

카트라이더 open API를 이용한 메타변화 분석

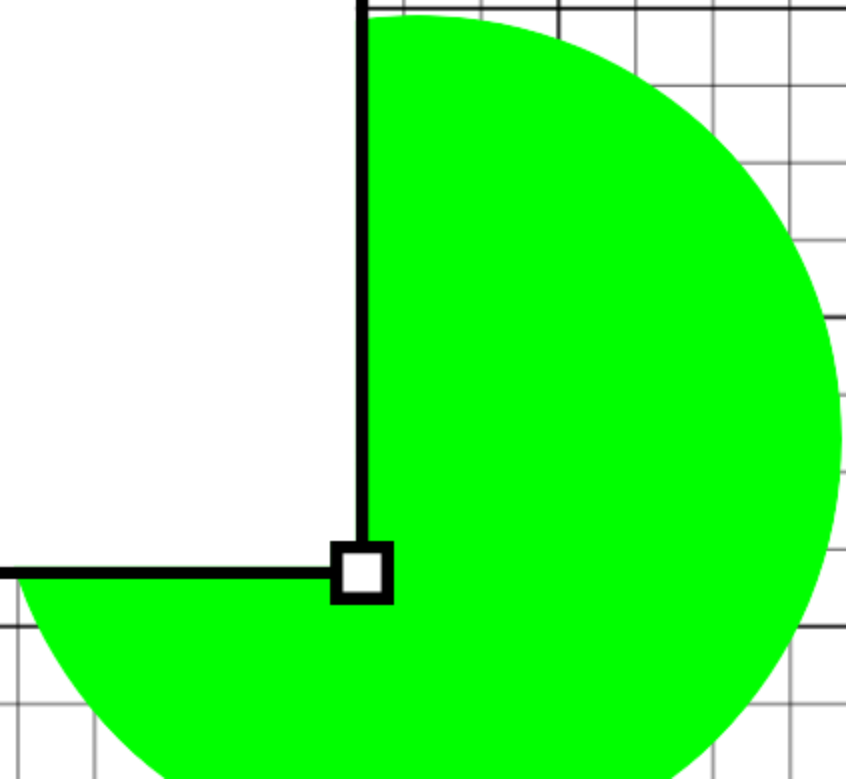
코드스테이츠 AI 11기 박성우





목차

- 1 데이터 선정 이유 및 문제 정의
- 2 가설 및 평가지표, 베이스라인 선택
- 3 EDA와 데이터 전처리
- 4 머신러닝 방식 적용 및 교차검증
- 5 머신러닝 모델 해석





최초의 V1 엔진 레전드 카트

- ☑ 비교적 이른 신규등급
1월 20일 출시
시간의 상점으로 출시

새로운 엔진과 메타변화

- ☑ 파츠의 급격한 공급
카트 파츠의 중요도 증가
기존 카트의 선호도 감소



문제 해결 과정

1 순위 예측

v1 엔진 레전드 카트 출시 이전, 이후의
순위를 예측합니다. (분류)

2 순위에 영향을 준 요소 탐색

각 순위에 영향을 준 요소들의 중요도를
파악합니다.

3 인사이트 도출

요소별 중요도를 통해 메타에 영향을 주는
요소들에 대해서 탐색합니다.



V1 엔진 레전드 카트 출시 이전

2021년 12월 4일 ~ 12월 12일

기간 내 매치정보 75401개

매치참여 유저수 113949명



V1 엔진 레전드 카트 출시 이후

2022년 3월 1일 ~ 3월 9일

기간 내 매치정보 75634개

매치참여 유저수 116394개

open API 뜯어보기

매치 리스트 조회

매치 고유 식별자

매치 게임 유형

매치 상세정보 조회

matchId : 매치 고유 식별자
matchType : 매치 종류
matchResult : 매치 결과
gameSpeed : 게임 스피드
startTime : 게임 시작시간
endTime : 게임 종료시간
playTime : 게임 진행시간
channelName : 채널명
trackId : 트랙 고유 식별자

player : 플레이어 정보

character : 캐릭터
kart : 카트
license : 라이선스
pet : 펫
flyingPet : 플라잉펫
partsEngine : 엔진파트
partsHandle : 핸들파트
partsWheel : 휠파트
partsKit : 킷파트
matchRank : 순위
matchWin : 승리여부
matchTime : 주행시간
matchRetired : 리타이어 여부
rankinggrade2 : 리뉴얼 라이선스
characterName : 캐릭터 닉네임

가설 및 평가지표, 베이스라인

가설 : v1 엔진 레전드 카트

출시 이후에 카트와 파츠의 영향력이 올라갔을 것이다.

