DATA133

2022 문화·관광 빅데이터 분석대회

# 스포츠 활동 및 산업 활성화 방안

## 스포츠 관련 데이터 분석을 통한 수요 증대 방법 제안

# Contents


01 주제 선정

02 분석 개요

03 스포츠경험자 분류

04 스포츠 추천시스템

05 스포츠 기피 이유 분석

06 결론 및 활용 방안

# 01 주제 선정 - 배경

- 오랫동안 강조되어 온 생활체육 활동의 중요성

전통적인 엘리트체육 중심의 정책에서 생활체육의 중요성이 부각되기 시작한 것은 체육에 대한 요구가 건강증진과 복지의 측면에서 고령화 사회, 생활습관병 유병율 증가, 주5일 근무제 시행으로 인한 여가시간의 증가 등 사회·환경적 변화에 대응하기 위한 해결방안으로 제시되고 있기 때문임.

- 정부 차원에서 강조하는 생활체육 정책

표 1-8. 문재인 정부 체육정책

| 부문별 목표 | 추진내용 |
|---|---|
| 스포츠 참여기회 확대, 국민스포츠 강화 | • 공공체육시설확충 및 스포츠클럽 지원<br>• 장애인형국민체육센터 건립 및 장애인체력인증센터 운영<br>• 장애인생활체육지도자 배치 |

표 1-9. 2030스포츠비전 추진전략 및 핵심과제

| 추진전략 | 10대 핵심과제 | 25개 세부과제 |
|---|---|---|
| 신나는 스포츠 | Ⅰ 평생 동안 즐기는 맞춤형 스포츠 프로그램 | (1) 3세부터 시작하는 스포츠 활동 습관화<br>(2) 청소년의 스포츠 경험 다양화<br>(3) 100세까지 이어지는 스포츠 활동 일상화 |
| | Ⅱ. 언제 어디서나 편하게 이용하는 스포츠 시설 | (4) 일상에서 편리하게 이용하는 스포츠시설<br>(5) 스포츠시설 및 정보의 체계적 관리 |

출처 : 문화체육관광부(2021), 《2020 체육백서》.

**60 모두를 위한 스포츠, 촘촘한 스포츠 복지 실현** (문체부)

□ **과제목표**

○ 생애주기별 스포츠활동 지원, 스포츠 인프라 확충 등으로 스포츠 저변 확대

출처 : 제20대 대통령실(2022), 《윤석열정부 110대 국정과제》.

- MZ세대의 골프, 테니스에 대한 관심 증대

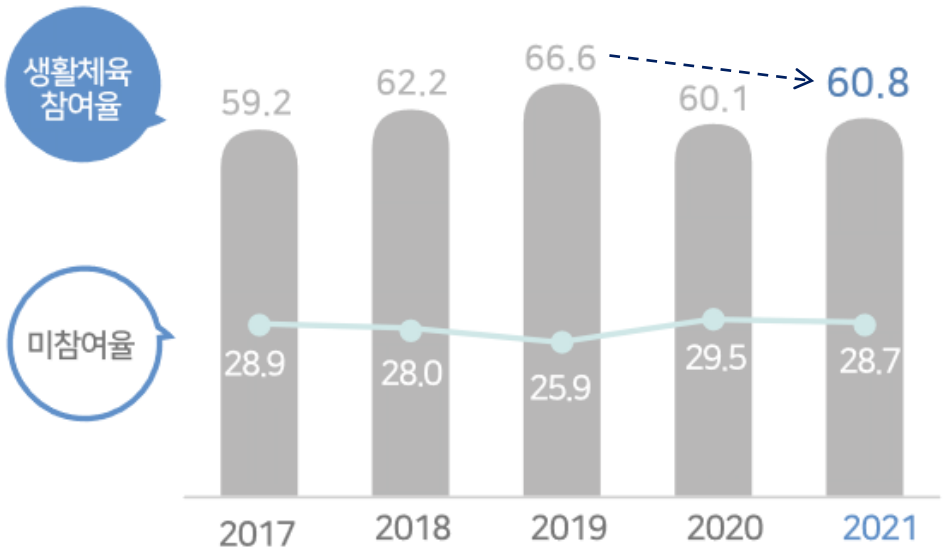
## 2030세대 골프인구만 115만명...

"돈 아깝지 않다" 골프에 이은 또다른 플렉스...'
귀족 스포츠' 테니스 즐기는 MZ


스포츠 여가활동에 대한 관심 증대

# 01 주제 선정 - 배경

- 코로나19 이후 회복되지 않은 생활체육 참여율




* 생활체육 참여율 : 주 1회 이상, 1회 운동 시 30분 이상 규칙적으로 생활체육에 참여한 비율

출처 : 문화체육관광부(2021), 《2021 국민생활체육조사》.

- 실내 스포츠 참여율의 감소

**2020년 생활체육 참여율 60.1%,**
**코로나19 영향으로 전년 대비 6.5%포인트 감소**
**- 실내종목 참여율 감소, 민간·공공체육시설 이용률 감소 -**

출처 : 문화체육관광부(2020.12.30), 《국민생활체육조사 보도자료》.

## 문제의식

스포츠 참여율을 회복시키려면?

## 분석 목표

스포츠 관련 데이터를 분석하여

스포츠 활동 참여율 증가 방법을 모색하자!

# 02 분석 개요 - 분석흐름

## EDA

데이터
- 국민여가활동조사

분석 과제 결정
- 이진 분류
- 추천 시스템

## 이진 분류

데이터 전처리

모형 생성
- 랜덤포레스트
- 그래디언트 부스팅
- 변수 중요도

분석 결과

## 추천 시스템

데이터
- 신한카드 데이터

데이터 전처리
- 사용자, 평점 변수 생성

모형 생성
- SVD
- KNNBaseline

## 기피 이유 분석

데이터
- 전국 체육시설 정보 데이터

시각화
- 체육시설 접근성 파악

# 02 분석 개요 - 데이터

## 1) 국민여가활동조사 (2021)

| 변수명 | 내용 |
|---|---|
| q1_C | 지난 1년간 한 번 이상 참가한 여가활동 (스포츠관람활동) |
| q1_D | 지난 1년간 한 번 이상 참가한 여가활동 (스포츠참여활동) |
| q11_3_D | 평일에 희망하는 여가활동 (스포츠참여활동) |
| q12_3_D | 휴일에 희망하는 여가활동 (스포츠참여활동) |
| q45 | 지난 1주일 간 경제활동 여부 |
| DM1~DM6 | 성별, 연령, 학력, 가구원수, 혼인상태, 가구주 여부 |
| DM8 | 가구소득 |
| DM11 | 거주도시 |

## 2) 신한카드 데이터 (18년 1월 ~ 22년 4월)

| 변수명 | 내용 |
|---|---|
| v1 | 거주지 |
| gb3 | 가맹점 업종 대분류 |
| gb2 | 가맹점 업종 소분류 |
| sex_ccd | 성별 |
| cln_age_r | 연령대 |
| ta_ym | 이용년월 |
| vlm | 취급액 (단위: 원) |
| usec | 이용건수 (단위: 건) |

# 02 분석 개요 - 데이터

### 3) 전국체육시설 상세 정보데이터 (2022) - 서울올림픽기념 국민체육진흥공단

전국체육시설안전을 위해 관리하는 체육시설 상세정보 데이터

| 변수명 | 내용 |
|---|---|
| addrCpNm | 시도 |
| addrCpbNm | 시군구 |
| addrEmdNm | 동읍면 |
| fcobNm | 체육시설업종 |
| ftypeNm | 체육시설유형 |
| faciPointX | 경도 |
| faciPointY | 위도 |

| 시도 | 시군구 | 동읍면 | 체육시설업종 | 체육시설유형 | 경도 | 위도 |
|---|---|---|---|---|---|---|
| 충청남도 | 아산시 | 덕지리 | 골프연습장업 | 실외 | 127.083229 | 36.836105 |
| 서울특별시 | 송파구 | 송파동 | 체력단련장업 | 체력단련장 | 127.107689 | 37.507343 |
| 서울특별시 | 은평구 | 녹번동 | 당구장업 | 당구장 | 126.929070 | 37.601665 |
| 서울특별시 | 영등포구 | 도림동 | 체력단련장업 | 체력단련장 | 126.895897 | 37.508331 |
| 충청남도 | 예산군 | 목리 | 당구장업 | 당구장 | 126.680237 | 36.661663 |
| ... | ... | ... | ... | ... | ... | ... |
| 서울특별시 | 강남구 | 신사동 | 체력단련장업 | 체력단련장 | 127.026062 | 37.524582 |
| 서울특별시 | 강남구 | 청담동 | 체력단련장업 | 체력단련장 | 127.044800 | 37.526165 |
| 서울특별시 | 강남구 | 신사동 | 체력단련장업 | 체력단련장 | 127.024025 | 37.519848 |
| 서울특별시 | 강남구 | 신사동 | 체력단련장업 | 체력단련장 | 127.027428 | 37.524422 |
| 충청남도 | - | 고남리 | 당구장업 | 당구장 | 126.408775 | 36.422363 |

