תרגיל בית 2 מבוא למערכות לומדות

דוח עבודה

מגישים:

אביב כספי – 311136691

יקיר יהודה - 205710528

<u>שלבי עבודה:</u>

1. **הכרת הדאטה:**

א. דבר ראשון שעשינו בתחילת העבודה, היה זיהוי הדאטה שלנו, הבנה מהם הפיצ'רים הקיימים, כיצד כל אחד מהם מפולג, ומאיזה סוג הוא.

סוג התכונה:

נומינליות	נומריות
Will_vote_only_large_party	Occupation_Satisfaction
Age_group	Last_school_grades
Most_Important_Issue	Number_of_differnt_parties_voted_for
Main_transportation	Number_of_valued_Kneset_members
Occupation	Num_of_kids_born_last_10_years
Gender	Avg_monthly_expense_when_under_age_21
Looking_at_poles_results	Avg_lottary_expanses
Married	Avg_monthly_expense_on_pets_or_plants
Financial_agenda_matters	Avg_environmental_importance
Voting_Time	Financial_balance_score_(0-1)
	%Of_Household_Income
	Yearly_IncomeK
	Avg_size_per_room
	Garden_sqr_meter_per_person_in_residancy_area
	Avg_Residancy_Altitude
	Yearly_ExpensesK
	%Time_invested_in_work
	Avg_education_importance
	Avg_Satisfaction_with_previous_vote
	Avg_monthly_household_cost
	Phone_minutes_10_years
	Avg_government_satisfaction
	Weighted_education_rank
	%_satisfaction_financial_policy
	Avg_monthly_income_all_years
	Political_interest_Total_Score
	Overall_happiness_score

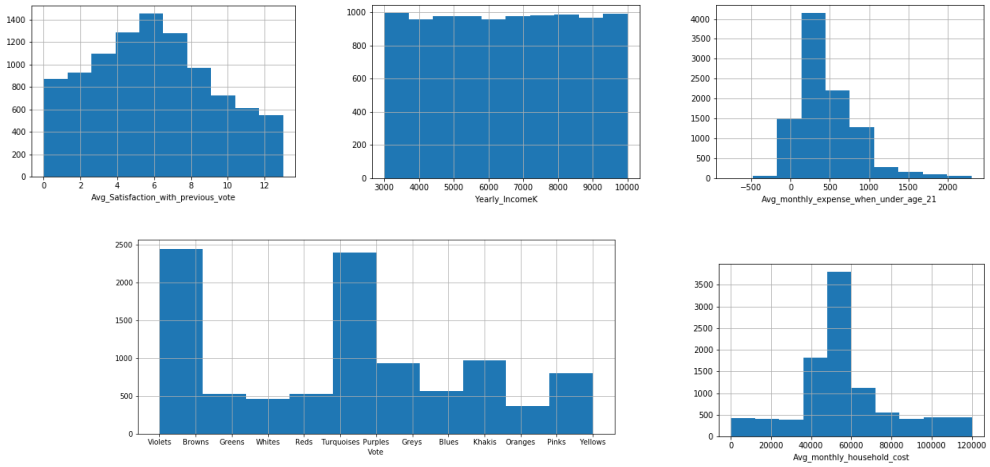
ב. לאחר זיהוי סוג הפיצ'רים, התמקדנו בפיצ'רים הנומינליים, והחלטנו באילו פיצ'רים ישנה חשיבות לסדר הערכים ולאלה לא. (אלה שלא הייתה חשיבות המרנו בהמשך לone-hot)

בלי חשיבות לסדר - One-hot	עם חשיבות לסדר	תכונות בינאריות
Most_Important_Issue	Will_vote_only_large_party	Gender
Main_transportation	Age_group	Looking_at_poles_results
Occupation		Financial_agenda_matters
		Married
		Voting_Time

ג. ויזואליזציה של הדאטה:

הדבר הבא שביצענו היה ויזואליזציה של הדאטה – הצגנו עבור כל פיצ'ר את התפלגות הערכים שלו, על מנת לראות כיצד הדאטה נראה ואיזה סוג התפלגות יש לכל תכונה.

כמה דוגמאות של היסטוגרמות חשובות:

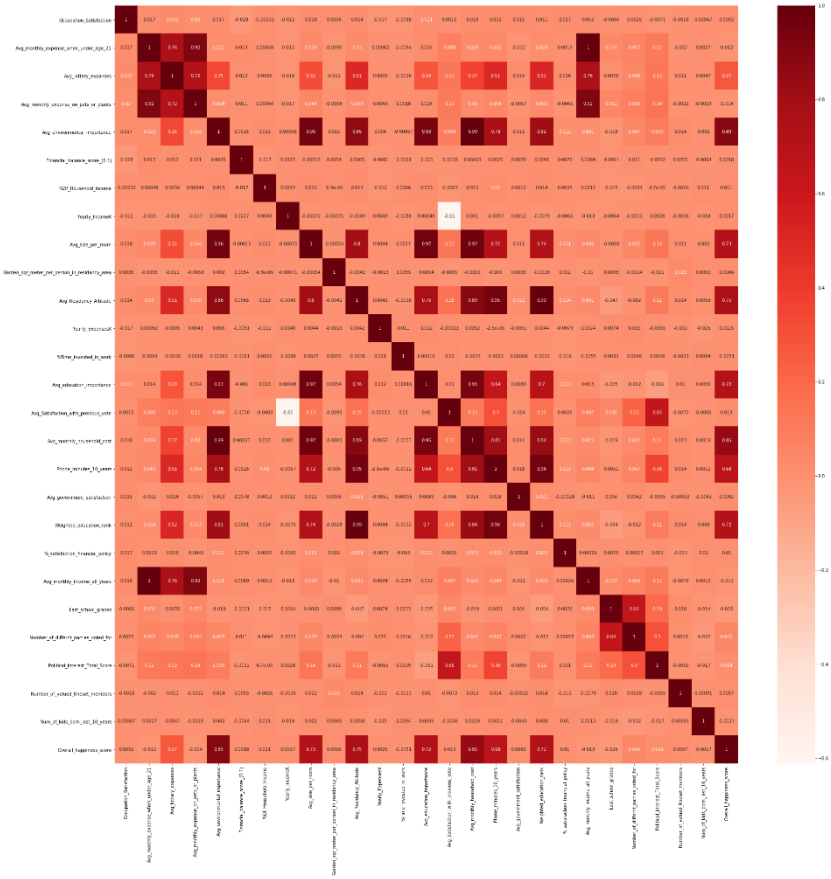


על פי היסטוגרמות אלה, זיהינו כי לכמה פיצ'רים יש ערכים שליליים למרות שערכי הפיצ'רים אמורים להיות חיוביים. בנוסף זיהינו כי הדאטה שלנו אינו מאוזן מבחינת לייבלים, וכי יש כמה מפלגות להן יש כמות דוגמאות גדולה יותר מהשאר. ובנוסף זיהינו כיצד פיצ'רים מתפלגים, כאשר חלקם מתפלגים בצורה יוניפורמית וחלקם נורמלית וכו'.

ג. קשרים בין פיצ'רים:

כעת רצינו לזהות קשרים בין הפיצ'רים לפני ביצוע שינויים לדאטה שלנו.

על מנת לבצע זאת, חישבנו את מטריצת הקורלציה והדפסנו אותה בצורה גרפית כדי לזהות קשרים בצורה קלה יותר:



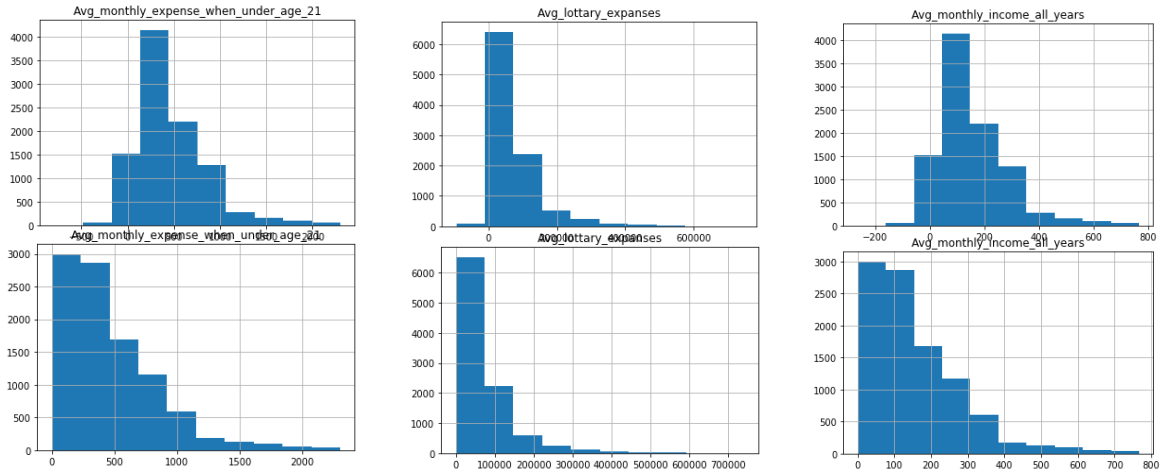
בעזרת מטריצה זו זיהינו כי יש כמה פיצ'רים שיש ביניהם קשר חזק מאד, ולכן נוכל בהמשך להוריד אותם מפני שאינם מוסיפים מידע לדאטה, לדוגמא Avg_monthly_expense_when_under_age_21 ו - Avg_monthly_income_all_years בעלי קורלציה 1, כלומר ישנו קשר לינארי מוחלט ביניהם.

כעת המשכנו לביצוע השינויים בדאטה:

2. **חלוקת הדאטה**:
תחילה ביצענו חלוקה של הדאטה ל3 קבוצות – train(70%), val(15%), test(15%)
ביצענו את החלוקה באמצעות הפונקציה StratifiedShuffleSplit של sklearn, כאשר החלוקה התבצעה בצורה אקראית כאשר נשמר היחס בין הלייבלים.
לאחר פיצול שמרנו את הדאטה המקורי ללא שינוי.

3. **המרת פיצ'רים למספריים**:
דבר ראשון שביצענו היה זיהוי כל הפיצ'רים וסוגם – כלומר זיהינו איזה פיצ'רים הינם נומינליים, בינארים, רציפים וכו'.
לאחר זיהוי זה, ביצענו המרה של הפיצ'רים לערכים מספריים על מנת שנוכל לעבוד איתם בהמשך, כאשר עבור פיצ'רים נומינליים ביצענו כמה שינויים – עבור פיצ'רים ללא חשיבות לסדר, ביצענו המרה ל one-hot, עבור פיצ'רים עם חשיבות לסדר ו-3 ערכים אפשריים (כמו "Age Group") ביצענו המרה לערכים 1 0 1-, לפי הסדר המתאים.
עבור פיצ'רים בינאריים ביצענו המרה ל 1 1- .
ביצענו זאת על מנת שהפיצ'רים יהיו ממורכזים סביב 0 (בהמשך נבצע scaling אשר ימרכז את כל הפיצ'רים ל0)
המרות אלה ממומשות בפונקציות :
convert_to_onehot, convert_to_categorical, change_binary_values

