

## ריאן אבוליל : 324886225

**שלב 3 :** שלב 3 מדבר על חלוקת הנתונים שלי לאימון ובדיקה שתלויה בפרמטרים :

*test size*

*train size*

*random state*

*shuffle*

*stratify*

כך שלכל אחד מהפרמטרים יש לו את החשיבויות שלו ואם אחד מהפרמטרים בחרנו באופן לא מתאים אז ישפיע משמעותי על התוצאה . למשל היה גודל הבדיקה קטן מדי יובל לתוצאות מוטעות בהערכת המודל וההפך אם יהיה גודל מדי אז המודל יתקשה ללמוד.

דוגמה נוספת אם לא היה מוגדר *random state*

אז הפיצול יהיה אקראי בכל הריצה ואינו עקבי וחוזר.

Accuracy: 0.49

Precision: 0.49

Recall: 0.46

F1-Score: 0.47

Confusion Matrix:

[[26 24]

[27 23]]

לכן בחרתי :

test size=0.2 : שומר על איזון בין כמות נתוני האימון והבדיקה

random state=42 : מבטיח תוצאה עקבית בכל ריצה -

stratify=y : שומר על יחס הקטגוריות, חשוב במיוחד בנתונים לא מאוזנים

**שלב 4:** שלב 4 מדבר על אימון עצי החלטה להלן יש 5 ניסיונות :

**ניסיון 1:**

```
model = DecisionTreeClassifier(  
    criterion='entropy',  
    max_depth=3,  
    min_samples_split=10,  
    min_samples_leaf=5,  
    random_state=42  
)
```

לקחתי הפרמטרים האלה וקיבלתי תוצאה לא טובה בכלל, נכון שקיבלתי רמת דיוק של 55% וזה נראה טוב אבל הדיוק יכול להיות מטעה. שאר התוצאות זה אפס וזה אומר

1: המודל לא מצליח לזהות אף דוגמה חיובית

2: המודל מסווג הכל לקטגוריה שלילית.

Accuracy: 0.55

Precision: 0.00

Recall: 0.00

F1-Score: 0.00

Confusion Matrix:

[[11 0]

[ 9 0]]

### **ניסיון 2:**

Accuracy: 0.45. דיוק נמוך מאוד, כלומר, המודל מנבא נכון רק ב-45% מהמקרים

Precision: 0.38, Recall: 0.33, F1-Score: 0.35 : מדדי הדיוק, השליפה והציון המשוקלל נמוכים מאוד, מה שמצביע על כך שהמודל אינו מתפקד היטב

6 מהדוגמאות שסווגו נכון True Negatives.  
5 מהדוגמאות שסווגו בטעות למרות שהם False Positives  
6 מהדוגמאות שסווגו בטעות למרות שהם False Negatives  
3 מהדוגמאות שסווגו נכון True Positives.

### **ניסיון 3:**

criterion='Gini'

max\_depth=5,

min\_samples\_split=4,

min\_samples\_leaf=2,

random state=42

בניסיון השני קיבלתי רמת דיוק של 45% מהמקרים אבל רואים שיש שיפור במדדים האחרים אך קיבלנו

Precision: מתוך כל התחזיות החיוביות רק 40% נכונות וזה שיפור מ 0.38 אבל עדין נמוך

Recall: המודל מזהה 0.44 מהדוגמאות החיוביות שקיימות בפועל.

5 מהדוגמאות שסווגו נכון True Negatives.  
6 מהדוגמאות שסווגו בטעות למרות שהם False Positives  
5 מהדוגמאות שסווגו בטעות למרות שהם False Negatives  
5 מהדוגמאות שסווגו נכון True Positives.

```
Accuracy: 0.45
Precision: 0.40
Recall: 0.44
F1-Score: 0.42
```

```
Confusion Matrix:
[[5 6]
 [5 4]]
```

## ניסיון 4

בניסיון הזה הוספתי כל הפרמטרים שצריך להוסיף כך שהשפיעו על התוצאה .

השתמשתי במדד ג'יני שזה נתן אופציה למודל לבחור את הפיצולים כך שיקטינו את חוסר הטוהר וזה עזר לי לייעל בניית העץ מבחינת דיוק.

