타격 지표 기반
타자의 타격 유형 자동 분류를 통한
이사만루 육성 가이드 시스템

과목 : 기계학습의이해
담당 교수 : 최승택 교수님
학과 : Language & AI융합학부
학번 : 202401312
이름 : 오승준
제출일 : 2025.12.12

Ⅰ. 프로젝트 개요

(1) 무엇을 만들었나

본 프로젝트는 '타격 지표 기반 타자의 타격 유형 자동 분류를 통한 이사만루 육성 가이드 시스템'입니다. 이 시스템은 사용자가 KBO 공식 기록실에서 확보할 수 있는 타자의 타격 지표를 입력하면, 학습된 머신러닝 모델이 데이터를 분석하여 해당 타자를 Power Hitter, Gap Hitter, Contact Hitter 중 하나의 유형으로 자동 분류합니다. 더불어 단순 분류에 그치지 않고, 분류된 유형에 따라 모바일 게임 '이사만루' 내에서 타자 육성에 사용할 수 있는 20포인트를 컨택, 파워, 선구에 어떻게 분배해야 하는지 최적의 육성 가이드라인을 제시하는 웹 서비스입니다.

(2) 어떤 문제를 해결했나

'이사만루' 게임에서 타자를 육성할 때는, 20포인트를 타자의 성향에 맞게 효율적으로 분배해야 합니다. 하지만 어떤 유형인지 잘 알지 못하는 타자를 획득했을 경우, 정확한 유형 파악이 어려워 육성에 실패하는 문제가 발생했습니다.

이를 해결하기 위해 머신러닝 모델을 통해 객관적인 타격 데이터에 기반하여 타자 유형을 판별하고, 각 유형별로 사전에 설계된 최적의 포인트 분배 로직을 자동으로 매칭해 줌으로써 사용자의 고민 시간과 육성 실패 가능성을 획기적으로 절감하였습니다.

Ⅱ. 진행 과정

(1) 주제 선정 및 문제 정의

본 프로젝트의 주제는 실제 '이사만루' 게임 플레이 과정에서 겪은 타자 육성 실패라는 구체적인 경험에서 착안했습니다. 게임 내에서 생소한 선수를 획득했을 때, 육성 방향을 잘못 설정하여 재화를 낭비하는 문제가 빈번하게 발생했습니다. 이에 따라 누구나 쉽게 구할 수 있는 타격 지표를 입력하면 타자의 유형을 자동으로 판별해주는 서비스가 있다면 이 문제를 해결할 수 있을 것이라고 생각하여 해당 주제를 선정하게 되었습니다.

또한 단순히 타자의 유형을 분류하는 것만으로는 '이사만루' 게임 내에서의 실질적인 문제 해결이 어렵다고 판단하여, 도메인 지식을 기반으로 각 타자 유형에 맞는 최적의 스탯 분배 로직을 사전에 정의하는 과정을 거쳤습니다. 이를 통해 모델의 분석 결과가 즉각적으로 게임 내의 문제를 해결할 수 있도록 시스템을 설계했습니다.

(2) 데이터 수집 및 분석

데이터 수집 대상은 2025년 KBO 골든글러브 후보 중 타자 50명으로 선정하였으며, 최근 5년간의 기록을 KBO 공식 기록실을 통해 확보하였습니다. 이때, 타격 성향을 파악하기에 표본이 부족한 시즌을 배제하기 위해 50타석 이상 소화한 시즌만을 선별하였고, 최종적으로 총 195개의 데이터 샘플을 수집하였습니다.

데이터셋은 식별을 위한 메타 데이터(ID, Batter_Type), KBO 기록실을 통해 확보한 원천 데이터(PA(타석), AB(타수), H(안타), 2B(2루타), 3B(3루타), HR(홈런), BB(볼넷), SO(삼진)), 그리고 원천 데이터를 비율로 변환한 파생 데이터(H_Ratio(H/AB), 2B_3B_Ratio((2B+3B)/AB), HR_Ratio(HR/AB), SO_Ratio(SO/PA), BB_Ratio(BB/PA))로 총 15개 컬럼으로 구성됩니다. 수동 라벨링 단계와 추후에 진행하였던 모델 학습 과정에서는 타석이나 타수 차이에 따른 단순 누적 기록의 왜곡을 방지하기 위해, 원천 데이터를 비율로 변환한 5가지 파생 변수를 핵심 입력 변수로 채택하여 활용하였습니다.

