**שלב 1: הגדרת המחלקות:**

השתמשתנו בפונקציה get\_excel כדי שאוכל לשלוף את המידע ומופרד לפי מה שמבוקש:

* קראית את קובץ ה excel שהוא הקורפוס
* עברנו על שורות של הקובץ:
  + בחרנו בעמוד של ה sentence\_text .
  + אם הוא committee אזי מוספים לרשימה של ה committee אחרת ל plenary .
* מחזרים את שתי הרשימות.

**שלב 2: איזון חלוקה ליחידות סיווג:**

השתמשנו בפונקציה בשם make\_chunks() :

* שהיא קורא לפונקציה get\_excel ומקבלת את כל המשפטים לפי committee ו plenary :
  1. עוברים על כל המשפטים של committee :
     + לוקחים כל 5 משפטים ואז שמים אותם בתוך chunk .
  2. עוברים על כל המפשטים של plenary :
     + לוקחים כל 5 משפטים ואז שמים אותם בתוך chunk .
  3. מחזרים את שני הרשימות של chunk ים.

**שלב 3: איזון החלקות:**

השתמשנו בפונקציה בשם down\_sampling(c\_chunks, p\_chunks) :

* בודקים אם גודל הרשימה של ה chunk ים של plenary גודלה מזה של committee אם כן:
  1. מחזירים את הרשימה של ה chunk ים של committee כפי שהיא ועשים random.sample לרשימה plenary לפי הגודל של ה committee
  2. אחרת מחזירים את הרשימה של ה chunk ים של plenary כפי שהיא ועשים random.sample לרשימה committee לפי הגודל של ה plenary

מספר הפרטים בכל מחלקה:

* לפני: # ה chunk ים:
  + Plenary chunks = 14104, Committee chunks = 5895.
* אחרי # ה chunk ים:
  + Plenary chunks = 5895, Committee chunks = 5895.
* לפני # המשפטים:
  + Plenary sentences = 29476, Committee sentences = 70524.
* אחרי # המשפטים:
  + Plenary sentences = 29475, Committee sentences = 29475.

**שלב 4: יצירת וקטור מאפיינים ( feature vector ):**

1. הבחירה שלנו הייתה ב TF-IDF :
   1. נורמליזציה למספר המופעים של מילה מסוימת, זה עוזר לנו לא לתת משקל דומיננטי למילות מסוימות כדי לא לתת משקל שהוא קטן מאוד מאוד למילות אחרות ואז לא נסתכל עליהם בכלל, גם עוזר לנו אם יש לנו קורפוס גדול כדי שהגודל לא ישפיע ממש.
   2. ההבדל בין chunks כלומר הוא נותן משקל יותר למילות שמבדילות בינם בהשוואה למילות שמופיעות הרבה פעמים בכל chunk .
   3. לפי הנ"ל כך הוא מקטין את הרעש שיכול לצור מהרבה מילים.

נוכל להראות גם כך הדיוק שאנו מקבלים כאשר אנחנו משתמשים ב TF-IDF vectorizer לעומת ה CountVectorizer .

1. הבחירה במאפיינים של ה feature vector :
   1. עברנו על כל feature שיש ב corpus :
      1. חישבנו את התדירות של feature זה ב plenary וגם ב committee .
      2. הכנסנו ברשימה את ה feature וההפרש בין התדירויות שחשבנו בשלב a ולוקחים את הערך המוחלט.
      3. נעשה מיון לרשימה ואז לקחת את אלה שיש להם הפרש הכי גדול.

הסבר כללי למה בחרנו את זה:  
לפי מה שעשינו יש לנו מילים שמופעים הרבה ב plenary וממש פחות בcommittee וההפך ולכן כך עוזר לנו כאשר אנו רואים מילה מסוימת ב chunk (או כמה) נוכל לסווג טוב כך אנו אומרים אם ראינו מילה בתוך ה chunk שהיא ברשימה אזי אנחנו נוכל להגיד שברוב המקרים שה chunk יסווג כ plenary או ההפך .

