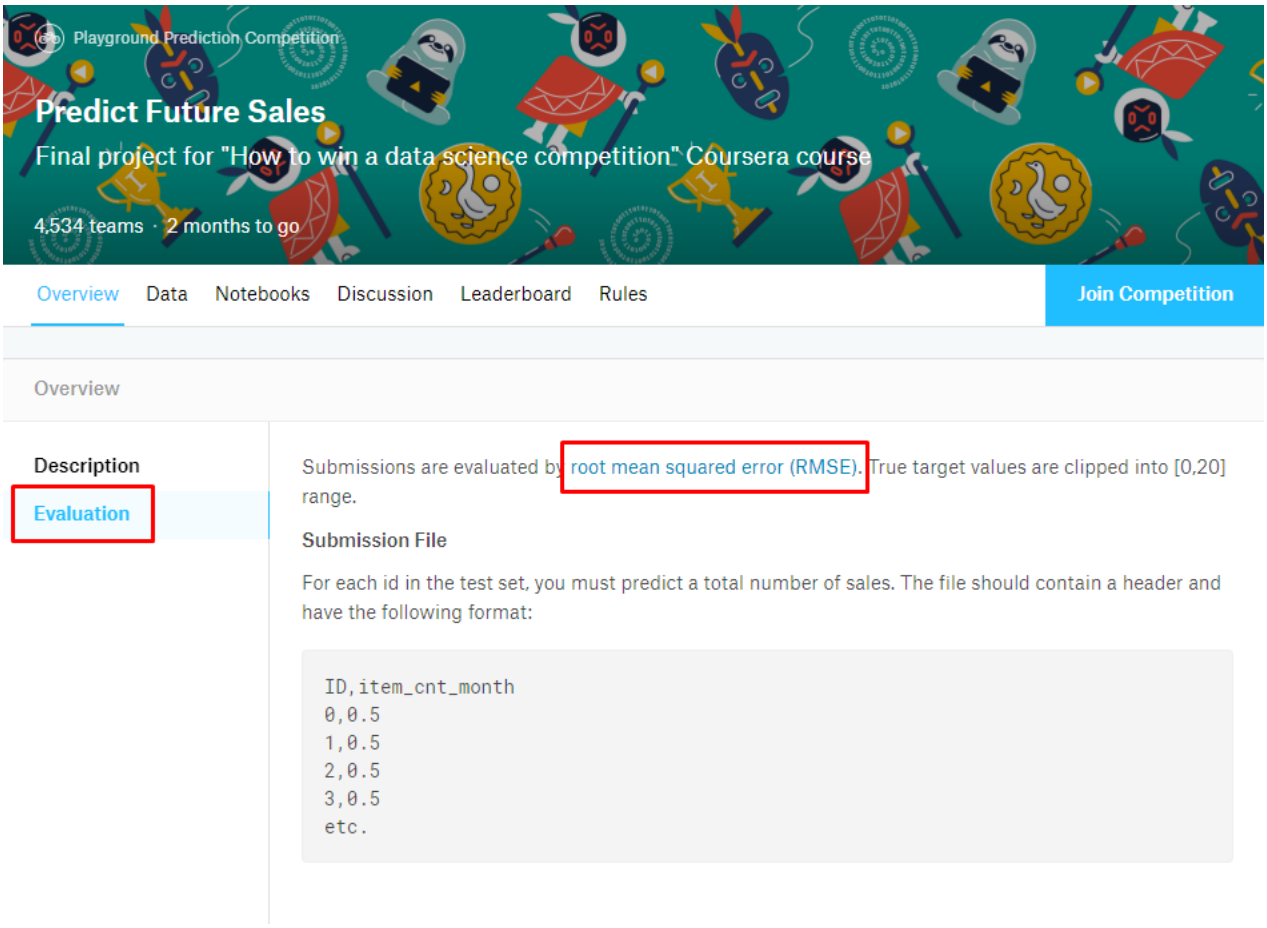
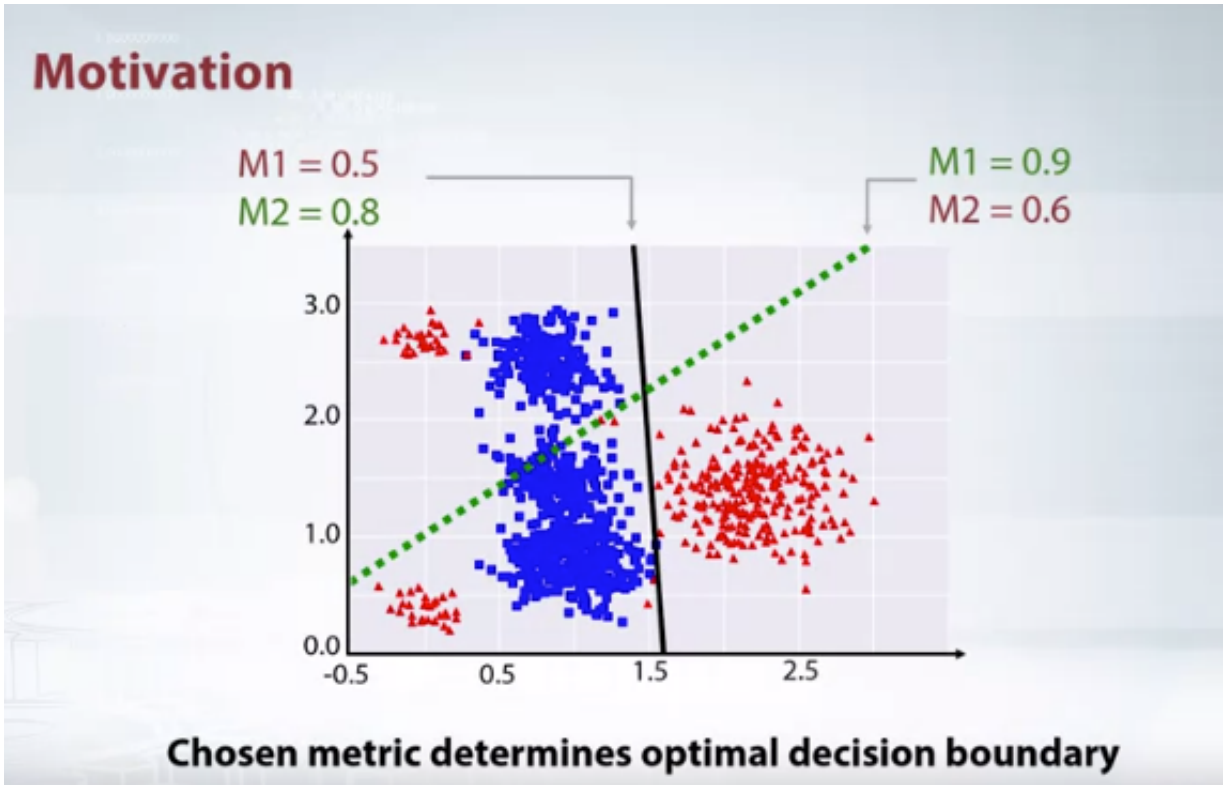
Coursera week3

