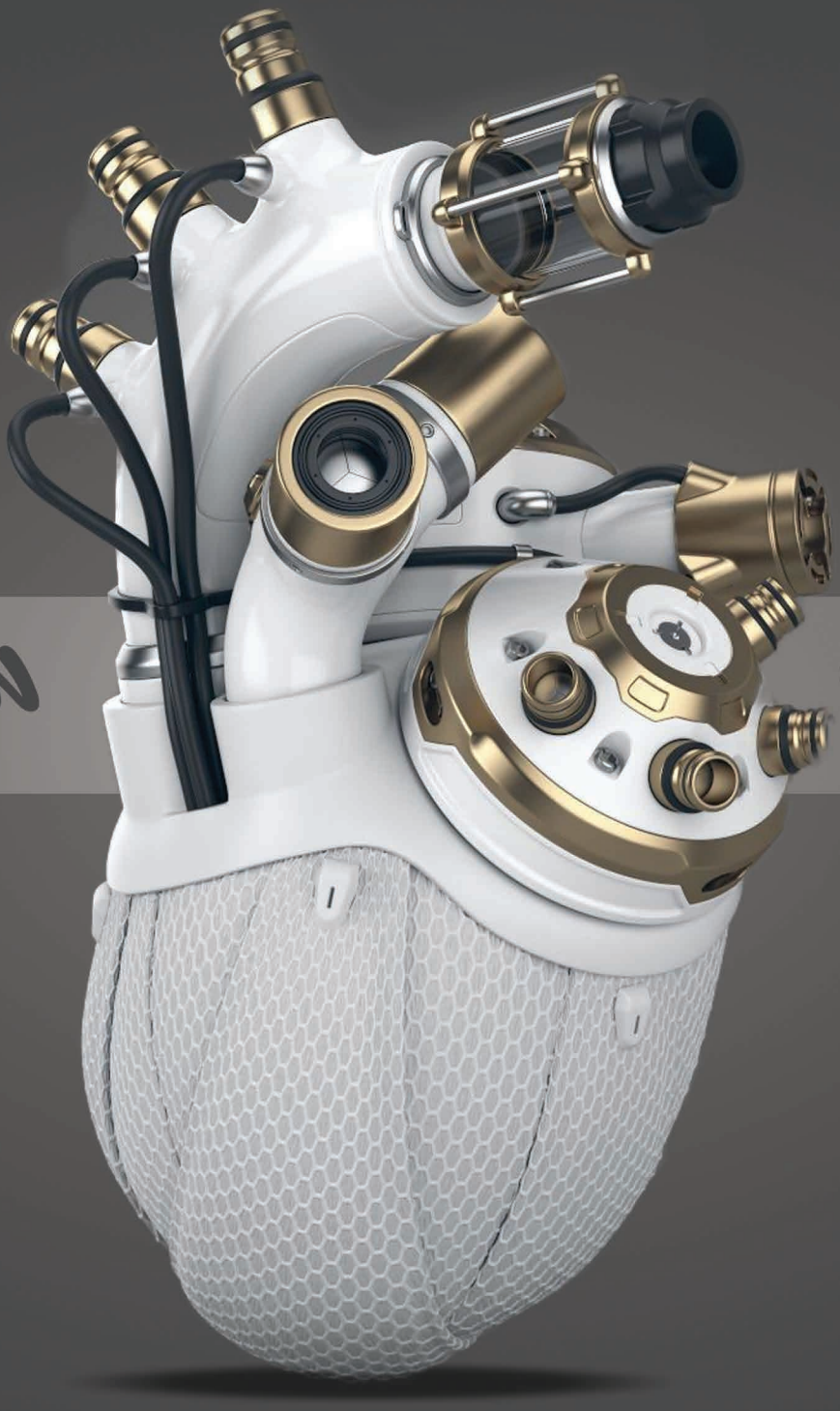


מא"ן 22

מתן כהן

חלק I

חוקי הקשר - Association Rules

א נבחר שני אלגוריתמים של חוקי הקשר, נתאר וננתח את האלגוריתמים תוך כדי נימוק בחירתינו.

בחרתי בשני אגלוריתמים שנלמדו במסגרת הקורס: אלגוריתם א־פריורי (A-priori) ואלגוריתם FP-Growth

אלגוריתם א־פריורי (A-priori) -

העיקרון ה א־פריורי: אם קבוצת פריטים היא שכיחה, אז כל תת־קבוצה של אותה קבוצה חייבת להיות שכיחה גם היא.

כמו־כן, תמיכה (Support) של קבוצה לעולם לא תהיה גבוהה מהתמיכה של תת הקבוצות שלה

האלגוריתם מוצא קבוצות תדירות באורכים הולכים וגדלים באופן איטרטיבי ע"י בניית קבוצות מועמדות ובדיקתן אל מול תנאי הסף של תמיכה מינימלית (min_support). פעולת הלאגוריתם היא איטרטיבית ומתבצעת באמצעות BFS.

בסיום ריצת האלגוריתם ומציאת קבוצות השכיחות, האלגוריתם מפיק את חוקי ההקשר החזקים שמצא, דהיינו, אלו שעמדו במבחן החסם התחתון של תמיכה מינימלית.

אלגוריתם 1 אלגוריתם א־פריורי

1. אתחל את k ל - 1
2. **כל עוד** אפשרי לבנות קבוצות חדשות:
3. בנה קבוצות C_k בגודל k של מועמדים בעזרת איחוד קבוצות מהאיטרציה הקודמת בגודל $k-1$
3. מחק על פי עיקרון הגיזום ה א־פריורי קבוצות מועמדות C_k שיש להן תת־קבוצה שאינה שכיחה
4. לכל קבוצה C_k חשב את התמיכה של כל הקבוצות המועמדות שנותרו
5. **אם** התמיכה קטנה מ $Support$:
6. **אז** נמחק את הקבוצה
7. **אחרת**:
8. היא מוגדרת כקבוצה L_k שעברה את החסם התחתון.
9. הגדל את k ב - 1
10. גזור את חוקי ההקשר על ידי בדיקת ה $Confidence$
11. **החזר** את איחוד הקבוצות L_k

אציין שהמועמדים בגודל k בכל איטרציה מאוחסנים במבנה נתונים של עץ גיבוב.

כיוון שיש מספר סופי של פריטים בבסיס הידע האלגוריתם יעצר.

חסרונות

1. האלגוריתם לא יעיל מבחינת מקום וזמן ריצה (נאמר גם בשיעור שמדובר בבעיית NP-Complete), גודל קבוצות המועמדים יכול להיות אקספוננציאלי.
2. העקרון ה א־פריורי יכול לגרום לטעויות ולפספוס של פריטים רצויים.

יתרונות

1. קל (מאוד) למימוש.
2. יותר טוב משימוש ב Brute force.

נימוק הבחירה - האלגוריתם פשוט מאוד וקל להבינו בתוספת העיקרון ה א־פריורי שמוצג לעיל.

אלגוריתם FP-Growth

בשונה מהאלגוריתם א־פריורי שהוצג, אלגוריתם זה מבצע חיפוש בעזרת DFS ומהווה שיפור לאלגוריתם א־פריורי.
אלגוריתם FP-Growth הוא אלגוריתם הפרד ומשול המוצא קבוצות שכיחות בלי לייצר מועמדים, בעזרת שימוש במבנה נתונים בשם FP-Tree של עץ השומר מידע סטטיסטי על הפריטים שבבסיס הנתונים שניתן לו.
האלגוריתם הנ"ל מייצר את העץ על ידי **שתי סריקות בלבד** של בסיס הנתונים.
אלגוריתם זה מתבסס על **הנחה** - הטרנזקציות בבסיס הנתונים ממוינות (יש לציין שגם חלק מהמימושים של אלגוריתם א־פריורי מניחים הנחה זו).
רעיון האלגוריתם הוא ייצור עץ שכיחויות, כאשר כל קודקוד בעץ מייצג פריט, שכיחותו בבסיס הנתונים וענף היוצא ממנו הכולל את כלל הטרנזקציות הקשורות אליו.
אלגוריתם זה הוא אלגוריתם יעיל במיוחד כאשר יש קבוצות שכיחות עם מספר פריטים גדול יחסית, מה שמשליך על עץ קומפקטי.
מאידך, עבור קבוצות שכיחות עם מספר מועט (יחסית) של פריטים, מתקבלים עצים גדולים שיש לבצע עליהם פעולות עיבוד כאלו ואחרות כמו פיצול.
בדומה לאלגוריתם א־פריורי, גם אלגוריתם זה מקבל כקלט את רמת התמיכה (Support) המינימלית בתור חסם תחתון.
נתאר את האלגוריתם בצורה דומה לתיאור האלגוריתם ה א־פריורי:

אלגוריתם 2 אלגוריתם FP-Growth

1. בצע סריקה על בסיס הנתונים וחשב את התמיכה (Support) של כל פריט בודד, ערוך רשימה של הפריטים עבורם התמיכה גבוהה מהתמיכה המינימלית שהתקבלה כקלט
2. בצע סריקה נוספת על בסיס הנתונים
3. **אם** פריט כבר בעץ:
4. **אז** הוסף את הטרנזקציות של הפריט או עדכן את שכיחותן אם קיימות
5. **אחרת**:
6. הוסף את הפריט כקודקוד בעץ FP-Tree
7. הוסף את הטרנזקציות תחת הקודקוד החדש בעזרת יצירת ענף וקשר כל קודקוד לקודקודים הזהים הקיימים בעץ בעזרת הצבעות
8. **החזר** את הקבוצות שנוצרו בעץ

חסרונות

1. לא יעיל במקרים של מספר מועט של פריטים

יתרונות

1. מצריך רק שתי איטרציות של מעבר על בסיס הנתונים
2. יעיל בהשוואה לאלגוריתם א־פריורי

נימוק הבחירה - קומפקטיות בזיכרון ושיפור משמעותי בזמן הריצה על פני אלגוריתם א־פריורי

ב בהנחה ש: $Min\_confidence = 60\%, Min\_support = 40\%$ נמצא את כל קבוצות התדירות תוך שימוש בשני האלגוריתמים שבחרנו בסעיף א'

