



מערכות המלצה בשירותי סטרימינג
אלכסנדרה חליבנקו
בהנחיית ד"ר מיה חרמן
2023א

1. מבוא

1.1 מה זה למידה לא מפוקחת

1.2 מה זה מערכות המלצה

1.3 מדוע מערכות המלצה

1.4 אתגרים וקשיים בדרך

2. איך מועברת מידע למערכות המלצה

2.1 תיאור כללי

2.2 שיטות העברה

1. Explicit ratings - דירוגים מפורשים

2. Implicit ratings – דירוגים מרומזים

3. Product similarity – דמיון המוצר

4. User similarity - דמיון המשתמש

5. מדידות דמיון

3. גישה אלגוריתמית

3.1 תיאור כללי

3.2 collaborative filtering

3.3 content-based filtering

4. מערכות המלצה בשירותי סטרימינג

4.1 סטרימינג – Netflix, prime video

4.2 מודל

4.3 תהליך למידה

4.4 אלגוריתם

5. סיכום

6. מקורות

1. מבוא

בתקופה האחרונה, טכנולוגיות חדשות כמו חיישנים, אחסון, מחשבים ורשתות תקשורת עם התפתחותם מהירה, גרמו לצמיחה מתמשכת של הנתונים. בעזרת טכנולוגיית האינטרנט והמחשוב המתקדם, גישה למידע הפכה לקלה יותר עבור המשתמשים. אולם, עם הצמיחה המתמשכת של הנתונים, משתמשים מתקשים למצוא מידע שמתאים לצרכים שלהם בזמן, וכך יוצרת רוב המידע המיותר והלא רלוונטי עומד בדרך של הבחירות הנכונות של המשתמשים. עם זאת, מערכות המלצה הפכו לכלי יעיל שמסייע בסינון המידע הלא רלוונטי ומאפשר למשתמשים למצוא את המידע הרלוונטי להם במהירות ובקלות.

עומס מידע מתייחס למצב בו אדם או ארגון מתמודדים עם שפע של מידע, המקשה על קבלת החלטות או פעולה. בעיה זו הופכת נפוצה יותר ויותר בעידן הדיגיטלי של היום, שבו אנו מוצפים בנתונים ממקורות שונים כמו מדיה חברתית, ערוצי חדשות ופלטפורמות קניות מקוונות.

מערכות המלצה יכולות לעזור לטפל בבעיית עומס המידע על ידי מתן המלצות מותאמות אישית המבוססות על העדפות המשתמש והתנהגותו בעבר. על ידי סינון מידע לא רלוונטי והצגת האפשרויות הרלוונטיות בלבד, מערכות ממליצים יכולות לחסוך זמן ולשפר את חווית המשתמש הכוללת.

בהקשר של שירותי סטרימינג, נפח התוכן הזמין יכול להקשות על המשתמשים לגלות תוכניות וסרטים חדשים שמתאימים לתחומי העניין שלהם. מערכות ממליצים יכולות לנתח את היסטוריית הצפייה, הדירוגים ושאלות החיפוש של המשתמש כדי לספק המלצות מותאמות אישית לתוכן חדש, להגדיל את הסבירות שהמשתמש ימשיך להשתמש בשירות הסטרימינג ויגלה תוכן חדש שהוא נהנה ממנו.

1.1 מה זה למידה לא מפוקחת

כפי שהשם מרמז, למידה לא מפוקחת היא טכניקת למידת מכונה שבה מודלים אינם מפוקחים באמצעות דאטה סט אימון, כלומר הדאטה סטים אינם מסווגים ומתויגים. במקום זאת, המודלים עצמם מוצאים את הדפוסים והתבונות הנסתרות מהנתונים. אפשר להשוות את זה ללמידה שמתרחשת במוח האנושי תוך כדי לימוד דברים חדשים.

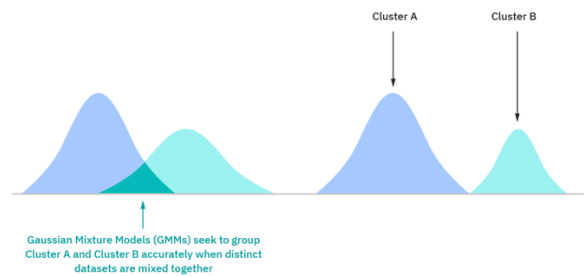
המטרה של למידה לא מפוקחת היא למצוא את המבנה הבסיסי של דאטה סט, לקבץ את הנתונים לפי קווי דמיון ולייצג את דאטה סט בפורמט דחוס, שזה נקראה clustering, או לקבוע כיצד הנתונים מופצים במרחב ידוע כי הערכת הצפיפות.

למידה לא מפוקחת משמשת לשלושה משימות עיקריות: אשכולות (clustering), הורדת ממד (dimensionality reduction) ולימוד חוקיות אסוציאטיביות (association rule learning).

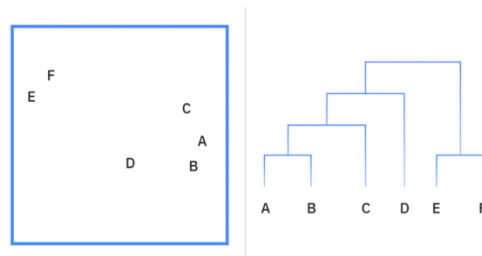
אשכולות (clustering)

היא טכניקה כריית נתונים המקבצת את הנתונים ללא תווית על סמך הדמיון או ההבדלים ביניהם. אלגוריתמי אשכולות משמשים לעיבוד אובייקטי נתונים גולמיים ולא מסווגים לקבוצות המיוצגות על ידי מבנים ודפוסים במידע. ניתן לסווג את אלגוריתמי אשכולות לכמה סוגים :

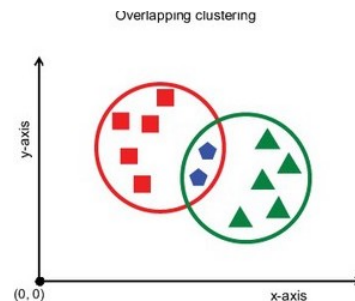
- **הסתברותי** - כל נקודת נתונים שייכת לכל קבוצות האשכולות עם דרגות שונות של חברות מ-0 עד 1. לדוגמה, במערך נתונים עם אשכולות A, B, C ו-D, ניתן לשייך נקודת נתונים לכל האשכולות עם דרגה $A = 0.5$, $B = 0.1$, $C = 0.4$ ו- $D = 0$. במקום שיוך מוגדר של נקודת נתונים עם אשכול אחד, אשכול מעורפל משייך הסתברות לחברות לכל האשכולות.



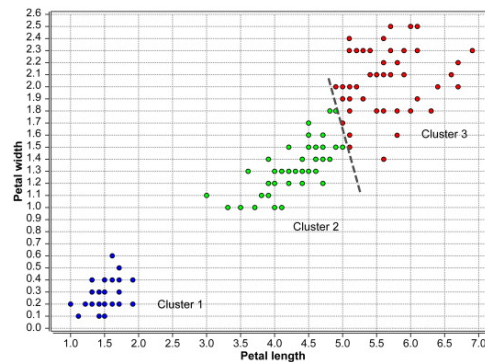
- **היררכי** - נקודות הנתונים שלו מבודדות בתחילה כקבוצות נפרדות, ולאחר מכן הן מתמזגות יחדיו באופן איטרטיבי על בסיס דמיון עד להשגת אשכול אחד.



- **חופף** - קבוצות האשכול אינן בלעדיות וכל אובייקט נתונים עשוי להיות שייך ליותר מאשכול אחד. אלה ידועים גם בתור אשכולות מרובים. לדוגמה, ניתן לקבץ לקוחות של חברה במקבץ לקוחות עתירי רווח ובאשכול לקוחות עתירי נפח בו זמנית.

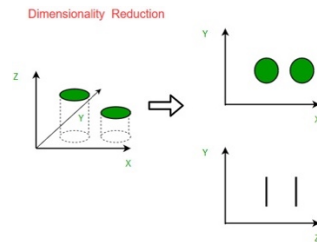


- **"יחודי" - כל אובייקט נתונים שייך לאשכול בלעדי אחד מיצג באיור מתחת. זהו הסוג הנפוץ ביותר של אשכול.**



הורדת ממד (dimensionality reduction)

תהליך הורדת ממד הוא תהליך של צמצום מספר משתנים מתוך כלל המידע, על מנת לשמר כמה שיותר מהמשמעות של המידע עצמו. פרשנות של הורדת ממד נראה באיור הבא, כך שפה בוצע הורדה מתלת ממד לדו-ממד.



ישנן כמה שיטות להורדת ממד :

- **בחירת מאפיינים** – בגישה הזו ננסה לבחור תת-קבוצות של משתנים מתוך כלל המשתנים, ולקבוצה הזו נקרא מאפיין (feature). המטרה של השיטה הזאת זה לבחור המאפיינים אופטימליים עבור המידע.
- **הטלת מאפיינים** – בשיטה הזאת ממירים את המידע המאופיין כמרחב בעל מספר ממדים לדאטה המיוצג במרחב עם פחות מאפיינים. עושים זאת כדי להוריד בעומס של החישובים על מאפיינים רב ממדיים. יש מספר שיטות שמבצעות את המרות הללו הידועה בניהם היא מסוג הלינארי ונקרת PCA ניתוח גורמים ראשיים.

לימוד חוקיות אסוציאטיבי (association rule learning) - זו שיטה לגילוי יחסים בין משתמים בדאטה סטים גדולים. תפקידה לזהות חוקיות חזקה בין המידע תוך שימוש במדידות ענייניות. שיטות אלה משמשות לעתים קרובות לניתוח סל שוק, מה שמאפשר לחברות להבין טוב יותר את הקשרים בין מוצרים שונים. השיטה הזאת גם משומשת שירותי סטרימינג כדי לזהות את הקשרים בין המשתמשים וסרטים שהם בוחרים לראות על מנת להבין מה מעניין אוכלוסייה ספציפית. וקיימים עוד שימושים רבים בשיטה הזו.

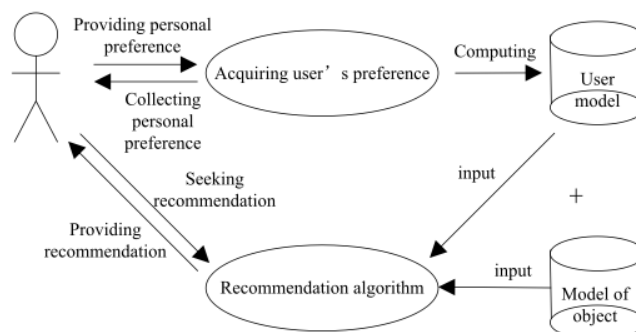
1.1 מה זה מערכות המלצה

מערכות המלצה זה מערכת של למידה חישובית שעוזרת למשתמשים לגלות מוצרים חדשים ושירותים שונים. כל פעם שאנחנו עושים קניות אונליין, מערכות המלצה מנחה אותנו לקראת מוצר הסביר ביותר שאנו עשויים לרכוש. בעידן החדש מערכות המלצה זה דבר כל כך שגרתי שאנו משתמשים בהם בלי שאנחנו מודעים בזה. בגלל שאנו לא יכולים לעבור על כל המוצרים או תוכן שמוצג באתרי האינטרנט, מערכות המלצה משחקים תפקיד חשוב בלעזור לשפר את חווית המשתמש.

