Predicting The Shot In Basketball

with Kobe Bryant 20-year Shot Data

Jaeyoon Han

*2016*년 *6*월 *14*일

1 Introduction

최근 스포츠 분야는 경쟁력 강화를 위해 적극적으로 데이터 분석과 기계학습을 활용하고 있다. 마이클 루이스는 자신의 저서인 머니볼을 통하여 통계학과 데이터 분석이 구단 운영과 전술 수립 및 선수단 운용에 미치는 긍정적인 영향에 관하여 보여준 바 있다.[1] 특히 야구의 경우, '데이터 야구'라는 말이 보편화될 만큼 많은 자료들이 제공되고 있다. 미국 메이저리그는 세이버메트릭스(sabermetrics)를 통해 선수들의 기술적 능력에 대한 객관적인 평가 척도를 제시하고 있다. 이러한 현상과 더불어 일반인들도 데이터를 기반에 둔 스포츠 토토 등을 통하여 스포츠 경기 결과를 예측하는 문화가 자리 잡고 있다.

농구의 경우, 미국 메이저리그에 새로운 혁신을 불러온 세이버메트릭스의 영향을 받은 APBR(Association for Professional Basketball Research)이 중심이 되어 APBR메트릭스를 개발하였다. NBA의 경우, 지금까지도 새로운 선수 평가 방법을 개발하여 일반인들에게 개방하고 있다. 한국 프로농구는 2015년부터 KBL 레퍼런스를 통해 APBR메트릭스를 제공하고 있다.[2]

데이터 기반의 스포츠 문화가 정착할 수 있었던 이유는 방대한 데이터가 축적되고 활용할 수 있는 인프라가 마련되었기 때문이다. 이러한 데이터를 통해 스포츠분야에서 데이터마이닝을 통해 승패를 예측하는 다양한 연구가 수행되어 왔다. Oh *et al.*은 의사결정나무(Decision tree), 랜덤포레스트(Random forest), 로지스틱 회귀분석(Logistic regression)을 활용하여 한국프로야구 경기의 승패를 예측하였다.[3] Lucey *et al.*은 2010-11 시즌 잉글리쉬 프리미어리그의 볼 트래킹(ball tracking) 시공간 데이터와 이벤트 데이터 등을 이용하여 경기의 승패를 예측하였다.[4] Miljkovic *et al.*은 NBA 2009~2010 시즌의 778경기 결과를 바탕으로 농구경기의 승패를 예측하기 위하여 나이브 베이즈 분류기와 선형회귀분석을 도입하는 시도를 하였다.[5] Stoudt *et al.*은 NCAA 농구 토너먼트에 참가하는 팀들의 이전 시즌 결과를 바탕으로 앙상블 기법(Ensemble method)을 활용한 경기 승패 예측 및 우승 예측을 시도했다.[6] 이상과 같이 기계학습을 활용한 스포츠 분야의 연구들은 대부분 선수 개개인보다 승패에 초점이 맞춰져있다. 본 프로젝트에서는 농구에서 주어진 상황 데이터와 그라디언트 부스팅(Gradient Boosting)을 사용하여 선수 개인의 슛 퍼포먼스를 예측하기 위한 시도를 하였다. 뿐만 아니라 선수의 슛 퍼포먼스에 영향을 미치는 중요 변수들을 탐색하였다.

2 Methodology

2.1 Data Description

본 프로젝트에서는 NBA 홈페이지에서 API를 통해 제공하는 선수의 슛 차트를 데이터로 활용하였다. 그 중에서도 NBA 역사상 최대 슛 시도 기록을 보유하고 있으며, 최근 2015-2016 시즌을 마지막으로 은퇴한 로스 엔젤레스 레이커스의 코비 브라이언트(Kobe Bryant)의 슛 데이터와 경기 정보 데이터를 조인(join)해서 사용하였다. 데이터 수집 및 전처리는 R(http://cran.r-project.org/) 을 이용하여 수행하였다. 수집한 데이터는 총 30,697개의 인스턴스와 21개의 변수로 구성되어 있다. 수집한 데이터의 각 변수들의 정보는 <표 1>과 같다.

Table 1: Description of variables for original data

변인	데이터 타입	설명	비고
action_type	factor	해당 슛의 모션 정보	
combined_shot_type	factor	해당 슛의 종류	
game_event_id	numeric	전체 게임에서의 슛의 순번	
game_id	numeric	경기 식별자	
loc_x	numeric	경기장 x 좌표	림 바로 밑이 0
loc_y	numeric	경기장 y 좌표	림 바로 밑이 0
minutes_remaining	numeric	남은 시간 (분)	
period	numeric	해당 쿼터	1~4 : 정규경기, 5~7 연장전
playoffs	logical	플레이오프 여부	
season	factor	해당 시즌	
seconds_remaining	numeric	남은 시간 (초)	
shot_distance	numeric	슛을 시도한 위치에서 림까지의 거리	단위 : 피트(feet)
shot_made_flag	logical	슛의 성공 여부	
shot_type	factor	2점슛, 3점슛 여부	
shot_zone_area	factor	슛을 시도한 곳의 위치	
shot_zone_basic	factor	슛을 시도한 곳의 명칭	
shot_zone_range	factor	슛을 시도한 위치에서 림까지의 거리	
game_date	factor	경기 일자	
matchup	factor	경기 매치업	
opponent	factor	상대팀 이름	
shot_id	numeric	슛 식별자	

2.2 Data Preprocessing

2.2.1 Feature Engineering

본 프로젝트의 목적은 주어진 상황에서의 슛 퍼포먼스를 예측하는 것이기 때문에, 수집한 데이터에서 필요한 변수를 추가로 생성하고 기존 변수에서 중요한 변수들을 요약했다. 우선 이전 경기와 당일 경기 사이의 기간이 짧을 수록 선수의 체력에 부정적인 영향을 미치기 때문에, `game_date` 변수를 사용하여 경기간 휴식 기간을 새로운 변수로 활용했다. 이 때 경기간 휴식 기간이 1일부터 수백일까지 다양하여, 선수들이 모두 쉬게 되는 올스타 브레이크 기간인 7일을 최댓값으로 활용하였다.

