Primerjava pristopov učenja pri matrični faktorizaciji v priporočilnih sistemih

Marjan Meglen
University of Primorska, FAMNIT
Koper, Slovenia
89242035@student.upr.si

***Izvleček*—V tej nalogi je sta bili analizirani napovedna napaka in razložljivost priporočilnega sistema, ki temelji na matrični faktorizaciji z uporabo odrezanega razcepa matrike na singularne vrednosti (SVD). Primerjani sta bili dve strategiji učenja latentnih faktorjev: sočasno učenja vseh faktorjev ter postopno učenje po posameznih faktorjih. Modela sta bila ocenjena na podatkovni zbirki MovieLens, pri čemer sta bili analizirani tako napovedna napaka kot ločljivost žanrov v latentnem prostoru. Rezultati kažejo, da sočasno učenje prinese nižjo validacijsko napako, medtem ko postopno učenje omogoča večjo razložljivost določenih latentnih faktorjev. Pristop s postopnim učenjem je pokazal statistično značilne razlike med nekaterimi žanri, kar nakazuje na potencial za razložljivost v določenih primerih.**

I. Uvod

V današnjem digitalnem okolju se uporabniki redno srečujejo z veliko količino vsebin in možnostmi izbire. Priporočilni sistemi pomagajo poenostaviti to izkušnjo tako, da predlagajo vsebine glede na posameznikove interese, vedenje ali pretekle izbire. Uporabljajo se v različnih kontekstih, kot so spletne trgovine (npr. Amazon), pretočne platforme (npr. Netflix, Spotify), družbena omrežja, novice in iskalniki. Njihov osnovni cilj je, da olajšajo dostop do relevantnih informacij in povečajo verjetnost, da bo uporabnik z vsebino dejansko tudi v interakciji.

Obstaja več vrst priporočilnih sistemov. Vsebinski sistemi delujejo na osnovi lastnosti vsebin in zgodovine uporabnikov na primer, če je uporabniku všeč določen žanr, bo sistem predlagal podobne vsebine. Sistemi s sodelovalnim filtriranjem pa temeljijo na podobnosti med uporabniki: če imata dva uporabnika podobne vzorce ocen, naj bi jima bile všeč podobne stvari. Obstajajo tudi hibridni pristopi, ki združujejo oba načina in poskušajo izkoristiti prednosti obeh [1].

Znotraj sodelovalnega filtriranja je ena izmed najpogosteje uporabljenih tehnik matrična faktorizacija, ki je postala bolj znana po Netflixovem tekmovanju za izboljšanje priporočilnega sistema. Ideja je preprosta: veliko matriko ocen (v kateri večina vrednosti manjka) razdelimo na dve manjši matriki, ki predstavljata uporabnike in vsebine. Te nove matrike zajemajo značilnosti (t. i. latentne faktorje), ki jih model sam odkrije iz podatkov.

Ena izmed najzgodnejših praktičnih implementacij matrične faktorizacije je bila FunkSVD, ki jo je med Netflix tekmovanjem populariziral Simon Funk [2]. Njegov pristop je temeljil na postopnem učenju, kjer se vsak latentni faktor uči posebej, enega za drugim. Tak način omogoča interpretabilnost kar pomeni, da posamezni faktorji pogosto predstavljajo smiselne značilnosti, kot na primer komedije proti dramam. Vendar se je kasneje pokazalo, da je učenje vseh faktorjev hkrati bolj učinkovito za validacijsko napako. Takács in sod. (2009) so razvili različico FunkSVD, kjer se vsi latentni faktorji posodabljajo hkrati z gradientnim spustom, kar je privedlo do hitrejšega učenja in boljših rezultatov na testnih podatkih [3].

Ob tem se v zadnjem času v stroki vse več pozornosti namenja tudi razložljivosti napovednih modelov. V kontekstu umetne inteligence in priporočilnih sistemov to pomeni, da mora biti delovanje modela razumljivo uporabniku in razvijalcu. Razložljivost postaja pomemben dejavnik pri sprejemanju odločitve o tem, katero metodo uporabiti še posebej v primerih, kjer je zaupanje v sistem ključno [4].

V tej raziskovalni nalogi preučujemo, kako različna učna pristopa vplivata na napovedno napako in razložljivost modela. Primerjana sta dva pristopa:

1) Učenje enega latentnega faktorja naenkrat, kjer se model uči postopoma, faktor za faktorjem.
2) Učenje vseh latentnih faktorjev hkrati.

