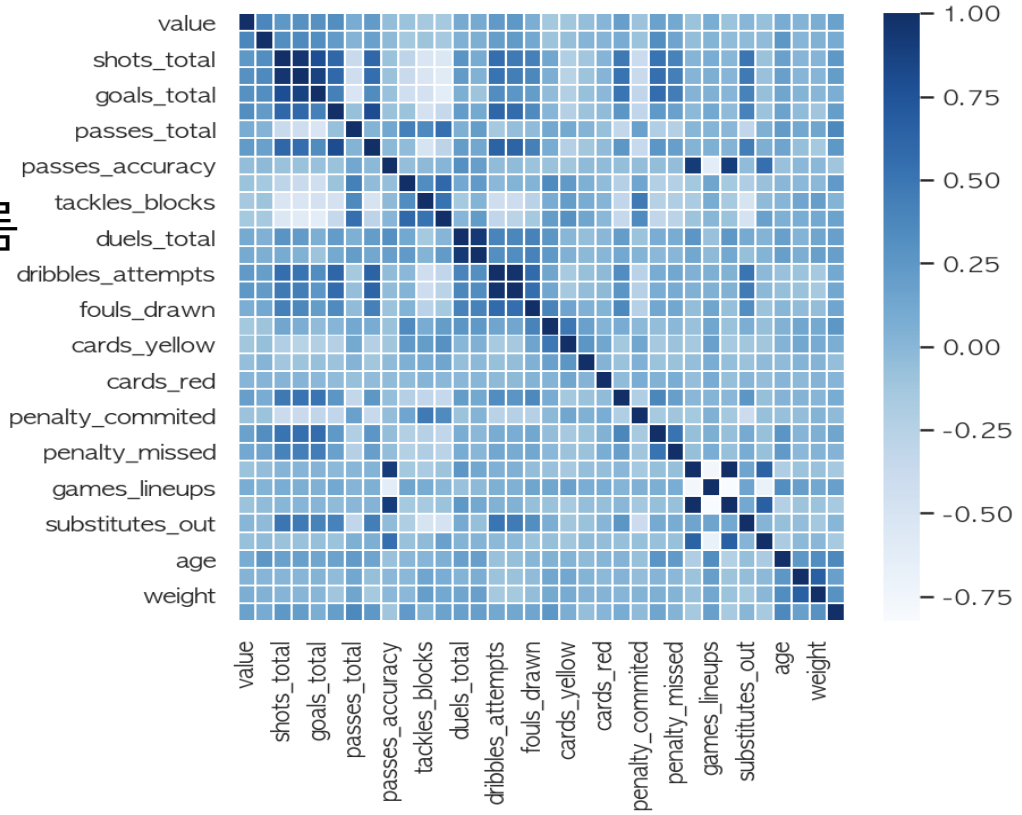
Utilization of SNS Index for Machine-Learning Based Market-Value Prediction of Football Players

기계학습 기반 축구선수 몸값 예측을 위한 SNS데이터의 활용

고건호, 김정섭, 이왕건

축구선수 시장 가치와 연구 배경

- **축구선수 별 몸값에 영향을 미치는 너무 많은 변수로 인한 변동성**
  - 구단의 자본 – 코로나로 수입이 감소한 구단들로 인해 감소한 선수들의 몸값
  - 선수의 포지션, 리그, 나이, 이미지 및 영향력
  - 2014년부터 확인된 시장 내 inflation
  - 돈이 많은 구단들로 인하여 과하게 책정되는 시장가치
- **예측의 어려움**
  - 포지션 별 선수 가치를 평가할 수 있는 데이터가 각기 다름
  - 접근 가능한 다수의 데이터셋들의 편향성
  - 공격수를 제외한 다른 포지션의 데이터 부족
  - 공격수만을 대상으로 발표한 논문 다수
  - 인지도는 포지션과 상관 없이 선수의 능력치 반영 값
- **복잡한 변수들의 구조**
  - 변수들 사이에 다중공산성 존재
  - 중요한 데이터 선별을 위한 추가 데이터 필요



선행 연구

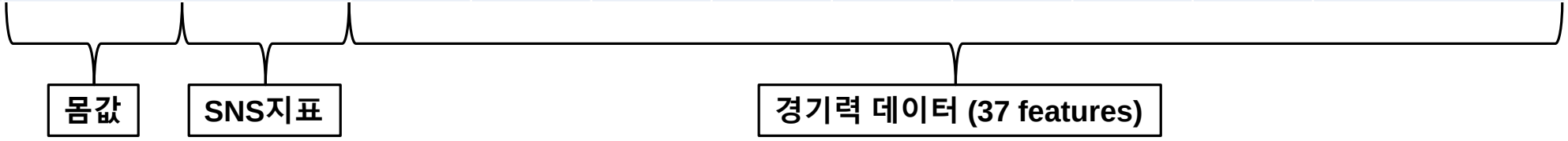
Author	Feature	Modeling	Description
Müller, O. (2017)	Facebook likes + Google search hits + UEFA mentions	Regression	Data-driven estimation of market value in association football
Yaldo, L. (2017)	17 Performance features	Machine Learning(Decision-Tree, Regression)	Non-Performance variable의 결합 모델 연구 필요성 제시
Nsolo, E. (2018)	37 performance features	Machine Learning(RandomForest, Decision-Tree, Logistic regression)	Study Focused on top tier players(upper 10% players)

문제 정의

- 기존 연구의 한계
  - 경기력만으로 선수의 시장가치 예측의 한계
  - 대부분의 연구에서 인지도 정보의 고려 X
- 한계점 보완 방법
  - SNS 지표 활용
  - 일방적 소통 가능한 Instagram 팔로워수 사용

