



Napredne arhitekture informacionih sistema

Vektorske baze podataka

Predmetni nastavnik:
dr Marko Vještica

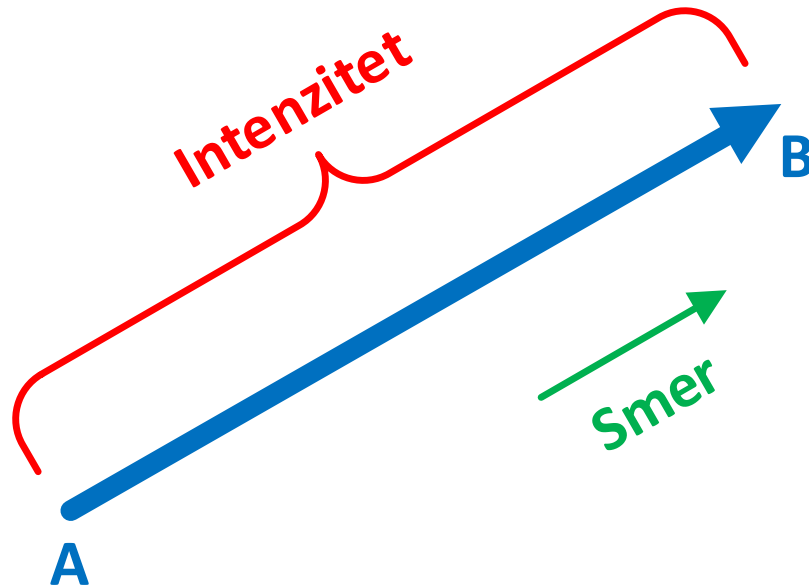


Sadržaj

- Vektori
- Ugrađeni vektori
- Modeli ugrađivanja
- Metrike rastojanja dva vektora
- Vektorska baza podataka
- Pretrage kNN i ANN
- Hibridna pretraga
- Primena vektorske baze podataka u arhitekturi RAG
- Poređenje relacije i vektorske baze podataka
- Literatura

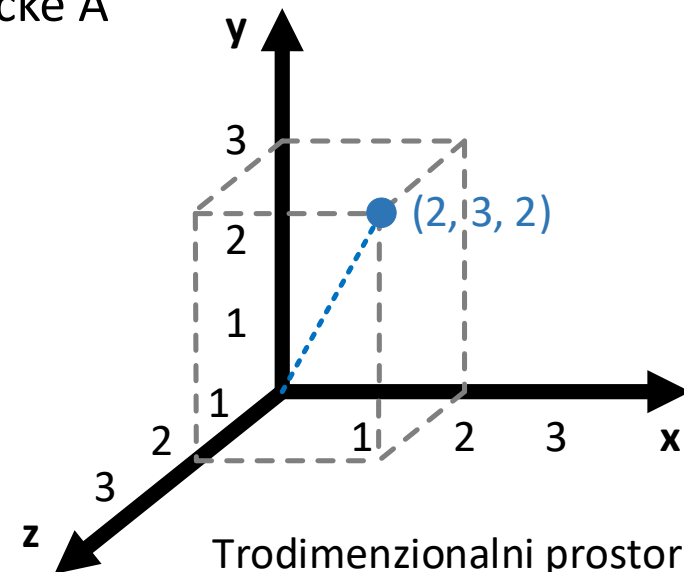
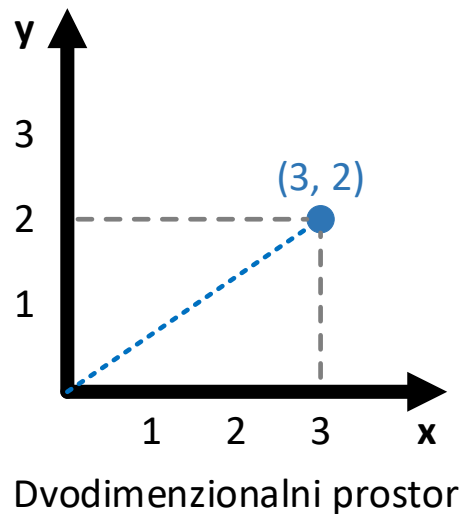
Vektori

- **Vektor** predstavlja matematički objekat koji ima **intenzitet i smer**
 - Može biti predstavljen kao **orijentisana duž**, odnosno kao **usmerena linija**, gde dužina linije predstavlja **intenzitet** vektora, a strelica linije predstavlja **smer** vektora
 - Npr. **intenzitet** vektora \vec{AB} predstavlja merni broj duži $|\vec{AB}|$, **pravac** vektora \vec{AB} je određen tačkama A i B, a **smer** vektora \vec{AB} je od A prema B

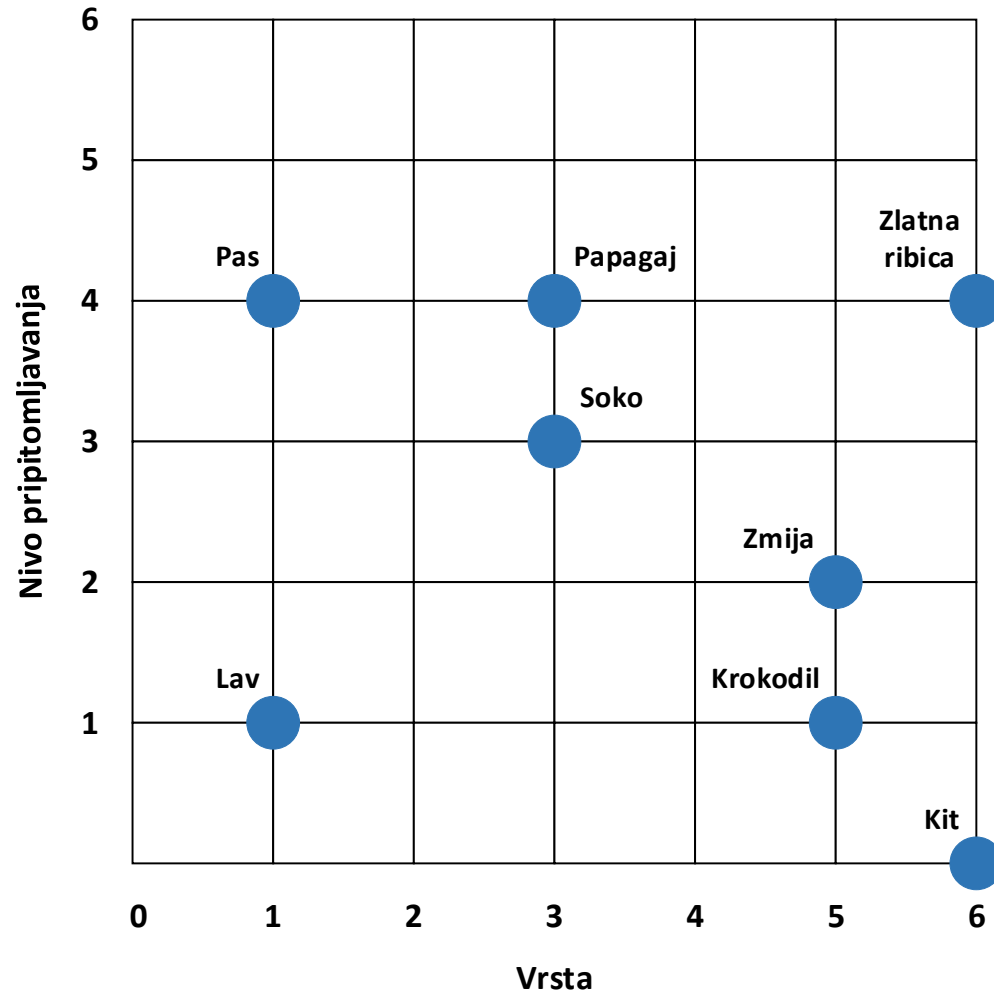


Vektorski prostori

- **Vektorski prostor** predstavlja **n-dimenzionalni prostor** koji sadrži skup vektora
- **Vektor** predstavlja **niz vrednosti (n-torku)**, gde svaka vrednost odgovara jednoj dimenziji u n-dimenzionalnom vektorskom prostoru
 - **Ne mogu biti iskazani** jednim brojem (**skalarom**)
 - Vektor \vec{OA} predstavlja **vektor položaja tačke** A, gde je O koordinatni početak, dok (x_1, x_2, \dots, x_n) predstavljaju koordinate tačke A



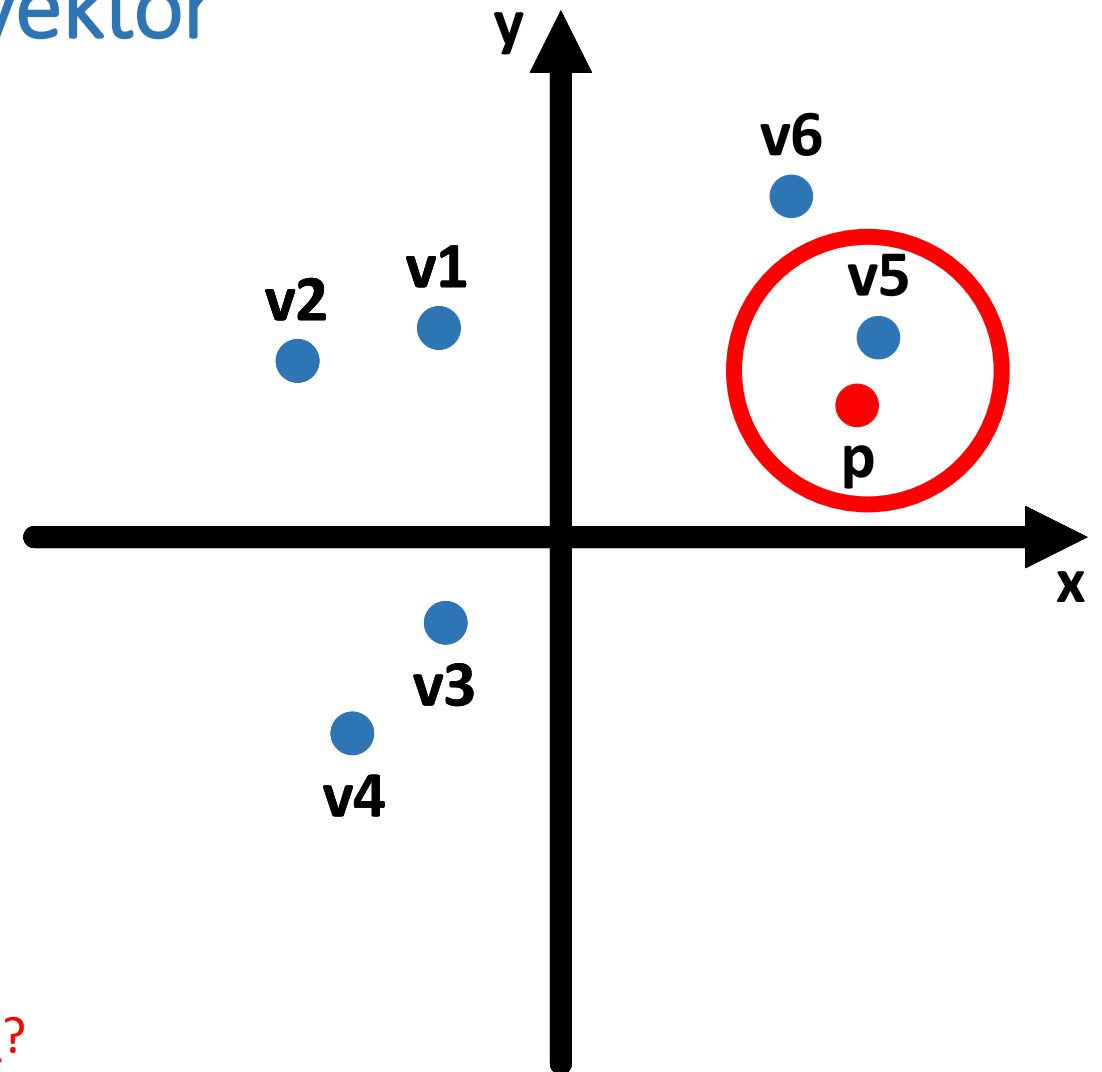
Primer vektorskog prostora



Životinja	Vektor
Lav	(1, 1)
Pas	(1, 4)
Soko	(3, 3)
Papagaj	(3, 4)
Krokodil	(5, 1)
Zmija	(5, 2)
Kit	(6, 0)
Zlatna ribica	(6, 4)

Primer transformacije teksta u vektor

- (v1) Mikroservisi su komponente malog obima koje razvijaju timovi sa malim brojem članova
- (v2) API Gateway dinamički preusmerava zahteve korisnika odgovarajućim mikroservisima
- (v3) Virtualna mašina predstavlja digitalnu kopiju računarskog hardvera na kojem je pokrenut operativni sistem gost
- (v4) Kontejner predstavlja izolovan proces koji sadrži aplikaciju koju je potrebno pokrenuti u okviru operativnog sistema
- (v5) Osnovni koncepti koje koristi grafska baza podataka su čvorovi i grane
- (v6) Grafske baze podataka primenjuju algoritme obilaska grafa prilikom izvršavanja upita
- **Pitanje: Koje koncepte koristi grafska baza podataka?**



Primena vektora

- **Primena** vektora u različitim domenima:
 - Matematika
 - Fizika (npr. jačina magnetnog polja)
 - Video igre (npr. udaljenost objekata)
 - Saobraćaj (npr. kretanje aviona i vetra prilikom sletanja)
 - Mašinsko učenje (npr. obrada slika)
 - Informacioni sistemi (npr. skladištenje teksta, slika)
 - ...

Sadržaj

- Vektori
- Ugrađeni vektori
- Modeli ugrađivanja
- Metrike rastojanja dva vektora
- Vektorska baza podataka
- Pretrage kNN i ANN
- Hibridna pretraga
- Primena vektorske baze podataka u arhitekturi RAG
- Poređenje relacije i vektorske baze podataka
- Literatura

Kvalitativni podaci

- **Kvalitativni (kategorijalni) podaci** predstavljaju **nenumeričke vrednosti** određenih karakteristika ili kategorija
 - Za razliku od **kvantitativnih podataka** koji predstavljaju **numeričke vrednosti**
- **Kvalitativna promenljiva** predstavlja **skup kategorija**, obuhvatajući ograničen broj različitih vrednosti
 - Koristi se često i termin **enumeracija**
- Kvalitativne promenljive dele se na **nominalne i ordinalne**

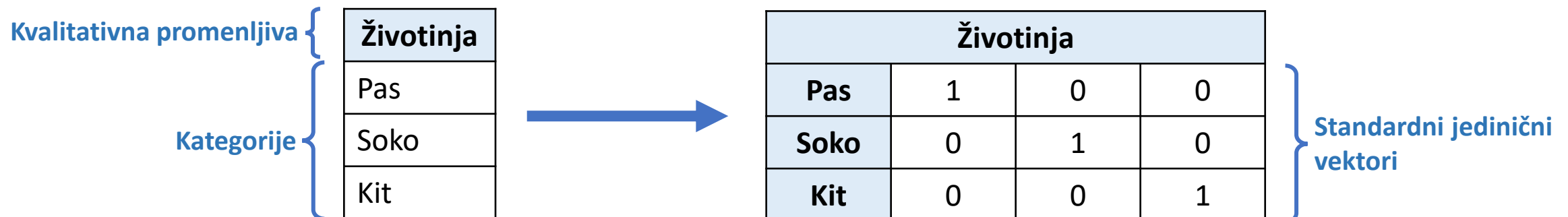
	Nominalne	Ordinalne
Značenje	Ne postoji uređenje među vrednostima (nema poretka)	Postoji uređenje među vrednostima (rangiranje)
Primer	Boja sa kategorijama: crvena, zelena, plava...	Stepen akademskih studija sa kategorijama: osnovne akademske studije, master akademske studije i doktorske akademske studije

Kodiranje karakteristika

- **Kvalitativni podaci** često se koriste za potrebe **mašinskog učenja**
 - Međutim, većina modela mašinskog učenja ili dubokog učenja **ne mogu da obrade** kvalitativne podatke
 - Potrebne su **numeričke vrednosti**
- **Kodiranje karakteristika** (engl. *Feature Encoding*) predstavlja **proces transformisanja kategorijalnih vrednosti** relevantnih karakteristika u **numeričke vrednosti** na način da ih modeli mašinskog učenja mogu obraditi
- Postoje različite **metode kodiranja karakteristika**, poput:
 - **Kodiranja standardnim jediničnim vektorima**
 - **Kodiranja ugrađenim vektorima**

Kodiranje standardnim jediničnim vektorima

- **Kodiranje standardnim jediničnim vektorima** (engl. *One-Hot Encoding*) je metod za predstavljanje kvalitativnih podataka kao **jedinstvenih binarnih vektora**
 - Svaka kategorija preslikana je u vektor koji sadrži **vrednost 1 na poziciji koja odgovara indeksu kategorije**, a **vrednost 0 na svim ostalim pozicijama**
 - Svaka kategorija preslikana je u **jedinstveni jedinični vektor**
 - Ne postoje identični jedinični vektori za različite kategorije



Standardni jedinični vektori

- **Standardni jedinični vektor** predstavlja vektor sa **binarnim vrednostima**
 - Predstavlja **redak** (engl. *Sparse*) vektor
 - Sadrži vrednost 1 na poziciji dimenzije koja reprezentuje odgovarajuću kategoriju, dok su vrednosti na svim ostalim pozicijama 0
 - **Intenzitet** takvih vektora je uvek **jednak 1**
 - Predstavlja **pogodan format** za obuku modela mašinskog učenja
 - **Ne ističe ni jednu kategoriju** u odnosu na ostale
- Pogodno ih je koristiti za **nominalne promenljive** (kada vrednosti kategorija nisu uređene) i kada obuhvataju relativno **mali broj kategorijalnih vrednosti**

Standardni jedinični vektori

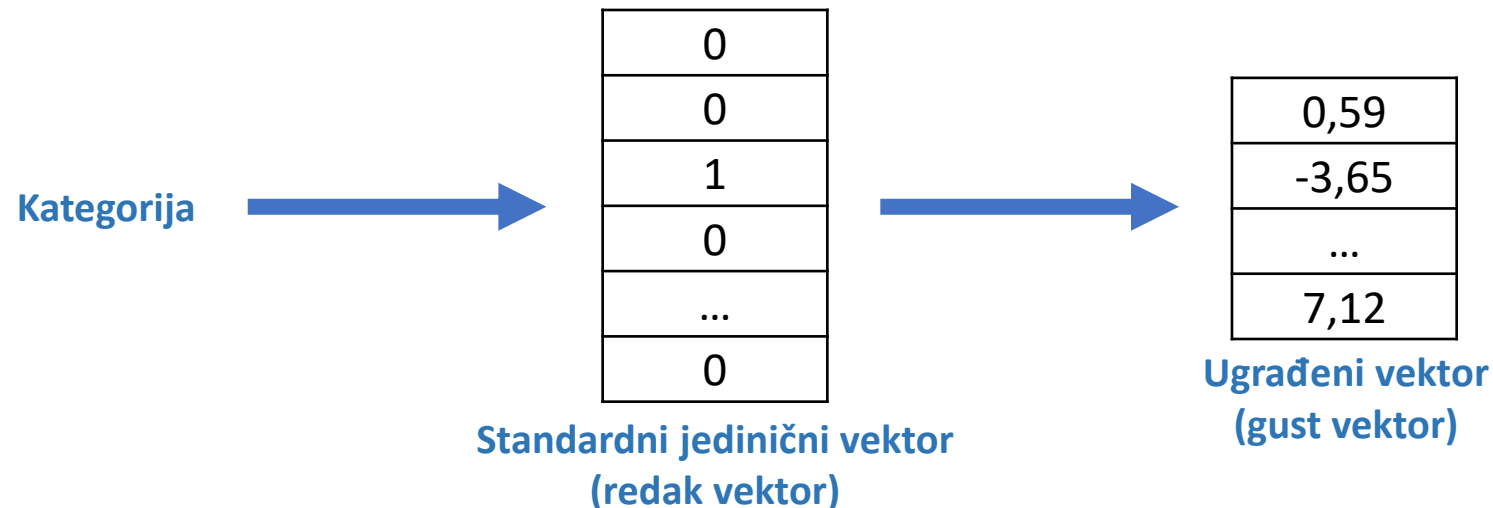
- **Nedostaci** kodiranja standardnim jediničnim vektorima:
 - U slučaju **velikog broja različitih kategorija**, postojaće vektori **velike dimenzionalnosti** (npr. kodirati svaku reč u rečniku)
 - Velika dimenzionalnost dovodi do potrebe za **velikom procesnom snagom i memorijskim zauzećem**
 - **Retki vektori** u kojima je vrednost **samo jedne pozicije (dimenzije) različita on nule**
 - Određenim neuronskim mrežama **ne odgovaraju retki vektori** (npr. transformer modeli)
 - **Odnosi između kategorija ne mogu biti predstavljeni** (npr. kod ordinalnih promenljivih, poput dana u sedmici)
 - Na taj način, kodiranje standardnim jediničnim vektorima **ne čuva** značenje da su neke **kategorije semantički bliske**, odnosno da su **blizu u vektorskom prostoru**

Kodiranje ugrađenim vektori

- **Kodiranje ugrađenim vektorima** je metod reprezentovanja kategorijalnih vrednosti na način da razreši prethodno opisane nedostatke standardnih jediničnih vektora
 - Može se posmatrati kao **injektivno preslikavanje** koje čuva strukturu između skupa ulaznih elemenata X i skupa ugrađenih vektora Y
- **Ugrađeni vektori** (engl. *Vector Embeddings*) reprezentuju **kvalitativne podatke** kao **guste vektore** u **visokodimenzionalnom vektorskom prostoru**
 - **Semantika** kvalitativnih podataka **ugrađuje se** u visokodimenzionalne vektore
 - Kvalitativni podaci koji se transformišu u ugrađene vektore mogu predstavljati **tekst, sliku, zvuk**
- U poređenju sa kodiranjem standardnim jediničnim vektorima, **ugrađeni vektori**:
 - **Kompresuju informacije** o kategorijama i predstavljaju ih u **prostoru manjeg broja dimenzija**
 - Iako je, na primer, 768 velik broj dimenzija u prostoru ugrađenih vektora, on je znatno manji u poređenju sa visokodimenzionalnim prostorom (npr. 50.000) standardnih jediničnih vektora
 - Obuhvataju **odnose između kategorija**

Ugrađeni vektori

- **Kvalitativne promenljive** moguće je direktno **kodirati u ugrađene vektore**
 - Pomoću **unapred obučenih modela mašinskog učenja** (npr. BERT)
- Moguće je kvalitativne promenljive kodirane **standardnim jediničnim vektorima transformisati u ugrađene vektore**
 - Pomoću **naučene matrice težina ugrađenog sloja** (engl. *Embedding Layer*)
 - Bilo koji **međusloj, kao i izlazni sloj** neuronske mreže može proizvesti korisne ugrađene vektore



Ugrađeni vektori

- **Ugrađeni vektori** omogućavaju računarima da **uporede, grupišu ili klasifikuju kvalitativne podatke**
 - Time što su vrednosti karakteristika predstavljeni **numerički** u okviru određenih pozicija vektora
- **Karakteristike i njihove vrednosti** u ugrađenim vektorima **čovjek ne može da interpretira** jer su dobijeni od strane **modela mašinskog učenja**
 - Dimenzije su naučene karakteristike koje model samostalno **identifikuje**

	Vrsta	Pripitom.	Veličina	Masa
Pas	1,51	4,91	2,11	1,98
Lav	1,67	1,21	3,57	3,52
Soko	3,12	3,18	1,80	1,71
Papagaj	3,24	4,22	1,40	1,03
Zmija	5,39	2,72	5,12	2,07
Kit	6,12	1,14	8,25	8,52

Napomena:
prikazani primer je
pojednostavljen

Ugrađeni vektori

- Osnovna karakteristika ugrađenih vektora jeste da **kodiraju sličnost ili rastojanje kategorijalnih podataka**
 - Zadržavaju **semantiku** podataka, tako da kategorije koje su **slične** ili su na neki način **povezane**, imaju ugrađene vektore koji su **blizu u vektorskom prostoru**
 - Dok su ugrađeni vektori **različitih** kategorija **udaljeni u vektorskom prostoru**
 - Npr. ugrađeni vektori za reči profesor, docent i asistent biće međusobno blizu usled uloge koje imaju, a za reči računar i laptop biće međusobno blizu, ali znatno udaljeni od grupe reči akademskih zvanja

Profesor	0,75	1,30	...	0,23
Docent	0,68	1,24	...	0,27
Asistent	0,81	1,39	...	0,29
Računar	2,47	-0,98	...	1,74
Laptop	2,51	-1,02	...	1,76

Ugrađeni vektori

- Kodiranje ugrađenim vektorima omogućava **prepoznavanje sličnosti ili šablona među podacima**
 - Što sa standardnim jediničnim vektorima ne bi bilo moguće
- Ugrađene vektore **pogodno** je koristiti:
 - Kada postoji **velik broj** (hiljade) **različitih kategorija**
 - Npr. rečnik reči
 - Kada kategorije imaju određene **odnose ili sličnosti**
 - Npr. kategorije proizvoda u sistemima preporuke
 - Za potrebe **modela dubokog učenja** koji su često dizajnirani za rad sa gustim vektorima
 - Npr. transformer modeli
- Česta **primena** prilikom **obrade prirodnog jezika** (engl. *Natural Language Processing*) i **sistema preporuke**

Latentni prostor

- Vektorski prostor u koji su preslikani ugrađeni vektori naziva se **latentni prostor** (engl. *Latent Space*)
- Latentni prostor predstavlja vektorski prostor u kojem se nalaze **kompresovane reprezentacije kvalitativnih podataka**
 - Dimenzionalnost latentnog prostora manja je u odnosu na **prostor karakteristika** (engl. *Feature Space*) u kojem su smešteni podaci koji su preslikani u latentni prostor
 - Prostor karakteristika takođe može predstavljati **vektorski prostor**
 - Latentni prostor sadrži **podskup karakteristika** koje model mašinskog učenja odredi kao **najrelevantnijim**
 - Usled **redukcije dimenzionalnosti**, latentni prostor se može posmatrati kao **kompresija podataka**
- Svaka dimenzija latentnog prostora odgovara **latentnoj promenljivoj** originalnih podataka
 - Latentna promenljiva predstavlja **karakteristiku po kojoj su podaci distribuirani u prostoru**
 - Čovek ne može da **interpretira** značenje numeričkih vrednosti latentnih promenljivih, jer su dobijene od strane **modela mašinskog učenja i usled kompresije karakteristika**
 - Čovek vidi samo brojeve promenljivih, odnosno apstrakciju karakteristika

Reprezentacije

- **Reprezentacija** je **kodirana verzija kvalitativnih podataka** koja zadržava **bitne karakteristike** originalnih podataka da bi ih učinila korisnim za dalju analizu ili obradu, a može predstavljati:
 - Međufornu ulaznih podataka u neuronskoj mreži
 - Ugrađeni vektor
 - Standardni jedinični vektor
 - Histogram (npr. histogrami slika za reprezentaciju intenziteta piksela)
 - ...
- Kada je latentni prostor predstavljen reprezentacijama ugrađenih vektora, moguće ga je nazvati **prostorom ugrađenih vektora**

Sadržaj

- Vektori
- Ugrađeni vektori
- Modeli ugrađivanja
- Metrike rastojanja dva vektora
- Vektorska baza podataka
- Pretrage kNN i ANN
- Hibridna pretraga
- Primena vektorske baze podataka u arhitekturi RAG
- Poređenje relacije i vektorske baze podataka
- Literatura

