

24_04_30

☰ 태그

주제 소개 및 선정 배경

1. 주제 소개 및 선정 배경

주제 소개 @서윤 박

이번 보고서에서는 National Basketball Association (전미 농구 연합, 이하 NBA)을 주제로 삼았습니다. NBA는 세계적으로 가장 유명한 프로 농구 리그로, 많은 팬들과 높은 수준의 경기를 자랑합니다. 이 리그에서는 공격과 수비를 기본으로 하여 어시스트, 스틸, 필드골, 3점슛 등 다양한 변수가 경기 결과에 큰 영향을 미칩니다. 이러한 변수를 분석하는 것은 매우 흥미롭고 유의미한 작업이 될 것입니다.

선정 배경 @성준 복

NBA의 경기 결과는 여러 요인에 의해 달라지며, 이를 예측하는 것은 매우 복잡한 일입니다. 하지만 과거 데이터를 통해 다양한 경기 기록, 선수 스탯, 팀 성적 등을 고려하면 어느 정도 예측할 수 있습니다. 이는 통계적 분석과 머신러닝 기법을 활용하여 가능하며, 이러한 접근은 우승팀을 예측하는 좋은 사례가 될 수 있습니다.

데이터 활용의 필요성 @땡섭

이전 데이터는 팀 운영진과 코칭 스태프에게 매우 중요한 자산입니다. 데이터를 분석함으로써 다음과 같은 인사이트를 얻을 수 있습니다:

- 팀 전략 수립: 과거 데이터를 기반으로 효과적인 팀 전략을 세울 수 있습니다.
- 선수 기용: 특정 상황에서 어떤 선수를 기용하는 것이 유리한지에 대한 인사이트를 제공할 수 있습니다.
- 경기 준비: 상대 팀의 강점과 약점을 분석하여 경기에 대한 준비를 보다 철저히 할 수 있습니다.

데이터 분석의 이점 @창원 문

데이터 분석을 통해 얻게 되는 인사이트는 다음과 같은 여러 측면에서 큰 이점을 제공합니다:

- 경기 성적 향상: 데이터 기반의 전략 수립으로 팀의 경기 성적을 향상시킬 수 있습니다.

- 선수 관리: 선수의 체력과 경기력을 최적화하여 지속적인 성장을 도모할 수 있습니다.
- 팬과의 소통: 데이터 분석 결과를 활용하여 팬들과의 소통을 증진하고, 팬들에게 더 많은 정보를 제공할 수 있습니다.

결론 @윤나 주

따라서 NBA 데이터를 활용한 분석은 팀의 성적을 향상시키고, 팬들의 관심을 증대시키는 데 큰 도움이 될 것입니다. 이번 보고서에서는 NBA의 다양한 경기 기록과 선수 스탯을 분석하여 팀 성적을 예측하고, 이를 바탕으로 효과적인 팀 운영 전략을 제시하고자 합니다. 이를 통해 팀 운영진과 코칭 스태프에게 유용한 인사이트를 제공하며, 데이터 기반의 의사결정을 지원할 수 있을 것입니다.

24_05_14

태그

데이터 소개

윤나 주 5월 16일 (편집됨)

이상치 리바운드 : 임계치를 얼마로 해야하는지 예)75% - 사분위수

윤나 주 5월 16일

팀단위는 스케일링 안해두됨 일단 확인해보시길

땡 땡 5월 17일 (편집됨)

창원이형이랑 서윤이 범주형 변수 coldel_split.ipynb의 인코딩 코드 참고해주세요
인코딩하면서 컬럼명 바꾸고 기존 것 삭제했음

땡 땡 5월 17일

각각 team_encoded랑 date_encoded,team_opp_encoded 확인 부탁드립니다

성준 복 5월 17일 (편집됨)

명섭이 column 은 전반적으로 농구의 클래식 스탯으로서 중요한 역할을 하니 이상치 제거같은거 할필요 없이 표준화정도 해서 쓰면 됩니다.(아마 굉장히 높은 확률로 정규분포를 띄게
갈기는 한데) 그리고 데이터가 팀 전체에 관한 거라 outlier가 발생할 가능성이 극히 드뭅니다.
(경기 전체에서 팀 전체가 기록한 것에 대한 값을 나타내기 때문에.)

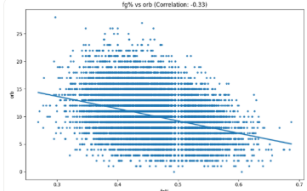
1

땡 땡 5월 17일

각각 고려해둔 스케일러를 기반으로, 신속하게 150개 교차 검증을 거칠지 묻고 싶습니다. 추가로, train test 분리 외에 검증용 validation dataset 구축 건의드립니다. 이번에 team 컬럼을 인코딩하며 데이터 누수 최소화를 위해 k-fold를 사용해봤지만, 이는 검증 시에도 사용하는 기법으로 알고 있습니다.

창원 문 5월 19일

나의 예측: orb(공격 리바운드)가 높으면 높을 수록 fg%(필드 골 성공률)이 높을것 이라 판단→
but, Correlation: -0.33으로 음의 상관관계를 가지는 것으로 나타남.



데이터 출처 @성준 복

이번 분석에 사용된 데이터는 [Basketball Reference](#) 웹사이트에서 수집되었습니다. Basketball Reference는 NBA와 관련된 다양한 통계 데이터를 제공하는 웹사이트로, 본 연구에서는 웹 크롤링을 통해 2015년부터 2022년까지의 데이터를 수집하였습니다. 수집된 데이터는 각 시즌의 다양한 경기 기록을 포함하고 있습니다.

- 웹사이트: Basketball Reference

데이터 구성 @윤나 주

수집된 데이터는 총 150개의 칼럼과 17,772개의 행으로 구성되어 있습니다. 각 행은 하나의 경기 기록을 나타내며, 각 칼럼은 해당 경기의 다양한 통계치를 포함하고 있습니다.

