

# פרויקט למידת מכונה - התקפי לב

## מבוא

בפרויקט זה בוצע ניתוח נתונים וחילוץ גיל באמצעות טכניקות שונות של למידת מכונה. הפרויקט כלל שימוש במספר אלגוריתמים כמו רגרסיה לינארית, Decision Tree, Random Forest ו-Gradient Boosting. הנתונים נלקחו מדאטאסט הקשור לניתוח התקפי לב וכוללים תכונות שונות המיוחסות למטופלים. הפרויקט נועד לבדוק איזה מודל מצליח לחזות את הגיל בצורה הטובה ביותר, כמו גם להבדיל בין קטגוריות גיל שונות באמצעות סיווג.

## הצגת הבעיה

בפרויקט זה אנחנו ננסה לחזות את הסיכון של אדם מסוים לקבלת התקף לב או לא בעזרת תשובה בינארית - כן או לא. תחילה, אנחנו ננסה לחזות את הגיל של האדם תוך ביצוע דיסקרטזציה ולאחר מכן נריץ כמה מסווגים כדי לראות את התוצאות שנציג לפניכם. בנוסף, אנחנו נריץ מספר מסווגים על הדאטא הכללי ונציג את התוצאות בדו"ח כאן. להלן השדות שקיימים ב-Data-Set שלנו:

- הגיל של האדם - Age
- המגדר של האדם - Gender
- כאבים בחזה - cp (1,2,3)
- לחץ דם במנוחה - trtbps
- רמת כולסטרול - chol
- רמת סוכר בדם - fbs
- מקסימום דפיקות לב - thalach
- הפיק האחרון - oldpeak
- תוצאות בדיקת סטרס - thall
- אנגינה - exng
- סיכון להתקף לב או לא - output

כדי לפתור את הבעיה נתמקד בשלבים פשוטים שיובילו אותנו לפתרון. בשלב הראשון, אנחנו נתמקד בעמודת הגיל. בשלב השני, אנחנו נתמקד בעמודת הoutput.

## שלבים כללים:

1. ביצוע עיבוד מקדים לDataset - בדיקת של עמודות שחסרים בהם ערכים, סינון עמודות לא רלוונטיות.
2. ביצוע רגרסיות (רלוונטי בעיקר לעמודת הגיל)
3. ביצוע דיסקרטזציה לעמודת הגיל.
4. נריץ מספר מודלים ונאמן אותם, נצפה לתוצאות.
5. נבצע השוואה בין התוצאות שיצאו לנו ונבחן את מדדי הדיוק שלנו.

## שלב ראשון - עמודת הגיל

### שלב העיבוד המקדים

- בשלב העיבוד המקדים, התמקדנו בכמה תחומים. תחילה רצינו לראות ולבדוק נתונים מספריים על הסט שלנו, זה חשוב כי אנחנו נרצה לראות את המינימום והמקסימום של הגיל לדוגמא כשנרצה לבצע על העמודה הזו דיסקרטזציה. להלן התוצאה של df.describe():

	age	sex	cp	trtbps	chol	fbs	restecg	thalachh	exng	oldpeak	slp	caa	thall	output
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	0.528053	149.646865	0.326733	1.039604	1.399340	0.729373	2.313531	0.544554
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	0.525860	22.905161	0.469794	1.161075	0.616226	1.022606	0.612277	0.498835
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	133.500000	0.000000	0.000000	1.000000	0.000000	2.000000	0.000000
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000	153.000000	0.000000	0.800000	1.000000	0.000000	2.000000	1.000000
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	1.000000	166.000000	1.000000	1.600000	2.000000	1.000000	3.000000	1.000000
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000	6.200000	2.000000	4.000000	3.000000	1.000000

שמנו לב מהנתונים הללו לדוגמא כי הגיל המינימלי הוא 29 ואילו הגיל המקסימלי הוא 77, לכן אנחנו מבינים כי אנחנו נצטרך לבצע דיסקרטזציה שתנוע בקטגוריות שלה סביב הגילאים הללו.

- דבר נוסף שניסינו לבדוק והיה רלוונטי הוא עבור עמודות ריקות או שורות חסרות ונתונים שהם לא קיימים בסט שלנו. לשמחתנו, הרצנו את הבדיקה דרך isnull וגילינו כי אין לנו דברים

חסרים.

- חישובנו מטריצת קולורציה עבור עמודת הגיל, רצינו לראות אילו עמודות כדאי להוריד על פי התוצאות. להלן התוצאות:

```
Correlation with 'age':
age      1.000000
trtbps   0.279351
caa      0.276326
chol     0.213678
oldpeak  0.210013
fbs      0.121308
exng     0.096801
thall    0.068001
cp       -0.068653
sex      -0.098447
restecg  -0.116211
slp      -0.168814
output   -0.225439
thalachh -0.398522
Name: age, dtype: float64
```

הלכנו על הערכים הקיצוניים יותר ובסופו של דבר החלטנו על הורדה של - 'thall', 'cp', 'sex', 'restecg', 'slp', 'output' ושמו לב שהם העמודות שפחות רלוונטיות לנו גם מבחינת היגיון הסט.

## שלב הרגרסיות

בשלב הרגרסיה החלטנו שאנחנו נלך על שתי שיטות - שיטה אחת היא לבצע רגרסיה ללא כל ביצוע נורמליזציה והשיטה השנייה היא תהיה עם ביצוע נורמליזציה לעמודות המספריות. רצינו לבדוק ולהשוות את ההבדלים בין ביצוע רגרסיה עם נורמליזציה וביצוע רגרסיה ללא נורמליזציה - אם בכלל יש הבדל. להלן התוצאות:

עץ החלטה:

Desicion-Tree Regresor w/Normalization	Desicion-Tree Regresor without/Normalization
MES: 65.37429297650952	MES: 66.13514704697964
R^2: 0.11005434928809565	R^2: 0.09969677997434978

המסקנה העיקרית שלנו היא כי אין הבדל מהותי בין רגרסור עץ החלטה עם נורמליזציה ובלי נורמליזציה. בנוסף התוצאות הן מאוד מאוד גרועות ולכן עולה לנו חשד כי הסט הוא לא מחולק בצורה הטובה ביותר בעמודת הגיל וכי יהיה לנו עבודה בשלב הקלסיפיקציות.

