SHAP의 주요 방법

2021.12.05 김은희

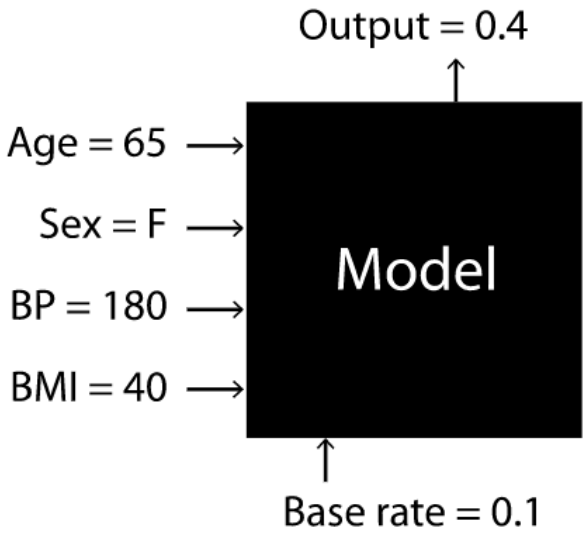
SHAP 리뷰

- Machine Learning, Deep Learning vs Regression
  - 성능이 좋고, 복잡하며, 해석 불가능(Black box)

  - Ensemble Tree의 feature importance?
    - 특정 결정을 하는데 해당 feature가 얼마나 기여했는지 설명 가능
    - 명확한 영향력을 설명할 수 없음

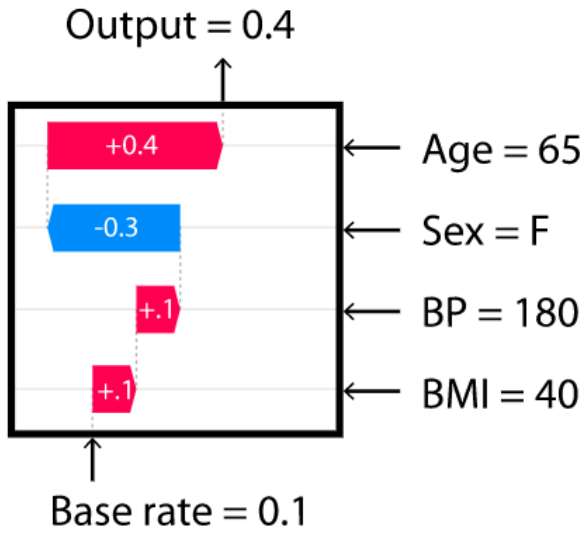
- 예측 결과에 대한 명확한 해석을 해 보자

SHAP의 아이디어



이런 black box 모델을

Explanation



각 요소별로 얼마나 기여했는지 설명
(baseline에 양의 영향력? 음의 영향력?)

KernelExplainer

- 각 feature의 중요성을 계산하기 위해 사용, 가중선형회귀모델의 계수를 반환
- 어떤 모델이든 사용 가능

Model output이 확률인 경우,
logit을 설정하면 feature importance 값이 log-odds* 값을 가지게 됨
(* 로그 승산 β = X가 한 단위 증가할 때 승산이 exp(β) 곱해짐)
승산 = prob of success / prob of failure

shap.KernelExplainer(model, data, link="identity 혹은 logit")

```
# use Kernel SHAP to explain test set predictions
explainer = shap.KernelExplainer(svm.predict_proba, X_train, link="logit")
shap_values = explainer.shap_values(X_test, nsamples=100)
```

DeepExplainer

- 딥러닝 모델에 SHAP을 적용
- Keras, TensorFlow 모델에 사용 가능 + PyTorch 예비 적용 중

전체 데이터를 사용하지 않고, 일부를 사용 (맨 처음 n개? 랜덤 n개?)

shap.DeepExplainer(model, data)

```
# select a set of background examples to take an expectation over
background = x_train[np.random.choice(x_train.shape[0], 100, replace=False)]

# explain predictions of the model on four images
e = shap.DeepExplainer(model, background)
# ...or pass tensors directly
# e = shap.DeepExplainer((model.layers[0].input, model.layers[-1].output), background)
shap_values = e.shap_values(x_test[1:5])
```

TreeExplainer

- 앙상블 트리 모델의 결과를 설명하기 위함
- XGBoost, LightGBM, CatBoost, scikit-learn, Pyspark, etc…

shap.TreeExplainer(model)

```
# train an XGBoost model
X, y = shap.datasets.boston()
model = xgboost.XGBRegressor().fit(X, y)

# explain the model's predictions using SHAP
# (same syntax works for LightGBM, CatBoost, scikit-learn, transformers, Spark, etc.)
explainer = shap.Explainer(model)
shap_values = explainer(X)
```

SHAP 공식 레포지토리

- https://github.com/slundberg/shap