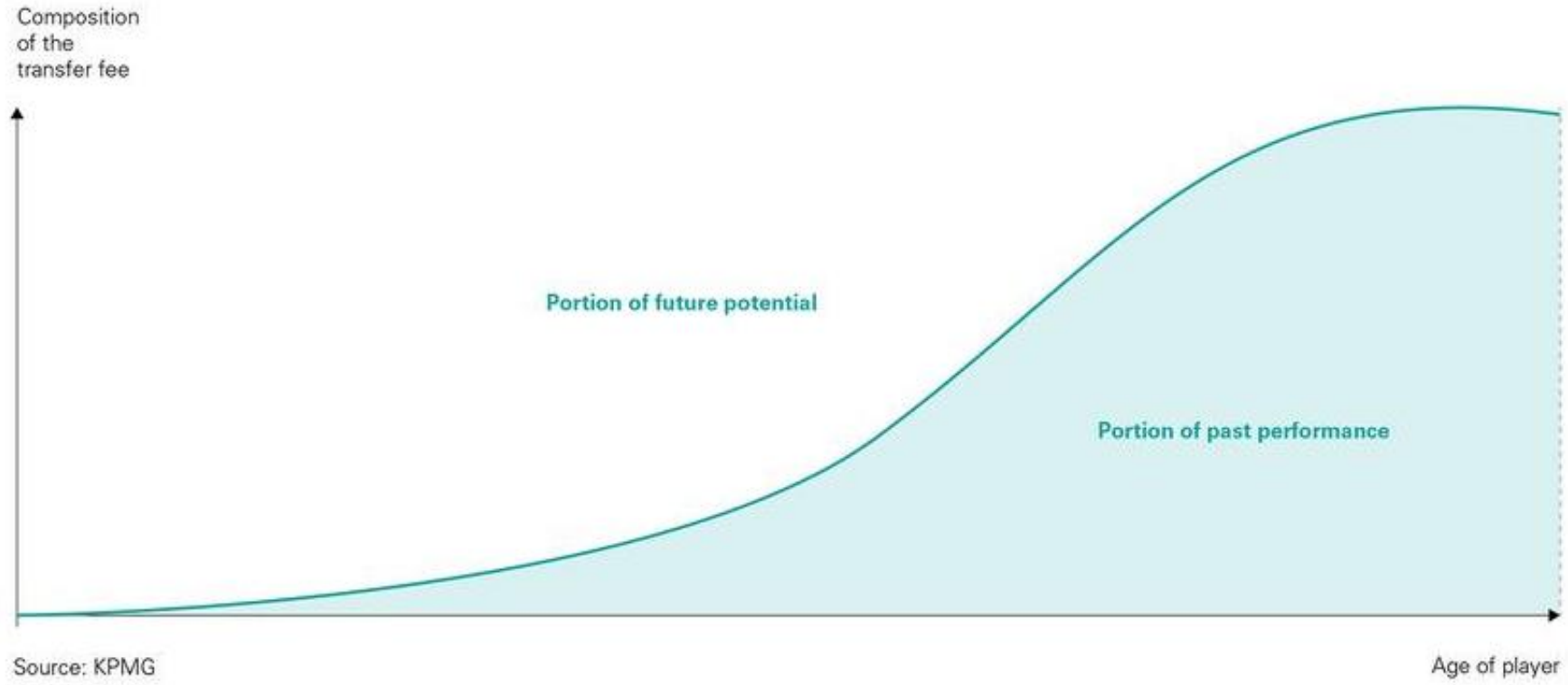
데이터 구조

	Market Value (€mil)	SNS Index (Followers)	Age	Height	Weight	Rating	Shots_ total	Goals_ total	Passes _total	Tackles _total	+ 29 more performance data
mean	35.63747	4538383	26.01538	180.4713	74.53705	5.677302	49.0561	0.212703	36.15977	0.904722	-
std	22.10662	16965945	3.062066	15.06138	10.83103	1.194869	2.887455	0.206722	16.71964	0.596441	-
min	16	0	20	0	0	1.166667	39.72333	0	2.531	0	-
max	180	222106900	35	199	100	8.222222	56.98	0.9859	81.7024	3.8404	-



- 경기력 관련 데이터 → 한 경기당 데이터로 변환
  - 경기 횟수 (총 경기 시간 / 90)로 나누어 전처리 작업
- SNS지표의 기여도 및 영향력을 확인을 위해 데이터 모두 사용