데이터 샘플

Unnamed: 0	mp	mp.1	fg	fga	fg%	3p	3pa
0	0	24.0	39.0	81.0	0.481	20.0	30.0
1	1	24.0	36.0	100.0	0.360	7.0	31.0
2	2	24.0	37.0	85.0	0.435	9.0	25.0

3	3	24.0	27.0	84.0	0.321	3.0	21.0
4	4	24.0	27.0	86.0	0.314	6.0	23.1

데이터 칼럼 소개 @창원 문 @성준 복 @명섭 @서윤 박 @윤나 주

수집된 데이터는 총 150개의 칼럼으로 구성되어 있으며, 각 칼럼은 경기의 다양한 통계치를 포함하고 있습니다. 아래는 주요 칼럼에 대한 설명입니다:

Column	데이터 설명	데이터 유형	(확)스케일링
mp	5 x 인당 플레이한 시간(분)		
fg	(팀)성공한 필드골		
fga	(팀)시도한 필드골		
fg%	(팀)필드골 성공률		
3p	(팀)성공한 3점슛		
3pa	(팀)시도한 3점슛		
3p%	(팀)3점슛 성공률		
ft	(팀)성공한 자유투		
fta	(팀)시도한 자유투		
ft%	(팀)자유투 성공률		
orb	(팀)공격 리바운드		
drb	(팀)수비 리바운드		
trb	(팀)총 리바운드		
ast	(팀)어시스트		
stl	(팀)스틸		
blk	(팀)블록		
tov	(팀)턴오버		
pf	(팀)개인 파울		
pts	(팀)득점		
ts%	(팀)야투율에 3점 슛과 자유투를 보정하여 만든 기록 유효슈팅 성공률	$TS\% = (\text{총 득점}) * 50 / \{ \text{야투시도} + (0.44 * \text{자유투시도}) \}$	
efg%	(팀)유효 필드골 성공률	$eFG = (\text{야투} + 0.5X\text{경기당 3점 슛}) / \text{야투시도}$	
3par	(팀)3점슛 시도 비율		
ftr	(팀)자유투 비율		
orb%	(팀)공격 리바운드 비율	내 공격리바운드 / 내 공격리바운드 + 상대 수비리바운드	
drb%	(팀)수비 리바운드 비율	내 수비리바운드 / 내 수비리바운드 + 상대 공격리바운드	
trb%	(팀)총 리바운드 비율	홈팀 리바운드 / 홈팀 리바운드 + 상대 팀 리바운드	
ast%	(팀)어시스트 비율	어시스트 비율 (ast%): 어시스트 비율은 플레이어가 자신의 팀원에게 어시스트를 주는 빈도를 나타냅니다. 이는 해당 플레이어가 자신의 팀원에게 어시스트를 통해 득점 기회를 제공하는 정도를 나타내는 지표	
stl%	(팀)스틸 비율	스틸 비율 (stl%): 스틸 비율은 플레이어가 상대편 플레이어로부터 스틸을 성공하는 비율을 나타냅니다. 이는 해당 플레이어가 수비에서 상대의 패스를 가로채거나 볼을 빼앗는 빈도를 측정합니다.	
blk%	(팀)블록 비율	블록 비율은 플레이어가 상대의 슈팅을 블로킹하는 비율을 나타냅니다. 이	

		는 해당 플레이어가 수비에서 상대의 슛을 방어하거나 차단하는 정도를 측 정합니다.		
tov%	(팀)턴오버 비율	턴오버 비율은 플레이어가 공격 중에 턴오버를 하는 비율을 나타냅니다. 이 는 해당 플레이어가 공을 잃는 빈도를 측정합니다.		
usg%	포제션을 마무리하는 슛을 던지는 비 율을 나타낸다	UsG = [{야투시도+(자유투시도 x0.44)+(어시스트x0.33)+턴오 버}x40x리그포제션]/(출장시간x팀포 제션)		
ortg	(팀)오펜시브 레이팅. 100번의 공격 기회에서 득점 기대치.			
drtg	(팀)디펜시브 레이팅. 100번의 수비 기회에서 실점 기대치.			
fg_max	(한 선수)가최대 성공한 필드골			
fga_max	(한 선수)가최대 시도한 필드골			
fg%_max	(한 선수)가최대 필드골 성공률			
3p_max	(한 선수)가최대 성공한 3점슛			
3pa_max	(한 선수)가최대 시도한 3점슛			
3p%_max	(한 선수)가최대 3점슛 성공률			
ft_max	(한 선수)가최대 성공한 자유투			
fta_max	(한 선수)가최대 시도한 자유투			
ft%_max	(한 선수)가최대 자유투 성공률			
orb_max	(한 선수)가최대 공격 리바운드			
drb_max	(한 선수)가최대 수비 리바운드			
trb_max	(한 선수)가최대 총 리바운드			
ast_max	(한 선수)가최대 어시스트			
stl_max	(한 선수)가최대 스틸			
blk_max	(한 선수)가최대 블록			
tov_max	(한 선수)가최대 턴오버			
pf_max	(한 선수)가최대 개인 파울			
pts_max	(한 선수)가최대 득점			
+/-_max	(한 선수의)최대 플러스-마이너스 통 계	한선수가 경기되면서 얻은 마진값	Maximum plus-minus statistic	
ts%_max	(한 선수)가야투율에 3점 슛과 자유 투를 보정하여 만든 기록 유효슈팅 성공률			
efg%_max	최대 유효 필드골 성공률			
3par_max	최대 3점슛 시도 비율			
frt_max	(한 선수의)최대 자유투 개수			
orb%_max	(한 선수의)최대 공격 리바운드 비율			
drb%_max	(한 선수의)최대 수비 리바운드 비율			
trb%_max	(한 선수의)최대 총 리바운드 비율	한 선수의 리바운드 / 선수가 얻은 리 바운드 기회		
ast%_max	최대 어시스트 비율			
stl%_max	(한 선수의)최대 스틸 비율			
blk%_max	(한 선수의)최대 블록 비율			
tov%_max	(한 선수의)최대 턴오버 비율			
usg%_max	포제션을 마무리하는 슛을 던지는 비 율을 나타낸다			
ortg_max	한 선수의 오펜시브 레이팅			

