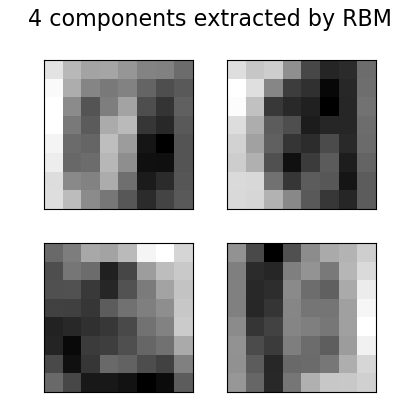
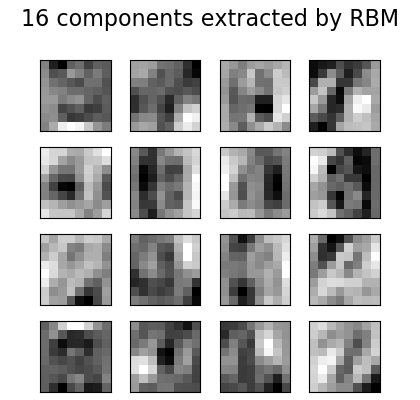
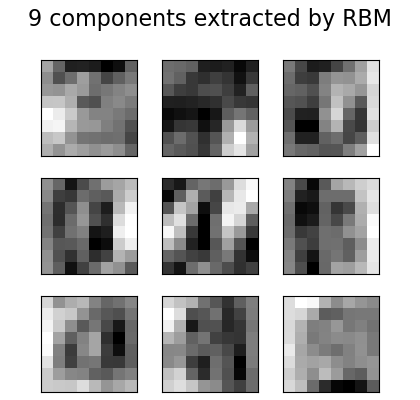
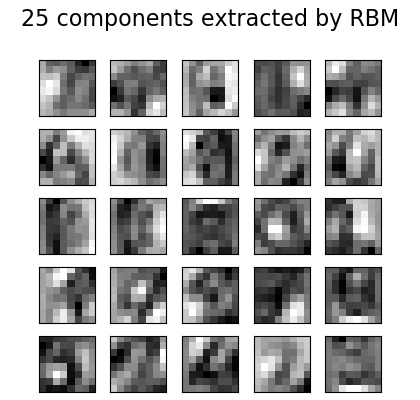
תרגיל 3

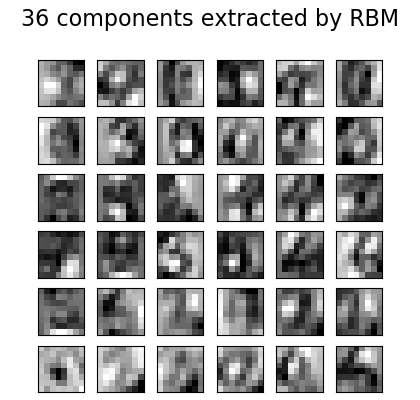
מגישים:

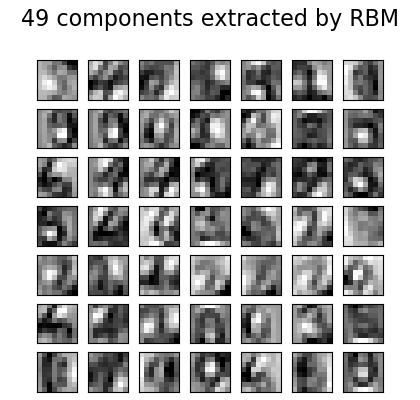
1.א. המספרים:

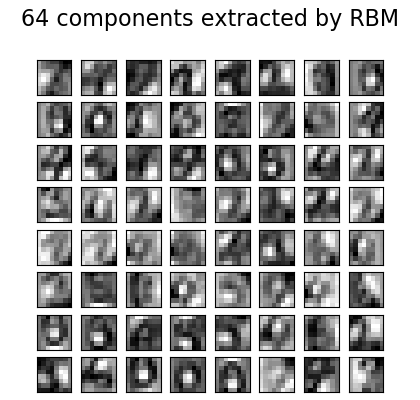


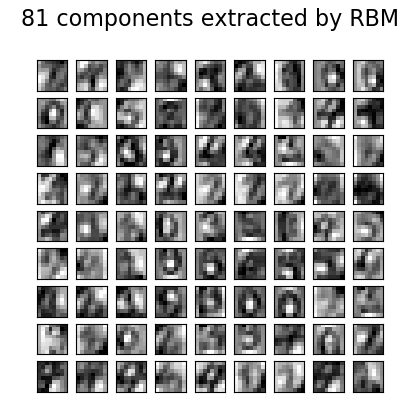


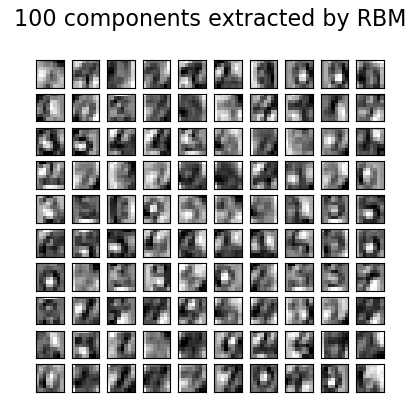


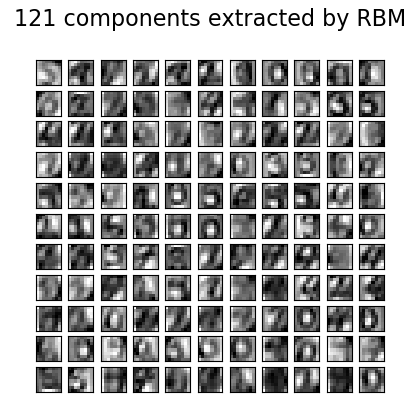


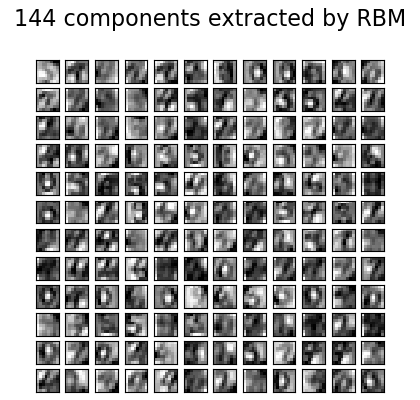


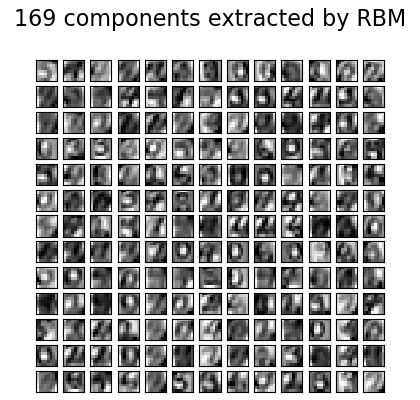


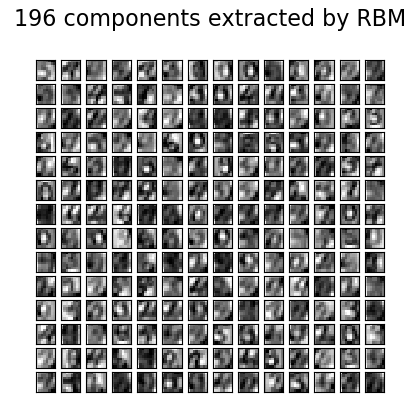






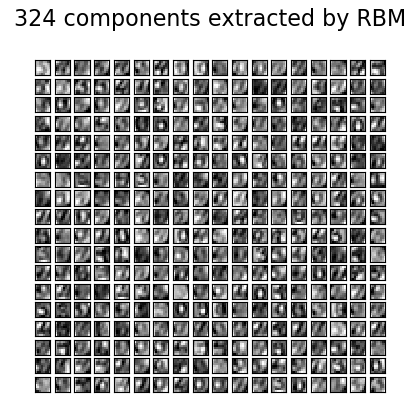


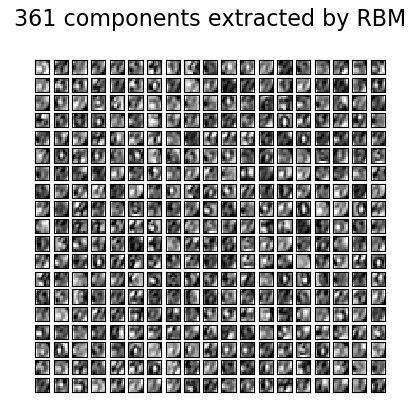


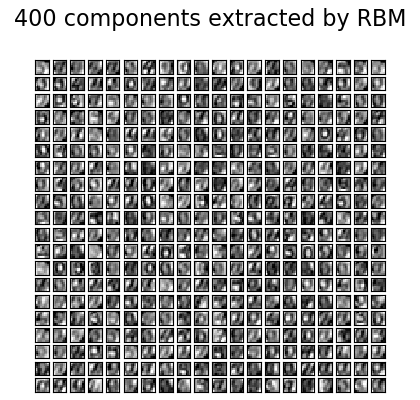


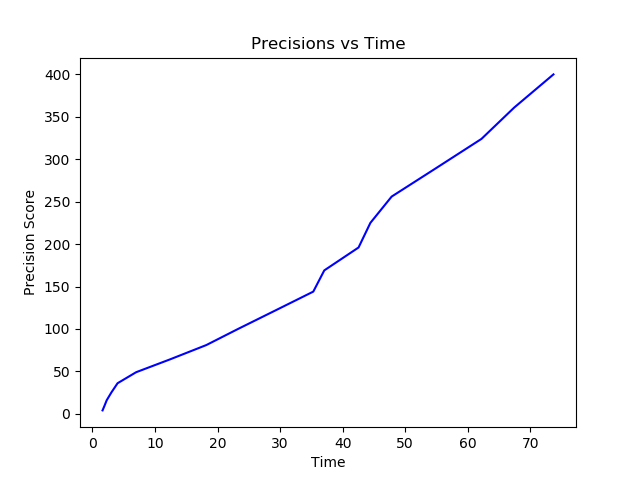


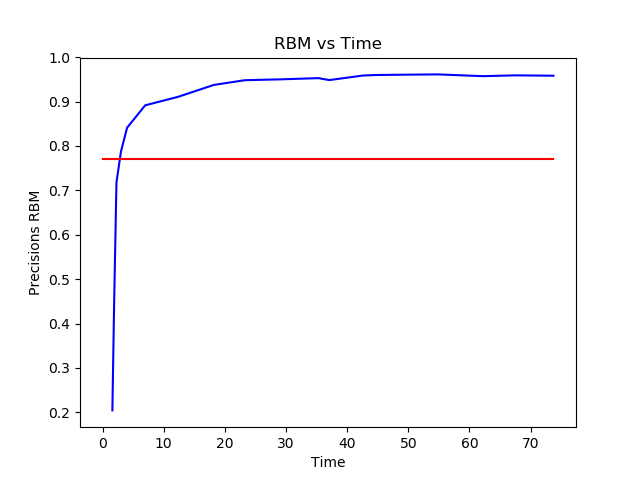








1.ג.



1.ד.1:

X\_train: (7188, 64)

X\_test: (1797, 64)

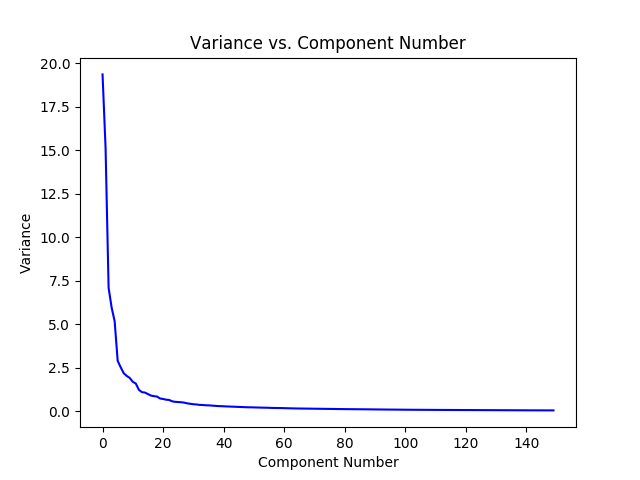
rbm.transform(X\_train): (7188, i^2) - i is the iteration value

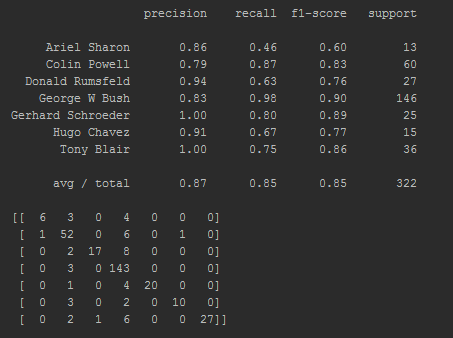
rbm.intercept\_hidden\_: (i^2, 1) - i is the iteration value

1.ד.2:

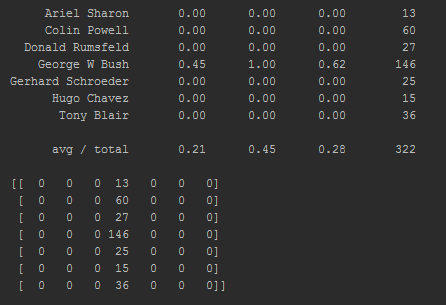
המימדים שלנו מסתדרים שכן אנו כופלים במכפלה וקטורית את מטריצת הנתונים ממימד (7188 X 64) בוקטור עמודה של  
