**תרגיל בית 4 מבוא למערכות לומדות**

**דוח עבודה**

**מגישים:**

**אביב כספי – 311136691**

**יקיר יהודה - 205710528**

שלבי עבודה:  
1. **Pre-processing:**דבר ראשון שעשינו היה ביצוע עיבוד ראשוני לדאטה שלנו.  
מימוש חלק זה נמצא בקובץ preprocessing.py.

1. המרה ל one-hot של הפיצ'רים הנומינליים ללא חשיבות לסדר :  
   מתוך הפיצ'רים שנבחרו היה עלינו לשנות רק את 'Most\_Important\_Issue' לone-hot.
2. המרת כל הפיצ'רים הקטגוריאליים למספריים:  
   על מנת שנוכל לעבוד על הדאטה, עלינו לשנות את כל הפיצ'רים שיהיו מספריים.
3. תיקון ערכים שליליים בפיצ'רים:  
   בדומה לתרגיל קודם, גם כעת ישנו פיצ'ר בעל ערכים שליליים למרות שלא ייתכן כי הערכים של הפיצ'ר יהיו שליליים.  
   בצורה דומה לתרגיל הקודם, התמודדנו עם בעיה זו על ידי ביצוע abs על הפיצ'ר הנ"ל. (כמו בתרגיל קודם, זיהינו כי התפלגות הדאטה השלילי מתאימה בדיוק להתפלגות הפיצ'ר בערכים החיובים, לכן החלטנו כי ייתכן כי בזמן הדגימה שונה הסימן בטעות, לכן החלטנו רק לבצע ערך מוחלט על הדאטה הנ"ל)
4. לאחר מכן ביצענו פיצול לדאטה שלנו:

בדומה לתרגיל הקודם בחרנו את הפיצול הבא – train(70%), val(15%), test(15%)

1. מחיקת outliers:

בשלב זה מצאנו את הדוגמאות בהן ערך הפיצ'ר במרחק גדול מ4.5 סטיות תקן מהממוצע של הפיצ'ר, והצבנו NAN במקומות אלה.  
פעולה זו ביצענו רק על סט האימון, והשתמשנו בלייבלים של הדאטה על מנת למצוא outliers יחסית ללייבל.

1. Imputation :   
   השלמת ערכים חסרים ביצענו בנפרד לסט האימון ולסט הולידציה והבדיקה -  
   עבור סט האימון בחרנו בשיטה הנקראת Bootstraping שלמדנו בכיתה.  
   בשיטה זו, עבור כל ערך חסר, השלמנו על ידי כך שדגמנו מתוך הדאטה שלנו עם חשיבות ללייבל, ערך חדש.

עבור סט הולידציה והבדיה בחרנו לטפל בצורה שונה.  
תחילה לקחנו את סט האימון, ועבור כל פיצ'ר חישבנו את ממוצע הפיצ'ר עבור ערכים נומריים ואת הערך הנפוץ ביותר עבור ערכים נומינליים (ללא חשיבות ללייבל).  
לאחר מכן, השלמנו את סט הולידציה והבדיקה בעזרת הערכים שחושבו מסט האימון.

1. Scaling:

עבור פיצ'רים יוניפורמים ביצענו scaling לטווח [1 1-]  
עבור פיצ'רים נורמלים ביצענו נורמליזציה עם ממוצע 0 וסטיית תקן 1.

