NLP – HW3

מגישים:

מוחמד חג'אזי – 213011026

מועאד עבד אללטיף –209418896

**קובץ ה csv-שעליו הסתמכנו הוא זה שאנחנו יצרנו בתרגיל בית 1 – שקוראים לו 'output_data.csv'.

**אופן הקלט של התוכנית שלנו הוא ששני הקבצים הנדרשים- שהם קובץ הקורפוס וקובץ הטקסט שמכיל שאנקס(שצורף עם התרגיל), יקרא אותם לפי השם שלהם 'output_data.csv' ו- 'knesset_text_chunks.txt' שהם נמצאים באותה תיקיה שבה קובץ הפיתון נמצא. זאת אומרת, לפני שמריצים את התוכנית, צריך להוסיף את שני הקבצים הנ"ל לתיקיה שבה נמצא הקוד כדי שהכל יעבוד כמו שצריך.

שלב 1:הגדרת המחלקות

באמצעות העמודה המציינת את סוג הפרוטוקול בCSV שיצרנו קודם קבענו את המחלקה של כל משפט: 0 למשפטים מוועדה ו-1 למשפטים ממליאה.

שלב 2:חלוקה ליחידות סיווג

חלקנו את הטקסט ל chunks של חמישה משפטים כל אחד. כל chunk נוצר ממשפטים השייכים לאותה מחלקה ומשמש כיחידת סיווג בניתוח שלנו.

במידה ומספר המשפטים במחלקה אינו מתחלק בחמישה, ויתרנו על שארית המשפטים.

דאגנו שכל מקבץ יכיל משפטים מאותה מחלקה בלבד.

הפונקציה שבנינו היא בונה chunks בגודל chunk_len שהעברנו אותו שקראנו לפונקציה(כדי לנשות את שאלת הבונוס):

```
def create_chunks(sentences, chunk_len):
    chunks = []
    sentences_list = sentences.split('\n')
    for i in range(0, len(sentences), chunk_len):
        chunk = sentences_list[i:i+chunk_len]
        if len(chunk) == chunk_len:
            chunks.append(' '.join(chunk))
    return chunks
```

שלב 3:איזון המחלקות

לפני האיזון, מספר השאנקים בכל מחלקה היה:

committee_chunks 7834

plenary_chunks15311

לאחר האיזון:

```
def down_sampling(committee_chunks, plenary_chunks):
    if len(committee_chunks) > len(plenary_chunks):
        committee_chunks = random.sample(committee_chunks,
len(plenary_chunks))
    else:
        plenary_chunks = random.sample(plenary_chunks, len(committee_chunks))

    return committee_chunks, plenary_chunks
```

הגענו למצב שבו שתי המחלקות היו עם מספר שאנקים שווה (7834).

שלב 4:יצירת וקטור מאפיינים

4.1: שימוש ב-Bag of Words

בחרנו להשתמש בשיטת ה-Bag of Words (BoW) דרך ה-TfidfVectorizer .

החלטנו על שימוש ב-TfidfVectorizer ולא ב-CountVectorizer מכיוון שהוא לא רק סופר את מספר ההופעות של כל מילה בתוך ה-chunk, אלא גם נותן משקל למילים לפי חשיבותן ותדירותן בקורפוס כולו.

גם אחרי שנסינו את שניהם TfidfVectorizer שיפר את התוצאות, במיוחד עבור KNN .

4.2: יצירת וקטור מאפיינים מותאם אישית

וקטור המאפיינים שלנו כלל מאפייני סגנון ותוכן המבוססים על הנתונים הקיימים בקורפוס. מאפיינים אלו כללו את אורך המשפט הממוצע בכל מקבץ, ספירת מילות מפתח שנבחרו מראש והיוו מדד להבדלים בין ועדות למליאות, וכן את מספר המילים הכולל בכל chunk. ניתוח הדאטה הצביע על מספר מילות מפתח וביטויים המאפיינים באופן חזק את ההבדלים בין פרוטוקולי הועדה לפרוטוקולי המליאה, מה שהוביל אותנו לכלול אותם כמאפיינים בווקטור המאפיינים שלנו.

