# Uvod 1

**1. Što je dubinska analiza podataka?**

Dubinska analiza podataka je proces pronalaženja skrivenih i korisnih informacija u podacima. (Slide 21)

**2. Klasifikacija vrsta strojnog učenja**

- Četiri vrste strojnog učenja (Slide 25):

1. Nadzirano učenje
2. Nenadzirano učenje
3. Polunadzirano učenje
4. Podržano učenje

Slika na kojoj se prikazuje tekst, dijagram, snimka zaslona, Font

Opis je automatski generiran

**3. Šest glavnih problema od interesa za dubinsku analizu podataka (Slide 26)**

- Otkrivanje anomalija (engl. anomaly detection)

- Modeliranje ovisnosti (engl. dependency modeling)

- Grupiranje (engl. clustering)

- Klasifikacija (engl. classification)

- Regresija (engl. regression)

- Sažimanje (engl. summary)

**4. Modeli procesa dubinske analize podataka (Slide 33):**

1. KDD (engl. Knowledge Discovery in Databases) – 1996.

2. CRISP-DM (engl. CRoss-Industry Standard Process for Data Mining) – 2000.

3. ASUM-DM (engl. Analytics Solutions Unified Method) – 2015.

4. CRISP-ML(Q) (engl. CRoss-Industry Standard Process model for the development of Machine Learning applications with Quality assurance methodology) – 2021.

**5. CRISP-DM**

- CRISP-DM je industrijski pristup standardizaciji projekata dubinske analize podataka. Proces je razložen u šest faza:

1. Razumijevanje poslovnih potreba

2. Razumijevanje podataka

3. Priprema podataka

4. Modeliranje

5. Vrednovanje

6. Puštanje u pogon (Slide 36)

- Detaljnije faze i generički zadatci su opisani na Slide-ovima 37, 38, i 39.

**6. ASUM-DM (Slide 41)**

- ASUM-DM je IBM-ovo proširenje metodologije CRISP-DM. Uključuje bolje razrađen operativni dio puštanja u pogon i poslovni aspekt upravljanja projektom. Faze su:

1. Analiza

2. Oblikovanje

3. Konfiguracija i izgradnja

4. Puštanje u pogon

5. Djelovanje i optimizacija

**7. CRISP-ML(Q) (Slide 45)**

- CRISP-ML(Q) je suvremeni model procesa namijenjen razvoju modela strojnog učenja. Sastoji se od šest koraka:

1. Razumijevanje poslovne strane i podataka

2. Priprema podataka

3. Modeliranje

4. Vrednovanje modela

5. Puštanje u pogon

6. Nadzor i održavanje

# 2. Priprema podataka

**1. Priprema podataka**

- Priprema podataka (engl. data preparation, data handling, data wrangling) slijedi nakon preuzimanja izvornih podataka i uključuje sve korake do početka analize. U širem smislu, uključuje pristup, pregled i izbor izvornih podataka. (Slide 3)

**2. Proces pripreme podataka (ali ne: Proces pripreme podataka – CRISP-DM)**

- Proces pripreme podataka je iterativan, ad hoc, ovisan o prostoru problema, prostoru rješenja i dostupnim podacima, te zahtijeva puno razmišljanja. Automatizacija je teška. (Slide 5)

- Ključni koraci: Otkrivanje podataka, Karakterizacija podataka, Izgradnja skupa podataka za modeliranje. (Slide 6)

**3. Nedostajući podaci – vrste nedostajućih podataka**

- Nedostajuće ali poznate vrijednosti: Vrijednosti koje nisu unesene u skup podataka, ali postoje u stvarnom procesu.

- Prazne i nepoznate vrijednosti: Vrijednosti koje se ne mogu pretpostaviti u stvarnom svijetu i nisu unesene. (Slide 22)

**4. Nedostajući podaci – rješavanje problema**

- Osoba koja modelira podatke mora imati kontrolu nad metodom rješavanja problema nedostajućih podataka.

- Alati ponekad nemaju transparentnu metodu rješavanja ovog problema, što može dovesti do pristranosti. (Slide 23)

- Metode:

* zanemarivanje primjeraka s nedostajućim vrijednostima
* zamjenu s nekom drugom vrijednosti (srednja vrijednost, medijan, mod, konstantna vrijednost, nova kategorija)
* linearna regresija
* algoritam k-najbližih susjeda
* nelinearna regresija. (Slides 24-30)

**5. Stršeći podaci**

- Stršeći podaci su podaci koji odskaču daleko izvan uobičajenih vrijednosti za određene značajke.

Razlozi mogu biti:

* neispravan unos
* greške mjerenja
* greške obrade podataka
* prirodno stanje. (Slide 34)

- Postupci otkrivanja uključuju vizualizaciju, statističke postupke (z-skor, vjerojatnosni modeli, linearna regresija), algoritme nenadziranog strojnog učenja (temeljeni na udaljenosti, gustoći, specifični za velike skupove podataka). (Slides 35-41)

**6. Stršeći podaci – rješavanje problema**

- Utvrditi je li podatak prirodan ili ne.

- Ako je prirodan ali smeta izgradnji modela, koristiti normalizaciju vrijednosti varijabli.

- Ako nije prirodan, tretirati ga kao nedostajući podatak i primijeniti odgovarajući postupak za nedostajuće podatke. (Slide 42)

**7. Šumoviti podaci**

- Šum u podacima je prisutan u svim podacima koji su rezultat mjerenja putem određenih senzora. Podatak = pravi signal + šum. (Slide 44)

- Postoje postupci za filtriranje šuma, ovisno o konkretnom problemu (npr. korekcija pomaka nulte linije kod EKG-a, pojasno propusno filtriranje kod EEG-a). (Slide 45)

**8. Monotone značajke**

- Monotone značajke su one čija vrijednost raste ili se smanjuje bez ograničenja. Najčešći primjeri su značajke povezane s protjecanjem vremena. Problem je nemogućnost dobivanja korisne informacije iz takve serije. (Slide 46)

**9. Konstantne značajke**

- Vrijednost ovih značajki je ista u cijelom skupu podataka, varijanca im je 0. Nisu informacijski relevantne i treba ih ukloniti. (Slide 47)

**10. Koraci pripreme podataka – pojednostavljeno**

- Sudjelovati u procesu prikupljanja i isporuke skupa podataka.

- Pomno pregledati skup podataka kako bi se ustanovili svi problemi.