drtg_max	한 선수의 디펜시브 레이팅			
team	팀 이름			
total	총 득점			
home	홈 경기 여부 (1이면 홈, 0이면 원정)			
season	시즌 연도			
date	경기 날짜			
won	팀이 경기에서 승리했는지 여부 (TRUE는 승리, FALSE는 패배)			

Column	142개	스케일링 기법	
mp	명	robust	
fg		standard	
fga		standard	
fg%		standard	
3p		standard	
3pa		standard	
3p%		standard	
ft		standard	
fta		standard	
ft%		standard	
orb		standard	
drb		standard	
trb		standard	
ast		standard	
stl		standard	
blk		standard	
tov		standard	
pf		standard	
pts		standard	
ts%		standard	
efg%		standard	
3par		standard	
ftr		standard	
orb%		standard	
drb%		standard	
trb%		standard	
ast%		standard	
stl%		standard	
blk%	성	standard	
tov%		standard	
usg%	column 제거 건의드립니다. 포제션을 마무리하는 슛을 던지는 비율을 나타낸다 인데 당연히 팀에 어느선수든 공격만 하면 usg%가 100이기때문에 모든 값이 100입니다.		
ortg		standard	
drtg		standard	
fg_max		1. 이상치 처리 후 Robust 2. 그냥 Robust	

fga_max		Normalize	
fg%_max		Normalize	
3p_max		Normalize	
3pa_max		Normalize	
3p%_max		Normalize	
ft_max		Normalize	
fta_max		Normalize	
ft%_max		Normalize	
orb_max		Normalize	
drb_max		Normalize	
trb_max		Normalize	
ast_max		Normalize	
stl_max		Normalize	
blk_max		Normalize	
tov_max		Normalize	
pf_max		Normalize	
pts_max		Normalize	
+/-_max		standard	
ts%_max		Normalize	
efg%_max		Normalize	
3par_max		Normalize	
ftr_max		Normalize	
orb%_max	창	스케일 기법 작성	
drb%_max		이상치 제거후 min-max scaler 적용	
trb%_max		이상치 제거후 min-max scaler 적용	
ast%_max		이상치 제거후 min-max scaler 적용	
stl%_max		이상치 제거후 min-max scaler 적용	
blk%_max		이상치 제거후 min-max scaler 적용	
tov%_max		이상치 제거후 min-max scaler 적용	
usg%_max		이상치 제거후 min-max scaler 적용	
ortg_max		이상치 제거후 min-max scaler 적용	
drtg_max		이상치 제거후 min-max scaler 적용	
team	target_encoding		
total		?	
home		?	
mp_opp		명섭이 컬럼이랑 동일 컬럼	
fg_opp		standard	
fga_opp		standard	
fg%_opp		standard	
3p_opp		standard	
3pa_opp		standard	

3p%_opp		standard	
ft_opp		standard	
fta_opp		standard	
ft%_opp		standard	
orb_opp		standard	
drb_opp		standard	
trb_opp		standard	
ast_opp		standard	
stl_opp		standard	
blk_opp	윤나	standard	
tov_opp		standard	
pf_opp		standard	
pts_opp		standard	
ts%_opp	비율 데이터 0~1 사이로정리	standard	
efg%_opp	비율 데이터 0~1 사이로정리	standard	
3par_opp	비율 데이터 0~1 사이로정리	standard	
ftr_opp	비율 데이터 0~1 사이로정리	standard	
orb%_opp	이상치가 많음	standard	
drb%_opp	비율 데이터 0~1 사이로정리	standard	
trb%_opp	비율 데이터 0~1 사이로정리	standard	
ast%_opp	비율 데이터 0~1 사이로정리	standard	
stl%_opp	비율 데이터 0~1 사이로정리	standard	
blk%_opp	비율 데이터 0~1 사이로정리	standard	
tov%_opp	비율 데이터 0~1 사이로정리	standard	
usg%_opp	팀단위면 의미가 없음. 공격을 얼마나 마무리했나의 의미기 때문에 개인이면 의미 있음.		
ortg_opp		standard	
drtg_opp		standard	
mp_max_opp.1			
fg_max_opp	이상치가 많음	RobustScaler	
fga_max_opp	이상치가 많음	RobustScaler	
fg%_max_opp	이상치가 많음	RobustScaler	
3p_max_opp	이상치가 많음	RobustScaler	
3pa_max_opp	이상치가 많음	RobustScaler	
3p%_max_opp	이상치가 많음	RobustScaler	
ft_max_opp	이상치가 많음	RobustScaler	
fta_max_opp	이상치가 많음	RobustScaler	
ft%_max_opp	큰 왜곡을 보임	Log Transformation 또는 Box-Cox	
orb_max_opp	서	스케일 기법 작성	
drb_max_opp		Robust	
trb_max_opp		Robust	
ast_max_opp		Robust	
stl_max_opp		Robust	
blk_max_opp		Robust	
tov_max_opp		Robust	
pf_max_opp		Robust	
pts_max_opp		Robust	