השתמשתי בפיצול האופטימלי BEST עבור כל צומת שזה מונע בחירות אקראיות

הגבלתי עומק העץ למניעת התאמת יתר כך ש העץ התמקד בפיצולים חשובים במקום להיכנס לפרטים קטנים מאוד.

הגבלתי גם מינימום הדוגמאות בכל צומת או עלה כדי למנוע פיצולים קטנים מאוד וזה עזר להפחית את הרעש .

לא השתמשתי בגיזום.

```
criterion='entropy' splitter='best' max_depth=5 min_samples_split=4 min_samples_leaf=2
min_weight_fraction_leaf=0.01 max_features=5 random_state=42 max_leaf_nodes=10
min_impurity_decrease=0.01 class_weight={0: 1, 1: 3} ccp_alpha=0.01
```

```
Accuracy: 0.60
Precision: 0.53
Recall: 0.89
F1-Score: 0.67
```

```
Confusion Matrix:
[[4 7]
 [1 8]]
```

בניסיון קיבלתי רמת דיוק של 60% מהמקרים זה שיפור מהתוצאה הקודמת אבל רואים שיש שיפור במדדים האחרים אך קיבלנו

Precision: מתוך כל התחזיות החיוביות רק 53% נכונות

המודל מזהה 0.89 מהדוגמאות החיוביות שקיימות בפועל זהו שיפור Recall: משמעותי בזיהוי דוגמאות חיוביות  
איוון טוב יותר בין זיהוי דוגמאות חיוביות לבין דיוק סיווג חיובי

4 מהדוגמאות שסווגו נכון True Negatives.

7 מהדוגמאות שסווגו בטעות למרות שהם False Positives

1 מהדוגמאות שסווגו בטעות למרות שהם False Negatives

8 מהדוגמאות שסווגו נכון True Positives.

## ניסיון 5

בניסיון הזה שינתי את הפרמטרים וקיבלתי תוצאה טובה. שינתי כך שהענף יתפצל ל 2 במקום 4 וזה ברירת המחדל ומאפשר גמישות יותר. חוץ מזה כל עלה בעץ חייב להכיל לפחות דוגמה אחת שעוזר לי לשמור על פשטות העץ וגם המודל יתן משקל גדול פי 2 לשגיאות במחלקה 1 ביחס למחלקה 0.

```
criterion='entropy', splitter='best', max_depth=5, min_samples_split=2, min_samples_leaf=1,
    min_weight_fraction_leaf=0.0, max_features=None, random_state=42,
max_leaf_nodes=15, min_impurity_decrease=0.0, class_weight={0: 1, 1: 2}, ccp_alpha=0.01)
```

Accuracy: 0.93  
Precision: 0.94  
Recall: 0.94  
F1-Score: 0.94

Confusion Matrix:  
[[38 4]  
 [ 4 68]]

	Predicted Negative	Predicted Positive
Actual Negative	38	4
Actual Positive	4	68

<https://scikit-learn.org/1.5/modules/generated/sklearn.tree.DecisionTreeClassifier.html#sklearn.tree.DecisionTreeClassifier> \*הוספתי בפרמטרים מהלינק הזה

## Random Forest Classifier

הוא מודל של למידת מכונה שמשמש באוסף של עצים חישוביים כדי לקבל תחזיות.

עכשיו כל פרמטר יש לו השפעה אחרת למשל :

n\_estimators: מספר העצים ביער כך שמספר גדול ממש של העצים מעלה את הדיוק של המודל.

max\_depth: העומק המקסימלי של כל עץ כך שעומק גדול יעשה לנו בעיית התאמת יתר ועומק קטן זה להפך. יגרום שהמודל לא ילמד מספיק את הנתונים .

min\_samples\_split: מספר המינימלי של דוגמאות כך שהשפעה שלו יכולה לגרום או להתאמת יתר או המודל לא ילמד מספיק .

min\_samples\_leaf: המספר המינימלי של דוגמאות הדרוש בכל עלה כך ש ערך קטן מדי יוביל לעצים עם רעש וההפך ערך גדול מדי יגרום ל שהמודל לא ילמד מספיק את הנתונים.