לאחר הצלחת ממ"ן 21 בעזרת שימוש ב Binning השתמשתי גם כאן בשיטה זו על מנת להתמודד עם המידע.
בסעיף זה, אנחנו משתמשים בחוקי הקשר ולכן אני אוותר על הטיוב שביצעתי ואשאיר את **כלל העמודות** המקוריות מבסיס הנתונים המקורי על מנת לא לפגוע בתהליך יצירת החוקים שכן הוא בעצמו מוצא קורלציה בין עמודות. בסט הנתונים המטויב ישנן 299 רשומות ולכן: $299 \cdot 0.4 \approx 119$ ולכן, על מנת שקבוצה תוגדר כשכיחה ותעמוד בתנאי התמיכה המינימלית, כל פריט שנמצא בה צריך להופיע בסט הנתונים לפחות 119 פעמים. לפיכך יש צורך לפצל את סט הנתונים הפעם למספר שונה של bins או לפי שיטה של עומק שווה, כיוון שחלוקה לחמישה לפי תדירות שווה כפי שביצעתי בממ"ן 21 יכולה להניב פגיעה בשימוש בחוקי הקשר. מהחלוקה שהתקבלה בממ"ן 21 לחמישה bins התקבל שבכל bin יהיו $\frac{297}{5} \approx 60$ ערכים, אי לכך ובהתאם לזאת בחרתי את מספר ה־ bins לחלוקה כשניים בשיטת equal frequency על מנת שאוכל לעמוד בתנאי הסף שניתנו בהוראת הממ"ן. לאחר הפיצול נתתי לכל אחד מה־ bins שם בהתאם לעמודה אליה הוא קשור.
אציין שהשמות שניתנו לערכים אינם משליכים על ערך שלא נמצא במידת הנורמה הרפואית אלא נוצר רק על מנת לעזור לאלגוריתמים של פייתון להתמודד עם הערכים

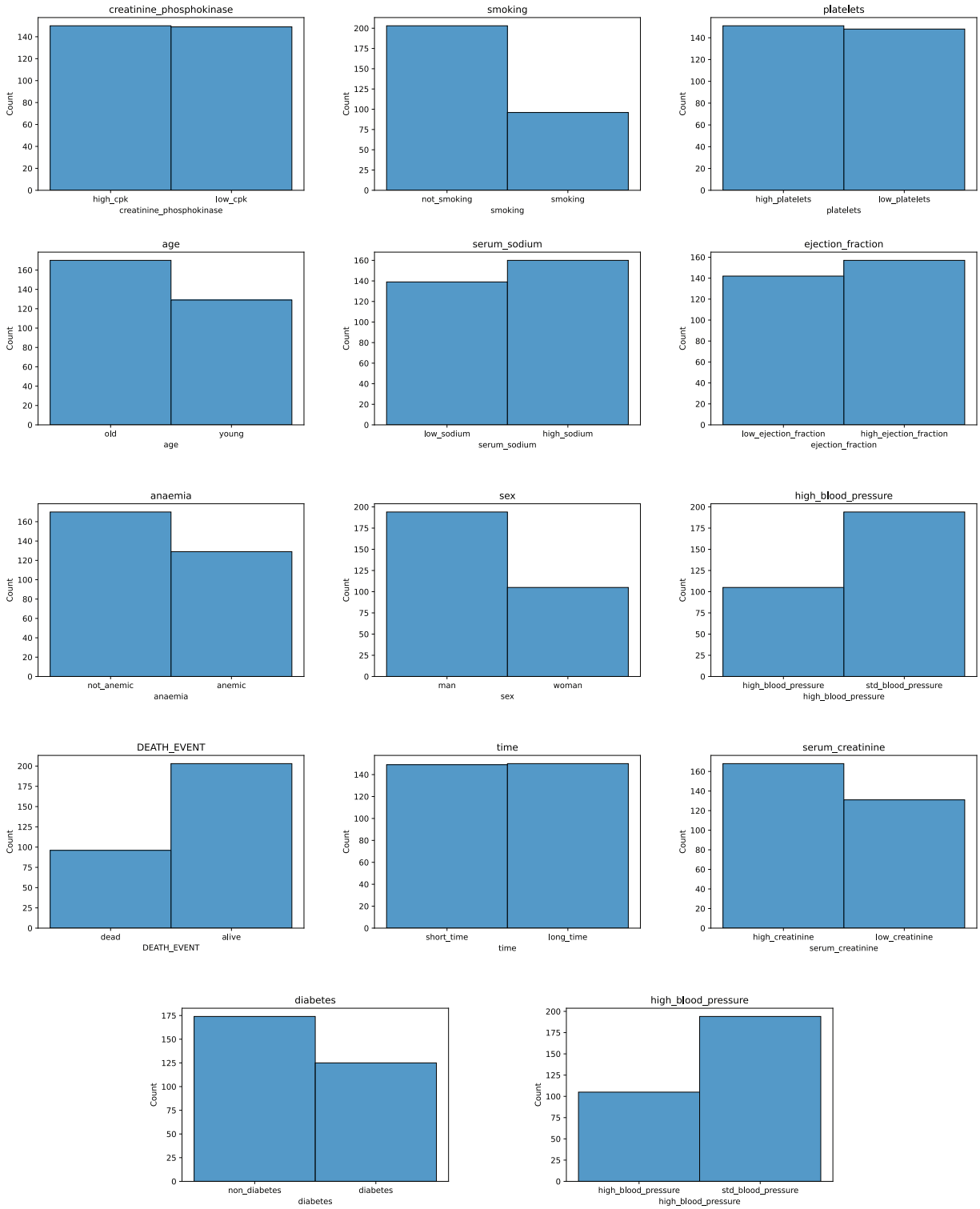
- עמודת הגיל קיבלה את הערכים: young, old
- עמודת ה־ CPK קיבלה את הערכים: low_cpk, high_cpk
- עמודת מקטע הפליטה קיבלה את הערכים: low_ejection_fraction, high_ejection_fraction
- עמודת טסיות הדם קיבלה את הערכים: low_platelets, hight_platelets
- עמודת הקריאטינין קיבלה את הערכים: low_creatinine, high_creatinine
- עמודת הנתרן קיבלה את הערכים: low_sodium, high_sodium
- עמודת זמן קיבלה את הערכים: short_time, long_time
- עמודת אנמיה קיבלה את הערכים: not_anemic, anemic
- עמודת לחץ דם גבוה קיבלה את הערכים: std_blood_pressure, high_blood_pressure
- עמודת מין קיבלה את הערכים: woman, man
- עמודת עישון קיבלה את הערכים: smoking, not_smoking
- עמודת סוכרת קיבלה את הערכים: non_diabetes, diabetes
- עמודת DEATH_EVENT קיבלה את הערכים: dead, alive

מצורפות ההיסטוגרמות בעמוד הבא.

בהתחלה בחרתי להשתמש בספריית פייתון מוכנה למימוש אלגוריתם א־פריורי ואלגורים FP-Growth שמקבלת כפרמטרים את ערכי min_confidence, min_support ובכך לחסוך לעצמי את מימוש האלגוריתמים, אך בהמשך ראיתי שהמימושים לא לטעמי, אז בחרתי ליישר קו ולהשתמש בתוכנת Weka על מנת להציג את התוצאות.

אציין שאת הכנת המידע עשיתי בעזרת פייתון כיוון שאני רגיל לפתח בסביבה הנ"ל ולכן הדברים נעשו בצורה די מהירה עבורי, לאחר pre-proccesing של המידע שברשותי ייצאתי אותו לקובץ csv ובו השתמשתי בתוכנת Weka.

איור 1: היסטוגרמות לאחר binning



קבוצות התדירות שנמצאו על ידי אלגוריתם A-priori

כפי שצוין והתבקש בהוראות הממ"ן, הגדרתי את weka כך שאקבל תוצאות עבור min_confidence, min_support רצוי והתוצאות שהתקבלו מהאלגוריתם עבור תמיכה מינימלית של 40% וביטחון מינימלי של 60% כללו:

- 22 קבוצות תדירות בנות איבר יחיד - לפיכך ישנם 22 פריטים שכיחים בבסיס הידע.
- נוצרו 9 קבוצות תדירות בעלות 2 איברים כל אחת
- לא נוצרו כלל קבוצות תדירות בעלות יותר משני איברים
- מידת הביטחון הגבוהה ביותר שנוצרה היא בעלת 86% ביטחון
- לא הייתה קבוצת תדירות עבור משתנה המטרה DEATH_EVENT=dead (וזה ברור למה, כיוון שלא עובר את מדד min_support=0.4)
- נוצרה קבוצה יחידה בעלת איבר יחיד שהוא משתנה המטרה (DEATH_EVENT)
- נוצרו 7 קבוצות בעלות שני איברים המכילות את משתנה המטרה (DEATH_EVENT)

קבוצות התדירות הכוללות את משתנה המטרה:

טבלה 1: קבוצות התדירות הכוללות את משתנה המטרה ותוכנן

#	גודל קבוצת התדירות	פריטים בקבוצת התדירות
1	1	DEATH_EVENT=alive
2	2	serum_sodium=high_sodium DEATH_EVENT=alive
3	2	time=long_time DEATH_EVENT=alive
4	2	anaemia=not_anemic DEATH_EVENT=alive
5	2	high_blood_pressure=std_blood_pressure DEATH_EVENT=alive
6	2	sex=man DEATH_EVENT=alive
7	2	smoking=not_smoking DEATH_EVENT=alive

לצערי, Weka איננה מציגה את קבוצות התדירות של אלגוריתם FP-Growth אלא רק את חוקי ההקשר שנוצרו, לכן לא הצגתי אותם.

ג נציג את חוקי ההקשר החזקים

כעת נציג את חוקי ההקשר החזקים המתקבלים מ־2 האלגוריתמים עליהם הרחבנו.
נחזור ונזכיר שחוקי ההקשר החזקים הם אלו אשר עומדים בתנאי ה־ min_support, min_confidence שקיבלנו.

