

KUBIG 22-2 장기 프로젝트

# 분류/예측 1팀:

## 인간지능, 머니&게임

Finda 어플 사용자 대출신청 예측  
/ Kaggle 게임 플레이어 레이팅 예측

16기 김상옥 김진서 노연수 민윤기 이수찬 천원준

# 인간지능, 머니&게임 팀 소개

16기 6명으로 이루어진 팀으로, 특정 주제에 얽매이지 않고  
방학 중 ML 분반에서 익힌 파이프라인 & 라이브러리를  
심화 연습하는 것에 초점을 맞추어 진행했습니다.

2차례의 단기 프로젝트를 진행하며 단순 성능 향상보다는  
다양한 갈래로 진행해보고 결과를 비교하는 것에 주력했습니다.

1주차

분류/예측 팀 분할  
**팀 구성**

2주차 ~ 1차 특별세션

**1차 프로젝트:머니**

Finda 어플 사용자  
대출신청 예측

5주차 ~ 7주차

**2차 프로젝트:게임**

Kaggle 게임 플레이어  
레이팅 예측

# CONTENTS

# 목차

2022 빅콘테스트 데이터분석부문 퓨처스리그 예선 참가

## I. Finda 어플 사용자 대출신청 예측

01 Data Overview & Pipeline

02 Preprocessing & Modeling A ver.

03 Preprocessing & Modeling B ver.

Kaggle Competition: Scrabble Player Rating

## II. Kaggle 게임 플레이어 레이팅 예측

01 Data Overview & Pipeline

02 Preprocessing

03 Modeling

KUBIG 22-2 장기 프로젝트 분류/예측 1팀  
인간지능, 머니&게임

2022 빅콘테스트 데이터분석부문 퓨처스리그 예선 참가

# I. Finda 어플 사용자 대출신청 예측

I. Finda 어플 사용자 대출신청 예측

## 01 Data Overview & Pipeline

📁 2022 빅콘테스트 데이터분석부문 퓨처스리그 예선 참가



핀다 홈화면 진입 고객 중 특정기간 안에 **대출신청 고객 예측**

예측문제

- 대출신청 고객을 예측
- 예측 정확도 - F1 Score

$$F1\ Score = 2 \times \frac{recall \times precision}{recall + precision}$$

팀 인원 제한에 맞춰 분할 등록

KUBIG A :  
김진서 민윤기 천원준

KUBIG B:  
김상옥 노연수 이수찬

### 프로젝트 타임라인



2주차

9/14 회의: 주제 선정  
9/16 참가신청 완료



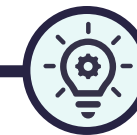
3주차

9/21 회의: EDA 결과 공유  
9/25 모델&전처리 자료조사



4주차

9/27 회의: 코딩 진행 논의  
9/30 사전점검 보고서 초안  
10/4 사전점검 보고서 제출



특별세션 주간

10/5 회의: 최종 진행 보고  
10/10 주요 모델링 마무리  
10/14 최종보고서 제출

I. Finda 어플 사용자 대출신청 예측

## 01 Data Overview & Pipeline

📁 데이터셋 요약



### 사용자 신용정보

user\_spec.csv

- 생년월일, 성별 등 개인사항
- 소득, 근로형태 등 직업 변수
- 기존 대출, 개인회생 등 신용 변수



### 대출 결과

loan\_result.csv

- 신청서 및 상품 id (Key)
- 신청 여부(Yes/No) (Target)
- 승인한도 및 금리 등 대출 정보



### 사용자 로그

log\_data.csv

- Finda 어플리케이션 사용 정보
- 앱 실행, 로그인, 신용정보 및 한도조회 등 활동 기록

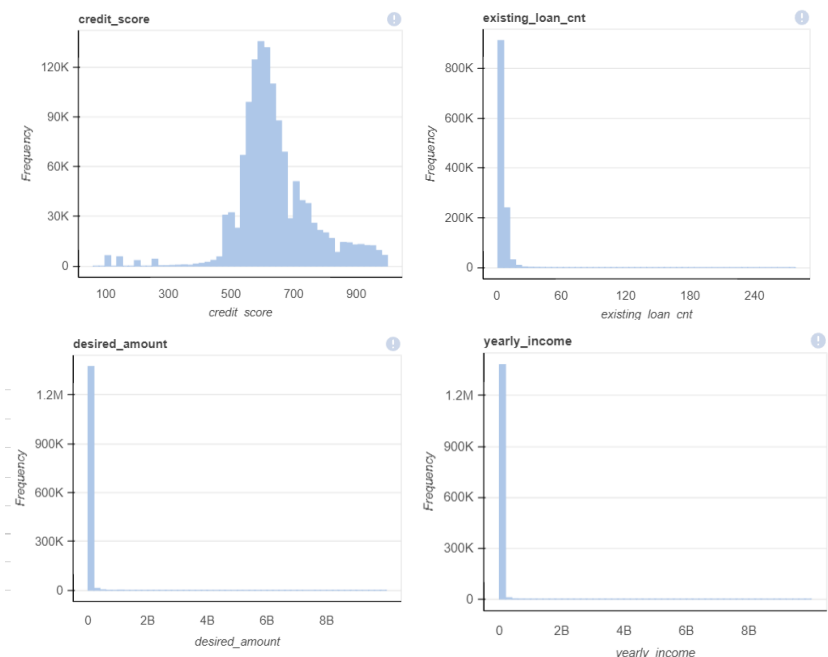
I. Finda 어플 사용자 대출신청 예측  
01 Data Overview & Pipeline

