תרגיל בית 3 – סיווג טקסטים

מטלה זו עוסקת בסיווג טקסטים בעזרת אחת החבילות הפופולאריות והשימושיות של פייתון - scikit-learn, אותה תכירו בתרגיל זה.
היכולת לקרוא ולהבין תיעוד של תוכנה ולהתנסות לבד בדברים חדשים היא יכולת חיונית לעבודתכם בהמשך. הפיקו מתרגיל זה את המירב.

יש להכיר את תיעוד החבילה scikit-learn (ומומלץ אף להתנסות מעשית). בפרט, הנושאים הרלבנטיים לתרגיל זה נמצאים בקישורים הבאים:

- http://scikit-learn.org/stable/tutorial/index.html תיעוד כללי ודוגמאות
- http://scikit-learn.org/stable/tutorial/text_analytics/working_with_text_data.html סיווג טקסטים
- http://scikit-learn.org/stable/modules/cross_validation.html הערכת ביצועים של מסווג
- http://scikit-learn.org/stable/modules/feature_selection.html בחירת features

המשימה בתרגיל זה היא לזהות באופן אוטומטי את הדובר של טקסט נתון, וביתר דיוק, להפריד בין טקסטים של שני דוברים שונים. תשתמשו בשיטת קלאסיות של authorship attribution, שמבוססות על סיווג אוטומטי של טקסטים. בשלב הראשון תגדירו את המאפיינים (features) בעצמכם ובשלב השני תעשו זאת בעזרת הפונקציות של scikit-learn.

לצורך המשימה נתון קורפוס של שני דוברים : ראובן ריבלין ויולי יואל אדלשטיין.

הקורפוס מופרד לקבצים, כך שבכל קובץ נמצאים רק טקסטים של אותו הדובר. שני הקורפוסים הופרדו למשפטים ועברו טוקניזציה. אינכם צריכים לערוך את הטקסט עצמו.

<u>שלב 1</u>

ניתוח מקדים:

כדי שמשימת הסיווג תהיה מאוזנת, אנחנו מעוניינים להשתמש במספר שווה של משפטים לכל דובר. עברו על שני הקורפוסים וצמצמו באופן אקראי את הגדול יותר כך שיהיה מספר שווה של משפטים לשני הדוברים.

סיווג בעזרת בחירת features באופן ידני:

א. חישבו על מאפיינים שיכולים לאפיין את הסגנון של הדוברים. מאפיין יכול להיות מילים מסוימות, מספר סימני פיסוק, אורך משפט וכו'. נסו לחשוב אלו מאפיינים הם ספציפיים לסגנון, ואלו משקפים יותר תוכן. המאפיינים צריכים להיות מיוצגים ע"י מספר.
ב. לכל משפט הגדירו וקטור של המאפיינים שהגדרתם. כך תקבלו וקטור של מספרים באורך קבוע.
ג. לכל משפט הגדירו את הדובר אליו הוא שייך (ריבלין או אדלשטיין): 0 או 1.
ד. סווגו את הנתונים מהסעיף הקודם בעזרת המסווגים הבאים (שכולם קיימים ב-scikit-learn):

- SVM (SVC)
- Naive Bayes (MultinomialNB)
- DecisionTree (DecisionTreeClassifier)
- KNN (KNeighborsClassifier)

ה. העריכו את הביצועים של כל מסווג בעזרת ten-fold-cross-validation ודווחו על הדיוק (accuracy) **הממוצע** של כל ה folds . פרטו את התוצאות שקבלתם ודונו בהם: האם הן כפי שציפיתם? האם יש מאפיינים שתורמים לסיווג טוב יותר? איזה מסווג עבד טוב יותר מאחרים?

<u>שלב 2</u>

נתון קובץ עם 100 המילים הנפוצות ביותר בשפה העברית.

א. בנו feature vector עבור כל אחד מהמשפטים. כל ערך בוקטור הוא אינדיקציה בוליאנית לקיום מילה (מתוך רשימת 100 המילים הנפוצות) במשפט. למשל, אם המילה "היה" מופיע במשפט, הערך המייצג אותה ב feature vector יהיה 1, אחרת 0. אין להעזר ב CountVectorizer של scikit-learn בסעיף זה, אלא **עליכם לבנות את מבנה הנתונים בעצמכם**.
ב. חזרו על סעיפים ג'-ה' מהשלב הקודם.

<u>שלב 3</u>

סיווג בעזרת:bag-of-words

בחלק זה נשתמש בשירותי scikit-learn על מנת לבנות את ה- feature vectors מהטקסט באופן אוטומטי.

