Term Project: Final Report

13조 배민성 장지윤 강현민

I. 주제 선정 및 문제 정의

1. 주제

NBA 슈팅 로그 데이터를 활용하여 NBA 선수의 슈팅이 득점으로 이어질지 예측하는 기계학습 모델 제작

2. 주제 선정 배경 및 목적

농구 경기에서의 승리 방법은 슈팅을 통한 득점으로 점수를 많이 획득하는 것이다. 슈팅은 자유투, 2점 슛, 3점 슛 총 3가지로 각 슈팅 별로 팀이 얻게 되는 득점의 양이 다르다. 또한 슈팅이 이루어지는 위치, 방식, 거리 등에 따라 다양한 유형의 슈팅이 존재한다. 그리고 이러한 변수들은 슈팅 성공률에 직접적인 영향을 준다. 이에 세계 최대의 농구 리그인 NBA에서 한 시즌 동안 던져진 약 13만 개의 슈팅 데이터를 이용하여, 여러 상황 속에서 슈팅에 영향을 주는 변수들을 알아보고 이러한 변수들을 활용해서 특정 상황에서의 슈팅이 득점으로 이어질 수 있을 것인지를 예측하려 한다.
슈팅 위치와 림과의 거리, 슈팅 시 남아 있던 게임 클락 및 샷 클락, 슈팅 전까지의 드리블 횟수 등 선수의 슈팅에 영향을 미칠 수 있는 다양한 변수들이 학습에 사용되는 만큼 선수들의 득점 확률이 가장 높은 상황을 예측할 수 있다. 이러한 결과는 선수들의 훈련 세션에 도움이 될 수 있다. 뿐만 아니라 상대 선수의 득점 확률 분석을 통하여 상대 선수 대응 방안 마련에도 좋은 효과를 보일 수 있을 것이다. 선수의 득점에 대한 예측은 팀 득점에 대한 예측으로 이어지고, 대진에서의 승패 예측도 가능할 것이라고 예상된다.

3. 활용 데이터 설명 및 머신러닝 방법론의 적합성

주요 데이터로는 Kaggle 사이트에 업로드되어있는 NBA 2014-2015 시즌 슈팅 로그 CSV 데이터를 사용한다(https://www.kaggle.com/dansbecker/nba-shot-logs). 해당 데이터에는 2014-15 시즌의 NBA 경기에서 선수들이 던진 모든 슈팅과 슈팅 관련 정보들이 기록되어 있다. 대표적인 정보로는 슈팅 선수, 슈팅 위치와 림 간의 거리, 슈팅 시 남은 게임 클락 등이 있다. 해당 데이터셋은 실제로 슈팅 성공/실패 여부에 주요한 영향을 미칠 수 있는 다양한 21개의 변수로 구성되어 있다. 해당 데이터는 슈팅에 직접적으로 영향을 미칠 수 있는 다양한 변수가 존재하는 정형 데이터로서 머신 러닝을 통한 예측 및 예측 결과에 대한 해석을 진행하기 용이한 구조를 가지고 있고, 약 13만 개라는 충분한 양을 가지고 있기에 머신 러닝 방법론을 적용하는 것에 대한 적합성이 높다고 판단했다.

4. 문제 상황 정의

상단에 서술된 NBA 2014-2015 시즌 슈팅 로그 CSV 데이터는 총 128069행 21열의 데이터로서, 2014-2015 시즌 동안 NBA 리그 내에서 모든 선수들이 시도한 128069개의 슈팅에 대해 해당 슈팅과 관련된 21개의 feature가 존재한다. 해당 feature들이 나타내는 의미는 다음과 같다 :

