파머완 2~3장: 사이킷런으로 시작하는 머신러닝&평가

상태	완료
담당자	 시현 이
마감일	@09/14/2025
작업 유형	개념정리 및 필사
설명	파이썬 머신러닝 완벽 가이드_개정2판_제2장 개념 정리+ 필사 링크
업데이트 시간	@September 15, 2025 4:09 AM

01. 사이킷런 소개와 특징

사이킷런(scikit-learn)

:가장 널리 사용되는 머신러닝 라이브러리. 다양한 알고리즘과 개발 프레임워크 및 API를 제공

02. 첫번째 머신러닝-붓꽃 품종 예측하기

지도학습(Supervised Learning): (先) 정답이 주어진 데이터 학습→ (後) new data의 정답을 예측

[개념 정리]

사이킷런 패키지 내 모듈명: sklearn

sklearn.datasets내의 모듈: 사이킷런에서 자체적으로 제공하는 dataset를 생성하는 모듈의 모임

sklearn.tree내 모듈: 트리 기반 ML 알고리즘을 구현한 클래스의 모임

sklearn.model_selection: 데이터를 학습/검증/예측 데이터로 분리하거나 최적의 하이퍼 파라미터*로 평가하기 위한 다양한 모듈의 모임

*하이퍼 파라미터: ML 알고리즘 별로 직접 입력하는 파라미터.

[실습]-붓꽃 품종 예측 프로세스

1. 데이터세트 분리: 학습/테스트 데이터로 분리
2. 모델 학습: 학습 데이터로 ML 알고리즘 적용해 모델 학습
3. 예측 수행: 테스트 데이터를 분류,예측
4. 평가: 예측값과 실제 결과값 비교해 모델의 성능 비교

03. 사이킷런의 기반 프레임워크 익히기

Estimator 이해 및 fit(), predict() 메서드

[Estimator 클래스]

:지도학습의 모든 알고리즘을 구현한 클래스

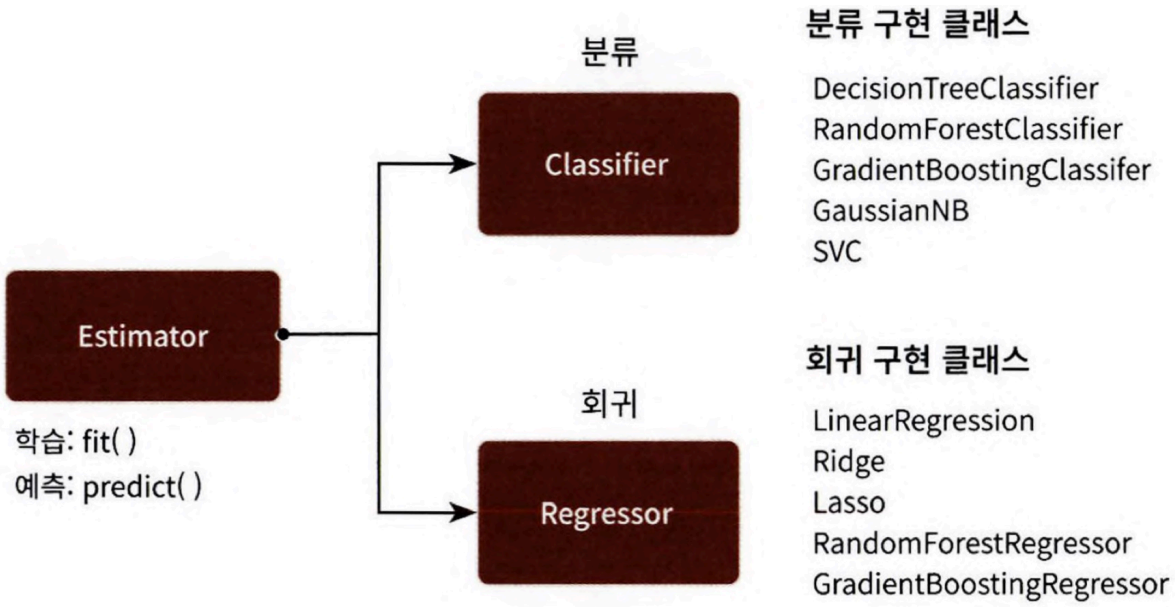
-분류 알고리즘 구현 클래스: Classifier

-회귀 알고리즘 구현 클래스: Regressor

cross_val_score()와 같은 evaluation 함수, GridSearchCV와 같은 하이퍼 파라미터 튜닝 지원 클래스

⇒ Estimator를 인자로 받음.