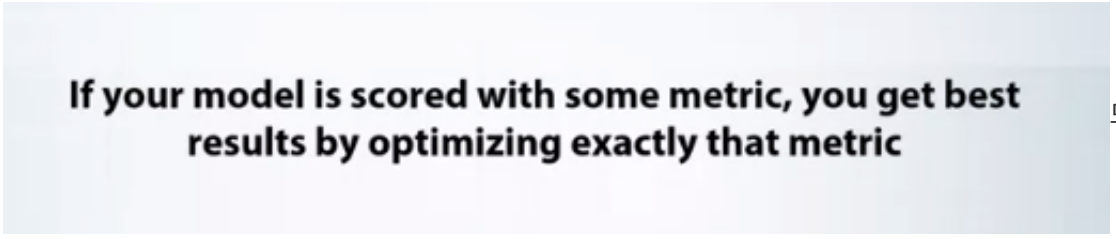
1. Motivation



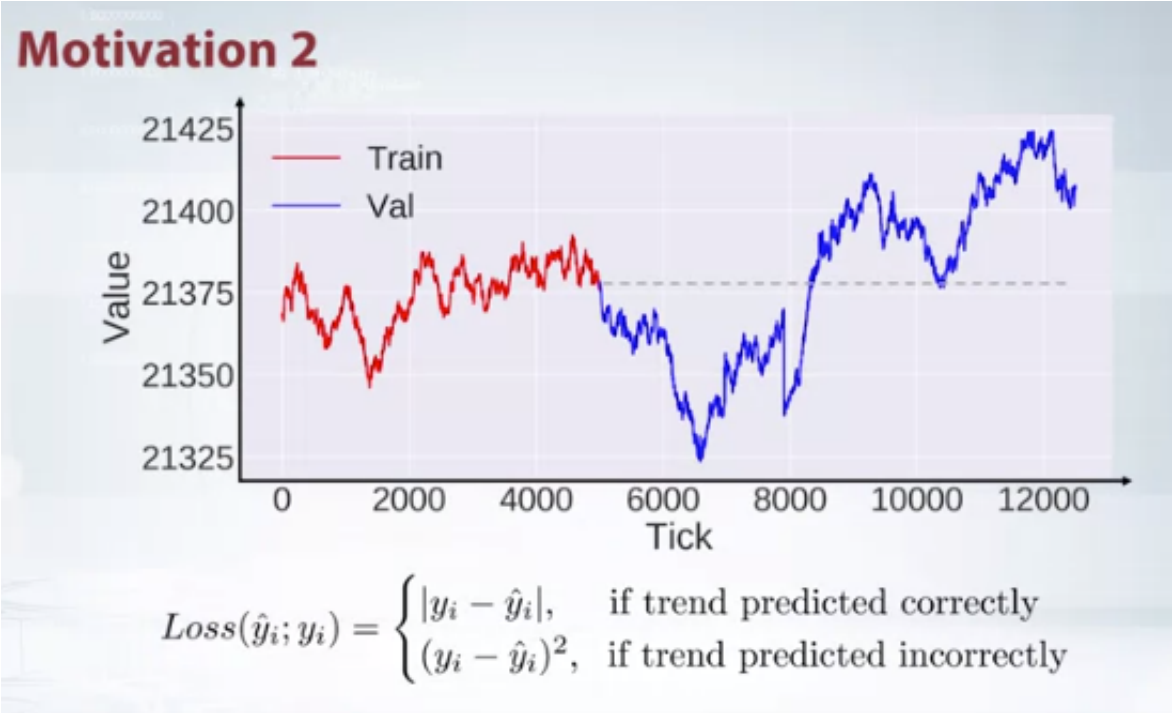


- 파란색 class 0
- 빨간색 class 1

Linear Classification 모델 2가지를 만들어 냈다고 하면 어떤 모델을 사용해야 할까요?

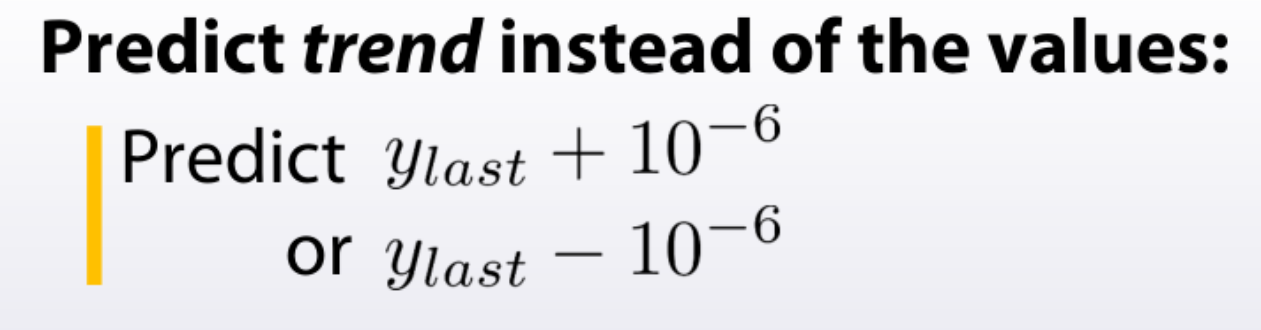


모델이 어떤 **metric**으로 점수가 매겨진다면, **그 metric**으로 최적화해야 최고의 결과를 얻을 수 있다.



이 컴피티션에서 **Loss**는

- 추세가 맞다면 (실제값 - 예측값)
- 추세가 틀리다면 (실제값 - 예측값) 의 제곱



추세(trend)를 맞추기 위해 예측값들에 작은 상수를 더하거나 뺐다.

> **결론**
>
> 컴피티션에서 요구한 metric 에 모델을 최적화하는 것이 리더보드에서 높은 랭커가 될 수 있는 방법이다.

2. Regression metrics

2-1. 공부할 내용

- MSE, RMSE, R-squared
- MAE
- MSPE, MAPE
- (R)MSLE

Notation

- N – number of objects
- $y \in \mathbb{R}^N$ – target values

  $\hat{y} \in \mathbb{R}^N$ – predictions
- $\hat{y}_i \in \mathbb{R}$ – prediction for i-th object

  $y_i \in \mathbb{R}$ – target for i-th object

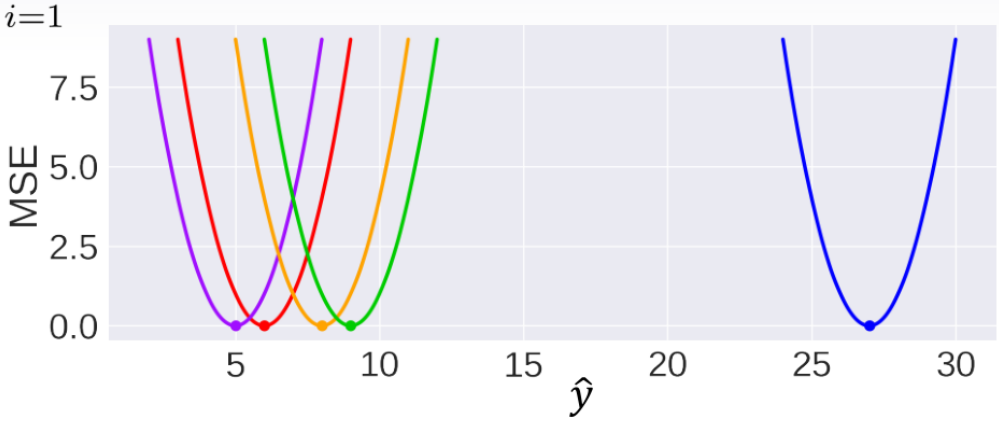
2-2. MSE: Mean Square Error

Regression 문제에 대한 가장 일반적인 metric

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

Data:

X	Y
...	5
...	9
...	8
...	6
...	27



만약에 모든 예측값을 하나로 통일한다면, 최소의 MSE를 얻기 위해 어떤 값으로 통일하는 게 좋을까요?

