מסחר אלקטרוני – 096211

תרגיל בית 1 – השפעה ברשתות

מגישות:

סתיו ספקטור - 315818666

סתיו בר חיים – 316377449

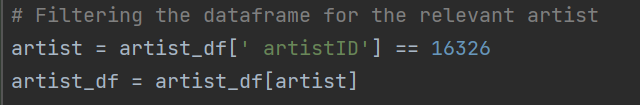
אופן בחירת המשפיענים

רקע

כהכנה ולצורך נוחות, הגדרנו לכל צומת מספר תכונות רלוונטיות:

- מספר שמיעות עבור כל אמן לפי הנתון ב- spotifly.cvs

- האם רכש את המוצר של כל אמן בזמן כלשהו (בוליאני)

את הבדיקה של חמשת המשפיענים עבור כל אמן ביצענו באופן ידני ע"י שינוי תעודת הזהות של האמן עליו יתבצעו כל הפעולות.   
למשל עבור האמן 16326:

הסבר

בשלב הראשון, רצינו לצמצם את הצמתים בגרף לרשימת הצמתים שמועמדים להיות משפיענים ברשת.

לשם כך, ראשית, מצאנו את רכיב הקשירות הגדול ביותר בגרף. גילינו שהוא מכיל את הרוב המוחלט של הצמתים (1843 צמתים).

שנית, בתוך רכיב קשירות זה, מצאנו מהו מספר השכנים המקסימלי עבור צומת כלשהו. בהתאם לכך, נתנו לכל צומת ציון (כתכונה- “grade\_neighbors”) שנקבע על פי מספר השכנים של הצומת חלקי מספר השכנים המקסימלי שמצאנו, זאת כדי לנרמל את הציון הנ"ל למספר בין 0 ל-1. הגדרנו ציון זה תוך מחשבה על כך שצומת בעל מספר שכנים רב כנראה יגדיל את מספר שכניו בהמשך לפי עיקרון ה- “rich getting richer”.

בנוסף, חישבנו עבור כל צומת את סכום השמיעות של כל שכניו, עבור כל האמן הרלוונטי, והכנסנו סכום זה כתכונה לצומת (“sum\_listened”).

לבסוף, הענקנו לכל צומת תכונת ציון סופי (“Grade”) שחושבה על ידי מכפלת ציון ה-“grade\_neighbors” ב- “sum\_listened”, תוך מחשבה שאלו הקריטריונים הרלוונטיים ביותר לבחירת משפיענים ברשת.

לאחר מכן, הורדנו את כל הצמתים שציונם היה 0, וחישבנו את ממוצע הציונים הסופיים של שאר הצמתים ברכיב הקשירות. צמצמנו את מספר הצמתים ברכיב אף יותר על ידי הכנסה לרשימה רלוונטית רק את הצמתים בעלי ציון סופי הגדול מהממוצע הנ"ל (משתנה עבור כל אמן).  
כעת, יש בידינו את רשימת הצמתים המצומצמת (“shrink\_graph\_nodes”), אם היא מכילה יותר מ-50 צמתים נצמצם אותה אף יותר לצמתים בעלי 50 הציונים המקסימליים.  
לבסוף, נבחר חמישה משפיענים מתוך רשימה זו ע"י השלב השני.

בשלב השני, השתמשנו ברעיון של האלגוריתם החמדן “hill climb” בחיפוש אחר חמשת המשפיענים הטובים ביותר.   
הגדרנו פונקציה (“hill\_climb”) אשר מקבלת *את רשימת הצמתים המצומצמת, ולבסוף מחזירה את רשימת חמשת המשפיענים הטובים ביותר.*בפונקציה מאותחלת רשימה ריקה בה בהמשך יאוחסנו המשפיענים.  
מתבצע מעבר על רשימת הצמתים המצומצמת, שלא כוללת את המשפיענים עצמם, ועבור כל צומת נבדקת, באמצעות קריאה לפונקציה “compute\_IC”, ההתפשטות הפוטנציאלית להדבקה בזכותו בגרף לאחר 6 זמנים.  
לבסוף, מאוחסן ברשימת המשפיענים הצומת בעל ההתפשטות הגדולה ביותר.  
התהליך חוזר עבור 4 משפיענים נוספים, כך שבכל פעם הוספנו את הטוב ביותר, ובכל קריאה לפונקציה “compute\_IC” נוספים הצמתים שכבר נבחרו להיות משפיענים באיטרציה הקודמת.

הפונקציה “compute\_IC” מתבססת על חישוב Influence cone, מקבלת את רשימת הצמתים (אשר מוגדרים ככאלה שקיבלו את המוצר בזמן 0) מהפונקציה “hill\_climb”, ומחזירה את מספר הצמתים הממוצע (מבין 100 סימולציות) שרכשו את המוצר לאחר 6 זמנים.  
בכל סימולציה מתבצעות 6 קריאות לפונקציה “purchase\_simulating\_per\_time”, תחילה עם גרף בזמן 0 עם רשימת הצמתים שקיבלו את המוצר בזמן 0, ומתעדכנת בכל קריאה לגרף בזמן הבא, עם רשימת הצמתים הכוללת שרכשו/קיבלו את המוצר.

בפונקציה “purchase\_simulating\_per\_time” מתבצע מעבר על רשימת הצמתים הכוללת שרכשו/קיבלו את המוצר עד זמן נתון (הגרף המייצג את הזמן הקודם) ונשמרת רשימת כל הצמתים ששכנים להם.  
כלומר, רשימה המכילה את כל הצמתים אשר עשויים לרכוש את המוצר בזמן הנוכחי (“may\_be\_infected”). מתבצע מעבר על כל הצמתים ברשימה זו, ונבדקת עבור כל אחד הסתברות הרכישה שלו למוצר בהתבסס על הקריטריונים הנדרשים (Bt, Nt,h). במידה ורכש, צומת זה נוסף לרשימת הצמתים שרכשו את המוצר בסך כל הזמנים עד כה (“infected”).

ביצענו ריצה חוזרת של כל הקוד עבור כל אמן ווידאנו שאכן מתקבלת רשימת משפיענים זהה בכל פעם, והם נבחרו להיות המשפיענים שלנו, כפי שמופיע בקובץ הCSV.