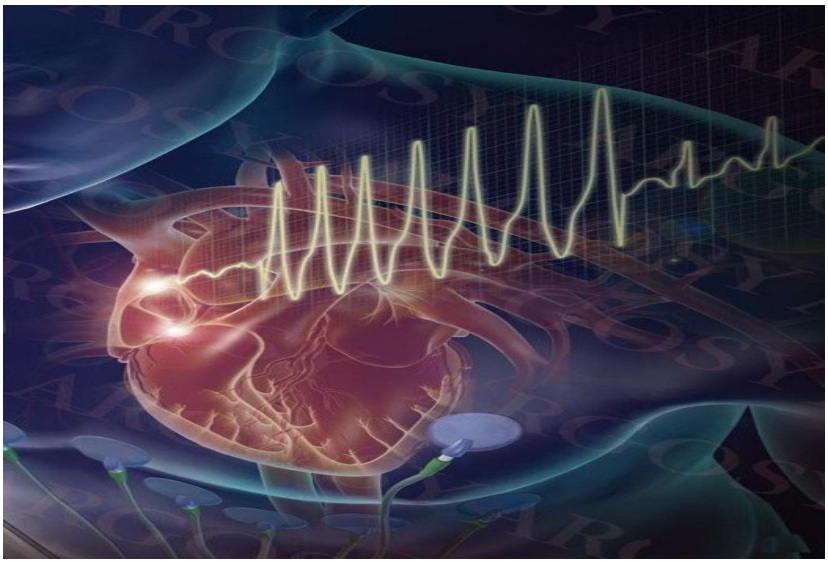
תרגיל בית 3:

למידה וסיווג הפרעות קצב לב

הערות:

- תאריך הגשה: 24.1.19
- את המטלה יש להגיש בזוגות בלבד!
- **בתרגיל זה יש להשתמש בספריות חיצוניות pickle, numpy מומלץ ואף רצוי להשתמש בספריות חיצוניות כמו sklearn אלא אם כן צוין מפורשות לא להשתמש בספריות אלו.**
- שאלות בנוגע לתרגיל יש לשלוח ליניב yanemcovsky@cs.technion.ac.il או לגיא guziel@campus.technion.ac.il
- בקשות מוצדקות לדחייה יש לשלוח למיכל: sbadian@cs.technion.ac.il
- קִראו היטב את ההסברים וההוראות במסמך זה, מטרתם לסייע לכם בהבנת הדרישות של התרגיל. שימו לב להוראות ההגשה המצורפות בסוף התרגיל.
- הִתְעַדְכְּנוּ ברשימת ה-FAQ באתר הקורס בתדירות גבוהה, לפני פנייה בשאלות דרך המייל ולפני הגשת התרגיל. ההערות שתתפרסמנה באתר הקורס מחייבות את כלל הסטודנטים בקורס!
- הקוד שלכם ייבדק על-ידי $Unit-Tests$, לכן עליכם לעקוב בִּתשומת לב רבה אחר הוראות ההגשה המצורפות במהלך התרגיל ובסופו לפני הגשתו.
- בתרגיל זה, כמו גם בתרגילים הבאים בקורס, הרצת הניסויים עשויה לקחת זמן רב ולכן מומלץ מאוד להימנע מדחיית העבודה על התרגיל לרגע האחרון. לא תינתנה דחיות על רקע זה.



חלק א' – מבוא והנחיות

אמיר יצא למילואים כפאראמדיק, אבל לצערו הוא לא זוכר איך לטפל בחולים. במטלה זו נעזור לאמיר לאבחן הפרעות בקצב הלב ע"י מימוש הידע באלגוריתמי למידה שצברנו במהלך הקורס. כחלק מהתרגיל, אתם תדרשו לממש אלגוריתם KNN ולהתנסות בעבודה עם בסיס ידע. מומלץ לחזור על שקפי ההרצאות הרלוונטיים לפני תחילת העבודה על התרגיל.

מהלך התרגיל:

ECG (רִשמת לב חשמלית) הוא רישום של הפעילות החשמלית של הלב הנעשה ע"י הצמדת אלקטרודות על חלקיו השונים של הגוף. קריאת המידע מאלקטרודות אלו מאפשרת לנתח את הפעילות החשמלית של הלב. האות החשמלי אשר נמדד נקרא $Electrocardiogram$ (ECG).

