ממ"ן 22 – גיאורגי מירזויאן

309096485

<u>שאלה 1</u>

א.

נגדיר מדדי איכות לחלוקה לאשכולות:
homogeneity - מדד זה מתאר את חוסר דגימות שלא שייכות לאשכול לפי הסיווג, כלומר מדד זה יהיה גבוה יותר ככול שיש פחות דגימות שונות.

completeness - מדד זה מתאר את טיב החלוקה לאשכולות לפי הסיווג האמיתי של הדגימות, כלומר מדד זה יהיה גבוה יותר ככול שיש יותר דגימות מאותו סיווג בתוך האשכול.

f1-score - מדד זה מתאר את נכונות הסיווג של המודל ביחס לכלל מאגר המידע.

ב.

בחרתי בשיטה DBSCAN.
אלגוריתם זה מתבסס על מציאת אשכולות לפי צפיפות, לצרוך זה יש שני מדדים הניתנים להגדרה:
eps - רדיוס סביב נק'
minPts - מס' מינימלי כדי להגדיר נק' כנקודת ליבה(core point)

תחילה בוחרים נק' שרירותית, בודקים אם יש minPts בתוך רדיוס eps. נקודה זו תוגדר להיות נק' ליבה\ אשכול. רצים על שאר הנק' בתוך הרדיוס ובכל פעם מגדילים את מס' נק' ליבה. כך האשכול גדל עד אשר אין יותר נק' להוסיף לאשכול. כעט עוברים לנק' חדשות שלא ביקרנו בהן והתהליך חוזר מהתחלה.שאר הנק' ללא אשכול נקראות רעש.

ALGORITHM 1: Pseudocode of Original Sequential DBSCAN Algorithm

```
Input: DB: Database
Input: ε: Radius
Input: minPts: Density threshold
Input: dist: Distance function
Data: label: Point labels, initially undefined
foreach point p in database DB do                    // Iterate over every point
    if label(p) ≠ undefined then continue            // Skip processed points
    Neighbors N ← RangeQuery(DB, dist, p, ε)         // Find initial neighbors
    if |N| < minPts then                             // Non-core points are noise
        label(p) ← Noise
        continue
    c ← next cluster label                           // Start a new cluster
    label(p) ← c
    Seed set S ← N \ {p}                             // Expand neighborhood
    foreach q in S do
        if label(q) = Noise then  label(q) ← c
        if label(q) ≠ undefined then continue
        Neighbors N ← RangeQuery(DB, dist, q, ε)
        label(q) ← c
        if |N| < minPts then continue                // Core-point check
        S ← S ∪ N
```

יתרונות:

- יכול להתמודד עם אשכולות בצורות שרירותיות.
- יכול להתמודד עם רעשים במידע.
- אין צורך לקבוע מספר אשכולות מראש.

חסרונות:

- טיב התוצאות תלוי במדד eps, minPts.
- לא עובד טוב עם אשכולות בעלי צפיפות שונים מדי.
- חישוביות גבוהה כאשר מאגר המידע גדול.

על מנת להתמודד עם החסרונות השתמשתי בשיטות הבאות:
הרצתי מספר גדול של ריצות את האלגו' תוך כדי למצוא את מדדי האלגו' eps, minPts הטובים ביותר.
בחרתי מתוך המאגר דוגמאות השייכות לשלוש הסיווגים קבוצות בעלות מספר דומה של תצפיות.
למזלנו מאגר המידע אינו גדול כל כך.

ג.+ד.

