

# 고객데이터 바탕으로 연령 예측하기

2021.05.26 ~ 2021.06.15

신예주+윤경서

20192776 신예주

20192784 윤경서

# 목차

## 1차 Competition

어떻게 진행할 것인지  
계획 수립 및 목표 설정

## Feature 생성

연령대에 영향을 미칠 만한  
유의미한 feature 생성  
(각자 따로 생성)

## Feature 완성

과적합의 위험이 있는 feature  
삭제 및 정규화 여부 결정

## 가중평균

기본 모델로 가중평균 진행

11등

## 2차 Competition

1차에서 보완해야 될 부분  
2차의 방향성 및 목표 설정

## Feature 완성

비슷한 방식으로 생성된  
feature를 제외한 feature  
추가 및 W2V사용

## Modeling Parameter Tuning

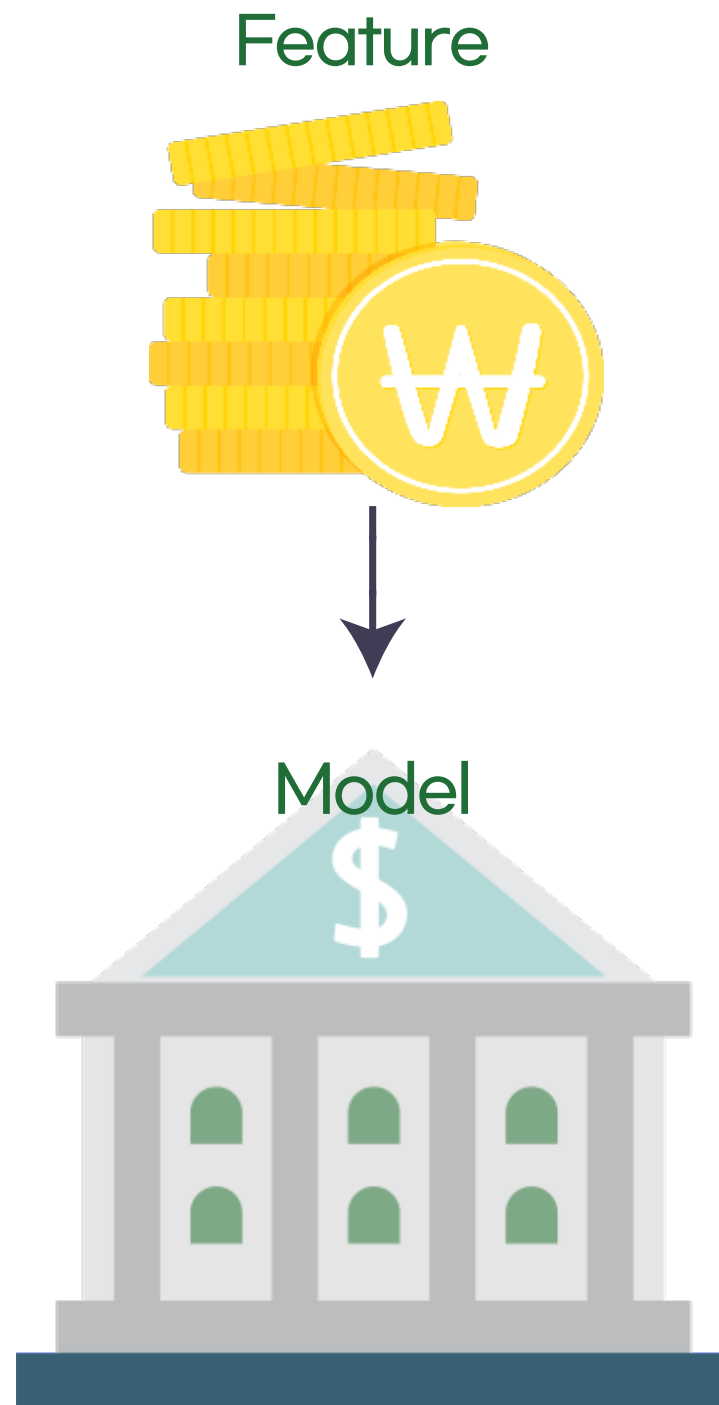
Optimization을 통해  
최적의 파라미터 튜닝

## 가중평균 Mean Ensemble

만들어진 csv파일을 가지고  
가중평균과 mean ensemble

3등

# 1차 COMPETITION



## 유의미한 feature 생성

예측값이 feature에 영향을 많이 받을 것이라고 예측되어 최대한 유의미한 결과를 이끌어 낼 수 있는 feature를 생성하고자 함.

## 모델링보다는 feature에 집중

1차 때에는 모델링보다 feature를 만드는 것에 집중을 하고, 2차 때 모델링에 집중하고자 함. 또한 feature의 수가 많으면 오버피팅의 위험이 덜 할 것이라고 생각되어, 최소 4000개의 feature를 생성하고자 함.

