**מערכות לומדות תרגיל 4**

**שם:** אסף חייק ברוך **ת.ז:** 206783441

**שם:** בן בנוז **ת.ז:** 207570573

**בתרגיל זה הוצבו בפנינו 3 בעיות:**

1. מציאת קואליציה יציבה כך שיותר מ-51% מהמצביעים הצביעו למפלגות המרכיבות אותה והמצביעים שהצביעו להם קרובים יחסית זה לזה אבל שונים מאוד ממצביעי מפלגות האופוזיציה
2. זיהוי תכונות שעל ידי מניפולציה שלהן נוכל להכריע את מנצחת הבחירות
3. זיהוי תכונות שעל ידי שינוי שלהן נוכל ליצור קואליציה יציבה כרצוננו.

**דרכי השגת הפיתרון לבעיות:**

ראשית כפי שצוין בתרגיל השתמשנו ב-generative models וב-clustering models כדי להגיע לפתרונות הרצויים, לצורך התרגיל גם עשינו שימוש במודל מהתרגיל הקודם עם היפר פרמטרים מסוימים שהראה את הperformance הטוב ביותר בקשר לבעיית חיזוי תוצאות הבחירות (היסטוגרמת התוצאה הסופית), וכמובן השתמשנו במידע המעובד מהתרגילים הקודמים.

**ראשית את בעיית הקמת הקואליציה היציבה תקפנו ב-2 דרכים:**

**א. דרך clustering**

ראשית אנחנו משתמשים ב-2 שיטות clustering כדי לפלג את המצביעים של כל מפלגה ל-3 קלאסטרים לשם כך השתמשנו בשיטות הclustering הבאות:

**\***MiniBatchKMeans- כלומר אלגוריתם kmeans שנלמד בהרצאה (עם שינוי קטן שמוסיף מהירות ושלא משנה תוצאות באופן משמעותי)

**\***BayesianGaussianMixture – גרסא יותר יציבה של GMM

לאחר מכן השתמשנו ב-clusters שכל שיטה הוציאה בתור labels לאימון מודל על ה-features המקוריים, בעצם אימנו 2 מודלים מסוג RandomForestClassifier, כאשר כל מודל מתאמן על תכונות המצביעים בתור features ועל הקלאסטרים שכל שיטה הוציאה בתור labels (זה על כל הקלאסטרים שמצאנו אצל המפלגות השונות ביחד).  
ההיפר פרמטרים שבחרנו הם אלה שמצאנו בתרגיל הקודם.

לאחר מכן השתמשנו בפונק' predict\_proba של כל אחד מהמודלים על סט המצביעים כך שהמודלים מחזירים את ההסתברות של כל מצביע להיות שייך לכל אחד מהקלאסטרים, והשתמשנו בdot product בין וקטורי ההסתברויות של כל 2 מצביעים כמדד לקרבת 2 המצביעים זה לזה, ועל ידי כך יצרנו מטריצת קישוריות שהיא בעצם מטריצה בממדים n\_samples\*n\_samples כך שמקום הi,j בה יש את רמת הקרבה (dot product) של המצביע ה-i למצביע ה-j (המטריצה הנ"ל סימטרית).  
לפני מכפלת הוקטורים החלטנו לאפס את כל ההסתברויות שקטנות מערך מסוים (בחרנו 0.3) בכדי לחסוך בזמן ריצה ולוודא שהדימיון נחשב רק החל מערכים שמביעים רמת ביטחון סבירה של השייכות.

כעת נשתמש במטריצת הקרבה בין samples כדי לבצע אלגוריתם clustering סופי שיחלק את המצביעים למצביעי קואליציה ולמצביעי אופוזיציה, אלגוריתם הclustering שהשתמשנו בו הוא AgglomerativeClustering , זה בעצם אלגוריתם שמאתחל כל דוגמא לקלאסטר משלה ולאחר מכן מאחד קלאסטרים לפי המרחק האוקלידי ביניהם לפי מטריקה מסויימת (אנחנו בחרנו אחת שמצמצמת variance של כל קלאסטר), הוא משתמש במטריצת הקרבה שהגדרנו כדי להעריך עד כמה הדוגמאות נחשבות דומות בנוסף למטריקה הרגילה (השימוש הוא כמטריצת connectivity בין הדוגמאות). הפלט הוא 2 קלאסטרים שונים ולכן נוצרות קואליציה ואופוזיציה עבור ה- clustering method שציינו בהתחלה.

**ב. דרך generative models**

בדרך זו השתמשנו בgenerative models הבאים:

* LinearDiscriminantAnalysis **-** מפריד לינארי (LDA).
* QuadraticDiscriminantAnalysis **-** מפריד ריבועי (QDA).

ראשית אימנו כל אחד מהמודלים הללו על סט המצביעים שלנו כאשר המפלגה לה הצביעו היא label.

לאחר מכן נייצר מכל אחד מהם קואליציה בצורה הבאה:

1. נשתמש במדד Jensen-Shannon שהינו מדד לקרבה בין שתי התפלגויות, במילים אחרות נמדוד קרבה בין מצביעי מפלגות על ידי קרבת ההתפלגות שלהם.

