R 기반 의학통계 및 머신러닝

박 승

충북대학교
CHUNGBUK NATIONAL UNIVERSITY

CHAPTER

06

특성 중요도 분석을 통한 의학적 해석

06 특성 중요도 분석을 통한 의학적 해석

1. 특성 중요도 분석의 중요성과 필요성

특성 중요도 분석 (Feature Importance Analysis)은 예측 모델이 특정 변수(특성)을 얼마나 중요하게 여기는지를 정량적으로 평가하는 과정

- 모델 해석 가능성 향상: 블랙박스 모델 (ex:딥러닝, XGboost)의 의사결정을 이해하고 설명할 수 있음

- 임상적 해석 및 의학적 의미 도출 가능: 예측 모델에서 어떤 변수가 가장 중요한 역할을 하는지 확인

- 모델 성능 최적화: 중요도가 낮은 변수를 제거하여 모델의 복잡도를 줄이고 과적합을 방지

- 의료 연구 및 정책 수립에 기여: 중요 변수를 바탕으로 임상적 표적 개입 전략을 수립함

06 특성 중요도 분석을 통한 의학적 해석

1. 특성 중요도 분석의 중요성과 필요성

특성 중요도 분석을 진행할 때는 아래와 같은 사항을 고려해야 함

** 변수 중요도가 항상 인과관계를 의미하지는 않음

- 특정 변수가 높은 중요도를 갖더라도, 이는 다른 숨은 변수에 의한 상관관계일 수 있음

⇒ 변수 간 상호작용을 고려해야 할 필요가 있음.

