*# File: WordCount\_JSON.py  
# Name: Gabriel Valenzuela  
# Date: 03/22/2020  
# Course: DSC 550 - Data Mining  
# Purpose: Pt. 1  
# Cleaning up and applying certain text techniques to a data frame consisting of controversial comments  
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -***import** pandas **as** pd  
**import** numpy **as** np  
**import** string  
**from** nltk.corpus **import** stopwords  
**from** nltk.tokenize **import** word\_tokenize  
**from** nltk.stem.porter **import** PorterStemmer  
**from** sklearn.feature\_extraction.text **import** TfidfVectorizer  
  
df = pd.read\_json(**r'C:\Users\Gabe\PycharmProjects\DataMining\_2.2\_BuildYourTextClassifiers\controversial-comments.jsonl'** , lines=**True**)  
  
*# A. Convert all text to lowercase letters*df[**'txt'**] = df[**'txt'**].str.lower()  
print(df)  
  
*# B. Remove all punctuation from the text*exclude = set(string.punctuation)  
  
  
**def** remove\_punctuation(x):  
 x = **''**.join(ch **for** ch **in** x **if** ch **not in** exclude)  
 **return** x  
  
  
df[**'txt'**] = df[**'txt'**].apply(remove\_punctuation)  
print(df)  
  
*# C. Remove stop words*stop\_words = stopwords.words(**'english'**)  
df[**'tokens'**] = df[**'txt'**].apply(word\_tokenize)  
print(df)  
  
  
**def** remove\_stopwords(x):  
 x = **' '**.join(word **for** word **in** x **if** word **not in** stop\_words)  
 **return** x  
  
  
df[**'tokens'**] = df[**'tokens'**].apply(remove\_stopwords)  
print(df)  
  
*# D. Apply NLTK's PoterStemmer*ps = PorterStemmer()  
[ps.stem(word) **for** word **in** df[**'tokens'**] ]  
print(df)  
  
*# E. Use a Tf-idf vector instead of the word frequency vector*text\_data = np.array(df[**'tokens'**])  
tfidf = TfidfVectorizer()  
feature\_matrix = tfidf.fit\_transform(text\_data)  
print(feature\_matrix)