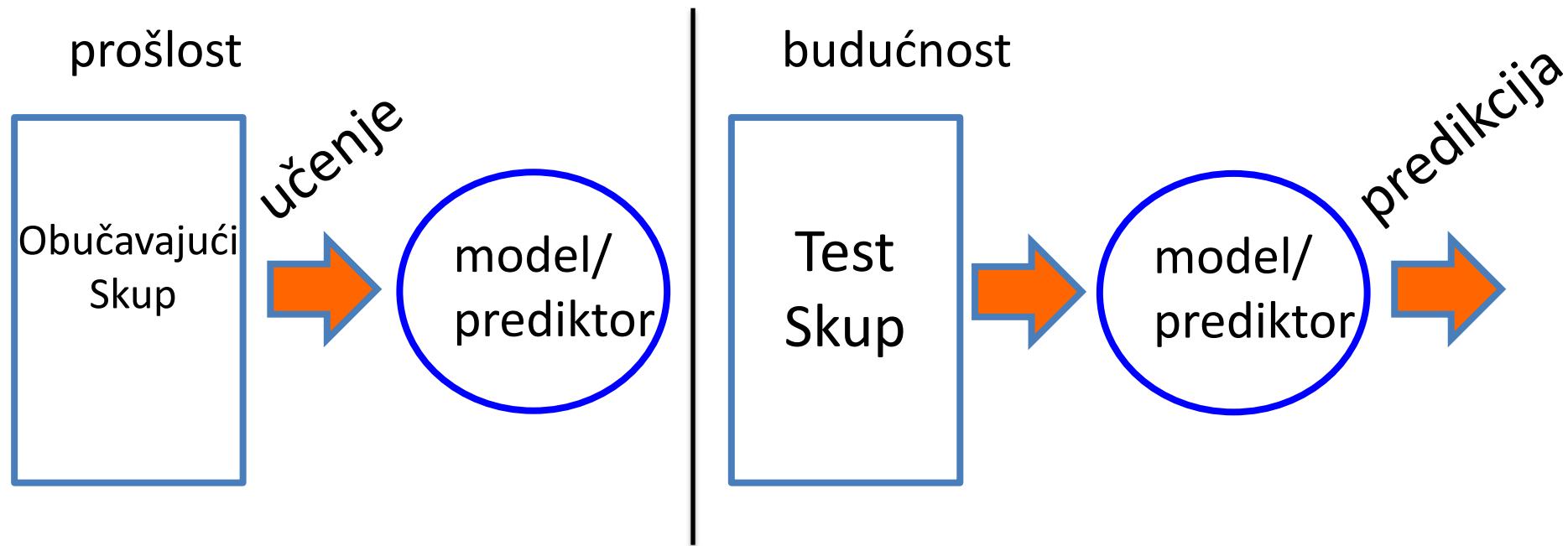
-- Hal Daume III



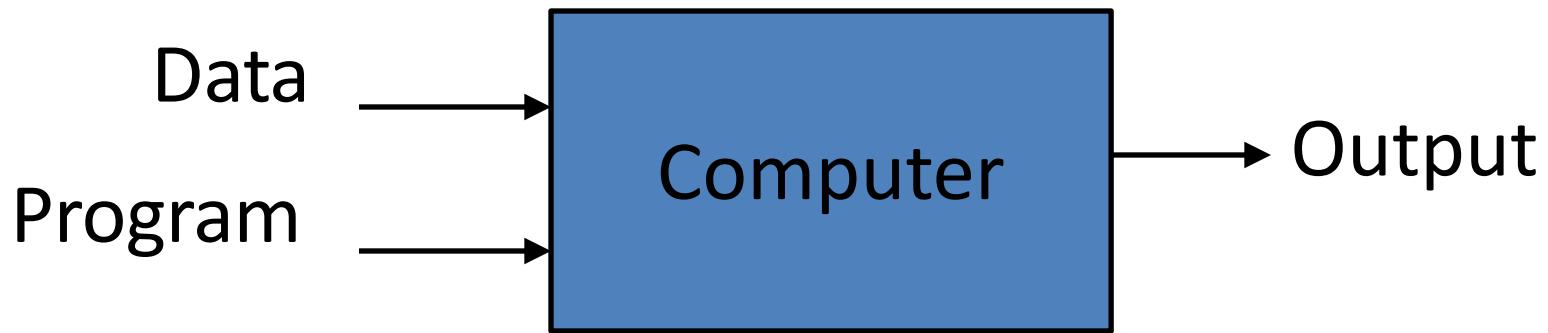
Šta je Mašinsko Učenje?

Mašinsko Učenje je predikcija budućnosti na osnovu prošlosti.

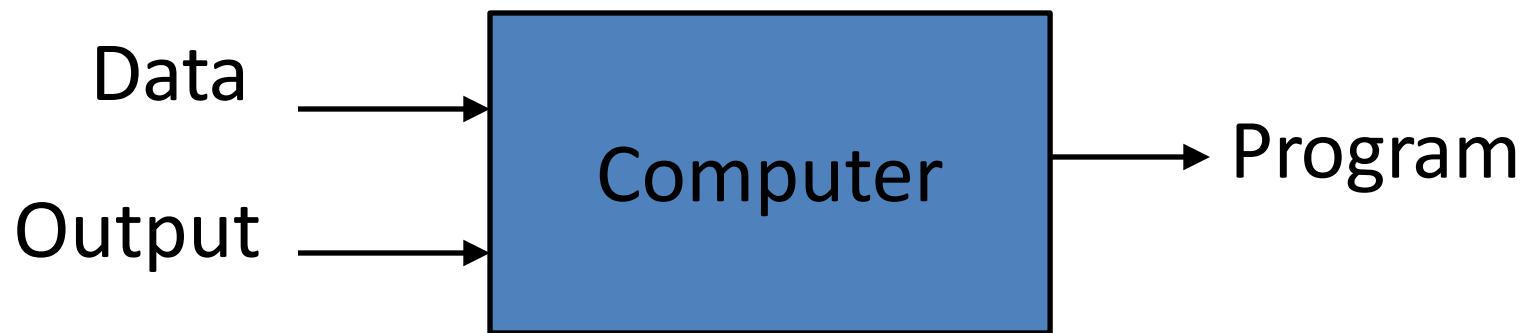
-- Hal Daume III



Tradicionalno programiranje



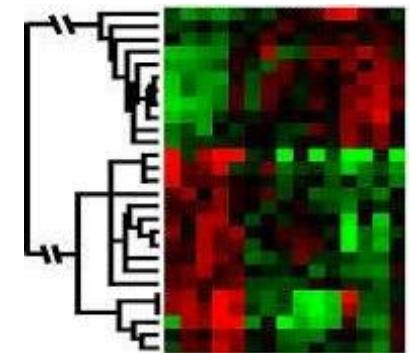
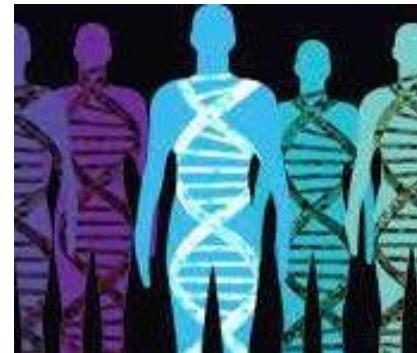
Mašinsko Učenje



Kada nam je potrebno Mašinsko Učenje?

ML se koristi kad:

- Ljudska ekspertiza ne postoji (npr. roboti koji istražuju druge planete)
- Ljudi ne mogu da objasne ekspertizu (prepoznavanje lica ili govora)
- Nivo kvaliteta, detaljnosti za koji ljudi nemaju vremena (personalizovana medicina)
- Količine podataka koje ljudi ne mogu da savladaju (bioinformatika - geni)



Mašinsko učenje nije uvek potrebno:

- Npr. ne treba nam ML da izračunamo neto od bruto plate.

Prepoznavanje pisanih cifara (ili slova) je klasičan zadatak za koji je potrebno Mašinsko Učenje

čoveku je jako teško da objasni kako zna da prepozna broj 2

0 0 0 1 1 1 1 1 2

2 2 2 2 2 2 3 3 3

3 4 4 4 4 4 5 5 5

6 6 7 7 7 7 8 8 8

8 8 8 8 9 9 9 9

Još neki zadaci za koje je prikladna i uspešna primena ML

- Prepoznavanje šablonu:
 - Prepoznavanje lica ili određenih izraza lica
 - Prepozvanje pisanih ili izgovorenih reči
 - Obrada medicinskih slika (npr. da li određeni snimak sadrži tumor ili ne)
- Generisanje šablonu:
 - Generisanje slika, teksta, govora ili muzike
- Detekcija anomalija:
 - Neuobičajene transakcije sa kreditnim karticama
 - Neuobičajne vrednosti na senzorima nuklearne elektrane
- Predikcija:
 - Predikcija vrednosti akcija ili kursa valuta

Neki od primera primene

- Web pretraživači (npr. Learn To Rank algoritam)
- Bioinformatika
- Biznis
- Elektronska trgovina
- Istraživanje svemira
- Robotika
- Ekstrakcija informacija
- Analiza društvenih mreža
- Itd.

Jedan od ranih primera: Igranje Dama

“Mašinsko učenje: Naučna disciplina koja omogućava računarima da uče bez eksplicitnog programiranja.”
-Arthur Samuel (1959)



Definisanje zadataka mašinskog učenja kroz primere

Poboljšati performanse za zadatak T, u odnosu na meru performansi P, na osnovu iskustva E

T: Igranje dama

P: Procenat pobjeda

E: Igranje sam protiv sebe

T: Prepoznavanje napisanih reči

P: Procenat tačno prepoznatih reči

E: Skup slika sa napisanim rečima koje su ljudi označili sa rečima koje sa njima nalaze

T: Autonomna vožnja po auto-putu

P: Prosečan put pređen do prve potrebe čoveka za intervencijom

E: Niz slika puta i poteza na volanu koje je uradio ljudski vozač

T: Kategorizacija e-mailova u spam i normalne

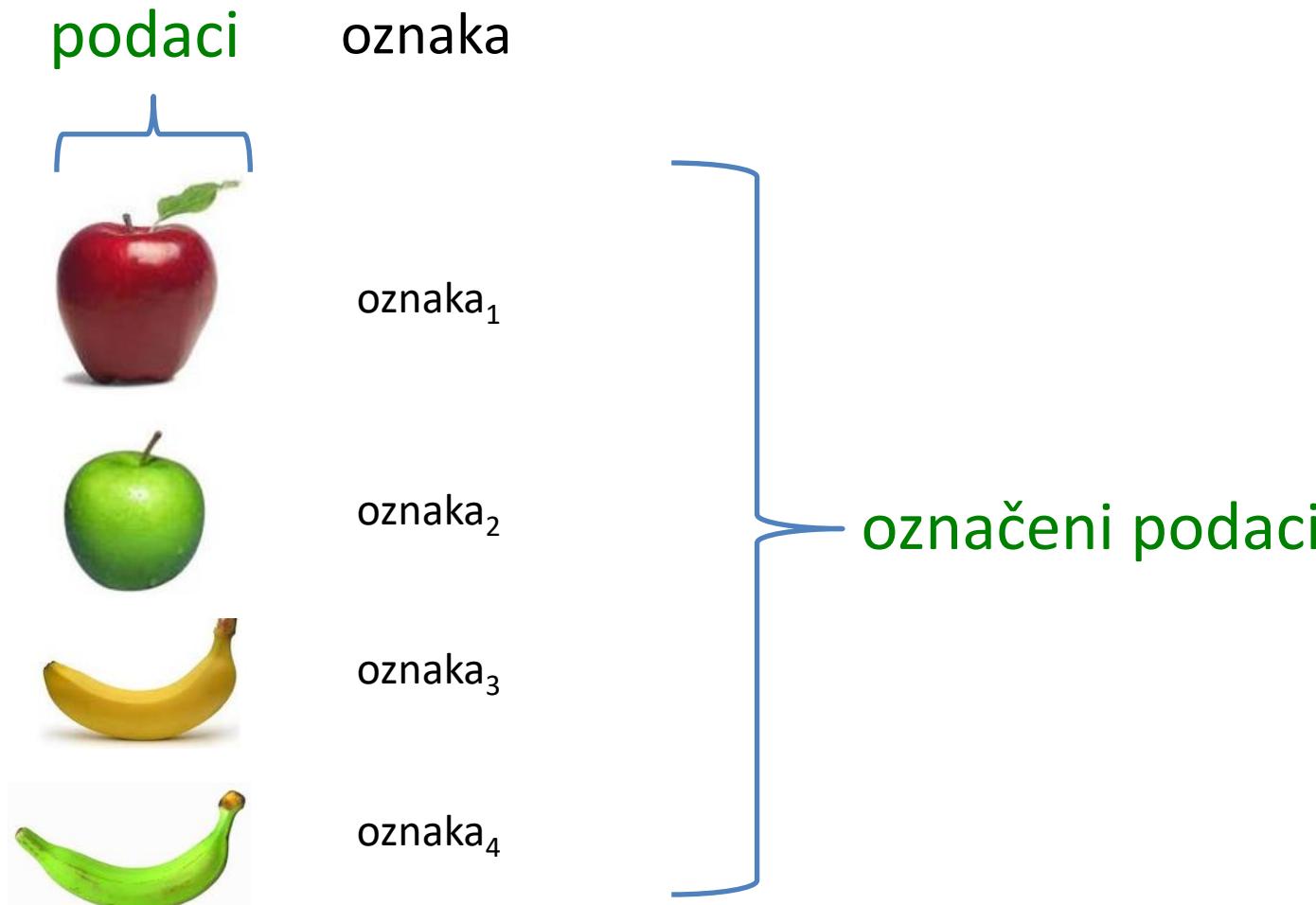
P: Procenata tačno klasifikovanih e-mailova.

E: Skup e-mailova koje su ljudi označili kao spam i normalne

Tipovi Učenja

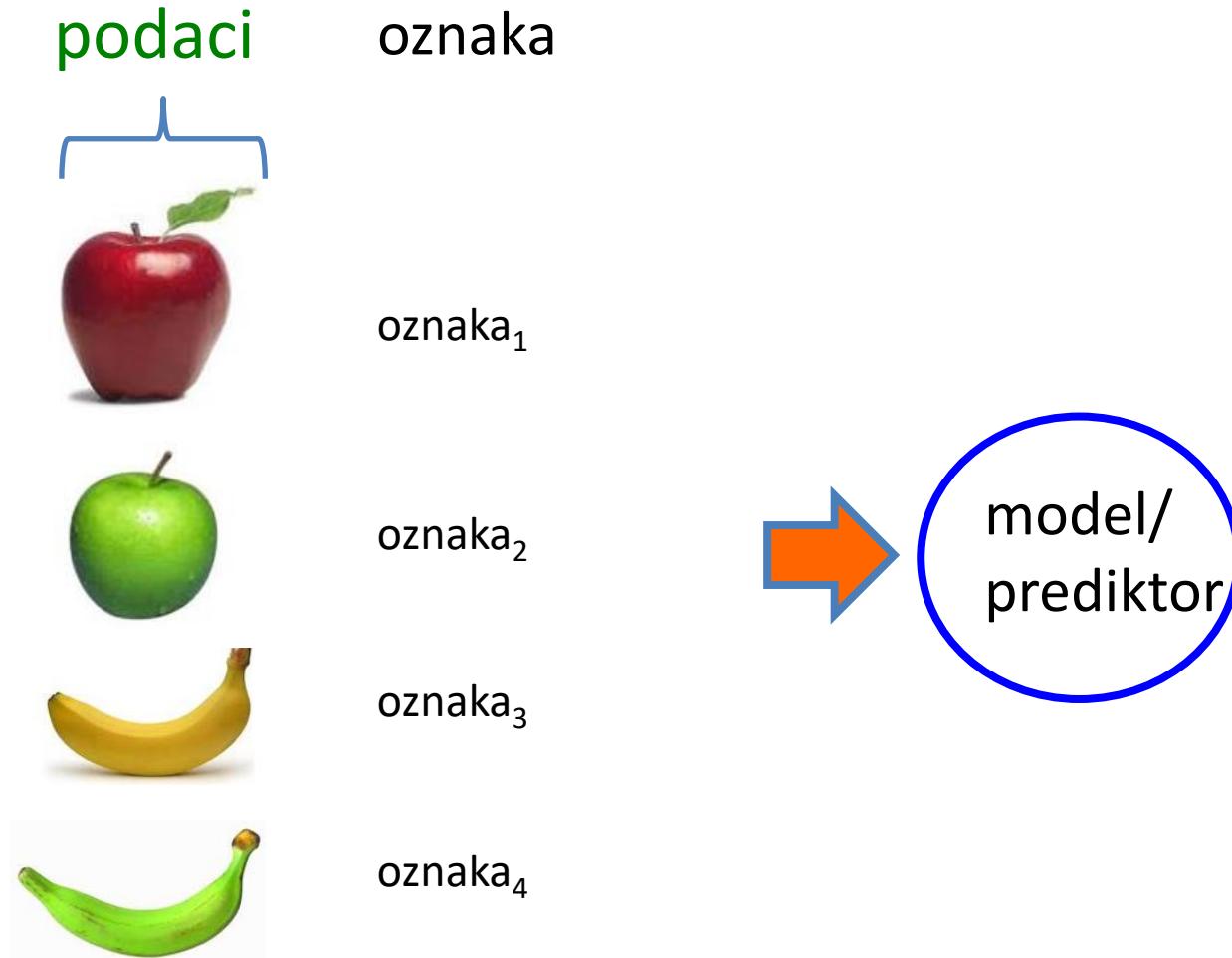
- Nadgledano (induktivno) učenje
 - Dato: skup podataka + oznake (npr. oznake klase)
- Nenadgledano učenje
 - Dato: skup podataka (bez oznaka)
- Polu-nadgledano učenje
 - Dato: skup podataka + vrlo malo označenih podatka
- Učenje Uslovljavanjem
 - Nagrade ili kazne nakon niza akcija

Nadgledano Učenje



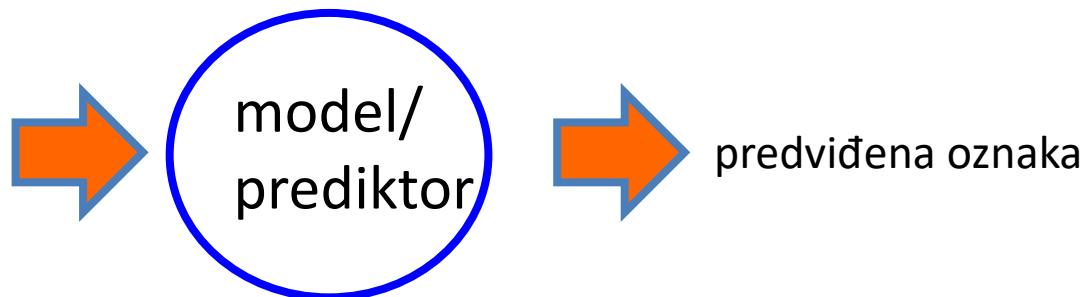
oznaka₁ – odnosi se na oznaku baš prvog primera tj. brojevi 1-4 su redni brojevi primera.
Primer 1 i 2 mogu imati istu oznaku npr. u smilu da su oba jabuke.

Nadgledano Učenje



Nadgledano Učenje

Kako označiti novi podataka – koji nema oznaku



Nadgledano učenje: klasifikacija

oznaka



jabuka



jabuka



banana

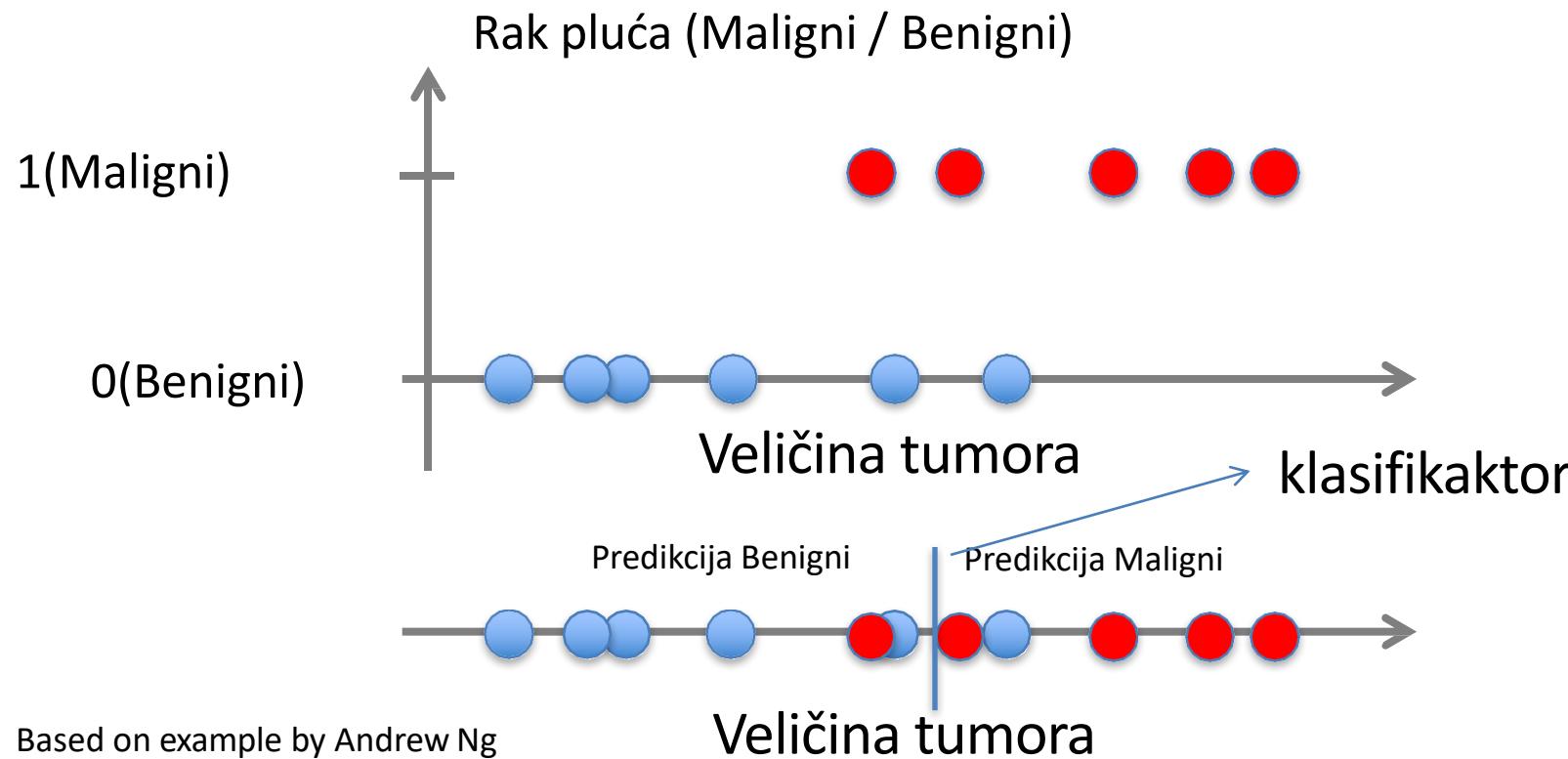


banana

Klasifikacija: konačan skup
diskretnih klasa

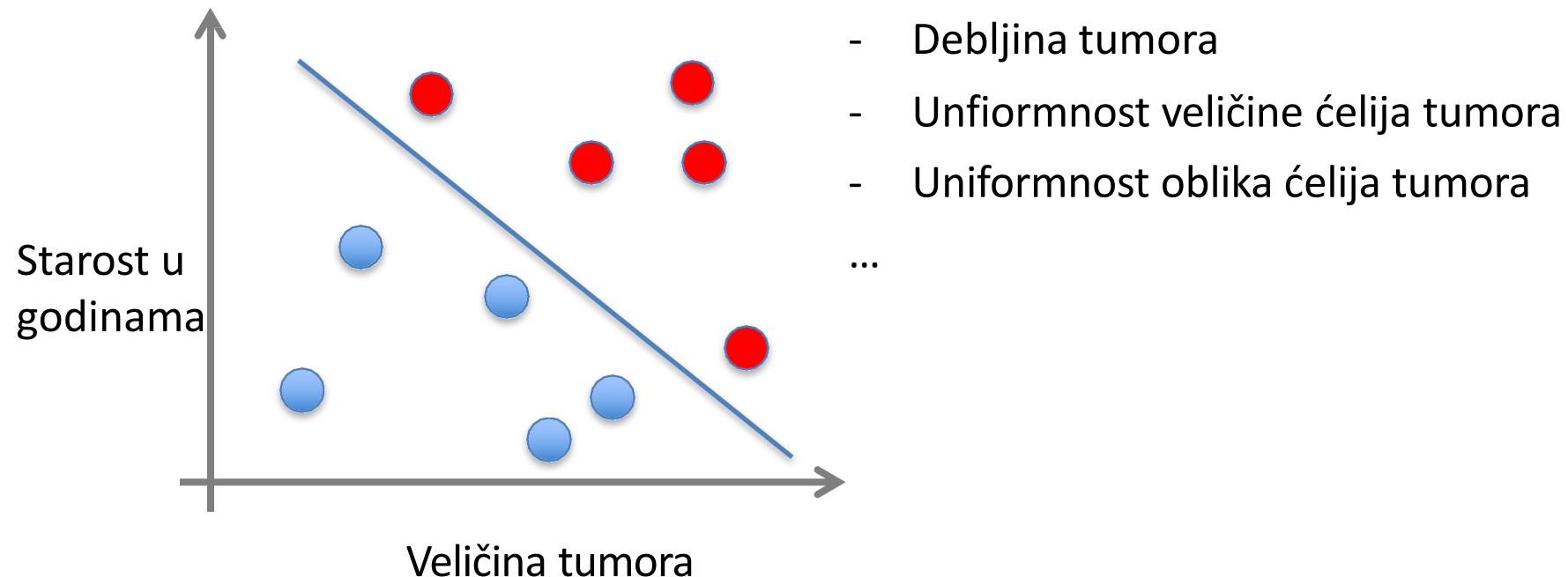
Nadgledano učenje: klasifikacija

- Dato je $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$
- Cilj je naučiti funkciju $f(x)$ koja predviđa y za dato x
 - y je diskretna vrednost



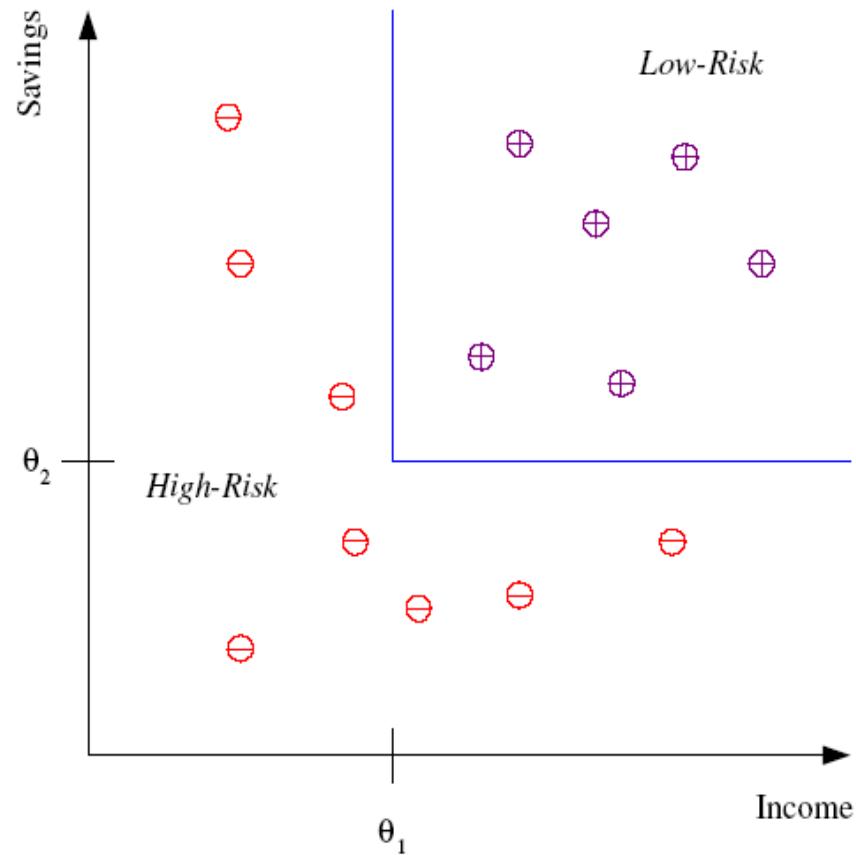
Nadgledano učenje: klasifikacija

- x može biti više-dimezionalni
 - Svaka dimenzija je jedan atribut



Klasifikacija – još primera

Klasifikovati aplikante za kredit u **nisko-rizične** i **visoko-rizične** na osnovu prihoda (*income*) i (*ušteđevine*)



Klasifikacija – primer sa obradom slika

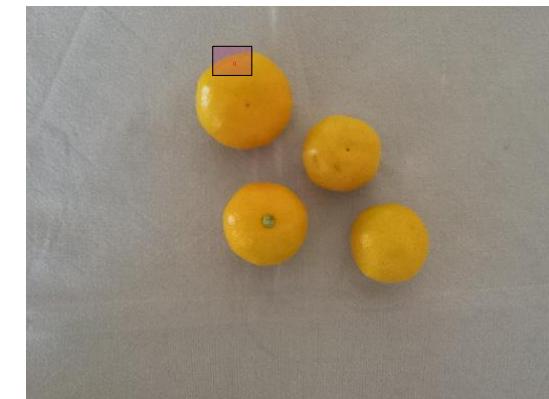
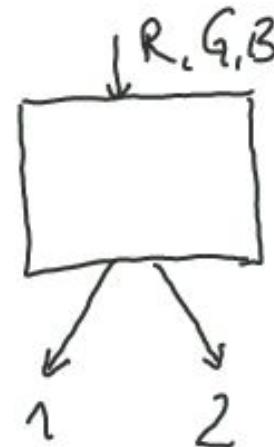
Klasifikacija

R	G	B	Klasa
153	141	125	K1
151	139	123	?
152	140	124	?
153	140	123	?
154	142	126	K1
154	141	124	?
156	143	126	?
155	142	125	?
151	138	121	?
155	143	127	?
152	139	122	?
150	138	122	?
197	142	23	?
158	145	128	?
201	146	27	K2
199	144	25	K2
149	136	119	?
156	144	128	?
157	144	127	?

Nadgledano učenje

Šta se uči?

Pravila po kojima se neoznačeni podaci klasifikuju u kategorije.



Klasifikacija – primer sa obradom slika

Klasifikacija

R	G	B	Klasa
153	141	125	K1
151	139	123	?
152	140	124	?
153	140	123	?
154	142	126	K1
154	141	124	?
156	143	126	?
155	142	125	?
151	138	121	?
155	143	127	?
152	139	122	?
150	138	122	?
197	142	23	?
158	145	128	?
201	146	27	K2
199	144	25	K2
149	136	119	?
156	144	128	?
157	144	127	?

Nadgledano učenje

Šta se uči?

Pravila po kojima se ne označeni podaci klasifikuju u kategorije.

Ako je vrednost obeležja $B < 50$ tada objekat pripada kategoriji K2 u protivnom pripada kategoriji K1

Klasifikacija – primer sa obradom slika

Klasifikacija

R	G	B	Region
153	141	125	K1
151	139	123	?
152	140	124	?
153	140	123	?
154	142	126	K1
154	141	124	?
156	143	126	?
155	142	125	?
151	138	121	?
155	143	127	?
152	139	122	?
150	138	122	?
197	142	23	?
158	145	128	?
201	146	27	K2
199	144	25	K2
149	136	119	?
156	144	128	?
157	144	127	?

Nadgledano učenje

Ako je vrednost obeležja $B < 50$ tada objekat pripada kategoriji K2 u protivnom pripada kategoriji K1

Da li je $B < 50$ jedino pravilo kojim se u ovom primeru može KVALITETNO izvršiti klasifikacija?

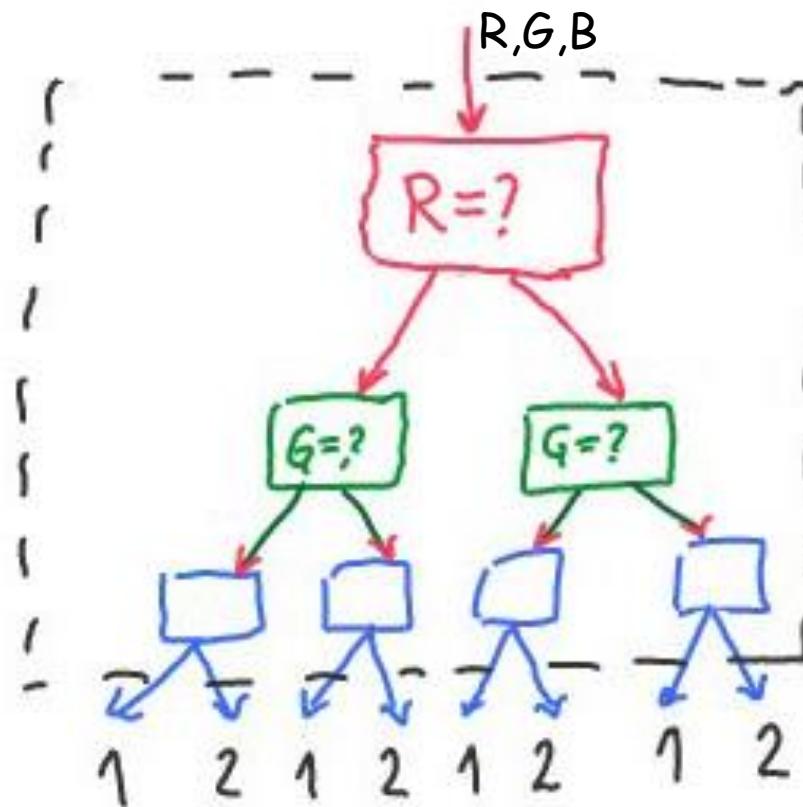
Klasifikacija – primer sa obradom slika

Klasifikacija

R	G	B	Region
153	141	125	K1
151	139	123	?
152	140	124	?
153	140	123	?
154	142	126	K1
154	141	124	?
156	143	126	?
155	142	125	?
151	138	121	?
155	143	127	?
152	139	122	?
150	138	122	?
197	142	23	?
158	145	128	?
201	146	27	K2
199	144	25	K2
149	136	119	?
156	144	128	?
157	144	127	?

Nadgledano učenje

Stabla odlučivanja



Klasifikacija – Probabilističke tehnike

Klasifikacija

R	G	B	Region
153	141	125	K1
151	139	123	?
152	140	124	?
153	140	123	?
154	142	126	K1
154	141	124	?
156	143	126	?
155	142	125	?
151	138	121	?
155	143	127	?
152	139	122	?
150	138	122	?
197	142	23	?
158	145	128	?
201	146	27	K2
199	144	25	K2
149	136	119	?
156	144	128	?
157	144	127	?

Nadgledano učenje

Probabilističke tehnike

Običan BAYES klasifikator

Markovljevi skriveni lanci

BAYES mreže

Klasifikacija – Druge tehnike

Klasifikacija

R	G	B	Region
153	141	125	K1
151	139	123	?
152	140	124	?
153	140	123	?
154	142	126	K1
154	141	124	?
156	143	126	?
155	142	125	?
151	138	121	?
155	143	127	?
152	139	122	?
150	138	122	?
197	142	23	?
158	145	128	?
201	146	27	K2
199	144	25	K2
149	136	119	?
156	144	128	?
157	144	127	?

Nadgledano učenje

Neuronske mreže

Višeslojni perceptron

SVM Suport Vector Machine

Još neki primeri klasifikacije sa ilustracijama: Spam Filter

podaci

Osman Khan to Carlos show details Jan 7 (6 days ago) Reply ▾

sounds good
+ok

Carlos Guestrin wrote:
Let's try to chat on Friday a little to coordinate and more on Sunday in person?

Carlos

Welcome to New Media Installation: Art that Learns

Carlos Guestrin to 10615-announce, Osman, Miche show details 3:15 PM (8 hours ago) Reply ▾

Hi everyone,

Welcome to New Media Installation:Art that Learns

The class will start tomorrow.
Make sure you attend the first class, even if you are on the Wait List.
The classes are held in Doherty Hall C316, and will be Tue, Thu 01:30-4:20 PM.

By now, you should be subscribed to our course mailing list: 10615-announce@cs.cmu.edu.
You can contact the instructors by emailing: 10615-instructors@cs.cmu.edu

Natural _LoseWeight SuperFood Endorsed by Oprah Winfrey, Free Trial 1 bottle, pay only \$5.95 for shipping mfw rlk Spam | X

Jaquelyn Halley to nherlein, bcc: thehorney, bcc: ang show details 9:52 PM (1 hour ago) Reply ▾

==== Natural WeightLOSS Solution ===

Vital Acai is a natural WeightLOSS product that Enables people to lose weight and cleansing their bodies faster than most other products on the market.

Here are some of the benefits of Vital Acai that You might not be aware of. These benefits have helped people who have been using Vital Acai daily to Achieve goals and reach new heights in their dieting that they never thought they could.

