# KBO 경기 별 득점 및 승률에 대한 통계적 분석

2022. 09.

이 영 찬

# 목 차

# Ⅰ. 서론

## 1. 개요

우리나라와 일본이 축구를 할 때 우리의 관심사는 무엇일까. 우리나라가 몇 골을 넣고 이기느냐가 아닐까. 그렇다, 스포츠 팬 입장에서 가장 큰 관심사는 우리 팀이 몇 점을 득점할지, 이길 확률이 얼마나 되는지와 같은 이야기이다.

프로젝트에서는 **한국프로야구 경기의 득점 및 승률 예측**을 주제로 매 경기 팀의 득점 분포와 승률을 예측하는 것을 목표한다. 예측한 득점과 승률은 경기 시작 전 주어지는 도박사의 예측과 비교한다.

**한국프로야구 경기의 점수 및 승률 예측**으로 주제를 선정한 이유는 크게 세가지이다.

### 1) 실용성

데이터분석에 들어가는 시간과 비용은 실용적이어야 한다. 주식 매매에 시계열 분석을 적용하고, 스포츠 경기 결과를 예측하는 것은 외부환경에서 일반인들이 할 수 있는 가장 실용적인 목적이다.

### 2) 데이터 수집

데이터 분석을 하려면 원하는 데이터를 확보해야한다. 파이썬 라이브러리를 활용하여 KBO사이트에 제공되는 야구 경기 결과를 크롤링 할 수 있었다.

### 3) 도메인지식

도메인지식은 데이터분석을 특별하게 만들어준다. 데이터를 전처리하고, 모형을 만드는 작업은 현실세계를 수학적으로 나타내는 과정이기 때문에, 현실세계에 대한 이해도가 클수록 좋은 데이터 분석을 할 수 있다. 2010년 이후 항상 관심을 가졌던 세이버매트릭스에 대한 지식을 통해 더 좋은 데이터 분석을 목표한다.

## 2. 프로세스

프로젝트는 데이터 수집부터 저장, 전처리, 분석, 결과 비교 및 송출까지 데이터분석의 모든 과정을 포함한다.


### 1) Data Crawling & My-SQL

KBO 홈페이지(www.koreabaseball.com)에 제공되는 야구 경기 결과 데이터를 파이썬 라이브러리 Selenium과 BeautifulSoup을 사용해 크롤링하여 My-SQL에 저장한다.

### 2) Precleaning & EDA

저장한 raw-data를 분석에 용이하도록 전처리한다. 도메인지식과 EDA를 통해 데이터 분석의 방향성을 정한다.

### 3) Modeling & Analysis

분석에 사용할 변수와 모형을 정하고 모델링한다. 퍼포먼스 비교를 통해 최적의 모형, 변수와 파라미터를 정한다.

### 4) Predict & Evaluate

경기 전 주어지는 정보를 예측 모형에 넣어 득점과 승률을 예측한다. 예측한 결과를 도박사들이 제시한 득점 및 승률과 비교하여 퍼포먼스를 검증한다.

### 5) AWS & GitHub

예측결과를 Django를 통해 개발한 홈페이지에 업로드 한다. 분석에 사용한 Python 코드는 Github에 업로드 한다.

**AWS:** 15.164.213.230 // **Github:** http://github.com/dldudcks91

# Ⅱ. 데이터 수집 및 구조

## 1. 데이터 수집

KBO(www.koreabaseball.com) 홈페이지에 제공되는 2017~2022년 경기 결과를 크롤링 하였다. 크롤링한 raw-data는 아래와 같다(Table1). 스코어보드는 이닝 당 팀 득점, 타자 기록은 타석 별 기록, 투수기록은 선발 투수와 불펜투수의 경기 기록으로 구성된다. 2022년 데이터는 경기가 끝나면 크롤링해 저장한다.

**Table1. Raw-data 형태**

**스코어 보드**

구장 : 잠실 관중 : 2,919 개시 : 18:30 종료 : 21:21 경기시간 : 2:51

| | TEAM | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | R | H | E | B |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 승 | 42승 27패 0무 | 2 | 3 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | - | - | - | 6 | 8 | 0 | 4 |
| 패 | 41승 31패 0무 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | - | - | - | 1 | 7 | 1 | 2 |

**타자기록**

LG 트윈스 타자 기록

| | | 선수명 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 타수 | 안타 | 타점 | 득점 | 타율 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 중 | 홍창기 | 중비 | | 삼진 | | 중안 | | 투안 | | | 4 | 2 | 0 | 0 | 0.339 |
| 2 | 좌 | 이형종 | 1파 | | 중안 | | 중비 | | 좌비 | | | 4 | 1 | 0 | 0 | 0.214 |
| 3 | 지 | 김현수 | 삼진 | | 중비 | | 포비 | | | 1땅 | | 4 | 0 | 0 | 0 | 0.301 |
| 4 | 우 | 채은성 | | 중비 | | 좌2 | | 1땅 | | 유땅 | | 4 | 1 | 0 | 1 | 0.315 |
| 4 | 우 | 김용의 | | | | | | | | | | 0 | 0 | 0 | 0 | 0.143 |
| 5 | 유 | 오지환 | | 중비 | | 2땅 | | 유파 | | | | 3 | 0 | 0 | 0 | 0.235 |
| 5 | 二 | 정주현 | | | | | | | | 3땅 | | 1 | 0 | 0 | 0 | 0.227 |
| 6 | 一 | 문보경 | | 4구 | | 삼진 | | 삼진 | | | 좌비 | 3 | 0 | 0 | 0 | 0.262 |
| 7 | 三 | 김민성 | | 4구 | | | | | | | | 0 | 0 | 0 | 0 | 0.199 |
| 7 | 三 | 이상호 | | | | 우중안 | | | 유직 | | 우중안 | 3 | 2 | 1 | 0 | 0.667 |
| 8 | 포 | 유강남 | | 2비 | | 2땅 | | | 삼진 | | | 3 | 0 | 0 | 0 | 0.250 |
| 8 | 포 | 김재성 | | | | | | | | | 우비 | 1 | 0 | 0 | 0 | 0.120 |
| 9 | 二유 | 이영빈 | | | 삼진 | | 삼진 | | 중안 | | 유땅 | 4 | 1 | 0 | 0 | 0.353 |
| | | | | | | | TOTAL | | | | | 34 | 7 | 1 | 1 | 0.253 |

