**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 20 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.

**שאלות**

1. **מה הם האתגרים שיכולים להיווצר בשימוש במחלקה "אחר" במשימת הסיווג?**

בניגוד למחלקות של שני הדוברים, המחלקה "אחר" מגוונת ולא ייחודית לאדם מסוים, ולכן יכול להיות שהמודל יתקשה במציאת דפוסים מסוימים בטקסטים של המחלקה או מאפיינים שיהוו בסיס לסיווג. כמו כן, ובשל היותה חסרת דפוס יחסית, משפטים הצהרתיים (Declarative sentences) או משפטים שלא תואמים לאופי של שני הדוברים עשויים לרוב להיסוג כ- "אחר".

1. **נניח שאתם משתתפים בתחרות מודלים לחיזוי בינארי שבה אם המודל שלכם יחזה נכון את כל הדוגמאות של הדובר הראשון, תקבלו פרס כספי גדול, ואם המודל שלכם יטעה על אפילו דוגמה אחת של הדובר הראשון תקבלו קנס כספי גבוה. מבין המדדים המופיעים ב classification report , איזה מדד תרצו למקסם? איזה מהמודלים שאימנתם תבחרו למטרה זו? הסבירו.**

לצורך התחרות, נבחר למקסם את המדד של Recall אשר מודד את כמות החיזויים הנכונים שלו לקטגוריה מבין כל הדוגמאות של הקטגוריה, בשאיפה להשגת Recall = 1 אשר יבטיח אפס טעויות. לגבי המודל, נבחר בזה שהביא ל- Recall הגבוה ביותר עבור הדובר הראשון. במקרה שלנו, זה היה Logistic Regression כאשר השתמשנו בווקטור Tf-idf, ו- KNN בהינתן ווקטור המאפיינים שיצרנו.

1. **ענו שוב על 1 כאשר שינו את החוקים בתחרות וכעת אם המודל שלכם יסווג נכון את כל הדוגמאות של שני הדוברים תקבלו פרס כספי גבוה, אבל אם המודל שלכם יסווג אפילו דוגמה אחת בצורה לא נכונה, תקבלו קנס כספי גבוה.**

במקרה הזה נרצה למקסם את ה- Accuracy, אשר מודד את אחוז הדוגמאות שסווגו נכון מבין כל הדוגמאות, בשאיפה להשגת 100%. באשר למודל, נבחר את המודל בעל ה- Accuracy הגבוה מבין השניים, ובמקרה של תוצאות Accuracy דומות, נעדיף את המודל שנותן איזון טוב יותר בין ה- Recall וה- Precision. בהסתמך על התוצאות שלנו, גם כאן, הבחירה במודל תלויה בווקטור המאפיינים הנבחר, כאשר בשימוש בווקטור Tf-idf מסווג ה- Logistic Regression השיג דיוק טוב יותר, ואילו בשימוש בווקטור המאפיינים שיצרנו התוצאות היו לטובת KNN.

1. **הסבירו מה היתרונות והחסרונות של שיטת ה cross validation על פני חלוקה פשוטה למחלקת אימון ובדיקה. איזו משיטות ההערכה אמינה יותר לדעתכם?**

היתרונות של ה- Cross Validation על פני החלוקה למחלקות אימון ובדיקה: נוטה להיות יותר יציבה ומדויקת בגלל שהיא ממצעת כמה חלוקות שונות, במקום חלוקה אחת שיכולה להיות מוטה, במיוחד כאשר כמות הנתונים מעטה ומכילה גיוון רב. משתמשת במספר גדול יותר של נתונים להערכה מפני שהיא משתמשת בכל הדאטה לאימון ובדיקה, בניגוד לשיטה האחרת אשר בה כל נתון בדאטה יכול לשמש או לאימון או לבדיקה.

החסרונות של ה- Cross Validation לעומת החלוקה למחלקות אימון ובדיקה: סיבוכיות חישוב גבוהה יותר בשל הצורך לבצע חלוקה, אימון והערכה מספר פעמים לעומת פעם אחת בשיטה האחרת. החיסרון הזה משמעותי יותר כאשר יש לנו נפח דאטה גדול המצריך זמן רב של עיבוד.

לדעתנו, ה- Cross Validation אמינה יותר, בשל הסיבות שהוזכרו כגון יציבות ושימוש במספר גדול יותר של נתונים להערכה אימון, כמו גם בממוצע של הערכות אשר אמור להפחית את האפקט של הרעש.

