Model-Agnostic Methods

윤지원

Model-Agnostic이란?
(=independent)

In this position paper, we argue for separating explanations from the model (i.e. being model agnostic).

In model-agnostic interpretability, the model is treated as a black box. The separation of interpretability from the model thus frees up the model to be as flexible as necessary for the task, enabling the use of any machine learning approach - including, for example, arbitrary deep neural networks.

Ribeiro MT, Singh S, Guestrin C. Model-Agnostic Interpretability of Machine Learning. Published online 2016.

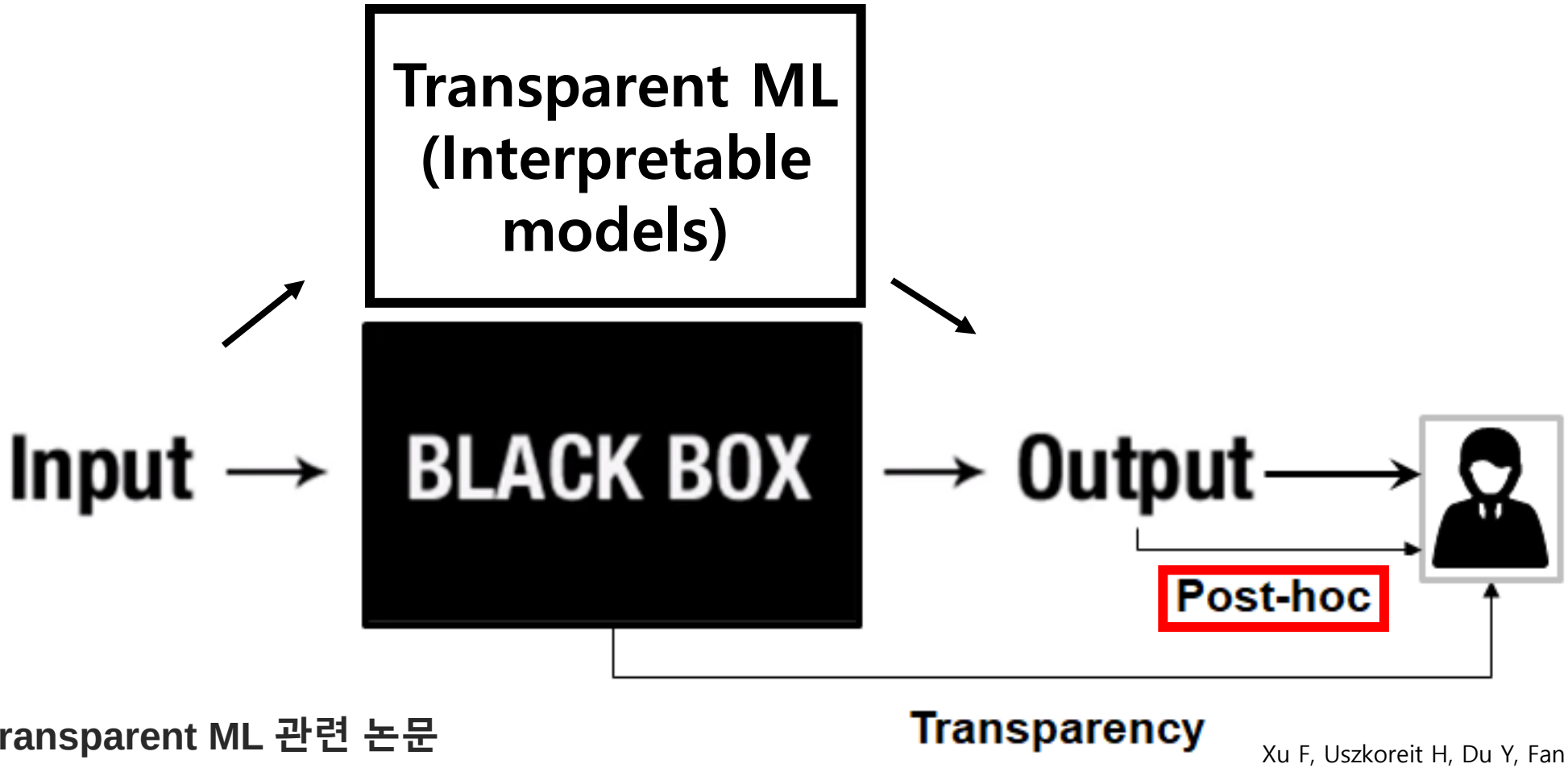
모델과 해석을 분리시킴 ▶ 유연성(flexibility)

유연성 (flexibility)

모델: 어떤 모델에도 적용 가능
ex) RF, NN

해석: 특정 해석 방식에 얽메이지 않음
ex) 수식, 그래프

표현: 표현하고 싶은 것을 표현
ex) 특정 feature



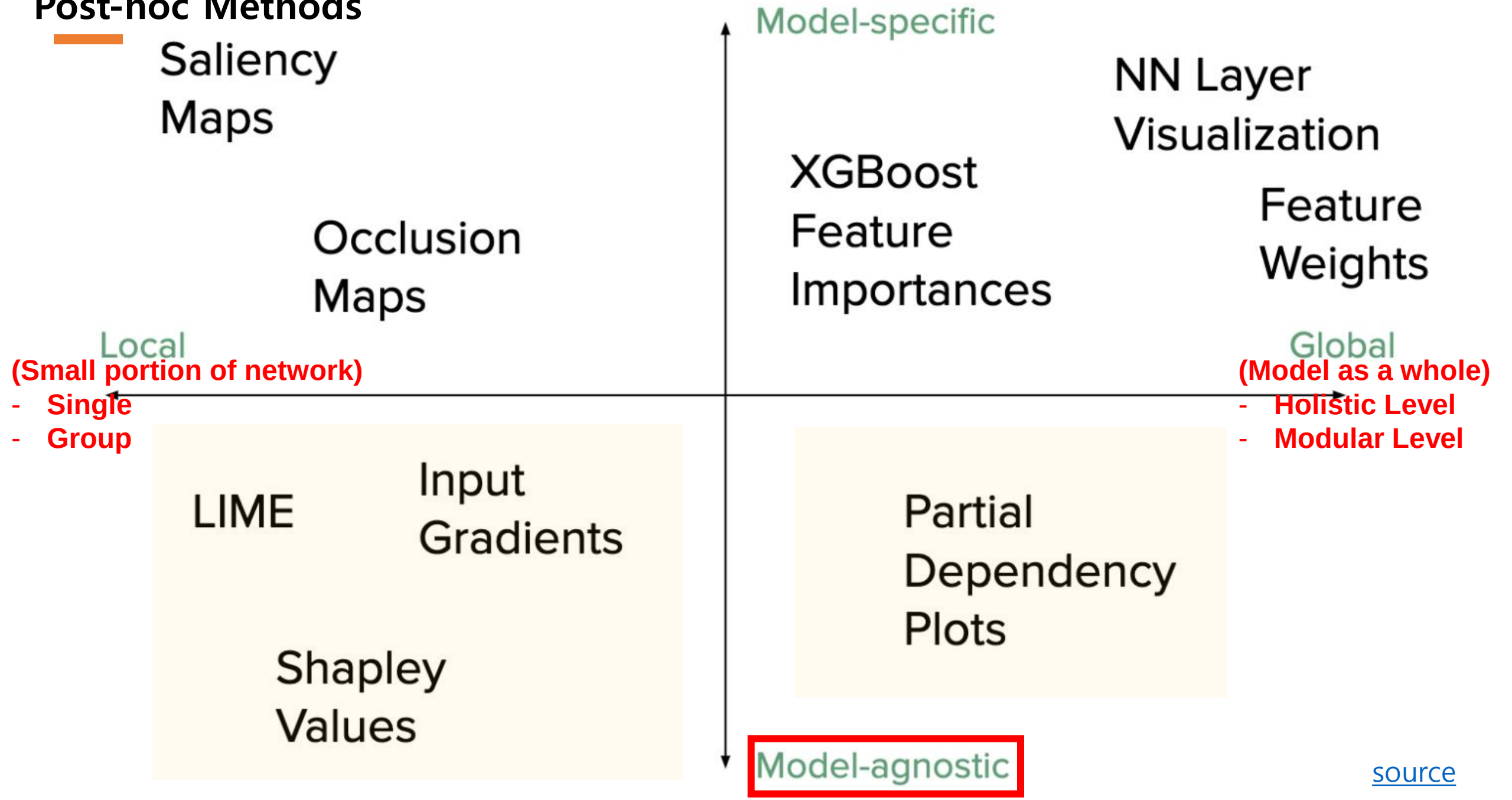
참고: Transparent ML 관련 논문

Alvarez Melis, D., Jaakkola, T., 2018. Towards robust interpretability with self-explaining neural networks, NIPS

Lin, J., Zhong, C., Hu, D., Rudin, C., Seltzer, M., 2020. Generalized and Scalable Optimal Sparse Decision Trees, ICML