רגרסיה ליניארית:

Desicion-Tree Regresor w/Normalization	Desicion-Tree Regresor without/Normalization
R^2: 0.16307494893904784	R^2: 0.16307494893904728
MES: 61.479466126579055	MES: 61.479466126579105

כפי שאפשר לראות, גם ברגרסיה הלינארית הערכים הם לא מספיק טובים. בנוסף ניתן לראות כי ההבדל בין ביצוע נורמליזציה לבלי נורמליזציה הוא ממש על האלפיות ואין הבדל מהותי כלל.

רגרסיית Random-Forest:

Desicion-Tree Regresor w/Normalization	Desicion-Tree Regresor without/Normalization
R^2: 0.13521846089119782	R^2: 0.13050268822711641
MES: 63.52576885245902	MES: 63.872183606557385

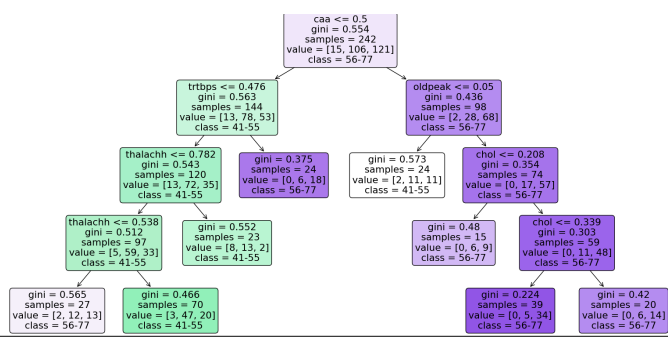
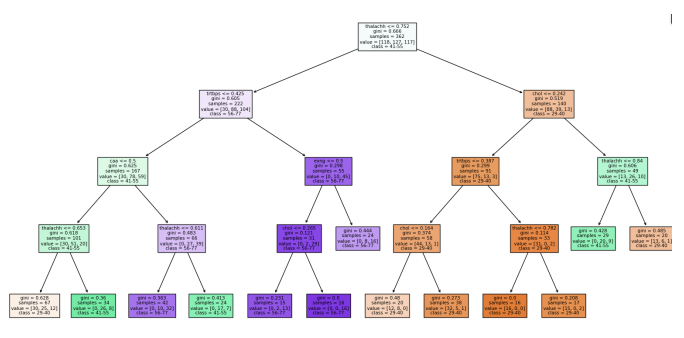
## שלב הדיסקרטזציה והרצת קליספקציות

לצורך ביצוע הדיסקרטזציה, החלטנו לחלק ל-3 קטגוריות שונות והן: גילאים בין 29-40, גילאים בין 41-55 וגילאים בין 56-77. תחילה הקטגוריות היו מחולקות בצורה אחרת, ועל פי הבחנה ובעיקר ניסוי ותהייה הבנו כי זה משפיע דרסטית על הדיוק של המודלים שלנו ומוריד אותם. ולכן תיקנו את זה.

בדקנו והבנו כי עם נורמליזציה כלל המסווגים יציעו ביצועים טובים יותר, לכן ביצענו אותם עם נורמליזציה. מנגד, גילינו כי יש לנו קטגוריה שסובלת מחוסר-איזון ולכן הרצנו כל מסווג פעמיים - פעם אחת עם oversampling ופעם אחת בלי כדי לכפר על חוסר האיזון ולראות את ההבדלים בדיוק.

## מסווגים:

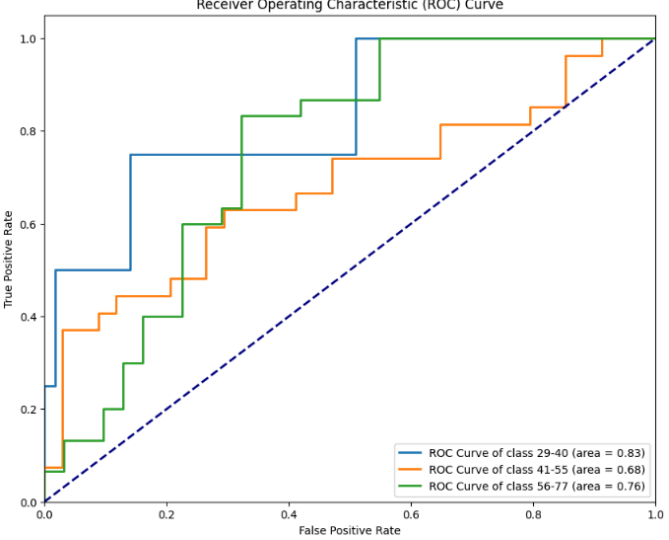
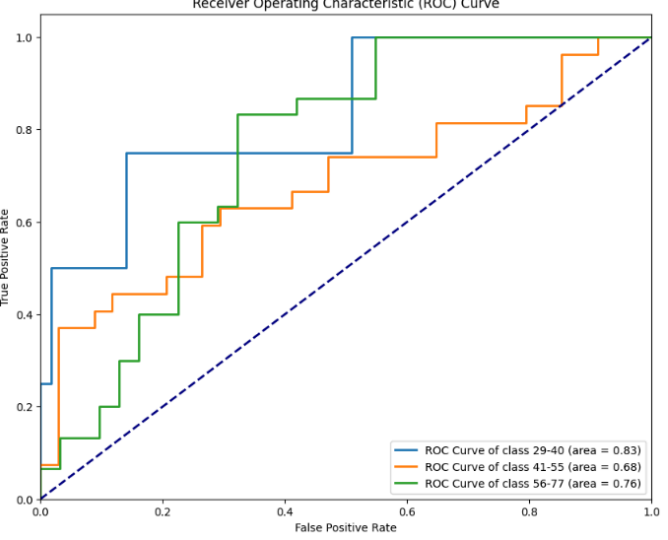
מסווג עץ החלטה עם נורמליזציה:

Desicion-Tree Classifaction w/Normalization, with No oversampling	Desicion-Tree Classifaction w/Normalization, with oversampling
	
Test-Accuracy: 0.7049180327868853	Test-Accuracy: 0.7472527472527473
Train-Accuracy: 0.6570247933884298	Train-Accuracy: 0.712707182320442
AUC ROC of Decision Tree-Test:29-40: 0.7412280701754387 AUC ROC of Decision Tree-Test:41-55: 0.7347494553376905 AUC ROC of Decision Tree-Test:56-77: 0.7612903225806452	AUC ROC of Decision Tree-Test:29-40: 0.9634273772204807 AUC ROC of Decision Tree-Test:41-55: 0.7723880597014925 AUC ROC of Decision Tree-Test:56-77: 0.8513931888544892

שני המסווגים אינם סובלים מ-**overfitting**, ואפשר לבצע את ההעדפה באיזה מודל להשתמש. ההתלבטות היא, האם לוותר על רמת דיוק גבוה יותר ברמת דיוק קצת יותר נמוכה אבל לא לעשות **oversample** או להפך.

