

### למידה חישובית – תרגיל מספר 3

1. (35 נקודות) במחקר שנערך באנטארקטיקה החוקרים מצאו כי קיימים 3 מיני פינגווינים (152), Chinstrap(68), Gentoo (124), וכי קיימים הבדלים בין המינים במספר מדדים כמו אורך ועומק החלק העליון של המקור המכונה Culmen, אורך הכנפיים (המכונות flippers) וכן מסת הגוף (ראו דיאגרמות פיזור של המדדים באיור).

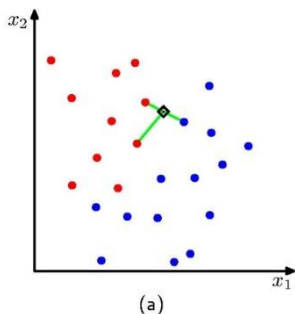
ה- dataframe מכיל 7 עמודות :

- species: penguin species (Chinstrap, Adélie, or Gentoo)
- culmen\_length\_mm: culmen length (mm)
- culmen\_depth\_mm: culmen depth (mm)
- flipper\_length\_mm: flipper length (mm)
- body\_mass\_g: body mass (g)
- island: island name

א. באמצעות הפקודות הבאות (ראו penguin\_data במודל בתיקה של exercise) materials קראו את הנתונים לתוך DataFrame :

ב. משני המינים Adélie ו- Gentoo בחרו ב- 100 ו- 80 הפרטים הראשונים בהתאמה, ומהמין Chinstrap 50 הפרטים הראשונים, ויצרו מטריצת אימון (np.array)  $X_{train}$  עם ארבע התכונות culmen\_length\_mm, culmen\_depth\_mm, flipper\_length\_mm and body\_mass\_g, וכן וקטור y\_train עם 230 תגיות מתאימות אותן יש לסמן ב- 0 (Adélie), 1 (Gentoo) ו- 2 (Chinstrap) כך שכל תגית (label) תתאים לפרט הנכון במטריצה  $X_{train}$ . באותו אופן יש ליצור מטריצת מבחן  $X_{test}$  ווקטור תגיות y\_test עבור הפרטים שלא נבחרו לנתוני האימון.

ג. כתבו פונקציית python למימוש אלגוריתם knn. נתאר את שלבי האלגוריתם. בהינתן וקטור תכונות x ומדד מרחק כלשהו :



- מתוך  $N$  וקטורי אימון, זהו את  $k$  השכנים הקרובים ביותר לוקטור  $\underline{x}$  ללא קשר לתיוג המחלקה שלהם.
- עבור  $k$  השכנים הקרובים ביותר, מנו את מספר הוקטורים  $k_i$  השייכים למחלקה  $\omega_i, i = 1, 2, \dots, M$ .
- שייכו את  $\underline{x}$  למחלקת הרוב, כלומר למחלקה עבורה  $k_i$  הוא המקסימלי.

הפונקציה תקבל כקלט את מטריצות האימון והמבחן וכן את וקטורי התגיות המתאימים, את התכונות הנבחרות מתוך המטריצות (לדוגמה- אם רוצים להשתמש רק בשתיים מתוך התכונות), את  $k$  מספר השכנים הדרוש. הפונקציה knn\_classifier תשתמש בפונקציה dist\_neigh לחישוב המרחק האוקלידי בין כל דוגמת מבחן לכל דוגמאות האימון. הפונקציה dist\_neigh

תשתמש בפקודה np.sort כדי לבצע sorting של המרחקים, תבחר את k השכנים הקרובים ביותר ותחזיר את התגיות של k השכנים הקרובים (list), ואת המרחק של כל אחד מהשכנים לדוגמת המבחן (numpy array).

