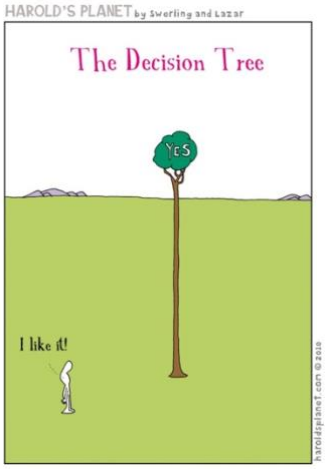
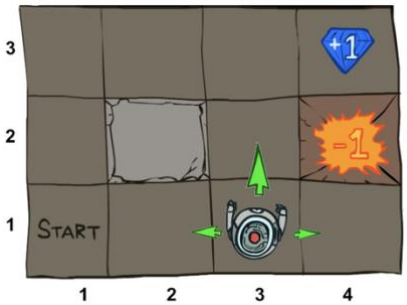
תרגיל בית 3 – MDP ומבוא ללמידה

הנחיות כלליות:

- תאריך ההגשה: 30/07/22 ב23:59
- את המטלה יש להגיש **בזוגות בלבד.**
- יש להגיש מטלות מוקלדות בלבד. פתרונות בכתב יד לא ייבדקו.
- התשובות צריכות להיות כתובות בשפה העברית. ניתן לבקש אישור פרטני להגשה באנגלית מהמתרגל האחראי על הקורס.
- ניתן לשלוח שאלות בנוגע לתרגיל בפיאצה בלבד.
- המתרגל האחראית על תרגיל זה: **רועי עוגן.**
- בקשות דחיה מוצדקות (מילואים, אשפוז וכו') יש לשלוח למתרגל האחראי (ספיר טובול) בלבד.
- במהלך התרגיל ייתכן שנעלה עדכונים, למסמך הנ"ל – תפורסם הודעה בהתאם.
- העדכונים הינם מחייבים, ועליכם להתעדכן עד מועד הגשת התרגיל.
- שימו לב, התרגיל מהווה כ- 10% מהציון הסופי במקצוע ולכן העתקות תטופלנה בחומרה.
- התשובות לסעיפים בהם מופיע הסימון ✍ צריכים להופיע בדוח.
- לחלק הרטוב מסופק שלד של הקוד
- אנחנו קשובים לפניות שלכם במהלך התרגיל ומעדכנים את המסמך הזה בהתאם. גרסאות עדכניות של המסמך יועלו לאתר. הבהרות ועדכונים שנוספים אחרי הפרסום הראשוני יסומנו כאן בצהוב. בנוסף, לכל עדכון יהיה מספר גרסה כדי שתוכלו לעקוב. ייתכן שתפורסמנה גרסאות רבות – אל תיבהלו מכך. השינויים בכל גרסה יכולים להיות קטנים.

https://emojipedia.org/apple/ios-14.6/writing-hand/





מומלץ לחזור על שקפי ההרצאות והתרגולים הרלוונטיים לפני תחילת העבודה על התרגיל.

חלק א' – MDP (30 נק')

רקע

בחלק זה נעסוק בתהליכי החלטה מרקובים, נתעניין בתהליך עם **אופק אינסופי** (מדיניות סטציונרית) ובתועלת המחושבת בעזרת-Discounted rewards.

חלק א' - חלק היבש ✍️

1. בתרגול ראינו את משוואת בלמן כאשר התגמול ניתן עבור המצב הנוכחי בלבד, כלומר $R: S \to \mathbb{R}$, למתן תגמול זה נקרא "תגמול על הצמתים" מכיוון שהוא תלוי בצומת שהסוכן נמצא בו.
בהתאם להגדרה זו הצגנו בתרגול את האלגוריתמים Value iteration ו-Policy Iteration למציאת המדיניות האופטימלית.

כעת, נרחיב את ההגדרה הזו, לתגמול המקבל את המצב הנוכחי, הפעולה לביצוע והמצב הבא שהסוכן הגיע אליו בפועל (בין אם הסוכן בחר לצעוד לכיוון הזה ובין אם לא), כלומר: $R: S \times A \times S' \to \mathbb{R}$, למתן תגמול זה נקרא "תגמול על הקשתות".

א. (יבש 1 נק') התאימו את הנוסחה של התוחלת של התועלת מהתרגול, עבור התוחלת של התועלת המתקבלת במקרה של "תגמול על הקשתות"?

