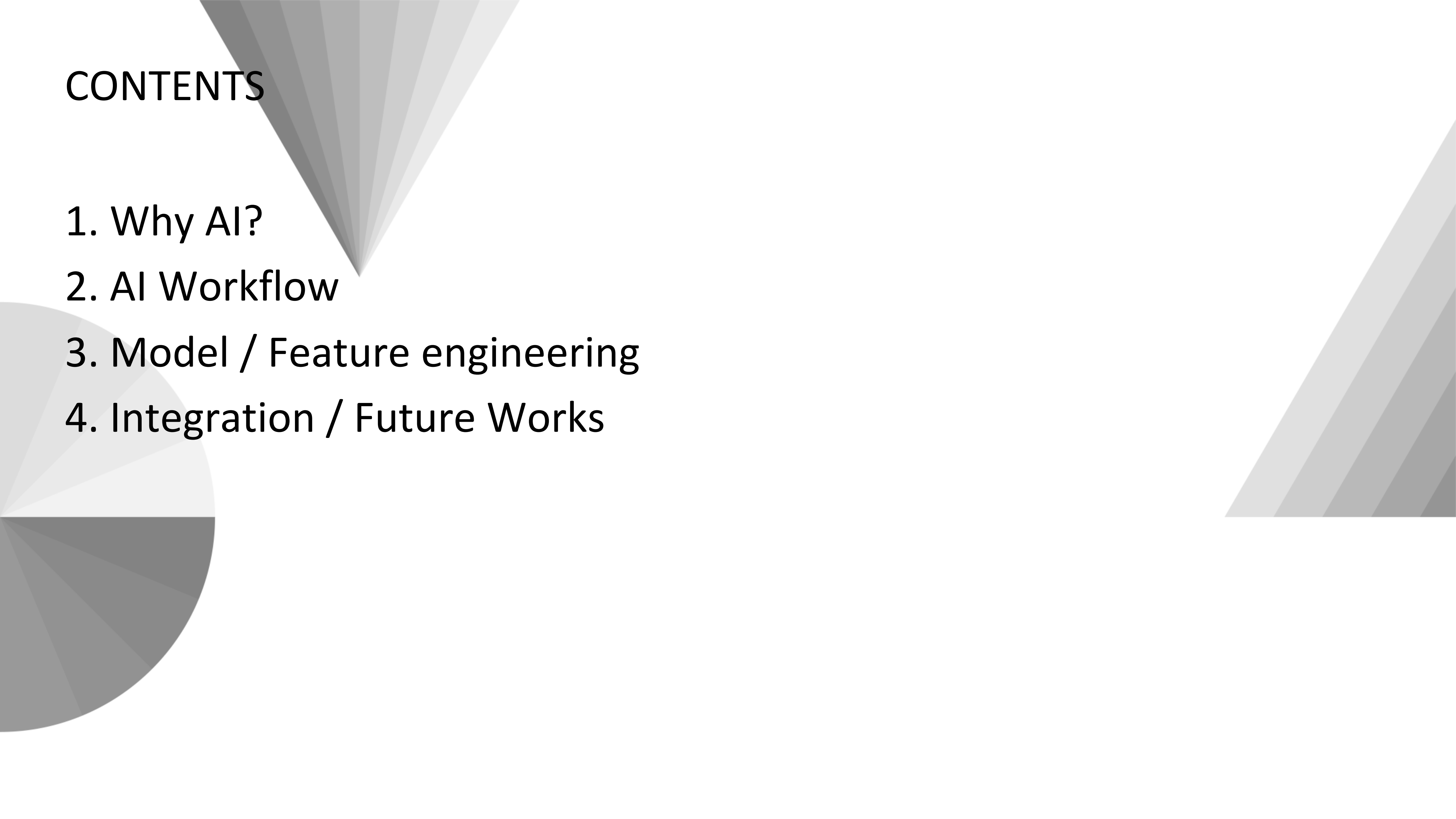


딥러닝 기술을 이용한 검색시스템 응답시간 예측

박기덕 NAVER / Search Serving Platform
김민수 NAVER / Search Serving Platform

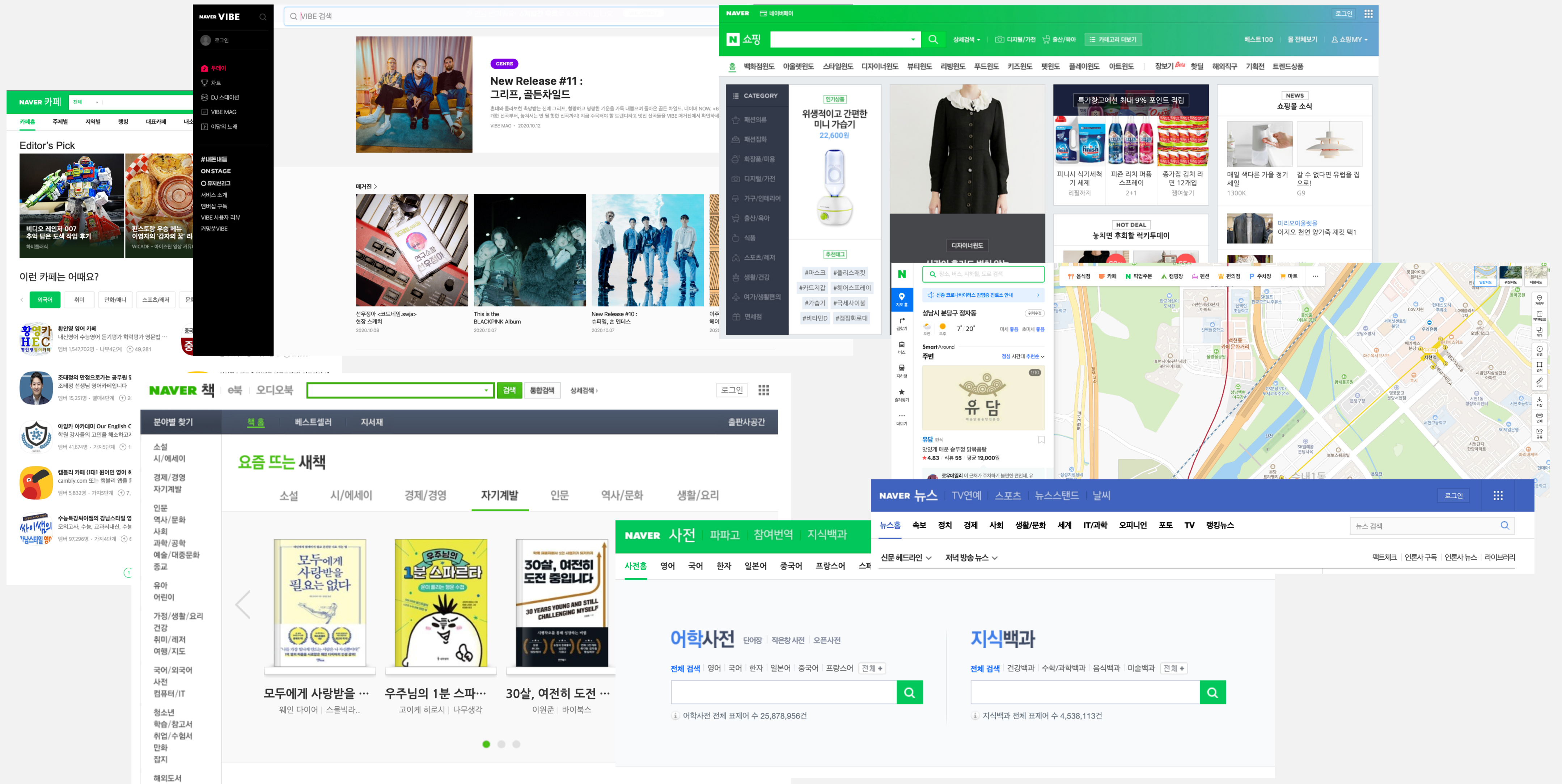


CONTENTS

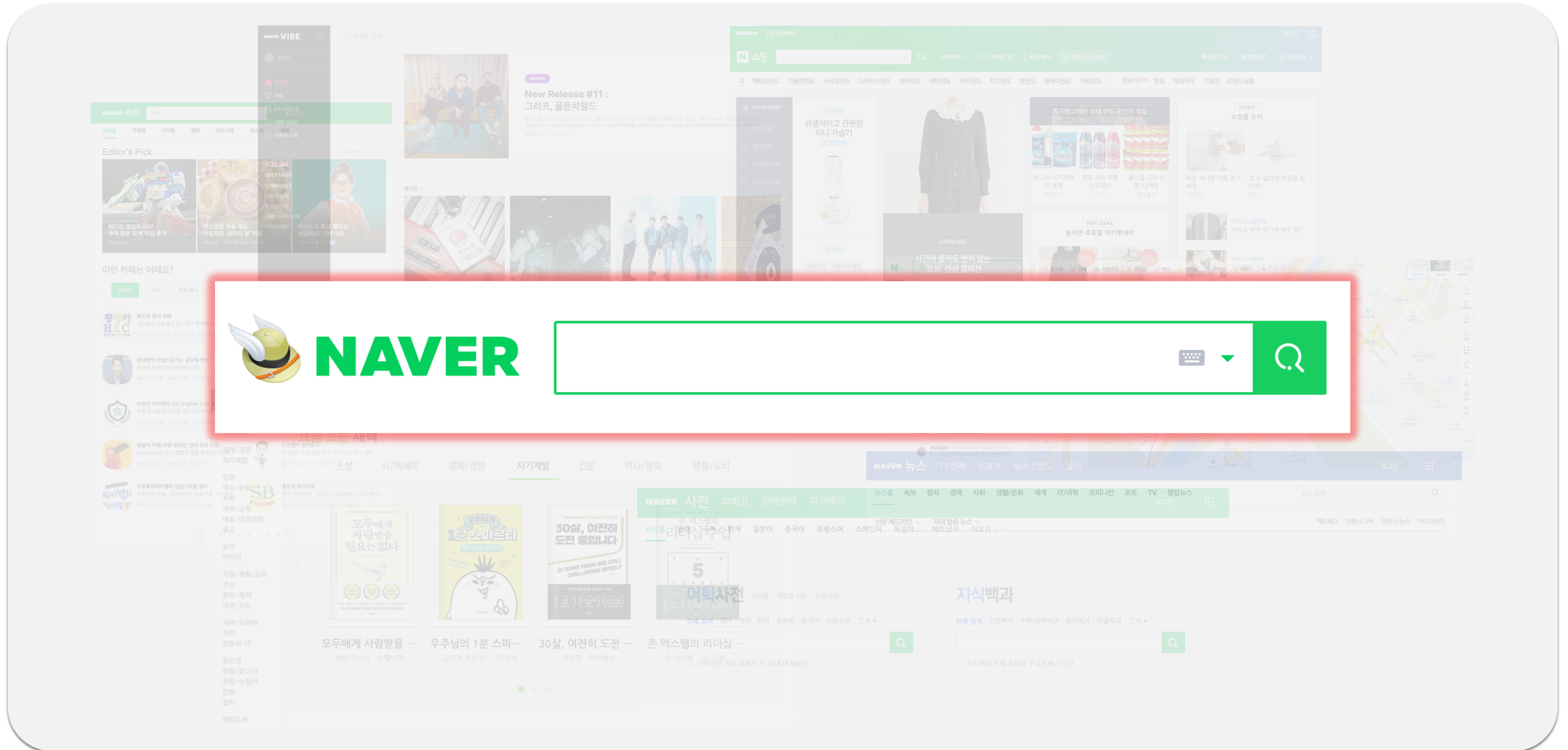
1. Why AI?
2. AI Workflow
3. Model / Feature engineering
4. Integration / Future Works

1. Why AI?

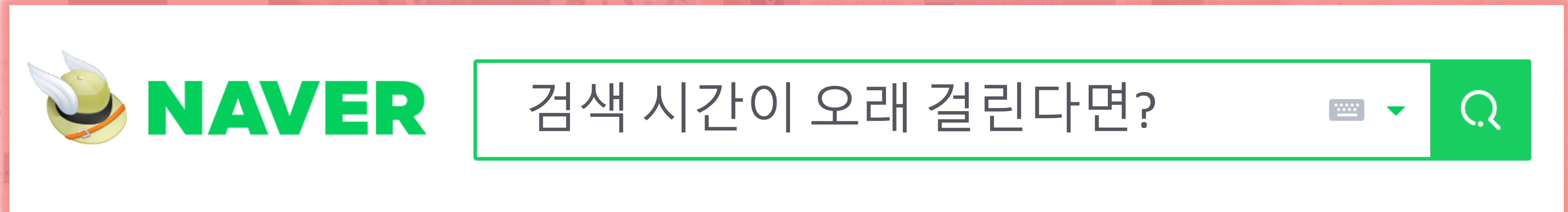
(휴리스틱에서 머신러닝으로)



1.1 네이버 검색 시스템



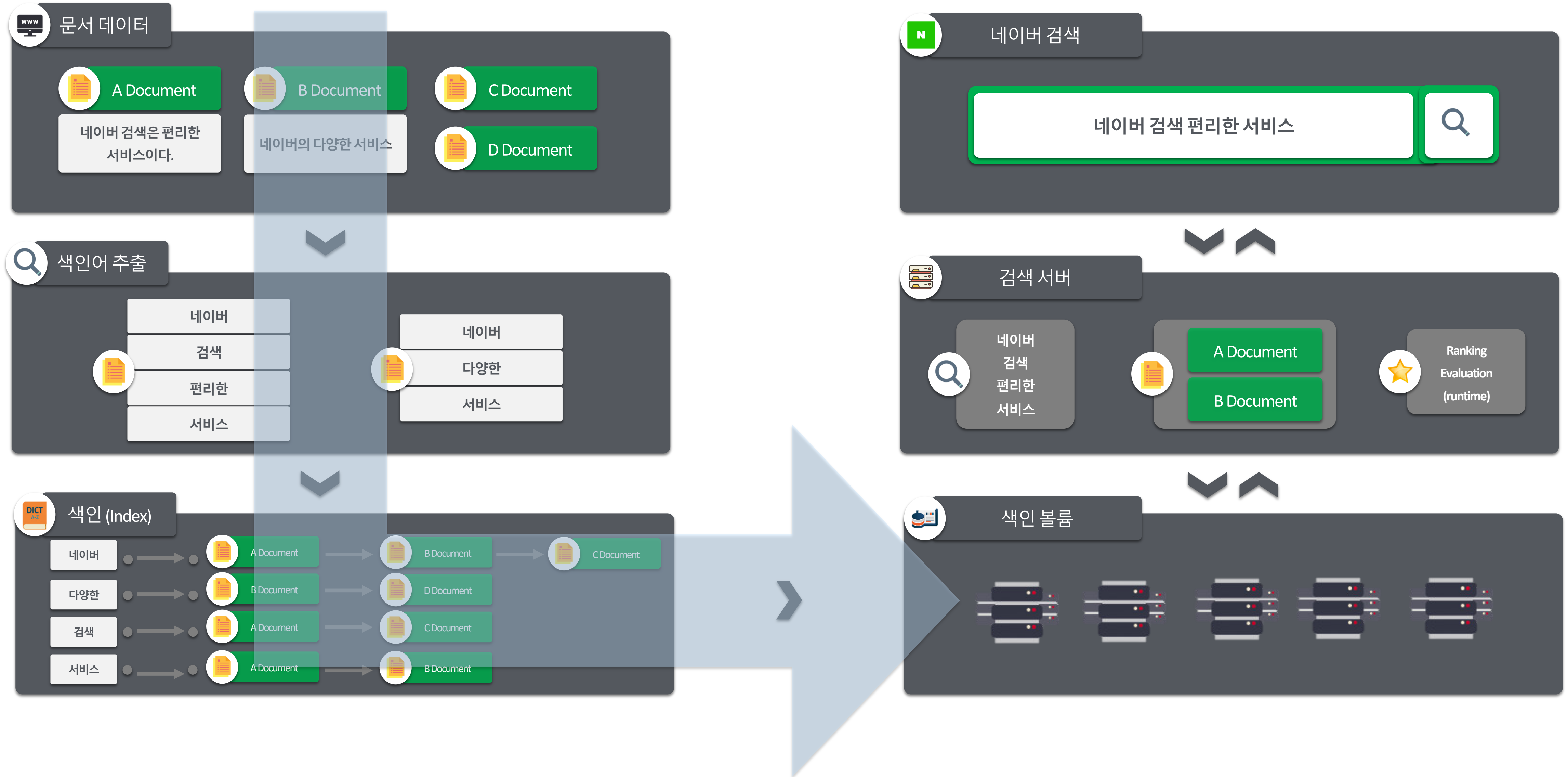
1.1 네이버 검색 시스템



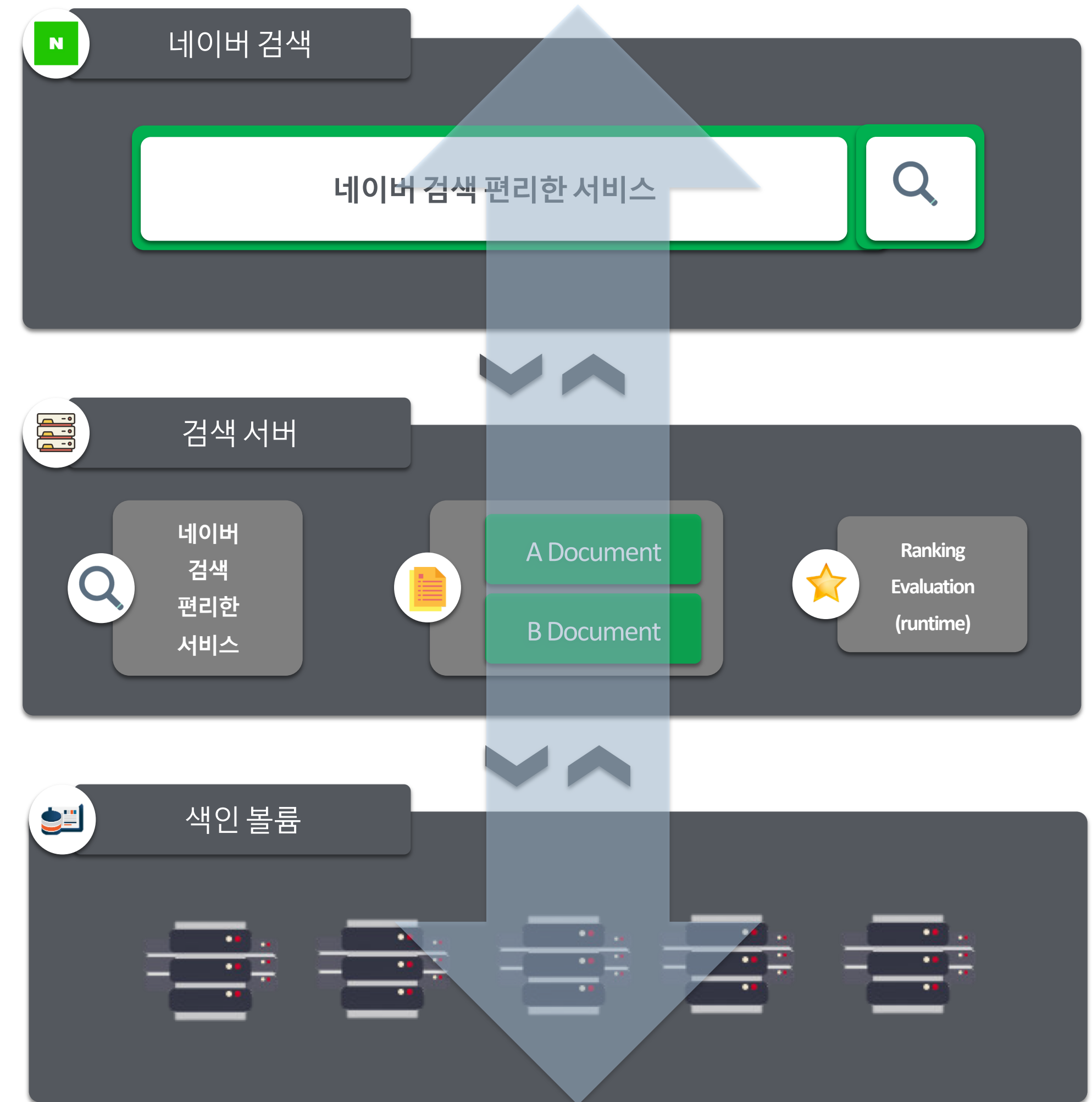
1.1 네이버 검색 시스템



1.1 네이버 검색 시스템



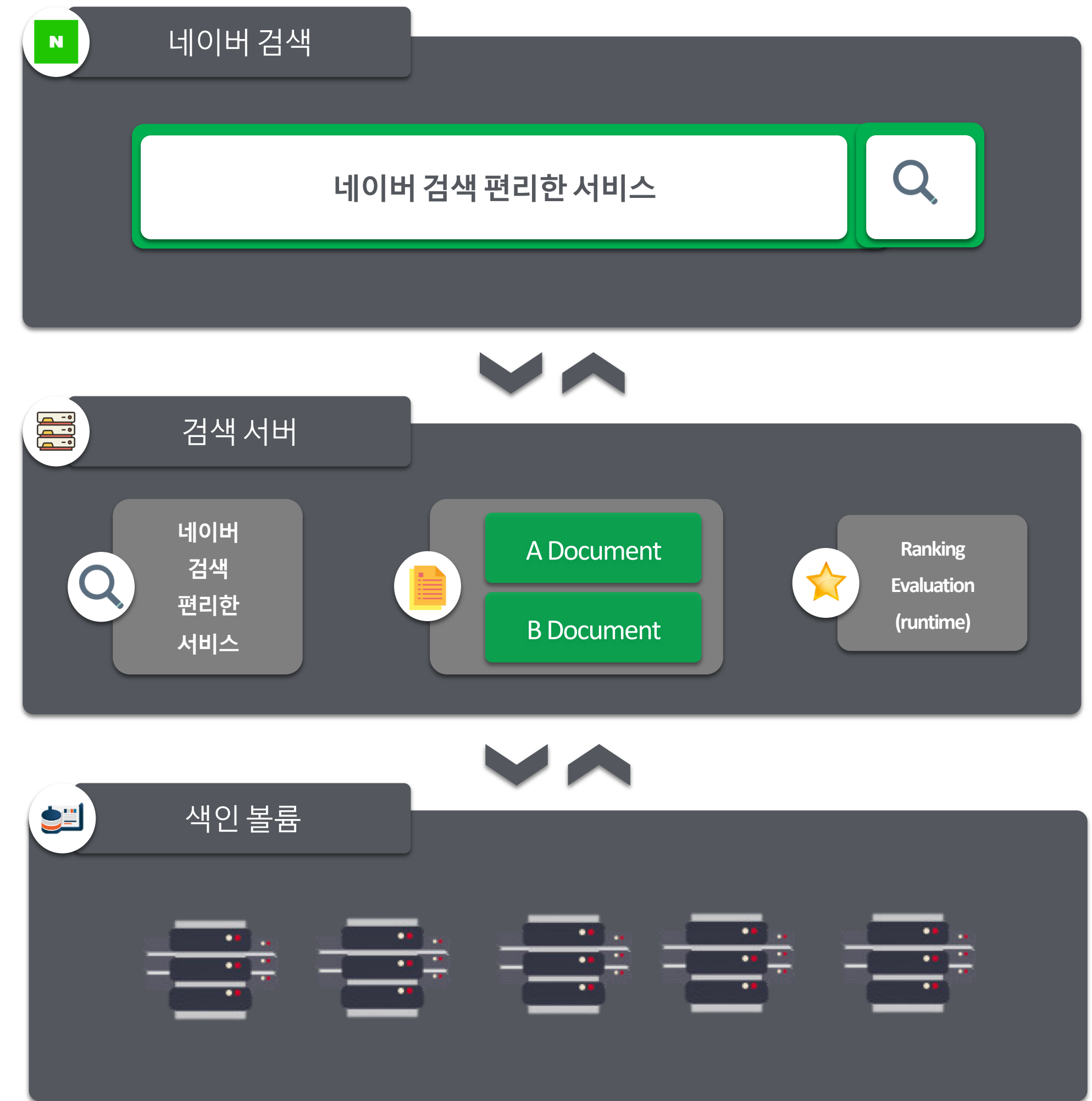
1.1 네이버 검색 시스템



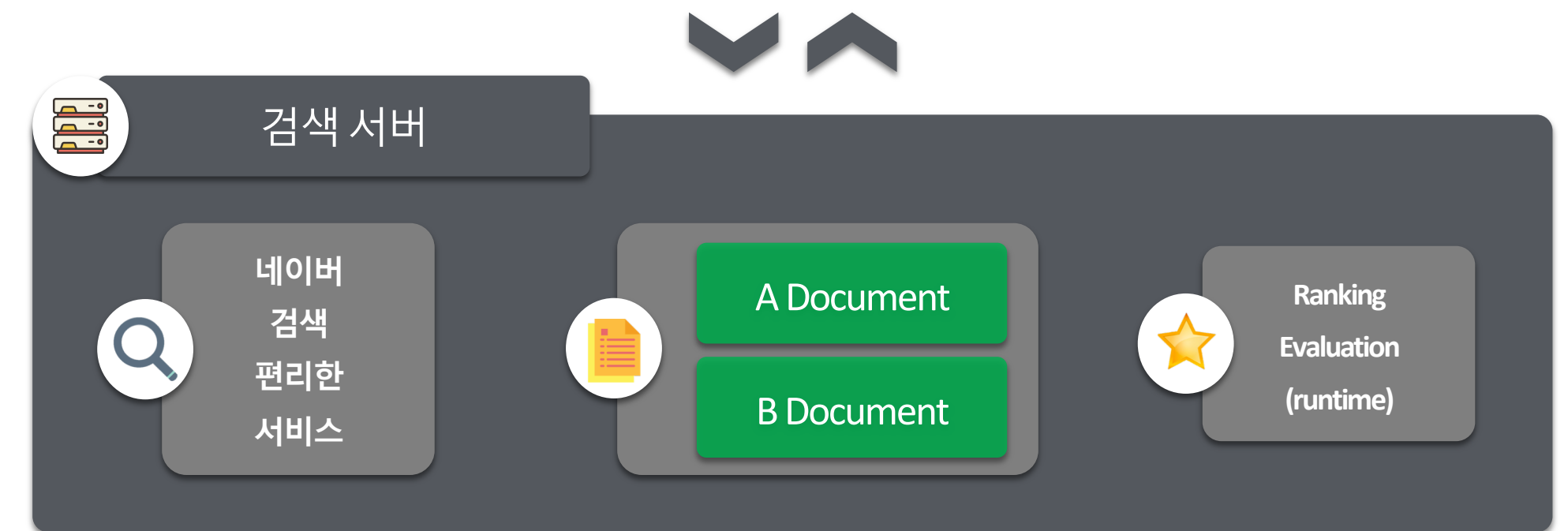
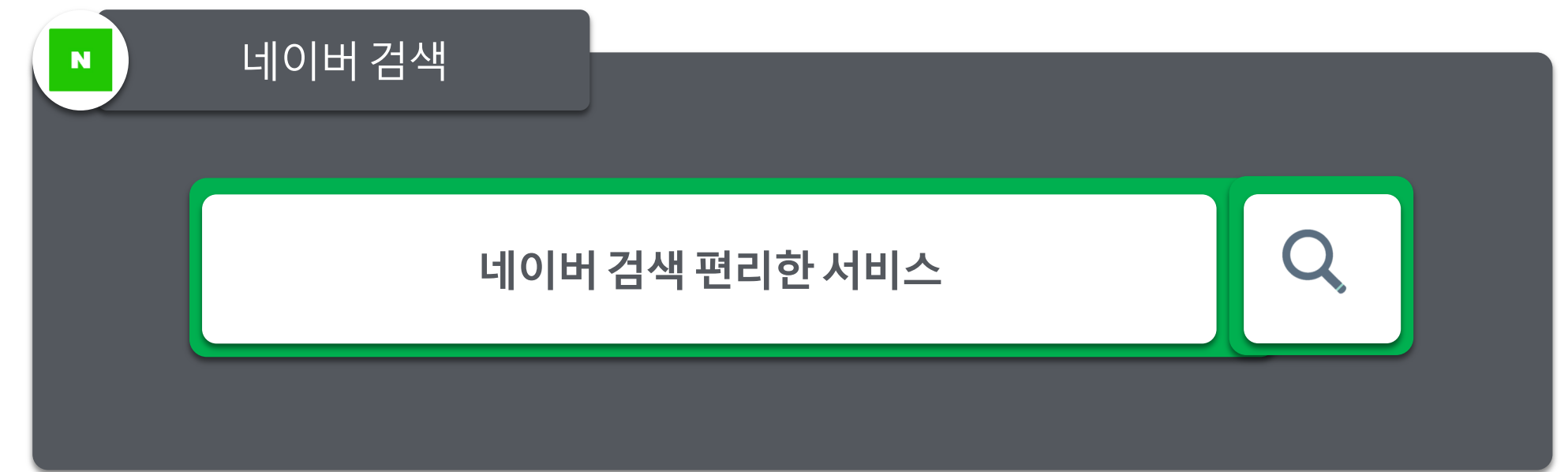
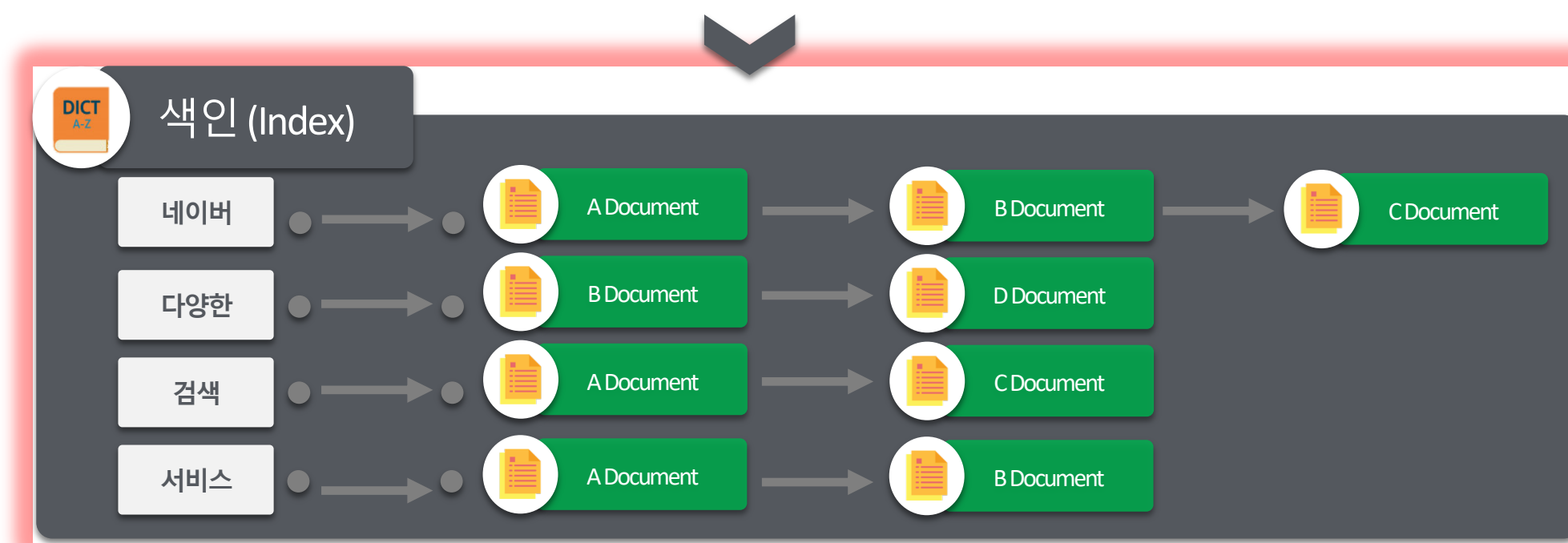
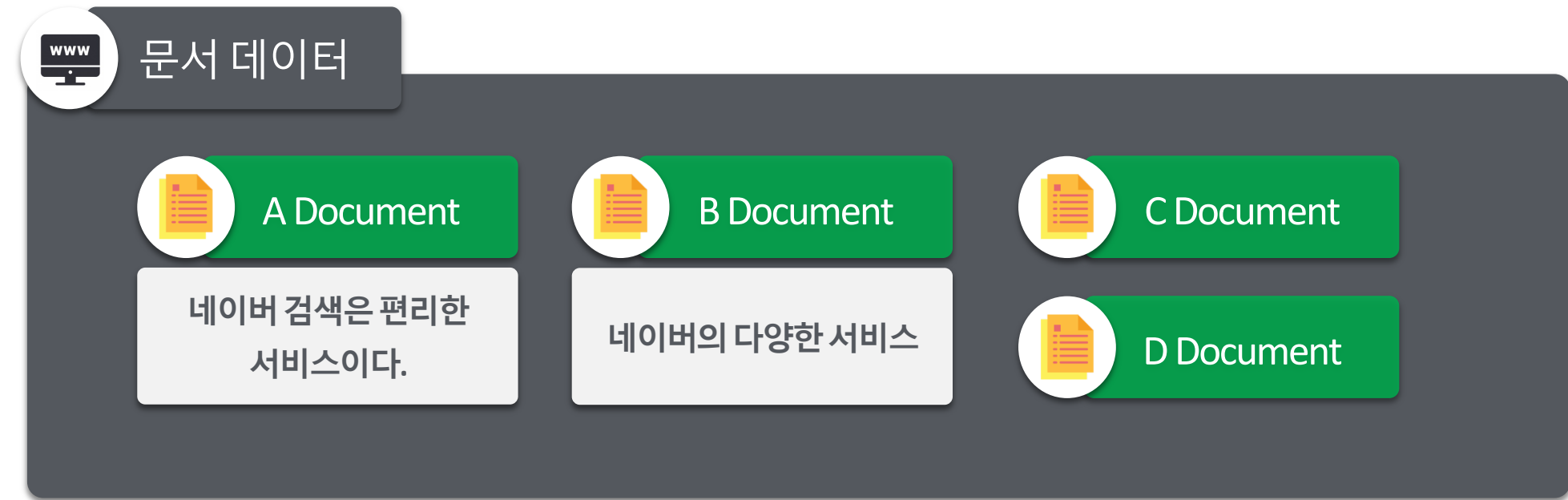
1.1 네이버 검색 시스템



