Validacioni postupci

- Potreba za procenom mere greške sistema za mašinsko učenje
  - Izbor optimalnog modela
  - Evaluacija uspešnosti modela
- Validacioni postupci
  - Stvarna mera greške i njena estimacija
    - Estimacija resupstitucijom
    - Estimacija na nezavisnom skupu za testiranje
    - Estimacija ponovnim uzorkovanjem
- Trostruka podela podataka

Validacioni postupci

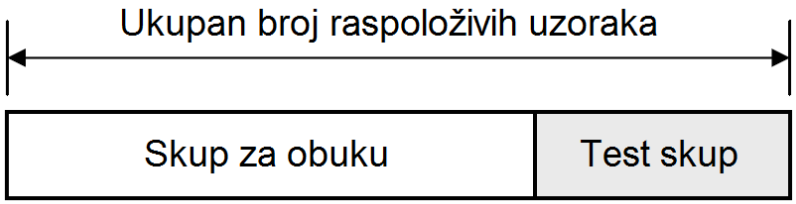
- Pri projektovanju sistema za klasifikaciju ili regresiju u praksi se javljaju sledeći problemi:
  - **Validacija modela**
    - Kako proceniti uspešnost modela (kao mera greške kod klasifikacije posmatra se **mera pogrešne klasifikacije**, a kod regresije obično **srednja kvadratna greška**)
  - **Selekcija modela**
    - Kako, za dati problem, odabrati optimalne vrednosti slobodnih parametara (npr. broj slojeva i neurona neuralne mreže, parametri Gaussove raspodele...)
- Evaluacija (validacija) sistema je *sastavni deo* procedure projektovanja
- *Stvarna mera greške* bila bi mera greške na celokupnoj populaciji
  - Kad bismo imali neograničenu populaciju i neograničeno vreme:
    - Odabrali bismo model koji daje najmanju meru greške na celokupnoj populaciji
    - Ta mera greške ujedno bi predstavljala stvarnu meru greške
  - Međutim, u praksi imamo na raspolaganju **konačan broj uzoraka**, i to često manji nego što bismo želeli (ovo je i jedan od najkritičnijih problema mašinskog učenja)
    - Proces prikupljanja podataka je izuzetno skup (i materijalno i vremenski)

Procena greške resupstitucijom

- Zašto se podaci za obuku ne bi iskoristili i za procenu stvarne mere greške, kao i za izbor optimalnog modela?
  Ovaj jednostavan pristup ima dva osnovna problema:
  - Krajnji model će biti natprilagođen podacima za obuku (*overfitting*) i neće imati dovoljnu sposobnost generalizovanja svog odlučivanja za nove podatke
    - Problem natprilagođenja je izraženiji kod složenijih modela, koji zavise od mnogo parametara
  - Estimacija stvarne mere greške biće suviše optimistična (niža od stvarne mere greške)
    - Nije retka pojava da se npr. kod klasifikacije postigne 100% tačnost na podacima iz skupa za obuku, ali to ne odslikava realnu situaciju
- Pitanje je kako na pravi način iskoristiti ograničen skup podataka za:
  - Obuku
  - Validaciju i selekciju modela
  - Procenu stvarnih performansi odabranog modela

Procena greške na nezavisnom test skupu

- Kod ovog metoda (eng. *holdout)* skup raspoloživih podataka deli se na dva disjunktna podskupa
  - Skup za obuku – koristi se za obuku modela
  - Skup za testiranje – služi za procenu stvarne mere greške obučenog modela



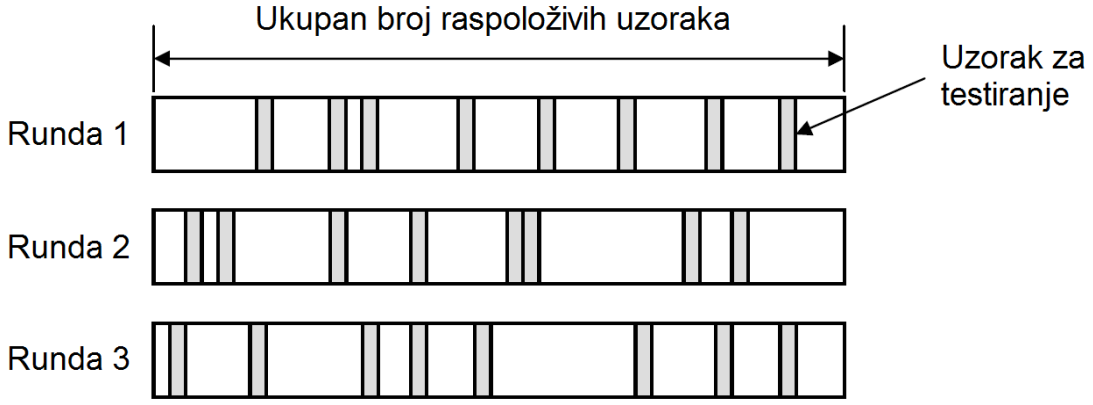
- *Holdout* metod ima dva osnovna nedostatka
  - Ako raspoloživih podataka ima malo, može biti problem izdvojiti deo tog skupa isključivo za testiranje i time dodatno smanjiti skup za obuku
  - Ako se dogodi da je podela na skupove za obuku i testiranje loša, procena mere greške će se znatno razlikovati od stvarne vrednosti
- Ova ograničenja se mogu prevazići metodama ponovnog uzorkovanja podataka
  - Svi raspoloživi podaci se na neki način koriste i za obuku i za testiranje, ali na način koji je ujedno ekonomičan i ne ugrožava procenu stvarne mere greške

