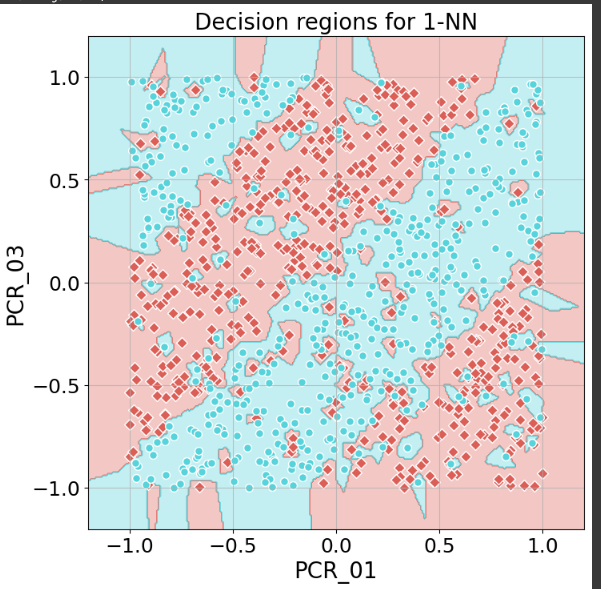
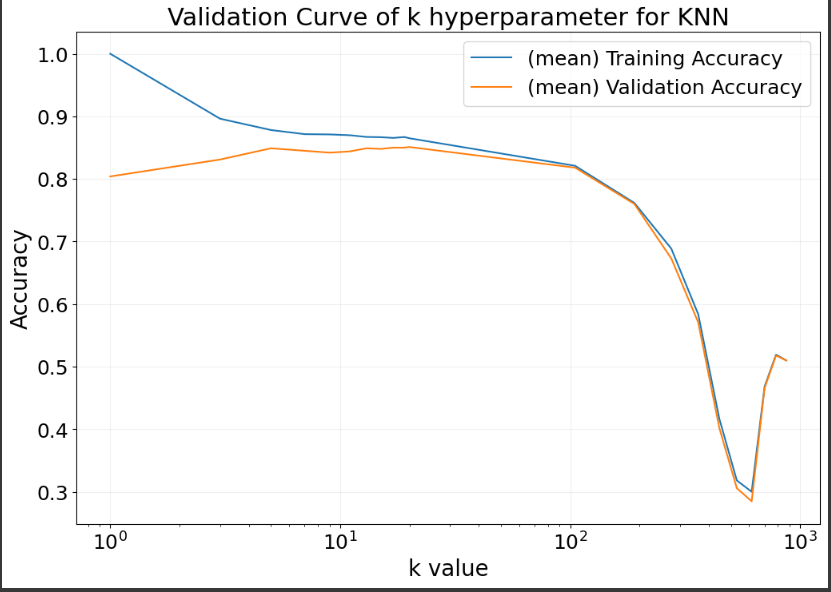
**מבוא למערכות לומדות- תרגיל בית 2- דו״ח עבודה**

**מגישים:   
  
גל קסטן 316353176**

**חן פרי – 313283657**

(Q1)



(Q2)

שתי העקומות מציגות לנו לכל k את מידת הדיוק הממוצעת מתהליך הcross-validation שהתקבלה מאימון של מודלי knn עם k שכנים.

אנחנו רוצים לבחור את הk שנתן לנו את הדיוק הכי טוב על הvalidation set (כיוון הvalidation set נותן לנו הערכה על ביצועי המודל על מידע שלא ראה). לכן, ניקח את הנקודה המקסימלית על העקומה הכתומה ונמצא את הk שנותן לנו את הדיוק המקסימלי.

עשינו את זה כמובן בעזרת קוד פייתון וקיבלנו :

A screenshot of a computer

Description automatically generated

כלומר הk הטוב ביותר שמצאנו הוא k=20.

**ערכי k נמוכים גורמים לתופעת הoverfitting.** ניתן לראות בגרף הvalidation curve כי עבור ערכי k נמוכים מידת הדיוק על קבוצת האימון היא גבוהה ומידת הדיוק על קבוצת הוולידציה נמוכה יותר. בתרגול ראינו כי מצב כזה מקושר לoverfit, כלומר דיוק טוב מדי על הtraining set הוא לא בהכרח טוב ואף יכול לפגוע ביכולת ההכללה על הtest set.

ראינו בשיעורי בית הקודמים שכאשר מס׳ השכנים ב KNN הוא קטן , מתרחשת תופעת הoverfitting מאחר ויש משקל גדול יותר למספר קטן של השכנים הקרובים ביותר ולכן ההשפעה של כל שכן קרוב גדולה יותר. כתוצאה מכך האלגוריתם רגיש לרעש, ובמידה ויש נקודות חריגות בtraining set עלול להיווצר אזור סיווג שגוי מסביב לנקודות הרעש, כפי שרואים גם בplot בשאלה 1.

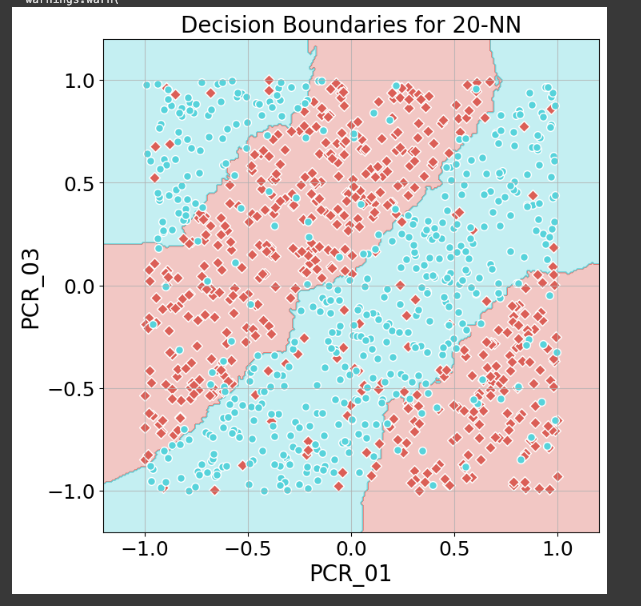
**ערכי k גבוהים גורמים לתופעת הunderfitting.** ניתן לראות בעקומה שלנו כי ככל שk גדל אנחנו מקבלים שהדיוק הולך וקטן והעקומות מתכנסות לאותה נקודה של דיוק נמוך מאוד.

תופעה זו קוריתמאחר וכאשר מספר השכנים גדול מדי התיוג של הנקודות מושפע גם משכנים רחוקים שלא רלוונטים לסיווג של אותה הנקודה. כתוצאה מכך, נוצר מצב שההשפעה של הנקודות הרלוונטיות (הקרובות יותר) יורדת ונקודות חסרות חשיבות נלקחות גם כן בחשבון. אם כמותן גדולה אז השפעתן יכולה להעפיל על המשקל של הנקודות הקרובות יותר ולייצר סיווג שגוי.

כתוצאה מכך, נוצר מצב של חוסר התאמה של המודל לtraining set וגם לtest set.

ככל שk גדל, המודל נעשה כללי מדי ואינו מסתגל היטב לניואנסים של קבוצת האימון. מכיוון שהוא לא מתאים, התחזיות של המודל אינן מושפעות במידה רבה מהמאפיינים של נתוני האימון, מה שמוביל לציוני דיוק דומים ונמוכים גם באימון וגם במבחן.