# 02 분석 개요 - EDA

## 국민여가활동조사 EDA

### 스포츠경험 여부 칼럼 생성

```
# 스포츠경험자 여부 칼럼 생성
h21['sports_exp'] = np.NaN

h21.loc[(h21['q1_C'] != '99') | (h21['q1_D'] != '99'), 'sports_exp'] = '1'
h21.loc[(h21['q1_C'] == '99') & (h21['q1_D'] == '99'), 'sports_exp'] = '0'
```

'99': 경험 없음

q1_C (스포츠관람), q1_D (스포츠활동) 중 경험 **有** -> sports_exp == '1'
q1_C (스포츠관람), q1_D (스포츠활동) 중 경험 **無** -> sports_exp == '0'

### 스포츠여가생활 희망 여부 칼럼 생성

```
# 스포츠희망 여부 칼럼 생성
h21['sports_hope'] = '0' # 스포츠비희망자 == 0
h21.loc[(h21['q11_3_D'] != 98) | (h21['q12_3_D'] != 98), 'sports_hope'] = '1'
```

'98': 관심 없음

q11_3_D (평일), q12_3_D (휴일) 중 희망 스포츠 **有** -> sports_hope == '1'
q11_3_D (평일), q12_3_D (휴일) 중 희망 스포츠 **無** -> sports_hope == '0'

| | sports_exp |
|---|---|
| 1 | 6826 |
| 0 | 3223 |

스포츠여가생활 경험자 6,826명

스포츠여가생활 비경험자 3,223명

교차표 생성

| | sports_hope |
|---|---|
| 1 | 6972 |
| 0 | 3077 |

스포츠여가생활 희망자 6,972명

스포츠여가생활 비희망자 3,077명

| sports_hope | 0 | 1 |
|---|---|---|
| sports_exp | | |
| 0 | 1968 | 1255 |
| 1 | 1109 | 5717 |

# 02 분석 개요 - EDA

## 국민여가활동조사 EDA

스포츠경험여부와 스포츠희망여부 간의 관련성 분석

: 카이제곱 검정 수행

```
sports_hope      0     1
sports_exp
0             1968  1255
1             1109  5717
```

```
import scipy.stats as stats

result = stats.chi2_contingency(observed = sports_crosstab)
```

H0: 스포츠경험여부와 스포츠희망여부는 서로 독립이다

H1: 스포츠경험여부와 스포츠희망여부는 서로 독립이 아니다

```
카이제곱검정 결과 P-value:  0.0
```

유의수준 0.05 하에서 스포츠경험여부와 스포츠희망여부는 서로 독립이 아니다.

즉, 스포츠경험여부와 스포츠희망여부에 유의한 상관관계가 있다.

스포츠경험여부와 스포츠희망여부 간의 상관관계 확인

: 범주형 변수 간의 상관계수로 phi 상관계수 사용

```
from sklearn.metrics import matthews_corrcoef

sports_hope = h21['sports_hope'].values
sports_exp = h21['sports_exp'].values
print('두 범주형 변수 간 상관계수: ', round(matthews_corrcoef(sports_hope, sports_exp), 3))
```

```
두 범주형 변수 간 상관계수:  0.454
```

스포츠경험자는 여가활동으로 스포츠를 희망하는 비율이 높음

-> 스포츠비경험자에게 최소한의 스포츠 경험을 유도한다면

여가활동으로 스포츠를 희망하게 될 확률이 높을 것!

# 02 분석 개요 – 과제 설정

**국민여가활동조사**

스포츠경험자/스포츠비경험자 칼럼 생성

스포츠경험자/스포츠비경험자 이진분류 모형 생성

분류 모형의 Feature Importance 파악

**신한카드 데이터**

스포츠 종목별 'rating' (평점) 행렬 생성

스포츠 종목 추천 알고리즘 적용

**체육시설 데이터**

체육시설 접근성 시각화

1. 분류 모형을 통해 분류하여
스포츠경험자/비경험자의 특성 파악

2. 중요한 Feature를 사용해
스포츠 추천 모형에 대입하여
스포츠비경험자를 위한 맞춤형 스포츠 추천

3. 스포츠활동 기피 이유를 분석하여
정책적 조언

# 03 스포츠경험자 분류 – 전처리

## 국민여가활동조사 전처리

[스포츠경험자/스포츠비경험자 칼럼 생성]

```
# 스포츠경험자/스포츠비경험자 칼럼 생성
h21['sports_exp'] = np.NaN # 스포츠경험자 칼럼 생성
h21.loc[(h21['q1_C'] != '99') | (h21['q1_D'] != '99'), 'sports_exp'] = '1'
h21.loc[(h21['q1_C'] == '99') & (h21['q1_D'] == '99'), 'sports_exp'] = '0'
```

[변수선택으로 분류모형에 사용할 변수 추출]

```
step(lm(formula = df$target ~ weekly_worked_2 + gender_2 + age_2 +
          age_3 + age_4 + age_5 + age_6 + age_7 + edu_2 + edu_3 + edu_4 +
          family_2 + family_3 + marriage_3 + householder_2 + income_2 +
          income_3 + income_4 + income_5 + income_6 + income_7 + city_10 +
          city_11 + city_12 + city_13 + city_14 + city_15 + city_17 +
          city_2 + city_3 + city_4 + city_5 + city_6 + city_7 + city_8 +
          city_9, data = df), direction = 'both')
```

(주간근무여부, 성별, 연령, 학력, 거주도시, 가구원수, 가구주여부, 가구소득) 총 8개 변수의 더미변수들이 선택됨

[칼럼 추출 및 칼럼명 변경]

```
df = h21[['q45', 'DM1', 'DM2', 'DM3', 'DM4', 'DM5',
'DM6', 'DM8', 'DM11', 'sports_exp']]
df.columns = ['weekly_worked', 'gender', 'age', 'edu',
'family', 'marriage', 'householder', 'income', 'city', 'target'] # 칼럼명 변경

 df = df.loc[(df['age'] < 7) & (df['age'] > 1)]
```

'sports_exp'칼럼을 'target'으로 칼럼명 변경

신한카드 데이터에 없는 10대, 70대 이상의 연령 행 삭제

### 분류모형에 사용할 데이터프레임 생성

| weekly_worked | gender | age | edu | family | marriage | householder | income | city | target |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 5 | 3 | 3 | 2 | 1 | 3 | 15 | 0 |
| 2 | 2 | 6 | 1 | 3 | 3 | 2 | 2 | 15 | 0 |
| 1 | 1 | 5 | 3 | 3 | 3 | 1 | 2 | 15 | 0 |
| 1 | 1 | 5 | 3 | 2 | 1 | 1 | 2 | 15 | 1 |
| 1 | 1 | 3 | 4 | 3 | 2 | 1 | 5 | 15 | 1 |

target (0/1)을 분류하는 모형으로

RandomForest, GradientBoosting 사용

# 03 스포츠경험자 분류 – 모형 비교

## 이진 분류모형 – 랜덤포레스트, 그래디언트 부스팅 비교

두 모형에 그리드 서치 적용하여 정확도 비교

```
from sklearn.model_selection import GridSearchCV

params = { 'n_estimators' : [10, 100],
           'max_depth' : [6, 8, 10, 12],
           'min_samples_leaf' : [8, 12, 18],
           'min_samples_split' : [8, 16, 20]
            }

# RandomForestClassifier 객체 생성 후 GridSearchCV 수행
rf_clf = RandomForestClassifier(random_state = 0, n_jobs = -1)
grid_cv = GridSearchCV(rf_clf, param_grid = params, cv = 3, n_jobs = -1)
grid_cv.fit(train_X, train_y)

print('최적 하이퍼 파라미터: ', grid_cv.best_params_)
print('최고 예측 정확도: {:.4f}'.format(grid_cv.best_score_))
```

```
최적 하이퍼 파라미터:  {'max_depth': 8, 'min_samples_leaf': 12, 'min_samples_split': 8, 'n_estimators': 10}
최고 예측 정확도: 0.7238
```