הפרעה בקצב הלב ($arrhythmia$) היא פעילות חשמלית לא תקינה של פעימות הלב. ההפרעה יכולה להתבטא בקצב איטי מדי, מהיר מדי, דילוג על פעימות, קצב המשתנה מפעימה לפעימה וכן בעיוותים נוספים אשר ניתן להבחין בהם ע"י ניתוח המידע הנרשם ב-ECG.

בתרגיל זה נשווה את ביצועיהם של מסווגים לזיהוי הפרעות קצב לב מתוך סט מאפיינים שמוצו מאות ה ECG.

בחלק הראשון של התרגיל נבנה ונבחן מסווג KNN ונשווה אותו למסווגים נוספים, בחלק השני של התרגיל נבנה את המסווג הטוב ביותר שנוכל.

בסיס הידע:

לרשותכם קובץ בשם $Data.pickle$ המכיל מידע מהקלטות ECG אמתיות אשר בוצעו על 1300 נבדקים (1000 אימון 300 טסט), כאשר כל בדיקה מופיעה כווקטור מאפיינים. מאפיינים אלו הם התכונות שישמשו אותנו לצורך סיווג בתרגיל הנוכחי.

בנוסף, נתונה לכם פונקציה בשם $load\_data$ המופיעה בקובץ $hw3\_utils.py$. הפונקציה מקבלת את הנתיב לקובץ $Data.pickle$ ומחזירה 3 מטריצות (מאפייני הECG עבור כל נבדק בקבוצת האימון, תיוג עבור כל נבדק בקבוצת האימון (חולה או בריא), ומטריצת דגימות ללא תיוגים המייצגת את 300 נבחני הטסט). כל שורה במטריצה מייצגת נבחן אחד וכל עמודה במטריצות הפיצ'רים היא פיצ'ר שונה

חלק ב' – פיתוח KNN:

בחלק זה, מלבד שאלה 7, אין להשתמש בספריות חיצוניות

בשלב ראשון, נבנה מסווג $k-NearestNeighbors$ (KNN) ונבחן את היכולת שלו להבחין בין האנשים הסובלים מהפרעות קצב לבין אנשים בריאים.

מימוש KNN:

כפי שנלמד בכיתה, אלגוריתם KNN מסווג אובייקט נתון על ידי מציאת k הדוגמאות הקרובות ביותר לאובייקט והחזרת הסיווג המתקבל לפי החלטת הרוב. לצורך זה, יש להגדיר מדד למרחק בין זוג דוגמאות. דרך מקובלת לבדוק מרחק בין ווקטורים של תכונות נומריות היא באמצעות מרחק אוקלידי ($euclideandistance$). נזכיר כי עבור שני ווקטורים $x,y \in R^n$ מרחק אוקלידי מחושב בדרך הבאה:

$$\|x-y\| = \sqrt{(x_1-y_1)^2+(x_2-y_2)^2+\ldots(x_n-y_n)^2}$$

1. ממשו פונקציה בשם $euclidean\_distance$ המקבלת שתי רשימות מאפיינים המייצגות נבדקים, ומחזירה את המרחק האוקלידי ביניהם לפי הנוסחה הנ"ל.

2. ממשו את המחלקות הבאות הדרושות לצורך בניית המסווג $nearest\_neighbors$ בקובץ $classifier.py$:
   1. ממשו מחלקה בשם $knn\_classifier$ היורשת מהמחלקה האבסטרקטית הנתונה $abstact\_classifier$. על המחלקה להכיל פונקציה בשם $classify$. פונקציה זו תקבל כקלט דוגמא (המיוצגת כווקטור המאפיינים שמוצו מאות ה ECG) ותחזיר את הסיווג המתקבל ע"י אלגוריתם KNN (0 אם הנבדק סובל מהפרעת קצב לב ו 1 אחרת). המסווג ישתמש במדד המרחק $euclidean\_distance$ שהוגדר בשאלה הקודמת.
   2. ממשו מחלקה בשם $knn\_factory$ היורשת מהמחלקה האבסטרקטית הנתונה $abstract\_classifier\_factory$. על המחלקה להכיל פונקציה בשם $train$. פונקציה זו תקבל כקלט את סט האימון ותחזיר אובייקט מסוג $knn\_classifier$. כל אובייקט ($instance$) של המחלקה ייצר מסווגים עבור ערך k (מספר שכנים) יחיד.

