למידה לא מפוקחת - עבודת סיום

מגיש: רותם גזית, ת"ז 318446697

סמסטר חורף 2020

תוכן העניינים

1 הקדמה

המידע אותו אני מנסה לנתח כאן הוא ניתוח הסרטים המופיעים בנטפליקס[1], כשהמטרה הכללית היא לחלק אותם בסופו של דבר לקבוצת סרטים ש"שווה לצפות בהם" וקבוצת סרטים ש"לא שווה לצפות בהם".

המידע מכיל בעבור כל סרט את שמו, שנת הפצתו, אורכו, הבמאים והשחקנים שלו, דירוג קבוצת הגיל המומלצת לצפייה, הקטגוריות אליו שויך הסרט בנטפליקס, המדינות בהן הופק וצולם, ותיאור הסרט במילים.

show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
60011649	Movie	Indiana Jones and the Raiders of the Lost Ark	Steven Spielberg	Harrison Ford, Karen Allen, Pau	United States	January 1, 2019	1981	PG	116 min	Action & Adventure, Children & Family Movies, Classic Movies	When Indiana Jones is hired by the government to locate the legendary A
60010488	Movie	Indiana Jones and the Temple of Doom	Steven Spielberg	Harrison Ford, Kate Capshaw, A	United States	January 1, 2019	1984	PG	119 min	Action & Adventure, Children & Family Movies, Classic Movies	Indiana Jones, his young sidekick and a spoiled songbird get more than t
60010487	Movie	Indiana Jones and the Last Crusade	Steven Spielberg	Harrison Ford, Sean Connery, D	United States	January 1, 2019	1989	PG-13	127 min	Action & Adventure, Children & Family Movies, Classic Movies	Accompanied by his father, Indiana Jones sets off on his third adventure

את המידע הצמדתי לשני נתונים נוספים: מה הדירוג הממוצע (1 עד 10) שנתנו לו צופים באתר IMDB[2], ובכמה פרסי אוסקר זכה הסרט[3]. הקוד לתהליך ההצמדה נמצא בנתיב python_code/data_prep והוא מכיל מספר ניואנסים קטנים (למשל חלק מהמקומות השתמשו ב & במקום במילה and בשמות חלק מהסרטים). לאחר תהליך העיבוד הראשוני המידע נשמר בקובץ data/netflix/stage.json.

ביציאה לדרך אני מניח מספר הנחות עבודה על עולם הבעיה:

1. סביר להניח שישנם שחקנים טובים ושחקנים גרועים, וששחקן טוב ישחק בעיקר בסרטים טובים (וסרט טוב בעיקר יכיל שחקנים טובים).
   כלומר, טיב הסרט יכול להיות קשור לסרט אחר לפי השחקנים שבו.
2. פרס אוסקר הוא מדד אבסלוטי לאיכות הסרט. הייתי רוצה למצוא קורולציה בין "קבוצת הסרטים ששווה לצפות בהם" והזכיות שלהם בפרסי אוסקר.
3. ישנם סינונים בסיסיים שאני יכול לעשות שימקדו אותי במציאת "קבוצת הסרטים ששווה לצפות בהם", למשל:
   (א) ככל הנראה שלא אכליל סרטים הודים בתוך הקבוצה הזו, היות וזה לא הטעם שאני מחפש בסרט טוב.
   (ב) דירוג נמוך מאוד לסרט ב IMDB הוא חיתוך פשוט שאפשר לעשות. למשל כל סרט שקיבל דירוג פחות מ 5, ישירות יעבור ל"קבוצת הסרטים שלא שווה לצפות בהם".

