

Day47; 20221111

날짜	@2022년 11월 11일
유형	@2022년 11월 11일
태그	

GitHub - u8yes/AI

You can't perform that action at this time. You signed in with another tab or window. You signed out in another tab or window. Reload to refresh your session. Reload to refresh your session.

<https://github.com/u8yes/AI>

u8yes/AI



Contributor 1 Issues 0 Stars 0 Forks 0

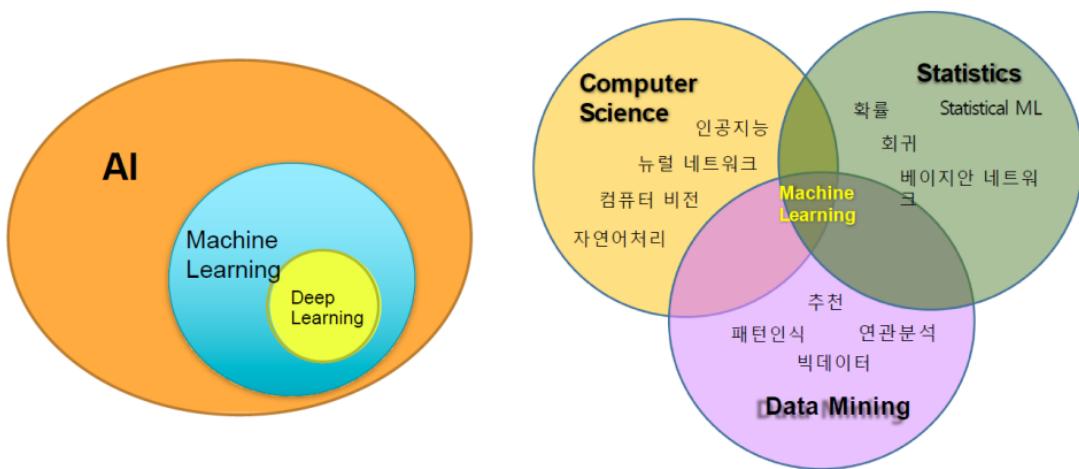
https://s3-us-west-2.amazonaws.com/secure.notion-static.com/721790e9-a573-43db-ba34-84873bdec998/02_%EB%A8%B8%EC%8B%A0%EB%9F%A%C%EB%8B%9D_%EC%95%8C%EA%B3%A0%EB%A6%AC%EC%A6%98.pdf

사이키린, 텐서플로우 배우게 됨

openCV도 배우게 됨.

머신러닝의 개념

인공지능 > 머신러닝 > 딥러닝



× 독립변수에 의해 영향을 받는 변수는 종속변수.

규칙성을 가지는 것이 함수.

× 정의역일 때 y치역이 나오는 것을 보고 규칙을 찾아야 함.

규칙을 찾기 위해 함수를 저장해서 컴퓨터 머신에 알려줘야 했지만

이제는 머신러닝이 알아서 찾아주게 설계됨.

머신러닝은 40년대부터 기본 이론이 시작됐음.

주어진 데이터를 보고 규칙성을 찾아서 어떤 임의의 x값에 대해서도 y값을 예측할 수 있게 하는 것이 회귀이론의 예측분석이다. (회귀분석)

이전에는 x값에 대한 질문을 사람이 던졌는데 이제는 그것 마저도 머신러닝이 해줌.

비지도 학습은 x값만 전달하는 것(정답은 없고 데이터만 전달), 규칙성을 머신러닝, 딥러닝이 찾음.

지도 학습은 x,y값, 정답을 전부 전달해주는 것.

비지도, 지도 무엇이 더 좋다 말할 수는 없다.

머신러닝, 딥러닝은 지도, 비지도 학습 전부다 사용.

BIG 데이터 분석 방법

기술통계분석			추론통계분석	데이터마이닝
■데이터 특성분석			■모집단을 표본으로 분석하여 추론	■변수의 관계와 패턴 분석으로 미래 예측
학력수준	실패	전략		
고졸 관찰빈도 기대빈도	40 36	49 54	Histogram for Parent height with Freedman-Diaconis 	Histogram for Parent Height with Sturge 
대학 관찰빈도 기대빈도	27 33	55 49	Histogram for Child height with Freedman-Diaconis 	Histogram for Child height with Sturge 
대학원졸 관찰빈도 기대빈도	23 21	31 32		
■ 합계, 평균, 빈도수, 비율, 표준편차, 분산, 교차분석 등 ■ 데이터의 특성 분석			■ 카이제곱검정, 비율 검정, 평균차이검정, 상관분석, 분산분석 ■ 집단간 차이 분석	■ 텍스트분석, 예측, 분류, 군집, 연관분석 ■ 패턴 및 규칙을 이용한 의사결정

추론통계분석 - 모집단을 표본으로 분석하여 추론

예측분석은 추론분석 영역에서 한 분야일뿐.