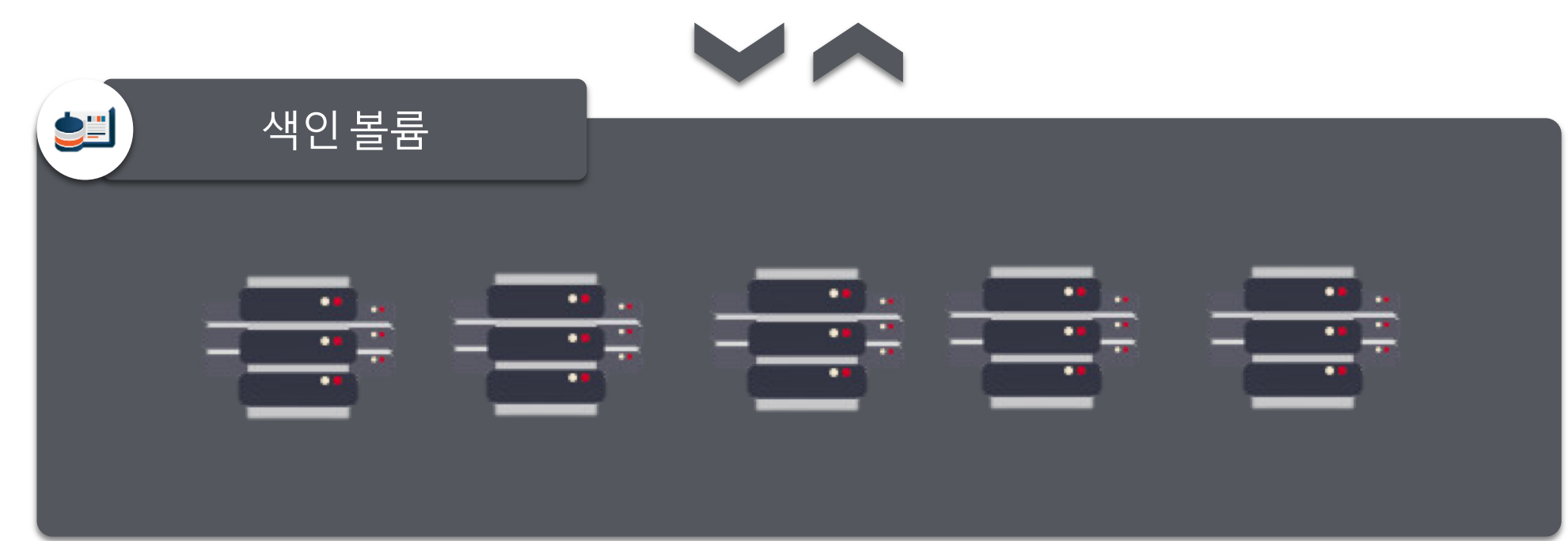
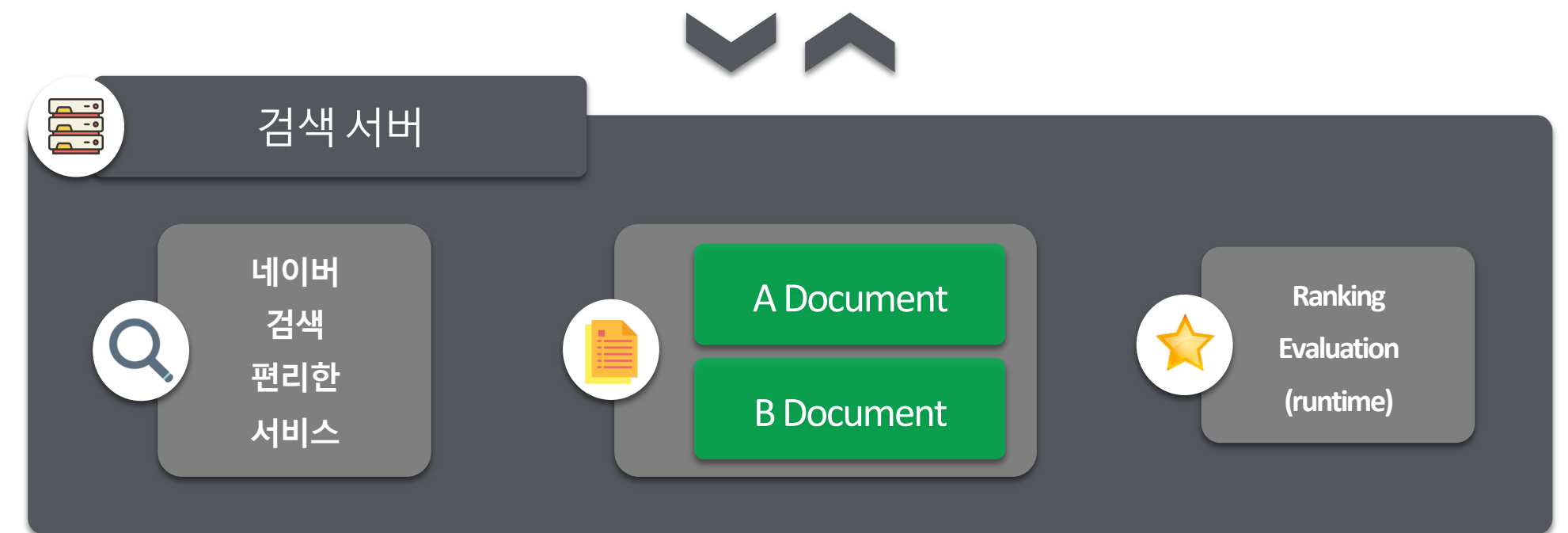
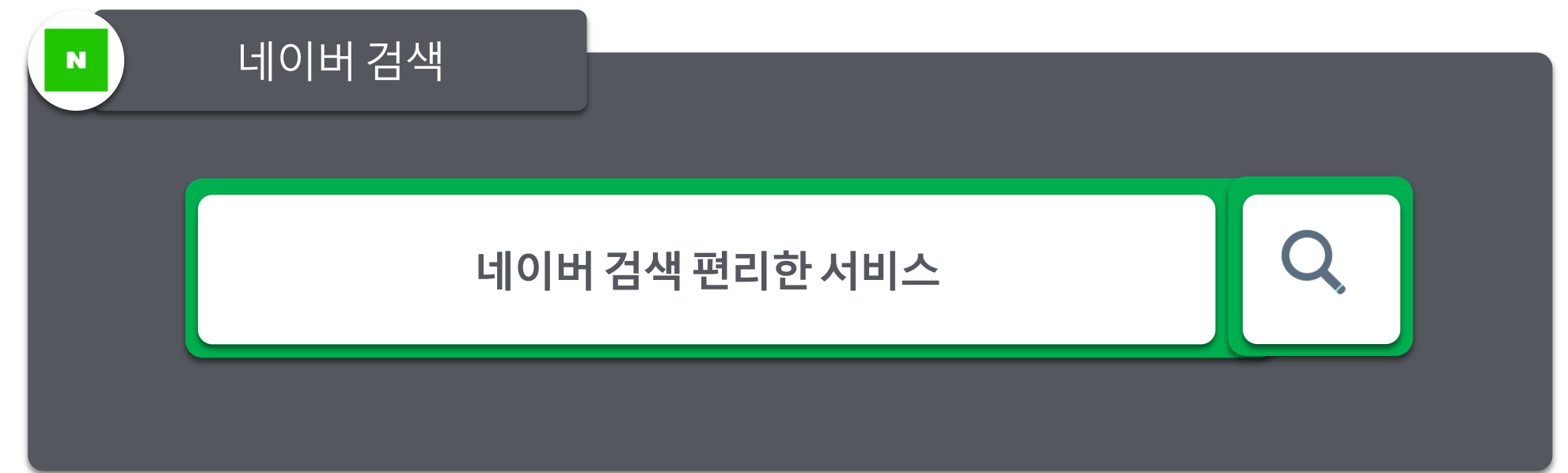
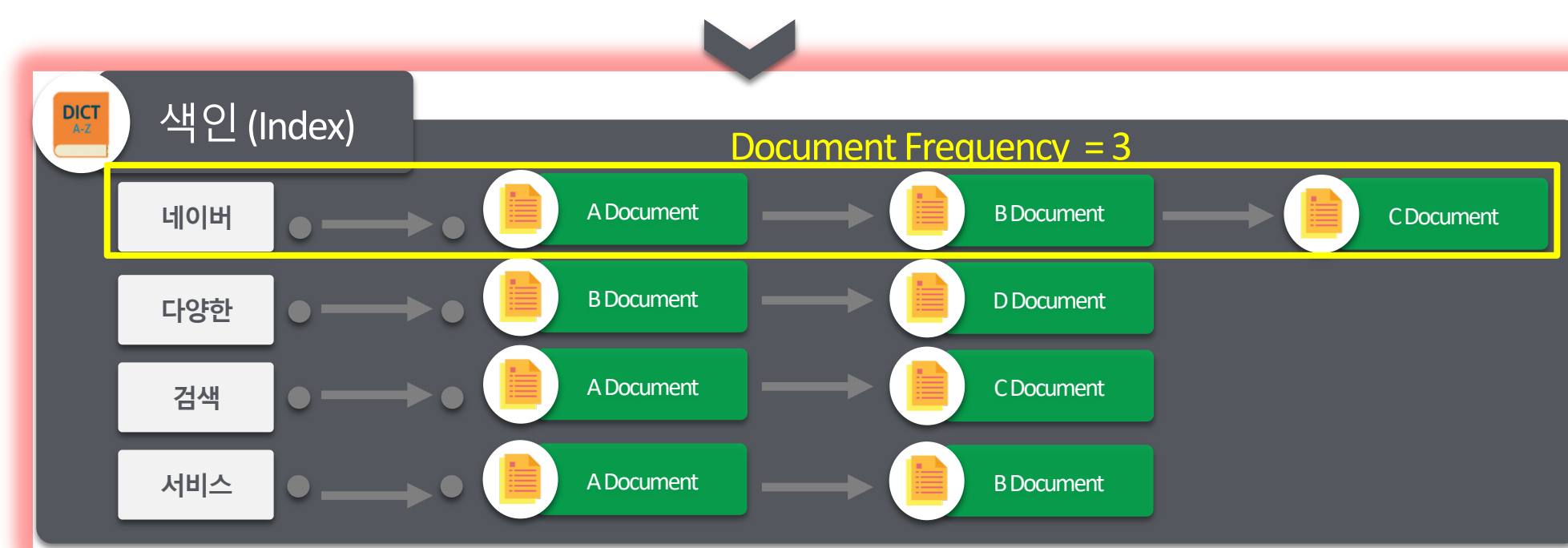
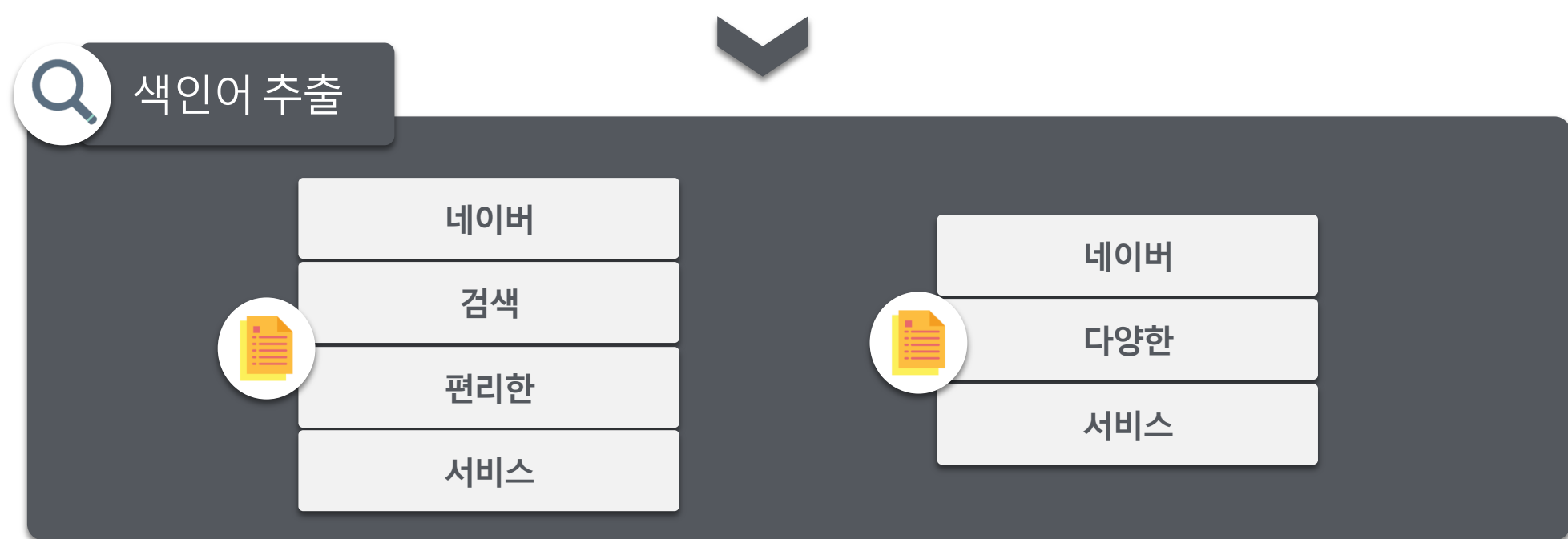
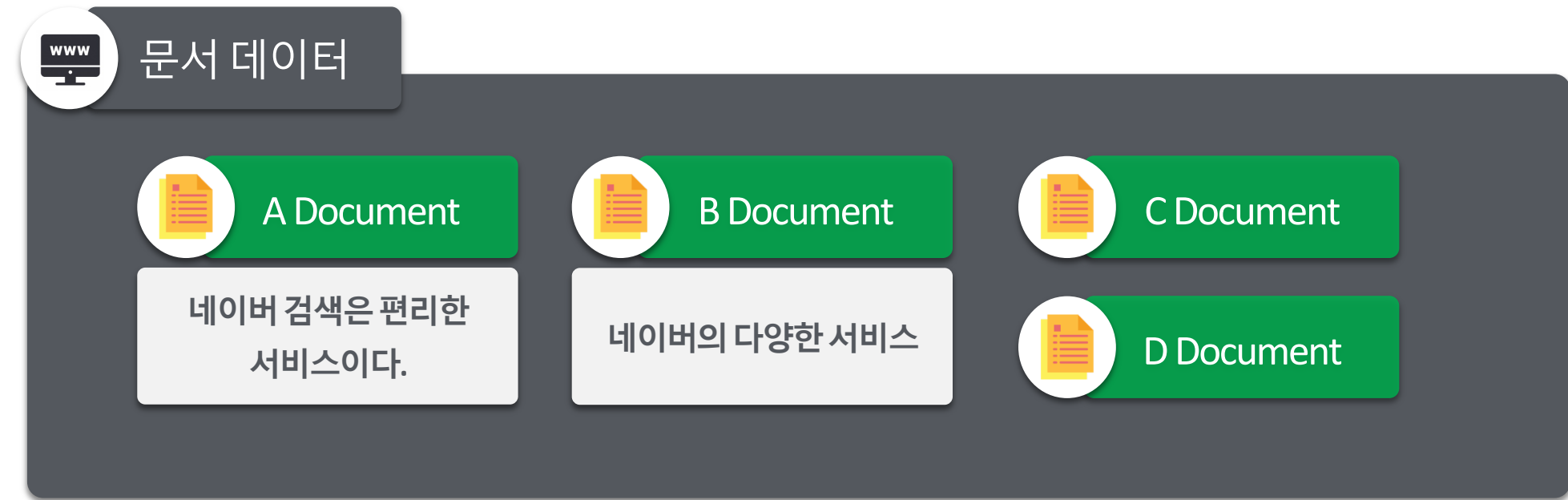
1.1 네이버 검색 시스템



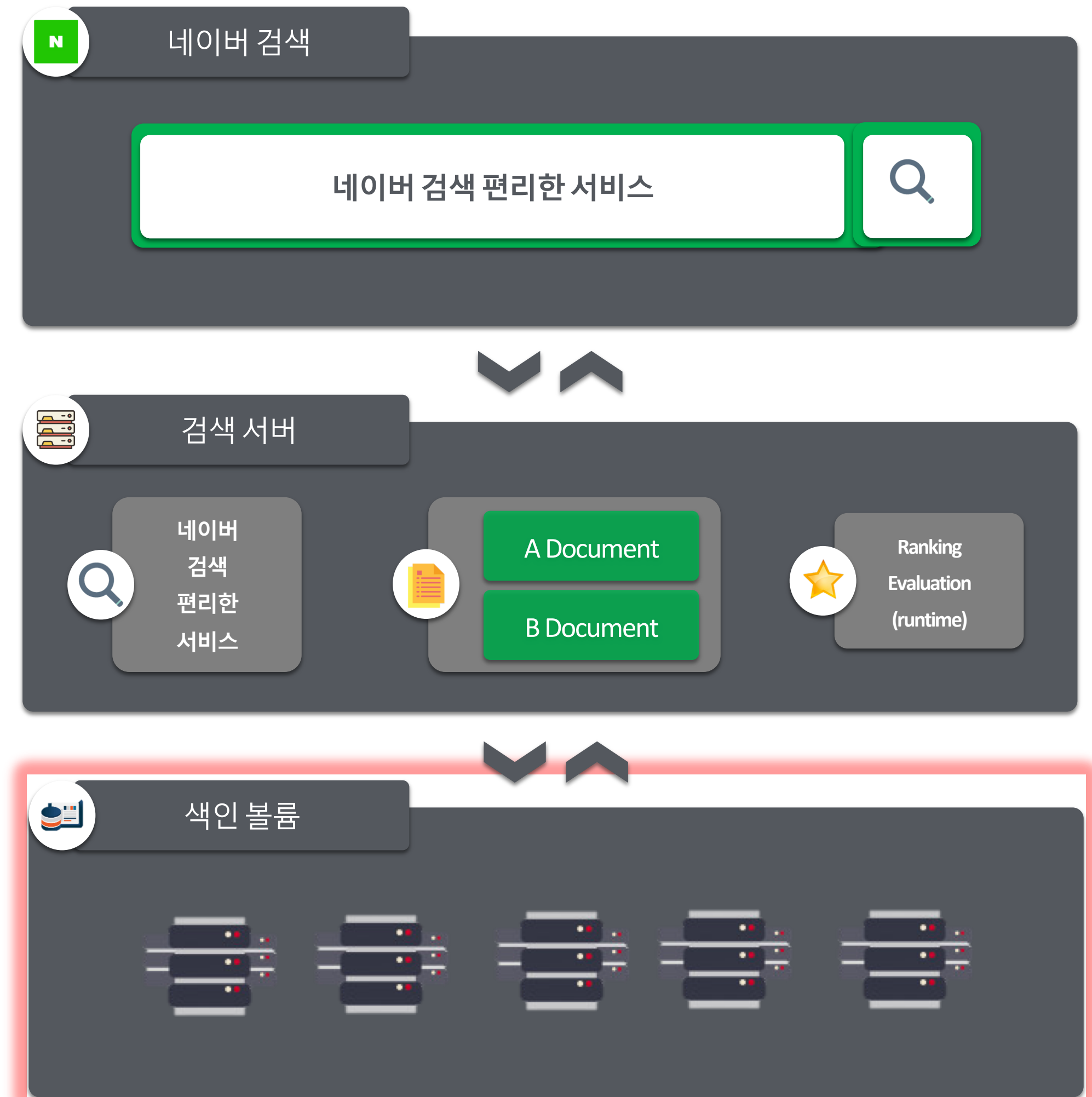
1.1 네이버 검색 시스템



1.1 네이버 검색 시스템



1.1 네이버 검색 시스템



1.1 네이버 검색 시스템



1.1 네이버 검색 시스템



1.1 네이버 검색 시스템



1.1 네이버 검색 시스템



검색어

검색 방식

검색 타겟

검색 옵션

다양한 검색



응용 서버



검색 서버

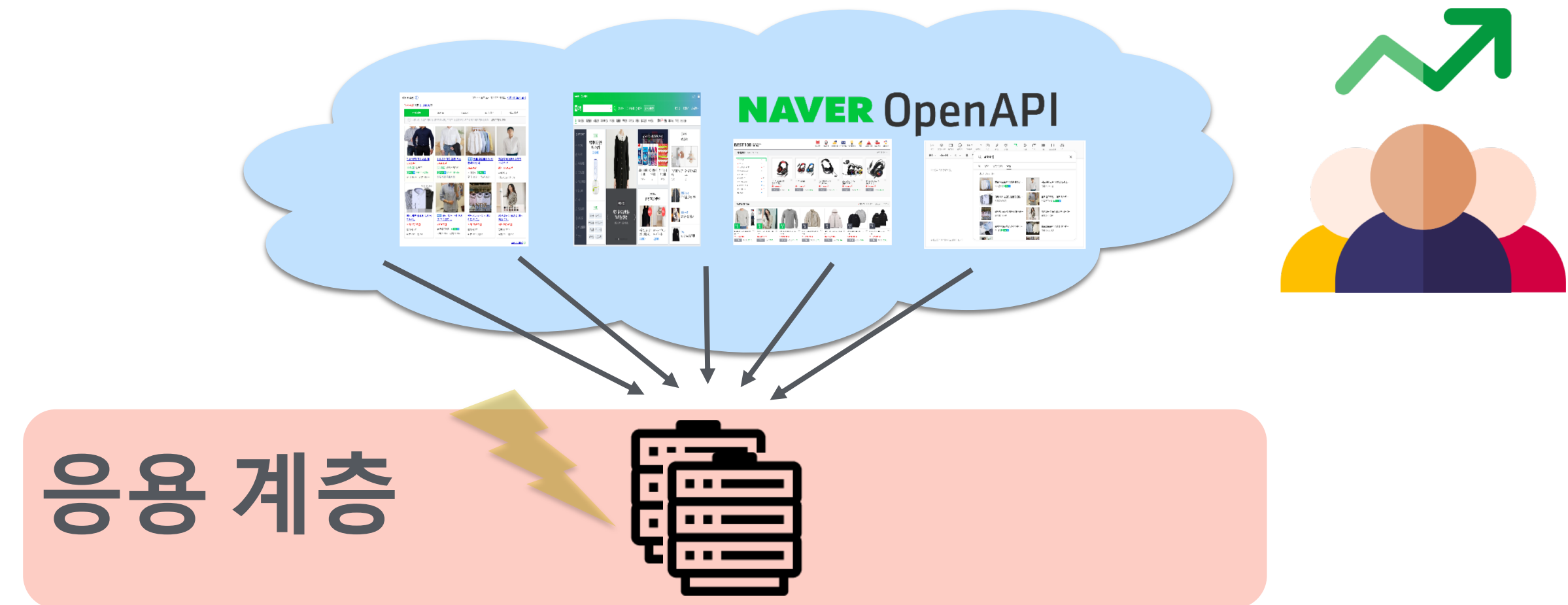


검색 볼륨

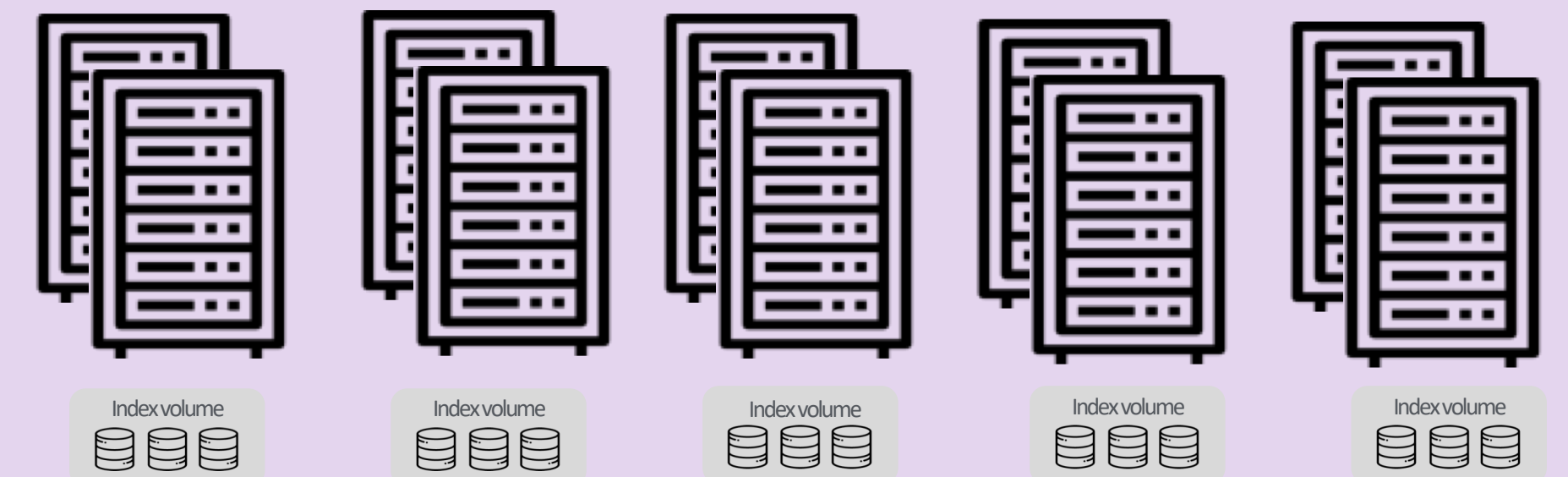
1.2 왜 검색 응답시간 예측이 필요할까?

검색 시스템

- 다양한 호출처로부터 검색 요청
- 검색어, 검색 방식, 검색 옵션에 따라서
수백 ~ 수천가지의 검색 유형이 발생
- 통합검색 0.9sec 타임아웃 룰



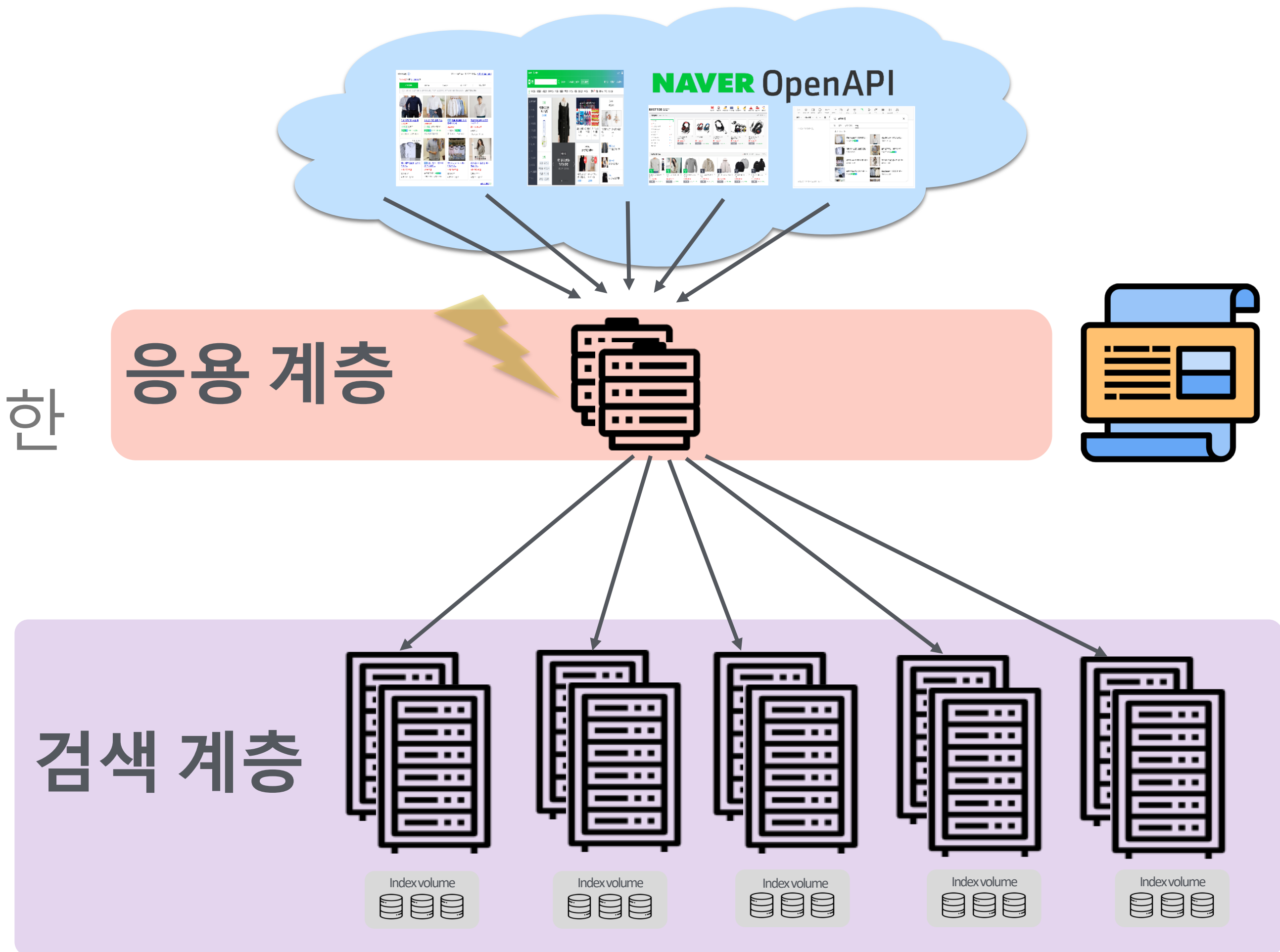
검색 계층



1.2 왜 검색 응답시간 예측이 필요할까?

