



Bike Sharing Demand

Forecast use of a city bikeshare system

3,251 teams · 3 years ago

[Overview](#)[Data](#)[Kernels](#)[Discussion](#)[Leaderboard](#)[Rules](#)[Team](#)[My Submissions](#)[Late Submission](#)

Overview

Description

Evaluation

[Get started on this competition through Kaggle Scripts](#)

Bike sharing systems are a means of renting bicycles where the process of obtaining membership, rental, and bike return is automated via a network of kiosk locations throughout a city. Using these systems, people are able rent a bike from a one location and return it to a different place on an as-needed basis. Currently, there are over 500 bike-sharing programs around the world.

The data generated by these systems makes them attractive for travel, departure location, arrival location, and time elapsed is ex therefore function as a sensor network, which can be used for st competition, participants are asked to combine historical usage , forecast bike rental demand in the Capital Bikeshare program in Washington, DC.

Bike Sharing Demand

tips and tricks

DS School

<https://dsschool.co.kr>

Classification vs Regression

Classification vs Regression

맞춰야 하는 정답(Label, y)이 categorical하면(특정 분류 중에 하나) **Classification** 문제라고 정의합니다.
반면 맞춰야 하는 정답이 continuous하면(높고 낮음을 비교할 수 있는 숫자) **Regression** 문제라고 정의합니다.

Classification

암 환자 예측(양성/음성)

스팸 필터링 예측(ham/spam)

광고성 게시물 예측(광고/광고 아님)

Regression

부동산 가격 예측

삼성전자 주가 예측

비트코인 가격 예측

Classification vs Regression

맞춰야 하는 정답(Label, y)이 categorical하면(특정 분류 중에 하나) **Classification** 문제라고 정의합니다.
반면 맞춰야 하는 정답이 continuous하면(높고 낮음을 비교할 수 있는 숫자) **Regression** 문제라고 정의합니다.

Classification

암 환자 예측(양성/음성)

스팸 필터링 예측(ham/spam)

광고성 게시물 예측(광고/광고 아님)

$$y = 0 \text{ or } 1$$

Regression

부동산 가격 예측

삼성전자 주가 예측

비트코인 가격 예측

$$y = -\infty \sim +\infty$$

Classification vs Regression

맞춰야 하는 정답(Label, y)이 categorical하면(특정 분류 중에 하나) **Classification** 문제라고 정의합니다.
반면 맞춰야 하는 정답이 continuous하면(높고 낮음을 비교할 수 있는 숫자) **Regression** 문제라고 정의합니다.

Classification

암 환자 예측(양성/음성)

스팸 필터링 예측(음성/양성)

광고성 게시물 예측(광고/광고 아님)

$y = 0 \text{ or } 1$

(categorical)

Regression

부동산 가격 예측

삼성전자 주가 예측

비트코인 가격 예측

$y = -\infty \sim +\infty$

(continuous)

Classification vs Regression

맞춰야 하는 정답(Label, y)이 categorical하면(특정 분류 중에 하나) **Classification** 문제라고 정의합니다.
반면 맞춰야 하는 정답이 continuous하면(높고 낮음을 비교할 수 있는 숫자) **Regression** 문제라고 정의합니다.

Classification

암 환자 예측(양성/음성)

스팸 필터링 예측(음성/양성)

광고성 게시물 예측(광고/광고 아님)

Regression

부동산 가격 예측

삼성전자 주가 예측

비트코인 가격 예측

```
from sklearn.tree import DecisionTreeClassifier

model = DecisionTreeClassifier()
model
```

```
from sklearn.tree import DecisionTreeRegressor

model = DecisionTreeRegressor()
model
```

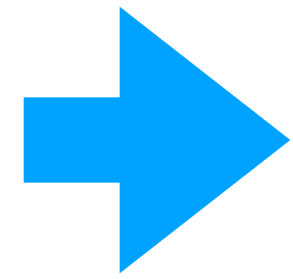
Classification 문제에서는 **DecisionTreeClassifier**를 사용하고,
Regression 문제에서는 **DecisionTreeRegressor**를 사용하면 됩니다

Model Validation

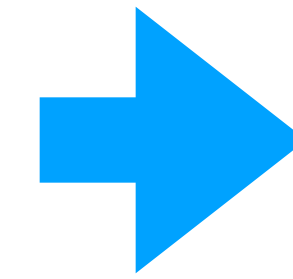
이전까지는 가설 -> 검증 -> 예측의 과정을 따랐는데,
엑셀에서는 pivot table을 통해 검증했지만, 파이썬에서는 pivot table이 아닌 다른 방식을 사용해야 한다.