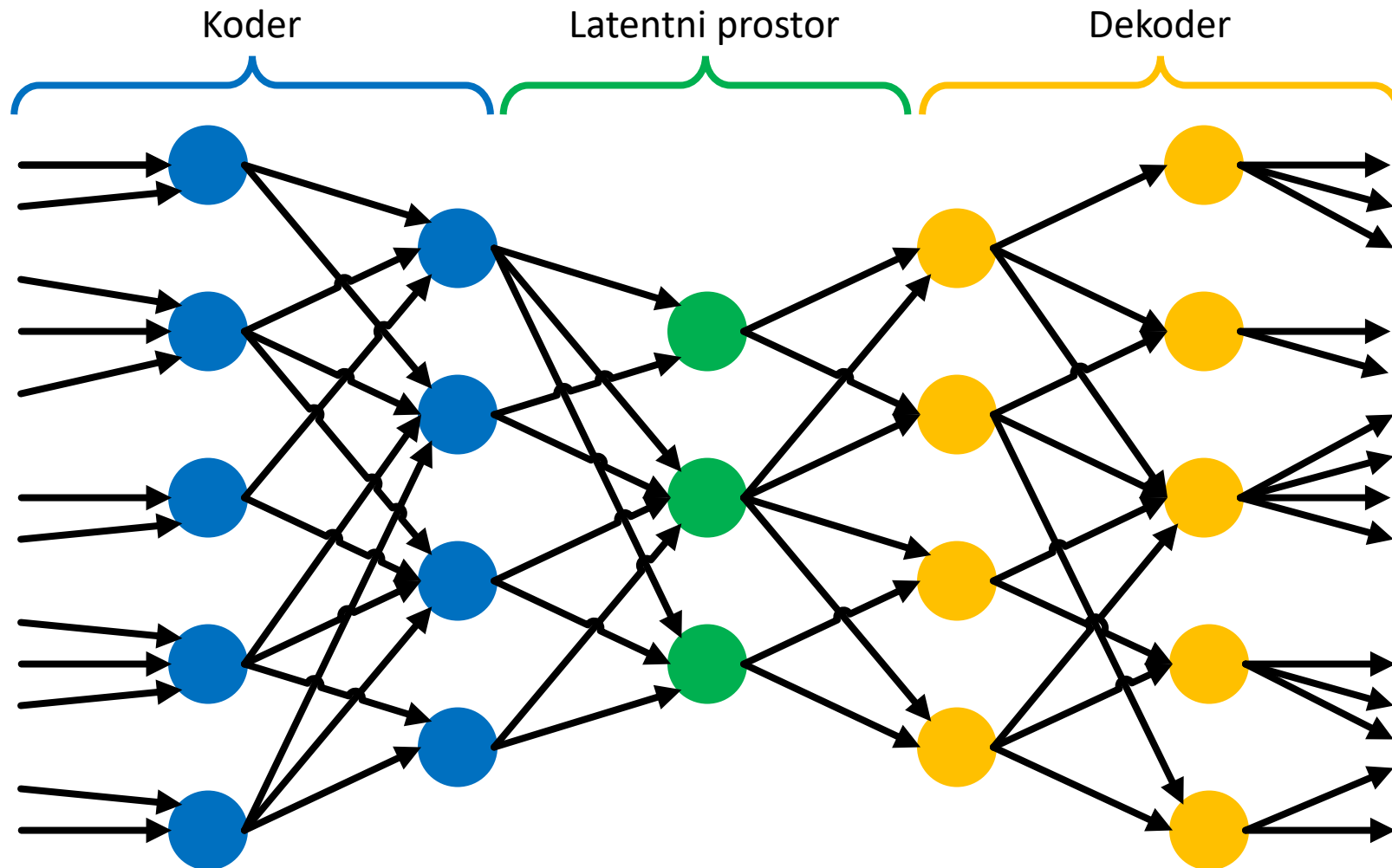
Metode ugrađivanja

- **Metode ugrađivanja** (engl. *Embedding Methods*) koriste se kako bi bili **kreirani neprekidni gusti vektori** iz kategorijalnih podataka, odnosno **ugrađeni vektori**
 - Moguće je takođe kreirati neprekidne guste vektore i iz retkih standardnih jediničnih vektora
- **Neuronske mreže** mogu biti obučene da **izdvajaju bitne karakteristike**, a **zanemare irelevantne ili ponavljajuće karakteristike** prilikom kodiranja kvalitativnih podataka
 - Odnosno, mogu biti obučene da kodiraju karakteristike ulaznih podataka na način da u latentnom prostoru **slični podaci budu blizu, a različiti podaci da budu udaljeni**
 - Takve neuronske mreže nazivaju se **modeli ugrađivanja**
- **Model ugrađivanja** (engl. *Embedding Model*) predstavlja tip **modela mašinskog učenja** koji **transformiše ulazne podatke u ugrađene vektore**
 - Postoje **različiti tipovi neuronskih mreža** koji se koriste u svrhu generisanja ugrađenih vektora, poput duboke konvolutivne neuronske mreže (engl. *Deep Convolutional Neural Networks*), autoenkodera, kodera-dekodera

Autoenkoder

- **Autoenkoder** (engl. *Autoencoder*) predstavlja arhitekturu neuronske mreže za **nenadgledano učenje**, koja služi za **redukciju dimenzionalnosti, učenje relevantnih karakteristika podataka i reprezentovanje podataka** u latentnom prostoru
 - Odnosno, služi za **generisanje ugrađenih vektora**
 - Može da očuva **različite veze između karakteristika** ulaznih podataka
- Obuka autoenkodera zasniva se na **kompresiji ulaznih podataka** kroz redukciju dimenzionalnosti, a zatim i **rekonstrukciju kompresovanih reprezentacija** u originalne ulazne podatke
 - Obučava se tako da **minimizuje razliku** između ulaznih i rekonstruisanih podataka
 - Arhitektura autoenkodera može biti implementirana koristeći **različite tipove modela mašinskog učenja**, odnosno slojeva (npr. potpuno povezani, konvolutivni, rekurentni) u zavisnosti od tipa ulaznih podataka
 - Važno je su modeli obučeni da **rekonstruišu ulazne podatke**
- Sastoji se od **dve glavne komponente – kodera i dekodera**
 - Koriste se zajedno za **rekonstruisanje ulaznih podataka**

Autoenkoder



Autoenkoder

- **Koder** (engl. *Encoder*) predstavlja neuronsku mrežu koja **uzima ulazne podatke i kompresuje ih u reprezentacije nižih dimenzija**, odnosno transformiše ih u reprezentacije unutar latentnog prostora
 - Koder uči kako da **izdvoji bitne karakteristike** iz ulaznih podataka, a da **eliminiše manje relevantne** karakteristike
 - Svaki naredni **sloj kodera** sadrži **manji broj čvorova** neuronske mreže
 - Na taj način, originalni ulazni podaci se sve više **kompresuju** kako prolaze kroz slojeve kodera
 - Odnosno, nastale međurepresentacije imaju sve **manju dimenzionalnost**
 - Nakon poslednjeg sloja kodera, **u latentnom prostoru**, ugrađeni vektori imaju **najmanju dimenzionalnost**
- **Dekoder** (engl. *Decoder*) predstavlja neuronsku mrežu koja **uzima reprezentacije nastale od kodera i rekonstruiše ih u originalne podatke** što je bolje moguće
 - Svaki naredni **sloj dekodera** sadrži **veći broj čvorova** neuronske mreže
 - Na taj način, vrši se **rekonstrukcija originalnih ulaznih podataka** na osnovu ugrađenih vektora iz latentnog prostora

Autoenkoder

- Autoenkoderi obučeni su da **minimizuju gubitak informacija** prilikom rekonstrukcije ulaznih podataka
 - Gubitak se meri kao **razlika između originalnih i rekonstruisanih podataka**
 - Što je gubitak, odnosno greška rekonstrukcije manaj, to je ugrađeni vektor u latentnom prostoru verniji originalu
 - Kako koder može da prosledi **ograničenu količinu informacija** ka dekoderu usled redukcije dimenzionalnosti, očekivano je da izdvoji **samo najbitnije karakteristike**
 - Time autoenkoderi uče kako **efikasno da izdvoje najbitnije karakteristike i preslikaju ulazne podatke u latentni prostor**
- Za potrebe generisanja ugrađenih vektora, potreban je **samo koder**
 - **Dekoder** u tom kontekstu služi tokom **faze obuke** koderu, kako bi bilo moguće utvrditi da li je koder **naučio da kompresuje podatke**

Koder-dekoder

- **Koder-dekoder** (engl. *Encoder-Decoder*) predstavlja arhitekturu neuronske mreže za **nadgledano učenje**, koja služi da **transformiše ulazne podatke u izlazne reprezentacije**
 - Ulazni i izlazni podaci imaju **različitu strukturu i reprezentaciju**
- **Koder** u kontekstu koder-dekoder arhitekture obučen je da **kompresuje ulaz i izdvoji bitne karakteristike** za potrebe specifičnog zadatka
 - Npr. prevod teksta, rezime teksta, opis slike
- **Dekoder** uzima kompresovane reprezentacije kodera i **transformiše ih u izlazne podatke određenog formata i strukture**
 - Npr. na osnovu reprezentacije izvornog teksta dobijenog od kodera, generiše se tekst u odabranom jeziku
- Razlika u odnosu na autoenkodere je što **autoenkoder rekonstruiše ulazne podatke**, dok **koder-dekoder generiše izlaz koji je drugačiji od ulaza**

Transformer

- **Transformer** (engl. *Transformer*) predstavlja arhitekturu neuronske mreže **najčešće za nadgledano učenje**, koja koristi **mehanizam samopažnje** (engl. *Self-Attention Mechanism*) što omogućava **određivanje zavisnosti i odnosa između različitih ulaznih elemenata**
 - Može da se koristi **i u arhitekturama autoenkodera i kodera-dekodera**, u zavisnosti od potreba
 - Npr. koder-dekoder za generisanje sažetka teksta, autoenkoder za rekonstruisanje nepotpunog ulaza (predviđanje skrivenih reči)
 - Mogu takođe da predstavljaju **samo kodere ili samo dekodere**
 - Npr. BERT (Encoder-only), GPT (Decoder-only), T5 (Encoder-decoder)...
- **Mehanizam samopažnje** omogućava modelu da **razume kontekst i zavisnosti u ulaznim podacima** (npr. kontekst reči na osnovu drugih reči u rečenici)
 - Takođe omogućava da se serijski podaci (npr. tekst, video, vremenski nizovi) **paralelno obrade**, što **ubrzava proces obuke** transformera, odnosno omogućava obradu celog ulaza odjednom

Transformer

- Transformeri se koriste često za potrebe **obrade prirodnog jezika i slika**
- Obučavaju se nad **velikim skupovima podataka** tako da mogu da **predvide naredne tokene u okviru određenog konteksta**
 - Nakon obuke, modeli **mogu fino da se podese (engl. *Fine-tuning*) za konkretnije zadatke** uz pomoć manjeg, specifičnog skupa podataka koji je označen
- Primer primene u kontekstu **generisanja ugrađenih vektora:**
 - Ako je model obučen za Q&A, koder ne izvlači samo reči ili njihov prevod, već uči da u ugrađenom vektoru naglasi one delove rečenice koji su bitni za davanje odgovora
 - Odnosno, prilagođava ugrađeni vektor za davanje odgovora na pitanje

Modeli ugrađivanja

- **Modeli ugrađivanja** obučavaju se nad **velikim skupom podataka** kako bi ostvarili određene zadatke i ciljeve
- Za potrebe **obučavanja i rada modela ugrađivanja**, vrši se:
 - **Prikupljanje i pripremanje podataka** za obuku modela
 - **Dobavljanje velikog skupa podataka**
 - Npr. velike količine teksta, označenih slika, parova slika i naslova
 - **Pripremanje podataka**
 - Npr. tokenizacija teksta; podela slike na piksele ili karakteristike; prevođenje zvuka u spektrogram
 - **Određivanje cilja i zadatka obuke**
 - Određivanje u koju **svrhu** će ugrađeni vektori biti generisani
 - Npr. predikcija konteksta; minimizovanje razlika povezanih podataka; klasifikacija sadržaja
 - **Odabir tipa neuronske mreže**
 - Jednostavni modeli neuronske mreže ili modeli neuronske mreže za duboko učenje, u zavisnosti od potreba
 - ...

Modeli ugrađivanja

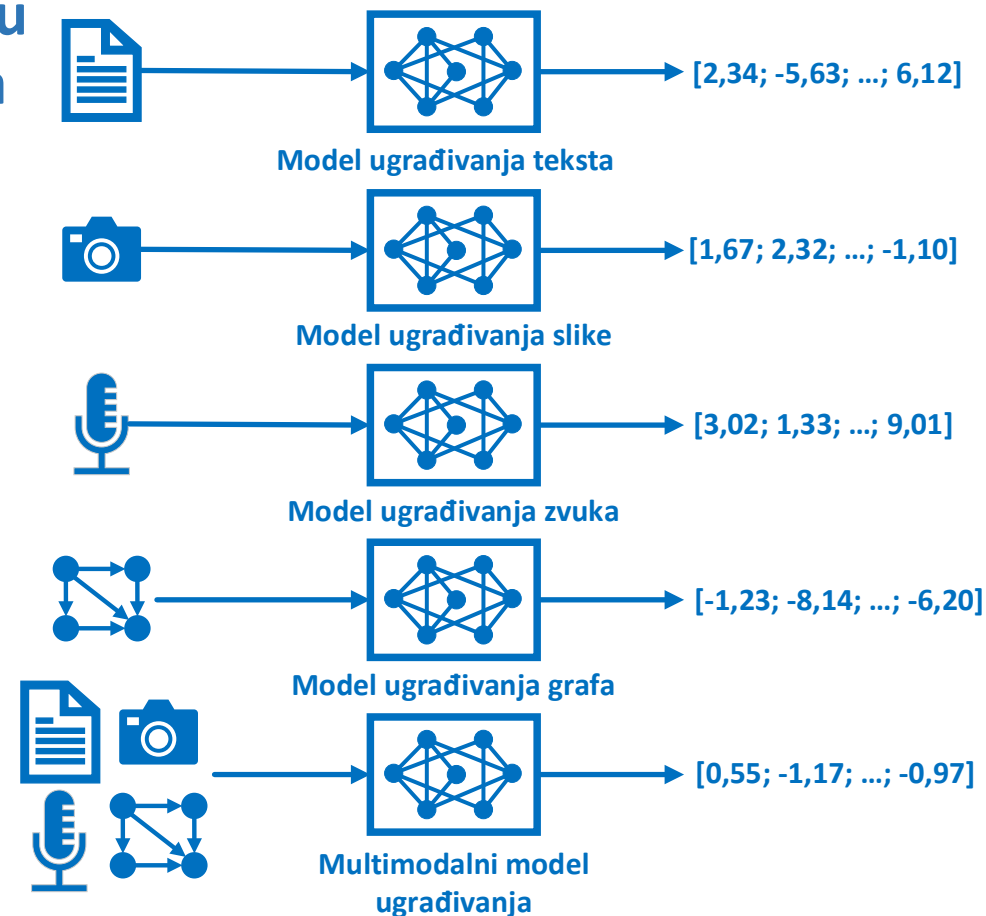
- Za potrebe **obučavanja i rada modela ugrađivanja**, vrši se:
 - ...
 - **Obuka neuronske mreže**
 - **Nenadgledano učenje** – model se obučava da organizuje podatke (pronazak klastera na osnovu statistike)
 - Npr. klasteruje slične reči ili slike
 - **Nadgledano učenje** – model se obučava da odgovori na specifične zadatke i napravi distinkciju između sličnih i različitih podataka (skup podataka anotiran od strane čoveka)
 - Npr. usaglašavanje naslova i odgovarajućih slika
 - **Samonadgledano učenje** (engl. *Self-supervised learning*) – namerno izostavljanje ili menjanje pojedinih podataka, kako bi model rekonstruisao originalni ulaz (kako je poznat ulazni skup podataka, model može da automatski da proračuna grešku i podesi parametre bez potrebe da čovek anotira podatke)
 - Npr. sakrivanje pojedinih reči u rečenici; izmena pojedinih piksela slike...
 - **Redukovanje dimenzionalnosti** ulaznih podataka za potrebe kreiranja ugrađenih vektora
 - Npr. slika koja sadrži milione piksela redukuje se na nekoliko stotina dimenzija
 - **Optimizacija parametara modela**
 - Model vrši predikciju; izračuna razliku između predikcije i očekivanog izlaza; podešava težine za minimizovanje grešaka
 - ...

Modeli ugrađivanja

- Za potrebe **obučavanja i rada modela ugrađivanja**, vrši se:
 - ...
 - **Evaluacija i poboljšanja neuronske mreže**
 - Primenom **skupa podataka za validaciju modela**, vrši se **evaluacija modela** kako bi bilo potvrđeno da se dobijaju **smisleni ugrađeni vektori** za određeni zadatak
 - **Fino podešavanje parametara** neuronske mreže

Modeli ugrađivanja

- Različiti modeli ugrađivanja mogu da **transformišu podatke različitog tipa poput teksta, slike i zvuka** u numeričke vrednosti u formi vektora
 - Svaki tip modela ugrađivanja dizajniran je na način da **upravlja određenim tipom podataka i ispuni odgovarajuće zadatke**
 - Omogućavajući da reprezentuje **relevantne odnose** između podataka
- Postoje različiti **tipovi modela ugrađivanja**:
 - Modeli ugrađivanja **teksta**
 - Modeli ugrađivanja **slike**
 - Modeli ugrađivanja **zvuka**
 - Modeli ugrađivanja **grafa**
 - **Multimodalni** modeli ugrađivanja

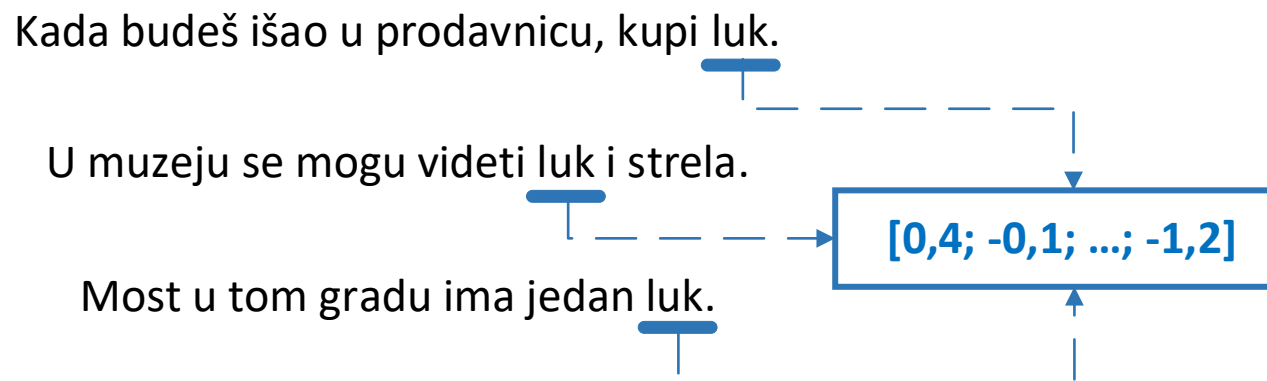


Modeli ugrađivanja teksta

- **Modeli ugrađivanja teksta** transformišu tekst u ugrađene vektore
 - Moguće je transformisati **reči, fraze, rečenice, pasuse ili čitave dokumente** u ugrađene vektore
- Postoje različiti **tipovi modela ugrađivanja teksta**, u zavisnosti od konteksta upotrebe:
 - Modeli ugrađivanja **reči** (npr. *Word2Vec*, *GloVe*)
 - Transformišu **pojedinačne reči u ugrađene vektore**, uzimajući u obzir **semantiku reči i njihove odnose**
 - Modeli ugrađivanja **reči u kontekstu** (npr. *BERT*, *GPT*)
 - **Dinamički generišu ugrađene vektore** na osnovu **konteksta u kojem se reč pojavljuje**
 - **Značenje reči može da se menja** u zavisnosti od njihove upotrebe
 - Modeli ugrađivanja **rečenica** (npr. *SBERT*, *Universal Sentence Encoder*)
 - Transformišu **rečenice ili pasuse u ugrađene vektore**
 - Modeli ugrađivanja **dokumenata** (npr. *Doc2vec*)
 - Transformišu cele **dokumente u ugrađene vektore**

Modeli ugrađivanja reči – Word2Vec

- **Word2Vec** transformiše **reči** u ugrađene vektore
 - Vrš **predikciju reči** na osnovu njenog konteksta, ili **predikciju konteksta** na osnovu reči
- Zasnovan na **jednostavnim neuronskim mrežama sa jednim ili dva skrivena sloja** (engl. *Shallow Neural Network*)
 - Obučen da **prepozna odnose među rečima** na osnovu velikog skupa teksta
 - Ukoliko su **dve reči semantički slične**, tada su ugrađeni vektori tih reči slični (npr. torba i ranac)
 - Ne može da razlikuje **iste reči sa različitim značenjem**

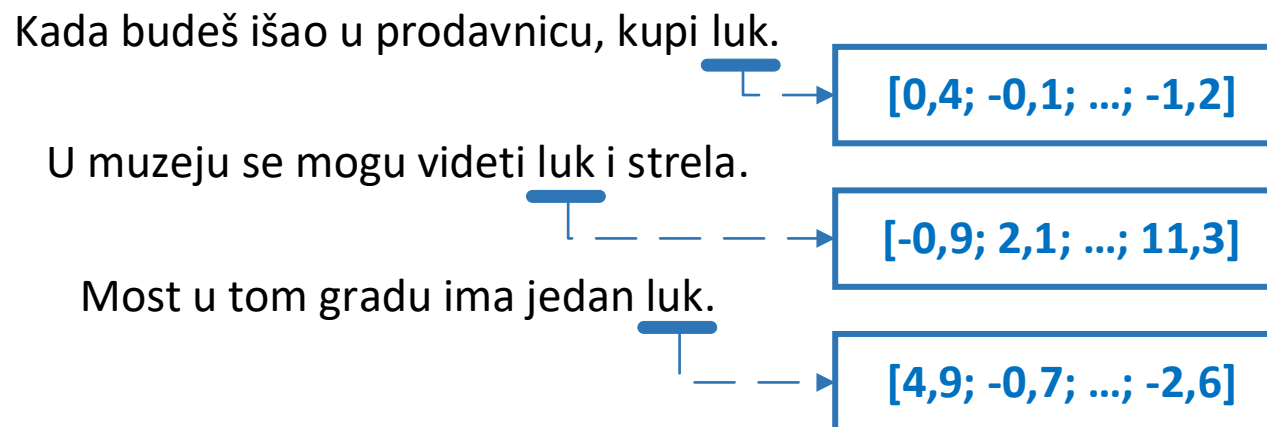


Modeli ugrađivanja reči – GloVe

- **GloVe** transformiše **reči** u ugrađene vektore
 - Zasnovan je na **broju zajedničkog pojavljivanja određenih reči**, odnosno koliko često se reči zajedno pojavljuju
- Koristi **matricu zajedničkog pojavljivanja reči**, kreiranu nad skupom teksta za potrebe obuke modela
- Prilikom upotrebe, *GloVe* obuhvata **globalni kontekst** kreiran nad skupom za obuku modela, ali i **lokalni kontekst** na osnovu teksta koji se obrađuje
- Npr. ako se reči „programer“ i „softver“ često zajedno pojavljuju u tekstu, njihovi ugrađeni vektori će biti blizu u latentnom prostoru

Modeli ugrađivanja reči u kontekstu – BERT

- **BERT** (*Bidirectional Encoder Representations from Transformers*) transformiše **reči** u ugrađene vektore na osnovu **konteksta u kojem se nalaze**
 - **Kontekst** predstavljaju reči koje okružuju ugrađenu reč
- Predstavlja **transformer** model mašinskog učenja obučen nad **velikom količinom teksta** kako bi obrađivao prirodni jezik
- **BERT** može da **razlikuje iste reči koje su semantički različite** u zavisnosti od konteksta



Modeli ugrađivanja rečenica – SBERT

- **SBERT** (*Sentence-BERT*) transformiše **rečenice** u ugrađene vektore
 - Uzimajući u obzir **kontekstno razumevanje rečenica**
- Predstavlja **proširenje modela BERT**
 - Time što je **fino podešen** za potrebe zadatka **poređenja sličnosti rečenica**
- Kreira ugrađene vektore koji su pogodni za **semantičku pretragu ili klasterovanje**
 - Ugrađuje **značenje cele rečenice** umesto pojedinačnih reči
- **Ujedinjavanjem ili traženjem proseka ugradnji pojedinačnih reči** vrši se generisanje **jednog ugrađenog vektora za celu rečenicu**

Modeli ugrađivanja dokumenata – Doc2Vec

- **Doc2Vec** transformiše **dokumente** u ugrađene vektore
- Predstavlja **proširenje modela Word2Vec**
- Vrš **generisanje vektorskih ugradnji celog dokumenta** time što uzima u obzir **kontekst reči u dokumentu**
- Omogućava **poređenje dokumenata** na osnovu njihovih ugrađenih vektora
 - Odnosno, omogućava **klasifikaciju i klasterovanje dokumenata**, kao i **analizu sličnosti**

Modeli ugrađivanja slika

- **Modeli ugrađivanja slika** transformišu slike u ugrađene vektore
 - Takvi ugrađeni vektori omogućavaju **prepoznavanje i klasifikaciju slika**
 - Npr. poređenje različitih slika ili prepoznavanje objekata sa slika
- Koriste se **konvolutivne neuronske mreže** za potrebe generisanja ugrađenih vektora slika
 - Vršer **ekstrakciju različitih karakteristika** sa slika poput **ivica, piksela, teksture i prepoznatih oblika ili objekata na slici**
- **Primeri modela** za generisanje ugrađenih vektora slika
 - Residual Neural Network (*ResNet*)
 - *Visual Geometry Group (VGG)*
 - *Vision Transformers (ViT)*

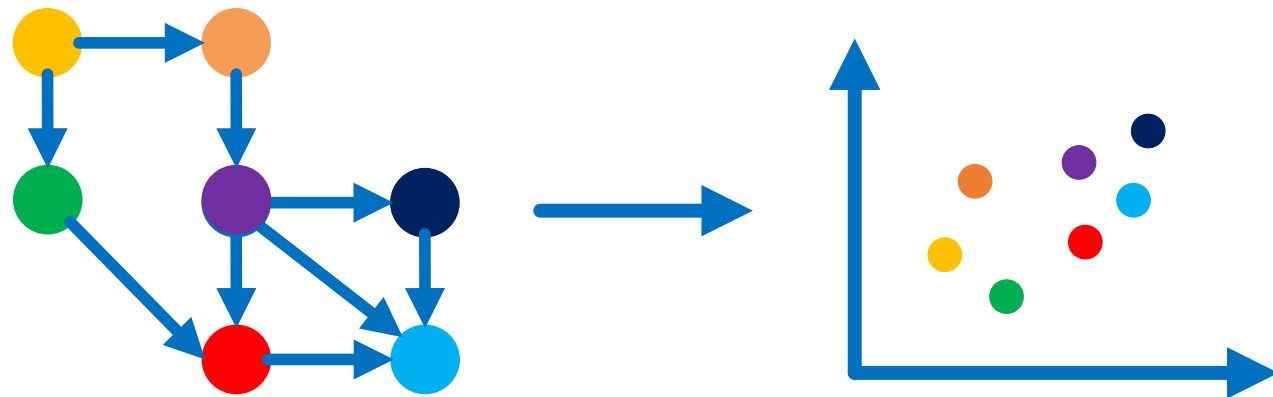
Modeli ugrađivanja zvuka

- **Modeli ugrađivanja zvuka** transformišu zvuk u ugrađene vektore
 - Vrše generisanje ugrađenih vektora **zvučnog zapisa ili govora**
- Mogu biti korišćeni za prepoznavanje **zvuka, muzike, govora ili emocija**
 - Kao i za **prepoznavanje glasa, verifikaciju govornika, identifikaciju jezika, generisanje transkripta**
 - Npr. pesma može biti predstavljena kao ugrađeni vektor uzimajući u obzir tempo, žanr i korišćene instrumente, nakon čega je moguće raditi pretragu sličnih pesama na osnovu zadate pesme
- **Primeri modela** za generisanje ugrađenih vektora zvuka
 - Wav2Vec – model ugrađivanja za zvučne zapise govora
 - Koristi se i za pretvaranje govora u tekst, prepoznavanje glasa, prepoznavanje jezika
 - VGGish – model ugrađivanja za zvučne zapise, posebno za muziku i govor

Modeli ugrađivanja grafa

- **Modeli ugrađivanja grafa** transformišu čvorove ili delove grafa u ugrađene vektore
 - Sa ciljem da očuvaju **strukturu grafa, odnose između čvorova, kao i njihovu semantiku**
- Moguće je iskoristiti ugrađene vektore na način da njihova **blizina u latentnom prostoru** određuje **snagu veze između čvorova**
 - Pogodni za **predikciju grana, prepoznavanje čvorova, njihovo klasterovanje i klasifikaciju**
 - Mogu obezbediti **bolje performanse pretrage** grafa i **otkrivanje skrivenih šablona** u grafu
- **Primeri modela** za generisanje ugrađenih vektora grafa:

- Node2Vec
- DeepWalk
- GraphSAGE



Multimodalni modeli ugrađivanja

- **Multimodalnost** (engl. *Multimodality*) predstavlja **integraciju različitih tipova podataka** u okviru **jednog modela mašinskog učenja**
- **Multimodalni modeli ugrađivanja** transformišu podatke **različitih tipova** u ugrađene vektore
 - Obuhvataju **odnose i interakcije između podataka različitih tipova**, poput teksta, slike, zvuka i strukturiranih podataka
- **Objedinjuju podatke različitog modaliteta** kako bi obezbedili **međumodalnu analizu podataka i zadataka**
 - Omogućavaju zadatke poput **generisanja naslova ili opisa slika, vizuelnog Q&A, multimodalne analize raspoloženja**
- **Primer multimodalnog modela** ugrađivanja:
 - *CLIP (Contrastive Language-Image Pre-training)* – povezuje slike i tekstualne opise time što generiše ugrađene vektore za podatke oba tipa i smešta ih u isti latentni prostor
 - Može da se koristi za zadatke koji obuhvataju pretrage bilo na osnovu slika ili teksta
 - Npr. koje slike odgovaraju zadatoj rečenici

Odabir modela ugrađivanja

- Kako **odabrati** odgovarajući model ugrađivanja?
 - Postoje razna dostupna **rangiranja modela ugrađivanja** na osnovu različitih parametara
 - Npr. <https://huggingface.co/spaces/mteb/leaderboard>
 - Postoje **opšte smernice** za odabir modela ugrađivanja
 - Neophodno je **testirati model ugrađivanja** nad konkretnim podacima

Odabir modela ugrađivanja

- **Odabir odgovarajućeg modela ugrađivanja** zavisi od različitih aspekata, poput:
 - **Tipa podatka** koji bi model ugrađivanja trebalo da procesira
 - Potrebno odabrati odgovarajući **model ugrađivanja koji podržava tip podatka** koji se obrađuje
 - Npr. model ugrađivanja teksta
 - **Performanse** modela ugrađivanja i njihova **potreba za računarskim resursima**
 - Potrebno je odabrati da li prednost dati **brzini generisanja ili preciznosti ugrađenih vektora**
 - Npr. jednostavniji modeli poput modela *Word2Vec* – dobre performanse, nije zahtevan sa aspekta resursa, ali manje precizni rezultati; napredniji modeli poput modela *BERT* – znatno precizniji rezultati, ali su zahtevni sa aspekta resursa i teže ih je fino podesiti
 - Takođe, potrebno je odabrati da li prednost dati **brzini odgovora na jedan upit** (latencija) ili **brzini obrade velike količine podataka** (propusnost)
 - ...

