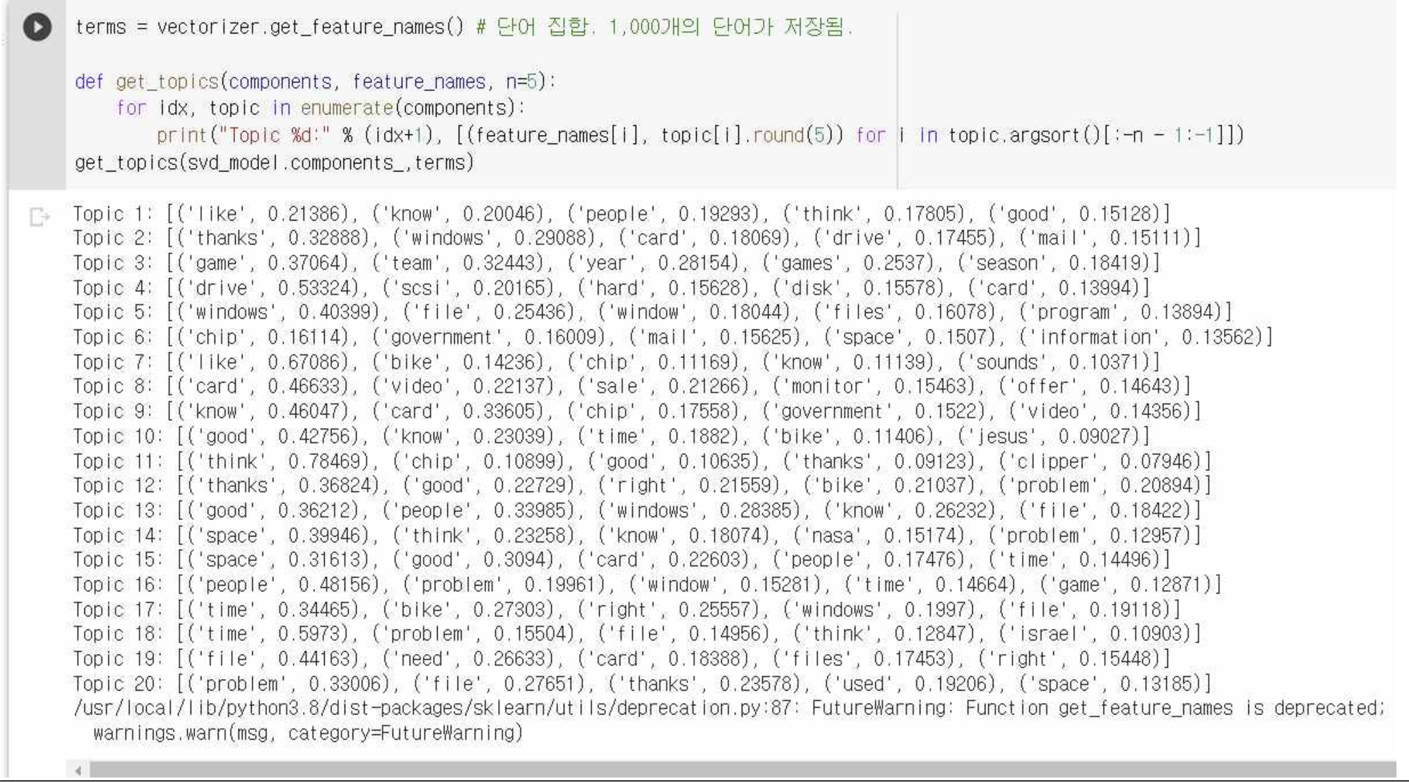
과제 #8

201904479 컴퓨터 전자 시스템 학과 임서연

1. 3쪽에서 지정된 과제를 수행

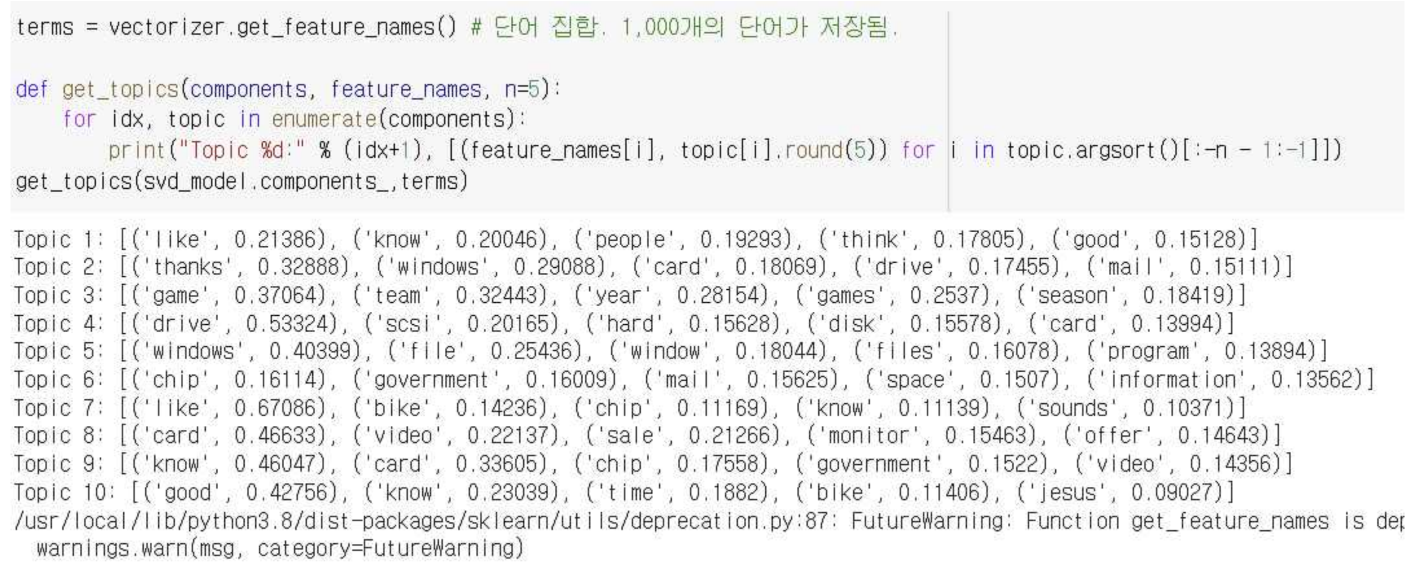


```
terms = vectorizer.get_feature_names() # 단어 집합. 1,000개의 단어가 저장됨.

def get_topics(components, feature_names, n=5):
    for idx, topic in enumerate(components):
        print("Topic %d:" % (idx+1), [(feature_names[i], topic[i].round(5)) for i in topic.argsort()[:-n - 1:-1]])
get_topics(svd_model.components_,terms)
```

```
Topic 1: [('like', 0.21386), ('know', 0.20046), ('people', 0.19293), ('think', 0.17805), ('good', 0.15128)]
Topic 2: [('thanks', 0.32888), ('windows', 0.29088), ('card', 0.18069), ('drive', 0.17455), ('mail', 0.15111)]
Topic 3: [('game', 0.37064), ('team', 0.32443), ('year', 0.28154), ('games', 0.2537), ('season', 0.18419)]
Topic 4: [('drive', 0.53324), ('scsi', 0.20165), ('hard', 0.15628), ('disk', 0.15578), ('card', 0.13994)]
Topic 5: [('windows', 0.40399), ('file', 0.25436), ('window', 0.18044), ('files', 0.16078), ('program', 0.13894)]
Topic 6: [('chip', 0.16114), ('government', 0.16009), ('mail', 0.15625), ('space', 0.1507), ('information', 0.13562)]
Topic 7: [('like', 0.67086), ('bike', 0.14236), ('chip', 0.11169), ('know', 0.11139), ('sounds', 0.10371)]
Topic 8: [('card', 0.46633), ('video', 0.22137), ('sale', 0.21266), ('monitor', 0.15463), ('offer', 0.14643)]
Topic 9: [('know', 0.46047), ('card', 0.33605), ('chip', 0.17558), ('government', 0.1522), ('video', 0.14356)]
