1. **단계적 변수 선택 (Stepwise Variable Selection)**: 이 방법은 다중 선형 회귀와 같은 모델에서 변수를 선택하는 데 사용됩니다. 순차적으로 변수를 추가하거나 제거하여 모델의 성능을 향상시키는 데 도움이 됩니다.
2. **상관 분석 (Correlation Analysis)**: 변수 간의 상관 관계를 분석하여 중요한 변수를 선택하는 데 사용됩니다. 높은 상관 관계를 가지는 변수는 중복 정보를 제공할 수 있으므로 중요한 변수만 선택할 수 있습니다.
3. **피처 중요도 분석 (Feature Importance Analysis)**: 머신 러닝 모델을 사용할 때, 모델의 피처 중요도를 확인하여 어떤 변수가 예측에 가장 중요한 역할을 하는지 파악할 수 있습니다. 랜덤 포레스트나 그라디언트 부스팅과 같은 앙상블 모델은 피처 중요도를 추정하는 데 사용됩니다.
4. **L1 규제 (L1 Regularization)**: L1 규제를 사용하는 선형 모델은 가중치가 0에 가까운 변수를 자동으로 선택하게 됩니다. 이를 통해 불필요한 변수를 제거할 수 있습니다.
5. **PCA (주성분 분석, Principal Component Analysis)**: 다차원 데이터를 주성분으로 변환하여 변수의 수를 줄이는 방법입니다. 주성분은 데이터의 분산을 최대화하는 방향으로 선택되며, 주성분만을 사용하여 데이터를 설명하는 데 사용됩니다.
6. **재귀적 변수 제거 (Recursive Feature Elimination, RFE)**: 이 방법은 모델을 학습하고 변수 중요도를 평가한 후, 중요도가 낮은 변수를 반복적으로 제거하는 방식으로 변수를 선택합니다.
7. **도메인 지식 (Domain Knowledge)**: 분야 전문가의 도움을 받아 중요한 변수를 선택하는 방법입니다. 도메인 지식을 활용하면 모델에게 중요한 변수를 미리 알려줄 수 있습니다.

Mlb savant 칼럼명 모음집

<https://ibaseballgen.tistory.com/701>

**변수선택법**

* 변수선택법은 최적의 회귀방정식 선택에 도움을 주는 방법론이다
* 변수선택법에는 전진선택법, 후진제거법, 단계선택법이 있다.
* 파이썬에서는 OLS결과를 보면서 수작업으로 변수를 조정해야하지만, R에서는 step()라는 함수를 활용하여 변수선택법을 쉽게 할 수 있다.

<https://ysyblog.tistory.com/121>

PFx에 관한 설명

https://kini.kr/602

칼럼설명 :

스탯캐스트 데이터에서 투구에 대한 정보로 x, y, z 축에 대한 속도(vx0, vy0, vz0) , 가속도(ax, ay, az), 실제로 포수에게 도달하는 지점과 공기저항이 없을 때 이론적으로 포수에게 도달하는 지점의 차이인 pfx값을 제공.

PFX 자료가 있으면 릴리스포인트를 떠난 이후 공의 움직임을 상하좌우로 나누어 그래프로 그릴 수 있다. 아래 그래프는 오른손 투수가 던진 각 구종이 일반적으로 나타나는 지점을 보여준다.

텍스트, 도표, 라인, 폰트이(가) 표시된 사진

자동 생성된 설명

PFX는 구종만 보여주는 것이 아니다. 투구의 로케이션 역시 PFX를 활용하면 실제와 거의 근접하게 표시할 수 있다. 그것도 좌우 타자를 상대로 각각 구할 수 있다.

PFX 데이터는 이외에도 투구 궤적, 구속 등 투구 자체에 관한 '실제적'인 데이터를 우리에게 제공한다. 원한다면 초속과 종속의 차이를 알아볼 수도 있고 공의 회전수 역시 구할 수 있다.  
  
단순히 기록을 가지고 노는 '숫자 놀음'을 벗어나 PFX는 세이버메트릭스를 또 한 단계 업그레이드시키고 있다.



텍스트, 스크린샷, 번호, 폰트이(가) 표시된 사진

자동 생성된 설명

텍스트, 폰트, 화이트, 영수증이(가) 표시된 사진

자동 생성된 설명

<https://issuu.com/randoll/docs/pfx>

"at\_bat\_number"는 야구에서 한 선수가 한 경기 동안의 타석 번호를 나타내는 변수입니다. 타석 번호는 선수가 해당 경기 동안 타석에 들어간 순서를 나타냅니다. 첫 번째 타석은 1번, 두 번째 타석은 2번, 그리고 이어지는 타석마다 번호가 증가합니다.