max_depth = 8, min_samples_leaf = 12,

min_sample_split = 8, n_estimators = 10

Accuracy Score: 0.724

```
from sklearn.model_selection import GridSearchCV
from sklearn.ensemble import GradientBoostingClassifier

params = { 'n_estimators' : range(5, 50, 10),
           'max_depth' : range(3, 5),
           'learning_rate' : np.linspace(0.1, 1, 10),
           'max_features' : range(1, 4)
            }

# RandomForestClassifier 객체 생성 후 GridSearchCV 수행
gb_clf = GradientBoostingClassifier(random_state = 0)
grid_cv = GridSearchCV(gb_clf, param_grid = params, cv = 3, n_jobs = -1)
grid_cv.fit(train_X, train_y)

print('최적 하이퍼 파라미터: ', grid_cv.best_params_)
print('최고 예측 정확도: {:.4f}'.format(grid_cv.best_score_))
```

```
최적 하이퍼 파라미터:  {'learning_rate': 0.2, 'max_depth': 3, 'max_features': 2, 'n_estimators': 45}
최고 예측 정확도: 0.7329
```

max_depth = 3, learning_rate: 0.2

max_features = 2, n_estimators = 45

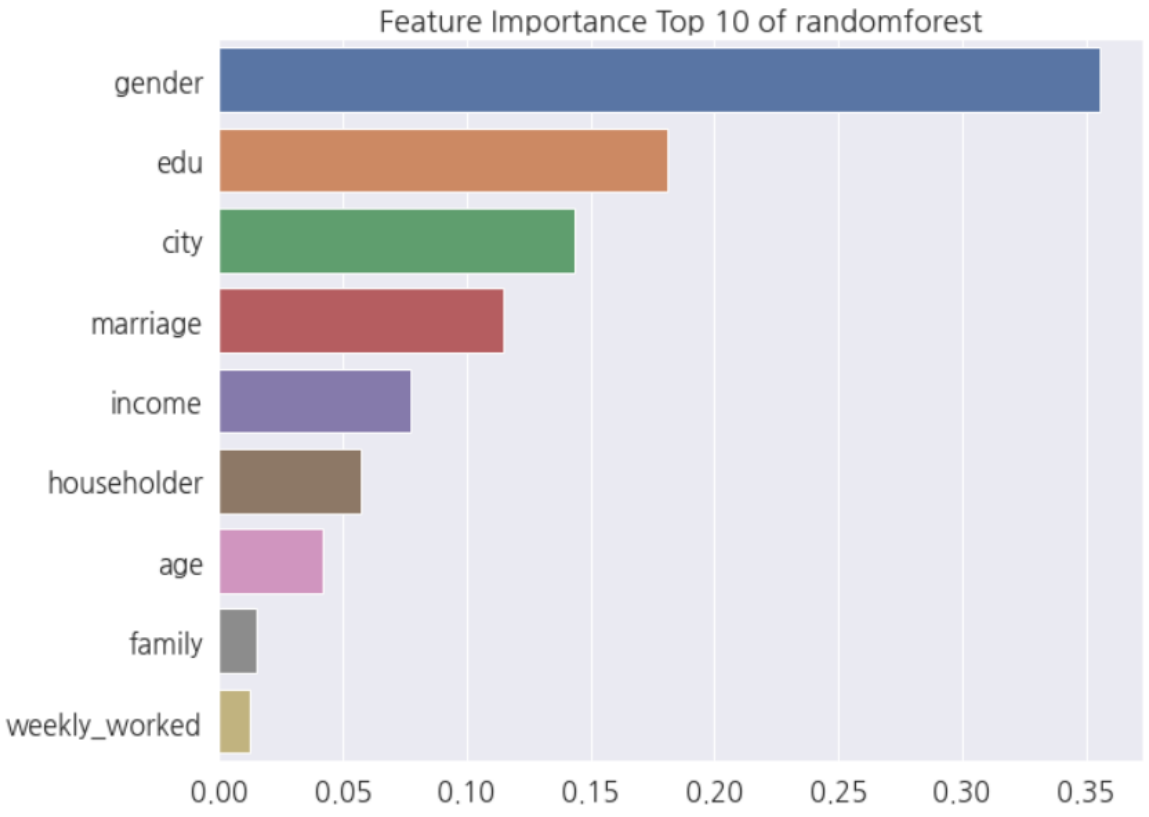
Accuracy Score: 0.733

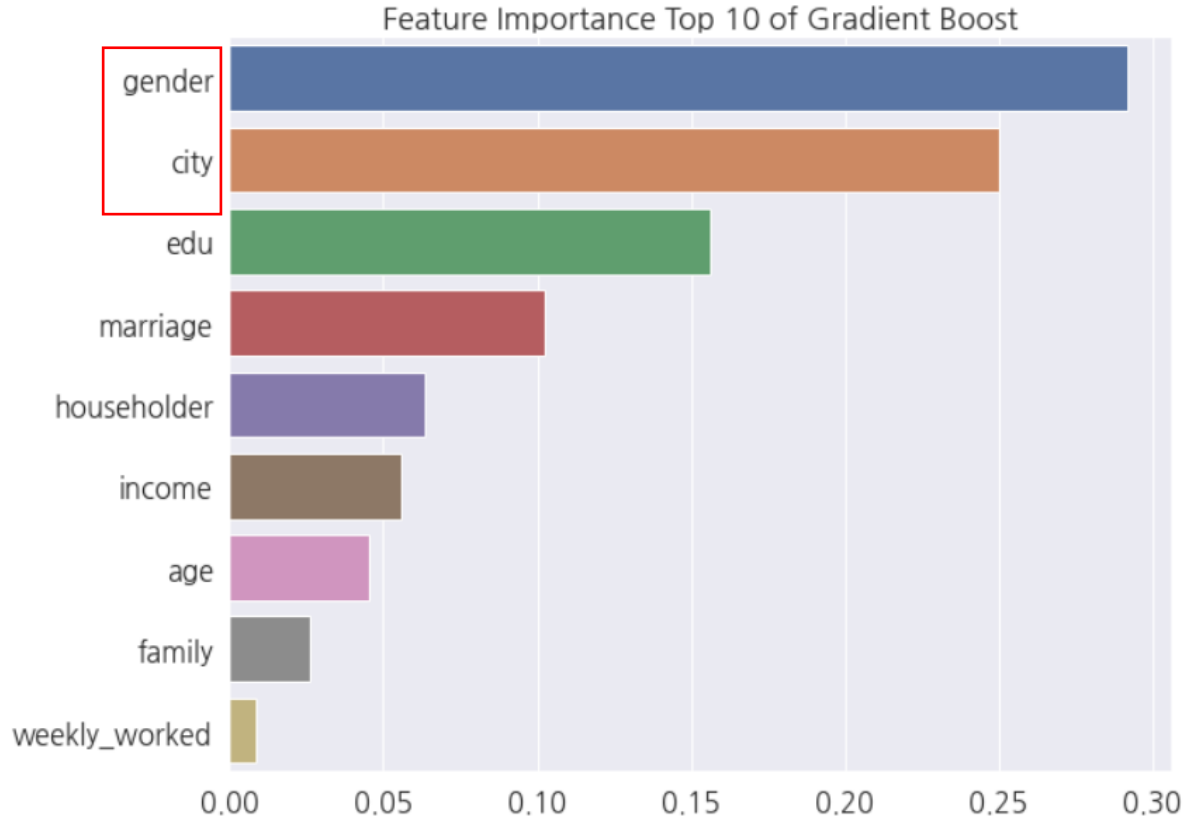
**정확도가 더 높은 그래디언트 부스팅 모형을 사용하기로 결정**

# 03 스포츠경험자 분류 – 변수 중요도

## 이진 분류모형 – 랜덤포레스트, 그래디언트 부스팅 비교

두 분류모형의 Feature Importance




두 모형 모두 성별, 거주지, 학력, 혼인여부를 중요한 변수로 선정 -> 주요 4개 변수와 연령 변수를 평가에 사용

# 03 스포츠경험자 분류 – 평가

## 이진 분류모형 – 그래디언트 부스팅 모형 평가

Train data / Test data 분리 = 7:3

5개의 명목형 변수 -> 더미변수화

[Grid Search 결과]

```
{'learning_rate': 0.7000000000000001,
 'max_depth': 4,
 'max_features': 1,
 'n_estimators': 15}
```

