Dubinska analiza podataka Ak. god. 2021./2022.

Teme koje mogu doći na završnom ispitu i rokovima

One teme koje su ovdje navedene obrađene su u okviru predavanja. Naziv teme odgovara naslovu slajd(ova) u prezentacijama. Na ispitu može doći pitanje koje je na neki način povezano s dotičnom temom (u pitanjima se neće tražiti navođenje matematičkih izraza). Za primjere uobičajenih pitanja na temelju navedenih tema može se pogledati u oglednom primjeru završnog ispita. Načelno, svako pitanje nosit će jednak broj bodova (2 boda), osim u iznimnim slučajevima kada pitanje s kraćim odgovorom može nositi 1 bod, a pitanje s opsežnijim odgovorom 3 boda.
Na završnom ispitu i rokovima u ak. godini 2021./2022. neće doći pitanja iz tema koje ovdje nisu pobrojane.

1. predavanje (Uvod)

- Što je dubinska analiza podataka?
- Klasifikacija vrsta strojnog učenja
- Šest glavnih problema od interesa za dubinsku analizu podataka
- Modeli procesa dubinske analize podataka
- CRISP-DM
- ASUM-DM
- CRISP ML(Q)

2. predavanje (Priprema podataka)

- Priprema podataka
- Proces pripreme podataka (ali **ne**: Proces pripreme podataka – CRISP-DM)
- Nedostajući podaci
- Nedostajući podaci – rješavanje problema
- Stršeći podaci (svi slajdovi)
- Stršeći podaci – rješavanje problema
- Šumoviti podaci
- Monotone značajke
- Konstantne značajke
- Koraci pripreme podataka – pojednostavljeno

3. predavanje (Transformacije podataka i inženjerstvo značajki)

- Značajke
- Značajke – razlozi korištenja
- Diskretizacija numeričkih vrijednosti
- Pretvorbe kategoričkih varijabli
- Normalizacija vrijednosti varijabli
- Uklanjanje nebitnih i redundantnih značajki
- Izgradnja značajki
- Analiza glavnih komponenti
- Višedimenzijsko skaliranje
- t-SNE

4. predavanje (Odabir značajki)

- Odabir značajki
- Ciljevi
- Podjela metoda odabira značajki
- Relevantnost i redundantnost značajki
- Pristupi odabiru značajki
- Filterski postupci
- Filterski postupci za pojedinačne značajke
- Informacijska dobit
- Simetrična nesigurnost
- Obitelj metoda Relief
- mRMR
- Postupci omotača
- Pohlepno pretraživanje
- Slijedna plutajuća selekcija
- Ugrađeni postupci
- Odabir značajki kod slučajne šume
- Logistička regresija s penalizacijom (samo prvi od tri slajda)
- Hibridni postupci

5. predavanje (Nebalansiranost podataka i pomak koncepta)

- Nebalansiranost podataka
- Stupanj nebalansiranosti klasa
- Ponovno uzorkovanje
- Naduzorkovanje
- Naduzorkovanje zasnovano na SMOTE-u
- Poduzorkovanje
- Poduzorkovanje – metoda NCL
- Poduzorkovanje – metoda TL
- Hibridni postupci uzorkovanja
- Učenje osjetljivo na cijenu
- Korištenje prikladnih mjera vrednovanja modela
- Pomak koncepta u podacima
- Online učenje
- Detektori pomaka zasnovani na monitoriranju
- Hoeffdingova stabla
- Odabir značajki u tokovima podataka

6. predavanje (Metode ansambala i njihovo tumačenje)

- Ansambli i njihovo korištenje
- Temeljna podjela pristupa ansambala
- Heterogeni jednostavni ansambl
- Stacking
- Bagging
- Boosting
- Slučajna šuma
- Slučajna šuma – značajke
- Slučajna šuma – početne postavke
- Slučajna šuma – algoritam
- Iznimno slučajna stabla

- Rotacijska šuma (samo prvi od tri slajda)
- AdaBoost i MultiBoost
- XGBoost (samo prvi od tri slajda)
- Permutacijska važnost
- SHAP (samo prva dva slajda)

7. predavanje (Strojno učenje s jasnim tumačenjem i indukcija pravila)

- Strojno učenje s jasnim tumačenjem
- Klasifikacija pristupa strojnog učenja s jasnim tumačenjem
- Indukcija pravila (induktivna pravila)
- Učenje pravila – konjunkcijsko pravilo
- Konjunkcijsko pravilo
- Učenje pravila po principu „razdvoji pa vladaj”
- Relaksacija potpunosti i konzistentnosti
- Heuristike pokrivanja
- Izbjegavanje prenaučenosti
- Višeklasna klasifikacija
- Preklapanje pravila i problem nepostojanja pravila
- OneRule – algoritam
- RIPPER
- RIPPER – algoritam (samo prvi slajd)
- Otkrivanje podgrupa

8. predavanje (Asocijativna pravila)

- Pronalazak čestih obrazaca i asocijativna pravila
- Itemset
- Potpora itemseta
- Asocijativna pravila i relevantnost
- Asocijativna pravila – cilj
- Algoritam Apriori
- Algoritam Apriori – ideja
- Algoritam Apriori – primjer
- Algoritam PCY (svi slajdovi)
- Algoritam FP-growth (samo prvi slajd)
- Algoritam EFIM
- Algoritam EFIM – definicija problema HUIM
- Algoritam EFIM – primjer

9. predavanje (Dubinska analiza vremenskih nizova)

- Dubinska analiza vremenskih nizova
- Vremenski niz podataka
- Glavni zadaci analize vremenskih nizova
- Terminologija (svi slajdovi)
- Predobrada vremenskog niza – uklanjanje šuma i skaliranje
- Predobrada vremenskog niza – ponovno uzorkovanje i interpolacija
- Predobrada vremenskog niza – podjela na prozore
- Reprezentacijske metode
- Neadaptivne reprezentacijske metode (prvi slajd)
- Adaptivne reprezentacijske metode (svi slajdovi)

- Reprezentacijske metode zasnovane na modelu
- Klasifikacijski algoritam: HIVE-COTE
- HIVE-COTE (prvi slajd)
- ARIMA (prvi slajd)
- PROPHET (prvi slajd)
- Modeliranje niza ekstrakcijom značajki (prvi slajd)

10. predavanje (Duboko učenje u dubinskoj analizi podataka)

- Duboko učenje
- Umjetne neuronske mreže
- Umjetne neuronske mreže – aktivacijske funkcije
- Algoritmi učenja i optimizacija hiperparametara (svi slajdovi)
- Funkcija gubitka
- Zašto duboke neuronske mreže uspijevaju?
- Umjetne neuronske mreže – konvolucijska (svi slajdovi)
- Umjetne neuronske mreže – rekurentne
- Umjetne neuronske mreže – rekurentne – LSTM
- Umjetne neuronske mreže – rekurentne – slaganje blokova
- Umjetne neuronske mreže – autoenkoderi (samo prvi slajd)
- Transformeri (samo prvi slajd)
- Problemi dubokih neuronskih mreža
- Obrada prirodnog jezika – BERT i RoBERTa (prva dva slajda)
- Analiza vremenskih nizova – ROCKET (samo prvi slajd)

Napomena: *S ovom skriptom sam ostvario 35 bodova na ZI bez učenja zadnje lekcije*

Dubinska analiza podataka – Teme za završni ispit i rokove

1. predavanje (Uvod)

1. Što je dubinska analiza podataka?
Proces pronalaženja skrivenih i korisnih informacija u podacima. Također se može reći da je dubinska analiza podataka = primijenjenom strojnom učenju.

2. Klasifikacija vrsta strojnog učenja.
Nadzirano (kategorička ili numerička ciljna varijabla), nenadzirano (nema ciljnu varijablu), polu-nadzirano (kategorička ciljna varijabla) i podržano učenje (kategorička ciljna varijabla ili je nema).

