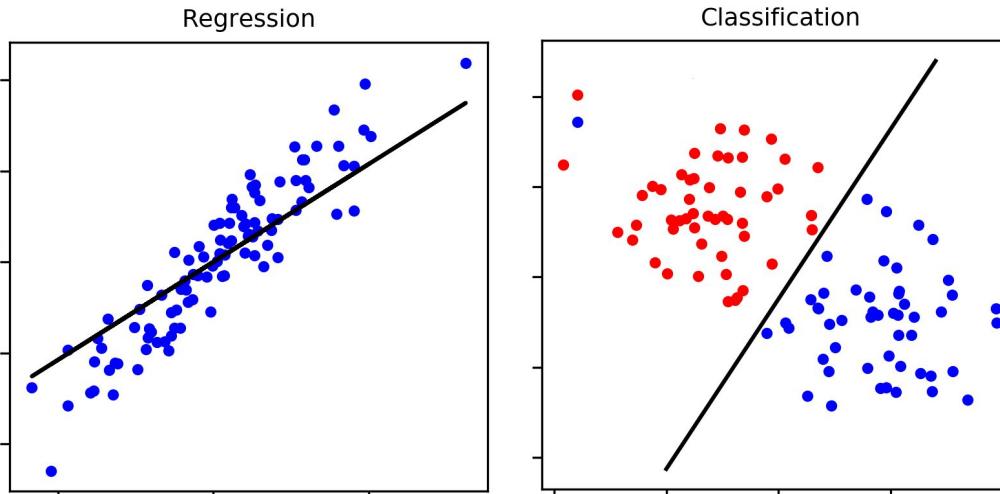


Logistička regresija i SVM

— Logistička regresija, Gradient
descent, SVM, Validacija —

Logistička regresija

- Algoritam za klasifikaciju
- Statistička metoda, binarna klasifikacija



Kako pretvoriti realnu vrijednost u klasu?

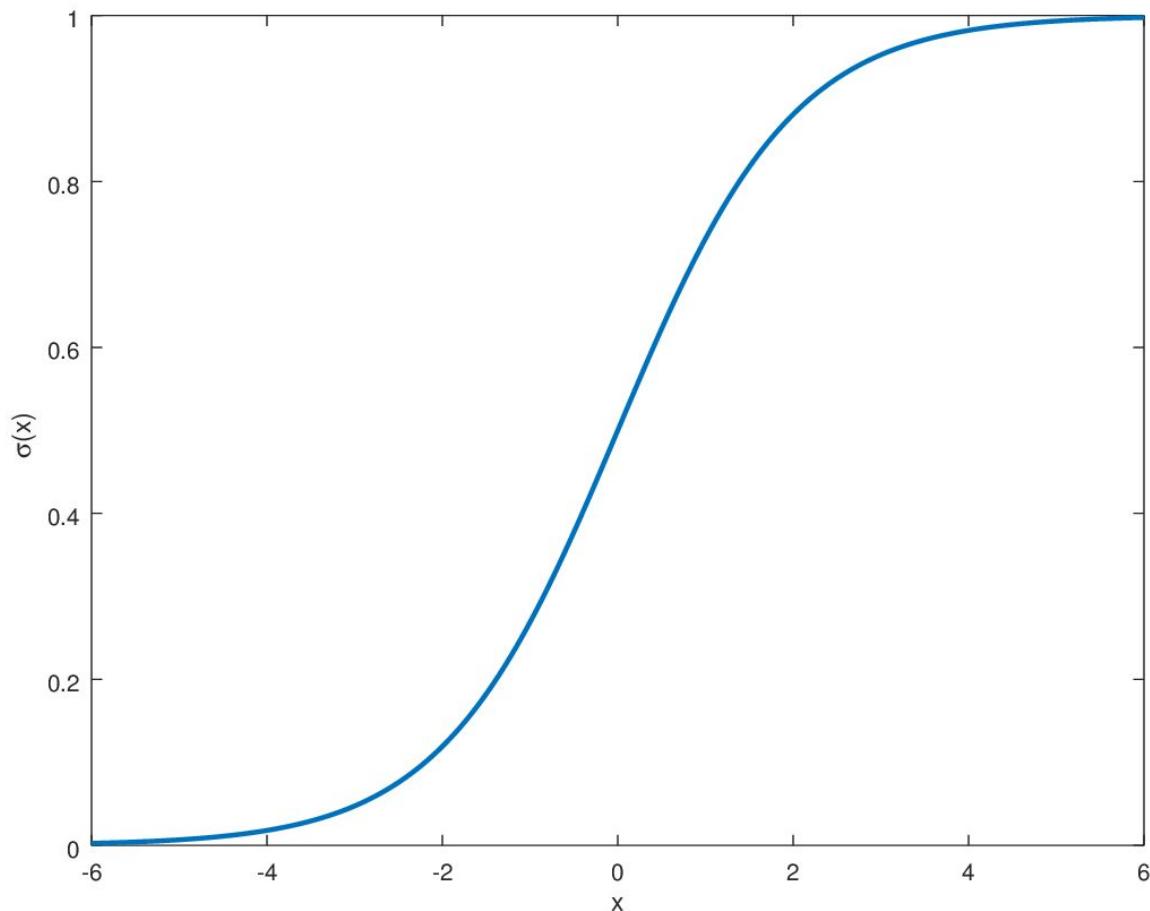
- Regresija = na izlazu realan broj
- Potrebno realan broj pretvoriti u klasu kako bismo jer rješavamo problem klasifikacije
- Sigmoid funkcija - pretvara realan broj u vrijednost koja se nalazi između 0 i 1
- Ako je vrijednost veća od 0.5 onda pripada klasi 1 u suprotnom pripada klasi 0
- Tehnika koju koriste i neuronske mreže