홈에서 경기를 치루는 경우, 선수들이 다른 지역으로 이동할 필요가 없기 때문에 체력적인 부담이 덜하고 관중의 대부분이 자신의 팀을 응원하기 때문에 체력적, 정신적인 부담이 최소화된다. 따라서 `matchup` 변수에서 홈과 어웨이를 구분하여 논리형 데이터로 생성하였다.

NBA의 경기는 기본적으로 12분 4쿼터에 걸쳐 진행되며, 승부가 결정나지 않은 경우 5분의 연장전을 실시한다. 연장전은 승부가 결정될 때까지 반복된다. 연장전은 선수들의 체력에 부담이 가기 때문에 슛 퍼포먼스에 영향을 미칠 것으로 예상된다. 따라서 `period` 변수를 활용해 연장전 여부를 새로운 변수로 생성하였다.

마지막으로, 경기장의 좌표는 림을 원점으로 하는 벡터와 동일하며, 좌표를 통하여 거리와 각도 정보를 얻을 수 있다. 두

벡터의 내적은

$$x \cdot y = \|x\|\|y\| \cos\theta$$

이므로, 두 벡터 사이의 각도는 다음과 같다.

$$\cos\theta = \frac{x \cdot y}{\|x\|\|y\|}$$
$$\theta = \cos^{-1}\left(\frac{x \cdot y}{\|x\|\|y\|}\right).$$

위 식을 이용하여 `loc_x`, `loc_y` 변수를 각도로 변환하였다. 새로 생성한 변수에 대한 정보는 <표 2>와 같다.

Table 2: Description of new variables derived by feature engineering

변인	데이터 타입	설명	비고
`day_difference`	numeric	이전 경기로부터의 휴식 기간	
`home`	logical	홈, 어웨이 여부	
`overtime`	logical	연장전 여부	
`angle`	numeric	슛을 시도한 위치의 각도	

2.2.2 Data Exploration & Visualization

피처 엔지니어링(Feature engineering)이 끝난 데이터의 변수들에 대한 분포 및 특성을 확인하기 위해 탐색적 데이터 분석(Exploratory Data Analysis, EDA)를 시도했다. <그림 1>은 모든 슛을 코트 위에 시각화한 결과물이고, <그림 2>는 모든 슛을 시즌 별로 나눠 시각화한 결과물이다. <그림 1>를 통하여 코트 전체적으로 고르게 슛을 시도하였고, 골밑에서의 슛 시도율 및 성공률이 높은 것으로 나타났다. 또한 <그림 2>를 통해 데뷔 후 몇 년간 저조한 슛 성공률을 보였지만, 그 이후는 준수한 슛 성공률을 보였으며, 2010-11 시즌과 2011-12 시즌에는 골밑 득점을 중심으로 한 슛 성공률이 높았음을 알 수 있다.