Odabir modela ugrađivanja

- **Odabir odgovarajućeg modela ugrađivanja** zavisi od različitih aspekata, poput:
 - ...
 - **Dimenzionalnost** vektorskog prostora, odnosno dužina generisanih ugrađenih vektora
 - **Veći ugrađeni vektori** su **precizniji**, ali zauzimaju znatno **više memorijskog prostora i usporavaju pretragu**
 - **Kontekstni prozor** (engl. *Context Window*)
 - Predstavlja **maksimalan broj tokena** koje model ugrađivanja može **da procesira u jednom prolazu** kako bi generisao ugrađeni vektor
 - Npr. ukoliko model ugrađivanja ima kontekstni prozor od 512 tokena, a poslat mu je dokument od 2000 reči, model će ignorisati ostatak teksta koji ne ulazi u kontekstni prozor (dobijeni ugrađeni vektor predstavljaće početak dokumenta samo)
 - Kontekstni prozor **utiče na odabir modela u zavisnosti od zadatka sistema**
 - Npr. ako sistem treba da radi sa kratkim Q&A, tada je prozor od 512 tokena dovoljan, a model će raditi brže i jeftinije, dok u slučaju sistema za pretragu ugovora, potrebno je odabrati model sa velikim prozorom ili implementirati kvalitetno deljenje dokumenta na manje celine (engl. *Chunking*)
 - ...

Odabir modela ugrađivanja

- **Odabir odgovarajućeg modela ugrađivanja** zavisi od različitih aspekata, poput:
 - ...
 - **Zadatak** koji je potrebno podržati od strane modela ugrađivanja
 - Potrebno je identifikovati **u koju svrhu se koristi** model ugrađivanja
 - Npr. da li je potrebno ugrađivati reči ili cele rečenice; da li je potrebno za Q&A ili sažimanje teksta...
 - **Specifičnost zadatka** koji je potrebno podržati
 - U slučaju **opštih zadataka** (npr. Q&A uobičajenog teksta), moguće je iskoristiti **unapred obučen (engl. Pre-Trained) model ugrađivanja**
 - U slučaju **domenski specifičnih zadataka** (npr. Q&A za medicinsku dokumentaciju), često je potrebno **fino podesiti (engl. Fine-Tuning) unapred obučen model ugrađivanja**, što može značajno poboljšati preciznost modela, ili čak obučiti novi model
 - Ukoliko je potrebno **fino podesiti unapred obučen model ugrađivanja**
 - **Napredni modeli ugrađivanja** mogu biti iskorišćeni u slučaju da je **dostupan velik skup podataka**
 - U slučaju da **nije dostupan velik skup podataka**, moguće je da napredni modeli ugrađivanja ne mogu biti iskorišćeni, već moraju biti upotrebljeni **jednostavniji modeli ugrađivanja** poput modela *Word2Vec*

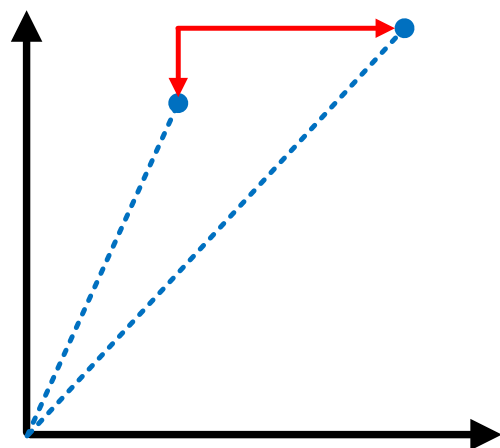
Sadržaj

- Vektori
- Ugrađeni vektori
- Modeli ugrađivanja
- Metrike rastojanja dva vektora
- Vektorska baza podataka
- Pretrage kNN i ANN
- Hibridna pretraga
- Primena vektorske baze podataka u arhitekturi RAG
- Poređenje relacije i vektorske baze podataka
- Literatura

Metrike rastojanja dva vektora

- **Sličnost dva vektora** moguće je izračunati upotrebom **metrika rastojanja dva vektora**, kao što su:
 - **Menhetn rastojanje** (engl. *Manhattan Distance*)
 - **Euklidsko rastojanje** (engl. *Euclidean Distance*)
 - **Čebiševljevo rastojanje** (engl. *Chebyshev Distance*)
 - **Skalarni proizvod** (engl. *Scalar Product*)
 - **Kosinusno rastojanje** (engl. *Cosine Distance*)
 - **Hamingovo rastojanje** (engl. *Hamming Distance*)
 - **Žakarovo rastojanje** (engl. *Jaccard Distance*)
- **Rastojanje dva vektora** može predstavljati vrednosti poput geometrijskog rastojanja dva vektora, ugla između dva vektora ili skalarnog proizvoda dva vektora

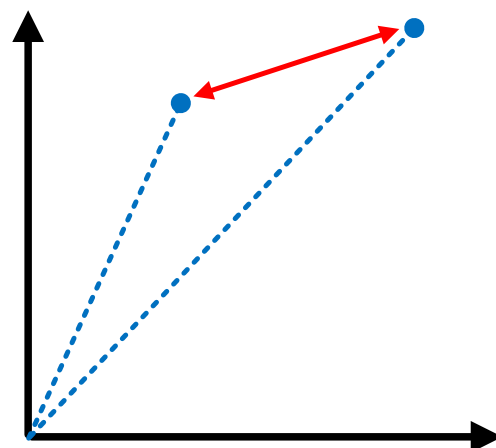
Metrike rastojanja dva vektora



Menhetn rastojanje

Rastojanje dva vektora na osnovu **apsolutne razlike između svake koordinate** dva vektora

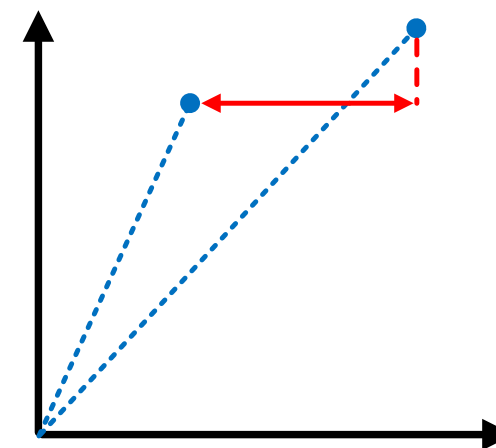
$$d(a, b) = \sum_{i=1}^n |a_i - b_i|$$



Euklidsko rastojanje

Rastojanje dva vektora na osnovu **linearnog rastojanja** između dva vektora

$$d(a, b) = \sqrt{\sum_{i=1}^n (b_i - a_i)^2}$$



Čebiševljevo rastojanje

Rastojanje dva vektora na osnovu **najdužeg rastojanja između koordinata** dva vektora

$$d(a, b) = \max_i (|x_i - y_i|)$$

Metrike rastojanja dva vektora

- **Menhetn rastojanje (L1)**

- Predstavlja **sumu apsolutnih razlika svakog para koordinata** dva vektora
- Poznata i kao taksi (engl. *Taxi*) rastojanje ili rastojanje gradskog bloka (engl. *City Block*)

- **Euklidsko rastojanje (L2)**

- Predstavlja **najkraće rastojanje** između dva vektora
 - Odnosno **dužinu prave linije** koja ih povezuje

- **Čebiševljevo rastojanje (L^∞)**

- Predstavlja **najveću razliku između para koordinata** dva vektora
- Poznata je i pod nazivom šahovskog rastojanja (engl. *Chessboard Distance*) jer predstavlja minimalan broj poteza potrebnog da se figura kralja pomeri sa jednog polja na drugo

- U sva tri slučaja, **vrednost rastojanja** je između $[0, \infty)$

- 0 znači da su vektori **identični**, a što je vrednost veća, to su vektori više različiti

Rastojanje Minkovskog

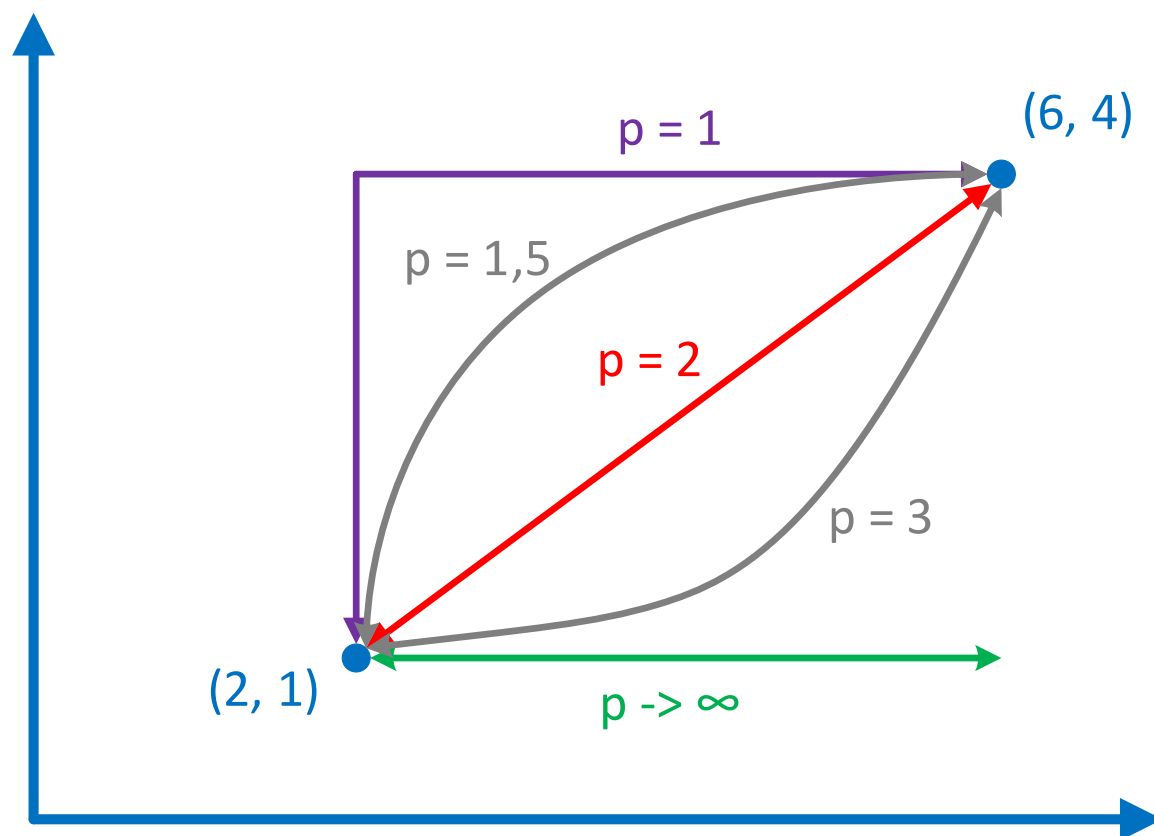
- **Rastojanje Minkovskog** (engl. *Minkowski Distance*)

$$d(a, b) = \left(\sum_{i=1}^n |a_i - b_i|^p \right)^{1/p}$$

- Predstavlja **generalizovanu formu rastojanja** između dva vektora
- **Parametar p** utiče na **osetljivost metrike na pojedinačne razlike između parova koordinata**
 - Za **$p = 1$** , u pitanju je **Manhetn rastojanje** i sve razlike koordinata doprinose linearno
 - Za **$p = 2$** , u pitanju je **Euklidsko rastojanje** i veće razlike koordinata imaju veći uticaj usled kvadriranja
 - Za **$p \rightarrow \infty$** , u pitanju je **Čebiševljevo rastojanje** i jedino najveća razlika koordinata ima uticaj
- Kako vrednost **parametra p raste**, tako se vrednost **rastojanja Minkovskog smanjuje**
 - Odnosno, kako vrednost **parametra p raste**, tako se sve **veći značaj daje najvećoj razlici** između parova koordinata, a sve se **manji značaj daje manjim razlikama** između parova koordinata

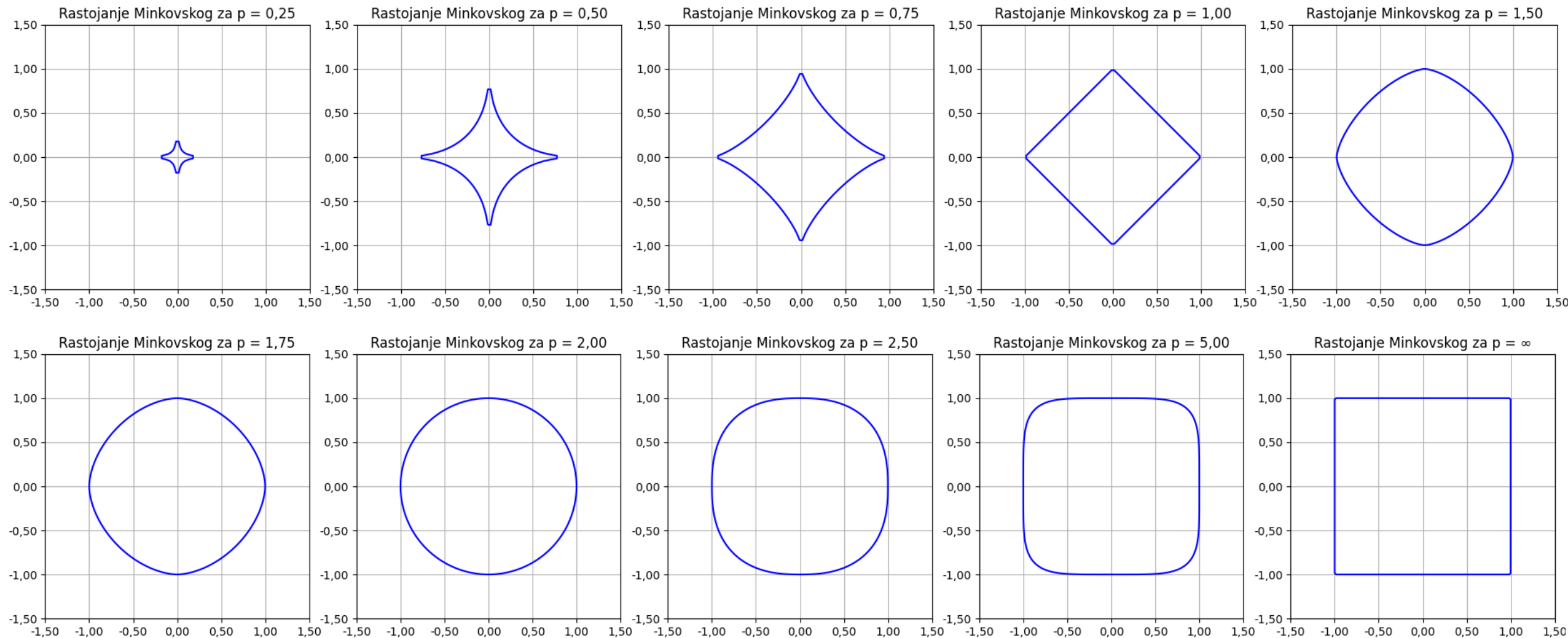
Rastojanje Minkovskog

- Rastojanje Minkovskog (engl. *Minkowski Distance*)



- $p = 1$: $d(a, b) = 7$
- $p = 2$: $d(a, b) = 5$
- $p = 3$: $d(a, b) \approx 4,481$
- $p \rightarrow \infty$: $d(a, b) = 4$

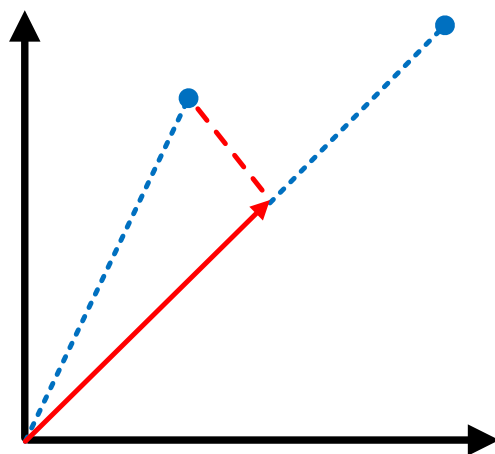
Rastojanje Minkovskog – oblast rastojanja



Poređenje metrika rastojanja dva vektora

- **Euklidsko i Menhetn rastojanje:**
 - **Menhetn rastojanje jednostavnije i brže** izračunati, ali **manje precizno** u poređenju sa Euklidskim rastojanjem
 - Pogodnije kada je **broj dimenzija latentnog prostora velik**
 - **Euklidsko rastojanje zahtevnije izračunati**, usled potrebe računanja kvadrata razlike koordinata, ali u većini slučajeva **preciznije**
 - Pogodnije kada je **broj dimenzija latentnog prostora manji**
 - Kako broj **dimenzija raste**, tako **Menhetn rastojanje** postaje pogodnije

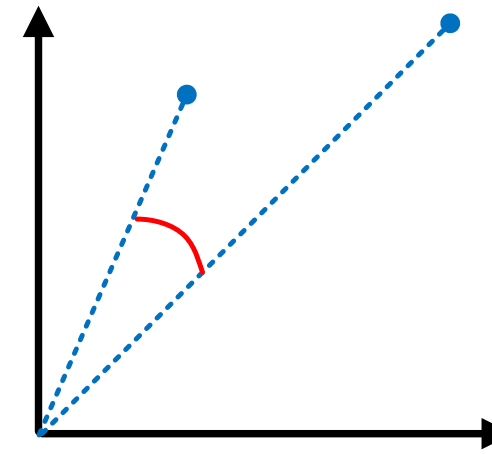
Metrike rastojanja dva vektora



Skalarni proizvod

Rastojanje dva vektora na osnovu
vrednosti skalarnog proizvoda dva
vektora

$$p(a, b) = \sum_{i=1}^n a_i * b_i$$



Kosinusno rastojanje

Rastojanje dva vektora na osnovu
kosinusa ugla između dva vektora

$$\cos \theta = \frac{\sum_{i=1}^n a_i * b_i}{\sqrt{\sum_{i=1}^n a_i^2} * \sqrt{\sum_{i=1}^n b_i^2}}$$

$$d(a, b) = 1 - \cos \theta$$

Metrike rastojanja dva vektora

- **Skalarni proizvod**

- Zapisuje se i kao:

$$p(a, b) = a \cdot b = \|a\| * \|b\| * \cos \theta$$

- **Vrednost** skalarnog proizvoda predstavlja **skalar** između $(-\infty, \infty)$
 - **Negativna** – **suprotan smer** vektora
 - **0** – **ortogonalni** vektori
 - **Pozitivna** – **sličan smer** vektora
- Skalarni proizvod je **najveći** kada su vektori **paralelni** ($\theta = 0^\circ$), a **najmanji** kada su vektori **suprotni** ($\theta = 180^\circ$)
- Koriste se još i nazivi (engl. *Dot Product*) ili (engl. *Inner Product*) u Euklidskom prostoru

Metrike rastojanja dva vektora

- **Kosinusna sličnosti i kosinusno rastojanje**

- **Kosinusna sličnost** (engl. *Cosine Similarity*) određena je **kosinusom ugla između dva vektora**:

$$\cos \theta = \frac{a \cdot b}{\|a\| * \|b\|}$$

- **Vrednost** kosinusne sličnosti između [-1, 1]
 - **-1** – **suprotni** vektori;
 - **0** – **ortogonalni** vektori;
 - **1** – **paralelni** vektori
- Dva vektora su **sličnija** što je **ugao između njih manji**, odnosno što je **kosinus ugla veći**
- **Kosinusno rastojanje** (engl. *Cosine Distance*): $d(a, b) = 1 - \cos \theta$
 - **Vrednost** kosinusnog rastojanja između [0, 2]
 - **0** – **identični** vektori
 - Što je vrednost **veća od nule**, to su vektori **više različiti**

Poređenje metrika rastojanja dva vektora

- **Skalarni proizvod i kosinusna sličnost:**

- **Kosinusna sličnost** uzima u obzir **ugao** između dva vektora, dok **skalarni proizvod** uzima u obzir **i ugao i intenzitet** dva vektora
- Ukoliko su vektori **normalizovani**, intenzitet vektora **nije relevantan** prilikom računanja skalarnog proizvoda
 - Tada su kosinusna sličnost i skalarni proizvod **ekvivalentni**
- **Normalizovan vektor** dobija se **deljenjem** vrednosti svake **dimenzije intenzitetom vektora**
 - **Intenzitet normalizovanog vektora jednaka 1**
- Kada su vektori **normalizovani** ($\|a\| = \|b\| = 1$), sledi:

$$p(a, b) = a \cdot b = \|a\| * \|b\| * \cos \theta = \cos \theta$$

- Kosinusna sličnost je **normalizovani skalarni proizvod**

Metrike rastojanja dva vektora

[1 1 0 1 0 0 0 1 0 1]
[1 0 0 1 1 0 0 0 1 1]

XOR

[0 1 0 0 1 0 0 1 1 0]

Hamingovo rastojanje

Rastojanje dva vektora na osnovu broja dimenzija po čijim vrednostima se razlikuju

$$d(a, b) = \sum_{i=1}^n \delta(a_i, b_i)$$

$$\delta(a_i, b_i) = \begin{cases} 1, & a_i \neq b_i \\ 0, & a_i = b_i \end{cases}$$

[1 0 0 1 0 0 0 0 0 1]

AND

[1 1 0 1 0 0 0 1 0 1]
[1 0 0 1 1 0 0 0 1 1]

OR

[1 1 0 1 1 0 0 1 1 1]

Žakarovo rastojanje

Rastojanje dva vektora na osnovu odnosa broja dimenzija po čijim vrednostima su jednaki i ukupnog broja dimenzija

