

I am Big fan of **Arsenal!!**

<18-19시즌에 아스널이 프리미어리그에서 우승하기 위해 필요한 것은?>

Contents

1. Do you know 사스널?
2. Premierleague에선 어떤 일이 일어나고 있을까?
3. 어떤 요소가 경기 승패에 큰 영향을 줄까?
4. 승무패 예측 모델을 만들어보자!
5. 내 모델은 과연 얼마나 뛰어날까?
6. 아스널은 어떤 요소를 중점적으로 준비해야 할까?

1. Do you know 사스널?



Do you know 사스널..?

4-4-3-4-3-4-3-4-4-3-2-5

<최근 12년간 아스널 성적>

우승은 못하지만 항상 4위 안에 들어 **사(4)스널** 이라는 별명을 얻게 됐습니다..



[아르센 벵거 감독]



[우나이 에메리 감독]

작년 시즌 5위까지 추락하자 22년간 아스널의 감독이었던 벵거를 경질 시키고

전 파리생제르망 감독인 에메리가 감독에 취임하며 변화를 모색하고 있는데요

1. Do you know 아스널?

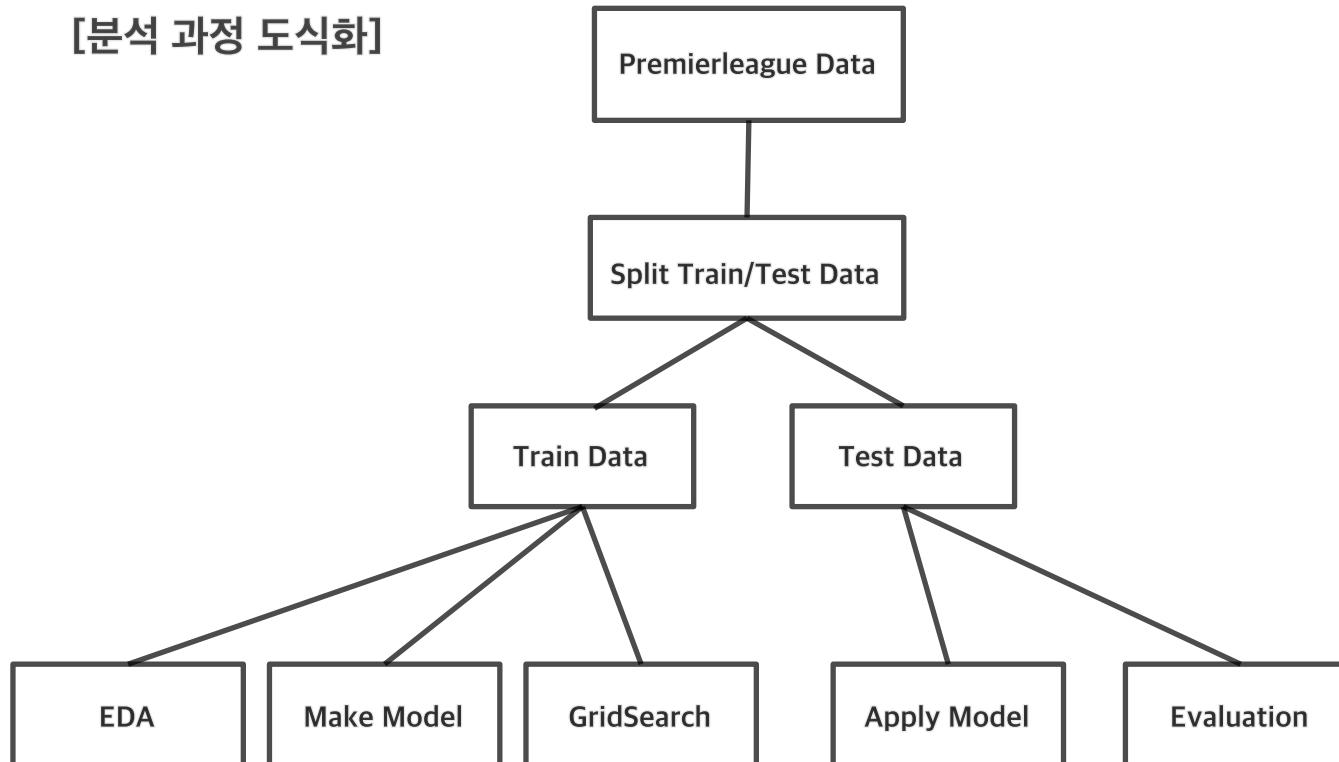


[무패우승을 경험한 03-04시즌 아스널]

다가오는 18-19시즌에 아스널이 좋은 성과를 거두려면 어떻게 해야 할까요?

프리미어리그 데이터를 통해 이를 알아보겠습니다!!

[분석 과정 도식화]



크롤링한 데이터를 Train Data와 Test Data로 나누고 Train Data에선 EDA 및 모델 생성, GridSearch를 통한 최적의 파라미터를 찾겠습니다. 그리고 Test데이터로 모델에 적용해보고 결과를 평가하겠습니다!

2. Premierleague에선 어떤 일이 일어나고 있을까?



Match Stats		
Huddersfield Town		Arsenal
45	Possession %	55
3	Shots on target	4
19	Shots	9
603	Touches	715
399	Passes	519
18	Tackles	11
19	Clearances	36
7	Corners	4
0	Offsides	2
1	Yellow cards	0
11	Fouls conceded	7

[www.premierleague.com 경기정보데이터]

[데이터 크롤링]

출처 : Premierleague 공식 사이트 ([Link](#))

수집 방법 : Selenium (Python 활용) ([Link](#))

수집 데이터 : 12-13시즌 - 17-18시즌 (6시즌)

38라운드 * 20팀 * 6년 = 4,560개 데이터

데이터 활용 : 12-13시즌 - 16-17시즌 (Train Data)

17-18시즌 (Test Data)

[데이터 소개]

Team : 팀 이름, 프리미어리그는 강등 제도가 있어 매년마다 3개의 팀이 바뀜

Possession : 점유율 (%)

SOT(Shots of Target) : 유효슈팅 (골문안으로 들어간 슛팅) (개)