3. Šest glavnih problema od interesa za dubinsku analizu podataka.
1. Otkrivanje anomalija – identifikacija neobičnih, potencijalno zanimljivih podataka
2. Modeliranje ovisnosti – traženje odnosa između više varijabli
3. Grupiranje – zadatak otkrivanja grupa sličnih podataka
4. Klasifikacija – zadatak generaliziranja poznate strukture za primjenu na nove podatke
5. Regresija – pokušava pronaći funkciju koja modelira ciljne numeričke podatke s najmanjom pogreškom
6. Sažimanje – zadatak pružanja kompaktne reprezentacije skupa podataka, najčešće u vidu vizualizacije ili izvještaja

4. Modeli procesa dubinske analize podataka
1. KDD – znanstveni pristup otkrivanju podataka u bazama podataka. Oslanja se na pronalaženje uzoraka u podacima koji se smatraju znanjem kada pređu neki definirani prag zanimljivosti.
2. CRISP-DM
3. ASUM-DM
4. CRISP-ML(Q)

5. CRISP-DM
- Industrijski pristup standardizaciji projekata dubinske analize podataka
- Proces razložen u 6 faza: **razumijevanje poslovnih potreba**, **razumijevanje podataka**, **priprema podataka**, **modeliranje**, **vrednovanje** i **puštanje u pogon**
- Faze se ostvaruju pomoću jednog ili više generičkih zadataka
- **Generički zadatci**: generalni i stabilni pristupi koji pokrivaju veći broj izazova dubinske analize
- **Specijalizirani zadatci**: izvode određeni generički zadatak
- **Instanca procesa**: zapis **stvarno** provedenih akcija, odluka i rezultata
- Glavni nedostatak: ne uključuje aktivnosti upravljanja projektom

6. ASUM-DM

- Proširenje CRISP-DM-a, uključuje i poslovni aspekt upravljanja projektom
- Kod puštanja u pogon dodane su nove značajke: **suradnja**, **kontrola verzija koda**, **sigurnost** i **usklađenost s regulativama**
- Pet faza: **analiza, oblikovanje, konfiguracija i izgradnja, puštanje u pogon, djelovanje i optimizacija**
- Prve tri faze su združene zbog iterativne prirode DAP projekata
- Upravljanje projektima je opcionalno

7. CRISP-ML(Q)

- Model procesa namijenjen razvoju modela strojnog učenja, njegovom puštanju u pogon i održavanju
- Šest faza: isto kao i CRISP-DM s dodatnom fazom **nadzor i održavanje**
- Svaki korak ima kontrolu kvalitete
- Nadzor i održavanje: prate se performanse modela, model se treba kontinuirano osvježavati uslijed promjena

2. predavanje (Priprema podataka)

1. Priprema podataka
Temeljni dio podatkovnog inženjerstva (**50%-80%** vremena). Općeniti naziv za sve operacije nad podacima koje slijede nakon preuzimanja izvornih podataka s mjesta pohrane sve do početka analize statističkim postupcima i postupcima strojnog učenja.

2. Proces pripreme podataka
- Iterativan proces
- Tri ključna koraka: **otkrivanje podataka** (pronalazak izvora podataka, ustanovljavanje strukture), **karakterizacija podataka** (pregled podataka, statistika, vizualizacija) i **izgradnja skupa podataka za modeliranje** (organiziranje i filtriranje podataka, inženjerstvo značajki)

3. Nedostajući podaci
- Vrste nedostajućih podataka: **nedostajuće, ali poznate vrijednosti** (nisu unesene u skup podataka, ali postoje u stvarnom procesu) i **prazne i nepoznate vrijednosti** (ne može se pretpostaviti vrijednost u stvarnom svijetu i ona nije unesena). Često nije jasno o kojoj se vrsti radi.
- Detekcija problema: **detaljni pregled skupa podataka** ili **korištenje vizualizacije**

4. Nedostajući podaci – rješavanje problema
- Najvažnije je da osoba koja modelira podatke ima **kontrolu nad metodom** rješavanja problema nedostajućih podataka
- **Obrazac nedostajućih vrijednosti ponekad zadrži važnu informaciju!**
- Jednostavni postupci: **zanemarivanje svih primjeraka koji sadrže nedostajuću vrijednost, zamjena nedostajuće vrijednosti nekom drugom** (bitno da se informacijski sadržaj ne degradira)
- Zamjena vrijednosti (pristupi)**: očuvanje mjere sredine**, **očuvanje varijabilnosti**, **proglašenje nedostajuće vrijednosti novom kategorijom** i **zamjena s konstantnom vrijednosti**
- Očuvanje mjere sredine i varijabilnosti nepristrano je **samo za jednu** zamjenu po značajki
- Složeni postupci: **linearna regresija, k-najbližih susjeda, nelinearna regresija** i **neuronske mreže**. (Složeni postupci promatraju odnos između **više** značajki za razliku od jednostavnih)

5. Stršeći podaci
- Podaci koji odskaču daleko izvan uobičajenih vrijednosti (problem ako nisu rezultat stvarnosti)
- Postupci otkrivanja stršećih podataka: **vizualizacija podatak i opažanje**, **statistički postupci** (z-skor, linearna regresija), **algoritmi nadziranog i polunadziranog učenja** i **algoritmi nenadziranog strojnog učenja** (temeljeni na udaljenosti – k-NN, ORCA, temeljeni na gustoći – LOF, temeljeni na kutovima – ABOD, algoritmi specifični za velike skupove podataka – Isolation Forest)

Algoritam LOF (lokalnih faktora stršećih vrijednosti):
- Uspoređuju se lokalne gustoće točke s gustoćama susjeda
- LOF=1 – slična točka, vjerojatno nije stršeća, LOF>1 – manja gustoća, vjerojatno stršeća točka, LOF<1 – točka veće gustoće, nije stršeća
- Prednosti: usporediv ili bolji od ostalih algoritama
- Nedostatci: nije pogodan za velike skupove podataka

Algoritam IsolationForest
- Pretpostavka: stršeće vrijednosti je lakše izdvojiti nego modelirati normalne podatke
- Algoritam rekurzivno generira binarne particije skupa podatka nasumičnim odabirom značajki i raspona vrijednosti te značajke
- Glavna ideja: nasumično particiranje daje **znatno kraće puteve za stršeće** nego za normalne vrijednosti
- s(x, n) blizu 1 označava da se vjerojatno radi o anomaliji, dok blizu 0.5 je sigurno normalan
- Prednosti: mala vremenska složenost i memorija, dobar s visokodimenzionalnim podacima
- Nedostatci: teško detektira grupne anomalije i mjera stršećih vrijednosti je dosta heuristička

6. Stršeći podaci – rješavanje problema
- Utvrditi je li stršeća vrijednost prirodna ili ne
- Ako nije prirodna: tretirati ga kao nedostajući podatak i primijeniti prikladne metode
- Ako je prirodan, ali se zna da će smetati: koristiti **normalizaciju**

7. Šumoviti podaci
- Šum u podacima je u nekoj mjeri prisutan u svim podacima koji su rezultat mjerenja putem senzora
- Podatak: pravi signal + šum
- Izvor šuma: prirodni procesi ili nesavršenost mjernih senzora
- Postoje postupci za filtriranje šuma, ali su **iznimno ovisni o konkretnom problemu**

8. Monotone značajke
- Značajke čija vrijednost raste (ili se smanjuje) bez ograničenja (redni brojevi, datumi i sl.)
- Rješenja problema: **zanemariti takvu značajku**, **transformirati u oblik pogodan za modeliranje** (npr. gledati na datume kao vremensku seriju ili ih pretvoriti u godišnje doba i sl.)

9. Konstante značajke

- Samo ime sve govori
- Nisu informacijski relevantne (varijanca = 0) i **uvijek ih treba ukloniti**

10. Koraci pripreme podataka – pojednostavljeno

1. Sudjelovati u procesu prikupljanja i isporuke skupa podataka (po mogućnosti)
2. Pomno pregledati skup podataka kako bi se ustanovili svi problemi u skupu
3. Raspisati sve uočene probleme i predložiti rješenja za njihovo uklanjanje
4. Prodiskutirati uočeno i predložena rješenja s relevantnim dionicima (kolege, inženjeri, nadređeni…)
5. Provesti usvojena rješenja.

3. predavanje (Transformacije podataka i inženjerstvo značajki)

1. Značajke

- Preoblikovane varijable koje se koriste na ulazu metode strojnog učenja nazivamo **značajkama**, a cijeli primjerak **vektor značajki**
- Nekad su izvorne varijable značajke, ali češće, značajke su rezultat posljednje **matematičke transformacije**
- Mogu biti vrlo specifične i **domenski definirane**, te često imaju razumljivu interpretaciju

2. Značajke – razlozi korištenja

Zašto nam trebaju značajke?

- **Sažetost**: Sirovih podataka ima previše za učinkovito korištenje algoritama strojnog učenja
- **Informativnost:** Sirovi podaci nisu toliko informativni kao značajke, jer se prava informacija krije u odnosima između njih
- **Interpretabilnost**: Ljudi bolje razumiju značajke od sirovih podataka kada se radi o interpretaciji modela strojnog učenja

3. Diskretizacija numeričkih vrijednosti

- Pretvorba numeričke vrijednosti neke varijable u kategoričke vrijednosti
- Pretpostavka: broj kategorija << broj numeričkih vrijednosti
- Često potreban korak jer neki algoritmi rade samo s kategoričkim varijablama
- Diskretizacijom se uvijek gubi određena informacija
- Postupci diskretizacije: Nenadzirani pristup (podjela u N jednakih intervala, podjela u intervale s jednakim brojem pojedinaca, diskretizacija k-means-om) i nadzirani pristup (diskretizacija minimizacijom entropije)

4. Pretvorbe kategoričkih varijabli

- Izravno preslikavanje kategoričke značajke u numeričku (label encoding)
- Svaka kategorija neke kategoričke značajke postaje nova binarna značajka (one-hot encoding)

5. Normalizacija vrijednosti varijabli

- Potrebno kada su različite značajke u skupu podataka mjerene na različitim skalama
- Postupci: **decimalno skaliranje**, **min-max**, **normalizacija z-skorom**
- Najčešća normalizacija je na raspon od 0 do 1

6. Uklanjanje nebitnih i redundantnih značajki

- Vrste nebitnih značajki: **monotone, konstantne i duplikatne značajke**
- Statistički redundantne značajke određuju se: **Korelacijskom analizom** (izbacivanje jedne od dviju visoko koreliranih značajki – obično korelacija > 0.9) i **Markovljevim pokrivačem**

