MAŠINSKO UČENJE

Šta je to mašinsko učenje?

Disciplina koja omogućava računarima da uče bez *eksplicitnog* programiranja (Arthur Samuel 1959).

1. Generalizacija znanja na osnovu prethodnog **iskustva** (podataka o pojavama/entitetima koji su predmet učenja)
2. Dobijeno znanje koristi se kako bi se dali odgovori na pitanja za entitete/pojave koji nisu ranije viđeni

Definicija (Tom Mitchell 1998)

Za kompjuterski program se kaže da uči
iz iskustva ***E (experience),***
vezanog za zadatak ***T (task),*** i
meru performansi ***P (performance),***
ukoliko se njegove performanse na zadatku ***T,***
merene metrikama ***P,***
unapređuju sa iskustvom ***E***

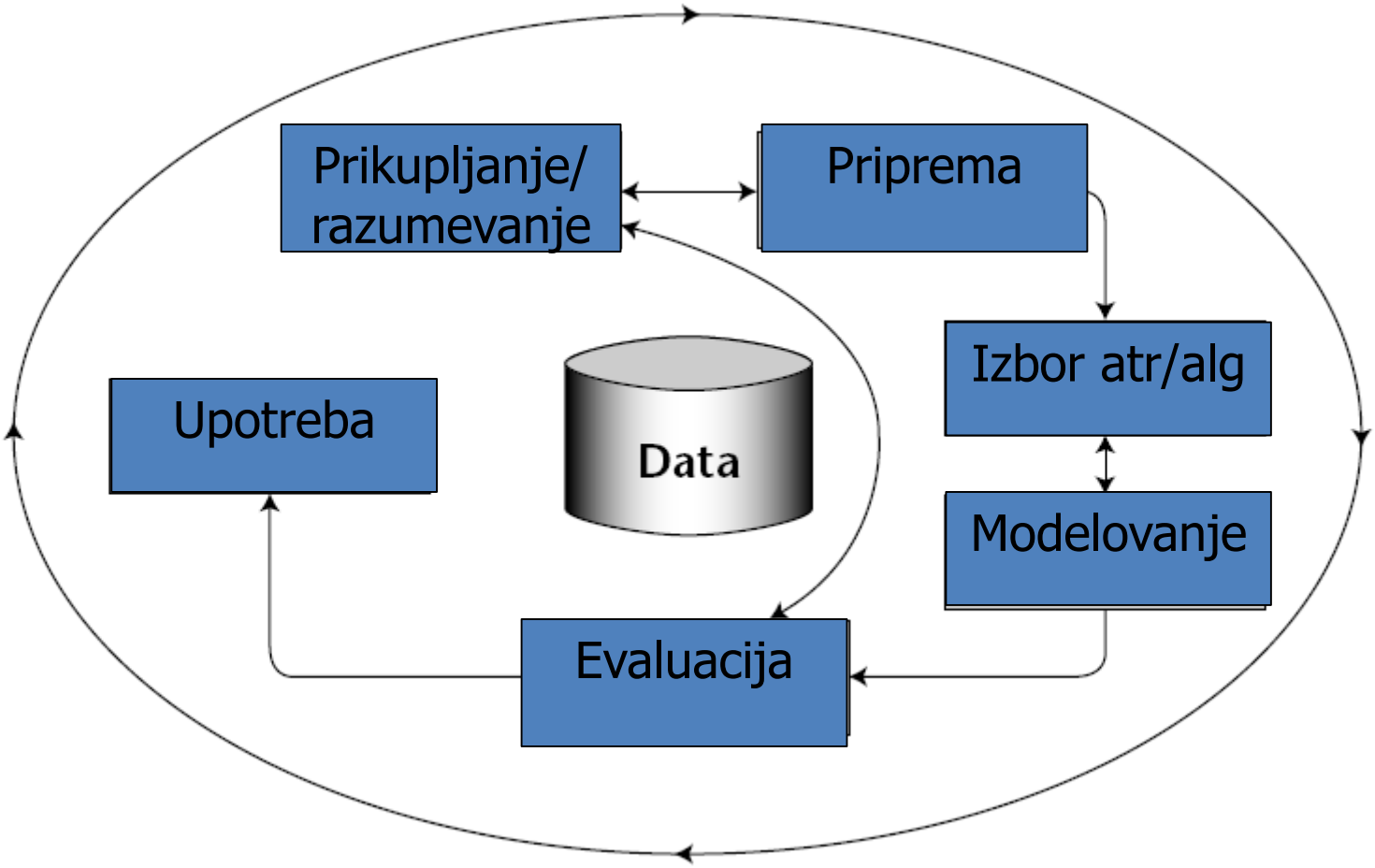
Prednosti ML-a?