검색 시스템

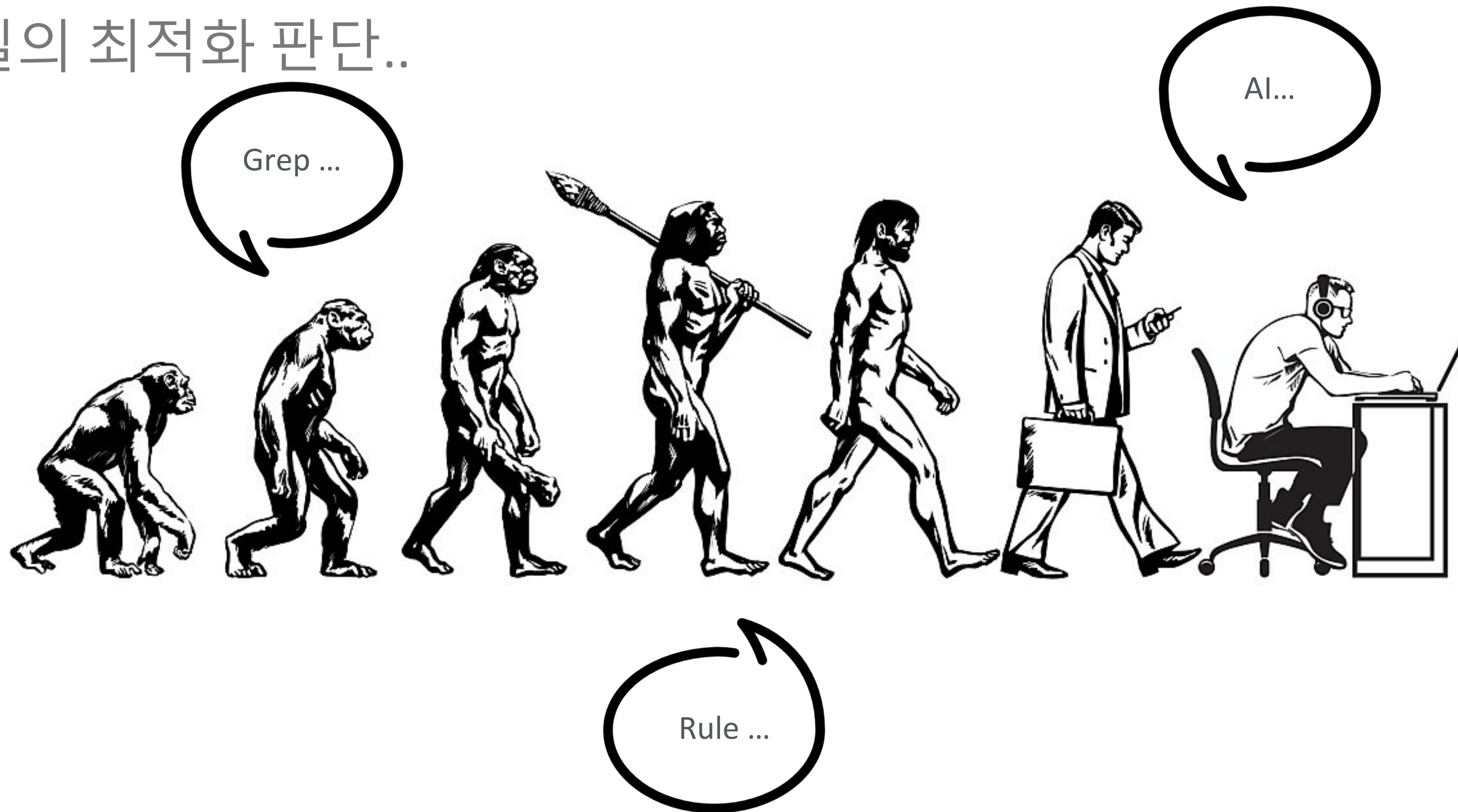
- 검색 유형에 따라 검색 서버의 컴퓨팅 비용, 응답시간 분포가 다름
- 초당 수천의 검색 요청을 처리하기 위한 높은 시스템 가용량 유지 필요
- 응용 서버의 로그를 모델 학습에 사용



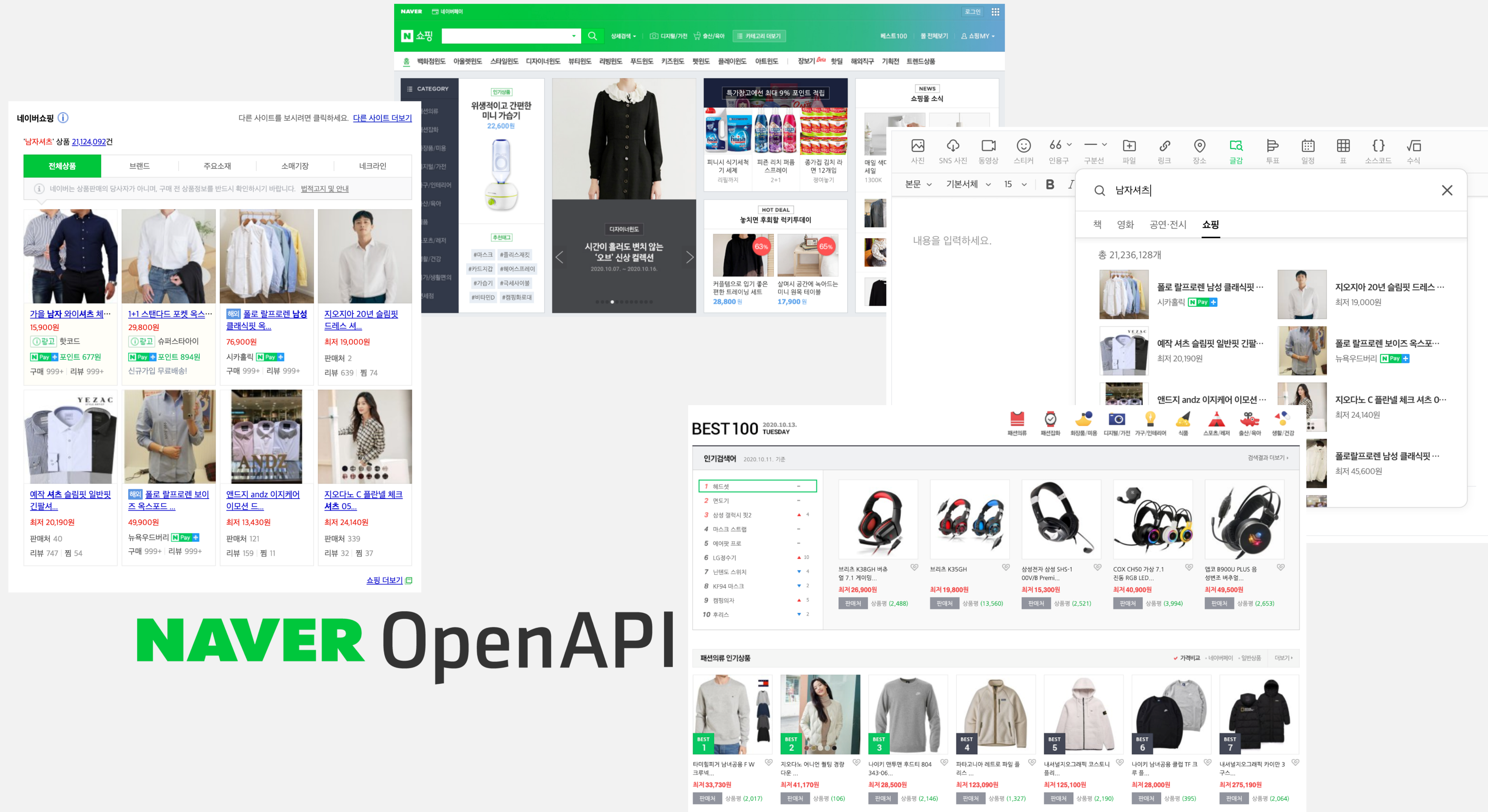
인간 시대의 끝이 도래했다

검색 시스템

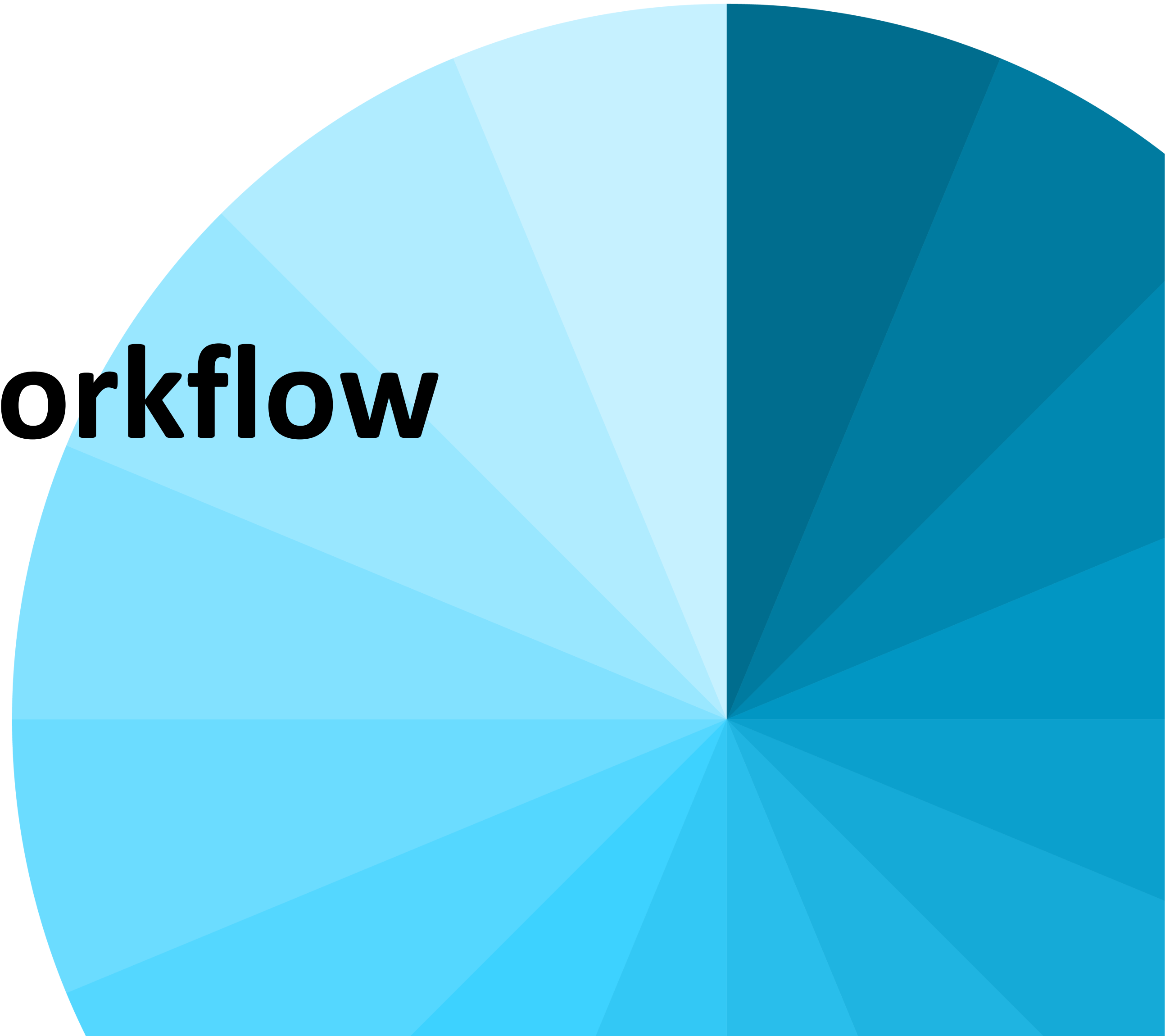
- 고비용 질의 최적화 판단..