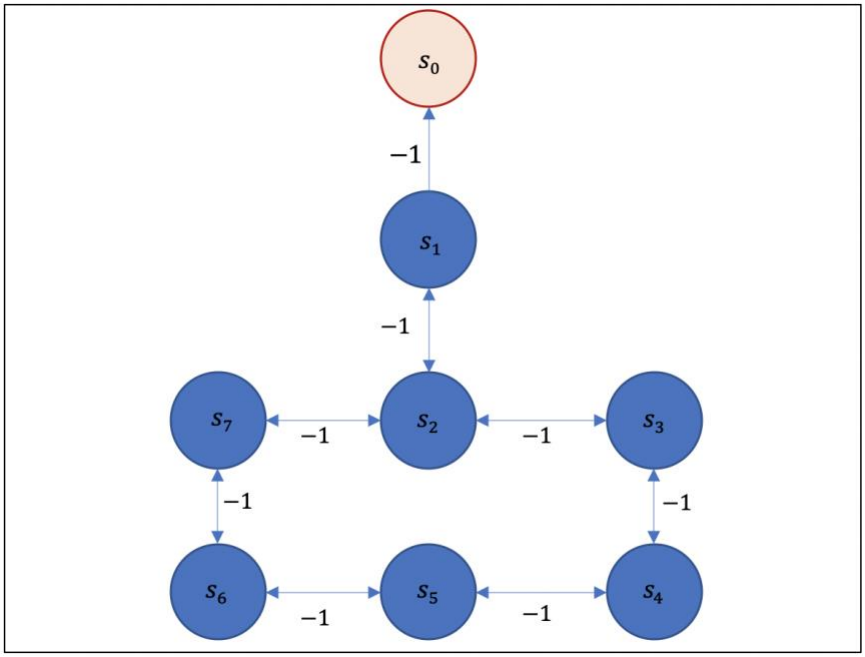
ב. (יבש 1 נק') כתבו מחדש את נוסחת משוואת בלמן עבור המקרה של "תגמול על הקשתות".

ג. (יבש 2 נק') נסחו את אלגוריתם Value Iteration עבור המקרה של "תגמול על הקשתות".

ד. (יבש 2 נק') נסחו את אלגוריתם Policy Iteration עבור המקרה של "תגמול על הקשתות".

הערה: בסעיפים 3 ו-4 התייחסו גם למקרה בו $\gamma = 1$, והסבירו מה לדעתכם התנאים שצריכים להתקיים על הסביבה\mdp על מנת שתמיד נצליח למצוא את המדיניות האופטימלית.

נתון הגרף הבא:



נתונים:

- $\gamma = 1$ (Discount factor).
- אופק אינסופי.
- $S = \{s_0, s_1, s_2, s_3, s_4, s_5, s_6, s_7\}$ – קבוצת המצבים – מתארים את מיקום הסוכן בגרף.
- $S_G = \{s_0\}$ – קבוצת המצבים הסופיים.
- קבוצת הפעולות לכל מצב (על פי הגרף), לדוגמא: $A(s_2) = \{\uparrow, \rightarrow, \leftarrow\}$.
- תגמולים ("תגמול על הקשתות"):

$$\forall s \in S \backslash S_G, a \in A(s), s' \in S: \quad R(s, a, s') = -1$$

- מודל המעבר הוא דטרמיניסטי, כלומר כל פעולה מצליחה בהסתברות אחת.

ה. (יבש 4 נק') הריצו את האלגוריתם Value iteration שכתבתם על הגרף הנתון. ומלאו את הערכים בטבלה הבאה, כאשר $\forall s \in S: U_0(s) = 0$. (ייתכן שלא צריך למלא את כולה).

	$U_0(s_i)$	$U_1(s_i)$	$U_2(s_i)$	$U_3(s_i)$	$U_4(s_i)$	$U_5(s_i)$	$U_6(s_i)$	$U_7(s_i)$	$U_8(s_i)$
s_1	0								
s_2	0								
s_3	0								
s_4	0								
s_5	0								
s_6	0								
s_7	0								

ו. (יבש 4 נק') הריצו את האלגוריתם Policy iteration שכתבתם על הגרף הנתון. ומלאו את הערכים בטבלה הבאה, כאשר המדיניות ההתחלתית π_0 מופיעה בעמודה הראשונה בטבלה. (ייתכן שלא צריך למלא את כולה).

