# KBO Scouting Challenge

2019년 11월 14일

이광진

KBO Scouting Challenge

1. Mission

# Past
# 2011 ~ 2018

# Forelgner Pltcher

# New
# 2019

# KBO Scouting Challenge
## 2. Data

1. **kbo_yearly_foreigner_2011_2018**
   - **2011**년부터 **2018**년까지 **KBO**에서 활약한 외국인 투수들의 역대 **KBO** 정규시즌 성적
2. **fangraphs_foreigners_2011_2018**
   - 외국인 투수들의 **KBO** 입성 전, 메이저리그에서의 역대 정규시즌 성적
3. **baseball_savant_foreigners_2011_2018**
   - 외국인 투수들의 **KBO** 입성 전, 메이저리그에서의 스탯캐스트 데이터

**Train Data**

4. **fangraphs_foreigners_2019**
   - **2019**년 신규 외국인 투수들의 메이저리그에서의 역대 정규시즌 성적
5. **baseball_savant_foreigners_2019**
   - **2019**년 신규 외국인 투수들의 메이저리그에서의 스탯캐스트 데이터

**Test Data**

# KBO Scouting Challenge
## 2. Data

### 1. kbo_yearly_foreigner_2011_2018

- 2011년부터 2018년까지 KBO에서 활약한 외국인 투수들의 역대 KBO 정규시즌 성적

| | Column_name | Description |
|---|---|---|
| 1 | pitcher_name | 투수 이름 |
| 2 | year | 년도 |
| 3 | team | 소속팀 |
| 4 | ERA | 평균자책점 |
| 5 | TBF | 상대한 타자수 |
| 6 | H | 피안타수 |
| 7 | HR | 피홈런수 |
| 8 | BB | 피볼넷수 |
| 9 | HBP | 피사구수 |
| 10 | SO | 삼진수 |
| 11 | year_born | 생년월일 |

# 2. fangraphs_foreigners_2011_2018

- 외국인 투수들의 KBO 입성 전,
  메이저리그에서의 역대 정규시즌 성적

| | Column_name | Description |
|---|---|---|
| 1 | pitcher_name | 투수 이름 |
| 2 | year | 년도 |
| 3 | ERA | 평균자책점 |
| 4 | WAR | 대체선수대비승리기여도 |
| 5 | TBF | 상대한 타자수(타석수) |
| 6 | H | 피안타수 |
| 7 | HR | 피홈런수 |
| 8 | BB | 피볼넷수 |
| 9 | HBP | 피사구수 |
| 10 | SO | 삼진수 |

| | Column_name | Description |
|---|---|---|
| 11 | WHIP | 이닝당 출루 허용률 |
| 12 | BABIP | 인플레이 타구 안타 비율 |
| 13 | FIP | 수비 무관 자책점 |
| 14 | LD% | 라인드라이브비율 |
| 15 | GB% | 땅볼 비율 |
| 16 | FB% | 플라이볼 비율 |
| 17 | IFFB% | 플라이볼 중 인필드 플라이볼 비율 |
| 18 | SwStr% | 헛스윙 비율 |
| 19 | Swing% | 스윙 비율 |

# 3. baseball_savant_foreigners_2011_2018

- 외국인 투수들의 KBO 입성 전,
  메이저리그에서의 스탯캐스트 데이터

| | Column_name | Description |
|---|---|---|
| 1 | game_date | 게임 날짜 |
| 2 | release_speed | 구속 |
| 3 | batter | 타자의 고유 Id |
| 4 | pitcher | 투수의 고유 Id |
| 5 | events | 해당 타석의 결과 |
| 6 | description | 해당 공의 결과 |
| 7 | zone | 공이 홈플레이트를 지날 때의 위치 |
| 8 | stand | 타자의 손잡이 |
| 9 | p_throws | 투수의 손잡이 |
| 10 | bb_type | 타구의 유형 |
| 11 | balls | 공을 던지기 직전 볼카운트 중 볼수 |
| 12 | strikes | 공을 던지기 직전 볼카운트 중 스트라이크수 |