1. **הסבירו מהם היתרונות והחסרונות של שני סוגי המסווגים KNN, LogisticRegression בהם השתמשתם. האם לדעתכם אחד מהם עדיף על פני השני, עבור משימות הסיווג שבתרגיל?**

היתרונות של מודל KNN:

* **הרעיון שלו פשוט ואינטואיטיבי** ומתבסס על חיפוש השכנים הקרובים ביותר, ולכן קל ליישום.
* **יכול להתמודד עם נתונים שהם לא לינאריים**.
* **תומך במאפיינים שהם בינאריים**, ויכול לשלב מאפיינים רציפים ובינאריים, דבר שהשתמשנו בו במטלה שלנו, כאשר כללנו את המאפיין הרציף אורך המשפט, ואת המאפיין הבינארי של אם המשפט מכיל ספרה או לא.

החסרונות של מודל KNN:

* **דורש זמן חישוב גבוה יחסית**.
* **קושי בלבחור את מספר השכנים המתאים** (את ה- k) והערכת הטווח של הערכים אשר לא יביאו ל- Underfitting או Overfitting. דבר שחווינו אותו בעת בחירת ה- k במטלה שלנו, ולמרות שערכי k יותר גבוהים מזה שבחרנו נתנו תוצאות טובות יותר בחרנו לא להסתכן ולהתרחק יותר מדי מערך ברירת המחדל.
* **רגיש לרעש**, ומספיק ששכן אחד יהיה מוטה, יכול להשפיע לרעה על התוצאה של הסיווג, במיוחד כאשר מספר קטן של שכנים נלקח בחשבון.
* למרות שהשילוב של מאפיינים רציפים ובינאריים הוזכר כיתרון, אולם בלי שימוש בנורמליזציה**, מאפיינים בעלי טווח ערכים גדולים יכולים להיות בעלי השפעה גדולה יותר ממאפיינים בינאריים, או כאלה הרציפים עם טווח ערכים קטן**.

היתרונות של מודל LogisticRegression:

* **זמן החישוב שלו מהיר** יותר בהשוואה ל- KNN.
* **פחות רגיש לרעשים** בהשוואה ל- KNN.

החסרונות של מודל LogisticRegression:

* **מניח קשר לינארי בין הנתונים**, ומתקשה להתמודד עם דאטה שאינה ניתנת להפרדה לינארית.
* **מתקשה להתמודד עם דאטה קטנה**.
* **מתקשה להתמודד עם דאטה לא מאוזנת**, ואם לא ננקטו צעדים של איזון, הסיווג שלו עלול להיות מוטה לטובת המחלקות בעלות מספר דוגמאות הגדול יותר.

למרות שהתוצאות של הערכת שניהם הניבו תוצאות קרובות, נעדיף להשתמש ב- **Logistic Regression** על פני KNN בשל יעילותו החישובית בהינתן כמות הדאטה הלא קטנה שיש לנו, בנוסף להיותו פחות רגיש לרעש, דבר שהוא אופייני במשימת סיווג משפטים.

1. **יחידת הסיווג בתרגיל היא משפט אחד. אם במקום זאת, היינו מחליטים על יחידת סיווג שמאחדת יחד מספר משפטים מאותה מחלקה, כך שכל דוגמה לסיווג הייתה מקבץ של משפטים. מה היו היתרונות והחסרונות בכך? התייחסו בתשובתכם ליחידות סיווג של 2, 5, 10, 100 משפטים.**

היתרונות של יחידת סיווג המורכבת ממספר משפטים:

* **הקשר רחב יותר שיפחית את את מידת השפעת המשפטים הרועשים**.
* **הכללת מספר רב יותר של מאפיינים בתוך היחידה, מה שהופך אותה ליציבה ומייצגת יותר**. למשל המספר הגדול ביותר של מאפיינים בווקטור המאפיינים שיצרנו מתבססים על מספר ההופעות של קולקציה מסוימת במשפט, וסביר להניח שרוב המאפיינים עבור יחידה בגודל משפט אחד יקבלו ערך 0 ושנדיר שאחת תקבל יותר מ- 2. לעומת זאת בשימוש ביחידה שמורכבת ממספר משפטים יהיו יותר מאפיינים שהם לא 0, בנוסף לכך שיהיה גיוון רב יותר בערכים, כמו גם יהיה קל יותר לזהות קולקציה המאפיינת דובר מסוים. בדאטה שלנו זה יכול להיות אפקטיבי מפני שהרבה מהמשפטים הם בגודל 4-6 מלים שעלולים לא להכיל מספיק מידע או ייצוג של הרבה מאפיינים.