Shots : 슛팅 (전체 슛팅) (개)

Touches : 공을 터치한 개수 (개)

Passes : 패스 (개)

Tackles : 태클 (개)

Clearances : 골문앞에서 공을 걷어낸 개수 (개)

Corners : 코너킥 (개)

Offsides : 오프사이드 (개)

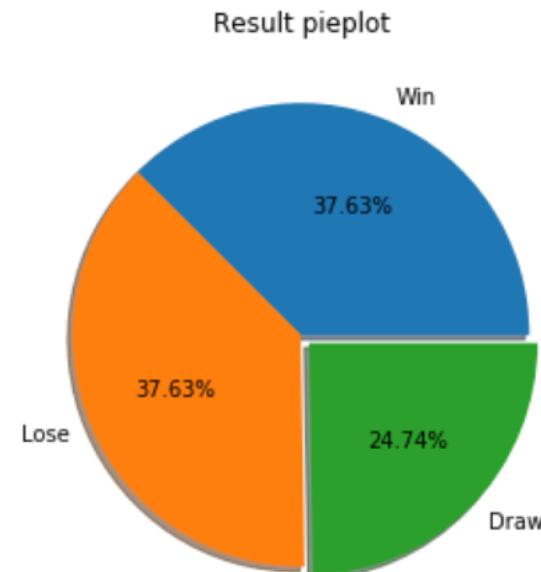
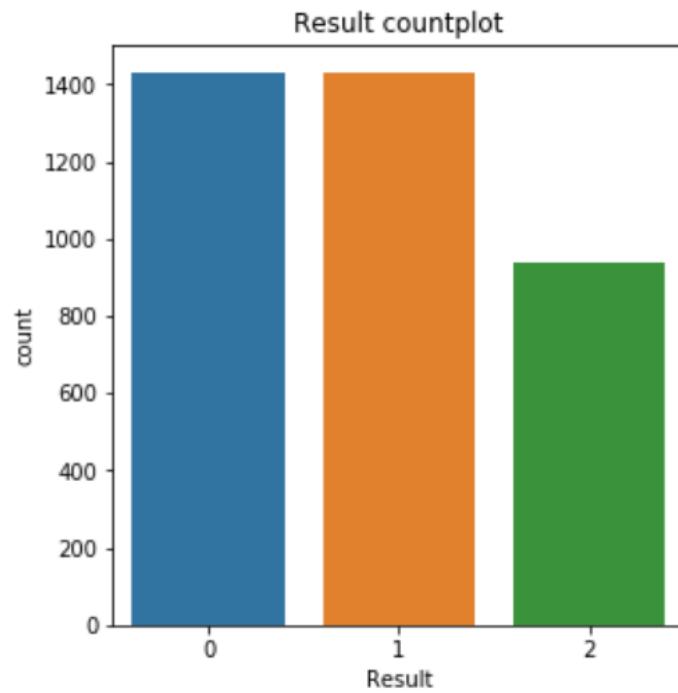
Goal : 골 (개)

Year : 연도 (2012, 2013, 2014, 2015, 2016, 2017)

Home : 홈 / 어웨이 여부 (어웨이 : 0, 홈 : 1)

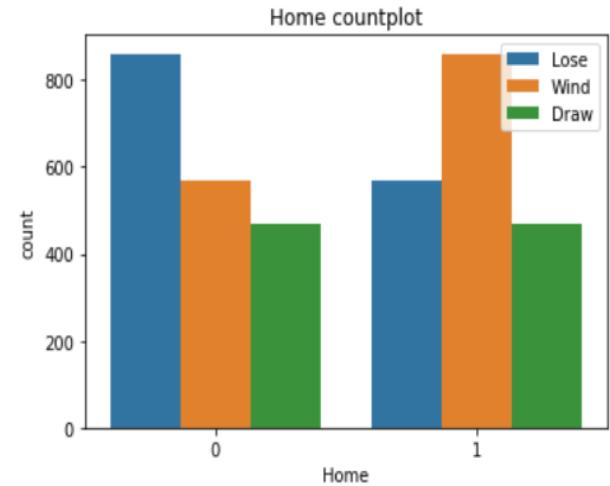
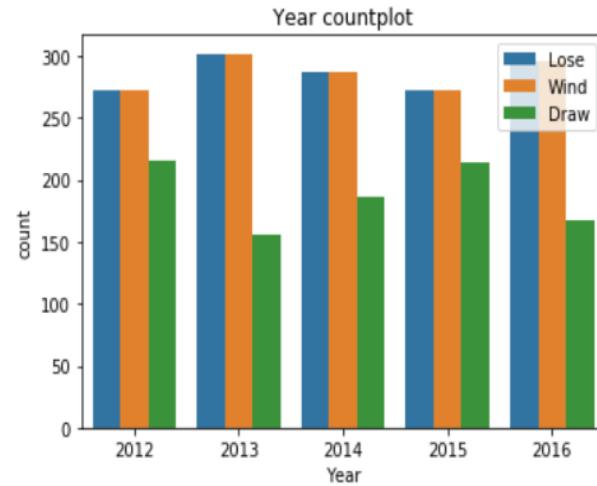
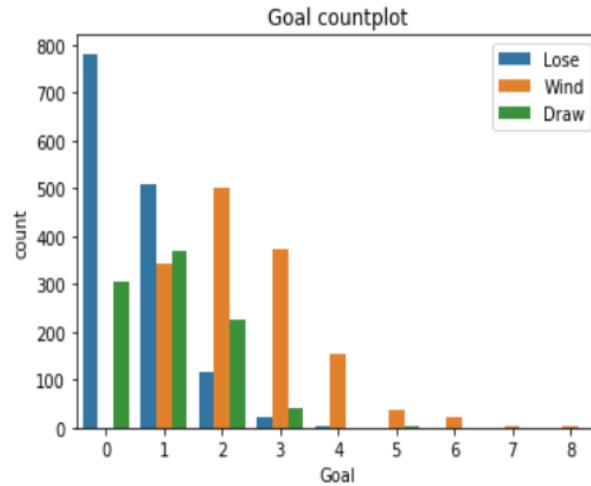
Result : 결과 (패 : 0, 승 : 1, 무 : 2)

[Y(결과)부터 살펴보겠습니다!]



