**שלב 1**

מציאת שני הדוברים בעלי המספר הכי גדול של משפטים בקורפוס התבצעה ע"י כך שעשינו חילוץ לכל השורות בקובץ ה- JSON לתוך מבנה נתונים מסוג list, בו כל תא הכיל מילון המייצג את המשפט והמיתא דאטה שלה. לאחר מכן, עברנו על השורות, ולכל דובר ספרנו את כמות המשפטים שהוא אמר, סידרנו את המונים של הדוברים החל מהגדול, והחזרנו את השניים הראשונים. התוצאה שהתקבלה, המופיעה למטה הם של "ראובן ריבלין" ו- "א' בורג".

בקוד, חילוץ השורות מתבצע ע"י הפונקציה 'json\_lines\_extract', וספירת הדוברים והחזרת השמות של השניים בעלי כמות המשפטים הגדולה ביותר מתבצעת ע"י הקריאה לפונקציה 'top\_two\_speakers'.



**שלבים 1.1, 1.2**

על סמך התוצאה הנ"ל, "ראובן ריבלין" ו- "א' בורג" יהוו את שתי המחלקות speaker1, speaker2 בהתאמה.

על מנת לענות על דרישת המשימה, יצרנו מתודה בשם 'split\_data\_by\_speaker'. המתודה מקבלת כקלט את רשימת השורות של ה- JSON ושתי מחרוזות של שמות דוברים, ומחזירה 3 רשימות כך שבאחת את השורות שמשויכות לדובר הראשון, בשנייה את השורות המשויכות לדובר השני, ובשלישית את שאר השורות. בשלב הזה עוד לא חילצנו את הטקסט של השורות (המשפט עצמו) באופן נפרד והמשכנו לשמור אותם עם המיטאדאטה בשל כך שנזדקק למיטא דאטה בשלב 3.2.

על מנת למקסם את כמות המשפטים עבור כל דובר, ועל סמך האבחנות שלנו מהמטלה הראשונה, הגרסאות הנפוצות בהם יכול שם הדובר להופיע: כשם מלא (שם פרטי + שם משפחה, למשל, ראובן ריבלין), כאות הראשונה בשם הפרטי + שם המשפחה (למשל, ר' ריבלין), ככינוי הידוע בו + שם המשפחה (למשל, רובי רבלין), כשם משפחה בלבד (למשל, ריבלין). כל הגרסאות נלקחו בחשבון, כאשר הראשונה נבדקה ע"י כך שהשם תואם בדיוק לשם של הדובר, השניים שאחרי נבדקו ע"י כך שהמילה האחרונה בשם תואמת למילה האחרונה בשם הדובר, בנוסף לכך שהאות הראשונה במילה הראשונה בשם תואמת לאות הראשונה במילה הראשונה שבשם הדובר, והגרסה האחרונה נבדקה ע"י כך שבמקרה שהשם מורכב ממילה אחת היא תואמת למילה האחרונה בשם הדובר. כך למשל כל השמות הבאים משויכים לאותו אדם על פי הכללים שהוזכרו:

ראובן ריבלין

ר' ריבלין

רובי ריבלין

ריבלין

הפונקציה 'split\_data\_by\_speaker' נקראה מה- main().

**שלב 2**

ה- down-sampling התבצע ע"י כך שמצאנו את האורך המינימלי מבין 3 הרשימות (המכילה את השורות של הדובר הראשון, המכילה את השורות של הדבור השני, המכילה את שאר השורות). ומתוך כולם בחרנו באופן אקראי שורות כמספר הזה.

מספר השורות בכל אחת מהמחלקות לפני ביצוע הפעולה:

תמונה שמכילה טקסט, גופן, צילום מסך

התיאור נוצר באופן אוטומטי

מספר השורות בכל אחת מהמחלקות לאחר ביצוע הפעולה:

תמונה שמכילה טקסט, גופן, צילום מסך

התיאור נוצר באופן אוטומטי

**שלב 3**

**שלב 3.1**

לשם יישום דרישות השלב, יצרנו את הפונקציה 'tfidf\_vector\_creator', אשר מקבלת אוסף של שורות כקלט, ומחלצת מתוכן את המשפטים (הטקסט). לאחר מכן, היא יוצרת ווקטוריזר מסוג Tf-idf, מתאימה אותו לטקסט ויוצרת מטריצה שמורכבת מהווקטורים של המשפטים, ומחזירה את המטריצה ביחד עם הווקטוריזר.

