תרגיל בית 3 – פתרון החלק היבש

<u>מגישים:</u>

רוני נודלמן – 209120112

רן ברשינסקי – 208387324

<u>תרגיל 1</u>

הוכחה

תחילה נראה כי פונקציית הנירמול משמרת סדר.

נראה כי מתקיים $x_1 < x_2 \Leftrightarrow Minmax(x_1) < Minmax(x_2)$, ואכן:

$$x_1 < x_2 \Leftrightarrow x_1 - x_{min} < x_2 - x_{min} \Leftrightarrow \frac{x_1 - x_{min}}{x_{max} - x_{min}} < \frac{x_2 - x_{min}}{x_{max} - x_{min}} \Leftrightarrow Minmax(x_1) < Minmax(x_2)$$

תהי תכונה f_i, ונניח כי בשלב בניית העץ, מיון הדוגמאות ע"י התכונה הוא:

$$f_i(x_1) < f_i(x_2) < \cdots < f_i(x_n)$$

כיוון שפונקציית הנירמול משמרת סדר, נקבל כי מתקיים:

$$Minmax\big(f_i(x_1)\big) < Minmax\big(f_i(x_2)\big) < \cdots < Minmax\big(f_i(x_n)\big)$$

נסתכל על $threshold$ כלשהו המוגדר ע"י $t_i = \frac{f(x_i)+f(x_{i+1})}{2}$, ולפי הגדרת החלוקה של הצומת לבנים, נקבל כי הדוגמאות $x_1, x_2, \dots, x_i$ יהיו בבן השמאלי ואילו הדוגמאות $x_{i+1}, x_{i+2}, \dots, x_n$ יהיו בבן הימני. אותה חלוקה לצומת ימני וצומת שמאלי תתרחש כאשר ה-$threshold$ יוגדר ע"י:

$$t_i^{Minmax} = \frac{Minmax\big(f(x_i)\big) + Minmax\big(f(x_{i+1})\big)}{2}$$

לכן, ה-IG עבור כל חלוקה לפי כל t_i^{Minmax} תהיה זהה ל-IG שהיה מתקבל אילו לא היינו משתמשים בפונקציית הנירמול, וזאת כיוון ש-IG מחושב רק ע"י סיווג הדוגמאות ומספרן בכל צומת בן.

לכן, לכל f_i ולכל צומת בתהליך בניית העץ, הערך $\max IG$ יהיה זהה בין אם פונקציית הנירמול מופעלת על התכונות ובין אם לא, ולכן העץ שיתקבל בסוף התהליך יהיה זהה.

לכן, לכל דוגמה בקבוצת המבחן, מסלול הסיווג של הדוגמה יהיה זהה בעץ שבו מופעל הנרמול ובעץ שבו לא מופעל הנרמול, ולכן הדוגמה תסווג באופן זהה. ועל כן, הדיוק ישאר זהה.

<u>תרגיל 2</u>

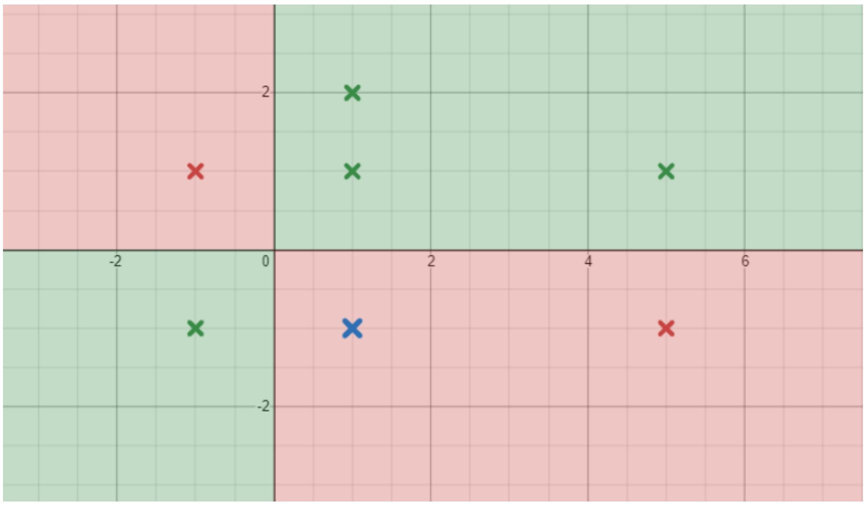
נסמן את הדוגמאות השליליות בצבע אדום.