(Q3)

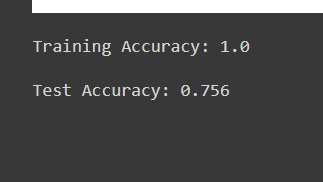


A black background with white text

Description automatically generated

(Q4)

התוצאות של המודל 1NN בסעיף 1 הן :



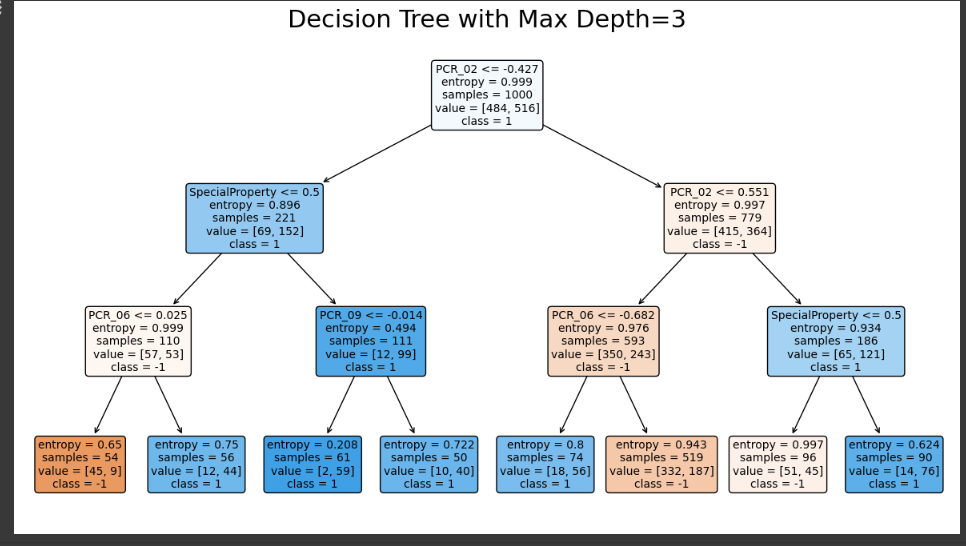
ניתן לראות שהדיוק של קבוצת האימון הוא גבוה בעוד שהדיוק של קבוצת המבחן נמוך. בנוסף גבולות ההחלטה של k=1 מתאימות באופן מושלם לקבוצת האימון. לכן, עבור k=1 מתרחשת תופעת הoverfitting, כלומר למודל אין יכולת הכללה מספיק טובה על קבוצות שונות מקבוצת האימון, והוא מתאים את עצמו לקבוצת האימון (מה שיצור שונות גבוהה של המודל על דאטה סטים שונים).

גבולות ההחלטה של המודל שלנו עם k=20 הם ״חלקים יותר״ ומצליחים לתפוס טוב יותר את ארבע אזורי ההחלטה העיקריים של הדאטה. בנוסף, אומנם מידת הדיוק על קבוצת האימון של המודל קטנה יותר אבל מידת הדיוק על קבוצת המבחן גבוהה יותר. ממצאים אלו מחזקים את הבחירה שלנו בפרמטר שמצאנו בתהליך הtuning ומצביעים על כך שמצאנו k באזור ה”sweet spot” שמאזן בין bias לvariance .

(Q5)

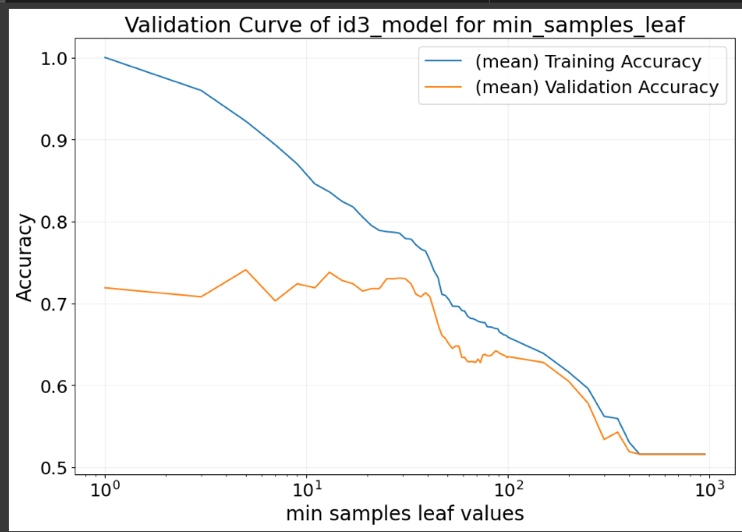
A grey background with white text

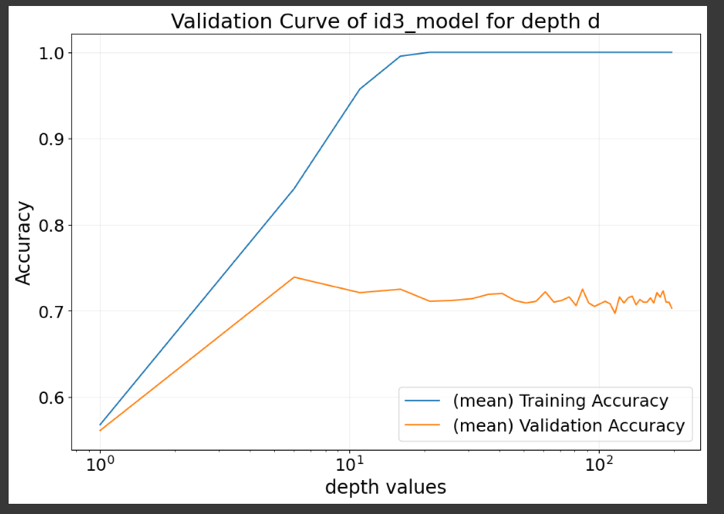
Description automatically generated



(Q6)

תחילה התייחסנו לפרמטרים כבלתי תלויים אחד בשני ובדקנו עבור כל פרמטר מתי מתקבל דיוק מקסימלי כפי שעשינו בשאלה 2 וזאת כדי להבין פחות או יותר הטווחים שכדאי להשתמש בהם.

התוצאה עבור max\_depth, והתוצאה עבור min\_samples\_leaf:



A black screen with white text

Description automatically generatedA screen shot of a computer

Description automatically generated

ניתן לראות כי עבור min\_samples\_leaf – תחילה יש הפרש גדול בין מידת הדיוק על קבוצת האימון לקבוצת המבחן (overfit ) , לאחר מכן יש התכנסות של מידת הדיוק על האימון לכיוון מידת הוולידציה ובאזור של כמות דגימות גדולה מ-40 מתחיל להיות underfit (ירידה הן של הדיוק על האימון והן על הוולידציה).

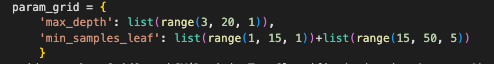
עוד נשים לב כי עם העלייה בכמות הדגימות בעלה באזור 0-40 אין עליה משמעותית ביכולת ההכללה של המודל ובמידת הדיוק על הוולידיציה (למרות שמידת הדיוק על האימון יורדת ויוצאים מאזור הoverfit). עם זאת, נראה כי באזור של 5-6 דגימות בעלה ובאזור 13-14 דגימות בעלה , מקבלים מידת דיוק מעט גדולה יותר על קבוצת הולידיציהץ.

ניתן לראות עבור max\_depth כי בעומקים נמוכים יכולת ההכללה של המודל היא נמוכה (מידת דיוק נמוכה הן על האימון והן על הוולידציה). ככל שעולים עד עומקים של כ5-7 מידת הדיוק על האימון והולידיציה עולות, כאשר לאחר עומקים אלו מגיעים לאזור של overfit- מידת הדיוק על האימון ממשיכה לעלות עד למידת דיוק מושלמת באזור ה-20 ואילו מידת הדיוק על הוולידיציה נשארת פחות או יותר יציבה.

