정보검색과데이터마이닝 추가 구현

주제: K-means clustering을 이용한 문서 분류 및 PCA를 이용한 시각화

1. K-means clustering

K-means clustering이란 데이터의 라벨 정보를 알지 못할 때, 비슷한 것끼리 묶어 k개의 cluster로 분류해주는 알고리즘이다. 어떤 k값이 최적의 k값일지 생각해보고, k값에 따른 분류 결과를 출력해볼 것이다.

```
def kmeans(k, points):
    prev_assignment = []

    centroids = points[np.random.choice(points.shape[0], replace=False, size=k)]  # 임의로 k개의 centroids 선택

    for epoch in range(10):
        assignments = [assign(p, centroids) for p in points]  # assign 함수 이용해서 각 점이 할당 될 centroid 계산
        centroids = compute_centroids(assignments, points, k)  # 새로운 centroids 계산

        if prev_assignment == assignments:
            break;
        prev_assignment = assignments

    cost = compute_cost(centroids, assignments, points)  # cost는 cosine distance의 합

    return assignments, centroids, cost
```

직접 구현한 kmeans 함수이다. 우선 centroids는 cluster의 중심점들의 집합이다. 우선 초기에는 k개의 중심점을 랜덤하게 생성한다. 그 이후 반복문을 돌며 assign 함수를 이용해 각 점이 할당될 centroid를 정하고, compute_centroids 함수를 이용해 적절한 새로운 centroids를 정한다. Cost는 compute_cost 함수를 이용하여 계산해주었다.

```python
def assign(point, centroids):
    max_cos_sim_centroids_idx = -1
    max_cos_sim = -2

    for i, c in enumerate(centroids):
        cos_sim = 0

        # 두 벡터 단위벡터로 만들기(0으로 나눈다는 경고 없애기 위해) --> 크기 1이니까 cos sim 구할때 따로 안나눠 줘도 됨
        point = point / np.sqrt(np.sum(point ** 2))
        c = c / np.sqrt(np.sum(c ** 2))

        cos_sim = np.sum(point * c)  # cluster 정하는 기준은 cosine similarity, 분모는 1이라서 생략

        if max_cos_sim < cos_sim:
            max_cos_sim_centroids_idx = i
            max_cos_sim = cos_sim

    return max_cos_sim_centroids_idx
```

위는 각 점이 할당될 centroid를 정해주는 함수이다. 이는 cosine similarity를 이용하여 가장 cosine similarity가 큰 centroid로 배정해준다.

```python
# 새로운 centroids 계산해주는 함수
def compute_centroids(assignments, points, k):
    clusters = [[] for _ in range(k)]

    # point를 해당 cluster 배열에 넣음
    for a, p in zip(assignments, points):
        clusters[a].append(p)

    return [np.mean(c, axis=0) for c in clusters]
```

위는 새로운 centroids를 지정해주는 함수이다. 위의 assign 함수에서 구한 값을 이용하여, 각 클러스터의 평균이 되는 점을 새로운 centroid라고 지정한다.

```python
# cosine distance의 합인 cost를 계산해주는 함수
def compute_cost(centroids, assignments, points):
    cost = 0

    for i, c in enumerate(centroids):
        for a, p in zip(assignments, points):
            if a != i:
                continue
            else:
                # 두 벡터 단위벡터 만들기
                c = c / np.sqrt(np.sum(c ** 2))
                p = p / np.sqrt(np.sum(p ** 2))

                cost += 1 - np.sum(c * p)  # cost는 cosine distance의 합, cos sim의 분모는 1이라 생략

    return cost
```

위는 cost를 계산해주는 compute_cost 함수이다. 각 클러스터의 centroid와 점들의 거리를 cosine distance를 이용하여 구하고, 이것들을 모두 더해 cost를 구하였다. 이 cost값을 적절하게 줄여주는 k를 찾아보도록 하겠다.