הרצת הניסויים:

נרצה להעריך את האלגוריתם שלנו בשיטת $Stratifiedk-foldCrossValidation$ שנלמדה בתרגול. בשיטה זו, נחלק את קבוצת הדוגמאות שלנו ל- k קבוצות שוות כך שבכל תת-קבוצה יחס הדוגמאות החיוביות והשליליות יהיה זהה לזה של המדגם כולו.

3.
   1. ממשו פונקציה בשם $split\_crosscheck\_groups(dataset, num\_folds)$. פונקציה זו תקבל רשימה של דוגמאות ומספר $folds$ ותחלק את הרשימה באופן אקראי ל num_folds קבוצות שוות כפי שנהוג ב $Stratifiedk-foldCrossValidation$. על כל קבוצה להכיל את מאפייני הECG והתיוג (חולה או בריא) עבור כל נבדק בקבוצה.

הפונקציה תשמור כל תת קבוצה לקובץ בשם $ecg\_fold\_\langle i\rangle.data$ כאשר $\langle i\rangle$ הוא האינדקס של תת הקבוצה ($i \in \{1, \ldots, num\_folds\}$). יש לממש פונקציה השם load_k_fold_data המקבלת כאינפוט את האינדקס i של תת הקבוצה ומחזירה טאפל (train_i , labels_i)

להוסיף להגשה!

2. הריצו את הפונקציה הנ"ל עבור $num\_folds = 2$, וצרפו את קבצי הפלט להגשת התרגיל. בכל הניסויים שתבצעו לאורך התרגיל עליכם להשתמש בחלוקה זו ל- $folds$ (כלומר, יש לקרוא לפונקציה פעם אחת בלבד). הסבירו מדוע חשוב לשמור על עקביות זו.

4. כתבו פונקציה בשם $evaluate(classifier\_factory, k)$ שבה $classifier\_factory$ הוא אובייקט של מחלקה היורשת מ $abstract\_classifier\_factory$ ו k הוא מספר ה $folds$. הפונקציה תריץ $k-fold\ cross\ validation$ כאשר בכל שלב נבחרת קבוצה אחת מהקבוצות שיצרתם בשאלה 4 להיות קבוצת ההערכה ושאר הקבוצות יאוחדו לקבוצת האימון. הפונקציה תחזיר את הדיוק ואת השגיאה הממוצעים שהתקבלו.

תזכורת:

$$Accuracy = \frac{True\ Pos. + True\ Neg.}{N}$$

$$Error = \frac{False\ Pos. + False\ Neg.}{N}$$

$(*)$ למען ההבהרה, בהינתן המימוש שלכם ניסוי KNN עבור $k = 3$ שכנים עם $2folds$ יכול להראות כך:

$$patient, labels, test = load-data\ \ ('Data.pickle')$$

$$split\_crosscheck\_groups(patients, labels, 2)$$

$$knn3 = knn\_factory(3)$$

$$accuracy, error = evaluate(knn3, 2)$$

5. עבור כל $k \in \{1, 3, 5, 7, 13\}$ היעזרו בפונקציית $evaluate$ שכתבתם בשאלה 4 בכדי לבחון את ביצועי האלגוריתם שמימשתם בשאלה 2 עם k שכנים.

1. הגישו קובץ בשם $experiments6.csv$ שיכיל את תוצאות הניסויים שהרצתם. על כל שורה בקובץ להכיל 3 עמודות: ערך ה- k הנבדק, דיוק ממוצע ושגיאה ממוצעת.

לנוחיותכם מסופקת שורה לדוגמא:

$$3, 0.82, 0.18$$

אין לכלול כותרות בקובץ, אלא רק את תוצאות הניסויים.

2. סרטטו גרף של הדיוק הממוצע של KNN (כפי שהוחזר מהפונקציה $evaluate$) כפונקציה של k.
3. עבור איזה ערך של k התקבלו הביצועים הטובים ביותר?
4. הסבירו ונתחו את הממצאים בגרף. באילו מגמות ניתן להבחין? מהם ערכי המקסימום והמינימום? מדוע?

7. לצורך השוואה, נרצה לאמן מסווגים נוספים ולהציג את ביצועיהם. אין צורך לממש את האלגוריתמים בסעיף זה, וניתן להשתמש במימושים הנמצאים בספריית sklearn
   1. אמנו עץ החלטה ללא גיזום שמשתמש ב- $3DI$ כפי שנלמד בכיתה.