- Raspisati sve uočene probleme i predložiti rješenja za njihovo uklanjanje.

- Prodiskutirati uočeno i predložena rješenja s relevantnim dionicima.

- Provesti usvojena rješenja. (Slide 51)

# 3. Transformacija podataka i inženjerstvo značajki

**1. Značajke**

- Transformirane varijable koje se koriste na ulazu metoda strojnog učenja nazivaju se značajkama (engl. feature), a cijeli primjerak naziva se vektor značajki (engl. feature vector). (Slide 7)

**2. Značajke – razlozi korištenja**

- Razlozi za korištenje značajki uključuju:

* sažetost (smanjenje broja sirovih podataka za učinkovitije korištenje algoritama)
* informativnost (matematički odnosi između varijabli otkrivaju zanimljive informacije)
* interpretabilnost (ljudi bolje razumiju značajke u odnosu na sirove podatke). (Slide 9)

**3. Diskretizacija numeričkih vrijednosti**

- Diskretizacija numeričkih varijabli uključuje pretvorbu numeričkih vrijednosti u kategoričke vrijednosti (engl. binning). Neki algoritmi zahtijevaju diskretne vrijednosti, a postupci diskretizacije uključuju:

* podjelu u intervale jednake širine
* jednake frekvencije
* k-srednje vrijednosti
* minimizaciju entropije. (Slides 14-18)

**4. Pretvorbe kategoričkih varijabli**

- Pretvorbe kategoričkih varijabli uključuju izravnu pretvorbu u numeričke vrijednosti (label encoding) i pretvorbu u binarne varijable (one-hot encoding). Ove metode su potrebne jer mnogi algoritmi strojnog učenja ne mogu raditi direktno s kategoričkim vrijednostima. (Slide 19)

**5. Normalizacija vrijednosti varijabli**

- Normalizacija je potrebna kada su značajke mjerene na različitim skalama. Najčešći postupci uključuju:

* decimalno skaliranje
* normalizaciju Min-Max
* normalizaciju z-skorom
* podrezivanje (clipping). (Slides 20-21)

**6. Uklanjanje nebitnih i redundantnih značajki**

- Nebitne značajke ne poboljšavaju uspješnost modela, a uključuju **monotone**, **konstantne**, **duplicirane** i **nekorelirane** značajke.

- Redundantne značajke su one koje ne poboljšavaju model zbog prisutnosti drugih značajki i određuju **se korelacijskom analizom** i **Markovljevim** **pokrivačem**.

(Slides 24-27)

**7. Izgradnja značajki**

- Izgradnja značajki uključuje iterativnu primjenu različitih operatora za izgradnju novih značajki.

Podjela pristupa za izgradnju značajki:

* zasnovani na podacima
* zasnovani na hipotezi
* zasnovani na znanju
* hibridni pristupi. (Slides 30-32)

**8. Analiza glavnih komponenti!**

- Analiza glavnih komponenti (PCA) je linearna tehnika smanjenja dimenzionalnosti koja koristi vlastite vektore matrice kovarijance za preslikavanje izvornih podataka u novi prostor značajki, pokrivajući najveću varijabilnost u podacima. (Slides 35-37)

**9. Višedimenzijsko skaliranje!**

- Višedimenzijsko skaliranje (MDS) je skup metoda koje predstavljaju mjere sličnosti između parova objekata u visokodimenzionalnom prostoru kao metričku udaljenost između točaka u niskodimenzionalnom prostoru, minimizirajući mjeru stresa. (Slides 38-39)

**10. t-SNE!**

- t-SNE traži niskodimenzionalnu strukturu koja čuva svojstva grupiranja u višoj dimenziji. Koristi Gaussove zajedničke vjerojatnosti za mjerenje sličnosti u izvornom prostoru i Studentove t-razdiobe za ugrađeni prostor. (Slides 40-41)

# 4. Odabir značajki

**1. Odabir značajki**

- Postupak smanjenja dimenzije (broja varijabli) skupa podataka, zadržavajući interpretaciju značajki. Cilj je zadržati ili poboljšati rezultat modeliranja, pojednostaviti model i smanjiti vrijeme potrebno za izgradnju modela. (Slide 4)

**2. Ciljevi**

- Specifični ciljevi odabira značajki uključuju:

* Najmanji podskup značajki koji daje bolje rezultate nego početni skup.
* Najmanji podskup značajki koji daje približno jednake rezultate kao početni skup.
* Bilo koji podskup značajki koji daje najbolje rezultate.
* Rangiranje značajki prema važnosti za zadani cilj.
* Odabir točno k od početnih M značajki koje daju najbolji rezultat. (Slide 6)

**3. Relevantnost i redundantnost značajki**

- Postoje četiri tipa značajki:

* snažno relevantne
* slabo relevantne ali neredundantne
* slabo relevantne i redundantne
* irelevantne

Relevantne značajke su bitne za optimalni podskup, dok redundantne značajke ne poboljšavaju model jer njihovu informaciju već pokrivaju druge značajke.

(Slides 9-12)

**4. Podjela metoda odabira značajki**

- Podjela metoda odabira značajki uključuje:

* Tablične metode:
* filterski postupci
* postupci omotača
* ugrađeni postupci
* hibridni postupci.
* Strukturne metode:
* graf struktura
* struktura stabla
* grupna struktura. (Slide 13)

**5. Dva pristupa odabiru značajki**

- Pristupi odabiru značajki uključuju:

* Univarijatni pristup:
* određivanje relevantnosti pojedinačnih značajki
* rangiranje prema mjeri
* uklanjanje ako ne zadovoljavaju prag.
* Multivarijatni pristup:
* određivanje relevantnosti i redundantnosti podskupa značajki
* računanje značaja podskupa
* pronalazak redundantnih značajki. (Slide 14)

**6. Filterski postupci**

- Filterski postupci ne koriste algoritam strojnog učenja za odabir značajki, već definiraju mjeru važnosti značajke ili skupa značajki za opis ciljne značajke. (Slide 15)

**7. Filterski postupci za pojedinačne značajke**

- Filterski postupci za **pojedinačne** značajke uključuju:

* informacijske mjere (informacijska dobit, simetrična nesigurnost, korelacijski koeficijent)
* obitelj metoda Relief
* hi-kvadrat
* Fisherov skor. (Slide 17)