שלבי ניתוח:

השתמשתי בנתונים ממ"ן 21 ואלה הם הטיפול בנתונים שעשיתי:

1. מחיקת נתונים:
   - תכונה hypopituitary מכילה רק ערכים שליליים, לא תורמת לנו מידע.
   - תכונה TBH מכילה 96% ערכים חסרים.
   - תכונות המכילות measured אלה תכונות המראות האם בוצע בדיקת דם, אם הערך שלילי אז לא בוצעה בדיקה ולא יהיה ערך מספרי של בדיקה.
   - תכונה referral source לא רלוונטי מאיפה הבדיקה הגיע.
   - מחיקת ערכים: ישנם הרבה מאוד ערכים חסרים לכן הסרתי אותם ממאגר הנתונים למרות הצמצם במספר הרשומות נשארנו עם יותר מחצי מהרשומות המקוריות.
   - רשומות כפולות: ישנן שלוש רשימות המכילות את אותן ערכים בדיוק גם מהן נפטר.
   - ערכים חריגים: יש מספר ערכי גיל חריגים שניתן להיתר מהם.
2. טרנספורמציית נתונים:
   - כלל הנתונים הקטגוריים שונו מ-t/f ל-1/0 בגלל אילוצים של הסיפריות בפייטון.
   - תכונת האיבחון מכילה הרבה ערכים שונים ביחד עם מספר האבחון/תיק רפואי, לכן נמחק את מספר האבחון ונשנה את הופעת הערכים לפי קבוצות אבחנה המופיעות בקובץ הסברים על מאגר הנתונים(התעלמתי משילובים והשארות באבחונים):

