Q2 (a) $\boxed{\text{MSE}}$

Let k be the number of x_i in R_{jm}

$$\gamma_{jm} = \underset{r}{\operatorname{argmin}} \sum_{x_i \in R_{jm}} (y_i - f_{m-1}(x_i) - r)^2$$

$$0 = \frac{\partial}{\partial r} \sum_{x_i \in R_{jm}} (y_i - f_{m-1}(x_i) - r)^2$$

$$0 = \sum_{x_i \in R_{jm}} 2(y_i - f_{m-1}(x_i) - r)$$

$$0 = \sum_{x_i \in R_{jm}} 2y_i - 2f_{m-1}(x_i) - 2r$$

$$0 = \sum_{x_i \in R_{jm}} 2y_i - 2f_{m-1}(x_i) - \sum_{x_i \in R_{jm}} 2r$$

$$0 = \sum_{x_i \in R_{jm}} 2y_i - 2f_{m-1}(x_i) - 2kr$$

$$2kr = \sum_{x_i \in R_{jm}} 2y_i - 2f_{m-1}(x_i)$$

$$r_{jm}^* = \frac{1}{2k} \sum_{x_i \in R_{jm}} 2y_i - 2f_{m-1}(x_i)$$

$$\gamma_{jm}^* = \frac{1}{k} \sum_{x_i \in R_{jm}} y_i - f_{m-1}(x_i)$$

2(a) $\boxed{\text{Binomial deviance}}$

$$\text{odds} = \frac{p}{1-p}$$

$$D = -2\,[y_i \cdot \log(p) + (1-y_i) \cdot \log(1-p)]$$

$$= -2\,[y_i \log(p) + \log(1-p) - y_i \log(1-p)]$$

$$= -2\,[y_i (\log(p) - \log(1-p)) + \log(1-p)]$$

$$= -2\,[y_i \log\left(\frac{p}{1-p}\right) + \log(1-p)]$$

$$= -2\,[y_i \log(\text{odds}) - \log(1-p)] \;*$$

$$= -2\,[y_i \log(\text{odds}) - \log(1+e^{\log(\text{odds})})]$$

*

$$\log(1-p) = \log\left(1 - \frac{e^{\log(\text{odds})}}{1+e^{\log(\text{odds})}}\right) = \log\left(\frac{1+e^{\log(\text{odds})}}{1+e^{\log(\text{odds})}} - \frac{e^{\log(\text{odds})}}{1+e^{\log(\text{odds})}}\right)$$

$$= \log\left(\frac{1}{1+e^{\log(\text{odds})}}\right)$$

$$= \log(1) - \log(1+e^{\log(\text{odds})})$$

$$= -\log(1+e^{\log(\text{odds})})$$

$$\frac{\partial}{\partial \log(\text{odds})} -2\,[y_i \log(\text{odds}) - \log(1+e^{\log(\text{odds})})]$$

$$= -2\,\left[y_i - \frac{e^{\log(\text{odds})}}{1+e^{\log(\text{odds})}}\right]$$

$$= -2\,[y_i - p]$$

$$\frac{\partial^2}{\partial^2 \log(odds)} -2\left[y_i \log(odds) - \log(1+e^{\log(odds)})\right]$$

$$= 2 \frac{\partial}{\partial^2 \log(odds)}\left[-y_i + \frac{e^{\log(odds)}}{1+e^{\log(odds)}}\right]$$

$$= 2\left[-(1+e^{\log(odds)})^{-2} e^{\log(odds)} \cdot e^{\log(odds)} + (1+e^{\log(odds)})^{-1} e^{\log(odds)}\right]$$

$$= 2\left[\frac{-e^{2\log(odds)}}{(1+e^{\log(odds)})^2} + \frac{e^{\log(odds)}}{(1+e^{\log(odds)})}\right]$$

$$= 2\left[\frac{-e^{2\log(odds)}}{(1+e^{\log(odds)})^2} + \frac{e^{\log(odds)}+e^{2\log(odds)}}{(1+e^{\log(odds)})^2}\right]$$

$$= 2\left[\frac{-e^{2\log(odds)}+e^{\log(odds)}+e^{2\log(odds)}}{(1+e^{\log(odds)})^2}\right]$$

$$= 2 \frac{e^{\log(odds)} \times 1}{(1+e^{\log(odds)})(1+e^{\log(odds)})}$$

$$= 2 \cdot \frac{e^{\log(odds)}}{1+e^{\log(odds)}} \cdot \frac{1}{1+e^{\log(odds)}}$$

$$= 2p(1-p)$$

$$L(y_i, F_{m-1}(x_i)+r) \overset{\text{Taylor expansion}}{\approx} L(y_i, F_{m-1}(x_i)+r) + \frac{\partial}{\partial F} L(y_i, F_{m-1}(x_i))\, r$$
$$+ \frac{1}{2}\frac{\partial^2}{\partial^2 F} L(y_i, F_{m-1}(x_i))\, r^2$$

$$\because\ r_{jm} = \underset{r}{\arg\min} \sum_{x_i \in R_{jm}} L(y_i, F_{m-1}(x_i)+r)$$

$$0 = \frac{\partial}{\partial r} \sum_{x_i \in R_{jm}} L(y_i, F_{m-1}(x_i)+r)$$

$$0 = \frac{\partial}{\partial r} \sum_{x_i \in R_{jm}} \Big\{ L(y_i, F_{m-1}(x_i)+r) + \frac{\partial}{\partial F} L(y_i, F_{m-1}(x_i))\, r$$
$$+ \frac{1}{2}\frac{\partial^2}{\partial^2 F} L(y_i, F_{m-1}(x_i))\, r^2 \Big\}$$

$$\sum_{x_i \in R_{jm}} \left[\frac{\partial}{\partial F} L(y_i, F_{m-1}(x_i)) + \frac{\partial^2}{\partial F^2} L(y_i, F_{m-1}(x_i)) r \right] = 0$$