נסמן את הדוגמאות החיוביות בצבע ירוק.

נסמן את דוגמאות המבחן בצבע כחול.

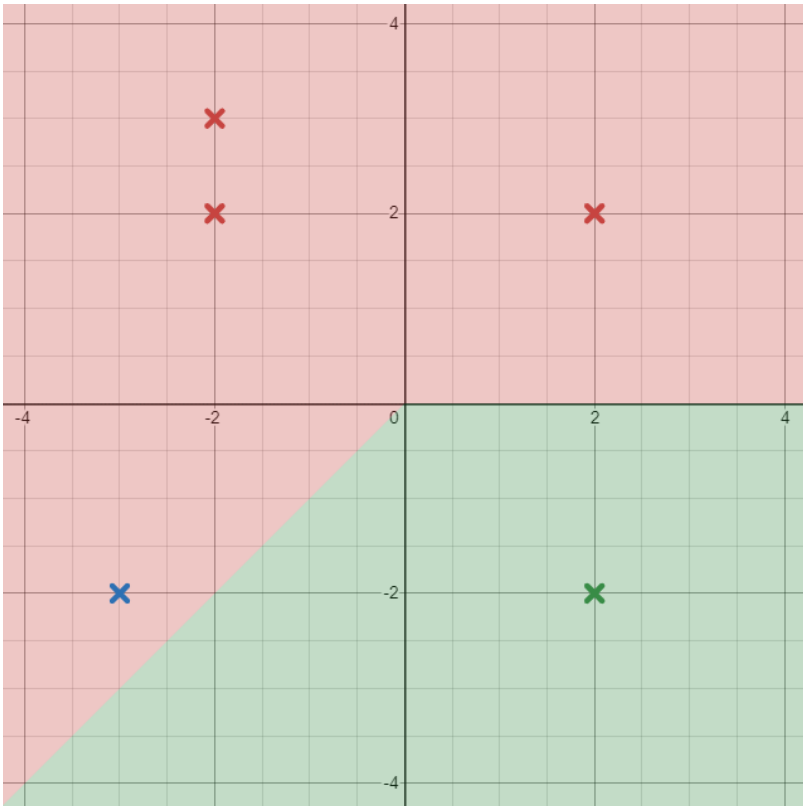
התחומים הצבועים מייצגים את פונקציית המטרה.

א.



בבניית $ID3$, תיבחר תחילה התכונה האופקית שתפצל את השורש ולאחר מכן התכונה האנכית תפצל את שני הבנים של השורש, כך שלבסוף המסווג שיבנה יהיה זהה למסווג המטרה. לעומת זאת, עבור $K = 1$ נקבל כי נקודת המבחן תסווג כחיובית למרות שהיא שלילית.

ב.

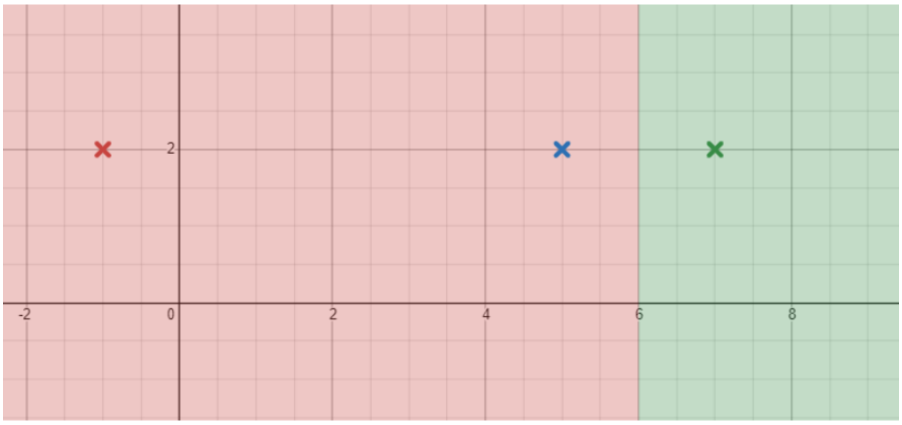


מסווג KNN עבור $K = 1$ אכן יסווג נכון כל דוגמת מבחן, אולם מסווג $ID3$ יניב את המסווג הבא:

$$f_{ID3}(v_1, v_2) = \begin{cases} - & v_2 > 0 \\ + & v_2 \leq 0 \end{cases}$$

ולכן יסווג את נקודת המבחן כחיובית, למרות שהיא שלילית.

