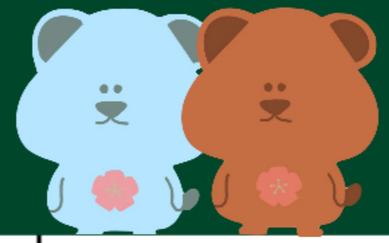
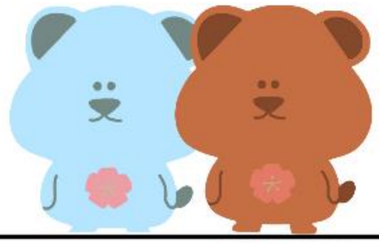


A Unified Approach to Interpreting Model Predictions

인공지능학과 2277018 예서연

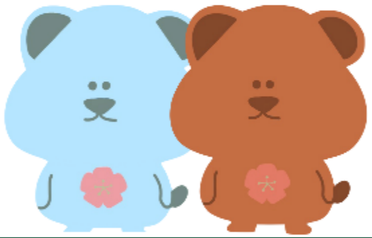
EWHA EURON

목차



EWHA EURON

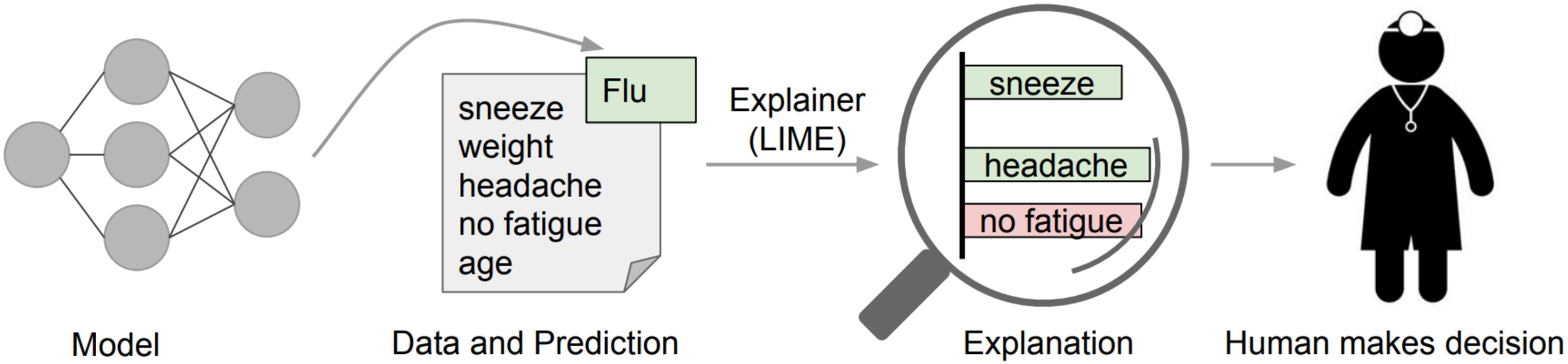
Introduction



EWHA EURON

#01 Introduction

- 인공지능 모델이 현업에서 잘 사용되기 위해서는 모델이 어떤 판단을 왜 내렸는지에 대한 투명한 설명이 요구됨
  e.g., 의료 분야 등
- 모델의 성능(정확도 등)과 해석 가능성 사이에는 trade-off가 존재
- 복잡한 모델을 활용하면서도 그 결과가 어떻게 도출되었는지 해석할 수 있도록 해 주는
  설명 모델(explanation model)의 중요성이 대두됨



이미지 출처: “Why Should I Trust You?” Explaining the Predictions of Any Classifier

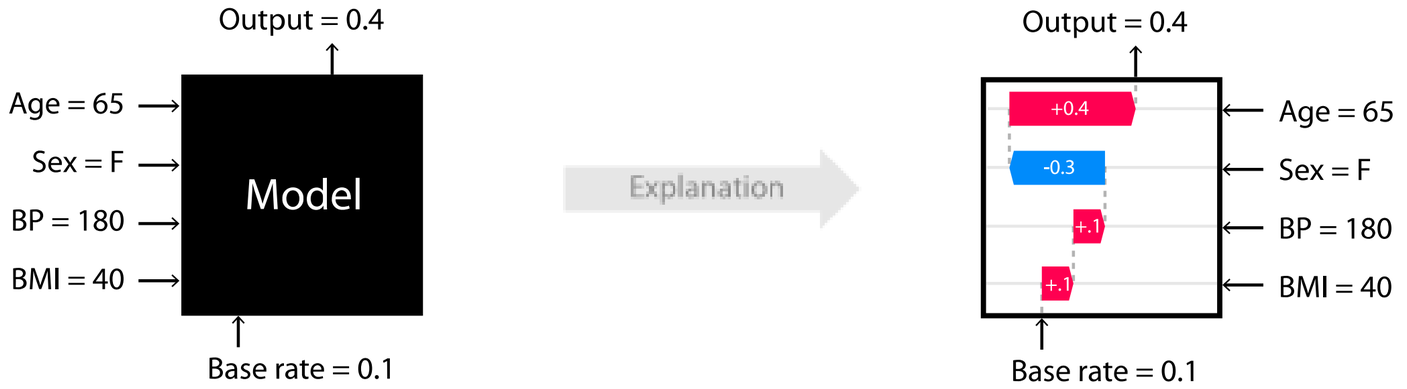
#01 Introduction

1. 모델의 예측 결과에 대한 설명을 제공하는 설명 모델(explanation model)의 개념을 제시하고 Additive Feature Attribution Methods를 정의함

