Ponovno uzorkovanje (Resampling methods)

Metode ponovnog uzorkovanja su nezaobilazan alat u savremenoj statistici. Oni uključuju ponovno uzimanje uzoraka iz trening skupa i ponovno treniranje našeg modela na svakom uzorku kako bi se dobile dodatne informacije o ugrađenom modelu. *Na primer, da bi se procenila varijabilnost modela linearne regresije, možemo više puta da uzimamo različite uzorke iz trening podataka, da prilagodimo linearnu regresiju svakom novom uzorku, a zatim da ispitamo u kojoj se meri dobijeni rezultati razlikuju.*

Pristupi ponovnog uzorkovanja mogu biti računski skupi, jer uključuju ponovno treniranje (eng. *fitting*) iste statističke metode više puta koristeći različite podskupove podataka za trening. Međutim, zbog nedavnog napretka u računarskoj snazi, metode ponovnog uzorkovanja se dosta koriste. To su tehnike gde **više puta "simuliramo nove skupove podataka" iz postojećeg**, da bismo videli: *koliko su naši rezultati stabilni i koliko možemo da im verujemo.*

Pomoću ovih metoda možemo uporediti različite algoritme (npr. Random Forest protiv SVM-a) ili različite hiperparametre unutar istog algoritma kako bismo utvrdili koji najbolje rešava konkretan problem.

Rešavanje nebalansiranih klasa

U situacijama kada jedna klasa dominira (npr. detekcija prevara gde je 99% transakcija legitimno), *resampling* se koristi za veštačko uravnotežavanje:

- **Oversampling**: Dupliranje primera iz manjinske klase (npr. SMOTE algoritam).
- **Undersampling**: Brisanje primera iz većinske klase.

Pomoću ovih metoda proveravamo tri stvari:

1. **Generalizaciju modela (**"*Koko će model raditi u realnosti?*"**)**
   - Da li model radi samo na trening skupu ili i na novim podacima?
   - Da li imamo *overfitting*?
   - (tipično: K-fold)
2. **Varijansu / stabilnost (***"Da li je model robustan ili 'osetljiv'?"***)**
   - Da li mali pomeraji u podacima menjaju rezultat drastično?
   - (tipično: bootstrap)
3. **Nesigurnost procene (***"Koliko je ovo tačno ± greška?"***)**
   - Koliko su metrike (RMSE, accuracy…) pouzdane?
   - Koliki je interval poverenja?

1. Unakrsna validacija (Cross-validation)

Test greška meri prosečno odstupanje predikcija modela na novim, neviđenim podacima. U kontekstu datog skupa podataka, metoda učenja se smatra adekvatnom ukoliko rezultuje niskom vrednošću test greške.

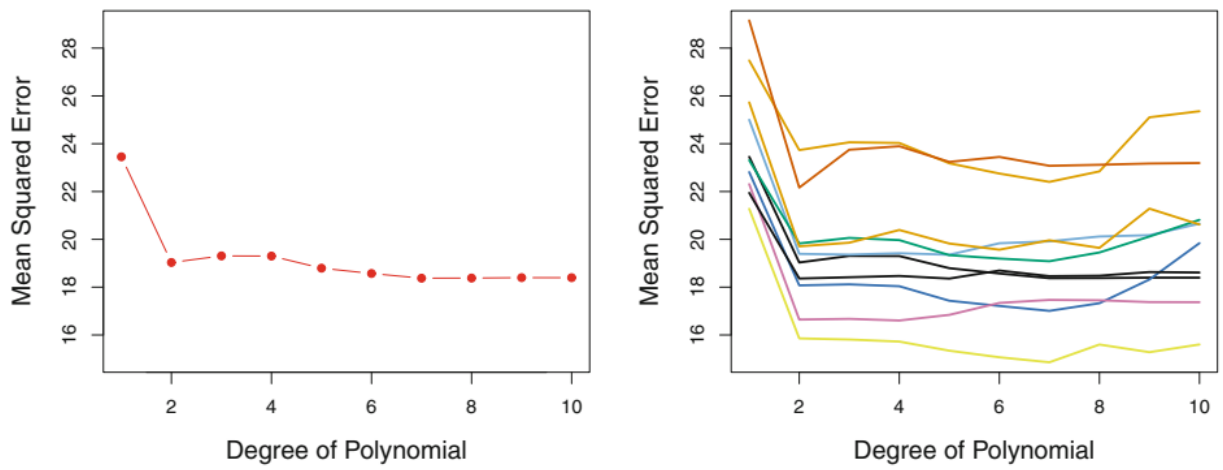
Pomoću unakrsne validacije proveravamo **sposobnost generalizacije modela**. Njen fokus je na proceni greške predviđanja na novim, neviđenim podacima.