היעזרו בפונקציית $evaluate$ שכתבתם בשאלה 4 בכדי לבחון את ביצועי האלגוריתם.
לצורך הגשת התוצאות נתייחס לניסוי זה כניסוי מספר 1

   2. חזרו על סעיף א' עבור אלגוריתם perceptron כפי שנלמד בכיתה, לצורך הגשת התוצאות נתייחס לניסוי זה כניסוי מספר 2
   3. הגישו קובץ בשם $experiments12.csv$ שיכיל את תוצאות הניסויים שהרצתם. על כל שורה בקובץ להכיל 3 עמודות: מספר ניסוי, דיוק ממוצע ושגיאה ממוצעת.

לנוחיותכם מסופקת שורה לדוגמא:

$$1, 0.87, 0.13$$

אין לכלול כותרות בקובץ, אלא רק את תוצאות הניסויים.

   4. מבין כל הניסויים שהרצתם, כולל אלו משאלה 6, באיזה ניסוי התקבלו התוצאות הטובות ביותר?

חלק ג' – תחרות ובונוס

נרצה לבחון את הניסיון שצברתם באימון מסווגים. לצורך זה שמרנו בצד תת קבוצה של הקלטות ECG, שעבורה לא נתונים לכם הסיווגים. קבוצה זו תשמש כקבוצת מבחן. עליכם להיעזר בידע שלכם בהרצת ניסויים כדי לאמן מסווג שיקבל אחוזי דיוק גבוהים ככל שניתן על קבוצת המבחן.

מתווה התרגיל:

- עליכם לבנות באופן מלומד מסווג כלשהו לבעיה הנתונה.
- הדוגמאות המופיעות בקובץ $Data.pickle$ ישמשו כקבוצת אימון לצורך אימון ובחירת המסווג, ואתם רשאים להיעזר בהן כפי שתראו לנכון. בנוסף, אתם רשאים להשתמש בכל קוד או חבילה שתרצו בתנאי שתציינו בתרגיל את מקור הקוד, ותספקו הסבר קצר של מה הקוד עושה.
- בעזרת המסווג שבניתם עליכם לתייג את קבוצת המבחן הנתונה.
- עליכם להגיש קובץ בשם $results.data$ עם הסיווגים שלכם לדוגמאות המבחן. לנוחיותכם, הפונקציה $write_{prediction}$ המסופקת מדפיסה את רשימת הסיווגים שלכם בפורמט הנכון.

שימו לב כי אתם רשאים להגיש קובץ אחד כזה בלבד.

- יש לצרף להגשה הסבר שבו אתם מפרטים איך המסווג שכתבתם עובד, ואת הניסויים שביצעתם כדי לבחון אותו.

המלצות ורמזים:

- עברו על כל סוגי המסווגים שלמדנו בכיתה ועל אפשרויות שונות לפרמטרים עבורם ובחנו את ביצועיהם בצורה איכותית ולא רק כמותית, בחנו אם ישנם דוגמאות עליהם מסווג אחד נוטה לטעות יותר מאשר מסווג אחר.
- נסו לשלב בין סוגים שונים של מסווגים, ובין מסווגים עם פרמטרים שונים.
- נסו להבין מהי הבעייתיות אתה אתם מתמודדים, חשבו וחפשו דרכי התמודדות.
- בחנו את סוגי הדוגמאות שעליהם המסווג נכשל, ונסו למצוא דרכים להתמודד איתם.
- עברו על הדרכים לשיפור ביצועי מסווגים שלמדנו בכיתה ובדקו אם הם עוזרים, אם כן חשבו מדוע.

- בחנו אפשרויות לעבד את הדוגמאות בשלבים שונים בתהליך, ייצוג שונה של התכונות יכול להקל על הלמידה ולעזור לביצועים.
- בחנו אפשרויות לבנות תכונות חדשות מתוך הקיימות, חשבו באילו מקרים נוכל להסיק יותר מתכונות כאלו.
- בדקו אם ישנם תכונות שאינן עוזרות לסיווג, נסו להבין מדוע ומה ניתן לעשות.

