דוח מסכם

הערה חשובה:

זמן הריצה של חילוץ הנתונים ארוך מאוד! (כשלוש שעות), לאחר שיחה עם חן הוספנו בקוד הוצאה של קובץ הנתונים לCSV וקריאה מחדש שלו ולכן ישנם 2 קבצי קוד בגיטהאב.

מגישים:

לידור דהרי

טוהר רחמין

סעיף 1 - מבוא

בפרויקט שלנו בחרנו להתמקד בהמלצת סרטים על פי הסרט האחרון הנצפה ע"י המשתמש.
הבעיה שרצינו לפתור היא המצב בו יש יותר ממשתמש אחד שמשתמש במערכת ההמלצה (לדוגמא זוג או משפחה), כך שמ"ע ההמלצה מציעה מגוון סרטים בהתאם למשתמשים שלה (בדוגמה של הזוג- הגבר חובב סרטי אימה והאישה קומדיות, נרצה שלאחר ראיית סרט אחד המערכת תמליץ למשתמש על הסרטים שהוא אוהב).
המטרה שלנו היא להגיע למצב בו כשהמשתמש רואה סרט שאותו הוא אוהב, תינתן לו אפשרות לראות סרט כמה שיותר דומה לסרט בו הוא צפה.
אנחנו רוצים להמליץ על מספר מצומצם של סרטים בכל פעם ובכך למנוע את הדפדוף הארוך (מרוב היצע אפשרויות).

סעיף 2 - ערכת נתונים ותכונות

שלב א- חילוץ הנתונים:

בפרויקט שילבנו 2 סוגי API- השתמשנו ב BeautifulSoup ע"מ לחלץ מהאתר IMDb את המאפיינים- שם הסרט, סוג ז'אנר, שנת יציאה, משך, דירוג הסרט ע"י המשתמשים, דירוג הסרט ע"י האתר, הצבעות, תיאור.
עברנו על 10 דפים של האתר ובכך חילצנו נתונים של 1000 סרטים.
נוכחנו לדעת שהנתונים לא מספקים ולכן השתמשנו בספריית IMDb ע"מ להוסיף לדאטה את הנתונים הבאים:
שפות בהם נתמך הסרט, גודל הקאסט, שם הבמאי, עלילה, האם הסרט חלק מסדרה(כמו הארי פוטר, שר הטבעות).

<u>שלב ב- עיבוד וניקוי הדאטה:</u>

- יצרנו עמודות נוספות שנתוניהם נלקחו מהעמודה המכילה את הנתונים שאספנו מהספרייה IMDb בשלב הקודם.
- ניקנו את הדאטה מסימני פיסוק וערכים חסרים (את חלקם השלמנו ואת חלקם הורדנו)
- **<u>המרת עמודות מספריות לערכים בינאריים:</u>**

  עמודת -Year of release החלטנו לחלק עפ"י שנת 1995 כיוון שבשנה זו היה גידול רב בצפיית הסרטים (עפ"י מידע מהאינטרנט), כך שסרטים שיצאו לפני 1995 יקבלו את הערך 0 והשאר 1.

  עמודת (Runtime (Minutes- הצגנו בגרף את התפלגות אורך הסרטים וחילקנו את הסרטים על פי הממוצע (מתחת לממוצע- 0 , והשאר 1)
- **<u>התמודדות עם עמודות קטגוריות:</u>**

  במאי- לקחנו את 20 הבמאים בעלי הכי הרבה סרטים בציון גבוה וחילקנו את העמודה על פיהם (סרט שהבמאי שלו מהרשימה יקבל 1, השאר 0)

  עלילה- בשלב הראשוני טיפלנו בדאטה בעזרת NLP- הורדנו stop words, והגענו לשורשי המילים בעזרת lemmatize .

  לאחר מכן חישבנו עבור כל מילה את ערך IDF*TF ולקחנו את 20 המילים בעלות הערך הגבוה ביותר וחילקנו את העמודה על פיהם (סרט שמופיעות בו אחת מהמילים יקבל 1, השאר 0)

  ז'אנר- ערך הז'אנר הגיע כרשימה המכילה את הז'אנרים שלהם שייך הסרט. מכלל הרשימות יצרנו רשימת ערכיים ייחודים פתחנו, עבור כל סוג ז'אנר עמודה ובה 1 אם הסרט שייך לז'אנר ו0 אם לא
- לאחר הטיפול בעמודות ומיכוון שיש לנו מספר סקאלות (שנים, הצבעות וכו) עשינו סטנדרטיזציה על הדאטה המספרי.
- הורדת מימדים בעזרת PCA – מיכוון שנוצרו מספר רב של עמודות, הורדנו מימים (השארנו 90% מהדאטה המקורי).

סעיף 3 - מתודולוגיה

- **PCA**:

עקב כמות העמודות הקטגוריות בדאטה נוצרו לנו מספר רב של עמודות בינאריות כאשר לכל אחת משקל נמוך.

עקב העובדה כי כמות המימדים (פיצ'רים) משפיעה לרעה על מודלי ה clustering השתמשנו בשיטה ע"מ להוריד מימדים אך לא לפגוע בדאטה.

- **DBSCAN** :

ניסינו לבצע clustering בעזרת מודל DBSCAN במחשבה שהעובדה כי המודל עובד על צפיפות תעזור לנו.