데이터셋 요약



사용자 신용정보  
user\_spec.csv

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 1394216 entries, 0 to 1394215  
Data columns (total 17 columns):  
#   Column                                     Non-Null Count  Dtype  
---  ---                                     -  
0   application_id                           1394216 non-null int64  
1   user_id                                  1394216 non-null int64  
2   birth_year                               1381255 non-null float64  
3   gender                                   1381255 non-null float64  
4   insert_time                              1394216 non-null object  
5   credit_score                             1289101 non-null float64  
6   yearly_income                           1394126 non-null float64  
7   income_type                             1394131 non-null object  
8   company_enter_month                     1222456 non-null float64  
9   employment_type                         1394131 non-null object  
10  houseown_type                           1394131 non-null object  
11  desired_amount                           1394131 non-null float64  
12  purpose                                  1394131 non-null object  
13  personal_rehabilitation_yn               806755 non-null float64  
14  personal_rehabilitation_complete_yn      190862 non-null float64  
15  existing_loan_cnt                       1195660 non-null float64  
16  existing_loan_amt                       1080442 non-null float64  
dtypes: float64(10), int64(2), object(5)  
memory usage: 180.8+ MB
```



- 140만개 가량의 row, 일부 열에서 10만개~100만개 정도의 결측치 존재
- 신용점수credit\_score 등 고르게 분포하는 변수도 있으나,
- 대출 잔액existing\_loan\_amt, 대출 희망액desired\_amount, 연수입yearly\_income 등 주로 금액 관련 변수에서 극단적 이상치 존재

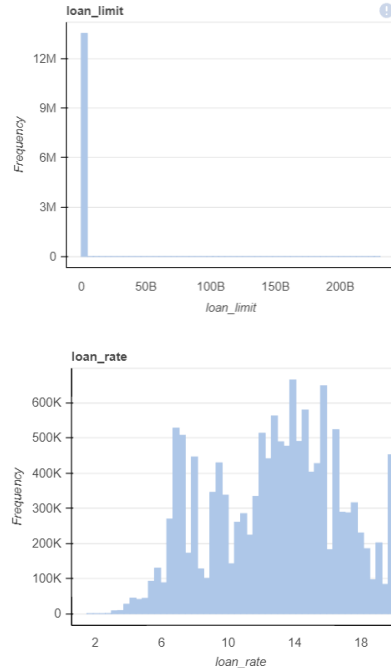
## I. Finda 어플 사용자 대출신청 예측

# 01 Data Overview & Pipeline

### 📁 데이터셋 요약



## 대출 결과 loan\_result.csv



- 타겟 변수 is\_applied와 대출 정보 loan\_limit, loan\_rate
- 마찬가지로 금액 변수의 극단값 확인



## 사용자 로그 log\_data.csv

	event	timestamp	mp_os
count	17843993	17843993	17843013
unique	11	6879764	4
top	OpenApp	2022-04-11 11:40:30	Android
freq	3460762	23	12331688

	mp_app_version	date_cd
count	17183396	17843993
unique	259	122
top	3.14.0	2022-06-27
freq	2339899	267738

- Finda 앱에서 유저 사용 기록 및 시간, 기기 OS와 앱 버전 등
- 1,700만개 가량의 row가 있는 대형 데이터



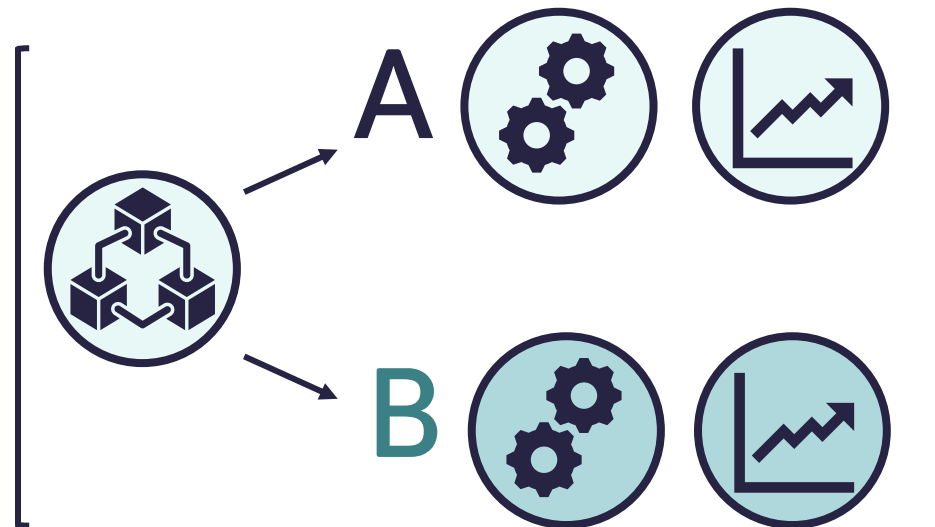
## I. Finda 어플 사용자 대출신청 예측

# 01 Data Overview & Pipeline

### 📁 파이프라인 요약

#### 공통 데이터 세팅

데이터 병합 및  
Train / Test 분리  
log\_data 데이터 추출  
일부 결측치 처리



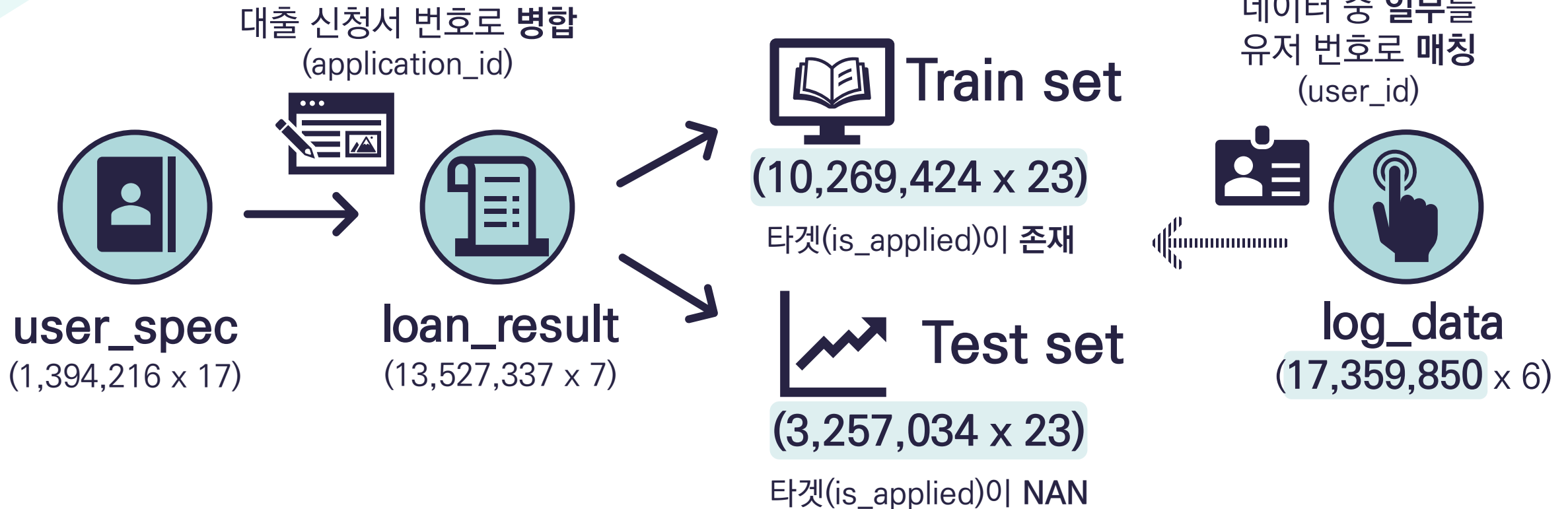
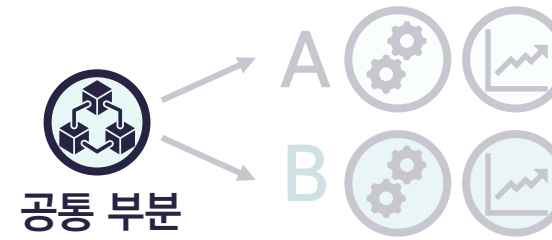
#### 세부적으로 다른 접근 시도

결측치 대치, 분포 변환 등  
모델링 및 최종 예측

I. Finda 어플 사용자 대출신청 예측

## 01 Data Overview & Pipeline

📁 파이프라인 요약: Train / Test 구성



공통적으로 데이터의 크기 Train set 1.89GB, Test set 607MB뿐 아니라,  
단위와 사용자가 다른 log\_data를 Train/Test에 맞게  
적절히 변환하는 것이 가장 큰 문제

# I. Finda 어플 사용자 대출신청 예측

## 01 Data Overview & Pipeline

📁 파이프라인 요약: log\_data 추출



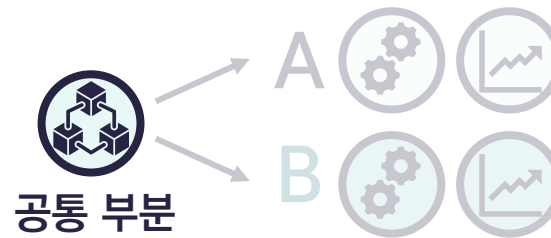
log\_data

(17,359,850 x 6)



user\_spec  
(1,394,216 x 17)

로그 데이터 특성 상 대부분의 경우  
복수의 행이 한개의 user\_id에 대응하므로  
user\_id에 일대일 대응하지 않음,  
단순 병합 불가



KUBIG 22-2 장기 프로젝트 분류/예측 1팀  
인공지능, 머니&게임

	user_id	event	timestamp	mp_os	mp_app_version	date_cd
0	576409	StartLoanApply	2022-03-25 11:12:09	Android	3.8.2	2022-03-25
1	576409	ViewLoanApplyIntro	2022-03-25 11:12:09	Android	3.8.2	2022-03-25
2	72878	EndLoanApply	2022-03-25 11:14:44	Android	3.8.4	2022-03-25
3	645317	OpenApp	2022-03-25 11:15:09	iOS	3.6.1	2022-03-25
4	645317	UseLoanManage	2022-03-25 11:15:11	iOS	3.6.1	2022-03-25
5	640185	UseLoanManage	2022-03-25 11:41:53	iOS	3.6.1	2022-03-25
6	640185	ViewLoanApplyIntro	2022-03-25 11:42:38	iOS	3.6.1	2022-03-25
7	640185	UsePrepayCalc	2022-03-25 11:43:07	iOS	3.6.1	2022-03-25
8	640185	UseLoanManage	2022-03-25 11:43:57	iOS	3.6.1	2022-03-25
9	640185	UseLoanManage	2022-03-25 11:44:04	iOS	3.6.1	2022-03-25

# I. Finda 어플 사용자 대출신청 예측

## 01 Data Overview & Pipeline

📁 파이프라인 요약: 로그 데이터 추출

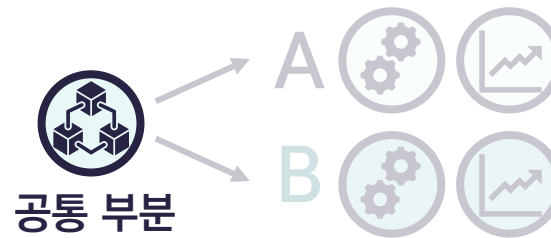
대안으로 user\_id별 event 실행 여부 추출

```
[ ] for i in list(log_data['event'].unique()):
    print(i)
    log_limit = log_data[log_data['event']==i]
    log_limit = log_limit[['user_id']]
    log_limit[i] = 1
    log_limit = log_limit.drop_duplicates(ignore_index = True)
    log_limit = log_limit.drop(log_limit[~log_limit['user_id'].isin(train_user['user_id'])].index)
    print(log_limit.shape)
    train_user = train_user.merge(log_limit, on='user_id', how='left')
    train_user = train_user.fillna(0)
    train_loan5[i] = train_user[i]
    train_user = train_loan5[['user_id']]
```

StartLoanApply  
/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:5: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row\_indexer,col\_indexer] = value instead

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/10min.html](https://pandas.pydata.org/pandas-docs/stable/user_guide/10min.html)