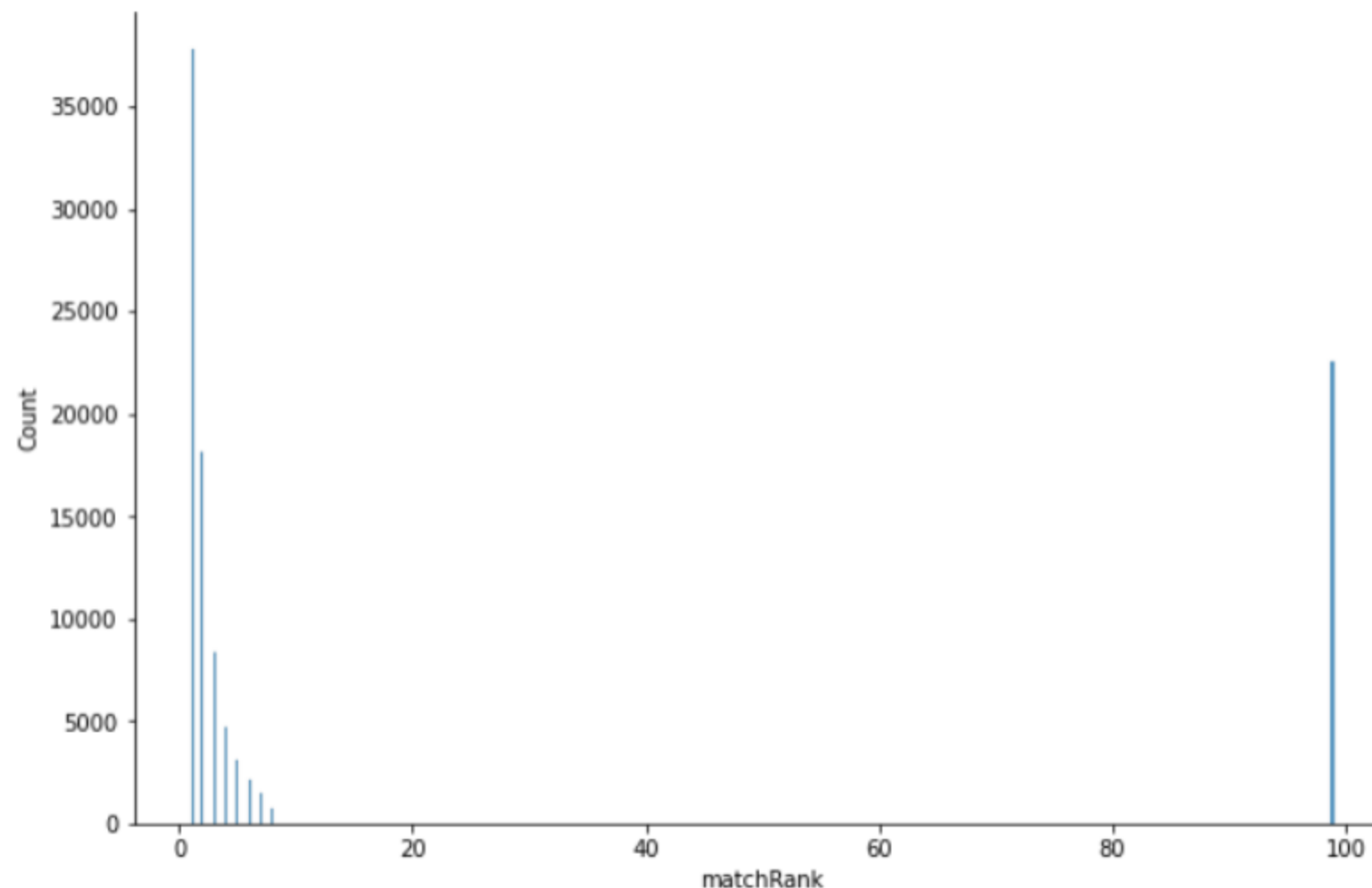
순위 예측이 필요하므로 분류모델
순위를 나타내는 matchRank 를 Target으로 지정

평가지표 : f1-score

불균형한 데이터, 다중분류 이므로
그에 맞는 평가지표인 f1-score를 이용

베이스라인 : 최빈값 예측모델

분류 모델이므로 최빈값을 이용



99는
리타이어를 의미

MatchRank의 분포

EDA와 데이터 전처리

Drop한 요소들

1. 중복값 제거

2. 관련없는 요소 제거

- 1) matchId : 매치 고유 식별자
- 2) character : 캐릭터는 플레이에 영향을 주지 않음
- 3) license : 이전 라이선스는 "값을 가짐
- 4) matchRetired : 순위로 리타이어 여부를 나타냄
- 5) characterName : 유저 닉네임
- 6) channelName : battle 유형의 게임은 협력전 (일부 행 제거)
- 7) matchRank : 0은 강제종료를 의미 (일부 행 제거)