כמה דוגמאות של מערכות האלה בפעולה כוללים המלצות למוצרים ב Amazon, הצעות של Netflix לסרטים וסדרות, סרטונים מומלצים ב YouTube, מוסיקה ב Spotify, חדשות ב Facebook וכך הלאה. מערכות המלצה לא דורשות ממשתמשים לספק צרכים ברורים. הרעיון המרכזי שלו הוא לנתח את ההתנהגות ההיסטורית של המשתמש ומידע העדפותיו, להקים מודל ולהמליץ אוטומטית

על הפריטים או המוצרים המעניינים את המשתמש, ולאחר מכן לקבל רשימה מותאמת אישית לשימוש. מערכת המלצה יכולה לא רק להמליץ למשתמש על פריטים בעלי העדפות דומות לפי העדפת המשתמש, אלא גם יכולה להמליץ על פריטים לא ידועים המעניינים אותם ללא העדפות המשתמש. מערכות האלה יכולות להקל בעיות אלו על ידי מציאת דרישות פוטנציאליות של משתמשים ובחירת פריטים רצויים מתוך כמות עצומה של מידע מועמד.

מערכת המלצה מורכבת משלושת חלקים הבאים: משתמש, משאב המוצר או פריט ואלגוריתם המלצה שמוצג באיור 1[1]. מודל של המשתמש הוקמה על ידי ניתוח תחומי עניין והעדפות של המשתמש, כמו כן מודל למשאבי הפריט (item resource) הוקמה בהתחשבות של מאפייני הפריטים. לאחר מכן מאפייני המשתמש מושווים למאפייני הפריטים לחיזוי המוצרים שיכולים לעניין את המשתמש בעזרת אלגוריתם ההמלצה ותוצאות החיזוי ייצגו למשתמש. ביניהם אלגוריתם המלצה הוא החלק החשוב ביותר במערכת. ביצועים של האלגוריתם המוצע משפיעים ישירות על ביצועים הכלליים של מערכת המלצה. בכללי מערכות המלצה מחולקות למספר הקטגוריות: מבוסס תוכן, סינון שיתופי והיברידי שמשלב את שניהם. בהמשך נעמיק בקטגוריות האלה. [1]



איור 1: מודל של מערכות המלצה

1.2 מדוע מערכות המלצה

במהלך השנים חברתי מלא ניסיון בשימוש של אפליקציות רבות ואתרי האינטרנט. לפני תחילת התואר במעדי המחשב, תהיתי מה גורם לי לקבל פרומות מתאימות ברשתות חברתיות או בגלישה באינטרנט. כשהתחלתי את התואר התחלתי להבין שמאחורי המלצות והפרסומות הללו עומדים מנגנונים לאיסוף ולשמירה של נתוני המשתמש וגם אלגוריתמים רבים לעיבוד הנתונים הללו. חידדה אצלי את המודעות לכך, חוויית השימוש באפליקציות כמו Spotify ו-Apple Music שהמלצות שלהם מבוצעות בצורה מקצועית ומאופקת. העדפתי את Spotify על פני Apple music מכיוון שמערכת המלצות שלה הייתה מצליחה יותר בהפצת רשימת השירים המתאימה לטעמי וגם הייתה מציעה שירים חדשים שאפילו לא נמצאו בתחום העניינים שלי, אך עם כל זאת התעניינתי לדעת אילו אלגוריתמים משתמשים כאלו גם בסטרימינג של יוטיוב, נטפליקס ושירותים נוספים. במהלך השימוש של האפליקציות נוצר בי רצון לחקור יותר לעומק את האלגוריתמים מסייעים בפיתוח רשימות ההמלצה למשתמשים. תהיתי איזה מידע משמש בפיתוח מודלי המצלה ואיך ממוש את האיסוף שלהם, איזה כלים עוזרים לעשות זאת. לגבי זה קיבלתי תובנות מאנשים בעבודה שעובדים על מערכות המלצה באחד חברות הסטרימינג המוכרות כמו prime video. נחשפתי לפרטים על הצוותים שונים שעוסקים בבניית המודלי המלצה וחיזוי, איסוף וניתוח המידע. כל זה עורר בי רצון להרחיב את ידע שלי בתחום של מערכות המלצה ולעשות ניסיונות עבור האלגוריתמים בבניית מערכת המלצה שמותאמת לי צרכים שלי. מומחיות במערכות המלצה יכול לסייע לא רק בבניית מערכות ושימוש באלגוריתמים, אלא גם בבניית אסטרטגיה ליצירת תוכן וסרטונים והפצתם ברשתות חברתיות. זה עוזר לחשיפה רבה יותר למשתמשים באפליקציות האלה. לדעתי הידע הזה שימושי בתחומים רבים של החיים ולא רק בשירותי סטרימינג. שימוש במערכות המלצה גדל משמעותי בתעשיות שונות כגון המזון ומסעדות, מוזיקה וסרטים, קניות באינטרנט ועוד. במסחר אלקטרוני, מערכות המלצה עוזרות למשתמשים למצוא מוצרים התואמים את העדפותיהם, מה שמוביל להגברת שביעות רצון הלקוחות ומכירות. בתחום הבריאות, מערכות האלה יכולות לספק המלצות טיפול מותאמות אישית מבוססות על היסטוריה הרפואית של מטופלים ומצבים, מה שמוביל לתוצאות בריאות טובות יותר. בסך הכל, החשיבות הגוברת של מערכות ממליצים משקפת את הצורך בהמלצות מותאמות אישית וממוקדות בעידן הדיגיטלי של היום, והשפעתן על תעשיות שונות צפויה להמשיך ולגדול בעתיד. טבלה הבאה מראה את השימוש במערכות המלצה בתחומים שונים.

טבלה 1: אפליקציות שמשמשות במערכת המלצה

Category	Recommender systems
Video	Netflix, YouTube, Amazon Prime ,Hulu
News	Google News, CNN, IFEG, DIGG

Music	Spotify, Pandora, Apple music, Google play, YouTube music
Social networking services	Facebook, Instagram, Twitter, LinkedIn, TikTok
E-business	Amazon, eBay, AliExpress, iHerb

1.3 אתגרים וקשיים בדרך

מערכות המלצה הפכו לכלי חיוני עבור תעשיות שונות להתאמה אישית של חוויות משתמש ולשיפור שביעות רצון המשתמש. עם זאת, הם גם מתמודדים עם מספר מגבלות ואתגרים שיכולים להפריע ליעילותם.

אחד האתגרים העיקריים הוא בעיית דלילות הנתונים, המתרחשת כאשר העדפות המשתמש והאינטראקציות עם פריטים אינן מיוצגות כראוי בנתוני המערכת. זה יכול להוביל להמלצות לא מדויקות, שכן ייתכן שלמערכת אין מספיק מידע כדי להציע הצעות מותאמות אישית.

מגבלה נוספת היא בעיית ה"התחלה הקרה", המתרחשת כאשר למשתמש או פריט חדש יש מעט או אין נתונים זמינים למערכת לניתוח. במקרים כאלה, המערכת עלולה להתקשה לקבל המלצות מדויקות עד לאיסוף נתונים נוספים.

בנוסף, מערכות ממליצים עלולות לסבול מבעיית "בועות סידור", כאשר המערכת המלצה רק על פריטים דומים לאלו שמשתמש קיים איתם אינטראקציה בעבר, מה שמוביל לחשיפה מוגבלת לתכנים חדשים ומגוון.

חששות לפרטיות מהווים אתגר גם עבור מערכות ההמלצה, שכן השימוש בנתונים אישיים למתן המלצות עלול להיתפס כפולשני על ידי חלק מהמשתמשים.

לבסוף, הבטחת הוגנות והימנעות מהטיה בהמלצות היא אתגר מתמשך. מערכות המלצה יכולות להנציח סטריאוטיפים ולהדיר קבוצות מסוימות אם לא מתוכננות ומיושמות בזהירות.

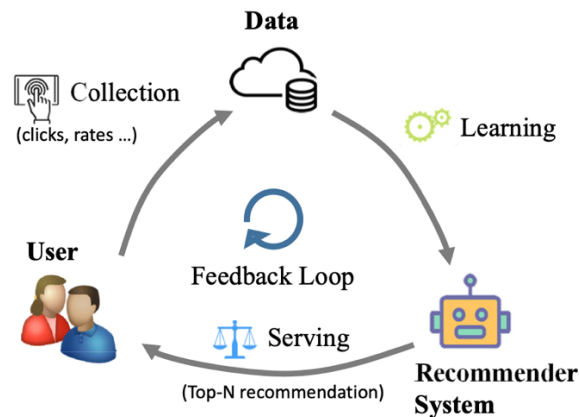
לסיכום, בעוד שמערכות ממליצים מציעות יתרונות משמעותיים הן למשתמשים והן לעסקים, הן מתמודדות גם עם מגבלות ואתגרים שונים הדורשים התייחסות ותשומת לב מדוקדקת.

2. איך מועברת מידע למערכות המלצה

2.1 תיאור כללי

אחד הדברים החשובים שבלעדיו לא נוכל לבנות מערכת המלצה טובה היא מידע עצמה שאותה מערכת תשתמש לחיזויים מדויקים עבור משתמשי המערכת. בענף של למידה חישובית איכות המידע שעליה היא לומדת ומבצעת חישובים זה דבר עיקרי ביותר ומבטיח הצלחה רבה למודלים חישוביים. מערכת המלצה שמבוססת על למידה חישובית (למידת מכונה) מונעת על ידי נתוני משוב המשתמש, למשל נתוני משוב מפורש של דירוגים ונתוני משוב מרומז של פעולות. בתולדות המחקר של מערכת המלצות חל מעבר מגמה משימוש נתוני משוב מפורש לנתוני משוב מרומז. איסוף של הנתונים לשימוש אותם במערכות המלצה הוא נעשה באפליקציות או אתרים עצמם שאנו משתמשים בהם, בקבלת ההודעות דירוג מוצר אחרי רכישה, אפליקציה שהייתה בשימוש, סרט או סרטון שהיה נצפה וכל תגובה ולחיצה של משתמש נשמר במאגר הנתונים. בהשארת תגובות או דירוג המוצרים לא עולה בפנינו מחשבה שזה יכול להישמר איפשהו. אך המידע הזאת נחוצה מאוד וחשובה מאוד שעוזרת לחברות שמספקות לנו שירות ומוצרים להבין איך לשפר אותו, להמליץ על מוצרים דומים. יש שיטות רבות לאוספת תגובות אללו שאנו נכיר בהמשך.

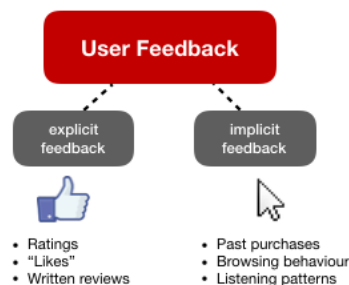
באיור 2 אפשר לראות את לולאת התגובה של המשתמש ו life cyclen של נתונים.