| | Column_name | Description |
|---|---|---|
| 13 | pfx_x | 공의 수평 움직임 |
| 14 | pfx_z | 공의 수직 움직임 |
| 15 | plate_x | 공이 홈플레이트를 지날 때의 수평 위치 |
| 16 | plate_z | 공이 홈플레이트를 지날 때의 수직 위치 |
| 17 | ax | 공의 가속도의 x성분 |
| 18 | ay | 공의 가속도의 y성분 |
| 19 | az | 공의 가속도의 z성분 |
| 20 | launch_speed | 타구의 속도 |
| 21 | launch_angle | 타구의 발사각도 |
| 22 | spin_rate | 투수가 던진 공의 회전율 |
| 23 | pitch_name | 구종 |
| 24 | pitcher_name | 투수 이름 |

## 4. fangraphs_foreigners_2019

- **2019**년 신규 외국인 투수들의
  메이저리그에서의 역대 정규시즌 성적
  **(fangraphs_foreigners_2011_2018**와 컬럼 동일**)**

## 5. baseball_savant_foreigners_2019

- **2019**년 신규 외국인 투수들의
  메이저리그에서의 스탯캐스트 데이터
  **(baseball_savant_foreigners_2011_2018**와 컬럼 동일**)**

# KBO Scouting Challenge
## 3. Preprocessing

### 1. Labelling
(KBO 역대 성적 2011-2018)

| pitcher_name | Label |
|---|---|
| past | ... |

MERGE

### 2. Train Data
(MLB 역대 성적 & 스탯캐스트 2011-2018)

| pitcher_name | feature 1 | ..... feature n |
|---|---|---|
| past | ... | ... |

Model Train

### 3. Test Data
(MLB 역대 성적 & 스탯캐스트 2019)

| pitcher_name | feature 1 | ..... feature n |
|---|---|---|
| New | ... | ... |

Predict

# 1. Labeling

## Q. 누가 성공한 선수인가?

| | pitcher_name | year | team | ERA | TBF | H | HR | BB | HBP | SO | year_born |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 니퍼트 | 2011 | 두산 | 2.55 | 763 | 150 | 8 | 64 | 10 | 150 | |
| 2 | 니퍼트 | 2012 | 두산 | 3.20 | 785 | 156 | 15 | 68 | 8 | 126 | |
| 3 | 니퍼트 | 2013 | 두산 | 3.58 | 482 | 108 | 7 | 34 | 4 | 104 | |
| 4 | 니퍼트 | 2014 | 두산 | 3.81 | 760 | 186 | 17 | 48 | 6 | 158 | |
| 5 | 니퍼트 | 2015 | 두산 | 5.10 | 404 | 104 | 4 | 33 | 4 | 76 | |
| 6 | 니퍼트 | 2016 | 두산 | 2.95 | 701 | 151 | 15 | 57 | 9 | 142 | |
| 7 | 니퍼트 | 2017 | 두산 | 4.06 | 782 | 175 | 20 | 77 | 10 | 161 | |
| 8 | 니퍼트 | 2018 | KT | 4.25 | 765 | 209 | 26 | 39 | 9 | 165 | |
| 9 | 다이아몬드 | 2017 | SK | 4.42 | 581 | 163 | 11 | 35 | 9 | 59 | |
| 10 | 듀브론트 | 2018 | 롯데 | 4.92 | 629 | 162 | 13 | 62 | 8 | 109 | |

→ 신규 선수들의 KBO 데뷔 성공을 예측
기존 선수들의 KBO 데뷔년도 성적 추출

# 1. Labeling

**Q.** 누가 성공한 선수인가?

**A.** 1. 낮은 **ERA** (평균자책점)

2. 낮은 **TBF / IP** (이닝당 상대한 타자 수)
   - **KBO** 기록실 선수별 **IP**(던진 이닝 수) 조사

3. 낮은 **FIP** (수비 무관 자책점)
   - **FIP** = (13*HR + 3*(BB+HBP) -2*K)/IP + C(상수)