Topic 10: [('good', 0.42756), ('know', 0.23039), ('time', 0.1882), ('bike', 0.11406), ('jesus', 0.09027)]
Topic 11: [('think', 0.78469), ('chip', 0.10899), ('good', 0.10635), ('thanks', 0.09123), ('clipper', 0.07946)]
Topic 12: [('thanks', 0.36824), ('good', 0.22729), ('right', 0.21559), ('bike', 0.21037), ('problem', 0.20894)]
Topic 13: [('good', 0.36212), ('people', 0.33985), ('windows', 0.28385), ('know', 0.26232), ('file', 0.18422)]
Topic 14: [('space', 0.39946), ('think', 0.23258), ('know', 0.18074), ('nasa', 0.15174), ('problem', 0.12957)]
Topic 15: [('space', 0.31613), ('good', 0.3094), ('card', 0.22603), ('people', 0.17476), ('time', 0.14496)]
Topic 16: [('people', 0.48156), ('problem', 0.19961), ('window', 0.15281), ('time', 0.14664), ('game', 0.12871)]
Topic 17: [('time', 0.34465), ('bike', 0.27303), ('right', 0.25557), ('windows', 0.1997), ('file', 0.19118)]
Topic 18: [('time', 0.5973), ('problem', 0.15504), ('file', 0.14956), ('think', 0.12847), ('israel', 0.10903)]
Topic 19: [('file', 0.44163), ('need', 0.26633), ('card', 0.18388), ('files', 0.17453), ('right', 0.15448)]
Topic 20: [('problem', 0.33006), ('file', 0.27651), ('thanks', 0.23578), ('used', 0.19206), ('space', 0.13185)]
/usr/local/lib/python3.8/dist-packages/sklearn/utils/deprecation.py:87: FutureWarning: Function get_feature_names is deprecated;
  warnings.warn(msg, category=FutureWarning)
```