Kada se koristi: Kada želite da procenite koliko će vaš model biti precizan u realnom svetu. Standard je za **evaluaciju performansi** i **tuning hiperparametara**.

a) Validacioni skup podataka

train-validation-test: 60%:20%:20% (za duboko učenje se koriste drugačiji odnosi: 98%:1%:1%)

Skup podataka Auto (ISLR), mpg~horsepower



Slika 1. Korišćenje validacionog skupa kako bi se procenila greška na test skupu, a koja je rezultat predviđanja *mpg* promenljive korišćenjem polinomskih funkcija za konjske snage motora.
Levo: Procene grešaka na validacionom skupu za jednu podelu na trening i validacioni skup.
Desno: Metoda validacije je ponovljena deset puta, svaki put koristeći različiti različitu podelu na trening i validacioni skup. Ovo ilustruje varijabilnost procenjenog testnog MSE koja proizilazi iz ovog pristupa.

Korišćenje validacionog skupa je konceptualno jednostavno i lako za primenu. Ali ima dva potencijalna nedostatka:

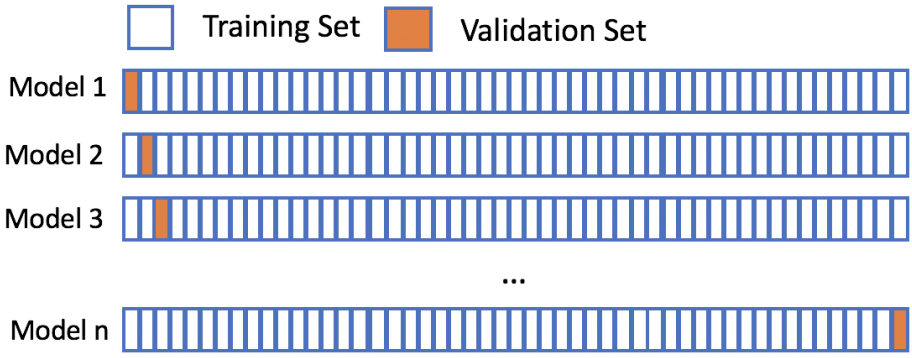
- Procena test greške je veoma promenljiva, u zavisnosti od toga koje observacije su uključene u trening skup, a koje observacije su uključene u validacioni skup.
- Za ovaj pristup važi da se samo podskup observacija koristi se za uklapanje u model (one koji su uključene samo u trening skup, a nisu u validacionom skupu). Budući da statističke metode teže da rade lošije kada se obučavaju na manje observacija, ovo sugeriše da stopa grešaka u validacionom skupu može da poveća (eng. *overestimate*) test grešku u odnosu na model koji se trenira na celom skupu podataka.

b) *Leave-One-Out unakrsna validacija (LOOCV)*

$$MSE_i = (y_i - \hat{y}_i)^2, i = 1, \dots, n$$

Procena LOOCV za test MSE je prosek od svih n procena greške na validacionim skupovima:

$$CV(n) = \frac{1}{n}\sum_{i=1}^{n} MSE_i$$



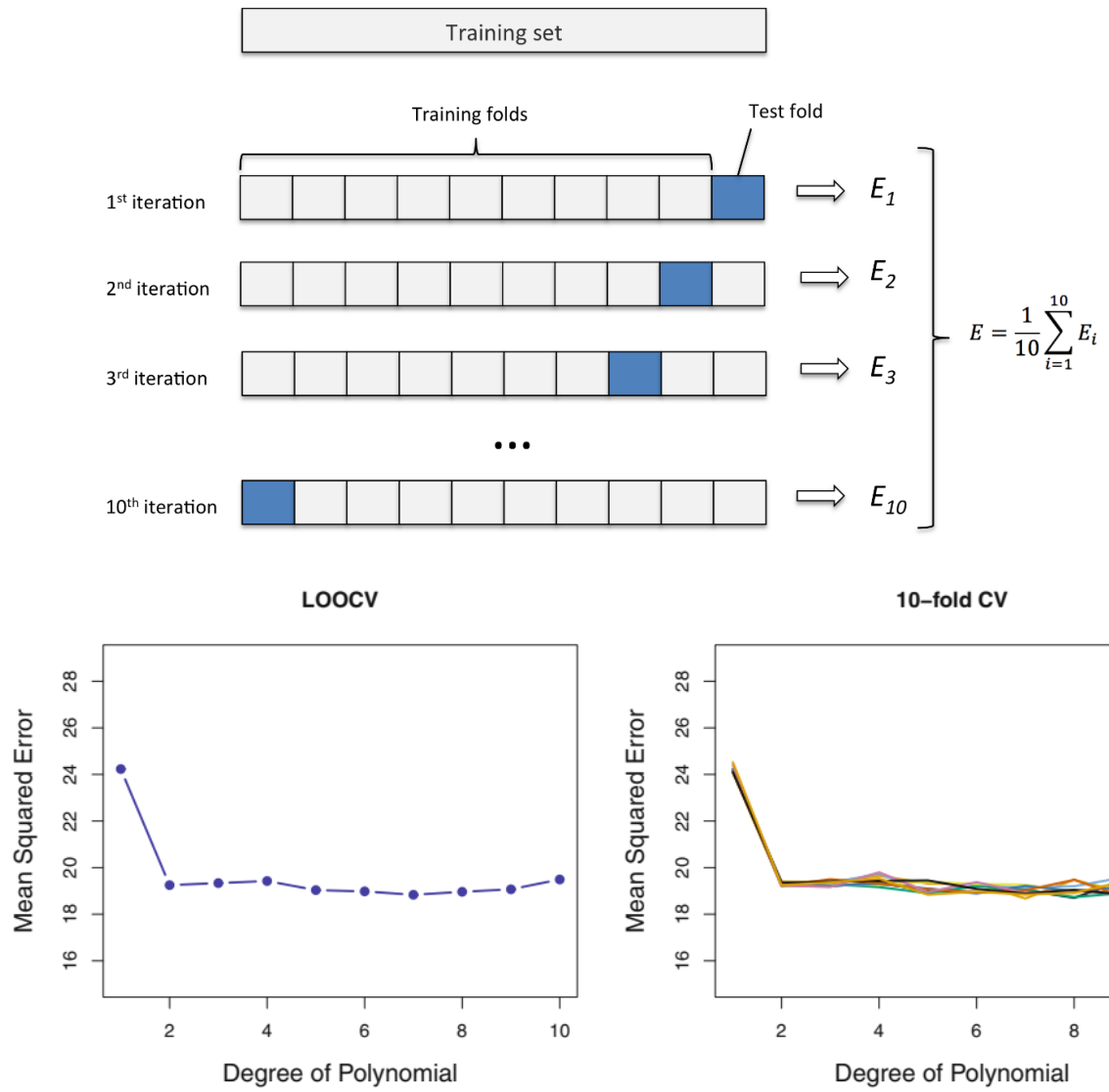
Prednosti:

- Ima daleko manji *bias*. LOOCV pristup teži da ne povećava test grešku onoliko koliko to čini pristup sa validacionim skupom podataka.
- Drugo, za razliku od prethodnog pristupa koji će dati različite rezultate ako se model primeni više puta (zbog slučajnosti u podelama skupa za trening/validaciju), izvođenje LOOCV više puta će uvek dati iste rezultate: *u podelama skupa za obuku/validaciju nema slučajnosti*.

c) k-Fold unakrsna validacija

Alternativa LOOCV je k-fold unakrsna validacija.

$$CV(k) = \frac{1}{k}\sum_{i=1}^{k} MSE_i$$



Slika 2. Unakrsna validacija je korišćena na skupu podataka *Auto* kako bi se procenila test greška modela koji predviđa *mpg* vrednost korišćenjem polinomskih funkcija konjskih snaga. **Levo**: LOOCV kriva greške. **Desno**: 10-fold CV bio je pokrenut devet odvojenih puta, svaki sa različitom nasumičnom podelom podataka na deset delova. Na slici je prikazano devet malo različitih CV krivih.