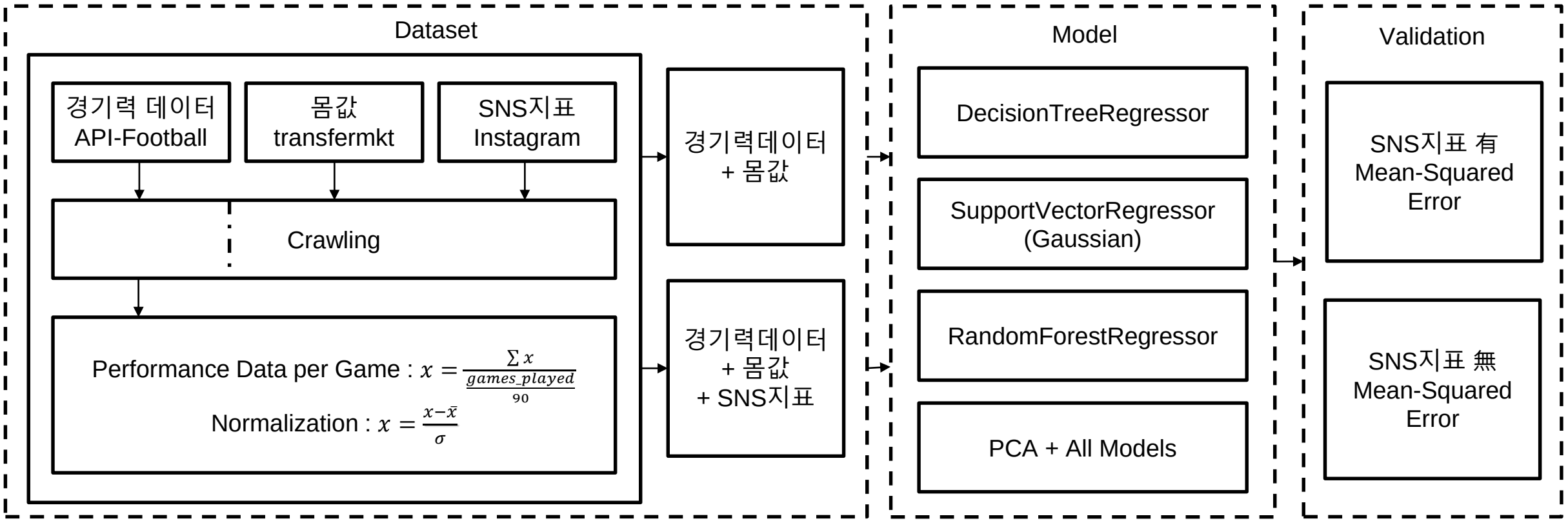
이전의 모든 데이터를 기반으로 만들어지는 몸값



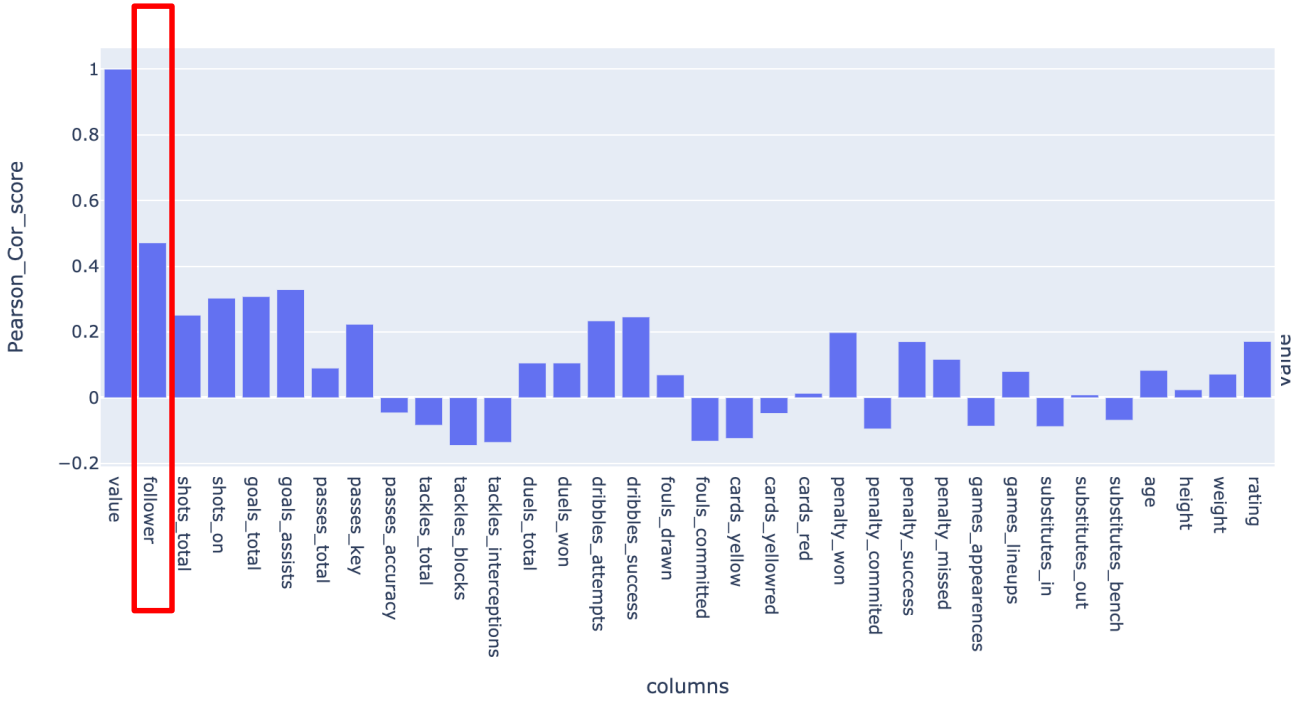
Valuation methods for football players

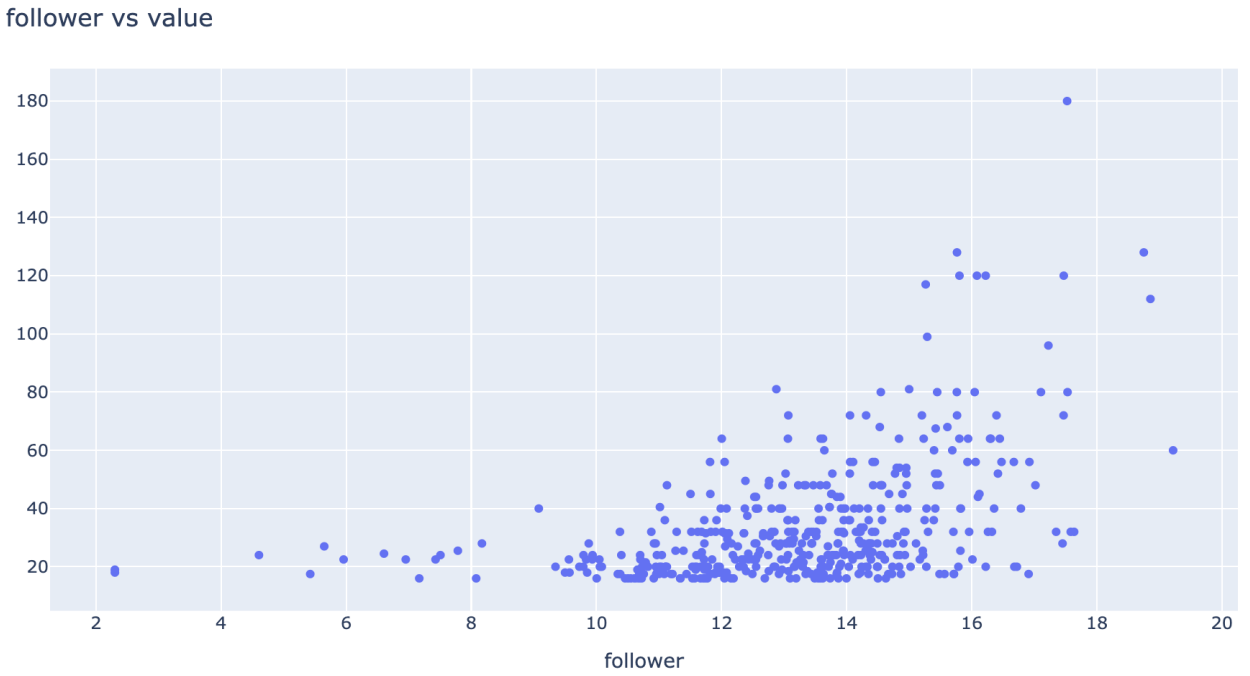
Generally speaking, players are valued in the same way as any other asset. But for various reasons, the only appropriate valuation method for a team sport player is the market approach, which is based on historical market prices realized between third parties.

SNS 데이터를 활용한 기계학습 방법론



SNS 데이터의 활용 가능성





다른 변수들에 비해 SNS지표가 몸값과 상당히 높은 상관관계

SNS 데이터의 수집과 가공

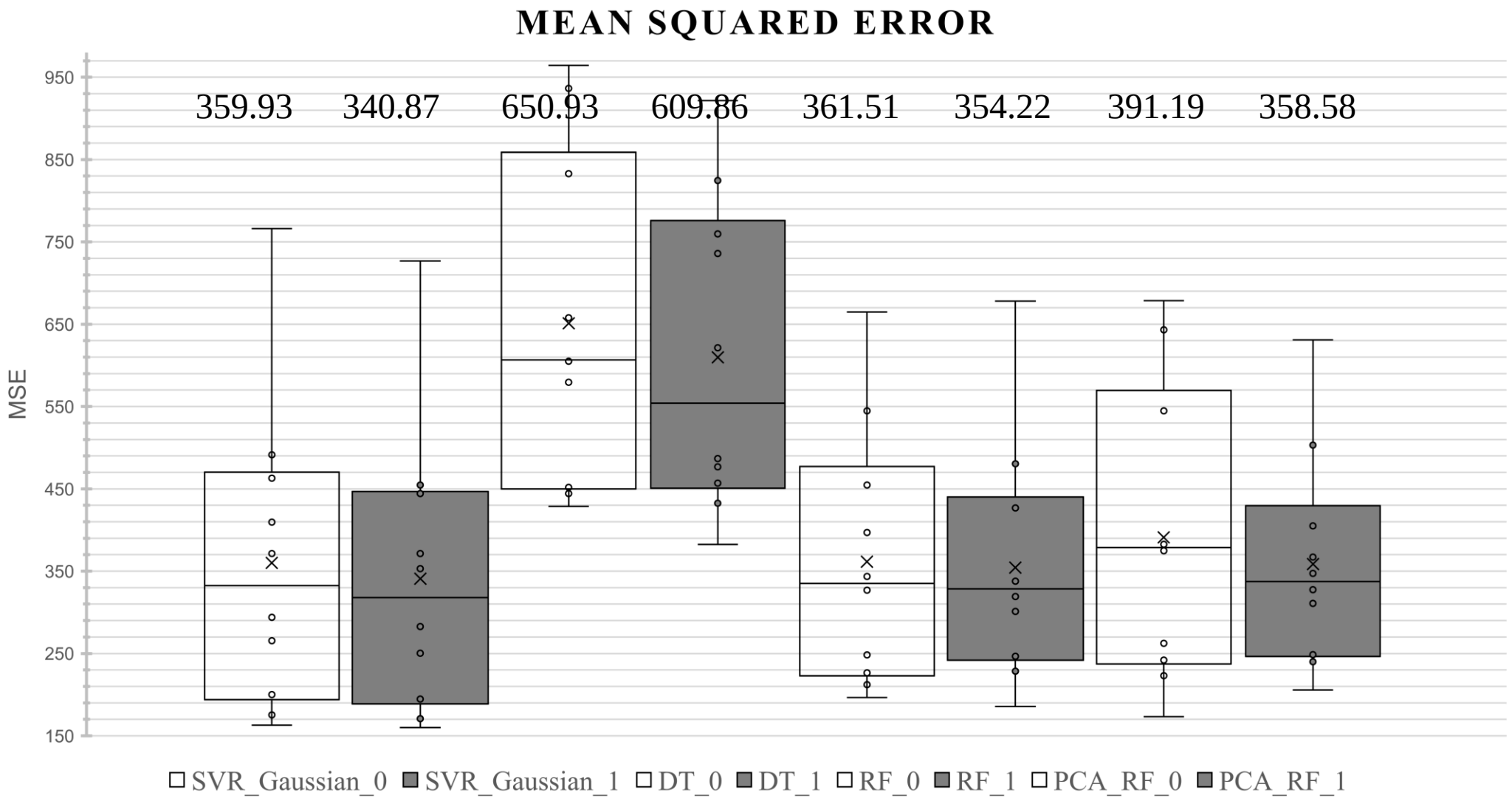
- 수집 방법
  - 수집된 몸값 데이터 기반 500명에 대해 Instagram Follower 정보 크롤링 (Selenium 활용)