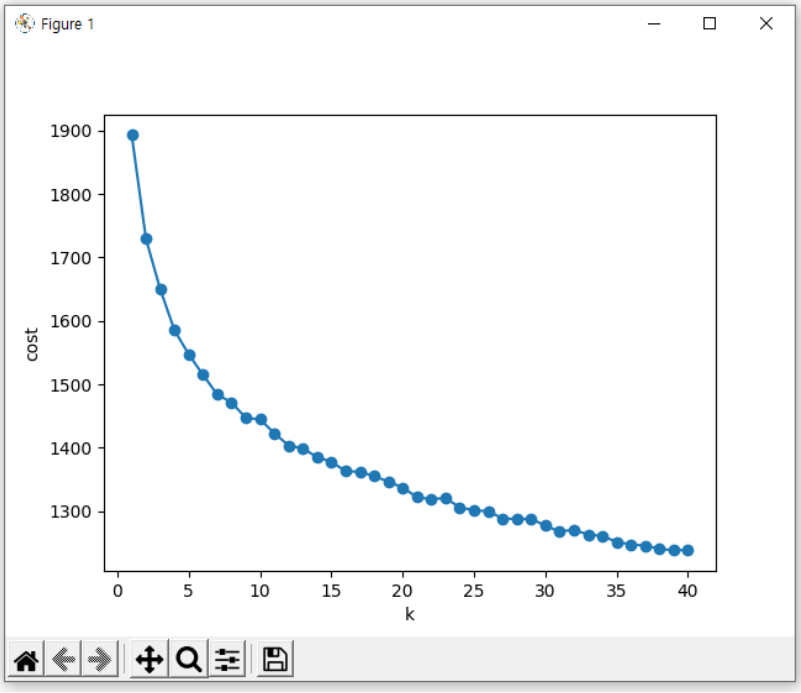
```
f_vector = open('document_vector.txt', 'r', encoding="utf-8")
vectors = f_vector.readlines()
f_vector.close()

M = []
for v in vectors:
    v = v.rstrip().split()
    M.append([float(vv) for vv in v])
```

우선 bigram으로 음절 토큰화 후, word2vec을 이용하여 만든 문서 벡터인 document_vector 파일을 불러왔다. 이를 다시 배열 형태로 만들어 M 배열이라고 하였다.

```
cost = 9 ** 9
cost_lst = []
declining = True  # cost 증감 현황 알려주는 용도

for k in range(1, 41):
    prev_cost = cost

    assignments, centroids, cost = kmeans(k, movie_points)
    cost_lst.append(cost)

    # 적절한 k 구하기: cost 감소가 15보다 크고 이전 cost가 갑자기 증가한 cost가 아닐 때만 optimal_k 업데이트
    if prev_cost - cost > 15 and declining == True:
        optimal_k = k

    if prev_cost < cost:
        declining = False
    else:
        declining = True

    print("if k = {}, cost = {}".format(k, cost))

print("\nOptimal k is {}".format(optimal_k))

X = [x for x in range(1, 41)]
Y = cost_lst
plt.plot(X,Y,'-o')
plt.xlabel("k")
plt.ylabel("cost")
plt.show()
```

위의 코드에서는 k 값을 1 에서 40 까지 변경해보며 최적의 k 값을 찾아보았다. 코드에서는 최적의 k 를 업데이트하는 조건을 prev_cost - cost > 15 **and** declining == **True** 와 같이 주었는데, 이는 적절하게 변경 가능하다. Elbow point 를 찾으려면 15 가 아닌 더 큰 값으로 주어야한다. 위의 코드의 실행 결과는 아래와 같다.

```
if k = 19, cost = 1346.4973848181824
if k = 20, cost = 1337.0269895139436
if k = 21, cost = 1322.0886543772483
if k = 22, cost = 1318.7661985203401
if k = 23, cost = 1320.709457020229
if k = 24, cost = 1305.7369524364483
if k = 25, cost = 1301.4459314640646
if k = 26, cost = 1299.590907068101
if k = 27, cost = 1288.4633833340627
if k = 28, cost = 1287.612461481752
if k = 29, cost = 1287.7616397913466
if k = 30, cost = 1276.9975122031158
if k = 31, cost = 1267.9924947629368
if k = 32, cost = 1270.4217137252115
if k = 33, cost = 1263.2088007369973
if k = 34, cost = 1260.7944080695188
if k = 35, cost = 1251.4344023345461
if k = 36, cost = 1247.7142886701201
if k = 37, cost = 1245.5727482061295
if k = 38, cost = 1240.7071827322145
if k = 39, cost = 1238.1993601524835
if k = 40, cost = 1239.0745965690921

Optimal k is 27
```