Xu F, Uszkoreit H, Du Y, Fan W, Zhao D, Zhu J. Explainable AI: A Brief Survey on History, Research Areas, Approaches and Challenges. In: ; 2019:563-574. doi:10.1007/978-3-030-32236-6_51

Post-hoc Methods



source

Post-hoc Method (참고)

model specific

: 특정 모델에 의존하여 해석

- Local
- Global

model-agnostic

: 모델에 상관없이 해석

- Local
- Global

Local

: 모델의 작은 부분만 다룸

- Single : 단일 instance를 자세히 보고, 어떻게 모델이 예측하는지 알아보는 방식
- Group : global method를 적용시킨 후, 관심 있는 예측 결과만 묶어서 전체 데이터 구성. 다시 local method 적용

Global

: 모델 전체를 다룸

- Holistic : 모델이 어떻게 결정을 내리고 데이터 feature를 바라보는지 해석. (현실적으로 힘듦)
- Modular : 모듈 단위로 해석 (ex. decision tree에서 splits, leaf node)

Explanation Method	Scope	Result
Partial Dependence Plot [142]	Global	Feature summary
Individual Condition Expectation [143]	Global/Local	Feature summary
Accumulated Local Effects Plot [144]	Global	Feature summary
Feature Interaction [145]	Global	Feature summary
Feature Importance [146]	Global/Local	Feature summary
Local Surrogate Model [98]	Local	Surrogate interpretable model
Shapley Values [147]	Local	Feature summary
BreakDown [148]	Local	Feature summary
Anchors [149]	Local	Feature summary
Counterfactual Explanations [87]	Local	(new) Data point
Prototypes and Criticisms [96]	Global	(existent) Data point
Influence Functions [150]	Global/Local	(existent) Data point

Carvalho D, Pereira E, Cardoso J. Machine Learning Interpretability: A Survey on Methods and Metrics. *Electronics*. 2019;8:832. doi:10.3390/electronics8080832

Model-Agnostic Methods

1. Partial Dependence Plot (PDP)
2. Individual Conditional Expectation (ICE)
3. Accumulated Local Effects (ALE)

1. Partial Dependence Plot (PDP)

Definition

Friedman (2001)가 PDP 개념 제안.

i번째 feature가 변화할 때 평균적으로 예측결과가 어떻게 변하는지를 보여줌.

$$\hat{f}_{x_S}(x_S) = E_{x_C}\left[\hat{f}(x_S, x_C)\right] = \int \hat{f}(x_S, x_C) d\mathbb{P}(x_C) = \frac{1}{N}\sum_{i=1}^{N} \hat{f}(\boldsymbol{x}_S, \boldsymbol{x}_{Ci})$$

- 보통 S에는 관심있는 feature 최대 두개까지 들어있음. (ex S={1,2})
- C는 S^c
- x는 feature space
- x_s와 x_c는 feature vector. (x_s =2x1 vector, 1과 2의 x좌표 담고있음)
- 요약: x_s에 관한 distribution을 구하는 과정으로 평균내는 걸 확인할 수 있음. (S가 예측에 어떤 영향을 주는지 알고싶을 때)

장단점

Q. 변수끼리 correlated 되지 않았는지 확인하려면 VIF TOL 같은거 체크?

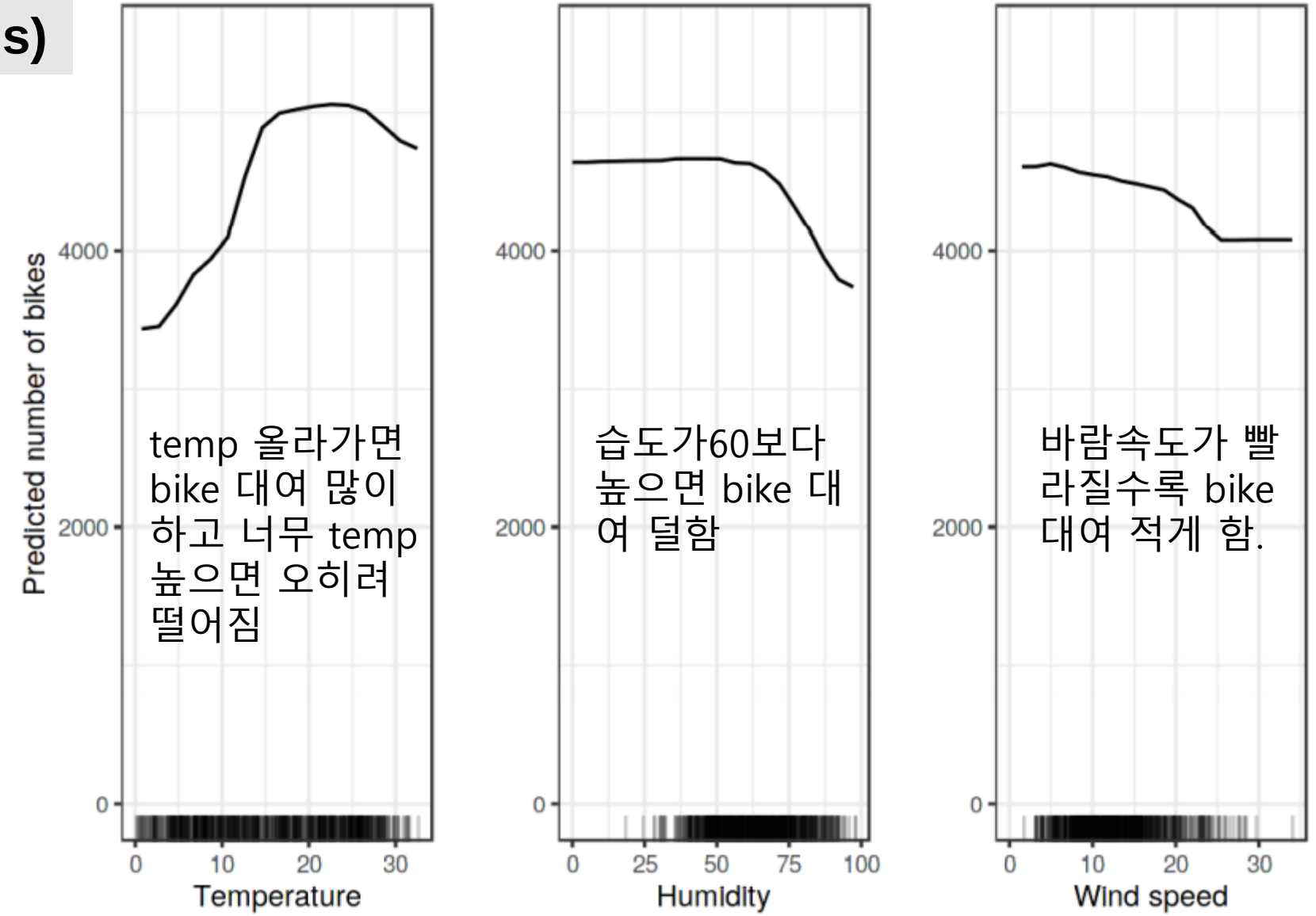
장점 (변수끼리 correlated되지 않은 경우)

- 계산이 직관적
- 해석이 명확
- 특정 변수가 예측에 어떤 영향을 끼치는지 잘 보여줌.
- 적용이 쉬움
- Causal (변수의 변화 -> 예측의 변화)