7. Izgradnja značajki

- Fokus na poboljšanju performansi, nema pokušane formule
- Iterativna primjena različitih operatora za izgradnju novih značajki: **konjunkcija, disjunkcija, negacija** za binarne varijable, **M od N – varijabla poprima vrijednost 1 ako je barem M od N uvjeta (ostale varijable) istinito, Ekvivalencija, nejednakost, zbrajanje, oduzimanje, množenje, dijeljenje** za numeričke varijable i **složenije transformacije podataka**
- Podjela pristupa za izgradnju značajki:
    - **Zasnovani na podacima** – na temelju opaženih podataka
    - **Zasnovani na hipotezi** – na temelju prethodno generirane hipoteze
    - **Zasnovani na znanju** – na temelju domenskog znanja
    - **Hibridni pristupi** – kombiniraju tri gornja pristupa

8. Analiza glavnih komponenti

- Metoda redukcije dimenzionalnosti, tradicionalno najčešća metoda transformacije varijabli
- Tehnika nenadziranog učenja, korisna za otkrivanje grupa podataka i otklanjanje šuma
- Cilj je prikazati što više varijance s nekoliko glavnih komponenti (npr. 95% varijance s 2 dimenzije, ako su originalni podaci npr. 4 dimenzionalni)

9. Višedimenzijsko skaliranje

Skup metoda koje predstavljaju **mjere sličnosti** između parova objekata u visokodimenzionalnom prostoru kao metričku **udaljenost** između točaka u niskodimenzionalnom prostoru.
- Cilj je minimizirati mjeru STRESS koja predstavlja razliku udaljenosti u nižedimenzionalnom prostoru i stvarne udaljenosti (različitosti) u originalnom prostoru.

10. t-SNE

Traži niskodimenzionalnu strukturu tako da **svojstva grupiranja** u višoj dimenziji ostanu sačuvana.
- Sličnost u visokodimenzionalnom prostoru predstavljeno je Gaussovim **zajedničkim vjerojatnostima** euklidskih udaljenosti dvaju objekata
- Sličnost u ugrađenom prostoru (najčešće 2D ili 3D) mjeri se zajedničkim vjerojatnostima Studentovim t-razdiobama euklidskih udaljenosti

Metoda je vrlo računski zahtjevna, ali je prednost što **čuva lokalnu bliskost točaka** pri preslikavanju u nižu dimenziju te može modelirati nelinearne odnosne između varijabli (za razliku od PCA).

4. predavanje (Odabir značajki)

1. Odabir značajki

- Postupak smanjenja dimenzije (broja varijabli); zadržava se interpretacija značajki
- Uobičajeno želimo postići: **zadržati rezultat** modeliranja početnog skupa značajki ili ga **poboljšati**, **smanjiti vrijeme** potrebno za izgradnju modela i **pojednostaviti model** radi boljeg razumijevanja

2. Ciljevi

1. Najmanji podskup značajki koji daje bolje rezultate (manju pogrešku) nego početni skup
2. Najmanji podskup značajki koji daje približno jednake rezultate kao početni skup značajki
3. Bilo koji podskup značajki koji daje najbolje rezultate
4. Rangiranje značajki prema važnosti za zadani cilj
5. Odabir točno k od početnih M značajki takvih da daju najbolji rezultat

- Iscrpna pretraga za optimalnim podskupom je **NP težak problem**

3. Podjela metoda odabira značajki

1. Tablične metode – znamo da su značajke neovisne jedna od druge
    - Filterski postupci
    - Postupci omotača
    - Ugrađeni postupci
    - Hibridni postupci

2. Strukturne metode – znamo da su značajke na neki način povezane jedna s drugom
    - Struktura grafa
    - Struktura stabla
    - Struktura grupe

4. Relevantnost i redundantnost značajki

- Četiri tipa značajki u skupu podataka: **Irelevantne**, **Slabo relevantne i redundantne, Slabo relevantne, ali neredundantne** i **Snažno relevantne**. (3 i 4 daju optimalni podskup)

5. Pristupi odabiru značajki

- Dva pristupa odabira značajki:
    - Određivanje relevantnosti **pojedinačnih značajki** (jednostavniji pristup)
        - Rangiraju se prema nekoj mjeri
        - Uklanjaju se ako ne zadovoljavaju neki postavljeni prag
        - Ne uzima se u obzir redundantnost
    - Određivanje relevantnosti i redundantnosti na **podskupu značajki**
        - Razmatraju se uvijek podskupovi značajki početnog skupa
        - Računa se njihov značaj, pronalazak redundantnih značajki je **implicitan**

6. Filterski postupci

- Glavna značajka: filterski postupci **ne koriste algoritam strojnog učenja** da bi napravili odabir značajki
- Filterski postupci definiraju **mjeru** koliko su određena značajka ili skup značajki bitni za opis ciljne značajke
- Razlikuju se filterski postupci za pojedinačne i za skupove značajki, kao i postupci za nadzirano i nenadzirano učenje

7. Filterski postupci za pojedinačne značajke

- Informacijske mjere:
    - **Korelacijski koeficijent** – za regresijske probleme
    - **Informacijska dobit** (očekivana zajednička informacija)
    - **Simetrična nesigurnost**
- **hi-kvadrat**, **Fisherov skor**, Obitelj metoda **Relief** itd.

8. Informacijska dobit

- Mjera informacijske dobiti je simetrična
- U praksi, razmatra se odnos između **prediktivne značajke X i ciljne značajke Y**
- Informacijske dobiti se rangiraju za sve prediktivne značajke X

9. Simetrična nesigurnost

Informacijska dobit **favorizira značajke s većim brojem vrijednosti**, a kako bi se doskočilo tome, predložena je mjera **simetrične nesigurnosti** koja ograničava vrijednosti informacijskog dobitka na interval [0, 1]. (0 – neovisnost, 1 – potpuna prediktivnost)

- SU za odnos prediktivne i ciljne značajke se naziva **C-korelacija**, a između dvije prediktivne **F-korelacija**

10. Obitelj metoda Relief

- Pojedinačno razmatranje i rangiranje značajki; detektiraju ovisnost među prediktivnim značajkama
- Ne razmatraju podskupove značajki nego su temeljeni na najbližim susjedima za određivanje mjere korisnosti pojedine značajke
- Karakteristike: nešto **složenije (i sporije)** od ostalih filterskih metoda, ali načelno **točnije** i **ne uklanjaju redundantne značajke**

11. mRMR

- Ideja je pronaći one značajke koje imaju **visoku korelaciju prema ciljnoj značajki i nisku korelaciju prema drugim značajkama**
- Gledaju se D (relevantnost) i R (redundantnost) te se maksimizira njihova razlika (D-R)

12. Postupci omotača

- Glavna značajka: **koriste algoritam strojnog učenja za evaluaciju** određenog podskupa značajki (često različit algoritam od onog koji se koristi za izgradnju modela)
- U pravilu: **sporiji, ali točniji postupci od filtera**
- I dalje vrijedi NP optimizacijski problem, ali je sad još istaknutiji jer postupci omotača nemaju neku jednostavnu mjeru važnosti podskupa, već moraju graditi model za svaki podskup

13. Pohlepno pretraživanje

- **U svakom koraku pretrage dodaje (ili uklanja) jednu značajku koja najviše poveća točnost algoritma strojnog učenja**. Zaustavlja se čim dođe do degradacije točnosti
- U pretrazi unaprijed ne smanjuje odabrani skup značajki, a u pretrazi unazad ga ne povećava

14. Slijedna plutajuća selekcija

1. korak: dodaje jednu značajku u skup koja najviše poveća točnost algoritma strojnog učenja
2. korak: ukloni jednu značajku iz skupa ako bilo koja od njih pri uklanjanju poveća točnost
- Korak 2 se ponavlja dok smanjenje broja značajki povećava točnost te se onda prelazi na 1.
- Izvođenje se zaustavlja čim korak 1 ne dovodi do povećanja točnosti

15. Ugrađeni postupci

- Izbor značajki koji se **temelji na nekom algoritmu strojnog učenja(Slučajna šuma, Lasso, SVM)**
- Unutarnja struktura modela oslikava važnost značajki

16. Odabir značajki kod slučajne šume

- U fazi učenja, u svakom čvoru stabla odabire se slučajno jedna značajka po kojoj se grana skup podatka. To grananje može prouzročiti veću ili manju **nečistoću**. Grananje je čisto kada u svakom listu postoje primjerci samo jedne klase.
- Ideja odabira značajki je **uprosječiti u cijeloj šumi koliko svaka značajka smanjuje nečistoću**

17. Logistička regresija s penalizacijom

- LogReg – regresijski model s nelinearnom funkcijom odluke
- Regularizacija – koristi **objektivnu funkciju cijene** za **minimizaciju pogreške fitanja modela**
- **Lasso (L1-reg)** – koeficijenti značajki koje **manje doprinose modelu se forsiraju na vrijednost 0**, a samo bitne značajke se zadržavaju (parametar λ definira strogost koja forsira značajke u 0)
- Funkcija cijene kod L1-reg koja se želi minimizirati se sastoji od: **funkcije cijene uobičajene logističke regresije i λ-parametrom kontroliranog utjecaja koeficijenata pojedinačnih značajki**