2. 게임 이론의 원리에 따라 feature 중요도를 측정할 수 있는 SHAP Values를 소개함

3. 기존의 설명 방법보다 SHAP Value를 이용한 방법론이 사람의 직관과 더욱 잘 부합함을 보임

SHAP

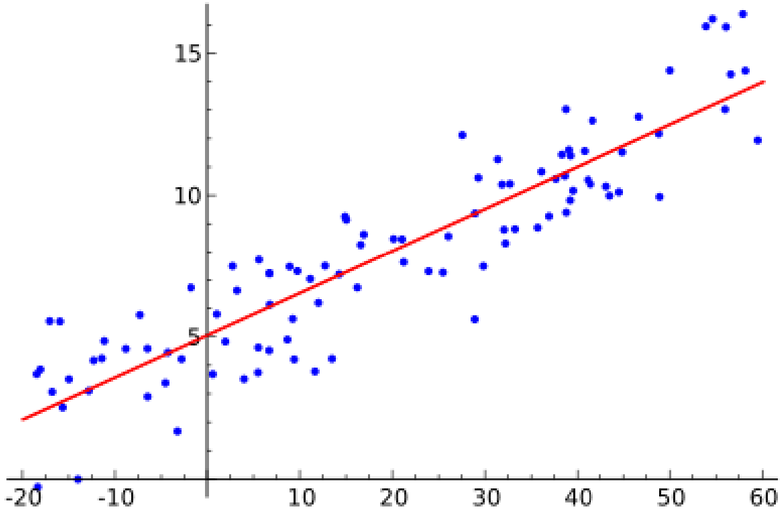


이미지 출처: https://github.com/shap/shap

Explanation Model

EWHA EURON

#02 Explanation Model



단순한 모델

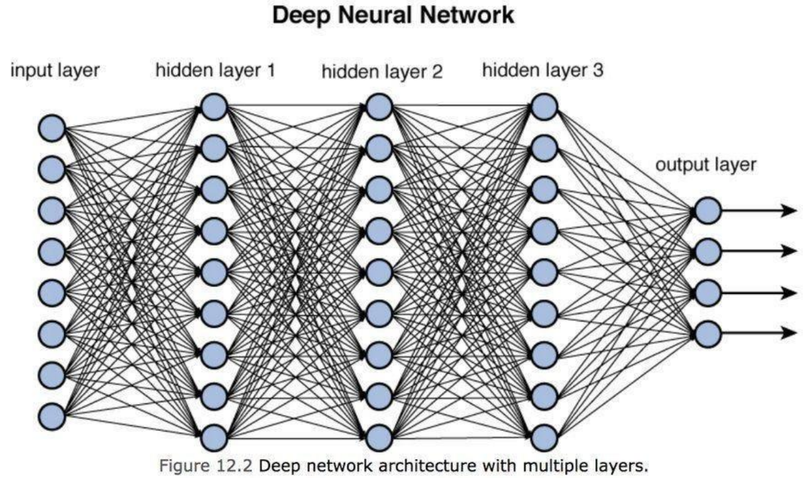
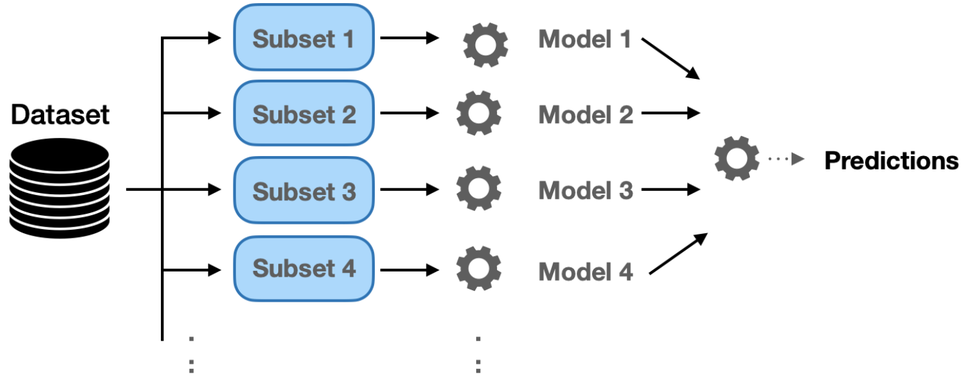


Figure 12.2 Deep network architecture with multiple layers.



복잡한 모델

e.g., deep neural networks, ensemble model

이미지 출처: Wikipedia, Towards Data Science

#02 Explanation Model

Additive Feature Attribution Methods

단순화된 input x를 사용해 기존 모델 f를 근사시켜, f와 유사하지만 해석이 가능한 설명 모델 g를 정의함

$$g(z') \approx f(h_x(z'))$$

$$g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z'_i$$

$z' \in \{0,1\}^M$

x ' 와 유사한, 단순화된 input feature

h_x x ' 를 x로 mapping하는 함수
$x = h_x(x')$

$\phi_i \in \mathbb{R}$

각 feature의 기여도

EWHA EURON

#02 Explanation Model

$$g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z'_i$$

본 논문에서 소개하는 모델 설명 기법은
모두 위의 수식을 따름

=> Additive Feature Attribution Methods의 일종

< Additive Feature Attribution Methods >

LIME	DeepLIFT	Layer-Wise Relevance Propagation	Classic Shapley Value Estimation

#02 Explanation Model

(1) LIME

Local Interpretable Model-agnostic Explanations (LIME)

모델이 현재 데이터의 어떤 영역을 집중 분석했으며 어떤 영역을 분류 근거로 사용했는지 알려주는 기법
특정 데이터 input에 대해 모델이 그것을 어떻게 해석하는지 파악할 수 있음