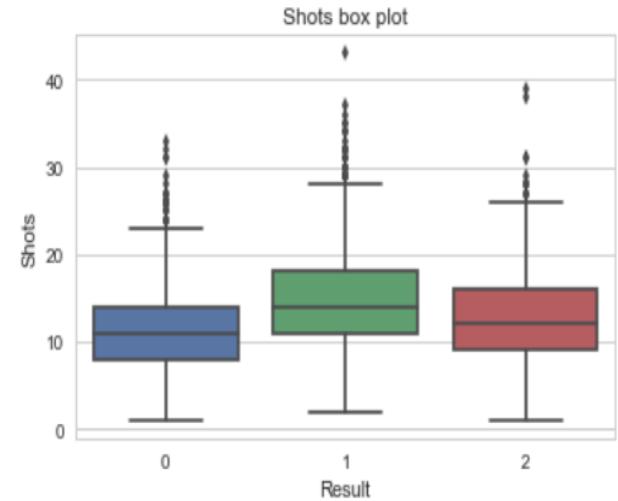
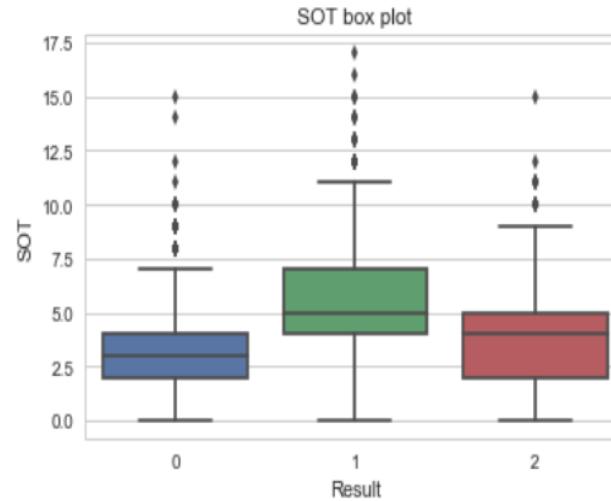
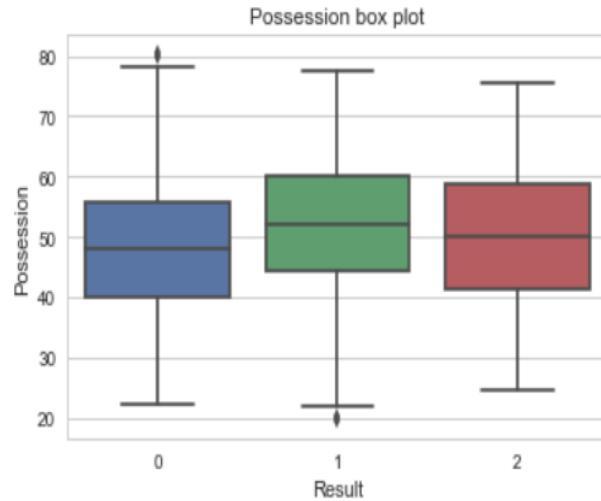
승 / 패 경기가 무 경기보다 상대적으로 많다는 것을 확인할 수 있습니다.

[Y와 카테고리형 변수의 관계를 살펴볼까요?]



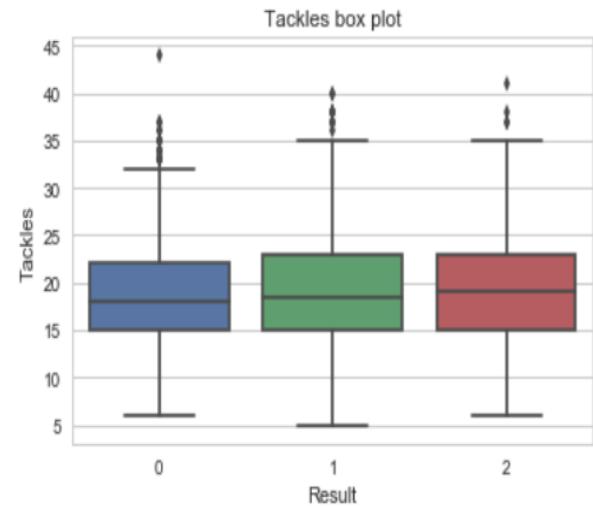
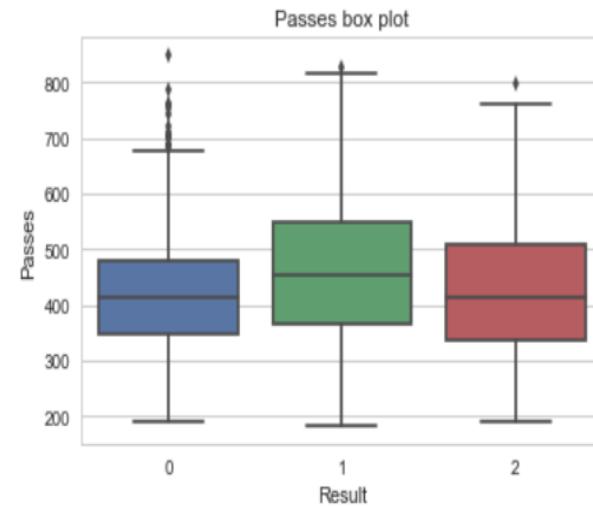
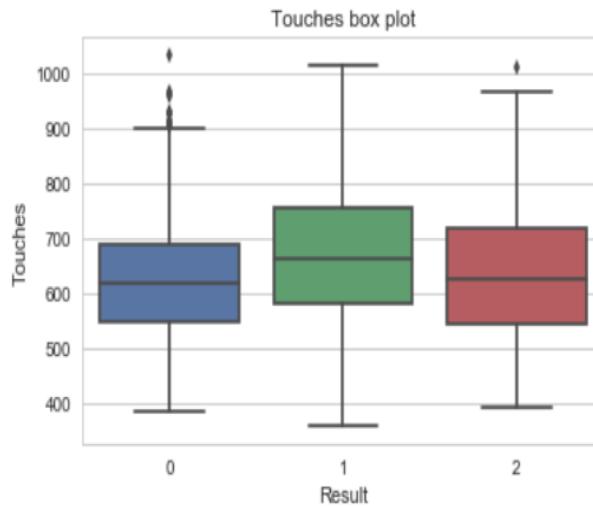
- 위의 그림은 카테고리 변수 별로 승/무/패에 관한 countplot입니다.
- Goal(골)이 적을수록 패배가, 많을수록 승리 횟수가 많다는 것을 확인할 수 있습니다. 분류하는데 큰 영향을 끼칠 겁니다.
- 매년마다 비교했을 때 Year(연도)는 크게 상관없어 보입니다. 일단 보류하겠습니다.
- Home(홈/어웨이)경우 확실히 홈경기(1)일 때 승리 횟수가 많다는 것을 확인할 수 있습니다. 분류하는데 좋은 변수라 판단됩니다.

