מטלה 3 – adabost: תז הגשה- 319092201

רקע למטלה:
במטלה זאת קיבלנו 150 נקודות דו ממדיות כאשר כל 2 נקודות מהוות קו (נוסחה למציאת קו בין שני נק (x1,y1) , (x2,y2) מציאת שיפוע: (y1-y2)/(x1-x2)=m
מציאת bais היא: y1=m*x1+b)

נחלק את הדאטה ל train/ test כך שכל אחד יהווה 50% מהנקודות כלומר 75 נקודות בכל חלוקה ואזי מספר החוקים (הקווים) ב test יהיו 5550.

מטרה: מציאת 8 חוקים הטובים ביותר.

עזרים: אלגוריתם adaboost למציאת 8 חוקים אילו.

תיאור מהלך העבודה:

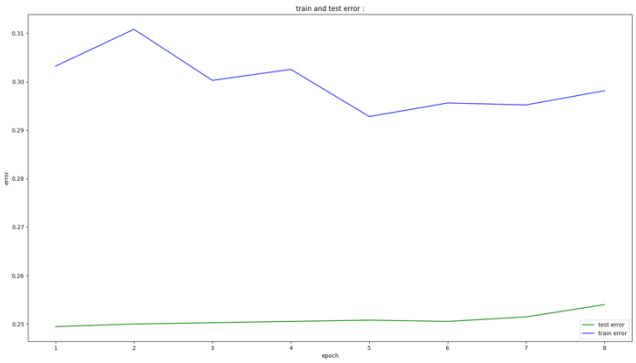
1) קיבלתי קובץ עם 150 נקודות , נבצע קריאה לקובץ , וסידור 150 הנקודות בשני list בהתאמה : points list, labels list.
2) לאחר מכן נבצע 100 איטרציות לאלגוריתם adaboost:
3) אלגוריתם: - נחלק את הדאטה לtrain/test בגודל 50% מהדאטה כל אחד מהם
מטרה- מציאת החוק בעל הטעות הנמוכה ביותר ועידכון המשקלים לפיו:
   - נעבור בfor לקבלת 8 חוקים
   - נעבור ב for על כל החוקים (קוים) שיצרתי מלמעלה
   - נעבור ב for על כל הנקודות של x_train
   - לאחר מכן נעבור על כל נקודה ובעזרת הפונקציה find_h_t נמצא את הסיוג המתאים
   - במידה ולא הצלחנו נעדכן את המשקל של weigth_error
   - נזכור את החוק בעל המשקל הקטן ביותר
   - נצא את alpha_t ואת המשקל ונעדכן לפיו את שאר המשקלים.
   - תוך כדי נמצא גם את test_err_result , train_err_result.
4) בכל 10 איטרציות נדפיס את הממוצע של train_error, test_error
5) לבסוף אדפיס גרף שייצג לי את הtrain/test error

(השוני בין שלב 4 ל5 הוא שבשלב 4 אני עושה ממוצע על הtrain/test error ללא התייחסות למספר החוקים אלא למספר האיטרציות ואילו בשלב 5 אני עושה את הממוצע על מספר החוקים כלומר ממוצא של כל חוק 1 של כל 2חוקים וכן הלאה...)