- * Rapid WeightLOSS
- * Increased metabolism - BurnFat & calories easily!
- * Better Mood and Attitude
- * More Self Confidence
- * Cleanse and Detoxify Your Body
- * Much More Energy
- * BetterSexLife
- * A Natural Colon Cleanse

predikcija

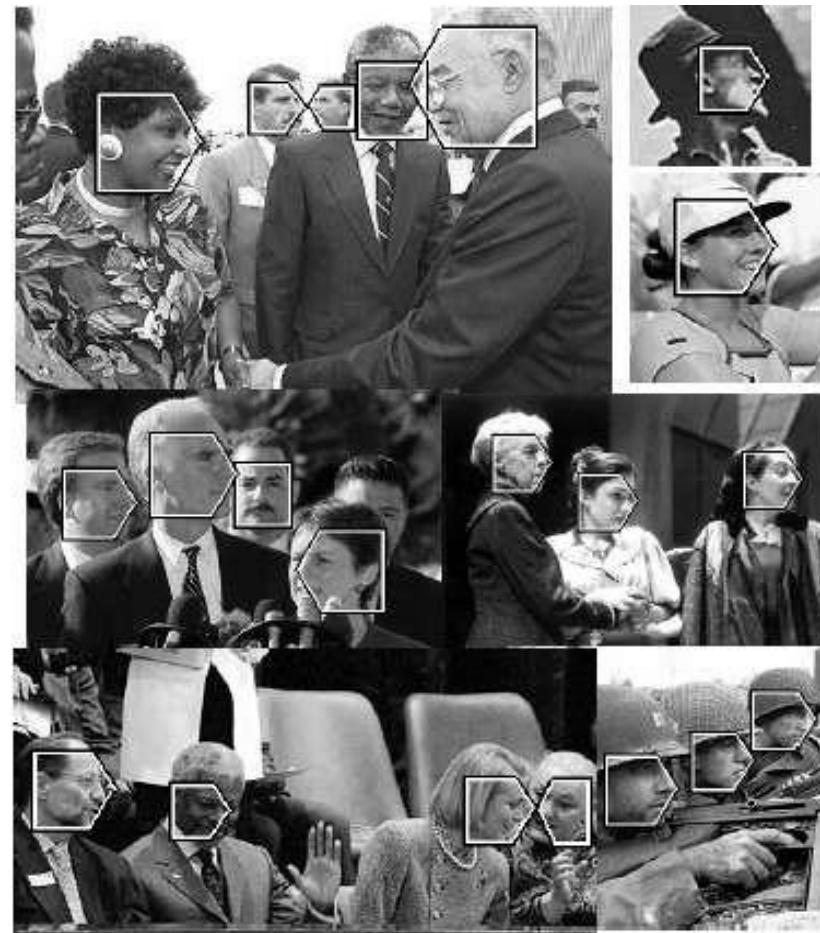


Spam
vs.
Normalan

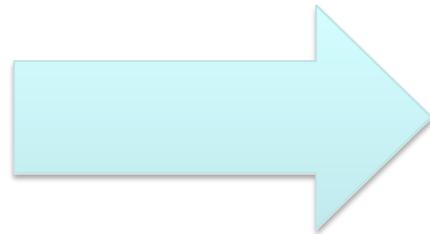
Još neki primeri klasifikacije sa ilustracijama: Prepoznavanje lica



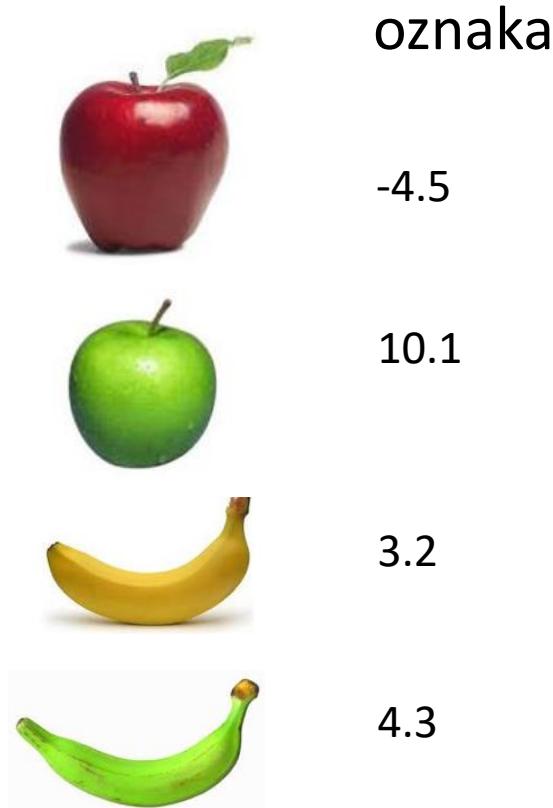
Primer iz obučavajućeg skupa –
isto lice iz različitih uglova



Još neki primeri klasifikacije sa ilustracijama:
Prognoza vremena (u diskretne klase:
oblačno, sunčano, kiša....)



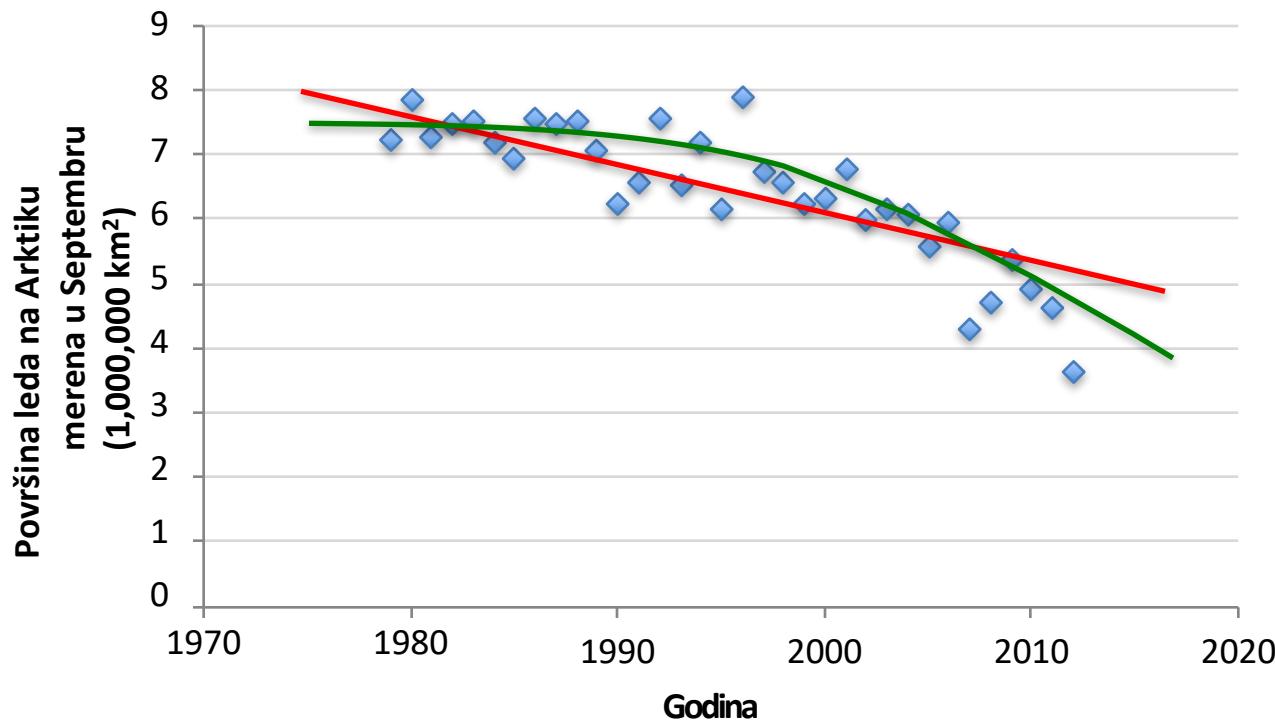
Nadgledano Učenje: Regresija



Regresija: Oznaka tj. cilja vrednost je kontinualna (obično realan broj), a ne diskretna

Nadgledano Učenje: Regresija

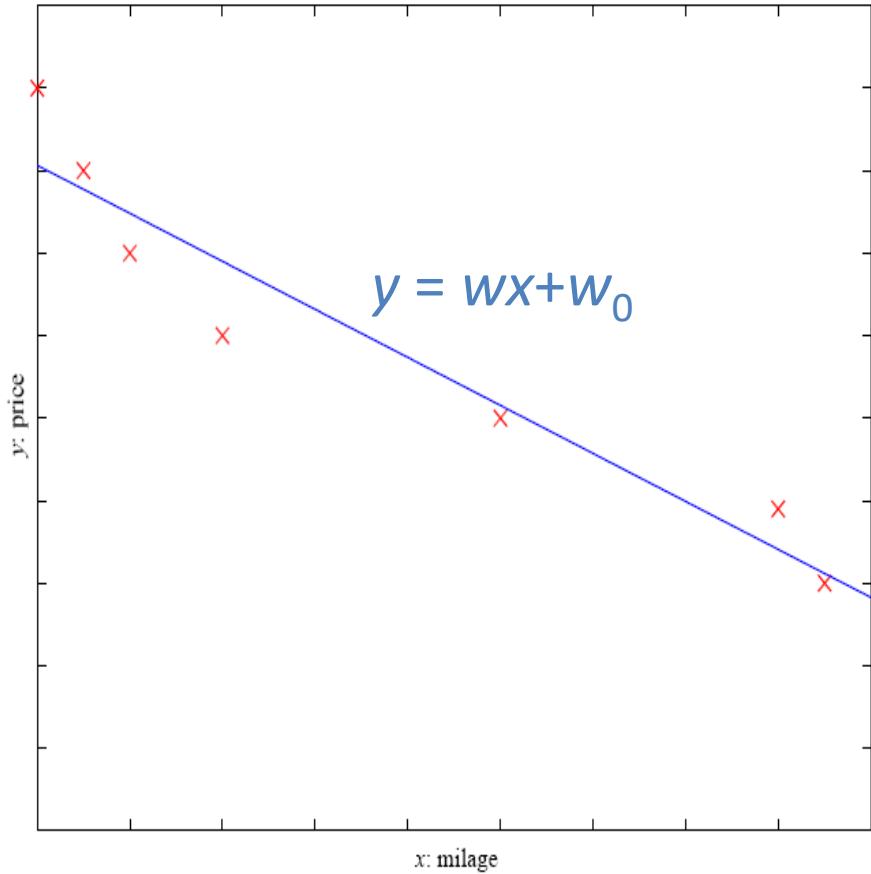
- Dato je $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$
- Cilj je naučiti funkciju $f(x)$ koja predviđa y za dato x
 - y je kontinunalna vrednost



Regresija – još primera

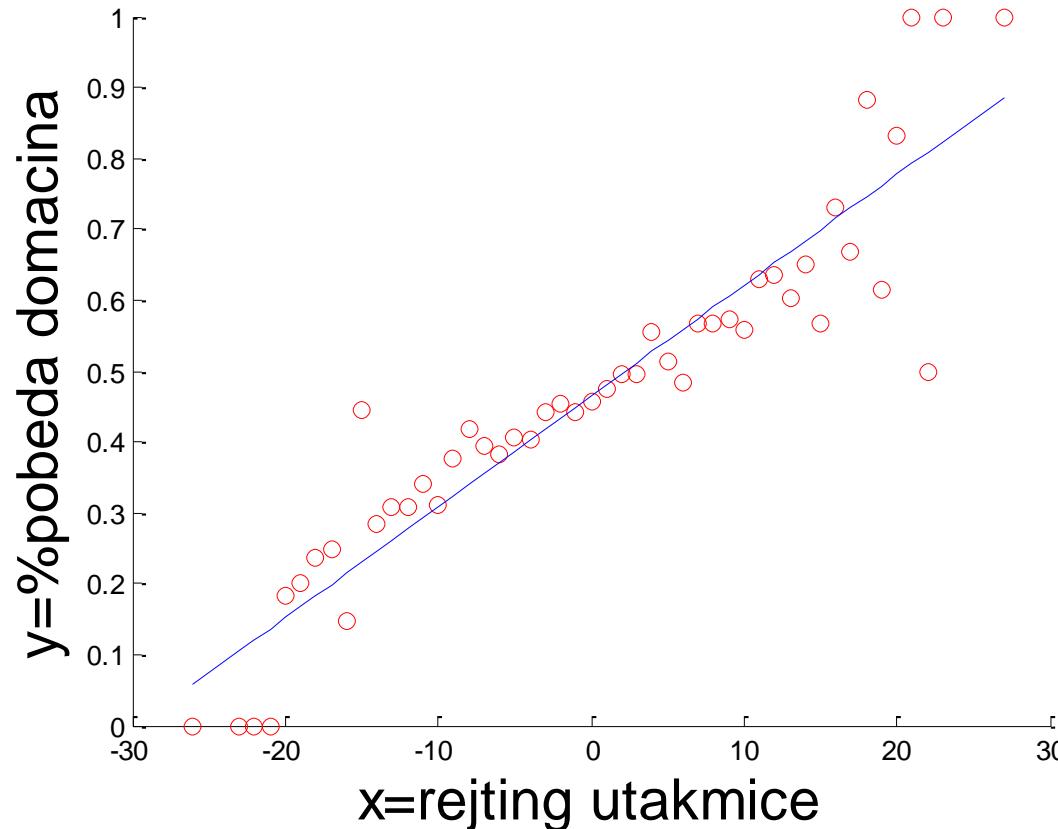
Cena polovnih automobila

x : atributi automobila
(npr. broj pređenih kilometara)
 y : cena



Regresija – još primera

Predikcija ishoda utakmica



Regresija – još primera

Ekonomija/Finansije: predikcija vrednosti akcija ili kripto-valuta...

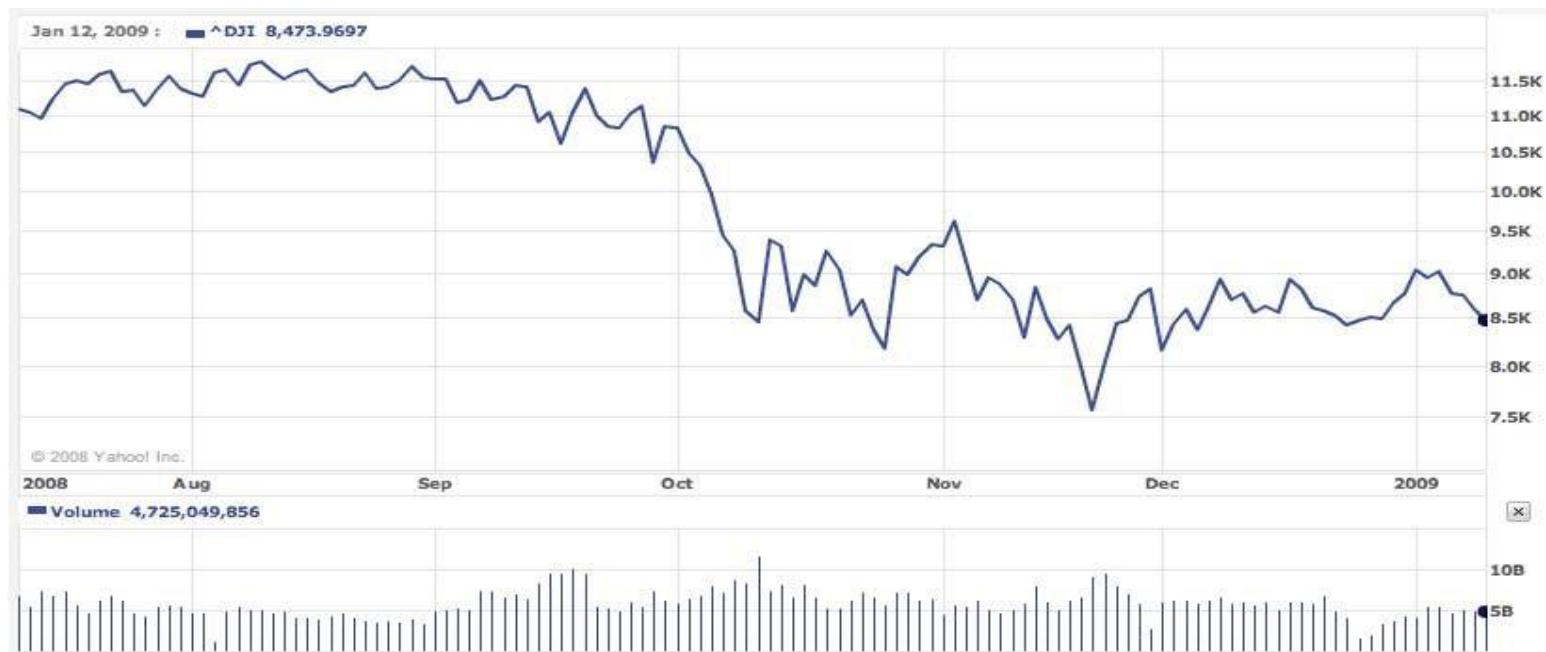
Epidemiologija – predikcija širenja neke bolesti

Navigacija vozila: ugao volana, ubrzanje, ...

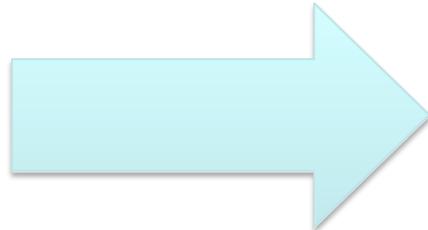
Prognoza temeprature vazduha

...

Još neki primeri regresije sa ilustracijama: Predikcija vrednosti akcija



Još neki primeri regresije sa ilustracijama:
Prognoza vremena (za razliku od klasifikacije
sad predviđamo temperaturu tj. kontinualnu
vrednost)



Nadgledano učenje: rangiranje

Oznaka



1



4



2

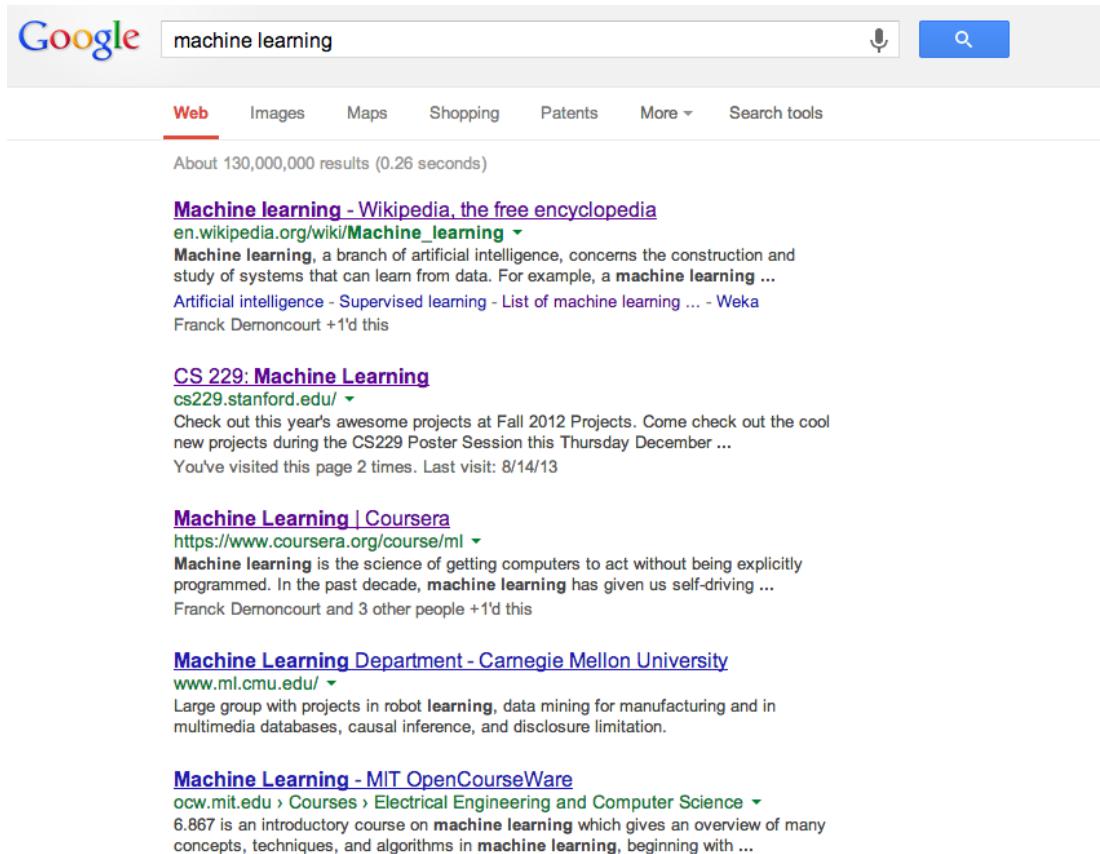


3

Rangiranje: oznaka je rang (npr.
1 je bolje od 2, a 2 od 3 itd.)

Rangiranje - primeri

Web pretraga:
Rangiranje web
strana po relevatnosti
u odnosu na upit



A screenshot of a Google search results page for the query "machine learning". The search bar at the top contains the text "machine learning". Below the search bar, the "Web" tab is selected, along with other options like Images, Maps, Shopping, Patents, More, and Search tools. A message indicates "About 130,000,000 results (0.26 seconds)". The results are listed in a vertical format, each with a blue link, the website URL in green, and a brief snippet of text. The first result is a link to Wikipedia's Machine Learning article. The second result is a link to CS 229: Machine Learning from Stanford University. The third result is a link to Coursera's Machine Learning course. The fourth result is a link to the Machine Learning Department at Carnegie Mellon University. The fifth result is a link to MIT's OpenCourseWare Machine Learning course.

Google machine learning

Web Images Maps Shopping Patents More Search tools

About 130,000,000 results (0.26 seconds)

[Machine learning - Wikipedia, the free encyclopedia](#)
en.wikipedia.org/wiki/Machine_learning ✓
Machine learning, a branch of artificial intelligence, concerns the construction and study of systems that can learn from data. For example, a machine learning ...

[Artificial intelligence - Supervised learning - List of machine learning ... - Weka](#)
Franck Demontcourt +1'd this

[CS 229: Machine Learning](#)
cs229.stanford.edu/ ✓
Check out this year's awesome projects at Fall 2012 Projects. Come check out the cool new projects during the CS229 Poster Session this Thursday December ...
You've visited this page 2 times. Last visit: 8/14/13

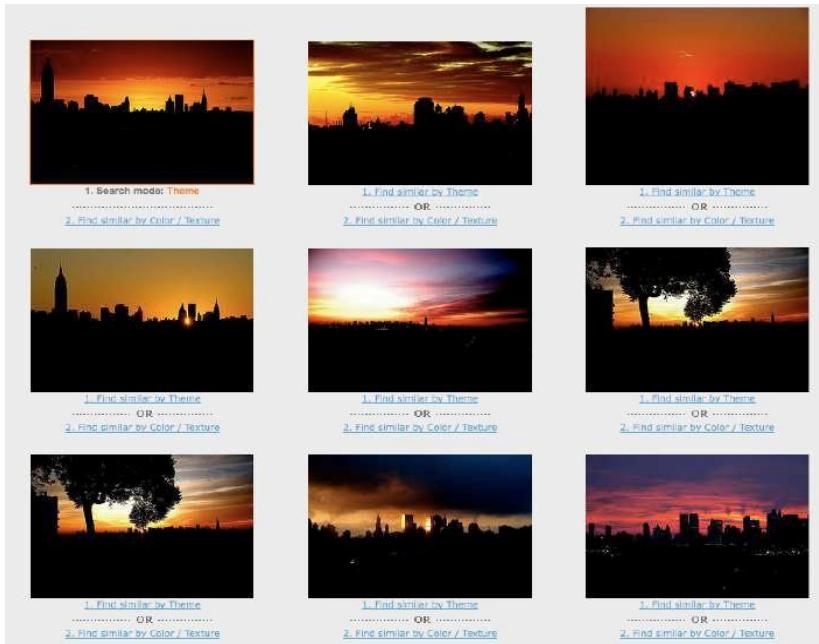
[Machine Learning | Coursera](#)
<https://www.coursera.org/course/ml> ✓
Machine learning is the science of getting computers to act without being explicitly programmed. In the past decade, machine learning has given us self-driving ...
Franck Demontcourt and 3 other people +1'd this

[Machine Learning Department - Carnegie Mellon University](#)
www.ml.cmu.edu/ ✓
Large group with projects in robot learning, data mining for manufacturing and in multimedia databases, causal inference, and disclosure limitation.

[Machine Learning - MIT OpenCourseWare](#)
[ocw.mit.edu > Courses > Electrical Engineering and Computer Science](http://ocw.mit.edu/Courses/Electrical_Engineering_and_Computer_Science/6-867-Machine_Learning) ✓
6.867 is an introductory course on machine learning which gives an overview of many concepts, techniques, and algorithms in machine learning, beginning with ...

Rangiranje - primeri

Za datu sliku pronaći slične slike



flickr

This screenshot shows a search results page from Flickr. At the top, it says "THIS PHOTO IS CURRENTLY UNAVAILABLE". Below that, it says "flickr". The search query is "candle". The results are displayed in a grid:

- Top row: A sunset image (labeled "Find similar by Theme") and a close-up of a sunflower (labeled "Find similar by Theme").
- Middle row: A red car (labeled "Find similar by Theme") and a man holding a lit candle (labeled "Find similar by Theme").
- Bottom row: A sunset image (labeled "Find similar by Theme") and a glowing lava lamp (labeled "Find similar by Theme").

Each result image has a caption below it with two options:

- 1. Find similar by Theme
- 2. Find similar by Color / Texture

<http://www.tiltomo.com/>

Rangiranje – Sistemi za preporuke

David's Amazon.com | Today's Deals | Gift Cards | Sell | Help

Shop by Department ▾ Search Books Go Hello, David Your Account ▾ Try Prime ▾ Cart ▾ Wish List ▾

Your Amazon.com Your Browsing History Recommended For You Amazon Betterizer Improve Your Recommendations Your Profile Learn More

Your Amazon.com > Recommended For You > Books > Subjects > Science & Math > History & Philosophy

Just For Today

Browse Recommended

Recommendations

History & Philosophy

History of Science

Philosophy of Biology

Philosophy of Medicine

These recommendations are based on items you own and more.

view: All | New Releases | Coming Soon

1. **Causality: Models, Reasoning and Inference**
by Judea Pearl (September 14, 2009)
Average Customer Review: ★★★★★ (10)
In Stock
List Price: \$60.00
Price: \$32.49
63 used & new from \$28.00

Add to Cart Add to Wish List

I own it Not interested Rate this item

Recommended because you purchased Probabilistic Graphical Models and more (Fix this)

2. **The Lady Tasting Tea: How Statistics Revolutionized Science in the Twentieth Century**
by David Salsburg (May 1, 2002)
Average Customer Review: ★★★★★ (76)
In Stock
List Price: \$18.99
Price: \$13.88
83 used & new from \$9.00

Add to Cart Add to Wish List

I own it Not interested Rate this item

Recommended because you added The Theory That Would Not Die to your Wish List (Fix this)

3. **The Eighth Day of Creation: Makers of the Revolution in Biology, 25th Anniversary Edition**
by Horace Freeland Judson (November 1, 1996)
Average Customer Review: ★★★★★ (10)
In stock on September 4, 2013
List Price: \$56.00
Price: \$36.09
59 used & new from \$26.95

Add to Cart Add to Wish List

I own it Not interested Rate this item

Recommended because you purchased Molecular Biology of the Cell (Fix this)

4. **The Machinery of Life**
by David S. Goodsell (April 28, 2009)
Average Customer Review: ★★★★★ (43)
In Stock
List Price: \$26.00
Price: \$17.49
92 used & new from \$12.00

Add to Cart Add to Wish List

Daily Lighting Deals
Back-to-School Savings
Shop now

Rangiranje – Sistemi za preporuke

Takmičenje sa nagradom od milion dolara

Leaderboard

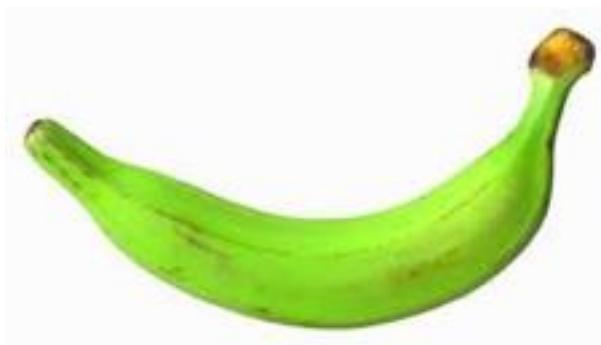
Display top 20 leaders.

Rank	Team Name	Best Score	% Improvement	Last Submit Time
1	The Ensemble	0.8553	10.10	2009-07-26 18:38:22
2	BellKor's Pragmatic Chaos	0.8554	10.09	2009-07-26 18:18:28
Grand Prize - RMSE <= 0.8563				
3	Grand Prize Team	0.8571	9.91	2009-07-24 13:07:49
4	Opera Solutions and Vandelay United	0.8573	9.89	2009-07-25 20:05:52
5	Vandelay Industries !	0.8579	9.83	2009-07-26 02:49:53
6	PragmaticTheory	0.8582	9.80	2009-07-12 15:09:53
7	BellKor in BigChaos	0.8590	9.71	2009-07-26 12:57:25
8	Dace	0.8603	9.58	2009-07-24 17:18:43
9	Opera Solutions	0.8611	9.49	2009-07-26 18:02:08
10	BellKor	0.8612	9.48	2009-07-26 17:19:11
11	BigChaos	0.8613	9.47	2009-06-23 23:06:52
12	Feeds2	0.8613	9.47	2009-07-24 20:06:46
Progress Prize 2008 - RMSE = 0.8616 - Winning Team: BellKor in BigChaos				
13	xiangliang	0.8633	9.26	2009-07-21 02:04:40
14	Gravity	0.8634	9.25	2009-07-26 15:58:34
15	Ces	0.8642	9.17	2009-07-25 17:42:38
16	Invisible Ideas	0.8644	9.14	2009-07-20 03:26:12
17	Just a guy in a garage	0.8650	9.08	2009-07-22 14:10:42
18	Craig Carmichael	0.8656	9.02	2009-07-25 16:00:54
19	J.Dennis Su	0.8658	9.00	2009-03-11 09:41:54
20	acmehill	0.8659	8.99	2009-04-16 06:29:35
Progress Prize 2007 - RMSE = 0.8712 - Winning Team: KorBell				
Cinematch score on quiz subset - RMSE = 0.9514				



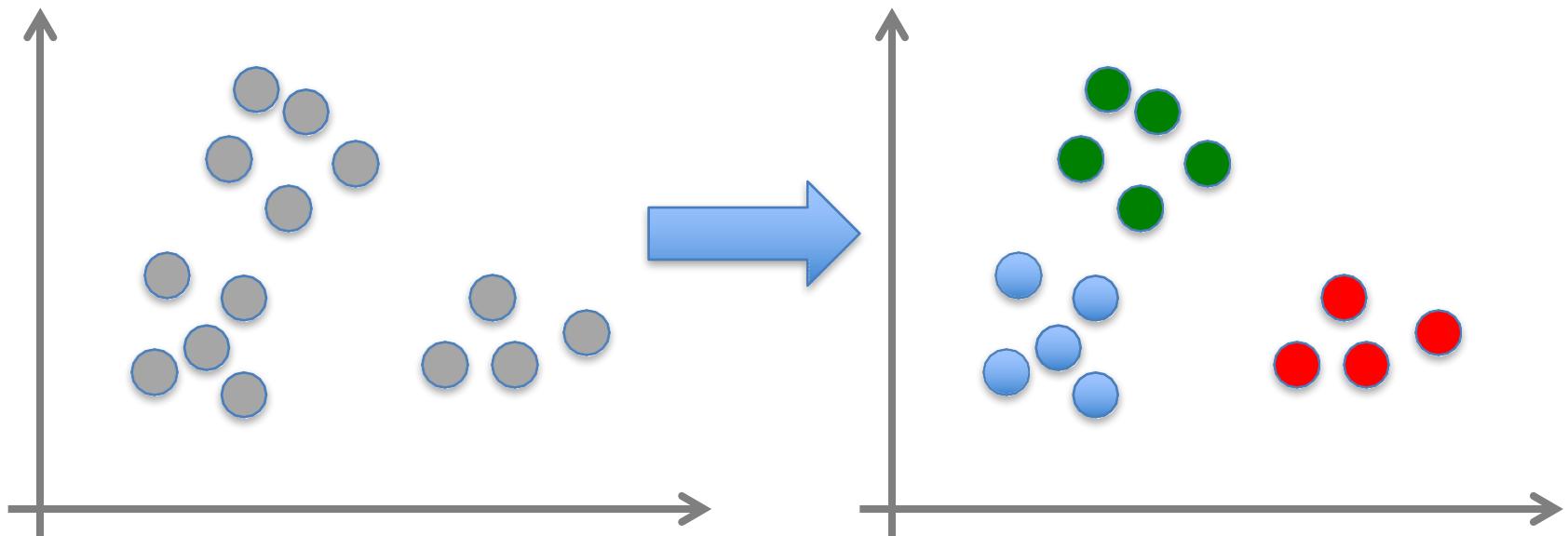
Nenadgledno Učenje

Nenadgledno učenje: dati su nam podaci ali bez oznaka



Nenadgledno Učenje

- Dato je x_1, x_2, \dots, x_n (bez oznaka)
- Izlaz je neka struktura (šablon) koji važi za x -ove
 - Npr. automatsko grupisanje (klasterovanje)



Nenadgledno Učenje - Konkretnije

Na osnovu čega se uči?

Nadgledano (označeni podaci)

Nenadgledano - bez učitelja (ne označeni podaci)

Sa uslovljavanjem

Nenadgledno Učenje - Konkretnije

Podeliti tačke u dva regiona

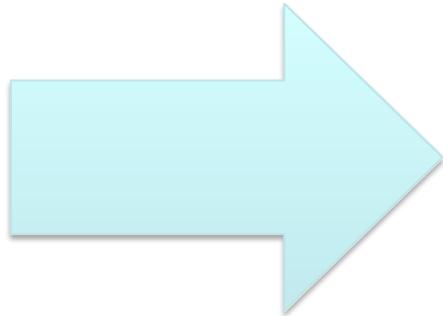
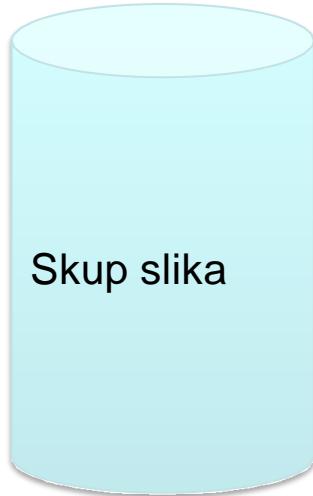


255, 127, 0

192, 192, 192

R	G	B	Region
153	141	125	
151	139	123	
152	140	124	
153	140	123	
154	142	126	
154	141	124	
156	143	126	
155	142	125	
151	138	121	
155	143	127	
152	139	122	
150	138	122	
197	142	23	
158	145	128	
201	146	27	
199	144	25	
149	136	119	
156	144	128	
157	144	127	

Još neki primeri nенадгледног уčenja sa ilustracijama: Klasterovanje slika



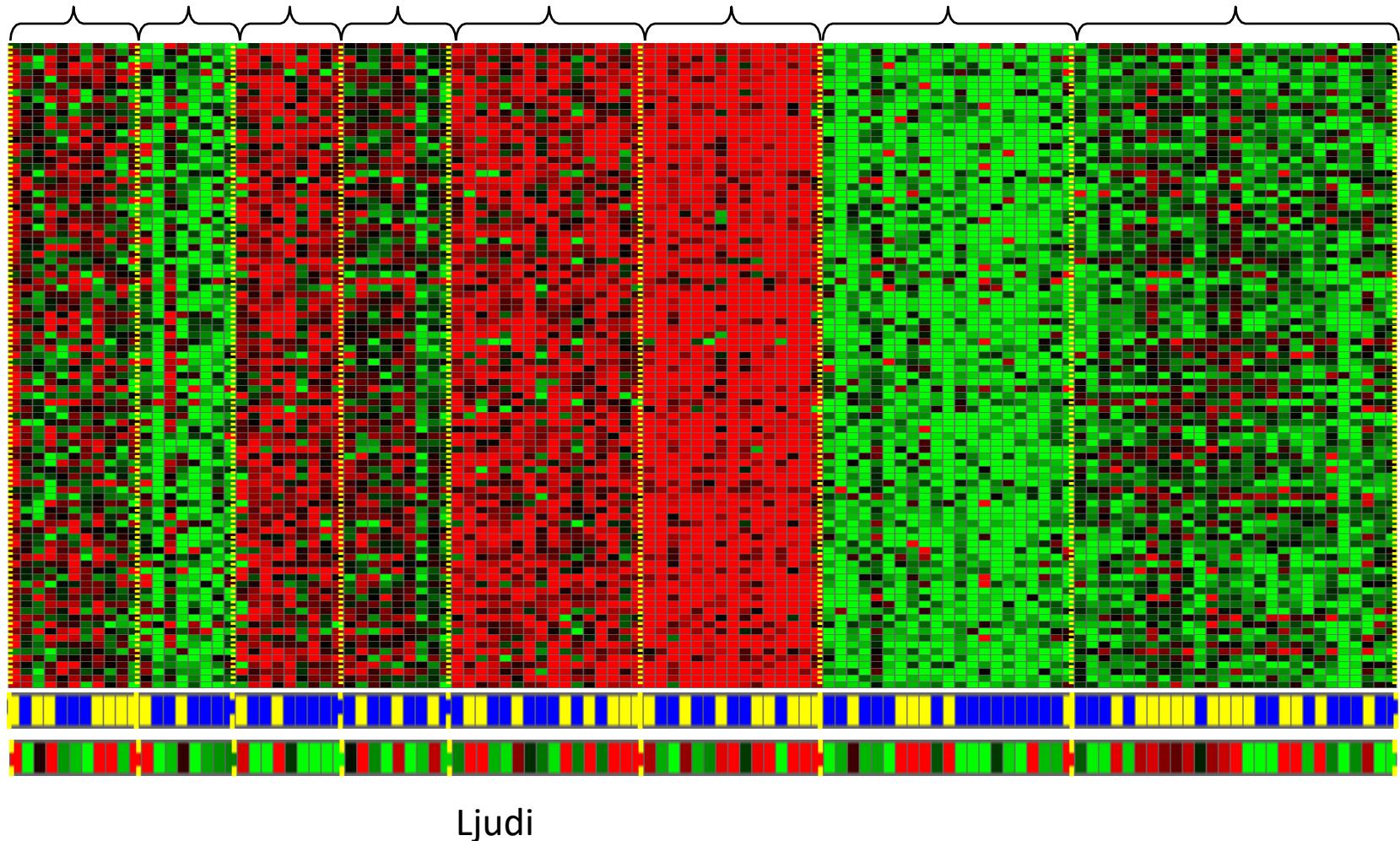
[Goldberger et al.]