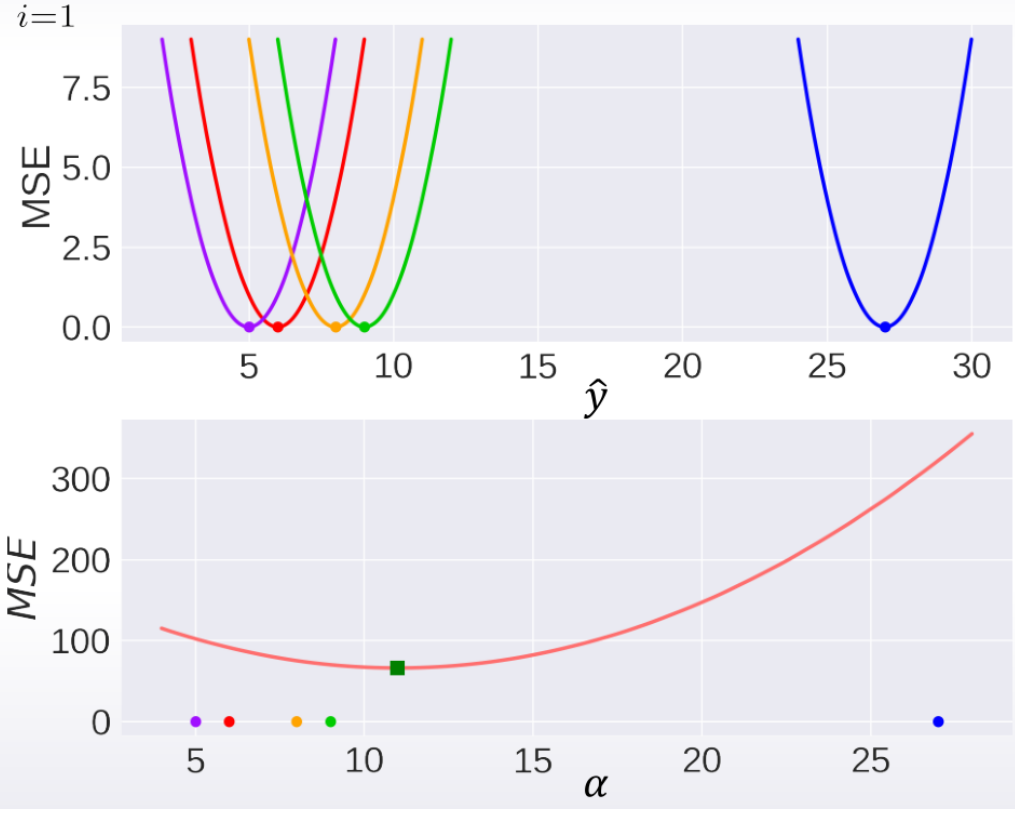
Best constant : target mean (11)

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \alpha)^2$$

Best constant: target mean

Data:

X	Y
...	5
...	9
...	8
...	6
...	27



2-3. RMSE: Root Mean Square Error

MSE 의 제곱근 (루트 씌운것)

일반적으로 MSE 와 RMSE 는 동일하게 해석 가능

- $$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} = \sqrt{\mathrm{MSE}}$$
- $$\mathrm{MSE}(a) > \mathrm{MSE}(b) \iff \mathrm{RMSE}(a) > \mathrm{RMSE}(b)$$

하지만, gradient 기반 해석에서는 동일하지 않을 수 있다. **?**

- $$\frac{\partial \mathrm{RMSE}}{\partial \hat{y}_i} = \frac{1}{2\sqrt{MSE}} \frac{\partial \mathrm{MSE}}{\partial \hat{y}_i}$$

> MSE 가 어느정도 되어야 좋은 모델?
>
> 32? 0.4?

2-4. R-squared

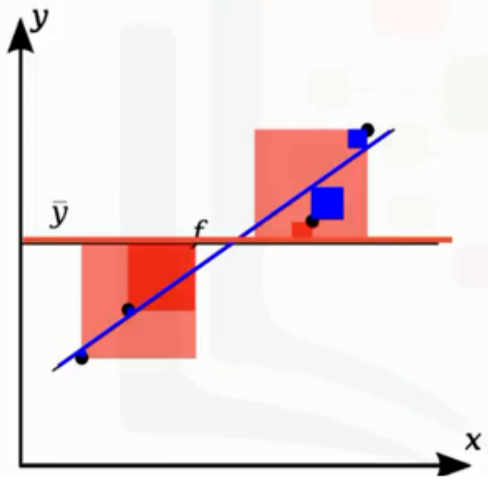
MSE 점수를 0점에서 1점 사이로.

$$R^2 = 1 - \frac{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\frac{1}{N}\sum_{i=1}^{N}(y_i - \bar{y})^2}$$

- `MSE = 0` ==> `R-squared = 1`
- `MSE = 상수모델의 MSE(R-zero)` ==> `R-squared = 0`

R-squared = 1

Coefficient of Determination (R^2)



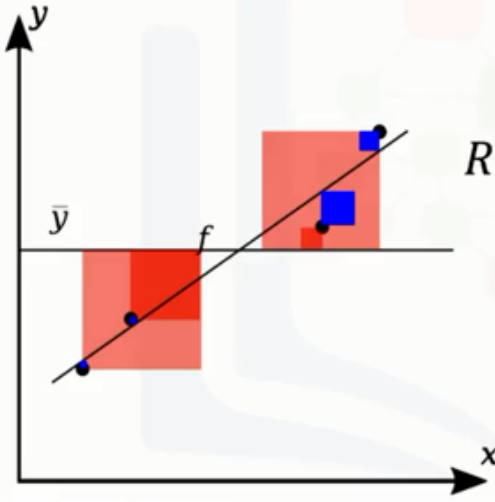
- The blue line represents the regression line
- The blue squares represents the MSE of the regression line
- The red line represents the average value of the data points
- The red squares represent the MSE of the red line
- We see the area of the blue squares is much smaller then the area of the red squares

Coefficient of Determination (R^2)

- In this case ratio of the areas of MSE is close to zero

$$\frac{\text{MSE of regression line}}{\text{MSE of } \bar{y}} = \frac{\blacksquare + \blacksquare}{\blacksquare + \blacksquare}$$

Coefficient of Determination (R^2)



$$R^2 = \left(1 - \frac{\text{MSE of regression line}}{\text{MSE of } \bar{y}}\right)$$

