תרגיל בית 3 – מבוא ללמידה

# הנחיות כלליות:

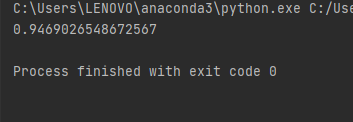
* תאריך ההגשה: 26/01/2021 ב23:59
* התרגיל הוא להגשה **ביחידים.**
* המתרגל האחראי על התרגיל: רפאל גד, נא לפנות בשאלות אך ורק למייל [ai.technion@gmail.com](mailto:ai.technion@gmail.com).
* הקוד שלכם ייבדק אוטומטית וגם ידנית, רמאות כלשהי תוביל לועדת משמעת. התרגיל מהווה חלק משמעותי מהציון הסופי שלכם בקורס, כל תקשורת בין סטודנטים בכל פלטפורמה שהיא בנוגע לתרגיל אסורה ומהווה רמאות. קבוצת הקורס תפעיל אמצעי בינה מלאכותית מתקדמים לעלות על הרמאים.
* אנא הקפידו על ההנחיות, כל הפרה תגרום להורדת ניקוד לרבות אי עמידה בציפיות פורמט הפלט של כל קובץ הנדרש מכם.
* בכל יום חמישי בשעה 14:30 רפאל יקיים כיתה הפוכה בה תוכלו לשאול שאלות לגבי התרגיל.
* בחלק מהסעיפים קיימת הגבלת שורות לפתרון שלכם. אם אתם חושבים שבסעיף מסוים הגבלה זו לא ריאלית אנא שילחו לרפאל נימוק על כך במייל ואם הנימוק יתקבל נשנה את ההגבלה לכלל הסטודנטים.
* הקוד שלכם צריך להתייחס לקבציי הדאטה כנמצאים בתיקייה הנוכחית ולא בתת תיקייה. אין לשנות את שמות קבצי הדאטה.
* בסעיפים הרטובים בתרגיל מסופקות לכן מספר מועט יחסית של הנחיות ביחס לתרגילים הקודמים. כל מימוש העונה על הדרישות יתקבל. יש לכם יד חופשית בתכנון הקוד שלכם. אנו ממליצים לממש כל מסווג כ- class שמממש פונקציות fit(X, Y) שמבצעת אימון, ו- predict(X) שמבצעת את הפרדיקציה.
* בחלק מהשאלות תתבקשו להמציא שיפור לאלגוריתם למידה, כתוצאה מכך יש בהן דרגות חופש, הציון בשאלות אלו יינתן בהתאם לטיב הפתרון, הן מבחינת יצירתיות והן מבחינת תוצאות אמפיריות. יש לתאר בצורה ברורה, פורמלית ומדויקת את הפתרון.
* מותר להשתמש בספריות ,sklearn, pandas ,numpy, random, matplotlib, argparse,abc,typing, all the built in packages in python אך כמובן שאין להשתמש באלגוריתמי הלמידה, או בכל אלגוריתם או מבנה נתונים אחר המהווה חלק מאלגוריתם למידה אותו תתבקשו לממש.

# רקע:

*לתרגיל מצורף קובץ נתונים על מחלה מסוימת, כאשר כל שורה מתארת אדם. העמודה הראשונה מציינת האם האדם חולה (M) או בריא (B). שאר העמודות מציינות כל מיני תכונות רפואיות של אותו אדם (התכונות קצת מורכבות ואינכם צריכים להתייחס למשמעות שלהן כלל). בשאלות הבאות תתבקשו לממש אלגוריתמי למידה שונים מנת לאפשר חיזוי מדויק ככל האפשר של היות אדם חולה במחלה.*

***בהצלחה!***

1. (15 נק') **אלגוריתם ID3:**
   1. ממשו את אלגוריתם ID3 כפי שנלמד בהרצאה.  
      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 אשר בהרצתו מאמן עץ על קבוצת האימון בעזרת המימוש שלכם ומדפיס את הדיוק שלו על קבוצת המבחן ללא כל מלל או הדפסות אחרות (ככה יהיו גם כל שאר ההדפסות שלכם בתרגיל).
   2. אמנו את האלגוריתם על קבוצת האימון ובדקו אותו על קבוצת המבחן. צרפו צילום מסך של תוצאת הדיוק בדו"ח.