머신러닝의 개념

- 어플리케이션을 수정하지 않고도, 데이터를 기반으로 패턴을 학습하고 결과를 예측하는 알고리즘 기법을 통칭.
- 데이터를 기반으로 통계적인 신뢰도를 강화하고, 예측 오류를 최소화하기 위한 다양한 수학적 기법을 적용해 데이터 내의 패턴을 스스로 인지하고, 신뢰도 있는 예측 결과를 도출.
- 데이터 분석 영역은 재빠르게 머신러닝 기반의 예측분석(Predictive Analysis)으로 재편되고 있음.
- 많은 데이터 분석가와 데이터 과학자가 머신러닝 알고리즘 기반의 새로운 예측 모델을 이용해 더욱 정확한 예측 및 의사 결정을 도출하고 있으며, 데이터에 감춰진 새로운 의미와 통찰력(insight)를 발굴해 놀랄 만한 이익으로 연결시키고 있음.
- 귀납적 학습 : 데이터만 주어지더라도 구조를 추론하려고 시도하기 때문.



Insight



탐색적 자료 분석(Exploratory Data Analysis)

데이터의 특징과 내재하는 구조적인 관계를 알아내기 위한 분석 기법으로 이러한 자료의 탐색 과정을 통하여 얻은 정보를 기초로 통계모형을 세울 수 있음
미지의 특성을 파악하고 자료구조를 파악할 수 있는 증거 수집의 과정

Looking at data to see what it seems to say. It's concentrates on simple arithmetic and easy-to-draw picture. *John Tukey, 1977*



EDA

Optimization

기계학습(Machine Learning)

어떻게 하면 더 빨리 학습시키고 어떻게 하면 더 정확히 예측할 수 있을까에 대한 연구를 하며 데이터 전처리, 파생 변수 추가, 모형 선택 등의 방법을 통해서 예측 모형의 평가 점수를 높이는 과정

A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E. *Tom. Mitchell, 1997*



ML

데이터 탐색 및 시각화

의미 있는 정보는 무엇이 있을까?

- 데이터를 잘 다룰 수 있어야 함
- 데이터의 부분집합 추출 및 병합 등의 작업이 주를 이룸
- 데이터 시각화를 통해 탐색 결과를 이해하기 쉽도록 함

파이썬의 패키지

- ❖ Numpy & Pandas : 데이터를 다루기 위한 패키지
- ❖ Matplotlib & Seaborn : 데이터를 시각화 하기 위한 패키지

예측력이 높은 모형 생성

어떻게 하면 모형이 예측을 잘 할 수 있을까?

- 데이터 전처리, 파생변수 추가
 - 머신러닝 모형 생성 및 예측
 - 모형 평가

파이썬의 패키지

- ❖ Scikit-learn : 기계학습 라이브러리
- ❖ Statsmodels : 통계 라이브러리

연관분석을 하려면 반드시 2개 이상이 있어야 한다.

머신러닝 모델 개선

- 최적의 머신러닝 알고리즘과 모델 파라미터를 구축하는 능력도 중요하지만,
- 데이터를 이해하고 효율적으로 가공, 처리, 추출해서 최적의 데이터를 기반으로 알고리즘을 구동할 수 있도록 준비하는 능력이 무엇보다 더 중요.
- 다양하고 광대한 데이터를 기반으로 만들어진 머신러닝 모델은 더 좋은 품질을 약속.
- 앞으로 많은 회사의 경쟁력은 어떠한 품질의 데이터로 만든 머신러닝 모델이냐에 따라 결정.

데이터 분석결과에 대한 통찰력을 가지는 것이 아주 중요하다. 정확도를 높여주기 때문이다.

머신러닝의 분류

- 지도학습(Supervised Learning)
 - 회귀(Regression) : 단순 선형 회귀 / 다항 회귀
 - 분류(Classification) : 이진 분류(로지스틱 회귀) / 다중 분류
 - 추천 시스템
 - 시각/음성 감지/인지
 - 텍스트 분석, NLP
- 비지도학습(Un-supervised Learning)
 - 군집(cluster) 분석 : ex) k 평균 알고리즘
 - 연관(association) 분석
 - 차원 축소 : ex) 주성분 분석

비지도 연차군, 지도 회분추시텍

회귀는 모든 연속적인 값을 가지는 것을 가지고 결과를 예측.