4. ~~**KBO** 활동연수~~ → 재계약 여부

# 1. Labeling

## Q. 누가 성공한 선수인가?

## A. 62명의 투수 성적 → 군집화 (K-Means)

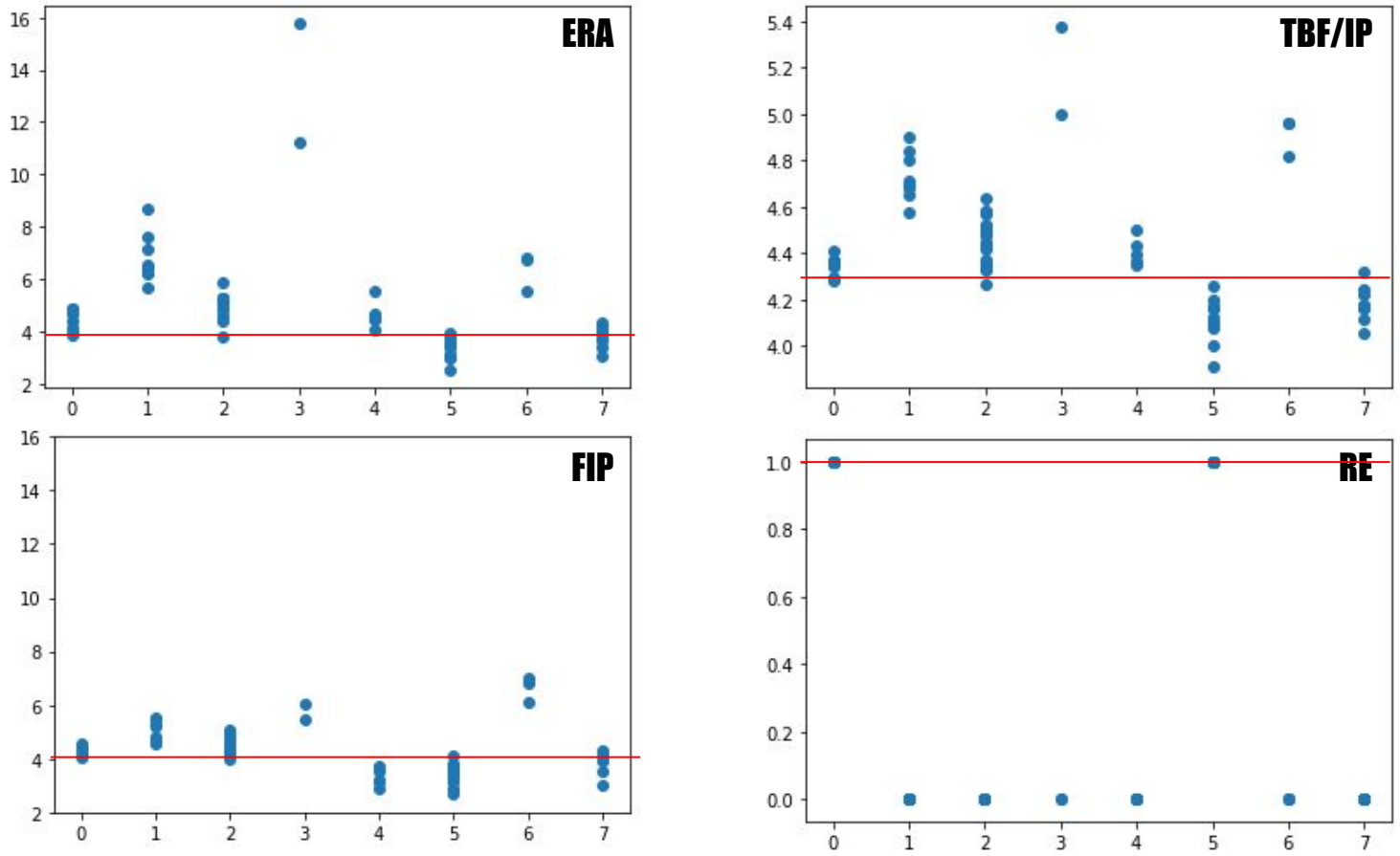
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 62 entries, 0 to 61
Data columns (total 9 columns):
pitcher_name    62 non-null object
years           62 non-null int64
year            62 non-null object
team            62 non-null object
ERA             62 non-null float64
IP              62 non-null float64
FIP             62 non-null float64
re              62 non-null int64
TBF/IP          62 non-null object
dtypes: float64(3), int64(2), object(4)
memory usage: 4.8+ KB
```

