APML | תרגיל 3

שם: אריאל ברוך | ת"ז: 205925704

8 בדצמבר 2020

1 Theoretical Part

1.1 PCA

בPCA מלכסנים את מטריצת הempirical covariance של הדאטא שמוגדרת כך: $S = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})(x_i - \overline{x})^T$, נניח כי $\overline{x} = 0$.

1.1.1 הראו שS היא מטריצת PSD

נרצה להראות שS ריבועית ושמתקיים לכל $v \in \mathbb{R}^m$ $v^T S v \geq 0$.
תהי $X \in M_{m\times n}$ מטריצת הדאטא, אז $S = \mathbb{E}[XX^T] \in M_{m\times m}$ ועל כן ריבועית.
יהי וקטור $v \in \mathbb{R}^m$, מתקיים:

$$v^T S v = v^T \left(\frac{1}{n-1} X X^T\right) v =$$

$$\frac{1}{n-1} v^T X X^T v =$$

$$\frac{1}{n-1}(X^T v)^T X^T v = \frac{1}{n-1}(X^T v)^2 \geq 0$$

והביטוי שקיבלנו בהכרח אי שלילי (בגלל החזקה), כנדרש .

1.1.2 הראו שהדאטא יושב בתת מרחב ממימד d $V \subset \mathbb{R}^n$ אם ורק אם דרגת S היא d

נגדיר $X = \begin{pmatrix} x_1 - \overline{x} \\ \vdots \\ x_n - \overline{x} \end{pmatrix}$ כיוון שלכל מטריצה A מתקיים $rank(A) = rank(AA^T)$ (*) נוכל להסיק:

$$rank(X) = rank(XX^T) = rank(S)$$

כלומר, אם $rank(S) = d$ נקבל $rank(X) = d$ וזה מתקיים אם הדאטה יושב בתת מרחב ממימד d.

(*) הטענה נובעת מפירוק SVD של המטריצה X: $U\Sigma V^T$ ושל XX^T: $V\Sigma^2V^T$. מתקיים $rank(\Sigma) = rank(\Sigma^2)$ מעצם היותן אלכסוניות ואז:

$$rank(X) = rank(\Sigma) = rank(\Sigma^2) = rank(XX^T)$$

1.1.3 הראו כי הקורדינטות החדשות שהתקבלו הן תוצאה של איזומטריה על תת המרחב V

הקורדינטות החדשות שמתקבלות מוגדרות כך: $y_i = x_i \cdot U_d$ כאשר U אורתוגונלית ולוקחים את d העמודות הראשונות שלה. יהיו $u_1, ...u_d$ העמודות של U, והיא אורתוגונלית לכן לכל i מתקיים $\langle u_i, u_i\rangle = 1$. בנוסף, $rank(X) = d$ לכן נוכל לכתוב: $x_i^T = \sum_{j=1}^{d}\langle x_i, u_j\rangle u_j$ לכל $1 \leq i \leq n$. כעת נתבונן בה"כ במרחק בין x_1 לx_2:

$$\| x_1^T - x_2^T \|_2^2 = \langle x_1^T - x_2^T, x_1^T - x_2^T\rangle = \langle\sum_{j=1}^{d}\langle x_1, u_j\rangle u_j, \sum_{j=1}^{d}\langle x_2, u_j\rangle u_j\rangle = \sum_{j=1}^{d}\langle x_1 - x_2, u_j\rangle^2$$

כעת יהיו y_1, y_2:

$$\| y_1^T - y_2^T \|_2^2 = \| U_d^T \sum_{j=1}^{d} x_1^T - U_d^T \sum_{j=1}^{d} x_2^T \|_2^2 = \| U_d^T \left(\sum_{j=1}^{d} x_1^T - x_2^T\right) \|_2^2 =$$

$$\| \sum_{i=1}^{d}\sum_{j=1}^{d} U_d^T \langle x_1 - x_2, u_j\rangle u_j \|_2^2 =$$

מאורתוגונליות U נקבל:

$$\| \sum_{i=1}^{d}\langle x_1 - x_2, u_j\rangle \|_2^2 =$$

$$\langle\sum_{i=1}^{d}\langle x_1 - x_2, u_j\rangle, \sum_{i=1}^{d}\langle x_1 - x_2, u_j\rangle\rangle =$$

$$\sum_{i=1}^{d}\langle x_1 - x_2, u_j\rangle^2$$

כנדרש.

