# **OSNOVNI KONCEPTI**

## **Tri pristupa strojnom učenju**

Nadzirano učenje – Podatci su parovi (ulaz, izlaz) = (x, y). Treba pronaći y' = f(x)

* Ako je y diskretna/nebrojčana vrijedost -> **klasifikacija**
* Ako je y kontinuirana/brojčana vrijednost -> **regresija**

Nenadzirano učenje – Dani su podatci bez ciljnih vrijednosti. Treba pronaći pravilnost u podatcima

* Grupiranje (engl. Clustering)
* Procjena gustoće (engl. Density estimation)
* Smanjenje dimenzionalnosti (engl. Dimensionality reduction)

Podržano učenje – učenje **optimalne strategije** na temelju pokušaja s odgođenom nagradom

## **Algoritam strojnog učenja**

1. Priprema podataka i preliminarna analiza
2. Ručno označavanje podataka za treniranje i ispitivanje
3. (opcionalno) Ekstrakcija značajki – na ulaz algoritma dovode se podatci u obliku skupa primjera.
4. (opcionalno) Redukcija dimenzionalnosti – ako imamo previše nepotrebnih značajki
5. **Odabir modela**
6. **Učenje modela**
7. **Vrednovanje modela**
8. Dijagnostika i ispravljanje
9. Instalacija

**Primjer** – jedna podatkovna točka. Svaki primjer prikazujemo kao vektor značajki **x**. Broj primjera označavamo sa *N*. Primjeri žive u **prostoru primjera** ili **ulaznom prostoru** (engl. instance space, input space). Taj prostor označavat ćemo s X . Ako su značajke numeričke, tj. xi ϵ R, onda je X ϵ Rn, tj. to

je n-dimenzijski vektorski prostor. Njegova dimenzija je n ona odgovara broju značajki.

**Značajka** – karakteristika primjera koja je bitna za problem koji rješavamo. Zovemo ih i atributi. Broj značajki označavamo sa *n*. Idealno je da je *N >> n*.

**Oznaka -** (engl. *label*) je oznaka klase primjera u klasifikaciji ili ciljna vrijednost primjera u regresiji. Oznaku označavamo sa y, a skup svih oznaka označavamo sa Y. Kada je K = 2 govorimo o **binarnoj klasifikaciji**.

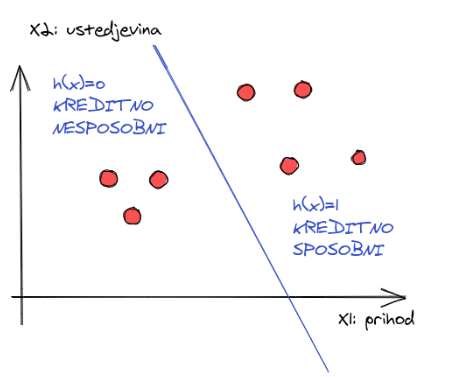
**Skup označenih primjera** označit ćemo sa *D*. To je skup parova

* Skup označenih parova *D* možemo prikazati kao matrice neoznačenih primjera **X** i vektora njihovih oznaka **y.** Svaki redak matrice **X** je jedan vektor značajki odnosno primjer **x(i)**. Svaki stupac matrice **X** je jedna značajka kroz sve primjere. Komponente vektora **y** odgovaraju oznaci za svaki pojedinačan primjer. Matrica **X** naziva se *matrica dizajna*.

primjer

značajka a

**Hipoteza -** funkcija koja svakom primjeru iz (prostora primjera) dodjeljuje oznaku klase (kod klasifikacije) ili brojčanu vrijednost (kod regresije). Ako govorimo o binarnoj klasifikaciji, funkcija *h* zapravo dijeli prostor primjera na dva poluprostora. Jedan za koji je *h(****x****) = 1* i jedan za koji je *h(****x****) = 0*



**Implicitna jednadžba pravca -** . Takva funkcija lako se generalizira na višedimenzionalne prostore. Koristeći eksplicitnu funkciju izgubili bi mogućnost orijentacije klasa u podprostorima. Općenito za n-dimenzionalni ulaz klasifikator treba imati n+1 parametara