בדרך כלל נעדיף להימנע מ**oversampling** כשמדובר על השלמת נתונים גדולה מידי כדי למנוע כפילויות בסט שיפגעו בתאימות, אבל זה בהחלט נתון לדיון בהתאם לנסיבות והתוצאות.

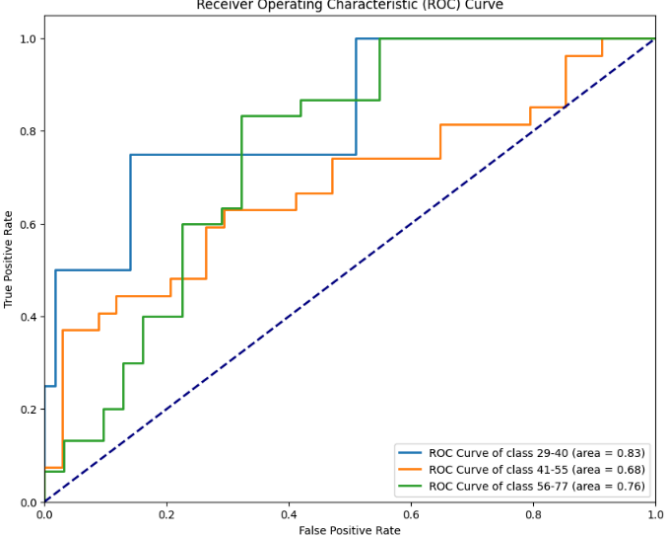
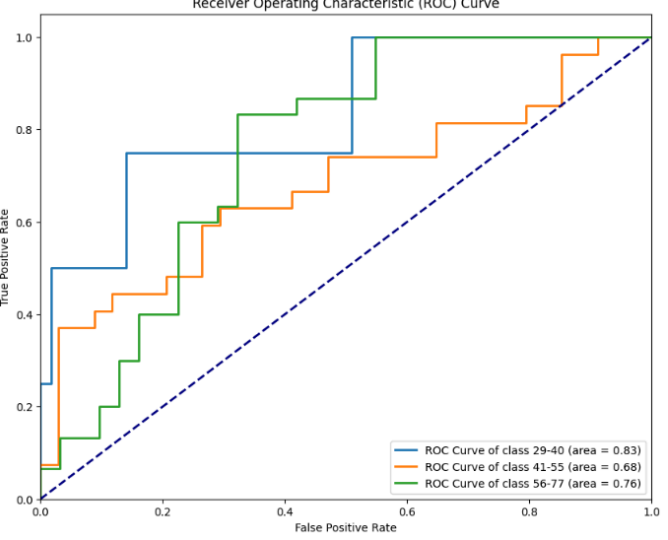
**מסווג ליניארי עם נורמליזציה:**

Linear Classification w/Normalization, with No oversampling	Linear Classification w/Normalization, with oversampling
	
Test-Accuracy: 0.6721311475409836	Test-Accuracy: 0.6373626373626373
Train-Accuracy: 0.6652892561983471	Train-Accuracy: 0.5994475138121547
<pre> AUC ROC of Logistic Regression-Test: 29-40: 0.8333333333333334 AUC ROC of Logistic Regression-Test: 41-55: 0.6797385620915033 AUC ROC of Logistic Regression-Test: 56-77: 0.7559139784946236 </pre>	<pre> AUC ROC of Decision Tree-Test:29-40: 0.8970741901776385 AUC ROC of Decision Tree-Test:41-55: 0.6803482587064675 AUC ROC of Decision Tree-Test:56-77: 0.8575851393188855 </pre>

המסקנה פה היא ברורה, המסווג שלא ביצע oversampling לא סובל מאובר-פיטינג וגם הביצועים שלו טובים יותר. המסווג שביצע oversampling גם עם ביצועים פחות טובים וגם סובל מאובר-פיטינג קל.

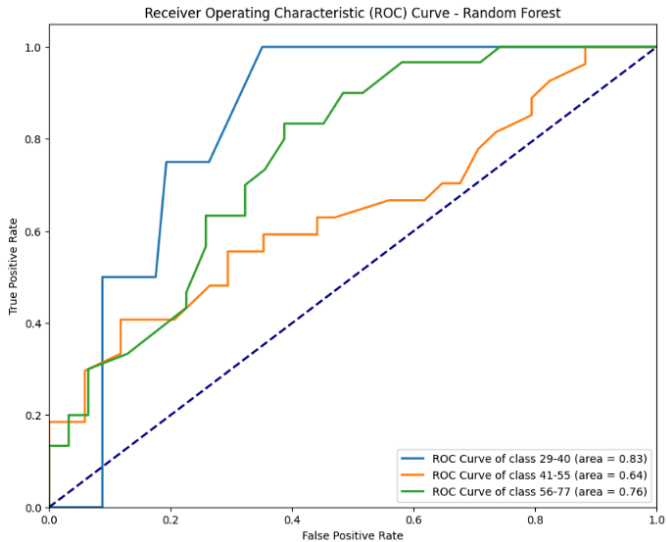
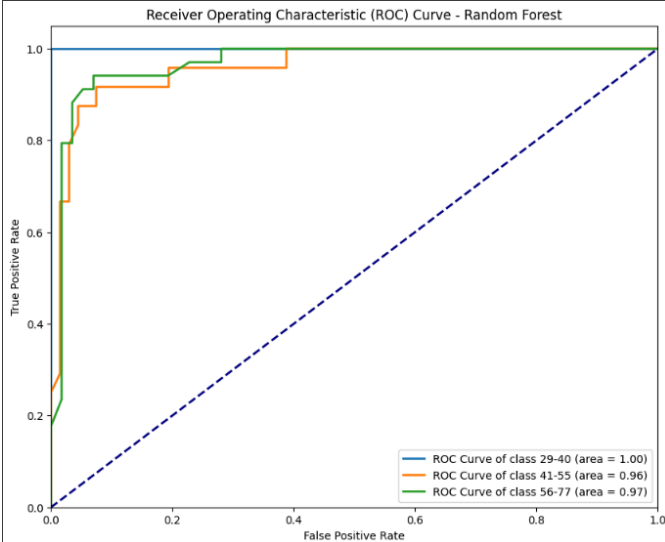
לכן המסקנה איזה מסווג עדיף פה היא ברורה. למרות, ששני המסווגים הליניארים הם לא בעלי הדיוק הגבוה ביותר והם בעדיפות נמוכה עבורנו.