1.3 쇼핑 검색



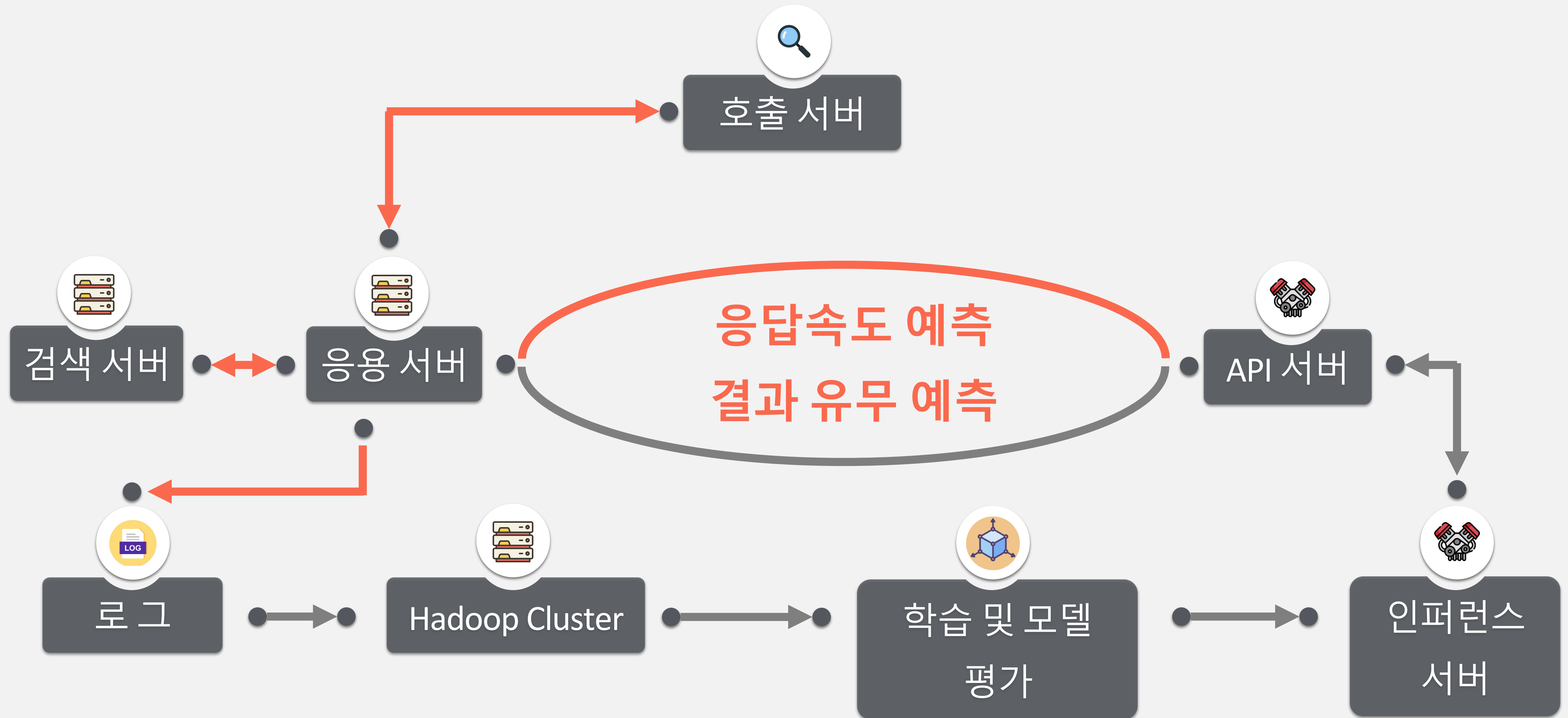
2. *AI Workflow*



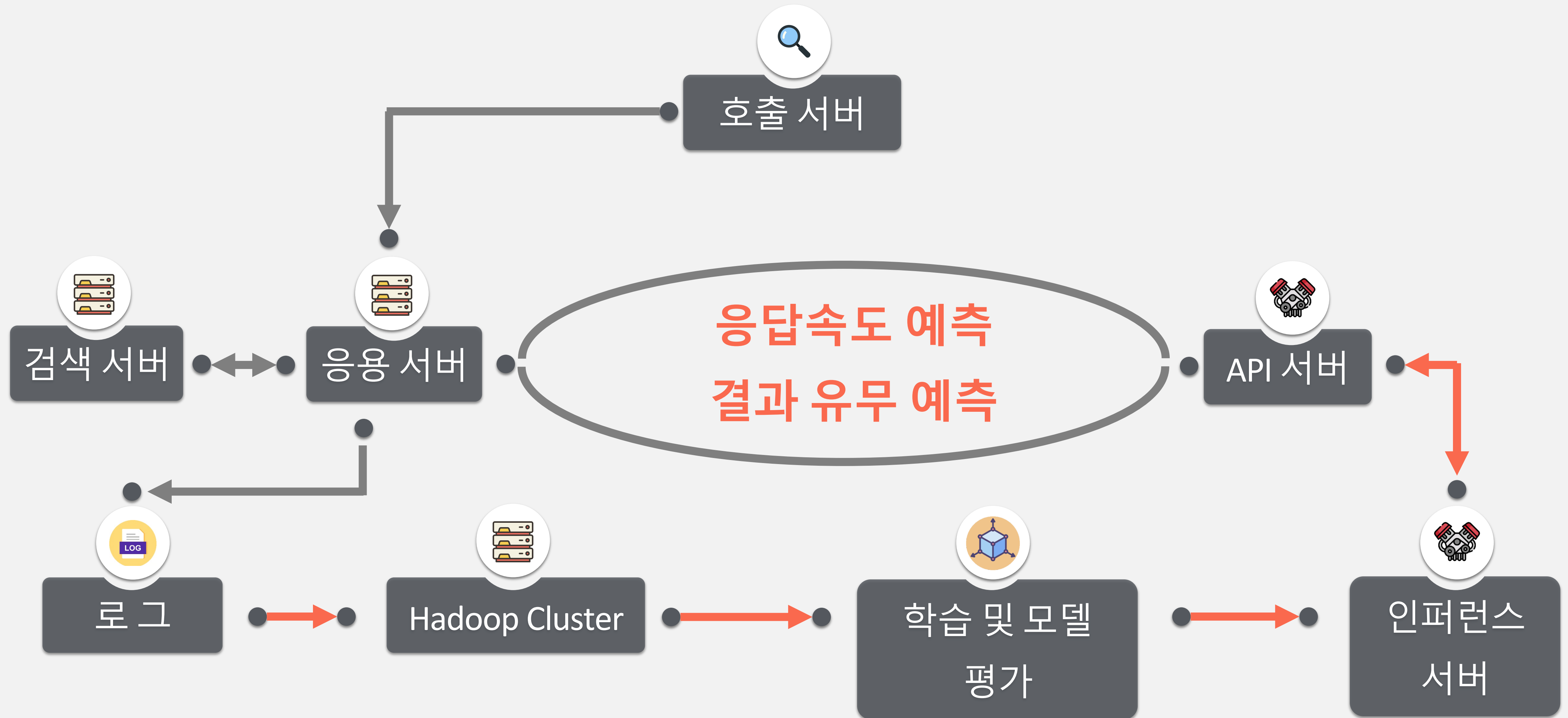
2.1 전체 구조



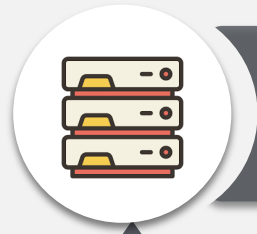
2.1 전체 구조



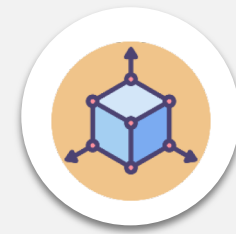
2.1 전체 구조



2.2 구조 설명



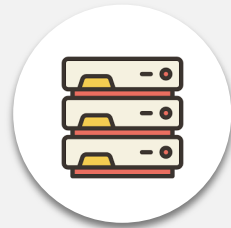
Hadoop Cluster



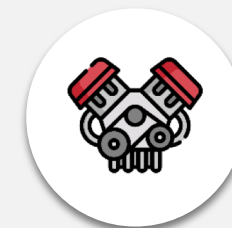
Hadoop Cluster

- 중복된 로그를 제거하여 학습 시간 단축
- DF 구간 별로 데이터 수집
- 로그 데이터를 학습 데이터 포맷으로 변경

2.2 구조 설명



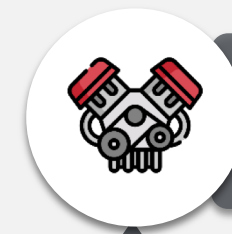
모델 학습 및 평가



모델 학습 및 평가

- Keras를 사용하여 모델 구현 및 평가
- 모델 평가는 학습에 참여하지 않은 데이터
- 평균 오차, 정확도를 측정

2.2 구조 설명



인퍼런스 서버 및 API 서버

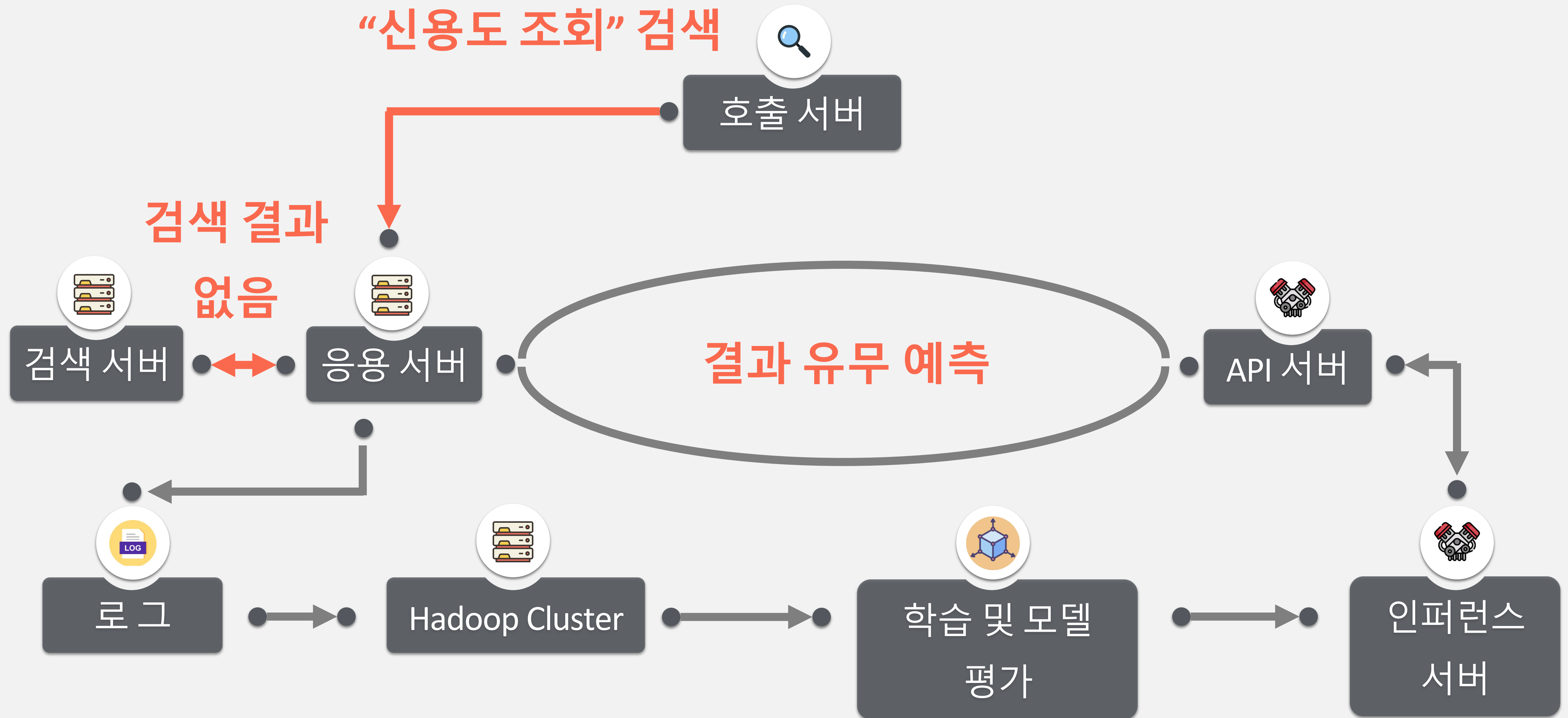
인퍼런스 서버

- 모델 평가에서 좋은 모델을 인퍼런스 서버로 전달

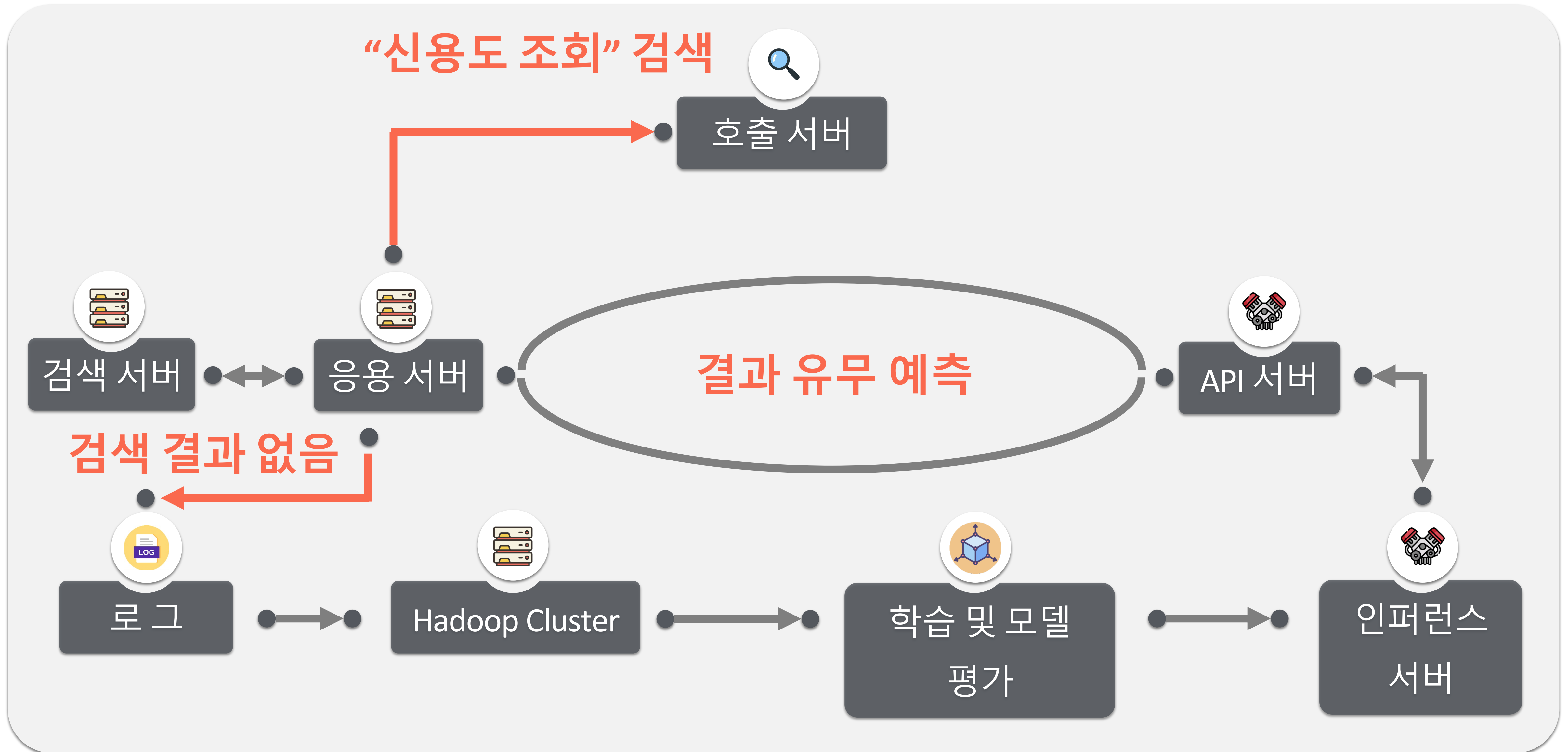
API 서버

- URL을 파싱하여 vector로 변환
- 인퍼런스 서버를 호출하여 예측 결과를 얻음

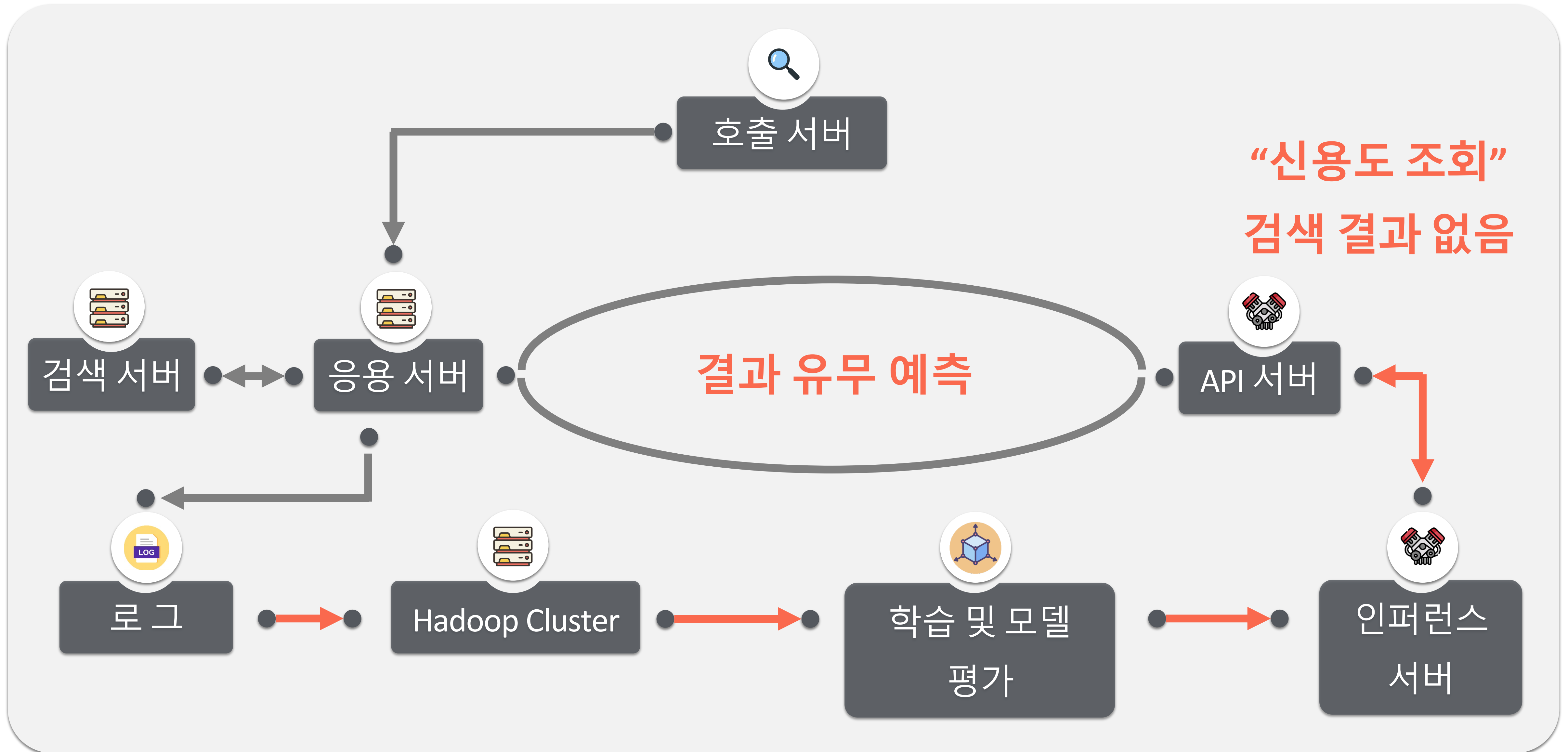
2.3 Feedback Loop



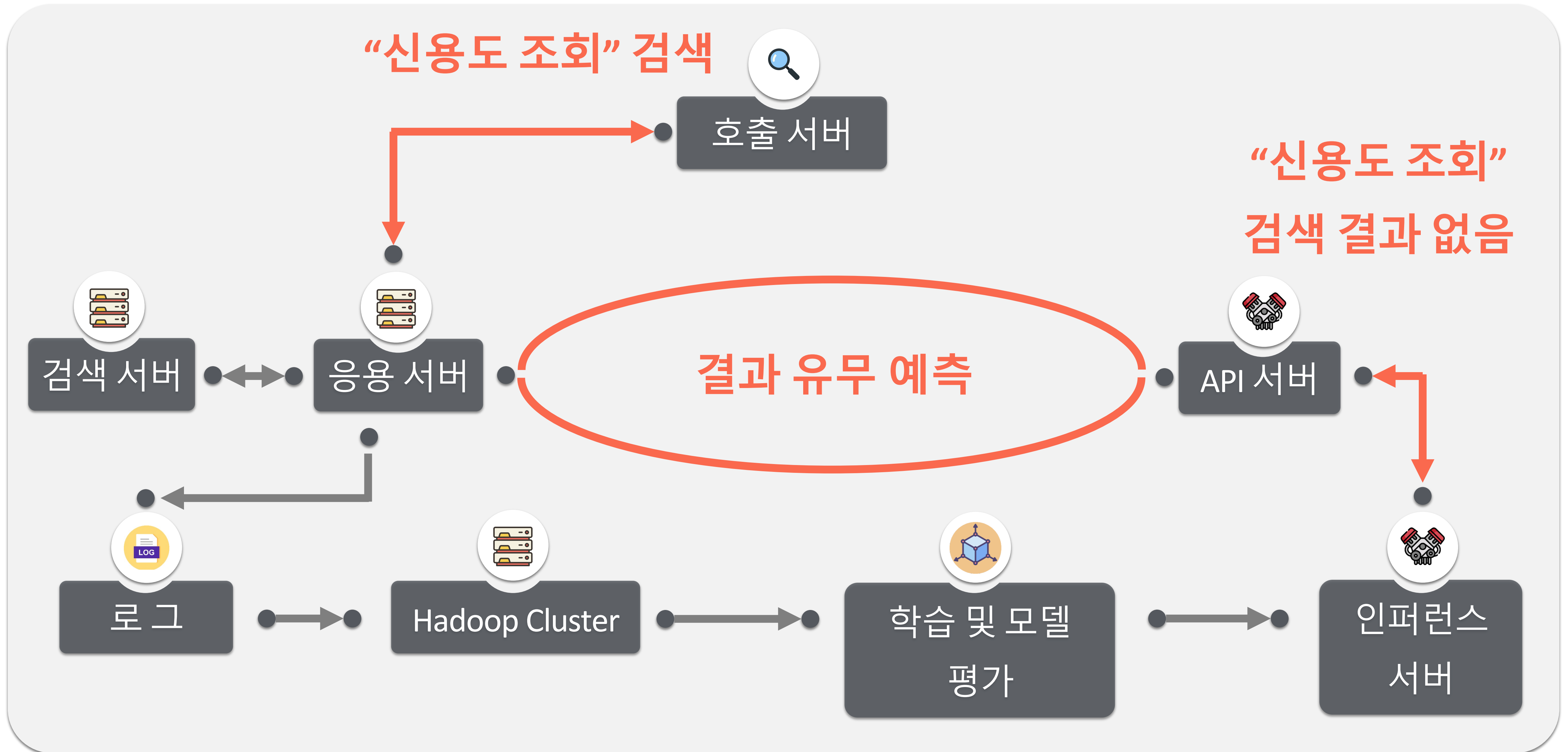
2.3 Feedback Loop



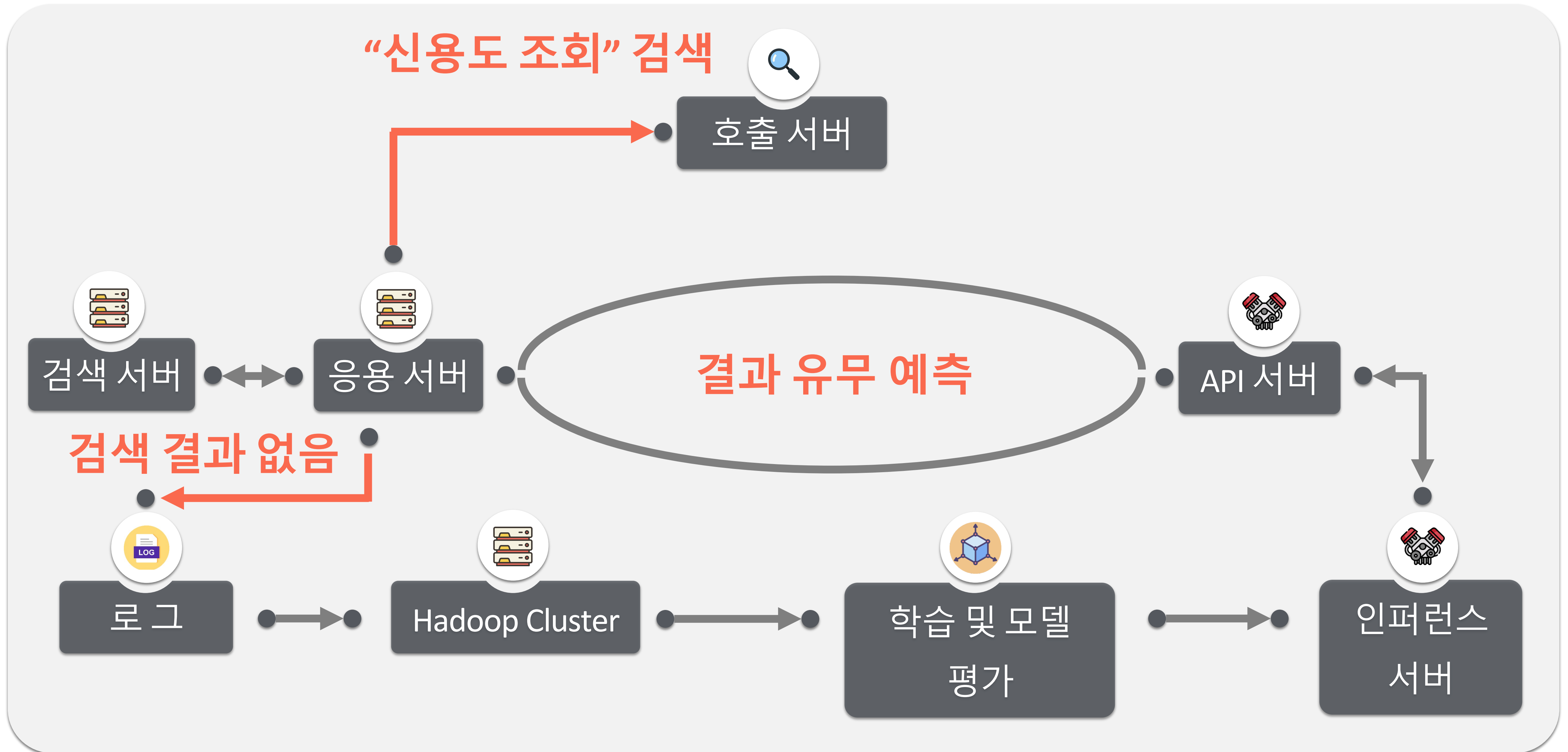
2.3 Feedback Loop



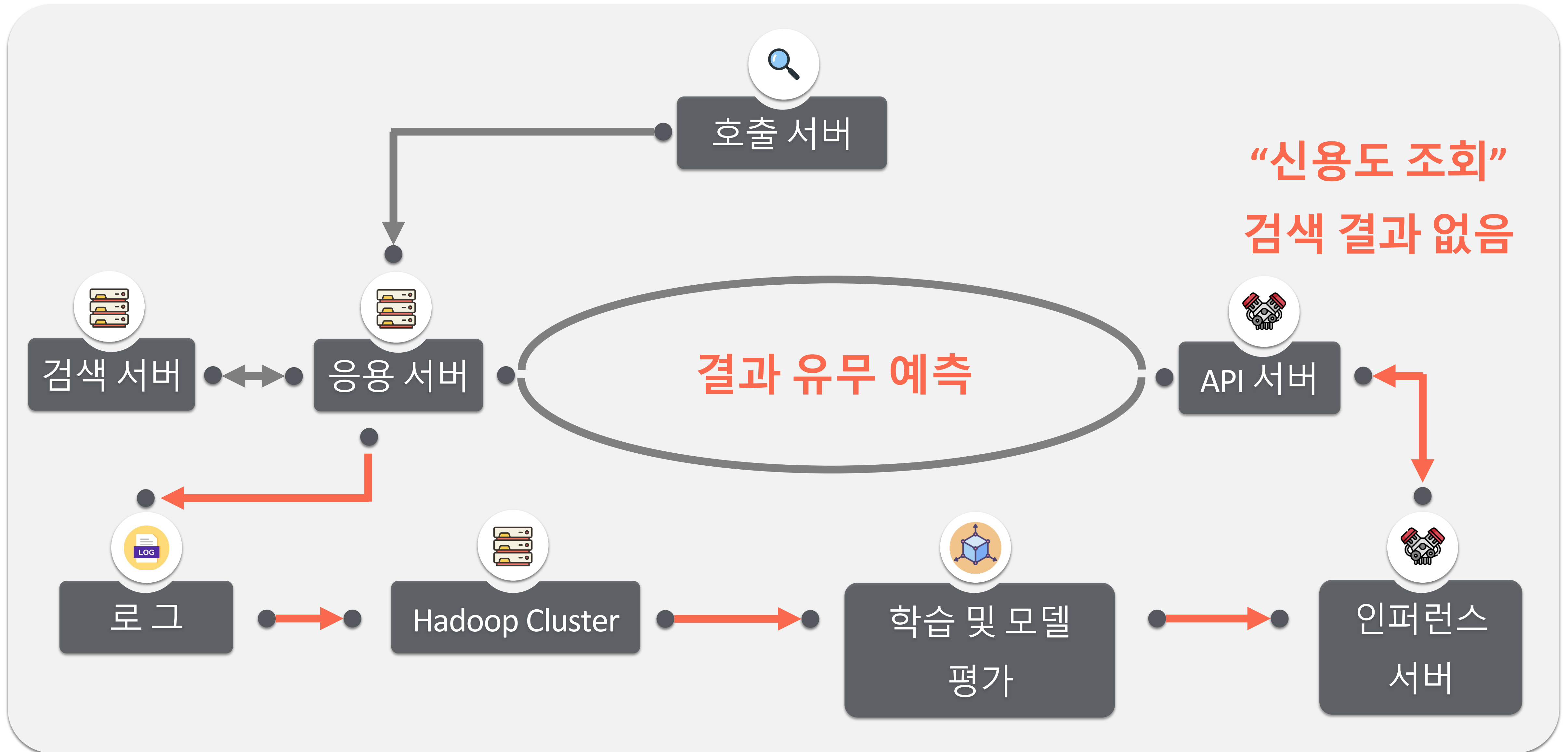
2.3 Feedback Loop



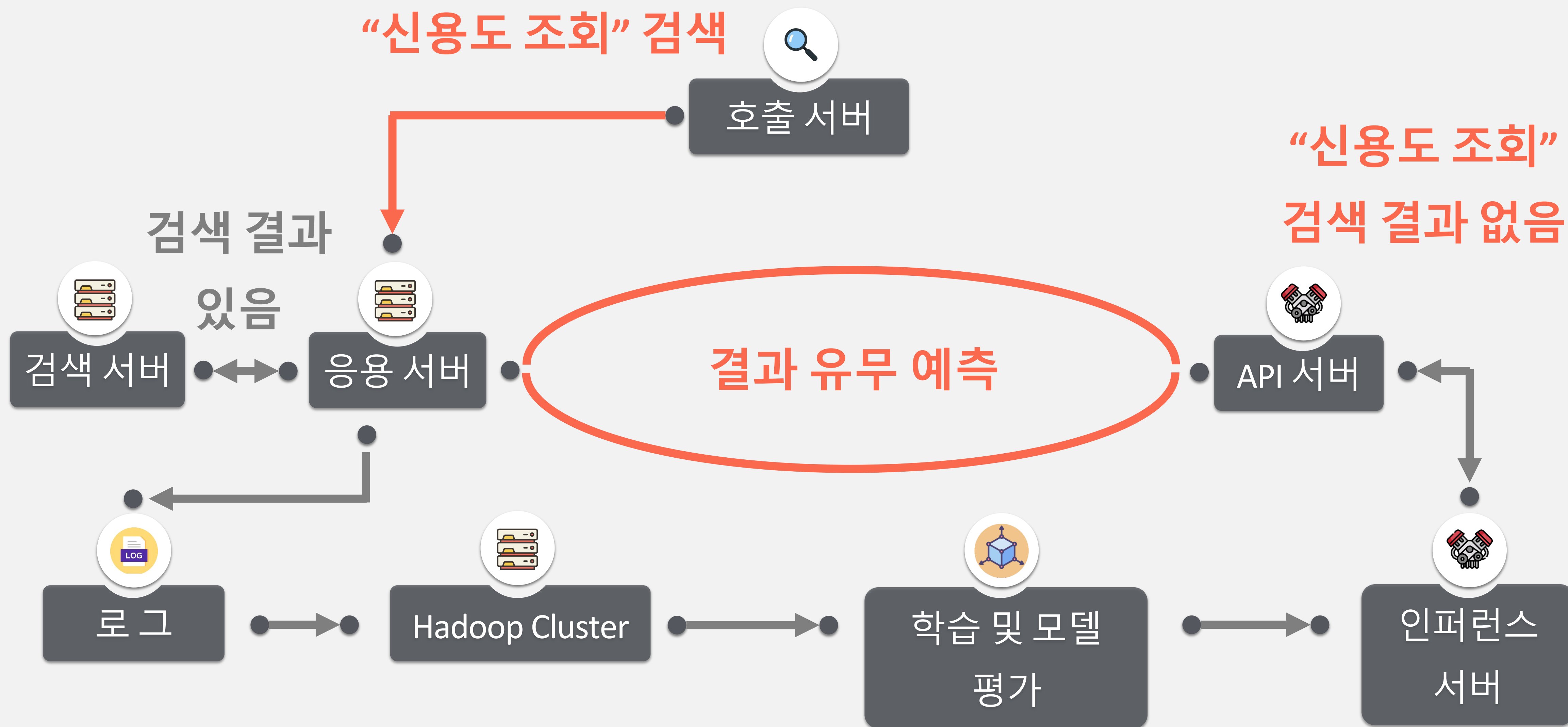
2.3 Feedback Loop



2.3 Feedback Loop



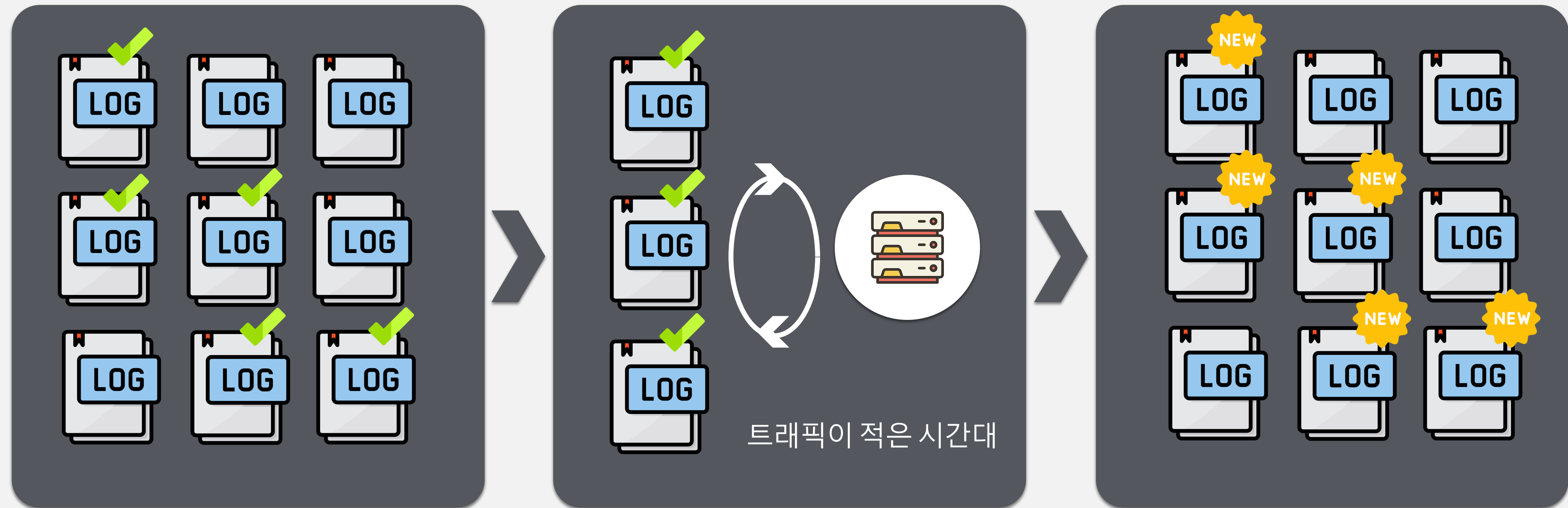
2.3 Feedback Loop



2.3 Feedback Loop



API 적용된 로그 처리



(사용된 로그는 별도 표시하고 자동 갱신합니다)

3. Model & Feature engineering

DF 기반 응답시간 예측

- 선형 회귀
- 다항 회귀
- 로지스틱 회귀(분류)

단어 기반 응답시간 예측

- CNN



어떤 모델로 응답 시간을 예측할 수 있을까???

DF는 응답시간과 연관이 있다.

- 회귀, 분류 두가지 방법으로 진행

학습 데이터

- 범주형 데이터는 one-hot encoding 변환
- 여러 값을 가지는 데이터는 0, 1로 변환

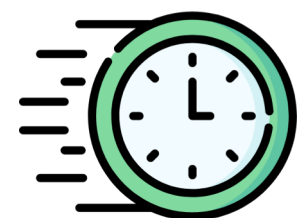
3.1 선형 회귀



Linear Regression



Term DF



응답 시간

3.1 선형 회귀



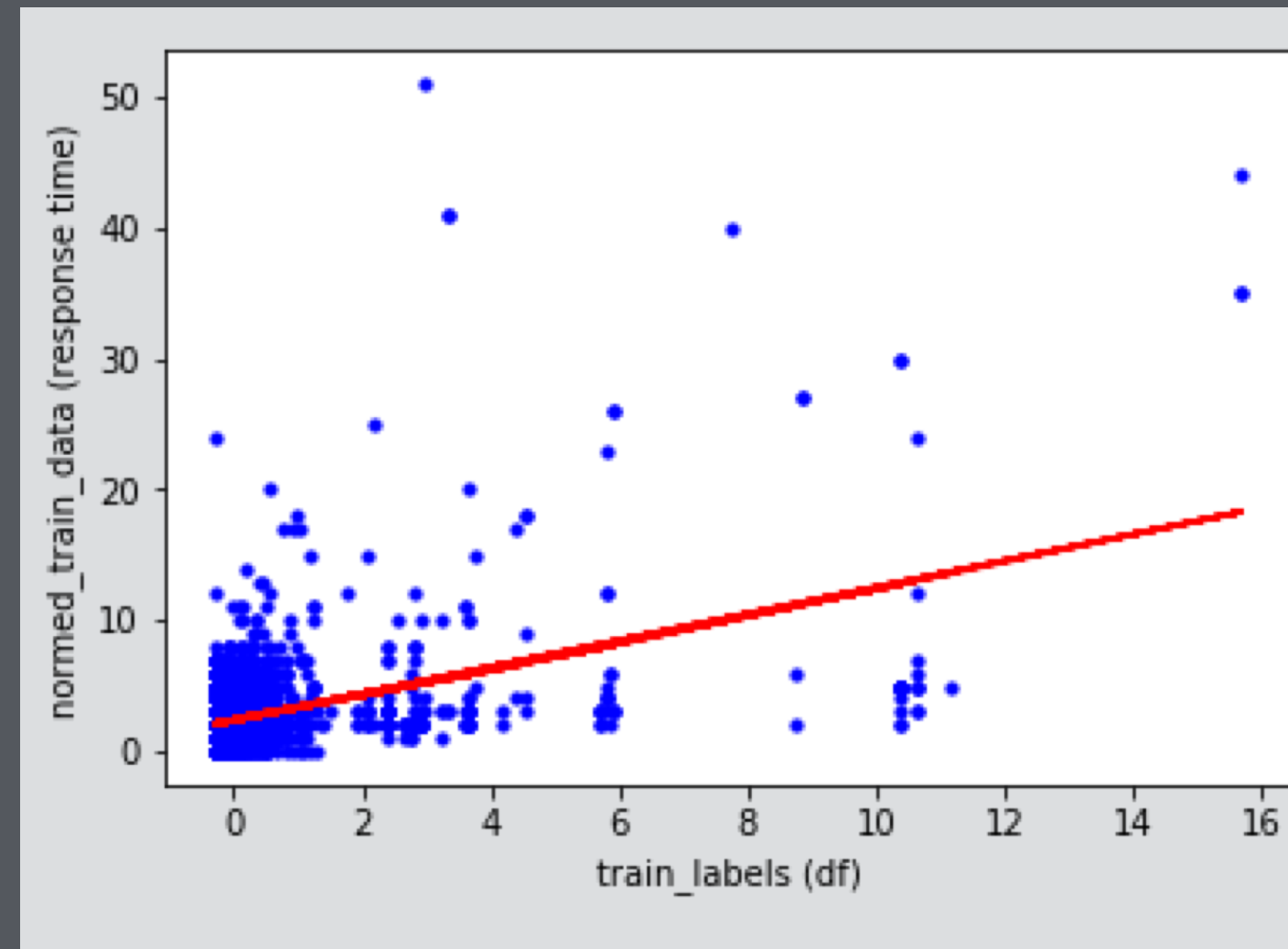
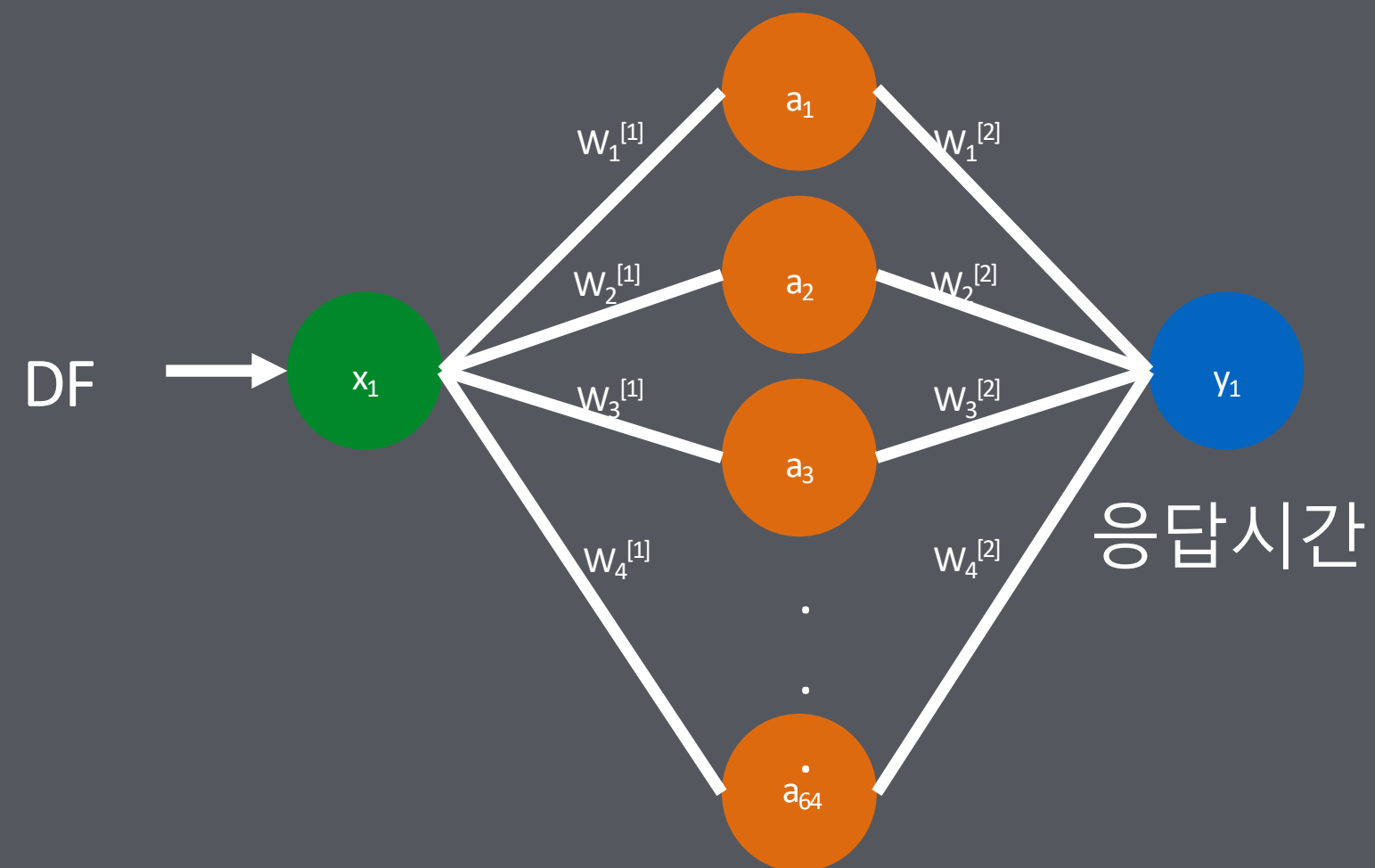
Linear Regression



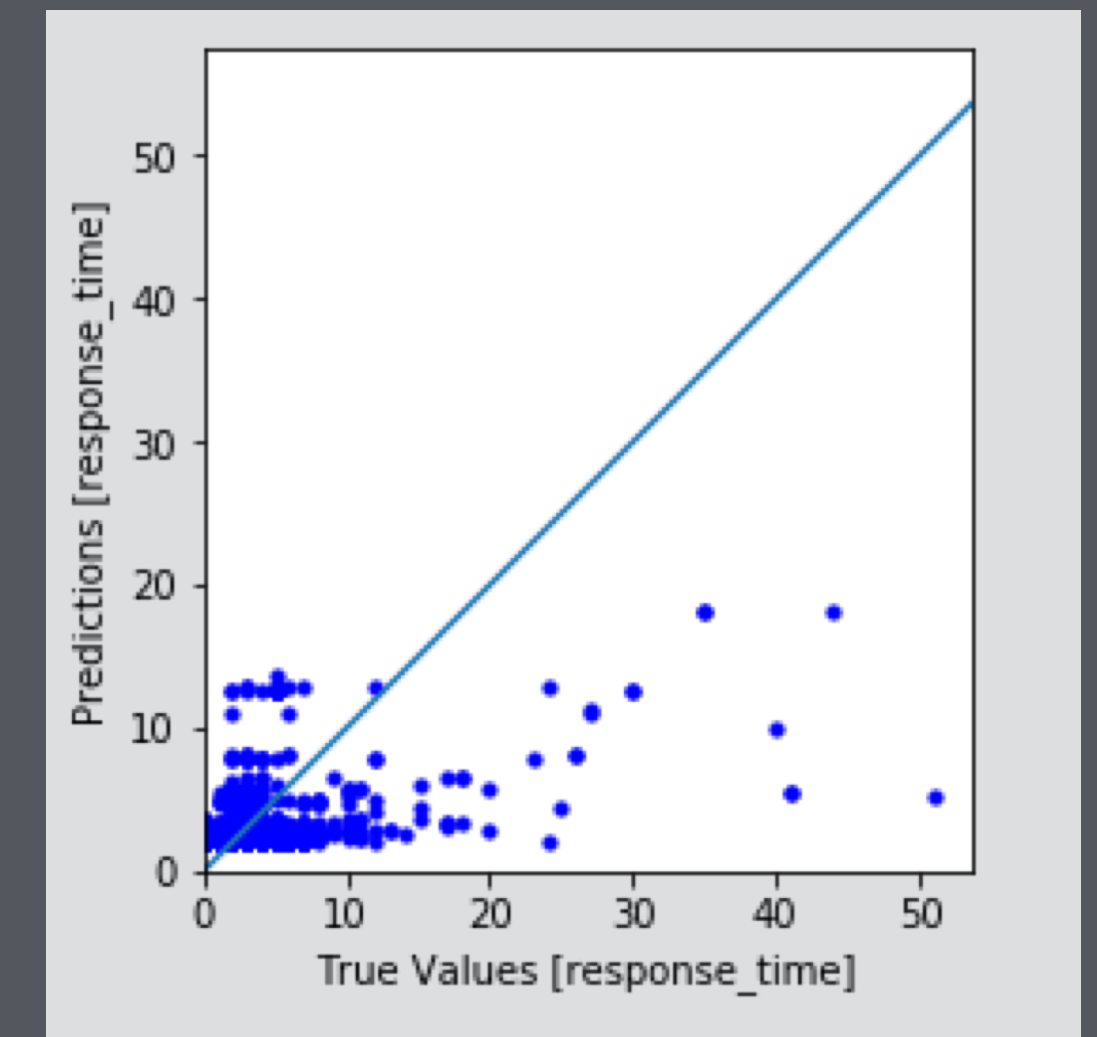
Input Layer

Hidden Layer

Output Layer



<학습 데이터와 예측선>



<실제값-예측값 비교>

DF와 검색 응답 시간은 비례 ??

3.1 선형 회귀(분석)