```
(205694, 2)
ViewLoanApplyIntro
(208407, 2)
EndLoanApply
(211758, 2)
OpenApp
(199218, 2)
UseLoanManage
(159545, 2)
UsePrepayCalc
(3098, 2)
Login
(169612, 2)
CompleteIDCertification
(203151, 2)
UseDSRCalc
(2373, 2)
SignUp
(12463, 2)
GetCreditInfo
(205325, 2)
```



공통 부분

```
cor3 = train_loan5[list(log_data['event'].unique())].corrwith(other = train_loan5['is_applied'])
cor3 = cor3.reset_index()
cor3.sort_values(0, ascending=False)
```

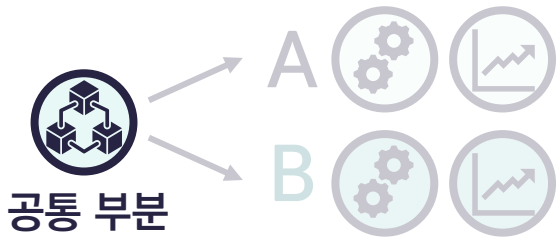
	index	0
0	StartLoanApply	0.041596
2	EndLoanApply	0.039095
3	OpenApp	0.036774
4	UseLoanManage	0.035475
1	ViewLoanApplyIntro	0.025215
9	SignUp	0.024246
7	CompleteIDCertification	0.024220
10	GetCreditInfo	0.023731
6	Login	0.020620
5	UsePrepayCalc	0.000309
8	UseDSRCalc	-0.007152

타겟과의 상관관계수 체크, 타겟과의 상관관계수가 비교적 큰  
**대출 신청 완료 여부** EndLoanApply,  
그 다음으로 유의하면서 상호 상관성이 과하지 않은  
**대출관리 서비스 이용 여부** UseLoanManage를 최종 활용

# I. Finda 어플 사용자 대출신청 예측

## 01 Data Overview & Pipeline

 파이프라인 요약: 일부 결측치 대처



결측치가 있는 행 중에서  
학습용 데이터셋 Train에만  
결측치가 있는 경우,  
비율이 매우 낮으므로 결측행 제거

```
[ ] #train에만 있는 결측치 비율
pd.options.display.float_format = '{:,.6f}'.format
train_loan[na_train].isnull().sum()/len(train_loan)

insert_time      0.000011
houseown_type    0.000011
employment_type  0.000011
purpose          0.000011
income_type      0.000011
user_id          0.000011
desired_amount   0.000011
dtype: float64
```

개인회생 관련 변수 rehabilitation\_complete, rehabilitation\_incomplete 더미화,  
결측치의 경우 기록 없음으로 간주해 0으로 처리

```
#trian 데이터에서 'rehabilitation_complete','rehabilitation_incomplete' 변수 고유값 확인
train_loan2[['rehabilitation_complete','rehabilitation_incomplete']].value_counts()

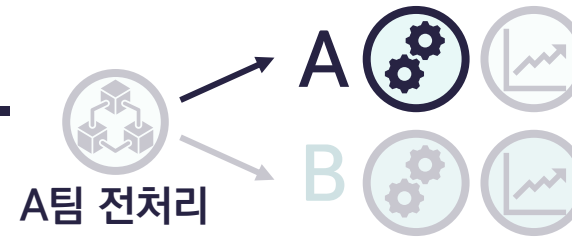
rehabilitation_complete  rehabilitation_incomplete
0                        0
                        1
1                        0
dtype: int64
```

9226739	← 개인회생 해당 없음
1033045	← 개인회생 진행 중
4110	← 개인회생 완료

# I. Finda 어플 사용자 대출신청 예측

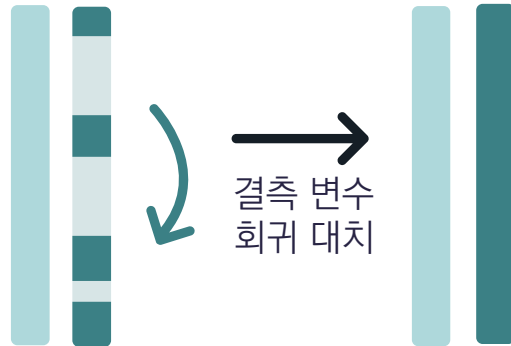
## 02 Preprocessing & Modeling A ver.

### 📁 A팀 전처리



### 이외 결측치 처리

IterativeImputer 사용해서  
수치형 변수 결측치 대체



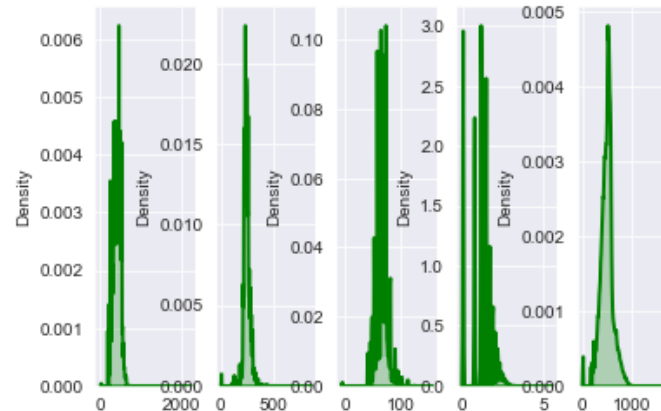
해당 변수: birth\_year, credit\_score,  
existing\_loan\_cnt, existing\_loan\_amt

범주형은 최빈값으로 대체  
gender, company\_enter\_month

### 이상치 완화&분포 변환

#### box-cox 변환

해당 변수: loan\_limit, yearly\_income,  
desired\_amount, existing\_loan\_cnt,  
existing\_loan\_amt



Train에서 매개변수 fitting 후  
Test에도 적용해 변환

### 범주형 데이터 변환 및 타겟변수 처리 Label-encoder 사용

```
from sklearn.preprocessing import LabelEncoder
encoder=LabelEncoder()
label=list(train_loan2.select_dtypes(include = 'object').columns)
oneset=[train_loan2, test_loan1]
```

```
for data in oneset:
    for i in label:
        data[i] = encoder.fit_transform(np.array(data[i]))
```