**투수기록**

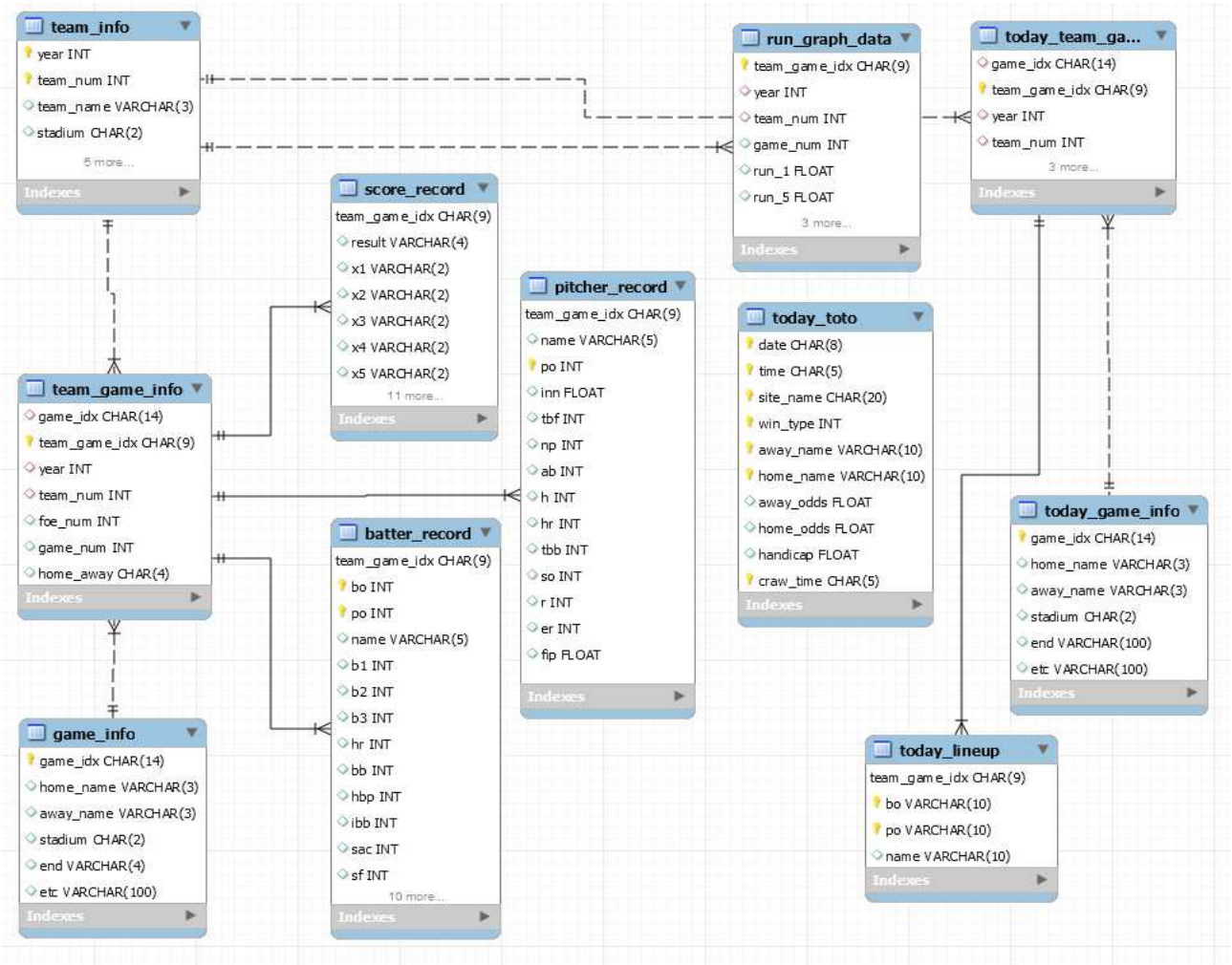
LG 트윈스 투수 기록

| 선수명 | 등판 | 결과 | 승 | 패 | 세 | 이닝 | 타자 | 투구수 | 타수 | 피안타 | 홈런 | 4사구 | 삼진 | 실점 | 자책 | 평균자책점 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 임찬규 | 선발 | 패 | 1 | 3 | 0 | 6 | 29 | 87 | 25 | 7 | 1 | 3 | 4 | 6 | 3 | 7.64 |
| 이우찬 | 7.3 | | 0 | 0 | 0 | 2 | 7 | 31 | 6 | 1 | 0 | 1 | 1 | 0 | 0 | 3.94 |
| 이상규 | 9.1 | | 0 | 0 | 0 | 1 | 3 | 12 | 3 | 0 | 0 | 0 | 1 | 0 | 0 | 0.00 |
| TOTAL | | | | | | 9 | 39 | 130 | 34 | 8 | 1 | 4 | 6 | 6 | 3 | 3.00 |

## 2. 데이터 구조

크롤링한 raw-data를 분석에 용이하도록 전처리해 My-sql에 저장하였다. 팀 정보, 경기 정보, 경기 기록 등의 테이블로 나누어 저장하였고 ERD는 아래와 같다(Figure 1).

**Figure 1. Baseball DB - ERD**




## 3. 데이터 요약

**Table2. Raw-data 요약**

| Name | Raw | Columns | 설명 |
|---|---|---|---|
| **Score_record** | 7,200~ | 25 | 스코어보드 |
| **Batter_record** | 91,737~ | 20 | 타자기록 |
| **Pitcher_record** | 32,117~ | 11 | 투수기록 |

# Ⅲ. 변수정의

## 1. 시작 전 정보

**FIgure 2. 경기 시작 전 주어지는 정보**


롯데

| Number | Position | Name | FIP/XR |
|---|---|---|---|
| 0 | 선발투수 | 박세웅 | |
| 1 | 유격수 | 마차도 | |
| 2 | 우익수 | 손아섭 | |
| 3 | 지명타자 | 이대호 | |
| 4 | 2루수 | 안치홍 | |
| 5 | 좌익수 | 전준우 | |
| 6 | 중견수 | 추재현 | |
| 7 | 1루수 | 정훈 | |
| 8 | 3루수 | 나승엽 | |
| 9 | 포수 | 안중열 | |

LG

| Number | Position | Name | FIP/XR |
|---|---|---|---|
| 0 | 선발투수 | 임찬규 | |
| 1 | 중견수 | 홍창기 | |
| 2 | 좌익수 | 김현수 | |
| 3 | 2루수 | 서건창 | |
| 4 | 1루수 | 보어 | |
| 5 | 3루수 | 문보경 | |
| 6 | 지명타자 | 이형종 | |
| 7 | 우익수 | 이재원 | |
| 8 | 포수 | 김재성 | |
| 9 | 유격수 | 오지환 | |

경기 시작 전 주어지는 정보에 기반하여 모형을 만들어 예측하는 것을 목표한다. 경기 시작 전 주어지는 정보(Figure 2)는 **선발투수, 선발 타자 라인업, 구장, 경기시간** 등이다.

공/수가 분리된 야구는 선수들의 기록이 공격과 수비로 명확하게 구분된다. 공격과 수비를 대표하는 변수를 생각하고 득점에 영향을 미치는 또 다른 변수와 최적의 예측을 위한 전처리를 생각한다.

**득점(A팀) = 공격력(A팀) + 수비력(B팀) + 기타 변수(홈&원정, 구장, 상대성...)**

## 2. 변수정의

### 1) 공격력

공격력을 대표할 지표를 생각해보자. 과거 득점을 사용하는 것이 쉽고 좋은 방법이 될 수 있으나 매 경기 바뀌는 라인업을 반영할 수 없다. 경기 전 주어지는 선발 라인업을 반영하기 위해 타자들의 과거 타격기록을 공격력 변수로 사용한다(Table3).

**Table 3. 타자의 타격기록**

| 변수명 | 설명 | 변수명 | 설명 |
|---|---|---|---|
| **BO** | 타순 | **SAC** | 희생타 |
| **Name** | 이름 | **SF** | 희생플라이 |
| **Run** | 득점 | **SO** | 삼진아웃 |
| **1B** | 1루타 | **GO** | 그라운드아웃 |
| **2B** | 2루타 | **FO** | 플라이아웃 |
| **3B** | 3루타 | **GIDP** | 병살타 |
| **HR** | 홈런 | **ETC** | 기타 |
| **BB** | 볼넷 | **H** | 안타 |
| **HBP** | 몸에맞는공 | **AB** | 타수 |
| **IBB** | 고의사구 | **PA** | 타석 수 |

### 2) 수비력

수비력 지표는 투수들의 기록을 사용하였다. 투수들의 기록은 선발투수와 중간계투의 투구기록으로 나뉘어진다. 투수들의 투구기록은 다음과 같이 정리된다(Table 4).

**Table 4. 투수의 투구기록**

| 변수명 | 설명 | 변수명 | 설명 |
|---|---|---|---|
| **PO** | 등판순서 | **H** | 피안타 |
| **Name** | 이름 | **HR** | 피홈런 |
| **INN** | 이닝 | **TBB** | 4사구 |
| **TBF** | 상대타자 수 | **SO** | 삼진 |
| **NP** | 투구 수 | **ER** | 자책점 |
| **AB** | 타수 | **R** | 실점 |

### 3) 기타변수

#### ① 홈/원정

홈/원정은 스포츠에서 중요한 요소 중 하나로 꼽힌다. 일반적으로 신체, 심리, 심판판정등 다양한 이유로 홈팀이 유리한 것으로 알려져 있다.

#### ② 구장별 차이

야구 구장은 구장별로 다양한 크기와 형태를 가진다. 같은 타구여도 구장 크기와 형태, 펜스 높이에 따라 결과가 달라지기 때문에 구장에 따른 득점의 유, 불리가 존재한다.

