זייד חביבאללה
322513433
למידת מכונה – מטלה 1

Decision Tree

Try 1 – decision tree

זה הניסוי הראשון של המודל decision tree,
עדכנו פרמטרים שונים לשני המודלים האחרים random forest – adaboost
והפרמטרים לא ישתנו לאורך חמישת הניסויים עבור המודל decision tree

Decision tree

```
# Define parameter values
criterion = 'gini'
splitter = 'best'
max_depth = 10
min_samples_split = 5
min_samples_leaf = 2
```

AdaBoost

```
# Define parameter values
n_estimators = 100
learning_rate = 0.1
estimator = DecisionTreeClassifier(max_depth=3)
```

Random Forest

```
# Define parameter values
n_estimators = 100
criterion = 'gini'
max_depth = 10
min_samples_split = 5
min_samples_leaf = 2
```

```
[8] --- Decision Tree ---
    Accuracy: 0.9298
    Classification Report:
                  precision    recall  f1-score   support

               0       0.91      0.91      0.91        43
               1       0.94      0.94      0.94        71

        accuracy                           0.93       114
       macro avg       0.93      0.93      0.93       114
    weighted avg       0.93      0.93      0.93       114

    Confusion Matrix:
    [[39  4]
     [ 4 67]]

    --- Random Forest ---
    Accuracy: 0.9649
    Classification Report:
                  precision    recall  f1-score   support

               0       0.98      0.93      0.95        43
               1       0.96      0.99      0.97        71

        accuracy                           0.96       114
       macro avg       0.97      0.96      0.96       114
    weighted avg       0.97      0.96      0.96       114

    Confusion Matrix:
    [[40  3]
     [ 1 70]]
```

```
    --- AdaBoost ---
    Accuracy: 0.9211
    Classification Report:
                  precision    recall  f1-score   support

               0       0.87      0.93      0.90        43
               1       0.96      0.92      0.94        71

        accuracy                           0.92       114
       macro avg       0.91      0.92      0.92       114
    weighted avg       0.92      0.92      0.92       114

    Confusion Matrix:
    [[40  3]
     [ 6 65]]

    The best model based on accuracy is: Random Forest
```

Try 2 – decision tree

Decision tree

```
# Define parameter values
criterion = 'gini'
splitter = 'random'
max_depth = 20
min_samples_split = 5
min_samples_leaf = 4
```

splitter – max depth – min leaf נעשה שינוי

AdaBoost

```
# Define parameter values
n_estimators = 100
learning_rate = 0.1
estimator = DecisionTreeClassifier(max_depth=3)
```

Random Forest

```
# Define parameter values
n_estimators = 100
criterion = 'gini'
max_depth = 10
min_samples_split = 5
min_samples_leaf = 2
```

```
--- Decision Tree ---
Accuracy: 0.9649
Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.95      0.95        43
           1       0.97      0.97      0.97        71

    accuracy                           0.96       114
   macro avg       0.96      0.96      0.96       114
weighted avg       0.96      0.96      0.96       114

Confusion Matrix:
[[41  2]
 [ 2 69]]

--- Random Forest ---
Accuracy: 0.9649
Classification Report:
              precision    recall  f1-score   support

           0       0.98      0.93      0.95        43
           1       0.96      0.99      0.97        71

    accuracy                           0.96       114
   macro avg       0.97      0.96      0.96       114
weighted avg       0.97      0.96      0.96       114

Confusion Matrix:
[[40  3]
 [ 1 70]]
```

```
--- AdaBoost ---
Accuracy: 0.9211
Classification Report:
              precision    recall  f1-score   support

           0       0.87      0.93      0.90        43
           1       0.96      0.92      0.94        71

    accuracy                           0.92       114
   macro avg       0.91      0.92      0.92       114
weighted avg       0.92      0.92      0.92       114

Confusion Matrix:
[[40  3]
 [ 6 65]]

The best model based on accuracy is: Decision Tree
```

Try 3 – decision tree

Decision tree

```
# Define parameter values
criterion = 'entropy'
splitter = 'random'
max_depth = 20
min_samples_split = 10
min_samples_leaf = 4
```