4. **תיקון ערכים:**
כאשר בחנו את הדאטה שקיבלנו, זיהינו כי ישנם כמה פיצ'רים אשר מכילים ערכים שליליים כאשר הפיצ'ר עצמו אינו אמור להכיל ערכים שליליים.
פיצ'רים אלה הם ' Avg_monthly_expense_when_under_age_21', 'Avg_lottary_expanses', ''Avg_monthly_income_all_years
לאחר שזיהינו זאת, בחנו את הפיצ'רים האלו על פי ההתפלגות שלהם וזיהינו כי הערכים השליליים נלקחים מתוך ההתפלגות של הפיצ'רים אם היו בעלי סימן חיובי.
כלומר לדעתנו בזמן הדגימה נפלה טעות והערכים נרשמו עם סימן שלילי במקום חיובי ולכן החלטנו לשנות את הסימן שלהם לחיובי במקום למחוק אותם.
פעולה זאת מתבצעת בפונקציה abs_negative.

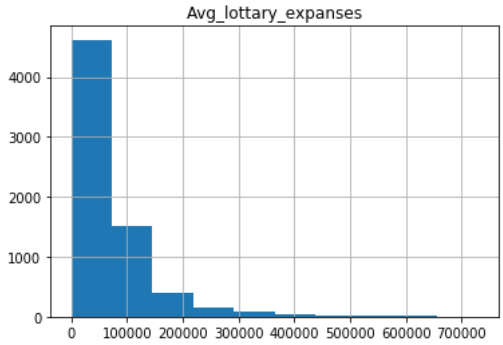


(ניתן לראות בהיסטוגרמות כי שני הפיצ'רים הקיצוניים בעלי התפלגות זהה לחלוטין – בדיוק כמו שציינו בחלק של מטריצת הקורלציה)

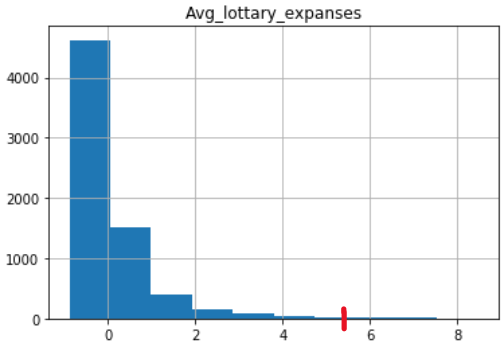
4. **Outlier detection:**
בהתמודדות עם outliers החלטנו לטפל רק בסט האימון שלנו, זה נובע מכך שבחרנו לבצע את זיהוי

ה – outlier על ידי התייחסות לליבל עצמו ולהתפלגות הדאטה ביחס לליבל.
עבורי הסטים של test, val בחרנו לא לבצע אף פעולה של זיהוי outlier בגלל שבמצב אמיתי, לא נוכל לדעת את הלייבלים של הדאטה ונצטרך לבצע פעולות ללא התייחסות לליבל.
לדעתנו אי טיפול זה גורם לכך שהבדיקה של הדאטה שלנו על סט הבדיקה תהיה כמה שיותר קרובה למצב אמיתי.
זיהוי outlier – על מנת לבצע זיהוי זה חישבנו z_score עבור כל אחד מהדוגמאות בדאטה שלנו ולכל פיצ'ר.
כעת בחרנו את הדוגמאות בעלות הפיצ'רים אשר מקיימים כי הz_score שלהם גדול מסף מסויים (בחרנו ב4.5 כסף) ומחקנו את המקומות המתאימים (כלומר הצבנו NAN בכל דוגמא עם פיצ'ר שזיהינו כ – outlier).
בשימוש בשיטה זאת, בעצם אפשרנו לדאטה שלנו בכל פיצ'ר להיות במרחק של 4.5 סטיות תקן מהממוצע. לדוגמה עבור התכונה avg_loattary expanese נקבל שדאטה מעל סטייה של 4.5 נמחק

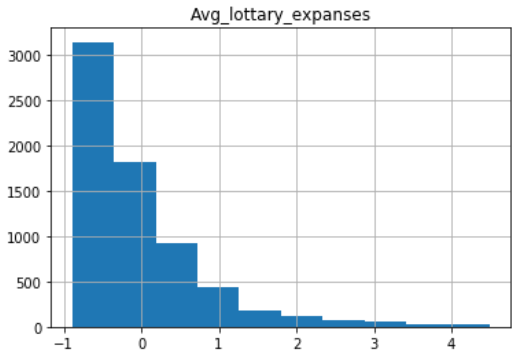
לפני נרמול



אחרי נרמול



אחרי הורדת outliers



פעולה זאת מתבצעת בפונקציה remove_outlier

5. **Imputation**:
את הטיפול בערכים חסרים ביצענו בשתי שיטות, כאשר שיטה אחת הופעלה את סט האימון שלנו, ושיטה שנייה על הסטים של הבדיקה.
נזכיר כי את הטיפול בסט האימון נוכל לבצע עם ייחוס לליבלים של הדאטה, כאשר עבור סט הולידציה והבדיקה לא ניתן להתייחס לליבל מפני שמידע זה לא ידוע לנו במקרה אמיתי.
א. טיפול בסט האימון:
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 לליבל ערך חדש.
כלומר, בהינתן דוגמה עם לייבל y, שלה חסר ערך בפיצ'ר מס' x, יצרנו סט חדש המכיל דוגמאות מתוך סט האימון שהלייבל שלהם הוא רק y (ללא ערכים חסרים בפיצ'ר מס' x), לאחר מכן דגמנו