```
test_loan1[label].head()
```

	loanapply_insert_time	insert_time	income_type	employment_type	houseown_type	purpose
0	146652	38913	0	3	2	1
1	146652	38913	0	3	2	1
2	146652	38913	0	3	2	1
3	146652	38913	0	3	2	1
4	146652	38913	0	3	2	1

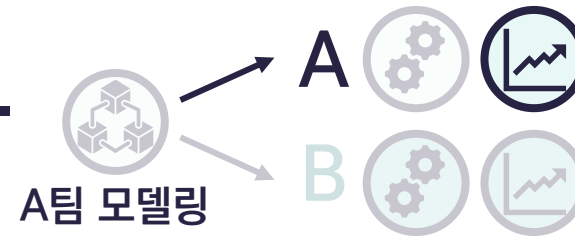
### 또한 smote로 타겟변수 균형화

SMOTE 적용 학습용 피쳐/레이블 데이터 세트: (19432836, 24) (19432836,)  
SMOTE 적용 후 레이블 값 분포:  
1.0 9716418  
0.0 9716418  
Name: is\_applied, dtype: int64

I. Finda 어플 사용자 대출신청 예측

## 02 Preprocessing & Modeling A ver.

### 📁 A팀 모델링: 검토한 기초 모형 및 조합



- **Logistic Regression**: 대표적인 선형 회귀 기반 분류 모델
- **Random Forest**: 신용평가에서 성능이 좋은 편 + 클래스 불균형 문제에도 잘 대응하는 것으로 알려짐, 또한 큰 데이터셋에서도 잘 작동 → **메인 모델로 사용 검토**
- **Decision Tree**: 특정 기준에 따라 데이터를 구분하는 모델. 고차원 대형 데이터셋에 강하다는 장점, Random Forest의 기반
- **Naive Bayes**: 빠르고 정확한 모델이나 모든 Feature가 독립이어야 한다는 한계
- **LightGBM**: 균형 트리 분할 기반 모델로 과적합에 강하지만 균형 잡힌 트리를 만들기 위한 소요시간 큼

### Soft Voting A

Random Forest, Decision Tree, Logistic Regression

- Random Forest로 안정적인 성능 확보
- 보조적으로 Decision Tree를 통해 고차원 데이터에 대한 적응성을 향상,
- Logistic Regression으로 overfitting 방지
- 실행 시간&메모리 부담 크나 성능 향상

### Soft Voting B

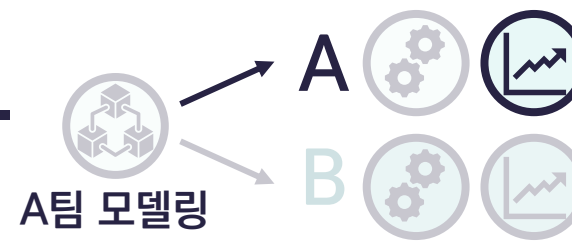
Random Forest, Decision Tree, Logistic Regression, Naïve Bayes

- A안에 나이브 베이즈 추가, 모델 4개 사용하는 대신 하이퍼 파라미터를 축소해 적용
- A안에 비해 실행 시간&메모리 부담 크나 성능 다소 하락

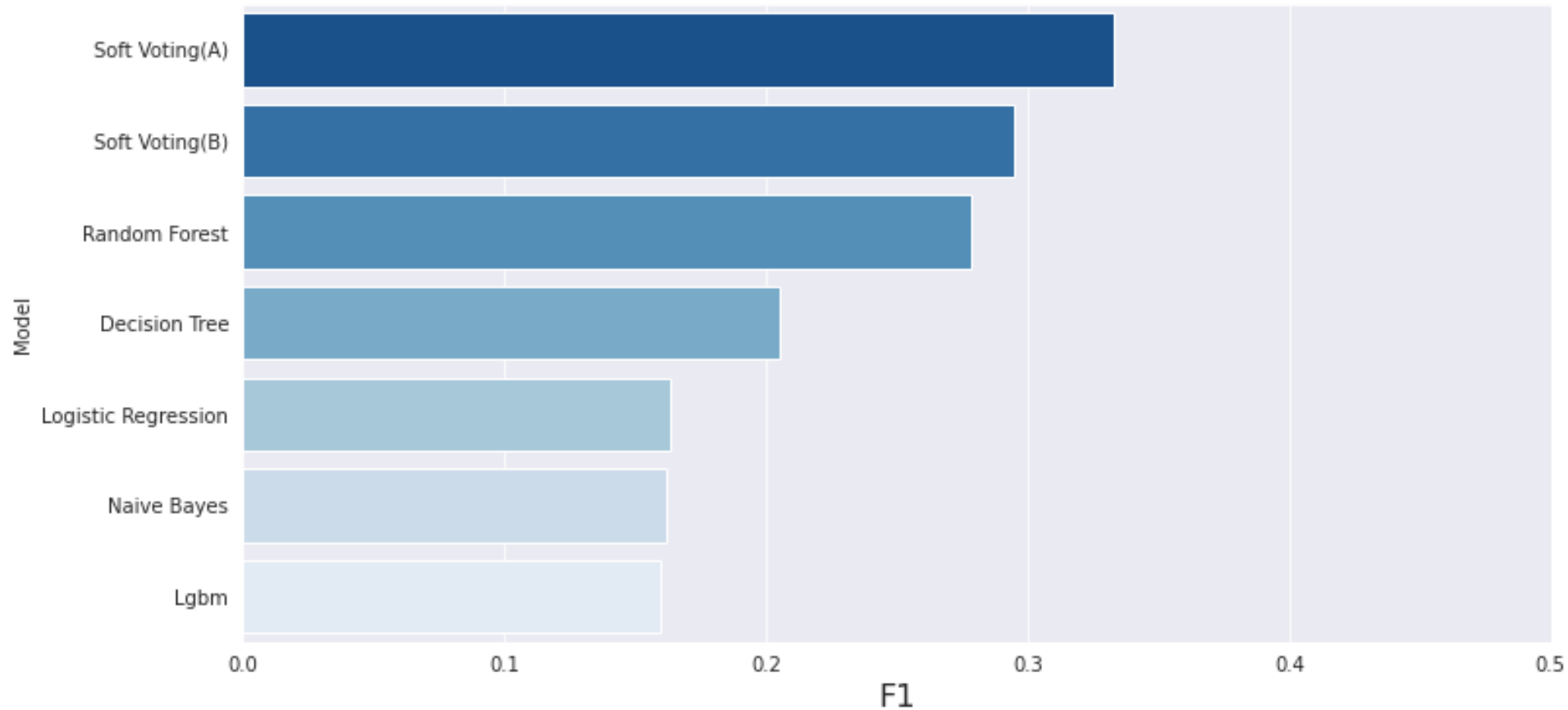
I. Finda 어플 사용자 대출신청 예측

## 02 Preprocessing & Modeling A ver.

📁 A팀 모델링 : 검토한 기초 모형 및 조합



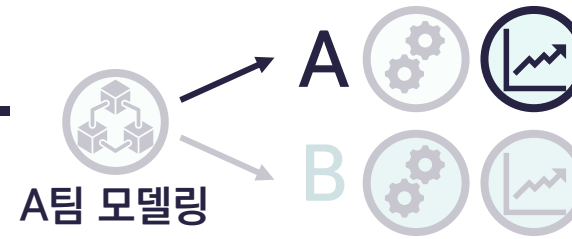
Train 셋 내부에서 모델 성능 확인 (Train : Test = 9 : 1)



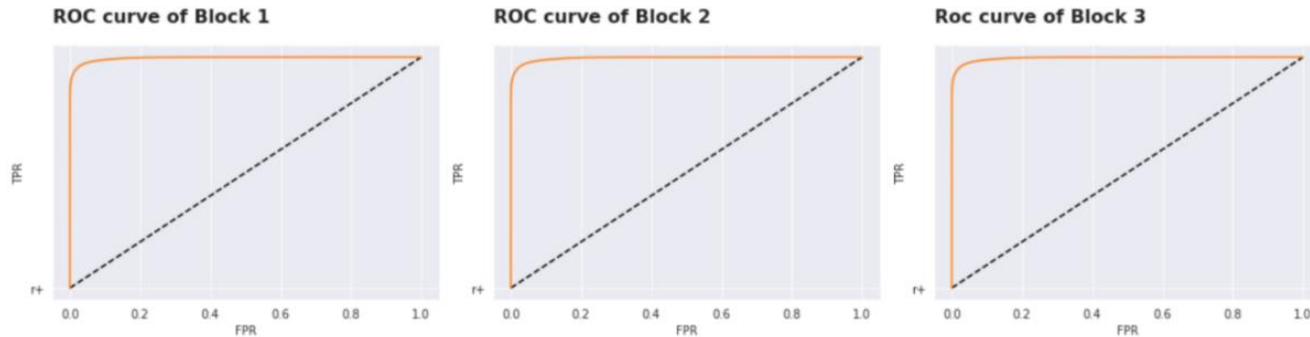


## 02 Preprocessing & Modeling A ver.

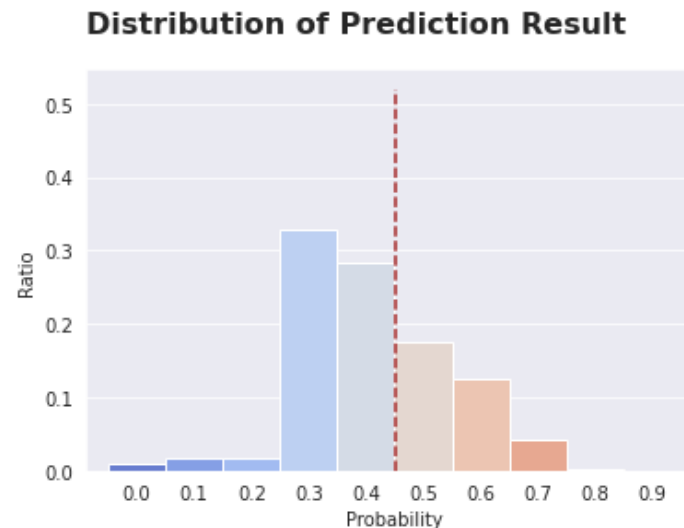
### 📁 A팀 모델링: K-fold Stacking



**K-fold**(fold=3) 사용해 최종 모델 도출 & 예측, Soft Voting A에서 파라미터 일부 축소  
랜덤 포레스트 estimators=15, 결정트리 max\_depth=10



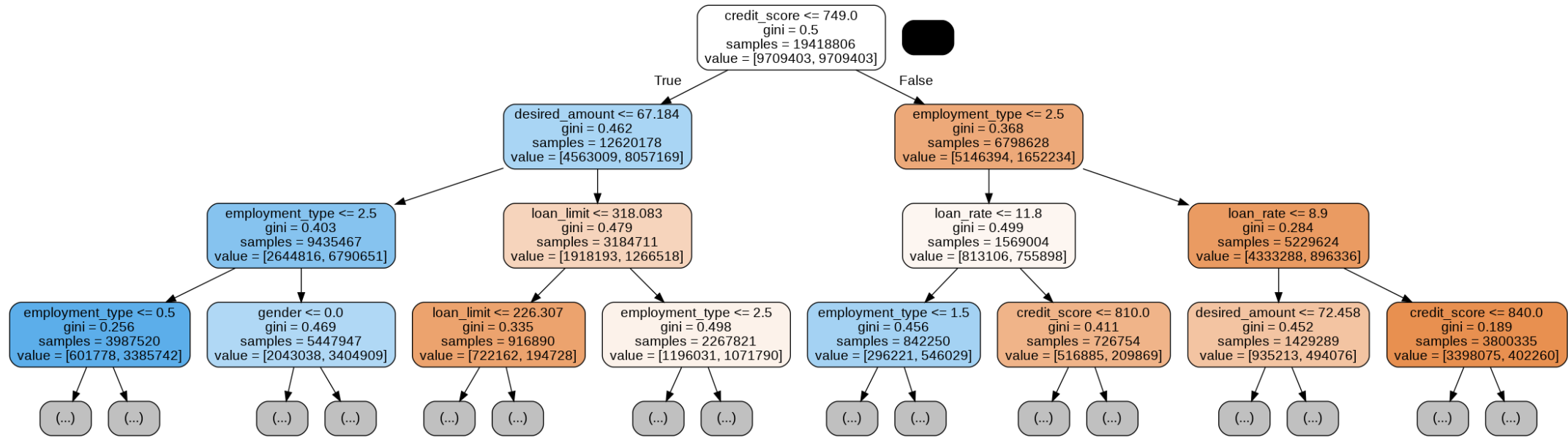
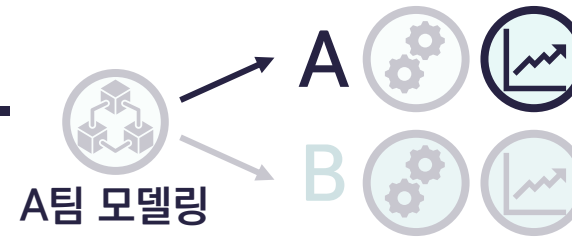
최종 Test 예측 확률 분포  
뚜렷하게 갈라지지 않는 경향



## 02 Preprocessing & Modeling A ver.

### A팀 모델링: 결과 시각화

### 결정 트리 상위 노드 시각화

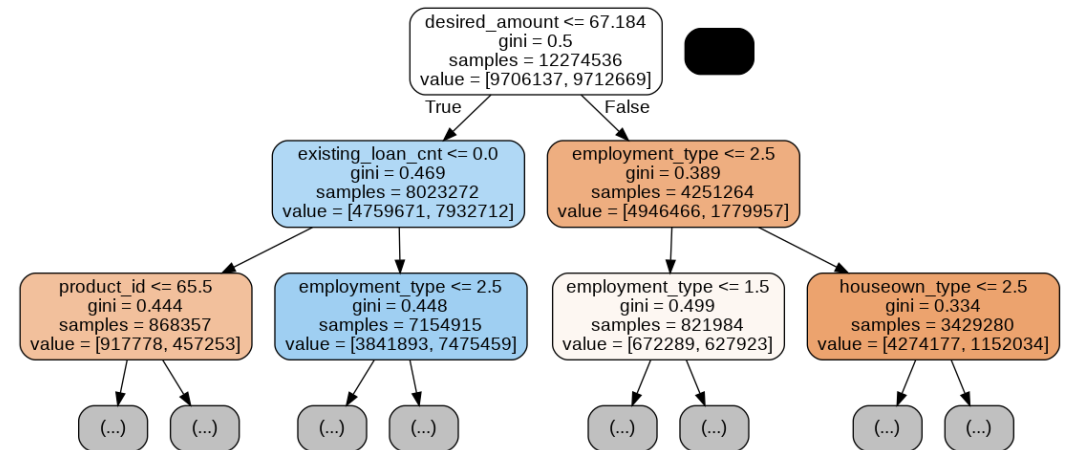
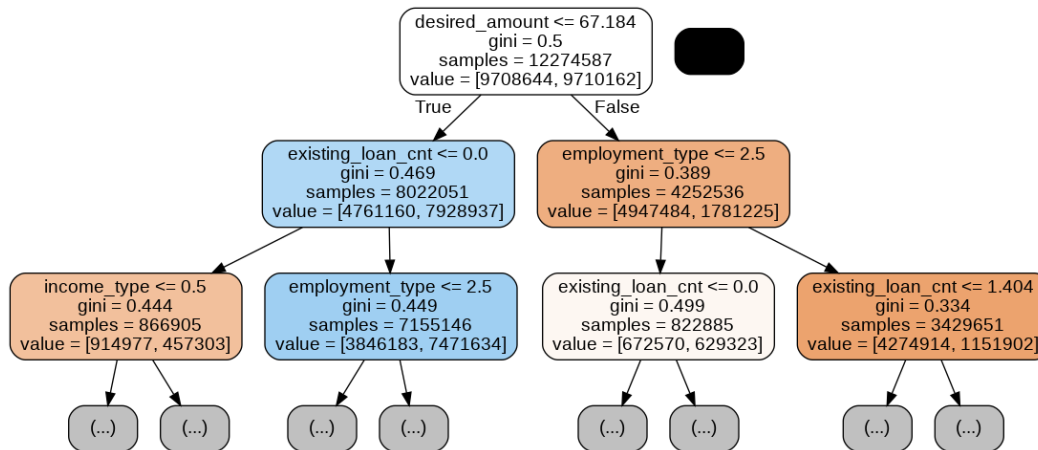
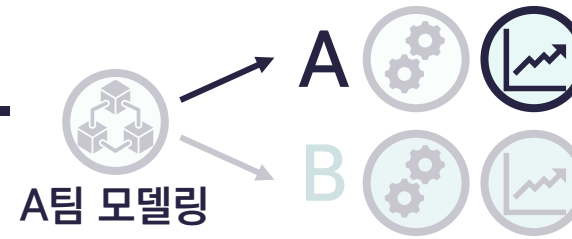


- 고차원&대형 데이터 특성 상 설정한 깊이 최대치인 10까지 모두 사용
- 상위 4단계를 확인한 결과 변수 중 **credit\_score**, **desired\_amount**, **employment\_type**, **loan** 관련 변수가 가장 결정적으로 작용하고 있는 것으로 파악됨