נעשה שינוי criterion – min split

AdaBoost

```
# Define parameter values
n_estimators = 100
learning_rate = 0.1
estimator = DecisionTreeClassifier(max_depth=3)
```

Random Forest

```
# Define parameter values
n_estimators = 100
criterion = 'gini'
max_depth = 10
min_samples_split = 5
min_samples_leaf = 2
```

```
--- Decision Tree ---
Accuracy: 0.9825
Classification Report:
              precision    recall  f1-score   support

           0       0.96      1.00      0.98        43
           1       1.00      0.97      0.99        71

    accuracy                           0.98       114
   macro avg       0.98      0.99      0.98       114
weighted avg       0.98      0.98      0.98       114

Confusion Matrix:
[[43  0]
 [ 2 69]]

--- Random Forest ---
Accuracy: 0.9649
Classification Report:
              precision    recall  f1-score   support

           0       0.98      0.93      0.95        43
           1       0.96      0.99      0.97        71

    accuracy                           0.96       114
   macro avg       0.97      0.96      0.96       114
weighted avg       0.97      0.96      0.96       114

Confusion Matrix:
[[40  3]
 [ 1 70]]
```

```
--- AdaBoost ---
Accuracy: 0.9211
Classification Report:
              precision    recall  f1-score   support

           0       0.87      0.93      0.90        43
           1       0.96      0.92      0.94        71

    accuracy                           0.92       114
   macro avg       0.91      0.92      0.92       114
weighted avg       0.92      0.92      0.92       114

Confusion Matrix:
[[40  3]
 [ 6 65]]

The best model based on accuracy is: Decision Tree
```

Try 4 – decision tree

Decision tree

```
# Define parameter values
criterion = 'entropy'
splitter = 'best'
max_depth = 10
min_samples_split = 5
min_samples_leaf = 4
```

נעשה שינוי splitter – max depth – min split

AdaBoost

```
# Define parameter values
n_estimators = 100
learning_rate = 0.1
estimator = DecisionTreeClassifier(max_depth=3)
```

Random Forest

```
# Define parameter values
n_estimators = 100
criterion = 'gini'
max_depth = 10
min_samples_split = 5
min_samples_leaf = 2
```

```
--- Decision Tree ---
Accuracy: 0.9561
Classification Report:
              precision    recall  f1-score   support

           0       0.97      0.91      0.94        43
           1       0.95      0.99      0.97        71

    accuracy                           0.96       114
   macro avg       0.96      0.95      0.95       114
weighted avg       0.96      0.96      0.96       114

Confusion Matrix:
[[39  4]
 [ 1 70]]

--- Random Forest ---
Accuracy: 0.9649
Classification Report:
              precision    recall  f1-score   support

           0       0.98      0.93      0.95        43
           1       0.96      0.99      0.97        71

    accuracy                           0.96       114
   macro avg       0.97      0.96      0.96       114
weighted avg       0.97      0.96      0.96       114

Confusion Matrix:
[[40  3]
 [ 1 70]]
```

```
--- AdaBoost ---
Accuracy: 0.9211
Classification Report:
              precision    recall  f1-score   support

           0       0.87      0.93      0.90        43
           1       0.96      0.92      0.94        71

    accuracy                           0.92       114
   macro avg       0.91      0.92      0.92       114
weighted avg       0.92      0.92      0.92       114

Confusion Matrix:
[[40  3]
 [ 6 65]]

The best model based on accuracy is: Random Forest
```

Try 5 – decision tree

Decision tree

```
# Define parameter values
criterion = 'entropy'
splitter = 'random'
max_depth = 25
min_samples_split = 10
min_samples_leaf = 5
```

splitter – max depth – min leaf – min split נעשה שינוי

AdaBoost

```
# Define parameter values
n_estimators = 100
learning_rate = 0.1
estimator = DecisionTreeClassifier(max_depth=3)
```