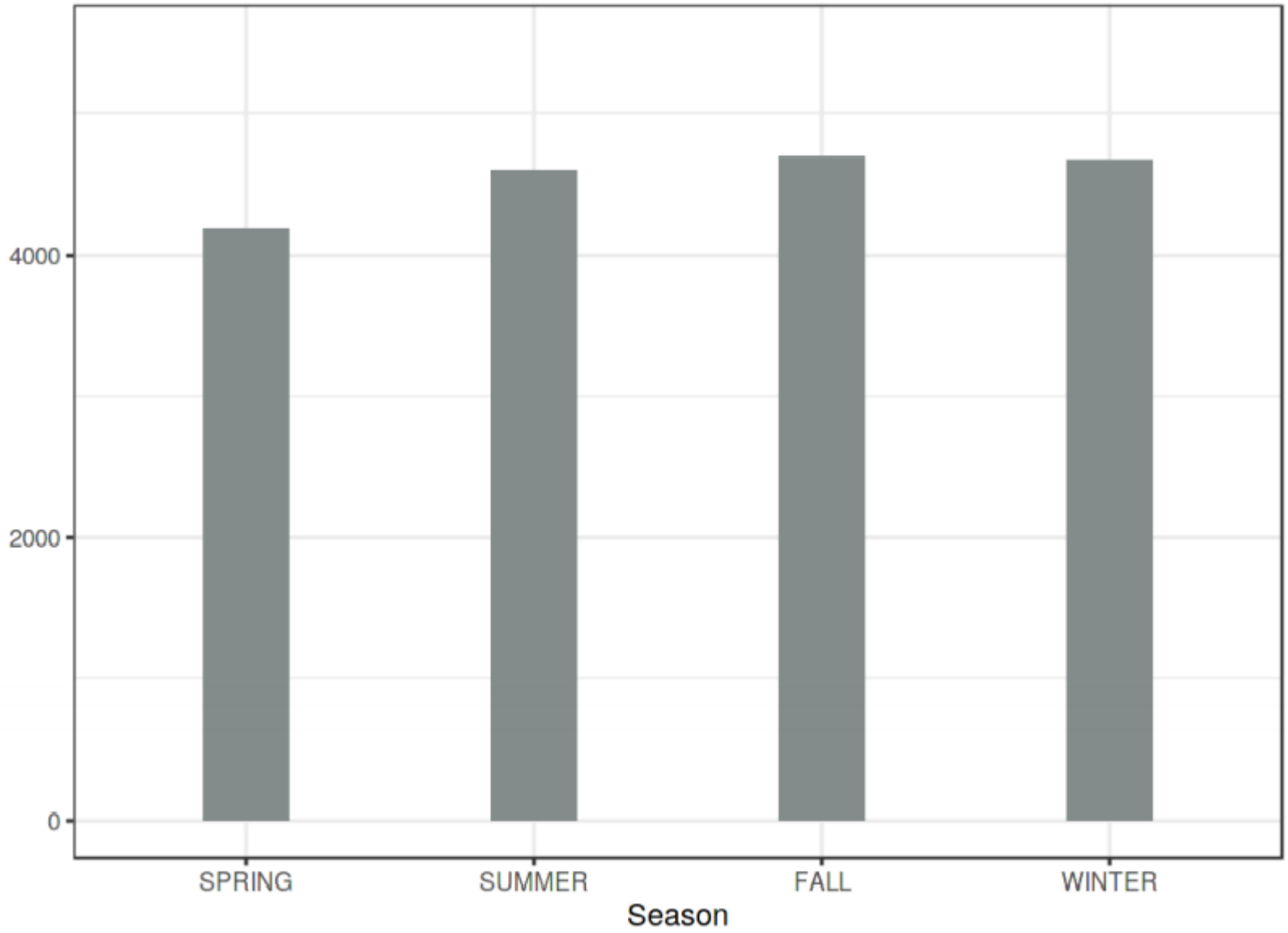
단점

- 이상적인 feature 최대 갯수 2개
- Feature의 distribution을 보여주지 않는 경우도 있음
- 변수끼리 correlated 되면 안됨
  correlated -> ALE plot 사용
- 평균을 사용하기 때문에 효과가 지워질 수 있음-> ICE와 함께 사용

Example (continuous)



Example (categorical)



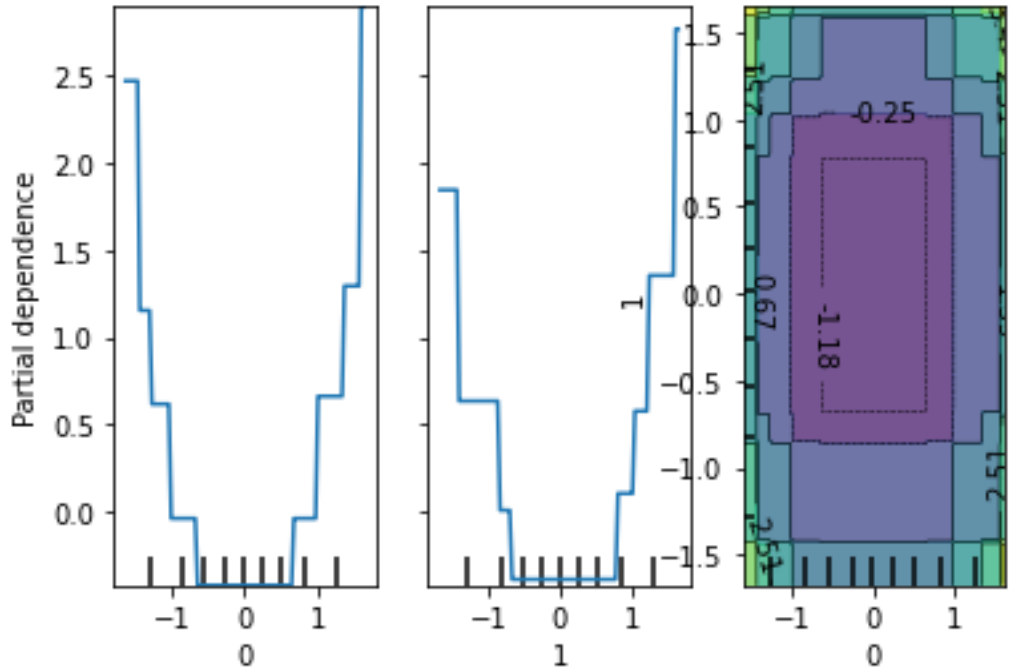
Q. 책에서는 그냥 별차이가 없다고만 했는데, 직관적으로만 확인할 수 있는건지 ANOVA같은 걸 써서 통계적으로 별차이 없다고 할 수 있는 건지.

Code

```
from sklearn.datasets import make_hastie_10_2
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.inspection import plot_partial_dependence
```

```
# data
X, y = make_hastie_10_2(random_state=0)
# model
clf = GradientBoostingClassifier(n_estimators=100, learning_rate=1.0,
    max_depth=1, random_state=0).fit(X, y)
features = [0, 1, (0, 1)]
# plot
plot_partial_dependence(clf, X, features)
```

```
<sklearn.inspection._plot.partial_dependence.PartialDependenceDisplay at 0x25b55637e
80>
```



[Scikit learn link]

Code

```
from sklearn.inspection import partial_dependence
# raw data
pdp, axes = partial_dependence(clf, X, [0])

pdp
axes
```

```
[array([-1.62497055, -1.59201391, -1.55905727, -1.52610063, -1.493144  ,
        -1.46018736, -1.42723072, -1.39427408, -1.36131745, -1.32836081,
        -1.29540417, -1.26244753, -1.22949089, -1.19653426, -1.16357762,
        -1.13062098, -1.09766434, -1.06470771, -1.03175107, -0.99879443,
        -0.96583779, -0.93288115, -0.89992452, -0.86696788, -0.83401124,
        -0.8010546 , -0.76809797, -0.73514133, -0.70218469, -0.66922805,
        -0.63627141, -0.60331478, -0.57035814, -0.5374015 , -0.50444486,
        -0.47148823, -0.43853159, -0.40557495, -0.37261831, -0.33966167,
        -0.30670504, -0.2737484 , -0.24079176, -0.20783512, -0.17487849,
        -0.14192185, -0.10896521, -0.07600857, -0.04305193, -0.0100953 ,
         0.02286134,  0.05581798,  0.08877462,  0.12173126,  0.15468789,
         0.18764453,  0.22060117,  0.25355781,  0.28651444,  0.31947108,
         0.35242772,  0.38538436,  0.418341  ,  0.45129763,  0.48425427,
         0.51721091,  0.55016755,  0.58312418,  0.61608082,  0.64903746,
         0.6819941 ,  0.71495074,  0.74790737,  0.78086401,  0.81382065,
         0.84677729,  0.87973392,  0.91269056,  0.9456472 ,  0.97860384,
         1.01156048,  1.04451711,  1.07747375,  1.11043039,  1.14338703,
         1.17634366,  1.2093003 ,  1.24225694,  1.27521358,  1.30817022,
         1.34112685,  1.37408349,  1.40704013,  1.43999677,  1.4729534 ,
         1.50591004,  1.53886668,  1.57182332,  1.60477996,  1.63773659])]
```

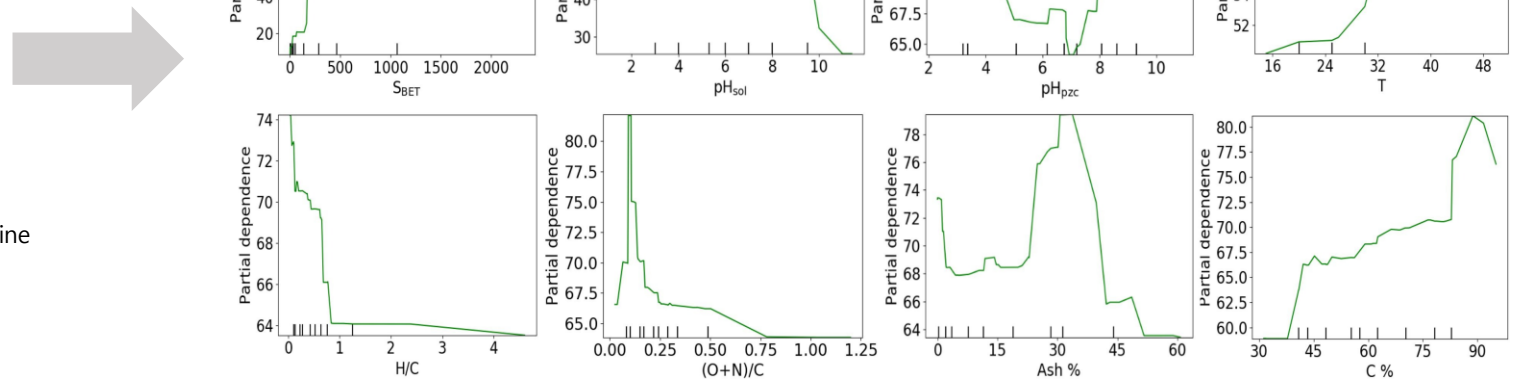
Application

Chemical Engineering

Random Forest (채택)