1. (4 נק') הוכח/הפרך: בהינתן דאטה כלשהו עם תכונות רציפות ותיוגים בינאריים המחולק לקבוצת אימון ומבחן, הפעלה של פונקציית נירמול MinMaxהנלמד בתרגול על הדאטה אינה משפיעה על דיוק של מסווגID3 הנלמד על קבוצת האימון והנבחן על קבוצת המבחן. ]אורך התשובה מוגבל ל20שורות]

**הוכחה** , האינטואיציה לכך הוא ש id3 מסתמך על פונקצית IG שהיא מחושבת עבור תכונה ספציפית באופן לוקלי ללא שום קשר לשאר התכונות , נראה שאם תכונה זו מנורמלת או לא , עדיין נקבל אותה חלוקה של השורש שזה גורר שלשניהם יש אותו IG.

**באופן יותר פורמלי** :

תהי תכונה f רציפה כלשהיא:

נסמן max\min להיות את הערך הכי הגדול\קטן של התכונה בקבוצת האימון.

ערכי התוכנה בקובצת האימון ממוינים הם :  *ערכים אילו מנורמלים הם :*

*הספים עבור התכונה המקורים הם : , כאשר הסף הטוב ביותר מתקבל עבור כלשהוא ()  
הספים עבור התכונה המנורמלת הם :*

*נראה כי גם הסף הטוב ביותר מתקבל עבור אותו j , כלומר:*

כדי להראות את זה נוכיח את הטענה הבאה לכל דגימה v כלשהיא (לא משנה אם היא מקבוצת המבחן או מקבוצת האימון):

*הוכחה:*

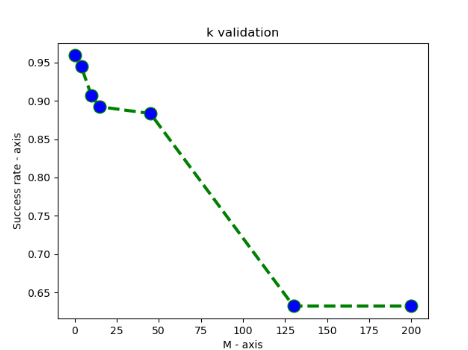
*כלומר נקבל אותה חלוקה בדיוק לא משנה אם עושים נרמול לתכונות או לא זה אומר אותו IG לשתי התכונות המנורמלת והרגילה.  
(כל הדגימות שהיו יותר גדולים מהסף ישארו יותר גדולים מהסף המנורמל אחרי הנרמול ואותו דבר עבור הדוגמאות שהיו יותר קטנות מסף).****לכן התנהגות id3 לא תשתנה****.*

1. (12 נק׳) **גיזום מוקדם.**
   1. הסבירו מה החשיבות של הגיזום באופן כללי ואיזה תופעה הוא מנסה למנוע? [אורך התשובה מוגבל ל3 שורות]

מטרת הגיזום באופן כלילי היא ליצור **עצים קטנים ולאו דווקא עקביים** (כי לפי המחקרים מראים שעצים יותר קטנים הם אלא שיותר טובים) , והוא מנסה למנוע את תופעת ה **overfitting** (התאמת יתר).

* 1. ממשו את הגיזום המוקדם כפי שהוגדר בהרצאה. הפרמטר M מציין את מספר המינימלי בעלה לקבלת החלטה. על המימוש של הגיזום המוקדם להימצא גם כן בקובץ ID3.py.
  2. 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](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.KFold.html#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).
     3. השתמשו בתוצאות שקיבלתם כדי ליצור גרף המציג את השפעת הפרמטר M על הדיוק. צרפו את הגרף בדו״ח.