분류는 뉴스 분류별 데이터를 넣고 어떤 분류인지 보는 것. 분류 모델에서 혼동행렬(Confusion Matrix) 사용. 혼동행렬이란 특정 분류 모델의 성능을 평가하는 지표.

- 예측하는 것이 컬럼으로 와야 한다. 그래서 혼동행렬 예측 클래스가 열로 돼있는 것이다.

머신러닝 모델의 성능 평가 - 혼동행렬(Confusion Matrix)

- 혼동행렬이란 특정 분류 모델의 성능을 평가하는 지표로, 실제값과 모델이 예측한 예측값을 한 눈에 알아볼 수 있게 배열한 행렬.

		예측 클래스	
		N	Y
실제 클래스	N	True/Negative 실제 N. 예측 Y	False/Positive 실제 N, 예측 Y
	Y	False/Negative 실제 Y. 예측 Y	True/Positive 실제 Y, 예측 Y

➤ 혼동행렬의 구성 요소

- True Negative(TN): 실제 값도 거짓이고 모델의 예측 값도 거짓인 경우.
- False Positive (FP): 실제 값은 거짓이나 모델의 예측 값이 참인 경우.
- False Negative(FN): 실제 값은 참이나 모델의 예측 값이 거짓인 경우.
- True Positive(TP): 실제 값이 참이고 모델의 예측 값도 참인 경우.

맞추려고 하는 대상이 Positive, 다른 대상이 Negative

머신러닝 모델의 성능 평가 - 혼동행렬 예

		Predicted Target	
		N(일반 고객)	Y(보험 사기자)
Actual Target	일반 고객 (Negative)	1613 True/Negative 실제 N. 예측 N	22 False/Positive 실제 N, 예측 Y
	보험 사기자 (Positive)	81 False/Negative 실제 Y. 예측 N	77 True/Positive 실제 Y, 예측 Y

- 실제 일반 고객을 일반 고객(Negative)으로 바르게 분류(True)한 고객의 수가 1613명
- 실제 일반 고객을 보험 사기자(Positive)로 틀리게 분류(False)한 고객의 수가 22명
- 실제 보험 사기자를 일반 고객(Negative)으로 틀리게 분류(False)한 고객의 수가 81명
- 실제 보험 사기자를 보험 사기자(Positive)로 바르게 분류(True)한 고객의 수가 77명

머신러닝 모델의 성능 평가 - 분류 모델 평가 지표

➤ 혼동행렬에 기반한 분류 모델 평가 지표

1. 정확도(Accuracy): 모델이 전체 중에서 예측을 올바르게 한 비율

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{FP} + \text{TN}}$$

2. 재현율(Recall): 실제 참인 값 중 모델이 참으로 예측한 값의 비율. 재현율이 낮을 경우 암인 사람에게 암이 아니라고 하였으니 심각한 결과를 초래할 것.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

3. 정밀도(Precision): 모델이 참으로 예측한 값 중 실제 참인 값의 비율. 정밀도가 낮을 경우 암이 아닌 사람에게 암이라고 했으니 불필요한 치료가 발생할 수 있음.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

4. F1 점수(F1-Score): 정밀도와 재현율의 조화 평균

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

validation ◎

[U] 확인, 비준

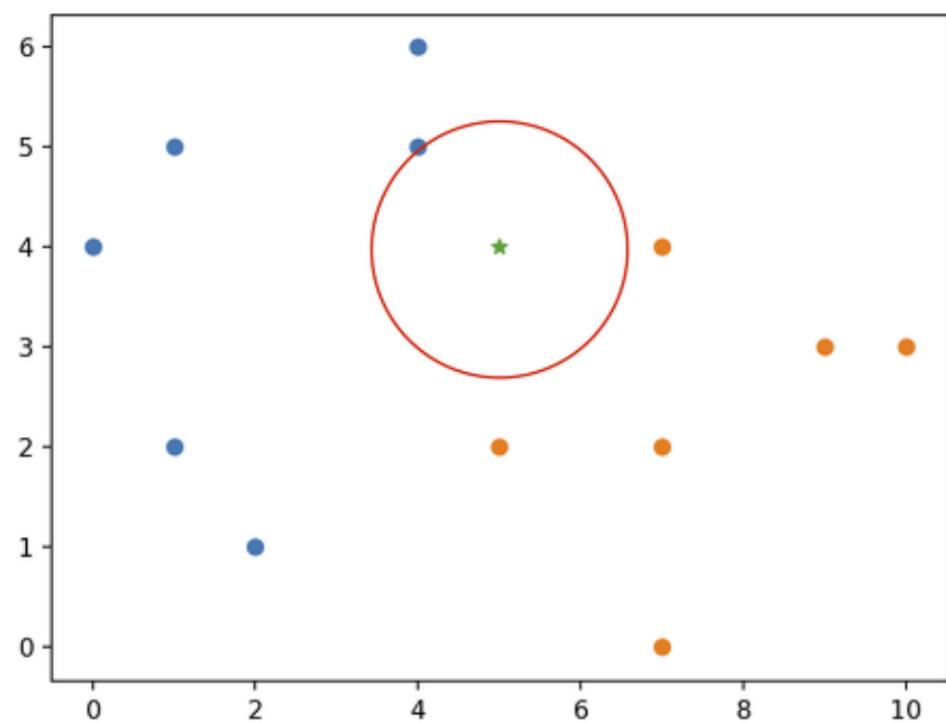
머신러닝은 sklearn을 가지고 살펴봄.