Gradient Boosting Trees

Artificial Neural Work

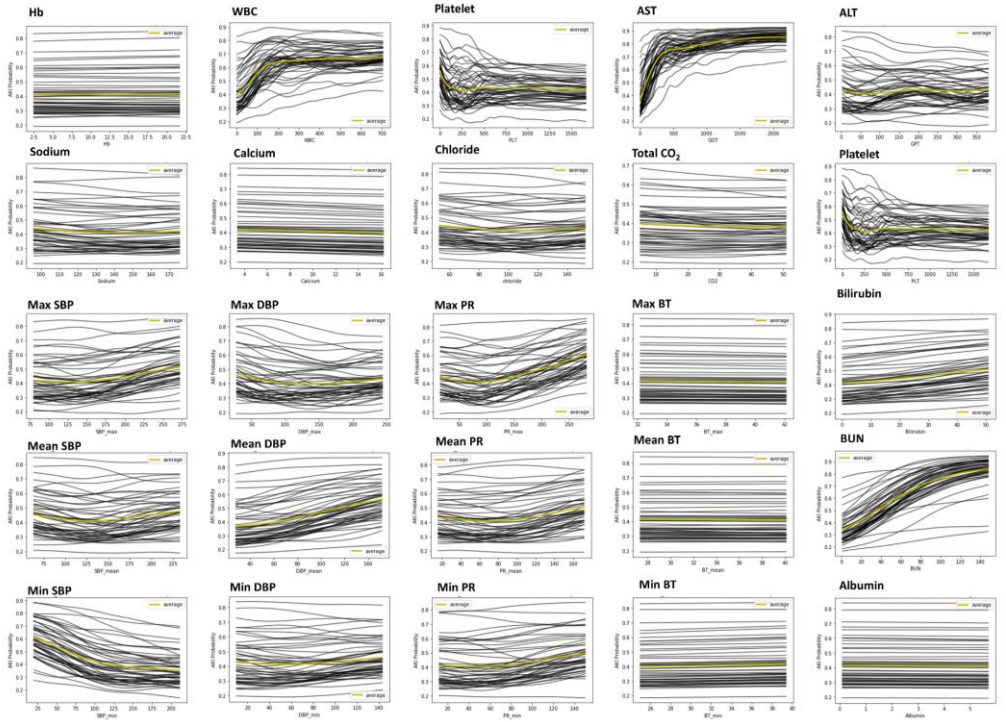


Zhu, X., Wan, Z., Tsang, D. C. W., He, M., Hou, D., Su, Z., & Shang, J. (2021). Machine learning for the selection of carbon-based materials for tetracycline and sulfamethoxazole adsorption. *Chemical Engineering Journal*, *406*, 126782. https://doi.org/https://doi.org/10.1016/j.cej.2020.126782

중요한 변수들을 골라서 PDP를 실행

Application

MED
RNN
Gradient Boosting with XGBoost algorithm



PDP(+ICE)로는 각 환자에서 feature-response 관계를 완전히 설명하지 못했다고 함.
반면 ALE에서 association을 잘 보여줬다는 결론.

Kim K, Yang H, Yi J, et al. Real-Time Clinical Decision Support Based on Recurrent Neural Networks for In-Hospital Acute Kidney Injury: External Validation and Model Interpretation. *J Med Internet Res*. 2021;23(4):e24120. Published 2021 Apr 16. doi:10.2196/24120

Application

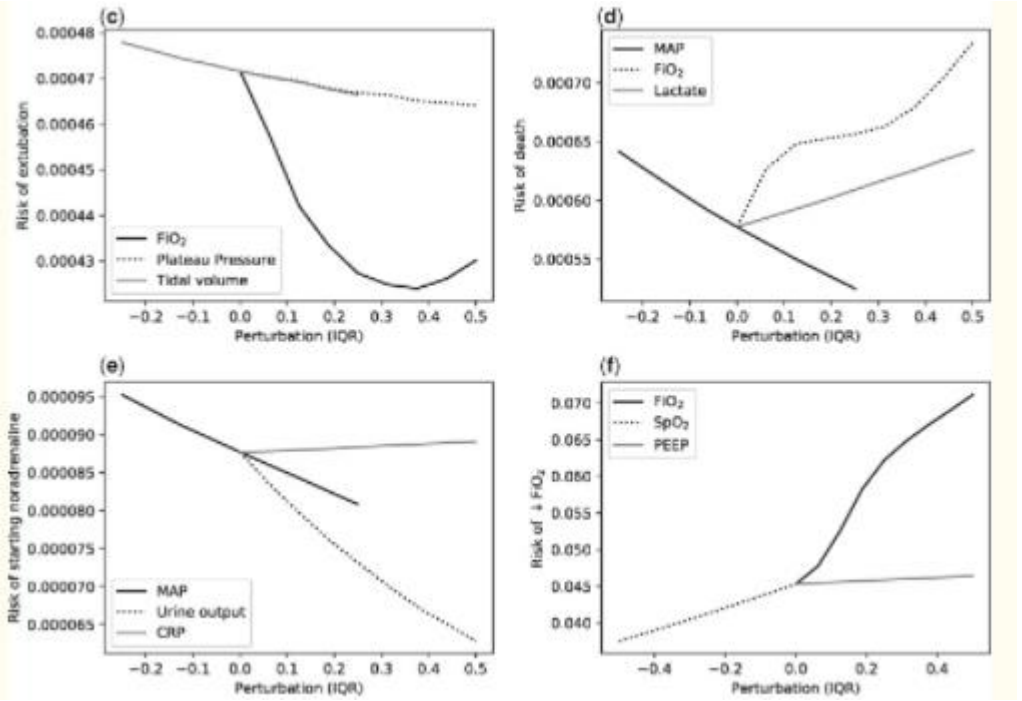
MED

Logistic regression

Feedforward neural network (FFNN)

LSTM-FFNN

TCN-FFNN



선택한 모델이 임상적으로 적용가능한 것인지 확인하기 위해 PDP 사용

Catling FJR, Wolff AH. Temporal convolutional networks allow early prediction of events in critical care. *J Am Med Inform Assoc*. 2020;27(3):355-365. doi:10.1093/jamia/ocz205

결론

Assumption이 필요하다는 단점이 있지만,

Black box 모델로 부터 임상적 의미를 부여하기 위해 은근 사용되는 것 같음 (ICE와 함께)

하지만 모든 PDP가 설명력이 있는것만은 아닌 것 같다.

따라서 여러 post-hoc 방법들을 적용해서 비교하는 것이 좋을 것 같다.

PDP가 세가지 방법중에서는 가장 많이 쓰여온 방법.

2. Individual Conditional Expectation (ICE)

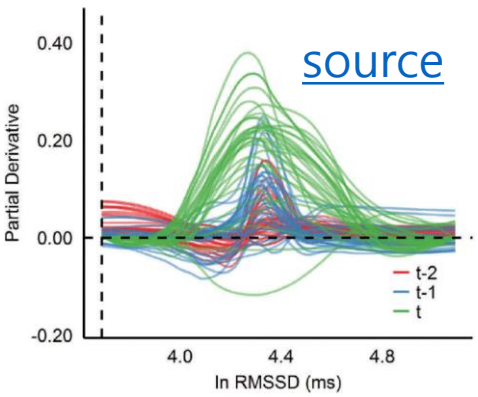
Definition

PDP랑 비슷한 개념인데, (PDP같이 평균이 아니라) 한개의 instance마다 feature에 대한 prediction의 의존성을 그래프로 나타낸것.
- PDP의 단점(평균) 보완

The Centered ICE Plot(c-ICE)
ICE의 문제점은 시작점이 다르기 때문에 변화의 차이를 확인하기 힘들 수도 있음. 따라서 해당 방법으로 특정포인트에서 시작하게 하고 그 변화를 그래프로 확인.

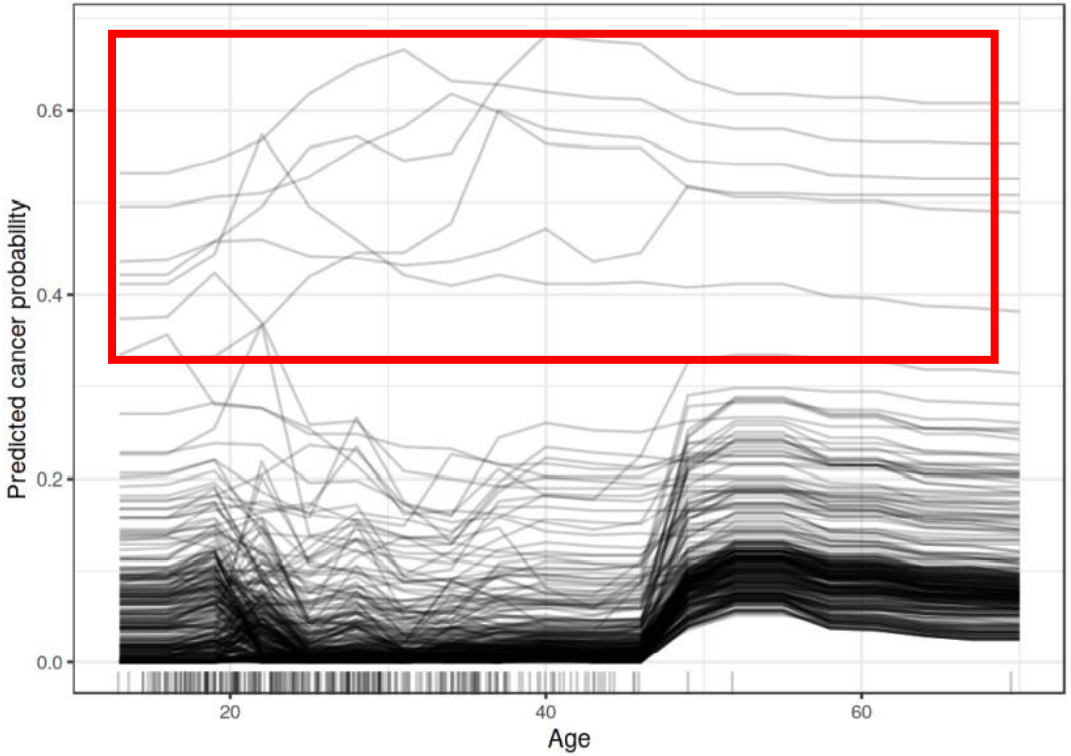
The Derivative ICE Plot(d-ICE)
d-ICE로는 변화의 방향성이나 변화가 일어났는지 여부를 알 수 있음.

source



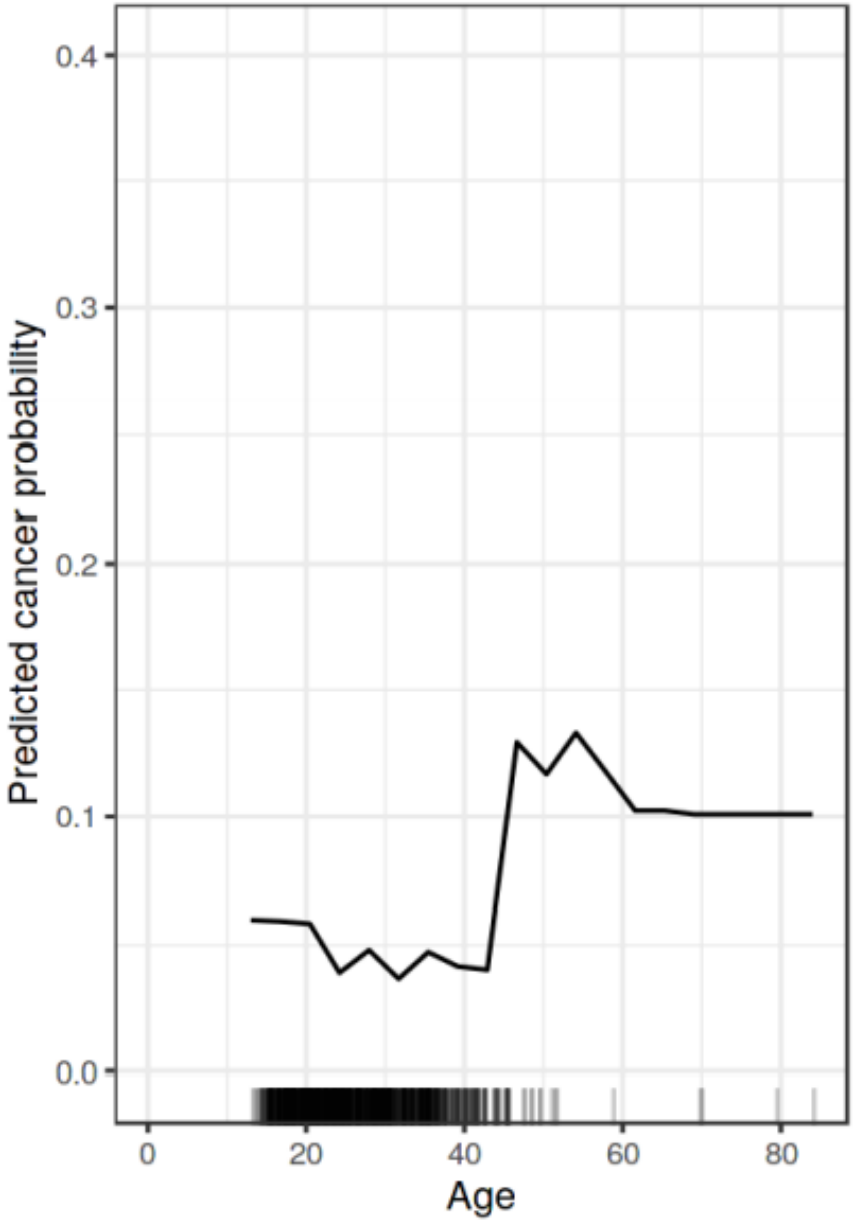
Example

ICE



VS

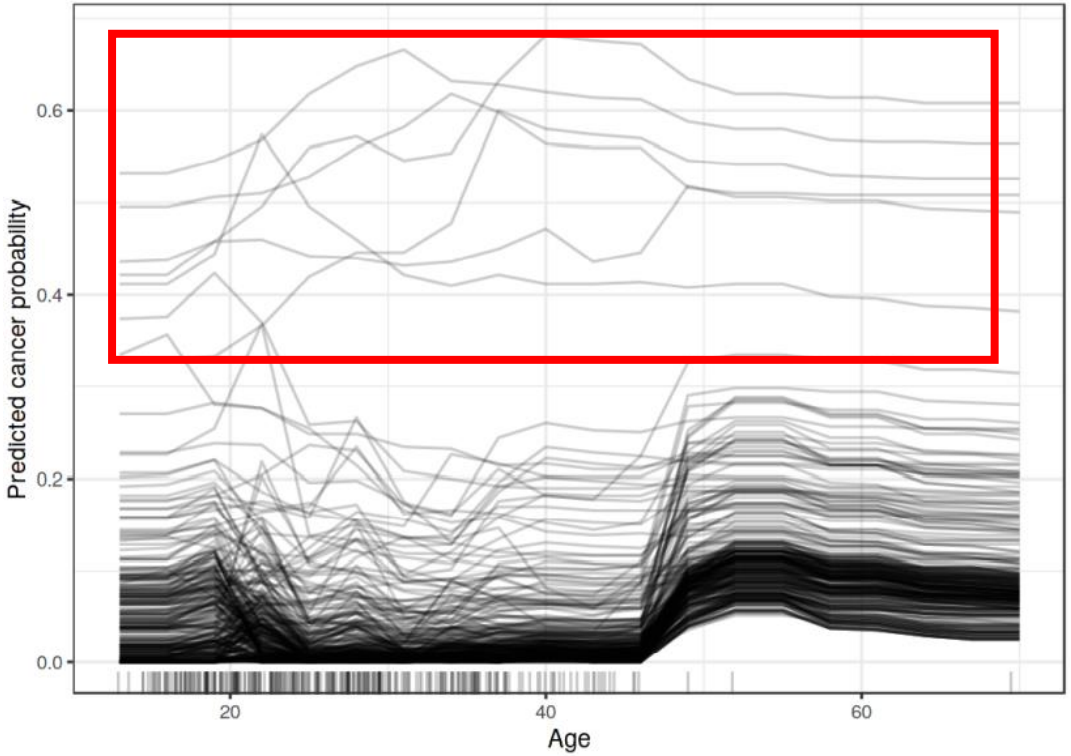
PDP



PDP만 보면 40를 전후로 암 확률이 높아지는 것 같지만, ICE를 보면, 처음부터 리스크가 컸던 사람은 40세와 상관없이 계속 리스크가 크다.