ג.

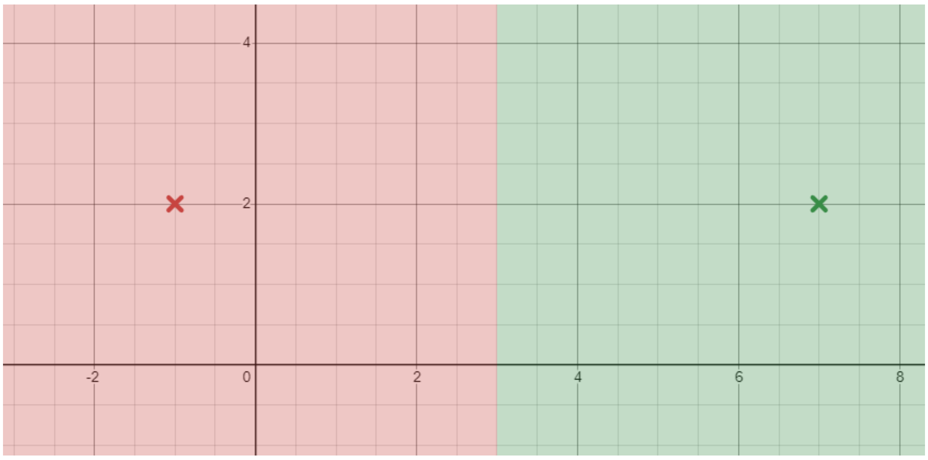


גם בבניית $ID3$ וגם בבניית KNN עבור $K = 1$ נקבל את המסווג:

$$f_{KNN,ID3}(v_1, v_2) = \begin{cases} + & v_1 > 3 \\ - & v_1 \leq 3 \end{cases}$$

ולכן נקודת המבחן תסווג כחיובית ע"י שני המסווגים למרות שהיא שלילית.

ד.



גם בבניית $ID3$ וגם בבניית KNN עבור $K = 1$ נקבל כי המסווגים שיתקבלו יהיו זהים למסווג המטרה:

$$f(v_1, v_2) = \begin{cases} + & v_1 > 3 \\ - & v_1 \leq 3 \end{cases}$$

תרגיל 3

א. לפי $Majority\ Classifier$, מכיוון שיש 5 דוגמאות עם סיווג 1 ו-5 דוגמאות עם סיווג 0, אז הסיווג של כל ערך ממשי יהיה 1. 5 דוגמאות אכן מסווגות עם סיווג 1, ויקבלו דרך המסווג הזה סיווג 1, ו-5 מהן מסווגות עם סיווג 0 אך יקבלו את הסיווג 1 דרך המסווג הזה. לכן הערך הדיוק יהיה 0.5.

ב. כאשר הקבוצה הראשונה היא קבוצת האימון, המסווג יתן את הסיווג 1 לכל אחת מדוגמאות הקבוצה השנייה. כיוון שבקבוצה השנייה רק אחת מתוך 5 הדוגמאות אכן מסווגת עם הסיווג 1 אז במקרה זה ערך הדיוק יהיה 0.2.
כאשר הקבוצה השנייה היא קבוצת האימון, המסווג יתן את הסיווג 0 לכל אחת מדוגמאות הקבוצה הראשונה. כיוון שבקבוצה הראשונה רק אחת מתוך 5 הדוגמאות אכן מסווגת עם הסיווג 0 אז במקרה זה ערך הדיוק יהיה 0.2.
נקבל כי ממוצע ערך הדיוק ע"י הרצת $2-fold\ Cross\ Validation$ יהיה 0.2.