[최종 모형 성능]

```
Train accuracy score is 0.737
Test Accuracy for gradient boost is 0.726
```

Train accuracy score: 0.737

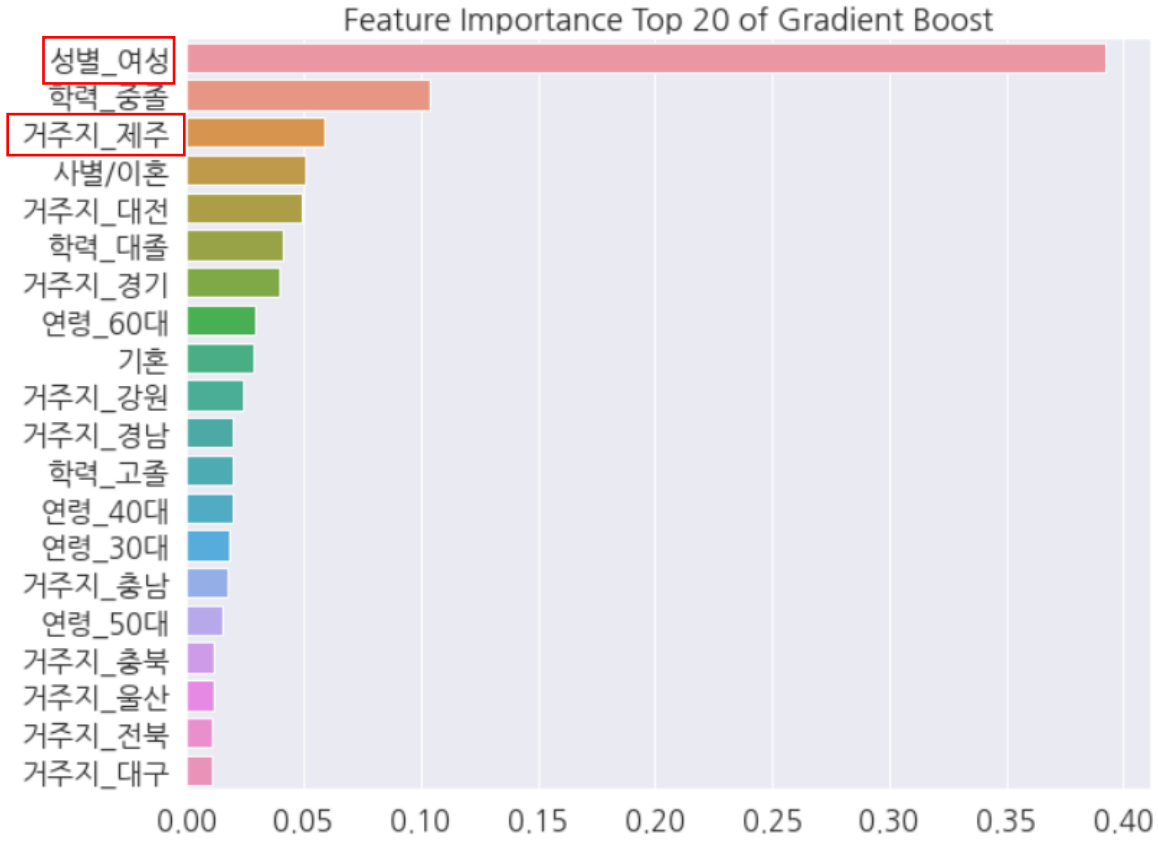
Test accuracy score: 0.726

```python
from sklearn.model_selection import train_test_split

df_gb = df[['gender', 'city', 'age', 'edu', 'target']]
train_gb, test_gb = train_test_split(df_gb, test_size = 0.3, random_state = 0, stratify = df['target'])
train_X_gb = train_gb[train_gb.columns[:-1]]
train_y_gb = train_gb[train_gb.columns[-1:]]
test_X_gb = test_gb[test_gb.columns[:-1]]
test_y_gb = test_gb[test_gb.columns[-1:]]

train_X_dummy_gb = pd.get_dummies(train_X_gb, columns = ['gender', 'city', 'age', 'edu'], drop_first = True)
test_X_dummy_gb = pd.get_dummies(test_X_gb, columns = ['gender', 'city', 'age', 'edu'], drop_first = True)

from sklearn.model_selection import GridSearchCV
from sklearn.ensemble import GradientBoostingClassifier

params = { 'n_estimators' : range(5, 50, 10),
           'max_depth' : range(3, 5),
           'learning_rate' : np.linspace(0.1, 1, 10),
           'max_features' : range(1, 4)
            }


gb_clf = GradientBoostingClassifier(random_state = 0)
grid_cv = GridSearchCV(gb_clf, param_grid = params, cv = 3, n_jobs = -1)
grid_cv.fit(train_X_dummy_gb, train_y_gb)

print('최적 하이퍼 파라미터: ', grid_cv.best_params_)
print('최고 예측 정확도: {:.4f}'.format(grid_cv.best_score_))

model = GradientBoostingClassifier(n_estimators = 15, learning_rate = 0.7,
max_depth = 4, max_features = 1, random_state = 0)
model.fit(train_X_dummy_gb, train_y_gb)
pred_y = model.predict(train_X_dummy_gb)
train_acc = accuracy_score(y_true = train_y_gb, y_pred = pred_y)
print('Train accuracy score is', round(train_acc, 3))
prediction_gb = model.predict(test_X_dummy_gb)
print('Test Accuracy for gradient boost is', round(metrics.accuracy_score(prediction_gb, test_y_gb), 3))
```

# 03 스포츠경험자 분류 – 최종 모형의 변수 중요도

## 이진 분류모형 – 그래디언트 부스팅 (변수 중요도)

중요변수만 더미변수화



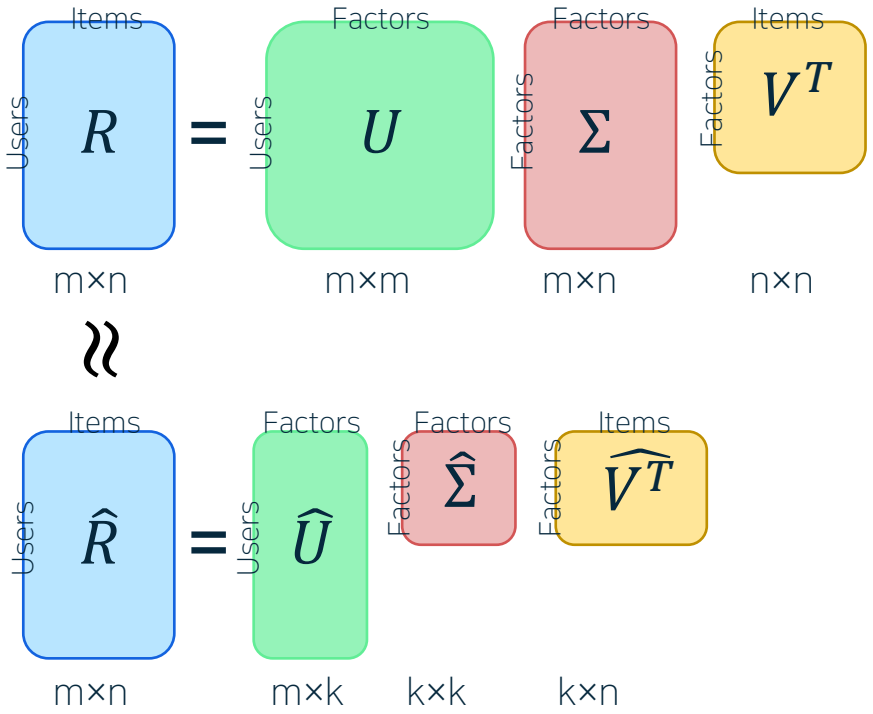

**성별**과 **거주지**가 스포츠경험자/비경험자를 분류하는 중요한 변수로 사용될 수 있음 확인

Test set에 대한 정확도: 0.726

# 04 스포츠 추천시스템 – 이론

## 추천 시스템 알고리즘

### SVD (특이값 분해)




Σ 행렬에서 k개의 대표 특이값만 사용해 차원을 축소하여 $\hat{\Sigma}$ 생성

분해된 행렬을 사용해 latent factors 계산이 가능

(latent factor: User, Item 각각의 특성)