א. בנו feature vectors מהטקסטים (למשל, בעזרת CountVectorizer). השתמשו **בכל המילים של הטקסט** ולא במילון מוגבל. המשיכו וחשבו שכיחויות (לא אינדיקציה בינארית) של כל המילים (השתמשו ב-tf-idf) בעזרת TfidfTransformer . כמה מילים שונות ישנן בטקסטים (במילים אחרות, מה אורך ה feature vectors -שנוצרו)?
ב. חזרו על סעיפים ג'-ה' משלב 1.

<u>שלב 4</u>

א. **בחירת המילים המשמעותיות ביותר לסיווג:**
בשלב 3 השתמשתם בכל מילות הטקסט כקלט למסווג. ברור כי חשיבות המילים ותרומתן למשימת הסיווג אינה זהה. ככל הנראה יש מילים שמאפיינות יותר את ריבלין ומילים אחרות שמאפיינות יותר את אדלשטיין. החבילה scikit-learn מספקת כלים לביצוע ניתוח הקלט, על מנת לאמוד את התרומה היחסית של כל feature (מילה, במקרה שלנו) למשימת הסיווג.
השתמשו ב SelectKBest על מנת לבחור את 50 המילים בעלות התרומה **הגבוהה ביותר** לסיווג. התרשמו מרשימת המילים והשוו אותה לרשימת המילים הכי נפוצות והמילים שחשבתם עליהן בסעיף 1.א. האם כל המילים שקבלתם כעת הן צפויות? פרטו את התוצאות (50 המילים).

ב. **סיווג בעזרת רשימת מילים מצומצמת (להבדיל מbag-of-words):**
חזרו על השלב הקודם , אבל כעת השתמשו ברשימת מילים סגורה
ל CountVectorizer (יש לו אופציה לקבל מילון, ראו תיעוד). השתמשו ברשימת 50 המילים בעלות התרומה הגבוהה ביותר לסיווג שקיבלתם בסעיף הקודם.

ג. השוו את התוצאות למספרים שקבלתם בשלב 3 bag-of-words האם ההבדלים משמעותיים?

הוראות הגשה:

- יש להשתמש בכל המסווגים עם **ערכי ברירת מחדל** (defaults) של פרמטרים, כלומר ליצור אותם עם constructor ריק. אנא הקפידו על כך בהגשה, כי התרגיל ייבדק בהנחה שהמסווגים עובדים עם ערכי ברירת מחדל.

- הגישו קובץ ZIP ובו קובץ קוד בשם **hw3.py** וקובץ תיעוד בשם **hw3report.pdf**. על הקוד להיות מופעל משורת הפקודה בעזרת

python hw3.py <input_dir1> <input_dir2> <top_hebrew_words> <best_words_file_output_path>

כאשר **input_dir1,input_dir2** הם תיקיות שבהן נמצאים הקבצים של הדובר הראשון והשני בהתאמה.

הקובץ **top_hebrew_words** הוא קובץ 100 המילים הכי נפוצות בעברית.

הקובץ **best_words_output_path** הוא קובץ פלט שיווצר בשלב 4 עם 50 המילים המשמעותיות ביותר לסיווג - כל מילה בשורה נפרדת, ללא סימנים נוספים.

על התוכנית להדפיס את **התוצאות של שלבים 1-4** כלומר את תוצאות הסיווג של ארבעה מסווגים כאשר משתמשים (1) באוסף מאפיינים שהוכנס באופן (2) בקובץ המילים הכי נפוצות בעברית (3) ב bag-of-words, ו- (4) ב-50 המילים שנבחרו ע"י SelectKBest . את ה accuracy-הדפיסו באחוזים, בדיוק של 2 ספרות אחרי הנקודה.

דוגמא לפלט מצופה:

step1 (my features):

- SVM: <accuracy>
- Naive Bayes: <accuracy>
- DecisionTree: <accuracy>
- KNN: <accuracy>

Step2 (top Hebrew words):

- SVM: <accuracy>
- Naive Bayes: <accuracy>
- DecisionTree: <accuracy>
- KNN: <accuracy>

Step3 (bag-of-words):

- SVM: <accuracy>
- Naive Bayes: <accuracy>
- DecisionTree: <accuracy>
- KNN: <accuracy>

Step4 (selected best features):

- SVM: <accuracy>
- Naive Bayes: <accuracy>
- DecisionTree: <accuracy>
- KNN: <accuracy>

קובץ התיעוד יכיל את התשובות לסעיפים הבאים:

1.א – פרטו את המאפיינים שבחרתם.

1.ה.

2.ב

3.א.

3.ב

4.א.

4.ג.

יש להקפיד על עבודה עצמאית. צוות הקורס יתייחס בחומרה להעתקות או שיתופי קוד.
תאריך הגשה: 9.12.2018 , עד השעה 23:59.

שאלות על התרגיל אפשר לשאול בפורום תרגילי בית.