Kada vršimo unakrsnu validaciju**, naš cilj može biti da utvrdimo koliko dobro se može očekivati da će dati postupak statističkog učenja raditi na nezavisnim podacima**; u ovom slučaju je stvarna procena test MSE od interesa.

Ali u nekim drugim slučajevima nas zanima samo mesto minimalne tačke u procenjenoj testnoj MSE krivoj. To je zato što možda vršimo unakrsnu proveru nad brojnim statističkim metodama učenja, ili na jednoj metodi koristeći različite nivoe fleksibilnosti, kako bismo identifikovali metodu koja rezultira najmanjom test greškom. U tu svrhu je važno mesto minimalne tačke u procenjenoj test MSE krivoj, ali stvarna vrednost procenjene test MSE krive nije.

Bias-Variance Trade-Off za k-Fold unakrsnu validaciju

Manje očigledna, ali potencijalno važnija prednost k-fold CV je da često daje tačnije procene stope test grešaka od LOOCV pristupa. To ima veze sa kompromisom između bias-a i varijanse.

2. Bootstrap

Bootstrap je široko primenljiv i izuzetno moćan statistički alat koji se može koristiti za kvantifikovanje neodređenosti koja proističe iz korišćene statističke metode. Kao jednostavan primer, bootstrap se može koristiti za procenu standardnih grešaka koeficijenata koji se dobiju korišćenjem linearne regresije. Snaga bootstrapa leži u činjenici da se on lako može primeniti na širok spektar drugih statističkih metoda učenja, uključujući neke za koje je meru varijabilnosti inače teško dobiti i statistički softver ih ne daje automatski.

Šta proverava: Proverava **stabilnost i varijansu** (preciznost) procene. Fokus je na razumevanju raspodele neke statistike (npr. kolika je standardna greška koeficijenta ili preciznosti modela).

Cilj: Procena intervala poverenja (confidence intervals) i smanjenje varijanse modela kroz ansambl metode.

Kada se koristi: Kada imamo **veoma mali skup podataka** gde bi k-fold previše smanjio trening set, ili kada pravimo **ansambl modele** (npr. Random Forest) da bismo smanjili preprilagođavanje (overfitting).

Bootstrap je osnova za **Bagging** algoritme (poput Random Forest-a) i koristi se za procenu standardne greške ili intervala poverenja parametara modela.

PRIMER: Pretpostavimo da želimo da uložimo fiksnu sumu novca u dva biznisa koja donose prinose X, odnosno Y, gde su X i Y slučajne veličine. Uložićemo delić α svog novca u prvi biznis i ostvariti profit X i hoćemo da uložimo preostalih $1-\alpha$ u drugi biznis, pri čemu je dobit Y. Budući da postoji varijabilnost povezana sa prihodima iz ova dva biznisa, želimo da izaberemo α tako da umanjimo ukupan rizik ili varijansu naše investicije. Drugim rečima, želimo da minimalizujemo $Var(\alpha X + (1-\alpha)Y)$. Može se pokazati da se vrednost α može dobiti korišćenjem formule:

$$\alpha = \frac{\sigma_Y^2 - \sigma_{XY}}{\sigma_X^2 + \sigma_Y^2 - 2\sigma_{XY}},$$

gde su $\sigma_X^2 = Var(X)$, $\sigma_Y^2 = Var(Y)$, i $\sigma_{XY} = Cov(X,Y)$. Međutim, ove veličine su nepoznate u realnosti. Umesto pravih vrednosti, σ_X^2, σ_Y^2 i σ_{XY} ove vrednosti se mogu proceniti na osnovu nekih prethodnih vrednosti za X i Y. Tada se može odrediti procena za α, tj. $\hat{\alpha}$. Dakle, na osnovu prethodnih vrednosti X i Y možemo napraviti simulaciju i generisati nekoliko skupova podataka (1000). Na primer, vrednost $\hat{\alpha}$ koja je rezultat korišćenja svakog od 1000 simuliranih skupova podataka kreće se u rasponu od **0.532** do **0.657** (vrednosti nisu bitne, date su samo radi ilustracije). Sada se postavlja pitanje koliko je naša procena dobra.