$$r = \frac{-\sum_{x_i \in R_{jm}} \frac{\partial}{\partial F} L(y_i, F_{m-1}(x_i))}{\sum_{x_i \in R_{jm}} \frac{\partial^2}{\partial^2 F} L(y_i, F_{m-1}(x_i))}$$

$$= \frac{\sum_{x_i \in R_{jm}} 2(y_i - p)}{\sum_{x_i \in R_{jm}} 2p(1-p)}$$

$$= \frac{\sum_{x_i \in R_{jm}} (y_i - p)}{\sum_{x_i \in R_{jm}} p(1-p)}$$

$$\because L(y_i, F_{m-1}(x_i)) = -2 \left[y_i F_{m-1}(x_i) - \log(1 + e^{F_{m-1}(x_i)}) \right]$$

$$\therefore \frac{\partial}{\partial F} L(y_i, F_{m-1}(x_i)) = -2(y_i - p)$$

$$\frac{\partial^2}{\partial^2 F} L(y_i, F_{m-1}(x_i)) = 2p(1-p)$$

$$\log\left(\frac{p}{1-p}\right) = \log(\text{odds}) = F_{m-1}(x_i)$$

$$e^{\log\left(\frac{p}{1-p}\right)} = e^{F_{m-1}(x_i)}$$

$$\frac{p}{1-p} = e^{F_{m-1}(x_i)}$$

$$p = e^{F_{m-1}(x_i)} - e^{F_{m-1}(x_i)} p$$

$$(1 + e^{F_{m-1}(x_i)}) p = e^{F_{m-1}(x_i)}$$

$$p = \frac{e^{F_{m-1}(x_i)}}{1 + e^{F_{m-1}(x_i)}}$$

$$r_{jm}^* = \frac{\sum_{x_i \in R_{jm}} (y_i - p)}{\sum_{x_i \in R_{jm}} p(1-p)} \quad \text{where} \quad p = \frac{e^{F_{m-1}(x_i)}}{1 + e^{F_{m-1}(x_i)}}$$

2(b) $$Obj(\theta)^t = \sum_{i=1}^{n} L(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) + \text{Constant}$$

where $$\Omega(f_t) = rT + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2$$

Let $$g_i = \frac{\partial}{\partial \hat{y}^{(t-1)}} L(y_i, \hat{y}^{(t-1)}) \qquad h_i = \frac{\partial^2}{\partial^2 \hat{y}^{(t-1)}} L(y_i, \hat{y}^{(t-1)})$$

Using taylor approximation:

$$Obj(\theta)^t \approx \sum_{i=1}^{n} [\underbrace{L(y_i, \hat{y}_i^{(t-1)})}_{\text{constant}} + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t) + C$$

$$\propto \sum_{i=1}^{n} [g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t)$$

$$= \sum_{i=1}^{n} [g_i w_{L(x_i)} + \frac{1}{2} h_i w_{L(x_i)}^2] + rT + \lambda \frac{1}{2} \sum_{j=1}^{T} w_j^2$$

$$= \sum_{j=1}^{T} [(\sum_{i \in I_j} g_i) w_j + \frac{1}{2} (\sum_{i \in I_j} h_i + \lambda) w_j^2] + rT$$

$$\boxed{\begin{aligned} &\sum_{j=1}^{T} (\Sigma_{i \in I_j} h_i + \lambda) w_j^2 \\ &= \sum_{j=1}^{T} \frac{1}{2} (\Sigma_{i \in I_j} h_i) w_j^2 + \frac{1}{2} \lambda w_j^2 \\ &= \sum_{j=1}^{T} \frac{1}{2} (\Sigma_{i \in I_j} h_i + \lambda) w_j^2 \end{aligned}}$$

Let $G_j = \Sigma_{i \in I_j} g_i$

$H_j = \Sigma_{i \in I_j} h_i$

$$\therefore \hat{Obj}(\theta)^t = \sum_{j=1}^{T} [G_j w_j + \frac{1}{2}(H_j + \lambda) w_j^2] + rT$$

$$\frac{\partial \hat{Obj}(\theta)^t}{\partial w_j} = G_j + (H_j + \lambda) w_j = 0$$

$$w_j = -\frac{G_j}{H_j + \lambda}$$

- MSE :

$$g_i = \frac{\partial}{\partial \hat{y}^{(t-1)}} (y_i - \hat{y}^{(t-1)})^2$$

$$= -2 (y_i - \hat{y}^{(t-1)})$$

$$h_i = \frac{\partial^2}{\partial^2 \hat{y}^{(t-1)}} (y_i - \hat{y}^{(t-1)})^2$$

$$= \frac{\partial}{\partial \hat{y}^{(t-1)}} -2 y_i + 2 \hat{y}^{(t-1)}$$

$$= 2$$

$$G_j = \Sigma_{i \in I_j} -2 (y_i - \hat{y}^{(t-1)})$$

$$H_j = \Sigma_{i \in I_j} 2$$

$$w_j = - \frac{\Sigma_{i \in I_j} -2 (y_i - \hat{y}^{(t-1)})}{\Sigma_{i \in I_j} 2 + \lambda}$$

- Binomial deviance :

$$\log\left(\frac{p}{1-p}\right) = \log(\text{odds}) = \hat{y}^{(t-1)}$$

$$p = \frac{e^{\hat{y}^{(t-1)}}}{1+e^{\hat{y}^{(t-1)}}}$$

$$g_i = -2(y_i - p)$$

$$h_i = 2p(1-p)$$

$$G_j = \sum_{i \in I_j} -2(y_i - p)$$

$$H_j = \sum_{i \in I_j} 2p(1-p)$$

$$w_j = -\frac{\sum_{i \in I_j} -2(y_i - p)}{\sum_{i \in I_j} 2p(1-p) + \lambda} \quad \text{where } p = \frac{e^{\hat{y}^{(t-1)}}}{1+e^{\hat{y}^{(t-1)}}}$$