특히 단일 변수의 중요도를 해석할 때는 모델 내 변수와의 상관관계도 고려해야 함

SHAP(SHapley Additive exPlanations) 같은 기법을 활용하면 변수간 상호작용까지 분석 가능

** 데이터셋과 모델의 특성을 고려할 필요가 있음

- 동일한 변수를 사용하더라도 데이터셋에 따라 중요도가 다르게 나타날 수 있음

또한 동일한 데이터셋이라도 예측 모델의 구조에 따라 중요도 계산 방식이 달리 나타날 수 있음

06 특성 중요도 분석을 통한 의학적 해석

2. 특성 중요도 분석 실습

** 실습 전 필요한 라이브러리 목록

dplyr , caret, randomForest, xgboost, SHAPforxgboost, Matrix, ggplot2

(1) 로지스틱 회귀분석 (이전과 동일)

```
set.seed(42)
data <- read.csv("heart_disease_uci.csv")
data$target <- ifelse(data$num > 0, 1, 0)
data2 <- filter(data, if_all(everything(),
                ~!is.na(.) & .!=""))
data2sub <- subset(data2, select=c(-id, -num))
train_index <- createDataPartition(data2sub$target,
                                   p = 0.8, list = FALSE)
train_data <- data2sub[train_index, ]
test_data <- data2sub[-train_index, ]

logit_model <- glm(target ~ ., data = train_data,
                   family = binomial)
summary(step(logit_model))
```

```
Coefficients:
                          Estimate Std. Error z value Pr(>|z|)
(Intercept)             -0.979459   2.334699  -0.420  0.67483
sexMale                  1.426840   0.551281   2.588  0.00965 **
cpatypical angina       -1.618272   0.645080  -2.509  0.01212 *
cpnon-anginal           -2.026112   0.512845  -3.951 7.79e-05 ***
cptypical angina        -1.906438   0.647534  -2.944  0.00324 **
trestbps                 0.017675   0.011374   1.554  0.12019
chol                     0.005674   0.003960   1.433  0.15192
thalch                  -0.029912   0.010855  -2.756  0.00586 **
oldpeak                  0.366012   0.215910   1.695  0.09004 .
ca                       1.189106   0.275660   4.314 1.61e-05 ***
thalnormal              -0.172780   0.790287  -0.219  0.82694
thalreversable defect    1.526224   0.772848   1.975  0.04829 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

2. 특성 중요도 분석 실습

(1) 로지스틱 회귀분석 (ggplot 부분을 이전코드와 비교해볼 것)

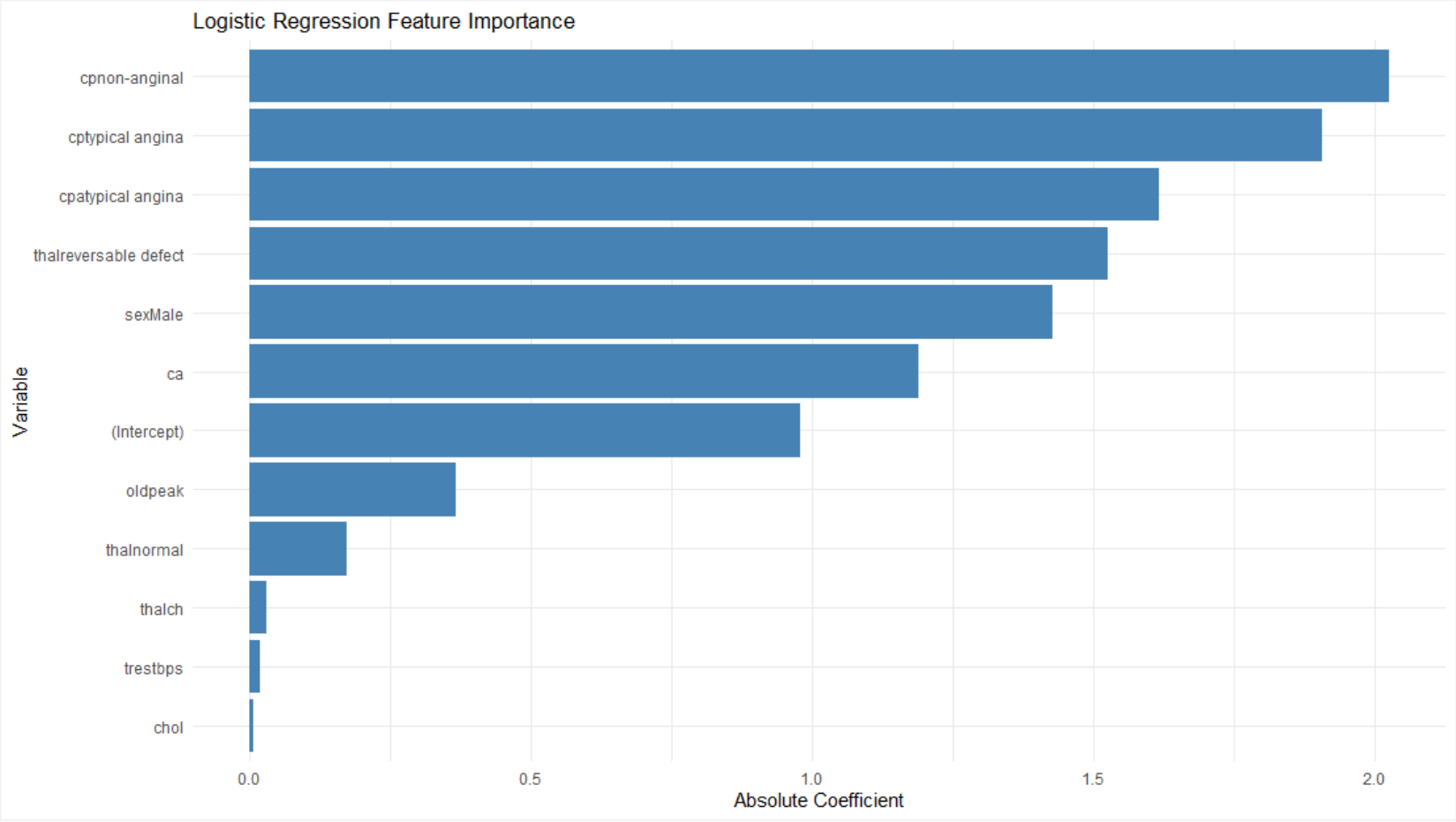
```
#계수들을 뽑아내어 데이터프레임화 하고 변수명을 맞춰줌
logit_importance <- as.data.frame(summary(step(logit_model))$coefficients)
logit_importance$Variable <- rownames(logit_importance)
colnames(logit_importance) <- c("Estimate", "Std. Error", "z value", "Pr(>|z|)", "Variable")