2. **מציאת קואליציות אפשריות על ידי מודל clustering:**

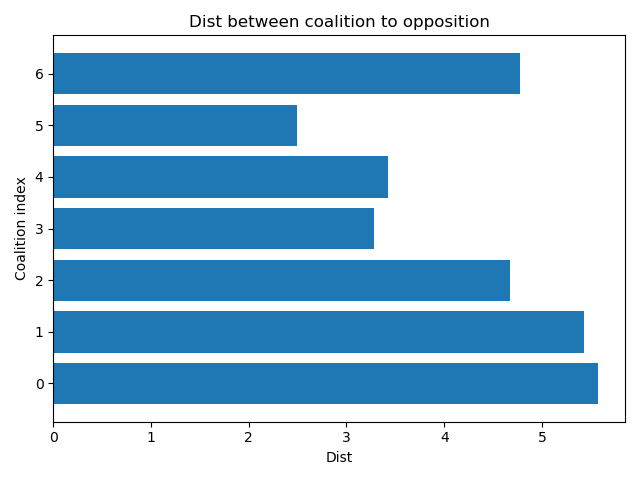
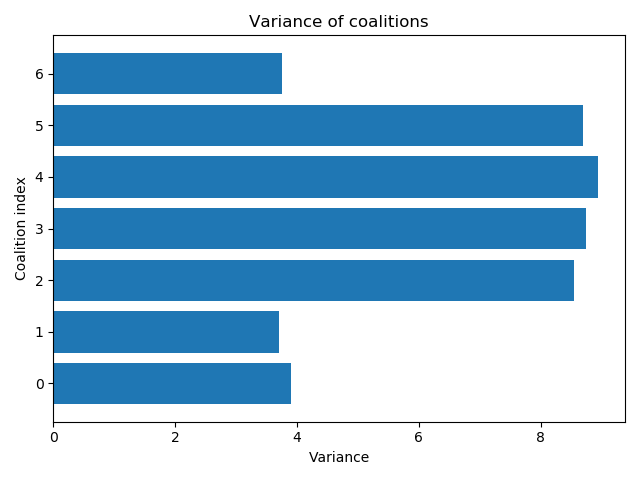
על מנת לבצע חלק זה, בחרנו בשני מודלים שלclustering אשר בעזרתם ננסה לבנות קואליציה טובה.  
המודלים שבחרנו הינם: KMeans , GaussianMixture.  
**שלב 1:** מציאת מספר הקלאסטרים האופטימלי עבור כל מודל.  
שלב זה ביצענו על ידי ביצוע CV על מספר הקלאסטרים שכל מודל מחפש, ובנינו פונקציית ניקוד אשר מנקדת כל קלאסטר שהוצע על ידי המודל ולפי כך בחרנו את הערך הטוב ביותר.  
פונקציית הניקוד – בהינתן מודל מאומן, וסט ולידציה, תחילה ביצענו חיזוי עבור כל נקודה בדאטה, לאיזה מהקלאסטרים היא שייכת.  
לאחר מכן עבור כל קלאסטר, עברנו על כל המפלגות שחזינו כי ישנם מצביעים הנמצאים באותו קלאסטר. עבור כל מפלגה בדקנו כמה אחוז מכלל המצביעים למפלגה זו נמצא בתוך הקלאסטר, אם אחוז זה גדול מ60% (כלומר רוב המצביעים למפלגה נמצאים בקלאסטר זה) הוספנו אחוז זה לציון הקלאסטר. (אם אחוז המצביעים למפלגה מסוימת היה קטן מ60% התעלמנו ממפלגה זו כי אינה שייכת לקלאסטר).  
פונקציית ניקוד זו, בעצם מקשרת בין הקלאסטרים למפלגות הבעיה שלנו, כלומר ככל שהניקוד גדול יותר, כך יותר מפלגות שייכות ברובן לקלאסטר מסויים. לדוגמא אם בחנו אלגוריתם מסוים עם 3 קלאסטרים וקיבלנו כי מפלגה A נמצא בצורה שווה בכל קלאסטר (30% בכל קלאסטר), מודל זה לא יקבל ניקוד עבור המפלגה הנ"ל. אך אם מפלגה B שייכת בעיקר לקלאסטר מסוים (מעל 60%), המודל יקבל ציון לפי הגודל היחסי של המפלגה השייך לקלאסטר.

בסוף שלב זה קיבלנו כי שני המודלים מקבלים ניקוד אופטימלי עבור 2 קלאסטרים.

**שלב 2:** מציאת קואליציות אפשריות על ידי המודלים שמצאנו.  
בשלב זה עבור כל מודל שמצאנו ביצענו:  
עבור כל קלאסטר שהמודל מצא –   
מצאנו את כל המפלגות אשר אחוז שייכות המצביעים לכל מפלגה בקלאסטר גדול מ 85% (בחרנו ערך זה מפני שהוא גבוה מספיק כדי להגיד שרוב גדול של מצביעים למפלגה שייכים לקלאסטר זה)  
לאחר מכן ספרנו את גודל סך המצביעים לכל מפלגות אלה, אם גודל זה גדול שווה ל51%, הוספנו את המפלגות הנ"ל כקואליציה לרשימת הקואליציות האפשריות.

בסוף שלב זה קיבלנו רשימה של קואליציות אפשריות, כך שכל קואליציה מכילה מספר מפלגות שסך המצביעים הכולל למפלגות אלה גדול מ51%.