+/_max_opp		standard	
ts%_max_opp		이상치제거 minmax	
efg%_max_opp		이상치제거 minmax	
3par_max_opp		이상치제거 minmax	
ftr_max_opp		이상치제거 minmax	
orb%_max_opp		이상치제거 minmax	
drb%_max_opp		이상치제거 minmax	
trb%_max_opp		이상치제거 minmax	
ast%_max_opp		이상치제거 minmax	
stl%_max_opp		이상치제거 minmax	
blk%_max_opp		이상치제거 minmax	
tov%_max_opp		이상치제거 minmax	
usg%_max_opp		이상치제거 minmax	
ortg_max_opp		이상치제거 minmax	
drtg_max_opp		이상치제거 minmax	
team_opp	target_encoding		
total_opp			
home_opp			
season		label-encoding	
date	명섭 - label encoding		
won	astype(int) 변환		
	standard	robust	

Standard Scaler : 팀 단위에 사용

[]

Robust Scaler : 이상치 존재 유무 / 개인 단위 데이터

['orb%_opp', 'fg_max_opp', 'fga_max_opp', 'fg%_max_opp', '3p_max_opp', '3pa_max_opp', '3p%_max_opp', 'ft_max_opp', 'fta_max_opp']

MinMax Scaler : / 개인 단위 데이터

['ts%_opp', 'efg%_opp', '3par_opp', 'ftr_opp', 'usg%_opp', 'drb%_opp', 'trb%_opp', 'ast%_opp', 'stl%_opp', 'blk%_opp', 'tov%_opp']

Log Transformation OR Box-Cox : 큰 왜곡

['ft%_max_opp']

결론

본 보고서에서는 Basketball Reference에서 수집한 NBA 데이터를 활용하여 다양한 통계 분석을 수행하고자 합니다. 이를 통해 팀 성적을 예측하고, 선수 기용 전략을 수립하며, 데이터 기반의 의사결정을 지원할 수 있을 것입니다. 수집된 데이터는 총 150개의 칼럼과 17,772개의 행으로 구성되어 있으며, 각 칼럼은 경기의 다양한 통계치를 포함하고 있습니다.

24_05_23

≡ 태그

데이터 전처리

1. 결측 및 중복 칼럼 제거 @윤나 주

제거된 칼럼 (6개):

- `+/-`
- `mp_max`
- `mp_max.1`
- `+/-_opp`
- `mp_max_opp`
- `mp_max_opp.1`

이 단계에서는 데이터셋에 존재하는 결측 값과 중복된 칼럼을 제거하였습니다. `+/-`, `mp_max`, `mp_max.1`, `+/-_opp`, `mp_max_opp`, `mp_max_opp.1` 칼럼들은 모두 동일하거나 매우 유사한 정보를 담고 있어 분석에 불필요하다고 판단되었습니다. 이러한 중복된 칼럼을 제거함으로써 데이터의 일관성을 확보하고 분석의 복잡성을 줄였습니다.

2. 중복 칼럼 제거 @땡섭

제거된 칼럼 (4개):

- `mp_opp.1`: 단순 수치만 중복
- `mp.1`: 단순 수치만 중복
- `index_opp`: `home_opp` 와 중복 정보 포함
- `pts`: `total` 칼럼의 내용과 일치하기 때문에 삭제

`mp_opp.1` 과 `mp.1` 칼럼은 단순히 기존 칼럼의 값을 반복하고 있어 데이터에 추가적인 정보를 제공하지 않습니다. `index_opp` 칼럼은 `home_opp` 칼럼과 중복되는 정보를 포함하고 있으며, `pts` 칼럼은 `total` 칼럼과 같은 내용을 담고 있기 때문에 이들을 제거하여 데이터의 중복성을 줄였습니다. 이러한 제거 작업을 통해 데이터셋이 보다 간결해지고 효율적인 분석이 가능해졌습니다.

3. 변동성 부족 칼럼 제거 @성준 복 @윤나 주 @땡섭

제거된 칼럼 (2개):

- usg%
- usg%_opp

usg% 와 usg%_opp 칼럼들은 모든 값이 100%로 동일하여 변동성이 전혀 없는 상태였습니다. 모든 값이 동일한 칼럼은 분산이 0이기 때문에, 다중공선성을 나타내는 VIF (Variance Inflation Factor) 값이 NaN으로 계산됩니다. 이런 칼럼들은 분석에 있어서 유의미한 정보를 제공하지 않으므로, 데이터를 정제하고 신뢰성을 높이기 위해 제거했습니다.

4. 이상치 제거

제거된 칼럼 (2개): @윤나 주

- ft%_max
- ft%_max_opp

ft%_max 와 ft%_max_opp 칼럼들은 전체 플레이어 데이터에서 개인 선수의 특정 수치가 지나치게 큰 값을 가지는 경우로, 이러한 값들은 분석 결과에 왜곡을 일으킬 수 있습니다. 특히, 데이터의 중심 경향을 파악하는 데 있어서 큰 영향을 미치지 않는 비정상적으로 큰 값들은 제거함으로써 데이터의 신뢰성을 높일 수 있습니다.

5. NaN VIF 값 변수 제거: 데이터 신뢰성 향상 @땡섭

NaN 값을 가지는 변수들은 다중공선성 문제를 나타내며, 이는 데이터의 이상을 의미합니다. 다중공선성은 모델의 안정성을 저해하고, 결과의 해석을 어렵게 만들 수 있기 때문에, 이러한 변수를 제거하는 것이 중요합니다.

NaN VIF 값을 가진 칼럼:

- ft%_max
- ft%_max_opp
- usg%
- usg%_opp

이 칼럼들은 VIF 값이 NaN으로 계산되어 다중공선성 문제를 가지고 있음을 나타냈습니다. 이러한 변수들을 제거함으로써 분석의 정확성과 신뢰성을 높일 수 있습니다.