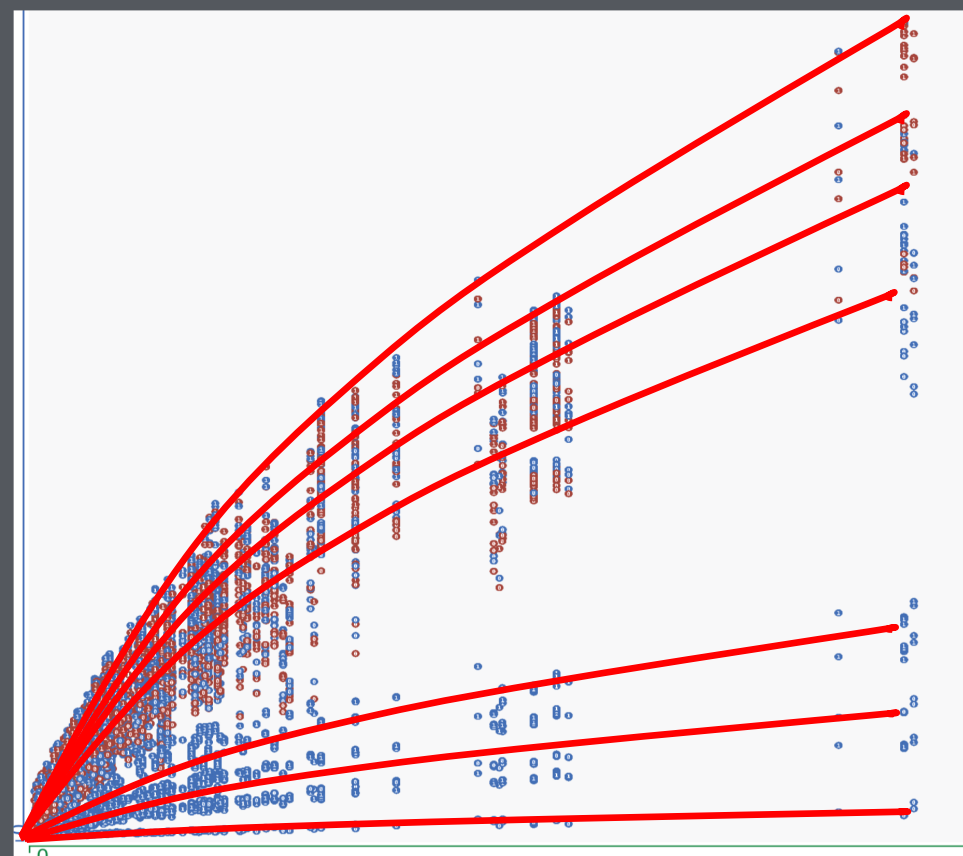


Linear Regression



검색 방식, 검색 옵션 등에 따라 응답시간이 다르다는 것을 확인

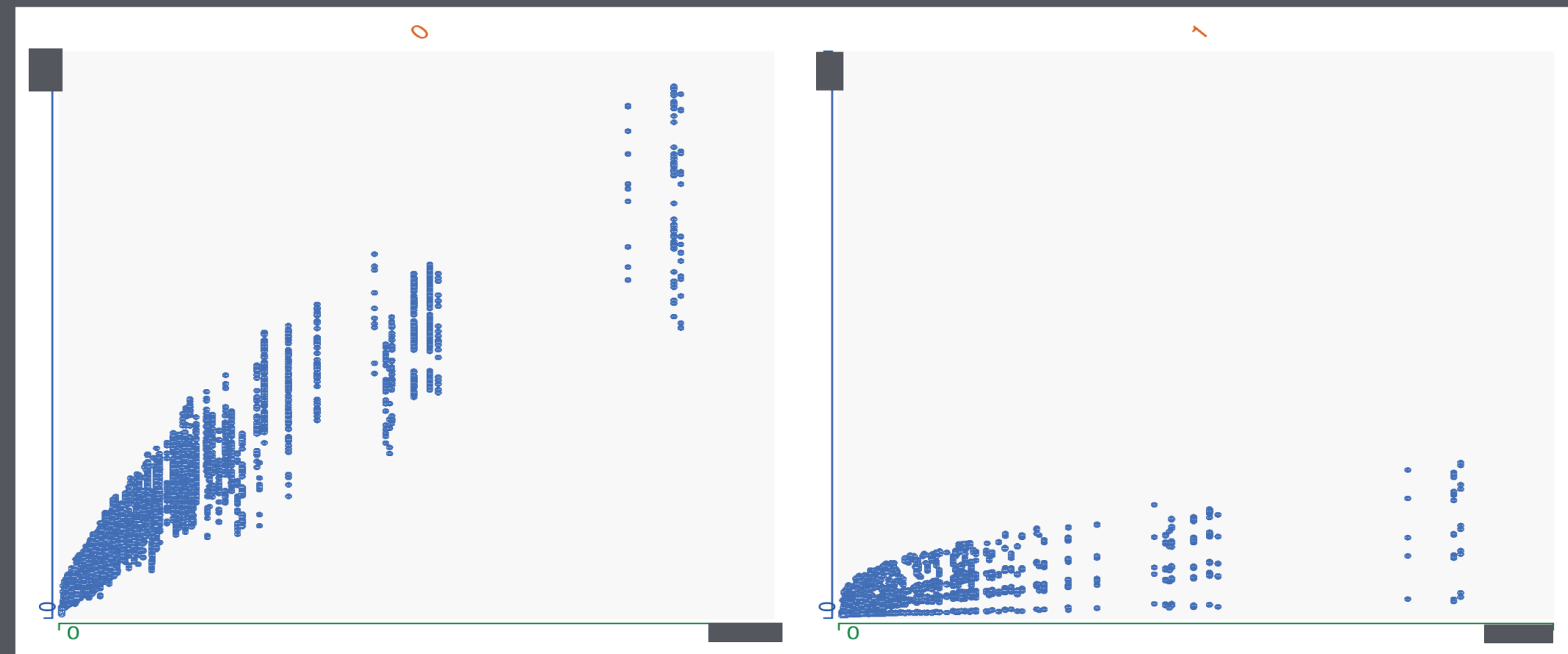
응답시간



DF

< DF-응답시간 분포 >

응답시간



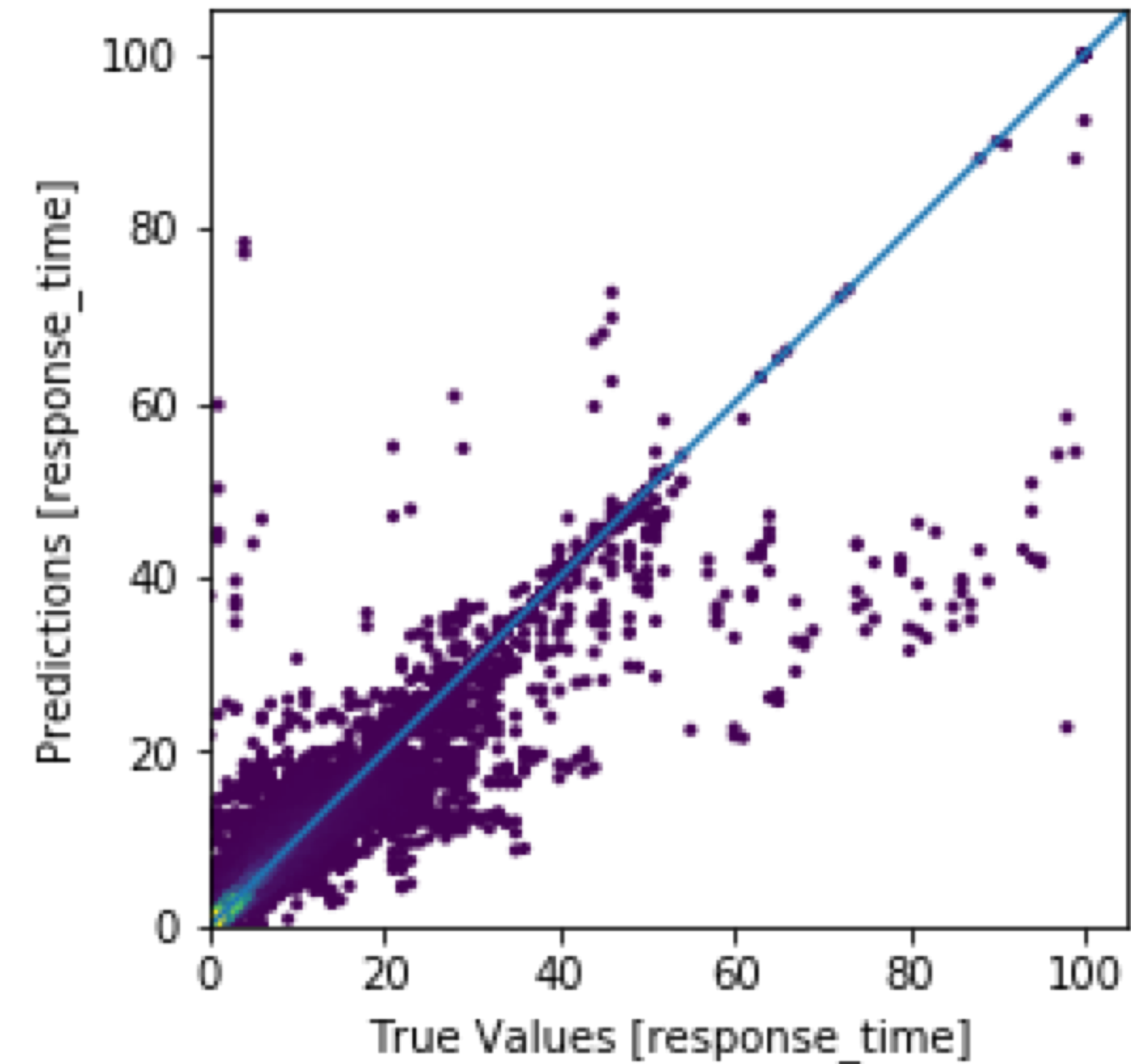
DF

< DF-응답시간 분포(특정 값 1개로 구분) >

3.2 다중 회귀

모든 파라미터 값을 다중 회귀 모델로 학습

- 오차가 크게 발생
- DF와 파라미터를 추가하면서 분석 진행



3.2 다중 회귀



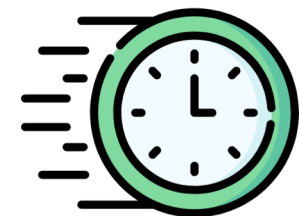
Multi Linear Regression



검색 방식



Term DF



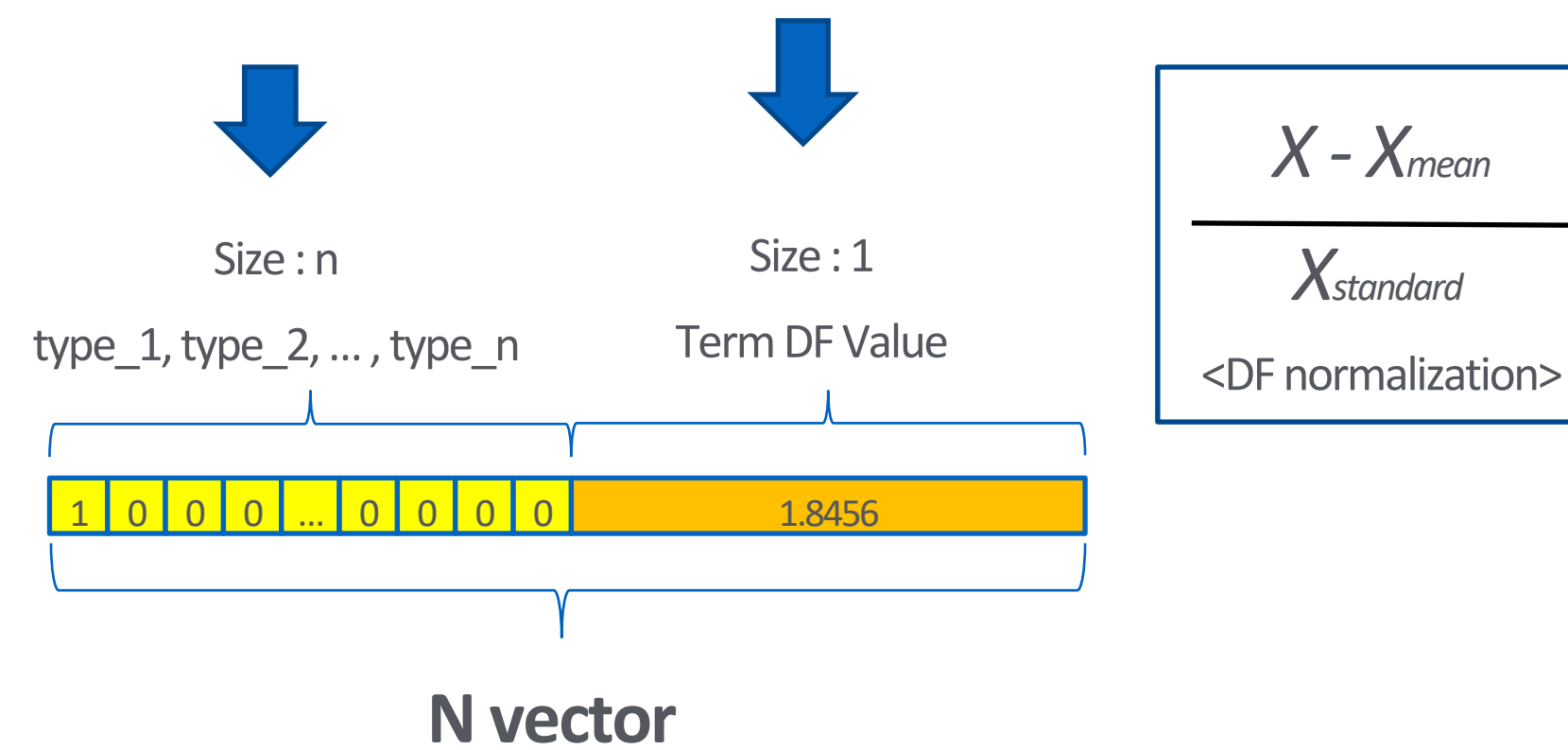
응답 시간

3.2 다중 회귀

학습 데이터

검색 방식

Term Document Frequency



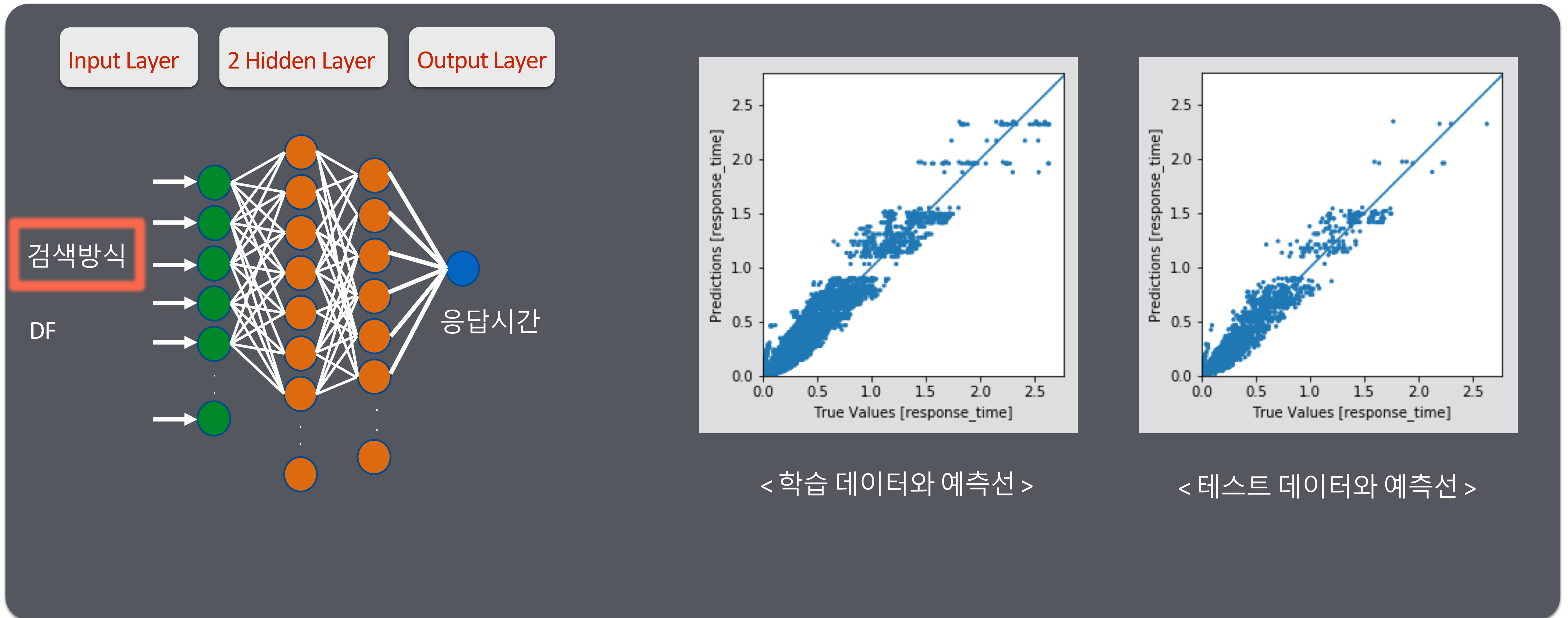
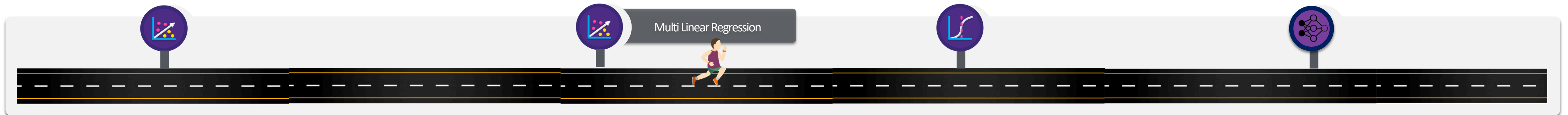
응답 시간



1.18

1 vector

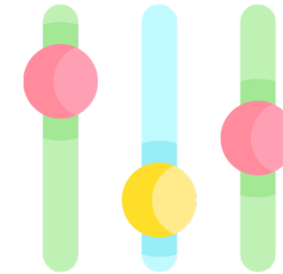
3.2 다중 회귀



3.2 다중 회귀



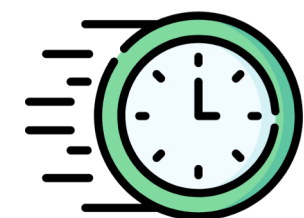
검색 방식



검색 기능



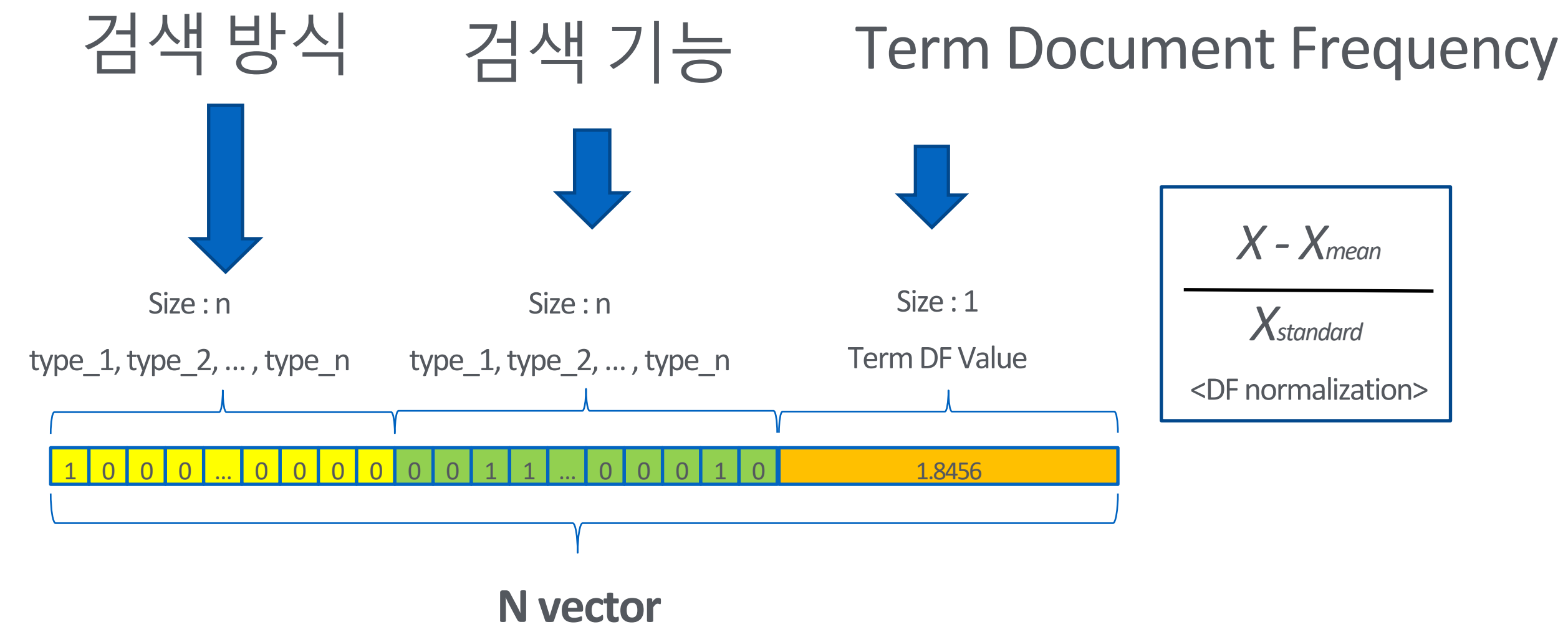
Term DF



응답 시간

3.2 다중 회귀

학습 데이터



응답 시간



1.18

1 vector

3.2 다중 회귀



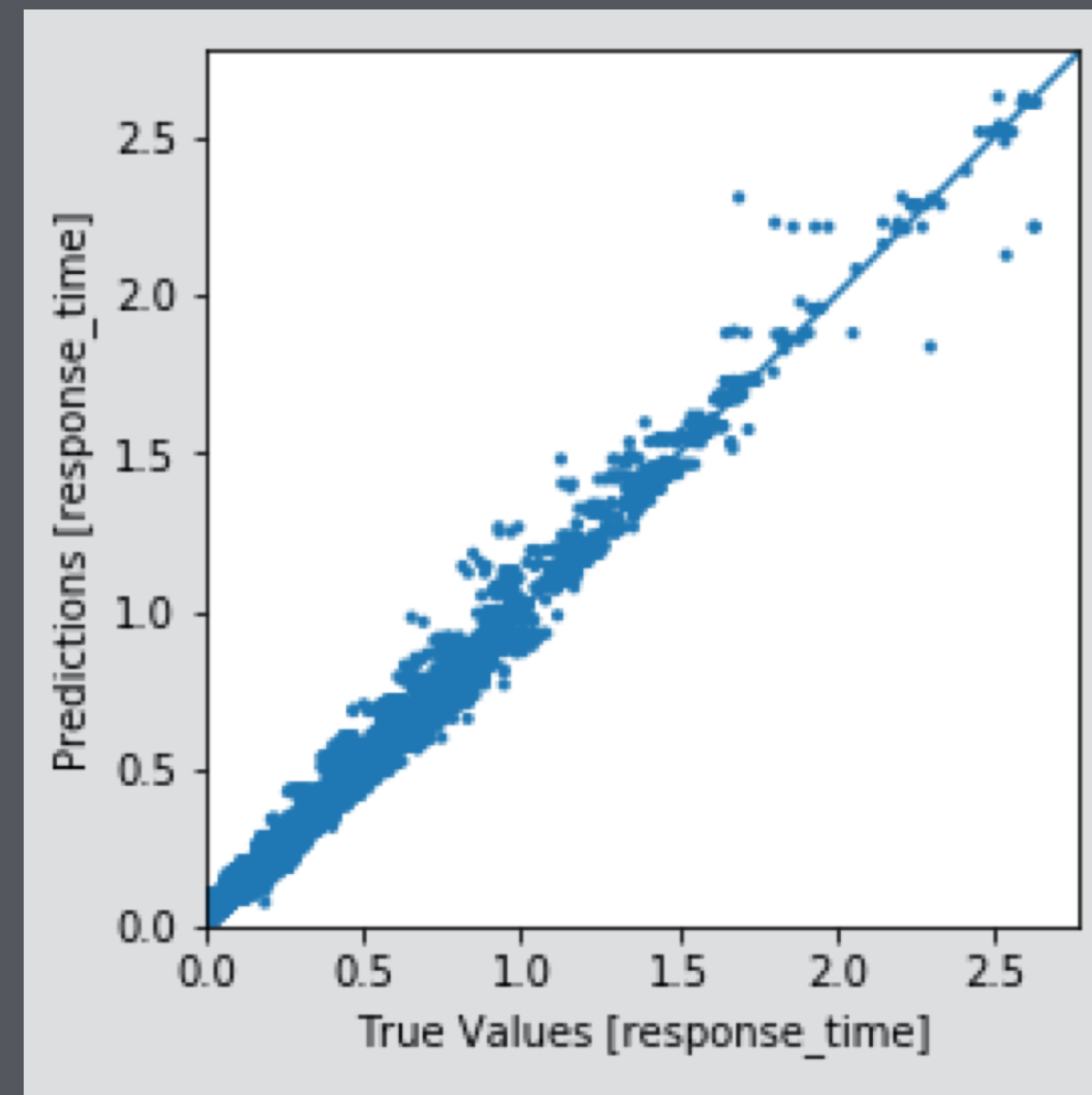
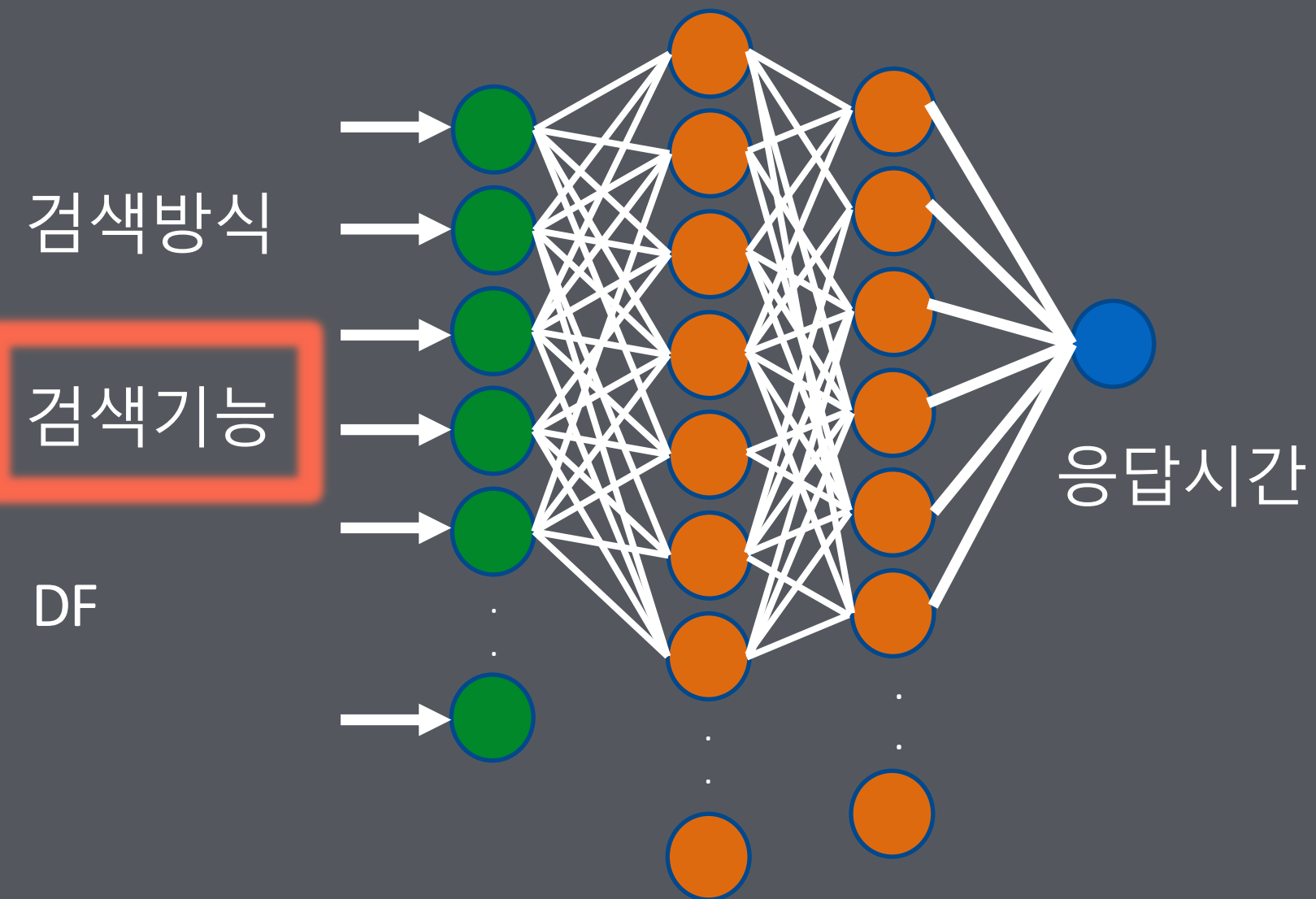
Multi Linear Regression



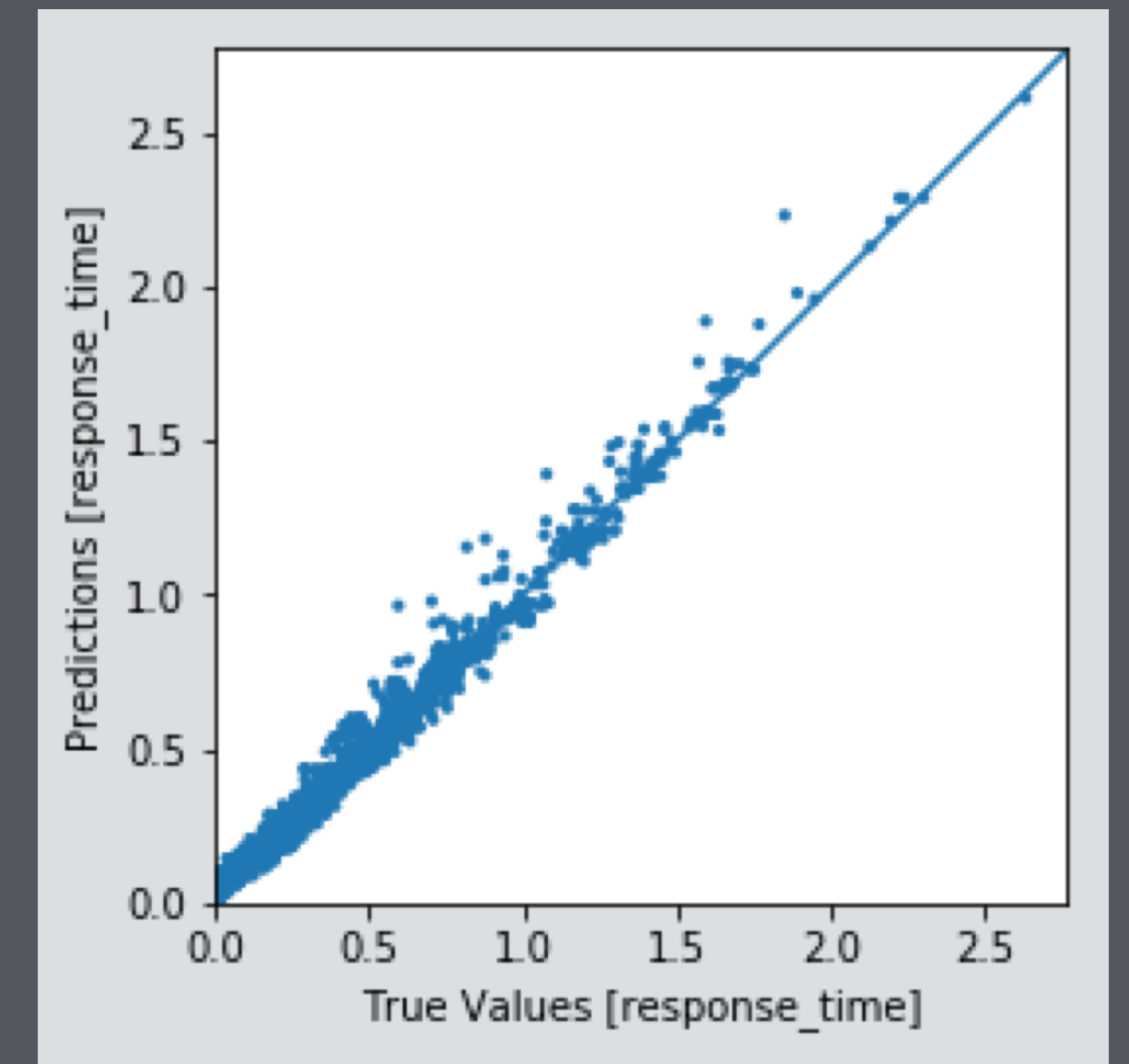
Input Layer

2 Hidden Layer

Output Layer

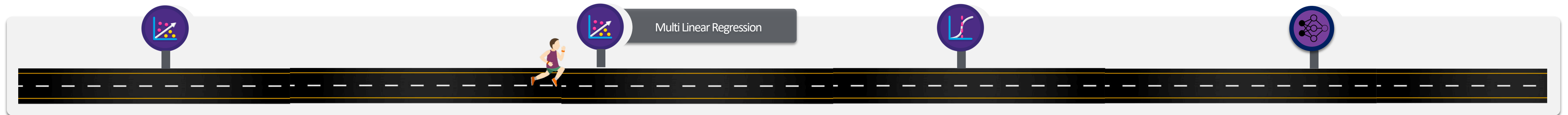


< 학습 데이터와 예측선 >



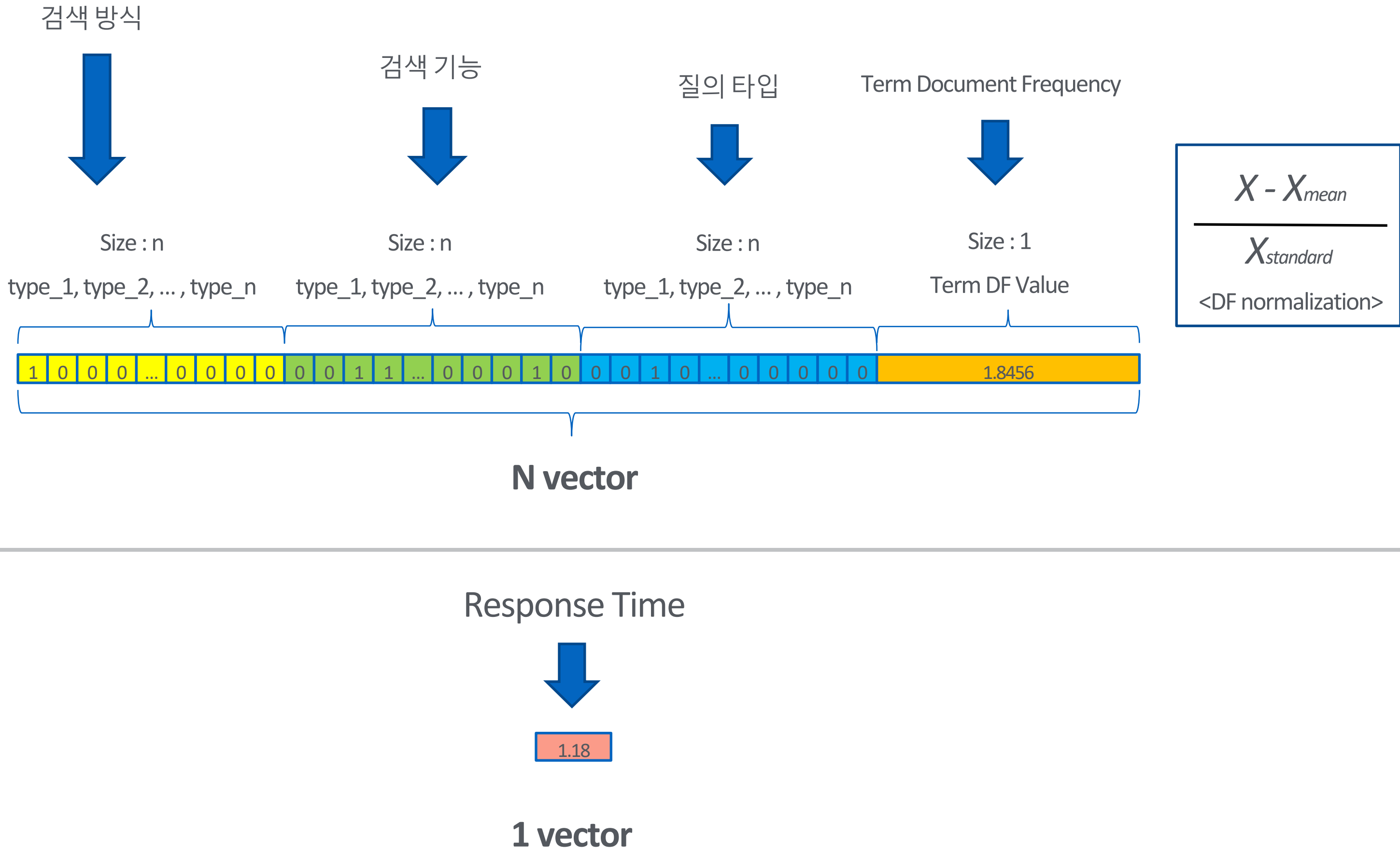
< 테스트 데이터와 예측선 >

3.2 다중 회귀

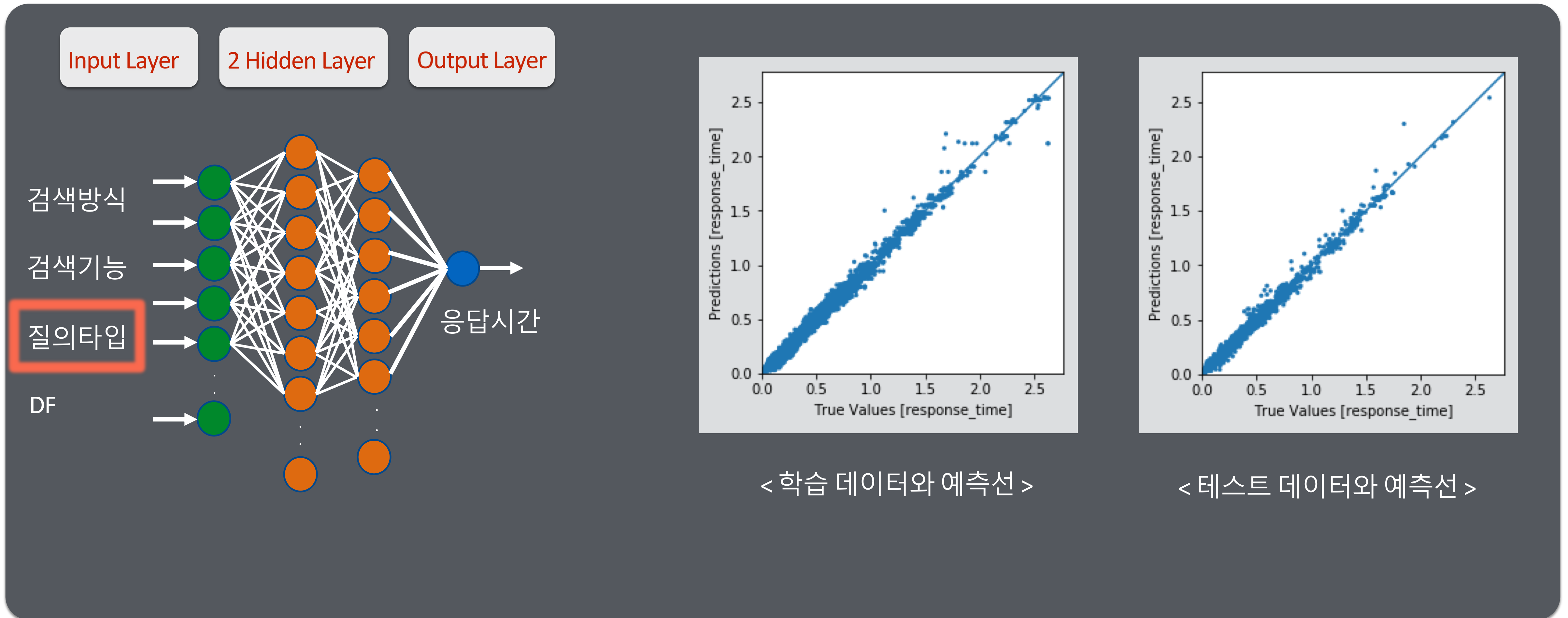
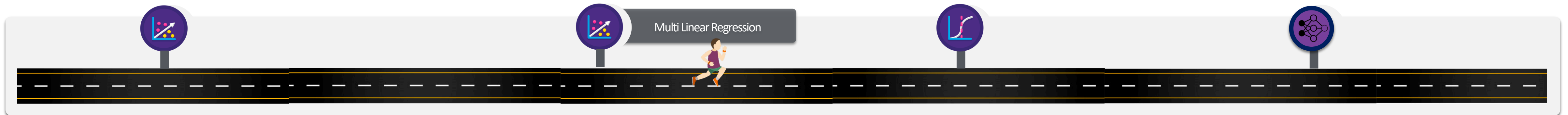