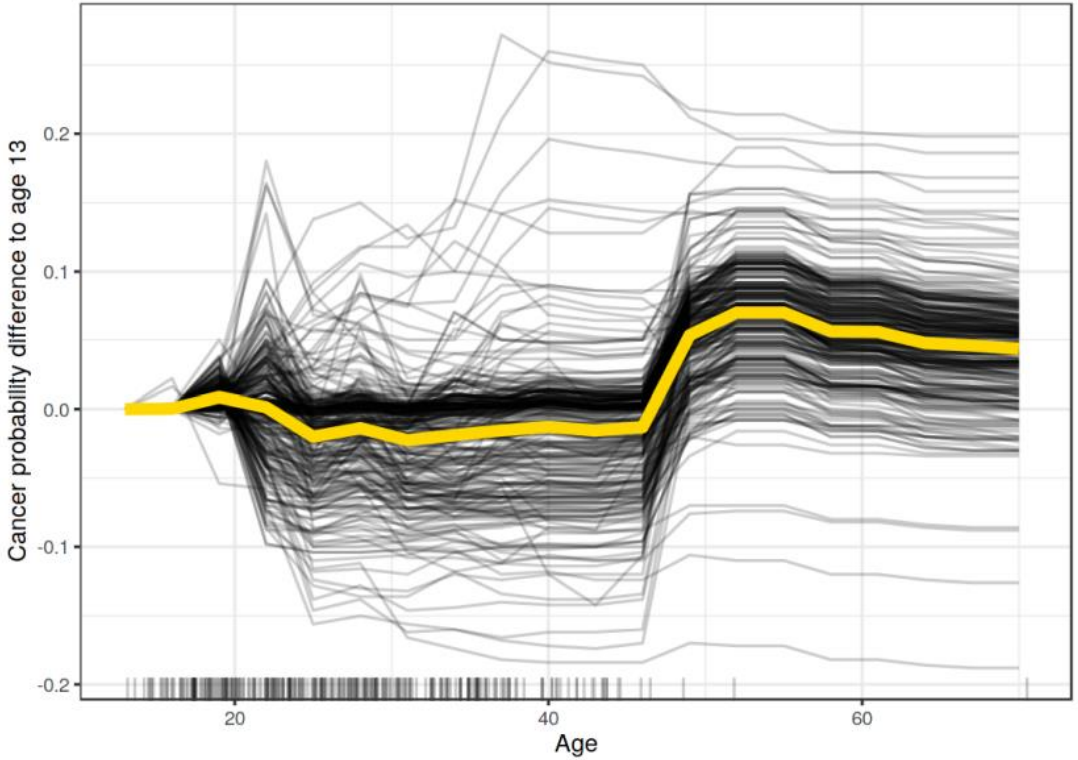
Example

ICE



VS

Centered ICE



predicted probability를 0에 고정.
암에 대한 예측은 대부분 45세 이전에는 바뀌지는 않는다는걸 확인 할 수 있음.

장단점

장점 (변수끼리 correlated되지 않은 경우)
- PDP보다 더 직관적
- 평균내는 걸로 인한 PDP의 단점을 보완

단점
- feature 한개씩만 그래프로 나타낼 수 있음. 그 이상의 feature 로 plot 나타내면 복잡해서 보기 힘듦
- 여전히 feature 끼리 correlated되어 있으면 안됨.
- instance가 너무 많으면 보기 힘듦 (그런 경우 샘플 추출)
- 평균이 뭔지 눈으로 알기 힘들다 (PDP와 함께 그리면 됨)

Code

PDP + ICE

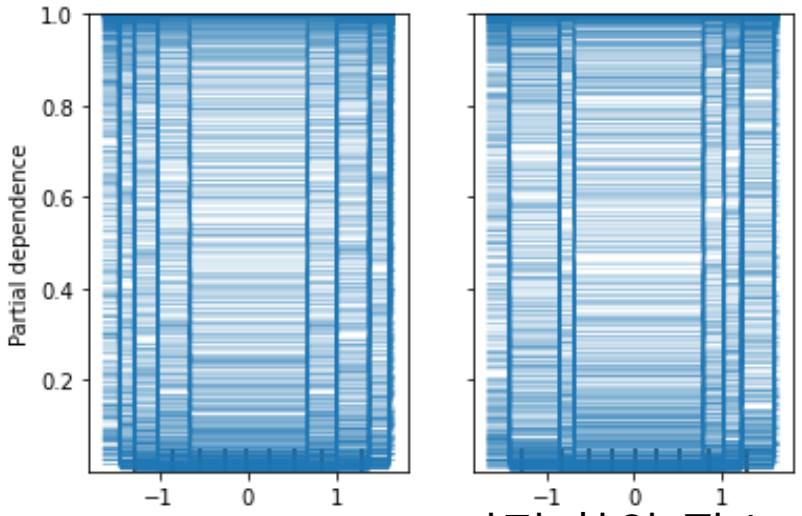
```
from sklearn.datasets import make_hastie_10_2
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.inspection import plot_partial_dependence
```

```
import sklearn
print(sklearn.__version__)
```

0.24.0

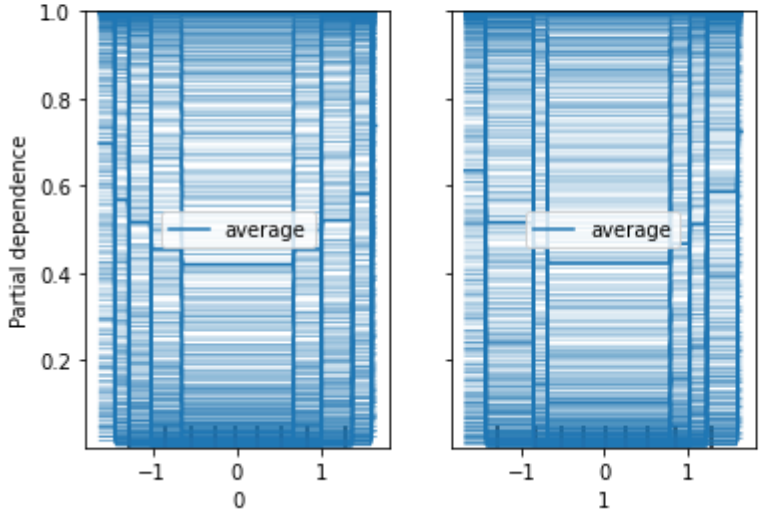
```
X, y = make_hastie_10_2(random_state=0)
clf = GradientBoostingClassifier(n_estimators=100,
                                 learning_rate=1.0,
                                 max_depth=1, random_state=0).fit(X, y)
features = [0, 1]
plot_partial_dependence(clf, X, features, kind='individual')
```

<sklearn.inspection._plot.partial_dependence.PartialDependenceDisplay at 0x1ba7ff75d60>




```
plot_partial_dependence(clf, X, features, kind='both')
```

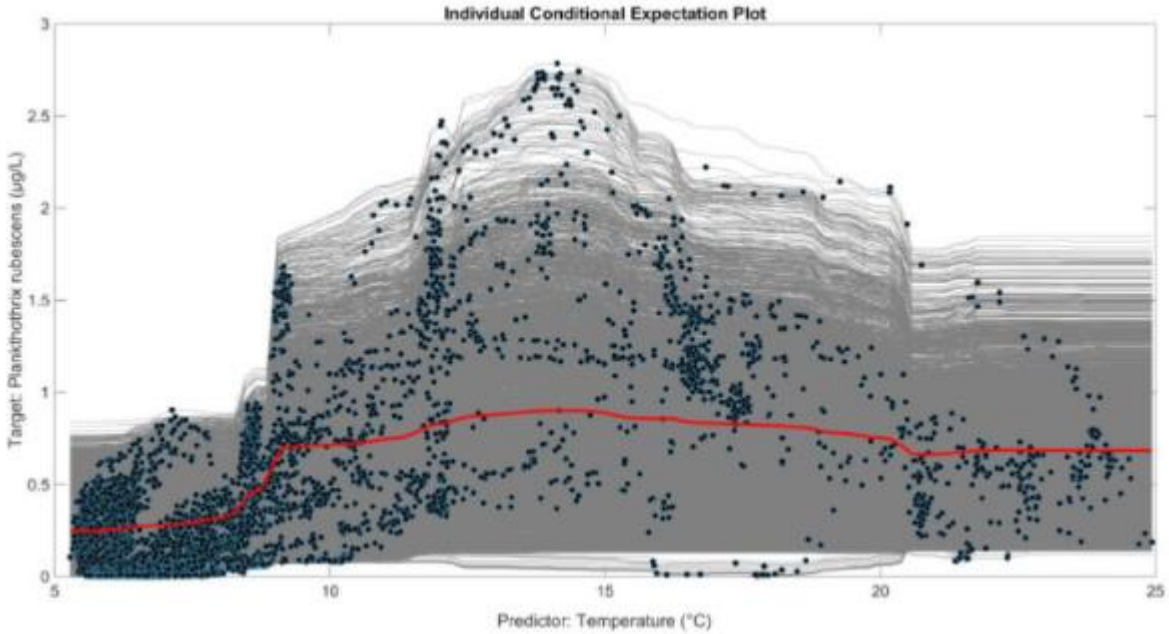
<sklearn.inspection._plot.partial_dependence.PartialDependenceDisplay at 0x1ba7e9b72b0>