     hyperthyroid conditions - מקבל ערך 2

     A hyperthyroid

     B T3 toxic

     C toxic goitre

     D secondary toxic

     hypothyroid conditions - מקבל ערך 1

     E hypothyroid

     F primary hypothyroid

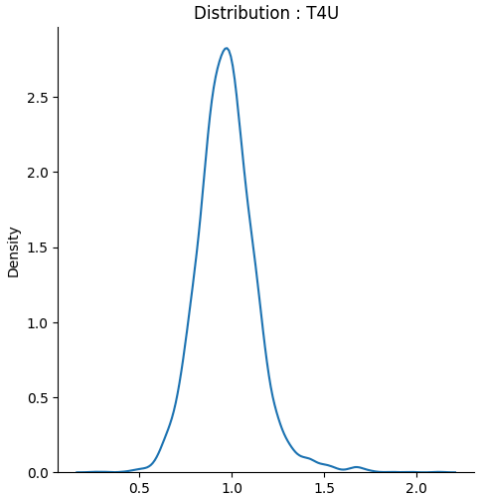
     G compensated hypothyroid

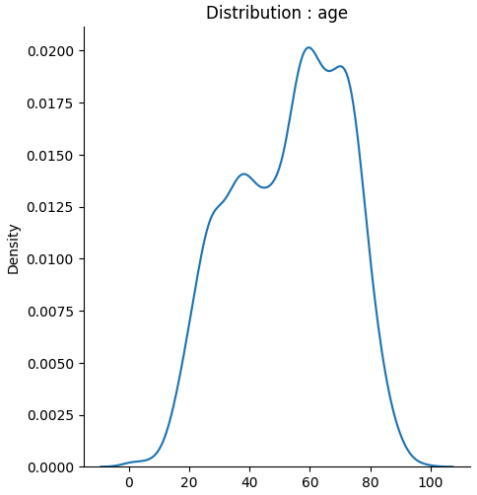
     H secondary hypothyroid

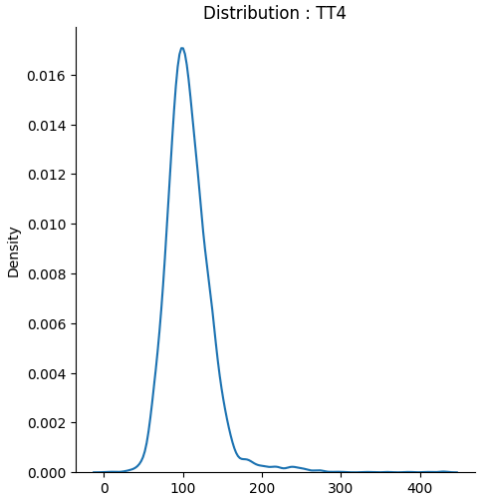
     אדם ללא הבחנה מקבל ערך 0

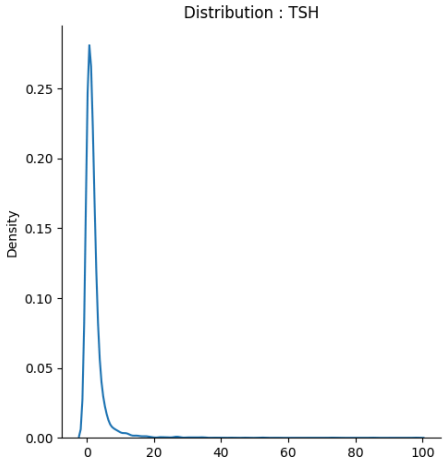
השתמשתי בנתונים ללא דיאגנוזה.ניתן לראות כי הנתונים הנומריים מתפלגים נורמלי.

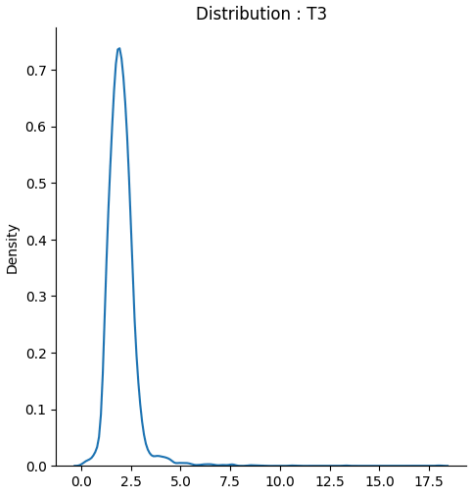
נעשה PCA, נשתמש בטכניקת הורדת מימד על מנת להוריד ל2 מימדים את הנתונים. על יהיה יותר קל להציג בצורה גרפית את הנתונים וגם נמענה מקללת הרב מימדיות המתקבלת ממספר רב של מאפיינים.