Random Forest

```
# Define parameter values
n_estimators = 100
criterion = 'gini'
max_depth = 10
min_samples_split = 5
min_samples_leaf = 2
```

```
--- Decision Tree ---
Accuracy: 0.9737
Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.98      0.97        43
           1       0.99      0.97      0.98        71

    accuracy                           0.97       114
   macro avg       0.97      0.97      0.97       114
weighted avg       0.97      0.97      0.97       114

Confusion Matrix:
[[42  1]
 [ 2 69]]

--- Random Forest ---
Accuracy: 0.9649
Classification Report:
              precision    recall  f1-score   support

           0       0.98      0.93      0.95        43
           1       0.96      0.99      0.97        71

    accuracy                           0.96       114
   macro avg       0.97      0.96      0.96       114
weighted avg       0.97      0.96      0.96       114

Confusion Matrix:
[[40  3]
 [ 1 70]]
```

```
--- AdaBoost ---
Accuracy: 0.9211
Classification Report:
              precision    recall  f1-score   support

           0       0.87      0.93      0.90        43
           1       0.96      0.92      0.94        71

    accuracy                           0.92       114
   macro avg       0.91      0.92      0.92       114
weighted avg       0.92      0.92      0.92       114

Confusion Matrix:
[[40  3]
 [ 6 65]]

The best model based on accuracy is: Decision Tree
```

Random forest

Try 1 – Random Forest

Random Forest

```
# Define parameter value
n_estimators = 200
criterion = 'entropy'
max_depth = 30
min_samples_split = 15
min_samples_leaf = 10
```

כרגע נתחיל עם הניסוי הראשון של המודל random forest
אחרי שסיימנו לאמן את המודל decision tree ניקח את הפרמטרים שהראו את התוצאה הכי טובה,
ונתחיל לעדכן בכל ניסוי את הפרמטרים של random forest

AdaBoost

```
# Define parameter values
n_estimators = 100
learning_rate = 0.1
estimator = DecisionTreeClassifier(max_depth=3)
```

Decision tree

```
# Define parameter values
criterion = 'entropy'
splitter = 'random'
max_depth = 20
min_samples_split = 10
min_samples_leaf = 4
```

```
--- Decision Tree ---
Accuracy: 0.9825
Classification Report:
              precision    recall  f1-score   support

           0       0.96      1.00      0.98        43
           1       1.00      0.97      0.99        71

    accuracy                           0.98       114
   macro avg       0.98      0.99      0.98       114
weighted avg       0.98      0.98      0.98       114

Confusion Matrix:
[[43  0]
 [ 2 69]]

--- Random Forest ---
Accuracy: 0.9649
Classification Report:
              precision    recall  f1-score   support

           0       0.98      0.93      0.95        43
           1       0.96      0.99      0.97        71

    accuracy                           0.96       114
   macro avg       0.97      0.96      0.96       114
weighted avg       0.97      0.96      0.96       114

Confusion Matrix:
[[40  3]
 [ 1 70]]
```

```
--- AdaBoost ---
Accuracy: 0.9211
Classification Report:
              precision    recall  f1-score   support

           0       0.87      0.93      0.90        43
           1       0.96      0.92      0.94        71

    accuracy                           0.92       114
   macro avg       0.91      0.92      0.92       114
weighted avg       0.92      0.92      0.92       114

Confusion Matrix:
[[40  3]
 [ 6 65]]

The best model based on accuracy is: Decision Tree
```

Try 2 – Random Forest

Random Forest

```
# Define parameter values
n_estimators = 30
criterion = 'entropy'
max_depth = 3
min_samples_split = 50
min_samples_leaf = 25
```

נעשה שינוי בכל הערכים של הפרמטרים חוץ מה criterion

AdaBoost

```
# Define parameter values
n_estimators = 100
learning_rate = 0.1
estimator = DecisionTreeClassifier(max_depth=3)
```