Cilj je bolje razumeti, kako učni postopek vpliva na kakovost modela tako z vidika napovedovanja kot razložljivosti. S tem lahko prispevamo k bolj premišljenemu razvoju priporočilnih sistemov, ki nimajo le nizke napovedne napake, temveč so tudi razumljivi.

II. Metodologija

A. Opis podatkovne zbirke

V tej nalogi je bila uporabljena podatkovna zbirka MovieLens ml-latest-small, ki so jo razvili raziskovalci skupine GroupLens na Univerzi v Minnesoti. Gre za eno najbolj razširjenih zbirk za testiranje algoritmov priporočilnih sistemov, saj so v njej vključeni kakovostni in raznoliki podatki o ocenah filmov. Vsebuje približno 100 000 ocen za 9724 filmov od 610 uporabnikov. Uporabniki so anonimizirani, pri čemer je bil pogoj za vključitev v podatkovno zbirko, da je vsak uporabnik ocenil vsaj 20 filmov [5].

Ocene so bile shranjene v datoteki `ratings.csv`, kjer je vsaka instanca predstavljena s štirimi atributi: identifikator uporabnika, identifikator filma, ocena (na lestvici od 0.5 do 5.0) ter časovni žig. Struktura teh podatkov je prikazana v tabeli I, kjer so prikazani prvi in zadnji zapisi v datoteki. Opis filmov je bil zajet v datoteki `movies.csv`, kjer so bili za

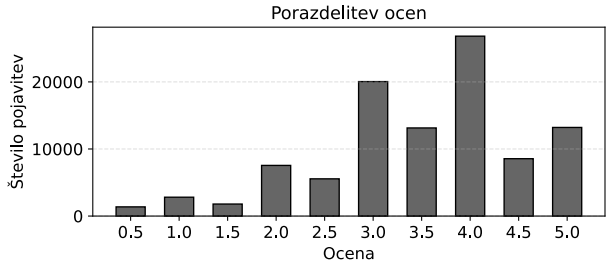
Index	userId	movieId	rating	timestamp
0	1	1	4.0	964982703
1	1	3	4.0	964981247
2	1	6	4.0	964982224
3	1	47	5.0	964983815
4	1	50	5.0	964982931
...	...	...	...	...
100831	610	166534	4.0	1493848402
100832	610	168248	5.0	1493850091
100833	610	168250	5.0	1494273047
100834	610	168252	5.0	1493846352
100835	610	170875	3.0	1493846415

Tabela I: `ratings.csv` - prvih in zadnjih 5 vrstic

vsak film navedeni naslov ter pripadajoči žanri. V tabeli II so prikazani začetni in končni primeri filmov.

Podatkovna zbirka je bila analizirana za porazdelitev ocen in porazdelitev žanrov. Na sliki 1 je prikazana porazdelitev vseh ocen, ki so bile zbrane v podatkovni zbirki. Iz prikazane porazdelitve je mogoče videti, da so bile srednje in visoke ocene pogosteje podane kot nizke, kar kaže na izrazito pozitivno pristranskost v ocenjevalnem vedenju uporabnikov.

Izvedena je bila tudi analiza najpogostejših žanrov. Pri tem je bil upoštevan zgolj prvi žanr v nizu, ki je bil dodeljen posameznemu filmu. Na sliki 2 je bila prikazana porazdelitev prvih žanrov, iz katere je bilo razvidno, da so nekatere zvrsti (npr. komedija, drama in akcija) močno prevladovale. Redkejše zvrsti, kot so muzikal, vojni filmi ali film-noir, pa so bile zastopane le z nekaj primeri. To pomeni, da so filmi v zbirki porazdeljeni neenakomerno glede na žanrsko klasifikacijo, kar lahko vpliva na pristranskost pri učenju modelov.

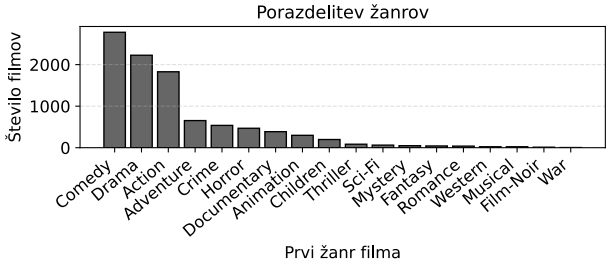


Slika 1: Porazdelitev ocen v podatkovni zbirki MovieLens.

B. Opis implementacije

V tej nalogi je bil za priporočilni sistem implementiran odrezani razcep matrike na singularne vrednosti (ang. truncated singular value decomposition). Gre za različico razcepa matrike na singularne vrednosti, kjer se iz razcepa ohranijo le najpomembnejše latentni faktorji, kar omogoča učinkovito kompresijo informacij [6].