Još neki primeri nenadglednog učenja sa ilustracijama: Klasterovanje rezultata Web pretrage

Cluster Human contains 8 documents.

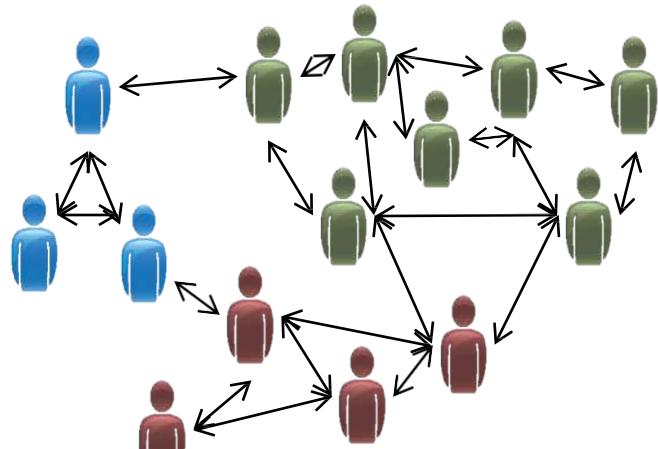
- Race (classification of human beings) - Wikipedia, the free ...**
The term **race** or racial group usually refers to the concept of dividing **humans** into populations or groups on the basis of various sets of characteristics. The most widely used **human racial** categories are based on visible traits (especially skin color, cranial or facial features and hair texture), and self-identification. Conceptions of **race**, as well as specific ways of grouping **races**, vary by culture and over time, and are often controversial for scientific as well as social and political reasons. History · Modern debates · Political and ...
[en.wikipedia.org/wiki/Race_\(classification_of_human_beings\)](http://en.wikipedia.org/wiki/Race_(classification_of_human_beings)) - [cache] - Live, Ask
- Race - Wikipedia, the free encyclopedia**
General. **Racing** competitions The **Race** (yachting **race**), or La course du millénaire, a no-rules round-the-world sailing event; **Race** (biology), classification of flora and fauna; **Race** (classification of human beings) **Race** and ethnicity in the United States Census, official definitions of "race" used by the US Census Bureau; **Race** and genetics, notion of racial classifications based on genetics. Historical definitions of **race**; **Race** (bearing), the inner and outer rings of a rolling-element bearing. **RACE** in molecular biology "Rapid ... General · Surnames · Television · Music · Literature · Video games
en.wikipedia.org/wiki/Race - [cache] - Live, Ask
- Publications | Human Rights Watch**
The use of torture, unlawful rendition, secret prisons, unfair trials, ... Risks to Migrants, Refugees, and Asylum Seekers in Egypt and Israel ... In the run-up to the Beijing Olympics in August 2008, ...
www.hrw.org/backgrounder/usa/race - [cache] - Ask
- Amazon.com: Race: The Reality Of Human Differences: Vincent Sarich ...**
Amazon.com: **Race**: The Reality Of Human Differences: Vincent Sarich, Frank Miele: Books ... From Publishers Weekly Sarich, a Berkeley emeritus anthropologist, and Miele, an editor ...
www.amazon.com/Race-Reality-Differences-Vincent-Sarich/dp/0813340861 - [cache] - Live
- AAPA Statement on Biological Aspects of Race**
AAPA Statement on Biological Aspects of **Race** ... Published in the American Journal of Physical Anthropology, vol. 101, pp 569-570, 1996 ... PREAMBLE As scientists who study **human** evolution and variation, ...
www.physanth.org/positions/race.html - [cache] - Ask
- race: Definition from Answers.com**
race n. A local geographic or global **human** population distinguished as a more or less distinct group by genetically transmitted physical
www.answers.com/topic/race-1 - [cache] - Live
- Dopefish.com**
Site for newbies as well as experienced Dopefish followers, chronicling the birth of the Dopefish, its numerous appearances in several computer games, and its eventual take-over of the **human race**. Maintained by Mr. Dopefish himself, Joe Siegler of Apogee Software.
www.dopefish.com - [cache] - Open Directory

Još neki primeri nenadglednog učenja sa ilustracijama: Klasterovanje ljudi po genetskoj sličnosti

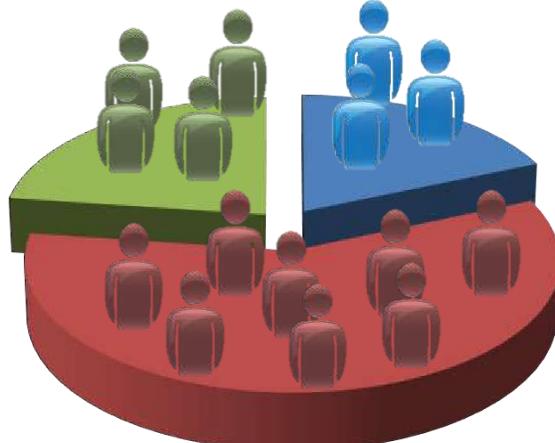


[Source: Daphne Koller]

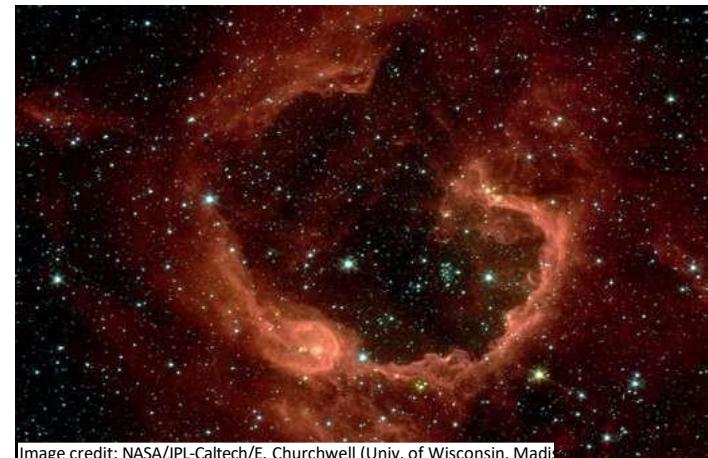
Još neki primeri nенадгледног уčenja sa ilustracijama



Otkrivanje zajednica u društvenim mrežama



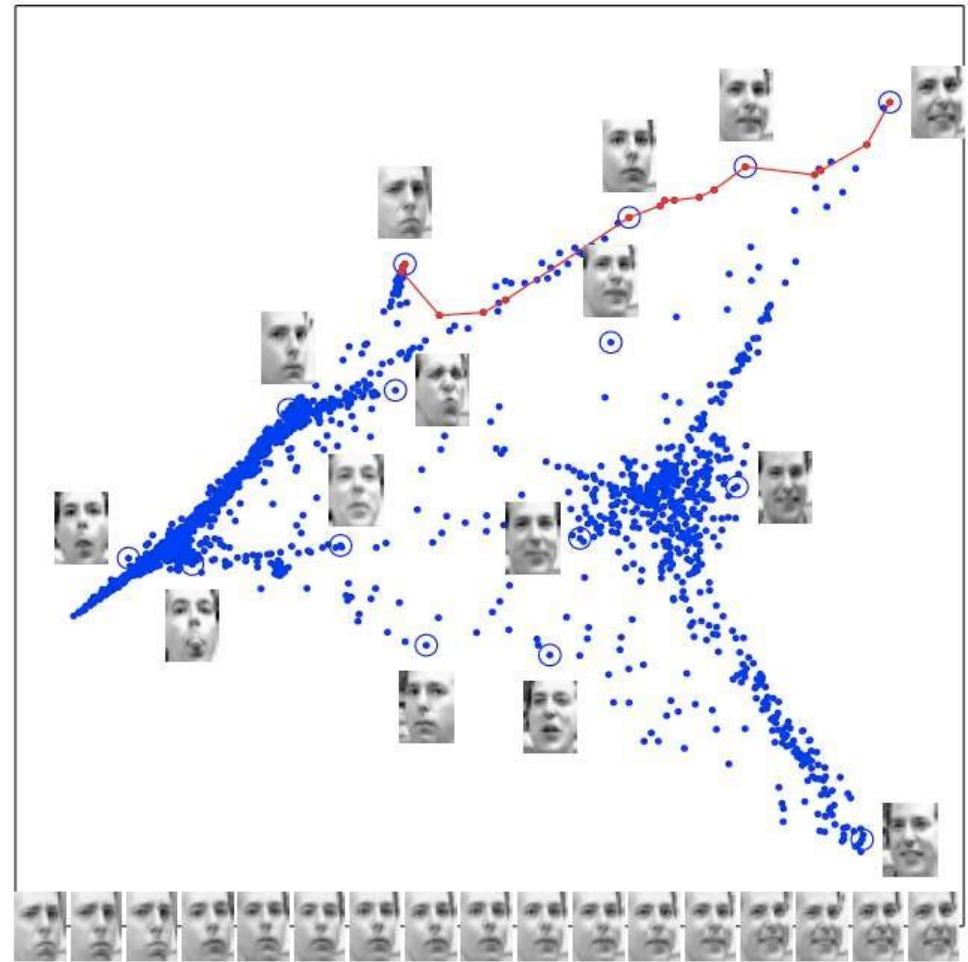
Otviranje grupa mušterija



Grupisanje nebskih tela na slikama

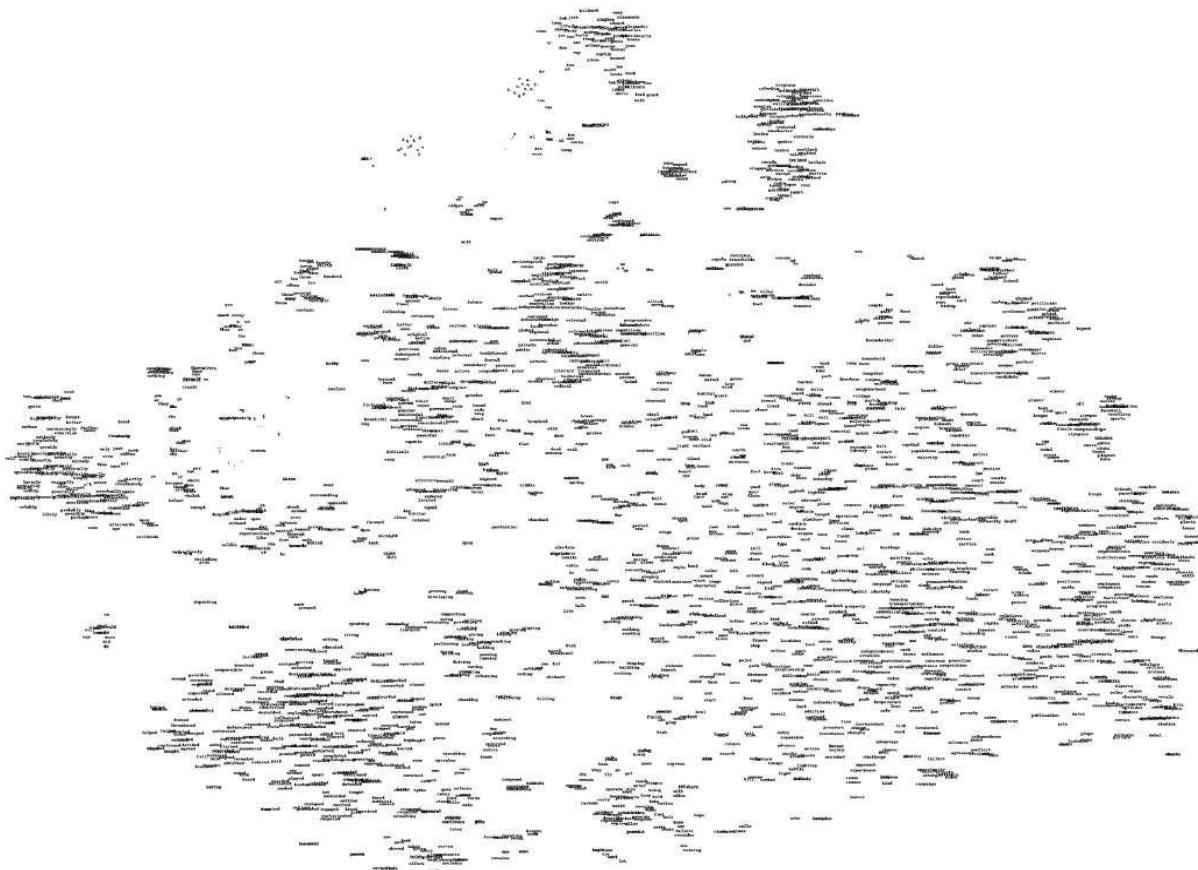
Nenadgledano učenje – redukcija dimenzionalnosti – Primer:slike

- Slike imaju hiljade ili milione piksela.
- Da li ih možemo nekako predstaviti u 2d tako da su slične slike blizu jedna druge?



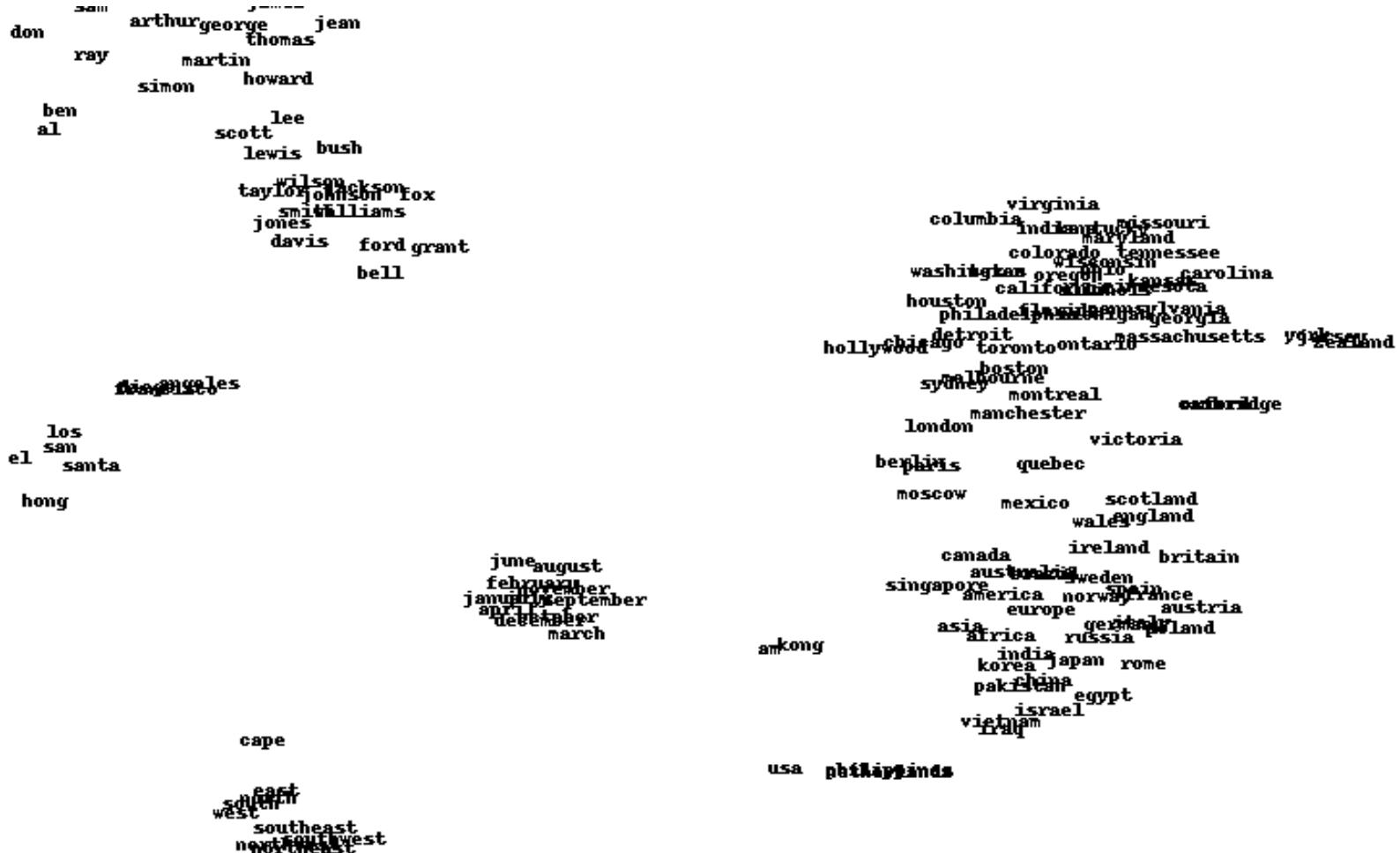
[Saul & Roweis '03]

Nenadgledano učenje – redukcija dimenzionalnosti – Primer:reči



[Joseph Turian]

Nenadgledano učenje – redukcija dimenzionalnosti – Primer:reči (zoom)



[Joseph Turian]

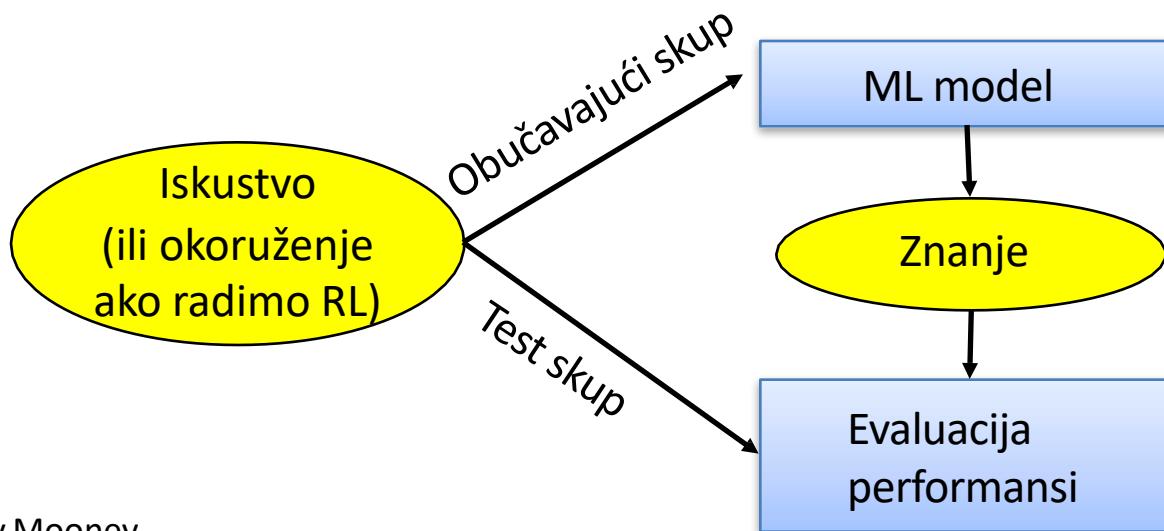
Učenje uslovljavanjem

- Učenje uslovljavanjem je vrsta mašinskog učenja
- Detaljno smo ga razmatrali na prethodnim predavanjima
- Iz tog razloga neće sad biti razmatrano

Preciznije formulisanje
rešavanja problema
pomoću mašinskog učenja

Kreiranje sistema za učenje

- Biramo iskustvo iz kojeg se uči tj. obučavajući skup
- Biramo šta želimo da naučimo
 - tj. **ciljnu funkciju (ne mora biti funkcija u matematičkom smislu)**
- Biramo način na koji reprezentujemo ciljnu funkciju
- Biramo algoritam mašinskog učenja koji bi trebalo da nauči ciljnu funkciju od obučavajućeg skupa



Distribucija obučavajućih i test podataka

- Prilikom razvoja ML algoritama generalno se pretpostavlja da su obučavajući i test podaci nezavisno izvučeni iz iste distribucije
 - Tipčna oznaka za to u ML literaturi je “i.i.d” - “independent and identically distributed”

ML u suštini

- Desetine hiljada algoritama
 - Stotine novih svake godine
- Svaki algoritam mašinskog učenja ima tri komponente:
 - Reprezentacija
 - Optimizacija
 - Evaluacija

Različiti načini reprezentacije ciljne funkcije

- Numeričke funkcije
 - Linerana regresija
 - Neuronske mreže
 - Mašine potpornih vektora (Support vector machines, SVM)
 - ...
- Simboličke funkcije
 - Stabla odlučivanja
 - Pravila u predikatskim logikama prvog reda
 -
- Funkcije zasnovane na instancama (podacima)
 - K-najbližih komšija
- Probabilistički modeli
 - Naivni Bajes
 - Bajesove mreže
 - Skriveni modeli Markova
 -

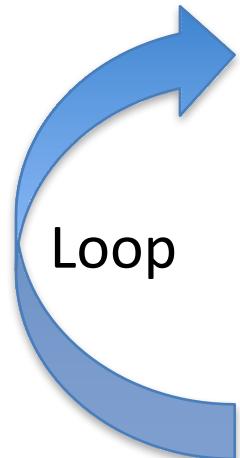
Različiti algoritmi za optimizaciju i pretragu

- Gradijentni spust
 - Perceptron
 - Backpropagation
- Dinamičko programiranje
 - Učenje Skrivenih Modela Markova
- „Podeli pa Vladaj“ - Divide and Conquer
 - Učenje stabala odlučivanja
 - Učenje pravila
- Evolutivni Algoritmi
 - Genetski algoritmi
 - ...

Evaluacija

- Tačnost (*Accuracy*)
- Preciznost i Odziv (*Precision and Recall*)
- Kvadrat greške (*Squared error*)
- Verovatnost (*Verodostojnost*) - *Likelihood*
- Posteriorna verovatnoća (*Posterior probability*)
- Trošak / Korisnot (*Cost / Utility*)
- Entropija
- KLdivergencija
- ...

Mašinsko Učenje u Praksi

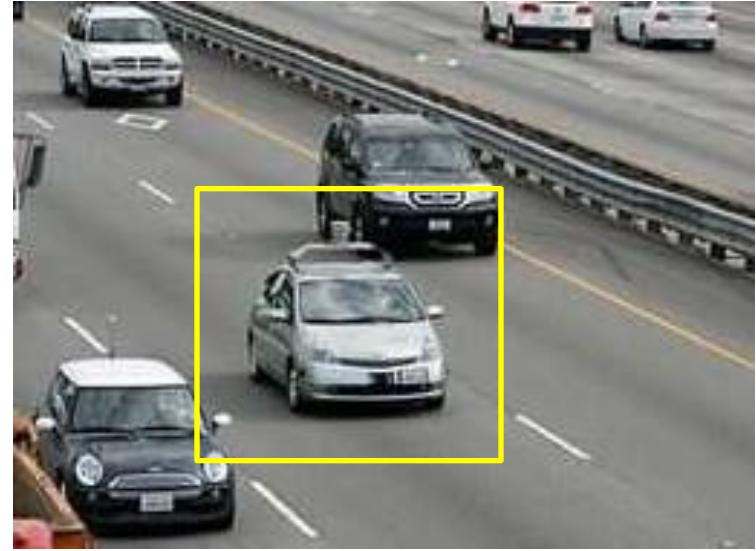
- 
- Razumeti problem i domen problema, shvatiti šta je dostupno od podataka, shvatiti šta su ciljevi
 - Integracija podataka, selekcija, čišćenje, pred-procesiranje...
 - Učenje (obučavanje) modela
 - Interpretacija rezultata
 - Konsolidacija i upotreba (deployment) dobijenog znanja

Šta smo do sad shvatili o ML?

- Mašinsko učenje se može posmatrati kao upotreba iskustva (podataka) za aproksimaciju neke ciljne funkcije.
- Proces aproksimacije funkcije može se posmatrati kao pretraga u prostoru različitih reprezentacija funkcija (npr. pravih linija) za onom koja se najbolje uklapa u podatke.
- Različiti algoritmi mašinskih učenja koriste različite reprezentacije funkcija i koriste (ili ne) različite tehnike pretrage u tom prostoru.

Vrhunske (State-of-the-Art) Primene Mašinskog Učenja

Autonomni Automobili



- Američka država Nevada je 2011 godine dozvolila da upotrebnu autonomih vozila na putevima.
- Još četiri države (Nevada, Florida, Kalifornija i Mičigen su legalizovale autonomne automobile)

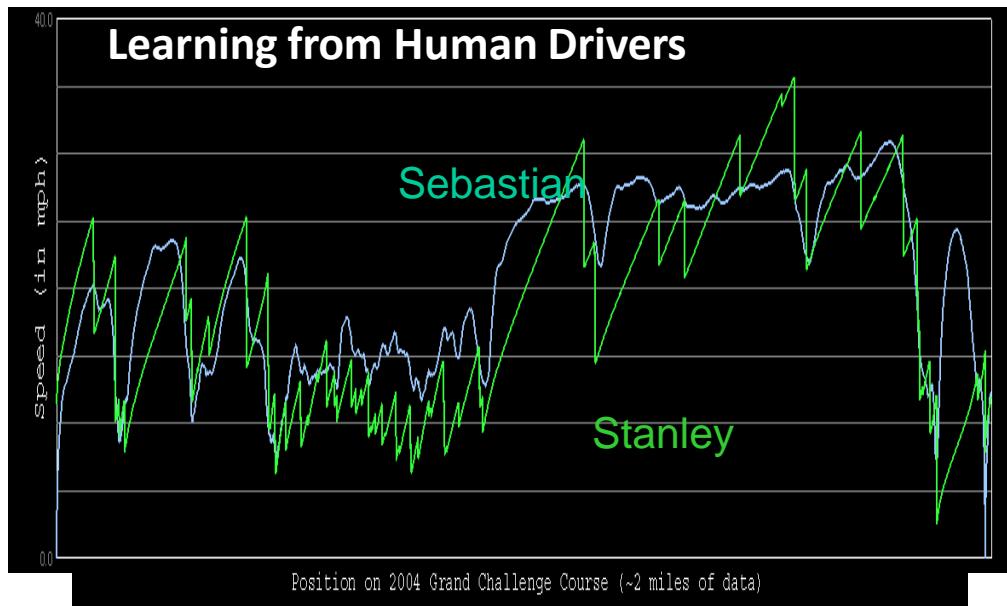
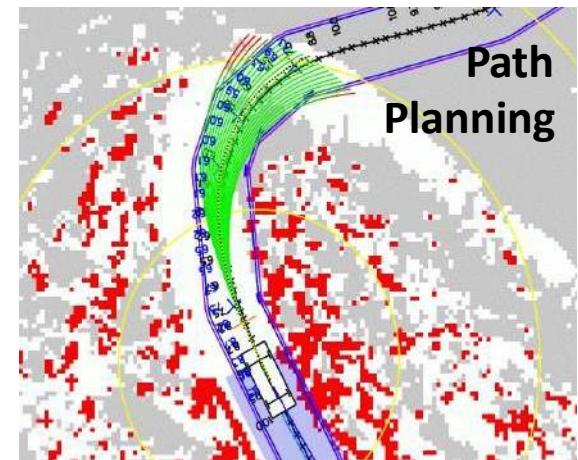
Georgia Tech's Autonomous Car
(Sting Racing Team)



Senzori na autonomnom automobilu



Tehnologije vezane za autonomne automobile



Deep Learning u medijima

BUSINESS NEWS

Is Google Cornering the Market on Deep Learning?

A cutting-edge corner of science is being wooed by Silicon Valley, to the dismay of some academics.

By Antonio Regalado on January 20, 2014



How much are a dozen deep-learning researchers worth? Apparently, more than \$400 million.

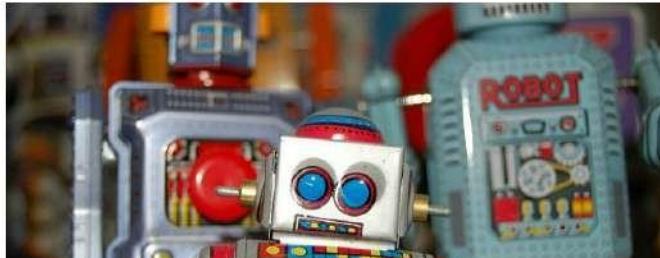


This week, Google reportedly paid that much to acquire DeepMind Technologies, a startup based in London that has developed a computer program that can learn to play video games.

WIRED GEAR SCIENCE ENTERTAINMENT BUSINESS SECURITY DESIGN
INNOVATION INSIGHTS | [community content](#) | ▾ featured

Deep Learning's Role in the Age of Robots

BY JULIAN GREEN, JETPAC 05.02.14 2:56 PM



Bloomberg Businessweek Technology

Acquisitions

The Race to Buy the Human Brains Behind Deep Learning Machines

By Ashlee Vance | January 27, 2014

intelligence projects. "DeepMind is bona fide in terms of its research capabilities and depth," says Peter Lee, who heads Microsoft Research.

According to Lee, Microsoft, Facebook (FB), and Google find themselves in a battle for deep learning talent. Microsoft has gone from four full-time deep learning experts to 70 in the past three years. "We would have more if the talent was there to



DEEP LEARNING

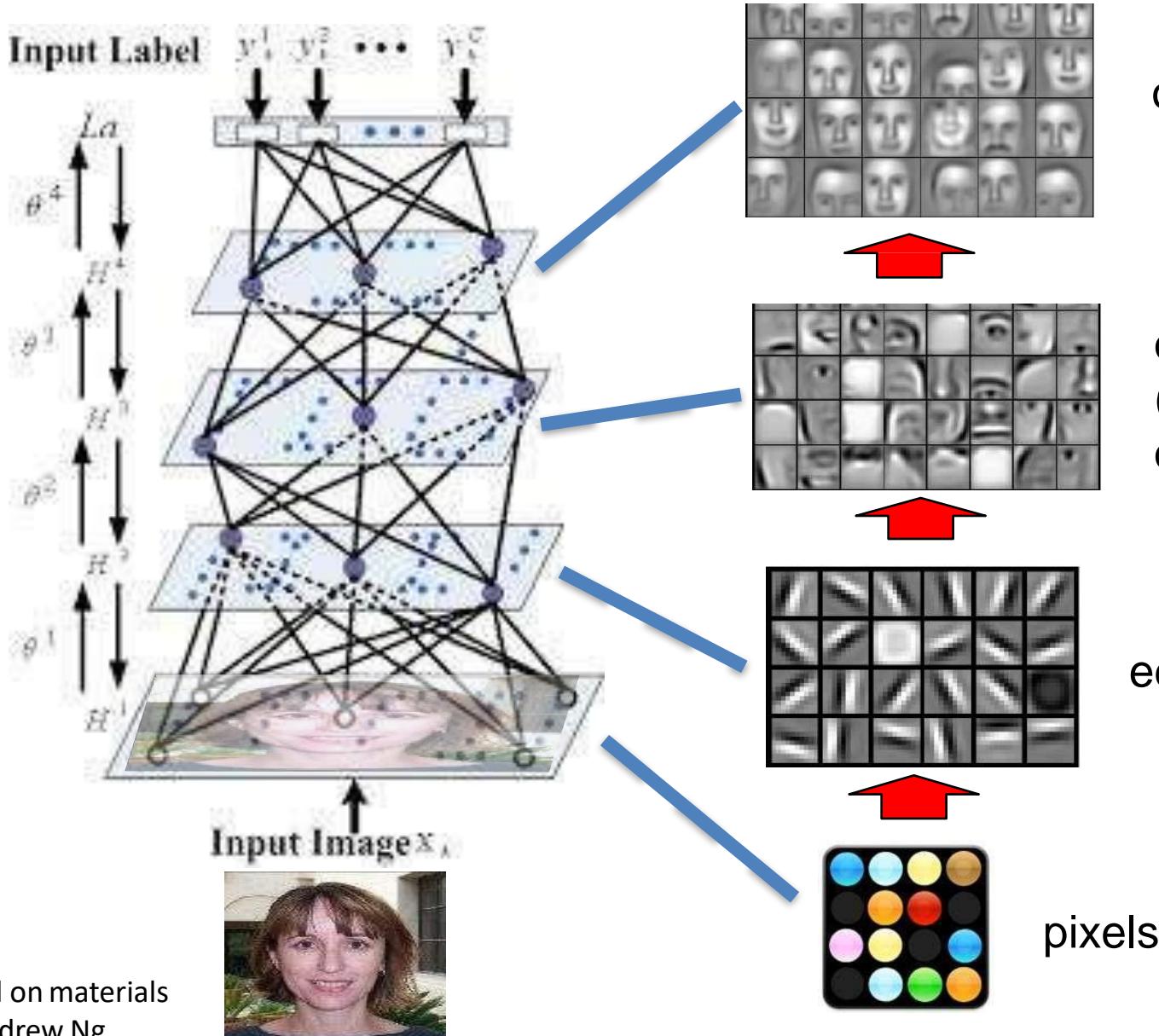
- » Computers learning and growing on their own
- » Able to understand complex, massive amounts of data