2. PCA

PCA는 고차원 데이터를 저차원 데이터로 차원 축소를 해주는 기법이다. 이를 이용하여 위의 K-means clustering 결과를 2차원으로 시각화 할 수 있다.

```
print(M.shape)
pca = PCA(n_components = 2)
pca.fit(M)
Zp = pca.transform(M)
print(Zp.shape)
```

```
(5000, 100)
(5000, 2)
```

M은 영화평 document vector의 앞부분 5000개만을 가져온 것이다. 이전에는 100차원 벡터였지만, PCA를 이용하여 차원 축소 후 2차원 데이터가 된 것을 볼 수 있다.

3. 실행 결과

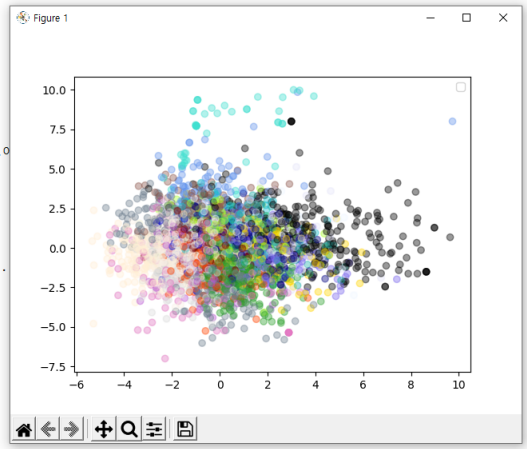
```
assignments, centroids, cost = kmeans(27, M) # cluster 27개(centroid 27개)로 설정

lst = [[] for _ in range(27)]
for i in range(len(assignments)):
    lst[assignments[i]-1].append(i)

f_all = open('naver_review.txt', 'r', encoding="utf-8")
f_label = open('label_train.txt', 'r', encoding="utf-8")

doc = f_all.readlines()
label = f_label.readlines()

f_all.close()
f_label.close()

for i in range(27):
    print("<", i+1, ">")
    for j in range(3):
        print(label[lst[i][j]], "  ", doc[lst[i][j]])

plt.scatter(Zp[:len(M),0], Zp[:len(M),1], c = [i for i in assignments], alpha=0.7)
plt.legend()
plt.show()
```

위에서 구현한 kmeans 함수를 이용하여 위에서 최적의 k라고 나온 27을 인자로 대입해보았다.

그 후, 각 클러스터 내의 영화평 출력 및 시각화 작업을 해주었다.

< k=27, centroid update 10회 >

```
< 1 >
-1
    카밀라벨 발연기

-1
    매우 실망.....

-1
    용가리 진짜짱짱맨이다ㅋ

< 2 >
1
    참 사람들 웃긴게 바스코가 이기면 락스코라고 까고바비가 이기면 아이돌이라고 깐다.그냥 까고싶어서 안달난것처럼 보인다

1
    보면서 웃지 않는 건 불가능하다

1
    이 영화가 왜 이렇게 저평가 받는지 모르겠다

< 3 >
-1
    주제는 좋은데 중반부터 지루하다

-1
    왕짜증.....아주 전개를 짬뽕으로 믹스했구나...음향만 무섭게하네..하아

-1
    솔직히 난 별루더라 시간낭비느낌


< 25 >
-1
    다 짤랐을꺼야. 그래서 납득할 수 없었던거야.. 그럴꺼야.. 꼭 그랬던걸꺼야..

-1
    이건 뭐냐? 우뢰매냐? ;;;

1
    근데 조미가 막문워 좋아한건가요??

< 26 >
-1
    별 반개도 아깝다 욕나온다 이응경 길용우 연기생활이몇년인지..정말 발로해도 그것보단 낫겟다 납치.감금만반복반복..ㅇ

-1
    졸쓰레기 진부하고말도안됌ㅋㅋ 아..시간아까워

-1
    '다 알바생인가 내용도 없고 무서운거도 없고 웃긴거도 하나도 없음 완전 별싱거운 영화.ㅇ.ㅇ내ㅇ시간 넘 아까움 .. .

< 27 >
1
    액션이 없는데도 재미 있는 몇안되는 영화

1
    이건 정말 깨알 캐스팅과 질퍽하지않은 산뜻한 내용구성이 잘 버무려진 깨알일드!!♥

-1
    재미없다 지루하고. 같은 음식 영화인데도 바베트의 만찬하고 넘 차이남....바베트의 만찬은 이야기도 있고 음식 보는재미도 있는데 ; 이건 볼게없다 음식도 별로 안나오고, 핀란드 풍경이라도 구경할랫는데 그것도 별로
```