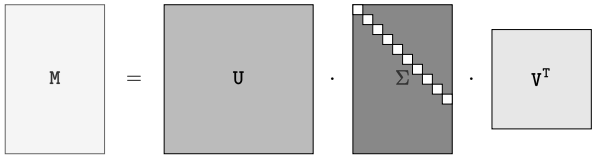
Pri tem pristopu je matrika ocen predstavljena kot produkt dveh matrik: ene, ki opisuje filme z latentnimi faktorji (lastnosti, ki jih model sam odkrije), in druge matrike, ki opisuje



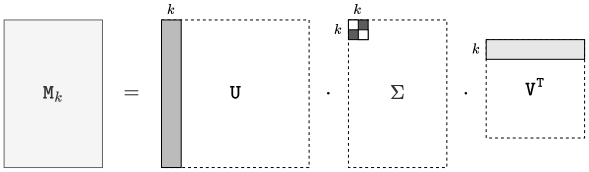
Slika 2: Porazdelitev prvih žanrov v podatkovni zbirki MovieLens.

uporabnike z njihovimi preferencami po istih faktorjih. Vsaka vrstica v prvi matriki predstavlja film, vsak stolpec v drugi pa uporabnika [2].

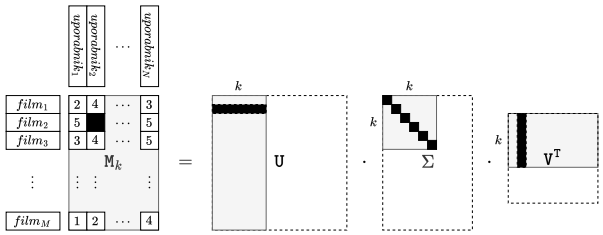
Na sliki 3 je prikazan splošni razcep matrike na singularne vrednosti, kjer je prvotna matrika ocen razceplenja na tri faktorje: ortonormirani matriki U in V^T ter diagonalno matriko singularnih vrednosti Σ. Na sliki 4 je prikazan odrezani razcep, kjer se ohranijo le največje k vrednosti, kar zmanjša velikost in kompleksnost modela. Na sliki 5 pa je prikazana shema, kako se z uporabo odrezanega razcepa na singularne vrednosti izvede napoved posamezne ocene med filmom in uporabnikom.



Slika 3: Razcep matrike ocen na singularne vrednosti. (Vir: https://timbaumann.info/svd-image-compression-demo/)



Slika 4: Odrezani (truncated) razcep na singularne vrednosti z zmanjšanim številom faktorjev. (Vir: https://timbaumann.info/svd-image-compression-demo/)



Slika 5: Napoved ocene z uporabo latentnih vektorjev uporabnika in filma.

movieId	title	genres
1	Toy Story (1995)	Adventure—Animation—Children—Comedy
2	Jumanji (1995)	Adventure—Children—Fantasy
3	Grumpier Old Men (1995)	Comedy—Romance
4	Waiting to Exhale (1995)	Comedy—Drama—Romance
5	Father of the Bride Part II (1995)	Comedy
...	...	...
193581	Black Butler: Book of the Atlantic (2017)	Action—Animation—Comedy—Fantasy
193583	No Game No Life: Zero (2017)	Animation—Comedy—Fantasy
193585	Flint (2017)	Drama
193587	Bungo Stray Dogs: Dead Apple (2018)	Action—Animation
193609	Andrew Dice Clay: Dice Rules (1991)	Comedy

Tabela II: `movies.csv` - prvih in zadnjih 5 vrstic

Za učenje modela je bil uporabljen pristop z gradientnim spustom, kjer se model optimizira na učnem naboru 80% ocen, njegovo delovanje pa se validira na preostalih 20% podatkov. Implementirani sta bili dve različni strategiji učenja latentnih faktorjev.

V prvi strategiji se uči en faktor naenkrat: za vsak faktor latentnega prostora se izvede več epoh učenja, med katerimi se posodablja le ena komponenta vektorjev, ki predstavljajo filme in uporabnike. Psevdokodo tega pristopa prikazije Algoritem 1.