## 02 Preprocessing & Modeling A ver.

### A팀 모델링: 결과 시각화

#### 랜덤 포레스트 일부 상위 노드 시각화

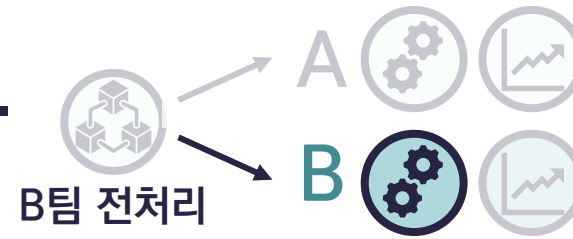


- 단일 결정 트리와 마찬가지로 `desired_amount`, `employment_type`, `loan` 관련 변수가 가장 결정적으로 작용하고 있는 것으로 파악됨
- 최상위 조건은 estimator 간 동일하나 세부 변수에서 - 3단계 노드부터 차이 발생

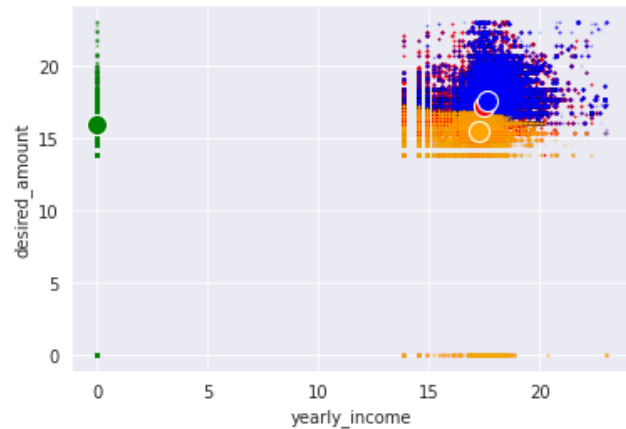
# I. Finda 어플 사용자 대출신청 예측

## 03 Preprocessing & Modeling B ver.

### 📁 B팀 전처리



### K-Means 클러스터링



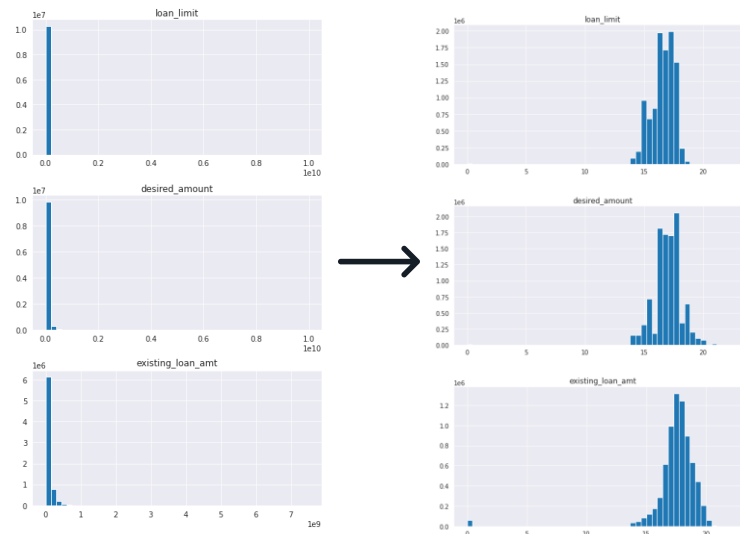
근거 변수: yearly\_income,  
desired\_amount, houseown\_type\_자가,  
houseown\_type\_기타가족소유, loan\_limit

타겟과 상관관계수 높은 5개 근거  
변수로 K-Means 클러스터링,  
군집별 중앙값으로 결측치 대체

### 이상치 완화&분포 변환

$\ln(1 + p)$ 로 로그변환

해당 변수: loan\_limit, yearly\_income,  
desired\_amount, existing\_loan\_cnt,  
existing\_loan\_amt



### 범주형 데이터 변환 및 타겟변수 처리

One-hot encoding 사용,  
그러나 Feature 수가 과하게 증가

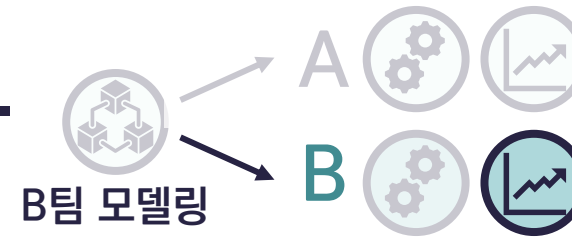
application\_id  
loanapply\_insert\_time  
bank\_id  
product\_id  
loan\_limit  
loan\_rate  
is\_applied  
user\_id  
birth\_year  
gender  
insert\_time  
credit\_score  
yearly\_income  
company\_enter\_month  
desired\_amount  
existing\_loan\_cnt  
existing\_loan\_amt  
rehabilitation\_complete  
rehabilitation\_incomplete  
income\_type\_EARNEDINCOME2  
income\_type\_FREELANCER  
income\_type\_OTHERINCOME  
income\_type\_PRACTITIONER  
income\_type\_PRIVATEBUSINESS  
employment\_type\_계약직  
employment\_type\_기타  
employment\_type\_일용직  
employment\_type\_정규직  
houseown\_type\_기타가족소유  
houseown\_type\_배우자  
houseown\_type\_자가  
houseown\_type\_전월세  
purpose\_BUSINESS  
purpose\_BUYCAR  
purpose\_BUYHOUSE  
purpose\_ETC  
purpose\_HOUSEDEPOSIT  
purpose\_INVEST  
purpose\_LIVING  
purpose\_SWITCHLOAN  
purpose\_기타  
purpose\_대환대출  
purpose\_사업자금  
purpose\_생활비  
purpose\_자동차구입  
purpose\_전월세보증금  
purpose\_주택구입  
purpose\_투자

차원의 저주 방지 위해  
타겟과의 상관관계수  
0.03 미만 필터링,  
19개 변수 제거

I. Finda 어플 사용자 대출신청 예측

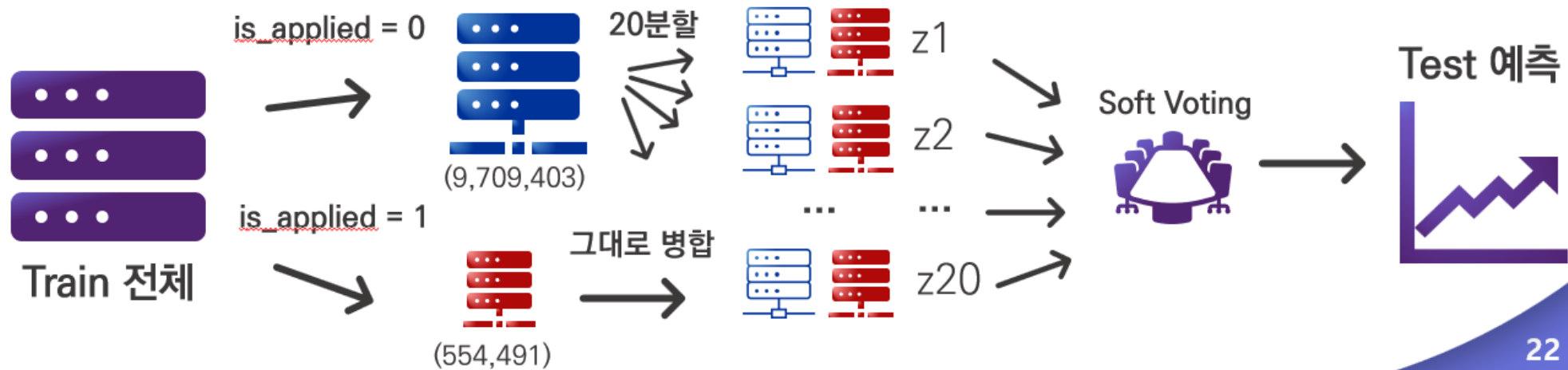
## 03 Preprocessing & Modeling B ver.

### 📁 B팀 모델링



### F1 score 최적화 기능에 초점을 맞춰 AutoML 적용 시도

하드웨어 문제로 데이터를 분할해서 학습시킨 다음 각 모델의 예측 결과(확률)를 Soft Voting 불균형 완화 위해 타겟이 No인 데이터 20분할, Yes인 경우를 각 분할에 그대로 병합



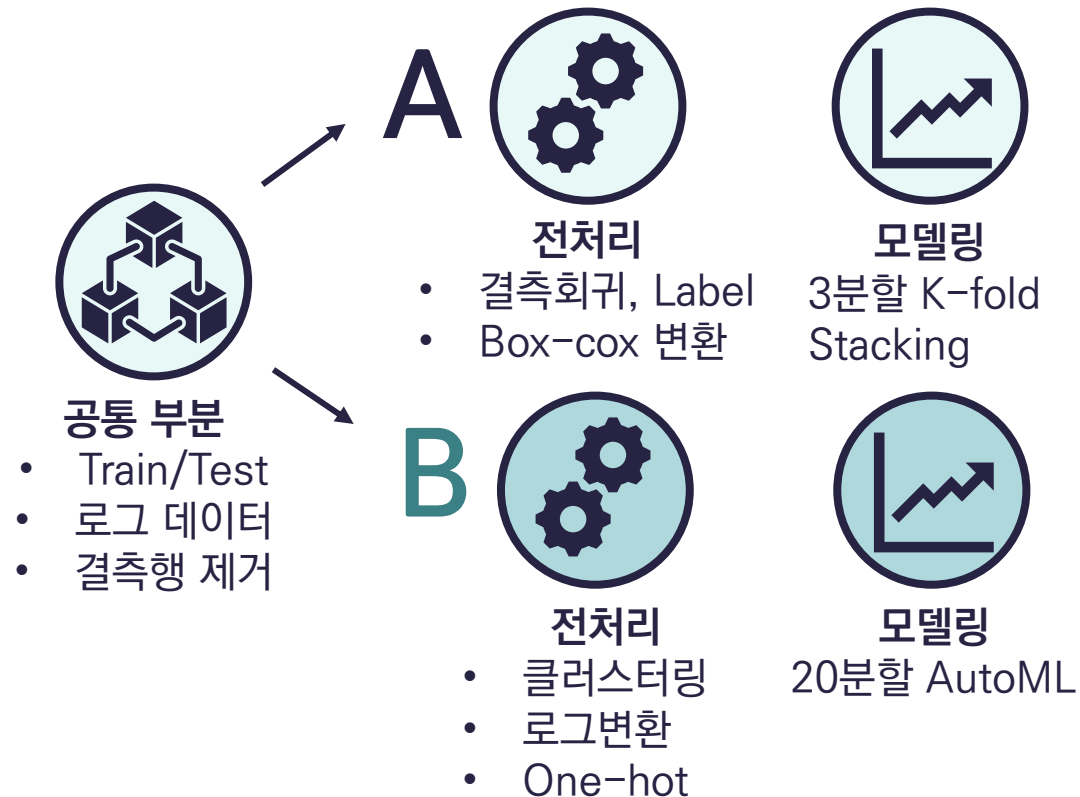
22  
page

(제출한 최종보고서 일부)

그러나 대출신청한 케이스  $is\_applied = 1$ 를 과다 학습했다는 점,  
개별 분할 관점에서 학습량 < 예측량이라는 점으로 인해 예측 결과는 적절하지 않았음

