מטלה: Book-Crossing — סינון שיתופי (Collaborative Filtering) מקבילי עם Apache Spark

מטרה

בנו מערכת זרימת נתונים (pipeline) מקצה לקצה לצורך המלצות מקבילית על בסיס מאגר הנתונים Book-Crossing החל מקובץ MySQL dump גולמי, דרך קליטת נתונים וניקוי, חקירת הנתונים, אל אימון מודלים סינון שיתופי (Collaborative Filtering — CF) מבוסס משתמשים ומבוסס פריטים תוך שימוש באלגוריתם מקבילי. לאחר מכן יש להעריך את ביצועי המודל באמצעות מטריצת בלבול, וליישם מודל כללי אסוציאציה להמלצות עבור משתמשים עליהם אין מספיק מידע (שהיסטוריית הדירוגים שלהם דלילה מכדי לאפשר CF אמין).

התרגיל מחולק על פני **חמישה קבצי Python**. כל קובץ חייב לכתוב **קובץ לוג** הכולל: שם הסקריפט וחותמת זמן, מימדי כל קלט ופלט משמעותיים (שורות × עמודות), וסיכום של נתונים שהוסרו או שונו. כל שלב חישובי חייב לכלול מדידת **זמן כולל** (wall-clock time); ערכים אלה ידווחו ב-README.

זרימת הנתונים

```
BX-dump.sql   (MySQL dump: CREATE TABLE + INSERT INTO)
      |
      ▼
part_a.py    ← populate database, clean schema                    [Part A]
      |
      ▼
part_b.py    ← statistics, histograms, top-10 lists               [Part B]
      |
      ▼
part_c.py    ← user similarity, parallel CF model, k-fold,
      |         confusion matrix, rating-count threshold           [Part C]
      |
      ▼
part_d.py    ← item similarity, item-based CF, evaluation          [Part D]
      |
      ▼
part_e.py    ← cold-start identification, association rules,
                recommendations for sparse users                   [Part E]
```

מאגר הנתונים

הורידו את מאגר Book-Crossing מתיבת ההגשה ה-dump מכיל שלוש טבלאות: BX-Books, BX-Users, BX-Book-Ratings

א. קליטת נתונים — `part_a.py`

משימה A1: אכלוס מסד הנתונים

טענו את ה-MySQL dump למסד נתונים לבחירתכם. אם אינכם מעוניינים להתקין MySQL, ניתן למפות את הקבצים ל- SQLite3 או Oracle לפי הנלמד בשיעור הדגמנו. תעדו את פעולות העמסת מסד הנתונים (הפקודות) ב-README.

משימה A2: ניקוי נתונים

בצעו ניקוי ראשוני ישירות במסד הנתונים (או באמצעות שאילתות SQL). לכל הפחות:

- ISBN חייב להיות בן 13 תווים בדיוק (או ISBN-10 תקני); הסירו רשומות שאינן תואמות.
- Book-Author ו-Book-Title חייבים להיות לא-ריקים; הסירו רשומות שבהן אחד מהם ריק או null.
- הסירו שורות כפולות מכל טבלה (אותו מפתח ראשי המופיע יותר מפעם אחת).
- בטבלת הדירוגים, הסירו שורות שה-ISBN שבהן אינו קיים בטבלת הספרים, ושורות שה-User-ID שבהן אינו קיים בטבלת המשתמשים. *אל תבצעו השלמה (imputation) — הסירו שורות אלה.*
- הסירו זוגות כפולים של (User-ID, ISBN) בטבלת הדירוגים, תוך שמירת המופע הראשון.

לאחר שלבי הניקוי, רשמו ב-לוג את מספר השורות שהוסרו ואת מספר השורות שנותרו. בצעו JOIN ודווחו את מספר השורות לאחריו, כל אלה ב**קובץ הלוג:** `part_a.log`

ב. ניתוח נתונים עיוני `part_b.py`

ניתן לחשב את הסטטיסטיקות ישירות ממסד הנתונים המנוקה.

משימה B1: סטטיסטיקות בסיסיות

- סה"כ משתמשים, ספרים ודירוגים ייחודיים (unique).
- דירוגים בעלי ערך 0 (מרומז) לעומת 1–10 (מפורש): מספר דירודים קיימים ואחוז דירוגים קיימים.
- התפלגות גיל המשתמשים: מינימום, מקסימום, ממוצע, חציון, מספר גילאים חסרים/לא תקינים.

