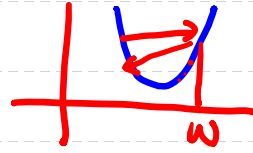


· 10/6 (화)

1. 성능지표 (metrics) [ Precision (정밀도)  
Recall (재현율)  
Accuracy (정확도)

2. learning rate [ 너무 크면 → **overshooting**  
너무 작으면 → **local minima**



⇒ loss 값을 보면서 조금씩 값을 조정해야 해요!

3. Normalisation (정제화) [ Min-Max Normalization ⇒ (0~1 범위로 scaling)  
Z-Score Normalization (Standardization) ⇒ (이상의 값도 포함)

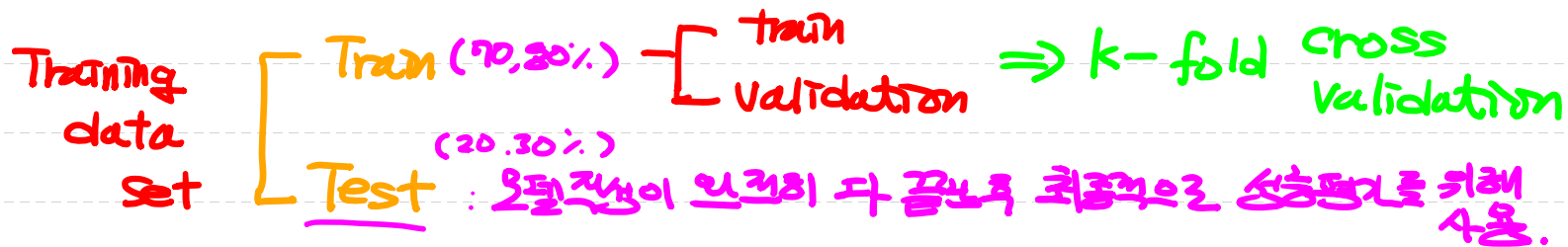
⇒ (이상의 값도 포함)  
범위를 조정해서 scaling이 돼요.

4. [ underfitting → 학습자처가 잘 안되는 경우!  
✓ overfitting → training data에 너무 잘 맞는 model이 생성되서  
실제 data에는 적용이 잘 안되는 경우!

- Overfitting을 피하려면 어떻게 해야 하나요?

- Training data set이 커야 해요. (?)
- "Feature Engineering"을 이용해서 feature의 개수를 줄여요!
- "Regularization"을 이용해서 Weight 값을 인위적으로 조절!
- "Dropout"
- 성능을 크게 ( Training data set을 가지고 성능평가  $\Rightarrow$  좋게 나옵니다!  $\frac{99\%}{}$  )  
 Validation data set을 " "  $\Rightarrow$  성능이 나옵니다!  $\frac{70\%}{}$  ]

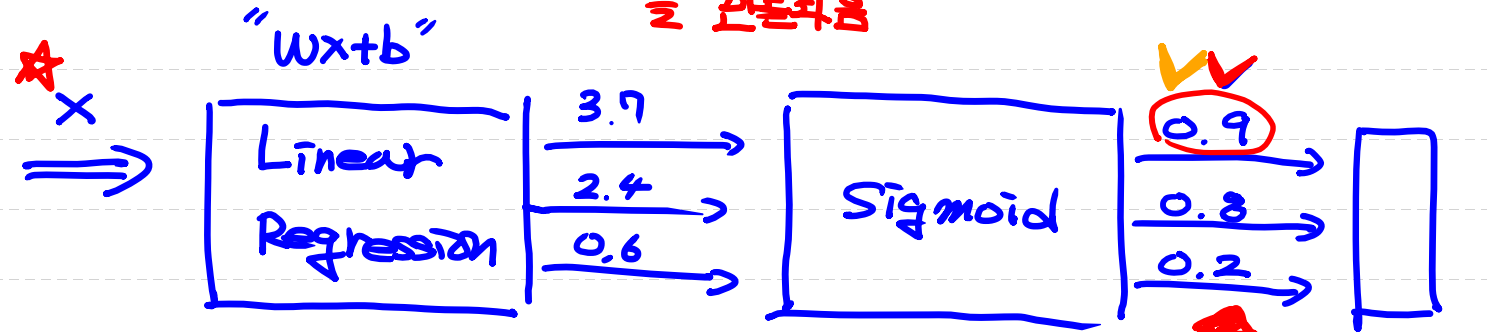
- Evaluation (평가) : 데이터를 어떻게 나누어서 성능평가를 할 것인지??