결론

이번 데이터 전처리 과정을 통해 총 150개의 칼럼 중 14개의 칼럼이 제거되었습니다. 최종적으로 136개의 칼럼이 남았으며, 이를 통해 데이터셋의 일관성을 확보하고 분석의 효율성을 크게 향상시킬 수 있었습니다. 중복된 칼럼, 변동성이 부족한 칼럼, 이상치 및 NaN VIF 값을 가지는 칼럼들을 제거함으로써 데이터의 신뢰성을 높이고, 보다 정확하고 유의미한 분석 결과를 도출할 수 있게 되었습니다.

피쳐스케일링

Feature Engineering 과정은 모델의 성능을 향상시키기 위해 데이터를 변환하고 특징을 추출하는 중요한 단계입니다. 이번 분석에서는 팀 단위 데이터와 개인 선수 단위 데이터를 각각 다르게 스케일링하여 보다 정밀한 분석을 수행했습니다.

피쳐 스케일링 @윤나 주 @창원 문 @성준 복 @서윤 박

1. [Team] 단위 데이터

- **대상 칼럼:** 팀 관련 칼럼
- **설명:** 팀 단위의 데이터는 개별 선수의 스탯이 누적된 값으로, 중간경향성이 잘 보였고 정규분포 평향성이 강한 특징이 있었습니다.
- **사용한 스케일러:** Standard Scaler
 - **적용 이유:** Standard Scaler는 데이터를 평균이 0, 분산이 1이 되도록 변환합니다. 이는 정규분포를 따르는 데이터에 적합합니다.

2. [Opponent] 단위 데이터

- **대상 칼럼:** 상대 팀 관련 칼럼 (칼럼 명 + `_opp`)
- **설명:** 상대 팀의 단위 데이터 역시 팀 단위 데이터와 동일한 특징을 가집니다.
- **사용한 스케일러:** Standard Scaler
 - **적용 이유:** 상대 팀 데이터도 팀 단위 데이터와 동일한 방식으로 스케일링하여 일관성을 유지합니다.

3. [Team] 개인 선수 단위 데이터

- **대상 칼럼:** 개인 선수 단위 데이터 (칼럼 명 + `_max`)
- **설명:** 각 팀 내 개별 선수의 최대 값으로 나타나는 스탯입니다. 이 데이터는 정규분포를 따르지 않을 수 있습니다.
- **사용한 스케일러:** MinMax Scaler 및 Robust Scaler

- **비율 데이터:** 사분위수(3배수 범위 지정) 후 MinMax Scaler 사용
 - **적용 이유:** MinMax Scaler는 데이터의 최대값과 최소값을 기준으로 0과 1 사이의 값으로 변환합니다. 비율 데이터는 특정 범위 내에 분포하기 때문에 이 스케일러가 적합합니다.
- **비율 외 데이터:** 사분위수(3배수 범위 지정) 후 Robust Scaler 사용
 - **적용 이유:** Robust Scaler는 중앙값과 IQR(Interquartile Range, 사분위 범위)을 사용하여 데이터를 스케일링합니다. 이는 이상치(outliers)의 영향을 줄일 수 있습니다.

4. [Opponent] 개인 선수 단위 데이터

- **대상 컬럼:** 상대 팀의 개인 선수 단위 데이터 (컬럼 명 + `_max_opp`)
- **설명:** 상대 팀의 개별 선수 스탯 역시 팀의 주요 선수들의 최대 값으로 나타납니다.
- **사용한 스케일러:** MinMax Scaler 및 Robust Scaler
 - **적용 이유:** 상대 팀의 개인 선수 단위 데이터도 동일한 스케일링 방법을 사용하여 분석의 일관성을 유지합니다.

1. 데이터 분할 (Split Train, Test Data) @땡섭 @서윤 박

설명:

- **훈련 세트 (train set):** 2016~2022년 데이터 (총 17,773개, 137개 피쳐)
- **테스트 세트 (test set):** 2023~2024년 데이터 (총 2,309개, 137개 피쳐)
- **비율:** 8:2로 나누어 모델의 일반화 성능을 높이고 최적의 모델을 선택하기 위해 train, validation, test 세트로 세분화.

세부사항:

- **train_X:** feature_columns
- **train_Y:** 'won' 컬럼

참고사항:

- **검증 세트 (validation set):** 모델의 일반화 능력을 높이기 위해 학습 중 평가에 사용되는 데이터.
- **교차 검증 (Cross Validation):** 과적합 방지 목적으로 사용.

2. 이상치 처리 및 추가 인코딩 (Outlier Processing + Additional Encoding) @서윤 박 @땡섭

설명:

- 이상치 처리 후 스케일링 및 추가 인코딩 작업.
- 스케일러 종류: Minmax Scaler, Robust Scaler, Standard Scaler
- 각 스케일러 별로 train, test 데이터 세트를 3쌍씩 설정하여 적용.

질문 및 답변:

- Q: 이상치 처리와 스케일러 사이의 관계?

A: 이상치 처리는 각 컬럼의 샘플 분포에 기반하며, 스케일러도 마찬가지. 따라서, 이상치 처리 후 데이터 프레임을 병합.

인코딩 대안:

1. 날짜 컬럼: 날짜 간 차이를 일수로 변환.
2. 팀 관련 컬럼: 타겟 인코딩 사용.

3. 시계열 데이터 분할 (TimeSplitSeries) @윤나 주 @창원 문

설명:

- 데이터 누출 방지 및 시계열 특성 반영을 위한 분할 방법.

왜 TSS?

1. 데이터 누출 방지: 데이터의 중요성을 유지하면서 스케일링 후 분할.
2. 시계열 모순 방지: 미래 데이터를 이용한 과거 예측 방지.
3. 교차 검증 문제 방지: 검증 데이터 설정 및 교차 검증 시 문제 발생 방지.

