בחירת קואליציה על ידי generative models:

ראשית ביצענו k cross validation על 2 מודלים גנרטיביים מסוג GaussianNB ו LinearDiscriminantAnalysis ובדקנו מי מקבל ציון גבוה ביותר בחיזוי הסיווג (לפי k-cross), קיבלנו שGaussianNB מקבל דיוק גבוה יותר.

לאחר מכן בנינו מטריצת הסתברויות לפי המודל הטוב מבין (GaussianNB) , מטריצה זו נבנתה בצורה שבה עבור כל מפלגה בדקנו את המצביעים אשר הצביעו למפלגה זו ומה ההסתברות לפי המודל שהם יצביעו למפלגה אחרת , זאת אומרת עבור כל מצביעי מפלגה א מה ההסתברות שהם הצביעו לכל אחד משאר המפלגות ובהתאם ממלאים את ערכים אלו במטריצה.

לאחר מכן מקבלים מטריצה אשר בכל איבר i,j מקבלים מה ההסתברות שבוחרי i יצביעו למפלגה j, ועל ידי קשר זה אנו בונים את האפשריות לקואליציה.

אנו עוברים על כל האפשריות לקואליציה שבה מתחילה ממפלגה א בודדת ואז מוסיפים לה את המפלגה ב שהבוחרים שלה הם עם הסתברות הכי גדולה לבחור במפלגה א. וכך עד שמגיעים לפחות ל 51% כדי לבנות קואליציה ומחשבים ציון כפי שיוסבר בהמשך.

לאחר שמגיעים ל 51% ממשיכים להוסיף מפלגות באותה צורה עד שלא נשאר מפלגות להוסיף ולכל אפשרות כזו מחשבים ציון . לבסוף הקואליציה שתקבל ציון גבוה ביותר תנצח.

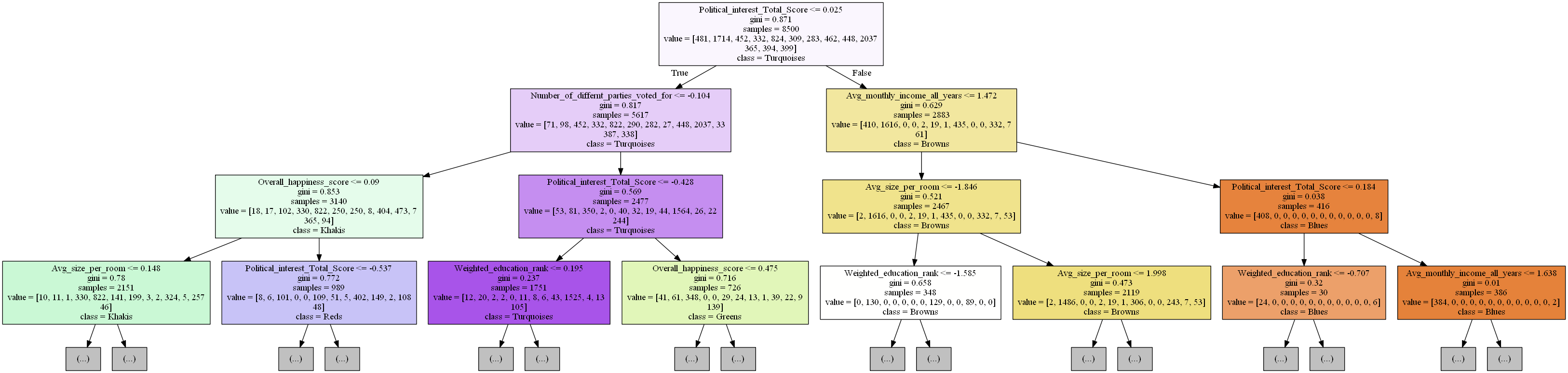
הציון מחושב על ידי חישוב הvariance של הקואליציה ובנוסף חישוב של מרחק אוקלידי בין הקואליציה לאופוזיציה , המטרה להגיע לvariance קטן ומרחק גדול ולבסוף בחרנו ידנית את המנצח.

הקואליציה שקיבלה את הציון הגבוה ביותר לפי מודל זה היא:

