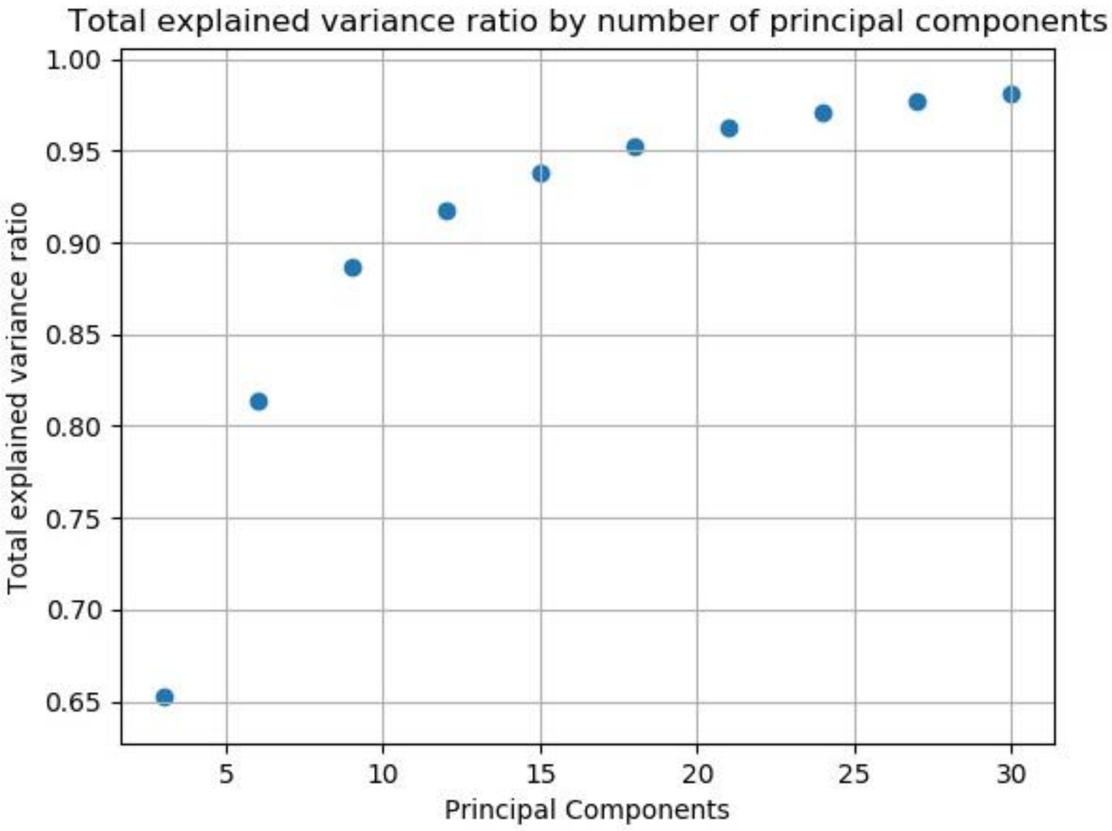
מבוא לבינה מלאכותית – פתרון תרגיל 4:

1. כמו שכבר דיברנו בכיתה, לעצי החלטה יש נטייה ל-overfitting. לכן, מצורף הקישור הבא להסבר על שליטה בפרמטרים שיעזרו (לנסות) למנוע זאת, ועוד דברים נוספים:
https://xgboost.readthedocs.io/en/latest/tutorials/param_tuning.html

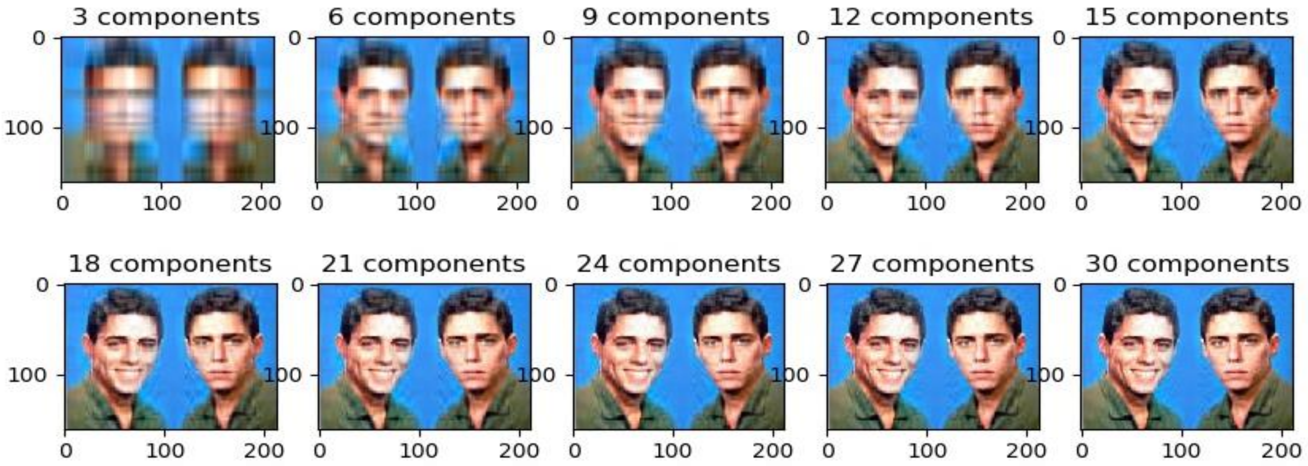
2. א.