Decision tree

```
# Define parameter values
criterion = 'entropy'
splitter = 'random'
max_depth = 20
min_samples_split = 10
min_samples_leaf = 4
```

```
--- Decision Tree ---
Accuracy: 0.9825
Classification Report:
              precision    recall  f1-score   support

           0       0.96      1.00      0.98        43
           1       1.00      0.97      0.99        71

    accuracy                           0.98       114
   macro avg       0.98      0.99      0.98       114
weighted avg       0.98      0.98      0.98       114

Confusion Matrix:
[[43  0]
 [ 2 69]]

--- Random Forest ---
Accuracy: 0.9561
Classification Report:
              precision    recall  f1-score   support

           0       0.97      0.91      0.94        43
           1       0.95      0.99      0.97        71

    accuracy                           0.96       114
   macro avg       0.96      0.95      0.95       114
weighted avg       0.96      0.96      0.96       114

Confusion Matrix:
[[39  4]
 [ 1 70]]
```

```
--- AdaBoost ---
Accuracy: 0.9211
Classification Report:
              precision    recall  f1-score   support

           0       0.87      0.93      0.90        43
           1       0.96      0.92      0.94        71

    accuracy                           0.92       114
   macro avg       0.91      0.92      0.92       114
weighted avg       0.92      0.92      0.92       114

Confusion Matrix:
[[40  3]
 [ 6 65]]

The best model based on accuracy is: Decision Tree
```

Try 3 – Random Forest

Random Forest

```
# Define parameter values
n_estimators = 500
criterion = 'gini'
max_depth = 40
min_samples_split = 2
min_samples_leaf = 1
```

נעשה שינוי בכל הערכים של הפרמטרים
למרות השינויים בכל הפרמטרים, הוא נתן אותה תשובה כמו בניסוי הראשון
0.9649

AdaBoost

```
# Define parameter values
n_estimators = 100
learning_rate = 0.1
estimator = DecisionTreeClassifier(max_depth=3)
```

Decision tree

```
# Define parameter values
criterion = 'entropy'
splitter = 'random'
max_depth = 20
min_samples_split = 10
min_samples_leaf = 4
```

```
--- Decision Tree ---
Accuracy: 0.9825
Classification Report:
              precision    recall  f1-score   support

           0       0.96      1.00      0.98        43
           1       1.00      0.97      0.99        71

    accuracy                           0.98       114
   macro avg       0.98      0.99      0.98       114
weighted avg       0.98      0.98      0.98       114

Confusion Matrix:
[[43  0]
 [ 2 69]]

--- Random Forest ---
Accuracy: 0.9649
Classification Report:
              precision    recall  f1-score   support

           0       0.98      0.93      0.95        43
           1       0.96      0.99      0.97        71

    accuracy                           0.96       114
   macro avg       0.97      0.96      0.96       114
weighted avg       0.97      0.96      0.96       114

Confusion Matrix:
[[40  3]
 [ 1 70]]
```

```
--- AdaBoost ---
Accuracy: 0.9211
Classification Report:
              precision    recall  f1-score   support

           0       0.87      0.93      0.90        43
           1       0.96      0.92      0.94        71

    accuracy                           0.92       114
   macro avg       0.91      0.92      0.92       114
weighted avg       0.92      0.92      0.92       114

Confusion Matrix:
[[40  3]
 [ 6 65]]

The best model based on accuracy is: Decision Tree
```

Try 4 – Random Forest

נעשה שינוי בכל הערכים של הפרמטרים

Random Forest

```
# Define parameter values
n_estimators = 60
criterion = 'entropy'
max_depth = 12
min_samples_split = 50
min_samples_leaf = 20
```

Decision tree

```
# Define parameter values
criterion = 'entropy'
splitter = 'random'
max_depth = 20
min_samples_split = 10
min_samples_leaf = 4
```