3.2 다중 회귀

입력 벡터

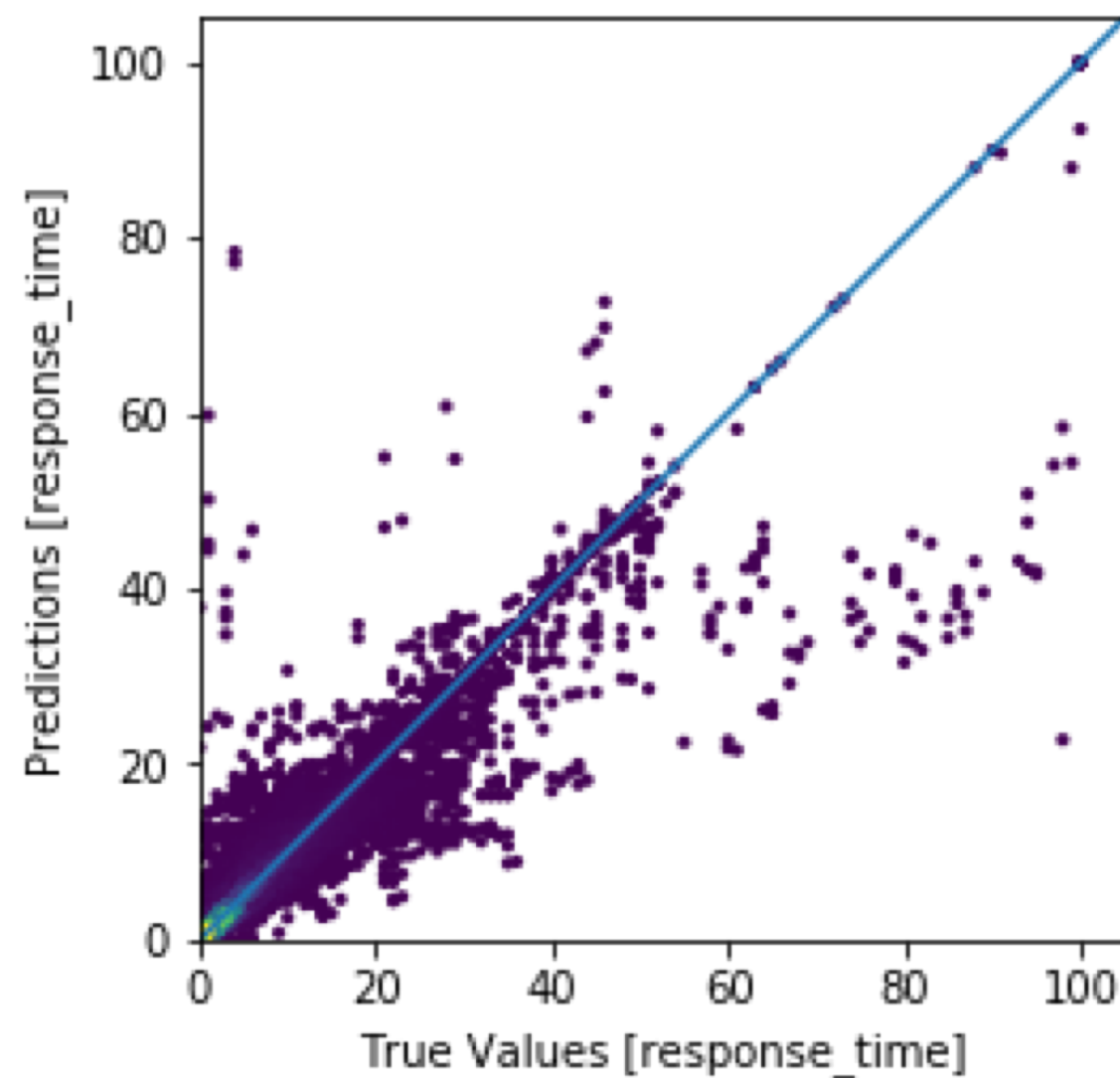


3.2 다중 회귀

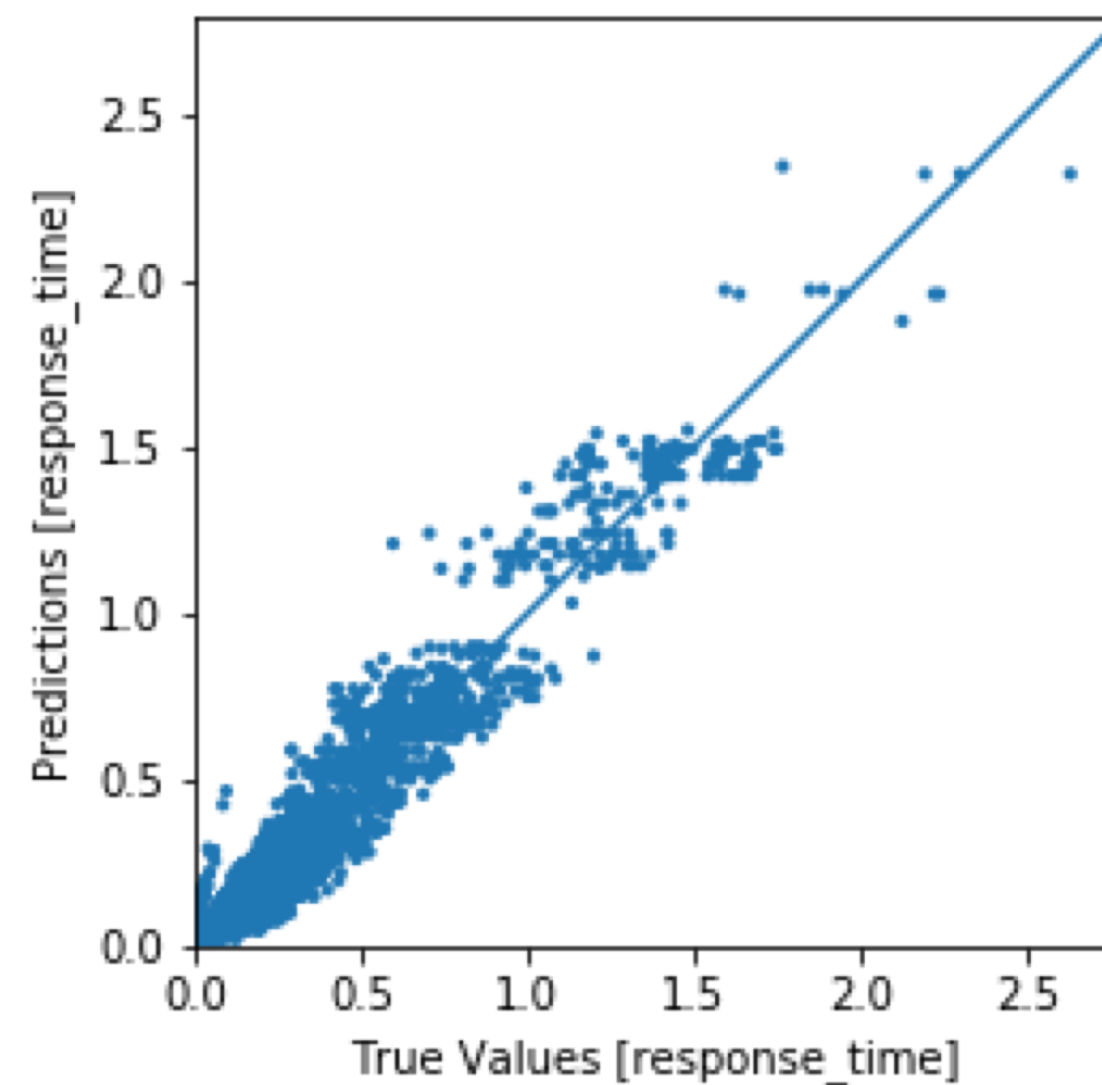


3.2 다중 회귀

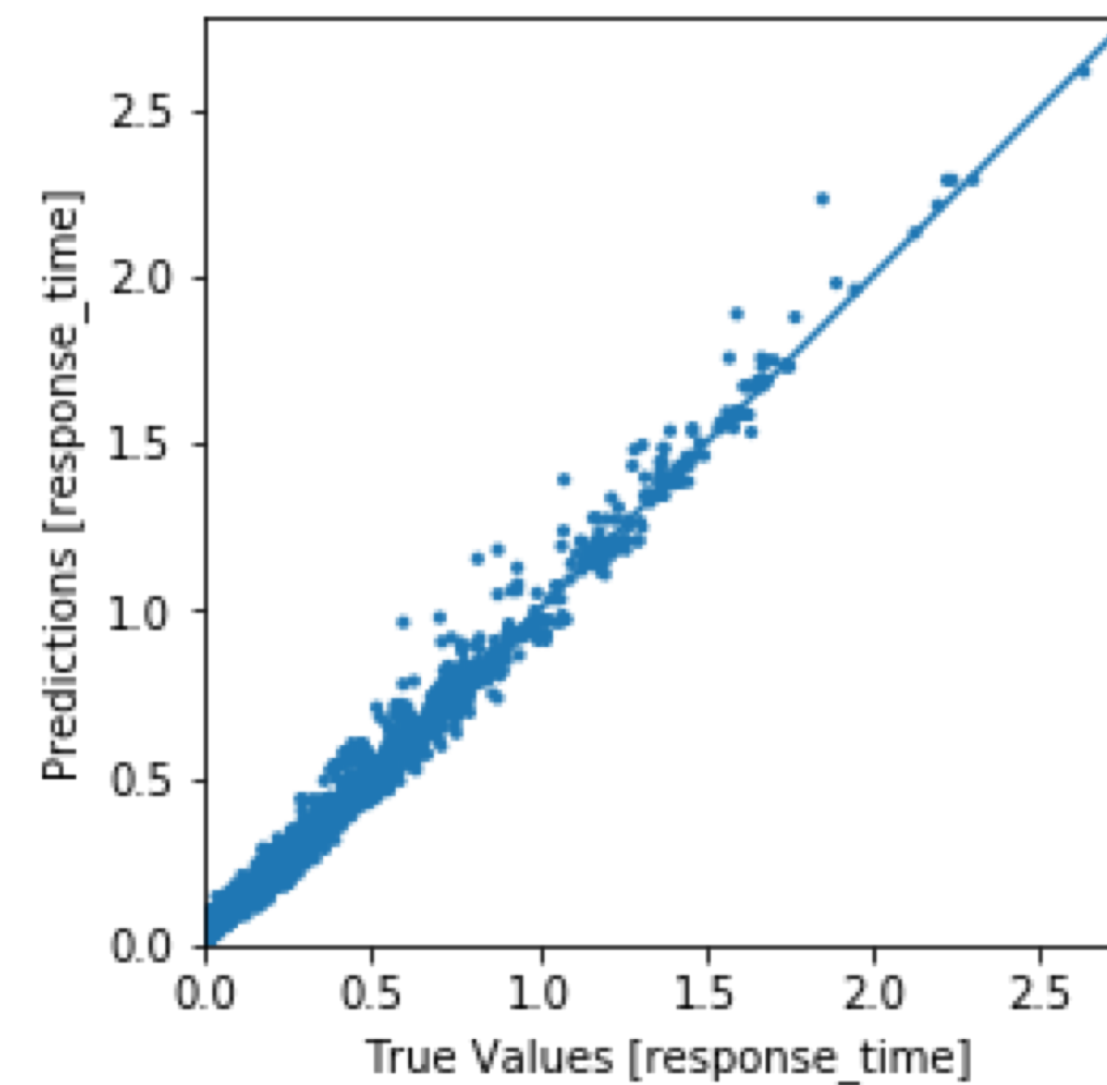
데이터 분석 및 파라미터 선택에 시간을 소요



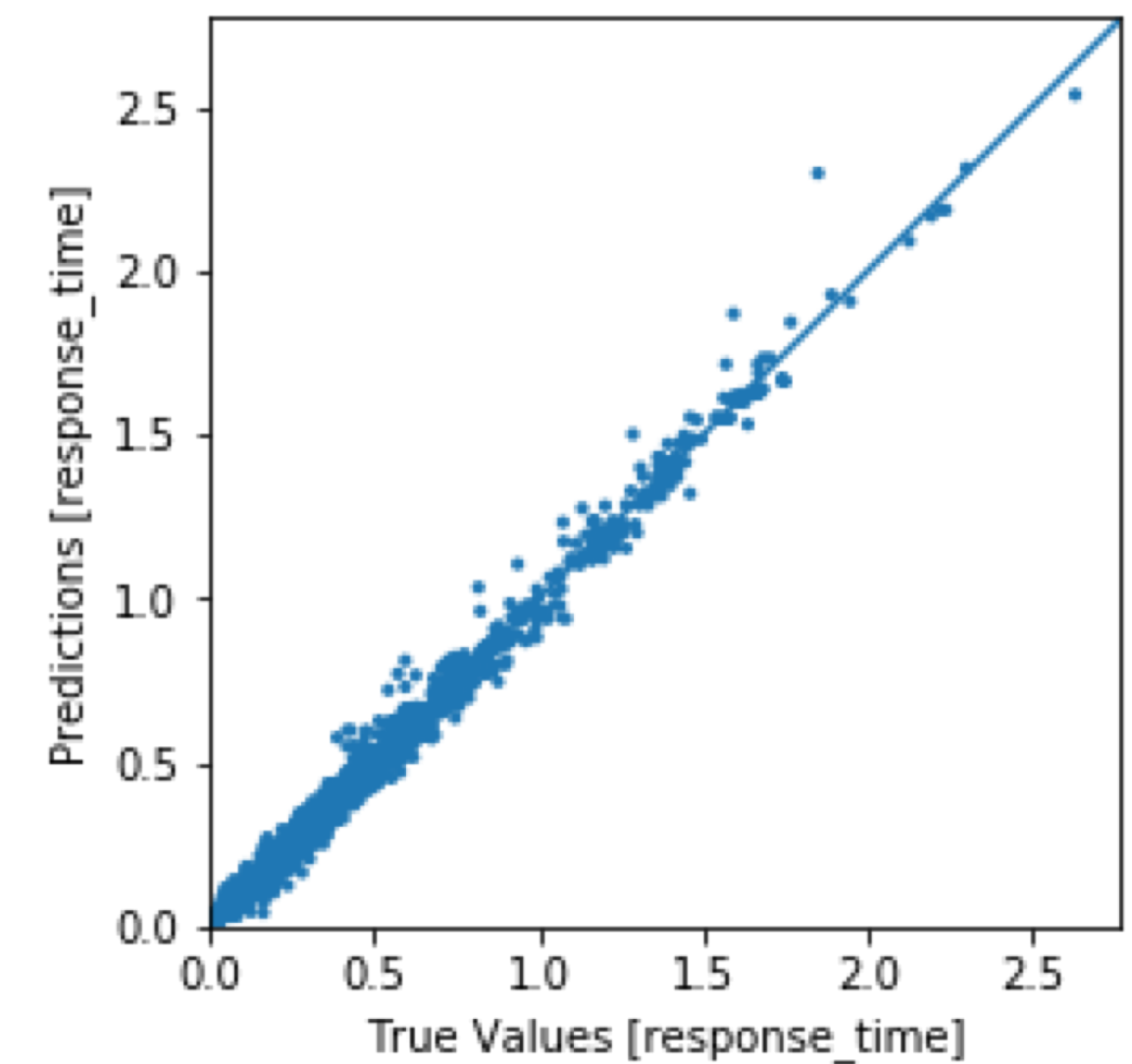
모든 파라미터



DF, 검색 방식



DF, 검색 방식,
검색 기능



DF, 검색 방식,
검색 기능,
질의 타입

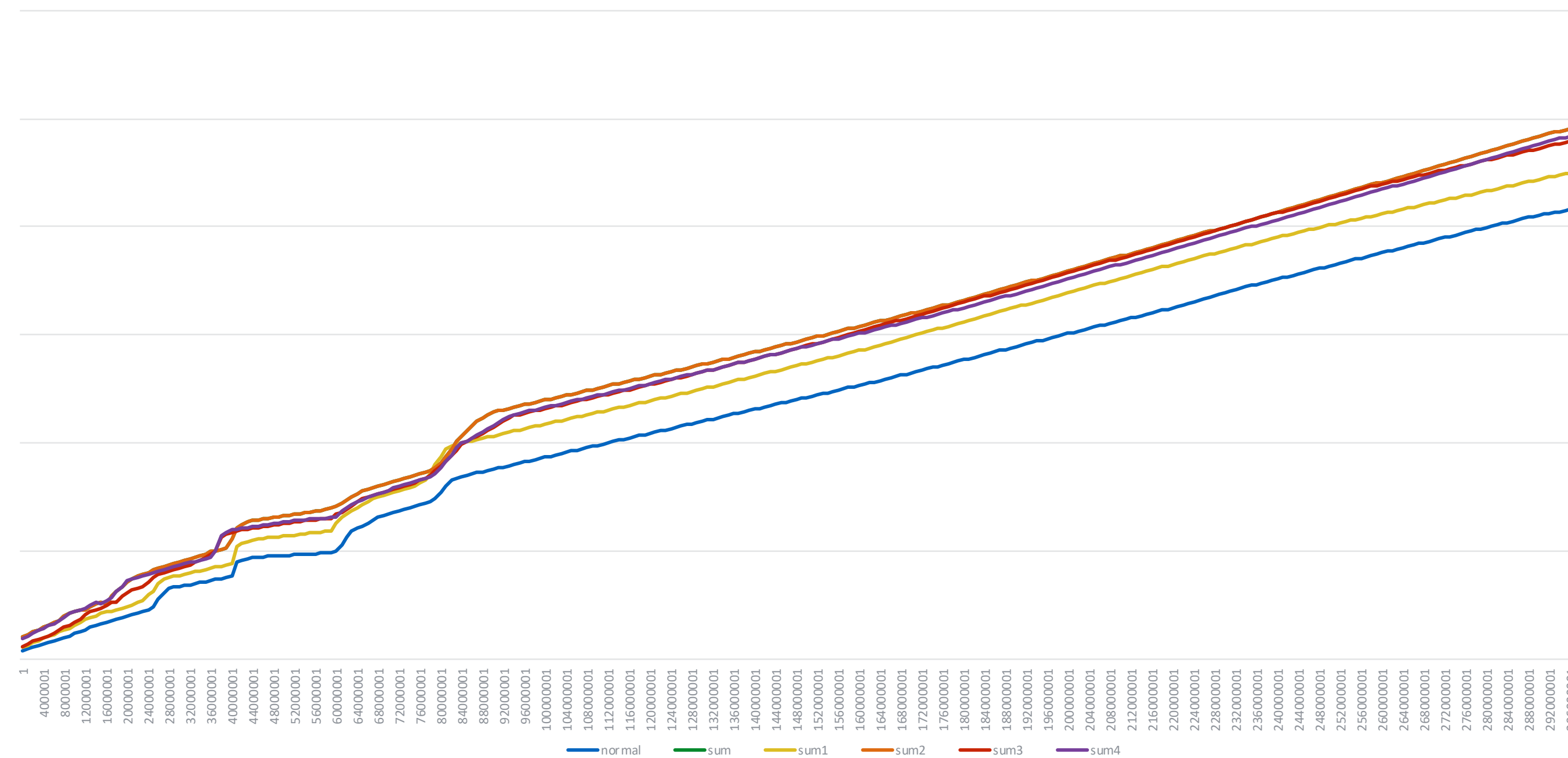
3.2 다중 회귀

인퍼런스 서버를 생성하고 API 서버와 연결

- API 서버에 DF를 일정 간격으로 증가시키면서 예측 응답 시간을 그래프로 표현

응답 시간

응답시간 예측



DF

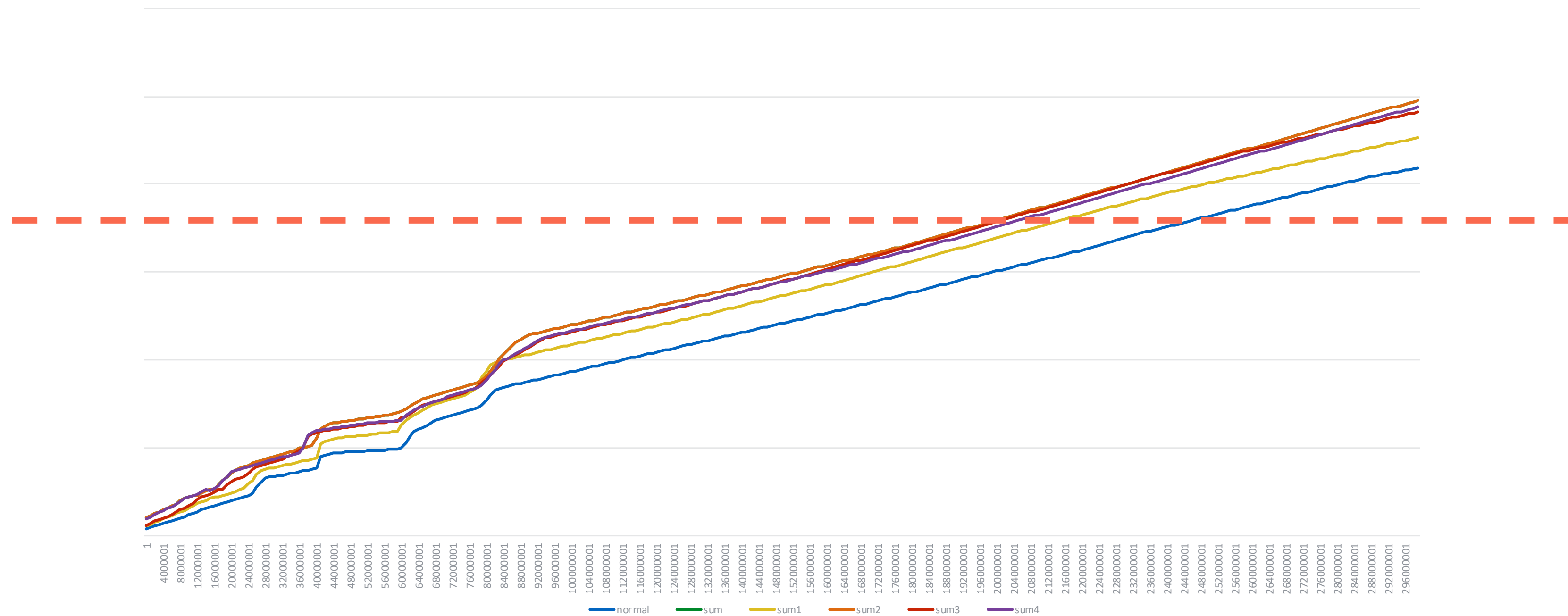
3.2 다중 회귀

인퍼런스 서버를 생성하고 API 서버와 연결

- API 서버에 DF를 일정 간격으로 증가시키면서 예측 응답 시간을 그래프로 표현

응답 시간

응답시간 예측



DF

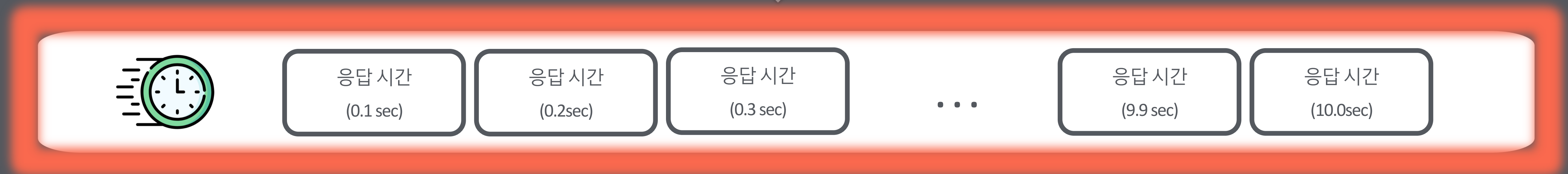
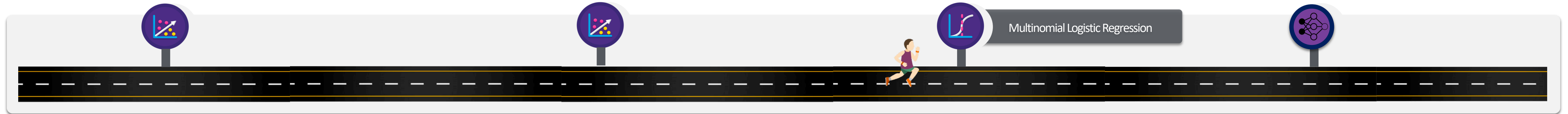
DF 기반 응답시간 예측

- 선형 회귀
- 다항 회귀
- 로지스틱 회귀(분류)



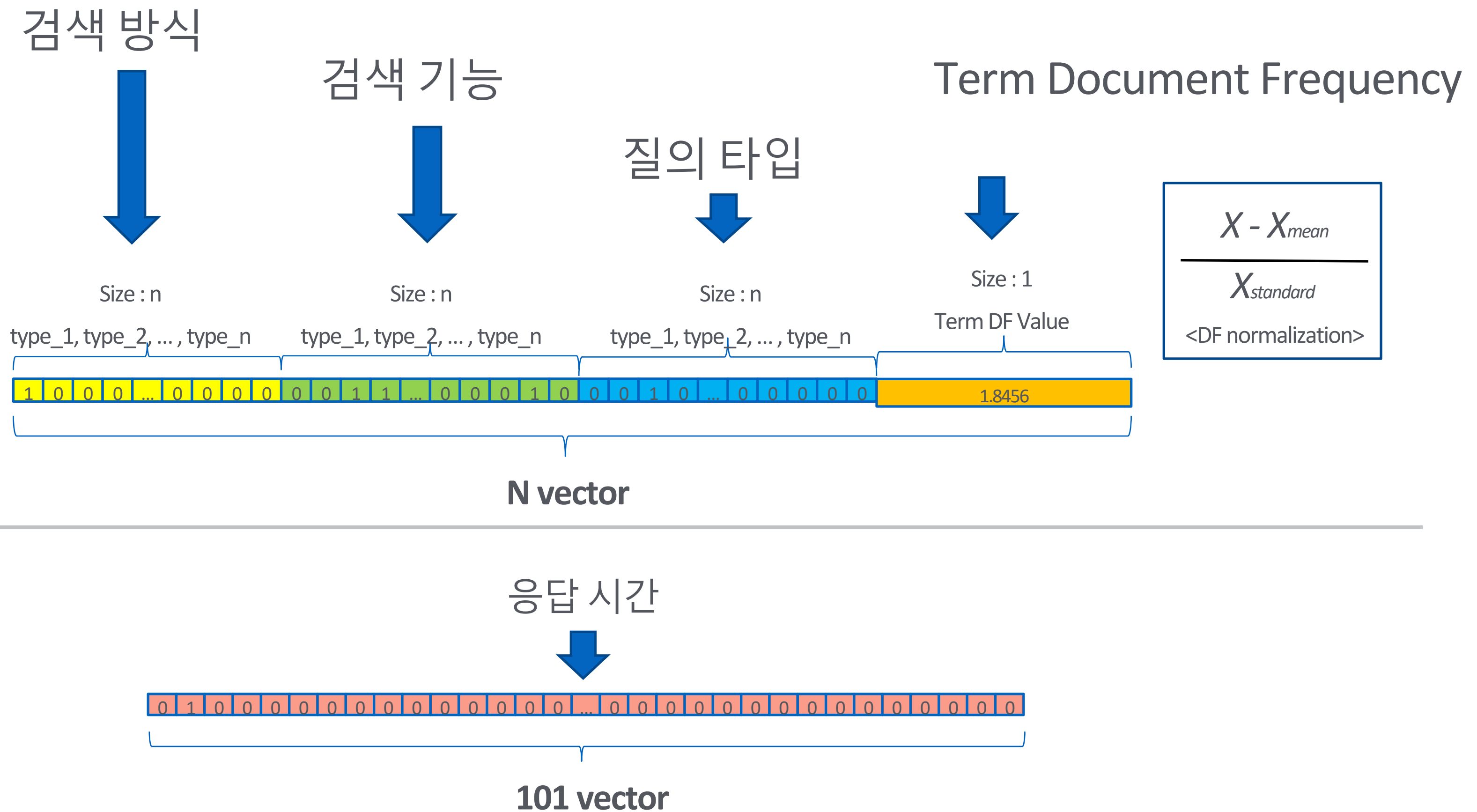
응답시간 예측을 회귀가 아닌 분류 방식으로 접근?