1. game_id : 해당 슈팅이 발생한 게임의 고유 번호
2. matchup : 해당 슈팅이 발생한 게임의 팀명 및 연/월/일
3. location : Home / Away로 해당 슈팅을 던진 선수가 속한 팀이 홈 팀인지 원정 팀인지를 나타냄
4. w : 해당 슈팅을 던진 선수가 속한 팀의 최종 승패 여부 (W/L)
5. final_margin : 해당 슈팅을 던진 선수가 속한 팀의 최종 점수 차
6. shot_number : 해당 슈팅의 번호 (선수가 한 경기에 던진 슛이 1부터 numbering되어 있음)
7. period : 해당 슈팅이 발생한 쿼터 (1~4쿼터, 이후 연장 쿼터는 5,6 … 등으로 처리)
8. game_clock : 해당 슈팅이 일어난 후 남아있던 게임 클락
9. shot_clock : 해당 슈팅이 일어난 후 남아있던 샷 클락
10. dribbles : 해당 슈팅을 던진 선수가 공을 받은 후 슈팅을 던지기 전까지 행한 드리블의 횟수
11. touch_time : 해당 슈팅을 던진 선수가 공을 받은 후 슈팅을 던지기 전까지 공을 소유하고 있었던 시간 (초)
12. shot_dist : 슈팅을 던진 위치와 림 간의 거리 (ft)
13. pts_type : 슈팅의 종류 (2점 / 3점)
14. shot_result : 슈팅의 결과 (made / missed)
15. closest_defender : 해당 슈팅을 던진 선수와 가장 인접해있던 상대 팀 수비수의 이름
16. closest_defender_id : 해당 슈팅을 던진 선수와 가장 인접해있던 상대 팀 수비수의 고유 번호
17. close_def_dist : 해당 슈팅을 던진 선수와 가장 인접해있던 상대 팀 수비수와의 거리 (ft)
18. fgm : 해당 슈팅의 결과 (1 / 0)
19. pts : 해당 슈팅을 통해 창출된 득점 (0 / 2 / 3)
20. player_name : 해당 슈팅을 던진 선수의 이름
21. player_id : 해당 슈팅을 던진 선수의 고유 번호

해당 데이터셋을 활용하여 다음 feature들을 통해 슈팅의 성공 여부를 예측하는 분류 모델을 제작하는 것을 목표로 한다. 그렇기에 여기서 예측 대상이 되는 target variable을 fgm으로 설정할 수 있을 것이다.

II. 적용 방법론 및 결과

1. EDA를 통한 Data Preprocessing

EDA (Exploratory Data Analysis)는 데이터 시각화 및 조사를 통해 데이터의 기본적인 구조 및 분포를 파악하고 결측치나 이상치를 제거하기 위해 탐색적으로 데이터를 분석하는 방법을 일컫는 용어이다. 이번 프로젝트에서는 데이터의 시각화 및 조사를 통해서 사용하고자 하는 데이터에 존재하는 결측치들을 제거하고, 데이터에 존재하는 비정상적 데이터를 처리하였으며, 높은 상관 관계를 가지는 변수들을 선택적으로 제거하거나 정규화하는 등의 다음과 같은 데이터 전처리 과정을

진행하였다.

1) 데이터 변환

본격적인 데이터 처리를 진행하기 전 학습에 활용하거나 처리하기 어려운 text 형태의 범주형 데이터를 int type으로 바꿔주는 과정을 진행하였다. 총 3개의 text 형태의 범주형 변수 (location, w, shot_result)에 대하여 해당 변수들을 모두 1/0의 binary 변수로 변환해주었다. 또한, (분 : 초) 형태로 되어 있어 학습에 활용하기 어려운 game_clock 변수의 경우 60 * 분 + 초의 연산을 통해 단순 정수 형태 (초)로 변환해주었다.

2) 결측치 (Nan) 탐색 및 imputation

현재 사용하고자 하는 데이터의 shot_clock 변수에 총 5567개의 결측치 (Nan)이 있는 것을 확인할 수 있었다. 농구 경기에서는 게임 클락이 24초 아래로 내려갈 경우, 샷 클락이 더 이상 흐르지 않는다. 결측치가 들어있는 행 내에서 game_clock 변수 값을 확인해본 결과, 24초보다 큰 값을 가지는 행이 총 2013개 존재하였고, 나머지 3554개의 행은 실제로 game_clock이 24초보다 작은 값을 가지고 있었다. 이 경우에는 게임 클락과 샷 클락이 동일하다고 판단할 수 있기 때문에, shot_clock이 Nan이면서 game_clock이 24초 이내인 경우에는 해당 행의 game_clock 변수 값으로 shot_clock 변수의 값을 바꿔주었다. 그리고 이 과정에서 shot_clock이 Nan이며, game_clock이 24초 보다 큰 행들은 데이터에 오류가 있다고 판단하고 삭제하였다.

