יבגני גאיסינסקי

רועי אמזלג

ברוך רוטמן

פרויקט מערכות המלצה

פרק 1:

את האוסף נתונים לקחנו ממאגרים של Movielens ו-Imdb. נעשה שימוש בתכונות ספציפיות שמעניינות אותנו בשביל המערכת המלצה כגון: שם הסרט, מספר מזהה של סרט, שנת ייצור, סוג סרט, תיאור סרט, במאי, מספר מזהה של משתמש וציוני סרטים שמשתמש נתן לכל סרט.

בגלל שלקחנו נתונים ממאגרים שונים נעשה השוואה בין מספר מזהה של סרט ואותו מספר שכל משתמש נתן לו ציון כך שיהיה לנו מידע שלם על סרטים וציונים של משתמשים שונים.

בשביל שדאטסט שלנו יהיה יותר רלוונטי, יעיל ויצטרך פחות ביצועים עשינו סינון ראשוני, כך שכל משתמש ייתן כמות ציונים סבירה לכלל הסרטים וגם שכל סרט במאגר יקבל דירוג מספיק מרשים ושיהיה למודל שלנו ממה ללמוד ולתת המלצה יותר רלוואנטית.

לפני סינון

:

	userId	movieId	rating
0	1	2	3.5
1	1	29	3.5
2	1	32	3.5
3	1	47	3.5
4	1	50	3.5
...	...	...	...
20000258	138493	68954	4.5
20000259	138493	69526	4.5
20000260	138493	69644	3.0
20000261	138493	70286	5.0
20000262	138493	71619	2.5

20000263 rows × 3 columns

אחרי סינון

	userId	movieId	rating
0	1	2	3.5
1	1	29	3.5
2	1	32	3.5
3	1	47	3.5
4	1	50	3.5
...	...	...	...
14808830	138493	68319	4.5
14808831	138493	68954	4.5
14808832	138493	69526	4.5
14808833	138493	69644	3.0
14808834	138493	70286	5.0

14808835 rows × 3 columns

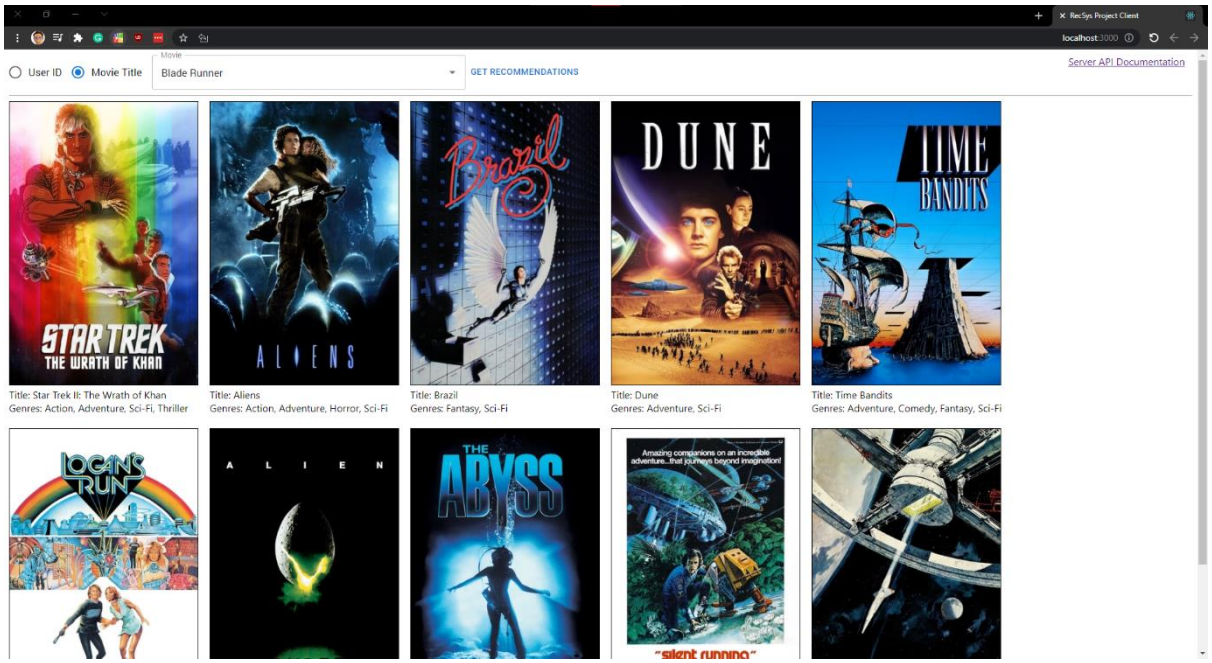
קישור למאגר Movielens: MovieLens 20M Dataset | Kaggle