2. 1번의 프로그램에서 토픽의 숫자를 10개로 바꾸고 이 때의 결과가 1번 결과와 어떻게 다른지 설명하라.



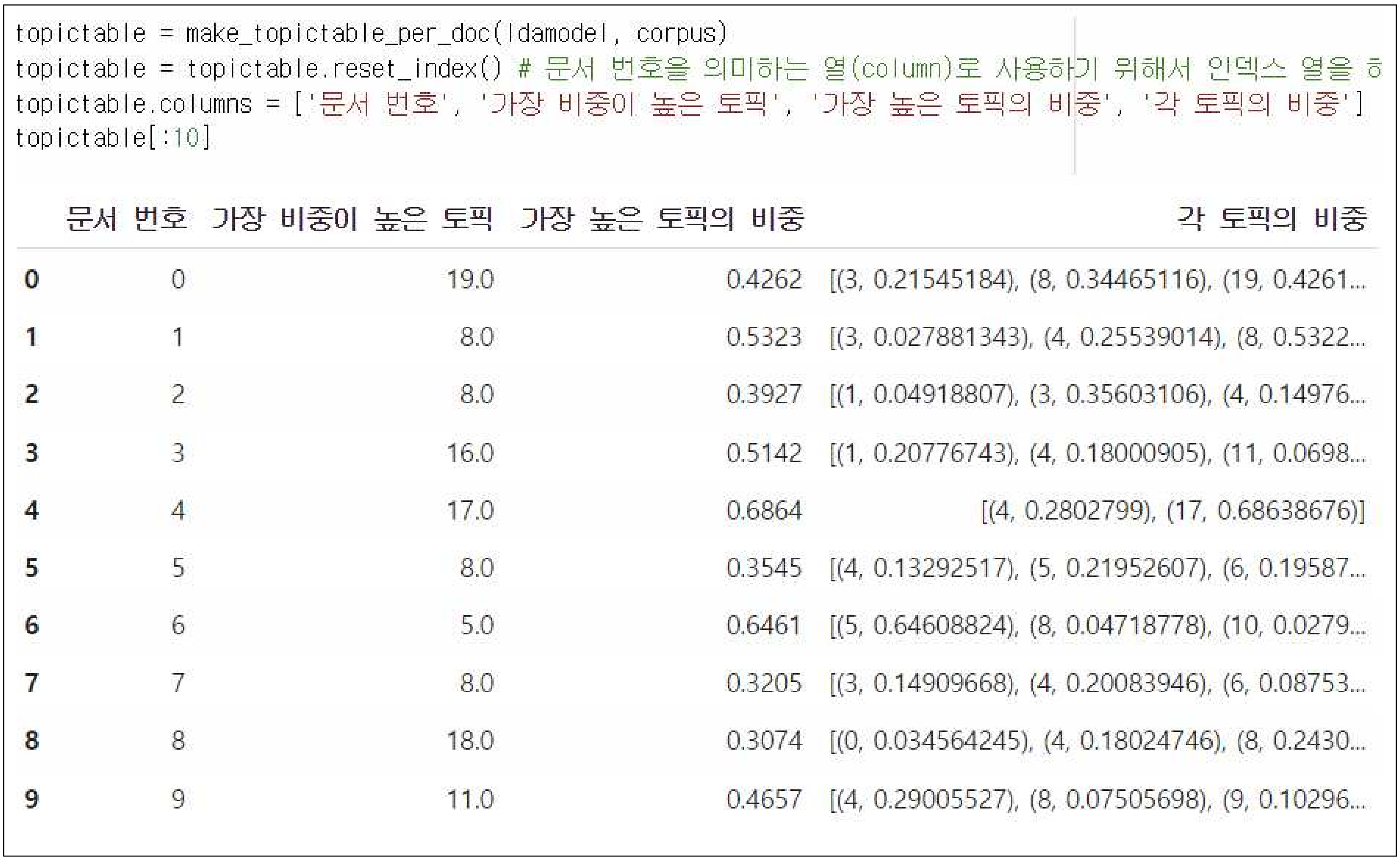
```
terms = vectorizer.get_feature_names() # 단어 집합. 1,000개의 단어가 저장됨.

def get_topics(components, feature_names, n=5):
    for idx, topic in enumerate(components):
        print("Topic %d:" % (idx+1), [(feature_names[i], topic[i].round(5)) for i in topic.argsort()[:-n - 1:-1]])
get_topics(svd_model.components_,terms)
```

```
Topic 1: [('like', 0.21386), ('know', 0.20046), ('people', 0.19293), ('think', 0.17805), ('good', 0.15128)]
Topic 2: [('thanks', 0.32888), ('windows', 0.29088), ('card', 0.18069), ('drive', 0.17455), ('mail', 0.15111)]
Topic 3: [('game', 0.37064), ('team', 0.32443), ('year', 0.28154), ('games', 0.2537), ('season', 0.18419)]
Topic 4: [('drive', 0.53324), ('scsi', 0.20165), ('hard', 0.15628), ('disk', 0.15578), ('card', 0.13994)]
Topic 5: [('windows', 0.40399), ('file', 0.25436), ('window', 0.18044), ('files', 0.16078), ('program', 0.13894)]
Topic 6: [('chip', 0.16114), ('government', 0.16009), ('mail', 0.15625), ('space', 0.1507), ('information', 0.13562)]
Topic 7: [('like', 0.67086), ('bike', 0.14236), ('chip', 0.11169), ('know', 0.11139), ('sounds', 0.10371)]
Topic 8: [('card', 0.46633), ('video', 0.22137), ('sale', 0.21266), ('monitor', 0.15463), ('offer', 0.14643)]
Topic 9: [('know', 0.46047), ('card', 0.33605), ('chip', 0.17558), ('government', 0.1522), ('video', 0.14356)]
Topic 10: [('good', 0.42756), ('know', 0.23039), ('time', 0.1882), ('bike', 0.11406), ('jesus', 0.09027)]
/usr/local/lib/python3.8/dist-packages/sklearn/utils/deprecation.py:87: FutureWarning: Function get_feature_names is dep
  warnings.warn(msg, category=FutureWarning)
```