⇒Estimator의 fit()와 predict()를 호출해서 평가 or 하이퍼 파라미터 튜닝 수행



비지도학습(차원 축소, clustering, feature Extraction) 등 구현 클래스 역시 fit()과 transform()을 적용

⇒여기서 fit() : 사전 구조 맞추는 작업. 입력 데이터의 형태에 맞춰 데이터 변환

⇒fit() 이후, (차원변환, clustering, feature Extraction) 실제 작업은 transform()으로 수행.

fit() + transform() == fit_transform()

(약간의 차이 존재 주의)

사이킷런 주요 모듈

분류	모듈명	설명
예제 데이터	sklearn.datasets	사이킷런에 내장돼 예제로 제공하는 데이터 세트
피처 처리	sklearn.preprocessing	데이터 전처리에 필요한 다양한 가공 기능
	sklearn.feature_selection	영향이 큰 feature 우선순위대로 셀렉션 작업을 수행하는 다양한 기능 제공

	sklearn.feature_extraction	텍스트 데이터나 이미지 데이터의 벡터화된 피를 추출하는데 사용됨.//텍스트 데이터에서 Count Vectorizer나 Tf-Idf Vectorizer 등을 생성하는 기능 제공. 텍스트 데이터의 피처추출: sklearn.feature_extraction.text/이미지 데이터의 피처추출: sklearn.feature_extraction.image 모듈
피처 처리&차원 축소	sklearn.decomposition	차원 축소와 관련한 알고리즘을 지원하는 모듈. PCA, NMF, Truncated SVD 등을 통해 차원 축소 기능 수행
데이터 분리,검증 &파라미터 튜닝	sklearn.model_selection	교차 검증을 위한 학습용/테스트용 분리. Grid Search로 최적 파라미터 추출 등의 API 제공
평가	sklearn.metrics	분류, 회귀, 클러스터링, 페어와이즈(Pairwise)에 대한 다양한 성능 측정 방법 제공.//Accuracy, Precision, Recall, ROC-AUC, RMSE 등 제공
ML 알고리즘	sklearn.ensemble	앙상블 알고리즘 제공/ Random-forest, adaboost, gradient boosting 등 제공
	sklearn.linear_model	주로 선형 회귀, 릿지(Ridge), 라쏘(Lasso) 및 로지스틱 회귀 등 회귀 관련 알고리즘을 지원.SGD(Stochastic Gradient Descent) 관련 알고리즘도 제공.
	sklearn.naive_bayes	naive bayes 알고지금 제공./가우시안 NB, 다항 분포 NB
	sklearn.neighbors	최근접 이웃 알고리즘 제공.K-NN 등
	sklearn.svm	서포트 벡터 머신 알고리즘 제공
	sklearn.tree	의사 결정 트리 알고리즘 제공
	sklearn.cluster	비지도 클러스터링 알고지즘 제공(k-평균, 계층형, DBSCAN 등)
유틸리티	sklearn.pipeline	피처 처리 등의 변환과 ML 알고리즘 학습. 예측 등을 함께 묶어서 실행할

		수 있는 유틸리티 제공

내장된 예제 데이터 세트

API 명	설명
datasets.load_boston()	회귀 용도-미국 보스턴 집 feature, price data set
datasets.load_breast_cancer()	분류 용도-위스콘신 유방암 feature, 악/음성 레이블 dataset
datasets.load_diabetes()	회귀 용도-당뇨 dataset
datasets.load_digits()	분류 용도-0~9까지 숫자의 이미지 픽셀 dataset
datasets.load_iris()	분류 용도-붓꽃에 대한 feature dataset

fetch 계열의 명령

:fetch 명령어-외부 데이터셋을 불러오는 함수

:인터넷에서 데이터셋을 다운로드하고 로컬에 캐시한 뒤, 머신러닝 모델 학습에 쓸 수 있도록 파이썬 객체 형태로 반환

fetch 함수 종류	설명
fetch_covtype()	회귀 분석용 토지 조사 자료
fetch_20newsgroups()	뉴스 그룹 텍스트 자료
fetch_olivetti_faces()	얼굴 이미지 자료
fetch_lfw_people()	얼굴 이미지 자료
fetch_rcv1()	로이터 뉴스 말뭉치
fetch_mldata()	ML웹사이트에서 다운로드

분류와 클러스터링을 위한 표본 데이터 생성기

APIㅁ명	설명
datasets.make_classifications()	분류를 위한 dataset 생성. 노이즈 효과를 위한 데이터를 무작위로 생성
datasets.make_blobs()	클러스터링을 위한 dataset 생성.