Funkcija *h* parametrizirana je parametrima θ (malo theta). Općenito, to neće biti samo jedan parametar, nego niz parametara. Tako da zapravo govorimo o **vektoru parametara θ.** Za zapis funkcije koristimo *h*(**x**; **θ**)

**Model –** skup hipoteza. Budući da su hipoteze funkcije, model je **skup funkcija**. Označavamo ga s *H*.

Supskript θ označava da su elementi tog skupa **indeksirani** parametrom θ tj. vektor θ odgovara jednoj funkciji iz skupa H. Za zadani vektor parametara **θ**, možemo dohvatiti njemu odgovarajuću funkciju *h ϵ H*. se preslikava u *h*.

**Učenje (treniranje modela)** nije ništa drugo nego **pretraživanje** skupa hipoteza *H* u nastojanju da se nađe najbolja hipoteza *h ϵ H.* Pretraživanje po kvantitativnom kriteriju dobrote, **optimizacijski problem.** Iscrpno pretraživanje je nemoguće, potrebna nam je **heuristička optimizacija**.

## **Empirijska pogreška i funkcija gubitka**

**Empirijska pogreška** - **numerička procjena** koliko je hipoteza dobra na skupu označenih primjera D. Govori nam koliko hipoteza griješi kad klasificira primjere (klasifikacija) ili koliko su vrijednosti udaljene od ciljnih vrijednosti (regresija).

– **Funkcija empirijske pogreške** hipoteze *h* na označenom skupu označenih primjera *D*

**Funkcija gubitka** (engl. *loss function*) – za dani označeni primjer, govori nam koliko je model izgubio na točnosti tj. koliko se povećala pogreška modela na tom jednom primjeru. Definiramo kao .

## **Tri komponente algoritma SU**

1. **Model**
2. **Funkcija gubitka** odnosno njoj pripadna funkcija pogreške
3. **Optimizacijski postupak** – Postupak kojim unutar modela *H* pronalazimo *h\** koja minimizira empirijsku pogrešku.

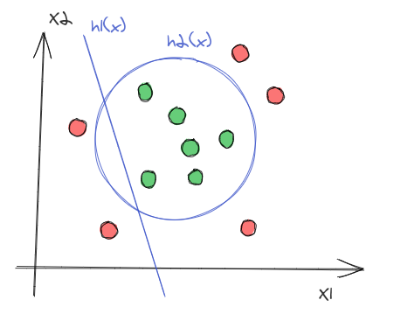


## **Složenost modela**

Moguće da

**Šum** (engl. *noise*) – neželjena anomalija u podatcima zbog koje dolazi do odstupanja opaženih vrijednosti od pravih vrijednosti. Mogući uzroci:

* **Nepreciznost** pri mjerenju značajki (mjerni instrumenti)
* **Pogreške u označavanju** (teacher noise). Krivo označeni podatci
* **Postojanje skrivenih značajki** (latentnih varijabli) npr. ako nam nedostaje jedna dimenzija (bitnu značajku smo zaboravili)
* **Nejasne granice** između klasa (subjektivnost)



Ako se naš model previše prilagodi šumu, **loše će generalizirati**.

**Podnaučenost** (engl. *underfitting*) – model H je prejednostavan u odnosu na stvarnu klasifikaciju/funkciju, što rezultira lošim predikcijama modela i na viđenim i na neviđenim primjerima.

**Prenaučenost** (engl. *overfitting*) – model H je previše složen u odnosu na stvarnu klasifikaciju/funkciju, što rezultira dobrim predikcijama na viđenim primjerima, ali lošim predikcijama na neviđenim primjerima.

Prednosti jednostavnijeg modela su: **bolja generalizacija, lakše učenje, lakše tumačenje**.

## **Odabir modela**

Odabiremo model iz skupa modela {H1, H2, ... , Hn}. Zapravo biramo **stupanj nelinearnosti**.

Klasifikacija – biramo hoće li granice klasa biti pravci, kružnice itd.