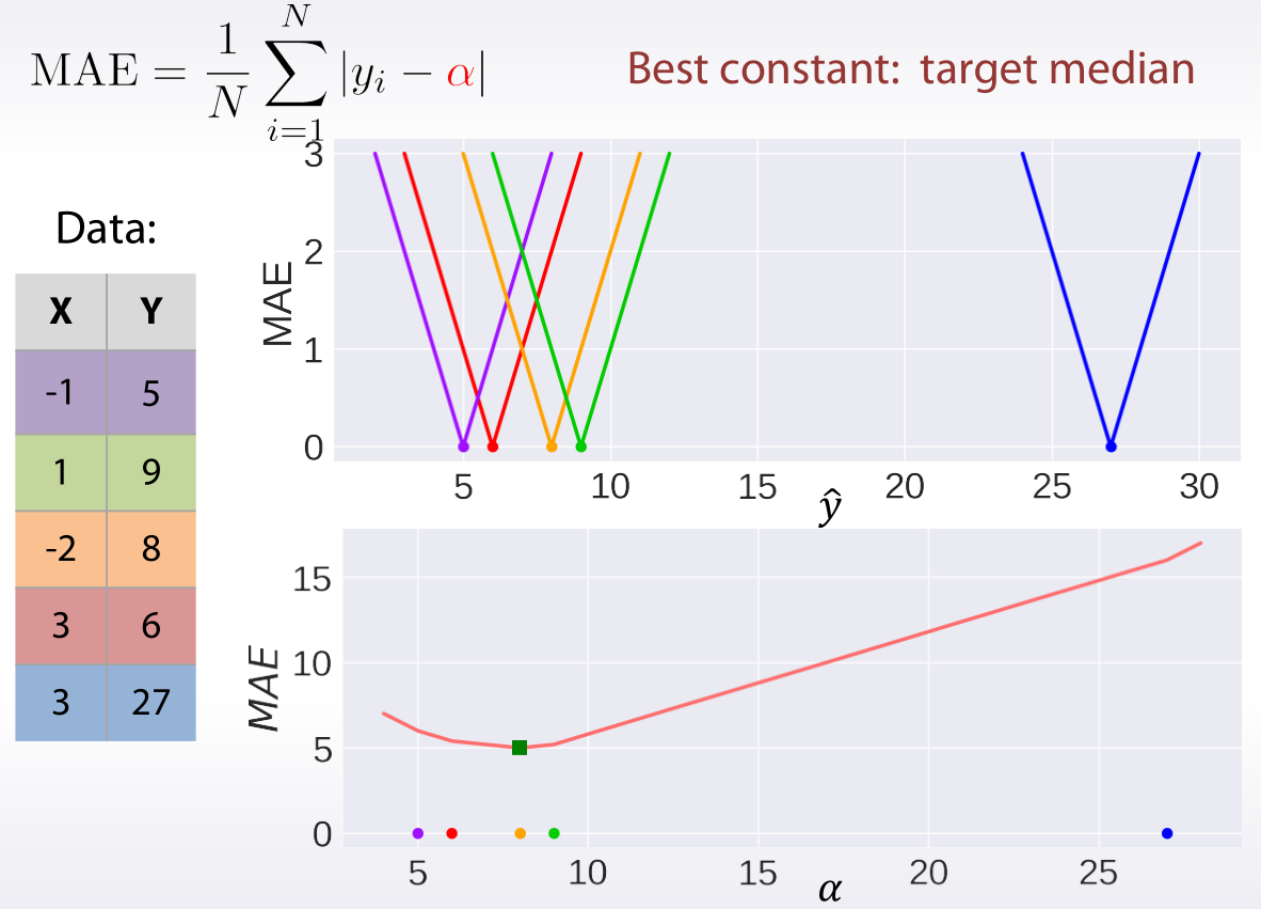
$$= (1 - 0)$$

$$= (1 - 0)$$

$$= 1$$

2-5. MAE: Mean Absolute Error

Best constant: target median (8)



> MAE 가 MSE 보다 robust 하다

2-6. MAE vs MSE

- **Do you have outliers in the data?**
  - Use MAE
- **Are you sure they are outliers?**
  - Use MAE
- **Or they are just unexpected values we should still care about?**
  - Use MSE

2-7. MSPE, MAPE

P: percentage

상대적 오류, MSE, MAE 의 **가중치 버전**

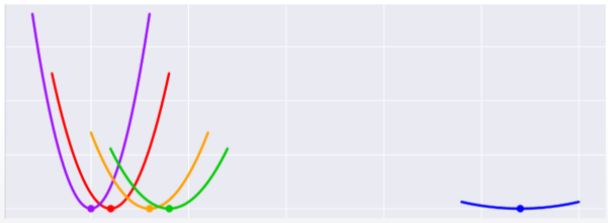
- **Shop 1**: predicted 9, sold 10, MSE = 1
- **Shop 2**: predicted 999, sold 1000, MSE = 1

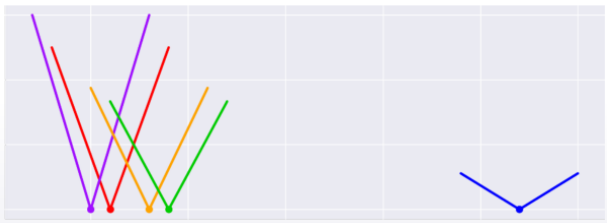
- **Shop 1**: predicted 9, sold 10, MSE = 1
- **Shop 2**: predicted 900, sold 1000, MSE = 10000

- **Shop 1**: predicted 9, sold 10, relative_metric = 1
- **Shop 2**: predicted 900, sold 1000, relative_metric = 1

실제 1000개 판매량 중 999개로 예측한 것 보다 10개 판매량 중 9개로 예측한 것이 상대적으로 더 큰 오류

$$\text{MSPE} = \frac{100\%}{N}\sum_{i=1}^{N}\left(\frac{y_i - \hat{y}_i}{y_i}\right)^2 \qquad \text{MAPE} = \frac{100\%}{N}\sum_{i=1}^{N}\left|\frac{y_i - \hat{y}_i}{y_i}\right|$$





target 값이 클수록 MSPE, MAPE 완만하게 증가

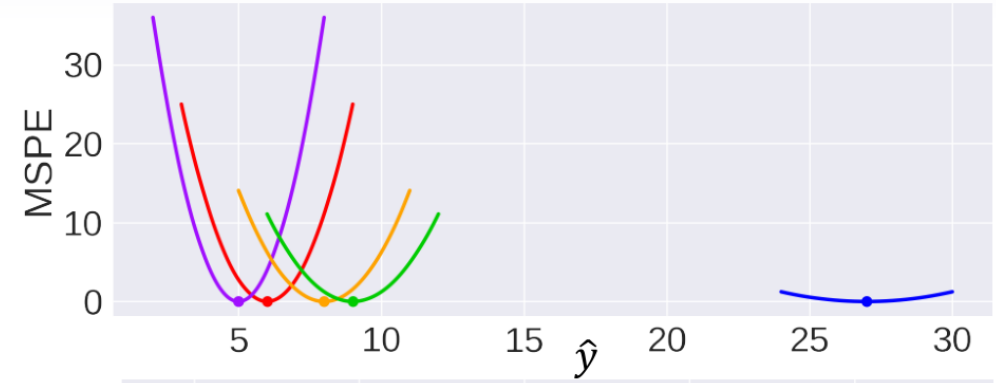
MSPE's Best constant: weight target mean (6.6)

