

20182830 최민석 20192761 김정하

# ML Final round

# CONTENTS INDEX

01

1라운드 분석 과정

02

2라운드 분석 과정

03

피드백



첫번째 챕터, ●

# 1라운드 분석 과정

## ● 1라운드 분석 과정

각자 다른 feature 사용



개별 성능 극대화




Ensemble

## ● 1라운드 분석 과정



Feature1  
( 35967 , 4142 )

Scaling -> 성능 하락  
Feature selection -> 성능 하락



Feature2  
( 35967 , 744 )

Scaling -> 성능 하락  
Feature selection -> 성능 향상

## ● 1라운드 분석 과정

Feature1  
( 35967 , 4142 )

Voting Regressor :  
LGBM , CatBoost

8.36151

Feature2  
( 35967 , 744 )

BayesianOptimization을 이용한  
LGBM , CatBoost 튜닝  
→ Voting Regressor

8.47188



# ● 1라운드 분석 과정



Voting Regressor :  
LGBM , CatBoost

8.36151



BayesianOptimization을 이용한  
LGBM , CatBoost 튜닝  
→ Voting Regressor

8.47188

8 : 2 비율로 평균  
**8.34843**

## ● 1라운드 피드백 및 2라운드 계획

1. 성능이 잘 나오는 모델은 LGBM, CatBoost로 비교적 한정적
2. 모델링을 동일하게 해도 feature가 다르다면 성능 향상 가능
3. 1라운드에서 만들었던 Feature1과 Feature2는 서로 상호보완적 관계라고 추정



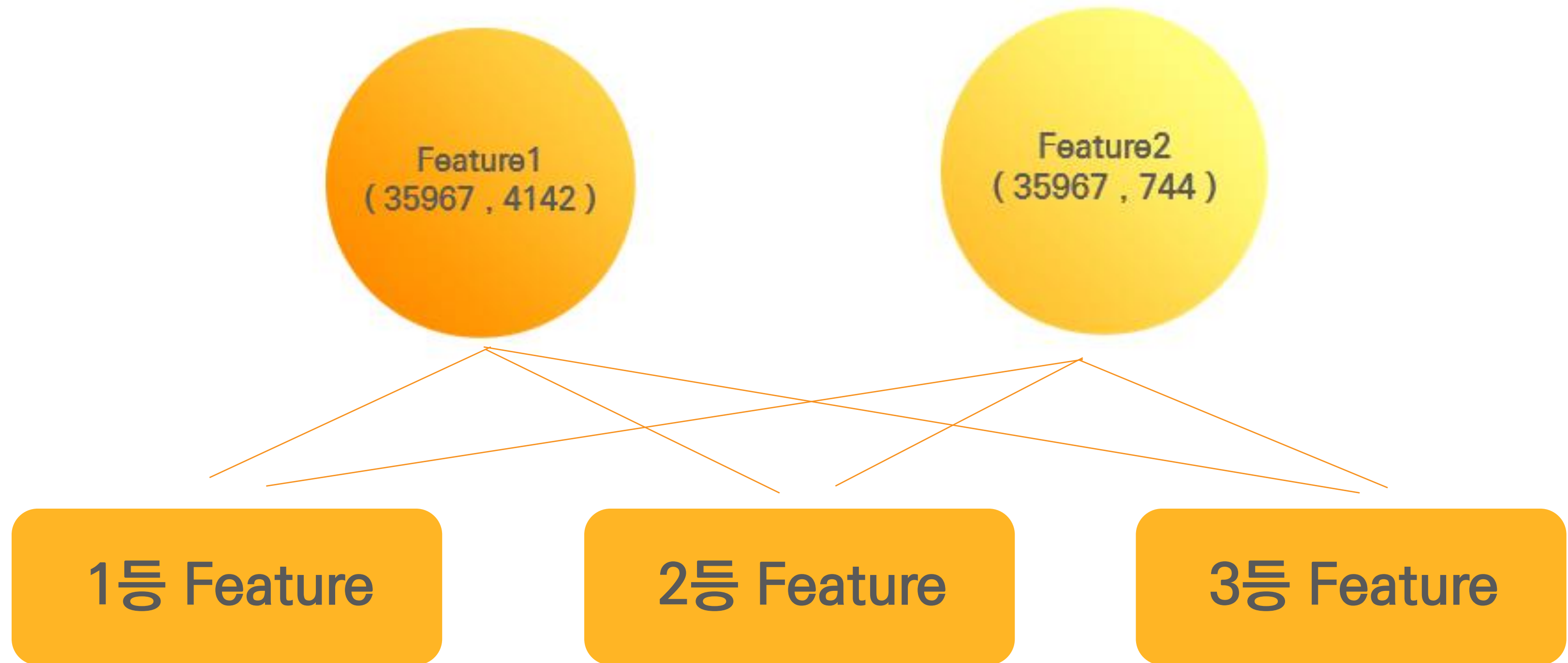
다수의 Feature csv를 만들어 모델링 후 앙상블 하자!