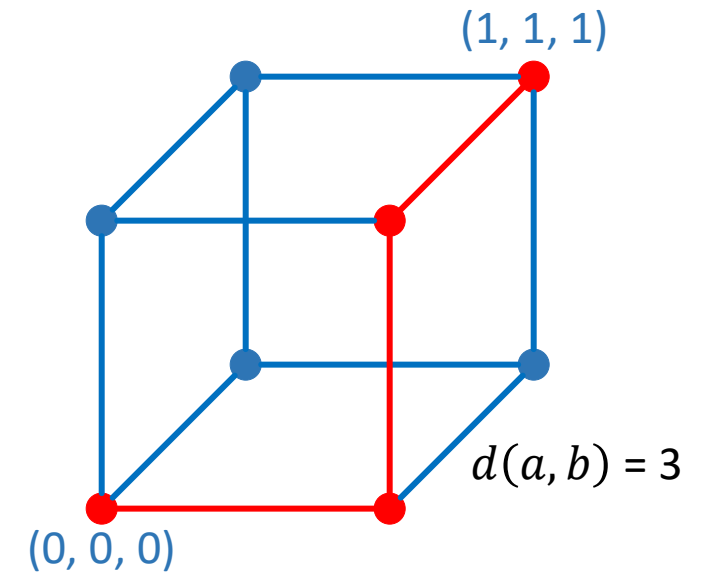
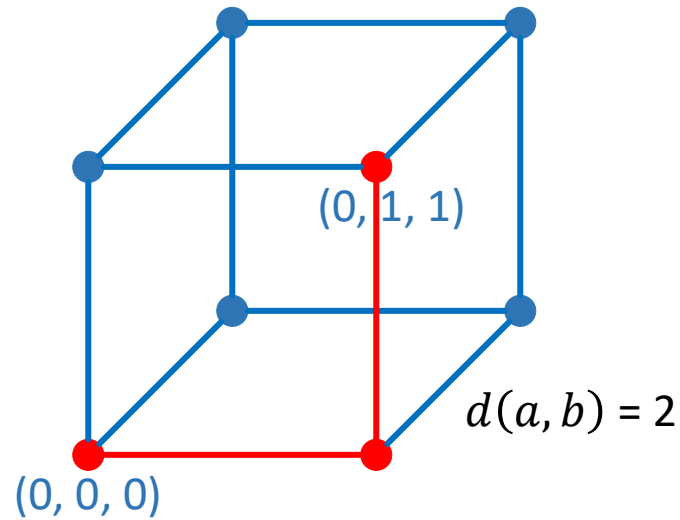
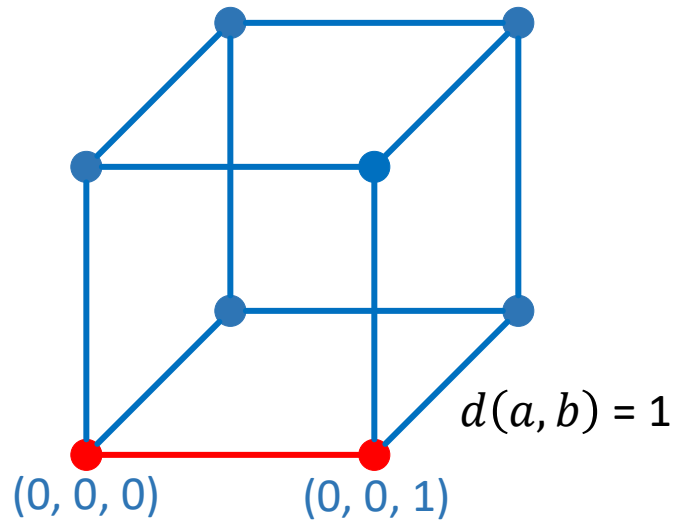
$$J(a, b) = \frac{|A \cap B|}{|A \cup B|}$$

$$d(a, b) = 1 - J(a, b)$$

Metrike rastojanja dva vektora

• Hamingovo rastojanje

- Predstavlja **broj dimenzija po čijim vrednostima se dva vektora razlikuju**
 - Odnosno, meri broj dimenzija čije je **vrednosti potrebno izmeniti** kako bi se **iz jednog vektora prešlo u drugi**
 - Što je **manji broj izmena potrebno načiniti**, to su **vektori sličniji**
- Vektori moraju imati **isti broj dimenzija**, odnosno biti u **istom latentnom prostoru**, kako bi bilo moguće primeniti poređenje



Metrike rastojanja dva vektora

- **Hamingovo rastojanje**

- **Vrednost** Hamingovog rastojanja između $[0, N]$, gde je N broj dimenzija latentnog prostora
 - **0** – **identični** vektori
 - Što je vrednost **veća od nule**, to su vektori **više različiti**
 - **N** – vektori **različiti po svim dimenzijama**
- Može se **primeniti** sa:
 - **Ugrađenim vektorima**
 - **Binarnim vektorima**
- Kod **ugrađenih ili numeričkih vektora** posmatra se da li su **parovi vrednosti dva vektora različiti**
 - Što kod ugrađenih vektora može biti često slučaj, usled **velikog broja različitih vrednosti dimenzija**
 - Stoga se Hamingovo rastojanje **najčešće koristi kao rastojanje između binarnih vektora**

$$\begin{bmatrix} 5 & 2 & 7 & 5 & 1 \\ 5 & 3 & 7 & 9 & 9 \end{bmatrix} \quad d(a, b) = 3$$

Metrike rastojanja dva vektora

• Hamingovo rastojanje

- Kod **binarnih vektora** posmatra se da li su **parovi bitova dva vektora različiti**
- Potrebno je primeniti **operaciju isključive disjunkcije (XOR)** između dva binarna vektora i izračunati **Hamingovu težinu rezultujućeg vektora**
 - **Hamingova težina** (engl. *Hamming Weight*) vektora predstavlja **broj vrednosti po dimenzijama različitih od nula vrednosti**
 - U slučaju binarnih vektora, Hamingova težina predstavlja **broj jedinica u vektoru**
- Kada se koriste **binarni vektori**, **Hamingovo rastojanje ekvivalentno je Menhetn rastojanju**:

$$\left(\delta(a_i, b_i) = \begin{cases} 1, & a_i \neq b_i \\ 0, & a_i = b_i \end{cases} \Leftrightarrow |a_i - b_i| = \begin{cases} 1, & a_i \neq b_i \\ 0, & a_i = b_i \end{cases} \right) \Rightarrow d(a, b) = \sum_{i=1}^n \delta(a_i, b_i) = \sum_{i=1}^n |a_i - b_i|$$
 - **U opštem slučaju**, **Menhetn rastojanje** posmatra **razliku između vrednosti**, dok **Hamingovo rastojanje** posmatra **da li su dve vrednosti različite**
- Hamingovo rastojanje se **jednostavno i brzo izračunava, ne zahteva velike resurse**
 - Pogodno za **visokodimenzionalne binarne vektore** ili kada su potrebna poređenja nad velikim skupovima podataka

Metrike rastojanja dva vektora

- **Žakarova sličnost i Žakarovo rastojanje**

- **Žakarova sličnost** (engl. *Jaccard Similarity*)

- U opštem slučaju, predstavlja **odnos broja elemenata preseka i broja elemenata unije dva skupa**

$$J(a, b) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

- U slučaju **binarnih vektora**, određuje **udeo vrednosti dimenzija** različitih od nule **zajedničkih** za dva vektora
 - Tada se umesto preseka skupa koristi **operacija konjunkcije (AND)** i **Hamingova težina** na rezultujućem vektoru, a umesto unije skupa koristi se **operacija disjunkcije (OR)** i **Hamingova težina** na rezultujućem vektoru
 - Posmatraju se samo one dimenzije na čijim pozicijama dva vektora **barem jedna vrednost je različita on nule**
 - Žakarova sličnost dobija se **deljenjem Hamingovih težina rezultujućih vektora**
 - **Vrednost** Žakarove sličnosti između [0, 1]
 - **0** – **potpuno različiti** vektori, odnosno ne dele ni jednu dimenziju različitu od nule
 - Što je vrednost **veća od nule**, to su vektori **sličniji**
 - **1** – **identični** vektori, odnosno dele sve dimenzije različite od nule

Metrike rastojanja dva vektora

- **Žakarova sličnost i Žakarovo rastojanje**

- **Žakarovo rastojanje** (engl. *Jaccard Distance*)

- Dobija se **oduzimanjem Žakarove sličnosti od jedinice**:

$$d(a, b) = 1 - J(a, b) = \frac{|A \cup B| - |A \cap B|}{|A \cup B|}$$

[1 0 0 1 0 0 0 0 0 1]

AND

[1 1 0 1 0 0 0 1 0 1]

[1 0 0 1 1 0 0 0 1 1]

OR

[1 1 0 1 1 0 0 1 1 1]

$$J(a, b) = \frac{3}{7} \approx 0,43$$

$$d(a, b) = 1 - \frac{3}{7} \approx 0,57$$

- **Vrednost** Žakarovog rastojanja između [0, 1]

- **0** – **identični** vektori, odnosno dele sve dimenzije različite od nule
 - Što je vrednost **veća od nule**, to su vektori **više različiti**, odnosno više udaljeni
 - **1** – **potpuno različiti** vektori, odnosno ne dele ni jednu dimenziju različitu od nule

- **Žakarova sličnost** pokazuje koliko su **slična dva vektora**, a **Žakarovo rastojanje** koliko su **dva vektora različita, odnosno udaljena**

- Žakarovo rastojanje se **jednostavno i brzo izračunava, ne zahteva velike resurse**

Poređenje metrika rastojanja dva vektora

- **Hamingovo i Žakarovo rastojanje:**

- Primarna razlika Hamingovog i Žakarovog rastojanja jeste **kako se posmatraju nule, odnosno odstustvo neke karakteristike**
- **Hamingovo rastojanje** tretira nule i jedinice **ravnopravno**, dok **Žakarovo rastojanje ignoriše** pozicije u kojima oba vektora imaju vrednost nula – gde oba vektora imaju odsustvo karakteristike
- U slučaju **standardnih jediničnih vektora ili retkih vektora** (npr. gde je svaka reč rečnika reprezentovana jednom dimenzijom, a dobijeni tekst ima samo nekoliko reči), čak i u slučaju da dva teksta nemaju ni jednu zajedničku reč, **Hamingovo rastojanje će biti malo**, jer se oba teksta razlikuju samo npr. po 10 od 10.000 reči – na svim ostalim rečima su zajedničke nule, što daje pogrešan utisak sličnosti dva teksta
 - **Žakarovo rastojanje posmatra samo poklapanje reči koje postoje** u oba teksta – ako nema zajedničkih reči, Žakarovo rastojanje je najveće

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

$$d_h(a, b) = 4$$

$$d_j(a, b) = 1 - \frac{0}{4} = 1$$

Sadržaj

- Vektori
- Ugrađeni vektori
- Modeli ugrađivanja
- Metrike rastojanja dva vektora
- Vektorska baza podataka
- Pretrage kNN i ANN
- Hibridna pretraga
- Primena vektorske baze podataka u arhitekturi RAG
- Poređenje relacije i vektorske baze podataka
- Literatura

Vektorska baza podataka

- **Vektorska baza podataka** predstavlja tip baze podataka koja se koristi se za skladištenje i upravljanje podacima u formi **ugrađenih vektora**
 - Može da skladišti **strukturirane i nestrukturirane podatke** u formi **ugrađenih vektora**
- **Ugrađeni vektor** predstavlja **visokodimenzionalnu numeričku reprezentaciju podataka**
 - Predstavlja podatke u **latentnom prostoru** vektorske baze podataka koji može da sadrži **stotine ili hiljade dimenzija**
- Usled čuvanja ugrađenih vektora, način **optimizacije** rada vektorske baze podataka i **postavljanja upita** je drugačiji u odnosu na relacionu bazu podataka
 - U relacionoj bazi podataka vrši se pretraga po redovima i kolonama na osnovu **tačno zadatih uslova** koji moraju biti ispunjeni
 - U vektorskoj bazi podataka vrši se pretraga ugrađenih vektora koji su **najsličniji postavljenom upitu** na osnovu određene **metrike rastojanja dva vektora**

Vektorska baza podataka

- Vektorska baza podataka služi primarno za **pretragu sličnosti između ugrađenih vektora**
 - **Pretraga sličnosti** (engl. *Similarity Search*) kod vektorskih baza podataka može biti **brzo izvršena** nad **velikim skupom ugrađenih vektora**
 - Sadržaji poput teksta, slika i zvuka transformišu se u ugrađene vektore kako bi bilo moguće **uporediti ih i pronaći slične sadržaje**
- Vektorska baza podataka **nije isto** što i **vektorski indeks ili biblioteka za pretragu vektora** korišćenih u različitim sistemima
 - Npr. *FAISS (Facebook AI Similarity Search)* nije vektorska baza podataka, već biblioteka za pretragu vektora
 - Iako je indekse vektora moguće skladištiti u masovnoj memoriji, postoje ograničenja poput nemogućnosti filtriranja vektora prilikom pretrage usled nepostojanja meta-podataka

Sistem za upravljanje vektorskom bazom podataka

- Vektorskom bazom podataka upravlja **sistem za upravljanje bazom podataka** koji pruža mogućnosti poput:
 - Pružanja **interfejsa** za pristup vektorskoj bazi podataka
 - **Skladištenja ugrađenih vektora i različitih atributa**
 - **Pretrage ugrađenih vektora** primenom različitih metoda
 - Kreiranja **indeksa** nad ugrađenim vektorima ili atributima
 - **Višekorisničkog** režima rada i **kontrole pristupa** vektorskoj bazi podataka
 - **Praćenja performansi** rada vektorske baze podataka
 - **Horizontalnog i vertikalnog skaliranja**
 - Kreiranja **rezervnih kopija** podataka (engl. *Backup*)
 - **Oporavka** vektorske baze podataka (engl. *Recovery*)

Skladištenje podataka u vektorskoj bazi podataka

- **Kolekcija** predstavlja **skup ugrađenih vektora i njihovih atributa**
 - Može da pripada unapred definisanoj **šemi**
- Ugrađeni vektor može da bude opremljen dodatnim **atributima**
 - Koriste se takođe termini **polja** ili **meta-podataka** vektora
- **Atributi** mogu da imaju svoje **meta-podatke**, odnosno podatke koji ih opisuju (npr. tip podatka, dužina)
 - **Ugrađeni vektor** takođe može da se posmatra kao **atribut** u okviru jedne kolekcije, predstavljen nizom numeričkih vrednosti

Kolekcija Filmovi				
Identifikator	Naziv	Godina	Opis	Opis-Ugrađeni-Vektor
1	Film 1	2023	Opis 1	[0.284, 0.880, -0.184, ...]
2	Film 2	2021	Opis 2	[0.764, -0.337, 0.107, ...]
3	Film 3	2022	Opis 3	[0.765, 0.280, -0.992, ...]

Skladištenje podataka u vektorskoj bazi podataka

- **Horizontalno skaliranje** podataka moguće je ostvariti kroz **fizičko deljenje skupa podataka** na različitim serverima
 - Npr. na osnovu opsega vrednosti ključa (engl. *Range Partitioning*) ili liste vrednosti (engl. *List Partitioning*)
- **Fragmentacija** vektorske baze podataka (engl. *Sharding*) predstavlja **raspodelu disjunktnih skupova slogova** na više **različitih servera**
 - Omogućava **paralelno i brže izvršavanje upita** i smanjuje opterećenje pojedinačnih servera
 - **Upit** mora da se pošalje svim serverima, nakon čega je potrebno objediniti rezultate upita
- **Replikacija** vektorske baze podataka (engl. *Replication*) predstavlja **kreiranje kopija skupova podataka** i njihovu raspodelu na više **različitih servera**
 - U slučaju **ispada jednog servera**, ostali serveri rade i mogu da pošalju odgovor na upit
 - Obezbeđuje **stabilnost sistema i otpornost na ispade**
- **Klaster** u vektorskoj bazi podataka može da **grupiše slične slogove**, odnosno one ugrađene vektore koji su **blizu u latentnom prostoru**
 - Na taj način može da pruži bržu pretragu
- Moguće je definisati i različita **ograničenja** na nivou klastera, kolekcije, particije ili atributa

Raznolikost primene terminologije i sintakse jezika

- Kod većine **nerelacionih baza podataka**, prelaskom sa jednog sistema za upravljanje bazom podataka na drugi, moguće je primetiti **različite koncepte i tehnologije** koje podržavaju, kao i **različitu primenu terminologije** za iste koncepte
- Takođe, **razlikuju se i jezici** koje koriste za pristup podacima i upravljanje sistemom za upravljanje bazom podataka
- Primena **različitih koncepata, tehnologije, terminologije i sintakse jezika** je **posebno izraženo** kod **vektorskih sistema za upravljanje bazom podataka**

Raznolikost primene terminologije

	Pinecone	Milvus	Chroma	Weaviate	Qdrant
Schema	-	Schema	-	Schema	-
Collection	Index	Collection	Collection	Collection	Collection
Limit	Meta-data	Limit, Constraint	Meta-data	Meta-data	Meta-data
Field	Meta-data	Field	Document	Property	Payload
Entity	Meta-data	Entity	-	Data object	Point
Embedding	Vector	Field	Embedding	Vector	Vector
Partition	Namespace	Partition	-	-	-
Shard	Pod	Shard	-	Shard	Shard
Cluster	Cluster	Cluster	Cluster	Cluster	Cluster
Vector index	Pod-index	Vector index	Index	Vector index	Vector index
Scalar index	Pod-index	Scalar index	-	Inverted index	Payload index

Izvor: Elena Akik, Marko Vještica, Vladimir Dimitrieski, Slavica Kordić, Sonja Ristić, Towards a Model-Driven Approach to Enable Uniform Access to Vector Databases, Communications in Computer and Information Science (CCIS), Vol. 2186, pp. 225-237, Springer, 2024.

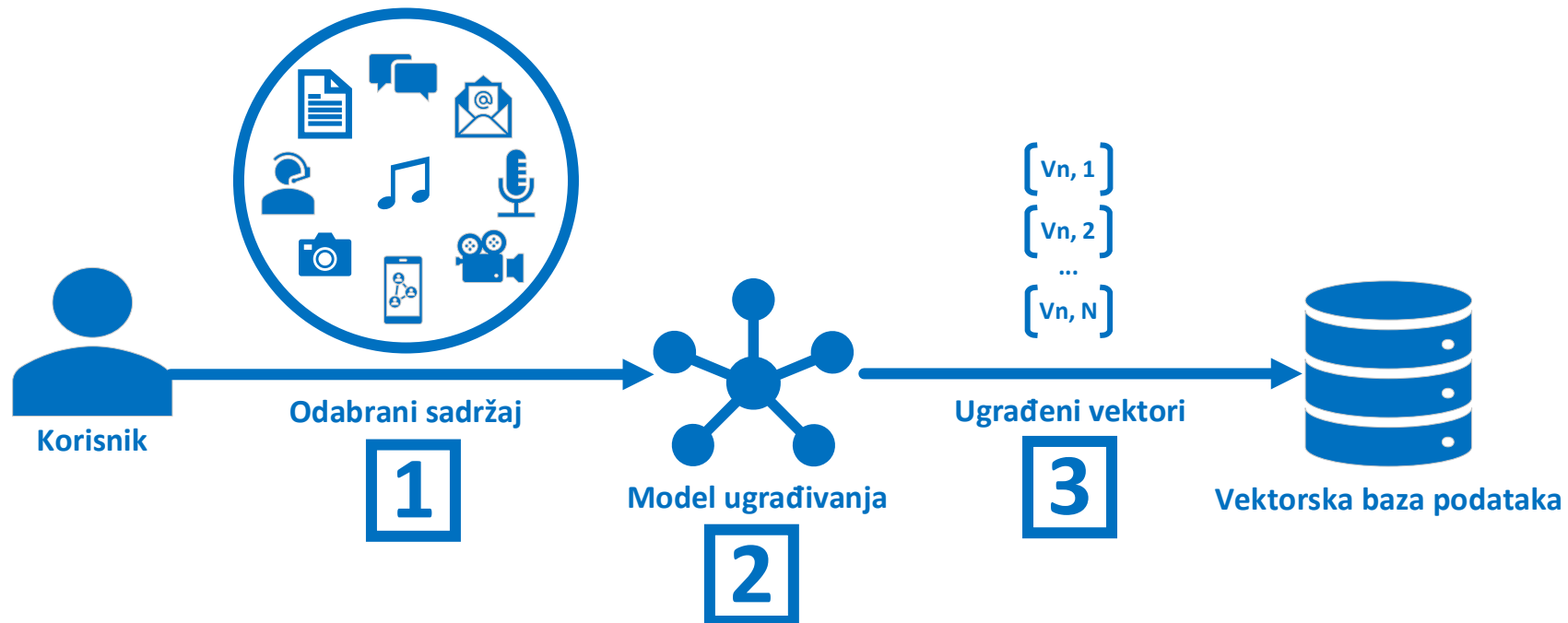
Raznolikost sintakse jezika

Pinecone (RESTful API DSL)		Chroma (JSON-like DSL)	
1	<code>{"collection": "ScienceCourses",</code>	1	<code>{"db": "ScienceCourses",</code>
2	<code> "query": {</code>	2	<code> "action": "query",</code>
3	<code> "select": ["courseName", "courseDescription", "creditHours"],</code>	3	<code> "vector": {"top_k": 10, "metric": "cosine", "range": 0.8, "radius": 0.9},</code>
4	<code> "vector": {"top_k": 10, "metric": "cosine", "range": 0.8, "radius": 0.9},</code>	4	<code> "fields": ["courseName", "courseDescription", "creditHours"],</code>
5	<code> "filter": {</code>	5	<code> "conditions": {</code>
6	<code> "\$and": [</code>	6	<code> "courseName": {"like": "Intro%"},</code>
7	<code> {"courseName": {"\$regex": "Întro"}},</code>	7	<code> "courseId": {"gt": 101},</code>
8	<code> {"courseId": {"\$gt": 101}},</code>	8	<code> "creditHours": {"lte": 4}}}</code>
9	<code> {"creditHours": {"\$lte": 4}}}]}</code>		
Milvus (RESTful API DSL)		DeepLake (TQL)	
1	<code>res = client.search(</code>	1	<code>data = deeplake.load("ScienceCourses")</code>
2	<code> collection_name="ScienceCourses",</code>	2	<code>result = data.query(</code>
3	<code> limit=10,</code>	3	<code> select = ["courseName", "courseDescription", "creditHours"],</code>
4	<code> output_fields=["courseName", "courseDescription", "creditHours"],</code>	4	<code> vector = {"top_k": 10, "metric": "cosine", "range": 0.8, "radius": 0.9},</code>
5	<code> filter='courseName like "Intro%" and courseId > 101 and creditHours <= 4',</code>	5	<code> where = {</code>
6	<code> search_params={</code>	6	<code> "courseName": lambda name: name.startswith("Intro"),</code>
7	<code> "metric_type": "COSINE",</code>	7	<code> "courseId": lambda cid: cid > 101,</code>
8	<code> "params": {"radius": 0.8, "range_filter": 0.9}})</code>	8	<code> "creditHours": lambda ch: ch <= 4})</code>

Izvor: Elena Akik, Marko Vještica, Vladimir Dimitrieski, Milan Čeliković, Slavica Kordić, Sonja Ristić, Interacting with Vector Databases by Means of Domain-Specific Language, Open Computer Science, De Gruyter, 2025.

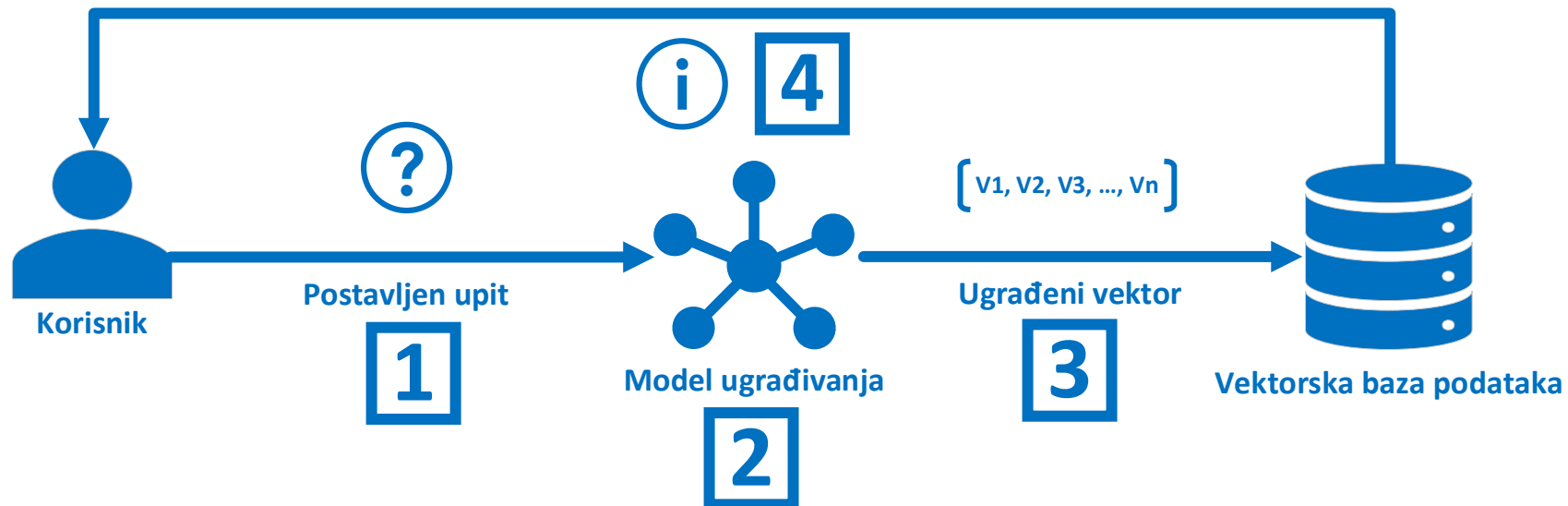
Osnovni koraci upotrebe vektorske baze podataka

- **Osnovni koraci upisa podataka** u vektorsku bazu podataka:
 - **Korak 1** – Slanje sadržaja u odabrani model ugrađivanja
 - **Korak 2** – Transformacija sadržaja u ugrađene vektore primenom modela ugrađivanja
 - **Korak 3** – Skladištenje ugrađenih vektora i ostalih atributa u vektorsku bazu podataka



Osnovni koraci upotrebe vektorske baze podataka

- **Osnovni koraci postavljanja upita i čitanja podataka** u vektorskoj bazi podataka:
 - **Korak 1** – **Postavljanje upita** koji se šalje istom modelu ugrađivanja
 - **Korak 2** – **Transformacija upita u ugrađeni vektor** primenom modela ugrađivanja
 - **Korak 3** – **Pretraga ugrađenih vektora sadržaja sličnih ugrađenom vektoru upita** u okviru vektorske baze podataka upotrebom odabrane metrike rastojanja vektora
 - **Korak 4** – **Slanje rezultata upita** korisniku u formi originalnog sadržaja



Osnovni koraci upotrebe vektorske baze podataka

- Prilikom skladištenja ugrađenog vektora sadržaja u vektorsku bazu podataka, skladišti se i **originalni sadržaj ili referenca ka originalnom sadržaju**
- Transformacija upita u ugrađeni vektor vrši se **istim modelom ugrađivanja** koji je korišćen za kreiranje ugrađenih vektora sadržaja
- Pretraga ugrađenih vektora vrši se odabranom **metrikom rastojanja dva vektora**, ili **pretragom indeksa** ukoliko su kreirani
- Nakon pronalaska ugrađenih vektora sadržaja koji odgovaraju upitu, korisniku se vraćaju njihovi **originalni sadržaji**

Vektorske baze podataka

- **Prednosti:**

- Efikasna pretraga **sličnih podataka**
- Moguća **analiza podataka u realnom vremenu** usled velike brzine izvršavanja upita
 - Kreiranjem **indeksa** moguće je brzo izvršiti upit i nad velikim skupom podataka
 - Npr. dinamički sistem preporuke ili detekcija prevara
- Jednostavno **upravljanje visokodimenzionalnim podacima**
- Mogućnost **horizontalnog skaliranja**, raspodeljujući aktivnosti na više servera
- Mogućnost **integracije sa modelima mašinskog učenja** (npr. velikim jezičkim modelima)
 - Služi kao **trajna memorija** za modele mašinskog učenja

- **Nedostaci:**

- Nisu predviđene za **izvršavanje kompleksnih upita u analitičke svrhe**
- Nisu jednostavne za **razumevanje** poput relacionih baza podataka
- Potrebno često **sprovoditi testiranje** performansi izvršavanja upita i preciznosti rezultata

Primena vektorskih baza podataka

- **Primena vektorskih baza podataka:**
 - **Pretraga i analiza teksta**
 - Npr. pretraga sličnog teksta ili dokumenata; analiza značenja tekstualnog sadržaja; klasterovanje dokumenata
 - **Dopremanje informacija**
 - Npr. Q&A na osnovu upita na prirodnom jeziku; upit i odgovor ne moraju biti iste modalnosti podataka
 - **Sistemi preporuke**
 - Npr. pronalazak proizvoda sličnih onima koje je korisnik želi (bez da mu je poznat tačan naziv proizvoda) – opisao je proizvod ili priložio sliku; preporuka muzike na osnovu zadate melodije ili ritma
 - **Detekcija anomalija**
 - Npr. u bezbednosnim ili proizvodnim sistemima, identifikujući podatke koji odstupaju od uobičajenih; identifikacija prevara na osnovu odstupanja od uobičajenih podataka
 - **Pretraga multimedija**
 - Npr. pretraga srodnih slika i audio zapisa; prepoznavanje lica
 - **Veštačka inteligencija**
 - Npr. vektorska baza podataka može da poboljša sposobnosti veštačke inteligencije time što služi kao trajna memorija i dopunjuje znanje