ggplot(logit_importance,
       aes(x = reorder(Variable, abs(Estimate)),
       y = abs(Estimate))) +
#변수명을 절대값 기준으로 정렬해서 x축에 표시
  회귀계수의 절대값을 y축에 표시
geom_bar(stat = "identity", fill = "steelblue") +  #막대그래프 생성. identity 옵션은 주어진 값 그대로 표시하는 옵션
coord_flip() +  #그래프 x축과 y축을 뒤집어 세로로
labs(title = "Logistic Regression Feature Importance",
       x = "Variable",
       y = "Absolute Coefficient") +
#제목, x축 이름, y축 이름
theme_minimal()
#배경 격자 제거
```

각 변수의 회귀계수의 절대값이 얼마나 큰지 보여주는 시각화

06 특성 중요도 분석을 통한 의학적 해석

2. 특성 중요도 분석 실습

(1) 로지스틱 회귀분석



2. 특성 중요도 분석 실습

(2) 랜덤 포레스트 (Random Forest)

여러 개의 의사 결정 나무(Decision Trees)를 조합하여 예측하는 앙상블 학습 방법

일반적인 의사 결정 나무는 과적합에 취약하지만, 랜덤 포레스트는 여러개의 트리를 활용하여 안정적임

학습 과정

- 데이터 샘플링(bootstrap Sampling)

  학습 데이터를 랜덤하게 샘플링하여 여러 개의 서브셋을 생성, 각 트리는 서로 다른 샘플을 사용하여 학습

- 무작위 특징 선택(Feature Randomness)

  트리를 분할할 때, 전체 변수 중 일부 변수만 랜덤하게 선택하여 최적의 분할 기준을 결정

  ⇒ 특정 변수가 과도하게 영향력을 행사하는 것을 방지하고 다양한 패턴을 학습 가능

- 다수결 투표(Voting) & 평균값 계산

  Classification: 여러 트리가 예측한 값 중 가장 많이 나온 값을 최종 예측값으로 선택(다수결)

  Regression: 여러 트리의 예측값을 평균 내어 최종 예측값으로 사용

2. 특성 중요도 분석 실습

(2) 랜덤 포레스트 (Random Forest)

장점: 과적합을 방지할 수 있음. 여러 개의 의사결정 나무의 조합은 일반화 성능을 향상시킴
고차원 데이터에서도 잘 작동함. 많은 변수가 있는 데이터에서 효과적
어떠한 변수가 예측에 중요한 지 정량적으로 측정 가능

단점: 트리 개수가 많아질수록 연산량 기하급수적 증가
단순 의사결정나무보다 어려운 해석
실시간 예측 속도 느림

2. 특성 중요도 분석 실습

(2) 랜덤 포레스트 (Random Forest)

변수 중요도 계산 방식

- 평균 정확도 감소 (Mean Decrease Accuracy) 방식

원래 데이터로 랜덤 포레스트를 학습하여 기준 정확도(Original Accuracy)를 측정

특정 변수의 값을 랜덤하게 섞음(Permutation Importance)

변수를 섞은 상태에서 다시 랜덤 포레스트를 학습하고 새로운 정확도(Permuted Accuracy)를 측정

두 정확도의 차이를 평균 내어 Mean Decrease Accuracy를 계산

$$\text{Mean Decrease Accuracy} = \frac{1}{N}\sum_{i=1}^{N}(\text{Original Accuracy} - \text{Permuted Accuracy})$$

N: 랜덤하게 변수를 섞는 횟수

**변수를 아예 제거하면 데이터 구조 자체가 달라져서 비교가 어려워짐. 변수를 섞으면 그 변수의 정보만 무작위화 되면서 모델이 실제로 변수에 얼마나 의존하는지 확인 가능

2. 특성 중요도 분석 실습

(2) 랜덤 포레스트 (Random Forest)

변수 중요도 계산 방식

- 평균 Gini 불순도 감소 (Mean Decrease Gini) 방식

특정 변수가 노드 분할(Split) 과정에서 Gini 불순도를 얼마나 줄이는지 측정

Gini 불순도는 노드가 얼마나 혼합되어 있는지를 나타냄(값이 낮을수록 순수한 분포)

$$Gini\ \text{index} = 1 - \sum_{i=1}^{C} p_i^2,\ \ p_i\text{:클래스 } i\text{에 속할 확률}$$

랜덤 포레스트를 학습하면서 각 트리에서 변수별 Gini Index를 기록

특정 변수를 기준으로 분할할 때, 해당 변수가 Gini 불순도를 얼마나 감소시키는지 합산

모든 트리에서 Gini 감소량을 평균 내어 Mean Decrease Gini 계산

$$Mean\ Decrease\ Gini = \frac{1}{T}\sum_{t=1}^{T}\sum_{s \in S_t} \Delta G(s)$$

T: 랜덤 포레스트의 트리 개수, S_t: 트리 t에서 특정 변수를 사용한 모든 분할(Split),

$\Delta G(s)$: 특정 변수 s를 사용하여 Gini 불순도가 감소한 양

2. 특성 중요도 분석 실습

(2) 랜덤 포레스트 (Random Forest)