# Feature

Data Cleansing

Engineering

Selection

**수치형**

주기 할인율  
금액 - 총, 실, 환불, 최대

**범주형**

날짜 브랜드  
상품군 지점

원본 데이터

표준화 + 정규화  
실행

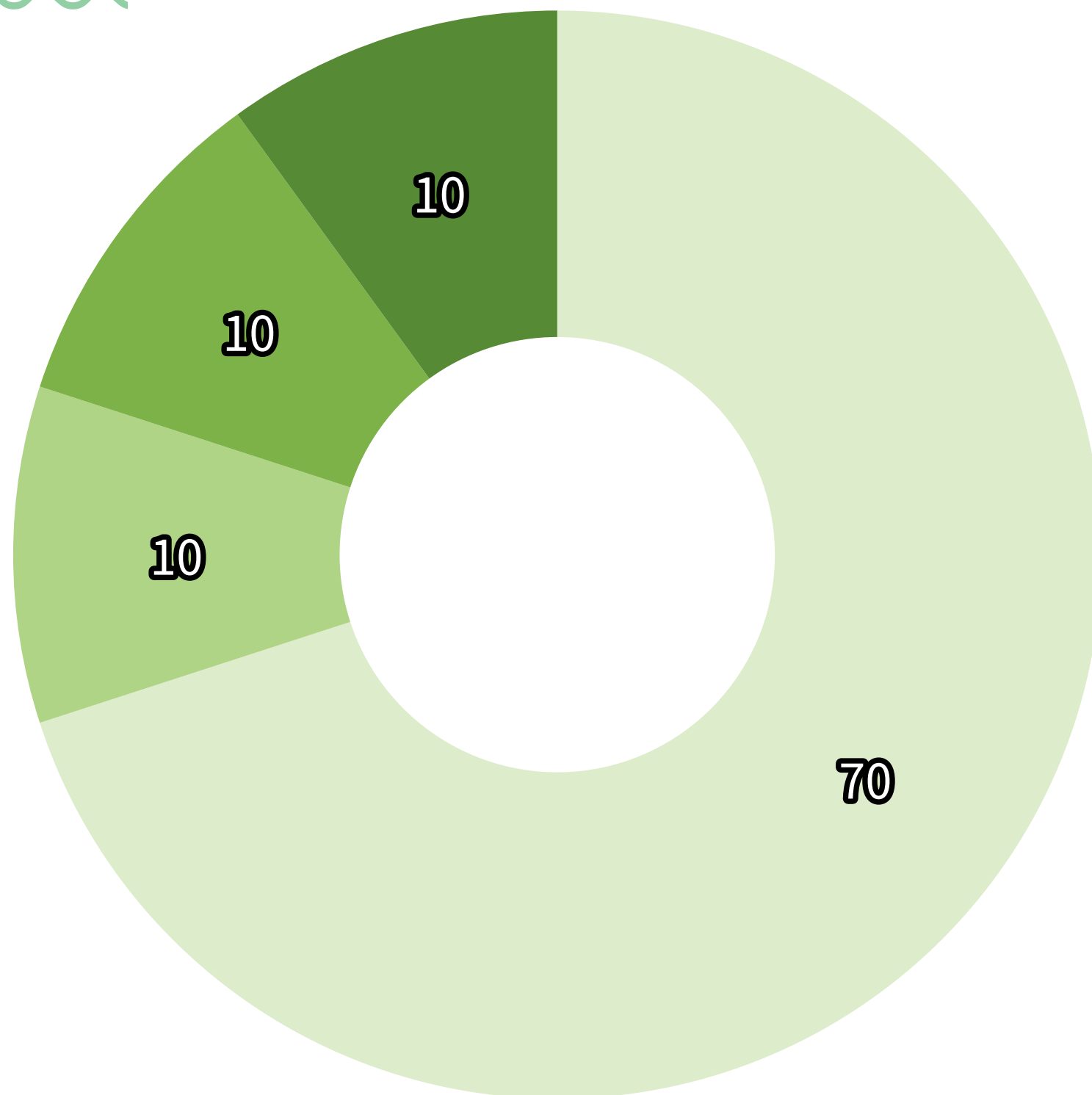
4705

Datetime 생성  
part\_nm 통일

가공하지 않은  
data사용

3387

# 가중평균



- Cat 70%
- XGB 10%
- RandomForest 10%
- Gradient 10%

# 2차 COMPETITION 방향성

---

오류때문에  
시도해보지 못했던 모델  
의 오류 해결

기존  
Model

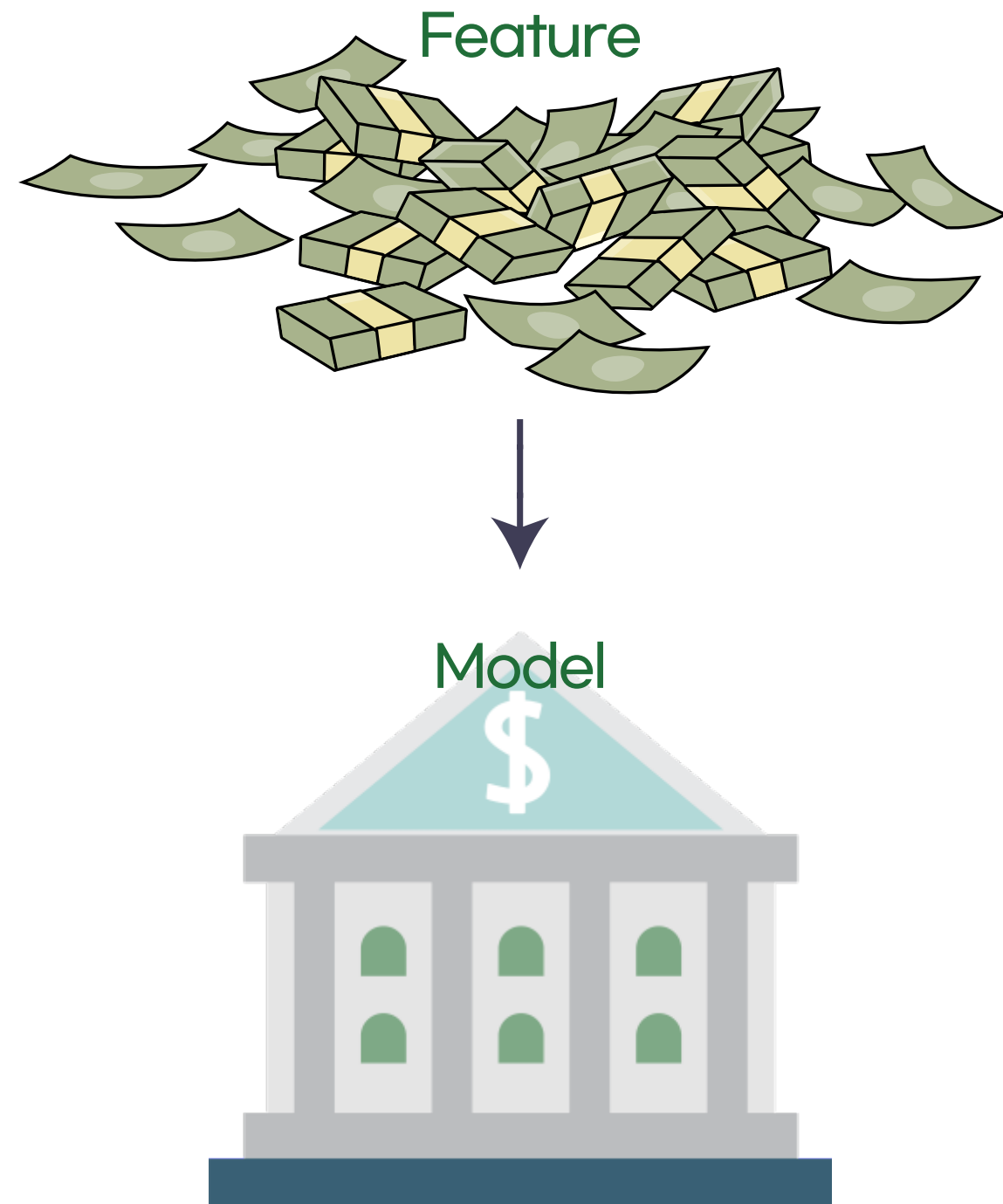
단일모델의  
성능을 높이는데 최적의  
값을 찾자!