```
# 표준화
from sklearn.preprocessing import scale
scale(df_train)
# KMeans
from sklearn.cluster import KMeans
model = KMeans(n_clusters=8)
model.fit(scale(df_train))
model.cluster_centers_
df['label'] = model.labels_
df
```

# 1. Labeling

## Q. 누가 성공한 선수인가?

## A. 5번 그룹 : Best / 0,7 그룹 : Good



# 2. Train

## Q. 어떤 Feature 들을 학습시킬 것인가?

## A. MLB 역대 성적

```
RangeIndex: 205 entries, 0 to 204
Data columns (total 19 columns):
pitcher_name    205 non-null object
year            205 non-null float64
ERA             205 non-null float64
WAR             205 non-null float64
TBF             205 non-null float64
H               205 non-null float64
HR              205 non-null float64
BB              205 non-null float64
HBP             205 non-null float64
SO              205 non-null float64
WHIP            205 non-null float64
BABIP           205 non-null float64
FIP             205 non-null float64
LD%             205 non-null float64
GB%             205 non-null float64
FB%             205 non-null float64
IFFB%           205 non-null float64
SwStr%          205 non-null float64
Swing%          205 non-null float64
```

어떤 **Feature**가 좋은지....
→ 일단 다 모델에 학습시켜보자....

단,
**KBO** 데뷔 이전의 성적만
한 투수의 역대 성적 평균을 가지고

# 2. Train

## Q. 어떤 Feature 들을 학습시킬 것인가?

## A. MLB 스탯캐스트