**['Greys', 'Khakis', 'Oranges', 'Pinks', 'Reds', 'Turquoises', 'Whites', 'Yellows]**

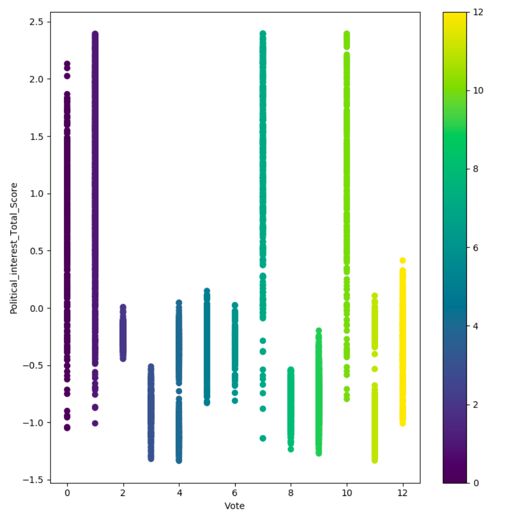
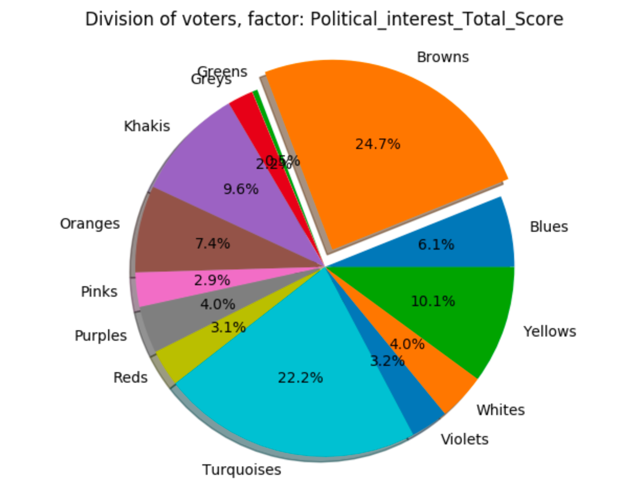
חלק זה נמצא בקובץ fourth\_prediction.

ראשית הסתכלנו על השכבות הראשונות של עץ החלטה וראינו מי מהתכונות בעלי משקל גדול על תוצאות המודל והסתכלנו על התכונות איך הן משפיעות על ההצבעה ואיך הן מפולגות.לכל תכונה בנפרד שינינו את קבוצת ה test עבור אותה תכונה כדי להשפיע על התוצאות של המנצח על ידי שהמסווג יטעה בגלל תכונה זו בסיווג. שינינו כל תכונה ולאחר שינוי קבוצת הtest בדקנו את המודל (שמאומן על train תקין) ואת המנצח.

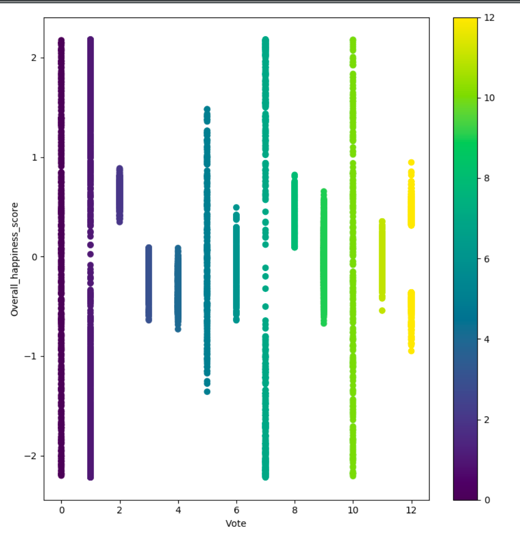
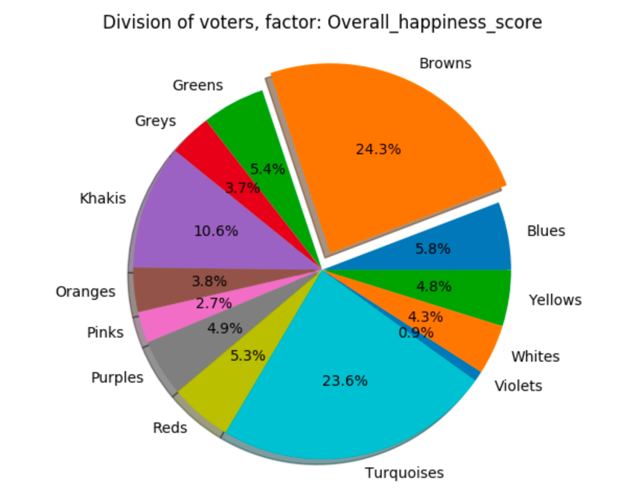


בנוסף ראינו מתרגיל קודם שהמנצח הוא (9)Turquoises ומקום שני הוא Brown (1) וקיבלנו את התכונות הבאות כקריטיות עבור תוצאות הסיווג

