**פרוטוקול ניסוי:**

**תאור האלגוריתם:**

השתמשנו בספריית WEKA למימוש האלגוריתם המבוקש. לצורך כך בנינו מחלקה C45PruneableAlg, שיורשת מ C45PruneableClassifierTree, והיא למעשה מבצעת את העבודה הנדרשת. שינינו את הקוד של getProbs() במחלקה ClassifierTree – כך כשמטפלים בעלה, ערך החזרה נקבע על ידי מתודה שנקראת handleLeaf(), ואותה דרסנו במחלקה שמימשנו, שתבצע חיפוש KNN ותחזיר תשובה בהתאם. התשובה הינה 1 במידה והסיווג הוא אכן הסיווג הנקבע על ידי KNN, ו-0 אחרת.  
בשביל לממש KNN בעלים, עם סיום בניית העץ והגיזום כפי שנעשה באלג' המקורי, ביצענו צעד נוסף – בו הזנו בשנית את כל הדוגמאות ופעפענו אותן לעלים. כל עלה שמר את כל הדוגמאות שמועברות אליו, ועם סיום הרצת הדוגמאות, בנינו בכל העלים מסווג KNN, באמצעות מימוש IBk שקיים ב WEKA.  
כעת, בכל פעם שהגיעה דוגמא לסיווג – היא עברה באלג' הרגיל של C4.5, עד שהגיעה לעלה המתאים, ושם סיווגה התקבל לפי תוצאת ה KNN שענה המסווג הנמצא באותו עלה.

שאלה 1:

להוסיף גרף של שני המסווגים כפונ' של K

ברצוננו לבחון את ההיפותזה – עבור k=7 אחוזי ההצלחה של שני האלגוריתמים נלקחו מאותה התפלגות. לצורך כך נשתמש במבחן Wilcoxon, מכיוון שההתפלגויות אינן נורמאליות (ניתן לראות זאת בהיסטוגרמה המצורפת) – ולכן paired t-test אינו מתאים, וכן חשוב לנו ההפרש בין האלגוריתם – ולא רק "ניצחון או הפסד".  
במבחן wilcoxon עבור k=7, אנו מקבלים כי אין מובהקות סטטיסטית, אנו מקבלים ודאות של 42.7% ששתי האוכלוסיות נלקחו מאותו המדגם.  
יצויין שכאשר חזרנו על הניסוי עבור k=5 ו k=3 קיבלנו מובהקות סטטיסטית טובה יותר: 14.9% שמדובר על אותו מדגם עבור k=5, ו 8.4% עבור k=3. עבור k>7 אנו מקבלים ודאות גבוהה יותר שמדובר על אותו מדגם, והשיא הוא עבור k=9 ועבורו 92.5% שמדובר באותו מדגם.  
מכך נסיק: למרות שישנו יתרון קל מבחינת ממוצעים במקסימום כאשר מתעלמים מתכונות ב KNN שבעלים, אין מובהקות סטטיסטית לדבר, פרט אולי ל k=3,k=5, ולכן נסיק שהאלגוריתמים שקולים.

להוסיף מבחן סטטיסטי של ממוצעים, להראות שגרף אחד שולט ממש על השני באופן מובהק סטטיסטית.

התוצאה הצפויה הייתה שהמנתח שמשתמש בכל התכונות לכל הפחות לא יהיה רע יותר מהמנתח השני – שכן הוא משתמש ביותר מידע על תכונות הדוגמאות, ואינו מתעלם ממידע – שעשוי להיות חיוני לקביעת הסיווג של הדוגמא.   
בפועל ראינו שאין מובהקות סטטיסטית בשוני בין שני האלגוריתמים. את ההבדל שבכל זאת קיים ניתן להסביר שהוא בא לידי ביטוי בעיקר בדוגמאות בעלות מאפיין נומרי – שהמרחק עצמו בין הדוגמאות – הינו מאפיין בעל משקל חשוב, דבר שבא לידי ביטוי ב KNN, ולא רק ה bin אליו משתייכת הדוגמא, כפי שנעשה בעצי סיווג. בתכונות נומינליות, ככל הנראה אין הבדל משמעותי בין 2 האלגוריתמים.