1.2 LLE

ראינו בהרצאה כי המטרה של שלב 2 של LLE היא לתאר כל נקודה x_i כקומבינציה אפינית של k השכנים הקרובים ביותר:

$$W_i = argmin_w \| x_i - \sum_{j\in N(i)} w_j x_j \|^2 = argmin_w \| \sum_{j\in N(i)} w_j z_j \|^2$$

כאשר $z_j = x_i - x_j$. תהי G מטריצת גרם של וקטורי z, $G_{a,b} = z_a^T z_b$

1.2.1 הראו כי $\| \sum_{j\in N(i)} w_j z_j\|^2 = w^T G w$

$$\| \sum_{j\in N(i)} w_j z_j \|^2 = \langle \sum_{j\in N(i)} w_j z_j, \sum_{k\in N(i)} w_k z_k \rangle = \sum_{j,k\in N(i)} w_j^T {z_j}^T z_k w_k = w^T G w$$

1.2.2 נוכל למצוא w **על ידי מזעור** $w^T G w$ **תחת המגבלה** $\sum_i w_i = 1$**. כתבו את הלגראנג'יאן וקבלו את הנוסחה:** $w = \frac{\lambda}{2} G^{-1}\mathbf{1}$

נשים לב: $\sum_i w_i = 1 \iff w\cdot\mathbf{1} = 1 \iff 1 - w\cdot\mathbf{1} = 0$. כעת:

$$L(w,\lambda) = w^T G w + \lambda(1 - w\cdot\mathbf{1})$$

נגזור לפי w ונקבל:

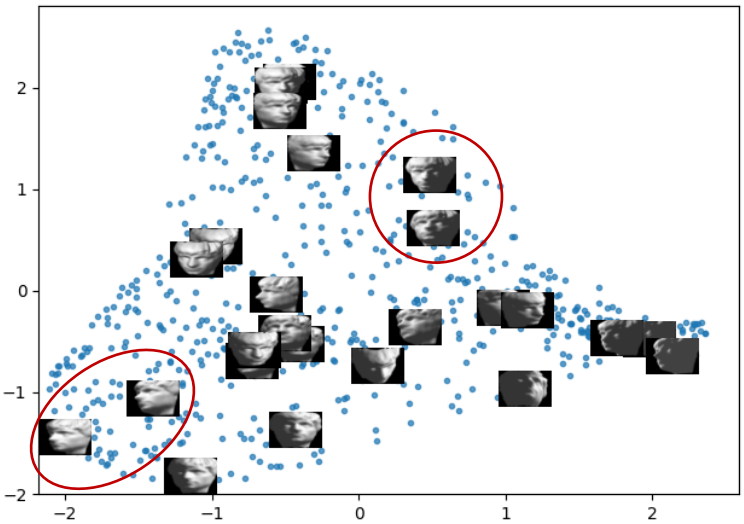
$$2Gw - \lambda\cdot\mathbf{1} = 0 \ \Rightarrow w = \frac{\lambda}{2} G^{-1}\mathbf{1}$$

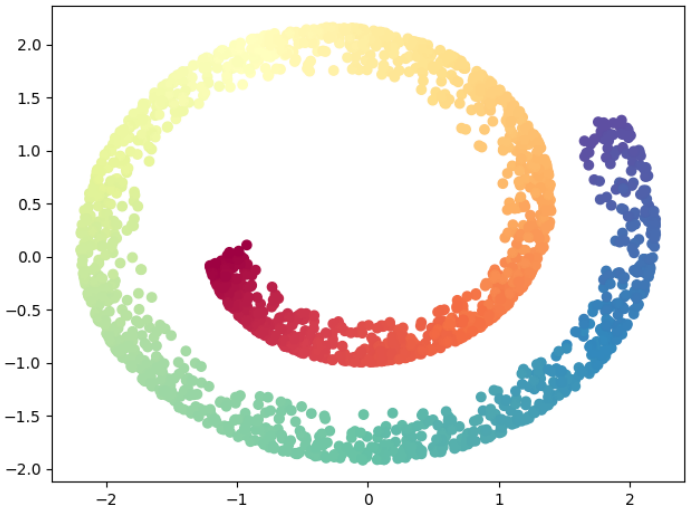
כנדרש.

Practical Part

1. מימוש האלגוריתמים:

a. MDS – האלגוריתם מקבל מטריצה בגודל N × N ומחזיר מטריצה בגודל N × d , המימד של תת המרחב עליו (בתקווה) יושב הדאטא. ביצועי האלגוריתם נבדקו על הדאטאסטים של swiss role והפרצופים. להלן התוצאות: בדאטא של הפרצופים האלגוריתם לא הצליח במידה רבה, אבל כן יש קלאסטרים שבהם הפרצופים פונים לאותו כיוון כפי שמוקף בעיגול. לעומת זאת, בדאטאסט של swissrole האלגוריתם לא מצליח לפתוח את הצורה באופן שפורש אותה בדו מימד, כיוון שבאלגוריתם נעשה שימוש במרחקים אוקלידיים, וכך נשמרים מרחקים קרובים אך גם רחוקים. .





b. קביעת הערך של d באמצעות scree plot – הערך של d , כלומר המימד של תת המרחב שעליו יושב הדאטא, קובעים באמצעות ויזואליזצית scree plot. מסדרים את הערכים העצמיים של המטריצה המתקבלת מהאלגוריתם בסדר יורד, ורואים מתי יש פער גדול ("מרפק") בין הערכים. סופרים כמה ערכים עצמיים יש עד הפער, וזה הd המבוקש. בתרגיל ביקשנו לבדוק כיצד רעש משפיע על הscree plot ובכך מטשטש את הסיגנל של הדאטא. הדבר נעשה כך:

i. יצירה של דאטה בגודל N × p שיושב למעשה בתת מרחב בגודל N × d

ii. סיבוב של הדאטה ע"י שימוש במטריצת סיבוב אורתוגונלית Q שמתקבלת מפירוק QR של מטריצה גאוסיאנית.