	$\pi_0(s_i)$	$\pi_1(s_i)$	$\pi_2(s_i)$	$\pi_3(s_i)$	$\pi_4(s_i)$	$\pi_5(s_i)$	$\pi_6(s_i)$	$\pi_7(s_i)$	$\pi_8(s_i)$
s_1	$\uparrow$								
s_2	$\uparrow$								
s_3	$\leftarrow$								
s_4	$\uparrow$								
s_5	$\rightarrow$								
s_6	$\rightarrow$								
s_7	$\downarrow$								

חלק ב' - היכרות עם הקוד

mdp.py – **אתם לא צריכים לערוך כלל את הקובץ הזה.**

בקובץ זה ממומשת הסביבה של ה-mdp בתוך מחלקת MDP. הקונסטרקטור מקבל:

- board - המגדיר את המצבים האפשריים במרחב ואת התגמול לכל מצב, תגמול על הצמתים בלבד.
- terminal_states – קבוצה של המצבים הסופיים (בהכרח יש לפחות מצב אחד סופי).
- transition_function – מודל המעבר בהינתן פעולה, מה ההסתברות לכל אחת מארבע הפעולות האחרות. ההסתברויות מסודרות לפי סדר הפעולות.
- gamma – discount factor המקבל ערכים - $\gamma \in (0,1]$.

הערה: קבוצת הפעולות מוגדרת בקונסטרקטור והיא קבועה לכל לוח שיבחר.

למחלקת MDP יש מספר פונקציות שעשויות לשמש אתכם בתרגיל.

- print_rewards() – מדפיסה את הלוח עם ערך התגמול בכל מצב.
- print_utility(U) – מדפיסה את הלוח עם ערך התועלת U לכל מצב.
- print_policy(policy) – מדפיסה את הלוח עם הפעולה שהמדיניות policy נתנה לכל מצב שהוא לא מצב סופי.
- step(state, action) – בהינתן מצב נוכחי state ופעולה action מחזיר את המצב הבא באופן דטרמיניסטי. עבור הליכה לכיוון קיר או יציאה מהלוח הפונקציה תחזיר את המצב הנוכחי state.

חלק ג' – רטוב

בקובץ value_and_policy_iteration.py

מותר להשתמש בספריות:

All the built in packages in python, numpy, matplotlib, argparse, os, copy, typing, termcolor

עליכם לממש את הפונקציות הבאות:

- (רטוב 4 נק'): value_iteration(mdp, U_init, epsilon) – בהינתן ה-mdp, ערך התועלת ההתחלתי U_init, וחסם העליון לשגיאה מהתוחלת של התועלת האופטימלי epsilon מריץ את האלגוריתם value iteration ומחזיר את U המתקבל בסוף ריצת האלגוריתם. TODO

- (רטוב 4 נק'): get_policy(mdp, U) – בהינתן ה-mdp וערך התועלת U (המקיים את משוואת בלמן) מחזיר את המדיניות (במידה וקיימת יותר מאחת, מחזיר אחת מהן). TODO

- (רטוב 4 נק'): policy_evaluation(mdp, policy) – בהינתן ה-mdp, ומדיניות policy מחזיר את ערכי התועלת לכל מצב. TODO

- (רטוב 4 נק'): policy_iteration(mdp, policy_init) - בהינתן ה-mdp, ומדיניות התחלתית policy_init, מריץ את האלגוריתם policy iteration ומחזיר מדיניות אופטימלית. TODO

main.py – דוגמת הרצה לשימוש בכל הפונקציות. עליכם להריץ את שורת הפקודה הבאה בטרמינל מהתיקייה:

python main.py board terminal_states transition_function

בתחילת הקובץ אנו טוענים את הסביבה משלושה קבצים (board, terminal_states, transition_function) ויוצרים מופע של הסביבה (mdp).

- שימו לב, שכרגע הקוד ב-main לא יכול לרוץ מכיוון שאתם צריכים להשלים את הפונקציות הרלוונטיות ב-value_and_policy_iteration.py.
- בנוסף, על מנת לראות את הלוח עם הצבעים עליכם להריץ את הקוד בIDE לדוגמה PyCharm.

הערות לרטוב:

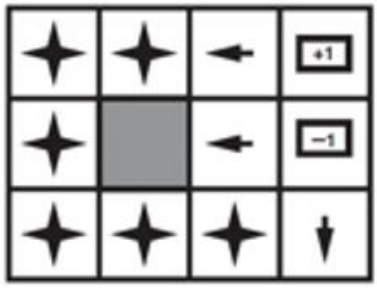
- עבור $\gamma = 1$ סגל הקורס יריץ רק סביבות בהן התועלת של כל מצב תתכנס למספר סופי.
- בסוף הקובץ יש נספח ובו דוגמת הרצה.