מילות מפתח שנבחרו כללו את 10 הגrams2 הכי נפוצים במדד PMI,וגם 3 הגrams2 הכי נפוצים בתדירות.

לא השתמשנו בעמודות נוספות מהדאטה כי כשעוים test לא נוכל להשתמש במאפיינים אלה.

גם חשבנו את המופעים של הפסיק "," כמאפיין.

שלב 5:אימון

שימוש ב-Bag of Words:

KNN 10-fold CV Accuracy: 0.85

SVM 10-fold CV Accuracy: 0.89

KNN Stratified Train-Test Split Evaluation:

	precision	recall	f1-score	support
0	0.82	0.87	0.84	784
1	0.86	0.81	0.83	783
accuracy			0.84	1567
macro avg	0.84	0.84	0.84	1567
weighted avg	0.84	0.84	0.84	1567

SVM Stratified Train-Test Split Evaluation:

	precision	recall	f1-score	support
0	0.87	0.89	0.88	784
1	0.88	0.87	0.88	783
accuracy			0.88	1567
macro avg	0.88	0.88	0.88	1567
weighted avg	0.88	0.88	0.88	1567

שימוש בווקטור המאפיינים שלנו:

KNN 10-fold CV Accuracy with Our Features: 0.68

SVM 10-fold CV Accuracy with Our Features: 0.71

KNN Stratified Train-Test Split Evaluation with **Our Features**:

	precision	recall	f1-score	support
0	0.69	0.70	0.70	784
1	0.70	0.68	0.69	783
accuracy			0.69	1567
macro avg	0.69	0.69	0.69	1567
weighted avg	0.69	0.69	0.69	1567

SVM Stratified Train-Test Split Evaluation with **Our Features**:

	precision	recall	f1-score	support
0	0.67	0.81	0.73	784
1	0.76	0.60	0.67	783
accuracy			0.71	1567
macro avg	0.72	0.71	0.70	1567
weighted avg	0.72	0.71	0.70	1567

שלב 6:סיווג

עשינו את הסיווג במודל SVM עם BoW, כי הוא המודל שעבורו קבלנו את התוצאות הטובות ביותר.

שאלות

1.האם היו הבדלים בpercision -וrecall -בין המחלקות? אם כן, מה ניתן להסיק מהם?

תזכורת: מחלקה 0 = וועדה, מחלקה 1 = מליאה.

כן, יש הבדלים ב-precision ו-recall בין המחלקות בשני המודלים השונים.

ב-KNN:

1. Precision:

- עבור מחלקה 0: precision = 0.69
- עבור מחלקה 1: precision = 0.70

מה שמציין שיש קצת יותר דיוק בזיהוי המחלקה 1 ביחס למחלקה 0 .

2. Recall:

- עבור מחלקה 0 : recall = 0.70
- עבור מחלקה 1 : recall = 0.68

מה שמציין שיש יתרון קל בזיהוי הוועדה ביחס למליאה בתוך הנתונים.

ב-SVM:

1. Precision:

- עבור מחלקה 0 : precision = 0.67
- עבור מחלקה 1 : precision = 0.76

מה שמציין יתרון בדיוק בזיהוי המליאה ביחס לוועדה.

2. Recall:

- עבור מחלקה 0 : recall = 0.81
- עבור מחלקה 1 : recall = 0.60

מה שמציין שיש יתרון קל בזיהוי הוועדה ביחס למליאה בתוך הנתונים.

מה שניתן להסיק מההבדלים הללו הוא שכל אלגוריתם וכל סוג של ערכי המאפיינים מצליחים לזהות ולסווג חלק מהמחלקות בצורה יותר טובה מאחרות.