תכנית העבודה היא כזו:

1. לבצע סינונים נאיביים לפי הנחות העבודה שלי על המידע
2. ליצור גרף קשרים בין הסרטים, בהתבסס בעיקר על השחקנים המשותפים בניהם
3. למצוא קהילות בגרף במגוון שיטות: אלגוריתם Louvain, או לחילופין ביצוע MDS ולאחריו אלגוריתמים לזיהוי קהילות במרחב
4. לאחד את הקהילות שנמצא לשתי קבוצות - קבוצת "הסרטים הטובים" וקבוצת "הסרטים הלא טובים"
5. לבצע מבחנים לאיכות הקהילות שמצאנו - מבחנים סטטיסטיים למובהקות ההפרדה שיצרנו, ואל מול ה "Groudtruth" של סרטים טובים - שהוא מבחינתי זכייה בפרסי אוסקר.

הנדסת תכנה

הפרויקט נבנה כך שיהיה ניתן בקלות מאוד להחליף את שיטת הקליסטור או את שיטת בניית הגרף. לצורך כך קיימים שני Interfaces:

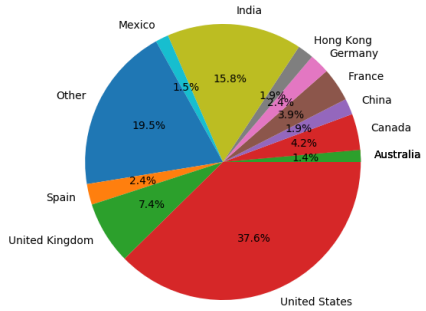
1. עבור יצירת הקשתות לגרף - IEdgeLogic
   מקבל את ה dataframe ומחזיר קשתות ממושקלות
2. עבור חלוקה לקהילות - ICluster
   מקבל את הגרף ומחזיר שיוך של כל node לקהילה

השיטה הזו מאפשרת לנו להחליף בסה"כ את המימוש הנבחר לאותו ממשק ב app.py כדי להחליף שיטת עבודה ולבדוק מיד את התוצאות. כל אחד מהמימושים יוסבר כאן ובנוסף מתועד כ docstring בקוד עצמו.

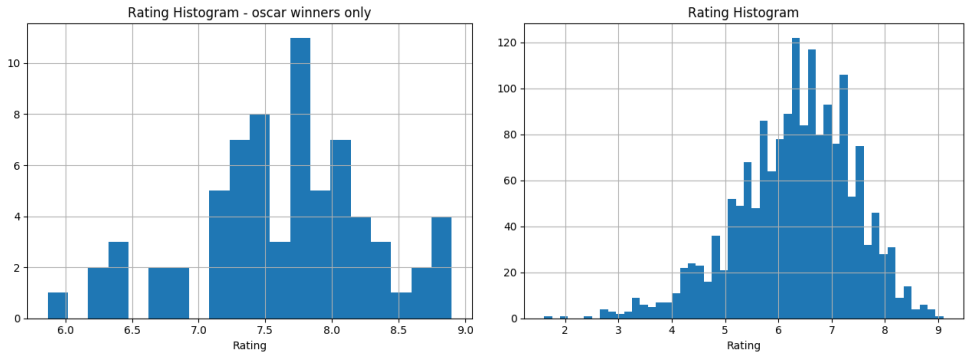
[1] https://www.kaggle.com/shivamb/netflix-shows/tasks?taskId=123
[2] https://www.kaggle.com/ashirwadsangwan/imdb-dataset
[3] https://www.kaggle.com/unanimad/the-oscar-award

2 ניתוח המידע

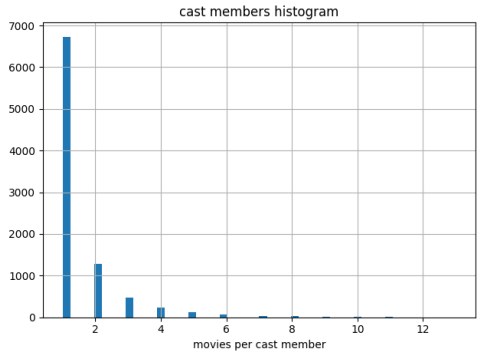
כלל התרשימים המוצגים נוצרים בקובץ python_code/learning/stats/data_analysis.py.
התרשים הבא מציג את הפילוח של הסרטים לפי מדינת המקור שלהן:



אני מסנן רק על סרטים אמריקאים ובריטים כי זו קבוצת העניין שלי. הסינון מתבצע ב python_code/learning/data_loader.py.
שני התרשימים הבאים מציגים את היסטוגרמת הדירוגים שניתנו לסרטים הללו בנטלפיקס, ואת היסטוגרמת הציונים של סרטים זוכי אוסקר.