ולכן המנתח שמשתמש בכל התכונות עדיף – ובו נשתמש בסעיפים הבאים.

היסטוגרמת אחוזי הצלחה בניסויים עבור k=7:

להוסיף היסטוגרמה

מההיסטוגרמה ניתן ללמוד כי ככל הנראה אחוזי ההצלחה בכל אלגוריתם אינם מתפלגים נורמאלית.

שאלה 2:

להוסיף היסטוגרמה  
מניתוח ההיסטוגרמה ומבדיקת הממוצעים, ניתן לראות שהאלגוריתם המשתף את C4.5 ו KNN מביא תוצאות טובות משניהם, ומבחינה זו – ההשערה מתקיימת.  
תחילה נבחן את ההבדל בין C4.5 לבין C4.5(7NN) – כאן ההבדל הוא משמעותי יותר – וניתן אף להוכיח סטטיסטית במובהקות טובה שהאלגוריתמים מביאים תוצאות שונות – בחינת Wilcoxon מחזירה שהסיכוי ששתי אוכלוסיות אחוזי ההצלחה נלקחו מאותו מדגם היא 3.4%, ולכן ניתן להסיק שקרוב לוודאי תוצאות האלגוריתמים שונות מהותית האחת מהשניה. עוד על מבחנים סטטיסטיים נרחיב בשאלה 3.   
הסבר אפשרי לכך מתקבל עבור תכונות שאינן ניתנות להפרדה (בעיית ה XOR), דוגמא בולטת לכך היא עבור המידע cylinder-bands. אלגוריתם J48 לא מצליח להפריד כלל את התכונות, והעץ שנוצר הוא בעל צומת אחד בלבד, ולכן התשובה שמוחזרת הינה בדיקת רוב, ובמקרה זה אלגוריתם J48 יסווג כל דוגמא בתור noband, מבלי כלל לבדוק את התכונות. אלגוריתם C4.5(7NN) במקרה זה יתנהג בדיוק כמו אלגוריתם 7NN, ויחזיר תוצאה טובה יותר מ J48.

כעת נבחן את ההבדלים בין KNN לבין C4.5(7NN) – ניתן לראות שההבדל אינו משמעותי, אך עדיין יש הפרש ממוצעים קל לטובת C4.5(7NN). עם זאת, חשוב לציין שאין הבדל סטטיסטי בין השניים. מבחן Wilcoxon מחזיר שהשערת האפס (שני מדגמי אחוזי ההצלחה נלקחו מאותה התפלגות) נכונה ב- 99.2% ודאות.

