מגישים:

מיכאל תורג'מן – 307910984

בני לודמן - 307270215

חלק 1

1. א. כדי לקבל דיוק יחיד נשלב את התוצאות של ה-cross validation ע"י ממוצע, כלומר לכל הרצה חישבנו את הדיוק והוספנו לסכום, ולבסוף חילקנו את הסכום ב-4.
הדיוק שקיבלנו הוא 0.709.
ב. באופן דומה לסעיף הקודם, סכמנו את המטריצות של כל הרצה, ולבסוף חילקנו את כל הערכים במטריצה ב-4.
המטריצה שקיבלנו:

```
[[ 59.25  11.25]
 [ 15.75  19.5 ]]
```

2. א. True Negative – מעבד מקולקל שסווג כמקולקל
True Positive – מעבד תקין שסווג כתקין
False Negative – מעבד תקין שסווג כמקולקל
False Positive – מעבד מקולקל שסווג כתקין
ב. נתסכל על הרווח הממוצע של כל מסווג:
(על כל מחשב תקין שעובר ולידציה החברה מרוויחה 9,000: מקבלת 10,000 על המעבד ומשלמת 1,000 על הוולידציה, על כל מעבד מקולקל שעובר סיווג ונכשל בוולידציה החברה מפסידה 1,000)
מסווג A: $900 \cdot 9{,}000 - 500 \cdot 1{,}000 = 7{,}600{,}000$
מסווג B: $990 \cdot 9{,}000 - 830 \cdot 1{,}000 = 8{,}080{,}000$
מסווג C: $1{,}000 \cdot 9{,}000 - 990 \cdot 1{,}000 = 8{,}010{,}000$
לכן נבחר במסווג B.

3. א. לא שינינו כלום. האלגוריתם הרגיל מספיק כדי לקבל overfitting, כי מקבלים את העץ הכי גדול שאנחנו יכולים לקבל – לא מתבצע גיזום, והעץ מפותח לגודל מקסימלי.
ב. שינינו את הפרמטר max_depth להיות 1, כי כפי שראינו בהרצאה ככל שהעץ קטן יותר כך מקבלים underfitting. בנוסף, שינינו את האלגוריתם כך שיתייחס רק למאפיין אחד, כדי להוריד כמה שיותר את ההתאמה לקבוצת האימון.
ג. עבור התאמת יתר קיבלנו:
סט האימון – 0.91
סט המבחן – 0.71
עבור התאמת חסר קיבלנו:
סט האימון – 0.66
סט המבחן – 0.72
התוצאות להתאמת יתר מתאימות למצופה משום שעבור סט האימון אנו מקבלים אחוזים גבוהים (עדיין לא 100% משום שה-dataset שקיבלנו אינו עקבי), ולעומת זאת עבור סט המבחן אנו מקבלים אחוזים נמוכים יותר – כלומר יכולת ההכללה שלו נמוכה.
התוצאות להתאמת חסר לא כל כך מתאימות לציפיות שלנו – היינו מצפים לקבל בערך 50% דיוק גם בסט האימון וגם בסט המבחן. אנו מעריכים שהתוצאה שקיבלנו נובעת מכך שאין איזון בין דוגמאות חיוביות ושליליות ב-dataset. בכל מקרה אנו מבחינים כי עבור סט המבחן אנו מקבלים דיוק טוב יותר מאשר מסט האימון, כצפוי למקרה של התאמת חסר.

חלק 2

4. 1. מספרי תתי הקבוצות של הקבוצה S הוא הגודל של קבוצת החזקה שלה - $2^{|S|} = 2^N$.
2. מספר תתי הקבוצות של הקבוצה S בגודל b הוא: $\binom{N}{b} = \frac{N!}{b!(N-b)!}$ (מספר האפשרויות לבחור b מאפיינים מתוך קבוצה של N, ללא חשיבות לסדר ביניהם).

5. את הביצועים נמדוד על קבוצת **המבחן –** עבור מדידה של ביצועים סופיים אחרי שכבר נבחרה תת הקבוצה הטובה ביותר (ע"ב cross validation שמדד ביצועים על קבוצת הוולידציה), אנו מודדים את הביצועים על קבוצת המבחן.
6. מימוש בקוד.
7. 1. הדיוק שהתקבל הוא 0.707.
2. הדיוק שהתקבל הוא 0.792.

בונוס:

נציע את המקרה הבא:

x1	x2	x3	x4	Classification
1	0	0	0	**0**
1	0	0	0	**0**
1	0	0	1	**1**
0	1	1	0	**1**
1	0	1	1	**0**
1	0	1	1	**0**

מתקיים כי:

$$U(x_1) = U(x_2) = \frac{5}{6}$$

$$U(x_3) = U(x_4) = \frac{4}{6}$$

אבל:

$$U(x_1, x_2) = \frac{5}{6}$$

$$U(x_3, x_4) = \frac{6}{6}$$

8. א. 0.735
ב. 0.82
ג. האלגוריתם עם גיזום נתן תוצאה טובה יותר, בדקנו על קבוצת המבחן (25% מהdataset).

9. קיבלנו שיפור בביצועים בעזרת בחירת מאפיינים (ברוב ההרצות שהרצנו, כאשר חלוקת ה-dataset התבצעה באופן רנדומלי, בהתאם ליחס הדרוש).
קיבלנו תוצאות טובות יותר עבור embedded method (ברוב ההרצות, אך בהפרש יחסית קטן).

10. <u>חיסרון</u>: Wrapper method יותר כבד חישובית מאשר Embedded method – יש צורך לאמן את המסווג מחדש עבור כל תת קבוצה של מאפיינים, לעומת הEmbedded שבו המאפיינים נבחרים תוך כדי הרצה אחת.
<u>יתרון</u>: לא מותאם ספציפית לאלגוריתם למידה מסוים, ניתן להשתמש בו לכמה אלגוריתמים שונים, לעומת הEmbedded שבו בחירת המאפיינים היא חלק מאלגוריתם הלמידה, לכן אם עוברים להשתמש באלגוריתם למידה אחר, צריך להתאים אותו שוב כדי שישלב את מנגנון בחירת המאפיינים בתוכו.