משימה B2: היסטוגרמות דירוגים

הכינו ושמרו את התרשימים הבאים *וכן* את טבלאות ההיסטוגרמות המקבילות להם ב-README:

- דירוגים למשתמש: ציר y מספר הדירוגים בסולם לוגריתמי, ציר x דרגת המשתמש (המשתמש הפעיל ביותר מדורג 1, המשתמש השני הפעיל ביותר מדורג 2, ...) בסולם לוגריתמי; `hist_ratings_per_user.png`
- דירוגים לספר: ציר y מספר הדירוגים בסולם לוגריתמי, ציר x דרגת המספר (הספר הפופולרי ביותר ביותר מדורג 1, ...) בסולם לוגריתמי; `hist_ratings_per_book.png`
- התפלגות ערכי הדירוג (ערכים 0..10) `hist_rating_values.png`

משימה B3: רשימות Top-10

תוך שימוש בדירוגים מפורשים בלבד (Book-Rating ≥ 1):

- 10 הספרים המובילים לפי מספר דירוגים מפורשים שקיבלו (כותרת, מחבר, מספר דירוגים) `top10_books.png`
- 10 המשתמשים המובילים לפי מספר דירוגים מפורשים שנתנו (User-ID, מספר) `top10_users.png`

ב**קובץ הלוג:** `part_b.log` יופיעו שם הסקריפט, חותמת זמן, מימדי כל תוצאה צוברת, זמן כולל לכל שלב, וטבלת הערכים שהזינו כל איור

ג. סינון שיתופי מבוסס משתמשים UBCF `part_c.py`

משימה C1: מטריצת משתמשים ופריטים (User-Item matrix)

עבדו עם דירוגים מפורשים בלבד (Book-Rating ≥ 1). החילו את חוקי הניקוי ממשימה A2 והרכיבו מטריצת המשתמשים ופריטים (UI-matrix) ב-Spark. רשמו ב-לוג את מימדי הטבלה ב-Spark.

משימה C2: מטריצת דמיון בין משתמשים

חשבו מטריצת דמיון בין משתמשים בעזרת פונקציית דימיון Cosine Similarity. מאחר שהמטריצה המלאה גדולה מכדי לחשב, שמרו רק את 20 המשתמשים הדומים ביותר לכל משתמש.

משימה C3: אימון מודל CF מקבילי

אמנו מודל סינון שיתופי (CF) מקבילי על ה-(UI-matrix). כל אלגוריתם מקבילי יתקבל; בשיעור ראינו את ALS של Spark MLlib.

משימה C4: אימות k-fold

העריכו את המודל באמצעות אימות k-fold כאשר k ≥ 3. דווחו על ממוצע וסטיית תקן של RMSE על פני כל הבחירות של אוכלוסיית אימון ובדיקה (folds).

משימה C5: הערכת ביצועים

השוו את הדירוגים החזויים לדירוגים האמיתיים בכל התאים *הנצפים* של ה-(UI-matrix). קבצו את הערכים החזויים ואת הערכים האמיתיים לשלוש מחלקות: **נמוך** (1–3), **בינוני** (4–6), **גבוה** (7–10). בנו מטריצת בלבול 3×3 וחשבו accuracy, recall, ו-F1 לכל מחלקה.

חלקו את המשתמשים לקבוצות לפי מספר הדירוגים שלהם (למשל 1–4, 5–9, 10–19, 20–49, 50+). לכל קבוצה, חשבו את מטריצת הבלבול ואת שיעור התחזיות **על האלכסון או בסמוך לו**. שרטטו שיעור זה מול מינימום מספר הדירוגים של הקבוצה. על בסיס ניתוח זה, קבעו באופן אמפירי את מינימום מספר הדירוגים למשתמש (K_{user}) ולספר (K_{book}) שמתחתיו איכות החיזוי ירודה. שמרו ערכים אלה לשימוש בחלק ה.

משימה C6: המלצות Top-10 למשתמשים לדוגמה

בחרו חמישה משתמשים אקראיים מבין אלו שיש להם k ≥ K_{user} דירוגים. לכל אחד, אחזרו את 10 הספרים המומלצים ביותר שהמשתמש טרם דירג. שמרו זאת ב- `user_cf_recommendations.csv` .

ב**קובץ לוג** `part_c.log` יופיעו שם הסקריפט, חותמת זמן, מימדי המטריצה בכל שלב, RMSE לכל חלוקה (fold), מטריצת בלבול, ספי הדירוגים שנגזרו, זמנים כוללים.

