תרגיל בית 3

תאריך הגשה: 1.7.2017 23:55

אחראי על התרגיל: נמרוד רייפר

מטרת התרגיל:

- היכרות מעמיקה עם python.
- עבודה עם מאגר של מסמכים ובניית ייצוג ווקטורי לכל מסמך.
- הכרת sklearn.
- הכרת NLTK.

מבוא:

לרשותכם מאגר עבור binary sentiment analysis בקובץ zip המכיל חוות דעת על סרטים. המאגר חצוי לשניים, כאשר חציו האחד משמש עבור שלב האימון (train) וחציו האחר עבור שלב המבחן (test).

כל קובץ הוא חוות דעת על סרט כלשהו כאשר שם הקובץ קובע את תיוגו. לדוגמא: קובץ טקסט ששמו 10_7.txt מתאר חוות דעת לסרט אשר הציון שניתן הוא 7 ומאחר שבמדד בין 1 ל-10 7 הוא מעל 5, חוות הדעת מוגדרת כחיובית (class 1).

שימו לב, כל חוות דעת תהיה שייכת לאחד מהמחלקות 1 או 0 כאשר ציון גדול ממש מ-5 משייך את חוות הדעת למחלקה 1 וקטן או שווה ל-5 למחלקה 0.

מטרתכם: לבנות מסווג אשר יצליח לחזות האם חוות דעת היא חיובית או שלילית וזאת ברמת דיוק של לפחות 65%.

כדי להשיג את המטרה, נפעל בשלבים. ראשית נעבד את המאגר לכדי קבצים אחודים כאשר כל שורה בהם מתארת חוות דעת יחידה. לאחר שנסיים את שלב זה נוכל לבחור את האלגוריתם המתאים ללמידת המאגר ומציאת הפרמטרים הטובים ביותר כך שנוכל לחזות ברמת דיוק טובה את מאגר המבחן.

שלב א 40 נק': מעבר מ-raw text ל-bag of words

בשלב זה עליכם לנתח את המידע הלא מעובד (raw text) להעבירו למצב מובנה בתצורת bag of words.

המאגר שלרשותכם מכיל קבצים, כאשר בכל קובץ, ישנה חוות דעת על סרט כלשהו. עליכם לכתוב סקריפט המקבל תיקיות של חוות דעת ומייצר קבצי פלט על פי התצורה שנקבעה ע"י המשתמש.

שם הסקריפט: bag_of_words.py

מבנה הרצת הסקריפט:

```
python bag_of_words.py folder_1 folder_2 … folder_n -t -s -l
```

כאשר folder_n … folder_2 folder_1 הינם נתיבים לתיקיות בהן מופיעים קבצים המכילים כל אחד חוות דעת.

והדגלים l– s– t– מגדירים את תצורת הרצת התכנית.

<u>תצורת t–:</u>

מאפשרת למשתמש בקריאה לסקריפט לייצר ייצוג tf.idf לכל חוות דעת (כאשר idf יחושב על פני כל התיקיות כמאגר אחוד עם חישוב ה-log כפי שנלמד בהרצאה).

ללא סימון תצורה זו הייצוג שיווצר ע"י הסקריפט הינו ייצוג tf.

<u>תצורת s–:</u>

מאפשרת למשתמש בקריאה לסקריפט לייצר תוצר לכל תיקייה בנפרד. במילים אחרות, לכל קובץ ייוצר בהתאמה קובץ פלט עבורו בתיקייה ממנה נקרא הסקריפט. ללא סימון תצורה זו תוצרי התיקיות יישמרו בקובץ יחיד אשר ייווצר בתיקייה ממנה נקרא הסקריפט. סדר הדוגמאות בקובץ האחוד הינו לפי סדר התיקיות שסופקו.

** במידה והמשתמש לא בחר בקונפיגורציה הזו יש לקרוא לתוצר המתקבל בשם dataset תוך צירוף הסיפא (סיומת) עליה נספר בהמשך.

<u>תצורת l–:</u>

מאפשרת למשתמש לסמן לסקריפט לייצר את התוצרים בפורמט svm_light . על הפורמט הזה למדתם במעבדה ותצטרכו להשתמש בו כאשר תעברו לחלק השני של התרגיל.