בודקים רק מילים מעניינות לא כמו סמני פסוק, וכך אנחנו לקוחים מילים שהם ממש מייצגים רק את ה plenary וגם רק את committee כלומר את המילים שאנחנו בסוף לוקחים הם או מופעים הרבה ב plenary ופחות משש (כמעט 0) ב committee או ההפך ולכן יש לנו מילים שהם ממש טובים כדי שנסווג.

השתמשנו בפונקציה בשם create\_feature\_vector(co\_chunks, pl\_chunks) :

* השתמשנו ב tfidfVectorizer() ועשינו fit ל plenary and committee וכך הוא לקח את כל המילים ש features (הוא מטפיל באופן אוטומטי בסמנים כלומר לא לוקח אותם) ואז כך יש לנו ווקטור feature עבור כל chunk ( שזה שלב א מהשאלה).
* עברנו על כל feature שיש ב corpus :
  + חישבנו את התדירות של feature זה ב plenary וגם ב committee .
  + הכנסנו ברשימה את ה feature וההפרש בין התדירויות שחשבנו בשלב a ולוקחים את הערך המוחלט.
  + נעשה מיון לרשימה ואז לקחת את אלה שיש להם הפרש הכי גדול.
* השתמשנו ב tfidfVectorizer(vocabulary = list of top 2000 words of what we got) ואז עשנו fit\_transform ל chunk שיש לנו.
* חזרנו את tfidf של שלב 1, our\_tfidf של שלב 2, our\_bow של שלב 2.

**שלב 5: אימון:**

משתמשים ב train\_test(vectorBoW, length) כדי לעשות חלוקה לקבוצת אימון 90% וקבוצת טסט 10% :

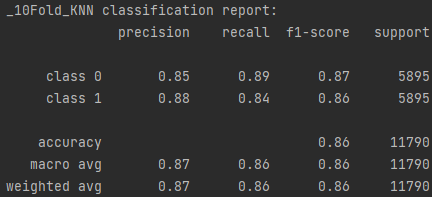
* מייצרים labels 0 עבור committee ו 1 עבור plenary ואז מחרבים אותם ביחד ואז נקבל ווקטור של labels .
* ואז מחלקים את הדאטא בפונקציה train\_test\_split ב 90% לאימון ו10% לטסט עם stratify .
* מאמנים מודל של svm ואז עושים predication ואז משתמשים ב classification\_report כדי לדעת את הדיוק וה precision, recall ..
* מאמנים מודל של KNN ואז עושים predication ואז משתמשים ב classification\_report כדי לדעת את הדיוק וה precision, recall ..
* ואז מחזירים את המודל שיש לו דיוק יותר טוב עם הדיוק שלו.

משתמשים ב k\_fold(vectorBoW, length) כדי לעשות 10 fold :

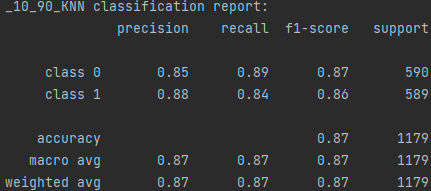
* מייצרים labels 0 עבור committee ו 1 עבור plenary ואז מחרבים אותם ביחד ואז נקבל ווקטור של labels .
* עשים StratifiedKFold כדי שנוכל להעביר ל cross\_val\_predict כדי שה labels יהיו מחולקים באותו אחוז כפי שעשינו באימון.
* ואז עושים cross\_val\_predict עם הווקטור שאנו מקבלים ועם ה labels וה StratifiedKFold ועם המודל knn .
* עושים cross\_val\_predict עם הווקטור שאנו מקבלים ועם ה labels וה StratifiedKFold ועם המודל svm .