3) 비정상 데이터 탐색 및 처리

다양한 조건에 따라 실제 상황에서는 존재할 수 없는 비정상적인 데이터가 존재하는지 확인하고, 존재하는 경우 이를 해결할 수 있도록 데이터 전처리 과정을 진행하였다.
먼저, 게임 클락이 샷 클락보다 큰 경우는 실제 농구 경기 상에서 존재할 수 없다. 이러한 데이터가 존재하는지 확인해본 결과 총 13개의 데이터에서 이와 같은 현상이 나타났는데, shot_clock의 소수점 값으로 인해 shot_clock 변수의 값이 game_clock 변수의 값보다 큰 현상이 나타나게 된 것이었다. 13개의 데이터 모두 shot_clock 변수의 값을 소수점 아래 버림했을 때의 값은 game_clock 변수의 값과 동일하여 오류라고 판단하지 않고 별도의 처리를 진행하지 않았다.
게임 클락과 샷 클락의 값이 정상 범위를 벗어나는 경우(0보다 작은 값이거나 24초 혹은 정규 쿼터 시간인 12분(720초)보다 큰 값)나 드리블 횟수가 0보다 작은 값을 가지는 등의 비정상적인 데이터는 발생하지 않은 것을 확인하였다.
다만, 해당 슈팅을 던진 선수가 공을 받은 후 슈팅을 던지기 전까지 공을 소유하고 있었던 시간을 나타내는 touch_time 변수의 값이 0보다 작은 데이터가 총 305개 존재하여, 이에 대한 처리가 필요했다. 일단, 해당 데이터가 잘못 기록된 데이터이거나 실제로는 존재하지 않는 데이터일 가능성을 확인해보기 위해 하나의 (748번) 행에 대해 대표로 해당 행의 matchup, period, game_clock 변수를 활용하여 해당 경기의 해당 쿼터 및 시간에 발생한 슈팅이 존재하는지 NBA 공식 사이트에서 확인해보았을 때, 슈팅이 실제로 존재하는 것은 확인할 수 있었다.

(https://www.nba.com/stats/events?CFID=&CFPARAMS=&GameEventID=429&GameID=0021400625&Season=2014-15&flag=1&title=Jefferson%201%27%20Layup%20(10%20PTS) / JAN 21, 2015 - CHA vs. MIA)
해당 슈팅의 경우 영상을 확인해보았을 때 리바운드 이후 곧바로 슈팅을 진행하였기 때문에 공을 들고 있는 시간을 측정하기에는 어려움이 있는 상황이었다. 따라서 touch_time 변수의 값이 0보다 작은 총 305개의 데이터에 대하여, 다음과 같은 상황에 의해 데이터에 오류가 발생한 것으로 판단하고 해당 데이터의 touch_time 변수 값을 0으로 변경해주었다.
또한, NBA의 경우 3점 슛 라인과 림 간의 거리가 양측면 22피트, 이외의 경우 23.9피트인데, shot_dist 변수의 값이 22피트 미만인 데이터 중에서 pts_type이 3으로 분류되어 있는 데이터들이 301개 존재하였다. 이러한 데이터들의 경우 위에서와 마찬가지로 해당 행의 matchup, period, game_clock 변수를 활용하여 해당 경기의 해당 쿼터 및 시간에 발생한 슈팅 5개에 대해 확인해보았을 때 모두 실제로는 3점 슛 라인 바깥에서 던진 슈팅인 것을 확인할 수 있었다.
(https://www.nba.com/stats/events?CFID=&CFPARAMS=&GameEventID=152&GameID=0021400071&Season=201415&flag=1&title=Neal%2024%27%203PT%20Jump%20Shot%20(10%20PTS)%20(Roberts%202%20AST)
따라서, 이는 shot_dist 변수의 기록에서 오류가 발생했으며 5개의 슈팅 모두 양측면 부근에서 이루어진 슈팅이었기 때문에 해당 301개의 행에 대한 shot_dist 변수를 22 (ft)로 변경해주었다.