$$\xi = \arg\min_{g \in \mathcal{G}} L(f, g, \pi_{x'}) + \Omega(g)$$

< Objective function of LIME >

Ω

g의 복잡도에 따라
부과되는 페널티

L

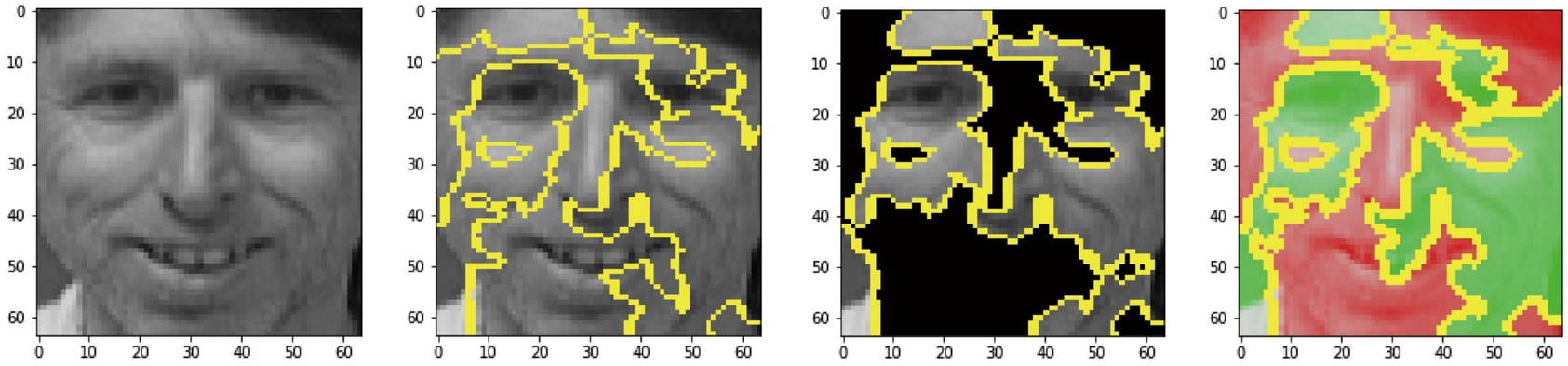
g의 예측값과
f의 예측값 간
squared loss

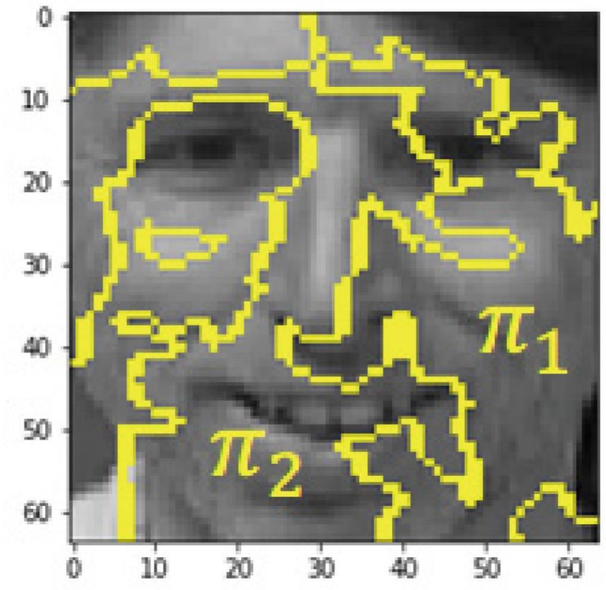
π'_x

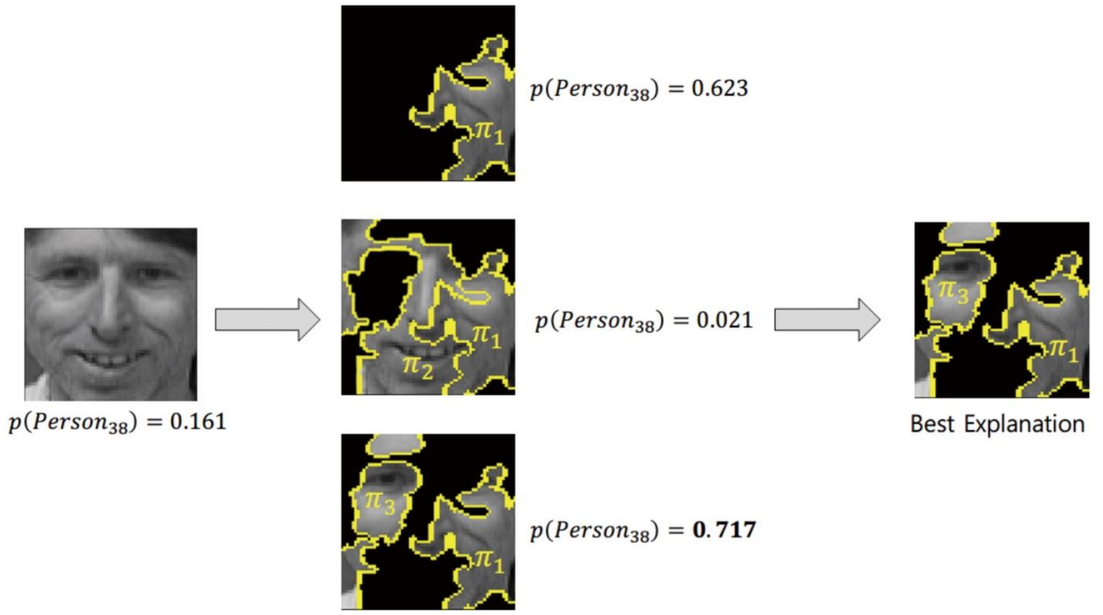
local kernel

#02 Explanation Model

(1) LIME







이미지 출처: XAI 설명 가능한 인공지능, 인공지능을 해부하다

#02 Explanation Model

(2) Classic Shapley Value Estimation

Classic Shapley Value Estimation

전체 성과를 창출하는 데 각 참여자가 얼마나 공헌했는지를 수치로 표현하는 기법
i번째 feature가 기여하는 정도 = 전체 기여도에서 i번째 feature가 제외된 기여도의 합을 뺀 값

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} \left[f_{S \cup \{i\}} x_{S \cup \{i\}} - f_S(x_S) \right]$$