בחרתי M\_List = [0,4,10,15,45,130 , 200]

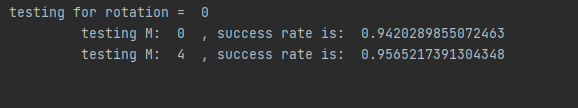


* + 1. הסבירו את הגרף שקיבלתם. לאיזה גיזום קיבלתם התוצאה הטובה ביותר ומהי תוצאה זו?

כפי שרואים הגיזום בכלל לא שיפר את אחוז ההצלחה (התחלתי מ ) , זה לא אומר שהגיזום הוא דבר רע , אלא יתכן שזה קרה כי ב train\_data יש מעט דוגמאות רועשות , במיוחד ש ב k\_fold אני לוקח רק חלק מה train\_data ומשאיר את החלק השני ל test ולכן ההסתברות שתיפול דוגמה רועשת בחלק הזה של ה train\_data קטנה (אפילו אם היא נפלה ב train\_data באיטרציה מסויימת , באיטרציה אחרת היא תהפוך ל test ולכן סה"כ בממוצע הגיזום לא שיפר).

לגבי ערכי M:

* ערך של M שהוא קיצוני (130 לדוגמא) הוא ממש רע כי אז נגזום ממש מוקדם , דבר זה גורם לעץ שנוצר להיות לא עקבי עם הרבה דוגמאות ולא רק הרועשות וזה בעייתי כמובן.
* זה לא משנה שעבור שני M שונים יש אותו אחוז הצלחה למשל עבור 130 ו 200 כי יש מצב ש ב node מסויים יש 120 דגימות ואז שני ה M יסווגו אותו דבר ולכן זה לא בעייתי.
* ככל שהגדלנו את M אחוז ההצלחה ירד , שוב זה לא אומר שהגיזום הוא משהו רע אלא יש מצב שב data\_train שקיבלנו יש בו יחסית מספר קטן של דוגמאות רועשות ולכן בממוצע עדיף לא לגזום אפילו. (עבור איטרציות ספיציופיות בתוך ה k\_fold דווקא הגיזום שיפר כי כנראה התמזל מזלנו ונפל ל train\_data חלק גדול של דוגמאות רועשות, אבל בממוצע הוא לא שיפר).



על מימוש כיוונון הפרמטר להימצא בפונקציה בשם experimentבקובץ ID3.py. הוסיפו הערה בתחילת פונקציה זו לגבי איך להריץ את הפונקציה שלכם לקבלת הגרף. שימו לב שבריצה של הקובץ ID3.py עדיין אך ורק הדיוק של ID3 ללא גיזום צריך להיות מודפס.

* 1. 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 3.3. ציינו בדו"ח את הדיוק שקיבלתם? האם הגיזום שיפר את הביצועים ביחס להרצה ללא גיזום בשאלה 1?

אז שוב , אני קיבלתי ש M הכי טוב הוא 0 , אבל בכל זאת הרצתי עבור (השני הכי טוב לפי הניסוי שלי) וקיבלתי את אחזו הדיוק הבא:



אני לא מופתע מזה ששיפרתי את אחוז ההצלחה למרות שבניסוי שלי לא שיפרתי , כי כפי שאמרתי בניסוי שלי , כנראה שיש מעט דוגמאות רועשות , וגם תשובה זו הינה ספציפית *ל test הזה , אבל תשובה היא עבור מקרה כללי יותר ולא עבור test ספציפי לכן לפי דעתי היא יותר טובה כי בממוצע היא תתנהג יותר טוב (זה מה שהראה הניסוי שלי)*