수집된 데이터의 분포를 분석한 결과, Gap Hitter 유형이 전체의 58.5%로 과반을 차지하였으며, Contact Hitter(22.1%) 유형과 Power Hitter(19,5%) 유형이 나머지를 구성하는 클래스 불균형 현상이 확인되었습니다. 하지만 도메인에 대한 지식을 바탕으로 이는 데이터 수집의 오류가 아닌 KBO 리그의 실제 선수 분포 특성이 정확히 반영된 결과로 해석하였습니다.

또한 데이터 시각화 분석을 수행한 결과, HR_Ratio, SO_Ratio, H_Ratio는 타격 유형별 분포가 뚜렷하게 구분되어 모델의 핵심적인 분류 기준이 될 것으로 파악되었습니다. 반면, 2B_3B_Ratio와 BB_Ratio는 유형 간 분포의 중첩이 많아 상대적으로 변별력이 낮을 것으로 분석되었습니다.

(3) ML 모델 학습 및 평가

MLP (Multi-Layer Perceptron) 구조를 최종적으로 채택하여 모델 아키텍처를 설계하였습니다. 각 은닉층에는 ReLU 활성화 함수를 사용하여 기울기 소실 문제를 방지하였으며, 5차원의 입력 데이터를 64차원으로 확장하여 숨겨진 패턴을 학습한 뒤 다시 32차원으로 압축하여 핵심 특징을 추출하도록 설계했습니다. 특히 195개의 적은 데이터 샘플로 인한 과적합 문제를 방지하기 위해 Dropout(0.3)을 적용하였고, 앞서 파악된 클래스 불균형 문제를 보완하기 위해 손실 함수로 Weighted CrossEntropyLoss를 사용했습니다.

총 195개의 데이터 샘플 중 20%에 해당하는 39개를 Test Set으로 분리하였고, 80%에 해당하는 156개만을 활용하여 Train과 Validation을 수행해야 했기 때문에 Cross Validation 기법을 도입하였습니다. 최적의 모델을 선정하는 기준으로는 Validation Macro F1 Score를 사용하였고, 4-Fold Cross Validation을 수행한 결과, Fold 3의 Epoch 23 시점에서 Validation Macro F1 0.9451을 기록한 모델이 가장 우수한 성능을 보여 최종 모델로 선정하였습니다.

최종 선정된 모델의 일반화 성능을 검증하기 위해, 학습 과정에서 제외해두었던 Test Set 39개를 활용하여 평가를 수행했습니다. 평가 결과, Macro F1 0.7137, Weighted F1 0.7206, Accuracy 0.7179를 기록하며 전반적인 평가 지표에서 0.7을 상회하는 준수한 분류 성능을 확보하였습니다.

특히 가장 주목할 만한 성과는 Top-2 Accuracy 100% 달성입니다. 이는 모델이

예측한 1순위와 2순위 후보 내에 실제 정답이 반드시 포함된다는 것을 의미합니다. 또한 오분류 패턴을 분석한 결과, 성향이 극명하게 대조되는 Power Hitter와 Contact Hitter 간의 혼동은 단 한 건도 발생하지 않았으며, 모든 오차는 중간 성향인 Gap Hitter와의 경계에서만 제한적으로 발생함을 확인하였습니다. 이를 통해 모델의 학습이 제대로 이루어졌으며, 모델이 예측하는 방향도 올바르게 진행되고 있음을 확신할 수 있었습니다. 최종적으로 모델이 타격 지표와 타자 유형 간의 관계를 올바르게 학습하고 있으며, 결과적으로 충분히 신뢰할 수 있는 예측 성능을 확보했다고 판단하였습니다.