Algoritem 1: Učenje po enem latentnem faktorju

Vhod: matrika ocen R, število latentnih faktorjev k, število epoh E
Izhod: matriki P in Q

```
for vsak latentni faktor f = 1 ... k do
    for vsako epoho e = 1 ... E do
        for vsak primer (i, j, r_ij) v učnem naboru do
            Napovej r_ij = Σ_k p_ik q_kj
            Izračunaj napako e_ij = r_ij − r̂_ij
            Posodobi p_if in q_fj z gradientnim spustom
        end
    end
end
```

Druga strategija vse faktorje uči hkrati, kar ustreza bolj klasični obliki matrične faktorizacije. V vsakem koraku se posodobita celotna vektorja latentnih predstavitev uporabnika in filma. Psevdokodo tega pristopa prikazije Algoritem 2.

C. Evalvacija modela

Napovedna uspešnost modela je bila ocenjena z uporabo korena povprečne kvadratne napake (RMSE), ki meri povprečno odstopanje med napovedano in dejansko oceno. Metrika RMSE je bila izračunana po vsaki epohi na validacijskem naboru podatkov, kar je omogočilo sprotno spremljanje konvergence in stabilnosti učnega procesa.

Algoritem 2: Sočasno učenje vseh latentnih faktorjev

Vhod: matrika ocen R, število latentnih faktorjev k, število epoh E
Izhod: matriki P in Q

```
for vsako epoho e = 1 ... E do
    for vsak primer (i, j, r_ij) v učnem naboru do
        Napovej r_ij = Σ_k p_ik q_kj
        Izračunaj napako e_ij = r_ij − r̂_ij
        Posodobi celotna vektorja p_i in q_j z
          gradientnim spustom
    end
end
```

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{(i,j)\in D} (\hat{r}_{ij} - r_{ij})^2} \tag{1}$$

kjer D predstavlja množico parov (film, uporabnik) v validacijskem naboru, r_{ij} dejansko oceno, in $\hat{r}_{ij}$ napovedano oceno.

Poleg ocenjevanja napovedne napake je bila izvedena tudi analiza razložljivosti modela. Za analizo in vizualizacijo sta bila uporabljena prva dva naučena latentna faktorja. Filmi z enim žanrom so bili razporejeni v dvorazsežni prostor, pri čemer so bile barve uporabljene za označevanje žanrskih skupin.

Za oceno ločjivosti žanrov glede na prvi latentni faktor so bili izvedeni statistični testi. Najprej je bila za vsak način učenja modela posebej preverjena razlika med žanri glede na vrednosti prvega latentnega faktorja s pomočjo enosmerne analize variance (ANOVA). Nato so bile izvedene še primerjave med pari žanrov z uporabo neodvisnih t-testov za vse kombinacije žanrov, pri čemer je bila uporabljena Bonferronijeva korekcija za nadzor nad večkratnim testiranjem.

III. Rezultati

A. Primerjava validacijske napake med pristopoma učenja

V tem razdelku so predstavljeni rezultati obeh pristopov učenja: (1) postopno učenje posameznih latentnih faktorjev in (2) sočasno učenje vseh latentnih faktorjev. Oba modela sta bila učena na isti učni in validacijski množici, pri čemer so bile uporabljene enake vrednosti hiperparametrov, kjer je bilo to mogoče. Vrednosti hiperparametrov so prikazane v tabeli III, rezultati napovedne napake pa v tabli IV.

Parameter	**Postopno**	**Sočasno**
Latentni faktorji	40	enako kot postopno
Ocene (učenje / validacija)	80668 / 20168	enako kot postopno
Epoh	100 na faktor	100
Hitrost učenja	0,001	enako kot postopno

Tabela III: Primerjava uporabljenih hiperparametrov pri postopnem in sočasnem pristopu.

Metrika	**Postopno**	**Sočasno**
RMSE (validacija)	1,384	1,010
Čas učenja	19 min	44 s

Tabela IV: Primerjava napovedne napake in časa učenja med postopnim in sočasnim pristopom.

B. Razložljivost modela

Za oceno razložljivosti modelov je bila analizirana porazdelitev filmov glede na vrednosti prvih dveh latentnih faktorjev (`f1`, `f2`). V analizo so bili vključeni samo filmi z enim žanrom. Vizualizacije za oba pristopa so prikazane na slikah 7 in 8.

Za kvantitativno analizo so bile izvedene enosmerne analize variance (ANOVA) za faktor `f1`. Rezultati so prikazani v tabeli V.

Pristop učenja	p**-vrednost**
Postopno učenje	$3{,}53 \times 10^{-17}$
Sočasno učenje	0,2611

Tabela V: Rezultati enosmerne ANOVA analize za latentni faktor `f1`

Dodatno so bili pri postopnem učenju izvedeni parni t-testi z Bonferronijevo korekcijo, prikazani v tabeli VI.