1. (20 נק׳) **למידה מוכוונת מחיר.**  
   במקרה שלנו, בשל אופי הבעיה, סיווג אדם חולה כבריא חמורה פי 10 מסיווג של אדם בריא כחולה. לפיכך, הדיוק שהצגתם בשאלות 1 ו- 3 אינו משקף היטב את טיב המסווג שלכם. בעזרת משקול סוגי השגיאות בהתאם למה שלמדתם בתרגול, נגדיר פונקציה חדשה שתייצג את טיב ביצועי המסווגים:  
   כשאר FP הינו מספר הסיווגים השגויים של בריאים כחולים על ידי המסווג C, FN הינו מספר הסיווגים השגויים של חולים כבריאים על ידי המסווג C ו- הוא מספר הדוגמאות בקבוצת המבחן.  
     
   בשאלה זו תתאימו את אלגוריתם ID3 ללמידה מכוונת מחיר.(אם כבר פתרתם עם KNN התשובות גם כן מתקבלות)
   1. מדדו את ערך ה- loss של ID3: חזרו על סעיף 3.4 ובמקום הדיוק תמדדו את ערך ה- loss.

עבור M=0:



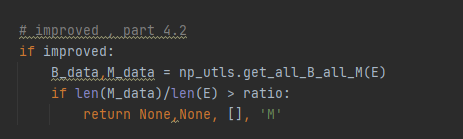
עבור M=4 (קיבלנו שיפור קטן):



* 1. תארו דרך לגרום לID3 ללמוד מסווג אשר ממזער את פונקציית ה- loss שהוצגה כאן בצורה טובה יותר מאשר האלגוריתם הרגיל.

השיפור שלי הוא ככה:

בתוך בניית ה TDIDT נוסיף עוד תנאי לעצירת הרקורסיה שהוא:



המשעות של תנאי זה הוא שאם אחוז החולים בצומת מסויים עולה על סף (שאותו אכוונן בעזרת kfold )

, אז תמיר צומת זו לעלה עם סיווג של חולה.

המטרה של שיפור זה היא לגרום ל id3 לסווג יותר אנשים כחולים על מנת שיקטין את ה loss.

אינטואציה לשיפור:

אם בצומת מסויים יש לנו כבר 90% חולים אז רוב הסיכוי שכל מי שנמצא בצומת זו הוא חולה בפועל ולכן נסווג את כל הצומת כחולים (ואז נרוויח במקרה והיה לנו דוגמאות רועשות).

כמובן דבר זה לא תמיד נכון כי יתכן שיש מקרי יוצא דופן (שיש אנשים מסכימים על רוב התכונות עם אנשים חולים אבל הם בפועל בריאים) .

* 1. ממשו הצעתכם בקובץ CostSensitiveID3.py. על קובץ זה להדפיס בהרצתו את ה- loss שקיבלתם מהרצת האלגוריתם המשופר כאשר הוא לומד על קבוצת האימון ונבחן על קבוצת המבחן. אם ביצעתם ניסויים לקביעת פרמטרים לאלגוריתם שלכם, פרטו זאת. כמו כן צרפו בדו"ח את ה-loss שקיבלתם. שימו לב, אינכם צריכים לדאוג מכך שהשיפורים שלכם יפגעו בדיוק ובלבד שהם ישפרו את הloss.

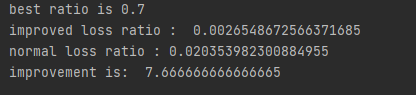
עשיתי kfold כדי לכוונן את הפרמטר ratio בפונקציה שנקראת experiement (כמובן שהיא כבוייה עכשיו).

אופן הכוונון:

לקחתי :

חלקתי את ה data ל 5 סטים , וכל פעם היה סט אחד שהוא לטסט ו4 הסטים האחרונים הם train data בתוך כל לולאה כזו בדקתי את ה loss rate עבור כל ה ratios השונים , ולבסוף בחזרתי את ה ratio עם הממוצע המינמלי (שמקטין את ה loss rate הכי הרבה).