4) feature 분석에 따른 feature selection

해당 데이터에는 20개의 활용 가능한 feature가 존재하였지만, 이 feature들을 모두 활용하기에는 redundant하거나 실제로 학습에 사용하기에는 어려운 feature들이 존재하였기에, 분석을 통하여 여러 가지 feature들 중 실제 활용할 feature들을 selection하는 과정을 진행하였다.
먼저, 다른 feature들과 유사한 의미를 지니고 있는 redundant한 feature들이 존재하였다. 두 변수 shot_result와 fgm의 경우 shot_result 변수를 1/0 binary 범주형 변수로 변환하고, fgm 변수와 값을 대조해봤을 때 완벽히 일치하는 것을 확인할 수 있었다. 또한 pts_type 변수와 pts 변수의 경우 pts 변수는 해당 슈팅을 통해 창출된 득점을 값으로 가지는 변수이기 때문에 0도 포함할 수 있다는 것을 제외하면, 실제로 2 또는 3으로 항상 같은 값을 나타내는 것을 확인할 수 있었다.
이후 데이터 시각화 및 분석, 학습에 활용하기 위한 feature들을 선택하는 과정에서 다음 변수들은 제거하였다.
- game_id, closest_defender_player_id, player_id : 고유 번호 값의 경우 학습에 사용하기에는 부적절하고 시각화나 분석에 활용할 수 있는 가능성도 낮기 때문에 활용성이 낮다고 판단하였다.
- matchup : 유효한 데이터인지 확인하는 과정에서 활용되었으나 추후 시각화나 분석, 학습에 활용하기에는 활용성이 낮다고 판단하였다.
- w, final_margin : 최종 승/패 결과와 팀 별 득점 차이가 모든 개별적인 슈팅에 영향을 주는 요소라고 보기에는 어렵다고 판단하였다.
- shot_number : 선수 별로 슈팅 예측을 진행하는 것도 아니며 특별한 의미를 가지는 값도 아니기 때문에 활용하기엔 어렵다고 판단하였다.
- shot_result : fgm과 완벽히 동일한 redundant한 변수이다.

- pts : pts_type과 매우 유사하며 시각화나 분석 과정에서는 0이 섞여 있는 pts보다는 2와 3으로만 구분되어 있는 pts_type이 활용성이 더 높다고 판단하였다.

5) feature별 데이터 시각화 및 분석

각 feature별로 데이터 시각화를 통해 해당 feature의 값이 어떤 분포를 가지는지 확인해보고 그 값을 분석해보았다.
NBA에서는 기본적으로 한 쿼터당 12분씩, 총 4개의 쿼터로 게임이 진행되지만, Period 변수가 5 이상의 값을 가진다는 것을 확인할 수 있다. 만약 4번째 쿼터 혹은 그 이상의 쿼터에서 게임이 동점으로 끝나게 된다면, 게임의 승패를 가르기 위해, 승패가 결정될 때까지 새로운 쿼터가 진행된다. 그러므로 Period 변수의 4 이상인 값은 결측치 혹은 이상치라고 간주하지 않아도 된다.
Plot을 통해 Game_clock 값이 첫번째 구간에서 다른 구간보다 조금 높은 빈도수를 보이고, 마지막 구간에서 다른 구간보다 조금 낮은 빈도수를 보이지만 대체로 비슷한 빈도수를 보인다고 판단할 수 있다. 따라서 Game_clock 변수는 거의 균등 분포를 따르며 선수들의 슈팅이 남은 경기 시간과 상관없이 이루어진다고 해석할 수도 있다.
Plot을 통해 볼 수 있듯이, 2점 득점 데이터의 수가 3점 득점 데이터 수의 대략 3배라고 볼 수 있다.
Shot_dist 변수의 plot을 통해 알 수 있듯이, 거릿값은 서로 다른 두 개의 최빈값을 가지므로, 쌍봉분포를 가진다. 2점 득점과 3점 득점의 규정 거리가 서로 다르므로, 이러한 분포가 나타났다고 이야기할 수 있다.
농구에서 공격 팀은 반드시 24초 이내에 슛을 시도해야 하므로, 데이터의 최댓값은 24이다. Shot_clock 변수는 중앙에 최빈값을 가지고 거의 좌우 대칭인 종형의 분포를 가지므로 정규분포로 근사할 수 있을 것이다. 이때 최댓값 부근에서 특이하게도 높은 빈도수를 보이는데, 아마도 슈팅 이후의 리바운드로 이러한 분포가 나타났을 것이다.
Plot을 통해 알 수 있듯이, 대부분의 수비수가 슈팅하는 선수와 대략 5ft 정도의 거리를 두는 경향이 있다.
Plot을 통해 알 수 있듯이, 상당히 대부분의 슈팅이 적은 드리블 이후에 시도되었으며, 이에 따라 공을 가지고 있는 시간도 대부분 최솟값 구간에 몰려서 있다. 대부분의 팀은 보통 많은 패스를 주고받으며 개인의 공 점유 시간을 짧게 가져가는 역동적인 패턴으로 공격을 진행하기에, 이러한 결과가 나왔다고 판단할 수 있다.

6) feature와 target variable 간의 상관 관계 분석

x축을 fgm 변수로, y축을 close_def_dist 변수로 설정한 box plot을 통해 선수 슈팅의 결과와 슈팅 지점과 가장 가까운 수비수 사이의 거리 관계를 알아보았다. 선수의 슈팅이 성공했을 때의, 즉 fgm 값이 1일 때의 close_def_dist 분포가 훨씬 많은 이상치를 가진다는 것을 알 수 있다.
선수 슈팅의 결과와 슈팅 거리, 공 점유 시간, 드리블 횟수 변수 사이의 관계를 box plot을 통해 알아보았다. 선수의 슈팅이 성공했을 때가 실패했을 때보다 짧은 슈팅 거리, 공 점유 시간, 드리블 횟수를 가진다는 것을 알 수 있다.