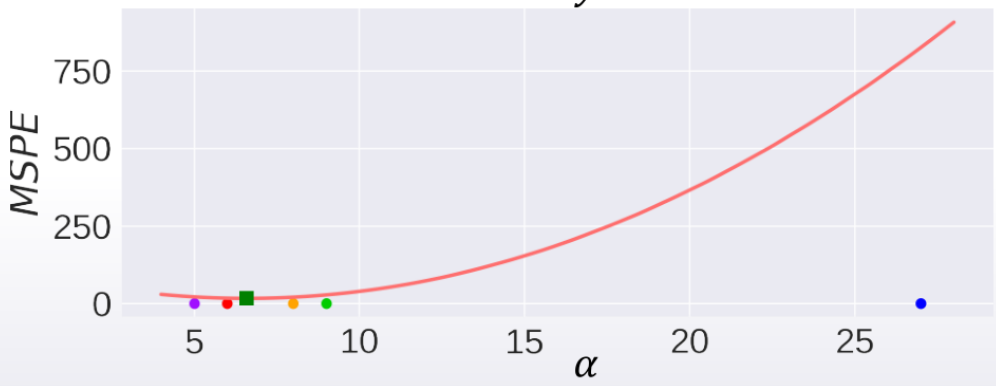
$$\text{MSPE} = \frac{100\%}{N}\sum_{i=1}^{N}\left(\frac{y_i - \alpha}{y_i}\right)^2$$

Best constant:
weighted target mean

Data:

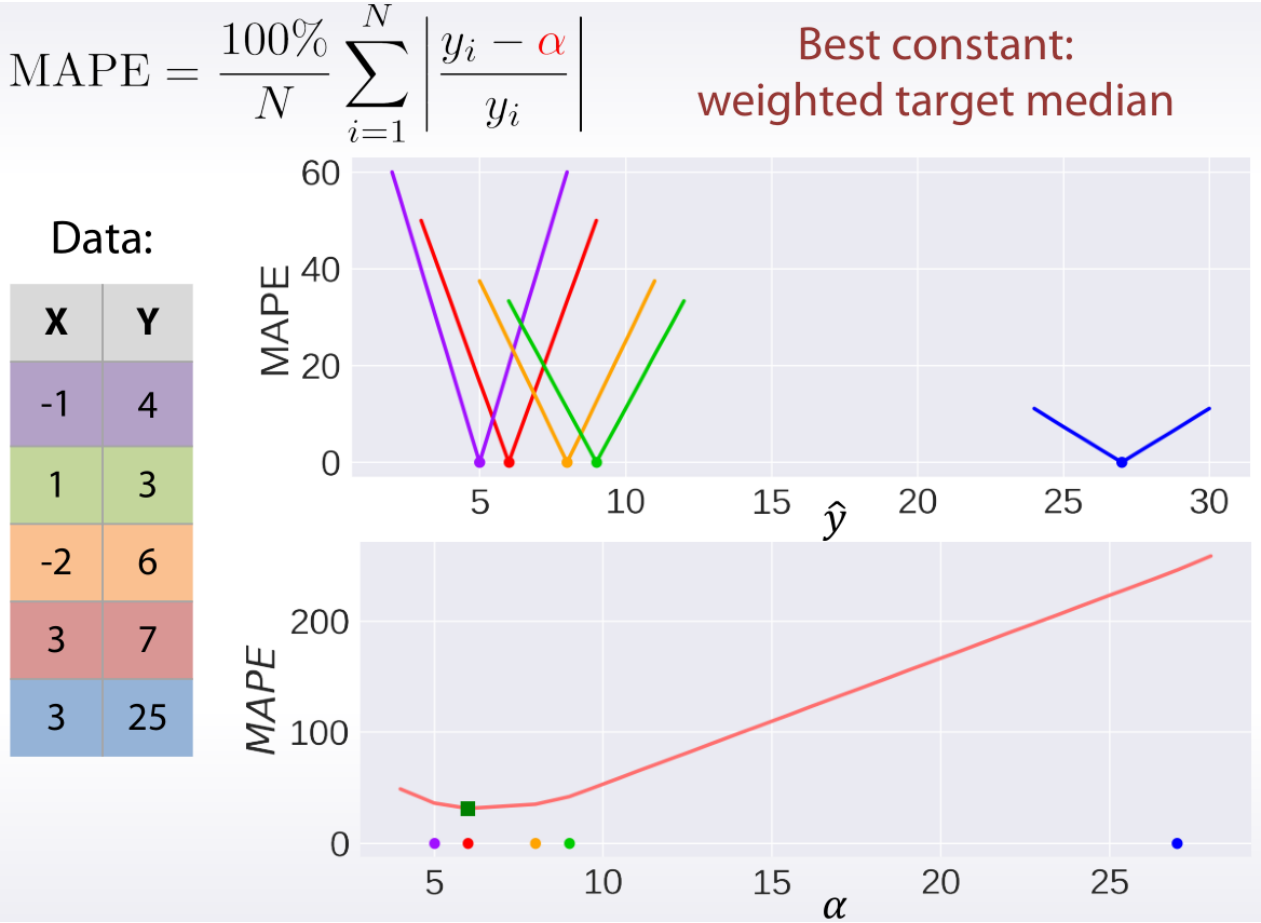
X	Y
-1	4
1	3
-2	6
3	7
3	25





> MSE's constant (11) 보다 작게 설정됨

MAPE's Best constant: weight target mean (6)



