오차행렬(confusion matrix)

- 기본 아이디어 : class A의 샘플이 class B로 분류된 횟수 셈
- 교차 검증시 예측값 만드는 코드 :

from sklearn.model_selection import cross_val_predict

-> K-Fold 교차 검증 수행 하고 평가 점수가 아닌 예측값을 반환

- 오차 행렬 만드는 코드:

from sklearn.model_selection import confusion_matrix

output type : matrix type

실제 \ 예측	음성	양성
음성	TN (True Negative)	FP (False Positive)
양성	FN (False Negative)	TP (True Positive)

example : confusion_matrix(true_y,pred) ~> 클래스가 A인것을 예측

[[53272,1307],
[1077,4344]]

TN : A가 아닌것을 아니다라고 정확하게 분류한 개수 ⇒ 53272

FP : A가 아닌 것을 A라고 분류한 개수 ⇒ 1307

FN : A인것을 A가 아니라고 분류한 개수 ⇒ 1077

TP : A인것을 A라고 정확 하게 분류한 개수 ⇒ 4344

정밀도 = $\frac{TP}{TP+FP}$ ⇒ 양성 예측의 정확도

(precision)

$\frac{\text{진짜 양성}}{\text{진짜 양성} + \text{가짜 양성}}$

재현율 = $\frac{TP}{TP+FN}$ ⇒ 정확하게 감지한 양성 샘플의 비율

(Recall) or (Sensitivity)

$\frac{\text{진짜 양성}}{\text{진짜 양성} + \text{거짓음성}}$

F1-score : 정밀도와 재현율의 조화 평균

$$F_1 = \frac{2}{\frac{1}{\text{정밀도}} + \frac{1}{\text{재현율}}} = 2 \times \frac{\text{정밀도} \times \text{재현율}}{\text{정밀도} + \text{재현율}} = \frac{TP}{TP + \frac{FN+FP}{2}}$$

$$F = \frac{1}{\frac{\alpha}{\text{정밀도}} + \frac{1-\alpha}{\text{재현율}}} = (\beta^2+1) \times \frac{\text{정밀도} \times \text{재현율}}{\beta^2 \times \text{정밀도} + \text{재현율}} \quad \left(\beta^2 = \frac{1-\alpha}{\alpha}\right)$$

f1 score : from sklearn.metrics import f1_score

: f1_score(true_y,pred)

상황에 따라 정밀도와 재현율의 중요도가 다를 수 있다.

따라서 정밀도와 재현율의 Trade off 발생

정밀도와 재현율의 Trade off

example : SGDClassifier의 분류 기법

- SGDClassifier는 결정함수를 사용하여 샘플의 점수를 계산
- 결정 임계값 보다 크면 양성 클래스에 할당하고 아니면 음성 클래스에 할당한다.

즉, 임계값을 높이면 정밀도는 높아지고 재현율은 낮아진다

ROC Curve(Receiver Operating Characteristic)