מסט זה דוגמא אקראית A והעתקנו את ערך פיצ'ר x של הדוגמה זו לערך החסר בדוגמה אותה רצינו להשלים.
בצורה זאת, אנחנו שומרים על פילוג הדאטה שלנו ומשלימים ערכים חסרים עם חשיבות ללייבל המתאים להם.
ב. טיפול בסט הולידציה והבדיקה:
עבור סטים אלו 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.
בשיטה זאת, אנחנו מונעים התייחסות ללייבל בסט הולידציה והבדיקה, ועדיין משלימים ערכים חסרים עם חשיבות לדאטה שלנו ולא בצורה עיוורת.
פונקציות הלו ממומשות במחלקה Imputation

6.**Scaling**:
שלב אחרון בעדכון הדאטה שלנו הינו שלב הscaling.
בשלב זה ביצענו scaling לכל פיצ'ר נומרי (לא ביצענו אף שינוי לפיצ'רים הנומינליים, מפני שחלקם מסוג one-hot והחלק השני כבר ממורכז סביב 0).
עבור פיצ'רים המתפלגים בצורה יוניפורמית ביצענו min max scaling כאשר העברנו את ההתפלגות שתהיה בין 1 1- .
עבור פיצ'רים המתפלגים בצורה נורמלית ביצענו נורמליזציה כך שההתפלגות החדשה תהיה בעלת ממוצע 0 וסטיית תקן של 1.
נציין כי את הscaling ביצענו תחילה על סט האימון, כאשר חישבנו את ממוצע וסטיית התקן של סט האימון, ולאחר מכן השתמשנו בערכים אלה על מנת לבצע scaling לסט הולידציה והבדיקה.

6. **Feature Selection:**
כעת התחלנו לבצע את בחירת הפיצ'רים המתאימים ביותר:
<u>Filter Methods</u> – את שיטה זו בחרנו לבצע על ידי אלגוריתם selectKBest של sklearn , אשר מדרג את הפיצ'רים בדאטה שלנו על ידי פונקציית ניקוד (אנחנו בחרנו להשתמש בmutual_info) ובוחר את k הפיצ'רים הטובים ביותר.
אלגוריתם שני שהשתמשנו בו הינו אלגוריתם relief אשר מימשנו בסעיפי הבונוס, הרצנו את האלגוריתם למשך 2000 איטרציות ובחרנו את k הפיצ'רים עם המשקולות הגבוהות ביותר.
<u>Wrapper Methods</u> – את שיטה זו בחרנו לבצע על ידי אלגוריתם RFE של sklearn , עם מסווג מסוג RandomForestClassifier, בנוסף לכך השתמשנו גם באלגוריתם Tree-based feature selection עם מסווג ExtraTreesClassifier , אשר מאמן מסווג על הדאטה שלנו, והמסווג מגדיר את חשיבות כל פיצ'ר. מתוך הפיצ'רים בחרנו את k הטובים ביותר.
בנוסף לכך השתמשנו באלגוריתם SFS שמימשנו בסעיפי הבונוס, עם מסווג מסוג KNN ועץ החלטה.

לאחר ביצוע כל השיטות הללו, ביצענו איחוד וחיתוך בין קבוצות הפיצ'רים שנבחרו, כאשר החלטנו שכל הפיצ'רים הנמצאים בחיתוך יהיו כחלק מהפיצ'רים הנבחרים.
וכעת נותר לנו לבחור אילו פיצ'רים נוסיף מתוך הפיצ'רים שנותרו באיחוד.
על מנת לבצע זאת ביצענו חיפוש נוסף במרחב הפיצ'רים הנמצאים באיחוד.
בכל שלב השתמשנו בפונקציה שמימשנו בקובץ test אשר מאתחלת 3 מסווגים בסיסיים ומאמנת את המסווגים על הדאטה המכיל את הפיצ'רים שבחרנו, ומוציאה לנו את הדיוק של כל מסווג.
לסיכום קיבלנו את הפיצ'רים הבאים:

Yearly_IncomeK
Last_school_grades
Avg_education_importance
Avg_monthly_expense_on_pets_or_plants

Avg_Residancy_Altitude
Avg_Satisfaction_with_previous_vote
Number_of_differnt_parties_voted_for
Avg_size_per_room
Avg_monthly_household_cost
Phone_minutes_10_years
Overall_happiness_score
Political_interest_Total_Score
Avg_environmental_importance
Most_Important_Issue

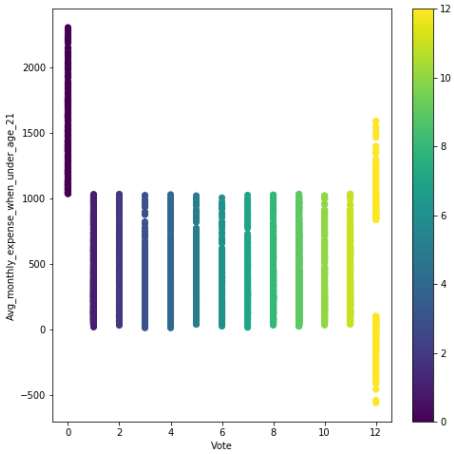
בונוס ראשון:

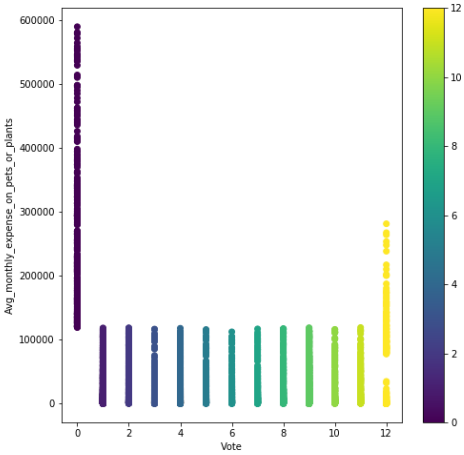
א. זיהוי קשר בין הלייבל לפיצ'רים:

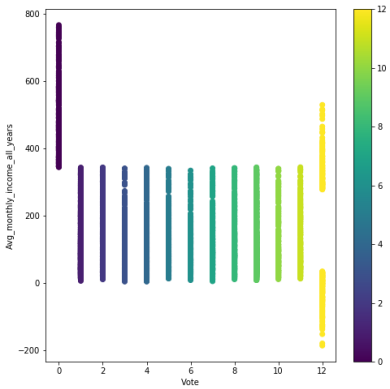
על מנת לזהות אם קיים קשר כלשהו בין הלייבל לפיצ'ר,
הדפסנו גרפים המתארים את הקשרים בין כל פיצ'ר ללייבל.
על ידי הסתכלות על גרפים אלו יכלנו לזהות המון קשרים חשובים בין התכונות של דגימה כלשהי לבין המפלגה לה הצביע אותו אדם.
מקרא המרה בין מפלגה למס' על מנת שיהיה נוח להציג את הדאטה:

'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

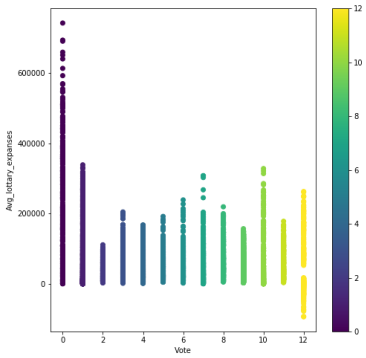
1. קשר בין מפלגה 0 לפיצ'רים הבאים:



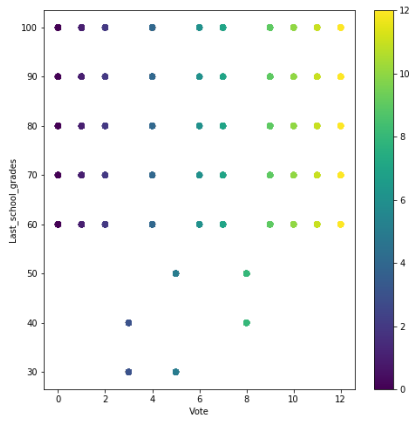




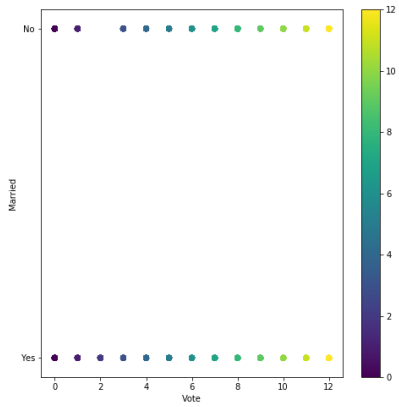
ניתן לראות כי המצביעים למפלגה 0, נוטים לבזבז יותר כסף עד גיל 21, ויותר כסף על חיות וצמחים ובנוסף ההכנסה השנתית שלהם גדולה מכל שאר המצביעים למלגות האחרות.
כלומר נסיק כי המצביעים למפלגה 0 הינם אנשים בעלי הון עצמי גדול משאר האוכלוסייה.
(אולי ניתן להסיק כי מפלגה 0 דואגת לעשירי החברה)
בנוסף לכך זיהינו כי מצביעי מפלגה 0, נוטים לבזבז יותר כסף בלוטו:



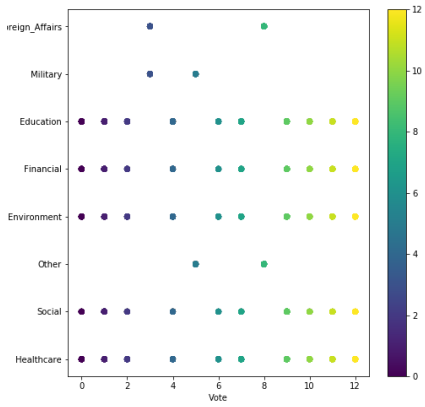
2. קשר בין ציוני בתי הספר לבין בחירת המפלגה:



ניתן לראות כי ישנן 3 מפלגות אליהן מצביעים אנשים בעלי ציונים נמוכים משאר המצביעים למפלגות אחרות: המצביעים למפלגות 3,5,8 בעלי ציוני בית ספר נמוכים מ-60 לעומת שאר המצביעים למפלגות האחרות אשר ציוני בית הספר שלהם גדולים שווים ל-60.
נוכל להסיק מכך כי ייתכן כי המצביעים למפלגות אלה הינם אנשים בעלי השכלה נמוכה יותר משאר המצביעים למפלגות האחרות.
3. קשר בין נישואים לבית הצבעה למפלגה 2:

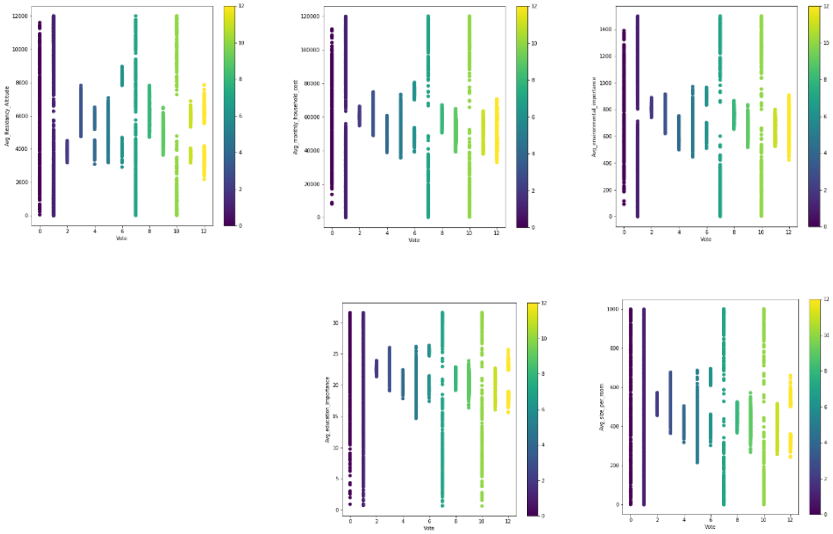


ניתן לראות כי לפחות לי הדאטה שקיים אצלנו, כל המצביעים למפלגה 2 הינם נשואים.
4. קשר בין נושא חשוב ביותר לבית המפלגות 3,5,8:

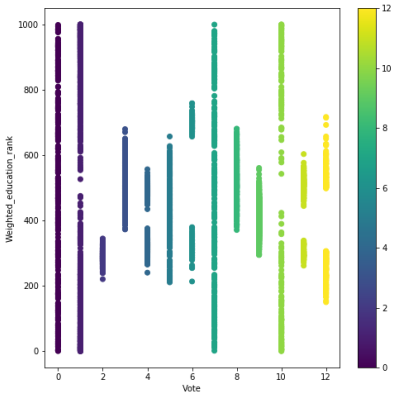


ניתן לראות כי המצביעים למפלגות 3,5,8 בחרו בנושא החשוב ביותר מבחינתם בצורה שונה משאר המצביעים למפלגות האחרות.

הנושאים אותם בחרו המצביעים למפלגות אלה הינם: Foreign Affairs, Military, Other
נזכיר, כי זיהינו קשר בין מצביעי מפלגות אלה לבין ציוני בית הספר שלהם, וכי מצביעים אלה בעלי ציוני נמוכים משאר המצביעים.
ייתכן כי ניתן לקשר בין הנושא החשוב ביותר לכל מצביע ובין הציוני שקיבל בבית הספר.
5. קשר בין כמה פיצ'רים ומפלגות 2-6, 8-9,11-12:

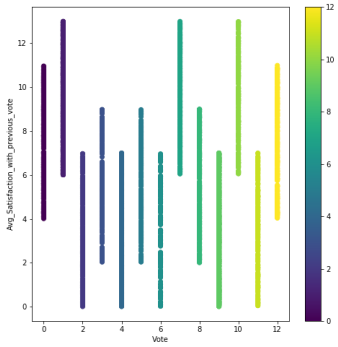


ניתן לראות בחמשת הפיצ'רים האלה, כי המצביעים למפלגות הנ"ל, בעלי נתונים עם טווח יותר מצומצם, וכי ניתן לכל מפלגה לראות איזה ערכים יש למצביעים שלה בניגוד למצביעים למפלגות אחרות.
בנוסף לכך ניתן לראות כי חלק מהגרפים מתארים את ההתפלגות בצורה כמעט זהה, וזה מרמז על קשר כלשהו בין הפיצ'רים שמתוארים בגרפים.
קשר נוסף למפלגות אלה:

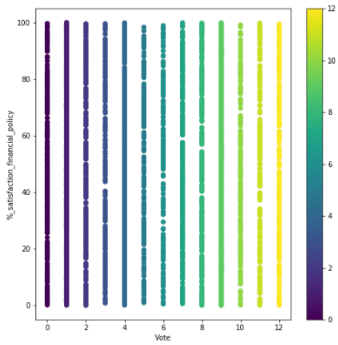


מבחינת דירוג השכלה, ניתן לראות כי המצביעים למפלגות הנ"ל הינם בעלי השכלה ממוצעת ופחות, לעומת המצביעים לשאר המפלגות אשר מתפלגים בצורה די אחידה ביחס לדירוג ההשכלה שלהם.

6. קשר בין שביעות רצון מהבחירה הקודמת לבין המפלגות הנבחרו כעת:



ניתן לראות כי המצביעים למפלגות 1,2,7,10,12
היו מרוצים יותר מהבחירה הקודמת שלהם מאשר שאר המצביעים, לכן נוכל אולי להסיק כי המצביעים הללו שמרו על ההצבעה שלהם לאותה מפלגה.
בנוסף נוכל להסיק בכיוון ההפוך כי ייתכן כי המצביעים לשאר המפלגות בחרו לשנות את ההצבעה שלהם בהצבעה הנוכחית.
7. בשיטה זו, ראינו כי ישנם המון פיצ'רים שלא ניתן לזהות קשר בין הפיצ'ר לבית המפלגה הנבחרת, לדוגמא:



ניתן לראות כי המצביעים לכל מפלגה מתפלגים בצורה די אחידה ולכן לא נוכל להסיק על קשר כלשהו.

בונוס relief

מימוש פונקציית relief אשר מקבלת data , מספר איטרציות (iteration) , וסף אשר יסמל את מספר הפיצ'רים בעלי משקל הגבוה ביותר שהפונקציה תחזיר.

פונקציה זו אשר משמשת לfeature selection יש יתרונות וחסרונות.

יתרונות:

- זמן ריצה קצר יחסית ,דוגמים כמספר האיטרציות ועבור כל דוגמה עוברים על כל הפיצ'רים ולוקחים min מאותו לייבל ו min מהשאר.
- אם עוברים הרבה איטרציות אז יש פחות חשיבות לדוגמאות רועשות.
- יש יחס לבחירת התכונות בהקשר ללייבל של התכונה.
- אפשר לשלוט על סף התכונות שנרצה לקבל ועל מספר האיטרציות שתעבור .
- אין קשר בין בחירת המסווג לבחירת התכונות.