איור 2: מחזור חיי הנתונים [2]

2.2 שיטות העברה

- **משוב מפורש (Explicit ratings)** – מסופקים על ידי המשתמש. הם מספקים את העדפת המשתמש. דוגמאות כוללות דירוגי כוכבים, ביקורות, משוב, לייקים ומעקבים. מכיוון שמשתמשים לא תמיד מדריגים מוצר, דירוגים מפורשים עלולים להיות קשים להשגה. לאחר איסוף המשוב, המערכת יודעת עד כמה הפריט רלוונטי או דומה להעדפת המשתמשים.
לדוגמה, משתמשי אמזון יכולים לדרג פרטים מ-1 עד 10. דירוגים אלה נשלחים ישירות על ידי משתמשים, והסקאלה מאפשרת לאמזון לכמת את העדפות המשתמש. דוגמה נוספת למשוב מפורש היא כפתור האגודל למעלה של YouTube, אשר לוכד את ההעדפות המפורשות של המשתמשים (כלומר, לייק או לא אהבתי) עבור סרטון ספציפי. הבעיה במשוב מפורש הוא נדיר. מתי בפעם האחרונה לחצו על כפתור הלייק בסרטון YouTube או דירגו רכישות מקוונות? רוב הסיכויים שמספר הסרטונים שנצפו ביוטיוב עולה על מספר הסרטונים שדורגו.
- **משוב מרומז (Implicit ratings)** – מסופקים כאשר משתמשים מקיימים אינטראקציה עם הפריט. הם מסיקים התנהגות של משתמשים וקלים להשגה מכיוון שמשתמשים לוחצים באופן לא מדוע לאומת דירוג מפורש איפה שמתשמישים מתבקשים לדרג את המוצר. דוגמאות כוללות קליקים, צפיות ורכישות. (הערה: צפיות ורכישות יכולות להיות ישות טובה יותר להמליץ עליה מכיוון שהמשתמשים יבזבזו זמן וכסף במה שהכי חיוני עבורם).
לדוגמה, גם אם הסרטונים אינם מדורגים במפורש, הסרטונים שנצפו ב YouTube משתמשים כמשוב מרומז להתאמה אישית של המלצות. היתרון של השיטה הזאת הוא שהוא בשפע ואפשר להסיג אותו בלי בקשה מיוחדת מהמשתמש.
מערכות המלצה מבוססות על דירוגים מרומסים מאפשר למערכת לכוון את ההמלצות שלה בזמן אמת עם כל אינטראקציה של המשתמש.
עם זאת, למשוב מרומז יש גם חסרונות. בניגוד למשוב מפורש, ההנחה היא שכל אינטראקציה חיובית ואינה מסוגלת ללכוד העדפות השליליות מהמשתמשים.



- **נתונים דמוגרפיים** – סוג זה של נתונים כולל מידע על המשתמשים כגון גיל, מין, מיקום ועיסוק. ניתן להשתמש בנתונים אלה כדי להכשיר אלגוריתמי סינון מבוסס תוכן שנגיר בהמשך.
- **נתוני תכונות הפריט** – סוג זה של נתונים כולל מידע על פרטים, כגון כותרת, תיאור, ז'אנר ותאריך שחרור. ניתן להשתמש בנתונים אלה כדי להכשיר אלגוריתמי סינון מבוסס תוכן שנגיר בהמשך.

- **נתוני גרף** - סוג זה של נתונים כולל מידע על יחסים בין משתמשים ופרטים, כגון רשתות חברתיות והיסטוריית הרכישות. ניתן להשתמש בנתונים האלה כדי לאמן אלגוריתם מבוססי גרפים. נתוני גרף עבור מערכות המלצה יכולים להיות מיוצגים בצורות שונות, בהתאם ליחסים המתוכננים בניהם, למשל:

1. **יחס משתמש – משתמש** – סוג זה של גרף מייצג את היחסים בין המשתמשים, כגון קשרים חברתיים או תחומי עניין משותפים. צמתים בגרף מייצגים משתמשים וקשתות מייצגות את היחסים בניהם. ניתן לשקלל את הקשת כדי לייצג את חוזק הקשר.
2. **יחס פריט-פריט** - סוג זה של גרף מייצג את היחסים בין הפרטים, כגון דמיון או התרחשות משותפת.
3. **יחס משתמש-פריט** - סוג זה של גרף מייצג את היחסים בין משתמש לפריט, כגון היסטוריית רכישות או דירוגים.
4. **יחס משתמש-משתמש-פריט** - סוג זה של גרף מייצג את היחסים בין משתמשים, פריטים ומשתמשים אחרים.

- **נתוני עם חותמת זמן** - סוג זה של נתונים כולל מידע על מתי משתמשים ביצעו אינטראקציה עם הפרטים, כגון תאריך רכשיה, תאריך דירוג ותאריך סריקה. ניתן להשתמש בנתונים אלה כדי לאמן אלגוריתמים מבוססי זמני.

3. אלגוריתמים למערכות המלצה

3.1 גישה אלגוריתמית

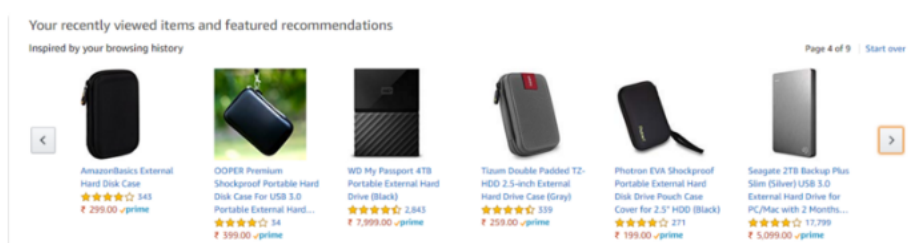
אחרי איסוף נתונים ממשתמש מערכת המלצה תשתמש באחד מהאלגוריתמים לאימון המודל שלה ובכך לחזות את העדפות של המשתמש ותציג לו את התוכן הרלוונטי.

ישנם מספר אלגוריתמים הנפוצים ביותר בשימוש של מערכות המלצה, כולל:

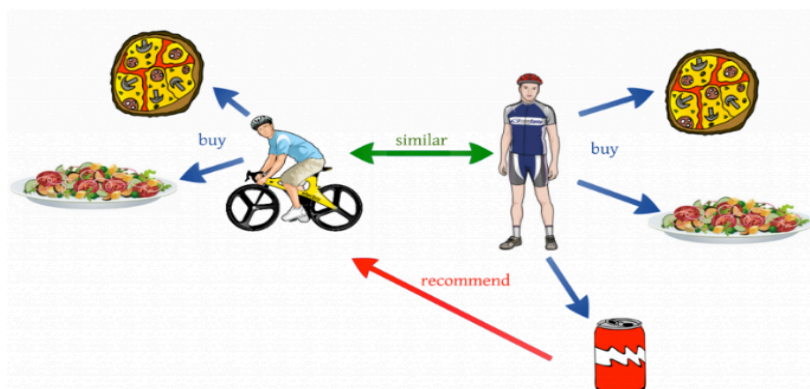
1. **סינון שיתופי (collaborative filtering)** - גישה זו משתמשת בהתנהגות העבר של המשתמש (כלומר מה שהם קנו או האבו) כדי להמליץ למשתמשים אחרים עם העדפות דומות. אלגוריתם סינון שיתופי יכול להיות מבוססי משתמש או מבוססי פרטים (נכיר אותם בהמשך).
2. **סינון מבוסס תוכן (content-based filtering)** - גישה זו משתמשת בתכונות של הפרטים המומלצים כדי להציע למשתמשים. ניתן להשתמש באלגוריתם זה כדי להמליץ על פרטים על סמך התכונות שלהם.
3. **מערכות המלצה היברידיות** – גישה זו משלבת גם שיטות סינון שיתופיות ומבוססות תוכן כדי להציע המלצות. הוא משלב את שתי החוזקות של שתי השיטות כדי לשפר את הדיוק הכולל של ההמלצות.

נכיר את הגישה של סינון מבוסס תוכן שמבוסס על דמיון המוצר וגם את הגישה של סינון שיתופי שמבוסס על דמיון המשתמש.

דמיון המוצר (Product similarity) – הוא מערכת השימושית ביותר להצעת המוצרים על סמך כמה משתמש היה רוצה את המוצר. אם המשתמש גולש ומחפש את המוצר מסוים, ניתן להציג לו מוצרים דומים. משתמשים מצפים לרוב למצוא מוצרים שהם רוצים במהירות ולהמשיך הלאה אם הם מתקשים במציאת הפריט הרלוונטי. כאשר משתמש לוחץ על מוצר אחד ניתן להציג מוצר דומה אחר, או אם יוזר קונה את המוצר נוכל לשלוח במייל פרסומות או קופונים המבוססים על פרטים דומים. דימון מוצר שימושי במיוחד כשאין הרבה נתונים עבור המשתמש, אבל כן ידוע באילו מהמוצרים הוא צופה.



דמיון המשתמש (User Similarity) – נועד לבדיקת ההבדל בין הדמיון של שני המשתמשים או מספר פרופילים תחת אותו משתמש כמו ב-Netflix למשל. אם לשני משתמשים יש העדפות דומות למוצר, אפשר להניח שיש להם תחומי עניין משותפים/דומים. אפשר להציג את זה כמו "חבר שממליץ על מוצר". עם זאת חסרון אחד של דמיון המשתמש הוא שדורש את כל נתוני המשתמש כדי להציע מוצרים שונים. זה נקרא בעיית התחלה קרה (cold start problem) מכיוון שבתחילת התהליך המצלה דורשת נתונים קודמים ממשתמשים. אתר מסחרי אלקטרוני שהושק לאחרונה, למשל, סובל מבעיית ההתחלה קרה כיוון שאין בו מספר גדול של היוזרים. לדמיון מוצר אין בעיה זו כי הוא רק דורש מידע על המוצר והעדפת המשתמש. Netflix ו-Spotify, למשל, נמנעות מבעיה הזו בכך שהן מבקשות מהמשתמשים את הלייקים ומציגות שאלון עבור כדי לדעת מה סגנון הסרטים ומוסיקה אהובים או מועדפים עבור המשתמש.



הזכרנו את בעיית ההתחלה הקרה (cold start problem) זה אחד האתגרים של מערכות המלצה.

