Ime, priimek:

LV08: Priporočilni sistemi z uporabo surprise knjižnice

Namen vaje:

- Spoznati knjižnico surprise za priporočilne sisteme
- Izračunati napoved ocene filma za izbranega uporabnika s postopkom kolaborativnega filtriranja
- Oceniti povprečno napako priporočilniega sistema

1 Priporočila filmov s suprise knjižnico

1.1 Podatkovni set MovieLens

```
MovieLens data sets were collected by the GroupLens Research Project
at the University of Minnesota.

This data set consists of:
        * 100,000 ratings (1-5) from 943 users on 1682 movies.
        * Each user has rated at least 20 movies.
        * Simple demographic info for the users (age, gender, occupation, zip)

The data was collected through the MovieLens web site
(movielens.umn.edu) during the seven-month period from September 19th,
1997 through April 22nd, 1998. This data has been cleaned up - users
who had less than 20 ratings or did not have complete demographic
information were removed from this data set. Detailed descriptions of
the data file can be found at the end of this file.
```

1.2 Nalaganje podatkov

```
from tqdm import tqdm
import numpy as np
import pandas as pd

from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import linear_kernel

from surprise import NormalPredictor, SVD, KNNBasic, NMF
```

```
from surprise import Dataset, Reader
from surprise import accuracy
from surprise.model_selection import cross_validate, KFold
```

Definiramo pot do podatkov

```
RATINGS_DATA_FILE = '../Data/ml-latest-small/ratings.csv'
MOVIES_DATA_FILE = '../Data/ml-latest-small/movies.csv'

ratings_data = pd.read_csv(RATINGS_DATA_FILE)
movies_data = pd.read_csv(MOVIES_DATA_FILE)
```

Preglej in izpiši nekaj vrstic podatkov o filmih:

Preglej in izpiši nekaj vrstic podatkov o ocenah:

Movies Data:

	movieId	title	genres
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
1	2	Jumanji (1995)	Adventure|Children|Fantasy
2	3	Grumpier Old Men (1995)	Comedy|Romance
3	4	Waiting to Exhale (1995)	Comedy|Drama|Romance
4	5	Father of the Bride Part II (1995)	Comedy

Ratings Data:

	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

1.3 Priprava surprise podatkovnega seta

Dataset objekt predstavlja podatke v surprise knjižnjici.

```
# 1 Pripravi Surprise dataset za filme
from surprise import Dataset
from surprise import Reader

# Get minimum and maximum rating from the dataset
min_rating = ratings_data.rating.min()
max_rating = ratings_data.rating.max()

reader = Reader(rating_scale=(min_rating, max_rating))
data = Dataset.load_from_df(ratings_data[['userId', 'movieId', 'rating']], reader)
```

1.4 Modeliranje podatkov in napoved ocene

Obstaja več algoritmov, s katerimi modeliramo podatke, oziroma prilagodimo model podatkom. Na osnovi modela (algoritma) potem lahko izračunamo napovedi podatkov (ocen).

Primer:

```
from surprise import KNNBasic

# Retrieve the trainset.
trainset = data.build_full_trainset()

# Build an algorithm, and train it.
algo = KNNBasic()
algo.fit(trainset)
```

Sedaj lahko generiramo napoved ocene za izbranega uporabnika in film:

```
uid = 610  # raw user id (as in the ratings file).
iid = 168252  # raw item id (as in the ratings file).

#
pred = algo.predict(uid, iid, r_ui=4, verbose=True)
```

Naloga: preskusi za različne uporabnike (dodaj vrstice za 5 uporabnikov), in vstavi rezultate:

user: 610 item: 168252 r_ui = 4.00 est = 4.31 {'actual_k': 16, 'was_impossible': False}

```
user: 1      item: 168252    r_ui = None   est = 4.33   {'actual_k': 16, 'was_impossible': False}
user: 2      item: 168252    r_ui = None   est = 4.28   {'actual_k': 15, 'was_impossible': False}
user: 3      item: 168252    r_ui = None   est = 4.42   {'actual_k': 8, 'was_impossible': False}
user: 4      item: 168252    r_ui = None   est = 4.31   {'actual_k': 14, 'was_impossible': False}
user: 5      item: 168252    r_ui = None   est = 4.11   {'actual_k': 16, 'was_impossible': False}
```

Kaj podamo kot parameter metodi?