A4_juh3

2023-04-04

Question1

```
# load package and data
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────────────── tidy
verse 2.0.0 ──
## ✔ dplyr     1.1.1     ✔ readr     2.1.4
## ✔ forcats   1.0.0     ✔ stringr   1.5.0
## ✔ ggplot2   3.4.1     ✔ tibble    3.2.1
## ✔ lubridate 1.9.2     ✔ tidyr     1.3.0
## ✔ purrr     1.0.1
## ── Conflicts ──────────────────────────────────────────────────────────
───── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the  ]8;;http://conflicted.r-lib.org/ conflicted package ]8;;  to force all conflic
ts to become errors
```

```
library(tidymodels)
```

```
## ── Attaching packages ─────────────────────────────────────────────────
───── tidymodels 1.0.0 ──
## ✔ broom        1.0.4     ✔ rsample      1.1.1
## ✔ dials        1.1.0     ✔ tune         1.0.1
## ✔ infer        1.0.4     ✔ workflows    1.1.3
## ✔ modeldata    1.1.0     ✔ workflowsets 1.0.0
## ✔ parsnip      1.0.4     ✔ yardstick    1.1.0
## ✔ recipes      1.0.5
## ── Conflicts ──────────────────────────────────────────────────────────
──── tidymodels_conflicts() ──
## ✖ scales::discard() masks purrr::discard()
## ✖ dplyr::filter()   masks stats::filter()
## ✖ recipes::fixed()  masks stringr::fixed()
## ✖ dplyr::lag()      masks stats::lag()
## ✖ yardstick::spec() masks readr::spec()
## ✖ recipes::step()   masks stats::step()
## • Search for functions across packages at https://www.tidymodels.org/find/
```

```
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
library(reshape2)
```

```
##
## Attaching package: 'reshape2'
##
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
library(vip)
```

```
##
## Attaching package: 'vip'
##
## The following object is masked from 'package:utils':
##
##     vi
```

```
tidymodels_prefer()
hd <- (
  read_csv('heart.csv',show_col_types = FALSE)
)
```

Description:

I found this dataset in kaggle which is a dataset for heart attack classification. This Heat Disease Dataset was compiled in 1988 and comprises four separate databases: Cleveland, Hungary,S witzerland, and Long Beach V. It comprises 76 different attributes, including the predicted attribute, but in all published experiments, only a subset of 14 attributes were utilized.

Initial investigation:

The attribute name is defined as following:

(1). Age : Age of the patient

(2). Sex : Sex of the patient

(3). exang: exercise induced angina (1 = yes; 0 = no)

(4). ca: number of major vessels (0-3)

(5). cp : Chest Pain type chest pain type

Value 1: typical angina Value 2: atypical angina Value 3: non-anginal pain Value 4: asymptomatic

(6). trtbps : resting blood pressure (in mm Hg)

(7). chol : cholestoral in mg/dl fetched via BMI sensor

(8). fbs : (fasting blood sugar > 120 mg/dl) (1 = true; 0 = false)

(9). rest_ecg : resting electrocardiographic results

Value 0: normal Value 1: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV) Value 2: showing probable or definite left ventricular hypertrophy by Estes' criteria

(10). thalach : maximum heart rate achieved

(11). slp

(12). caa

(13). thall

(14). target : 0= less chance of heart attack 1= more chance of heart attack

```
# summary of data
hd |> glimpse() |> summary()
```