**8. Informacijska dobit**

- Informacijska dobit (IG) za podjelu značajke X prema vrijednostima od značajke Y računa se kao razlika između entropije značajke X i uvjetne entropije značajke X uz poznavanje Y: \( IG(X|Y) = H(X) - H(X|Y) \). (Slide 18)

**9. Simetrična nesigurnost**

- Simetrična nesigurnost (SU) ograničava vrijednosti informacijske dobiti na interval [0,1], gdje je 1 potpuna prediktivnost, a 0 neovisnost. (Slide 19)

**10. Obitelj metoda Relief (prvi slajd)**

- Metode obitelji Relief rangiraju značajke uzimajući u obzir **ovisnost među prediktivnim značajkama**, temeljene su na **najbližim susjedima** za određivanje mjere korisnosti pojedinačne značajke. (Slide 20)

**11. mRMR**

- Minimalno redundantne maksimalno relevantne značajke (mRMR) odabiru se tako da skup značajki ima maksimalnu relevantnost prema ciljnoj značajki i minimalnu redundantnost prema drugim značajkama (Slides 24-25)

**12. Postupci omotača**

Postupci omotača koriste algoritam strojnog učenja za evaluaciju podskupa značajki kako bi se donijela odluka o njegovoj važnosti. Sporiji su ali točniji od filterskih metoda. (Slides 28-29)

**13. Pohlepno pretraživanje**

Pohlepno pretraživanje (greedy search) dodaje ili uklanja značajke koje najviše povećavaju točnost modela u svakom koraku, zaustavljajući se kad dođe do degradacije točnosti. (Slide 31)

**14. Slijedna plutajuća selekcija**

Slijedna plutajuća selekcija (sequential floating selection) iterativno dodaje i uklanja značajke ovisno o povećanju točnosti modela, s praćenjem da ne dođe do beskonačnih petlji. (Slide 32)

**15. Ugrađeni postupci**

Ugrađeni postupci temelje se na algoritmu strojnog učenja gdje unutarnja struktura modela oslikava važnost značajki.

Primjeri uključuju:

* slučajnu šumu
* logističku regresiju s penalizacijom
* stroj s potpornim vektorima. (Slide 35)

**16. Odabir značajki kod slučajne šume**

Kod slučajne šume, značajke se ocjenjuju prema tome koliko smanjuju nečistoću u čvorovima stabla, pri čemu značajke bliže korijenu stabla obično više smanjuju nečistoću. (Slide 37)

**17. Hibridni postupci**

Hibridni postupci kombiniraju najbolje značajke **filtara** i **omotača**, najčešće primjenjujući filtar kako bi se **smanjio prostor značajki**, a zatim omotač za **pronalaženje optimalnog podskupa značajki**. (Slide 41)

# 5. Nebalansiranost podataka i pomak koncepta

**1. Nebalansiranost podataka**

- Problem skupa podataka u kojem postoji neravnoteža (nebalansiranost) u broju primjeraka pojedinih klasa ciljne značajke. To otežava izgradnju modela koji će jednako dobro klasificirati i većinske i manjinske klase. (Slide 4)

**2. Stupanj nebalansiranosti klasa**

- Ne postoji strogo definirani kriterij za postojanje nebalansiranosti klasa. Omjer većinske klase prema manjinskoj klasi (npr. 2:1, 9:1, 99:1) može varirati i utjecati na sposobnost ispravne klasifikacije. Viši omjeri često ukazuju na veće poteškoće za klasifikatore. (Slides 6-7)

**3. Ponovno uzorkovanje**

- Izmjena razdiobe primjeraka ciljne značajke po klasama kako bi bila približno jednaka za sve klase. To uključuje:

* naduzorkovanje
* poduzorkovanje
* hibridne metode. (Slide 10)

**4. Naduzorkovanje**

- Generiranje novih primjeraka za manjinsku klasu kako bi se postigla ravnoteža s većinskom klasom. Metode uključuju:

* kopiranje postojećih primjeraka
* slučajno naduzorkovanje s ponavljanjem
* generiranje sintetskih primjeraka. (Slides 11-12)

**5. Naduzorkovanje zasnovano na SMOTE-u**

- SMOTE (Synthetic Minority Over-sampling TEchnique) generira sintetske primjerke na temelju postojećih primjeraka manjinske klase pomoću najbližih susjeda. Sintetski primjerak se nalazi na liniji koja povezuje primjerak i njegove susjede. (Slide 14)

**6. Poduzorkovanje**

- Uklanjanje primjeraka većinske klase kako bi se postigla ravnoteža s manjinskom klasom. Metode uključuju:

* slučajno poduzorkovanje
* metodu NCL (Neighborhood Cleaning Rule)
* metodu TL (Tomek Links). (Slide 18)

**7. Poduzorkovanje – metoda NCL**

- Metoda NCL koristi pravilo uređenih najbližih susjeda (ENN) kako bi uklonila primjerke većinske klase koji imaju barem jednog susjeda iz manjinske klase. (Slide 19)

**8. Poduzorkovanje – metoda TL**

- Metoda Tomekovih poveznica uklanja primjerke većinske klase koji formiraju Tomekove veze s manjinskom klasom. Tomekove veze su parovi primjeraka različitih klasa koji su najbliži susjedi jedan drugom. (Slide 20)

**9. Hibridni postupci uzorkovanja**

- Kombiniranje poduzorkovanja i naduzorkovanja. Jednostavan pristup je slučajna ravnoteža (kombinacija slučajnog poduzorkovanja i SMOTE-a). Složeniji postupci uključuju SMOTE+ENN i SMOTE+TL. (Slide 22)

**10. Učenje osjetljivo na cijenu**

- Osnovna pretpostavka je da neispravna klasifikacija primjerka manjinske klase ima veću cijenu. Cijena se izražava u obliku matrice cijene. Svakom primjerku se dodjeljuje težina prema cijeni iz matrice. (Slides 25-26)

**11. Korištenje prikladnih mjera vrednovanja modela**

- Za nebalansirane skupove podataka, ukupna klasifikacijska točnost (ACC) nije dobra mjera. Preporučuju se mjere poput preciznosti (precision), odziva (recall) i F1-mjere. (Slide 30)

**12. Pomak koncepta u podacima**

- Pomak koncepta označava promjenu statističkih svojstava ciljne značajke tijekom vremena. To može biti nagli ili inkrementalni pomak, a ogleda se u smanjenju točnosti modela. (Slides 32-33)