```
train_data$target <- as.factor(train_data$target)
rf_model <- randomForest(target ~ ., data = train_data, importance = TRUE, ntree = 500)
rf_importance<- importance(rf_model)
```

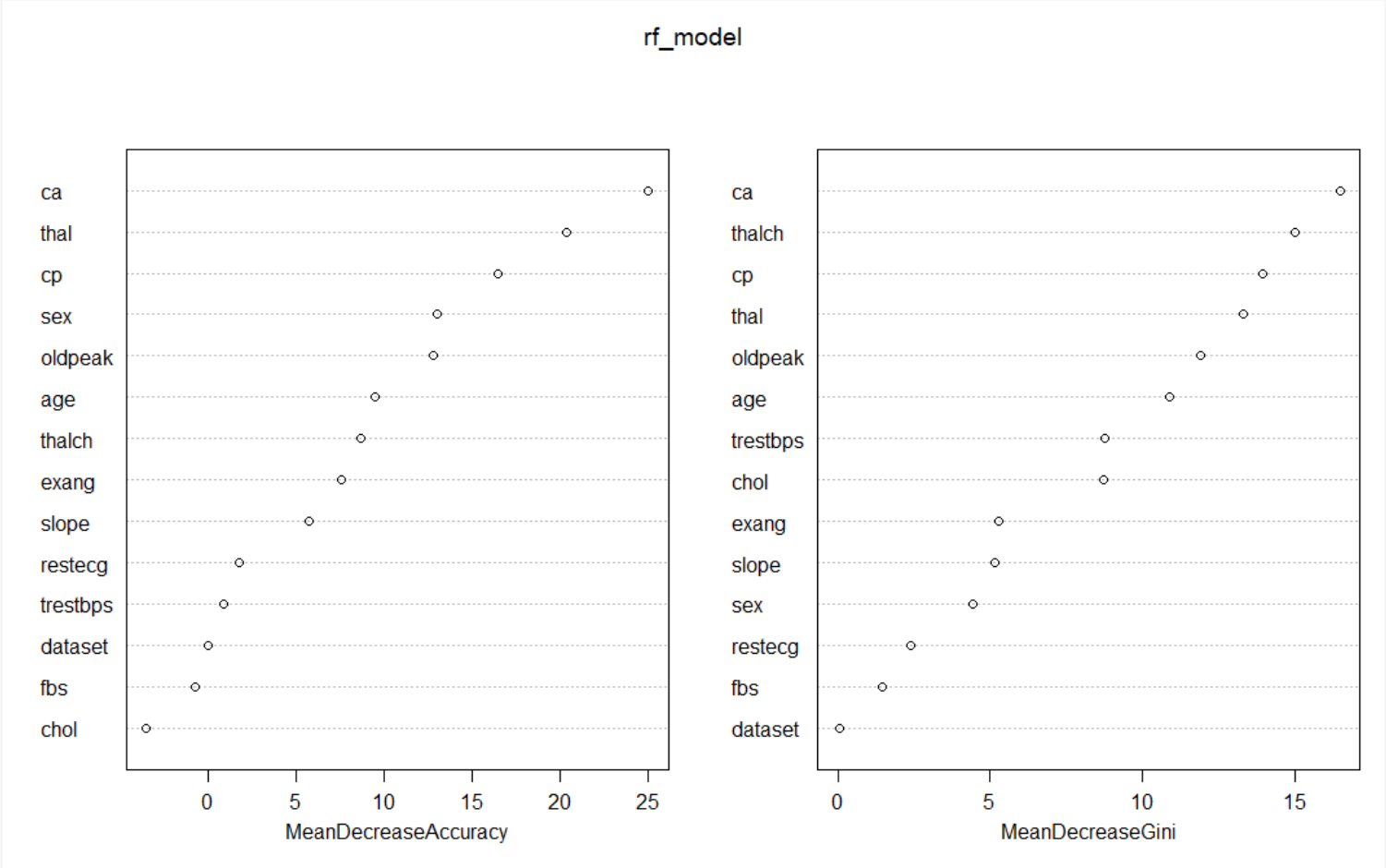
#Mean Deacrease Accuracy를 클래스 별로 분해한 값
이진 분류 문제에서, 각 클래스 별 별도의 변수 중요도를 의미

```
> rf_importance
                0          1 MeanDecreaseAccuracy MeanDecreaseGini
age      7.98509224   4.473983           9.5174452      10.86322164
sex     12.09344810   6.728731          13.0484643       4.42149754
dataset  0.00000000   0.000000           0.0000000       0.05575115
cp      13.44348944 12.110261          16.5050084      13.90337613
trestbps 0.37321276   1.085814           0.9025281       8.73864379
chol    -1.40817663 -3.787745          -3.5126326       8.72502872
fbs      0.03995804 -1.047028          -0.7199068       1.46309991
restecg  1.06653898   1.477225           1.7694057       2.39269919
thalch   6.85006200   5.285068           8.6981574      14.96835011
exang    1.44091001   8.857823           7.5912144       5.28087274
oldpeak  7.32496132 10.678866          12.7688493      11.89760285
slope    0.46240885   7.045169           5.7672541       5.13783646
ca      21.94981833 16.988113          25.0108594      16.45703302
thal    18.00327911 14.114945          20.3749488      13.26855090
```

2. 특성 중요도 분석 실습

(2) 랜덤 포레스트 (Random Forest)

```
varImpPlot(rf_model)
```



2. 특성 중요도 분석 실습

(3) XGboost

Gradient Boosting Machine을 개선한 알고리즘, 여러개의 약한 학습기(Weak Learner, 의사결정나무 등)를 순차적으로 학습하여 점진적으로 모델 성능을 개선하는 방식

내부적으로 병렬 처리(parallel processing)를 지원하여 학습 속도가 빠름

트리 가지치기(Pruning) 및 정규화로 과적합을 방지하여 높은 성능을 낼 수 있음

수백만 개의 데이터도 처리할 수 있도록 메모리 최적화 가능

데이터를 자동으로 학습하여 결측값을 자체적으로 처리 가능

다양한 손실함수를 지원으로 Classification, Regression, Learning to Rank 등 다양한 문제 해결 가능

2. 특성 중요도 분석 실습

(3) XGboost

알고리즘

- 초기 모델 생성

  첫 번째 결정 트리를 만들고, 해당 모델이 만든 예측값과 실제 값의 차이를 계산

- 오차(Residual)를 예측하는 새로운 트리 추가

  기존 모델의 오차를 최소화하는 방향으로 새로은 트리를 학습

- 각 트리의 가중치 업데이트

  Gradient Descent(경사하강법)를 통해 각 트리의 기여도를 조정

- 반복적으로 트리 추가(Boosting 과정)

  원하는 개수만큼(nrounds) 트리를 추가하여 최적의 예측 모델을 생성

2. 특성 중요도 분석 실습

(3) XGboost

장점

- 속도와 성능

  기존 Gradient Boosting Machine보다 빠르고, 성능도 뛰어남이 검증됨

- 과적합 방지 기능 내장

  L1, L2 정규화 (Ridge, Lasso), Tree Pruning 등으로 과적합을 효과적으로 방지

단점

- 복잡한 하이퍼파라미터 튜닝

  XGboost는 다양한 하이퍼파라미터(learning_rate, max_depth, n_estimator, subsample 등)를 제공하지만, 최적값을 찾기 어려움

- 트리 기반 모델이기 때문에 CNN(Convolution Neural Net)처럼 이미지 처리에는 적절하지 않음

2. 특성 중요도 분석 실습

(3) XGboost