1. Vrlo je teško precizno (algoritamski) opisati neke vrste zadataka koje ljudi lako rešavaju. Primeri: prepoznavanje lica (face detection), prepoznavanje govora (speech recognition)
2. Za neke vrste zadataka mogu se definisati algoritmi za rešavanje, ali su ti algoritmi vrlo složeni i/ili zahtevaju velike baze znanja Primer: automatsko prevođenje (MT)

Prednosti ML-a?

3. U mnogim oblastima se kontinuirano prikupljaju podaci sa ciljem da se iz njih “nešto sazna”; npr:
   - u medicini: podaci o pacijentima i terapijama
   - u marketingu: o korisnicima/kupcima i tome šta su kupili, za šta su se interesovali, kako su proizvode ocenili,…
- Analiza podataka ovog tipa zahteva pristupe koji će omogućiti da se otkriju pravilnosti, zakonitosti u podacima koje nisu ni poznate, ni očigledne, a mogu biti korisne (Data mining)

Proces mašinskog učenja



Podaci

- Potrebni su za trening, validaciju i testiranje modela
- Tipična podela na 60% za trening,
  20% za validaciju i 20% za testiranje
- Izbor uzoraka treba uraditi na slučajan način (random selection)

Atributi (*features*)

- Model treba da “verno” opisuje pojave/entitete
- Zato prepoznajemo osobine i odnose u datom domenu i predstavljamo ih atributima
- Izazov je odabrati prave atribute

Atributi - primeri

- Za kreditne zahteve: vrednost imovine podnosioca, primanja, zaposlenje, bračno stanje, itd.
- Za identifikaciju nepoželjne elektronske pošte (spam): naslov, prisustvo tipičnih reči (buy, visit,…), dužina email-a, broj primalaca, itd.

Testiranje

- Procena uspešnosti modela
- Koriste se podaci kojima model nije imao pristup u fazi učenja (20-30% ukupnih podataka)
- Uspešnost se utvrđuje različitim metrikama: tačnost, preciznost, odziv, …

TRAIN/VALIDATE/TEST

- Pored treniranja i testiranja modela, najčešće se radi i validacija modela kako bi se:
  a) izabrao najbolji model između više kandidata
  b) odredila optimalna konfiguracija parametara modela
  c) izbegli problemi *over/under-fitting-a*
- Podaci za validaciju koriste se za poređenje performansi
  - različitih modela (a);
  - izabranog modela sa različitim vrednostima parametara (b)

Analiza greške

- “ručno” pregledanje primera na kojima je model pogrešio
- Pomaže da se stekne osećaj zbog čega model greši i šta bi se moglo uraditi da se greške otklone; npr:
  - identifikovati suvišne atribute
  - identifikovati atribute koji nedostaju
  - drugačije podesiti parametre modela

