

אלגוריתם המלצות למשתמש

חלק תיאורטי:

בחנות המשתמשים שלנו תפקיד האלגוריתם הוא לבצע הערכה לאילו משחקים נוספים הלקוח שלנו יאהב ולהמליץ לו עליהם.

האלגוריתם בו השתמשנו לקוח מתחום למידת המכונה ומבוסס על פי עיקרון של User-Based Collaborative Filtering, עקרון זה מתבסס על כך שאנו אוספים מידע על העדפות של המשתמשים, נשים לב שכדי לאסוף העדפות של המשתמשים השתמשנו במערכת דירוג עבור המשחקים שבאמצעותה המשתמש יכול לתת דירוג של 0.5-5 בקפיצות של חצי כאשר הדירוג 0 אומר שהמשתמש לא דירג עדיין את המשחק.

מערכת הדירוג מאפשרת לנו לקבל מידע על העדפות של כלל המשתמשים באמצעות הדירוגים השונים נוכל למצוא משתמשים הדומים בהעדפותיהם אל המשתמש שלנו נרצה להמליץ על משחקים.

אילוסטרציה:

משתמש	משתמש	משתמש	משתמש	משתמש	
1	2	3	4	5	
משחק 1	1	5	5	1.5	
משחק 2	2.5	4	1	2.5	0.5
משחק 3	1	3.5	2	3	2.5
משחק 4	3	5	3.5	2.5	5
משחק 5	1	2	0	1	4
משחק 6	5	2	2	4.5	3
משחק 7	4.5	4.5	1	5	2
משחק 8	5	5	4	3	2

על מנת לחשב את הדמיון בין משתמשים נשתמש בחישוב באמצעות שיטת cosine similarity שהנוסחה שלו היא:

$$\text{similarity}(x, y) = \cos(\vec{x}, \vec{y}) = \frac{\vec{x} \times \vec{y}}{\|\vec{x}\| \times \|\vec{y}\|} = \frac{\sum_{i \in I_{xy}} r_{x,i} \times r_{y,i}}{\sqrt{\sum_{i \in I_x} r_{x,i}^2} \times \sqrt{\sum_{i \in I_y} r_{y,i}^2}}$$

\vec{x} - וקטור הדירוגים של משתמש

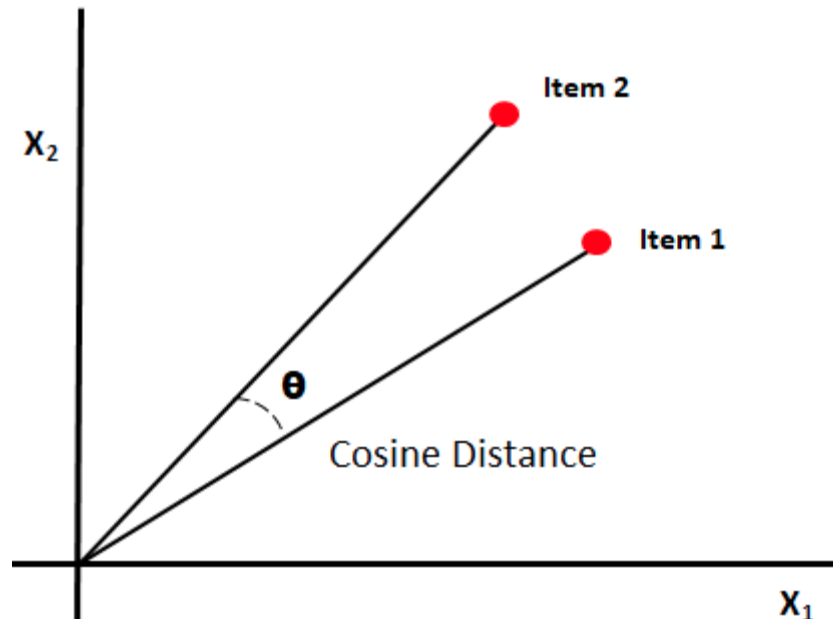
\vec{y} - וקטור הדירוגים של משתמש

$r_{j,i}$ - דירוג של משתמש j עבור משחק i

I_{xy} - משחקים ששני המשתמשים דירגו

I_j - משחקים שמשתמש j דירג

Cosine Distance/Similarity



נשים לב שניתן לראות המחשה של cosine similarity כאשר ככל שהמרחק בין הווקטורים (cosine distance) קטן כך הם דומים כאשר במערכת שלנו הווקטורים יהיו מורכבים מהצבעות של כל משתמש ככל ששני משתמשים הצביעו דומה כך הווקטורים שלהם יהיו קרובים יותר.