#### ③ 상대성

스포츠에선 리그 최고의 강팀이 라이벌 구단에게 덜미를 잡히거나 전술 간의 상대성으로 어려움을 겪기도 한다. 또한, 야구에선 좌타자가 우투수에 강하다는 오래된 속설이 있다.

#### ④ 분위기

스포츠에서 분위기는 매우 중요하다. 분위기를 어떻게 표현할 수 있을까.

### 4) 누적데이터

시즌 N번째 경기를 예측할 때 N-1의 과거 데이터가 주어진다. N-1의 과거 데이터 중 몇 번의 기록을 사용할 것인지, 시점에 따른 가중치를 줄 것인지에 대한 정리가 필요하다.

#### ① 최근 n경기

최근 n경기를 선택하는 것은 시계열 이동평균과 Kernel-regression bandwidth를 정하는 과정과 비슷하다. n이 너무 크면 최근 정보를 담지 못하고 n이 너무 작으면 운에 좌우된다.

#### ② 가중치

최근 9경기 데이터를 다음 경기 예측에 사용한다고 해보자. 첫 3경기와 최근 3경기를 동일하게 바라볼 것이냐는 가중치 문제가 생긴다. 프로젝트에서는 Kernel-regression에서 사용하는 커널함수 Uniform-kernel과 Epanechnikov-kernel을 사용하였다. 커널 함수는 적분해서 1이 나오는 가중함수로 프로젝트에서는 기준점의 왼쪽 데이터만을 사용하기 때문에 약간의 수정 후 사용하였다.

$$K(u) = \frac{3}{2}(1-u^2), \ -1 \leq u \leq 0$$

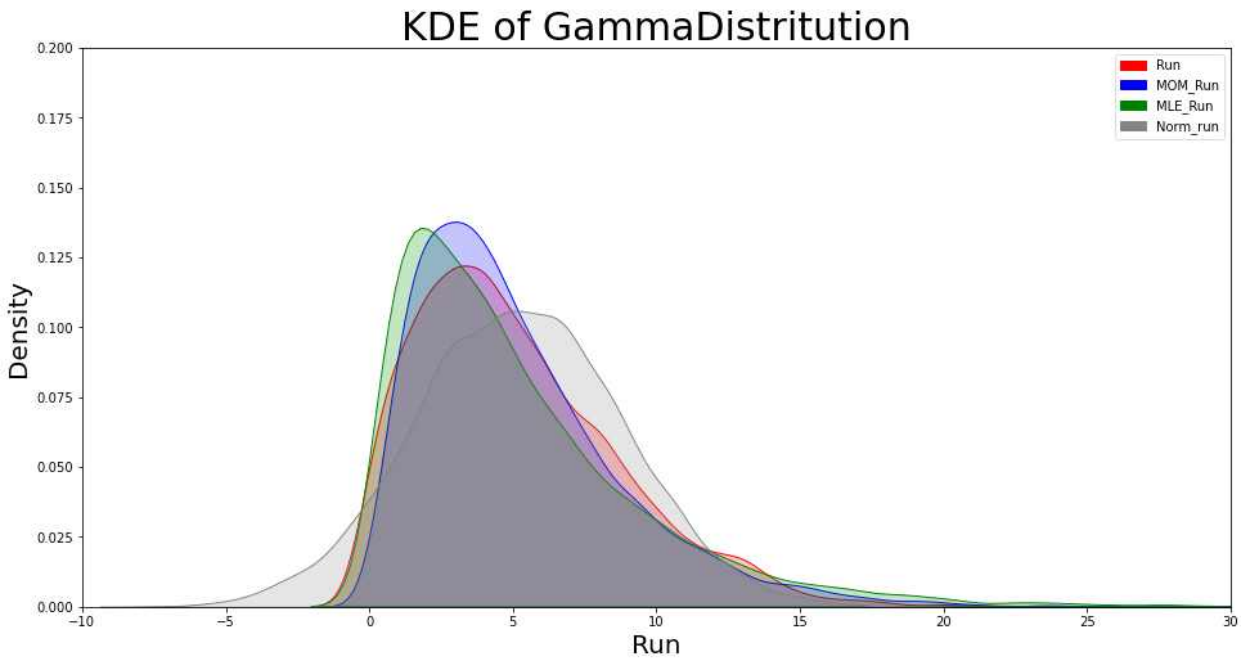
# Ⅳ. 탐색적 분석

## 1. 득점의 분포

### 1) 감마분포

야구경기에서 득점이 가지는 분포를 알아보자. 득점이 가지는 분포를 알 수 있다면 시뮬레이션을 통해 두 팀의 승률을 쉽게 비교할 수 있다. 득점이 감마분포, 정규분포를 따른다고 가정한 뒤 MOM, MLE를 통해 추정한 모수를 따르는 분포의 난수를 생성하고 KDE를 통해 실제 득점분포와 비교해 보았다(Figure 3).

**Figure 3. MOM, MLE로 추정한 모수를 따르는 감마분포와 실제 득점 분포 비교**




득점 분포(빨강)가 감마분포에 근사하다는 것을 알 수 있다. 또한 득점이 0이상의 값을 가지기 때문에 감마분포를 가정하는 것이 타당해보인다.

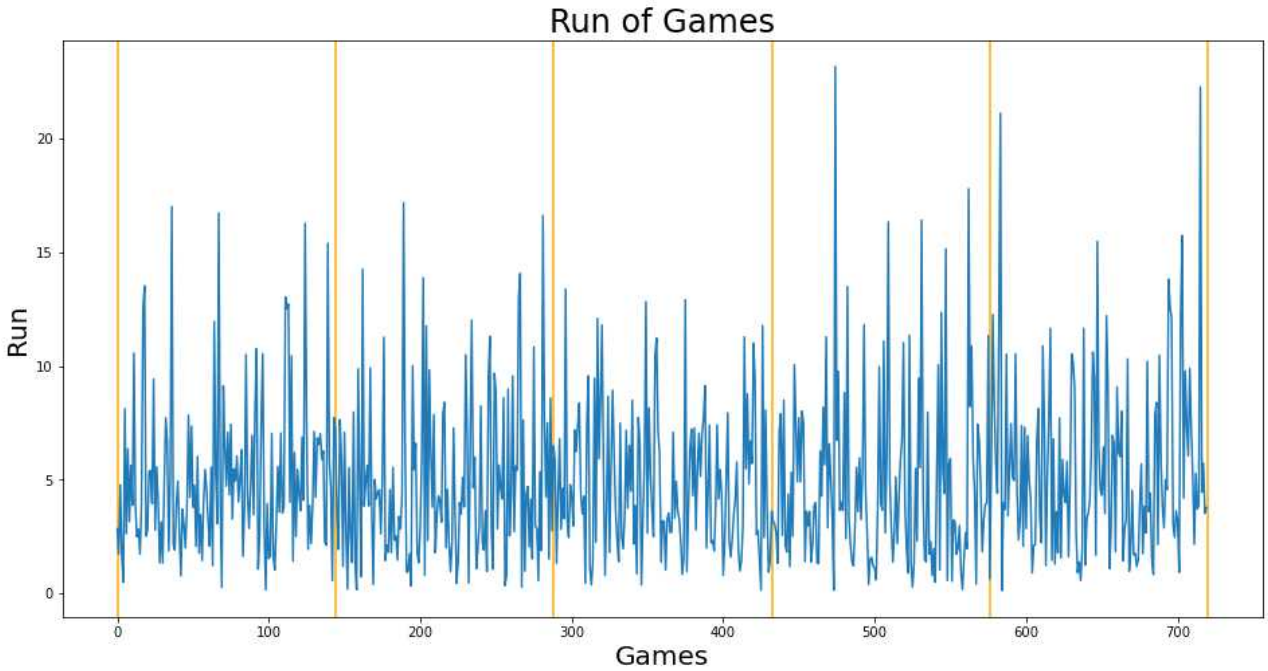
득점이 감마분포를 따른다고 가정하면 시뮬레이션을 통해 두 팀의 승률을 쉽게 비교할 수 있다. 득점 분포를 감마분포로 가정하고 모수(α, β)를 추정해 승률을 예측한다.