מסווג Naive-Bayse:

Naive-Bayse Classification w/Normalization, with No oversampling	Naive-Bayse Classification w/Normalization, with oversampling
	
Test-Accuracy: 0.4835164835164835	Test-Accuracy: 0.2459016393442623
Train-Accuracy: 0.42265193370165743	Train-Accuracy: 0.18181818181818182
<pre>AUC ROC of Naive Bayes-Test: 29-40: 0.8490073145245559 AUC ROC of Naive Bayes-Test: 41-55: 0.6511194029850746 AUC ROC of Naive Bayes-Test: 56-77: 0.7915376676986584</pre>	<pre>AUC ROC of Naive Bayes-Test: 29-40: 0.7894736842105263 AUC ROC of Naive Bayes-Test: 41-55: 0.4281045751633987 AUC ROC of Naive Bayes-Test: 56-77: 0.7236559139784946</pre>

כפי שניתן לראות, אומנם אף אחד מהמסווגים לא סובל מאובר-פיטינג, אבל מאוד ברור שהמודל לא מתבצע כראוי והדיוק שלו הוא מאוד מאוד נמוך, ופחות עדיף אפילו מניחוש.

מסווג Random-Forest:

Random-Forest Classification w/Normalization, with No oversampling	Random-Forest Classification w/Normalization, with oversampling
	
Test-Accuracy: 0.639344262295082	Test-Accuracy: 0.8901098901098901
Train-Accuracy: 0.9958677685950413	Train-Accuracy: 1.0
<pre> AUC ROC of Naive Bayes-Test: 29-40: 0.8490073145245559 AUC ROC of Naive Bayes-Test: 41-55: 0.6511194029850746 AUC ROC of Naive Bayes-Test: 56-77: 0.7915376676986584 </pre>	<pre> AUC ROC of Random Forest-Test: 29-40: 1.0 AUC ROC of Random Forest-Test: 41-55: 0.9595771144278608 AUC ROC of Random Forest-Test: 56-77: 0.968782249742002 </pre>

**כאן ניתן לראות באופן חד-משמעי כי אומנם עם אובר-סמפלינג הדיוק של הטסט-סט הוא מעולה, אבל שניהם סובלים מoverfitting חריג במיוחד ולא תקין.**

## Fold-10

AUC ROC of Decision Tree - Test: 56-77: 0.2057

AUC ROC of Decision Tree - Test: 41-55: 0.7452

AUC ROC of Decision Tree - Test: 29-40: 0.1105

General AUC ROC of Decision Tree - Test: 0.3538

--

AUC ROC of Random Forest - Test: 56-77: 0.2440

AUC ROC of Random Forest - Test: 41-55: 0.9448

AUC ROC of Random Forest - Test: 29-40: 0.0257

General AUC ROC of Random Forest - Test: 0.4048

--



AUC ROC of Logistic Regression - Test: 56-77: 0.2120

AUC ROC of Logistic Regression - Test: 41-55: 0.6776

AUC ROC of Logistic Regression - Test: 29-40: 0.2091

General AUC ROC of Logistic Regression - Test: 0.3662

--

AUC ROC of Naive Bayes - Test: 56-77: 0.2529

AUC ROC of Naive Bayes - Test: 41-55: 0.6129

AUC ROC of Naive Bayes - Test: 29-40: 0.1776

General AUC ROC of Naive Bayes - Test: 0.3478

--

# מטרת העל של הפרוייקט

**לחזות האם אדם בעל סיכון מוגבר לקבל התקף לב או לא.**

השלב הראשון - בחינת המודלים ללא נורמליזציה

התחלנו בבחינת המודלים:

· עצי החלטה

· Knn

· Linear Classification

· Random Forest

**עצי ההחלטה:**

בחלק זה השתמשנו בשתי סוגים של עצי החלטה: id3 ו gini שהראו בהשוואה אחד לשני תוצאות מאוד דומות.

: test set accuracy

עבור id3: 0.7049180327868853

עבור gini: 0.7049180327868853

: train set accuracy

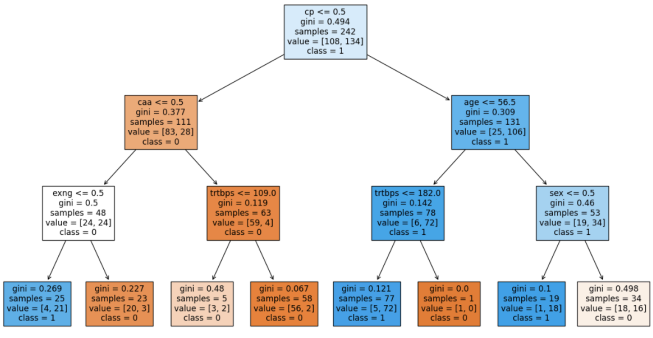
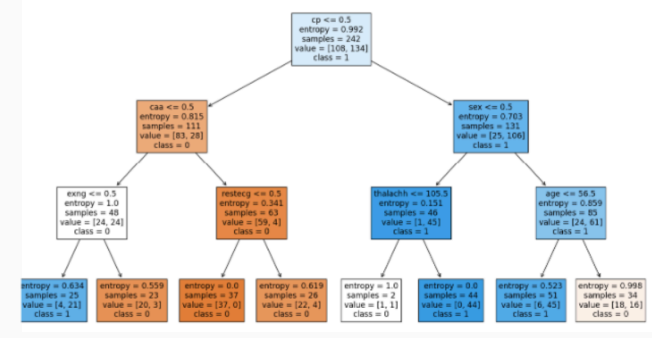
עבור id3: 0.859504132231405