לאחר שחישבנו את similarity בין כל המשתמשים שלנו לבין המשתמש שלו אנחנו רוצים להמליץ, נשתמש בשיטה K-Nearest neighbors כלומר ניקח את k השכנים הקרובים (אצלנו בתוכנית לקחנו $k=5$) אלינו ונעשה ביניהם חישוב על מנת למצוא את הדירוג שנצפה שהמשתמש ידרג את המשחקים השונים שלא דירג זאת באמצעות הנוסחה הבאה:

$$r_{u,i} = \bar{r}_u + \frac{\sum_{u \in U} \text{similarity}(u, \hat{u}) \times (r_{\hat{u},i} - \bar{r}_{\hat{u}})}{\sum_{u \in U} |\text{similarity}(u, \hat{u})|}$$

$r_{j,i}$ - דירוג של משתמש j עבור משחק i

\bar{r}_j - דירוג ממוצע של משתמש j

$\text{similarity}(u, \hat{u})$ - דמיון בין משתמש u לבין משתמש \hat{u}

לאחר החישוב נקבל את הדירוג שהיינו מעריכים שהמשתמש היה מדרג את המשחק נעשה את החישוב הזה לכל המשחקים שלא קנה/ דירג ומתוכם ניקח רק את 3 המשחקים עם הדירוג הגבוה ביותר ועליהם נמליץ למשתמש.

חלק מעשי:

בתוכנית שלנו פיצלנו את האלגוריתם לכמה פונקציות

חלק 1:

```
public static Vector<Integer> recommendedAlgo(int clientId)
```

הפונקציה העיקרית היא קוראת לשאר הפונקציות כמו כן שולחת בסופו של דבר את 3 מספרי הזיהוי של המשתמשים שהמלצנו עליהם לפי בעל הדירוג הגבוה ביותר ראשון וכן הלאה.

חלק 2:

```
public static HashMap<Integer, HashMap<Integer, Double>>  
getRatingTable()
```

פונקציה זאת יוצרת לנו טבלה בעצם עם כל הדירוגים של המשתמשים לכל המשתמשים השונים שיש לנו היא מחזיקה את הטבלה בתוך HashMap שממופת על ידי מספר משתמש ומכילה HashMap שמכילה בתוכה את הדירוגים של כל משתמש למשתמשים השונים.

חלק 3:

```
public static HashMap<Integer, Double> averageRating()
```

פונקציה זאת מחשבת לכל אחד מהלקוחות את הדירוג הממוצע מתחשבת רק במשתמשים שדירג פונקציה זו תחזיר לנו HashMap שממופת על ידי מספר משתמש ומכילה את הדירוג הממוצע של כל משתמש.

חלק 4:

```
public static HashMap<Integer, Double>  
cosineCorrelation(HashMap<Integer, Double> ourUserRate, int clientId)
```

פונקציה זאת מחשבת את דמיון בין המשתמש לשאר המשתמשים שלנו לפי הנוסחה ומחזירה HashMap שממופת על ידי מספר המשתמש ומכילה בתוכה את ערך ה-similarity.

חלק 5:

```
public static Vector<Integer> fiveNeighbors(HashMap<Integer, Double>  
cosineCorrelation)
```

פונקציה זו מוצאת עבורנו את המשתמשים בעלי פקטור הדמיון הגדול ביותר למשתמש שלנו ומחזירה את מספרי המשתמש שלהם אם אין לפחות חמישה משתמשים במערכת תשים את כל המשתמשים שיש.

חלק 6:

```
public static HashMap<Integer, Double> expectedRating(Vector<Integer>  
similar, HashMap<Integer, Double> cosineCorrelation, int clientId,  
HashMap<Integer, Double> averageRateUsers)
```

פונקציה המחשבת את הדירוג המוערך של כל אחד מהמשתמשים שהמשתמש לא קנה/ דירג ומחזירה HashMap שממופת על ידי מספר המשתמש ומכילה את ערך הדירוג הצפוי שלו.

חלק 7:

```
public static Vector<Integer> ostRecomeddedGames(HashMap<Integer,  
Double> expectedRating,HashMap<Integer, Double> ourUserRate, int  
clientId)
```

פונקציה שמחזירה מתוך את הדירוגים המוערכים את מספרי שלושת המשחקים בעלי
הדירוג הגבוהה ביותר שהמשתמש עוד לא קנה/דירג.