결과 값이 10개로 나온다

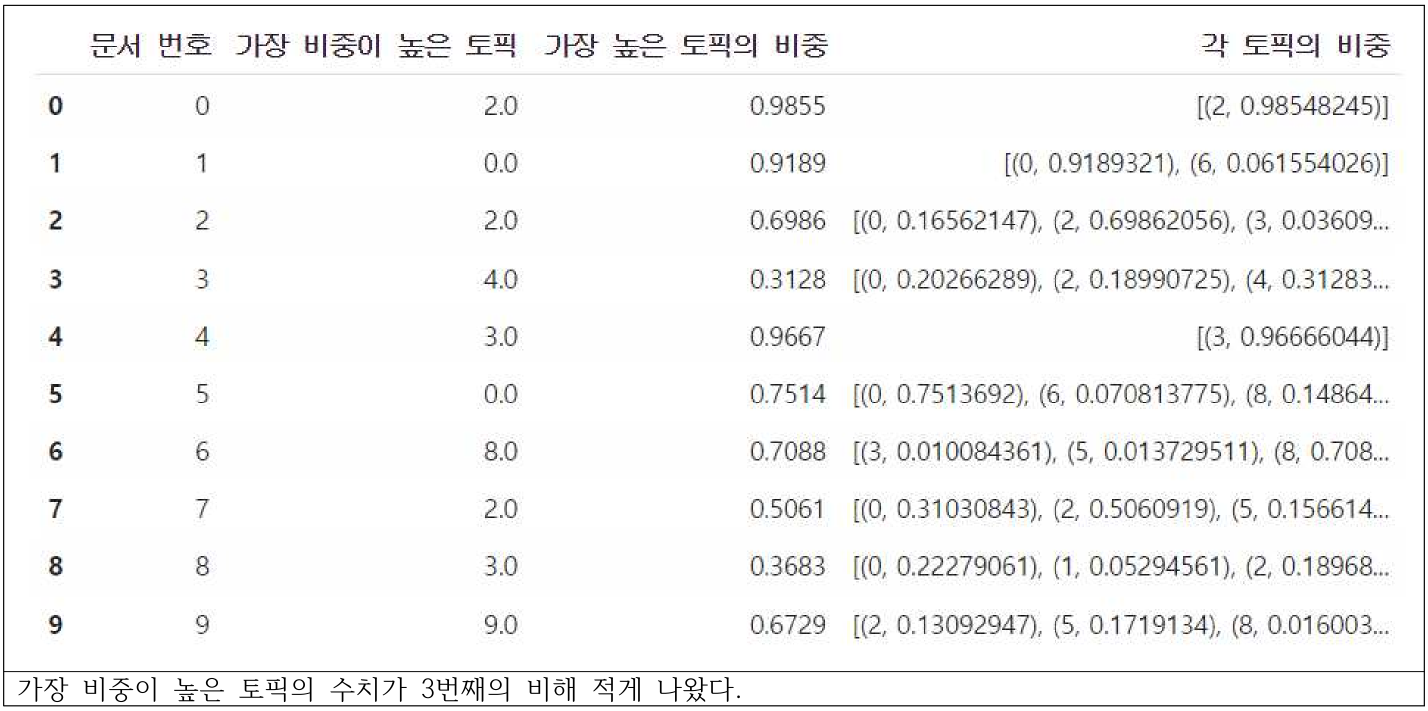
3. 4쪽에서 지정된 과제를 수행



```
topictable = make_topictable_per_doc(ldamodel, corpus)
topictable = topictable.reset_index() # 문서 번호을 의미하는 열(column)로 사용하기 위해서 인덱스 열을 ㅎ
topictable.columns = ['문서 번호', '가장 비중이 높은 토픽', '가장 높은 토픽의 비중', '각 토픽의 비중']
topictable[:10]
```

	문서 번호	가장 비중이 높은 토픽	가장 높은 토픽의 비중	각 토픽의 비중
0	0	19.0	0.4262	[(3, 0.21545184), (8, 0.34465116), (19, 0.4261...
1	1	8.0	0.5323	[(3, 0.027881343), (4, 0.25539014), (8, 0.5322...
2	2	8.0	0.3927	[(1, 0.04918807), (3, 0.35603106), (4, 0.14976...
3	3	16.0	0.5142	[(1, 0.20776743), (4, 0.18000905), (11, 0.0698...
4	4	17.0	0.6864	[(4, 0.2802799), (17, 0.68638676)]
5	5	8.0	0.3545	[(4, 0.13292517), (5, 0.21952607), (6, 0.19587...
6	6	5.0	0.6461	[(5, 0.64608824), (8, 0.04718778), (10, 0.0279...
7	7	8.0	0.3205	[(3, 0.14909668), (4, 0.20083946), (6, 0.08753...