(64 X i^2)T , כלומר מקבלים את המכפלה הוקטורית של ולכן נקבל תוצאה שהיא בגודל של (7188 X i^2) והוספנו לכל שורה את הוקטור RBM.hiddenIntercept מה שנתן לנו את הגודל שצפינו לקבל מהסעיף הקודם.

שאלה 2:

2.א.

2.ב.

2.ה.



2.ו. ניתן לראות כי ללא הפכת המימדים שביצע ה PCA הגענו לאחוזי דיוק נמוכים במיוחד תוך שימוש ב GridSearch על ה dataset ללא PCA.

בעת ביצוע ה gridsearch ללא PCA אנו מתקשים למצוא הפרדה לינארית בין הנתונים, שיתן לנו מסווג משמעותי עבור הנתונים – במיוחד כאשר המימד שלהם גדול מאוד ביחס לכמות הנתונים שאינה גדולה דיה.

לעומת זאת, PCA מסוגל למצוא סיווג משמעותי באמצעות מציאת מערכות צירים להטיל עליהן ולצמצם את המימדים ובכך לאפשר מסווג שמגיע לדיוק גבוה (הרבה) יותר עם אותה כמות של נתונים.

בשאלה הקודמת, הפחתת המימדים הועילה לנו לקיצור זמן הריצה של אלגוריתם הלמידה, אך דווקא הוספת המימדים הועילה לנו (עד גבול מסויים) לשיפור הדיוק של אלגוריתם הלמידה – לפיכך לא הועילה לנו הפחתת המימדים בשאלה הראשונה.