עבור gini: 0.8636363636363636

: ROC AUC accuracy

id3 0.7473118279569892 עבור

gini: 0.7629032258064516 עבור

GINI	ID3
	
Test-Accuracy:0.7049180327868853	Test-Accuracy:0.7049180327868853
Train-Accuracy: 0.8636363636363636	Train-Accuracy:0.859504132231405
Accuracy of gini decision tree test set 0.7049180327868853 Accuracy of gini decision tree train set 0.8636363636363636 ROC AUC for Gini: 0.7629032258064516	Accuracy of ID3 decision tree on test set: 0.7049180327868853 Accuracy of ID3 decision tree on train set: 0.859504132231405 ROC AUC for ID3: 0.7473118279569892

## מסקנות:

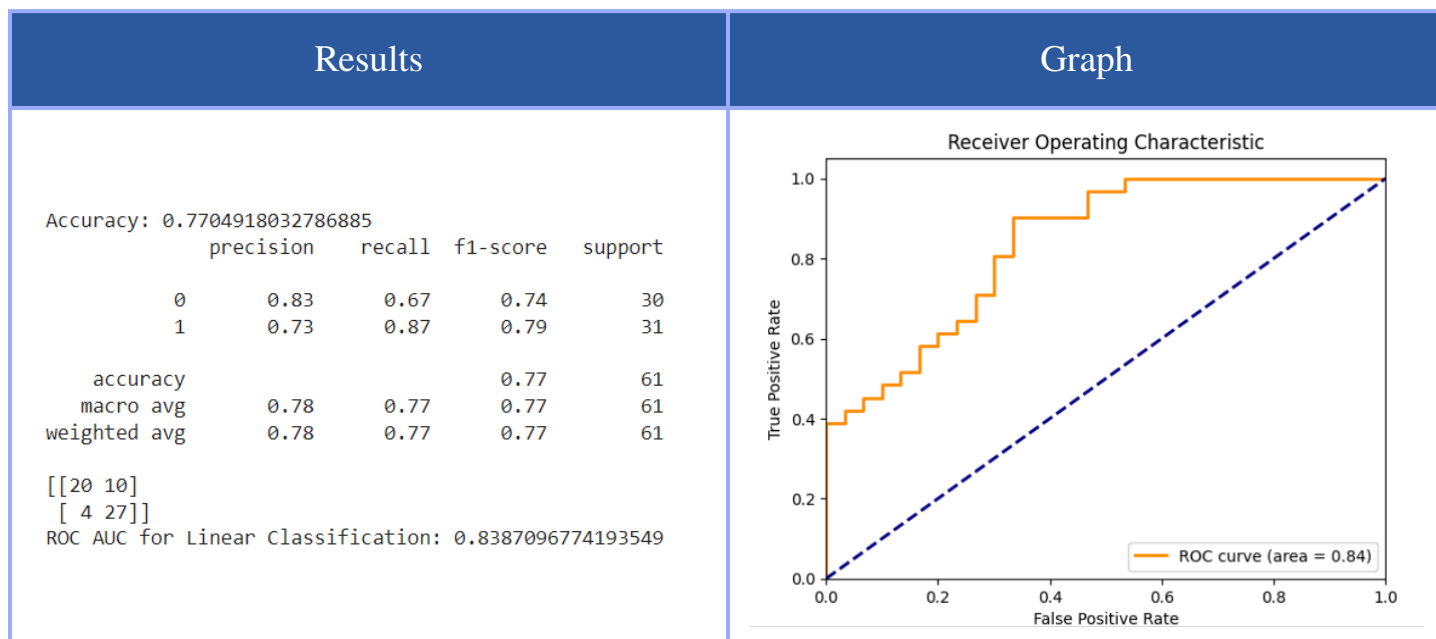
לסיכום אנחנו רואים כי ההבדלים בין עצי ההחלטה הם מאוד קטנים אך התוצאות הכלליות מראות כי אנחנו סובלים במודל זה מ overfitting לכן כאן אנחנו רואים כי מודל זה לא עזר לנו באופן מדויק לחזות את מטרת העל שלנו.

## :Linear Classification

במסווג הלינארי שהפעלנו התקבל דיוק של 77% ו-ROC AUC של 0.84 מה שמצביע על יכולת טובה של המודל להבחין בין המחלקות במונחי דיוק הסינוג המודל מציג ביצועים גבוהים יותר עבור המחלקה 0 (83%) בהשוואה למחלקה 1 (73%) כלומר כאשר המודל חוזה שמדובר במחלקה 0 הוא מדייק יותר בתחזיות שלו. עם זאת, מבחינת רגישות המודל (Recall) המודל מצליח בצורה מרשימה יותר בזיהוי מחלקה 1 עם רגישות של 87% מה שמצביע על כך שהוא מזהה את רוב הדוגמאות של מחלקה זו בצורה נכונה.

המודל מאוזן יחסית, עם F1-Score של 0.74 עבור מחלקה 0 ו-0.79 עבור מחלקה 1, מה שמדגיש את היכולת שלו לשמור על איזון בין דיוק ורגישות בשתי המחלקות. עם זאת, נצפה כי המודל מתפקד טוב יותר בזיהוי מחלקה

1, ואילו במחלקה 0 יש מקום לשיפור, במיוחד כדי להפחית את מספר הדוגמאות שמסווגות בטעות כמחלקה 1. זהו אספקט חשוב שיש לקחת בחשבון כאשר נשקול התאמות ושיפורים עתידיים במודל, כדי להגביר את הדיוק הכולל ולהפחית טעויות סיווג.



## :KNN

המודל k-Nearest Neighbors השיג דיוק של 57.38%, מה שמעיד על כך שביצועיו מוגבלים בהבחנה בין המחלקות. תוצאה זו משקפת את הקושי של המודל להבדיל בצורה מדויקת בין הדוגמאות השונות בסט הבדיקה. בנוסף, מדד ה-ROC AUC של 0.53 מצביע על כך שהיכולת של המודל להבחין בין המחלקות קרובה ליכולת אקראית, כלומר המודל אינו מצליח להפריד היטב בין המחלקות בצורה מובהקת.

ניתוח מדדי ה-Precision וה-Recall מראה שהמודל מתקשה במיוחד בזיהוי מדויק של המחלקה 0, עם רגישות נמוכה של 40% בלבד במחלקה זו. המשמעות היא שהמודל מצליח לזהות פחות מחצי מהדוגמאות השייכות למחלקה 0, מה שמוביל לאחוז טעויות גבוה כאשר הוא מנסה לסווג דוגמאות למחלקה זו. לעומת זאת, במחלקה 1 המודל מציג רגישות גבוהה יותר של 74%, כלומר, המודל מזהה נכון את רוב הדוגמאות השייכות למחלקה זו, אך עדיין ישנו אחוז טעויות לא מבוטל.