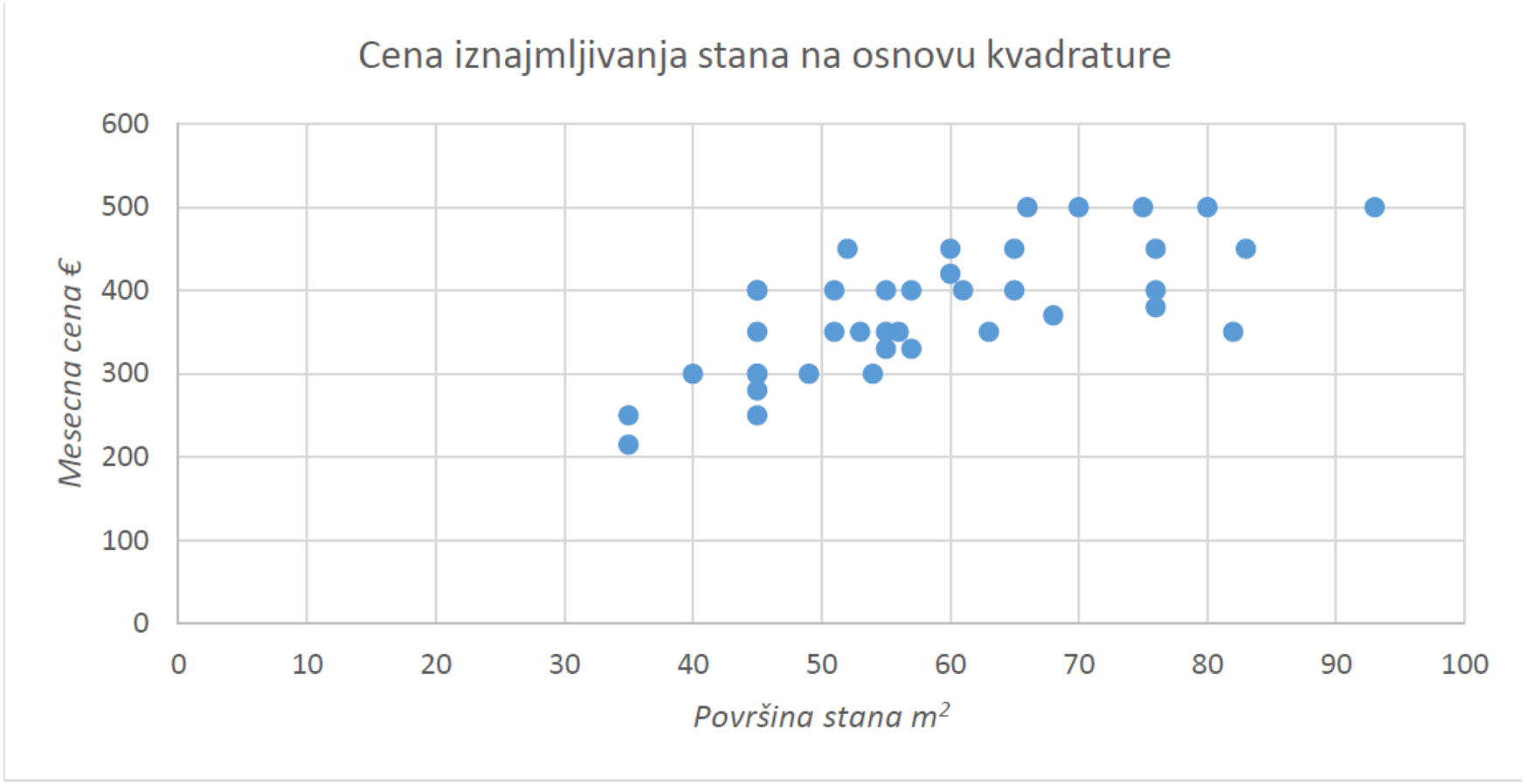
Podela

- Postoji nekoliko različitih tipova algoritama u mašinskom učenju. Dva najčešća i najosnovnija oblika mašinskog učenja su:
- Supervised learning (nadgledano učenje)
- Unsupervised learning

Nadgledano učenje

- Supervised learning (nadgledano učenje) predstavlja oblik mašinskog učenja na osnovu obeleženog skupa podataka za treniranje.
- Obeležen primer iz skupa podataka za treniranje, sastoji se od ulaznih podataka, tipično predstavljenih nizom realnih brojeva, kao i željene očekivane izlazne vrednosti.
- Ovi algoritmi, jednom trenirani sa obeleženim skupom podataka, kao svrhu imaju da izračunaju rezultat, za neobeležen skup podataka
- Izlazna vrednost može biti kontinualan realan broj ili neka diskretna vrednost.