לגבי Jensen-Shannon:

כל אחד מהמודלים הנ"ל שומר אצלו חישוב של ממוצע הדוגמאות של כל מחלקה ואת השונות של כל סט המידע עליו התאמן.

נשתמש במידע זה כדי לדגום סט דוגמאות כל פעם של מצביעים עבור כל 2 מפלגות ונשתמש בהם כדי לחשב את מדד Jensen-Shannon לגבי הקרבה בין הקבוצות.

1. לאחר שחישבנו את המדד עבור כל שתי מפלגות נתחיל לאחד מפלגות שמצביעיהם קרובים זה לזה עד אשר נקבל קואליציה כאשר נעצור לאחר שקיימת קבוצת מפלגות שיש לה יותר מחצי מהמצביעים והאיחוד הבא לא ישפר את מדד  
   Davies-Bouldin שלה.  
   Davies-Bouldin הוא מדד עבור קלאסטרים שמציג את רמת הקרבה של דוגמאות שנמצאות באותו קלאסטר וכמה דוגמאות מקלאסטרים שונים שונות זו מזו, וכך נמשיך להוסיף עוד מפלגות לקואליציה עד אשר לא נקבל שיפור במדד.
2. נחזיר קואליציה עבור כל אחד מה- הgenerative models .

לבסוף נעבור על כל הקואליציות שקיבלנו מכל השיטות וננקד את טיב הפירוק לקואליציות על פי מדד Davies-Bouldin וניקח את הטובה ביותר.

בתוצאות שלנו יצא כי GMM הוציא את הקואליציה הכי יציבה, עם ניקוד Davies-Bouldin של 1.75. הקואליציה שמורה בקובץ coalition.csv. לקואליציה ניקוד completeness של 0.5, והיא לוקחת את המצביעים של רוב המפלגות חוץ מ-  
Khakis, Turquoise, Violets, שמהן היא לוקחת בערך חצי מהמצביעים.

אנחנו ממליצים להריץ את קובץ ה-main איך שהוא בהגשה בכדי לראות את פילוג הקלאסטרים, המפלגות והקואליציות השונות בתלת מימד לפי ערכי PCA.

**מציאת התכונות שיקבעו את מנצחת הבחירות ואת הקואליציה**

את הבעיה הזו והבעיה השלישית פתרנו על ידי בדיקת שינוי התכונות לטווחים שונים על התפלגות קולות הבחירות שהמודל שלנו מהתרגיל הקודם זוכר, כלומר מהתרגיל הקודם יש ברשותנו מודל שחוזה בדיוק רב את התפלגות הקולות בין המפלגות בבחירות, נאמן אותו על סט המצביעים ואז נשנה את התכונות של המצביעים לטווחים מסוימים שאנחנו בודקים ונבדוק מה המודל חוזה על סט המצביעים הנ"ל לגבי התפלגות הקולות בין המפלגות, ולאחר מכן נבדוק את השינוי שחל בין תוצאות האמת (היסטוגרמת ההצבעה למפלגות של סט המידע המקורי) ולפיו נראה מה השפעתו על תוצאות הבחירות.  
בתוך הטווחים הגרלנו ערכים אקראיים באופן אחיד, ולקחנו את הממוצע של החיזוי מתוך 50 הגרלות שונות.

בתוצאות האמת מפלגת הסגולים אמורה לנצח עם כרבע מכלל הקולות.

בצורה זו שמנו לב לתכונות הבאות:

ראשית כאשר מספר חברי הכנסת המוערכים הם בטווח של 20 ומעלה מפלגת האפורים לוקחת יותר מ60% מהקולות ומנצחת בצורה מוחצת בבחירות.

וכאשר מספר חברי הכנסת המוערכים הם 0 או 1 אז מפלגת החומים סוחפת 38% מהקולות בעוד המנצחת המקורית הסגולים נופלת ל0% מהקולות, וכך החומים לוקחים את הבחירות.

-----חלק של אסף----

לאחר בחינת התכונות כפי שתוארה קודם הגענו למסקנות הבאות לגבי האפשרות להרכבת קואליציה יציבה כרצוננו:

ראשית כאשר מספר חברי הכנסת המוערכים הם בטווח של 20 ומעלה מפלגת האפורים לוקחת יותר מ60% מהקולות ומקימה את הקואליציה הכי יציבה שאפשרית שכן היא מורכבת ממפלגה אחת ויחידה.

שנית אם נרצה להעלות את מספר הקולות למפלגת הסגולים ולחזק אותה בתור מובילת הקואליציה נדאג שיהיו בין 5 ל10 חברי כנסת מוערכים עבור כל מצביע , נשמור על רמת חינוך נמוכה בין 0 ל200 ונדאג שכל מצביע יוציא הוצאה שנתית של בין 5000 ל 7000 שקלים, ונדאג לresidency\_altitude בין 0 ל7 .

מדוגמאות אלו ניתן לראות שמספר חברי הכנסת המוערכים וההוצאות השנתיות ורמת החינוך הם גורמים ראשיים בבחירת הקואליציה.