Regresija – biramo stupanj polinoma

Ako složenost modela možemo nekako kvantificirati i dovesti u vezu s nekim parametrom koji upravlja složenošću modela, onda takav parametar nazivamo **hiperparametar modela**. Hiperparametri određuju **model**, a parametri određuju **hipotezu** koju koristimo. **Odabir modela moramo napraviti mi (čovjek) dok treniranje modela radi algoritam strojnog učenja.**

Implementacija točnih modela -> **unakrsna provjera** (engl. *cross-validation*).

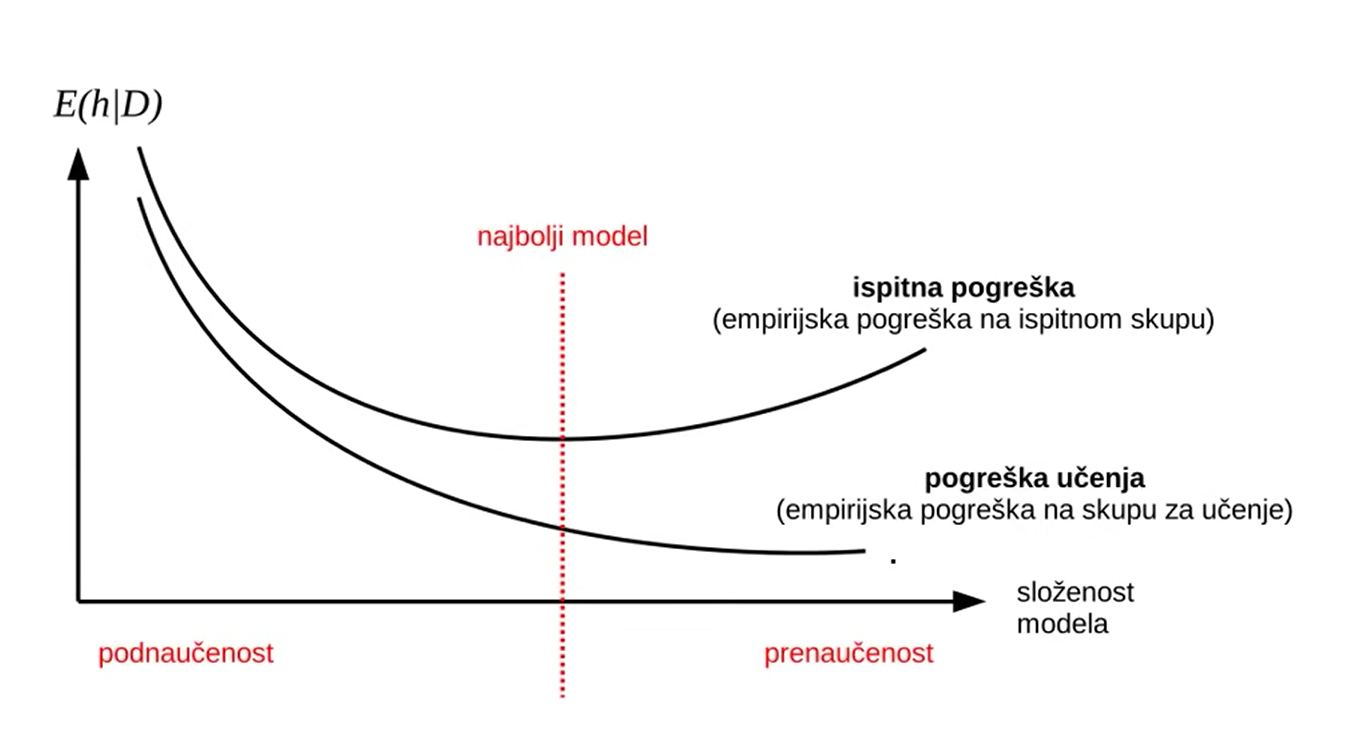
## **Unakrsna provjera**

Metoda za procjenu sposobnosti generalizacije modela. Unutar familije modela koju razmatramo, želimo odabrati onaj model koji najbolje **generalizira**, tj. koji dobro radi na **neviđenim primjerima**.