데이터분석가가 데이터를 분석하는 방식

가설



검증



예측

데이터를 살펴보고 가설을 세운다
(ex: 불래지수가 높으면 자전거를 덜 탈 것이다)

가설이 맞는지 검증한다
(ex: pivot table)

검증을 통해 가설이 맞다면 예측한뒤
캐글에 제출하거나 실제 서비스에 올린다.

이전까지는 가설 -> 검증 -> 예측의 과정을 따랐는데,
엑셀에서는 pivot table을 통해 검증했지만, 파이썬에서는 pivot table이 아닌 다른 방식을 사용해야 한다.

데이터분석가가 데이터를 분석하는 방식



엑셀에서는 pivot table을 사용해서 검증했지만,
파이썬에서는 pivot table으로 검증할 수 없다.

Hold-out Validation

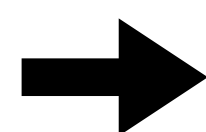
train 데이터 전체로 fit하지 않고 일부분만 fit하고, 남은 부분을 predict합니다.
이렇게 하면 남은 부분의 정답(actual)과 예측값(predict)을 비교할 수 있습니다.

Train 데이터

데이터 갯수: 10000 개



8,000 개



2,000 개

8,000 개로 학습(fit)한 뒤
2,000개를 예측(predict)한다

데이터를 8:2로 나눈 뒤 낸 뒤,
큰 조각을 train 데이터로, 작은 조각을 test 데이터로 가정한다.

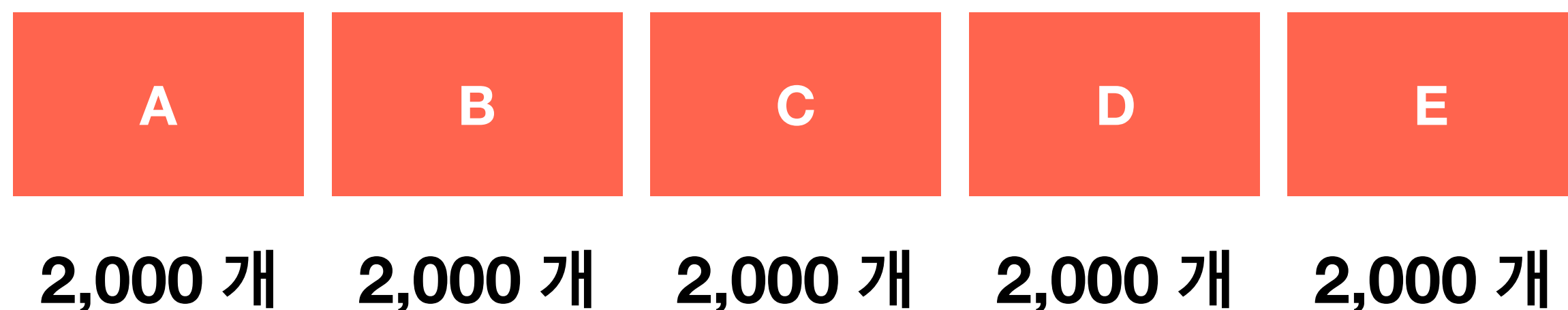
Label의 예측값, 즉 $y(\text{predict})$ 가 나온다.
이 $y(\text{predict})$ 를 Label의 정답인 $y(\text{actual})$ 과 비교한다

Cross Validation

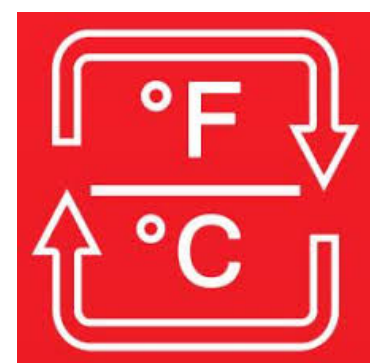
train 데이터를 $1/n$ 로 나눈 뒤 한 조각을 제외한 나머지로 fit하고 한 조각을 predict합니다.
이 방식을 n 번 반복하면 결과적으로 train데이터의 갯수와 일치하는 정답(actual)과 예측값(predict)이 나옵니다.