18. Hibridni postupci

- Kombiniraju najbolja svojstva filtera i postupaka omotača
- Primjena **dvaju ili više** različitih postupaka filtera, omotača i ugrađenih postupaka (najčešće prvo filter pa postupak omotača)
- U praksi se pokazuju **točnijima od filterskih postupaka i bržim od postupaka omotača**

5. predavanje (Nebalansiranost podataka i pomak koncepta)

1. Nebalansiranost podataka

- Problem skupa podataka u kojem postoji **neravnoteža (nebalansiranost) u broju primjeraka pojedinih klasa ciljne značajke**. Neravnoteža otežava izgradnju modela koji će jednako dobro klasificirati i **većinske i manjinske klase.**

2. Stupanj nebalansiranosti podataka

Stupnjevi nebalansiranosti (**omjer nebalansiranosti**):

- Do omjera 4 (4:1) – blaga neuravnoteženost (klasifikatori većinom mogu dobro učiti)
- Do omjera 9 (9:1) – srednja neuravnoteženost (klasifikatori uglavnom imaju poteškoća)
- Do omjera 99 (99:1) – velika neuravnoteženost (klasifikatori najčešće ne uče dobro)
- Do omjera 100 (100:1) – rijetki događaji (klasifikatori gotovo sigurno ne uče dobro)

Nebalansiranost nije problem ako su klase potpuno odvojive, ali to je rijedak slučaj.

3. Ponovno uzorkovanje

- Osnovna ideja: **izmijeniti razdiobu primjeraka ciljne značajke po klasama kako bi bila približno jednaka za sve klase**
- Tri najčešća načina provedbe: **naduzorkovanje, poduzorkovanje, hibridni način uzorkovanja**

4. Naduzorkovanje

- Generiranje **novih primjeraka za učenje za manjinsku klasu** kako bi se postigla ravnoteža s većinskom klasom (češće od poduzorkovanja). Problem je ako je neuravnoteženost velika.
- Novi primjerci mogu biti generirani: **slučajnim naduzorkovanjem s ponavljanjem** postojećih primjeraka ili kao izmijenjeni, **sintetski primjerci** koji uzimaju u obzir postojeće primjerke (zasnovani na varijantama algoritma **SMOTE**)

5. Naduzorkovanje zasnovano na SMOTE-u

SMOTE – generiranje sintetskih primjeraka na temelju postojećih primjeraka manjinske klase

- **Primjerak se generira na linijskim segmentima koji povezuju primjerak i njegovih k susjeda**
    - Npr. za k = 2: Za svakog susjeda izračuna se **razlika (udaljenost)** između susjeda i primjerka, pomnoži se **slučajnim brojem između 0 i 1** i **doda primjerku**, čime se dobiva novi sintetski primjerak na liniji koja povezuje primjerak i najbližeg susjeda

6. Poduzorkovanje

- **Uklanjanje primjeraka većinske klase kako bi se postigla ravnoteža s manjinskom klasom**
- Problematično jer se **gubi dio informacije** sadržan u većinskim primjercima većinskih klasa
- Nekoliko češćih pristupa: **RUS, NCL, TL, CBU**

7. Poduzorkovanje – metoda NCL

- **Pravilo čišćenja susjedstva**: Koristi pravilo uređenih najbližih susjeda za uklanjanje primjerka
- **Pravilo ENN uklanja bilo koji primjerak čija klasa se razlikuje od klasa tri do pet njegovih najbližih susjeda**

8. Poduzorkovanje – metoda TL

- **Metoda Tomekovih poveznica**: modifikacija metode zgusnutog najbližeg susjeda (CNN)
- Dva primjerka **različitih klasa** formiraju Tomekove poveznice ako je euklidska udaljenost između njih manja nego udaljenost između svakog od njih i bilo kojeg drugog primjerka
- Iz podataka se **uklanjaju primjerci većinske klase koji formiraju Tomekove poveznice s manjinskom klasom**

9. Hibridni postupci uzorkovanja

- Kombiniranje poduzorkovanja i naduzorkovanja. Najčešći pristup: **slučajna ravnoteža**
- Uzme se slučajni broj S unutar raspona [2, N-2]. Većinska klasa se slučajno poduzorkuje dok ne postigne broj primjeraka = S. Manjinska klasa se SMOTE-a dok ne ostvari broj primjeraka = N-S.

10. Učenje osjetljivo na cijenu

- Osnovna pretpostavka: **neispravna klasifikacija primjerka manjinske klase u odnosu na većinsku** smatra se da je problematičnija (ima veću cijenu). Koristi se **matrica cijene**.
- Matrica cijene se može odrediti automatski: npr. cijena 1 za većinsku klasu, a omjer nebalansiranosti za manjinsku klasu.
- **Primjeri za učenje** se izmjenjuju tako da su manjinski primjeri otežani cijenom

11. Korištenje prikladnih mjera vrednovanja modela

- Ako se radi o nebalansiranom skupu podataka, ukupna klasifikacijska točnost nije dobra mjera
- Najčešće je poželjno primijeniti mjere uspješnosti:
    - **preciznost** i **odziva** – ako želimo procjenu za pojedinačne klase
    - **F1-mjeru** – ako želimo kvalitetnu ukupnu procjenu uspješnosti modela
    - **G-mean mjeru** – geometrijska srednja vrijednost – isto dobra za ukupnu procjenu

12. Pomak koncepta u podacima

Pomak koncepta u podacima označava pomak statističkog **svojstva ciljne značajke kroz vrijeme**

- Može biti nepredvidiv, a ogleda se u **smanjenju točnosti modela tijekom vremena**
- Ako se promijeni distribucija, ali ne i predikcijska funkcija – to je virtualni pomak i nije problem
- Pristupi detekciji pomaka u podacima: **Monitoring točnosti modela** (zahtijeva labelirane ulazne podatke za izgradnju modela), **Slijepa adaptacija modela** (model se tijekom vremena adaptira s novim podacima bez verifikacije promjena i **Nenadzirano ili polunadzirano učenje za detekciju**.

13. Online učenje

- Zajednička značajka metoda koje se predlažu za otkrivanje i karakterizaciju pomaka koncepta je **online učenje**. Algoritmi *online učenja* obrađuju **svaki primjerak za učenje** samo jednom, najčešće bez njegovog pohranjivanja. Primjerci najčešće dolaze u obliku **toka podataka,** u kojem slučaju je **fiksan broj značajki**.
- Metode online učenja mogu detektirati pomak **aktivno**(monitoriranje) / **pasivno**(slijepa adap.)
- Aktivne metode nakon detekcije promjene ponovno uče model, a pasivne ga uče cijelo vrijeme

14. Detektori pomaka zasnovani na monitoriranju

- Detektori pomaka zasnovani na monitoriranju koriste **statističke testove da prate je li došlo do promjene u razdiobi ciljne značajke** na temelju performanse pogreške modela.
- Smanjenje točnosti ima dvije razine: **upozorenje** i **pomak koncepta** (oba imaju **prag**)
- Po pojavi upozorenja počnu se pamtiti pristigli primjerci, a po pojavi pomaka koncepta **gradi se novi model na temelju starih podataka i prikupljenih zapamćenih primjeraka**
- Primjeri: **Metoda** (ranog) **otkrivanja pomaka, Metoda adaptivnog prozora**

15. Hoeffdingova stabla

Stabla odluke koja se grade na **inkrementalan način** tako da uzorci koji najprije dođu sudjeluju u izgradnji prvo korijena stabla, a kasniji uzorci se proslijede u grane prema podjeli u korijenu te se model kontinuirano nadograđuje.

- Ideja: **izgradnja novih čvorova nije automatska**, jer se grananje na nekoj značajki događa samo ako je razlika **ΔG** informacijske mjere u određenom čvoru između najinformativnije i druge najinformativnije značajke za grananje **veća od tzv. Hoeffdingove granice** ε.
- **Čvor treba postepeno akumulirati primjerke** koji pristižu iz toka podataka sve dok Hoeffdingova granica za te primjerke ne postane veća od razlike informacijske mjere **najinformativnijih značajki** u skupu **ΔG**

16. Odabir značajki u tokovima podataka

Problem gdje se broj značajki u skupu podataka mijenja. Nije poželjno čekati dok se prikupi veliki broj značajki kako bi se napravio dobar odabir značajki. Značajke treba razmotriti čim se pojave i odlučiti želimo li ih ostaviti u skupu podataka.

- Koraci odabira značajki u tokovima podataka:

1. Naučiti novu značajku iz toka podataka
2. Odlučiti želimo li uključiti novu značajku u skup podataka
3. Izmijeniti skup podataka s nadodanom novom značajkom (s povratkom na 1. korak)

- Primjer algoritma: Grafting, **Fast-OSFS**

6. predavanje (Metode ansambla i njihovo tumačenje)

1. Ansambli i njihovo korištenje

Ansambl u strojnom učenju je skup od dvaju ili više modela strojnog učenja koji ima cilj poboljšati uspješnost rezultata u odnosu na pojedinačne modele. Tipično se razmatraju u kontekstu **nadziranog učenja**.