< Definition of Shapley Value >

ϕ_i

feature i에 대한 shapley value

F

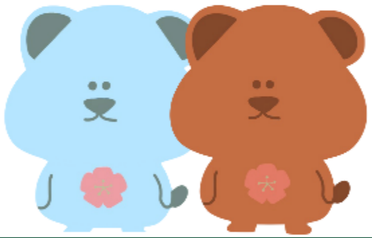
모든 feature 집합

S

F에서 feature i를 제외한 subset

EWHA EURON

Properties



EWHA EURON

#03 Properties

1. Local accuracy

단순화된 input x ' 를 설명 모델에 넣었을 때의 output은, 원래 input x를 원래 모델에 넣었을 때의 output인 f(x)와 일치해야 함

$$f(x) = f(x') = \phi_0 + \sum_{i=1}^{M} \phi_i x'_i$$

2. Missingness

x_i ' = 0은 원래 input x에 feature i가 존재하지 않음을 의미하며, 따라서 feature i의 기여도인 \phi_i 또한 00이 되어야 함

$$x'_i = 0 \Rightarrow \phi_i = 0$$

3. Consistency

모델이 변경되었을 때, 단순화된 input x ' 의 기여도가 증가 또는 유지되었다면 해당 input의 기여도는 감소되어서는 안 됨

$$f'_x(z') - f'_x(z' \setminus i) \geq f_x(z') - f_x(z' \setminus i)$$

EWHA EURON

#03 Properties

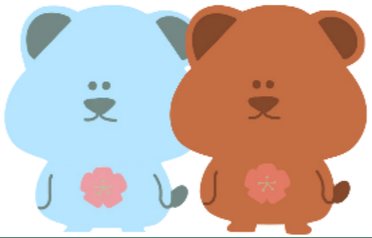
이러한 3개의 property들을 모두 따르며, additive feature attribution methods의 정의를 만족하는 설명 모델은 다음의 수식 하나뿐임

SHAP Value Estimation

$$\phi_i(f, x) = \sum_{z' \subseteq x'} \frac{|z'|!(M - |z'| - 1)!}{M!} \left[f_x(z') - f_x(z' \setminus i) \right]$$

- Classic Shapley Value Estimation과 거의 동일함
- 차이점은 z ' 를 정의할 때 0인 값들을 모두 제외한다는 것뿐임!
  (Shapley Value Estimation이 Property 2. Missingness를 만족시키도록 한 수식)

SHAP Values

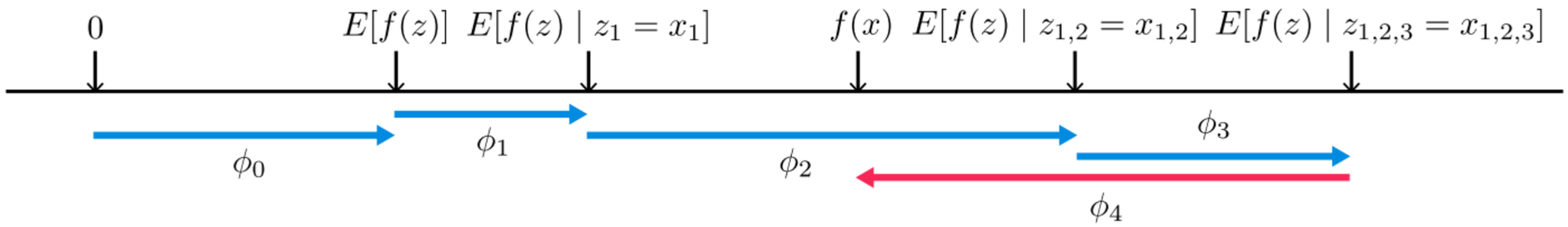


EWHA EURON

#04 SHAP Values

SHapley Additive exPlanation

SHAP은 모델의 출력을 각 feature의 기여도로 분해하는데, 이때 모든 가능한 feature 순서쌍을 이용하므로 각 feature 간의 의존성까지 고려된다는 점에서 일반적인 feature importance 계산과 차이가 있음