(4) 서비스 개발

학습 및 평가가 완료된 모델을 Python 기반의 Gradio 라이브러리를 사용하여 직관적인 웹 인터페이스를 구축했습니다. 복잡한 파생 변수를 직접 계산할 필요 없이 KBO 기록실에서 확보할 수 있는 원천 데이터만 입력하면, 시스템이 내부적으로 파생 변수를 계산하여 처리하도록 설계했습니다. 최종 결과물은 Hugging Face Spaces를 통해 배포하여, 별도의 설치 없이 웹 브라우저를 통해 언제 어디서든 접속 가능한 서비스 형태로 개발을 완료했습니다.

서비스 링크: https://huggingface.co/spaces/ohsj611/space_batter_classiier

Ⅲ. 모델을 서비스로 만든 구조

(1) 학습한 모델을 어떻게 실제 사용 가능한 서비스로 만들었나

학습이 완료된 모델 가중치 파일과 스케일러를 로드하여 추론 엔진을 구축하고, 이를 웹 인터페이스와 연동하였습니다. 또한 모델을 학습시킬 때에는 단순 누적 기록 데이터를 비율로 변환하여 사용하였으나, 실제 서비스를 사용할 때, 매번 비율 계산을 직접 수행하여 입력하는 것은 번거로울 것으로 판단하였습니다. 이에 따라 단순 누적 기록 데이터를 입력하면, 이를 곧바로 모델 학습 시와 동일한 공식을 적용하여 비율로 변환하는 과정을 자동화하여 서비스를 만들었습니다.

(2) 시스템 구조

시스템 구조는 입력 - 처리 - 출력의 3단계 파이프라인으로 구성됩니다.

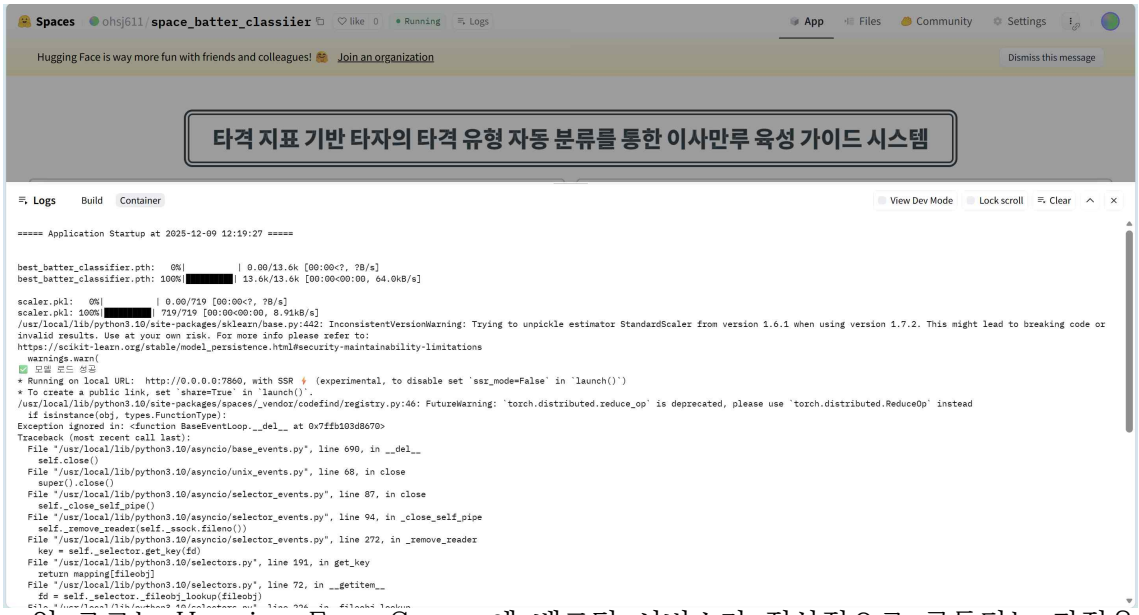
1. 입력: 웹 인터페이스를 통해 타자의 이름, 입력할 타격 지표에 해당하는 연도, 8가지 단순 누적 타격 지표(타석, 타수, 안타, 2루타, 3루타, 홈런, 삼진, 볼넷)를 입력받습니다.