Train 데이터

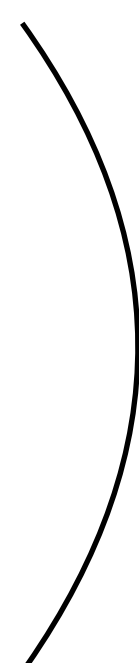
데이터 갯수: 10000 개



데이터를 5조각 낸 뒤,
조각을 뺀 나머지로 알고리즘을 학습하고, 조각을 예측한다.



1. $B + C + D + E \rightarrow A$
2. $A + C + D + E \rightarrow B$
3. $A + B + D + E \rightarrow C$
4. $A + B + C + E \rightarrow D$
5. $A + B + C + D \rightarrow E$



Label의 예측값, 즉 $y(\text{predict})$ 가 나온다.
이 $y(\text{predict})$ 를 Label의 정답인 $y(\text{actual})$ 과 비교한다

Hold-out Validation vs Cross Validation









실행 속도를 중시하면 hold-out validation, 정확도가 중요하다면 cross-validation을 사용합니다.

	Hold-Out Validation	Cross Validation
장점	실행 속도가 빠르다(fit을 한 번 밖에 안 함)	실행 속도가 느리다(조각만큼 fit을 해줘야 함)
단점	train과 valid를 균등하게 나눠주지 않으면 점수의 정확도가 떨어질 수 있다	train과 valid를 균등하게 나눠주지 않아도 점수의 정확도가 크게 떨어지지 않는다

Evaluation Metric

Evaluation Metric

예측한 모델이 잘 구현되었는지, 이전에 구현한 모델에 비해 발전하였는지를 ‘정량적으로’ 측정하기 위해 측정 공식(Evaluation Metric)을 사용한다.

3783	▼ 323	xianglicun		0.78468	3	2mo
3784	▼ 323	Jenelyn Tidalgo		0.78468	1	2mo
3785	▼ 323	Shayne Kang JP		0.78468	3	1mo
<div>Your Best Entry ↑</div> <div>Your submission scored 0.75598, which is not an improvement of your best score. Keep trying.</div>						
3786	▼ 323	Tomasz Sikora		0.78468	3	2mo
3787	▼ 323	rucizaihuni00		0.78468	1	2mo
3788	▼ 323	Arijit Basak		0.78468	3	2mo
3789	▼ 323	Thomas HelmKay		0.78468	5	2mo
3790	▼ 323	Dinesh Kalithasan		0.78468	6	2mo

타이타닉 경진대회에서 측정한 결과를
측정 공식 중의 하나인 정확도(accuracy)라고 한다

Evaluation Metric (for regression)

Regression 문제에서는 정답(actual, a)과 예측값(predict, p)의 차이를 비교하는데, 이 수치가 0에 가까울수록 좋은 모델이라 판단하고, 0에서 멀어질수록 안 좋은 모델이라 판단합니다.

1. Mean Absolute Error(MAE)

$$\frac{1}{n} \sum_{t=1}^n |p^t - a^t|$$

예시

정답(actual) = 100대
모델 A의 예측 = 120대
모델 B의 예측 = 70대

Evaluation Metric (for regression)

Regression 문제에서는 정답(actual, a)과 예측값(predict, p)의 차이를 비교하는데, 이 수치가 0에 가까울수록 좋은 모델이라 판단하고, 0에서 멀어질수록 안 좋은 모델이라 판단합니다.

1. Mean Absolute Error(MAE)

$$\frac{1}{n} \sum_{t=1}^n |p^t - a^t|$$

간략화 버전

$$|p - a|$$

예시

정답(actual) = 100대
모델 A의 예측 = 120대
모델 B의 예측 = 70대

Evaluation Metric (for regression)

Regression 문제에서는 정답(actual, a)과 예측값(predict, p)의 차이를 비교하는데, 이 수치가 0에 가까울수록 좋은 모델이라 판단하고, 0에서 멀어질수록 안 좋은 모델이라 판단합니다.