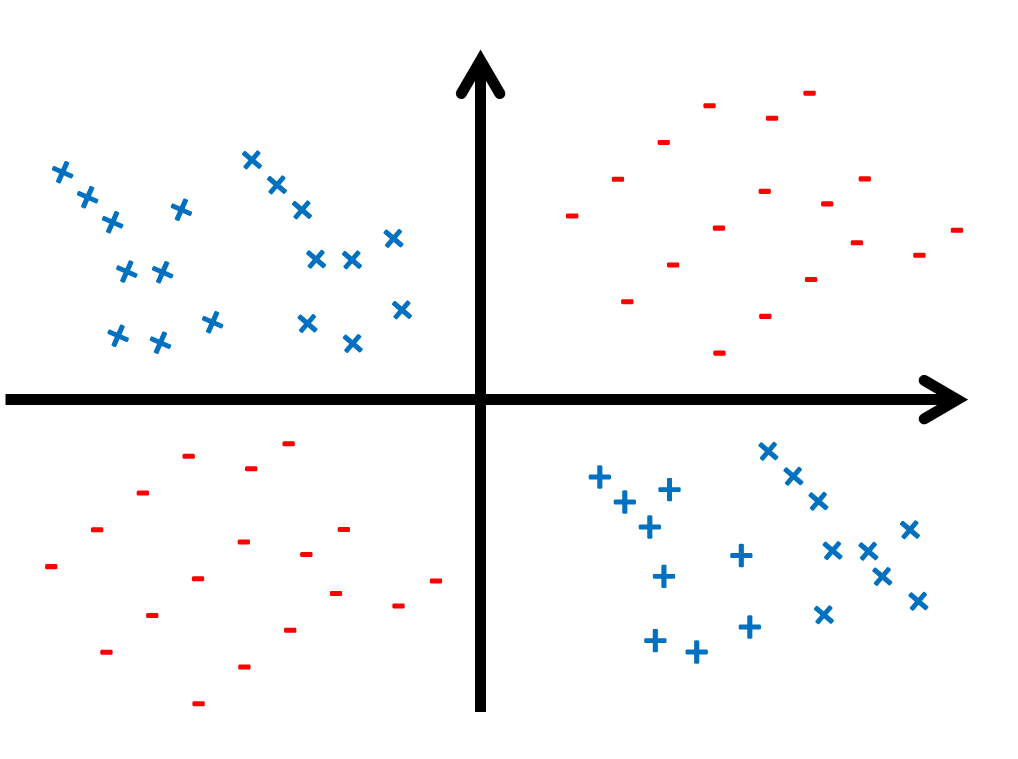
ניתן לראות עבור דוגמאות ספציפיות, כגון madelon – שהאלג' C4.5(7NN) טוב משמעותית מאלג' KNN, דוגמא זו הינה דוגמא שמייצגת היטב היכן עצים מצליחים ו KNN נכשל, מכיוון שכאן – יש מספר רב (500) תכונות, שלכל אחת מהן ב KNN משקל שווה, ולכן המשקל של כל תכונה נמוך יחסית. דבר זה פעמים רבות אינו נכון, שכן יש תכונה או מספר תכונות מצומצם שמכריעות עבור סיווג המידע, בעוד ששאר התכונות הינן "זבל", ולכן רק מקשות על KNN להביא לסיווג נכון, בעוד שאלגוריתמי עצים יתעלמו מתכונות אלה. אלגוריתם C4.5(7NN) במקרים אלו יבנה תחילה את העץ, ורק בעלים תסווג כל דוגמא בהתאם ל 7NN, שכאן כבר כמות הרעש תפחת משמעותית, שכן אם העץ כבר סיווג בצורה טובה את הדוגמאות, הרי שבכל מקרה השכנים הקרובים כנראה יהיו בסיווג הנכון, ולכן הרעש הצפוי מריבוי התכונות יפחת משמעותית.

