Mohammad Arabi 206985533

Saji Assi 314831207

דוו"ח

תרגיל בית 3

שלב 1+2:

בשביל שלב 1 פשוט השתמשנו בשדה protocol type כדי להגדיר המחלקה של כל משפט. ובשביל שלב 2 גם פשוט חלקנו קורפוס המשפטים למשפטי וועדה ומשפטי מליאה אחרי זה פשוט עבור כל סוג משפטים נחלק כל 5 משפטים ביחד.

שלב 3:

אחרי שחלקנו המשפטים ל-chunks כל סוג בנפרד פשוט בדקנו מי מבין שני הסוגים יש לו יותר chunks ואז השתמשנו בפקודה random.sample שמתוך הסוג שיש לו יותר chunks תבחור באופן ייחודי ורנדומי מספר chunks שיהיה שווה למספר ה-chunks שיש בסוג שיש לו פחות chunks. ואז התוצאות שקבלנו :

Number of committee chunks before the balance : 8596

Number of plenary chunks before the balance : 13102

Number of committee chunks after the balance : 8596

Number of plenary chunks after the balance : 8596

שלב 4:

בשלב הזה בחרנו ב-TfidfVectorizer כי אחרי הבדיקה יצא לנו שהוא יותר טוב מ-CountVectorizer כי הוא ייתן לנו יותר דיוק כי השיטה הזאת לא רק מתחשבת במספר הפעמים שמונח מופע כמו

ש-CountVectorizer עושה אלא גם בתדירות של המונחים הדבר שמשפר הדיוק בתוצאות.

בשביל בחירת וקטור מאפיינים שלנו כתבנו הפונקציה def choose_MyFeatures(chunks)

כך כמו רואים מקבלת רשימה של chunks ומחזירה וקטור מאפיינים כך הפיצ'רים שבחרנו הם :

ממוצע גודל המשפט, רשימה של מילות תוכן.

לגבי רשמית מילות תוכן בחרנו במילים שיהיו נפוצות בסוג מסוים וקשורות לסוג מסוים וגם לקחנו מהתוצאות של הקולוקציות שעשינו בתרגיל 2.

שלב 5:

בשביל הערכת הדיוק בעזרת 5-fold Cross Validation הגדרנו הפונקציה

```
def k_fold_cross_validation(model, X, y, cv=5):
```

שמקבלת המודל והדאתא והליבלים ו-cv שברירת המחדל שלו 5 שזה גם מה רוצים במקרה שלנו, בשביל החישובים הפונקציה משתמשת ב-cross_val_predict ובשביל ה-report השתמשנו

ב-classification_report. ולגבי ההערכה train_test_split כתבנו הפונקציה

```
def fun_eval_train_test_split(model, X_train, y_train, X_test, y_test):
```

שאחרי שנעשה split קוראים לה.

וזה classification_report שקבלנו :

```
KNN tfidf 5-fold_cross_validation Classification Report:
              precision    recall  f1-score   support

   committee       0.80      0.70      0.74      8596
     plenary       0.73      0.83      0.78      8596

    accuracy                           0.76     17192
   macro avg       0.77      0.76      0.76     17192
weighted avg       0.77      0.76      0.76     17192

Logistic Regression tfidf 5-fold_cross_validation Classification Report:
              precision    recall  f1-score   support

   committee       0.85      0.82      0.84      8596
     plenary       0.83      0.86      0.84      8596

    accuracy                           0.84     17192
   macro avg       0.84      0.84      0.84     17192
weighted avg       0.84      0.84      0.84     17192

KNN My_Features 5-fold_cross_validation Classification Report:
              precision    recall  f1-score   support

   committee       0.64      0.84      0.73      8596
     plenary       0.77      0.53      0.63      8596

    accuracy                           0.69     17192
   macro avg       0.71      0.69      0.68     17192
weighted avg       0.71      0.69      0.68     17192

Logistic Regression My_Features 5-fold_cross_validation Classification Report:
              precision    recall  f1-score   support

   committee       0.68      0.86      0.76      8596
     plenary       0.81      0.59      0.68      8596

    accuracy                           0.72     17192
   macro avg       0.74      0.72      0.72     17192
weighted avg       0.74      0.72      0.72     17192

KNN tfidf Train_Test_Split Classification Report:
              precision    recall  f1-score   support

   committee       0.83      0.91      0.87       860
     plenary       0.90      0.82      0.86       860

    accuracy                           0.86      1720
   macro avg       0.87      0.86      0.86      1720
weighted avg       0.87      0.86      0.86      1720

Logistic Regression tfidf Train_Test_Split Classification Report:
              precision    recall  f1-score   support

   committee       0.86      0.92      0.89       860
     plenary       0.91      0.85      0.88       860

    accuracy                           0.89      1720
   macro avg       0.89      0.89      0.89      1720
weighted avg       0.89      0.89      0.89      1720

KNN My_Features Train_Test_Split Classification Report:
              precision    recall  f1-score   support

   committee       0.64      0.75      0.69       860
     plenary       0.70      0.58      0.64       860

    accuracy                           0.67      1720
   macro avg       0.67      0.67      0.66      1720
weighted avg       0.67      0.67      0.66      1720

Logistic Regression My_Features Train_Test_Split Classification Report:
              precision    recall  f1-score   support

   committee       0.67      0.85      0.75       860
     plenary       0.80      0.57      0.66       860

    accuracy                           0.71      1720
   macro avg       0.73      0.71      0.71      1720
weighted avg       0.73      0.71      0.71      1720
```