Par žanrov	**Prvoten** p	**Prilagojen** p	**Značilno**
Comedy vs. Drama	0,0001	0,0001	da
Comedy vs. Documentary	0,0001	0,0001	da
Drama vs. Documentary	0,0001	0,0001	da
Documentary vs. Horror	0,0015	0,0232	da
Documentary vs. Thriller	0,0042	0,0627	ne
Comedy vs. Horror	0,0103	0,1552	ne
Action vs. Documentary	0,0216	0,3245	ne
Comedy vs. Action	0,1594	1,0000	ne
Comedy vs. Thriller	0,1274	1,0000	ne
Action vs. Drama	0,6845	1,0000	ne
Action vs. Thriller	0,9131	1,0000	ne
Action vs. Horror	0,8467	1,0000	ne
Drama vs. Thriller	0,5130	1,0000	ne
Drama vs. Horror	0,7678	1,0000	ne
Thriller vs. Horror	0,7197	1,0000	ne

Tabela VI: Rezultati parnih t-testov z Bonferronijevo korekcijo (postopno učenje)

Na sliki 9 so prikazane žanrske porazdelitve filmov z najnižjimi (0–10 %) in najvišjimi (90–100 %) vrednostmi faktorja `f1` za pristop s postopnim učenjem.

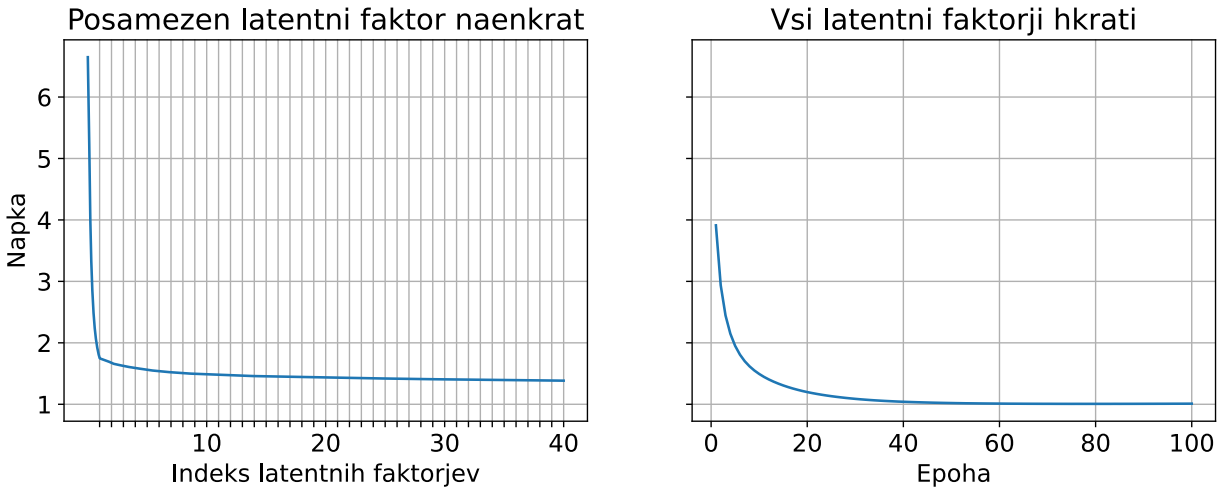
IV. Diskusija

Rezultati eksperimenta kažejo na pomembne razlike med pristopoma učenja, ki jih je mogoče razložiti z načini, kako model obravnava latentne faktorje med optimizacijo.

Pri metodi s sočasnim učenjem vseh latentnih faktorjev so se parametri vektorskih predstavitev uporabnikov in filmov prilagajali istočasno. Tak pristop omogoča boljšo optimizacijo celotnega sistema kot usklajene celote, kar vodi do hitrejše konvergence in nižje napovedne napake. Model v tem primeru deluje predvsem kot kompresijska shema za zajem kompleksnih uporabniško-vsebinskih interakcij, vendar brez nujne povezave med posameznimi faktorji in semantičnimi lastnostmi vsebin.

Po drugi strani pristop s postopnim učenjem vsakega latentnega faktorja posebej omogoča, da se vsak faktor razvije neodvisno, brez motenj iz drugih faktorjev. To ustvari pogoje, v katerih lahko faktorji spontano postanejo bolj strukturirani glede na skupne vsebinske značilnosti, kot so žanri. Ker model v danem trenutku optimizira le en vidik latentnega prostora, je večja verjetnost, da bodo faktorji odsevali posamezne, razložljive faktorje razlikovanja med filmi. Vendar ta prednost prihaja z očitno slabostjo: omejeno zmogljivostjo hkratnega učenja več informacij, kar se kaže v višji napovedni napaki.