ג.1 אלגוריתם א־פריורי (A-priori) - התקבלו 17 חוקי הקשר חזקים

17 חוקי ההקשר הללו כוללים 11 חוקי הקשר בהם עמודת המטרה מופיעה, 6 בהם אינה מופיעה, כלל חוקי ההקשר מורכבים משני פריטים סה"כ, נציג את חוקי ההקשר שהתקבלו הכוללים את עמודת המטרה ביחד עם מידת ה confidence ומידת ה lift שמייצגת את מידת הקורלציה:

טבלה 2: חוקי ההקשר שהתקבלו מאלגוריתם א־פריורי

association rule	confidence	lift	#
time=long_time ⇒DEATH_EVENT=alive	0.86	1.27	1
serum_sodium=high_sodium ⇒ DEATH_EVENT=alive	0.77	1.13	2
high_blood_pressure=std_blood_pressure ⇒ DEATH_EVENT=alive	0.71	1.04	3
anaemia=not_anemic ⇒ DEATH_EVENT=alive	0.71	1.04	4
sex=man ⇒ DEATH_EVENT=alive	0.68	1	5
DEATH_EVENT=alive ⇒ high_blood_pressure=std_blood_pressure	0.67	1.04	6
DEATH_EVENT=alive ⇒ smoking=not_smoking	0.67	0.99	7
smoking=not_smoking ⇒ DEATH_EVENT=alive	0.67	0.99	8
DEATH_EVENT=alive ⇒ sex=man	0.65	1	9
DEATH_EVENT=alive ⇒ time=long_time	0.64	1.27	10
DEATH_EVENT=alive ⇒ serum_sodium=high_sodium	0.61	1.13	11

מצאנו שאלגוריתם א־פריורי מצא 6 חוקי הקשר חזקים בהם עמודת המטרה (DEATH_EVENT) נמצאת בצד הנגרר (סיפא) ו־5 חוקי הקשר בהם נמצא בצד הגורר (רישא).
חוק ההקשר החזק ביותר עם מידת confidence הגבוהה ביותר הוא החוק הראשון בעל מידה של 86%.
מחוקים אלו אנו למדים על מידת הקשר בין צירופי ערכים של פריטים כאלו ואחרים הנמצאים בטרנזקציות לבין עמודת המטרה - הסיכוי לפטירה של מטופל, וליתר דיוק למאורע בו המטופל נשאר בחיים (חייב לשים לב לעניין כיוון שהסיכוי הוא המאורע המשלים - **הודגש לאחר שיחה עם שולה**).
נבחין כעת, כי מדדי ה־lift המצויים בטבלה מעלה אשר רובם מעל הערך 1 מראים על מידת קורלציה חיובית בין הפריטים המצויים באותה שורה בטבלה ואילו ערכים הקטנים מ 1 מראים על מידת קורלציה שלילית.
שורה מספר 8 בטבלה מראה על מידת קורלציה אפסית ובמילים אחרות מצביעים על חוסר תלות בין הפריטים sex ו־ DEATH_EVENT (כפי שהיינו מצפים כמובן). ערכים דומים קיימים גם עבור ערך העישון (0.99)
עוד אנחנו למדים שיש קשר חזק בין התוצאות שהתקבלו מאלגוריתם א־פריורי והתוצאות שהתקבלו בממ"ן 21 שכן גם שם בשורשי העץ שהתקבלו היה ערך time - הדבר אינו מפתיע כלל ומראה שאכן פעלנו נכון כיוון שיש חיתוך נרחב בין כלל השיטות בהן השתמשנו עד כה!
יתרה מזאת, ניווכח כי הערך השני הטוב ביותר בטבלה מעלה שנמצא ברישא הוא ערך הנתרן וגם הוא התקבל כבן לשורש העץ בממ"ן 21 באחד המימושים שעשיתי - שוב, דבר המראה על חיתוך בין השיטות.
מששת חוקי ההקשר בעלי עמודת המטרה בסיפא אנחנו למדים שערך time וערך הנתרן הם בעלי הקורלציה הגבוהה ביותר לעמודת המטרה ומצביעים על השפעה גדולה ביחס לשאר הפריטים.
עם זאת ערכי לחץ הדם, העישון והאנמיה גם הם משפיעים אך מאוד קרובים לערך חוסר תלות ואילו ערך המין לא משפיע כלל.

ג.2 אלגוריתם FP-Growth - התקבלו 6 חוקי הקשר חזקים

ששת החוקים כוללים כולם את עמודת המטרה, כלל חוקי ההקשר מורכבים מ 2 פריטים סה"כ, נציג את חוקי ההקשר שהתקבלו ביחד עם מידת ה- confidence ומידת ה- lift שמייצגת את מידעת הקורלציה:

טבלה 3: חוקי ההקשר שהתקבלו מאלגוריתם FP-Growth

association rule	confidence	lift	#
time=long_time ⇒ DEATH_EVENT=alive	0.86	1.27	1
serum_sodium=high_sodium ⇒ DEATH_EVENT=alive	0.77	1.13	2
high_blood_pressure=std_blood_pressure ⇒ DEATH_EVENT=alive	0.71	1.04	3
DEATH_EVENT=alive ⇒ high_blood_pressure=std_blood_pressure	0.67	1.04	4
DEATH_EVENT=alive ⇒ time=long_time	0.64	1.27	5
DEATH_EVENT=alive ⇒ serum_sodium=high_sodium	0.61	1.13	6

מצאנו שאלגוריתם FP-Growth מצא 6 חוקי הקשר חזקים בהם קיימת עמודת המטרה (DEATH_EVENT), כאשר 3 מהם מכילים את עמודת המטרה בצד הנגרר (סיפא) ושלושת האחרים מכילים את עמודת המטרה בצד הגורר (רישא).
חוק ההקשר החזק ביותר עם מידת confidence הגבוהה ביותר הוא החוק הראשון בעל מידה של 86%.
מחוקים אלו אנו למדים על מידת הקשר בין צירופי הערכים בדיוק כמו הפלט שהתקבל מאלגוריתם א-פריורי - מדובר באירועים בהם המטופל נשאר בחיים.
גם כאן, נבחין כי כפי שרצוי מידת ה- confidence מעל 60% בכלל חוקי ההקשר ומידות ה- lift כולן מעל 1 ולכן נמצא שבכלל חוקי ההקשר שנמצאו על ידי אלגוריתם FP-Growth הם בעלי קורלציה חיובית.
גם בתוצאות אלגוריתם זה נוכל להבחין בבירור שערך ה- time הוא הפריט עם הקורלציה החיובית הגבוהה ביותר לעמודת המטרה כפי שצוין כבר בממ"ן 21 - שוב, אנחנו לא יכולים להיות מופתעים מהדבר - קיבלנו תוצאות כמעט זהות לשל אלגוריתם א-פריורי.

ד הרצה ודיווח התוצאות של שני האלגוריתמים

לפני הרצת האלגוריתמים בתוכנת Weka עשיתי pre-processing לסט הנתונים בעזרת python כפי שכבר הסברתי **בסעיף ב,** את תהליך ה- pre-process ביצעתי בצורה שונה מהצורה של ממ"ן 21 שכן אנחנו עוסקים בממ"ן זה בלמידה שהיא unsupervised ולא רציתי לפגום בטיב האלגוריתמים עם מחיקת הנתונים שביצעתי בעזרת בדיקת מדדי קורלציה בעצמי.
על בעיית ניקוי הנתונים מממ"ן 21 עליתי בזמן הרצת האלגוריתמים בפעם הראשונה - נראה שניקוי הנתונים שלי גרם לאלגוריתמים להיות יחסית מנוונים ולא לייצר פלטים מספקים של חוקי הקשר או קבוצות תדירות בגלל דלות הנתונים, לפיכך בחרתי **לשנות את טיוב הנתונים שלי למען שאלה זו.**
תוצאות ההרצות מצויות בדף הבא.