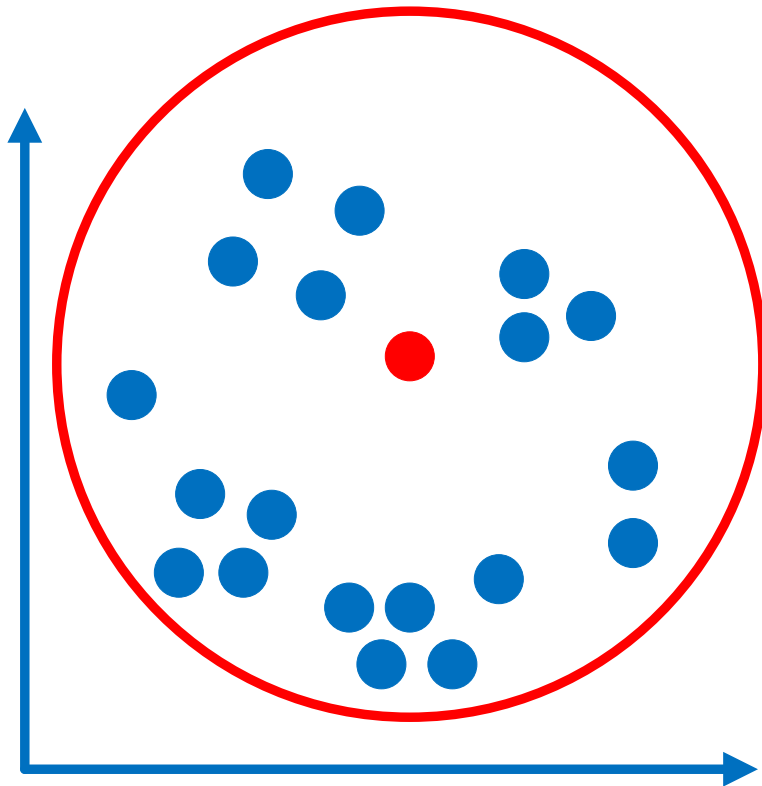
Sadržaj

- Vektori
- Ugrađeni vektori
- Modeli ugrađivanja
- Metrike rastojanja dva vektora
- Vektorska baza podataka
- Pretrage kNN i ANN
- Hibridna pretraga
- Primena vektorske baze podataka u arhitekturi RAG
- Poređenje relacije i vektorske baze podataka
- Literatura

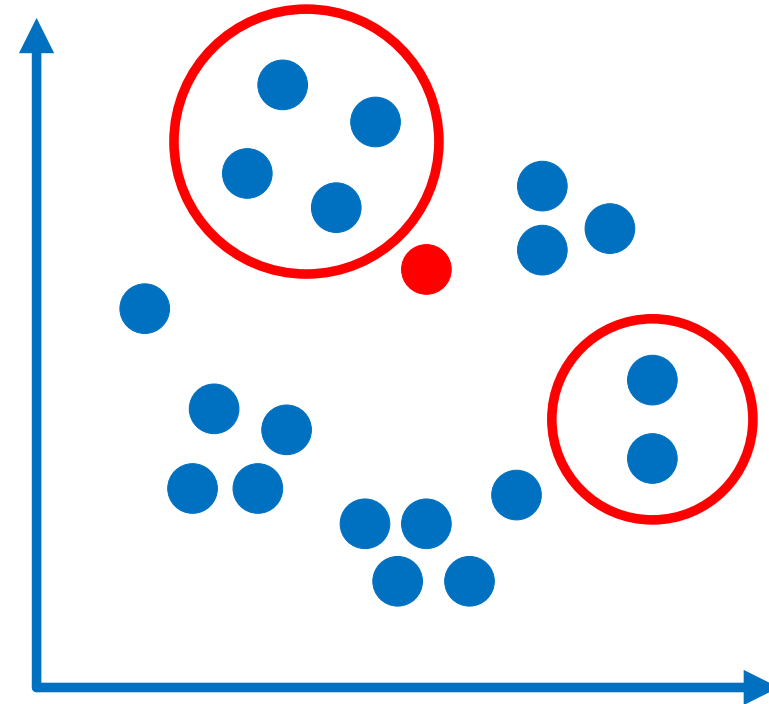
Pretrage podataka u vektorskim bazama podataka

- Vektorska baza podataka primarno koristi **pretragu sličnosti ugrađenih vektora** prilikom izvršavanja upita, ali i druge vrste pretrage
 - **Pretraga sličnosti** (engl. *Similarity Search*), **semantička pretraga** (engl. *Semantic Search*) i **pretraga vektora** (engl. *Vector Search*) često se koriste kao sinonimi
- **Dve osnovne grupe algoritama pretraga vektora** predstavljaju:
 - **Pretraga k najbližih suseda** (engl. *k-Nearest Neighbor (kNN) Search*) – nude precizne odgovore, ali mogu biti procesno zahtevne, čime su manje pogodne za velike skupove podataka
 - **Pretraga približno najbližih suseda** (engl. *Approximate Nearest Neighbor (ANN) Search*) – koriste tehnike indeksiranja i nude balans između preciznosti i brzine odgovora, čime su pogodne za velike skupove podataka
- Savremeni pristupi u vektorskim bazama podataka omogućavaju i **hibridne pretrage**

Poređenje kNN i ANN



Pretraga kNN



Pretraga ANN

Pretraga kNN

- **Pretraga kNN** traži **k ugrađenih vektora sadržaja** najbližijih ugrađenom vektoru upita
 - Nakon što je sadržaj smešten u vektorsku bazu podataka i upit postavljen od strane korisnika, primenjuje se odabrana **metrika rastojanja dva vektora** nad **svim ugrađenim vektorima sadržaja**
 - Koristi se često termin **metrika sličnosti** (engl. *Similarity Metric*) za metriku rastojanja dva vektora
 - Na osnovu izračunatih rastojanja ugrađenih vektora, vrši se njihovo **rangiranje po koeficijentu rastojanja** (ili sličnosti)
 - **Preuzima se k ugrađenih vektora sadržaja** najbližih ugrađenom vektoru upita
 - **Odabir vrednosti k** : mala vrednost k može biti osetljiva na šum, odnosno da **ne vrati željeni sadržaj**, dok velika vrednost k može da vrati **previše nerelevantnog sadržaja**
- Izvršava se **sporije sa porastom broja vektora ili dimenzija latentnog prostora**

Pretraga kNN – odabir metrike sličnosti

- Kako **odabrati** odgovarajuću metriku sličnosti?
 - Uopšteno, ukoliko je poznato, najpogodnije je odabrati metriku sličnosti koja je **korišćena prilikom obuke modela ugrađivanja** koji se koristi za generisanje ugrađenih vektora
 - Tokom obučavanja, model ugrađivanja koristi **funkciju gubitka** (engl. *Loss Function*) koja aktivno računa metriku sličnosti između različitih primera
 - Npr. model generiše vektore za dva slična teksta; izračuna njihovu kosinusnu sličnost (npr. 0,5); cilj je da sličnost bude blizu 1 jer su slični; kako je razlika (gubitak) velika, model menja svoje unutrašnje težine (parametre) isključivo sa ciljem da smanji ugao između ta dva vektora u sledećoj iteraciji
 - **Usklađivanje metrika** (engl. *Metric Alignment*) – metrika kojom se pretražuje sadržaj mora biti ista metriki koju je funkcija gubitka koristila tokom obučavanja modela
 - Odabir metrike sličnosti zavisi takođe od specifičnosti **podataka i domena primene**
 - Postoje **opšte smernice** za odabir metrike sličnosti, ali je **neophodno testirati** različite metrike sličnosti za konkretan slučaj

Pretraga kNN – odabir metrike sličnosti

- **Odabir odgovarajuće metrike sličnosti:**
 - **Manhattan rastojanje**
 - Pogodna za:
 - **Klasterovanje sadržaja** – posebno sa visokodimenzionalnim podacima jer nije osetljiva na velika odstupanja (engl. *Outliers*) pojedinačnih dimenzija
 - **Identifikovanje srodnih slika** – poređenje vrednosti piksela, detekcija objekata ili lica, posebno nad velikim skupovima podataka usled brzog izvršavanja
 - **Detekcija anomalija** – poput prevara na mrežama ili novčanim transakcijama; efikasna jer odstupanje po pojedinačnoj dimenziji ne utiče mnogo, ali odstupanje po više dimenzija će biti detektovano
 - **Prednosti:** pogodna za visokodimenzionalne latentne prostore usled jednostavnosti računanja rastojanja; nije osetljiva na velika odstupanja pojedinačnih dimenzija
 - **Nedostaci:** ne predstavlja najkraće rastojanje između dve tačke; manje precizna od Euklidskog rastojanja

Pretraga kNN – odabir metrike sličnosti

- **Odabir odgovarajuće metrike sličnosti:**
 - **Euklidsko rastojanje**
 - Slično kao i Menhetn rastojanje, pogodna za **klasterovanje sadržaja i identifikovanje srodnih slika**
 - **Prednosti:** nudi najkraće linearno rastojanje između dva vektora
 - **Nedostaci:** zahtevnije izračunati u odnosu na Menhetn rastojanje; osetljiva na velika odstupanja pojedinačnih dimenzija; osetljiva na visokodimenzionalne ugrađene vektore kada se više malih razlika po dimenzijama akumulira
 - **Čebiševljevo rastojanje**
 - Pogodna za:
 - **Detekciju anomalija** – kada je najveće rastojanje bilo koje dimenzije bitnije nego celokupno rastojanje
 - **Filtriranje podataka na osnovu praga** – može brzo da eliminiše vektore koji se mnogo razlikuju po jednoj dimenziji na osnovu zadatog praga
 - **Prednosti:** brzo izvršavanje i detektovanje pojedinačnog velikog odstupanja
 - **Nedostaci:** nije pogodna za pretragu sličnosti (npr. tekst) gde je rastojanje na osnovu svih dimenzija važnije

Pretraga kNN – odabir metrike sličnosti

- **Odabir odgovarajuće metrike sličnosti:**
 - **Skalarni proizvod**
 - Pogodna za:
 - Poređenje podataka koji **nisu normalizovani** ili kada je **bitan intenzitet i ugao vektora** (npr. poređenje slika)
 - **Sistemi preporuka** – poređenje karakteristika proizvoda ili muzike
 - **Prednosti:** brzo izvršavanje, jednostavna implementacija
 - **Nedostaci:** osetljiva na **intenzitet vektora** – ukoliko se vektori značajno razlikuju po intenzitetu, moguće je da određivanje rastojanja ne bude precizno
 - **Kosinusno rastojanje**
 - Pogodna za:
 - Kada su podaci **normalizovani** ili kada **intenzitet vektora nije bitan**
 - **Analizu teksta** – meri **sličnost između teksta** nezavisno od intenziteta vektora, što je pogodno ako su dva teksta udaljena po intenzitetu, ali je ugao između njih mali (npr. ista reč se pojavljuje u dva teksta različit broj puta usled različitih dužina teksta, tada će postojati razlika u intenzitetu vektora, ali ugao može biti mali)
 - **Identifikovanje srodnih slika** – kada intenzitet osvetljenja piksela nije veoma relevantan
 - **Prednosti:** pogodna je za poređenje teksta različite dužine; osetljiva na ugao, ali ne i na intenzitet
 - **Nedostaci:** nula vektori su nedefinisani

Pretraga kNN – odabir metrike sličnosti

- **Odabir odgovarajuće metrike sličnosti:**
 - **Hamingovo rastojanje**
 - Pogodna za:
 - **Detekciju anomalija** – greške u komunikaciji ili kriptografiji; odstupanje po bitovima
 - **Sistemi preporuke** – meri sličnost binarnih ili kategorijalnih podataka
 - **Prednosti:** brzo izvršavanje; rad sa binarnim vektorima za čije skladištenje nije potrebno puno memorijskog prostora
 - **Nedostaci:** nije pogodna za ugrađene vektore jer posmatra samo da li su vrednosti po dimenzijama različite; nije pogodna za retke vektore zbog velikog broja nula – navelo bi da su dva vektora slična iako su različiti
 - Primena Hamingovog rastojanja sa binarnim vektorima:
 - Moguće je iskoristiti **model ugrađivanja** koji generiše **binarne vektore**
 - Moguće je **ugrađeni vektor transformisati u binarni vektor**
 - Npr. vektorska baza podataka *Weaviate* koristi metod *Binary Passage Retrieval (BPR)* da transformiše ugrađeni vektor u binarni vektor uz zadržavanje semantičke strukture

Pretraga kNN – odabir metrike sličnosti

- **Odabir odgovarajuće metrike sličnosti:**
 - **Žakarov rastojanje**
 - Pogodna za:
 - **Analizu teksta i detekciju plagijarizma** – poređenje dokumenata na osnovu preklapanja reči
 - **Sisteme preporuke** – merenje sličnosti korisnika na osnovu interesovanja ili zajedničkih osobina
 - **Prednosti:** ne računa zajedničke nule dva vektora kao preklapanje, što sprečava da velik broj zajedničkih nula "razvodni" metriku i prikaže različite stvari kao slične; pronalazi sličnost samo na osnovu prisutnih atributa, ignorišući attribute koji nedostaju u oba vektora
 - **Nedostaci:** nije pogodna za ugrađene vektore jer posmatra isključivo diskretne vrednosti

Pretraga kNN – odabir metrike sličnosti

- **Odabir odgovarajuće metrike sličnosti:**

- Potrebno je **testirati različite metrike sličnosti** nad **konkretnim podacima**
 - Pripremiti **skup podataka i skup upita za testiranje**
 - Izvršiti testove u kojima se **jedino menjaju metrike sličnosti**, dok su **sve ostale promenljive konstantne** (npr. ne menja se model ugrađivanja, ne dodaju se indeksi, ne menjaju se upiti)
 - Dakle, testirati upite uz primenu različitih metrika sličnosti, ali nad **identičnim skupom podataka**
 - **Pratiti performanse** poput vremena izvršavanja upita (engl. *Latency*), koliko relevantnih rezultata je dopremljeno (engl. *Recall*), zauzeća računarskih resursa
- Nekada je potrebno **odabrati između preciznosti rezultata upita i brzine izvršavanja upita**
- Moguće je pripremiti i **različite test scenarije**, poput različite **distribucije podataka**, **različite veličine skupa** test podataka, primena **normalizovanih i nenormalizovanih** ugrađenih vektora
- Moguće primeniti različite **tehnike indeksiranja** koje mogu da ubrzaju izvršavanje upita, ali smanje preciznost rezultata upita

Pretraga kNN – filtriranje podataka

- **Rezultat** postavljenog upita predstavlja **listu ugrađenih vektora sadržaja** uređenih po koeficijentu sličnosti od najvećeg ka najmanjem, zajedno sa originalnim sadržajem i dodatnim atributima, u zavisnosti od postavljenog upita
 - Moguće je **pristupiti originalnom sadržaju** ugrađenih vektora direktno ili posredstvom reference, u zavisnosti od načina čuvanja originalnog sadržaja
- **Broj ugrađenih vektora sadržaja** koji se dobavlja zavisi od postavljenog upita
 - Moguće je definisati **vrednost parametra k**
 - Biće dobavljeno k najsličnijih ugrađenih vektora
 - Moguće je definisati **opseg koeficijenta sličnosti**
 - Biće dobavljeni ugrađeni vektori čiji koeficijent sličnosti spada pod zadati opseg vrednosti

Pretraga kNN – filtriranje podataka

- Moguće je da postoje **dodatni uslovi** u upitu koji se odnose na **attribute kolekcije** (meta-podatke ugrađenih vektora) **ili koeficijente sličnosti**
- **Filtriranje ugrađenih vektora** služi da ukloni nerelevantan sadržaj pre nego što bude vraćen kao rezultat pretrage
- Postoje **dva pristupa** filtriranja ugrađenih vektora:
 - Pre-filtriranje (engl. *Pre-Filtering* ili *Pre-Processing*)
 - Uklanjanje na **nivou meta-podataka**
 - Post-filtriranje (engl. *Post-Filtering* ili *Post-Processing*)
 - Uklanjanje **velikog odstupanja** (engl. Outliers) vektorske pretrage

Pretraga kNN – filtriranje podataka

• Pre-filtriranje

- Filtriranje na nivou meta-podataka izvršava se **pre pretrage sličnosti** ugrađenih vektora, koristeći **vrednosti meta-podataka**
 - **Sužava prostor pretrage sličnosti** ugrađenih vektora, što može ubrzati izvršavanje upita
 - Omogućava pretragu na osnovu **tačnog poklapanja** sa vrednostima meta-podataka
 - Npr. korisnički upit: „Koji dokumentarci postoje iz 2023. godine o računarstvu?“, vektorska pretraga bi mogla da vrati semantički slične filmove iz 2022. godine usled visokog koeficijenta sličnosti, dok bi pre-filtriranje najpre trebalo da suzi prostor pretrage filmova samo za 2023. godinu
- Pre-filtriranje je takođe način za primenu **kontrole pristupa zasnovane na ulogama** (engl. *Role-Based Access Control* (RBAC)).
 - Ugrađeni vektori su filtrirani tako da se pretražuju samo oni koje određeni **korisnik ima dozvolu da čita**
- U slučaju vrlo **restriktivnih uslova** (eliminisanja velikog broja ugrađenih vektora), postoji mogućnost da rezultujuća lista ugrađenih vektora ne predstavlja sadržaj sličan upitu (izostanak relevantnog sadržaja)
- Moguće je iskoristiti **indeksiranje meta-podataka** kako bi se proces filtriranja ubrzao

Pretraga kNN – filtriranje podataka

- **Post-filtriranje**

- Filtriranje velikog odstupanja (engl. *outliers*) vektorske pretrage izvršava se **nakon pretrage sličnosti** ugrađenih vektora, analizom **koeficijenata sličnosti**
 - Inicijalna pretraga sličnosti vrši se nad **celim skupom** ugrađenih vektora
- U slučaju vrlo **restriktivnih uslova**, postoji mogućnost da budu **uklonjeni svi ugrađeni vektori** dobijeni nakon pretrage sličnosti
- Pretraga sličnosti vrši se i nad ugrađenim vektorima koji će **biti uklonjeni nakon pretrage**, što narušava performanse izvršavanja upita
- Glavni **pristupi post-filtriranja**:
 - **Pretraga opsega** (engl. *Range Search*): Postavljanje fiksnog praga minimalnog i/ili maksimalnog koeficijenta sličnosti
 - **Dinamički prag** (engl. *Autocut*): Heuristički pristup koji identifikuje nagli pad u vrednostima koeficijenta sličnosti između dva ugrađena vektora i uklanja ostatak ugrađenih vektora
 - **Primena modela mašinskog učenja**: Korišćenje manjeg modela mašinskog učenja za proveru i uklanjanje nerelevantnog sadržaja koji je dopremljen pretragom sličnosti (npr. *LLM-as-a-Judge*)

Pretraga ANN

- **Vreme** potrebno za izvršavanje pretraga *kNN* **raste povećanjem broja ugrađenih vektora u bazi podataka i broja dimenzija latentnog prostora**
- Metrike sličnosti su u pojedinim vektorskim bazama podataka (npr. *Weaviate*), implementirane pomoću **Asemblera** kako bi bila postignuta veća brzina izvršavanja upita
- Vektorske baze podataka mogu da **izvršavaju upit relativno brzo** čak i za velik broj ugrađenih vektora u visokodimenzionalnim latentnim prostorima primenom **tehnika indeksiranja ugrađenih vektora**

Pretraga ANN

- **Pretraga ANN** traži **ugrađene vektore sadržaja približno** najbližijih ugrađenom vektoru upita
 - Prilikom smeštanja sadržaja u vektorsku bazu podataka, vrši se **indeksiranje ugrađenih vektora** i njihovo **preslikavanje na odabranu strukturu podataka**, kako bi se omogućile brže pretrage
 - Nakon postavljanja upita od strane korisnika, **algoritmi ANN određuju delove latentnog prostora** u kojima se potencijalno nalaze najbliži ugrađeni vektori sadržaja
 - Umesto da se meri udaljenost svih ugrađenih vektora sadržaja u odnosu na upit, primenjuje se odabrana **metrika sličnosti** nad **ugrađenim vektorima odabranih regiona**
 - Na osnovu izračunatih rastojanja ugrađenih vektora, vrši se njihovo **rangiranje po koeficijentu sličnosti**
 - **Preuzima se k ugrađenih vektora sadržaja** koji bi **trebalo da su slični** ugrađenom vektoru upita
- Efikasna sa **velikim brojem vektora ili dimenzija latentnog prostora**
 - Međutim, **ne garantuje** da su odabrani ugrađeni vektori zaista najbliži ugrađenom vektoru upita

Pretraga ANN

- Postupak **indeksiranja ugrađenih vektora** predstavlja njihovo **preslikavanje na nove strukture** kako bi se povećala **brzina pretrage**, i/ili smanjio **memorijski prostor** potreban za njihovo skladištenje, uz što bolje **očuvanje preciznosti** odgovora
 - Predstavlja **postupak organizovanja (grupisanja) sličnih ugrađenih vektora**
- **Indeksiranje ugrađenih vektora** moguće je izvršiti pomoću algoritama i tehnika poput:
 - **Heširanja** (engl. *Hashing*)
 - **Nasumične projekcije** (engl. *Random Projection*)
 - **Kvantizacije** (engl. *Quantization*)
 - **Invertovanih datoteka** (engl. *Inverted File*)
 - **Tehnika zasnovanih na grafu** (engl. *Graph-Based Techniques*)

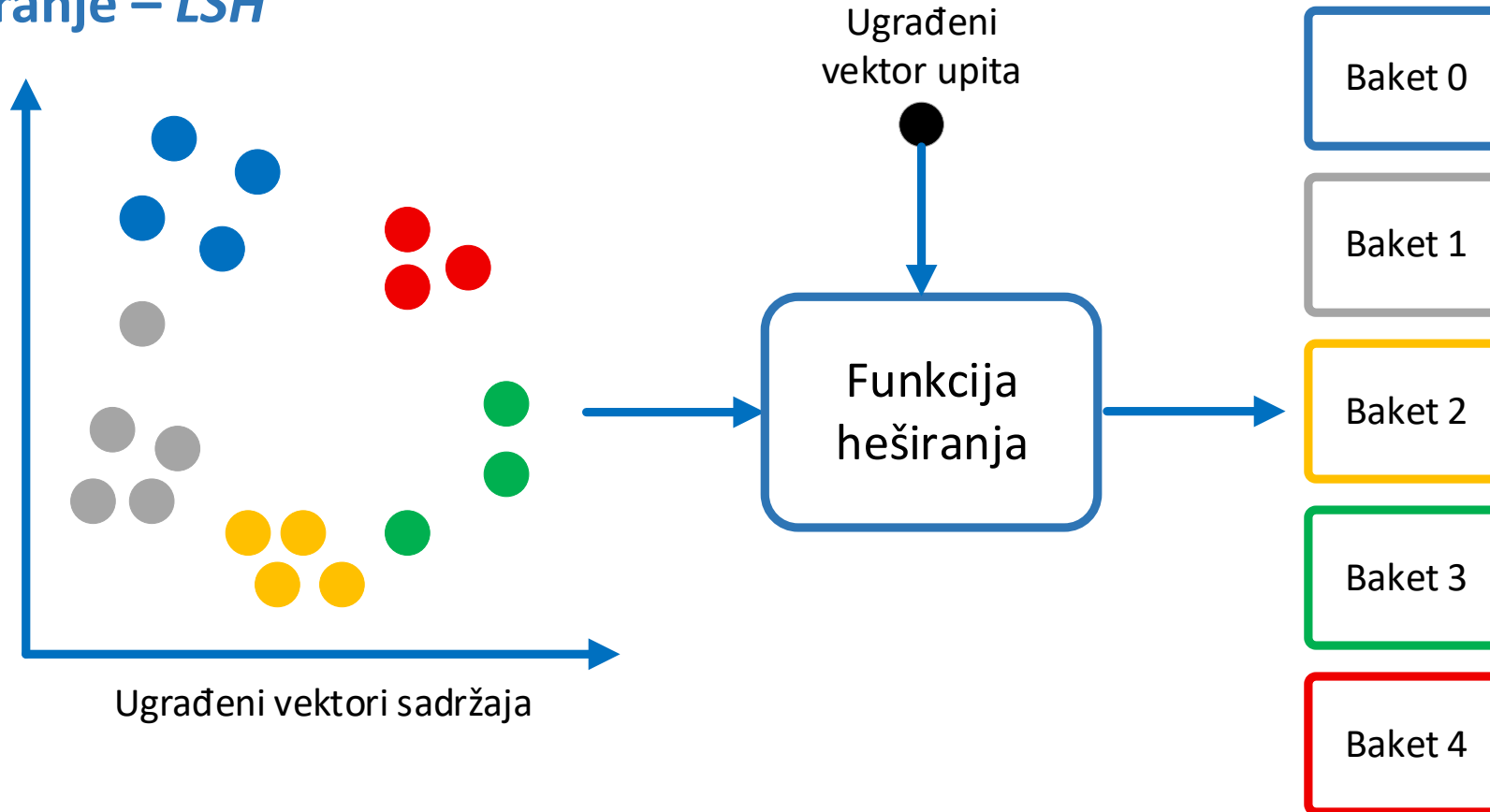
Pretraga ANN – vektorski indeksi

• Heširanje – *LSH*

- Algoritam **heširanja osetljivog na lokalnost** (engl. *Locality-Sensitive Hashing (LSH)*) koristi specifične **funkcije heširanja** projektovane da maksimizuju verovatnoću da **slični ugrađeni vektori dobiju istu heš vrednost** i budu smešteni u isti **heš baket** (tabelu)
 - Za razliku od **uobičajenih funkcija heširanja** koje teže da različite vrednosti, čak iako one bile veoma slične, **smeštaju u različite bakete**
- Prilikom izvršavanja upita, ugrađeni vektor upita se propušta kroz **istu funkciju heširanja** korišćenu za ugrađene vektore sadržaja kako bi bio **preslikan u odgovarajući baket**
 - Zatim se uz pomoć odabrane **metrike sličnosti** vrši poređenje ugrađenog vektora upita isključivo sa **ugrađenim vektorima unutar tog baketa**
 - Omogućeno **brzo izvršavanje upita**, jer postoji znatno manje ugrađenih vektora u heš baketima nego u celoj vektorskoj bazi podataka
- Da bi se smanjila verovatnoća da se neki relevantan ugrađeni vektor sadržaja propusti, odnosno da bi se povećao odziv (engl. *recall*), često se **paralelno pretražuje više nezavisnih heš baketa**
 - **Odziv, odnosno sveobuhvatnost upita** predstavlja **procenat relevantnog sadržaja** koji je dopremljen
 - Predstavlja **odnos između broja dopremljenih relevantnih vektora i ukupnog broja relevantnih vektora** koji postoje u vektorskoj bazi podataka