"pitch\_number"는 한 타석 동안의 투구 횟수를 나타내는 변수입니다. 타자가 공을 쳐서 안타 또는 아웃이 되거나 볼넷을 받을 때까지 투수가 공을 던지는 횟수를 나타냅니다. 따라서 "pitch\_number"가 증가함에 따라 한 타석 동안 투수와 타자 간의 투구가 이루어집니다.

이 두 변수는 야구 경기 데이터를 분석할 때 타자의 타석별 성적과 투수의 투구 횟수를 파악하는 데 사용될 수 있습니다.

"delta\_home\_win\_exp"는 한 타석 동안의 야구 게임에서 홈 팀의 승률 기대치(Win Expectancy)의 변화를 나타내는 변수입니다. 이 변수는 한 타석이 끝난 후의 승률 기대치와 그 타석이 시작되기 전의 승률 기대치 간의 변화를 측정합니다. 승률 기대치는 특정 시점에서 팀이 게임을 얻을 확률을 나타내며, 경기 진행 상황에 따라 변합니다. 따라서 "delta\_home\_win\_exp"는 타석 전후의 게임 상황 변화를 추적하는 데 사용될 수 있습니다.

"delta\_run\_exp"는 한 투구 동안의 야구 게임에서 득점 기대치(Run Expectancy)의 변화를 나타내는 변수입니다. 이 변수는 한 투구가 끝난 후의 득점 기대치와 그 투구가 시작되기 전의 득점 기대치 간의 변화를 측정합니다. 득점 기대치는 특정 상황에서 팀이 득점할 확률을 나타내며, 경기 상황에 따라 변동합니다. "delta\_run\_exp"는 한 투구마다 득점 상황의 변화를 추적하는 데 사용될 수 있습니다.

이러한 변수들은 야구 데이터 분석에서 경기 상황의 변화와 선수나 팀의 성적을 평가하거나 비교하는 데 사용됩니다. 예를 들어, 특정 타자의 타석 전후의 "delta\_home\_win\_exp"와 "delta\_run\_exp"를 분석하면 그 타자가 팀의 승률과 득점에 어떤 영향을 미치는지를 파악할 수 있습니다.

텍스트, 영수증, 패턴이(가) 표시된 사진

자동 생성된 설명

참고하기 좋은 논문.

<https://www.dbpia.co.kr/pdf/pdfView.do?nodeId=NODE11199476>

로지스틱을 활용한 변수선택법

Stepwise, backward, forward

모든 변수를 돌렸을때 오버피팅 이슈 , 다중공선성 문제가 생김.

예측력이 0.99

그래서 domain knowledge 조사를 통해, 다중공선성과 데이터 분석 목적에 필요없는 변수 제거.

남은 변수들 :

pitch\_type : 명목변수

release\_speed : 연속변수

release\_pos\_x : **연속변수**

release\_pos\_z : **연속변수**

events : 명목변수 (1,0)

zone : 명목변수

stand : 명목변수

p\_throws : 명목변수

balls : 이산변수

strikes : 이산변수

pfx\_x : 연속변수

pfx\_z : 연속변수

plate\_x : 연속변수

plate\_z : 연속변수

outs\_when\_up : 이산형 변수

inning : 이산형 변수

release\_spin\_rate : 연속변수

release\_extension : 연속변수

release\_pos\_y : 연속변수

at\_bat\_number : 이산형 변수

pitch\_number : 이산형 변수

spin\_axis : 연속변수

그래서 일단 선택된 컬럼들로 모델 돌렸을때,

Accuracy: 0.7578979995708257

Precision: 0.8459411634594116

Recall: 0.2879460924559554

F1 Score: 0.4296466887603213

데이터 불균형이 있어서 smote 후 모델을 다시 돌림.

정확도 6% 상승

Precision 11% 정도 상승.

Accuracy: 0.8149848963698905

Precision: 0.9539794397716653

Recall: 0.6610021176429409

F1 Score: 0.7809159516179055

로지스틱 회귀분석으로 변수 선택후,

20개 변수로 줄어들음. 로지스틱 회귀분석 중 가장 좋았던 Stepwise를 활용해서 모델을 돌린결과,

2%, 5% 하강. Xgb. Light gbm

Accuracy: 0.7942850657400778

Precision: 0.9095233448797385

Recall: 0.6525315458793468

F1 Score: 0.7598870920077854

하이퍼 파라미터 튜닝 시(randomserch\_cv기준) 2% 정도 상승, 사실 하이퍼파리미터 조절에 대한 감이 아직 잘모르겠음.

새로운 방법론 :

Cnn 딥러닝 이용시 모델결과

Accuracy: 0.7126468893506313

Precision: 0.7208738306679553

Recall: 0.6918567003272721

F1 Score: 0.7060672632124168

생각보다 딥러닝이 효과가 없음