### KNNBaseline (협업필터링 방법)

$$\hat{r}_{ui} = b_{ui} + \frac{\sum_{v \in N_i^k(u)} \text{sim}(u,v) \cdot (r_{vi} - b_{vi})}{\sum_{v \in N_i^k(u)} \text{sim}(u,v)}$$

$$\hat{r}_{ui} = b_{ui} + \frac{\sum_{j \in N_u^k(i)} \text{sim}(i,j) \cdot (r_{uj} - b_{uj})}{\sum_{j \in N_u^k(i)} \text{sim}(i,j)}$$

사용자(혹은 상품)의 유사도가 구해지면 유사도가 큰 k개의 사용자(혹은 상품)

벡터를 사용해 가중평균을 구하여 가중치를 예측

특히 KNNBaseline은 평점들을 베이스라인 모형의 값을 기준으로

가중 평균함

# 04 스포츠 추천시스템 – 적용 예시

## 추천 시스템 라이브러리 - Surprise

추천시스템을 위한 라이브러리

### 알고리즘

| Name | Description |
|---|---|
| NormalPredictor | 정규분포로 가정한 학습 데이터셋(trainset)의 평점(rating) 분포에서 랜덤하게 샘플링하는 알고리즘. |
| BaselineOnly | User와 Item의 Baseline을 이용한 평점 예측 알고리즘. |
| KNNBasic | 기본적인 협업필터링(Collaborative Filtering) 알고리즘. |
| KNNWithMeans | 기본적인 협업필터링(Collaborative Filtering) 알고리즘, 추가적으로 평균값을 더해준다. |
| KNNWithZScore | 기본적인 협업필터링(Collaborative Filtering) 알고리즘, 추가적으로 z-score 분포를 적용한다. |
| KNNBaseline | 기본적인 협업필터링(Collaborative Filtering) 알고리즘, 추가적으로 baseline을 더해준다. |
| SVD | 특이값 분해(SVD) 알고리즘, 넷플릭스 Prize에서 Simon Funk에 의해서 유명해짐. |
| SVD++ | 특이값 분해(SVD++) 알고리즘, 추가적으로 암시적 rating이 더해진다. |
| NMF | Non-negative 행렬분해 |

협업필터링(Collaborative Filtering) 알고리즘과 특이값 분해(SVD) 알고리즘을 주로 사용

### 대표적인 예: 넷플릭스 영화 추천 시스템

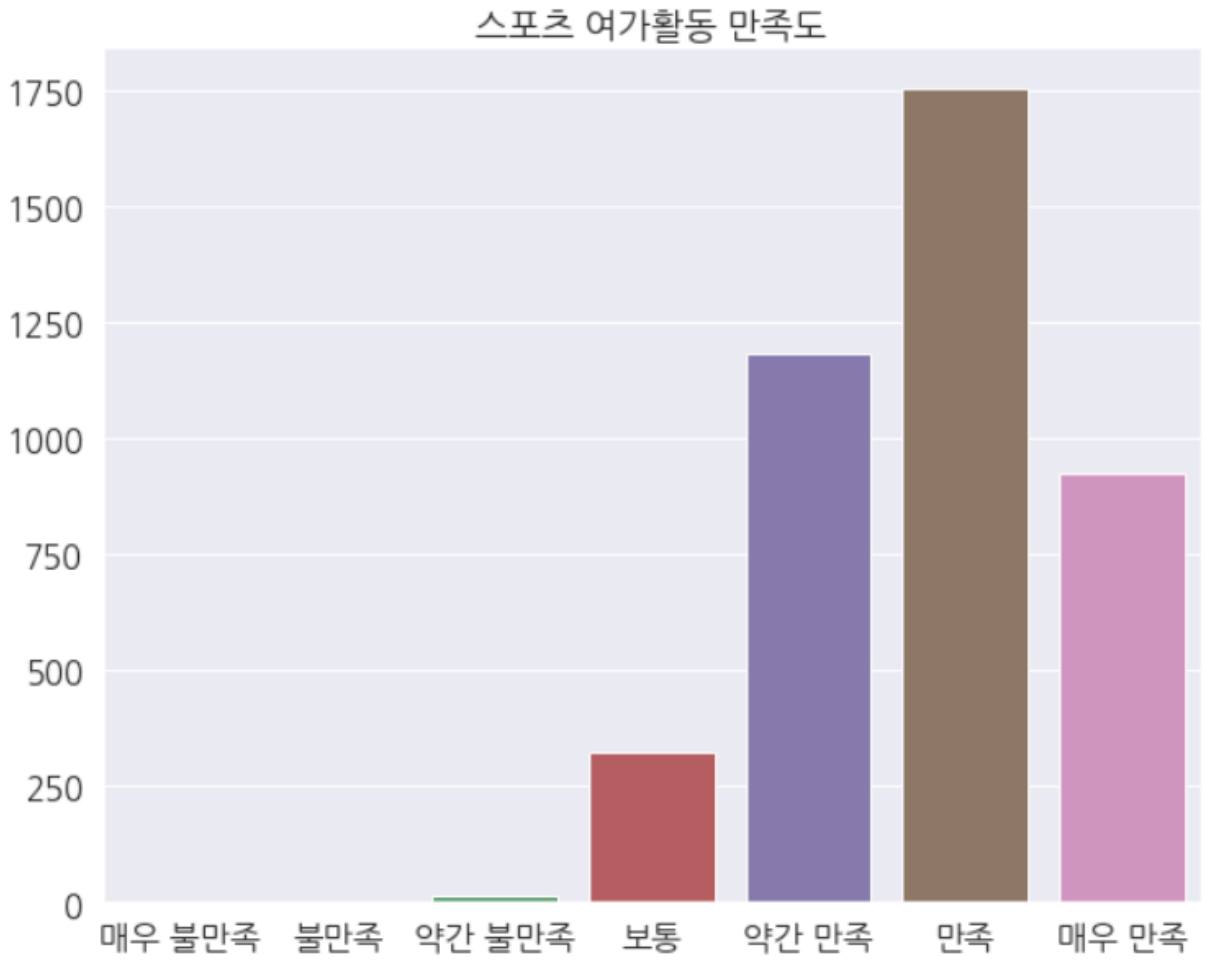
| item | 211 | 212 | 213 | 214 | 215 | 216 | 217 | 218 |
|---|---|---|---|---|---|---|---|---|
| user | | | | | | | | |
| 290 | 3 | | | | | 4 | | 2 |
| 291 | | 4 | | 4 | 4 | | | 4 |
| 292 | | | | 3 | | | | |
| 293 | 4 | | 3 | | 4 | 4 | 3 | 2 |

'User'(사용자)의 'Item'(영화)에 대한 'rating'(평점)을 평점 행렬로 만들고
사용자와 영화가 가진 특징 (Latent factor)을 찾아서 평점 예측
가장 높은 평점으로 예측되는 영화를 추천

추천 알고리즘을 신한카드 데이터에 적용
개인 사용자별 추천 스포츠를 제안하자!

# 04 스포츠 추천시스템 – 방법 설명

## 왜 추천 시스템에 신한카드 데이터를 사용하는가?




대다수의 사람들이 여가활동에 대해 객관적이기보다
긍정적인 평가를 내리는 경향성이 강함

-> 추천에 사용할 평점으로써 해당 점수를 반영하기에는
강한 편향때문에 부적합

평점을 나타낼 객관적이고 새로운 지표가 필요

평균취급액: 사람들이 평균적으로
돈을 쓸 의향이 얼마나 있는지 나타낼 수 있음

즉, 평균취급액 = 수요와 만족도의 지표

# 04 스포츠 추천시스템 – 방법 설명

## 신한카드 데이터에 추천 시스템을 적용하려면?

[문제점]

1. 주어진 신한카드 데이터는 개별 사용자 단위로 주어지지 않음
2. 스포츠에 대한 평점에 해당하는 값이 존재하지 않음

solution

[해결 방법]

1. 거주지, 성별, 연령 칼럼을 묶어 하나의 사용자로 가정 (3장 변수 중요도 기반)
2. 각 (거주지, 성별, 연령)의 스포츠별 총취급액과 총이용건수를 계산
3. "총취급액/총이용건수 = 평균취급액" 계산
4. 평균취급액의 구간을 동일한 길이의 다섯 구간으로 나눔
5. 각 구간을 1~5점으로 두어 평점으로 사용

[가정]

1. (거주지, 성별, 연령)으로 묶은 조합은 개인화된 특징을 갖는다.
2. 평균취급액이 높은 스포츠일수록 만족도가 높아서 소비액이 높다.