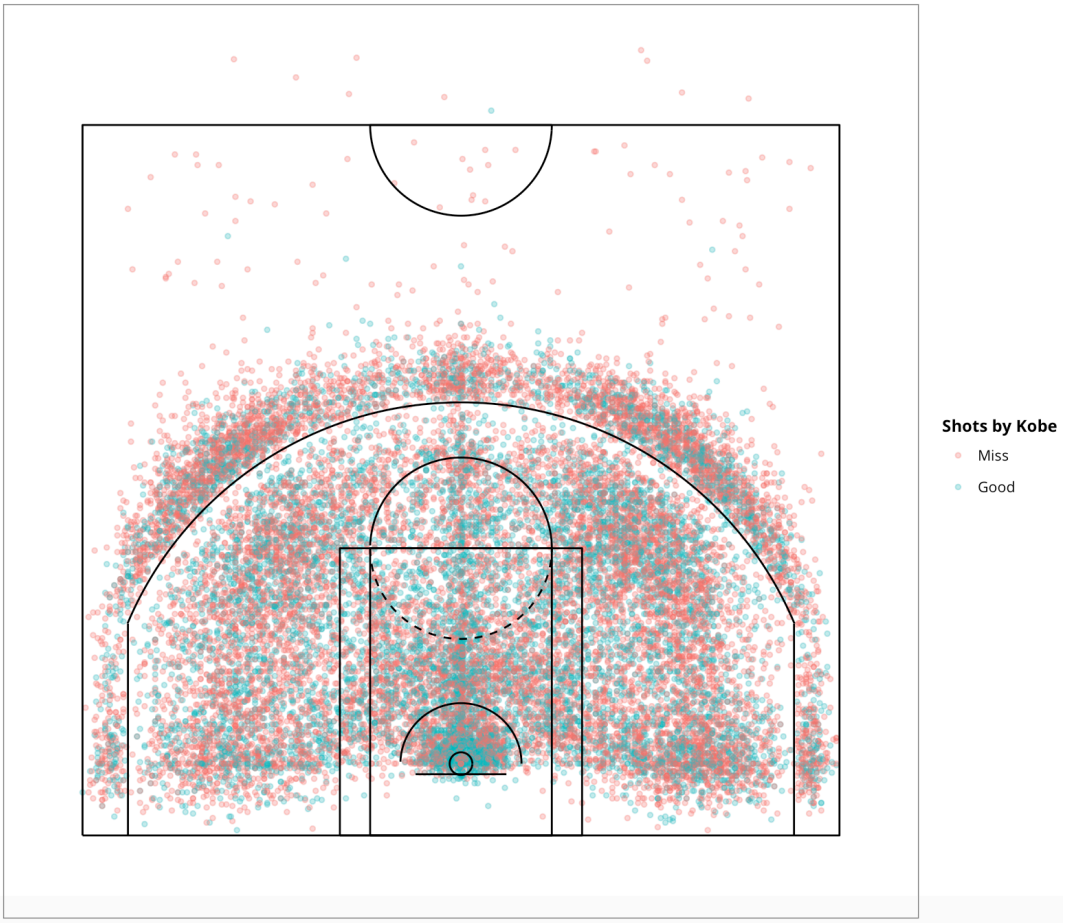

Figure 1: Shot chart of Kobe Bryant about 20 years

<그림 3>은 해당 슛의 모션 정보에 따른 성공률을 클리블랜드 점 그래프로 시각화한 것이다. 슛의 종류가 57가지로 굉장히 다양하지만, 이 중 24가지의 슛은 시도 횟수가 20번도 되지 않는 슛이므로, 해당 슛들은 많은 카테고리에 의한 복잡한 모델 구축을 피하기 위해 하나로 묶어 기타 카테고리로 설정하였다. 새로운 카테고리를 만들어 성공률을 시각화한 결과물은 <그림 4>와 같다.