אחרי זה יצרתי מסווג עם ה ratio הכי טוב ובחנתי אותו על הטסט המקורי וקיבלתי את ה loss rate הבא:



קיבלתי שה ratio הכי טוב הוא 0.7.

כלומר אם מספר החולים בצומת מסויים הוא יותר גבוה מ 70% מסה"כ האנשים בצומת זו אז ישירות תסווג את כולם כחולים.

1. (9 נק׳) נגדיר דטה סט שבו n דוגמאות מתויגות עם סיווג בינארי . כל דוגמה היא וקטור תכונות המורכב משתי תכונות רציפות .  
   הניחו כי קיים מסווג מטרה שאותו אנו מעוניינים ללמוד (הוא אינו ידוע לנו) וכן שהדוגמאות ב- עקביות עם מסווג המטרה (כלומר שאין דוגמאות רועשות ב- ).  
   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 מקסימלי ובמקרה של שוויון בערך של , מתחשבים קודם בדוגמאות עם ערך מקסימלי. הניחו כי אין דוגמאות זהות לחלוטין (כלומר גם עם ערך זהה וגם עם ערך זהה).  
   מסווג ID3 כאן צריך לתאום לכל תנאי קצה שהצגנו למימוש שלכם.  
   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 טריוויילי, דהיינו שמסווג כל הדוגמאות כחיוביים או כל הדוגמאות כשליליים. [2 שורות לכל סעיף, אין הגבלה על הגרפים, יש להימנע ממלל ופתרון שאינו מוגדר היטב כמתבקש לא מקבל ניקוד]

**סעיף (א)**  
הציגו מסווג מטרה 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. (2 נק')

נגדיר את מסווג המטרה הבא לכל :

נגדיר את קבוצת הדוגמאות הבאה:

*לגבי id3: נמיין את ערכי , מקבלים : -2,-1,1 id3 ינסה לפצל קודם כל לפי -1.5 וישאר עם אי וודאות שגדולה מ 0 כי תשאר לו צומת שתכיל (-1,0) ו (1,0) ושניהם עם סימנים הפוכים. עכשיו הוא ינסה לפצל לפי 0 , ואז יקבל אי וודאות של 0 (יקבל צומת אחד שבה בן אחד חיובי , וצומת שבה שני בנים שלילים) לכן הוא יבחר עם סף 0 שזו בדיוק פונקצית המטרה , לכן הוא לעולם לא יטעה.*

לעומת זאת אם הוא החליט לפצל לפי יקח את הממוצע שזה 5 ויקבל אי וודאות יותר מ 0 (כי הוא ישאר לו צומת אחד עם 2 בנים שהסימונים שלהם הפוכים (-2,6) ו (1,10)) אז יחליט לפצל לפי  *עם סף 0 שזה בדיוק פונקצית המסווג לכן הוא תמיד יפגע.*

יהא  *כלשהו , נחלק למקרים:  
אם , ניקח את הנקודה (-0.00000000001, 10) אז KNN יניב תוצאה + כי (1,10) היא הנקודה הכי קרובה עליה , אבל זו טעות כי הסיווג האמיתי הוא – כי עבורה הוא שלילי.*

*אם : ניקח (2,0) , KNN תמיד יחזיר - כי הרוב הוא - אבל בפועל היה צריך להחזיר + כי לכן גם פה יטעה.   
(הראינו שלכל K , קיימת דוגמה שעבורה KNN יטעה והוכחנו ש זהה לפונקצית המטרה).  
\*נקודה אחרונה: הנחתי שלא צריך להתיחס למקרים בהם k זוגיים כי זה בעייתי.*

**סעיף (ב)**הציגו מסווג מטרה 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. (2 נק')

נגדיר את מסווג המטרה הבא לכל :

(אינטואיציה לפונקציה: אם הנקודה יותר קרובה ל (1,1) מאשר (2,2) אז תחזיר 0 , אחרת תחזיר 1)