3.3 다중 로지스틱 회귀



3.3 다중 로지스틱 회귀

학습 데이터



3.3 다중 로지스틱 회귀



Multinomial Logistic Regression

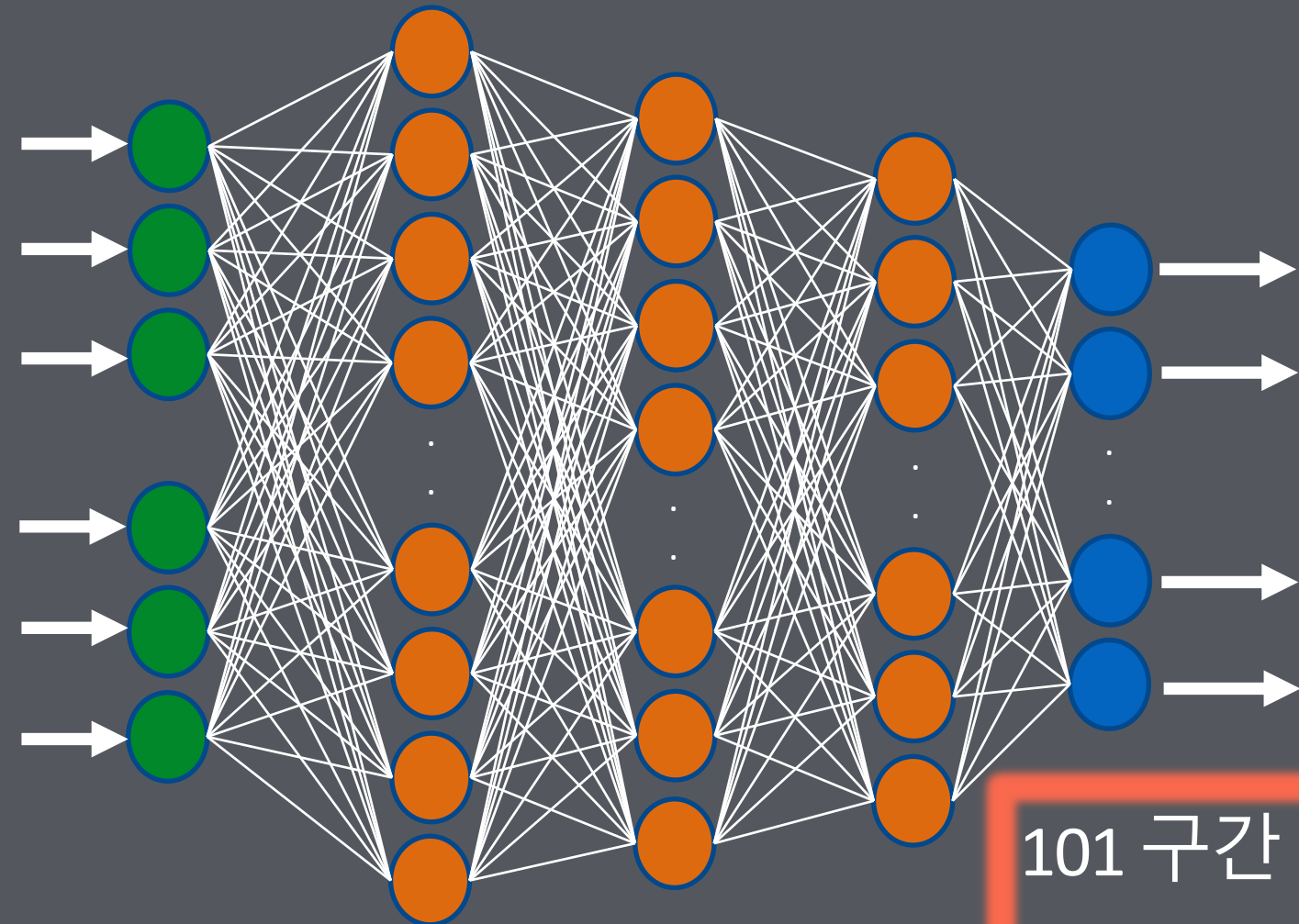


Input Layer

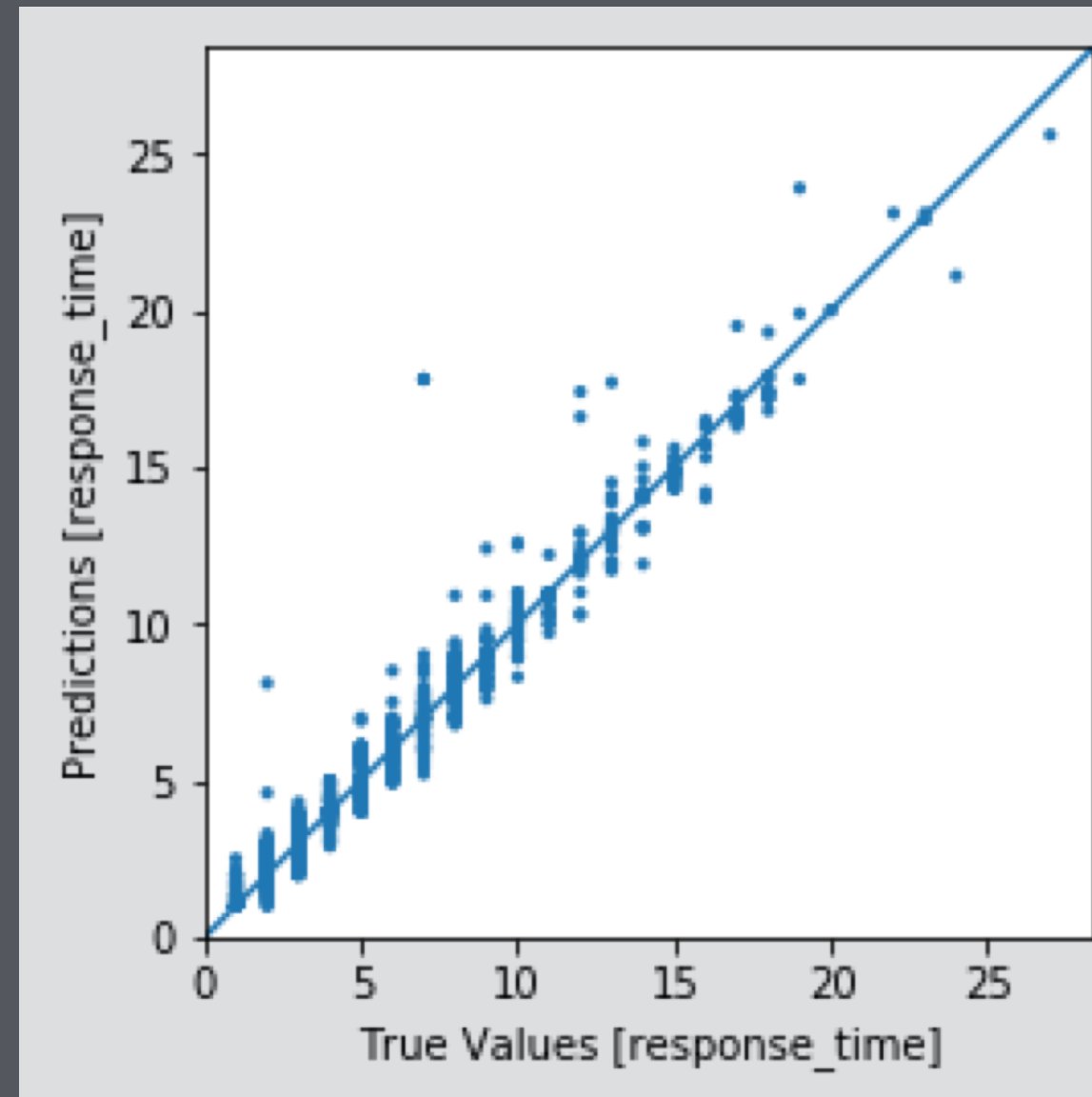
3 Hidden Layer

Output Layer

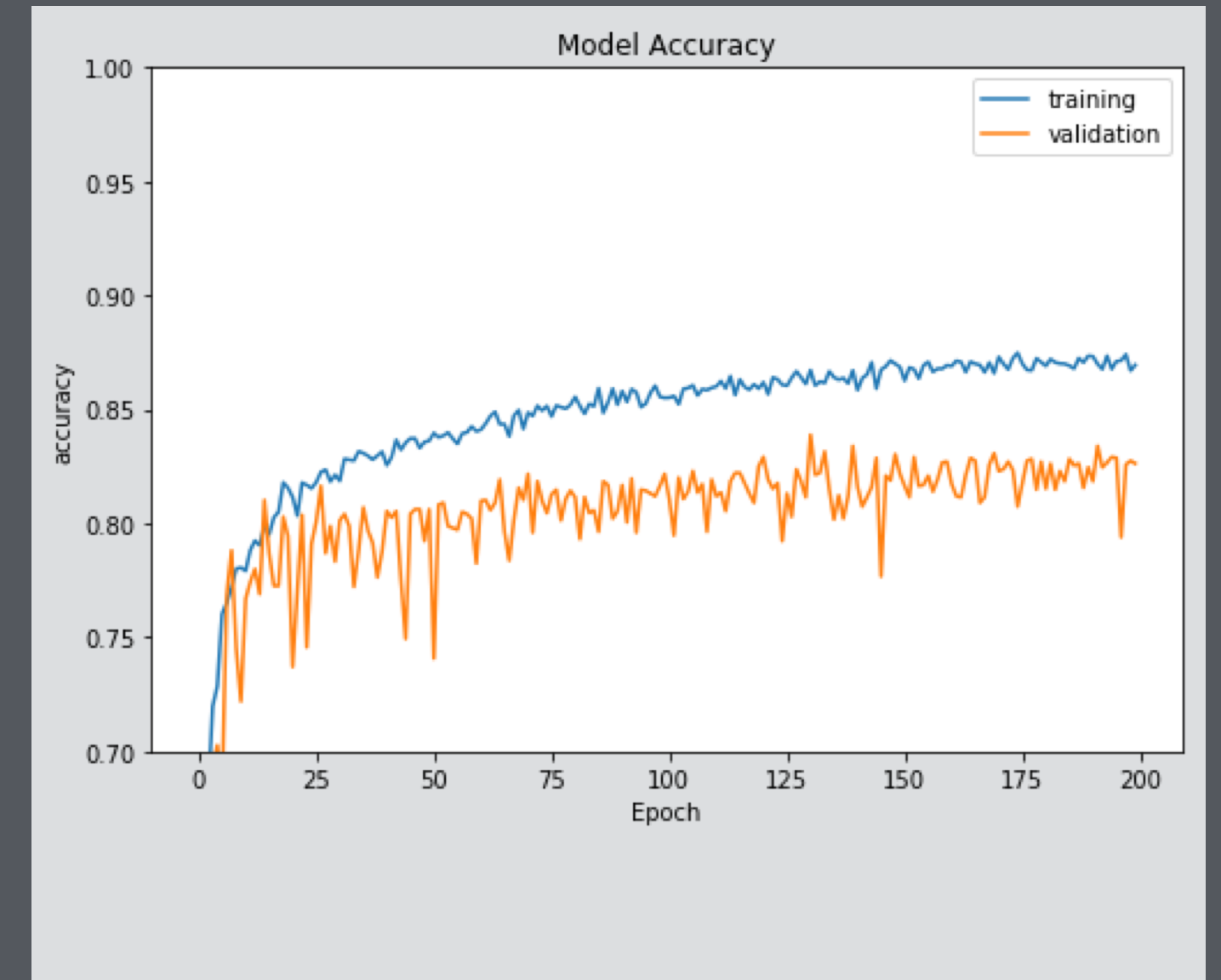
검색방식
검색기능
질의타입
DF



101 구간
응답시간 확률



< 테스트 데이터와 예측선 >



< 정확도 >



- DF를 일정 간격으로 증가하면서 예측 응답 시간을 그래프로 표현

3.3 다중 로지스틱 회귀

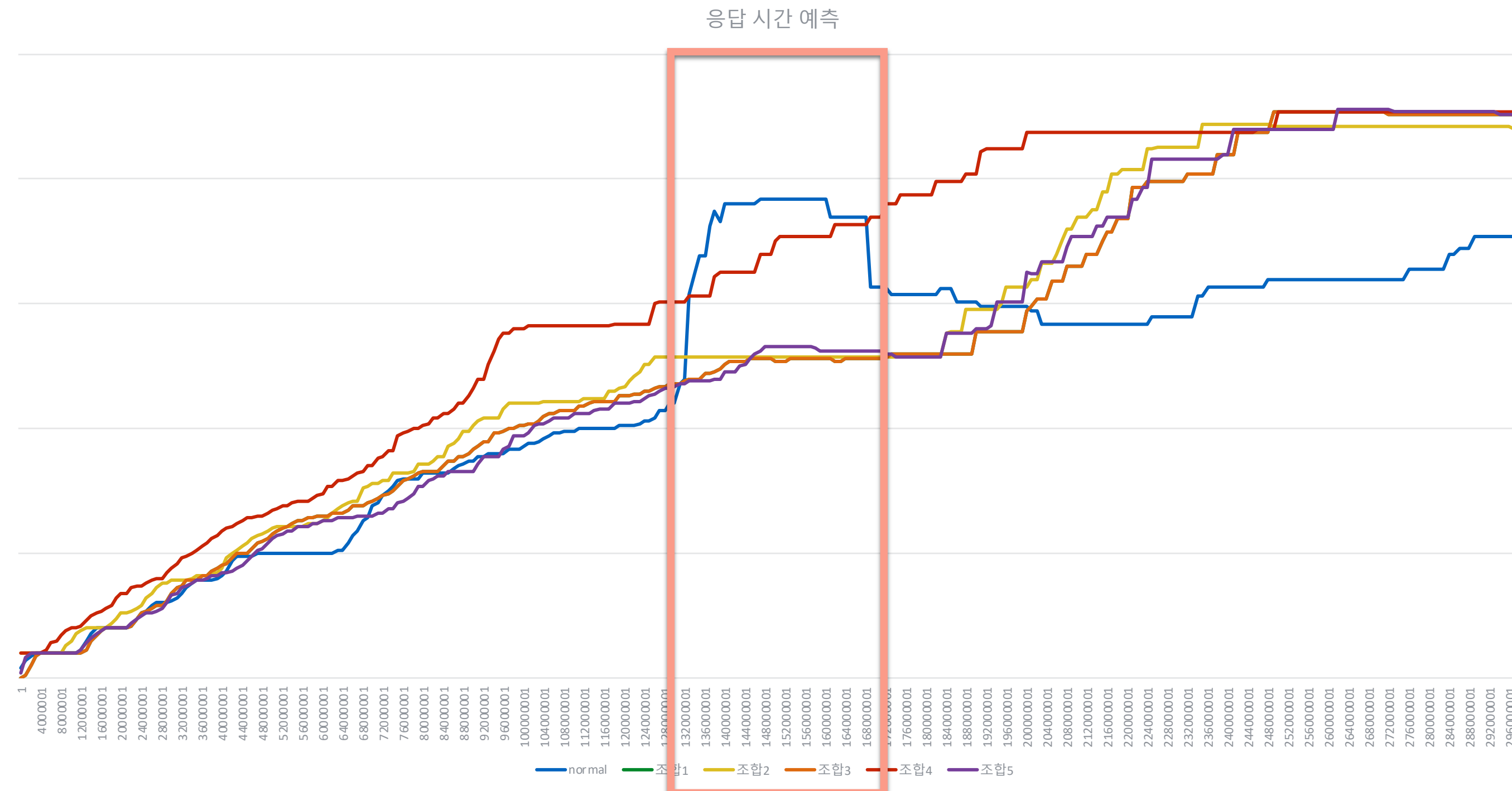


Multi Linear Regression



인퍼런스 서버를 생성하고 API 서버와 연결

- DF를 일정 간격으로 증가하면서 예측 응답 시간을 그래프로 표현



3.3 다중 로지스틱 회귀



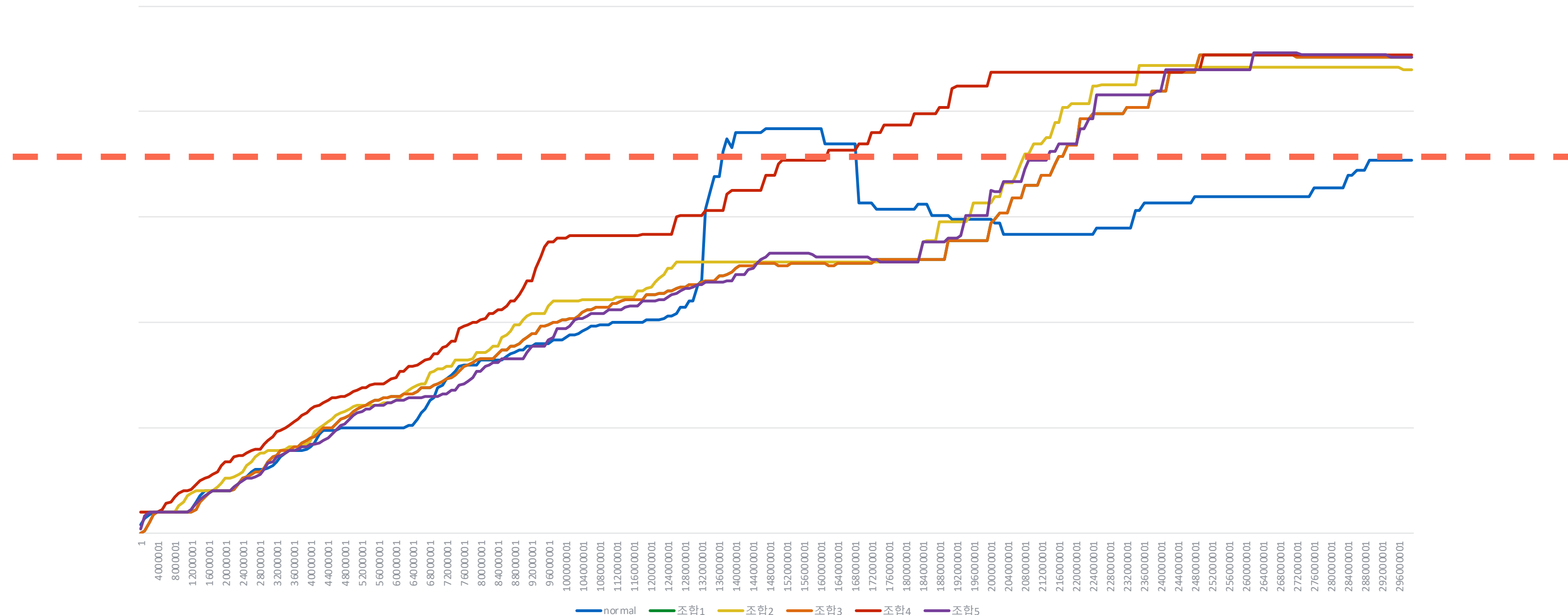
Multi Linear Regression



인퍼런스 서버를 생성하고 API 서버와 연결

- DF를 일정 간격으로 증가하면서 예측 응답 시간을 그래프로 표현

응답 시간 예측



DF 기반 응답시간 예측

- 선형 회귀
- 다항 회귀
- 로지스틱 회귀(분류)

단어 기반 응답시간 예측

- CNN

검색어를 **단일 색인어**로 이루어진 경우
색인어 DF 기반 응답시간 예측 정확도 높지만,

검색어가 **다중 색인어**로 이루어진 경우
색인어 DF 로는 응답시간 예측 정확도 떨어짐

3.4 DF 기반 예측의 한계점



N 쇼핑 노트북					
상세검색	카테고리 더보기				
전체 28,645,330	가격비교 97,590	네이버페이 5,406,067	백화점/홈쇼핑 684,037	쇼핑원도 16,721	해외직구 3,859,866

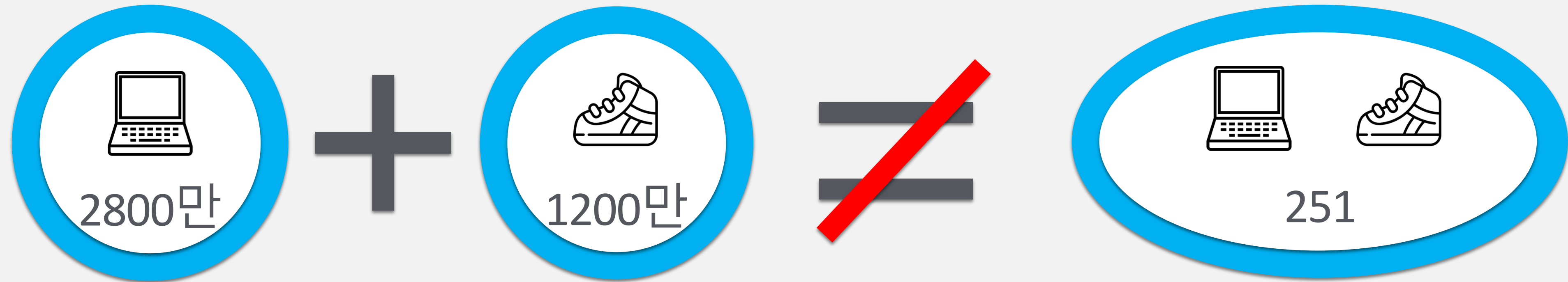


N 쇼핑 운동화					
상세검색	카테고리 더보기				
전체 11,938,661	가격비교 922,662	네이버페이 4,179,641	백화점/홈쇼핑 863,841	쇼핑원도 86,279	해외직구 5,233,746

3.4 DF 기반 예측의 한계점



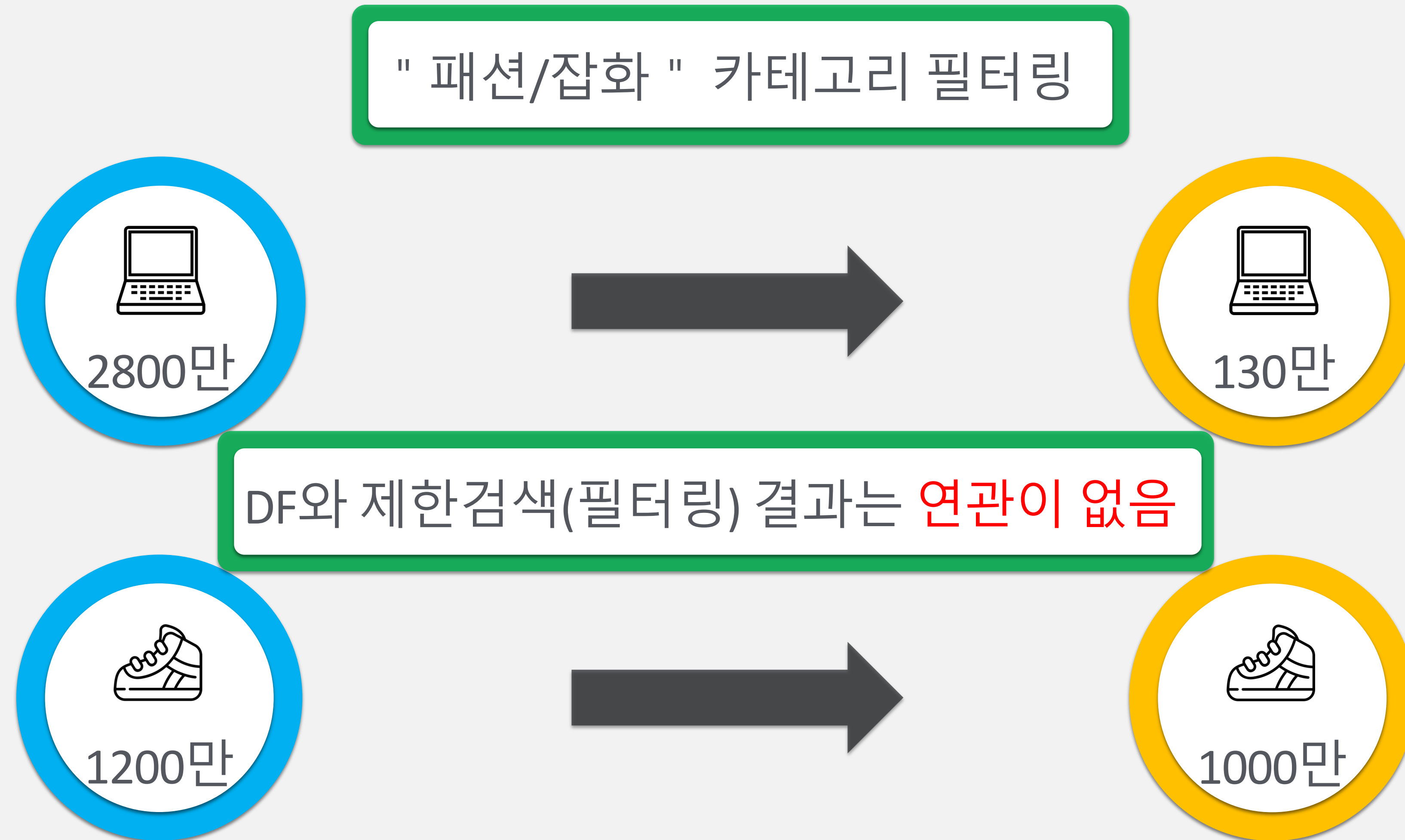
3.4 DF 기반 예측의 한계점



N 쇼핑 노트북 운동화 상세검색 카테고리 더보기					
전체 251	가격비교 9	네이버페이 127	백화점/홈쇼핑 1	쇼핑원도 3	해외직구 167

검색어 **제한검색(필터링)** 은
색인어 DF 기반으로
응답시간 예측 정확도가 떨어진다.

3.4 DF 기반 예측의 한계점



그럼 예측 할 수 있는 또 다른 방법은?

그럼 예측 할 수 있는 또 다른 방법은?



색인어 워드임베딩 모델

3.5 CNN

- [Convolutional neural networks for sentence classification](#) (yoon kim 2014)

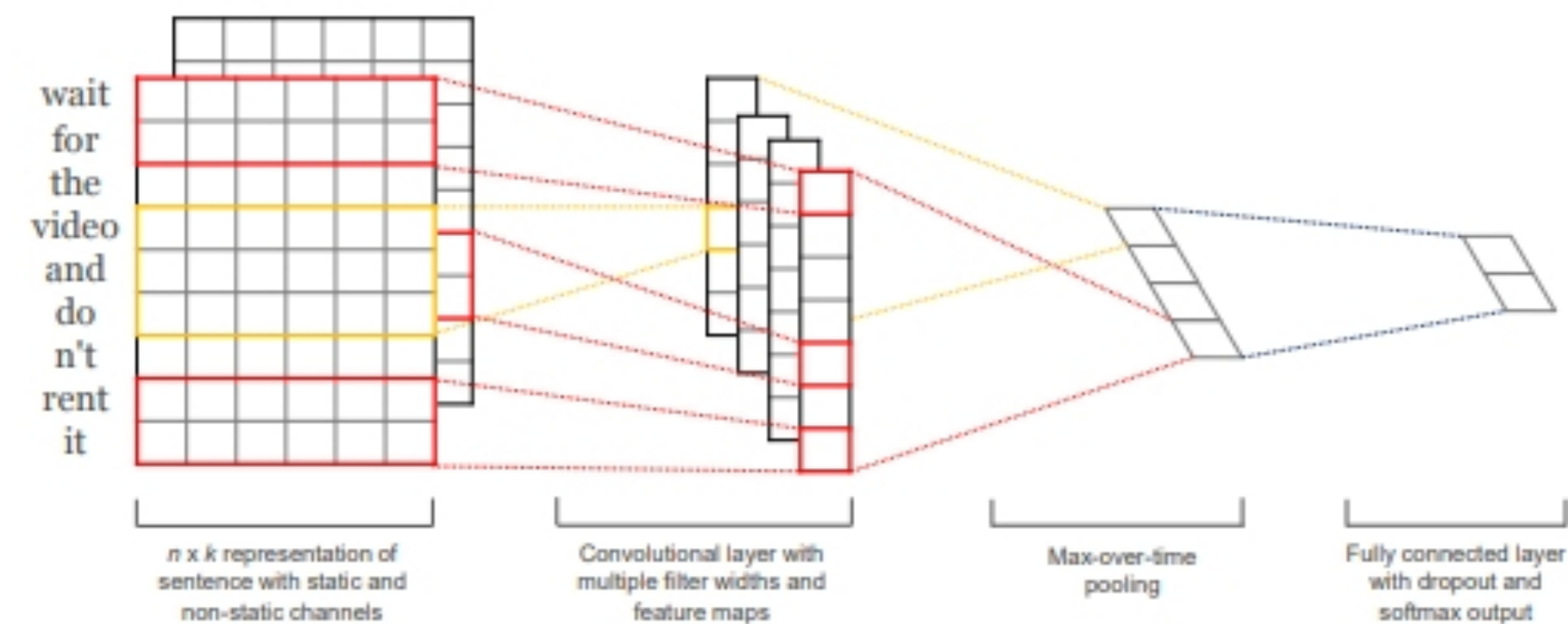


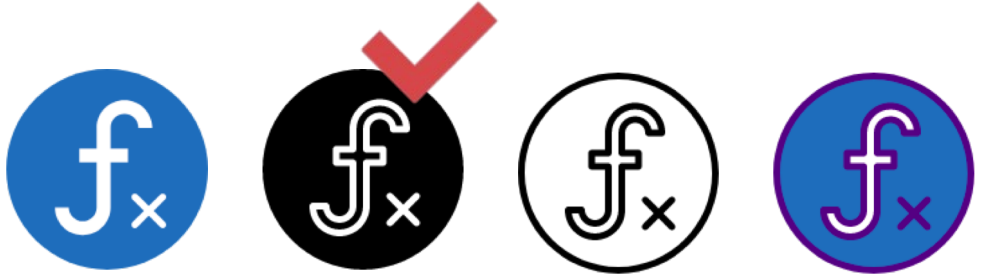
Figure 1: Model architecture with two channels for an example sentence.

- 각 레이어의 입출력 데이터의 형상 유지
- 단어의 공간 정보를 유지하면서 인접 단어와의 특징을 효과적으로 인식
- 복수의 필터로 문장의 특징 추출 및 학습
- 추출한 문장의 특징을 모으고 강화시키고 데이터의 크기를 줄이는 pooling 레이어

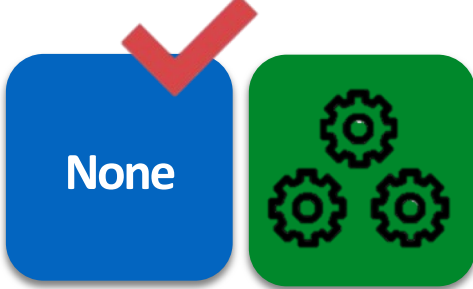
3.5 CNN



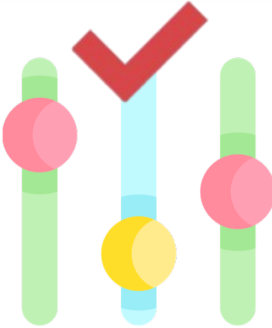
Convolutional Neural Network



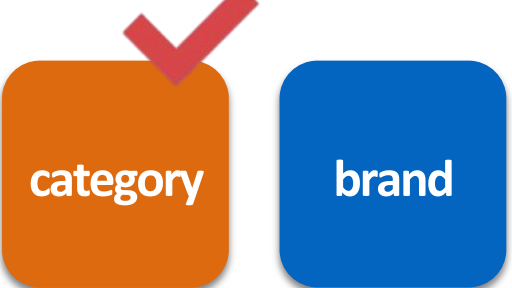
검색 방식



검색 기능



검색 옵션

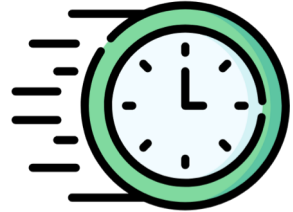


질의 타입



N Terms





응답 시간
(0.1 sec)

응답 시간
(0.2sec)

응답 시간
(0.3 sec)

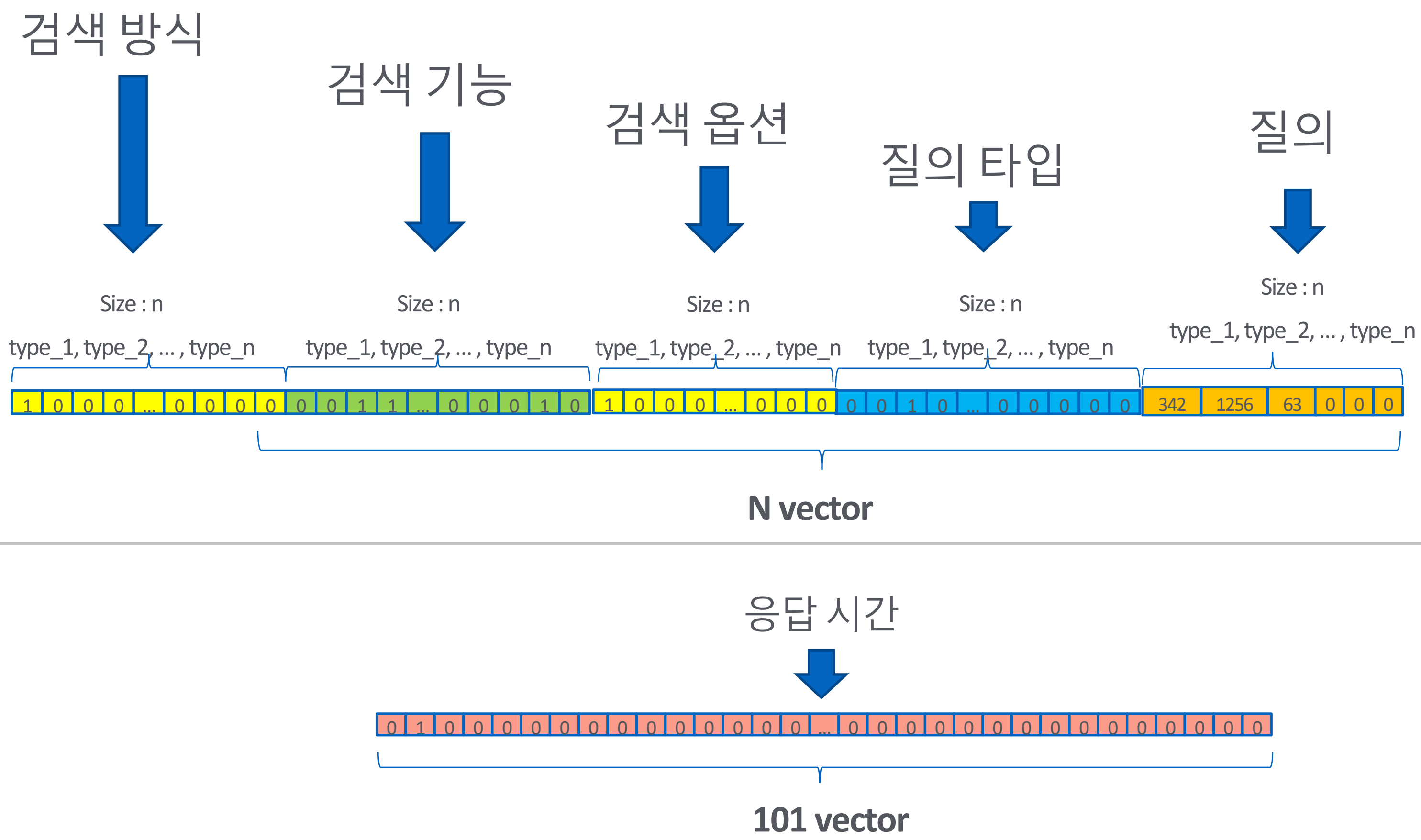
...

응답 시간
(9.9 sec)

응답 시간
(10.0sec)

3.5 CNN

입력 벡터



3.5 CNN



Input Layer

embedding

conv1d

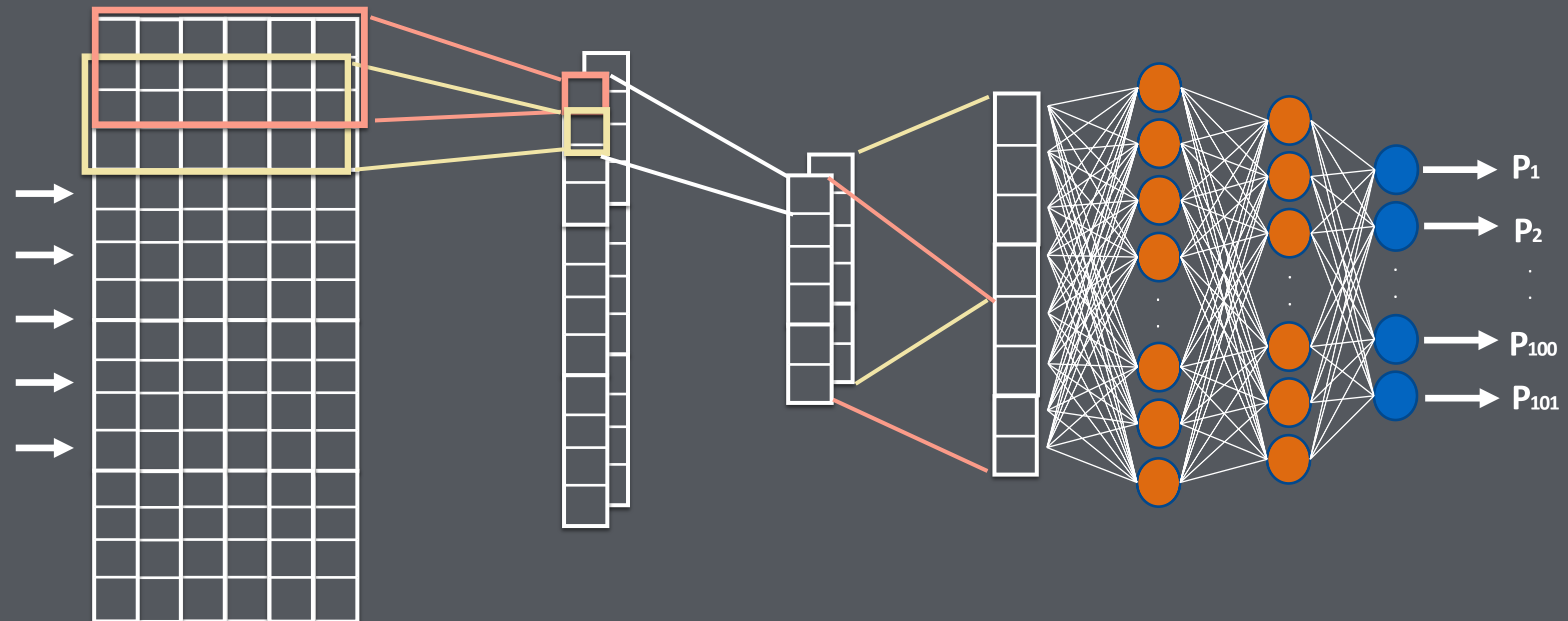
Max pooling1d

flatten

Fully Connected Layer

Output layer

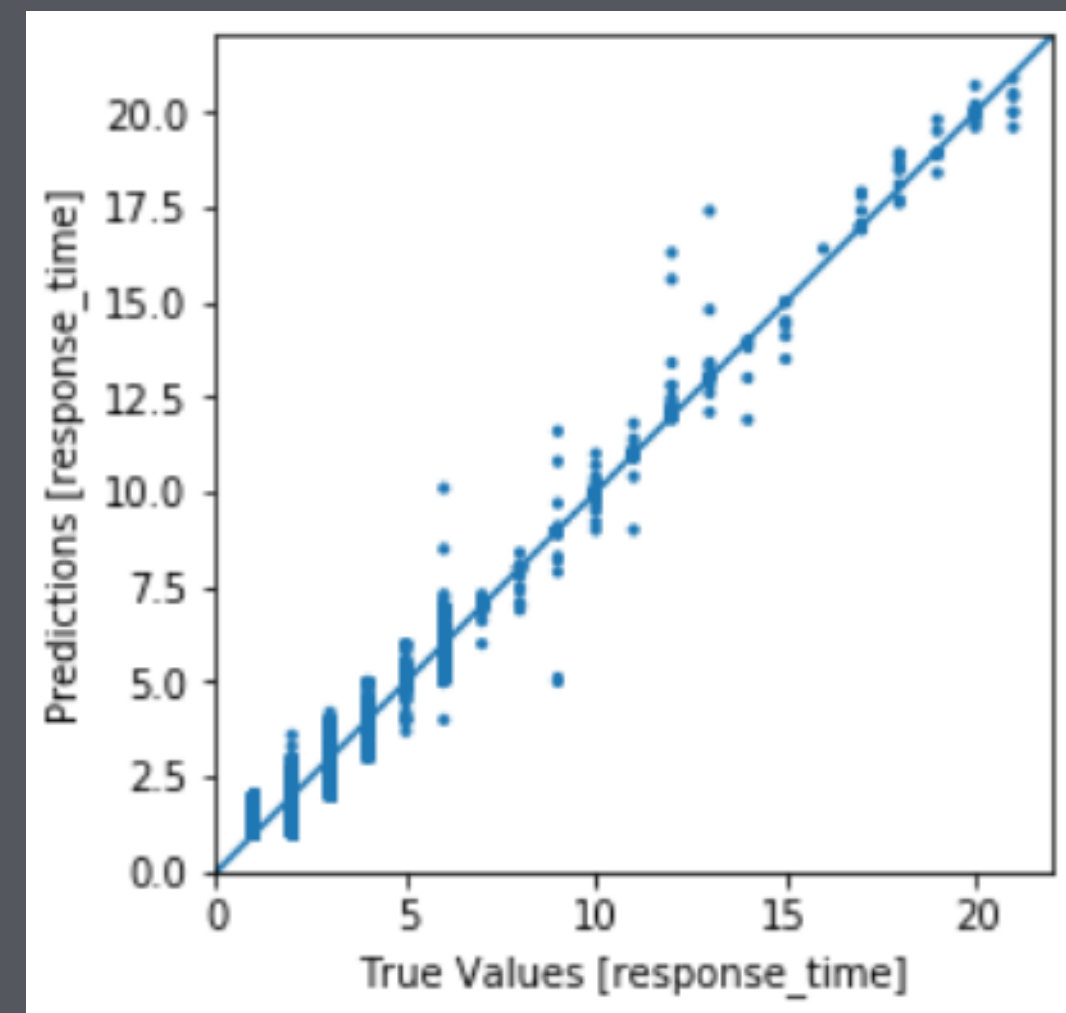
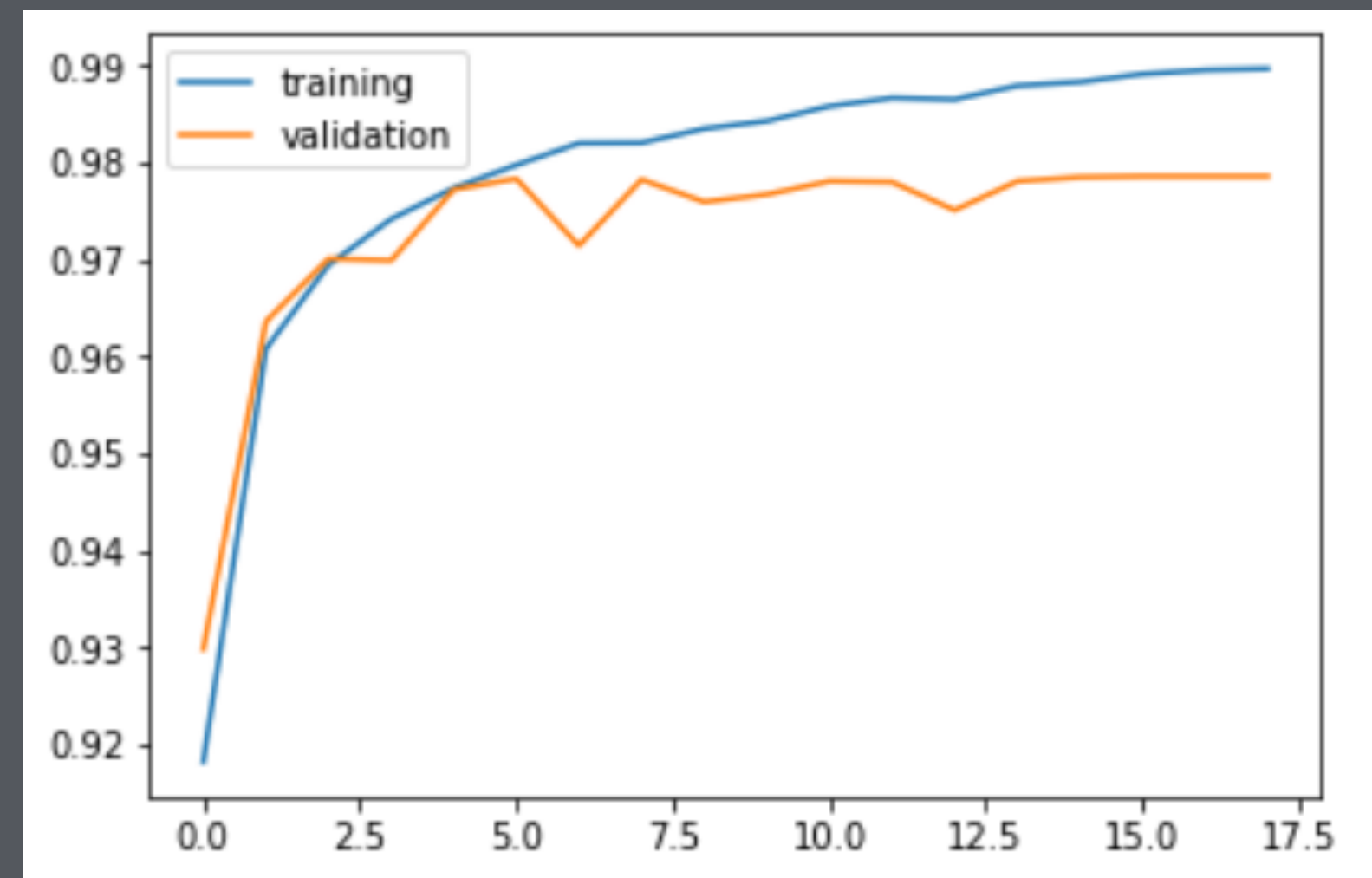
검색 방식
검색 기능
검색 옵션
질의 타입
질의 텀