שימו לב שהחלק הזה מצריך חשיבה בהיבט ההנדסי שכן, כעת ייצוג המזהה לתכוניות (features) הינו ע"י מספר ולא ע"י המילה עצמה. לכן יש צורך לשמור על שפה משותפת בין כלל המאגר, לרבות התיקיות אשר נקראו לסקריפט. נראה דוגמה בהמשך.

מומלץ לתצורה זו – למיין את כל המילים אשר הופיעו בכל חוות הדעת לפי סדר הא"ב, מה שיאפשר מספור נוח של התכוניות.

**שימו לב: כל התצורות יכולות להיות מופעלות יחד או בכל שילוב אחר של 3 התצורות, כך שלמעשה הסקריפט שתכתבו בחלק זה יכיל 3^2 קונפיגורציות.

כיצד המשתמש יזהה את התצורה שנוצרה ע״י הסקריפט?

כל קובץ שנוצר ע״י הסקריפט מקבל סיפא המתארת את התצורה. לדוגמא:

```
python bag_of_words.py -s -l -t aclImdb/train/pos aclImdb/train/neg aclImdb/test/pos aclImdb/test/neg
```

ייצר לנו 4 תוצרים בתיקייה ממנה נקרא הסקריפט.

test_pos_SLT.txt, test_neg_SLT.txt, train_pos_SLT.txt, train_neg_SLT.txt

שימו לב כי בסיפא, יש חשיבות לסדר ולכן לעולם בכל תוצר שהסקריפט יוצר, יהיו 3 אותיות כאשר אות גדולה מסמנת שהתצורה נבחרה ואות קטנה שהיא לא נבחרה.

דוגמא נוספת:

```
python bag_of_words.py -s -t aclImdb/train/pos aclImdb/train/neg aclImdb/test/pos aclImdb/test/neg
```

ייצר לנו 4 תוצרים בתיקייה ממנה נקרא הסקריפט.

```
test_pos_SIT.txt, test_neg_SIT.txt, train_pos_SIT.txt, train_neg_SIT.txt
```

שימו לב כי הסקריפט בונה לכל תיקייה קובץ בעל שם המתאר את שם התיקייה הקרובה ביותר לקבצים ואת שם התיקייה הקרובה ביותר אליה המכילה אותה. בדוגמא מטה שימו לב כי הסקריפט לוקח את שם התיקייה האחרונה ושם התיקייה המכילה אותה ומהם בונה את שם הקובץ הנוצר.

למשל test/neg מקבל תוצר בעל השם test_neg_***.txt והכוכביות יוחלפו בתצורה המתאימה שנבחרה.

כיצד נראה ייצוג bag of words:

שימו לב: דוגמא זו מתארת את המקרים בהם הרצת הסקריפט היא ללא התצורה l– .

כדי להבין את הייצוג, נראה זאת ע״י דוגמא.

קובץ הטקסט בשם 10_7.txt מהצורה הזו

```
John likes to watch movies. Mary likes movies too
```

מקבל שורה אחת בלבד המתאימה בקובץ התוצר של הסקריפט

```
7 john:1 likes:2 to:1 watch:1 movies:2 mary:1 too:1 # 10_7.txt
```

במקרה הזה המילה הראשונה היא דירוג הסרט ע"י חוות הדעת, המילה האחרונה היא שם הקובץ, והתיאור הפנימי הוא פירוק הטקסט ל-bag-of-words. למשל המילה movies הופיעה פעמיים ולכן היא מופיעה בצורה movies:2.

זוהי הצגה מוכרת בעולם ה-data במבנה של key-value כאשר במקרה שלנו key הוא המילה, וה-value הוא מספר ההופעות של המילה בטקסט.

<u>שימו לב כי על הפירוק של הטקסט ל-bag-of-words לבצע:</u>

1. המרה ל-lower-case.
2. הורדת מילות stopwords.
3. ביצוע stemming.
4. ספירה של מספר הפעמים בהם המילה מופיעה בחוות הדעת.
5. פירוק שם הקובץ לדירוג ומספר הסרט כדי לשמור זאת לצד המבנה key-value של ה- bag of words.