선수 슈팅의 결과와 슈팅 시간, 득점의 종류와 가장 가까운 수비수와의 거리의 관계를 box plot을 통해 알아보았다. 3점 슛을 통한 득점은 가장 가까운 수비수와 슈팅하는 선수 사이의 거리가 멀수록 훨씬 빈번하게 이루어진 것을 알 수 있다. 또한 슈팅은 샷 클락이 많이 남았을 때 성공 빈도가 높음을 알 수 있다.

7) feature간의 correlation 분석

산포도를 통해 알 수 있듯이, dribbles와 touch_time은 0.93이라는 양의 선형 관계를 가진다. 많은 드리블 횟수를 기록하기 위해서는 당연하게도 선수가 공을 점유하고 있는 시간이 높아야 한다. 그러므로 dribbles와 touch_time은 높은 선형 상관계수 값을 가진다.
산포도를 통해 shot_dist와 close_def_dist 변수는 0.52라는 어느 정도 양의 선형 상관계수 값을 가진다고 볼 수 있다. 경기에서 선수들이 수비를 진행할 때, 대부분의 수비수는 공격수보다 골대와 거리를 가까이할 것이다. 따라서 '골대와의 거리'와 '가장 가까운 수비수와의 거리'가 양의 선형상관관계가 가진다고 일부 설명할 수 있을 것이다. 하지만 리바운드 혹은 역습과 같은 다양한 경기 상황으로 상관계수 값이 다소 낮아졌다고 볼 수 있다.

8) outlier 탐색 및 제거

데이터에 존재하는 outlier 값들을 처리하기 위해, IQR 값을 이용하였다. 여기서 IQR이란, Q3 (전체 데이터에서 25%로 높은 값)와 Q1 (전체 데이터에서 75%로 높은 값)의 차이를 의미하며, 여기서 IQR에 따른 outlier는 Q3 + 1.5 * IQR 보다 높거나, Q1 – 1.5 * IQR보다 낮은 값으로 정의된다. 이에 따라 한 feature에서 outlier에 해당하는 (true) 데이터의 비율이 0.05를 넘으면 그 feature에 대해서는 outlier 처리를 위한 process를 진행하였다. outlier에 해당하지 않는 데이터 (false) 의 비율이 1이 아닌 feature는 'period', 'shot_clock', 'dribbles', 'touch_time' 이 있었으며, 여기서 true 비율이 0.05가 넘는 것은 'dribbles', 'touch_time'로 총 두 개의 feature였다. 이에, 두 feature에서만 outlier 처리를 위한 process를 진행하였다.
outlier replacement의 경우 min-max scaling 방식을 이용하여 진행하였다. min-max scaling 방식은 모든 x에 대해서 $x_{scaled} = (x - x\_\min)/(x\_\max - x\_\min)$ 에 대입한 값으로 변화시킨다. 이는 값의 범위를 0에서 1로 제한하여 데이터를 좁은 범위로 압축하는 효과를 준다. 이에 dribbles, touch_time 두 가지 변수에 min-max Scaling을 적용하여 outlier 값들을 처리해주었다.

2. 추가 데이터셋을 활용한 feature engineering & construction

학습의 주 데이터셋으로 활용하기로 한 데이터셋은 슈팅이 일어난 시점의 슈팅과 관련된 다양한 데이터를 담고 있으나, 해당 슈팅을 수행한 선수나 그 선수를 수비한 상대 팀 선수에 대한 정보가 오직 해당 선수들의 이름으로만 존재하기에 학습에 해당 feature를 반영하기 어렵다. 이에, 이 데이터를 유의미하게 사용할 수 있도록 당시 시즌의 선수들의 슈팅 및 수비 스탯 데이터셋을 추가 데이터셋으로 활용하여 새로운 feature를 만들어내고자 하였다.
추가 데이터셋은 NBA 14-15시즌 각 선수 및 팀 별 개인 스탯이 상세하게 정리되어 있는 웹사이

트인 basketball-reference.com 사이트를 이용하였고, 이 웹사이트에서 data table을 csv 파일로 제공하기에 해당 데이터를 추가 데이터셋으로 쉽게 활용할 수 있었다.
(https://www.basketball-reference.com/leagues/NBA_2015_per_game.html)