DATA ECONOMY
DEEP LEARNING

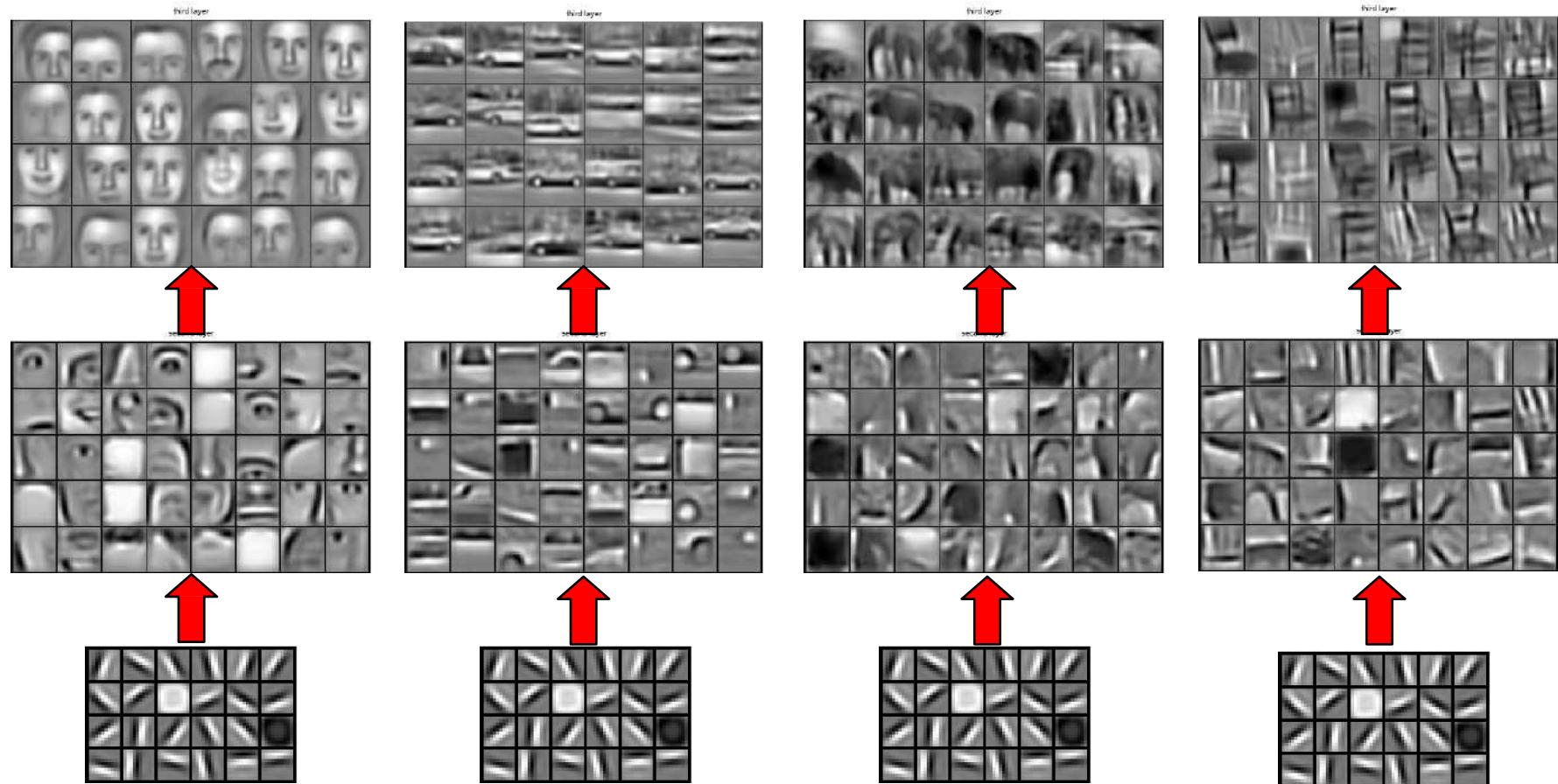
BROUGHT TO YOU BY: GE

CNBC

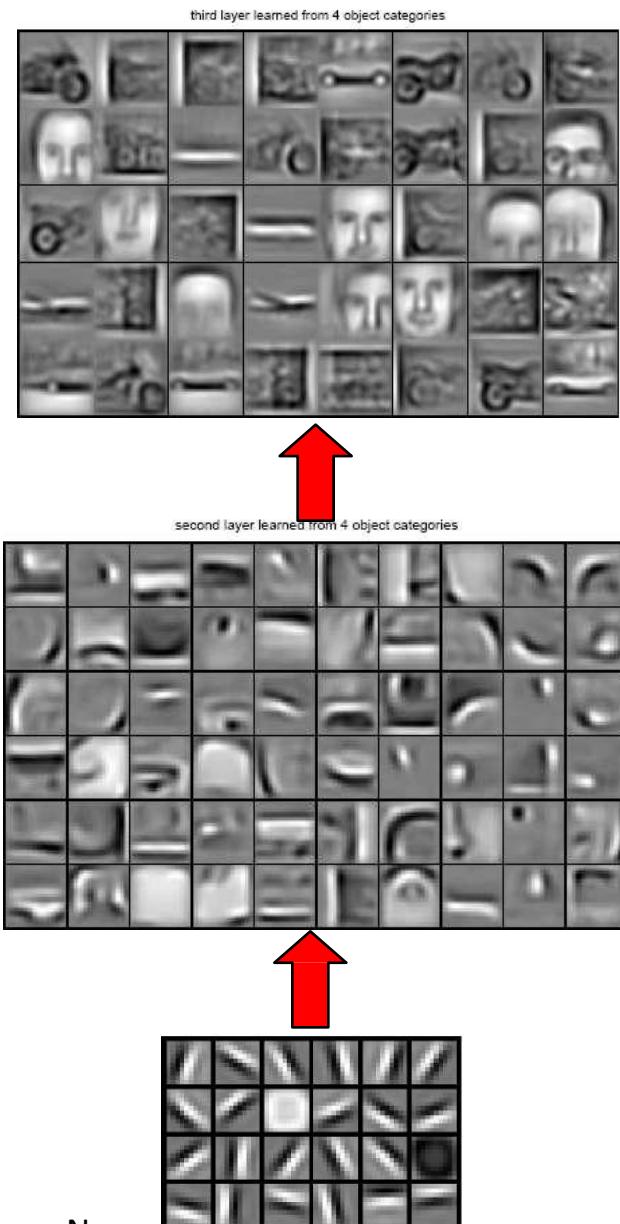
Deep Belief Net za prepozvanje lica



Učenje karakteristika objekata na slikama



Obučavanje na mnogo različitih objekata

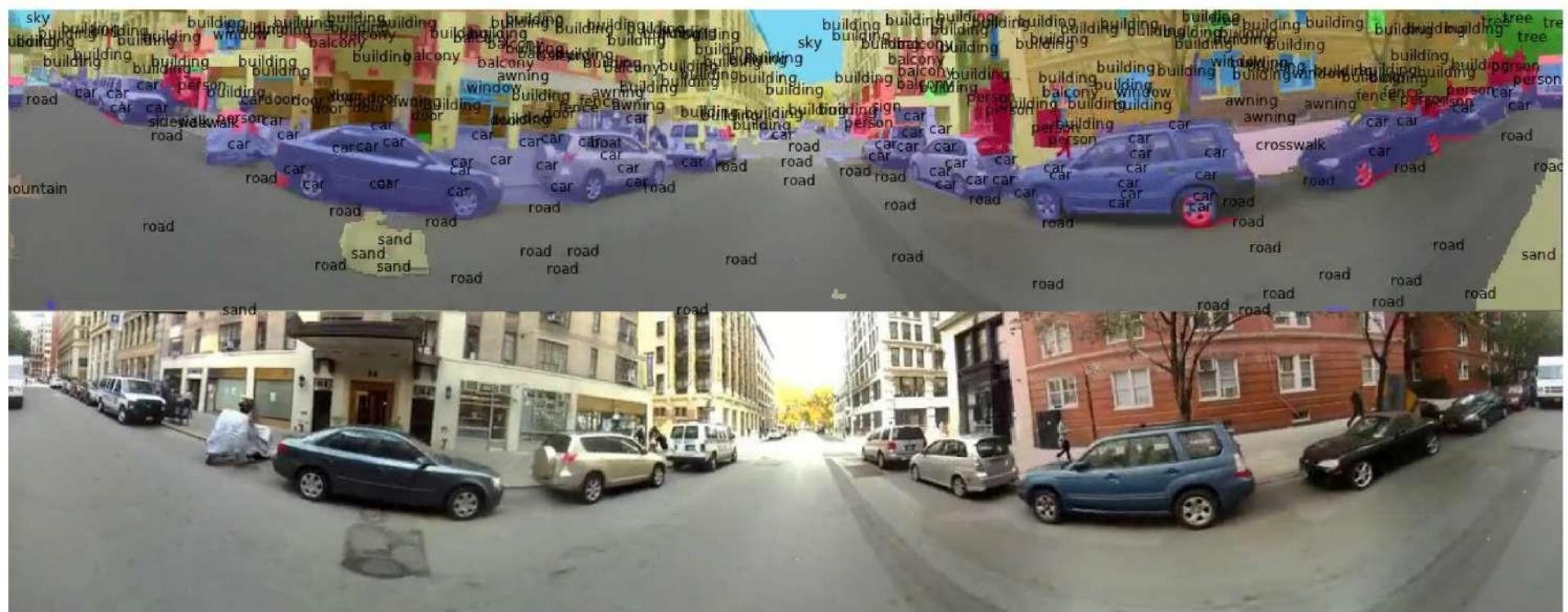


Na primer, 4 klase objekata (automobili, motori, lica i avioni).

Drugi sloj: Karakteristike koje su specifične za svaki objekat i karakteristike koje dele svi objekti

Treći sloj: Još specifičnije karakteristike objekata

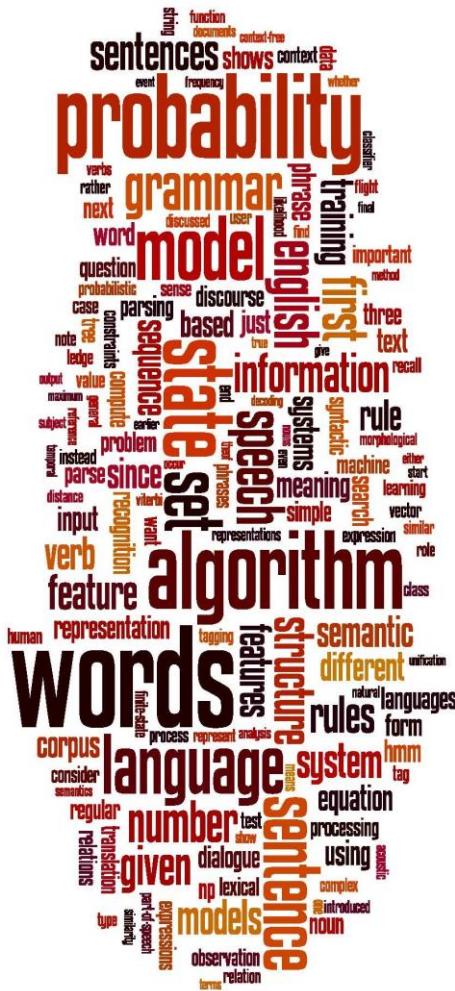
Obeležavanje Scena (*Scene Labeling*) pomoću Deep Learning



[Farabet et al. ICML 2012, PAMI 2013]

Značajan uticaj Deep Learning na sve tehnologije vezane za govor





Naivni Bajesov Klasifikator

Klasifikacija Teksta

Predavač: Aleksandar Kovačević

Slajdovi preuzeti sa CS 124, Stanford

<https://web.stanford.edu/class/cs124/>



Da li je e-mail spam?

Subject: Important notice!

From: Stanford University <newsforum@stanford.edu>

Date: October 28, 2011 12:34:16 PM PDT

To: undisclosed-recipients:;

Greats News!

You can now access the latest news by using the link below to login to Stanford University News Forum.

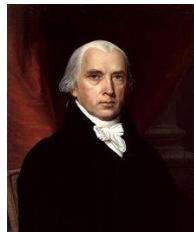
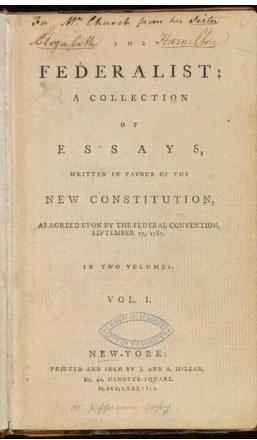
<http://www.123contactform.com/contact-form-StanfordNew1-236335.html>

Click on the above link to login for more information about this new exciting forum. You can also copy the above link to your browser bar and login for more information about the new services.

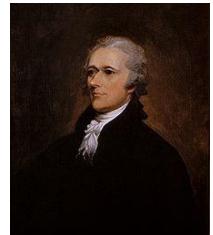


Ko je napisao "The Federalist" dokumente?

- 1787–8: anonimni tekstovi sa ciljem da ubede New York da ratifikuje Ustav S.A.D – prvi autori su Jay, Madison i Hamilton.
- Za 12 dokumenta su se vodile rasprave o tome ko je tačno od trojice navedenih prvi autor
- 1963: problem je rešen od strane Mosteller i Wallace upotrebom Bajesovskih metoda



James Madison



Alexander Hamilton



Da li je autor muško ili žensko?

1. By 1925 present-day Vietnam was divided into three parts under French colonial rule. The southern region embracing Saigon and the Mekong delta was the colony of Cochinchina; the central area with its imperial capital at Hue was the protectorate of Annam...
2. Clara never failed to be astonished by the extraordinary felicity of her own name. She found it hard to trust herself to the mercy of fate, which had managed over the years to convert her greatest shame into one of her greatest assets...



Da li je recenzija filma (knjige, prozivoda...) pozitvina ili negativna?



- unbelievably disappointing
- Full of zany characters and richly applied satire, and some great plot twists



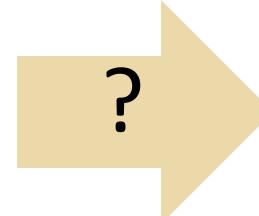
- this is the greatest screwball comedy ever filmed
- It was pathetic. The worst part about it was the boxing scenes.





Šta je tema ovog rada?

Rad sa
MEDLINE



MeSH - Hijerarhija Tema:

- Antagonists and Inhibitors
- Blood Supply
- Chemistry
- Drug Therapy
- Embryology
- Epidemiology
- ...



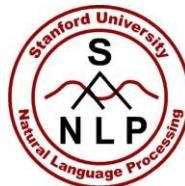
Klasifikacija teksta

- Dodela kategorija, tema ili žanrova
- Detekcija spama
- Identifikacija autora
- Identifikacija pola ili godišta autora
- Identifikacija jezika
- Analiza (detekcija) sentimenta
- ...



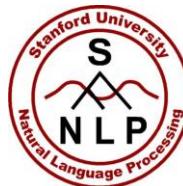
Klasifikacija Teksta: definicija

- *Ulaz:*
 - dokument d
 - skup klasa $C = \{c_1, c_2, \dots, c_J\}$
- *Izlaz:* prediktovana klasa $c \in C$



Klasifikacione metode: Ručno kreirana pravila

- Pravila zasnovana na kombinacijama reči i drugih osobina
 - spam: Na primer, detektovana je e-mail adresa sa „crne liste“ ILI teksta sadrži reči “dollars” i “have been selected”.
- Tačnost može biti visoka
 - Ako se pravila pažljivo kreiraju uz pomoć eksperata
- Ali, kreiranje i održavanje ovih pravila je vremenski zahtevan proces



Klasifikacione metode: Nadgledano Mašinsko Učenje

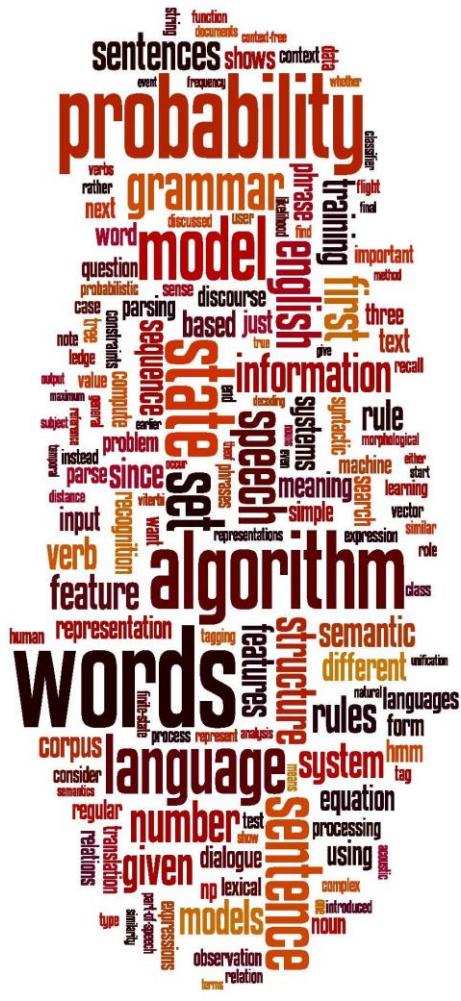
- *Ulaz:*
 - dokument d
 - skup klasa $C = \{c_1, c_2, \dots, c_J\}$
 - Obučavajući skup od m ručno-označenih dokumenata $(d_1, c_1), \dots, (d_m, c_m)$
- *Izlaz:*
 - klasifikator $\gamma: d \rightarrow c$



Klasifikacione metode:

Nadgledano Mašinsko Učenje

- Može se koristiti bilo koji klasifikator
 - Naivni Bajes (*Naïve Bayes*)
 - Logistička regresija
 - Maštine Potpornih Vektora
 - K-Najbližih Komšija
 - ...



Klasifikacija Teksta i Naivni Bajes



Naivni Bajes Intuicija

- Jednostavan (“naivan”) klasifikacioni metod zasnovan na Bajesovoj teoremi
- Oslanja se na jako jednostavnu reprezentaciju dokumenata
 - Vreća reči (*Bag of words*)



Bag-of-words reprezentacija

Y(

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet.

)=C





Bag-of-words reprezentacija

Y(

I **love** this movie! It's **sweet**, but with **satirical** humor. The dialogue is **great** and the adventure scenes are **fun**... It manages to be **whimsical** and **romantic** while **laughing** at the conventions of the fairy tale genre. I would **recommend** it to just about anyone. I've seen it **several** times, and I'm always **happy** to see it **again** whenever I have a friend who hasn't seen it yet.

)=C





Bag-of-words reprezentacija: koristimo podskup svih reči – termin koji se koristi za ovaj podskup je rečnik

Y(

```
x love XXXXXXXXXXXXXXXX sweet
XXXXXX satirical XXXXXXXXXX
XXXXXXXXXX great XXXXXXXX
XXXXXXXXXXXXXX fun     XXXX
XXXXXXXXXXXXXX whimsical XXXX
romantic   XXXX laughing
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXX recommend XXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
xx several XXXXXXXXXXXXXXXX
XXXXX happy XXXXXXXXX again
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXX
```

)=C





Bag-of-words reprezentacija

$\gamma($

great	2
love	2
recommend	1
laugh	1
happy	1
...	...

) = c





Bag-of-words reprezentacija za klasifikaciju dokumenata

Test dokument

parser
language
label
translation
...

?

Machine Learning

learning
training
algorithm
shrinkage
network...

NLP

parser
tag
training
translation
language...

Garbage Collection

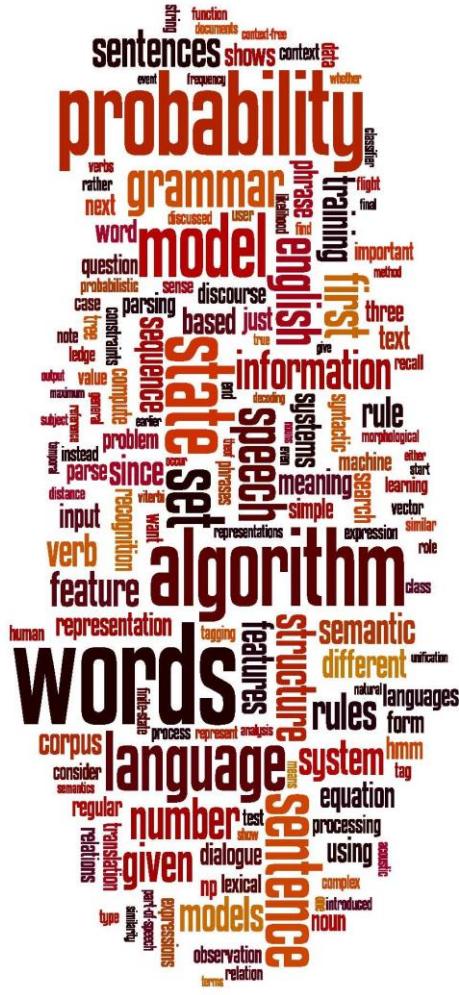
garbage
collection
memory
optimization
plan
region...

Planning

planning
temporal
reasoning
optimization
plan
language...

GUI

...



Naivni Bajesov Klasifikator – Formalnija Definicija



Bajesova teorema primenjena na dokumente i klase

- Za dokument d i klasu c

$$P(c | d) = \frac{P(d | c)P(c)}{P(d)}$$



Bayes-ова теорема

Bayes-ова теорема:

информације које нам доносе подаци

(вероватноћа да се догодио А ако знамо да се догодио С)

априорна вероватноћа- претходно знање

(оно што знамо о догађајима А и С без скупа података)

$$P(C | A) = \frac{P(A | C)P(C)}{P(A)}$$

апостериорна вероватноћа
комбинација претходног
знања и доказа из података



Пример Bayes-ове теореме

Лекар зна да менингитис узрокује укочен врат у 50% случајева
(информације које нам доносе подаци)

Априорна вероватноћа да пациент има менингитис је 1/50,000
(претходно знање)

Априорна вероватноћа да пациент има укочен врат је 1/20
(претходно знање)

Која је вероватноћа да пациент који има укочен врат има менингитис?

$$P(M | S) = \frac{P(S | M)P(M)}{P(S)} = \frac{0.5 \times 1/50000}{1/20} = 0.0002$$



Naivni Bajesov klasifikator

$$c_{MAP} = \operatorname{argmax}_{c \in C} P(c | d)$$

MAP - "maksimalna
aposteriorna verovatnoća" =
najverovatnija klasa

$$= \operatorname{argmax}_{c \in C} \frac{P(d | c)P(c)}{P(d)}$$

Bajesova teorema

$$= \operatorname{argmax}_{c \in C} P(d | c)P(c)$$

Izbacujemo
imenilac



Naivni Bajesov klasifikator

$$c_{MAP} = \operatorname{argmax}_{c \in C} P(d | c)P(c)$$

$$= \operatorname{argmax}_{c \in C} P(x_1, x_2, \dots, x_n | c)P(c)$$

Dokument d
reprezentovan
pomoću osobina
(*features*) $x_1..x_n$



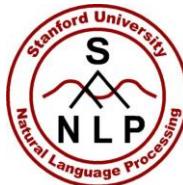
Naivni Bajesov klasifikator

$$c_{MAP} = \operatorname{argmax}_{c \in C} P(x_1, x_2, \dots, x_n | c)P(c)$$

$P(x_1 \dots x_n | C)$ - Može samo da se proceni ako imamo jako jako puno primera u obučavajućem skupu.

$P(c)$ - Koliko se često javlja klasa c ?

Možemo samo da izbrojimo tj. izračunamo relativne frekvencije u korpusu.



Multinomialni Naivni Bajes – pretpostavka o nezavisnosti osobina (atributa)

$$P(x_1, x_2, \dots, x_n | c)$$

- **Pretpostavka Bag-of-Words reprezentacije:**
Prepostavljamo da pozicija reči u tekstu nije važna
- **Uslovna nezavisnost:** Prepostavljamo da su verovatnoće osobina tj. $P(x_i | c_j)$ nezavisne u odnosu na klasu c . Na taj način verovatnoću $P(x_1, \dots, x_n | C)$ računamo kao:

$$P(x_1, \dots, x_n | c) = P(x_1 | c) \bullet P(x_2 | c) \bullet P(x_3 | c) \bullet \dots \bullet P(x_n | c)$$



Multinomialni Naivni Bajes – pretpostavka o nezavisnosti osobina (atributa) $P(x_1, x_2, \dots, x_n | c)$

- Napomena:
- Termin multinominalni odnosi se na način izračunavanja verovatnoća $P(x_i | c)$.
- Pretpostavlja se da vrednosti $P(x_i | c)$ prate multinominalnu distribuciju.
- Postoje NB klasifikatori kod kojih se pretpostavju druge distribucije (binominalna, normalna itd.).
- Primere takvih klasifikatora radićemo na predmetu mašinsko učenje i SIAP.

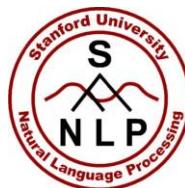
$$P(x_1, \dots, x_n | c) = P(x_1 | c) \bullet P(x_2 | c) \bullet P(x_3 | c) \bullet \dots \bullet P(x_n | c)$$



Multinomialni Naivni Bajes klasifikator

$$c_{MAP} = \operatorname{argmax}_{c \in C} P(x_1, x_2, \dots, x_n | c)P(c)$$

$$c_{NB} = \operatorname{argmax}_{c \in C} P(c_j) \prod_{x \in X} P(x | c)$$



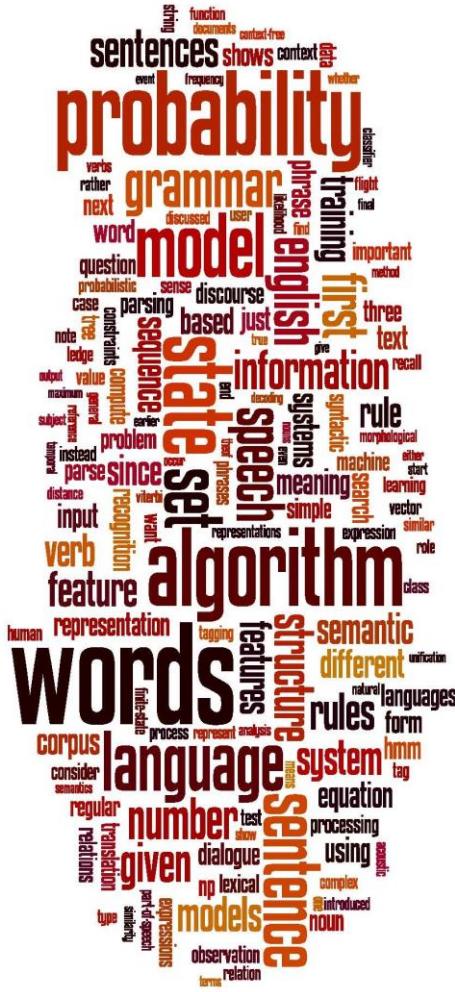
Primena NB klasifikatora na klasifikaciju teksta

`positions ← sve pozicije reči u test dokumentu`

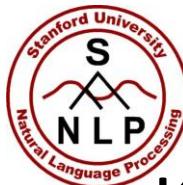
$$c_{NB} = \operatorname{argmax}_{c_j \in C} P(c_j) \prod_{i \in positions} P(x_i | c_j)$$

	Doc	Words	Class
Training	1	Chinese Beijing Chinese	c
	2	Chinese Chinese Shanghai	c
	3	Chinese Macao	c
	4	Tokyo Japan Chinese	j
Test	5	Chinese Chinese Chinese Tokyo Japan	?

Za ovaj primer `positions = {0,1,2,3,4}`



Naivni Bajesov Klasifikator - Obučavanje



Obučavanje NB klasifikatora

- Koristimo metod maksimalne verovatnosti (*maximum likelihood estimates*)
 - konkretno koristimo frekvencije dobijene iz korpusa

$$\hat{P}(c_j) = \frac{doccount(C = c_j)}{N_{doc}}$$

Broj dokumenta klase c_j u korpusu podeljen sa brojem svih dokumenata u korpusu N_{doc}

$$\hat{P}(w_i | c_j) = \frac{count(w_i, c_j)}{\sum_{w \in V} count(w, c_j)}$$

V - predstavlja rečnik odnosno skup svih reči koje koristimo za reprezentaciju dokumenata.



Izračuvanje verovatnoća

$$\hat{P}(w_i | c_j) = \frac{\text{count}(w_i, c_j)}{\sum_{w \in V} \text{count}(w, c_j)}$$

broj pojavljivanja reči w_i u dokumentima koji imaju klasu c_j podeljen sa brojem pojavljivanja svih reči iz rečnika V u svim dokumentima koji imaju klasu c_j

- Kreiramo mega-dokument za klasu j tako što sve dokumente ove klase spojimo u jedan dokument
 - Računamo frekvenciju reču w_i u tom mega-dokumentu



Problem sa prethodnim formulama

- Šta ako nijedan od dokumenata **pozitivne (thumbs-up)** klase u obučavajućem skupu nema reč ***fantastic***?

$$\hat{P}("fantastic" | \text{positive}) = \frac{\text{count}("fantastic", \text{positive})}{\sum_{w \in V} \text{count}(w, \text{positive})} = 0$$

- Bez obzira na vrednosti drugih verovatnoća ako je jedna od verovatnoća 0 verovatnoća klase c je 0!

$$c_{MAP} = \operatorname{argmax}_c \hat{P}(c) \prod_i \hat{P}(x_i | c)$$



Laplasovo (dodaj-1) poravnavanje za NB

Laplace (add-1) smoothing

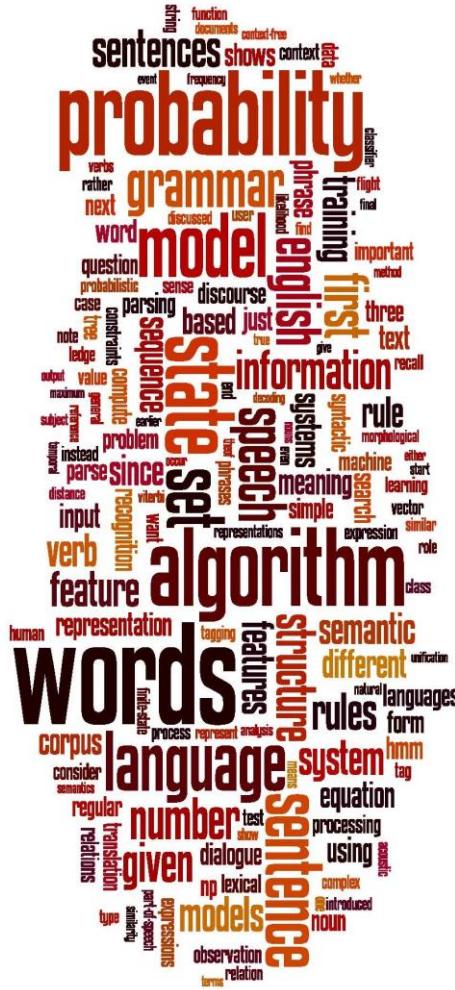
$$\hat{P}(w_i | c) = \frac{\text{count}(w_i, c) + 1}{\sum_{w \in V} (\text{count}(w, c) + 1)}$$

$$= \frac{\text{count}(w_i, c) + 1}{\left(\sum_{w \in V} \text{count}(w, c) \right) + |V|}$$



Multinomialni Naivni Bajes: obučavanje

- Iz obučavajućeg skupa odrediti *Rečink* (*Vocabulary*) – npr. naš rečnik čine 100 najfrekventijih reči u korpusu – naravno postoje i mnoge druge sofisticirane metode za određivanje rečnika.
- Izračunati $P(c_j)$
 - Za svako c_j uraditi
 $docs_j \leftarrow$ svi dokumenti klase c_j
$$P(c_j) \leftarrow \frac{|docs_j|}{|\text{total \# documents}|}$$
- Izračunati $P(w_k | c_j)$
 - $Text_j \leftarrow$ jedan dokument koji sadrži sve $docs_j$
 - Za svaku reč w_k u *Rečinku*
 $n_k \leftarrow$ # pojavljivanja w_k u $Text_j$
$$P(w_k | c_j) \leftarrow \frac{n_k + \alpha}{n + \alpha |Vocabulary|}$$



Naivni Bajes – relacija sa modelima jezika (*language models*)

A word cloud diagram illustrating various terms related to NLP and probability. The most prominent words include 'probability', 'grammar', 'model', 'information', 'state', 'algorithm', 'words', 'language', 'sentence', 'number', and 'models'. Other visible words include 'sentences', 'show', 'context', 'event', 'frequency', 'rather', 'next', 'discussed', 'word', 'first', 'just', 'three', 'text', 'note', 'ledge', 'constraints', 'based', 'give', 'recall', 'subject', 'value', 'maximum', 'minimum', 'subject', 'problem', 'instead', 'parse', 'since', 'distance', 'input', 'recognition', 'verb', 'feature', 'algorithm', 'human', 'representation', 'finite-state', 'process', 'represent', 'analysis', 'semantic', 'different', 'unification', 'natural languages', 'rules', 'form', 'corpus', 'tag', 'equation', 'processing', 'using', 'complex', 'introduced', 'noun', 'type', 'relations', 'translation', 'given', 'dialogue', 'np lexical', 'increasing', 'models', 'observation', 'relation', 'terms'.

NB klasifikator

Primer



$$\hat{P}(c) = \frac{N_c}{N}$$

$$\hat{P}(w|c) = \frac{\text{count}(w,c)+1}{\text{count}(c)+|V|}$$

Verovatnoće klasa:

$$P(c) = \frac{3}{4}$$

$$P(j) = \frac{1}{4}$$

$$V = \{\text{Beijing}, \text{Chinese}, \text{Japan}, \text{Macao}, \text{Tokyo}, \text{Shanghai}\}$$

Određivanje klase:

$$P(c|d_5) \propto \frac{3}{4} * \left(\frac{3}{7}\right)^3 * \frac{1}{14} * \frac{1}{14} \\ \approx 0.0003$$

Uslovne verovatnoće:

$$P(\text{Chinese}|c) = (5+1) / (8+6) = 6/14 = 3/7$$

$$P(\text{Tokyo}|c) = (0+1) / (8+6) = 1/14$$

$$P(\text{Japan}|c) = (0+1) / (8+6) = 1/14$$

$$P(\text{Chinese}|j) = (1+1) / (3+6) = 2/9$$

$$P(\text{Tokyo}|j) = (1+1) / (3+6) = 2/9$$

$$P(\text{Japan}|j) = (1+1) / (3+6) = 2/9$$

	Dok	Reč	Klasa
Ob. skup	1	Chinese Beijing Chinese	c
	2	Chinese Chinese Shanghai	c
	3	Chinese Macao	c
	4	Tokyo Japan Chinese	j
Test skup	5	Chinese Chinese Chinese Tokyo Japan	?



NB – spam filter

- Neke od mogućih osobina – konkretno softver SpamAssassin:
 - Pominjanja: "Generic Viagra", "Online Pharmacy" i slično...
 - Pominjanja: millions of (dollar) ((dollar) NN,NNN,NNN.NN)
 - Fraze: impress ... girl
 - From: ima puno brojeva
 - Subject e-maila je sav velikim slovima all capitals
 - HTML ima malo teksta u odnosu na slike
 - "One hundred percent guaranteed"
 - Recenica u kojoj se tvrdi da mozete da se ojdavite sa liste
 - "Prestigious Non--Accredited Universities"
 - http://spamassassin.apache.org/tests_3_3_x.html



Naivni Bajes nije baš tako naivan

- Vrlo brz, ne treba mu puno memorije
- Robusan na beznačajne osobine
 - Osobine koje su jako slične za sve primere ne menjaju verovatnoće
- Ako prepostavka o nezavisnosti stvarno važi NB je optimalan model
- Čak iako ne važi praksa je pokazala da je NB dobar klasifikator
- U trenutnom stajnu ML oblasti NB je pozdan model i dobar osnonovni model



Matrica Konfuzije (Confusion Matrix)

- Odlučimo koja je klasa za nas pozitivna, koja negativna i formiramo matricu.

	starno poz.	stvarno neg.
predikcija poz.	tp	fp
predikcija neg.	fn	tn



Preciznost i Odziv (*Precision and recall*)

- **Preciznost (*Precision*)** : % predikcija poz. koji su stvarno poz tj.

$$\textit{precision} = \frac{tp}{tp + fp}$$

	starno poz.	stvarno neg.
predikcija poz.	tp	fp
predikcija neg.	fn	tn



Preciznost i Odziv - Primer

- **Preciznost (*Precision*)** : % predikcija poz. koji su stvarno poz tj.

$$\text{precision}_A = \frac{tp_A}{tp_A + fp_A} = \frac{90}{90 + 10} = 0.9$$

	Truth A:	Truth B:
Classifier A:	90	10
Classifier B:	20	890



Preciznost i Odziv (*Precision and recall*)

- **Odziv (Recall)**: % stvarno poz od predikcija poz.

$$\text{recall} = \frac{tp}{tp + fn}$$

	starno poz.	stvarno neg.
predikcija poz.	tp	fp
predikcija neg.	fn	tn



Preciznost i Odziv - Primer

- **Odziv (Recall):** % stvarno poz od predikcija poz.

$$\text{recall}_A = \frac{tp_A}{tp_A + fn_A} = \frac{90}{90 + 20} = 0.81$$

	Truth A:	Truth B:
Classifier A:	90	10
Classifier B:	20	890



F-mera

- Mera koja procenjuje koji je balans između P i R:

$$F = \frac{2PR}{P + R}$$



F-mera - Primer

- Mera koja procenjuje koji je balans između P i R:

$$F_A = \frac{2P_A R_A}{P_A + R_A} = \frac{2 \cdot 0.9 \cdot 0.81}{0.9 + 0.81} = 0.85$$

	Truth A:	Truth B:
Classifier A:	90	10
Classifier B:	20	890



Ako imamo više od dve klase: Supovi binarnih klasifikatora

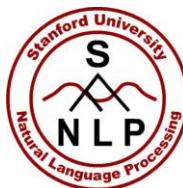
- Problemi kod kojih imamo multi-klasnu klasifikaciju
 - Kasifikujemo dokument koji može da pripada 0, 1, ili >1 klasa.
 - Kad kažemo 0 klasa misli se na situaciju npr. kad dozvoljavamo recimo da dokument bude ni pozitivan ni negativan, ali pripada nekoj klasi koju obično obeležavamo sa *Ostalo*.
- Za svaku klasu $c \in C$
 - Obučiti klasifikator y_c tako da može da razdvoji klasu c od ostalih klasa $c' \in C$
- Za test dokument d ,
 - Primenti svaki model y_c
 - d pripada svakoj klasi c za koju y_c vrati true
 - Ako želimo jasnu klasifikaciju tj. samo jedno c možemo npr. da koristimo y_c koji vraćaju pouzdanaost u true vrednost. Konačna klasa bila bi ona c čiji je model vratio najveću pouzdanost.



Evaluacija:

Jako poznat skup: Reuters--21578 Data Set

- Najčešće korišćen skup, 21,578 dokumenata
- 9603 obučavajući skup, 3299 test skup (ModApte/Lewis split)
- 118 klase
 - Članak (dokument) može da bude u više od jedne kategorije
 - Učimo 118 binarnih klasifikatora
- Samo oko 10 od 118 kategorija ima veći broj članaka
 - Earn (2877, 1087)
 - Acquisitions (1650, 179)
 - Money-fx (538, 179)
 - Grain (433, 149)
 - Crude (389, 189)
 - Trade (369, 119)
 - Interest (347, 131)
 - Ship (197, 89)
 - Wheat (212, 71)
 - Corn (182, 56)



Primer dokumenta iz Reuters--21578

<REUTERS TOPICS="YES" LEWISSPLIT="TRAIN" CGISPLIT="TRAINING-SET" OLDID="12981" NEWID="798">

<DATE> 2-MAR-1987 16:51:43.42</DATE>

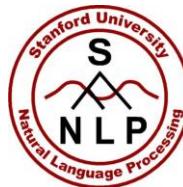
<TOPICS><D>livestock</D><D>hog</D></TOPICS>

<TITLE>AMERICAN PORK CONGRESS KICKS OFF TOMORROW</TITLE>

<DATELINE> CHICAGO, March 2 - </DATELINE><BODY>The American Pork Congress kicks off tomorrow, March 3, in Indianapolis with 160 of the nations pork producers from 44 member states determining industry positions on a number of issues, according to the National Pork Producers Council, NPPC.