Unakrsnu provjeru provodimo tako da skup označenih primjera *D* podjelimo na **skup označenih primjera za učenje** i **skup označenih primjera za ispitivanje**. Ti skupovi moraju biti **disjunktni**.

**Empirijska pogreška učenja** – pogreška hipoteze *h* izračunata na skupu za učenje *Dtrain*.

**Empirijska ispitna pogreška** – pogreška hipoteze *h* izračunata na skupu za ispitivanje *Dtest*.



**DODATNO**

**DODATNO**

Pristupima temeljenima na dubokom strojnom učenju nastoji se izbjeći **ekstrakcija značajki**, tako da neuronska mreža ne uči samo klasifikaciju/regresiju već implicitno uči ekstrakciju bitnih značajki tj. **uči reprezentaciju.** Učenjem reprezentacije posao ekstrakcije značajki prebacuje se sa čovjeka na algoritam strojnog učenja.

Različite hipoteze imat će različite vektore parametara θ, ali obrat ne vrijedi. Na primjer, ako je ulazni prostor diskretan onda možemo imati dva pravca koja su malo različita , ali ipak daju identičnu klasifikaciju.

**Induktivna pristranost** – skup komponenti koje algoritam strojnog učenja koristi za predviđanje rezultata kada su informacije nepotpune ili nesigurne. Ove pretpostavke pomažu algoritmu da generalizira iz primjera za učenje na nove, neviđene primjere.

* Pretpostavke koje uvodi induktivna pristranost mogu ograničiti sposobnost algoritma da uči, npr. ako algoritam SU-a pretpostavlja da su svi podatci linearno odvojivi, onda će se jako loše nositi s podatcima koji to nisu
* Pristranost može biti **eksplicitna** (jasno definirane pretpostavke)ili **implicitna** (nejasne ili neizrečene)
* Pristranost jezikom i pristranost preferencijom

# **LINEARNA REGRESIJA I.**

## **Jednostavna regresija**

**– ulazne varijable** (nezavisne varijable, prediktorske varijable, kovarijanti)

**– izlazna varijabla** (zavisna varijabla, kriterijska varijabla)

**Model** – vektor parametara modela označavamo s (**težine** ili **beta-koeficijenti**)

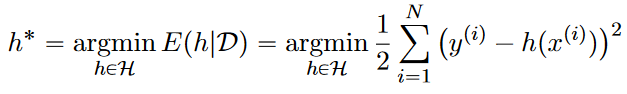
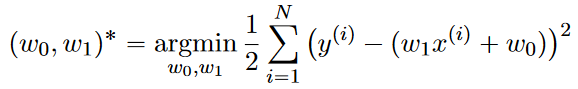
**Funkcija gubitka** – **kvadratno odstupanje** između predviđene i ciljne vrijednosti (kvadrat reziduala)

* + Obična razlika – negativni i pozitvni reziduali se poništavaju
  + Apsolutna vrijednost – nije derivabilna

Koristimo ½ radi kasnije matematičke jednostavnosti. Nema nikakvog utjecaja na daljnju optimizaciju. Minimum funkcije ostaje isti

**Funkcija pogreške (quadratic error function)** – očekivana vrijednost funkcije gubitka (očekivanje) na skupu označenih primjera

**Optimizacijski postupak** – potraga za hipotezom u modelu koja minimizira funkciju pogreške

* + - Konveksna funkcija – jedna nultočka derivacije **minimum**

Rješenje je u **zatvorenoj formi**

|  |  |  |
| --- | --- | --- |
| **Vrste Regresije** | **Jedan izlaz (y)** | **Više izlaza** |
| ***Jedan ulaz* (x)** | (Univarijatna) jednostavna | Multivarijatna jednostavna |
| ***Više ulaza*** | (Univarijatna) višestruka | Multivarijatna višestruka |

## **Tri komponente algoritma LINEARNE REGRESIJE** (Matrični račun)

1. **Model**

* Svaki vektor primjera *x(i)* proširimo tzv. **dummy značajkom**. . koristimo pri množenju s w0