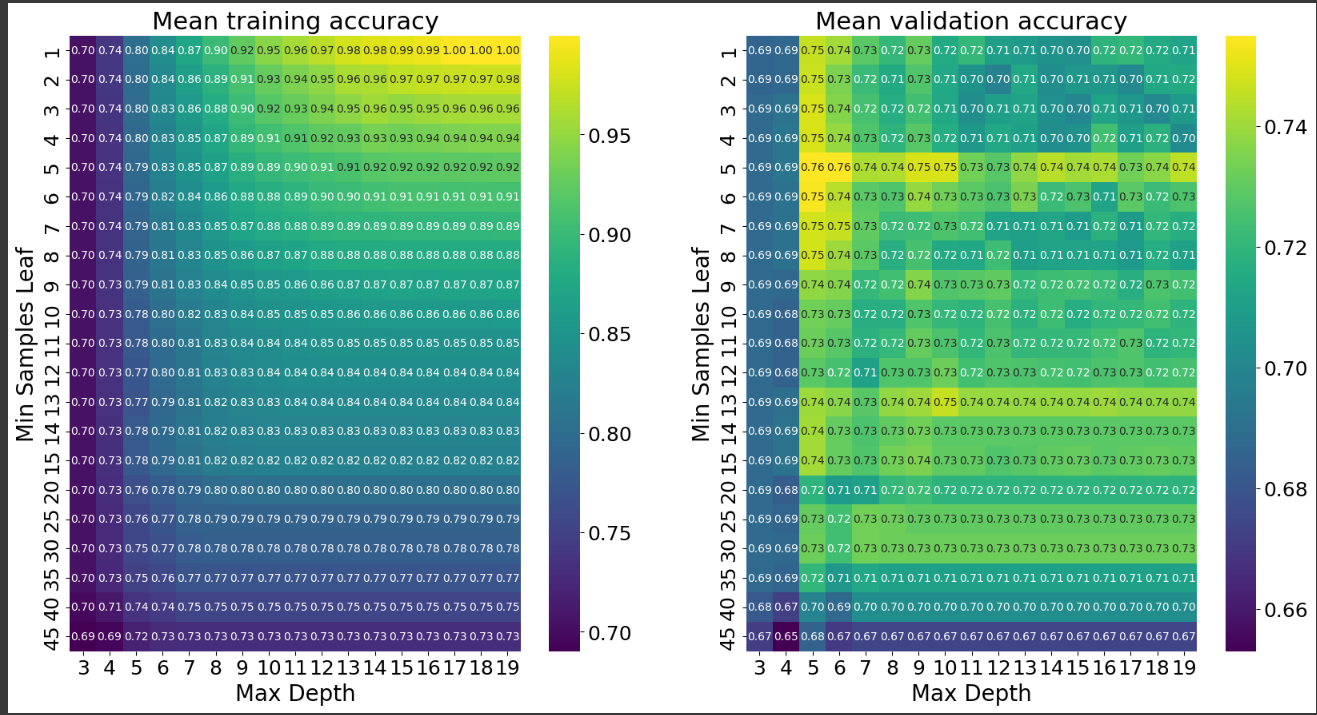
1. לאחר החיפוש הראשוני בחרנו בשני טווחים:

עבור הmax-depth נראה שאנחנו יכולים לתפוס את ה”sweet spot” באזור 3-20 (אזורים של עומק 1-2 הם נמוכים מדי ואזורים מעל 20 הם כבר ממש באזור הoverfit).

עבור הmin\_samples leaf- - בחרנו לעשות חיפוש מדויק יותר על הטווח של 1-15 וחיפוש מעט פחות מדויק על הטווח 15-45.



**ב+ג.**



**קומבינציה אופטימלית:**

****

**ד+ו.**

**קומבינציה שגורמת לunder fitting:**

Max\_depth=3

Min\_samples\_leaf=45

עבור קומבינציה זו קיבלנו מידת דיוק נמוכה הן על קבוצת אימון והן על קבוצת המבחן.

סיבה לתופעת הunderfitting על קומבינציה זו- התופעה מתרחשת כאשר למודל אין יכולת לתפוס את קשרים מורכבים בדאטה. ערכים מסוימים של היפר-פרמטרים מסוימים עשויים למנוע ממנו ללמוד ניואנסים מסוימים שקיימים בקבוצת האימון ולכן מקבלים מודל פשוט מדי שלא מתאים לדאטה. כאשר עומק העץ קטן, המודל לא יכול לבצע מספיק החלטות כדי לבצע פרדיקציה ולכן לא יכול לחלק את הדאטה למספיק קבוצות (למשל עץ בגובה 3 יחלק את הדאטה לכל היותר ל8 קבוצות). באותו אופן, כאשר המספר המינמלי שעלה יכול להחזיק הוא גדול , תהליך יצירת העץ יעצור גם כאשר האנטרופיה בעלה גבוהה יחסית וזה לא יאפשר לעץ לבצע מאוד נוספים הדרושים על מנת ללמוד קשרים עדינים בדאטה.

**ה+ו.**

**קומבינציה שגורמת לOverfit fitting:**

Max\_depth=19

Min\_samples\_leaf=1

עבור קומבינציה זו קיבלנו מידת דיוק גבוהה על קבוצת אימון ומידת דיוק נמוכה על קבוצת המבחן.

סיבה לתופעת הOverfitting על קומבינציה זו-

תופעה זו מתרחשת כאשר המודל מתאים את עצמו בצורה יותר מדי טובה לקבוצת האימון, מה שפוגע ביכולת ההכללה שלו על חיזוי נקודות שהוא לא התאמן עליהם.

max\_depth מאפשר לחלק את הדאטה לקבוצות יותר מדי קטנות (למשל עבור גובה עץ 19 ניתן לחלק את הדאטה ל קבוצות. כתוצאה מכך, יש רגישות -יתר לרעש בדאטה והתאמה יותר מדי מושלמת לדאטה סט אחד , כך שיכולת ההכללה כבר נפגעת.

באופן דומה, מס׳ דגימות קטן בעלה, מאפשר למודל לייצר קבוצות אפילו עבור דגימות בודדות, שעשויות להיות רעש, ולכן גם במקרה נוצרת התאמת- יתר ופגיעה נוספת ביכולת ההכללה.

(Q7)

מספר הקומבינציות שנבדקו שווה למכפלת מספר הערכים האפשריים לכל פרמטר.

לפרמטר Max\_depth הטווח הוא מ-3 ל-20 (לא כולל )ולכן יש 17 ערכים אפשריים.

לפרמטר Min\_samples\_leaf הטווח הוא מ1 ל15 בהפרשים של 1 ומ15-45 בהפרשים של 5 ולכן יש 21 ערכים אפשריים.

סה"כ יש 357 קומבינציות אפשריות (17\*21).

במידה והיינו מכווננות פרמטר נוסף, מספר הקומבינציות היה גדל פי מס׳ הערכים שהיינו בוחנות עבור פרמטר זה. למשל אם היינו בוחנות 10 ערכים אפשריים עבור הפרמטר השלישי , היינו מקבלות 3570 קומבינציות.

באופן כללי, בכל פעם שמוסיפים היפר -פרמטר נוסף למרחב החיפוש , מס׳ הקומבינציות שאנו בוחנים גדל באופן מעריכי (עם כל היפר-פרמטר מכפילים את מס׳ הקומבינציות במס׳ האפשרויות עבור ההיפר פרמטר החדש). לכן, הוספת פרמטר נוסף תגדיל את העלות החישובית ואת הזמן הנדרש למציאת השילוב האופטימלי.