Sigmoid funkcija

- $\text{sigmoid}(x) = 1 / (1 + \exp(-x))$
- Domen funkcije od $-\infty$ do $+\infty$
- Funkcija diferencijabilna - jako bitno
- Model logističke regresije zasnovan na formuli:

$$P(y=1 \mid x) = \text{sigmoid}(w^T x + b)$$

- w i b težine koje treba obučiti (kao kod linearne regresije)
- Korisna osobina - numerčki dovoljno da računamo za ulaze između -10 i 10 , inače teži 0 odnosno 1



Funkcija greške

- Ne možemo više da koristimo MSE, ne optimizujemo realnu vrijednost već klasu koju dobijemo na osnovu realne vrijednosti
- Ideja = za ulaze koji pripadaju klasi 1, želimo da maksimizujemo vrijednost koju generiše sigmoid funkcija
- Maximum Likelihood Estimation (MLE) - maksimizacija sljedeće funkcije: $L(w, b) = \prod [P(y=1 | x)^y * (1 - P(y=1 | x))^{1-y}]$
- Intuitivno - MLE mjeri koliko klase pravilno pogađamo

Logaritam greške

- Obično se ne koristi osnovna formula za MLE već se koristi njen logaritam:

$$LL(w, b) = \sum [y * \log(P(y=1 | x)) + (1-y) * \log(1 - P(y=1 | x))]$$

- Logaritam uvodi brojne prednosti:
 - suma uvodi numeričku stabilnost i lakše računanje (underflow kod malih i overflow kod velikih vrijednosti)
 - monotono rastuća funkcija
 - gubi se eksponent zbog algoritma
- Maksimizacija LL ili minimizacija od -LL (u ML uvijek minimizacija)

Gradient descent

- Spuštanje niz gradijent - jedan od glavnih algoritama u mašinskom učenju
- Želimo da minimizujemo funkciju greške
- Idealno - nađemo nulu prvog izvoda, često nemoguće zbog kompleksnosti funkcija
- Šta je prvi izvod? - brzina promjene funkcije, da li funkcija raste ili opada
- Prvi izvod 0 - globalni minimum ili maksimum
- Kretanje suprotno od pravca prvog izvoda - kretanje ka minimumu