Feature

Ridge, DNN 등  
새로운 모델 사용

새로운  
Model

# 2차 COMPETITION



## feature 보완

보다 유의미한 데이터를 포함한 feature를 추가하여 모델의 성능을 최대치로 발휘할 수 있는 환경을 만들고자 함.

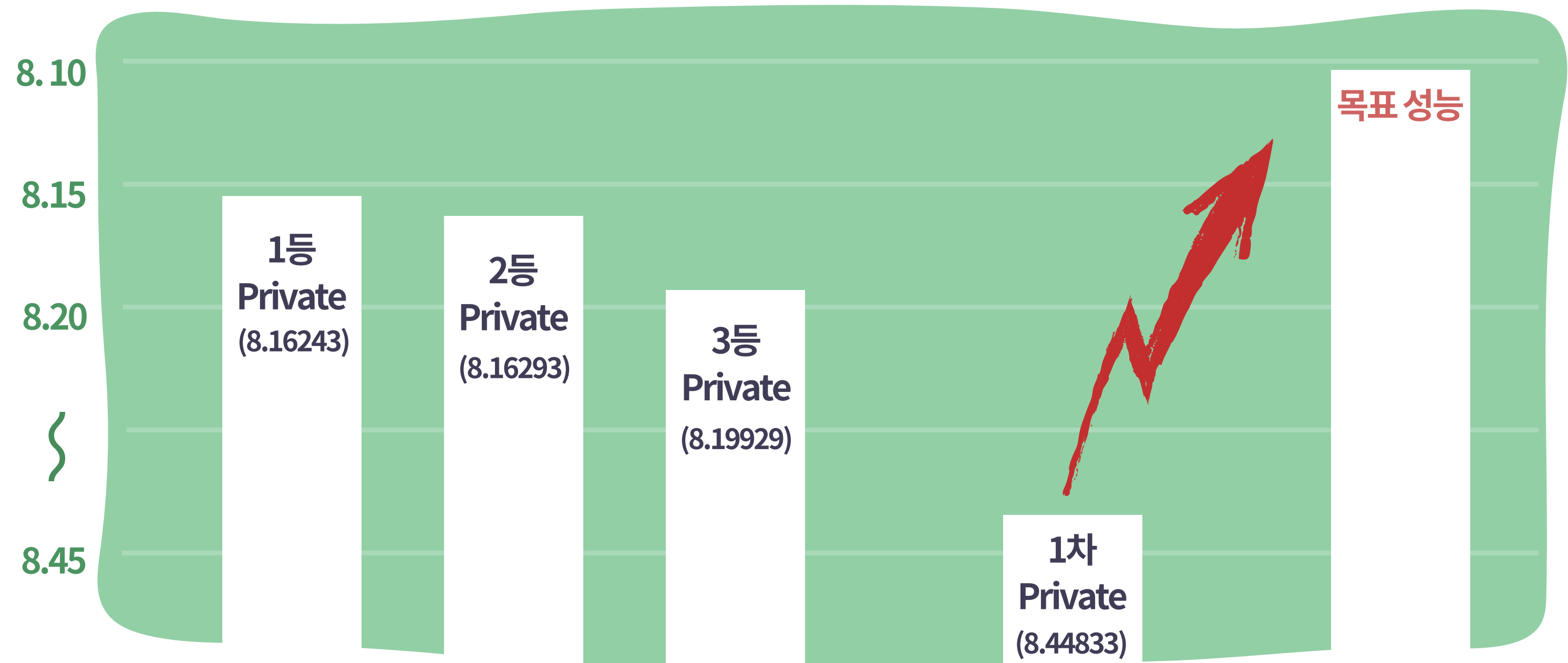
## 모델링 & 튜닝

1차 때는 시도하지 못했던 다양한 모델링을 통해, 우리의 feature에 맞는 모델을 찾고 튜닝을 하면서 모델 성능의 최대치를 이끌어내고자 함.

## 가중평균, 앙상블, 스택킹 시도

성능이 오른 모델을 바탕으로 1차 때에는 효과가 미미했던 앙상블과 가중 평균의 효과를 극대화하고, 스택킹을 시도하고자 함.

# 2차 COMPETITION 목표



피쳐를 보완하여 단일모델의 성능으로 1등, 2등, 3등의 Private 값을 넘기자!