קישור למאגר Imdb: IMDb Dataset | Kaggle

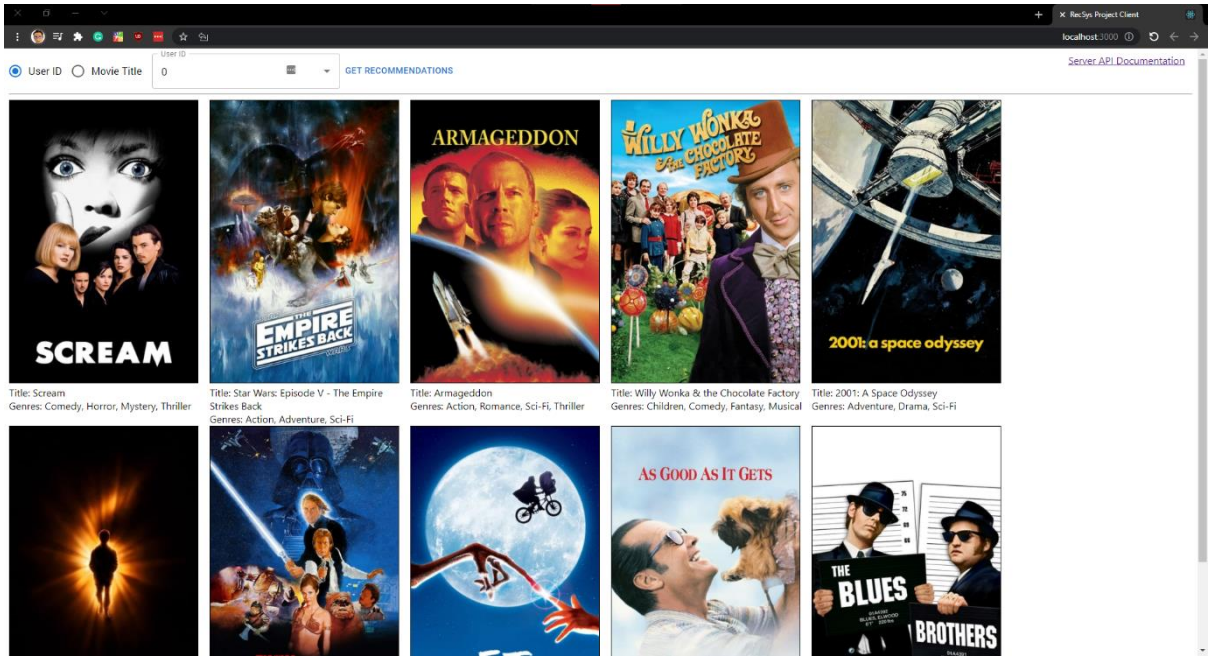
פרק 2:

במערכת קיימות שתי אופציות בחירה:

ראשונה זה קבלת המלצות לפי שם הסרט. מקלידים שם של סרט לשדה המיועד ואז מקבלים רשימת סרטים ותכונותיהם לפי מערכת ההמלצה.



אופציה שניה זה לפי משתמש שרשום במערכת. בוחרים משתמש מסוים ולפי הדירוג שהוא נתן לסרטים נקבל סרטים הכי מתאימים לפי משתמשים שדירגו סרטים באופן דומה או קרוב.



מערכת מקבלת קובץ שעבר עיבוד מקדים עם משתמשים וסרטים ותכונותיהם. זה הנתונים העיקריים במערכת. משווים מספרי זהות של סרטים עם קבצים אחרים ושולפים נתונים נוספים כמו סוג של סרט.

אחרי שקיים מאגר שלם מחלקים דאטה לקטגוריות ספציפיות כך שנוכל לטפל בכל תכונה בנפרד לצורך בניית מודל.

מאמנים מודל לפי תכונות שנבחרו ועושים התאמה.