(Q8)

לאחר אימון המודל עם ההיפר פרמטר האופטימליים שמצאנו (min samples leaf=5, max\_depth=6),

קיבלנו את התוצאות הבאות:

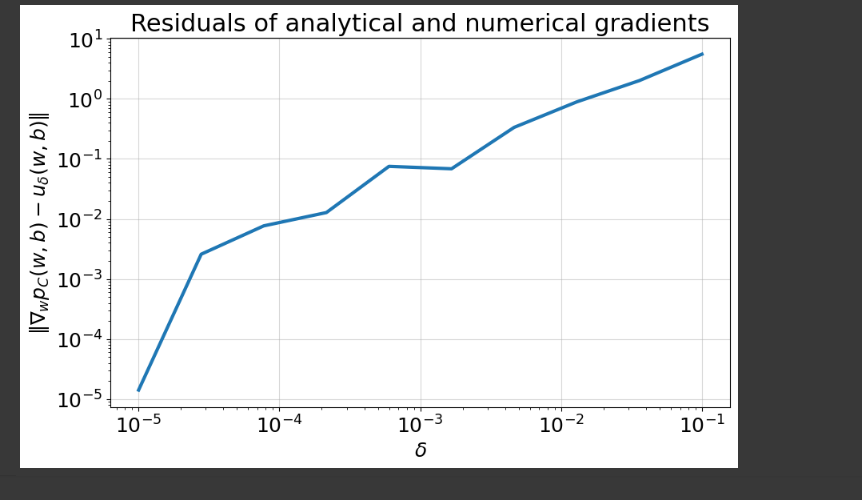


(Q9)

הגדרת הנגזרת כפי שלמדנו בקורסים קודמים משאיפה את לאפס היא .

החישוב הנומרי שהוצג לנו הוא קירוב לנגזרת החלקית לפי . ככל ש קטן נצפה שהחישוב הנומרי יהיה יותר מדויק ויתקרב לנגזרת האמיתית ולכן החישוב האנליטי והנומרי צפויים להיות קרובים יותר.

העקומה הבאה מראה ששתי השיטות של החישוב מתקרבות ככל ש קטן, מה שמצביע על נכונות המימוש של הגראדיאנט שלנו.



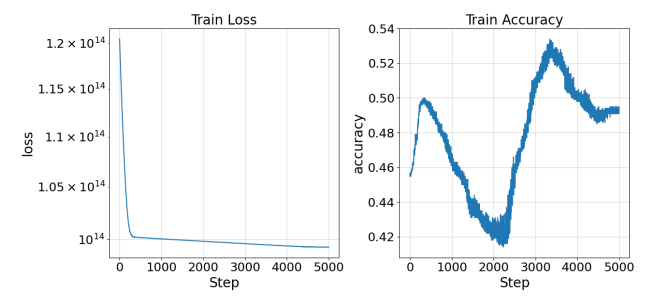
ככל ש- גדל אנחנו רואים שההפרש בין הגראדיאנט האנליטי והנומרי גדל . תופעה זאת קורית, כיוון ככל ש- גדל -> הקירוב הנומרי מתרחק מהגרדיאנט האמיתי ולכן התוצאה דומה למה שהיינו מצפים.

נשים לב שאנחנו מקבלים בכל הרצה תשובות שונות כי אנחנו מחשבים את הגרדיאנט על w,b שנבחרים באופן רנדומלי ובודקים את הערכים שמתקבלים ולכן בכל פעם יכולים להתקבל מעט ערכים שונים.

מה שחשוב שבכל ההרצות אנו מקבלים שככל קטן כך שהרנדומליות של לא משפיעה ואנחנו מקבלים קירוב טוב בין הגרדיאנט שלנו האנליטי לבין הנומרי.

ייתכן גם שיש רעשים מהחילוק במספרים אינפינטימליים במהלך תהליך הקירוב אבל אנחנו מסתכלים על ההתנהגות הכללית של העקומות שקיבלנו בשביל לאמת את החישוב האנליטי שלנו לגרדיאנט - ככל קטן הresiduals קטן.

(Q10)



בשאלה הנתונה אנו פותרים בעיית

אנחנו מנסים למצוא את ה שיביאו למינימום את הביטוי של soft-SVM formulation ואנחנו מבצעים זאת על ידי שימוש בSGD.

SGD עובד באיטרציות(steps), בכל איטרציה, האלגוריתם לוקח כמות(batch) דגימות מסוימת (ושונה) ומחשב את הgradient של הביטוי על פי . לאחר מכן האלגוריתם מעדכן את ע״י החסרת הגרדיאנט שחושב כפול גודל הצעד (learning rate) .

בגרף הloss הנתון אנחנו יכולים לראות שהעקומה יורדת במהלך האימון ולאט לאט מתכנסת כפי שהיינו מצפים מאחר מחפשים מינימום לפונקצית הloss.

ההתכנסות היא יחסית איטית וניתן ליחס זאת לlearning rate הקטן ולא בהכרח נוכל להגיע למינימום במספר הצעדים הנתון(5000).

בגרף הaccuracy ניתן לראות תנודות במהלך האימון ושהדיוק לא מאוד גבוה בתום האימון, בנוסף לא הייתה התכנסות לנקודת דיוק אחת.

האלגוריתם שלנו מבצע מינימיזציה על פונקצ׳ הכוללת בתוכה את הhinge loss , ולא את ה0-1 loss. לכן, אין הבטחה שפתרון של בעיית הsoft-svm הוא גם פתרון מינימלי ל0-1 loss. הaccuracy מתבסס על חישוב ה0-1 loss (ככל ש0-1 loss קטן , הaccuracy עולה).

אנו מצפות להתנהגות הזאת במקרה הספציפי הזה כי :

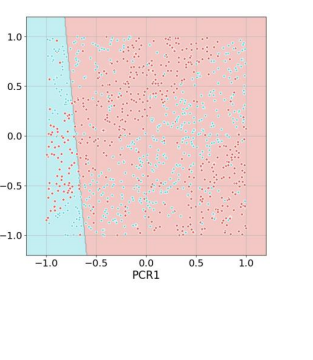
* הhinge loss נותן משקל גדול יותר לנקודות שמסווגות לא נכון על פי המרחק שלהן. ככל שהן רחוקות יותר המשקל גדול יותר, אנחנו יכולים לראות לפי התוצאה שהתקבלה שהדאטה לא פריד וקשה למצוא היפר- מישור שהנקודות שמסוגוות נכון יהיו קרובות אליו,
* שהדאטה אינו פריד לינארית ולכן קשה למצוא היפר- מישור שייתן דיוק מספיק טוב.

בנוסף נשים גם לב שהדאטה אינו פריד לינארית ולכן קשה למצוא היפר- מישור שייתן דיוק מספיק טוב.