2. האם תוצאות הסיווג בחלוקת אימון-בדיקה של 10%-90% דומות לתוצאות הcross validation? -בין אם כן ובין אם לאו, נסו לשער מדוע

מודל	וקטור	CV Accuracy	Stratified Train-Test **90%-10%**Accuracy
KNN	עבור Bow	0.85	0.84
SVM	עבור Bow	0.89	0.88
KNN	עבור וקטור המאפיינים	0.68	0.69
SVM	עבור וקטור המאפיינים	0.71	0.71

ניתן לראות שבאופן כללי התוצאות כמעט דומות(הפרש 1 בניהם).

העובדה שהתוצאות כמעט דומות מצביעה על כך שהמודלים יציבים ורובוסטיים גם כאשר מוצגים להם נתונים חדשים שלא היו חלק מהאימון. זו תוצאה חיובית שמראה שהמודלים לא סובלים מ-overfitting מובהק לנתוני האימון ומסוגלים להכליל את למידתם לנתוני בדיקה שלא ראו במהלך האימון.

3.הסבירו מהם היתרונות והחסרונות של שני סוגי המסווגים KNN, SVMבהם השתמשתם. האם לדעתכם אחד מהם עדיף על פני השני, עבור משימת הסיווג שבתרגיל?

היתרונות והחסרונות של KNN ו-SVM

:KNN

יתרונות: פשוט להבנה וליישום, אינו דורש הנחות מוקדמות על ההתפלגות של הנתונים.

חסרונות: רגיש לנתונים רועשים ולממדיות גבוהה. דורש חישוב רב עבור כל דוגמא חדשה.

:SVM

יתרונות: יעיל במרחבים עם מימדיות גבוהה, רגישות נמוכה יותר לנתונים רועשים.

חסרונות: יכול להיות קשה לאימון, במיוחד כאשר מספר הדוגמאות גדול מאוד. היה איטי בשלב האימון.

נראה שה-SVM עדיף על פני ה-KNN למשימת הסיווג הנתונה, מכיוון שיש לו רגישות נמוכה יותר לרעש, והוא הציג דיוק גבוה יותר בשני ווקטורי המאפיינם. וכשהשתמשנו ב-CountVectorizer הדיוק היה 0.65 , כנראה המודל היה רגיש מאוד.

4.פרטו את היתרונות והחסרונות ליצירת יחידות הסיווג. מה יהיו ההשלכות אם נגדיל ואם נקטין אותן באופן משמעותי?

יתרונות וחסרונות יצירת יחידות הסיווג(chunks):

יתרונות: יחידות סיווג גדולות יותר מאפשרות לכלול יותר הקשר ומידע תוכני שיכול לסייע בזיהוי המחלקה של הטקסט.

חסרונות: גודל גדול מדי של יחידת סיווג עלול להכיל יותר רעש ולהקשות על המודל למצוא מאפיינים ספציפיים הקשורים לסיווג. כמו כן, גודל קטן מדי עלול לא להכיל מספיק מידע לסיווג מדויק.

הגדלת יחידות הסיווג עלולה להוביל למודלים שמתמודדים טוב יותר עם הקשרים ארוכי טווח אך עלולים להיות פחות מדויקים בזיהוי תכונות ספציפיות. הקטנת יחידות הסיווג תכלול פחות הקשר אך עשויה לאפשר זיהוי יעיל יותר של תכונות מפתח לכל מחלקה.

שאלת בונוס:
התנסו בגדלים שונים של יחידות סיווגchunks( ) והסבירו מה לדעתכם מספר המשפטים האידיאלי למשימת הסיווג בתרגיל זה? ענו על כך בדו"ח

נסינו גודל chunks משתנה מ-2 משפטים ועד 100 (מצורף תרשים) משפטים בכל יחידת סיווג.

נראה שגודל אידיאלי של יחידות סיווג עבור המשימה זו נע בין 15 ל-22.

גודל זה מאפשר למודלים לנצל הקשר רחב ומידע תוכני ללא הצורך להתמודד עם רעש מיותר.