Interpretabilnost, ki se pojavi pri postopnem učenju, je bila kljub vsemu omejena. Le nekateri faktorji so izražali jasne razlike med žanri, in še te predvsem za tiste žanre, ki so bili



Slika 6: Primerjava validacijske napake med obema pristopoma. Levo: učenje posameznega latentnega faktorja naenkrat. Desno: sočasno učenje vseh faktorjev.

pogosteje zastopani. To nakazuje, da je razložljivost odvisna tudi od porazdelitve podatkov, saj redke kategorije model težje učinkovito loči. Poleg tega obstaja možnost, da so razlike v interpretabilnosti tudi posledica interakcij med faktorji: pri sočasnem učenju lahko vplivi posameznih faktorjev ostanejo prekriti, saj se optimizacija dogaja v visoko soodvisnem prostoru.

Praktične implikacije teh razlik so odvisne od konteksta uporabe. Če so priporočila namenjena zgolj izboljšanju uporabniške izkušnje na podlagi nizke napovedne napake (npr. v komercialnih aplikacijah), je sočasno učenje verjetno bolj primerno. Če pa se model uporablja v okoljih, kjer je pomembna razlaga rezultatov, na primer v izobraževalnih ali zdravstvenih sistemih, potem lahko postopno učenje prispeva k večjemu zaupanju v delovanje sistema.

V prihodnjem delu bi bilo smiselno raziskati hibridne pristope. Možna je na primer uporaba regularizacijskih izrazov, ki spodbujajo ločljivost faktorjev glede na znane lastnosti (npr. žanrske oznake), ali metod, ki omogočajo nadzorovano strukturiranje latentnega prostora. Druga možnost so večfazni modeli, pri katerih se sprva optimizira napovedna natančnost, nato pa sledi dodatna faza, v kateri se latentni prostor prilagodi za večjo interpretabilnost brez bistvenega poslabšanja napovedne napake.

V. Zaključek

V tej raziskavi je bil implementiran in ocenjen priporočilni sistem, ki temelji na razcepu matrike ocen na singularne vrednosti. Primerjana sta bila dva različna pristopa učenja: sočasno in postopno učenje latentnih faktorjev. Rezultati so pokazali, da metoda sočasnega učenja doseže nižjo napako napovedi in hitrejšo konvergenco. Po drugi strani pa je pristop s postopnim učenjem omogočil delno razlago nekaterih faktorjev glede žanrskih razlik.

Analize so pokazale, da razložljivost ni enakomerno prisotna za vse žanre, temveč je omejena na določen podnabor, kjer je bila podprta z vizualno in statistično analizo. Zato se zdi, da izbira metode učenja ni odvisna le od napovedne napake, temveč tudi od tega, ali je v danem kontekstu pomembna možnost razlage rezultatov.

V prihodnjem delu bi bilo smiselno raziskati kompromisne pristope, ki bi omogočili boljšo interpretacijo brez večje napovedne napake. Možnosti vključujejo kombiniranje obeh metod ali uvajanje dodatnih struktur v model, ki bi omogočile bolj nadzorovano razporeditev semantično pomembnih informacij v latentni prostor.

Literatura

[1] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, 2005.