연습용 예제 데이터 구성 요소

:일반적으로 딕셔너리 형태

키	설명	type
data	feature의 dataset	ndarray
target	분류-레이블 값/회귀-숫자 결과값 데이터세트	ndarray
target_names	개별 레이블의 이름	list
feature_names	feature의 이름	list
DESCR	데이터 세트에 대한 설명, 각 feature의 설명	String

04. Model selection 모듈 소개

model_selection 모듈

: 학습/테스트 데이터 세트 분리 or 교차 검증 분할 및 평가 / Estimator의 하이퍼 파라미터 튜닝하기 위한 다양한 함수와 클래스 제공

교차검증

:학습/테스트 데이터를 분리하면 과적합(Overfitting)에 취약한 약점을 가질 수 있음.

*과적합: 모델이 학습 데이터에만 과도하게 최적화돼, 실제 예측을 수행할 경우 예측 성능이 떨어지는 것

고정된 학습/테스트 데이터 사용 시, 해당 테스트 데이터에만 과적합되는 경우 발생.

⇒ 이를 해결하기 위해 교차검증이 필요

*교차검증: 데이터 편중(특정 알고리즘에 최적화된 데이터를 선별)을 막기 위해 별도의 여러 세트로 구성된 학습/검증 데이터 세트에서 학습과 평가를 수행

ML 모델 성능평가 ⇒ 1. 교차검증 기반 1차 평가/ 2. 테스트 데이터에 적용

데이터세트 세분화⇒ 학습, 검증/ 테스트

K 폴드 교차검증

: k 개의 데이터 폴드 세트를 만들어서 k번 학습과 검증 평가를 반복적으로 수행하는 방법

방법:

1. 데이터 세트 k등분
2. 첫번째 반복⇒ 1~4번째 등분을 학습 데이터 세트로, 5번째 등분을 검증 데이터 세트로 설정
3. 학습 데이터 세트에서 학습 수행, 검증 데이터 세트에서 평가 수행
4. 위와 같은 방식을 총 k번 반복.(첫번째 반복에서 검증 데이터 세트로 설정한 등분을 제외한 다른 등분을 검증 데이터 세트로 설정)
5. k개의 예측 평가를 구했으면 이를 평균해서 k 폴드 평가 결과로 반영.

Stratified K 폴드

: 불균형한 분포도를 가진 레이블(결정 클래스) 데이터 집합을 위한 k 폴드 방식.

:학습/테스트 데이터가 제대로 분배될 수 있도록 함.

:Classification 에서의 교차검증에서 Stratified K 폴드로 분할돼야 함.

→회귀의 결정값은 이산값X, 연속값이기 때문에 결정값 별로 분포하는 Stratified K 폴드가 의미 없음. 따라서, Regression에서는 Stratified K 폴드가 지원되지 않음.

05.데이터 전처리

결손값(Null || NaN)

1)Null 값이 얼마 없을 경우,

→feature의 mean 값으로 대체

2)Null 값이 대부분

→해당 feature drop

3)Null 값이 일정 수준 이상

→중요도가 높은 feature이고 feature mean 값으로 대체할 경우, 예측 왜곡이 심하다면 정밀한 대체 값을 선정해야함

숫자형 변환

ML 알고리즘은 문자열 값을 입력값으로 허용X

'문자열 값 —인코딩—> 숫자형'으로 변환이 필요함

[문자열 feature]

1) 카테고리형 피처: 코드값. 범주형 feature

2)텍스트형 피처: 단순히 데이터 로우를 식별하는 용도로 사용됨. 피처 벡터화 등의 기법으로 벡터화 or 불필요한 feature라면 삭제

데이터 인코딩

인코딩 방식

1)레이블 인코딩(Label encoding)

:카테고리 피처를 코드형 숫자 값으로 변환(ex- TV: 1, 냉장고:2 ....)