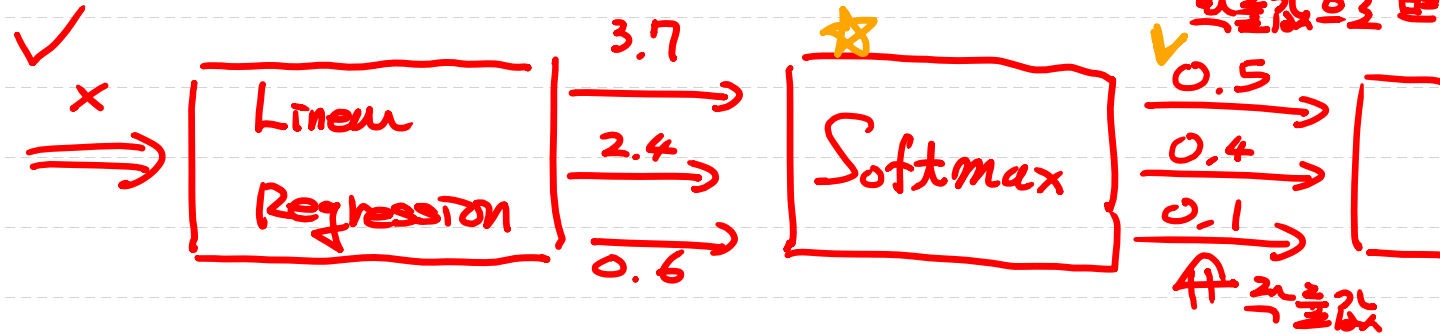
# · Multinomial Classification (다중분류)

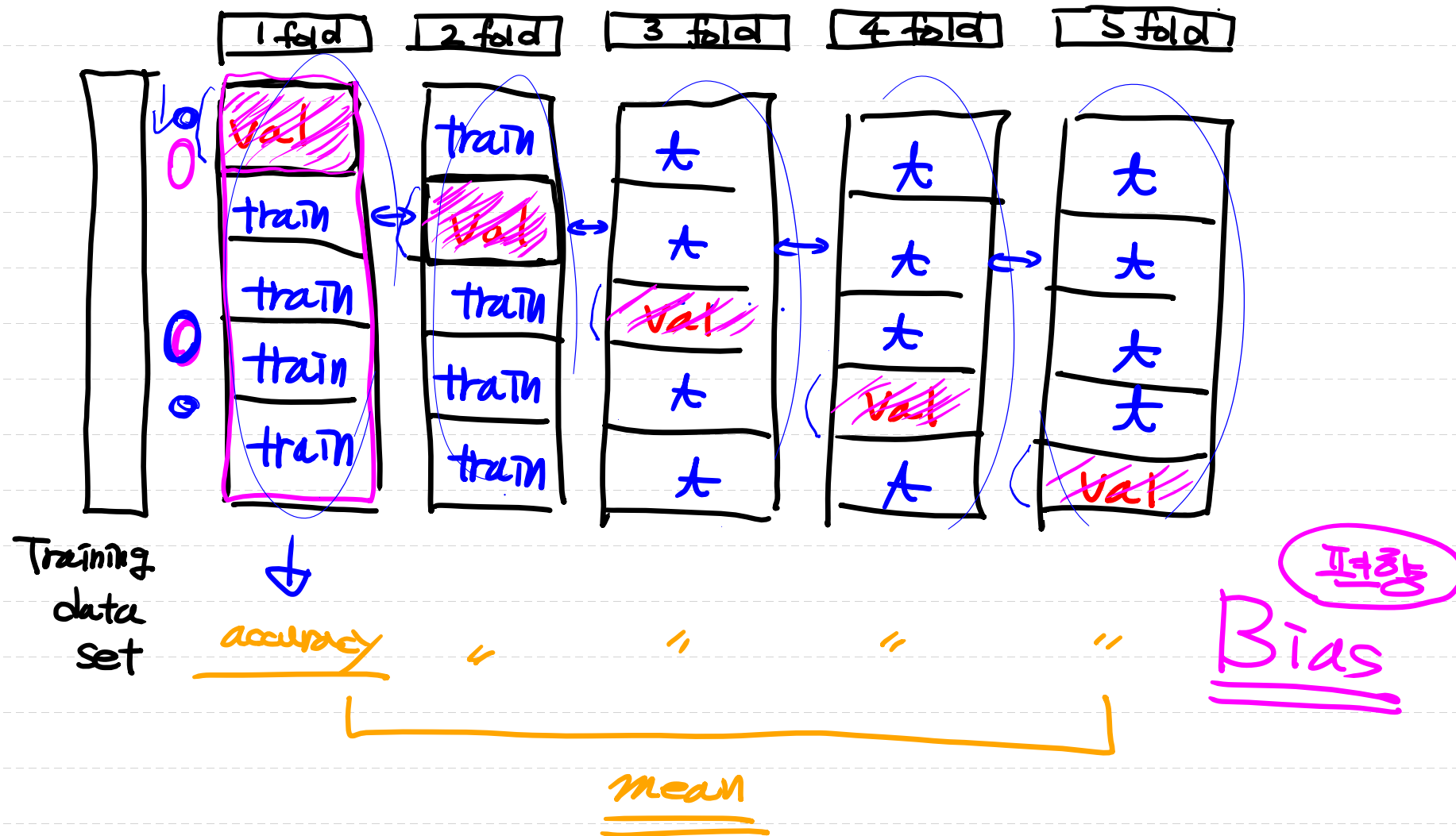
⇒ "logistic의 실용"