הגדרת הבעיה : אחת הבעיות המרכזיות של מערכות המלצה בכללי היא בעיית התחלה קרה. כשהמשתמש נרשם למערכת או מוצר חדש התווסף לרשימת המוצרים ולמערכת אין מספיק נתונים הקשורים לפרטים/משתמשים האלו. במקרה כזה, מערכת לא יכולה להמליץ כראוי על פרטים קיימים למשתמש חדש (בעיית משתמש חדש) או להמליץ על פריט חדש למשתמשים הקיימים (בעיית פריט חדש). [3] בעיית משנה נוספת של התחלה קרה היא בעיית הדלילות שמתייחסת לעובדה שמספר הדירוגים הנתונים נמוך בהרבה ממספר הדירוגים האפשריים, מה שסביר במיוחד כאשר מספר המשתמשים ופרטים גדול. ההפך של היחס בין דירוג נתון לדירוג אפשרי נקראה **דלילות**. דלילות גבוהה מתורגמת לכיסוי דירוג נמוך מכיוון שרוב המשתמשים נוטים לדרג רק חלק קטן מהפרטים. ההשפעה היא שלעתים קרובות ההמלצות הופכות ללא אמינות. ערכים טיפוסיים של דלילות הם די קרובים ל-100% ברוב מערכות ההמלצה בעולם האמיתי. למשל בדאטה סט סרטים של Netflix יש דלילות של "רק" 98.82%.

פתרונות אפשריים לבעיה

כבר הוצעו מספר גישות להתמודדות עם בעיית התחלה קרה ובראשן גישות מבוססות תוכן (content based), הכלאה, המלצה בין תחומים ולמידה פעילה.

אחד הבעיות נוספות היא **הערכת מערכת ההמלצה**.

הגדרת הבעיה : עם שורשיו בלמידת מכונה (השוואה חיזוי דירוג) ואחזור מידע (השוואה פרטים "אחזור" המבוססים על "שאלות" מרומזות שניתנו על ידי העדפת המשתמש), תחום מערכות המלצה אימץ במקור מדדי הערכה מהתחומים האלו. למעשה דיוק ומדדים כמותיים קשורים, כגון מדדי דיוק, זכירה או שגיאה (בין דירוגים חזויים אמיתיים), הם עדיין הקריטריונים הנפוצים ביותר לשפוט את איכות המלצה של המערכת.

3.2 סינון שיתופי (collaborate filtering)

גישת סינון השיתופי (CF) הרוויחה יותר עניין לאחרונה והתקדמות רבה. העובדה שהיא שיחקה תפקיד המרכזי בהתקדמות של ה Netflix בין המתחרים שלה והיא תרמה מאוד לפופולריות שלה. בפרק הקודם של סריקת האלגוריתמים הקיימים בלמידת מכונה שעוזרים לאמן את המודל של מערכת המלצה כבר הגדרנו מה זה סינון שיתופי. מערכת המלצה המבוססת CF ניתן לחלק לשתי גישות משנה: מבוססות זיכרון ומבוססות מודלים. [4]

הגישה שמבוססת על זיכרון משתמשת במאגר מידע כולו על מנת למצוא קבוצה של משתמשים/פרטים הדומים למשתמש/פרט היעד. בעוד שהגישה השנייה מנסה לבנות מודל (למידת מכונה) המתארת את התנהגות המשתמש על מנת לחזות את בחירותיו.

בפעול, הוכח שהגישה מבוססת על זיכרון מראה ביצועים יותר טובים במונחים של דיוק כך שהגישה מבוססת מודל יעיל יותר בטיפול של דאטה סט גדולים. [4] יתר על כן, גישה מבוססת מודלים מורכבת יותר מאחר שהן כרוכות באימון מודל וכוונון מספר היפרפרמטרים.

מערכות המלצה מבוססות CF מתנהלים לעתים קרובות על ידי השוואת המצב החדש עם מצבים הקודמים. ואכן, מערכת מנתחת את ההתנהגות של קבוצת המשתמשים הדומים למשתמש היעד או מאפיינים של קבוצת מוצרים הדומים למוצר היעד. לדוגמה, אם משתמש רוכש מוצר המערכת תמליץ על אותו פריט

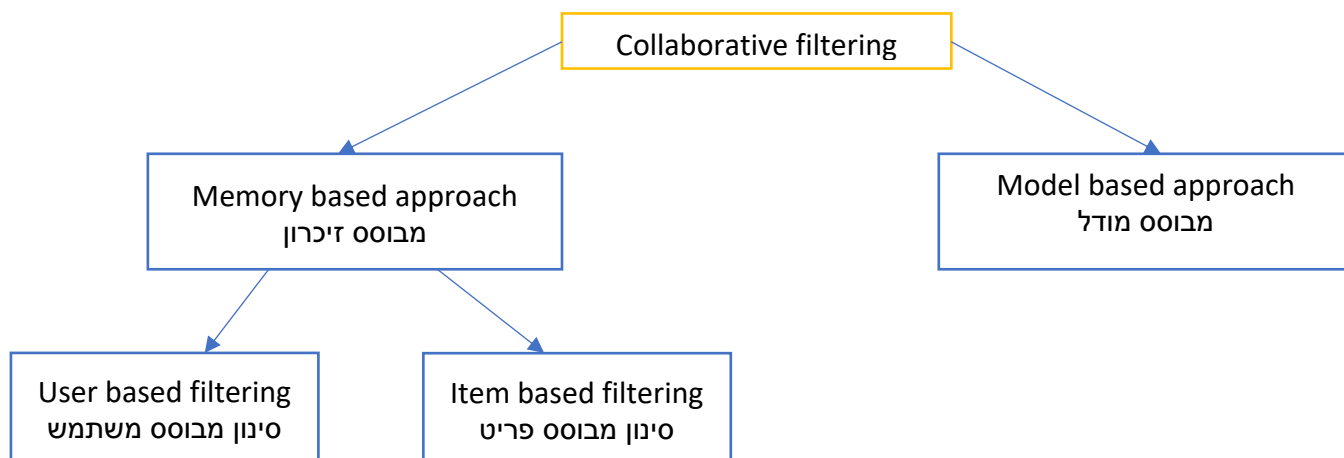
לסט של המשתמשים בעלי טעם דומה למשתמש הזה. בעיה זו מוגדרת כי תהליך אשכוליות (clustering) מכיוון שהוא נועד לסווג קבוצה של המשתמשים/פרטים לקבוצה הומוגניות. בעיקר מסתמכים על אשכול זה חישוב מרחק הסמנטי או הדמיון המחבר את האלמנטים בתוך קבוצה אחת עם השנייה. ככל שמספר אלמנטים משתפים אותו סט תכונות, ככל שערך הדמיון עולה. מסיבה זו, בחירת ממד דמיון המתאים בין קבוצה גדולה מאוד של מדדים הנתונים נחשבת כי אחד המשימות מרכזיות בעת יישום מערכת המלצה.[4]

Persons \ Movies	Joe	Bob	Carol
Titanic	5	1	5
The reader	1	5	2
Harry potter	4	2	?

טבלה 2: תהליך המלצה [5]

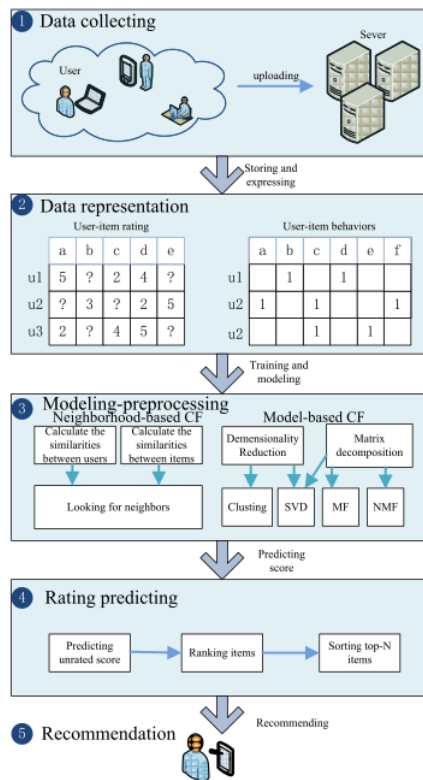
טבלה 2 מראה את התהליך ההמלצה בקצרה, כאשר ראשית עלינו להעריך את דעתה הפוטנציאלית החיובית של קרול על הארי פוטר, אפשר להשתמש בדמיון שלה עם ג'ו. לחלופין, אפשר לציין שהדירוגים של טיטאניק והארי פוטר עוקבים אחר דפוס זהה, מה שמראה שאנשים אהבו את הראשון עשויים לאהוב גם את השני. טבלה הזו נותנת מושג קצר לגבי סינון שיתופי. [5]

סינון שיתופי מסווג תחת גישות מבוססות זיכרון ומבוססות מודלים:



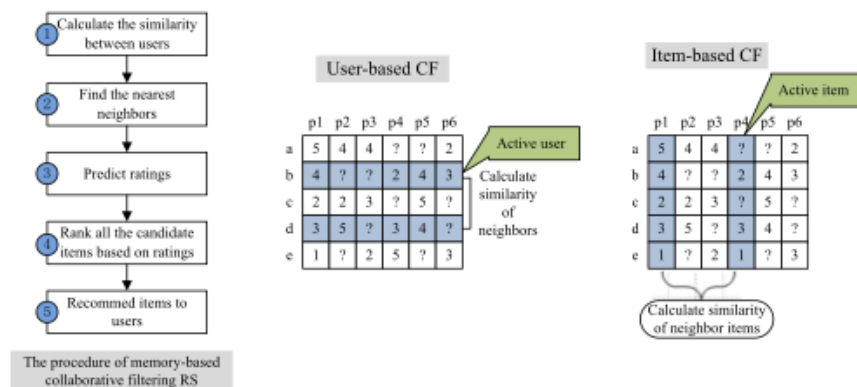
טכנולוגיית סינון שיתופי מבוסס זיכרון

סינון שיתופי מבוסס זיכרון הוא סוג של אלגוריתם המלצות המשתמש בהתנהגות משתמשים בעבר כדי להמליץ על פריטים למשתמשים. זה עובד על ידי ניתוח האינטראקציות ההיסטוריות של משתמש-פריט ודמיון בין משתמשים או פריטים כדי להציע המלצות.



איור 3: מסגרת של מערכת המלצה מבוססת סינון שיתופי [6]

קיימות שתי גישות עיקריות לסינון שיתופי מבוסס זיכרון: CF מבוסס משתמש ו CF מבוסס פריט. הרציונל של CF מבוסס שתמש ומבוסס פריט מוצג באיור 4.



איור 4: הרציונל של CF מבוסס שתמש ומבוסס פריט [6]

מבוסס משתמש - גישה זו מחפשת משתמשים עם העדפות דומות וממליצה על פריטים שהם אהבו בעבר. זה עובד על ידי חישוב הדמיון בין משתמש היעד לכל שאר המשתמשים במערכת. לאחר מכן, ההמלצות נוצרות על סמך הפריטים שמשתמשים דומים אהבו.

מבוסס פריט - גישה זו מתמקדת במציאת פריטים הדומים לאלו שהמשתמש אהב בעבר. זה עובד על ידי חישוב הדמיון בין פריט היעד לכל שאר הפריטים במערכת. לאחר מכן, ההמלצות נוצרות על סמך הפריטים הדומים ביותר לאלו שהמשתמש אהב.