데이터셋 및 실험 설계

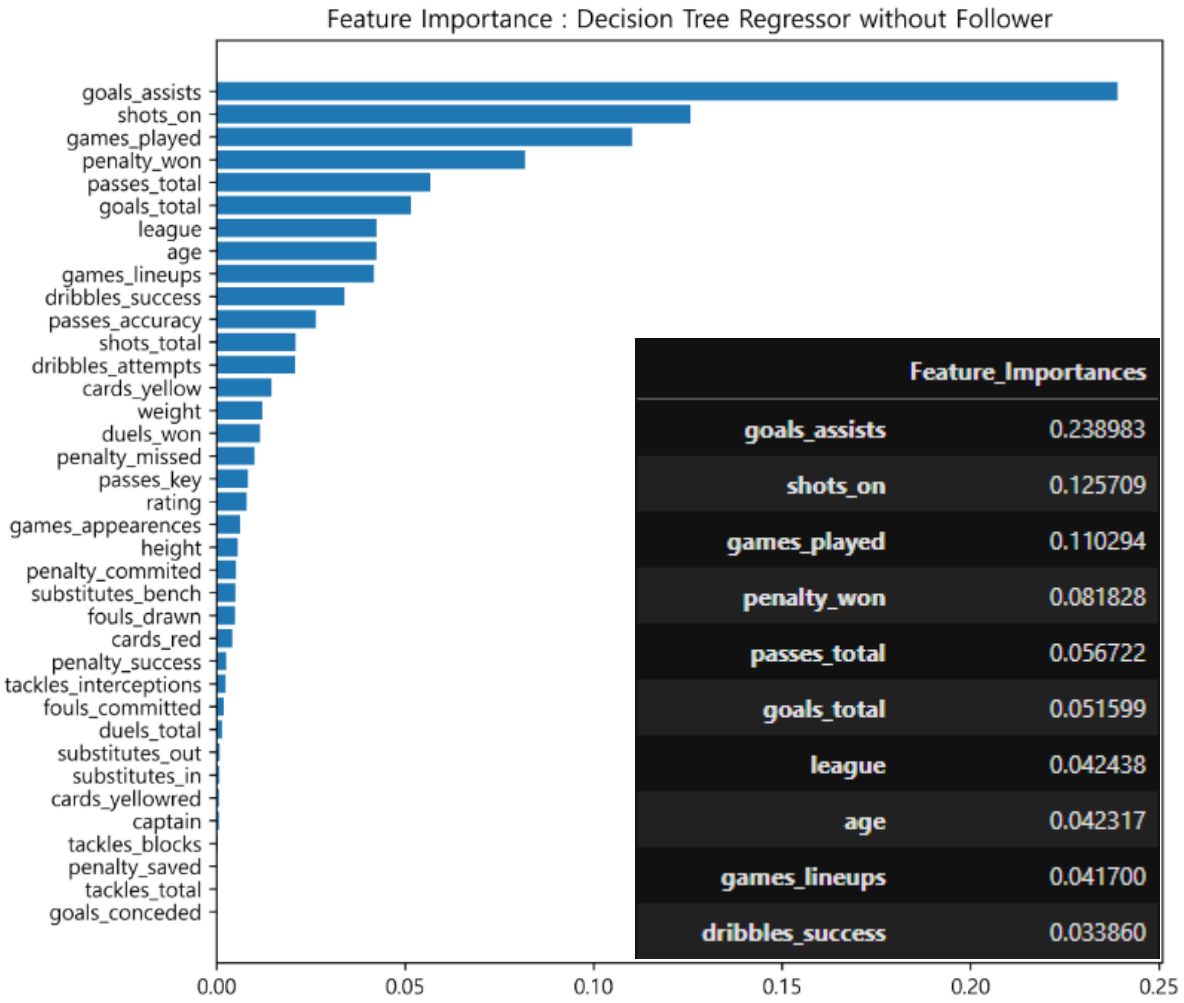
- 데이터셋
  - 선수 경기력 – API-Football
  - 시장가치 – transfermkt
  - SNS지표 – Instagram

- 실험 설계
  - 기계학습 방법 별 SNS지표 유/무 10겹 교차검증
  - 최적화 및 하이퍼파라미터 튜닝
    - Random Forest Regressor – RandomizedSearchCV
    - Support Vector Regressor – Cost Variation
  - SNS 지표 유/무 Feature Importance 확인
  - 95% 설명력 유지 PCA 후, 모델의 변화 확인

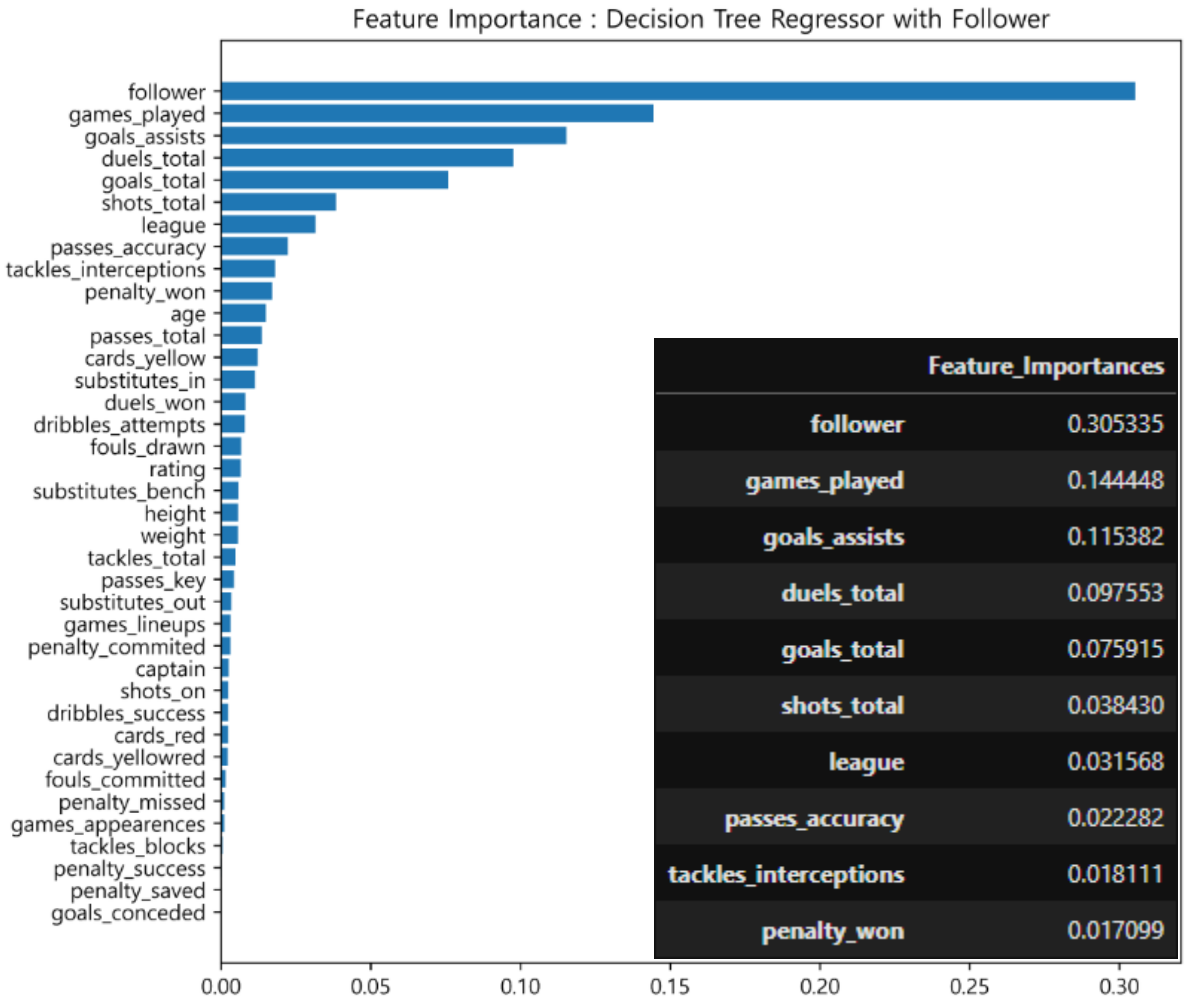
기계학습 방법 별 SNS 지표 유/무 시, 10겹 교차 검증



실험 1.1 : Feature Importance_Decision Tree Regressor



	Feature_Importances
goals_assists	0.238983
shots_on	0.125709
games_played	0.110294
penalty_won	0.081828
passes_total	0.056722
goals_total	0.051599
league	0.042438
age	0.042317
games_lineups	0.041700
dribbles_success	0.033860