Procena greške ponovnim uzorkovanjem

- Svi raspoloživi podaci se, u opštem slučaju, koriste i za obuku i za testiranje
- Dve osnovne grupe metoda:
  - Unakrsna validacija
    - Jedna *runda* unakrsne validacije obuhvata:
      - podelu skupa raspoloživih uzoraka na dva komplementarna podskupa (za obuku i testiranje)
      - obuke korišćenjem podskupa za obuku i validacije na podskupu za testiranje
    - Procena performansi vrši se usrednjavanjem procena dobijenih u okviru više rundi sa različitim podelama skupa raspoloživih uzoraka
    - U zavisnosti od načina podele razlikuje se nekoliko pristupa:
      - Stohastičko poduzorkovanje
      - Unakrsna validacija sa *K* particija
      - Unakrsna validacija sa jednim izdvojenim elementom (eng. *leave-one-out*)
  - *Bootstrap*
    - Ponovno uzorkovanje *sa vraćanjem*
      - Zahvaljujući vraćanju, uzorkovanjem se ne narušavaju postojeće statističke zakonitosti u skupu za obuku (npr. kod klasifikacije se tokom čitavog procesa selekcije uzoraka očuvavaju apriorne verovatnoće pojedinih klasa)

Stohastičko poduzorkovanje

- U svakoj rundi se *na slučajan način* bira fiksan broj uzoraka za test skup
  - Za svaku podelu model se iznova obučava pomoću uzoraka iz skupa za obuku, a zatim se stvarna mera greške estimira na uzorcima iz test skupa



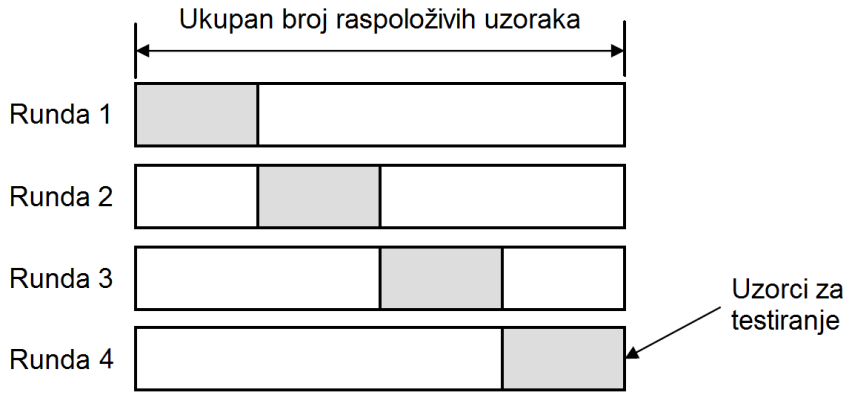
- Konačna estimacija mere greške se dobija kao prosečna vrednost estimacija $\hat{E}_i$