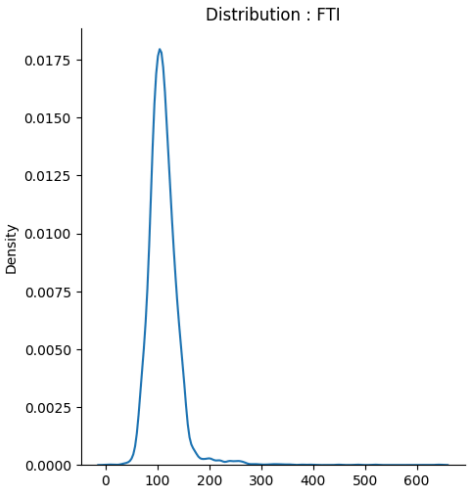








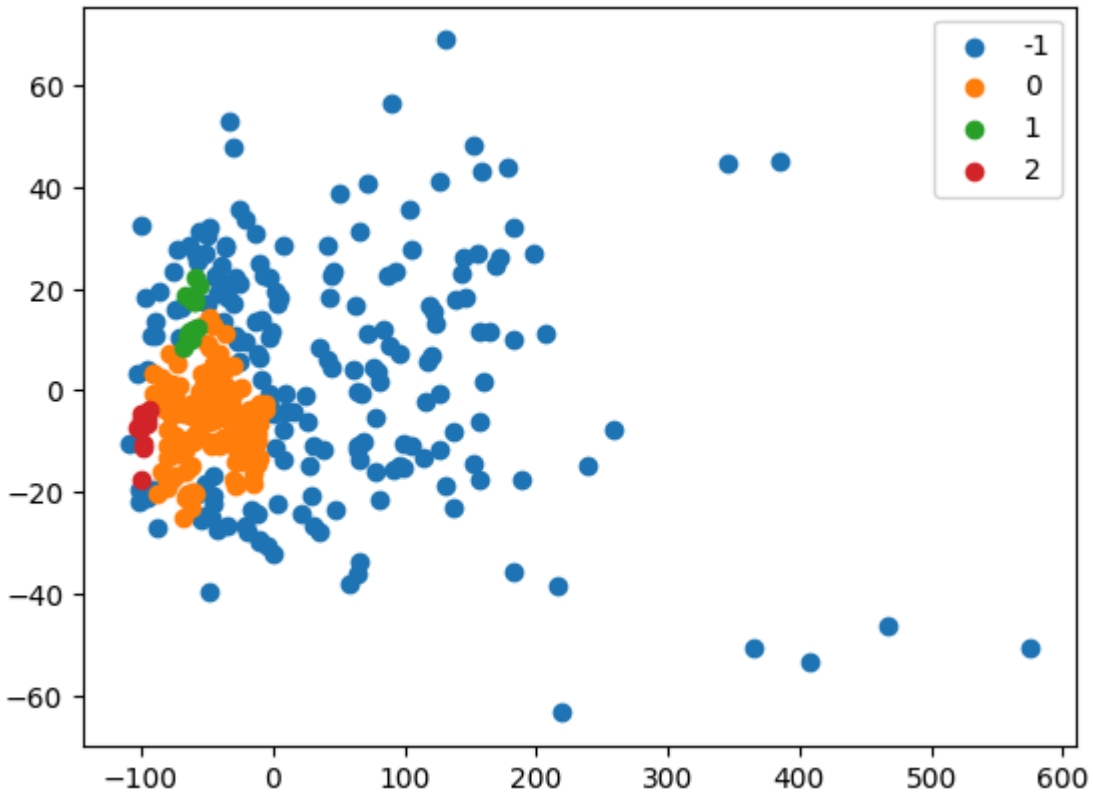
ית DBSCAN ע
neity, complete
כ 12,000 איטרצ



כתו
בכל
הפו
את
נגד

homogeneity_score_max: 0.16166726253861613
completeness_score_max: 0.21042887290910123
f1_score_max: 0.317479

weighted avg	macro avg	micro avg	hypo_cond	hyper_cond	healthy	
0.409242	0.409242	0.458599	0.090909	0.666667	0.470149	**precision**
0.206897	0.206897	0.206897	0.008621	0.068966	0.543103	**recall**
0.214916	0.214916	0.285149	0.015748	0.125000	0.504000	**f1-score**
348.000000	348.000000	348.000000	116.000000	116.000000	116.000000	**support**



כפי שניתן לראות זו לא שיטה טובה לחלוקה לאשכולות. סהכ 3 אשכולות(מס' 0,1,2) ו1- עבור רעש\חריג. כל מדדי האיכות די נמוכים למרות שאלה התוצאות בעלות הערך המקסימלי במרחב החיפוש.אני משאר כי יש מעט נתונים מהסיווגים הלא בריאים וזה השפיע על הקטנת הקבוצה שיכולנו לבצע עליה אשכול. תובנה מעניינת על הנתונים היא למרות הנסיון לייצר 3 קבוצות בעלות כמות סיווגים שווה האלגוריתם בחר לסווג הרבה נק' בתור חריגים. אני מניח כי הורדת מימד הייתה יכולה להשפיע.

<u>שאלה 2</u>

א.
נגדיר את ארכיטקטורת הרשת.
נבנה רשת נוירונים הזנה קדימה שבה כל נוירון בשכבה i מחובר לכל הנוירונים בשכבה i+1.
אין קשתות עצמיות או מעגלים ברשת הזו וכל המידע זורם מהשכבה i לשכבה i+1 עד אשר מגיע לשכבת הפלט.

הרשת בנויה מ-3 שכבות:
שכבה ראשונה - מייצגת את שכבת הקלט בעלת 21 נוירונים, כל נוירון מייצג מאפיין במאגר המידע ann-train.
שכבה שניה - שכבה חבויה המכילה 15 נוירונים.∇
שכבה שלילית - שכבת פלט המכילה 3 נוירונים, כמספר הסיווגים, בריא, היפר, היפו.