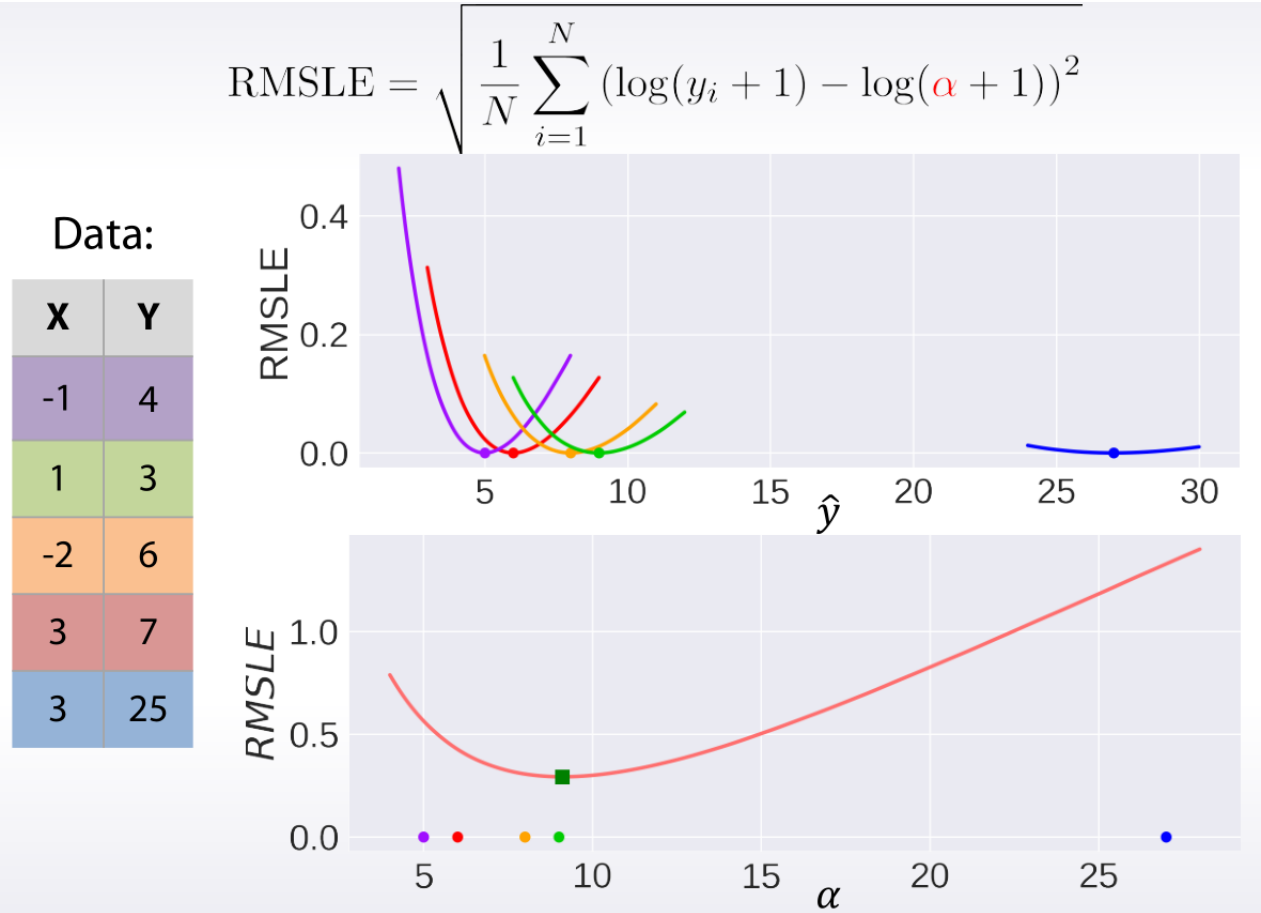
하지만 MSPE 나 MAPE 가 이상치에 대해 robust 한 metric 은 아니라고 설명. 오히려 아주아주 작은 이상치가 존재할 경우 작은 쪽으로 편향되어 best constant 가 설정될 수 있다.

2-8. (R)MSLE: Root Mean Square Logarithmic Error

$$\mathrm{RMSLE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\log(y_i+1) - \log(\hat{y}_i+1)\right)^2} =$$
$$= RMSE\left(\log(y_i+1), \log(\hat{y}_i+1)\right) =$$
$$= \sqrt{MSE\left(\log(y_i+1), \log(\hat{y}_i+1)\right)}$$

log 0 이 정의되지 않으므로 상수 더한다. 상수는 1이 아니라 임의의 수가 될 수 있다.

RMSLE's Best constant: 9.11



RMSLE 를 사용할 경우, target 값보다 높게 예측하는 것이 더 작은 RMSLE 값을 갖게 한다.

2-9. Wrap-up

Optimal constant 비교

Metric	Constant
MSE	11
RMSLE	9.11
MAE	8
MSPE	6.6
MAPE	6

- **MSE**: 큰 값에 대해 편향을 가진다
- **MAE**: 큰 값에 대해 MSE보다 적은 편향을 가진다.
- **MSPE/MAPE**: 작은 값에 대해 편향을 가진다.
- **RMSLE**: MAPE 보다 나은 metric 으로 여겨지는데, 작은 값에 대해 덜 편향적이고 상대적인 오류에 더 잘 작동하기 때문

3. Classification metrics

3-1. 공부할 내용

- Accuracy
- Logarithmic loss
- Area under ROC curve
- (Quadratic weighted) Kappa

Notation

- N – is number of objects
- L – is number of classes

- y – ground truth
- $\hat{y}$ – predictions
- $[a = b]$ – indicator function

- Soft labels (soft predictions) are classifier's scores

- Hard labels (hard predictions):
  - $\arg\max_i f_i(x)$
  - $[f(x) > b]$, b – threshold

- Soft predictions
- Hard predictions
  - multi class 문제에서는 argmax 로,
  - binary class 문제에서는 threshold 로

3-2. Accuracy

전체 예측셋 중 ground truth 와 일치하는 예측셋의 비율

`Hard predictions` 필요

Best constant : 가장 많이 등장하는 class

Dataset:
- 10 cats
- 90 dogs

Predict always dog:
Accuracy = **0.9**!

> - Accuracy 는 직관적이지만 최적화 하는것은 어렵다.
> - Accuracy 에서는 `soft predictions` 는 중요하지 않다.

3-3. Logarithmic loss (logloss)

`Soft predictions` 필요

Binary

Binary:

$$\text{LogLoss} = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)$$

$$y_i \in \mathbb{R}, \quad \hat{y}_i \in \mathbb{R}$$

- class(y_i) 가 0 또는 1이 있다고 생각

```
def logloss(true_label, predicted, eps=1e-15):
    p = np.clip(predicted, eps, 1-eps)
    if true_label == 1:
        return -log(p)
    else:
        return -log(1-p)
```

Multiclass

- Multiclass:

$$\text{LogLoss} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{i=1}^{L} y_{il} \log(\hat{y}_{il})$$

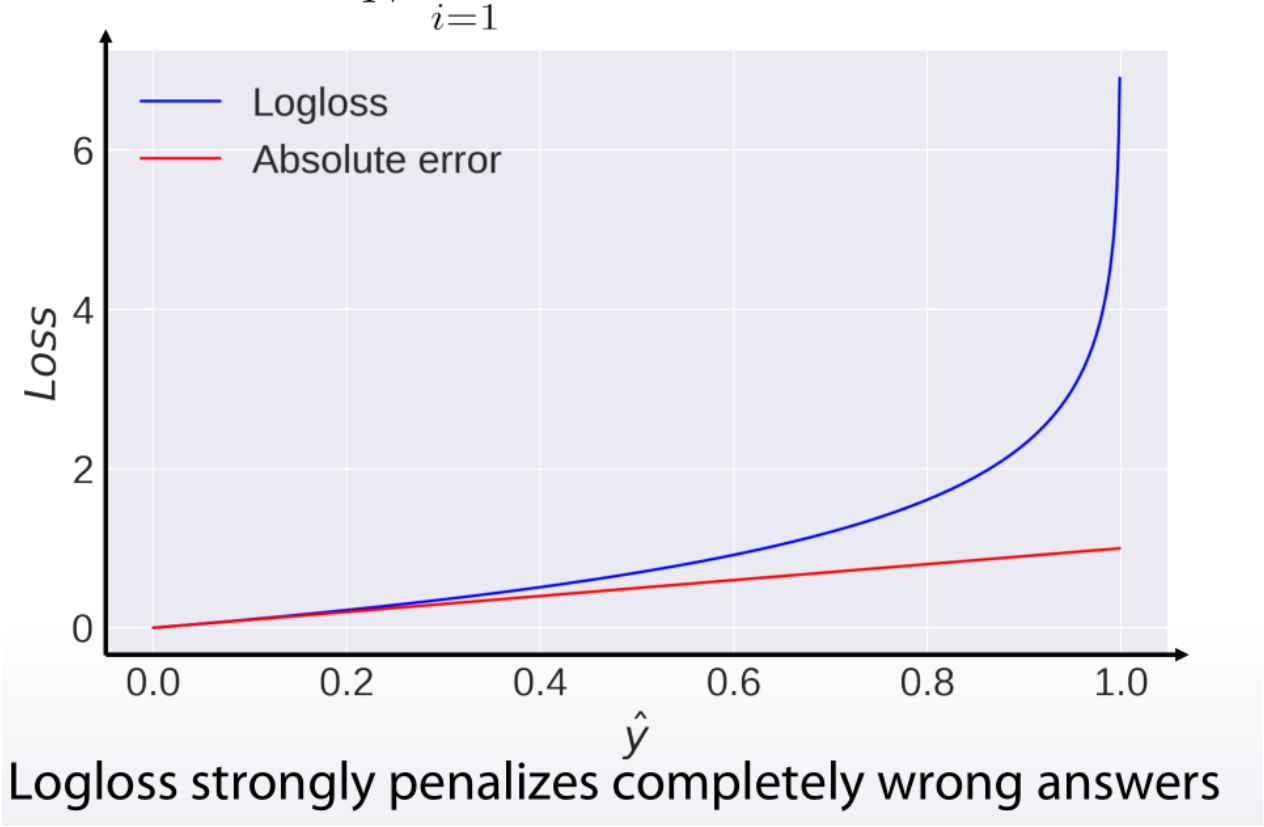
$$y_i \in \mathbb{R}^L, \quad \hat{y}_i \in \mathbb{R}^L$$