# Feature 완성

## 기존 feature에 추가

기존의 feature에 수치형 데이터가 부족하다고 생각되어, 수치형 feature 추가 및 범주형 데이터 선택적 추가

## feature selection

총 2번의 공통된 feature selection 진행  
19096 → 10502 → 7351

## Word 2 Vec

각 단어의 조합을 통해 새로운 feature를 생성해주는 Word 2 Vec 기법 사용  
7351 → 7751

여러번의 feature selection을 통해  
유의미한 feature를 남기고자 함.  
공통 3번 + 모델링 전 1번(각 모델별)  
= 총 4번의 selection 진행

## 모델 별 feature selection

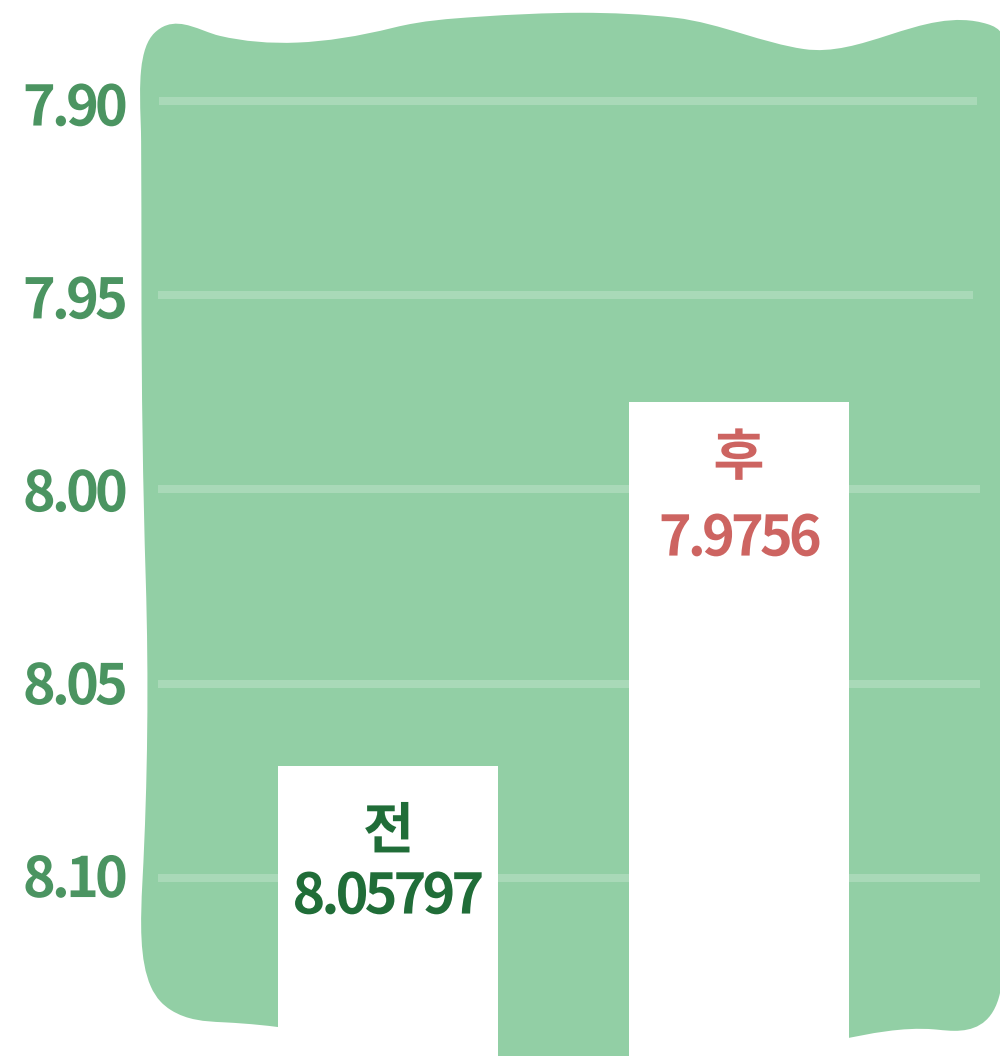
각 모델을 돌리기 전,  
코드에 포함된 feature selection을 통해  
최적의 성능을 위한 selection 진행

## feature selection

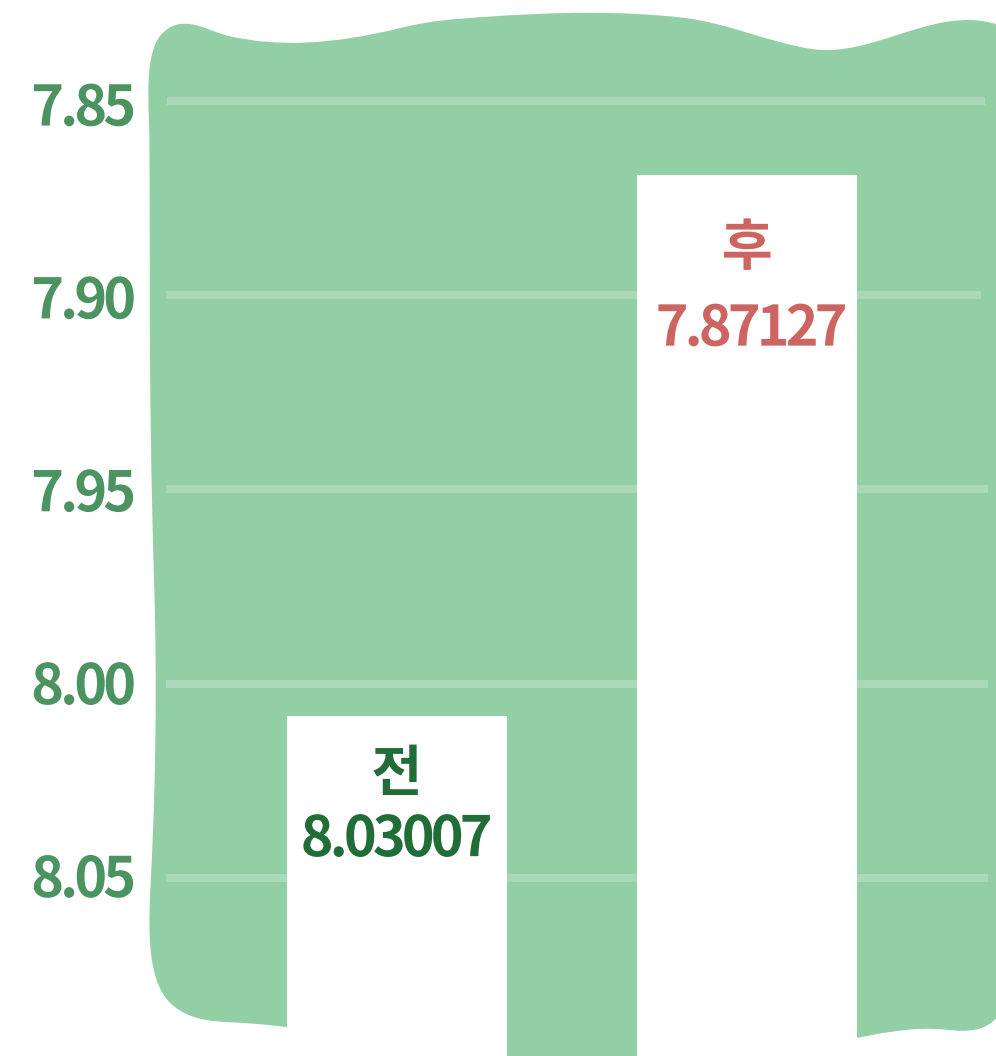
많은 양의 feature로 인해 모델이  
돌아가는 데 너무 오랜 시간이 소요되어  
feature selection 한 번 더 진행  
7751 → 7595