העדפנו להשתמש ב- Tf-idf על פני CountVectorizer, מפני שהוא ממשקל את הטוקנים לא רק לפי כמות הופעתם אלא גם לפי הייחודיות שלהם על פני המסמך, ומפני שחשוב לנו להדגיש את המלים הייחודיות של הדובר, דבר היכול לעזור בסיווג של משפט, בנוסף לרצון שלנו בלהמעיט מהערך של טוקנים נפוצים כמו "של" ו- "את" ו- "," אשר מצמצמים את ההבדלים בין משפטים, ראינו ש- Tf-idf הוא הבחירה הנכונה.

**שלב 3.2**

על מנת לענות על דרישות המשימה, יצרנו את הפונקציה 'custom\_vector\_creator' אשר מקבלת כקלט אוסף של שורות (המשפט עם המטא-דאטה שלו), ורשימה של שמות הדוברים, ומייצרת וקטור מאפיינים לכל משפט המכיל את המאפיינים הבאים עם ההסבר של הסיבות מאחורי בחירת המאפיין:

1. **מספר הכנסת**: זהו מאפיין חשוב, מפני שחברי הכנסת נבדלים במספרי הכנסת בהם השתתפו, ולא נרצה שמשפט יסווג לחבר כנסת שלא היה חלק מהכנסת בה המשפט נאמר. מספר הכנסת יכול להיות משמעותי גם עבור חברי כנסת שהשתתפו באותה כנסת אם הם נבדלים בתפקידים שמילאו, מכיוון שתפקיד יכול להשפיע על כמות המשפטים שהוא אומר לצד התוכן שלהן. למשל: חבר כנסת הממלא תפקיד יו"ר מדבר יותר לעומת חברי הכנסת האחרים, כמו גם מרבה לומר משפטים מסוימים כמו "אני קורא לך לסדר בפעם .." לעומת חבר כנסת אחר. מצד שני המשפט "אדוני היושב – ראש" צפוי יותר מחבר כנסת שהוא לא יו"ר לעומת יו"ר.
2. **סוג הפרוטוקול**: מאפיין זה יכול להיות משמעותי בשל היותו משקף את ההקשר בו הדובר נוהג להשתתף. למשל, דובר בעל תפקיד מרכזי בכנסת, כמו יו"ר, צפוי להשתתף יותר בדיוני המליאה, לעומת חבר כנסת שהוא חבר ועדה אשר נוטה יותר להשתתף בדיונים של הועדות.
3. **אורך המשפט**: מאפיין זה יכול להבדיל בין שני דוברים בשל כמה סיבות. סיבה אחת היא אופי אישי, יש דוברים שנוטים לדבר משפטים יותר ארוכים מאחרים. סיבה אחרת היא שאורך המשפט יכול להעיד על סוג הדיון, אנחנו מצפים מדיוני ועידות שיהיו יותר מקצועיים ויכילו משפטים המכילים פירוט רב, ובכך גם אורכים יותר. סיבה שלישית היא שאורך המשפט מושפע מתפקיד הדובר בכנסת. למשל, אנחנו מצפים שליו"ר יהיו הרבה משפטים בעלי אורך קצר כמו "בבקשה חבר הכנסת", "תודה לחבר הכנסת", "אני פותח את הישיבה".
4. **אם המשפט מכיל ספרה**: הכלת המשפט לספרה או שלא יכולה גם כן להעיד על סגנון אישי, למשל דוברים מסוימים יכול להיות עם נטייה של להזכיר עובדות או סטטיסטיקות יותר מאחרים. המאפיין יכול גם לשקף את ההקשר וסוגי הדיונים בהם הדובר נוטה להשתתף, כאשר למשל, השפה המקצועית של דיוני הועדות יכולה להכיל יותר מספרים מאשר הדיונים של המליאות.
5. **מאפיינים הנוגעים לכמות הופעת קולקציה מסוימת בתוך המשפט מתוך קבוצת קולקציות**: מאפיין זה של קולקציות חשוב מפני שדוברים יכול להיות בעל נטייה להשתמש בקולקציות מסוימות כמו "אני", "אדוני", ו- "גברתי". כמו כן, הקולקציות יכולים לשקף את הנושאים בהם חבר הכנסת יותר מעורב, או אם הוא חלק מהקואליציה או האופוזיציה. למשל בכנסת הנוכחית, השימוש במילה "רפורמה משפטית" יכול להעיד על חבר כנסת מהקואליציה, לעומת חבר כנסת מהאופוזיציה שיעדיף את הביטוי "מהפכה משפטית". כמו כן, הקולקציה יכולה להעיד על התפקיד של הדובר, כמו בדוגמאות שהוזכרו לפני כן.