**שלב 3:** מציאת קואליציה הומוגנית ביותר ושונה ביותר מהאופוזיציה.  
בשלב זה עבור כל קואליציה שמצאנו בשלב הקודם, חישבנו את שונות הפיצ'רים של מצביעיה (המצביעים של הקואליציה – מצביעים לכל אחת מהמפלגות בקואליציה)  
ובנוסף חישבנו את המרחק של ממוצע הפיצ'רים של הקואליציה לממוצע הפיצ'רים של האופוזיציה.  
(כלומר כמה שונה האופוזיציה מהקואליציה)  
ביצענו מיון על כל אחת מהרשימות הנ"ל (שונות ומרחק), והסתכלנו על התוצאות, כך שהרשימה של השונות ממוינת מהקטן לגדול (ערך קטן יותר – קואליציה הומוגנית יותר) והמרחק ממוין מהגדול לקטן (ערך גדול יותר – קואליציה שונה מהאופוזיציה יותר).  
לאחר הסתכלות על התוצאות בחרנו בקואליציה אשר מיקומה היה בראשונים בכל רשימה, כך שערך השונות שלה קטן והמרחק מהאופוזיציה גדול.

הגרפים שקיבלנו עבור השונות והמרחק של הקואליציות:  


מתוך רשימת הקואליציות הבאות:

[(2, 3, 4, 5, 6, 8, 9, 11, 12), (2, 3, 4, 5, 6, 8, 9, 11), (0, 1, 2, 4, 6, 7, 9, 10, 11, 12), (0, 1, 2, 4, 6, 7, 9, 10, 11), (0, 1, 2, 4, 6, 7, 9, 10), (0, 1, 2, 4, 6, 7, 9), (2, 3, 4, 5, 6, 8, 9)]

כאשר:

|  |
| --- |
| **'Violets'** - 10 |
| **'Browns'** - 1 |
| **'Greens'** - 2 |
| **'Whites'** - 11 |
| **'Reds'** - 8 |
| **'Turquoises'** - 9 |
| **'Purples'** - 7 |
| **'Greys'** - 3 |
| **'Blues'** - 0 |
| **'Khakis'** - 4 |
| **'Oranges'** - 5 |
| **'Pinks'** - 6 |
| **'Yellows'** - 12 |

הקואליציה שנבחרה היא קואליציה 0

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

אשר מכילה 9 מפלגות מתוך 13.  
בעלת מרחק מקסימלי ושונות כמעת מינימלית.

הערות:  
בחיפוש הקואליציות האפשריות, ביצענו אימון של המודל על סט האימון, ולאחר מכן מצאנו את הקואליציות האפשריות מתוך סט הולידציה וגם מתוך סט האימון (ביצענו חיתוך בין התוצאות)  
זאת על מנת לזהות כי אכן הקואליציות שקיבלנו על ידי אימון על סט האימון, מתאימות גם לסט הולידציה. (סוג של בדיקת שפיות, על מנת לוודא שהמסווג שלנו עובד בצורה טוב מספיק).  
בנוסף, לאחר בחירת הקואליציה האופטימלית, השתמשנו במודל חיזוי מהתרגיל הקודם (RandomForest) וביצענו חיזוי ללייבים של סט הבדיקה.  
לאחר מכן ביצענו את אותו תהליך של בחירת קואליציה על סט הבדיקה, ווידאנו כי תוצאת הקואליציה מתאימה לקואליציה שמצאנו בחלקים הקודמים. (בדיקה נוספת כי המודל שלנו יודע להכליל גם על ידי חיזוי הלייבלים).