이후 선수 별 슈팅 거리에 따른 슛 성공률, 슈팅 시도 횟수, 스틸/블록 성공률, 개인 수비 기여도 등 추가 데이터셋의 다양한 슈팅 확률 및 수비 스탯을 새로운 feature로서 활용하고, 추가적으로 새로운 feature를 제작하였다. 다만, 기존의 main 데이터셋과 추가 데이터셋은 다른 출처로부터 온 데이터셋이기에 같은 의미를 가지는 데이터임에도 불구하고 두 데이터셋의 데이터 사이에 값의 차이가 존재하는 경우가 있었다. 예를 들어 기존 데이터셋에 'Redick, JJ'라고 표기되어 있는 선수명이 추가 데이터셋에서는 'J.J. Redick'으로 표기되어 있거나, 추가 데이터셋에는 스탯이 존재하지 않는 선수가 실제 데이터셋에는 존재하는 경우도 있었다. 이런 경우는 예외로서 별도의 수정 및 처리 과정을 거쳐주었다.

3. 모델 제작 및 학습

Proposal에서 서술한 XGBoost 및 Multi-layer perceptron을 이용한 ensemble method를 통한 본격적인 학습 과정 뿐만 아니라, binary classification task를 해결하기 위해 사용될 수 있는 대표적인 단순한 머신 러닝 알고리즘인 Random Forest 및 Logistic Regression을 활용하여 학습을 진행해보고 그 결과를 확인해보았다.
최종적으로 모델의 학습에는 ['location', 'period', 'game_clock', 'shot_clock', 'touch_time', 'shot_dist', 'close_def_dist', 'STL%', 'BLK%', 'DBPM', 'shot_acc', 'fg'] 총 12개의 feature를 사용하였다.
'STL%', 'BLK%', 'DBPM'은 각각 선수 별 스틸 시도 시 성공 확률, 블락 시도 시 성공 확률, Defensive Box score Plus/Minus를 나타내는 수비 선수의 수비 스탯과 관련된 feature이며, 'shot_acc', 'fg'는 'shot_dist' 데이터 및 선수 별 슈팅 거리에 따른 성공 확률 및 2/3점 슈팅 성공 횟수, 2점 슛과 3점 슛의 득점량을 환산한 순수 슈팅 성공률 등의 슈팅을 시도하는 선수와 관련된 다양한 슈팅 스탯을 활용하여 생성해낸 새로운 feature이다. 해당 feature들은 추가 데이터셋의 feature들을 주로 활용해서 만들어졌다.
기존 데이터셋에 있던 feature 중 'pts_type'의 경우 2와 3만으로 이루어진 단순 범주형 데이터를 학습을 위한 변수에 포함시키기에는 큰 의미가 없고 다른 변수들과 그 의미가 중복된다고 판단하여 제외하였고, 학습에 사용할 수 없는 non-numerical 데이터의 경우도 제외하였다.
따라서 이 12개의 feature를 활용해 target variable인 'fgm'을 예측하는 binary classifier를 제작하였다.
각 모델 모두 데이터를 train/test/validation 총 3개의 데이터셋으로 split하여 학습을 진행하였다. Logistic Regression과 Random Forest 기반 모델의 경우 먼저 validation error를 계산하고 validation error 계산 이후 validation data는 다시 train data와 concatenate하여 다시 train을 진행, 그 이후 test에 활용하였다.

1) Logistic Regression

분류 문제에서 활용 가능한 가장 단순한 머신 러닝 알고리즘 중 하나로, scikit-learn 라이브러리의 LogisticRegression 메소드를 활용하여 구현하였다. Logistic Regression의 경우 각 feature들의 coefficient (weight) 값들을 쉽게 확인해볼 수 있는데, 해당 결과에 따라 다음과 같은 해석이 가능할 수 있다.

- close_def_dist 변수의 경우 해당 값이 클 수록 수비수와의 거리가 멀리 떨어져있다는 것이기 때문에 슈팅 성공 확률이 높아지는데 기여할 것이라고 생각해볼 수 있고, 양의 값을 가지는 것이 타당하다고 볼 수 있다.
- shot_clock 변수의 경우 해당 값이 클 수록 슈팅 당시에 샷 클락이 많이 남아 있는 여유 있는 상태에서 슈팅을 던졌다는 것이기 때문에 슈팅 성공 확률이 높아지는데 기여할 것이라고 생각해볼 수 있고, 양의 값을 가지는 것이 타당하다고 볼 수 있다.
- shot_dist 변수의 경우 해당 값이 클 수록 슈팅 거리가 멀다는 것이므로 슈팅 성공 확률이 낮아지는데 기여할 것이며, 음의 값을 가지는 것이 타당하다고 볼 수 있다.
- weight의 magnitude로 고려해보았을 때, shot_dist / close_def_dist는 학습에서 매우 유의미한 기여를 한다고 볼 수 있다. 반면, period나 game_clock 같은 변수는 유의미한 기여를 한다고 보기는 어려울 가능성이 높다.