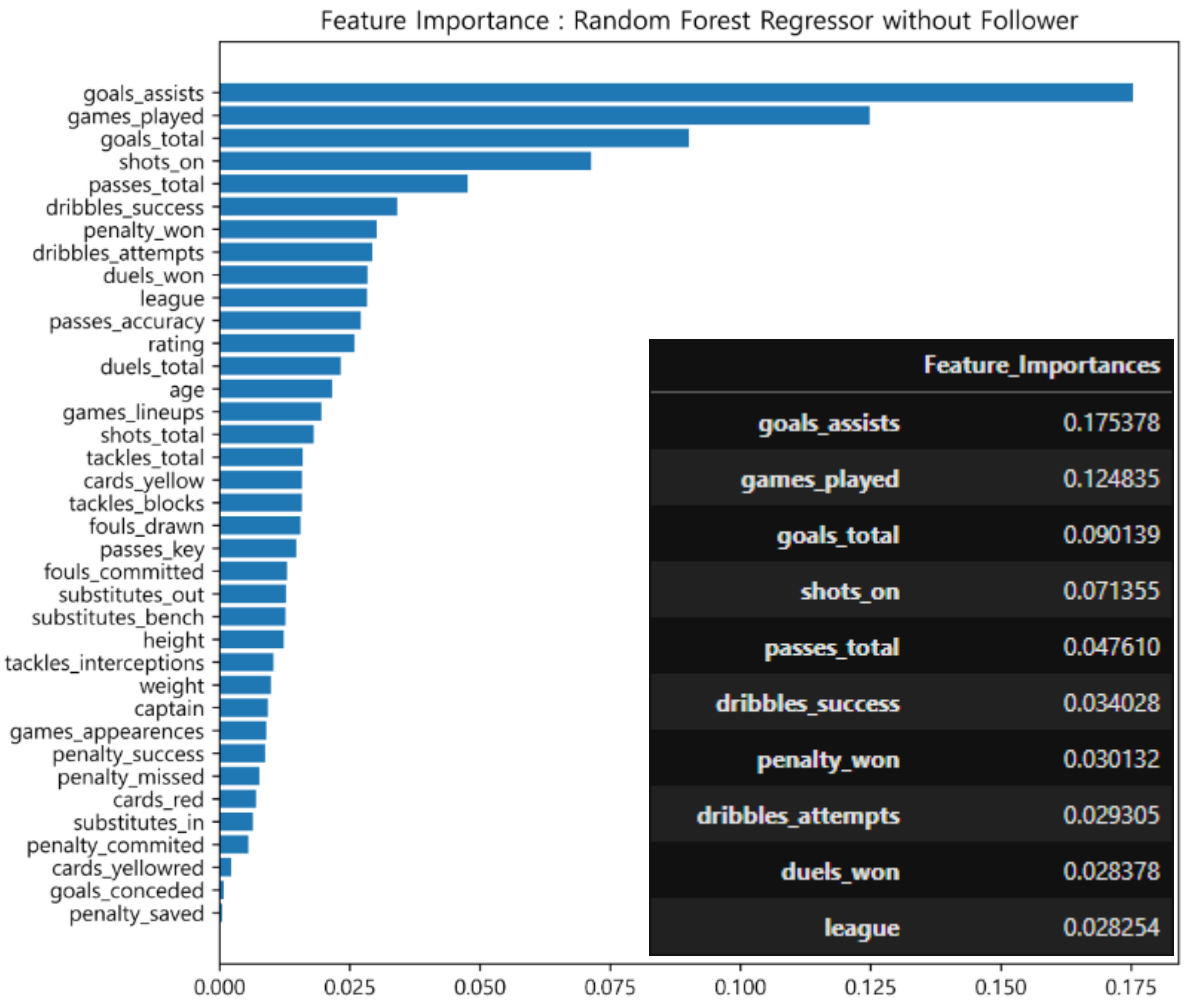


	Feature_Importances
follower	0.305335
games_played	0.144448
goals_assists	0.115382
duels_total	0.097553
goals_total	0.075915
shots_total	0.038430
league	0.031568
passes_accuracy	0.022282
tackles_interceptions	0.018111
penalty_won	0.017099

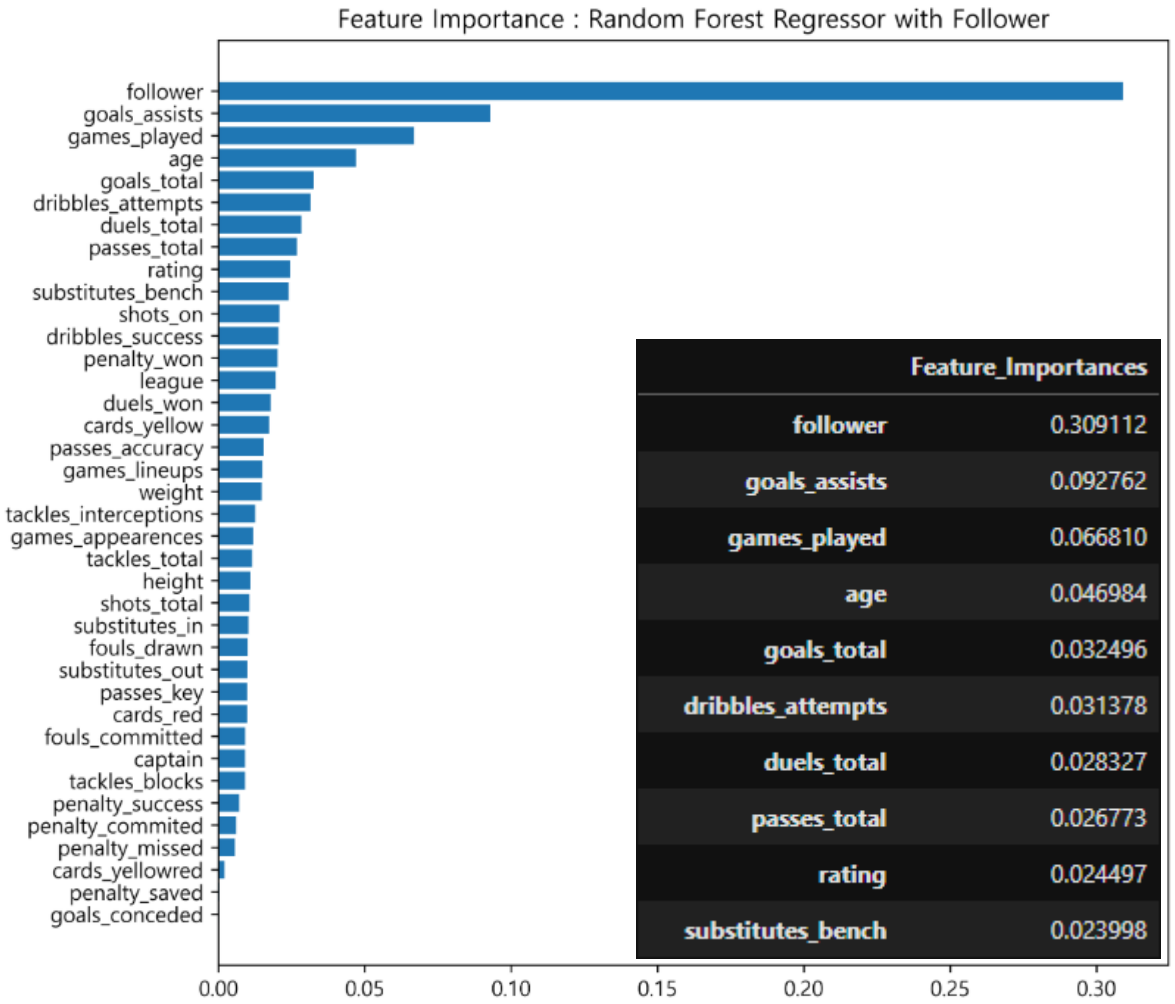
goal_assists : 0.238938

follower : 0.305335

실험 1.2 : Feature Importance_Random Forest Regressor



	Feature_Importances
goals_assists	0.175378
games_played	0.124835
goals_total	0.090139
shots_on	0.071355
passes_total	0.047610
dribbles_success	0.034028
penalty_won	0.030132
dribbles_attempts	0.029305
duels_won	0.028378
league	0.028254