Pretraga ANN – vektorski indeksi

- Heširanje – *LSH*



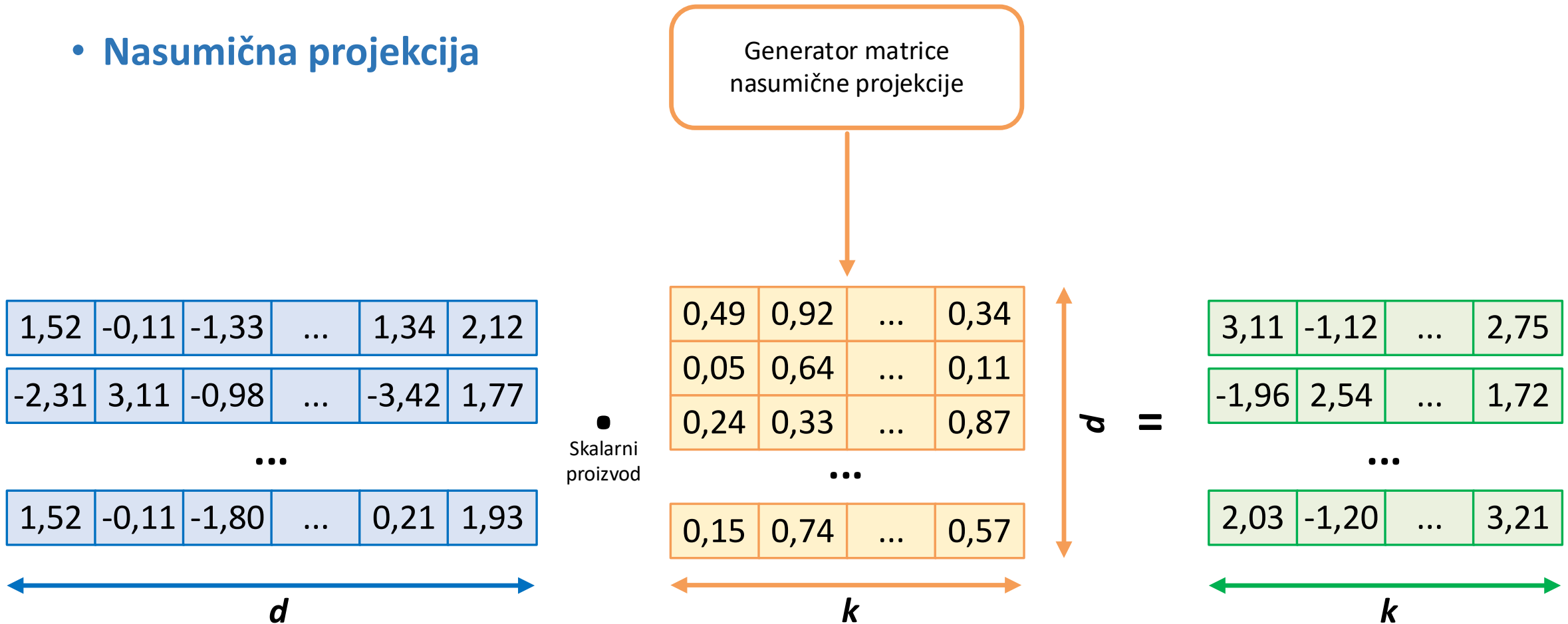
Pretraga ANN – vektorski indeksi

- **Nasumična projekcija**

- Algoritam nasumične projekcije **vrši projekciju** visokodimenzionalnog ugrađenog vektora u ugrađeni vektor **manjih dimenzija** primenom matrice nasumične projekcije (engl. *Random Projection Matrix*)
 - Vrš **smanjenje broja dimenzija** latentnog prostora kako bi se znatno **ubrzalo izvršavanje upita**
- **Matrica nasumične projekcije** ima dimenzije **$d \times k$**
 - **d** predstavlja broj dimenzija originalnih vektora, a **k** predstavlja ciljni (projektovani) broj dimenzija
 - Vrednosti matrice se generišu iz **specifičnih raspodela** (npr. Gausova raspodela), oslanjajući se na lemu Džonson-Lindenštraus (engl. *Johnson-Lindenstrauss lemma*)
- **Lema Džonson-Lindenštraus**
 - Podaci iz **prostora veoma visokih dimenzija** mogu se projektovati u **prostor znatno nižih dimenzija**, a da se pri tome **skoro potpuno očuvaju rastojanja** između svih parova tačaka
- Vrš **množenje ugrađenih vektora i matrice nasumične projekcije**, čime se dobija ugrađeni vektor **manjih dimenzija**, uz **očuvanje sličnosti** (relativnih udaljenosti) između vektora
- Prilikom postavljanja upita, **ista matrica nasumične projekcije** množi se sa ugrađenim vektorom upita, kako bi se dobio ugrađeni vektor manjih dimenzija, a zatim **poredio sa prethodno projektovanim** ugrađenim vektorima sadržaja

Pretraga ANN – vektorski indeksi

- Nasumična projekcija



Pretraga ANN – vektorski indeksi

- **Kvantizacija**

- Kvantizacija vektora predstavlja opštu metodu za **sažimanje vektora u kompaktnije reprezentacije**, čime se optimizuje efikasnost skladištenja vektora i brzina izvršavanja upita
- Predstavlja metodu **kompresije** visokodimenzionalnih ugrađenih vektora na način da **smanjuje opseg vrednosti** za svaku dimenziju
 - Ne vrši se redukcija dimenzionalnosti, kao kod nasumične projekcije, već **redukcija memorijskog prostora i reprezentacije ugrađenih vektora**
 - **Broj dimenzija latentnog prostora ostaje isti**
 - Npr. Decimalna vrednost svake dimenzije koja zauzima 32 b pretvoriti u celobrojnu vrednost od 8 b
 - Originalni vektor od 1536 dimenzija zauzima 6 KB memorije; 1,000,000 vektora zauzima 6 GB memorije
 - Kompresovani vektor zauzima 1,5 KB memorije

Pretraga ANN – vektorski indeksi

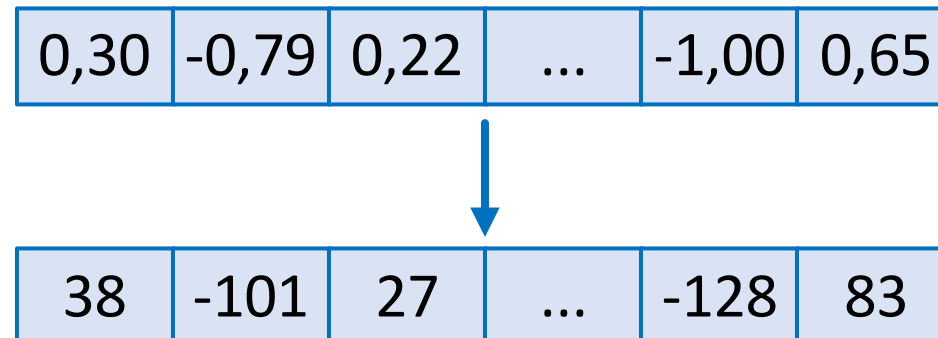
- **Kvantizacija**

- Kvantizacija **ubrzava pretragu vektora** time što pojednostavljuje proračun rastojanja dva vektora
 - **Smanjuje memorijski prostor** potreban za skladištenje ugrađenih vektora
 - Međutim, **rezultati upita su manje precizni** (smanjen odziv)
- Primjenjuje se kada je **brzina izvršavanja upita bitnija od sveobuhvatnosti odgovora**
 - Npr. Dopremanje sličnih slika ili sistemi preporuka
- Postoje **različite tehnike kvantizacije** vektora, poput:
 - **Skalarne kvantizacija** (engl. Scalar Quantization)
 - **Binarne kvantizacija** (engl. Binary Quantization)
 - **Produktne kvantizacija** (engl. Product Quantization)

Pretraga ANN – vektorski indeksi

- **Kvantizacija – SQ**

- **Skalarna kvantizacija** preslikava **decimalne vrednosti dimenzija** (npr. *float32* koji zauzima 32 bita) **u celobrojne vrednosti** (npr. *int8* koji zauzima 8 bita)
 - Npr. Opseg vrednosti za *float32* [-1, 1], dok je opseg vrednosti za *int8* [-128, 127]
- Moguće je **smanjiti zauzeće memorijskog prostora 4 puta**
- Skalarna kvantizacija je pogodna kada je potrebno ubrzati izvršavanje upita i kompresovati podatke **bez velikog gubitka sveobuhvatnosti odgovora**



Pretraga ANN – vektorski indeksi

- **Kvantizacija – BQ**

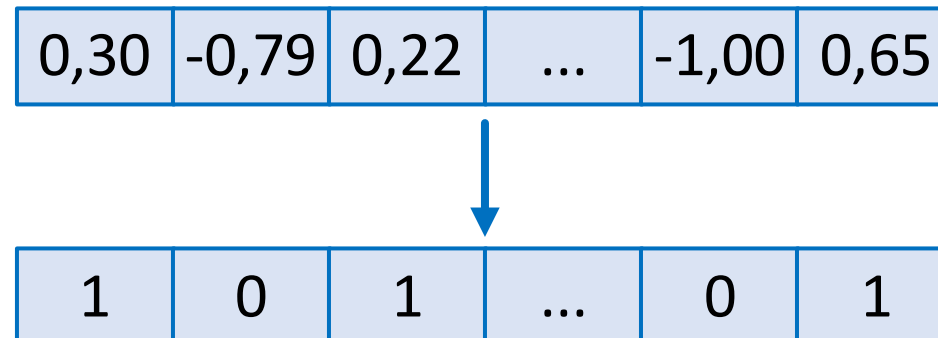
- **Binarna kvantizacija** preslikava **decimalne vrednosti dimenzija u binarne vrednosti**

- Vrednosti veće od 0 preslikavaju se na 1, dok vrednosti manje ili jednake 0, preslikavaju se na 0
- Npr. Opseg vrednosti za *float32* [-1, 1], dok je opseg vrednosti za *boolean* {0, 1}

- Moguće je **smanjiti zauzeće memorijskog prostora 32 puta**

- Binarna kvantizacija pruža **najbolje ubrzanje izvršenja upita** u odnosu na *SQ* i *PQ*, jer omogućava korišćenje **binarnih operacija**, odnosno **Hamingovog i Žakarovog rastojanja**

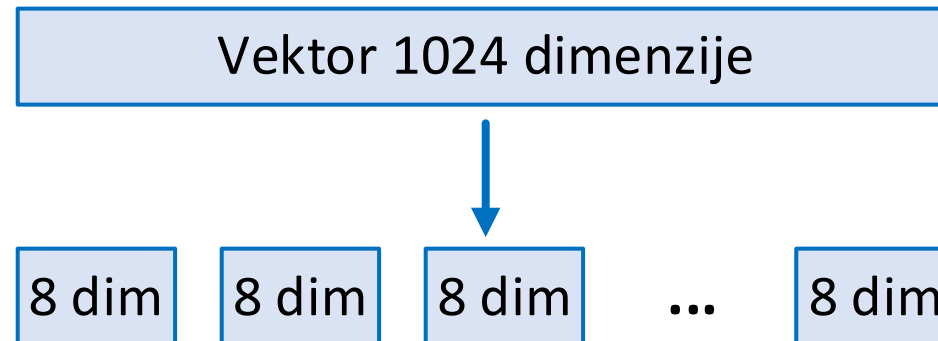
- Međutim dolazi do **smanjenja sveobuhvatnosti odgovora**, stoga je preporuka da se **koristi za dimenzije veće od 1024**, kako bi se smanjio gubitak sveobuhvatnosti



Pretraga ANN – vektorski indeksi

- **Kvantizacija – PQ**

- **Produktna kvantizacija** preslikava **visokodimenzionalni vektor** u manji skup reprezentacija
- Proces kvantizacije započinje **podelom originalnog vektora** u manje **podvektore**
 - Podvektori obuhvataju **različite karakteristike** vektora
- Svaki podvektor pripada **šifarniku** (engl. *Codebook*) koji predstavlja **skup regiona u latentnom prostoru** gde se pojavljuju uobičajeni obrasci

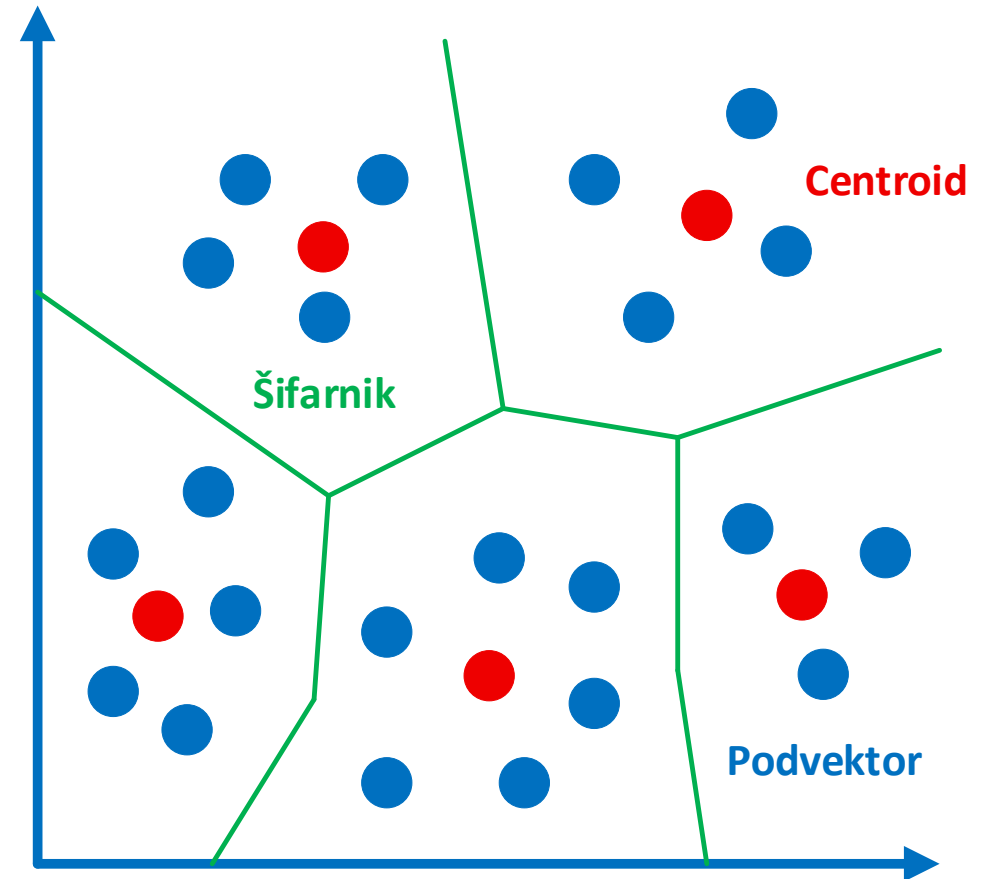


Pretraga ANN – vektorski indeksi

• Kvantizacija – PQ

- Šifarnik je kreiran treniranjem na osnovu postojećih vektora u vektorskoj bazi podataka, koristeći algoritam klasterizacije *k*-srednjih vrednosti (engl. *k-Means Clustering*)

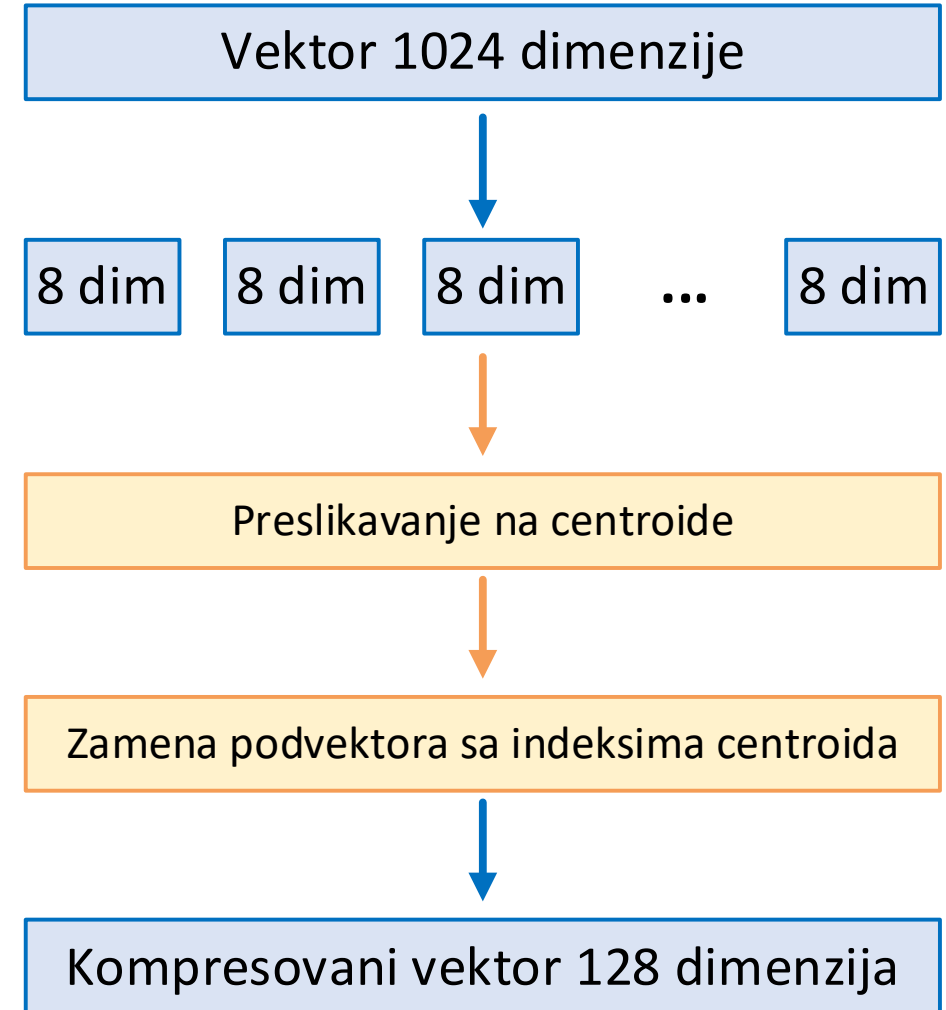
- Svaki region šifarnika određen je težištem klastera, odnosno **centroidom**, koji služi kao **reprezentativna tačka**, odnosno sažetak karakteristika tog regiona
- Na taj način, vrši se **grupisanje svih sličnih podvektora** u jednu tačku
- **Broj centroida** zavisi od implementacije SUBP
 - Npr. *Qdrant* uvek kreira 256 centroida, kao najveći broj koji stane u 1 B, stoga je potreban 1 B za skladištenje indeksa centroida



Pretraga ANN – vektorski indeksi

• Kvantizacija – PQ

- Svaki **podvektor** originalnog vektora preslikan je na **indeks centroida**
 - Odnosno, **kompresovani vektor skladišti indekse centroida** za svaki podvektor
 - Npr. Originalni vektor od 1024 dimenzije zauzima 4096 B memorije, a nakon primene PQ, od 128 podvektora dobija se 128 dimenzija, gde svaka dimenzija reprezentuje indeks centroida veličine 1 B
 - **Smanjenje memorijskog prostora od 32 puta**, a u pojedinim konfiguracijama moguće i do 64 puta
 - Moguće je **odabrati broj podvektora i veličinu šifarnika**
- Može da obezbedi **najbolju uštedu memorijskog prostora**
 - Međutim, može da dovede do **značajnijeg pada u sveobuhvatnosti odgovora**, kao i u **brzini izvršavanja upita**



Pretraga ANN – vektorski indeksi

- **Kvantizacija – PQ**

- **Broj centroida u šifarniku** predstavlja kompromis između **preciznosti reprezentacije i potrebnih računarskih resursa** za pretraživanje šifarnika
 - Što je **više centroida u šifarniku**, to je **reprezentacija podvektora preciznija**, ali je računarski zahtevnija pretraga šifarnika
- **Postupak izvršavanja upita:**
 - **Vektor postavljenog upita deli se na podvektore** za koje se zatim računaju rastojanja od svih centroida
 - **Rastojanja** svih podvektora upita od svih centroida čuvaju se u **tabeli pretraživanja** (engl. *Lookup Table*)
 - Za **svaki kvantizovani vektor sadržaja** proverava se **rastojanje od vektora upita**
 - Kako su rastojanja između podvektora upita i centroida već izračunata, **ne računa se rastojanje između vektora**
 - Nego se za **svaki identifikator centroida** kvantizovanog vektora sadržaja **proveri njegovo rastojanje od podvektora upita u tabeli pretraživanja**, nakon čega se **koeficijenti sličnosti podvektora saberu**

- Često se tehnike kvantizacije koriste **u kombinaciji sa drugim ANN pristupima**

Pretraga ANN – vektorski indeksi

• Kvantizacija – Poređenje

- Primer za ugrađene vektore od 1024 dimenzije tipa *float32*
- Poređenje u odnosu na upotrebu vektora bez kvantizacije

Tehnika	Sveobuhvatnost	Brzina	Kompresija	Karakteristike	Primena
Skalarna	0,99	Do 2x veća	4x	Minimalni gubitak sveobuhvatnosti odgovora na upit	Uobičajen izbor za većinu primena; dobar balans sveobuhvatnosti, brzine i kompresije
Binarna	0,95 (u slučaju velikog broja dimenzija)	Do 40x veća	32x	Najbrža za izvršavanje upita i memorijski efikasna	Primena samo nad testiranim modelima ugrađivanja koji nude latentni prostor velikog broja dimenzija; kada su brzina i memorijska efikasnost bitni
Produktna	0,70	2x manja	Do 64x	Najveća mogućnost kompresije podataka	Kada je minimizovanje zauzeća memorije prioritet; prihvatljivo kada se toleriše gubitak preciznosti i sporije kreiranje indeksa

Izvor: Dokumentacija vektorskog SUBP *Qdrant*

Pretraga ANN – vektorski indeksi

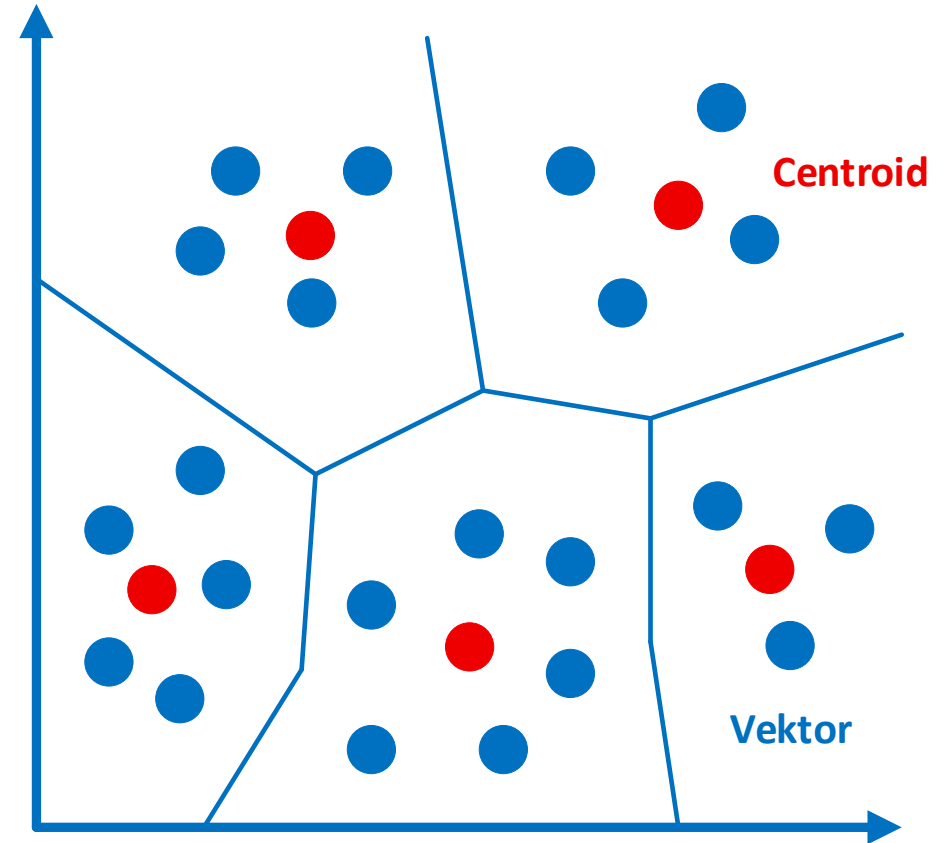
- **Invertovana datoteka**

- Algoritam **invertovane datoteke (IVF)** deli celokupan vektorski prostor na određeni broj regiona koristeći **tehniku klasterizovanja**
- Koristi se algoritam **klasterizacije k -srednjih vrednosti** kako bi se identifikovali regioni i centroidi
 - Svaki **region** reprezentovan je svojim vektorom **centroidom** koji predstavlja prosečni vektor u regionu
 - Svaki **ugrađeni vektor sadržaja** dodeljen je na **listu centroida kojem je prostorno najbliži**
 - Time se stvara struktura **obrnutog indeksa** (engl. *Inverted Index*) gde svaki **ključ predstavlja jedan centroid**, a njegova **vrednost je lista ugrađenih vektora sadržaja** koji su u njegovoj blizini
 - Odnosno, vrši se **preslikavanje centroida u ugrađene vektore** koji pripadaju njihovim klasterima
- **Izvršavanje upita** se vrši u dva koraka:
 - Najpre se **vektor upita poredi sa svim centroidima** kako bi se **odredilo kojim n klasterima pripada**
 - Zatim se vrši **poređenje vektora upita isključivo sa vektorima sadržaja unutar n klastera** čiji su centroidi najbliži vektoru upita

Pretraga ANN – vektorski indeksi

- **Invertovana datoteka**

- Moguće je **podesiti sledeće parametre:**
 - **Broj klastera koji će biti kreiran** prilikom indeksiranja vektora – veći broj klastera smanjuje broj vektora po klasteru, što ubrzava pretrage unutar pojedinačnih klastera, ali povećava broj centroida koji će morati biti pretraženi u prvom koraku izvršavanja upita
 - **Broj klastera koji će biti obuhvaćen pretragom** prilikom izvršenja upita – veći broj klastera povećava sveobuhvatnost odgovora (engl. *Recall*), ali usporava izvršavanje upita (engl. *Latency*)
- Usled mogućnosti podešavanja parametara za različite slučajeve, pogodno je *IVF* koristiti za dopremanje slika ili semantičku pretragu teksta



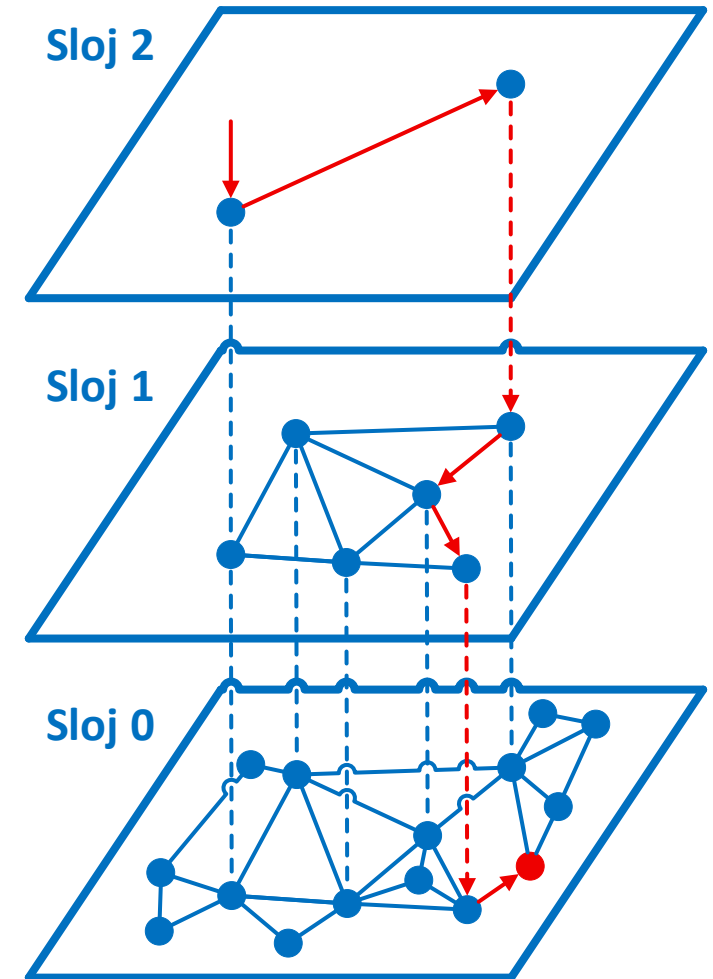
Pretraga ANN – vektorski indeksi

- **Tehnike zasnovane na grafu – HNSW**

- Algoritam **hijerarhijski navigacioni mali svetovi** (engl. *Hierarchical Navigable Small World (HNSW)*) kreira **graf** u kojem svaki **čvor** predstavlja **jedan ugrađeni vektor**, a **grane povezuju vektore koji su međusobno najsličniji**
 - Za svaki čvor **formiraju se grane između n najsličnijih ugrađenih vektora** dobijenih odabranom **metrikom sličnosti**
- **HNSW** predstavlja **višeslojni graf**
 - Na **najnižem sloju** nalaze **svi ugrađeni vektori** povezani u gustu mrežu susedstva na osnovu sličnosti
 - Svaki **viši sloj** sadrži **eksponencijalno manji podskup nasumično odabranih čvorova** sa nižeg sloja
 - Čvorovi na višim slojevima povezani su „**dužim**“ **granama** koje omogućavaju **brzo preskakanje velikih delova vektorskog prostora**
 - **Isti čvorovi na različitim slojevima** su takođe **povezani**, kako bi bilo moguće **kretati se između slojeva**
- Obezbeđuje **veliku sveobuhvatnost odgovora** na upit
 - Međutim, može da **smanji brzinu izvršavanja upita i zahteva velik memorijski prostor** usled skladištenja vektora u punom obimu, zajedno sa velikim brojem čvorova i grana u grafu

