תרגיל בית 3

חלק א' – רטוב (85%):

סט הנתונים שנתון לנו הוא "Leukemia.csv" והוא זמין להורדה במודל. סט הנתונים מתאר מדגם בגודל של 281 דגימות עם 22,284 תכוניות כאשר כל תכונית מייצגת גן (יחידת תורשה), ולכל דוגמה תיוג של אחד מסוגי סרטן הדם (לוקמיה). לפי סט הנתונים יש 7 תגיות של גידולים סרטניים והם:

B-CELL_ALL
B-CELL_ALL_TCF3-PBX1
B-CELL_ALL_HYPERDIP
B-CELL_ALL_HYPO
B-CELL_ALL_MLL
B-CELL_ALL_T-ALL
B-CELL_ALL_ETV6-RUNX1

התגית של כל רשומה נתונה תחת העמודה "type". מספר הדגימה נתון תחת העמודה "samples". הנתונים הותאמו לטובת התרגיל, ולכן עליכם לעבוד אך ורק עם הקובץ הנתון במודל.

למידע נוסף על סט הנתונים, וסטים נוספים דומים לו, ראו:

https://sbcb.inf.ufrgs.br/cumida

תיאור המשימות

בתרגיל בית זה נעסוק בקיבוץ היררכי - Agglomerative clustering. לצורך המשימה נגדיר:

- **מרחק אוקלידי**:

$$d(\vec{x},\vec{y}) = \left|\left|\vec{x}-\vec{y}\right|\right| = \sqrt{\sum_{i=1}^{n}(\vec{x_i}-\vec{y_i})^2}$$

במקרה שלנו, כל דגימה בנתונים היא וקטור במימד 20,531, ולכן מייצגת נקודה כלשהי במימד 20,531 (n=20,531).

- **מדד Silhouette:**

תהיי $x_i \in S$ נקודה בדאטא המשויכת לקלאסטר C_j (שימו לב שבשונה מהגדרת מרחק אוקלידי, כאן x_i הינה נקודה בדאטא, ולא קורדינאטה אחת של נקודה בדאטא), נגדיר:

$$in(x_i) = \frac{1}{|C_j|-1}\sum_{x_k \text{ s.t. } x_k \neq x_i} d(x_i, x_k)$$

זהו המרחק הממוצע של x_i מהנקודות בקלאסטר שאליו היא משויכת, C_j.

$$out(x_i) = \min_{l \neq j} \frac{1}{|C_l|}\sum_{x_k \in C_l} d(x_i, x_k)$$

זהו המרחק המינימלי של הנקודה x_i מקלסטר אליו היא לא משויכת, כאשר המרחק בין x_i לקלסטר C_l הוא המרחק הממוצע בין x_i לנקודות בC_l.

מדד הsilhouette לנקודה x_i הינו:

$$s(x_i) = \frac{out(x_i) - in(x_i)}{\max\left(in(x_i), out(x_i)\right)} \; if \; |C_j| > 1, otherwise \; s(x_i) = 0$$

מדד הsilhouette לקלאסטר C_j הינו:

$$s(C_j) = \frac{1}{|C_j|}\sum_{x_i \in C_j} s(x_i)$$

מדד הsilhouette למדגם S הינו:

$$s(S) = \frac{1}{|S|}\sum_{x_i \in S} s(x_i)$$

- **מדד Rand index**

יהי $S = \{x_1, \dots, x_n\}$ המדגם. נשים לב שישנם $\frac{n*(n-1)}{2} = \binom{n}{2}$ זוגות שונים של נקודות בS. נגדיר:

TP – מספר הזוגות שאלג' האשכול שייך לאותו אשכול, ואכן שייכים לאותו אשכול.

TN – מספר הזוגות שאלג' האשכול שייך לאשכולות שונים, ואכן שייכים לאשכולות שונים.

FP – מספר הזוגות שאלג' האשכול שייך לאותו אשכול, אך הם שייכים לאשכולות שונים.

FN – מספר הזוגות שאלג' האשכול שייך לאשכולות שונים, אם הם שייכים לאותו אשכול.