תרגיל 5 – סעיף ב'

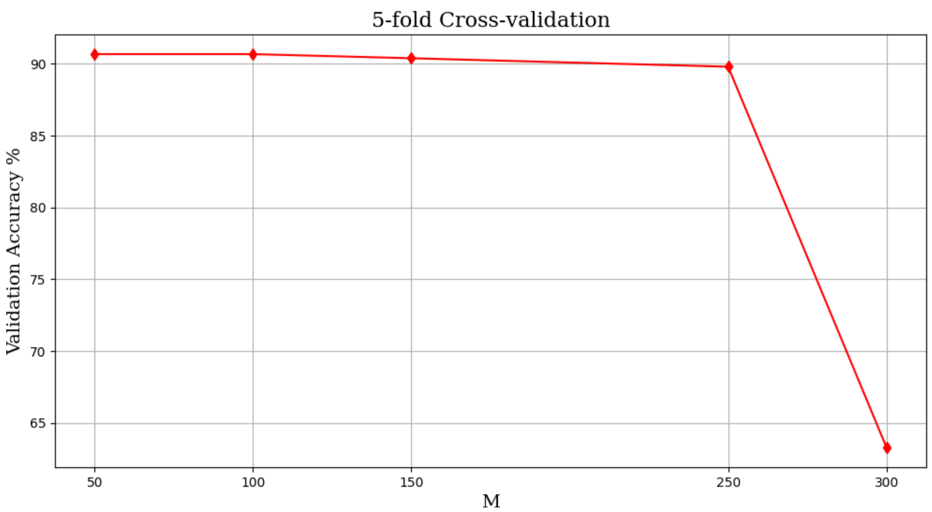
ערך הדיוק שהתקבל הוא 94.69%.

תרגיל 6 – סעיף א'

הגיזום מנסה למנוע את התופעה $overfitting$, שמביאה להקטנת שגיאת האימון אך כנראה מגדילה את שגיאת המבחן. תופעה זו יכולה להיווצר ע"י דוגמאות אימון רועשות, מידע חסר ומחסור בדוגמאות. הגיזום מונע קבלה של עצים גדולים שנוצרים ע"י סיווגים רבים מדי, וזאת ע"י פיצול צמתים עם סף דוגמאות מסוים.

תרגיל 6 – סעיף ג'

ג. להלן הגרף:



ד. לפי הגרף, נראה כי כאשר בונים את העץ $ID3$ עם גיזום באמצעות פרמטר $M \cong 100$, דיוק האלגוריתם גבוה – כ-90%. אולם כאשר הפרמטר גבוה מדי ($M = 300$) הגיזום מונע פיצול חיוני של צמתים ועל כן שגיאת המבחן עולה משמעותית.
כאשר $M = 100$ קיבלנו את הדיוק הגבוה ביותר שהוא 90.68%.

<u>תרגיל 6 – סעיף ד'</u>

לאחר הרצת האלגוריתם $ID3$ עם הגיזום המוקדם עם הפרמטר $M = 100$, קיבלנו כי הדיוק הינו 97.35%.
אכן כצפוי, הגיזום העלה את הדיוק של העץ $ID3$ וצמצם משמעותית את שגיאת המבחן, וזאת ע"י מניעת התופעה של $Overfitting$.

<u>תרגיל 7</u>

בשלב האימון האלגוריתם שומר את קבוצת האימון בזיכרון. בשלב הסיווג האלגוריתם מסווג דוגמה לפי מחלקת הרוב של K הנקודות הקרובות ביותר לאותה דוגמה, כאשר נקודות אלו נלקחות מקבוצת האימון.
המרחק בין הדוגמה לנקודות קבוצת האימון נקבע ע"י הגדרת מרחק של התכונות.
יתרונות האלגוריתם הם שלב אימון מהיר שלא דורש חישוב, ופשטות המימוש של האלגוריתם.
חסרונות האלגוריתם הם זמן חישוב יקר בשלב הסיווג ורגישות לתכונות לא רלוונטיות.

<u>תרגיל 8</u>

א. מספר כל תתי הקבוצות של S הוא $2^{|S|}$.

ב. מספר כל תתי הקבוצות של הקבוצה S בגודל b הוא $\binom{|S|}{b}$.

<u>תרגיל 9</u>

א. יש לבדוק את הביצועים על קבוצת המבחן, שבה לא השתמשנו כלל על מנת לבחור את תת הקבוצה האופטימלית של המאפיינים, שהרי קבוצת הדוגמאות שבאמצעותה מצאנו את תת הקבוצה האופטימלית של המאפיינים הינה קבוצת האימון.