נגדיר את קבוצת הדוגמאות הבאה:

לגבי ID3 : לא משנה עבור איזו פיצר הוא יפצל הוא יקבל אותם תוצאת שזה שני צמתים ובכל צומת דגימה בודדת (כי הערכים של הפיצרים השונים שווים) *לכן הוא יבחר לפצל לפי עם סף 1.5 (כי האינקס שלו יותר גדול). אז הוא יטעה עבור דוגמה (0,2), כי הוא יסווג אותה כחיובית אבל בפועל היא שלילית כי היא יותר קרובה לדוגמא (1,1).  
מרחק בין (1,1) ל (2,0) (השורש לא משנה):*

*מרחק בין (2,2) ל (2,0) (השורש לא משנה):*

*לכן היסווג האמיתי של נקודה (0,2) הוא שלילת כי היא יותר קרובה ל (1,1) אבל הוא יסווג אותה כחיובית היא ערך ה שלה יותר גדול מ 1.5.*

*לגבי KNN: נראה כי KNN תמיד פוגע עבור , לכל נקודה K מחזיר סיווג שלילי אם הנקודה יותר קרובה ל (1,1) מאשר (2,2)אחרת הוא יחזיר סיווג חיובי. זה בדיוק מה פונקצית מסווג המטרה עושה.*

**סעיף (ג)**  
הציגו מסווג מטרה 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. (2 נק')

נגדיר את מסווג המטרה הבא לכל :

נגדיר את קבוצת הדוגמאות הבאה:

*לגבי ID3 : הוא יסווג את כל הדוגמאות שליליות בפרט עבור אבל זה לא נכון כי עבור דוגמא זו הסיווג האמיתי שלה הוא חיובי כי .*

*דוגמה שבה יכשל KNN: גם (1,1) , כי KNN יסווג את כל הדוגמאות כשליליות בגלל שהרוב שקרוב לו (רק אחד) הוא שלילי , אבל סיווג דוגמא זו הוא חיוב ולכן גם הוא יטעה.*

**סעיף (ד)**הציגו מסווג מטרה 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). (2 נק')

נגדיר את מסווג המטרה הבא לכל :

נגדיר את קבוצת הדוגמאות הבאה:

*לגבי ID3 הוא יפצל לפי עם סף 0. כי עבורו האי וודאות היא 0 , לעומת זה אם הוא פיצל לפי עם סף 0 אז הוא ישאר עם אותה צומת כלומר אי וודאות יותר מ 0 , ולכן הוא יבחר לפצל עבור עם סף 0. לכן הוא יהיה בדיוק כמו פונקצית מסווג המטרה ולא יטעה אף פעם.*

*לגבי KNN: ניקח , ונראה שלכל מתקיים שהסיווג של KNN נכון.*

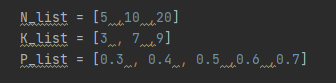
*עבור :  
נראה שהמרחק בין (-1,0) לבין (v1,v2) יותר קטן מהמרחק בין (1,0) לבינו , אם הוכחנו את זה אז זה מראה ש 1NN תמיד פוגע נכון עבור דוגמאות שהסיווג האמיתי שלהם הוא חיובי.*

*לכן עבור , 1NN תמיד יבחר אותו כחיובי כי תמיד הוא יהיה יותר קרוב לוקטור  
(-1,0).   
המקרה הוא דואלי עבור ולכן גם פונקצית ה 1NN תהיה שווה לפונקצית מסווג המטרה.*