בונוס (6 נקודות לציון התרגיל):

bonus.py:

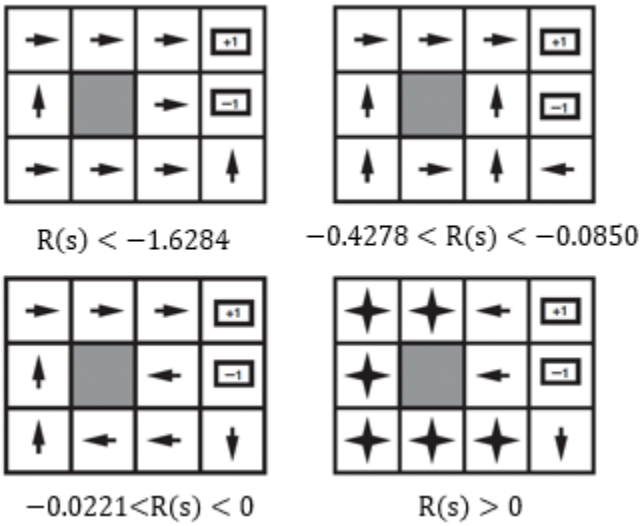
עליכם להשלים את הפונקציות הבאות:

- get_all_policies(mdp, U, ...) – בהינתן ה-mdp, וערך התועלת U (המקיים את משוואת בלמן) מדפיס\מציג את **כל** המדיניות המקיימות ערך זה בלוח בודד (יש לבצע ויזואליזציה להצגת כל המדיניות), לדוגמא:



הפונקציה מחזירה את מספר המדיניות (policies) השונות הקיימות המקיימות את U.TODO

- get_policy_for_different_rewards(mdp, ...) – בהינתן ה-mdp מדפיס\מציג את המדיניות האופטימלית כתלות ב-r (ערכי התגמול לכל מצב שאינו סופי). TODO דוגמא חלקית של פתרון אפשרי:



✍️ בנוסף לקוד עליכם לצרף להגשה היבשה את התצוגות של הפונקציות על הסביבה שניתנה בתרגיל, לציין באילו ספריות השתמשתם, להסביר מהם הקלטיים לפונקציה שבחרתם להוסיף ולמה.

הערה: רק בקובץ bonus.py ניתן לבצע Import לספריות נוספות (לדוגמא pygame) ולהוסיף קלטים לפונקציות.

חלק ב' - מבוא ללמידה (70 נק')

רקע

בחלק של הלמידה, נעזר ב $data-set$, הדאטה חולק עבורכם לשתי קבוצות: קבוצת אימון train.csv וקבוצת מבחן test.csv. ככלל, קבוצת האימון תשמש אותנו לבניית המסווגים, וקבוצת המבחן תשמש להערכת ביצועיהם.

בקובץ utils.py תוכלו למצוא את הפונקציות הבאות לשימושכם:
load_data_set, create_train_validation_split, get_dataset_split בקובץ utils.py אשר טוענות/מחלקת את הדאטה למערכים np.array בצורה נוחה (קראו את תיעוד הפונקציות).

הדאטה של ID3 עבור התרגיל מכיל מדדים שנאספו מצילומים שנועדו להבחין בין גידול שפיר לגידול ממאיר. כל דוגמה מכילה 30 מדדים כאלה, ותווית בינארית **diagnosis** הקובעת את סוג הגידול (0=שפיר, 1=ממאיר). כל התכונות (מדדים) רציפות . העמודה הראשונה מציינת האם האדם חולה (M) או בריא (B). שאר העמודות מציינות כל תכונות רפואיות שונות של אותו אדם (התכונות מורכבות ואינכם צריכים להתייחס למשמעות שלהן כלל).

חלק א' - היכרות עם הקוד

<u>תיקית $ID3-dataset$:</u>

- תיקיה זו אלו מכילה את קבצי הנתונים עבור $ID3$.

<u>קובץ $utils.py$:</u>

- קובץ זה מכיל פונקציות עזר שימושיות לאורך התרגיל, כמו טעינה של $dataset$ וחישוב הדיוק.
- עליכם לממש את הפונקציות $l2\_dist$ ו $accuracy$. קראו את תיעוד הפונקציות ואת ההערות הנמצאות תחת התיאור.

<u>קובץ $unit\_test.py$:</u>

- קובץ בדיקה בסיסי שיכול לעזור לכם לבדוק את המימוש.

<u>קובץ $DecisionTree.py$:</u>

- קובץ זה מכיל 3 מחלקות שימושית לבניית עץ $ID3$ שלנו.
  - <u>המחלקה $Question$:</u> מחלקה זו מממשת הסתעפות של צומת בעץ. היא שומרת את התכונה ואת הערך שלפיהם מפצלים את הדאטה שלנו.
  - <u>המחלקה $DecisionNode$:</u> מחלקה זו מממשת צומת בעץ ההחלטה. הצומת מכיל שאלה $Question$ ואת שני הבנים $true\_branch, false\_branch$ כאשר $true\_branch$ הם הענף חלק של הדאטה שעונה True על שאלת הצומת (הפונקצייה $match$ של ה$Question$ מחזירה $True$) ו- $false\_branch$ באופן דומה.
  - <u>המחלקה $Leaf$:</u> מחלקה זו מממשת צומת שהוא עלה בעץ ההחלטה. העלה מכיל לכל אחד מהמחלקות בדאטה את מספר הדוגמאות בעלה עבור כל מחלקה (למשל: $\{'B' : 5, 'M' : 6\}$).