두번째 챕터, ●●

# 2라운드 분석 과정

## ● 2 라운드 분석 과정



## ● 2라운드 분석 과정



Scaling  
Selection  
W2V 및 기타 feature 일부 추가



6개의 feature csv 생성

## ● 2라운드 분석 과정



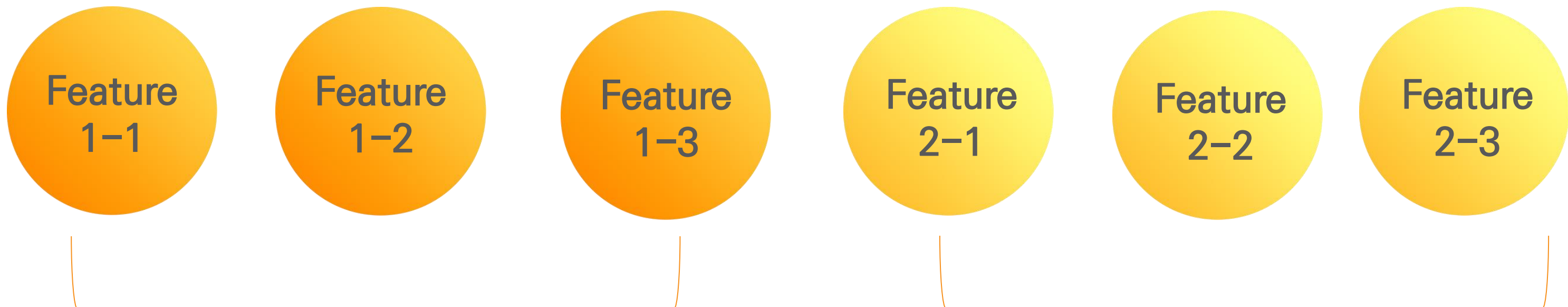
LGBM – Bayesian 튜닝  
Voting Regressor – LGBM, CatBoost, Ridge



12개의 pred csv 생성



## ● 2라운드 분석 과정



LGBM – Bayesian 튜닝  
성능 : 8.14 ~ 8.17

Voting Regressor  
– LGBM, CatBoost, Ridge  
성능 : 8.20 ~ 8.21

## ● 2라운드 분석 과정

Feature  
1-1

Feature  
1-2

Feature  
1-3

LGBM – Bayesian 튜닝  
성능 : 8.14 ~ 8.17

Feature  
2-1

Feature  
2-2

Voting Regressor  
– LGBM, CatBoost, Ridge  
성능 : 8.20 ~ 8.21



5. dnn\_submission\_8.15961.csv

8 : 2 비율로 평균  
**8.03990**

## ● 2라운드 분석 과정

Feature  
1-1

Feature  
1-2

Feature  
1-3

LGBM – Bayesian 튜닝  
성능 : 8.14 ~ 8.17



5. dnn\_submission\_8.15961.csv

Feature  
2-1

Feature  
2-2

Voting Regressor  
– LGBM, CatBoost, Ridge  
성능 : 8.20 ~ 8.21

**POWER MEAN**  
**8.04685**

## ● 2라운드 분석 과정

8 : 2 비율로 평균  
8.03990



POWER MEAN  
8.04685



최종 csv 8.04232  
(7.98236 )



세번째 챕터



# 피드백

## ● 피드백

- 1라운드에 최초로 설계하였던 대로  
feature1과 feature2 간의 상호보완을 통해서 성능향상을 이뤄냄
- 하지만 Public 점수를 높이기 위해서 특정 Feature에 가중치를 주어,  
Public 성능 향상을 이뤘지만 과적합이 발견

결론 : 상호보완 가능하고 성능이 비슷한 feature끼리 앙상블을 할 때에는  
가중치를 적절하게 배분하는 것이 중요.



감사합니다.

20182830 최민석 20192761 김정하