Pretraga ANN – vektorski indeksi

- **Tehnike zasnovane na grafu – HNSW**
 - Prilikom **izvršavanja upita**, algoritam kreće od **nasumično odabranog čvora najvišeg sloja** i traži čvor koji je **najsličniji ugrađenom vektoru upita**
 - Koristi se **pohlepna pretraga** (engl. *Greedy Search*) koja na osnovu **metrike sličnosti** proverava **tekući čvor i sve čvorove susede** i traži čvor najbliži vektoru upita, a zatim prelazi na taj čvor
 - Pohlepna pretraga se **ne vraća na prethodne čvorove**, već uvek ide unapred, sve dok ne dođe do **čvora čiji susedi nisu sličniji upitu od njega**
 - Kada pronade **lokalni optimum** na trenutnom sloju, algoritam se kroz isti čvor **spušta na niži sloj**
 - Proces se nastavlja sa sve **finijom navigacijom** dok se ne **pronađu najbliži susedi ugrađenog vektora upita na najnižem sloju** – na tom sloju se traži k najbližih čvorova



Pretraga ANN – vektorski indeksi

- **Odabir** odgovarajućeg vektorskog indeksa zavisi od:
 - **Domena primene**
 - Željenog **odnosa brzine izvršavanja upita i preciznosti odgovora**
 - **Memorijskog prostora** koji indeks zauzima
 - Vremena i resursa potrebnih za **kreiranje** indeksa
 - Vremena i resursa potrebnih za **ažuriranje** indeksa
 - U zavisnosti od tipa indeksa, dodavanjem novih ugrađenih vektora, postoji mogućnost da je potrebno ponovo kreirati ceo indeks, ili je moguće dodati samo nove ugrađene vektore u okviru indeksa
- Pored **vektorskih indeksa**, postoje i različiti tipovi **skalarnih indeksa**, odnosno indeksiranja atributa (meta-podataka) koji ne predstavljaju vektore
 - Mogu da se iskoriste prilikom **filtriranja slogova** u upitu

Pretraga ANN – vektorski indeksi

- Ukoliko je broj ugrađenih vektora **relativno mali**, kao i broj dimenzija latentnog prostora, **nije potrebno izvršiti indeksiranje** ugrađenih vektora
- Skladištenje ugrađenih vektora u njihovom **originalnom obliku**, bez bilo kakvih izmena ili upotrebe dodatnih struktura podataka, naziva se **Flat Indexing**
 - U kontekstu vektorskih baza podataka, često se takvo skladištenje ugrađenih vektora u latentnom prostoru takođe naziva **indeksiranjem**, jer su originalni podaci transformisani u ugrađene vektore i raspoređeni na način da **omogućavaju efikasnu pretragu sličnosti**
 - Zbog toga pojedini vektorski SUBP-ovi koriste termin indeksa umesto termina kolekcije
- **Flat Index** pruža **najveću preciznost** jer se ugrađeni vektor upita **poredi sa svim ugrađenim vektorima sadržaja**, što može **biti vremenski zahtevno**
 - Ovakav pristup koristi se kada je bitna **velika preciznost odgovora**, a **vreme** izvršavanja upita je **manje relevantno**

Sadržaj

- Vektori
- Ugrađeni vektori
- Modeli ugrađivanja
- Metrike rastojanja dva vektora
- Vektorska baza podataka
- Pretrage kNN i ANN
- Hibridna pretraga
- Primena vektorske baze podataka u arhitekturi RAG
- Poređenje relacije i vektorske baze podataka
- Literatura

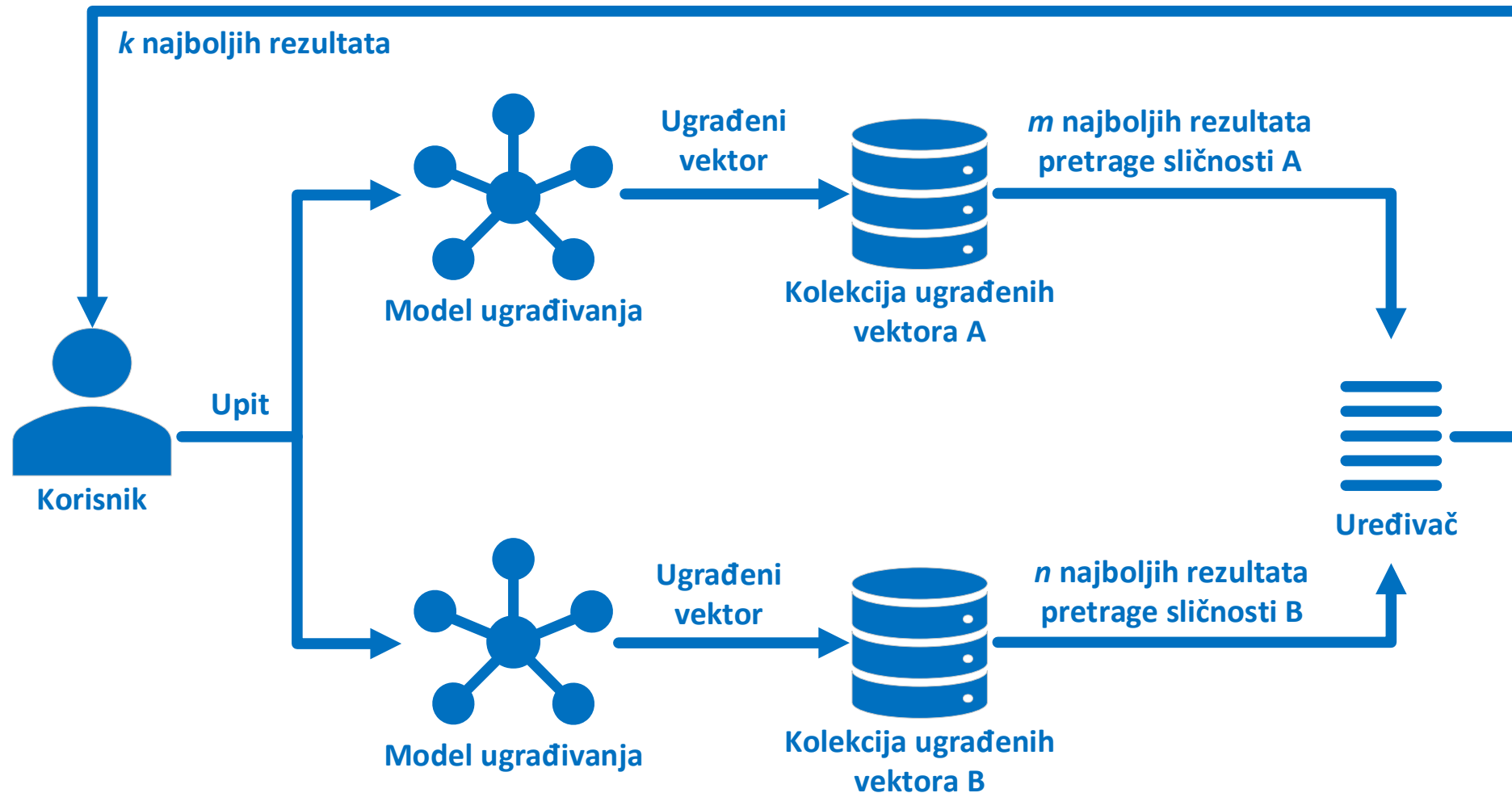
Hibridna pretraga

- **Hibridna pretraga** (engl. *Hybrid Search*) predstavlja **pristup koji integriše različite tipove pretraga** kako bi bili obezbeđeni precizniji i temeljniji rezultati
 - Naziva se i **viševektorska hibridna pretraga** (engl. *Multi-Vector Hybrid Search*)
- Postoje **dva osnovna tipa** hibridne pretrage:
 - **Multimodalna pretraga vektora** (engl. *Multimodal Vector Search*)
 - Kombinuje pretrage sličnosti ugrađenih vektora čiji su originalni **podaci različitog modaliteta**
 - **Pretraga retkih i gustih vektora** (engl. *Sparse-Dense Vector Search*)
 - Kombinuje rezultate **pretrage sličnosti** (engl. *Similarity Search*) i **pretrage tekstualnog sadržaja** (engl. *Keyword Search*)

Hibridna pretraga – multimodalna pretraga vektora

- **Multimodalna pretraga vektora** predstavlja objedinjenu pretragu **ugrađenih vektora** čiji sadržaj odgovara **različitim tipovima podataka**, poput teksta, slike i zvuka
 - Npr. korisnik može da postavi upit u formi teksta, nakon čega se pretražuju i opisi i slike proizvoda
- **Proces multimodalne pretrage vektora**
 - Korisnik postavlja **upit** koji se **transformiše u različite ugrađene vektore**, u zavisnosti od tipa podataka čiji će ugrađeni vektori biti pretraživani
 - Vrše se **različite pretrage sličnosti u paraleli** u **jednoj ili više kolekcija**
 - Posebno se **rangiraju rezultati za svaku od pretraga** na osnovu koeficijenta sličnosti (engl. *Similarity Score*)
 - Dopremaju su i **spajaju rezultati pretraga**
 - Vrši se **ponovno rangiranje** nad objedinjenim skupom rezultata i **odabira se k najboljih rezultata** od strane **uređivača** (engl. *Ranker*)
 - Moguće je **podešavanje parametara rangiranja** kako bi određena **pretraga dobila prednost**
 - **Pojedinačne pretrage** trebalo bi da dobave **više od k rezultata** kako bi se radilo filtriranje rezultata

Hibridna pretraga – multimodalna pretraga vektora



Hibridna pretraga – pretraga retkih i gustih vektora

- **Pretraga retkih i gustih vektora** predstavlja objedinjeno **semantičku pretragu** poređenjem ugrađenih vektora i **pretragu teksta** na osnovu reči zadatih u upitu
 - Time je moguće pretražiti **konkretne reči** tekstualnog sadržaja u vektorskoj bazi podataka, a ujedno i **semantiku reči**
- **Unapred obučen model ugrađivanja** često **ne može da generiše precizne ugrađene vektore** za terminologiju koja je **specifična za određeni domen**
 - **Ne može da obuhvati semantiku termina** specifičnih za domen, jer nije obučen u tu svrhu
 - Moguće je da **semantička pretraga ne vrati odgovarajući rezultat** na osnovu pojedinih reči zadatih u upitu (npr. poput identifikatora, konkretnih naziva, tehničke terminologije)
 - Dok **pretraga teksta** obezbeđuje da **reči zadate u upitu budu obuhvaćene pretragom**
- **Semantička pretraga** pogodna je za razumevanje **značenja teksta**, ali ne i za **specifičnu terminologiju, nazive, žargon, ili reči koje se ne koriste često**

Hibridna pretraga – pretraga retkih i gustih vektora

- Iskorišćene su **prednosti oba tipa pretraga**:
 - Pretraga konkretnih reči – da **važni termini ne budu izostavljeni**
 - Pretraga semantike reči – da **semantički slične reči**, poput sinonima, budu obuhvaćene pretragom
 - Koje inače ne bi bile obuhvaćene isključivo pretragom konkretno zadatih reči
- **Primer primene** – pretraga dokumenata iz oblasti medicine
 - Unapred obučeni modeli ugrađivanja ne mogu najbolje generirati ugrađene vektore za medicinsku terminologiju, poput naziva dijagnoza i lekova
 - Pretraga teksta na osnovu zadatih reči može da pokaže bolje rezultate od semantičke pretrage
 - Npr. semantička pretraga za simptome (npr. lek za glavobolju), dok pretraga konkretnih reči za tačne nazive lekova odnosno terapije (npr. Ibuprofen 400mg)
- Postoje različiti **algoritmi i tipovi pretraga teksta**
 - Pretrage poput tačno navedenih reči, fazi pretrage reči i pretrage reči na osnovu šablona
 - Algoritmi poput TF-IDF i BM25
 - Mnogi algoritmi i tipovi pretraga koriste se već u raznim sistemima za analizu teksta poput sistema *Elasticsearch*, ali i vektorskim bazama podataka

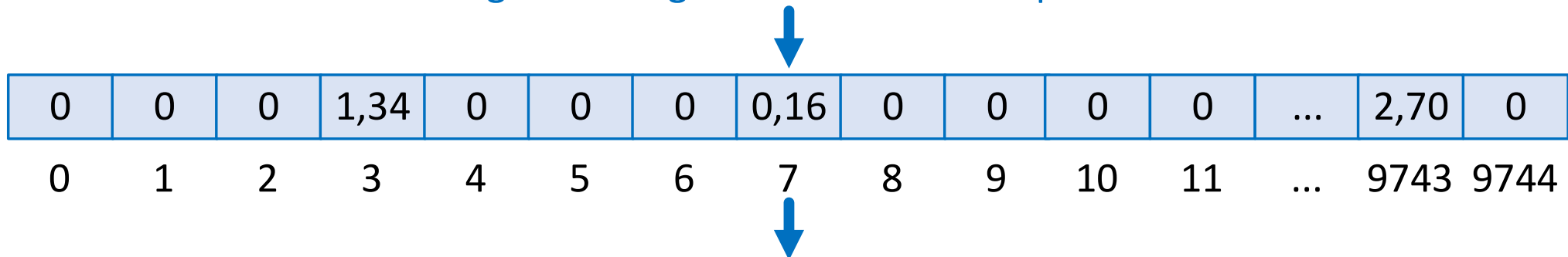
Hibridna pretraga – pretraga retkih i gustih vektora

- Za potrebe pretrage retkih i gustih vektora koriste se **dva tipa vektora**:
 - **Retki vektori** (engl. *Sparse Vectors*) – koriste se za **tekstualnu pretragu**
 - **Svaka dimenzija** retkih vektora predstavlja **jedinstveni temin u rečniku**
 - Stoga ovakvi vektori **sadrže velik broj nula**
 - Dimenzije čije su vrednosti **različite on nule** reprezentuje **termine korišćene u upitu ili sadržaju**
 - **Gusti vektori** (engl. *Dense Vectors*) – koriste se za **semantičku pretragu**
 - **Ugrađeni vektori**
 - **Nemaju većinski nula vrednosti** kao retki vektori
 - **Manjih su dimenzija** od retkih vektora
 - Dobijeni su pomoću **modela ugrađivanja**
- Retki i gusti vektori se **odvojeno skladište** u vektorskim bazama podataka

Hibridna pretraga – pretraga retkih i gustih vektora

- Usled potrebe za efikasnijim skladištenjem retkih vektora, reprezentuju se **samo vrednosti različite od nule i indeks odgovarajuće dimenzije**
 - Svaka vrednost različita od nule predstavlja se kao par **ključ-vrednost – {indeks: težina}**
 - **Indeks** predstavlja **redni broj ili identifikator reči u rečniku**
 - **Težina** predstavlja **značaj reči** – vrednost dobijena od **algoritama poput TF-IDF ili BM25** na osnovu učestalosti pojavljivanja reči i njene važnosti, ili od **modela mašinskog učenja za generisanje retkih vektora** (npr. *Sparse Lexical and Expansion Model – SPLADE*)

“Pretraga retkih i gustih vektora u bazi podataka”

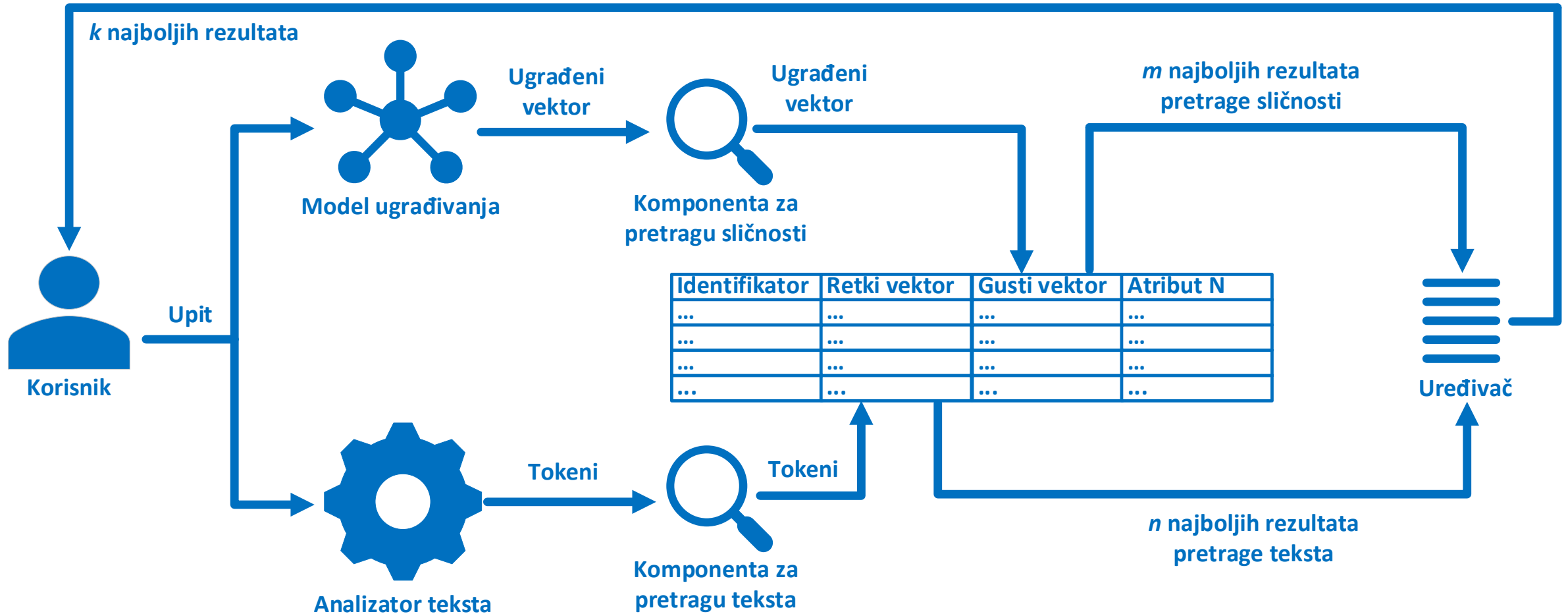


[{3: 1,34}, {7: 0,16}, {102: 4,89}, {721: 0,21}, {2003: 4,71}, {4689: 2,42}, {7632: 5,81}, {9743: 2,70}]

Hibridna pretraga – pretraga retkih i gustih vektora

- **Proces pretrage retkih i gustih vektora**
 - Korisnik postavlja **upit** koji se **transformiše** i u **gusti** i u **retki vektor**
 - Vršiti se **pretraga sličnosti gustih vektora** i **pretraga teksta retkih vektora u paraleli**
 - Posebno se **rangiraju rezultati obe pretrage** na osnovu koeficijenta sličnosti
 - Dopremaju se i **spajaju rezultati obe pretrage**
 - Vršiti se **ponovno rangiranje** nad objedinjenim skupom rezultata i **odabira se k najboljih rezultata** od strane **uređivača**
 - Moguće je **podešavanje parametara rangiranja** kako bi određena **pretraga dobila prednost**
 - **Pojedinačne pretrage** trebalo bi da dobave **više od k rezultata** kako bi se radilo filtriranje rezultata

Hibridna pretraga – pretraga retkih i gustih vektora



Hibridna pretraga – pretraga retkih i gustih vektora

- **Prednosti, nedostaci i primena** pretrage retkih i gustih vektora:
 - **Pretraga retkih vektora** – koriste se **konkretne ključne reči** u upitu
 - **Prednosti:** brza i efikasna pretraga konkretnih reči
 - **Nedostaci:** izostavlja tekst koji sadrži reči sa sličnim ili istim značenjem, poput sinonima
 - **Primena:** pretraga konkretno zadatih reči
 - **Pretraga gustih vektora** – koristi se **semantika reči** u upitu
 - **Prednosti:** može da prepozna sinonime, semantiku i kontekst reči
 - **Nedostaci:** zahtevnije sa stanovišta resursa; može da izostavi tačna poklapanja sa rečima iz upita
 - **Primena:** semantička pretraga na osnovu prirodnog jezika
 - **Pretraga retkih i gustih vektora** – koristi se **semantika i tačno poklapanje reči iz upita**
 - **Prednosti:** može da pruži preciznije odgovore na upite usled upotrebe oba tipa pretraga
 - **Nedostaci:** kompleksnija za implementaciju i održavanje; zahteva složenije testiranje i podešavanje parametara uređivača; zahteva više memorijskog prostora i procesne snage
 - **Primena:** semantička pretraga i pretraga konkretnog teksta

Hibridna pretraga – pretraga retkih i gustih vektora

- Pored retkih i gustih vektora, u vektorskim bazama podataka koriste se i **binarni vektori** dobijeni **transformacijom ugrađenih vektora u binarne**
 - **Pretraga binarnih vektora** – koristi se **prirodni jezik** u upitu
 - **Prednosti:** kompresuju veličinu vektora; zauzimaju manje memorijskog prostora (1 bit po dimenziji); manje računarskih resursa za procesiranje
 - **Nedostaci:** manja semantička preciznost; teže da obuhvate kompleksne semantičke odnose; koriste se često u kombinaciji sa drugim tipovima vektora
 - **Primena:** brzo poređenje sličnosti i ukljanjanje duplikata podataka; mogu se koristiti kao pre-filtriranje vektora koje prati preciznija pretraga gustih i/ili retkih vektora; kada je važna brzina izvršavanja upita i optimizacija skladištenja podataka (npr. mobilni uređaji); upotreba Hamingovog i Žakarovog rastojanja
- **Binarni vektor:** [0; 1; 0; 0; 1; 1; 1; 0; 1]
- **Retki vektor:** [{5: 0,74}; {21: 1,84}; {104: 0,12}; {576: 4,90}; {1042: 3,23}]
- **Gusti vektor:** [3,41; -2,12; 0,43; 5,67; -1,55]

Hibridna pretraga – ponovno rangiranje

- **Uređivač** (engl. *Ranker*) predstavlja komponentu vektorske baze podataka koja služi za **spajanje i ponovno rangiranje rezultata različitih pretraga**
- Postoje **različiti uređivači**, odnosno **strategije ponovnog rangiranja** rezultata, poput:
 - **Težinski uređivač** (engl. *Weighted Ranker*)
 - Dodeljuje veći ili manji **značaj određenim pretragama** na osnovu **zadatih težina**
 - Na taj način, moguće je dati **veću prednost određenim pretragama**, ukoliko se ispostavi da su **bolje za određeni domen**
 - Odnosno, koristi se kada je **poznato koji tip pretrage daje bolje rezultate**
 - **Uređivač recipročnog spajanja rangova** (engl. *Reciprocal Rank Fusion (RRF) Ranker*)
 - **Balansira relevantnost** svakog tipa pretrage
 - Umesto koeficijenta sličnosti, posmatra **rang rezultujućih vektora** iz pojedinačnih pretraga
 - Koristi se kada **nije potrebno da se određeni tip pretrage naglasi**
- Moguće je koristiti i različite **modele mašinskog učenja kao uređivače**
 - Na osnovu **semantike dopremljenog sadržaja** vrše rangiranje rezultata pretrage
 - Mogu da prate performanse izvršavanja upita na osnovu korisničke povratne informacije i **dinamički menjaju težine** pretraga

Hibridna pretraga – težinski uređivač

- **Težinski uređivač: postupak formiranja rezultata** hibridne pretrage
 - **Korak 1: Prikupljanje rezultujućih vektora** zajedno sa odgovarajućim **koeficijentima sličnosti** za svaku pretragu posebno
 - **Korak 2: Normalizacija koeficijenata sličnosti** rezultata od svake pretrage posebno
 - Prilikom upotrebe **različitih metrika sličnosti**, moguće je da **koeficijenti sličnosti nisu uporedivi**
 - Npr. Kosinusno rastojanje $[0, 2]$, Euklidsko rastojanje $[0, \infty)$, Skalarni proizvod $(-\infty, \infty)$
 - Svi koeficijenti mogu biti stavljeni pod **isti opseg**, npr. $[0, 1]$
 - **Korak 3: Dodeljivanje težina rezultatima** svake pojedinačne pretrage
 - **Novi koeficijent sličnosti** pojedinačnog sloga računa se kao **suma njegovih koeficijenata sličnosti** za svaku pretragu pomnoženu sa **zadatom težinom** za odgovarajuću pretragu

$$f(s) = \sum_{i=1}^n w_i * s_i$$

n – broj različitih pretraga
 s – niz koeficijenata sličnosti sloga
 w – niz dodeljenih težina

- **Korak 4: Uređivanje slogova** na osnovu novih koeficijenata sličnosti i **filtriranje k najbolje rangiranih slogova**

Hibridna pretraga – težinski uređivač

- **Primer primene težinskog uređivača** za rezultate dobijene iz kolekcije proizvoda – postoje ugrađeni vektori slika proizvoda, kao i ugrađeni vektori opisa proizvoda

$$w_{\text{slika}} = 0,65 \quad \text{top_k_slika} = 3 \quad \text{top_k} = 3$$

$$w_{\text{opis}} = 0,35 \quad \text{top_k_opis} = 3$$

ID	Koef. slika
7	0,78
47	0,85
61	0,68

ID	Koef. opis
47	0,76
61	0,82
98	0,91

ID	Koef. slika	Koef. opis	Novi koeficijent
7	0,78	-	$0,65 * 0,78 + 0,35 * 0 = 0,51$
47	0,85	0,76	$0,65 * 0,85 + 0,35 * 0,76 = 0,82$
61	0,68	0,82	$0,65 * 0,68 + 0,35 * 0,82 = 0,73$
98	-	0,91	$0,65 * 0 + 0,35 * 0,91 = 0,32$

ID	Novi koef.
47	0,82
61	0,73
7	0,51

Hibridna pretraga – težinski uređivač

- Potrebno je **testirati uređivač nad specifičnim domenom**, podešavajući **različite vrednosti težina**
 - **Svaka pretraga** može imati **drugačiju težinu** u opsegu [0, 1]
 - Što je **zadata težina veća** za neku pretragu, odnosno bliža vrednosti 1, time je odgovarajuća **pretraga značajnija**
 - **Slog dobija sve veću prednost** ukoliko se pojavi u **više različitih pretraga**
- Težinski uređivač koristi se često kod **multimodalne pretrage vektora**, kada **određene pretrage treba značajnije da doprinesu** finalnom rezultatu
- **Primeri primene:**
 - **Internet prodavnica** – pretraga proizvoda na osnovu slike i opisa
 - Slike mogu biti pogodnije za jedan tip proizvoda, dok opis može biti pogodniji za drugi tip proizvoda
 - **Pretraga dokumenata** – upotreba različitih ugrađenih vektora za različite delove dokumenata
 - Veći značaj mogu dobiti naslovi u odnosu na pasuse
 - Veći značaj može dobiti semantička ili tekstualna pretraga