[Y와 연속형 변수의 관계를 살펴보겠습니다 (1)]



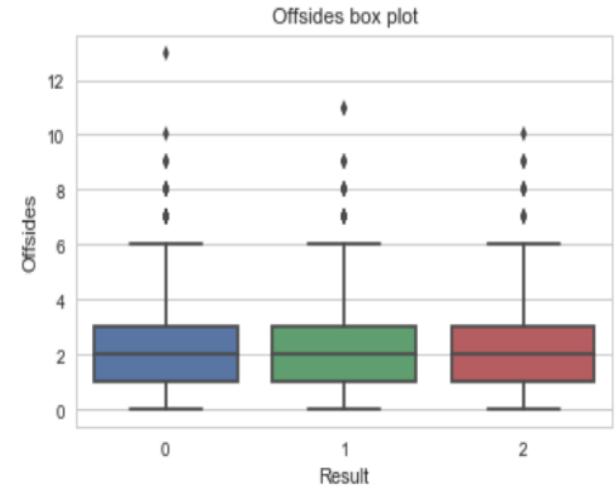
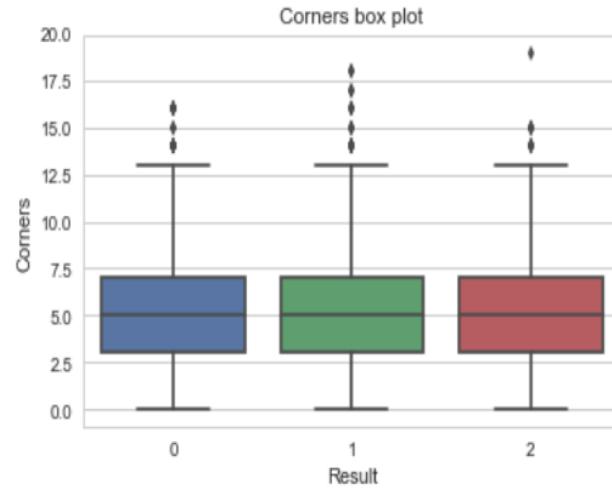
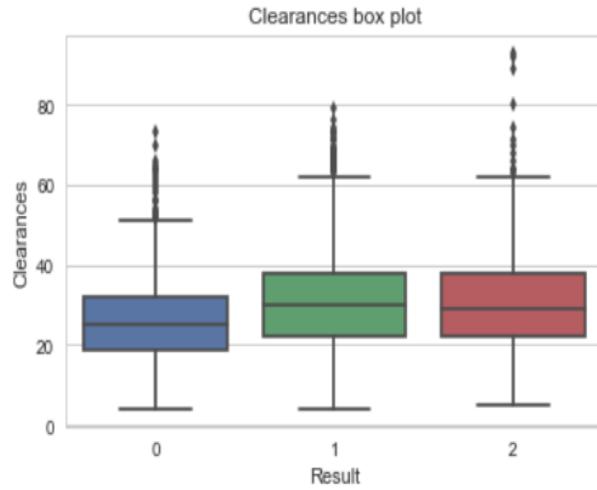
- 위의 그림은 연속형 변수 별로 승/무/패에 관한 Boxplot입니다.
- Possession(점유율)을 보면 각 결과 별 평균은 비슷해 보입니다. 그림으로 그룹별로 차이가 나는지는 확실히 모르겠습니다.
- SOT(유효슈팅)을 보면 각 결과 별 평균이 조금씩 차이가 나는 것을 확인할 수 있습니다. 분류에 도움이 될 것이라 생각합니다.
- Shot(슈팅)을 보면 각 결과 별 평균은 크게 차이가 없어 보입니다.

[Y와 연속형 변수의 관계를 살펴보겠습니다 (2)]



- Touches(볼터치)를 보면 승일 경우가 평균이 제일 높은 것을 확인할 수 있습니다. 패/무는 크게 차이가 나지 않습니다.
- Passes(패스)도 Touches와 비슷한 분포를 보입니다. 두 변수가 과연 분류에 도움이 될지는 모르겠습니다.
- Tackles(태클)은 세 집단이 비슷해 보입니다. 분류엔 크게 도움이 안될 것으로 판단됩니다.

[Y와 연속형 변수의 관계를 살펴보겠습니다 (3)]



- Clearances(걷어냄)은 승/무 의 경우는 비슷한 평균을 보이고 패를 하는 경우가 조금 낮은 모습을 보입니다.
- Corners(코너킥)을 보면 각 결과 별 평균은 비슷해 보입니다. 그림으로 그룹별로 차이가 나는지는 확실히 모르겠습니다.
- Offsides(오프사이드)도 각 결과 별 평균은 비슷해 보입니다. 그림으로 그룹별로 차이가 나는지는 확실히 모르겠습니다.

종합적으로 봤을 때, 단순히 그림으로 그룹간 비교하기엔 무리가 있어 보입니다. **분산분석(ANOVA)**을 통해 통계적으로 검정해보겠습니다!