ד. סינון שיתופי מבוסס פריטים IBCF `part_d.py`

משימה D1: מטריצת דמיון פריטים

חשבו מטריצת דמיון לפי Cosine Similarity בין ספרים על בסיס וקטורי הדירוג של הספרים. שמרו את 20 הספרים הדומים ביותר לכל ספר. השוו את הזמן הכולל לזה של משימה C2.

משימה D2: מודל CF מבוסס פריטים

אמנו מודל CF מבוסס פריטים על אותה מטריצה מנוקה מחלק ג. השוו את הזמן הכולל לזה של משימה C3.

משימה D3: הערכה והשוואה

החילו את אותה מתודולוגיית מטריצת בלבול 3×3 ממשימה C5 על המודל מבוסס הפריטים. דווחו על שיעור האלכסון והשוו אותו זה לצד זה עם התוצאה מבוסס המשתמשים. מציינו את שיעור האלכסון של *המודל הטוב יותר* מבין השניים.

משימה D4: Top-10 עשרת הספרים הדומים לפריטים לדוגמה

בחרו חמישה ספרים אקראיים מבין אלו שיש להם $k \geq K_{book}$ דירוגים. לכל אחד, רשמו את 10 הספרים הדומים ביותר ממטריצת דמיון הפריטים. שמרו זאת ב- `item_cf_recommendations.csv` .

ב**קובץ לוג:** `part_d.log` יופיעו שם הסקריפט, חותמת זמן, מימדי הקלט, מטריצת בלבול, שיעור האלכסון, זמנים.

ה. כללי אסוציאציה למשתמשי ללא מידע מספקי `part_e.py`

משימה E1: זיהוי משתמשים

בהתבסס על K_{user} ממשימה C5, חלקו את כלל המשתמשים ל: **משתמשים חמים** ($\leq K_{user}$ דירוגים) ו**משתמשים קרים** ($< K_{user}$ דירוגים). דווחו על גודל (ואחוז) של כל קבוצה.

משימה E2: בניית סלי טרנסטקציות

לצורך כריית כללי אסוציאציה, התייחסו לקבוצת הספרים שכל משתמש דירג כסל עסקה אחד (שדירוג ספר מרמז על רכישתו). שמרו רק ספרים העומדים בסף K_{book}.

משימה E3: כריית כללי אסוציאציה

הפעילו את FP-Growth (המימוש המקבילי של Spark MLlib) על סלי העסקאות. קבעו סיפים מזעריים support ו- confidence כך שתתקבל קבוצת כללים בגודל סביר (יעד: אלף כללים). דווחו על הערכים שנבחרו ועל מספר הכללים שנוצרו. הציגו את 20 הכללים המובילים לפי lift (בתצוגה: הכלל, support, confidence, lift).

משימה E4: המלצות למשתמשים "קרים"

לכל משתמש איתחול, אתרו כללים שהאנטסדנט שלהם הוא תת-קבוצה של הספרים שהמשתמש דירג. דרגו את הספרים המוצעים (שטרם דורגו) לפי confidence × lift והחזירו את 10 הראשונים. החילו זאת על מדגם אקראי של 10 משתמשים **"קרים"**. שמרו ב- `apriori_recommendations.csv` .

ב**קובץ הלוג** `part_e.log` יופיעו שם הסקריפט, חותמת זמן, מספר הסלים, מספר הכללים, פיצול משתמשים חמים/קרים, זמנים.

ו. אילוצים

- אין להשלים נתונים חסרים או לא תואמים. הסירו שורות לא תקינות.
- שלב אימון מודל ה-CF חייב להשתמש באלגוריתם מקבילי הפועל בתשתית Spark.
- K_{user} חייב להיגזר באופן אמפירי מניתוח מטריצת הבלבול; אין לקבוע ערך זה ידנית.

ז. בונוס: 3 נקודות

- **(+1)** הפעילו מימוש סדרתי של אפריורי (Apriori) ואת FP-Growth המקבילי של Spark על אותה קבוצת סלים. שרטטו זמן כולל מול גודל מדגם (25%, 50%, 75%, 100%) עבור שני השיטות. פרשו את התנהגות הסקלה.
- **(+2)** לבעיית הסיווג הבינארי של חיזוי דירוגים גבוהים לעומת בינוניים/נמוכים, שרטטו עקומת ROC אחת לכל קבוצת ספירת דירוגים (כהגדרה במשימה C5) על אותם צירים. חשבו AUC לכל קבוצה וקשרו את הממצאים לסף K_{user}.