**הערה חשובה:** class 0 = committee, class 1 = plenary .

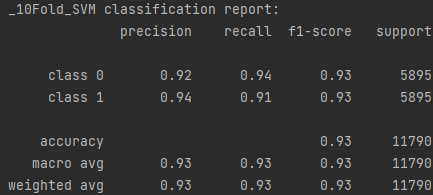
* כל BoW הוא וקטור מאפיינים:
  + KNN :
    - 10-fold Cross Valdidation :



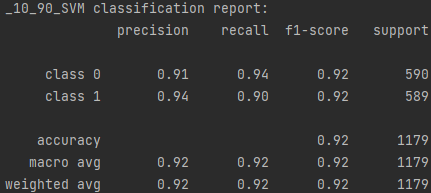
* + - חלוקה לקבוצת אימון 90% וקבוצת טסט 10%:



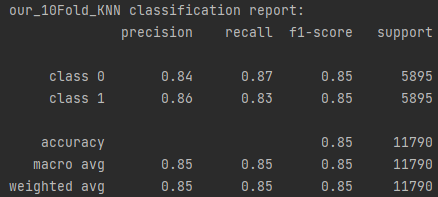
* + SVM :
    - 10-fold Cross Valdidation :



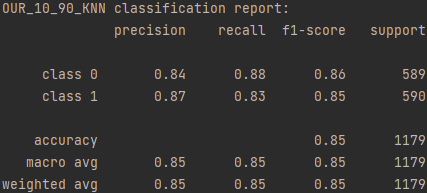
* + - חלוקה לקבוצת אימון 90% וקבוצת טסט 10%:



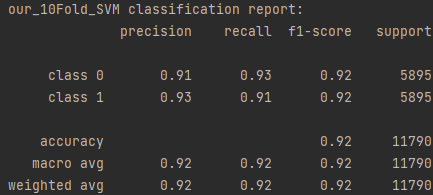
* וקטור שלנו:
  + KNN :
    - 10-fold Cross Valdidation :



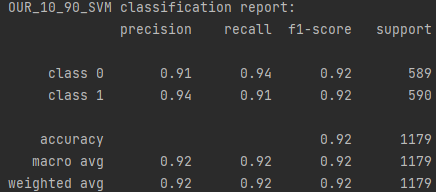
* + - חלוקה לקבוצת אימון 90% וקבוצת טסט 10%:



* + SVM :
    - 10-fold Cross Valdidation :



* + - חלוקה לקבוצת אימון 90% וקבוצת טסט 10%:



**שלב 6: סיווג:**

אנחנו משתמשים ב classification(classify\_model, classify\_tfidf) כדי לסווג את הקובץ החדש שאנו מקבלים:

* עוברים על הקובץ כל שורה היא chunk מכנסים את כולם לרשימה אחד.
* מפעילים transform על הרשימה.
* ואז עשים predict לפי המודל שאנו מקבלים ואז לפי ה labels אם זה 0 אזי מכנסים לקובץ חדש שורה שזה committee אחרת plenary .

