מבוא לאיחזור מידע

המשך מעבדה 1 - ויזואליזציה של טקסט ועיבוד מקדים

וויזואליזציה

1. הציגו עננת מילים (כמה פעמים מופיעה כל מילה) עבור כל סיווג של הטוויטים ועבור האוסף המלא .(wordcloud)
2. אריחים המייצגים את 20 המילים הנפוצות ביותר בכל סיווג של הטוויטים.
3. גרף של כמות מילים ביחס לטוויטים.
4. גרף של כמות תווים ביחס לטוויטים (למה זה רלוונטי).
5. חפשו רעיון לשלוש ויזואליזציות אחרות רלוונטיות והציגו אותן - גרף או תצוגה גרפית אחרת שתעזור לנו להבין רעיון או איפיון מסויים של הטקסט. (חיפוש אחר EDA בתחומי הטקסט וה -NLP מומלץ)

עיבוד מקדים ליצירת מילון

שימו לב יש להציג נתונים וויזואליזציות רלוונטיות עבור כל שלב עיבוד מקדים

1. היפכו אותיות גדולות לקטנות.
2. הסירו סימני ניקוד ותווים מיוחדים (רמז regexp [^\s\w] will do the matching)
3. הסירו stopwords - היזכרו כיצד ספרנו את מס' המילים.
4. בשלב זה זהו את המילים הנפוצות ביותר - שימו לב לוויזואליזציות, האם יש מילים שאינן תורמות לנו ויש להסירן ?
5. זהו מילים נדירות, האם יש מילים שאינן תורמות לנו ויש להסירן גם ?
6. תקנו שגיאות כתיב - עליכם לחפש כלי מתאים בפייתון המטפל בשגיאות כתיב.

היפכו אותיות גדולות לקטנות

פייתון

כאן ניתן לראות טיפול בכל מילה בנפרד אולם היה ניתן גם להפעיל את lower על כל ה string

```
train['tweet2'] = train['tweet'].apply(lambda x: " ".join(x.lower() for x in x.split()))
```

אפשרות נוספת להשתמש במתודות של str

```
train['tweet2'] = train['tweet'].str.lower()
```

הסירו סימני ניקוד ותווים מיוחדים (רמז regexp [^\s\w] will do the matching)

```
train['tweet'] = train['tweet'].str.replace('[^\w\s]','')
```

הסירו stopwords - היזכרו כיצד ספרנו את מס' המילים

```
from nltk.corpus import stopwords

stop = stopwords.words('english')

train['tweet'] = train['tweet'].apply(lambda x: " ".join(x for x in x.split() if x not in stop))
```

מציאת 10 המילים הנפוצות ביותר

```
freq = pd.Series(' '.join(train['tweet']).split()).value_counts()[:10]
```

לאחר בדיקת המלים, ניתן להסירן

```
freq = list(freq.index)

train['tweet'] = train['tweet'].apply(lambda x: " ".join(x for x in x.split() if x not in freq))
```