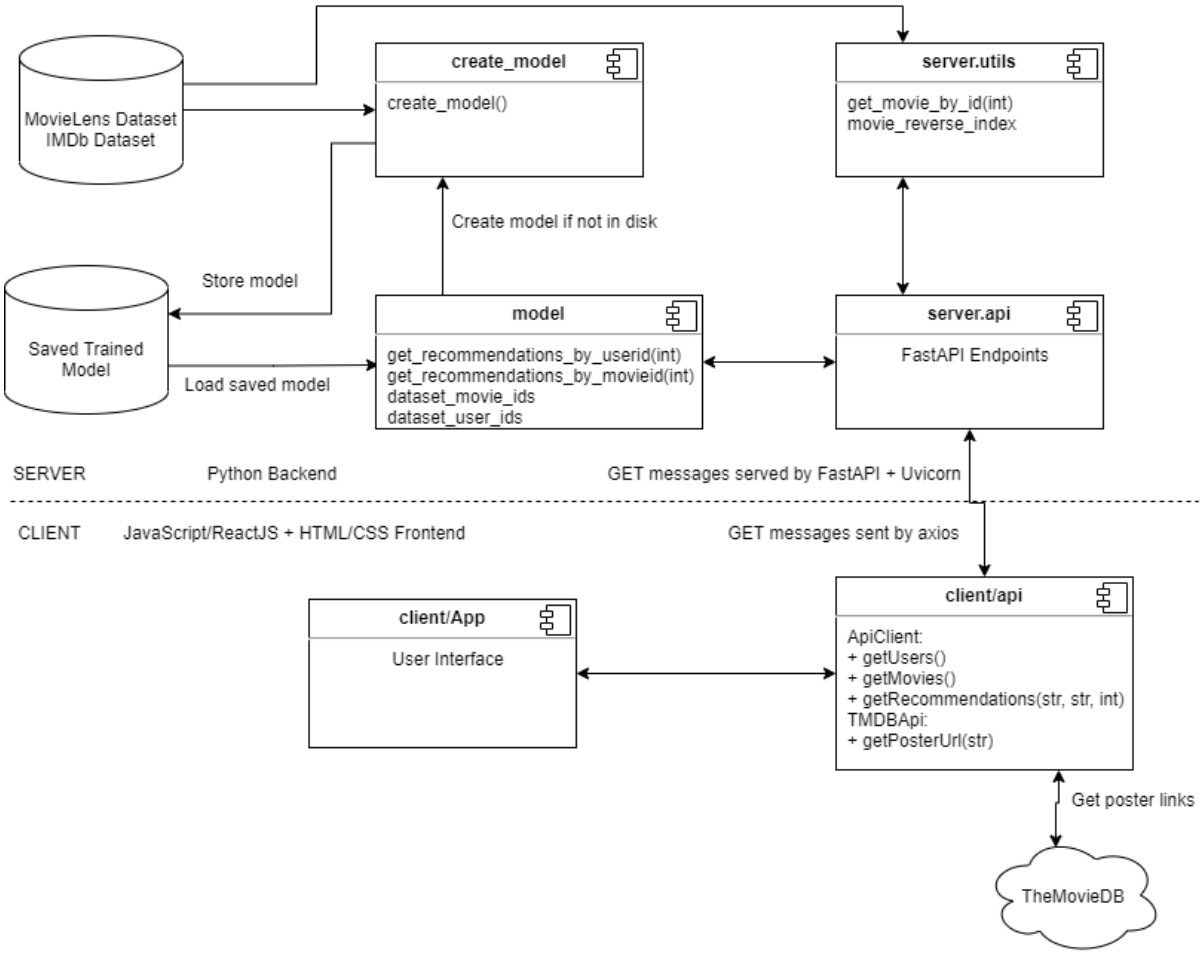
אחר כך עושים חלוקה בין ניבוי סרטים לפי ציונים ושאר התכונות שלהם שלהם או לפי משתמשים שנתנו ציונים ואז מוצאים הדיוק הגבוהה כדי לתת המלצה.

ואז בוחרים את כמות התוצאות הרצויה.

כדי להשתמש במערכת משולבת היברידית נעשה שימוש ב-hybrid matrix factorization וייצוג נתונים נעשה במטריצות של נתונים. כל תכונה מקבלת נתונים מספריים כמו דירוג סרט או טקסטואליים כמו שם של סוג הסרט.

המודל שבחרנו להשתמש בה נקרא **LightFM** שהיא בעצם משלבת תכונות של content-based and collaborative filtering. כלומר השתמשנו במודל היברידי.