**13. Online učenje**

- Algoritmi online učenja obrađuju svaki primjerak samo jednom bez pohranjivanja ili ponovne obrade. Ovi algoritmi donose odluku kada primjerak postane dostupan, omogućujući prilagodbu modela s novim podacima. (Slide 35)

**14. Detektori pomaka zasnovani na monitoriranju**

- Ovi detektori koriste statističke testove za praćenje promjena u razdiobi ciljne značajke na temelju performanse pogreške modela. Uključuju metode poput DDM, ADWIN i EDDM. (Slide 36)

**15. Hoeffdingova stabla**

- Stabla odluke koja se grade na inkrementalan način. Grananje na nekoj značajki događa se samo ako je razlika u informacijskoj mjeri veća od Hoeffdingove granice. (Slide 41)

**16. Odabir značajki u tokovima podataka**

- Odabir značajki u tokovima podataka uključuje razmatranje novih značajki čim se pojave i odlučivanje o njihovom uključivanju u skup podataka. Koraci uključuju napučivanje nove značajke, odlučivanje o njenom uključivanju i izmjenu skupa podataka. (Slides 47-48)

# 6. Metode ansambala

**1. Ansambli i njihovo korištenje**

- Ansambl u strojnom učenju je skup od dva ili više modela strojnog učenja s ciljem poboljšanja uspješnosti rezultata u odnosu na pojedinačne modele. Tipično se koriste u kontekstu nadziranog učenja za klasifikacijske ili regresijske probleme. (Slide 4)

**2. Temeljna podjela pristupa ansambala**

- Pristupi ansambala se dijele na:

* Sustavi ansambala više stručnjaka (multi-expert systems)
* Heterogeni jednostavni ansambl
* Stacking
* Bagging
* Sustavi ansambala u više koraka (multi-stage systems), koji uključuju Boosting i kaskadirajući klasifikatori. (Slide 6)

**3. Heterogeni jednostavni ansambl**

- Sastoji se od nekoliko različitih pojedinačnih algoritama strojnog učenja, gdje svaki algoritam može, ali ne mora biti ansambl. Na primjer, kombinacija MLP, k-NN i slučajne šume za detekciju "spamajućih" recenzija proizvoda. Odluke se najčešće donose većinskim glasanjem. (Slide 7)

**4. Stacking**

- Stacking se sastoji od **dvije razine** modela. Niža razina (razina 0) sastoji se od jednog ili više modela različitih algoritama koji uče na jednom dijelu ulaznog skupa, dok viša razina (razina 1) koristi izlaze modela razine 0 kao ulazne podatke za svoj model. (Slides 9-10)

**5. Bagging**

- Bagging (bootstrap aggregation) je ansambl koji se sastoji od većeg broja modela istog tipa. Ulazni podaci za svaki model dobivaju se uzorkovanjem tipa bootstrap. Bagging smanjuje varijancu pojedinačnih modela. (Slides 11-12)

**6. Boosting**

- Boosting gradi ansambl modela iterativno, nastojeći poboljšati decizijsku granicu među klasama iz koraka u korak. U svakoj iteraciji uči se novi model na skupu za učenje koji se mijenja prema rezultatima prethodnog modela. (Slide 13)

**7. Slučajna šuma**

- Slučajna šuma je algoritam ansambla slučajnih stabala odluke koji koristi **kombinaciju izvora slučajnosti, veliki broj stabala i većinsko glasanje kako bi postigao izvrsne rezultate**. (Slide 16)

**8. Slučajna šuma – značajke**

- Minimizira pogrešku nastalu zbog varijance i pristranosti. Pristranost se minimizira građenjem stabla do kraja, a varijanca uz pomoć bootstrap uzorkovanja, slučajnog odabira atributa i velikog broja stabala. (Slide 17)

**9. Slučajna šuma – početne postavke**

- Ansambl se sastoji od K slučajnih stabala odluke. Postavlja se broj značajki koje će se razmatrati prilikom podjele svakog čvora u stablu (m), najmanji dozvoljeni broj primjeraka u čvoru koji nije list (nmin), i broj stabala (K). (Slide 19)

**10. Slučajna šuma – algoritam**

- Provodi se uzorkovanje tipa bootstrap za svako stablo. Za svaki čvor u stablu, odabiru se m značajki kandidata, generiraju m podjele, te se uzima najinformativnija podjela. Stablo se gradi u dubinu dok čvor ne postane list. (Slide 20)

**11. Iznimno slučajna stabla**

- Iznimno slučajna stabla uvode dodatne izvore slučajnosti. Mogu se podijeliti na iznimno slučajna stabla (Extra-Trees) i potpuno slučajna stabla. Iznimno slučajna stabla slučajno odabiru podjelu značajke na čvoru, ali biraju najbolju podjelu. (Slide 23)

**12. Rotacijska šuma (samo prvi od tri slajda)**

- Ansambl klasifikacijskih stabala odluke koji se temelji na bootstrap uzorkovanju, analizi glavnih komponenti (PCA) i ansamblu stabala odluke C4.5. Postiže vrhunske rezultate na skupovima podataka s numeričkim vrijednostima, ali je značajno sporiji od slučajne šume. (Slide 25)

**13. AdaBoost i MultiBoost**

- AdaBoost i MultiBoost algoritmi su ansambala temeljeni na iterativnom postupku izgradnje modela korištenjem postupka boostinga nad inicijalno slabim modelom. AdaBoost koristi panjeve odluke, dok MultiBoost koristi stabla odluke C4.5 i utežani bagging. (Slides 28-30)

**14. XGBoost**

- eXtreme Gradient Boosting (XGBoost) je napredan algoritam temeljen na boostingu i optimizaciji gradijentnim spustom. Koristi stabla odluke CART koja su asimetrična i podrezana na kraću dubinu. Algoritam dodaje nova stabla koja smanjuju pogrešku iz prethodnih iteracija. (Slides 31-34)

**15. Permutacijska važnost**

- Permutacijska važnost značajke definira se kao smanjenje rezultantne mjere modela kada se vrijednosti te značajke nasumično promiješaju. Važnost značajke je to veća što permutiranje njezinih vrijednosti više smanjuje rezultantnu mjeru modela. (Slides 38-39)

**16. SHAP**

- SHAP (SHapley Additive exPlanations) je objašnjivi postupak lokalnog tipa koji se koristi za objašnjavanje predikcije primjerka bilo kojeg modela strojnog učenja. Vizualizira doprinos pojedinih značajki predikciji primjerka. (Slides 40-41)

