**15강 k-평균 알고리즘 작동 방식을 이해하고 비지도 학습 모델 만들기**

* 텍스트, 스크린샷, 도표, 폰트이(가) 표시된 사진

  자동 생성된 설명타깃 없이(진정한 비지도학습)으로 분류하는 방법 : k평균 알고리즘

정확하게 구분되지 않은 이미지들을 사용하여 자동으로 분류할 수 있도록 하는 모델을 제작하는 것이 목표

* 군집 알고리즘으로 분류하는 것
  + 군집: 클러스터 / 과정: 클러스터링

k평균이 하는 일은 클러스터의 중심을 찾는 것 \*자동으로 ex) 각 픽셀의 평균값

그림, 텍스트, 스케치, 폰트이(가) 표시된 사진

자동 생성된 설명

k는 몇 개의 군집으로 나눌 지를 정해주는 하이퍼 파라미터

군집에 따라 클러스터 중심을 찾고 이 클러스터 중심과 과일간의 거리를 확인한다. 그 거리를 반복적으로 확인하는 과정에서 종류에 따라 모일 것이고 클러스터 중심이 더 이상 움직이지 않게 될 것이다. 🡪 알고리즘이 수렴함

텍스트, 스크린샷, 폰트, 라인이(가) 표시된 사진

자동 생성된 설명

\*사이킷런에는 여러 군집 알고리즘을 사용함

클러스터의 개수를 미리 지정

\*초기에 랜덤하게 중심을 지정함 \*클러스터의 중심을 센트러드라고 부르기도 함.

\*훈련타겟이 없으므로 과대적합, 과소적합 이랑 다른 방식으로 진행함 🡪 훈련데이터를 통쨰로 전달

 #3개의 array 각각에 111개, 98개, 91개로 사진들이 분류된 것을 알 수 있음

디폴트로 KMeans가 10번 반복해서 K-중심값을 이동함 \*n\_init 매개변수

출력해서텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명 분류가 잘 되었는 지 확인

Draw\_fruits()

Arr은 출력할 이미지의 배열

Rows는 몇 개의 열이 저장될 건지 기입

행이 1개밖에 없으면 열이 10개가 아니라 샘플의 개수를 따름 \*샘플개수가 10개 미만일경우

행과 열을 순회하면서 imshow함수로 이미지들을 그려냄 🡪 사과 9개는 빠졌지만 91개 골라진 사과는 정확하게 골라짐

**figsize=(cols\*ratio, rows\*ratio)** 부분에서 **cols**와 **rows**에 비례하여 전체 그림의 크기를 결정합니다. 그리고 **squeeze=False**는 서브플롯이 1행 또는 1열일 때에도 2D 배열로 반환하도록 지정하는 옵션입니다.

\*ratio는 가로세로의 길이 비율. 2로 하면 가로길이가 세로 길이의 두배가 됨

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명\*draw\_fruits(fruits[km.lables\_==1])에서의 labels값에 따라 출력되는 클러스터가 달라짐

다시 클러스터를 3차원의 배열로 바꿔서 draw\_fruits에 넣으면 3개의 중심값들이 출력될 것임

인덱스0: 사과, 1: 바나나, 2: 파인애플

Transform 매써드를 통해 새로운 샘플이 나타날 때 각 중심점까지의 거리를 통해 3개의 특성이 있는 벡터로 줄임 \*중심값까지의 거리를 특성으로 고려

Km.transform()을 통에 Fruits\_2D에 101번째 샘플을 골라서 넣었을 때 각 센트로이드까지의 거리를 출력할 수 있음. 🡪 3번째 클러스터의 센트로이드가 가장 가까우므로 predict()매써드를 쓰면 세번쨰 클러스트에 해당된다고 출력할 것이고 draw\_fruits를 통해 실제로 출력하면 파인애플이라는 것을 알 수 있음.