*'01', '02'는 문자열이므로 1,2,와 같은 숫자형으로 값 변환

주의)

일괄적인 숫자값으로 변환되면서 예측 성능이 떨어질 수 있음

⇒ TV:1 , 냉장고: 2가 됐을 때, ML 알고리즘에서 2에 가중치를 더 부여할 수도 있음(숫자에 크고 작음이 존재하기 때문)

⇒이때문에, 선형 회귀와 같은 ML 알고리즘에는 적용X

2)원-핫 인코딩(One Hot encoding)

: 피처값의 유형에 따라 새로운 피처를 추가해 고유 값에 해당하는 칼럼에만 1을 표기. 나머지 칼럼은 0 표시

→ 행 형태의 피처 고윳값을 열 행태로 차원 변환.

(레이블 인코딩의 단점 보완)

원본 데이터

상품 분류
TV
냉장고
전자레인지
컴퓨터
선풍기
선풍기
믹서
믹서

원-핫 인코딩

상품분류_ TV	상품분류_ 냉장고	상품분류_ 믹서	상품분류_ 선풍기	상품분류_ 전자레인지	상품분류_ 컴퓨터
1	0	0	0	0	0
0	1	0	0	0	0
0	0	0	0	1	0
0	0	0	0	0	1
0	0	0	1	0	0
0	0	0	1	0	0
0	0	1	0	0	0
0	0	1	0	0	0

피처 스케일링과 정규화

피처 스케일링(feature scaling)

: 서로 다른 변수의 값 범위를 일정한 수준으로 맞추는 작업

ex) 표준화(Standardization), 정규화(Normalization)

1)표준화

:데이터의 피처 각각이 평균이 0, 분산이 1인 가우시안 정규분포를 가진 값으로 변환하는 것.

xi_new = 표준화를 통해 변환될 피처 x의 새로운 i번째 데이터

$$x_i\_new = \frac{x_i - mean(x)}{stdev(x)}$$

2)정규화

:서로 다른 피처의 크기를 통일하기 위해 크기를 변환해주는 개념. 개별 데이터의 크기를 통일된 단위로 변경하는 것.

$$x_i\_new = \frac{x_i - \min(x)}{\max(x) - \min(x)}$$

2-1) Normalizer 모듈(사이킷런)

: 개별 벡터의 크기를 맞추기 위해 변환. (선형 대수의 정규화 개념)

$$x_i\_new = \frac{x_i}{\sqrt{x_i^2 + y_i^2 + z_i^2}}$$

개별 벡터를 모든 피처 벡터의 크기로 나눠줌.

⇒ 일반적인 표준화,정규화를 피처 스케일링으로 통칭/선형대수 개념의 정규화를 벡터 정규화로 지칭

[제 3장-평가]

머신러닝 프로세스: 데이터 가공/변환 → 모델 학습/예측 → 평가

성능 평가 지표(Evaluation Metric)

1. 회귀

:실제값-예측값 = 오차 평균값에 기반

ex) 오차의 절댓값→ 평균 오차

2. 분류

:결정 클래스 값 종류의 유형에 따라 긍정/부정과 같은 2개의 결과값만을 가지는 '이진 분류' or 여러 개의 결정 클래스 값을 가지는 '멀티분류'로 나눌 수 있음.

: 1)정확도(accuracy) 2)오차행렬(Confusion Matrix) 3)정밀도(Precision) 4)재현율(Recall)