위의 코드의 실행 결과이다. 이는 centroid를 update하는 작업을 10번 했을 때의 결과인데, 20번, 30번 했을 경우의 결과도 보도록 하겠다.

< k=27, centroid update 20회 >

< 1 >
-1
너무재밌었다그래서보는것을추천한다

1
재미있어요

1
전 좋아요

< 2 >
1
로큰롤!!!!!!!!!!!!!!

-1
그랜드 부다페스트 호텔과 시리즈 같네.. 캐스팅이 아까워~

-1
페이스 허거 같음ㅋㅋㅋㅋㅋ

< 3 >
1
백봉기 언제나오나요?

-1
이건 뭐냐? 우뢰매냐? ;;;

-1
클라라볼라고화신본거아닌데

< 24 >
-1
심심한영화.

-1
정말쓰레기영화입니다

1
인상적인 영화였다

< 25 >
-1
1%라도 기대했던 내가 죄인입니다 죄인입니다....

-1
난 우리영화를 사랑합니다....^^;

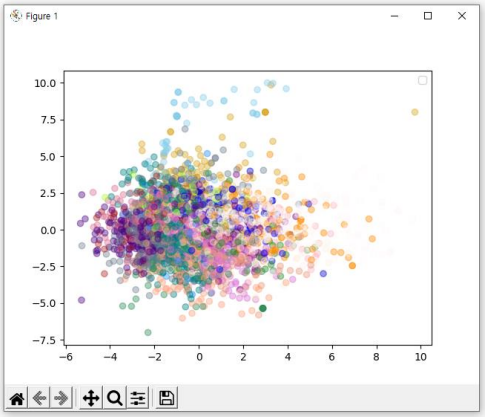
1
어내스트와 셀레스틴 완전 강추에요~ 정말 재밌습니다^^

< 26 >
1
강인피니트가짱이다.진짜짱이다♥

-1
매우 실망.....

1
대박

< 27 >
1
정말 맘에 들어요. 그래서 또 보고싶은데 또 보는 방법이 없네? >.. ㅜㅡ

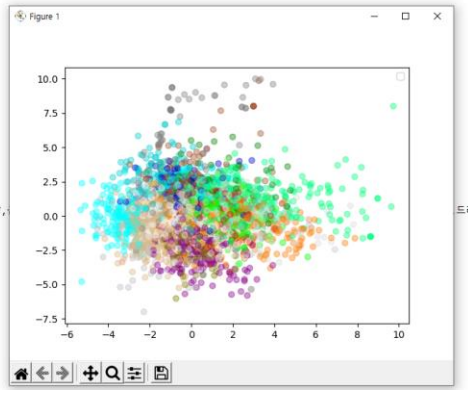


1
이틀만에 다 봤어요 재밌어요 근데 차 안에 물건 넣어 조작하려고 하면 차 안이 열려있다던지 집 안이 활짝 열려서 아무나 들어간다던가 문자를 조작하려고하면 비번이 안 걸려있고 ㅋㅋㅋ 그런 건 억지스러웠는데 그래도 내용 자체는 좋았

1
음악에 완전히 빠져서 볼 수 있었던 영화. 좀 산만하긴 하면데;;