לסינון שיתופי מבוסס זיכרון יש כמה יתרונות על פני אלגוריתמי המלצה אחרים, כמו הפשטות שלו והעובדה שהוא אינו דורש משאבי חישוב רבים. עם זאת, זה יכול לסבול מבעיות מדרגיות ככל שמספר המשתמשים והפריטים במערכת גדל. בנוסף, הוא יכול גם לסבול מבעיית ההתחלה הקרה, שבה קשה להמליץ על פריטים למשתמשים חדשים או על פריטים עם מעט או ללא נתונים היסטוריים.

אלגוריתמי המלצה CF מבוסס משתמש

בהתאם לדירוגים המחושבים. בכך, ניתן ליצור מערכת ההמלצה מבוססת משתמש שמתבססת על ההתנהלות ההיסטורית של המשתמש ופעולות הדירוג של משתמשים אחרים עם עניין דומה. מודל כזה עוזר להציע מוצרים ופריטים רלוונטיים בהתאם לדרישות המשתמש ולחזות את ההעדפות הבאות שלו.

1. איסוף נתונים

תחילה האלגוריתם אוסף נתונים על אינטראקציה עם הפריט-משתמש, כגון דירוגים, ביקורות ורכישות. בפרקים הקודמים הוסברו שיטות לעברת הנתונים למערכת ההמלצה. לכן האלגוריתם משתמש באחד מהם.

2. חישוב הדמיון בין המשתמשים

לאחר איסוף נתונים, נרצה לחשב את הדמיון בין המשתמשים על סמך הדירוגים שלהם. דירוגים של המשתמשים בדרך כלל מיוצגים כי ווקטור דירוג $r_u = \{r_{u1}, r_{u2}, \dots, r_{un}\}$. הדמיון בין שני משתמשים מחושב בעזרת שני וקטורי דירוג של אותם משתמשים ונוסחה למציאת הדמיון בין שני וקטורים. מדד הקלאסי לחישוב דמיון בין היוזרים הם דמיון הקוסינוס (cosine similarity) ומתאם פירסון או בשם המלא מקדם המתאים של פירסון (Pearson correlation coefficient).

דמיון הקוסינוס: דירוג המשתמש יכול להיות מוגדר כי וקטור ממד n, ודמיון בין היוזרים נעשה דרך מציאת זווית בין וקטורי דירוג. באופן כללי, ככל שהזווית קטנה יותר, הדמיון גבוה יותר. הנוסחה לחישוב היא:

$$sim_{uv} = \cos(r_u, r_v) = \frac{\langle r_u, r_v \rangle}{||r_u||_2 * ||r_v||_2} = \frac{\sum_{i \in I_{uv}} r_{ui} * r_{vi}}{\sqrt{\sum_{i \in I_u} r_{ui}^2} * \sqrt{\sum_{i \in I_v} r_{vi}^2}}$$

איפה ש sim_{uv} מייצג דמיון בין היוזרים u ו-v, r_u, r_v מייצגים וקטורי דמיון שלהם, בהתאמה, $||r_u||_2, ||r_v||_2$ מייצגים את נורמה-2 של u ו-v, בהתאמה, r_{ui}, r_{vi} מייצגים את הדירוג של u ו-v עבור פריט i. I_u, I_v מייצגים סט של פריטים שדורגו על ידי u ו-v, ו I_{uv} מייצג סט המוצרים שדורגו בדרך כלל על ידי שניהם u ו-v.

מקדם המתאים של פירסון (Pearson correlation coefficient) מחושב לפי נוסחה הבאה:

$$sim_{uv} = \frac{\sum_{i \in I_{uv}} (r_{ui} - \hat{r}_u)(r_{vi} - \hat{r}_v)}{\sqrt{\sum_{i \in I_u} (r_{ui} - \hat{r}_u)^2} * \sqrt{\sum_{i \in I_v} (r_{vi} - \hat{r}_v)^2}}$$

איפה ש \hat{r}_u ו \hat{r}_v מייצגים את התוחלת של הדירוגים מ u ו v , בהתאמה.

3. **מציאת שכנים הקרובים ביותר.** שיטה הזאת מאוד דומה לשיטה אחרת בתחום של למידת מכונה כי k -means, השיטה הזאת מחלקת את הדאטה ל k קבוצות כך שהמרחק בין האיברי קבוצה הוא קטן ביותר וגדול בין האיברים שנמצאים בקבוצות שונות. כלומר דאטה מחולק כך שבכל קבוצה נתונים מאוד קרובים אחד לשני או מתארים דפוס דומה. אז ישנן שתי שיטות למציאת k - שכנים הקרובים ביותר והגדרת הסף ($threshold$). אז שיטת מוצאת k שכנים הקרובים היא בוחרת את k משתמשים הראשונים שהדמיון בניהם ובין המשתמש הפעיל u הוא הקרוב ביותר. שיטת הסף פירושה סוף δ נקבע בתחילה, כאשר הדימיון בין המשתמש v למשתמש הפעיל u גדול מ δ , והמשתמש v נבחר כאחד השכנים הקרובים ביותר.

4. **חיזוי רייטינג.** ישנן שתי שיטות מרכזיות להמלצה עבור משתמש הפעיל: חיזוי הרייטינג וסיפוק של רשימת הטופ של N מוצרים. דירוג החזוי מחושב על יד נוסחה הבאה:

$$\hat{r}_{ui} = \bar{r}_u + \frac{(\sum_{v \in N_u} sim_{uv}(r_{vi} - \bar{r}_v))}{\sum_{v \in N_u} |sim_{uv}|}$$

איפה N_u מציין את קבוצת השכנים הדומים למשתמש u . המלצת שהם מדוגרות כי Top-N מוצרים משמשת בעיקר בתרחישים הבאים: אתרי קניות או אתרים שאין בהם דירוג מפורש של מוצרים. במצב כזה, המידע ממשב המרוזם של המשתמש ייתן אינדיקציה על הרשימה של המוצרים שעשויים לעניין את המשתמש ובכך יתווספו לרשימת ההמלצה עבור המשתמש הפעיל, וגם מידע השימושי יכולה להיות מחולצת מהמשוב לבניית המטריצה משתמש-פריט איפה של כל אלמנט הוא 0 או 1. באופן כללי, העדפות המשתמש מעוצבות בצורה נקודתית, דירוג של כל משתמש עבור כל פריט (ערך הסתברותי בין 0 ל 1) נחזה, ולאחר מכן הפריטים המדורגים ממוינים בסדר יורד, ולבסוף פריטים מובילים את הרשימה הם מומלצים למשתמש. CF מבוסס זיכרון עבור נתונים בינאריים יכול להיחשב למעשה כמקרה מיוחד של השיטה הזאת עבור דירוגים. הדירוג $r_{ui}=1$ אם נפצה צמד הפריטים של המשתמש (u,i) , $r_{ui}=0$ אחרת במטריצה המשוב R . לכן, דמיון וקטור הקוסינוס עבור דירוגים בינאריים מחושב באופן הבא:

$$sim_{uv} = \cos(\vec{r}_u, \vec{r}_v) = \frac{\sum_{i \in I_{uv}} r_{ui} \cdot r_{vi}}{\sqrt{\sum_{i \in I_u} r_{ui}^2} \sqrt{\sum_{i \in I_v} r_{vi}^2}} = \frac{|I_u \cap I_v|}{\sqrt{|I_u|} \sqrt{|I_v|}}$$

איפה ש I_u ו I_v מצינים את הסט הפריטים נצפים על ידי משתמשים u ו v , I_{uv} מציין את הפריטים המשותפים שנצפו על ידי שני המשתמשים.

אלגוריתם המלצות CF מבוסס פריט

אלגוריתמים מבוססי פריטים הם סוג של אלגוריתם סינון שיתופי המשמש למתן המלצות מותאמות אישית למשתמשים על סמך קווי הדמיון בין הפריטים. אלגוריתמים אלו פועלים על ידי ניתוח ההתנהגות ההיסטורית של המשתמשים ומציאת פריטים אחרים הדומים לפריטים שהמשתמש קיים איתם אינטראקציה בעבר. לאחר מכן נוצרות ההמלצות על סמך פריטים דומים אלה.

בדומה לאלגוריתם המלצה CF מבוסס משתמש, גם אלגוריתם המלצה CF מבוסס פריט מבצע את אותם שלבים עם שינויים מינוריים:

- (1) איסוף נתונים נעשה בדומה לאלגוריתם CF מבוסס משתמש.
- (2) דמיון הפריט: האלגוריתם מחשב את הדמיון בין פריט היעד לכל שאר הפריטים במערכת בעזרת אותם נוסחאות לחישוב הזווית בין שני ווקטורים.
- (3) בחירת השכנים הקרובים ביותר: לאחר מכן האלגוריתם בוחר קבוצת משנה של הפריטים הדומים ביותר כשכנים של פריט היעד. מספר השכנים שנבחר יכול להשתנות בהתאם לאלגוריתם ולמקרה השימוש הספציפי.
- (4) הפקת המלצות: לבסוף, האלגוריתם מייצר המלצות על ידי צבירת הדירוגים או ההעדפות של המשתמשים עבור השכנים הקרובים ביותר של פריט היעד. ניתן להשתמש בשיטות צבירה שונות, כגון ממוצע משוקלל והצבעה של השכן הקרוב.

לאלגוריתמים המבוססים על פריטים יש כמה יתרונות על פני אלגוריתמי המלצות אחרים, כמו יכולתם להתמודד עם מערכי נתונים גדולים ועמידותם לבעיית הדלילות (כלומר, הבעיה של מספר רב של פריטים עם מעט דירוגים). עם זאת, הם יכולים לסבול מבעיית הטיית הפופולריות (כלומר, הנטייה להמליץ על פריטים פופולריים) ומבעיית ההתחלה הקרה (כלומר, הבעיה של המלצה על פריטים שיש להם מעט נתונים היסטוריים או ללא נתונים היסטוריים).

טכנולוגיית סינון שיתופי מבוססת מודל

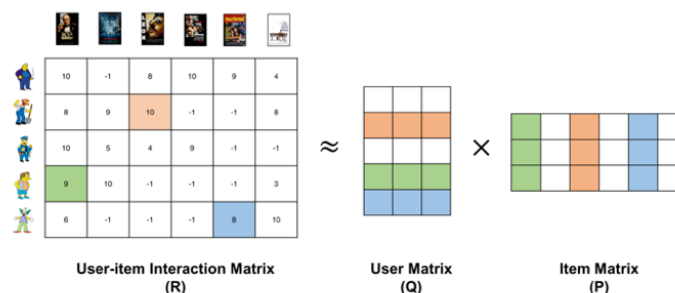
אלגוריתמים מבוססי מודלים של סינון שיתופי הם סוג של האלגוריתם המלצה המשתמשים במודלים סטטיסטיים או למידת מכונה כדי ליצור המלצות המבוססות על אינטראקציות היסטוריות בין פריט משתמש. אלגוריתמים אלה מנתחים את הדפוסים והיחסים בין משתמשים ופריטים כדי לחזות העדפות המשתמש ולייצר המלצות.

לדוגמה, [7] תאר גישה מוקדמת למיפוי CF לבעיית סיווג, ובניית מסווג עבור כל משתמש פעיל המייצג פריטים כווקטורי תכונה על פני משתמשים ודירוגים זמנים כתוויות, אולי בשילוב עם טכניקות הפתחת ממד (PCA) כדי להתגבר על דלילות הנתונים נושאים. טכניקות דוגמנות חזויות אחרות יושמו גם בדרכים קשורות.