Ako imamo 1000 simulacija skupa podataka, tada je:

$$\bar{\alpha} = \frac{1}{1000}\sum_{i=1}^{1000}\hat{\alpha}_i$$

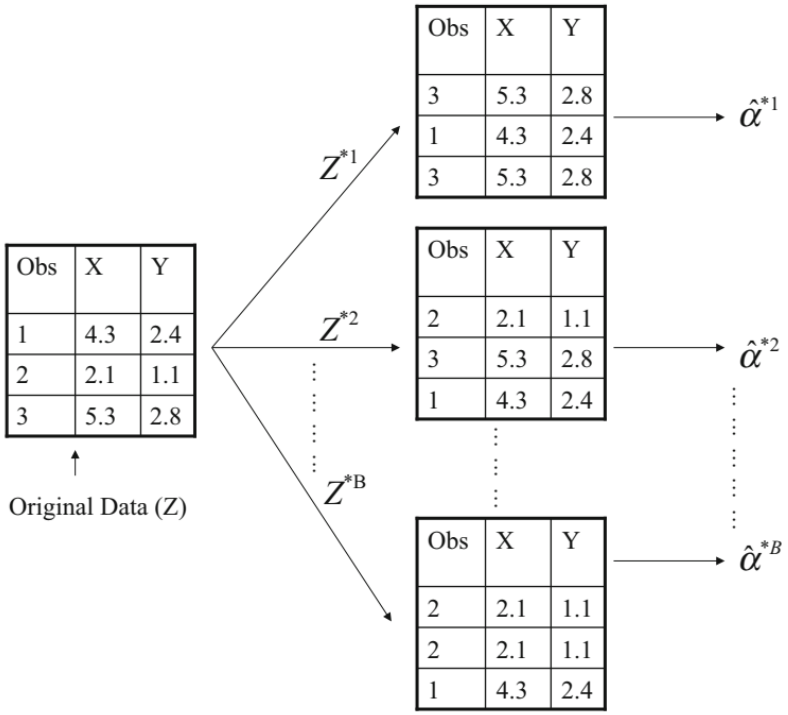
A standardna devijacija:

$$SE(\hat{\alpha}) = \sqrt{\frac{1}{1000-1}\sum_{i=1}^{1000}(\hat{\alpha}_i - \bar{\alpha})^2}$$

Ako je, na primer, $SE(\hat{\alpha}) = 0.08$ tada se može reći da za svaku populaciju očekujemo da se $\hat{\alpha}$ u proseku razlikuje za 0.08 u odnosu na pravu vrednost α.

Međutim, u praksi se gore opisani postupak za procenu $SE(\hat{\alpha})$ ne može primeniti, jer za stvarne podatke ne možemo generisati nove uzorke od prvobitne populacije. *Ali, bootstrap pristup nam omogućava da pomoću računara oponašamo postupak dobijanja novih skupova uzoraka, tako da možemo proceniti varijabilnost $\hat{\alpha}$ bez generisanja dodatnih uzoraka.* Umesto da više puta dobijamo nezavisne skupove podataka od populacije, umesto toga dobijamo različite skupove podataka višestrukim uzorkovanjem zapažanja **iz originalnog skupa podataka.**

Vrednosti za $\hat{\alpha}$ dobijene pomoću bootstrapa ili preko nezavisno simuliranih skupova su veoma slične.



Slika ##. Grafička ilustracija bootstrap pristupa na malom uzorku koji sadrži $n = 3$ observacija. Svaki set podataka bootstrapa sadrži n observacija, uzorkovanih zamenom iz originalnog skupa podataka. Svaki skup podataka bootstrapa koristi se za dobijanje procene α.