Metodi algo.predict podamo naslednje parametre:

- uid: ID uporabnika
- iid: ID artikla (filma)
- r_ui: (neobvezno) dejanska ocena, če je znana
- verbose: (neobvezno) če je nastavljeno na `True`, bo metoda izpisala dodatne informacije

Kaj predstavljajo vrednosti v izpisu rezultata?

Vrednosti v izpisu rezultata predstavljajo:

- `user`: ID uporabnika
- `item`: ID artikla (filma)
- r_ui: dejanska ocena (če je podana)
- `est`: ocenjena ocena
- `{actual_k, was_impossible}`: dodatne informacije o napovedi

Kako izračunamo, kolikšna je napaka napovedi?

Napako napovedi izračunamo z uporabo metrik, kot sta RMSE (Root Mean Squared Error) in MAE (Mean Absolute Error).

V dokumentaciji knjižnice surprise poišči opis izbranega algoritma zgoraj:
(https://surprise.readthedocs.io/en/stable/index.html)

Za algoritem KNNBasic, katere podatke upošteva, ter kaj pomeni parameter k ? Spreminjaj ta parameter (nekaj vrednosti), in ugotovi, ali se napaka napovedi poveča ali zmanjša?

Algoritem `KNNBasic` uporablja podatke o interakcijah med uporabniki in predmeti ter izračuna matriko podobnosti za iskanje najbližjih sosedov. Parameter `k` predstavlja število najbližjih sosedov, ki jih upošteva pri napovedovanju.

Iz rezultatov:

- `k=5, RMSE=0.9554`
- `k=10, RMSE=0.9395`
- `k=20, RMSE=0.9394`
- `k=40, RMSE=0.9464`
- `k=80, RMSE=0.9526`

Ko se `k` povečuje od 5 do 20, se RMSE zmanjšuje, kar kaže na izboljšano natančnost napovedi. Vendar pa nadaljnje povečanje `k` na 40 in 80 povzroči višji RMSE, kar nakazuje, da se natančnost napovedi zmanjšuje. To kaže, da obstaja optimalno območje za `k`, ki uravnoteži pristranskost in varianco.

Preglej dokumentacijo za similarity options. Kaj pomeni parameter user_based? Preskusi, ali se napovedi spremenijo, če spremeniš user_based parameter.

Parameter `user_based` v algoritmu `KNNBasic` določa, ali se podobnost izračuna med uporabniki ali predmeti. Če je `user_based=True`, algoritem izračuna podobnost med uporabniki; če je `user_based=False`, izračuna podobnost med predmeti.

Kaj pomeni parameter name, in katere so možne vrednosti ?

Parameter `name` v algoritmu `KNNBasic` določa, katero metriko podobnosti uporabiti. Možne vrednosti so:

- `'cosine'`: Kosinusna podobnost
- `'msd'`: Povprečna kvadratna razlika
- `'pearson'`: Pearsonov korelacijski koeficient
- `'pearson_baseline'`: Pearsonov korelacijski koeficient z osnovnimi ocenami

1.5 Naloga 1

Definiraj različne mere podobnosti v algoritmu KNNBasic (cosine, pearson, msd) in preveri, ali se vrednost napovedi razlikuje (za istega izbranega uporabnika). Vstavi kodo, rezultate in komentiraj:

1.6 Testiranje modela, izračun natančnosti napovedi (napake)

Za testiranje modela uporabljamo princip križne validacije, podatke razdelimo v učno in testno množico.