פונ' הפעלה שנבחרה היא ReLu

$$ReLu = \begin{Bmatrix} x & if x > 0 \\ 0 & otherwise \end{Bmatrix}$$

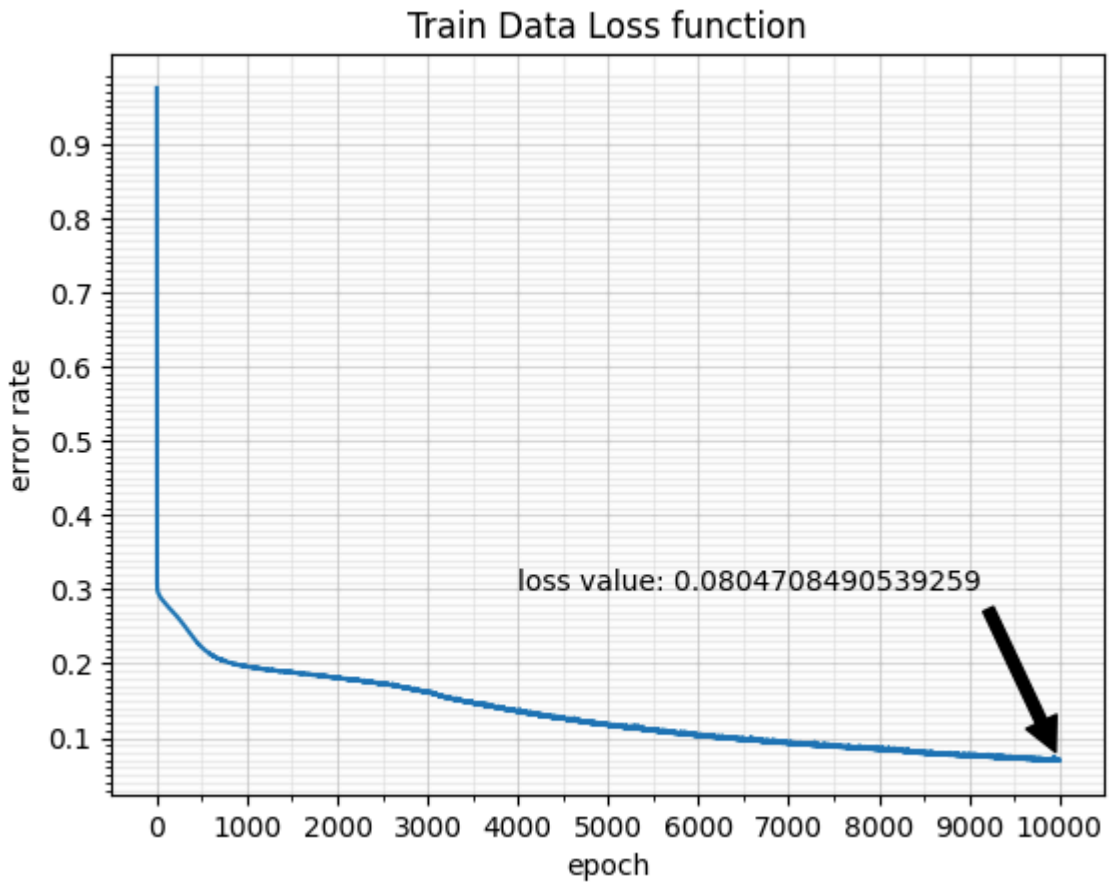
ב.
פונקציית העלות מודדת את המרחק בין התוצאה החזויה לבין התוצאה בפועל, במקרה שלנו(ומגבלות של MLPClassifier) אנו נשתמש במורד הגרדיאנט סטוכסטי.

$$w_{i+1} \rightarrow w_i - \eta \nabla L(w_i)$$

כאשר:
w - הוא וקטור משקלים
η - קצת לימוד, קצב השפעה של הגרדיאנט של פונקצית הפסד על משקלים
L(w∇) - פונקצית הפסד.
לפי המסמכים של skilearn, פונ' ההפסד היא cross-entropy והיא מודדת את מס' הביטים הנחוצים לזיהוי של המאורע החזוי.
מאחר ואנחנו משתמשים במורד הגרדיאנט סטוכסטי אין לנו באצ'ים, זוהי למדיה מכוונת(online).
קצת הלימוד הוא פרמטר המגדיר את גודל הצעד שיש לעשות בכיוון המינימום של פונקצית ההפסד, במקרה שלנו השארתי את הגודל הדיפולטי של 0.001.