간략화 버전

1. Mean Absolute Error(MAE)

$$\frac{1}{n} \sum_{t=1}^n |p^t - a^t|$$

$$|p - a|$$

2. Mean Squared Error(MSE)

$$\frac{1}{n} \sum_{t=1}^n (p^t - a^t)^2$$

$$(p - a)^2$$

예시

정답(actual) = 100대
모델 A의 예측 = 120대
모델 B의 예측 = 70대

Evaluation Metric (for regression)

Regression 문제에서는 정답(actual, a)과 예측값(predict, p)의 차이를 비교하는데, 이 수치가 0에 가까울수록 좋은 모델이라 판단하고, 0에서 멀어질수록 안 좋은 모델이라 판단합니다.

간략화 버전

1. Mean Absolute Error(MAE)

$$\frac{1}{n} \sum_{t=1}^n |p^t - a^t|$$

$$|p - a|$$

2. Mean Squared Error(MSE)

$$\frac{1}{n} \sum_{t=1}^n (p^t - a^t)^2$$

$$(p - a)^2$$

3. Root Mean Squared Error(RMSE)

$$\sqrt{\frac{1}{n} \sum_{t=1}^n (p^t - a^t)^2}$$

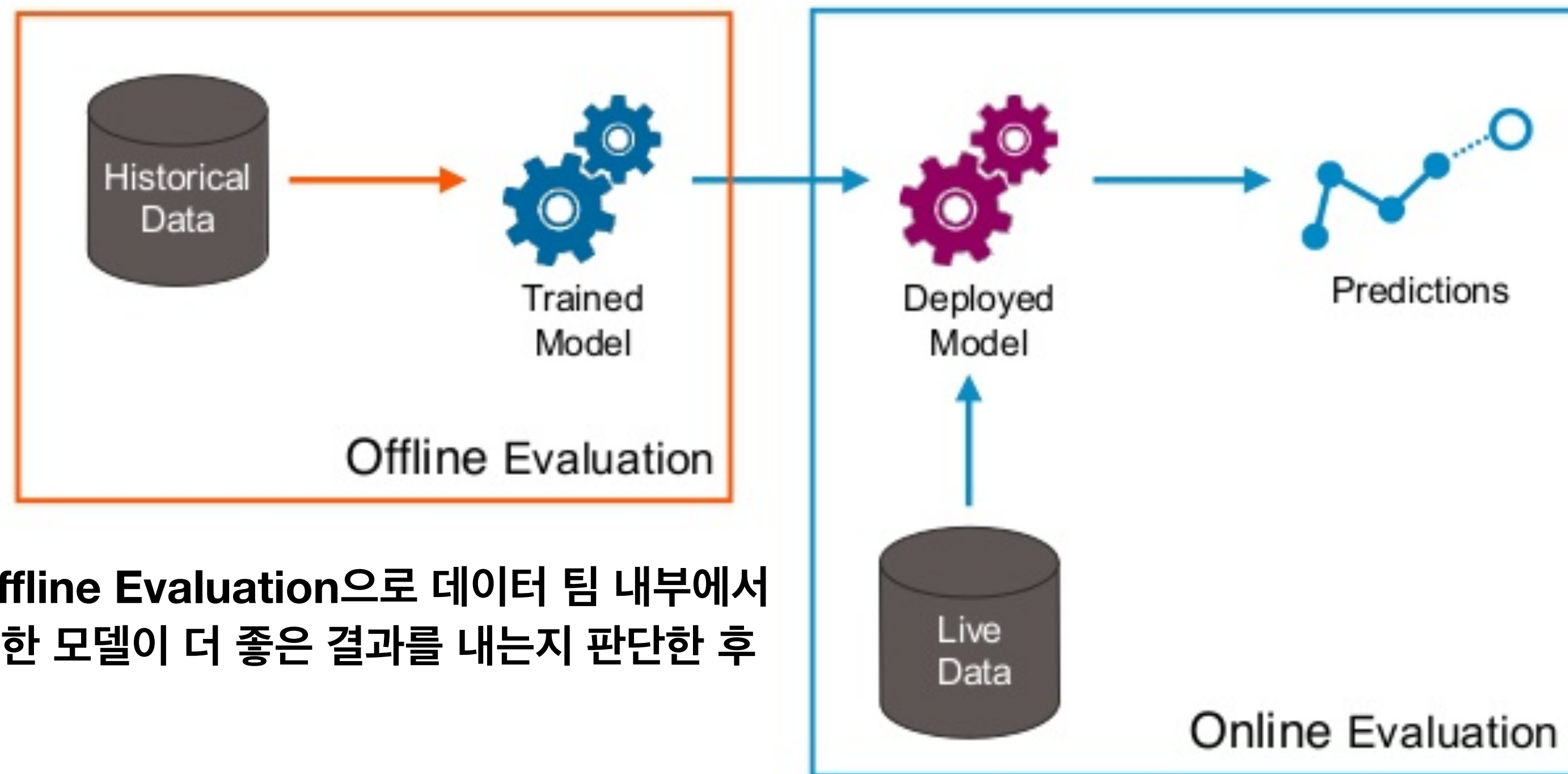
$$\sqrt{(p - a)^2}$$

예시

정답(actual) = 100대
모델 A의 예측 = 120대
모델 B의 예측 = 70대

Evaluation Metric

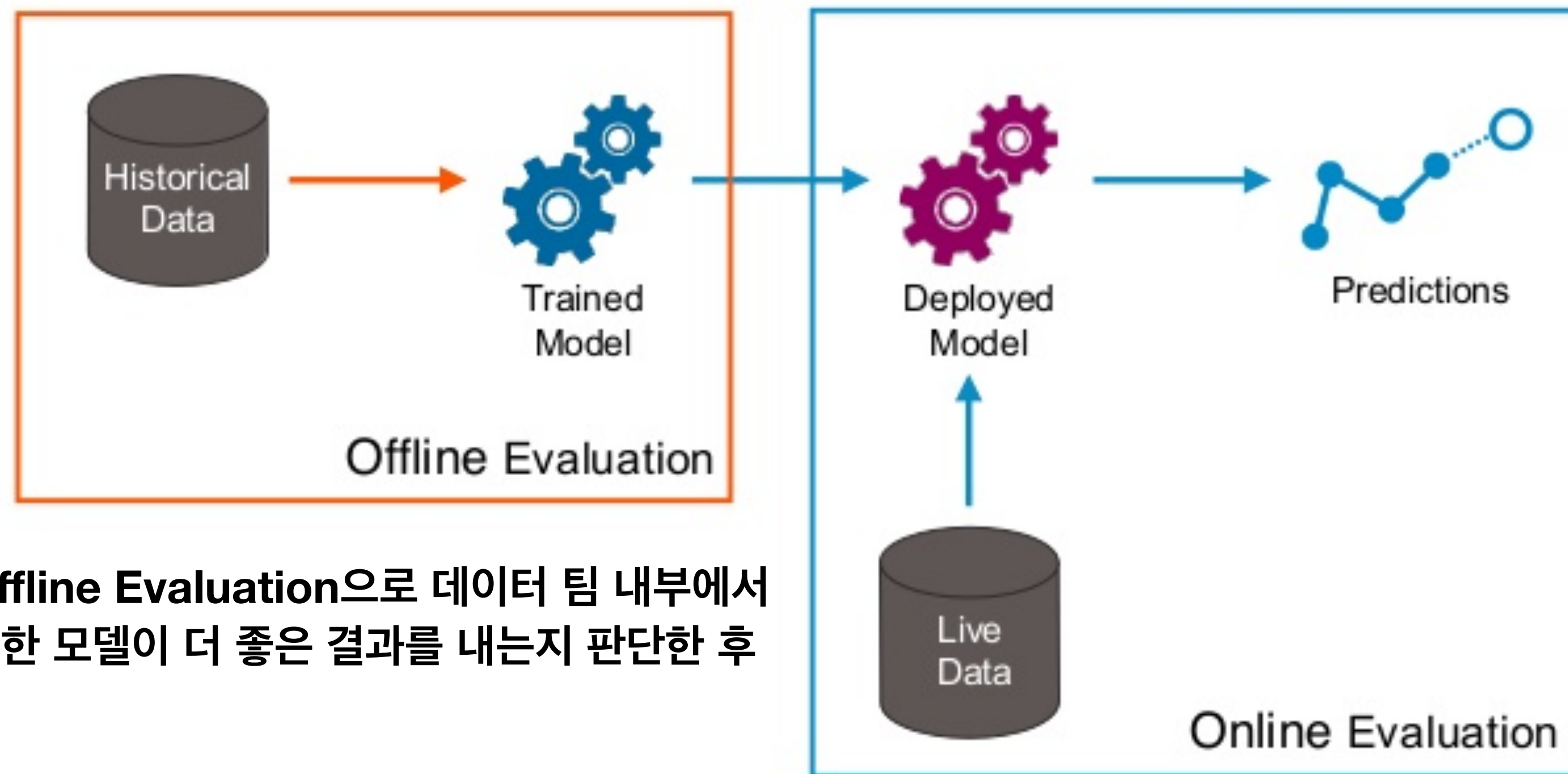
앞서 설명한 측정 공식을 오프라인 지표(offline metric)이라고 하는데, 회사(또는 부서, 서비스)에서 일반적으로 사용하는 온라인 지표(online metric)와 가장 일치하는 오프라인 지표를 사용하면 됩니다.



