שלב 1

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



שלב 1.1

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

על מנת לענות על דרישת המשימה, יצרנו מתודה בשם 'split\_data\_by\_speaker'. המתודה מקבלת כקלט את רשימת שורות של ה- JSON ושתי מחרוזות של שמות דוברים, ומחזירה 3 רשימות כך שבאחת את המשפטים שמשויכות לדובר הראשון, בשנייה את המשפטים המשויכות לדובר השני, ובשלישית את שאר המשפטים.

על מנת למקסם את כמות המשפטים עבור כל דובר, ועל סמך האבחנות שלנו מהמטלה הראשונה, הגרסאות הנפוצות בהם יכול שם דובר להופיע: כשם מלא (שם פרטי+שם משפחה), כאות הראשונה בשם הפרטי + שם המשפחה, ככינוי הידוע בו (רובי, בני) + שם המשפחה, כשם משפחה בלבד. כל הגרסאות נלקחו בחשבון, כאשר הראשונה נבדקה ע"י כך שהשם תואם בדיוק לשם של הדובר, השניים שאחרי נבדקו ע"י כך שהמילה האחרונה תואמת למילה 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 משוייכים לאותו אדם:

ראובן ריבלין

ר' ריבלין

רובי ריבלין

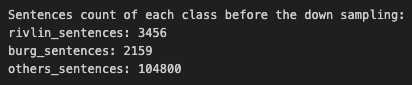
ריבלין

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

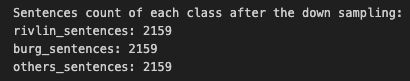
שלב 2

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

מספר המשפטים בכל אחת מהמחלקות לפני ביצוע הפעולה:



מספר המשפטים בכל אחת מהמחלקות לאחר ביצוע הפעולה:



שלב 3