## בניסיון הראשון:

```
n_estimators=100, criterion='gini', max_depth=None, min_samples_split=2,
    min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features='sqrt',
```

```
max_leaf_nodes=None, min_impurity_decrease=0.8, bootstrap=True,
oob_score=True, n_jobs=-1, random_state=42, verbose=0, warm_start=False,
class_weight=None, ccp_alpha=0.89, max_samples=None
```

Test Accuracy: 0.63

```
Classification Report:
      precision    recall  f1-score   support

 malignant      0.00      0.00      0.00        42
   benign      0.63      1.00      0.77        72

 accuracy          0.32      0.50      0.63       114
  macro avg          0.32      0.50      0.39       114
 weighted avg          0.40      0.63      0.49       114
```

קיבלתי את התשובה הזאת כך שיש לי בעיה אך הערכים לא מתאימים של הפרמטרים בגלל ששמתי ערך 0.8 min\_impurity\_decrease.

זה גרם שהמודל לא עשה פיצולים מספקים בצמתים חוץ מזה המודל לא קיבל משקולות למחלקות.

### בניסיון השני:

```
n_estimators=200, criterion='gini', max_depth=15, min_samples_split=5, min_samples_leaf=2,
min_weight_fraction_leaf=0.0, max_features='sqrt', max_leaf_nodes=None,
min_impurity_decrease=0.01, bootstrap=True, oob_score=True, n_jobs=-1, random_state=42,
verbose=0, warm_start=False, class_weight='balanced', ccp_alpha=0.0, max_samples=None
```

Test Accuracy: 0.92

```
Classification Report:
      precision    recall  f1-score   support

 malignant      0.85      0.95      0.90        42
   benign      0.97      0.90      0.94        72

 accuracy          0.91      0.93      0.92       114
  macro avg          0.91      0.93      0.92       114
 weighted avg          0.93      0.92      0.92       114
```

זה שיפור משמעותי כך שהמודל בצורה נכונה 92% מהדוגמאות במבחן. התוצאה מספקת הצלחת זיהוי malignant

הפרמטרים הללו יצרו איזון בין פשטות המודל למורכבותו, עם דגש על מניעת Overfitting בצורה טובה. והתאמה לאי-שוויון בנתונים.

n\_estimators: מספר העצים ביער כך שמספר גדול ממש של העצים מעלה את הדיוק של המודל. ואני שמתי 200

max\_depth: הגבלת העומק מונעת התאמת יתר. ואני שמתי 15

מינימום דוגמאות לפיצול ועלה: (min\_samples\_split=5, min\_samples\_leaf=2)

מבטיח שהמודל לא יפצל צמתים או ייצור עלים על בסיס מעט מדי דוגמאות

(criterion='gini').

class\_weight='balanced' זה עזר מבחינת התחשבות באי שוויון בין הקטגוריות.

min\_impurity\_decrease=0.01)

ניסיון 3:

המודל מאוזן היטב עם ביצועים מצוינים ב-Precision, Recall, ו-F1-Score וזה שיפור טוב כך ש  
השינויים שבצעתי היא :

- n\_estimators=300 - הגדלת מספר העצים לשיפור יציבות וחזזי מדויק -
- criterion='gini' - שימוש בקריטריון מהיר ופשוט לחישוב אי-טוהר -
- max\_depth=15 - ולשיפור הכללה Overfitting הגבלת עומק למניעת -
- min\_samples\_split=8 - העלאת הדרישה לדוגמאות לפיצול לשיפור יציבות -
- min\_samples\_leaf=4 - דרישה למינימום דוגמאות לכל עלה, מונע פיצולים קטנים מדי -
- max\_leaf\_nodes=100 - הגבלת מספר העלים לצורך פשטות המודל ושיפור הכללה -
- min\_impurity\_decrease=0.005 - דרישה לירידה משמעותית יותר באי-טוהר, לשיפור יעילות הפיצולים -
- class\_weight='balanced' - איזון משקלים בין מחלקות, מתאים לנתונים לא מאוזנים -
- ccp\_alpha=0.001 - הוספת עונש לגיזום, מה שמפחית מורכבות עודפת בעץ -
- max\_samples=0.8 - שימוש ב-80% מהדגימות בכל עץ, משפר את גיוון העצים -