AdaBoost

```
# Define parameter values
n_estimators = 100
learning_rate = 0.1
estimator = DecisionTreeClassifier(max_depth=3)
```

```
--- Decision Tree ---
Accuracy: 0.9825
Classification Report:
              precision    recall  f1-score   support

           0       0.96      1.00      0.98        43
           1       1.00      0.97      0.99        71

    accuracy                           0.98       114
   macro avg       0.98      0.99      0.98       114
weighted avg       0.98      0.98      0.98       114

Confusion Matrix:
[[43  0]
 [ 2 69]]

--- Random Forest ---
Accuracy: 0.9474
Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.91      0.93        43
           1       0.95      0.97      0.96        71

    accuracy                           0.95       114
   macro avg       0.95      0.94      0.94       114
weighted avg       0.95      0.95      0.95       114

Confusion Matrix:
[[39  4]
 [ 2 69]]
```

```
--- AdaBoost ---
Accuracy: 0.9211
Classification Report:
              precision    recall  f1-score   support

           0       0.87      0.93      0.90        43
           1       0.96      0.92      0.94        71

    accuracy                           0.92       114
   macro avg       0.91      0.92      0.92       114
weighted avg       0.92      0.92      0.92       114

Confusion Matrix:
[[40  3]
 [ 6 65]]

The best model based on accuracy is: Decision Tree
```

Try 5 – Random Forest

Random Forest

```
# Define parameter values
n_estimators = 280
criterion = 'gini'
max_depth = 12
min_samples_split = 50
min_samples_leaf = 40
```

נעשה שינוי ב- n estimators - criterion – min leaf -

AdaBoost

```
# Define parameter values
n_estimators = 100
learning_rate = 0.1
estimator = DecisionTreeClassifier(max_depth=3)
```

Decision tree

```
# Define parameter values
criterion = 'entropy'
splitter = 'random'
max_depth = 20
min_samples_split = 10
min_samples_leaf = 4
```

```
--- Decision Tree ---
Accuracy: 0.9825
Classification Report:
              precision    recall  f1-score   support

           0       0.96      1.00      0.98        43
           1       1.00      0.97      0.99        71

    accuracy                           0.98       114
   macro avg       0.98      0.99      0.98       114
weighted avg       0.98      0.98      0.98       114

Confusion Matrix:
[[43  0]
 [ 2 69]]

--- Random Forest ---
Accuracy: 0.9737
Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.93      0.96        43
           1       0.96      1.00      0.98        71

    accuracy                           0.97       114
   macro avg       0.98      0.97      0.97       114
weighted avg       0.97      0.97      0.97       114

Confusion Matrix:
[[40  3]
 [ 0 71]]
```

```
--- AdaBoost ---
Accuracy: 0.9211
Classification Report:
              precision    recall  f1-score   support

           0       0.87      0.93      0.90        43
           1       0.96      0.92      0.94        71

    accuracy                           0.92       114
   macro avg       0.91      0.92      0.92       114
weighted avg       0.92      0.92      0.92       114

Confusion Matrix:
[[40  3]
 [ 6 65]]

The best model based on accuracy is: Decision Tree
```

AdaBoost

Try 1 – AdaBoost

כרגע נתחיל עם הניסוי הראשון של המודל adaboost
אחרי שסיימנו לאמן את המודל decision tree + random forest ניקח את הפרמטרים שהראו את התוצאות הכי טובות עבור שני המודלים שאמנו,
ונתחיל לעדכן בכל ניסוי את הפרמטרים של adaboost

AdaBoost

```
# Define parameter values
n_estimators = 250
learning_rate = 0.1
estimator = DecisionTreeClassifier(max_depth=3)
```

Random Forest

```
# Define parameter values
n_estimators = 280
criterion = 'gini'
max_depth = 12
min_samples_split = 50
min_samples_leaf = 40
```

Decision tree

```
# Define parameter values
criterion = 'entropy'
splitter = 'random'
max_depth = 20
min_samples_split = 10
min_samples_leaf = 4
```

```
--- Decision Tree ---
Accuracy: 0.9825
Classification Report:
              precision    recall  f1-score   support

           0       0.96      1.00      0.98        43
           1       1.00      0.97      0.99        71

    accuracy                           0.98       114
   macro avg       0.98      0.99      0.98       114
weighted avg       0.98      0.98      0.98       114

Confusion Matrix:
[[43  0]
 [ 2 69]]

--- Random Forest ---
Accuracy: 0.9737
Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.93      0.96        43
           1       0.96      1.00      0.98        71

    accuracy                           0.97       114
   macro avg       0.98      0.97      0.97       114
weighted avg       0.97      0.97      0.97       114

Confusion Matrix:
[[40  3]
 [ 0 71]]
```

```
--- AdaBoost ---
Accuracy: 0.9298
Classification Report:
              precision    recall  f1-score   support

           0       0.89      0.93      0.91        43
           1       0.96      0.93      0.94        71

    accuracy                           0.93       114
   macro avg       0.92      0.93      0.93       114
weighted avg       0.93      0.93      0.93       114

Confusion Matrix:
[[40  3]
 [ 5 66]]

The best model based on accuracy is: Decision Tree
```