שלב 6:

אחרי חישוב הדיוק עבור כל המקרים בדקנו מי המודל שנותן הדיוק הכי טוב והשתמשנו במודל הזה בשביל סיווג ה-chunks שבקובץ.

שלב 7:

1) מהתוצאות המתקבלות, נראה שגודלchunk = 20 הוא האידיאלי עבור משימת הסיווג קיבלנו דיוק של 0.95ב KNN. גם KNN וגם Logistic Regression מראים ביצועים גבוהים ביותר עם TF-IDF ועם מאפיינים מותאמים אישית בגודל chunk זה.
2) יתרונות וחסרונות ל CHUNKS בגדלים שונים היא שבגדלים קטנים למשל 1 או 3 יהיה לנו יותר CHUNKS לאימון, מה שיכול להגדיל את מגוון הדוגמאות ולשפר את היכולת של המודל להכליל. והחסרנות היא שמקבלים דיוק נמוך למשל שבדקנו על גדלים כמו 1 ו 3 הדיוק היה מאוד נמוך וזה כמובן בגלל חוסר במידע שמספקים.
מצד שני CHUNKS יותר גדולים למשל 10 או 20 מקבלים דיוק מאוד גדול אבל זה יכול לגרום לפחות הכללה זאת אומרת Overfitting.

שאלות:

1) במקרה זה, נרצה למקסם את מדד (Recall) עבור קטגוריית ה-committee.כי ה RECALL מודד את יכולת המודל לזהות את כל הדוגמאות הנכונות לכן לפי התוצאות נעדיף את מודל Logistic Regression עם TF-IDF בגודל Chunk = 20 שכן הוא נותן דיוק גבוה.

2) במקרה זה, נרצה למקסם את מדד הדיוק (Accuracy) לכן על פי התוצאות המתקבלות, נעדיף את מודל KNN עם TF-IDF בגודל Chunk = 20, שכן הוא נותן את הדיוק הכולל הגבוה ביותר.
חשוב לנו שהמודל יהיה בעל דיוק כולל גבוה ככל האפשר.

3) תוצאות ה-cross-validation והחלוקה לסט אימון וסט בדיקה קרובות אחת לשנייה אבל ה--cross validation אמינה יותר. ה-cross-validation מבצע הערכה על מספר חלקי נתונים שונים, מה שמאפשר לתת תמונה מקיפה יותר על ביצועי המודל על הנתונים. לעומת זאת, חלוקה לסט אימון וסט בדיקה תלויה בחלוקה אקראית אחת, מה שגורם לתוצאות להיות מושפעות מאקראיות זו.

4)
Logistic Regression עשוי להיות עדיף מכיוון שהוא מניב תוצאות טובות יותר עבור בעיות סיווג ונותן דיוק יותר גבוהה לכן הוא עדיף על KNN(בChunk עד 20).
לגבי יתרונות וחסרונות אני אסביר בכללי (זה מה שהבנתי מהשאלה):
יתרונות וחסרונות KNN היא שהוא מודל קל ולא עושה הנחות התפלגות על הנתונים אבל מצד שני הוא צריך בחירת K אופטימלי ומאוד מושפע מרעש מסביב.
יתרונות וחסרונות Logistic Regression:
בניגוד ל KNN פחות רגיש לרעש ונותן ביצועים טובים במיוחד כאשר יש קשר לינארי בין התכונות לתוצאה אבל מצד שני מניח קשר לינארי בין התכונות לתוצאה.