Primer

```
from surprise.model_selection import KFold
from surprise import accuracy, Dataset, SVD

kf = KFold(n_splits=3)

algo = KNNBasic()

for trainset, testset in kf.split(data):

    # učenje
    algo.fit(trainset)
    # napoved
    predictions = algo.test(testset)

    # ocena natančnosti
    accuracy.rmse(predictions, verbose=True)
```

Spremeni parameter k algoritma KNNBasic, in primerjaj rezultate (povrečno RMSE napako), za nekaj primerov.

Rezultati:

- `k=5, Average RMSE=0.9708`
- `k=10, Average RMSE=0.9532`
- `k=20, Average RMSE=0.9527`

Izpiši objekt predictions, kaj vsebuje, in koliko je teh podatkov?

Objekt `predictions` vsebuje napovedi za vsak testni primer. Število podatkov je enako številu napovedi, v tem primeru 100836. Prva napoved vsebuje informacije o uporabniku, artiklu, dejanski oceni (r_ui), ocenjeni oceni (`est`), in dodatne informacije (`actual_k`, `was_impossible`).

1.7 Križna validacija

```
# Testiraj natancnost algoritma s krizno validacijo

from surprise import SVD, KNNBasic, accuracy
from surprise.model_selection import cross_validate
```

```
# Izberi algoritem in parametre
algoritem = SVD(n_epochs=10)
# Izvedi križno validacijo (učenje in testiranje)
results = cross_validate(algoritem, data, measures=['RMSE', 'MAE'], cv=10, verbose=True)
```

Izpiši rezultat:

Evaluating RMSE, MAE of algorithm SVD on 10 split(s).

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10	Mean	Std
RMSE (testset)	0.8773	0.8614	0.8840	0.8753	0.8690	0.8852	0.8887	0.8740	0.8839	0.8833	0.8782	0.0080
MAE (testset)	0.6723	0.6676	0.6813	0.6754	0.6695	0.6816	0.6829	0.6731	0.6800	0.6797	0.6763	0.0052
Fit time	0.59	0.67	0.57	0.60	0.61	0.60	0.66	0.60	0.64	0.63	0.62	0.03
Test time	0.17	0.03	0.05	0.09	0.05	0.05	0.05	0.12	0.05	0.03	0.07	0.04

---

komentiraj, kaj pomenijo številke:

-RMSE (Root Mean Squared Error):

- MAE (Mean Absolute Error):

- Fit time: Čas, potreben za treniranje modela na posameznem delu podatkov.

- Test time: Čas, potreben za napovedovanje ocen na testnem setu.

Kateri so parametri križne validacije ?

· cv=10 : Število delitev (foldov) v križni validaciji.

· measures=['RMSE' , 'MAE'] : Merila za ocenjevanje

· return_train_measures=True : Vračanje meril za učni set

Preveri dokumentacijo algoritma SVD, za kakšen algoritem gre? Kaj pomenita parametra n_factors, n_epochs.

SVD (Singular Value Decomposition) se uporablja se za napovedovanje manjkajočih ocen v priporočilnih sistemih.

Parametra:

. n_factors : Število faktorjev (dimenzij) za faktorizacijo. Večje vrednosti lahko zajamejo več informacij, a povečujejo kompleksnost modela.

· n_epochs : Število ponovitev treniranja (epoh). Več epoh lahko vodi do boljšega učenja, vendar tudi daljšega časa treniranja.

Spremeni parametra algoritma, ponovi poskus in komentiraj spremenjene rezultate (napako, čas) :

· Napaka (RMSE in MAE): Sprememba parametrov je privedla do zmanjšanja napak, kar pomeni izboljšanje modela v smislu natančnosti napovedi. To kaže, da večje število latentnih faktorjev in epoh omogoča modelu, da se bolje prilagodi podatkom.

· Čas (Fit time in Test time): Čeprav se je čas treniranja povečal zaradi večjih parametrov, je čas napovedovanja zmanjšan, kar je pozitivno za učinkovitost modela med izvajanjem na novih podatkih.

1.8 Dodatna naloga: iskanje optimalnih parametrov

Kako bi z uporabo knjižnjice surprise poiskal najboljše parametre algoritma, ki dajo najmanjšo napako predikcije ?

Rešitev (koda, komentar):