```
RangeIndex: 135753 entries, 0 to 135752
Data columns (total 24 columns):
game_date            135753 non-null object
release_speed        135534 non-null float64
batter               135684 non-null float64
pitcher              135753 non-null int64
events               35707 non-null object
description          135753 non-null object
zone                 135534 non-null float64
stand                135753 non-null object
p_throws             135753 non-null object
bb_type              26575 non-null object
balls                135753 non-null int64
strikes              135753 non-null int64
pfx_x                135534 non-null float64
pfx_z                135534 non-null float64
plate_x              135534 non-null float64
plate_z              135534 non-null float64
ax                   135534 non-null float64
ay                   135534 non-null float64
az                   135534 non-null float64
launch_speed         6849 non-null float64
launch_angle         6850 non-null float64
release_spin_rate    24604 non-null float64
pitch_name           135423 non-null object
pitcher_name         135753 non-null object
```

1. 선수 별 구종 수
2. 선수 별 **Max_speed**
3. 선수 별 **Min_speed**

# 2. Train

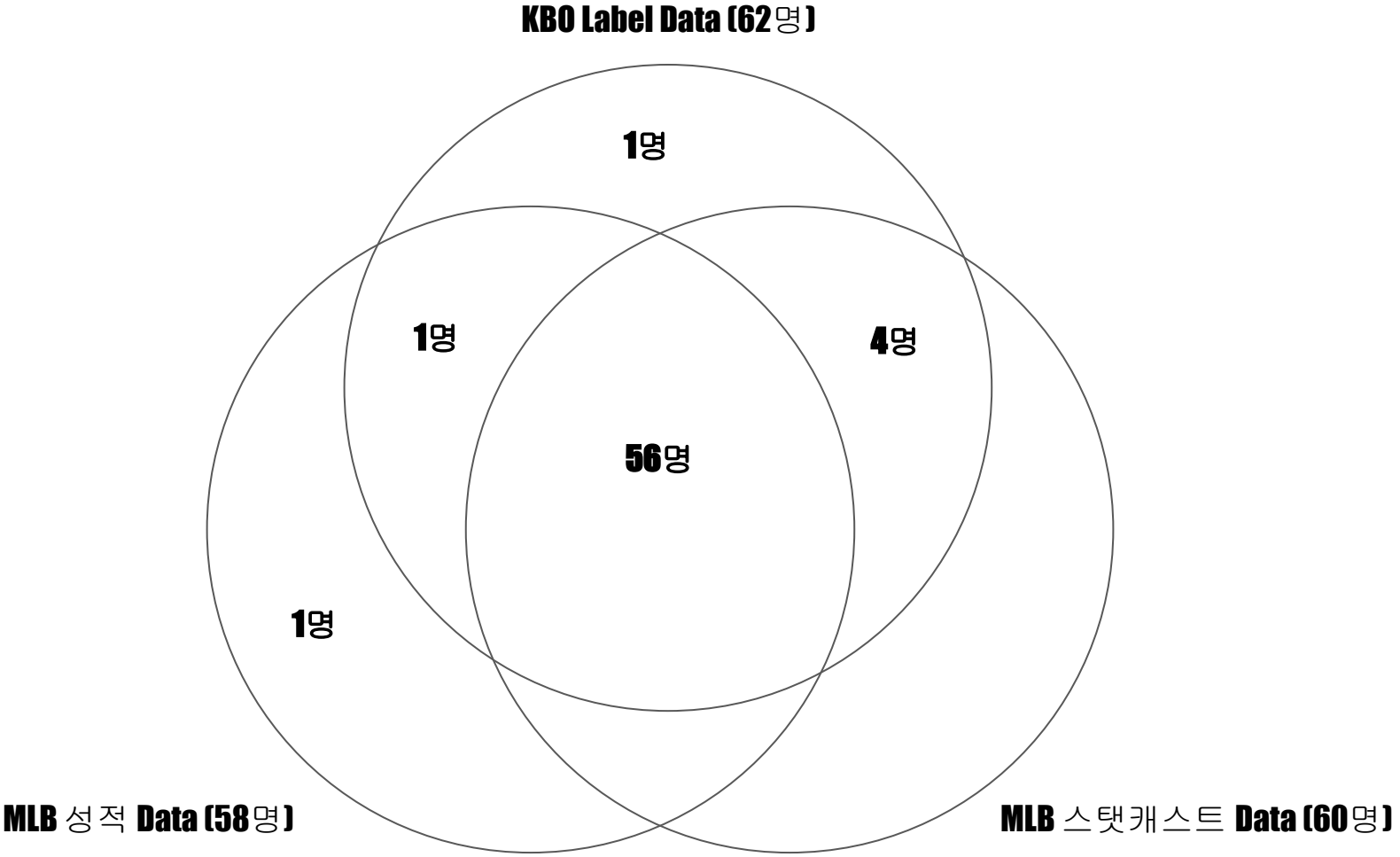
## Q. 어떤 Feature 들을 학습시킬 것인가?

```
#구종 수 뽑기
bsf_11['pitch_name'].unique()
# nan, Unknown, Intentional Ball, Pitch Out, Fastball etc로 변경 후 제거
bsf_11.loc[bsf_11['pitch_name'].isnull(),'pitch_name']
bsf_11.loc[bsf_11['pitch_name'].isnull(),'pitch_name'] = 'etc'
bsf_11['pitch_name'].unique()
bsf_11.loc[bsf_11['pitch_name'].isin(['Unknown', 'Intentional Ball', 'Pitch Out', 'Fastball']),'pitch_name']
bsf_11.loc[bsf_11['pitch_name'].isin(['Unknown', 'Intentional Ball', 'Pitch Out', 'Fastball']),'pitch_name'] = 'etc'
bsf_11['pitch_name'].unique()
bsf_11 = bsf_11[bsf_11['pitch_name'] != 'etc']

pitch_uni = bsf_11.groupby('pitcher_name')['pitch_name'].unique()

sav_11 = pd.DataFrame()
for i in range(len(pitch_uni)):
    temp = pd.DataFrame({'pitcher_name':[pitch_uni.index[i]],'pitch_cnt':[len(pitch_uni[i])]})
    sav_11 = sav_11.append(temp, ignore_index=True)
sav_11

