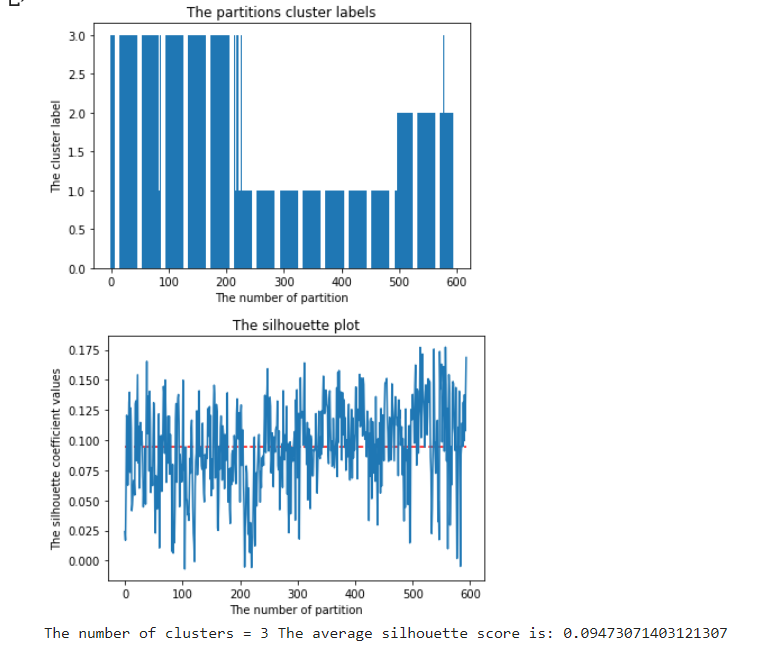
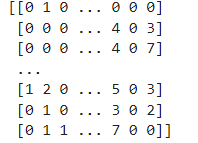
1. Open and read 3 the files 'Eliot.txt','Tolkien.txt', and HP\_Small.txt
2. Concatenate the text files into a single string.
3. Do the text preprocessing **without changing the text size** (do not remove extra blanks).
4. Calculate the reduced word frequency matrix.
5. Use the k-means algorithm to find the books boundaries. Show the clustering results in subplots.



מטריצת השכיחויות שחישבנו להכנה לאלגוריתם: (מאוד גדולה)



הסבר : בחלק זה של הקוד ביצענו ערבוב מספר 1 , מטרתו לקחת את כל הקבצים יחדיו להפוך לסטרינג אחד גדול ולאחר מכן לחלק לצאנקים

המחשבה בצורה הזאת היא שישארו פחות חלקים לקראת סוף החלוקה שלא מצליחים להיות בדיוק בגודל האצנק.

בנוסף אנחנו רואים שגודל הסילוט בממוצע הוא 0.095 , מעבדה קודמת ראינו שהקלאסטר האידאלי הוא 3 עבור 3 קבצים ולכן הפקטור של חלוקת לקלסטרים כידוע אינה משפיעה בצורה גדולה על הסילואט ולכן הבעיה הגדולה היא חלוקת הצאנקים.

בצורה שבה ביצענו חלוקה אפשרנו לפחות DATA להיכנס לגודל כל צאנק ולכן לאלגוריתם היה בהשוואה למעבדה קודמת פחות מידע על מנת לבצע קליסטור של המידע בצורה הטובה ביותר.

כלומר המסקנה היא ככל שנגדיל את גודל הצאנק (עד חסם כלשהו) יותר DATA יכנס לכל צאנק ולכן גודל הסילואט יתקרב ל1.

בנוסף , אם מתעלמים מהרעש של הסיפוח הלא נכון של DATA לקלאסטר לא שלו

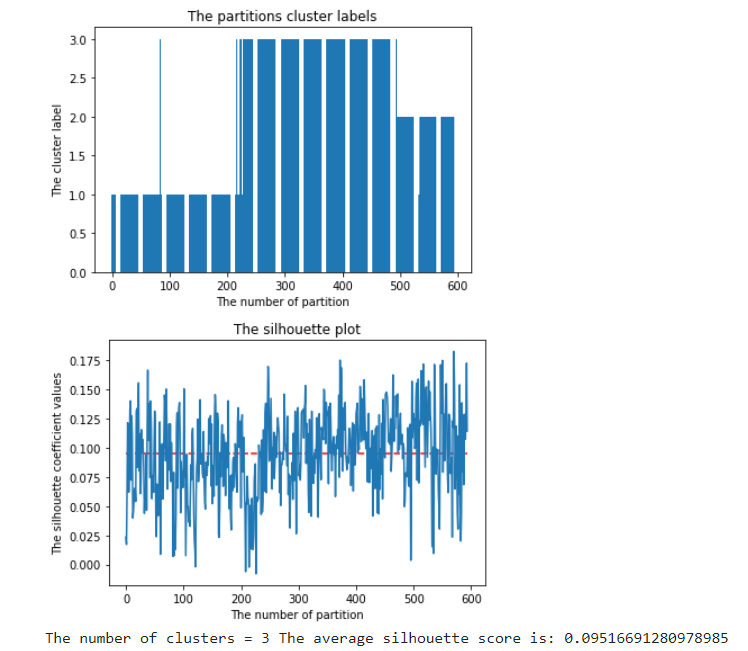
אפשר לראות שספר מתחיל מ0 עד בערך 210

מיד אחריו מ210 עד בערך 490

ולאחר מכן ספר אחרון מ490 עד בערך 590

אם נכפיל הכל ב5000 נוכל לדעת בדיוק באיזה תו במחרוזת הארוכה ספר כלשהו מתחיל ונגמר, בהמשך הדוח כתבנו פונקציה שאחראית על לגלות גבולות ולסנן רעש בהתאם.