$$\hat{E} = \frac{1}{K}\sum_{i=1}^{K} E_i$$

  - Ova estimacija je značajno bolja od *holdout* estimacije

Unakrsna validacija sa *K* particija

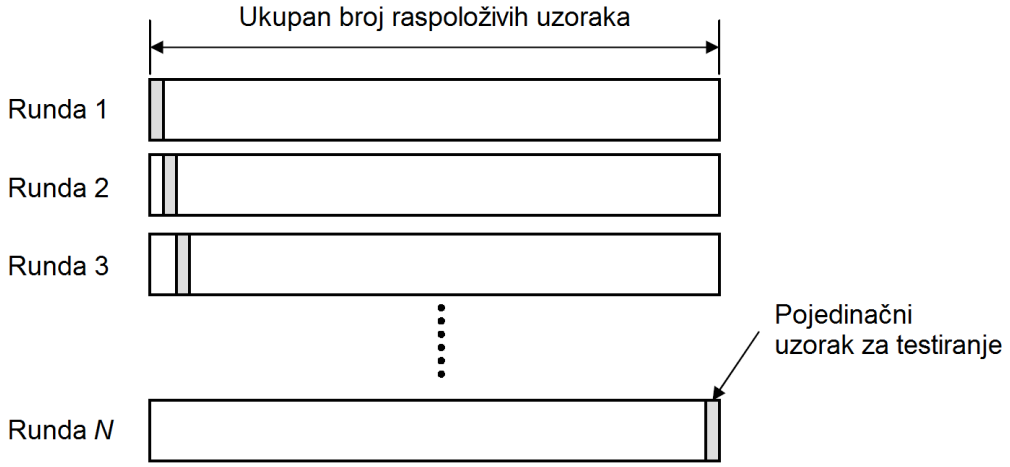
- Skup raspoloživih uzoraka deli se na *K* (približno) jednakih podskupova
- U svakoj rundi jedan od podskupova izdvaja se za testiranje, a obuka se vrši na preostalim podskupovima



- Konačna estimacija mere greške je prosečna vrednost estimacija $\hat{E}_i$
  - Prednost u odnosu na prethodni metod je što će ovde svaki uzorak u određenom trenutku biti iskorišćen i za obuku i za testiranje
  - S porastom *K* povećava se tačnost estimacije, ali i količina izračunavanja
  - Za veliko *N* čak i vrlo mali broj particija je dovoljan (npr. $K = 3$), a u praksi je najčešće dovoljno odabrati (makar za početak) $K = 10$

Unakrsna validacija sa jednim izdvojenim elementom

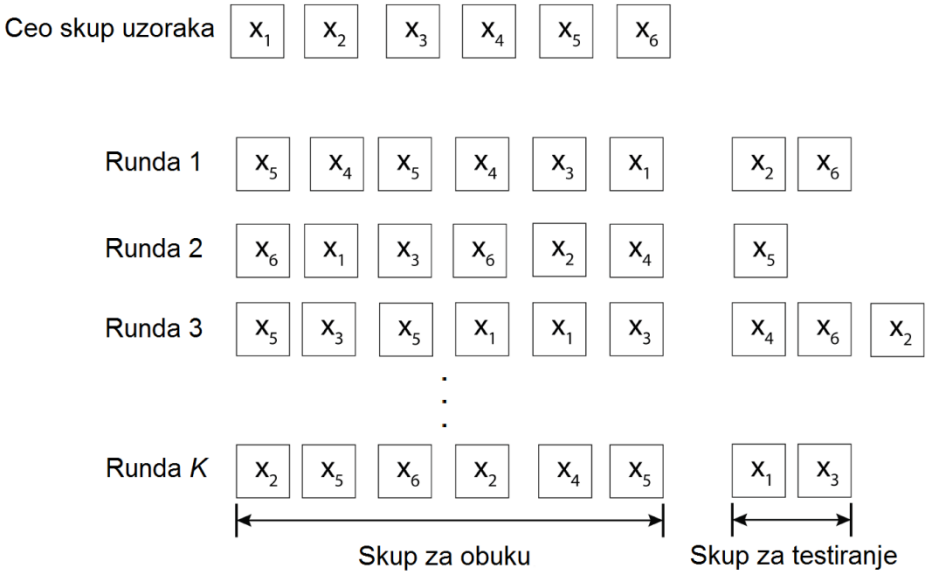
- U svakoj rundi jedan uzorak izdvaja se za testiranje (*leave-one-out*), a obuka se vrši na preostalim uzorcima



- Konačna estimacija mere greške je prosečna vrednost estimacija $\hat{E}_i$
- Najopštija tehnika, koja maksimalno koristi skup dostupnih uzoraka te stoga ima i minimalnu šansu za natprilagođenje
  - Mana ove tehnike je izuzetno visoka računska kompleksnost
    - Postoje brojne tehnike kojima je cilj da se ova računska kompleksnost smanji, a da se ne izgubi mnogo na tačnosti estimacije