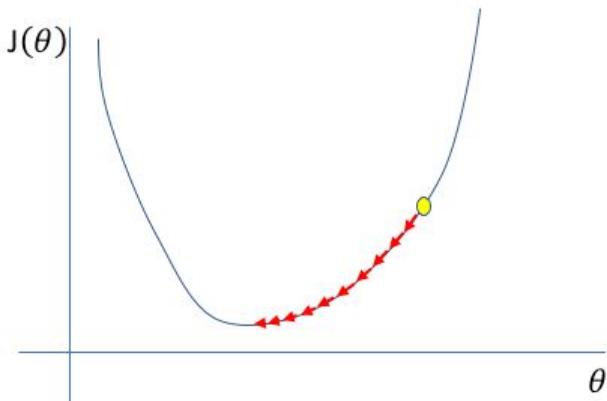
Gradient descent

- Koraci:
 - inicijalizujemo model sa nasumičnim parametrima w i b
 - izračunamo vrijednost gradijenta funkcije greške za date parametre
 - promijenimo parametre tako da se krećemo u pravcu suprotnom od gradijenta
- $w = w - \alpha * \nabla f(w)$ i $b = b - \alpha * \nabla f(b)$
- Nakon određenog broja koraka dolazimo blizu minimuma
- Bitno - moramo znati funkciju izvoda

Faktor obučavanja

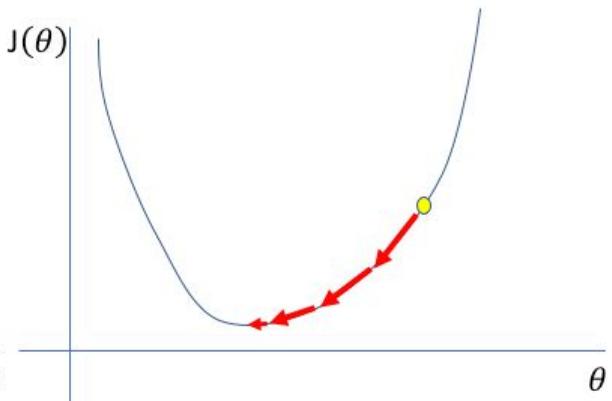
- Faktor obučavanja alfa određuje koliko brzo će se mijenjati vrijednosti naših parametara
- Premalen broj - prespora konvergencija, potencijalno nikad nećemo doći do minimuma
- Prevelik broj - prebrza konvergencija, stalno ćemo preskakati minimum
- Eksperimentalno određujemo najbolju vrijednost
- Postoje neke heuristike za dobre vrijednosti faktora u zavisnosti od algoritma

Too low



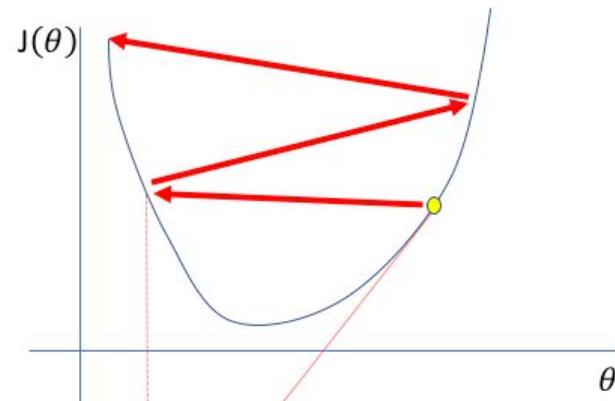
A small learning rate requires many updates before reaching the minimum point

Just right



The optimal learning rate swiftly reaches the minimum point

Too high



Too large of a learning rate causes drastic updates which lead to divergent behaviors

Bitna pitanja?

- Koji su problemi sa logističkom regresijom?
- Postoje li neki slučajevi kad logistička regresija neće raditi?
- Kako bismo rješili te slučajeve?