## 2. 시계열

### 1) 기본가정

야구 데이터는 시간의 흐름에 따라 얻어지는 시계열 데이터이다. 정상성을 가지는지 확인하기 위해 추세와 계절성 등 시계열 데이터의 기본 가정을 확인해 보았다(Figure 4).

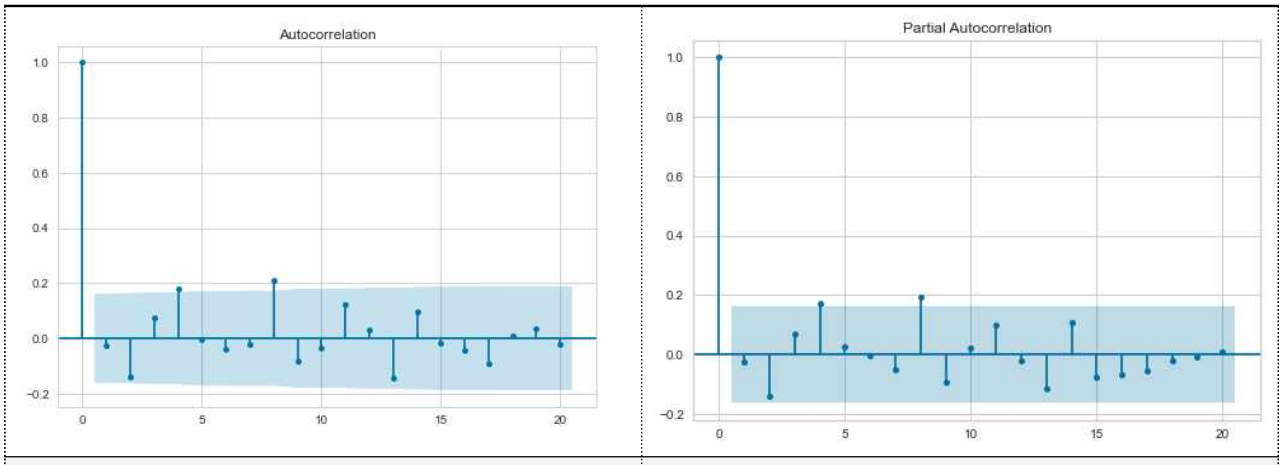
**FIgure 4. 2017~2021년 LG의 득점(720경기)**




득점의 평균과 분산에 특별한 추세와 계절성이 보이지 않는다.

다음은 과거 데이터와의 상관성을 보기 위해 acf, pacf그래프를 확인해보았다(Figure 5).

**FIgure 5. 2017년 LG 득점에 대한 acf, pacf**




| acf | pacf |
|---|---|

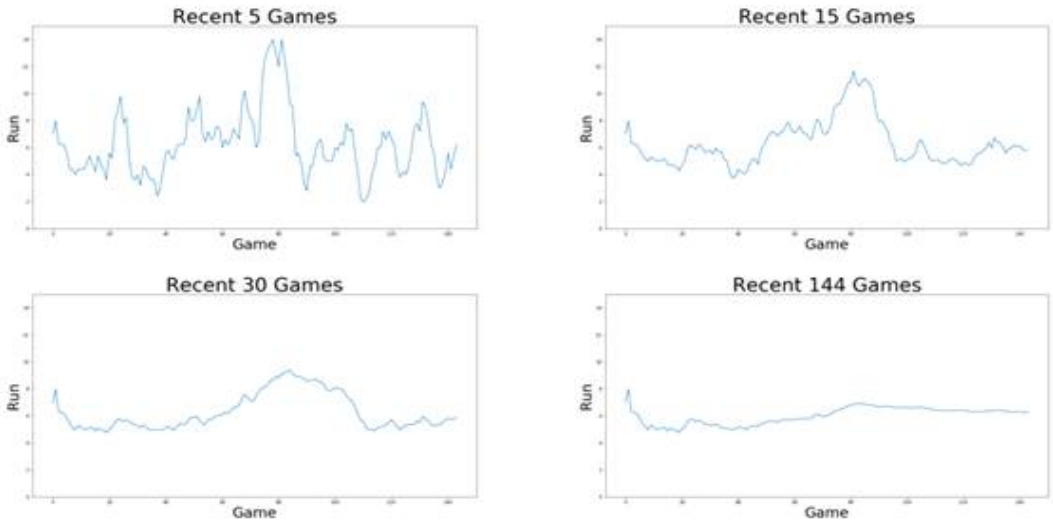
acf, pacf를 확인해 본 결과 과거 데이터에 영향을 받지만 시점에 따른 영향이 무작위성을 띄는 것을 알 수 있다. 득점에 미치는 변수들은 공격력에 더해 상대 투수, 구장, 홈/원정, 상대성 등 다른 변수들에 많은 영향을 받기 때문으로 보인다. 추가로 27개의 아웃카운트로 진행되는 야구 득점은 타 스포츠에 비해 큰 변동성을 가진다.

이제 목표는 간단해졌다. 특정 시점의 공격력과 수비력을 가장 잘 나타내는 변수를 만들고 득점에 영향을 미치는 다른 변수를 찾는 것이다.

## 2) 누적데이터

스포츠에선 최근 경기력에 따른 경기력의 편차가 존재한다. 우리는 최근 경기력과 모집단의 실력사이에서 최적의 선택을 목표한다. 아래는 최근 경기 수(N)에 따른 득점 이동평균 그래프이다(Figure 6).

**FIgure 6. 2017년 KIA팀의 최근 경기 수(N)에 따른 득점 이동평균**




최근 경기 수 N의 변화에 따라 강한 over-fitting(N=5)에서 강한 under-fitting(N=144)까지 다양한 형태를 보인다(Figure 6).

우리의 주된 목표는 최적의 최근 경기 수 N을 구하는데 있다. 그래프 상으로는 최근 15-50경기 정도의 데이터를 사용하는 것이 예측에 용이할 것으로 보인다. 주변 데이터를 통해 특정 지점의 데이터를 예측하는 Kernel-Regression의 경우 전체의 1/3정도 되는 주변 데이터를 사용하는 것이 가장 좋은 결과를 보인다고 알려져 있다. 본 프로젝트에 응용하면 약 50경기 정도를 사용하는 것이 예측에 좋은 결과를 보일 것으로 기대한다.