```
train_matrix <- model.matrix(target ~ . - 1, data = train_data)
train_label <- as.numeric(as.character(train_data$target))
dtrain <- xgb.DMatrix(data = train_matrix, label = train_label)

params <- list(
  objective = "binary:logistic",
  eval_metric = "logloss",
  tree_method = "auto"
)

xgb_model <- xgb.train(params, dtrain, nrounds = 100, verbose = 0)
importance <- xgb.importance(model = xgb_model)
importance
xgb.plot.importance(importance)
```

#XGboost는 절편이 필요없으므로 제거

#XGboost는 종속변수가 수치형이어야 함

#XGboost는 xgb.DMatrix라는 자체 데이터 구조로 변환해야 함.
메모리 최적화 및 학습 속도 향상을 위한 것

#XGboost는 범주형 변수를 직접 처리하지 못함. model.matrix를 통해 범주형 변수를
one-hot-encoding 변환하여 모델 학습

2. 특성 중요도 분석 실습

(3) XGboost

model.matrix()에서 절편 삭제가 필요한 이유

```
train_matrix <- model.matrix(target ~ . - 1, data = train_data)
train_label <- as.numeric(as.character(train_data$target))
dtrain <- xgb.DMatrix(data = train_matrix, label = train_label)
```

model.matrix는 $y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \cdots + \beta_p X_{ip}$ 즉, $\mathrm{Y} = X\beta$ 형태로 구성

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_i \end{pmatrix} = \begin{pmatrix} 1 & X_{11} & X_{12} & & X_{1p} \\ 1 & X_{21} & X_{22} & \cdots & X_{2p} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & X_{i1} & X_{ip} & & X_{ip} \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{pmatrix}$$

model.matrix()를 사용하지 않으려면

```
dtrain <- xgb.DMatrix(data = as.matrix(train_data[, -which(names(train_data) == "target")]),
                      label = train_data$target)
```

이렇게 작성해도 동일

2. 특성 중요도 분석 실습

(3) XGboost

Gain

분할 시 해당 변수가 기여한 정보량

Cover

해당 변수를 사용한 데이터 샘플의 비율

Frequency

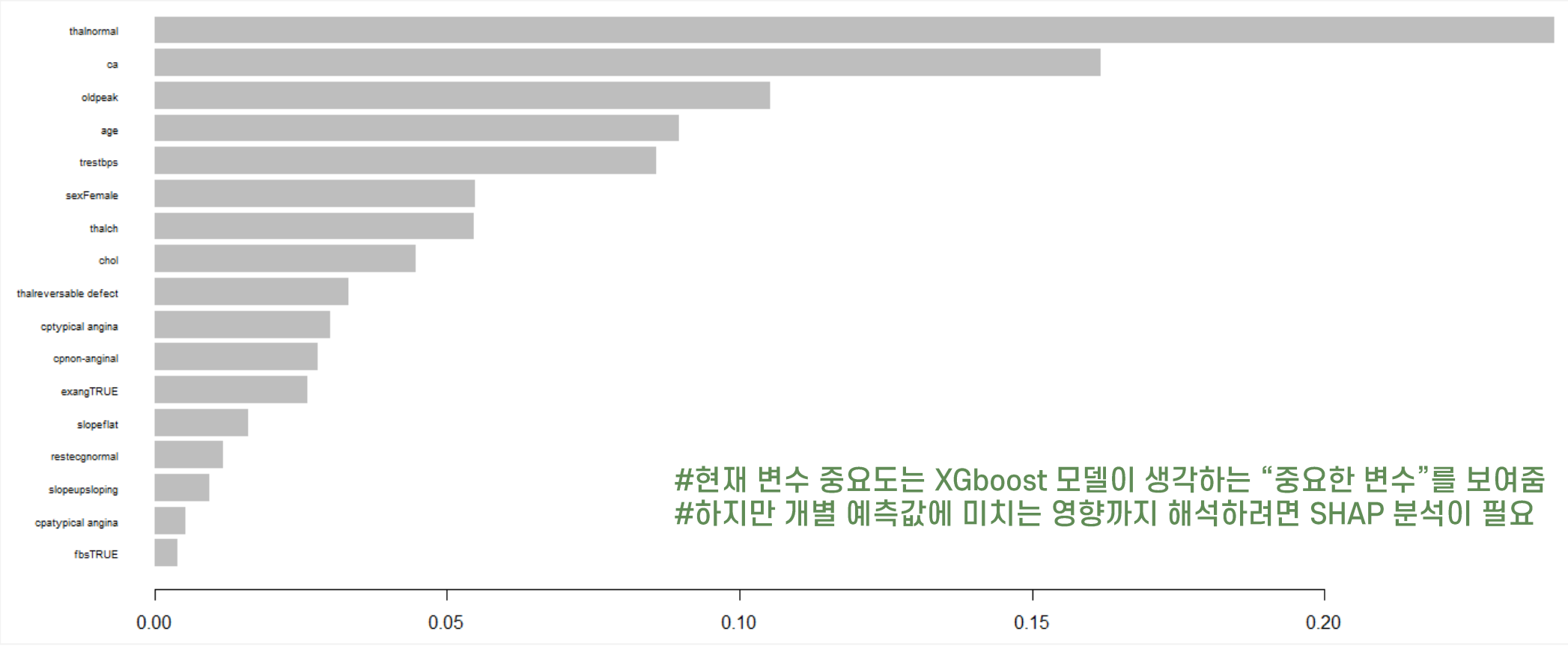
값이 클 수록 자주 등장한 변수