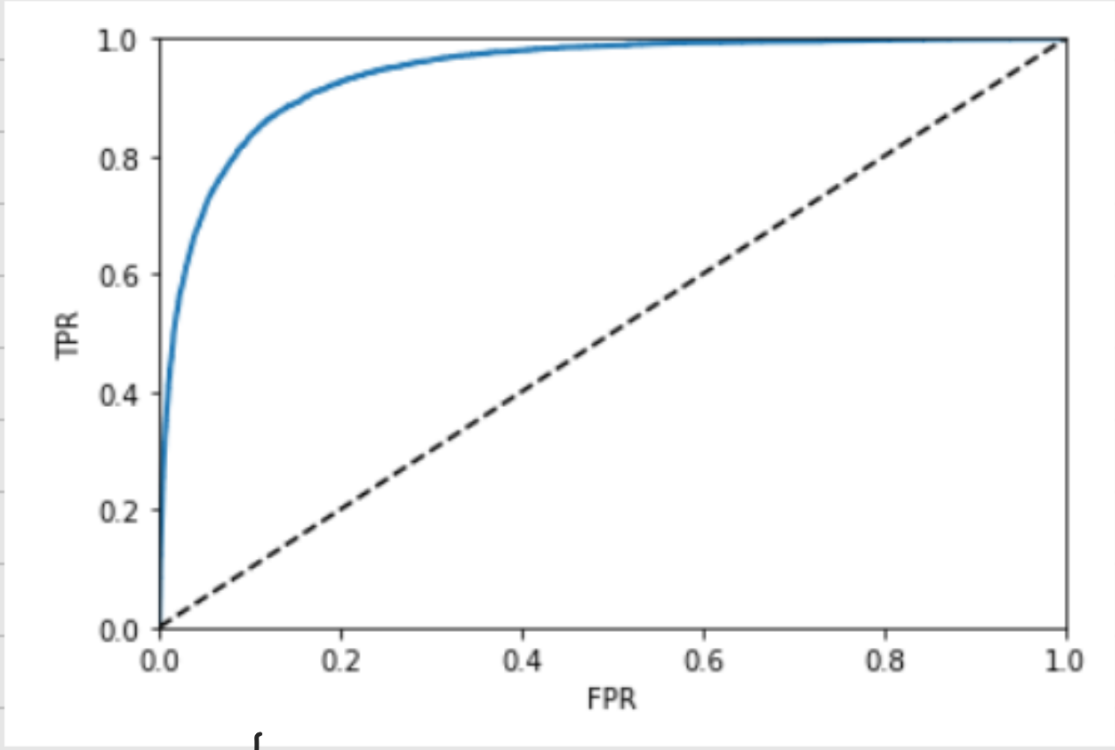
- ROC : 거짓 양성 비율(FPR)에 대한 진짜 양성 비율(TPR)
- FPR : 양성으로 잘못 분류된 음성 샘플의 비율(1-TNR)

특이도(Specificity) ↔ 진짜 음성 비율

따라서 ROC를 민감도에 대한 1-특이도 그래프라고 함

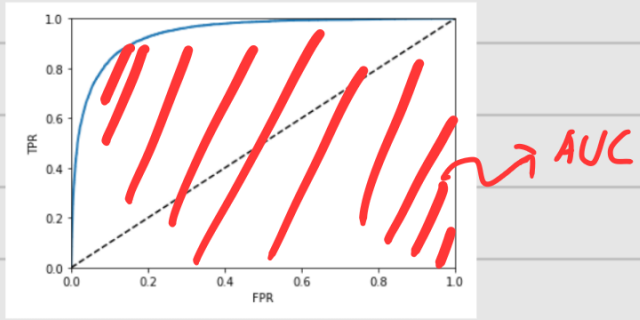
$$FPR = \frac{FP}{FP+TN} = \frac{FP+TN-TN}{FP+TN} = 1-\frac{TN}{FP+TN} = 1-TNR$$

```
from sklearn.metrics import roc_curve
-> fpr,tpr,thresholds = roc_curve(true_y,pred)
```



↳ SGD classifier ROC

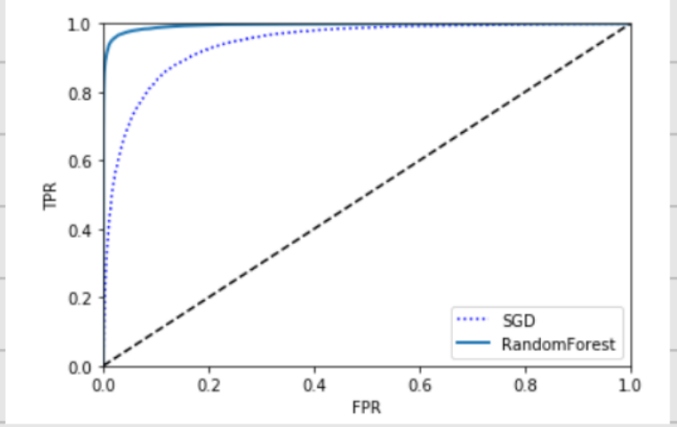
AUC : ROC curve에서 곡선 아래의 면적(클 수록 좋다)



from sklearn.metrics import roc_auc_score

roc_auc_curve(true_y,pred)

-> 0.924



실선 : RandomForest

=> 점선 : SGD

RF AUC => 0.995

SGD AUC => 0.924