אזי מדד RI הינו:

$$RI = \frac{TP + TN}{TP + TN + FP + FN} = \frac{TP + TN}{\binom{n}{2}}$$

מדד RI מודד בכמה החלטות האלג' שלנו צדק מתוך סך ההחלטות שעליו לקבל, כלומר, זהו בדיוק מדד הAccuracy.

נרצה לבחון אילו clusters נקבל בכל אחת משתי הגישות שנלמדו בהרצאה: Single Link, ו- Complete Link.

כל אחת מהשיטות תופעל עד קבלה של 7 clusters בלבד.

מטרתכם בחלק הרטוב הינה להפעיל את שתי הגישות, ולכל cluster לציין את 1. הלייבל הדומיננטי בו, 2. מדד הsilhouette שלו. בנוסף, עליכם לציין את מדד הsilhouette למדגם S, ואת מדד הRI למדגם S.

פלט משימה זו צריך להיות מהצורה:

Method:

Cluster Z: [x,…,y], dominant label = value, silhouette = value

Whole data: silhouette = value, RI = value

כאשר:

- Method הינו "single link" או "complete link".
- $x, y, z \in \{1, \dots, 281\}$
- הערך z הינו המספר המזהה של הקלסטר.
- הקלסטרים יסודרו בסדר עולה בערכי z.
- לכל קלסטר, הערכים (x,..y) הינם מספרי הדגימות בקלאסטר, <u>מסודרים בסדר עולה.</u>
- כל הערכים value (למעט הערך של dominant label) הם ברמת דיוק של 3 ספרות אחרי הנקודה. ניתן להשתמש בפו' המובנית round.
- dominant label הינה המחלקה הנפוצה ביותר בקלסטר. במידה ושתי מחלקות נפוצות באותה מידה, המחלקה בעלת השם הקטן יותר (לקסיקוגרפית) תקבל קדימות.

בונוס: חמשת התוכניות עם זמני הריצה הנמוכים ביותר, יזכו את המגישים ב**חמש נקודות בונוס** לציון של תרגיל זה.

דוגמה לפלט שמבוסס על מדגם קטן ושרירותי של תצפיות מתוך סט הנתונים:

```
single link:
Cluster 1: [1, 15, 17, 18, 21, 23, 24, 30, 45, 49, 54, 56, 63, 64, 83, 85, 94, 101, 103, 104, 111, 118, 122, 123, 124, 130, 131, 133, 139, 147, 149, 151, 180, 234, 247, 248, 249, 258], dominant label = B-CELL_ALL, silhouette = 0.111
Cluster 35: [35], dominant label = B-CELL_ALL, silhouette = 0.0
Cluster 41: [41], dominant label = B-CELL_ALL, silhouette = 0.0
Cluster 135: [135], dominant label = B-CELL_ALL_HYPERDIP, silhouette = 0.0
Cluster 184: [184, 195, 201, 205, 209, 216, 221, 223], dominant label = B-CELL_ALL_T-ALL, silhouette = 0.115
Cluster 213: [213], dominant label = B-CELL_ALL_T-ALL, silhouette = 0.0
Cluster 239: [239], dominant label = B-CELL_ALL_ETV6-RUNX1, silhouette = 0.0
Whole data: silhouette = 0.101, RI = 0.549

complete link:
Cluster 1: [1, 15, 18, 21, 24, 45, 49, 54, 56, 63, 64, 83, 85, 101, 103, 104, 111, 118, 123, 147, 151, 180, 234], dominant label = B-CELL_ALL, silhouette = 0.041
Cluster 17: [17, 124, 131, 139, 149, 247, 248, 249, 258], dominant label = B-CELL_ALL_ETV6-RUNX1, silhouette = 0.071
Cluster 23: [23, 94], dominant label = B-CELL_ALL, silhouette = 0.112
Cluster 30: [30, 35, 122, 130, 133], dominant label = B-CELL_ALL_HYPERDIP, silhouette = -0.003
Cluster 41: [41, 239], dominant label = B-CELL_ALL, silhouette = 0.031
Cluster 135: [135], dominant label = B-CELL_ALL_HYPERDIP, silhouette = 0.0
Cluster 184: [184, 195, 201, 205, 209, 213, 216, 221, 223], dominant label = B-CELL_ALL_T-ALL, silhouette = 0.193
Whole data: silhouette = 0.07, RI = 0.725
```

שימו לב: המדגם הנ"ל נתון לכם תחת הקובץ Leukemia_sample.csv.

