תרגיל בית 4 מבוא למערכות לומדות

דוח עבודה

מגישים:

אביב כספי – 311136691

יקיר יהודה - 205710528

<u>שלבי עבודה:</u>

1. **Pre-processing:**

דבר ראשון שעשינו היה ביצוע עיבוד ראשוני לדאטה שלנו.
מימוש חלק זה נמצא בקובץ preprocessing.py.

א. המרה לone-hot של הפיצ'רים הנומינליים ללא חשיבות לסדר :
מתוך הפיצ'רים שנבחרו היה עלינו לשנות רק את 'Most_Important_Issue' ל-one-hot.

ב. המרת כל הפיצ'רים הקטגוריאליים למספריים:
על מנת שנוכל לעבוד על הדאטה, עלינו לשנות את כל הפיצ'רים שיהיו מספריים.

ג. 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)

ד. לאחר מכן ביצענו פיצול לדאטה שלנו:
בדומה לתרגיל הקודם בחרנו את הפיצול הבא – train(70%), val(15%), test(15%)

ה. 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 לליבל.

ו. 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.

ז. 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 של clustering אשר בעזרתם ננסה לבנות קואליציה טובה.
המודלים שבחרנו הינם: GaussianMixture , KMeans.
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.

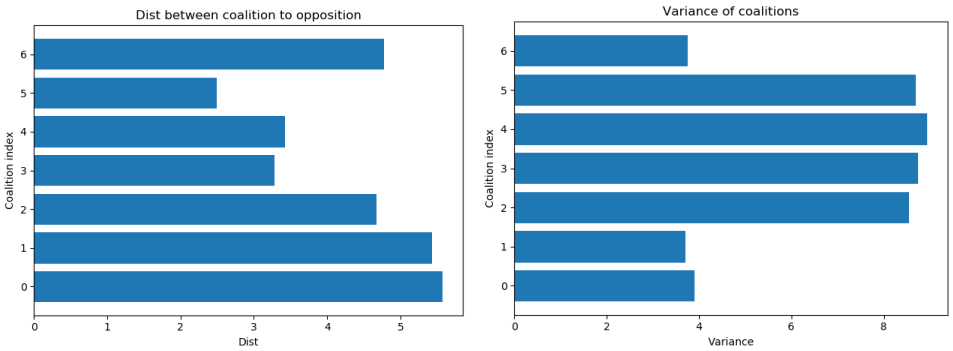
בסוף שלב זה קיבלנו כי שני המודלים מקבלים ניקוד אופטימלי עבור 2 קלאסטרים.

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 לייבים 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).

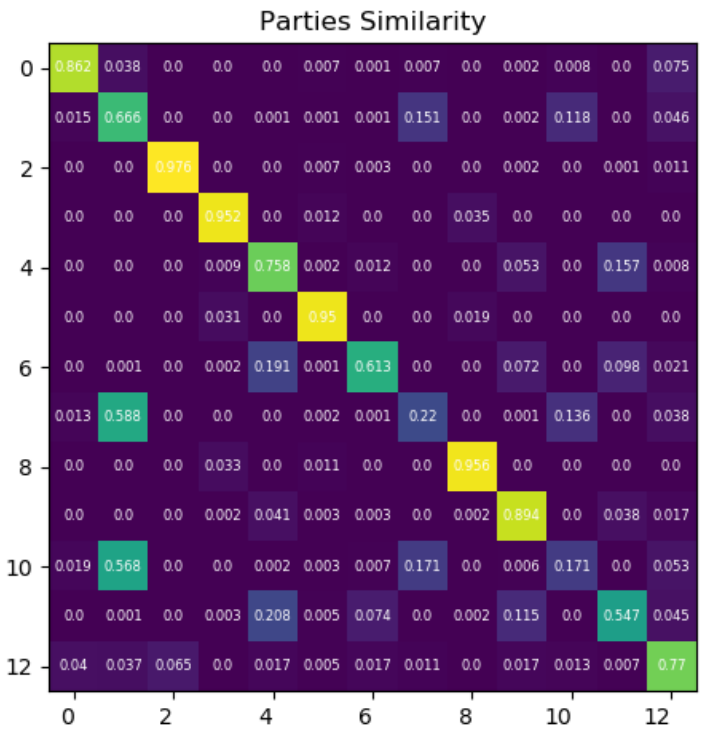
3. **מציאת קואליציות אפשריות על ידי מודל generative:**