2. 처리 1: 입력받은 단순 누적 타격 지표를 모델 학습 시와 동일한 공식을 적용하여 5가지 변수(H_Ratio, 2B_3B_Ratio, HR_Ratio, SO_Ratio, BB_Ratio)로 자동 변환합니다.

3. 처리 2: MLP 모델이 변환된 데이터를 입력받아 3가지 클래스에 대한 확률 점수를 계산하고, 최종 유형을 결정합니다.

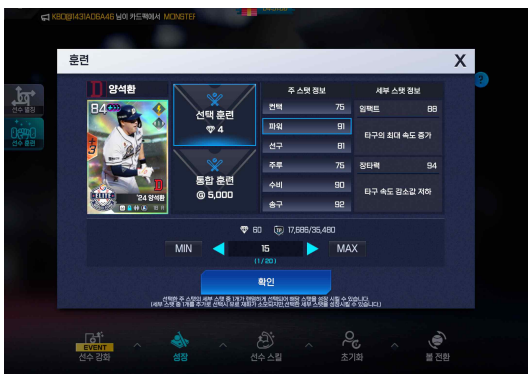
4. 출력: 결정된 타자 유형과 이에 따른 이사만루 육성 가이드를 텍스트 형태로 화면에 출력합니다.

Ⅳ. 실제 사용 결과

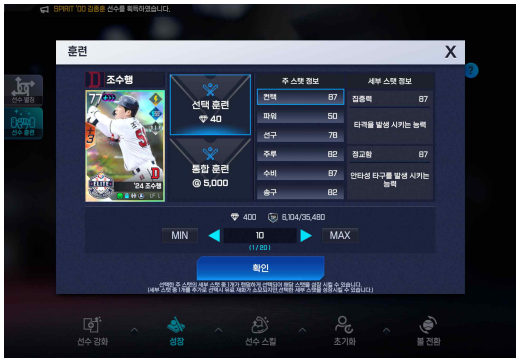


위 로그는 Hugging Face Spaces에 배포된 서비스가 정상적으로 구동되는 과정을 보여줍니다.

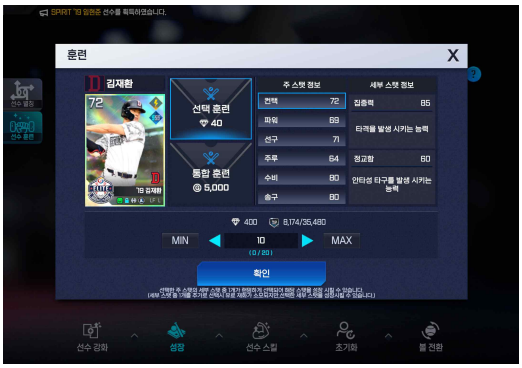
1. 양석환(2024) 선수를 분석하여, Power Hitter 결과를 받았고, 이에 따라 육성을 진행하는 모습입니다.



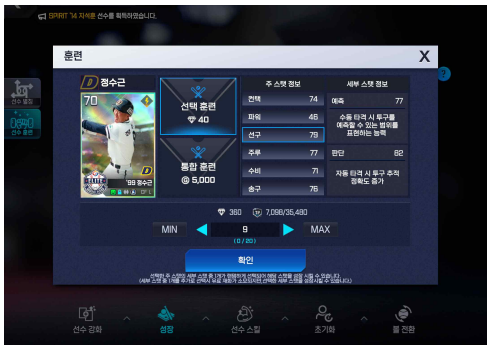
2. 조수행(2024) 선수를 분석하여, Contact Hitter 결과를 받았고, 이에 따라 육성을 진행하는 모습입니다.