	Feature_Importances
follower	0.309112
goals_assists	0.092762
games_played	0.066810
age	0.046984
goals_total	0.032496
dribbles_attempts	0.031378
duels_total	0.028327
passes_total	0.026773
rating	0.024497
substitutes_bench	0.023998

goal_assists : 0.175378

follower : 0.309112

실험 2 : RFR RandomizedSearchCV를 통한 Hyper-parameter Tuning

```python
from sklearn.model_selection import RandomizedSearchCV
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error

n_estimators = [int(x) for x in np.linspace(start = 100, stop = 1200, num = 23)]
max_features = ['auto', 'sqrt', 'log2']
max_depth = [int(x) for x in np.linspace(5, 100, num = 20)]
min_samples_split = [2, 5, 10, 15, 30, 50, 75, 100]
min_samples_leaf = [1, 2, 4, 10]

random_grid = {'n_estimators': n_estimators,
               'max_features': max_features,
               'max_depth': max_depth,
               'min_samples_split': min_samples_split,
               'min_samples_leaf': min_samples_leaf
              }
```

```python
rf_random = RandomizedSearchCV(estimator=rf,
                               param_distributions=random_grid, n_iter=10,
                               cv=5,
                               verbose=2,
                               random_state=42,
                               n_jobs = 1
                              )

X_train, X_test, y_train, y_test = train_test_split(df_0.drop('value', axis=1), df_0.value, test_size=0.2)
```

```
RandomForestRegressor(bootstrap=True, ccp_alpha=0.0, criterion='mse',
                      max_depth=30, max_features='auto', max_leaf_nodes=None,
                      max_samples=None, min_impurity_decrease=0.0,
                      min_impurity_split=None, min_samples_leaf=2,
                      min_samples_split=10, min_weight_fraction_leaf=0.0,
                      n_estimators=800, n_jobs=None, oob_score=False,
                      random_state=None, verbose=0, warm_start=False)
```

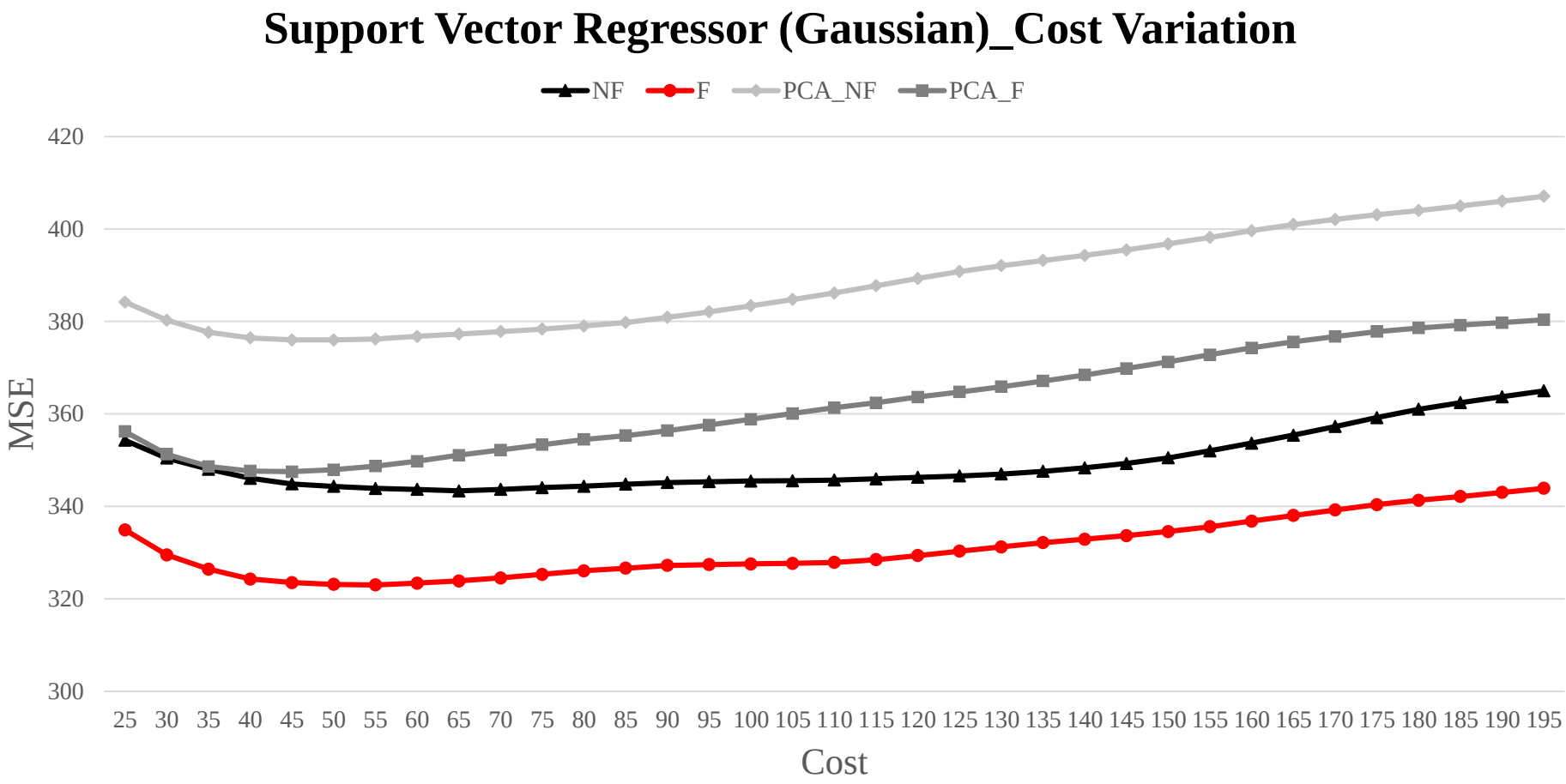
SNS X

```
Base Accuracy:  20.8117530897007
Random Accuracy:  20.44462771070915
Improvement of -0.02%.
```