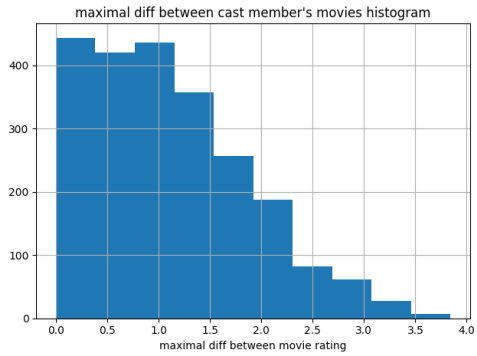
אפשר לראות מכך שני דברים מעניינים: דבר ראשון - סרט זוכה אוסקר מקבל תמיד ציון של מעל 5, אך אולי בניגוד למצופה - אין מתאם חד משמעי בין הציון לזכייה. כשמחשבים את מקדם מתאם פירסון בין שני הנתונים[4] מקבלים $R = 0.391695$, כלומר קשר ליניארי חיובי אך לא משמעותי מאוד.
כיוון שאין טעם להתעסק בסרטים עם דירוג נמוך מאוד, אנחנו מבצעים סינון על סרטים שקיבלו רק דירוג גדול מ 5: data_loader.py
התרשים הבא הוא היסטוגרמה של כמות הסרטים שכל שחקן שיחק בהם. ניתן לראות שבאופן מובהק רוב השחקנים (במידע שלנו) הופיעו בסרט יחיד.



נבדוק את הנחת העבודה שלנו - "סביר להניח שישנם שחקנים טובים ושחקנים גרועים, וששחקן טוב ישחק בעיקר בסרטים טובים".
התרשים הבא הוא היסטוגרמה[5] של הפרשי הציונים בין הסרט הטוב ביותר לגרוע ביותר של כל שחקן (בעבור אלו עם יותר מסרט אחד). כלומר - הפרש 0 אומר שכל הסרטים שלו קיבלו את אותו הציון, והפרש 10 אומר שהיה לו סרט מושלם וסרט נורא ואיום.

[4] corrolation_between_raiting_and_oscars
[5] cast_member_rating_variance

הממוצע הוא 1.111, סטיית התקן היא 0.78, ואחוזון 95 עומד על 2.597. כלומר, המידע אכן מקיים במידת מה את ההנחה שלנו - ההפרשים בין הסרטים של שחקן מסוים אינם עולים על 30%.



3 קליסטור המידע

1.3 בניית גרף

1.1.3 יצירת הקשתות

המטרה שלנו אם כן היא לבנות תחילה את גרף הקשרים בין הסרטים, או במילים אחרות - להחליט מה הן הקשתות המחברות סרטים. לצורך כך נבדקו מספר גישות.

תהי $\{M_i\}_{i=1}^N$ קבוצת הסרטים השונים, ויהי $C_i = \{c_i^1, ..., c_i^k\}$ קבוצת השחקנים המשחקים בסרט i. יהי S_i הדירוג של הסרט i.