## I. Finda 어플 사용자 대출신청 예측 Discussion

### 📁 요약 및 고찰



- loan\_result의 1,000만개 타겟을 학습하고 300만개를 예측하는 **대형 문제**
- application\_id, user\_id를 키로 매칭, 키 중복으로 인한 거대화 문제를 완화
- 그러나 보다 근본적인 효율화 방법이 필요했음
- 시간과 경험의 부족 및 데이터 전처리의 어려움으로 **도메인 지식 적용이나 모델링에 투자할 여유를 확보하지 못한 점**이 한계

KUBIG 22-2 장기 프로젝트 분류/예측 1팀  
인공지능, 머니&게임

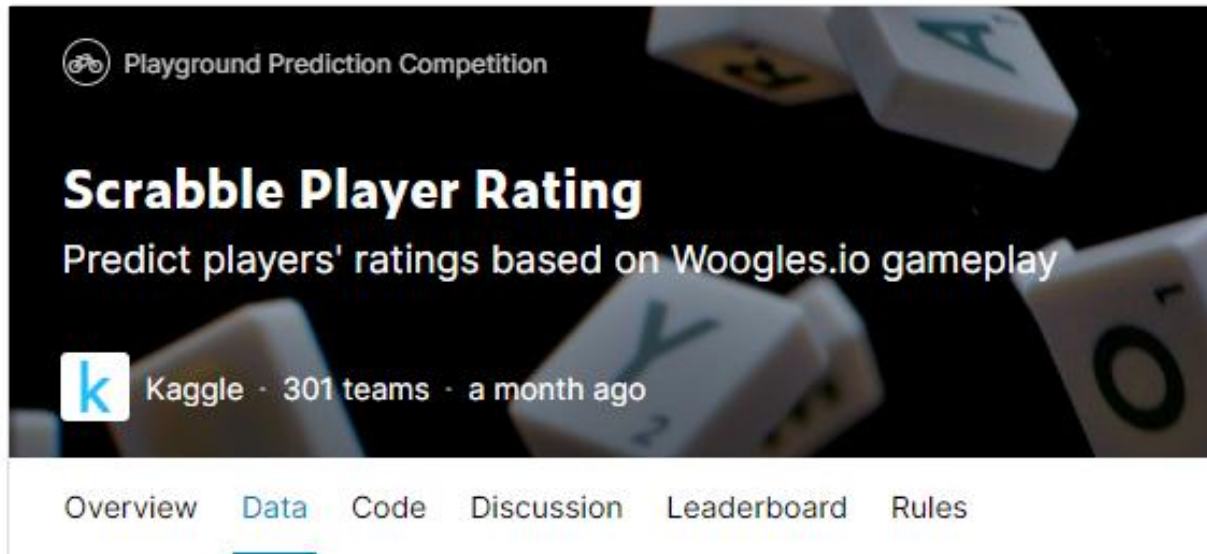
Kaggle Competition: Scrabble Player Rating

## II. Kaggle 게임 플레이어 레이팅 예측

## II. Kaggle 게임 플레이어 레이팅 예측

# 01 Data Overview & Pipeline

### Kaggle Competition: Scrabble Player Rating



**Problem:** Predict the ratings of players based on Scrabble gameplay

**Evaluation:** RMSE

**상대 Bot 종류:** BetterBot (beginner), STEEBot (intermediate), and HastyBot (advanced)

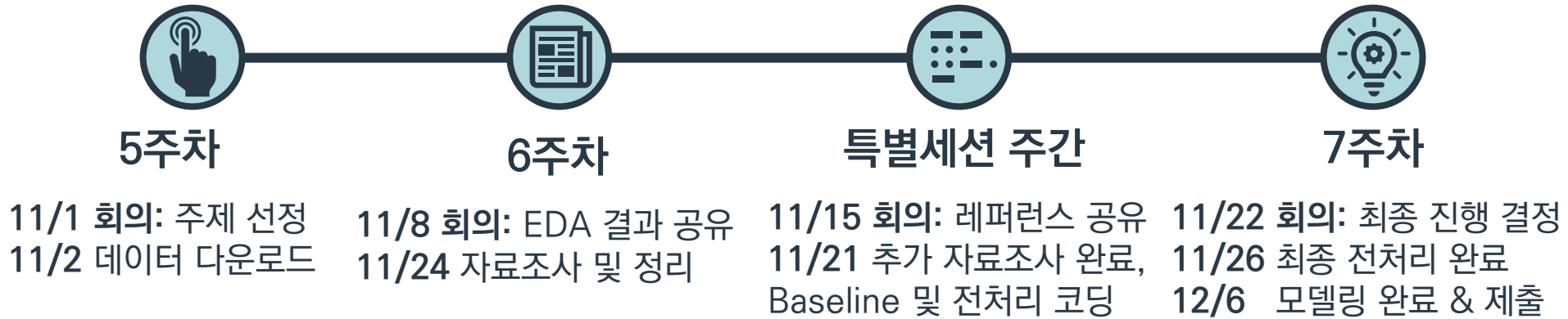
**취지:** 컴퓨터(Bot) 상대의 Scrabble 게임에서 지금까지의 플레이 데이터를 바탕으로 새로운 유저의 플레이를 관전한다면 **게임 점수를 넘어 그 플레이어의 실력**을 가늠할 수 있는가?



## II. Kaggle 게임 플레이어 레이팅 예측

# 01 Data Overview & Pipeline

### 📁 프로젝트 타임라인 & 데이터셋



### 데이터셋 파일 구성



**games.csv:** 게임 세부 사항 데이터  
12 x 72,773 game\_id, first, game\_end\_reason,  
winner, game\_duration\_seconds, lexicon 등



**turns.csv:** 턴 진행 데이터  
9 x 2,005,498 game\_id, turn\_number, rack, location,  
move, score, turn\_type 등



**train.csv:** 게임 결과 학습 데이터  
4 x 100,820 game\_id, nickname, score, rating



**test.csv:** 게임 결과 테스트 데이터  
4 x 44,726 game\_id, nickname, score, rating



**sample\_submission.csv:** 제출 형식  
2 x 22,363, game\_id, rating

## II. Kaggle 게임 플레이어 레이팅 예측

# 01 Data Overview & Pipeline

### 📁 Scrabble 게임 이해: Woogles.io

데이터 출처: 온라인 Scrabble 게임 사이트, Woogles.io

## 게임 규칙

1. 양측이 번갈아 게임판에 단어를 채우며, 이미 배치되어 있는 알파벳과 인접하게 배치해야 함
2. 전체 게임 시간 20분 내에 사전에 있는 모든 알파벳이 소진되면 게임이 종료되고 점수가 높은 사람이 승리
3. 전체 시간을 초과하면 **overtime**으로 진입 가능, 이때 추가 시간 (1분, 5분 등) 소진 시 점수 무관 패배 처리
4. 알파벳에 표기된 숫자는 점수, 추가 점수가 부여되는 (2배, 3배) **특정 지점에** 단어를 완성하면 고득점
5. 추가 옵션 1. **pass**: 차례 건너뛰고 상대방에게 턴 넘김  
2. **exchange**: 자신의 알파벳 세트에서 원하는 알파벳을 버리고 새로운 알파벳으로 교환

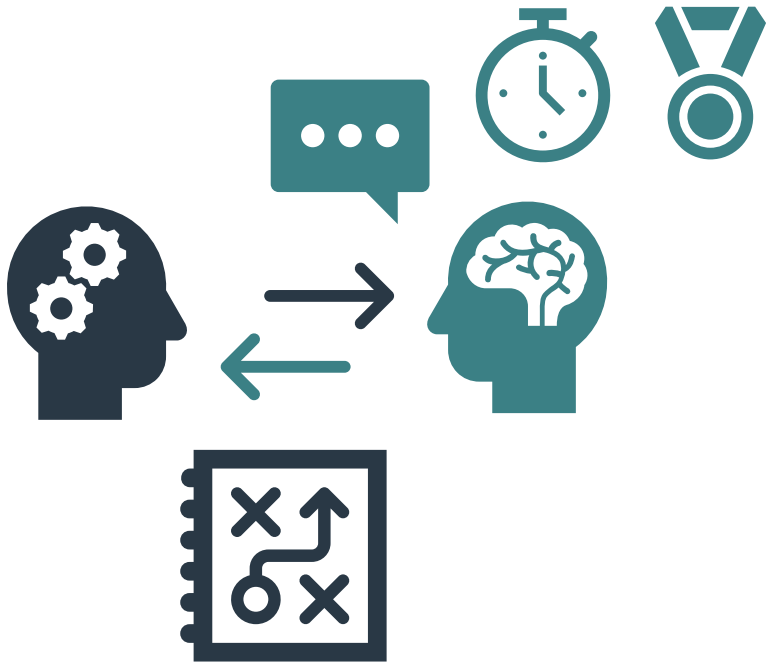


Woogles.io 실제 플레이 화면

## II. Kaggle 게임 플레이어 레이팅 예측

# 01 Data Overview & Pipeline

📁 Scrabble 게임 이해: Woogles.io



### 도메인: 실력 유관 요소 추정 & Bot에 대한 분석(1)

- Bot에게는 시간 제한이 의미 없음 – 체감상 Bot은 시간 지연 없이 최선의 수 선택, 따라서 시간이 많이 주어질수록 플레이어의 점수는 승패 무관 높아지는 경향
- 승리한 게임 시간이 길수록 더 높은 추가 레이팅 점수 획득 – overtime 규칙 적용 여부가 플레이어 입장에서 시간적으로 유불리에 영향 가능성
- 대개 게임 종료는 regular이며 타임아웃, 기권은 흔치 않음