5)F1-score 6) ROC AUC

⇒이진/멀티 분류에 모두 적용됨. but 이진 분류에서 강조하는 지표들

01.정확도

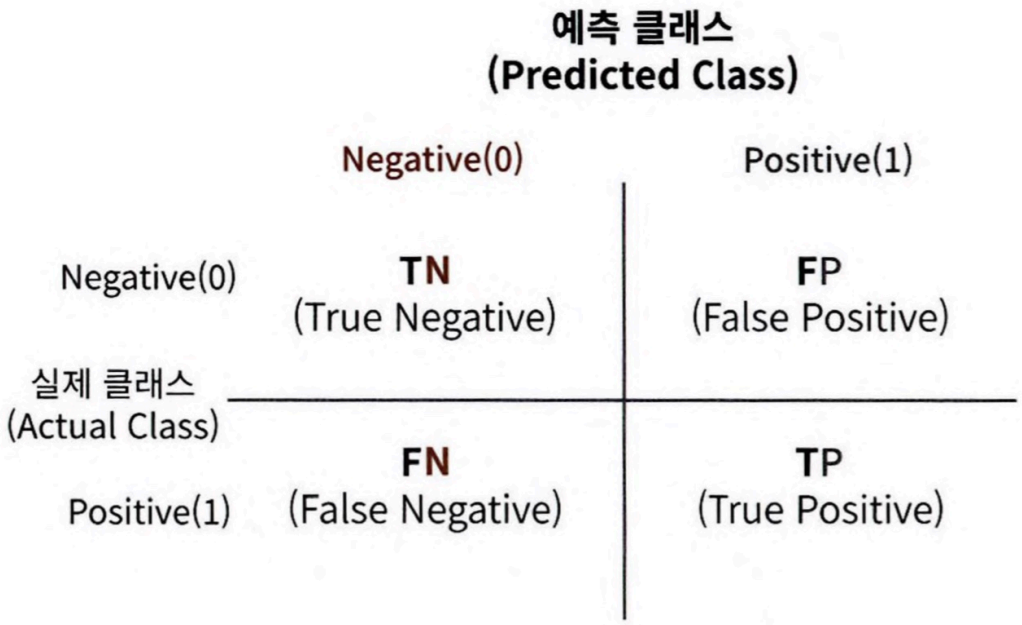
: 실제 데이터에서 예측 데이터가 얼마나 같은지를 판단

정확도 = (예측 결과가 동일한 데이터 건수)/(전체 예측 데이터 건수)

⇒이진 분류의 경우, 데이터 구성에 따라 ML 모델의 성능 왜곡 가능

02. 오차행렬

:이진 분류에서 성능 지표로 활용됨. 학습된 분류 모델이 예측을 수행하며 얼마나 헷갈리고 있는지 함께 보여주는 지표.



*TN: 예측값을 Negative 값 0으로 예측/ 실제값 역시 Negetive 값 0

*FP: 예측값을 Positive 값 1로 예측/ 실제값은 Negetive 값 0

*FN: 예측값을 Negative 값 0으로 예측/실제값은 Positive 값 1

*TP: 예측값을 Posiviet 값 1로 예측/ 실제값 역시 Positive 값 1

이 값을 조합해 classifier의 성능을 측정할 수 있는 주요 지표인 정확도, 정밀도, 재현율 값을 알 수 있음

→ 정확도 = 예측결과와 실제값이 동일한 건수/전체 데이터 수 = (TN + TP)/(TN+FP+FN+TP)

03. 정밀도와 재현율

: Positive 데이터 세트의 예측 성능에 초점을 맞춘 평가지표

1)정밀도 = TP/(FP+TP) = 예측을 Positive로 한 대상 중에 실제값이 Positive로 일치한 데이터 비율

→ 실제 Negative 음성인 데이터 예측을 Positive 양성으로 잘못 판단하게 되면 업무상 큰 영향이 발생하는 경우

ex) 스팸메일 여부 판단.

→FP를 낮추는데 초점

2)재현율 = TP/(FN +TP) = 실제값이 Positive인 대상 중에 예측과 실제값이 Positive로 일치한 데이터 비율

→실제 Positive 양성 데이터를 Negative로 잘못 판단하면 업무상 큰 영향이 발생하는 경우 사용

ex) 암환자 진단, 금융 사기 적발 모델

→FN을 낮추는데 초점

FN + TP ⇒ 실제값이 Positive 한 모든 데이터 건수

TP ⇒ 민감도 or TPR(Treu Positive Rate)라고 부름

정밀도/재현율 트레이드오프

:임계값(Threshold)을 조정해 정밀도 또는 재현율의 수치 높일 수 있BUT, 정밀도와 재현율은 '상호 보완적인 평가 지표' ⇒ 한쪽이 높아지면 다른 한 수치는 떨어짐.

=⇒ 정밀도/재현율의 트레이드오프(Trade-off)

사이킷런의 분류 알고리즘은 예측 데이터가 특정 레이블에 속하는지를 계산하기 위해 먼저 개별 레이블별로 결정 확률을 구함. 이후, 예측 확률이 큰 레이블값으로 예측

개별 데이터별로 예측 확률을 반환하는 메세드 : **predict_proba()**

→학습이 완료된 사이킷런 Classifier 객체에서 호출이 가능. 파라미터로 테스트 피처 데이터 세트 입력.

테스트 피처 레코드의 개별 클래스 '예측 확률'을 반환함

*predict()와 유사, but 반환 결과가 예측 결과 클래스값X. 예측 확률 결과

입력 파라미터	테스트 피처 데이터 세트
반환값	개별 클래스의 예측 확률을 ndarray mxn(m: 입력값의 레코드 수, n: 클래스 값 유형) 형태로 반환.

