

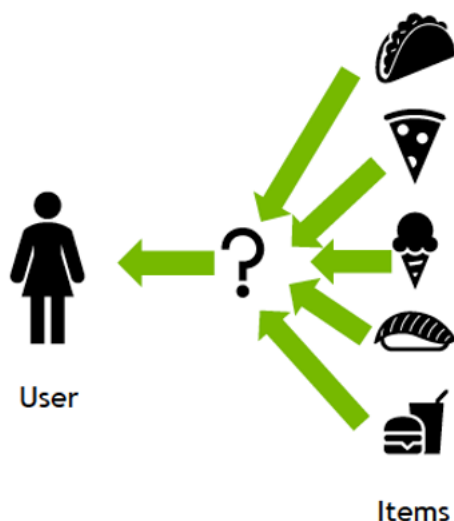
תרגיל 3 – מערכות המלצה

הקדמה

בתרגיל זה תבנו מערכת המלצה לסרטים.

התרגיל יחולק ל 3 חלקים

- ניתוח הנתונים
- בניית מערכת המלצה CF
- הערכות ביצועים



בקובץ zip המצורף נתונים לכם הקבצים הבאים:

תיקית plot ריקה לתוכה יכנסו תמונות הגרפים השונים שתבקשו להציג.

main.py – הקובץ הראשי, דרכו נקרא לכל המימושים השונים.

אין לשנות קובץ זה!

data.py – כאן תממשו פונ' שונות להבנת והצגת הנתונים.

collaborative_filtering.py – כאן תממשו מערכת המלצה CF מסוג user-based ו item-based

evaluation.py – כאן תממשו פונ' הערכה שונות למערכת ההמלצה.

בנפרד, קיים ZIP המכיל תיקיה בשם data עם קבצי הנתונים הרלוונטיים. הורידו אותם ושימו בסיפריית השורש.

הגשה

במהלך התרגיל תערכו את הקבצים הבאים: data.py, collaborative_filtering.py ו evaluation.py

עליכם לשלוח קבצים אלה עם הקוד והערות שלכם. נא לא לשנות את הקבצים האחרים או לשלוח אף אחד מהקבצים המקוריים מלבד קבצים אלה. בראש כל אחד מקבצים אלו נא לכתוב את שם הסטודנט ות.ז.

בנוסף לקבצים אלו עליכם להגיש דו"ח העונה על שאלות בתרגיל בקובץ בשם report_<id>.pdf, כאשר id יהיה ת.ז. של הסטודנט וכן קובץ פרטים אישיים בשם detail.txt והוא יכיל את שם הסטודנט בשורה הראשונה ות.ז. בשורה השנייה.

ההגשה דרך ה Moodle בלבד!! לא יתקבלו הגשות ב classroom משום סיבה.

חלק ראשון – נתונים

בקובץ `data.py` נתונה לכם הפונקציה `watch_data_info(data)`. העזרו בפונקציה כדי להבין את קבצי הנתונים שצורפו לתרגיל.

נתייחס כעת לקובץ הדירוגים:

1. כמה משתמשים יחודיים דרגו את הסרטים? כמה סרטים יחודיים דורגו? כמה דירוגים קיימים בקובץ שניתן?
2. מהו מספר הדירוגים המינימלי והמקסימלי שניתן לסרט?
3. מהו מספר הדירוגים המינימלי והמקסימלי שמשתמש דירג?

את המימוש לקבלת התשובות הנ"ל יש לכתוב בפונקציה `print_data(data)`. (הפונקציה מקבלת את קבצי הנתונים המתאימים כ `tuple` במשתנה `data`) מהי התפלגות הדירוגים? הציגו גרף מתאים.

את המימוש לקבלת הגרפים הנ"ל יש לכתוב בפונקציה `plot_data(data, plot = True)` המקבלת את קבצי הנתונים המתאימים כ `tuple` במשתנה `data`, ומשתנה `plot` המקבל כבירור מחדל `true` – ובמצב זה הפעילו את פונקציה `plot.show()` עבור הגרפים השונים

על כל השאלות הנ"ל והגרפים שהתבקשתם להציג, יש לענות בקובץ הדו"ח אותו אתם מגישים.