한계: SHAP Value를 정확히 계산하는 것이 어려움

Model-agnostic Approximations

Model-specific Approximations

이미지 출처: A Unified Approach to Interpreting Model Predictions

#04 SHAP Values

Explainer 객체 생성 및 SHAP Value 살펴보기

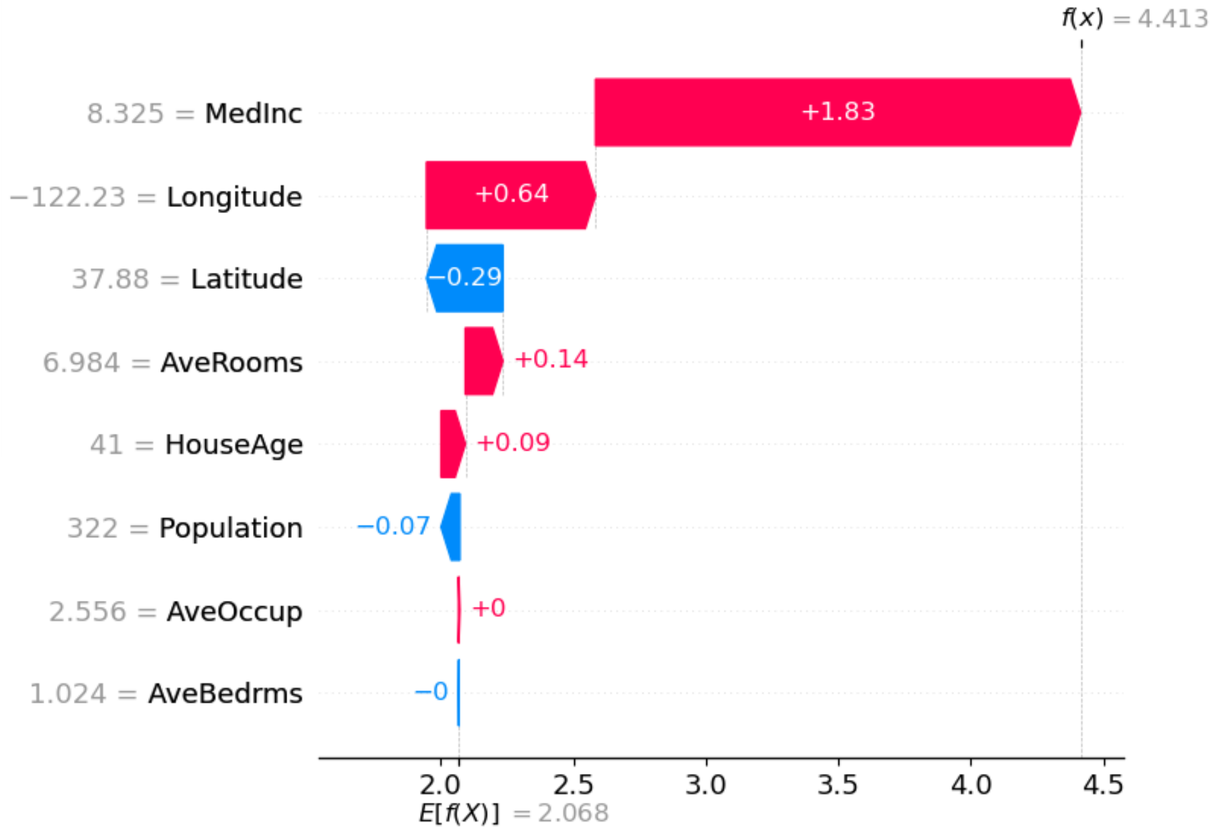
```python
import xgboost
import shap

# XGBoost 모델 학습
X, y = shap.datasets.california()
model = xgboost.XGBRegressor().fit(X,
y)
# SHAP을 이용한 모델 예측 설명
explainer = shap.Explainer(model)
shap_values = explainer(X)

print(shap_values)
```

```
.values =
array([[ 1.7081785 ,  0.09363674,  0.19277047, ...,  0.01571906,
        -0.39385185,  0.55515116],
       [ 1.426717  ,  0.03108795,  0.00601703, ...,  0.2112088 ,
        -0.36280793,  0.5884698 ],
       [ 1.3600677 ,  0.16082455,  0.47361216, ..., -0.02257477,
        -0.5582292 ,  0.5463798 ],
       ...,
       [-0.5842778 ,  0.01744973, -0.0949486 , ...,  0.10111337,
        -0.9798146 ,  0.3479332 ],
       [-0.6035651 ,  0.03118367, -0.05752674, ...,  0.23118298,
        -1.051862  ,  0.32962263],
       [-0.44976887,  0.02051439, -0.12479055, ..., -0.00343278,
        -0.85543966,  0.33553985]], dtype=float32)

.base_values =
array([2.0684865, 2.0684865, 2.0684865, ..., 2.0684865, 2.0684865,
       2.0684865], dtype=float32)

.data =
array([[   8.3252    ,   41.        ,    6.98412698, ...,    2.55555556,
          37.88      , -122.23      ],
       [   8.3014    ,   21.        ,    6.23813708, ...,    2.10984183,
          37.86      , -122.22      ],
       [   7.2574    ,   52.        ,    8.28813559, ...,    2.80225989,
          37.85      , -122.24      ],
       ...,
       [   1.7       ,   17.        ,    5.20554273, ...,    2.3256351 ,
          39.43      , -121.22      ],
       [   1.8672    ,   18.        ,    5.32951289, ...,    2.12320917,
          39.43      , -121.32      ],
       [   2.3886    ,   16.        ,    5.25471698, ...,    2.61698113,
          39.37      , -121.24      ]])
```

#04 SHAP Values

SHAP Value 시각화 – Waterfall Plot

```
# 첫 번째 예측에 대한 설명 시각화
shap.plots.waterfall(shap_values[0])
```



모델 예측 결과에 대한 각 feature의 기여도를
plot으로 볼 수 있음 (훈련데이터 전체 평균에 대하여)

예측값을 높게 해 주는 변수는 빨간색,
예측값을 낮게 해 주는 변수는 파란색으로 표시됨

이미지 출처: https://github.com/shap/shap

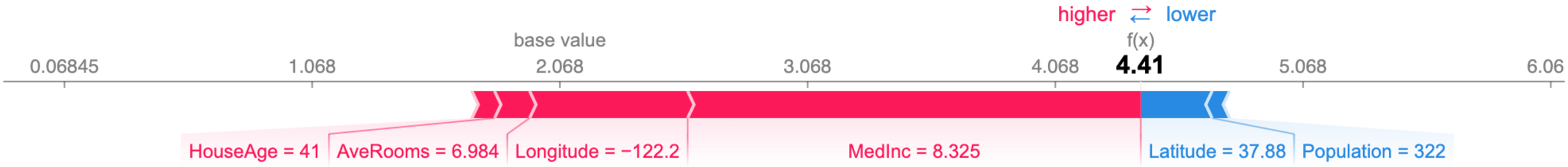
#04 SHAP Values

SHAP Value 시각화 – Force Plot

```
# 첫 번째 예측에 대한 설명 시각화 2
shap.plots.force(shap_values[0])
```

모델 결과에 대한 각 feature의 기여도를
plot으로 볼 수 있음 (특정 훈련데이터 하나에 대하여)

예측값을 높게 해 주는 변수는 빨간색,
예측값을 낮게 해 주는 변수는 파란색으로 표시됨



이미지 출처: https://github.com/shap/shap

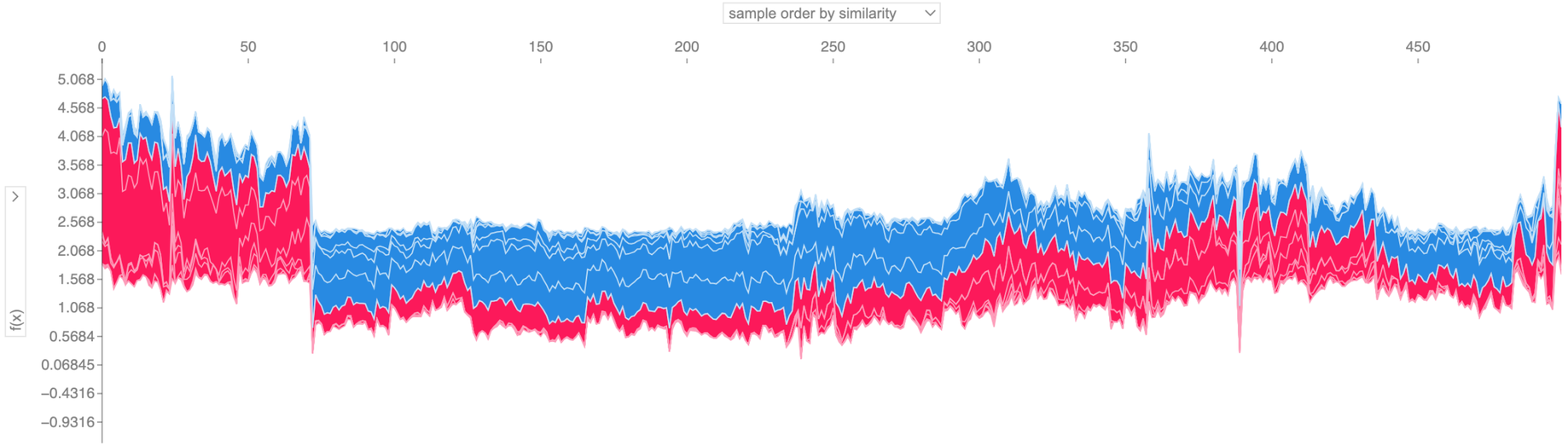
#04 SHAP Values

SHAP Value 시각화 – Force Plot

```
# 첫 번째 예측에 대한 설명 시각화 3
shap.plots.force(shap_values[:500])
```