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

לאחר מכן בנינו מטריצת הסתברויות לפי המודל הטוב יותר (GaussianNB) על ידי שימוש בתכונה כי מודל גנרטיבי משערך את התפלגות הדאטה שלנו, השתמשנו בפונקציה predict_proba לבניית המטריצה.
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 A 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.
מטריצה זו, נותנת לנו את הדמיון הקיים בין המפלגות (ככל שהערך במיקום מסוים (A,B) גדול יותר, כך יש הסתברות גבוהה יותר כי אדם אשר הצביע למפלגה A יצביע למפלגה B ולכן נוכל להסיק כי ישנו דמיון בין המפלגות)

המטריצה שקיבלנו הינה:



כעת אנו עוברים על כל האפשריות לקואליציה שבה מתחילה ממפלגה A בודדת ואז מוסיפים לה את המפלגה B שהבוחרים שלה הם עם הסתברות הכי גדולה לבחור במפלגה A.(דמיון גדול ביותר), נמשיך כך עד שמגיעים לפחות ל 51% מהמצביעים, כעת נוסיף קבוצת מפלגות זו לרשמית הקואליציות האפשריות.

לאחר שמגיעים ל 51% ממשיכים להוסיף מפלגות באותה צורה עד שלא נשאר מפלגות להוסיף וכל אפשרות כזו של קואליציה נכניס לרשימת הקואליציות האפשריות.

כעת, יש בידינו רשימה של קואליציות אפשריות, נרצה לבחור מתוכן את הקואליציה הטובה ביותר, כאשר נגדיר טובה ביותר בצורה הבאה:
קואליציה כמה שיותר גדולה, בעלת שונות קטנה, ומרחק כמה שיותר גדול מהאופוזיציה. לאחר

חישוב שלושת הערכים הנ"ל, ביצענו מיון כל רשימה (בצורה דומה לבחירת קואליציה בסעיף הקודם) ובחרנו בצורה ידנית את הקואליציה הטובה לדעתנו.

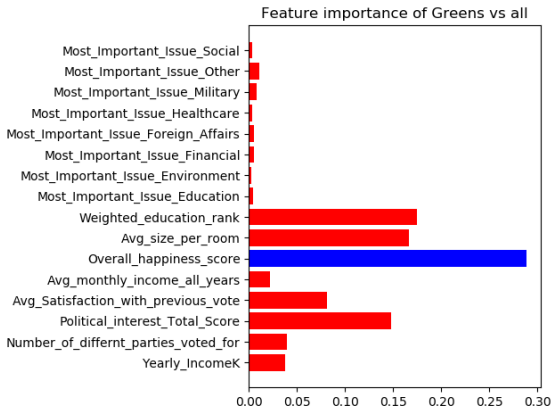
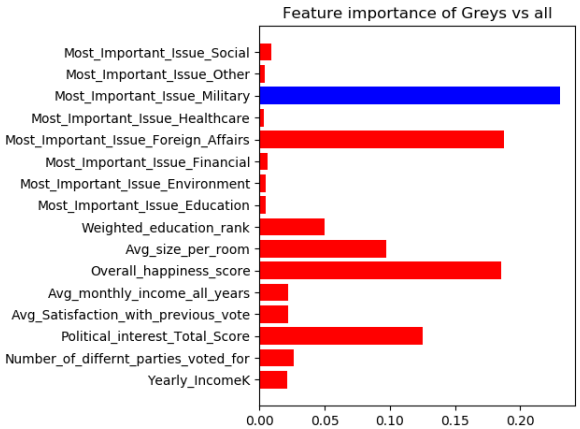
הקואליציה שבחרנו על ידי הנתונים מהמודל היא:

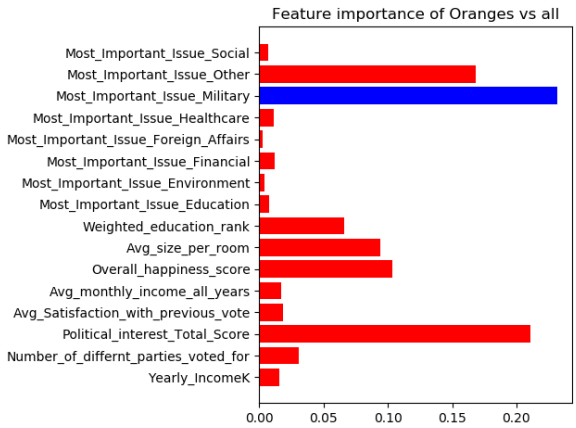
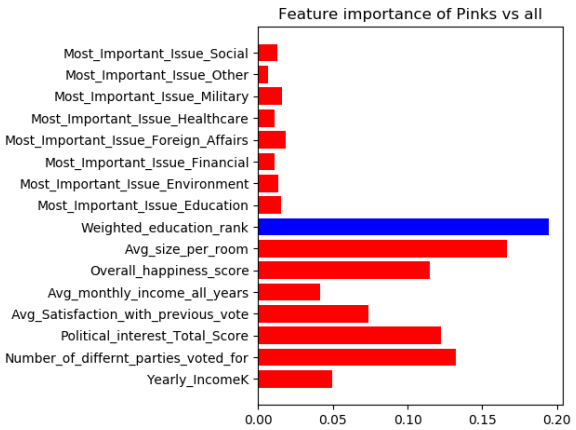
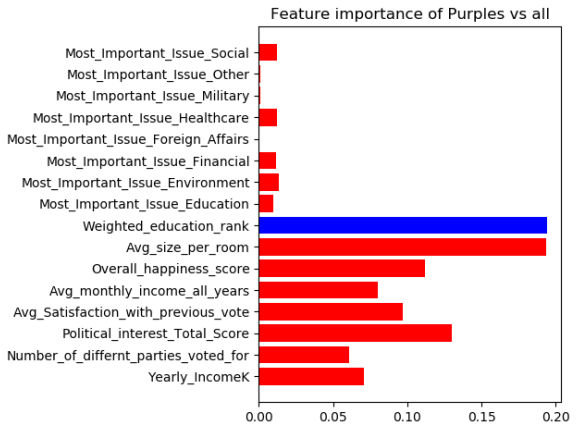
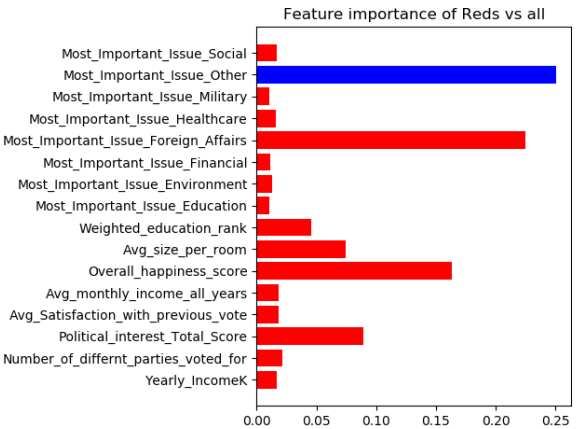
['Greys', 'Khakis', 'Oranges', 'Pinks', 'Reds', 'Turquoises', 'Whites', 'Yellows]

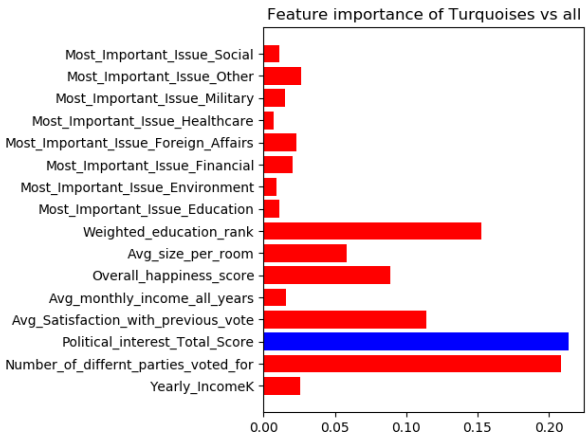
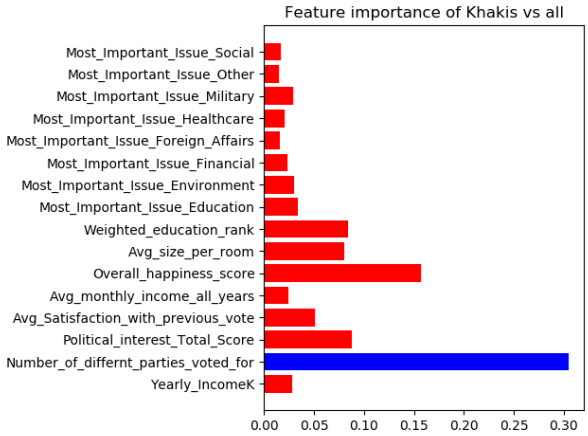
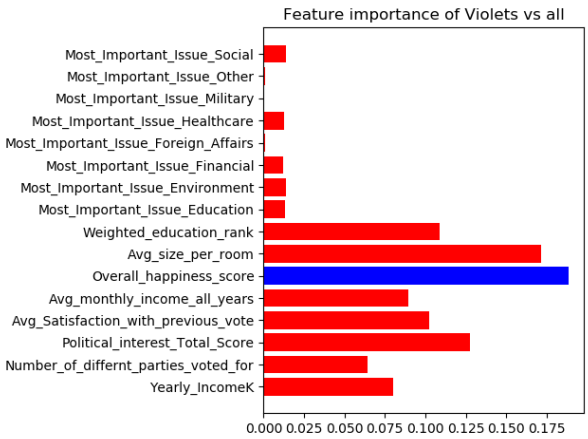
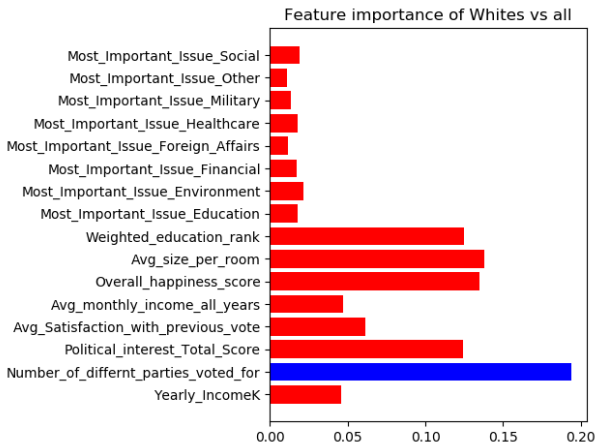
נשים לב כי זאת כמעט אותה קואליציה כמו שקיבלנו מהמודל של הcluster למעט המפלגה Greens