1. Offline Evaluation으로 데이터 팀 내부에서 수정한 모델이 더 좋은 결과를 내는지 판단한 후

Evaluation Metric

앞서 설명한 측정 공식을 오프라인 지표(offline metric)이라고 하는데, 회사(또는 부서, 서비스)에서 일반적으로 사용하는 온라인 지표(online metric)와 가장 일치하는 오프라인 지표를 사용하면 됩니다.

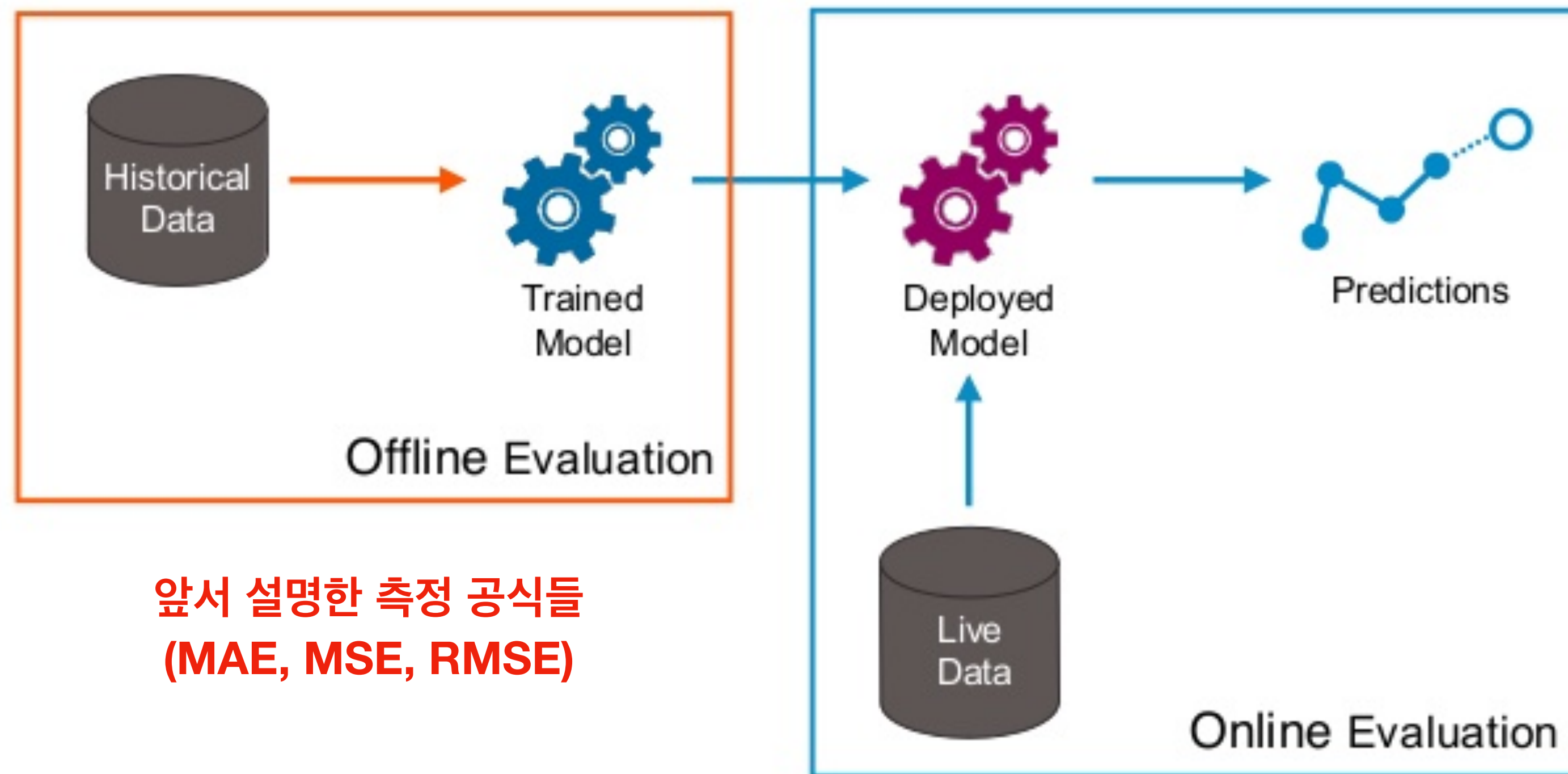


1. Offline Evaluation으로 데이터 팀 내부에서 수정한 모델이 더 좋은 결과를 내는지 판단한 후

2. Online Evaluation으로 실 서비스에서 비즈니스 지표를 기준으로 확실히 검증한다

Evaluation Metric

앞서 설명한 측정 공식을 오프라인 지표(offline metric)이라고 하는데, 회사(또는 부서, 서비스)에서 일반적으로 사용하는 온라인 지표(online metric)와 가장 일치하는 오프라인 지표를 사용하면 됩니다.



앞서 설명한 측정 공식들
(MAE, MSE, RMSE)

회사(내지는 부서, 서비스)에서 사용하는 지표들(일명 KPI)
(LTV/CAC, Churn, 잔존 시간 등)

Root Mean Squared Logarithmic Error? (RMSLE)

Bike Sharing Demand에서는 위 세 가지 측정 공식이 아닌 새로운 측정 공식을 사용하는데, 이를 **Root Mean Squared Logarithmic Error (RMSLE)**라고 한다.

Submissions are evaluated on the Root Mean Squared Logarithmic Error (RMSLE). The RMSLE is calculated as

$$\sqrt{\frac{1}{n} \sum_{i=1}^n (\log(p_i + 1) - \log(a_i + 1))^2}$$

Where:

- n is the number of hours in the test set
- p_i is your predicted count
- a_i is the actual count
- $\log(x)$ is the natural logarithm

Root Mean Squared Logarithmic Error? (RMSLE)

Bike Sharing Demand에서는 위 세 가지 측정 공식이 아닌 새로운 측정 공식을 사용하는데, 이를 **Root Mean Squared Logarithmic Error (RMSLE)**라고 한다.

Submissions are evaluated on the Root Mean Squared Logarithmic Error (RMSLE). The RMSLE is calculated as

$$\sqrt{\frac{1}{n} \sum_{i=1}^n (\log(p_i + 1) - \log(a_i + 1))^2}$$

Where:

- n is the number of hours in the test set
- p_i is your predicted count
- a_i is the actual count
- $\log(x)$ is the natural logarithm

$$\sqrt{\frac{1}{n} \sum_{t=1}^n (p^t - a^t)^2}$$

사실상 **Root Mean Squared Error(RMSE)**를 기반으로 **predict(p)**와 **actual(a)**에 **log + 1**을 씌운 것과 동일하다.