< k=27, centroid update 30회 >

```
< 1 >
1
    아직도 이 드라마는 내인생의 최고!

-1
    최고

1
    단연 최고라고 할수있지

< 2 >
1
    걍인피니트가짱이다.진짜짱이다♥

1
    허허...원작가 정신나간 유령이라... 재미있겠네요!

-1
    뭐냐..시작하고 3분만에 나왔다. 리플릿 사진 보며 불안하더니만..

< 3 >
-1
    너무재밌었다그래서보는것을추천한다

1
    엄포스의 위력을 다시 한번 깨닫게 해준 적.남 꽃검사님도 연기 정말 좋았어요! 완전 명품드라마!

-1
    1%라도 기대했던 내가 죄인입니다 죄인입니다....

< 25 >
-1
    너무재밌었다그래서보는것을추천한다

-1
    1%라도 기대했던 내가 죄인입니다 죄인입니다....

1
    전 좋아요

< 26 >
-1
    포스터는 있어보이는데 관객은 114명이네

1
    오게두어라! 서리한이 굶주렸다!

1
    데너리스 타르 가르엔...나도 용의주인이 되고 싶다...누이랑,                                                드래곤(용)이 제일 멋지네(웃음)감독님 토르-2 다크 월드는 말아 잡수셨을지라도,기본 선방은

< 27 >
-1
    평범함속에 녹아든 평범한 일상. 조금 밋밋한게 흠.

-1
    내용전개가 너무나 느리다.....

-1
    볼만해;
```



아무래도 k값이 너무 커서 matplotlib의 색상 코드 개수의 한계로 완벽한 시각화는 불가능하다(색상 중복 생김). 그래도 centroid update 횟수가 10회일 때보다, 30회일 때 좀 더 깔끔하게 클러스터링 된 것 같다. 이제는 이전의 지도학습을 통해 classification하는 SVM과의 비교를 위해 k=2인 경우 클러스터링 결과를 확인해보도록 하겠다.

< k=2, centroid update 10회 >

< 1 >
-1
아 더빙.. 진짜 짜증나네요 목소리

1
흠...포스터보고 초딩영화줄....오버연기조차 가볍지 않구나

-1
너무재밋었다그래서보는것을추천한다

-1
교도소 이야기구먼 ..솔직히 재미는 없다..평점 조정

-1
별 반개도 아깝다 욕나온다 이응경 길용우 연기생활이몇년인지..정말 발로해도 그것보단 낫겟다 납치.감금만반복반복..이드라마는 가족도없다 연기못하는사람만모엿네

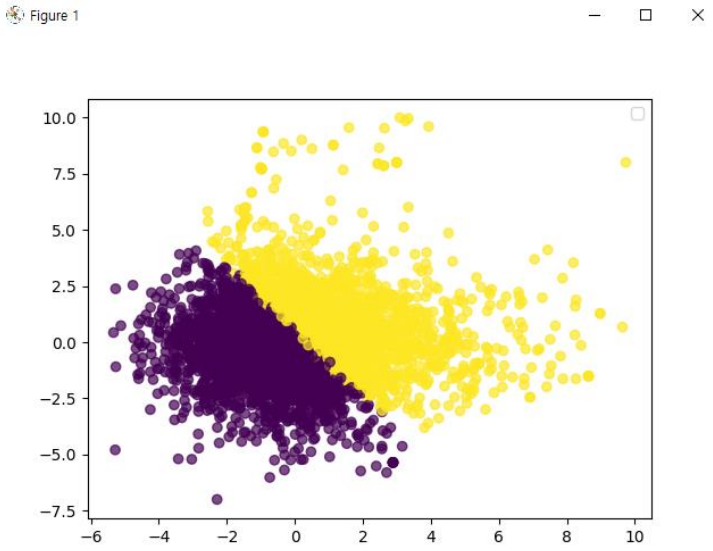
< 2 >
1
사이몬페그의 익살스런 연기가 돋보였던 영화!스파이더맨에서 늙어보이기만 했던 커스틴 던스트가 너무나도 이뻐보였다

-1
막 걸음마 뗀 3세부터 초등학교 1학년생인 8살용영화.ㅋㅋㅋ...별반개도 아까움.

-1
원작의 긴장감을 제대로 살려내지못했다.

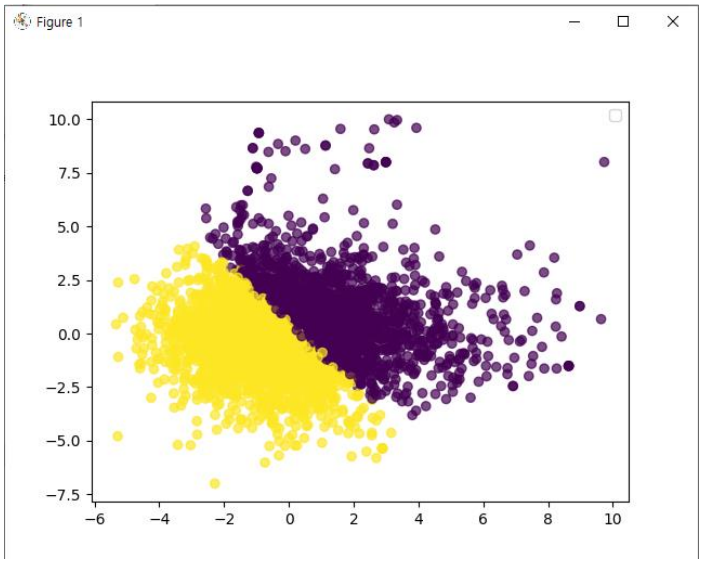
1
액션이 없는데도 재미 있는 몇안되는 영화

1
볼때마다 눈물나서 죽겠다90년대의 향수자극!!허진호는 감성절제멜로의 달인이다~



< k=2, centroid update 20회 >

```
< 1 >
1
    사이몬페그의 익살스런 연기가 돋보였던 영화!스파이더맨에서 늙어보이기만 했던 커스틴 던스트가 너무나도 이뻐보였다

-1
    막 걸음마 뗀 3세부터 초등학교 1학년생인 8살용영화.ㅋㅋㅋ...별반개도 아까움.

-1
    원작의 긴장감을 제대로 살려내지못했다.

-1
    별 반개도 아깝다 욕나온다 이응경 길용우 연기생활이몇년인지..정말 발로해도 그것보단 낫겟다 납치.감금만반복반복..이드라마는 가족도없다 연기못하는사람만모엿네

1
    액션이 없는데도 재미 있는 몇안되는 영화

< 2 >
-1
    아 더빙.. 진짜 짜증나네요 목소리

1
    흠...포스터보고 초딩영화줄....오버연기조차 가볍지 않구나

-1
    너무재밋었다그래서보는것을추천한다

-1
    교도소 이야기구먼 ..솔직히 재미는 없다..평점 조정

1
    왜케 평점이 낮은건데? 꽤 볼만한데.. 헐리우드식 화려함에만 너무 길들여져 있나?
```



k의 값이 작아 centroid update 횟수와 무관하게 비슷한 결과를 나타낸다. K의 값이 작아 잘 clustering 된 것을 볼 수 있다. 출력 결과의 라벨을 보았을 때 비지도학습인 k-means clustering의 특성이 잘 보인다. SVM과 다르게 라벨 값을 보고 분류하지 않는다는 것이다. 따라서 이를 활용해서 단순 긍/부정 분류가 아니라 k값을 크게 해서 비슷한 영화평끼리 묶어주는 역할을 할 수 있다. 따라서 이번엔 k=100으로 실행시켜보았다.

< k=100, centroid update 10회 >

< 10 >
1
ㄱ냥 매번 긴장되고 재밋음ㅠㅠ

1
몬스터 주식회사 3D 재밌게 봤다

1
재밌는데

< 13 >
1
넘 사랑스러운 영화다 ㅠㅠ 1보고 2 연이어 봤다~!! 넘 귀여워 ㅠㅠ♥♥

1
ㅠㅠ 슬픔

1
슬프다 ㅠㅠ

< 14 >
1
니노의 이중인격연기 두근거리네요♥

1
신화 화이팅 에릭 화이팅>< 모두 힘내요 !

1
정말 최고였어요 ㅠㅠ 이민정씨와 신하균씨 연기 너무 좋았어요!!

< 15 >
1
패션에 대한 열정! 안나 윈투어!

1
단순하면서 은은한 매력의 영화

1
자극적인 것에 익숙해진 현대인이 봐도 눈을 떼기 힘든 연출력.

< 16 >
1
재미있어요

1
전 좋아요

1
재밋네요 달팽이가 빨라서 더 재밌었어요

< 18 >
-1
이건좀 아니잖아...

-1
이게 21c영화냐? 90년도영화냐? 쓰래기들아

-1
ㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋ옛다 1점이나 처먹어랔ㅋㅋㅋㅋㅋㅋㅋㅋㅋ이게 7점급?이건 1점짜리. 내용도 결말도. 살다 살다 뭐 이딴...

< 19 >
-1
평점에속지마시길시간낭비 돈낭비임

-1
졸작

-1
절대비추.....

< 20 >
-1
교도소 이야기구먼 ..솔직히 재미는 없다..평점 조정

-1
평범함속에 녹아든 평범한 일상. 조금 밋밋한게 흠.

-1
진심 재미 없는데 너무 평점 높아서 화남;
< 40 >
-1
재미없다 지루하고. 같은 음식 영화인데도 바베트의 만찬하고 넘 차이남....바베트의 만찬은 이야기도 있고 음식 보는재미도 있는데 ;

-1
'다 알바생인가 내용도 없고 무서운거도 없고 웃긴거도 하나도 없음 완전 별싱거운 영화.ㅇ.ㅇ내ㅇ시간 넘 아까움 .. . 완전 낚임

-1
무섭지도 않았고 스토리도 ..——

< 41 >
1
재밋는뎅

1
요즘 재밌음!

1
재밌군

< 42 >
-1
졸쓰레기 진부하고말도안됌ㅋㅋ 아..시간아까워

-1
배우들 연기력이 아깝다. 별 한 개도 아깝다.

-1
배우가 아깝다 ,,

< 56 >
1
왜케 평점이 낮은건데? 꽤 볼만한데.. 헐리우드식 화려함에만 너무 길들여져 있나?

1
재밌는데 별점이 왜이리 낮은고

1
이 영화가 왜 이렇게 저평가 받는지 모르겠다

< 70 >
1
아직도 이 드라마는 내인생의 최고!

1
드라마 너무 재밌당

1
맛깔 나는 드라마.

< 76 >
-1
볼만해;

1
ㅋㅋ 조금은 유치했지만 왕조현 주윤발 이것만으로도 충분히 볼만했다.

-1
볼만 한데

< 77 >
-1
별루 였다..

-1
솔직히 난 별루더라 시간낭비느낌

-1
별로다.

< 80 >
1
연기 굿

1
굳굳

-1
굿 좋아

< 97 >
1
나름 괜찮은 작품입니다

1
고다미 괜찮음

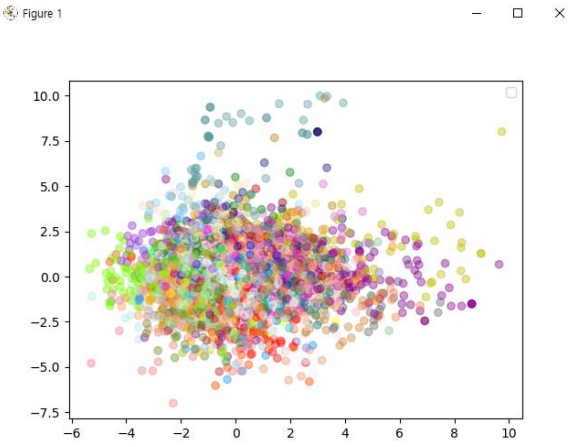
1
책과는 분명히 다른시선. 괜찮네요

< 98 >
-1
매우 실망.....

-1
정말 실망 스러웠음..

-1
.....더빙이 이상해요.....할머니는 월래익숙한 ㄷㅐ.....

K=100으로 클러스터링 한 경우 위의 출력 결과처럼 꽤나 유사한 영화평들을 잘 묶은 것을 볼 수 있다.



PCA로 차원 축소 후 시각화 한 결과이다. Matplotlib의 경우 나타낼 수 있는 색상에 한계가 있어 100개에 대하여 완벽히 분리된 시각화는 불가능하다.

지금까지 네이버 영화평 5000개에 대하여 K-means clustering을 여러 경우로 나누어 실행해보고, PCA를 이용하여 시각화 해보았다. 비지도학습의 특성상 긍/부정으로 클러스터링 하는 것보다 k의 값을 크게 해서 비슷한 영화평끼리 묶어주는 데에 좋은 성능을 나타냈다. 이를 활용한다면 역으로 영화평(라벨 없음)만을 가지고 라벨링을 할 수 있고(라벨링 범위는 1~k), 각 영화별로 이 라벨 정보를 수집하여 유사 영화 추천 시스템을 만들 수 있을 것 같다.