דרישות למימוש:

המימוש **חייב** להכיל לפחות את המודולים, המחלקות והמתודות הבאות:

- ניתן להוסיף data members למחלקות המתוארות.
- ניתן לשנות טיפוסים של data members.
- ניתן לשנות חתימות של מתודות, **אך לא את ייעודן.**

1. **main.py** – ממשק ראשי לריצת התוכנית, כפי שהיא מתוארת בפרק "תיאור המשימות". בין היתר, בקובץ זה יופיעו השורות:

```
def main(argv):
    pass


if __name__ == '__main__':
    main(sys.argv)
```

כאשר במקום "pass" יופיע קטע הקוד. עליכם לייבא את הספריה sys בראש הקובץ, ע"י import sys. ע"י המבנה הנ"ל תוכלו להריץ את התוכנית שלכם באמצעות:

python **your_path**/main.py arguments

בתרגיל בית זה:

python /home/student/**your_path**/main.py /home/student/**your_path**/Leukemia.csv

שימו לב:

argv[0] = /home/student/**your_path**/main.py,

argv[1] = /home/student/**your_path**/Leukemia.csv

כאשר **your_path** הוא הנתיב בו שמור המודול main.py, שאר המודולים, וקובץ הנתונים Leukemia.csv.

(6%)

2. **data.py** – ממשק לאיסוף הנתונים.

מחלקה Data:

בנאי המחלקה __init__ מקבל כקלט את path (הנתיב המלא של קובץ הנתונים).

Data members:

data - מילון שהkeys שלו הם תכוניות מסט הנתונים, והvalues הם רשימות שמכילים את ערכי התכוניות.

עליכם לבנות את data באותה צורה בדיוק כמו בתרגיל בית 2. לנוחיותכם, קטע הקוד המתאים:

```
df = pandas.read_csv(path)
self.data = df.to_dict(orient="list")
```

לצורך כך יש להשתמש בספרייה pandas, (על ידי Import pandas).

מתודה:

create_samples(self)

פלט המתודה הוא list של אובייקטים מטיפוס המחלקה Sample. בפרט, ה-list מכיל את כל הדגימות שמופיעות תחת העמודה samples בסט הנתונים.

(5%)

3. **sample.py**

מחלקה Sample:

בנאי המחלקה __init__ מקבל כקלט את מספר הדגימה s_id, את ערכי הגנים genes, ו-label.

Data members:

s_id - מספר מזהה של הדגימה מטיפוס Integer. זהו הערך של הרשומה בעמודה samples.

genes – רשימה של ערכי הגנים של הדגימה.

label – המחלקה שהרשומה שייכת אליה, מחרוזת. זהו הערך של הרשומה בעמודה type.

מתודה:

compute_euclidean_distance(self, other)

המתודה מקבלת כקלט את other, שזהו אובייקט מטיפוס המחלקה Sample. תפקיד המתודה הינו לחשב מרחק אוקלידי בין הדגימה, לדגימה other.

(8%)

4. **cluster.py**

מחלקה **Cluster:**

בנאי המחלקה __init__ מקבל כקלט את c_id וsamples.

Data members:

c_id – מספר מזהה של הcluster, מטיפוס Integer.

samples – list אובייקטים מטיפוס Samples. בAgglomerative clustering אנו מבצעים את הclustering בשיטת "bottom-up", ולכן בעת אתחול האלגוריתם, גודלו של samples הינו 1.

מתודות:

merge(self, other)

תפקיד המתודה הוא למזג בכל צעד של האלגוריתם שני clusters לכדי cluster בודד, כאשר נקודות other יוספו לself. המספר המזהה של הcluster החדש יהיה המינימום בין המספרים המזהים של שני הclusters. לדוגמה, אם נמזג את הclusters:

$$self = \{c_{id} = 1, samples = \{x\}\}$$

$$other = \{c_{id} = 2, samples = \{y\}\}$$

ע"י קריאה למתודה מהאובייקט c, נקבל בסוף פעולת המתודה:

$$self = \{c_{id} = 1, samples = \{x, y\}\}$$

יש לוודא שמזהי השדה samples של הקלסטר המאוחד ממוינים בסדר עולה (ניתן להשתמש בפו' מיון מובנות בשפה). לבסוף על הפו' למחוק את *other* ע"י:

```
del other
```

(8%)

print_details(self, silhouette)

פרמטרים:

- silhouette – מדד הsilhouette של הcluster.