ב. גודל תת הקבוצה האופטימלית של המאפיינים שקיבלנו הינה 4.
כאשר מריצים את האלגוריתם KNN, ללא סינון של מאפיינים, מקבלים כי דיוק האלגוריתם עבור קבוצת המבחן הינו 75.5%.
לעומת זאת, כאשר מריצים את האלגוריתם KNN עם תת הקבוצה של המאפיינים האופטימלית שקיבלנו, מקבלים כי דיוק האלגוריתם עבור קבוצת המבחן הינו 79%.
אכן, דיוק האלגוריתם גדל כאשר צמצמנו את קבוצת המאפיינים, ובנוסף לכך, שלב הסיווג נעשה מהיר יותר. זאת כיוון שבעת יש לחשב מרחקים בין דוגמאות במרחב קטן יותר, שהרי כעת יש פחות מאפיינים לכל דוגמה.

ג. נסביר את האלגוריתם שמימשנו:

1. לוקחים את קבוצת האימון ואת קבוצת המאפיינים ונניח כי גודל קבוצת המאפיינים הוא n.
2. לכל $i = 1,2, \dots, n$:
   מריצים $K - fold\ cross\ validation$ על קבוצת האימון, כאשר מסירים מקבוצת האימון את המאפיין ה-i, ושומרים את הדיוק שהתקבל.
3. בודקים עבור איזה i קיבלנו את הדיוק הגבוה ביותר ונסמנו i_{max}.
4. מוציאים את המאפיין ה-i_{max} מקבוצת המאפיינים.
5. אם גודל קבוצת המאפיינים הינו b אז מסיימים, ומחזירים את קבוצת המאפיינים שהתקבלה.
6. אחרת, חוזרים לשלב 1.

נסביר את הסיבה לכך שבשלב 4 מוציאים את המאפיין ה-i_{max} מקבוצת המאפיינים.
בכל אחת מהאיטרציות בשלב 2, אנחנו מריצים $K - fold\ cross\ validation$ על קבוצת האימון ללא המאפיין ה-i. אם קיבלנו דיוק גבוה באיטרציה מסוימת, משמע המאפיין ה-i לא היה נחוץ לנו שהרי התקבל דיוק גבוה ללא כל שימוש בו.
לכן, המאפיין שהוצאתו גרמה לקבלת הדיוק הגבוה ביותר, הוא המאפיין שתרם הכי פחות להגדלת הדיוק של האלגוריתם, ולכן נוציא אותו.

למעשה, אנחנו מבצעים חיפוש לוקאלי באופן חמדני: אנחנו מתחילים עם קבוצת המאפיינים המלאה, ובכל שלב בוחרים את המאפיין שתורם הכי פחות (כלומר המאפיין שבלעדיו, דיוק האלגוריתם עדיין גבוה). מוציאים מאפיין זה, וכך ממשיכים עד לקבלת תת קבוצה מהגודל הרצוי.

ננתח את סיבוכיות הזמן של האלגוריתם:
נניח כי סיבוכיות הזמן של $K - fold\ cross\ validation$ הינה t.
מספר הפעמים שחוזרים על שלבים $1,2, \dots, 6$ הינו $D - b$.
מספר האיטרציות בשלב 2 הינו כגודל תת הקבוצה של המאפיינים הנוכחית.
סיבוכיות הזמן של השלבים האחרים זניחה ביחס לשלב 2.
בסך הכל נקבל:

$$T = \sum_{k=b+1}^{D} O(k \cdot t) = t \cdot \sum_{k=b+1}^{D} O(k) \leq t \cdot \sum_{k=1}^{D} O(k) = t \cdot O(D^2)$$

בסך הכל קיבלנו כי סיבוכיות זמן הריצה של האלגוריתם הינה ריבועית בגודל קבוצת המאפיינים.
נשים לב כי חיפוש $brute\ force$ מתבצע בסיבוכיות זמן אקספוננציאלית $O(2^D)$, ועל כן מצאנו אלגוריתם יעיל שאכן מעלה את דיוק האלגוריתם ומקטין את הזמן של שלב הסיווג.