Da li je jedan pristup dovoljan?

U većini praktičnih situacija u mašinskom učenju, **k-fold cross-validation je dovoljan** za procenu kvaliteta modela. On je manje pristrasan (biased) od bootstrap-a kada je u pitanju procena greške predviđanja.

Međutim, jedan pristup često nije dovoljan ako:

1. *Radite sa ansamblima*: Tada vam treba bootstrap unutar samog algoritma (Bagging), a k-fold spolja da proverite taj ceo algoritam.
2. *Statistička sigurnost*: Ako vam nije dovoljno da znate da je tačnost 92%, već vam je neophodan interval poverenja (npr. 92% ± 2%), bootstrap je neophodan dodatak.
3. *Vremenske serije*: Ovde standardni k-fold i bootstrap ne rade dobro, jer se ne sme narušiti hronološki redosled. Tada se koriste specifične metode poput *Time Series Split*.

Izbor linearnog modela i regularizacija

Zašto su razvijene nove statističke metode i zašto se postojeće unapređuju? Dva su osnovna razloga za to:

1. Tačnost predikcije
2. Intrepretabilnost modela

Postoji mnogo alternativa, i klasičnih i modernih, koje mogu da iskoriste metodu najmanjih kvadrata za fitovanje podataka, a neki najčešće korišćeni pristupi su:

- Izbor prediktora (feature selection)
- Regularizacija
- Redukcija dimenzija (PCA)

Izbor prediktora (feature selection)

1. Izbor najboljeg podskupa prediktora

Da bismo izvršili najbolji odabir podskupa, prilagođavamo posebnu regresiju najmanjih kvadrata za svaku moguću kombinaciju p prediktora. Napraviti p modela koji imaju tačno jedan prediktor, $\binom{p}{2} = \frac{p(p-1)}{2}$ modela koji imaju 2 prediktora itd.

Algoritam:

1. Neka je $\mathcal{M}_0$ null model koji ne sadrži prediktore. Ovaj model predviđa prosek za svaku observaciju.
2. For $k = 1, \dots, p$:
   a) Napraviti svih $\binom{p}{k}$ modela koji imaju k prediktora
   b) Uzeti najbolji model iz ovog skupa i nazvati ga $\mathcal{M}_k$. Model koji ima najmanju **train** RSS vrednost, ili najveću R^2 vrednost.
3. Izabrati najbolji model od $\mathcal{M}_0, \mathcal{M}_1, \dots, \mathcal{M}_p$ koristeći grešku predikcije primenom unakrsne validacije, ili procenom test greške pomoću AIK, BIC ili adjusted R^2 vrednosti.

BELEŠKA: Odnos broja modela koji se prave u koraku 3 kada se koristi unakrsna validacija i kada se koristi procena test greške: ((p+1) * 10-fold) vs (p+1)

2. Selekcija po koracima

Selekcija sa korakom unapred

Algoritam:

1. Neka je $\mathcal{M}_0$ null model koji ne sadrži prediktore. Ovaj model predviđa prosek za svaku observaciju
2. For $k = 0, \dots, p-1$:
   a) Razmotriti svih $p-k$ modela koji proširuju prediktore u $\mathcal{M}_k$ sa jednim dodatnim prediktorom.
   b) Uzeti najbolji model od ovih $p-k$ modela i nazvati ga $\mathcal{M}_{k+1}$. Model koji ima najmanju **train** RSS vrednost, ili najveću R^2 vrednost.
3. Izabrati najbolji model od $\mathcal{M}_0, \mathcal{M}_1, \dots, \mathcal{M}_p$ koristeći grešku predikcije primenom unakrsne validacije, ili procenom test greške pomoću AIC, BIC ili adjusted R^2 vrednosti.

Ovo je značajna razlika: kada je $p = 20$, za izbor najboljeg podskupa potrebno je napraviti **1 048 576** modela, dok za selekciju sa korakom unapred potrebno je napraviti samo **211** modela.