8	8	18.0	0.3074	[(0, 0.034564245), (4, 0.18024746), (8, 0.2430...
9	9	11.0	0.4657	[(4, 0.29005527), (8, 0.07505698), (9, 0.10296...

4. 3번의 프로그램에서 토픽의 숫자를 10개로 바꾸고 이 때의 결과가 3번 결과와 어떻게 다른지 설명하라.



	문서 번호	가장 비중이 높은 토픽	가장 높은 토픽의 비중	각 토픽의 비중
0	0	2.0	0.9855	[(2, 0.98548245)]
1	1	0.0	0.9189	[(0, 0.9189321), (6, 0.061554026)]
2	2	2.0	0.6986	[(0, 0.16562147), (2, 0.69862056), (3, 0.03609...
3	3	4.0	0.3128	[(0, 0.20266289), (2, 0.18990725), (4, 0.31283...
4	4	3.0	0.9667	[(3, 0.96666044)]
5	5	0.0	0.7514	[(0, 0.7513692), (6, 0.070813775), (8, 0.14864...
6	6	8.0	0.7088	[(3, 0.010084361), (5, 0.013729511), (8, 0.708...
7	7	2.0	0.5061	[(0, 0.31030843), (2, 0.5060919), (5, 0.156614...
8	8	3.0	0.3683	[(0, 0.22279061), (1, 0.05294561), (2, 0.18968...
9	9	9.0	0.6729	[(2, 0.13092947), (5, 0.1719134), (8, 0.016003...

가장 비중이 높은 토픽의 수치가 3번째의 비해 적게 나왔다.