Try 2 – AdaBoost

נעשה שינוי בכל הערכים של הפרמטרים

AdaBoost

```
# Define parameter values
n_estimators = 300
learning_rate = 0.2
estimator = DecisionTreeClassifier(max_depth=5)
```

Random Forest

```
# Define parameter values
n_estimators = 280
criterion = 'gini'
max_depth = 12
min_samples_split = 50
min_samples_leaf = 40
```

Decision tree

```
# Define parameter values
criterion = 'entropy'
splitter = 'random'
max_depth = 20
min_samples_split = 10
min_samples_leaf = 4
```

```
--- Decision Tree ---
Accuracy: 0.9825
Classification Report:
              precision    recall  f1-score   support

           0       0.96      1.00      0.98        43
           1       1.00      0.97      0.99        71

    accuracy                           0.98       114
   macro avg       0.98      0.99      0.98       114
weighted avg       0.98      0.98      0.98       114

Confusion Matrix:
[[43  0]
 [ 2 69]]

--- Random Forest ---
Accuracy: 0.9737
Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.93      0.96        43
           1       0.96      1.00      0.98        71

    accuracy                           0.97       114
   macro avg       0.98      0.97      0.97       114
weighted avg       0.97      0.97      0.97       114

Confusion Matrix:
[[40  3]
 [ 0 71]]
```

```
--- AdaBoost ---
Accuracy: 0.9649
Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.91      0.95        43
           1       0.95      1.00      0.97        71

    accuracy                           0.96       114
   macro avg       0.97      0.95      0.96       114
weighted avg       0.97      0.96      0.96       114

Confusion Matrix:
[[39  4]
 [ 0 71]]

The best model based on accuracy is: Decision Tree
```

Try 3 – AdaBoost

נעשה שינוי ב- n estimators – learning rate

AdaBoost

```
# Define parameter values
n_estimators = 380
learning_rate = 0.3
estimator = DecisionTreeClassifier(max_depth=5)
```

Random Forest

```
# Define parameter values
n_estimators = 280
criterion = 'gini'
max_depth = 12
min_samples_split = 50
min_samples_leaf = 40
```

Decision tree

```
# Define parameter values
criterion = 'entropy'
splitter = 'random'
max_depth = 20
min_samples_split = 10
min_samples_leaf = 4
```

```
--- Decision Tree ---
Accuracy: 0.9825
Classification Report:
              precision    recall  f1-score   support

           0       0.96      1.00      0.98        43
           1       1.00      0.97      0.99        71

    accuracy                           0.98       114
   macro avg       0.98      0.99      0.98       114
weighted avg       0.98      0.98      0.98       114

Confusion Matrix:
[[43  0]
 [ 2 69]]

--- Random Forest ---
Accuracy: 0.9737
Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.93      0.96        43
           1       0.96      1.00      0.98        71

    accuracy                           0.97       114
   macro avg       0.98      0.97      0.97       114
weighted avg       0.97      0.97      0.97       114

Confusion Matrix:
[[40  3]
 [ 0 71]]
```

```
--- AdaBoost ---
Accuracy: 0.9386
Classification Report:
              precision    recall  f1-score   support

           0       0.91      0.93      0.92        43
           1       0.96      0.94      0.95        71

    accuracy                           0.94       114
   macro avg       0.93      0.94      0.93       114
weighted avg       0.94      0.94      0.94       114

Confusion Matrix:
[[40  3]
 [ 4 67]]

The best model based on accuracy is: Decision Tree
```