## 3. 구장 및 홈/원정

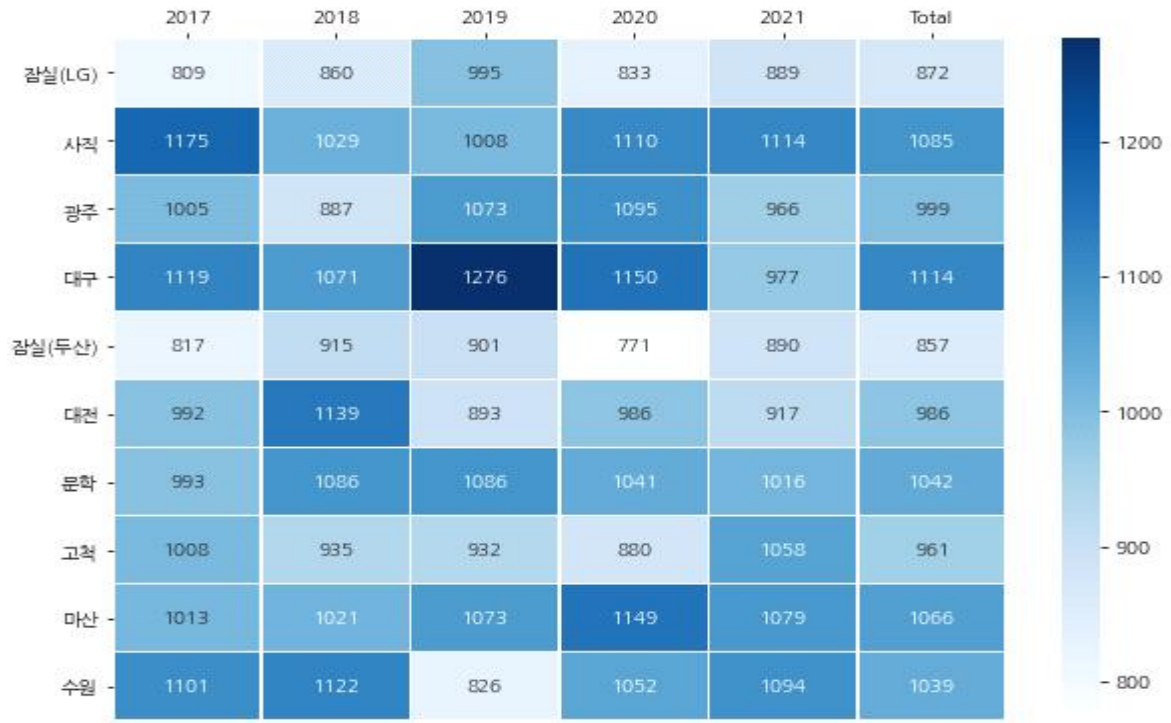
### 1) Park-Factor

구장에 따른 득/실점에 대해 이야기해보자. 득점에 유리한 구장과 불리한 구장의 득점이 가지는 가치는 다르지만 raw-data는 그것을 반영하지 않는다. 야구에서는 이를 보정하기 위해 Park-Factor를 통해 구장 별 유/불리를 판단한다. Park-Factor는 원정과 비교해 홈에서 얼마나 더 많은 득/실점을 했는지를 알 수 있는 지표이다.

$$Parkfactor = \frac{\frac{HomeRun + HomeLoss}{HomeGames}}{\frac{AwayRun + AwayLoss}{AwayGames}}$$

아래는 2017~2021 KBO 구장별 Park-Factor를 나타낸 히트맵이다(Figure 7). 1,000보다 높으면 평균적인 구장과 비교해 득점이 쉬운 것을 의미한다.

**FIgure 7. 2017~2021 KBO 팀/년도 별 Park-Factor**




히트맵을 보면 LG와 두산이 홈으로 사용하는 잠실구장이 가장 투수 친화적인 구장인 것을 알 수 있다. Park-Factor의 경우 년도에 따른 편차가 있기 때문에 일반적으로 누적된 기록을 사용한다. 프로젝트에서는 야구 통계사이트 스탯티즈에서 제공하는 누적된 Park-Factor를 통해 raw-data를 전처리하였다.

## 2) 홈 어드밴티지

스포츠에선 일반적으로 홈 팀이 더 좋은 결과를 가져간다. 2012년 발간된 스코어 캐스팅에 따르면, 미국 프로스포츠 10만 경기에서 야구 54%, 미식축구 57%, 농구 60%, 축구 69%의 홈 승률을 보였다고 한다. 익숙한 홈구장과 압도적인 관중의 응원이 홈팀 선수들에게 심리적, 신체적인 안정감을 주는 반면, 원정팀 선수들에게는 새로운 환경에 대한 심리적인 위축과 피로감으로 작용하기 때문이다. 미국의 한 대학교 연구에서는 홈팀에 대한 유리한 판정을 홈팀이 유리한 가장 큰 이유로 분석하기도 했다.

아래는 2017~2021 KBO 홈/원정에 따른 승률 오즈비를 나타낸 히트맵이다(Figure 8). 승률 오즈비는 원정 대비 홈경기의 승률을 나타낸 것으로, 1보다 높으면 원정보다 홈에서 좋은 승률을 기록하는 것을 의미한다.

$$OddsRatio = \frac{Home\,Win Rate}{Away\,Win Rate}$$

**FIgure 8. 2017~2021 KBO 팀/년도 별 승률 오즈비(홈/원정)**


히트맵을 보면 대부분의 팀/년도에서 원정에 비해 홈 승률이 높은 것을 알 수 있다. 일반적으로 원정보다 홈에서 좋은 결과를 기대할 수 있음을 의미한다.

만약 코로나, 공인구 변경과 같은 특이한 상황이 발생하여 시즌 절반이 지났을 때 원정에 비해 홈에서 많이 불리한 결과가 나온다면 남은 시즌 예측은 어떻게 하는 것이 옳을까?

## 4. 도박사

우리나라에서 열리는 프로스포츠에 대해 도박사(오즈메이커)들은 각 팀의 승률을 예측해 공시한다. 도박사들은 양 팀의 승률을 정확하게 예측하고 수수료를 통해 수익을 챙기는 것을 목표하기 때문에 결과를 비교하기에 좋은 잣대가 된다.

2017-2021 KBO 경기에 대해 도박사들이 예측한 승률 및 결과는 다음과 같다(Table 5).

**Table 5. KBO 도박사 예측결과(2017~2021)**

| 년도 | 예측(%) | [50,55) | [55,60) | [60,65) | [65,70) | [70,75) | 정분류율 | AUC |
|---|---|---|---|---|---|---|---|---|
| 2017 | 승 | 80 | 123 | 105 | 57 | 13 | .577 | .610 |
| | 경기 | 163 | 217 | 159 | 95 | 20 | | |
| | 승률(%) | .491 | .567 | .660 | .600 | .650 | | |
| 2018 | 승 | 143 | 132 | 77 | 14 | 2 | .559 | .557 |
| | 경기 | 254 | 235 | 140 | 27 | 2 | | |
| | 승률(%) | .563 | .562 | .55 | .519 | 1 | | |
| 2019 | 승 | 122 | 112 | 109 | 42 | 10 | .599 | .631 |
| | 경기 | 221 | 200 | 171 | 57 | 11 | | |
| | 승률(%) | .552 | .560 | .637 | .737 | .909 | | |
| 2020 | 승 | 83 | 128 | 97 | 51 | 21 | .607 | .641 |
| | 경기 | 154 | 210 | 158 | 74 | 25 | | |
| | 승률(%) | .539 | .610 | .614 | .689 | .840 | | |
| 2021 | 승 | 95 | 99 | 78 | 44 | 7 | .545 | .587 |
| | 경기 | 206 | 177 | 135 | 63 | 9 | | |
| | 승률(%) | 206 | 177 | 135 | 63 | 9 | | |
| **전체** | 승 | 528 | 600 | 473 | 216 | 62 | **.578** | **.608** |
| | 경기 | 1003 | 1045 | 770 | 324 | 76 | | |
| | **승률(%)** | **.526** | **.574** | **.614** | **.667** | **.816** | | |

2017~2021년 도박사 예측 결과 전체 정분류율 .578, AUC .608으로 낮은 정확도를 가지는 듯 보인다. 그러나 동전 던지기를 예측할 때 50:50으로 예측하는 것이 가장 정확한 것처럼, 통제할 수 없는 영역이 크다면 예측한 확률에 실제 확률이 수렴하는 지를 확인하는 것이 중요하다.

예측 구간 별 정확도를 확인해보자. 공인구 변경으로 큰 혼란을 겪었던 2018년을 제외하면 예측승률과 결과가 준수한 결과를 보인다. 특히, 데이터가 쌓이면서 도박사들의 구간별 예측에 실제 결과 값이 수렴하는 것을 확인할 수 있다.

# Ⅴ. 득점 예측

## 1. 득점예측모형

팀의 공격력과 상대팀의 수비력 그리고 기타변수가 포함된 Regression 모형으로 득점을 설명하는 것을 목표한다. RMSE를 기준으로 다양한 모형을 비교하였다.

$$Y = X\beta + \epsilon,\ \epsilon \sim N(0, \sigma^2)$$

추가적으로 데이터 탐색을 통해 득점(Y)이 감마분포를 따르는 것을 확인하였기 때문에 좋은 퍼포먼스를 보인 모형과 GLM-GAMMA를 비교하였다.

$$g(Y) = X\beta + \epsilon,\ g(Y) \sim \Gamma(\alpha, \beta)$$

득점 모형은 파이썬 라이브러리 scikit-learn과 pycaret을 사용해 분석, 비교하였다.