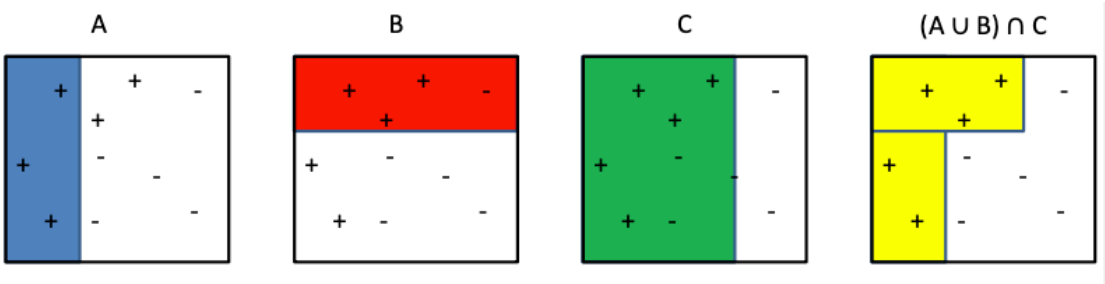
צילום שלב 4):

```
train errors mean in iterations:  0  :  0.24166666666666667
test errors mean iterations:  0  :  0.2833333333333333
--------------------------------
train errors mean in iterations:  10  :  0.2569696969696972
test errors mean iterations:  10  :  0.2939393939393938
--------------------------------
train errors mean in iterations:  20  :  0.2572222222222225
test errors mean iterations:  20  :  0.3093650793650791
--------------------------------
train errors mean in iterations:  30  :  0.2533333333333337
test errors mean iterations:  30  :  0.3072043010752685
--------------------------------
train errors mean in iterations:  40  :  0.2532520325203253
test errors mean iterations:  40  :  0.3102439024390243
--------------------------------
train errors mean in iterations:  50  :  0.25013071895424843
test errors mean iterations:  50  :  0.30499999999999994
--------------------------------
train errors mean in iterations:  60  :  0.25117486338797845
test errors mean iterations:  60  :  0.30494535519125693
--------------------------------
train errors mean in iterations:  70  :  0.25072769953051677
test errors mean iterations:  70  :  0.29985915492957754
--------------------------------
train errors mean in iterations:  80  :  0.24938271604938306
test errors mean iterations:  80  :  0.3008847736625517
--------------------------------
train errors mean in iterations:  90  :  0.24846153846153887
test errors mean iterations:  90  :  0.30239926739926787
--------------------------------
```

צילום שלב 5):



ניתוח התנהגות האלגוריתם :
נשים לב כי בחישוב הtrain/test erorr אשר מחושב במהלך ריצת האלגוריתם נראה שהוא במצב של עליה וירידה עליה וירידה... בעת הוספת חוקים (ניתן לראות זאת בצילום שלב 5 של המטלה) ונראה כי זה חל גם בחישוב הטעות בשלב האימון וגם בשלב הבדיקה.
לאחר חלוקת הדאטה לאימון ובדיקה – אנו מריצים את האלגוריתם על שלב האימון ונעבור על כל החוקים עד לקבלת החוק בעל הטעות הנמוכה ביותר לאחר מכן נעדכן את המשקלים כך שהמשקלים של הנקודות שלא זוהו כנכונות יקבלו משקל גדול יותר ואילו הנקודות שקיבלו זיהוי נכון יתעדכן משקלם למשקל קטן יותר – דבר זה יצור את המצב הבא : שבשלב הבא כאשר נחפש חוק נוסף עם הטעות הקטנה ביותר הוא "יתפוס" כנראה יותר חוקים שלא נתפסו קודם ואז נחזור על התהליך של עדכון המשקלים .
וזאת במטרה למצוא כמה שיותר חוקים אשר יביאו לנו סיווג נכון לחוקים השייכים ל (-1) ולאילו ששייכים ל(1)
מצרפת תמונה שנלמדה בהרצה שמשקפת את התהליך שהסברתי למעלה :



בתמונה , החוקים הם a,b,c וניתן לראות שכאשר יש בינהם "שיתוף" הם מצליחים לסווג בצורה האופטילית.

ותהליך זה שהסברתי והציור מסביר את ההתנהגות של הtrain/test error שקיבלנו – שבו באוסף החוקים נראה תנודות של עליה וירידה בחישוב הטעות.

התאמת יתר- overfitting – שבה המודל מותאם יתר על המידה לאוסף מסוים של נתונים (למשל האוסף שהיה זמין לשם אימונו) ועל כן מצליח פחות בביצוע תחזיות. התאמת יתר מתרחשת כאשר המודל נקבע על ידי יותר פרמטרים מאשר הנתונים מצדיקים. עודף הפרמטרים מאפשר למודל ללמוד את הרעש הסטטיסטי כאילו הוא מייצג התנהגות אמיתית.(ויקיפדיה)
ניתן לראות לפי צילום תמונה של שלב 4 ושל שלב 5 כי במקרה שלנו ובחישוב מודל זה אין התאמת יתר , והמודל שלנו אכן מצליח יותר בבוצוע התחזית לאחר אימון על הדאטה.