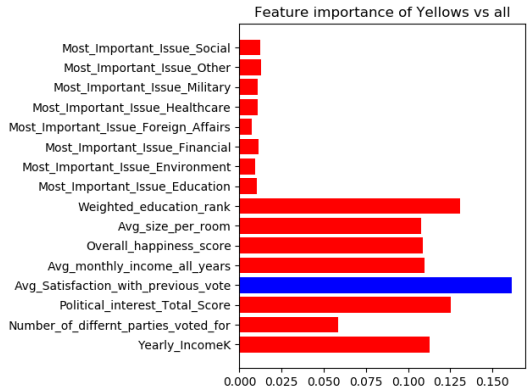
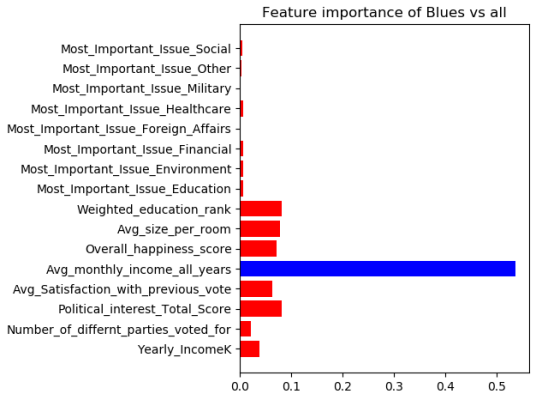
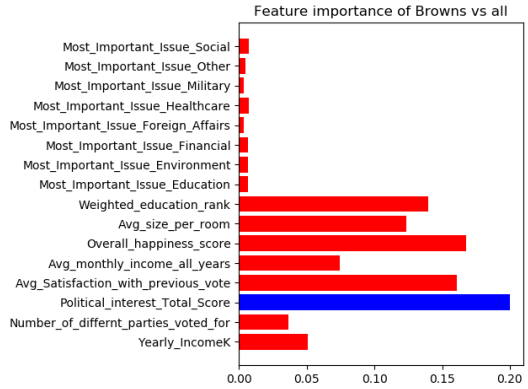
4. **זיהוי פיצ'רים מובילים עבור כל מפלגה:**

על מנת לבצע חלק זה, השתמשנו בשיטת של feature selection אשר השתמשנו בתרגיל בית 2.
שיטה זו הינה שיטת Wrapper אשר משתמשת במודל ExtraTreeClassifier המאפשר לקבל מידת חשיבות לכל פיצ'ר.
מימשנו חלק זה על ידי אימון מודל מסוג זה, על משימת סיווג בינארית בין מפלגה יחידה לכל השאר.
כלומר, ביצענו עבור כל מפלגה:
חלוקת הדאטה לone vs all , כל שהמפלגה הנוכחית היא 1 וכל שאר המפלגות הם 1- .
לאחר מכן אימנו את המודל על דאטה זה, והשתמשנו באופציה של feature_importances_ על מנת לקבל ניקוד עבור כל פיצ'ר.
ציון זה בעצם מציין כמה פיצ'ר זה מפריד בין הדוגמאות המתאימות למפלגה שלנו לאלה שלא שייכות אליה.
לאחר הסתכלות על התוצאות הללו, בחרנו את מספר הפיצ'רים בעלי החשיבות הגבוה ביותר להיות הפיצ'רים המובילים של מפלגה זו.
להלן התוצאות עבור כל מפלגה:

מפלגה	פיצ'רים מובילים	Features importance
Greens	1.Overall_happiness_score 2.Weighted_education_rank 3.Avg_size_per_room 4.Political_interest_Total_score	
Greys	1.Most_Important_Issue_Military 2. Most_Important_Issue_Foreign_Affairs 3. Overall_happiness_score	

	1.Most_Important_Issue_Military 2. Political_interest_Total_score 3. Most_Important_Issue_Other	Oranges
	1.Weighted_education_rank 2.Avg_size_per_room 3.Number_of_different_parties_voted_for 4.Political_interest_Total_score 5.Overall_happiness_score	Pinks
	1.Weighted_education_rank 2.Avg_size_per_room 3.Political_interest_Total_score	Purples
	1. Most_Important_Issue_Other 2. Most_Important_Issue_Foreign_Affairs 3. Overall_happiness_score	Reds

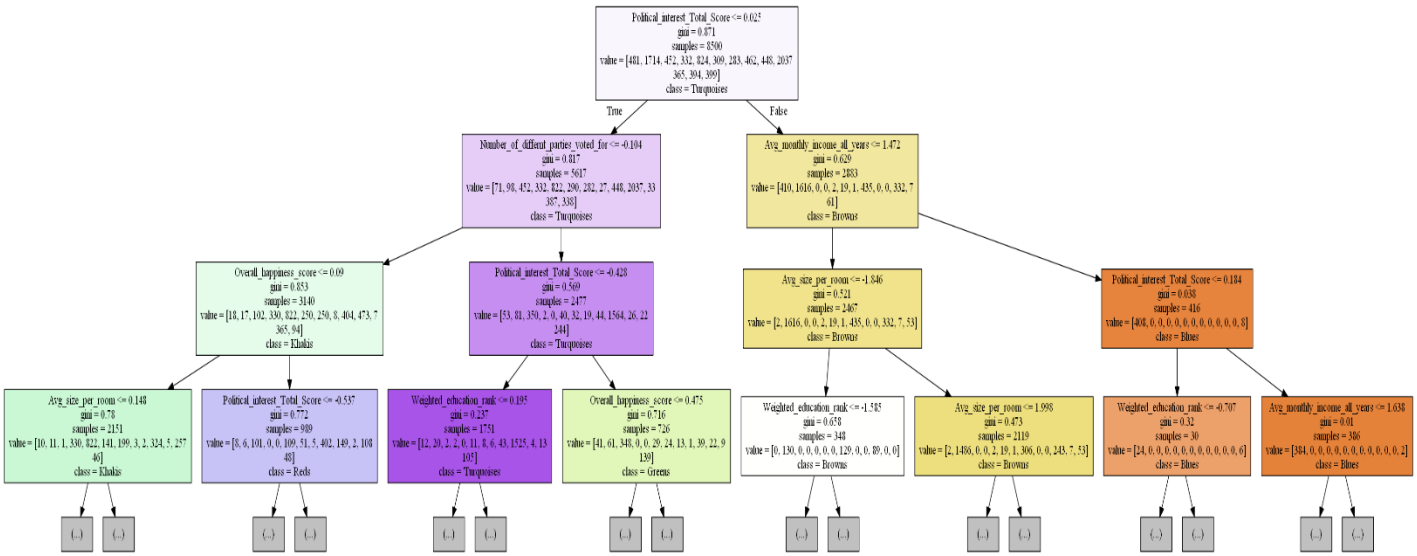
	1. Political_interest_Total_score 2. Number_of_different_parties_voted_for 3. Weighted_education_rank	Turquoises
	1. Number_of_different_parties_voted_for 2. Overall_happiness_score	Khakis
	1. Overall_happiness_score 2. Avg_size_per_room 3. Political_interest_Total_score	Violets
	1. Number_of_different_parties_voted_for 2. Avg_size_per_room 3. Overall_happiness_score 4. Political_interest_Total_score 5. Weighted_education_rank	Whites

	1. Avg_Satisfaction_with_previous_vote 2. Weighted_education_rank 3. Political_interest_Total_score	Yellows
	1. Avg_monthly_income_all_years	Blues
	1. Political_interest_Total_score 2. Overall_happiness_score 3 Avg_Satisfaction_with_previous_vote	Browns

5. מציאת פיצ'רים שעל ידי שינויים נקבל תוצאת בחירות שונה:

חלק זה נמצא בקובץ fourth_prediction.