1.1 מודל פירוק מטריצה (matrix factorization model)

הRS מבוסס מודל דורש שלב למידה מראש לבירור פרמטרי המודל האופטימליים לפני מתן המלצה. לאחר סיום שלב הלמידה, הRS מבוסס מודל יכול לחזות את דירוג המשתמשים במהירות רבה. ביניהם, מודל הגורמים הסמויים (latent factor model) ומודל פירוק המטריצה (matrix factorization model). לעומת השיטה שהכרנו קודם דמיון השכנים שמייצר המלצות על בסיס הסטטיסטי של הדמיון בין המשתמשים או בין המוצרים, מודל הגורמים הסמויים (LFM) מניחה שדמיון בין משתמשים ופריטים נגרמת בו זמנית על ידי מבנה ממדים תחתונים ונסתרים בנתונים. לדוגמה, ניתן להניח שהדירוג משתמש נותן לסרט תלוי בכמה גורמים מרומזים כמו טעמו האישי בז'אנרים שונים של הסרט. אלגוריתם פירוק מטריצות פועלים על ידי פירוק מטריצת האינטראקציה של משתמש-פריט למכפלה של מטריצות מלבניות בעלות ממד נמוך יותר. משפחת השיטות זו זכתה לפרסום נרחב במהלך אתגר פרס Netflix בשל יעילותו כפי שדווח על ידי סיימון פונק בפוסט שלו בבלוג משנת 2006, שם שיתף את ממצאיו עם קהילת המחקר. ניתן לשפר את תוצאות החיזוי על ידי הקצאת משקלי רגולציה שונים לגורמים סמויים על סמך הפופולריות של הפרטים ופעילות המשתמשים.

אלגוריתמי פירוק מטריצות משתמשים בטכניקות אלגברה ליניאריות כדי לפרק את מטריצת האינטראקציה של משתמש-פריט לשתי מטריצות ממדיות נמוכות יותר המייצגות תכונות משתמש ותכונות פריט. אלגוריתמים אלה לומדים את התכונות הסמויות של משתמשים ופריטים ומשתמשים בהם כדי לחזות את הדירוגים או ההעדפות החסרים. אלגוריתמי פירוק מטריצה יכולים להתמודד עם בעיות הדלילות והמדורג של אלגוריתמי סינון שיתופי, והם משמשים לעתים קרובות ביישומים בעולם האמיתי. גישה MF (פירוק מטריצה) נמצאת כי גישה מדויקת ביותר לצמצום בעיה מרמות הייחודיות הגבוהות במאגרי המידה, מחקרים מסוימים השתמשו בטכניקות הפחתת הממדיות. MF משמש במיוחד לאיבוד מאגרי מידע עצומים של מערכת המלצה ומספקת גישות ניתנות להרחבה. [8]



איור 5: דוגמה של פירוק מטריצה על המלצת הסרט (נלקח מפוסט ב-LinkedIn)

נתבונן בהליכי המלצה של מערכת המלצה מבוססת פירוק מטריצה.

בנה מודל תכונות הסמויים: $R \approx QP$

למטריצה משתמש-פריט $R \in \mathbb{R}^{(users \times items)}$ מודל פירוק מטריצה מצוין בערך את מטריצה R כי מכפלה של מטריצת פיצורים עבור משתמש $Q \in \mathbb{R}^{users \times latent \ factors}$ ומטריצה פיצורים עבור הפריטים $P \in \mathbb{R}^{latent \ factors \times items}$.

באופן ספציפי, דירוג החיזוי שמשתמש u תיתן לפריט i מחושב כך:

$$\tilde{r}_{ui} = \sum_{f=0}^{n \text{ factors}} Q_{u,f} P_{f,i}$$

מודל MF נועד בדרך כלל למזער את הסטייה בין הפירוק של המטריצה המשוערת למטריצת דירוג המקורית של פריט-משתמש. לכן, אנו מאמנים את המודל באמצעות שיטת הירידה בשיפוע (gradient descent) כדי להשיג את הפתרון האופטימאלי עבור Q ו-P. פונקציית המטרה נראית כך:

$$\min ||R - \hat{R}|| = \min \left(\sum r_{ui} - q_{uk} p_{ki} \right)^2 + \lambda_q ||q_{uk}||^2 + \lambda_p ||p_{ki}||^2$$

איפה ש r_{ui} מצין את הדירוג של משתמש u על הפריט i במטריצה מקורית, q_{uk} ו p_{ki} מציינים את הפיטצ'ר k ממשתמש u ומפריט i ב-Q ו-P. התווספו פרמטרים λ_q, λ_p כדי לעשות רגולציה למקרה של overfitting.

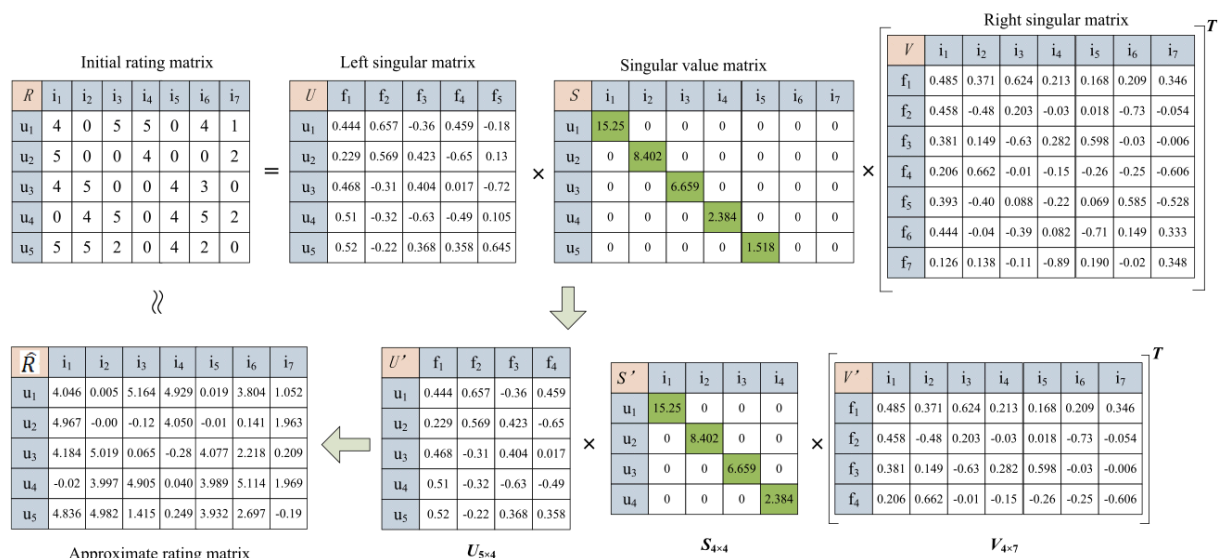
1.2 פירוק לערכים סינגולריים (SVD – singular value decomposition)

פירוק לערכים סינגולריים היא טכניקה מאוד עוצמתית להורדת ממד. זה מימוש מסוים של גישה פירוק מטריצה MF וגם קשורה ל-PCA. הנושא המרכזי בפירוק SVD הוא מרחב תכונות ממד נמוך יותר. למשל נתונים לנו קבוצת התכונות כך: $\{\vec{x}_n\}_{n=1}^N$ איפה שמוגדרת מטריצה R ש SVD עבור מטריצה R : $SVD(R) = USV^T$

איפה ש U ו- V מטריצות אורתוגונליות $U \in \mathbb{R}^{m \times m}, V \in \mathbb{R}^{n \times n}, S \in \mathbb{R}^{m \times n}$ מטריצה סינגולרית ואורתוגונלית עם r ערכים אי שליליים. איברי אלכסון כל S מקיימים $(\lambda_1, \lambda_2, \dots, \lambda_r)$ את: $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_r > 0$. r עמודות ראשונות של מטריצות U ו- V מייצגות את הווקטורים עצמיים שקשורות ל r ערכים עצמיים של RR^T and $R^T R$. [9]

ל-SVD יש תכונה חשובה שהופכת אותה למעניין במיוחד עבור היישומים שלנו. היא מספקת את הקירוב לינארי הטוב ביותר בדרגה קטנה יותר של המטריצה המקורית. ניתן לשמור רק $k \ll r$ ערכים סינגולריים עד ידי ביטול של האחרים. אנו מגדירים את S_k אחרי הורדת ממד. מאשר שארכי האלכסון ממוינים, אז תהליך של הורדת ממד נעשה על ידי שמירה של k ערכים סינגולריים ראשונים, כלומר הגדולים ביותר מבין כל r ערכים סינגולריים. מאשר שהורדנו ממד עבור S , אז גם נוריד ממד עבור U ו- V ונקבל את המטריצות U_k ו- V_k . מטריצה U_k נוצרה על ידי הורדת $(r-k)$ עמודות מ- U ומטריצה V_k נוצרה על ידי הורדת $(r-k)$ שורות מ- V .

מטריצה משוחזרת $\hat{R} = U_k S_k V_k^T$ מדרגה k היא הקירוב הקרוב ביותר למטריצה מקורית R . גישה הפחתת ממד ב-SVD יכול להיות שימושי מאוד עבור תהליך סינון שיתופי. SVD מייצר קבוצה של וקטורים עצמיים לא מתואמים. כל לקוח ומוצר מיוצגים על ידי הווקטור עצמי המתאים לו. תהליך הפחתת ממד עשוי לסייע למשתמשים שדירגו מוצרים דומים (אך לא בדיוק אותם מוצרים) להיות ממופים לתוך מרחב הנפרש על ידי אותם וקטורים עצמיים.



איור 6: תהליך פירוק מטריצה של SVD [6]

לאלגוריתמים מבוססי מודלים של סינון שיתופי יש כמה יתרונות על פני אלגוריתמי המלצות אחרים, כמו יכולתם להתמודד עם בעיות התחלה קרה והדיוק שלהם בניבוי העדפות המשתמש. עם זאת, הם יכולים לסבול מהתאמת יתר, כאשר המודל מתאים מדי לנתוני האימון ואינו מכליל היטב לנתונים חדשים. בנוסף, הם דורשים יותר משאבי חישוב ומומחיות לאימון ופריסה בהשוואה לאלגוריתמים מבוססי זיכרון.

3.3 סינון מבוסס תוכן (content-based filtering)

סינון מבוסס תוכן הוא טכניקה המשמשת במערכות המלצה כדי להמליץ על פריטים למשתמשים על סמך העדפותיהם ואינטראקציות קודמות. גישה זו מתמקדת במאפיינים של הפריטים עצמם, ולא בהתנהגות המשתמש או בדמיון בין המשתמשים.

בסינון מבוסס תוכן, פריטים מתוארים לפי קבוצה של תכונות או תכונות, כגון ז'אנר, במאי, שחקנים, דירוג וכו'. לאחר מכן המערכת מנתחת את התכונות הללו כדי ליצור פרופיל משתמש, המכיל את העדפותיו ותחומי העניין של המשתמש.