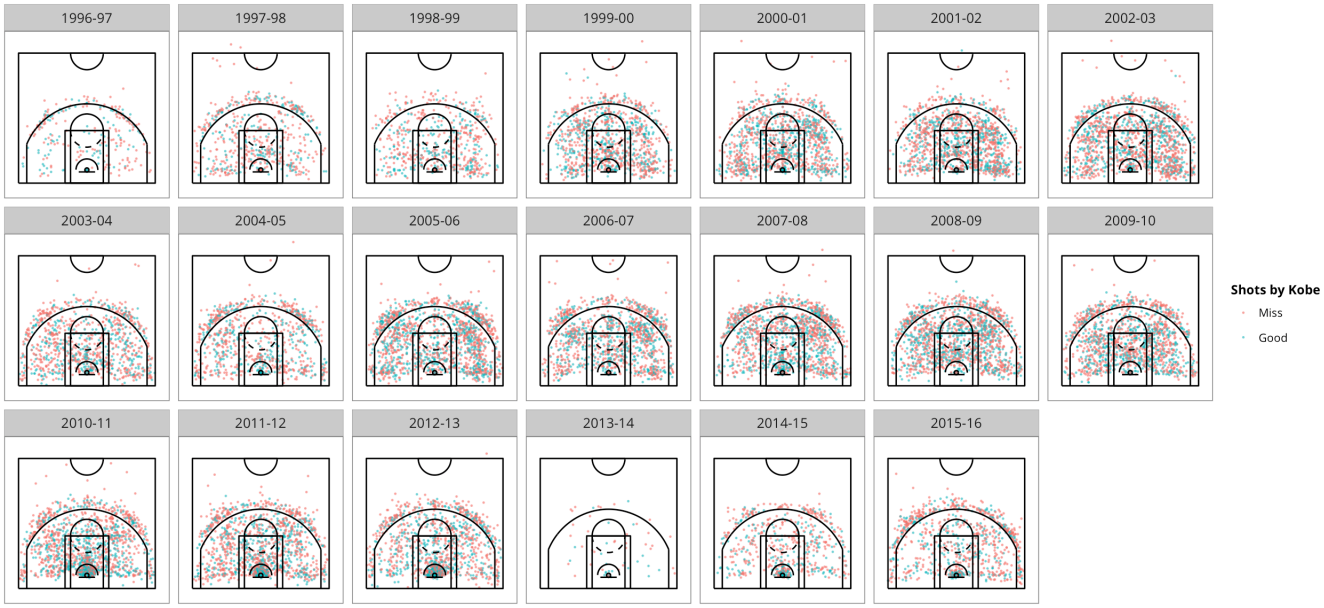

Figure 2: Shot chart for Kobe Bryant about 20 years splitted by season

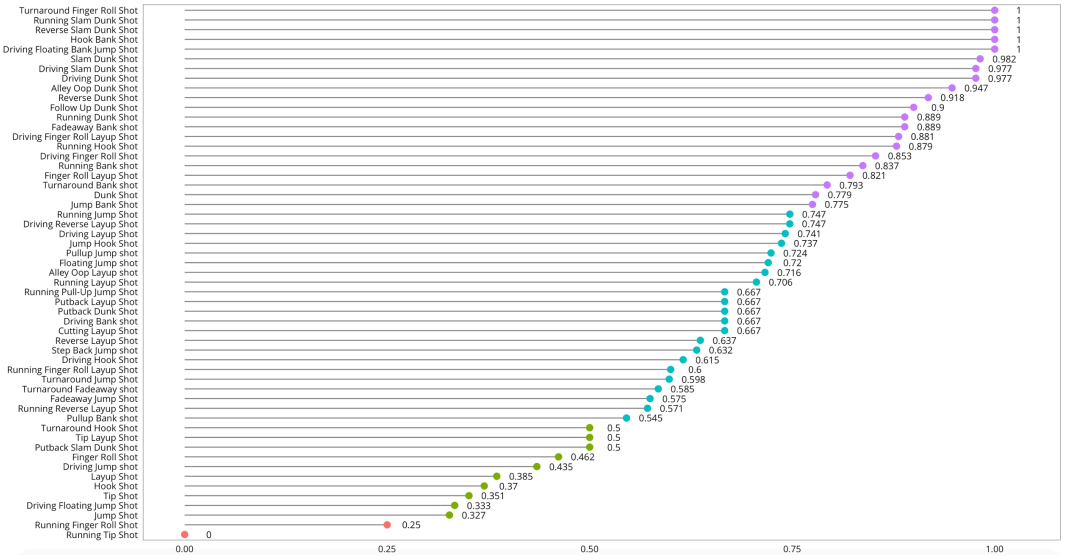

Figure 3: Shot types

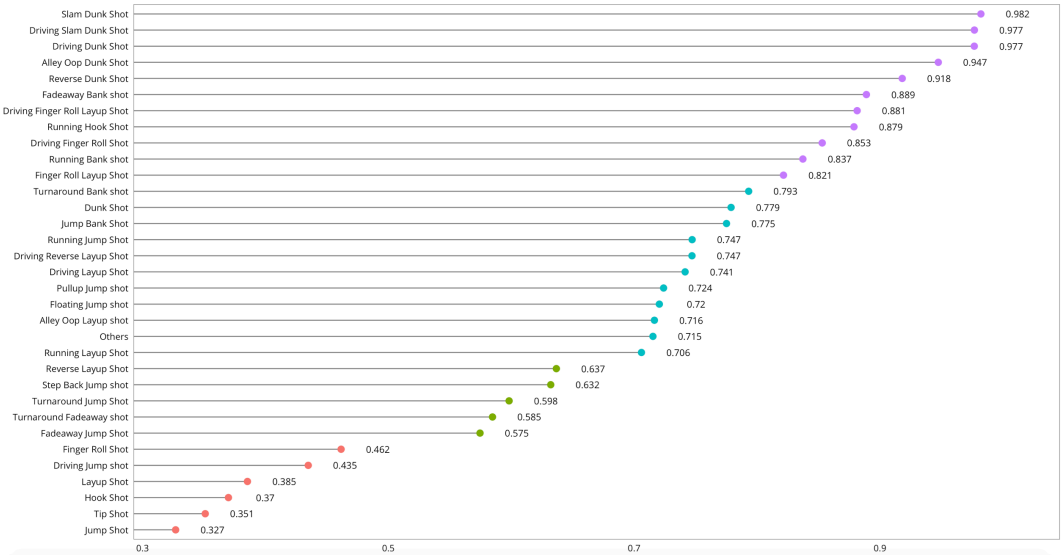

Figure 4: Shot types with rag bag

<그림 5>는 거리에 따른 슛 성공률을 시각화한 결과물이다. 거리가 멀어질 수록 슛 성공률이 급격히 떨어짐을 알 수 있다. <그림 6>은 두 경기 사이의 휴식 기간에 따른 슛 성공률로 7일 이상인 경우 슛 성공률이 가장 낮으며, 6일을 쉰 경우 가장 슛성공률이 높음을 알 수 있다. <그림 7>은 연장전 여부에 따른 슛 성공률이다. 슛 성공률의 차이가 0.01 미만으로 연장전 여부에 따른 슛 성공률 차이는 적으며, 따라서 피처 엔지니어링에 의해 생성된 `overtime` 변수를 폐기하였다. <그림 8>은 쿼터별 슛 성공률을 나타낸 그래프로, 4쿼터에 가장 낮은 슛 성공률을 보임을 알 수 있다. 마지막으로 <그림 9>는 홈 경기와 어웨이 경기 여부에 따른 슛 성공률로, 홈에서의 슛 성공률이 0.02 높은 것을 알 수 있다. 이에 따라 피처 엔지니어링에 의해 생성된 `home` 변수가 유의미함을 알 수 있었으며, 이를 예측 모델에 사용했다.