→ 여러개의 logistic을 구현하는 행렬식 ⇒ 이것 리파로 표현 !!!  
을 만든다음

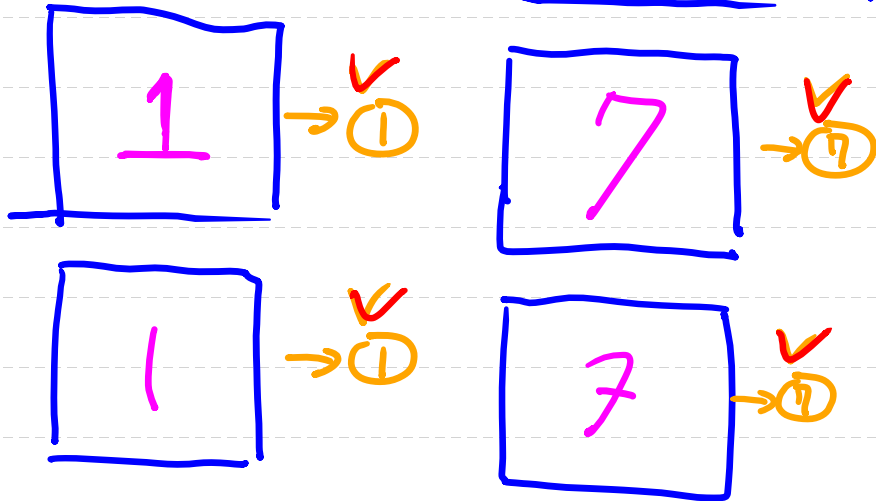
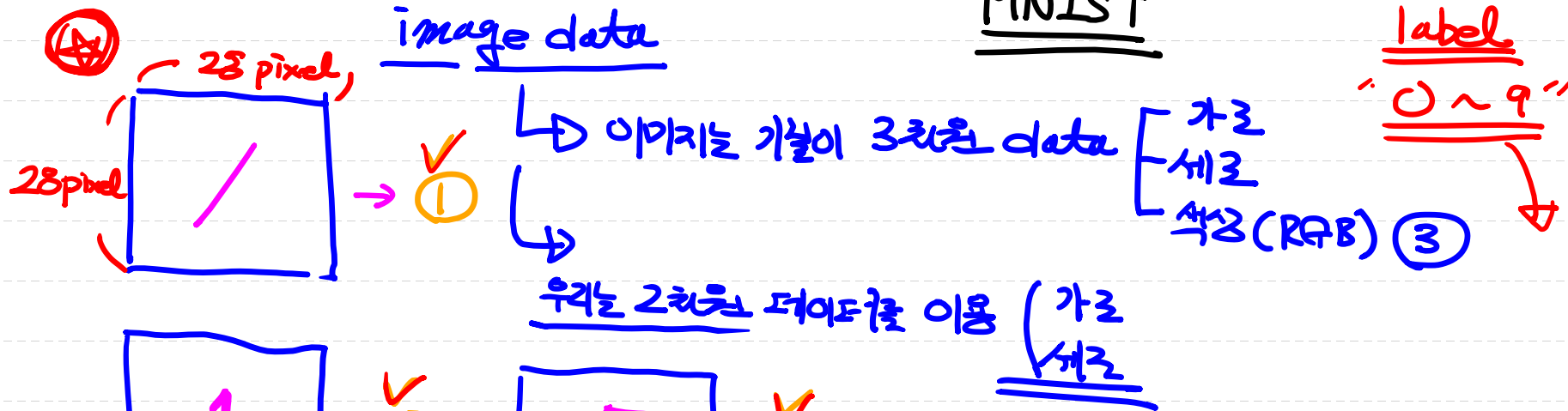


↑  
확률값으로 바꾼.





- Multinomial classification의 "Hello world" 라고 불리는  
"MNIST"



⇒ 왜 필요한가요??

✓ 3 2 3 - 1 0 0

"Kaggle" ⇒ "Digit Recognizer"