# Modeling & Parameter Tuning

## LGBM



## Catboost



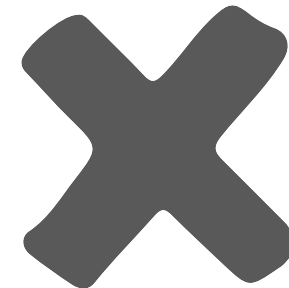
## 그 밖의 모델들

- Ridge : 여러 번의 튜닝과 파라미터 조정을 해도 9.26이 최대  
→ 다른 모델과의 성능차이가 너무 커서 빠기로 결정
- DNN : 모델 튜닝을 통해 8.5까지 성능 향상
- Gradient : 성능은 괜찮지만, 시간이 너무 오래 걸려서 빠기로 결정
- XGB : LGBM의 성능이 더 높은 관계로 빠기로 결정

# 가중평균 & Mean Ensemble 1

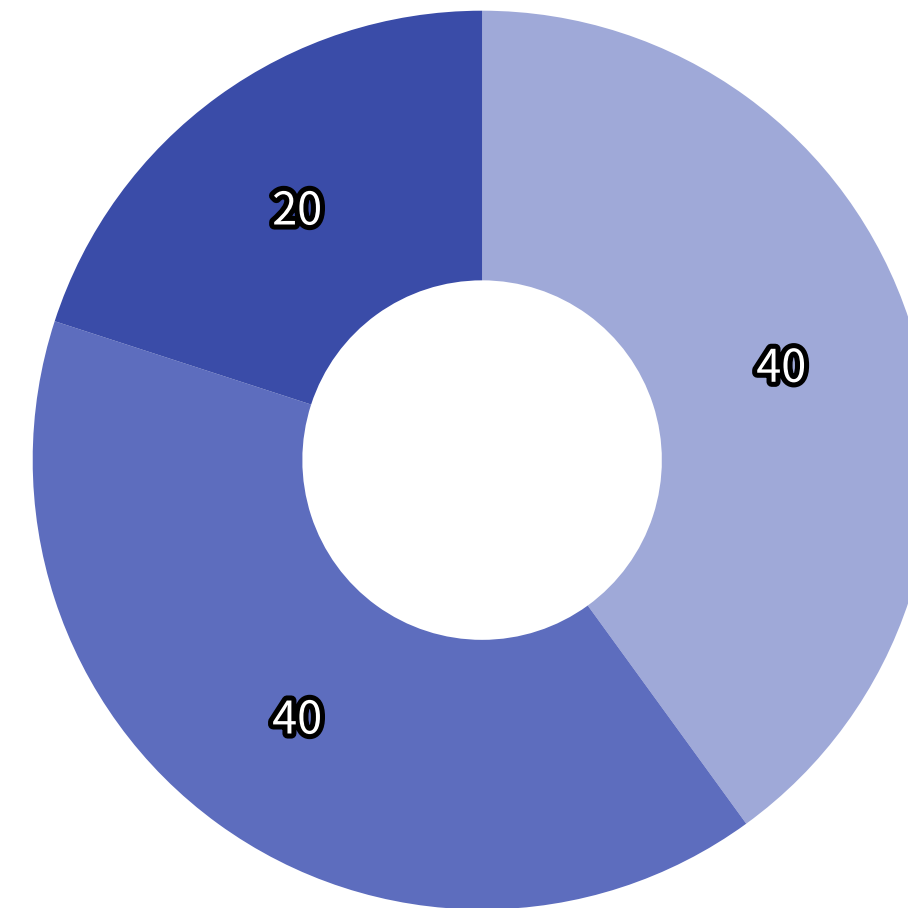


1등 submission  
2등 submission  