대표적 머신러닝 알고리즘

- k-최근접 이웃(k-Nearest Neighbor, kNN)
 - 서포트 벡터머신(Support Vector Machine, SVM)
 - 의사결정 트리(Decision Tree)
 - 나이브 베이즈(Naïve Bayes)
 - 앙상블(Ensemble)
 - 군집화(Clustering)
 - 주성분 분석(Principal Component Analysis, PCA)
 - 선형 회귀(Linear Regression)
 - 로지스틱 회귀(Logistic Regression)
 - 딥러닝(Deep Neural Network, DNN)
-

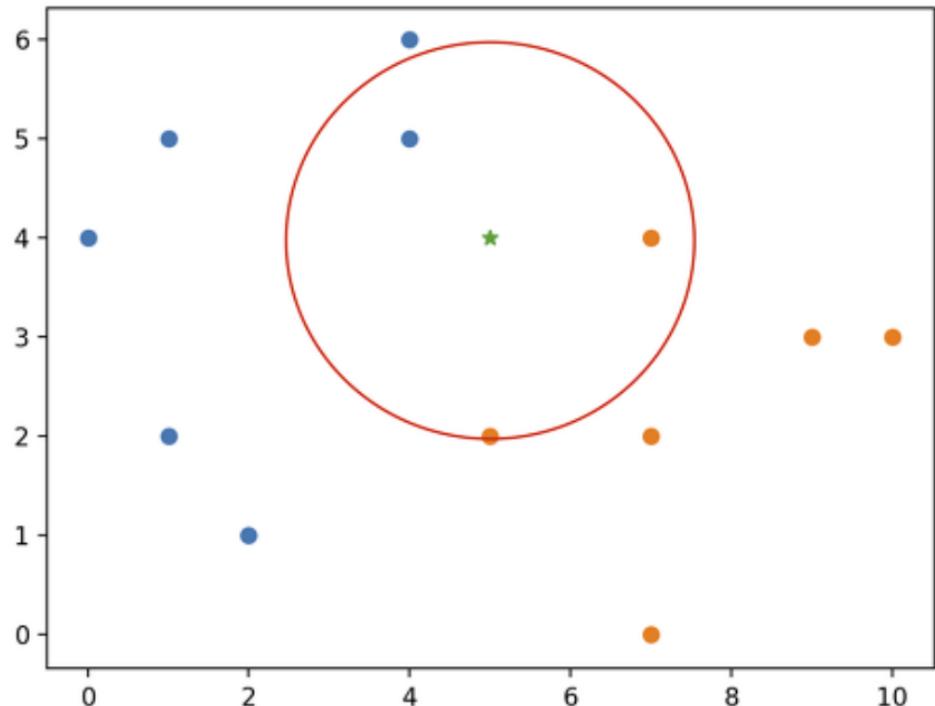
k-최근접 이웃(k-Nearest Neighbor, kNN)