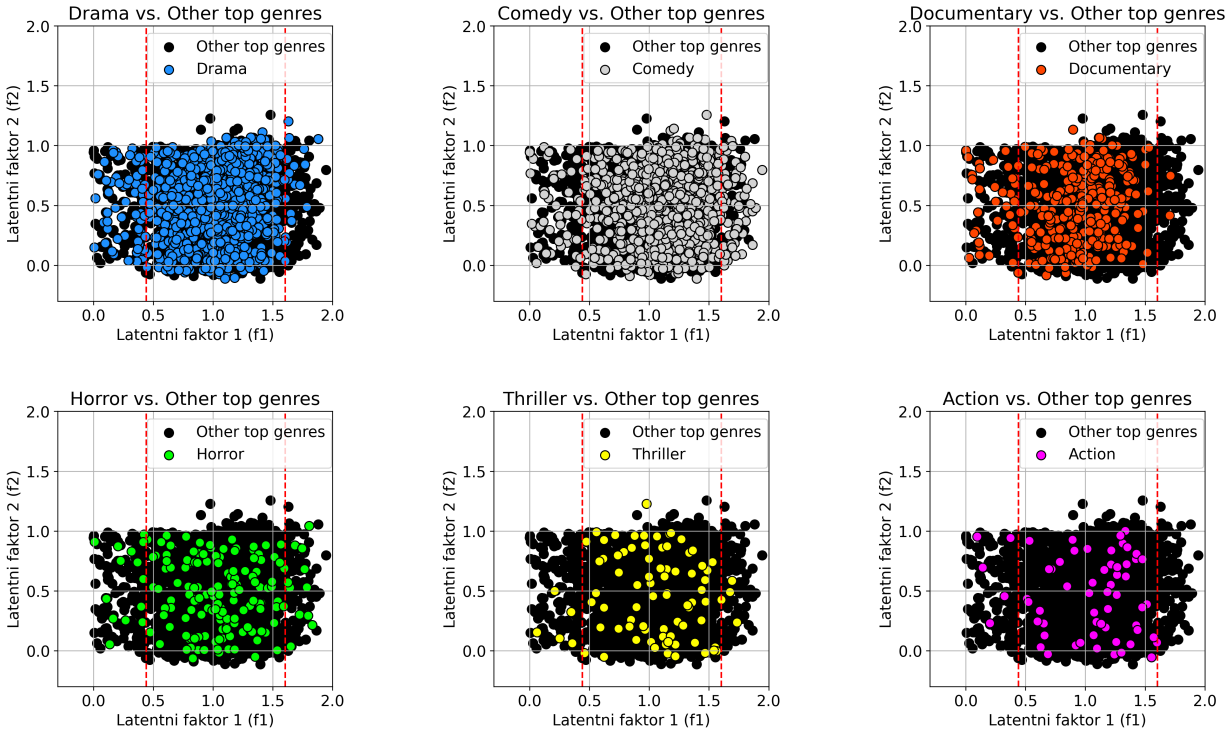
[2] S. Funk, "Netflix update: Try this at home," https://sifter.org/simon/journal/20061211.html, 2006, accessed: 2025-05-18.

[3] G. Takács, I. Pilászy, B. Németh, and D. Tikk, "Scalable collaborative filtering approaches for large recommender systems," *J. Mach. Learn. Res.*, vol. 10, p. 623–656, Jun. 2009.

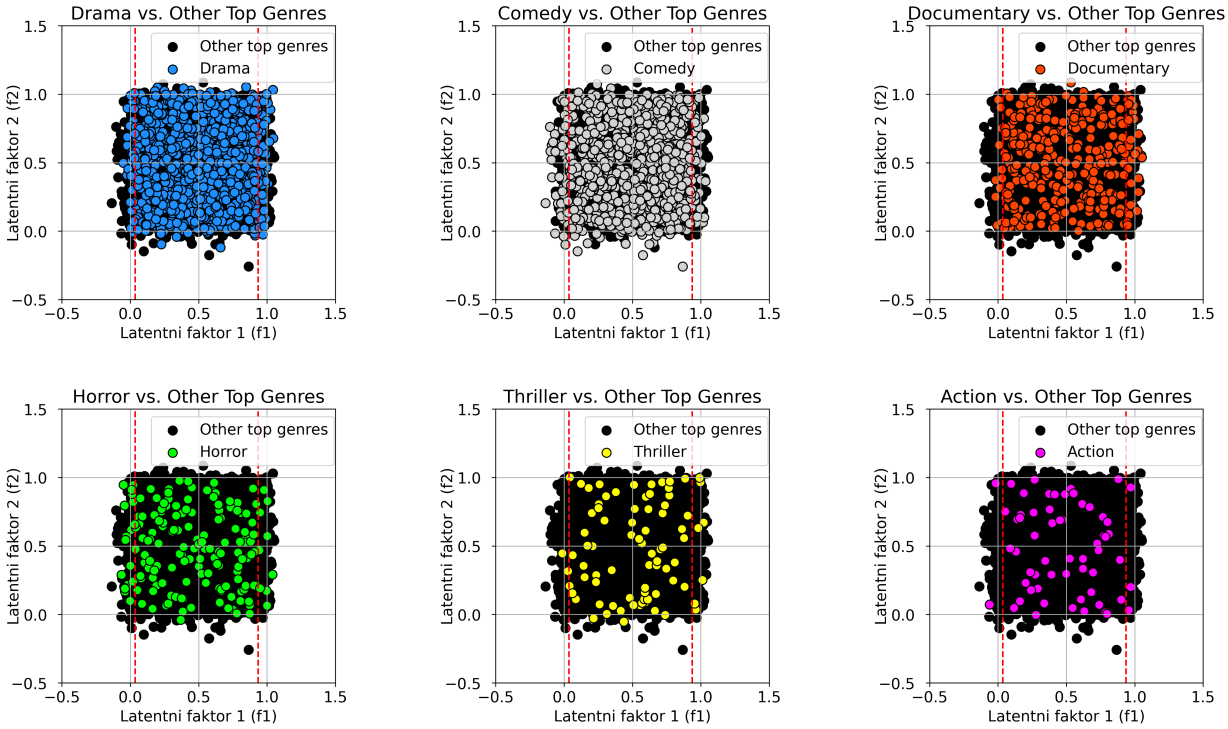
[4] P. Linardatos, V. Papastefanopoulos, and S. Kotsiantis, "Explainable ai: A review of machine learning interpretability methods," *Entropy*, vol. 23, no. 1, 2021. [Online]. Available: https://www.mdpi.com/1099-4300/23/1/18