Delegates to the three day Congress will be considering 26 resolutions concerning various issues, including the future direction of farm policy and the tax law as it applies to the agriculture sector. The delegates will also debate whether to endorse concepts of a national PRV (pseudorabies virus) control and eradication program, the NPPC said.

A large trade show, in conjunction with the congress, will feature the latest in technology in all areas of the industry, the NPPC added. Reuter



Matrica Konfuzije

- Za svaki par klase $\langle c_1, c_2 \rangle$ koliko dokumenata klase c_1 su pogrešno klasifikovani kao c_2 ?
 - $c_{3,2}$: 90 dokumenata klase *wheat* je pogrešno klasifikovano u klasu *poultry*

Dokumenti u test skup	Predikcija UK	Predikcija poultry	Predikcija wheat	Predikcija coffee	Predikcija interest	Predikcija trade
Stvarno UK	95	1	13	0	1	0
Stvarno poultry	0	1	0	0	0	0
Stvarno wheat	10	90	0	1	0	0
Stvarno coffee	0	0	0	34	3	7
Stvarno interest	-	1	2	13	26	5
Stvarno trade	0	0	2	14	5	10



Evalucioni mere za jednu klasu

Odziv:

Deo dokumenata koji stvano imaju klasu i koji su tačno klasifikovani:

$$\frac{c_{ii}}{\sum_j c_{ij}}$$

, i je indeks vrste, a j kolone u tabeli sa prethodnog slajda – delimo element sa glavne dijagonale sa zbirom elemenata cele vrste.



Evalucioni mere za jednu klasu

Preciznost:

Deo dokumenata kojima je klasifikator doelio klasu i , a koji su stvarno u klasi i :

$$\frac{c_{ii}}{\sum_j c_{ji}}$$

, i je indeks vrste, a j kolone u tabeli sa prethodnog slajda – delimo element sa glavne dijagonale sa zbirom elemenata cele kolone.



Evalucionarna mera na za sve klase zajedno

Tačnost: (1 - error rate)

Deo svih dokumenata koji su tačno klasifikovani, u odnosu a sve
59 dokumente u korpusu:

$$\frac{\sum_i c_{ii}}{\sum_j \sum_i c_{ij}}$$

, i je indeks vrste, a j kolone u tabeli sa prethodnog slajda – delimo zbir svih elemenata sa glavne dijagonale sa zbirom elemenata cele kolone.



Mikro i Makro proseci

- Ako imamo više klasa, na koji način kombinujemo mere na nivou jedne klase u jednu meru?
- **Makro prosek:** Izračunamo meru za svaku klasu i onda uzmemmo prosek (npr. prosek F-mera)
- **Mikro prosek:** Formiramo matricu konfuzije i onda iz nje računamo mere na nivou cele matrice. Npr. za Odziv uradimo zbir svih tp u celoj matrici i podelimo sa svim (tp+fn)



Mikro i Makro proseci: Primer

Klasa 1

	Truth: yes K1	Truth: no K1
Classifier: yes K1	10	10
Classifier: no K1	10	970

Klasa 2

	Truth: yes K2	Truth: no K2
Classifier: yes K2	90	10
Classifier: no K2	10	890

Mikro Prosek Tabela

	Truth: yes	Truth: no
Classifier: yes	100	20
Classifier: no	20	1860

- Makro preciznost: $(0.5 + 0.9)/2 = 0.7$
- Mikro preciznost: $100/120 = 0.83$
- Mikro vrednostima dominiraju klase koje su najfrekventnije



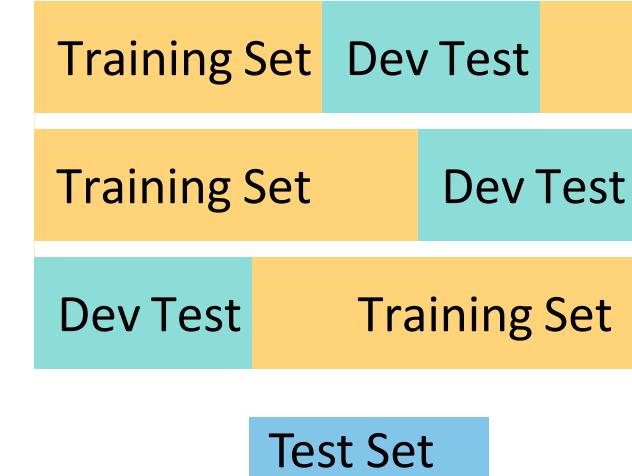
Unakrsna Validacija i Validacioni skup

Obučavajući skup

Validacioni skup

Test skup

- Validacioni skup – koristimo ga tokom razvoja modela za podešavanje parametara i selekciju osobina.
- Test skup – koristimo ga za krajnju procenu kvaliteta modela
 - može da bude varljiv (previše lak ili previše težak test skup)
- Unakrsna validacija (objašnjeno na sledećem slajdu) – može da se koristi i tokom razvoja ili za krajnju procenu. Nikako za oba u isto vreme. Ako je koristimo u toku razvoja, moramo da imamo odvojen test skup.





K-tostruka unakrsna validacija (k-fold cross-validation)

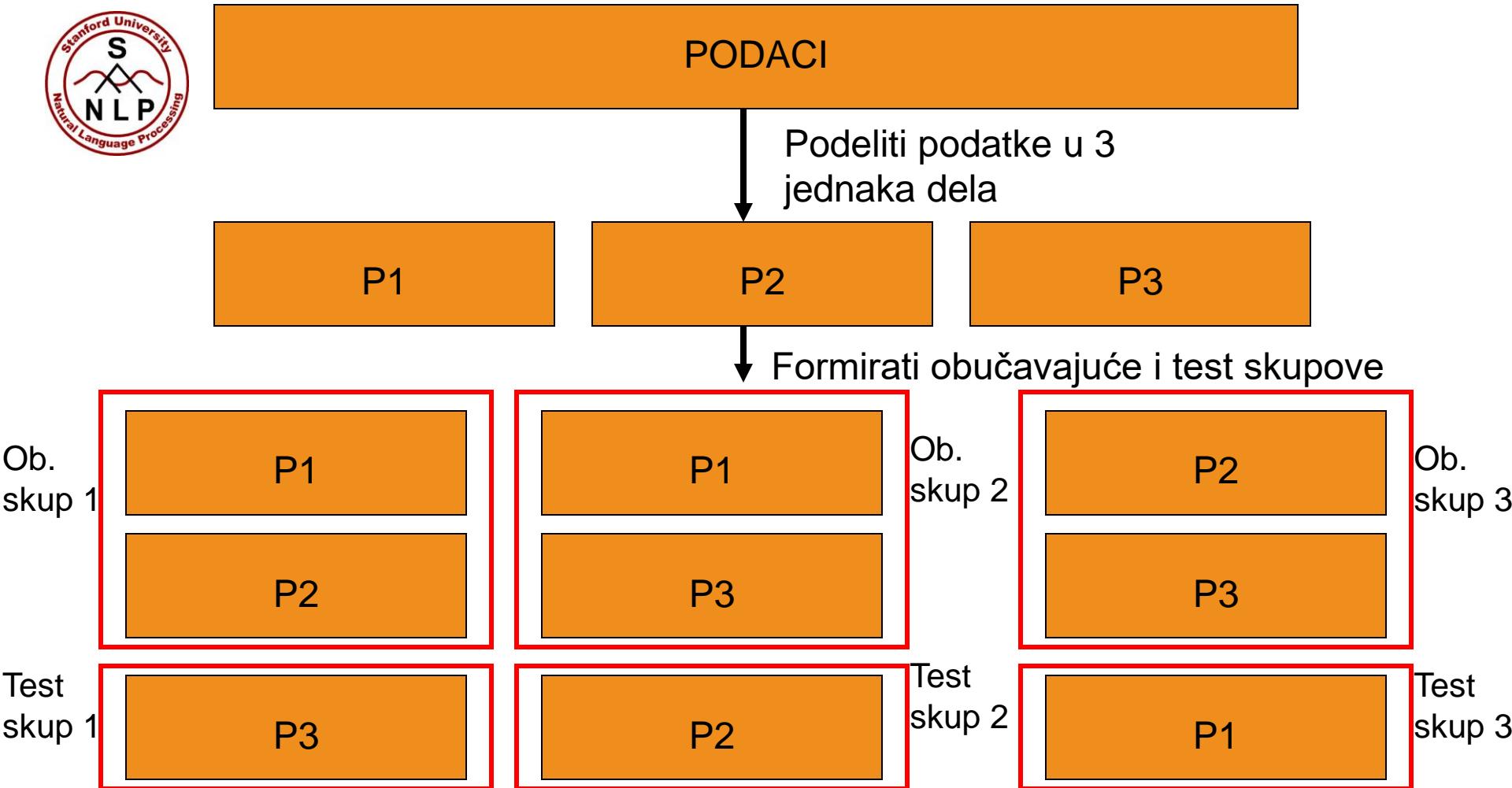
1. Podeliti obučavajući skup u k jednakih delova (*folds*).
 2. Formirati sve moguće kombinacije delova na ovaj način:
 - obučiti model na $(k_1 + \dots + k_{n-1})$, testirati na k_n
 - obučiti model na $(k_1 + \dots + k_{n-2} + k_n)$, testirati na k_{n-1}
 -
- Izračunati prosek performansi svih modela.

Obično se podela vrši na slučajan način.

Moguće je i koristi stratifikovano uzorkovanje kada želimo da odnos klase koji je ubučavajućem skupu bude isti u svakom delu.
Npr. u celom skupu ima 2/3 pozitivne i 1/3 neg klase – tako onda uzorkujemo svaki deo.



Trostruka unakrsna validacija (3-fold cross-validation)





NB za klasifikaciju teksta

Problemi u praksi



Nedostatak obučavajućeg skupa?

Treba koristiti Ručno kreirana pravila

Ako sadrži reči "wheat" ili "grain", a ne sadrži "whole" ili "bread" onda je kategorija *grain*

- Moramo pažljivo da kreiramo pravila
 - Treba nam bar neki manji označen skup da bi mogli da naštelujemo pravila
 - Kreiranje pravila je vremenski zahtevan posao



Jako mali obučavajući skup?

- Korisiti Naivni Bajesov model
 - NB ima veliki *bijas* i malu *varijansu*
 - Ova dva pojma ću grubo objasniti na predavanju, a u detalje ih učite na predmetu Mašinsko Učenje i SIAP
- Povećati obučavajući skup
 - „Ubediti“ ljude da vam označe podatke
- Probati metode polu-nadgledanog učenja:
 - Co-training itd. – više na predmetu MU



Relativno veliki obučavajući skup?

- Savršena postavka za kompleksnije modele
 - SVM
 - Regularized Logistic Regression
 - Oba modela ćemo objasniti kasnije tokom kursa
- Možete koristiti i stabla odlučivanja
 - Interpretabilan model
 - Lako je objasniti zašto je neki dokument dobio neku klasu



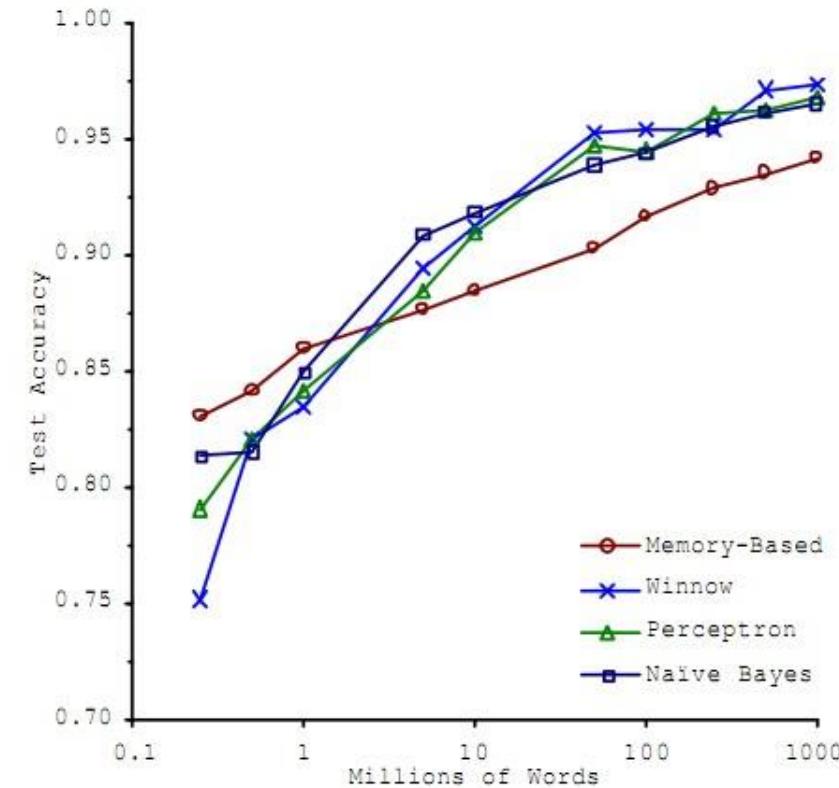
Jako veliki obučavajući skup?

- Možemo dobiti jako veliku tačnost!
- Cena je naravno brzina:
 - SVM (sporo obučavanje) ili kNN (spora primena na test skupu)
- Naivni Bajes je ovde odlična opcija jer je brz!
- Naravno *Deep Learning* je uvek odlična opcija kada imate puno podataka.



Tačnost kao funkcija veličine obučavajućeg skupa

- Sa velikim ob. skupom
 - Skoro da nije važno koji model koristite



Brill and Banko on spelling correction



Sistemi koji se koriste u praksi: često kombinuju ML modele i ručno kreirana pravila

- Automatska klasifikacija
- Analiza grešaka
- Ručno kerirati pravila da reše tipične greške tj. slučajeve koji su koji su teški za ML



U praksi imamo problem sa jako malim vrednostima (*underflow*) Rešenje: logaritmovanje

- Množenje puno verovatnoća može da rezultuje jako malim vrednostima.
- Pošto važi $\log(xy) = \log(x) + \log(y)$
 - Bolje je sabrati logaritme verovatnoća nego množiti verovatnoće.
- Log je monotona funkcija pa se rezultati modela ne menjaju

$$c_{NB} = \operatorname{argmax}_{c_j \in C} \log P(c_j) + \sum_{i \in positions} \log P(x_i | c_j)$$



Kako doštelovati (tweak) performanse modela

- Osobine specifične za domen – nije isto kad klasifikujemo tvitove ili blogove, nije isto kad radimo sa naučim radovima ili *Blic* vestima
- Nekada moramo da normalizujemo neke delove:
 - Umesto 1234, 258 imamo BROJ, BROJ, slično za hemijske formule, ...
- Nekada pomaže *Upweighting*: Neke reči brojimo kao da se pojavljuju dva puta, npr:
 - naslov ([Cohen & Singer 1996](#))
 - prva rečenica svakog pasusa ([Murata, 1999](#))
 - rečenice koje sadrže reči iz naslova ([Ko et al, 2002](#))

Klasifikacija Slika

K-Najbližih Komšija

Uvod u Linearne Klasifikatore

Predavač: Aleksandar Kovačević

Slajdovi preuzeti sa CS 231n, Stanford

<http://cs231n.stanford.edu/>

Klasifikacija Slika: Važan zadatak u Kompjuterskoj Viziji (*Computer Vision*)



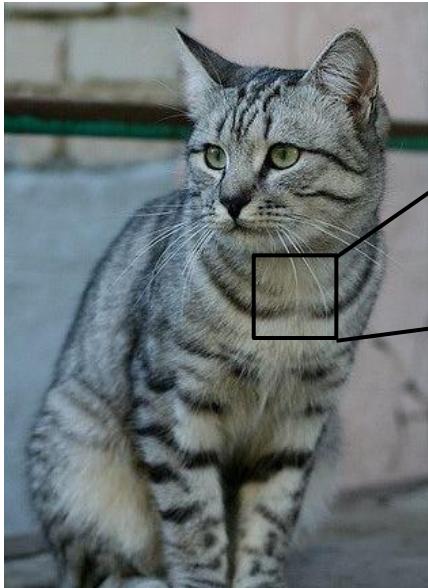
This image by [Nikita](#) is
licensed under [CC-BY2.0](#)

(dat je skup diskretnih klasa)
{dog, cat, truck, plane, ...}



cat

Problem: Ono što računar vidi vs. Ono što želimo da uradi



[[105 112 108 111 104 99 106 99 96 103 112 119 104 97 93 87]
[91 98 102 106 104 79 98 103 99 105 123 136 110 105 94 95 85]
[76 85 90 105 128 105 87 96 95 99 115 112 106 103 99 85]
[99 81 81 93 120 131 127 100 95 98 102 99 96 93 101 94]
[106 91 61 64 69 91 88 85 101 107 109 98 75 84 96 95]
[114 108 85 55 55 69 64 54 64 87 112 129 98 74 84 91]
[133 137 147 103 65 81 80 65 52 54 74 84 102 93 85 82]
[128 137 144 140 109 95 86 70 62 65 63 63 60 73 86 101]
[125 133 148 137 119 121 117 94 65 79 80 65 54 64 72 98]
[127 125 131 147 133 127 126 131 111 96 89 75 61 64 72 84]
[115 114 109 123 150 148 131 118 113 109 100 92 74 65 72 78]
[89 93 90 97 108 147 131 118 113 114 113 109 106 95 77 80]
[63 77 86 81 77 79 102 123 117 115 117 125 125 130 115 87]
[62 65 82 89 78 71 80 101 124 126 119 101 107 114 131 119]
[63 65 75 88 89 71 62 81 128 138 135 105 81 98 110 118]
[87 65 71 87 106 95 69 45 76 130 126 107 92 94 105 112]
[118 97 82 86 117 123 116 66 41 51 95 93 89 95 102 107]
[164 146 112 88 82 120 124 104 76 48 45 66 88 101 102 109]
[157 170 157 128 93 86 114 132 112 97 69 55 70 82 99 94]
[130 128 134 161 139 100 109 118 121 134 114 87 65 53 69 86]
[128 112 96 117 150 144 128 115 104 107 102 93 87 81 72 79]
[123 107 96 86 83 112 153 149 122 109 104 75 80 107 112 99]
[122 121 102 88 82 86 94 117 145 148 153 102 58 78 92 107]
[122 164 148 103 71 56 78 83 93 103 119 139 102 61 69 84]]

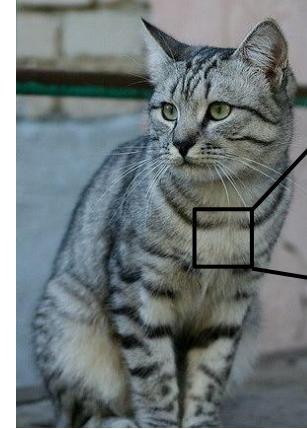
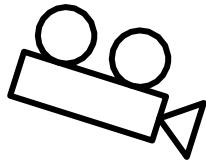
Ono što računari vidi

Slika je matrica brojeva u rasponu [0, 255]:

npr. 800 x 600 x 3
(3 kanala - RGB)

This image by [Nikita](#) is
licensed under [CC-BY2.0](#)

Izazovi: Pomeranje tačke iz koje gledamo



```
[1185 112 188 111 184 99 186 99 96 183 112 119 184 97 93 87]  
[ 91 98 182 106 184 79 98 103 99 105 123 136 118 105 94 85]  
[ 76 85 98 105 128 105 87 96 95 99 115 112 106 103 99 85]  
[ 99 80 81 86 102 103 127 96 98 99 100 101 102 108 109 98 95]  
[104 91 85 86 64 60 91 68 85 101 102 108 109 98 75 84 96 95]  
[114 108 85 55 55 69 64 54 64 87 112 129 98 74 84 94 91]  
[133 137 147 103 65 81 80 65 52 54 74 84 102 93 85 82]  
[128 137 144 148 105 95 86 78 62 65 63 63 68 73 86 101]  
[102 125 131 147 133 127 116 131 111 98 89 75 61 64 72 84]  
[127 125 131 147 133 127 116 131 111 98 89 75 61 64 72 84]  
[115 115 189 123 150 148 131 118 113 109 108 92 74 65 72 78]  
[ 89 93 98 97 108 147 131 118 113 113 114 113 108 106 95 77 80]  
[ 63 77 86 81 77 79 182 123 137 115 111 125 125 130 115 87]  
[ 62 85 88 89 73 62 81 128 138 135 105 81 98 118 118]  
[ 63 65 75 88 89 73 62 81 128 138 135 105 81 98 118 118]  
[ 87 65 71 87 100 95 69 45 76 126 126 107 92 94 105 112]  
[118 97 82 86 117 123 116 66 41 51 95 93 89 95 102 107]  
[164 140 112 88 83 102 126 184 175 48 66 70 101 102 108]  
[137 98 109 134 108 86 104 128 125 115 105 95 70 84 94]  
[138 128 134 161 139 180 109 118 121 134 114 87 65 53 69 86]  
[128 112 96 117 150 144 120 115 104 107 102 93 87 81 72 79]  
[123 107 96 86 83 112 153 149 122 189 104 75 88 107 112 99]  
[122 121 102 88 82 86 94 117 145 148 153 105 58 78 92 107]  
[22 164 148 103 71 56 78 83 93 103 119 139 102 61 69 84]
```

Svi pikseli se promene
kad pomerimo
kameru!

Izazovi: Osvetljenost



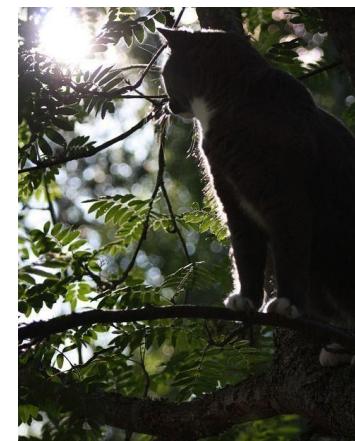
[This image](#) is [CC0 1.0](#) public domain



[This image](#) is [CC0 1.0](#) public domain



[This image](#) is [CC0 1.0](#) public domain



[This image](#) is [CC0 1.0](#) public domain

Izazovi: Deformacija



[This image](#) by [Umberto Salvagnin](#)
is licensed under [CC-BY 2.0](#)



[This image](#) by [Umberto Salvagnin](#)
is licensed under [CC-BY 2.0](#)



[This image](#) by [sare bear](#) is
licensed under [CC-BY 2.0](#)



[This image](#) by [Tom Thai](#) is
licensed under [CC-BY 2.0](#)

Izazovi: Okluzija



[This image](#) is [CC0 1.0](#) public domain



[This image](#) is [CC0 1.0](#) public domain



[This image](#) by [jonsson](#) is licensed
under [CC-BY 2.0](#)

Izazovi: Pretrpana pozadina



[This image](#) is [CC0 1.0](#) public domain



[This image](#) is [CC0 1.0](#) public domain

Izazaovi: Varijabilnost unutar iste klase



[This image](#) is [CC0 1.0](#) public domain

Klasifikator Slika

```
def classify_image(image):  
    # Some magic here?  
    return class_label
```

Za razliku od npr. sortiranja liste,

ne postoji jednostavan način da hard-kodujemo (*hard-code*) algoritam za prepoznavanje mačaka ili drugih klasa.

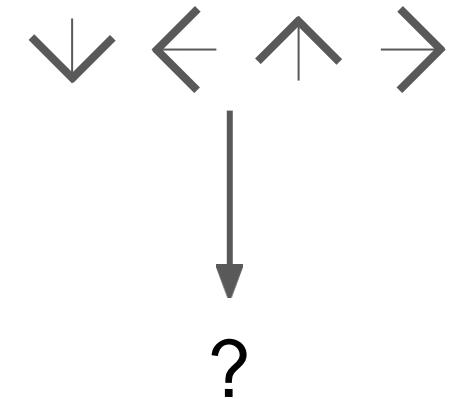
Postoji jako puno prisupa problemu klasifikacije slika



Pronaći
ivice



Pronaći
uglove



John Canny, "A Computational Approach to Edge Detection", IEEE TPAMI 1986

Upotreba Nadgledanog Učenja

1. Prikupiti skup slika i oznaka klase
2. Upotrebiti nadgledano učenje za obučavanje klasifikatora
3. Evaluirati klasifikator na novim slikama

Primer Obučavajućeg skupa

```
def train(images, labels):  
    # Machine learning!  
    return model
```

```
def predict(model, test_images):  
    # Use model to predict labels  
    return test_labels
```

airplane



automobile



bird



cat



deer



Klasifikator K-Najbližih Komšija *K-Nearest Neighbours, KNN*

```
def train(images, labels):  
    # Machine learning!  
    return model
```

→ Čuvamo (pamtimo)
sve slike i njihove
oznake

```
def predict(model, test_images):  
    # Use model to predict labels  
    return test_labels
```

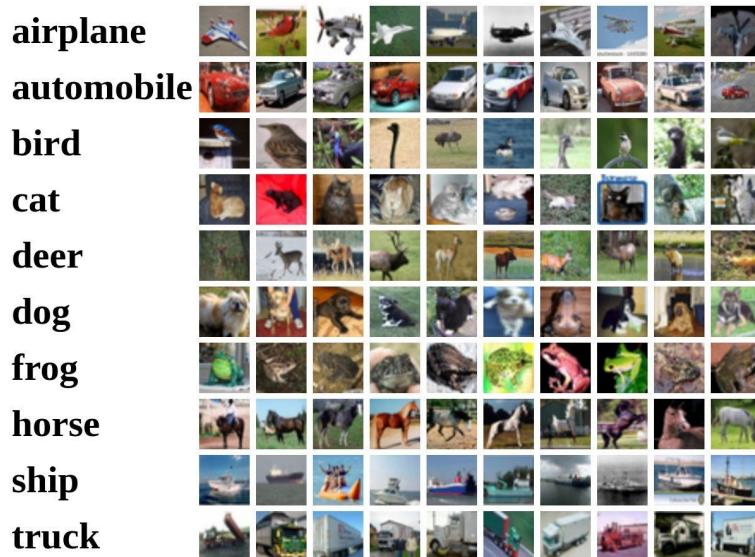
→ Za novu (test) sliku:
Rezultat predikcije je
oznaka klase najsličnije
slike koju imamo u
obučavajućem skupu

Primer – skup podataka CIFAR10

10 klasa

50,000 slika u obučavajućem skupu

10,000 slika u test skupu



Alex Krizhevsky, "Learning Multiple Layers of Features from Tiny Images", Technical Report, 2009.

Primer – skup podataka CIFAR10

10 klasa

50,000 slika u obučavajućem skupu

10,000 slika u test skupu

airplane



automobile



bird



cat



deer



dog



frog



horse



ship



truck



Test slike, Primena KNN



Alex Krizhevsky, "Learning Multiple Layers of Features from Tiny Images", Technical Report, 2009.



Mera sličnost (udaljenosti) za poređenje slika

L1 mera:

$$d_1(I_1, I_2) = \sum_p |I_1^p - I_2^p|$$

test image				training image				pixel-wise absolute value differences			
56	32	10	18	10	20	24	17	46	12	14	1
90	23	128	133	8	10	89	100	82	13	39	33
24	26	178	200	12	16	178	170	12	10	0	30
2	0	255	220	4	32	233	112	2	32	22	108

-

=

add → 456

```
import numpy as np

class NearestNeighbor:
    def __init__(self):
        pass

    def train(self, X, y):
        """ X is N x D where each row is an example. Y is 1-dimension of size N """
        # the nearest neighbor classifier simply remembers all the training data
        self.Xtr = X
        self.ytr = y

    def predict(self, X):
        """ X is N x D where each row is an example we wish to predict label for """
        num_test = X.shape[0]
        # lets make sure that the output type matches the input type
        Ypred = np.zeros(num_test, dtype = self.ytr.dtype)

        # loop over all test rows
        for i in xrange(num_test):
            # find the nearest training image to the i'th test image
            # using the L1 distance (sum of absolute value differences)
            distances = np.sum(np.abs(self.Xtr - X[i,:]), axis = 1)
            min_index = np.argmin(distances) # get the index with smallest distance
            Ypred[i] = self.ytr[min_index] # predict the label of the nearest example

        return Ypred
```

KNN klasifikator

```

import numpy as np

class NearestNeighbor:
    def __init__(self):
        pass

    def train(self, X, y):
        """ X is N x D where each row is an example. Y is 1-dimension of size N """
        # the nearest neighbor classifier simply remembers all the training data
        self.Xtr = X
        self.ytr = y

    def predict(self, X):
        """ X is N x D where each row is an example we wish to predict label for """
        num_test = X.shape[0]
        # lets make sure that the output type matches the input type
        Ypred = np.zeros(num_test, dtype = self.ytr.dtype)

        # loop over all test rows
        for i in xrange(num_test):
            # find the nearest training image to the i'th test image
            # using the L1 distance (sum of absolute value differences)
            distances = np.sum(np.abs(self.Xtr - X[i,:]), axis = 1)
            min_index = np.argmin(distances) # get the index with smallest distance
            Ypred[i] = self.ytr[min_index] # predict the label of the nearest example

        return Ypred

```

KNN klasifikator

Čuvamo (pamtimo)
obučavajući skup

```

import numpy as np

class NearestNeighbor:
    def __init__(self):
        pass

    def train(self, X, y):
        """ X is N x D where each row is an example. Y is 1-dimension of size N """
        # the nearest neighbor classifier simply remembers all the training data
        self.Xtr = X
        self.ytr = y

    def predict(self, X):
        """ X is N x D where each row is an example we wish to predict label for """
        num_test = X.shape[0]
        # lets make sure that the output type matches the input type
        Ypred = np.zeros(num_test, dtype = self.ytr.dtype)

        # loop over all test rows
        for i in xrange(num_test):
            # find the nearest training image to the i'th test image
            # using the L1 distance (sum of absolute value differences)
            distances = np.sum(np.abs(self.Xtr - X[i,:]), axis = 1)
            min_index = np.argmin(distances) # get the index with smallest distance
            Ypred[i] = self.ytr[min_index] # predict the label of the nearest example

        return Ypred

```

KNN klasifikator

Za test sliku:

Naći nasličniju (najближу) sliku
iz obučavajućeg skupa
Vratiti oznaku klase te slike kao
predikciju klase za test sliku

```

import numpy as np

class NearestNeighbor:
    def __init__(self):
        pass

    def train(self, X, y):
        """ X is N x D where each row is an example. Y is 1-dimension of size N """
        # the nearest neighbor classifier simply remembers all the training data
        self.Xtr = X
        self.ytr = y

    def predict(self, X):
        """ X is N x D where each row is an example we wish to predict label for """
        num_test = X.shape[0]
        # lets make sure that the output type matches the input type
        Ypred = np.zeros(num_test, dtype = self.ytr.dtype)

        # loop over all test rows
        for i in xrange(num_test):
            # find the nearest training image to the i'th test image
            # using the L1 distance (sum of absolute value differences)
            distances = np.sum(np.abs(self.Xtr - X[i,:]), axis = 1)
            min_index = np.argmin(distances) # get the index with smallest distance
            Ypred[i] = self.ytr[min_index] # predict the label of the nearest example

        return Ypred

```

KNN klasifikator

Sa N slika u obučavajućem skupu, koliko traje obučavanje i primena na test sliku?

```

import numpy as np

class NearestNeighbor:
    def __init__(self):
        pass

    def train(self, X, y):
        """ X is N x D where each row is an example. Y is 1-dimension of size N """
        # the nearest neighbor classifier simply remembers all the training data
        self.Xtr = X
        self.ytr = y

    def predict(self, X):
        """ X is N x D where each row is an example we wish to predict label for """
        num_test = X.shape[0]
        # lets make sure that the output type matches the input type
        Ypred = np.zeros(num_test, dtype = self.ytr.dtype)

        # loop over all test rows
        for i in xrange(num_test):
            # find the nearest training image to the i'th test image
            # using the L1 distance (sum of absolute value differences)
            distances = np.sum(np.abs(self.Xtr - X[i,:]), axis = 1)
            min_index = np.argmin(distances) # get the index with smallest distance
            Ypred[i] = self.ytr[min_index] # predict the label of the nearest example

        return Ypred

```

KNN klasifikator

Sa N slika u obučavajućem skupu, koliko traje obučavanje i primena na test sliku?

Odgovor:

Obučavanje $O(1)$,
Primena na test sliku $O(N)$

```

import numpy as np

class NearestNeighbor:
    def __init__(self):
        pass

    def train(self, X, y):
        """ X is N x D where each row is an example. Y is 1-dimension of size N """
        # the nearest neighbor classifier simply remembers all the training data
        self.Xtr = X
        self.ytr = y

    def predict(self, X):
        """ X is N x D where each row is an example we wish to predict label for """
        num_test = X.shape[0]
        # lets make sure that the output type matches the input type
        Ypred = np.zeros(num_test, dtype = self.ytr.dtype)

        # loop over all test rows
        for i in xrange(num_test):
            # find the nearest training image to the i'th test image
            # using the L1 distance (sum of absolute value differences)
            distances = np.sum(np.abs(self.Xtr - X[i,:]), axis = 1)
            min_index = np.argmin(distances) # get the index with smallest distance
            Ypred[i] = self.ytr[min_index] # predict the label of the nearest example

        return Ypred

```

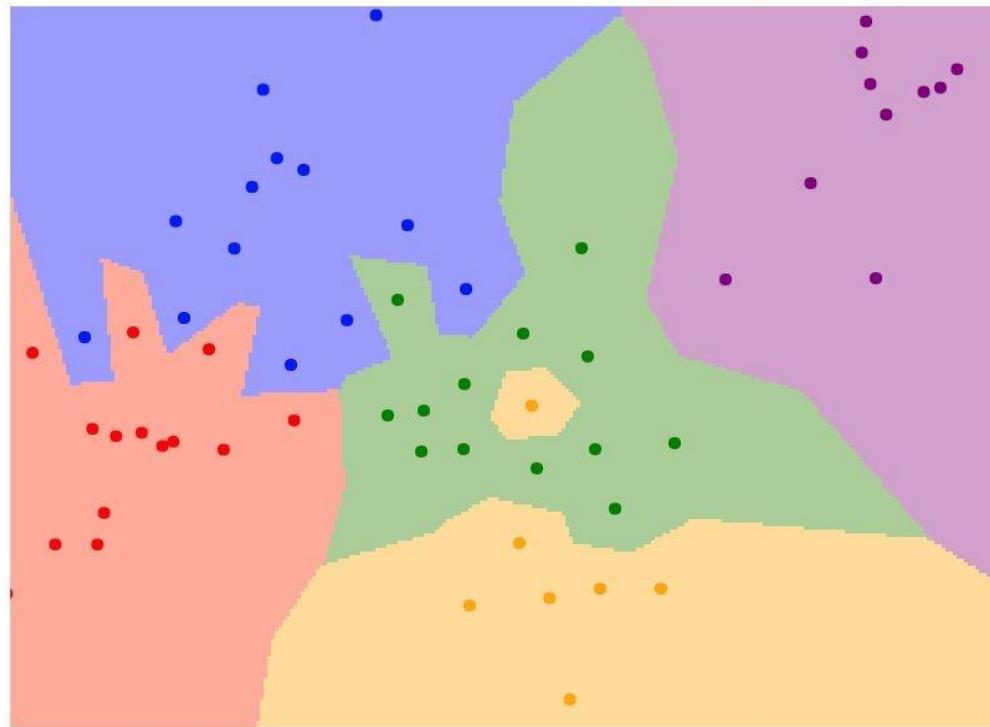
KNN klasifikator

Sa N slika u obučavajućem skupu, koliko traje obučavanje i primena na test sliku?

Odgovor:
 Obučavanje O(1),
 Primena na test
 sliku O(N)

To nije ono što želimo:
 hoćemo klasifikatore koji se
brzo primenjuju, dok je
sporo obučavanje prihvatljivo

KNN u 2d

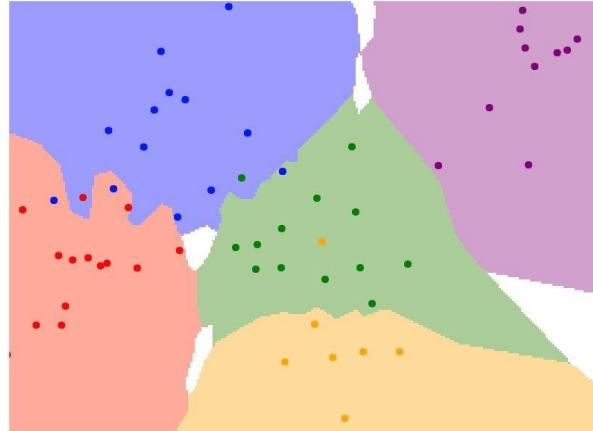


KNN u 2d

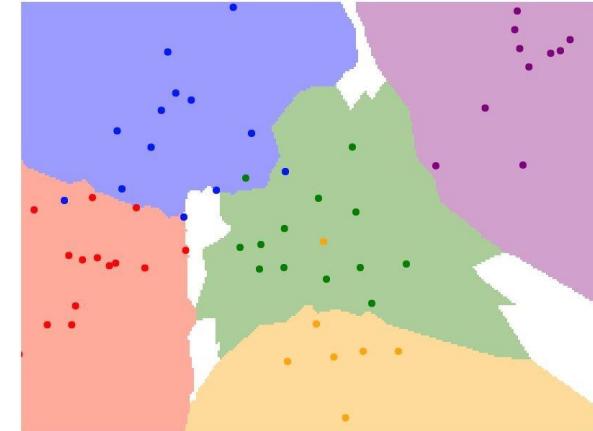
Umesto da gledamo klasu samo jedne najблиže slike,
uradimo **većinsko glasanje** prvih K najблиžih slika



$K = 1$



$K = 3$



$K = 5$

Kako to izgleda u praksi?