Primer


Cena iznajmljivanja stana na osnovu kvadrature
Mesecna cena €
600
500
400
300
200
100
0
0
10
20
30
40
50
60
70
80
90
100
Površina stana m2

Nagledano učenje

- Ukoliko je pak, izlazna vrednost diskretnog tipa, tada se govori o klasifikaciji
- Neki od primera klasifikacija su:
  - Medicinsko testiranje radi utvrđivanja da li pacijent ima bolest ili ne
  - Prošao/pao metod testiranja kod provere kvaliteta u fabrikama
  - Testiranje krvne grupe koje pokazuje da li pacijent ima A, B, AB, ili O
  - Određivanje da li je dobijen email, spam ili ne

Nenadgledano učenje

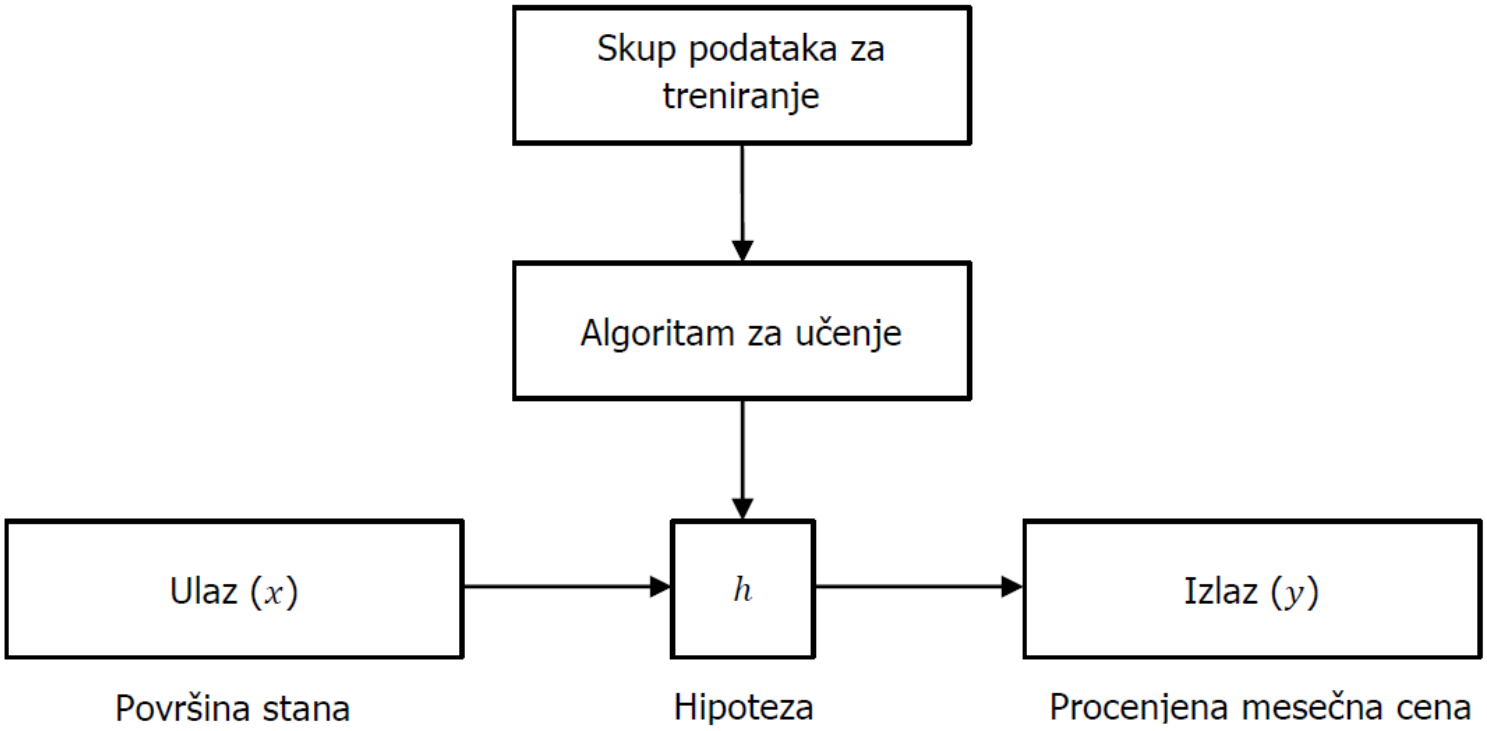
- Unsupervised learning (nenadgledano učenje) je drugi najčešći oblik mašinskog učenja, koji treniranje algoritama obavlja na osnovu neobeleženog skupa podataka.
- Neobeležen podatak iz skupa podataka za treniranje sastoji se samo iz ulaznih podataka, tipično predstavljenih nizom realnih brojeva.
- Zadatak algoritama koji spadaju u ovu grupu, jeste da otkriju zakonitosti u podacima.
- Nakon što je zakonitost u podacima pronađena, svrha ovih algoritama je da za novi skup podataka (takođe neobeležen) izračunaju neki zaključak koji zavisi od samog algoritma.