1. (20 נק׳) **יער.**  
   ננסה עכשיו לשפר את הביצועים שלנו על ידי שימוש בועדות כפי שלמדתם בהרצאה. נציע כאן סוג חדש של יער שתממשו בעצמכם בשם KNN-decision-tree.  
   האלגוריתם תחילה ילמד N (פרמטר) עצי החלטה, כל אחד עם תת קבוצה שונה של דוגמאות מקבוצת האימון. בעת סיווג דוגמא חדשה האלגוריתם יבחר K (פרמטר) עצים מתוך הN שנלמדו, יסווג את הדוגמא לפי כל אחד מK העצים ויחזיר את הסיווג הנפוץ ביותר.   
   בחירת דוגמאות לכל עץ מתוך N העצים שנלמד: גודל קבוצת האימון יסומן ב-n. עבור כל עץ מ-N העצים הגרילו (באופן רנדומלי) p\*n דוגמאות מקבוצת האימון, כאשר p הוא פרמטר לאלגוריתם וערכו בין 0.3 ל0.7.  
   בחירת K עצים בהם נשתמש לסיווג דוגמא: עבור כל עץ, נחשב centroid: וקטור ממוצע של כל הדוגמאות שבהם השתמשתם לבניית העץ (למשל, אם p=0.5,n=200 השתמשנו ב100 דוגמאות לבניית עץ מסוים. כל "פיצ'ר" בcentroid של עץ זה יהיה ממוצע 100 הפיצ'רים המתאימים ב100 הדוגמאות בהם השתמשנו). לאחר שחישבנו centroid לכל עץ (תוכלו לעשות זאת כבר בזמן האימון) כאשר נרצה לסווג דוגמא חדשה נבחר את K העצים שהcentroid שלהם קרוב ביותר לדוגמא שאנו רוצים לסווג. למדידת מרחק נשתמש במרחק אוקלידי.
   1. עליכם לממש בקובץ KNNForest.py את אלגוריתם הלמידה KNN-decision-tree. בהרצת הקובץ האלגוריתם צריך להתאמן על כל קבוצת האימון, ולהדפיס את הדיוק שלו על קבוצת המבחן ללא כל מלל או הדפסות. עליכם לבחור בעצמכם את כל הפרמטרים השונים לאלגוריתם. מומלץ לבצע ניסויים כדי למצוא ערכי פרמטרים טובים. הסבירו אילו ניסויים ביצעתם, וציינו מהו הדיוק המקסימלי שקיבלתם על קבוצת המבחן. שימו לב שמטרת האלגוריתם הינה להשיג דיוק (רגיל) מקסימלי ולא למזער את פונקציית ה- loss.

עשיתי כוונון לפרמטרים N , K , P בעזרת kfold.

(שוב כמו סעיף 4.3 חלקתי את ה data לחמישה סטים כאשר אחד מהן עבור ה טסט ושאר עבור הלמידה וכל פעם עושים רוטציה בינהם , בתוך כל רוטציה עברתי על כל ערכי N , K , P ולבסוף בחרתי את הכי טובים , זאת אומרת שיש להם את הממוצע של אחוז הצלחה הכי גבוה).



נשים לב שלמשל יש לי K = 7 שהוא יותר גדול מ N=5 כי רציתי לתת לניסוי גם את האופציה לבחור לפי כל העצים.

ערכי K הם אי זוגיים כפי שלמדנו בהרצאה כדי שתהיה הערכה לגבי ה majority.

קיבלתי שהערכים הכי טובים הם :

ערך p קטן , כי כנראה יותר טוב שהעצים ילמדו על דאטא שונה לחלוטין (לעומת זאת אם למשל  
 אז תהיה חפיפה בין כל שני עצים ) וזה גם יגדיל את האפשרות שהעץ ילמד על דאטא שאין בו אף דוגמה רועשת.

בגלל שיש עניין של רנדומליות אז הרצתי את knn\_forest עם פרמטרים אילו 10 פעמים והחזרתי את הדיוק המקסימלי והדיוק הממוצע.