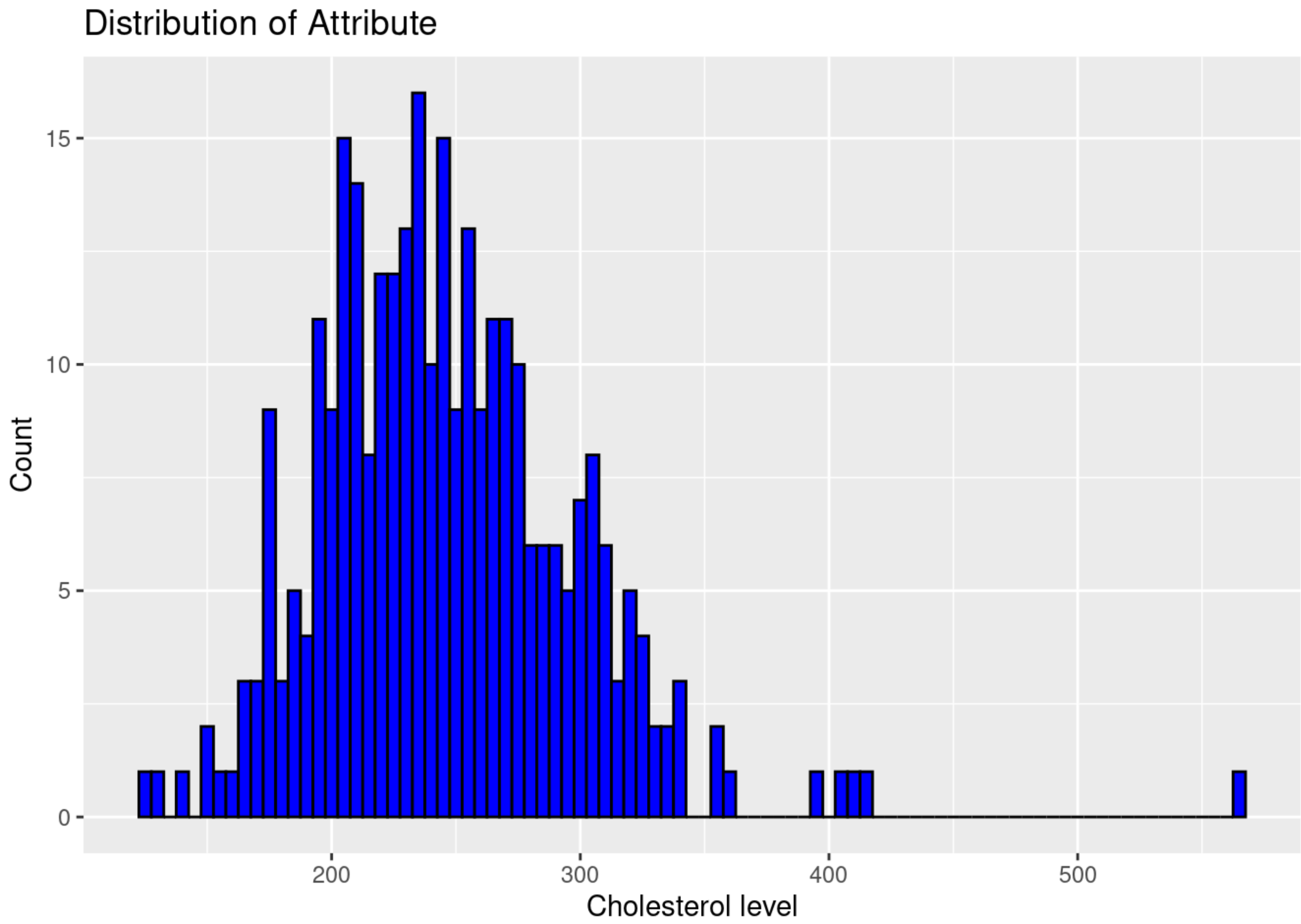
```
## Rows: 303
## Columns: 14
## $ age      <dbl> 63, 37, 41, 56, 57, 57, 56, 44, 52, 57, 54, 48, 49, 64, 58, 5…
## $ sex      <dbl> 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 1…
## $ cp       <dbl> 3, 2, 1, 1, 0, 0, 1, 1, 2, 2, 0, 2, 1, 3, 3, 2, 2, 3, 0, 3, 0…
## $ trtbps   <dbl> 145, 130, 130, 120, 120, 140, 140, 120, 172, 150, 140, 130, 1…
## $ chol     <dbl> 233, 250, 204, 236, 354, 192, 294, 263, 199, 168, 239, 275, 2…
## $ fbs      <dbl> 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0…
## $ restecg  <dbl> 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1…
## $ thalachh <dbl> 150, 187, 172, 178, 163, 148, 153, 173, 162, 174, 160, 139, 1…
## $ exng     <dbl> 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0…
## $ oldpeak  <dbl> 2.3, 3.5, 1.4, 0.8, 0.6, 0.4, 1.3, 0.0, 0.5, 1.6, 1.2, 0.2, 0…
## $ slp      <dbl> 0, 0, 2, 2, 2, 1, 1, 2, 2, 2, 2, 2, 2, 1, 2, 1, 2, 0, 2, 2, 1…
## $ caa      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0…
## $ thall    <dbl> 1, 2, 2, 2, 2, 1, 2, 3, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3…
## $ output   <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1…
```

```
##       age             sex               cp            trtbps     
##  Min.   :29.00   Min.   :0.0000   Min.   :0.000   Min.   : 94.0  
##  1st Qu.:47.50   1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:120.0  
##  Median :55.00   Median :1.0000   Median :1.000   Median :130.0  
##  Mean   :54.37   Mean   :0.6832   Mean   :0.967   Mean   :131.6  
##  3rd Qu.:61.00   3rd Qu.:1.0000   3rd Qu.:2.000   3rd Qu.:140.0  
##  Max.   :77.00   Max.   :1.0000   Max.   :3.000   Max.   :200.0  
##       chol            fbs            restecg          thalachh    
##  Min.   :126.0   Min.   :0.0000   Min.   :0.0000   Min.   : 71.0  
##  1st Qu.:211.0   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:133.5  
##  Median :240.0   Median :0.0000   Median :1.0000   Median :153.0  
##  Mean   :246.3   Mean   :0.1485   Mean   :0.5281   Mean   :149.6  
##  3rd Qu.:274.5   3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.:166.0  
##  Max.   :564.0   Max.   :1.0000   Max.   :2.0000   Max.   :202.0  
##       exng           oldpeak          slp             caa        
##  Min.   :0.0000   Min.   :0.00   Min.   :0.000   Min.   :0.0000  
##  1st Qu.:0.0000   1st Qu.:0.00   1st Qu.:1.000   1st Qu.:0.0000  
##  Median :0.0000   Median :0.80   Median :1.000   Median :0.0000  
##  Mean   :0.3267   Mean   :1.04   Mean   :1.399   Mean   :0.7294  
##  3rd Qu.:1.0000   3rd Qu.:1.60   3rd Qu.:2.000   3rd Qu.:1.0000  
##  Max.   :1.0000   Max.   :6.20   Max.   :2.000   Max.   :4.0000  
##      thall           output      
##  Min.   :0.000   Min.   :0.0000  
##  1st Qu.:2.000   1st Qu.:0.0000  
##  Median :2.000   Median :1.0000  
##  Mean   :2.314   Mean   :0.5446  
##  3rd Qu.:3.000   3rd Qu.:1.0000  
##  Max.   :3.000   Max.   :1.0000
```