כדי להמליץ למשתמש על פריטים, המערכת משווה את פרופיל המשתמש עם התכונות של הפריטים במסד הנתונים ובוחרת את אלו המתאימים ביותר להעדפות המשתמש. לדוגמה, אם משתמש הראה העדפה לסרטי פעולה עם דירוג גבוה, המערכת תמליץ על סרטי פעולה עם דירוג גבוה.

יתרון אחד של סינון מבוסס תוכן הוא שהוא יכול להמליץ על פריטים למשתמשים בעלי טעמים או העדפות ייחודיות, מכיוון שהוא אינו מסתמך על דמיון בין משתמשים. עם זאת, היא עשויה להתקשות להמליץ על פריטים חדשים או לא מוכרים, מכיוון שיתכן שלא יהיה מספיק מידע זמין לגביהם. בנוסף, אם המערכת מסתמכת יותר מדי על סט מוגבל של תכונות או תכונות, היא עלולה לפספס היבטים חשובים של פריט שרלוונטיים למשתמש.

הקצאת תכונות בסינון מבוסס תוכן

הקצאת תכונות בסינון מבוסס תוכן כרוכה בבחירת התכונות המתארות את הפריטים במסד הנתונים והקצאת ערכים לאותם תכונות עבור כל פריט. תכונות אלו משמשות לאחר מכן לבניית מטריצת התכונות ויצירת המלצות למשתמשים על סמך העדפותיהם.

הקצאת תכונות כוללת בדרך כלל שילוב של שיטות ידניות ואוטומטיות. תכונות מסוימות, כגון הכותרת או המחבר של ספר, עשויות להיות פשוטות להקצאה ידנית. תכונות אחרות, כגון הז'אנר או הטון של ספר, עשויות לדרוש שיפוטיות סובייקטיביות יותר ועשויות להיות מוקצות באמצעות שיטות אוטומטיות, כגון עיבוד שפה טבעית או אלגוריתמים של למידת מכונה.

בעת הקצאת תכונות, חשוב לקחת בחשבון את הרלוונטיות והאינפורמטיביות של כל תכונה למטרות המלצה. תכונות המתואמות מאוד עם העדפות המשתמש עשויות להיות אינפורמטיביות יותר מתכונות שהן פחות מתואמות. בנוסף, תכונות שהן ספציפיות מאוד לפריט מסוים עשויות להיות פחות אינפורמטיביות מתכונות המשותפות לפריטים רבים.

לאחר שהוקצו התכונות ומטריצת התכונות נבנתה, ניתן להשתמש במדדי דמיון כגון דמיון קוסינוס או מרחק אוקלידי כדי לחשב את הדמיון בין פריטים ולייצר המלצות למשתמשים.

הקצאת תכונות היא תהליך מתמשך, שכן פריטים חדשים מתווספים למסד הנתונים והעדפות המשתמש מתפתחות עם הזמן. עדכונים ותחזוקה שוטפים של מטריצת התכונות יכולים לעזור להבטיח שההמלצות יישארו מדויקות ורלוונטיות לאורך זמן.

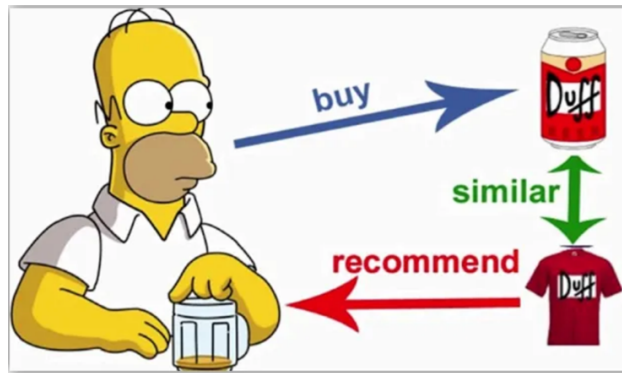
בניית פרופיל המשתמש

בניית פרופיל משתמש בסינון מבוסס תוכן כרוכה בניחוח אינטראקציות העבר של המשתמש עם פריטים במסד הנתונים כדי לקבוע את העדפותיו ותחומי העניין שלו. פרופיל המשתמש משמש לזיהוי פריטים שצפויים לעניין את המשתמש וליצירת המלצות מותאמות אישית.

כדי לבנות פרופיל משתמש, המערכת מזהה תחילה את הפריטים שהמשתמש מקיים איתם אינטראקציה, כמו סרטים שצפו בהם או מוצרים שרכשו. לאחר מכן, המערכת מחלצת את התכונות או התכונות של פריטים אלה ומצבירה אותם כדי ליצור סיכום של העדפות המשתמש.

פרופיל המשתמש מורכב בדרך כלל מווקטור תכונה, שהוא רשימה של התכונות שהמשתמש מקיים איתן אינטראקציה והערכים המשויכים אליהן. לדוגמה, אם משתמש צפה במספר סרטי פעולה עם דירוג גבוה, וקטור התכונה עשוי לכלול את התכונות "ז'אנר: פעולה" ו"דירוג: גבוה". ניתן לנרמל את וקטור התכונה כדי להבטיח שתכונות עם ערכים גדולים יותר לא ישלטו בחישוב הדמיון.

לאחר מכן המערכת יכולה להשתמש בפרופיל המשתמש כדי ליצור המלצות על ידי השוואת וקטור התכונות של המשתמש לווקטורי התכונות של פריטים במסד הנתונים. הפריטים בעלי ציוני הדמיון הגבוהים ביותר מומלצים למשתמש.



יתרון אחד של סינון מבוסס תוכן הוא שניתן לעדכן את פרופיל המשתמש בזמן אמת כאשר המשתמש מקיים אינטראקציה עם פריטים חדשים, מה שמאפשר המלצות מותאמות אישית ומותאמות. עם זאת, ייתכן שהמערכת תתקשה להמליץ על פריטים מחוץ להעדפותיו הידועות של המשתמש, מכיוון שהיא מסתמכת אך ורק על אינטראקציות העבר של המשתמש עם פריטים.

מטריצה תכונות

בסינון מבוסס תוכן, התכונות של פריטים מיוצגות לרוב בצורת מטריצה, המכונה מטריצת תכונה. מטריצה זו מכילה שורה אחת עבור כל פריט במסד הנתונים ועמודה אחת עבור כל תכונה או תכונה המתארת את הפריט.

כל רכיב במטריצת התכונה מייצג את הערך של תכונה עבור פריט מסוים. לדוגמה, מטריצת תכונה עבור מסד נתונים של סרט עשויה לכלול עמודות עבור שם הסרט, הז'אנר, הבמאי, השחקנים, הדירוג ושנת היציאה. כל שורה תייצג סרט בודד, והרכיבים בכל עמודה ייצגו את הערך של תכונה זו עבור הסרט המתאים.

	Feature 1	Feature 2	Feature 3	Feature 4
Product 1	1		1	2
Product 2		1	4	
Product 3	3			1
User Data	2		1	1

ניתן להשתמש במטריצת התכונות כדי לחשב את הדמיון בין פריטים, שהוא מרכיב מרכזי של סינון מבוסס תוכן. ניתן למדוד דמיון באמצעות מגוון שיטות, כגון דמיון קוסינוס או מרחק אוקלידי. שיטות אלו מחשבות את המרחק או הזווית בין וקטורי התכונה של שני פריטים ומייצרות ציון דמיון, שניתן להשתמש בו כדי לדרג פריטים להמלצה.

יתרון בשימוש במטריצת תכונה הוא שהיא מאפשרת חישוב יעיל של דמיון בין פריטים, אפילו במסדי נתונים גדולים. זה גם מאפשר למערכת לשלב בקלות פריטים חדשים במסד הנתונים, פשוט על ידי הוספת שורה

חדשה למטריצת התכונות. עם זאת, יצירת מטריצת תכונות מקיפה ומדויקת יכולה להיות מאתגרת, מכיוון שהיא דורשת בחירה קפדנית של תכונות רלוונטיות ותווית מדויקת של פריטים.

4. מערכות המלצה בשירותי סטרימינג

כדי להצליח כשירות סטרימינג, חברה צריכה להתמקד בספק הצרכים של הלקוח ולספק לו את התוכן הרלוונטי עבורו. למען כך, החברות המספקות שירותי סטרימינג משתמשות במערכות המלצה המובנות במוצר שלהן. כאשר המערכת לומדת את העדפות המשתמש ומשתמשת בצפיות, לייקים ומידע נוסף, היא יכולה ליצור רשימות אישיות של תוכן מתאים למשתמש ולהמליץ עליהם. וסף לספק תוכן מתאים, חשוב לחברה להציע מערכת המלצה חזקה, שתוכל להמליץ על תוכן רלוונטי באופן יעיל ומהיר. זהו מדד חשוב לבחירת הלקוח לספק סטרימינג כך שלא רק התוכן ומחיר המנוי החודשי ישפיעו על הבחירה שלו.

4.1 שירותי סטרימינג

קיימים מגוון רחב של שירותי סטרימינג המציעים יתרונות שונים. השירותים הללו שונים מאוד בתוכן המוצע, במחירי המנוי וביכולות ההמלצה שלהם. כל אחת מהחברות שמוצגת ברשימה הבאה מייצרת את הסרטים בכוחות עצמה ולכן תוכן תוצרת בית של אחד החברות לא יהיה מוצע ברשימות הסרטים של חברה מתחרה. לכן לצרכי המשתמש יש שיקולים רבים בבחירת שירות המתאים שירותי סטרימינג הפופולריים ביותר:

1. Netflix

2. Amazon Prime Video

3. HBO Max

4. Hulu

5. Disney+

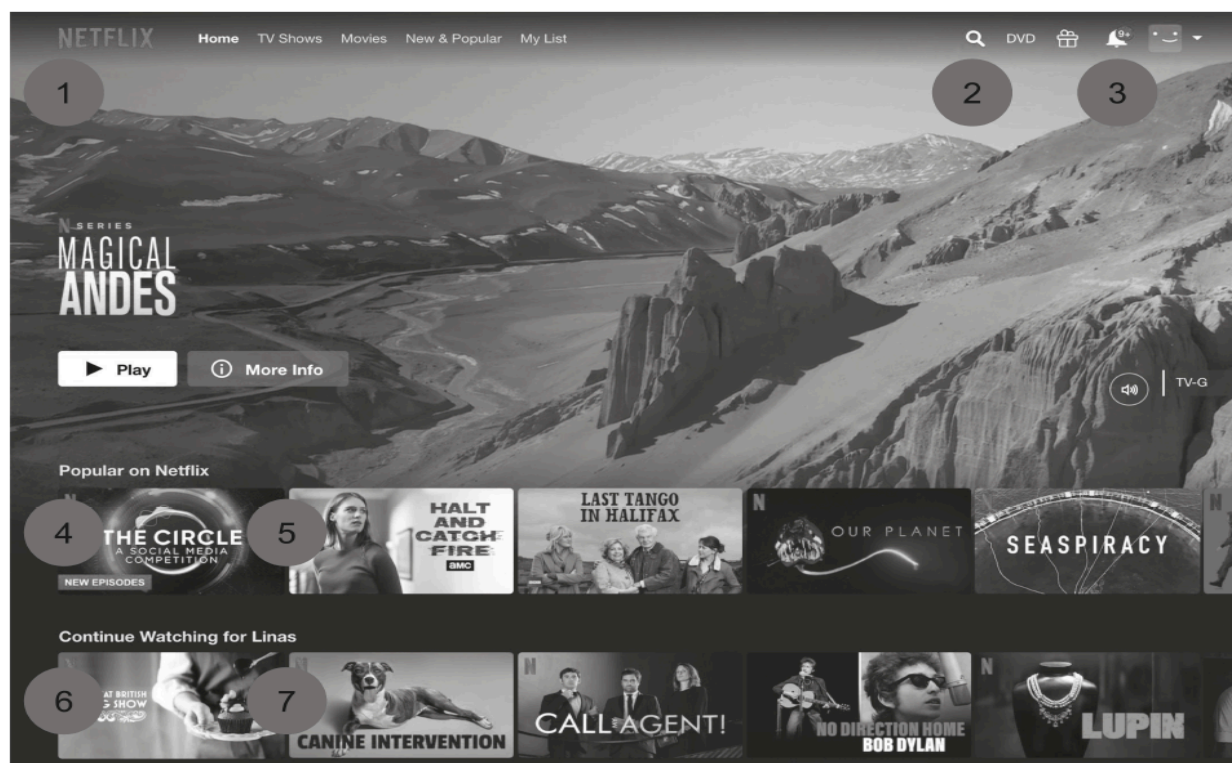
קיימות עוד חברות שלא מופיעות ברשימה הזו. כל אחת מהן משתמשת במערכות המלצה ובונה את המודלי חישוב של ההמלצה בצורה ייחודית לנתונים שלהן, אך כל מערכת זו מבוססת על השיטות המפורטות בסעיפים הקודמים. ישנן שיטות רבות ליישום מערכות המלצה, כגון שימוש ברשתות נוירונים או שילוב של כמה שיטות בו זמנית. הבחירה בסוג המערכת מתבצעת על ידי המחקר והפיתוח של החברות, תוך התחשבות רבה בצרכי המשתמשים.

נטפליקס היא אחת חברות שקפצו לעולם שירותי הסטרימינג המקוונים. נטפליקס, שנוסדה ב-1999 כחנות וידאו מקוונת, הפכה לספקית הסטרימינג המקוונת האמריקאית המשומשת ביותר, ועדיין צומחת חזקה, המתמחה בהפצת וידאו על פי דרישה. נכון לעכשיו, הם פעילים ביותר מ-190 מדינות בכל רחבי העולם עם למעלה מ-100 מיליון מנויים. לאחרונה, מספר המנויים של נטפליקס בארה"ב עלה על מספר המנויים

לטלוויזיה בכבלים רגילה בתשלום, ראה איור 2. מדי יום נצפים למעלה מ-125 מיליון שעות של וידאו בנטפליקס, ומספר הכותרים ממשיך לגדול. [10]

מחקרי צרכנים מראים שמשתמש נטפליקס טיפוסי מאבד עניין לאחר אולי 60 עד 90 שניות של בחירה, לאחר שבדק 10 עד 20 כותרים (אולי 3 בפירוט) על מסך אחד או שניים. המשתמש מוצא משהו מעניין או שהסיכון שהמשתמש ינטש את השירות גדל באופן משמעותי. בעיית ההמלצה היא לוודא שבשני המסכים האלה כל חבר במאגר המגוון ימצא משהו משכנע לצפייה, ויבין למה זה עשוי לעניין. [11]

נטפליקס לא משתמשת באלגוריתם שמניע את הכלל המלצות אלא קבוצה של טכניקות שכן כולם מיושרים למטרתם להגביר את שביעות הרצון של המשתמשים. בקריאה של מאמרים שונים של אלגוריתמי המלצה ואלה שנפוצות ביותר, מצאתי שאין שיטה אחת בעלת הביצועים הטובים ביותר (בין אם למידה עמוקה או אחר) תלוי בהמלצה הספציפית משימה שיש לפתור וכן על הנתונים הזמינים. לסיבה זו, סוגים שונים של מודלים של למידת מכונה משמשים ליצירת המלצות מותאמות אישית עבור חלקים השונים של עמוד הבית של נטפליקס.



איור 7: עמוד בית של נטפליקס [1]

באיור 7 אפשרת לראות דוגמה לדף בית של שירות סטרימינג נטפליקס. עיגולים ממוספרים מונה משימות המלצה שונות, כל אחת מהן מופעלת על ידי אלגוריתם אחר. תחת עיגול מספר 1 יש אלגוריתם ייחודי לבחירת הסרט שיוצג בתורה בולט בראש דף הבית. אחד אחר לדירוג הסרטים שנצפו בעבר כך שמשתמש ירצה להמשיך לראות אותו עיגול מספר 7 מסמן זאת. וגם תחת עיגול 5 מוצג סרט חדש שדוגר על ידי משתמשים אחרים ונהייה פופולארי. פלט של כל אחד מהאלגוריתמים יכול להיות מוצג כי רשימות שונות

של סרטים מומלצים לצפייה. יש עוד הרבה אפשרויות להתאמה אישית של סרטים עבור המשתמש כך שאפשר לבנות את המערכת המלצה בצורה מאוד אישית.

4.2 מודל, תהליך למידה ואלגוריתמים

את המימוש של האלגוריתמים המיוצגים בעבודה זו ניתן לראות במחברת Jupyter המצורפת [recommender system.ipynb](#) כולל כל תהליכי הלמידה ובניית המודלים והתוצאות שלהם. במסגרת העבודה, בחרתי לממש את האלגוריתמים החשובים של שיטות המבוססות על סינון שיתופי - פירוק מטריצה ופירוק לערכים עצמיים, וסינון מבוסס תוכן בו השתמשתי בחישוב דמיון בין הסרטים על מנת לייצר רשימת סרטים על פי הדמיון שלהם. כך, המערכת יכולה להמליץ על הסרטים הדומים שכבר נצפו או דורגו על ידי המשתמש. בעזרת ספריית pandas ממשיך מודל המלצה מבוסס שהוא מסוגל לייצר את רשימת הסרטים המדורגים הגבוהה ביותר.

5. סיכום

סקירת מערכות המלצה בשירותי סטרימינג מתמקדת בכלי המיועדים להמלצה על תוכניות טלוויזיה וסרטים בפלטפורמות סטרימינג מובילות כמו נטפליקס, אמזון פריים והולו. המערכות מבוססות על שיטות שונות, כגון פילוח תוכן, סינון שיתופי ומודלים שימושיים מבוססי מלל, ופועלות לצורך המלצת תוכניות וסרטים למשתמשים על בסיס היסטוריית הצפייה שלהם, ביקורות של משתמשים אחרים, ותכונות של הסרטים עצמם.

מערכות המלצה מבוצעות על בסיס נתוני היסטוריית הצפייה של המשתמש ומציעות לו סרטים ותוכניות טלוויזיה שהם עשויים לאהוב, בכדי להשיג חווית משתמש מותאמת אישית ומקובלת. חשיבותן של מערכות המלצה מבוססת על העובדה שהמשתמשים מפספסים תוכניות וסרטים שהם יכולים להתאהב בהם עקב המגבלות הזמניות והרצון לצפות במה שהם יודעים שם מראש. מערכות המלצה בשירותי סטרימינג מהוות כיום כלי חשוב לקידום תוכניות וסרטים.

סקרנו את השיטות השונות לבניית מערכות המלצה והמודלים הסטטיסטיים המשמשים להם. השיטות המרכזיות לבניית מערכות המלצה הן סינון מבוסס תוכן וסינון שיתופי. בסינון מבוסס תוכן, המערכת מציעה מוצרים דומים לפי מאפיינים מסוימים, כמו תיאור, ז'אנר, שחקנים ועוד. בסינון שיתופי, המערכת מציעה מוצרים לפי המידע של משתמשים אחרים עם טעם דומה. ישנם מודלים סטטיסטיים מסוימים כמו פירוק לערכים סינגולריים (SVD) ומודל פירוק מטריצה (MF) וניתוח המשמשים לבניית מערכות שיתופיות.

בנוסף, המאמר דן בכלים שימושיים לבניית מערכות המלצה, כגון ספריות למיפוי נתונים ולביצוע ניתוח סטטיסטי, וכן טכניקות לביצוע סדר על המידע כדי לאפשר למערכות המלצה לפעול ביעילות ובמהירות.

לסיכום, מערכות המלצה הן כלי עז של חברות הסטרימינג להציע מוצרים מותאמים אישית למשתמשים ולשפר את חווית המשתמש.

- [1] Steck, H., Baltrunas, L., Elahi, E., Liang, D., Raimond, Y., & Basilico, J. (2021). Deep Learning for Recommender Systems: A Netflix Case Study. *AI Magazine*, 42(3), 7-18.
- [2] Chen, Jiawei, et al. "Bias and debias in recommender system: A survey and future directions." *ACM Transactions on Information Systems* (2020).
- [3] Schedl, Markus, et al. "Current challenges and visions in music recommender systems research." *International Journal of Multimedia Information Retrieval* 7 (2018): 95-116.
- [4] Fkih, Fethi. "Similarity measures for Collaborative Filtering-based Recommender Systems: Review and experimental comparison." *Journal of King Saud University-Computer and Information Sciences* 34.9 (2022): 7645-7669.
- [5] Steck, H., Baltrunas, L., Elahi, E., Liang, D., Raimond, Y., & Basilico, J. (2021). Deep Learning for Recommender Systems: A Netflix Case Study. *AI Magazine*, 42(3), 7-18.
- [6] Chen, Rui, et al. "A survey of collaborative filtering-based recommender systems: From traditional methods to hybrid methods based on social networks." *IEEE Access* 6 (2018): 64301-64320.
- [7] M. Jalili, S. Ahmadian, M. Izadi, P. Moradi and M. Salehi, "Evaluating Collaborative Filtering Recommender Algorithms: A Survey," in *IEEE Access*, vol. 6, pp. 74003-74024, 2018, doi: 10.1109/ACCESS.2018.2883742.
- [8] Bokde, Dheeraj, Sheetal Girase, and Debajyoti Mukhopadhyay. "Matrix factorization model in collaborative filtering algorithms: A survey." *Procedia Computer Science* 49 (2015): 136-146. 2015
- [9] Sarwar, Badrul, et al. "Incremental singular value decomposition algorithms for highly scalable recommender systems." *Fifth international conference on computer and information science*. Vol. 1. No. 012002. 2002.
- [10] Postmus, Steven, and S. Bhulai. "Recommender system techniques applied to Netflix movie data." *Research Paper Business Analytics, Vrije Universiteit Amsterdam* (2018).
- [11] Gomez-Urbe, Carlos A., and Neil Hunt. "The netflix recommender system: Algorithms, business value, and innovation." *ACM Transactions on Management Information Systems (TMIS)* 6.4 (2015): 1-19.