תפקיד המתודה הינו להדפיס את נתוני הקלסטר (מזהי הנקודות שהוא מכיל בסדר עולה, dominant label, ומדד silhouette) כפי שמתואר בפרק "תיאור המשימה".

(8%)

5. **link.py** – ממשק לחישוב מרחקים בין שני clusters על פי שתי הגישות: single link and complete link.

מחלקה **Link:**

המחלקה Link מכילה את המתודה האבסטרקטית הבאה:

compute(self, cluster, other)

המחלקות SingleLink ו CompleteLink -צריכות לממש את המתודה האבסטרקטית compute לפי השיטה single link or complete link, בהתאמה.

מחלקה **SingleLink:**

המחלקה SingleLink מכילה את המתודה הבאה:

compute(self, cluster, other)

על המתודה להחזיר את המרחק בין cluster לother בגישת single link.

מחלקה **CompleteLink:**

המחלקה CompleteLink מכילה את המתודה הבאה:

compute(self, cluster, other)

על המתודה להחזיר את המרחק בין cluster לother בגישת complete link.

(15%)

6. agglomerative_clustering.py – ממשק להפעלת אלגוריתם הקיבוץ.

מחלקה **AgglomerativeClustering:**

בנאי המחלקה __init__ מקבל כקלט:

את link שהוא אובייקט מטיפוס SingleLink או CompleteLink,

וsamples שהוא list של אובייקטים מטיפוס המחלקה Sample, המכיל את כל הדגימות בדאטא.

Data members:

link - אובייקט מטיפוס המחלקה SingleLink או CompleteLink.

clusters - list של אובייקטים מטיפוס המחלקה Cluster. בנאי המחלקה יאתחל את האובייקט הנ"ל.

מתודה:

compute_silhoeutte(self)

תפקיד המתודה הינו dictionary שהמפתחות בו הם מזהי (כל) הדגימות בדאטא, והערך המתאים לכל מפתח הוא מדד ההsilhouette של הדגימה הו, כפי שהוגדר בפרק "תיאור המשימה".

(5%)

מתודה:

compute_summery_silhoeutte(self)

תפקיד המתודה הינו להחזיר מילון שמסכם את מדד הsilhouette עבור תוצאת אלג' האשכול. ה-keys של המילון יהיו מזהי הclusters (c_id), ומפתח נוסף: 0.

הvalues של המילון יהיו ערכי הsilhouette של כל קלסטר (עבור הkeys המתאימים), והvalue של המפתח 0 יהיה ערך הsilhouette של כל הדגימות, כפי שהוגדרו בפרק "תיאור המשימה".

(5%)

מתודה:

compute_rand_index(self)

תפקיד המתודה הינו לחשב את ערך הRand Index של האלגוריתם כפי שהוגדר בפרק "תיאור המשימה".

(5%)

מתודה:

run(self, max_clusters)

תפקיד המתודה הוא להריץ את אלגוריתם האשכול, כך שבסיום ריצת האלגוריתם, מספר הclusters לא יעלה על max_clusters. פלט המתודה הינו הדפסה של תוצאת אלגוריתם הקיבוץ כפי שמתואר בפרק "תיאור המשימה".

(20%)

הערה

לצורך המימוש במודול זה בלבד, ניתן לייבא (אך לא חובה) את הספריות הבאות, באופן הבא:

```
import math
from collections import OrderedDict
```

כמו כן ניתן להשתמש בפונקציות ומחלקות שמימשתם בתרגילי בית קודמים.

חלק ב' – יבש (15%):

1. פו' המטרה של אלגוריתם **k-means** הינה פו' הSSE:

$$G_j = \sum_{x \in C_j} d(\mu_j, x)^2 = \sum_{x \in C_j} \left|\left|x - \mu_j\right|\right|^2$$

$$SSE = \sum_{i=1}^{k} G_i$$

האם ייתכן שהSSE יהיה שווה לאפס? הסבירו.