```
>  importance
                    Feature        Gain       Cover   Frequency
                     <char>       <num>       <num>       <num>
 1:             thalnormal 0.239381346 0.066407931 0.021156559
 2:                     ca 0.161868183 0.136630332 0.086036671
 3:                oldpeak 0.105262925 0.123037142 0.128349788
 4:                    age 0.089769241 0.114758539 0.156558533
 5:                trestbps 0.085867558 0.085681489 0.105782793
 6:              sexFemale 0.054893241 0.052844170 0.049365303
 7:                  thalch 0.054687836 0.105151415 0.132581100
 8:                    chol 0.044676500 0.074569874 0.105782793
 9: thalreversable defect 0.033124510 0.053427809 0.043723554
10:      cptypical angina 0.030045115 0.027995615 0.021156559
11:         cpnon-anginal 0.027899448 0.041561912 0.029619182
12:             exangTRUE 0.026243780 0.018669557 0.019746121
13:             slopeflat 0.016012908 0.033762992 0.025387870
14:          restecgnormal 0.011762552 0.025179257 0.033850494
15:        slopeupsloping 0.009372873 0.021823415 0.022566996
16:     cpatypical angina 0.005230221 0.006474729 0.007052186
17:               fbsTRUE 0.003901765 0.012023822 0.011283498
```

06 특성 중요도 분석을 통한 의학적 해석

2. 특성 중요도 분석 실습

(3) XGboost



2. 특성 중요도 분석 실습

(3) XGboost - SHAP

```
shap_values <- predict(xgb_model, dtrain, predcontrib = TRUE) #예측값이 아닌 SHAP값을 반환
shap_values <- as.data.frame(shap_values)
colnames(shap_values) <- c(colnames(train_matrix), "bias")
shap_values <- shap_values[, !colnames(shap_values) %in% "bias"]

shap_long <- shap.prep(shap_contrib = shap_values,
                       X = as.data.frame(train_matrix))