[한계점]

스포츠별로 발생하는 비용 특징 반영하지 않음 예) 고비용의 골프시설 등

[기존 신한카드 데이터]

| v1 | v2 | v3 | gb3 | gb2 | sex_ccd | cln_age_r | ta_ym | daw_ccd_r | apv_ts_dl_tm_r | vlm | usec | avg_spend |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 세종 | 세종 | . | 취미오락 | 종합쇼핑 | F | 30 | 2021-08-01 | WHITE | 휴식 | 325008700 | 12639 | 25715 |
| 세종 | 세종 | . | 취미오락 | 종합쇼핑 | M | 30 | 2021-12-01 | WHITE | 휴식 | 599872600 | 20805 | 28833 |
| 서울 | 세종 | . | 취미오락 | 종합쇼핑 | F | 50 | 2021-06-01 | WHITE | 활동 | 30363400 | 1075 | 28245 |
| 부산 | 세종 | . | 취미오락 | 외식 | F | 30 | 2022-01-01 | WHITE | 활동 | 6028600 | 428 | 14086 |
| 서울 | 세종 | . | 취미오락 | 인터넷게임 | F | 20 | 2021-08-01 | WHITE | 활동 | 630200 | 182 | 3463 |

전처리

[전처리한 데이터]

| userId | sportsId | rating |
|---|---|---|
| 부산F30 | 자전거 | 2 |
| 충북F60 | 레저스포츠 | 1 |
| 인천F30 | 헬스 | 4 |
| 울산F50 | 골프 | 4 |
| 전북M30 | 헬스 | 3 |
| ... | ... | ... |

→ '부산거주/여성/30대' 군집의 '자전거' 활동에 대한 평점은 '2'점 으로 해석

전처리한 데이터프레임을 추천 시스템에 적용

# 04 스포츠 추천시스템 – 데이터 전처리

## 신한카드 데이터 전처리

```
card = card.loc[card['gb3'] == '스포츠활동']
card['avg_spend'] = round(card['vlm'] / card['usec'], -1).astype(int)
index_to_drop = card.loc[(card['gb2'] == '스포츠용품구매') |
(card['gb2'] == '운동경기관람')].index
card = card.drop(index_to_drop, axis = 0)
q1, q3 = card['avg_spend'].quantile([0.25, 0.75])
iqr = q3 - q1
df = card.loc[card['avg_spend'] < q3 + 1.5 * iqr]
index_to_drop = df.loc[df['vlm'] == 0].index
df = df.drop(index_to_drop, axis = 0)
# 나이, 성별, 거주지, 종목명 별로 병합
new_df = df.groupby(['v1', 'sex_ccd', 'cln_age_r', 'gb2']).sum()
new_df = new_df.reset_index() # 멀티인덱스 해제
new_df['avg_spend'] = round(new_df['vlm'] / new_df['usec'], -1).astype(int)
new_df['cln_age_r'] = new_df['cln_age_r'].astype(str)
new_df['as_person'] = new_df['v1'] + new_df['sex_ccd'] + new_df['cln_age_r']

### 새로운 칼럼 'rating' 생성
# new_df['avg_spend'] 5개 구간으로 나눠서 레이팅 지표로 삼기
bins = pd.cut(new_df['avg_spend'], 5, labels = ['1', '2', '3', '4', '5'])
new_df['rating'] = bins

### SVD에 사용할 칼럼만 다시 추출
recommend_df = new_df[['as_person', 'gb2', 'rating']]
recommend_df.columns = ['userId', 'sportsId', 'rating']
recommend_df['rating'].astype(int)
```

'gb3' == '스포츠활동' 행 추출

'gb2' == '스포츠용품구매' or 'gb2' == '운동경기관람' 인 행 제거

'avg_spend' 칼럼 생성: 평균취급액 계산하여 칼럼으로 생성

'avg_spend' 중 q3 + 1.5*IQR 보다 큰 행 제거

'vlm' (취급액)이 0인 행 제거 (이상치로 판단)

칼럼 추출 및 병합

평균취급액 칼럼 다시 추가

'as_person' 칼럼 생성: 'userId'(사용자)로 사용

'rating' 칼럼 생성: 'avg_spend'(평균취급액)을 다섯 구간으로 나누어 각 구간을 평점으로 취급

추천시스템에 사용할 데이터프레임 생성

# 04 스포츠 추천시스템 - 데이터

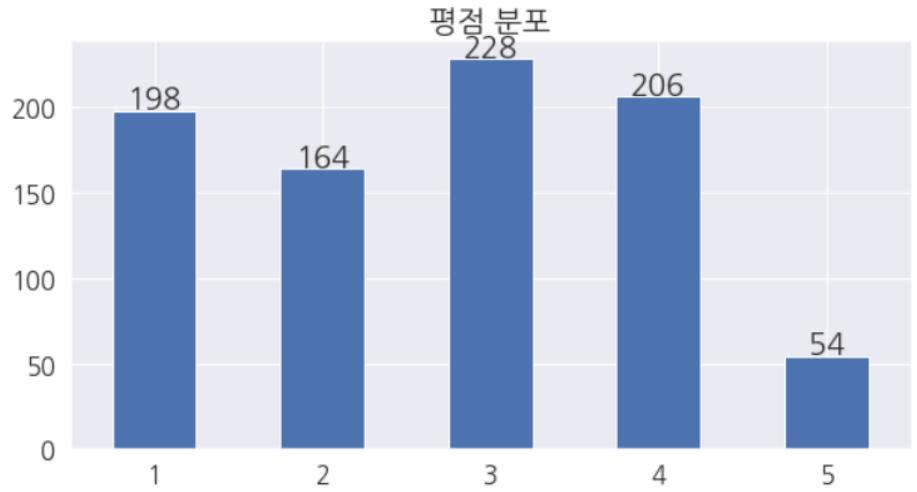
## 추천 시스템 - 적용 데이터

<전처리된 데이터>

| userId | sportsId | rating |
|---|---|---|
| 강원F20 | 골프 | 2 |
| 강원F20 | 레저스포츠 | 1 |
| 강원F20 | 스키 | 2 |
| 강원F20 | 자전거 | 1 |
| 강원F20 | 헬스 | 4 |
| ... | ... | ... |
| 충북M60 | 골프 | 4 |
| 충북M60 | 레저스포츠 | 1 |
| 충북M60 | 스키 | 2 |
| 충북M60 | 자전거 | 4 |
| 충북M60 | 헬스 | 4 |

850 rows x 3 columns

<rating 분포>




1점과 3점이 많은 다봉(multi-modal)분포

<신한카드 데이터 내 스포츠 >

```
card['gb2'].unique()
array(['레저스포츠', '헬스', '골프', '자전거', '스키'],
```

레저스포츠, 헬스 골프, 자전거, 스키

<데이터 형식>

```
reader = Reader(rating_scale = (1, 5))
data = Dataset.load_from_df(recommend_df[[
    'userId', 'sportsId', 'rating']], reader = reader)
```

surprise 라이브러리에서 제시한 데이터 형식 사용

**추천 시스템 적용 목표**

사용자 유사도를 기반으로 특정 사용자에게 평점이 가장 높을 것으로 예상되는 종목 추천

# 04 스포츠 추천시스템 – 모형 평가

## 추천 시스템 – 그리드 서치

### [SVD]

```
### 그리드서치로 SVD 모형의 최적의 파라미터 찾기
from surprise import SVD, accuracy # SVD model, 평가
from surprise.model_selection import GridSearchCV

param_grid = {'n_factors': [25, 50, 75, 100],
'lr_all': np.linspace(0.01, 0.1, 10),
'reg_all': np.linspace(0.01, 0.1, 10)}
gs = GridSearchCV(algo_class = SVD, measures = ['RMSE'], param_grid = param_grid)
gs.fit(data)

print('Best RMSE Score: ', gs.best_score['rmse'])
print('Best Parameters: ', gs.best_params['rmse'])
```

```
Best RMSE Score:  0.7693038295619763

Best Parameters:  {'n_factors': 50, 'lr_all': 0.1, 'reg_all': 0.020000000000000004}
```

n_factors = 50, lr_all = 0.1, reg_all  0.02

RMSE: 0.769

```
from surprise import accuracy
model = SVD(n_factors = 50, lr_all = 0.1, reg_all = 0.02)
model.fit(trainset)
predictions = model.test(testset)
print('SVD RMSE:', accuracy.rmse(predictions))
```

```
RMSE: 0.7575

SVD RMSE: 0.7574913511318275
```