Test Accuracy: 0.94

	Classification Report:			
	precision	recall	f1-score	support
malignant	0.91	0.93	0.92	42
benign	0.96	0.94	0.95	72
accuracy			0.94	114
macro avg	0.93	0.94	0.93	114
weighted avg	0.94	0.94	0.94	114

#### ניסיון 4:

המודל מפגין ביצועים גבוהים בשתי המחלקות, עם איזון מצוין בין Precision ו-Recall.  
דיוק המודל במבחן (95%) מצביע על יכולת הכללה טובה, מה שמעיד על התאמה מוצלחת של הפרמטרים

Test Accuracy: 0.95

	Classification Report:			
	precision	recall	f1-score	support
malignant	0.93	0.93	0.93	42
benign	0.96	0.96	0.96	72
accuracy			0.95	114
macro avg	0.94	0.94	0.94	114
weighted avg	0.95	0.95	0.95	114

- שיפור יציבות:** הגדלת מספר העצים ושיפור הקריטריון לפיצול הביאו ליציבות גבוהה יותר
- שיפור הכללה:** הגבלת מספר העלים ומינימום הדוגמאות לעלה ל 2 שזה מנע את התאמת יתר
- שיפור ביצועים באיזון:**
- שימוש באיזון משוקלל:**
- הוספת ccp\_alpha=0.0005:** עזרה לגזום צמתים לא משמעותיים, מה ששיפר את דיוק הפיצולים והפחית טעויות

השינויים הללו הפכו את המודל ליותר יציב, מדויק ומאוזן  
הגעתי לאחוז הצלחה של 95%

ניסיון 5:

בניסיון הזה ביצעתי כמה שינויים וקיבלנו שיפור כך ש

`n_estimators=300`: הגדלתי את מספר העצים, מה שמשפר יציבות ודיוק.

`criterion='entropy'`

`max_depth=20`: מאפשר לי עומק גדול יותר .

`min_samples_split=4`

ויש עוד מלא שהם:

`min_samples_leaf`

`max_features`

`min_impurity_decrease`

`class_weight`

`ccp_alpha`

`max_samples`

Test Accuracy: 0.96

Classification Report:				
	precision	recall	f1-score	support
malignant	0.95	0.93	0.94	42
benign	0.96	0.97	0.97	72
accuracy			0.96	114
macro avg	0.96	0.95	0.95	114
weighted avg	0.96	0.96	0.96	114

## AdaBoostClassifier

בניסיון הראשון : לקחתי את הפרמטרים האלה :

```
estimator=base_estimator,  
n_estimators=100,  
learning_rate=0.5,  
algorithm='SAMME.R',  
random_state=42
```

Test Accuracy: 0.84

Classification Report:					
	precision	recall	f1-score	support	
0	0.80	0.87	0.84	93	
1	0.88	0.81	0.84	107	
accuracy			0.84	200	
macro avg	0.84	0.84	0.84	200	
weighted avg	0.84	0.84	0.84	200	

בניסיון השני: לקחתי את הפרמטרים האלה והיה לי שיפור ממש קטן :

```
n_estimators=200,  
learning_rate=0.3,  
algorithm='SAMME.R',  
random_state=42
```

Test Accuracy: 0.85

Classification Report:					
	precision	recall	f1-score	support	
0	0.82	0.87	0.84	93	
1	0.88	0.83	0.86	107	
accuracy			0.85	200	
macro avg	0.85	0.85	0.85	200	
weighted avg	0.85	0.85	0.85	200	