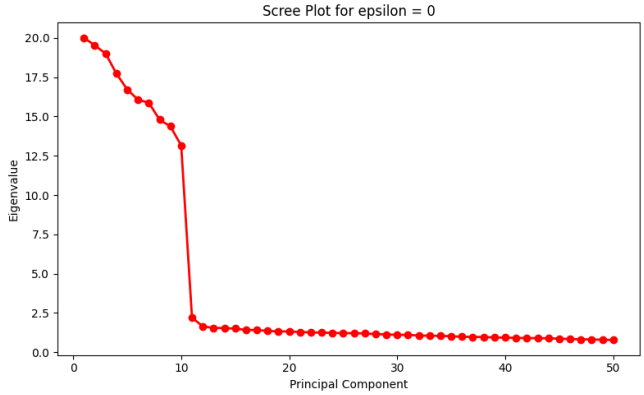
iii. יצירת מטריצת רעש Z עם תוחלת 0 ושונות 1 בגודל N × p.

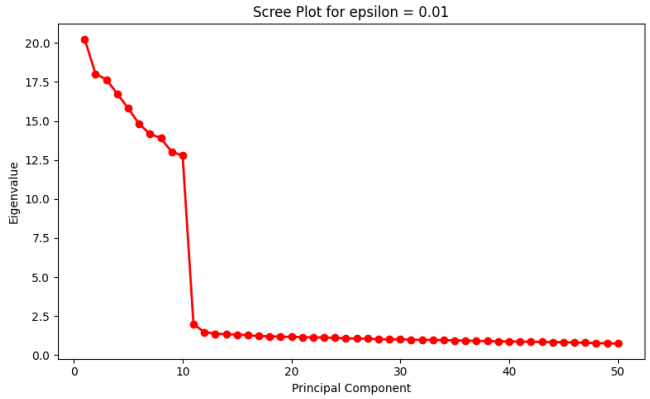
iv. הרצת אלגוריתם MDS על $X \cdot Q + \varepsilon Z$ כאשר אפסילון קובע כמה רעש נוסיף.

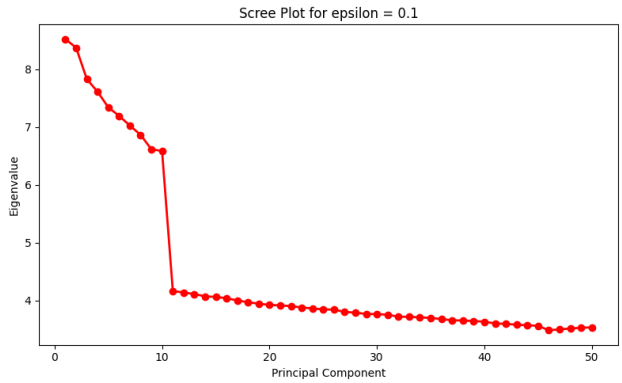
הרצתי על ערכי אפסילון של: [0, 0.01, 0.1, 0.15 1]

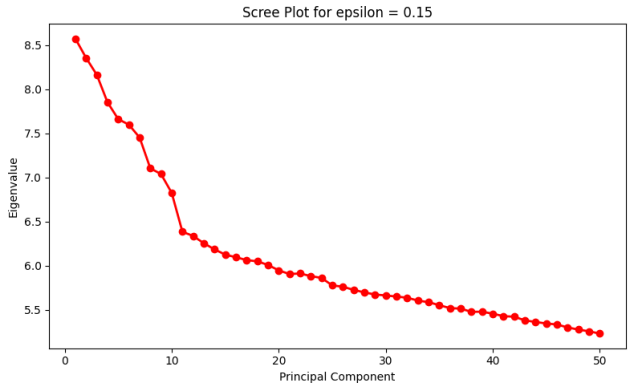
להלן התוצאות עבור n=500, p=1000, d=10:

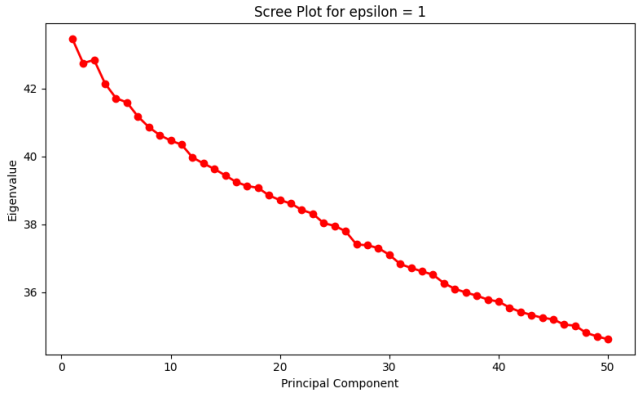
כאשר אין רעש, וכאשר הרעש קטן ניתן לזהות בבירור את d כי הקפיצה אכן מתרחשת לאחר 10 נקודות, כלומר 10 ערכים עצמיים שגדולים משמעותית משאר הערכים העצמיים. עבור e=0.1 הפער מצטמצם במעט, ועבור e=0.15 הפער מצטמצם כמעט לחלוטין עד שבערכים גדולים מכך, ובפרט עבור e=1 אין פער ואי אפשר לזהות את d.