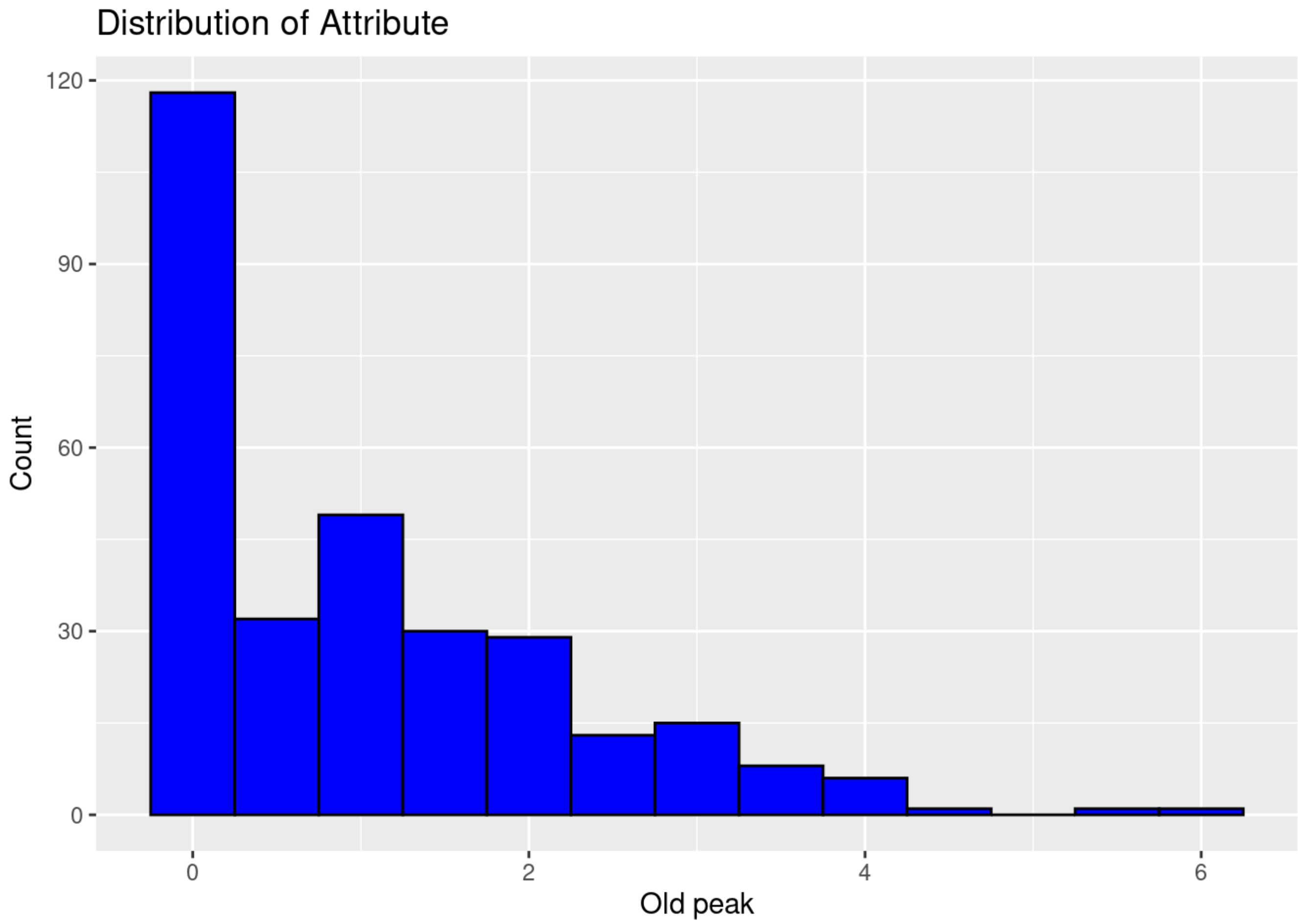
Firstly, there are 8 categorical attributes (sex, cp, fbs, restecg, exng, slp, caa, thall) and 5 numerical attributes (age, trtbps, chol, thalachh, oldpeak). There are some unusual:

1. The average cholesterol level (chol) is 246.3 and with 1st Qu. 211.0 and 3rd Qu.274.5, but the maximum is 564.0, this could be a outlier.
2. The maximum heart rate (thalachh) is 202.0 which is unusual.

```
# distribution of cholesterol level
hd |> ggplot(aes(x = chol)) +
  geom_histogram(binwidth = 5, fill = "blue", color = "black") +
  labs(x = "Cholesterol level", y = "Count", title = "Distribution of Attribute")
```



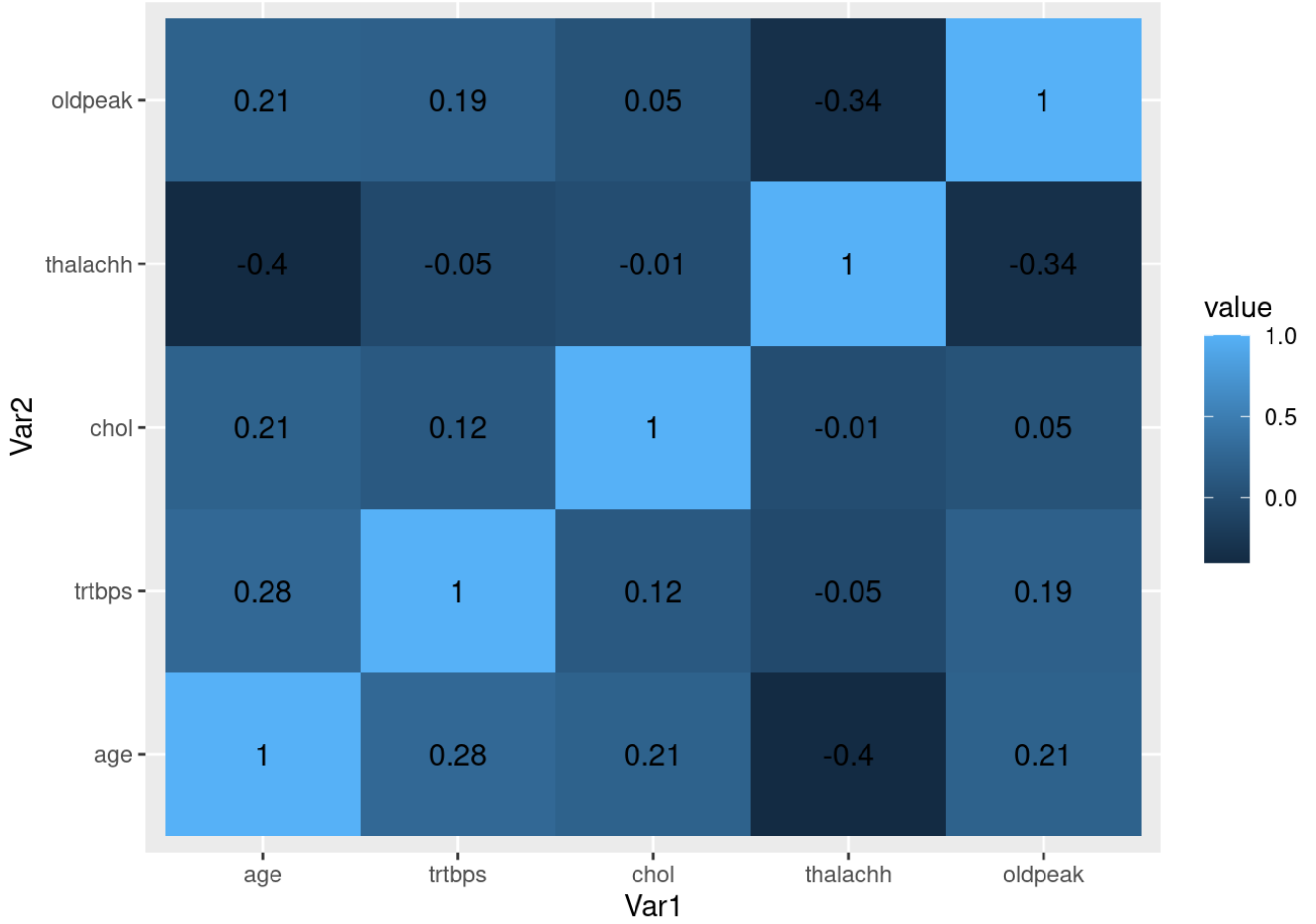

```
# distribution of old peak
hd |> ggplot(aes(x = oldpeak)) +
  geom_histogram(binwidth = 0.5, fill = "blue", color = "black") +
  labs(x = "Old peak", y = "Count", title = "Distribution of Attribute")
```




```
# heatmap for the correlation
hd_num <- hd[c('age','trtbps', 'chol', 'thalachh', 'oldpeak')]
hd_num_tbl <- as_tibble(hd_num)

corr_matrix <- hd_num_tbl |>
  cor()
corr_matrix <- round(corr_matrix,2)

ggplot(data = melt(corr_matrix), aes(x=Var1, y=Var2, fill=value)) +
  geom_tile() +
  geom_text(aes(Var2, Var1, label = value),
          color = "black", size = 4)
```