<u>קובץ $ID3.py$:</u>

- קובץ זה מכיל את המחלקה של $ID3$, עיינו בהערות ותיעוד המתודות.

<u>קובץ $ID3\_experiments.py$:</u>

- קובץ הרצת הניסויים של ID3, בקובץ זה יש את הניסויים (שנסביר עליהם בהמשך):
  cross_validation_experiment, basic_experiment

✍️ חלק ב' – חלק היבש (28 נק')

1. (8 נק') השתמשו בdataset הנתון על מנת ללמוד מסווג (עץ) אשר יחזה האם התלמדים יעברו את הקורס באלגברה א'

Average	Studied	Passed
Low	No	No
Low	Yes	Yes
Medium	No	No
Medium	Yes	Yes
High	No	Yes
High	Yes	Yes

הערה, ניתן לרשום את התשובות בעזרת $\log_2$

א. מה האנטרופיה H(Passed)?
ב. מה האנטרופיה H(Passed | Average)?
ג. מה האנטרופיה H(Passed | Studied) ?
ד. צייר את עץ ההחלטה הנלמד עבור בdataset הנתון

2. (12 נק') נגדיר דאטה סט $D = \{(x_1, y_1), \dots, (x_n, y_n)\}$ שבו n דוגמאות מתויגות עם סיווג בינארי $y_i \in \{0,1\}$. כל דוגמה היא וקטור תכונות המורכב משתי תכונות רציפות $x_i = (v_{1,i}, v_{2,i})$.
הניחו כי קיים מסווג מטרה $f(x): R^2 \to \{0,1\}$ שאותו אנו מעוניינים ללמוד (הוא אינו ידוע לנו) וכן שהדוגמאות ב-D עקביות עם מסווג המטרה (כלומר שאין דוגמאות רועשות ב- D).
בסעיפים הבאים, עבור KNN, הניחו פונק' מרחק אוקלידי.
כמו כן, הניחו שאם קיימות נקודות במרחב כך שעבורן יש מספר דוגמאות במרחק זהה, קודם מתחשבים בדוגמאות עם ערך v_1 מקסימלי ובמקרה של שוויון בערך של v_1, מתחשבים קודם בדוגמאות עם ערך v_2 מקסימלי. הניחו כי אין דוגמאות זהות לחלוטין (כלומר גם עם ערך v_1 זהה וגם עם ערך v_2 זהה).
בכל סעיף, **הציגו מקרה המקיים את התנאים המוצגים בסעיף, הסבירו במילים, וצרפו תיאור גרפי (ציור) המתאר את המקרה (הכולל לפחות תיאור מסווג המטרה והדוגמאות שבחרתם)** . סמנו דוגמאות חיוביות בסימן '+' (פלוס) ודוגמאות שליליות בסימן '-' (מינוס). בכל אחת מתתי הסעיפים הבאים אסור להציג מסווג מטרה טריוויאלי, דהיינו שמסווג כל הדוגמאות כחיוביים או כל הדוגמאות כשליליים.

[3 שורות לכל סעיף, אין הגבלה על הגרפים, מלל ופתרון שאינו מוגדר היטב כמתבקש לא יקבל ניקוד]

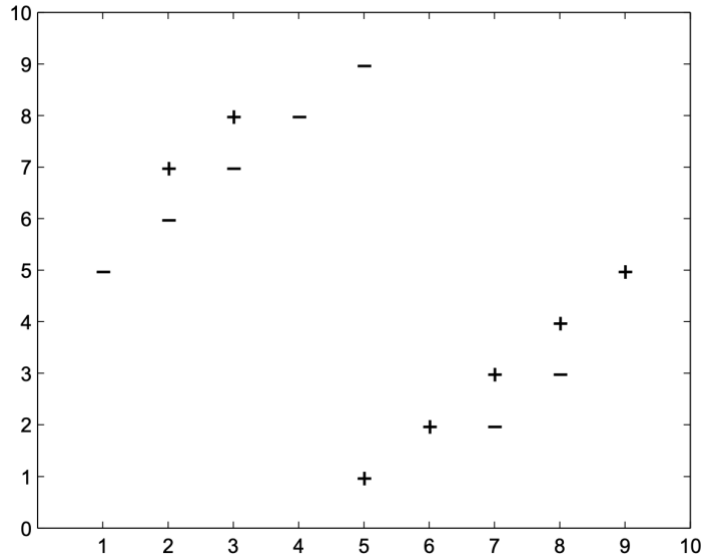
א. (3 נק') הציגו מסווג מטרה $f(x): R^2 \to \{0,1\}$ וקבוצת אימון בעלת לכל היותר 10 דוגמאות כך שלמידת עץ ID3 תניב מסווג אשר עונה נכון עבור כל דוגמת מבחן אפשרית (כלומר יתקבל מסווג המטרה), אך למידת KNN תניב מסווג שעבורו קיימת לפחות דוגמת מבחן אחת עליה הוא יטעה, לכל ערך K שייבחר.

