**AI-HW3**

שאלה 1

1. בכדי לקבל מדד יחיד נבצע ממוצע על כל ההרצות, כלומר נסכום את הדיוק של כל ההרצות ונחלק ב-4. הדיוק שקיבלנו הינו: **0.709263714303** .
2. בדומה לסעיף א' גם כן נבצע ממוצע, כלומר נסכום למטרציה אחת את כל המטריצות המתקבלות מההרצות ונחלק כל תא ב-4. המטרציה שהתקבלה:

|  |  |  |
| --- | --- | --- |
|  | + | - |
| + | 59.75 | 10.75 |
| - | 20 | 15.25 |

שאלה 2

1. נגדיר את הבעיה כך:

True Positive – מעבד תקין שהאלגוריתם אומר שהוא תקין.

True Negative – מעבד מקולקל שהאלגוריתם אומר שהוא מקולקל.

False Positive – מעבד תקין שהאלגוריתם אומר שהוא מקולקל.

False Negative – מעבד מקולקל שהאלגוריתם אומר שהוא תקין.

1. נבדוק עבור כל מסווג את גודל הרווח (סכום הכסף אותו החברה הרוויחה ממכירת המעבדים פחות הסכום שהוציאה על ורפיקציה) כתוצאה משימוש במסווג זה.

עבור נסווג A נקבל כי בממוצע על כל 2000 מעבדים 900 יזוהו כתקינים ויעברו ורפיקציה לכן כאשר ימכרו ב10,000 נרוויח עלהם 9,000 (בגלל מחיר הורפקציה), בנוסף עבור 500 מעבדים נבצע ורפקציה ונגלה שהם אינם תקינים ולכן נפסיד על כל אחד 1,000 ₪. סה"כ נרוויח: ₪.

באופן דומה עבור מסווג B נמכור 990 מעבדים תקינים ב10 אלף ₪ ונבצע ורפקציה על 830 מעבדים לא תקינים לכן: ₪.

ועבור מסווג C נמכור אלף מעבדים תקינים ונבצע ורפקציה עבור 990 לא תקינים ולכן נרוויח:

**=> נעדיף לבחור את מסווג B בכדי לייצר רווח מקסימלי ממכירת המעבדים.**

שאלה 3

1. לא שינינו פרמטרים, האלגוריתם המקורי כבר מקיים Over Fitting
2. בכדי לגרום לתת התאמה גרמנו למסווג להתייחס רק לתכונה אחת וכן הגבלנו את עומק העץ ל-1.
3. התוצאה שקיבלנו במקרה של התאמת יתר:
   1. עבור סט האימון: 0.909397377391
   2. עבור סט המבחן: 0.716340025695

הסבר: תוצאות אלו מתאימות להתאמת יתר מכיוון שכפי שניתן לראות האלגוריתם מאפשר זיהוי ברמה גבוהה על איברים מתוך סט האימון (מעל 90%) אך נותן תוצאות משמעותית פחות טובות עבור (רק 70%)

התוצאה שקיבלנו במקרה של תת התאמה:

1. עבור סט האימון: 0.676927383014
2. עבור סט המבחן: 0.671324029524

הסבר: במקרה זה המסווג מבסס את החלוקה שלו רק על פי תכונה אחת ובנוסף עומק העת הוא 1, כלומר, המסווג בוחר סיווג לפי היחס בין המופעים בתכונה מסוימת ואכן נראה כי עבור רוב התכונות ישנו יחס של 1:2 בין הערכים. בנוסף, ניתן לראות כי התוצאות על סט המבחן וסט האימון דומות.

שאלה 4

שאלה 5

יש לבדוק את הביצועים של כל מסווג (המסווגים שונים זה מזה בתת הקבוצה של התכונות הנבחרות במצב הנוכחי) על קבוצת ולידציה ולא על קבוצת המבחן מכיוון שתמיד נרצה לבחון את המסווג ה"סופי"\"נבחר" על קבוצת אובייקטים שמעולם לא למד\ נבחן עליהם כדי שנוכל לקבל תמונת מצב קרובה ככל האפשר על ההתנהגות של המסווג על אובייקטים מהעולם החיצוני.

שאלה 6

מומש

שאלה 7

1. 0.7075471698113207 KNN=5 , no feature choosing , precision =
2. [0, 4, 28, 30, 1, 2, 8, 26, 29]features\_subset =

knn = 5 , b = 8 precision is 0.8018867924528302

שאלה 8

1. 0.72641509434
2. 0.801886792453
3. האלג' עם הגיזום המוקדם נתן דיוק גבוהה יותר, הדיוק נמדד על קבוצת המבחן.

שאלה 9

השיפור הגדול יותר היה כאשר בחרנו תכונות בשיטת wrapper (שיפור של 13% לעומת 11.8% )

שאלה 10

הערה – מ2 הניסויים שביצענו קשה להפיק לקחים לגבי יתרונות וחסרונות של השיטות. יתכן ואם היינו בוחרים תת קבוצה של תכונות גדולה יותר או קטנה יותר, או שאם היינו משנים את המספר המקסימלי בעלה בשיטה השנייה היינו מקבלים מסווגים עם דיוקים שונים מהותית. לכן נענה על השאלה בצורה תאורטית.

Wrapper

חיסרון - אלגוריתם חמדני ולכן הוא אינו יכול ל"ראות" ערך של יותר מהוספת תכונה אחת. יתכן מאוד ששילוב של 2 תכונות או יותר היה משפר את הדיוק של המסווג אך תכונות אלו בנפרד אינן הכי טובות ולכן לא יבחרו לתת הקבוצה.

יתרון – יותר קל לצפות בכמה תכונות ישתמש המסווג בסוף (טריוויאלי לשיטת הWRAPPER. בשיטת הembedded חייבים להריץ כדי לגלות)

Embedded

חיסרון – יש חשיבות לסדר של הדוגמאות באימון המסווג. סדר אחד יגרום לעלה להכיל 20 אובייקטים בשלב X של הריצה וסדר אחר בשלב Y מה שיגרום לעץ הבחירה ליהיות שונה.

יתרון – חיסכון בזמן – (מנגנון בחירת המאפיינים משולב באימון המסווג)