ICE -> scikit learn 버전 확인 필수 upgrade

Application

Random Forest



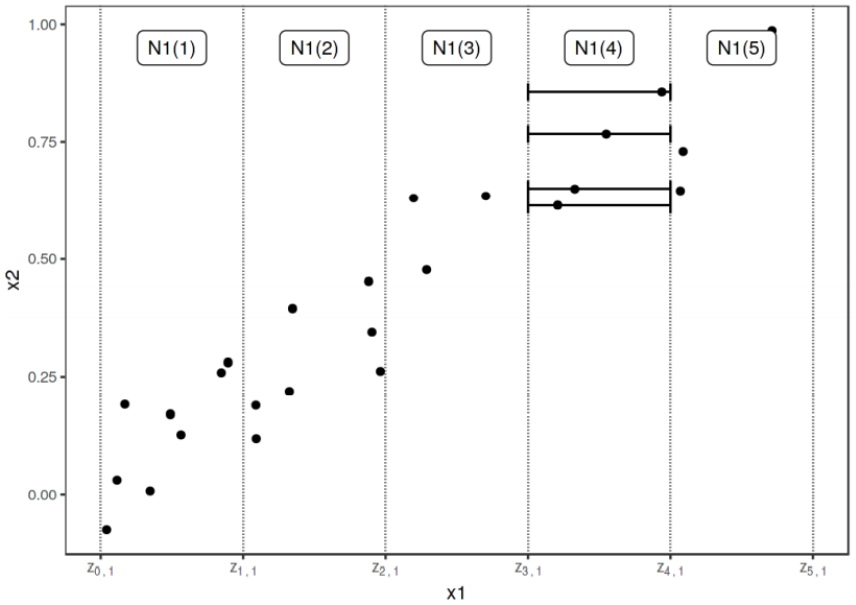
Derot, J., Yajima, H., & Jacquet, S. (2020). Advances in forecasting harmful algal blooms using machine learning models: A case study with Planktothrix rubescens in Lake Geneva. *Harmful Algae*, *99*, 101906. https://doi.org/https://doi.org/10.1016/j.hal.2020.101906

보통 PDP+ICE로 많이 쓰는 것 같습니다.

3. Accumulated Local Effects (ALE)

Definition

앞에 것들과 마찬가지로 feature들이 예측에 어떤 영향을 미치는지를 보는 것.
단, 특정 변수의 효과를 분리시켜 상관관계에 있는 변수들 사용할 수 있음.



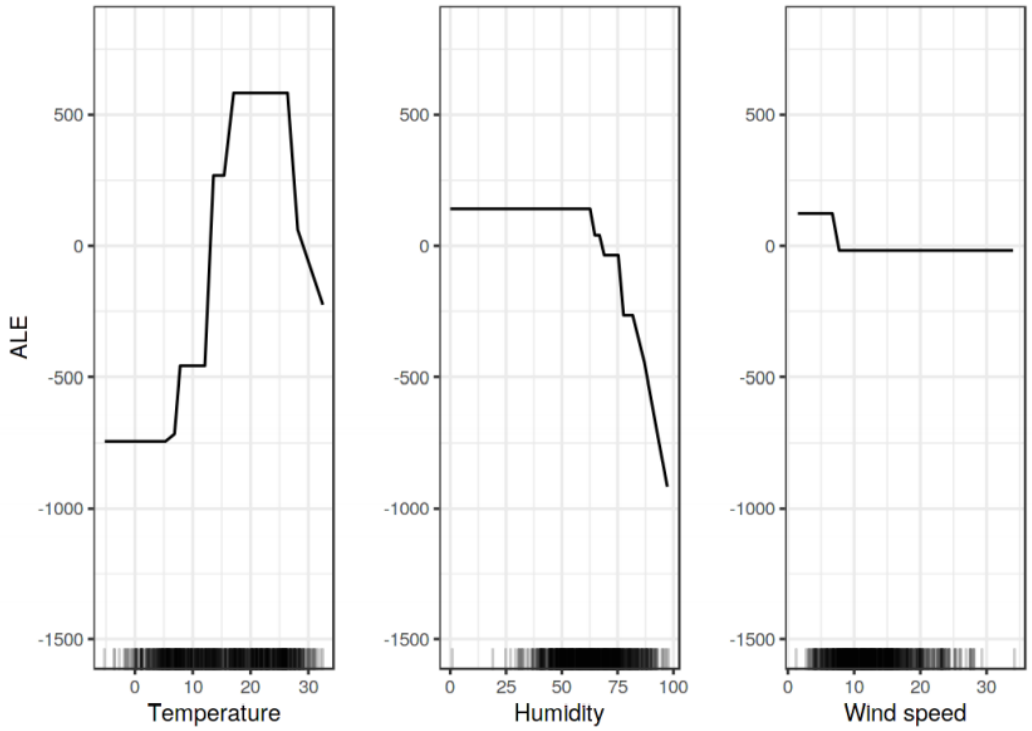
- Conditional distribution을 사용해서 augmented data 생성.
=> 현실적인 데이터 생성 위해 (correlated 되면 비현실적 데이터 생김)

1. trained model 준비
2. 설명하고 싶은 feature들 고르기.
3. feature가 증가/감소 했을 때의 예측 차이 interval 계산.
 - 연속형: 선택된 feature의 분산 계산. 여기서 몇개의 interval사용할지, 분산 범위는 어떻게 할지 고르게 됨.
 - 범주형: 범주들 사이의 순서를 정해서(similarity 측정) interval 계산.
4. 각 interval 마다 관심있는 샘플 주위의 샘플들을 찾아서 conditional distiribution을 측정.
그리고 그 샘플들의 값을 lower,upper value로 바꿔서 차이를 구함.
5. 구한 차이들을 평균내고 차례대로 축적.
6. 앞에서 축적한 것들을 가운데로 위치시켜 평균 효과가 0이 되게 만든다.

STEP 참고 source

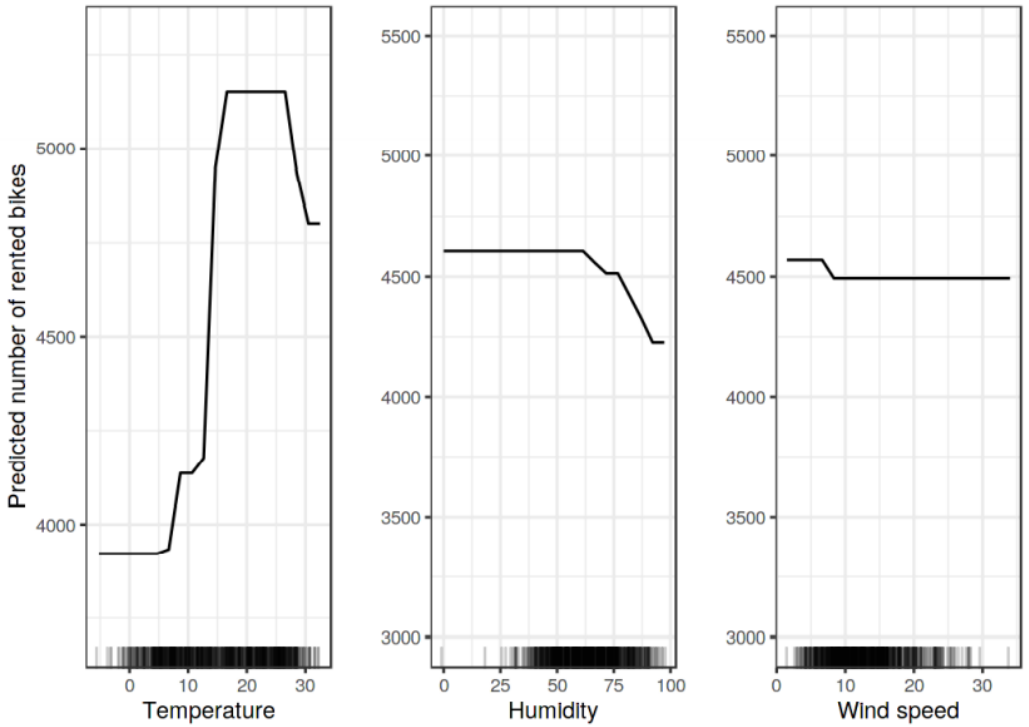
Example (continuous)