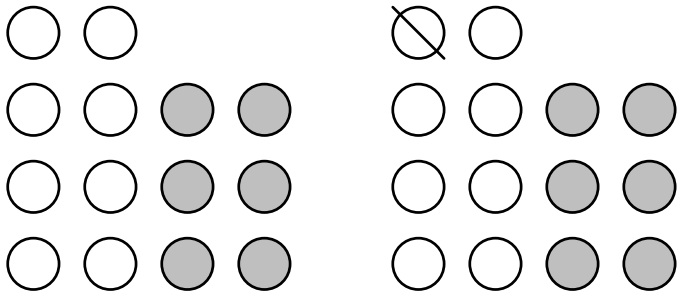
Bootstrap metod

- Metod ponovnog uzorkovanja podataka sa vraćanjem
  - U svakoj od *K* rundi od celokupnog skupa za obuku koji ima *N* elemenata, bira ih se *N*, ali sa vraćanjem
    - Očigledno, neki od uzoraka naći će se više puta u skupu za obuku u svakoj rundi
  - Test skup se u svakoj rundi formira od uzoraka koji nisu nijednom odabrani
    - Veličina test skupa neće biti ista u svakoj rundi
  - Konačna estimacija mere greške je prosečna vrednost estimacija $\hat{E}_i$



Bootstrap metod

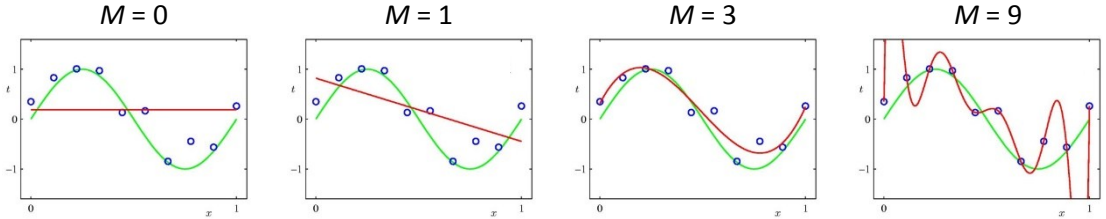
- Da nema vraćanja uzoraka, izvlačenje konkretnog uzorka u određenoj meri bi narušilo postojeće statističke zakonitosti u skupu za obuku
  - Npr. kod klasifikacije bi se izvlačenjem određenog uzorka iz skupa za obuku narušile apriorne verovatnoće pojedinih klasa



    - Neka se radi o klasifikacionom problemu sa C klasa, sa ukupno N uzoraka od kojih po N_i pripada svakoj klasi ω_i
    - Apriorna verovatnoća izbora uzorka iz određene klase ω_i je N_i/N
    - Ako se izvuče uzorak klase ω_i i ne vrati se u skup za novo izvlačenje (a ne vraća se npr. kod unakrsne validacije), apriorna verovatnoća će se promeniti i iznosiće $(N_i - 1)/(N - 1)$
- *Bootstrap* metod značajan je i sa teorijskog stanovišta
  - omogućuje i procenu centriranosti i varijanse estimacije stvarne mere greške