- Pojedinačni modeli izgrađeni su s **istim** ili **različitim** temeljnim algoritmom strojnog učenja
- Ne garantiraju uspjeh, ali temeljna pretpostavka uspješnih ansambala je da **pojedinačni modeli trebaju biti raznoliki**

2. Temeljna podjela pristupa ansambla

- **Sustavi ansambala više stručnjaka:**
    - Heterogeni jednostavni ansambli
    - **Stacking**
    - **Bagging**
- **Sustavi ansambala u više koraka:**
    - **Boosting**
    - Kaskadirajući klasifikatori

3. Heterogeni jednostavni ansambl

Sastoji se od nekoliko (najčešće 3) različita pojedinačna algoritma strojnog učenja

- Pretpostavka dobrog korištenja u praksi: **algoritmi moraju biti značajno različiti**
- Odluke u fazi testiranja donose se najčešće **većinskim glasanjem**

4. Stacking

Smatra se nadogradnjom heterogenog jednostavnog modela

- Ansambl metoda gradi se u dvije razine:
    - Niža razina (razina 0) sastoji se od **jednog ili više modela** različitih algoritama koji uče na jednom većem dijelu ulaznog skupa
    - Viša razina (razina 1) sastoji se od **jednog** modela koji uči na temelju izlaza modela razine 0 za preostali dio ulaznog skupa
- **Zadatak modela razine 1 je odrediti kako najbolje iskombinirati doprinose modela razine 0**

5. Bagging

Ansambl koji se sastoji od većeg broja istog ili različitih modela. Ulazni podaci za svaki pojedinačni model dobivaju se uzrokovanjem tipa *bootstrap***.** Glavna značajka: Bagging **smanjuje varijancu** pojedinačnih modela

- Odluke u fazi testiranja donose se agregacijom rezultata pojedinačnih modela i to **većinskim glasanjem**

6. Boosting

Postupak *boosting* gradi ansambl modela **iterativno**, nastojeći poboljšati decizijsku granicu. U svakoj novoj iteraciji uči se novi model na skupu za učenje te se mijenja skup za učenje. Na početku svi primjerci u skupu za učenje imaju jednaku težinu, a u idućim iteracijama oni **primjerci koji nisu točno klasificirani dobivaju veću težinu**, dok se težina točnih smanjuje.

- *Boosting* poglavito **smanjuje pristranost** modela

7. Slučajna šuma

Poznati ansambl slučajnih stabala odluke, pripada u skupinu vrhunskih algoritama. U klasifikacijskim problemima točnost rezultata je usporediva s najboljim algoritmima, a **brzina mu je često bitno veća**. Može se koristiti i za regresiju, ali najčešće se koristi za klasifikaciju.

- Koristi **kombinaciju izvora slučajnosti**, **veliki broj stabala** i **većinsko glasanje**

8. Slučajna šuma – značajke

- **Minimizira pogrešku nastalu zbog varijance i pristranosti**
    - Pristranost minimizira tako što se stablo gradi do kraja i ne podrezuje se
    - Pogreška zbog varijance se minimizira uz pomoć **bootstrapa, slučajnog odabira atributa u stablu i velikog broja stabala**
- **Postupak je neosjetljiv na prenaučenost**
- Osigurava visoku točnost rezultata na većini problema zbog **velike snage pojedinog stabla** i **niske korelacije (značajne različitosti) među stablima**

9. Slučajna šuma – početne postavke

Ansambl se sastoji od **K slučajnih stabala odluke**, svaki s **vlastitim skupom** za učenje.

- Određuje se broj $m = \sqrt{M}$, gdje je M ukupan broj značajki. m je broj značajki koje će se **razmatrati prilikom podjele svakog čvora** u stablu.
- Postavlja se najmanji dozvoljeni broj primjeraka u čvoru koji nije list na $n_{min} = 2$
- Postavlja se broj stabala K = 100

10. Slučajna šuma – algoritam

Za svako stablo:

1. Provodi se uzrokovanje tipa **bootstrap**
2. Za svaki čvor u stablu:

    Ako čvor nije list:

    - odaberi m značajki kandidata iz skupa svih značajki M
    - generiraj m podjela, takvu da je podjela najinformativnija za dotični atribut
    - uzmi najinformativniju od m podjela, podijeli primjerke u čvoru na lijevi i desni čvor prema toj podjeli i pokreni 2. korak prvo za lijevi, zatim za desni čvor

    Inače: vrati list označen s najčešćom klasom

11. Iznimno slučajna stabla

Ansambli iznimno slučajnih stabala nastoje popraviti točnost slučajnih šuma tako što uvode neke **dodatne izvore slučajnosti, dok neke zanemaruju**. Dijele se na **iznimno slučajna stabla i potpuno slučajna stabla**.

- Iznimno slučajna stabla slučajno odabiru podjelu neke značajke na čvoru, ali izabiru najbolju takvu podjelu (prema mjeri Gini) između slučajno odabranih $\sqrt{M}$ značajki u čvoru.
- Potpuno slučajna stabla su iznimno slučajna stabla koja u svakom čvoru **slučajno biraju značajku koju slučajno dijele.**
- Za razliku od slučajne šume **ne rade bootstrap** nego se na svakom stablu koristi čitav skup. Za klasifikaciju su 2-10 puta **brža** od slučajnih šuma, ali zauzimaju 2-4 puta **više prostora**.

12. Rotacijska šuma

Ansambl klasifikatora stabala odluke. Postiže izvrsne rezultate na skupovima podataka s numeričkim vrijednostima. Značajno je sporiji od slučajne šume na velikim skupovima podataka.

- Zasniva se na **bootstrapu, PCA** i **ansamblu stabala odluke C4.5**

13. AdaBoost i MultiBoost

Algoritmi ansambala temeljeni na **iterativnom pristupu izgradnji modela korištenjem postupaka *boosting* nad inicijalno slabim modelom. Prednosti:** učinkovita i brza izgradnja modela, visoka točnost. **Nedostatci**: nije uvijek točniji od *bagging* metoda, lakše se prenauči. **MultiBoost** je varijanta AdaBoosta u kojoj se koristi stablo odluke **C4.5** u svakoj iteraciji umjesto panja odluke. Također, MultiBoost koristi **utežani bagging**. (Za više info 6.preza, slajdovi 29-31)

14. XGBoost

Napredan algoritam strojnog učenja temeljen na postupku ***boostinga*** i **optimizaciji gradijentnim spustom.** Kod „gradijentnog boostinga“, *boosting* se razmatra kao problem numeričke optimizacije gdje je cilj **minimizirati gubitak modela** u svakom koraku tako da se u ukupni model dodaju slabi modeli iterativno, koristeći proceduru **sličnu** minimizaciji grad. s.

- Kao slabi modeli koriste se **stabla odluke CART** (obično podrezana na neku kraću dubinu, da budu malo točnija od panja, a opet brza za izgradnju)

15. Permutacijska važnost

Definira se kao **smanjenje rezultantne mjere modela kada se vrijednosti značajke po primjercima nasumično promiješaju.**

- Miješanje vrijednosti primjeraka **prekida odnos između značajke i ciljne značajke** pa pad rezultantne mjere modela pokazuje koliko model ovisi o svakoj značajki.

16. SHAP

Objašnjivi postupak **lokalnog tipa**. Postupak koji se koristi za objašnjavanje bilo kojeg modela strojnog učenja. Model (crna kutija) se objašnjava u vizualizacijom doprinosa pojedinih značajki.

7. predavanje (Strojno učenje s jasnim tumačenjem i indukcija pravila)

1. Strojno učenje s jasnim tumačenjem

Uključuje model strojnog učenja čije tumačenje je: **jasno razumljivo čovjeku** (white-box modeli) i modele strojnog učenja koji nemaju jasno tumačenje (black-box) **ali koji se mogu protumačiti** korištenjem **post-hoc** tehnika. Tumačenje black-box modela je tema **objašnjive UI.**

- Prednost **black-box modela** je najčešće **veća točnost**, a prednost **white-box modela** je **jednostavnost tumačenja.**

2. Klasifikacija pristupa strojnog učenja s jasnim tumačenjem

- **Prema tipu metode**
    - **Intrinzične metode** – metoda interpretacije intrinzična je samom modelu (white-box)
    - **Metode za post hoc interpretaciju** – metoda interpretacije primjenjuje se na već naknadno izgrađeni model (black-box)
- **Prema specifičnosti u odnosu na model**
    - **Specifične za pojedini model ili klasu modela** – npr. za regresiju, neuronske mreže
    - **Agnostičke prema modelu** – mogu se primijeniti na sve modele
- **Prema lokalnosti**
    - **Lokalne** – objašnjavaju odluku za **pojedinačne primjerke** ili skupove primjeraka
    - **Globalne** – objašnjavaju odluke za čitavi skup podataka
- **Prema rezultatu metode tumačenja**
    - Statistika ili vizualizacija značajki
    - Interni sadržaj modela – tumačenje kroz unutrašnjost modela, npr. težine
    - Zasnovane na primjercima

3. Indukcija pravila (induktivna pravila)

Zadatak: naučiti korisna **if-then** pravila **automatski na temelju skupa podataka**. Najčešća varijanta: na **then** strani se nalazi **ciljna značajka**. Najčešće se koriste za klasifikaciju.