חסרונות:

- בלי ידע קודם קשה לדעת איזה סף לקבוע למספר התכונות שהפונקציה תחזיר.
- צריך להריץ הרבה איטרציות כדי לקבל תוצאה טובה.
- פונקציית המרחק נותנת משקל גבוה יותר לתכונות נומינליות (שלא בהכרח תכונות חשובות) ואז תכונות אלו מקבלות משקל גבוה יותר ביחס לתכונות אחרות שהdata שלו מנורמל.
- הפונקציה בוחרת רנדומלי דוגמאות ולפי דוגמאות אלה מחשבת משקלים ודוגמאות אלה לא בהכרח מיצגות.

תכונות אשר נבחרו על ידי relief ולא על ידי השאר:

- 'Avg_lottary_expanses'
- 'Avg_monthly_income_all_years'
- 'Avg_size_per_roomWeighted_education_rank'

תכונות אשר לא נבחרו על ידי relief ועל ידי השאר:

- 'Last_school_grades'
- 'Avg_environmental_importance'
- 'Yearly_IncomeK'
- 'Avg_size_per_room'
- 'Phone_minutes_10_years'
- 'Avg_Satisfaction_with_previous_vote'
- 'Avg_monthly_household_cost'

בונוס SFS

מימוש האלגוריתם נמצא בקובץ SFS.py בפונקציה SFS

פונקציה זו מקבלת:

model – מודל בעל פונקציות של fit ו – predict , אשר מאמנות את המודל ומבצעות חיזוי בנוגע לדאטה.

X_train, y_train, X_test, y_test – קבוצות האימון והבדיקה שלנו, אליהם נאמן את המודל.

והפונקציה מחזירה את שמות הפיצ'רים שנבחרו.

אנחנו בחרנו לממש גירסא אשר משתמשת ביוריסטיקה כי יש לעצור את החיפוש ברגע שיש רק ירידה בדיוק של המסווג, כלומר בכל איטרציה נבצע אימון על סט הפיצ'רים שנבחרו עד עכשיו, פלוס פיצ'ר נוסף שעוד לא נבחר. כאשר נראה כי הוספת כל פיצ'ר שנשאר, גורמת לירידה בדיוק, נעצור את החיפוש ונחזיר את התוצאה.

הרצנו את האלגוריתם עם שני מודלים בסיסים:

תוצאות ההרצה:

KNN – selected features :
['Avg_monthly_expense_on_pets_or_plants', 'Avg_environmental_importance', 'Avg_size_per_room', 'Avg_Residancy_Altitude', 'Avg_education_importance', 'Avg_monthly_household_cost', 'Phone_minutes_10_years', 'Weighted_education_rank', 'Last_school_grades', 'Number_of_differnt_parties_voted_for', 'Political_interest_Total_Score', 'Overall_happiness_score', 'Most_Important_Issue_Foreign_Affairs', 'Most_Important_Issue_Military', 'Most_Important_Issue_Other']
Accuracy before: 0.675
Accuracy after: 0.881

Forest – selected features :
['Avg_monthly_expense_when_under_age_21', 'Avg_environmental_importance', 'Avg_size_per_room', 'Avg_Residancy_Altitude', 'Phone_minutes_10_years', 'Weighted_education_rank', 'Last_school_grades', 'Number_of_differnt_parties_voted_for', 'Overall_happiness_score', 'Most_Important_Issue_Foreign_Affairs', 'Main_transportation_Car']
Accuracy before: 0.897
Accuracy after: 0.91

יתרונות וחסרונות של SFS:

<u>יתרונות</u>:

- קל מאד למימוש
- מימוש דינמי, ניתן בקלות לשנות את המטריקה המשמשת למדד ביצועים בכל איטרציה, ולהתאים לצורך הנוכחי
- ניתן לממש גירסא המחזירה מספר פיצ'רים לפי בקשת המשתמש

<u>חסרונות</u>:

- איטי מאד בגלל הצורך לבצע מספר אימונים בכל איטרציה
- לפעמים נפגע מבעיית האופק (לפי המימוש שלנו), ייתכן כי הוספת שני פיצ'רים תעלה את הדיוק אבל כל פיצ'ר בנפרד אינו משפר ולכן האלגוריתם לא יוסיף אף אחד מהפיצ'רים
- מתאים למודל המתקבל – לא תמיד פיצ'רים המתאימים למודל שהתקבל באלגוריתם יתאימו לכל מודל אחר
- בעיית overfitting – בחירת הפיצ'רים מתבצעת לפי סט הולידציה ולכן ישנה סכנה של overfitting

- שימוש ביוריסטיקה שבחרנו אינו בהכרח אופטימלי, וייתכן כי לכל בעיה יש יוריסטיקה המתאימה לה יותר.

תכונות אשר נבחרו על ידי SFS ולא על ידי השאר:

- 'Main_transportation_Car'
- 'Avg_monthly_expense_when_under_age_21'
- 'Weighted_education_rank'
- 'Most_Important_Issue_Foreign_Affairs'