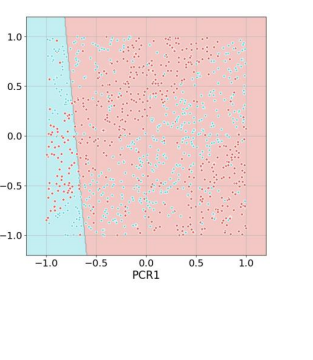
* פונקציית הloss בsoft-svm מושפעת גם בהתאם להיפר פרמטרים שלנו C וlr. במקרה שלנו, מאחר וC מאד גדול אז יש משקל גדול על הhinge loss ולכן האלגוריתם ינסה יותר לצמצם את הhinge loss מאשר לחפש מפריד עם margin גדול. עם זאת, מאחר והדאטה לא פריד לינארית ובנוסף הlr יחסית קטן, הloss של הפונק׳ עדיין יחסית גבוה בתום 5000 הצעדים (אין דרך לגרום לhinge loss להיות מאוד נמוך) .
* ניתן גם לשים לב שממספר צעדים מסויים העקומה יורדת בצורה איטית הרבה פחות כי אחרי מספר מסוים של צעדים הגראדיאנט קטן. בנוסף, ככל שמתקרבים למינימום האמיתי , באה לידי ביטוי העובדה שאנו משתמשים בsubgradients . נק׳ המינימום של הhinge loss היא נק׳ לא גזירה, ולכן אין הבטחה בכלל שנגיע לנק׳ מינימום עם קצב learning rate קבוע, גם אם הוא מאוד קטן.

לסיכום, אנחנו חושבות שסה״כ הגרפים תואמים את מה שהיינו מצפות , לאור הנקודות שהעלנו בדיון.

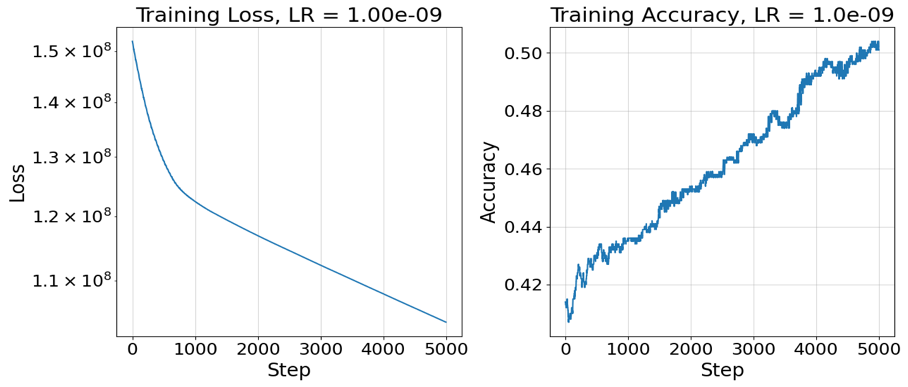
ניתן לראות שבסופו של דבר, המסווג שהאלגוריתם מצא הוא לא אופטימלי מבחינת הaccuracy (כי זה לא מה אנחנו מנסים למצוא לו מינימום בsoft-svm) וגם לא ניתן לדעת האם מה שקיבלנו נק׳ אופטימום מבחינת הhinge, כי גם לכך אין הבטחה. ניתן לראות אבל שכן באיזשהו מקום, המפריד הקטין את הhinge לפחות לנקודות האדומות באזור הכחול (אומנם הן מסווגות לא נכון, אבל הן כן יותר קרובות לmargin שלהן שנמצא בצד האדום, וזה נכון גם לחלק מנקודות הכחולות שנמצאות בחלק האדום.

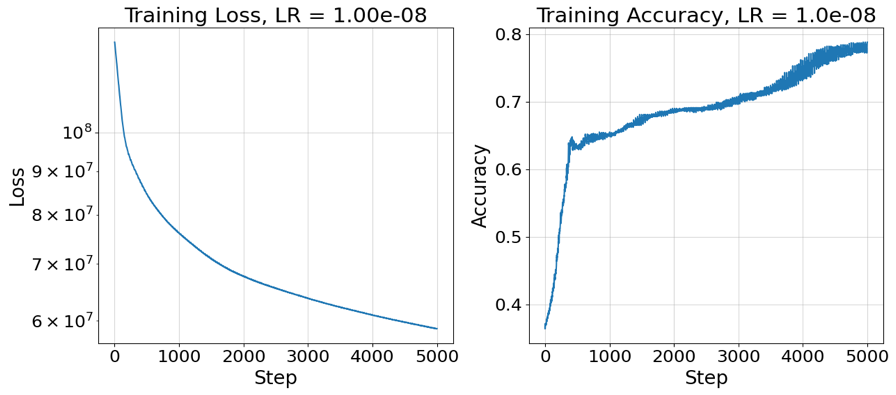
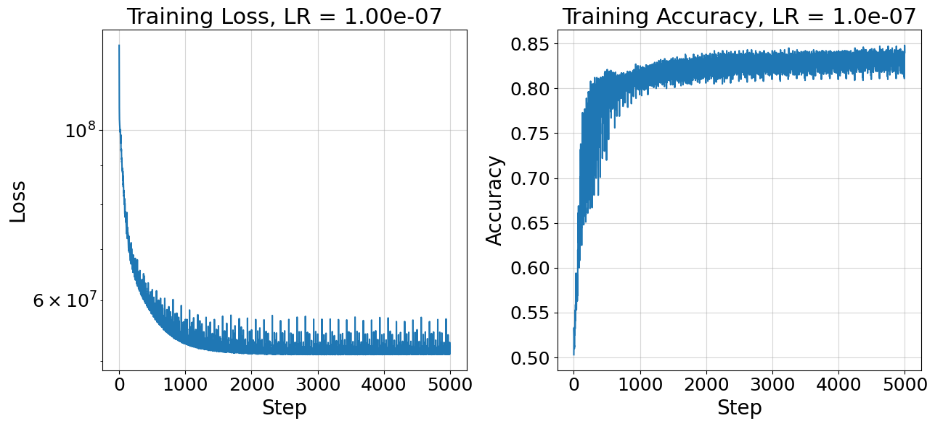
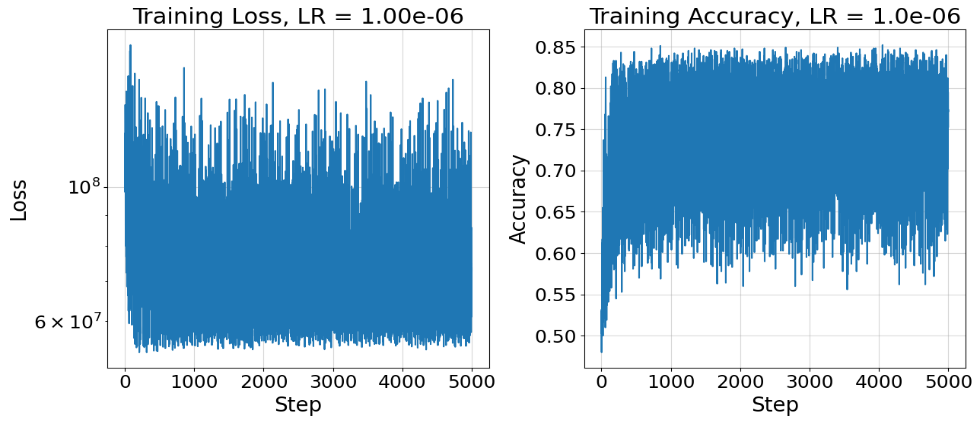


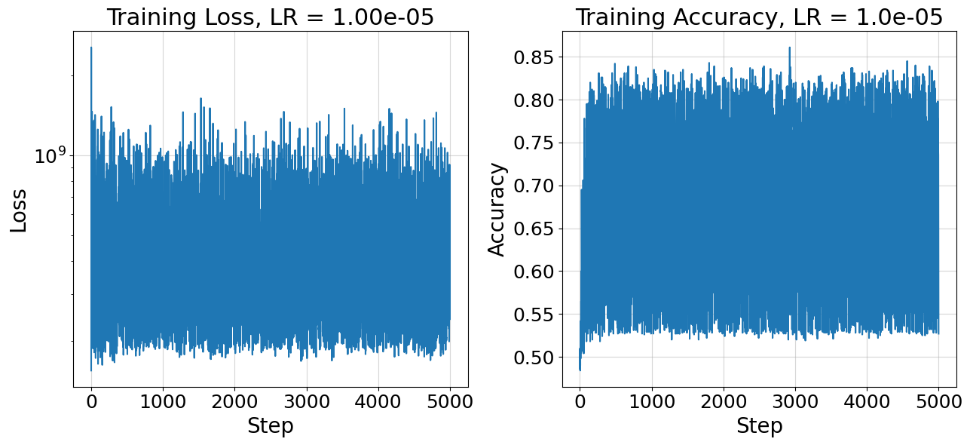
המפריד הלינארי שעובר באלכסון יכל לתת accuracy טוב יותר, אבל אנחנו חושבות שלווא דווקא הוא מצמצם את הhinge loss כיוון שניתן לראות שהנקודות שמסווגות לא נכון הן רחוקות מאד מהמפריד ולכן מגדילות את הhinge loss.