- In practice:

$$\text{LogLoss} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{i=1}^{L} y_{il} \log(\min(\max(\hat{y}_{il}, 10^{-15}), 1 - 10^{-15}))$$

???

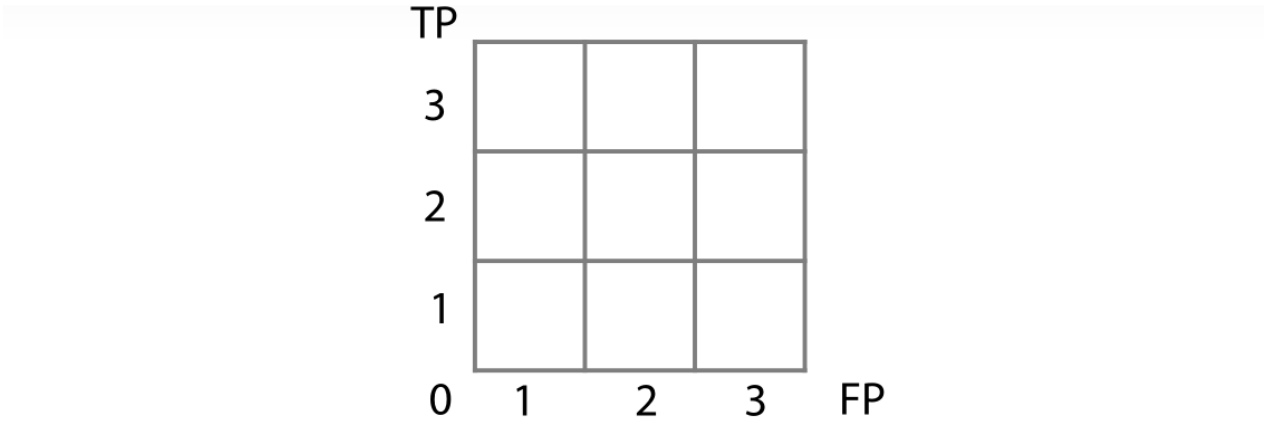
$$\text{LogLoss} = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)$$



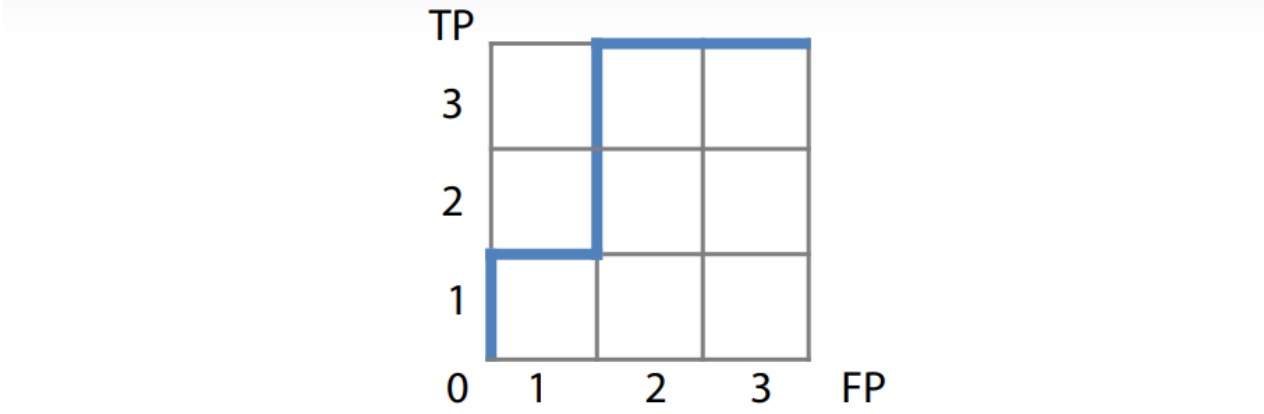
> - logloss 는 완전히 잘못된 예측에 대해 더 많은 페널티를 부여
> - 적게 잘못된 예측이 많은 모델이 더 좋은 점수를 받는다.

3-4. Area Under ROC Curve

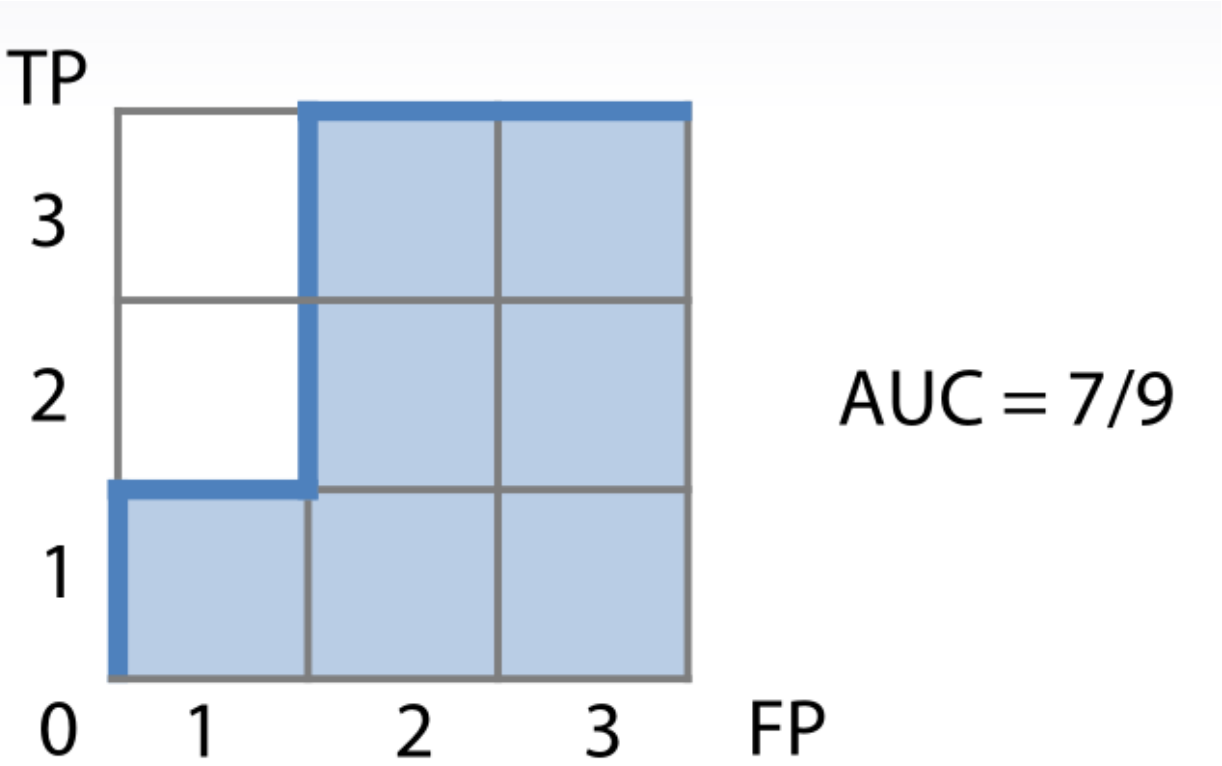
ROC (Receiver Operating Characteristic)