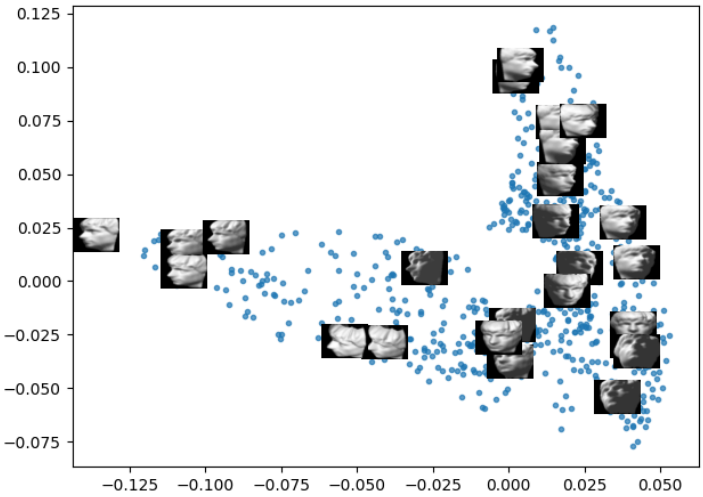


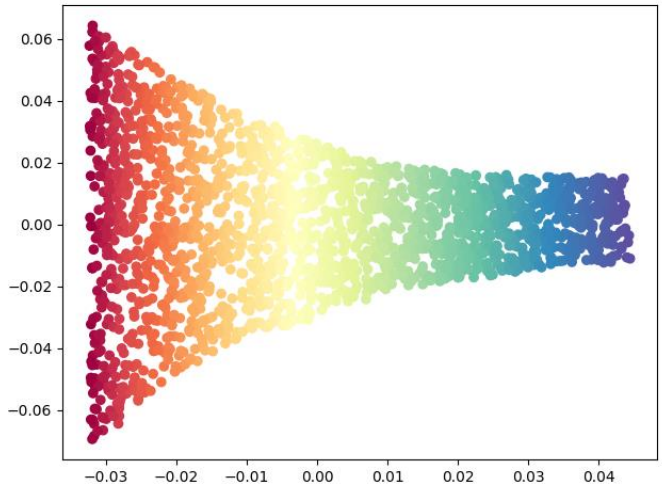




c. **LLE –** בדקתי את ביצועי האלגוריתם LLE על שני הדאטאסטים השונים. השתמשתי במימוש של האלגוריתם מ sklearn.manifold ועם הפרמטרים d=2, k=20. ניתן להבחין כי ביצועי האלגוריתם על הדאטא של הפרצופים טובים מאשר MDS, האלגוריתם מפריד את הפרצופים לקלאסטרים ברורים לפי הזווית היחסית של הפנים. גם בדאטאסט של swissrole האלגוריתם מצליח לפרוש בצורה טובה את

האובייקט התלת מימדי ולייצג אותו בדו מימד, כיוון שהוא משמר רק מרחקים קטנים בניגוד לMDS.