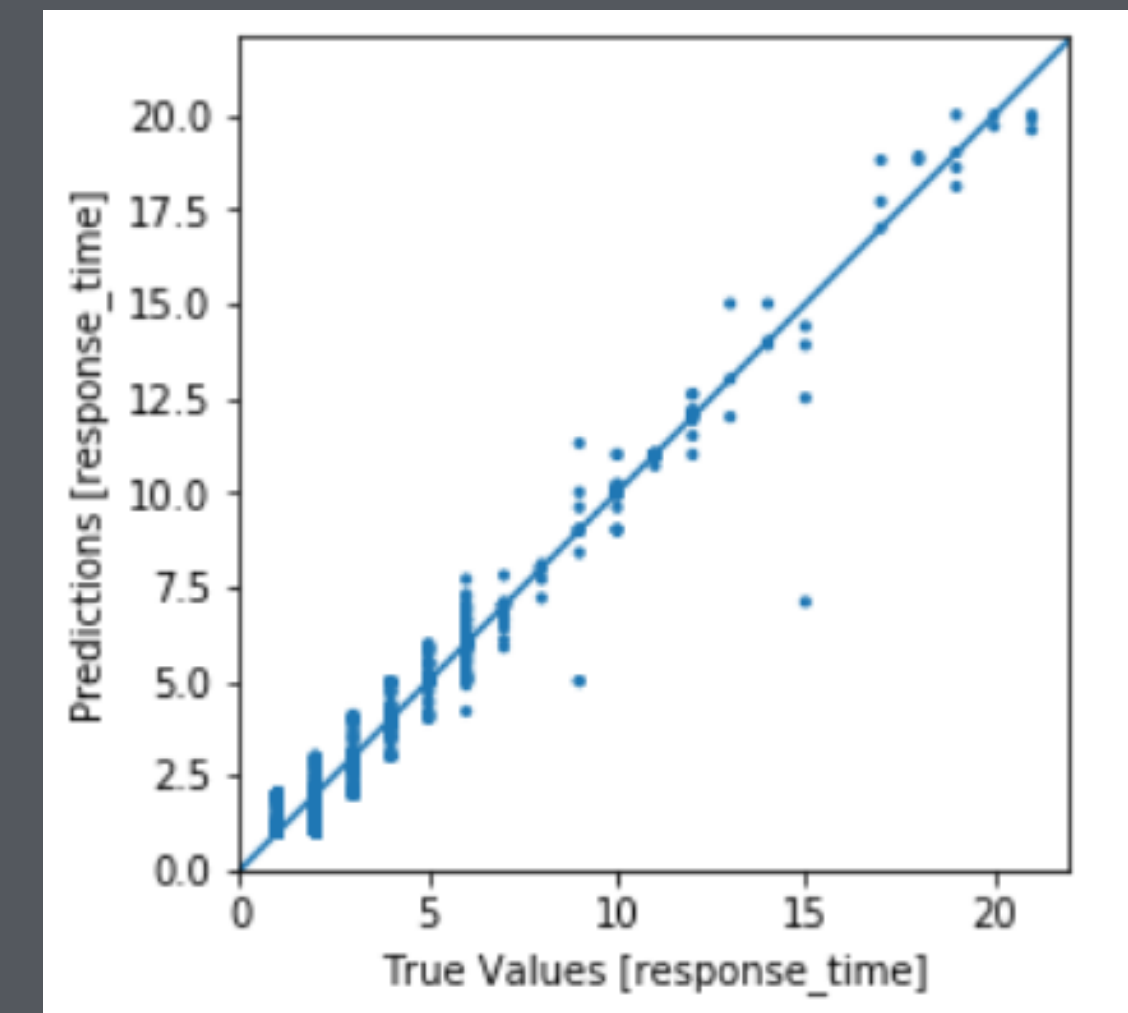
3.5 CNN



Convolutional Neural Network



< 학습 데이터와 예측선 >



< 테스트 데이터와 예측선 >

$$\bar{x} = \frac{\sum_{i=1}^n (x_i * w_i)}{\sum_{i=1}^n w_i}$$

```
===== [Weighted arithmetic mean] =====
[0.08 sec]
```

```
===== [Weighted arithmetic mean] =====
[1.72 sec]
```

$$\bar{x} = \frac{\sum_{i=1}^n (x_i * w_i)}{\sum_{i=1}^n w_i}$$

2nd one-hot encoding

```
===== [model predict] =====
[[0.2 0.8 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  (1, 101)
```

```
===== [Weighted arithmetic mean] =====
[0.08 sec]
```

```
[[[0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.3 0.2  
   0.5 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  
   0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  
   0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  
   0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  
   0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0. ]]  
(1, 101)]
```

```
===== [Weighted arithmetic mean] =====
[1.72 sec]
```

$$\bar{x} = \frac{\sum_{i=1}^n (x_i * w_i)}{\sum_{i=1}^n w_i}$$

```
===== [s_label] =====
a0 a1 a2 a3 a4 a5 a6 a7 a8 a9 a10 a11 a12 a13
a14 a15 a16 a17 a18 a19 a20 a21 a22 a23 a24 a25 a26 a27 a28
a29 a30 a31 a32 a33 a34 a35 a36 a37 a38 a39 a40 a41 a42 a43
a44 a45 a46 a47 a48 a49 a50 a51 a52 a53 a54 a55 a56 a57 a58
a59 a60 a61 a62 a63 a64 a65 a66 a67 a68 a69 a70 a71 a72 a73
a74 a75 a76 a77 a78 a79 a80 a81 a82 a83 a84 a85 a86 a87 a88
a89 a90 a91 a92 a93 a94 a95 a96 a97 a98 a99 a100
2153 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0

=====

===== [model predict] =====
[[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.3 0.2
 0.5 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. ]]
(1, 101)

=====

===== [Weighted arithmetic mean] =====
[1.72 sec]
=====
```


$$\bar{x} = \frac{\sum_{i=1}^n (x_i * w_i)}{\sum_{i=1}^n w_i}$$

```

===== [s_label] =====
a0 a1 a2 a3 a4 a5 a6 a7 a8 a9 a10 a11 a12 a13
a14 a15 a16 a17 a18 a19 a20 a21 a22 a23 a24 a25 a26 a27 a28
a29 a30 a31 a32 a33 a34 a35 a36 a37 a38 a39 a40 a41 a42 a43
a44 a45 a46 a47 a48 a49 a50 a51 a52 a53 a54 a55 a56 a57 a58
a59 a60 a61 a62 a63 a64 a65 a66 a67 a68 a69 a70 a71 a72 a73
a74 a75 a76 a77 a78 a79 a80 a81 a82 a83 a84 a85 a86 a87 a88
a89 a90 a91 a92 a93 a94 a95 a96 a97 a98 a99 a100
2153 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
=====
===== [model predict] =====
[[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.3 0.2
 0.5 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
(1, 101)
=====
===== [Weighted arithmetic mean] =====
[1.72 sec]
=====

```

$$\bar{x} = \frac{\sum_{i=1}^n (x_i * w_i)}{\sum_{i=1}^n w_i}$$

16th one-hot encoding

```
===== [Weighted arithmetic mean] =====
→ [0.08 sec]
```

```
===== [Weighted arithmetic mean] =====
[1.72 sec]
```

3.5 CNN

모델이 예측한 output vector의 가중 산술 평균
(= 예측 응답시간)

$$\bar{x} = \frac{\sum_{i=1}^n (x_i * w_i)}{\sum_{i=1}^n w_i}$$

[실제 응답시간]

0.1 초

2nd one-hot encoding

===== [s_label] =====														
	a0	a1	a2	a3	a4	a5	a6	a7	a8	a9	a10	a11	a12	a13
a14	a15	a16	a17	a18	a19	a20	a21	a22	a23	a24	a25	a26	a27	a28
a29	a30	a31	a32	a33	a34	a35	a36	a37	a38	a39	a40	a41	a42	a43
a44	a45	a46	a47	a48	a49	a50	a51	a52	a53	a54	a55	a56	a57	a58
a59	a60	a61	a62	a63	a64	a65	a66	a67	a68	a69	a70	a71	a72	a73
a74	a75	a76	a77	a78	a79	a80	a81	a82	a83	a84	a85	a86	a87	a88
a89	a90	a91	a92	a93	a94	a95	a96	a97	a98	a99	a100			
3010	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

[예측 응답시간]

0.0 초 일 확률 20%

0.1 초 일 확률 80%

가중 산술 평균 : 0.08 초

===== [model predict] =====																
[[	0.2	0.8	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
(1, 101)																

===== [Weighted arithmetic mean] =====													
→	[0.08	sec]											
=====													

[실제 응답시간]

1.6 초

16th one-hot encoding

===== [s_label] =====														
	a0	a1	a2	a3	a4	a5	a6	a7	a8	a9	a10	a11	a12	a13
a14	a15	a16	a17	a18	a19	a20	a21	a22	a23	a24	a25	a26	a27	a28
a29	a30	a31	a32	a33	a34	a35	a36	a37	a38	a39	a40	a41	a42	a43
a44	a45	a46	a47	a48	a49	a50	a51	a52	a53	a54	a55	a56	a57	a58
a59	a60	a61	a62	a63	a64	a65	a66	a67	a68	a69	a70	a71	a72	a73
a74	a75	a76	a77	a78	a79	a80	a81	a82	a83	a84	a85	a86	a87	a88
a89	a90	a91	a92	a93	a94	a95	a96	a97	a98	a99	a100			
2153	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

===== [model predict] =====																
[[	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.3	0.2
0.5	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
(1	101)															

```
===== [Weighted arithmetic mean] =====
[1.72 sec]
```

[예측 응답시간]

1.6 초 일 확률 30%

1.7 초 일 확률 20%

1.8 초 일 확률 50%

3.5 CNN

모델이 예측한 output vector의 가중 산술 평균
(= 예측 응답시간)

$$\bar{x} = \frac{\sum_{i=1}^n (x_i * w_i)}{\sum_{i=1}^n w_i}$$

[실제 응답시간]

0.1 초

2nd one-hot encoding

===== [s_label] =====														
	a0	a1	a2	a3	a4	a5	a6	a7	a8	a9	a10	a11	a12	a13
a14	a15	a16	a17	a18	a19	a20	a21	a22	a23	a24	a25	a26	a27	a28
a29	a30	a31	a32	a33	a34	a35	a36	a37	a38	a39	a40	a41	a42	a43
a44	a45	a46	a47	a48	a49	a50	a51	a52	a53	a54	a55	a56	a57	a58
a59	a60	a61	a62	a63	a64	a65	a66	a67	a68	a69	a70	a71	a72	a73
a74	a75	a76	a77	a78	a79	a80	a81	a82	a83	a84	a85	a86	a87	a88
a89	a90	a91	a92	a93	a94	a95	a96	a97	a98	a99	a100			
3010	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

[예측 응답시간]

0.0 초 일 확률 20%

0.1 초 일 확률 80%

가중 산술 평균 : 0.08 초

===== [model predict] =====													
[[0.2	0.8	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
(1, 101)													

===== [Weighted arithmetic mean] =====													
[0.08	sec]												

[실제 응답시간]

1.6 초

16th one-hot encoding

===== [s_label] =====														
	a0	a1	a2	a3	a4	a5	a6	a7	a8	a9	a10	a11	a12	a13
a14	a15	a16	a17	a18	a19	a20	a21	a22	a23	a24	a25	a26	a27	a28
a29	a30	a31	a32	a33	a34	a35	a36	a37	a38	a39	a40	a41	a42	a43
a44	a45	a46	a47	a48	a49	a50	a51	a52	a53	a54	a55	a56	a57	a58
a59	a60	a61	a62	a63	a64	a65	a66	a67	a68	a69	a70	a71	a72	a73
a74	a75	a76	a77	a78	a79	a80	a81	a82	a83	a84	a85	a86	a87	a88
a89	a90	a91	a92	a93	a94	a95	a96	a97	a98	a99	a100			
2153	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

===== [model predict] =====													
[[0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
(1, 101)													

===== [Weighted arithmetic mean] =====													
[1.72	sec]												

[예측 응답시간]

1.6 초 일 확률 30%

1.7 초 일 확률 20%

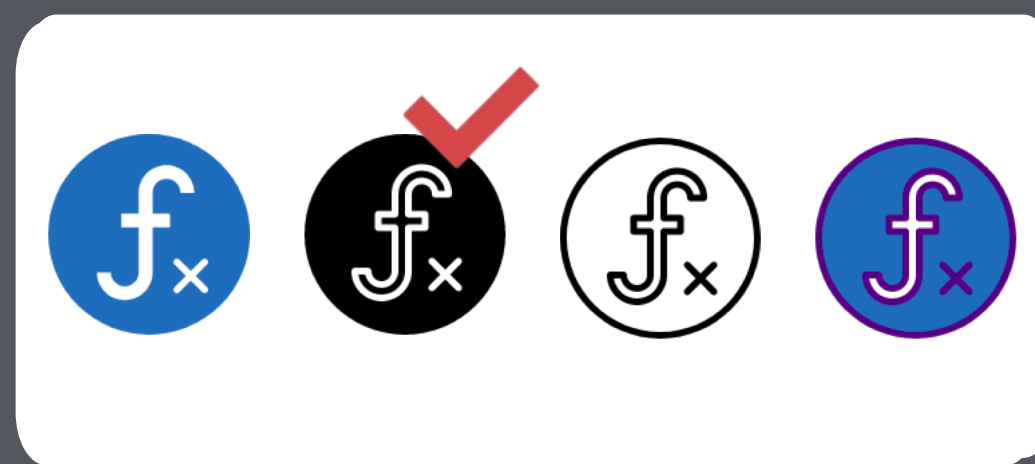
1.8 초 일 확률 50%

가중 산술 평균 : 1.72 초

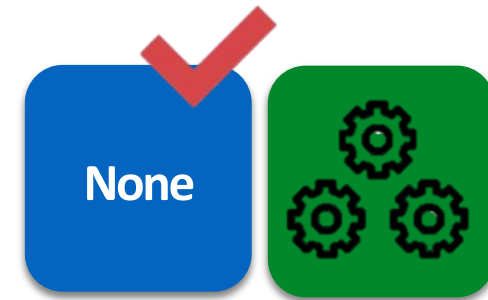
3.6 CNN 검색 결과 유무 (feat.)

검색어를 가지고 검색 결과 유무를 예측하자

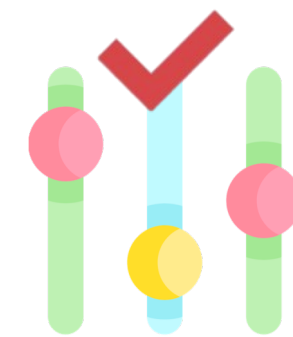
3.6 CNN 검색 결과 유무 (feat.)



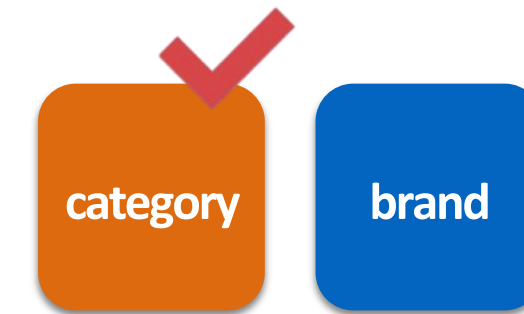
검색 방식



검색 기능



검색 옵션



질의 타입



N Terms



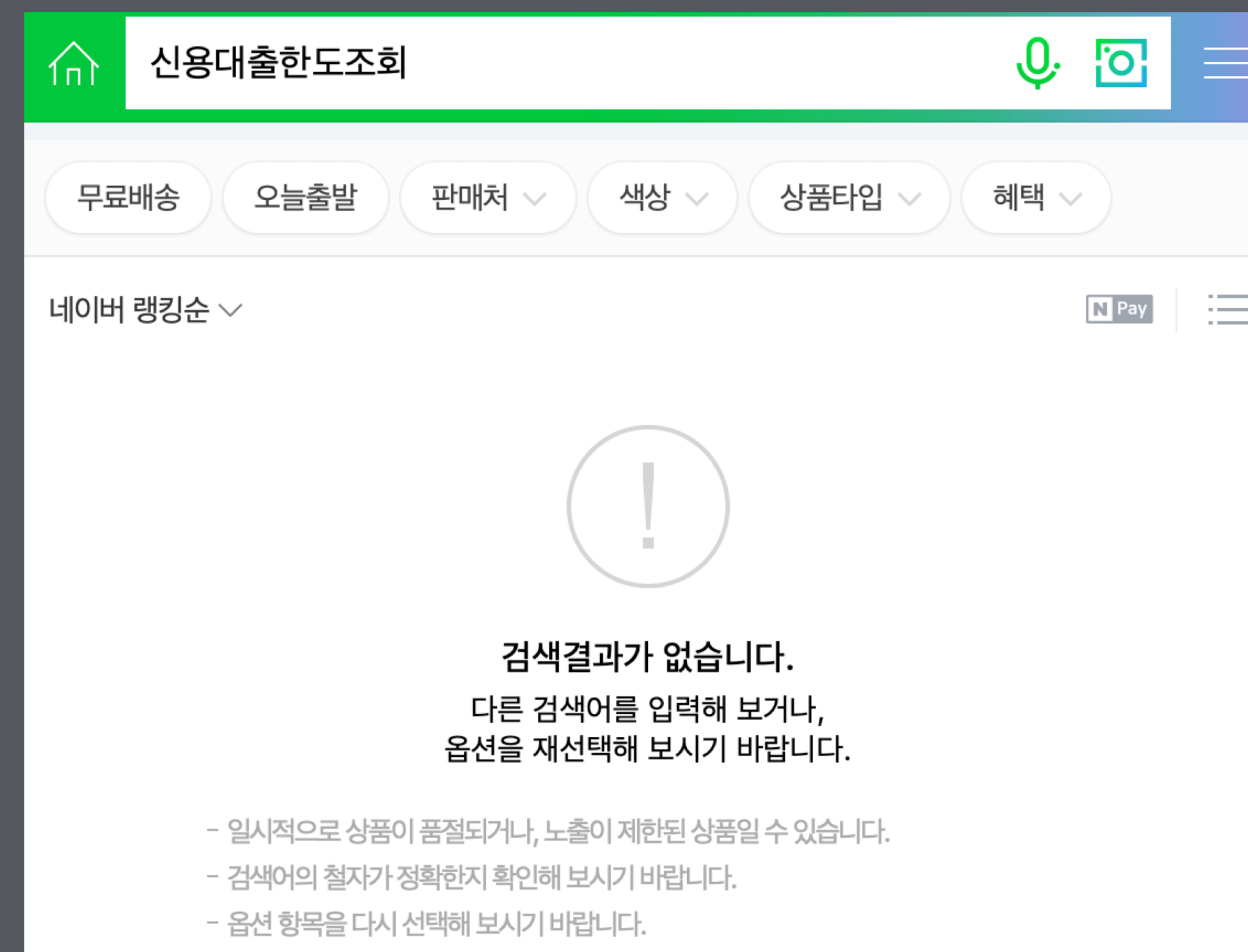
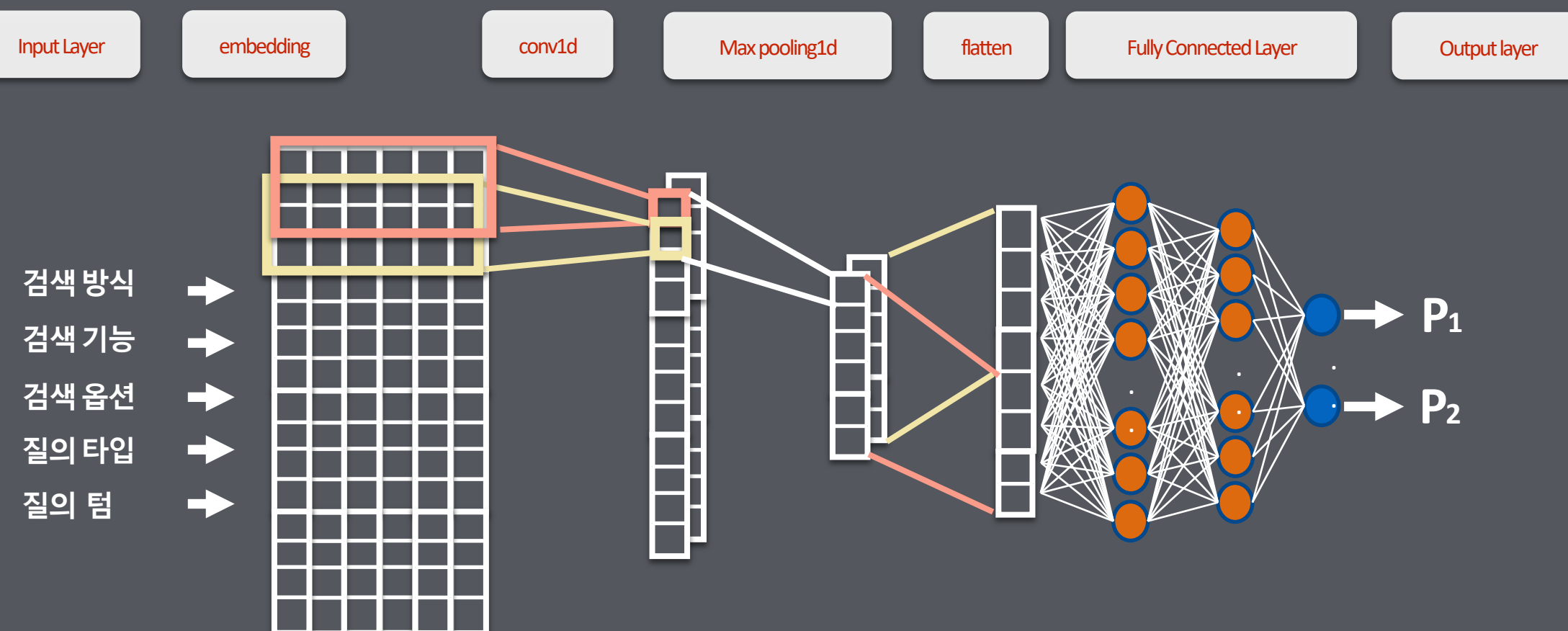
결과 없음

결과 있음

3.6 CNN 검색 결과 유무 (feat.)

검색 결과 유무 예측

- CNN 응답 시간 예측 모델에서 출력 레이어를 검색결과 존재 유무 Binary classification 으로 교체



< 검색 결과 없는 질의 >

4. Integration / Future Works

4.1 시스템 도입 요구 사항

빠른 응답

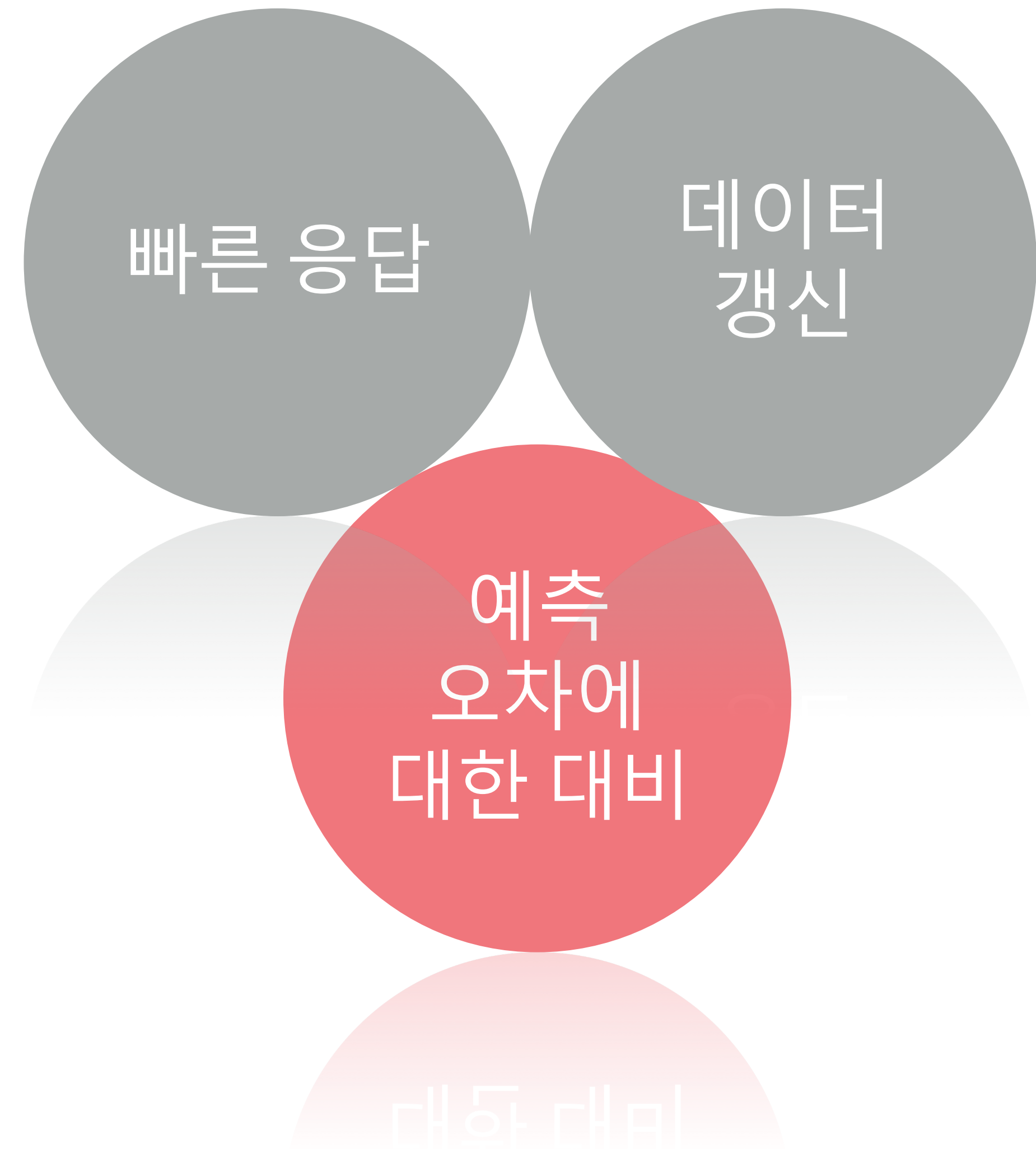
- 응답 시간에 예측 모델을 호출한 시간이 포함

데이터 갱신

- 매일 데이터 추가 / 삭제가 발생하므로 학습모델을 매일 갱신 필요

예측 오차에 대한 대비

- 예측이 잘못된 경우 검색 품질 저하 가능성



4.2 Integration



4.3 시스템 도입 효과

응답시간 예측

- 시스템 효율성 향상
- 고비용 검색 질의를 기준 시간 내 검색결과를 반환할 수 있도록 단계적인 랭킹과 같은 최적화 알고리즘 구동
- 시스템 가용량의 증가에 따른 검색 요청 처리량 증가

검색 결과 유무 예측기대효과

- 하단 검색 서버의 불필요한 작업을 줄임
- 수 천대에 이르는 서버간 통신 네트워크 비용을 절감

4.3 시스템 도입 효과

시스템 적용 전 테스트 결과

- 고비용 검색 유형을 추출하여 알고리즘 최적화 적용
- Full set 데모 환경에서 테스트 진행

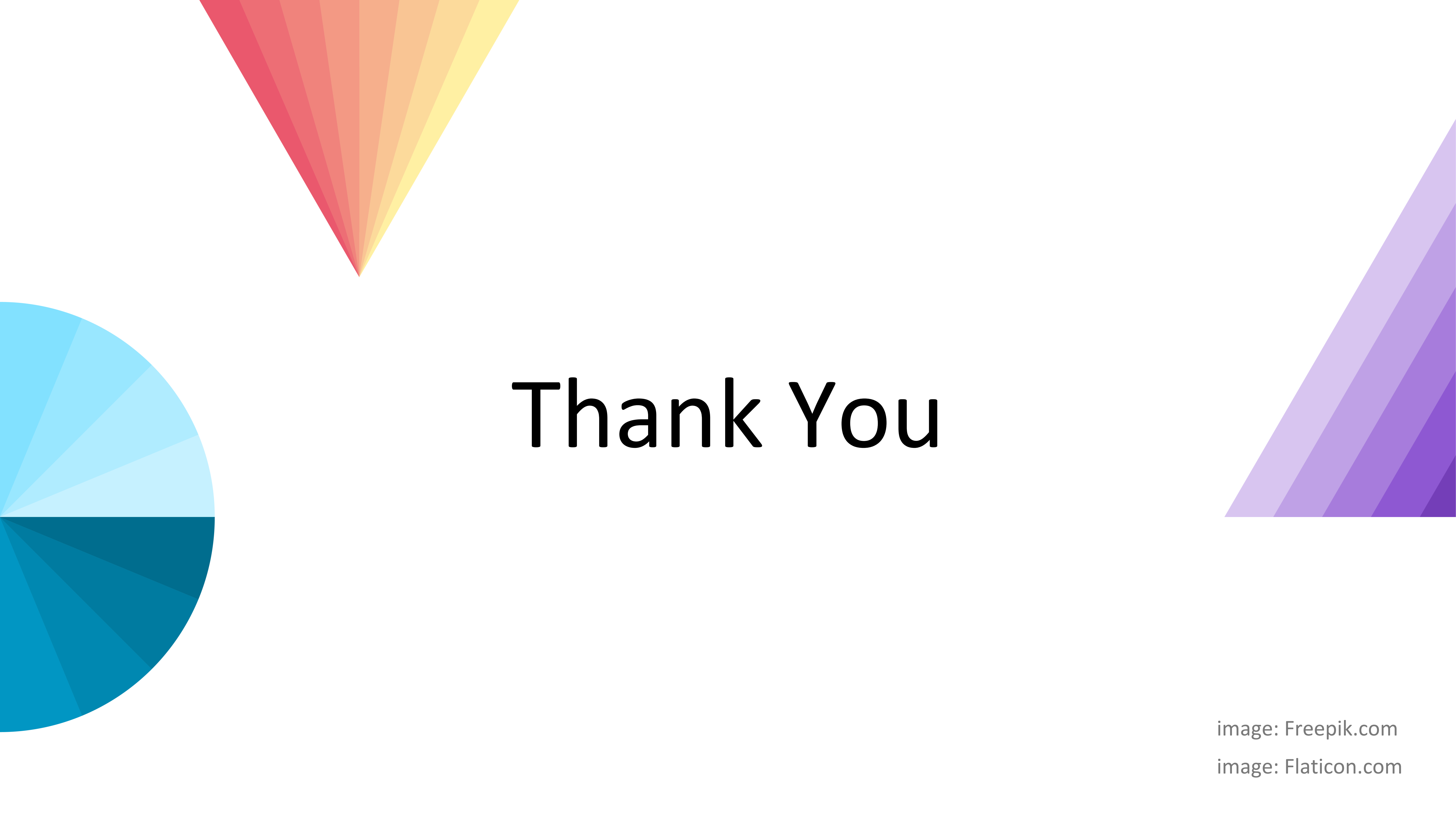
적용 효과	
응용 계층	CPU 사용률 비슷 최대 응답시간 95% 감소
검색 계층	최대 CPU 사용률 95% 감소 최대 검색시간 95% 감소

4.4 모델 Summary

	장점	한계점
회귀 (1 Term 대응)	- 코퍼스 관리 필요 없음. - 새로운 질의에 대해서도 예측할 수 있음.	- N term, 필터링과 같이 DF와 연관성이 없는 케이스는 예측 어려움.
CNN (N Term 대응)	- 여러 단어로 구성된 검색어의 응답시간, 응답결과 예측할 수 있음. - 필터링 적용된 검색유형 예측 가능.	- 기존에 존재하지 않는 단어이거나, 학습되지 않은 유형에 대해서는 정확한 예측이 어려움.



Q & A



Thank You

image: Freepik.com

image: Flaticon.com