ב. (3 נק') הציגו מסווג מטרה $f(x): R^2 \to \{0,1\}$ וקבוצת אימון בעלת לכל היותר 10 דוגמאות כך שלמידת מסווג KNN עבור ערך K מסוים תניב מסווג אשר עונה נכון עבור כל דוגמת מבחן אפשרית (כלומר יתקבל מסווג המטרה), אך למידת עץ ID3 תניב מסווג אשר עבורו קיימת לפחות דוגמת מבחן אפשרית אחת עליה הוא יטעה.

ג. (3 נק') הציגו מסווג מטרה $f(x): R^2 \to \{0,1\}$ וקבוצת אימון בעלת לכל היותר 10 דוגמאות כך שלמידת מסווג KNN עבור ערך K מסוים תניב מסווג אשר עבורו קיימת לפחות דוגמת מבחן אפשרית אחת עליה הוא יטעה, וגם למידת עץ ID3 תניב מסווג אשר עבורו קיימת לפחות דוגמת מבחן אחת אפשרית עליה הוא יטעה.

ד. (3 נק') הציגו מסווג מטרה $f(x): R^2 \to \{0,1\}$ וקבוצת אימון בעלת לכל היותר 10 דוגמאות כך שלמידת מסווג KNN עבור ערך K מסוים תניב מסווג אשר עונה נכון עבור כל דוגמת מבחן אפשרית (כלומר יתקבל מסווג המטרה), וגם למידת עץ ID3 תניב מסווג עונה נכון עבור כל דוגמת מבחן אפשרית (כלומר יתקבל מסווג המטרה).

3. (8 נק') בשאלה נשתמש במסווג k-nearest neighbour באמצעות מרחק אוקלידי, במשימת סיווג בינארי. אנו מגדירים את הסיווג של נקודת המבחן להיות הסיווג של רוב ה-K השכנים הקרובים ביותר (שימו לב שנקודה יכולה להיות שכנה של עצמה).



א. (2 נק') איזה ערך של k ממזער את שגיאת האימון עבור קב' הדגימות הנ"ל? מהי שגיאת האימון כתוצאה מכך?

ב. (2 נק') מדוע שימוש בערכי k גדולים מדי יכול להיות גרוע עבור קב' הדגימות הנ"ל? למה אולי גם ערכים קטנים של k זה רעיון רע?

ג. (2 נק') איזה ערך של k ממזער את שגיאת Leave-One-Out Cross Validation עבור קב' הדגימות? מהי השגיאה שנוצרה?
https://www.statology.org/leave-one-out-cross-validation/

ד. (2 נק') שרטט את גבול ההחלטה של 1-nearest neighbor עבור הגרף.

חלק ג' – חלק רטוב ID3 (42 נק')

מותר להשתמש בספריות:

All the built in packages in python, sklearn, pandas ,numpy, random, matplotlib, argparse, abc, typing,

אך כמובן שאין להשתמש באלגוריתמי הלמידה, או בכל אלגוריתם או מבנה נתונים אחר המהווה חלק מאלגוריתם למידה אותו תתבקשו לממש.

4. (5 נק') השלימו את הקובץ utils.py ע"י מימוש הפונקציות $l2\_dist$ ו $accuracy$. קראו את תיעוד הפונקציות ואת ההערות הנמצאות תחת התיאור TODO.
(הריצו את הטסטים המתאימים בקובץ $unit\_test.py$ לוודא שהמימוש שלכם נכון)

5. (25 נק') **אלגוריתם ID3:**

a. ממשו את אלגוריתם $ID3$ כפי שנלמד בהרצאה. TODO
שימו לב שכל התכונות רציפות. אתם מתבקשים להשתמש בשיטה של חלוקה דינמית המתוארת בהרצאה. כאשר בוחנים ערך סף לפיצול של תכונה רציפה, דוגמאות עם ערך השווה לערך הסף משתייכות לקבוצה עם הערכים הגדולים מערך הסף. במקרה שיש כמה תכונות אופטימליות בצומת מסוים בחרו את התכונה בעלת האינדקס המקסימלי.
המימוש צריך להופיע בקובץ בשם $ID3.py$ (השלימו את הקוד החסר אחרי שעיינתם והפנמתם את הקובץ $DecisionTree.py$ ואת המחלקות שהוא מכיל).