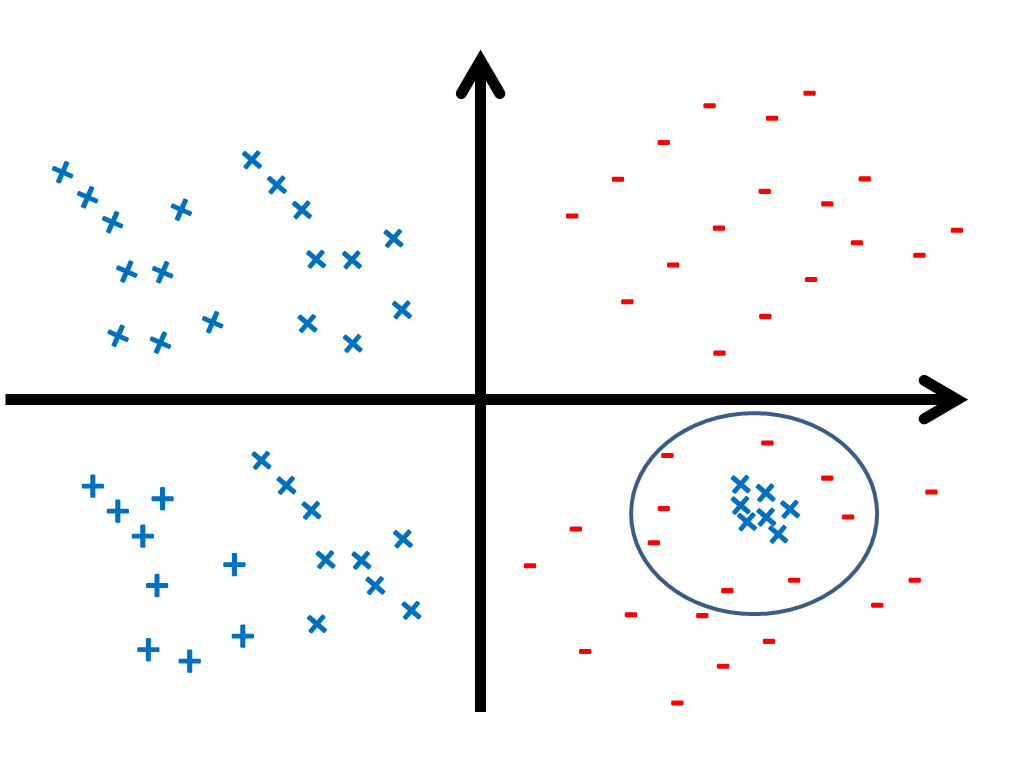
סעיף שלישי:על סמך התוצאות מ 1+2, לקחת ולהריץ מבחנים סטטיסטיים.

שאלה 4:

C4.5 יכשל לעומת C4.5(7NN), אל מול בעיית ה XOR. C4.5 לא יצליח למצוא הפרדה טובה על סמך עצים בין התכונות על מנת לסווג דוגמאות, ולכן ייתכן בהחלט שהאלגוריתם יחזיר עץ בעל שורש בלבד, ולמעשה סיווג יעשה על פי רוב, כל הקלטים יסווגו אותו הדבר – לפי הסיווג הנפוץ ביותר על ה training set.

דבר זה בולט במידע שניתן לתרגיל עבור cylinder-bands, בו העץ המוחזר הינו בעל שורש בלבד, והכרעה נעשית על פי רוב.

דוגמא בדו מימד:  


אלגוריתם C4.5 עשוי להיות טוב מאלגוריתם C4.5(7NN) כאשר לדוגמא באחד מעלי העץ יש מקבץ דוגמאות (שלא ניתן להפריד בצורה יעילה) שהינו דחוס (מבחינת מספר הדוגמאות במרחק נמוך יחסית) יותר בצורה משמעותית משאר הדוגמאות בעלה, וכתוצאה נקבל שמרחב גדול יחסית יסווג בהתאם למקבץ זה, במקום בסיווג הנכון.  
דוגמא בדו מימד:  
  
בדוגמא הנ"ל, יש תחום קטן בתוך המרחב שברובו מסומן ב (-), שמסומן ב (+). הדוגמאות שקיבלנו מדגישות את מקטע זה, ולכן המידע במקטע זה דחוס מאוד, ולכן יש הרבה מאוד דוגמאות בו, וכתוצאה – כל הקטע המסומן בעיגול כחול (העיגול מסומן סכמתית) – יסומן ב (+), וקל לראות כי חלק גדול מהדוגמאות בתוך העיגול יסווגו למעשה כטעויות. במקרה זה לדוגמא – נסמן 8 דוגמאות (-) בתור (+), בעוד שעץ היה מסמן את כולן כ (-) [לפי רוב הדוגמאות בעלה], ולכן היינו מקבלים רק 7 טעויות.  
(\*) הערה: הסבר זה מניח שייתכן שהדוגמאות לא מגיעות מהתפלגות אחידה ממרחב המדגם (המדגם אינו מדגם מקרי פשוט).