שימו לב, חלק העיבוד המקדים הוא חלק חיוני בהצלחת התרגיל. כאן גם המקום להיות יצירתיים להחליט אילו מילים אנחנו מנקים לחלוטין מהעיבוד שלנו כדי שישפרו את תהליך הלמידה. כאן המקום להביא expert knowledge ולהיות סטטיסטיקאים ולא רק מהנדסים.

אתם רשאים להביא כל מידע חיצוני שיסייע לכם בבניית הייצוג של חוות הדעת אך עליכם להכיר את הכלים שאתם משתמשים בהם ולהסביר מדוע הם מסייעים לכם.

כיצד נוכל לייצר מנשק משתמש התומך בתצורות וקבלת פרמטרים לסקריפט מהמשתמש:

האופציה המועדפת היא להשתמש בחבילה הנקראת OptionParser. לשם כך אפשר להתקין באופן מקומי את החבילה של אנקונדה ולהתקין עליה באמצעות הפקודה הבאה את החבילה:

```
pip install optparse
```

האופציה השנייה היא להשתמש במשתני המערכת argc ו-argv אבל היא תייצר לכם שעות עבודה נוספות שלא לצורך.

<u>הדגמת המעבר לפורמט svm_light:</u>

לשם כך, עליכם לסדר את כלל המילים במאגר בסדר א"ב, ולמספרן לפי סדר מקטן לגדול (את המספור יש להתחיל מ-1).

לאחר מכן, ייצוג של כל חוות דעת הינה מהצורה key-value כאשר key הוא לא המילה עצמה, אלא המספר שלה כפי שהתקבל מסידור המילים לפי סדר הא"ב.

לדוגמא:

עבור קובץ הטקסט בשם 10_7.txt מהצורה הזו

```
John likes to watch movies. Mary likes movies too
```

ועבור קובץ הטקסט בשם 14_3.txt מהצורה הזו

```
Mary likes movies too
```

נקבל בסידור לפי א"ב

```
7 john:1 likes:2 mary:1 movies:2 to:1 too:1 watch:1 # 10_7.txt
3 mary:1 likes:1 movies:1 too:1 # 14_3.txt
```

ובפורמט svm_light (הפורמט הסופי בו אנו חפצים בשלב זה)

```
7 1:1 2:2 3:2 4:1 5:1 6:1 7:1 # 10_7.txt
3 2:1 3:1 4:1 6:1 # 14_3.txt
```

אם זיהיתם נכון, כאן התצורה היא ללא t כלומר, ללא הפעלת idf.

שלב ב 60 נק': הרצת התוצרים ללמידה והסקה

כעת לאחר שיצרתם סקריפט אשר מנתח טקסט חופשי במאגר וממיר אותו לפורמט svm_light נרצה להשתמש בתוצרים המתקבלים בשלב ב' על מנת לאמן ולבחון מודל svm (תוכלו להשתמש באלגוריתם אחר בחבילת sklearn ובפרט שתהיו מסוגלים להסבירו כפי שאתם נדרשים בהמשך).

מהו אלגריתם svm? על קצה המזלג, זהו אלגוריתם המשמש לסיווג בין שתי קטגוריות. דוגמאות האימון מיוצגות ע"י נקודות במרחב ומטרת האלגוריתם הינה למצוא מישור מפריד בין הנקודות משתי הקטגוריות. המסווג מנסה למצוא את המישור המפריד בעל המרחק המקסימלי בין הדוגמאות הקרובות לו משתי הקטגוריות.

בנו סקריפט בשם learn.py אשר מקבל קובץ train וקובץ test בפורמט svm_light ומייצר קובץ אשר מחזיר את תוצאת החיזוי לכל אחד מחוות הדעת בקובץ ה-test.

עבור כל שורה שהתקבלה בקובץ מדגם המבחן יש להחזיר שורה מתאימה:

```
file_name:prediction
```

כאשר ערך הprediction הוא החיזוי למחלקה אליה שייכת הדוגמה ע"פ אלגוריתם הלמידה בו השתמשתם.