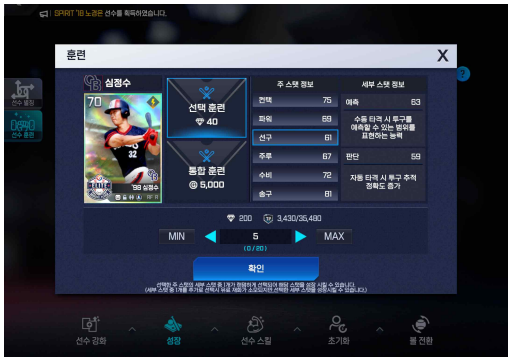
3. 김재환(2019) 선수를 분석하여, Gap Hitter 결과를 받았고, 이에 따라 육성을 진행하는 모습입니다.



4. 정수근(1999) 선수를 분석하여, Contact Hitter 결과를 받았고, 이에 따라 육성을 진행하는 모습입니다.



5. 심정수(1996) 선수를 분석하여, Gap Hitter 결과를 받았고, 이에 따라 육성을 진행하는 모습입니다.



위의 양석환(2024), 조수행(2024), 김재환(2019), 정수근(1999), 심정수(1996) 선수를 포함하여 총 9명의 타자들을 직접 만든 서비스를 이용해 분석한 후, 훈련을 진행했습니다. 그리고 해당 9명의 타자들로 선발 라인업을 구성하여 프로 난이도에서 자동 진행 모드로 한 시즌을 플레이하였습니다.

결과적으로 팀은 정규시즌 순위 1위를 기록하였으며, 타자 1위 수상자는 무려 8명, 골든글러브 수상자도 3명을 배출하면서 성공적인 결과를 확인할 수 있었습니다. 비록 또한 세부적인 타격 지표를 자세히 살펴보면 대부분의 타자들이 리그 평균을 상회하는 준수한 성적을 기록하면서 전반저으로 성공적인 육성을 해냈다고 확신하게 되었습니다. 이러한 성공적인 경험을 바탕으로 앞으로도 해당 게임을 플레이할 때 애용하게 될 것이라 생각합니다.

V. 배운 점 및 개선 방향

수업 시간에 다루었던 다양한 머신러닝 이론들이 실제 프로젝트에서 핵심적이고 필수적인 도구로 활용되는 과정을 몸소 경험할 수 있었습니다. MLP 모델의 아키텍처를 설계하고, Dropout을 통해 과적합을 제어하며, Cross Validation으로 모델의 신뢰성을 검증하는 등의 여러 과정들을 직접 수행하면서 이러한 이론들에 대해 깊이 있게 이해하는 계기가 되었습니다.

이번 프로젝트를 진행하면서 가장 많은 시간을 소모했고, 가장 어렵다고 느꼈던 단계는 데이터 수집 및 분석 단계였습니다. 단순히 모델을 설계하고 학습하고 평가하는 활동보다, 어떤 데이터를 수집하고 어떻게 분석하는지가 정말 중요한 활동이고, 동시에 어려운 활동이라는 사실을 알게 되었습니다. 해당 단계에서 도메인에 대한 풍부한 지식을 활용해 클래스 불균형을 문제를 처리했던 경험은 스스로 자신감을 얻을 수 있었던 경험이었습니다.

어떤 프로젝트를 진행할 것인지 선정하는 단계부터 최종적으로 완성된 서비스를 배포하는 전체의 과정을 온전히 스스로의 힘으로 수행하는 경험은 처음이었습니다. 결과적으로 이러한 프로젝트를 성공적으로 완수하면서 이론을 넘어서는 귀중한 경험을 얻게 되었다고 생각합니다.

다만, 본 프로젝트에서 사용한 195개의 데이터 샘플은 모델으 성능을 극대화하기에 다수 부족하다고 생각합니다. 따라서 향후 데이터셋의 크기를 늘려 모델을 다시 학습

시킨다면 좀 더 좋은 성능을 보이는 모델을 만들 수 있을 것이라 생각합니다. 뿐만 아니라 타구 속도나 발사 각도와 같은 트래킹 데이터나 헛스윙 비율 같은 좀 더 전문적인 데이터를 추가 변수로 확보한다면, 오분류가 발생하는 Gap과 Contact, 그리고 Gap과 Power 유형 간의 분류 성능을 향상시킬 수 있을 것이라 생각합니다.