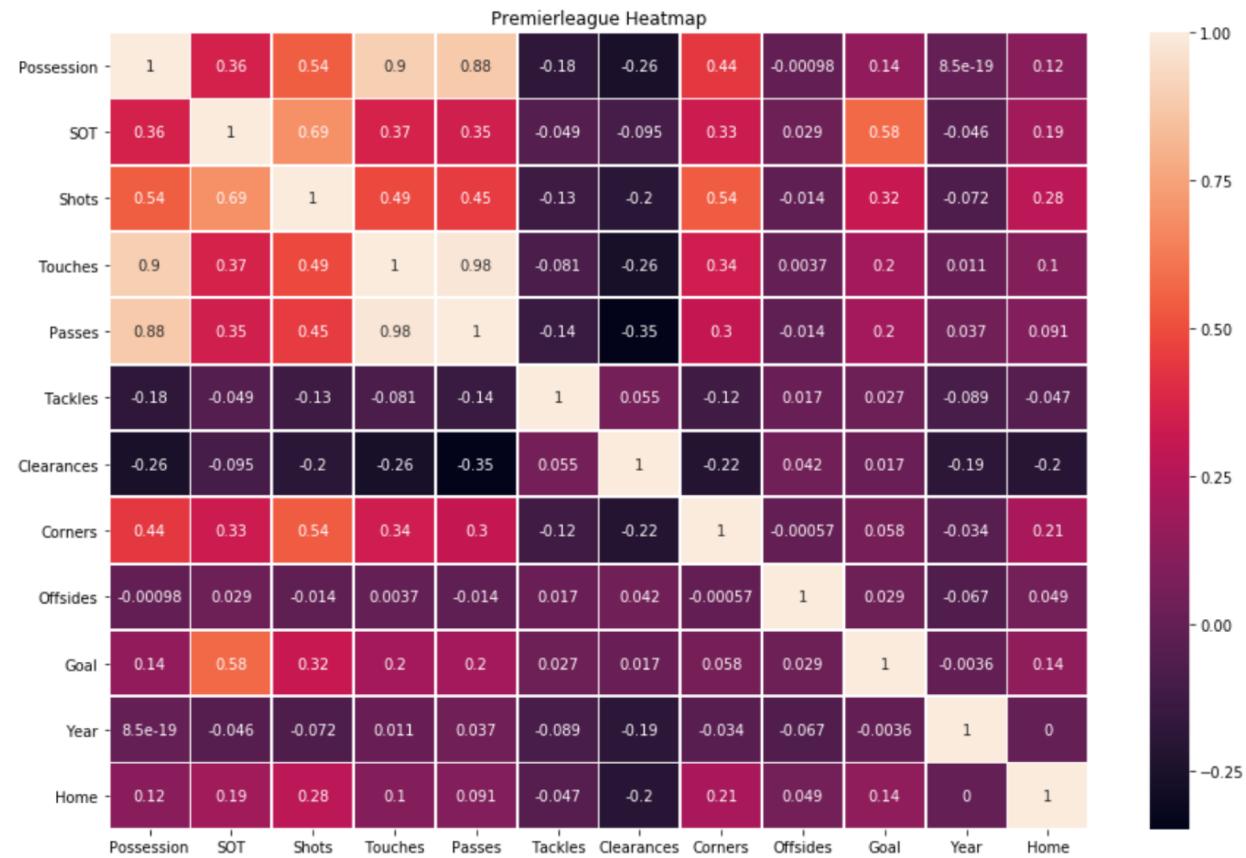
[Y별 각 변수 분산분석]

변수	F-값	P-value	H0 기각 여부
Possession	39.3	0.00000	기각
SOT	411.2	0.00000	기각
Shots	151.4	0.00000	기각
Touches	63.3	0.00000	기각
Passes	55.9	0.00000	기각
Tackles	1.6	0.20113	채택
Clearances	66.8	0.00000	기각
Corners	11.0	0.00002	기각
Offsides	1.6	0.21154	채택
Goal	1421.3	0.00000	기각
Year	0.5	0.60054	채택
Home	61.5	0.00000	기각

- 분산분석(ANOVA)은 세 집단 이상의 평균이 같은지를 판단하기 위해 만들어진 분석 방법입니다.
- H0(귀무가설)은 세 집단의 평균이 동일하다는 의미입니다. 분류 문제 경우는 집단끼리 평균이 달라야 분류에 도움이 된다고 생각하기 때문에 귀무가설을 따르면 안됩니다.
- 일반적으로 귀무가설을 판단하는 방법으로 P-value를 사용하며 보통 P-value가 0.05이하면 집단간 평균이 같지 않다고 판단합니다.
- 분산분석 결과 Tackles, Offsides, Year의 P-value값이 0.05가 넘게 나왔습니다. 이는 집단 간 평균이 같다고 본다는 의미고 제 분석엔 적합하지 않다고 판단했습니다.
- 나머지 변수들은 P-value값이 0.05보다 작게 나왔습니다. 이는 집단 간 평균이 다르게 볼 수 있다는 의미고 제 분석에 적합한 변수라 생각했습니다.

3. 어떤 요소가 경기에 승패에 큰 영향을 줄까?

[X변수간 상관 관계]



각 변수별로 상관관계를 살펴본 결과 Passes(패스)-Touches(볼터치)간 관계가 **0.98**로
가장 높은 상관관계를 지니고 있음을 확인할 수 있습니다.

3. 어떤 요소가 경기에 승패에 큰 영향을 줄까?