החסרונות של יחידת סיווג המורכבת ממספר משפטים:

* **קושי בסיווג משפטים בודדים**, מפני שיהיה שסקאלת המרחק תגדל בין המשפט ליחידות סיווג, בדרך שתקשה להבדיל בין מרחק קטן לגדול.
* **חוסר עקבות בסיווג**, כאשר הסיווג עלול להיות מושפע מהאופן בו איחדנו את המשפטים.
* **הפחתת מספר הדוגמאות לסיווג**, זהו חסרון במקרה שגודל הדאטה קטן או כאשר המסווג מתמודד יותר טוב כאשר הדאטה גדולה כמו במקרה של Logistic Regression.
* **משקל פחות למאפיינים יחידים שהופעתם יכולה להכריע את הסיווג**, בשל כך שמספר רב של מאפיינים אמורים להופיע בו זמנית כאשר היחידה מורכבת ממספר משפטים, אם לא נלקחו צעדים שאמורים למנוע זאת. למשל, במקרה של המאפיין "הופעת ספרה ביחידת הסיווג", שהופך להיות נכון ברוב המקרים. צעד שאפשר לקחת על מנת למנוע זאת, הוא החלפת המאפיין ב- "מספר הספרות ביחידת הסיווג".

השימוש ביחידת סיווג בגודל 2 יכולה להוסיף זמן חישוב בשל הזמן הנוסף שהאיחוד יצטרך, בלי להוסיף יתרון משמעות, כך שהייצוג של המאפיינים והפחתת הרעש לא הולכים להשתפר בצורה גדולה.

השימוש ביחידת סיווג בגודל 100 אולם אמור להפחית רעש באופן משמעותי, כמו גם להכיל מידע גדול יותר לגבי המאפיינים, כמו כן ובשל כך שהרבה פעמים יהיה קשה למצוא 100 משפטים של אותו דובר באותו הקשר, נצטרך לשלב בין משפטים מהקשרים שונים, למשל, במטלה שלנו, משפטים של הדובר מפרוטוקולים שונים. הדבר יכול להפחית את האפקט של המאפיינים התלויי הקשר לטובת המאפיינים תלויי אישיות. מצד שני, יחידת סיווג גדולה כזאת יכולה להכיל מספר רב של מאפיינים ולמנוע מהמסווג לזהות את הקשר בין המאפיינים ואיזה מהמאפיינים סבירים להופיע עם מאפיינים אחרים. למשל, כאשר דובר ממלא תפקיד יו"ר נצפה שהמאפיינים של הופעת הקולקציה אני קורא לך לסדר", ו- "אורך קצר של משפט" יופיעו ביחד רבות במשפטים, לעומת כאשר הוא נמצא בתור חברת הכנסת שהמאפיינים של "הופעת ספרה במשפט" ו- "הופעת הקולקציה אני" ו- "אורך גדול של משפט" יותר מתלוות. ואנחנו לא נרצה יחידה שמשלבת בין כל המאפיינים. חסרון נוסף, שהוזכר למעלה, הוא בזה שמאפיינים מסוימים שנותנים תרומה רבה לסיווג נכון, ובשל ריבוי המאפיינים, הולכים להיות בעלי משקל נמוך יותר.

השימוש ביחידות סיווג בגדלים של 5 או 10, לדעתנו, יכול להיות הכי אידאלי, בשל היותו מפחית רעש ומכיל מידע גדול באופן משמעותי, אשר מפצים על זמן החישוב הנוסף, כמו גם, שמספר כזה של משפטים יכולים להיות בהקשר אחד או לכל היותר שני הקשרים כך שהחסרונות הנובעות מאיחוד משפטים מהקשרים שונים שהוזכרו למעלה קורים לעיתים רחוקות.

1. **איזה גודל של יחידת סיווג עדיף לדעתכם (1 , 2 , 5 , 10 , 100) במשימות שלנו? הסבירו.**

נעדיף להשתמש ביחידות סיווג בגדלים של 5 או 10, בשל הסיבות שנזכרו בשאלה הקודמת, שהגודל הזה מפחית רעש הנוצר ממשפטים חריגים, מוסיף ייצוג יותר של מאפיינים ביחידה, מונע באופן גדול את הסבירות של איחוד בין משפטים שנלקחים מהקשרים שונים.