b. ממשו $basic\_experiment$ שנמצאת ב $ID3\_experiments.py$ TODO
והריצו את החלק המתאים ב $main$ ציינו בדו"ח את הדיוק שקיבלתם. ✍️

6. (12 נק') **גיזום מוקדם.**

פיצול צומת מתקיים כל עוד יש בו יותר דוגמאות מחסם המינימום m, כלומר בתהליך בניית העץ מבוצע "גיזום מוקדם" כפי שלמדתם בהרצאות. שימו לב כי פירוש הדבר הינו שהעצים הנלמדים אינם בהכרח עקביים עם הדוגמאות .לאחר סיום הלמידה (של עץ יחיד), הסיווג של אובייקט חדש באמצעות העץ שנלמד מתבצע לפי רוב הדוגמאות בעלה המתאים.

✍️ **a.** הסבירו מה החשיבות של הגיזום באופן כללי ואיזה תופעה הוא מנסה למנוע?
[אורך התשובה מוגבל ל 3 שורות]

b. ממשו את הגיזום המוקדם כפי שהוגדר בהרצאה. הפרמטר M מציין את מספר המינימלי בעלה לקבלת החלטה. על המימוש של הגיזום המוקדם להיות גם כן בתוך המחלקה ID3 שנמצאת בקובץ $ID3.py$ TODO

c. **סעיף זה בונוס (4 נקודה לציון התרגיל):** בצעו כיוונון לפרמטר M על קבוצת האימון:
1. בחרו לפחות חמישה ערכים שונים לפרמטר M.
2. עבור כל ערך, חשבו את הדיוק של האלגוריתם על ידי $K-fold$ cross validation על קבוצת האימון בלבד. כדי לבצע את חלוקת קבוצת האימון ל- K קבוצות יש להשתמש בפונקציה sklearn.model_selection.KFold עם הפרמטרים $n\_split = 5$, $shuffle = True$ ו- random_state שווה למספר תעודת הזהות שלכם. (כל כיוונון פרמטרים בתרגיל יעשה בצורה דומה).

✍️ i. https://emojipedia.org/apple/ios-14.6/writing-hand/ השתמשו בתוצאות שקיבלתם כדי ליצור גרף המציג את השפעת הפרמטר M על הדיוק. צרפו את הגרף בדו״ח. (לשימושכם הפונקציה $util\_plot\_graph$ בתוך הקובץ $utils.py$).

✍️ ii. הסבירו את הגרף שקיבלתם. לאיזה גיזום קיבלתם התוצאה הטובה ביותר ומהי תוצאה זו?

כעת תצטרכו לממש כיוונון הפרמטר נמצא בפונקציה בשם $cross\_validation\_experiment$ (קראו את התיעוד והשלימו אותה) בקובץ $ID3\_experiments.py$.

d. השתמשו באלגוריתם ID3 עם הגיזום המוקדם כדי ללמוד מסווג מתוך **כל** קבוצת האימון ולבצע חיזוי על קבוצת המבחן. השתמשו בערך ה- M האופטימלי שמצאתם בסעיף c. (ממשו $best\_m\_test$ שנמצאת ב $ID3\_experiments.py$ והריצו את החלק המתאים ב $main$) ציינו בדו"ח את הדיוק שקיבלתם. האם הגיזום שיפר את הביצועים ביחס להרצה ללא גיזום בשאלה 5?

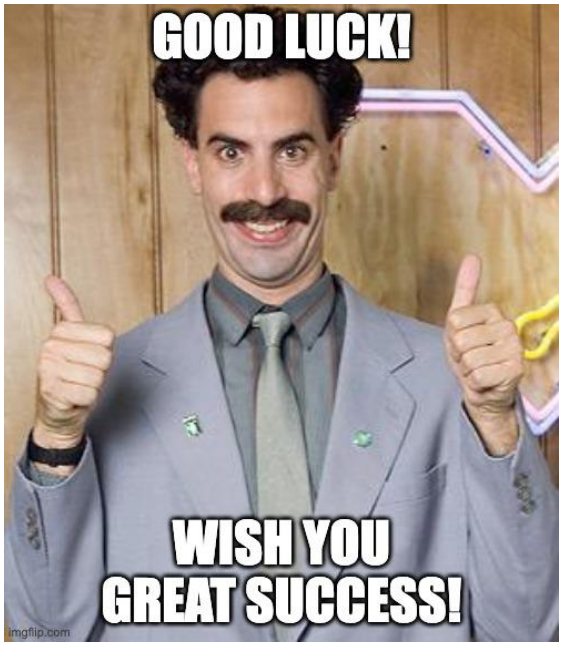
<u>הערה:</u> בסעיף זה אם לא מימשתם את סעיף c השתמשו בערך $M = 50$.