4. 다중공선성 해결 (Multicollinearity Resolution) @땡섭 @성준 복 @서윤 박

(1) 상관관계 (Correlation)

설명:

- 상관관계 계수: 피쳐 간 상관관계를 분석하여 다중공선성 문제 해결.
- 피쳐 선택 기준:
 - 컬럼의 개수 ≥ 3 , 절대 상관관계 값 ≥ 0.9

- 일반 지식: $|\text{corr}| > 0.7$: Strong, $0.3 < |\text{corr}| < 0.7$: Moderate, $|\text{corr}| < 0.3$: Weak

적용 방법:

- **기준 1-1:** 상관관계 계수 기반으로 피쳐 선택.
- **기준 1-2:** 상관관계 매트릭스를 통해 피쳐 선택.

(2) 분산 팽창 요인 (Variance Inflation Factor, VIF)

설명:

- **VIF 값**을 활용한 다중공선성 문제 해결.
- **피쳐-VIF 데이터 프레임:** 각 인스턴스는 (피쳐, VIF 값)으로 구성.

일반 지식:

- $VIF \geq 10$ 인 컬럼 제거.
- 주의: 한 번에 모든 컬럼 제거 금지, 점진적 제거 필요.

적용 방법:

- **기준 2:** X와 X 간 VIF 비교를 통해 다중공선성 감소.

(3) RidgeClassifier

설명:

- **RidgeClassifier**를 이용한 ML 기반 피쳐 선택.
- **Corr & VIF:** 탐색적 데이터 분석 기반 피쳐 선택.

결론:

- 위의 피쳐 엔지니어링 방법들을 통해 데이터의 품질을 향상시키고, 모델의 예측 성능을 높이는 데 주력.
- 상관관계 및 VIF를 활용하여 다중공선성 문제를 해결하고, RidgeClassifier를 통해 최적의 피쳐를 선택.

24_06_06

☰ 태그

모델 선정 및 최적화

모델 선정 @성준 복 @윤나 주 @창원 문 @서윤 박 @땡섭

SVM, LGBM, GBM, Logistic, Decision Tree, Random Forest, XGBoost

모델 최적화

최적화 전

모델 성능 평가

평균 결과

=====:

Model: Logistic Regression

Average Accuracy: 0.6876

Average Precision: 0.6732

Average Recall: 0.7284

Average F1 Score: 0.6997

Model: Decision Tree

Average Accuracy: 0.5074

Average Precision: 0.1048

Average Recall: 0.1554

Average F1 Score: 0.1252

Model: Random Forest

Average Accuracy: 0.6006

Average Precision: 0.6044

Average Recall: 0.7898

Average F1 Score: 0.6597

Model: XGBoost
Average Accuracy: 0.6846
Average Precision: 0.7244
Average Recall: 0.5955
Average F1 Score: 0.6536

SVM
Accuracy: 0.50018
Precision: 0.09994
Recall: 0.20000
F1 Score: 0.13328

LGBM
Accuracy: 0.50220
Precision: 0.20100
Recall: 0.39312
F1 Score: 0.26596

GBM
Accuracy: 0.53872
Precision: 0.54210
Recall: 0.66216
F1 Score: 0.50952

1. 모델 최적화 방법

1.1. grid_search 방법을 이용함.

각 모델의 파라미터 설정

XGBoost, Random Forest : n_estimators, learning_rate, max_depth

Logistic : C, solver

SVM: C

GBM: learning_rate, max_depth, n_estimators

LGBM: learning_rate, max_depth, n_estimators

Decision Tree : criterion, n_estimators

2.2. 각 모델에 grid_search 진행 후 모델 성능 평가.

최적화 후

모델 성능 평가

```
평균 결과
=====
Model: Logistic Regression
Average Accuracy: 0.6823
Average Precision: 0.6659
Average Recall: 0.7307
Average F1 Score: 0.6968
-----

Model: Decision Tree
Average Accuracy: 0.6817
Average Precision: 0.6674
Average Recall: 0.7236
Average F1 Score: 0.6944
-----

Model: Random Forest
Average Accuracy: 0.6413
Average Precision: 0.7372
Average Recall: 0.5397
Average F1 Score: 0.5389
-----

Model: XGBoost
Average Accuracy: 0.6805
Average Precision: 0.6201
Average Recall: 0.9312
Average F1 Score: 0.7444
-----
```

```
LightGBM training accuracy may be bad since you didn't export
Model Accuracy Precision Recall F1 Score
0 SVM 0.732073 0.706636 0.792428 0.747077
1 LightGBM 0.646675 0.858974 0.349869 0.497217
2 Gradient Boosting 0.500652 0.000000 0.000000 0.000000
3 Logistic Regression 0.682300 0.665900 0.730700 0.696800
```

```

Model Accuracy Precision Recall F1 Score
0 SVM 0.683833 0.671551 0.718016 0.694006
1 LightGBM 0.642112 0.760192 0.413838 0.535926
2 Gradient Boosting 0.500652 0.000000 0.000000 0.000000
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification

```

```

Model Accuracy Precision Recall F1 Score
0 SVM 0.689700 0.666667 0.757180 0.709046
1 LightGBM 0.681226 0.663133 0.734987 0.697214
2 Gradient Boosting 0.501304 0.500328 0.994778 0.665793

```

```

Model Accuracy Precision Recall F1 Score
0 SVM 0.678618 0.664656 0.719321 0.690909
1 LightGBM 0.661669 0.739806 0.497389 0.594848
2 Gradient Boosting 0.499348 0.497006 0.216710 0.301818

```

```

_ensemble_ensemble, model_ensemble, model_ensemble, model_ensemble,
Model Accuracy Precision Recall F1 Score
0 SVM 0.458279 0.46172 0.511749 0.485449
1 LightGBM 0.500652 0.00000 0.000000 0.000000
2 Gradient Boosting 0.504563 0.62500 0.019582 0.037975