85% מהתחזיות של המודל נכונות. 0.85 Accuracy

0.82 Precision (0) מתוך כל מה שסווג כ-0 (Negative), 82% אכן נכונים.

0.88 Precision (1) מתוך כל מה שסווג כ-1 (Positive), 88% אכן נכונים.

0.87 Recall (0) מתוך כל הדוגמאות שבאמת שייכות ל-0, 87% זוהו נכון.

0.83 Recall (1) מתוך כל הדוגמאות שבאמת שייכות ל-1, 83% זוהו נכון.

0.84 F1-Score (0) מדד מאוזן של דיוק ושליפה עבור מחלקה 0.

0.86 F1-Score (1) מדד מאוזן של דיוק ושליפה עבור מחלקה 1.



בניסיון השלישי : לקחתי את הפרמטרים האלה שמציינים את :

```
estimator=base_estimator,  
  
    n_estimators=50,  
    learning_rate=1.0,  
    algorithm='SAMME.R',  
    random_state=42
```

מדי `n_estimators=50`. מספר ה-"לומדים החלשים" כך שהערך גדול מדי עלול לגרום לאיחור בזמן האימון וערך נמוך עלול לגרום ה לתת תאמה

`learning_rate`. קובע את התרומה של כל "לומד חלש" לאנסמבל הכולל  
ערך נמוך יותר מפחית את משקל כל מודל, ולכן לעיתים דורש יותר לומדים חלשים

\*\*\*\* ערך גבוה מדי עלול לגרום למודל לא יציב

\*\*\* ערך נמוך מדי עלול להוביל לאימון איטי

`algorithm`: משתמשת לעדכון משקלות הדגימות ולביצוע חיזוק השתמשתי בקוד שלי SAMME.R

#### שחזור התוצאות

וקיבלתי תוצאה של

```
Test Accuracy: 0.87  
Classification Report:  
              precision    recall  f1-score   support  
0               0.83        0.90        0.87         93  
1               0.91        0.84        0.87        107  
accuracy                0.87         200  
macro avg              0.87        0.87        0.87        200  
weighted avg           0.87        0.87        0.87        200
```

בניסיון 4 : לקחתי את הפרמטרים האלה שמציינים את :

```
estimator=base_estimator,  
n_estimators=300,  
learning_rate=0.1,  
algorithm='SAMME.R',  
random_state=42  
)
```

Test Accuracy: 0.90

```
Classification Report:  
              precision    recall  f1-score   support  
0               0.84        0.96        0.89         93  
1               0.96        0.84        0.90        107  
accuracy                0.90         200  
macro avg              0.90        0.90        0.89        200  
weighted avg           0.90        0.90        0.90        200
```

```

pipeline = Pipeline([
    ('scaler', StandardScaler()),
    estimator=base_estimator,
    n_estimators=500,
    algorithm='SAMME.R',
    ('adaboost', AdaBoostClassifier(
        learning_rate=0.05,
        random_state=42
    ))
])

```

Test Accuracy: 0.94

Classification Report:

	precision	recall	f1-score	support
0	0.94	0.93	0.94	106
1	0.93	0.94	0.93	94
accuracy			0.94	200
macro avg	0.93	0.94	0.93	200
weighted avg	0.94	0.94	0.94	200

קיבלתי את התוצאה הזאת אחרי שעשיתי נרמול

## **מדדים נחשבים טובים במספרים:**

1. **Accuracy (דיוק):**
  - בבעיות מאוזנות 80%-90% מעל: **טוב**.
  - במיוחד בנתונים מאוזנים, 70% מתחת: **בעייתי**.
2. **Precision (דיוק ספציפי):**
  - 0.8 (80%) מעל: **טוב**.
  - False Positives במיוחד אם יש הרבה, 0.7 (70%) מתחת: **בעייתי**.
3. **Recall (רגישות):**
  - 0.8 (80%) מעל: **טוב**.
  - False Negatives במיוחד אם, 0.7 (70%) מתחת: **בעייתי**.
4. **F1-Score:**
  - 0.8 (80%) מעל: **טוב**.
  - כי זה משלב דיוק ורגישות, 0.7 (70%) מתחת: **בעייתי**.
5. **Confusion Matrix:**
  - True Positives (TP) ו-True Negatives (TN) ערכים גבוהים ב False Positives (FP) ו-False Negatives (FN).