3등 submission



양상블

[우리모델\_1]  
우리 모델을 이용한 csv 파일 가중평균

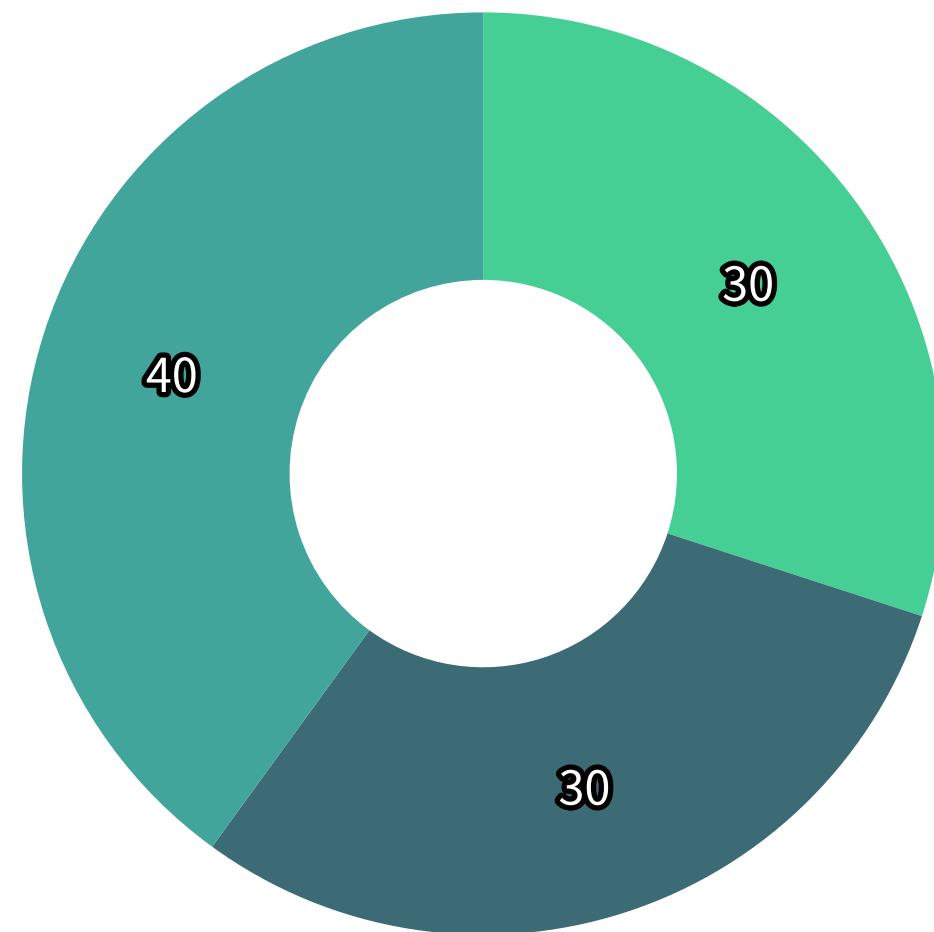


LGBM 40% Cat 40% DNN 40%

LGBM과 CAT이 같은 트리계열 부스팅 모델이라 둘만 넣어 가중평균을 하는 것보다 dnn을 포함하는 것이 낫다고 판단. dnn의 성능이 생각보다 낮아서 위와 같이 가중치를 부여함

# 가중평균 & Mean Ensemble 2

[sub\_123]  
1, 2, 3등 submission 가중평균

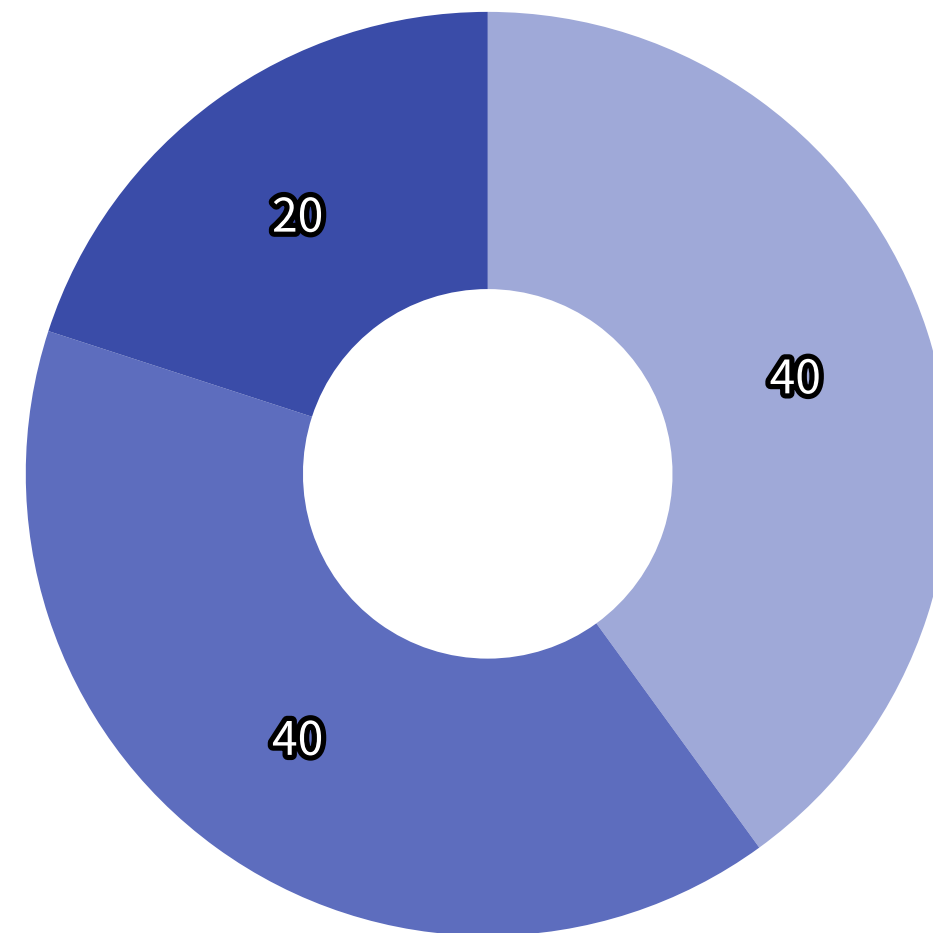


1등 30% 2등 30% 3등 40%



양상블

[우리모델\_1]