ח. תבנית README

הגישו קובץ בשם `README.md` על פי מבנה זה:

```
# ID1 Student_Name_1
# ID2 Student_Name_2
# team: team-name
# date: YYYY-MM-DD

# Part A
Database choice: [SQLite3 / MySQL / Oracle] — [one sentence rationale]

Run instructions:
  python part_a.py --input BX-dump.sql --db books.db
  python part_b.py --db books.db
  python part_c.py --db books.db
  python part_d.py --db books.db
  python part_e.py --db books.db

# Part B
## Statistics
Total users   : ...
Total books   : ...
Total ratings : ...
Implicit (0)  : ...   (  ...%)
Explicit (1-10): ...  (  ...%)

## Histogram — ratings per user
Bin (ratings)   | Frequency
----------------|----------
1               | ...
2-4             | ...
5-9             | ...
10-19           | ...
20-49           | ...
50+             | ...

## Histogram — ratings per book
Bin (ratings)   | Frequency
----------------|----------
1               | ...
2-4             | ...
5-9             | ...
10-19           | ...
20+             | ...

## Histogram — rating value distribution
Rating | Frequency
```

```
-------|----------
0      | ...
1      | ...
...
10     | ...

## Timing — Part B
Step                              | Wall-clock (s)
----------------------------------|---------------
B1 distinct counts                |
B2 ratings-per-user aggregation   |
B2 ratings-per-book aggregation   |
B3 top-10 books                   |
B3 top-10 users                   |

# Part C
K_user (empirically derived)  : ...
K_book (empirically derived)  : ...

## Data-flow summary
Stage                                  | Rows     | % of start
---------------------------------------|----------|----------
Explicit ratings (start)               |          | 100%
After Gate 1 (ISBN/User-ID cleaning)   |          |
After Gate 2 (K_user filter)           |          |

## k-fold results (k = ...)
Fold | RMSE
-----|-----
1    |
...  |
Mean RMSE : ...
Std  RMSE : ...

## Confusion matrix — User-Based CF (Low / Medium / High)
            Pred Low | Pred Med | Pred High
Actual Low  |        |          |
Actual Med  |        |          |
Actual High |        |          |

Diagonal fraction: ...%

## Timing — Part C
Step                              | Wall-clock (s)
----------------------------------|---------------
C1 User-Item matrix build         |
C2 User similarity                |
C3 Model fit                      |
C4 k-fold (total)                 |
C5 Prediction vs ground-truth     |

# Part D
## Confusion matrix — Item-Based CF (Low / Medium / High)
            Pred Low | Pred Med | Pred High
Actual Low  |        |          |
Actual Med  |        |          |
Actual High |        |          |
```

```
Diagonal fraction: ...%

## User-CF vs Item-CF comparison
Metric                | User-CF | Item-CF
---------------------|---------|--------
Model fit time (s)   |         |
Prediction time (s)  |         |
Diagonal fraction    |         |

## Timing — Part D
Step                                | Wall-clock (s)
-----------------------------------|---------------
D1 Item similarity                  |
D2 Model fit                        |

# Part E
Cold-start users : ... ( ...%)
Warm users       : ... ( ...%)
K_user used      : ...
FP-Growth minSupport    : ...
FP-Growth minConfidence : ...
Rules generated  : ...

## Timing — Part E
Step                                | Wall-clock (s)
-----------------------------------|---------------
E1 Cold-start partitioning          |
E3 FP-Growth fit                    |
```

ט. קבצי הגשה

קובץ	תיאור
README.md	תבנית מלאה לעיל
part_a.py	אכלוס מסד הנתונים וניקוי
part_b.py	ניתוח עיוני
part_c.py	CF מבוסס משתמשים
part_d.py	CF מבוסס פריטים
part_e.py	כללי אסוציאציה ומשתמשי איתחול
part_a.log	לוג הרצה — חלק א
part_b.log	לוג הרצה — חלק ב
part_c.log	לוג הרצה — חלק ג
part_d.log	לוג הרצה — חלק ד
part_e.log	לוג הרצה — חלק ה

קובץ	תיאור
hist_ratings_per_user.png	היסטוגרמה — חלק ב
hist_ratings_per_book.png	היסטוגרמה — חלק ב
hist_rating_values.png	היסטוגרמה — חלק ב
top10_books.png	תרשים — חלק ב
top10_users.png	תרשים — חלק ב
user_cf_recommendations.csv	המלצות — חלק ג
item_cf_recommendations.csv	המלצות — חלק ד
apriori_recommendations.csv	המלצות — חלק ה