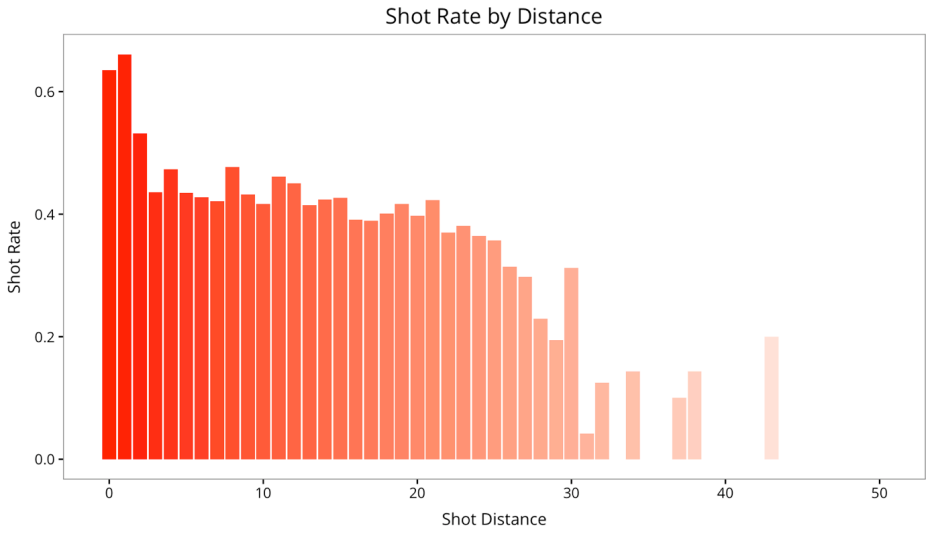

Figure 5: Shot rate by distance

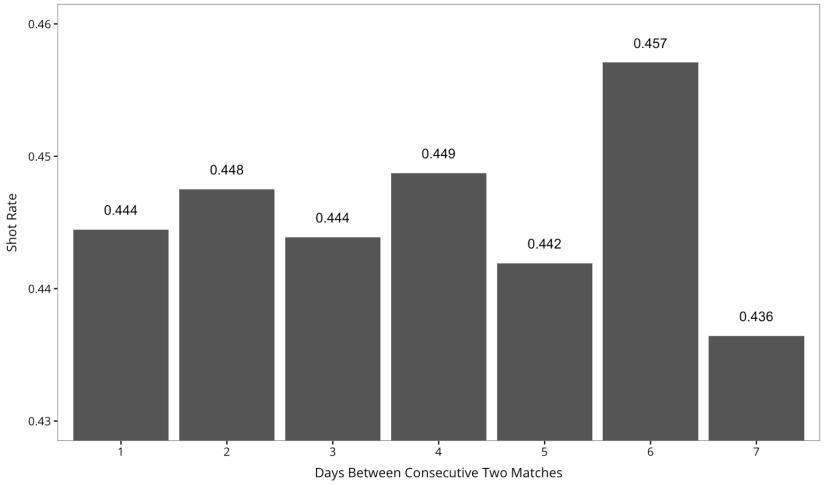

Figure 6: Shot rate by days between two consecutive matches

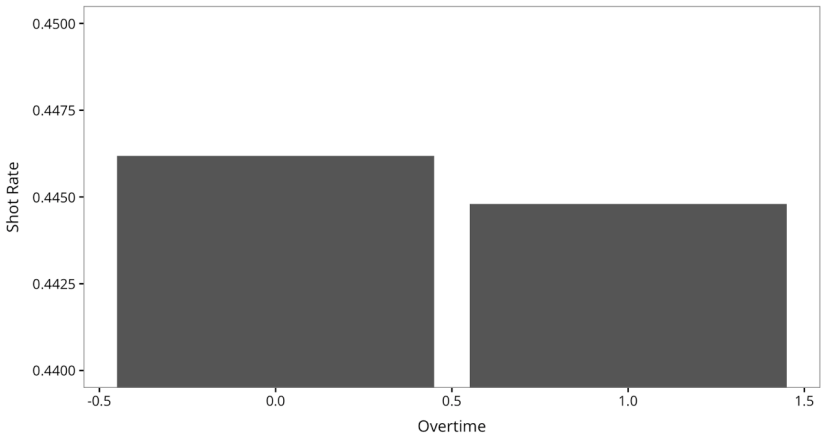

Figure 7: Shot rate by overtime

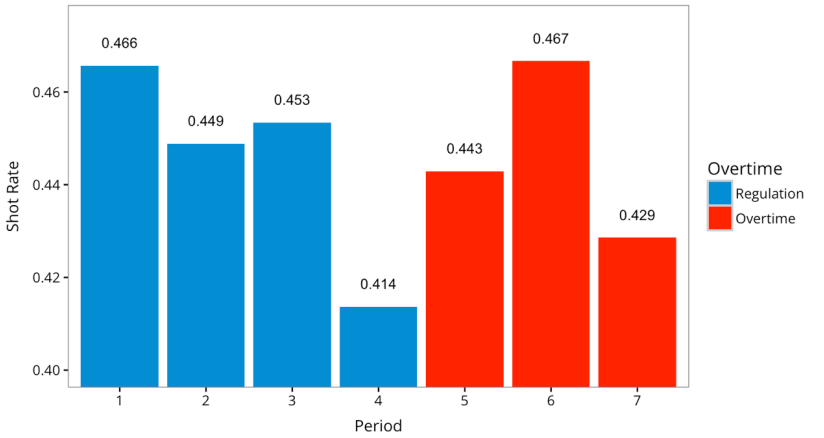

Figure 8: Shot rate by periods

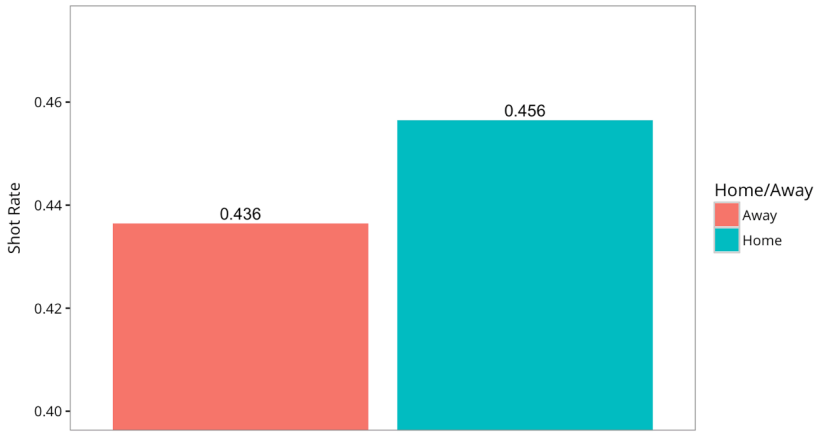

Figure 9: Shot rate by Home/Away

2.3 Gradient Boosting

본 프로젝트에서는 슛 퍼포먼스 예측을 위해 그라디언트 부스팅(Gradient Boosting)을 사용하였다. 그라디언트 부스팅은 앙상블 기법인 부스팅(Boosting)과 기울기 강하법(Gradient Descent)을 결합한 기법이다. 부스팅의 기본 아이디어는 예측력이 약한 예측 모형(weak learner)들을 결합하여 강한 예측 모형을 만드는 것이다. 이 때, t번째 분류기 c_t는 $t-1$번째 분류기인 c_{t-1}와 연관성을 가진 채로 생성된다. 분류기 c_t에서 데이터 집합 X의 샘플을 올바르게 분류한 경우, 정답을 맞춘 샘플은 이미 올바르게 분류했기 때문에 이후의 분류기에서 가중치를 낮추어 분류하고, 잘못 분류한 경우 이후 분류기에서 올바른 분류를 할 수 있게 가중치를 높여준다. 여기서 가중치는 다음 샘플 집합을 뽑는데 중요한 역할을 한다. 즉, 가중치가 높은 샘플이 뽑힐 가능성을 높게 조절하는 정책을 사용한다. Friedman은 부스팅 알고리즘을 최적화 알고리즘의 일종인 기울기 강하법으로 해석하였고, 이를 통하여 지수 손실함수 이외의 다양한 손실함수에서 사용할 수 있는 부스팅 알고리즘인 그라디언트 부스팅을 개발하였다.[7]