# 7. Strojno učenje s jasnim tumačenjem, indukcija pravila i optimalna stabla odluke

**1. Strojno učenje s jasnim tumačenjem**

- Uključuje modele strojnog učenja čije tumačenje je jasno razumljivo čovjeku (engl. white box models). Ne miješati s objašnjavanjem black box modela strojnog i dubokog učenja u okviru tzv. objašnjive umjetne inteligencije (engl. eXplainable AI, kraće: XAI). (Slide 4)

**2. Metode s inherentno jasnim tumačenjem**

- Linearna regresija, logički modeli (if-then, and, or...), induktivna pravila, stabla odluke, rasuđivanje zasnovano na slučajevima (najbliži susjedi, prototipovi), generalizirani aditivni modeli (GAM). (Slide 6)

**3. Indukcija pravila (induktivna pravila)**

- Zadatak: naučiti korisna if-then pravila automatski na temelju skupa podataka. Najčešća primjena je na klasifikacijske probleme, ali može i na regresijske. Prednost je vrlo jasno tumačenje modela, dok je nedostatak ponekad slabija točnost. (Slide 9)

**4. Učenje pravila – konjunkcijsko pravilo**

- Tijelo pravila sadržava konjunkciju uvjeta, a glava pravila predviđanje. Za binarnu klasifikaciju: pozitivnu klasifikaciju (+) ako primjerak pripada određenom konceptu, a negativnu (-) ako mu ne pripada. (Slide 12)

**5. Konjunkcijsko pravilo – terminologija**

- Pokrivanje (engl. rule coverage): Pravilo pokriva primjerak ako primjerak ispunjava sve uvjete pravila.

- Predviđanje pravila (engl. rule prediction): Ako pravilo pokriva primjerak, glava pravila je predviđena za taj primjerak.

- Konzistentnost skupa pravila (engl. rule set consistency): Skup pravila je konzistentan ako sva pravila u skupu pokrivaju samo primjerke jedne određene klase.

- Potpunost skupa pravila (engl. rule set completion): Skup pravila je potpun ako su svi primjerci pokriveni nekim pravilom. (Slide 13)

**6. Učenje pravila po principu „razdvoji pa vladaj”**

- Engl. separate-and-conquer rule learning, covering rule learning. Pravila se grade postepeno i tako da razdvoje primjerke od primjeraka. (Slide 17)

**7. Relaksacija potpunosti i konzistentnosti**

- Izgradnja skupa pravila koji je kompletan i konzistentan pokazano dovodi do prenaučenosti. Potrebno je balansirati između potpunosti pokrivanja pozitivnih primjeraka i smanjenja konzistentnosti pravila. (Slide 18)

**8. Heuristike pokrivanja (samo prvi slajd)**

- Dodavanje pravila treba povećati broj pokrivenih pozitivnih primjeraka (povećati stupanj potpunosti) i smanjiti broj pokrivenih negativnih primjeraka (ne smanjivati konzistentnost). (Slide 21)

**9. Izbjegavanje prenaučenosti**

- Većina algoritama fokusira se na kraća pravila koja pokrivaju puno pozitivnih primjeraka (i neke negativne) umjesto duljih pravila koja pokrivaju samo mali broj pozitivnih primjeraka. Metode uključuju podrezivanje pravila (pre-pruning i post-pruning). (Slide 25)

**10. Višeklasna klasifikacija**

- Problem: većina algoritama prilagođena je samo za binarne klasifikacijske probleme. Rješenje: binarizacija klasa – jedan višeklasni problem pretvara se u skup binarnih problema, najčešće koristeći pristup "jedan protiv svih". (Slide 28)

**11. Preklapanje pravila i problem nepostojanja pravila**

- Preklapanje pravila: Ako više pravila pokrivaju istog primjerka, problem je ako su predviđanja kontradiktorna. Rješenja uključuju liste odlučivanja i skupove odlučivanja. Problem nepostojanja pravila: Ako primjerak nije pokriven ni jednim pravilom, koristi se defaultno pravilo. (Slide 29)

**12. OneRule – algoritam**

- Algoritam uči pravila na temelju jedne, najinformativnije značajke. Kontinuirane značajke se diskretiziraju, stvara se unakrsna tablica između prediktivne značajke i ciljne varijable, gradi se pravilo za svaku vrijednost prediktivne značajke, računa se ukupna pogreška pravila, i izabire se značajka s najmanjom ukupnom pogreškom kao rezultantna značajka. (Slide 32)

**13. RIPPER – algoritam (samo prvi slajd)**

- RIPPER (Repeated Incremental Pruning to Produce Error Reduction) gradi pravila za svaku pojedinu klasu ciljne značajke u četiri glavne faze: izgradnja pravila, optimizacija pravila, pometanje preostalih primjeraka, čišćenje loših pravila. (Slide 37)

**14. Stabla odluke**

- Prediktivni modeli u obliku stabla, svaki čvor sastoji se od ispitivanja uvjeta, a list daje predviđanje modela. Potpuna optimizacija stabla odluke je NP-težak problem. Postoje dvije faze razvoja izgradnje stabala odluke: heuristička i optimizacijska. (Slide 48)

**15. OCT (OCT-H) (samo prvi slajd)**

- Optimal Classification Trees (OCT) su predvodnik optimizacijskih metoda za izgradnju stabala odluka iz 2017. godine. Koriste mixed-integer optimizaciju (MIO) i multivarijatno grananje za povećanje točnosti i smanjenje veličine stabla. (Slide 53)

# 8. 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 (itemsets), podstrukture skupa podataka (grafovi), podsekvence (vremenski nizovi podataka). Na temelju itemsetova najčešće se grade asocijativna pravila. (Slide 3)

**2. Itemset**

- Itemset je skup koji se sastoji od jednog ili više artikala. Podrazumijeva se da svaki artikl naveden u itemsetu ima vrijednost 1. Veličinu itemseta označavamo s k-itemset. Primjeri itemsetova uključuju {Kava}, {Čokolada, Mlijeko}, {Čaj, Kava, Čokolada}. (Slide 7)

**3. Pronalazak čestih obrazaca – terminologija**

- Apsolutna potpora itemseta A (ili samo: potpora, engl. support) = broj transakcija koje sadrže itemset A. Itemset A je čest u skupu podataka ako je njegova potpora veća ili jednaka minimalnom pragu potpore (min\_sup). Relativna potpora je omjer broja transakcija koje sadrže A i ukupnog broja transakcija. (Slide 8)