```

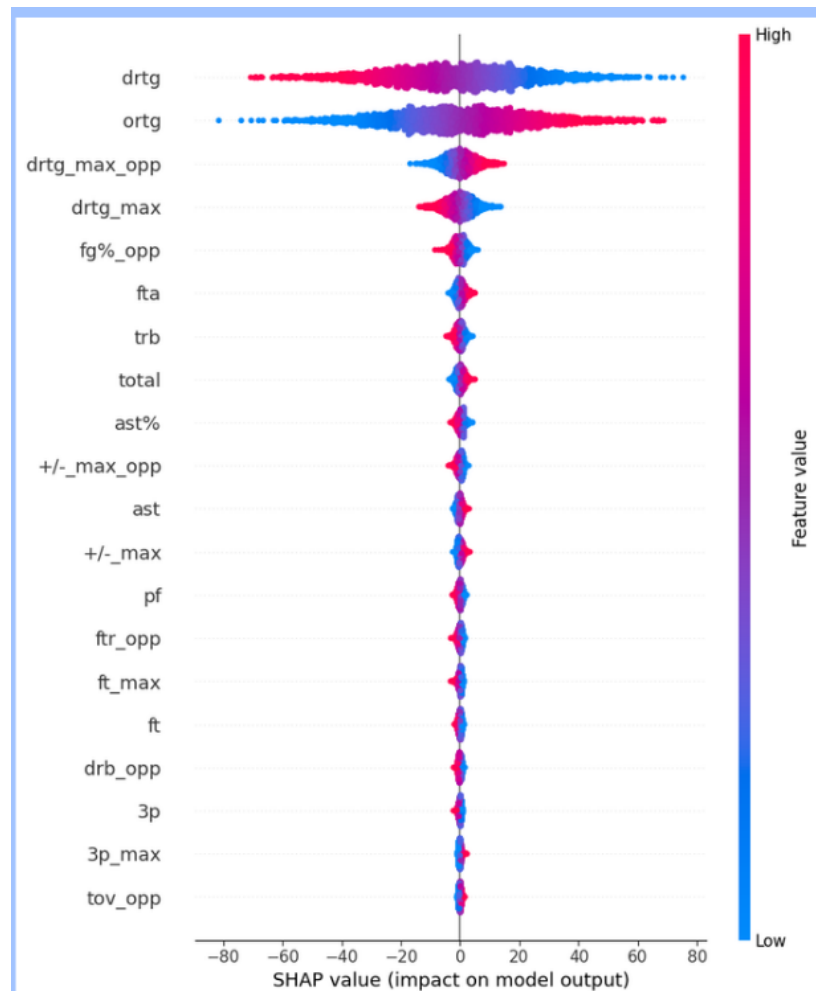
```

Model Accuracy Precision Recall F1 Score
0 SVM 0.679270 0.661557 0.732376 0.695167
1 LightGBM 0.500652 0.000000 0.000000 0.000000
2 Gradient Boosting 0.634941 0.601378 0.797650 0.685746

```

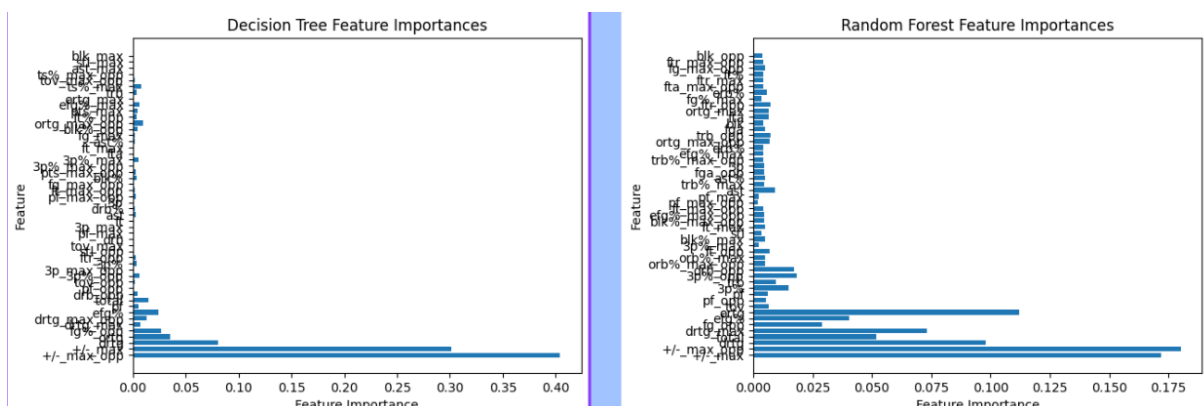
XAI (SHAP/LIME)

Logistic regression



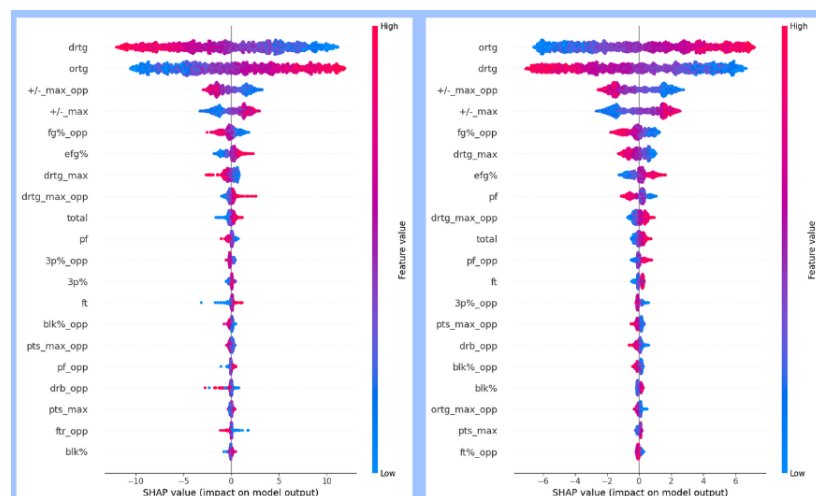
- 'drtg'와 'ortg' 피쳐는 SHAP 값이 크고 양수/음수 방향으로 널리 퍼져 있어서 모델 예측에 중요한 영향을 미치고 있음
- 반면, 'drtg_max_opp', 'drtg_max', 'fg%_opp', 'fta', 'trb' 등 나머지 피쳐는 SHAP 값이 0에 가까워 예측에 큰 영향을 미치지 않음을 나타냄