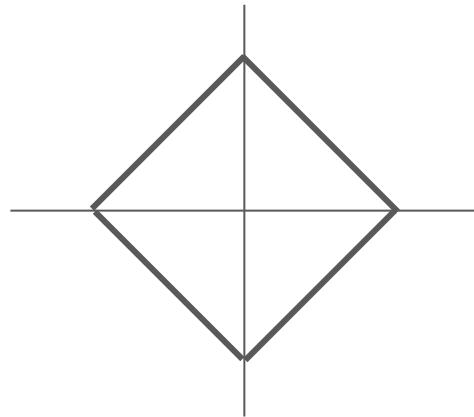
Kako to izgleda u praksi?



K-Najbližih Komšija: Mera Sličnosti

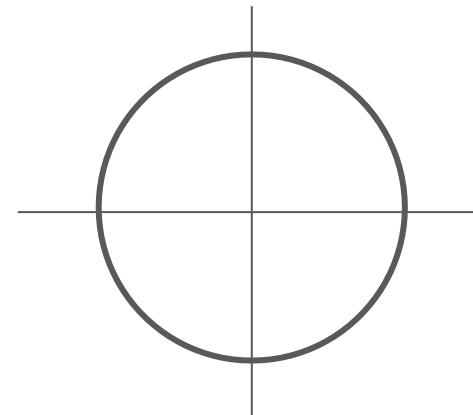
L1 (Menhetn) rastojanje

$$d_1(I_1, I_2) = \sum_p |I_1^p - I_2^p|$$



L2 (Euklidsko) rastojanje

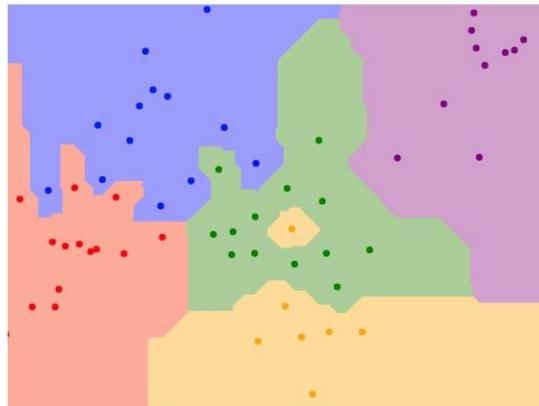
$$d_2(I_1, I_2) = \sqrt{\sum_p (I_1^p - I_2^p)^2}$$



K-Najbližih Komšija: Mera Sličnosti

L1 (Menhetin) rastojanje

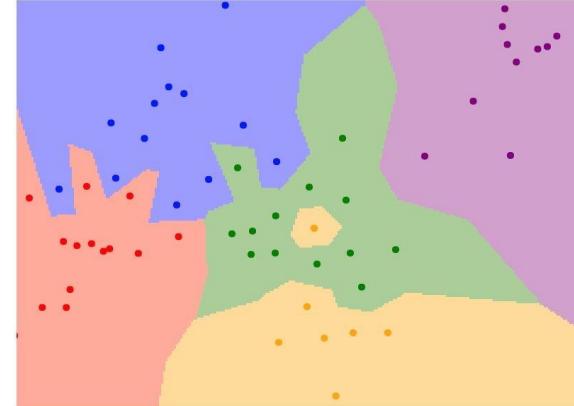
$$d_1(I_1, I_2) = \sum_p |I_1^p - I_2^p|$$



$$K = 1$$

L2 (Euklidsko) rastojanje

$$d_2(I_1, I_2) = \sqrt{\sum_p (I_1^p - I_2^p)^2}$$



$$K = 1$$

Hiper-parametri (*Hyperparameters*)

Koja je najbolja vrednost za k ?

Koja je nabolja **mera sličnosti**?

Ovo su **hiper-parametri**: odluke koje moramo da donesemo o modelu pre procesa učenja.

Postoje metode koje nam mogu pomoći da donešemo odluku, neke od njih ćemo raditi tokom krusa

Hiper-parametri (*Hyperparameters*)

Koja je najbolja vrednost za za k?

Koja je nabolja **mera sličnosti**?

Ovo su **hiper-parametri**: odluke koje moramo da donesemo o modelu pre procesa učenja.

Hiper-parametri jako zavise od problema. Nema univerzalnih vrednosti!

Ako je moguće treba probati sve izbore da bi našli najbolji.

Kako određujemo hiper-parametre?

Način #1: Biramo vrednosti koje daju najbolje rezultate na obučavajućem skupu

Obučavajući Skup

Kako određujemo hiper-parametre?

Način #1: Biramo vrednosti koje daju najbolje rezultate na obučavajućem skupu

Loš način: Za $K = 1$ uvek imamo savršene rezultate na ob. skupu

Obučavajući Skup

Kako određujemo hiper-parametre?

Način #1: Biramo vrednosti koje daju najbolje rezultate na obučavajućem skupu

Loš način: Za $K = 1$ uvek imamo savršene rezultate na ob. skupu

Obučavajući Skup

Način #2: Delimo ob. skup na **obučavajući i test**, biramo vrednosti koje daju najbolje rezultate na test skupu

Obučavajući Skup

test

Kako određujemo hiper-parametre?

Način #1: Biramo vrednosti koje daju najbolje rezultate na obučavajućem skupu

Loš način: Za $K = 1$ uvek imamo savršene rezultate na ob. skupu

Obučavajući Skup

Način #2: Delimo ob. skup na **obučavajući i test**, biramo vrednosti koje daju najbolje rezultate na test skupu

Obučavajući Skup test

Loš način: Nećamo imati nikakvu realnu predstavu o tome kako će se model ponašati na stvarno nepozatim podacima
Da li znate zašto je tako?

Kako određujemo hiper-parametre?

Način #1: Biramo vrednosti koje daju najbolje rezultate na obučavajućem skupu

Loš način: Za $K = 1$ uvek imamo savršene rezultate na ob. skupu

Obučavajući Skup

Način #2: Delimo ob. skup na **obučavajući i test**, biramo vrednosti koje daju najbolje rezultate na test skupu

Loš način: Nećamo imati nikakvu realnu predstavu o tome kako će se model ponašati na stvarno nepozatim podacima
Da li znate zašto je tako?

Obučavajući Skup

test

Način #3: Delimo ob. skup na **obučavajući, validacioni, i test**; biramo vrednosti koje daju najbolje rezultate na validacionom, konačnu evalaciju radimo na test skupu

Sad je sve korektno!

Obučavajući Skup

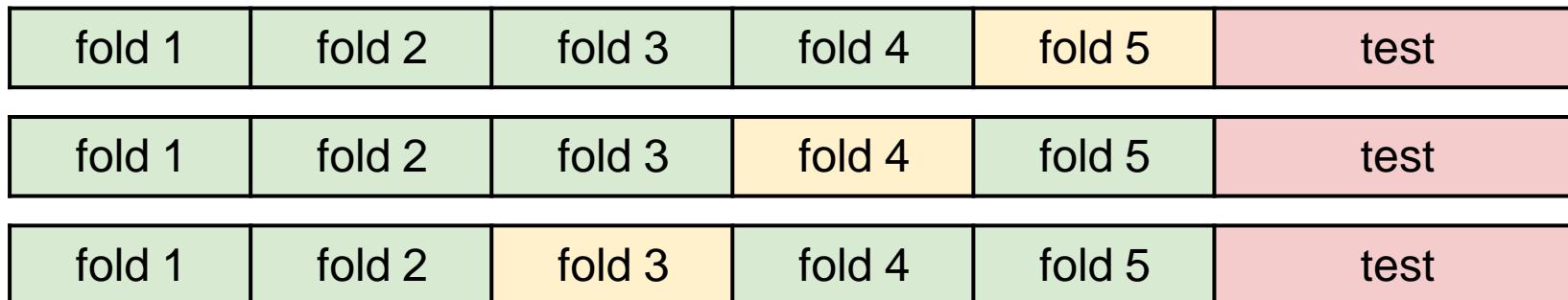
validacioni skup

test

Kako određujemo hiper-parametre?

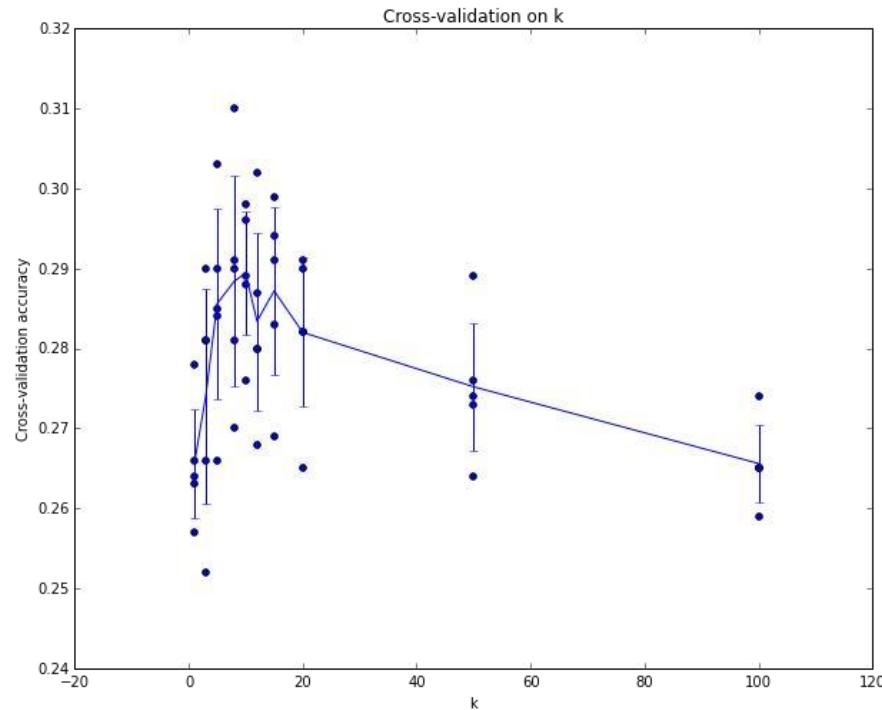
Obučavajući skup

Način #4: Unakrsna Validacija: Delimo podatke u delove (*fold*), svaki deo koristimo kao validacioni skup, pa uzimamo prosek rezultata



Korisno kad imamo manji obučavajući skup, ali retko upotrebljeno u deep learning oblasti

Određivanje hiper-parametara



Koristimo petostruku unakrsnu validaciju za određivanje k .

Svaka tačka: tačnost dobijena unarksnom validacijom za jedno k .

Ako delove u unakrs. val. biramo na slučajan način za isto k možemo dobiti više različitih rezultata. Linija koja ide kroz ceo grafikon povezuje te srednje vrednosti. Vertikalne linije su standardne devijacije

(Izgleda da za $k \approx 7$ imamo najbolje rezultate za ovaj ob. skup)

KNN se u današnje vreme nikada ne koristi za klasifikaciju slika

- Vrlo spor kad se primenjuje na test
- Mere sličnosti koje smo pokazali nisu dovoljno dobre kada se primene na piksele

Original



Neki delovi zacrnjeni



Malo pomerena slika



Malo zatamnjena



(Sve 3 slike su na istoj L2 udaljenosti od prve slike)

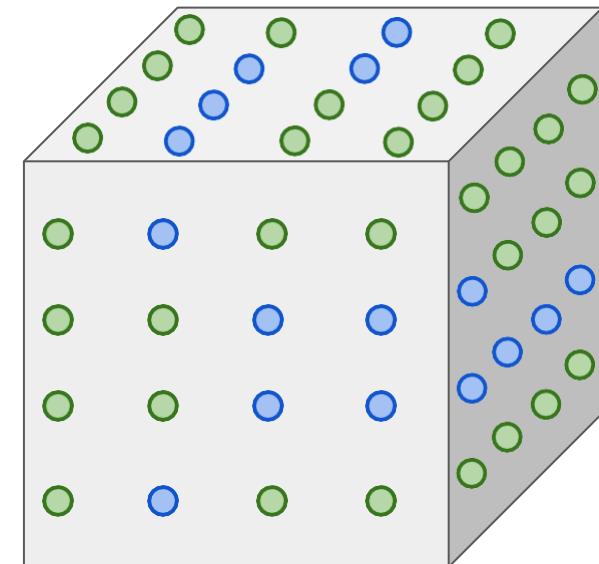
[Original image is
CC0 public domain](#)

KNN se u današnje vreme nikada ne koristi za klasifikaciju slika

- Problem sa visoko-dimenzionim prostorima (*Curse of dimensionality*)

- Broj tačka za „gust“ komšiluk potreban za dobru klasifikaciju raste eksponencijalno sa dimenzionalnošću prostora

Dimenzija = 3
Tačaka = 4^3

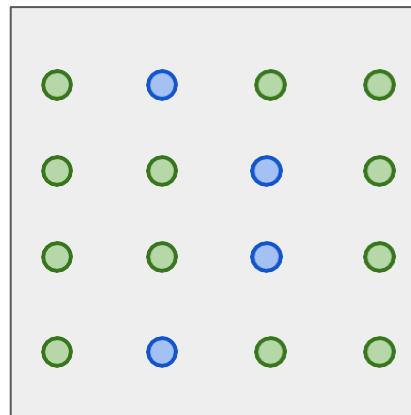


Dimenzija = 1
Tačaka = 4



- boje su različite klase
tačaka.

Dimenzija = 2
Tačaka = 4^2



KNN: Rezime

Kod **klasifikacije slika** počinjemo sa **obučavajućim skupom** slika i njima dodeljenih klasa i cilj nam je da predvidimo klase za **test skup**

Klasifikator **K-Najbližih Komšja** vrši predikciju klasa na osnovu najbližih slika iz obučavajućeg skupa

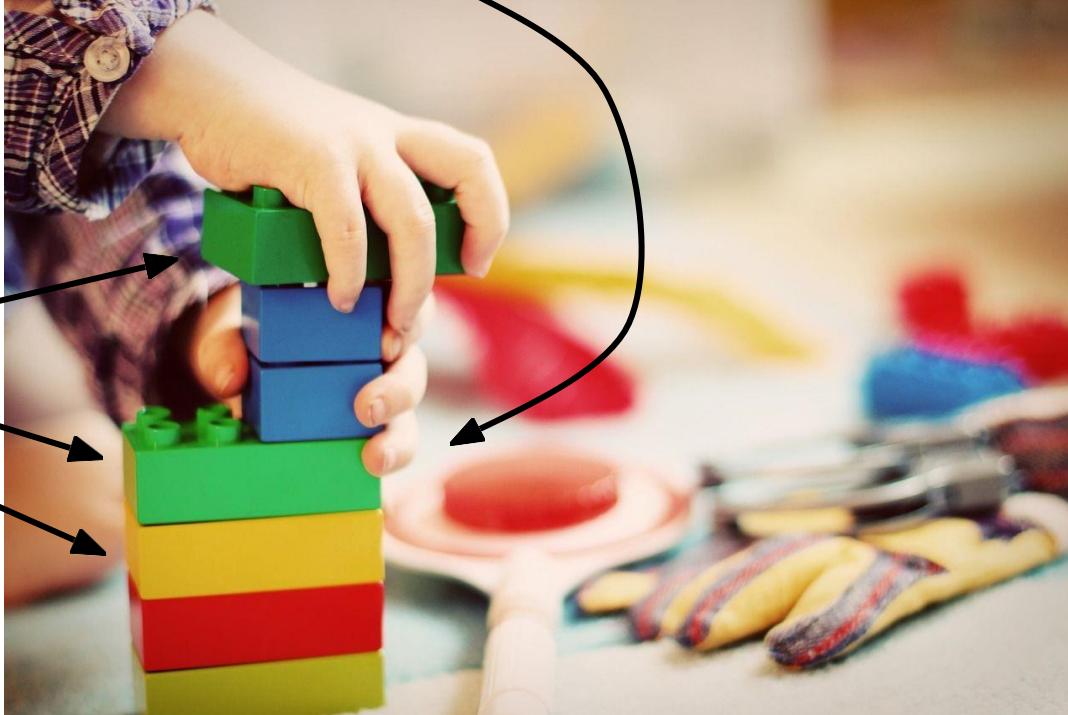
Mera udaljenosti (sličnosti) i K su **hiper-parametri**

Hiper-parametre biramo pomoću **validacionog skupa ili unakrsne validacije**; model primenjujemo na test skup jednom i to na kraju kad smo završili sa određivanjem svih parametra! Bilo kakavo štelovanje modela prema test skupu smatra se varanjem tj. objavljivanje takvih rezultata nije etički.

Linearni Klasifikatori

Neuronska Mreža

Linearni
Klasifikatori



[This image](#) is CC0 1.0 public domain

Two young girls are playing with lego toy. *Boy is doing backflip on wakeboard*



Man in black shirt is playing guitar.

Construction worker in orange safety vest is working on road.

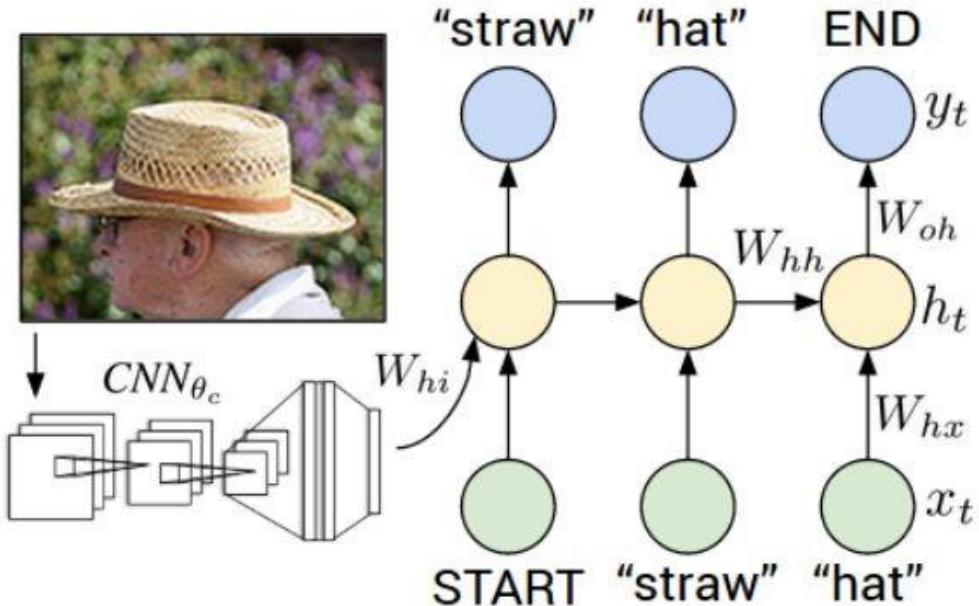
Karpathy and Fei-Fei, "Deep Visual-Semantic Alignments for Generating Image Descriptions", CVPR 2015
Figures copyright IEEE, 2015. Reproduced for educational purposes.

Two young girls are playing with lego toy. *Boy is doing backflip on wakeboard*



Man in black shirt is playing guitar.

Construction worker in orange safety vest is working on road.



Karpathy and Fei-Fei, "Deep Visual-Semantic Alignments for Generating Image Descriptions", CVPR 2015
Figures copyright IEEE, 2015. Reproduced for educational purposes.

CIFAR10

airplane



automobile



bird



cat



deer



dog



frog



horse



ship



truck



50,000 obučavajućih slika
svaka slika je dimenzionalnosti
32x32x3

10,000 test slika

Parametrizovani Pristupi za Klasifikaciju

Slika



Niz $32 \times 32 \times 3$ brojeva
(3072 brojeva ukupno)

$$\xrightarrow{f(x, W)}$$



W

parametri ili
težine

10 brojeva koji
odgovaraju 10 klasa

Parametrizovani Pristupi: Linearni Klasifikator

Slika



$$f(x, W) = Wx$$

Niz $32 \times 32 \times 3$ brojeva
(3072 brojeva ukupno)

$$f(\mathbf{x}, \mathbf{W})$$



\mathbf{W}
parametri ili
težine

10 brojeva koji
odgovaraju 10 klasa

Parametrizovani Pristupi: Linearni Klasifikator

Slika



Niz $32 \times 32 \times 3$ brojeva
(3072 brojeva ukupno)

$$f(x, W) = Wx$$

10×1 10×3072

10 brojeva koji
odgovaraju 10 klasa

W

parametri ili
težine

Parametrizovani Pristupi: Linearni Klasifikator

Slika



Niz $32 \times 32 \times 3$ brojeva
(3072 brojeva ukupno)

$$f(x, W) = Wx + b$$

3072x1
10x1 10x3072 10x1

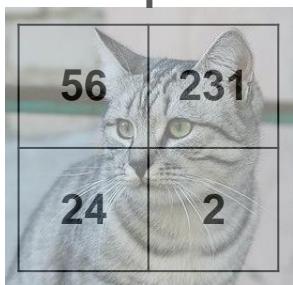
10 brojeva koji odgovaraju 10 klasa

$f(x, W)$

parametri ili težine

Primer sa slikom od 4 piksela i 3 klase (cat/dog/ship)

Smeštamo sliku u niz



Ulagna slika

0.2	-0.5	0.1	2.0
1.5	1.3	2.1	0.0
0	0.25	0.2	-0.3

W



1.1
3.2
-1.2

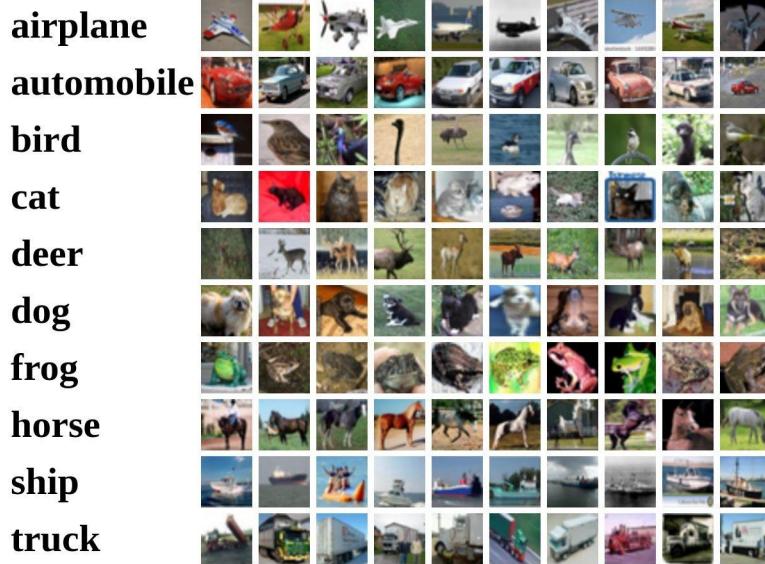
+

-96.8
437.9
61.95

=

Cat skor
Dog skor
Ship skor

Interpretacija linearog klasifikatora



$$f(x, W) = Wx + b$$

Šta tačno radi?

Interpretacija linearog klasifikatora

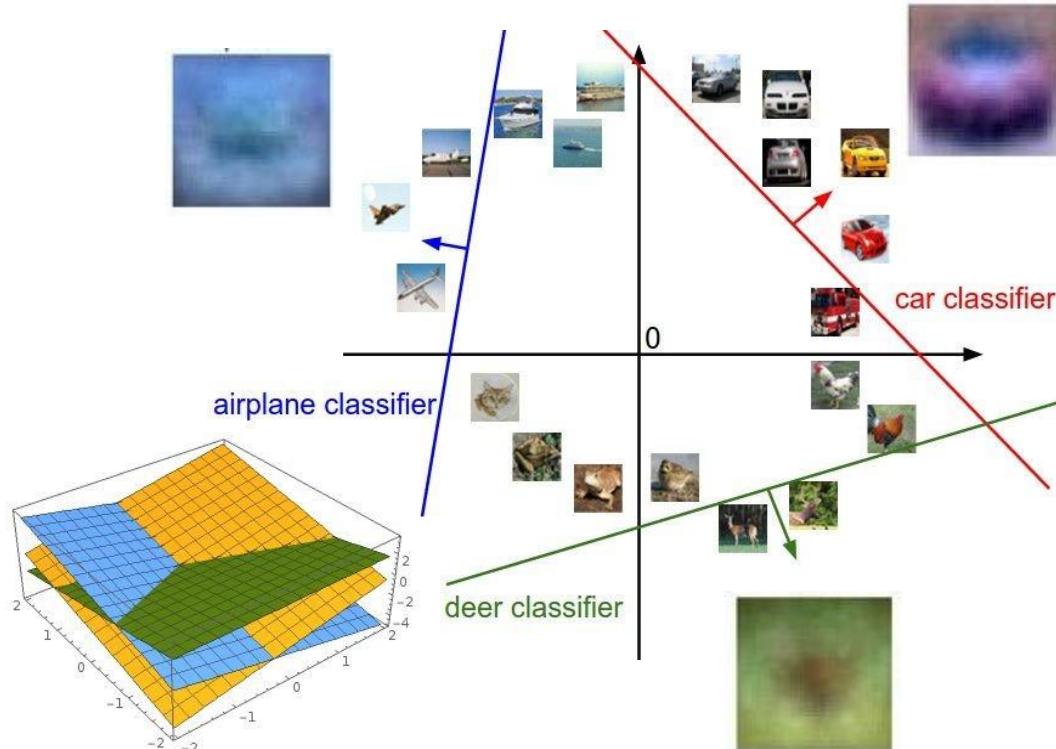
airplane	
automobile	
bird	
cat	
deer	
dog	
frog	
horse	
ship	
truck	

$$f(x, W) = Wx + b$$

Dole su vizualizacije
matrica težina W za sve
klase CIFAR-10:



Interpretacija linearog klasifikatora



$$f(x, W) = Wx + b$$



Niz **32x32x3** brojeva
(3072 brojeva ukupno)

Plot created using [WolframCloud](#)

[Cat image](#) by [Nikita](#) is licensed under [CC-BY 2.0](#)

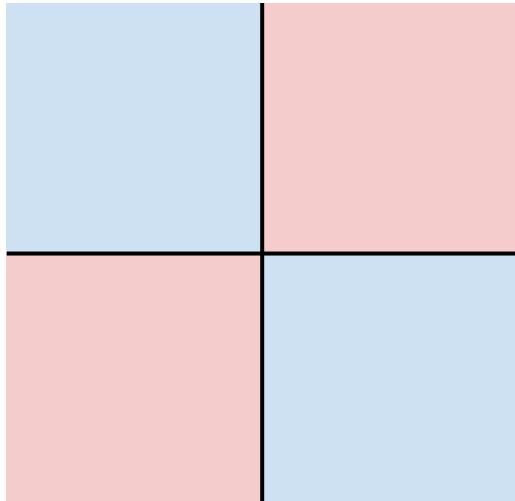
Problemi koji su teški za linearni klasifikator

Klasa 1:

broj piksela > 0 neparan

Klasa 2:

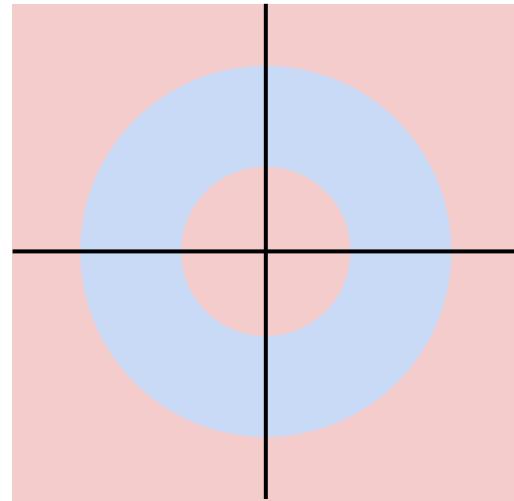
broj piksela > 0 paran



Klasa 1:

$1 \leq L_2 \text{ mera} \leq 2$

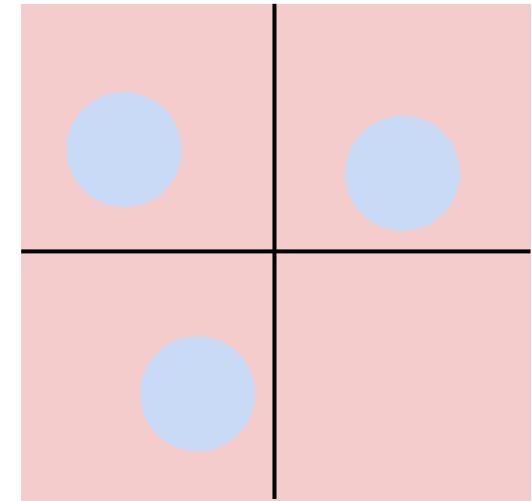
Klasa 2: Sve
ostalo



Klasa 1: 3

različita kruga

Class 2: Sve
ostalo



Do sada: Definisali smo (linearu) skor funkciju $f(x,W) = Wx + b$

Primeri
skorova za 10
klasa za 3
slie i neko W :

Kako da znamo
da li je ovo W
dobar ili loš
klasifikator?



airplane	-3.45	-0.51	3.42
automobile	-8.87	6.04	4.64
bird	0.09	5.31	2.65
cat	2.9	-4.22	5.1
deer	4.48	-4.19	2.64
dog	8.02	3.58	5.55
frog	3.78	4.49	-4.34
horse	1.06	-4.37	-1.5
ship	-0.36	-2.09	-4.79
truck	-0.72	-2.93	6.14

[Cat image](#) by [Nikita](#) is licensed under [CC-BY 2.0](#)

[Car image](#) is [CC0 1.0](#) public domain

[Frog image](#) is in the public domain

U nastavku kursa:

- Funkcija greške
- Optimizacija
- Neuronske i kovolutivne mreže!

$$f(x, W) = Wx + b$$

(kvantifikujemo šta je tačno “dobro” W)

(krećemo od slučajno odabranog W i tražimo W koje mimimizuje funkciju greške)

(malo ćemo menjati sam oblik funkcije f)

Funkcija Greške i Optimizacija

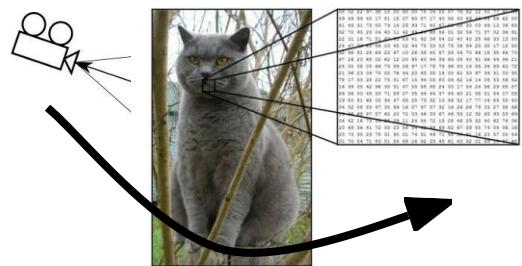
Predavač: Aleksandar Kovačević

Slajdovi preuzeti sa CS 231n, Stanford

<http://cs231n.stanford.edu/>

Sa prošlog predavanja ... Izazovi kod klasifikacije slika

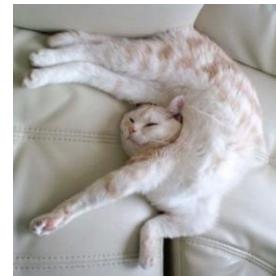
Pozcija kamere



Osvetljenje



Deformacija



Okluzija



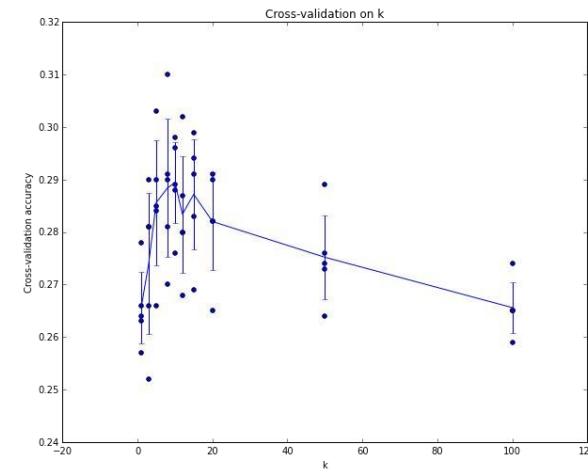
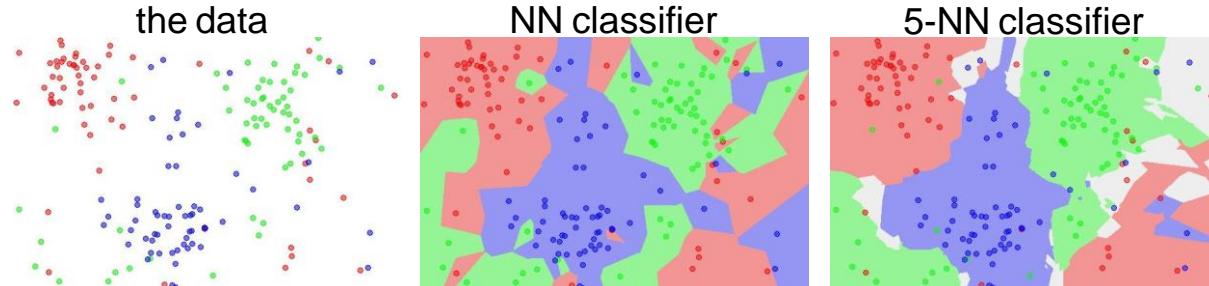
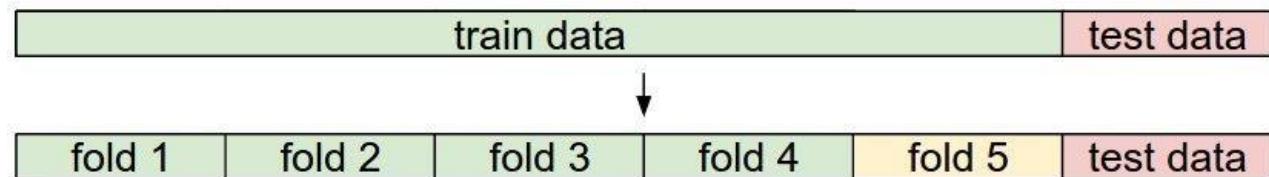
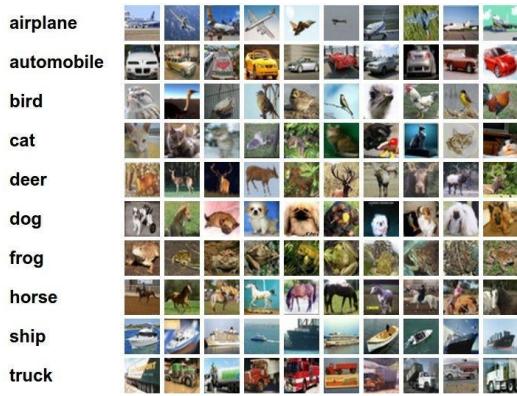
Pretrpana pozadina



Varijabilnost unutra klase



Sa prošlog predavanja... nadgledano učenje, kNN



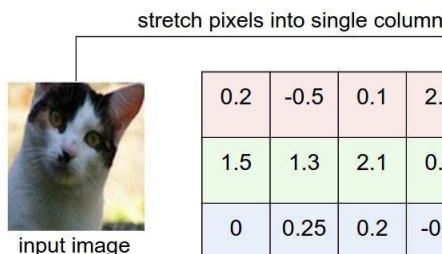
Sa prošlog predavanja ... Linearni klasifikator



[32x32x3]

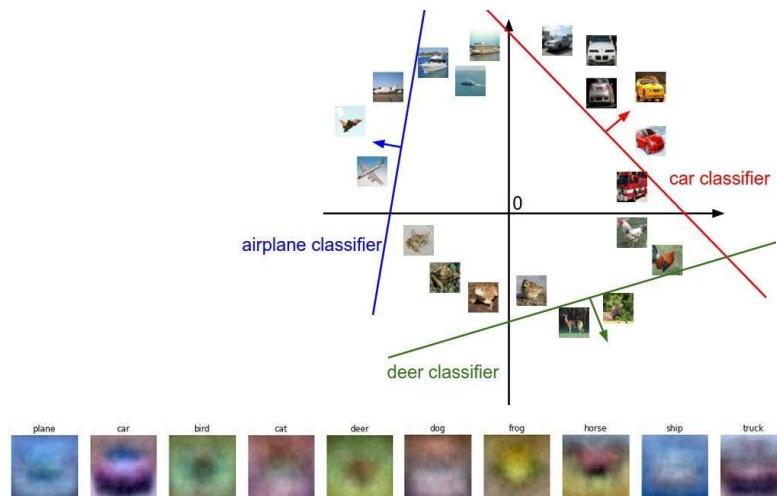
niz od 3072 brojeava

slika parametri
 $f(\mathbf{x}, \mathbf{W})$



56	1.1	-96.8	cat score
231	3.2	437.9	dog score
24	-1.2	61.95	ship score
2			

10 brojeva koji su skorovi klasa



Sa prošlog predavanja ... Danas: Funkcija greške /Optimizacija



airplane	-3.45	-0.51	3.42
automobile	-8.87	6.04	4.64
bird	0.09	5.31	2.65
cat	2.9	-4.22	5.1
deer	4.48	-4.19	2.64
dog	8.02	3.58	5.55
frog	3.78	4.49	-4.34
horse	1.06	-4.37	-1.5
ship	-0.36	-2.09	-4.79
truck	-0.72	-2.93	6.14

TODO:

1. Definisanje funkcije greške (**loss function**) koja kvantifikuje kvalitet skorova za slike iz obučavajućeg skupa
2. Nađemo način da automatski odredimo W i b tako da minimizujemo funkciju greške.
(optimizacija)

Na primer: 3 slike u obučavajućem skupu, 3 klase.