[X변수간 VIF를 통한 다중공선성 분석]

변수	VIF Factor
Possession	136.174
SOT	10.936
Shots	19.661
Touches	1583.094
Passes	732.884
Tackles	15.933
Clearances	12.093
Corners	6.988
Offsides	2.706
Goal	3.580
Year	110.940
Home	2.301

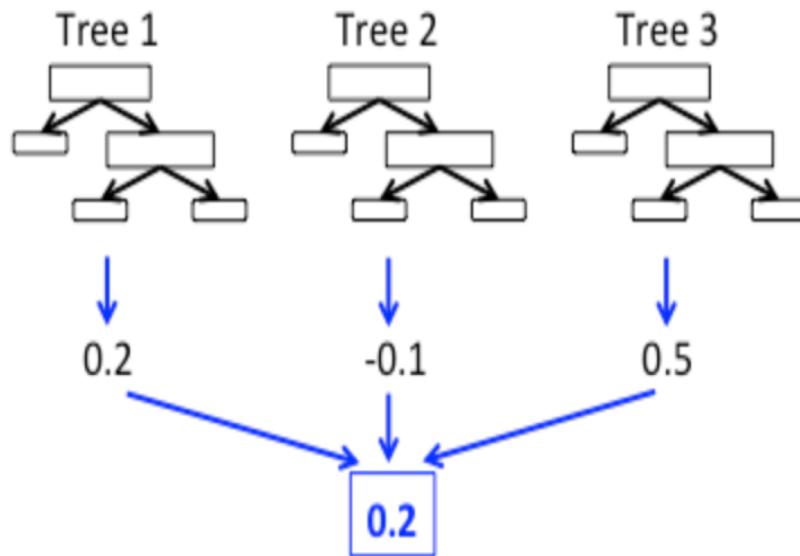


분산분석, 상관관계를 고려한
변수 제거후 VIF
(Possession, Touches,
Offsides, Year 제거)

변수	VIF Factor
SOT	10.877
Shots	18.093
Passes	13.363
Tackles	8.649
Clearances	5.448
Corners	6.223
Goal	3.461
Home	2.120

- VIF(Variance Inflation Factor, 분산팽창요인)는 주로 회귀문제에서 많이 다루는 문제지만 분류 문제에도 부정적인 영향을 미칠 것으로 판단해 실행했습니다.
- 일반적으로 VIF Factor값이 100이 넘어가면 다중공선성 문제를 지니고 있다고 판단합니다.
- 앞에서 진행했던 분산분석과, 상관관계를 고려해 총 4개의 변수(Possession, Touches, Offsides, Year)를 제거했습니다.

[RandomForest를 통한 분류 모델 생성]



- 랜덤포레스트(RandomForest)는 의사결정트리(DecisionTree)를 이용해 만들어진 알고리즘입니다.
- 랜덤포레스트는 양상을 학습방법의 일종으로 훈련 과정에서 구성한 다수의 결정 트리로부터 분류 또는 평균 예측치를 동시에 출력합니다.
- 한마디로 RandomForest는 여러 개의 의사결정 트리를 만들고 투표를 시켜 다수결로 결과를 결정하는 방법입니다.
- 모델을 장점은 오버피팅이 생길 경우를 대비할 수 있다는 겁니다.
- 분류문제를 해결하는 대표적인 알고리즘이며 성능이 제일 우수하게 나와 RandomForest로 분석을 진행했습니다.

[RandomForest를 통한 분류 모델 생성]

```

1 # train에서 train/test split
2 X1_train, X1_test, y1_train, y1_test = train_test_split(X_train, y_train, test_size=0.2, random_state=42)
3
4 # modeling
5 clf = RandomForestClassifier()
6 model = clf.fit(X1_train, y1_train)
7 predict_proba = model.predict_proba(X1_test)
8
9 # comparison
10 y_true = y1_test
11 y_pred = []
12
13 for i in range(760) :
14     y_pred.append(np.argmax(predict_proba[i]))
15
16 target_names = ['Lose', 'Win', 'Draw']
17 print('Confusion Matrix : \n\n',confusion_matrix(y_true, y_pred))
18 print('\n\n Classification Report : \n\n', classification_report(y_true, y_pred, target_names=target_names))

```

Confusion Matrix :

```

[[228 35 37]
 [ 30 214 32]
 [ 90 58 36]]

```

- 모델을 만들기 전 Train Data 자체에서 학습을 진행하고 싶어 8:2 비율로 X1_train과 X1_test를 만들어 진행했습니다.

Classification Report :

	precision	recall	f1-score	support
Lose	0.66	0.76	0.70	300
Win	0.70	0.78	0.73	276
Draw	0.34	0.20	0.25	184
avg / total	0.59	0.63	0.60	760

- Confusion Matrix를 대략적으로 보면 승/패는 어느정도 예측을 했지만 무는 예측율이 낮았습니다.
- Default 파라미터를 적용했는데 Precision(정밀도) 0.59, Recall(재현율) 0.63, F1-score 0.60이 나왔습니다.. 낮은 정밀도를 보이고 있습니다.

[Train Data에서 GridSearch를 최적의 파라미터 구하기]

```
1 # choose parameter
2 param_grid = [
3     {'n_estimators' : [10, 50, 100, 200, 300, 500, 1000], 'max_depth' : [2, 4, 6, 8, 10],
4      'min_samples_split' : [5, 10, 15, 20, 30]}]
5
6 model = RandomForestClassifier()
7 grid_search = GridSearchCV(model, param_grid, cv = 5, return_train_score = True).fit(X1_train, y1_train)
8
9 print('Best Parameter :\n\n', grid_search.best_params_)
```

Best Parameter :

```
{'max_depth': 6, 'min_samples_split': 15, 'n_estimators': 100}
```

GridSearch를 통해 최적의 파라미터를 구해보겠습니다. 우선, RandomForest의 n_estimators, max_depth, min_samples_split 3개의 파라미터에 관한 dictionary를 제작했습니다. 그리고 Dictionary를 GridSearchCV에 넣어 최적의 파라미터를 구해봤습니다. 그 결과 max_depth는 6, min_samples_split은 15, n_estimator는 100이 최적의 파라미터로 선정됐습니다.