Linear regression

- Uvodi se notacija za određena svojstva skupa podataka za treniranje.
- Definisani su sledeći simboli:
  - m – broj primera u skupu podataka za treniranje
  - x – ulazni podatak
  - y – izlazni podatak

	Površina stana m^2 (x)	Mesečna cena € (y)
$i = 1$	35	215
$i = 2$	45	250
	35	250
	45	280
	40	300
$i = m$	...	...

Linear regression

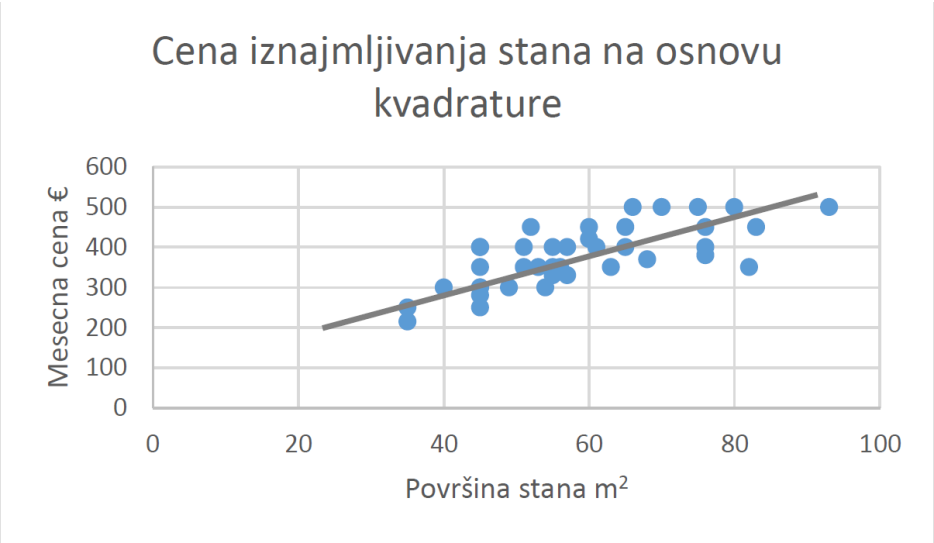


Linear regression

- h je simbol koji predstavlja hipotezu definisanu funkcijom koja vrši mapiranje iz ulaznog podatka x u izlazni y.
- Pri dizajniranju algoritma za učenje bitno je odlučiti kako definisati funkciju koja predstavlja hipotezu.
- Radi jednostavnosti, biće usvojeno da je hipoteza definisana na sledeći način:
- $h\theta\ (x) = \theta_0 + \theta_1 x$
- Simbol θ predstavlja skup parametara ove funkcije.

Linear regression

- Ovako definisana hipoteza predviđa da je rešenje predstavljano linearnom funkcijom od x.
- Primer sa slike ne mora uvek da bude rešenje
- U ovom slučaju radi se o modelu sa jednom promenljivom, ali hipoteza može da bude dosta složenija



Linear regression

- Odabrana funkcija za hipotezu je $h_\theta\,(x)$ i pitanje koje se postavlja jeste kako odabrati parametre θ_0 i θ_1.
- Za različite parametre dobijaju se različite funkcije
- Parametri koji najviše odgovaraju podacima su oni za koje je $h\,(x)$ najbliže y za svaki primer (x, y)

$$\underset{\theta_0,\theta_1}{minimum}\frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right)-y^{(i)}\right)^2$$