**4. Asocijativna pravila – terminologija**

- 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 definirana pomoću mjere pouzdanosti (engl. confidence). (Slide 9)

**5. Asocijativna pravila – cilj**

- Cilj je za dani skup transakcija T otkriti sva asocijativna pravila (A → B) čiji itemsetovi imaju apsolutnu potporu veću ili jednaku minimalnom pragu potpore (min\_sup) i koji imaju pouzdanost veću ili jednaku minimalnom pragu pouzdanosti (min\_conf). (Slide 10)

**6. Algoritam Apriori**

- Temeljni iscrpni algoritam za učenje asocijativnih pravila na temelju pronađenih čestih itemsetova. Prednost: garantirano pronalazi sve itemsetove koji imaju potporu veću od minimalnog praga potpore i sva pravila koja imaju pouzdanost veću od minimalnog praga pouzdanosti. Nedostatak: dosta je spor jer ne optimira pretraživanje skupa mogućih itemsetova. (Slide 13)

**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. (Slide 14)

**8. Algoritam Apriori – primjer**

- Primjer rada algoritma Apriori prikazuje kako se prolazi kroz razine pretrage itemsetova i kako se određuje koji itemsetovi ostaju česti na temelju potpore. (Slides 17-19)

**9. Algoritam PCY (svi slajdovi)**

- Algoritam PCY (Park et al., 1997) je algoritam pronalaska čestih itemsetova s manjom vremenskom složenosti od Apriorija. Koristi funkciju **raspršenog adresiranja** (hashing function) prilikom prvog prolaska kroz skup kako bi se smanjio broj parova artikala koje treba razmatrati u kasnijim koracima. Algoritam koristi tablicu raspršenog adresiranja (hash-table) i bitmapu za eliminaciju kandidata koji sigurno nisu česti. (Slides 20-24)

**10. Algoritam FP-growth (samo prvi slajd)**

- FP-Growth (Frequent Pattern Growth) je algoritam koji koristi stablastu strukturu podataka (FP-Tree) za sažeti prikaz ulaznih podataka i pronalaženje čestih itemsetova. Algoritam je značajno brži od Apriorija za pronalazak čestih itemsetova jer izbjegava generiranje kandidata. (Slide 25)

**11. Algoritam EFIM**

- EFIM (EFficient high-utility Itemset Mining) je algoritam za otkrivanje visokokorisnih itemsetova (HUIM). Za razliku od tradicionalnih algoritama, EFIM se fokusira na itemsetove koji imaju visoku korisnost u smislu profita. Koristi optimizacije za pretraživanje prostora itemsetova koje omogućuju bolje rezultate od konkurentskih algoritama. (Slide 34)

**12. Algoritam EFIM – definicija problema HUIM**

- HUIM (High-Utility Itemset Mining) baze sadržavaju uz svaki kupljeni artikl informaciju o težini, koja govori koliki je jedinični profit za dotični artikl. Zadatak HUIM-a je pronaći sve itemsetove koji imaju korisnost veću od minimalno zadane. (Slide 35)

**13. Algoritam EFIM – primjer**

- Primjer korisnosti artikala u transakcijama prikazuje kako se računa korisnost itemsetova u bazi podataka. (Slide 36)

**14. Sustavi preporučivanja**

- Sustavi za preporučivanje sadržaja razvijeni u svrhu povećanja prometa. Glavne vrste sustava preporučivanja uključuju suradničko filtriranje (collaborative filtering), filtriranje zasnovano na sadržaju (content-based filtering), sustavi preporučivanja zasnovani na znanju (knowledge-based recommender system), i sustavi preporučivanja zasnovani na dubokom učenju (deep learning-based recommender system). Hibridni sustavi kombiniraju više pristupa. (Slides 45-46)

# 9. Dubinska analiza vremenskih nizova

**1. Dubinska analiza vremenskih nizova**

- Dubinska analiza vremenskih nizova (engl. time series data mining, time series analysis) je područje dubinske analize podataka koje analizira vremenske nizove (slijedove, sekvence) podataka s ciljem pronalaska zanimljivih obrazaca. (Slide 4)

**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. (Slide 5)

**3. Glavni zadaci analize vremenskih nizova**

- Glavni zadaci uključuju upit prema sadržaju, detekciju anomalija, otkrivanje motiva, predviđanje vrijednosti, grupiranje, klasifikaciju i segmentaciju ili reprezentaciju. (Slide 6)

**4. Terminologija**

- Vremenski niz T je uređeni slijed od n realnih brojeva. Niz može biti univarijatan ili multivarijatan, konačan ili polubeskonačan, stacionaran ili nestacionaran, periodički, skoro periodički, kvaziperiodički, kaotičan ili slučajan. (Slides 9-16)

**5. Predobrada vremenskog niza – uklanjanje šuma i skaliranje**

- Uklanjanje šuma uključuje korištenje digitalnih filtera poput pojasno propusnog filtera, filtra pomičnog prosjeka i eksponencijalnog zaglađivanja. Skaliranje se može provesti min-max normalizacijom ili standardizacijom z-skaliranjem. (Slide 19)

**6. Predobrada vremenskog niza – ponovno uzorkovanje i interpolacija**

- Ponovno uzorkovanje (resampling) uključuje promjenu frekvencije uzorkovanja, dok interpolacija podrazumijeva pronalaženje kontinuirane funkcije na temelju diskretnih točaka vremenskog niza kako bi se nadomjestile nepoznate vrijednosti. (Slide 21)

**7. Predobrada vremenskog niza – podjela u prozore**

- Podjela niza u prozore (windowing) dijeli niz u manje podnizove određene širine, što smanjuje računske zahtjeve i povećava preciznost izvedbe zadataka kao što su grupiranje, klasifikacija i predviđanje. (Slide 23)

**8. Reprezentacijske metode**

- Izvlačenje najvažnijih informacija iz vremenskog niza. Reprezentacijske metode se dijele na neadaptivne, adaptivne i metode zasnovane na modelu. (Slide 25)

**9. Neadaptivne reprezentacijske metode (prvi slajd)**

- Parametri transformacije ostaju isti za svaki niz, neovisno o njegovoj prirodi. Primjeri uključuju DFT (diskretna Fourierova transformacija), DWT (diskretna transformacija valićima), PAA (aproksimacija agregacijom po dijelovima) i HHT (Hilbert-Huangova transformacija). (Slide 26)