לצורך סעיף זה השתמשתי בפונקציות PCA ו-FastICA מתוך sklearn.decomposition.
השלבים שנעשו לכל מספר רכיבים:
ראשית, מבצעים את התהליך המתאים למציאת הרכיבים העיקריים (ובמקרה של PCA, גם השונות המוסברת. נעשה באמצעות PCA.fit(X) ודומה לגבי FastICA).
לאחר מכן, מבצעים את הורדת הממדים (מעשית, זו הטלה של הווקטורים הקיימים על הרכיבים העיקריים שמצאנו לקבלת ייצוג בעזרת מספר פרמטרים קטן יותר לכל "נקודה" בדאטא. נעשה באמצעות PCA.transform(X) ודומה לגבי FastICA) ומשחזרים את התמונה המקורית באמצעות הווקטורים והרכיבים (סכום של כל פרמטר חדש כפול הרכיב העיקרי המתאים לו. נעשה באמצעות PCA.inverse_transform(transformed_X) ודומה לגבי FastICA). ומקבלים:

PCA:
שונות מוסברת:



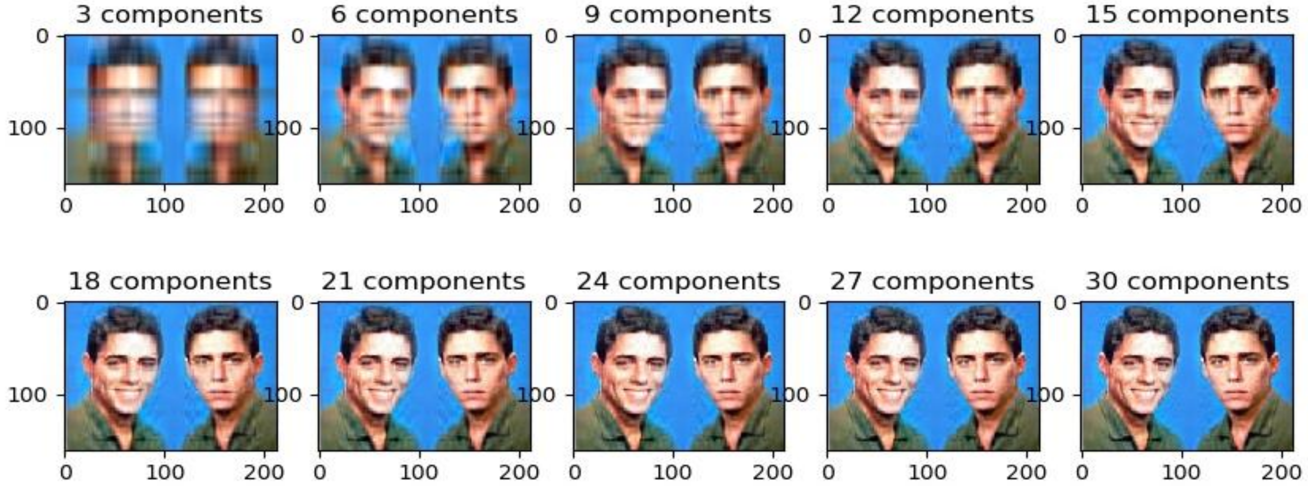
התמונות החדשות:



לי התמונה נעשית ברורה כבר כשמשתמשים ב-15 רכיבים.

ICA:

התמונות החדשות:

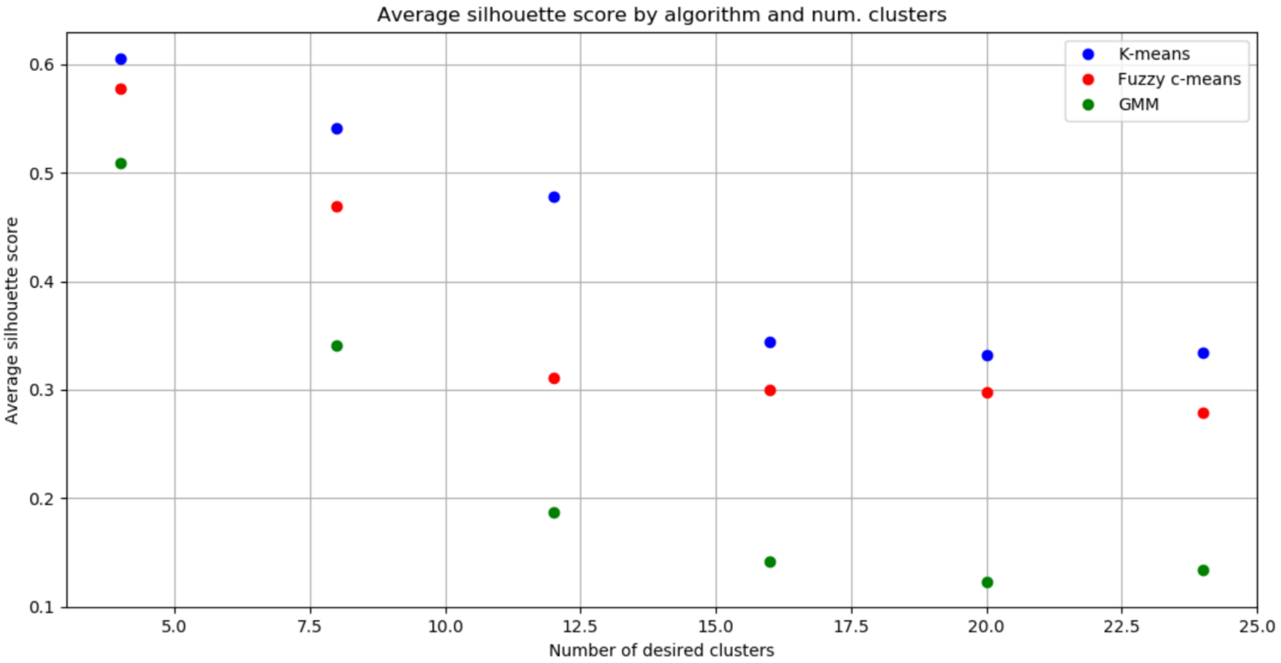


די דומה ל-PCA.

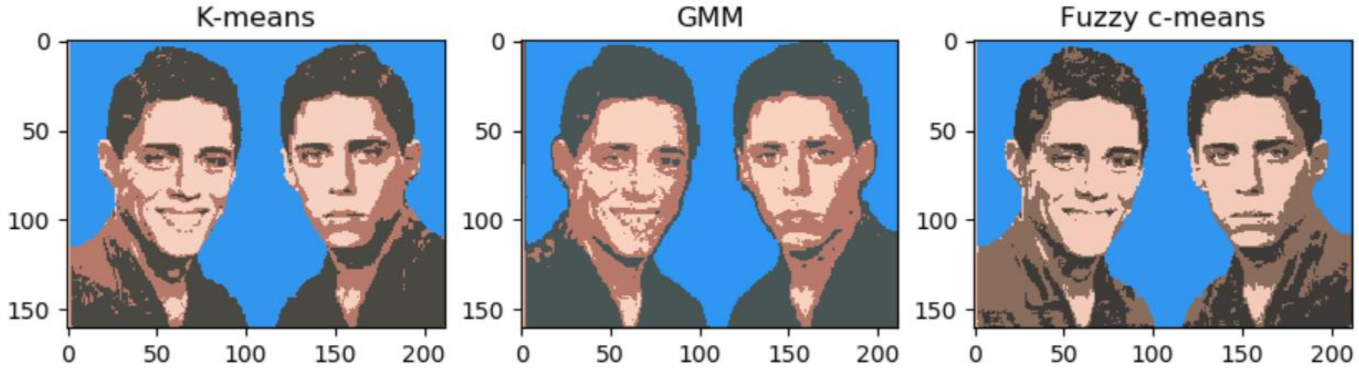
ב.
לצורך יישום האלגוריתמים השתמשתי ב- sklearn.cluster.KMeans, sklearn.mixture.GaussianMixture, skfuzzy.cluster (האחרון הוא מהחבילה scikit-fuzzy שהורדתי).

רק אציין שיש ב-GMM וב-fuzzy c-means פרמטרים חופשיים שלא התעסקתי בהם (כמו סוג מטריצת ה-Covariance ב-GMM או הקבוע m ב-fuzzy c-means).

התוצאה:



מכאן, עדיף לנו לקחת 4 אשכולות לאשכול הפיקסלים שלנו במרחב.
בלקיחת כמות מרכזים כזו, נקבל:



מצד אחד זה נראה נחמד ואולי הייתי עושה מזה תמונה לחדר, אבל רואים שיש פער לא קטן בין התמונה המקורית לבין התמונות האלה. נסו לבדוק מה יקרה אם תיקחו יותר מרכזים (כלומר יותר אפשרויות לצבעים במרחב), אני מנחש שעם יותר מ-4 מרכזים תקבלו תמונות קצת יותר נאמנות למקור, ולכן במקרה הזה silhouette score לא ממש מתאים למטרה של שחזור התמונה.