איור 2: פלט תוכנת Weka לאלגוריתם א־פריורי

```
== Run information ==

Scheme:       weka.associations.Apriori -I -N 10000 -T 0 -C 0.6 -D 0.05 -U 1.0 -M 0.4 -S -1.0 -c -1
Relation:     heart_failure_clinical_records_dataset_binned_bool
Instances:    299
Attributes:   13
              age
              creatinine_phosphokinase
              ejection_fraction
              platelets
              serum_creatinine
              serum_sodium
              time
              anaemia
              diabetes
              high_blood_pressure
              sex
              smoking
              DEATH_EVENT
== Associator model (full training set) ==


Apriori
=======

Minimum support: 0.4 (120 instances)
Minimum metric <confidence>: 0.6
Number of cycles performed: 12

Generated sets of large itemsets:

Size of set of large itemsets L(1): 22

Large Itemsets L(1):
age=old 170
age=young 129
creatinine_phosphokinase=high_cpk 150
creatinine_phosphokinase=low_cpk 149
ejection_fraction=low_ejection_fraction 142
ejection_fraction=high_ejection_fraction 157
platelets=high_platelets 151
platelets=low_platelets 148
serum_creatinine=high_creatinine 168
serum_creatinine=low_creatinine 131
serum_sodium=low_sodium 139
serum_sodium=high_sodium 160
time=short_time 149
time=long_time 150
anaemia=not_anemic 170
anaemia=anemic 129
diabetes=non_diabetes 174
diabetes=diabetes 125
high_blood_pressure=std_blood_pressure 194
sex=man 194
smoking=not_smoking 203
DEATH_EVENT=alive 203

Size of set of large itemsets L(2): 9

Large Itemsets L(2):
serum_sodium=high_sodium DEATH_EVENT=alive 123
time=long_time DEATH_EVENT=alive 129
anaemia=not_anemic DEATH_EVENT=alive 120
diabetes=non_diabetes sex=man 124
high_blood_pressure=std_blood_pressure sex=man 133
high_blood_pressure=std_blood_pressure smoking=not_smoking 128
high_blood_pressure=std_blood_pressure DEATH_EVENT=alive 137
sex=man DEATH_EVENT=alive 132
smoking=not_smoking DEATH_EVENT=alive 137

Best rules found:

    1. time=long_time 150 ==> DEATH_EVENT=alive 129    <conf:(0.86)> lift:(1.27) lev:(0.09) [27] conv:(2.19)
    2. serum_sodium=high_sodium 160 ==> DEATH_EVENT=alive 123    <conf:(0.77)> lift:(1.13) lev:(0.05) [14] conv:(1.35)
    3. diabetes=non_diabetes 174 ==> sex=man 124    <conf:(0.71)> lift:(1.1) lev:(0.04) [11] conv:(1.2)
    4. high_blood_pressure=std_blood_pressure 194 ==> DEATH_EVENT=alive 137    <conf:(0.71)> lift:(1.04) lev:(0.02) [5] conv:(1.07)
    5. anaemia=not_anemic 170 ==> DEATH_EVENT=alive 120    <conf:(0.71)> lift:(1.04) lev:(0.02) [4] conv:(1.07)
    6. sex=man 194 ==> high_blood_pressure=std_blood_pressure 133    <conf:(0.69)> lift:(1.06) lev:(0.02) [7] conv:(1.1)
    7. high_blood_pressure=std_blood_pressure 194 ==> sex=man 133    <conf:(0.69)> lift:(1.06) lev:(0.02) [7] conv:(1.1)
    8. sex=man 194 ==> DEATH_EVENT=alive 132    <conf:(0.68)> lift:(1) lev:(0) [0] conv:(0.99)
    9. DEATH_EVENT=alive 203 ==> high_blood_pressure=std_blood_pressure 137    <conf:(0.67)> lift:(1.04) lev:(0.02) [5] conv:(1.06)
   10. DEATH_EVENT=alive 203 ==> smoking=not_smoking 137    <conf:(0.67)> lift:(0.99) lev:(-0) [0] conv:(0.97)
   11. smoking=not_smoking 203 ==> DEATH_EVENT=alive 137    <conf:(0.67)> lift:(0.99) lev:(-0) [0] conv:(0.97)
   12. high_blood_pressure=std_blood_pressure 194 ==> smoking=not_smoking 128    <conf:(0.66)> lift:(0.97) lev:(-0.01) [-3] conv:(0.93)
   13. DEATH_EVENT=alive 203 ==> sex=man 132    <conf:(0.65)> lift:(1) lev:(0) [0] conv:(0.99)
   14. sex=man 194 ==> diabetes=non_diabetes 124    <conf:(0.64)> lift:(1.1) lev:(0.04) [11] conv:(1.14)
   15. DEATH_EVENT=alive 203 ==> time=long_time 129    <conf:(0.64)> lift:(1.27) lev:(0.09) [27] conv:(1.35)
   16. smoking=not_smoking 203 ==> high_blood_pressure=std_blood_pressure 128    <conf:(0.63)> lift:(0.97) lev:(-0.01) [-3] conv:(0.94)
   17. DEATH_EVENT=alive 203 ==> serum_sodium=high_sodium 123    <conf:(0.61)> lift:(1.13) lev:(0.05) [14] conv:(1.17)


```

```
== Run information ==

Scheme:       weka.associations.FPGrowth -P 2 -I -1 -N 10000 -T 0 -C 0.6 -D 0.05 -U 1.0 -M 0.4
Relation:     heart_failure_clinical_records_dataset_binned_bool
Instances:    299
Attributes:   13
              age
              creatinine_phosphokinase
              ejection_fraction
              platelets
              serum_creatinine
              serum_sodium
              time
              anaemia
              diabetes
              high_blood_pressure
              sex
              smoking
              DEATH_EVENT
== Associator model (full training set) ==

FPGrowth found 6 rules (displaying top 6)

1. [time=long_time]: 150 ==> [DEATH_EVENT=alive]: 129   <conf:(0.86)> lift:(1.27) lev:(0.09) conv:(2.19)
2. [serum_sodium=high_sodium]: 160 ==> [DEATH_EVENT=alive]: 123   <conf:(0.77)> lift:(1.13) lev:(0.05) conv:(1.35)
3. [high_blood_pressure=std_blood_pressure]: 194 ==> [DEATH_EVENT=alive]: 137   <conf:(0.71)> lift:(1.04) lev:(0.02) conv:(1.07)
4. [DEATH_EVENT=alive]: 203 ==> [high_blood_pressure=std_blood_pressure]: 137   <conf:(0.67)> lift:(1.04) lev:(0.02) conv:(1.06)
5. [DEATH_EVENT=alive]: 203 ==> [time=long_time]: 129   <conf:(0.64)> lift:(1.27) lev:(0.09) conv:(1.35)
6. [DEATH_EVENT=alive]: 203 ==> [serum_sodium=high_sodium]: 123   <conf:(0.61)> lift:(1.13) lev:(0.05) conv:(1.17)


```

ה ננתח השוואתית את התוצאות של שני האלגוריתמים ונסיק מסקנות

כפי שראינו באיור 2 ובטבלאות 2 ו־ 3 ניתן להבחין **בשוויון** בין תוצאות ההרצה של FP-Growth לבין חלק מהתוצאות של א־פריורי, לצורך כך נגדיר את A להיות קבוצת חוקי ההקשר שהחזיר אלגוריתם א־פריורי ואת B להיות קבוצת חוקי ההקשר של אלגוריתם FP-Growth, מתקיים $A \cap B = B$ והתוצאה:

$A \cap B =$

#	lift	confidence	association rule
1	1.27	0.86	time=long_time ⇒ DEATH_EVENT=alive
2	1.13	0.77	serum_sodium=high_sodium ⇒ DEATH_EVENT=alive
3	1.04	0.71	high_blood_pressure=std_blood_pressure ⇒ DEATH_EVENT=alive
4	1.04	0.67	DEATH_EVENT=alive ⇒ high_blood_pressure=std_blood_pressure
5	1.27	0.64	DEATH_EVENT=alive ⇒ time=long_time
6	1.13	0.61	DEATH_EVENT=alive ⇒ serum_sodium=high_sodium

מכך נוכל להסיק ששני האלגוריתמים הסיקו מסקנות דומות ועל כך **כל אחד** מחוקי ההקשר של אלגוריתם FP-Growth זהים ל 6 מתוך 10 מחוקי ההקשר של אלגוריתם א־פריורי.

עם זאת, אלגוריתם א־פריורי הניב 5 חוקי הקשר נוספים,אשר אינם כוללים את עמודת המטרה.

מבחינת קבוצות השכיחות (תדירות) לא ניתן להבחין בהבדלים **מהסיבה ש־ Weka** לא מחזירה עבור אלגוריתם FP-Growth את קבוצות אלו.

מדד נוסף להשוואה הוא זמן ריצת האלגוריתמים ־ מכך ששט המידע שלנו לא רחב מספיק, קשה מאוד להבחין בין זמני הריצה של האלגוריתמים ותוכנת Weka גם לא מציגה אותם.

מבחינת זמן ממשי, כזה שלא נמדד בעזרת profiler כזה או אחר, נראה ששני האלגוריתמים סיימו לרוץ בשבריר של שנייה אף על פי העובדה שהאחד פותר את הבעיה בזמן (ומקום) אקספוננציאלי.

מה שכן ניתן להבין לגבי אלגוריתם א־פריורי הוא שהוא ביצע לכל היותר 3 איטרציות כיוון שגודל קבוצת התדירות הגדולה ביותר הוא 2, אך עם זאת ניתן להבחין שמספר המחזורים שבוצעו באלגוריתם א־פריורי הוא 12 ־ מספר שנקבע ככל הנראה על ידי הדלתא (δ) שהוגדרה לפני הרצת האלגוריתם.

מבחינת הסקת המסקנות ־ נוכל להסיק שהעבודה שביצענו בממ"ן 21 היא טובה, שכן שני חוקי הקשר שכוללים את עמודת המטרה תחת **שני האלגוריתמים** מקבילים למה שהסקנו בממ"ן 21 וגם מאוד הגיוניים בהתאם לבעיה הנתונה לנו!

שוב, גם בממ"ן זה וגם על פי שיטת הכרייה הנ"ל של חוקי הקשר מצאנו שיש קורלציה יחסית גבוהה בין עמודת ה־ time לעמודת המטרה שלנו DEATH_EVENT.

נוכל להסיק מהנאמר לעיל שאכן השיטות שנחברו מועילות ומתאימות לבעיה הנתונה, כך גם ה־ pre-proccesing שהתבצע.

עוד ניתן להסיק שאולי סט נתונים רחב יותר עם יותר פריטים יכל לתרום לחוקי הקשר יותר חזקים עם מידת confidence גבוהה יותר **ובנוסף** יכל לתרום לחלוקת הנתונים ליותר bins ־ דבר שיכול לגרור חוקי הקשר יותר מדויקים והוספת הערך ההפוך של עמודת המטרה לחוקי ההקשר (ערך dead).

כמו־כן יש לציין שבאותה נשימה עם המסקנה לגבי גודל סט הנתונים יש להבחין כי הגדלת סט הנתונים חייבת להתבצע יד ביד עם נתונים איכותיים הכוללים התפלגות נורמלית על מנת למנוע צידוד (רפרנס לממ"ן 21 שלי) ולעזור לתקינות הערכים תוך שימור ממד התכונות על מנת לא להתקל בבעיות כאל ואחרות שנוגעות לזה.

חלק II

ניתוח אשכולות - Cluster analysis

א נגדיר מהו ניתוח אשכולות

ניתוח אשכולות (Cluster analysis) הוא תהליך של חלוקת קבוצת רשומות (תצפיות) בסט נתונים לתתי־קבוצות, באופן בו כל תת־קבוצה מכילה רשומות הדומות זו לזו ע"פ סטנדרטים או מאפיינים כאלו ואחרים ובו־זמנית שונות מרשומות בתתי־הקבוצות האחרות.
ניתוח אשכולות מאפשר לסווג תת־קבוצה על פי מכנה משותף מסוים המבדיל אותה משאר תתי־הקבוצות.
עד כה, התעמקנו בתהליכי סיווג שהם supervised, אך ניתוח אשכולות הוא תהליך סיווג unsupervised - תהליך לימוד לא מפוקח של למידה מדוגמאות, כלומר למידה של מאפיינים כאלו ואחרים מסט נתונים לא מתוייג.
הקבוצות (אשכולות) שיימצאו לאחר החלוקה אינן ידועות מראש, לא מבחינת כמות ולא מבחינת תוכן ו/או קשר ביניהן.
מכך, נוכל להבין שניתוח אשכולות יכול להסיק מסקנות שלא נראות לעין האנושית ובכך יכול לתרום בגילוי תבניות חדשות, חריגות או מכנים משותפים שתהליכי סיווג אקטיביים בהם אנחנו משחקים תפקיד חשוב של תיוג מחלקות או קבוצות בעצמינו לא יכולים לגלות.

