

Opis metod. IME DATOTEKE: ismf_bias_ones.py

V datoteki sem uporabil postopek inkrementalne simultane matrične faktorizacije ali ISMF, kjer skušamo originalno matriko ocen R aproksimirati z manjšima matrikama P in Q , tako da velja $R \approx PQ$. Prav tako sem uporabil regularizacijo, da preprečim preveliko prileganje učnim podatkom. Vsakemu od podatkov odštejem povprečje ocen ustreznega uporabnika in to povprečje pri napovedovanju ocene prištejem nazaj, s čimer v podatke 'zakodiram' pristranskost uporabnikov. Za upoštevanje pristranskosti uporabnikov in tudi pristranskosti do izvajalcev sem v matriki P prvi stolpec in v matriki Q drugo vrstico nastavil na 1.

IME DATOTEKE: naloga6.py*

Implementacija v datoteki *naloga6.py* prav tako uporablja ISMF in zgoraj opisane postopke priprave podatkov. Glavna razlika je v dodani logiki za napovedovanje v primeru neznanih uporabnikov ali izvajalcev, ki se nahaja v funkciji *predict()*. V primeru, da je uporabnik za katerega skušamo napovedati oceno še neznan (t.j. se ni pojavil v učnih podatkih), izvajalca pa poznamo, kot napoved uporabim povprečje ocen, ki jih je izvajalec že prejel od drugih uporabnikov. V primeru, da uporabnika poznamo, izvajalec pa je neznan (se ni pojavil v učnih podatkih), kot napoved uporabim povprečje uporabnikovih ocen. Kadar pa sta tako uporabnik kot izvajalec neznana, pa napovemo kar povprečje ocen celotne učne množice (1.82106569656).

Rezultati. V spodnji tabeli 1 so prikazani rezultati z ocenami iz tekmovalnega strežnika.

Tabela 1: Rezultati iz tekmovalnega strežnika.

ime	ocena
ismf_bias_ones.py	1.85206
naloga6.py*	1.79691

Izjava o izdelavi domače naloge. Domačo nalogo in pripadajoče programe sem izdelal sam.