Km.n\_iter를 통해 세번의 반복만에 알고리즘이 종료됨을 알 수 있음 🡪 클러스터 중심이 더 이상 움직이지 않음

* k평균알고리즘에서 가장 최적의 k 찾기 \*타겟값이 정확히 몇 개 있을 지 사전에 알기 힘들기 떄문에

엘보우텍스트, 폰트, 라인, 그래프이(가) 표시된 사진

자동 생성된 설명 매쏘드를 사용하여 최적의 k값을 찾을 수 있음

우선 여러 개의 k값으로 나눠본 다음에 km.inertia\_값을 계산 \* 클러스터의 중심에서 클러스터에 속한 샘플들까지의 거리를 평균화한 것 🡪 inertia\_가 낮으면 조밀조밀하게 모여있는 것이고 그렇지 않으면 중심에서 벗어난 샘플들도 존재한다는 것

* 엘보우그림을 그리면 딱 꺽이는 부분이 나타나는데 이는 k값이 늘어나도 inertia값이 더 이상 줄어들지 않는 위치라는 것으로 인지할 수 있음 🡪 최적의 k값

구형의 클러스터를 만들 땐 k평균클러스터가 좋지만 그렇지 않을 때는 다른 알고리즘을 쓰는 게 좋음 \*또한 이미지가 아닌 다른 자료형태를 쓸 때는 표준화와 같은 전처리를 한 후 사용하는 것이 좋음 \*대게 지도학습을 이용한 후속작업을 또 진행해야 함 = 또 다른 k값이나 하이퍼 파라미터를 조정하는 등의 클러스터링 결과를 확인하고 보완하는 과정

**16강 주성분 분석: 차원축소 알고리즘**

* 다수의 이미지를 분류해야 할 경우 이미지의 차원을 축소하여 분류하기 용이한 형태로 만드는 것

차원도표, 텍스트, 스케치, 평면도이(가) 표시된 사진

자동 생성된 설명:

벡터: 벡터에 있는 원소의 개수를 의미

배열: 축을 의미

차원 축소: 벡터로서의 차원을 축소하는 것임

머신러닝 알고리즘에서 2차원배열의 특성의 개수를 줄이는 차원 축소임. 🡪 배열을 텐서라고 불리기도 함

결론: 차원축소 = 벡터의 특성의 개수를 줄이는 것임

* 도표, 라인이(가) 표시된 사진

  자동 생성된 설명주성분 분석(PCA)

훈련데이터가 산점도로 놓여져 있을 때 어느 한 방향으로 편향되어 퍼져 있는 경우 가장 많이 퍼져 있는 쪽을 찾는 것

* Because 그 쪽 방향이 데이터를 표현하는 데에 가장 적합함

훈련데이터를 두 축의 원점에 맞춰서 데이터를 가장 잘 표현하는 벡터를 찾는 것이 좋음 그리고 이것을 주성분이라고 일컬음. 🡪 PCA알고리즘이 이 주성분을 찾는 것임

\*주성분 차원도 = 원본공간차원 \*개수가 아닌 주성분 각각의 차원

여기에선 특성 2개중 주성분 1개를 찾아 두개의 특성을 하나의 특성으로 축소한 것임.

왼쪽 아래 4,2라는 데이터포인트를 찾은 벡터에 수직으로 투영하면 4.5로 투영할 수 있음 🡪 4,2를 4.5라는 하나의 특성으로 변환한 것임

주성분을 하나 더 찾는 방법: 원래 찾는 주성분의 수직인 방향으로 두번째 주성분을 찾음 🡪 결국엔 원본세트에 있는 차원의 개수만큼 찾을 수 있음 \*데이터가 독립적으로 퍼져 있는 방향을 찾는 것.

픽셀의텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명 개수가 100\*100일 때 pca를 사용하여 50개의 주성분만 찾도록 하는 것

훈련데이터를 통해 PCA를 훈련시키고 도출된 pca 주성분은 components\_에 들어가게 된다.

\*학습하여 도출한 데이터는 끝에 \_(underscore)가 붙음.