Za neko W skorovi $f(x, W) = Wx$ su:



cat	3.2	1.3	2.2
car	5.1	4.9	2.5
frog	-1.7	2.0	-3.1

Na primer: 3 slike u obučavajućem skupu, 3 klase.
Za neko W skorovi $f(x, W) = Wx$ su:



cat	3.2	1.3	2.2
car	5.1	4.9	2.5
frog	-1.7	2.0	-3.1

Fukcija greške modela SVM za više klasa:

Za primer (x_i, y_i)

gde je x_i slika

gde je y_i klasa (ceo broj),

Gde se skor računa kao: $s = f(x_i, W)$

funkcija greške za SVM je oblika:

$$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$

Na primer: 3 slike u obučavajućem skupu, 3 klase.
Za neko W skorovi $f(x, W) = Wx$ su:



cat	3.2	1.3	2.2
car	5.1	4.9	2.5
frog	-1.7	2.0	-3.1

Fukcija greške modela SVM za više klasa:

Za primer (x_i, y_i)

gde je x_i slika

gde je y_i klasa (ceo broj),

Gde se skor računa kao: $s = f(x_i, W)$

funkcija greške za SVM je oblika:

$$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$

$$s_j - s_{y_i} + 1 \leq 0$$

$$-s_{y_i} \leq -s_j - 1$$

$$s_{y_i} \geq s_j + 1$$

Na primer: 3 slike u obučavajućem skupu, 3 klase.
Za neko W skorovi $f(x, W) = Wx$ su:



cat	3.2	1.3	2.2
car	5.1	4.9	2.5
frog	-1.7	2.0	-3.1
Greške:			

Fukcija greške modela SVM za više klasa:

Za primer (x_i, y_i)

gde je x_i slika

gde je y_i klasa (ceo broj),

Gde se skor računa kao: $s = f(x_i, W)$

funkcija greške za SVM je oblika:

$$\begin{aligned}L_i &= \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1) \\&= \max(0, 5.1 - 3.2 + 1) \\&\quad + \max(0, -1.7 - 3.2 + 1) \\&= \max(0, 2.9) + \max(0, -3.9) \\&= 2.9 + 0 \\&= 2.9\end{aligned}$$

Na primer: 3 slike u obučavajućem skupu, 3 klase.
Za neko W skorovi $f(x, W) = Wx$ su:



cat	3.2	1.3	2.2
car	5.1	4.9	2.5
frog	-1.7	2.0	-3.1
Greške:	2.9	0	

Fukcija greške modela SVM za više klasa:

Za primer (x_i, y_i)

gde je x_i slika

gde je y_i klasa (ceo broj),

Gde se skor računa kao: $s = f(x_i, W)$

funkcija greške za SVM je oblika:

$$\begin{aligned}
 L_i &= \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1) \\
 &= \max(0, 1.3 - 4.9 + 1) \\
 &\quad + \max(0, 2.0 - 4.9 + 1) \\
 &= \max(0, -2.6) + \max(0, -1.9) \\
 &= 0 + 0 \\
 &= 0
 \end{aligned}$$

Na primer: 3 slike u obučavajućem skupu, 3 klase.
Za neko W skorovi $f(x, W) = Wx$ su:



cat	3.2	1.3	2.2
car	5.1	4.9	2.5
frog	-1.7	2.0	-3.1
Greške:	2.9	0	10.9

Fukcija greške modela SVM za više klasa:

Za primer (x_i, y_i)

gde je x_i slika

gde je y_i klasa (ceo broj),

Gde se skor računa kao: $s = f(x_i, W)$

funkcija greške za SVM je oblika:

$$\begin{aligned}
 L_i &= \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1) \\
 &= \max(0, 2.2 - (-3.1) + 1) \\
 &\quad + \max(0, 2.5 - (-3.1) + 1) \\
 &= \max(0, 5.3) + \max(0, 5.6) \\
 &= 5.3 + 5.6 \\
 &= 10.9
 \end{aligned}$$

Na primer: 3 slike u obučavajućem skupu, 3 klase.
Za neko W skorovi $f(x, W) = Wx$ su:



cat	3.2	1.3	2.2
car	5.1	4.9	2.5
frog	-1.7	2.0	-3.1
Greške:	2.9	0	10.9

Fukcija greške modela SVM za više klasa:

Za primer (x_i, y_i)

gde je x_i slika

gde je y_i klasa (ceo broj),

Gde se skor računa kao: $s = f(x_i, W)$
funkcija greške za SVM je oblika:

$$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$

Greška za sve tri slike je prosek
grešaka:

$$L = \frac{1}{N} \sum_{i=1}^N L_i$$

$$\begin{aligned} L &= (2.9 + 0 + 10.9)/3 \\ &= 4.6 \end{aligned}$$

Na primer: 3 slike u obučavajućem skupu, 3 klase.
Za neko W skorovi $f(x, W) = Wx$ su:



cat	3.2	1.3	2.2
car	5.1	4.9	2.5
frog	-1.7	2.0	-3.1
Greške:	2.9	0	10.9

Fukcija greške modela SVM za više klasa:

Za primer (x_i, y_i)

gde je x_i slika

gde je y_i klasa (ceo broj),

Gde se skor računa kao: $s = f(x_i, W)$
funkcija greške za SVM je oblika:

$$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$

Šta bi bilo ako bi L_i računali za sve klase tj. i za $j=y_i$?

Na primer: 3 slike u obučavajućem skupu, 3 klase.
Za neko W skorovi $f(x, W) = Wx$ su:



cat	3.2	1.3	2.2
car	5.1	4.9	2.5
frog	-1.7	2.0	-3.1
Greške:	2.9	0	10.9

Fukcija greške modela SVM za više klasa:

Za primer (x_i, y_i)

gde je x_i slika

gde je y_i klasa (ceo broj),

Gde se skor računa kao: $s = f(x_i, W)$
funkcija greške za SVM je oblika:

$$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$

Šta bi bilo ako bi
korsitili:

$$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)^2$$

Na primer: 3 slike u obučavajućem skupu, 3 klase.
Za neko W skorovi $f(x, W) = Wx$ su:



cat	3.2	1.3	2.2
car	5.1	4.9	2.5
frog	-1.7	2.0	-3.1
Greške:	2.9	0	10.9

Fukcija greške modela SVM za više klasa:

Za primer (x_i, y_i)

gde je x_i slika

gde je y_i klasa (ceo broj),

Gde se skor računa kao: $s = f(x_i, W)$
funkcija greške za SVM je oblika:

$$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$

Koje su min i max vrednosti za funkciju greške?

Na primer: 3 slike u obučavajućem skupu, 3 klase.
Za neko W skorovi $f(x, W) = Wx$ su:



cat	3.2	1.3	2.2
car	5.1	4.9	2.5
frog	-1.7	2.0	-3.1
Greške:	2.9	0	10.9

Fukcija greške modela SVM za više klasa:

Za primer (x_i, y_i)

gde je x_i slika

gde je y_i klasa (ceo broj),

Gde se skor računa kao: $s = f(x_i, W)$
funkcija greške za SVM je oblika:

$$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$

Obično se vrednosti W inicijalizuju na jako male brojeve ~ 0 .
Koje su tada vrednosti funkcije greške?

numpy kod za SVM primer:

$$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$

```
def L_i_vectorized(x, y, W):
    scores = W.dot(x)
    margins = np.maximum(0, scores - scores[y] + 1)
    margins[y] = 0
    loss_i = np.sum(margins)
    return loss_i
```

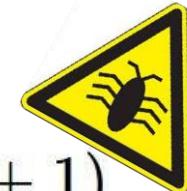
$$f(x, W) = Wx$$

$$L = \frac{1}{N} \sum_{i=1}^N \sum_{j \neq y_i} \max(0, f(x_i; W)_j - f(x_i; W)_{y_i} + 1)$$

Naša funkcija greške ima bag:

$$f(x, W) = Wx$$

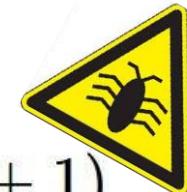
$$L = \frac{1}{N} \sum_{i=1}^N \sum_{j \neq y_i} \max(0, f(x_i; W)_j - f(x_i; W)_{y_i} + 1)$$



Naša funkcija greške ima bag:

$$f(x, W) = Wx$$

$$L = \frac{1}{N} \sum_{i=1}^N \sum_{j \neq y_i} \max(0, f(x_i; W)_j - f(x_i; W)_{y_i} + 1)$$



Recimo da smo pronašli W tako da je $L = 0$.
Da li je to W jedinstveno?

Na primer: 3 slike u obučavajućem skupu, 3 klase.
 Za neko W skorovi $f(x, W) = Wx$ su:



cat	3.2	1.3	2.2
car	5.1	4.9	2.5
frog	-1.7	2.0	-3.1
Greške:	2.9	0	

$$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$

Ranije:

$$\begin{aligned}
 &= \max(0, 1.3 - 4.9 + 1) \\
 &\quad + \max(0, 2.0 - 4.9 + 1) \\
 &= \max(0, -2.6) + \max(0, -1.9) \\
 &= 0 + 0 \\
 &= 0
 \end{aligned}$$

Sa duplo većim W :

$$\begin{aligned}
 &= \max(0, 2.6 - 9.8 + 1) \\
 &\quad + \max(0, 4.0 - 9.8 + 1) \\
 &= \max(0, -6.2) + \max(0, -4.8) \\
 &= 0 + 0 \\
 &= 0
 \end{aligned}$$

Regularizacija težina

lambda = jačina regularizacije
(hiper-parametar)

$$L = \frac{1}{N} \sum_{i=1}^N \sum_{j \neq y_i} \max(0, f(x_i; W)_j - f(x_i; W)_{y_i} + 1) + \boxed{\lambda R(W)}$$

Tipično se koristi:

L2 regularizacija

$$R(W) = \sum_k \sum_l W_{k,l}^2$$

L1 regularizacija

$$R(W) = \sum_k \sum_l |W_{k,l}|$$

Elastic net (L1 + L2)

$$R(W) = \sum_k \sum_l \beta W_{k,l}^2 + |W_{k,l}|$$

Dropout (kasnije tokom kursa)

L2 regularizacija: motivacija

$$x = [1, 1, 1, 1]$$

$$w_1 = [1, 0, 0, 0]$$

$$w_2 = [0.25, 0.25, 0.25, 0.25]$$

$$w_1^T x = w_2^T x = 1$$

Softmax Klasifikator (Multinominalna Logistička Regresija)



cat	3.2
car	5.1
frog	-1.7

Softmax Klasifikator (Multinominalna Logistička Regresija)



skorovi = nenormalizovane log verovatnoće klase
(nenormalizovane znači da se ne sabiraju na 1)

$$s = f(x_i; W)$$

cat	3.2
car	5.1
frog	-1.7

Softmax Klasifikator (Multinomialna Logistička Regresija)



skorovi = nenormalizovane log verovatnoće klase

$$P(Y = k | X = x_i) = \frac{e^{s_k}}{\sum_j e^{s_j}} \quad s = f(x_i; W)$$

cat	3.2
car	5.1
frog	-1.7

Softmax Klasifikator (Multinomialna Logistička Regresija)



skorovi = nenormalizovane log verovatnoće klase

$$P(Y = k | X = x_i) = \frac{e^{s_k}}{\sum_j e^{s_j}} \quad s = f(x_i; W)$$

cat	3.2
car	5.1
frog	-1.7

Softmax funkcija

Softmax Klasifikator (Multinominalna Logistička Regresija)



cat	3.2
car	5.1
frog	-1.7

skorovi = nenormalizovane log verovatnoće klasa

$$P(Y = k | X = x_i) = \frac{e^{s_k}}{\sum_j e^{s_j}} \quad s = f(x_i; W)$$

Maksimizujemo log verovatnost (log likelihood) – obično se minimizuje negativna log verovatnost:

$$L_i = -\log P(Y = y_i | X = x_i)$$

Softmax Klasifikator (Multinominalna Logistička Regresija)



cat	3.2
car	5.1
frog	-1.7

skorovi = nenormalizovane log verovatnoće klasa

$$P(Y = k | X = x_i) = \frac{e^{s_k}}{\sum_j e^{s_j}} \quad s = f(x_i; W)$$

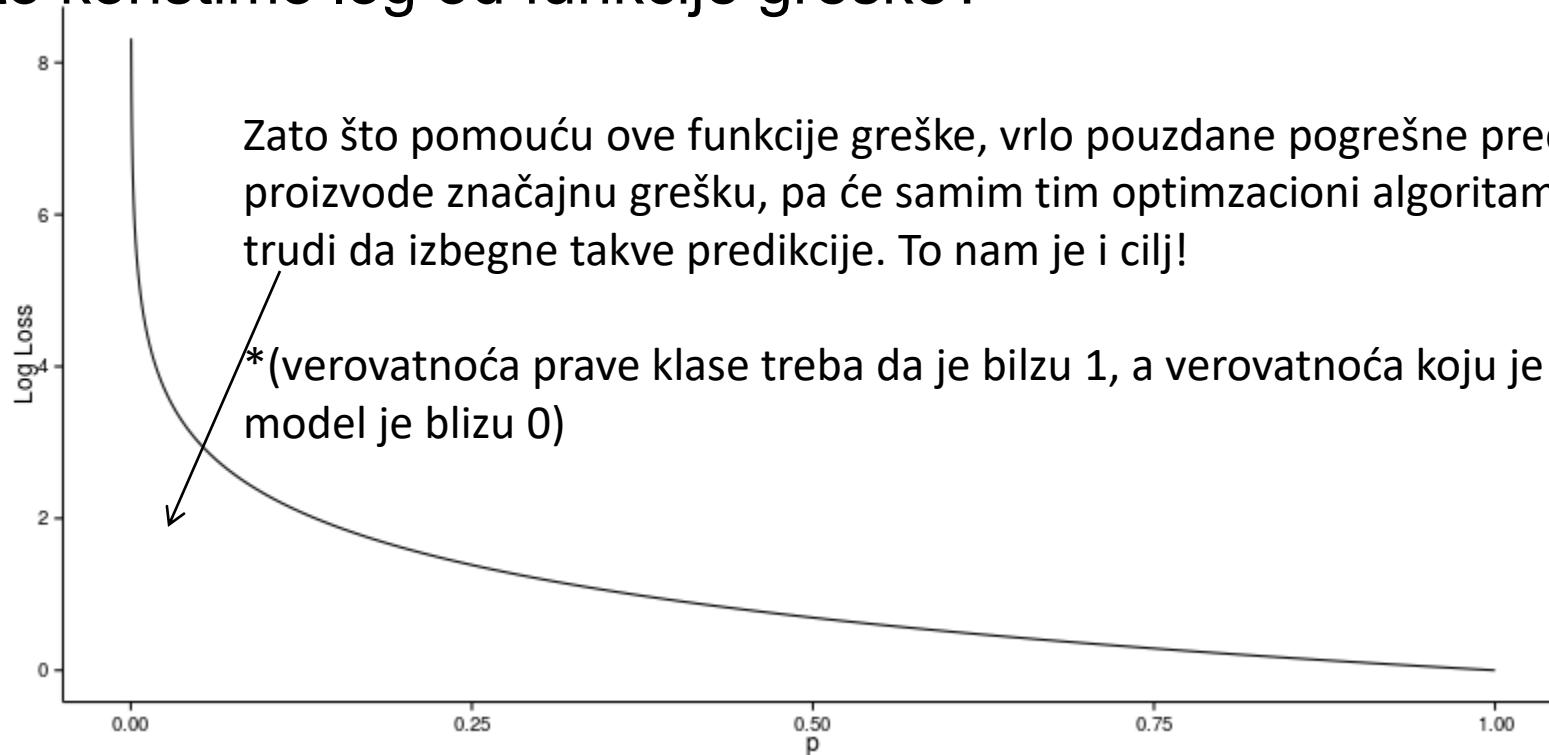
Maksimizujemo log verovatnost (log likelihood) – obično se minimizuje negativna log verovatnost:

$$L_i = -\log P(Y = y_i | X = x_i)$$

$$L_i = -\log\left(\frac{e^{s_{y_i}}}{\sum_j e^{s_j}}\right)$$

Softmax Klasifikator (Multinominalna Logistička Regresija)

Zašto koristimo log od funkcije greške?



Softmax Klasifikator (Multinominalna Logistička Regresija)



$$L_i = -\log\left(\frac{e^{s_{y_i}}}{\sum_j e^{s_j}}\right)$$

cat

3.2

car

5.1

frog

-1.7

nenormalizovane log verovatnoće

Softmax Klasifikator (Multinominalna Logistička Regresija)



$$L_i = -\log\left(\frac{e^{s_{y_i}}}{\sum_j e^{s_j}}\right)$$

nenormalizovane verovatnoće

cat

car

frog

	3.2	24.5
	5.1	164.0
	-1.7	0.18

exp

nenormalizovane log verovatnoće

Softmax Klasifikator (Multinominalna Logistička Regresija)



$$L_i = -\log\left(\frac{e^{s_{y_i}}}{\sum_j e^{s_j}}\right)$$

nenormalizovane verovatnoće

cat
car
frog

3.2
5.1
-1.7

exp

24.5
164.0
0.18

normalizacija

0.13
0.87
0.00

nemormalizovane log verovatnoće

verovatnoće

Softmax Klasifikator (Multinominalna Logistička Regresija)



$$L_i = -\log\left(\frac{e^{s_{y_i}}}{\sum_j e^{s_j}}\right)$$

nenormalizovane verovatnoće

cat
car
frog

3.2
5.1
-1.7

exp

24.5
164.0
0.18

normalizacija

0.13
0.87
0.00

→

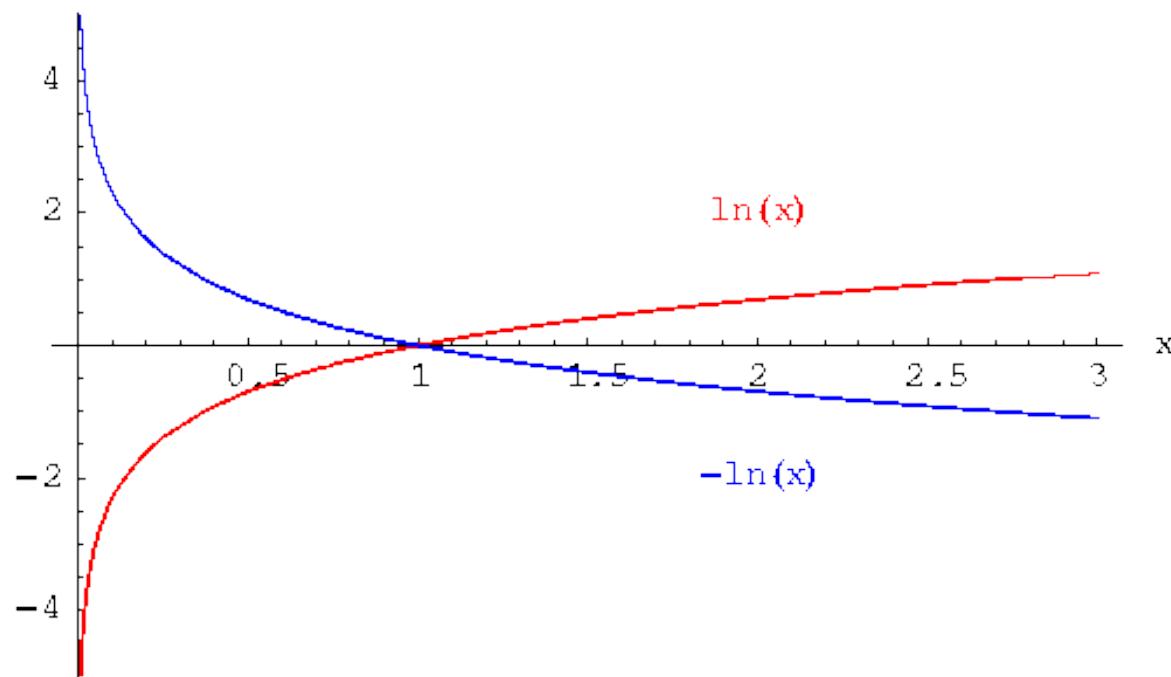
$$\begin{aligned} L_i &= -\log(0.13) \\ &= 0.89 \end{aligned}$$

nemormalizovane log verovatnoće

verovatnoće

Softmax Klasifikator (Multinominalna Logistička Regresija)

log vs. -log



Softmax Klasifikator (Multinominalna Logistička Regresija)



$$L_i = -\log\left(\frac{e^{s_{y_i}}}{\sum_j e^{s_j}}\right)$$

Koje su moguće min i max vrednosti za L_i?

nenormalizovane verovatnoće

cat
car
frog

3.2
5.1
-1.7

exp

24.5
164.0
0.18

normalizacija

0.13
0.87
0.00

$$\rightarrow L_i = -\log(0.13) = 0.89$$

nemormalizovane log verovatnoće

verovatnoće

Softmax Klasifikator (Multinominalna Logistička Regresija)



$$L_i = -\log\left(\frac{e^{s_{y_i}}}{\sum_j e^{s_j}}\right)$$

nemormalizovane verovatnoće

Obično se vrednosti W inicijalizuju na jako male brojeve ≈ 0 .
Koje su tada vrednosti funkcije greške?

cat
car
frog

~ 0
~ 0
~ 0

exp

~ 1
~ 1
~ 1

normalizacija

$\sim 1/(1+1+1)$
$\sim 1/(1+1+1)$
$\sim 1/(1+1+1)$

nemormalizovane log verovatnoće

verovatnoće

$$L_i = -\log(\sim 1/3)$$

hinge loss (SVM)

matrix multiply + bias offset

0.01	-0.05	0.1	0.05
0.7	0.2	0.05	0.16
0.0	-0.45	-0.2	0.03

W

-15	0.0
22	0.2
-44	-0.3
56	

x_i

y_i 2

-15	0.0
22	0.2
-44	-0.3
56	

b

-2.85
0.86
0.28

$$\begin{aligned} & \max(0, -2.85 - 0.28 + 1) + \\ & \max(0, 0.86 - 0.28 + 1) \\ & = \\ & \textcolor{red}{1.58} \end{aligned}$$

cross-entropy loss (Softmax)

-2.85	0.058	0.016
0.86	2.36	0.631
0.28	1.32	0.353

\exp

$\xrightarrow{\text{normalize}}$
(to sum to one)

$$\begin{aligned} & -\log(0.353) \\ & = \\ & \textcolor{green}{0.452} \end{aligned}$$

Softmax vs. SVM

$$L_i = -\log\left(\frac{e^{s_{y_i}}}{\sum_j e^{s_j}}\right)$$

$$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$

Softmax vs. SVM

$$L_i = -\log\left(\frac{e^{s_{y_i}}}{\sum_j e^{s_j}}\right)$$

$$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$

recimo da imamo skorove:

[10, -2, 3]

[10, 9, 9]

[10, -100, -100]

$$y_i = 0$$

Šta se dešava sa obe funkcije greške ako uzmemos jednu tačku i malo je pomjeramo u prostoru?

Optimizacija

Rezime

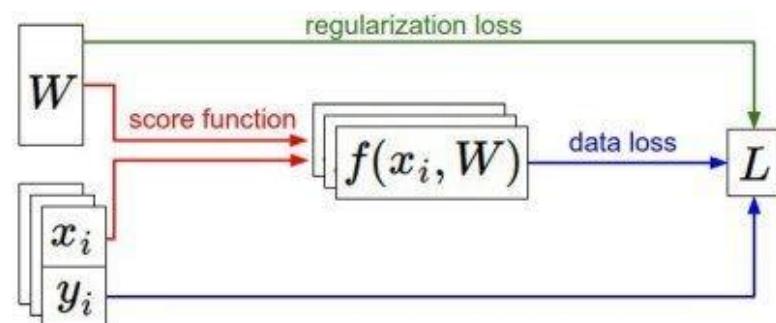
- Imamo skup parova (x, y)
- Imamo **skor funkciju**:
- Imamo **funkciju greške**:

$$s = f(x; W) \stackrel{\text{npr.}}{=} Wx$$

$$L_i = -\log\left(\frac{e^{sy_i}}{\sum_j e^{sj}}\right) \quad \text{Softmax}$$

$$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1) \quad \text{SVM}$$

$$L = \frac{1}{N} \sum_{i=1}^N L_i + R(W) \quad \text{Greška + Regularizacija}$$



Optimizacija - Strategija #1: Random pretraga

```
# assume X_train is the data where each column is an example (e.g. 3073 x 50,000)
# assume Y_train are the labels (e.g. 1D array of 50,000)
# assume the function L evaluates the loss function

bestloss = float("inf") # Python assigns the highest possible float value
for num in xrange(1000):
    W = np.random.randn(10, 3073) * 0.0001 # generate random parameters
    loss = L(X_train, Y_train, W) # get the loss over the entire training set
    if loss < bestloss: # keep track of the best solution
        bestloss = loss
        bestW = W
    print 'in attempt %d the loss was %f, best %f' % (num, loss, bestloss)

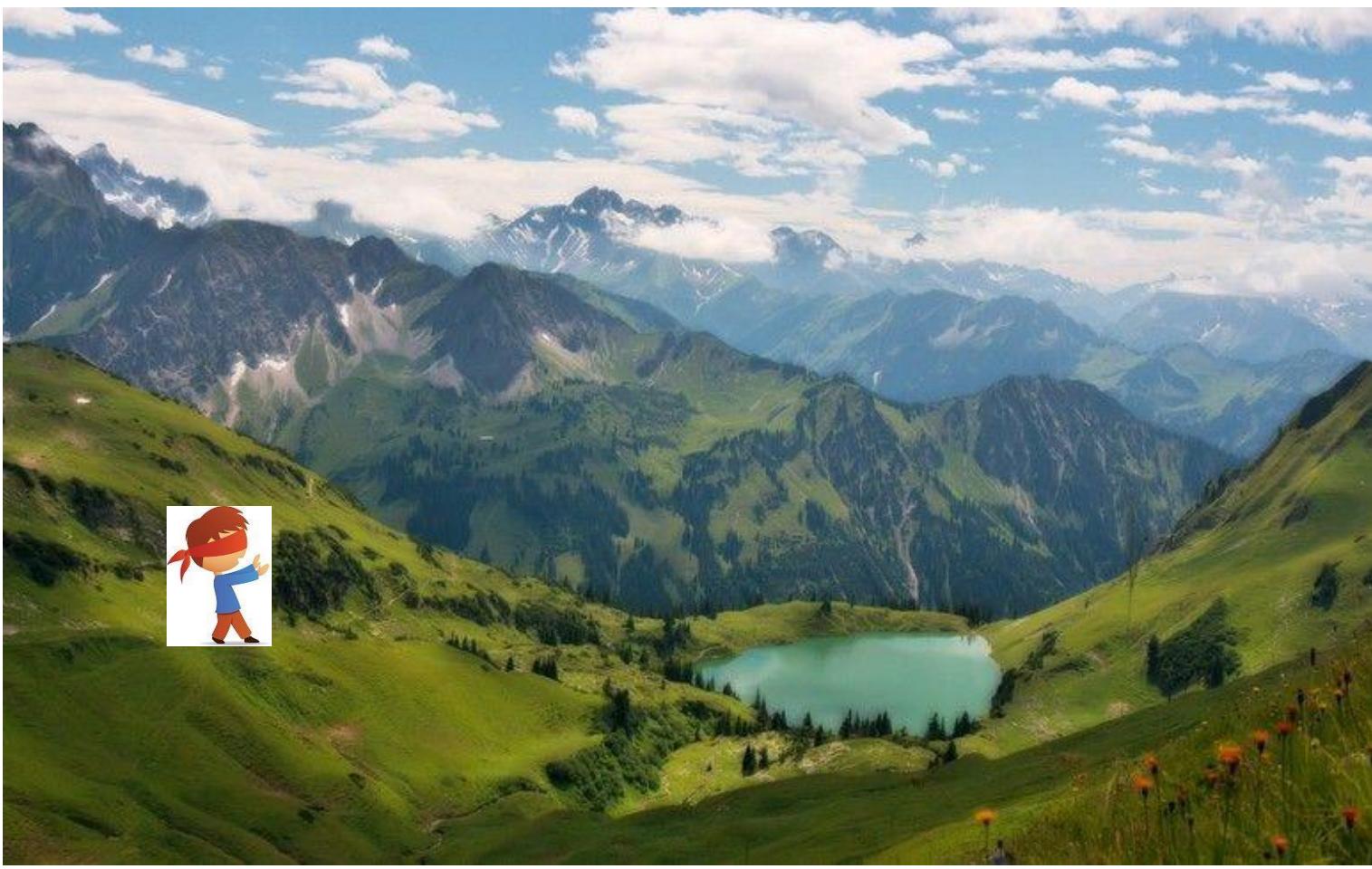
# prints:
# in attempt 0 the loss was 9.401632, best 9.401632
# in attempt 1 the loss was 8.959668, best 8.959668
# in attempt 2 the loss was 9.044034, best 8.959668
# in attempt 3 the loss was 9.278948, best 8.959668
# in attempt 4 the loss was 8.857370, best 8.857370
# in attempt 5 the loss was 8.943151, best 8.857370
# in attempt 6 the loss was 8.605604, best 8.605604
# ... (truncated: continues for 1000 lines)
```

Šta dobijamo na test skupu...

```
# Assume X_test is [3073 x 10000], Y_test [10000 x 1]
scores = Wbest.dot(Xte_cols) # 10 x 10000, the class scores for all test examples
# find the index with max score in each column (the predicted class)
Yte_predict = np.argmax(scores, axis = 0)
# and calculate accuracy (fraction of predictions that are correct)
np.mean(Yte_predict == Yte)
# returns 0.1555
```

15.5% tačnost! nije loše!
(State-of-the-Art, SOTA je ~95%)





Strategija #2: Pratimo nagib (*slope*)

Kod funkcije jedne promenljive, izvod funkcije dat je sa:

$$\frac{df(x)}{dx} = \lim_{h \rightarrow 0} \frac{f(x + h) - f(x)}{h}$$

Kad imamo više promenljivih (atributa), **gradijent** je vektor parcijalnih izvoda po svim atributima.

W:

[0.34,
-1.11,
0.78,
0.12,
0.55,
2.81,
-3.1,
-1.5,
0.33,...]

greška 1.25347

gradijent dW:

[?,
?,
?,
?,
?,
?,
?,
?,
?,
?,...]

W:

[0.34,
-1.11,
0.78,
0.12,
0.55,
2.81,
-3.1,
-1.5,
0.33,...]

greška 1.25347

W + h (za prvi atribut):

[0.34 + 0.0001,
-1.11,
0.78,
0.12,
0.55,
2.81,
-3.1,
-1.5,
0.33,...]

greška 1.25322

gradijent dW:

[?,
,
,
,
,
,
,
,
,...]

W:

[0.34,
-1.11,
0.78,
0.12,
0.55,
2.81,
-3.1,
-1.5,
0.33,...]
greška 1.25347

W + h (za prvi atribut):

[0.34 + **0.0001**,
-1.11,
0.78,
0.12,
0.55,
2.81,
-3.1,
-1.5,
0.33,...]
greška 1.25322

gradijent dW:

[-2.5,

?,
?,

$$\frac{(1.25322 - 1.25347)}{0.0001} = -2.5$$

$$\frac{df(x)}{dx} = \lim_{h \rightarrow 0} \frac{f(x + h) - f(x)}{h}$$

?,

?,...]

W:

[0.34,
-1.11,
0.78,
0.12,
0.55,
2.81,
-3.1,
-1.5,
0.33,...]

greška 1.25347

W + h (drugi atribut):

[0.34,
-1.11 + **0.0001**,
0.78,
0.12,
0.55,
2.81,
-3.1,
-1.5,
0.33,...]

greška 1.25353

gradijent dW:

[-2.5,
?,
?,
?,
?,
?,
?,
?,
?,
?,...]

W:

[0.34,
-1.11,
0.78,
0.12,
0.55,
2.81,
-3.1,
-1.5,
0.33,...]
greška 1.25347

W + h (drugi atribut):

[0.34,
-1.11 + **0.0001**,
0.78,
0.12,
0.55,
2.81,
-3.1,
-1.5,
0.33,...]
greška 1.25353

gradijent dW:

[-2.5,
0.6,
?,
?]

$$\frac{(1.25353 - 1.25347)}{0.0001} = 0.6$$

$$\frac{df(x)}{dx} = \lim_{h \rightarrow 0} \frac{f(x + h) - f(x)}{h}$$

?,...]

W:

[0.34,
-1.11,
0.78,
0.12,
0.55,
2.81,
-3.1,
-1.5,
0.33,...]

greška 1.25347

W + h (treći atribut):

[0.34,
-1.11,
0.78 + **0.0001**,
0.12,
0.55,
2.81,
-3.1,
-1.5,
0.33,...]

greška 1.25347

gradijent dW:

[-2.5,
0.6,
?,
?,
?,
?,
?,
?,
?,
?,...]

W:

[0.34,
-1.11,
0.78,
0.12,
0.55,
2.81,
-3.1,
-1.5,
0.33,...]

greška 1.25347

W + h (treći atribut):

[0.34,
-1.11,
0.78 + **0.0001**,
0.12,
0.55,
2.81,
-3.1,
-1.5,
0.33,...]

greška 1.25347

gradijent dW:

[-2.5,
0.6,
0,
?,
?]

$$\frac{(1.25347 - 1.25347)}{0.0001} = 0$$

$$\frac{df(x)}{dx} = \lim_{h \rightarrow 0} \frac{f(x + h) - f(x)}{h}$$

?, ...]

Numerički izvod nije dobar pristup ovde. Greška je funkcija od W , pa možemo da koristimo analitički izvod.

$$L = \frac{1}{N} \sum_{i=1}^N L_i + \sum_k W_k^2$$

$$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$

$$s = f(x; W) = Wx$$

treba $\nabla_W L$
nam

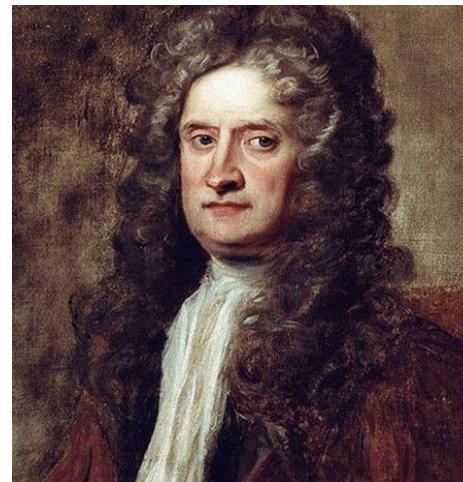
Numerički izvod nije dobar pristup ovde. Greška je funkcija od W , pa možemo da koristimo analitički izvod.

$$L = \frac{1}{N} \sum_{i=1}^N L_i + \sum_k W_k^2$$

$$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$

$$s = f(x; W) = Wx$$

treba $\nabla_W L$
nam



Numerički izvod nije dobar pristup ovde. Greška je funkcija od W , pa možemo da koristimo analitički izvod.