We can see cholesterol level is normally distributed and most have 0 old peak. By looking at the heat map of correlation for numerical attributes, we can see there's no highly correlated attributes, the age and maximum heart rate are negatively correlated.

preprocessing steps:

The categorical varaibles are represented by integer in the original dataset, we first convert it to factor. We checked there's no NA value. After that, we do train/test split which using %75 data as training set. Then, we create recipes:

1. performs a stepwise centering for numerical variables
2. sacle numerical variables
3. performs a stepwise feature selection operation based on the Near Zero Variance (NZV) criterion for numerical variables
4. one-hot endcoding for categorical variables.

```
# since categorical variable are represented by integers, we have to first convert it to charac
ter

hd[,"sex"] <- as.factor(hd$sex)
hd[,"cp"] <- as.factor(hd$cp)
hd[,"fbs"] <- as.factor(hd$fbs)
hd[,"restecg"] <- as.factor(hd$restecg)
hd[,"exng"] <- as.factor(hd$exng)
hd[,"slp"] <- as.factor(hd$slp)
hd[,"caa"] <- as.factor(hd$caa)
hd[,"thall"] <- as.factor(hd$thall)
hd[,"output"] <- as.factor(hd$output)
hd <- hd %>% drop_na()
hd |> glimpse()
```

```
## Rows: 303
## Columns: 14
## $ age      <dbl> 63, 37, 41, 56, 57, 57, 56, 44, 52, 57, 54, 48, 49, 64, 58, 5…
## $ sex      <fct> 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 1…
## $ cp       <fct> 3, 2, 1, 1, 0, 0, 1, 1, 2, 2, 0, 2, 1, 3, 3, 2, 2, 3, 0, 3, 0…
## $ trtbps   <dbl> 145, 130, 130, 120, 120, 140, 140, 120, 172, 150, 140, 130, 1…
## $ chol     <dbl> 233, 250, 204, 236, 354, 192, 294, 263, 199, 168, 239, 275, 2…
## $ fbs      <fct> 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0…
## $ restecg  <fct> 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1…
## $ thalachh <dbl> 150, 187, 172, 178, 163, 148, 153, 173, 162, 174, 160, 139, 1…
## $ exng     <fct> 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0…
## $ oldpeak  <dbl> 2.3, 3.5, 1.4, 0.8, 0.6, 0.4, 1.3, 0.0, 0.5, 1.6, 1.2, 0.2, 0…
## $ slp      <fct> 0, 0, 2, 2, 2, 1, 1, 2, 2, 2, 2, 2, 2, 1, 2, 1, 2, 0, 2, 2, 1…
## $ caa      <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0…
## $ thall    <fct> 1, 2, 2, 2, 2, 1, 2, 3, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3…
## $ output   <fct> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1…
```

```
# split data into train and test data
split <- initial_split(hd, prop = 0.75, strata = sex)
train_data <- training(split)
test_data <- testing(split)



# preprocessing data using recipe
hd_recipe <- (
  recipe(output ~., data = train_data)
  |> step_center(all_numeric())
  |> step_scale(all_numeric())
  |> step_nzv(all_numeric())
  |> step_dummy(all_nominal(), one_hot = TRUE,-output)
)

hd_prepped <- prep(hd_recipe)

hd_prepped |> bake(train_data) |> rsample::vfold_cv(v=10)
```

splits	id
<list>	<chr>
<S3: vfold_split>	Fold01
<S3: vfold_split>	Fold02
<S3: vfold_split>	Fold03
<S3: vfold_split>	Fold04
<S3: vfold_split>	Fold05
<S3: vfold_split>	Fold06
<S3: vfold_split>	Fold07
<S3: vfold_split>	Fold08
<S3: vfold_split>	Fold09
<S3: vfold_split>	Fold10

1-10 of 10 rows

Model

The model I choose is random forest since the data has many features. And from previous step, we found that there may be some outliers in the data, random forest are good at handling them. Random Forest can provide feature importance measures, which can help understand which features are driving the predictions.

```
model_decision <- parsnip::rand_forest(mtry=tune(), min_n = tune(), trees = 1000) |>
  set_engine("ranger") |>
  set_mode("classification")
```

```
#create the work flow
hd_wflow <- workflow() |>
  add_recipe(hd_prepped) |>
  add_model(model_decision)
```

Parameter tuning

I tuned hyperparameters “mtry” and “min_n” in random forest. mtry controls the number of features that are used at each split of the decision tree. And “min_n” controls the minimum number of observations in the leaf of the tree.

```
tt<-tune_grid(
  hd_wflow,
  grid = 100,
  resamples = vfold_cv(train_data)
)
```

```
## i Creating pre-processing data to finalize unknown parameter: mtry
```

```
#select best parameter using accuracy
show_best(tt,metric = "accuracy")
```

m…	mi…	.metric	.estimator	mean	n	std_err	.config
<int>	<int>	<chr>	<chr>	<dbl>	<int>	<dbl>	<chr>
24	11	accuracy	binary	0.8685771	10	0.02401951	Preprocessor1_Model37
26	15	accuracy	binary	0.8685771	10	0.02401951	Preprocessor1_Model98
4	5	accuracy	binary	0.8681818	10	0.02051289	Preprocessor1_Model96
30	5	accuracy	binary	0.8642292	10	0.02351336	Preprocessor1_Model01
18	18	accuracy	binary	0.8642292	10	0.02439028	Preprocessor1_Model04

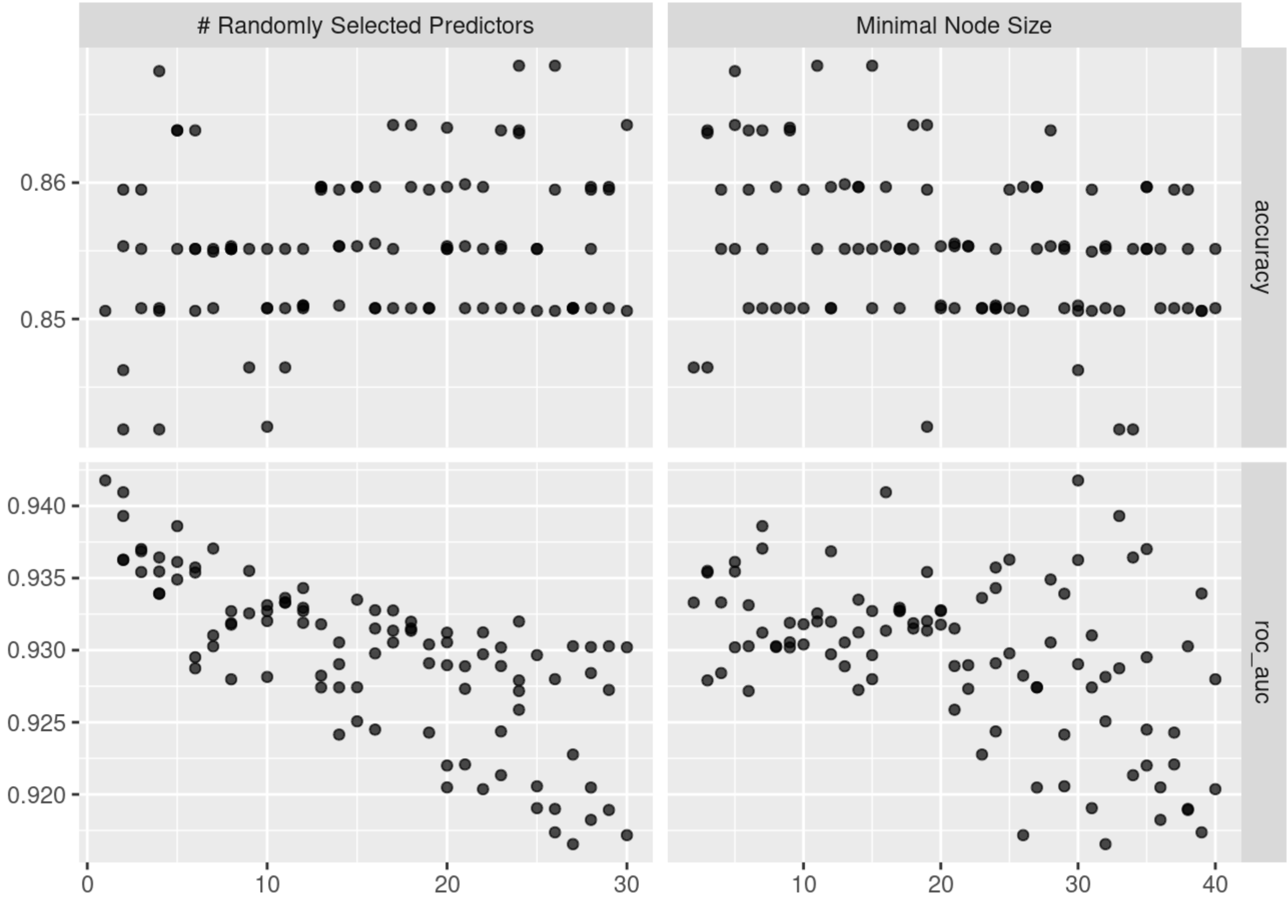
5 rows

```
#select best parmeter using roc
show_best(tt,metric = "roc_auc")
```

m…	mi…	.metric	.estimator	mean	n	std_err	.config
<int>	<int>	<chr>	<chr>	<dbl>	<int>	<dbl>	<chr>
1	30	roc_auc	binary	0.9417685	10	0.01694481	Preprocessor1_Model55
2	16	roc_auc	binary	0.9409500	10	0.01635997	Preprocessor1_Model91
2	33	roc_auc	binary	0.9393047	10	0.01699927	Preprocessor1_Model13
5	7	roc_auc	binary	0.9386015	10	0.01687916	Preprocessor1_Model97
7	7	roc_auc	binary	0.9370482	10	0.01791022	Preprocessor1_Model58