3. Data Leakage 방지

- 1) matchResult, matchWin : target에 직접적 영향을 줌
- 2) startTime, endTime ,playTime, matchTime : target에 직접적 영향을 줌

EDA와 데이터 전처리

결측치 처리

kart : api 자체 공백값 (drop)

pet : 펫 미장착 (non_pet 대체)

flyingPet : 플라잉 펫 미장착 (non_flyingPet 대체)

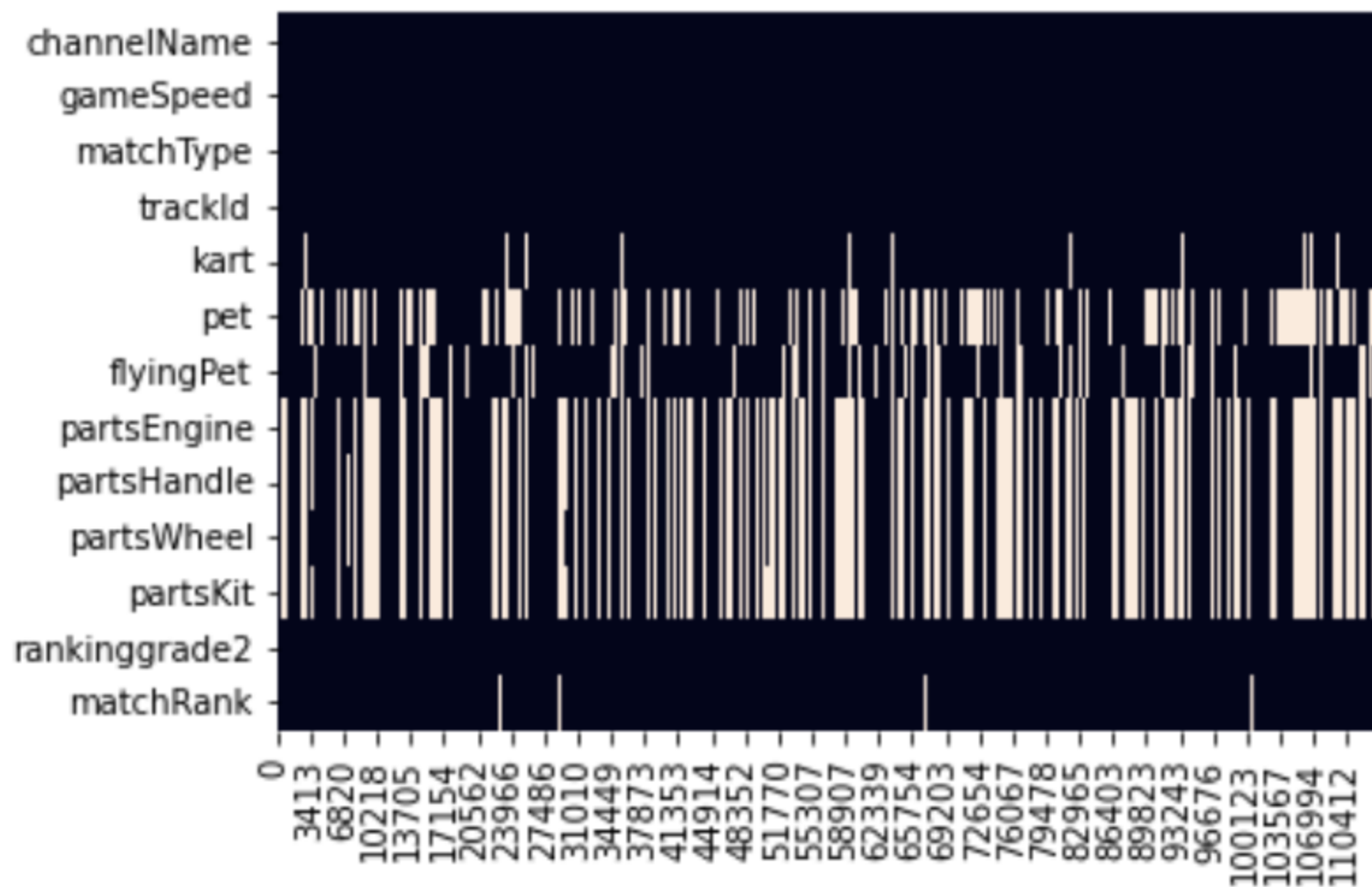
partsEngine : X엔진 이상 버전 미등록 (over_ver_engine 대체)

partsHandle : X엔진 이상 버전 미등록 (over_ver_handle 대체)

partsWheel : X엔진 이상 버전 미등록 (over_ver_wheel 대체)

partsKit : X엔진 이상 버전 미등록 (over_ver_kit 대체)

matchRank : 강제종료로 인한 공백값 (drop)