Test set RMSE: 0.757

### [KNNBaseline]

```
param_grid = {'bsl_options': {'method': ['als', 'sgd'],
                              'reg': [1, 2]},
              'k': range(15, 40),
              'verbose': [False],
              'sim_options': {'name': ['msd', 'cosine', 'pearson_baseline'],
                              'min_support': [1, 5],
                              'user_based': [True]}
              }
gs = GridSearchCV(KNNBaseline, param_grid, measures = ['rmse', 'mae'], cv = 7)

gs.fit(data)
```

```
Best RMSE Score:  0.7352993920289499

{'bsl_options': {'method': 'als', 'reg': 1}, 'k': 28, 'verbose': False,

 'sim_options': {'name': 'pearson_baseline', 'min_support': 1, 'user_based': True}}
```

bsl_options = {'method' : 'als', 'reg' : 1}, k = 28,

sim_options = {'name' : 'pearson_baseline', 'min_support': 1, user_base' :[True]}

RMSE: 0.735

```
algo = KNNBaseline(
    sim_options = {'name': 'pearson_baseline', 'min_support': 1, 'user_based':[True]},
    k = 28,
    bsl_options = {'method': 'als', 'reg': 1},
    verbose = False)

algo.fit(trainset)
predictions = algo.test(testset)
print('KNNBaseline RMSE:', accuracy.rmse(predictions))
```

```
RMSE: 0.7191

0.7190972607150229
```

Test set RMSE: 0.719

# 04 스포츠 추천시스템 - 적용

**추천 시스템 - 적용**

[예] 서울에 사는 30대 남성에게 가장 추천하는 스포츠는?

KNNBaseline

<KNNBaseline 모형이 서울 에 사는 30 대 남성 에게 추천하는 종목>

골프 에 대한 예상 평점: 3.1

레저스포츠 에 대한 예상 평점: 1.0

스키 에 대한 예상 평점: 2.3

자전거 에 대한 예상 평점: 3.0

헬스 에 대한 예상 평점: 4.9

가장 추천하는 운동 종목은 헬스 , 그 평점은 4.9 로 예상됩니다.

<실제 평점>

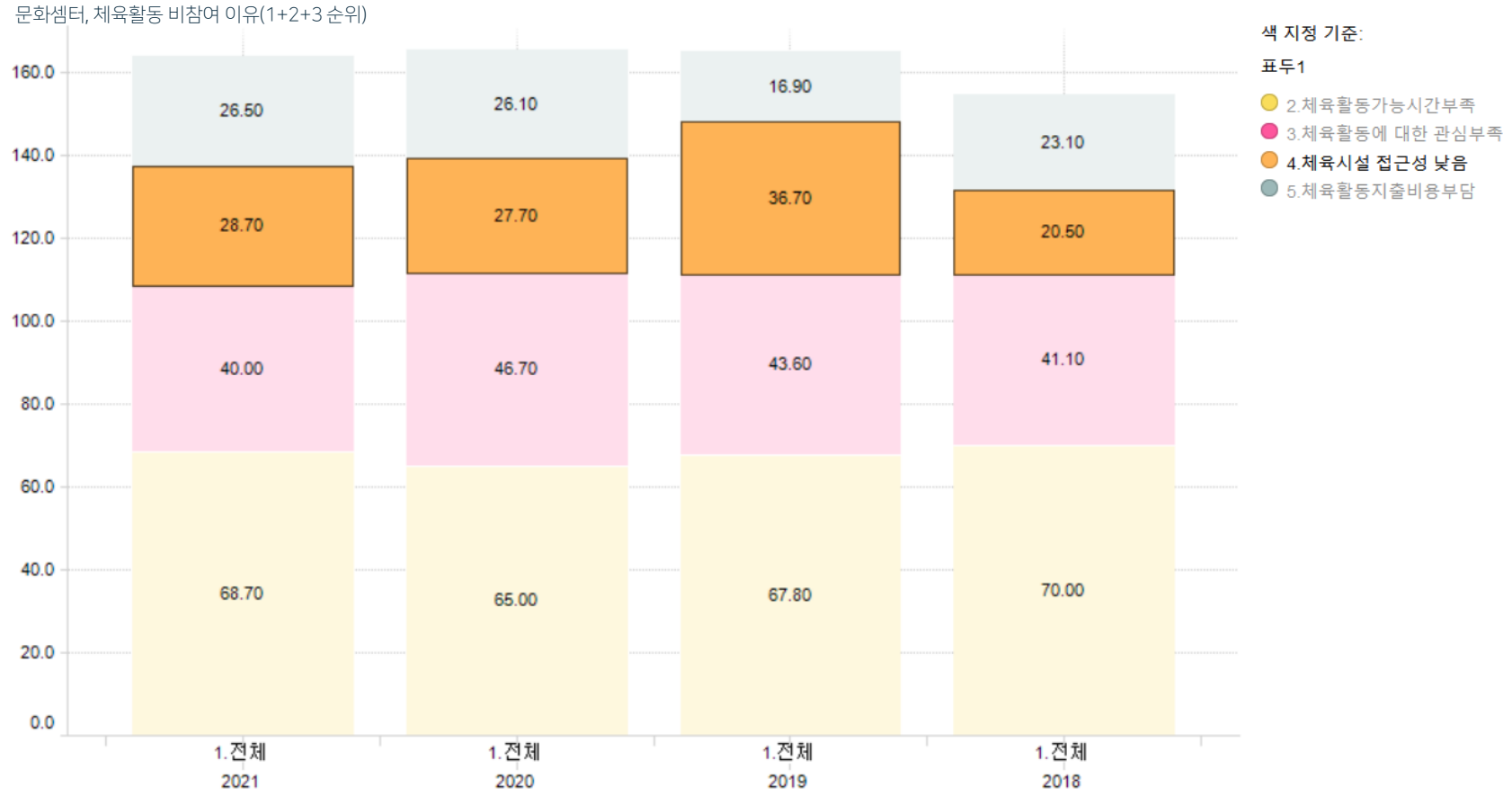
| | userId | sportsId | rating |
|---|---|---|---|
| 430 | 서울M30 | 골프 | 3 |
| 431 | 서울M30 | 레저스포츠 | 1 |
| 432 | 서울M30 | 스키 | 2 |
| 433 | 서울M30 | 자전거 | 3 |
| 434 | 서울M30 | 헬스 | 5 |