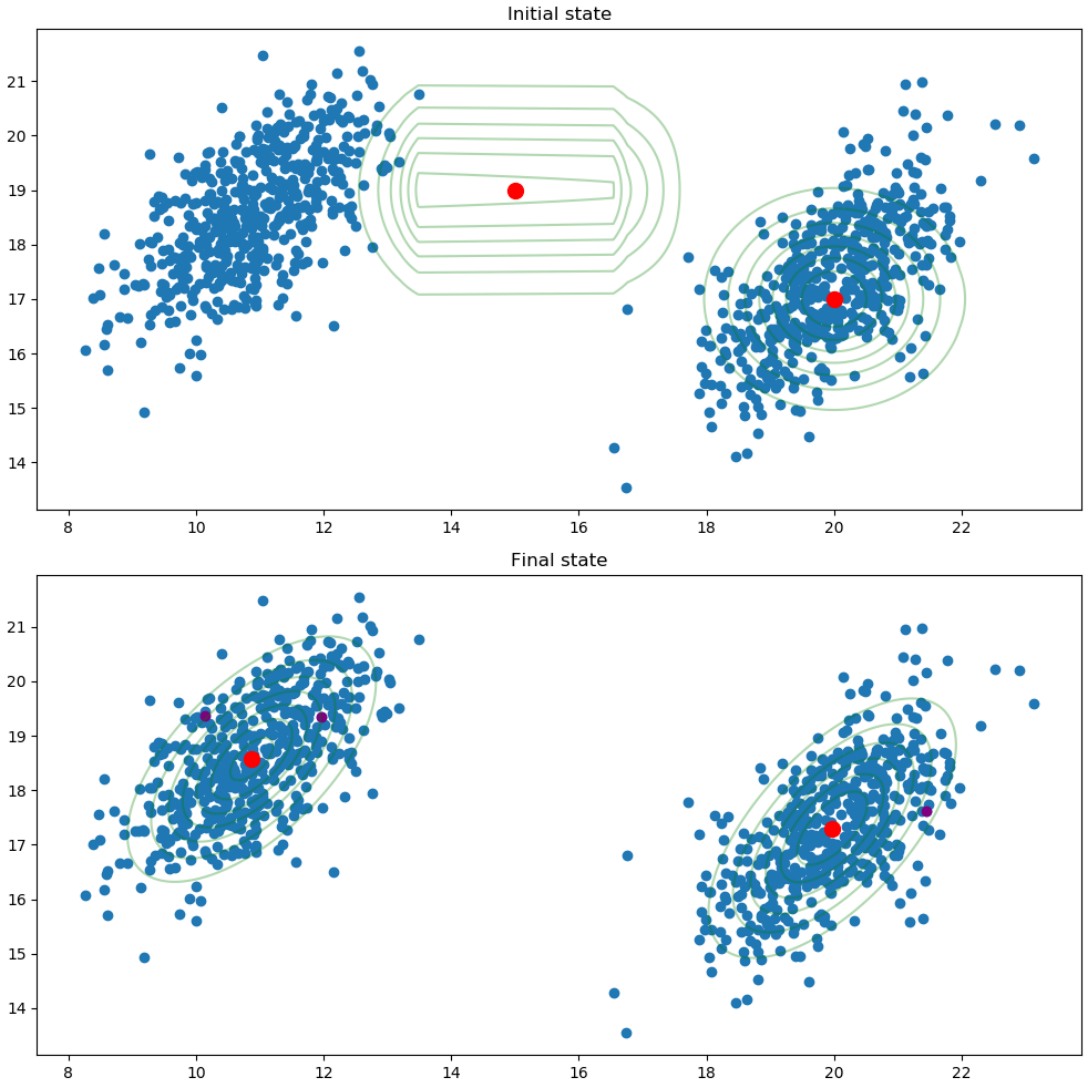
תכונות אשר לא נבחרו על ידי SFS ועל ידי השאר:

- 'Yearly_IncomeK'
- 'Avg_education_importance'
- 'Avg_monthly_expense_on_pets_or_plants'
- 'Most_Important_Issue_Military'
- 'Avg_monthly_household_cost'
- 'Avg_Satisfaction_with_previous_vote'
- 'Most_Important_Issue_Other'
- 'Political_interest_Total_Score'

בונוס EM

מימוש האלגוריתם נמצא בקובץ EM.py תחת המחלקה EM
בנוסף לכך הוספנו לקובץ אופציה הרצה של סט שייצרנו בעל 2 התפלגויות נורמליות אשר מייצג את תוצאות האלגוריתם.
על מנת לאתחל אובייקט של EM יש לספק את מספר ההתפלגויות שמצפים למצוא, ואת מספר האיטרציות המקסימלי שמאפשרים לאלגוריתם לרוץ (האלגוריתם עוצר ברגע שהתכנס או ברגע שעבר את מספר האיטרציות המקסימלי)
על ידי פונקציית FIT אשר מקבלת את סט האימון, אנחנו מבצעים את האלגוריתם ומוצאים את הפרמטרים החסרים.
בנוסף הוספנו פונקציה של predict אשר מקבלת נק' (לאחר ביצוע פונקציית FIT) ומחזירה עבור כל נקודה מה ההסתברות שהיא שייכת לכל אחת מההתפלגויות שנמצאו.

דוגמא לתוצאה הרצה:



הנקודות האדומות הינן הדאטה שלנו, הנקודות האדומות הן הממוצע בכל איטרציה של ההתפלגויות וניתן לראות את ההתפלגות הנורמלית בצבע ירוק.
בתמונה העליונה מתוארת המערכת לפני אימון, כלומר ההתפלגות מוגרלת בצורה אקראית בתחום הדאטה שלנו.
בתמונה התחתונה מתוארת המערכת לאחר אימון, ניתן לראות כי כל התפלגות התכנסה לאזור אחר של הדאטה וכי הן מתאימות לדאטה.
בנוסף הוספנו סימון של 3 נק' בצבע סגול, אשר בקוד שהגשנו ישנה הפעלה של פונקציה predict על נק' אלה על מנת להציג את תוצאות האימון.
נציין כי עבור סט אימון בעל מעל 3 מוקדים, היה צורך בהפעלה חוזרת של האלגוריתם לפני שהוא הצליח להתכנס להתפלגויות האמיתיות (נובע מאתחול לא טוב מספיק וכי האלגוריתם מתכנס למקס' מקומי ולא גלובלי)