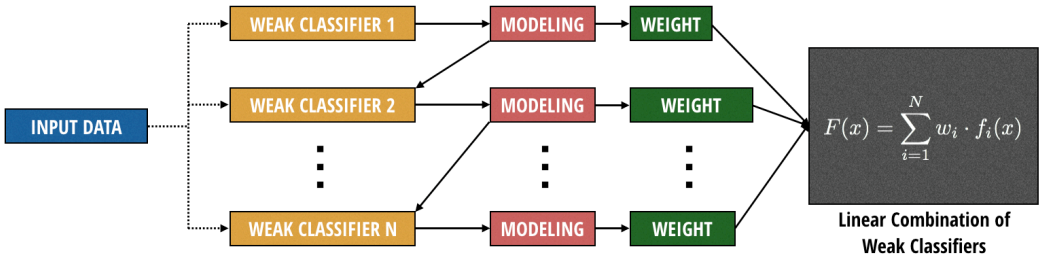

Figure 10: Gradient Boosting

그라디언트 부스팅은 뛰어난 예측력을 바탕으로 캐글(kaggle)과 같은 머신러닝 컴피티션에서 수상한 많은 참가자들이 즐겨 사용하는 알고리즘으로 유명하며, 텐센트(Tencent)와 같은 기업에서도 활발하게 사용하고 있는 머신러닝 기법이다. 하지만 오분류한 데이터 샘플을 중점적으로 학습하는 특성으로 인하여 과적합(overfitting) 문제가 발생할 수 있으므로 파라미터 튜닝(parameter tuning)이 반드시 필요하다는 단점이 있다. 특히 학습률과 반복 횟수와 같은 하이퍼파라미터(hyperparameter)의 설정이 그라디언트 부스팅 기법의 성패를 좌우한다. 따라서 본 프로젝트에서는 다양한 파라미터들에 대해서 최적의 모델을 찾기 위해 학습률, 반복 횟수, 트리의 최대 깊이 등의 설정을 바꿔가며 예측 모델을 수립하였다. 본 프로젝트의 예측 모델 수립은 R을 이용하여 수행되었으며, `xgboost` 패키지를 활용하였다.[8]

2.4 Logarithmic Loss Function

예측 모델을 평가하기 위해서 로그손실함수(Logarithmic Loss Function)를 사용하였다. 로그손실함수는 베르누이 임의 분포에서 우도 함수를 로그로 나타낸 값이다. 일반적으로 분류 모델에서 결괏값이 확률값으로 나타나고 실제 결과값이 참과 거짓으로 각각 1, 0으로 나타내질 때, 실제값과 확률값의 오차를 확인할 때 사용한다. 함수식은 (1)과 같다. 전체 데이터 수 N, i번째 데이터의 실제값 y_i, i번째 데이터에 대한 예측값 p_i에 대하여

$$LogLoss = -\frac{1}{N}\sum_{i=1}^{N}(y_i \log(p_i) + (1-y_i)\log(1-p_i)) \tag{1}$$

이다. 일반적으로 벤치마크를 위해 예측값을 0.5로 설정하였을 때의 로그손실함수값인 0.69315를 기준값으로 한다. 이 때, 실제값이 1일 때 예측값이 0이거나, 실제값이 0일 때 예측값이 1인 경우 음의 무한대로 발산하기 때문에 실제 구현할 때는 굉장히 작은 값을 사용하여 구간을 유계(bounded)로 두었다.

```
MultiLogLoss <- function(act, pred){
        eps = 1e-15;
        nr <- nrow(pred)
        pred = matrix(sapply( pred, function(x) max(eps,x)), nrow = nr)
        pred = matrix(sapply( pred, function(x) min(1-eps,x)), nrow = nr)
        ll = sum(act*log(pred) + (1-act)*log(1-pred))
        ll = ll * -1/(nrow(act))
        return(ll);
}
```

3 Result

최초 예측 모델에 사용한 변수는 총 15개로, `action_type`, `combined_shot_type`, `game_event_id`, `period`, `minutes_remaining`, `seconds_remaining`, `playoffs`, `season`, `shot_distance`, `shot_zone_basic`, `shot_zone_area`, `opponent`, `day_difference`, `home`, `angle`이다. 하이퍼파라미터 설정은 <표 3>과 같다.

Table 3: Hyperparameters

	최솟값	최댓값	단위
학습률 (Learning Rate)	0.02	0.05	0.01
학습 반복횟수 (Iterations)	5	155	5
최대 트리 깊이 (Max Depth)	5	10	1

학습률이 높고, 트리의 깊이가 깊고, 학습 반복횟수가 많을 수록 과적합이 발생할 가능성이 높아지기 때문에, 예측력에 있어서 이상의 하이퍼파라미터들의 설정들이 매우 중요하다.[8] 하이퍼파라미터들에 대한 그라디언트 부스팅 모델의 로그손실함수값은 다음과 같다.