임계값 변화에 따른 평가지표 값 알아보는 get_eval_by_threshold()함수와 유사한:
precision_recall_curve() API

입력 파라미터	y_true: 실제 클래스값 배열 (배열 크기 = [데이터 건수])
	probas_pred: Positive 칼럼의 예측 확률 배열(배열 크기 = [데이터 건수])
반환 값	정밀도: 임곗값별 정밀도 값을 배열로 반환
	재현율: 임곗값별 재현율 값을 배열로 반환

일반적으로 0.11~0.95 정도의 임곗값을 담은 넘파이 ndarray와 이 임곗값에 해당하는 정밀도 및 재현율 값을 담은 ndarray를 반환

정밀도와 재현율의 맹점

:두개의 수치를 상호보완할 수 있는 수준에서 적용돼야함.

1)정밀도가 100%되는 법%되는 버

: 확실한 기준이 되는 경우만 Positive로 예측.나머지는 모두 Negative로 예측

2)재현율 100%가 되는 법

: 모든 환자를 Positive로 예측

⇒어느 한쪽 수치만 참조하면 수치 조작이 가능

04.F1-Score

: 정밀도 + 재현율 결합한 지표

$$F1 = \frac{2}{\frac{1}{recall} + \frac{1}{precision}} = 2 * \frac{precision * recall}{precision + recall}$$

정밀도와 재현율이 중용의 수치를 나타낼 때 상대적으로 높은 값을 나타냄.

f1_score() API

05.ROC 곡선과 AUC

ROC곡선(Receiver Operation Characterisitic Curve)

: 수신자 판단 곡선. FPR(False Positive Rate)이 변할 때, TPR(True Positive Rate)이 어떻게 변하는지를 나타내는 곡선(FPR-X축. TPR-Y축)

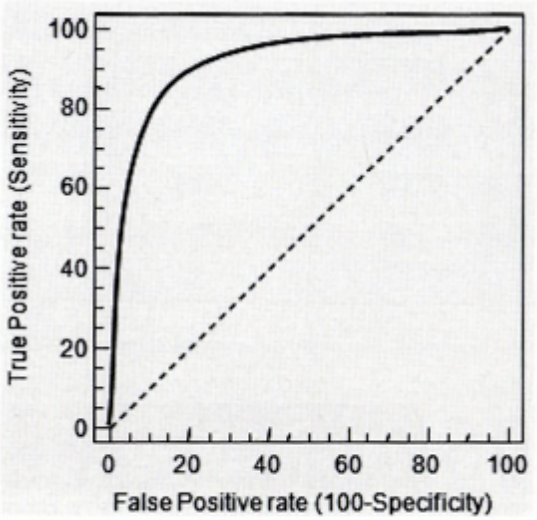
→ TPR == 재현율 ==TP/(FN+TP) == 민감도

TNR(True Negative Rate) == 특이성

→특이성: 실제값 Negative가 정확히 예측돼야하는 수준

: TNR = TN/(FP+TN)

FPR = 1-TNR



〈 ROC 곡선 예시 〉

⇒ROC 곡선의 예

곡선이 ROC 직선에 가까울 수록 성능이 떨어지고, 멀어질수록 성능이 뛰어남.

FPR의 변화값에 대한 TPR의 변화값을 나타낸다.

1)FPR 0~1까지 변경하는 방법

: 분류 결정 임곗값(Positive 예측값 결정하는 확률의 기준)을 1로 지정.→데이터를 positive 로 예측할 수 없음

2)FPR을 1로 만드는 방법

TN을 0으로 만들기. → 분류 결정 임곗값을 0으로 지정.→데이터를 negative로 예측 할 수 없음

ROC 곡선을 구하는: **roc_curve()** API

입력 파라미터	y_true: 실제 클래스 값 array(array shape = [데이터 건수])
	y_score: predict_proba()의 반환값 array에서 Positive 칼럼의 예측 확률이 보통 사용됨. array.shape= [n_samples]
반환 값	fpr: fpr값을 array로 반환

	tpr: tpr 값을 array로 반환
	thresholds: threshold 값 array

AUC(Area Under Curve)

: ROC 곡선 밑의 면적을 구한 것. 일반적으로 1에 가까울 수록 좋은 수치

AUC 수치 커지려면 FPR이 작은 상태에서 얼마나 큰 TPR을 얻을 수 있느냐가 관건

보통의 AUC 값은 0.5

코드 필사(colab)

https://colab.research.google.com/drive/1WcQotQFujz6hcFqv91iwkAhAHinD6BCv?usp=sharing