Descion tree & Random Forest



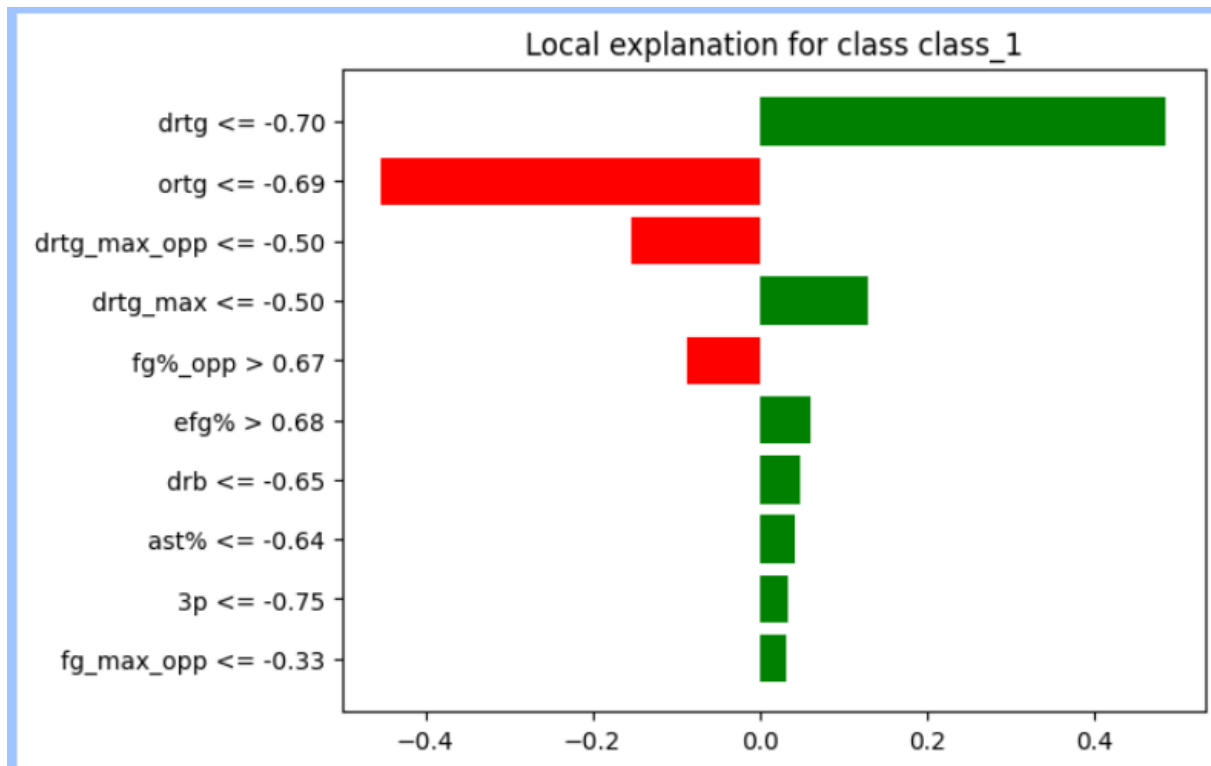
- 중요한 피처는 +/-_max_opp, +/-_max으로, 피처 중요도가 각각 0.4, 0.3 임을 알 수 있음.
- 나머지 피처들은 중요도가 매우 낮으며, 대부분 0에 가까운 값을 보임.
- 중요한 피처는 +/-_max_opp, +/-_max으로, 피처 중요도가 각각 0.175, 0.125 임을 알 수 있음.
- drtg와 ortg 피처들도 상대적으로 높은 중요도를 보임.
- 나머지 피처들은 중요도가 매우 낮으며, 대부분 0에 가까운 값을 보임.

Gradient Boosting & XGBoosting



- +/-_max_opp, +/-_max, drtg, ortg가 두 모델 모두에서 중요한 피처로 나타남
- +/-_max_opp와 +/-_max 피처는 예측에 매우 큰 영향을 미치며, 특히 XGBoost와 Gradient Boosting 모두에서 높은 중요도를 보임

SVM(LIME)



SVM은 고차원에서의 결정 경계로 예측을 수행하기 때문에 해석이 어려움 -> LIME을 사용하여 개별 예측

- drtg (Defensive Rating) ≤ -0.70 은 매우 긍정적인 영향을 미치는 것으로 나타남. 즉, 팀의 수비력이 강할수록 승률 예측에 도움이 된다고 볼 수 있음.
- ortg (Offensive Rating) ≤ -0.69 는 부정적인 영향을 미침. 공격력이 약한 팀일수록 승률이 낮다고 해석할 수 있음.
- drtg_max_opp ≤ -0.50 와 drtg_max ≤ -0.50 은 긍정적인 영향을 보이는데, 이는 상대 팀의 수비력과 자신의 수비력이 강할수록 승률이 높다는 것을 의미
- fg%_opp > 0.67 과 efg% > 0.68은 부정적인 영향을 미치는데, 이는 상대 팀의 슈팅 효율이 높을수록 승률이 낮다는 것을 나타냄