בחירת הקולקציות נעשית באופן דינמי ע"י אותה פונקציה 'custom\_vector\_creator'. הפונקציה עוברת על כל זוג דוברים מבין הדוברים הנתונים לה ברשימת הדוברים תוך התחשבות ברצף (כלומר, בהינתן שני הדוברים speaker1, speaker2 קיימים עבורם שני זוגות (speaker1, speaker2) ו- (speaker2, speaker1)), ומוצאת את 10 הקולקציות בעלות ההבדל הגדול יותר בתדירות ההופעה לטובת הדובר הראשון בזוג. הקולקציות של כל זוג מתווספות ל- set על מנת להימנע מכפילויות.

**שלב 4**

לצורך המשימה, אימנו 4 מסווגים, 2 (KNN, Logistic Regression) לכל סוג וקטור מאפיינים. המסווגים הוערכו באמצעות שיטת Cross Validation עם 5 folds כנדרש. התוצאות עבור כל מסווג מצורפות למטה.

למסווגים מסוג Logistic Regression הוחלט להשתמש בערכי ברירת מחדל, מלבד הפרמטר של **מס' האטירציות המקסימלי (max\_iter)** שנבחר להיות 1000, מפני שערכים נמוכים יותר, כולל של בחירת המחדל, לא הבטיחו תמיד התכנסות בסיום האטירציות, במיוחד במקרה של Tf-idf, כנראה בשל היותו בעל מספר תכונות גדול. ההחלטה על השארת ערכי ברירת המחדל של הפרמטרים האחרים נובעת מהתאמתם למשימה שלנו שכוללת ערכים צפופים, כמו למשל, l2 המונע התאמת יתר.

למסווגים מסוג KNN כן ראינו לנכון להשתמש בערכים השונים מהערכים של ברירת המחדל. עבור שני המסווגים השתמשנו ב- **n\_neighborhood = 8**, מתוך מחשבה שהערך 5 של ברירת המחדל לא נותן מספיק דיוק ועלול להיות רגיש לרעשים במשימה של סיווג משפטים, למשל במקרה שהמשפט מכיל באופן מקרי מלים הייחודיות לדובר האחר. על כן, השאיפה הייתה להגדיל את מספר השכנים באופן שיבטיח יציבות אך לא יביא להתאמת יתר, במיוחד שיש ברשותנו מספר גדול של דוגמאות אימון לכל מחלקה. פרמטר אחר שבחרנו לשנות הוא **weights**, בכך שהחלטנו להשתמש בשיטת המשקול של 'distance' עבור שני המסווגים מפני שהשכנים יכולים להיות בעלי אחוזי דמיון משתנה מאחד לשני, וחשוב לנו שהמשקל יהיה יחסי לקרבה ושתהיה העדפה לשכנים הקרובים יותר. עבור שיטת חישוב המרחק, בווקטור Td-idf הוחלט להשתמש בערך ברירת המחדל של המרחק האוקלידי, מפני שהשוני בין ערכי תכונות הווקטורים קריטיים ויש משמעות למרחק בהם. מצד שני, בווקטור המאפיינים שלנו, רוב התכונות לא היו בעלי ערכים רציפים שהמרחק בהם בעל משמעות, אלא בינאריים המצביעים על קיום התכונה או אי-קיומה. למשל: האם המשפט מכיל ספרה, או האם הוא מפרוטוקול המליאות, ובשל כך הוחלט להשתמש במרחק מנהטן ע"י לתת ערך 1 לפרמטר p.

תמונה שמכילה טקסט, צילום מסך, גופן, מספר

התיאור נוצר באופן אוטומטי

תמונה שמכילה טקסט, צילום מסך, גופן, מספר

התיאור נוצר באופן אוטומטי

תמונה שמכילה טקסט, צילום מסך, גופן, מספר

התיאור נוצר באופן אוטומטי



**שלב 5**

בהתאם לנדרש במשימה, יצרנו פונקציה בשם 'sentences\_classify' אשר מקבלת כקלט מודל, וקטורייזר, שמות שני הדוברים שנמצאנו בשלב הראשון, ושתי כתובות אחת לקלט ממנו יקראו משפטים, ואחת לפלט אליה יוכתבו התוצאות. הפונקציה קוראת את הקלט, מבצעת את הסיווג לאחד מבין שלושת הקטגוריות (שני הדוברים, ו- "other"), וכותבת את התוצאה לפלט תוך שהיא מחליפה את שמות שני הדוברים ל- first ו- second.

במשימה הוחלט להשתמש במסווג Logistic Regression בגלל שהציג יכולת דיוק טובה יותר בשלב 4.