בסך הכל, התוצאות מצביעות על כך שהמודל kNN מתקשה להכליל את הלמידה שלו בצורה מספקת על נתוני הבדיקה. הקושי שלו בזיהוי מחלקה 0 והקרבה ליכולת אקראית במבחן ה-ROC AUC מעידים על כך שיש מקום לשיפור משמעותי במודל, בין אם על ידי שינוי המאפיינים של המודל, או בחינת מודלים אחרים שיכולים להתמודד טוב יותר עם המשימה.

## Results

```
Accuracy: 0.5737704918032787
      precision    recall  f1-score   support

     0       0.60      0.40      0.48        30
     1       0.56      0.74      0.64        31

 accuracy          0.57        61
 macro avg         0.58      0.57      0.56        61
 weighted avg      0.58      0.57      0.56        61

[[12 18]
 [ 8 23]]
ROC AUC for k-Nearest Neighbors: 0.5360215053763441
```

### Random Forest:

מודל ה-Random Forest השיג דיוק של 75.41%, מה שמעיד על יכולת טובה להבחין בין המחלקות ולהכליל את הלמידה שלו על נתוני הבדיקה. דיוק זה מצביע על כך שכשלושה רבעים מהדוגמאות סווגו בצורה נכונה, מה שמראה שהמודל מצליח להבחין בצורה משמעותית בין המחלקות השונות. המדד ROC AUC של 0.83 מחזק את התובנה הזו, ומצביע על כך שהמודל מצליח להבדיל היטב בין הדוגמאות השייכות למחלקות השונות, עם יכולת הבחנה ברורה בין הדוגמאות החיוביות והשליליות.

מבחינת Recall-Precision, המודל מציג איזון טוב בין המחלקות. עבור מחלקה 0, ה-Precision עומד על 78% וה-Recall על 70%, מה שמעיד על כך שהמודל מצליח לזהות בצורה נכונה את רוב הדוגמאות השייכות למחלקה זו, אם כי ישנם מקרים בהם הוא מסווג בטעות דוגמאות השייכות למחלקה 1 כ-0. במחלקה 1, ה-Recall עומד על 81%, כלומר המודל מזהה את רוב הדוגמאות השייכות למחלקה זו, וה-Precision של 74% מצביע על כך שרוב התחזיות למחלקה זו היו נכונות.

בנוסף, המודל מציג F1-Score של 0.74 עבור מחלקה 0 ו-0.77 עבור מחלקה 1, מה שמצביע על איזון טוב בין דיוק (Precision) ורגישות (Recall) בשתי המחלקות. מדד זה, שמאזן בין היכולת לזהות נכון דוגמאות חיוביות לבין היכולת להימנע מסיווג שגוי של דוגמאות שליליות, מצביע על כך שהמודל מצליח לסווג בצורה טובה יחסית את רוב הדוגמאות.

בסך הכל, התוצאות מראות שהמודל Random Forest הוא מודל חזק ומאוזן, עם יכולת להבחין היטב בין המחלקות השונות ולהכליל את הלמידה שלו על נתונים חדשים. הביצועים המאוזנים בין המחלקות מראים שהמודל מתפקד בצורה יציבה ומספקת, עם פוטנציאל לשימוש במשימות סיווג דומות נוספות.

## Results

Accuracy: 0.7540983606557377

	precision	recall	f1-score	support
0	0.78	0.70	0.74	30
1	0.74	0.81	0.77	31
accuracy			0.75	61
macro avg	0.76	0.75	0.75	61
weighted avg	0.76	0.75	0.75	61

[[21 9]

[ 6 25]]

ROC AUC for Random Forest: 0.8360215053763441

## השלב השני - ביצוע נורמליזציה על מאפייני הדאטה סט

בחירת סט מאפיינים ספציפי על מנת לחזות בצורה יותר טובה ובדיוק יותר טוב את עמודת המטרה.

המאפיינים שבחרנו:

גיל(age)

דופק מקסימלי שהושג(thalachh)

דיכאון ST שנגרם על ידי פעילות גופנית יחסית למנוחה(oldpeak)

מספר כלי הדם העיקריים שנצבעו בעזרת פלואורוסופיה(caa)

תוצאה של הבדיקה הרפואי תלסמיה(thall)

עץ החלטה:

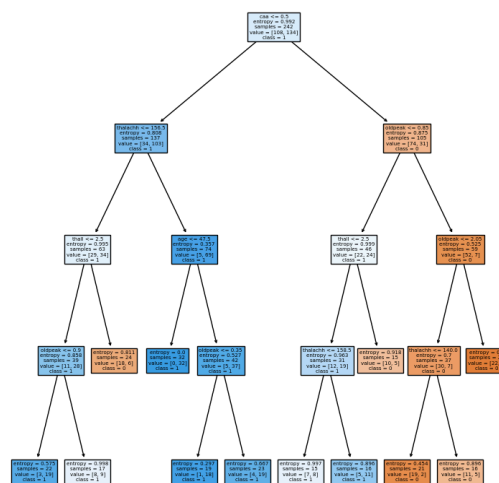
בחלק זה השתמשנו בעץ החלטה מסוג ID3

: test set accuracy  
0.8032786885245902

: train set accuracy  
0.8099173553719008

: ROC AUC accuracy  
0.7876344086021505

## ID3



Test-Accuracy : 0.8032786885245902

Train-Accuracy : 0.8099173553719008

Accuracy of id3 decision tree test set 0.8032786885245902  
Accuracy of id3 decision tree train set 0.8099173553719008  
ROC AUC for ID3: 0.7876344086021505

## מסקנות:

המודל ID3 Decision Tree שנבנה לאחר נורמליזציה של מאפייני הדאטה סט ובחירת סט מאפיינים ספציפי, הציג תוצאות חיוביות המעידות על איזון טוב בין דיוק בסט האימון ובסט הבדיקה. דיוק המודל בסט הבדיקה עמד על 80.33%, מה שמראה שהמודל מצליח לסווג נכון את רוב הדוגמאות החדשות בצורה עקבית. הדיוק בסט האימון

היה דומה מאוד, 80.99%, מה שמעיד על כך שהמודל אינו סובל מ-Overfitting משמעותי, והוא מצליח להכליל את הלמידה שלו על נתונים חדשים בצורה טובה.