הערך המקסימלי שקיבלתי:



ממוצע 10 הריצות:

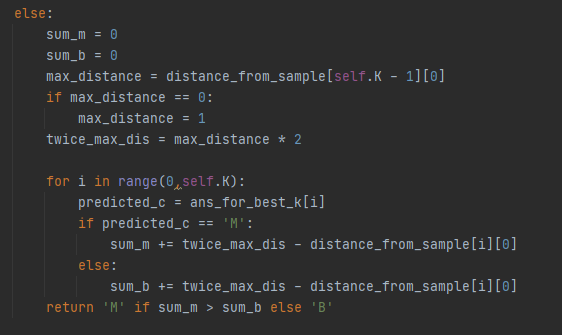


1. (20 נק׳) **שיפור ליער.**
   1. תארו שיפור לאלגוריתם KNN-decision-tree. השיפור צריך להתייחס למקרה הכללי, ולא ללמידה מוכוונת מחיר בלבד.

הבעיה בשיטה הקודמת היא שלא מתיחסים למרחקים בין הסנטריוד לבין הדגימה עצמה , נכון שלוקחים את K הכי קרובים , אבל אם למשל מתוך K אלו יש סינטרויד עם מרחק 0 מהדגימה , לא מתחשבים בזה ונותנים לתשובה שלו משקל כמו העץ עם המרחק הכי גדול (עד K) שיתכן שבכלל המרחק שלו 1000 מהדגימה.

לכן הצעתי את השיפור הבא:

מתוך K הכי קרובים , אתן משקל לתשובה של העצים לפי המרחק שלהם.



קודם כל אני מחשב את המרחק הגדול בין העצים לבין דוגמת המבחן , וכל שהעץ קרוב יותר אני נותן יותר משקל לתשובה שלו.



למשל עץ עם מרחק 0 , אני נותן משקל לתשובה שלו שהוא twice\_max\_distance.

כמה הערות טכניות:

הכפלתי את המרחק המקסימלי פי 2 , כדי גם לתת משקל לעץ האחרון , כי אחרת הוא היה מקבל ניקוד 0 ולתשובה שלו אין משמעות , כמובן דבר זה לא רצוי.

אם המרחק המקסימלי הוא 0 , אז פשוט הגדרתי אותו להיות 1 כדי שהאלגוריתם יעבוד תקין.

תוצאות:

שוב הרצתי 10 פעמים ולקחתי את הממוצע והמקסימום.



יש שיפור קטן (לא הכי משמעותי ☹) אבל בכל זאת שיפור.

אני אציין גם שיש מצב שהשיפור הוא לא כזה משמעותי כי אני הרצתי את האלגוריתם המשופר עבור אותם N,K,P שמצאתי בסעיף 6. לא היה לי את הזמן להריץ שוב את הניסוי כי הרצת הניסוי לוקחת כמה שעות טובות ולבדוק מהם הפרמטרים הכי טובים לאלגוריתם המשופר (אם עושים זאת יש מצב שמקבלים שיפור יותר רציני).

* 1. ממשו הצעתכם בקובץ ImprovedKNNForest.py. על קובץ זה להדפיס בהרצתו את הדיוק שקיבלתם מהרצת האלגוריתם המשופר כאשר הוא לומד על קבוצת האימון ונבחן על קבוצת המבחן. אם ביצעתם ניסויים לקביעת פרמטרים לאלגוריתם שלכם, פרטו זאת. כמו כן צרפו בדו"ח את הדיוק שקיבלתם.

# הוראות הגשה:

* הגשה ביחידים בלבד.
* הגישו קובץ zip בודד המכיל את כל הקבצים שהתבקשתם לממש יחד עם קבצים נוספים כרצונכם בלי קבצי הדאטה שצורפו לכם.
* ודאו שכל הקבצים שלכם מחזירים מה שביקשנו מכם, שאינם פולטים גרפים, ושלא שכחתם להתייחס לנקודות מסויימות.
* נוהל האיחורים מופיע בסילבוס הקורס.

הרבה בריאות והצלחה.