shap.plot.summary(shap_long)
```

#SHAP 값에는 각 변수 외에 bias term이 마지막에 포함되므로, 이를 지정해줌

#bias term을 분석하지 않으려면 제거

#shap.prep 함수로 long format 데이터로 변환
시각화를 위해 원래 학습데이터 train_matrix도 같이 전달

06 특성 중요도 분석을 통한 의학적 해석

2. 특성 중요도 분석 실습

(3) XGboost - SHAP

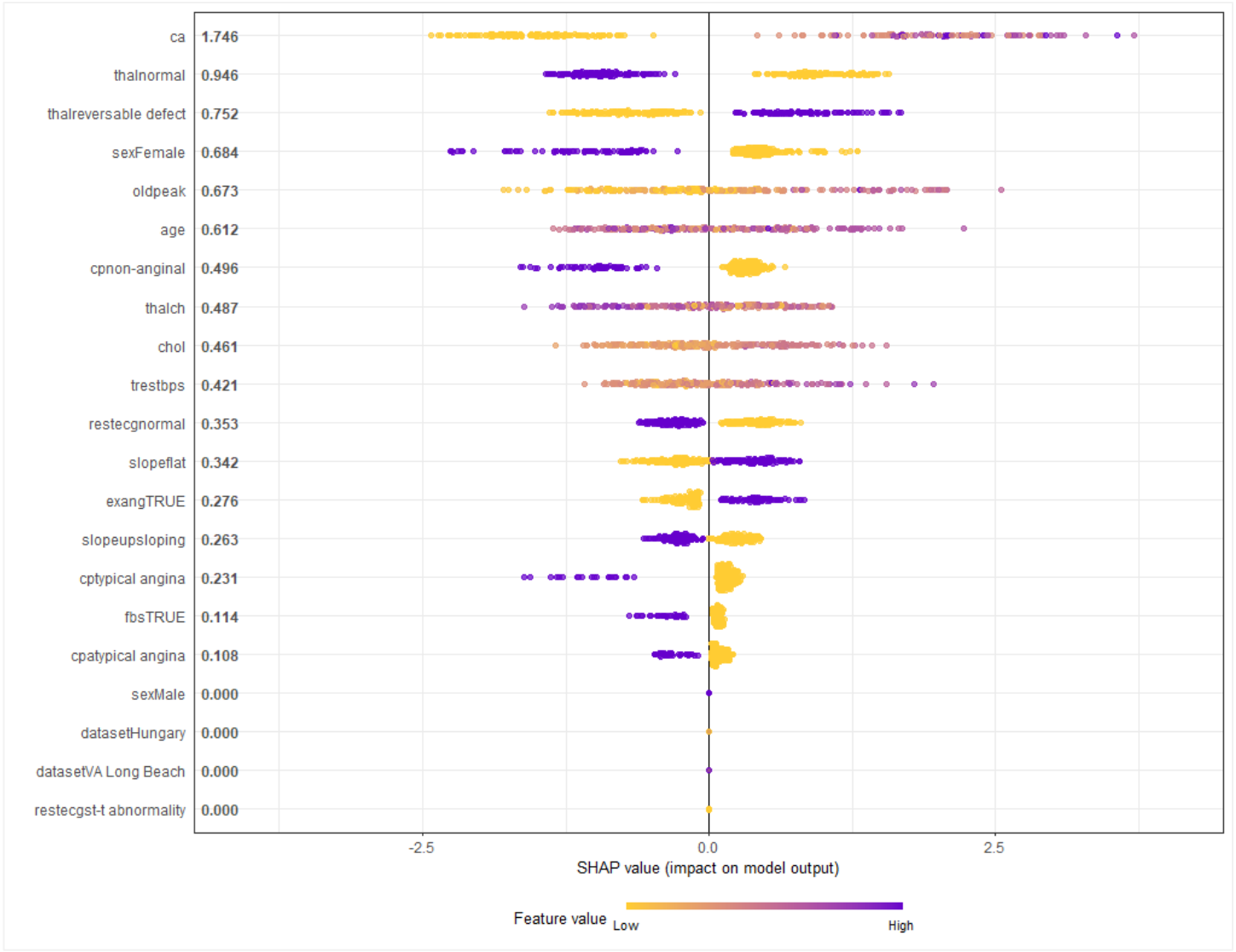
해석 :

X축이 양수로 갈수록 예측값을 증가
(사망 위험 증가)

보라색(Feature값이 큼)

⇒ 특정 변수가 클수록
예측값에 미치는 영향

ex) ca 변수는 값이 작을수록(노란색)
SHAP값이 감소(사망위험 감소)



2. 특성 중요도 분석 실습

(3) XGboost - SHAP

변수의 패턴 분석

ex) age에 따른 SHAP 값 변화 추이

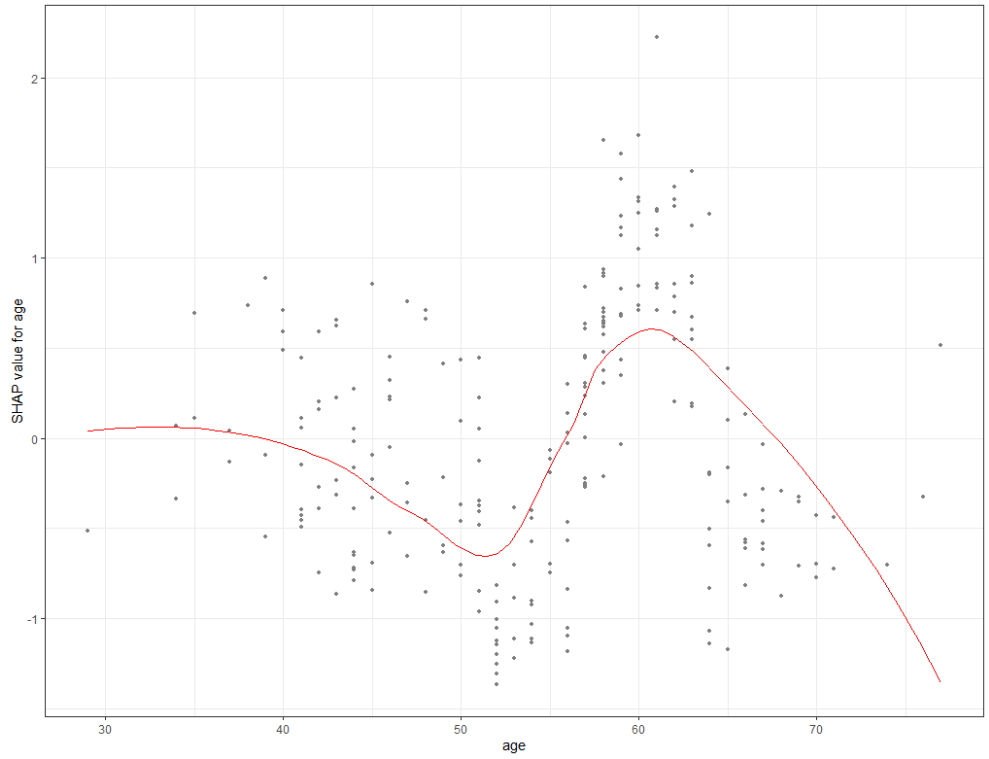
30~45 : SHAP값이 0 근처 ⇒ 사망 위험에 큰 영향이 없음

45~50 : SHAP값이 감소 ⇒ 나이가 증가할수록 사망 위험이 감소

50~60 : SHAP값이 증가 ⇒ 나이가 많아질수록 사망 위험이 증가

60 이후 : SHAP값이 하락 ⇒ 사망 위험에 대한 영향이 감소