**d.** חלק זה נמצא בקובץ fourth\_prediction.

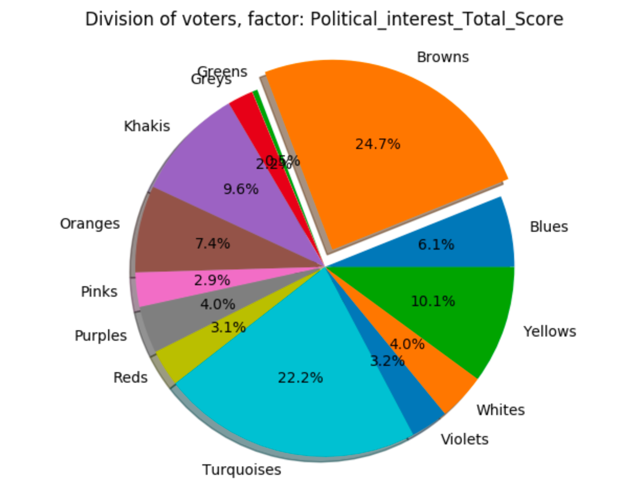
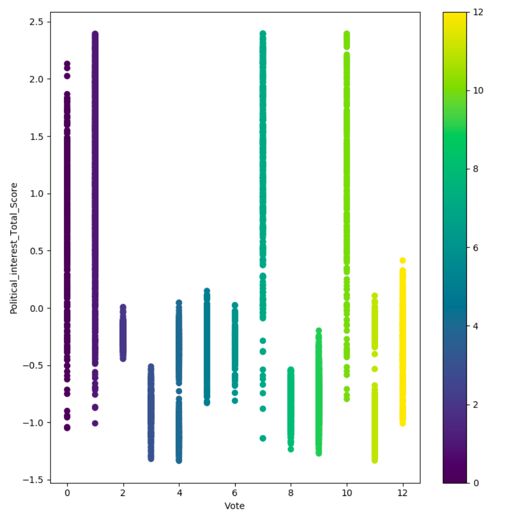
בחלק זה התייחסנו לחלוקת הדאטה ל train ו test .

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 מסעיפים קודמים שהמנצח הוא (9)Turquoises ומקום שני הוא Brown (1) וקיבלנו את התכונות הבאות כקריטיות עבור תוצאות הסיווג

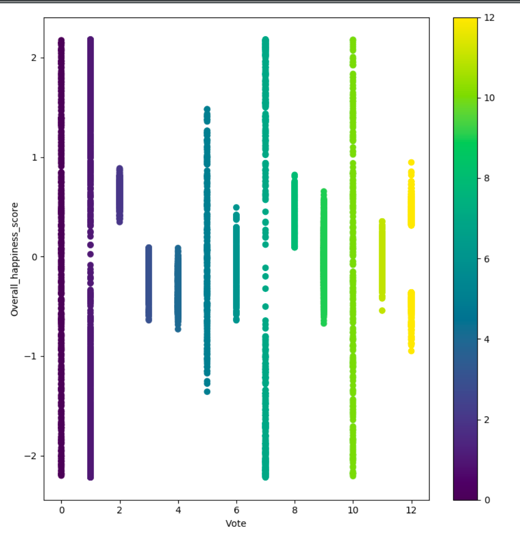
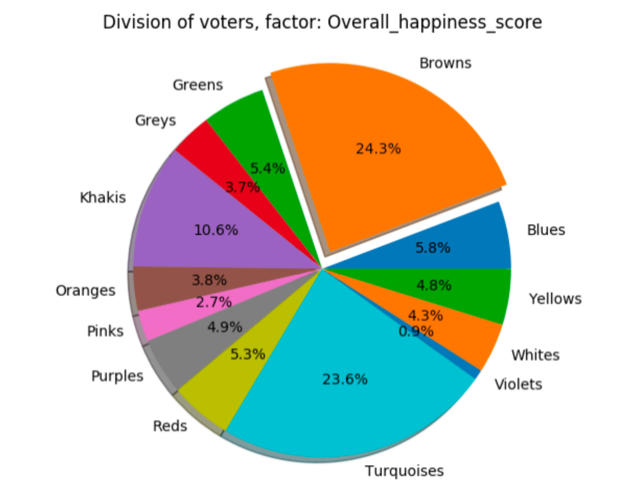
Political\_interest\_Total\_Score **:** ניתן לראות כי המצביעים של 1 ו-9 מפולגים בצורה כזאות שעבור 1 יש בוחרים בטווח גדול יותר ובאזור הגבוהה של הערכים והטווח עבור 9 קטן יותר ולמטה.