ניתוח אשכולות יכול לשמש לסיווג וכיום הוא ידוע במספר תחומים כמו ראייה ממוחשבת, מיפוי גיאוגרפי, מודיעין ואפילו לניתוח ראשוני של סט נתונים כזה או אחר על מנת לעזור לצוותי data-science במשימות של feature-engineering ועוד.

ב נגדיר מדדי איכות לאשכולות

ניתן לחלק את מדדי איכות האשכולות לשני מדדים מרכזיים, האחד הוא מדד פנימי שבוחן כל אשכול שנוצר תחת מודל הפרדיקציה, השני הוא מדד חיצוני שמודד את כלל המודל ולא מתמקד באשכולות עצמם.
מדד חיצוני יכול להיעזר בקריטריונים הבאים

1. יכולת התמודדות עם כמות גדולה של מידע
2. יכולת התמודדות עם נתונים רועשים ו- outliers
3. יכולת התמודדות עם נתונים רב־מימדיים מטיפוסים שונים
4. יכולת הסתגלות של מודל הפרדיקציה לנתונים חדשים שנכנסים

מדד פנימי יכול להיעזר בשני מדדי איכות בסיסיים:

- **הומוגניות ושלמות -** מדד שעל פיו נוכל להחליט עד כמה "המרחק" בין עצמים בכל אשכול הוא קטן, או, במילים אחרות - עד כמה העצמים בכל אשכול ואשכול דומים או שונים מעצמים באשכולות האחרים. ככל שעצמים בתוך האשכול דומים האחד לשני אך שונים מעצמים באשכולות אחרים, כך החלוקה לאשכולות יכולה להמדד כיותר איכותית.
- **מגמתיות -** מדד שעל פיו בודקים האם קיימת מגמה או שמא תופעה, שלא נתפסת בעין האדם שניתן ללמוד עליה מהחלוקה לאשכולות. מדד זה יכול להעיד על חלוקה טובה במידה והאשכולות שנוצרו הניבו מבנים לא אקראיים שאכן מלמדים אותנו על תובנות כאלו ואחרות שלא ידענו עליהן טרם החלוקה לאשכולות.

תחת ספריית sklearn איעזר במדד איכות בשם **silhouette_score - (מדד סילואט)** שמאפשר למדוד עד כמה כחלוקת הנתונים בסט הנתונים לאשכולות נכונה ומתאימה. המימוש בספרייה מחשב את ממוצע מקדם הסילואט לכלל הדגימות, גם מדד זה הוא מדד פנימי ומבוסס על מדידת מרחקים בקלאסטרים שנוצרו. מדד סילואט מוגדר בפשטות, פונקציית המטריקה מחזירה ערך בתחום $[-1,1]$ כאשר 1 הוא הערך הטוב ביותר, -1 הרע ביותר ו־0 מייצג אשכולות חופפים. **אשתמש** גם במדד ההומגניות כמובן.

ג נבחר שתי גישות לניתוח אשכולות, נסביר אותן ואת הנימוק לבחירתן

נבחר בשתי גישות מוכרות, אלגוריתם K-Means ואלגוריתם DBSCAN.

אלגוריתם K-Means המבוסס על עקרון החלוקה

עקרון החלוקה: פיזור עצמים או פריטים בין מחיצות (partitions) זרות בעזרת פונקציית מרחק/דמיון כך שבכל מחיצה נמצאים פריטים קרובים/דומים

לעיתים המחיצות נקראות גם מרכזי כובד (Centroid).

אלגוריתם K-Means, הוא אלגוריתם איטרטיבי המקבל ערך k הקבוע את מספר האשכולות הנדרשים (היפר־פרמטר) וסט נתונים בגדול n ובעזרת עקרון החלוקה ופונקציית מרחק מתאימה יוצר k מחיצות בהן הוא מקבץ את n הפריטים שהתקבלו בסט הנתונים תוך מזעור ריבועי המרחקים (WCSS) מכל מרכז באשכול. כמו כל אלגוריתם השייך ל־ Unsupervised Learning, גם באלגוריתם זה **לא מתבצע אימון**, למעשה התחזית מתבצעת על כלל סט הנתונים הנתון. כמו־כן, נזכיר שקיים trade-off בין השאיפה למזעור WCSS ומספר האשכולות הרצויים - ככל ש־k גדל, כך WCSS יקטן - דבר שמתיישב עם הגיון הגדרת הסנטרואיד. לפיכך, מצד אחד נרצה לבחור k גדול שימזער את WCSS אך מאידך, חלק מנימוק הבחירה שלי באלגוריתם זה כולל את הרצון לפשט את הנתונים למספר סביר של אשכולות, כזה שיגרור אנליזה נוחה של המידע. נשתמש בשיטת המרפק (Elbow Method) לפתרון סוגייה זו. הרעיון הוא לבחור את ה־k הקטן ביותר שממנו שיפור המדד ה־WCSS הוא מתון במידה סבירה. שיטה זו היא היריסטית ואין דרך חד־משמעית לקבוצע שה־k הנבחר הוא האופטימלי - משמע ההחלטה הסופית לגבי ערך k נתונה לשיקול דעתנו.

נציג את האלגוריתם כפי שהצגנו את האלגוריתמים בחלק הראשון של הפרויקט:

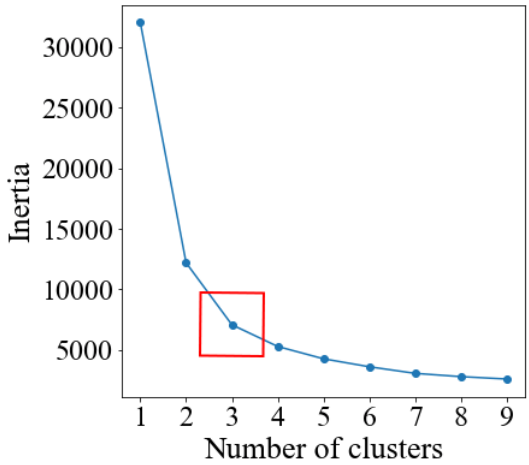
אלגוריתם 3 אלגוריתם K-Means

1. בחר k פריטים רדומליים מתוך סט הנתונים
2. מקם את סדרת מרכזי הכובד $C = \{c_1, ..., c_k\}$ בצורה רנדומלית עבור k הפריטים
3. **כל עוד** לא התכנסנו או לא הגענו למגבלת האיטרציות:
4. **עבור כל** פריט x_i:
5. מצא את מרכז הכובד הקרוב ביותר ל־ x_i מכלל מרכזי הכובד C והכניסו ל־ c_{x_i}
6. הכנס את הנקודה x_i לאשכול של מרכז הכובד c_{x_i}
7. **עבור כל** אשכול **מ־** 1 **עד** k :
8. מקם את מרכז הכובד המורכב ממוצע הפריטים שנמצאים באשכול זה מחדש
9. **החזר** את האשכולות שנוצרו

התכנסות משמעה אי־אפשר ליצור עוד מרכזי כובד חדשים, או, במילים אחרות: C של איטרציה קודמת שווה ל C של איטרציה נוכחית.

איור 4 בעמוד הבא מתאר את שיטת המרפק שהוצגה מעלה, בתוספת יתרונות, חסרונות ונימוק הבחירה באלגוריתם.

איור 3: שיטת המרפק



חסרונות

1. אלגוריתם זה אינו אופטימלי בהכרח באופן החלוקה לאשכולות, שכן מבוצעות תחילה בחירות רנדומליות כפי שמוגדר בשלבים 1 ו־ 2 באלגוריתם
2. הגדרת פונקציית המרחק יכולה לפגום באופטימליות האלגוריתם במידה ולא מתאימה לבעיה.

יתרונות

1. אלגוריתם טבעי ופשוט להבנה
2. קל למימוש
3. ביצועים טובים מבחינת סיבוכיות זמן הריצה, עבור n נקודות, k אשכולות ומגבלת איטרציות t נקבל חסם $O(nkt)$ כך שברוב המקרים זמן הריצה הוא לינארי

(א) נציין שהבעיה של מזעור מרחקים בין נקודות היא בעיה NP-hard, לפיכך לא ידוע על פתרון פולינומיאלי עבורה, אך כפי שהוזכר, אלגוריתם זה לא מניב את החלוקה האופטימלית בהכרח.

נימוק הבחירה - כיוון שהאלגוריתם פשוט מאוד להבנה, קל למימוש, עוזר בפישוט המידע למספר סביר של אשכולות וביצועיו טובים בחרתי בו.

אלגוריתם DBSCAN המבוסס על עקרון הצפיפות

עקרון הצפיפות: ייצור אשכולות על פי אזורים עם צפיפות גבוהה של פריטים המוגדרת על ידי פרמטר ε המהווה רדיוס מנימאלי לשכנות בין 2 נקודות.

אלגוריתם DBSCAN (Density-based spatial clustering of applications with noise) הוא אלגוריתם שנועד לגשר על פערי הרגישות לרעשים שחווים אלגוריתמים כמו K-Means, אשר מושפע מנקודות קיצון שיכולות לפגום במרכזי הכובד ולבסוף ביצירת האשכולות. האלגוריתם משומש כיום בהרבה עבודות computer vision.

עקרון הצפיפות המוצג מעלה, מציג גישות ואלגוריתמים שיכולים להתמודד עם רעשים ומצוא מבנים לא טרוויאלים על בסיס תכונות הצפיפות בין נקודות בסט הנתונים.

על מנת להציג את האלגוריתם, נציג תחילה 3 הגדרות בסיסיות:

הגדרה ג.1 רדיוס מינימלי לשכנות בין 2 נקודות - ε

הגדרה ג.2 מספר הנקודות המינימלי בסביבת ε של נקודה x יסומן בשם $MinPts$ או תחת הסימון $N_\varepsilon(x)$

הגדרה ג.3 נקודות ליבה הן נקודות הנמצאות במרכז צפוף דיו, באפן כזה שמרוכזות סביבן ברדיוס ε לכל הפחות $N_\varepsilon(x)$ נקודות.