- 먼저 게임에서 승리해야만 레이팅 점수를 얻을 수 있어 승리 여부가 1차 타겟으로 적절
- 스코어와 레이팅의 관계도 중요 – 자신보다 레이팅이 높은 Bot 상대로 승리하면 더 많은 추가 레이팅 점수 획득, 반대의 경우 더 많은 손실

## II. Kaggle 게임 플레이어 레이팅 예측

# 01 Data Overview & Pipeline

📁 Scrabble 게임 이해: Woogles.io



### 도메인: 실력 유관 요소 추정 & Bot에 대한 분석(2)

- **location 활용 가능성** - 한 턴에서 다수의 단어가 동시에 완성되면 각 단어마다 점수가 한번에 획득되므로 인접한 위치에 이미 알파벳이 있다면 획득 점수 역시 높을 것으로 예상
- **move 활용 가능성** - move는 실제 배치 완료한 단어 표시,
  1. 단어 길이가 길수록 고득점으로 이어질 가능성이 높고,
  2. 알파벳마다 점수가 다르게 부여되어 **희귀한 알파벳**(V, Q, Z 등)으로 단어 배치를 성공하면 고득점 가능
- 단어를 완성시키지 못하면 실력 부족 의심 가능, **패스**(0점 처리)가 많을수록 고득점 확률이 적다고 판단 가능

## II. Kaggle 게임 플레이어 레이팅 예측

### 02 Preprocessing

#### 전처리: 최종 데이터프레임 구성 및 주요 수치형 변수 요약

game_id :	게임 번호-Key
nickname :	플레이어 닉네임
score :	해당 게임 내 총 득점
rating :	게임 후 레이팅 점수-Target
first / winner :	선공 / 승패 여부
max_overtime_minutes :	최대 추가 시간
opp_rating :	상대 봇의 레이팅
opp_score :	상대 봇의 게임 내 총 득점
bingo :	빙고 횟수
Turn_ ... _sum :	Six-Zero Rule 적용 여부
points_mean :	1턴당 득점 평균
move_len_mean :	제시한 단어 길이 평균
difficult_word_mean :	단어 난이도 평균
rack_ ... _sum :	덱 내 알파벳 7개 미만 턴

game_id	nickname	score	rating	first	time_control_name	game_end_reason	winner	lexicon	rating_mode	max_overtime_minutes	opponent
1	stevy	429.0	1500.0	0.0	regular	STANDARD	0.0	NWL20	CASUAL	1.0	BetterBot
3	davidavid	440.0	1811.0	0.0	regular	STANDARD	0.0	CSW21	RATED	5.0	BetterBot
4	Inandoutworker	119.0	1473.0	0.0	regular	RESIGNED	1.0	CSW21	CASUAL	1.0	BetterBot
5	stevy	325.0	1500.0	0.0	regular	STANDARD	1.0	NWL20	CASUAL	1.0	STEEBot
6	HivinD	378.0	2029.0	1.0	regular	STANDARD	0.0	CSW21	RATED	1.0	STEEBot
8	AliSalman1	414.0	2067.0	0.0	regular	STANDARD	1.0	CSW21	RATED	1.0	HastyBot
9	cccc	364.0	1641.0	0.0	regular	STANDARD	1.0	NWL20	RATED	1.0	BetterBot
10	squashy	299.0	1838.0	1.0	regular	STANDARD	0.0	CSW21	RATED	1.0	BetterBot
12	BB-8	351.0	1500.0	0.0	regular	STANDARD	1.0	ECWL	CASUAL	10.0	HastyBot
13	Trayz	434.0	2017.0	0.0	regular	STANDARD	0.0	CSW21	CASUAL	1.0	STEEBot

opp_rating	opp_score	bingo	turn_type_Six-Zero Rule_sum	points_mean	move_len_mean	difficult_word_mean	rack_len_less_than_7_sum
1637.0	335.0	2.0	0	30.642857	3.857143	0.071429	2
2071.0	318.0	1.0	0	31.428571	4.357143	0.142857	1
1936.0	478.0	0.0	0	8.500000	2.928571	0.071429	0
1844.0	427.0	1.0	0	20.312500	3.375000	0.062500	2
2143.0	427.0	1.0	0	31.500000	4.750000	0.166667	0
2244.0	528.0	2.0	0	37.636364	4.636364	0.272727	0