## 2. Feature-Engineering

일차적인 데이터 전처리 후 Feature-Engineering을 진행하였다. 모든 Feature는 2017~2020 데이터를 5-fold Cross-Validation을 통해 비교 검증하였다.

### 1) Select-N

최근 경기 수 N에 따른 공격력과 수비력 변수를 만들어 보자. 공격력은 17개의 공격력 변수에 대해 선발 라인업(9명) 선수들의 최근 n경기 데이터를 합산하여 사용하였다.

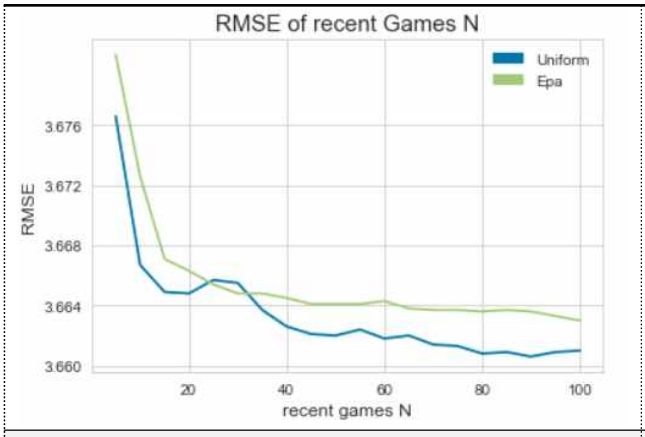
$$\text{공격력: } X_i = \sum_{j=1}^{9}\sum_{k=1}^{n} x_{ijk}\ \ (j: line-up,\ k: historical-data)$$

수비력의 경우 선발투수와 중간계투로 나누었다. 선발투수는 공시된 선발투수의 최근 n경기 데이터, 중간계투는 모든 선수의 과거 n경기 데이터를 합산하여 사용하였다.

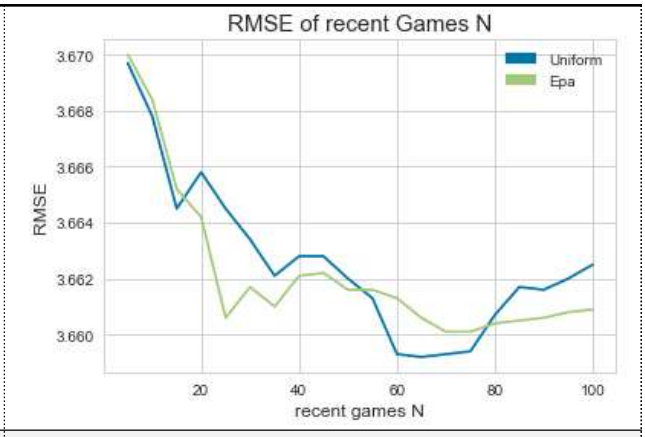
$$\text{수비력: } X_i = \sum_{j=1}^{m}\sum_{k=1}^{n} x_{ijk}\ \ (j: historical-pitcher,\ k: historical-data)$$

먼저 공격력 변수와 중간계투 변수에 대해 최근 경기 수 n과 가중치에 따른 RMSE를 비교해 보았다. 가중치는 Uniform, Epanechnikov-kernel을 비교하였다.

**FIgure 9. 최근 경기 수와 가중치에 따른 RMSE**




최근 경기 수(공격력) 변화에 따른 RMSE



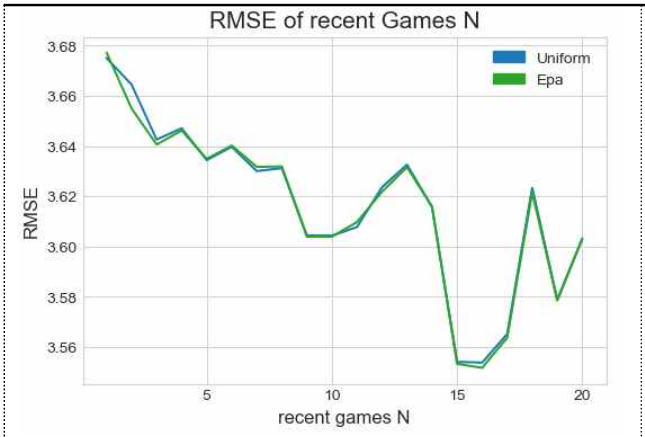

최근 경기 수(중간계투) 변화에 따른 RMSE

RMSE를 기준으로 평가한 결과 공격력과 중간계투 모두 큰 차이는 아니지만 데이터가 어느정도 쌓일수록 더 좋은 결과를 나타내는 것으로 보인다.

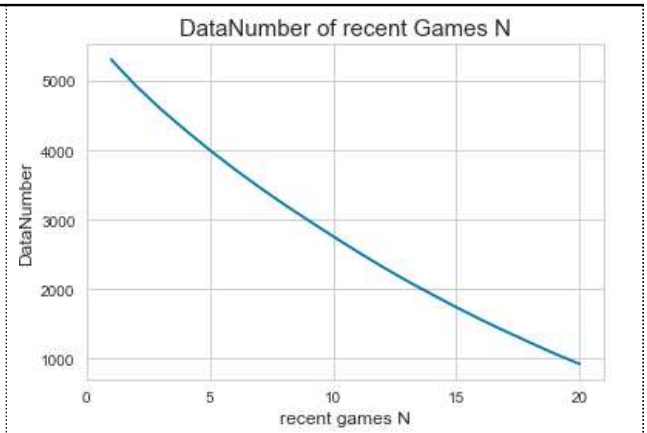
Uniform-Kernel의 경우 일정 경기 수를 넘어가면 정확도가 떨어지는 결과를 보인다. 반면 Epa-Kernel의 경우 경기수가 늘어나면서 RMSE가 조금씩 낮아지는 것을 확인할 수 있다. 많은 범위의 데이터를 사용하되 최근 기록에 대한 가중치를 높여주는 것이 약간은 더 나은 결과를 보인다고 볼 수 있다.

다음은 선발투수에 대해 생각해보자. 선발투수는 정규 9이닝 중 60-70%에 달하는 5-7이닝을 책임진다. 선발투수의 경우 등판 횟수가 적기 때문에 한 시즌 쌓이는 데이터가 최대 20-30경기에 불과하다. 아래는 선발투수 최근 경기 수에 따른 RMSE와 2017-2020년 5,760개 데이터 중 최근 경기 수에 따른 표본 개수에 대한 그래프이다.

**FIgure 10. 선발투수 최근 경기 수에 따른 RMSE와 표본 개수**




최근 경기 수(선발투수) 변화에 따른 RMSE




최근 경기 수(선발투수) 변화에 따른 표본 수

먼저 왼쪽의 최근 경기 수에 따른 RMSE를 살펴보자. 선발투수의 경기 수가 증가하면서 RMSE가 줄어드는 것을 알 수 있다. 데이터가 누적되면서 분산이 줄어들어(선수들의 기록이 본연의 실력에 수렴) 예측이 용이해진다. 하지만 최근 경기 수가 일정 수준 이상을 넘어가면 예측력이 약간 떨어지는데, 이는 최근 경기력을 담지 못하거나 표의 오른쪽 결과를 볼 때 데이터 부족 현상이 일어날 수 있음을 시사한다.

### 2) Transform-Data

공격력, 수비력의 최근 경기 수를 고정하고 도메인지식을 통해 추가적으로 변수를 조정하였다. 조정은 크게 세 가지로 진행하였다.

**① Park-Factor를 통한 변환**

* 경기 후 얻게 되는 종속, 독립변수 데이터가 구장에 따라 다른 분포에서 생성된다.

