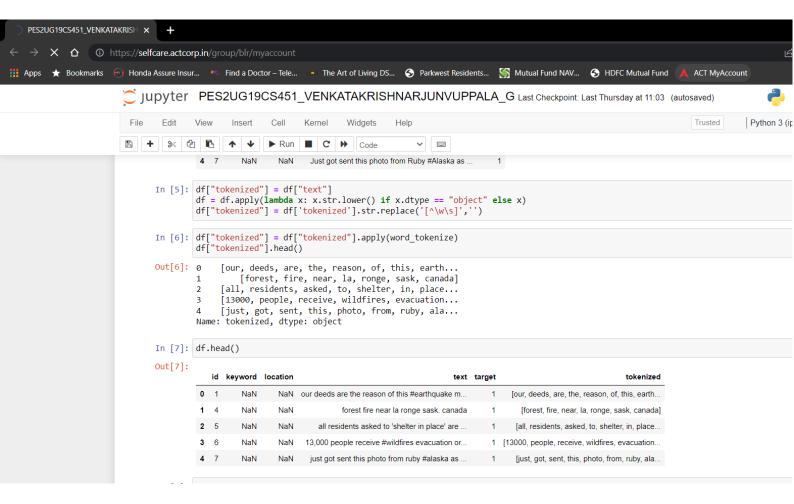
Name: Venkata Krishnarjun Vuppala
SRN: PES2UG19CS451
Section: G