[최적의 파라미터 적용해 Test Data 적용]

```

1 # 랜덤포레스트 (RandomForest)
2 clf = RandomForestClassifier(max_depth = 6, min_samples_split = 15, n_estimators = 100, criterion = 'entropy')
3 model = clf.fit(X_train, y_train)
4 predict_proba = model.predict_proba(X_test)
5
6 # comparison
7 y_true = test['Result']
8 y_pred = []
9
10 for i in range(760) :
11     y_pred.append(np.argmax(predict_proba[i]))
12
13 target_names = ['Lose', 'Win', 'Draw']
14 print('Confusion Matrix : \n\n',confusion_matrix(y_true, y_pred))
15 print('\n\n Classification Report : \n\n', classification_report(y_true, y_pred, target_names=target_names))

```

Confusion Matrix :

```

[[256  25   0]
 [ 47 232   2]
 [135  58   5]]

```

Classification Report :

	precision	recall	f1-score	support
Lose	0.58	0.91	0.71	281
Win	0.74	0.83	0.78	281
Draw	0.71	0.03	0.05	198
avg / total	0.67	0.65	0.56	760

- GridSearch의 결과로 나온 파라미터를 적용해 모델을 만들어봤습니다.
- Confusion Matrix를 보면 승/패의 예측은 높아졌지만 무승부를 거의 잡지 못했습니다..
- Report를 살펴보면 Precision 0.67, Recall 0.65, F1-score가 0.56의 결과가 나왔습니다.
- Precision과 Recall 값이 상승한 반면 F1-score는 떨어졌습니다.

[Cross Validation을 통해 모델 적합성 평가하기]

```
1 # validation score
2 clf = RandomForestClassifier(max_depth = 6, min_samples_split = 15, n_estimators = 100).fit(X_train, y_train)
3 scores = cross_val_score(clf, X_train, y_train, cv= 10)
4 print('Corss Validation Score :\n\n', scores)
```

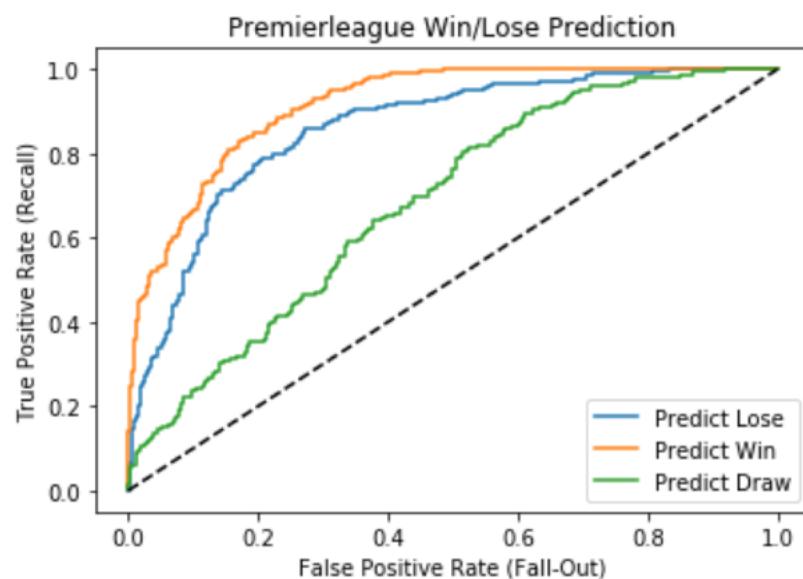
Corss Validation Score :

```
[0.62105263 0.64473684 0.64736842 0.65526316 0.62894737 0.63157895
 0.62631579 0.62894737 0.66052632 0.67894737]
```

혹시나 제 모델이 오버피팅이 된 것은 아닌지 검증하기 위해 K-Fold Cross Validation을 적용해봤습니다. Fold를 10개로 설정해 검증한 결과 각 Fold별 Score가 대부분 0.62 ~ 0.67사이에 분포하고 있음을 확인할 수 있습니다. 크게 오버피팅이 된 모델은 아니라고 할 수 있습니다.

[ROC Curve, AUC 확인]

ROC Curve :



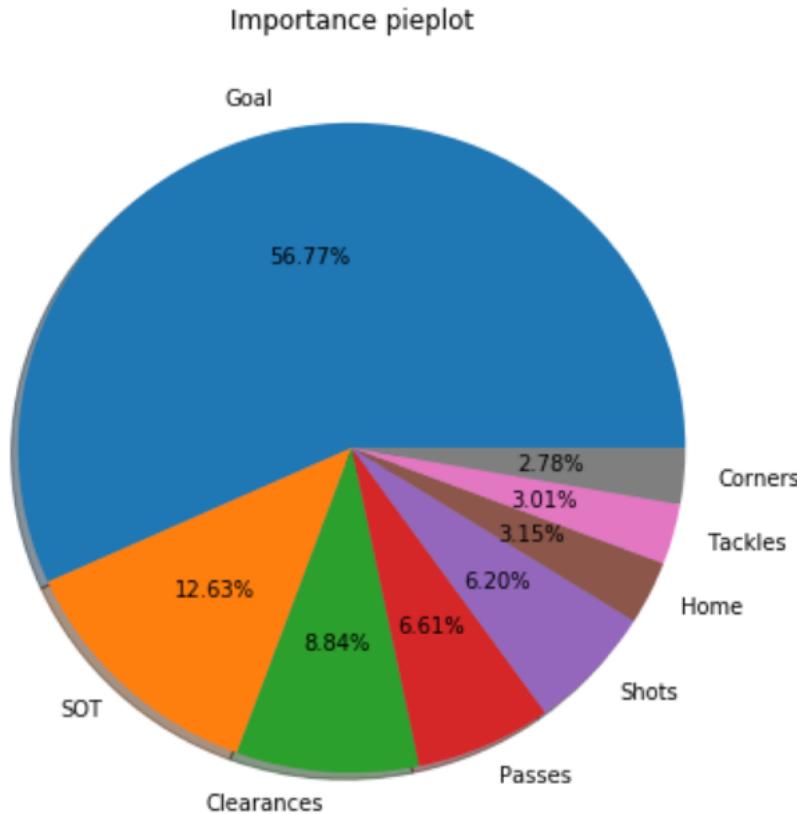
- ROC Curve는 클래스 판별 기준값의 변화에 따른 위양성률(Fall-out)과 재현율 (Recall)의 변화를 시각화한 것입니다.
- 좋은 분류기일수록 가운데 대각선 점선으로부터 멀어야 합니다. 보면 승/패에 관한 ROC Curve는 대각선과 어느정도 떨어진 모습을 보이나 무에 관련된 ROC커브는 대각선과 가까이 있습니다. 좋은 분류 모델이라고 평가할 수 없습니다.
- AUC(Area Under the Curve)는 ROC curve의 면적을 의미합니다. 따라서 좋은 모델일수록 면적이 크게됩니다. AUC Score를 보면 승을 예측하는 모델이 0.915로 가장 높고 패를 예측하는 모델이 0.859, 무를 예측하는 모델이 0.683으로 가장 낮은 면적을 지녔습니다.