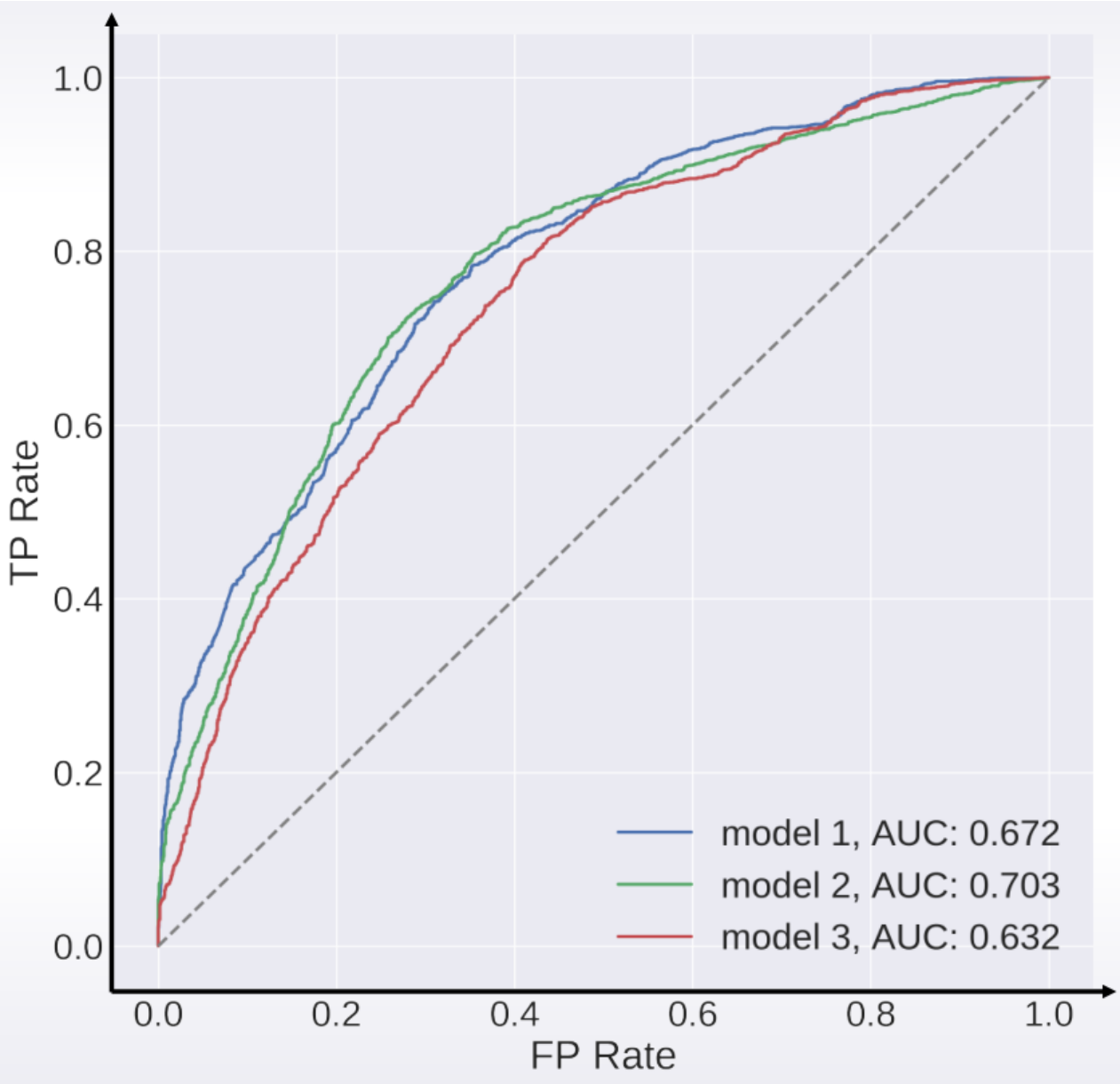
TP – true positives, **FP** – false positives



AUC (Area Under Curve)



- binary task 에서만 사용
- 예측의 순서에 의존적



3-5. Kappa

Motivation

R-squared 와 비슷

Dataset:

- 10 cats
- 90 dogs

Baseline accuracy = 0.9

$$\text{my\_score} = 1 - \frac{1 - \text{accuracy}}{1 - \text{baseline}}$$

- accuracy = 1 ⟹ my_score = 1
- accuracy = 0.9 ⟹ my_score = 0

Kappa는 랜덤을 사용해 baseline 을 설정

Dataset:

- 10 cats
- 90 dogs

Predict *20 cats* and *80 dogs* at random: *accuracy* ~ 0.74

0.2*0.1 + 0.8*0.9 = 0.74

$$\text{Cohen's Kappa} = 1 - \frac{1 - \text{accuracy}}{1 - p_e}$$

p_e – what accuracy would be on average, if we randomly permute our predictions

$$p_e = \frac{1}{N^2} \sum_k n_{k1} n_{k2}$$

해석: 무작위로 class를 예측했을때의 accuracy baseline

여기부터 이해가 잘 안됩니다...

---

weighted Kappa and weighted error

Confusion matrix *C*

pred\ true	cat	dog	tiger
cat	4	2	3
dog	2	88	4
tiger	4	10	12

Weight matrix *W*

pred\ true	cat	dog	tiger
cat	0	1	10
dog	1	0	10
tiger	1	1	0

$$\text{weighted error} = \frac{1}{const} \sum_{i,j} C_{ij} W_{ij}$$

$$\text{weighted kappa} = 1 - \frac{\text{weighted error}}{\text{weighted baseline error}}$$

Quadratic and Linear Weighted Kappa

Linear weights

pred\ true	1	2	3
1	0	1	2
2	1	0	1
3	2	1	0

Quadratic weights

pred\ true	1	2	3
1	0	1	4
2	1	0	1
3	4	1	0

$$w_{ij} = |i - j|$$

$$w_{ij} = (i - j)^2$$

$$\text{weighted kappa} = 1 - \frac{\text{weighted error}}{\text{weighted baseline error}}$$