- **EM**:

בחרנו במודל מכיוון שהוא אפקטיבי יותר מ K-means ויכול לתאר מספר רב יותר של צורות

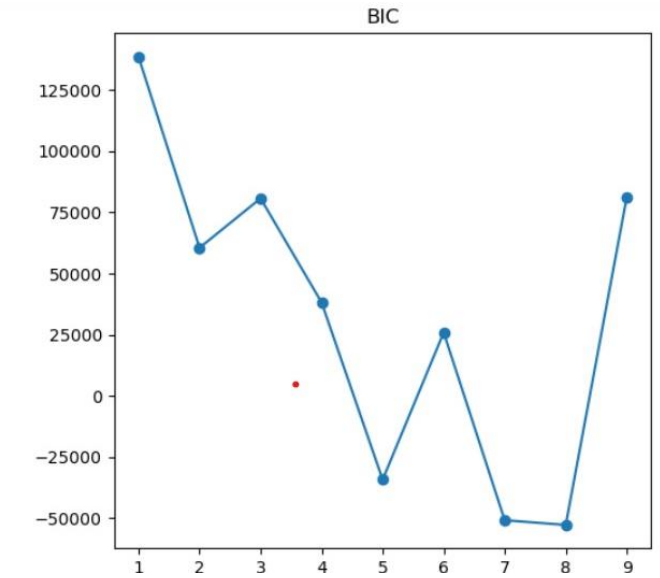
- **מציאת דמיון:**

ע"מ למדוד את מ"ע ההמלצה מצאנו את הדמיון בין הסרטים שהומלצו לסרט הנצפה.

מדדנו את הדמיון בעזרת זווית הקוסינוס בין 2 הוקטורים מ2 סיבות:

1. את המרחקים בין הסרטים מצאנו בעזרת מרחק אוקלידי, רצינו למצוא דמיון בעזרת קוסינוס כך שהסרטים יהיו דומים בשני מדידים שונים

2. זווית הקוסינוס מוצאת את הדמיון בין וקטורים כך שאנחנו יכולים להשתמש בדאטה לפי הPCA והסטנדרטיזציה ע"מ למדוד את הדמיון. זאת דרך נוספת למדוד את הדמיון ולוודא שהמניפולציות שעשינו על הדאטה לא הזיקו.



סעיף 4 -ניסויים/תוצאות/דיון

- **DBSCAN**:

הרצנו את המודל על טווח של epsilon ,min_samples ע"מ למצוא את הערכים שיבנו מודל שיגיע לsilhouette גבוה, אך נוכחנו לדעת כי ערכי ה silhouette היו נמוכים כך שלא הצלחנו להגיע למספר קבוצות מיטבי.

- **EM:**

הרצנו את המודל עבור טווח קבוצות כאשר לכל מספר חישבנו את ערך BIC שלו (מקביל לSSE ב K-mean), ומצאנו כי הערך המינימלי הוא כאשר יש 6 קבוצות

- **ערך BIC:**

מדדנו את מודל EM בעזרת ערך BIC מיכוון שהוא מוצא את הפשרה האופטימלית בין מספר הפרמטרים הנדרשים לתיאור האשכולות מצד אחד, לבין הסבירות לתוצאת האשכול מצד שני.

- **מדדי הערכה- דמיון:**

בחרנו לא לעשות ממוצע על הדמיון בין הסרט שנצפה לסרטים שהוחזרו מיכוון שבעניינו שלושת הסרטים אמורים להופיע יחד על המסך ולא אחד אחרי השני, כדי למנוע דפדוף מצד המשתמש.

- **תוצאות:**

לאחר החלוקה ל clustering הוצאנו את שלושת הסרטים שהכי קרובים באותו cluster של הסרט הנצפה אחרון. חישבנו את הדמיון בין הסרט הנצפה לסרטים שהוחזרו (הרצנו מספר פעמים וראינו כי מדד הדמיון גבוה)

- **מבט מקיף:**

המגבלות בפרויקט היו בעיקר ההתמודדות עם העמודות הקטגוריות בדאטה, וכן מגבלות בחילוץ הנתונים שהוביל לכך שלא היו נתונים מסוימים שחשבנו שהיו יכולים להועיל בדאטה.

ציפינו לכמות קבוצות גבוה יותר משום שישנם המון סוגי סרטים והמון דרכים לסווגם. ההצלחה נבעה בעיקר מהטיפול בעמודות הקטגוריות משום שהם היו מרובות וטיפול לא נכון היה מוביל לאי הבנת הנתונים ע"י המודל.

סעיף 5- סיכום ועבודה עתידית

הפרויקט משליך על אופציות לקיום מערכת המלצה שתפעל על נתוני עבר אחרונים כדי להקל על מספר משתמשים במערכת.

ניתן להמשיך לחקור את הנושא ולהכניס עוד מאפיינים הנוגעים לסרט- תקציב, הכנסות, פרסים. אפשר להעמיק את נושא העלילה כדי שתוכן הסרט יקבל ביטוי גדול יותר.
אפשר להכניס את נושא השחקנים- לתת משקל גדול יותר בהמלצה לסרט בעל אחד הכוכבים שנראו בסרט קודם.

סעיף 6- תרומות

לידור- שליפת נתונים בעזרת BeautifulSoup, ניתוח וטיפול בעמודות קטגוריות, מודל DBSCAN
טוהר- שליפת נתונים בעזרת ספריית IMDb , טיפול בעמודת עלילה, מודל EM, מדידת דמיון