Try 4 – AdaBoost

נעשה שינוי בכל הערכים של הפרמטרים

AdaBoost

```
# Define parameter values
n_estimators = 500
learning_rate = 0.7
estimator = DecisionTreeClassifier(max_depth=4)
```

Random Forest

```
# Define parameter values
n_estimators = 280
criterion = 'gini'
max_depth = 12
min_samples_split = 50
min_samples_leaf = 40
```

Decision tree

```
# Define parameter values
criterion = 'entropy'
splitter = 'random'
max_depth = 20
min_samples_split = 10
min_samples_leaf = 4
```

```
--- Decision Tree ---
Accuracy: 0.9825
Classification Report:
              precision    recall  f1-score   support

           0       0.96      1.00      0.98        43
           1       1.00      0.97      0.99        71

    accuracy                           0.98       114
   macro avg       0.98      0.99      0.98       114
weighted avg       0.98      0.98      0.98       114

Confusion Matrix:
[[43  0]
 [ 2 69]]

--- Random Forest ---
Accuracy: 0.9737
Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.93      0.96        43
           1       0.96      1.00      0.98        71

    accuracy                           0.97       114
   macro avg       0.98      0.97      0.97       114
weighted avg       0.97      0.97      0.97       114

Confusion Matrix:
[[40  3]
 [ 0 71]]
```

```
--- AdaBoost ---
Accuracy: 0.9737
Classification Report:
              precision    recall  f1-score   support

           0       0.98      0.95      0.96        43
           1       0.97      0.99      0.98        71

    accuracy                           0.97       114
   macro avg       0.97      0.97      0.97       114
weighted avg       0.97      0.97      0.97       114

Confusion Matrix:
[[41  2]
 [ 1 70]]

The best model based on accuracy is: Decision Tree
```

Try 5 – AdaBoost

נעשה שינוי בכל הערכים של הפרמטרים

AdaBoost

```
# Define parameter values
n_estimators = 130
learning_rate = 0.1
estimator = DecisionTreeClassifier(max_depth=3)
```

Random Forest

```
# Define parameter values
n_estimators = 280
criterion = 'gini'
max_depth = 12
min_samples_split = 50
min_samples_leaf = 40
```

Decision tree

```
# Define parameter values
criterion = 'entropy'
splitter = 'random'
max_depth = 20
min_samples_split = 10
min_samples_leaf = 4
```

```
--- Decision Tree ---
Accuracy: 0.9825
Classification Report:
              precision    recall  f1-score   support

           0       0.96      1.00      0.98        43
           1       1.00      0.97      0.99        71

    accuracy                           0.98       114
   macro avg       0.98      0.99      0.98       114
weighted avg       0.98      0.98      0.98       114

Confusion Matrix:
[[43  0]
 [ 2 69]]

--- Random Forest ---
Accuracy: 0.9737
Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.93      0.96        43
           1       0.96      1.00      0.98        71

    accuracy                           0.97       114
   macro avg       0.98      0.97      0.97       114
weighted avg       0.97      0.97      0.97       114

Confusion Matrix:
[[40  3]
 [ 0 71]]
```

```
--- AdaBoost ---
Accuracy: 0.9211
Classification Report:
              precision    recall  f1-score   support

           0       0.87      0.93      0.90        43
           1       0.96      0.92      0.94        71

    accuracy                           0.92       114
   macro avg       0.91      0.92      0.92       114
weighted avg       0.92      0.92      0.92       114

Confusion Matrix:
[[40  3]
 [ 6 65]]

The best model based on accuracy is: Decision Tree
```

הערכים הכי טובים של כל אחד מהמודלים

AdaBoost

```
# Define parameter values
n_estimators = 500
learning_rate = 0.7
estimator = DecisionTreeClassifier(max_depth=4)
```

Random Forest

```
# Define parameter values
n_estimators = 280
criterion = 'gini'
max_depth = 12
min_samples_split = 50
min_samples_leaf = 40
```

Decision tree

```
# Define parameter values
criterion = 'entropy'
splitter = 'random'
max_depth = 20
min_samples_split = 10
min_samples_leaf = 4
```