머신러닝 방식 적용 및 교차검증

기준모델(베이스라인) 의 성능

최빈값인 1위를 기준으로 한 모델

프로젝트 1 - 최빈값 비율 : 38.13%, f1-score : 0.2105

프로젝트 2 - 최빈값 비율 : 37.85%, f1-score : 0.2078

Train, Test 셋 분류

7대 3의 비율로 나눔

프로젝트 1 - Train(69322개), Test(29710개)

프로젝트 2 - Train(71387개), Test(30595개)

머신러닝 방식 적용 및 교차검증

시도한 모델의 종류

1. RandomForestClassifier

과적합 문제를 회피하기 좋음

2. CatBoostClassifier

boosting의 장점을 가져가며 과적합 방지

3. XGBClassifier

강력한 병렬처리, 과적합방지, 자체 교차검증 가능

3개의 모델 모두 불균형 데이터 조정 option을 지원합니다

머신러닝 방식 적용 및 교차검증

RandomForestClassifier (프로젝트1)

1. 하이퍼파라미터 조정

max_depth : 7

max_features : 4

n_estimators : 271

class_weight = "balanced"

2. 머신러닝 평가지표

정확도 : 25.82%

정밀도 : 39.17%

재현율 : 25.82%

3. 기준모델 비교 평가지표

기준모델 f1 : 0.2105

제작모델 f1 : 0.2897

RandomForestClassifier (프로젝트2)

1. 하이퍼파라미터 조정

max_depth : 7

max_features : 4

n_estimators : 271

class_weight = "balanced"

2. 머신러닝 평가지표

정확도 : 24.92%

정밀도 : 35.52%

재현율 : 24.92%

3. 기준모델 비교 평가지표

기준모델 f1 : 0.2078

제작모델 f1 : 0.2838

머신러닝 방식 적용 및 교차검증

CatBoostClassifier (프로젝트1)

1. 하이퍼파라미터 조정

```
loss_function = 'MultiClass'  
class_weights=[0.619,0.818,0.916,0.  
953,0.969,0.979,0.986,0.993,0.773]  
early_stopping_rounds=35  
verbose=100
```

2. 머신러닝 평가지표

정확도 : 39.63%
정밀도 : 33.56%
재현율 : 39.63%

3. 기준모델 비교 평가지표

기준모델 f1 : 0.2105
제작모델 f1 : 0.3540

CatBoostClassifier (프로젝트2)

1. 하이퍼파라미터 조정

```
loss_function = 'MultiClass'  
class_weights=[0.622,0.77,0.928,0.9  
28,0.974,0.982,0.987,0.993,0.789]  
early_stopping_rounds=35  
verbose=100
```

2. 머신러닝 평가지표

정확도 : 40.42%
정밀도 : 35.70%
재현율 : 40.42%

3. 기준모델 비교 평가지표

기준모델 f1 : 0.2078
제작모델 f1 : 0.3701

머신러닝 방식 적용 및 교차검증

XGBClassifier (프로젝트1)

1. 하이퍼파라미터 조정

max_depth = 21

n_estimators = 190

XGBClassifier 자체로는 다중클래스에 가
중치를 줄 수 없어 class_weight 이용

2. 머신러닝 평가지표

정확도 : 31.81%

정밀도 : 25.59%

재현율 : 31.81%

3. 기준모델 비교 평가지표

기준모델 f1 : 0.2105

제작모델 f1 : 0.2767

XGBClassifier (프로젝트2)