כאשר נקודות הליבה ומספר הנקודות המינימלי בסביבת ε מבטיח שנקודות קיצון ורעשים כאלו ואחרים לא יפגמו בתהליך החלוקה לאשכולות.

האלגוריתם:

אלגוריתם 4 אלגוריתם DBSCAN

1. מצא את הפריטים שנמצאים בסביבת ε של כל פריט בסט הנתונים וזהה את נקודות הליבה המקיימות את הגדרה **ג.3**
2. מצא את רכיבי הקשירות של נקודות הליבה
3. **לכל** נקודה שהיא לא נקודת ליבה p_i:
4. **אם** p_i בסביבת ε של אשכול שהוגדר על פי נקודת ליבה:
5. קשר את נקודה p_i לאשכול זה
6. **אחרת**:
7. הגדר את נקודה p_i כרעש
8. **החזר** את האשכולות

חסרונות

1. מתבסס גם הוא על פונקציית מרחק מספיק טובה, אותה אנחנו צריכים לוודא

יתרונות

1. חסין לרעשים
2. יעיל - בהינתן סט נתונים בגודל n ניתן לחסום את ריצת האלגוריתם $O(n \lg n)$ במקרה הממוצע ו- $O(n^2)$ במקרה הגרוע.

נימוק הבחירה - כיוון שבממ"ן 21 ראינו שאנחנו חשופים לרעשים בסט הנתונים הקיים, נראה שאלגוריתם זה יכול לבוא לידי ביטוי בצורה טובה, כמו-כן לא יצא לי לעבוד איתו עדיין, רציתי להתנסות.

ד נתאר את שלבי ניתוח האשכולות עבור 2 הגישות שציינו בסעיף ג' תוך התייחסות לאופן הכנת הנתונים, הפרמטרים וערכם

ד.1 הכנת הנתונים

בחנתי מספר סטים שונים של נתונים, בין אם סטים גולמיים ובין אם סטים שביצעתי עליהם מניפולציות כאלו ואחרות. התוצאות תחילה, לא היו מזהירות, הקלאסטרים היו בעלי מדדים נמוכים והפיזור היה מאוד מוזר. לבסוף בחרתי להשתמש בסט הנתונים שהשתמשתי בו בתחילת הממ"ן הנ"ל אשר חולק ל 2 bins.
את כלל הנתונים בדקתי עם עמודת המטרה *לאחר שיחה עם ד"ר שולה שצמן בשעות ההנחייה הטלפונית שהמליצה לי לעשות כך.

ד.2 פרמטרים וערכיהם

ד.2.1 אלגוריתם K-Means

כפי שניתן להבין משם האלגוריתם, הפרמטר העיקרי הוא פרמטר k שמייצג את מספר האשכולות המבוקשים לחלוקה (בהתאם לפרמטר זה, כפי שכבר הצגנו, מתבצעת חלוקה ראשונית לאשכולות הראשונים).
מספר האיטרציות המקסימלי הוגדר כ־ 300, מספר הפעמים שהאלגוריתם ירוץ עם centroids שונים הוגדר כ־ 10 והאלגוריתם לחישוב הקלאסטרים היה אלגוריתם $elkan$ שמומש תחת ספריית $sklearn$ וחוזקתו היא העובדה שהוא משתמש באי־שוויון המשולש על מנת לייצר אשכולות בצורה יעילה יותר (אלגוריתם ברירת המחדל של $sklearn$), כמו־כן השתמשתי בשתי שיטות לחלוקת k הפריטים הראשונים, האחת היא השיטה האקראית והמוכרת המתוארת בפירוט בתיאור האלגוריתם והשנייה היא שיטת k++ שממומשת על ידי ספריית $sklearn$ ובוחרת את k הפריטים הראשונים בצורה "חכמה" על מנת להתכנס במהירות.
פונקציית המרחק בה משתמש המימוש של $sklearn$ היא מרחק אוקלידי בין נקודות שמתואר כך: $d(x,y)=\sqrt{\sum_{i=1}^{n}(x_i-y_i)^2}$ לכל זוג נקודות x,y מכל ממד.

ד.2.2 אלגוריתם DBSCAN

כפי שכבר צוין בהסבר על האלגוריתם, אלגוריתם זה מקבל 2 פרמטרים עיקריים:

1. **פרמטר** ε (eps) ־ **גודל רדיוס מינימלי** המייצג שכנות בין 2 נקודות: במטרה להשתמש בעקרון הצפיפות שהוצג, פרמטר זה מתקבל על מנת לקבץ יחד את הנקודות הקרובות, הוא מוגדר תחילה כערך 0.5 אך שומש כהיפטר־פרמטר על מנת לבדוק התכנסות עם ערכים שונים. נבחין כי ככל שנקטין את ערך ε כך יהיה "קשה" יותר לייצר אשכולות, שכן אנחנו מחמירים את חוק הקירבה בין נקודות/פריטים בסט הנתונים ־ ניקח זאת בחשבון.

2. **פרמטר** $N_\varepsilon(x)$ (min_samples) ־ **מספר נקודות מינימלי** בסביבת ε של נקודה x כלשהי על מנת שתהווה נקודת ליבה: גם כאן, כמו עם פרמטר ε ישנו טרייד־אוף די צפוי ומובן, מצד אחד, ניתן להגיד את המספר במטרה לקבל אשכולות עם יותר נפח, מאידך הגדלה יתר על המידה עלולה להוביל למצב בו כלל הנקודות יסווגו לאשכול יחיד והשאר יסווגו כרעש. **השארתי פרמטר זה כהיפר־פרמטר.**

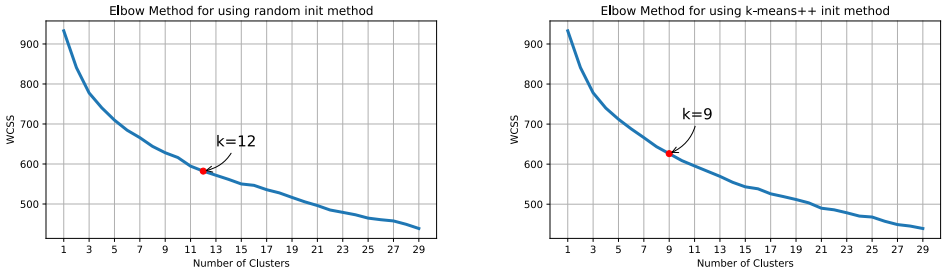
עם זאת, ביחד עם 2 הפרמטרים העיקריים, מתקבלים עוד פרמטרים חשובים, ביניהם פרמטר metric שמייצג את פונקציית המרחק בה ישתמש האלגוריתם, כאשר ערך ברירת המחדל הוא שימוש בפונקציית המרחק האוקלידי שצוין כבר תחת אלגוריתם K-Means. עם זאת, בחרתי להשתמש גם בפונקציית מרחק מנהטן שמוגדרת כדלהלן: $d(x,y)=\sum_{i=1}^{n}|x_i-y_i|$ לכל זוג נקודות x,y מכל ממד.
פרמרט נוסף שניתן להעביר למימוש של $sklearn$ הוא פרמטר algorithm שאחראי לבחירת מימוש NearestNeighbours איתו יחשבו את השכנים הקרובים ־ בחרתי לא לשנות את הפרמטר הנ"ל כיוון שמימוש $sklearn$ משנה אותו באופן דינאמי על ידי בחינת סט הנתונים בזמן אמת.

ה נדווח את תוצאות הניתוחים עבור כל גישה

1.ה תוצאות אלגוריתם K-Means

נעזרתי במדד ה ־ score שמספקת ספריית *sklearn* שמביע את ערך WCSS עבור K-Means ובעזרתו בניתי גרף שישמש את Elbow Method, מצאתי את התוצאות הבאות:

איור 4: תוצאות Elbow Method לכל אחד מן הסטים

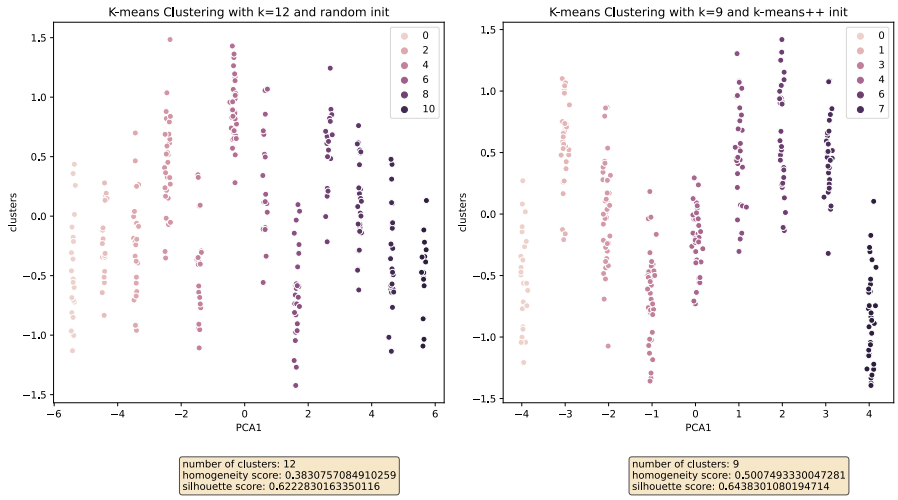


ניתן להבחין כי תוצאות השיטה מראות שעבור סט הנתונים, אין נקדה בה שיפוע הפונקציה קטן בצורה דרסטית אך קל לראות שצפיפות הנקודות הולכת וגדלה בערך בנקודה של 9 אשכולות עבור איתחול עם k++ ו 12 עבור אתחול אקראי ־ אבדוק את שתי השיטות.