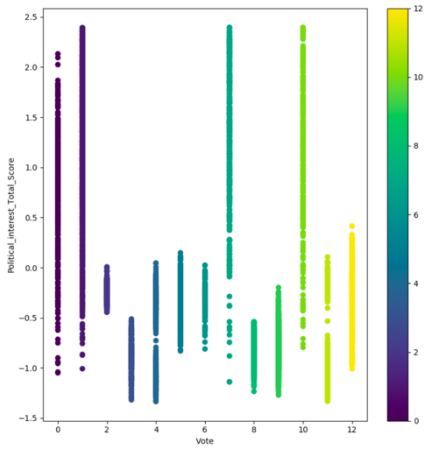
ראשית הסתכלנו על השכבות הראשונות של עץ החלטה וראינו מי מהתכונות בעלי משקל גדול על תוצאות המודל והסתכלנו על התכונות איך הן משפיעות על ההצבעה ואיך הן מפולגות.לכל תכונה בנפרד שינינו את קבוצת ה test עבור אותה תכונה כדי להשפיע על התוצאות של המנצח על ידי שהמסווג יטעה בגלל תכונה זו בסיווג. שינינו כל תכונה ולאחר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



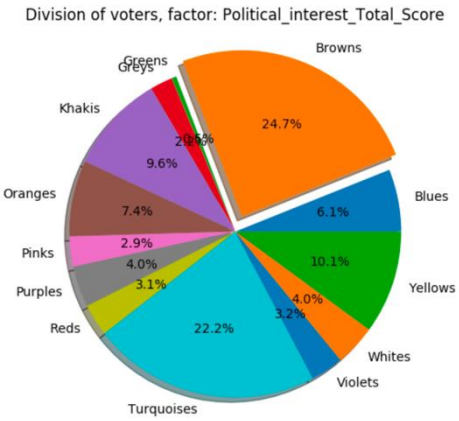
בנוסף ראינו מתרגיל קודם שהמנצח הוא Turquoises(9) ומקום שני הוא Brown (1) וקיבלנו את התכונות הבאות כקריטיות עבור תוצאות הסיווג

Political_interest_Total_Score : ניתן לראות כי המצביעים של 1 ו-9 מפולגים בצורה כזאת 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:

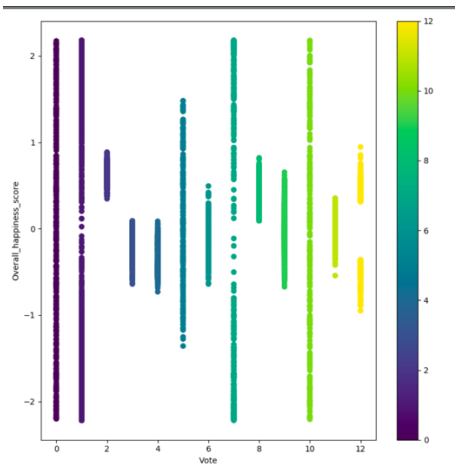


brown מנצח לאחר שינוי תכונה זו בtest:

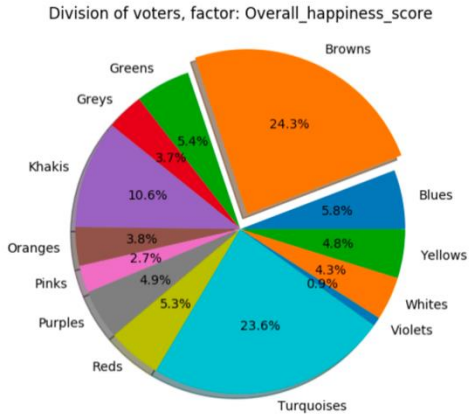


Overall_happiness_score : ניתן לראות כי המצביעים של 1 ו-9 מפולגים בצורה כזאת שעבור 1 יש בוחרים בטווח גדול יותר בכל הטווח מלבד האמצע ב והטווח עבור 9 קטן יותר ובאמצע.