הכוונה:

- הריצו את אלגוריתם ID3, הסתכלו על הדיוק של העץ המתקבל ונסו לבחון את המקרים שבהם הסיווג שגוי, נצלו את פשטות העץ המתקבל ונסו להבין מדוע מקרים אלה בעייתיים, בדקו גם אם יש דרך להתמודד עם מקרים אלו על ידי ייצוג שונה של התכונות.
- תוכלו לנסות לקבל ועדה של עצים על ידי בחינה של אלגוריתמים שונים מהמשפחה של TDIDT, וגם על ידי גיזום. וועדה יכולה להיות גם ממשוקללת (משקל שונה לסיווג המתקבל מכל עץ) והמשקלים יכולים להיות פרמטרים של האלגוריתם, בדומה לאלגוריתם AdaBoost.
- בחנו את הביצועים של המסווג המתקבל כמותית ואיכותית ונסו להמשיך ולשפר את הדיוק ככל שניתן.

ניקוד:

- סגל הקורס יקיים תחרות בין התשובות שהתקבלו כדי לקבוע את הציונים עבור חלק זה, ומי יקבל את הבונוס.
- התרגיל שיגיע לדיוק הגבוה ביותר יקבל בונוס של 15 נקודות, מקום שני יקבל בונוס של 10 נקודות ומקום שלישי 7 נקודות.
- כל פתרון יצירתי שלא יכנס לשלישיה העליונה יקבל בונוס של 5 נקודות.
- נקודות הבונוס יתווספו לציון תרגיל הבית ולא לציון הסופי בקורס.

הוראות הגשה

- הגשת התרגיל תתבצע **אלקטרונית בלבד**.
- עליכם להגיש קובץ ארכיון יחיד בשם: AI3_<id1>_<id2>.zip (ללא הסוגריים המשולשים). קובץ זה יכיל:
  - קובץ בשם readme.txt בפורמט הבא:

    name1 id1 email1

    name2 id2 email2
  - קובץ בשם AI_HW3.PDF המכיל את דו"ח הניסויים שערכתם, תשובות לחלק היבש והערות לקוד שהגשתם (כולל תפקיד כל קובץ הנמצא בתיקייה שהגשתם).
  - קובץ בשם requirements.txt שיכיל כל חבילה חיצונית שאינה מותקנת כחלק מ-Anaconda Python. על אחריותכם לוודא כי ניתן להתקין כל חבילה כנ"ל באמצעות הרצת השורה: pip install –r requirements.txt
  - כל הפונקציות וקבצי הקוד שהתבקשתם לממש בתרגיל זה.
  - כל קוד עזר שכתבתם/השתמשתם בו לשם הרצת הניסויים או יצירת הגרפים.
  - כל קובץ פלט שהתבקשתם ליצור לצורך התרגיל.
- אין להעתיק את הקבצים המסופקים לכם אל תוך תיקיית ההגשה. הניחו כי קבצים אלו יהיו זמינים בעת בדיקת התרגיל.

- שימו לב שכל הפנייה למיקום קובץ/תיקייה כלשהם בקוד תהיה רלטיבית (*relativepath*) ולא אבסולוטית, כך שהקוד יעבוד כפי שהוא על כל מחשב בכל מיקום שנבחר לתיקיית הפרוייקט. הקפידו לבדוק זאת לפני ההגשה!
- "המצאת" נתונים לצורך בניית הגרפים **אסורה** ותוביל לדיון בבית הדין המשמעתי של הטכניון.
- אתם רשאים לעשות שימוש בכל קוד שתמצאו ברשת, אך כל קוד חיצוני **מחייב הצהרה מפורשת** על המקור שלו בקובץ AI_HW3.PDF. אי-קיום דרישה זו מהווה עבירה משמעתית!
- הקפידו על קוד **ברור, קריא ומתועד!** עליכם לתעד כל חלק שאינו טריוויאלי בקוד שלכם. בפרט, אם השתמשתם בקוד שנמצא ברשת וביצעתם בו שינויים, עליכם לתעד זאת.

בהצלחה!

נספח – תיאור התכונות

אות ECG מורכב ממספר לידים (I, II, II AVL, AVR, AVF, V1, V2, V3, V4, V5, V6). האות בכל ליד מתחלק למספר גלים (P, Q, R, S, T וכולי), צורת ונוכחות כל גל מעידה על התנהגות מסוימת של הלב, או על בעיה. מידע נוסף נתן למצוא בלינק : https://en.wikipedia.org/wiki/Electrocardiography