Linear regression

- Problem se dakle predstavlja kao potraga za θ_0,θ_1 takvim da prosečna kvadratna greška između procenjene vrednosti $h\theta$ ($x^{(i)}$) i prave izlazne vrednosti $y(i)$ bude minimalna.
- Po konvenciji, definiše se optimizaciona funkcija kao:

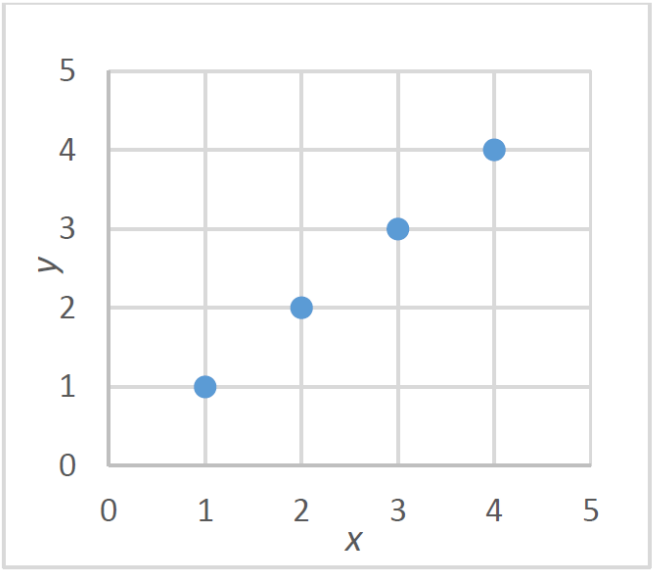
$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$$

- Ovakav oblik optimizacione funkcije, menja prethodni cilj u:

$$\underset{\theta_0,\theta_1}{\text{minimum}}\, J(\theta)$$

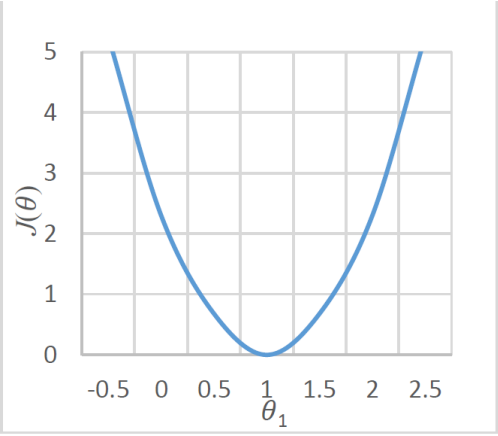
Linear regression

- Optimizaciona funkcija se može definisati i na drugačije načine, ali se prosečna kvadratna greška pokazala kao jedna od najboljih i najčešće korišćenih funkcija kada se radi o regresionim problemima.



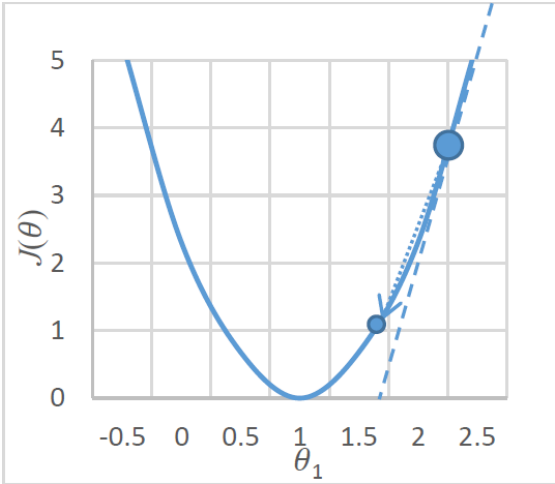
Linear regression

- Radi jednostavnosti recimo da je parametar θ_0 pronađen i fiksiran na 0, kako bi još jednostavnije prikazali optimizacionu funkciju, za parametar θ_1.
- Za različite vrednosti θ_1 dobijaju se različite vrednosti $J(\theta)$.
- Treba pronaći θ_1 za koje $J(\theta)$ ima najmanju vrednost, odnosno treba pronaći parametar θ_1 za koji je $h\ (x)$ najbliže y za svaki primer (x, y).
- U datom primeru za $_0$ = 0, θ_1 = 1, $J(\theta)$ ima najmanju vrednost što znači da hipoteza $h\ (x) = x$ najbolje odgovara datom skupu podataka.