#최고 구속 뽑기
bsf_11.groupby(['pitcher_name','pitch_name'])['release_speed'].max()
bsf_11.info()
r_Mspeed = bsf_11.groupby('pitcher_name')['release_speed'].max()
sav_11 = pd.merge(sav_11, r_Mspeed, left_on='pitcher_name', right_index=True)
sav_11.info()
sav_11 = sav_11.rename({'release_speed':'Max_speed'}, axis=1)
sav_11.info()
#최저 구속 뽑기
bsf_11.groupby(['pitcher_name','pitch_name'])['release_speed'].min()
r_mspeed = bsf_11.groupby('pitcher_name')['release_speed'].min()
sav_11 = pd.merge(sav_11, r_mspeed, left_on='pitcher_name', right_index=True)
sav_11.info()
sav_11 = sav_11.rename({'release_speed':'Min_speed'}, axis=1)
sav_11.info()
sav_11
```

```
Data columns (total 4 columns):
pitcher_name    60 non-null object
pitch_cnt       60 non-null int64
Max_speed       60 non-null float64
Min_speed       60 non-null float64
```

# 2. Train

## # Merge (on='pitcher_name')



# 3. Test

## #13명의 투수 성적 & 스탯캐스트 정제

## → Train 정제와 동일하게 진행

**Train**

```
Data columns (total 17 columns):
pitcher_name    56 non-null object
label           56 non-null int32
KBOFIP          56 non-null float64
ERA             56 non-null float64
WAR             56 non-null float64
WHIP            56 non-null float64
BABIP           56 non-null float64
FIP             56 non-null float64
LD%             56 non-null float64
GB%             56 non-null float64
FB%             56 non-null float64
IFFB%           56 non-null float64
SwStr%          56 non-null float64
Swing%          56 non-null float64
pitch_cnt       56 non-null int64
Max_speed       56 non-null float64
Min_speed       56 non-null float64
```

**Test**