אפשר לראות פה בתמונה שאם היה מפריד כזה כך שצד ימין כחול וצד שמאל אדום, היינו יכולים להשיג מפריד עם דיוק טוב יותר, אבל הנקודות שמסומנות בעיגול הן רחוקות מאוד מהמפריד ולכן מגדילות את ההינג׳.

(Q11)





על בסיס. התרשימים החלטנו לבחור ב.

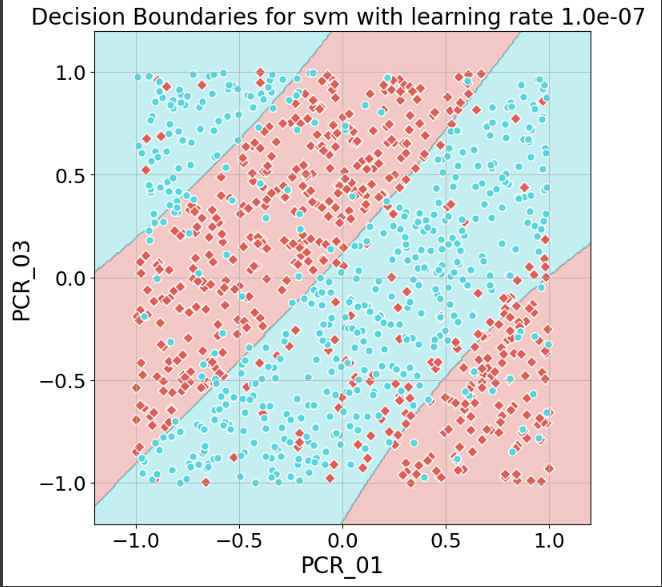
ראשית, עבור אין התכנסות בכלל של פונק׳ הloss, אלא היא מתבדרת לגמרי ולכן הlearning rate גדול מדי.

בנוסף, נראה כי ­ הוא קטן מדי עבור מס׳ הsteps. כן יש ירידה בפונקציית הloss אבל היא איטית וניתן לראות שבתום האימון ערכה עדיין גדול לעומת מינימומים אחרים. בנוסף, ה accuracy רק 0.5 בסוף האימון, בהשוואה לlearning rates אחרים שמידת הדיוק שם היתה יחסית גדולה (כמובן, ע״פ מה שרשמנו קודם שאין התחייבות שתהיה שיפור במידת הדיוק, כי זה לא הגורם לו אנחנו מחפשים מינימום אבל כן נרצה לבחור learning rate שמגדיל אותו במידת האפשר).

ההתלבטות שלנו הייתה בין הקצבים כיוון ששניהם מראים ירידה גדולה ביותר בloss ומראים שיפור במידת הaccuracy.

החלטנו לבחור בסופו של דבר ב , על אף שהוא מעט רועש יותר, מאחר ונראה כי הוא מביא את הloss לערך מינמלי יותר ב5000 צעדים, וגם הaccuracy מגיע לדיוק מעט גבוה יותר.

(2Q1)



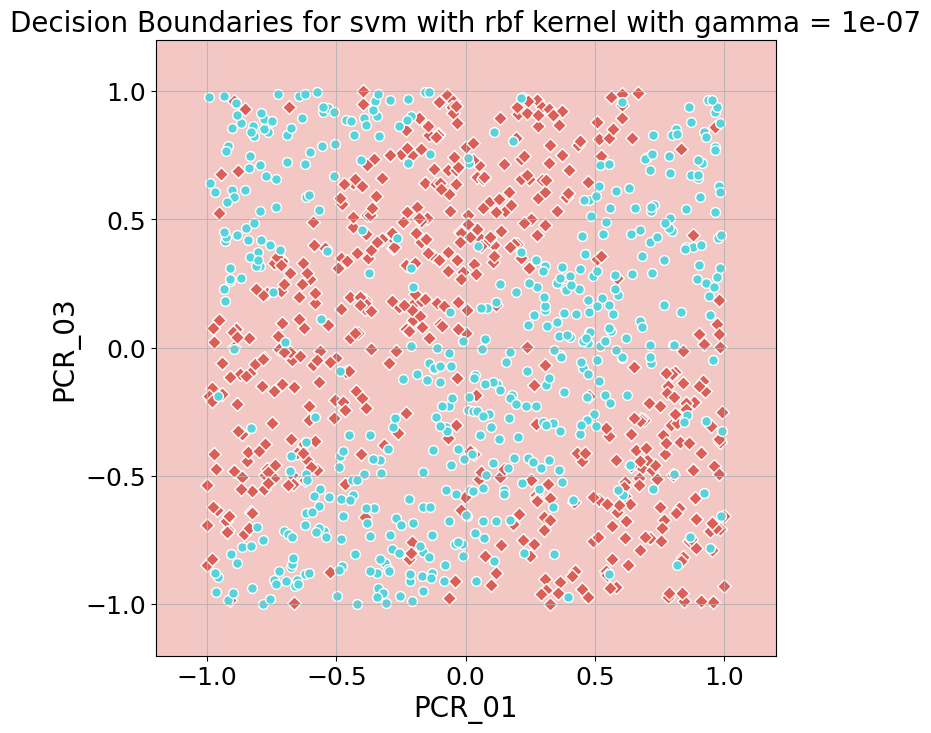


(Q13)

1. הוכחה(להשלים):
2. ב. הוכחנו קודם כי כאשר אנחנו מקבלים שכלל הפרדיקציה הוא ובמקרה שלנו כלל ההחלטה הוא - מאחר שאין דוגמאות בקבוצת האימון שעליהם מקדמי האלפא האפס, הפרדיקציה של כל נקודה נקבעת על פי תיוג של רוב הנקודות בדאטה סט. נשים לב, שככל שגמא שואף ל0 כמעט ואין חשיבות למרחק של הנקודה x משאר הדוגמאות ולכן ככל שגמא קטן לכיוון האפס אנחנו מקבלים מודל שמחזיר פרדיקציה קבועה (-1 או 1 על פי תיוג הרוב שיש בדאטה סט). אפשר גם להגיד שככל שגמא שואף ל0 אנחנו מקבלים מודל knn שההחלטה שלו מתקבלת על פי m שכנים (m דוגמאות בקבוצת האימון), כאשר שככל שגמא קטן ככה ״מס השכנים״ מתקרב לm.

