בס"ד

תרגיל 3 – מערכות המלצה

בתרגיל זה נרצה לתרגל את אלגוריתמי מערכות המלצה השונים שנלמדו בכיתה:

- Non-personalized
- Contact based filtering
- Collaborative filtering

נתונים לכם הקבצים הבאים:

books – המכיל נתוני ספרים.

users – המכיל נתונים על המשתמשים.

ratings – המכיל דירוגים של המשתמשים על הספרים.

tags - המכיל נתוני תיוגים שונים.

book_tags - המכיל תיוגים על הספרים.

test - זהו קובץ המכיל דירוגים של משתמשים על הספרים לצורך הערכת האלגוריתמים.

<u>חלק א – Non-personalized</u>

בחלק זה נממש אלגוריתם המלצה non-personalized

1. מבין כל הנתונים הקיימים לכם בקבצים השונים, בחרו את הערכים ליצירת מדד הדמיון עבור מערכת המלצה זו. כתבו בדוח והסבירו את בחירתכם. (הראנו בכיתה אופציות שונות. אין לבחור מדד נאיבי!)
2. כתבו פו' כללית לקבלת המלצות non-personalized:
   get_simply_recommendation(k)
   המקבלת ערך K המייצג כמה פריטי המלצה להחזיר ותחזיר את K הספרים המומלצים.
   ציינו בדוח את רשימת 10 הספרים המומלצים, ה id של הספר ואת הציון שהוא קיבל.
3. כתבו פו' לקבלת המלצות non-personalized המזהה את מקום מגורי המשתמש
   get_simply_place_recommendation(place, k)
   המקבלת את מקום מגוריו של המשתמש וערך K המייצג כמה פריטי המלצה להחזיר ותחזיר את K הספרים המומלצים בהתאם למיקום המשתמש.
   ציינו בדוח את רשימת 10 הספרים המומלצים עבור משתמש הגר באוהיו (Ohio), ה id של הספר ואת הציון שהוא קיבל.
4. כתבו פו' לקבלת המלצות non-personalized המזהה את גיל המשתמש:

   get_simply_age_recommendation(age, k)

   המקבלת את גיל המשתמש וערך K המייצג כמה פריטי המלצה להחזיר ומחזירה את רשימת הספרים בהתאם לגילו.
   בהינתן גיל, המערכת תחשב המלצות בטווח הגילאים x1-y0 (למשל: אם התקבל הגיל 55 יתקבלו המלצות לטווח הגילאים 51-60)
   ציינו בדוח את רשימת 10 הספרים המומלצים עבור משתמש בן בטווח הגילאים בין 28, ה id של הספר ואת הציון שהוא קיבל.

חלק ב - Collaborative filtering

בחלק זה נממש את אלגוריתם Collaborative filtering ע"פ user-based כפי שנלמד בכיתה עם מטריקת הדמיון cosine.

5. ממשו את האלגוריתם ע"פ user-based. לצורך בניית מטריצת הדמיון השתמשו בקובץ rating
כתבו פו' לבניית מטריצת החיזוי:

build_CF_prediction_matrix(sim)

המשתנה sim מייצג את מדד הדמיון (-עבור בחירת מדד הדמיון כפי שמוגדר בהמשך.)
- יקבל ערך cosine עבור מדד דמיון cosine
- יקבל ערך euclidean עבור מדד דמיון euclidean
- יקבל ערך jaccard עבור מדד דמיון jaccard

6. כתבו פו' לקבלת המלצות

get_CF_recommendation(user_id, k)

המקבלת id של משתמש ומשתנה K המייצג את אורך רשימת ההמלצות המוחזרת.
הפו' תחזיר את K הספרים המומלצים עבור אותו משתמש תוך שימוש במטריצת הפרדיקציה שנבנתה בסעיף 5.