2) Random Forest

Decision Tree 기반의 ensemble 모델로서 ensemble 기법을 통해 overfitting을 해결하는 효과를 가지고 있어 classification이나 regression 두 task 모두에서 많이 활용되는 머신 러닝 알고리즘 중 하나이다. scikit-learn 라이브러리의 RandomForestClassifier 메소드를 통해서 구현하였고, GridSearchCV 메소드를 통해 다양한 hyperparameter candidate들에 대해서 grid search 방식으로 hyperparameter tuning을 진행하였다.

3) XGBoost (Extreme Gradient Boosting)

XGBoost는 여러 개의 의사결정 트리를 조합하여 최종 결과를 도출해내는 ensemble 기법 중 boosting 방법을 기반으로 한 알고리즘이다. 이번 프로젝트에서 사용하고자 하는 데이터도 정형 데이터로서 여러 변수들에 따른 슈팅 성공 여부를 예측해야 하기 때문에 해당 방법론을 활용하기로 했다.

모델 제작에서는 XGBoost 라이브러리를 활용하여 단순한 형태의 XGBoost classifier를 제작하여 학습시키는 것을 목표로 하였다. 또한, scikit-learn 라이브러리의 confusion_matrix와 classification_report method를 활용하여 이진 분류가 어떻게 이루어졌는지 확인해보았다.

4) MLP (Multi-layer Perceptron)

여러 개의 perceptron으로 이루어져 있는 층을 연속적으로 붙여 놓은 다층신경망 구조를 가지는

모델로서 회귀/분석 문제를 다루는 가장 기본적이고 대표적인 Deep Learning 방법론이다. MLP는 대규모 데이터를 처리하고, 예측을 진행함에 있어 유연한 예측이 가능하다. 현재 사용하려고 하는 데이터는 다양한 feature를 가지고 있는 대량의 데이터이기 때문에 해당 방법론을 사용했다.
Tensorflow를 기반으로 하는 딥러닝 라이브러리 keras를 활용하여 sequential한 형태의 multi-layer perceptron 기반의 슈팅이 성공할 확률을 예측하는 0/1 classifier를 제작하였고, output layer에서 0~1 사이의 값을 return하는 sigmoid function을 activation function으로 사용하였다.

이와 같은 네 가지의 머신 러닝 알고리즘을 활용한 단순 분류기 모두 validation/test dataset에 대해서 약 62%를 상회하는 정확도를 보여주었다. 이는 validation data의 target variable에 대해서 모든 prediction 결과값을 majority class인 0, 즉 슈팅 실패로 예측하였을 때의 정확도인 54%보다 약 8% 정도 높은 수치이다. 또한 실제로 추가 데이터셋을 통해 생성해낸 feature를 추가하여 학습한 결과 Random Forest 기반 모델 등에 대해 정확도가 중간 결과에서 기존 데이터셋으로만 학습했던 모델보다 약 3-4% 정도 향상된 모습을 보였다.

최종적으로 여러 모델들의 예측 결과를 결합해 최종 결과를 예측하는 방법론인 ensemble method를 활용하여, 앞선 여러 모델들을 기반으로 예측을 진행해보았다. Random Forest / XGBoost / Multi-Layer Perceptron 기반의 3개의 분류 모델의 각각의 예측 결과, 즉 슈팅이 성공할 확률을 나타내는 0~1 사이의 값을 hyperparameter tuning을 통해 얼마나 고려할 지 결정한 이후 weighted sum을 통해 최종 예측을 진행하였다. 그 결과 Random Forest 기반 모델은 ensemble 기법을 통한 성능 향상에 크게 기여하지 않았고, XGBoost와 MLP 기반의 모델을 동시에 기여한 예측 결과는 약 62%로 ensemble 기법을 수행하지 않았을 때와 큰 차이를 보이지는 않았다.