1. Partition the text back into its original 3 different strings. **The partition point might be in a small interval** that is not greater than 5000 characters..
2. Add your results and your explanations to **lab4**\_**result.docx**.



בחלק זה ביצענו חלוקה שונה , הפעם חילקנו כל קובץ לצאנקים ולאחר מכן איחדנו הכל לסטרינג הציפיה הייתה שיהיו קיימים צאנקים מסויימים שיהיה בהם הרבה פחות תווים מ5000 משום שזה כביכול סוף ההטקסט שלהם, אפשר לראות שהסילואט גדל בממש קצת אולי בגלל החלוקה השונה של הצאנקים אך עדיין בהתאם לדברים שנאמרו בגרפים הקודמים קיימת בעיתיות נוכח גודל כל צאנק

1. Write a function that will automatically find the boundaries between the books in the text files. The function will print the boundaries between the books

**Example**:

“Book 1 begins in character 0”.

“Book 1 ends in the interval between 205000 and 210000 characters”.

“Book 2 begins in the interval between 205000 and 210000 characters”.

“Book 2 ends in the interval between 380000 and 385000 characters”.

1. **Using the found boundaries**, save each **book from the original concatenated text** (before text filtering) in different files, using the write() function.
2. Describe your function and the results.

הקבצים נמצאים בתקיה להגשה

בקוד שלנו השתמשנו בשיטה של סעיף 6

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

הסבר :

השיטה שלנו במעבדה הזאת הייתה לשים לב שאת רשימת הלייבלים קיבלנו עם רעשים , כלומר אנחנו צריכים לשים לב שבתוך למשל רצף של ספר שהוא למשל 1 סופך DATA של ספר של 2.

כדי "לנקות" את הרעש הזה בעצם יצרנו מעין מסכה , עברנו בלולאה על הערכים ברשימת הלייבלים וכל פעם שנתקלנו באות ששברה לנו את הרצף הפעלנו מיסוך (change\_val ) כלומר , בקוד זה ניקח אותיות שלא יחשבו כרעש אם הם הופיעו לפחות 4 פעמים ולכן אם נתקלנו באותיות ששברו לנו את הרצף ומופיעות פחות מ4 פעמים ברצף נמסך אותם עם -1

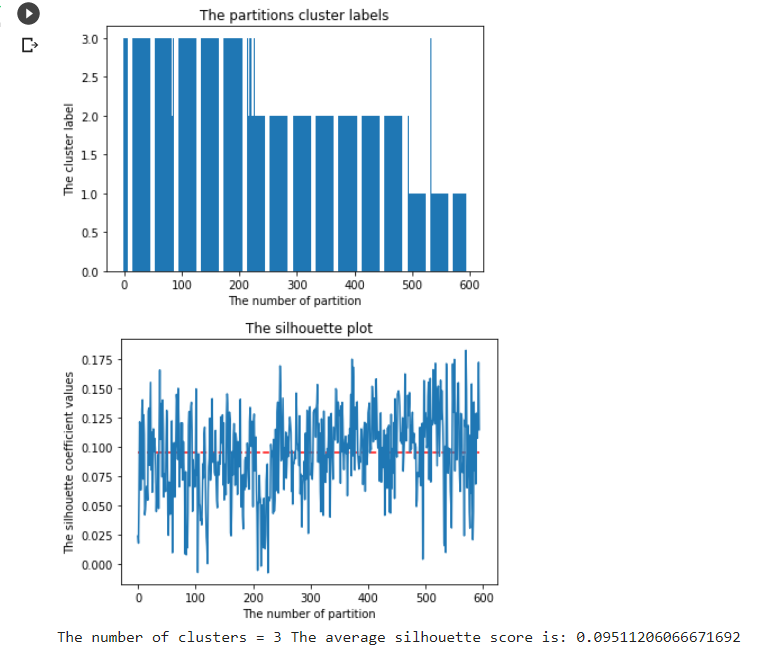
לאחר שעברנו בצורה כזאת על כל מערך הלייבלים יש לנו מערך "נקי" ללא רעשים וכעת יכלנו להתחיל לחשב באונדריז.

בשלב הבא, דאגנו ליצור STRING חדש המכיל את הקבצים שלנו ללא פילטר כלשהו 🡨 יעזור לכתיבת הקבצים.

ולבסוף בלולאה על רשימה שנקראת clean\_list (רשימה זאת מכילה את כל הערכים בלייבלס ללא כפילויות – בעצם עוזר לנו לראות מה הסדר האמיתי של הקלאסטר שחזר) באמצעות מתודת where קיבלנו את ערכי האינדקסים שבהם הספר המסוים מופיע בהם ברשימה של הלייבלים ולכן לקחנו את האינדקס האחרון והאינדקס הראשון.

הערה : המטרה לייצר מיסוך של מינוס 1 היא בגלל המתודה WHERE רצינו לגרום למתודה להתעלם מערכים שהם רעש על ידי חיפוש ערך שבוודאות לא ימצא ברשימת הclean\_list

הפלט :



תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

אם נבצע חלוקה של כל הערכים ב5000

אפשר לראות בבירור שניקינו את הרעשים , למשל ניקח את ספר מספר 3 שמסופח (מסומן באדום) לקלאסטר של ספר 1.

אך בבירור אפשר לראות כי ספר מספר 3 לפי הקוד שלנו מסתיים לכל היותר בלייבל 209 (=1045000/5000)

**הערה נוספת : בכתיבת הקבצים נתנו לכתיבה להיות מההתחלה עד לסוף האינטרבל כלומר החלון הגדול ביותר שהספר יכול להיות בו**