המודל הזה משתמש בפריות **LightFM** לצורך אימון מודל וספריות שונות של **recommenders** לצורך ביצוע הערכות שונות ששיפור מערכת. וגם ספריות **seaborn** ו-**matplotlib** לצורך וויזואליזציה והערכות.



פרק 3:

בבניית מערכת בחרנו לבנות שתי מודלים. אחת עם תכונות דירוג בסיסיות ושנייה עם תכונות דירוג פשוטות וגם תכונות נוספות של משתמשים וסרטים.

קודם כל בבניית המערכת נעשה עיבוד מקדים של נתונים כדי לייעל את תהליך הבניית המלצות. לכל מאגר של משתמשים וסרטים עשינו סינון כך שכל הסרטים ומשתמשים יהיו רלוונטיים ויהיה להם ערך גבוהה ומשמעותי בבניית המלצה.

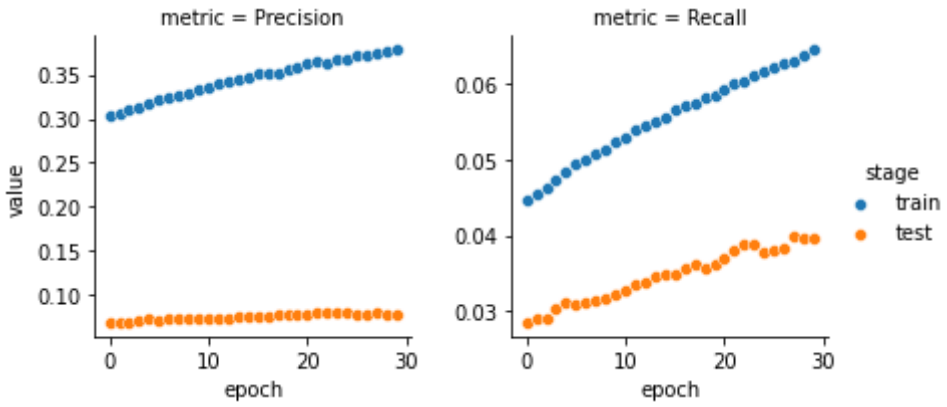
סה"כ השתמשנו ב-5000 סרטים ו-100000 משתמשים שקיבלנו ונתנו הכי הרבה ציונים. כך מאגר הנתונים שלנו יהיה הרבה יותר יעיל ומשפיע.

לצורך הערכה בחרנו להשתמש ב-**recall** ו-**precision**. בהמשך נראה וויזואליזציות רלוונטיות למדד הזה ומה משפיע עליו.

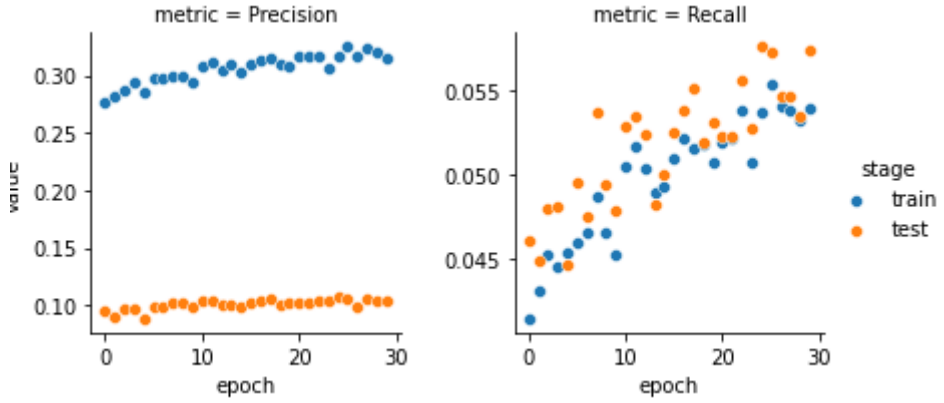
למודל קיימת כמות פרמטרים שמשפיעים על תוצאותיה אבל אנחנו התשמשנו בכמות ה-**epochs** ו-**split** של נתונים לצורך השוואה בין תוצאות וחיפוש פרמטרים הכי מתאימים.