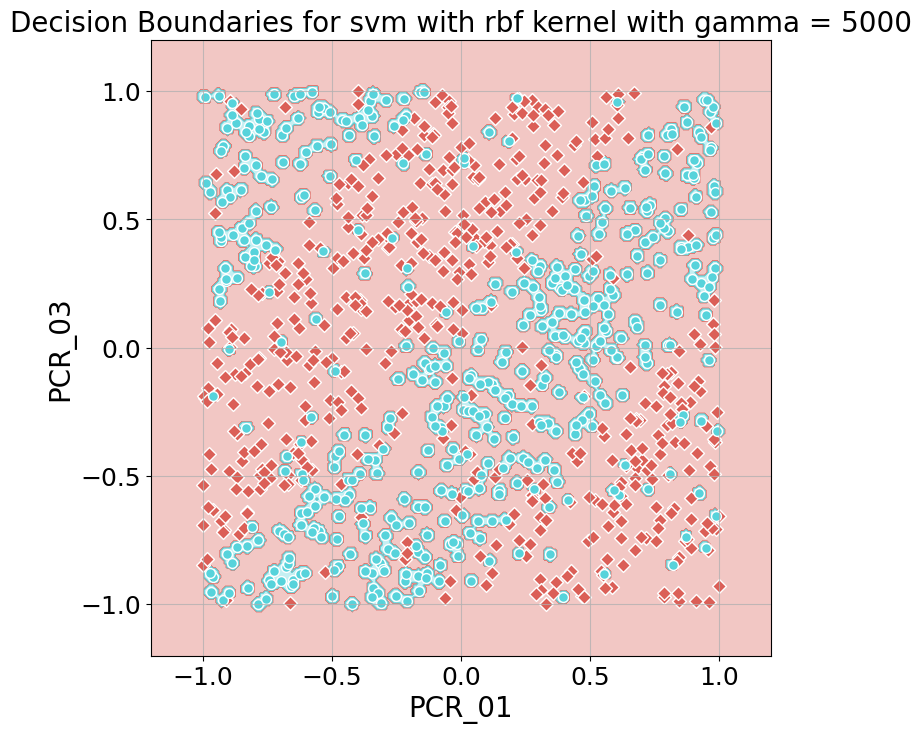
(Q14)

נשים לב ש מאוד קטן ואכן גבולות ההחלטה של המודל מתאימים לכלל ההחלטה הנידון בסעיף 13 ב, כפי שניתן לראות קיבלנו את המסווג ה״טריוואלי״ שפולט פרידקציה קבועה לכל הנקודות.





(Q15)

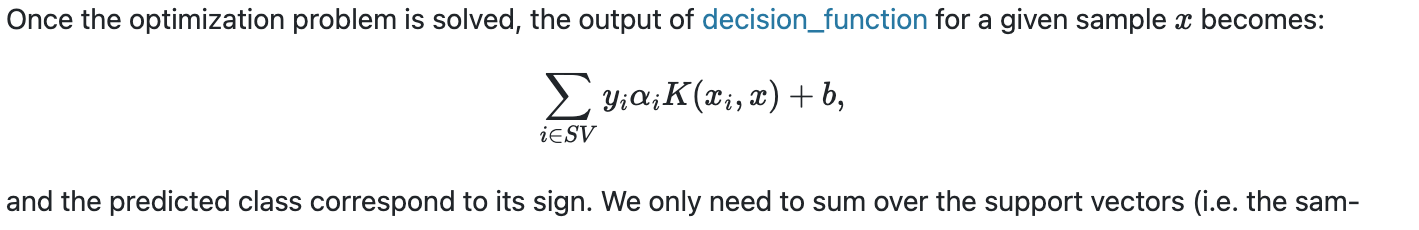


נשים לב כי המודל שקיבלנו דומה למודל 1-nn בכך שהוא מתאים את עצמו לקבוצת האימון בצורה מושלמת (יוצר סביבות קטנות סביב נק׳ חריגות). על פי כלל הפרידקציה, מאחר וגמא גדול מאד, נקבל שההשפעה של נק׳ support vector מקבוצת האימון תבוא לידי ביטוי רק במידה והנקודה שרוצים לחזות את התיוג שלה קרובה אליה מאד. לכן, במובן הזה, המודל אמור להיות דומה מאד למודל knn.

עם זאת, ניתן לראות כי לעומת מודל הknn, המודל הזה נותן פרדיקציה קבועה (במקרה שלנו הצבע האדום מייצג פרדיקציה -1) לרוב הנקודות, ויודע לתת פרדיקציה 1 רק שהנקודה נמצאת ממש קרוב לנקודה עם תיוג 1 . כלומר המודל לא יודע לתת פרדיקציה 1 לנקודה שהשכן הקרוב שלה הוא כחול כאשר המרחק בין הנקודות לא מאד מאד קרוב.

אנחנו חושבות שתופעה זו קורית מאחר וכאשר גם גמא גדול וגם המרחק גדול נקבל כי הוא כמעט 0, ומבחינה נומרית -> המחשב עשוי להתייחס אליו כאפס.

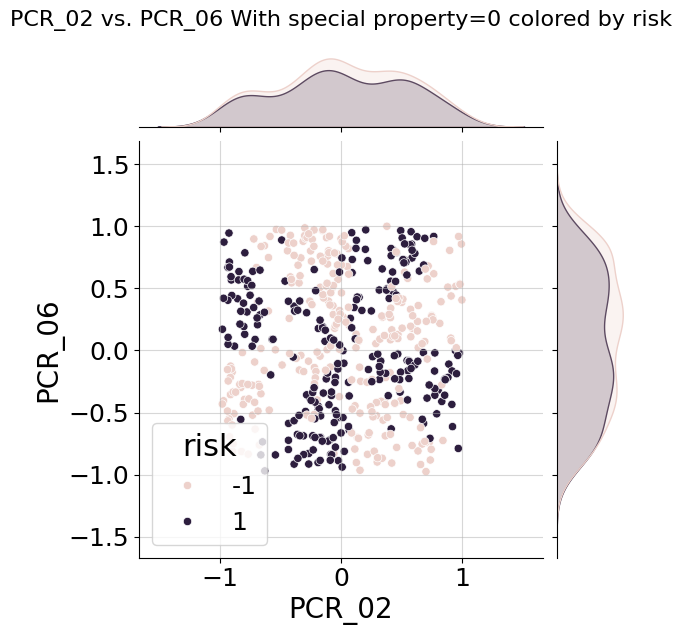
במצב כזה, כאשר לנקודה אין אף שכן מספיק קרוב מבחינה נומרית, הפרידקציה שלה נקבעת לפי גורם הb. כפי שניתן לראות כלל החלטה של המודל שאימנו לפי הדוקומנטציה של sklearn הוא:



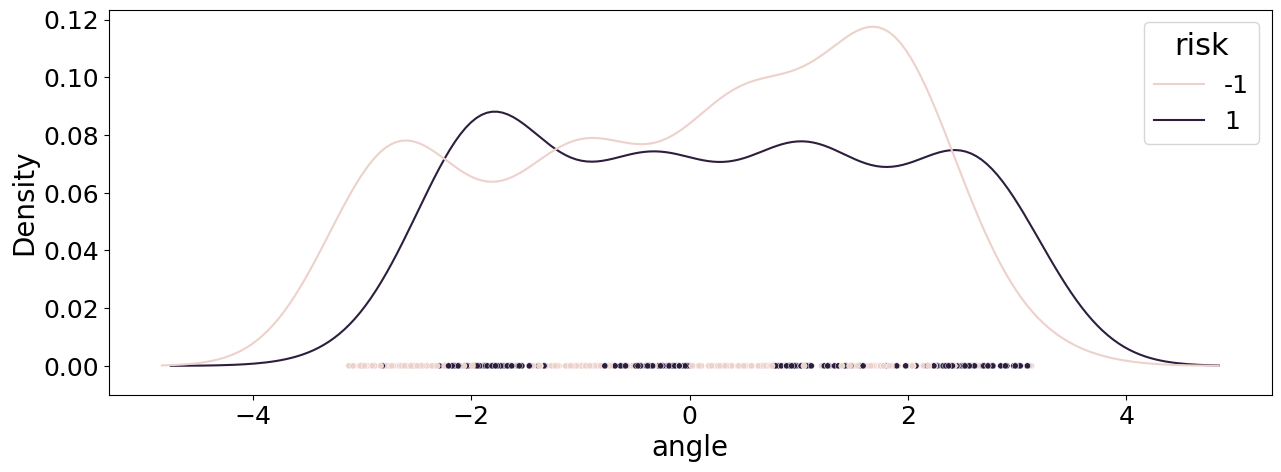
במקרה שלנו הדפסנו וראינו כי :

ולכן הגיוני שהחיזוי שלנו במקרים של נקודות ללא שכנים מספיק קרובים יוצא שלילי (שכן החיזוי נקבע ע״פ הסימן של b שהוא שלילי(.

(Q16)



על בסיס התרשים, ניתן לראות שבדאטה יש תבנית של 4 קבוצות (הקבוצות נוצרות הרביעים שנגזרים מהצירים x=0, y=0) כך שכל קבוצה היא כמעט פרידה לינארית. ניתן לראות שעבור כל רביע, בערך בזווית 45 מעלות התיוג של הדאטה משתנה.

(Q17)

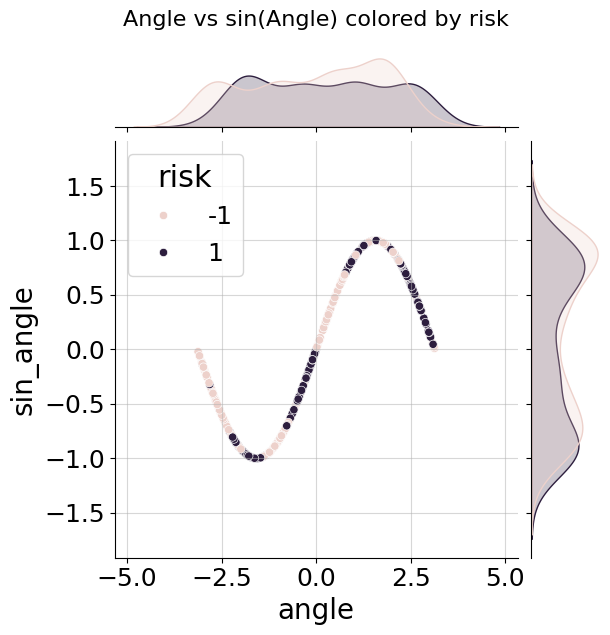
בתרשים ניתן לראות כי יצירת הפיצ׳ר החדש של הזווית , מחלקת את הדאטה (לא בצורה מושלמת) כך שניתן לחלקו לטווחים של תיוגים מסוימים במרווחים של רדיאנים (כפי שראינו בplot הראשוני).

הדאטה אינו פריד לינארית (לא ניתן להפרידו ל2 קבוצות בלבד באמצעות מפריד לינארי).

עוד ניתן לראות ע״פ פונקציות הצפיפות, שסה״כ משתנה הזווית מתפלג יוניפורמית על פני הטווח פאי למינוס פאי, כלומר הזוויות שהווקטורים שלנו יוצרים נעים סביב כל הטווח בין [0,360] . בנוסף ניתן לראות שבטווחים מסוימים יש אכן יותר סיכוי לקבל מחלקה -1 על פי פונקציית הצפיפות (בהתאם לטווח) וכן גם עבור המחלקה 1.

(Q18)

על פי התרשים הפיצ׳רים עדיין אינם פרידים לינארית:

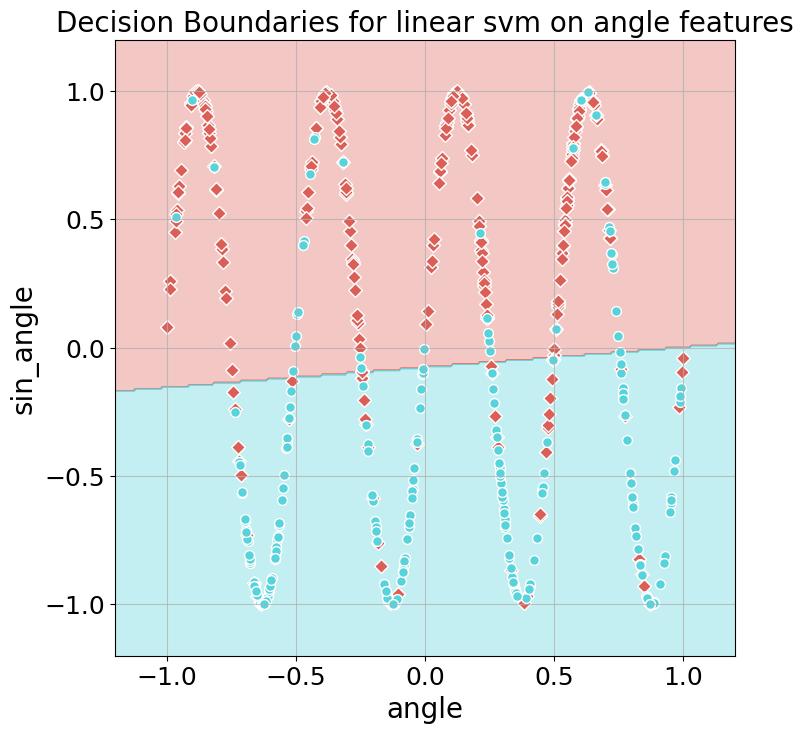


(Q19)

ראינו קודם שניתן להפריד את הדאטה לחלוקה לטווחים, כל שבערך כל רדיאנים מקבלים מחלקה אחרת. אנחנו יודעים שפונ׳ הסינוס היא פונ׳ מחזורית הנעה בין 1 ל-1, לכן נרצה לצמצם את המחזור שלה כך שנקבל שהפונקציה משלימה מחזור שלם ב רדיאנים. כך אכן נקבל שכל רדיאנים יש נק׳ קיצון והסימן של הפונקציה משתנה. כך נוכל לדאוג שבטווחים של מחלקה risk=1 ערך הסינוס יהיה בעל סימן מסוים ובטווחים של מחלקה risk=-1 ערך הסינוס יהיה בעל ערך אחר, מה שיצור הפרדה לינארית כפי שאנו רוצים.

לצורך כך חשבנו להגדיר למשל ואז נקבל שהמחזור של פונק׳ סינוס משתנה כפי שרצינו:

קיבלנו את המודל הבא:



ניתן לראות כי למודל שלנו ביצועים טובים ביחס למודל הקודם. למודל קודם היה אחוז דיוק נמוך על קבוצת האימון וקבוצת הטסט, מה שמצביע על underfit ועל כך שההיפותזה של מודל לינארי לא התאימה לפיצ׳רים הקודמים. כעת, לאחר שיישמנו את הפיצ׳רים החדשים, ניתן לראות שהדאטה יחסית יותר פריד לינארית, ואכן קיבלנו אחוז דיוק של כ0.83 שזו עלייה משמעותית ביחס לקודם (ואפילו לא כווננו היפר פרמטרים במקרה זה).