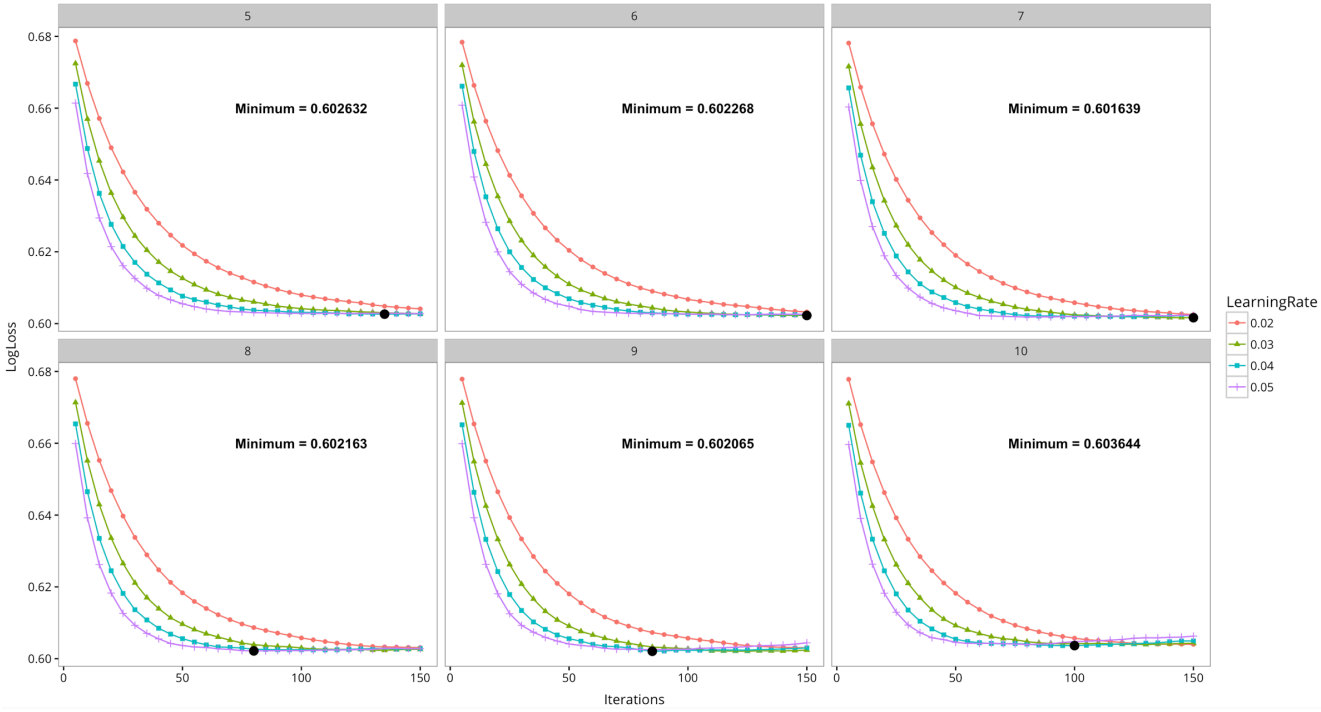

Figure 11: Log Loss of Prediction Model with Gradient Boosting

가장 좋은 예측력을 보여준 모델은 학습률 0.04, 학습 반복횟수 150, 최대 트리 깊이 7로 설정한 모델이었으며, 로그손실함수값은 0.601639였다. <그림 11>을 통하여 로그손실함수값은 최솟값을 기록한 후 과적합으로 인해 다시 증가하는 것을 볼 수 있다. 또한 학습률이 낮을 수록 로그손실함수값의 감소율이 매우 낮았으며, 높을 수록 감소율이 높은 반면 과적합이 빠르게 나타났다. 최대 트리깊이의 경우, 예측력에 큰 영향을 미치지 않지만 트리의 깊이가 깊을 수록 과적합이 더 빠르게 나타나는 것을 마지막 그림을 통하여 알 수 있다.

예측 모델에서 중요한 변수는 <그림 12>와 같다. `action_type`, `combined_shot_type`, `shot_distance`, `game_event_id`, `angle`등의 변수는 중요한 변수로 나타났지만, 반대로 `shot_zone_area`, `shot_zone_basic` 등은 예측에 있어서 중요하지 않은 변수로 나타났다. 슛의 거리와 각도, 슛의 종류는 슛 퍼포먼스에서 굉장히 중요한 부분을 차지하지만, 그보다 큰 범주에 해당하는 `shot_zone_area`, `shot_zone_basic`은 예측력 관점에서 큰 영향을 미치지 못함을 알 수 있다.

Figure 12: Feature Importance

3.1 Improving the Model

예측 모델 개선을 위하여 특성 선택(Feature Selection)과 파라미터 튜닝 과정을 거쳤다. 중요하지 않은 변수로 나타난 `shot_zone_area`와 `shot_zone_basic` 변수를 삭제하고, 가장 좋은 예측력을 보여줬던 모델의 하이퍼파라미터인 학습률 0.04, 최대 트리 깊이 7을 적용하여 학습 반복횟수에 따른 예측력을 확인했다. 결과는 <그림 13>과 같다. 학습 반복횟수가 105번일 때, 로그손실함수값이 0.600639로 가장 예측력이 높은 것으로 나타났다.