שימו לב שכדי לחזות על ה-test באמצעות ה-train שלב בניית הקבצים צריך להתבצע על כל המאגר ללא הפרדה וזאת על מנת שבמידה ובחרתם לבצע חישובי הtf.idf הכניסות בווקטורים יהיו תואמים.

כמו כן, לצורך האצת התהליך תוכלו להשתמש ב-MinMaxScaler המגיע יחד עם חבילת sklearn.

עליכם להשתמש ב- sklearn לצורך קריאת הנתונים, בניית המודל והסקה באמצעות המודל על מאגר המבחן.

על הסקריפט לעבוד בתצורה הבאה:

```
learn.py train test
```

כאשר train מציין את הנתיב לקובץ מאגר האימון ואילו test מציין את הנתיב לקובץ מאגר הבדיקה.

לחלק זה עליכם לצרף בנוסף **דו"ח** המכיל את הפרטים הבאים:

בהרצת אלגוריתם הנבחר:

1. עקומה של המדדים בהינתן threshold (ציר ה-x) מ-3 ועד 7 לקביעה האם חוות דעת שייכת לקטגוריה של חוות הדעת הטובות או חוות הדעת הרעות, עבור המדדים: accuracy,

precision ו-f1 כאשר השניים האחרונים הם תוך התחשבות בכל אחת מהמחלקות (חוות דעת חיוביות או חוות דעת שליליות).

2. עבור threshold=5 הציגו את המשקלות המתקבלים ל-10 המילים בעלות הערך הגבוה ביותר ול-10 המילים בעלות הערך הנמוך ביותר.

את המשקלות תוכלו לשלוף בקלות באמצעות השדה _coef.

כאן תוכלו להשתמש בכל אלגוריתם של חבילת sklearn ובתנאי שאתם מסוגלים לספק הסבר לסעיפים 1 ו-2.

- חלק זה הוא חובה עבור כל הסטודנטים, לרבות אלו אשר אינם משתתפים בתחרות.

שימו לב שבחלק זה עליכם לספק אלגוריתם אשר מצליח לסווג במידת דיוק של לפחות 65% תחת נתונים מעובדים לפי קונפיגורציה שמתחשבת במדד idf.

שלב ג 10 נק' בונוס: תחרות (רשות)

נרצה לשפר את אלגוריתם בו השתמשתם בסעיף קודם. כתבו סקריפט בשם learn_expert.py אשר מבצע חיזוי בדיוק באותו אופן בשלב ב רק כאשר הפעם באפשרותכם לבצע שינויים על הנתונים ועל האלגוריתם אשר יניבו תוצאות טובות יותר.

את השיפור יש לקודד ולהציג תוצאות כפי שנתבקשתם בשלב ב'.

לשם כך עליכם להכין מצגת בת 2 שקופיות, בשקופית הראשונה, עליכם להציג את שמכם, ואת השיטה בה השתמשתם לרבות האופן בו ניתחתם את הטקסט(שלב העיבוד ועד לשלב הייצוג של הדוגמאות ע״י תכוניות) והאופן בו השתמשתם באלגוריתם. בשקף השני, עליכם להציג ניתוח של התוצאות כפי שנדרשתם לו בחלק ב' ואת 5 המילים בעלות המשקל הגבוה ו-5 המילים בעלות המשקל הנמוך.

כמו כן הציגו בשקף מדדים כמו accuracy, precision ומדד f1.

<u>הניקוד לתחרות</u>

התחרות תתקיים בין הסטודנטים לפי רמת הדיוק המתקבלת על מאגר המבחן כאשר ניתן לקבל עד 10 נקודות בונוס לציון התרגיל.

סטודנטים שישיגו את רמת הדיוק הגבוהה ביותר יזכו ל-10 נקודות בונוס כאשר הסטודנטים הבאים אחריהם יזכו ל-8 נקודות ואלו שלאחר מכן יזכו ל-6 וכן הלאה.

שימו לב כי רעיונות מקוריים במיוחד גם אם לא יצליחו להשיג רמת דיוק גבוהה מספיק עשויים גם הם לזכות בבונוס לתרגיל.