מדד מרחק אוקלידי: אם x הוא וקטור המבחן, ו-y הוא וקטור מקבוצת האימון

$$d = \sqrt{\sum_{i=1}^L (x_i - y_i)^2}$$

אזי המרחק האוקלידי d מוגדר על-ידי: כאשר L הוא מימד

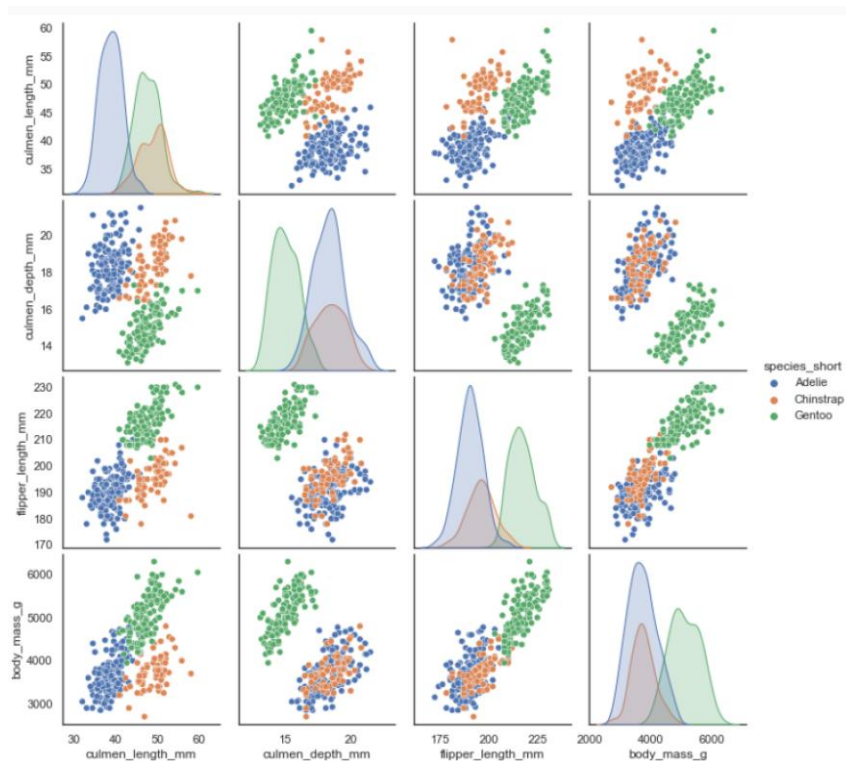
כל וקטור.

הפונקציה knn\_classifier תשתמש ב-np.unique כדי לקבוע מהם התיוגים של k השכנים, וכן את מספר השכנים בעלי אותו תיוג (השייכים לאותה מחלקה), ותשיך את דוגמת המבחן למחלקת הרוב (כלומר למחלקה עבורה מספר השכנים הוא מקסימלי). במידה ויש מספר זהה של שכנים משתי מחלקות או יותר, הפונקציה תחשב את סכום המרחקים של השכנים של כל מחלקה מדוגמת המבחן, ותשיך את דוגמת המבחן למחלקה עבורה סכום המרחקים הוא מינימלי.

ד. הפעילו את הפונקציה שכתבתם על ה-penguin dataset, ובדקו מהו אחוז הזיהוי

הנכון עבור  $k=1, 3, 5$ . הדרכה: נתוני האימון הם שורות X\_train, וקבוצת המבחן היא X\_test, כאשר מתוך שתי המטריצות התכונות הרלבנטיות הן culmen\_length\_mm ו-flipper\_length\_mm.

ה. חזרו על הסעיף הקודם אך עם כל התכונות של ה-penguins dataset.



2. (15 נקודות) וקטורי התכונות הבאים (נקודות ב- $R^2$ ), מייצגים מדדים של שנבוב (aardvark) שלא היה ידוע לחוקרים. הזואולוגים טוענים כי קיימים שני סוגים של שנבוב. כדי להבחין בין הסוגים מפעילים את אלגוריתם k-means עבור  $k=2$ . הצנטרואידים ההתחלתיים הוגרלו בנקודות  $C1=(5,5)$ ,  $C2=(8,2)$ .