- $k = 1$



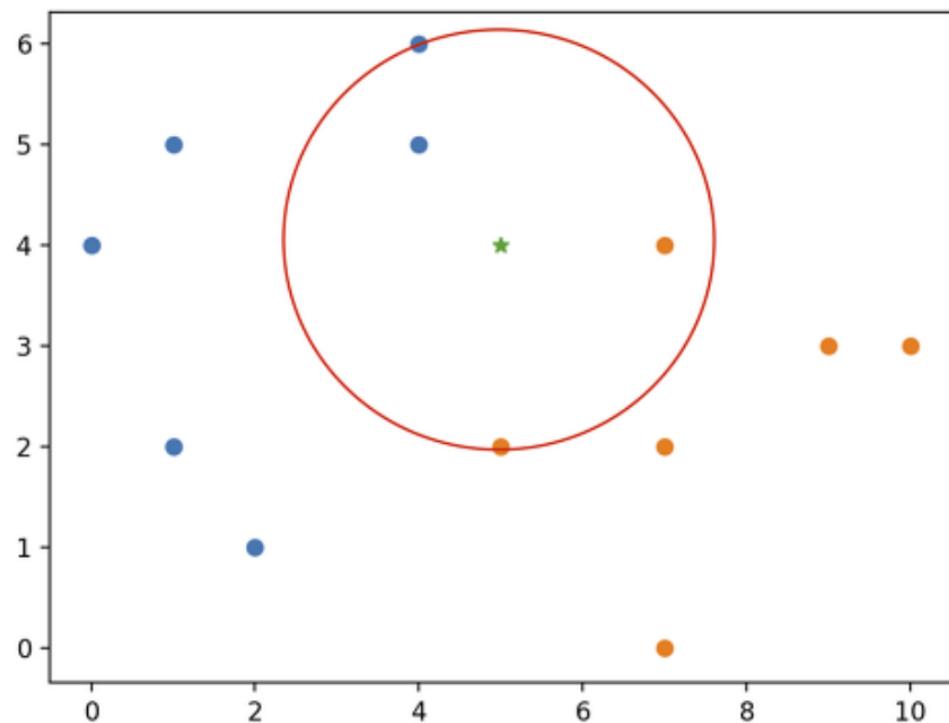
k-최근접 이웃(k-Nearest Neighbor, kNN)

- $k = 3$



k-최근접 이웃(k-Nearest Neighbor, kNN)

- $k = 4$



분류분석이다. 주황색일지, 파란색일지 선택하는 것이라서.

k 값을 홀수로 지정해야 유리하다. 짝수 기준으로 애매하다.

- kNN

하이퍼 파라미터 - k 값(이웃의 갯수)을 튜닝

학습이 필요없다.

거리 계산식이 유클리디안(내 기준 좌표 - 상대 기준좌표) 2 .

- kNN 단점

예측 속도가 느리다. - 가까운 거리를 다 계산하면서

- EDA

문제 정의가 프로젝트에서 중요하다.

예측만을 위해 생각하고 어떤 결과를 만들지는 생각 안 하면 위험하다.

라이브러리 임포트_20221111

- 실습에 필요한 라이브러리를 임포트

```
[6]: import numpy as np  
import pandas as pd
```

문제 정의

- 농구 선수의 경기 기록을 바탕으로, 그 선수의 포지션을 예측해보는 모델을 생성.

데이터 수집

```
10]: # 데이터를 수집  
df = pd.read_csv('data/csv/basketball_stat.csv')  
df.head()
```

```
10]:  
      Player  Pos   3P   2P   TRB   AST   STL   BLK  
0    Alex Abrines  SG  1.4  0.6  1.3  0.6  0.5  0.1  
1  Steven Adams  C  0.0  4.7  7.7  1.1  1.1  1.0  
2   Alexis Ajinca  C  0.0  2.3  4.5  0.3  0.5  0.6  
3  Chris Andersen  C  0.0  0.8  2.6  0.4  0.4  0.6  
4    Will Barton  SG  1.5  3.5  4.3  3.4  0.8  0.5
```

```
17]: # 현재 가지고 있는 데이터에서, 포지션의 갯수를 확인  
df.Pos.value_counts() # 시리즈의 특징을 파악 # R에서 table()과 같다.  
# Name: Pos - 열이름 # dtype: int64 - long형(8byte)이다.
```

```
17]: SG      50  
C       50  
Name: Pos, dtype: int64
```