SNS O

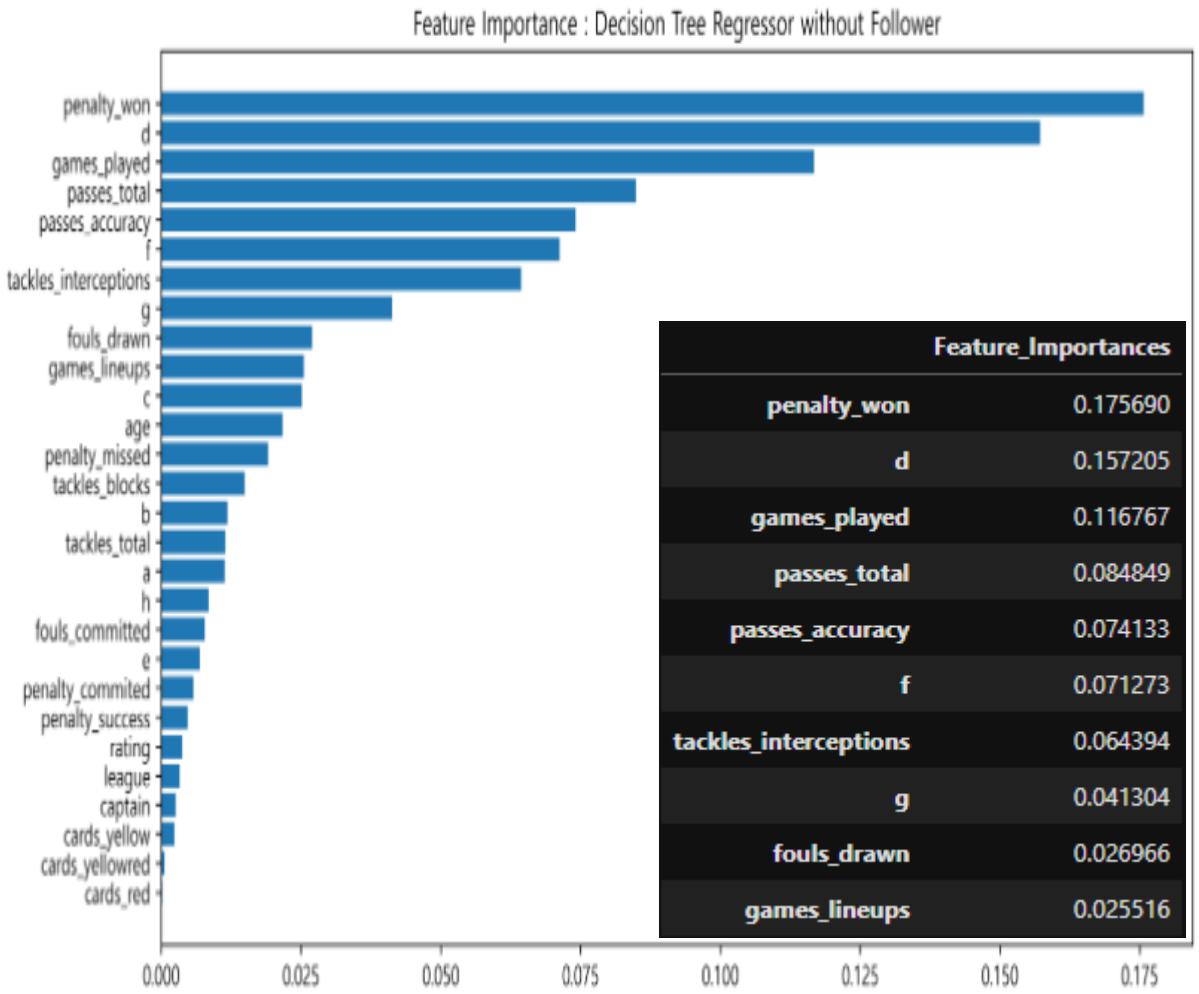
```
Base Accuracy:  18.38475817989819
Random Accuracy:  17.841292266584837
Improvement of -0.03%.
```

실험 3 : Support Vector Regressor(Gaussian)_Cost Variation

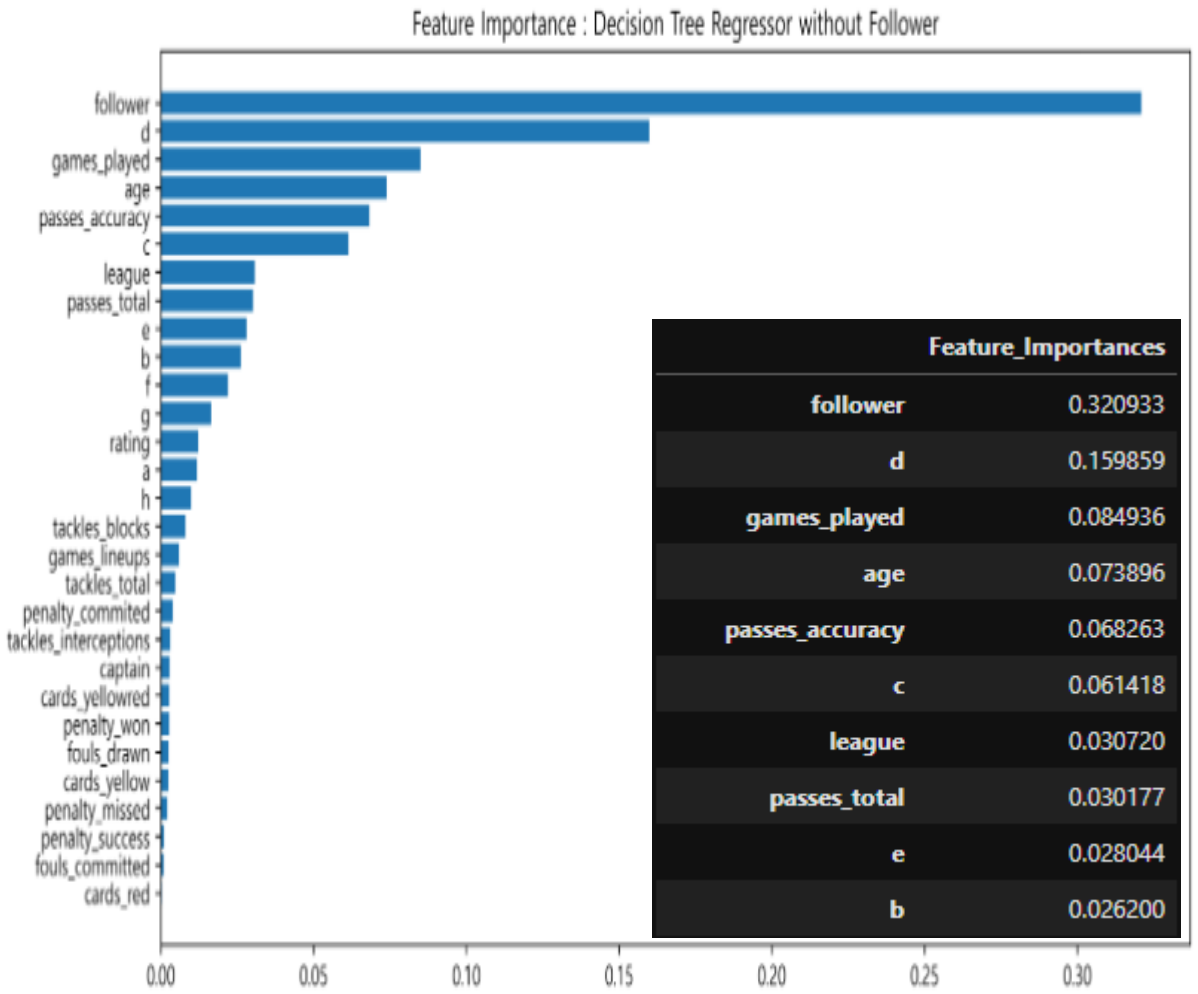


- $Cost \propto Accuracy$
- Cost의 변동에도 SNS지표를 활용했을 때 MSE 감소

실험 4.1 : PCA + Decision Tree Regressor_Feature Importance



	Feature_Importances
penalty_won	0.175690
d	0.157205
games_played	0.116767
passes_total	0.084849
passes_accuracy	0.074133
f	0.071273
tackles_interceptions	0.064394
g	0.041304
fouls_drawn	0.026966
games_lineups	0.025516