**10. Adaptivne reprezentacijske metode**

- Definiraju parametre transformacije ovisno o konkretnim podacima. Većina neadaptivnih metoda može se prilagoditi da postanu adaptivne, kao npr. SAX (Symbolic Aggregate approXimation) i shapeleti. (Slides 28-30)

**11. Reprezentacijske metode zasnovane na modelu**

- Pretpostavljaju da je opaženi niz rezultat nekog modela. Metode uključuju modeliranje ekstrakcijom značajki, modeliranje učenjem značajki i ARMA modele (autoregressive moving average). (Slide 31)

**12. Mjere sličnosti**

- Mjera sličnosti između vremenskih nizova omogućava provođenje različitih zadataka nad vremenskim nizovima. Primjeri uključuju DTW (Dynamic Time Warping) i CDM (Compression-Based Distance Measure). (Slides 32-33)

**13. Predikcijski algoritam: ARIMA (prvi slajd)**

- ARIMA (Auto-Regressive Integrated Moving Average) je statistički parametarski model namijenjen za analizu i predviđanje vremenskih nizova. (Slide 36)

**14. Predikcijski algoritam: PROPHET (prvi slajd)**

- PROPHET je generalizirani aditivni model za predikciju vrijednosti vremenskih nizova s tri glavne komponente: trendom, sezonalnošću i praznicima. (Slide 41)

**15. Klasifikacijski algoritam: CIF**

- CIF (Canonical Interval Forest) je napredni klasifikacijski ansambl stabala odluke učenih na određenim značajkama iz slučajno odabranih intervala vremenskih nizova. (Slide 45)

**16. CIF – opis algoritma**

- CIF koristi 25 značajki (22 od catch22, 3 sumarne statistike) i gradi šumu od r stabala (default: 500), poduzorkujući značajke i intervale za svaki čvor. (Slides 46-47)

**17. Klasifikacijski algoritam: HIVE-COTE v2 (prvi slajd)**

- HIVE-COTE v2 (The HIerarchical VotE Collective Of Transformation-based Ensembles) je state-of-the-art klasifikator za općenite vremenske nizove, uključujući meta-ansambl od 4 komponente. (Slide 49)

**18. Modeliranje niza ekstrakcijom značajki (prvi slajd)**

- Ekstrakcija različitih značajki vremenskog niza koje će ga reprezentirati te njihovo korištenje za učenje klasifikatora ili regresora. (Slide 54)

# 10. Duboko učenje u DAP-u

1. \*\*Duboko učenje\*\*

- Duboko učenje (engl. Deep learning) je područje strojnog učenja koje koristi različite arhitekture dubokih neuronskih mreža s odgovarajućim postupcima učenja. Razvija se od 2011., s najvećim napretkom od 2016. do danas. Nije zamjena za strojno učenje, koristi se zajedno s tradicionalnim algoritmima strojnog učenja. (Slide 4)

2. \*\*Umjetne neuronske mreže\*\*

- Umjetne neuronske mreže su matematički model bioloških neuronskih mreža, sastavljene od umjetnih neurona povezanih u slojeve: jedan ulazni sloj, jedan ili više skrivenih slojeva i jedan izlazni sloj. (Slide 5)

3. \*\*Aktivacijske funkcije\*\*

- Aktivacijske funkcije razvijaju se od 1990-tih do danas i najčešće rade nelinearnu transformaciju vrijednosti modela težina u nekom neuronu. Primjeri uključuju sigmoidalnu, tanh, ReLU, eksponencijalnu i linearnu funkciju. (Slide 7)

4. \*\*Algoritmi učenja\*\*

- Najšire korišteni algoritam za učenje neuronske mreže je gradijentni spust (engl. gradient descent), koji revidira težine veza između neurona. Popularna proširenja uključuju optimizatore poput Adam, RMSProp, itd. (Slides 8-10)

5. \*\*Optimizacija hiperparametara\*\*

- Optimizacija hiperparametara uključuje metode kao što su slučajna pretraga, ručni izbor, pretraga po rešetki (grid search) i Bayesova optimizacija. Hiperparametri se postavljaju prije učenja mreže. (Slides 11-14)

6. \*\*Funkcije gubitka\*\*

- Funkcija gubitka ili sloj gubitka (engl. loss function) specificira kako učenje penalizira razliku između predviđenog i stvarnog izlaza. Primjeri uključuju kategorijsku unakrsnu entropiju, srednju kvadratnu pogrešku i fokalni gubitak. (Slides 15-16)

7. \*\*Zašto duboke neuronske mreže uspijevaju?\*\*

- Tehnološki razlozi uključuju povećanje količine podataka i sklopovskih performansi. Arhitekturni razlozi uključuju specijalizirane arhitekture, nove aktivacijske funkcije i optimizacije algoritama učenja. (Slides 17-18)

8. \*\*Problemi dubokih neuronskih mreža (prva tri slajda)\*\*

- Problemi uključuju slabu objašnjivost modela, složenost postupka učenja, problem nestajućeg i eksplodirajućeg gradijenta. (Slides 19-21)

9. \*\*Konvolucijska mreža (svi slajdovi)\*\*

- Konvolucijska mreža koristi konvolucijske slojeve za obradu podataka. Konvolucija je linearna operacija koja množi skup težina s receptivnim poljem i primjenjuje aktivacijsku funkciju. (Slides 25-29)

10. \*\*Konvolucija\*\*

- Konvolucija je linearna operacija koja množi skup težina u obliku matrice s receptivnim poljem u vidu skalarnog produkta te rezultira jednim brojem (skalarom). (Slide 26)

11. \*\*Konvolucijska mreža – hiperparametri\*\*

- Hiperparametri uključuju veličinu jezgre, korak, veličinu dopune, dilataciju jezgre, broj različitih filtara u svakom sloju i veličinu sloja sažimanja. (Slide 30)

12. \*\*Rekurentne neuronske mreže\*\*

- Rekurentne mreže (RNN) su dizajnirane za rukovanje slijednim podacima, koriste rekurentne blokove koji primaju vrijednost i izlaz prethodnog bloka te daju novi izlaz. (Slide 31)