데이터 시각화

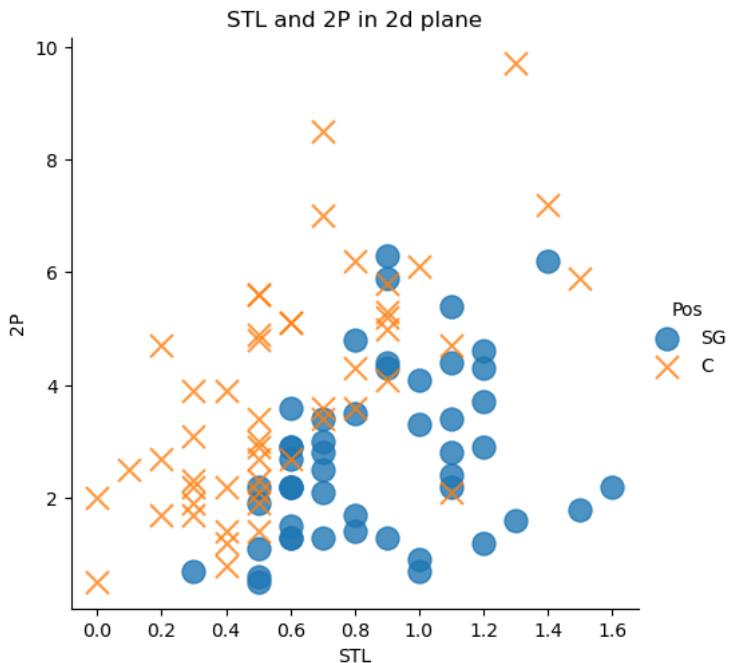
- 데이터 특징을 바탕으로 한 공간에 시각화함으로써, 우리는 머신러닝 학습에 필요한 특징과 불필요한 특징을 쉽게 구분할 수 있다.

```
[1]: import matplotlib.pyplot as plt # py = python
import seaborn as sns

# 스플트, 2점차 데이터 시각화
sns.FacetGrid(x='STL', y='2P', data=df, fit_reg=False, scatter_kws={'s':150}, markers=['o', 'x'], hue='Pos')
# Y축이 STL, X축이 2P # x축, y축, 데이터, line없음 # 좌표 상의 '점'의 크기 # hue - 색상값

# title
plt.title('STL and 2P in 2d plane')

[2]: Text(0.5, 1.0, 'STL and 2P in 2d plane')
```



seaborn 매개변수 설명 사이트

seaborn.lmplot - seaborn 0.12.1 documentation

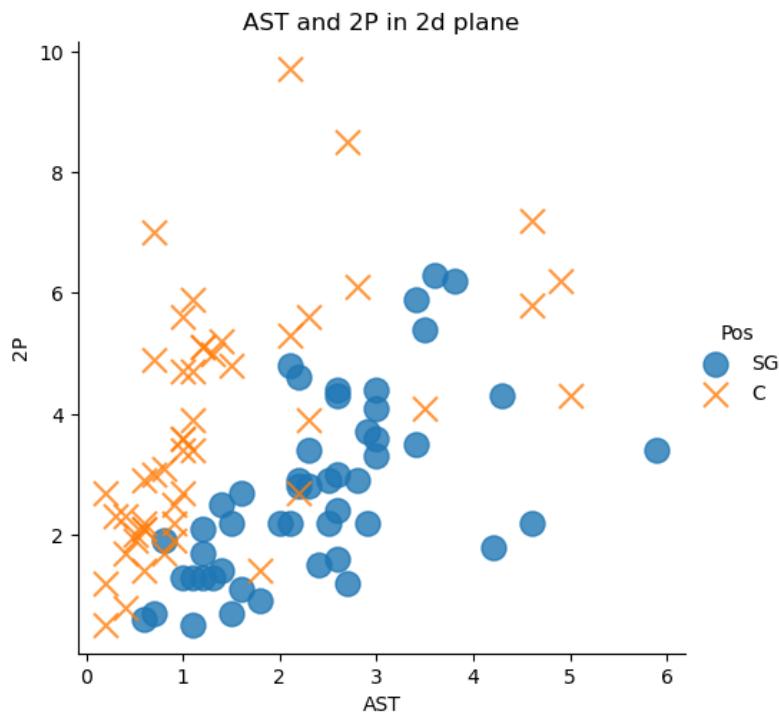
 <https://seaborn.pydata.org/generated/seaborn.lmplot.html>

```

30]: # 어시스트, 2점슛 데이터 시각화
sns.lmplot(x='AST', y='2P', data=df, fit_reg=False, scatter_kws = {'s':150}, markers=['o', 'x'], hue='Pos')
# Y축이 STL, X축이 2P # x축, y축, 대비타, line없음 # 좌표 상의 '점'의 크기 # hue - 예측값
# https://seaborn.pydata.org/generated/seaborn.lmplot.html
# title
plt.title('AST and 2P in 2d plane')

```

30]: Text(0.5, 1.0, 'AST and 2P in 2d plane')