Linear regression

- U prethodnom primeru, definisana je funkcija $J(\theta)$ i pokazano je da minimizacija te funkcije daje potreban skup parametara θ tj. parametre θ_0 i θ_1.
- Moguće je koristiti i odgovarajuci gradijent koji se koristi u raznim algoritmima mašinskog učenja.
- Koristi se za optimizaciju funkcije, u ovom slučaju $J(\theta)$ - traženje parametara dok se ne nađu oni za koje data funkcija ima najmanju vrednost
  - Započeti sa nekim skupom parametara θ
  - Menjati parametre θ, u cilju smanjivanja $J(\theta)$, dok se ne nađe minimum funkcije
- Neka je početna vrednost parametra $\theta_1 = 2.25$, i radi jednostavnosti prikaza neka je $\theta_0 = 0$.



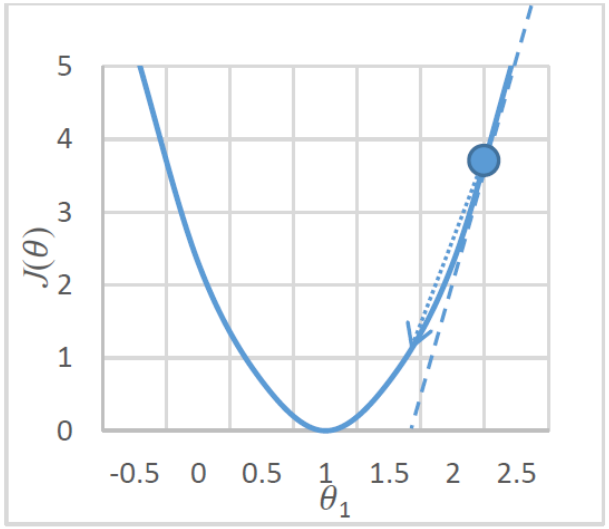
Linear regression

- Potrebno je odrediti nagib funkcije za određenu tačku koji se može izračunati parcijalnim izvodom funkcije $J(\theta)$, po nekom parametru θj , gde je j definisano kao indeks parametra (za dati primer j = 0 i j = 1).
- Veličina koraka se može kontrolisati novim parametrom α, koji se naziva stopa učenja (Learning rate)
- Ovaj parametar se koristi u većini algoritama mašinskog učenja, kao pozitivan realan broj veći od nula.

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

Linear regression

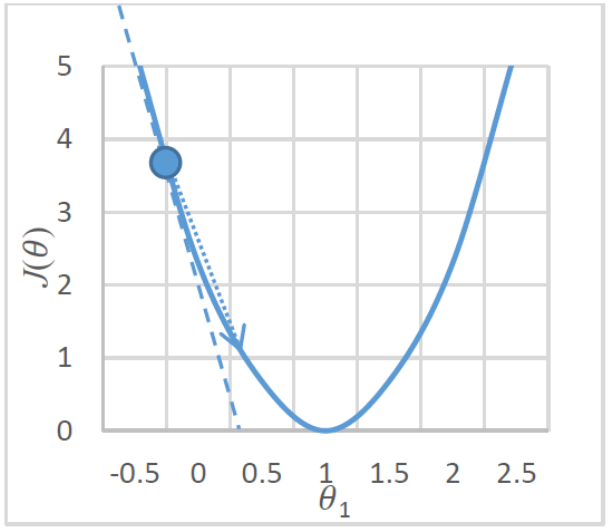
- Promena parametra u zavisnosti od nagiba optimizacione funkcije