```
shap.plot.dependence(data_long = shap_long, x = "age")
```



** 이러한 비선형적 패턴은 로지스틱 회귀에서는 잡아내기 어려운 부분이며, SHAP분석을 통해 변수간 상호작용을 고려한 해석이 가능해짐

06 특성 중요도 분석을 통한 의학적 해석

2. 특성 중요도 분석 실습

(3) XGboost - SHAP

변수의 상호작용

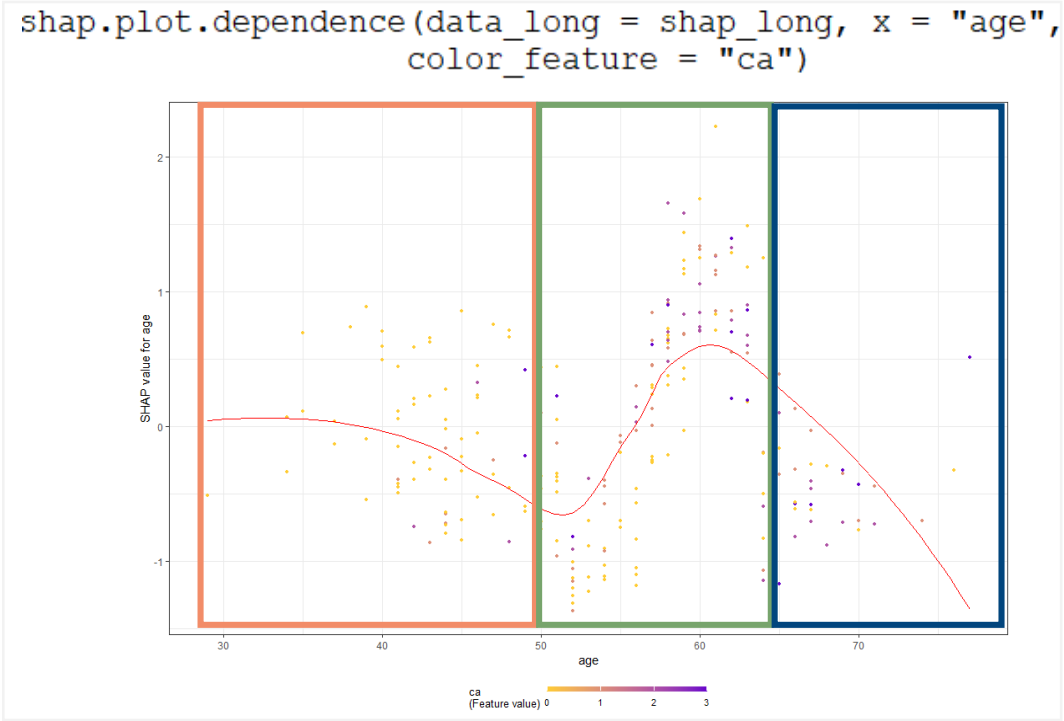
- 두 변수간의 관계를 통해 사망 위험 예측에서 어떻게 상호작용하는지 확인 가능

ex) age vs ca

age (30,50) 구간에서는 SHAP값이 대체로 낮아지면서 ca값이 대부분 노란색

age (50,65) 구간은 SHAP값이 급격히 증가하면서 ca값도 서서히 증가하는 추세를 보임

```
shap.plot.dependence(data_long = shap_long, x = "age",
                     color_feature = "ca")
```



age(65,) 구간에서는 SHAP값이 다시 하락하면서 ca값이 평균적으로 높아지는 것을 확인

**age와 ca의 특정 연령대에서 강한 상호작용을 보이며, 55~65세 구간에서 ca값이 높다면 사망위험이 매우 커질 가능성이 있다.

감사합니다

Q&A

충북대학교
CHUNGBUK NATIONAL UNIVERSITY