1. 하이퍼파라미터 조정

max_depth = 21

n_estimators = 190

XGBClassifier 자체로는 다중클래스에 가
중치를 줄 수 없어 class_weight 이용

2. 머신러닝 평가지표

정확도 : 32.98%

정밀도 : 27.12%

재현율 : 32.98%

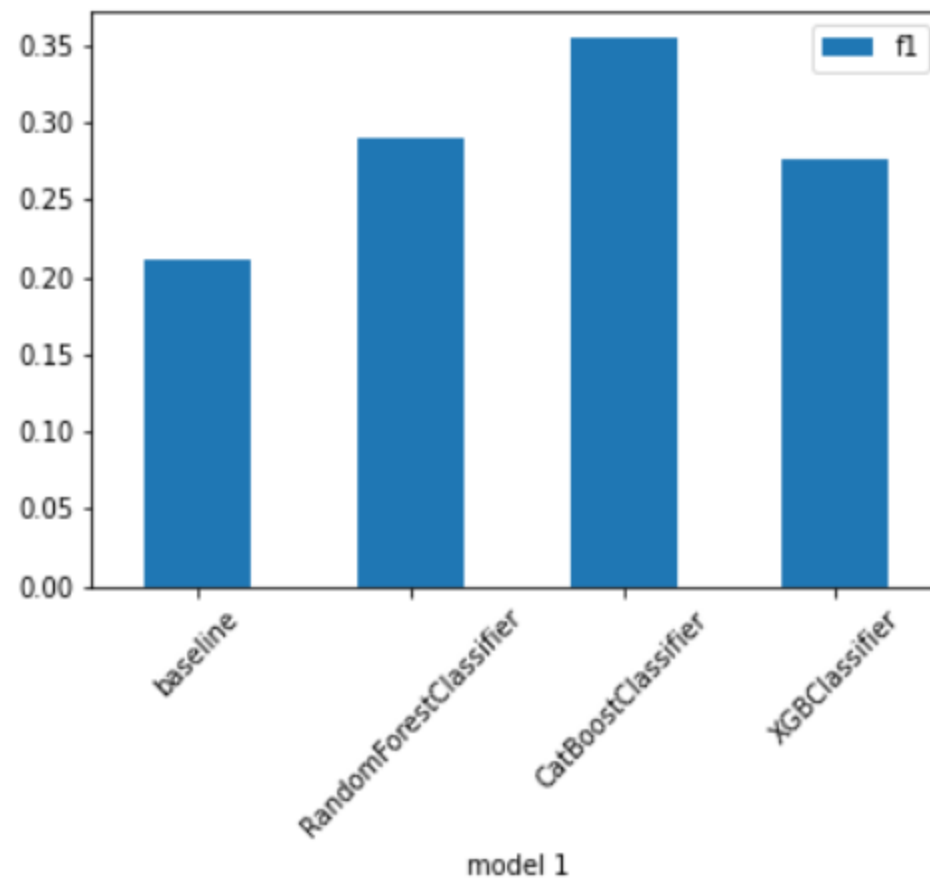
3. 기준모델 비교 평가지표

기준모델 f1 : 0.2078

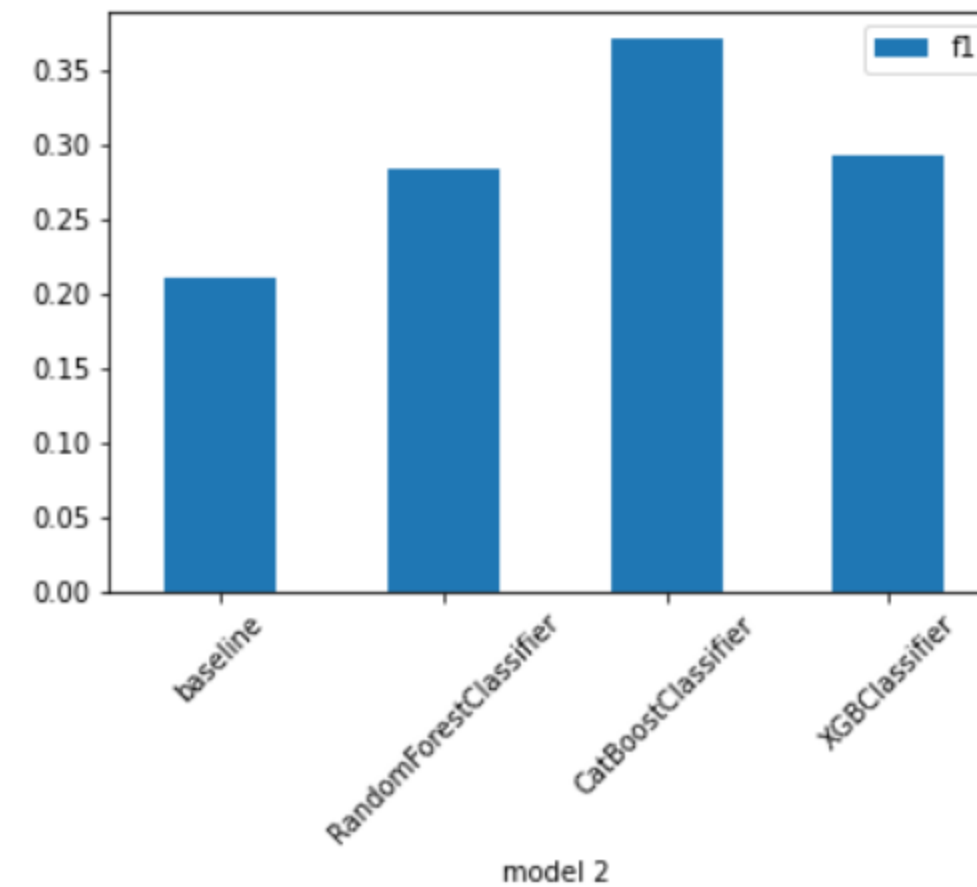
제작모델 f1 : 0.2931

베이스라인과의 성능 비교

프로젝트 1