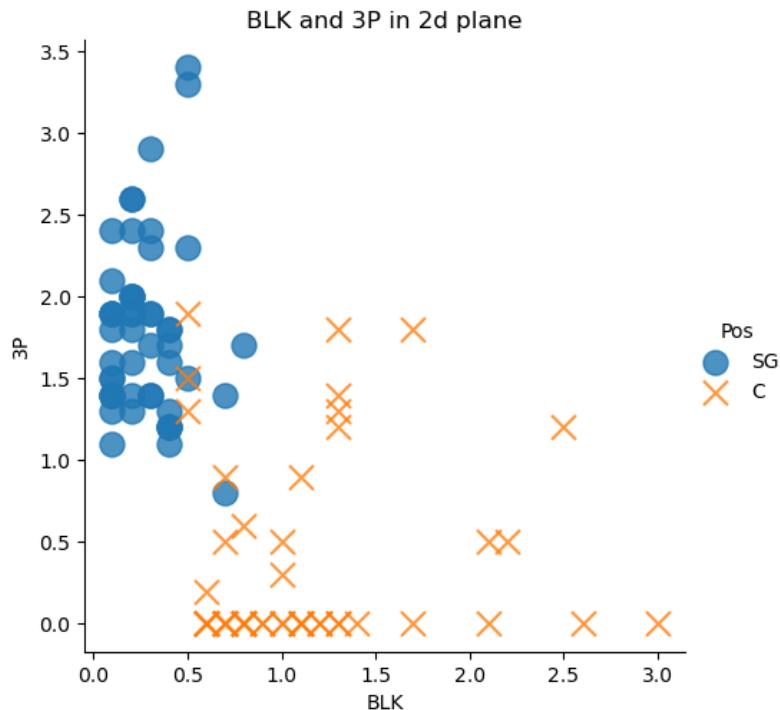
3점 슛을 많이 쏘는 사람들이 슈팅가드,

블로킹을 많이 하는 사람들이 센터

```
[31]: # 블로킹, 3점슛 데이터 시각화
sns.lmplot(x='BLK', y='3P', data=df, fit_reg=False, scatter_kws = {'s':150}, markers=['o', 'x'], hue='Pos')
# Y축이 STL, X축이 2P # x축, y축, 데이터, line없음 # 좌표 상의 '점'의 크기 # hue -色泽값

# title
plt.title('BLK and 3P in 2d plane')
```

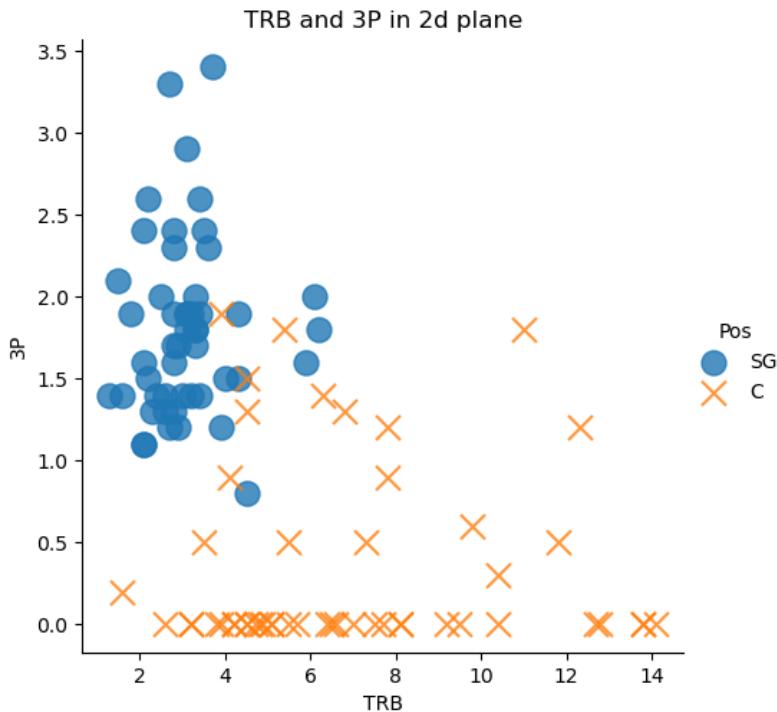
[31]: Text(0.5, 1.0, 'BLK and 3P in 2d plane')



```
[33]: # 리바운드, 3점슛 데이터 시각화
sns.lmplot(x='TRB', y='3P', data=df, fit_reg=False, scatter_kws = {'s':150}, markers=['o', 'x'], hue='Pos')
# Y축이 STL, X축이 2P # x축, y축, 데이터, line없음 # 좌표 상의 '점'의 크기 # hue - 색상값

# title
plt.title('TRB and 3P in 2d plane')
```

: [33]: Text(0.5, 1.0, 'TRB and 3P in 2d plane')