חלק שני - collaborative-filtering

בקובץ `collaborative_filtering.py` ישנה מחלקה למימוש מערכת המלצה מסוג CF. עבור בניית מטריצת החיזוי, נשתמש בשדה בממד הדמיון `cosine` - מדד המחשב את הזווית בין שני וקטורים במרחב:

$$\cos(x, y) = \frac{x \cdot y^T}{\|x\| \cdot \|y\|} = \frac{\sum_{i=1}^n x_i \cdot y_i^T}{\sqrt{\sum_{i=1}^n (x_i)^2} \sqrt{\sum_{i=1}^n (y_i)^2}}$$

ממשו את שתי הפונ:

`create_user_based_matrix(data)`

`create_item_based_matrix(data)`

המופיעות בקובץ את מטריצת החיזוי עבור `CF-user based` – `CF-item based` בהתאמה. הפונקציה מקבלת את המשתנה `data` – המכיל את קבצי הנתונים המתאימים כ `tuple`. הפונקציה יבנו מטריצות חיזוי לתוך המשתנים `user_based_matrix` ו-`item_based_matrix`.

טיפ – לצורך נוחות, מומלץ לבצע המרה לשדות `movieId` ו-`userId` למספרים עוקבים מ 0 עד למספר הערכים היחודיים של אותו שדה.

שימו לב – לא נבקש יותר מ 10 המלצות עבור משתמש.

בנוסף, ממשו פונקציה `predict_movies(user_id, k, is_user_based)` המקבלת 3 משתנים: `user_id` – המכילה מספר ID של משתמש בעבורו נרצה לבצע חיזוי `k` – מספר הסרטים המומלצים שנרצה לקבל.

`is_user_based` – המקבלת ערך `true` עבור חיזוי מסוג `CF-user based` (ברירת המחדל) וערך `false` עבור חיזוי מסוג `CF-item based`

הפונקציה תחזיר את רשימת שמות הסרטים המומלצים עבור אותו משתמש.

אתם רשאים להוסיף פונקציות ומשתנים בקובץ `collaborative_filtering.py` כרצונכם, מבלי לשנות את הקוד הקיים.

4. מה יהיה החיזוי עבור משתמש "283225" ב `cf_user_based` - $k=5$? האם התוצאות תואמות את אופי המשתמש? הציגו את תוצאות החיזוי עבור משתמש זה והסבירו את תשובתכם בדוח.
5. בקובץ `collaborative_filtering.py` קיימת פונ' `create_fake_user(rating)` המקבלת את נתוני הדירוגים. בעזרת פונ' זו, הוסיפו משתמש פקטיבי (`userId = 283238`) לנתוני הדירוגים וחזו עבורו רשימת המלצות. מה תהיה ההמלצה הטובה ביותר עבור משתמש זה ב `cf_user_based` - $k=5$? הסבירו את המימוש ואופי המשתמש בדוח והציגו את תוצאות החיזוי עבור משתמש זה.

חלק שלישי – הערכות

קובץ `test.csv` הינו קובץ טסט של דירוגי משתמשים, בעזרתו תעריכו את מערכות ההמלצה CF השונות.

מדדי הערכה בהם נשתמש הם:

Precision@10 - מדד לדיוק ב- k , הוא החלק הרלוונטי של הפריטים המומלצים בערכת ה- k -top

במקרה שלנו נתייחס לדירוגים גבוהים בערכי 4 ו-5 כאשר $K = 10$

$$P@k = \frac{\#hits}{k}$$

ARHA - מדד אשר לוקח בחשבון רק היכן התוצאה הרלוונטית מתרחשת. אנו מקבלים יותר קריטי על המלצה על פריט שבו משתמש מדורג בראש הדירוג מאשר בתחתית הדירוג. גבוה יותר זה יותר טוב.

$$ARHR = \frac{1}{\#users} \sum_{i=1}^{\#hits} \frac{1}{pos_i}$$

RMSE - מדד שנמצא בשימוש תכוף להבדלים בין ערכים חזויים על ידי מודל או אומדן לבין הערכים האמיתיים ומוגדר כך:

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (Predicted_i - Actual_i)^2}{N}}$$

בקובץ `evaluation.py` מופיעות הפונ' הבאות::

```
precision_10( test_set, cf, is_user_based)
ARHA( test_set, cf, is_user_based)
RSME( test_set, cf, is_user_based)
```

ממשו כל אחת מהפונ' הנ"ל של מחלקת `collaborative_filtering`, המקבלת את ה `test_set`, המתשנה `is_user_based` - `cf` שבעבור ערך `true` תבצע חיזוי עבור מסוג `CF-user based` (ברירת המחדל) וערך `false` עבור חיזוי מסוג `CF-item based`

הפונ' תדפיס את ערך ההערכה שחישבה (כפי שמופיע בתוך הפונ')

גם כאן אתם רשאים להוסיף פונ' ומשתנים כרצונכם, מבלי לשנות את הקוד הקיים.

6. הוסיפו בדוח את הטבלה הנ"ל המכילה את מדדי ההערכה שחישבתם.

	precision@10	ARHR	RMSE
User based			
Item based			

בהצלחה רבה!