** Park-Factor 0.9인 구장에서 1점은 1/0.9 ≒ 1.111 점으로 변환

**② 한 경기 9이닝을 가정했을 때 선발투수가 n이닝 던지면 중계투수가 9-n이닝 던지는 것을 감안하여 변수에 가중치를 줌**

* 중계투수진이 아무리 안 좋아도 좋은 선발투수가 나오면 대부분의 경기를 책임짐

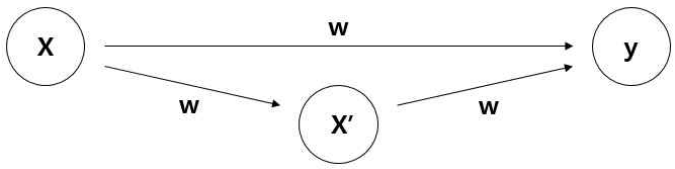
** 수비력(B) = (선발투수 * Inn) + (중간계투 * (9-inn))  ** inn: 선발투수 평균이닝

**③ 야구계에서 선수를 평가할 때 쓰는 지표 추가(XR, FIP)**

* 도메인지식을 통해 변수들의 선형, 비선형함수로 구성된 지표를 추가함으로써 적은데이터로 인공신경망의 히든레이어와 같은 효과를 얻을 수 있음

** XR, FIP: 선수들의 가치 비교를 위해 쓰는 지표로 변수들의 선형, 비선형 함수로 구성

*** 독립변수들 사이의 다중공선성을 높이는 결과로 이어질 수 있으므로 사용에 주의가 필요




위 세 가지 방법을 하나씩 추가해가며 RMSE를 비교하였다.(Table 6).

**Table 6. 변수 조정에 따른 RMSE(5-Cross Validation)**

| 기존 | Park-Factor 추가<br>① | 이닝 가중치 추가<br>① + ② | 지표(XR, FIP) 추가<br>① + ② + ③ |
|---|---|---|---|
| 3.851 | 3.788 | 3.761 | 3.732 |

비교 결과를 살펴보자. 먼저 Park-Factor 변환 결과를 통해 같은 분포에서 생성되지 않은 변수를 변환하는 것이 더 좋은 예측력을 보인다(①). 또한, 도메인지식을 통해 예측변수들을 변환하고 새롭게 생성하는 것이 더 좋은 예측력을 보였다(②, ③). 데이터분석에서 도메인지식이 중요한 부분을 차지하는 것을 나타내는 결과라고 볼 수 있다.

## 3. Model

MAE와 RMSE를 기준으로 모델별 성능을 확인해보았다. 2017-2020년 5760개 데이터를 Train-data, 2021년 1440개 데이터를 Test-data로 사용하였다. Train-data를 5fold-CV를 통해 비교 검증 후 Test-data에 적용하였다. 차원은 Train(5760x56), Test(1440x56)이다.

**Table7. Model에 따른 MAE, RMSE**

| Model | Valid-data | | Test-data | |
|---|---|---|---|---|
| | MAE | RMSE | MAE | RMSE |
| Generalized Linear Regression | 2.9500 | 3.7282 | 2.8325 | 3.5427 |
| Linear Regression | 2.9504 | 3.7262 | 2.8224 | 3.5401 |
| Huber Regression | 2.8900 | 3.7665 | 2.7425 | 3.5484 |
| Ridge Regression | 2.9504 | 3.7261 | 2.8222 | 3.5399 |
| Gradient Boosting | 2.9597 | 3.7415 | 2.8375 | 3.5445 |

파이썬 라이브러리 Pycaret을 통해 모형을 비교한 결과 Regression모형이 ensemble모형에 비해 좋은 결과를 보였다. 이러한 결과는 위 데이터에서 독립변수와 종속변수의 관계가 단순한 선형관계를 보이고 특별한 과적합 문제가 없기 때문으로 보인다(Linear-Regression이 좋은 퍼포먼스를 보이고 Ridge-Regression과 계수 차이가 거의 없다).

Robust-Regression의 일종인 Huber-Regression의 경우 MAE를 기준으로 한 모형평가에서 우위에 있음을 보여줬다. 야구 데이터의 경우 20:0과 같은 이상치 값이 가끔 등장하기 때문에 이를 해소하는데 도움을 줄 것으로 보인다.

마지막으로 GLM-GAMMA는 Linear, Ridge-Regression에 약간 못 미치는 예측 결과를 보였다. GLM-GAMMA의 경우 시즌 별 새로 제공되는 데이터를 사용한 모형을 다룰 때 모형의 형태를 고정시켜 좋은 결과를 보일 것으로 예상된다.

모형 별 예측 결과 차이가 크지 않은 것을 감안하여 위 5개 모형을 모두 승률예측에 사용하였다.

# Ⅵ. 승률 예측

## 1. 승률예측

### 1) 득점분포를 통한 승률 예측

득점분포를 예측하고 시뮬레이션을 통해 승률을 예측해보자. A팀과 B팀의 경기에서 A팀이 이길 확률은 아래와 같이 표현이 가능하다.

$$\text{승률(A)} = P(A > B) \fallingdotseq \lim_{n\to\infty}(\sum_{i=1}^{n} I(A_i > B_i)/n)$$

우리는 탐색적분석을 통해 득점이 감마분포를 따르는 것을 알았다. 득점이 감마분포를 따른다고 가정하고 시뮬레이션을 통해 승률을 계산하면 아래와 같다.

$$\text{추정량 } \widehat{\theta_A} = \Gamma(3,2),\ \widehat{\theta_B} = \Gamma(2,2.5)\text{라고 하면 } \sum_{i=1}^{k} I(A_i > B_i)/k \fallingdotseq 60.08\%$$

### 2) 승률예측 모형

시뮬레이션을 통해 예측한 승률을 분류 모형을 통한 예측 결과와 비교한다. 분류 모형은 크게 확률모형과 판별모형으로 나뉜다.

### 3) 도박사

승률예측 결과를 최종적으로 도박사의 결과와 비교한다. 각 시즌 예측은 해당 시즌 결과를 제외한 시즌 데이터를 사용한 모형으로 예측하였다.

## 2. 예측결과

프로젝트의 최종목표인 승률 예측에 다다랐다. 승률예측은 위에서 구한 득점 추정량에 GAMMA-분포를 가정하고 난수를 생성하여 승률을 예측하였다. GAMMA-분포의 모수 $(\alpha,\beta)$는 조건부 기댓값 $E(Y|X)$와 RMSE를 적률추정량으로 계산하였다.

아래는 난수생성을 통해 구한 2017년~2021년 까지 승률 예측 결과이다. 각 년도 별 예측 결과는 해당 시즌을 제외한 나머지 시즌 데이터를 Train-data로 사용하여 득점 모형을 적합시킨 후 난수생성을 통해 예측하였다. 모형은 GLM-GAMMA, Linear-Regression, Ridge-Regression을 사용하였다(Table-8).

**Table 8. KBO 리그 경기 예측결과(2017~2021)**

| 년도 | 예측(%) | [50,55) | [55,60) | [60,65) | [65,70) | [70,75) | 정분류율 | AUC |
|---|---|---|---|---|---|---|---|---|
| 2017 | 승 | 137 | 119 | 88 | 20 | 11 | .577 | .601 |
| | 경기 | 253 | 207 | 138 | 35 | 13 | | |
| | 승률(%) | .542 | .575 | .638 | .571 | .846 | | |
| 2018 | 승 | 168 | 116 | 46 | 28 | 3 | .553 | .567 |
| | 경기 | 320 | 198 | 90 | 40 | 5 | | |
| | 승률(%) | .525 | .586 | .511 | .700 | .600 | | |
| 2019 | 승 | 113 | 115 | 83 | 36 | 26 | .603 | .655 |
| | 경기 | 220 | 199 | 122 | 51 | 36 | | |
| | 승률(%) | .514 | .578 | .680 | .706 | .722 | | |
| 2020 | 승 | 106 | 128 | 84 | 37 | 22 | .595 | .641 |
| | 경기 | 213 | 210 | 132 | 53 | 28 | | |
| | 승률(%) | .498 | .610 | .636 | .698 | .786 | | |
| 2021 | 승 | 115 | 137 | 64 | 31 | 6 | .594 | .627 |
| | 경기 | 221 | 224 | 102 | 45 | 9 | | |
| | 승률(%) | .520 | .612 | .627 | .689 | .667 | | |
| **전체** | 승 | 644 | 621 | 372 | 160 | 77 | **.585** | **.621** |
| | 경기 | 1,232 | 1,044 | 591 | 232 | 100 | | |
| | **승률(%)** | **.523** | **.595** | **.629** | **.699** | **.770** | | |

예측 결과를 살펴보자. 정분류율 .585, AUC .621로 도박사와 비슷한 결과를 보였다. 다음은 구간 별 예측 결과를 보자. 시즌 별 구간예측은 차이가 있으나 데이터가 쌓이면서 구간별 예측에 실제 결과가 수렴하는 것을 알 수 있다.