אחרי שהגדרנו כמות ה-**epochs** להיות 30 קיבלנו תוצאות הבאות:

עבור מודל ראשון:

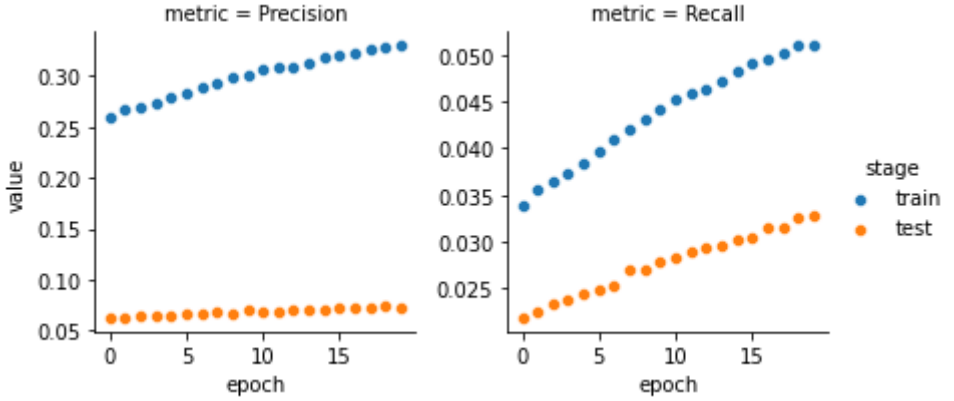


עבור מודל שני:

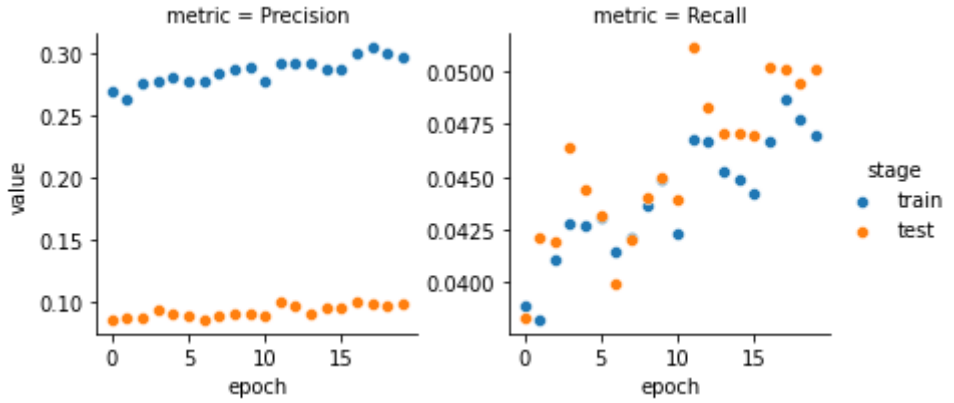


אחרי שהגדרנו כמות ה-**epochs** להיות 20 קיבלנו תוצאות הבאות:

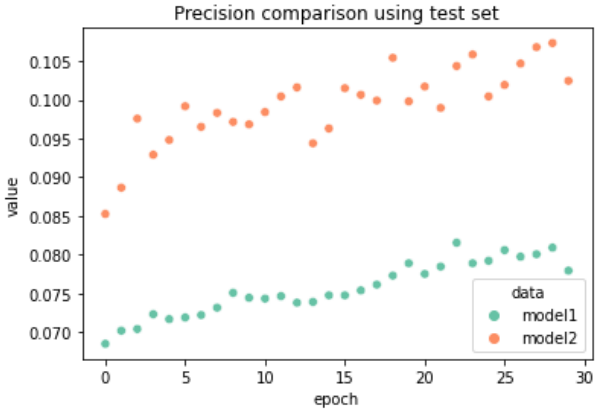
עבור מודל ראשון:



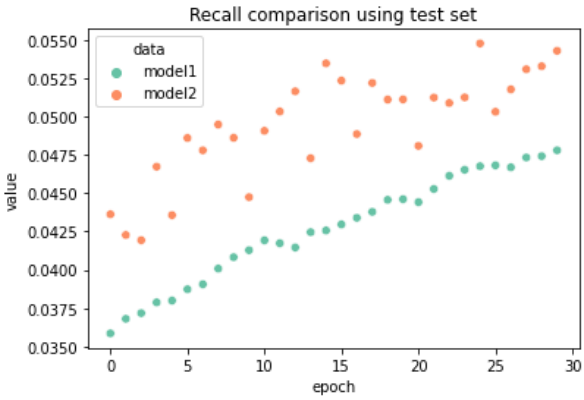
עבור מודל שני:

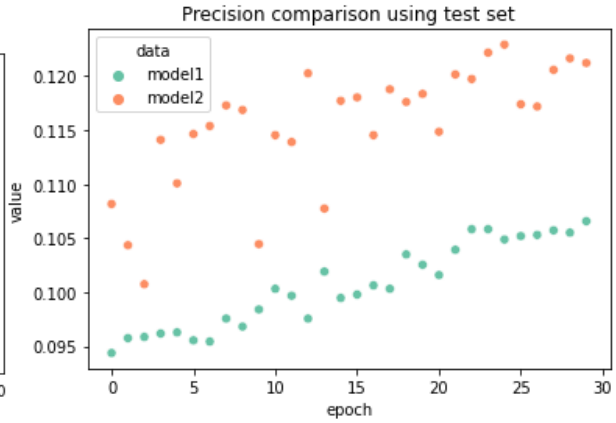


אחרי שהגדרנו אחוז ה-**split** להיות 25 קיבלנו תוצאות הבאות:



אחרי שהגדרנו אחוז ה-**split** להיות 30 קיבלנו תוצאות הבאות:





מסקנות:

כפי שאנחנו רואים כמות ה-epochs יותר טובה במספר 30 כי אז התוצאות של נתוני אימון ובדיקה הן יותר קרובים אחד לשני ומקבלים תוצאות יותר קרובות ומדויקות.

ואחרי שחלקנו מתונים לאימון ובדיקה עשינו נתוני בדיקה של 25 אחוז אז כל התוצאות היו יותר קרובות אחד לשני בכל שלב ולפי גרף אפשר לראות את זה בקלות לפי הנקודות פחות מפוזרות מאשר ב30 אחוז.

בנוסף אפשר לראות במדויק שמודל השני שהוא עם שימוש ביותר תכונות של משתמשים וסרטים הוא מודל עם תוצאות הרבה יותר נכונות וקרובות לאמיתיים ולכן המודל הזה מתאימה לנו הרבה יותר.

פרק 4:

בפרויקט זה נעשה שימוש ולימוד עם מערכות היברידיות שמשתפות סינון שיתופי וסינון מבוסס תוכן.

למדנו ספרייה חדשה שמשתמשת בשיטת מערכת היברידית ואיזה תכונות מתאימות לנו לצורך ייעול ניבוי המלצות. בנוסף עישנו שימוש בפרמטרים שונים לצורך הערכת מודל ובחירת פרמטרים מתאימים.

בנינו אתר עם ממשק נוח למשתמש שכולל חיפוש ברטים לפי שמות שלהם וגם לפי דירוג של משתמשים שונים לצורך חיפוש התאמות בין משתמשים והצעת סרטים רלוונטיים.

במהלך הפרויקט נתקלנו במספר אתגרים שהיינו צריכים להתגבר עליהם.

אחד מהם זה כמות הנתונים גדולים של משתמשים וסרטים שדורשים המון זיכרון וביצועי מחשב. כדי להקל על זה וגם לשמור על איכות המערכת ואפילו שיפורה, ניקינו נתונים כך שהשארנו סרטים עם הכי הרבה ציונים ומשתמשים עם הכי הרבה דירוגים. וכך קיבלנו תוצאות יותר רלוונטיות.

בנוסף אחד הבעיות הייתה התקנת הספריות הרלוונטיות. חלק מהם דורשים ספריות נוספות וחלקם דורשים עדכון או מחיקה. אז אחרי חיפוש ברשת מצאנו פתרון ועשינו התקנה מסודרת.

בסוף השתמשנו במודל שמשתמשת בתכונות נוספות של משתמשים וסרטים. ברגע שמוסיפים יותר נתונים, ככב למערכת יש יותר מה ללמוד ותוצאות יוצאות יותר מדויקות. עשינו השוואה של מודלים לפי פרמטרים שונים ובערת וויזואליזציות שונות מצאונו פרמטרים הכי מתאימים למערכת יותר יעילה.

לצורך שיפור המערכת היינו מחפשים יותר תכונות למשתמשים וסרטים. ככה יהיה למערכת יותר כיוונים לאן להסתכל וממה ללמוד כדי למצוא קשרים חדשים וסרטים רלוונטיים. בנוסף תמיד אפשר להגדיל את המאגר נתונים כך שיהיה יותר משתמשים וסרטים ויהיה אפשר להשתמש יותר ברשתות נאורוניות ולקבל שיפור בנתונים. כמובן צריך גם שיהיו מספיק דירוגים כדי שיהיה ממה ללמוד.

קישור לסרטון של מערכת: https://youtu.be/BH7AsB_np90