5 rows

```
autoplot(tt)
```



Fitting the best model

Based on my our hyperparamter turing, we choose the highest roc_auc as the best parameter and refit to the data.

```
final_hd <- hd_wflow |> finalize_workflow(select_best(tt,metric = "roc_auc"))
final_hd
```

```
## ══ Workflow ════════════════════════════════════════════════════════════════════════════════
## Preprocessor: Recipe
## Model: rand_forest()
##
## ── Preprocessor ────────────────────────────────────────────────────────────────────────────
## 4 Recipe Steps
##
## • step_center()
## • step_scale()
## • step_nzv()
## • step_dummy()
##
## ── Model ───────────────────────────────────────────────────────────────────────────────────
## Random Forest Model Specification (classification)
##
## Main Arguments:
##   mtry = 1
##   trees = 1000
##   min_n = 30
##
## Computational engine: ranger
```

```
# last fit
hd_fit <- last_fit(final_hd,split)
hd_fit
```

splits <list>	id <chr>	.metrics <list>	.notes <list>	.predictions <list>
<S3: initial_split>	train/test split	<tibble[,4]>	<tibble[,3]>	<tibble[,6]>

1 row

```
# metric on the test set
collect_metrics(hd_fit)
```

.metric <chr>	.estimator <chr>	.estimate <dbl>	.config <chr>
accuracy	binary	0.7631579	Preprocessor1_Model1
roc_auc	binary	0.8556548	Preprocessor1_Model1

2 rows

Variable Importance plot

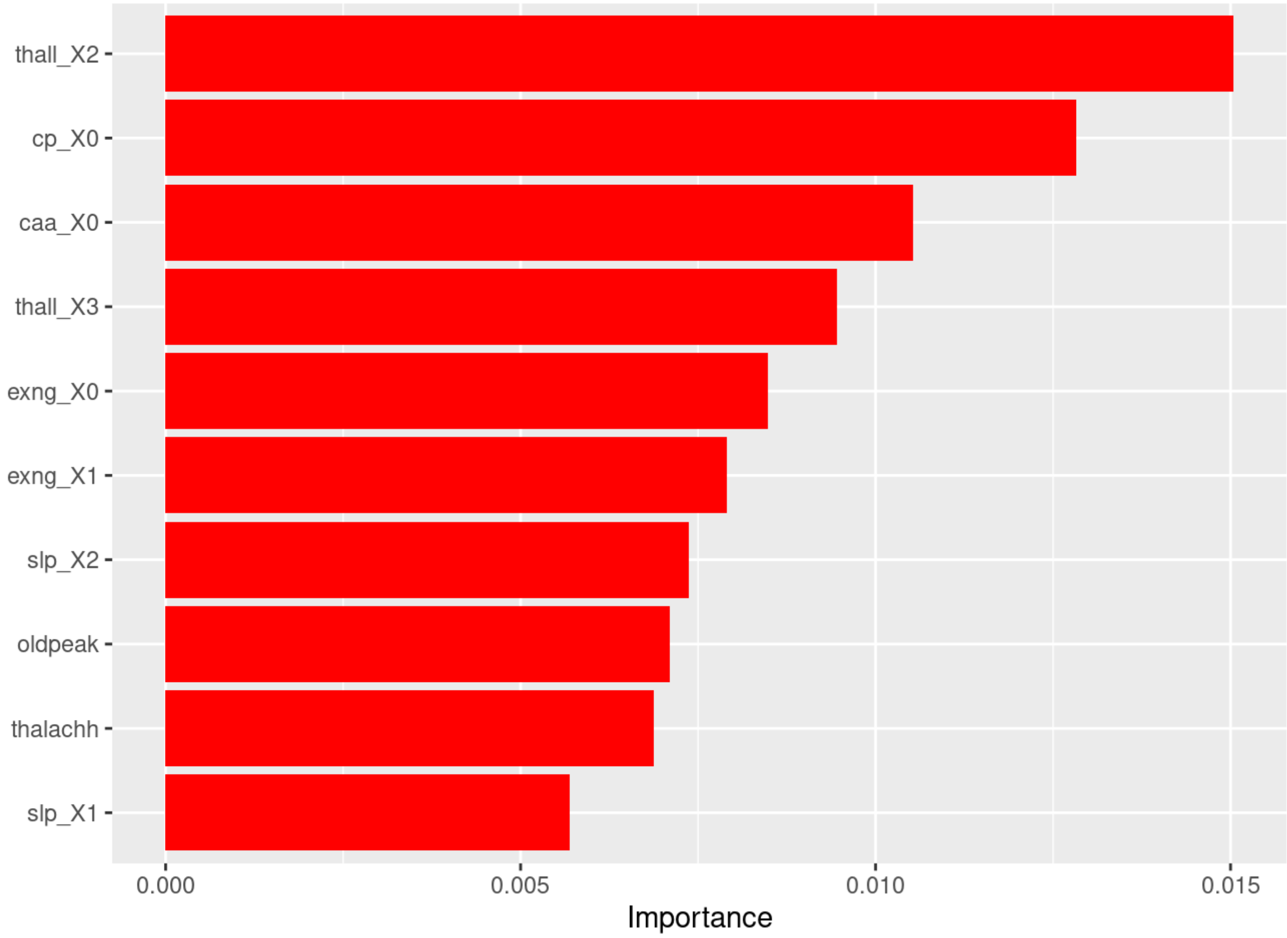
Then we plot the variable importance plot, we can see is the most important factor.

```
# fit model again, set imprtance = "permutation"
imp_hd <- model_decision |>
  finalize_model(select_best(tt)) |>
  set_engine("ranger", importance = "permutation")
```

```
## Warning: No value of `metric` was given; metric 'roc_auc' will be used.
```

```
workflow() |>
  add_recipe(hd_recipe) |>
  add_model(imp_hd) |>
  fit(train_data) |>
  pull_workflow_fit() |>
  vip(aesthetics = list(fill = "red"))
```

```
## Warning: `pull_workflow_fit()` was deprecated in workflows 0.2.3.
## ℹ Please use `extract_fit_parsnip()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



reference: https://juliasilge.com/blog/ikea-prices/ (https://juliasilge.com/blog/ikea-prices/)