Iako selekcija sa korakom unapred u praksi radi dobro, nije zagarantovano da će pronaći najbolji mogući model od svih 2^p modela koji sadrže podskupove p prediktora. Na primer, pretpostavimo da u datom skupu podataka sa $p = 3$ prediktora, najbolji mogući model sa jednom promenljivom sadrži X_1, a najbolji mogući model sa dve promenljive umesto X_1 sadrži X_2 i X_3.

Selekcija sa korakom unazad

1. Neka je $\mathcal{M}_\mathrm{p}$ *full model* sa svim prediktorima.
2. For $k = p, p-1, \dots, 1$:
   a) Razmotriti svih k modela koji sadrže sve sem jednog prediktora iz modela $\mathcal{M}_k$ – ukupno $k-1$ prediktor.
   b) Uzeti najbolji model od ovih k modela i nazvati ga $\mathcal{M}_{k-1}$. Model koji ima najmanju **train** RSS vrednost, ili najvecu R^2 vrednost.
3. Izabrati najbolji model od $\mathcal{M}_0, \mathcal{M}_1, \dots, \mathcal{M}_p$ koristeći grešku predikcije primenom unakrsne validacije, ili procenom test greške pomoću AIC, BIC ili adjusted R^2 vrednosti.

Selekcija sa korakom unazad zahteva da broj observacija n bude veći od broja prediktora p (tada se može napraviti *full* model). Za razliku od toga, selekcija sa korakom unapred se može koristiti i kada je $n < p$, pa je ovo jedini mogući pristup kada je p veliko.

Hibridni pristupi

Izbor najboljeg podskupa, selekcija sa korakom unapred i unazad uglavnom daju slične, ali ne i identične modele. Kao druga alternativa dostupne su hibridne selekcija sa korakom unapred i unazad, u kojima se modelu sekvencijalno dodaju promenljive, analogno selekciji sa korakom unapred. Međutim, nakon dodavanja svake nove promenljive, metoda takođe može ukloniti sve promenljive koje više ne pružaju poboljšanje modela.

Tips and tricks:

https://scikit-learn.org/stable/modules/feature_selection.html
https://en.wikipedia.org/wiki/Relief_(feature_selection)
https://www.geeksforgeeks.org/feature-selection-techniques-in-machine-learning/
(primetite da smo mi obradili ono što se u literaturi zove Wrapper and Embedded methods)

METRIKE

Da bismo izabrali najbolji model u odnosu na test grešku, mi moramo da procenimo ovu test grešku. Postoje dva uobičajena pristupa:

1. Možemo indirektno da procenimo test grešku prilagođavanjem vrednosti greška dobijene tokom treninga, a u ciju objašnjavanja bias-a zbog overfitting-a. Na primer, korišćenjem *Cp*, AIC, BIC, i Adjusted *R*2 vrednosti.

2. Možemo direktno proceniti test grešku, koristeći bilo validacioni skup ili pristup unakrsne validacije.

Cp, AIC, BIC, and Adjusted *R*2

Neka model ima *d* prediktora.

1) C_p

$$C_p = \frac{1}{n}(RSS + 2d\hat{\sigma}^2)$$

gde je $\hat{\sigma}^2$ procena varijanse greške. U osnovi, statistika Cp dodaje penal od $2d\hat{\sigma}^2$ u trening RSS kako bi se prilagodila činjenici da trening greška ima tendenciju da potceni (underestimate) test grešku.

2) *Akaike information criterion (AIC)*

$$AIC = \frac{1}{n\hat{\sigma}^2}(RSS + 2d\hat{\sigma}^2)$$

3) *Bayesian information criterion (BIC)*

$$BIC = \frac{1}{n}(RSS + \log(n)d\hat{\sigma}^2)$$

4) $Adjusted\ R^2$

$$adjustedR^2 = 1 - \frac{RSS/(n-d-1)}{TSS/(n-1)}$$

$$TSS = \sum(y_i - \bar{y})^2$$

Grafike ćemo prikazati na vežbama (8. termin)

Validacija i unakrsna validacija

Ova tema je kroz primere već obrađena na predmetu Uvod u veštačku inteligenciju.