```
Data columns (total 14 columns):
ERA             13 non-null float64
WAR             13 non-null float64
WHIP            13 non-null float64
BABIP           13 non-null float64
FIP             13 non-null float64
LD%             13 non-null float64
GB%             13 non-null float64
FB%             13 non-null float64
IFFB%           13 non-null float64
SwStr%          13 non-null float64
Swing%          13 non-null float64
pitch_cnt       13 non-null int64
Max_speed       13 non-null float64
Min_speed       13 non-null float64
```

# KBO Scouting Challenge
## 4. Model Train

**Q.** 어떤 모델을 이용할 것인가?

**A.** 1. Decision Tree
2. Random Forest
3. Logistic Regression
4. K Nearest Neighbor

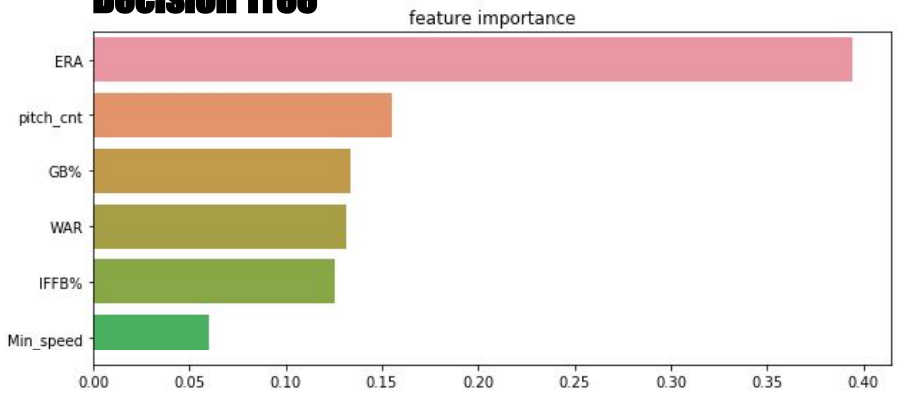
} 'Label' Classify

5. Linear Regression } 'KBO_FIP' Predict

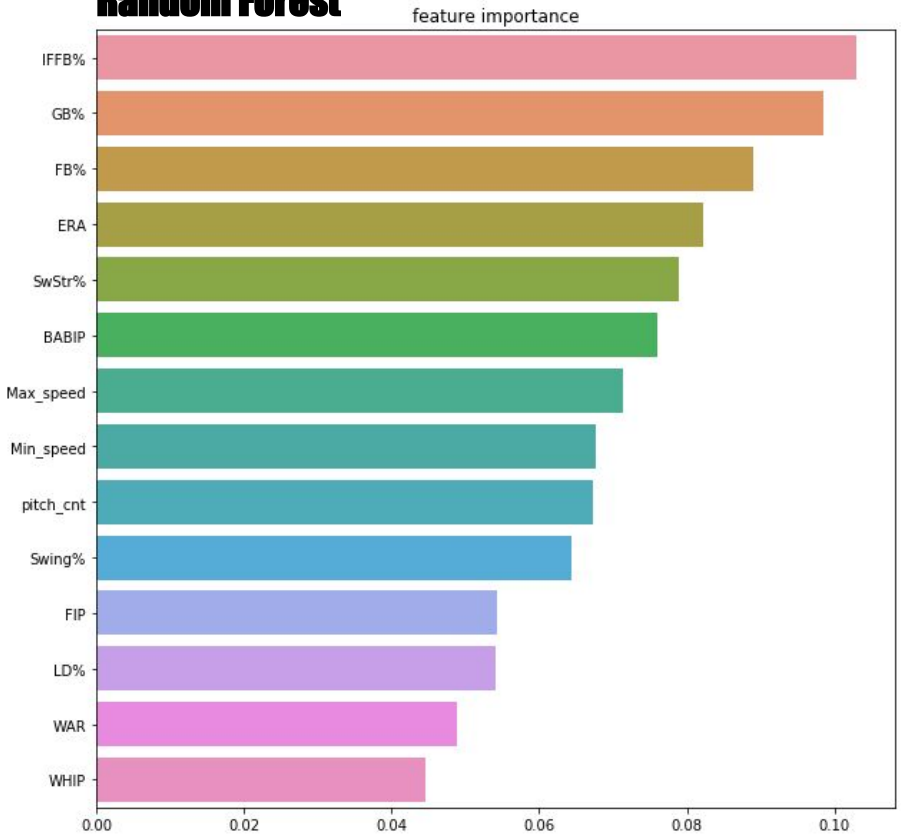
# KBO Scouting Challenge
# 4. Model Train

```
# Decision Tree
from sklearn.tree import DecisionTreeClassifier
model1 = DecisionTreeClassifier(max_depth=5)
model1.fit(x,y)
model1.score(x,y)

# random forest
from sklearn.ensemble import RandomForestClassifier
model2 = RandomForestClassifier(n_estimators=100, oob_score=True)
model2.fit(x,y)
model2.score(x,y)

#로지스틱
from sklearn.linear_model import LogisticRegression
model3 = LogisticRegression()
model3.fit(x, y)
model3.score(x, y)

#knn
from sklearn.neighbors import KNeighborsClassifier
model4 = KNeighborsClassifier(n_neighbors=3)
model4.fit(x, y)
model4.score(x, y)