תוצאות ההרצות הטובות ביותר ביחד עם תיאור פרמטרי ההרצה, $k \equiv$ מספר האשכולות, מדד סילואט ומדד הומוגניות מצורפות באיור 5 ונראה שערך מדד הסילואט הטוב ביותר שנמדד הוא 0.648 ומדד הומגניות 0.45 עם 12 אשכולות ואלגוריתם חילוק התחלתי אקראי ומדד סילואט 0.643 עם הומגניות 0.5 ו 9 אשכולות עבור אתחול עם k++. יש להבחין בעובדה ששיניתי את הפרמטרים שצוינו בסעיף 2.1.ד, ביניהם פרמטר שיטת החלוקה הראשונית וכמובן k על מנת לוודא את תקינות הערכת ה Elbow method.

בנוסף, צריך להבחין שעל מנת לייצר scatter_plot יש צורך לבצע טרנספורמציה לינארית למזעור מממדי הטבלה, השתמשתי ב sklearn ו PCA על מנת לבצע זו (שיטה מקובלת שמבצעים לפעמים גם לפני חלוקה לאשכולות על מנת לבצע רדוקציה לממד הבעיה).

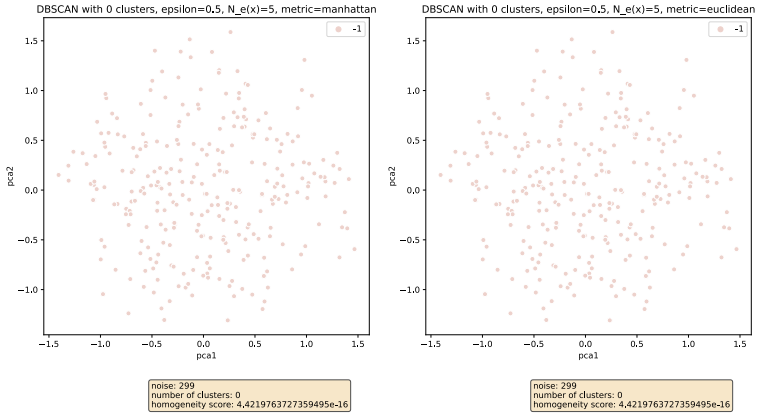
איור 5: תוצאות ההרצה הטובות ביותר של אלגוריתם K-Means



2.ה תוצאות אלגוריתם DBSCAN

תחילה נבחין שעבור הפרמטרים שמתקבלים כברירת מחדל על ידי ספריית sklearn לא נקבל כלל אשכולות **וכל הנקודות** מוגדרות כרעש, קל להבין מדוע, הרי החלוקה ל־bins גרמה למנעד הערכים להיות שייך ל $\mathbb{N}$ ולכן הפרמטרים ברירת המחדל $\varepsilon = 0.5, N_\varepsilon(x) = 5$ יהיו בעייתיים.

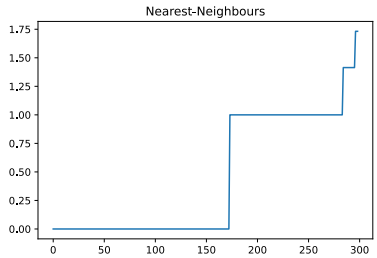
איור 6: דוגמה עבורה $\varepsilon = 0.5$ מתקבל שכלל הפריטים הם רעש (גם עבור מרחק מנהטן שהושמט מתוקף כפילות)



לכן מצאתי עצמי מכוון אחר היפר פרמטר ε. לאחר חיפוש וקריאה של מאמר בנושא, מצאתי שיש דרך לקבוע ε אופטימלי על ידי שימוש ב־ Nearest-Neighbours, אך הדבר לא עזר בהרבה, התוצאות שקיבלתי מהרצת האלגוריתם לא היו מספקות ולכן בחרתי לנסות לבצע דיקרטיזציה שונה לנתונים. לשם כך, בחרתי לשנות את ה־bins מעומק שווה לרוחב שווה, השינוי הנ"ל עשה "ניסים".

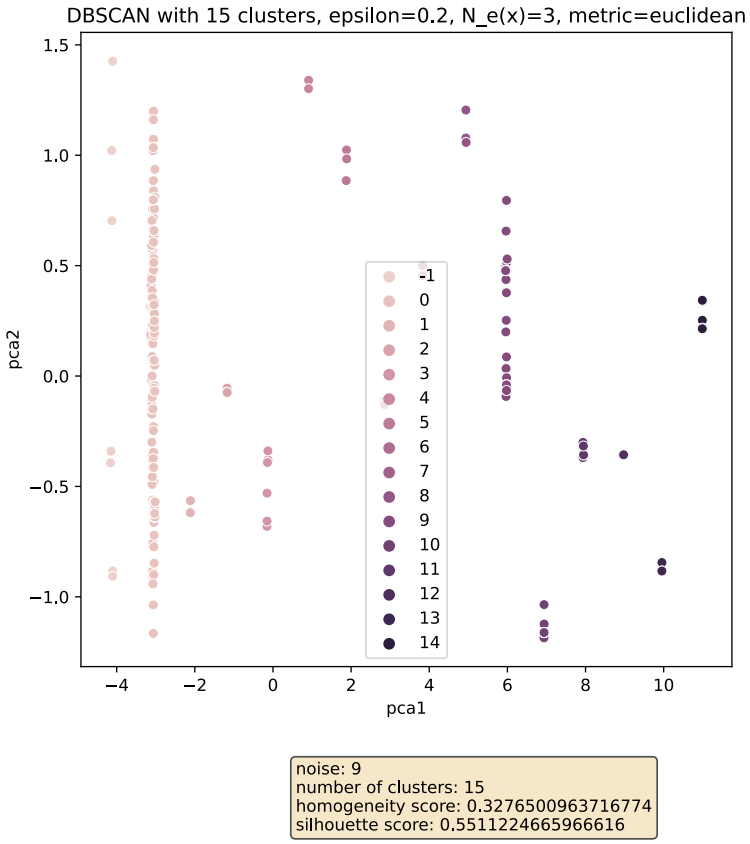
בבדיקת Nearest-Neighbours קיבלתי את הגרף הבא:

איור 7: בדיקת Nearest-Neighbours למציאת ערכי אפסילון אופטימליים

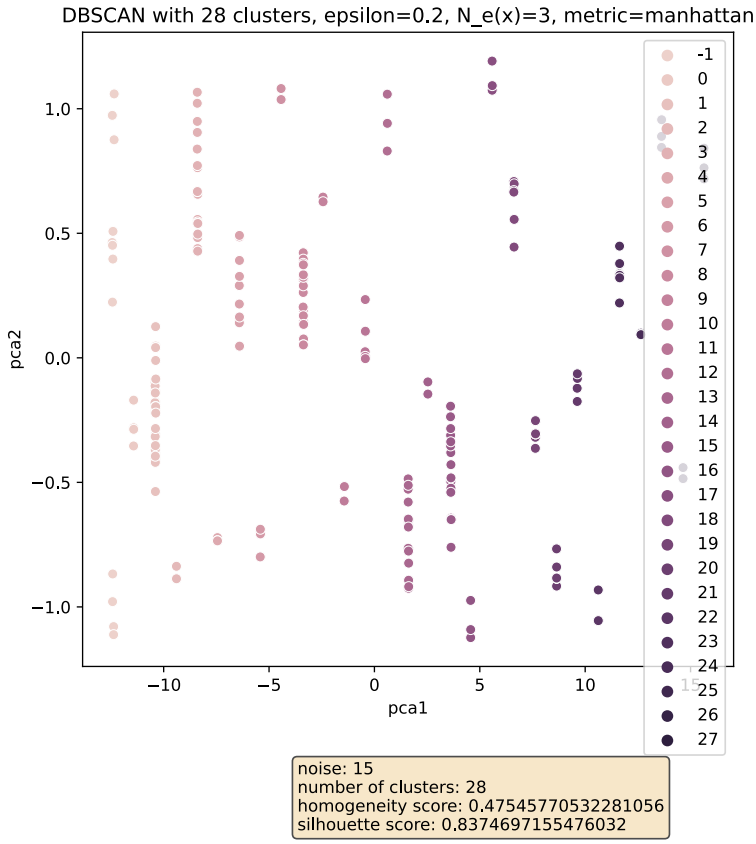


קל לראות שלא צריך סביבת אפסילון גדולה בשביל לקבל "שכנים", בחרתי ב $\varepsilon = 0.2$ וניסיתי שלל פרמטרים עבור $N_\varepsilon(x)$. תוצאות ההרצה מצורפת בעמוד הבא ביחד עם הפרמטרים

איור 8: תוצאות הרצת DBSCAN למציאת חסמי ε תחתונים לכל סט נתונים
שימוש במרחק אוקלידי, $\varepsilon = 0.2, N_\varepsilon(x) = 3$:



שימוש במרחק מנהטן, $\varepsilon = 0.2, N_\varepsilon(x) = 3$:



ריכוז תוצאות האלגוריתמים וסיכום השלב

עבור אלגוריתם K-Means

מצאנו ערכי k אופטימליים לפי Elbow-Method ולאחר מכן בעזרת מימוש של $sklearn$ תוצאות האלגוריתם היו כמעט זהות עבור שתי שיטות האיתחול k++ ו־ random.

- שימוש ב k++ עם $k = 9$ הניב לנו תוצאת מדד סילואט של 0.643 והומגניות 0.5
- שימוש באיתחול אקראי עם $k = 12$ הניב לנו תוצאת מדד סילואט של 0.648 והומגניות 0.45

עבור אלגוריתם DBSCAN

לאחר מחקר קצר ובעזרת שינוי חלוקת הנתונים לבינים ברוחב שווה ושימוש ב־ Nearest-Neighbours הצלחתי למצוא ערכי ε מתאימים לאלגוריתם DBSCAN, והתוצאות נראות טובות מבחינת תוצאות המדדים אך פחות מזהירות מבחינת החלוקה הויזואלית לאשכולות (נוצרו סטריפים של ערכים כתוצאה מהאפסילון שנבחר ומהחלוקה לבינים).