- gdje je skalarni produkt dvaju vektora

1. **Funkcija gubitka** (i pripadna **funkcija pogreške**)
2. **Optimizacijski postupak** (metoda najmanjih kvadrata, OLS)



## **Postupak najmanjih kvadrata**

Kako izračunati optimalne parametre i to za višestruku linearnu regresiju gdje je n > 1

Egzaktno rješenje je krivi pristup. Ako je onda je i to donosi probleme:

1. Rješenje može biti **loše definirano** (engl. *ill-defined*). Sustav nema rješenja (**nekonzistentan, rang** matrice X manji je od ranga X|y) ili pak **nema jedinstvenog rješenja** (beskonačno rješenja, rang matrice X manji je od broja stupaca tj. n + 1)
2. Da bi matrica dizajna *X* imala inverz, mora biti **kvadratna.** Možemo koristiti Gaussovu metodu eliminacije, ali opet nema garancije da se problem pod brojem 1 neće ispuniti.
   * Preodređen sustav (engl. *overdetermined*)
   * Pododređen sustav (engl. *underdetermined*)

Skalare možemo transponirati bez da išta promijenimo :

Zbog ograničenja koje smo ustanovili umjesto da tražimo egzaktno rješenje tražit ćemo **približno** rješenje. Stoga uvodimo **matrični račun** za računanje minimalnih kvadratnih odstupanja **višestruke** **regresije**.

Primjer x(i) je dimenzija **[1 x (n+1)]**. je dimenzija **[1 x (n+1)]**. Matrica dizajna X je dimenzija **[N x (n+1)]**. y je dimenzija **[1x1]**, a Y je dimenzija **[Nx1]**. Zbog toga se dogodi zamjena kada sumu pretvaramo u matrični račun.

Pravila za deriviranje matrica

Sada želimo naći minimum ove funkcije po parametrima :

**Pseudoinverz** (Moore-Penroseov inverz) – poopćenje je koncepta inverza matrice. Za razliku od inverza koji postoji samo za kvadratnu matricu, pseudoinverz **postoji za** **baš svaku matricu**. Ako je **X** **kvadratna** i **punog** **ranga** onda

Rješenje koje minimizira **L2-normu**

**Gramova matrica –** **inverz** umnoška matrica dizajna sa samom sobom dimenzije su joj [n+1 x n+1]

* + Može biti poprilično velika
  + Skup proces (uobičajno O3), ovisi o broju značajki *n*
  + Rang Grahamove matrice jednak je rangu **matrice dizajna X** (detaljnije u DODATNO)
  + Kada je N < n+1 pseudoinverz računamo **rastavom na singularne vrijednosti (SVD)**
  + Matrica je **simetrična** i **pozitivno semidefinitna**

## **Probabilistička interpretacija regresije**

Gubitak smo definirali kao **kvadratno odstupanje**, a empirijsku pogrešku kao **sumu razlike reziduala**. Zašto?

U označenim podatcima neminovno postoji **šum**, koji se superponira na vrijednost funkcije.

vrijednost funkcije + šum

* + Šum modeliramo kao slučajno varijablu iz normalne distribucije
  + Gustoća normalne razdiobe

Za zadani , izlazna vrijednost biti će distribuirana oko vrijednosti

Vjerojatnost da je skup primjera X označen oznakama y jednaka je:

Pretpostavka je da su oznake primjera **nezavisno i identično distribuirane.**

Nalazimo hipotezu koja **maksimizira vjerojatnost oznaka**. Želimo da oni primjeri koje smo dobili budu najvjerojatniji.

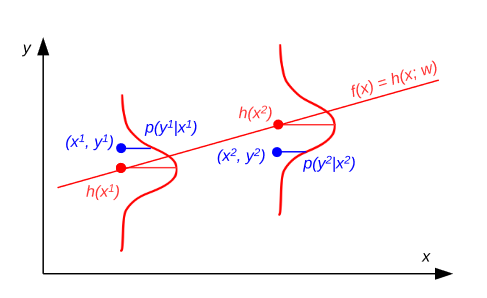
konstanta

Nas zanima član nakon sume, pošto je prvi član konstanta i derivacija mu je 0 jer deriviramo ovu vjerojatnost po parametru . Ostaje nam negativni konstantni član i suma :

Pošto je cijeli izraz negativan, a mi ga želimo maksimizirati, zapravo ćemo minimizirati izraz pod sumom. Taj izraz je upravo **pogreška kvadratnog odstupanja**. Drugim riječima, uz pretpostavku normalno distribuiranog šuma, hipoteza koja će maksimizirati vjerojatnost da su podatci označeni tako kako jesu je upravo ona koja minimizira kvadratno odstupanje predviđenih i stvarnih oznaka.