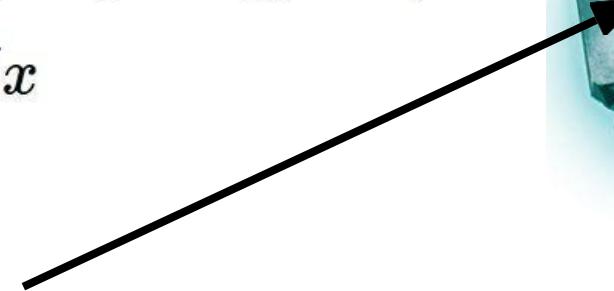
$$L = \frac{1}{N} \sum_{i=1}^N L_i + \sum_k W_k^2$$

$$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$

$$s = f(x; W) = Wx$$

treba nam $\nabla_W L$

Analiza



Numerički izvod nije dobar pristup ovde. Greška je funkcija od W , pa možemo da koristimo analitički izvod.

$$L = \frac{1}{N} \sum_{i=1}^N L_i + \sum_k W_k^2$$

$$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$

$$s = f(x; W) = Wx$$

$$\nabla_W L = \dots$$

W:

[0.34,
-1.11,
0.78,
0.12,
0.55,
2.81,
-3.1,
-1.5,
0.33,...]

loss 1.25347

gradijent dW:

[-2.5,
0.6,
0,
0.2,
0.7,
-0.5,
1.1,
1.3,
-2.1,...]

dW = ...
(neka funkcija
od W)



Rezime:

- Numerički izvod: aproksimacija, spor, ali jednostavna formula
- Analitički izvod: tačan, brz, nije ga uvek lako naći, pa postoji mogućnost greške

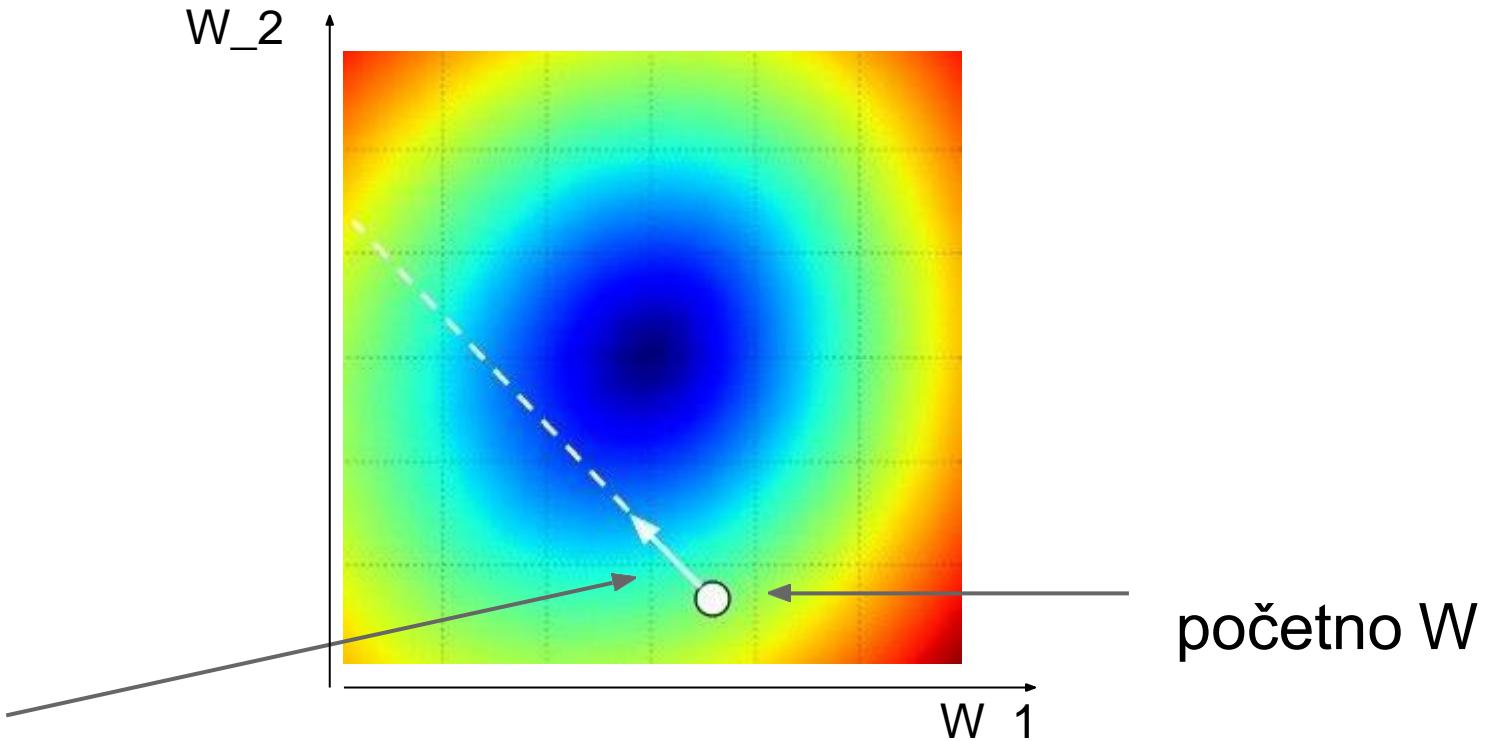
=>

U praksi: Uvek koristimo analitički izvod, ali ga proverimo numerički. Ovo se zove provera gradijenta (***gradient check***).

Gradijetni Spust (*Gradient Descent*)

```
# Vanilla Gradient Descent

while True:
    weights_grad = evaluate_gradient(loss_fun, data, weights)
    weights += - step_size * weights_grad # perform parameter update
```



pravac negativnog gradijenta

Gradijetni Spust sa Mini-Podskupovima

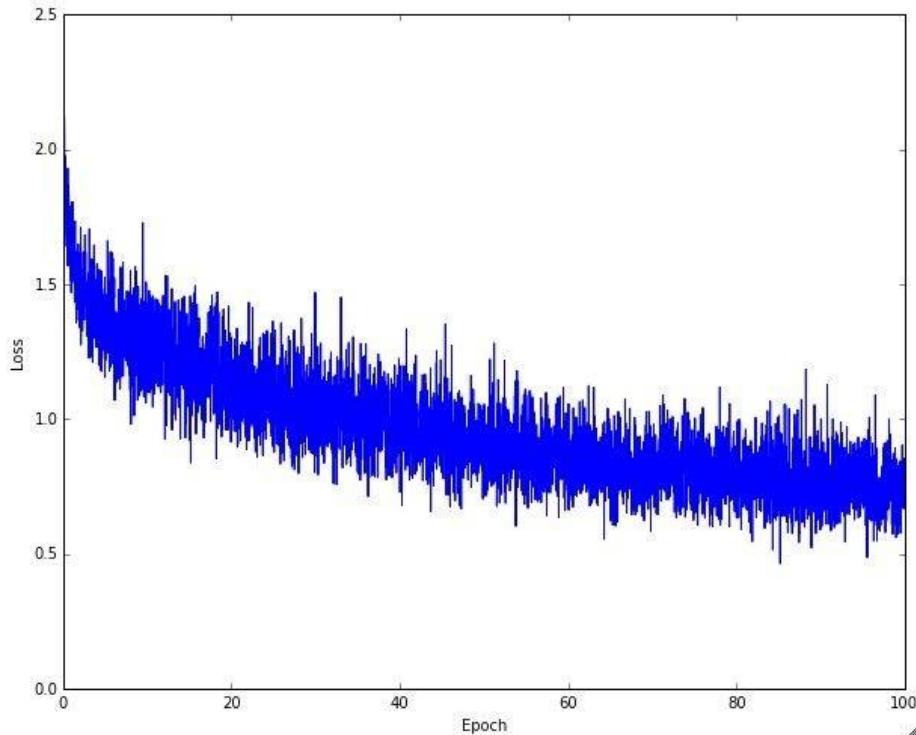
Mini-batch Gradient Descent

- koristimo samo deo obučavajućeg skupa da izračunamo gradijent.

```
# Vanilla Minibatch Gradient Descent

while True:
    data_batch = sample_training_data(data, 256) # sample 256 examples
    weights_grad = evaluate_gradient(loss_fun, data_batch, weights)
    weights += - step_size * weights_grad # perform parameter update
```

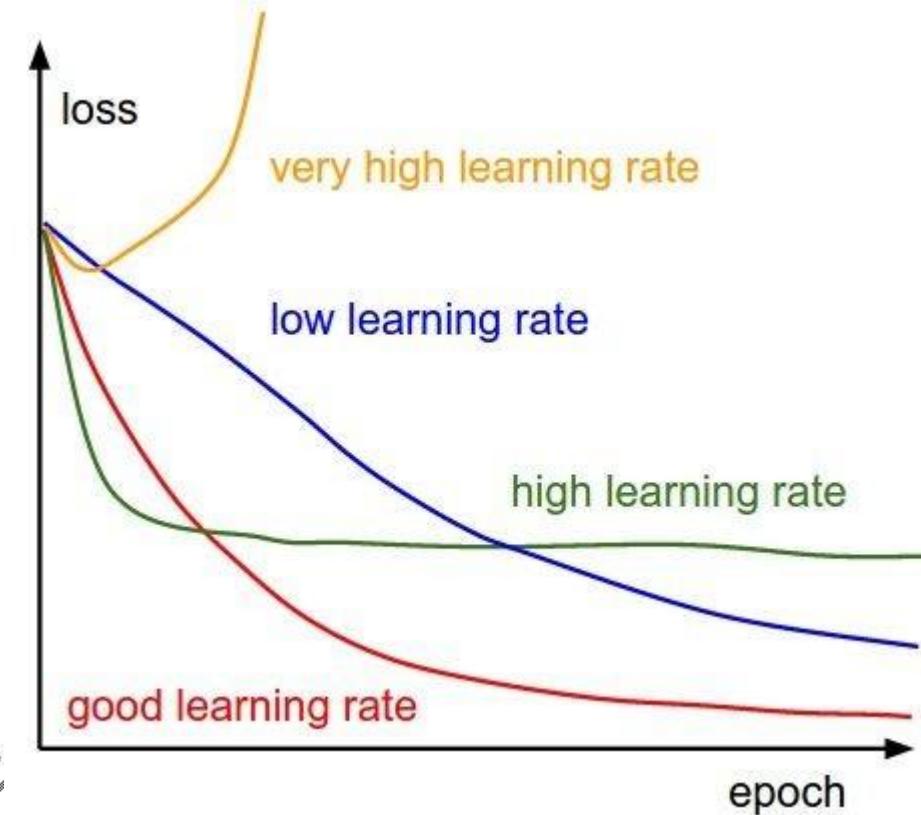
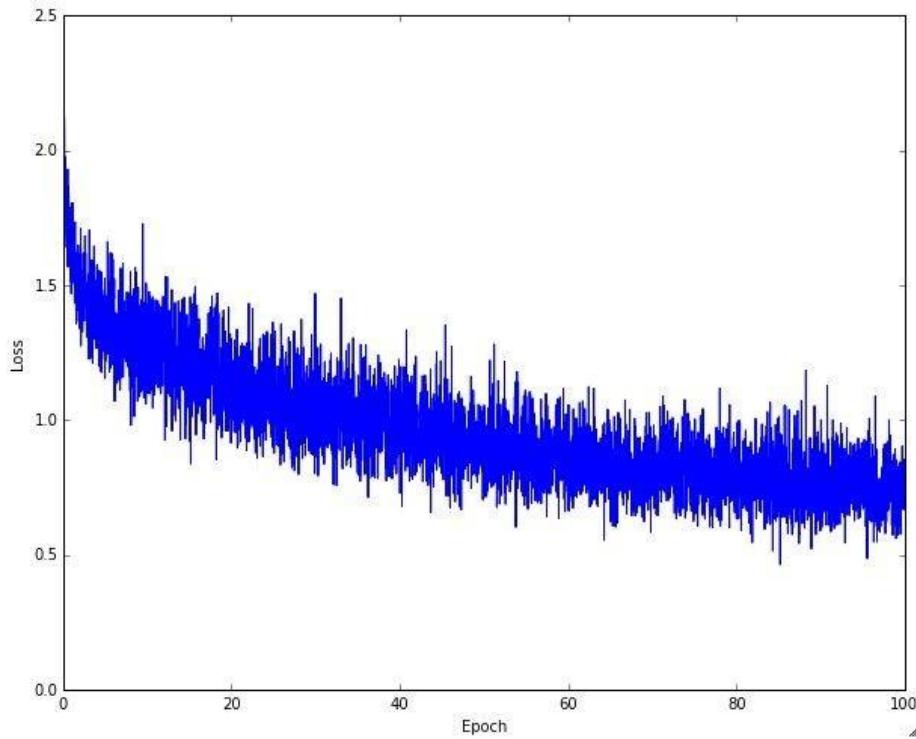
Tipične veličine podskupova su 32/64/128/256 primera



Primer optimizacije funkcije greške.

(Funkcija greške opada kroz
iteracije - epohe.)

Efekat veličine koraka (ili tempa učenja - *learning rate*)



Gradijetni Spust sa Mini-Podskupovima

Mini-batch Gradient Descent

- koristimo samo deo obučavajućeg skupa da izračunamo gradijent.

```
# Vanilla Minibatch Gradient Descent
```

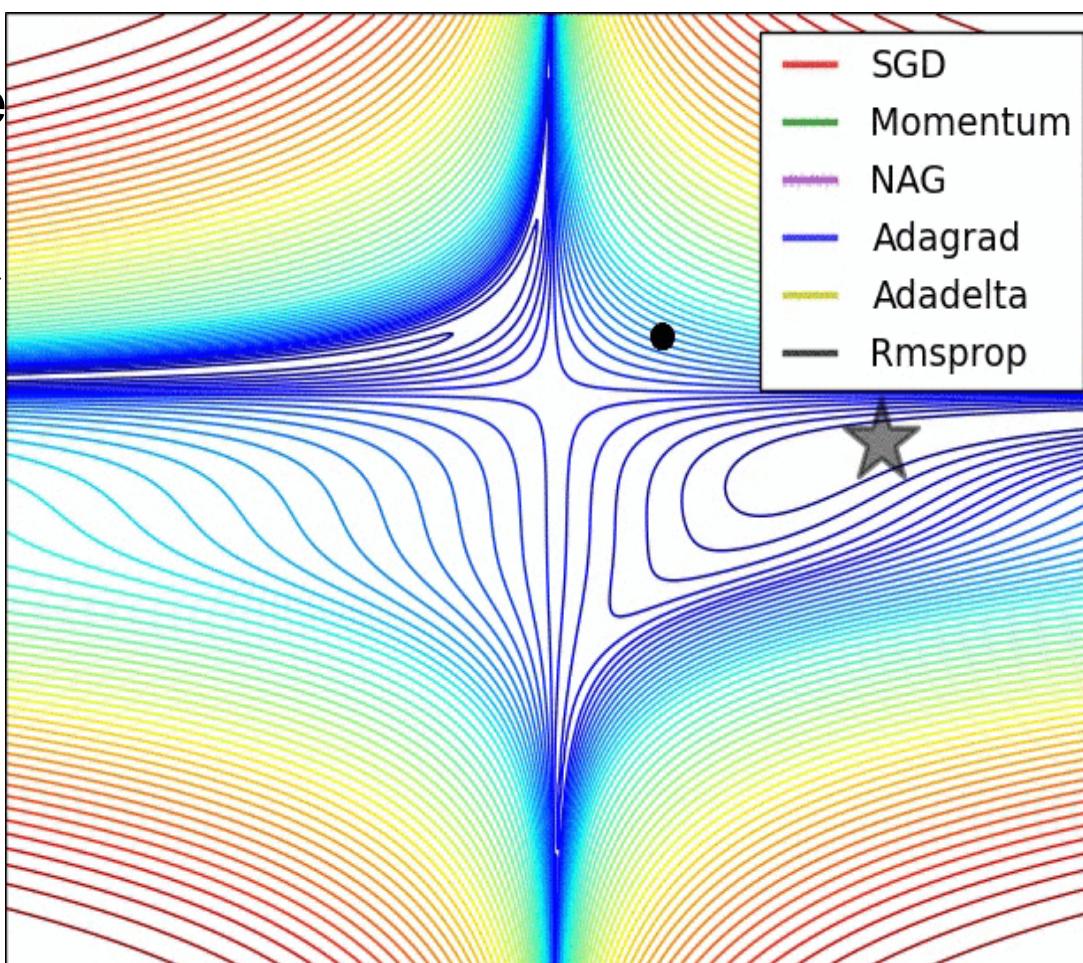


```
while True:  
    data_batch = sample_training_data(data, 256) # sample 256 examples  
    weights_grad = evaluate_gradient(loss_fun, data_batch, weights)  
    weights += - step_size * weights_grad # perform parameter update
```

Tipične veličine podskupova su 32/64/128/256 primera

Kasnije tokom kursa radićemo novije metode za promenu težina (momentum, Adagrad, RMSProp, Adam, ...)

Ponašanje različitih formula za promene težina

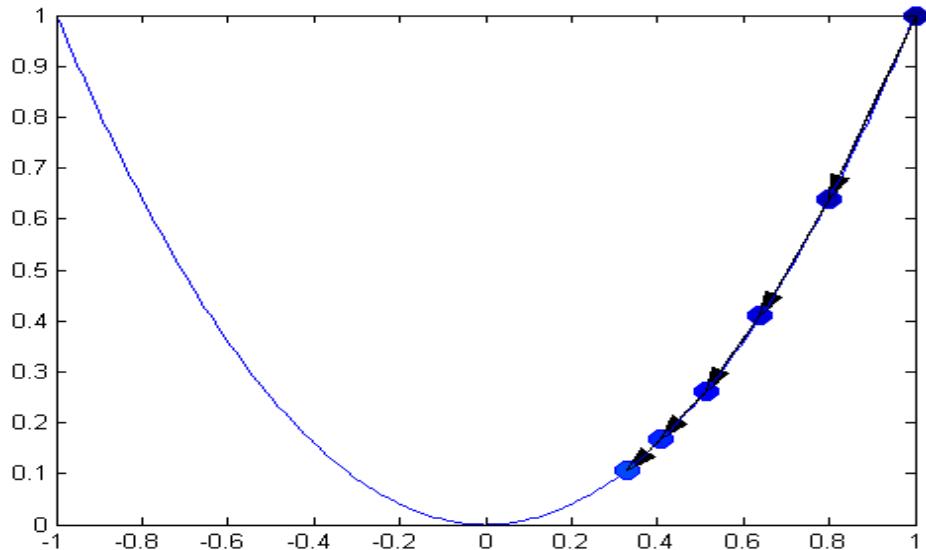
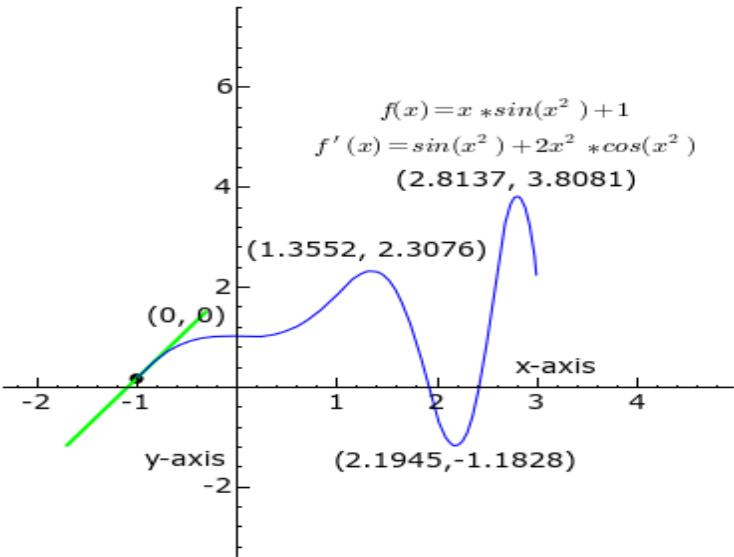


(image credits to Alec Radford)

Gradijetni Spust ponavljanje

Kako pronaći minimum funkcije?

- Izvod funkcije predstavlja nagib tangente na krivu funkcije
- Ideja: iterativno ćemo se pomerati ka minimumu
 - levo od minimuma: gradijent je negativan – pomeramo se u desno
 - desno od minimuma: gradijent je pozitivan – pomeramo se u levo



Gradient descent

- **Optimizaciona tehnika:** data je (proizvoljna) funkcija $J(\theta)$. Želimo da pronađemo $\min_{\theta} J(\theta)$

Ulaz	<ul style="list-style-type: none">• $J(\theta)$ – funkcija koja se optimizuje• θ_0 – početno rešenje• α – <i>learning rate</i> (veličina koraka)• $maxIters$ – maksimalan broj iteracija	Gradijent – određuje smer pretrage
Postupak	for $t = 1, 2, \dots, maxIters$: for $d = 1, 2, \dots, D$ • $\theta_d^{(t+1)} = \theta_d^{(t)} - \alpha \frac{\partial}{\partial \theta_d} J(\theta)$	
Izlaz	θ – tačka u kojoj funkcija $J(\theta)$ ima minimum	

Primena na linearu regresiju

- Fitovanje modela: želimo da pronađemo parametre θ za koje funkcija greške ima najmanju vrednost

$$J(\theta) = \frac{1}{2N} \sum_{i=1}^N (\theta_1 x^{(i)} + \theta_0 - y^{(i)})^2$$

- Ovo možemo uraditi primenom *gradient descent* algoritma:
 - Ponavljati do konvergencije:

$$\theta_d^{(t+1)} = \theta_d^{(t)} - \alpha \frac{\partial}{\partial \theta_d} J(\theta) \text{ za } d \in \{0,1\}$$

$$\frac{\partial J}{\partial \theta_0} = \frac{1}{N} \sum_{i=1}^N (\theta_1 x^{(i)} + \theta_0 - y^{(i)}) = \frac{1}{N} \sum_{i=1}^N (h_\theta(x^{(i)}) - y^{(i)}) \cdot 1$$

$$\frac{\partial J}{\partial \theta_1} = \frac{1}{N} \sum_{i=1}^N (\theta_1 x^{(i)} + \theta_0 - y^{(i)}) x^{(i)} = \frac{1}{N} \sum_{i=1}^N (h_\theta(x^{(i)}) - y^{(i)}) x^{(i)}$$

Primena na linearu regresiju

$$X = \begin{bmatrix} 1 & x^{(1)} \\ 1 & x^{(2)} \\ \dots & \dots \\ 1 & x^{(N)} \end{bmatrix} \quad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix} \quad h_{\theta}(x) = X\theta$$

$$\frac{\partial J}{\partial \theta_0} = \frac{1}{N} \sum_{i=1}^N (h_{\theta}(x^{(i)}) - y^{(i)}) \cdot 1 \qquad \rightarrow \qquad \frac{\partial J}{\partial \theta_d} = \frac{1}{N} \sum_{i=1}^N (h_{\theta}(x^{(i)}) - y^{(i)}) x_d^{(i)}$$
$$\frac{\partial J}{\partial \theta_1} = \frac{1}{N} \sum_{i=1}^N (h_{\theta}(x^{(i)}) - y^{(i)}) x^{(i)}$$
$$\theta_d^{(t+1)} = \theta_d^{(t)} - \frac{\alpha}{N} \sum_{i=1}^N (h_{\theta}(x^{(i)}) - y^{(i)}) x_d^{(i)}$$

Gradient descent

- Ako bismo imali samo jedan primer u skupu podataka, pravilo za ažuriranje parametara je:

$$\theta_d^{(t+1)} = \theta_d^{(t)} - \alpha(h_\theta(x) - y)x_d$$

- Ovo pravilo se naziva [LMS \(Least Mean Squares\) update rule](#) ili [Widrow-Hoff learning rule](#)
- Magnituda promene je proporcionalna greški $h_\theta(x) - y$
 - ako za dati primer naš prediktor daje vrednost veoma sličnu tačnoj vrednosti – nećemo mnogo menjati parametar
 - ako za dati primer prediktor ima veliku grešku – promena parametra će biti velika

Batch GD vs. Stochastic GD

- *Batch GD*: u svakom koraku istovremeno ažuriramo parametre modela koristeći sve trening podatke

for $t = 1, 2, \dots, \text{maxIters}$:

 for $d = 1, 2, \dots, D$

$$\bullet \quad \theta_d^{(t+1)} = \theta_d^{(t)} - \frac{\alpha}{N} \sum_{i=1}^N (h_\theta(x^{(i)}) - y^{(i)}) x_d^{(i)}$$

moramo da izračunamo grešku za sve primere pa tek onda radimo promenu

- *Stochastic GD* (ili *Incremental GD*): više puta prolazimo kroz skup podataka. Kad god najđemo na trening podatak, ažuriramo gradijent na osnovu tog (jednog) trening podatka

for $t = 1, 2, \dots, \text{maxIters}$:

 for $d = 1, 2, \dots, D$

 for $i = 1, 2, \dots, N$

$$\bullet \quad \theta_d^{(t+1)} = \theta_d^{(t)} - \alpha (h_\theta(x^{(i)}) - y^{(i)}) x_d^{(i)}$$

nema sume ovde, promenu radimo odmah za svaki primer

Mini-Batch Stochastic GD

- *Batch GD*: u svakom koraku istovremeno ažuriramo parametre modela koristeći sve trening podatke

batch_size = 32

for $t = 1, 2, \dots, \text{maxIters}$:

mini_batch = slučjano odabrana 32 primera iz skupa podataka

for $d = 1, 2, \dots, D$

$$\bullet \theta_d^{(t+1)} = \theta_d^{(t)} - \frac{\alpha}{batch_size} \sum_{i=1}^{batch_size} (h_\theta(x^{(i)}) - y^{(i)}) x_d^{(i)}$$

ovde koristimo samo primere iz skupa mini_batch

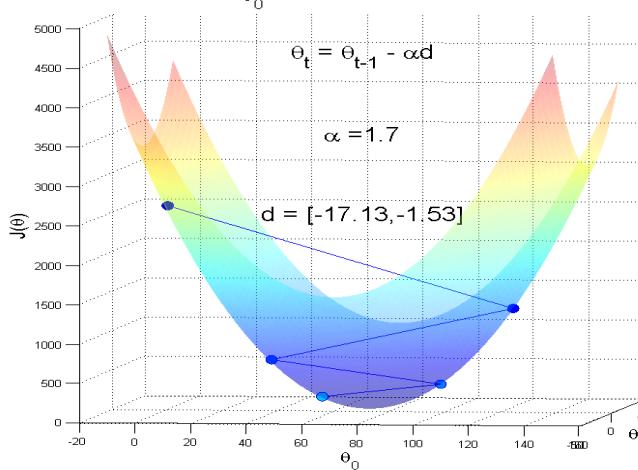
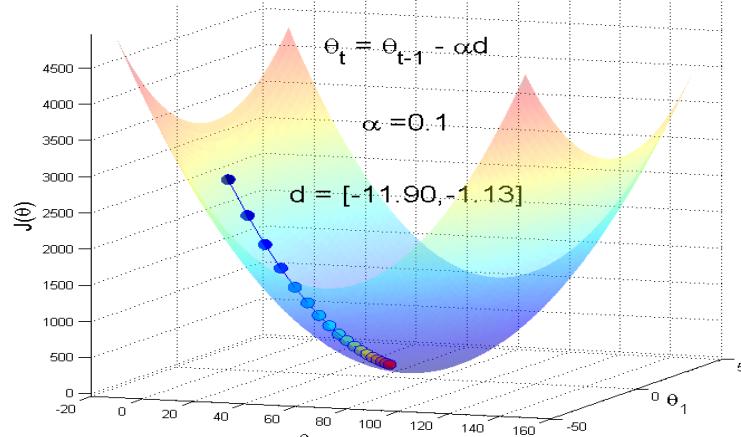
Batch GD vs. Stochastic GD

- *Batch GD* mora da skenira ceo skup podataka da bi napravio jedan korak
 - Ovo je skupa operacija ako je broj podataka N velik
- *Stochastic GD* može da napreduje odmah, i napreduje sa svakim uočenim trening podatkom
- Često, *stochastic GD* dovede θ „blizu“ minimuma mnogo brže od *batch GD*
- Međutim, dešava se da nikada ne „konvergira“ u minimum (parametri θ osciluju oko minimuma)
 - U praksi, tačke „blizu“ minimuma su dovoljno dobre
- Iz ovih razloga, kada je skup podataka velik, preferiramo *stochastic GD*

Mini-Batch Stochastic GD

- Predstavlja ravnotežu koja ispravlja mane prethodna dva algoritma
- Najčešće se koristi u praksi

Odabir α

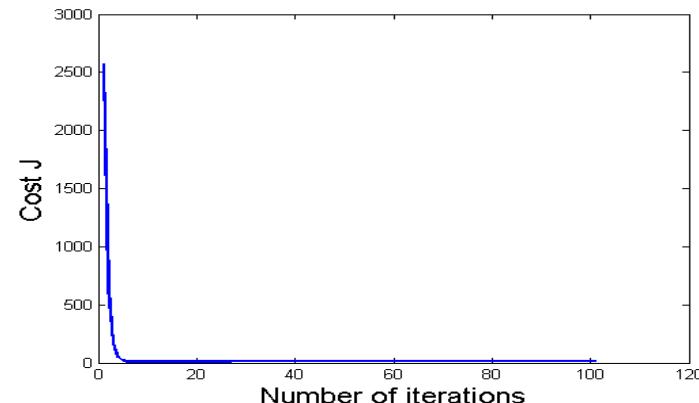
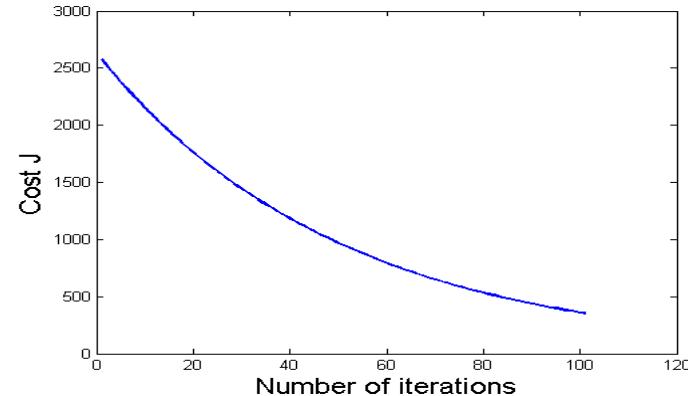
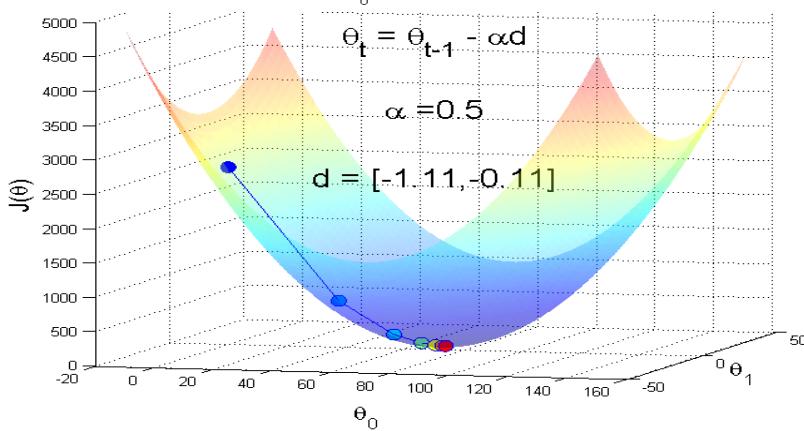
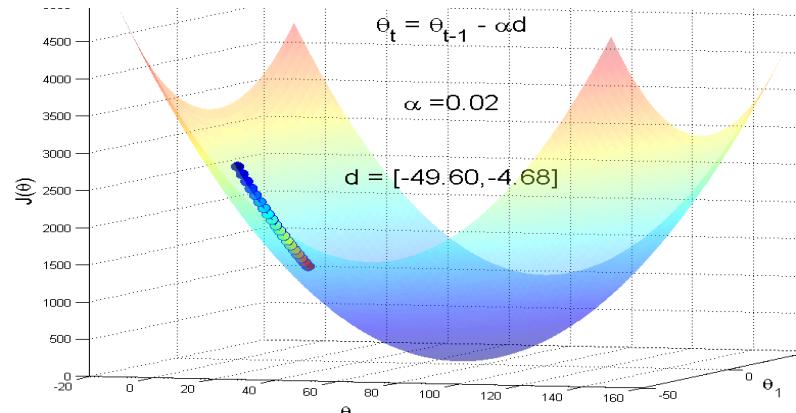


$$\theta_d^{(t+1)} = \theta_d^{(t)} - \alpha \frac{\partial}{\partial \theta_d} J(\theta)$$

- *Gradient descent* može da konvergira u minimum za fiksiranu vrednost α : kako se približavamo minimumu koraci su sve manji jer je gradijent (nagib) sve manji
- Ako se nalazimo levo od minimuma vrednost gradijenta je negativna: θ će da raste
- Ako se nalazimo desno od minimuma vrednost gradijenta je pozitivna: θ se smanjuje

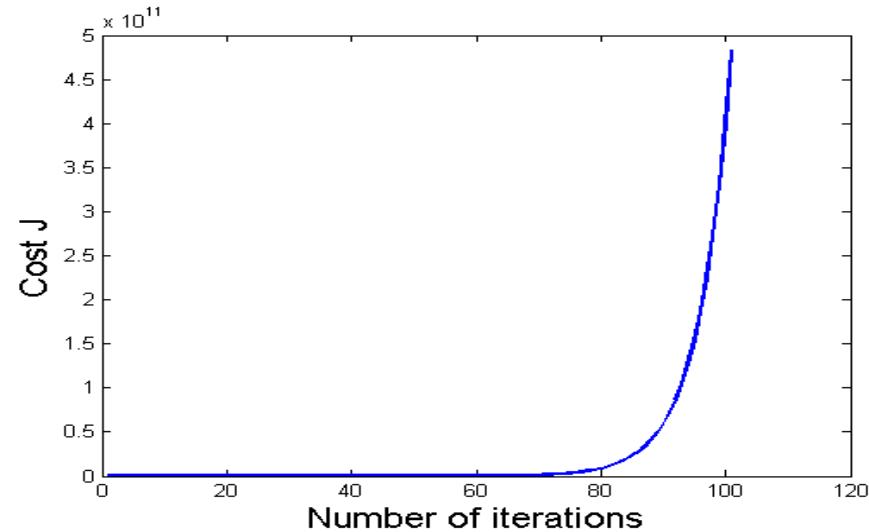
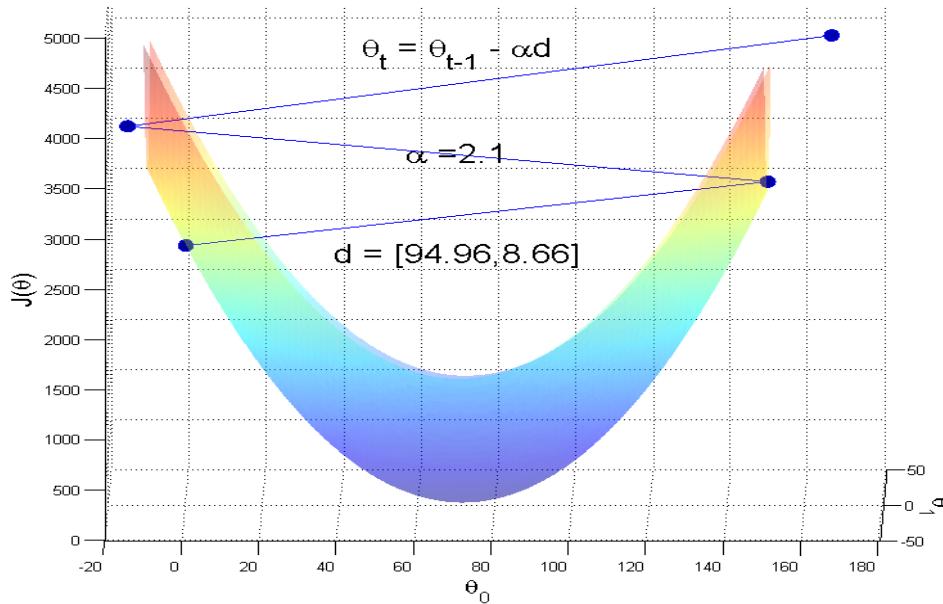
Odabir α

- Za male vrednosti α gradient descent je spor



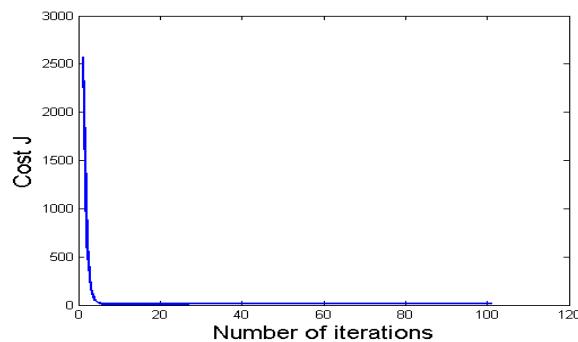
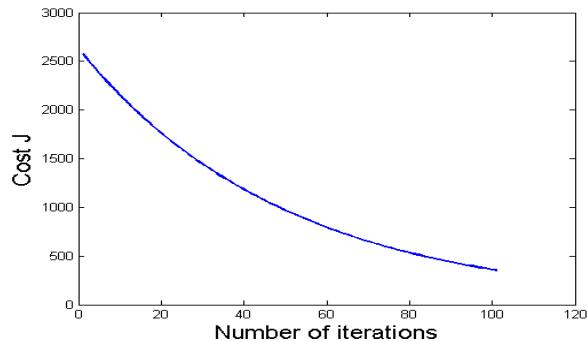
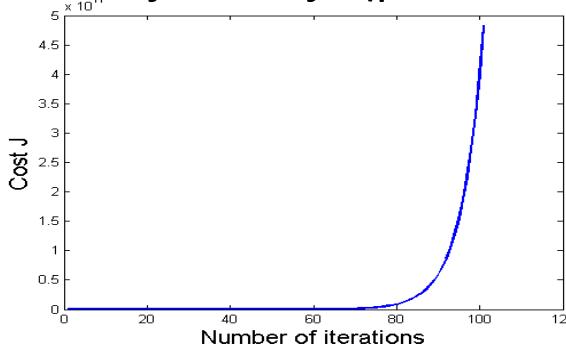
Odabir α

- Za preveliko α *gradient descent* ne konvergira, a može čak i da divergira



Odabir α - zaključak

- Da bismo odredili α , najbolje je posmatrati grafik funkcije greške $J(\theta)$ u odnosu na broj iteracija (probati $\alpha = 0.001, \alpha = 0.01, \alpha = 0.1, \alpha = 1, \dots$):



Došlo je do divergencije.
Smanjiti α

Algoritam je spor. Nije
konvergirao u zadatom broju
iteracija. Povećati α

Dobra vrednost α . Algoritam je
brzo konvergirao.

- Pored fiksne vrednosti α (koje radi kada je funkcija “*strongly convex*”), čest izbor je i “*stepsize schedule*” – smanjivanje koraka α sa brojem iteracija:

- $\eta_t = \alpha/t$ ili $\eta_t = \alpha/\sqrt{t}$