- שימוש במרחק אוקלידי הניב לנו 15 אשכולות, מדד סילואט 0.55 והומגניות 0.32, רק 9 נקודות הוגדרו כרעש
- שימוש במרחק מנהטן הניב לנו 28 אשכולות, מדד סילואט 0.837 והומגניות 0.475 רק 15 נקודות הוגדרו כרעש

ו ננתח השוואתית את התוצאות ונסיק מסקנות

להלן טבלה שמשווה בין התוצאות:

טבלה 4: תוצאות האלגוריתמים

אלגוריתם	פונקצית מרחק	שיטת איתחול	Silhouette Score	הומוגניות	מספר אשכולות
K-Means	אוקלידי	אקראית	0.643	0.45	12
K-Means	אוקלידי	k++	0.648	0.5	9
DBSCAN	אוקלידי	-	0.55	0.32	15
DBSCAN	מנהטן	-	0.837	0.475	28

כפי שניתן לראות, אלגוריתם DBSCAN הוא זה שהניב את התוצאות הטובות ביותר מבחינת מדד סילואט. עם זאת, לא ניתן להתעלם מהעובדה שמדד ההומוגניות של אלגוריתם K-Means עם אתחול k++ הציג תוצאה של 0.5, תוצאה לא רעה כלל.
עוד ניתן להבחין, שבהתאם לסט הנתונים הנ"ל, השינוי בפונקציית המרחק שינה במידה גדולה את תוצאות DBSCAN, לצורך ההבנה, ההבדל באלגוריתם K-Means נוצר כתוצאה משיטת אתחול הנקודות למרכזי הכובד ולא על פי המרחקים!
בנוסף, לאחר בדיקה, נראה שתצורת האשכולות לא מניבה לנו מסווג כלשהו, בכל אחד מארבעת התוצאות הטובות ביותר, ערכי עמודת המטרה DEATH_EVENT מפוזרים בין האשכולות השונים ולא נראה שיש קורלציה מסויימת בין עמודת המטרה לאשכול כזה או אחר.
להלן פילוח הערכים בין האשכולות לכל אחד מהאלגוריתמים שמוצגים בטבלה 4:

טבלה 5: פילוח הערכים בין האשכולות לכל אלגוריתם

(א) 12-Means random

עמודת מטרה	# אשכול											
DEATH_EVENT	0	1	2	3	4	5	6	7	8	9	10	11
0	24	8	34	4	21	6	25	24	5	9	23	20
1	2	18	2	24	3	16	4	-	14	13	-	-

(ב) 9-Means random

עמודת מטרה	# אשכול								
DEATH_EVENT	0	1	2	3	4	5	6	7	8
0	11	34	28	4	37	28	28	30	3
1	14	-	13	31	1	4	2	1	30

(ג) DBSCAN euclidean

עמודת מטרה	# אשכול															
DEATH_EVENT	-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	2	123	1	-	-	-	-	-	-	4	37	10	9	5	3	9
1	7	57	2	5	8	3	4	5	3	1	1	-	-	-	-	-

(ד) DBSCAN manhattan

עמודת מטרה	# אשכול																													
DEATH_EVENT	-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
0	5	-	14	1	16	-	6	-	-	22	-	-	1	5	20	-	23	-	4	14	12	20	9	4	13	5	7	3	9
1	10	6	18	2	10	5	5	5	3	4	3	5	5	-	4	3	3	3	1	1	-	-	-	-	-	-	-	-	-

אסכם את הנתונים לכל טבלה ואסיק מסקנות

1. **א** - ניתן להבחין בבירור ש 171 (84%) ערכי 0 של עמודת המטרה נמצאים באשכולות $0, 2, 4, 6, 7, 10, 11$ כאשר בבירור מבחינים בעובדה שאשכולות $7, 10, 11$ שייכים לעמודת המטרה עם ערך 0. מאידך, 75% מערכי עמודת המטרה עם הערך 1 נמצאים באשכולות $1, 3, 5, 8$, מכך נוכל להבין את ההפרדה של האשכולות כלפי ערכי עמודת המטרה.

2. **ב** - ניתן להבחין במשקל גדול של ערכי 0 של עמודת המטרה עבור אשכולות $1, 4, 5, 6, 7$ עם תפוסה של 77%. באשכולות $3, 8$ יש ריכוז של ערך 1 של עמודת המטרה שמהווים 63% מכלל ערכי 0.

3. **ג** - ניתן להבחין בצורה מאוד ברורה שאשכולות $10, 11, 12, 13, 14$ שייכים לערך 0 של עמודת המטרה וכי יש רוב מוחלט של ערכי 0 באשכול 0. כמו־כן, אשכולות $2, 3, 4, 5, 6, 7$ שייכים לערך 1 של עמודת המטרה.

4. **ד** - ניתן להבחין בעובדה שאשכולות $19 - 27$ מיועדים לערך 0 של עמודת המטרה, כמו כן מכלל 103 הערכים הנותרים (שלא באשכולות שצוינו תחילה) כ - 63% מערכי 0 של עמודת המטרה מרוכזים באשכולות $8, 13, 15$ ולכן גם אותם ניתן לסווג כאשכולות שמתאימים לעמודת המטרה 0. מאידך, אשכולות $0, 4, 6, 7, 9, 10, 14, 16$ הם אשכולות המיועדים ככל הנראה לערך 1 של עמודת המטרה.

יש להבחין בעובדה שמבחינת התפלגות הנתונים של DEATH_EVENT (עמודת המטרה) , יש חוסר איזון בין כמות הנבדקים שנפטרו לאלו שלא כאשר כפי שצוין בממ"ן 21, ישנם 203 ערכי 0 ו- 96 ערכי 1 $\Leftarrow$ הדבר מסביר בצורה ברורה את התוצאות שאנחנו רואים בחלוקה לאשכולות!

- **עלו רעיונות לשפר את הביצועים** של האלגוריתמים בעזרת הכנת הנתונים בצורה שונה, נסיון להשתמש בסוגי מרחקים אחרים ממה שבחרתי ואולי אפילו לשנות את ערכי $N_\varepsilon(x)$ ו- ε יותר.

חלק III

סיכום ומסקנות

מבין כלל המודלים והאלגוריתמים שהשתמשנו בהם בכלל הפרויקט, נראה שאלגוריתם CART (עץ החלטה) הניב את התוצאות הטובות ביותר עם אחוזי דיוק גבוהים ביותר.
בנוסף, באופן עקבי מצאנו דפוסים דומים בין השיטות המפוקחות (עצי החלטה) ללא מפוקחות (חוקי הקשר), אשר הראו קורלציה כזו או אחרת בין מדדים שניתנו בסט הנתונים לבין עמודת המטרה, כאשר אלגוריתמי עץ ההחלטה ואלגוריתמי חוקי ההקשר הצליחו לזהות אותם בצורה ברורה, למשל העובדה שעמודות serum_sodium ו־ time מופיעות בעצים של ממ"ן 21 ושני אלגוריתמי חוקי ההקשר בתחילת ממ"ן זה, מחזקת את הטענה שעבודת חקר הנתונים הראשונית התבצעה כראוי.
בנוסף, נוכל להסיק שכמות הנתונים שהתקבלה בסט הנתונים היא יחסית נמוכה עבור אלגוריתמי ייצור אשכולות, בנוסף לעובדה שהיא יחסית מוטה ביחס לעמודת המטרה ויש בה רעשים כאלו ואחרים שפוגמים בטיב אלגוריתם כמו K-Means.
ארצה להציג כמה נקודות חשובות לטעמי:

- תהליך הכנת הנתונים הוא תהליך (מאוד) חשוב שיכול לשנות תוצאות אלגוריתמים ב־ $180°$.
- אין תהליך נכון או לא נכון ־ הכנת הנתונים בצורה כזו או אחרת יכולה לתרום או לפגום לחלוטין באלגוריתמים שאנחנו מריצים, הדבר בא לידי ביטוי בעיקר בחלק השני של הממ"ן הנ"ל בחלק של ניתוח אשכולות. נושא ה־ pre-proccesing הוא נושא חשוב שחובה להעמיק בו יותר ויותר ויתרה מזאת ־ יש להבין שאין דרך נכונה לעשות אותו!
- שימוש בכמה וכמה מתודולוגיות עבודה וסוגים שונים של למידה יכול להביע לנו עד כמה ההנחות שלנו על הקלט נכונות ועד כמה הכנת הנתונים שלנו התבצעה כראוי
- אין אפשרות לכתוב את ממ"ן 21 או 22 בצורה חלקה ובבת אחת ־ במשך הכתיבה והעבודה מצאתי עצמי לומד המון מעבר לחומר הקורס, תוך העמקה אל white-papers כאלו ואחרים ושכתוב קוד יחסית גדול בכל פעם שהבנתי "קצת יותר" את הבעיה שלי ואת דרכי הפתרון שבחרתי.

בנימה אישית

אני יודע שהפרויקט ארוך (אם כי יש חלק לא קטן של איורים וטבלאות), אך הוא כזה כי יצר בי עניין שאף קורס אחר עד כה לא הצליח ליצור.
האתגר בלהבין סט נתונים שלא פגשתי בחיים, לנסות להסיק מסקנות ממידע שאני לא מכיר, ללמוד איך לבטא את המילה "טסיות דם" בשפה האנגלית, אלו דברים לא טרוויאלים שאני מעריך מאוד.
תהליך כריית המידע הוא תהליך מחזורי, תהליך של כישלון, הצלחה, עצבים ושמחה ואולי קצת בכי כי שכחת לשמור את העבודה שביצעת ב־ Jupyter-Notebook וצריך לכתוב את הקוד מחדש.
תודה על ההזדמנות ללמוד את הנושא!

רשימת מקורות

[1] https://stanford.edu/~cpiech/cs221/handouts/kmeans.html - אוניברסיטת סטנפורד ומידע על K-Means

[2] https://en.wikipedia.org/wiki/Principal_component_analysis - שיטת PCA לרדוקצית ממד לסט נתונים

[3] https://en.wikipedia.org/wiki/Elbow_method_(clustering) - שיטת המרפק Elbow-Method למציאת K אופטימלי עובר אלגוריתם K-Means

[4] https://en.wikipedia.org/wiki/Silhouette_(clustering) - מדד סילואט למדידת איכות רמת אשכולות

[5] https://scikit-learn.org - דוקומנטציית ספריית *sklearn* בה השתמשתי במימושים

[6] https://iopscience.iop.org/article/10.1088/1755-1315/31/1/012012/pdf - מאמר שקראתי על מציאת ε אופטימלי לאלגוריתם DBSCAN