모델 결과에 대한 각 feature의 기여도를
plot으로 볼 수 있음 (훈련데이터 전체에 대하여)

예측값을 높게 해 주는 변수는 빨간색,
예측값을 낮게 해 주는 변수는 파란색으로 표시됨

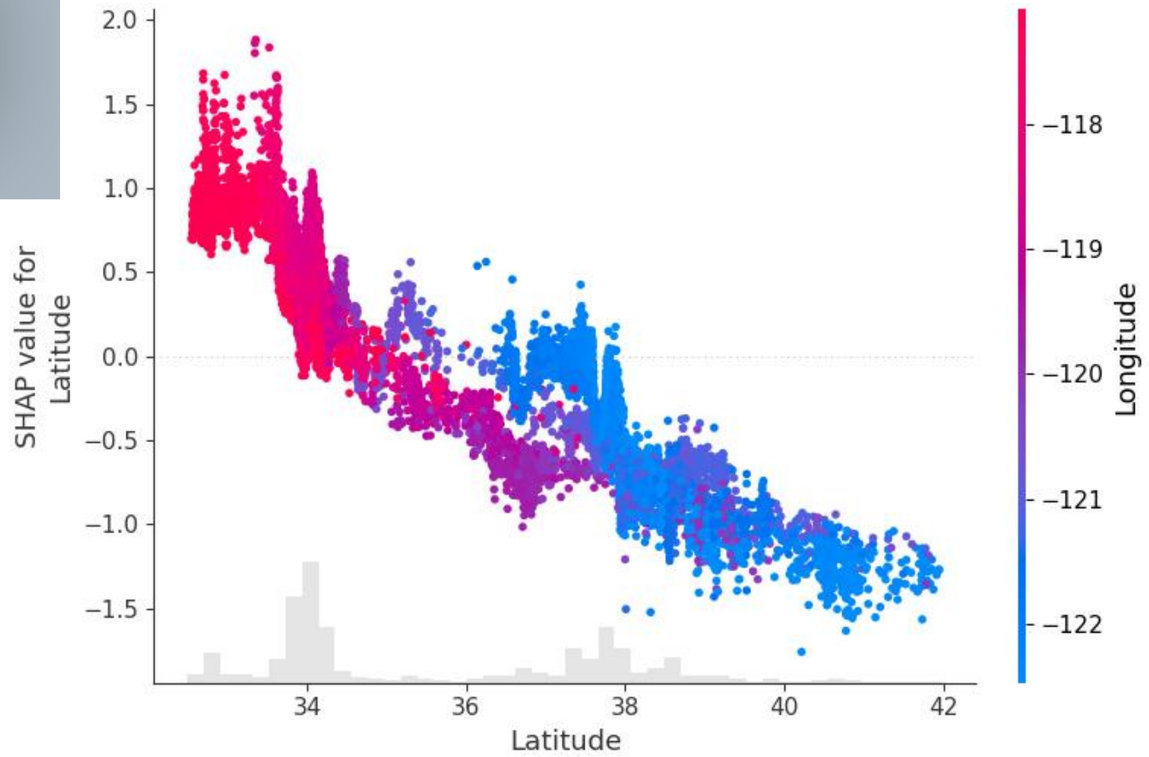


이미지 출처: https://github.com/shap/shap

#04 SHAP Values

SHAP Value 시각화 – Scatter Plot

```
# 전체 데이터 세트에서 단일 feature의 기여도를 표시하는 종속성 산점도 생성
shap.plots.scatter(shap_values[:, "Latitude"], color=shap_values)
```



특정 feature가 모델의 예측에 미치는 영향 시각화
다른 feature와의 상호작용 효과 또한 보여줌

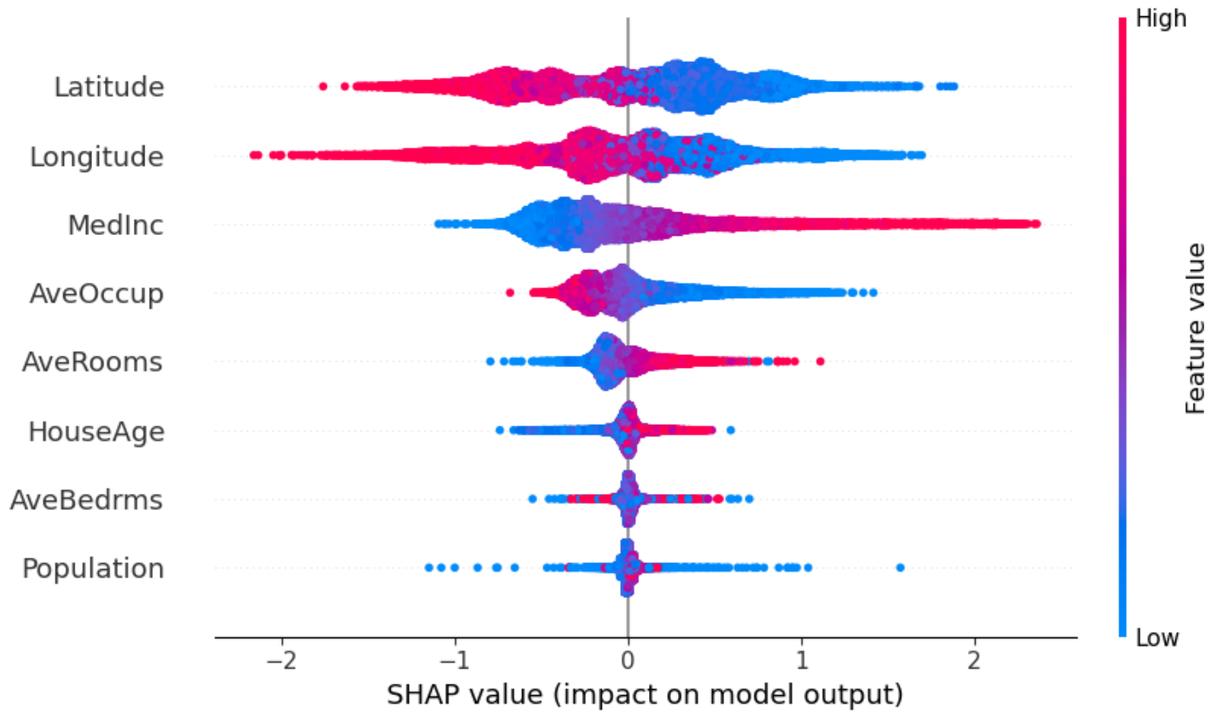
예측값을 높게 해 주는 변수는 빨간색,
예측값을 낮게 해 주는 변수는 파란색으로 표시됨