Subject: Algorithms for Intelligence Web and Information Retrieval

# **Assignment 3**

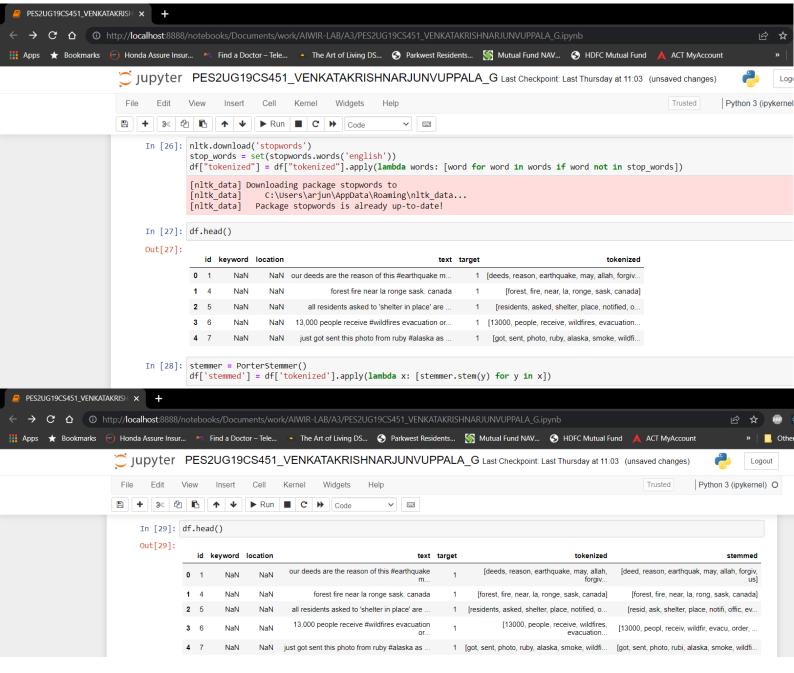
1. tCase folding

### **Code and Output:**



#### 2. stemming/lemmatization

# **Code and Output:**



### **Code and Output:**

jupyter PES2UG19CS451\_VENKATAKRISHNARJUNVUPPALA\_G Last Checkpoint: Last Thursday at 11:03 AM (autosaved)



```
Python 3 (
                                                Cell
                                                        Kernel
                                                                Widgets
                                                ► Run ■ C → Code
                 Standard inverted index construction
                      In [121]: my_list = df["stemmed"]
                                my_list[:5]
                     Out[121]: 0
                                      [deed, reason, earthquak, may, allah, forgiv, us]
[forest, fire, near, la, rong, sask, canada]
                                      [resid, ask, shelter, place, notifi, offic, ev...
[13000, peopl, receiv, wildfir, evacu, order, ...
                                      [got, sent, photo, rubi, alaska, smoke, wildfi...
                                Name: stemmed, dtype: object
                        In [*]: words = []
                                 for key in my_dict:
                                     for i in range(len(my_dict[key])):
                                        term = my_dict[key][i]
                                         words.append(term)
                                 length = len(my_list)
                                 inverted_index = {}
                                 for i in range(length):
                                     check = my_list[i]
for item in words:
                                         if item in check:
                                             if item not in inverted_index:
                                                 inverted_index[item] = []
                                             if item in inverted_index:
                                                 inverted_index[item].append(i+1)
                                 for key in inverted_index:
                                     inverted_index[key] = list(set(inverted_index[key]))
In [*]: for key in inverted index:
             print(key,": ",inverted_index[key])
         reason: [4992, 1, 1921, 4998, 5001, 5002, 782, 6167, 5373, 305, 306, 7219, 6454, 2748, 6460, 4670, 318, 320, 2113, 2253, 69 92, 6233, 6243, 747, 4844, 4334, 6899, 4085, 764, 3453, 895]
         earthquak : [1, 6025, 6031, 7311, 6946, 6947, 7590, 6955, 7600, 6973, 6975, 7137, 3028, 3029, 3030, 3031, 3033, 3034, 3035,
         3036, 7130, 3038, 3039, 3040, 7132, 3042, 7133, 3044, 3045, 7136, 3047, 3048, 3049, 3050, 3051, 3052, 5738, 7143, 3055, 7144,
         3059, 3060, 3061, 3062, 3063, 3065, 3066]
         may: [1, 5662, 5157, 3113, 4139, 4141, 2115, 4178, 1634, 4720, 4724, 4216, 5245, 5246, 5248, 5249, 5250, 5251, 5252, 5254,
         4233, 5258, 3724, 4236, 5260, 5261, 5263, 5265, 5267, 3222, 5270, 5271, 5272, 5273, 5274, 5275, 5276, 5277, 5278, 4256, 5802,
         3774, 5840, 4817, 7383, 5853, 2785, 2794, 241, 5368, 5395, 5411, 2342, 808, 1833, 3394, 6468, 2375, 3400, 6485, 4955, 7004, 2
         400, 5472, 6501, 895, 6533, 1935, 1936, 5011, 2463, 4517, 939, 4011, 7084, 7602, 2997, 1977, 5054, 1476, 2002, 3026, 4563, 20
         15, 2016, 6647]
         allah : [4256, 1, 4291, 4680, 4011, 4300, 3726, 6448, 4313]
         forgiv: [1, 2266, 2533]
         us: [2560, 1, 514, 5122, 5123, 5125, 3078, 2056, 5130, 2059, 1036, 3084, 7178, 5139, 5142, 3096, 7193, 4636, 5150, 5157, 4
         3, 5677, 3634, 7225, 7228, 4670, 7233, 5698, 5189, 1097, 2633, 1100, 4177, 85, 1627, 1116, 2141, 5214, 3680, 1633, 7268, 112
         5, 4200, 2153, 1646, 628, 629, 1158, 4230, 4232, 7305, 1675, 7311, 6290, 3734, 7320, 5793, 4258, 1699, 1193, 682, 7340, 4782, 2224, 4282, 1221, 1741, 6350, 2262, 6875, 2274, 3309, 3824, 3313, 247, 3834, 251, 252, 254, 260, 2820, 267, 269, 271, 272, 38
         60, 1304, 4889, 284, 6946, 6947, 2854, 2855, 3368, 2857, 2858, 4908, 6967, 6969, 2877, 830, 6461, 6975, 7486, 2373, 2885, 596
         0, 2892, 2384, 338, 4436, 856, 4980, 4981, 4985, 6011, 3967, 7042, 3979, 2956, 5008, 7063, 6553, 1439, 5537, 945, 6577, 5559,
```

#### 4. Positional index construction

# **Code and Output:**