AUC Score :

0.8590331280321548 0.915014227445969 0.683417807972968

6. 아스널은 어떤 요소를 중점적으로 준비해야 할까?

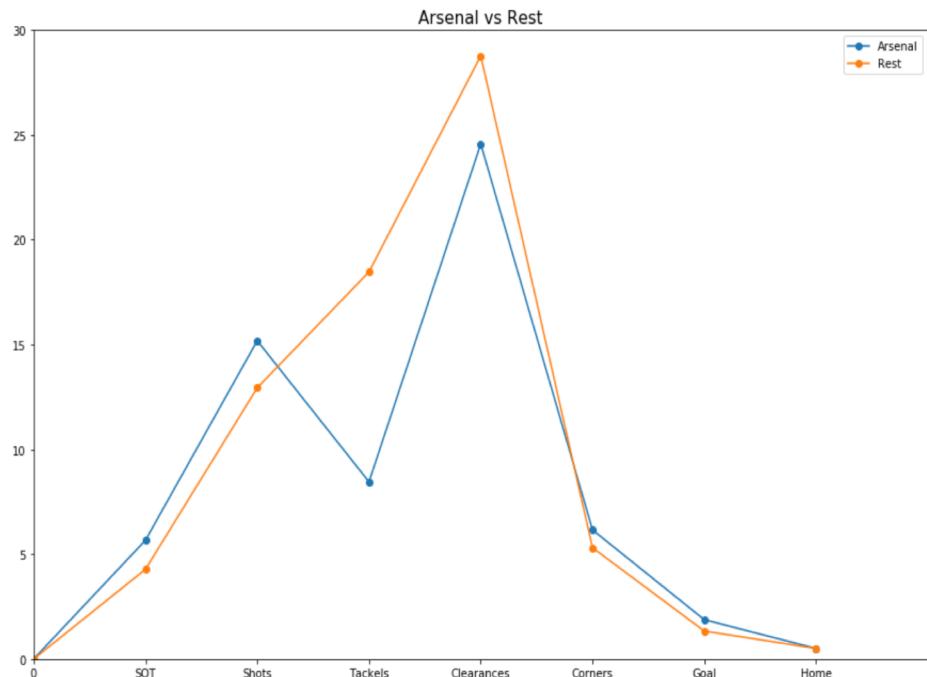
[GridSearch를 통한 변수 별 중요도 파악]



- GridSearch를 진행할 때 어떤 요소를 중점적으로 분류 모델에 적용했는지를 확인할 수 있습니다. 이를 통해 아스널이 나아가야 할 방향을 모색해볼겁니다!
- 각 변수 별 중요도를 도출한 결과 Goal(골)이 56.77%로 모델에 엄청난 영향을 끼쳤습니다. 사실 골은 승리와 직접적으로 관련이 있을 수 밖에 없습니다. 골을 한 골도 못 넣는다면 그 경기를 이길 순 없기 때문입니다.
- 두번째로 높은 영향을 변수는 SOT(유효슈팅)입니다. 골문 안으로 들어가는 숫이 많으면 그만큼 골이 들어갈 확률도 높아지기 때문에 나온 결과가 아닐까 싶습니다.
- 세번째로 높은 영향을 끼친 변수는 Clearances(걷어냄)입니다. 골문 앞에 결정적인 슈팅을 걷어 낸 것은 다른 관점으로 보면 한 골을 넣은 것과 다름 없다고 생각합니다.
- 그 다음으로는 Passes, Shots, Home, Tackles, Corners가 뒤따랐습니다.

6. 아스널은 어떤 요소를 중점적으로 준비해야 할까?

[Arsenal vs Rest를 통해 Arsenal이 나아갈 길을 알아보겠습니다!]



- 지난 7년간 아스널의 데이터를 종합해 나머지 팀들과 비교하면서 아스널이 보완해야 할 점을 알아보겠습니다.
- 제일 중요한 Goal(골) 변수를 보면 평균보다는 높지만 그리 월등한 수준은 아닙니다. 우승을 하기 위해선 더 많은 골이 필요하고 기존보다 더 공격적인 모습이 필요할 것으로 보입니다.
- 다음으로 SOT(유효슈팅) 변수를 보면 Goal과 비슷한 양상을 보입니다. 공격적인 모습을 통해 유효슈팅을 늘려야 할 것으로 보입니다.
- Clearances(걷어냄)은 오히려 평균보다 떨어집니다. 이는 당장 아스널이 수비 보완이 시급한 상황이라는 것을 알 수 있고 이번 여름 이적시장에서 수비 보완은 필수적이라 생각합니다.
- 전반적으로 봤을 땐 기존보다 더 적극적으로 공격적인 전술이 필요한 것 같고 수비력도 지금보다 더 보완해야 할 것으로 보입니다.

[많은 한계점을 지닌 이번 프로젝트..]

축구는 기록이 전부가 아니다

스포츠는 기록이 전부인 경우가 없습니다. 축구의 경우는 그날 선수들의 컨디션, 경기장 상황, 라이벌 관계, 감독 전술 등 여러 요소가 복합적으로 이뤄집니다. 이런 요소를 적용하지 못해서 굉장히 어렵다고 생각했습니다.

다양한 변수가 있었으면..

승부에 영향을 주는 다양한 변수들이 있었으면 더 좋은 모델을 만들 수 있을 것이라 생각합니다. 예를 들어 팀의 연봉에 따라 어느정도 실력의 차이가 있다고 생각합니다. 또한 부상선수가 많은 팀은 그만큼 실력이 줄어들 수도 있을 것이라 생각합니다. 해당 경기를 하기 전 며칠간 휴식을 취했는지 등도 승부에 영향을 미치는 요인들이라 생각합니다. 이런 다양한 데이터들을 구하는데 한계를 느껴 아쉬웠습니다.

감사합니다