(5%)

2. אחזור אד-הוקי הינה משימה בה משתמש מציג למנוע חיפוש שאילתא המבטאת את הצורך של המשתמש במידע, ועל מנוע החיפוש לאחזר (להציג למשתמש) מסמכים הרלוונטים לצורך במידע של המשתמש, כאשר לרשות מנוע החיפוש עומד מאגר מסמכים C.

נניח שנתונים לנו חיווי רלוונטיות לצורך במידע של המשתמש (שאילתא) ביחס לכל המסמכים במאגר: כלומר, לכל מסמך במאגר C נשייך את הלייבל 0 אם המסמך אינו רלוונטי לצורך במידע של המשתמש, ואת הלייבל 1 אם הוא אכן רלוונטי לצורך במידע של המשתמש.

נתונה שאילתא q, ונתון מנוע החיפוש S, שבתגובה לכל שאילתא אפשרית, מאחזר 0 מסמכים מתוך המאגר C. נרצה לשערך את איכות מנוע החיפוש S.

2.1. הציגו דוגמה לחיווי רלוונטיות לq ביחס למאגר, שבה מנוע החיפוש S יקבל ערך *accuracy* השווה ל1.

(5%)

2.2. מדוע ערך ה*accuracy* מטעה בדוגמה שהצגתם? **רמז**: התייחסו לprecision ולrecall של המנוע S בעבור הדוגמה שהצגתם.

(5%)

דגשים נוספים:

1. עליכם לכתוב את הקוד בהתאם לדגשים והסטנדרטים לפי pep8. לשימושכם המסמך Code Quality " Requirements " באתר ה moodle- של הקורס. קוד אשר לא יעמוד בסטנדרטים הנדרשים, יקבל ניקוד מופחת.
2. ניתן להוסיף מתודות נוספות, במידה ותמצאו לנכון. יש להימנע מכפילויות קוד.
3. ניתן להשתמש במתודות שהן in-built בשפה. קרי, מתודות אשר **לא** דורשות ייבוא של ספריות.
4. יש לתת שמות בעלי משמעות לכל משתנה.
5. חובה לתעד את הקוד באנגלית. בפרט עליכם לכתוב עבור כל מתודה docstring

הוראות הגשה:

- התרגיל להגשה בזוגות בלבד.
- לפני ההגשה, חובה לוודא שהתוכנית עובדת במעבדת ההוראה ולא בסביבה אחרת.

- ההגשה חייבת להכיל קובץ אחד (קובץ zip) :
  - שם הקובץ חייב להיות hw3 _xxxxxxxx_yyyyyyyy.zip כאשר xxxxxxxx ו-yyyyyyyy הם מספרי תעודות הזהות של המגישים, כולל ספרת ביקורת.
  - הקובץ מכיל את כל קבצי הקוד. אין להכיל תיקייה ובתוכה קבצי הקוד, אלא את קבצי הקוד עצמם.
  - **הערה :** עליכם לוודא שהתוכנית מתחילה לפעול מקובץ "main.py" בלבד.
  - תשובות לחלקים יבשים יש להקליד במעבד תמלילים. אין להגיש תשובות בכתב יד.

- ההגשה היא אלקטרונית בלבד, דרך אתר ה-moodle של הקורס. תרגילים שיוגשו בכל דרך אחרת לא ייבדקו.
- אין להגיש את אותו הקובץ פעמיים. התרגיל יוגש ע"י אחד מבני הזוג.
- שימו לב שההגשה תיחסם בדיוק בשעה 23:55 ביום ההגשה. מומלץ להגיש לפחות שעה לפני המועד האחרון.
- ניתן להגיש כמה פעמים. רק ההגשה האחרונה תישמר.
- תרגיל בית שלא יוגש לפי הוראות ההגשה – לא ייבדק (כלומר יקבל ציון 0).
- לצורך תרגיל הבית ייפתח פורום. ניהול שאלות ומתן תשובות בנושא התרגיל יתבצע דרך הפורום בלבד.

בהצלחה!