הוראות הגשה

✓הגשת התרגיל תתבצע אלקטרונית בזוגות בלבד.
✓הקוד שלכם ייבדק (גם) באופן אוטומטי ולכן יש להקפיד על הפורמט המבוקש. הגשה שלא עומדת בפורמט תקבל קנס.
✓המצאת נתונים לצורך בניית הגרפים אסורה ומהווה עבירת משמעת.
✓הקפידו על קוד קריא ומתועד. התשובות בדוח צריכות להופיע לפי הסדר.
✓יש להגיש קובץ zip יחיד בשם AI3_<id1>_<id2>.zip (ללא סוגריים משולשים) שמכיל:

✓ קובץ בשם AI_HW3.PDF המכיל את תשובותיכם לשאלות היבשות.
✓ כל קבצי הקוד שנדרשתם לממש בתרגיל:

- בחלק של עצי החלטה – utils.py,ID3.py, ID3_experiments.py
- בחלק של mdp - value_and_policy_iteration.py , bonus.py (רק מי שעשה את הבונוס)
- כל קוד עזר שמימשתם בתרגיל.



נספח MDP:

דוגמת הרצה

יצירת הסביבה:

```
mdp = MDP(board=board_env,
          terminal_states=terminal_states_env,
          transition_function=transition_function_env,
          gamma=1.0)
```

הדפסת הלוח עם התגמולים לכל מצב:

```
print('@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@')
print("@@@@@@ The board and rewards @@@@@@")
print('@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@')
mdp.print_rewards()
```

פלט:

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@ The board and rewards @@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
| -0.04 | -0.04 | -0.04 | +1    |
| -0.04 | WALL  | -0.04 | -1    |
| -0.04 | -0.04 | -0.04 | -0.04 |
```

Value iteration:

```
print('@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@')
print("@@@@@@@@@ Value iteration @@@@@@@@@")
print('@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@')

U = [[0, 0, 0, 0],
     [0, 0, 0, 0],
     [0, 0, 0, 0]]
print("\nInitial utility:")
mdp.print_utility(U)
print("\nFinal utility:")
U_new = value_iteration(mdp, U)
mdp.print_utility(U_new)
print("\nFinal policy:")
policy = get_policy(mdp, U_new)
mdp.print_policy(policy)
```

פלט:

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@ Value iteration @@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

Initial utility:
| 0.0   | 0.0   | 0.0   | 0.0   |
| 0.0   | WALL  | 0.0   | 0.0   |
| 0.0   | 0.0   | 0.0   | 0.0   |


Final utility:
| 0.812 | 0.868 | 0.918 | 1.0   |
| 0.762 | WALL  | 0.66  | -1.0  |
| 0.705 | 0.655 | 0.611 | 0.388 |


Final policy:
| RIGHT | RIGHT | RIGHT | +1    |
| UP    | WALL  | UP    | -1    |
| UP    | LEFT  | LEFT  | LEFT  |
```

:Policy iteration

```python
print('@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@')
print("@@@@@@@@@ Policy iteration @@@@@@@@")
print('@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@')

print("\nPolicy evaluation:")
U_eval = policy_evaluation(mdp, policy)
mdp.print_utility(U_eval)

policy = [['UP', 'UP', 'UP', 0],
          ['UP', 'WALL', 'UP', 0],
          ['UP', 'UP', 'UP', 'UP']]
print("\nInitial policy:")
mdp.print_policy(policy)
print("\nFinal policy:")
policy_new = policy_iteration(mdp, policy)
mdp.print_policy(policy)

print("Done!")
```

פלט:

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@ Policy iteration @@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

Policy evaluation:
| 0.812 | 0.868 | 0.918 | 1.0   |
| 0.762 | WALL  | 0.66  | -1.0  |
| 0.705 | 0.655 | 0.611 | 0.388 |


Initial policy:
| UP    | UP    | UP    | +1    |
| UP    | WALL  | UP    | -1    |
| UP    | UP    | UP    | UP    |


Final policy:
| RIGHT | RIGHT | RIGHT | +1    |
| UP    | WALL  | UP    | -1    |
| UP    | LEFT  | LEFT  | LEFT  |

Done!
```