Hibridna pretraga – RRF uređivač

- **RRF uređivač: postupak formiranja rezultata** hibridne pretrage
 - **Korak 1: Prikupljanje rezultujućih vektora i njihovih rangova** za svaku pretragu posebno
 - **Korak 2: Računanje novih koeficijenata sličnosti**
 - **Novi koeficijent sličnosti** pojedinačnog sloga računa se kao **suma recipročnih vrednosti zbira koeficijenta poravnanja i ranga sloga** za određenu pretragu
 - **Ista vrednost koeficijenta poravnanja** koristi se prilikom zbira sa rangom sloga za bilo koju pretragu

$$f(r) = \sum_{i=1}^n \frac{1}{k + r_i}$$

n – broj različitih pretraga u kojima slog pronađen
 r – niz rangova sloga
 k – koeficijent poravnanja

- **Korak 3: Uređivanje slogova** na osnovu novih koeficijenata sličnosti i **filtriranje k najbolje rangiranih slogova**

Hibridna pretraga – RRF uređivač

- **Primer primene RRF uređivača** za rezultate dobijene iz kolekcije proizvoda – postoje ugrađeni vektori slika proizvoda, kao i ugrađeni vektori opisa proizvoda

$$k = 10 \quad \text{top_k_slika} = 3 \quad \text{top_k} = 3 \\ \text{top_k_opis} = 3$$

ID	Rang slika
7	1
47	2
61	3

ID	Rang opis
47	1
98	2
7	3

ID	Rang slika	Rang opis	Novi koeficijent
7	1	3	$\frac{1}{10+1} + \frac{1}{10+3} = 0,1678$
47	2	1	$\frac{1}{10+2} + \frac{1}{10+1} = 0,1742$
61	3	-	$\frac{1}{10+3} = 0,0769$
98	-	2	$\frac{1}{10+2} = 0,0833$

ID	Novi koef.
47	0,1742
7	0,1678
98	0,0833

Hibridna pretraga – RRF uređivač

- Potrebno je **testirati uređivač nad specifičnim domenom**, podešavajući **različite vrednosti koeficijenta poravnanja**
 - **Vrednost** koeficijenta poravnanja može biti **jednak ili veći od 0**
 - Određuje **koliki uticaj imaju visoko rangirani slogovi** u određenim pretragama
 - Što je **koeficijent manji**, time **visoko rangirani slogovi** po bilo kojoj pretrazi imaju **veći uticaj**, a što je **koeficijent veći**, time se **ublažava uticaj visoko rangiranih slogova** po bilo kojoj pretrazi
 - Odnosno, za **veliku** vrednost koeficijenta, **ravnomernije se posmatraju svi rangovi**
 - Često se koeficijentu poravnanja dodeljuje **vrednost 60**
 - **Balans uticaja** visoko rangiranih slogova i **ravnomerne zastupljenosti** ostalih rangova
 - **Slog dobija sve veću prednost** ukoliko se pojavi u **više različitih pretraga**

ID	Rang slika
7	1
47	2
61	3

ID	Rang opis
47	1
98	2
7	3

ID	k=10	k=60
47	0,1742	0,0325
7	0,1678	0,0323
98	0,0833	0,0161
61	0,0769	0,0159

Hibridna pretraga – RRF uređivač

- **Primeri primene:**

- **Internet prodavnica** – pretraga proizvoda na osnovu slike i opisa
 - Kada su i slika i opis podjednako važni
- **Pretraga dokumenata** – napisani na različitim jezicima
 - Modeli ugrađivanja za različite jezike mogu dati različite koeficijente sličnosti
 - Preuređivanje rezultata nezavisno od jezika originalnog teksta
- Prilikom upotrebe **više modela ugrađivanja za isti sadržaj**
 - Kombinuje rezultate pretrage sličnosti ugrađenih vektora dobijenih od različitih modela ugrađivanja
 - Na način da ni jedan model ugrađivanja ne dobija prednost

Sadržaj

- Vektori
- Ugrađeni vektori
- Modeli ugrađivanja
- Metrike rastojanja dva vektora
- Vektorska baza podataka
- Pretrage kNN i ANN
- Hibridna pretraga
- Primena vektorske baze podataka u arhitekturi RAG
- Poređenje relacije i vektorske baze podataka
- Literatura

Veliki jezički modeli

- **Veštačka inteligencija** (engl. *Artificial Intelligence*) predstavlja pojam koji obuhvata uopšteno inteligentne mašine
- **Mašinsko učenje** (engl. *Machine Learning*) predstavlja polje veštačke inteligencije namenjeno za učenje i prepoznavanje šablona u podacima
- **Duboko učenje** (engl. *Deep Learning*) predstavlja polje mašinskog učenja namenjeno za prepoznavanje šablona i rad sa nestrukturiranim podacima, oslanjajući se na neuronske mreže
- **Veliki jezički modeli** (engl. *Large Language Models*) predstavljaju polje dubokog učenja namenjeno za razumevanje i obradu prirodnog jezika, odnosno teksta



Veliki jezički modeli

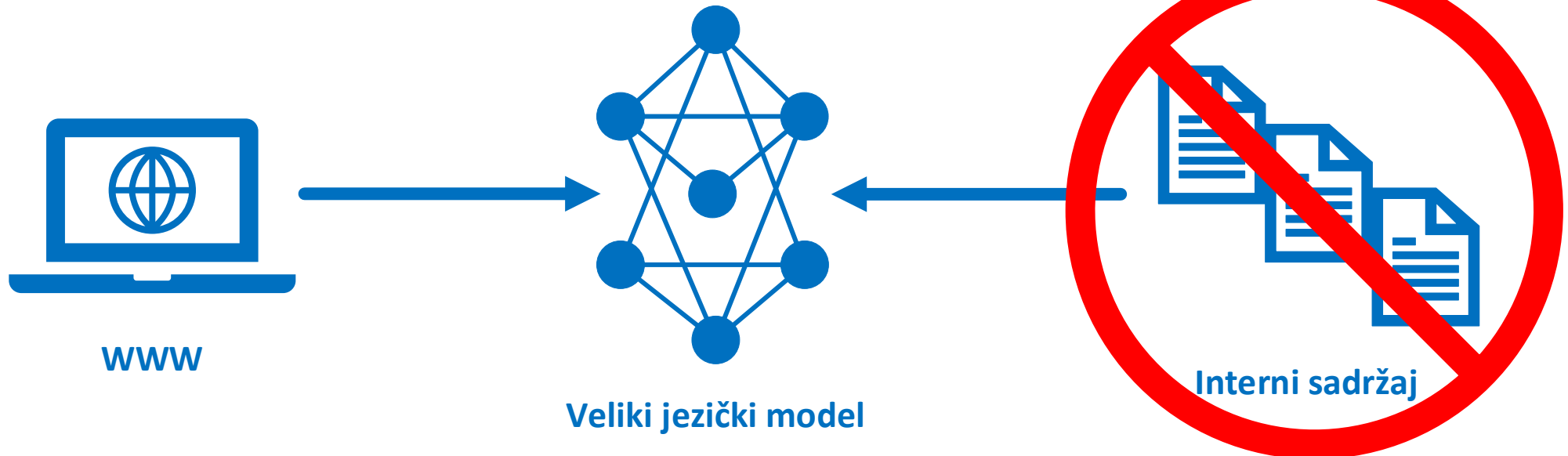
- **Generativna veštačka inteligencija** (engl. *Generative AI*) predstavlja **model dubokog učenja** koji služi da **generiše različit sadržaj**, poput teksta, slika, audio i video zapisa
- **Veliki jezički model** predstavlja **podskup** generativne veštačke inteligencije i može da obavlja različite zadatke **generisanja i procesiranja sadržaja nastalog prirodnim jezikom** (engl. *Natural Language Processing (NLP) Tasks*)
 - **NLP** omogućava računaru da **analizira, razume i komunicira koristeći ljudski govor ili napisan tekst** primenom mašinskog učenja
- Postoje **unapred obučeni veliki jezički modeli** (engl. *Pre-Trained LLMs*) dostupni za korišćenje
 - **Obuka neuronske mreže** vrši se nad velikim skupom podataka
 - Npr. javno dostupnog teksta na internetu
 - **Primeri** modela LLM: GPT, PaLM, BERT, Claude, LLaMA

Veliki jezički modeli – primena

- Modeli LLM mogu da **generišu tekst** time što **predviđaju pojavljivanje reči** u tekstu
 - Takođe mogu da **generišu i programski kôd**
 - Neuronske mreže modela LLM sadrže **milijarde neurona, odnosno parametara**
- **Pogodni** su za zadatke poput:
 - Formiranja **sažetka teksta**
 - **Prevođenje ili parafraziranje teksta**
 - Pružanja **odgovora** na zadata **pitanja** (Q&A)
 - **Transformacije programskog kôda** iz jednog jezika u drugi
 - **Transformacije datoteka** iz jednog formata u drugi
 - **Generisanje** novog teksta ili programskog kôda
 - ...

Veliki jezički modeli – izazovi

- Potencijalni **nedostaci** prilikom primene modela LLM:
 - Na određena pitanja specifična za domen **nema odgovor** usled **nedostatka znanja**
 - Npr. nema znanje o internim dokumentima kompanije
 - Mogućnost **halucinacija** – navođenje **odgovora koji nije tačan ili izmišljanje „činjenica“**
 - Moguće da pruži odgovor na osnovu **zastarelih podataka**



Veliki jezički modeli – pristupi pružanja dodatnog znanja

- **Pružanje dodatnog znanja i konteksta** modelu LLM može da reši ili ublaži navedene nedostatke
- **Osnovni pristupi pružanja novog znanja** modelima LLM:
 - **(A) Obuka novog modela LLM**
 - **(B) Fino podešavanje** (engl. *Fine-Tuning*) unapred obučenog modela LLM
 - **(C) Dodavanje konteksta u okviru prompta** unapred obučenog modela LLM
 - **(D) Primena arhitekture RAG** (engl. *Retrieval-Augmented Generation*)
- **(A) Obuka novog modela LLM:**
 - Omogućava **kontrolu nad arhitekturom** modela LLM, **sadržajem** koji se koristi za obuku, **domenom** primene, kao i **načinom** na koji model **odgovora na upite**
 - Međutim, veoma zahtevno – potrebni su **izuzetno veliki računarski resursi i potencijalno meseci rada**
 - Ovakav pristup obično koriste **specijalizovane kompanije**

Veliki jezički modeli – pristupi pružanja dodatnog znanja

- **(B) Fino podešavanje** unapred obučenog modela LLM
 - Pristup u kojem se **menjaju parametri**, odnosno poslednji slojevi neuronske mreže
 - **Prilagođavanje** unapred obučenog modela LLM određenom **domenu ili specifičnom zadatku**
 - Pogodno za **učenje šablona**
 - Npr. za generisanje programskog kôda nekog jezika, uz pridržavanje određenih pravila pisanja kôda; ili za potrebe određenog dijalekta programskog jezika
 - Pogodno za **izmenu stila pisanja odgovora** ili upotrebu **specifične terminologije**
 - Npr. medicinska ili pravna terminologija
 - Zahteva **velike računarske resurse i vreme**, kao i pažljivo **pripremljene podatke i validaciju**
 - Značajno **jednostavnije od obučavanja novog modela LLM** za specifičan domen, ali i dalje zahtevno

Veliki jezički modeli – pristupi pružanja dodatnog znanja

- **(C) Dodavanje konteksta u okviru prompta unapred obučenog modela LLM**
 - **Dodavanje konteksta** poput dokumenata, slika i programskog kôda, **u prompt modela LLM**, a zatim **postavljanje upita**
 - **Najjednostavniji način** pružanja konteksta
 - Međutim, kontekst postoji **samo u toku sesije**
 - Kako pronaći odgovor u **dokumentu od nekoliko hiljada strana?**
 - Postoji **ograničenje u broju ulaznih tokena** koji mogu da se pošalju po upitu (**kontekstni prozor**)
 - **Cena pružanja servisa** ukoliko se model LLM koristi u oblaku
 - Veća verovatnoća da **odgovor ne bude generisan**

Can you explain this code written in Python?

```
with open('example.txt', 'a') as file:  
    data = "This is the new data to be inserted into the file.\n"  
  
    file.write(data)  
  
print("Data has been inserted into the file.")
```

 Copilot

Absolutely! Let's break down this Python code step by step:

1. Opening the File:

```
1 with open('example.txt', 'a') as file:  
2
```

 - This line opens the file named `example.txt` in append mode (`'a'`).
 - The `with` statement ensures that the file is properly closed after the block of code is executed, even if an error occurs.
 - `file` is the file object that you can use to interact with the file.
2. Defining the Data:

```
1 data = "This is the new data to be inserted into the file.\n"  
2
```

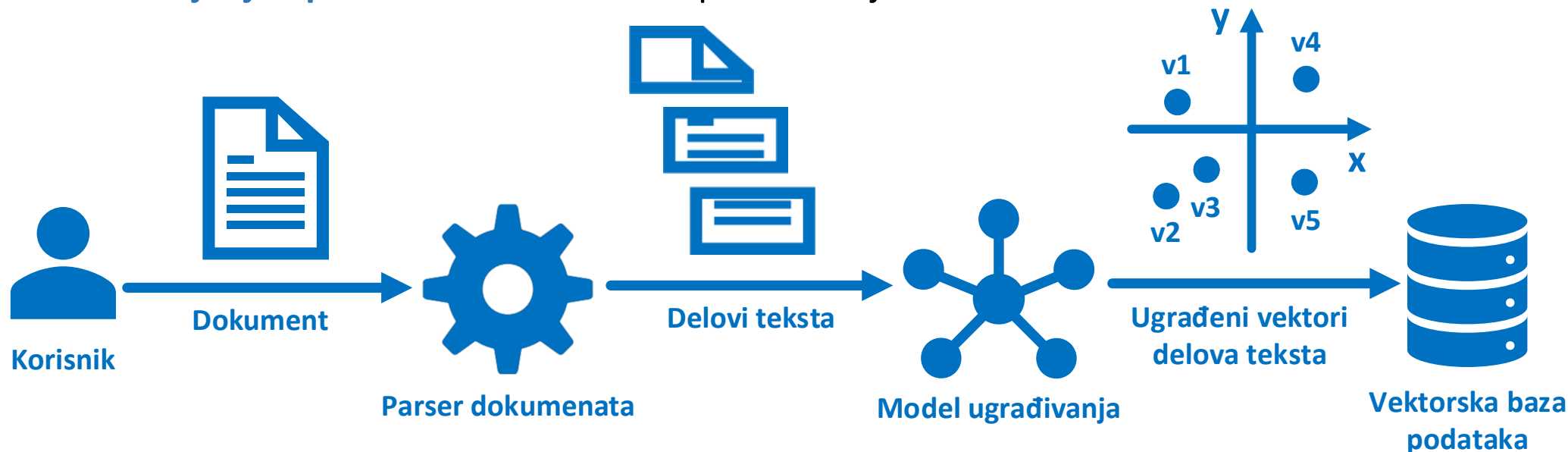
Veliki jezički modeli – pristupi pružanja dodatnog znanja

• (D) Primena arhitekture RAG

- **Arhitektura RAG** omogućava modelu LLM da bude povezan sa **eksternim skladištem podataka** kako bi samo određeni **podaci relevantni za upit** bili poslani modelu LLM prilikom generisanja odgovora
 - Odnosno, potrebno je **poslati samo relevantan kontekst** na postavljen upit
- **Kontekst** iz kojeg je potrebno pronaći odgovor na pitanje moguće je dopremiti iz trajne memorije, poput **vektorske baze podataka**
- **Smanjuje halucinacije** modela LLM i moguće je prikazati **referencu na izvor** na osnovu kojeg je odgovor generisan
- Moguće je koristiti **unapred obučen ili fino podešen** model LLM
- Potrebna **dodatna infrastruktura** za podršku rada arhitekture RAG

Arhitektura RAG i vektorska baza podataka

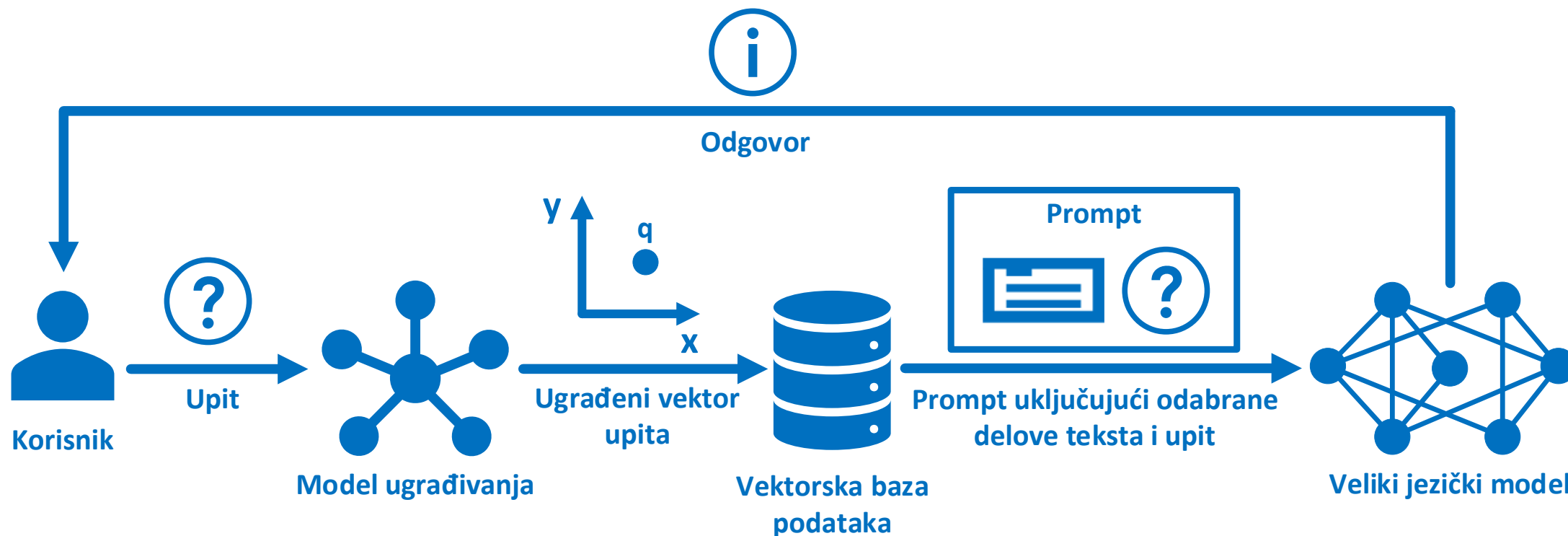
- **Primer primene vektorske baze podataka u arhitekturi RAG za Q&A nad dokumentima**
- **Dve osnovne aktivnosti:**
 - **Unos dokumenata u sistem**
 - **Postavljanje upita od strane korisnika prirodnim jezikom**



Arhitektura RAG i vektorska baza podataka

- **Osnovni koraci unosa dokumenata u sistem:**
 - **Korak 1 – Slanje dokumenta** parseru i **ekstrakcija njegovog teksta**
 - **Ekstrakcija** teksta iz segmenata dokumenta poput pasusa, tabela i slika
 - **Uklanjanje** nepotrebnog ili ponavljajućeg sadržaja
 - **Deljenje** dokumenta na manje segmente teksta
 - **Korak 2 – Slanje delova teksta** u model ugrađivanja i **generisanje ugrađenih vektora** teksta
 - **Korak 3 – Skladištenje ugrađenih vektora delova teksta** u kolekcije vektorske baze podataka

Arhitektura RAG i vektorska baza podataka



Arhitektura RAG i vektorska baza podataka

- **Osnovni koraci postavljanja upita:**

- **Korak 1 – Postavljanje upita** u formi pitanja na prirodnom jeziku koje se šalje modelu ugrađivanja i **generisanje ugrađenog vektora** upita
- **Korak 2 – Slanje ugrađenog vektora upita** vektorskoj bazi podataka i **pretraga sličnih** ugrađenih vektora delova dokumenata
- **Korak 3 – Formiranje instrukcija, odnosno prompta** koji se šalje modelu LLM, a može da sadrži:
 - **Smernice** na koji način model LLM treba da generiše odgovor
 - **Kontekst**, odnosno originalni tekstualni sadržaj delova dokumenata koji odgovaraju ugrađenim vektorima dobijenim pretragom u vektorskoj bazi podataka
 - Originalni **upit** koji je korisnik poslao
- **Korak 4 – Generisanje odgovora** na osnovu poslatog prompta od strane modela LLM i **slanje odgovora korisniku**

Arhitektura RAG i vektorska baza podataka – primer upita

- Upit: *Koje modele podataka koriste grafske baze podataka?*
- Odgovor unapred obučenog modela LLM bez primene arhitekture RAG

Grafske baze podataka koriste grafski model podataka, koji se zasniva na teoriji grafova iz matematike. Umesto tradicionalnih tabela kao u relacionim bazama, grafske baze organizuju podatke kao čvorove (nodes) i veze (edges). Evo pregleda ključnih modela koje koriste:

Osnovni grafski model:

- *Čvorovi (Nodes): Predstavljaju entitete (npr. osoba, mesto, dokument)*
- ...

Property Graph Model (najčešći)

...

RDF Model (Resource Description Framework)

...

- **Napomena: model hipergrafa nije spomenut u odgovoru**

Arhitektura RAG i vektorska baza podataka – primer upita

- Upit: *Koje modele podataka koriste grafske baze podataka?*
- Odgovor unapred obučenog modela LLM primenom arhitekture RAG
 - Uneta prezentacija sa predavanja u vektorsku bazu podataka

Prema informacijama iz datog konteksta, grafske baze podataka koriste tri osnovna grafska modela podataka:

1. *Model grafa sa svojstvima (engl. Property Graphs)*
2. *Model grafa trojki (engl. Triples)*
3. *Model hipergraфа (engl. Hypergraphs)*

Zajednička osobina za sva tri modela je da poseduju čvorove i grane.

Ova informacija je navedena u dokumentu ["NAIS_P3-4-5_Grafske_baze_podataka.pdf"](#) na strani 28.

Dodatno, važno je napomenuti da grafske baze podataka generalno koriste strukturu grafa za skladištenje podataka, gde su osnovni koncepti čvorovi i grane. Čvorovi reprezentuju realne entitete, dok grane reprezentuju odnose između entiteta.

Arhitektura RAG i vektorska baza podataka – primer upita

- Prva dva rangirana ugrađena vektora dobijena iz vektorske baze podataka:
- Ugrađeni vektor 1: Slajdovi 27, 28, 29

• Koriste algoritme obilaska grafa za izvršavanje upita • Prateći grane između čvorova Napredne arhitekture informacionih sistema: Grafske baze podataka 27 • Tri osnovna grafska modela podataka: • Model grafa sa svojstvima (engl. Property Graphs) • Model grafa trojki (engl. Triples) • Model hipergrafa (engl. Hypergraphs) • Zajednička osobina za sva tri modela je da poseduju čvorove i grane Napredne arhitekture informacionih sistema: Grafske baze podataka 28 • Model grafa sa svojstvima sadrži čvorove, grane i svojstva

- Ugrađeni vektor 2: Slajdovi 26 i 27

• Osnovni koncepti koje koristi grafska baza podataka su čvorovi i grane • Grafske baze podataka reprezentuju realne entitete kao čvorove • Dok odnose između entiteta reprezentuju kao grane • Grafske baze podataka uglavnom su napravljene za transakcione, OLTP sisteme • Optimizovane za CRUD • Dve osnovne komponente sistema za upravljanje grafskom bazom podataka: • Skladište podataka • Endžin za obradu podataka Napredne arhitekture informacionih sistema: Grafske baze podataka 26 • Skladište podataka: • Grafske baze podataka koriste izvorno grafsko skladište podataka (engl. Native Graph Storage) koje čuva podatke u formi grafa

Sadržaj

- Vektori
- Ugrađeni vektori
- Modeli ugrađivanja
- Metrike rastojanja dva vektora
- Vektorska baza podataka
- Pretrage kNN i ANN
- Hibridna pretraga
- Primena vektorske baze podataka u arhitekturi RAG
- Poređenje relacije i vektorske baze podataka
- Literatura

Poređenje relacione i vektorske baze podataka

- Za razliku od relacione baze podataka u kojoj se čuvaju podaci raspoređeni po **redovima i kolonama**, vektorska baza podataka čuva **ugrađene vektore**
 - Način **optimizacije** rada vektorske baze podataka je drugačiji, kao i način **postavljanja upita**
- U relacionoj bazi podataka vrši se **pretraga po redovima i kolonama** na osnovu **tačno zadatih uslova**, dok u vektorskoj bazi podataka vrši se primarno **pretraga sličnosti ugrađenih vektora**
- Za razliku od vektorskih baza podataka, **relacione baze podataka nisu pogodne** za skladištenje **nestrukturiranih podataka**
 - Mogućnost **proširenja** relacione baze podataka da podrži rad sa ugrađenim vektorima (npr. *PostgreSQL pgvector*)

Poređenje relacione i vektorske baze podataka

Relaciona baza podataka	Vektorska baza podataka
Pogodnije za strukturirane podatke	Pogodnije za nestrukturirane podatke
Skladištenje podataka u formi slogova relacije	Skladištenje podataka u formi ugrađenih vektora
Pretraga podataka na osnovu konkretnih uslova	Pretraga podataka na osnovu sličnosti
Rezultat upita predstavljaju slogovi koji zadovoljavaju konkretne uslove	Rezultat upita predstavlja sadržaj koji je najbliži zadatom upitu
Nisu pogodne za semantičku pretragu	Pogodne su za semantičku pretragu
Pogodne su za kompleksne uslove pretrage	Nisu pogodne za kompleksne uslove pretrage koji prevazilaze pretragu sličnosti
Tehnike indeksiranja poput B-stabla i bitmap indeksa	Tehnike indeksiranja poput heširanja i kvantizacije
Izazovnije skaliranje	Jednostavnije skaliranje
Jednostavnije za primenu i razumevanje	Izazovnije za primenu i razumevanje

Sadržaj

- Vektori
- Ugrađeni vektori
- Modeli ugrađivanja
- Metrike rastojanja dva vektora
- Vektorska baza podataka
- Pretrage kNN i ANN
- Hibridna pretraga
- Primena vektorske baze podataka u arhitekturi RAG
- Poređenje relacije i vektorske baze podataka
- Literatura

Literatura

- Nitin Borwankar, *Vector Databases: A Practical Introduction*, 1st Edition, O'Reilly Media, Inc., 2026.
- Rachelle Palmer, Ben Perlmutter, Ashwin Gangadhar, Nicholas Larew, Sigfrido Narváez, Thomas Rueckstiess, Henry Weller, Richmond Alake, Shubham Ranjan, *Building AI Intensive Python Applications: Create intelligent apps with LLMs and vector databases*, 1st Edition, Packt Publishing Ltd., 2024.
- James Phoenix, Mike Taylor, *Prompt Engineering for Generative AI: Future-Proof Inputs for Reliable AI Outputs*, 1st Edition, O'Reilly Media, Inc., 2024.
- Chip Huyen, *AI Engineering: Building Applications with Foundation Models*, 1st Edition, O'Reilly Media, Inc., 2025.



Napredne arhitekture informacionih sistema

Vektorske baze podataka Pitanja?

Predmetni nastavnik:
dr Marko Vještica