הגישה הפשוטה ביותר[6] אומרת את הדבר הבא: הציון לקשת בין הסרט M_i לסרט M_j יהיה ההפרש בין הדירוגים של הסרטים (במידה ויש להם שחקנים משותפים). כיוון שאנחנו רוצים שציון גדול יותר יעיד על קשר חזק יותר, וההפרש בין הציונים חסום (10 הוא הציון המקסימלי), נכתוב כך:

$$w_{i,j} = \begin{cases} 10 - |S_i - S_j| & C_i \cap C_j \neq \phi \\ 0 & \text{otherwise} \end{cases}$$

גישה נוספת [7] מגדירה את הציון בין הסרט M_i לסרט M_j ע"י:

$$w_{i,j} = |C_i \cap C_j|$$

הגישה הזו נתנה תוצאות טובות, אבל גישה בריאה יותר היא לתת ציון שונה לכל אחד מהשחקנים בסרט. הסיבה היא שמניתוח ה dataset עולה באופן מאוד ברור שרשימת השחקנים בסרט ממויינת, כלומר ככל שהשחקן מופיע מוקדם יותר ברשימה - כך חשיבותו בסרט עולה.

בנוסף, צריך להוסיף את במאי הסרט לרשימה (היות והוא לא חלק מצוות השחקנים). התוספת הזו מתבצעת ב data_loader.py.

לכן, הגרסה המתקדמת יותר[8] של המשקל הזה תהיה זו: כעת נניח שיש חשיבות לסדר בקבוצה C_i. נגדיר את המשקל החדש כך: תהי הפונקציה

$$C(i,c) = \begin{cases} j & \exists j : c_i^j = c \\ 0 & \text{otherwise} \end{cases}$$

כלומר $C(i,c)$ מחזירה את האינדקס של השחקן c בקבוצה C_i. אזי המשקל החדש יהיה

$$w_{i,j} = \sum_{c \in C_i \cap C_j} 1 + 2^{-C(i,c)} + 2^{-C(j,c)}$$

[6] python_code/learning/graph/edges_logics/rating_diff.py
[7] python_code/learning/graph/edges_logics/cast_count.py
[8] python_code/learning/graph/edges_logics/cast_priority_factor.py

כלומר בעבור שני סרטים $C_1 = \{a,b,c\}$ $C_2 = \{a,c\}$ אזי $w_{1,2} = \underbrace{1+1+1}_{a} + \underbrace{1+\frac{1}{4}+\frac{1}{2}}_{c} = 4.75$.

היוריסטיקה נוספת שנרצה להוסיף היא התייחסות לסוג הסרט[9]. סביר להניח שזה ששחקן x שיחק בשתי קומדיות יצביע על קשר גבוה יותר בין טיב הסרטים לעומת קומדיה מול דרמה. על כן אם $G_i = \{g_1, ..., g_t\}$ קבוצת הז'אנרים של הסרט i, נגדיר את המשקל החדש כך:

$$w'_{i,j} = w_{i,j} \cdot (|G_i \cap G_j| + 1)$$

2.1.3 מדידת איכות המשקלים

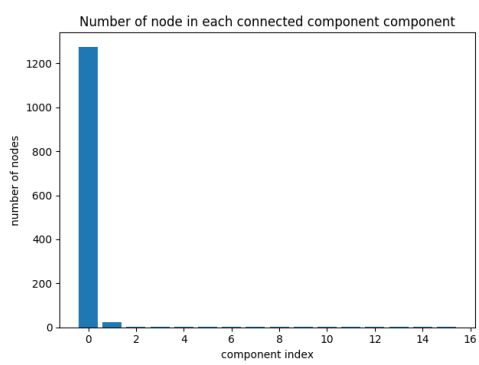
הדרך הטובה ביותר שיש לי כרגע כדי למדוד את איכות המשקלים של הקשתות ביחס לבעיה אותה אני מנסה לפתור, היא לחשב בעבור כל לוגיקה את מקדם המתאם של פירסון ביחס להפרש בין הציונים של הסרטים. כלומר, היינו רוצים לראות ש $R(|S_i - S_j|, w_{i,j}) < 0$, כלומר קיים יחס ליניארי שלילי בניהם (ככל שהציון לקשת גבוה יותר - כך ההפרש בין הדירוגים קטן יותר).
נאמר ששיטה אחת טובה יותר מהשנייה אם מקדם המתאם של הראשונה גדול בערכו המוחלט מזו של השנייה (קשר מובהק יותר). צריך לזכור שזו היוריסטיקה בלבד (אם היה ברור שזה יתן תוצאות טובות יותר - היינו נשארים עם הדרך הראשונה, שבה ברור שהמקדם הוא -1)
הטבלה המצורפת מציגה את תוצאות הניסוי, הקוד שיוצר אותה נמצא בpython_code/learning/stats/edge_logic_test.py.