ג.

כל הנתונים נמצאים בטווח[0,1] או בינאריים.
קיימת מגבלה של 10,000 אפוכים ובמקרה שיש 500 אפוכים ללא שינוי משמעותי של 1e-4 אז מגדירים כי הרשת התכנסה.



	hyper_cond	hypo_cond	healthy	accuracy	macro avg	weighted avg
precision	0.955556	0.914286	0.971601	0.96951	0.947148	0.968175
recall	0.843137	0.528926	0.998087	0.96951	0.790050	0.969510
f1-score	0.895833	0.670157	0.984666	0.96951	0.850219	0.965848
support	51.000000	121.000000	2091.000000	0.96951	2263.000000	2263.000000

confusion_matrix(y_train, rf_pred)

```
array([[ 43,   4,   4],
       [  0,  64,  57],
       [  2,   2, 2087]])
```

weighted avg	macro avg	accuracy	healthy	hypo_cond	hyper_cond	
0.970147	0.929360	0.97112	0.974438	0.927928	0.885714	**precision**
0.971120	0.808944	0.97112	0.995595	0.581921	0.849315	**recall**
0.968473	0.855771	0.97112	0.984903	0.715278	0.867133	**f1-score**

3428.000000	3428.000000	0.97112	3178.000000	177.000000	73.000000	**support**

```
confusion_matrix(df_test['diagnosis'], rf_pred)
array([[ 62,   2,   9],
       [  0, 103,  74],
       [  8,   6, 3164]])
```

ד.

train-data חולק ל60% קבוצת אימון ו40% קבוצת מבחן, test-data נבחן על המודל הגמור אחרי האימון. בtrain_data test_data אחוז הדיוק גבוהה מאוד ואחוז הrecall גבוה גם כן אך לא באותה מידה, תוצאה מעניינת היא f1_score ו recall של שני המודלים עבור חולים בתת פעילות נמוכה בהרבה משאר המאפיינים באופן משמעותי. משמעות הדבר כי המודל לא מסווג נכונה בערך 50% מהחולים בתת פעילות. f1_score הוא ממוצע הרמוני של accuracy ו recall אז recall "מושך" למטה את המדד f1_score.

ה.

המודל מסווג את רוב הדגימות נכון, כאשר רוב החולים הם אנשים בריאים, לא הייתי מסתמך על מודל זה או מודל של אשכולות משאלה 1 לחיזוי מחלה רק בגלל חוסר הדיוק של האשכולות וחוסר יכולת של מודל הרשת נויירונים לחזות אנשים חולים בוודאות, זה מצב מסוכן עם השלכות חמורות

<u>שאלה 3</u>

AdaBoost	Random Forest	DBscan	MLP	
0.864985	1	0.409242	0.929360	precision
0.933405	1	0.206897	0.808944	recall
0.895450	1	0.214916	0.855771	f1-score

הדיוק של יער אקראי מאוד גבוהה, הסכנה שיש התאמת יתר למרות שהשתמשנו ב-k-fold cross validation. אחוזי הדיוק נשמרו לאורך כל הבדיקות.DBscan ראינו שאינו ממולץ לשימוש על גבי הנתונים כפי שהם, אולי היה צריך לעבד את הנתונים כך שיהיה פיזור ברור ואחיד בתוך האשכולות אך זה לא תמיד מתאים לחיים האמיתיים.MLP עדיף על AdaBoost עקב הגמישות הגדולה יותר של רשת נוירונים(פרמטרים שניתן לשחק איתם ולהגיע לתוצאה טובה יותר).בכל אופן ניתן להגיע לתוצאות טובות יותר עם ריבוי של תצפיות של חולים, יש חוסר איזון מאוד גבוהה בין הבריאים לחולים.