1624.0	464.0	2.0	0	24.266667	4.133333	0.133333	1
1972.0	415.0	1.0	0	21.357143	3.714286	0.071429	0
1614.0	408.0	1.0	0	23.400000	4.000000	0.000000	1
2122.0	381.0	4.0	0	39.454545	4.000000	0.272727	0

## II. Kaggle 게임 플레이어 레이팅 예측

### 02 Preprocessing

#### 📁 전처리: 최종 데이터프레임 구성 및 범주형 변수 처리

time\_control\_name : 제한 시간 설정  
game\_end\_reason : 게임 종료 사유  
lexicon : 사용한 단어사전  
rating\_mode : 레이팅 반영 여부  
opponent : 상대 봇 닉네임

time_control_name	game_end_reason	winner	lexicon	rating_mode	max_overtime_minutes	opponent
regular	STANDARD	0.0	NWL20	CASUAL	1.0	BetterBot
regular	STANDARD	0.0	CSW21	RATED	5.0	BetterBot
regular	RESIGNED	1.0	CSW21	CASUAL	1.0	BetterBot
regular	STANDARD	1.0	NWL20	CASUAL	1.0	STEEBot
regular	STANDARD	0.0	CSW21	RATED	1.0	STEEBot
regular	STANDARD	1.0	CSW21	RATED	1.0	HastyBot
regular	STANDARD	1.0	NWL20	RATED	1.0	BetterBot
regular	STANDARD	0.0	CSW21	RATED	1.0	BetterBot
regular	STANDARD	1.0	ECWL	CASUAL	10.0	HastyBot
regular	STANDARD	0.0	CSW21	CASUAL	1.0	STEEBot

#### ➡ 1. 그대로 사용(none ver.)

Tree 기반 등 범주형 변수를  
그대로 인식 가능한 모델에 적용

#### ➡ 2. Label Encoding ver.

One-Hot Encoding으로 인한  
용량 문제, 차원의 저주 고려한 대안

#### ➡ 3. One-Hot Encoding ver.

더미 변수 가능한 모두 반영

## II. Kaggle 게임 플레이어 레이팅 예측

### 03 Modeling

#### 📁 모델링 및 성능 비교

##### A. 단일 모델 파라미터 튜닝













RF, LR, Ridge, Lasso, KNN, Elastic Net과  
 1차 비교 결과 Xgboost가 적합, Gridsearch로 튜닝  
 3. One-Hot 사용하여 RMSE 145.37739

##### B. Sklearn으로 모델 블렌딩

DT, LR, Ridge, Lasso, Elastic Net과  
 1차 비교 결과 RF와 Xgboost가 적합, Soft Voting  
 2. Label 사용하여 RMSE 142.34571

##### C. AutoML으로 모델 블렌딩

특정 지표 최적화 기능 고려하여 블렌딩 및 예측,  
 RMSE 145.15021

129	▼ 1	wonjunc		142.34571
130	▲ 2	Fragrant Alligator		142.52291
131	▼ 1	アンサンブル	  	142.71589
132	▼ 44	CoTi22 Keen Mole	  	143.32364
133	▲ 1	guoyaobit		143.93634
134	▼ 1	Juwon Yeo		144.50110
135	—	eliot1113		145.15021
136	—	Min YoonKi		145.37739

## II. Kaggle 게임 플레이어 레이팅 예측 Discussion

### 📁 전체 요약 및 고찰

1주차

분류/예측 팀 분할  
**팀 구성**

2주차 ~ 1차 특별세션

**1차 프로젝트:머니**

Finda 어플 사용자  
대출신청 예측

5주차 ~ 7주차

**2차 프로젝트:게임**

Kaggle 게임 플레이어  
레이팅 예측

- 10만~1,000만 행 단위의 대형 데이터셋 여러 개를 연결하여 ML 모델이 학습할 수 있도록 전처리
- 1차 프로젝트의 경우 결측치 및 분포 왜곡 문제 해결과, 대형 데이터 예측을 위해 다방면으로 접근
- 2차 프로젝트의 경우 해당 게임을 직접 플레이하는 등 도메인 지식 확충 및 적절한 변수 탐색, 적용
- Sklearn과 AutoML 라이브러리의 다양한 기능 활용해 모델 탐색 및 최종 모델 튜닝, 예측



이상으로 발표를 마칩니다.

# 감사합니다!

KUBIG 22-2 장기 프로젝트 분류/예측 1팀  
인간지능, 머니&게임