לוגיקה	R
הפרש בין הדירוגים של סרטים עם שחקנים משותפים	-1
התבססות על כמות שחקנים משותפים	-0.039672
התבססות על כמות שחקנים משותפים לפי סדרם ברשימה	-0.042627
התבססות על כמות קטגוריות משותפות	-0.014543
שילוב לוגיקה 2 ולוגיקה 4	-0.06095
שילוב לוגיקה 3 ולוגיקה 4	-0.058427

אפשר לשים לב שהקשר ממש לא מובהק באף אחת מהלוגיקות, ואף הקשר של "שילוב לוגיקות 2 ו 4" נראה על פניו טוב יותר מהקשר של "שילוב לוגיקות 3 ו 4", אבל במבחן התוצאה דווקא הלוגיקה אחרונה הייתה המוצלחת ביותר מבניהן, ולכן זו הלוגיקה שאיתה נמשיך בפרויקט.
מכאן והלאה נבחן את החלוקה שלנו בראי שתי שיטות ליצירת הקשתות: שיטת ההפרש בין הציונים, ושיטת "שילוב לוגיקות 3 ו 4" (שלה נקרא מעתה "שיטת רשימת השחקנים").

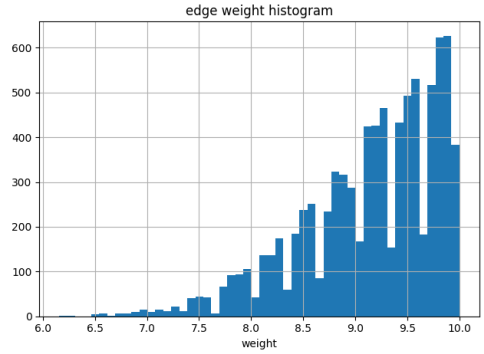
3.1.3 סטטיסטיקות על הגרף

התרשים הבא מראה את כמות הסרטים הנמצאים בכל אחד מרכיבי הקשירות בגרף. זה נתון חשוב מאוד, כי אם יש לנו הרבה מאוד רכיבי קשירות קטנים - אין לנו בעצם במה למצוא את האשכולות שלנו.

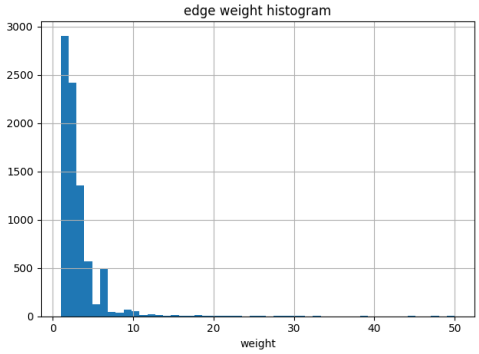


התרשימים הבאים מציגים היסטוגרמה של משקלי הקשתות בגרף. משמאל - הלוגיקה מבוססת הפרש דירוגי הסרטים, מימין - הלוגיקה המבוססת על שיטת רשימת השחקנים.

[9] python_code/learning/graph/edges_logics/category.py



איור 1: לוגיקת הפרש דירוגים



איור 2: לוגיקת רשימת השחקנים

2.3 חלוקת הגרף לאשכולות

3.3 מדידת איכות התוצר

4 סיכום