על כל מקרה שבו הציון הסופי לתרגיל עובר את סך 100 הנקודות, הציון הסופי לתרגיל על 100 נקודות.

הנחיות נוספות

1. עליכם ליישם את העקרונות אשר נלמדו בקורס:
   i. דגש מיוחד יושם על מימוש מחלקות, העמסת אופרטורים וחלוקה למודולים וקבצים שונים.
   ii. עצם העובדה שהוגדרו שמות לסקריפטים אין זה אומר שלא ניתן להוסיף סקריפטים נוספים אשר הסקריפטים הראשיים יבצעו בהם שימוש. זוהי פרקטיקה חשובה שתיבדק בתרגיל.
   iii. למדתם בקורס על העמסת אופרטורים ההופכים אובייקט לבר-סיור (iterable) עליכם לדאוג שלפחות אחת מן המחלקות אשר תכתבו בתרגיל תהיה ברת סיור ועליכם לבצע שימוש בתכונה זו בסקריפטים אשר תכתבו.
   iv. למדתם בקורס על property. עליכם לעשות בו שימוש לפחות באחת המחלקות שתבחרו לכתוב כמו גם השימוש במתודה סטטית.
   v. עליכם לתעד בפורמט שנלמד בקורס את הפונקציות והאובייקטים השונים.
2. את קבצי הקוד, תוצאות החיזוי לכל אחד מהשלבים ואת המצגת והדו"ח יש להגיש בקובץ ZIP. כאשר שם הקובץ הוא id1_id2_p2.zip בהתאמה למספרי תעודות הזהות של הסטודנטים המגישים אם ההרצה היא על פייתון 2 . ו- id1_id2_p3.zip אם ההרצה היא על פייתון 3.
   i. קובץ תוצאות החיזוי לחלק ב ייקרא learn.txt.
   ii. קובץ תוצאות החיזוי לחלק התחרות ייקרא learn_expert.txt.
3. את השאלות על תרגיל הבית יש לשאול אך ורק באמצעות הפורום המיועד לתרגיל. אין לשלוח שאלות למייל למעט עניינים אישיים.
4. התרגיל ייבדק בדיקה מקיפה:
   i. הקפידו על מוסכמות קוד.
   ii. תעדו את הקוד שלכם היכן שצריך ובאופן קריא ותמציתי.
   iii. הקוד חייב לרוץ ללא שגיאות. שגיאות בריצת התוכנית לא יתוקנו ומשמעותן ציון 0 לתרגיל. כמו כן תרגיל שיספק תוצאות חיזוי אך הקוד לתרגיל מתרסק במהלך ריצת התכנית משמעותו ציון 0 לתרגיל.
   iv. כתבו יחידות בדיקה לתרגיל שלכם על מנת לוודא שאכן אתם עומדים בהנחיות התרגיל כנדרש.
5. סטודנטים שיבחרו להטמיע באופן מלא את mypy בקוד שלהם יקבלו נקודת בונוס (ניתן להגיע עד ל10 נקודות בונוס לתרגיל).

טיפים:

1. שמירת קבצים בפורמט svm-light תאפשר לכם לפתוח את הקבצים בקלות באמצעות הפונקציה הזו.
2. כדי לפתוח את קבצי חוות הדעת בצורה נוחה מומלץ להשתמש בחבילה codecs. לדוגמא:
   i. codecs.open('words_df.txt', 'w', encoding='utf-8')
   ii. שאר השימוש באובייקט שנוצר זהה לכפי שלמדנו עבודה עם קבצים עד כה.
3. מומלץ להשתמש בחבילת NLTK לצורך עבודה עם טקסט וניפוי של מילות stopwords למשל.
   i. הפעילו כאן שיקול דעת: האופן שבו תעבדו את המידע ישפיע באופן ישיר על איכות החיזוי של האלגוריתם שלכם.
   ii. הרחיבו מעט על חוות דעת של סרטים על מנת להבין מהן המילים המשפיעות על חוות דעת להיות טובה או רעה.
4. עבדו על גרסה לוקאלית של anaconda ועל מדגם קטן של המידע על מנת שתוכלו לנפה שגיאות בקלות ותוכלו להתנסות בשלל חבילות.

בהצלחה!