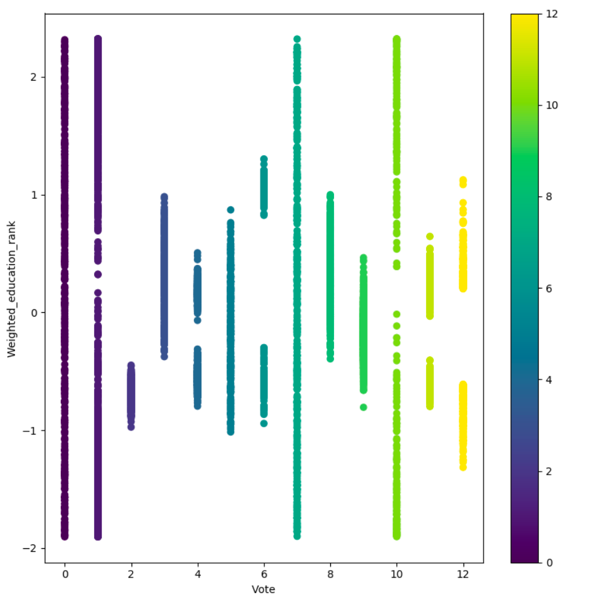
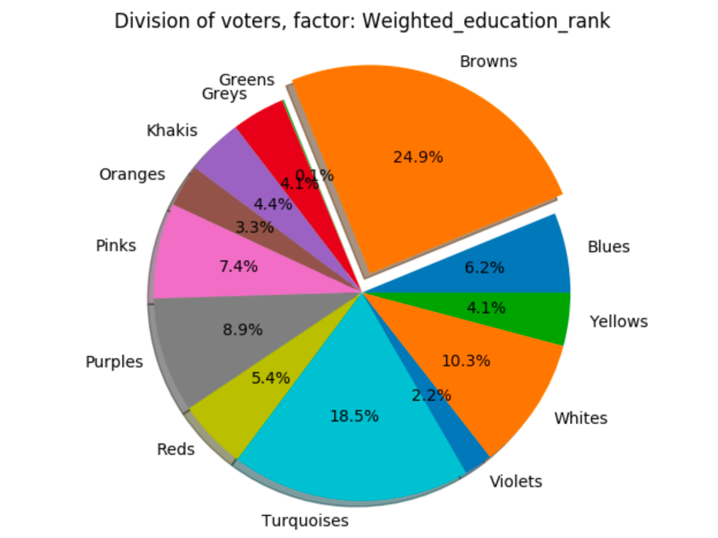
Political\_interest\_Total\_Score **:** ניתן לראות כי המצביעים של 1 ו-9 מפולגים בצורה כזאות שעבור 1 יש בוחרים בטווח גדול יותר ובאזור הגבוהה של הערכים והטווח עבור 9 קטן יותר ולמטה.

****פילוג של train data על תכונה זו ביחס להצבעות: brown מנצח לאחר שינוי תכונה זו בtest:

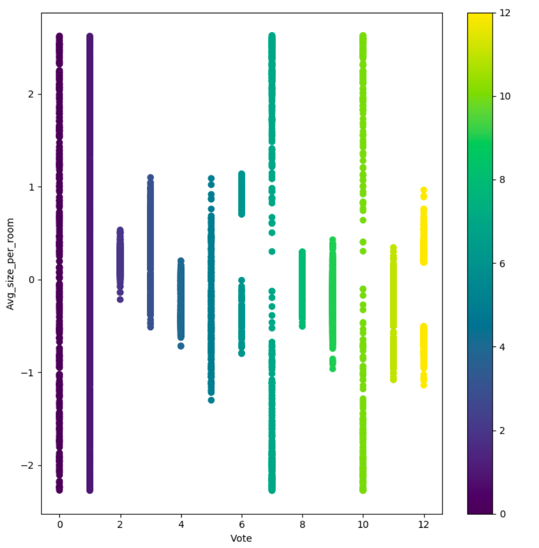
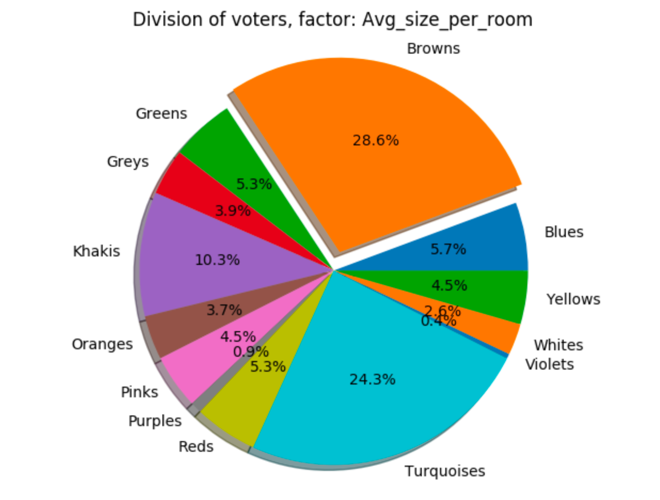
Overall\_happiness\_score **:** ניתן לראות כי המצביעים של 1 ו-9 מפולגים בצורה כזאות שעבור 1 יש בוחרים בטווח גדול יותר בכל הטווח מלבד האמצע ב והטווח עבור 9 קטן יותר ובאמצע.

****פילוג של train data על תכונה זו ביחס להצבעות: brown מנצח לאחר שינוי תכונה זו בtest:

Weighted\_education\_ran**:** ניתן לראות כי המצביעים של 1 ו-9 מפולגים בצורה כזאות שעבור 1 יש בוחרים בטווח גדול יותר בכל הטווח מלבד האמצע והטווח עבור 9 קטן יותר ובאמצע.

****פילוג של train data על תכונה זו ביחס להצבעות: brown מנצח לאחר שינוי תכונה זו בtest:

Size per room **:** ניתן לראות כי המצביעים של 1 ו-9 מפולגים בצורה כזאות שעבור 1 יש בוחרים בטווח גדול יותר בכל הטווח והטווח עבור 9 קטן יותר ובאמצע.

****פילוג של train data על תכונה זו ביחס להצבעות: brown מנצח לאחר שינוי תכונה זו בtest: