

정보와 지식 봄학기 프로젝트 과제



|  |  |
| --- | --- |
| **학 과 :** | **컴퓨터공학과** |
| **학 번 :** | **20171006** |
| **이 름 :** | **이찬영** |

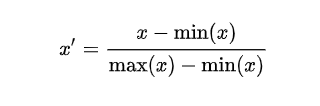
* K Means 알고리즘 설계

1. Null value 체크

: df.info()를 통해 data에 결측치가 있는지 체크한다.

1. Min max scaling

: 최소값을 0, 최대값을 1로하여 데이터 값의 범위를 0 ~ 1로 변환한다. Feature 마다 scale이 다르기 때문에 kmean 알고리즘 적용시 scale에 따라 중요도가 달라지기 때문에 min-max scaling으로 feature들의 범위를 같게 만들어준다.



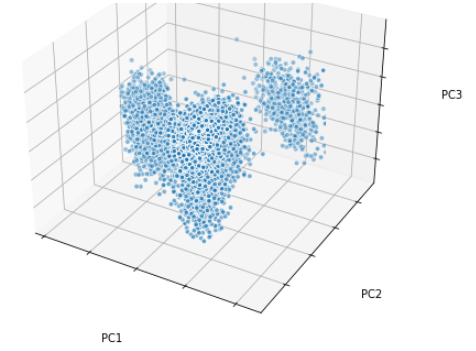
<min-max 정규화>

1. PCA

: features가 10개로 10차원 데이터이기 때문에 PCA를 통해 2차원, 3차원 데이터로 차원 축소를 해서 군집 비교를 하였다.

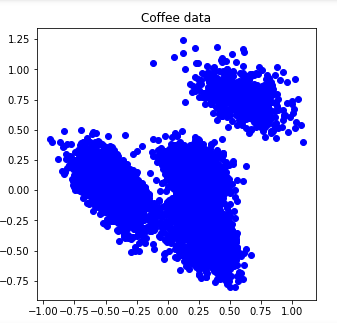
: sklearn의 패키지를 이용하여 간단하게 차원 축소를 진행했다.

3-1) 3차원으로 축소 후 결과



<3차원 축소 데이터 분포 그래프>

3-2) 2차원으로 축소 후 결과



<2차원 축소 데이터 분포 그래프>

1. K-means 알고리즘을 위한 함수

4-1) euclidean\_distance 함수

* feature의 수가 n ( n > 2)개로 많기 때문에 모든 feature와 center의 차를 제곱해서 더한 값의 제곱근을 구한다.
* 리스트에 data와 center 사이의 거리를 저장한다.
* Numba 라이브러리 활용을 통해 python for문 속도 향상을 시킨다.

텍스트이(가) 표시된 사진

자동 생성된 설명

<euclidean\_distance 함수>

4-2) get\_centroid 함수

* Data의 분류된 군집에 따라서 centoid를 다시 계산해주는 함수이다.
* 먼저 centroid를 계산할 군집의 수와 index를 체크하고, 해당하는 data를 저장한 뒤에 평균값을 통해 군집 별 centroid를 다시 설정해준다. Loss 값은 앞서 정의한 euclidean\_distance 함수로 구한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

<get\_centroid 함수>

5-3) get\_label 함수

* 리스트 내에서 최빈값을 예측값으로 결정한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

<get\_label 함수>

5-4) 반복을 통해 centroid값 refitting

* 초기에 data의 군집을 randint을 이용하여 임의로 분류한다.

(seed = 0 고정을 통해 random값은 다시 실행해도 동일함)

* get\_centorid 함수로 centroid들의 좌표를 찾는다.
* 찾은 centroid좌표와 data 사이의 거리를 Euclidean\_distance 함수를 통해 계산하고 이 거리 값으로 군집을 재배치한다.
* 다시 위 과정을 반복하면서 centroid 좌표를 재배치 한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

<Refitting Step>

5-5) Converges

* Refitting step을 반복할수록 거리의 합은 점점 감소하게 되고 일정 step이후에는 수렴에 가까워 지기 때문에 loss\_curve를 그려보며 적당한 반복 횟수를 찾았다.

5-6) plt\_centroid 함수

* centroid 값이 재배치되는 것을 그래프를 통해 시각화하였다.

5-7) get\_contingency\_table 함수

* 군집을 분류한 결과를 csv 형태의 분할표로 제작한다.

테이블이(가) 표시된 사진

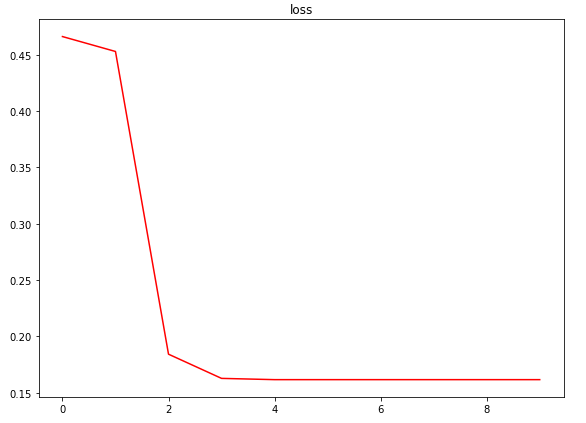
자동 생성된 설명

<contingency table>

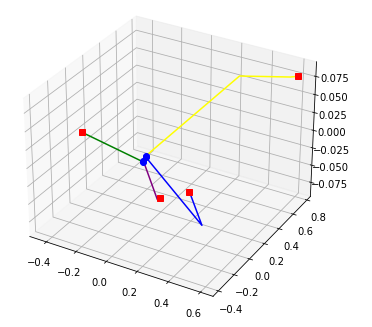
* 조건별 분류 결과

1. 3차원 축소, 4개의 군집으로 분류

1-1) loss 값 시각화



1-2) centroid 이동 좌표 시각화



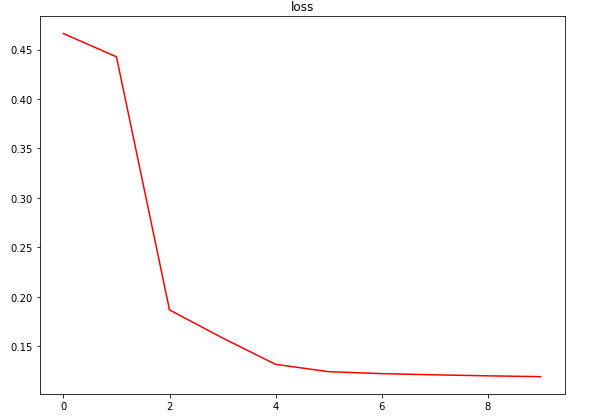
1-3) 분류 결과

명함, 봉투이(가) 표시된 사진

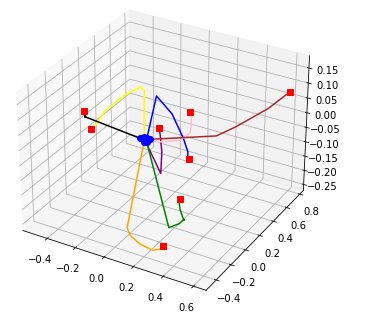
자동 생성된 설명

1. 3차원 축소, 8개의 군집으로 분류

2-1) loss 값 시각화



2-2) centroid 이동 좌표 시각화



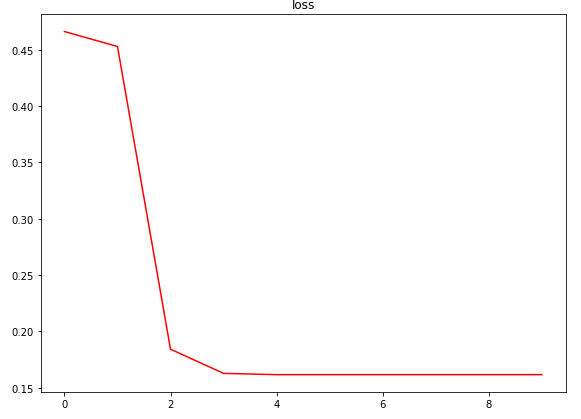
2-3) 분류 결과

문구, 명함, 봉투, 벡터그래픽이(가) 표시된 사진

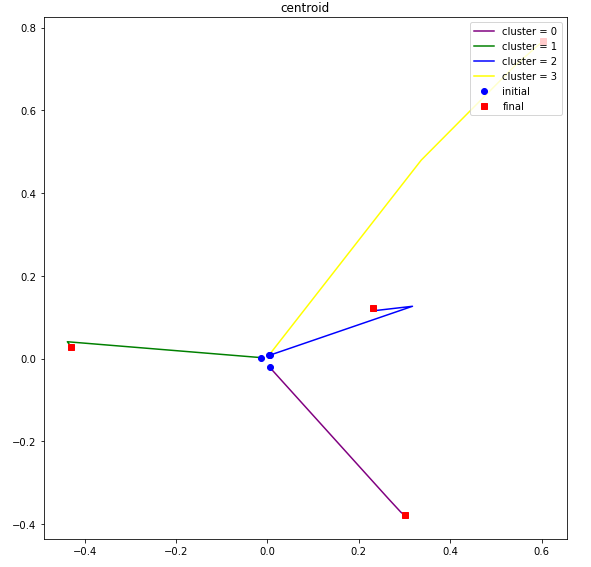
자동 생성된 설명

1. 2차원 축소, 4개의 군집으로 분류

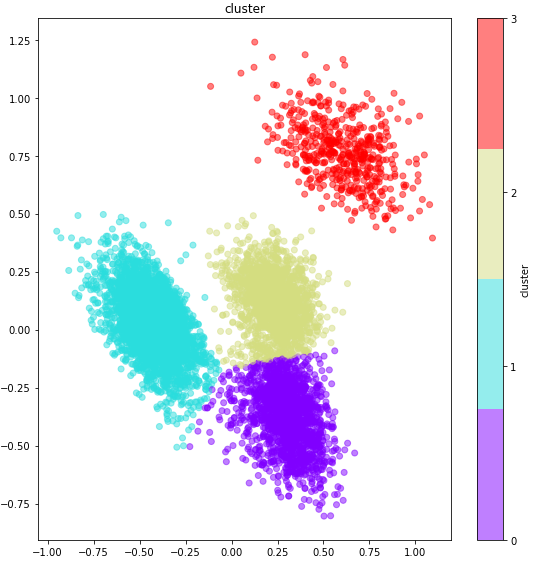
3-1) loss 값 시각화



3-2) centroid 이동 좌표 시각화

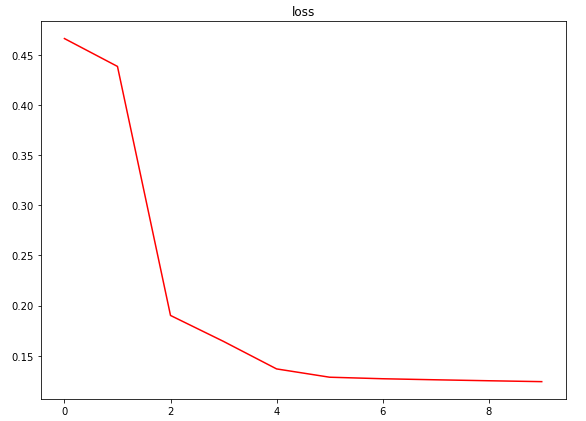


3-3) 분류 결과

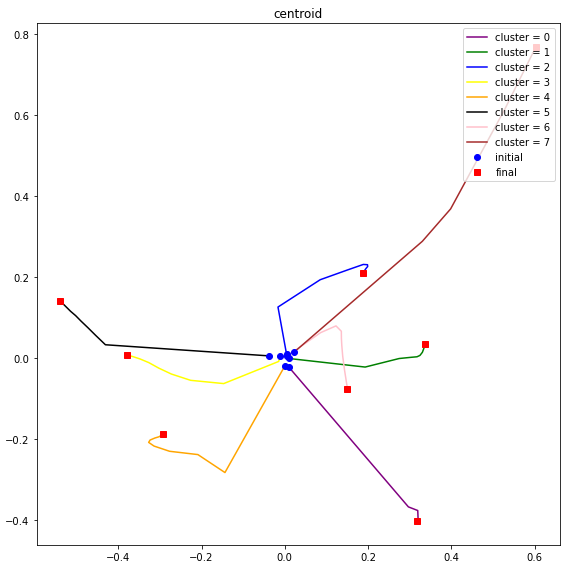


1. 2차원 축소, 8개의 군집으로 분류

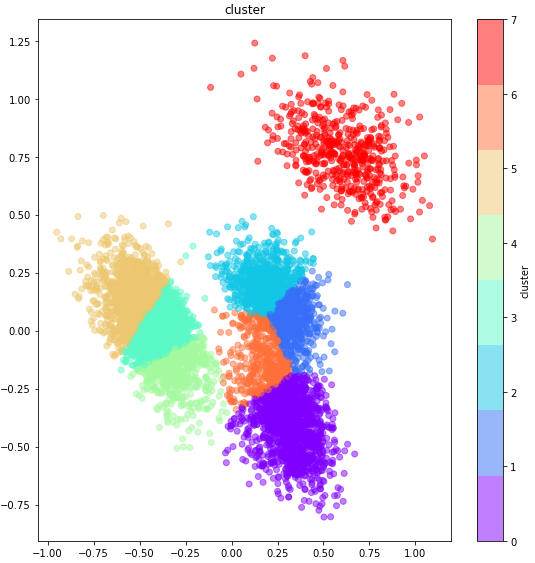
4-1) loss 값 시각화



4-2) centroid 이동 좌표 시각화



4-3) 분류 결과



* K means 알고리즘 실행 시간 측정
* Datetime library를 이용하여 시간 측정

Refitting step 반복문 시작부터 끝날 때 까지 시간을 체크해서 실행 시간으로 정했다.

1. 3차원 축소, 4개의 군집으로 분류



1. 3차원 축소, 8개의 군집으로 분류



1. 2차원 축소, 4개의 군집으로 분류



1. 2차원 축소, 8개의 군집으로 분류



* 분할표 설명

1. 20171006-1.csv 파일

: 3차원 축소, 4개의 군집으로 분류한 결과

1. 20171006-2.csv 파일

: 3차원 축소, 8개의 군집으로 분류한 결과

1. 20171006-3.csv 파일

: 2차원 축소, 4개의 군집으로 분류한 결과

1. 20171006-4.csv 파일

: 2차원 축소, 8개의 군집으로 분류한 결과

* 분할표 읽기 방법

: pd.read\_csv("20171006-1.csv", index\_col = 0)

* 사용 언어

: 파이썬

* 소스코드 실행 환경 및 실행 방법
* 실행 환경 : jupyter notebook
* 실행 방법 : kernel -> Restart & Run All 클릭
* 실행 파일 생성 & 실행 방법
* 실행 파일 실행으로 분할표 csv 파일을 편리하게 생성할 수 있다.

1. Ipynb 파일 .py파일로 변환

(그래프 plot하는 함수는 실행파일에 제외하였음.)

1. 클릭 (동일 경로에 coffee\_data.csv 파일이 있어야 실행가능)
2. 분할표 생성

텍스트이(가) 표시된 사진

자동 생성된 설명

<py.exe 실행 스크린샷>