III. 결과 해석 및 추후 연구 개발 방향

1. 결과 해석

실제 학습 및 테스트 결과 모델의 예측 정확도는 기준 정확도로 잡았던 validation 정확도인 54%보다 약 8% 정도 상회하는 62%의 선에서 머물렀다. Raw data에 대한 EDA 및 feature engineering을 통한 data preprocessing과 다층 퍼셉트론이나 XGBoost, Random Forest 등 Tabular data의 classification에 주로 활용되는 다양한 머신 러닝 기법들을 활용하였음에도 불구하고 이에 비해 만족스러운 결과값을 얻지는 못하였다. 그 이유를 다음과 같이 예측하였다.

- 실제 슈팅에는 단순히 슈팅 시 슈터와 림 간의 거리, 수비수와의 거리, 해당 선수의 슈팅 성공률 및 수비수의 수비 능력 등 1차원적인 요소만 관여를 하는 것이 아니라 양 팀의 점수차, 1차원적인 수비수와의 거리 뿐만 아니라 슈팅 시 해당 선수가 얼마나 상대 팀으로부터 자유로운 상황인지 등 굉장히 많고 다양한 심리적/물리적 요인이 존재하기에, 단순한 feature들만으로는 예측이 어려울 가능성이 있다.
- main 데이터셋의 데이터가 feature engineering 및 EDA 과정에서 확인할 수 있듯이 noise가 기본적으로 많이 존재하는 데이터였기에, 데이터에 대한 신뢰도 자체가 크게 높다고 보기는 어렵다.

- 현재의 분류 모델의 경우 2014~2015 시즌 NBA 정규 시즌 경기에서 발생했던 모든 선수들의 슈팅에 대해 일괄적으로 여러 가지 feature에 대해 그 관계를 설명하고자 하였다. 그러나 실제 농구의 경우 각 선수는 가드/포워드/센터와 같이 할당받는 공격 역할과 그에 따른 주된 슈팅 위치가 존재하기 때문에 슈팅의 시도 횟수나 성공률 자체가 이에 bias되어 있을 가능성이 높다. 또한, 각 선수 별로 슈팅을 시도하는 방법이나 선호하는 슈팅 구역 등 슈팅 결과가 매우 상이할 것인데 이를 모든 선수들에 대해 일괄적으로 학습을 진행하였기 때문에 좋은 결과를 얻지 못했을 가능성이 있다. 그러나 이 데이터셋만을 가지고 각각의 선수에 대한 모델을 제작하기에는 각 선수에 대한 데이터는 매우 적다.
- ensemble 기법의 구현에서 오직 3개의 모델만을 고려하였기에 각 개별 모델에서의 성능에 비해 주목할만한 성능 향상이 일어나지 않은 것으로 보인다.

2. 추후 연구 개발 방향

이러한 결과 해석에 따라 추후 연구를 진행할 경우 다음과 같은 방향으로 연구를 개선할 수 있을 것으로 보인다.

- 각각의 슈팅에 대한 더 자세한 description과 context를 제공하는 신뢰 가능한 데이터를 활용하는 것이 좋을 것이다. 단순한 슈팅 거리 뿐만 아니라 정확한 슈팅 위치, 슛을 진행하는 상황에서 두 팀의 점수 차이 및 어떤 팀이 리드를 가지고 가고 있는지, 해당 선수의 최근 슛 성공률 및 경기 평점, 해당 선수와 함께 코트 위에서 뛰고 있는 선수들에 대한 정보 등 실시간으로 다양한 변수가 발생하고 예측이 어려운 해당 task에 대해서는 더 정확하고 자세한 데이터가 필요할 것으로 보인다.
- 충분한 양의 데이터를 얻을 수 있다면, 모든 선수 별 예측보다는 각 선수 별 슈팅 데이터를 모아서 선수 포지션, 선호 슈팅 구역 등의 선수 별 개인 데이터를 함께 활용해 해당 선수에 대한 슈팅 예측 모델을 제작하는 것이 더 합리적으로 보인다. 이와 같이 선수 개인 별 모델을 학습할 수 있다면 같은 팀에 속해 있는 선수들의 모델을 ensemble하여 팀의 기대 득점이나 득점 성공률을 창출해내는 등의 다양하고 유용한 variant도 가능할 것으로 보인다.
- computing resource의 한계로 진행하기 어려웠던 더 많은 layer를 가지고 있는 MLP를 활용하여, 다양한 parameter를 가지는 XGBoost 모델과 MLP 모델을 학습시켜서 15~20개 정도의 모델을 ensemble시킬 경우 유의미한 ensemble 결과 값을 얻어낼 가능성이 있을 것이다.