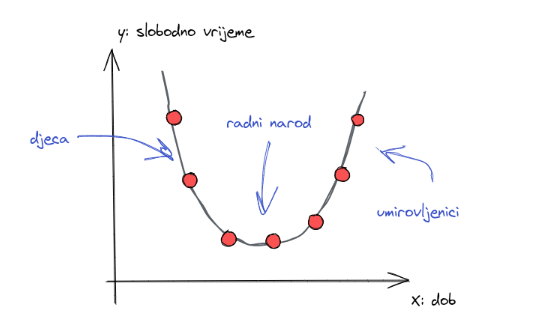
**DODATNO**

**L2-norma** (druga norma, euklidska norma) vektora je . Suma pogreške upravo je kvadrat L2-norme vektora



Vrijedi da je za svaku matricu.

# **LINEARNA REGRESIJA II.**



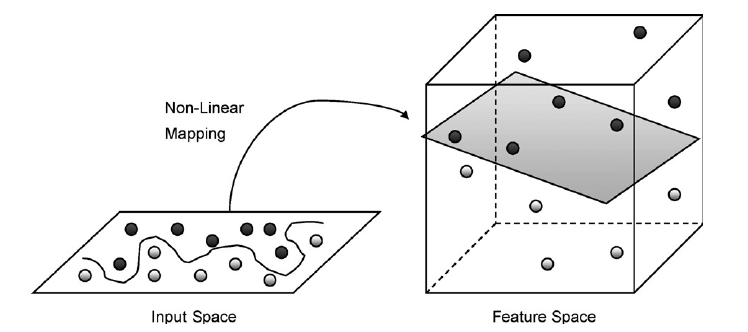
## **Nelinearna regresija**

**Interakcijska značajka** (cross-term)

Model npr. 2. stupnja

* Linearna višestruka regresija:
* Jednostruka (jednostavna) polinomijalna regresija (n=1):
* Višestruka polinomijalna regresija (n=2, d=2):

Unifikacija – različite modele regresije **unificirat** ćemo preslikavanjem primjera iz **ulaznog prostora** u **prostor značajki.** Kada želimo preći s linearne regresije u nelinearnu regresiju, nećemo mijenjati model, već ćemo mijenjati **podatke**.Ideja je da će podatci, iako nisu bili linearni u ulaznom prostoru, u prostoru značajki biti linearni.



**DODATNO**

U jeziku numeriˇcke matematike, uporaba polinomijalne funkcije za modeliranje nelienarnosti funkcije

hpx; wq jest problem polinomijalne interpolacije. Jedna alternativa polinomijalnoj interpretaciji, a

koja se vrlo ˇcesto koristi u statistici, jest interpolacija pomo´cu spline funkcija. Spline funkcija

je funkcija koja je definirana po dijelovima pomo´cu funkcije polinoma. Interpolacija pomo´cu spline

funkcije daje sliˇcne rezultate kao i polinomijalna interpolacija, ali je rezultat stabilniji, u smislu da, za

razliku od polinomijalne funkcije, spline funkcije ne dovode do oscilacija pri rubovima interpolacijskog

intervala (tzv. Rungeov fenomen), ˇsto je u stvari manifestacija prenauˇcenosti modela. U statistici

se dominantno koriste ograniˇcene (prirodne) kubne spline funkcije (engl. restricted (natural) cubic

splines), koje su po dijelovima definirane kao polinom tre´ceg stupnja, uz dodatnu ogradu da je

spline funkcija linearna prije prvog segmenta i na kraju zadnjeg segmenta.