[5] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," *ACM Transactions on Interactive Intelligent Systems (TiiS)*, vol. 5, no. 4, pp. 19:1–19:19, 2015. [Online]. Available: https://doi.org/10.1145/2827872

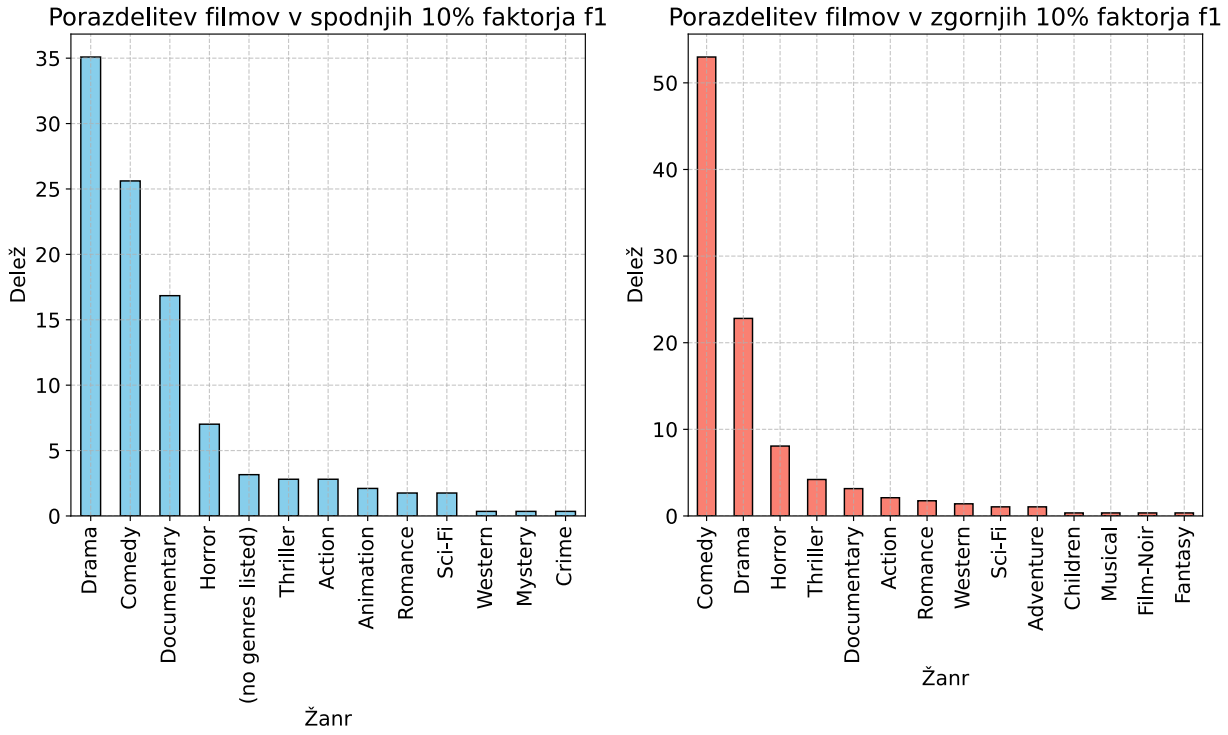
[6] A. Falini, "A review on the selection criteria for the truncated svd in data science applications," *Journal of Computational Mathematics and Data Science*, vol. 5, p. 100064, 2022. [Online]. Available: https://doi.org/10.1016/j.jcmds.2022.100064



Slika 7: Porazdelitev šestih najpogostejših žanrov glede na prva dva latentna faktorja - metoda z učenjem enega faktorja naenkrat.



Slika 8: Porazdelitev šestih najpogostejših žanrov glede na prva dva latentna faktorja - metoda s sočasnim učenjem vseh faktorjev.



Slika 9: Žanrska sestava filmov v skrajnih vrednostih faktorja `f1` - pristop s postopnim učenjem.