- א. סמנו את הצנטרואידים ההתחלתיים על התרשים.
- ב. בצעו את צעד השיוך ההתחלתי, רשמו את הנוסחה באמצעותה מתבצע השיוך כתבו את החישובים שאתם מבצעים. באיזה כלל השתמשתם?
- ג. חשבו את הצנטרואידים של הצעד השני, ורשמו את הנוסחה בה השתמשתם.
3. (35 נקודות) א. ממשו את אלגוריתם ה-k-means. הפונקציה הראשית תיקרא k\_means ותעשה שימוש בשלוש פונקציות python:
- פונקציית איתחול הצנטרואידים – init\_centroid, הפונקציה תבחר באופן אקראי k צנטרואידים. הצנטרואידים ייבחרו מתוך הנקודות של ה-dataset.
  - פונקציית שיוך נקודות הדגימה הנתונות במטריצה X לפי המרחק מהצנטרואידים assign\_samples (זוהי למעשה פונקציה המבצעת 1-NN).
  - פונקציית חישוב הצנטרואידים לכל קבוצת שיוך – centroid\_calc
- הפונקציה K\_means תקבל כקלט נקודות X data, מספר clusters רצוי k, ערך סף מתוך שגיאת ה-clustering לעצירת הריצה כאשר השינוי לא יותר גדול מערך הסף הנ"ל, וכן מספר איטרציות מקסימלי.
- הפונקציה תחזיר וקטור y עם שיוך כל נקודת data ל-cluster, כאשר כל אחד מה-clusters יסומן במספר 1,2,3 וכו' לפי מספר ה-clusters k. כמו כן הפונקציה תחשב ותחזיר את שגיאת ה-clustering.
- ב. הפעילו את האלגוריתם על ה-penguins dataset עבור כל נקודות הנתונים אך רק עבור התכונות culmen\_length\_mm ו-flipper\_length\_mm, וצבעו כל אחת מהנקודות בצירור על-פי הצביר (cluster) אליו היא שייכת. סמנו את הצנטרואידים של כל אחד מהצבירים על-ידי x בצבע המתאים ל-cluster.

חשבו וכתבו כמה פרטים מכל אחד מהמינים שויכו על-ידי אלגוריתם ה- k-means לאותו cluster. הפעילו מספר פעמים את הפונקציה שכתבתם על הנתונים, כאשר בכל פעם בדקו את שגיאת ה-clustering. בחרו את הפתרון המביא לשגיאה מינימלית.  
ג. חזרו על הסעיף הקודם אך עם כל התכונות.

4. (10 נקודות) תהי  $X$  קבוצת הלימוד הבאה:

$$X = \{x_i, i = 1, 2, 3, 4, 5\}$$

$$x_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, x_2 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}, x_3 = \begin{pmatrix} 5 \\ 4 \end{pmatrix}, x_4 = \begin{pmatrix} 6 \\ 5 \end{pmatrix}, x_5 = \begin{pmatrix} 6.5 \\ 6 \end{pmatrix}$$

$$D(x) = \begin{pmatrix} 1 & 1 \\ 2 & 1 \\ 5 & 4 \\ 6 & 5 \\ 6.5 & 6 \end{pmatrix}$$

השתמשו במרחק אוקלידי כדי ליצור את מטריצת המרחק וציירו את הדנדרוגרמה לאחר הפעלת hierarchical clustering עם מרחק מינימלי.

(אפשר לכתוב את כל החישובים באמצעות ה-python, ולהשתמש בפונקציה ...

5. (5 נקודות) בחרו dataset כדוגמת mnist (ספרות בכתב יד, אותיות בכתב יד, iris data, penguins data וכו'). יצרו רשת FCN עם 3 שכבות – שתי שכבות חבויות (hidden) עם 512 ו-256 יחידות בהתאמה, ושכבת פלט מתאימה למספר המחלקות. חלקו את הנתונים לנתוני אימון (עם התגיות המתאימות) ולנתוני מבחן (עם התגיות שלהם). אמנו את הרשת על ה-train dataset (70%-80%) לפי הדוגמה של FCN MNIST 2022.py (ראו בתיקיה ANN במודל). לאחר מכן בחנו את הרשת המאומנת על נתוני המבחן ורשמו את תוצאת אחוז הזיהויים הנכונים (ה-accuracy) עבור נתוני המבחן.