Odgovori

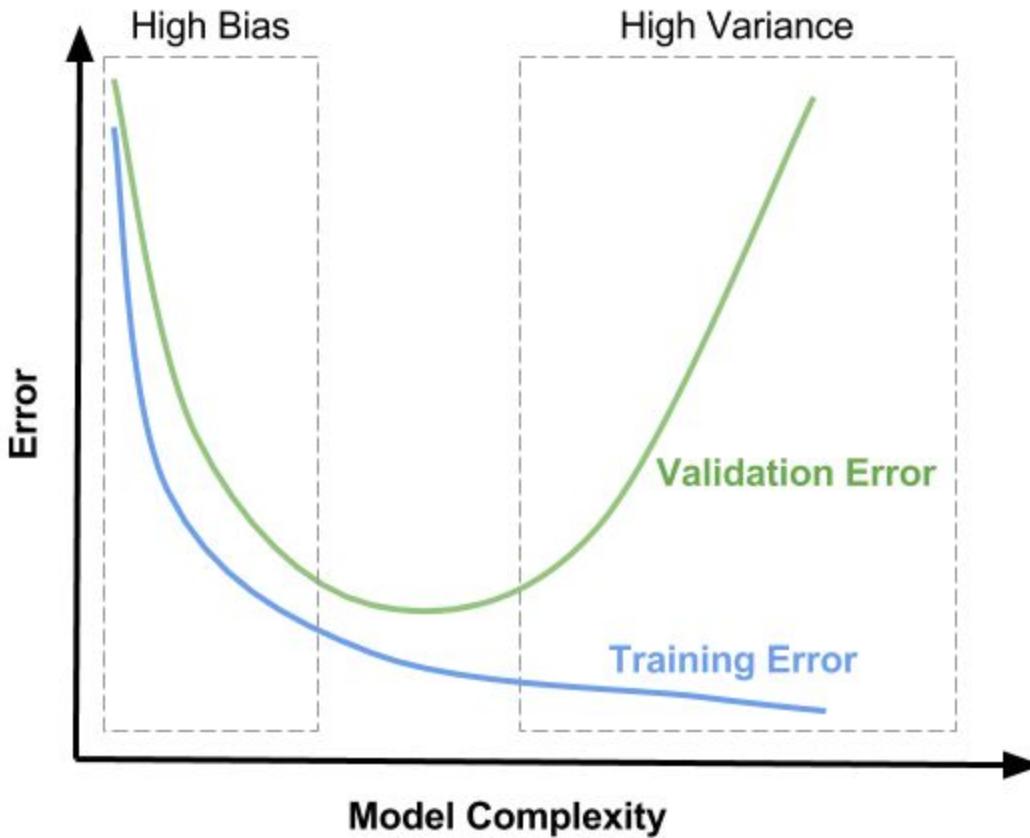
- Koji su problemi sa logističkom regresijom?
 - Rigidan linearni separator
 - Velika osjetljivost na outliere i granične slučajeve
- Postoje li neki slučajevi kad logistička regresija neće raditi?
 - Problemi koji nisu linearno separabilni (problem krofne)
 - Visoko dimenzioni problemi sa malo uzoraka (moguće ali jako loše performanse)

Kako rješiti prethodne probleme?

- Osnovna ideja - margina, a ne pojedinačna tačka
 - jedan outlier u logističkoj regresiji će značajno pomjeriti granicu odlučivanja
 - fokus na granične tačke - najproblematičnije - ako datu tačku udaljimo od granice postajemo sigurniji
- Ne želimo da gubitku previše doprinose tačke za koje imamo visok stepen sigurnosti da pripadaju nekoj klasi (daleko od margine)
- SVM algoritam - fokusira se na optimizaciju margine