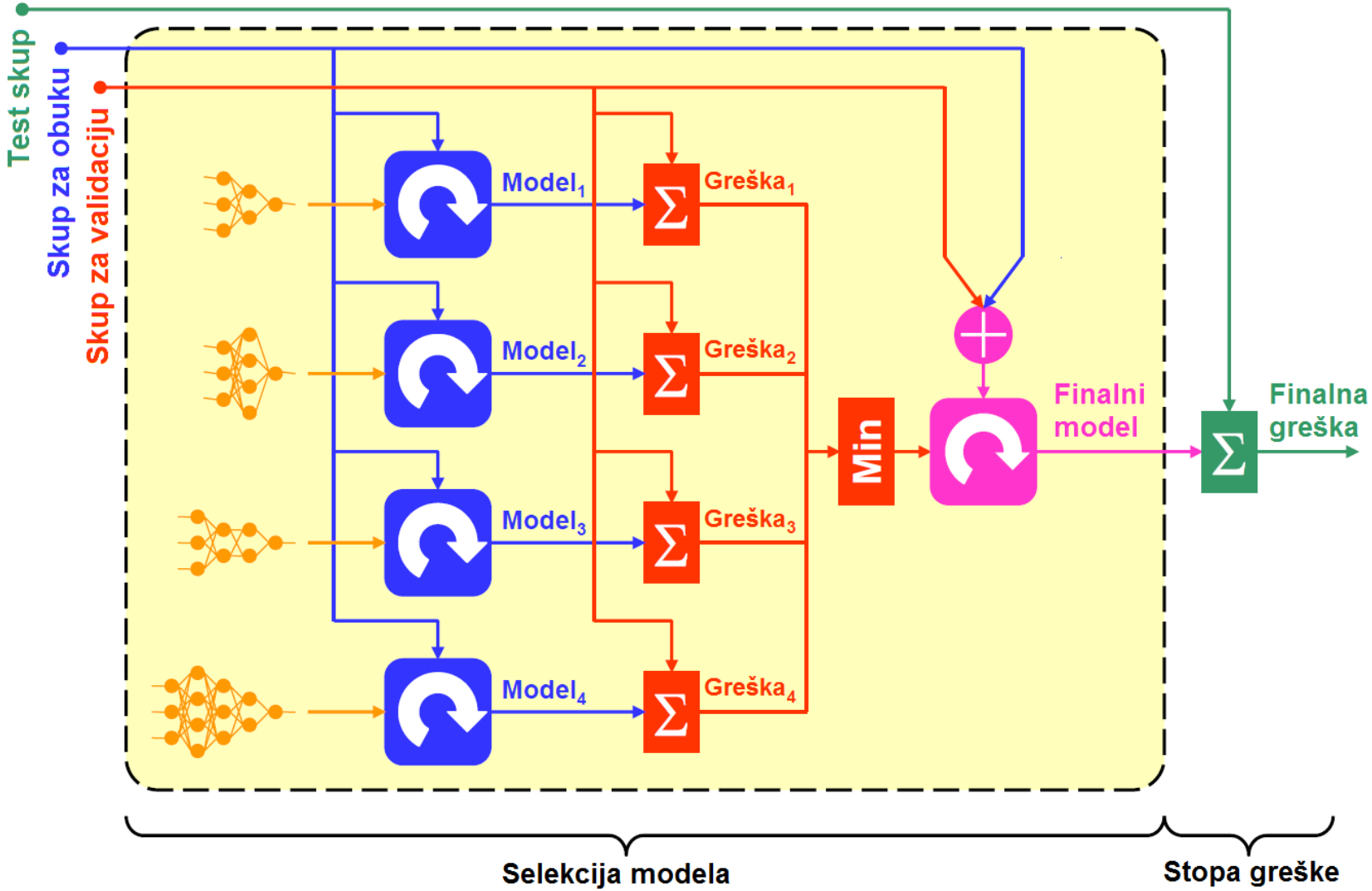
Trostruka podela podataka

- Ako treba istovremeno sprovesti selekciju modela i estimaciju stvarne mere greške, raspoloživi skup podataka treba podeliti na tri disjunktna skupa
  - **Skup za obuku** – skup uzoraka koji se koristi za formiranje modela
    - Kod linearne regresije polinomijalnom aproksimacijom, za *dati red polinoma*, skup za obuku se koristi za nalaženje optimalnih vrednosti koeficijenata polinoma



  - **Skup za validaciju** – skup uzoraka koji se koristi za podešavanje slobodnih parametara modela
    - Kod linearne regresije, skup za validaciju se koristi za testiranje više modela sa različitim redom polinoma, i na osnovu toga bira se najuspešniji
  - **Skup za testiranje** – skup uzoraka koji se koristi samo za procenu performansi odabranog modela
    - Kod linearne regresije, skup uzoraka za testiranje bi se koristio za estimaciju stvarne mere greške nakon izbora konačnog modela (sa optimalnim redom polinoma)
    - Nakon procene konačnog modela na test skupu, model se *ne sme* dalje podešavati!

Trostruka podela podataka



Trostruka podela podataka

- Zašto treba razdvojiti skupove za testiranje i validaciju?
  - Estimacija stvarne mere greške konačnog modela dobijena na podacima za validaciju biće necentrirana (manja od stvarne mere greške), pošto se sâm skup podataka za validaciju koristi za izbor krajnjeg modela (skup nije objektivan)
- Procedura trostruke podele podataka predviđa sledeće korake:

1. Podeliti raspoloživi skup podataka na skupove za obuku, validaciju i testiranje
2. Izabrati modele i vrednosti njihovih slobodnih parametara
3. Obučiti svaki model koristeći skup podataka za obuku
4. Evaluirati svaki model koristeći skup podataka za validaciju
5. Izabrati najbolji model kao onaj sa najmanjom greškom na skupu za validaciju
6. Obučiti izabrani model koristeći podatke iz skupova za obuku i validaciju
7. Estimirati performanse konačnog modela koristeći skup podataka za testiranje

- Procedura je izložena za slučaj da se primenjuje *holdout* metoda za procenu greške
  - U slučaju korišćenja unakrsne validacije ili *bootstrap* metode, korake 3. i 4. treba ponoviti za svaku od *K* rundi