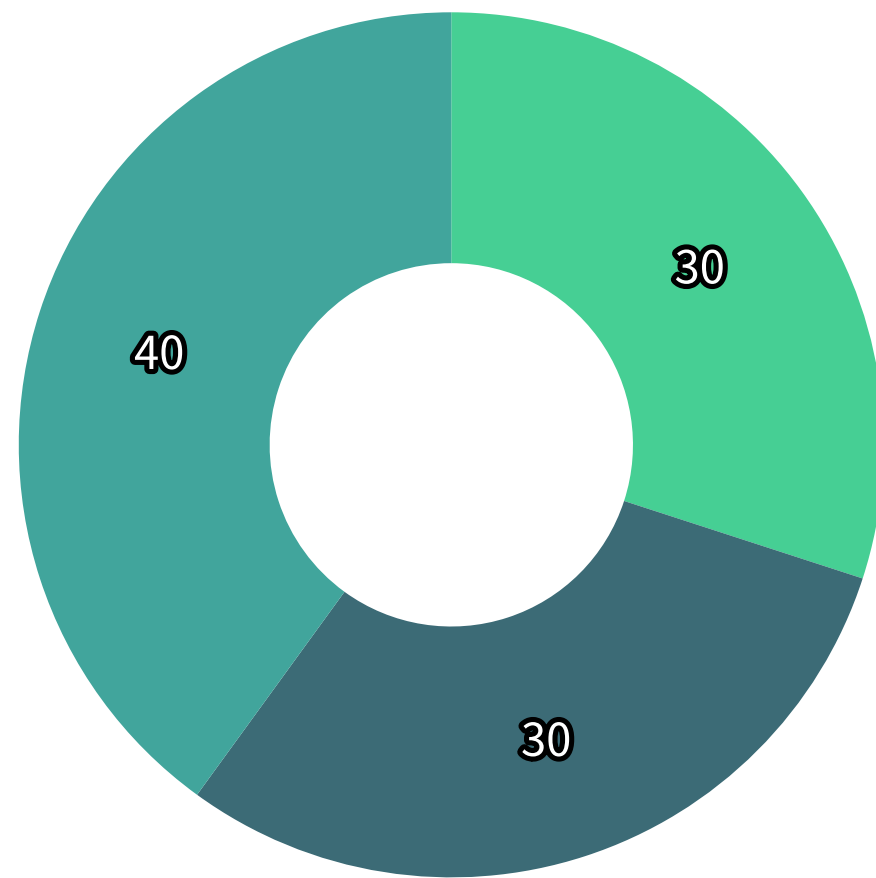
LGBM 40% Cat 40% DNN 40%

우리 csv 파일과 3등의 submission 파일이 가장 상관관계  
가 낮을 것이라고 생각되어, 위와 같이 가중치를 부여함

# 가중평균 & Mean Ensemble 3

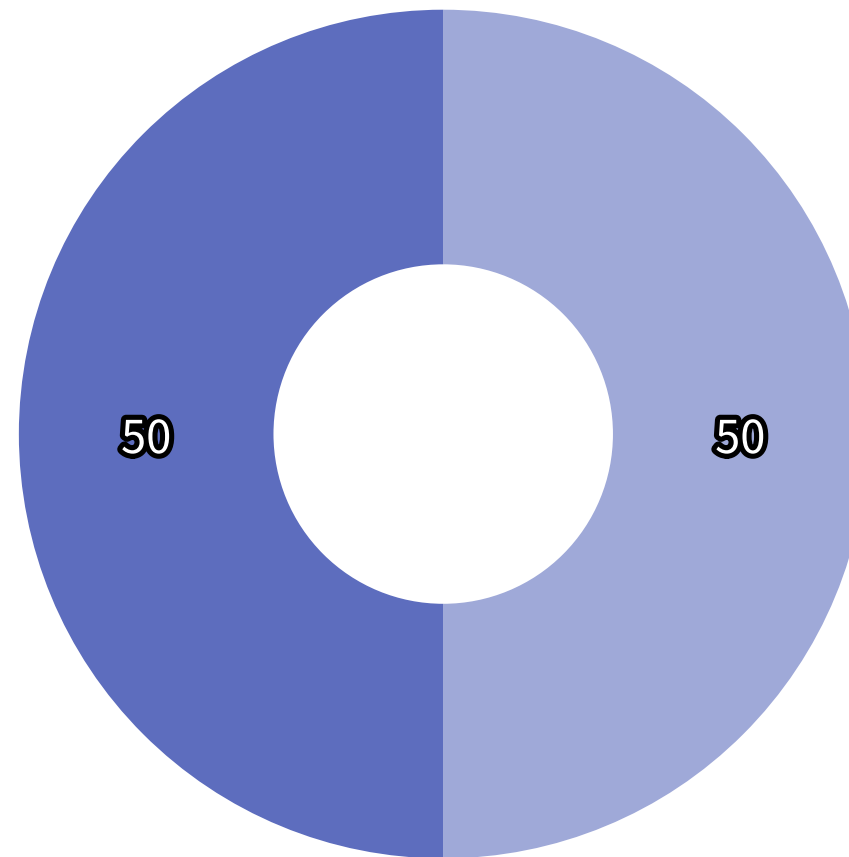


[sub\_123]

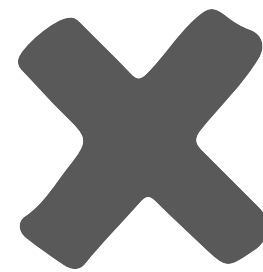


1등 30% 2등 30% 3등 40%

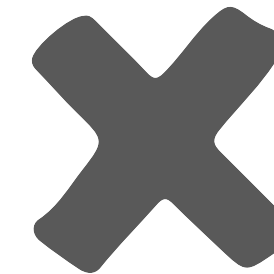
[우리모델\_2]