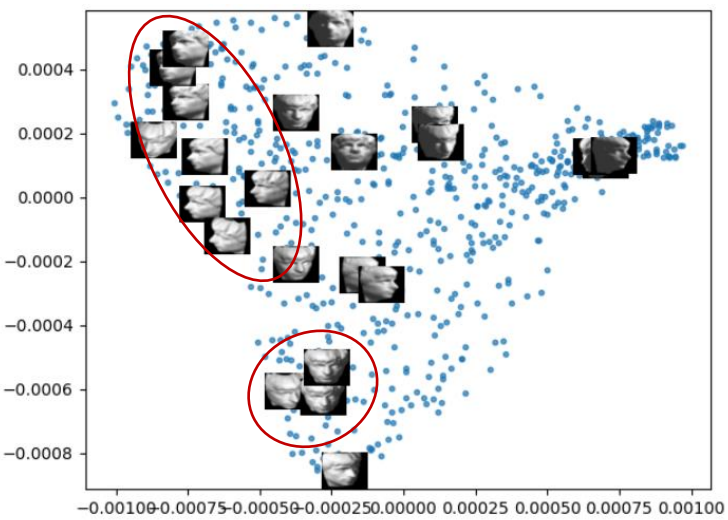


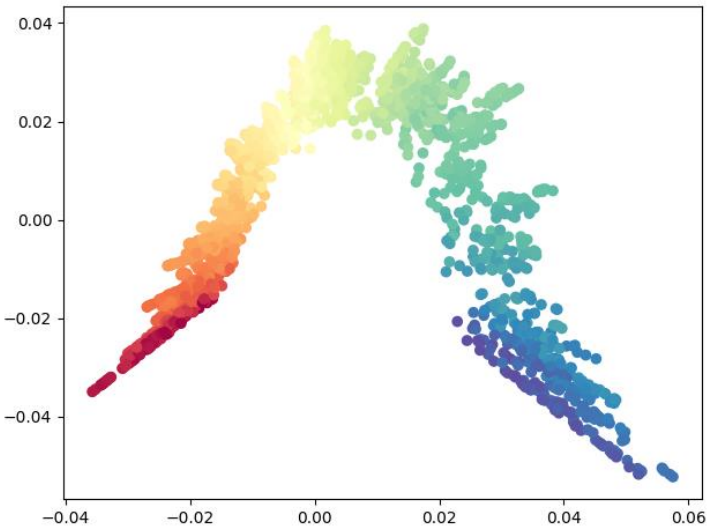


d. Diffusion map – את אלגוריתם diffusion map מימשתי בהתאם למה שהוסבר בהרצאה. הפרמטרים של האלגוריתם הם:

- d - המימד הסופי של הדאטא
- sigma -הערך שמשתמשים בו לחישוב מטריצת האפיניות בשיטת heat kernel
- t - קובע את רמת הdiffusion

האלגוריתם הורץ עם הערכים t=2, sigma=1 עבור swissrole וt=2,sigma=1000 עבור הפרצופים. ניתן לראות כי בפרצופים האלגוריתם מצליח להפריד בצורה סבירה, נראה שיש ריכוזים של פרצופים שפונים לאותו כיוון (מסומן בגרף). בswissrole הביצועים אמנם מעט יותר טובים מאשר MDS, אבל עדיין אין פריסה של המבנה התלת מימדי באופן שLLE עושה. הסיבה כנראה היא שעדיין מסתמכים על מרחקים אוקלידיים בחישוב הheat_kernel לכן לא נעדיף לשמור על מרחקים קצרים בלבד.





2. Netflix Prize Dataset

a. **Pre-processing –** הדאטאסט מכיל 17,770 סרטים ו 2,649,430 יוזרים, כאשר כל לכל יוזר דירוגים על חלק מהסרטים. בנוסף, יש טבלה נוספת של metadata שמפרטת עבור כל סרט את שנת היציאה שלו ולאילו ז'אנרים הוא שייך (27 ז'אנרים סה"כ). כיוון שהגודל של הטבלה הראשית כל כך עצום, יש לעשות לה סינון משמעותי לפני שניתן לעשות עליה אנליזות. מאפיין נוסף של הטבלה הוא שהיא מאוד דלילה (sparse) כיוון שכל יוזר נתן דירוג למספר מצומצם מתוך סך כל הסרטים, ויש יוזרים שלא נתנו דירוגים לשום סרט. לכן נעשה סינון של הדאטא באופן הבא:

i. סכימת מספר הדירוגים שכל יוזר נתן (או במילים אחרים, בעמודה שלו כמה תאים אינם ריקים)

ii. סינון של היוזרים כך שרק האחוזון ה5% ישאר (כלומר, 5% היוזרים שדירגו הכי הרבה סרטים)

iii. באותו אופן, סכימה של מספר הדירוגים שכל סרט קיבל, וסינון של הסרטים כך שרק האחוזון ה-5% ישאר

iv. התוצאה – גודל הטבלה לאחר הפילטור הוא 23932 × 534

b. **הורדת מימד עם manifold learning**

השתמשתי בשתי שיטות לא ליניאריות להורדת מימד: LLE וdiffusion maps. לאחר הפעלת כל אחד משני האלגוריתמים המימד של הדאטא היה 2 × 534 (הורדנו למימד זה כדי לעשות ויזואליזציה). במטרה לנסות להסיק מסקנות מהגרפים, עשיתי שתי צביעות של הנקודות לפי הmetadata– צביעה אחת לפי הז'אנר של הסרט ונוספת לפי שנת היציאה של הסרט. כמו כן, באופן רנדומלי הוספתי את שמות של 1 מכל 10 סרטים כדי לאפשר להגיע למסקנות נוספות ע"י בדיקת הסרטים שמתקבצים יחד.

לאחר בדיקה של מספר ערכים אפשריים, LLE מומש עם k=12 וdiffusion map עם k=20 alpha=0.001 וepsilon=bgh. המימוש של LLE היה באמצעות החבילה של sklearn, ושל diffusion map באמצעות החבילה של pydiffmap.

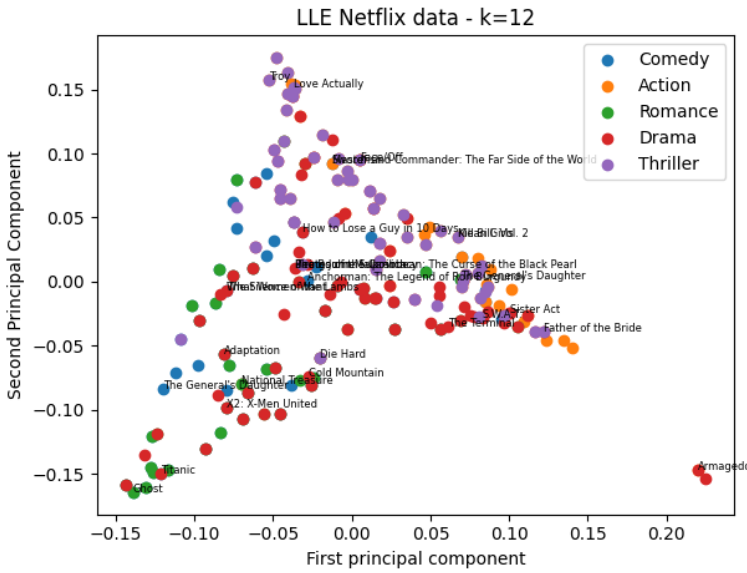
צביעה לפי הז'אנר של הסרט:

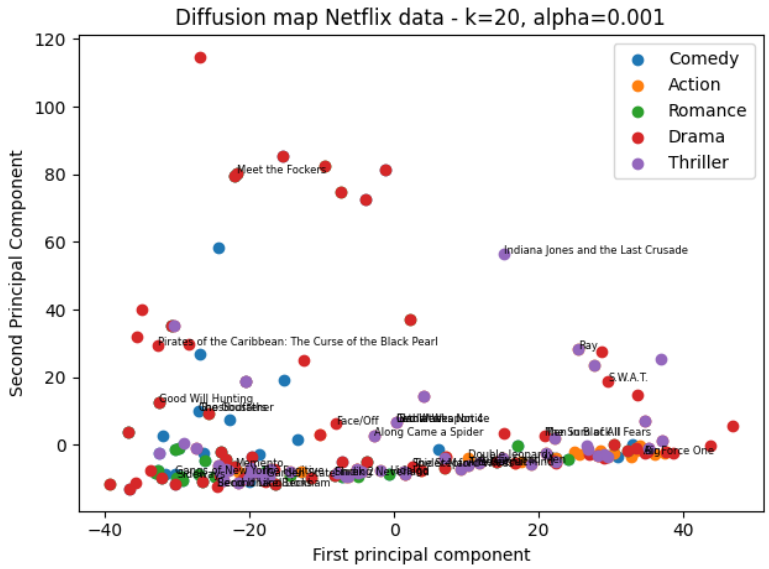
לשם הצביעה לפי ז'אנר נלקחו רק 5 הז'אנרים הכי נפוצות (Comedy, Action, Romance, Drama, Thriller) שמרבית הסרטים אכן משתייכים לפחות לאחת מהן (סרטים שלא השתייכו לאף אחת מהן לא נכללו).

בגרף שנוצר בעקבות הורדת מימד של LLE ניתן להבחין באזורים בהם הסרטים הדומיננטים הם מותחנים (thriller בסגול) ודרמה (drama באדום). כמו כן, יש מעט יותר ריכוז של סרטי רומנטיקה (romance בירוק) בקצה השמאלי התחתון וסרטי אקשן (action בכתום) בקצה הימני התחתון. שמות הסרטים לא סייעו במציאת משמעות נוספת למבנה.

בגרף שנוצר בעקבות הורדת מימד של diffusion map יש ריכוז של נקודות על ציר מרכזי, ומעט מאוד נקודות במקומות אחרים. לא הצלחתי להביא לשינוי עם ערכים שונים של הפרמטרים. הדבר הקשה על הגעה למסקנות, למעט העובדה שרוב הנקודות שלא נמצאות על הציר הן אדומות כלומר שייכות לז'אנר drama.

יש לציין כי אופן הצביעה לפי הז'אנר של הסרט היא בעייתית, כיוון שלכל סרט עשויה להיות יותר מקטגוריה אחת, והצביעה היא רק לפי אחת מהן באופן שרירותי. יתכן שזאת אחת הסיבות לכך שלא נבעו מסקנות חזקות מהצביעה.





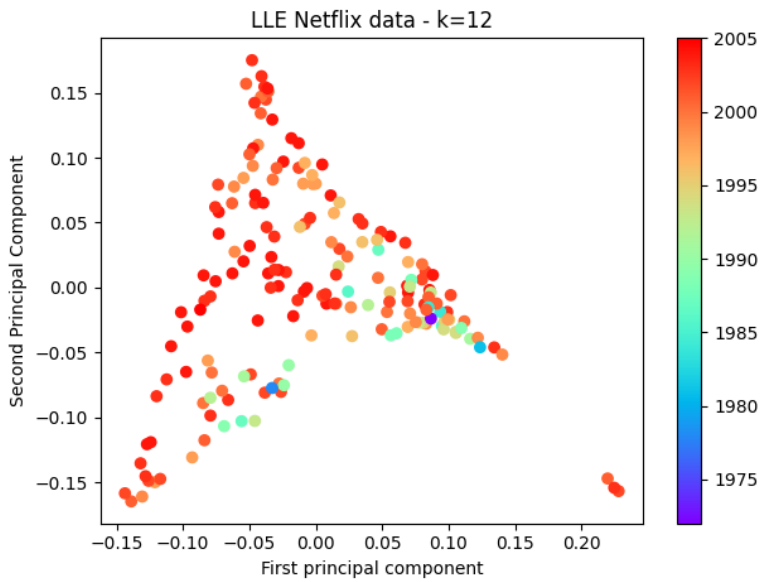
צביעה לפי שנת יציאת הסרט:

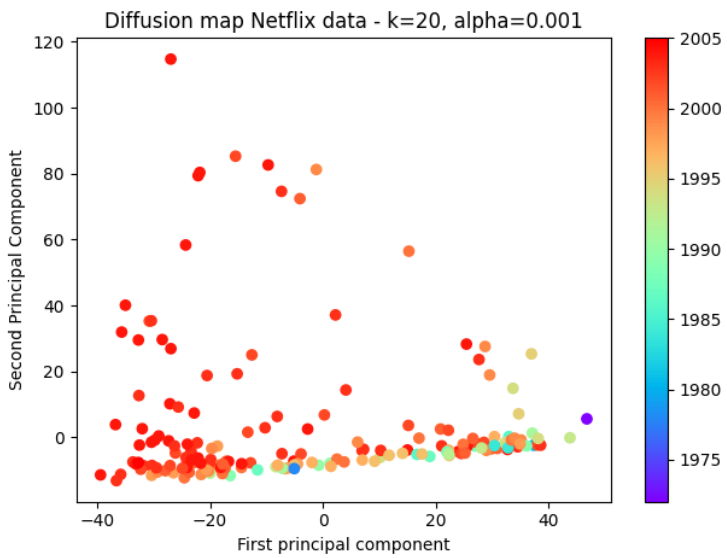
הנקודות נצבעו לפי שנת היציאה של הסרט, כאשר טווח הצבעים הוא רציף כדי שיהיה ברור אילו נקודות קרובות אחת לשניה בשנת היציאה.

בגרף של LLE קיבלנו ריכוז של מרבית הסרטים הישנים במספר מצומצם של אזורים, ואילו ה"חץ" שנוצר מורכב בעיקר מהסרטים החדשים ביותר במאגר.

בגרף של diffusion maps נראה שמרבית הסרטים הישנים נמצאים בריכוז של הנקודות, ואילו הנקודות היחידות שלא בריכוז הן ברובן סרטים חדשים.

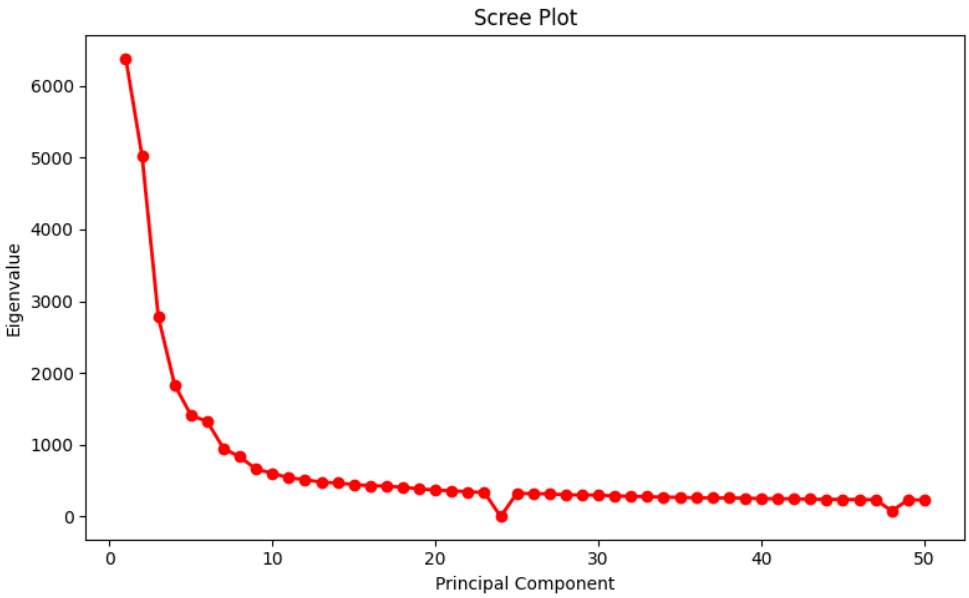
יש לשים לב שכנראה בשל אופן סינון הדאטא (לקיחת הסרטים שדורגו על ידי הכי הרבה יוזרים) ככל הנראה הרבה דגימות של סרטים ישנים ירדו בתהליך הסינון כיוון שבסבירות נמוכה יותר דורגו על ידי יוזרים רבים.





c. Spectral Clustering

כדי לראות את הקלאסטרים שהדאטא מתחלק אליהם, השתמשתי בשיטה של spectral clustering שנלמדה בתרגול. המימוש היה באמצעות החבילה של sklearn. כדי לבחור את מספר הקלאסטרים השתמשתי ב scree plot של הערכים העצמיים של מטריצת האפיניות שנבחרה להיות nearest neighbours. הערך k נבחר להיות האינדקס של הערך העצמי שממנו מתחיל ה"מרפק" של הפונקציה, כפי שניתן לראות k=4/5:

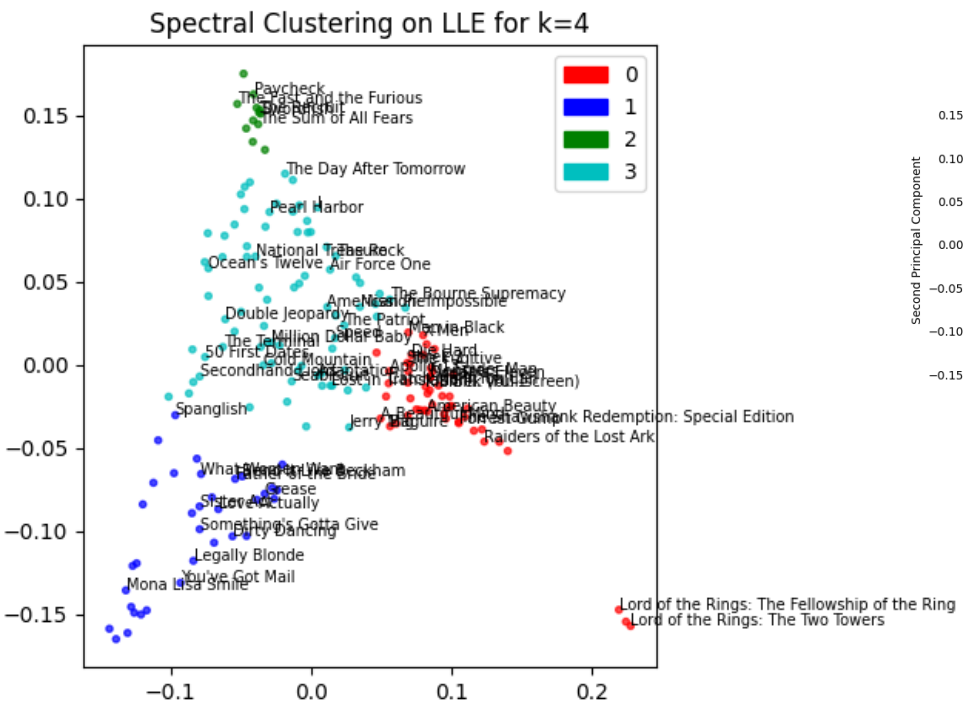


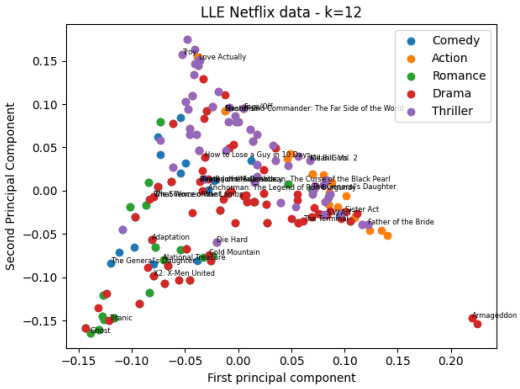
בחרתי לעשות את הקליסטור על הגרף שהתקבל מLLE שכן הוא היה יותר טוב מהגרף שהתקבל מdiffusion maps (שלא הצליח לפרוש את הנקודות אלא צופף אותן בציר אחד). בגרף שהתקבל ניתן להבחין בתאימות יחסית בין הקלאסטרים שנבחרו לצביעה לפי ז'אנרים, באופן הבא: קלאסטר 0 תואם לסרטי הפעולה, קלאסטר 1 תואם לסרטי הרומנטיקה קלאסטר 2 תואם למותחנים וקלאסטר 3 תואם לסרטי הדרמה.

הגרף של הצביעה לפי ז'אנרים צורף בשנית למטרות השוואה.

בנוסף, נשים לב כי יש כמה נקודות מבודדות בצד הימני התחתון של הגרף, כאשר לפי הגרף שתיים מהן הם מסדרת הסרטים "שר הטבעות" ובבדיקה שנעשתה בנפרד גם הסרט השלישי נמצא שם. התוצאה מעיד במידה מסוימת על איכות התוצאה של הורדת המימד עם LLE שאכן הצליח לשמר את הקירבה בין הסרטים הללו על סמך דירוג היוזרים בלבד, וכנראה גם שהדבר נובע מכך שיוזרים זהים נתנו לסרטים דירוג דומה.

(פה הדפסתי 30% משמות הסרטים, כדי שיהיה אפשר להתרשם יותר בקלות מההתאמה לז'אנר וכן הקרבה לסרטים אחרים בקלאסטר).





d. לסיכום:

ניתן לראות שרק על סמך דירוגי היוזרים אפשר להגיע להורדת מימד שמשמרת סרטים מז'אנרים דומים, כלומר נובעת המסקנה המתבקשת שאנשים נוטים לאהוב את אותם סגנונות סרטים.

עם השיטה LLE הצלחתי להגיע לתוצאות יחסית סבירות, בעיקר עם ההשוואה אל מול הקלאסטרים שהתקבלו, אך לא עם diffusion maps. יתכן כי הדבר נובע מבחירת פרמטרים לא טובה (למרות שניסיתי הרבה ואריאציות) שהובילה לביצועים פחותים של האלגוריתם.