

## מבוא לבינה מלאכותית – תרגיל 4:

1. עצי החלטה – לעצמכם בלבד!! (כלומר – אין מה להגיש לי כי אין פה הוראות מסודרות)

המטרה בחלק זה היא שתכירו סיווג בעזרת עץ החלטה, XGBoost ו-Random Forest, ברמה הבסיסית של השימוש בהם וברמת ההבדלים שלהם בהתמודדות עם Overfitting.

לשימושכם:

עץ החלטה - [sklearn.tree.DecisionTreeClassifier](https://sklearn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html)

Random Forest - [sklearn.ensemble.RandomForestClassifier](https://sklearn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html)

XGBoost – חבילת [xgboost](https://xgboost.ai/)

קחו דאטאסט בסיסי מ-sklearn (למשל: iris), חלקו אותו לקבוצות training ו-test. כעת, אמנו כל אחד מהמודלים בעזרת קבוצת ה-training ובדקו את הדיוק שלכם על הפרדיקציות על ה-training ועל ה-test. האם אתם overfitting?  
שנו פרמטרים שאמורים להקטין את ה-overfitting ונסו שוב, האם הצלחתם?  
מי המודל הכי מוצלח?

2. אשכול והורדת ממדים:

א. כיווץ תמונה באמצעות הורדת ממדים -

בחלק זה תקבלו תמונה המיוצגת באמצעות ערכי פיקסלים RGB (מצורפת ב-math wiki בתור קובץ image.txt של שלשות פיקסלים. הממדים המקוריים של התמונה הם  $161 \times 212$ ). עליכם להפעיל PCA ו-ICA לאוסף הפיקסלים, באופן הבא:  
ראשית, צרו מטריצה לכל פיקסל (אחת של  $161 \times 212$  ל-r, אחת ל-g ואחת ל-b), ואח"כ צרפו אותן למטריצה אחת של  $161 \times 636$ .  
הסתכלו על התמונה הנוצרת לאחר הורדת הממדים כתלות בכמות הרכיבים שלקחתם (עשו זאת עבור 3, 5, 10, 15, 20, 30 רכיבים. למען הסר ספק, הממדים של התמונה לא משתנים במהלך התהליך, רק מספר הרכיבים הראשיים בהם תשתמשו).  
ציירו את השונות המוסברת כתלות במספר הרכיבים. כמה מתוך השונות הספיקה לכם כדי לזהות את התמונה כמו שצריך?  
לשימושכם, להעלאת התמונה והצגתה:

```
import numpy as np
import matplotlib.pyplot as plt
img = np.loadtxt('image.txt').reshape((161, 212, 3)) / 255.
plt.imshow(img, interpolation='nearest')
plt.show()
```

**בהצלחה!**