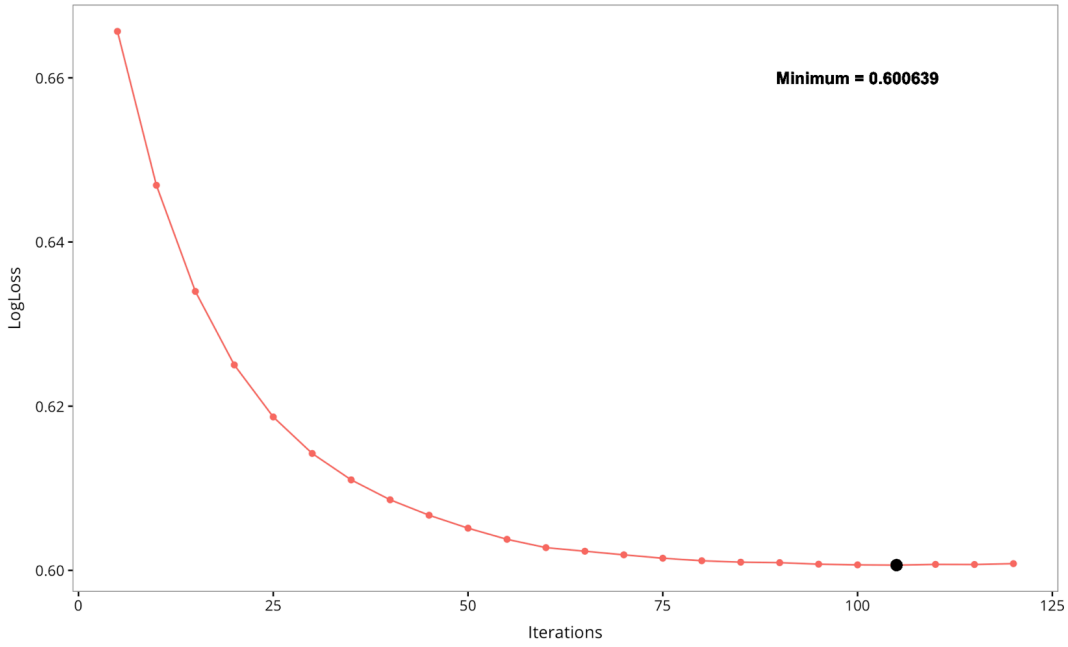

Figure 13: Hyperparameters for Best Performance

4 Discussion

본 프로젝트에서는 상황적인 요소들을 이용하여 특정 농구 선수의 슛 퍼포먼스를 예측하였으며, 그 결과 로그손실함수값은 0.60064로 벤치마크값인 0.69315보다 훨씬 뛰어난 예측력을 보이는 모델을 얻어냈다. 본 프로젝트를 통해 알 수 있었던 점들은 다음과 같다.

그라디언트 부스팅을 통한 예측 모델을 수립할 경우, 교차검증법이나 Brute-Force한 방법을 통하여 최적의 하이퍼파라미터들을 얻어내는 것이 중요하다. 부스팅의 특성 상, 특정 데이터 샘플에 가중치를 주어 모델을 수립하는 경우 해당 데이터 샘플을 지나치게 학습하여 일반적인 샘플에 적용하기 어려워질 수 있다. 따라서 이와 관련된 하이퍼파라미터 값들에 변화가 생길 경우, <그림 11>처럼 과적합이 빠른 시기에 나타나거나 예측력에 큰 변화가 생길 수 있다.

슛 퍼포먼스 예측에 있어서 <그림 12>를 통하여 슛의 종류, 슛의 거리, 림과 슛을 쏜 위치 사이의 각도, 홈 경기 여부, 이전 경기로부터 당일까지의 휴식 기간 등이 매우 중요한 요소임을 확인할 수 있었다. 이 중에서 슛의 거리, 각도, 홈 경기 여부, 휴식 기간은 선수의 특성과 관련 없는 일반적인 상황적 요소이기 때문에, 다른 선수들의 슛 퍼포먼스를 예측할 때도 중요한 특성으로 활용할 수 있을 것으로 예상된다. 또한 피처 엔지니어링을 통하여 얻어낸 `day_difference`, `home`, `angle` 등의 변수가 슛 퍼포먼스 예측에 중요한 요소로 작용한다는 점에서, 이후 유사한 예측 모델을 구축할 때 보다 높은 예측력의 모델을 수립하는 데 도움이 될 것으로 생각한다.

특정 선수의 데이터를 활용하여 수립한 슛 퍼포먼스 예측 모델은 다른 선수들에게 바로 적용할 수 없다는 문제가 있다.

일반적인 상황적 요소는 다른 예측 모델에도 활용 가능할 수 있지만, 선수의 특성을 나타내는 슛의 종류와 같은 특성은 예측에 있어서 문제를 발생시킬 수 있다. 따라서 일반적인 슛 퍼포먼스 예측 모델을 수립하기 위해서는 특정 포지션에 국한하여 다양한 선수들의 슛 데이터를 수집하여 활용하여야 할 것으로 보인다.

5 Reference

[1] Lewis, M., "Moneyball: The Art of Winning an Unfair Game", Norton, 2003.

[2] http://bookyoon.dothome.co.kr/g5/

[3] Oh, Y. *et al.*, "데이터마이닝을 활용한 한국프로야구 승패예측모형 수립에 관한 연구", 대한산업공학회지, 40(1), pp.8-17, 2014.

[4] Lucey, P. *et al.*, "Assessing Team Strategy Using Spatiotemporal Data.", Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp.1366-1374, 2013.

[5] Milijkovic, D. *et al.*, "The Use of Data Mining for Basketball Matches Outcomes Prediction.", Intelligent Systems and Informatics (SISY), 2010 8th International Symposium on IEEE, pp.309-312, 2010.

[6] Stoudt, S. *et al.*, "In Pursuit of Perfection: An Ensemble Method for Predicting March Madness Match-Up Probabilities", 2014.

[7] Friedman, J. H., "Greedy Function Approximation: A Gradient Boosting Machine", Annals of Statistics, pp.1189-1232, 2001.

[8] https://xgboost.readthedocs.org