서울에 사는 30대 남성에게는 “헬스”를 추천 → 스포츠비경험자에게 맞춤형 스포츠로 추천

# 05 스포츠 기피 이유 분석

## 스포츠 기피 이유 분석

<체육활동 비참여 이유 (1+2+3 순위)>

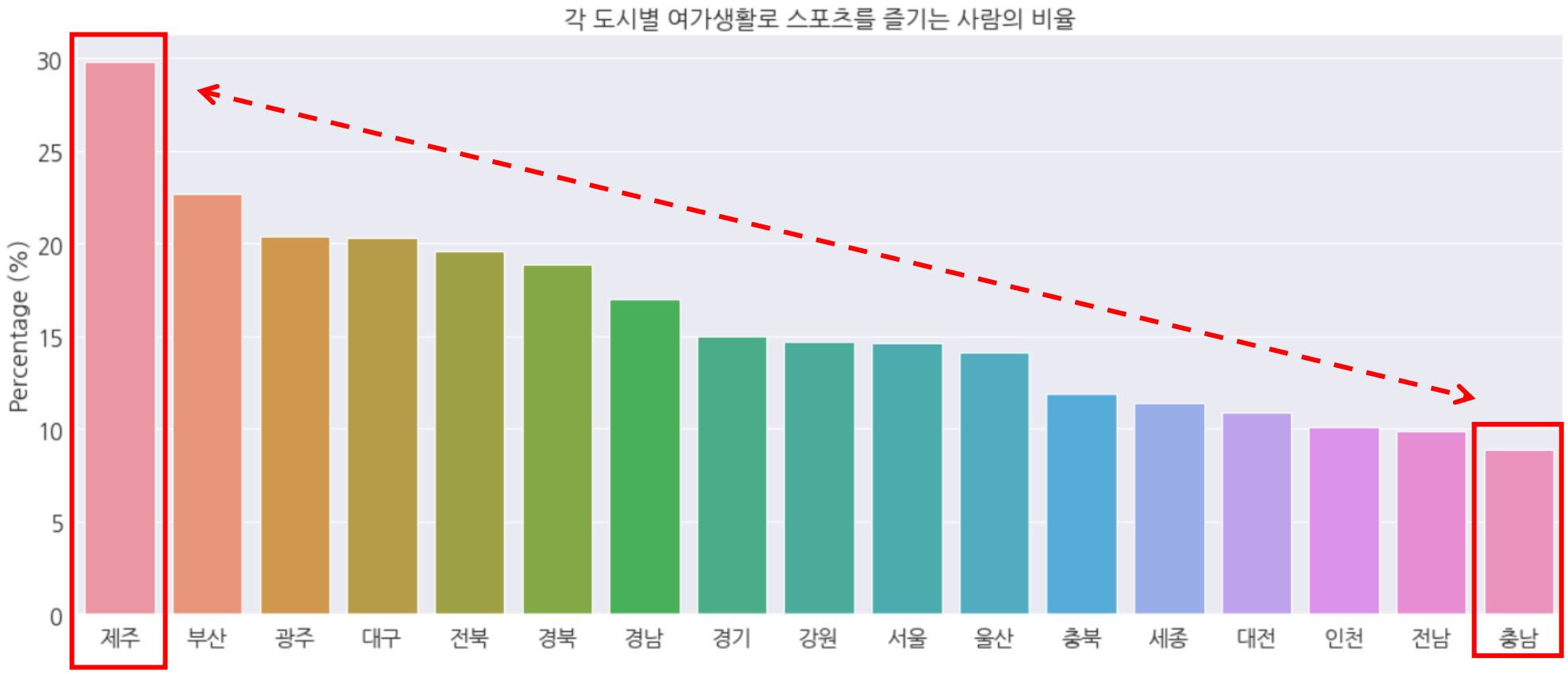



1위 시간 부족 2위 관심 부족 3위 체육시설 접근성 낮음

# 05 스포츠 기피 이유 분석

## 스포츠 기피 이유 분석

지역별 인구 중 스포츠 여가활동을 꾸준히 즐기는 인구의 비율 (국민여가활동조사 기반)



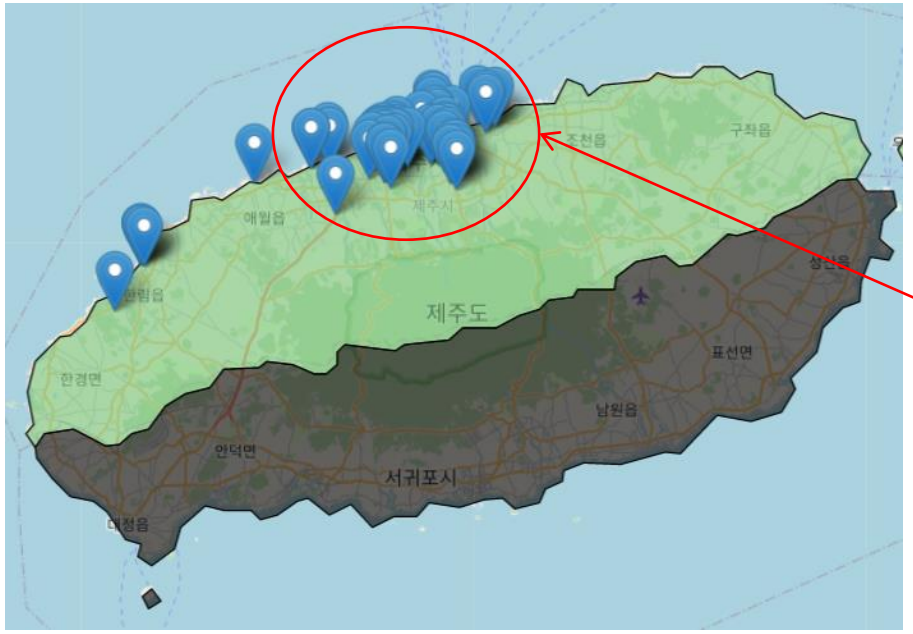

제주도가 29.8%로 가장 많고, 충남이 8.9%로 스포츠인이 가장 적었음

-> 지역별로 차이가 큰 이유는?

# 05 스포츠 기피 이유 분석

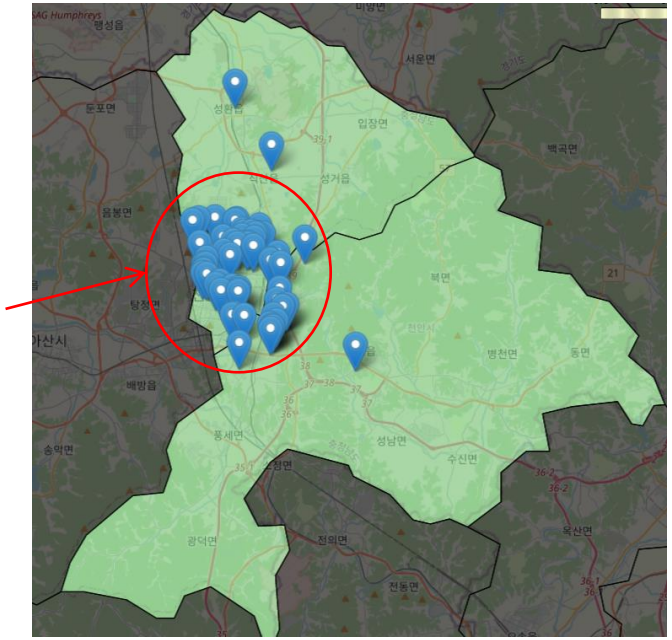
**대표 지역별 체육시설 시각화 – 헬스장** (제주도, 충청남도 내 가장 인구가 많은 지역 기준)

제주시




중심지에 집중적으로 분포

충남 천안시




인구: 493,538명

헬스장 수: 56곳

8,813명/헬스장

2,000명 차

인구: 656,702명

헬스장 수: 61곳

10,765명/헬스장

1. 두 도시 모두 헬스장이 중심에만 집중적으로 분포되어 있음

2. 체육시설 대비 인구수의 지역별 격차가 큼

# 06 결론 및 활용방안

## 1. 스포츠 여가활동 경험자와 비경험자 분류

- 스포츠경험여부와 스포츠 여가활동 희망 여부에 유의한 상관관계 확인
  - 스포츠비경험자에게 스포츠 경험을 유도한다면 스포츠 여가활동을 희망할 가능성이 큼

- 스포츠경험자/비경험자를 분류하는 중요한 변수는 "성별"과 "거주지"
  - 스포츠비경험자의 특징을 이용해 맞춤형 스포츠 추천에 활용

## 2. 스포츠비경험자를 위한 스포츠 추천 시스템

- (지역, 성별, 연령대) 군집별 스포츠에 대한 평점 예측 가능
  - 스포츠 추천 시스템을 지역주민에게 제공함으로써 스포츠 경험 증가와 스포츠 여가활동 수요 증대 기대
  - 지방자치단체에서 지역별 주민 특성을 확인하여 가장 많이 추천되는 스포츠에 투자할 수 있도록 활용 가능

# 06 결론 및 활용방안

## 3. 스포츠 여가활동 기피 이유 분석

- 제주시와 천안시의 스포츠 여가활동 인구 비율의 격차가 큰 이유 확인
  - 체육시설 대비 인구수 격차가 크기 때문으로 유추
  - 체육시설 대비 인구수가 많은 천안시의 경우 체육시설 확충 필요

- 제주시와 천안시 모두 체육시설이 중심지 위주로 분포
  - 중심지 외 지역의 접근성 확보로 스포츠 여가활동 기피 이유 해소

- 체육시설 접근성 확보를 통한 스포츠 활동 참여율 증가 기대

DATA 133

# 감사합니다