Reshape을 통해 다시 본래의 3차원 배열로 구분

* 50개가 행으로 늘어져 있고 열에는 10000개의 특성으로 표현된 것(2차원공간) = 10000개의 특성이 존재하는 2차원 공간을 가로지르는 벡터가 하나의 샘플임. 🡪 이것을 다시 100\*100 벡터의 형태로(이미지형태)로 만들어 학습한 결과물을 관찰 = draw\_fruits까지의 과정

Pca.transform(원본데이터의 2차원 배열 )을 활용하여 원본데이터를 주성분만을 사용하여 다시 표현할 있음 = 만개의 특성을 50개의 특성으로 축소하는 것

원본데이터를텍스트, 스크린샷, 폰트, 디자인이(가) 표시된 사진

자동 생성된 설명 50개의 주성분으로 축소하면서 당연히 손실되는 값들이 있을 것임

Inverse\_transform()을 통해 원래 데이터셋에 있던 크기만큼 복원(만개의 특성으로 다시 복원)

* Reshape을 사용하여 이미지로 출력해서 실제 원본데이터와 비교 🡺 원본이미지의 특성이 어느정도 유지되는 것을 볼 수 있음 \*분산을 가장 잘 나타내는(고유특성을 잘 보여주는) 주성분들로 하여금 추출한 후 복원을 한 것이기에 이러한 결과를 보임

설명된텍스트, 스크린샷, 라인, 폰트이(가) 표시된 사진

자동 생성된 설명 분산: 실질적 원본 데이터와 주성분을 토대로 원본데이터의 크기로 재구성한 데이터 간의 분산(차이)을 비교하는 것임

Pca 객체 훈련 후에 Explained\_variance\_ratio\_라는 속성이 저장되는데 이 속성은 50개의 배열(주성분)으로 이루어져 있으며 각 배열이 얼마만큼 분산을 잘 설명하는 지를 알려줌. 모두 합쳐서 출력하면 원래 훈련데이터의 분산을 몇 퍼센테지만큼 보존하고 있는 지 확인할 수 있음

🡪 그래프는 50개까지 각각의 주성분이 설명하고 있는 분산의 양을 나타냄. 10개까지만 해도 분산을 크게 설명하고 있다는 것을 알 수 있음

\*변환기로 사용할 수 있다는 것은 다른 모델과 연결해서 사용할 수 있다는 것을 의미하기도 함

* 분류기와 함께 사용하기

타겟을텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명 임의로 100개의 0,1,2로 구성

교차검증을 활용한 로지스틱회귀모델을 통해 사진 분류

\*시간은 1.83초가 걸림

Fruits\_pca를 활용하여 돌렸을 때 100퍼센트의 확률로 맞으며 0.039초밖에 걸리지 않은 것을 확인 할 수 있음

설명된 분산의 비율로 n\_components를 사용할 수 있음 🡪 1이하의 수를 매개변수값으로 넣으면 설명된 분산이 예를 들어 0.5가 될때까지만 주성분을 찾을 수 있음 = 원본 데이터의 50%의 분산만 설명할수 있으면 될 때의 주성분 개수를 찾아줘

N\_components\_ : 주어진 조건에 따라 찾은 주성분의 개수

위의 경우에서는 2개의 특성만을 뽑은 것이고 이 두개의 특성을 토대로 교차검증을 활용한 로지스틱회귀모델을 사용하면 0.99의 정확도로 분류함.

* 텍스트, 스크린샷, 폰트, 라인이(가) 표시된 사진

  자동 생성된 설명군집과 함께 사용하기

주성분을 추출한 데이터를 활용하여 clustering하기.

* Np.unique(km.labels\_ ~를 통해 클러스터링해서 나온 고유한 군집들을 array로 출력하고 return\_counts = True를 통해 각 군집 0,1,2번에 대한 총 샘플 개수를 함께 출력함.

텍스트, 스크린샷, 폰트, 도표이(가) 표시된 사진

자동 생성된 설명주성분 분석의 또 하나의 중요한 용도는 시각화를 용이하게 해준 다는 것이다.

3개의 차원이하로 바꿔준다면 시각화에 용이하게 만들 수 있다.

fruits\_pca가 2개의 차원을 가지고 있는 배열이므로 x,y 평면에 그릴 수 있음

for 루프에 따라 k.labels를 0,1,2로 돌아가면서 지정해 3개의 군집에 같은 과정을 반복시키고 2차원 배열인 fruits\_pca을 [:,0]과 [“,1]로 x,y에 따라 구분하여 그래프에 입력한다.

주성분을 통해 시각화 한 것에서 보다시피 사과와 파인애플 사이에 근접한 주성분들이 존재한다. 🡪 두 군집간의 오차값의 원인