#회귀분석
from sklearn.linear_model import LinearRegression
model5 = LinearRegression()
model5.fit(x, y1)
model5.score(x, y1)
```

## Decision Tree




## Random Forest



# KBO Scouting Challenge
## 5. Classify / Predict

# # Result & Score

| | pitcher | DT | RF | LG | KNN | pre_FIP |
|---|---|---|---|---|---|---|
| 1 | 루친스키 | bad | bad | bad | good | 3.854776 |
| 2 | 맥과이어 | bad | bad | bad | bad | 3.996178 |
| 3 | 버틀러 | bad | bad | bad | bad | 4.427409 |
| 4 | 서폴드 | good | bad | bad | good | 3.993384 |
| 5 | 알칸타라 | bad | bad | bad | bad | 4.815770 |
| 6 | 요키시 | bad | bad | bad | best | 3.796188 |
| 7 | 윌랜드 | good | bad | bad | bad | 4.853554 |
| 8 | 채드벨 | good | bad | bad | bad | 3.884712 |
| 9 | 켈리 | bad | bad | bad | good | 4.413401 |
| 10 | 쿠에바스 | best | bad | bad | good | 3.323674 |
| 11 | 터너 | good | bad | good | bad | 4.569228 |
| 12 | 톰슨 | bad | bad | bad | good | 4.057246 |
| 13 | 헤일리 | good | bad | bad | bad | 4.698123 |

| | pitcher | DT | RF | LG | KNN | ling | sum |
|---|---|---|---|---|---|---|---|
| 1 | 루친스키 | 0 | 0 | 0 | 1 | 2 | 3 |
| 2 | 맥과이어 | 0 | 0 | 0 | 0 | 1 | 1 |
| 3 | 버틀러 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 서폴드 | 1 | 0 | 0 | 1 | 1 | 3 |
| 5 | 알칸타라 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 요키시 | 0 | 0 | 0 | 2 | 2 | 4 |
| 7 | 윌랜드 | 1 | 0 | 0 | 0 | 0 | 1 |
| 8 | 채드벨 | 1 | 0 | 0 | 0 | 1 | 2 |
| 9 | 켈리 | 0 | 0 | 0 | 1 | 1 | 2 |
| 10 | 쿠에바스 | 2 | 0 | 0 | 1 | 2 | 5 |
| 11 | 터너 | 1 | 0 | 1 | 0 | 0 | 2 |
| 12 | 톰슨 | 0 | 0 | 0 | 1 | 1 | 2 |
| 13 | 헤일리 | 1 | 0 | 0 | 0 | 0 | 1 |

**best 2점 / good 1점 / bad 0점**
**pre_FIP 3등까지 2점 / 8등까지 1점 / 나머지 0점**

# 올 시즌 예측 선수 성적

- 10승이상의 신규 외국인 투수 성적

| 선수명 | 팀명 | ERA | G | W | L | SV | HLD | WPCT | IP | H | HR | BB | HBP | SO | R | ER | WHIP |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 켈리 | LG | 2.55 | 29 | 14 | 12 | 0 | 0 | 0.538 | 180 1/3 | 164 | 7 | 41 | 16 | 126 | 70 | 51 | 1.14 |
| 요키시 | 키움 | 3.13 | 30 | 13 | 9 | 0 | 0 | 0.591 | 181 1/3 | 166 | 9 | 39 | 11 | 141 | 72 | 63 | 1.13 |
| 쿠에바스 | KT | 3.62 | 30 | 13 | 10 | 0 | 0 | 0.565 | 184 | 153 | 18 | 63 | 12 | 135 | 80 | 74 | 1.17 |
| 서폴드 | 한화 | 3.51 | 31 | 12 | 11 | 0 | 0 | 0.522 | 192 1/3 | 191 | 8 | 54 | 8 | 135 | 84 | 75 | 1.27 |
| 채드벨 | 한화 | 3.50 | 29 | 11 | 10 | 0 | 0 | 0.524 | 177 1/3 | 169 | 14 | 63 | 10 | 134 | 73 | 69 | 1.31 |
| 알칸타라 | KT | 4.01 | 27 | 11 | 11 | 0 | 0 | 0.500 | 172 2/3 | 189 | 15 | 27 | 8 | 100 | 80 | 77 | 1.25 |

# 예측한 두 선수의 승수(W)는 13승
# 승률(WPCT)은 신규 외국인 선수들 TOP 1, 2

## Domain Knowledge

## Materialize Problem

## More Data

# KBO Scouting Challenge
## 7. Reference

### DACON
**Mission 7: KBO** 외국인 투수 스카우팅 대회**!**
**- https://dacon.io/cpt7**

### KBO
**- https://www.koreabaseball.com**

### 야구공작소 위키
**- http://ko.yagongso.wikidok.net**

# Q&A