

# 재현가능한 데이터 사이언스(Data Science) - BMI

Reproducible Data Science

true

2019-08-29

## 비즈니스 설명

캐글, “500 Person Gender-Height-Weight-Body Mass Index - Height and Weight random generated, Body Mass Index Calculated”에서 데이터를 바탕으로 고객에게 체중과 키 정보만 제공하면 체질량 지수(Body Mass Index, BMI)를 예측하는 모델을 개발하여 고객이 궁금해하는 서비스를 개발하고자 한다.

체질량 지수(體質量指數, Body Mass Index, BMI)는 인간의 비만도를 나타내는 지수로, 체중과 키의 관계로 계산된다. 키가  $t$  미터, 체중이  $w$  킬로그램일 때, BMI는 다음 수식으로 표현된다. (키의 단위가 센티미터가 아닌 미터임에 유의해야 한다.)

$$BMI = \frac{w}{t^2}$$

체질량지수 (BMI지수)로 과체중 혹은 비만을 판정하는 한국 사례 <sup>1</sup>

구분	BMI 지수
고도 비만	40 이상
중등도 비만 (2단계 비만)	35 - 39.9
경도 비만 (1단계 비만)	30 - 34.9
과체중	25 - 29.9
정상	18.5 - 24.9
저체중	18.5 미만

## 데이터

캐글, “500 Person Gender-Height-Weight-Body Mass Index - Height and Weight random generated, Body Mass Index Calculated”에서 데이터를 바탕으로 고객에게 체중과 키 및 라벨 데이터 index가 준비되어 있어 키와 몸무게를 통해 BMI 예측한다.

## 데이터 사전

- Gender : Male / Female
- Height : Number (cm)
- Weight : Number (Kg)
- Index :
  - 0 : Extremely Weak
  - 1 : Weak
  - 2 : Normal
  - 3 : Overweight
  - 4 : Obesity

---

<sup>1</sup> 위키백과, “체질량 지수” (2019-04-22 접근함)

## 데이터 가져오기

캐글에서 내려받은 원본 데이터를 살펴본다.

```
library(tidyverse)
library(here)

file_path <- here("data", "500_Person_Gender_Height_Weight_Index.csv")

bmi_dat <- read_csv(file_path)

glimpse(bmi_dat)
```

Observations: 500

Variables: 4

\$ Gender <chr> "Male", "Male", "Female", "Female", "Male", "Male", "Ma...

\$ Height <dbl> 174, 189, 185, 195, 149, 189, 147, 154, 174, 169, 195, ...

\$ Weight <dbl> 96, 87, 110, 104, 61, 104, 92, 111, 90, 103, 81, 80, 10...

\$ Index <dbl> 4, 2, 4, 3, 3, 3, 5, 5, 3, 4, 2, 4, 3, 2, 2, 5, 5, 5, 5...

```
bmi_dat %>%
  DT::datatable()
```

Show 10 entries

Search:

	Gender	Height	Weight	Index
1	Male	174	96	4
2	Male	189	87	2
3	Female	185	110	4
4	Female	195	104	3
5	Male	149	61	3
6	Male	189	104	3
7	Male	147	92	5
8	Male	154	111	5
9	Male	174	90	3
10	Female	169	103	4

Showing 1 to 10 of 500 entries

Previous **1** 2 3 4 5 ... 50 Next

## 탐색적 데이터 분석

### 요약 통계

```
bmi_df <- bmi_dat %>%
  mutate(Index = factor(Index, levels = c(0,1,2,3,4,5), labels = c(" ", " ", " ", " ", " ", " ")),
         Gender = factor(Gender, levels = c("Male", "Female")))

bmi_df %>%
```

```
group_by(Index) %>%
  summarise(  = mean(Height),
              = mean(Weight))
```

```
# A tibble: 6 x 3
  Index
  <fct>    <dbl>    <dbl>
1      188.    51.7
2      185.    59.4
3      174.    69.1
4      176.    86.9
5      174.   108.
6      161.   133.
```

시각화

```
bmi_df %>%
  gather( , , -Gender, -Index) %>%
  ggplot(aes(x=Index, y= , fill=Gender)) +
  geom_boxplot(show.legend = FALSE) +
  facet_grid( ~ Gender, scales="free") +
  theme_bw(base_family = "AppleGothic") +
  labs(x="", y="",
       title="Height and Weight")
# title=" , ")
```

## 예측모형 - BMI 예측

$$\text{BMI 그룹} = f(\text{성별}, \text{키}, \text{몸무게}) + \epsilon$$

BMI 그룹: “극저체중”, “저체중”, “정상”, “과체중”, “비만”, “고도비만”

```
# 0. -----
library(caret)
library(doSNOW)

set.seed(777)

# 1. -----
# bmi_df

# 2. -----

# 3. -----
## 3.1.
num_cores <- parallel::detectCores()
start_time <- Sys.time()

cl <- makeCluster(num_cores, type = "SOCK")
registerDoSNOW(cl)

## 3.2. vs /
```

```

train_test_index <- createDataPartition(bmi_df$Index, p = 0.7, list = FALSE)

train <- bmi_df[train_test_index, ]
test <- bmi_df[-train_test_index, ]

## 3.3. / -----
cv_folds <- createMultiFolds(train$Index, k = 10, times = 5)
cv_ctrl <- trainControl(method = "cv", number = 10,
                        index = cv_folds,
                        verboseIter = TRUE)

## 3.2.
### ranger
gc_ranger_model <- train(Index ~., train,
                        method = "ranger",
                        tuneLength = 7,
                        trControl = cv_ctrl)

```

note: only 2 unique complexity parameters in default grid. Truncating the grid to 2 .

Aggregating results

Selecting tuning parameters

Fitting mtry = 2, splitrule = extratrees, min.node.size = 1 on full training set

```

# 4. -----
gc_pred_class <- predict(gc_ranger_model, newdata = test, type="raw")
## -----
bmi_conf <- confusionMatrix(gc_pred_class, test$Index)

bmi_conf$table

```

	Reference					
Prediction	3	0	0	0	0	0
	0	4	1	0	0	0
	0	2	15	1	0	0
	0	0	2	16	0	0
	0	0	2	3	38	4
	0	0	0	0	1	55

```
cat(" : ", scales::percent(bmi_conf$overall[["Accuracy"]]))
```

```
: 89.1%
```

```
stopCluster(cl)
```

## BMI 예측

```

bmi_test_dat <- tribble(
  ~"Gender", ~"Height", ~"Weight",
  "Male", 149, 61,
  "Female", 172, 67
)

```

```
predict(gc_ranger_model, newdata = bmi_test_dat, type="raw")
```

```
[1]
```

```
Levels:
```