

Python으로 머신러닝 입문



세션장: 구은아, 김은기



PCA

PCA 클래스 및 하이퍼파라미터

`sklearn.decomposition.PCA`

```
class sklearn.decomposition.PCA(n_components=None, *, copy=True, whiten=False, svd_solver='auto', tol=0.0,
iterated_power='auto', random_state=None)
```

[\[source\]](#)

<https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html>

파라미터 이름	설명
n_components	몇 개의 주성분으로 축소할 것인지 결정 디폴트는 None



LDA

LDA 클래스 및 하이퍼파라미터

`sklearn.discriminant_analysis.LinearDiscriminantAnalysis`

```
class sklearn.discriminant_analysis.LinearDiscriminantAnalysis(solver='svd', shrinkage=None, priors=None, n_components=None, store_covariance=False, tol=0.0001, covariance_estimator=None)
```

[source]

https://scikit-learn.org/stable/modules/generated/sklearn.discriminant_analysis.LinearDiscriminantAnalysis.html

파라미터 이름	설명
n_components	몇 개의 주성분으로 축소할 것인지 결정 디폴트는 None

SVD

SVD 클래스

```
linalg.svd(a, full_matrices=True, compute_uv=True, hermitian=False)
```

<https://numpy.org/doc/stable/reference/generated/numpy.linalg.svd.html>

Truncated SVD

```
scipy.sparse.linalg.svds(A, k=6, ncv=None, tol=0, which='LM', v0=None, maxiter=None,  
return_singular_vectors=True, solver='arpack') \[source\]
```

<https://numpy.org/doc/stable/reference/generated/numpy.linalg.svd.html>

sklearn.decomposition.TruncatedSVD

```
class sklearn.decomposition.TruncatedSVD(n_components=2, *, algorithm='randomized', n_iter=5, random_state=None, tol=0.0) \[source\]
```

<https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.TruncatedSVD.html>

NMF

NMF 클래스 및 하이퍼파라미터

`sklearn.decomposition.NMF`

```
class sklearn.decomposition.NMF(n_components=None, *, init='warn', solver='cd', beta_loss='frobenius', tol=0.0001, max_iter=200, random_state=None, alpha='deprecated', alpha_W=0.0, alpha_H='same', l1_ratio=0.0, verbose=0, shuffle=False, regularization='deprecated')
```

[source]

<https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.NMF.html>

파라미터 이름	설명
n_components	몇 개의 주성분으로 축소할 것인지 결정 디폴트는 None

과제 안내

과제: 신용카드 고객 연체 예측

	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_1	PAY_2	PAY_3	PAY_4	PAY_5	...	BILL_AMT4	BILL_AMT5	BILL_AMT6	PAY_AMT1
0	20000	2	2	1	24	2	2	-1	-1	-2	...	0	0	0	0
1	120000	2	2	2	26	-1	2	0	0	0	...	3272	3455	3261	0
2	90000	2	2	2	34	0	0	0	0	0	...	14331	14948	15549	1518

- 1) 지금까지 배웠던 분류기 중 하나를 사용하여 교차검증(cv=3) 평균 정확도를 측정해봅시다. (train, test 세트 분류)
- 2) n_components=2인 PCA를 만들어서 차원을 축소하고, 그 데이터를 산점도로 시각화해봅시다.
- 3) n_components를 다양하게 설정하여 PCA를 진행하고, 1번에서 사용한 분류기로 교차검증(cv=3) 평균 정확도를 측정해봅시다. n_components가 몇으로 설정되었을 때 평균 정확도가 가장 높나요? 그때 정확도는 원본 데이터의 평균 정확도와 얼마나 차이가 나나요?
- 4) 이 데이터가 왜 PCA로 축소되어도 좋은 성능을 보이는지 이유를 찾아봅시다.

수고하셨습니다!
과제 열심히 하시고 다음 주에 보어요~