13. \*\*Rekurentne neuronske mreže – LSTM\*\*

- LSTM (Long Short-Term Memory) je vrsta rekurentnog bloka koji koristi vrata zaboravljanja, vrata novog ulaza i izlazna vrata za učenje dugotrajnih ovisnosti. (Slide 34)

14. \*\*Autoenkoderi\*\*

- Autoenkoderi su neuronske mreže koje se koriste za nenadzirano učenje, sastoje se od kodirajućeg i dekodirajućeg dijela. (Slide 36)

15. \*\*Autoenkoderi – naslagani autoenkoderi\*\*

- Naslagani autoenkoderi koriste se za skupove podataka s kompleksnim odnosima među značajkama, svaki autoenkoder u nizu dobiva predikciju od prethodnog kako bi bolje naučio značajke. (Slide 40)

16. \*\*Autoenkoderi – varijacijski autoenkoderi\*\*

- Varijacijski autoenkoderi (VAE) preslikavaju ulaze u latentni prostor opisan srednjom vrijednosti i standardnom devijacijom. Uče se pomoću regularizacije KL-divergencijom i funkcijom gubitka. (Slide 41)

17. \*\*Transformeri\*\*

- Transformeri su arhitektura najpoznatija po primjeni u velikim jezičnim modelima. Koriste mehanizam pažnje (attention) za povezivanje dijelova sekvenci i učinkovitu reprezentaciju pojmova. (Slide 42)

18. \*\*Transformeri – kratki opis izvorne arhitekture\*\*

- Koderski dio transformera sastoji se od N identičnih slojeva koji obrađuju ulazne podatke, dok dekoderski dio koristi informacije iz kodera za generiranje izlaza. (Slide 44)

19. \*\*Učenje transferom i fino podešavanje\*\*

- Učenje transferom koristi prednaučeni model na sličnom problemu, dok fino podešavanje prilagođava prednaučeni model za specifičan zadatak. (Slide 50)

# 11. Arhitekture dubokog učenja u područjima primjene

1. \*\*ROCKET (prva dva slajda)\*\*

- ROCKET (RandOm Convolutional KErnel Transform) je state-of-the-art algoritam za klasifikaciju vremenskih nizova. Koristi veliki broj slučajnih konvolucijskih jezgri za transformaciju vremenskog niza, pri čemu je jedini hiperparametar broj kernela. Transformirane značajke se koriste za učenje jednostavnog klasifikatora poput ridge regresije ili logističke regresije. (Slides 4-5)

2. \*\*Hydra+MultiROCKET (prva dva slajda)\*\*

- Hydra kombinira slučajno postavljene konvolucijske kernele podijeljene u grupe za transformaciju vremenskog niza, dok MultiROCKET koristi specifične duljine kernela i generira veći broj dilatacija i biasa za transformaciju. Hydra+MultiROCKET je kombinirani ansambl koji koristi oba pristupa za veću preciznost. (Slides 7-8)

3. \*\*ResNet (prvi slajd)\*\*

- ResNet (Deep Residual Learning for Image Recognition) je arhitektura dubokih konvolucijskih neuronskih mreža s mnogo slojeva, zasnovana na rezidualnim blokovima koji koriste skip veze za olakšavanje učenja i rješavanje problema nestajućeg gradijenta. (Slide 17)

4. \*\*YOLO (prvi slajd)\*\*

- YOLO (You Only Look Once) je brza i točna neuronska mreža za detekciju objekata u stvarnom vremenu. Dijeli sliku na mrežu ćelija i predviđa postojanje objekta i koordinate bounding boxa za svaku ćeliju. (Slide 22)

5. \*\*Transformerske arhitekture za klasifikaciju slika\*\*

- Vision Transformers (ViT) tretiraju sliku kao slijed fragmenata fiksne veličine i koriste transformersku arhitekturu za obradu tih fragmenata za klasifikaciju. Uključuju mehanizam samopozornosti za učenje odnosa između fragmenata. (Slide 25)

6. \*\*Objašnjavanje dubokih modela\*\*

- Objašnjavanje dubokih modela koristi post-hoc metode za objašnjenje rada modela nakon što je izgrađen. Metode uključuju globalna i lokalna objašnjenja, s fokusom na vizualizaciju aktivacija i atribucije unutar mreže. (Slide 27)

7. \*\*Klasifikacija metoda za vizualizaciju (objašnjenje) modela\*\*

- Metode za vizualizaciju modela uključuju vizualizaciju aktivacije slojeva, atribuciju zasnovanu na gradijentima (npr. mape istaknutosti, Grad-CAM), atribuciju zasnovanu na perturbaciji (ablacija značajki, uzorkovanje Shapleyjevih vrijednosti) i vizualizaciju mehanizma pažnje za transformere. (Slide 28)

8. \*\*BERT\*\*

- BERT (Bidirectional Encoder Representations from Transformers) je prednaučeni jezični model zasnovan na transformerskoj arhitekturi. Koristi se za zadatke poput klasifikacije teksta, analize sentimenta i prepoznavanja imenovanih entiteta. (Slide 35)

9. \*\*RoBERTa\*\*

- RoBERTa (A Robustly optimized BERT Approach) je poboljšana verzija BERT-a koja koristi veći skup podataka za učenje, veći batch size, dulje sekvence podataka i dinamičnu promjenu parametra maskiranja tijekom učenja. (Slide 37)

10. \*\*GPT-4\*\*

- GPT-4 (Generative Pre-trained Transformer v4) je veliki jezični model tvrtke OpenAI, multimodalan je i na ulazu prima tekst i/ili slike te odgovara tekstom. Ima preko 1 bilijun parametara i koristi se za različite jezične zadatke. (Slide 39)

11. \*\*Uvod u generiranje slika iz teksta\*\*

- Generiranje slika iz teksta uključuje preslikavanje tekstnog opisa u vektorsku reprezentaciju te izgradnju generativnog modela za transformaciju te reprezentacije u sliku. Model se vrednuje vizualnom procjenom kvalitete dobivenih slika ili objektivnim kriterijima poput indeksa strukturne sličnosti (SSIM). (Slide 43)

12. \*\*Stabilna difuzija (prva dva slajda)\*\*

- Stabilna difuzija koristi generativni model, često varijacijski autoenkoder, za modeliranje šuma ili slučajnosti u latentnom prostoru. Proces uključuje transformaciju inicijalne razdiobe šuma u niz međurazdioba sve veće složenosti, čime se generiraju uzorci s različitim značajkama i stilovima. (Slides 44-45)