מדד ה-ROC AUC עמד על 0.79, מה שמצביע על כך שהמודל מצליח להבחין בצורה טובה בין המחלקות השונות, עם יכולת הבחנה גבוהה יחסית בין הדוגמאות החיוביות לשליליות. בסך הכל, ניתן לומר שהמודל ID3 Decision Tree שנבנה על בסיס המאפיינים שנבחרו לאחר נורמליזציה מציג ביצועים יציבים ומאוזנים, ומצליח לסווג בצורה מדויקת את הדוגמאות בסט הבדיקה, תוך שמירה על איזון טוב בין ביצועי האימון והבדיקה.

## :Linear Classification

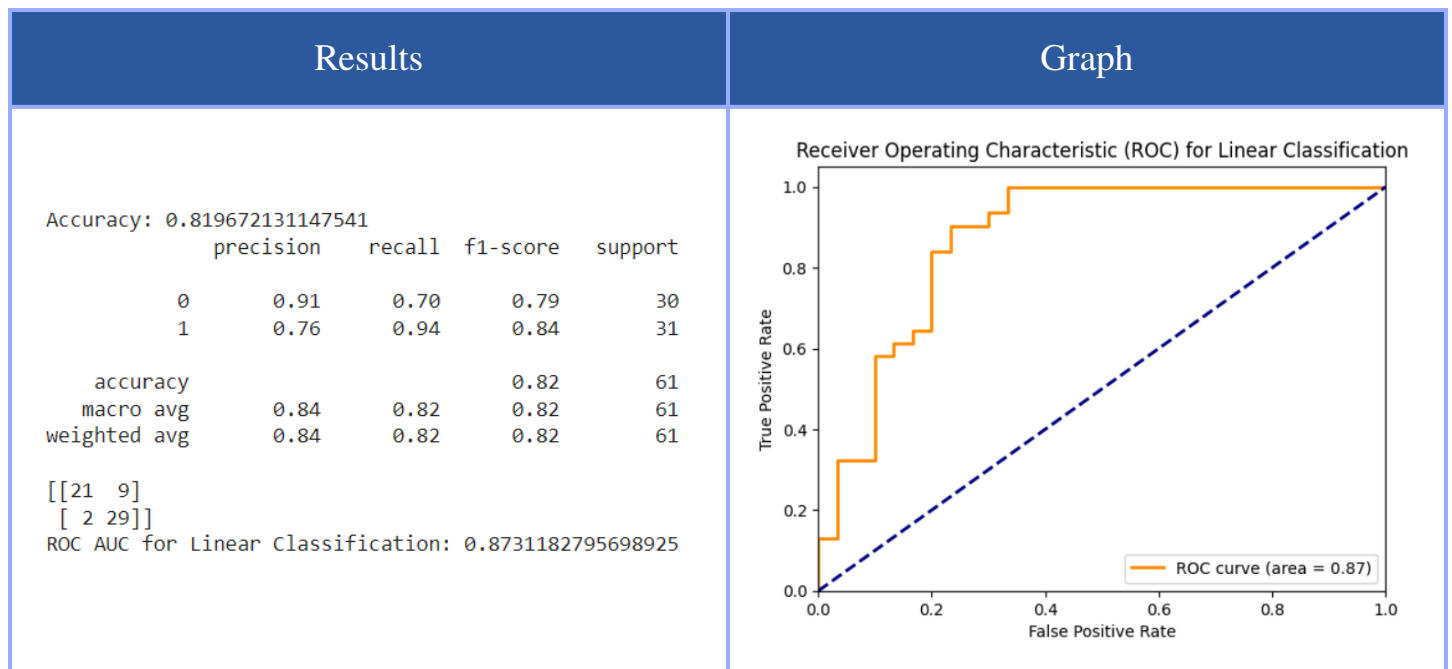
המסווג הליניארי שהפעלנו הציג ביצועים מרשימים עם דיוק של 81.97% מה שמעיד על יכולתו להבחין בצורה טובה בין הדוגמאות השייכות למחלקות השונות.

דיוק זה מצביע על כך שהמודל מצליח לסווג נכון את רוב הדוגמאות, ומספק תחזיות אמינות על סמך הנתונים שהוזנו לו. בנוסף, מדד ה-ROC AUC של 0.87% מעיד על כך שהמודל מצליח להבחין בצורה יעילה בין מחלקה 0 למחלקה 1, עם יכולת גבוהה להפריד בין הדוגמאות החיוביות והשליליות. משמעות ה-ROC AUC הגבוהה היא שהמודל מצליח להבחין באופן עקבי בין הדוגמאות השייכות לשתי המחלקות, תוך הפחתת הסיכון לטעויות סיווג.

במונחי Precision, המחלקות מציגות הבדלים מעניינים: מחלקה 0 נהנית מ-Precision של 91%, מה שמעיד על כך שכאשר המודל חוזר בדובר בדוגמה השייכת למחלקה 0, הוא מדייק מאוד בתחזיותיו, ורוב התחזיות האלו נכונות. לעומת זאת, במחלקה 1, ה-Precision עומד על 76%, מה שמעיד על כך שבמחלקה זו יש עדיין מקום לשיפור, במיוחד בהפחתת שיעור התחזיות השגויות.

המגוון הרחב של המידע שמספק המודל מצביע על איזון טוב בין דיוק ורגישות (Recall), מה שמעיד על כך שהמודל לא רק מצליח לזהות נכון את רוב הדוגמאות מכל מחלקה, אלא גם שומר על איזון בין היכולת להימנע מטעויות סיווג חיוביות שגויות לבין היכולת לזהות את הדוגמאות החיוביות בצורה נכונה.

בסך הכל, המסווג הליניארי מציג ביצועים חזקים ואיזון בסיווג הדוגמאות בשתי המחלקות, ומצליח להתמודד היטב עם המשימה של סיווג הדוגמאות בצורה מהימנה. התוצאות מעידות על כך שהמודל מתאים מאוד למשימה הנוכחית, עם פוטנציאל להמשך שיפור והתאמה במידת הצורך.



## :KNN

במודל ה-kNN התקבל דיוק של 63.93%, המצביע על כך שהמודל מצליח לסווג נכון כ-64% מהדוגמאות בסט הבדיקה. דיוק זה מצביע על כך שהמודל מתקשה להבחין בין המחלקות השונות בנתונים ומציג יכולת מוגבלת להכליל את הלמידה שלו על נתונים חדשים.