פילוג של train data על תכונה זו ביחס להצבעות:

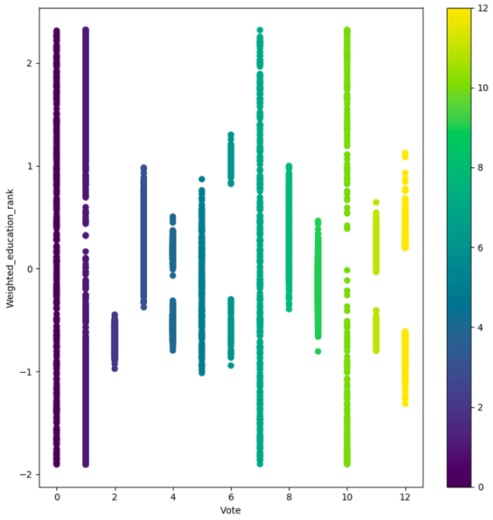


brown מנצח לאחר שינוי תכונה זו בtest:

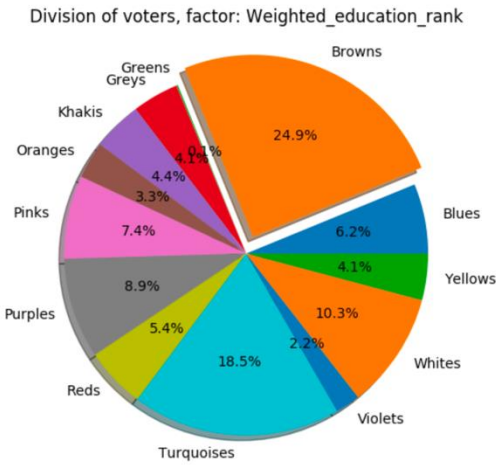


Weighted_education_ran: ניתן לראות כי המצביעים של 1 ו-9 מפולגים בצורה כזאת שעבור 1 יש בוחרים בטווח גדול יותר בכל הטווח מלבד האמצע והטווח עבור 9 קטן יותר ובאמצע.

פילוג של train data על תכונה זו ביחס להצבעות:

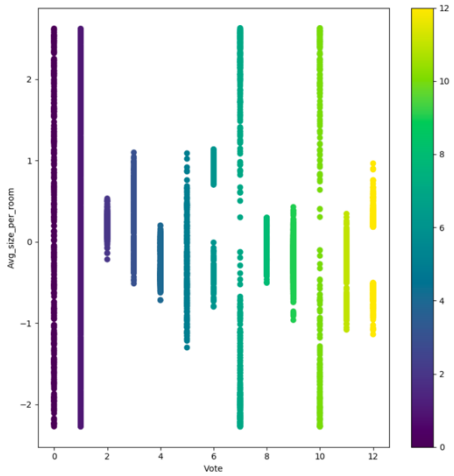


brown מנצח לאחר שינוי תכונה זו בtest:

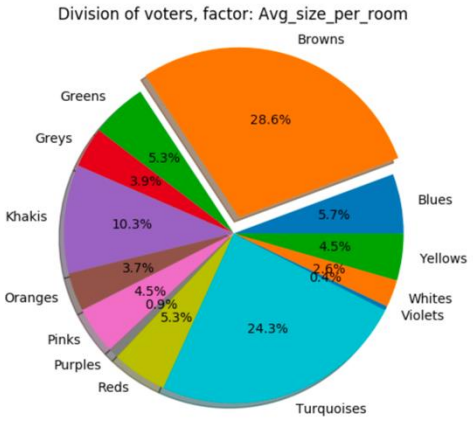


Size per room : ניתן לראות כי המצביעים של 1 ו-9 מפולגים בצורה כזאת שעבור 1 יש בוחרים בטווח גדול יותר בכל הטווח והטווח עבור 9 קטן יותר ובאמצע.

פילוג של train data על תכונה זו ביחס להצבעות:



brown מנצח לאחר שינוי תכונה זו בtest:



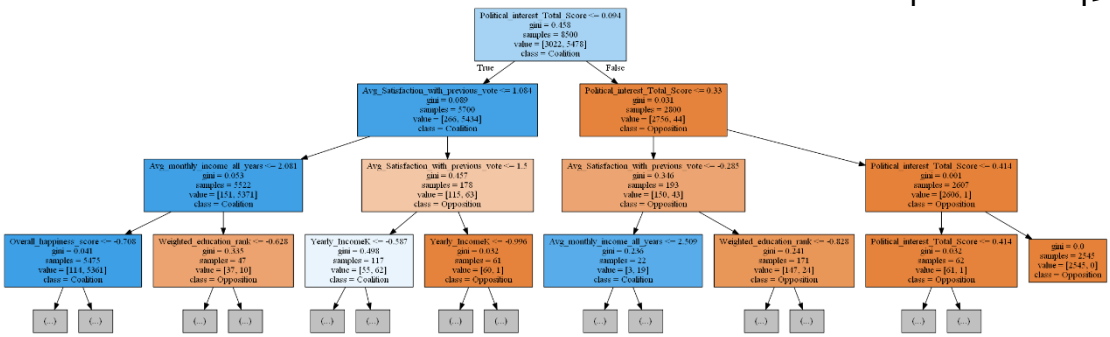
6. זיהוי הפיצ'רים שעל ידי שינויים נוכל לחזק את הקואליציה שלנו:

שלב זה ביצענו בצורה דומה לסעיף קודם.
תחילה בחרנו את הקואליציה שהחזיר לנו אלגוריתם ה cluster , כעת שינינו את הדאטה שלנו כך שני לייבלים, קואלציה ואופוזיציה.
כלומר עברנו על סט האימון, ועבור כל לייבל השייך לקואליציה הצבנו 1 ובכל שאר הלייבלים 1-.
כעת ביצענו אימון של עץ החלטה על סט אימון זה.
כעת עלינו לחזק את הקואליציה, הגדרנו חיזוק זה בצורה הבאה : חיזוק הקואליציה – הגדלת מספר המצביעים לקואליציה, הקטנת שונות הקואליציה (יותר הומוגנית), הגדלת מרחק הקואליציה מהאופוזיציה.
כעת על מנת למצוא את הפיצ'רים שעלינו לשנות השתמשנו ב3 דרכים:

א. הסתכלנו על עץ ההחלטה שקיבלנו, כאשר הסתכלנו על הרמות העליונות ביותר. רמות אלה מבצעות את ההפרדה הגדולה ביותר ולכן על ידי שינוי של אחד מהפיצ'רים המופיעים שם נוכל להשפיע בצורה גדולה יותר.
ב. חישוב ה Variance של כל הפיצ'רים של מצביעי הקואליציה.
לאחר שחישבנו מידע זה, הסתכלנו איזה מהפיצ'רים בעל שונות גדולה ביותר וכך נכוון להשפיע עליו על מנת להקטין את השונות של הקואליציה.
ג. חישוב מרחק ממוצע כל פיצ'ר ממצביעי הקואליציה לממוצע כל פיצ'ר ממצביעי האופוזיציה. לאחר חישוב זה הסתכלנו איזה פיצ'רים מבדילים כמה שיותר בין הקבוצות וננסה לשנות אותם עוד יותר כדי להגדיל את המרחק בין הקבוצות.

תוצאות כל אחד מהדרכים:

א. עץ ההחלטה שקיבלנו:



ניתן לראות כי הפיצ'ר Political_interest_total_score הינו המשפיע ביותר על פיצול הדוגמאות, לכן על ידי הקטנת פיצ'ר זה בכל הדוגמאות , נהפוך יותר מצביעים לקואליציה.

ב. פיצ'ר Number_of_differnt_parties_voted_for בעל השונות הגדולה ביותר, לכן נקרב את ערכיו לממוצע על מנת להקטין את השונות.
(נשים לב כי הכפלה זו תגרום גם להקטנת המרחק בין הקואליציה והאופוזיציה אבל בצורה קטנה יותר מאשר הקטנת השונות)
ג. פיצ'ר Political_interest_Total_Score בעל ההבדל הגדול ביותר בין הקואליציה לאופוזיציה, כאשר ממוצע הקואוליציה גדול מ-0 וממוצע האופוזיציה קטן מ-0.
ננסה לפצל עוד יותר את הדאטה על ידי הכפלה של הערכים.
(נשים לב כי הכפלה זו תגרום גם להגדלת השונות של פיצ'ר זה, אז בצורה פחות משמעותית)

תוצאות:
לפני השינוי –

גודל הקואליציה: 64.13%
שונות הקואליציה: 3.843
מרחק מהאופוזיציה: 5.843

אחרי השינוי –

גודל הקואליציה: 68.2%
שונות הקואליציה: 3.213
מרחק מהאופוזיציה: 6.375

נשים לב כי הצלחנו להגדיל את גודל הקואליציה, להקטין את שונות הקואליציה (יותר הומוגנית), ולהגדיל את המרחק של הקואליציה מהאופוזיציה. כלומר קיבלנו קואליציה יותר יציבה וחזקה.