מטריקות דמיון

7. שנו את מטריקת הדמיון לכל אחד ממטריקות הדמיון המצויינות להלן:

Euclidean – מדד זה מחשב את המרחק בין שני וקטורים במרחב

$$s(q,x) = \|q-x\| = \left[\sum_{i=1}^{d}(q_i - x_i)^2\right]^{\frac{1}{2}}$$

Jaccard - מדד זה מחשיב רק את מספר הפריטים המדורגים על ידי שני משתמשים במקום את הדירוגים. (מה שמציין שככל שהפריטים מדורגים יותר, כך דומה יותר).

$$\mathrm{Sim}\Big(u,v\Big)^{\mathrm{Jaccard}} = \frac{|I_u \cap I_v|}{|I_u \cup I_v|}$$

חלק ג – contact based filtering

בחלק זה נממש את אלגוריתם ההמלצות contact based filtering

8. ממשו את אלגוריתם contact based filtering.
כתבו פו' לבניית מטריצת הדמיון:

build_contact_sim_metrix()

ציינו בדוח מהם הפיצ'רים בהם בחרתם להשתמש.
השתמשו בפו' דמיון cosine

9. ממשו פו' לקבלת המלצות

get_contact_recommendation(book_name, k)

המקבלת שם של ספר וערך K המייצג כמה פריטי המלצה להחזיר ותחזיר את K הספרים המומלצים.

10. עבור הספר "Twilight" – מהם רשימת הספרים אותם החזיר המודל? ציינו בדוח.

<u>חלק ד - מדדי הערכה</u>

בחלק זה נממש פו' מדדי הערכה שונים להערכת ההמלצות.

Precision@k - מדד לדיוק ב- k, הוא החלק הרלוונטי של הפריטים המומלצים בערכת ה- top-k במקרה שלנו נתייחס לדירוגים גבוהים בערכי 4 ו-5.

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

ARHA - מדד אשר לוקח בחשבון רק היכן התוצאה הרלוונטית מתרחשת. אנו מקבלים יותר קרדיט 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:

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

11. כתבו פו' עבור כל אחד ממדדי ההערכה שהוגדרו לעיל שתחשב עבור עבור CF-user based תוך שימוש בנתונים מקובץ test המצורף וערך K=10

שמות הפו':

precision_k(k)

ARHR(k)

RMSE()

כשבכל פעם תשתמשו במדד דמיון אחר:

- cosine
- euclidean
- jaccard

הוסיפו בדוח טבלה בפורמט הנ"ל המכילה את מדדי ההערכה שחישבתם:

	precision_k	ARHR	RMSE
cosine			
euclidean			
jaccard			

12. הסבירו את התוצאות שקיבלתם.
13. **בונוס**: 18 הסטודנטים עם תוצאות הערכה הטובות ביותר יקבלו בונוס של 5 נק' (2 סטודנטים מכל מטריקת הערכה\פו' דמיון)

<u>הוראות הגשה:</u>

התרגיל ביחידים בלבד!

את התרגיל יש להגיש דרך מערכת ה moodle

את פו' הפקת ההמלצות וההערכות יש להגיש בקובץ ex3.py. וכמובן כל קובץ קוד נוסף שיש.

הקפידו על קוד ברור, קריא ומתועד! עליכם לתעד כל חלק שאינו טריוויאלי בקוד שלכם. בפרט, אם התשמשתם בקוד שנמצא ברשת וביצעתם בו שינויים, עליכם לתעד זאת.

קובץ בשם *txt.stnemeriuqer* שיכיל כל חבילה חיצונית שאינה מותקנת כחלק מ-*nohtyP adnocanA* . על אחריותכם לוודא כי ניתן להתקין כל חבילה כנ"ל באמצעות הרצת השורה: *pip install -r requirements.txt*

בנוסף עליכם להגיש דו"ח בקובץ בשם report.pdf המכיל את ההסברים והתוצאות אותם ביקשתם להפיק.

אין להעתיק את הקבצים המסופקים לכם אל תוך תיקיית ההגשה. הניחו כי קבצים אלו יהיו זמינים בעת בדיקת התרגיל.

תאריך הגשה אחרון: 19.1.21 בשעה 23:59

בהצלחה רבה!