מדד ה-ROC AUC של 0.67 מעיד על כך שהמודל מצליח להבחין בין המחלקות בצורה מתונה, אך רחוק מביצועים אידיאליים. ה-Precision עבור מחלקה 0 הוא 70%, אך הרגישות נמוכה, מה שמעיד על כך שהמודל אינו מזהה את כל הדוגמאות השייכות למחלקה זו בצורה טובה, מה שמוביל לטעויות סיווג.

במחלקה 1, המודל מציג רגישות גבוהה יותר של 81%, מה שמצביע על יכולת טובה יותר לזהות דוגמאות השייכות למחלקה זו. עם זאת, ה-Precision במחלקה זו נמוך יותר ועומד על 61%, מה שמצביע על כך שחלק מהתחזיות של המודל כמחלקה 1 אינן מדויקות. בסך הכל, המודל מציג ביצועים בינוניים, כאשר הוא מצליח יחסית בזיהוי מחלקה 1 אך מתקשה בזיהוי נכון של מחלקה 0, ומכאן נובע הצורך בשיפור הביצועים הכוללים.



## Results

```

Accuracy: 0.639344262295082
      precision    recall  f1-score   support

     0       0.70      0.47      0.56         30
     1       0.61      0.81      0.69         31

   accuracy          0.64         61
  macro avg       0.65      0.64      0.63         61
 weighted avg       0.65      0.64      0.63         61

[[14 16]
 [ 6 25]]
ROC AUC for k-Nearest Neighbors: 0.6682795698924732

```

## Random Forest:

מודל ה-Random Forest השיג דיוק של 81.97%, מה שמעיד על יכולת גבוהה לסווג נכון את הדוגמאות בסט הבדיקה. דיוק זה מצביע על כך שהמודל מצליח להכליל היטב את הלמידה שלו על נתונים חדשים, ומספק תחזיות מהימנות ומדויקות. בנוסף, מדד ה-ROC AUC של 0.87 מראה שהמודל מצליח להבחין בצורה מצוינת בין המחלקות השונות, עם יכולת הבחנה ברורה בין הדוגמאות השייכות למחלקה החיובית לאלו השייכות למחלקה השלילית. ערך זה קרוב למושלם ומעיד על כך שהמודל מסוגל לסווג את הדוגמאות בצורה מהימנה ועקבית.

במונחי Precision ו-Recall, המודל מציג ביצועים מרשימים בשתי המחלקות. במחלקה 0, ה-Precision עומד על 88% עם רגישות של 73%, מה שמעיד על כך שהמודל מדויק מאוד בזיהוי דוגמאות השייכות למחלקה זו. עם זאת, רגישות של 73% מצביעה על כך שהמודל עלול לפספס חלק מהדוגמאות השייכות למחלקה 0, כלומר ישנם מקרים שבהם הדוגמאות סווגו למחלקה הלא נכונה. לעומת זאת, במחלקה 1, המודל מציג רגישות גבוהה של 90% יחד עם Precision של 78%, מה שמצביע על יכולת מצוינת לזהות את רוב הדוגמאות השייכות למחלקה זו, אם כי ישנם עדיין מקרים שבהם התחזיות למחלקה 1 לא היו מדויקות במלואן.

בסך הכל, המודל מציג ביצועים מאוזנים וחזקים, תוך שמירה על איזון טוב בין דיוק ורגישות בשתי המחלקות. המודל מסוגל לספק תחזיות מדויקות ואמינות, תוך הפחתת שיעור הטעויות למינימום, מה שהופך אותו למתאים במיוחד למשימות סיווג הדורשות איזון בין זיהוי נכון של דוגמאות חיוביות ושליליות. התוצאות מראות כי המודל מתאים מאוד להמשך שימוש במשימות דומות, עם פוטנציאל להרחבה ושיפור נוסף במידת הצורך.

## Results

```
Accuracy: 0.819672131147541
      precision    recall  f1-score   support

     0       0.88      0.73      0.80         30
     1       0.78      0.90      0.84         31

 accuracy          0.82         61
  macro avg       0.83      0.82      0.82         61
 weighted avg     0.83      0.82      0.82         61

[[22  8]
 [ 3 28]]
ROC AUC for Random Forest: 0.8715053763440861
```

## לסיכום

במהלך העבודה על המודלים השונים, נצפו הבדלים משמעותיים בביצועים בהתאם לסוג המודל ושימוש בנורמליזציה של הנתונים. מודל ה-Random Forest הציג את הביצועים החזקים והמאוזנים ביותר, עם יכולת גבוהה להבחין בין המחלקות השונות ולהכליל את הלמידה שלו על נתוני הבדיקה. המסווג הליניארי הציג גם הוא ביצועים מרשימים, במיוחד בזיהוי מדויק של מחלקה 1, אך עם מקום לשיפור בזיהוי מחלקה 0. לעומת זאת, מודל ה-kNN הציג ביצועים בינוניים, עם קושי משמעותי בזיהוי נכון של מחלקה 0, ויכולתו להבחין בין המחלקות התבררה כקרובה ליכולת אקראית. עצי ההחלטה, הן ID3 והן Gini, הציגו ביצועים דומים מאוד, אך סבלו מבעיה של overfitting, מה שמצביע על כך שהם פחות מתאימים למשימת הסיווג הספציפית ללא התאמות נוספות.

לאחר ביצוע הנורמליזציה על מאפייני הדאטה סט, נצפה שיפור בביצועים של חלק מהמודלים, ובמיוחד במודל ה-ID3 Decision Tree, שהפך למאוזן יותר והפחית את הנטייה ל-overfitting. ההשפעה של הנורמליזציה בלטה גם בביצועים של המסווג הליניארי, שהצליח להבחין בין המחלקות בצורה טובה יותר לאחר הנורמליזציה. המסקנה העיקרית מהניתוח היא שלנורמליזציה של הנתונים יש השפעה מכרעת על הביצועים של חלק מהמודלים, וכי יש לבחון אותה כשלב קריטי בעיבוד הנתונים לפני אימון מודלים. בנוסף, מודל ה-Random Forest הוכיח עצמו ככלי חזק ואמין לסיווג במקרים שבהם נדרש איזון בין דיוק ורגישות, מה שמצביע על פוטנציאל לשימוש בו במשימות סיווג נוספות בעתיד.