이미지 출처: https://github.com/shap/shap

#04 SHAP Values

SHAP Value 시각화 – Beeswarm Plot

```
# 모든 feature의 기여도 요약
shap.plots.beeswarm(shap_values)
```



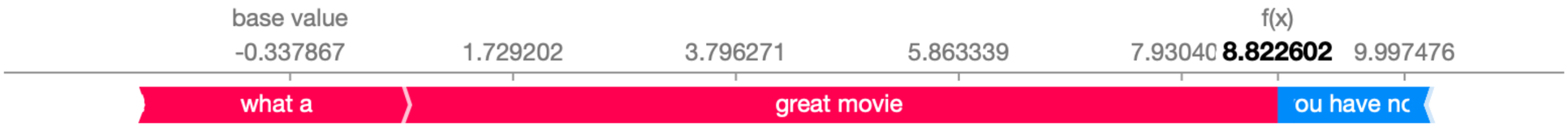
모든 sample에 대한 모든 feature의 SHAP Value
모델 결과에 미치는 영향이 가장 큰 feature가 가장 위에 배치됨 (모델에 가장 중요한 feature)

예측값을 높게 해 주는 변수는 빨간색,
예측값을 낮게 해 주는 변수는 파란색으로 표시됨

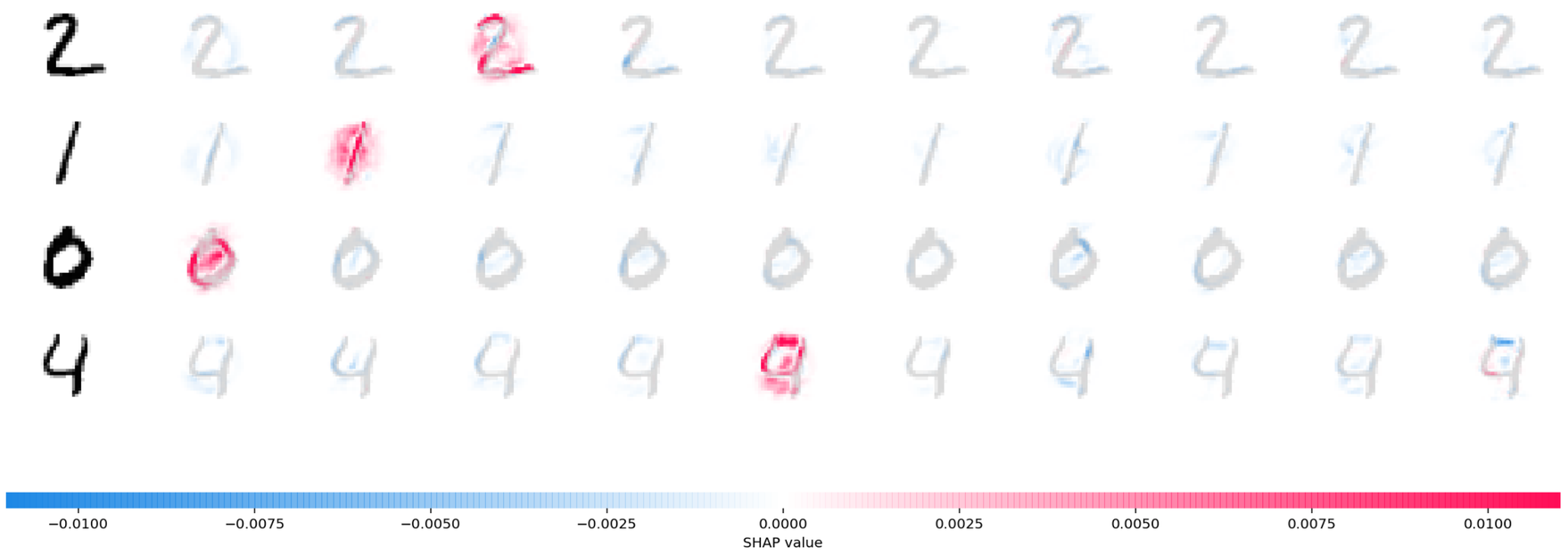
이미지 출처: https://github.com/shap/shap

#04 SHAP Values

SHAP Value 시각화 – NLP, Deep Learning



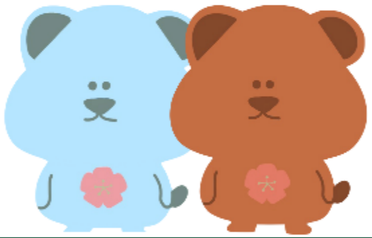
in NLP task…



in MNIST classification…

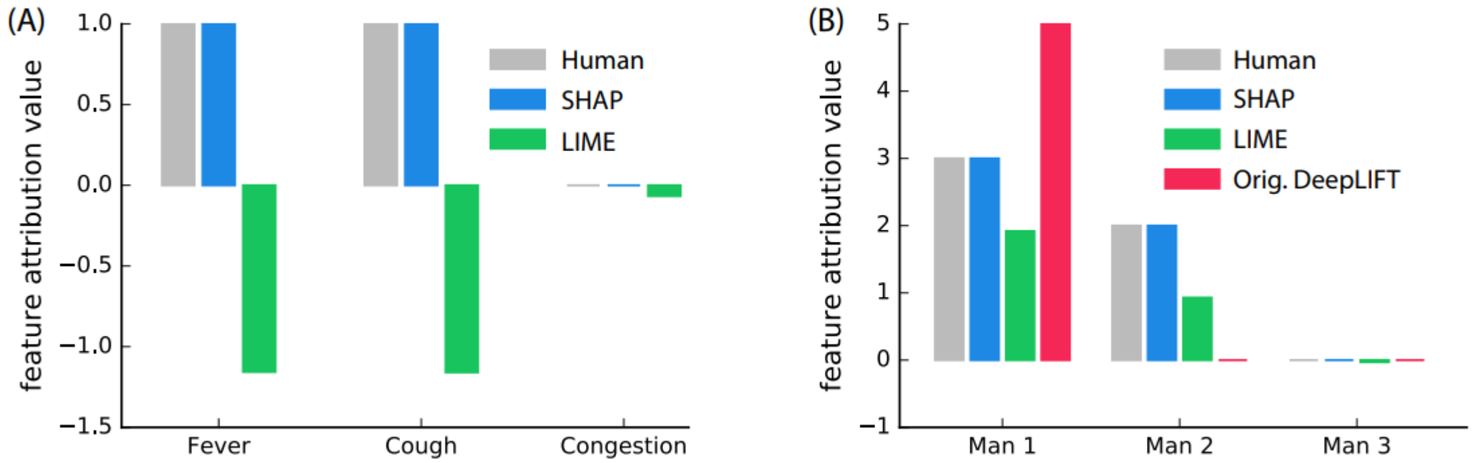
이미지 출처: https://github.com/shap/shap