$\frac{\partial}{\partial\theta_1}J(\theta) > 0$

$\theta_1 := \theta_1 - \alpha * pozitivan\ broj$

θ_1 se smanjuje



$\frac{\partial}{\partial\theta_1}J(\theta) < 0$

$\theta_1 := \theta_1 - \alpha * negativan\ broj$

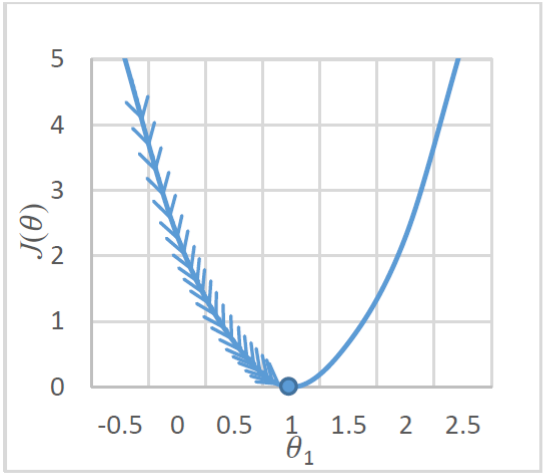
θ_1 se povećava

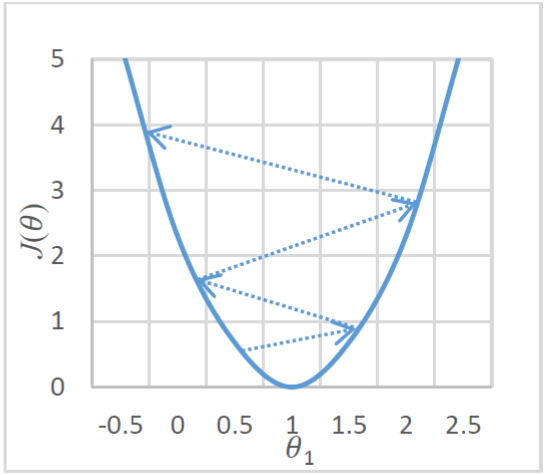
Linear regression

- Približavanje minimumu smanjuje faktor, pa će gradijent automatski praviti manje korake - stopu učenja nije potrebno vremenom smanjivati kako bi se ostvarila konvergencija ka minimumu funkcije.
- Međutim od stope učenja veoma zavisi ponašanje algoritma kao i uopšte pronalazak svih parametara θ, za koje $J(\theta)$ ima minimalnu vrednost.
- Ukoliko je stopa učenja premala, algoritam će biti veoma spor
- Ukoliko je stopa učenja prevelika moguće je da se minimum funkcije nikada neće pronaći, što znači da neće doći do konvergencije, a može se desiti da dođe i do divergencije.

Linear regression

- Primer premale i prevelike stope učenja





Linear regression

Ponavljati do konvergencije {

$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$ (istovremeno promeniti θ_j za $j = 0$ i $j = 1$)

}

Istovremena promena parametara je veoma bitna jer jedino tako će $J(\theta)$, vraćati tačne vrednosti iz prethodne iteracije petlje. Ispravna implementacija za dati primer je data u nastavku.

$$temp0 := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta)$$

$$temp1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta)$$

$$\theta_0 := temp0$$

$$\theta_1 := temp1$$

Linear regression

Cilj je naravno pronalazak skupa parametara za koje optimizaciona funkcija $J(\theta)$ ima najmanju vrednost. Gradient descent ne koristi funkciju $J(\theta)$ u njenom originalnom obliku već kroz parcijalni izvod, pa je potrebno izračunati šta taj faktor predstavlja.

$$\frac{\partial}{\partial \theta_j} J(\theta) = \frac{\partial}{\partial \theta_j} \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)} \right)^2$$

$$\frac{\partial}{\partial \theta_j} J(\theta) = \frac{\partial}{\partial \theta_j} \frac{1}{2m} \sum_{i=1}^{m} \left(\theta_0 + \theta_1 x^{(i)} - y^{(i)} \right)^2$$

Posmatrajući dobijeno opšte rešenje, mogu se izračunati parcijalni izvodi u zavisnosti o indeksa parametra j.

$$j = 0: \frac{\partial}{\partial \theta_0} J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)} \right)$$

$$j = 1: \frac{\partial}{\partial \theta_1} J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)} \right) x^{(i)}$$

Linear regression

Ponavljaj do konvergencije {

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta \left(x^{(i)} \right) - y^{(i)} \right)$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta \left(x^{(i)} \right) - y^{(i)} \right) x^{(i)}$$

}

- Prikazana implementacija algoritma u svakoj iteraciji koristi sve primere iz skupa podataka za treniranje.
- Ovakav pristup se naziva Batch Gradient descent