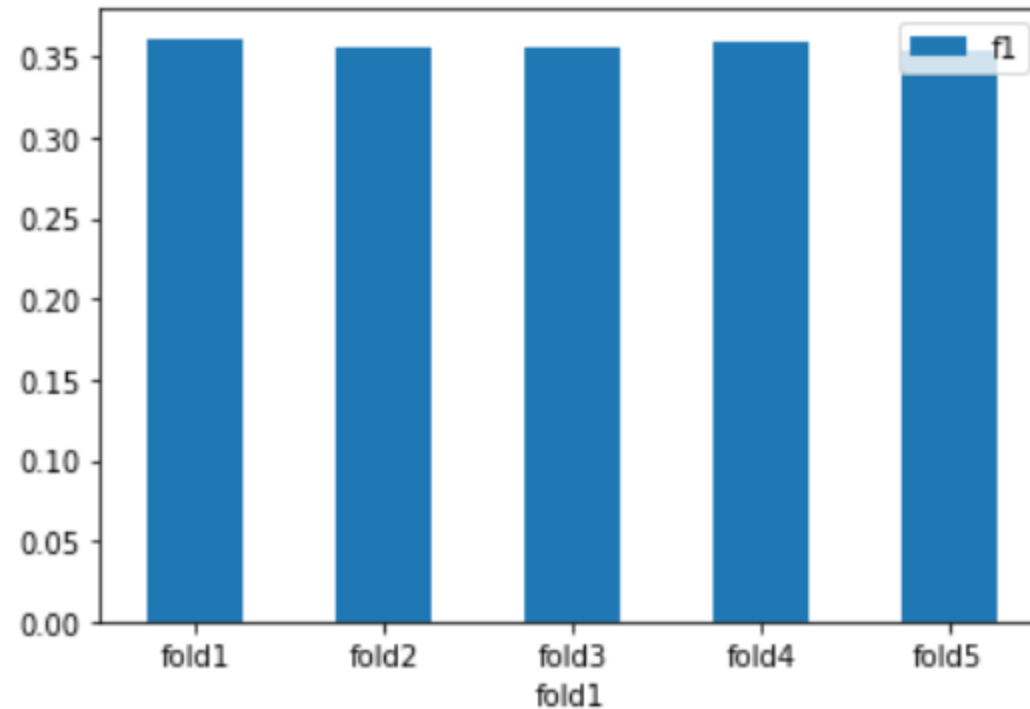
프로젝트 2



모든 모델들이 기준모델보다 성능 향상, CatBoostClassifier가 가장 성능이 좋았다

CatBoostClassifier 교차검증

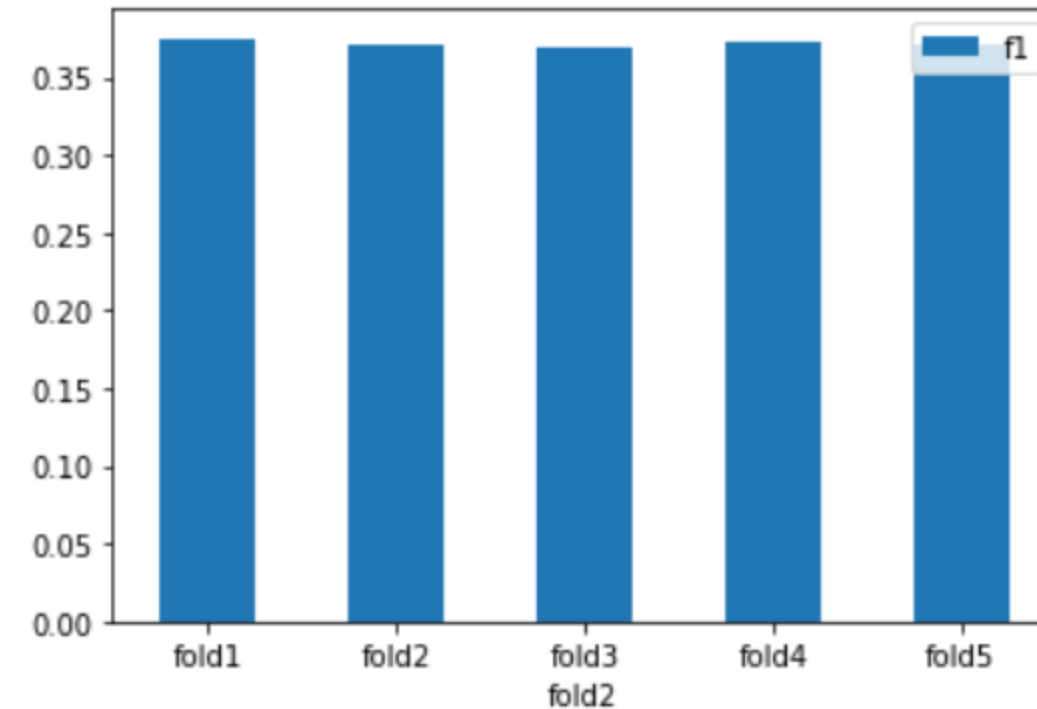
프로젝트 1



mean : 0.3575

std : 0.0028

프로젝트 2



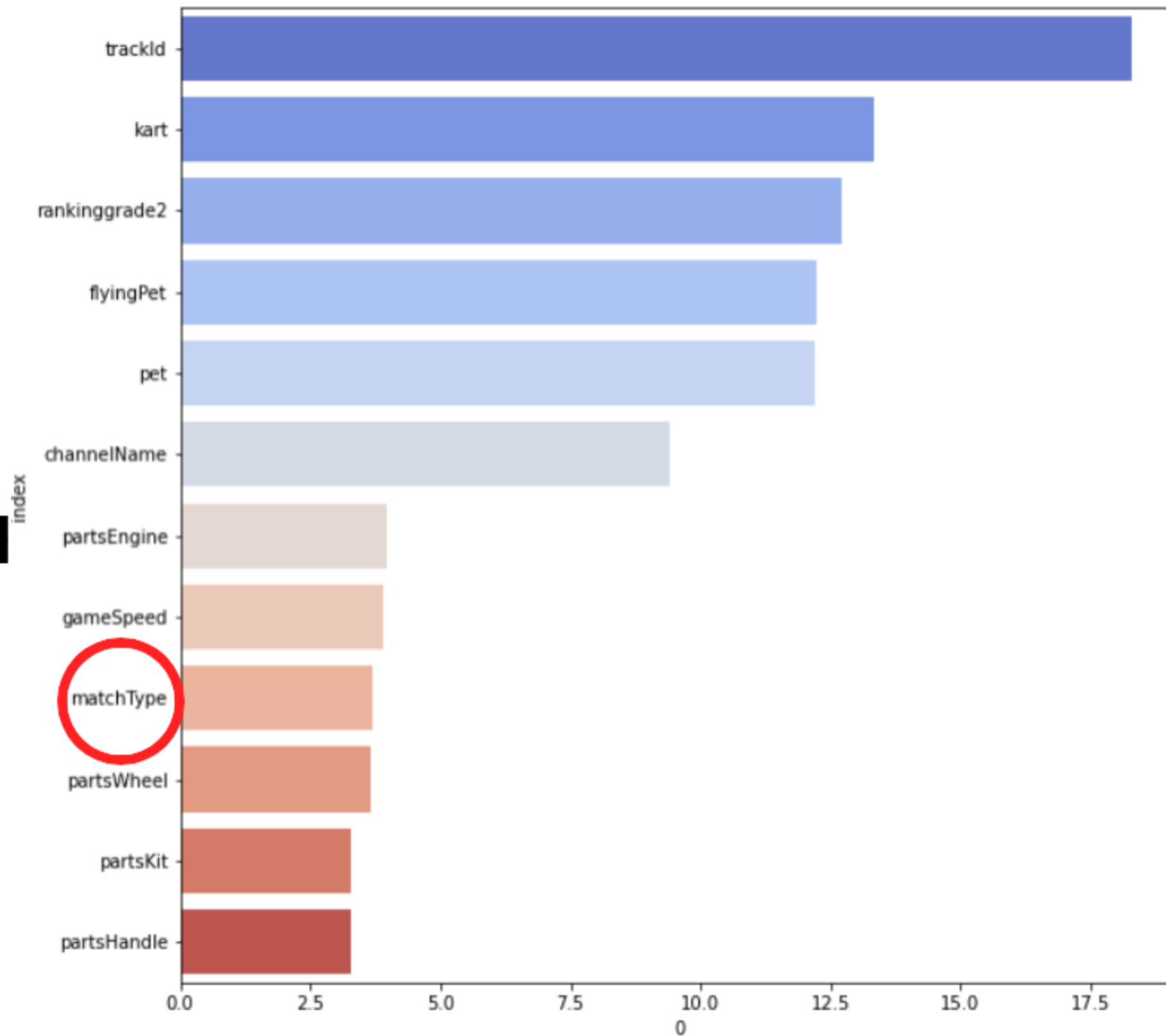
mean : 0.3718

std : 0.0018

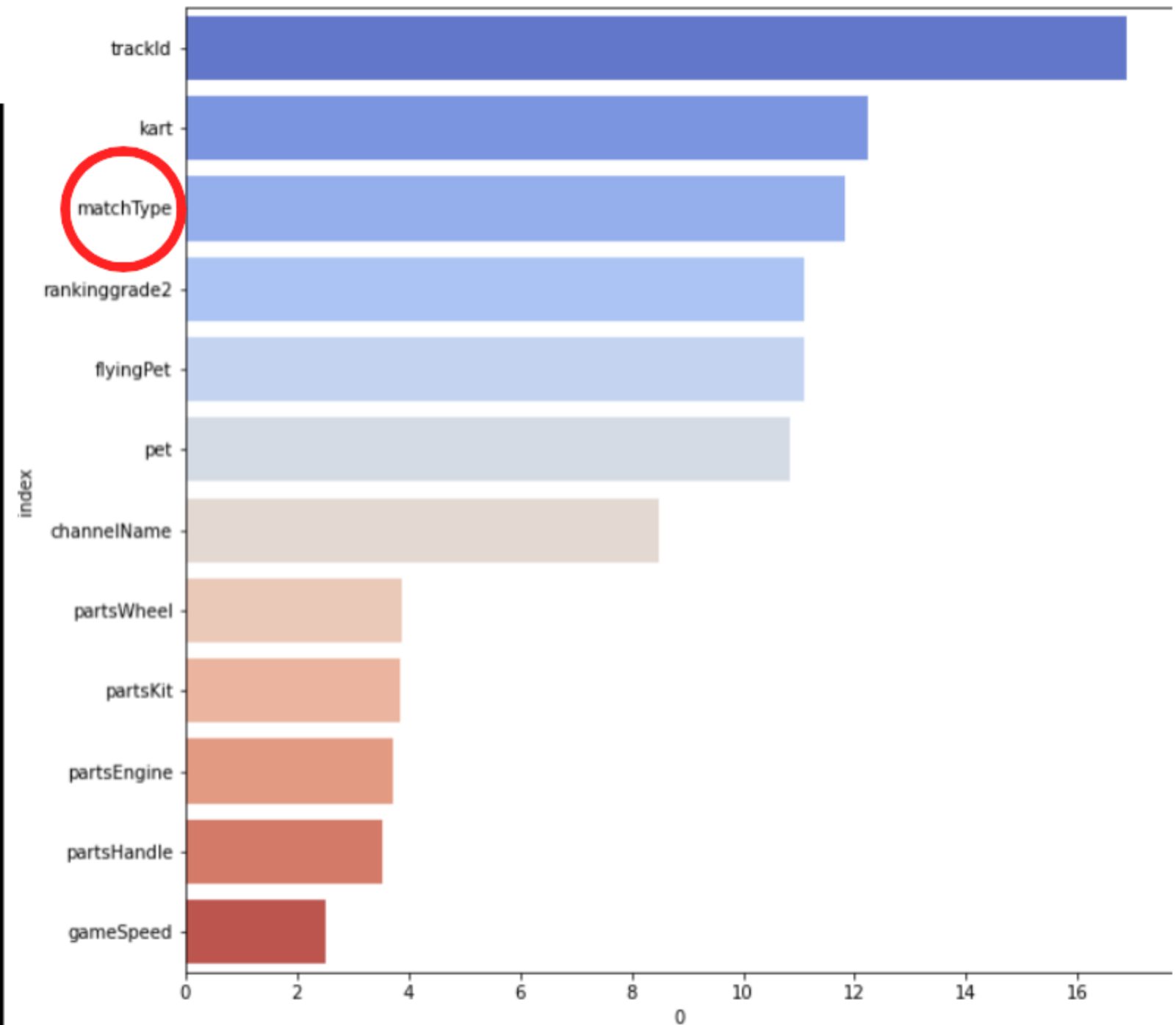