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

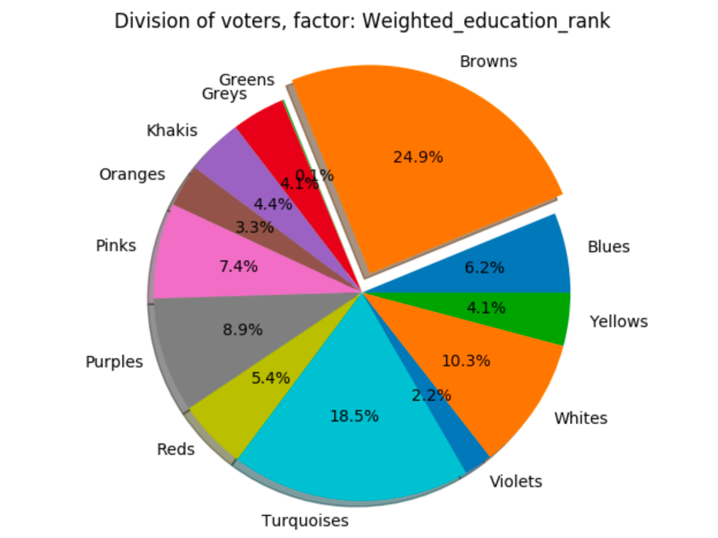
****

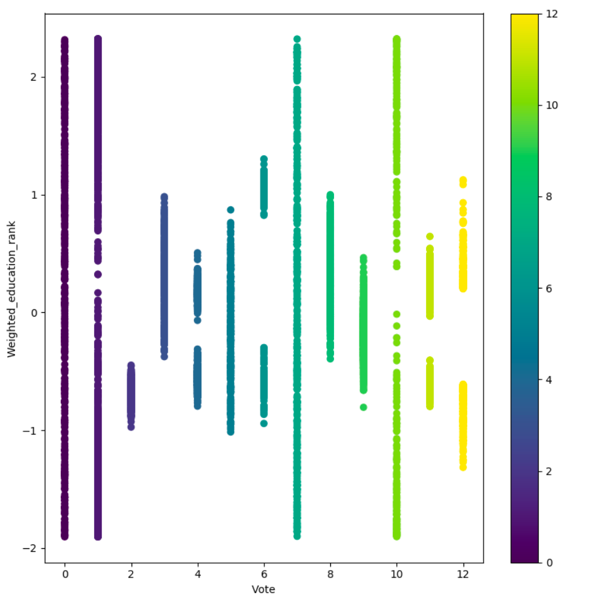
Overall\_happiness\_score **:** ניתן לראות כי המצביעים של 1 ו-9 מפולגים בצורה כזאות שעבור 1 יש בוחרים בטווח גדול יותר בכל הטווח מלבד האמצע ב והטווח עבור 9 קטן יותר ובאמצע.

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

****

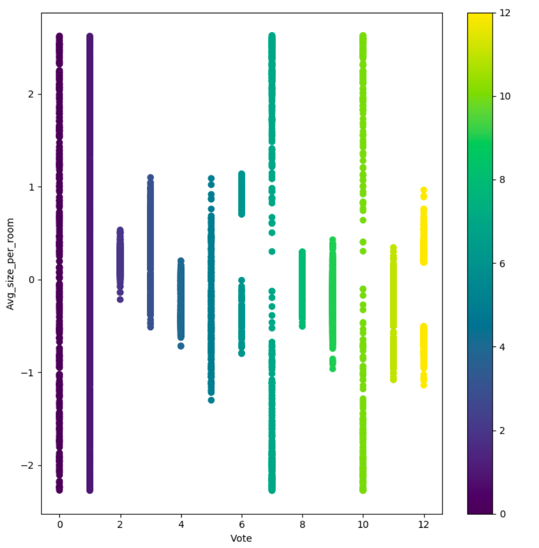
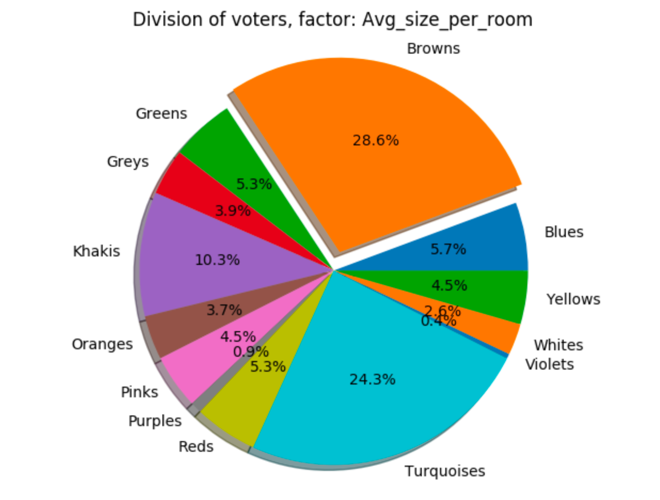
Weighted\_education\_ran**:** ניתן לראות כי המצביעים של 1 ו-9 מפולגים בצורה כזאות שעבור 1 יש בוחרים בטווח גדול יותר בכל הטווח מלבד האמצע והטווח עבור 9 קטן יותר ובאמצע.

****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:

****