- Prednosti: **najobjašnjiviji modeli od svih**. Nedostatci: **često slabija točnost od drugih modela**

4. Učenje pravila – konjunkcijsko pravilo

- **Tijelo** pravila (dio **if**) sadržava **konjunkciju uvjeta** – uvjet tipično sadrži **usporedbu (op) vrijednosti prediktivne značajke A_i s nekom njenom vrijednosti (=, <, >, ≤, ≥)**
- **Glava** pravila (dio **then**) sadržava **predviđanje**

5. Konjunkcijsko pravilo

- **Pokrivanje pravila**: pravilo pokriva primjerak ako primjerak ispunjava sve uvjete pravila
- **Predviđanje pravila:** ako pravilo pokriva primjerak, glava pravila je predviđena za taj primjerak
- **Konzistentnost skupa pravila:** ako sva pravila u skupu pokrivaju samo primjerke jedne klase
- **Potpunost skupa pravila**: ako su svi primjerci pokriveni nekim pravilom

6. Učenje pravila po principu „razdvoji pa vladaj“

Pravila se grade postepeno i tako da razdvoje primjerke od primjeraka. Algoritam:

1. Započni s praznim modelom T i skupom za učenje E
2. Nauči jedno konzistentno pravilo R iz E i dodaj ga u T
3. Ako je T zadovoljavajuća (najčešće: potpuna) vrati T
4. Inače: **razdvoji** (ukloni primjerke koji pokrivaju R iz skupa E), **vladaj** (vrati se na korak 2)

7. Relaksacija potpunosti i konzistentnosti

Izgradnja skupa pravila koji je kompletan i konzistentan pokazano dovodi do **prenaučenosti**.

- Primjerci koje pokriva pravilo:
    - Uistinu pozitivni: **p** – pozitivni primjerci koje pokriva pravilo
    - Lažno pozitivni: **n** – negativni primjerci koje pokriva pravilo
- Primjerci koje ne pokriva pravilo: (P – broj ukupnih pozitivnih primjera, N – negativnih)
    - Lažno negativni: **P – p** – pozitivni primjerci ne pokriveni pravilom
    - Uistinu negativni: **N – n** – negativni primjerci ne pokrivni pravilom
- **Pokrivanje ili potpora** pravila: **COV = SUP = (p + n) / (P + N)** – najčešće tim bolje što je veće

8. Heuristike pokrivanja

Dodavanje pravila treba: **što više povećati broj pokrivenih pozitivnih primjera** i **što manje povećati broj pokrivenih negativnih primjera.** Česte heuristike: **Točnost, Utežana relativna točnost, Preciznost (pouzdanost)** i **Korelacija**

9. Izbjegavanje prenaučenosti

Učenje koncepata tako da nisu svi pozitivni primjerci pokriveni s pravilom, neki negativni primjerci mogu biti pokriveni s pravilom. **Većina algoritama se fokusira na kraća pravila** koja pokrivaju puno pozitivnih primjera (neke negativne) umjesto duljih pravila s malo poz. primjera.

- **Podrezivanje pravila** – način pojednostavljenja kompleksnih pravila
    - **Predpodrezivanje** – podrezivanje tijekom učenja pravila
    - **Postpodrezivanje** – podrezivanje nakon što je pravilo već izgrađeno
    - Kombinacija jednog i drugog

10. Višeklasna klasifikacija

Većina algoritama prilagođena je samo za binarne klasifikacijske probleme. Rješenje: **binarizacija klasa** – jedan višeklasni problem pretvara se u **skup binarnih problema.**

- U praksi najboljim rješenjem pokazala se binarizacija **one-versus-all.**

11. Preklapanje pravila i problem nepostojanja pravila

Ako se pravila **preklapaju** (pokrivaju istog pojedinca) postoje dva moguća rješenja: **lista odlučivanja** (poredak važnosti pravila) **ili skup odlučivanja** (pravila moraju biti međusobno isključiva ili mora postojati strategija za rješavanje konflikta). Ako primjer nije pokriven nijednim pravilom koristi se **defaultno pravilo** (najčešće dodjela većinskoj klasi).

12. OneRule – algoritam

- Kontinuirane značajke se najprije diskretiziraju odabravši prikladne intervale
- Za svaku značajku:
    - Stvara se unakrsna tablica između vrijednosti prediktivnih značajki i kategoričke ciljne varijable
    - Za svaku vrijednost značajke gradi se pravilo koje predviđa najčešću klasu za primjerke koje imaju tu određenu vrijednost prediktivne značajke
    - Računa se ukupna pogreška pravila (udio primjeraka koji ne pripadaju najčešćoj klasi)
- Izabere se značajka s najmanjom ukupnom pogreškom kao rezultantna značajka (model)
    - Ako dvije značajke imaju istu najmanju pogrešku, odabire se jedna slučajno

13. RIPPER

Zasnovan na **inkrementalnom podrezivanju sa smanjenom pogreškom**. Složeni algoritam učenja klasifikacijskih pravila s dobrom generalizacijom (koristi nekoliko pametnih heuristika).

- Prednosti: brz, dobro radi s numeričkim značajkama i nalazi kompaktan skup pravila.
- Nedostatci: ako je problem složen, nema garancije da će pronaći korisna pravila

14. RIPPER – algoritam (samo prvi slajd)

- Inicijalizira se E kao skup svih primjeraka za učenje
- Za svaku klasu C, **od najmanje do najveće** (po broju primjeraka)
    1. **Izgradnja pravila (BUILD)**
    2. **Optimizacija pravila (OPTIMIZE)**
    3. **Pometanje preostalih primjeraka (MOP UP)**
    4. **Čišćenje loših pravila (CLEAN UP)**

    - Uklanjaju se iz E svi primjerci klase C_i, uči se na primjercima preostalih klasa, zadnja klasa koja ostaje je defaultna klasa

15. Otkrivanje podgrupa

- **Zadatak:** zaključiti **pojedinačna interpretabilna pravila iz podataka**, ranije labelirana s predefiniranim **svojstvom od interesa**.
    - Jedno svojstvo od interesa – binarna, višeklasna ili numerička ciljna klasa
    - Složeno svojstvo od interesa – **pronalazak iznimnog modela**
    - Svojstvo od interesa definira se kao **funkcija kvaliteta** ili **mjera zanimljivosti**
- Algoritmi uče **nekoliko neovisnih if-then pravila** koja opisuju grupe primjeraka istog svojstva
    - Podgrupe trebaju biti **dovoljno velike i zanimljive**
    - Za razliku od općenitih, globalnih algoritama pravila, ne opisuje se nužno cijela klasa, već samo neki primjerci (lokalna pravila)

8. predavanje (Asocijativna pravila)

1. Pronalazak čestih obrazaca i asocijativna pravila
Područje dubinske analize podataka koje ima zadatak pronaći **česte i relevantne kombinacije vrijednosti značajki** (tzv. **obrasce**) u velikim skupovima podataka. Obrasci mogu biti: **skupovi artikala** (engl. **Itemsets**), podstrukture skupova podataka (grafovi) i podsekvence (time-series).
- Na temelju itemsetova najčešće se grade **asocijativna pravila**

2. Itemset
Itemset je skup koji se sastoji od jednog ili više artikala, **neovisno** o transakciji. Podrazumijeva se da svaki artikl naveden u itemsetu ima vrijednost 1. Veličinu itemseta označavamo s k-itemset (2-itemset sadrži 2 artikla).

3. Potpora itemseta
Itemset A je **čest** u skupu podataka ako je njegova **potpora** veća ili jednaka **minimalnom pragu potpore** (hiperparametar algoritma)**. Apsolutna potpora = broj transakcija koje sadrže A.**

4. Asocijativna pravila i relevantnost
Asocijativno pravilo definira se između **dva itemseta A i B** kao implikacija: **A → B** u značenju: ako u transakciji imamo itemset A onda imamo i itemset B.
- **Relevantnost** asocijativnog pravila numerički je opisana pomoću mjere **pouzdanosti** pravila.
- **Minimalni prag pouzdanosti** je također hiperparametar.

5. Asocijativna pravila – cilj
- **Cilj:** za dani skup transakcija T otkriti sva asocijativna pravila ($A \rightarrow B$) čiji itemsetovi (dovoljno je promatrati itemset $A \cup B$) imaju apsolutnu potporu veću ili jednaku minimalnom pragu potpore i isto tako za pouzdanost.
- Pretraživanje potpore je računski zahtjevnije od određivanja pouzdanosti, stoga je naglasak algoritma na **pronalasku čestih itemsetova**.
- Ukupan broj mogućih asocijativnih pravila **eksponencijalno** ovisi o broju artikala d u podacima

6. Algoritam Apriori
Temeljni iscrpni algoritam za učenje asocijativnih pravila na temelju pronađenih čestih itemsetova. **Prednost:** garantirano pronalazi sve itemsetove koji imaju potporu $\geq$ min_sup i pouzdanost $\geq$ min_conf. **Nedostatak:** spor, ne optimizira pretraživanje skupa svih itemsetova.

7. Algoritam Apriori – ideja

- Pretraga za itemsetovima počinje od najopćenitijih obrazaca – pojedinačnih artikala, pretragom u širinu
- Iterativno se računa apsolutna potpora itemsetova kandidata i pohranjuju se za iduću iteraciju česti itemsetovi
- U idućoj iteraciji, oni itemsetovi kandidati koji imaju barem jedan nečesti podskup se ne razmatraju
- Konačno, algoritam testira sva česta asocijativna pravila i odabire ona koja su pouzdana

8. Algoritam Apriori – primjer

Pogledati 8. prezu – slajdovi 17-19

9. Algoritam PCY

Algoritam pronalaska čestih itemsetova s manjom vremenskom složenosti od Apriorija. Razmatra problem kandidatnih itemsetova kod Apriorija – nakon inicijalnog pronalaska čestih 1-itemsetova, potrebno je razmotriti **sve** 2-itemsetove da se vidi koji su česti.

- Osnovna ideja: koristiti ***hashing* funkciju** prilikom prvog prolaska kroz skup – kada se otkrivaju česti 1-itemsetovi
- U prvom prolasku razmatraju se transakcije redom te se:
    - pobrojavaju pojedinačni artikli (kao kod Apriorija) – **broji im se potpora** i
    - izgrađuju parovi **čestih** artikala (2-itemsetovi) u pojedinačnoj transakciji, na njima se primijeni hashing funkcija i prati se broj pojavljivanja svake pojedinačne hash vrijednosti
    - specifičnije, za praćenje broja pojavljivanja vrijednosti parova koristi se **izmijenjena hash-tablica** (svaka lokacija u tablici odgovara jednom hashu i sadrži jedan cijeli broj – koji broji koliko puta se došlo na tu lokaciju – **ne pamte se parovi, samo broj posjeta**)
- Odlučujući faktor u tome je li PCY brži ili ne u odnosu na Apriori je **postotak mogućih parova kandidata koje eliminira tehnika raspršenog adresiranja. Najveći dobitak algoritma je u ranoj fazi – generiranje 2-itemsetova.**

10. Algoritam FP-growth

Spada u grupu *Pattern Growth* algoritama koji najprije skeniraju bazu kako ne bi razmatrali sve kandidate. Algoritam koji koristi **stablastu strukturu podataka FP-Tree** za sažeti prikaz ulaznih podataka i pronalaženje čestih itemsetova koja se konstruira čitanjem jedne po jedne transakcije i bilježenjem svake transakcije u nekoj grani stabla.

- Budući da transakcije često imaju neke zajedničke artikle, njihovi putovi u stablu se dijelom preklapaju i time se ostvaruje sažimanje velike količine podataka.
- FP-growth može biti i **nekoliko redova veličine brži od Apriorija**

11. Algoritam EFIM

Spada u grupu algoritama za **otkrivanje visokokorisnih itemsetova**, smatra se state-of-the-art algoritmom. Za razliku od transakcija koje samo imaju oznaku je li artikl kupljen ili ne, HUIM baze sadržavaju uz svaki kupljeni artikl i **informaciju o težini**, koja govori koliki je jedinični profit za dotični artikl. Zadatak je pronaći sve itemsetove koji imaju **korisnost** (ovisnu o težini) **veću od minimalno zadane.**

- Glavna značajka: **EFIM koristi nekoliko optimizacija (heuristika) za pretraživanje prostora itemsetova** koje mu omogućuju bitno bolje rezultate od konkurentskih algoritama.

12. Algoritam EFIM – definicija problema HUIM

- Korisnost artikla i u transakciji T_c označava se s: $u(i, T_c)$ i iznosi $p(i) \times q(i, T_c)$ ako je $i \in T_c$, gdje je $q(i, T_c)$ **unutarnja korisnost** artikla i u transakciji T_c (tzv. kupljena količina), a $p(i)$ je **vanjska korisnost** artikla i (tzv. jedinična cijena, profit).
- Itemset X je visokokoristan ako je $u(X) \geq$ minutil, a inače je niskokoristan
- **Problem HUIM:** potrebno je pronaći sve visokokorisne itemsetove u skupu podataka (puno teži problem od svih čestih itemsetova)

13. Algoritam EFIM – primjer

Pogledati 8. prezu – slajd 40

9. predavanje (Dubinska analiza vremenskih nizova)

1. Dubinska analiza vremenskih nizova
Područje dubinske analize podataka koje analizira **vremenske nizove podataka** s ciljem pronalaska zanimljivih obrazaca. U elektrotehnici se poistovjećuje s **jednodimenzionalnim signalom.**

2. Vremenski niz podataka
Vremenski niz je kolekcija vrijednosti koja je dobivena slijednim mjerenjima tijekom vremena
- U analizi vremenskih nizova fokus je na **diskretnim** mjerenjima dobivenima u **jednolikim** vremenskim intervalima

3. Glavni zadaci analize vremenskih nizova
- **Upit prema sadržaju** – pronalazak vremenskih nizova **najsličnijih** zadanom vremenskom nizu
- **Detekcija anomalija** – otkrivanje **neuobičajenih podnizova** u vremenskom nizu
- **Otkrivanje motiva** – otkrivanje svih **tipičnih**, **nepreklapajućih podnizova** u duljem vrem. nizu
- **Predviđanje vrijednosti** – predviđanje **sljedećih k vrijednosti niza** koje su najizglednije
- **Grupiranje** – pronalazak određenog broja **grupa** sličnih vremenskih nizova
- **Klasifikacija** – na temelju značajki vremenskog niza, pridruži nekom neoznačenom vremenskom nizu jednu od predefiniranih **oznaka klase**
- **Segmentacija ili reprezentacija** – za izvorni vremenski niz izgradi se njegov **nižedimenzijski model** takav da ga on blisko **aproksimira**

4. Terminologija
Vremenski niz T je uređeni slijed o n realnih brojeva. Rezultat je opažanja nekog procesa pri čemu su mjerenja vrijednosti napravljena u **vremenskim trenucima** s **pravilnim** ili **nepravilnim** međusobnim razmakom. Za pravilna mjerenja za svaki niz definirana je **stopa uzrokovanja.** Vremenski niz može biti **univarijatan** ili **multivarijatan.** Može biti **konačan** ili **polubeskonačan**.
- **Mjera sličnosti** između vremenskih nizova T i U je funkcija koja prima dva niza na ulazu i vraća udaljenost između tih nizova.
- **Mjera sličnosti podniza** predstavlja udaljenost između podniza T i njemu najbližeg podniza S' u nizu S.
- Vremenski niz može biti **stacionaran** (ne mijenjaju se **srednja vrijednost, varijanca i autokovarijanca**) i **nestacionaran**. Također može biti i **periodički i neperiodički.**
- Podjela po periodičnosti: **skoro periodički, kvaziperiodički, kaotičan niz, slučajan niz**

5. Predobrada vremenskog niza – uklanjanje šuma i skaliranje
Zadatak je ukloniti različite izvore šuma koji djeluju na niz (**digitalni filtri, određivanje pragova valića, korištenje transformacija podataka**). Skaliranje se koristi za uspoređivanje, grupiranje i klasifikaciju. Potreban oprez ako je značajan dio signala šum kako se on ne bi prenaglasio.

6. Predobrada vremenskog niza – ponovno uzrokovanje i interpolacija
Ponovno uzorkovanje – npr. umjesto 100 Hz uzrokovanja niz se preuzrokuje na 50 Hz. Korisno za dobivanje više vremenskih nizova iste duljine.
Interpolacija – zasniva se na procesu pronalaska odgovarajuće kontinuirane funkcije na temelju diskretnih točaka vremenskog niza s ciljem nadomještanja nepoznatih vrijednosti. Najčešće interpolacije: **konstantna, najbliži susjed, linearna, kvadratna, kubni spline**

7. Predobrada vremenskog niza – podjela na prozore
Podjela niza od n mjerenja u **prozore** širine k << n mjerenja. Važan pretkorak za algoritme grupiranja, klasifikacije, predviđanja i sl. Podjela može biti **ovisna o periodičnosti i drugim značajkama niza**. Podjela može biti s **preklapanjem prozora** ili bez preklapanja.

8. Reprezentacijske metode
Odnosi se na izvlačenje najvažnijih informacija iz vremenskog niza. Zahtjevi na reprezentacijsku metodu vremenskog niza su: **značajna redukcija dimenzionalnosti (broja točaka), naglasak na temeljnim karakteristikama niza, niski vremenski zahtjevi na računanje reprezentacije, neosjetljivost na šum ili njegovo implicitno uklanjanje.**
Klasifikacija reprezentacijskih metoda: **neadaptivne s obzirom na podatke, adaptivne s obzirom na podatke** i **zasnovane na modelu podataka**

9. Neadaptivne reprezentacijske metode
Parametri transformacije podataka ostaju isti za svaki niz neovisno o njegovoj prirodi.
Primjeri: **DFT, DWT, DCT, PAA, HHT**

10. Adaptivne reprezentacijske metode
Definiraju parametre transformacije koji će biti zadržani u reprezentaciji **ovisno o konkretnim podacima.** Neke metode su inherentno adaptivne, npr. **SVD**. **Metode simboličke reprezentacije** su većinom adaptivne – vremenski niz **diskretiraju u niz simbola**.
Primjeri metoda: **SAX** – kao osnovu ima PAA, diskretizira srednje vrijednosti u prozorima u abecedu s jednakom frekvencijom simbola, ***Shapeleti*** – podnizovi koji su **maksimalno reprezentativni za neku klasu** i stoga mogu u potpunosti diskriminirati klase.

11. Reprezentacijske metode zasnovane na modelu
Pretpostavljaju da je opaženi niz **rezultat nekog modela koji se nalazi u pozadini** i koji ga generira, zadatak je pronaći parametre tog modela kao reprezentaciju. Metode: **Modeliranje ekstrakcijom značajki**, **Modeliranje učenjem značajki**, **ARMA modeli**, **Skriveni Markovljevi modeli**

12. Klasifikacijski algoritam: HIVE-COTE

State-of-the-art klasifikator za općenite vremenske nizove. Uključuje **37** klasifikatora izgrađenih na **5** podatkovnih reprezentacija univarijatnih vremenskih nizova. Prednosti: **vrlo točna klasifikacija vremenskih nizova**. Nedostatci: **velika sporost**.

13. HIVE-COTE

- Meta-ansambl uključuje 5 modula:
    - **Elastic Ensemble** – uključuje mjere sličnosti s cijelim nizom
    - **HESCA** – heterogeni ansambl standardnih klasifikacijskih algoritama
    - **Time Series Forest** – ansambl zasnovan na intervalnim značajkama
    - **BOSS** – ansambl zasnovan na rječniku
    - **RISE** – spektralni ansambl
- Odluka cijelog meta-ansambla donosi se kao utežani probabilistički model

14. ARIMA

Statistički parametarski model namijenjen za analizu i predviđanje vremenskih nizova. Predikcija ovisi o parametrima (p, d, q). **p** – **autoregresija** je zavisnost buduće vrijednosti promatranog niza o određenom broju prethodnih vrijednosti niza. **d** – **diferenciranje** označava da se razmatra d-ta razlika između vrijednosti niza i time se uklanja trend. **q** – **pomični prosjek** razmatra ovisnost između vrijednosti niza i rezidualne pogreške modelom **utežanog pomičnog prosjeka**.

15. PROPHET

Generalizirani aditivni model za predikciju vrijednosti vremenskih nizova s tri glavne komponente modela: trendom, sezonalnošću i praznicima. Pokušava „fitati“ i linearne i nelinearne funkcije tako da najbolje odgovaraju opaženom vremenskom nizu.

16. Modeliranje niza ekstrakcijom značajki

Moguće je izračunati različite **značajke vremenskog niza** koje će ga reprezentirati te ih potom iskoristiti za učenje klasifikatora ili regresora. U ovom pristupu važno je da broj izlučenih značajki m bude **značajno manji** od broja točaka u nizu n kako bi se smanjila dimenzionalnost problema, ali **ne premali** kako bismo pokrili sve potrebne informacije.

- Neke vrste značajki: **linearne, nelinearne, entropijske, prostorne itd.**

10. predavanje (Duboko učenje u dubinskoj analizi podataka)

1. Duboko učenje
Područje strojnog učenja koje koristi različite **arhitekture dubokih neuronskih mreža** s odgovarajućim postupcima učenja. Zasnovano na konceptu **učenja značajki** ili **učenja reprezentacije**.

2. Umjetne neuronske mreže
Mrežu čine umjetni neuroni povezani u slojeve (**1 ulazni sloj, 1 ili više skrivenih** i **1 izlazni sloj**). **Memoriju mreže čine iznosi težina veza** između neurona u slojevima. Umjetni neuron je model koji uzima u obzir vektore **težina w** prema neuronima najčešće prethodnog sloja i određene **aktivacijske funkcije**. Mreže mogu imati različite varijacije: **povezanosti slojeva, aktivacijske funkcije, algoritama učenja i optimizatora, funkcija gubitka koju treba minimizirati.**

3. Umjetne neuronske mreže – aktivacijske funkcije
Najčešće rade **nelinearnu transformaciju** vrijednosti modela težina u nekom neuronu. Izbor funkcije neki put značajno utječe na uspješnost neuronske mreže.

4. Algoritmi učenja i optimizacija hiperparametara
Najpoznatiji i najšire korišteni algoritam za učenje neuronske mreže je **stohastički gradijentni spust**. Koristi se algoritam propagacije **pogrešaka predviđanja** unazad. Time se revidiraju **težine veza**.
- Optimizacija se radi: **slučajno, ručnim izborom, pretragom po rešetki** ili **Bayesovom optimizacijom**
- Hiperparametri neuronske mreže: **Broj slojeva i neurona**, **Dropout**, **Aktivacijske funkcije**, **Stopa učenja**, **Broj epoha učenja**, **Batch size**, **Hiperparametri optimizatora**

5. Funkcija gubitka
Završni sloj neuronske mreže koji specificira kako učenje **penalizira razliku između predviđenog izlaza iz mreže i stvarnog izlaza.** Na temelju tih funkcija dobivaju se gradijenti. Neke on najčešćih funkcija su **unakrsna entropija** i funkcija **srednje kvadratne pogreške.**

6. Zašto duboke neuronske mreže uspijevaju?
Arhitekturni razlozi: Uvođenje **specijaliziranih arhitektura** za određene namjene, pronalazak **novih aktivacijskih funkcija**, istraživanje **regularizacijskih metoda**, **optimizacijski algoritmi učenja** i **algoritmi predobrade.**

7. Umjetne neuronske mreže – konvolucijska
Pogledati 10. prezu – slajdovi 16-21 – previše za pisati

8. Umjetne neuronske mreže – rekurentne
Mreže koje mogu pamtiti prijašnje ulaze i sljedivost podataka. **Rekurentni sloj** – izlaz neuronske mreže ovisi o nizu aktivacija ulaznih vrijednosti koje imaju određeni poredak. Rekurentni sloj prima vrijednost x i aktivaciju prethodnog sloja h te množeći ih odgovarajućim težinama i koristeći aktivacijsku funkciju daje novu aktivaciju neurona sljedećem sloju.
- **Vrste rekurentnih slojeva: Vanilla RNN**, **GRU** (brz i točan na kratkotrajnim sekvencama), **LSTM** (dobar za učenje dugotrajnih ovisnosti na većim skupovima podataka)

9. Umjetne neuronske mreže – rekurentne – LSTM
Tri glavna dijela: **Vrata zaborava** – odlučuje koji dio informacije iz prethodnog sloja treba maknuti. **Vrata ulaza** – odlučuje koju informaciju treba propustiti. **Vrata izlaza** – omogućuje propuštenoj informaciji da utječe na izlaz trenutačnog stanja.

10. Umjetne neuronske mreže – rekurentne – slaganje blokova
- 4 varijante: **Osnovna 1 na 1**, **Varijanta 1 na N** (npr. tekstni opis nekog ulaza), **Varijanta N na 1** (npr. analiza sentimenta), **Varijanta N na N** (npr. prevođenje)

11. Umjetne neuronske mreže – autoenkoderi
Neuronska mreža za **nenadzirano učenje** koja se sastoji od dva simetrična dijela: **kodirajućeg sloja** i **dekodirajućeg sloja**. **Glavni zadatak kodera** je naučiti unutarnju niskodimenzionalnu reprezentaciju ulaznog sloja tzv. kod, a **glavni zadatak dekodera** je transformirati kod u izmijenjenu verziju ulaznih podataka.

12. Transformeri
Slično RNN-u primjenjuju se na sekvencijalne podatke, no transformer istovremeno radi nad svim ulaznim podacima, npr. primjenjuju se na sve riječi u rečenici. Transformerska arhitektura je **kodersko-dekoderska arhitektura** s određenim brojem (stogom) kodera i dekodera koja djeluje po **principu samopozornosti**, uklanjajući potrebnu za rekurentnim blokovima.
- Samopozornost koristi sloj pozornosti za pristup stanju bilo kojeg bloka u sekvenci

13. Problemi dubokih neuronskih mreža
- **Nedostatak interpretabilnosti** – crne kutije
- **Složenost postupka učenja**
- **Problem nestajućeg gradijenta**
- **Problem eksplodirajućeg gradijenta**

14. Obrada prirodnog jezika – BERT

Algoritam za predučenje i fino podešavanje modela zasnovan na transformerskoj arhitekturi od 12 kodera / dekodera. BERT na ulazu prima konkatenaciju dva segmenta duljina M i N ukupne duljine manje od hiperparametra T.

- Tijekom učenja BERT ima dva cilja: **modeliranje maskiranog jezika (MLM)** i **predikciju sljedećeg segmenta (NSP)**
- Optimizator je Adam, dropout iznosi 0.1 i aktivacijska funkcija je GELU

15. Analiza vremenskih nizova – ROCKET

State-of-the-art algoritam za **klasifikaciju vremenskih nizova** po pogledu točnosti i brzine izgradnje modela. Nije neuronska mreža, ali je motiviran njihovim uspjehom.

- ROCKET **transformira** vremenski niz koristeći velik broj **slučajnih konvolucijskih jezgri** (10 000)
- Transformirane značajke se koriste za učenje jednostavnog klasifikatora