Pretraga hiperparametara

- Većina ozbiljnijih algoritama ima veliki broj parametara koje treba isprobati da bismo vidjeli koja kombinacija daje najbolje performanse na našem skupu podataka
- Validacioni skup podataka - specijalni skup podataka koji se koristi za pretragu hiperparametara
- trening skup + testni skup + validacioni skup
- Validacioni skup se koristi i da spriječi overfitting - treniramo model sve dok performanse na validacionom skupu ne počnu da opadaju

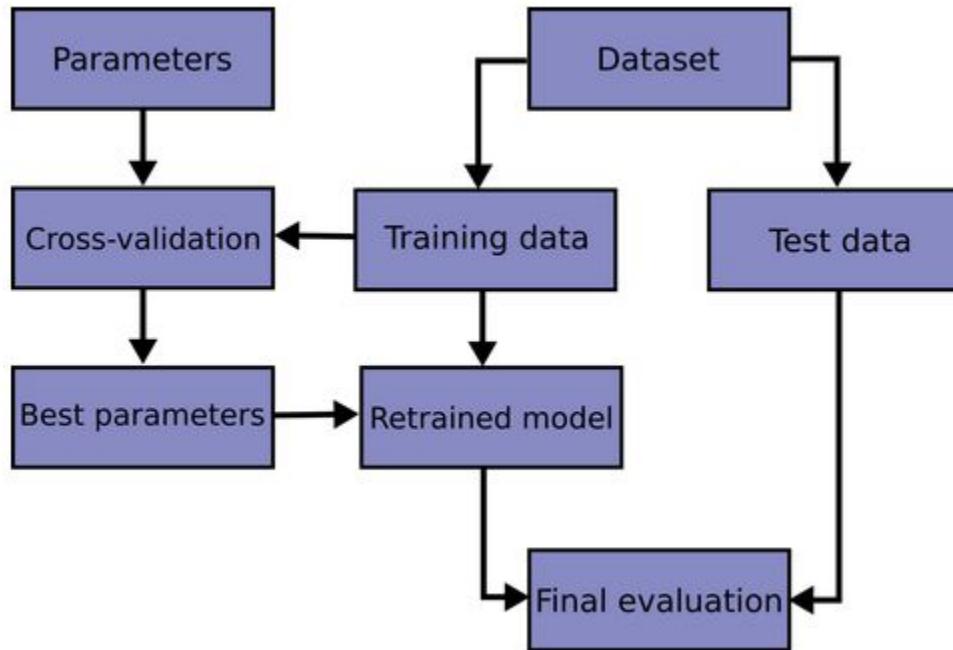


Proces pretrage hiperparametara

- Izaberemo kombinaciju vrijednosti
- Treniramo podatke upotrebom trening skupa
- Nakon svake iteracije provjeravamo i performanse na validacionom skupu, ukoliko počnu da opadaju prekidamo trening i pamtimo performanse na validacionom skupu
- Biramo kombinaciju parametara koja ima najbolje performanse na validacionom skupu
- Testiramo na testnom skupu da dobijemo konačne metrike

Cross validation

- Drugi pristup validaciji
- Nemamo više validacioni skup podataka
- Trening skup se dijeli na k dijelova (k folds)
- Model se trenira sa $k-1$ dijelova i na kraju se validira preostalim dijelom
- Ukupna performansa - srednja vrijednost performansi za svaki od foldova
- Nema "gubitka" podataka, ali dodatna računska kompleksnost



Kompromisi

- Idealno - da pretražimo što veći broj hiperparametara
- Problem - vrijeme izvršavanja
- Heuristički odabrat bitne metrike - Google i iskustvo
- Pokrenuti na sa grubim razmacima pa sužavati prostor pretrage na bolje odabrane vrijednosti - moguće manuelno ali postoje i gotovi algoritmi
- Bitno - validation i testni set su različiti

Zaključak

- Logistička regresija
- Sigmoid funkcija - transformacija realne vrijednosti u vrijednost između 0 i 1
- Funkcija greške i MLE
- Gradijentni spust i faktor obučavanja
- SVM algoritam
- Validacioni skup i pretraga hiperparametara
- Cross validation