LGBM 50% Cat 50%



양상블

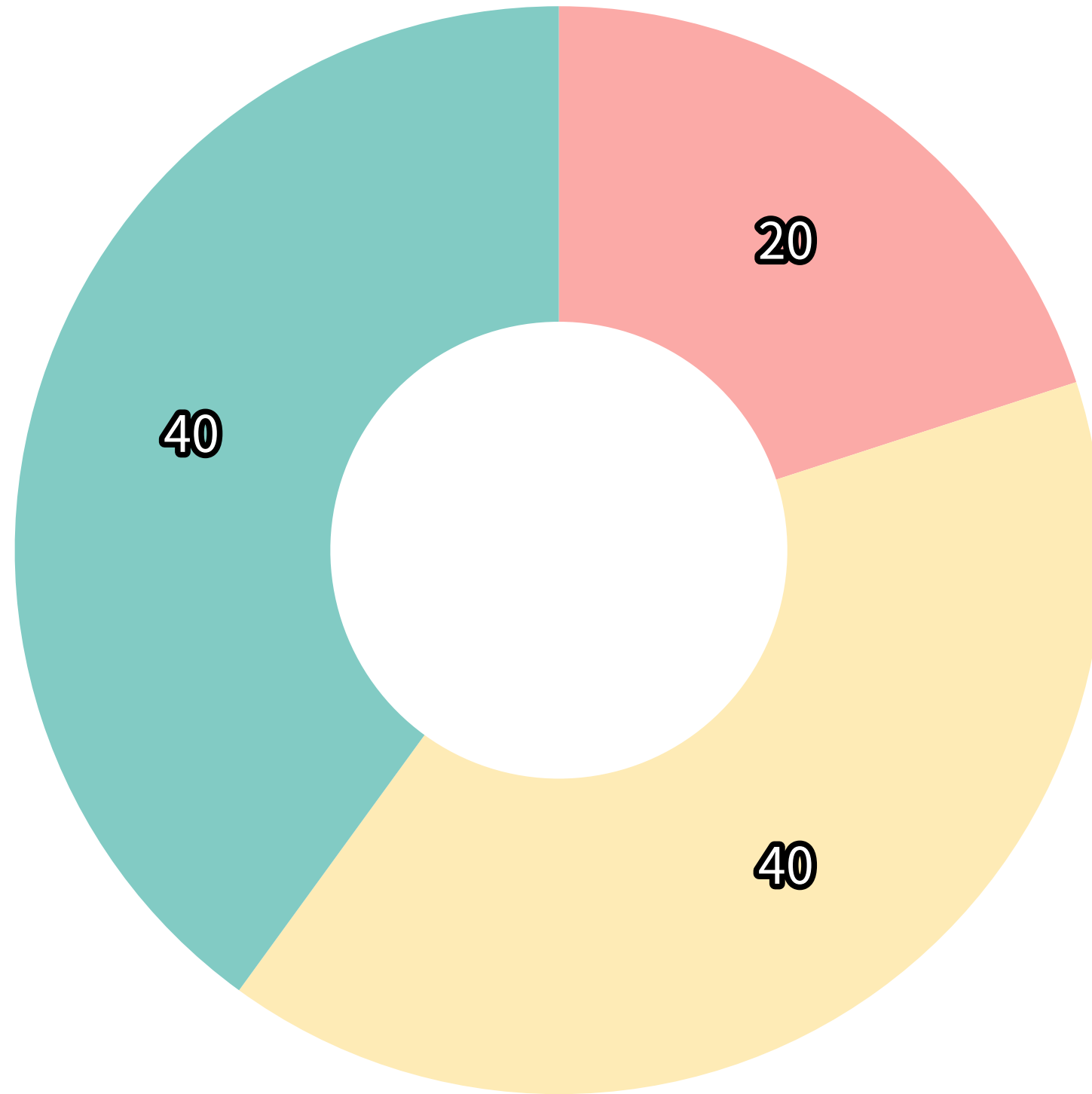


양상블

교수님이 올려주신  
DNN.csv

우리 모델로 만든 DNN 모델보다 교수님이 올려주신  
DNN의 성능이 더 좋아서 가중평균을 할 때,  
DNN을 제외한 두 모델의 평균을 구함

# 최종모델



- sub\_123 20%
- 우리모델\_2 40%
- 교수님이 올려주신 DNN 40%

# 아쉬운 점



## 예상치 못한 문제

각각 모델마다 선택을 해서 모델마다 fitting된 열의 수가 달라짐

→ 스택킹이나 가중평균의 비율을 알려주는 코드 실행 불가

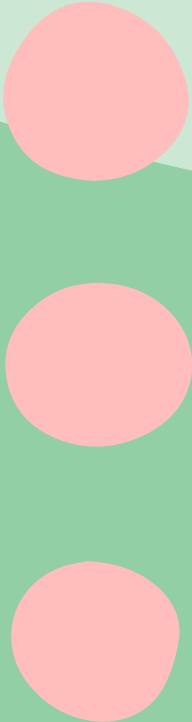
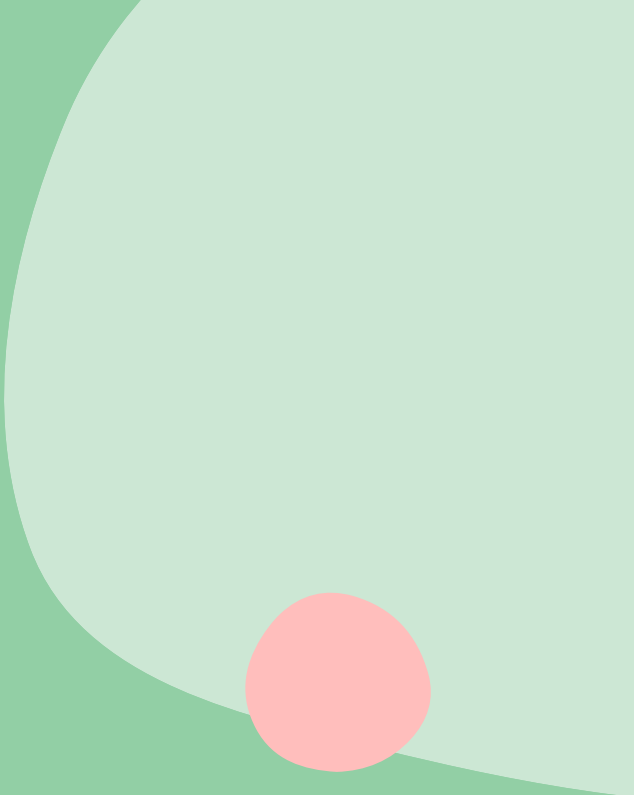

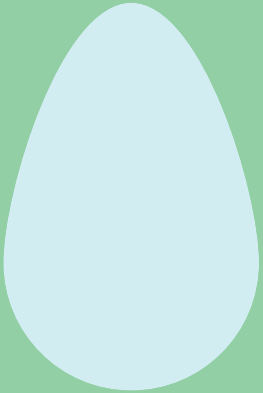
→ 임의로 가중평균하기로 결정

## 모델

feature 정규화 및 로그화, 선택성 진행 모델 변경, 파라미터 튜닝 등 dnn과 ridge의 성능을 올리기 위해 노력했지만 결국 원하던 성능을 이끌어내지 못함  
두 모델의 성능을 더 높였다면, 더 좋은 결과가 나왔을 것으로 예상됨

## 시간상의 제약

stacking을 시도했으나, 시간상의 문제로 중단하여 결과를 확인하지 못함  
feature가 너무 많아서 모델을 돌아가는 데 시간이 오래걸려 다양한 시도를 하지 못함



# 감사합니다

신예주 X 윤경서