std 가 작다는 것은 모델 complexity에 비해 sample수가 충분히 많다

머신러닝 모델 해석

매치 타입: 클럽대전 등
특수모드를 포함한 매치 id



프로젝트 1의 특성중요도



프로젝트 2의 특성중요도

머신러닝 모델 해석

가설 : v1 엔진 레전드 카트
출시 이후에 카트와 파츠의 영향력
이 올라갔을 것이다.

결과 : v1 엔진 레전드 카트
출시 이후에 매치타입의 영향력이 올라갔다



해석 :
v1 엔진 레전드 카트
출시는 특수모드의 승률에 영향을 주었다.

프로젝트 피드백

아쉬운 점

1. 특정 ver 이후의 파트들이 Nan 값
파트의 다양성을 제대로 보여주지 못함
(좋지 못한 데이터)

2. 낮은 f1-score
MatchTime을 제거하기 전 f1-score : 0.65
제거한 후 f1-score : 0.37
(하지만 데이터 누수를 예상해 제거했다.)

3. 그 외 아쉬운 점
데이터 추출에 6시간, 모델의 적용과정에 몇십분
처음해보는 api 사용으로 인한 시간지연 +α
(더 많은 내용을 다루지 못했다)

이상으로 발표를 마칩니다

**[https://github.com/colacan100/Kart
rider_Meta_Analysis_Project](https://github.com/colacan100/Kart_rider_Meta_Analysis_Project)**

THANK YOU!