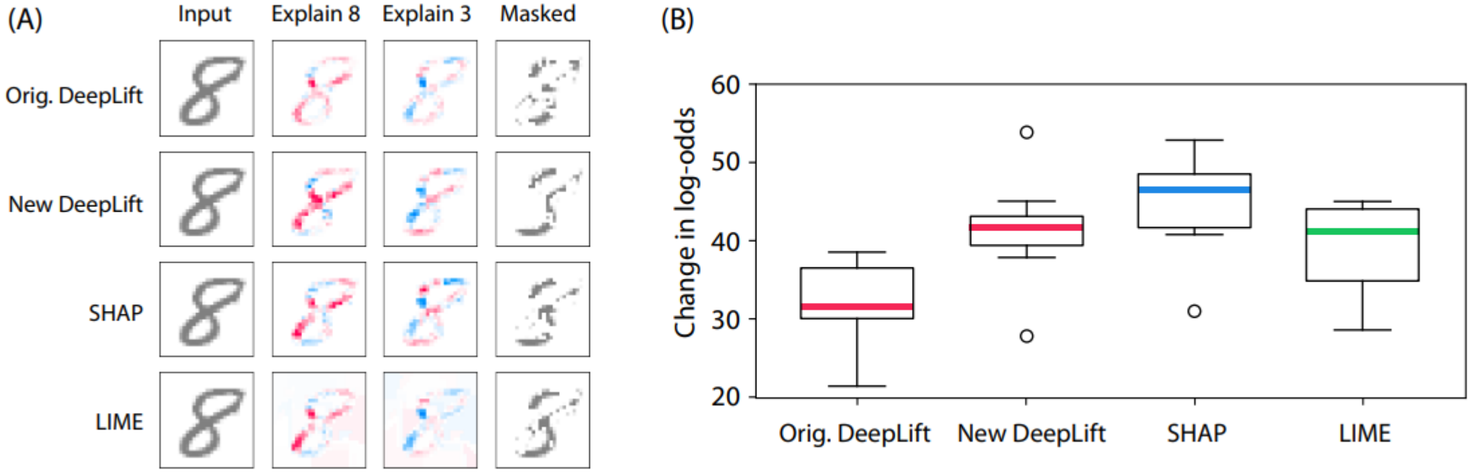
Advantages



EWHA EURON

#05 Advantages



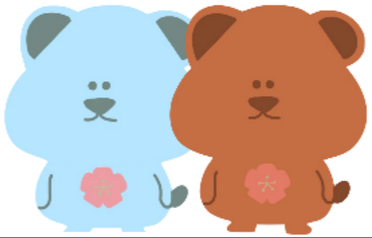


Kernel SHAP을 사용하면 계산 효율성을 높일 수 있으며, LIME과 비교했을 때 보다 높은 local accuracy와 consistency를 보임

또한 다른 방법보다 SHAP이 사람의 직관과 더 잘 부합하는 좋은 설명을 내놓음을 user study를 통해 확인함

Deep SHAP을 사용하면 이미지에서 클래스별 차이를 더 잘 설명할 수 있음

Conclusion

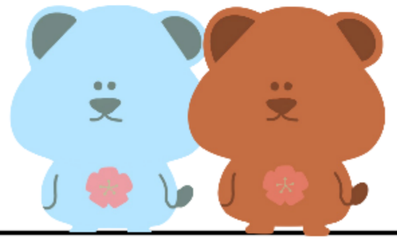


EWHA EURON

#06 Conclusion

- SHAP 프레임워크는 모델 예측의 정확성과 해석 가능성 간의 균형을 맞추기 위해 개발됨
- 모델을 해석하는 데 있어 중요한 property들을 정의하고, 이를 만족하는 공통적인 원칙을 제안함
- 다양한 SHAP Value Estimation 방법을 제시하고, 그 유용함을 증명함
- 향후 연구에서는 더 적은 가정을 필요로 하는 추정 방법이 개발되어야 할 것임

THANK YOU



EWHA EURON