**שאלות:**

1. האם היו הבדלים ב-precision ו-recall בין המחלקות? אם כן, מה ניתן להסיק מהם ?
   * כן ניתן להסיק:
     1. אנחנו רואים שב precision שב committee הוא נמוך מזה של ה plenary ולכן נוכל להגיד שאנו מסווגים ב committee יותר דברים שהם plenary כ committee ולכן כך יעלה את ה precision של ה plenary כי אנו יודעים שאם לא מסווגים שום דבר כ plenary אזי ה precision מושלם ולכן נוכל להסיק שהקבוצה של ה plenary קטנה ולכן הprecision עלה אזי נוכל להסיק שככל שב committee מסווגים דברים שהם Plenary כ committee אזי ה precision ב committee ירד וה precision של הplenary יעלה.
     2. אנו רואים שה recall ב committee יותר גדול מזה של ה plenary כלומר אנו מסווגים את committee מאלה שהם באמת committee יותר טוב מה plenary וזה אומר שאנחנו מצליחים לסווג צורה טובה את committee .
     3. כפי שלמדנו בהרצאה שיש tradeoff בין precision ו recall ולכן ב committee ה precision הוא קטן מזה של ה plenary אבל ה recall הוא גדול מזה של plenary .
2. האם תוצאות הסיווג בחלוקת אימון-בדיקה של 90%-10% דומות לתוצאות ה-validation cross ?בין אם כן ובין אם לאו, נסו לשער מדוע.
   * כן, יכולים להגיד שהמודל שלנו הוא מכליל בצורה טובה כי אנחנו לפי ה cross\_val אנחנו עושים 10 folds שזה כל פעם חלוקה שונה ובכל זאת מקבלים כמעט אותו דיוק של train\_test אזי זה קרה משום שאחרי שעשינו אימון וטסט למודל שלנו הוא ממש מכליל לא עושה overfitting .
3. הסבירו מהם היתרונות והחסרונות של שני סוגי המסווגים SVM ,KNN בהם השתמשתם. האם לדעתכם אחד מהם עדיף על פני השני, עבור משימת הסיווג שבתרגיל?
   * יתרונות של svm :
     1. הוא פחות נוטה ל overfitting ויכול להכליל בצורה טובה.
     2. אחרי שאנחנו מאמין את המודל אנחנו יכולים לזרוק את המידע ולכן זה יעיל מבחינת space complexity .
     3. הוא עובד טוב כאשר אנחנו בממדים גבוהים.
   * חסרונות של svm :
     1. הוא לוקח הרבה זמן בשלב האימון של המודל ובמיוחד כאשר יש לנו דטאט סט שהוא ממש גדול.
     2. כאשר יש לנו מלא outliers אזי זה יכול להשפיע בצורה לא טובה על המודל.
   * יתרונות של knn :
     1. אין זמן אימון במודל הזה.
     2. קל ליישום.
     3. אנחנו יכולים להוסיף מידע חדש למודל שלנו בכל זמן וזה לא משפיע על המודל.
   * חסרונות של knn :
     1. רגיש לרעש אם בחרנו k לא טוב אז אנחנו נקבל תוצאות שממש גרועות.
     2. יכול להוביל ל overfitting אם בחרנו k לא טוב.
     3. שומר כל הדאטא כדי שיכול לסווג את המידע החדש שהוא מקבל.
     4. צריכים להחליט על איך מטפלים בתיקו של סווג מסוים.
   * עדיף להשתמש ב SVM :
     1. במודל של svm הוא פחות נוטה ל overfitting ויכול להכליל בצורה טובה יותר לעומת knn שהוא תלוי ב k וזה יכול להוביל ל overfitting .
     2. במודל של svm הוא טוב גם אם יש רעש ומילים שהם לא מעניינות לעומת שבמודל knn אנחנו צריכים לבדוק ולוודא שה k שלנו הוא טוב.
4. פרטו את היתרונות והחסרונות ליצירת יחידות הסיווג. מה יהיו ההשלכות אם נגדיל ואם נקטין אותן באופן משמעותי?
   * היתרונות של יחידות הסיווג:
     1. יכול להכיל מספיק תכונות שיהוו מאפיינים משמעותיים למסווג.
     2. כאשר יש לנו יותר ממשפט אחד בתוך יחיד הסיווג אזי זה יגדיל את הסיכויים לראות את מילה מסוימת יותר באותו יחידת סווג וזה יעזור לסווג נכונה ומדויקת יותר.
     3. כאשר יש לנו יחידת סווג שהיא מכילה יותר ממשפט אחד אזי זה יקטן את הרעש ואת המשקל של מילים שהם לא רלוונטיים.
   * החסרונות של יחידות הסיווג:
     1. כאשר אנו משתמשים במשפט ספציפי אזי נוכל לסווג אותו לפי מילה מיוחדת שנמצאת בה אבל כאשר היא בתוך יחידת הסווג זה יכול להוביל שהמילה הזו לא תישאר מיוחדת וניתן לה משלק קטן ממש.
     2. אנחנו צריכים למוצא איזה שהיא שיטה למצוא את גודל יחידת הסיווג הכי מתאימה לבעיה שלנו (כלומר לדאטא שלו) כדי שנוכל לסווג בצורה טובה.
   * אם נגדיל :
     1. ה chunk יכיל הרבה מילים והרבה רעש שזה פוגע במילים הרלוונטיים יותר.
     2. ככל שה chunk יותר גדול אזי הוא יהי ממש מתאים עבור סווג מסיום ואז יהי לנו overfitting .
   * אם נקטין:
     1. ה chunk יושפע באופן משמעותי מהרעש כלומר מהמילים שהם לא רלוונטיים ואז המילים המיוחדים לא יקבלו משקל .
     2. מקשה על המסווג כך גם features ממש יהיו דלילים ולא נוכל לסווג בצורה טובה.