데이터 전처리_20221111

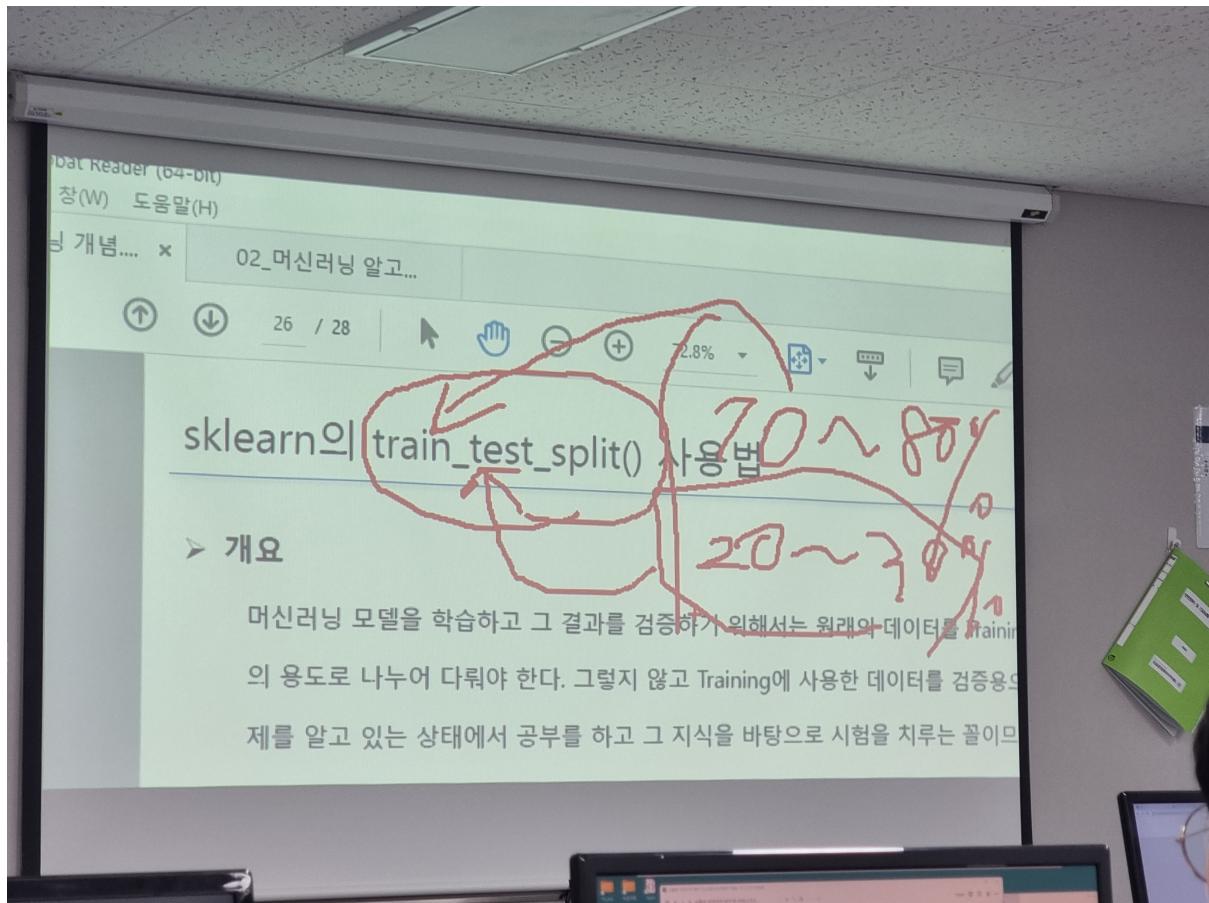
```
: # 분별력이 없는 특징(feature)을 데이터에서 제거.
df.drop(['2P', 'AST', 'STL'], axis=1, inplace=True)
df.head()
```

:

	Player	Pos	3P	TRB	BLK
0	Alex Abrines	SG	1.4	1.3	0.1
1	Steven Adams	C	0.0	7.7	1.0
2	Alexis Ajinca	C	0.0	4.5	0.6
3	Chris Andersen	C	0.0	2.6	0.6
4	Will Barton	SG	1.5	4.3	0.5

훈련 70~80%

검증 20~30% - 한번도 보지 않은 데이터(random으로 뽑아냄 random.rand)



sklearn의 train_test_split() 사용법

➤ 개요

머신러닝 모델을 학습하고 그 결과를 검증하기 위해서는 원래의 데이터를 Training, Validation, Testing의 용도로 나누어 다뤄야 한다. 그렇지 않고 Training에 사용한 데이터를 검증용으로 사용하면 시험문제를 알고 있는 상태에서 공부를 하고 그 지식을 바탕으로 시험을 치루는 꼴이므로 제대로 된 검증이 이루어지지 않기 때문이다.

딥러닝을 제외하고도 다양한 기계학습과 데이터 분석 툴을 제공하는 scikit-learn 패키지 중 model_selection에는 데이터 분할을 위한 train_test_split 함수가 들어 있다.

데이터 나누기(학습 데이터, 테스트 데이터)

```
1]: # scikit-learn library 설치  
# !conda install scikit-learn
```

```
2]: from sklearn.model_selection import train_test_split
```