ALE



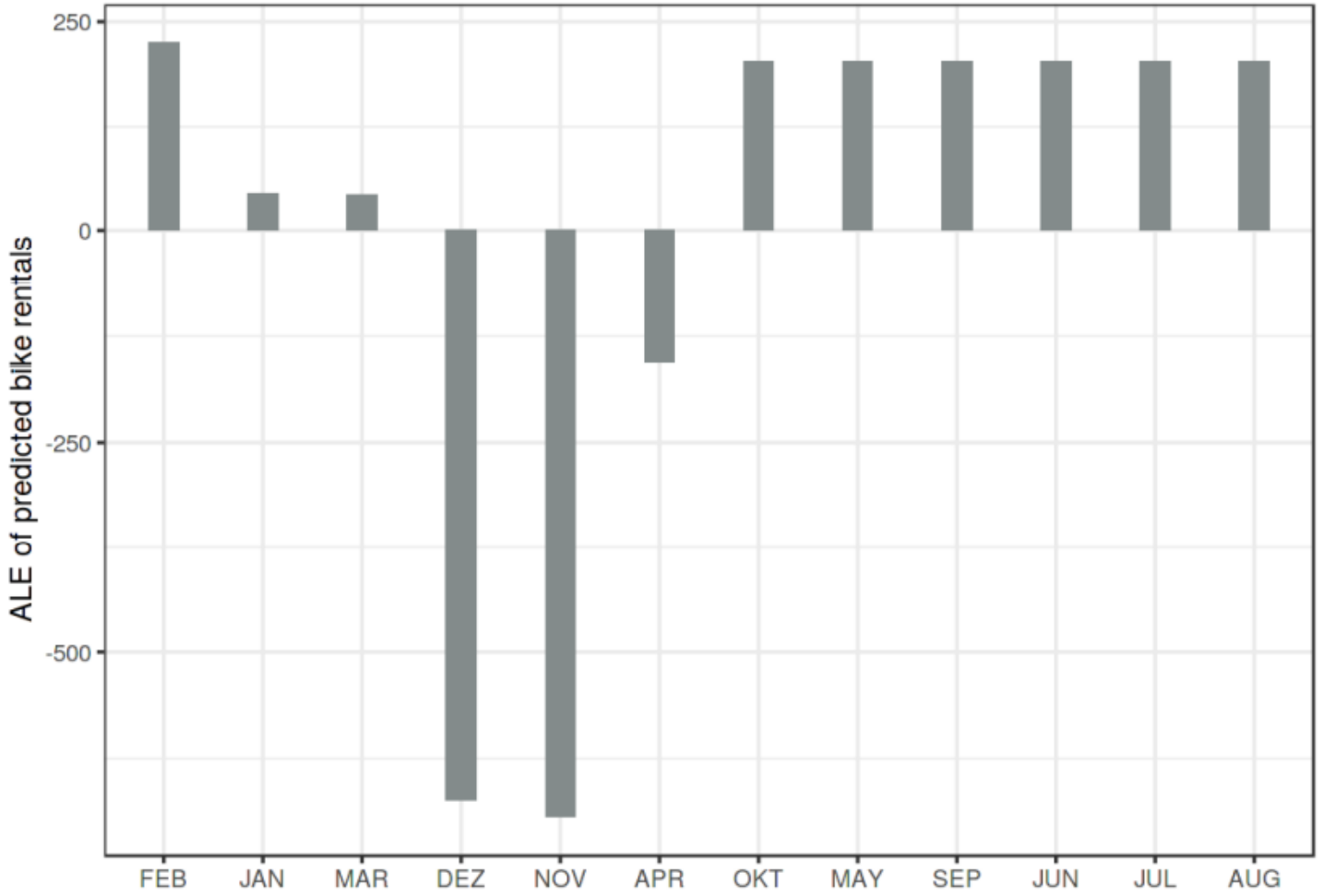
VS

PDP



ALE에 비해 PDP는 temperature 와 humidity에서 변화가 적은 편.

Example (categorical)



추운 계절들과 따뜻한 계절들 사이의 차이를 확인할 수 있습니다.

장단점

장점

- 편향되지 않음. (변수끼리 correlated 되어도 됨)
- 계산이 더 빠르다.
- 해석이 명확하다. (특정 값이 주어졌을 때, feature의 상대적 변화 읽을 수 있음)

단점

- interval이 커서 그래프가 약간 지저분할 수 있음.
- 2차 ALE는 시각화하기 힘듦 (안정x), 해석도 힘듦.
- ALE plot 적용은 생각보다 복잡함
- correlated될 때 사용가능하긴 하나 해석이 약간 어렵다.

Code

Scikit에는 아직은 ALE 없음

-> PyALE (1D, 2D 가능, categorical 코딩 방식에 따라 다르게 코드 작성)

-> ADSModel

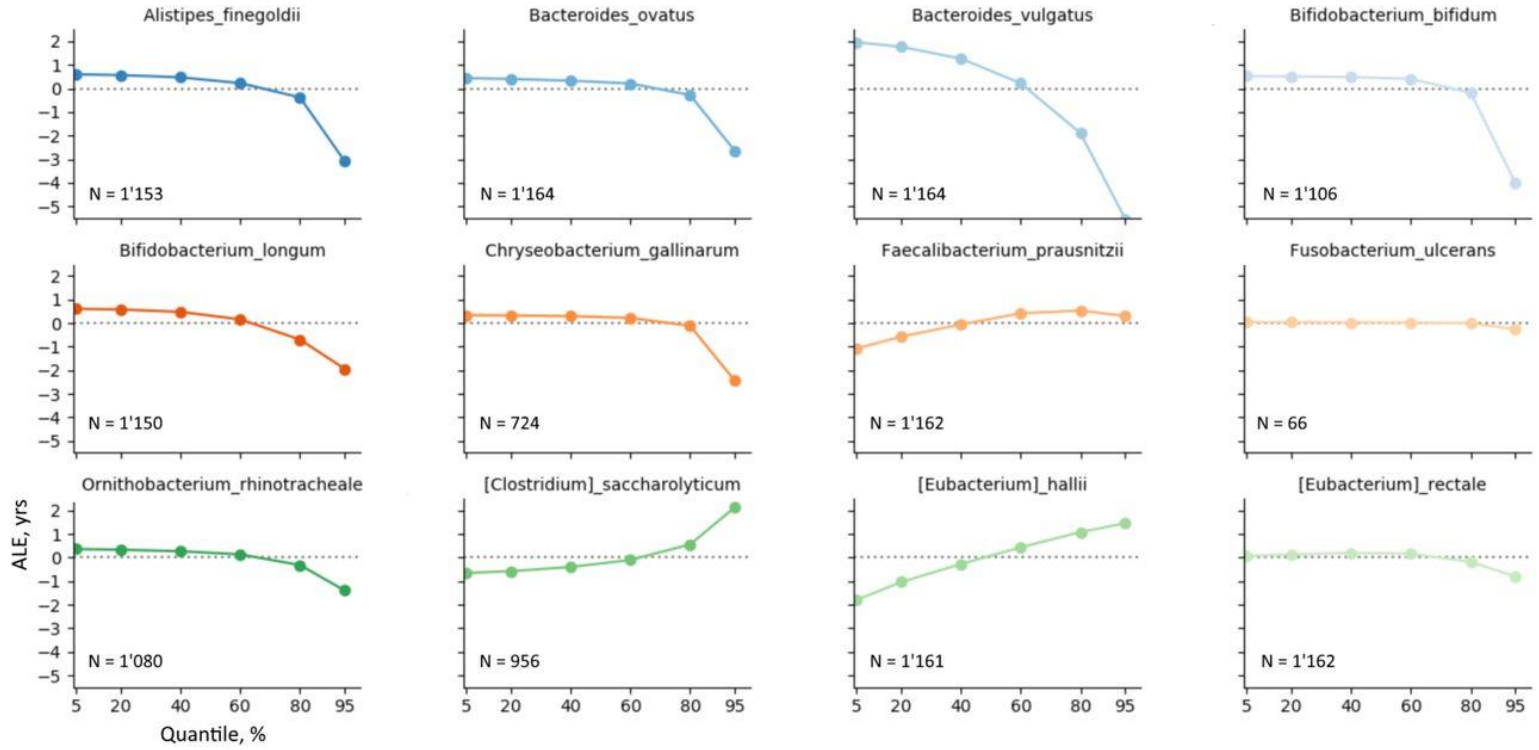
-> 다른사람 git

https://htmlpreview.github.io/?https://github.com/DanaJomar/PyALE/blob/master/examples/Examples.html

-> 직접 제작

Application

DNN



나이 변수가 다른 95개의 feature에 어떤 영향을 끼치는지 보기 위해 ALE를 사용함.

결론

계산방식은 PDP ICE와 좀 다르지만, 그래프 표현은 비슷.

PDP ICE ALE 다 그려보고 선택하는 것도 좋을 것 같음.

책 저자는 ALE 방식을 추천

<Google Scholar 2017년/any time>
Partial Dependence Plot - 1300/1760건
Individual Conditional Expectation – 545/594건
Accumulated Local Effects – 284/296건