다음은 시즌별 예측모형과 도박사의 예측정확도(정분류율, AUC)를 비교해보았다(Table-7).

**Table 9. 시즌 별 예측정확도 비교(득점예측모형, 도박사)**

| 구 분 | 정확도 | 2017 | 2018 | 2019 | 2020 | 2021 | 전체 |
|---|---|---|---|---|---|---|---|
| 예측모형 | 정분류율 | .577 | .553 | .603 | .595 | .594 | .585 |
| | AUC | .601 | .567 | .655 | .641 | .627 | .621 |
| 도박사 | 정분류율 | .577 | .559 | .599 | .607 | .545 | .578 |
| | AUC | .610 | .557 | .631 | .641 | .587 | .608 |

비교 결과 예측모형과 도박사의 시즌 별 예측정확도가 유사한 흐름을 보이는 것을 알 수 있다. 이를 통해 유추할 수 있는 건 야구에서 예측을 하는데 쓰이는 모형은 매년 크게 차이가 없는 것으로 보인다. 다만, 시즌별 특성에 따른 차이가 존재하는지 운에 의한 차이인지를 확인해 볼 필요가 있어 보인다.

## 3. Classification and Gambler

최종적으로 득점예측모형을 통한 승률예측, RandomForest, 도박사의 결과를 비교하였다.

**Table 10. 득점예측모형, Logistic, 도박사 결과 최종비교**

| 구 분 | 예측(%) | [50,55) | [55,60) | [60,65) | [65,70) | [70,75) | 정분류율 | AUC |
|---|---|---|---|---|---|---|---|---|
| 득점예측모형 | 승 | 644 | 621 | 372 | 160 | 77 | **.585** | **.621** |
| | 경기 | 1,232 | 1,044 | 591 | 232 | 100 | | |
| | **승률(%)** | **.523** | **.595** | **.629** | **.699** | **.770** | | |
| Random Forest | 승 | 698 | 638 | 357 | 177 | 68 | **.553** | **.577** |
| | 경기 | 1,367 | 1,168 | 603 | 283 | 93 | | |
| | **승률(%)** | **.511** | **.546** | **.590** | **.625** | **.731** | | |
| 도박사 | 승 | 528 | 600 | 473 | 216 | 62 | **.578** | **.608** |
| | 경기 | 1003 | 1045 | 770 | 324 | 76 | | |
| | **승률(%)** | **.526** | **.574** | **.614** | **.667** | **.816** | | |

비교 결과 득점예측모형, 도박사, RandomForest 순으로 정확한 결과를 보였다. 분류모형의 경우 3:2, 6:2, 17:1과 같은 결과를 모두 같은 1과 0으로 나타내기 때문에 정보의 손실이 있는 것으로 보인다.

# Ⅶ. 결론

## 1. 결론

### 1) 결과

KBO리그 경기의 득점 및 승률 예측을 주제로 데이터 수집부터 분석, 결과 비교, 송출까지 데이터분석의 전 과정을 진행해보았다. 먼저 크롤링을 통해 2017~2021 KBO 데이터를 수집 및 저장하였고, 탐색적분석을 통해 변수들의 기본적인 특징을 파악하였다. 다음으로 일차적인 전처리 후 탐색적분석을 통해 얻은 정보와 도메인지식을 활용하여 변수와 파라미터를 조절하면서 퍼포먼스를 끌어올렸다. 적절한 Feature-Engineering과 모델 간 비교, 선택을 통해 최종적으로 최적화 된 모형을 도박사들의 예측결과와 비교하였고 도박사들의 예측과 유사한 결과를 보였다.

### 2) 한계점

프로젝트를 진행하며 느낀 한계점을 크게 두 가지로 뽑아보았다.

먼저, 데이터 자체가 가지는 한계점이다. 타격 및 투구 결과 테이블을 크롤링 하다 보니 트레킹 데이터와 같은 세부적인 데이터를 얻을 수 없었다. Garbage in Garbage out. 주어진 데이터를 활용하여 최대한의 퍼포먼스를 보였지만 데이터의 한계가 느껴졌다.

* 트레킹 데이터란? 투수들의 투구(속도, 회전 수, 무브먼트 등)와 타자들의 타격((타구속도, 발사각도 등)을 세부적으로 나타내는 데이터로 선수들의 대한 세부 평가와 미래 예측에 사용한다.

다음은 데이터 부족에 따른 주관의 적용이다. 예를 들어, 내일 경기에 출장하는 선발투수 데이터가 2~3경기 밖에 쌓이지 않았다고 가정해보자. 이 때 2-3경기 기록에 대한 평가는 실력과 운 사이에서 줄다리기를 타게 된다. 이와 같은 이유로 선발투수의 최근경기 수가 늘어날수록 선발투수 변수의 가중치가 늘어난다. 만약 류현진이 첫 2경기에서 부진한 경기력을 보였다고 한다면 그 결과를 그대로 적용할 것인가? 데이터가 부족한 시작 지점의 예측은 과거 지식을 활용한 주관적 요소의 개입이 필요하다.

## 2. 계획

### 1) 통계적 모형

야구는 매년 새로운 시즌을 맞이하고 오프시즌 새로운 변화를 일으킨다. 2018년 공인구 변화가 그랬고 2022년 스트라이크존 변화가 그랬다. 이러한 큰 변화는 야구라는 모집단의 변화를 일으킨다. 이에 따라, 새로 수집되는 데이터의 가중치에 대한 고민이 필요하다. 새롭게 얻는 데이터가 유의미한 변화를 일으킨 모집단에서 생성된다고 가정하면 그 데이터에 대한 가중치를 높이는 것이 옳을 것이다.

예를 들어, 작년까지 엄청난 퍼포먼스를 보인 투수가 스트라이크 존의 변화 등으로 첫 2~3경기에서 크게 부진했다고 하면 어떻게 바라봐야할까. 만약 기존에 스트라이크 존의 변화, 공인구의 변화 등 투수에게 크게 영향을 미치는 변화가 없었다면?

프로젝트의 다음 방향성은 새로 얻게 되는 시즌 데이터에 대한 관점을 집중적으로 분석하는데 있다.

### 2) 완전자동화

프로젝트는 데이터의 수집과 결과의 송출을 포함한다. 따라서 데이터 분석과 별개로 데이터 수집과 결과 송출에 대한 완전자동화를 목표한다.

현재 스케줄러를 통해 도박사 예측 및 결과에 대한 자동화를 진행하고 있다. 하지만 데이터 수집부분은 크롤링에 오류가 있어 데이터가 오염되거나 손실이 있을 경우 후작업에 많은 시간이 드는 것을 확인하여 이를 해결할 방법을 찾는 중이다. 프로젝트의 최종 목표는 주식의 퀀트매매처럼 모든 과정의 완전한 자동화를 목표한다.

### 3) 구조의 변화

현재 주어진 데이터를 통해 엄청난 퍼포먼스의 변화를 일으키는 것은 어려워 보인다. KBO 리그의 세부적인 데이터를 얻는 방법(MLB에서는 트레킹 데이터와 다양한 세부 데이터들을 API로 제공 중)이 생기거나 찾는다면 더 나은 퍼포먼스를 위해 새로운 DB와 예측모형을 구축하는 것을 목표한다. 위와 같은 구조의 변화는 긴 시간을 두고 진행될 것으로 보인다. 도박사를 이기는 그 날 까지.

## 3. 홈페이지

경기의 득점과 승률 예측을 매일 홈페이지에 업데이트한다. 홈페이지는 Django와 JS, CSS등을 사용해 개발하여 AWS를 통해 운영하고 있다(URL: http://15.164.213.230).