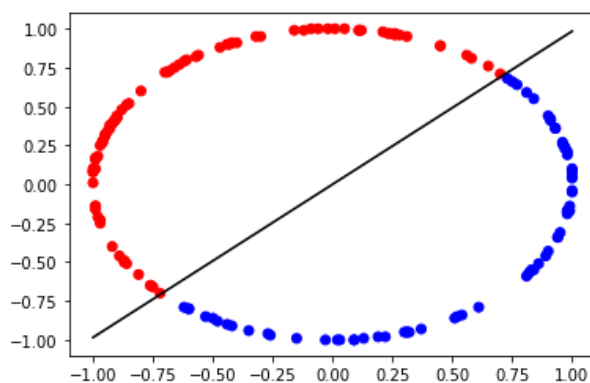


מטלה 3 – למידת מכונה

מגישים: איתי רפיעי, אלמוג יעקב מעטוף
(מצ"ב קבצי קוד עבור כל שאלה בהתאמה)

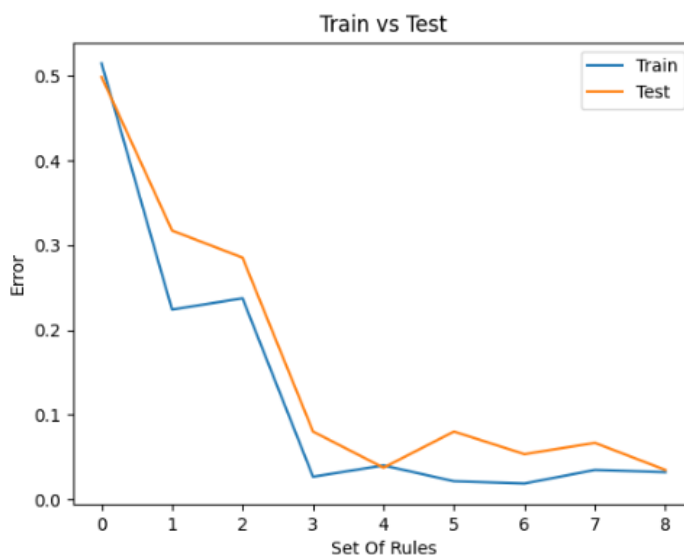
שאלה 1:

בהרצת אלגוריתם perceptron על הסט הנתון קיבלנו וקטור משקלים: $[-2.46 \ 2.5]$
מספר הטעויות שהתקבלו במהלך האלגוריתם הינו 22
כזכור, וקטור המשקלים המייצג את הקו הינו מנח לקו.
הפלט עבור הסט הנ"ל הינו:



שאלה 2:

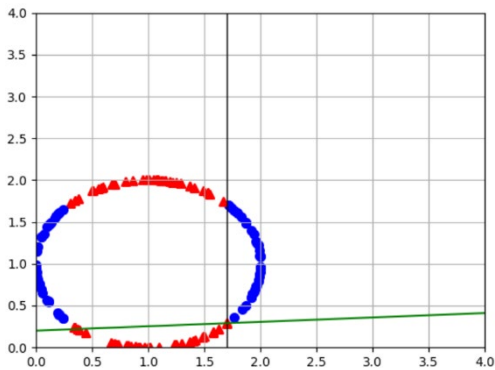
1. נתח את ההתנהגות של Adaboost על ה- train וה- test . האם יש התנהגות יוצאת דופן:



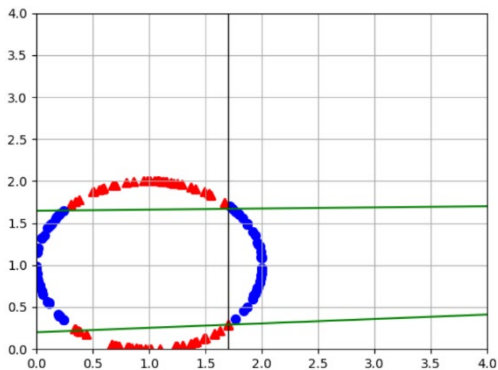
(ניתן לראות בתמונה למעלה את הגרף המתאר את השגיאות של ה- Train וה- Test עבור קבוצות של חוקים בגודל 1 עד 8)
התוצאות:

	0	1	2	3	4	5	6	7	8
Train	0.527	0.213	0.242	0.04	0.034	0.02	0.022	0.034	0.037
Test	0.485	0.338	0.290	0.08	0.024	0.102	0.053	0.077	0.041

לאחר שהרצנו את האלגוריתם כמה פעמים עבור קבוצת הנקודות הנתונה הבחנו כי מתקיים:



א. כבר בשלושה חוקים אנו מקבלים תוצאה דיי טובה גם עבור Train וגם עבור Test.
עבור קבוצת הנקודות שלנו ידוע כי מספיק שני קווים כדי להפריד בין הנקודות אך האלגוריתם מביא לנו את שני הקווים הראשונים בצורה אנכית כמו בתרשים הבא:
(עבור 2 קווים בTest מתקבל שגיאה של 0.29)



ורק בעזרת הקו השלישי נקבל את ההפרדה הרצויה:
(עבור 3 קווים בTest מתקבל שגיאה של 0.08)

ב. בנוסף ניתן לראות כי ברוב קבוצות הקווים אכן השגיאה של Train יותר נמוכה מהשגיאה של Test.

2. האם יש overfitting?

Overfitting יכול להיגרם משתי סיבות:

- כאשר החוקים שקיבלנו יהיו מותאמים במדויק עבור Train אז גם עבור השינוי הכי קטן בTest נקבל שגיאה גדולה.
- כאשר משתמשים בהרבה חוקים כך שהחוק החדש הוא חוק קשה שמתאים במיוחד לTrain אך Test הוא יפספס חלק מהנקודות וכך השגיאה תגדל.

ניתן לראות בריצה אצלנו כי עבור קבוצה של קו אחד נראה כי מתקיימת הסיבה הראשונה ל'overfitting', החוק לא מסובך אך נראה שהוא מתאים במיוחד עבור Train לאומת Test (הפרש בין השגיאות $\approx 10\%$)

בנוסף ניתן לראות כי עבור קבוצה של 5 קווים נראה כי מתקיימת הסיבה השנייה ל'Overfitting', חמישה קווים זה כבר חוק מסובך יחסית שעובד דיי טוב עבור Train אך מגדיל את השגיאה של Train לאחר שגיאה יחסית נמוכה עבור 4 קווים. (גם כאן הפרש בין השגיאות $\approx 9\%$).