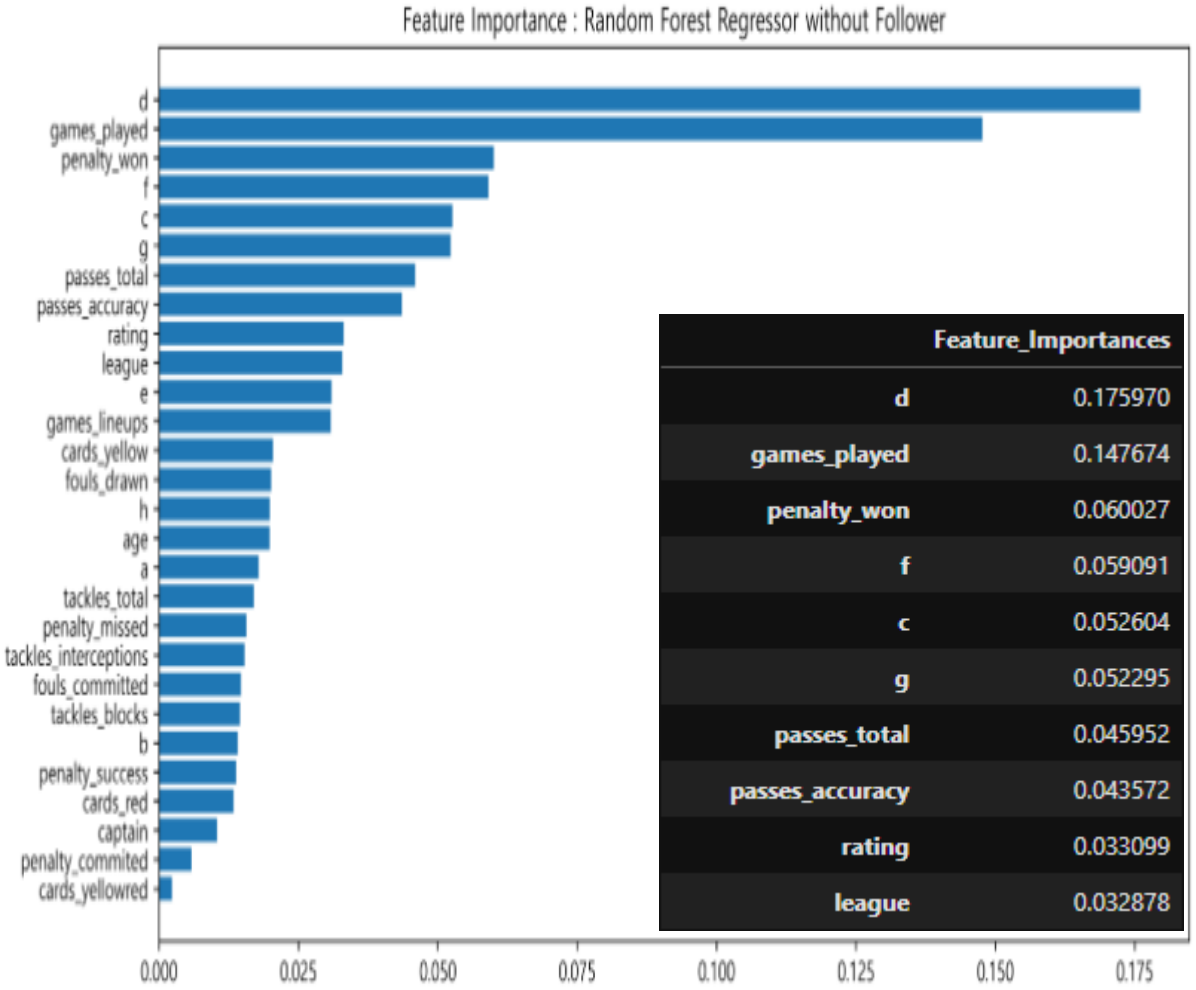


	Feature_Importances
follower	0.320933
d	0.159859
games_played	0.084936
age	0.073896
passes_accuracy	0.068263
c	0.061418
league	0.030720
passes_total	0.030177
e	0.028044
b	0.026200

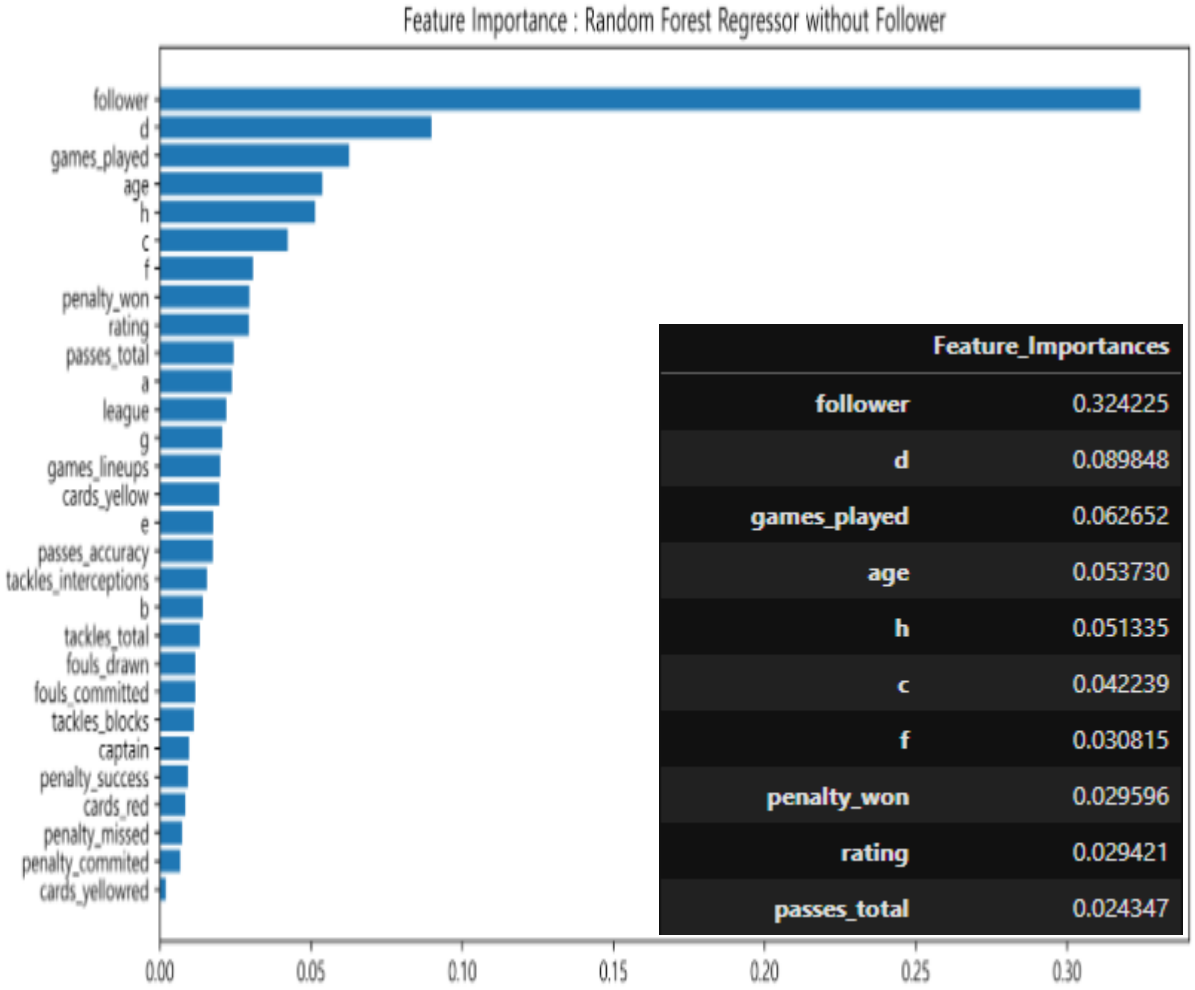
penalty_won : 0.175690

follower : 0.320933

실험 4.2 : PCA + Random Forest Regressor_Feature Importance



	Feature_Importances
d	0.175970
games_played	0.147674
penalty_won	0.060027
f	0.059091
c	0.052604
g	0.052295
passes_total	0.045952
passes_accuracy	0.043572
rating	0.033099
league	0.032878



	Feature_Importances
follower	0.324225
d	0.089848
games_played	0.062652
age	0.053730
h	0.051335
c	0.042239
f	0.030815
penalty_won	0.029596
rating	0.029421
passes_total	0.024347

penalty_won : 0.175970

follower : 0.324225

결론 및 향후 연구

- 축구선수 몸값 예측 시, SNS 지표 활용
  - SV Regressor, DT Regressor, RF Regressor 예측 모델 10겹 교차검증 결과
    - SNS데이터 활용 후, MSE 모두 감소
  - 모델 최적화 후,
    - MSE 감소 확인
    - 모든 모델의 가장 높은 Feature Importance로 SNS데이터(Follower) 확인

- 향후 연구
  - 유동적인 몸값이 아닌, 정량화된 연봉을 SNS 지표 활용 예측
  - 가중치 적용 – 리그, 포지션, 국가

Q&A

감사합니다.